百度智能云

OCR 文档



【版权声明】

版权所有©百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司。 未经本公司书面许可，任何单位和个人不得擅自摘抄、复制、传播本文档内容，否则本公司有权依法追究法律责任。

【商标声明】



和其他百度系商标，均为百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司的商标。 本文档涉及的第三方商标，依法由相关权利人所有。未经商标权利人书面许可，不得擅自对其商标进行使用、复制、修改、传播等行为。

【免责声明】

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导。 如您购买本文档介绍的产品、服务，您的权利与义务将依据百度智能云产品服务合同条款予以具体约定。本文档内容不作任何明示或暗示的保证。

目录

目录 2
功能发布记录 8
快速入门 16
新手操作指引 16
如何用Postman调用OCR服务 19
如何用代码调用OCR服务 22
购买指南 25
计费概述 25
免费测试资源 26
产品价格 35
共享资源包 35
通用场景文字识别 37
卡证文字识别 49
交通场景文字识别 62
财务票据文字识别 70
医疗票据文字识别 79
教育场景文字识别 84
其他场景文字识别 85
智能文档分析平台 87
自定义开发平台 88
文档图像处理 90
费用计算示例 94
如何购买 95
API文档 103
简介 103
调用方式 108
通用场景文字识别 110
通用文字识别（高精度版） 110
通用文字识别（标准含位置版） 127
表格文字识别V2 159
印章识别 164
数字识别 169
二维码识别 172
智能结构化 176
接口描述 176
卡证文字识别 197
身份证识别 197
身份证混贴识别 206
身份证识别（金融加密版） 211
银行卡识别 214

营业执照识别 217
护照识别 223
护照识别（港澳台地区及境外） 227
接口描述 227
出生医学证明识别 244
企业工商信息查询（标准版） 248
企业工商信息查询（高级版） 251
企业二要素核验 263
企业三要素核验 264
企业四要素核验 266
港澳台证件识别 268
接口描述 268
驾驶证识别 337
车辆证照混贴识别 343
车牌识别 349
VIN码识别 352
在线调试 353
机动车销售发票识别 355
二手车销售发票识别 358
车辆合格证识别 362
机动车登记证书识别 365
磅单识别 368
增值税发票识别 413
增值税发票验真 420
银行回单识别 435
接口描述 435
火车票识别 452
出租车票识别 456
飞机行程单识别 459
汽车票识别 465
网约车行程单识别 479
词典笔文字识别 605
文档图像处理 622
文档矫正增强 622
图片去摩尔纹 625
文档图像去底纹 627
文件检测分类 627
错误码 634
常见问题 636
售前咨询 636
产品使用问题 638

计费问题 640
HTTP-SDK文档 641
Python语言 641
简介 641
快速入门 643
接口说明 644
错误信息 794
Java语言 796
简介 796
快速入门 800
接口说明 802
错误信息 945
PHP语言 947
简介 947
快速入门 949
接口说明 950
错误信息 1030
C#语言 1032
简介 1032
快速入门 1035
接口说明 1036
错误信息 1104
Node.js语言 1106
简介 1106
快速入门 1108
接口说明 1110
错误信息 1197
C++语言 1199
简介 1199
快速入门 1201
接口说明 1202
出租车票识别 1238
VIN码识别 1239
火车票识别 1240
数字识别 1242
错误信息 1290
Android SDK 1292
简介 1292
快速入门 1296
接口调用说明 1300

FAQ 1337
错误码表 1337
SDK合规使用指南 1339
iOS SDK 1341
简介 1341
版本更新记录 1343
快速入门 1343
SDK集成图文教程 1345
SDK接口调用 1350
FAQ 1354
错误码表 1354
SDK合规使用指南 1356
离线SDK 1358
概览 1358
试用或购买授权 1358
授权方式 1359
错误码 1363
常见问题 1363
智能文档分析平台 1364
产品简介 1364
快速入门 1365
API参考 1377
功能发布记录 1377
API文档 1378
错误码 1426
购买指南 1427
产品价格 1427
iOCR自定义文字识别 1429
更新记录 1429
概览 1430
iOCR全场景识别 1435
简介 1435
API文档 1436
iOCR通用版 1447
简介 1447
使用流程 1449
API文档 1457
iOCR财会版 1461
简介 1461
使用流程 1464
API文档 1472

常见问题 1486
错误码 1489
EasyDL OCR 1490
产品介绍 1491
API文档 1491
请求说明 1491
错误码 1493
私有化部署服务 1494
产品介绍 1494
部署流程 1498
部署前环境检查（必看） 1498
机器指纹提取 1499
基于docker容器化部署说明 1501
运维手册 1516
修改模型服务端口号 1516
扩充实例步骤 1517
鉴权服务重启 1520
修改容器网络模式 1520
License更新步骤 1520
修改docker的默认存储路径 1521
机器指纹鉴权切换加密狗硬件鉴权 1522
常见问题 1523
安装部署问题排查 1523
鉴权服务排查 1526
视频专区 1529
产品介绍 1529
操作指南 1529
常见问题 1533
前期准备 1533
调用操作 1533
查询调用情况 1533
历史版本 1535
表格文字识别(同步接口) 1535
表格文字识别(异步接口) 1536
多卡证类别检测 1541
彩票识别 1546
保险单识别 1549
台湾通行证识别 1552
港澳通行证识别 1555
名片识别 1557

健康码识别 1559
核酸证明识别 1563
通用票据识别 1565
HTTP-SDK文档 1568
Python语言 1568
Java语言 1576
PHP文档 1584
C#语言 1592
Node.js语言 1600
C++语言 1608
文档解析（旧接口） 1616

功能发布记录

发布时间	更新日志
2025-06-24	**文档解析新增mhtml格式解析及多语种识别：**可对 mhtml 格式的流式文档进行解析，输出文档的版面等信息；可支持识别中、英、日、韩、法等 20+ 语言类型
2025-06-04	**企业四要素核验正式商用，**企业工商信息核验场景新增企业四要素核验，通过校验企业名称、统一社会信用代码、法人姓名、注册证件号的一致性，验证企业工商信息，快速核验企业资质。可开通按量后付费、购买次数包。详情参见价格文档
2025-01-02	**离婚证识别正式开放公测，离婚证识别**支持对离婚证进行结构化识别，包括**姓名_男**、**身份证件号_男**、**出生日期_男**、**国籍_男**、**性别_男**、**姓名_女**、**身份证件号_女**、**出生日期_女**、**国籍_女**、**性别_女**、**离婚证字号**、**持证人**、**备注**、**登记日期**，全部14 个字段，详情参见技术文档
2024-09-19	**文件检测分类正式商用**，可开通按量后付费或购买预付费资源包，按次计费，详情参见价格文档
2024-08-07	**智能结构化正式商用**，可开通按量后付费或购买预付费资源包，按次计费，详情参见价格文档
2024-06-13	**文档解析正式开放公测**，支持对doc、xlsx、pdf、图片等16种格式文档进行解析，输出文档的版面、表格、阅读顺序、标题层级等信息，详情参考技术文档
2024-05-21	**房产证识别正式商用**，支持识别不动产证、房屋所有权证、村屋所有权证等多种房产证，详情参见价格文档
2024-04-24	**身份证识别新增PS判断功能**，可支持从图像维度判断图片是否被PS，详情参见技术文档
2024-03-20	**食品经营许可证、食品生产许可证识别正式开放公测**，两种证件均可支持识别 14 个字段信息，详情参见技术文档
2024-03-15	**智能结构化正式开放邀测**，支持智能提取图片中的字段结构化信息，适用于各类证照、票据、表单等版式中的结构化信息录入场景，详情参见技术文档
2024-03-13	**外国人永久居住证识别正式开放邀测**，可支持识别 Name、Nationality、Sex、出生日期、国籍等 11 个字段，详情参见技术文档
2024-02-07	**开户许可证识别正式开放公测**，可支持识别公司名称、开户银行、核准号、法人、编号、账号 6 个字段，详情参见技术文档
2024-01-10	**护照识别（港澳台地区及境外护照）正式开放公测**，可支持识别 MRZCode1、MRZCode2、国家码、国籍、姓名拼音、护照号等 10 个关键字段，详情参见技术文档
2023-12-28	**试卷切题识别上线邀测**，支持对图片/PDF格式文档内的题目自动切分与结构化识别，可按题输出题干、选项、答案等信息，适用于整页试卷、习题册、课本等，详情参见技术文档
2023-12-21	**房产证识别正式开放公测**，可支持识别不动产证、房屋所有权证、村镇房屋所有权证 3 类房产证件的 11 个关键字段，详情参见技术文档
2023-11-07	**医疗发票识别能力升级：**支持识别全国各地门诊/住院发票的全字段信息，新增返回各省直辖市的专有信息，支持收费项目信息的医保三目录信息核验，支持包括北京、上海、广州、深圳等21 个城市的医保三目录信息核验。详情参见技术文档
2023-09-13	**企业二要素核验正式上线商用：**企业工商信息核验场景新增企业二要素核验，通过核验统一企业名称、社会信用代码一致性，快速核验企业资质。可开通按量后付费、购买次数包。详情参见技术文档
2023-08-16	**图文转换器（接口版）正式商用：**可开通按量后付费、购买次数包，支持图文转换器的在线工具和接口调用两种使用方式。详情参见价格文档
2023-08-09	**企业工商信息查询（标准版）能力升级：**增加返回注册资本币种、组织机构代码、许可经营范围等9个字段，详情参见技术文档

	见技术文档
2023-07-07	**结婚证、港澳台证件识别正式商用：**可开通按量后付费、购买次数包，详情参见价格文档
2023-06-01	**通用文字识别高精度版、高精度含位置版、标准版、标准含位置版模型升级：支持OFD 格式文件**，在请求参数中传入"ofd_file" 字段，即可识别OFD 格式文档，详情参见技术文档
2023-05-31	**词典笔文字识别正式开放公测**：面向词典笔/扫读笔场景，提供文字扫描与识别能力。详情参见技术文档
2023-05-30	**表格文字识别V2模型升级：新增Excel文档还原功能，**在请求参数中传入“return_excel” 字段，即可将图片中的表格信息还原为Excel文档，输出Excel对应的base64编码，详情参见技术文档
2023-05-26	**图文转换器（接口版）正式开放公测：直接调用API对文件进行识别还原，方便二次开发。**可将图片/PDF文件，转换为保留原文档版式的Word、Excel文档输出，详情参见技术文档
2023-05-19	**增值税发票识别模型升级：支持OFD 格式文件**，在请求参数中传入"ofd_file" 字段，即可识别OFD 格式票据，详情参见技术文档
2023-05-18	**办公文档识别模型升级：新增表格识别和印章识别能力**，在请求参数中开启“recg_tables”、“recog_seal”字段，即可识别文档中的表格或印章信息，详情参见技术文档
2023-05-05	**快递面单识别模型升级：支持识别隐私面单**，开启是否识别隐私面单参数后，新增返回隐私面单相关的3个字段，详情参见技术文档
2023-04-10	**港澳台证件、结婚证识别正式开放公测：港澳台证件识别**支持识别4类港澳台出入境证件识别，包含港澳通行证正/反面、台湾通行证正/反面、台胞证（台湾居民来往大陆通行证）正/反面、返乡证（港澳居民来往内地通行证）正/反面，可支持识别以上4类证件的全部字段信息，详情参见技术文档；**结婚证识别**支持对结婚证进行结构化识别，包括姓名男、身份证件号男、出生日期男、国籍男、性别男、姓名女、身份证件号女、出生日期女、国籍女、性别女、结婚证字号、持证人、备注、登记日期，全部14 个字段，详情参见技术文档
2023-04-07	**银行回单识别正式商用：**可开通按量后付费、购买次数包，详情参见价格文档
2023-03-16	**文档矫正增强、文档去手写、图片去摩尔纹、文档图片去底纹4项文档图像处理能力全面商用：**可开通按量后付费、购买次数包、QPS叠加包，详情参见价格文档
2023-03-01	**银行回单识别全新公测：**支持对不同版式银行回单进行结构化识别，包括标题、付款人户名、付款人开户银行、付款人账号、收款人户名、收款人开户银行、收款人账号、大写金额、小写金额、流水号、回单编号、交易日期、摘要、用途 14个关键字段。详情参见技术文档
2022-12-28	**驾驶证识别模型升级：**新增支持识别电子驾驶证正页，包含生成时间、条形码下编号、证号、准驾车型、当前时间、初次领证日期、出生日期、姓名、性别、失效日期、有效起始日期、国籍、档案编号、状态、累积记分 15个字段，详情参见技术文档
2022-10-28	全新发布**企业工商信息查询（标准版）、企业工商信息查询（高级版）、企业三要素核验、企业四要素核验，**营业执照核验 4 项能力，面向企业工商信息识别场景客户，核查是否存在公司主体信息与法人信息不一致等异常情况，并对基于公司名称/社会统一信用代码，对企业的工商照面信息进行基础/详细查询，详情参见技术文档
2022-10-28	**病案首页、出院小结、医疗费用结算单、诊断报告单、 购物小票、社保卡、车辆证照混贴识别 7 项能力正式商用：**可开通按量后付费、购买次数包，详情参见价格文档
2022-09-28	**表格文字识别V2正式商用：**可开通按量后付费、购买次数包，详情参见价格文档
2022-09-28	**财务场景五项能力：定额发票、网约车行程单、过桥过路费发票、船票、汽车票识别正式商用：**可开通按量后付费、购买次数包，详情参见价格文档
2022-08-09	医疗票据场景，病历单据识别新增**入院识别、手术记录识别**2项能力，应用于智能理赔核保场景，详情参见技术文档
2022-08-02	**交通场景四项能力：二手车销售发票、快递面单、磅单、道路运输证识别正式商用：**可开通按量后付费、购买次数包，详情参见价格文档
	医疗诊断报告单上线邀测支持识别全国各地各医院医疗诊断报告单，包括医院名称、报告名称、姓名、性别、年

2022-07-13	**医疗诊断报告单上线邀测**支持识别全国各地各医院医疗诊断报告单，包括医院名称、报告名称、姓名、性别、年龄、科室、临床诊断、报告日期、检查部位、检查方法、检查所见、检查提示、建议、肉眼可见14个字段，详情参见技术文档
2022-07-08	医疗票据场景病历单据识别，全新发布**门诊病历识别、诊断证明识别、处方笺识别**三项能力，可支持识别全国各地各医院的病历单据，应用于智能理赔核保场景，详情参见技术文档
2022-07-04	全新发布**健康码识别、行程卡识别、电子核酸证明识别**三大防疫相关能力，覆盖全国各地不同版式，可结构化识别个人信息、健康状态和核验结果等，适用于出入口管理、员工考勤、门禁管控等场景。**企业认证即可享有不限量免费调用，**如需更高并发可付费扩充 QPS 或本地化部署。助力抗疫，共克时艰，百度AI与您同心向前，详情参见技术文档
2022-06-24	1. **文档矫正增强**新能力上线，可对图片中的文件、卡证、票据等内容进行四角点检测定位，提取主体内容并对其进行矫正，详情参见技术文档； 2. **文档去手写**新能力上线，可去除图片中的手写内容，详情参见技术文档； 3. **图片去摩尔纹**新能力上线，可去除翻拍电脑、手机等显示屏照片中的摩尔纹，使图片更加清晰，详情参见技术文档
2022-06-21	**增值税发票识别**新增判断印章是否存在的功能，并支持返回印章的具体文字信息，详情参见技术文档
2022-05-20	**医疗费用明细识别、医疗检验报告单识别2项能力正式商用：**可开通按量后付费、购买次数包，详情参见价格文档
2022-04-18	1. **表格文字识别V2**新能力上线，支持识别图片/PDF格式文档中的表格内容，返回各表格的表头表尾内容、单元格文字内容及其行列位置信息，全面覆盖各类表格样式，包括常规有线表格、无线表格、含合并单元格表格，且识别效果更优，详情见技术文档； 2. **二维码识别**新增位置信息返回，方便定位图中二维码位置进行二次处理，详情见技术文档； 3. **试卷分析与识别**能力全面升级，识别准确率大幅提升，且版面分析能力新增分栏、页眉、页脚、页码、脚注识别功能，详情见技术文档
2022-03-29	**通用文字识别（高精度版/高精度含位置版）**在原有中、英、日、韩、法、西、葡、德、俄等二十种语言文字识别基础上，新增支持识别**泰语、越南语、阿拉伯语、印第安语及部分中国少数民族语言**，详情见技术文档，也可在线试用
2022-03-28	**身份证混贴识别正式商用：**可开通按量后付费、购买次数包、QPS叠加包使用，详情参见价格文档
2022-03-03	**增值税发票验真能力升级：**新增支持区块链发票（深圳地区）、全电发票（增值税专用发票）、全电发票（普通发票）3类发票验真，详情参见技术文档
2022-03-02	**道路运输证识别上线邀测：**结构化识别道路运输证的业户名称、地址、车辆号牌、经营许可证、经济类型、车辆类型、吨座位、车辆规格、经营范围、初领日期、备注、发证日期等14个关键字段，支持识别横版及竖版两种道路运输证，详情参见技术文档
2022-02-25	**身份证识别模型升级：**新增质量检测告警，开启后可提醒身份证边框/四角不完整、身份证头像或关键字段被遮挡等告警提示，详情参见技术文档
2022-02-22	**行驶证、驾驶证、营业执照识别三项模型升级：**新增风险检测告警，开启后可提醒复印、翻拍、PS告警等，详情参见技术文档
2022-02-22	**购物小票识别上线邀测：**支持识别各类商场、超市及药店的购物小票，包括店名、小票号码、机器编号、工号、消费日期、消费时间、总金额、找零、币种、实收金额、优惠金额、打印日期、打印时间、明细商品名称、单价、数量、小计金额等信息，详情参见技术文档
2021-12-08	**出院小结识别上线邀测：**支持识别全国出院小结的科室、姓名、性别、年龄、入院日期、出院日期、住院天数、入院诊断、出院诊断、出院医嘱等关键字段，详情参见技术文档
2021-12-02	**快递面单识别上线邀测：**支持市面上常见版式的快递面单识别，结构化识别运单号、收/寄件人姓名、收/寄件人电话、收/寄件人地址等字段，详情参见技术文档
2021-11-26	**出生证明识别正式商用：**可开通按量后付费、购买次数包使用，详情参见价格文档

2021-11-11	**智能财务票据识别正式商用** 可开通按量后付费、购买次数包使用，详情参见 价格文档
2021-11-01	**医疗检验报告单上线邀测** 支持识别全国各地医疗检验报告单的姓名、性别、医院名称、报告单名称等关键字段，支持识别检查具体项目的项目名称、结果、单位、参考区间、结果提示等明细字段，详情参见 技术文档
2021-10-29	**行驶证、驾驶证识别模型升级** 新增质量检测告警，开启后可提醒证件边框不完整、遮挡提示告警，详情参见 技术文档
2021-10-29	**身份证识别模型升级** 新增支持身份证裁剪，返回身份证裁剪图的base64编码及位置信息，去掉证件外围多余背景、自动矫正拍摄角度，详情参见 技术文档
2021-10-29	**银行卡识别模型升级** 新增“持卡人”字段，可识别信用卡 持卡人姓名 英文字段，详情参见 技术文档
2021-10-15	**户口本识别新增户主页** 支持识别户口本内户主页的5个关键字段，详情参见 技术文档
2021-09-18	**社保卡识别上线邀测** 支持识别全国各地社保卡识别，详情参见 技术文档
2021-09-16	**智能票据识别上线公测** 支持财务场景中13种常见票据的分类及结构化识别，详情参见 技术文档
2021-09-02	**车辆证照混贴识别上线公测** 支持识别行驶证正副页、驾驶证正副页在同一张图片上的混贴场景，自动分类并检测识别所有字段，详情参见 技术文档
2021-08-30	**身份证识别接口支持AES加密** 支持对身份证图片及识别结果进行AES加密后传输，使用简单，加解密速度快，详情参见 技术文档
2021-08-26	**车牌识别Android离线SDK，发布6.0版本** 1. 支持旋转车牌的识别 2. 更新demo交互与设计，详情参见 技术文档
2021-08-19	**磅单识别上线邀测** 对磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段进行结构化识别。可用于货运及快递物流等场景，有效提升磅单数据的录入效率，详情参见 技术文档
2021-08-11	**机动车销售发票识别** 模型升级，“购买方名称”、“购买方身份证号码/组织机构代码”字段效果提升，专项优化新版机动车销售发票版式的识别效果，详情参见 技术文档
2021-07-27	**出生医学证明识别** 新增识别 出生体重、出生地点、母亲住址、父亲住址等17个字段，累计已支持23个字段的识别，详情参见 技术文档
2021-07-22	**医疗发票识别正式商用** 可开通按量后付费、购买次数包使用，详情参见 价格文档
2021-07-16	**车牌识别离线SDK发布Linux版本** 适用于ARM架构的Linux系统，支持Armv7/Armv8架构，请提交合作咨询 联系我们
2021-07-15	**通用文字识别离线SDK Windows平台发布C#语言版本** 适用于Windows7、Windows10系统，支持64位模式，采用C#调用C++的动态库dll的方式，实现C#语言版本，请提交合作咨询 联系我们
2021-07-15	**通用文字识别离线SDK Android平台发布V3.2版本** 基于更新文本检测模型与中文识别量化模型，提升识别效果，降低耗时，相同图片在相同环境下性能提升16%，可在 控制台 下载新版本
2021-07-14	**机动车登记证书识别正式商用** 可开通按量后付费、购买次数包使用，详情参见 价格文档
2021-06-29	**增值税发票识别** 新增“车牌号”、“类型”、“通行日期起”、“通行日期止”、“电子支付标识”5个字段，新增支持识别区块链发票、通用机打电子发票，详情参见 技术文档
2021-06-10	**驾驶证支持副页识别** 结构化识别机动车驾驶证副页的姓名、记录、证号、档案编号4个字段，详情参见 技术文档
2021-05-25	**身份证混贴识别上线邀测** 支持自动检测与识别身份证正反面在同一张图片上的场景，一次识别图片中身份证正反面所有字段，同时，支持对用户上传的身份证图片进行图像风险和质量检测，可识别图片是否为复印件或临时身份

05-25	证，是否被翻拍或编辑，是否存在正反颠倒、模糊、欠曝、过曝等质量问题，详情参见 技术文档
2021-05-14	**网约车行程单识别上线邀测** 支持识别各大主要服务商的网约车行程单，包括滴滴打车、花小猪打车、高德地图、曹操出行、阳光出行，支持识别服务商、行程开始及结束时间、车型、总金额等14个关键字段，详情参见技术文档
2021-05-11	**机动车登记证书识别上线邀测** 支持对机动车登记证书的编号、机动车所有人、登记机关、登记日期、登记编号、车辆类型等15个关键字段进行结构化识别，同时支持检测发证机关章，详情参见技术文档
2021-05-11	**驾驶证识别** 新增“发证单位”字段，可识别驾驶证正页左下角印章，详情参见技术文档
2021-05-07	**车牌识别Android离线SDK，发布5.0版本** 1. 新增支持白色、黑色车牌号码及颜色识别，优化车牌颜色识别效果 2. 支持返回车牌识别分数与质量控制分数，详情参见 技术文档
2021-04-20	**医疗费用明细识别上线邀测，**支持识别全国各地医疗费用明细小票的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目，详情参见技术文档
2021-04-20	**二手车销售发票识别上线邀测** 支持对二手车销售发票的发票代码、发票号码、开票日期、买方、卖方、车牌号、车辆类型、二手车市场等25个关键字段进行结构化识别，详情参见技术文档
2021-04-20	**行驶证识别** 新增“发证单位”字段，可识别行驶证正页左下角印章，详情参见技术文档
2021-04-14	**文字识别离线SDK上新** 新增支持身份证识别离线SDK安卓版本、通用文字识别离线SDK安卓版本，可在控制台试用与购买 立即使用
2021-04-06	**营业执照识别**新增次数包售卖方式，详情参见价格文档
2021-03-25	**增值税发票验真**正式商用，可开通按量后付费进行使用，详情参见价格文档
2021-03-24	**车牌识别模型升级** 1. 优化蓝牌、绿牌、黄牌等的颜色识别； 2.优化夜间监控场景下的车牌文字与颜色识别。详情参见技术文档
2020-03-23	**船票识别上线邀测，**对全国范围内不同版式的客运船票、货运船票进行结构化识别支持发票代码、发票号码、出发地点、到达地点等7个关键字段，详情参见技术文档
2021-03-16	**车牌识别模型全面升级** 1. 支持旋转车牌的识别，可识别旋转90度、180度、270度等角度的旋转车牌； 2.优化监控高拍等车牌较小场景下的车牌检测与识别，提升多车牌识别能力。详情参见技术文档
2021-02-04	**身份证识别（加密版）**正式商用，可开通按量后付费或购买次数包进行使用，详情参见价格文档
2021-02-02	**HTTP-SDK能力升级：**新增支持 增值税发票识别、出租车票识别、VIN码识别、火车票识别、数字识别 5个接口能力，服务端Java语言、Python语言、PHP语言、C++语言、C#语言、Node.js语言SDK均已支持，Android及iOS端已同步支持，详情参见 技术文档
2021-01-26	**医疗发票识别全面升级：** 1. 新增省市、医院名称、收款人、医保统筹支付、个人账户支付、预缴金额、补缴金额、退费金额、门诊号等通用字段；新增医疗发票项目明细识别，包括对项目大类、明细类别的字段识别；新增支持各省市特有字段的识别，根据不同省市返回对应的字段。 2. 整图字段识别准确率提升，北京/广东/河北/河南/江苏/山东/上海/天津/浙江地区识别效果优化。详情参见 技术文档
2021-01-25	**行驶证识别模型升级：**行驶证正页注册日期、发证日期字段识别优化，车辆类型字段针对货车相关车辆类型识别优化，整图字段识别准确率提升，详情参见 技术文档
2021-01-18	**行驶证识别模型升级：**行驶证副页检验记录字段识别优化，详情参见 技术文档

2021-01-14	**营业执照识别模型升级：**各字段识别准确率提升，字段平均识别准确率提升5%，经营范围字段准确率大幅度提升，详情参见 技术文档
2020-12-30	**通用文字识别Windows离线SDK，发布3.0版本** 1.优化文字识别性能，耗时降低约1/2 2.优化文字识别效果，准确率提升约20%，详情参见 技术文档
2020-12-29	**4款财务票据识别能力全面升级** 1. 增值税发票识别：增值税卷票新增省、市字段识别，累计已支持21个字段，详情参见 技术文档 2. 行程单识别：新增客票级别、座位等级、销售单位号、签注、免费行李、验证码字段识别，累计已支持24个字段，详情参见 技术文档 3. 定额发票识别：新增发票所在地、发票金额小写、省、市字段识别，累计已支持7个字段，详情参见 技术文档 4. 通行费发票：新增省、市字段识别，累计已支持9个字段，详情参见 技术文档
2020-12-24	**驾驶证识别能力优化：**初次领证日期、有效期限字段识别优化，空白、遮挡情况下的识别优化
2020-12-18	**户口本识别正式商用：**可开通按量后付费、购买次数包使用，详情参见价格文档
2020-12-18	**驾驶证识别优化拒识别能力：**对于非驾驶证的图片不返回识别结果，返回“图片目标识别错误，请确保图片中包含对应行驶证”提示，提升产品易用性，不影响对驾驶证的识别效果
2020-12-02	**行驶证识别优化拒识别能力：**对于非行驶证的图片不返回识别结果，返回“图片目标识别错误，请确保图片中包含对应行驶证”提示，提升产品易用性，不影响对行驶证的识别效果
2020-12-01	**身份证识别模型架构升级：**各字段准确率提升，住址字段准确率提升3.4%，姓名、出生日期、身份证号、性别、民族、签发日期、失效日期关键字段准确率高于99%；支持自动检测身份证正反面，支持自动检测图像旋转角度，详情参见 技术文档
2020-11-28	**12项OCR API支持图片URL格式** 行程单、汽车票、试卷分析与识别、网图含位置、名片、保单、仪器仪表、多卡证类别检测、医疗发票、结算单、病案首页、智能结构化识别，以上12个接口支持图片URL传参方式，和base64编码二选一，详情参见 技术文档
2020-11-25	**通用文字识别Windows离线SDK，发布2.1版本** 支持单台设备在线/离线鉴权、批量设备鉴权，详情参见 技术文档
2020-11-23	**车牌识别Android离线SDK，发布3.1版本** 1.支持单台设备在线/离线鉴权、批量设备鉴权 2.包大小减少1/2，详情参见 技术文档
2020-11-17	**银行卡升级新架构，采用新的检测和识别模型：**银行卡识别模型性能大幅度提升，相比原架构提升135% **银行卡识别接口新增发卡行及卡类型信息：**新增700+条发卡行信息，新增“准贷记卡”、“预付费卡”卡类型，原“信用卡”拆分为“贷记卡”及“准贷记卡”
2020-11-06	**8项OCR API支持图片URL格式** 身份证、行驶证、营业执照、增值税发票、车牌、定额发票、户口本、驾驶证，以上8个接口支持图片URL传参方式，和base64编码二选一，详情参见 技术文档
2020-11-05	**4款财务票据识别能力全面升级** 1. 增值税发票：新增省、市、密码区、代开字段识别，累计已支持32个字段，详情参见 技术文档 2. 出租车票：新增省、市、单价、里程字段识别，累计已支持12个字段，详情参见 技术文档 3. 汽车票：新增开票日期字段识别，累计已支持10个字段，详情参见 技术文档 4. 通用机打发票：新增省、市、开票时间、联次字段识别，累计已支持23个字段，详情参见 技术文档
2020-10-15	**行驶证识别模型升级**，“所有人”及“住址”字段准确率提升 **驾驶证识别模型升级**，“有效期限”及“住址”字段准确率提升
2020-09-24	**户口本识别模型全面升级，新增 15 个字段识别支持，**包括户号、本市县其他住址、曾用名、籍贯、宗教信仰、身高、血型、文化程度、婚姻状况、兵役状况、服务处所、职业、何时由何地迁往本市、何时由何地迁往本址、登记日期，现已支持识别户口本内页（常住人口登记卡）全部字段，详情参见 技术文档
	3 款文字识别能力正式商用:

2020-09-08	1. **机动车销售发票识别**正式商用，可开通按量后付费或购买次数包进行使用，详情参见价格文档 2. **车辆合格证识别**正式商用，可开通按量后付费或购买次数包进行使用，详情参见价格文档 3. **通用机打发票识别**正式商用，可开通按量后付费或购买次数包进行使用，详情参见价格文档 **4 款文字识别能力全面升级：** 1. **混贴票据识别**模型升级，新增支持汽车票、通行费发票混贴识别能力，详情参见技术文档 2. **增值税卷票** 全面升级，各字段识别准确率大幅度提升，同时新增支持「收款人」字段，详情参见技术文档 3. **通用机打发票** 关键字段准确率提升，同时**新增13个字段**：机打代码、机打号码、校验码、购买方名称、购买方纳税人识别号销售方名称、销售方纳税人识别号、商品单位、单价、数量、金额、合计金额大/小写，详情参见技术文档 4. **护照识别** 模型升级，新增4个字段：护照类型、国籍、MRZCode1、MRZCode2，详情参见技术文档
2020-08-25	**营业执照识别**模型全面升级，支持不同年份不同版式执照识别，各字段识别准确率大幅度提升，平均响应时长缩短一半
2020-08-13	**智能结构化识别上线邀测，**无需配置结构化对应关系、无需提取关键词、无需定制开发，对于各类卡证、票据直接上传图片即可获得结构化识别信息，详情参见技术文档
2020-08-12	**3 款医疗票据识别能力上线邀测：** 1. **医疗发票识别：**支持识别全国各地门诊/住院发票的业务流水号、发票号、住院号、病例号、姓名、性别、社保卡号、金额大/小写等 16 个关键字段，详情参见技术文档 2. **医疗费用结算单识别：**支持识别全国医疗费用结算单（包含费用汇总信息，非药品/项目列举清单）的姓名、出/入院时间、发票总金额、自费金额、医保支付金额等 6 个关键字段，详情参见技术文档 3. **病案首页识别：**支持识别全国各地病案首页的病案号、姓名、性别、出生日期、身份证号、出/入院科别、住院次数、药物过敏情况等 15 个关键字段，详情参见技术文档
2020-08-11	**车牌识别** 模型升级，在常见蓝牌、绿牌、黄牌的基础上，新增支持识别大型新能源牌（黄绿牌）、领使馆车牌、警牌、武警牌、军牌、港澳牌、民航车牌等特殊车牌
2020-08-05	**银行卡识别** 模型升级，卡号字段识别准确率进一步提升，平均响应时长缩短一半
2020-07-02	iOCR自定义模板文字识别「**自定义字段类型**」功能全面升级，可针对填写的字段词典自动生成虚拟训练集，并训练专属切片模型，可用于自定义模板时对识别区字段类型的调整，提高字段识别准确率，iOCR通用版 及 iOCR财会版 均已升级，可进行试用
2020-06-30	1. **增值税卷票识别** 全面升级，各字段识别准确率大幅度提升，同时新增支持「销售方名称」、「购买方名称」2 个字段的识别，详情参见技术文档 2. **行程单识别** 升级，新增多航班信息识别，详情参见技术文档
2020-06-28	1. **通行费发票识别** 上线邀测，支持对全国范围不同版式过路、过桥费发票的发票代码、发票号码、入口、出口、日期、时间、金额 7 个字段进行结构化识别，详情参见技术文档 2. **汽车票识别** 上线邀测，支持对全国范围不同版式汽车票的发票代码、发票号码、到达站、出发站、日期、时间、金额、身份证号、姓名 9 个字段进行结构化识别，详情参见技术文档
2020-06-27	**多卡证类别检测** 上线邀测，支持对同一张图片中的身份证正反面、行驶证正副页、驾驶证正副页、银行卡、营业执照等 8 种卡证进行类别检测及定位，详情参见技术文档
2020-06-03	网络图片文字识别（含位置版）正式商用，可开通按量后付费、购买次数包、购买QPS叠加包使用，详情参见价格文档。服务支持识别艺术字体或背景复杂的文字内容，除文字信息外，还可返回文字的位置信息、行置信度、单字符内容和位置等，详情参见技术文档
2020-05-20	仪器仪表盘读数识别上线邀测，支持不同品牌、不同型号的仪器仪表盘识别，适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号，支持液晶屏、字轮表等表型，详情参见技术文档
2020-05-15	1.增值税发票、出租车票、火车票、行程单、通用机打发票、定额发票整体能力升级，识别效果大幅度提升。 2.行程单新增 **保险费、订票渠道、身份证号、时间、印刷序号、承运人**6个字段识别。详情参见技术文档 3.火车票新增 **身份证号、售站、序列号、时间**4个字段识别。详情参见技术文档
2020-	文档版面分析与识别上线邀测，可对文档版面进行分析与识别，支持中、英文两种语言文字识别，纯手写、纯印刷

2020-05-11	和手写印刷混排情况下的文字识别；同时支持文档的版面分析，输出图、表、标题、文本的位置和信息，详情参见技术文档
2020-05-10	1.表格同步识别效果显著提升 2.二维码、护照识别性能提升
2020-04-09	通用文字识别（高精度版/高精度含位置版）全面升级，新增语种自动检测功能，请求参数中添加“language_type=auto_detect”即可开启功能，无需指定语种即可对20种语言及混排情况进行自动检测并识别，详情参见技术文档
2020-03-27	1. 行程单识别正式商用，可开通按量后付费或购买次数包进行使用，详情参见价格文档 2. 银行卡、驾驶证、车牌、手写文字识别、iOCR通用版支持 QPS叠加包售卖，详情参见价格文档 3. 机动车销售发票、公式识别开放公测，注册即享免费额度，详见免费额度
2020-03-26	1. iOCR通用版及iOCR财会版整体升级，新增预置模板功能及关键词辅助分类功能： - **预置模板**类型提升至20余种.无需上传训练集训练，直接选择系统预置模板即可识别。 - 新增**关键词辅助模板分类**，进一步提升模板分类准确率，实现对不同版式图片的自动分类和结构化识别。 2. OCR 全部接口增加对 PNM/TIFF/WebP 格式图片识别支持，图片请求限制详情参照技术文档 3. 行驶证识别增加输出字段归一化处理功能，可统一不同版本行驶证正页输出字段名，方便进行后续解析，详情参照技术文档
2020-03-06	护照识别全面升级，识别准确率大幅度提升，新增 **签发机关** 字段识别支持，并优化返回字段格式，详情参照技术文档
2020-01-17	通用机打发票识别准确率提升
2020-01-02	1. 机动车销售发票识别新增 **购方名称、购方纳税人识别号、车辆类型、产地、合格证号、销方电话、销售方银行账号、销售方地址、销售方开户银行、税务机关、税务机关代码、限乘人数** 等12个字段识别支持，并新增方向检测功能，可对旋转图片进行自动矫正，详见 技术文档 2. 车辆合格证识别新增 **发证日期、车辆制造企业名称、车辆型号、车身颜色、排量、轴距、转向形式、总质量、半挂车鞍座最大允许总质量、准乘人数、最高设计时速、制造日期** 等12个字段识别支持，并新增方向检测功能，可对旋转图片进行自动矫正，详见 技术文档 3. 公式识别识别准确率提升 4. 增值税发票识别准确率提升
2019-12-26	1. 新增QPS叠加包计费方式，支持已开通付费的接口扩充 QPS 上限，可以更高并发数进行请求（调用量照常计费），现已支持 **通用文字识别 4 个版本及身份证识别、数字识别** 共 6 款产品，更多产品陆续接入中 2. 行程单识别准确率提升
2019-12-19	1. 身份证识别 **住址** 字段识别准确率提升 2. 行驶证正页识别准确率提升
2019-12-06	护照识别、VIN码识别、二维码识别正式商用，可开通按量后付费或购买次数包进行使用，详情参见价格文档
2019-11-21	1. 通用文字识别（高精度版/高精度含位置版）支持小语种识别，目前支持20种语言，可加入请求参数“language_type=语种类型”进行控制，支持语种列表及对应代码参照 技术文档 2. 增值税发票识别支持增值税卷票识别，添加请求参数 type=roll 即可，详情参照 技术文档 3. 网络图片文字识别模型升级，识别准确率大幅度提升，针对电商场景图片提升显著 4. VIN码识别升级，识别准确率大幅度提升
2019-11-07	1. 身份证、银行卡、iOCR、增值税发票、数字识别、手写文字识别增加 次数包售卖方式 2. 身份证识别新增头像检测功能，可输出头像部分 base64 编码及位置，需增加请求参数：detect_photo=true，详情参照 技术文档 3. 行程单识别模型升级，识别准确率提升约 30%，并增加票价、民航发展基金、燃油附加费、其他税费、填开日期 5 个新字段，详情参照 技术文档
2019-10-26	火车票识别模型升级，识别准确率提升约 20%，各字段平均识别准确率可达 97% 以上

2019-09-21	新增次数包售卖方式，通用文字识别（高精度版）及通用文字识别（高精度含位置版）已支持 购买
2019-06-05	新增 iOCR财会版。iOCR财会版是一款针对各类财务票据，如常用报销发票、银行回单、对账单进行自动分类及结构化识别，并支持用户为特殊票据/单据创建结构化模板或分类器的自定义OCR平台

快速入门

新手操作指引

本文主要介绍如何快速开通文字识别服务，并完成接口调用。

一、注册及实名认证

使用百度智能云文字识别服务前，您需要一个百度智能云账号并完成实名认证。具体操作如下：

1. 注册并登录百度智能云平台，请参考注册和登录。个人用户可以直接使用自己的百度账号进行登录，企业用户建议注册账号，避免后续人员变动带来的账号归属问题。
2. 完成实名认证，操作细节请参考实名认证。只有完成了实名认证才能购买并使用文字识别服务。

二、领取免费测试资源

登录并进入文字识别控制台。如果您的账号已经完成了实名认证，系统会自动为您的账号发放免费测试资源。测试资源会在您进入控制台后约10分钟内发放。

领取到的免费资源可以在资源列表里查看。

三、创建应用

应用是调用服务的主体，您可以划定该应用有权限调用的接口范围，以此实现权限管理。后续也可以查看每个应用上产生的接口调用量。

方式一：在概览页上快速创建应用

进入文字识别控制台，点击使用指引模块的快速接入服务按钮。





应用名称和应用描述应当尽量反应应用的实际用途，方便您后续管理应用。在服务接口列表勾选您希望该应用能够调用的接口，勾选完毕后点击"立即创建"。

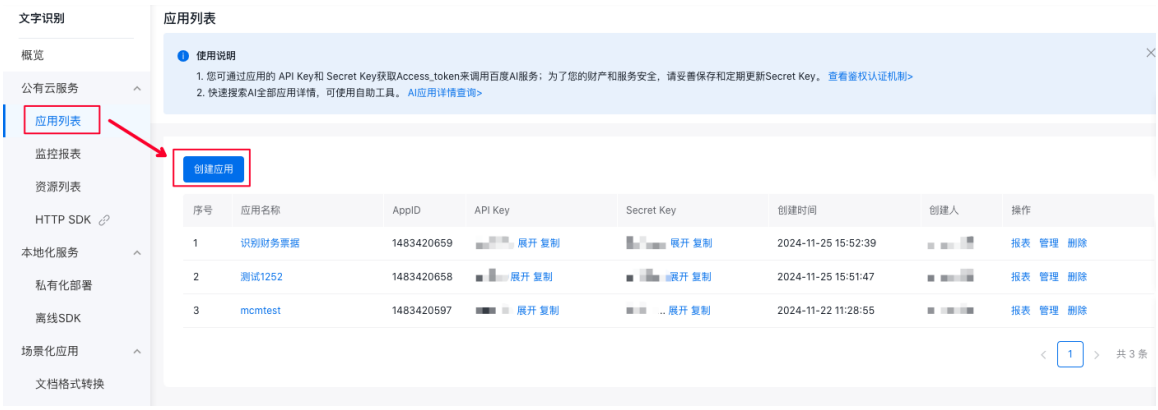


创建成功！API Key和Secret Key是您调用该应用内接口的凭证，如果泄露会导致资源被盗刷，请妥善保管，避免外泄。

本方式创建的应用只能选择当前产品方向的服务接口。如果您希望创建跨产品方向的应用，请参考下面的方式二。

方式二：在应用管理页创建应用

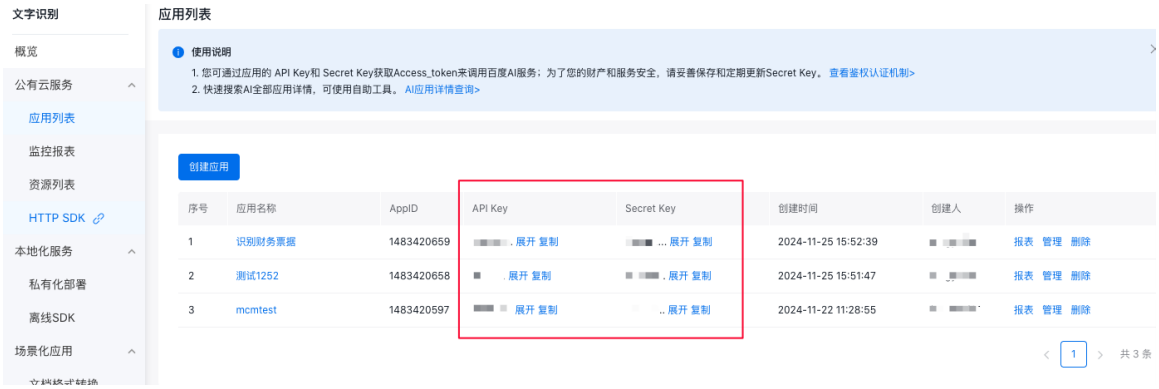
进入文字识别控制台，选择左侧导航的"应用列表"，点击"创建应用"。





应用名称和应用描述应当尽量反应应用的实际用途，方便您后续管理应用。在服务接口列表勾选您希望该应用能够调用的接口，勾选完毕后点击"立即创建"。

创建成功后，您可以在应用列表页里查看应用的API Key和Secret Key。API Key和Secret Key是您调用该应用内接口的凭证，如果泄露会导致资源被盗刷，请妥善保管，避免外泄。



四、调用接口服务

您可以根据以下介绍选择合适的使用方式：

- **通过 百度智能云 - 示例代码中心 在线调用文字识别服务 API**

如果您是开发初学者，对HTTP请求与API调用有一定的了解，您可以通过此方式快速体验文字识别服务。该方式无需编码，只需要输入相关参数，即可在线调用API，并查看返回结果。视频教程请参见如何使用 API Explorer 调用API接口（视频版）。

- **通过可视化工具（如 Postman）调用文字识别服务 API**

如果您是开发初学者，熟悉HTTP请求与API调用，您可以通过 Postman 调用、调试 API。具体请参见如何使用 Postman 调用文字识别服务。

- **通过编写代码调用文字识别服务 API**

如果您是开发工程师，熟悉代码编写，您可以通过编写代码的方式调用文字识别服务。具体请参见如何使用代码快速调用文字识别服务 API。

- **通过软件开发工具包（HTTP-SDK）调用文字识别服务**

如果您是开发工程师，熟悉代码编写，您可以通过已编写好的软件开发工具包（HTTP-SDK）来调用文字识别服务 API 。SDK 已支持多种语言，包括 Java、 Python、PHP、C++、C#、NodeJS、Android ADK、iOS SDK 等。您可点击下载对应的 SDK。

了解更多

示例源代码

您可以在我们的官方 github 上下载示例源码。

`https://github.com/Baidu-AIP/QuickStart/tree/master/OCR`

更多参考

- 文字识别API文档
- 如何获取 API Key 和 Secret Key

如何用Postman调用OCR服务

如何使用 Postman 调用文字识别服务 API

本文提供通过可视化工具 Postman 调用 OCR 通用文字识别（高精度版）API 的样例，帮助您零编码快速体验并熟悉文字识别服务。视频教程请参见如何用可视化工具调用API服务（视频版）。

1. 下载并安装接口调用工具

1.1 下载接口调用工具 — Postman

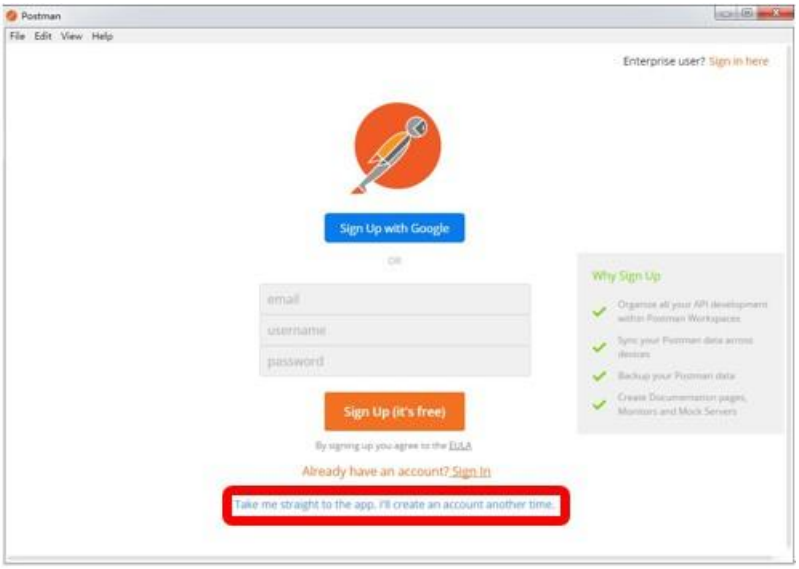
下载地址如下：

- **Mac 下载地址**，点击前往>>
- **Windows 下载地址**，点击前往>>

1.2 Postman 安装教程

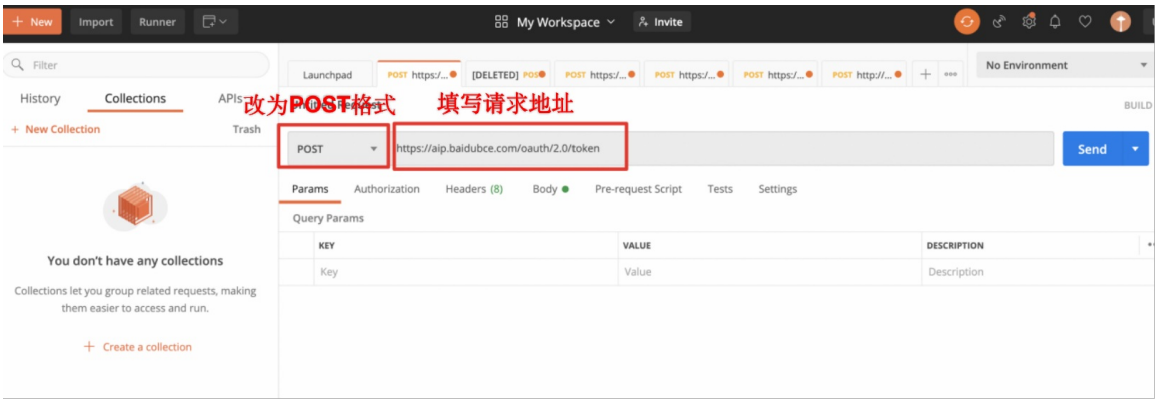
（1）双击安装包。

（2）初次登录无账号，可点击图示最下方蓝色字体部分，直接进入 postman 主界面。



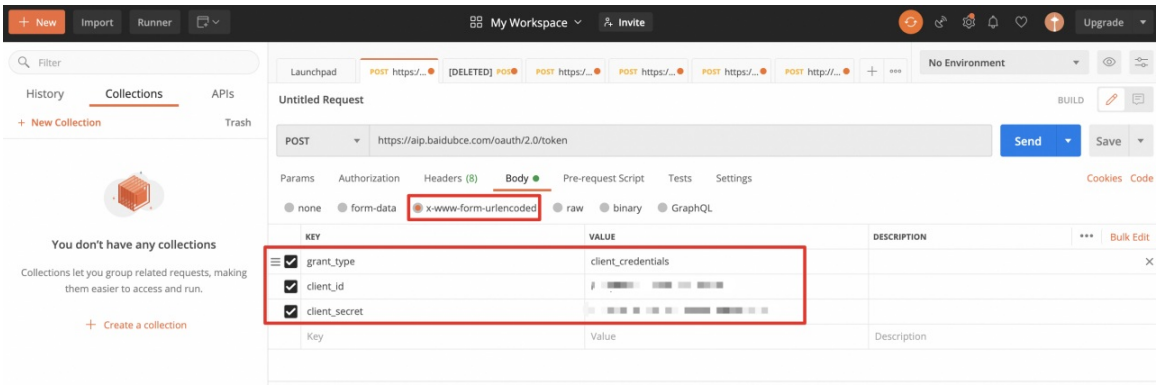
2. 获取 Access Token

将请求格式改为 “POST” 并填写请求地址：`https://aip.baidubce.com/oauth/2.0/token`

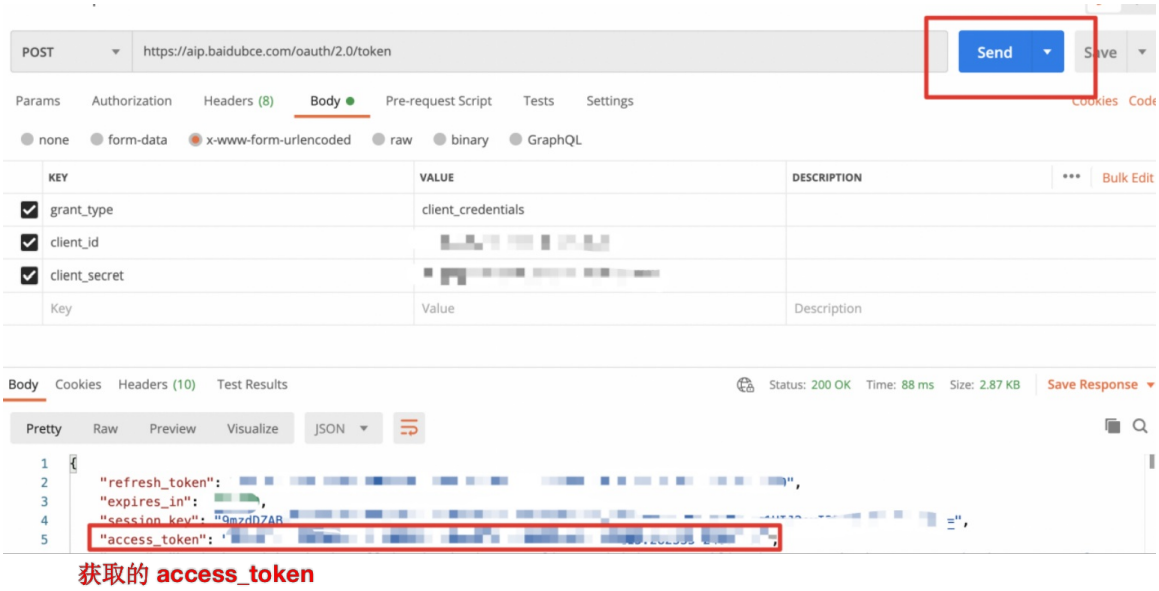


点击 Body，选择“x-www-form-urlencoded”，在 key 和 value 中分别输入以下3个请求参数。

- **grant_type**： 必须参数，固定为 `client_credentials`；
- **client_id**： 必须参数，应用的 `API Key`；
- **client_secret**： 必须参数，应用的 `Secret Key`；



点击右上角蓝色“send”，即可在下方返回值区域中获取 access_token。



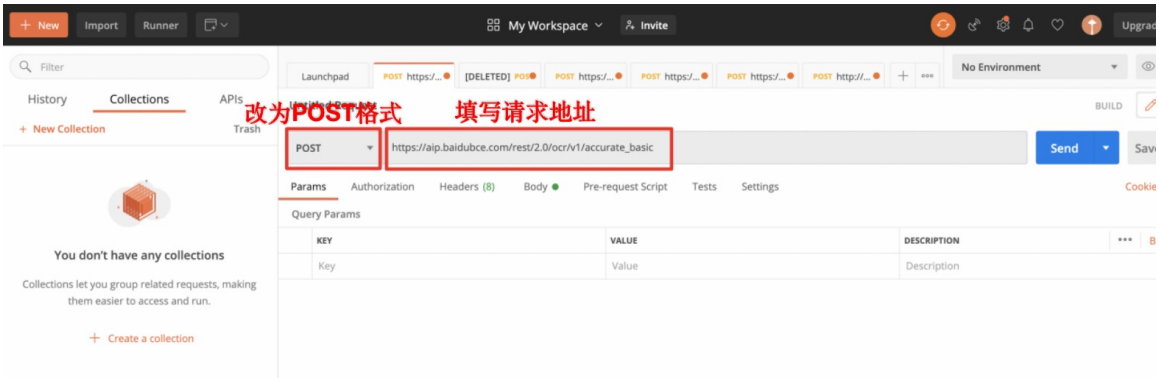
获取的 access_token

3. 进行接口调用

3.1 接口调用

具体操作如下：

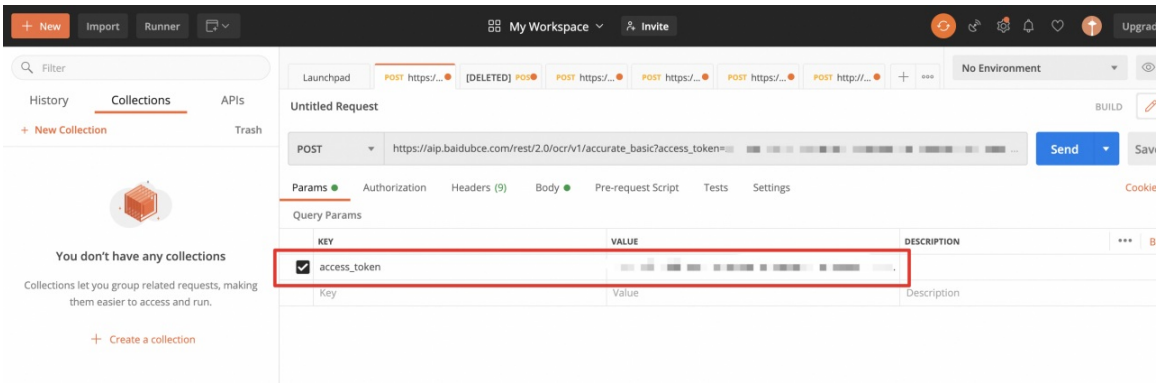
（1） 将请求格式改为 “POST” 并填写请求地址（以通用文字识别高精度版为例）：https://aip.baidubce.com/rest/2.0/ocr/v1/accurate_basic



（2） 点击 Params，在 key 和 value 中分别输入1个请求参数

key 栏输入：access_token

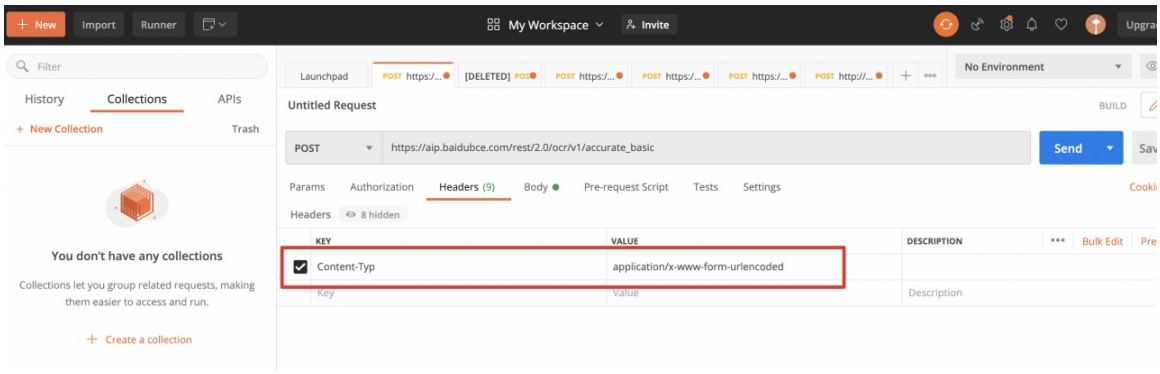
value 栏输入：上一步中获取的的 access_token



（3） 修改请求头，点击 Headers，在 key 和 value 中分别输入1个请求参数

key 栏输入：Content-Type

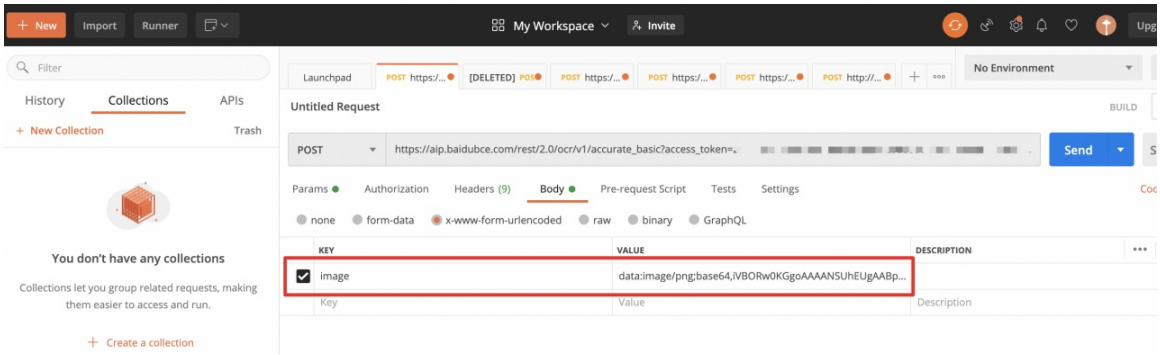
value 栏输入：application/x-www-form-urlencoded



（4） 点击 Body，选择“x-www-form-urlencoded”，在 key 和 value 中分别输入1个请求参数

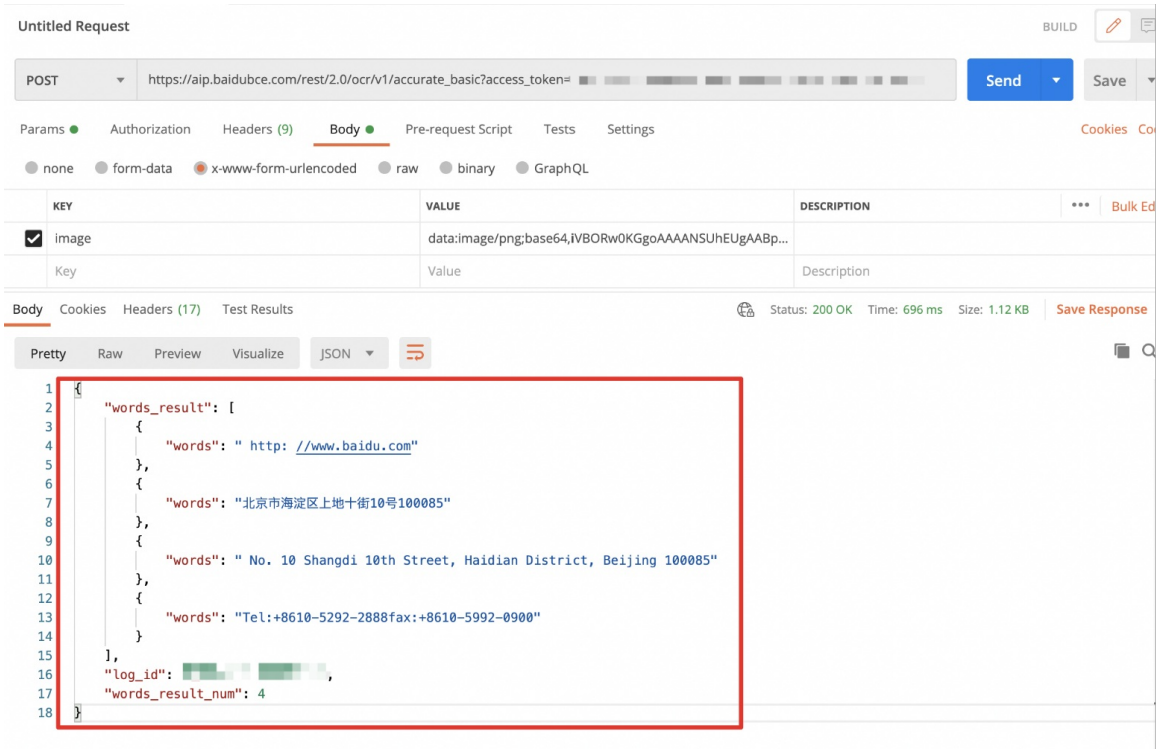
key 栏输入：image

value 栏输入：任意文字图片的base64编码后的结果



3.2 开始请求

点击右上角蓝色“send”，即可在下方返回值区域中获取识别结果。



如何用代码调用OCR服务

如何使用代码调用文字识别服务 API

本文提供通过代码快速调用 OCR 通用文字识别（高精度版）API 的样例，帮助您通过简单的代码编写快速熟悉并使用文字识别

服务。视频教程请参见如何用代码调用API服务（视频版）。

1. 准备开发环境

我们选择用 Python 来快速搭建一个原型，关于如何安装 Python。可以参考下表列出的不同操作系统的安装方法进行安装。

Python的官方下载地址：下载Python

系统是Windows	系统是Linux	系统是macOS
✓ 在Python官网下载Python ✓ 解压并双击exe文件安装	✓ 执行`python --version`看是否输出python版本； ✓ 若回显版本则系统已预装Python； ✓ 若无版本则执行`sudo apt-get update & sudo apt-get install python3.6`	✓ macOS一般自带了Python，无需自行安装

****Windows 快速测试包****

Windows 平台的用户如果对上述的 Python 安装感到困难，您可以下载我们的一键测试包，下载地址：Windows测试包。

解压 zip 文件后，双击 run.bat 即可测试。

2. 编写代码

新建一个 **main.py**

粘贴以下内容，不要忘记替换您的 **API_KEY** 以及 **SECRET_KEY**：

```
 #### coding=utf-8

import sys
import json
import base64


#### 保证兼容python2以及python3
IS_PY3 = sys.version_info.major == 3
if IS_PY3:
    from urllib.request import urlopen
    from urllib.request import Request
    from urllib.error import URLError
    from urllib.parse import urlencode
    from urllib.parse import quote_plus
else:
    import urllib2
    from urllib import quote_plus
    from urllib2 import urlopen
    from urllib2 import Request
    from urllib2 import URLError
    from urllib import urlencode

#### 防止https证书校验不正确
import ssl
ssl._create_default_https_context = ssl._create_unverified_context

API_KEY = 'GmhC18eVP1Fo1ECX911dtOzw'

SECRET_KEY = 'PQ2ukO4Aec2PTsgQU9UkiEKYciavlZk8'


OCR_URL = "https://aip.baidubce.com/rest/2.0/ocr/v1/accurate_basic"


""" TOKEN start """
TOKEN_URL = 'https://aip.baidubce.com/oauth/2.0/token'
```

```
"""
    获取token
"""
def fetch_token():
    params = {'grant_type': 'client_credentials',
              'client_id': API_KEY,
              'client_secret': SECRET_KEY}
    post_data = urlencode(params)
    if (IS_PY3):
        post_data = post_data.encode('utf-8')
    req = Request(TOKEN_URL, post_data)
    try:
        f = urlopen(req, timeout=5)
        result_str = f.read()
    except URLError as err:
        print(err)
    if (IS_PY3):
        result_str = result_str.decode()


    result = json.loads(result_str)

    if ('access_token' in result.keys() and 'scope' in result.keys()):
        if not 'brain_all_scope' in result['scope'].split(' '):
            print ('please ensure has check the  ability')
            exit()
        return result['access_token']
    else:
        print ('please overwrite the correct API_KEY and SECRET_KEY')
        exit()

"""
    读取文件
"""
def read_file(image_path):
    f = None
    try:
        f = open(image_path, 'rb')
        return f.read()
    except:
        print('read image file fail')
        return None
    finally:
        if f:
            f.close()


"""
    调用远程服务
"""
def request(url, data):
    req = Request(url, data.encode('utf-8'))
    has_error = False
    try:
        f = urlopen(req)
        result_str = f.read()
        if (IS_PY3):
            result_str = result_str.decode()
        return result_str
    except  URLError as err:
```

```
        print(err)

if __name__ == '__main__':

    # 获取access token
    token = fetch_token()

    # 拼接通用文字识别高精度url
    image_url = OCR_URL + "?access_token=" + token

    text = ""

    # 读取测试图片
    file_content = read_file('./text.jpg')

    # 调用文字识别服务
    result = request(image_url, urlencode({'image': base64.b64encode(file_content)}))

    # 解析返回结果
    result_json = json.loads(result)
    for words_result in result_json["words_result"]:
        text = text + words_result["words"]

    # 打印文字
    print(text)
```

3. 运行代码

在命令行中运行`python main.py`

4. 获取识别结果

代码正确运行后，命令行界面上会显示出如下运行结果：



返回的数据包含了图片中所有文字，详细的接口返回可以查看文档 文字识别API文档 。

购买指南

计费概述

计费简介

文字识别各服务均提供一定额度的 **免费测试资源** 供测试使用，免费测试资源使用完毕可选择按照 **预付费** 和 **后付费** 方式进行计费，两种计费方式均可在 控制台 直接开通或购买。当发生接口调用时，系统会按照如下顺序依次抵扣：**免费测试资源 > 专项资源包 > 共享资源包 > 按量后付费**。

免费测试资源

免费测试资源 是指免费调用次数，供测试使用。在您完成实名认证后，首次进入控制台即自动领取。免费测试资源使用完毕后可开通付费按次计费。各服务可领取的免费测试资源及赠送方式可查看 免费测试资源。

预付费资源包

预付费资源包 是指您根据业务量级一次性付费购买对应规格的资源包，购买后一年内有效，**超出有效期未抵扣额度自动失效，无法继续使用。** 预付费资源包分为两类：

- **专项资源包**：次数包，支持指定的单个接口使用专项资源包额度，详细价格见 产品价格；
- **共享资源包**：点数包，支持多个接口共用资源包额度，不同接口单次成功调用所抵扣的点数不同，详细价格见 产品价格。

按量后付费

按量后付费 是指对实际产生的计费调用量 **按自然月进行阶梯统计，到达相应阶梯的计费调用量按照所在阶梯单价进行计费** ，系统每小时从您的百度云账户中扣除对应的消费额。各服务详细价格见 产品价格，您也可使用 文字识别价格计算器，预设月调用量查看相应的预估价格。其余未计费服务免费调用量如果无法满足需求，可 申请调额。

QPS叠加包

以上两种计费方式开通后均可保证单账户 **10 QPS** 并发需求，如果并发量无法满足您业务需求，您可购买 **QPS叠加包** 用于提升并发量，以更高的并发进行调用，详细价格见 产品价格。请注意，各账户针对单个接口可购买的最大 QPS 额度为 **100 QPS**。

如您有很高的调用量 / 并发量需求，或需其他特殊的付费方式欢迎进行 商务咨询。

免费/付费配置

免费使用和开通付费所使用服务相同 ，识别效果与性能相同，无需更改调用方式。**但开通付费后的服务配置有较大提升** ，具体对比如下：

状态	免费额度	超过免费配额	QPS限制
免费状态	拥有	不响应请求	不保证并发
付费状态	拥有	响应请求	保证10次并发

计费与付费

- 对于免费测试资源未使用完毕的接口服务，开通付费后，先使用免费测试资源中的调用额度，使用完毕后优先抵扣预付费资源包额度，抵扣完毕才根据调用量进行阶梯计费；
- 采用按量后付费的计费方式，每小时对您的百度云账户进行扣费；使用预付费资源包计费方式，同样每小时进行抵扣，从已购买的资源包中按照**专项资源包优先于共享资源包**，按照购买**时间顺序由早至晚**，按照**规格由小至大**依次扣除相应额度；
- 用户开通付费前需保证账户已完成个人/企业实名认证，且账户未欠费。

欠费处理

余额不足提醒

根据您历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心进行充值。

欠费处理

- 按照北京时间整点检查您的账户余额是否足以支付最近1小时的账单费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，停止调用，并发送欠费通知；
- 欠费后您开通付费的产品将进入欠费状态，各服务只能使用每日的免费配额及已购买的资源包额度，超出部分系统将不再响应，且不再保证并发数量。

免费测试资源

开通文字识别服务并完成实名认证后，即可自动获取各接口的免费测试资源，免费测试资源使用完毕后，可开通付费针对超出部分进行计费调用。

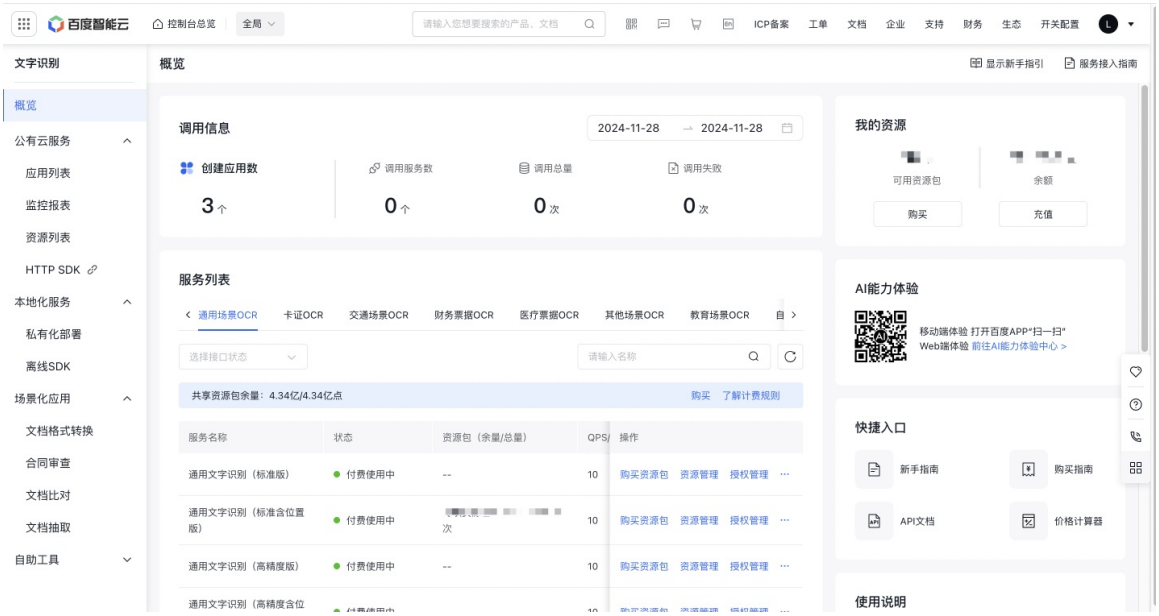
注意：

- **成功调用与失败调用均消耗免费测试资源** ，免费测试资源使用完毕后，可开通付费或申请免费调额；

- 开通多项文字识别服务的用户，每项服务都将自动获取对应的免费测试资源，但请注意，按月赠送的免费测试资源如果未使用完毕，将不会流转到下一自然月；
- 如有作弊或违禁使用情况，百度有权停止使用或发放免费测试资源。

获取方式

1. 登录并进入文字识别控制台， 如果您的账号已经完成了实名认证，系统会自动为您的账号发放免费测试资源。



2. 领取到的免费资源可以在资源列表里查看，测试资源会在您进入控制台后约10分钟内发放，**若领取接口长时间未在「资源列表」上生效显示，请提交工单咨询。**

各接口免费测试资源

类别	服务名	免费测试资源	超出免费测试资源
	通用文字识别（标准版）	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量阶梯后付费或购买预付费资源包，按次计费
	通用文字识别（标准含位置版）	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量阶梯后付费或购买预付费资源包，按次计费
	通用文字识别（高精度版）	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	通用文字识别（高精度	个人认证：500次/月；	可开通按量后付费或购买预付费资源包，按次计费

	含位置版）	企业认证：1,000次/月	
	网络图片文字识别	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	网络图片文字识别（含位置版）	个人认证：500次； 企业认证：1,000次	可开通按量后付费或购买预付费资源包，按次计费
通用场景文字识别	办公文档识别	个人认证：500次； 企业认证：1,000次	可开通按量后付费或购买预付费资源包，按次计费
	表格文字识别V2	个人认证：500次/月； 企业认证：1,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	数字识别	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	手写文字识别	个人认证：500次/月； 企业认证：1,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	二维码识别	个人认证：500次； 企业认证：1,000次	可开通按量后付费或购买预付费资源包，按次计费
	印章识别	个人认证：500次； 企业认证：1,000次	可开通按量后付费或购买预付费资源包，按次计费
	智能结构化	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费或购买预付费资源包，按次计费
	文档解析	个人/企业认证：200页	可开通按量阶梯后付费或购买预付费资源包，按页计费
		个人认证：1,000	

		次/月； 企业认证：2,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	身份证识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	身份证识别（加密版）	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	银行卡识别	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	营业执照识别	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	护照识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	护照识别（港澳台地区及境外护照）	个人认证：200次； 企业认证：500次	公测阶段，暂不支持开通付费，可申请调额
	社保卡识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费
卡证文字识别	港澳台证件识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	结婚证识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	离婚证识别	个人认证：200次； 企业认证：500次	公测阶段，暂不支持开通付费，可申请调额
	户口本识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费

	出生医学证明识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	房产证识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	开户许可证识别	个人认证：200次； 企业认证：500次	公测阶段，暂不支持开通付费，可申请调额
	食品经营许可证识别	个人认证：200次； 企业认证：500次	公测阶段，暂不支持开通付费，可申请调额
	食品生产许可识别	个人认证：200次； 企业认证：500次	公测阶段，暂不支持开通付费，可申请调额
	外国人永久居住证识别	个人认证：200次； 企业认证：500次	邀测阶段，暂不支持开通付费，可申请调额
	企业工商信息查询（标准版）	企业认证： 50次	可开通按量后付费或购买预付费资源包，按次计费
	企业工商信息查询（高级版）	企业认证： 50次	可开通按量后付费或购买预付费资源包，按次计费
	企业二要素核验	企业认证： 50次	可开通按量后付费或购买预付费资源包，按次计费
	企业三要素核验	企业认证： 50次	可开通按量后付费或购买预付费资源包，按次计费
	企业四要素核验	企业认证： 50次	可开通按量后付费或购买预付费资源包，按次计费
	行驶证识别	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	驾驶证识别	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	车辆证照混贴识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	车牌识别	个人认证：1,000次/月； 企业认证：2,000	可开通按量后付费或购买预付费资源包，按次计费

		次/月	
交通场景文字识别	VIN码识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	机动车销售发票识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	二手车销售发票识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	车辆合格证识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	机动车登记证书识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	磅单识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	快递面单识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	道路运输证识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	智能财务票据识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	增值税发票识别	个人认证：1,000次/月； 企业认证：2,000次/月	可开通按量后付费或购买预付费资源包，按次计费
	增值税发票验真	个人认证：20次； 企业认证：50次	可开通按量后付费或购买预付费资源包，按次计费
	定额发票识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费

财务票据文字识别	通用机打发票识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	火车票识别	个人认证：200次/月； 企业认证：500次/月	可开通按量后付费或购买预付费资源包，按次计费
	出租车票识别	个人认证：200次/月； 企业认证：500次/月	可开通按量后付费或购买预付费资源包，按次计费
	飞机行程单识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	网约车行程单识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	汽车票识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	过路过桥费发票识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	船票识别	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
	购物小票识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费
	银行回单识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费
	医疗发票识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费

		企业认证：500次	
	医疗费用明细识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费
	医疗检验报告单识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费
	医疗诊断报告单识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费
	医疗费用结算单识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费
医疗票据文字识别	病案首页识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费
	出院小结识别	个人认证：200次； 企业认证：500次	可开通按量阶梯后付费，按次计费
	入院小结识别	个人/企业认证：200次/天	邀测阶段，暂不支持开通付费，可申请调额
	门诊病历识别	个人/企业认证：200次/天	邀测阶段，暂不支持开通付费，可申请调额
	诊断证明识别	个人/企业认证：200次/天	邀测阶段，暂不支持开通付费，可申请调额
	处方笺识别	个人/企业认证：200次/天	邀测阶段，暂不支持开通付费，可申请调额
	手术记录识别	个人/企业认证：200次/天	邀测阶段，暂不支持开通付费，可申请调额
	医疗票据类别检测	个人/企业认证：500次	邀测阶段，暂不支持开通付费，可申请调额
教育场景文字识别	试卷分析与识别	个人认证：500次； 企业认证：1,000次	可开通按量后付费或购买预付费资源包，按次计费
	词典笔文字识别	个人认证：500次； 企业认证：1,000次	公测阶段，暂不支持开通付费，可申请调额
	试卷切题识别	个人认证：200次； 企业认证：500次	邀测阶段，暂不支持开通付费，可申请调额
		个人认证：500次；	

其他文字识别	门脸文字识别	企业认证：1,000次	可开通按量后付费或购买预付费资源包，按次计费
	仪器仪表盘读数识别	个人认证：500次； 企业认证：1,000次	可开通按量后付费或购买预付费资源包，按次计费
文档图像处理	文档矫正增强	个人认证：50次； 企业认证：100次	可开通按量后付费或购买预付费资源包，按次计费
	文档去手写	个人认证：50次； 企业认证：100次	可开通按量后付费或购买预付费资源包，按次计费
	图片去摩尔纹	个人认证：50次； 企业认证：100次	可开通按量后付费或购买预付费资源包，按次计费
	文档图像去底纹	个人认证：50次； 企业认证：100次	可开通按量后付费或购买预付费资源包，按次计费
	文件检测分类	个人认证：200次； 企业认证：500次	可开通按量后付费或购买预付费资源包，按次计费
智能文档分析平台	文档格式转换（原图文转换器）	个人/企业认证：200页	可开通按量后付费或购买预付费资源包，按页计费；支持在线工具和接口调用两种方式
	合同审查	个人/企业认证：200页	可开通按量后付费或购买预付费资源包，按页计费；支持在线工具和接口调用两种方式
	文档比对	个人/企业认证：200页	可开通按量后付费或购买预付费资源包，按页计费；支持在线工具和接口调用两种方式
	文档抽取	个人/企业认证：200页	可开通按量后付费或购买预付费资源包，按页计费；支持在线工具和接口调用两种方式
	文档解析	个人/企业认证：200页	可开通按量后付费或购买预付费资源包，按页计费；支持接口调用方式
iOCR自定义模板文字识别	iOCR通用版	个人认证：500次； 企业认证：1,000次	可开通按量后付费或购买预付费资源包，按次计费
	iOCR财会版	个人认证：500次； 企业认证：1,000次	可开通按量阶梯后付费，按次计费
EasyDL OCR自训练平台		个人/企业认证：500次	公测阶段，暂不支持开通付费，可申请调额

产品价格

共享资源包

共享资源包

商品说明 共享资源包，是指您根据业务量级一次性付费购买的**点数包**，**可用于文字识别 OCR 下所有付费接口（核验类接口、智能文档平台除外）**。若您存在文字识别 OCR 多接口同时使用的需求，建议直接 购买共享资源包，可极大提升业务灵活性，后续若需使用新增付费接口，无需再购买专项资源包。

- **共享资源包为点数抵扣制，不同接口产生调用时，单次成功调用抵扣的点数不同**，详见下文 抵扣规则；
- **支持自主配置共享范围，共享范围内的接口才可抵扣共享资源包额度**。您可在购买时通过勾选接口进行配置，也可在购买后前往 资源包管理 页面进行配置。购买共享资源包后，**新开通按量后付费的接口，也将自动纳入共享范围**，无需手动配置；
- **共享范围内的接口，默认开通按量后付费**，若终止某接口的按量后付费，则该接口无法再使用共享资源包额度。

价格与购买

规格（点）	价格（元）
10万	330
50万	1540
100万	2570
500万	12000
1000万	18000
5000万	60000
1亿	100000
更高规格	商务咨询

> **说明：**
>
> - 共享资源包购买后一年内有效，有效期内产生计费调用量，按照如下顺序依次抵扣：**专项资源包 > 共享资源包 > 按量后付费**，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 共享资源包购买后7天内若未产生调用可前往 退订管理 页面进行自助退订，详细退订规则见 退订说明，**购买超过 7 天或已使用的资源包不支持退订**

抵扣规则 可抵扣共享资源包额度的接口分为四档，单次成功调用抵扣 5 个点、 10 个点、25 个点、200 个点，具体抵扣规则见下表。

- **计算示例**：当月调用通用文字识别（高精度版）的总次数中，需计费的调用量为 10 万次，当月调用身份证识别的总次数中，需计费的调用量为 5 万次，且仅购买了共享资源包，则需抵扣共享资源包点数为 **125 万点** = 10 万次 x 10 点/次 + 5 万次 x 5 点/次

类别	服务名	单次成功调用所抵扣的点数
	通用文字识别（标准版）	5
	通用文字识别（标准含位置版）	5
	通用文字识别（高精度版）	10
	通用文字识别（高精度含位置版）	10
	网络图片文字识别	5

	网络图片文字识别	5
	网络图片文字识别（含位置版）	5
通用场景文字识别	办公文档识别	25
	表格文字识别V2	25
	智能结构化	25
	数字识别	5
	手写文字识别	5
	二维码识别	5
	印章识别	5
卡证文字识别	身份证识别	5
	身份证混贴识别	5
	身份证识别（加密版）	10
	银行卡识别	5
	营业执照识别	10
	护照识别	10
	社保卡识别	5
	港澳台证件识别	10
	结婚证识别	10
	户口本识别	10
	出生医学证明识别	10
	房产证识别	10
交通场景文字识别	行驶证识别	10
	驾驶证识别	10
	车辆证照混贴识别	10
	车牌识别	10
	VIN码识别	10
	机动车销售发票识别	10
	二手车销售发票识别	10
	车辆合格证识别	10
	机动车登记证书识别	10
	磅单识别	10
	快递面单识别	10
	道路运输证识别	10
财务票据文字识别	智能财务票据识别	25
	增值税发票识别	10
	定额发票识别	10
	通用机打发票识别	10
	火车票识别	10
	出租车票识别	10
	飞机行程单识别	10

	网约车行程单识别	10
	汽车票识别	10
	过路过桥费发票识别	10
	船票识别	10
	购物小票识别	10
	银行回单识别	10
医疗票据文字识别	医疗发票识别	200
	医疗费用明细识别	200
	医疗检验报告单识别	200
	医疗诊断报告单识别	200
	医疗费用结算单识别	200
	病案首页识别	200
	出院小结识别	200
教育场景文字识别	试卷分析与识别	25
其他文字识别	仪器仪表盘读数识别	5
文档图像处理	文档矫正增强	25
	文档去手写	25
	文件检测分类	10

> **说明：**
>
> - 对于同一个接口，使用共享资源包或专项资源包，单次调用成本略有差别。您可综合业务需求、价格等进行判断，选择适合的资源包类型

通用场景文字识别

通用文字识别（标准版）

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可点击 购买次数包 或开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包 **预付费次数包**

规格（次）	**价格（元）**
1万	50
5万	248
10万	470
20万	860
50万	1850
100万	3250
500万	12000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超

> 出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
>
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 0<调用次数<=5 | 0.0050 | | 5<调用次数<=10 | 0.0045 | | 10<调用次数<=20 | 0.0040 | | 20<调用次数<=50 | 0.0035 | | 50<调用次数<=100 | 0.0030 | | 100<调用次数 | 0.0025 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	10 元/天/QPS
按月购买	180 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

通用文字识别（标准含位置版）

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可点击 购买次数包 或开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包 **预付费次数包**

规格（次）	价格（元）
1万	100
5万	490
10万	880
20万	1500
50万	3100
100万	5400
500万	23000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 0<调用次数<=5 | 0.0100 | | 5<调用次数<=10 | 0.0080 | | 10<调用次数<=20 | 0.0065 | | 20<调用次数<=50 | 0.0055 | | 50<调用次数<=100 | 0.0050 | | 100<调用次数 | 0.0047 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	10 元/天/QPS
按月购买	180 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

通用文字识别（高精度版）

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	价格（元）
1万	280
5万	1350
10万	2300
20万	3600
50万	7000
100万	11000
500万	38000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.030
5<调用次数<=10	0.024
10<调用次数<=20	0.019
20<调用次数<=50	0.015
50<调用次数<=100	0.012
100<调用次数	0.010

> **说明：**“调用次数”只包括成功调用，调用失败不计费也不抵扣次数包额度

QPS叠加包

购买方式	价格
按天购买	10 元/天/QPS
按月购买	180 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

通用文字识别（高精度含位置版）

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次/月，企业认证 1,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	价格（元）
1万	380
5万	1800
10万	3200
20万	5400
50万	11000
100万	18000
500万	72000

> **说明：**
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.040
5<调用次数<=10	0.034
10<调用次数<=20	0.029
20<调用次数<=50	0.025
50<调用次数<=100	0.022
100<调用次数	0.020

> **说明：**“调用次数”只包括成功调用，调用失败不计费也不抵扣次数包额度

QPS叠加包

购买方式	价格
按天购买	10 元/天/QPS
按月购买	180 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

网络图片文字识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	价格（元）
1万	100
5万	490
10万	880
20万	1500
50万	3100
100万	5400
500万	23000

> **说明：**
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 0<调用次数<=5 | 0.0100 | | 5<调用次数<=10 | 0.0080 | | 10<调用次数<=20 | 0.0065 | | 20<调用次数<=50 | 0.0055 | | 50<调用次数<=100 | 0.0050 | | 100<调用次数 | 0.0047 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	15 元/天/QPS
按月购买	270 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

网络图片文字识别（含位置版）

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次，企业认证 1,000 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	价格（元）
1万	280
5万	1350
10万	2300
20万	3600
50万	7000
100万	11000
500万	38000

> **说明：**
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	价格（元/次）
不限量	0.03

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	15 元/天/QPS
按月购买	270 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包

- 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

表格文字识别V2

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次/月，企业认证 1,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	价格（元）
1000	120
1万	1100
5万	5000
10万	9000
20万	16000
50万	35000
100万	60000
500万	250000

说明：

- 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
- 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数	0.12

说明：“调用次数”只包括成功调用，调用失败不计费也不抵扣次数包额度

印章识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次，企业认证 1,000 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，开通按量后付费后，可购买QPS叠加包。

预付费次数包

规格（次）	价格（元）
1万	500
5万	2000
10万	3000
20万	4000
50万	7500
100万	12000
500万	40000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	价格（元/次）
不限量	0.06

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	20 元/天/QPS
按月购买	360 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

数字识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	价格（元）
1万	38
5万	170
10万	300
20万	520
50万	1000
100万	1800
500万	8000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=2	0.0040
2<调用次数<=5	0.0034
5<调用次数<=10	0.0029
10<调用次数<=20	0.0025
20<调用次数<=30	0.0022
30<调用次数	0.0020

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	10 元/天/QPS
按月购买	180 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

手写文字识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次/月，企业认证 1,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	手写文字识别（元）
1万	98
5万	420
10万	740
20万	1250
50万	2400
100万	4200
500万	19000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	手写文字识别（元/次）
0<调用次数<=2	0.0100
2<调用次数<=5	0.0080
5<调用次数<=10	0.0065
10<调用次数<=20	0.0055
20<调用次数<=30	0.0050
30<调用次数	0.0045

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	20 元/天/QPS
按月购买	360 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

二维码识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次，企业认证 1,000 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	价格（元）
1千	30
1万	280
5万	1,300
10万	2,200
20万	3,600
50万	6,500
100万	11,000
500万	34,000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
调用次数 > 0	0.03

> **说明：**“调用次数”只包括成功调用，调用失败不计费

办公文档识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次，企业认证 1,000 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	办公文档识别（元）
1万	1500
5万	6500
10万	12000
20万	22000
50万	50000
100万	80000
500万	300000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	办公文档识别（元/次）
不限量	0.16

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	50 元/天/QPS
按月购买	900 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

智能结构化

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	智能结构化（元）
1万	1500
5万	6500
10万	12000
20万	22000
50万	50000
100万	80000
500万	300000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	智能结构化（元/次）
不限量	0.16

> **说明：**“调用次数”只包括成功调用，调用失败不计费

文档解析

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 页，企业认证 200 页。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买资源包或 开通按量后付费

预付费资源包 | 规格（页） | 价格（元） | | ---------- | ---------- | | 1000 | 180 | | 5000 | 850 | | 1万 | 1600 | | 5万 | 7500 | | 10万 | 14000 | | 20万 | 26000 | | 50万 | 55000 | | 100万 | 90000 | | 500万 | 350000 |

> **说明：**
>
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 文档解析（元/页） | | ---------- | ---------- | | 不限量 | 0.18 | **说明：** “调用次数”只包括成功调用，调用失败不计费

卡证文字识别

身份证识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	价格（元）
1万	200
5万	950
10万	1700
20万	3000
50万	6000
100万	10000
500万	40000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.0200
5<调用次数<=10	0.0160
10<调用次数<=20	0.0130
20<调用次数<=50	0.0110
50<调用次数<=100	0.0100
100<调用次数	0.0090

> **说明：**需要计费的错误码：216633 - 未识别到身份证

QPS叠加包

购买方式	价格
按天购买	15 元/天/QPS
按月购买	270 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

身份证混贴识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次**。

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	价格（元）
1万	200
5万	950
10万	1700
20万	3000
50万	6000
100万	10000
500万	40000

> **说明：**
>
> - **单次成功调用中，如果一张图片上包含了多张身份证图片，按照实际识别出身份证图片数量计费（如一张图片上有身份证头像面、国徽面各一张，计费数量为 2 次）**
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.0200
5<调用次数<=10	0.0160
10<调用次数<=20	0.0130
20<调用次数<=50	0.0110
50<调用次数<=100	0.0100
100<调用次数	0.0090

> **说明：**
>
> - **单次成功调用中，如果一张图片上包含了多张身份证图片，按照实际识别出身份证图片数量计费（如一张图片上有身份证头像面、国徽面各一张，计费数量为 2 次）**
> - “调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	25 元/天/QPS
按月购买	420 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

身份证识别（金融加密版）

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证200 次，企业认证500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	价格（元）
1万	300
5万	1400
10万	2400
20万	4000
50万	7800
100万	12000
500万	43000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超

出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买

- 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.0300

说明：需要计费的错误码：216633 - 未识别到身份证

QPS叠加包

购买方式	价格
按天购买	15 元/天/QPS
按月购买	270 元/月/QPS

- 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
- 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

银行卡识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1万	200
5万	950
10万	1700
20万	3000
50万	6000
100万	10000
500万	40000

说明：

- 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
- 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.0200
5<调用次数<=10	0.0160
10<调用次数<=20	0.0130
20<调用次数<=50	0.0110
50<调用次数<=100	0.0100
100<调用次数	0.0090

> **说明：**需要计费的错误码：216631 - 未识别到银行卡

QPS叠加包

购买方式	价格
按天购买	20 元/天/QPS
按月购买	360 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

营业执照识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1万	280
5万	1200
10万	1900
20万	3000
50万	5500
100万	9000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=2	0.030
2<调用次数<=5	0.024
5<调用次数<=10	0.019
10<调用次数<=20	0.015
20<调用次数<=30	0.012
30<调用次数	0.010

> **说明：**“调用次数”只包括成功调用，调用失败不计费

护照识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	价格（元）
1千	60
1万	550
5万	2,500
10万	4,600
20万	5,800
50万	8,000
100万	11,000
500万	34,000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.06

> **说明：**“调用次数”只包括成功调用，调用失败不计费

户口本识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	价格（元）
1千	88
1万	720
5万	3,000
10万	4,800
20万	7,200
50万	12,000
100万	18,000
500万	48,000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.12

> **说明：**“调用次数”只包括成功调用，调用失败不计费

结婚证识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	价格（元）
1千	88
1万	720
5万	3,000
10万	4,800
20万	7,200
50万	12,000
100万	18,000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.12

> **说明：**“调用次数”只包括成功调用，调用失败不计费

港澳台证件识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	价格（元）
1千	88
1万	720
5万	3,000
10万	4,800
20万	7,200
50万	12,000
100万	18,000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.12

> **说明：**“调用次数”只包括成功调用，调用失败不计费

出生证明识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	价格（元）
1千	88
1万	720
5万	3,000
10万	4,800
20万	7,200
50万	12,000
100万	18,000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.12

> **说明：**“调用次数”只包括成功调用，调用失败不计费

社保卡识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1000	20
1万	200
5万	950
10万	1700
20万	3000
50万	6000
100万	10000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.0200

> **说明：**“调用次数”只包括成功调用，调用失败不计费

房产证识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	价格（元）
1千	88
1万	720
5万	3,000
10万	4,800
20万	7,200
50万	12,000
100万	18,000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.12

> **说明：**“调用次数”只包括成功调用，调用失败不计费

企业工商信息查询（标准版）

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**企业认证 50 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
100	30
1000	300
1万	2800
5万	13000
10万	23000
20万	40000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.3
5<调用次数<=10	0.28
10<调用次数	0.25

> **说明：**“调用次数”只包括成功调用，调用失败不计费

企业工商信息查询（高级版）

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**企业认证 50 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
100	50
1000	500
1万	4800
5万	21000
10万	36000
20万	60000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.5
5<调用次数<=10	0.45
10<调用次数	0.4

> **说明：**“调用次数”只包括成功调用，调用失败不计费

企业二要素核验

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**企业认证 50 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
100	25
1000	250
1万	2300
5万	10000
10万	18000
20万	30000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.25
5<调用次数<=10	0.22
10<调用次数	0.2

> **说明：**“调用次数”只包括成功调用，调用失败不计费

企业三要素核验

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**企业认证 50 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
100	30
1000	300
1万	2800
5万	13000
10万	23000
20万	40000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.3
5<调用次数<=10	0.28
10<调用次数	0.25

> **说明：**“调用次数”只包括成功调用，调用失败不计费

企业四要素核验

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**企业认证 50 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
100	100
1000	1000
1万	9500
5万	45000
10万	85000
20万	160000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	1
5<调用次数<=10	0.95
10<调用次数	0.9

> **说明：**“调用次数”只包括成功调用，调用失败不计费

交通场景文字识别

行驶证识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1万	400
5万	1800
10万	3200
20万	5800
50万	12500
100万	18000
500万	50000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 0<调用次数<=2 | 0.040 | | 2<调用次数<=5 | 0.035 | | 5<调用次数<=10 | 0.031 | | 10<调用次数<=20 | 0.028 | | 20<调用次数<=30 | 0.026 | | 30<调用次数 | 0.025 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

驾驶证识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或 开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1万	400
5万	1800
10万	3200
20万	5800
50万	12500
100万	18000
500万	50000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 0<调用次数<=2 | 0.040 | | 2<调用次数<=5 | 0.035 | | 5<调用次数<=10 | 0.031 | | 10<调用次数<=20 | 0.028 | | 20<调用次数<=30 | 0.026 | | 30<调用次数 | 0.025 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	20 元/天/QPS
按月购买	360 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

车辆证照混贴识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或开通按量后付费。

预付费次数包

规格（次）	价格（元）
1000	40
1万	400
5万	1800
10万	3200
20万	5800
50万	12500
100万	18000

> **说明：**
>
> - **单次成功调用中，如果一张图片上包含了多张行驶证/驾驶证图片，按照实际识别出行驶证/驾驶证图片数量计费（如一张图片上有行驶证正页、驾驶证正页各一张，计费数量为 2 次）**
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 不限量 | 0.040 |

> **说明：**
>
> - **单次成功调用中，如果一张图片上包含了多张行驶证/驾驶证图片，按照实际识别出行驶证/驾驶证图片数量计费（如一张图片上有行驶证正页、驾驶证正页各一张，计费数量为 2 次）**
> - “调用次数”只包括成功调用，调用失败不计费

车牌识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或 开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1万	200
5万	950
10万	1800
20万	3200
50万	7000
100万	12000
500万	45000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买

> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 0<调用次数<=5 | 0.02000 | | 5<调用次数<=10 | 0.01740 | | 10<调用次数<=20 | 0.01500 | | 20<调用次数<=50 | 0.01280 | | 50<调用次数<=100 | 0.01080 | | 100<调用次数 | 0.00900 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	20 元/天/QPS
按月购买	360 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

VIN码识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1千	60
1万	550
5万	2500
10万	4600
20万	5800
50万	8000
100万	11000
500万	34000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=2	0.060
2<调用次数<=5	0.048
5<调用次数<=10	0.038
10<调用次数<=20	0.030
20<调用次数<=30	0.024
30<调用次数	0.020

> **说明：**“调用次数”只包括成功调用，调用失败不计费

机动车销售发票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1千	72
1万	560
5万	2400
10万	3500
20万	6000
50万	12000
100万	15000
500万	35000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.08

> **说明：**“调用次数”只包括成功调用，调用失败不计费

二手车销售发票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1千	72
1万	560
5万	2400
10万	3500
20万	6000
50万	12000
100万	15000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.08

> **说明：**“调用次数”只包括成功调用，调用失败不计费

车辆合格证识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1千	72
1万	560
5万	2400
10万	3500
20万	6000
50万	12000
100万	15000
500万	35000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.08

> **说明：**“调用次数”只包括成功调用，调用失败不计费

机动车登记证书识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1千	69
1万	540
5万	2400
10万	3200
20万	5800
50万	11000
100万	15000
500万	50000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.12
5<调用次数<=10	0.09
10<调用次数	0.07

> **说明：**“调用次数”只包括成功调用，调用失败不计费

磅单识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1千	69
1万	540
5万	2400
10万	3200
20万	5800
50万	11000
100万	15000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.08

> **说明：**“调用次数”只包括成功调用，调用失败不计费

快递面单识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1千	69
1万	540
5万	2400
10万	3200
20万	5800
50万	11000
100万	15000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.08

> **说明：**“调用次数”只包括成功调用，调用失败不计费

道路运输证识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1千	69
1万	540
5万	2400
10万	3200
20万	5800
50万	11000
100万	15000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.08

> **说明：**“调用次数”只包括成功调用，调用失败不计费

财务票据文字识别

智能财务票据识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1000	100
1万	900
5万	4000
10万	6000
20万	10000
50万	20000
100万	30000

> **说明：**
>
> - **单次成功调用中，如识别混贴票据，即一张图片上包含了多张不同财务票据图片，按照实际识别出财务票据图片数量计费（如一张图片上有火车票、出租车票各一张，计费数量为 2 次）**
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=2	0.100
2<调用次数<=5	0.095
5<调用次数<=10	0.080
10<调用次数<=20	0.070
20<调用次数<=30	0.060
30<调用次数	0.040

> **说明：**
>
> - **单次成功调用中，如识别混贴票据，即一张图片上包含了多张不同财务票据图片，按照实际识别出财务票据图片数量计费（如一张图片上有火车票、出租车票各一张，计费数量为 2 次）**
> - “调用次数”只包括成功调用，调用失败不计费

增值税发票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 1,000 次/月，企业认证 2,000 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1万	480
5万	2100
10万	3600
20万	6000
50万	12000
100万	20000
500万	85000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=2	0.050
2<调用次数<=5	0.040
5<调用次数<=10	0.032
10<调用次数<=20	0.026
20<调用次数<=30	0.022
30<调用次数	0.020

> **说明：**“调用次数”只包括成功调用，调用失败不计费

增值税发票验真

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证20 次，企业认证 50 次。**。

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1000	240
1万	2300
5万	11000
10万	20000
50万	90000
100万	150000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超

出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买

- 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=5	0.24
5<调用次数<=10	0.22
10<调用次数	0.20

说明：“调用次数”只包括成功调用，调用失败不计费

银行回单识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1000	50
1万	480
5万	2100
10万	3600
20万	6000
50万	12000
100万	20000

说明：

- 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
- 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.050

说明：“调用次数”只包括成功调用，调用失败不计费

购物小票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	价格（元）
1000	50
1万	480
5万	2100
10万	3600
20万	6000
50万	12000
100万	20000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
不限量	0.050

> **说明：**“调用次数”只包括成功调用，调用失败不计费

火车票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次/月，企业认证 500 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或 开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1万	400
5万	1800
10万	3200
20万	5600
50万	11500
100万	18000
500万	50000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 0<调用次数<=2 | 0.040 | | 2<调用次数<=5 | 0.034 | | 5<调用次数<=10 | 0.029 | | 10<调用次数<=20 | 0.025 | | 20<调用次数<=30 | 0.022 | | 30<调用次数 | 0.020 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

出租车票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次/月，企业认证 500 次/月。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1万	400
5万	1800
10万	3200
20万	5600
50万	11500
100万	18000
500万	50000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 0<调用次数<=2 | 0.040 | | 2<调用次数<=5 | 0.034 | | 5<调用次数<=10 | 0.029 | | 10<调用次数<=20 | 0.025 | | 20<调用次数<=30 | 0.022 | | 30<调用次数 | 0.020 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

飞机行程单识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1千	120
1万	1,180
5万	3,600
10万	4,800
20万	7,200
50万	15,000
100万	24,000
500万	90,000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 不限量 | 0.12 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

通用机打发票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1千	48
1万	350
5万	1,400
10万	2,100
20万	3,600
50万	7,500
100万	10,000
500万	32,000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 不限量 | 0.05 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

定额发票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或 开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1000	40
1万	400
5万	1800
10万	3200
20万	5600
50万	11500
100万	18000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 不限量 | 0.040 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

网约车行程单识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或 开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1000	40
1万	400
5万	1800
10万	3200
20万	5600
50万	11500
100万	18000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买

> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 不限量 | 0.040 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

船票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或 开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1000	40
1万	400
5万	1800
10万	3200
20万	5600
50万	11500
100万	18000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 不限量 | 0.040 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

汽车票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或 开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1000	40
1万	400
5万	1800
10万	3200
20万	5600
50万	11500
100万	18000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 不限量 | 0.040 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

过路过桥费发票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买次数包 或 开通按量后付费。 **预付费次数包**

规格（次）	价格（元）
1000	40
1万	400
5万	1800
10万	3200
20万	5600
50万	11500
100万	18000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费 | 月调用量（万次） | 价格（元/次） | | ---------- | ---------- | | 不限量 | 0.040 |

> **说明：**“调用次数”只包括成功调用，调用失败不计费

医疗票据文字识别

医疗发票识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	医疗发票识别（元）
1000	580
1万	5500
5万	26500
10万	50000
20万	90000
50万	200000
100万	350000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	医疗发票识别（元/次）
不限量	0.6

> **说明：**“调用次数”只包括成功调用，调用失败不计费

医疗费用明细识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	医疗费用明细识别（元）
1000	1000
1万	9500
5万	45000
10万	85000
20万	160000
50万	350000
100万	600000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	医疗费用明细识别（元/次）
不限量	1.2

> **说明：**“调用次数”只包括成功调用，调用失败不计费

医疗费用结算单识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	医疗费用结算单识别（元）
1000	580
1万	5500
5万	26500
10万	50000
20万	90000
50万	200000
100万	350000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	医疗费用结算单识别（元/次）
不限量	0.6

> **说明：**“调用次数”只包括成功调用，调用失败不计费

病案首页识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	病案首页识别（元）
1000	580
1万	5500
5万	26500
10万	50000
20万	90000
50万	200000
100万	350000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	病案首页识别（元/次）
不限量	0.6

> **说明：**“调用次数”只包括成功调用，调用失败不计费

出院小结识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	出院小结识别（元）
1000	580
1万	5500
5万	26500
10万	50000
20万	90000
50万	200000
100万	350000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	出院小结识别（元/次）
不限量	0.6

> **说明：**“调用次数”只包括成功调用，调用失败不计费

医疗检验报告单识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	医疗检验报告单识别（元）
1000	600
1万	5500
5万	25000
10万	45000
20万	80000
50万	175000
100万	300000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	医疗检验报告单识别（元/次）
不限量	0.8

> **说明：**“调用次数”只包括成功调用，调用失败不计费

医疗诊断报告单识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费。

预付费次数包

规格（次）	医疗诊断报告单识别（元）
1000	600
1万	5500
5万	25000
10万	45000
20万	80000
50万	175000
100万	250000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	医疗诊断报告单识别（元/次）
不限量	0.8

> **说明：**“调用次数”只包括成功调用，调用失败不计费

教育场景文字识别

试卷分析与识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次，企业认证 1,000 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，可购买 QPS 叠加包

预付费次数包

规格（次）	试卷分析与识别（元）
1万	1500
5万	6500
10万	12000
20万	22000
50万	50000
100万	80000
500万	300000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	试卷分析与识别（元/次）
不限量	0.16

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	50 元/天/QPS
按月购买	900 元/月/QPS

- 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
- 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

其他场景文字识别

仪器仪表盘读数识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次，企业认证 1,000 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，开通按量后付费后，可购买QPS叠加包。

预付费次数包

规格（次）	价格（元）
1万	100
5万	490
10万	880
20万	1500
50万	3100
100万	5400
500万	23000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	价格（元/次）
不限量	0.015

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	10 元/天/QPS
按月购买	180 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

门脸文字识别

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次，企业认证 1,000 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，开通按量后付费后，可购买QPS叠加包。

预付费次数包

规格（次）	价格（元）
1万	95
5万	450
10万	850
20万	1600
50万	3750
100万	7000
500万	30000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量	价格（元/次）
不限量	0.01

> **说明：**“调用次数”只包括成功调用，调用失败不计费

QPS叠加包

购买方式	价格
按天购买	20 元/天/QPS
按月购买	400 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

智能文档分析平台

文档格式转换

支持文档格式转换的在线工具和接口调用两种使用方式。 首次进入在线工具-文档格式转换 页面即可自动领取免费资源，或者在文字识别控制台 也可以自动领取免费资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。 免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或 开通按量后付费。 支持整份多页PDF/图片/OFD转换，**按文件页数计费**，转换一页PDF/一张图片扣除一页。 **预付费资源包** | 规格（页） | 价格（元） | | ---------- | ---------- | | 100 | 18 | | 1000 | 180 | | 5000 | 850 | | 1万 | 1600 | | 5万 | 7500 | | 10万 | 14000 | | 20万 | 26000 | | 50万 | 55000 | | 100万 | 90000 | | 500万 | 350000 |

> **说明：**
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 文档格式转换（元/页） | | ---------- | ---------- | | 不限量 | 0.18 | **说明：** “调用次数”只包括成功调用，调用失败不计费

合同审查

支持合同审查的在线工具和接口调用两种使用方式。 首次进入在线工具-合同审查 页面即可自动领取免费资源，或者在文字识别控制台 也可以自动领取免费资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。 免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或开通按量后付费。 **预付费资源包** | 规格（页） | 价格（元） | | ---------- | ---------- | | 100 | 200 | | 1000 | 1800 | | 1万 | 16000 | | 5万 | 75000 | | 10万 | 140000 | | 50万 | 550000 | | 100万 | 900000 |

> **说明：**
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 合同审查（元/页） | | ---------- | ---------- | | 不限量 | 2 | **说明：** “调用次数”只包括成功调用，调用失败不计费

文档比对

支持文档比对的在线工具和接口调用两种使用方式。 首次进入在线工具-文档比对 页面即可自动领取免费资源，或者在文字识别控制台 也可以自动领取免费资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或开通按量后付费。 **预付费资源包** | 规格（页） | 价格（元） | | ---------- | ---------- | | 100 | 20 | | 1000 | 180 | | 1万 | 1600 | | 5万 | 7500 | | 10万 | 14000 | | 50万 | 55000 | | 100万 | 90000

|

> **说明：**
>
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 文档比对（元/页） | | ---------- | ---------- | | 不限量 | 0.2 | **说明：** “调用次数”只包括成功调用，调用失败不计费

文档抽取

支持文档抽取的在线工具和接口调用两种使用方式。 首次进入在线工具-文档抽取 页面即可自动领取免费资源，或者在文字识别控制台 也可以自动领取免费资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。 免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或开通按量后付费。

预付费资源包 | 规格（页） | 价格（元） | | ---------- | ---------- | | 1000 | 200 | | 5000 | 950 | | 1万 | 1800 | | 5万 | 8500 | | 10万 | 16000 | | 20万 | 30000 | | 50万 | 65000 | | 100万 | 110000 | | 500万 | 450000 |

> **说明：**
>
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 文档抽取（元/页） | | ---------- | ---------- | | 不限量 | 0.2 | **说明：** “调用次数”只包括成功调用，调用失败不计费

文档解析

文档解析支持接口调用的使用方式。 已实名用户可进入文字识别控制台 ，即可自动领取免费测试资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或开通按量后付费。 **预付费资源包** | 规格（页） | 价格（元） | | ---------- | ---------- | | 1000 | 180 | | 5000 | 850 | | 1万 | 1600 | | 5万 | 7500 | | 10万 | 14000 | | 20万 | 26000 | | 50万 | 55000 | | 100万 | 90000 | | 500万 | 350000 |

> **说明：**
>
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 文档解析（元/页） | | ---------- | ---------- | | 不限量 | 0.18 | **说明：** “调用次数”只包括成功调用，调用失败不计费

自定义开发平台

iOCR通用版

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次，企业认证 1,000 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费

预付费次数包

规格（次）	价格（元）
1万	480
5万	2100
10万	3600
20万	6000
50万	12000
100万	20000
500万	85000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 次数包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明

按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=2	0.050
2<调用次数<=5	0.040
5<调用次数<=10	0.032
10<调用次数<=20	0.026
20<调用次数<=30	0.022
30<调用次数	0.020

> **说明：**
>
> - “调用次数”只包括成功调用，调用失败不计费；
> - iOCR识别及分类功能采取统一计费方式：通过iOCR使用预置模板（如身份证、银行卡等），计费标准与iOCR价格保持一致。

QPS叠加包

购买方式	价格
按天购买	20 元/天/QPS
按月购买	360 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

iOCR财会版

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 500 次，企业认证 1,000 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可开通按量后付费

月调用量（万次）	价格（元/次）
0<调用次数<=2	0.080
2<调用次数<=5	0.068
5<调用次数<=10	0.058
10<调用次数<=20	0.050
20<调用次数<=30	0.045
30<调用次数	0.040

> **说明：**
>
> - “调用次数”只包括成功调用，调用失败不计费；
> - 通过iOCR财会版使用预置票据/单据模板（如增值税发票、定额发票、火车票、出租车票等），计费标准以上述表格内价格为准；
> - 如使用混贴票据识别功能，则以实际识别的发票数量为准进行计费（如一张粘贴单上粘贴 4 张火车票，则调用次数计作 4 次）

文档图像处理

文档矫正增强

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 50 次，企业认证 100 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，开通按量后付费后，可购买QPS叠加包。

预付费次数包

规格（次）	价格（元）
1万	1100
5万	5000
10万	9000
20万	16000
50万	35000
100万	60000
500万	250000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买。
> - 次数包购买后7天内若未产生调用可前往 退订管理 页面进行自助退订，详细退订规则见 退订说明。

按量后付费

月调用量	价格（元/次）
不限量	0.13

> **说明：**“调用次数”只包括成功调用，调用失败不计费。

QPS叠加包

购买方式	价格
按天购买	30 元/天/QPS
按月购买	480 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包。
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

文档去手写

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 50 次，企业认证 100 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，开通按量后付费后，可购买QPS叠加包。

预付费次数包

规格（次）	价格（元）
1万	1100
5万	5000
10万	9000
20万	16000
50万	35000
100万	60000
500万	250000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买。
> - 次数包购买后7天内若未产生调用可前往 退订管理 页面进行自助退订，详细退订规则见 退订说明。

按量后付费

月调用量	价格（元/次）
不限量	0.13

> **说明：**“调用次数”只包括成功调用，调用失败不计费。

QPS叠加包

购买方式	价格
按天购买	30 元/天/QPS
按月购买	480 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包。
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

图片去摩尔纹

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 50 次，企业认证 100 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，开通按量后付费后，可购买QPS叠加包。

预付费次数包

规格（次）	价格（元）
1万	800
10万	6000
50万	25000
100万	40000
500万	150000

> **说明：**
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买。
> - 次数包购买后7天内若未产生调用可前往 退订管理 页面进行自助退订，详细退订规则见 退订说明。

按量后付费

月调用量	价格（元/次）
不限量	0.1

> **说明：**“调用次数”只包括成功调用，调用失败不计费。

QPS叠加包

购买方式	价格
按天购买	20 元/天/QPS
按月购买	300 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包。
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

文档图片去底纹

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 50 次，企业认证 100 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，开通按量后付费后，可购买QPS叠加包。

预付费次数包

规格（次）	价格（元）
1万	800
10万	6000
50万	25000
100万	40000
500万	150000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买。
> - 次数包购买后7天内若未产生调用可前往 退订管理 页面进行自助退订，详细退订规则见 退订说明。

按量后付费

月调用量	价格（元/次）
不限量	0.1

> **说明：**“调用次数”只包括成功调用，调用失败不计费。

QPS叠加包

购买方式	价格
按天购买	20 元/天/QPS
按月购买	300 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包。
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

文件检测分类

已完成实名认证的用户，登录进入文字识别控制台，即可自动获取所需接口的免费测试资源。**个人认证 200 次，企业认证 500 次。**

免费测试资源用尽后按照如下价格进行计费。如需付费使用，可 购买次数包 或 开通按量后付费，如需扩充 QPS，开通按量后付费后，可购买QPS叠加包。

预付费次数包

规格（次）	价格（元）
1万	380
5万	1800
10万	3200
20万	5400
50万	11000
100万	18000
500万	72000

> **说明：**
>
> - 次数包购买后一年内有效，有效期内产生计费调用量优先抵扣次数包额度，抵扣完毕后自动转为按量后付费方式，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买。
> - 次数包购买后7天内若未产生调用可前往 退订管理 页面进行自助退订，详细退订规则见 退订说明。

按量后付费

月调用量	价格（元/次）
不限量	0.04

> **说明：**“调用次数”只包括成功调用，调用失败不计费。

QPS叠加包

购买方式	价格
按天购买	10 元/天/QPS
按月购买	180 元/月/QPS

> - 购买 QPS 叠加包需保证已开通按量后付费或购买次数包。
> - 购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效。

费用计算示例

费用计算示例

按量后付费

当月调用身份证识别的总次数中，需计费的调用量为 11 万次，费用计算如下：

付费阶梯	单价	当前阶梯次数	费用
0 < 月调用量 ≤ 5万	0.020元/次	5万次	1000元
5万 < 月调用量 ≤ 10万	0.016元/次	5万次	800元
10万 < 月调用量 ≤ 20万	0.013元/次	1万次	130元

因此，本月调用该接口产生的费用为 50000 x 0.02 + 50000 x 0.016 + 10000 x 0.0013 =**1930 元**

预付费资源包

专项资源包：当月调用身份证识别的总次数中，需计费的调用量为 11 万次，且已购买 10 万次的专项资源包，按照资源包抵扣完毕后自动转为按量后付费方式进行计费，则产生的总花销为：1700 + 10000 x 0.013 = **1890 元**

> **说明**：1700 为 10 万次专项资源包的价格

共享资源包：当月调用身份证识别的总次数中，需计费的调用量为 11 万次，且已购买了 50 万点的共享资源包，按照资源包抵扣完毕后自动转为按量后付费方式进行计费，则产生的总花销为：1540 + 10000 x 0.013 = **1670元**

> **说明**：1540 为 50 万点共享资源包的价格，身份证识别单次成功调用需抵扣 5 个点

如何购买

开通付费

免费资源耗尽后，您可以在控制台选择开通按量后付费或购买预付费资源包。目前文字识别支持的付费方式包括：

1. 按量后付费：基于已产生的调用量进行扣费，支持随开随停，灵活方便。
2. 共享资源包：支持多个接口共用额度的资源包，不同接口单次成功调用所抵扣的点数有所不同。购买后一年有效。
3. 专项资源包：专用于特定一个接口的资源包。购买后一年有效。

当赠送的QPS不足以满足您的业务需求时，您还可以购买QPS叠加包，增加QPS上限。

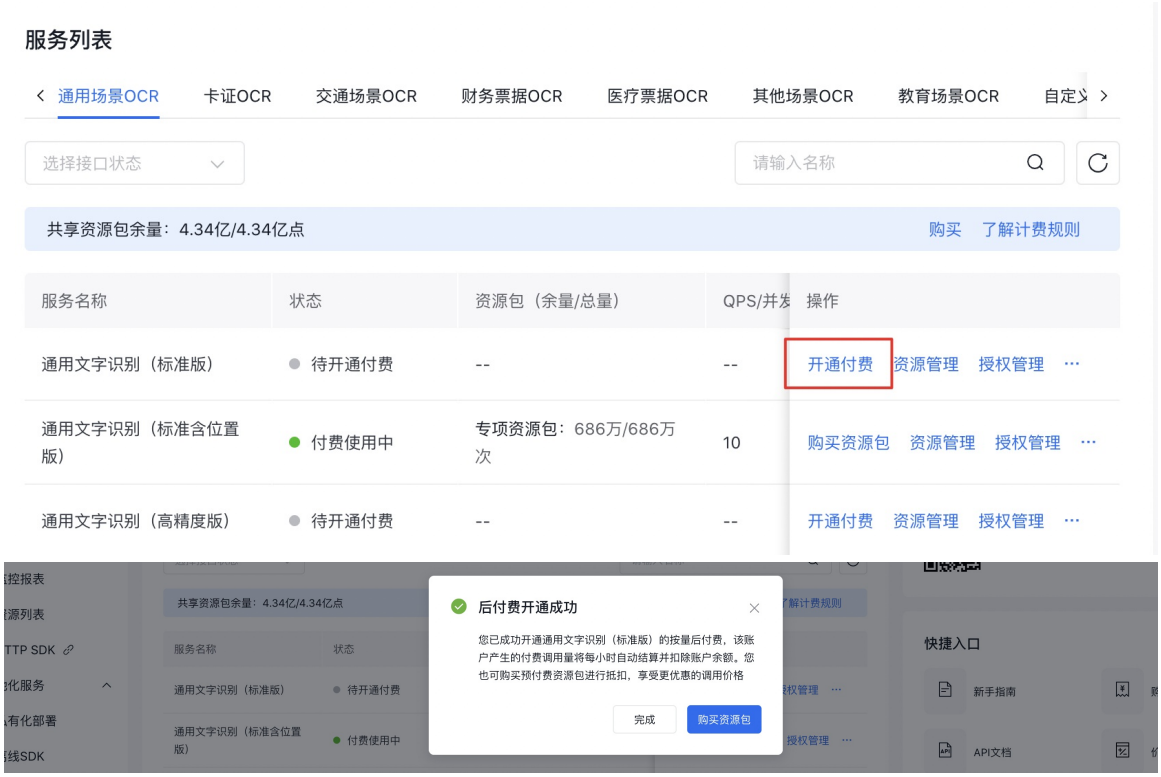
计费规则详情参见计费概述。

您可以在控制台概览页"服务列表"处开通付费并购买所需的资源。具体购买方式如下：

开通按量后付费

支持随开随停，适用于需灵活付费，或前期小规模测试的企业。

在服务列表找到需要开通的服务接口，点击“开通付费”，即可完成开通



如您希望批量开通其他接口，可点击“购买资源包”，在购买页里"接口名称"处批量勾选接口，并调整计费方式为按量后付费。点击确认，即可批量开通接口的按量后付费。

购买预付费资源包

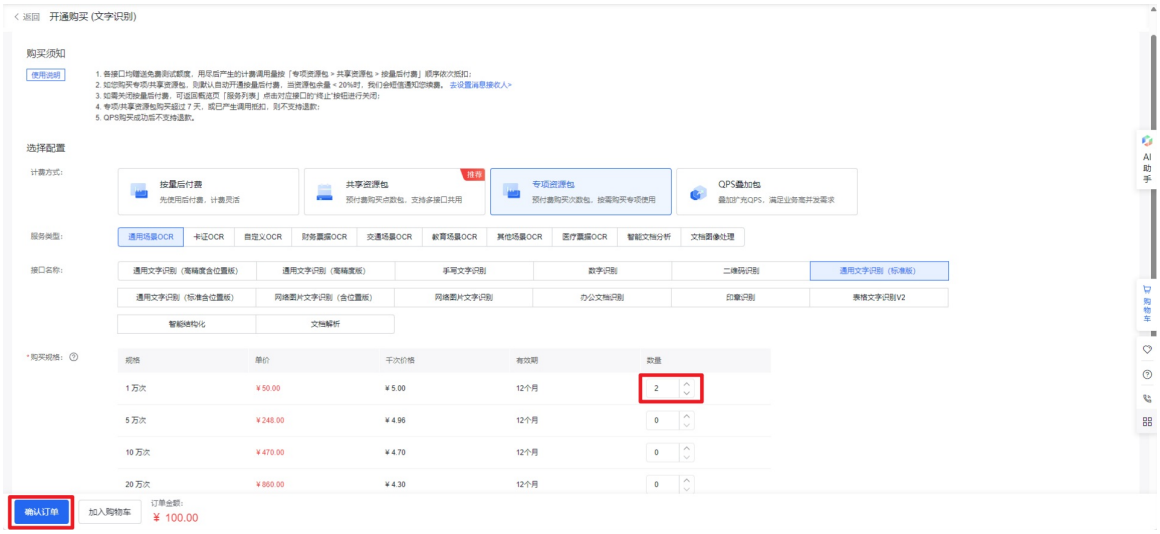
支持预付费，一次购买全年使用，适用于调用量可预估的企业。

- **购买专项资源包**

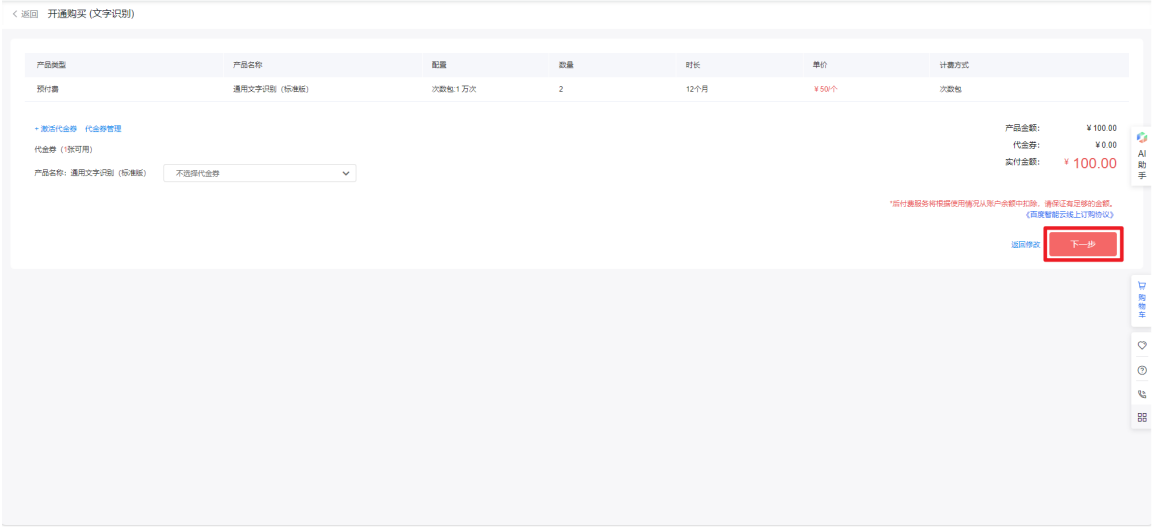
1. 在完成了开通按量后付费后，点击“购买资源包”，进入购买页。



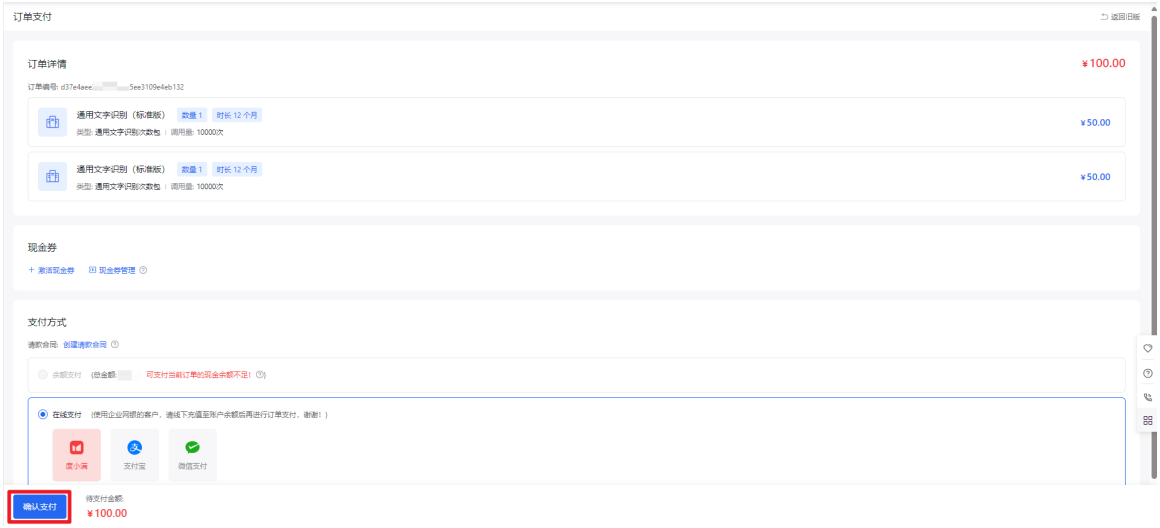
2. 选择所需资源包类别及数量，点击“立即购买”。



3. 进入确认订单页面，确认产品、代金券等信息后，点击“下一步”。



4. 提交订单成功，点击“确认支付”后，完成专项资源包购买。



5. 成功购买专项资源包。



- **购买共享资源包**

1. 在完成了开通按量后付费后，点击“购买资源包”，进入购买页。



2. 切换计费方式到“共享资源包”，点击“添加接口”，可配置共享范围。



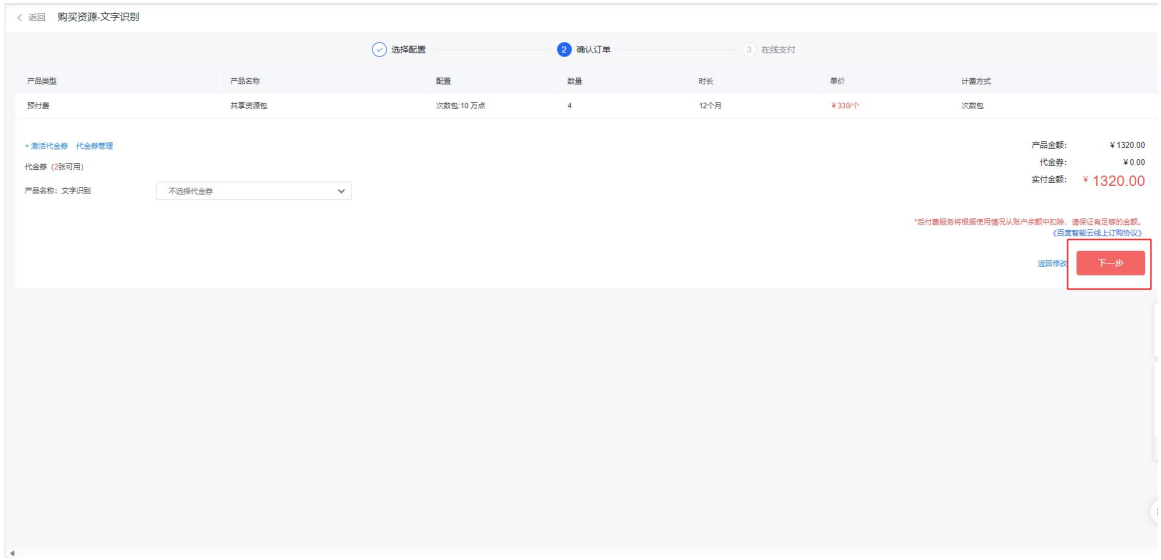
3. 进入添加共享范围弹窗，选择需要加入共享范围的接口，点击“确认”。



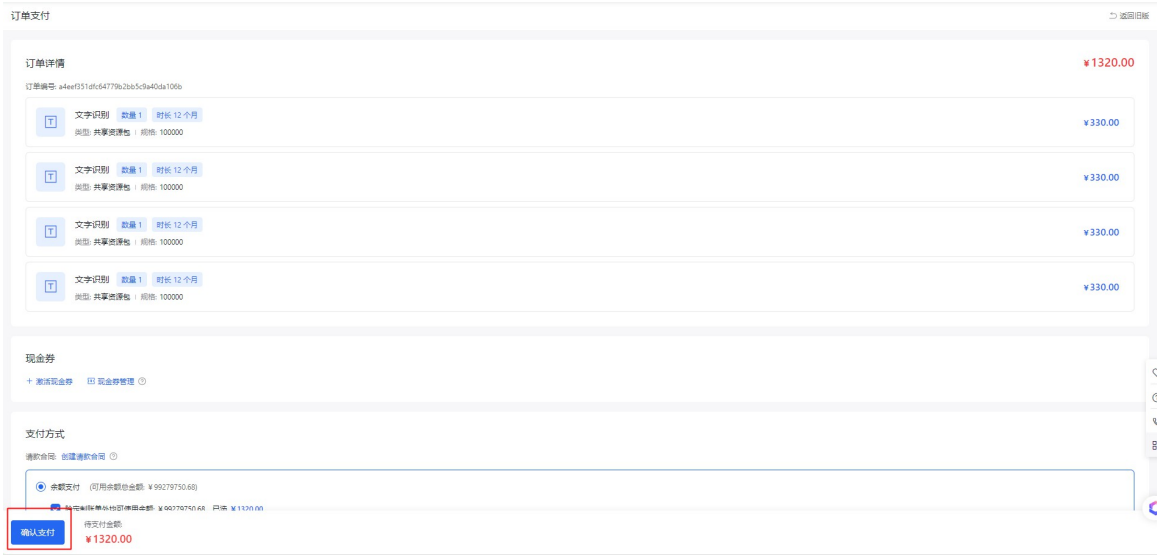
4. 回到购买资源页面，选择所需资源包规格及数量，点击“确认订单”。



5. 进入确认订单页面，确认产品、代金券等信息后，点击“下一步”。



6. 提交订单成功，点击“确认支付”后，完成共享资源包购买。



7. 成功购买共享资源包。



- **管理预付费资源包**

1. 进入文字识别控制台后，选择所需服务，如图位置点击“资源管理”。



2. 在资源管理页里，您可以查看当前服务接口的资源情况详情。顶部可以切换查看其他接口。



购买QPS叠加包

当现有QPS配额不能满足您的业务需求时，您可以购买QPS叠加包，增加QPS上限。

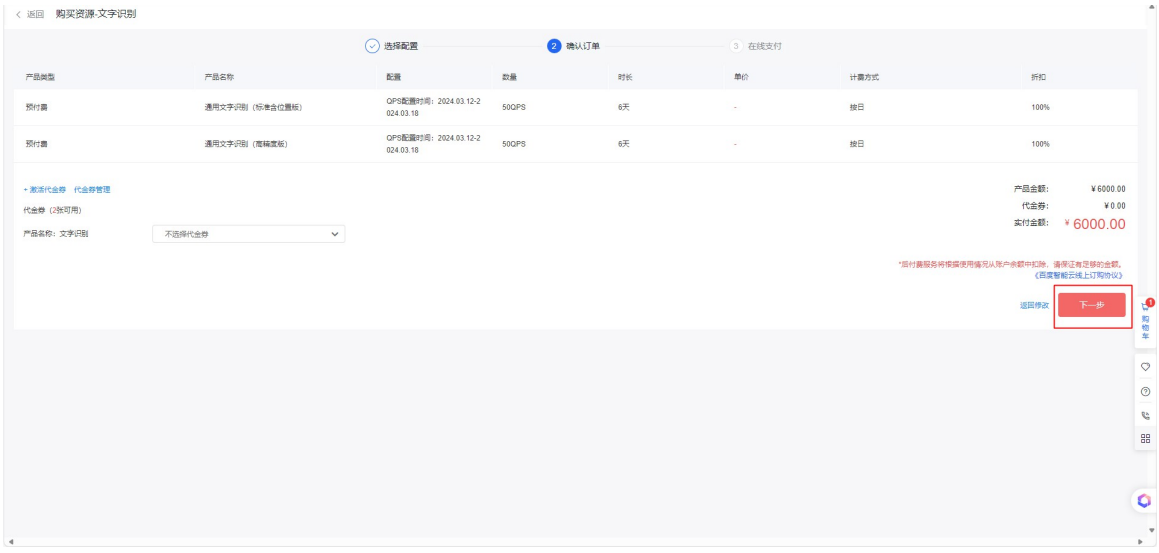
1. 在完成了开通按量后付费后，点击“购买资源包”，进入购买页。



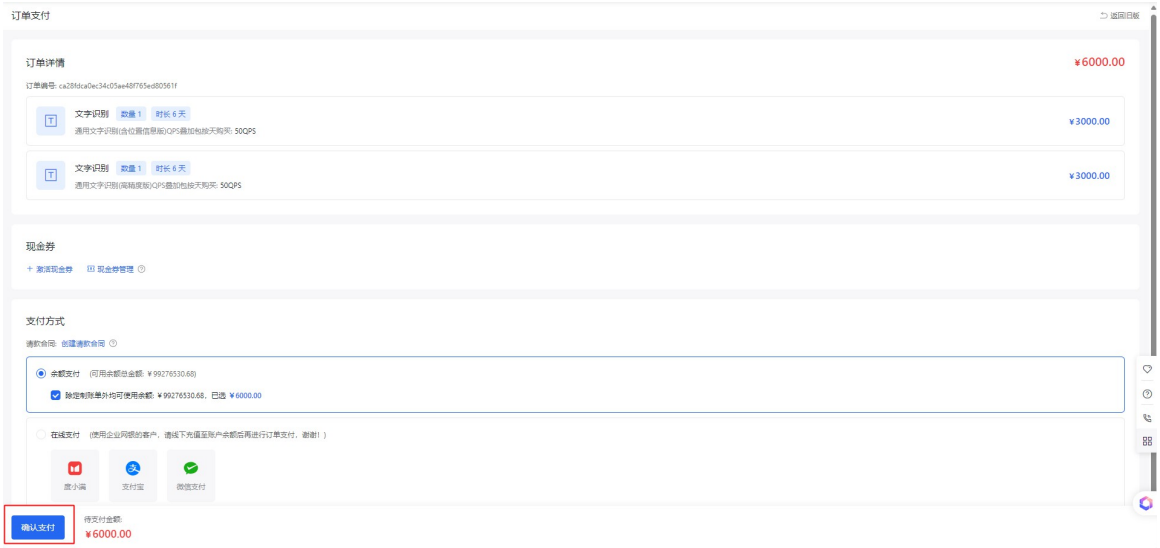
2. 计费方式选择"QPS叠加包"，选择要购买叠加包的接口和购买方式后，点击“确认订单”



3. 进入确认订单页面，确认产品、代金券等信息后，点击“下一步”。



4. 提交订单成功，点击“确认支付”后，完成QPS叠加包购买。



5. 购买QPS后，您可以在资源管理页中查看对应服务的QPS配额。



账户充值

1. 您可以直接点击充值，或者通过以下方式进入充值页面：

- 方式一：在控制台首页，将鼠标移到右上角导航栏的“财务”按钮上方，弹出小窗口，点击“充值”按钮。



- 方式二：在控制台首页，点击右上角导航栏的“财务”按钮，进入“财务中心”，在“账务总览”页面，点击"充值"按钮。



2. 进入充值页面后，您可以在“充值方式”处选择“在线支付”或“线下汇款”。

   - 在线支付：选择支付平台“银联个人”、“银联企业”、“快捷支付”、“支付宝”或“微信支付”。
   - 线下汇款：您可获取您的专属账号，并线下汇款至专属账户，系统会将汇款直接匹配到您的百度开放云账户，实现自动加款，快速到账。适用于所有通过百度智能云完成个人认证及企业认证的用户（暂不支持从百度钱包同步实名状态的用户）。具体操作如下：- 点击汇款信息栏对应的“获取您的专属汇款账号”，获取银行电汇自动加款专属账号。



   - 点击“发送验证码”，输入收到的验证码后，点击“确定”按钮，获取您的专属汇款账号。
   - 到任意银行网点或网银，通过银行电汇完成百度智能云账户加款。

> 1.为保证汇款顺利进行，请务必保证汇款方名称与实名认证名称一致。
>
> 2.汇款方名称将是您后期开增值税专用发票的名称。

3. 选择“充值金额”或手动输入金额，点击“确认充值”。
4. 充值成功后，可点击导航栏的收支明细，查看充值记录。
5. 若充值未成功，点击提交工单并说明情况，我们会立刻派专人处理。

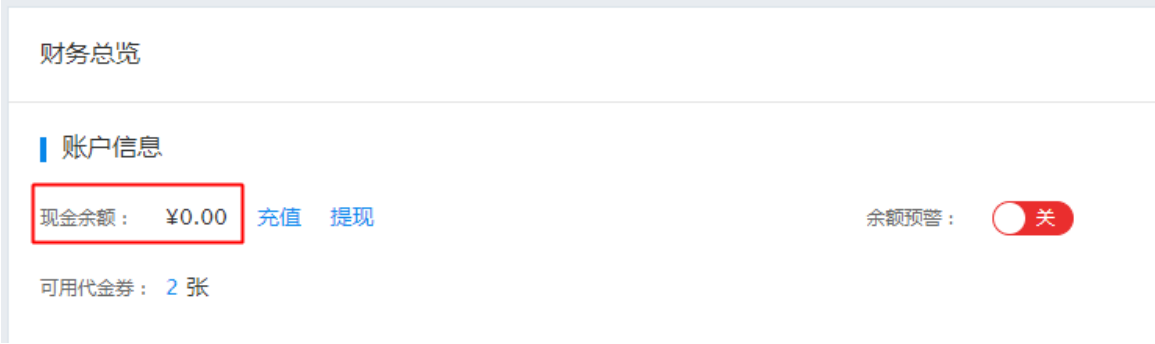
查看余额

有如下两种方式可查看当前账户余额：

- 方式一：在控制台首页会显示当前账户余额。



- 方式二：由控制台的导航栏进入财务中心，在“账务总览”页面，“账户信息”区域会显示账户余额。



> **注意**：如遇产品相关问题，您可寻求 智能助手帮助，或 提交工单，会有专人跟进处理。

API文档

简介

Hi，您好，欢迎使用百度文字识别（OCR）服务。

本文档主要针对API开发者，描述百度OCR文字识别接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 在百度智能云控制台内提交工单，咨询问题类型请选择**人工智能-文字识别**；
- 百度AI社区--文字识别官方版块：http://ai.baidu.com/forum/topic/list/164
- 客服电话：400-920-8999

注意！

请勿通过任何第三方插件使用百度OCR服务，使用第三方非法插件会导致您的 Access Token、Secret Token 泄露，他人即可盗用您的账户进行任意消费，如因此产生恶意消费，需您自行承担责任。

如您已经使用了第三方插件，建议您立即删除对应appid并更新账户密码！

接口能力

接口名称	接口能力简要描述
通用场景文字识别	**对各类通用场景、文件的识别接口，按行返回识别结果**
通用文字识别（标准版）	识别图片中的文字信息
通用文字识别（高精度版）	更高精度地识别图片中的文字信息
通用文字识别（标准含	识别图片中的文字信息（包含文字区域的坐标信息）

位置版）	
通用文字识别（高精度含位置版）	更高精度地识别图片中的文字信息（包含文字区域的坐标信息）
网络图片文字识别	针对网络图片进行专项优化，对艺术字体或背景复杂的文字内容具有更优的识别效果
办公文档识别	可对办公类文档的版面进行分析，输出图、表、标题、文本、目录、栏、页眉、页脚、页码和脚注的位置，并输出分版块内容的OCR识别结果
数字识别	识别图片中的数字，适用于手机号提取、快递单号提取、充值号码提取等场景
手写文字识别	对手写汉字或手写数字进行识别
表格文字识别V2	支持识别图片/PDF格式文档中的表格内容，返回各表格的表头表尾内容、单元格文字内容及其行列位置信息，全面覆盖各类表格样式，包括常规有线表格、无线表格、含合并单元格表格。同时，支持多表格内容识别。
二维码识别	对二维码、条形码中对应的文字内容进行识别
印章识别	检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别
智能结构化	支持智能提取图片中的字段结构化信息，无需训练灵活提取。适用于各类证照、票据、表单等版式中的结构化信息录入场景。
文档解析	文档解析支持对doc、pdf、图片、xlsx等16种格式文档进行解析，输出文档的版面、表格、阅读顺序、标题层级、旋转角度等信息，可返回Markdown格式内容，将非结构化数据转化为易于处理的结构化数据
卡证文字识别	**对各类卡片、证照进行结构化识别，按字段返回识别结果**
身份证识别	对二代居民身份证正反面所有8个字段进行结构化识别
身份证混贴识别	支持自动检测与识别二代居民身份证正反面在同一张图片上的场景，对身份证正反面所有8个字段进行结构化识别
身份证识别(金融加密版)	支持对身份证图片及识别结果进行信息传输加密，对二代居民身份证正反面所有8个字段进行结构化识别
银行卡识别	对银行卡的卡号、有效期、发卡行、卡片类型、持卡人进行结构化识别
营业执照识别	可结构化识别各类版式的营业执照，返回证件编号、社会信用代码、单位名称、地址、法人、类型、成立日期、有效日期、经营范围等关键字段信息
护照识别	支持对中国大陆居民护照的资料页进行结构化识别，包含国家码、姓名、性别、护照号、出生日期、签发日期、有效期至、签发地点
护照识别（港澳台地区及境外）	支持对港澳台地区及境外护照进行结构化识别，包括MRZCode1、MRZCode2、出生日期、国家码、国籍、姓名拼音、性别、护照号、护照类型、有效期
社保卡识别	支持识别全国各地社保卡，支持识别社会保障卡号、姓名、性别、出生日期、银行卡号、有效期限等8个字段
港澳台证件识别	支持识别4类港澳台出入境证件，包含港澳通行证正/反面、台湾通行证正/反面、台胞证（台湾居民来往大陆通行证）正/反面、返乡证（港澳居民来往内地通行证）正/反面，支持识别以上4类证件的全部字段信息。
户口本识别	对出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码字段进行识别
出生医学证明识别	对出生时间、姓名、性别、出生证编号、父亲姓名、母亲姓名字段进行识别
	支持对结婚证进行结构化识别，包括**姓名_男**、**身份证件号_男**、**出生日期_男**、**国籍_男**、**性别_男**、**姓名_**

结婚证识别	**女**、**身份证件号_女**、**出生日期_女**、**国籍_女**、**性别_女**、**结婚证字号**、**持证人**、**备注**、**登记日期**，全部14 个字段
离婚证识别	支持对离婚证进行结构化识别，包括**姓名_男**、**身份证件号_男**、**出生日期_男**、**国籍_男**、**性别_男**、**姓名_女**、**身份证件号_女**、**出生日期_女**、**国籍_女**、**性别_女**、**离婚证字号**、**持证人**、**备注**、**登记日期**，全部14个字段
企业工商信息查询（标准版）	传入企业名称、注册号、统一社会信用代码中的任意一种，即可查询企业的基本信息，包括法人、注册资本、信用代码、经营状态等20+字段
企业工商信息查询（高级版）	传入企业名称、注册号、统一社会信用代码中的任意一种，具体返回企业全维度信息，包括工商基本信息、分支机构信息、企业变更信息、纳税信息、联系信息、企业高管信息、经营异常信息、动产抵押信息、曾用名信息、股东信息、行政处罚信息、行政许可信息、股权出资信息、失信信息、被执行信息等
房产证识别	支持对房产证进行结构化识别，包括权利人、坐落、权利类型、面积、字第号 、不动产单元号、共有情况、用途、使用期限 、登记日期、共有人，全部 11 个字段
企业二要素核验	通过核验企业名称、统一社会信用代码一致性，快速核验企业资质
企业三要素核验	通过核验企业名称、统一社会信用代码、法人姓名一致性，快速核验企业资质
企业四要素核验	比对校验企业名称、统一社会信用代码、法人姓名、注册证件号的一致性，验证企业工商信息
开户许可证识别	支持对开户许可证进行结构化识别，包括公司名称、开户银行、核准号、法人、编号、账号，全部 6 个字段
外国人永久居住证识别	支持对外国人永久居住证进行结构化识别，识别字段包括 Name、Nationality、Sex、出生日期、国籍、失效日期、姓名、性别、签发日期、证件号码、证件版本，全部 11 个字段
食品经营许可证识别	支持对食品经营许可证进行结构化识别，包括经营者名称、社会信用代码、法定代表人、住所、经营场所、主体业态、经营项目、有效期至、许可证编号、日常监督管理机构、日常监督管理人员、发证机关、签发人、签发日期，全部 14 个字段
食品生产许可证识别	支持对食品生产许可证进行结构化识别，包括生产者名称、社会信用代码、法定代表人、住所、生产地址、食品类别、有效期至、许可证编号、日常监督管理机构、日常监督管理人员、投诉举报电话等信息、发证机关、签发人、签发日期，全部 14 个字段
交通场景文字识别	**针对汽车相关场景的各类证件、票据结构化识别**
车牌识别	对机动车蓝牌、绿牌、单/双行黄牌的车牌号码进行识别，并能同时识别图像中的多张车牌
VIN码识别	对车辆车架、挡风玻璃上的VIN码进行识别
驾驶证识别	对机动车驾驶证正本所有10个字段进行结构化识别
行驶证识别	对机动车行驶证主页及副页所有22个字段进行结构化识别
车辆证照混贴识别	对机动车行驶证主页及副页、驾驶证主页及副页在同一张图片上的场景进行结构化识别
机动车销售发票识别	对机动车销售发票的号码、代码、日期、价税合计等26个关键字段进行结构化识别
车辆合格证识别	对车辆合格证的编号、车架号、排放标准、发动机编号等23个关键字段进行结构化识别
二手车销售发票识别	对二手车销售发票的号码、代码、日期、买方、卖方、车牌号、车辆类型、二手车市场等25个关键字段进行结构化识别
机动车登记证书识别	对机动车登记证书的编号、机动车所有人、登记机关、车辆类型、发证机关章等15个关键字段进行结构化识别

磅单识别	结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单
快递面单识别	支持市面上常见版式的快递面单识别，包括申通/圆通/中通/百世汇通/韵达/顺丰/京东/邮政/极兔/天天等面单版式，结构化识别运单号、收/寄件人姓名、收/寄件人电话、收/寄件人地址等字段
道路运输证识别	结构化识别道路运输证的业户名称、地址、车辆号牌、经营许可证、经济类型、车辆类型、吨座位、车辆规格、经营范围、初领日期、备注、发证日期等14个关键字段，支持识别横版及竖版两种道路运输证
财务票据文字识别	**对财务及金融场景各类票据进行结构化识别，按字段返回识别结果（通用票据识别除外）**
智能财务票据识别	对增值税发票、卷票、火车票、出租车票、机票行程单等13类财务票据混贴的图片进行切分识别
银行回单识别	对各大银行的收/付款人户名、账号、开户银行、金额、日期等关键字段进行结构化识别
增值税发票识别	对增值税发票进行文字识别，并结构化返回字段信息，支持增值税专票、普票、电子发票、卷票、区块链发票
增值税发票验真	支持12种增值税发票的真伪及字段信息准确性校验，包括增值税专票、电子专票、普票、电子普票、卷票、通行费增值税电子普票、货运专票、机动车销售发票、二手车销售发票，支持返回票面的全部信息
定额发票识别	对各类定额发票进行结构化识别，可识别发票代码、发票号码、金额、发票所在地、发票金额小写、省、市7个关键字段
通用机打发票识别	对国家/地方税务局发行的横/竖版通用机打发票的号码、代码、日期、合计金额、类型、商品名称字段进行结构化识别
火车票识别	支持对大陆火车票的车票号、始发站、目的站、车次、日期、票价、席别、姓名进行结构化识别
出租车票识别	针对全国各大城市出租车票的发票号码、发票代码、车号、日期、时间、金额进行结构化识别
飞机行程单识别	对飞机行程单中的姓名、始发站、目的站、航班号、日期、票价字段进行结构化识别
汽车票识别	支持对全国范围不同版式汽车票的发票代码、发票号码、到达站、出发站、日期、时间、金额、身份证号、姓名、开票日期10个字段进行结构化识别
过路过桥费发票识别	支持对全国范围不同版式过路、过桥费发票的发票代码、发票号码、入口、出口、日期、时间、金额、省、市9个字段进行结构化识别
船票识别	对全国范围内不同版式的客运船票、货运船票进行结构化识别，包括发票代码、发票号码、发票日期、发票类型、总金额、出发地点、到达地点等7个字段
网约车行程单识别	对各大主要服务商的网约车行程单进行结构化识别，包括滴滴打车、花小猪打车、高德地图、曹操出行、阳光出行，支持识别服务商、行程开始时间、行程结束时间、车型、总金额等16 个关键字段
购物小票识别	支持识别各类商场、超市及药店的购物小票，包括店名、小票号码、机器编号、工号、消费日期、消费时间、总金额、找零、币种、实收金额、优惠金额、打印日期、打印时间、明细商品名称、单价、数量、小计金额等信息
医疗票据文字识别	**对医疗场景各类票据进行结构化识别，按字段返回识别结果**
医疗发票识别	支持识别全国各地门诊/住院发票的 业务流水号、发票号、住院号、门诊号、病例号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付等关键字段。支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数
医疗费用明细识别	支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含项目类型、项目名称、单价、数量、规格、金额
医疗费用结	支持识别全国医疗费用结算单的姓名、出/入院时间、发票总金额、自费金额、医保支付金额等 6 个关键字段

算单识别	支持识别全国医疗费用结算单的姓名、出/入院时间、发票总金额、自费金额、医保支付金额等 6 个关键字段
医疗检验报告单识别	支持识别全国各地医疗检验报告单的姓名、性别、医院名称、报告单名称等关键字段，支持识别检查具体项目的项目名称、结果、单位、参考区间、结果提示等明细字段
医疗诊断报告单识别	支持识别全国各地各医院医疗诊断报告单，包括医院名称、报告名称、姓名、性别、年龄、科室、临床诊断、报告日期、检查部位、检查方法、检查所见、检查提示、建议、肉眼可见 14个字段
病案首页识别	支持识别全国各地病案首页的病案号、姓名、性别、出生日期、身份证号、出/入院科别、住院次数、药物过敏情况等 15 个关键字段
出院小结识别	支持识别全国出院小结的科室、姓名、性别、年龄、入院日期、出院日期、住院天数、入院诊断、出院诊断、出院医嘱等关键字段
入院小结识别	【请点击申请邀测提交需求申请开通该接口使用权限】支持识别全国各地各医院入院小结的姓名、性别、年龄、入院时间、主诉、身份证号、联系人电话、病史采集日期、既往史、现病史、个人史、月经婚育史、工作单位、可靠程度 14个关键字段
诊断证明识别	【请点击申请邀测提交需求申请开通该接口使用权限】支持识别全国各地各医院诊断证明的姓名、性别、年龄、科室、入院时间、出院时间、住院号、出院诊断、出院医嘱 9个关键字段
门诊病历识别	【请点击申请邀测提交需求申请开通该接口使用权限】支持识别全国各地各医院门诊病历的姓名、日期、诊断、检查、主诉、现病史 6个关键字段，及表格区清单项目名称、规格、单价、数量、金额、项目类型等字段
处方笺识别	【请点击申请邀测提交需求申请开通该接口使用权限】支持识别全国各地各医院处方笺的姓名、日期、病人ID、科别 4个关键字段，及表格区清单项目名称、规格、单价、数量、金额、频率、用量、用法等字段
手术记录识别	【请点击申请邀测提交需求申请开通该接口使用权限】支持识别全国各地各医院手术记录的姓名、性别、年龄、科室、麻醉方式、手术操作名称、手术步骤、手术结果、术中所见、术中诊断、术前诊断、术后诊断 12个关键字段
医疗票据类别检测	【请点击申请邀测提交需求申请开通该接口使用权限】支持对医疗发票、医疗费用明细、医疗费用结算单、病案首页、出院小结、增值税发票6项医疗票据的分类，并返回对应票据的图片质量及位置方向信息
教育场景文字识别	**针对教育相关场景所涉及的文字、数字、符号进行识别**
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果
公式识别	对试卷中的数学公式及题目内容进行识别
词典笔文字识别	用于词典笔扫描文字并识别
试卷切题识别	支持对图片/PDF格式文档内的题目自动切分与结构化识别，可按题输出题干、选项、答案等信息，适用于整页试卷、习题册、课本等，可广泛应用于拍照搜题、题库录入、智能判卷等场景
其它场景文字识别	**对一些特殊场景所涉及图片中的文字内容进行识别**
仪器仪表盘读数识别	广泛适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号
门脸文字识别	识别图片中的门脸文字信息，自动过滤非门脸文字内容，接口返回门脸名称、描述文字和置信度
文档图像处理	**对各类文档图片进行矫正、效果增强等操作，提升图片质量**
文档矫正增强	对图片中的文件、卡证、票据等内容进行四角点检测定位，提取主体内容并对其进行矫正，同时可选图片增强效果进一步提升图片清晰度，达到主体检测矫正并增强的目的，提升图片整体质量
文档去手写	去除图片中的手写内容，保留印刷体内容，可用于试卷去手写还原等场景
图片去摩尔纹	去除翻拍电脑、手机等显示屏照片中的摩尔纹，使图片更加清晰

纹	
文档图片去底纹	自动识别并去除文档图片中的底纹，使图片更加清晰，便于阅读
文件分类检测	【请点击申请邀测提交需求申请开通该接口使用权限】对图片中的文档、卡证、票据等含文字的主体内容进行检测、分类，返回类别及位置信息

调用方式

请求格式

POST方式调用

注意：Content-Type为`application/x-www-form-urlencoded`，然后通过`urlencode`格式化请求体。

返回格式

JSON格式

请求限制

请求图片需经过`base64编码`及`urlencode`后传入：图片的base64编码指将一副图片数据编码成一串字符串，使用该字符串代替图像地址。您可以首先得到图片的二进制，然后去掉编码头后再进行urlencode。

注意：

1. 图片的base64编码是不包含图片头的，如`（）`；
2. 使用 Postman 工具或 Python、PHP 等请求库会自动进行 urlencode，无需自行处理。

请求格式支持：PNG、JPG、JPEG、BMP、TIFF、PNM、WebP

接口名称	图片编码后大小限额
百度文字识别所有接口的图像大小限制	base64编码urlencode后大小不超过4M，最短边至少15px，最长边最大4096px

调用方式

调用AI服务相关的API接口有两种调用方式，两种不同的调用方式采用相同的接口URL。

区别在于**请求方式**和**鉴权方法**不一样，请求参数和返回结果一致。

【如果您对于使用API调用的方式很陌生，您可以参见:文字识别API在线调用教程】

调用方式一

请求URL数据格式

向API服务地址使用POST发送请求，必须在URL中带上参数：

access_token： 必须参数，参考“Access Token获取”。

> 注意：`access_token`的有效期为30天，需要每30天进行定期更换；

POST中参数按照API接口说明调用即可。

例如文字识别API，使用HTTPS POST发送：

```
https://aip.baidubce.com/rest/2.0/ocr/v1/general?
access_token=24.f9ba9c5241b67688bb4adbed8bc91dec.2592000.1485570332.282335-8574074
```

获取access_token示例代码

Bash

PHP

Java

Python

CPP

C#

Node

```
#### !/bin/bash
curl -i -k 'https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&client_id=【百度云应用的AK】&client_secret=【百度云应用的SK】'
```

> **说明**：方式一鉴权使用的Access_token必须通过API Key和Secret Key获取。

调用方式二

请求头域内容

在请求的HTTP头域中包含以下信息：

- host（必填）
- x-bce-date （必填）
- x-bce-request-id（选填）
- authorization（必填）
- content-type（必填）
- content-length（选填）

作为示例，以下是一个标准的文字识别的请求头域内容:

```
POST /rest/2.0/ocr/v1/accurate_basic HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: 2015-03-24T13:02:00Z
connection: keep-alive
accept: */*
host: aip.baidubce.com
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
content-type: application/x-www-form-urlencoded
authorization: bce-auth-v1/46bd9968a6194b4bbdf0341f2286ccce/2015-03-24T13:02:00Z/1800/host;x-bce-date/994014d96b0eb26578e039fa053a4f9003425da4bfedf33f4790882fb4c54903
```

> **说明**：方式二鉴权使用的API认证机制authorization必须通过百度云的AK/SK生成。

通用场景文字识别

通用文字识别（高精度版）

接口描述

在通用文字识别的基础上，提供更高精度的识别服务，支持更多语种识别（丹麦语、荷兰语、马来语、瑞典语、印尼语、波兰语、罗马尼亚语、土耳其语、希腊语、匈牙利语、泰语、越语、阿拉伯语、印地语及部分中国少数民族语言），并将字库从1w+扩展到2w+，能识别所有常用字和大部分生僻字。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/accurate_basic`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和image/url/pdf_f	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存

	ile 四选一			在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
language_type	否	string	auto_detect CHN_ENG ENG JAP KOR FRE SPA POR GER ITA RUS DAN DUT MAL SWE IND POL ROM TUR GRE HUN THA VIE ARA HIN	识别语言类型，默认为CHN_ENG 可选值包括： - auto_detect：自动检测语言，并识别 - CHN_ENG：中英文混合 - ENG：英文 - JAP：日语 - KOR：韩语 - FRE：法语 - SPA：西班牙语 - POR：葡萄牙语 - GER：德语 - ITA：意大利语 - RUS：俄语 - DAN：丹麦语 - DUT：荷兰语 - MAL：马来语 - SWE：瑞典语 - IND：印尼语 - POL：波兰语 - ROM：罗马尼亚语 - TUR：土耳其语 - GRE：希腊语 - HUN：匈牙利语 - THA：泰语 - VIE：越南语 - ARA：阿拉伯语 - HIN：印地语
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向
paragraph	否	string	true/false	是否输出段落信息
probability	否	string	true/false	是否返回识别结果中每一行的置信度
multidirectional_recognize	否	string	true/false	是否开启行级别的多方向文字识别，可选值包括: - true：识别 - false：不识别 若图内有不同方向的文字时，建议将此参数设置为“true”

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/accurate_basic?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	识别结果数组
+ words	否	string	识别结果字符串
+ probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值，当 probability=true 时返回该字段
paragraphs_result	否	array[]	段落检测结果，当 paragraph=true 时返回该字段
+ words_result_idx	否	array[]	一个段落包含的行序号，当 paragraph=true 时返回该字段
paragraphs_result_num	否	uint32	识别结果数，表示 paragraphs_result的元素个数，当 paragraph=true 时返回该字段
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
ofd_file_size	否	string	传入OFD文件的总页数，当 ofd_file 参数有效时返回该字段

返回示例


```
{
  "log_id": 1390582998516105216,
  "words_result_num": 2
  "words_result": [
    {
      "words": " OCR"
    },
    {
      "words": "百度通用文字识别高精度版"
    }
  ]
}
```


通用文字识别（高精度含位置版）

接口描述

提供多场景、多语种、高精度的整图文字检测和识别服务，支持生僻字识别，并支持 25 种语言识别，相对于通用文字识别（含位置信息版）该产品精度更高，但是识别耗时会稍长。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668425998119#/api?product=AI&project=文字识别&parent=通用场景OCR&api=rest/2.0/ocr/v1/accurate&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/accurate`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
language_type	否	string	auto_detect CHN_ENG ENG JAP KOR FRE SPA POR GER ITA RUS N	识别语言类型，默认为CHN_ENG 可选值包括： - auto_detect：自动检测语言，并识别 - CHN_ENG：中英文混合 - ENG：英文 - JAP：日语 - KOR：韩语 - FRE：法语 - SPA：西班牙语 - POR：葡萄牙语 - GER：德语 - ITA：意大利语 - RUS：俄语 - DAN：丹麦语 - DUT：荷兰语 - MAL：马来语 - SWE：瑞典语 - IND：印尼语 - POL：波兰语 - ROM：罗马尼亚语 - TUR：土耳其语 - GRE：希腊语 - HUN：匈牙利语 - THA：泰语 - VIE：越南语 - ARA：阿拉伯语 - HIN：印地语
eng_granularity	否	string	word/letter	表示识别语言类型为「中英文（CHN_ENG）」的情况下，英文的单字符结果是按照单词（word）维度输出还是字母（letter）维度输出，当 recognize_granularity=small 时生效
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向 - false：不检测朝向 输入非正向图片时，若想要达到较好识别效果，建议将此参数设置为“true”
vertexes_location	否	string	true/false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
paragraph	否	string	true/false	是否输出段落信息
[illegible]	否	[illegible]	[illegible]	是否返回识别结果中每一行的置信度

| probability | 否 | string | true/false | 是否返回识别结果中每一行的置信度
|
| char_probability | 否 | string | true/false | 是否返回单字符置信度，默认不返回，当 recognize_granularity = small 时，参数有效。可选值包括：
- true：返回单字符置信度，
- false：不返回单字符置信度
|
| multidirectional_recognize | 否 | string | true/false | 是否开启行级别的多方向文字识别，可选值包括:
- true：识别
- false：不识别
若图内有不同方向的文字时，建议将此参数设置为“true”
 |

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/accurate?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
##### encoding:utf-8

import requests
import base64

'''
通用文字识别（高精度含位置版）
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/accurate"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 通用文字识别（高精度含位置版）
*/
public class Accurate {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String accurate() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/accurate";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        Accurate.accurate();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/accurate";
static std::string accurate_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    accurate_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 通用文字识别（高精度含位置版）
* @return 调用成功返回0，发生错误返回其他错误码
*/
int accurate(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = accurate_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/accurate?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class Accurate
    {
        // 通用文字识别（高精度含位置版）
        public static string accurate()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/accurate?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("通用文字识别（高精度含位置版）:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	识别结果数组
+ words	否	string	识别结果字符串
+ location	是	array[]	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标

++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时返回该字段
++ char	否	string	单字符识别结果，当 recognize_granularity=small 时返回该字段
++ char_prob	否	uint32	单字符置信度，当 recognize_granularity=small 且 char_probability=true 时返回该字段
++ location	否	array[]	位置数组（坐标0点为左上角），当 recognize_granularity=small 时返回该字段
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标，当 recognize_granularity=small 时返回该字段
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标，当 recognize_granularity=small 时返回该字段
+++ width	否	uint32	表示定位定位位置的长方形的宽度，当 recognize_granularity=small 时返回该字段
+++ height	否	uint32	表示定位位置的长方形的高度，当 recognize_granularity=small 时返回该字段
+ probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值，当 probability=true 时返回该字段
+ vertexes_location	否	array[]	识别结果中每一行的外包四边形点坐标，当 vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ finegrained_vertexes_location	否	array[]	识别结果中每一行的多边形轮廓点坐标，当 vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ min_finegrained_vertexes_location	否	array[]	表示 finegrained_poly_location对应的最小外包矩形点坐标，当 vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
paragraphs_result_num	否	uint32	识别结果数，表示 paragraphs_result的元素个数，当 paragraph=true 时返回该字段
paragraphs_result	否	array[]	段落检测结果，当 paragraph=true 时返回该字段
+ words_result_idx	否	array[]	一个段落包含的行序号，当 paragraph=true 时返回该字段
+ finegrained_vertexes_location	否	array[]	识别结果中每一行的多边形轮廓点坐标，当 paragraph=true && vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ min_finegrained_vertexes_location	否	array[]	识别结果中每一行的多边形轮廓点坐标，当 paragraph=true && vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
ofd_file_size	否	string	传入OFD文件的总页数，当 ofd_file 参数有效时返回该字段

返回示例

```JSON
{
    "log_id": 1390584857033179136,
    "words_result_num": 2
    "words_result": [
        {
            "words": " OCR",
            "location": {
                "top": 19,
```

```
            "left": 54,
            "width": 119,
            "height": 46
        }
    },
    {
        "words": "百度通用文字识别高精度版",
        "location": {
            "top": 85,
            "left": 54,
            "width": 206,
            "height": 37
        }
    }
],
}
```


通用文字识别（标准版）

接口描述

基于业界领先的深度学习技术，提供多场景、多语种、高精度的整图文字检测和识别服务，多项ICDAR指标居世界第一。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668425998119#/api?product=AI&project=文字识别&parent=通用场景OCR&api=rest/2.0/ocr/v1/general_basic&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url、pdf_file、ofd_file 字段失效

：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效 |
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
language_type	否	string	CHN_ENG ENG JAP KOR FRE SPA POR GER ITA RUS	识别语言类型，默认为CHN_ENG 可选值包括： - CHN_ENG：中英文混合 - ENG：英文 - JAP：日语 - KOR：韩语 - FRE：法语 - SPA：西班牙语 - POR：葡萄牙语 - GER：德语 - ITA：意大利语 - RUS：俄语
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true/false	是否检测语言，默认不检测，即：false。当前支持中文、英语、日语、韩语
paragraph	否	string	true/false	是否输出段落信息
probability	否	string	true/false	是否返回识别结果中每一行的置信度

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
通用文字识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 通用文字识别
*/
public class GeneralBasic {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String generalBasic() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        GeneralBasic.generalBasic();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic";
static std::string generalBasic_result;
/**
```

```
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    generalBasic_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 通用文字识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int generalBasic(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = generalBasic_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
```

```
    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class GeneralBasic
    {
        // 通用文字识别
        public static string generalBasic()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("通用文字识别:");
            Console.WriteLine(result);
            return result;
```

```
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```

返回说明

返回参数

字段	是否必选	类型	说明
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	识别结果数组
+ words	否	string	识别结果字符串
+ probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值，当 probability=true 时返回该字段
paragraphs_result	否	array[]	段落检测结果，当 paragraph=true 时返回该字段
+ words_result_idx	否	array[]	一个段落包含的行序号，当 paragraph=true 时返回该字段
paragraphs_result_num	否	uint32	识别结果数，表示 paragraphs_result的元素个数，当 paragraph=true 时返回该字段
language	否	int32	语种类型，当 detect_language=true 时返回该字段。 - -1：未定义， - 0：英文， - 1：日文， - 2：韩文， - 3：中文
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
ofd_file_size	否	string	传入OFD文件的总页数，当 ofd_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 2471272194,
  "words_result_num": 2,
  "words_result":
   [
    {"words": " TSINGTAO"},
    {"words": "青岛啤酒"}
   ]
}
```


通用文字识别（标准含位置版）

接口描述

基于业界领先的深度学习技术，提供多场景、多语种、高精度的整图文字检测和识别服务。在通用文字识别的基础上，返回文字在图片中的位置信息，方便用户进行版式的二次处理。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/general`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_n	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的

um	否			对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
language_type	否	string	CHN_ENG ENG JAP KOR FRE SPA POR GER ITA RUS	识别语言类型，默认为CHN_ENG 可选值包括： - CHN_ENG：中英文混合 - ENG：英文 - JAP：日语 - KOR：韩语 - FRE：法语 - SPA：西班牙语 - POR：葡萄牙语 - GER：德语 - ITA：意大利语 - RUS：俄语
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向 - false：不检测朝向 输入非正向图片时，若想要达到较好识别效果，建议将此参数设置为“true”
detect_language	否	string	true/false	是否检测语言，默认不检测，即：false。当前支持中文、英语、日语、韩语
paragraph	否	string	true/false	是否输出段落信息
vertexes_location	否	string	true/false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	string	true/false	是否返回识别结果中每一行的置信度

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/general?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
language	否	int32	语种类型，当 detect_language=true 时返回该字段。 - -1：未定义， - 0：英文， - 1：日文， - 2：韩文， - 3：中文
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	识别结果数组
+ words	否	string	识别结果字符串
+ location	是	array[]	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时返回该字段
++ char	否	string	单字符识别结果，当 recognize_granularity=small 时返回该字段
++ location	否	array[]	位置数组（坐标0点为左上角），当 recognize_granularity=small 时返回该字段
			表示定位位置的长方形左上顶点的水平坐标，当 recognize_granularity=small 时返

+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标，当 recognize_granularity=small 时返回该字段
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标，当 recognize_granularity=small 时返回该字段
+++ width	否	uint32	表示定位定位位置的长方形的宽度，当 recognize_granularity=small 时返回该字段
+++ height	否	uint32	表示位置的长方形的高度，当 recognize_granularity=small 时返回该字段
+ probability	否	object	表示识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值，当 probability=true 时返回该字段
+ vertexes_location	否	array[]	识别结果中每一行的外包四边形点坐标，当 vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ finegrained_vertexes_location	否	array[]	识别结果中每一行的多边形轮廓点坐标，当 vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ min_finegrained_vertexes_location	否	array[]	表示 finegrained_poly_location对应的最小外包矩形点坐标，当 vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
paragraphs_result	否	array[]	段落检测结果，当 paragraph=true 时返回该字段
+ words_result_idx	否	array[]	一个段落包含的行序号，当 paragraph=true 时返回该字段
+ finegrained_vertexes_location	否	array[]	识别结果中每一行的多边形轮廓点坐标，当 paragraph=true && vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ min_finegrained_vertexes_location	否	array[]	表示 finegrained_poly_location对应的最小外包矩形点坐标，当 paragraph=true && vertexes_location=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
paragraphs_result_num	否	uint32	识别结果数，表示 paragraphs_result 的元素个数，当 paragraph=true 时返回该字段
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
ofd_file_size	否	string	传入OFD文件的总页数，当 ofd_file 参数有效时返回该字段

返回示例


```
{
  "log_id": 1390595283741573120,
  "words_result_num": 2
  "words_result": [
    {
      "words": " OCR",
      "location": {
```

```
            "location": {
              "top": 14,
              "left": 47,
              "width": 132,
              "height": 54
            }
        },
        {
            "words": "百度文字识别",
            "location": {
              "top": 84,
              "left": 47,
              "width": 204,
              "height": 39
            }
        }
    ],
}
```


办公文档识别

接口描述

支持对各类办公文档进行版面分析和文字识别，输出图、表、印章、标题等元素及位置信息，并分版块输出文字识别结果。可支持中、英、日、韩、法等 20+ 语言类型，印刷、手写、混排等多种场景。

在线调试

您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668425998119#/api?product=AI&project=文字识别&parent=通用场景OCR&api=rest/2.0/ocr/v1/doc_analysis_office&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis_office`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片

base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链** |
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码，要求编码后大小不超过10M，最短边至少15px，最长边最大8192px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
language_type	否	string	auto_detect CHN_ENG ENG JAP KOR FRE SPA POR GER ITA RUS N	识别语言类型，默认为CHN_ENG 可选值包括： -auto_detect：自动检测语言，并识别 -CHN_ENG：中英文 -ENG：英文 - JAP：日语 - KOR：韩语 - FRE：法语 - SPA：西班牙语 - POR：葡萄牙语 - GER：德语 - ITA：意大利语 - RUS：俄语 - DAN：丹麦语 - DUT：荷兰语 - MAL：马来语 - SWE：瑞典语 - IND：印尼语 - POL：波兰语 - ROM：罗马尼亚语 - TUR：土耳其语 - GRE：希腊语 - HUN：匈牙利语 - THA：泰语 - VIE：越南语 - ARA：阿拉伯语 - HIN：印地语
result_type	否	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 -big：返回行识别结果 -small：返回行识别结果之上还会返回单字结果
char_probability	否	string	true/false	是否返回单字符置信度，默认不返回，当 result_type = small 时，参数有效。可选值包括： - true：返回单字符置信度 - false：不返回单字符置信度
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， -0 ：正向 -1：逆时针旋转90度 -2：逆时针旋转180度 -3：逆时针旋转270度
line_probability	否	string	true/false	是否返回每行识别结果的置信度。默认为false
disp_line_poly	否	string	true/false	是否返回每行的四角点坐标。默认为false
words_type	否	string	handwring_only/ handprint_mix	文字类型。 默认：手写印刷混排识别 -handwring_only：手写文字识别 - handprint_mix： 手写印刷混排识别
layout_analysis	否	string	true/false	是否分析文档版面：包括layout（图、表、标题、段落、目录、印章）；attribute（栏、页眉、页脚、页码、脚注）的分析输出。默认为false
recg_tables	否	string	true/false	是否识别并输出表格相关信息，包括单元格内容。默认为false
recog_seal	否	string	true/false	是否识别并输出印章相关信息。默认为false
recg_formula	否	string	true/false	是否检测并识别公式，公式以Latex格式返回。默认为false
erase_seal	否	string	true/false	是否先擦除水印、印章后再识别文档。默认为false
disp_underline_analysis	否	string	true/false	是否识别并输出下划线，默认false
请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
##### 办公文档识别
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis_office?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
办公文档识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis_office"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 办公文档识别
*/
public class AnalysisOffice {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String analysisOffice() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis_office";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        AnalysisOffice.analysisOffice();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis_office";
static std::string analysisOffice_result;
/**
 * curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
 * @param 参数定义见libcurl文档
 * @return 返回值定义见libcurl文档
 */
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    analysisOffice_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
 * 办公文档识别
 * @return 调用成功返回0，发生错误返回其他错误码
 */
int analysisOffice(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = analysisOffice_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
 * 发起http post请求(REST API), 并获取REST请求的结果
 * @param string $url
 * @param string $param
 * @return - http response body if succeeds, else false.
 */
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis_office?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class AnalysisOffice
    {
        // 办公文档识别
        public static string analysisOffice()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis_office?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("办公文档识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

```
 ###### 返回说明

**返回参数**

| 字段    | 是否必选 | 类型       | 说明                                  |
| ------- | -------- | ---------- | ------------------------------------- |
| log_id  | 是    | uint64     | 唯一的log id，用于问题定位            |
| img_direction | 否 | int32 |detect_direction=true 时返回该字段。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度 |
| results_num       | 是     | uint32  | 识别结果数，表示results的元素个数   |
| results | 是   | array[] | 识别结果数组，当 recg_formula=true 时，返回含有公式的结果 |
| + words_type | 是 | string | 文字属性（手写、印刷），handwriting 手写，print 印刷 |
| + words | 是 | array[] | 整行的识别结果数组 |
| ++ line_probability | 否 | array[] | line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值 |
| +++ average | 否 | float | 行置信度 |
| +++ min | 否 | float | 整行中单字的最低置信度 |
```

++ word	是	float	整行的识别结果
++ poly_location	否	array[]	是否返回每行的四角点坐标，自左上角点顺时针排列，disp_line_poly=true时返回
++ words_location	是	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位定位位置的长方形的宽度
+++ height	是	uint32	表示位置的长方形的高度
+ chars	否	array[]	result_type=small时返回。单字符结果数组
++ char	否	string	result_type=small时返回。每个单字的内容
++ char_prob	否	uint32	result_type=small 且 char_probability=true 时返回。单字符置信度
++ chars_location	否	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
underline	否	array[]	识别到的下划线识别结果，当disp_underline_analysis=true时返回
+points	否	array[]	下划线坐标信息
++end_x	否	uint32	下划线终点x坐标
++end_y	否	uint32	下划线终点y坐标
++start_x	否	uint32	下划线起点x坐标
++start_y	否	uint32	下划线起点y坐标
+prob	否	array[]	下划线置信度，取值范围在[0，1]之间
layouts_num	否	uint32	版面分析结果数，表示layout的元素个数。layout_analysis=true时返回
layouts	否	array[]	每个「栏：section」里面的文档版面模块数组，包含表格、图、段落文本、段落标题、表标题、图标题、文档标题、目录、印章等9个模块；每个模块的坐标位置；段落文本和表格内文本内容对应的行序号id。layout_analysis=true时返回
+ layout	否	string	版面分析的标签结果。表格:table， 图:figure，文本:text，段落标题:title ，目录:contents，印章:seal，表标题: table_title，图标题: figure_title，文档标题：doc_title
+layout_prob	否	float	当前版式检测框的概率大小
+ layout_location	否	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	水平坐标（坐标0点为左上角）
+ layout_idx	否	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）
sec_rows	否	uint32	将所有的版面中的「栏:section」内容表示成 M x N 的网格，sec_rows = M。layout_analysis=true时返回
sec_cols	否	uint32	将所有的版面中的「分栏」内容表示成 M x N 的网格，sec_cols = N。layout_analysis=true时返回
sections	否	array[]	一张图片中包含的5大版面属性，包含：栏，页眉，页脚，页码，脚注，该数组里有属性的标签、属性的位置、属性所包含文本内容的id序号。 其中，栏（section）里面包含9个模块内容，有：表格、图、段落文本、段落标题、表标题、图标题、文档标题、目录、印章（在返回参数layouts里输出）。layout_analysis=true时返回
+ attribute	否	string	版面分析的属性标签结果，栏:section, 页眉:header, 页脚:footer, 页码:number，脚注:footnote。
+sections_prob	否	float	当前版面检测框的概率大小
+ attri_location	否	object	版面分析的属性所在位置，四个顶点: 左上，右上，右下，左下
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	水平坐标（坐标0点为左上角）
+ sec_idx	否	object	sections返回参数中的5个版面属性里，包含的内容序号标识数组
++ idx	否	string	sections返回参数中的5个版面属性里，每个属性下包含的文本行id序号
++ para_idx	否	string	当且仅当attribute=section时才会返回。表示，返回参数中的「栏：section」里面，所包含的表格、图、段落文本、段落标题、表标题、图标题、文档标题、目录、印章等9个模块返回的顺序号id（即layouts返回结果中，每个模块的返回顺序号）
++ row_idx	否	string	当且仅当attribute=section时才会返回。表示，将所有栏表示成 M xN 的网格，所属网格的行的id。
++ col_idx	否	string	当且仅当attribute=section时才会返回。表示，将所有栏表示成 M xN 的网格，所属网格的列的id。
table_num	是	int	检测到的表格数量，当recg_tables=true时返回
tables_result	是	array[]	每个表格的内容数组，当recg_tables=true时返回
+table_location	是	array[]	单个表格位置，四角点的x,y 坐标
+header	是	array[]	表头信息
++ location	是	array[]	表头位置，四角点的x,y 坐标

++words	是	string	表头信息，按行拆分
+body	是	array[]	单元格信息
++cell_location	是	array[]	单元格四角点的x,y 坐标
++row_start	是	array[]	单元格行起始编号，横线编号从0开始
++ row_end	是	array[]	单元格行终止编号
++ col_start	是	array[]	单元格列起始编号，竖线编号从0开始
++ col_end	是	array[]	单元格列终止编号
++ words	是	string	单元格文字内容
++contents	否	array[]	单元格里的文本信息，分行展示
+++poly_location	否	array[]	单元格每行文本位置信息
+++ word	否	string	单元格每行文本内容
+footer	是	array[]	表尾信息
++ location	是	array[]	表尾位置，四角点的x,y 坐标	
++ words	是	string	表尾信息，按行拆分	
seal_recog_num	否	uint32	识别到的印章结果数，当recog_seal=true时返回	
seal_recog_results	否	array[]	印章内容数组，当recog_seal=true时返回	
+location	否	object	印章位置信息（坐标0点为左上角）	
++left	否	uint32	表示定位位置的长方形左上顶点的水平坐标	
++top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标	
++width	否	uint32	表示定位位置的长方形的宽度	
++height	否	uint32	表示定位位置的长方形的高度	
+probability	否	float	每一个印章的置信度值	
+type	否	string	印章的类别，共有circle（圆章），ellipse（椭圆章），rectangle（方章）三种
+major	否	object	印章内主字段信息
++words	否	string	主字段识别内容，即章内上环弯曲文字结果
++probability	否	float	主字段识别内容的置信度
+minor	否	array[]	印章内其他字段信息，即除主字段外的识别内容均放置于该参数中返回，若章内不存在其他字段则该参数为空
++words	否	string	其他字段识别内容
++probability	否	float	其他字段识别内容的置信度
formula_result	否	array[]	识别到的公式数组，包括公式位置和公式内容，当 recg_formula=true 时返回
+ form_location	否	array[]	公式的位置信息，矩形框坐标数组（坐标0点为左上角）
+ form_words	否	string	公式的内容信息，以Latex格式返回
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
ofd_file_size	否	string	传入OFD文件的总页数，当 ofd_file 参数有效时返回该字段

返回示例

```JSON
{
 "results_num": 5,
 "log_id": "1410491260247950412",
 "results": [
  {
   "words_type": "print",
   "words": {
    "words_location": {
     "top": 88,
     "left": 442,
     "width": 142,
     "height": 49
    },
    "word": "行程单"
   }
  },
  {
   "words_type": "print",
   "words": {
    "words_location": {
     "top": 241,
     "left": 439
```

```
    "left": 439,
    "width": 393,
    "height": 37
   },
   "word": "美国东海岸名校8天7晚"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 318,
    "left": 436,
    "width": 774,
    "height": 31
   },
   "word": "国会大厦位于华盛顿25米高的国会山上,是美国的心脏建筑。"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 374,
    "left": 434,
    "width": 805,
    "height": 31
   },
   "word": "中央顶楼上的大圆顶上立有一尊6米高的自由女神青铜雕像。"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 431,
    "left": 436,
    "width": 556,
    "height": 31
   },
   "word": "东面的大草坪是历届总统举行就职典礼的地方。"
  }
 }
]
}
```

网络图片文字识别

接口描述

针对网络图片进行专项优化，支持识别艺术字体或背景复杂的文字内容。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668425998119#/api?product=AI&project=文字识别&parent=通用场景OCR&api=rest/2.0/ocr/v1/webimage&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/webimage`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true/false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/webimage?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64
```
~~~

```
'''
网络图片文字识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/webimage"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 网络图片文字识别
*/
public class WebImage {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String webImage() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/webimage";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
```

```
        }
        return null;
    }

    public static void main(String[] args) {
        WebImage.webImage();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/webimage";
static std::string webImage_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    webImage_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 网络图片文字识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int webImage(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = webImage_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
```

```
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/webimage?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class WebImage
    {
        // 网络图片文字识别
        public static string webImage()
        {
```

```
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/webimage?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("网络图片文字识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```

返回说明

返回参数

字段	是否必选	类型	说明
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
```

```
{
  "log_id": 1390650576630448128,
  "words_result_num": 2,
  "words_result": [
    {
      "words": "梦想起航"
    },
    {
      "words": "前往下一个目的地"
    },
    {
      "words": "开始新的旅程"
    }
  ],
}
```


网络图片文字识别（含位置版）

接口描述

支持识别艺术字体或背景复杂的文字内容，除文字信息外，还可返回每行文字的位置信息、行置信度，以及单字符内容和位置等。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668425998119#/api?product=AI&project=文字识别&parent=通用场景OCR&api=rest/2.0/ocr/v1/webimage_loc&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/webimage_loc`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式</br>**优先级

：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>请注意关闭URL防盗链** |
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向
probability	否	string	true/false	是否返回每行识别结果的置信度。默认为false
poly_location	否	string	true/false	是否返回文字所在区域的外接四边形的4个点坐标信息。默认为false
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
##### 网络图片文字识别（含位置版）
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/webimage_loc?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```bash label=Bash
##### encoding:utf-8

import requests
import base64

'''
网络图片文字识别（含位置版）
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/webimage_loc"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 网络图片文字识别（含位置版）
*/
public class WebImageLoc {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String webImageLoc() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/webimage_loc";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        WebImageLoc.webImageLoc();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/webimage_loc";
static std::string webImageLoc_result;
/**
 * curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
 * @param 参数定义见libcurl文档
 * @return 返回值定义见libcurl文档
 */
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    webImageLoc_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
 * 网络图片文字识别（含位置版）
 * @return 调用成功返回0，发生错误返回其他错误码
 */
int webImageLoc(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = webImageLoc_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
 * 发起http post请求(REST API), 并获取REST请求的结果
 * @param string $url
 * @param string $param
 * @return - http response body if succeeds, else false.
 */
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/webimage_loc?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class WebImageLoc
    {
        // 网络图片文字识别（含位置版）
        public static string webImageLoc()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/webimage_loc?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("网络图片文字识别（含位置版）:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。检测到的图像朝向： - -1：未定义； - 0 ：正向； - 1：逆时针旋转90度； - 2：逆时针旋转180度； - 3：逆时针旋转270度
words_result	是	array[]	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	是	string	整行的识别结果
+ location	是	object	整行的矩形框坐标。位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度

+ probability	否	string	当 probability=true 时返回该字段。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ poly_location	否	array[]	当 probability=true 时返回该字段。文字所在区域的外接矩形的4个点坐标信息
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时返回该字段
++ char	否	string	单字符识别结果
++ location	否	object	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示定位定位位置的长方形的高度
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```JSON
{
    "log_id": 1390656223866519552,
    "words_result_num": 3,
    "words_result": [
        {
            "words": "梦想起航",
            "location": {
                "top": 328,
                "left": 1079,
                "width": 56,
                "height": 262
            },
        },
        {
            "words": "前往下一个目的地",
            "location": {
                "top": 329,
                "left": 1160,
                "width": 63,
                "height": 446
            },
        },
        {
            "words": "开始新的旅程",
            "location": {
                "top": 455,
                "left": 1246,
                "width": 63,
                "height": 340
            },
        }
    ],
}
```

手写文字识别

接口描述

支持对各类手写文字进行检测和识别，包括中、英、日、韩、法等 20+ 语言类型，可按单字维度输出置信度、候选字等信息。针对手写作文场景，支持涂改痕迹检测。

在线调试

您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668425998119#/api?product=AI&project=文字识别&parent=通用场景OCR&api=rest/2.0/ocr/v1/handwriting&method=post) 中调试该接口，可进行签名验证、

查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/handwriting`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
recognize_granularity	否	string	big/small	是否定位单字符位置，默认不定位。可选值包括： big：不定位单字符位置； small：定位单字符位置，可返回单字符候选字及置信度
probability	否	string	true/false	是否返回识别结果中每一行的置信度，默认为false，不返回置信度
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: true：检测朝向； false：不检测朝向
detect_alteration	否	string	true/false	是否检测涂改痕迹，适用于手写作文场景，默认不检测，可选值包括： true：检测，涂改痕迹部分用“〓”返回； false：不检测
language_type	否	string|auto_detect</br>CHN_ENG</br>ENG</br>JAP</br>KOR</br>FRE</br>SPA</br>POR</br>GER</br>ITA</br>RUS	识别语言类型，默认为 CHN_ENG。可选值包括：</br> - auto_detect：自动检测语言，并识别</br>- CHN_ENG：中英文混合</br>- ENG：英文</br>- JAP：日语</br>- KOR：韩语</br>- FRE：法语</br>- SPA：西班牙语</br>- POR：葡萄牙语</br>- GER：德语</br>- ITA：意大利语</br>- RUS：俄语</br>- DAN：丹麦语</br>- DUT：荷兰语</br>- MAL：马来语</br>- SWE：瑞典语</br>- IND：印尼语</br>- POL：波兰语</br>- ROM：罗马尼亚语</br>- TUR：土耳其语</br>- GRE：希腊语</br>- HUN：匈牙利语</br>- THA：泰语</br>- VIE：越南语</br>- ARA：阿拉伯语</br>- HIN：印地语</br>

HIN：印地语</br>"

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/handwriting?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
手写文字识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/handwriting"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 手写文字识别
*/
public class Handwriting {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String handwriting() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/handwriting";
```
~~~

```
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        Handwriting.handwriting();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/handwriting";
static std::string handwriting_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    handwriting_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 手写文字识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int handwriting(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
```

```
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = handwriting_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/handwriting?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);
```

```
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class Handwriting
    {
        // 手写文字识别
        public static string handwriting()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/handwriting?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("手写文字识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```
```

返回说明

返回参数
```

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时存返回该字段
++ char	否	string	单字符识别结果
++ candidates	否	array[]	单字符识别结果的候选词内容
+++ word	否	string	单字符识别结果的候选词文字
+++ prob	否	string	单字符识别结果的候选词置信度
++ location	否	object	位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
probability	否	object	当 probability=true 时返回该字段，表示识别结果中每一行的置信度值，包含： - **average**： 行置信度平均值 - **variance**：行置信度方差 - **min**：行置信度最小值
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
      "location": {
        "left": 56,
        "top": 0,
        "width": 21,
        "height": 210
      },
      "words": "3"
    }
  ],
  "words_result_num": 1
}
```


表格文字识别V2

接口描述

支持识别图片/PDF格式文档中的表格内容，返回各表格的表头表尾内容、单元格文字内容及其行列位置信息，全面覆盖各类表格样式，包括常规有线表格、无线表格、含合并单元格表格。同时，支持多表格内容识别。

> 视频教程请参见 表格文字识别V2使用教程

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/table`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
return_excel	否	string	true/false	是否输出excel文件，默认不输出，即：false。可选值包括： - **true**：输出excel，base64编码后输出 - **false**：不输出excel
cell_contents	否	string	true/false	是否输出单元格文字位置信息，可选值包括： - **false**： 默认值，仅输出单元格行列信息及四角点坐标，不输出单元格内文字位置信息 - **true**： 输出单元格内文字的外接四边形四角点坐标，若文字折行，则分行分别输出

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/table?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	string	日志id，用于问题定位
table_num	是	int	检测到的表格数量
tables_result	是	array[]	表格内容
+ table_location	是	array[]	单个表格的四角点x,y坐标
+ header	是	array[]	表头信息
++ location	是	array[]	表头位置，四角点 x,y 坐标
++ words	是	string	表头文字内容，按行拆分
+ body	是	array[]	单元格信息
++ cell_location	是	array[]	单元格四角点x,y坐标
++ row_start	是	int	单元格行起始编号，横线编号从0开始
++ row_end	是	int	单元格行终止编号
++ col_start	是	int	单元格列起始编号，竖线编号从0开始
++ col_end	是	int	单元格列终止编号
++ words	是	string	单元格文字内容
++ contents	否	array[]	单元格内文字内容，分行显示，当请求参数 cell_contents = true 时返回
+++ poly_location	否	array[]	单元格内文字各行的四角点x,y坐标
+++ word	否	string	单元格内分行文字内容
+ footer	是	array[]	表尾信息
++ location	是	array[]	表尾位置，四角点 x,y 坐标
++ words	是	string	表尾信息，按行拆分
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
excel_file	否	string	图像内表格转换为excel文件的base64编码，当 return_excel 参数为true时返回该字段

返回示例


```
{
  "tables_result": [
    {
      "table_location": [
        {
          "x": 67,
          "y": 43
        },
        {
          "x": 708,
          "y": 43
        },
        {
          "x": 708,
          "y": 200
        },
        {
          "x": 67,
          "y": 200
        }
      ],
      "header": [
        {
          "location": [
            {
              "x": 101,
              "y": 16
            },
            {
              "x": 264,
              "y": 16
            },
            {
              "x": 264,
              "y": 34
            },
            {
              "x": 101,
              "y": 34
            }
          ],
          "words": "1.营业收入/营业成本"
        }
      ],
      "body": [
        {
          "cell_location": [
            {
              "x": 68,
              "y": 44
            },
            {
              "x": 188,
              "y": 44
            },
            {
              "x": 188,
              "y": 101
            },
            {
```

```
                "x": 68,
                "y": 101
            }
        ],
        "col_start": 0,
        "row_start": 0,
        "row_end": 2,
        "col_end": 1,
        "words": "项目",
        "contents": [
            {
                "poly_location": [
                    {
                        "x": 84,
                        "y": 60
                    },
                    {
                        "x": 128,
                        "y": 61
                    },
                    {
                        "x": 128,
                        "y": 76
                    },
                    {
                        "x": 84,
                        "y": 75
                    }
                ],
                "word": "项目"
            }
        ]
    },
    {
        "cell_location": [
            {
                "x": 192,
                "y": 43
            },
            {
                "x": 442,
                "y": 43
            },
            {
                "x": 442,
                "y": 68
            },
            {
                "x": 192,
                "y": 68
            }
        ],
        "col_start": 1,
        "row_start": 0,
        "row_end": 1,
        "col_end": 3,
        "words": "本期数",
        "contents": [
            {
                "poly_location": [
                    {
                        "x": 308,
```

```
                        "y": 49
                    },
                    {
                        "x": 349,
                        "y": 49
                    },
                    {
                        "x": 348,
                        "y": 63
                    },
                    {
                        "x": 307,
                        "y": 63
                    }
                ],
                "word": "本期数"
            }
        ]
    }
],
"footer": []
}
],
"table_num": 1,
"log_id": 1516052468533474289
}
```


印章识别

接口描述

检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，**支持识别印章编码**，可覆盖圆形章、椭圆形章、方形章等常见种类的印章。

印章编码示例图片：



① 印章辅助防伪纹线

② 印章编码

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/seal`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
return_image	否	string	true/false	是否返回印章切片图片，默认不返回，可选值包括： true：返回印章的 base64 编码信息； false：不返回
flatten_image	否	string	true/false	是否返回印章展平图片，默认不返回（仅支持对圆章、椭圆章进行展平，其他印章不展平），可选值包括： true：返回印章展平后的 base64 编码信息； false：不返回

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/seal?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
result_num	是	uint32	识别结果数，表示results的元素个数
result	是	array[]	定位结果数组
+ seal_image	否	string	印章切图的 base64 编码，return_image=true 时返回
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ probability	是	float	每一个识别结果的置信度值
+ type	是	string	印章的类别，共有circle（圆章），ellipse（椭圆章），rectangle（方章）三种
+ major	是	object	主字段内容，即章内上环弯曲文字
++ words	是	string	主字段识别内容
++ flatten_image	否	string	主字段展平图的 base64 编码，即章内上环弯曲文字切片后拼接图片，flatten_image=true 时返回
++ probability	是	float	主字段识别内容的置信度
+ minor	是	array[]	其他字段内容，即除主字段外的文字识别内容均放置于该参数中返回，**数字编码也在该字段返回**。若章内不存在其他字段文字，则该参数为空
++ words	是	string	其他字段识别内容
++ flatten_image	否	string	其他字段展平图的 base64 编码，即印章其他文字切片后拼接图片，flatten_image=true 时返回
++ probability	是	float	其他字段识别内容的置信度
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
"result": [
 {
  "major": {
   "probability": 0.99759155511856,
   "words": "峨眉山旅游股份有限公司成都峨眉山雪芽大酒店分公司"
  },
  "minor": [
   {
    "probability": 0.99994027614594,
    "words": "前厅部"
   }
  ],
  "probability": 0.9936261177063,
  "location": {
   "top": 594,
   "left": 918,
   "width": 150,
   "height": 142
  },
  "type": "circle"
 }
],
"log_id": "1349006147834609664",
"result_num": 1
}
```


数字识别

接口描述

对图片中的数字进行提取和识别，自动过滤非数字内容，仅返回数字内容及其位置信息，识别准确率超过99%。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/numbers`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。 可选值包括 - true：检测朝向； - false：不检测朝向

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/numbers?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时返回该字段
++ char	否	string	单字符识别结果
++ location	否	object	位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
      "location": {
        "left": 56,
        "top": 0,
        "width": 21,
        "height": 210
      },
      "words": "3"
    }
  ],
  "words_result_num": 1
}
```


二维码识别

接口描述

对图片中的二维码、条形码进行检测和识别，返回存储的文字信息及其位置信息

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/qrcode`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
location	否	string	true/false	是否输出二维码/条形码位置信息 - **false**：默认值，不返回位置信息； - **true**：返回图中二维码/条形码的位置信息，包括上边距、左边距、宽度、高度

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/qrcode?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
codes_result_num	是	uint32	识别结果数，表示codes_result的元素个数
codes_result	是	array[]	定位和识别结果数组
+ type	是	string	识别码类型条码类型包括：9种条形码（UPC_A、UPC_E、EAN_13、EAN_8、CODE_39、CODE_93、CODE_128、ITF、CODABAR），4种二维码（QR_CODE、DATA_MATRIX、AZTEC、PDF_417）
+ text	是	array[]	条形码/二维码识别内容,目前仅支持输出中英文结果
+location	否	object{}	条形码/二维码位置信息，包括上边距、左边距、宽度、高度，当请求参数 location = true 时返回
++ top	否	unit32	条形码/二维码的上边距
++ left	否	unit32	条形码/二维码的左边距
++ width	否	unit32	条形码/二维码的宽度
++ height	否	unit32	条形码/二维码的高度

返回示例

示例1（单个二维码的情况）：

```
{
    "codes_result": [
        {
            "text": [
                "百度大脑"
            ],
            "type": "QR_CODE"
        }
    ],
```

```
    "codes_result_num": 1,
    "log_id": 1516059338033334646
}
```


示例2（多个二维码/条码不含位置的情况）：


```
{
    "log_id": 1508509437,
    "codes_result": [
        {
            "type": "QR_CODE",
            "text": [
                "HTTP://Q8R.HK/YELZ0"
            ]
        },
        {
            "type": "PDF_417",
            "text": [
                "PDF417倢丄TL-30偱撉傒庢傝壜擻偱偡丅"
            ]
        },
        {
            "type": "CODABAR",
            "text": [
                "000800"
            ]
        },
        {
            "type": "CODE_39",
            "text": [
                "1234567890"
            ]
        },
        {
            "type": "AZTEC",
            "text": [
                "www.tec-it.com"
            ]
        },
        {
            "type": "DATA_MATRIX",
            "text": [
                "Wikipedia, the free encyclopedia"
            ]
        },
        {
            "type": "CODE_93",
            "text": [
                "123456789"
            ]
        },
        {
            "type": "CODE_128",
            "text": [
                "50090500019191"
            ]
        },
        {
            "type": "EAN_8",
            "text": [
                "12345670"
            ]
        },
        {
```

```
        {
          "type": "EAN_13",
          "text": [
            "6901234567892"
          ]
        },
        {
          "type": "UPC_E",
          "text": [
            "01234565"
          ]
        }
    ],
    "codes_result_num": 11
}
```


智能结构化

接口描述

支持智能提取图片中的字段结构化信息，无需训练灵活提取。适用于各类证照、票据、表单等版式中的结构化信息录入场景。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/smart_struct`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码，大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码，大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
return_relation	否	string	true/false	是否返回结构化对应关系及单文本行结果，默认为 false，即不返回，为 true 时返回，针对relations、line_info 2个数组

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/smart_struct?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必	类型	说明

字段	选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
object_id_num	否	uint32	文本行结果数，表示 object_id 的个数，当 if_relation=true 时返回
words_result	是	object{}	识别结果数组
+ struct_info	是	object{}	**非表格区的 k-v 按结构化对应展示信息**
++ group	是	array[]	key 文字行的信息
+++ key	是	array[]	key 文字行的信息
++++ word	是	string	key 文字行的文字结果
++++ line__probability	是	string	key 文字行的文字结果识别置信度
++++ line_location	是	object{}	key 文字行的位置
+++++ left	是	uint32	表示 key 文字行定位位置的长方形左上顶点的水平坐标
+++++ top	是	uint32	表示 key 文字行定位位置的长方形左上顶点的垂直坐标
+++++ width	是	uint32	表示 key 文字行定位位置的长方形的宽度
+++++ height	是	uint32	表示 key 文字行定位位置的长方形的高度
+++ value	是	array[]	value 文字行的信息
++++ word	是	string	value 文字行的文字结果
++++ line__probability	是	string	value 文字行的文字结果识别置信度
++++ line_location	是	object{}	value 文字行的位置
+++++ left	是	uint32	表示 value 文字行定位位置的长方形左上顶点的水平坐标
+++++ top	是	uint32	表示 value 文字行定位位置的长方形左上顶点的垂直坐标
+++++ width	是	uint32	表示 value 文字行定位位置的长方形的宽度
+++++ height	是	uint32	表示 value 文字行定位位置的长方形的高度
+ relations	否	object{}	**结构化对应关系，当 if_relation=true 时返回，以下字段均返回**
++ kv_relations	否	array[]	**非表格区的 k-v 结构化关系**，可支持一对一、一对多的关系
+++			

+++ root_node	否	uint32	根节点的 object_id，即 k-v 区的 key
+++ leaf_nodes	否	array[]	由根节点指向叶子节点的 object_id，及 k-v 区的 values
++ table_relations	否	object{}	**表格区的结构关系**
+++ kk_relations	否	array[]	**表格区的 k-k 结构关系**，即多级表头的结构关系，可支持一对一、一对多的关系
++++ root_node	否	uint32	根节点的 object_id，即表格/表单区 k-k 关系的【前者 key】
++++ leaf_nodes	否	array[]	由【前者 key】根节点指向叶子节点【后者 key】的object_id
+++ kv_relations	否	array[]	**表格区的k-v结构关系**，可支持一对一、一对多的关系
++++ root_node	否	uint32	根节点的 object_id，即表格区 k-v 关系的 key
++++ leaf_nodes	否	array[]	由根节点指向叶子节点的 object_id，即表格/表单区k-v关系的values
+++ vv_relations	否	array[]	**表格区的v-v结构关系**，可支持一对一、一对多的关系
++++ root_node	否	uint32	根节点的 object_id，即表格区v-v关系的【前者 value】
++++ leaf_nodes	否	array[]	由【前者 value】根节点指向叶子节点【后者 value】的 object_id
+ line_info	否	object{}	**文字行的识别结果、类别、置信度、位置信息等，当 if_relation=true 时返回，以下字段均返回**
++ object_id	否	uint	文字行的 id，唯一标识，按从上到下从左到右顺序，依次顺位排列
++ block_id	否	uint	换行文字的 id，另一唯一标识，属于同一个词义的 n 个文字行的 block_id一致，按从上到下从左到右顺序，依次顺位排列。**说明：**block 只针非表格区 key、表格区 key、表格区 value。非换行的 block 元素固定返回 -1
++ word	否	string	文字行的文字结果
++ line_class	否	string	文字行的类别，key 表示非表格区的 key 值，value 表示非表格区的 value 值，table_value 表格区的 value 值，other 表示无结构化关系的文本行
++ line_class_probability	否	string	line_class 的分类置信度
++ line__probability	否	string	文字行的文字结果识别置信度
++			

line_location	否	object{}	文字行的位置
+++ left	否	uint32	表示文字行定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示文字行定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示文字行定位位置的长方形的宽度
+++ height	否	uint32	表示文字行定位位置的长方形的高度

返回示例


```
{
  "words_result": {
    "struct_info": {
      "group": [
        {
          "key": [
            {
              "word": "MORITA"
            }
          ],
          "value": [
            {
              "word": "SHUNSUKE"
            }
          ]
        },
        {
          "key": [
            {
              "word": "出生日期"
            }
          ],
          "value": [
            {
              "word": "Date of Birth"
            }
          ]
        },
        {
          "key": [
            {
              "word": "性别/Sex"
            }
          ],
          "value": [
            {
              "word": "男/M"
            }
          ]
        },
        {
          "key": [
            {
              "word": "国籍/Nationality"
            }
          ],
          "value": [
            {
              "word": "日本/JPN"
            }
          ]
```

```
            }
          ]
        }
      },
    "log_id": 1768260983737031188,
}
```

文档解析

接口描述

文档解析支持对doc、pdf、图片、xlsx等18种格式文档进行解析，输出文档的版面、表格、阅读顺序、标题层级、旋转角度等信息，支持中、英、日、韩、法等20余种语言类型，可返回Markdown格式内容，将非结构化数据转化为易于处理的结构化数据，识别准确率可达 90% 以上。

文档解析API服务为**异步接口**，需要先调用**提交请求接口**获取 task_id，然后调用**获取结果接口**进行结果轮询，建议提交请求后 5～10 秒轮询。**提交请求接口**QPS为2，**获取结果接口**QPS为10。

提交请求接口

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/brain/online/v2/parser/task`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
file_data	和file_url二选一	string	-	文件的base64编码数据：</br>-版式文档：pdf、jpg、jpeg、png、bmp、tif、tiff、ofd、ppt、pptx</br>-流式文档：doc、docx、txt、xls、xlsx、wps、html、mhtml</br>PDF文档大小不超过300M，非PDF文档大小不超过50M，文档页数不超过2000页（流式文档按2000字算一页）</br>**优先级：** file_data > file_url，当file_data字段存在时，file_url字段失效
file_url	和file_data二选一	string	-	文件数据URL，URL长度不超过1024字节，支持单个URL传入，若文件大小超过50M，须通过该方式上传。其余文件准入标准与file_data一致。</br>**优先级：** file_data > file_url，当file_data字段存在时，file_url字段失效</br>**请注意关闭URL防盗链**
file_name	是	string	-	文件名，请保证文件名后缀正确，例如 "1.pdf " |</br>
return_doc_chunks	否	string	-	JSON 字符串，文档切分的传参结构，Python中，json字符串示例 json.dumps({"switch": True, "chunk_size": -1 })，详细结构详见下方说明，具体调用方式详见下方请求示例 |</br>
+ switch	否	bool	True/False	是否进行文档内容切分，「default=False」 |</br>
+ chunk_size	否	int	(0,∞)	切分块的大小，按照字符数统计。若段落字符数小于或等于chunk_size，则按chunk_size切分；若大于chunk_size，则整段作为一个切分块，不因chunk_size限制而额外切分。</br>「default=-1」，表示不限制切分块的大小，按照文档的段落进行切分 |</br>

| + split_type | 否 | string | chunk/mark | 切分方式。</br> chunk:按照字符数「chunk_size」切分; </br> mark:按照「separators」中的标点符号切分; </br>「default=chunk」|</br>
| + separators | 否 | list | 。/ ；/ ！/ ？/ ; / ! / ? | 切分标点，仅包含表示一句话结束的标点。</br>「default=["。", "；", "！", "？", ";", "!", "?"]」 |</br>
| recognize_formula | 否 | bool | True/False | 是否对版式类型文档进行公式识别 |</br>
analysis_chart	否	bool	True/False	是否对统计图表进行解析
language_type	否	string	-	识别语种类型，默认为 CHN_ENG，可选值如下： </br>-CHN_ENG：中英文 </br>- JAP：日语</br>- KOR：韩语</br>- FRE：法语</br>- SPA：西班牙语</br>- POR：葡萄牙语</br>- GER：德语</br>- ITA：意大利语</br>- RUS：俄语</br>- DAN：丹麦语</br>- DUT：荷兰语</br>- MAL：马来语</br>- SWE：瑞典语</br>- IND：印尼语</br>- POL：波兰语</br>- ROM：罗马尼亚语</br>- TUR：土耳其语</br>- GRE：希腊语</br>- HUN：匈牙利语</br>- THA：泰语</br>- VIE：越南语</br>- ARA：阿拉伯语</br>- HIN：印地语
switch_digital_width	否	string	-	是否对符号进行全半角转换，默认为 auto，可选值如下： </br>-auto：不转换，按模型识别结果输出 </br>- half：将所有的符号转换为半角输出</br>- full：将所有的符号转换为全角输出

请求代码示例

提示：使用示例代码前，请记得替换其中的示例Token、文档地址或Base64信息。

~~~codeset
```python label=Python
import requests
import os
import base64


def create_task(url, file_path, file_url):
    """
    Args:
        url: string, 服务请求链接
        file_path: 本地文件路径
        file_url: 文件链接
    Returns: 响应
    """
    # 文件请求
    with open(file_path, "rb") as f:
        file_data = base64.b64encode(f.read())
    data = {
        "file_data": file_data,
        "file_url": file_url,
        "file_name": os.path.basename(file_path)
    }

    # 文档切分参数，非必传
    # return_doc_chunks = json.dumps({"switch": True, "chunk_size": -1})
    # data["return_doc_chunks"] = return_doc_chunks

    headers = {'Content-Type': 'application/x-www-form-urlencoded'}

    response = requests.post(url, headers=headers, data=data)
    return response

if __name__ == '__main__':
    request_host = "https://aip.baidubce.com/rest/2.0/brain/online/v2/parser/task?" \
                   "access_token={token}"
    file_path = "./test.pdf"
    response = create_task(request_host, file_path, "")
    print(response.json())
```
~~~

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ task_id	string	该请求生成的task_id，后续使用该task_id获取审查结果

返回示例

成功返回示例：

```JSON
{
    "error_code": 0,
    "error_msg": "",
    "log_id": "101385981311373626852735056654 33",
    "result": {
        "task_id": "task-3zy9Bg8CHt1M4pPOcX2q5bg28j26801S"
    }
}
```

失败返回示例（详细的错误码说明见[API文档-错误码](https://ai.baidu.com/ai-doc/OCR/wlqc0cfdq)）：

```JSON
{
    "error_code": 282003,
    "error_msg": "missing parameters",
    "log_id": "3750763103358554450798325392414 1",
    "result": "null"
}
```

获取结果接口

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/brain/online/v2/parser/task/query`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
task_id	是	string	发送提交请求时返回的task_id

请求代码示例

提示：使用示例代码前，请记得替换其中的示例Token、task_id。

~~~codeset
```python label=Python

import requests

def query_task(url, task_id):
    """
    Args:
        url: string, 请求链接
        task_id: string, task id
    Returns: 响应
    """
    data = {
        "task_id": task_id
    }
    headers = {'Content-Type': 'application/x-www-form-urlencoded'}
    print(url)
    response = requests.post(url, headers=headers, data=data)
    return response


if __name__ == '__main__':
    # 需要替换为实际的任务id
    task_id = "task_id"
    # {access_token} 需要替换为实际调用鉴权接口获取的access_token
    request_host = "https://aip.baidubce.com/rest/2.0/brain/online/v2/parser/task/query?access_token=
{access_token}"
    resp = query_task(request_host, task_id)
    print(resp.json())

```
~~~

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ task_id	string	任务ID
+ status	string	任务状态，pending：排队中；processing：运行中；success：成功；failed：失败
+ task_error	string	解析报错信息，包含任务失败、额度不够
+ markdown_url	string	文档解析结果的markdown格式链接，**链接有效期30天**
+ parse_result_url	string	文档解析结果的bos链接，**链接有效期30天**

可通过parse_result_url下载解析结果的JSON文件，parse_result_url的返回参数如下：

字段	类型	说明
file_name	string	文档名称
file_id	string	文档ID
+ pages	list	文件单页解析内容
++ page_id	string	页码ID
++ page_num	int	页码数
++ text	string	当前页的所有纯文字内容
++ layouts	list	页面内容版式分析的结果
+++ layout_id	string	layout ID，layout元素唯一标志，以"xxxxx-layout-{global_layout_index}"形式，global_layout_index为layout元素整个文档的全局索引
+++ text	string	layout对应的文本内容。注：当type为table, image时该字段为空, 需要根据type和layout_id分别到tables, images字段里找到对应的内容
+++ position	list	layout元素在页面中的位置，[x, y, w, h] box框，左上角和宽高
+++ type	string	layout元素类型, 当前可取值： • para：段落 • table：表格 • head_tail：页面顶部 • image：文档中的插图 • contents：目录 • seal：印章 • title：标题 • formula：公式
+++ sub_type	string	layout元素子类型, 当type为title、image时，subtype有值。 title类的 subtype 包含： • title_{n}，代表n级标题, 比如title_2代表二级标题 • image_title：图标题 • table_title：表标题 image类的 subtype 包含： • chart：统计图表 • figure：普通插图 • QR_code：二维码 • Bar_code：条形码

+++ parent	string	标题层级树中父节点的layout ID，若当前layout为一级标题，其parent为 "root"。在table和image的内嵌版面信息中暂时都为空
+++ children	list	标题层级树中子节点的layout ID。在table和image的内嵌版面信息中暂时都为空
++ tables	list	页面表格解析结果
+++ layout_id	string	layout ID，与layouts中的元素type为table的元素的layout ID对应
+++ markdown	string	表格内容的markdown形式
+++ table_title_id	list	表格标题对应的layout_id，默认为null
+++ position	list	边框数据 「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点），版式格式时有效
+++ cells	list	单元格的内嵌版面信息，layout类型为表格时有值
+++ matrix	list	二位数组 表示表格内布局位置信息，每个元素对应cells列表中元素的索引
+++ merge_table	string	「begin」- 跨页表格开始、「inner」- 跨页表格中间表格(表格跨页超过两页)、「end」- 跨页表格结束；非跨页表格该字段为空
++ images	list	页面中图片解析结果
+++ layout_id	string	layout ID，与layouts中的元素type为image的元素的layout ID对应
+++ image_title_id	list	图片标题对应的layout_id，默认为null
+++ position	list	边框数据 「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点），版式格式时有效
+++ content_layouts	list	图片的内嵌版面信息
+++ data_url	string	图片存储链接
+++ image_description	string	对统计图表进行内容解析和描述，输出结果为json字符串，可通过json.loads结构化为json格式
++ meta	dict	页面元信息
+++ page_width	int	页面宽度
+++ page_height	int	页面高度
+++ is_scan	bool	是否扫描件
+++ page_angle	int	页面倾斜角度
+++ page_type	string	页面属性「text」- 正文、「contents」- 目录、「appendix」- 附录、「others」- 其他
+++ sheet_name	string	excel的sheet名
+ chunks	list	文件内容切分结果，return_doc_chunks中switch为True时有值
++ chunk_id	string	切片的ID
++ content	string	切片的内容
++ type	string	切片类型, 为text或者table
++ meta	dict	chunk元信息
+++ title	list	chunk所属的多级标题内容
+++ position	list	chunk的位置，根据分块算法有可能chunk跨多个页

+++ box	list	chunk的位置坐标
+++ page_num	int	chunk内容所在页数

表格解析结构说明

以下图为例：




```
{
##### cells列表包含14个元素，matrix中的每个数字表示一个单元格在cells列表中的索引。
  "cells": [
    {"layout_id": "layout-xxxx",
     "position": [90, 376, 21, 10],
     "text": "序号"
     ...
    }, # ... 其他单元格信息
  ],
  "matrix": [
    [0, 1, 2],
    [3, 4, 5],
    [6, 7, 8],
    [9, 10, 11],
    [12, 12, 13],
    [12, 12, 14]

}
```

返回示例

成功返回示例：

```
{
  "log_id": "2359659789928692176157936558237 3",
  "error_code": 0,
  "error_msg": "",
  "result":
  {
    "task_id": "task-UnvGsgbYZp9pS3BZRHn11ifzjNvKzTgf",
    "status": "success",
    "task_error": null,
    "duration": 902.0,
    "parse_result_url": "https:xxxxxxxxxxxxxxxxxx"
  }
}
```

解析结果示例：

```
{
  "file_name": "示例文件1（文字+表格）更新-改-1.pdf"
```

```
    "file_name": "示例文件1（文字+表格）更新版-1.pdf",
    "file_id": "file-u9kVDu6dtwMyNrizbejMIF8A852aJLm2",
    "pages": [
      {
        "page_id": "2aJLm2-page-0",
        "page_num": 0,
        "text": "买卖合同\n甲、乙双方根据《中华人民共和国合同法》及其它相关法律、法规的规定，本着平等、自愿、互利的原则，经友好协商，订立本合同，以资共同信守：\n1.合同标的物信息\n| 序号 | 商品名称 | 产品简称 | 单价 | 数量 | 总价 | 税率 | 备注 |\n| --- | --- | --- | --- | --- | --- | --- | --- |\n| 1 | 软件-AI | 中台内网-推理平台+训练平台 | 95,000 | 1 | 905,000 | 13% | 第一单元 |\n| 2 | 硬件【A】 | 服务器 | 30,250 | 37 | 1,11950 | 13% | 第一单元 |\n\n本合同一式【2】份，经双方代表签字盖章生效。甲乙双方各执【3】份，具有同等法律效力。\n1 \n",
        "layouts": [
          {
            "layout_id": "2aJLm2-layout-1",
            "text": "买卖合同",
            "position": [
              263,
              109,
              103,
              28
            ],
            "type": "title",
            "sub_type": "title_1",
            "parent": "root",
            "children": [
              "2aJLm2-layout-2",
              "2aJLm2-layout-3"
            ]
          },
          {
            "layout_id": "2aJLm2-layout-2",
            "text": "甲、乙双方根据《中华人民共和国合同法》及其它相关法律、法规的规定，本着平等、自愿、互利的原则，经友好协商，订立本合同，以资共同信守：",
            "position": [
              84,
              160,
              444,
              31
            ],
            "type": "text",
            "sub_type": "",
            "parent": "2aJLm2-layout-1",
            "children": [

            ]
          },
          {
            "layout_id": "2aJLm2-layout-3",
            "text": "1.合同标的物信息",
            "position": [
              79,
              206,
              110,
              14
            ],
            "type": "title",
            "sub_type": "title_2",
            "parent": "2aJLm2-layout-1",
            "children": [
              "2aJLm2-layout-4",
              "2aJLm2-layout-5",
              "2aJLm2-layout-6"
            ]
```

```
            },
            {
                "layout_id": "2aJLm2-layout-4",
                "text": "",
                "position": [
                    82,
                    224,
                    452,
                    97
                ],
                "type": "table",
                "sub_type": "",
                "parent": "2aJLm2-layout-3",
                "children": [

                ]
            },
            {
                "layout_id": "2aJLm2-layout-5",
                "text": "本合同一式【2】份，经双方代表签字盖章生效。甲乙双方各执【3】份，具有同等法律效力。",
                "position": [
                    79,
                    348,
                    456,
                    31
                ],
                "type": "text",
                "sub_type": "",
                "parent": "2aJLm2-layout-3",
                "children": [

                ]
            },
            {
                "layout_id": "2aJLm2-layout-6",
                "text": "1",
                "position": [
                    305,
                    745,
                    11,
                    12
                ],
                "type": "head_tail",
                "sub_type": "",
                "parent": "2aJLm2-layout-3",
                "children": [

                ]
            }
        ],
        "tables": [
            {
                "layout_id": "2aJLm2-layout-4",
                "markdown": "| 序号 | 商品名称 | 产品简称 | 单价 | 数量 | 总价 | 税率 | 备注 |\n| --- | --- | --- | --- | --- | --- | --- | --- |\n| 1 | 软件-AI | 中台内网-推理平台+训练平台 | 95,000 | 1 | 905,000 | 13% | 第一单元 |\n| 2 | 硬件【A】 | 服务器 | 30,250 | 37 | 1,11950 | 13% | 第一单元 |\n",
                "position": [
                    82,
                    224,
                    452,
                    97
                ],
```

```
            "cells": [
                {
                    "layout_id": "2aJLm2-layout-4-0",
                    "text": "序号",
                    "position": [
                        82,
                        224,
                        36,
                        28
                    ],
                    "type": "text",
                    "sub_type": "",
                    "parent": "",
                    "children": null
                },
                {
                    "layout_id": "2aJLm2-layout-4-1",
                    "text": "商品名称",
                    "position": [
                        118,
                        224,
                        78,
                        28
                    ],
                    "type": "text",
                    "sub_type": "",
                    "parent": "",
                    "children": null
                },
                {
                    "layout_id": "2aJLm2-layout-4-2",
                    "text": "产品简称",
                    "position": [
                        196,
                        224,
                        92,
                        28
                    ],
                    "type": "text",
                    "sub_type": "",
                    "parent": "",
                    "children": null
                },
                {
                    "layout_id": "2aJLm2-layout-4-3",
                    "text": "单价",
                    "position": [
                        288,
                        224,
                        57,
                        28
                    ],
                    "type": "text",
                    "sub_type": "",
                    "parent": "",
                    "children": null
                },
                {
                    "layout_id": "2aJLm2-layout-4-4",
                    "text": "数量",
                    "position": [
                        345,
                        224
```

```
            224,
            43,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-5",
        "text": "总价",
        "position": [
            387,
            224,
            61,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-6",
        "text": "税率",
        "position": [
            448,
            224,
            46,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-7",
        "text": "备注",
        "position": [
            494,
            224,
            41,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-8",
        "text": "1",
        "position": [
            82,
            252,
            36,
            44
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
```

```
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-9",
            "text": "软件-AI ",
            "position": [
                118,
                252,
                78,
                44
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-10",
            "text": "中台内网-推理平台+训练平台",
            "position": [
                196,
                252,
                92,
                44
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-11",
            "text": "95,000",
            "position": [
                288,
                252,
                57,
                44
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-12",
            "text": "1",
            "position": [
                345,
                252,
                43,
                44
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-13",
            "text": "905,000",
            "position": [
```

```
            387,
            252,
            61,
            44
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-14",
        "text": "13% ",
        "position": [
            448,
            252,
            46,
            44
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-15",
        "text": "第一单元",
        "position": [
            494,
            252,
            41,
            71
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-16",
        "text": "2",
        "position": [
            82,
            295,
            36,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-17",
        "text": "硬件【A】",
        "position": [
            118,
            295,
            78,
            28
        ],
        "type": "text",
```

```
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-18",
            "text": "服务器",
            "position": [
                196,
                295,
                92,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-19",
            "text": "30,250",
            "position": [
                288,
                295,
                57,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-20",
            "text": "37",
            "position": [
                345,
                295,
                43,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-21",
            "text": "1,11950",
            "position": [
                387,
                295,
                61,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-22",
            "text": "13% "
```

```
                "text": "15%",
                "position": [
                  448,
                  295,
                  46,
                  28
                ],
                "type": "text",
                "sub_type": "",
                "parent": "",
                "children": null
              }
            ],
            "matrix": [
              [
                0,
                1,
                2,
                3,
                4,
                5,
                6,
                7
              ],
              [
                8,
                9,
                10,
                11,
                12,
                13,
                14,
                15
              ],
              [
                16,
                17,
                18,
                19,
                20,
                21,
                22,
                15
              ]
            ],
            "merge_table": ""
          }
        ],
        "images": [
          {
            "layout_id": "Kr9RM7-layout-10",
            "image_title_id": null,
            "position":
            [
              90,
              549,
              422,
              221
            ],
            "data_url": " ",
            "image_description": " {\n    \"title\": \"None\",\n    \"source\": \"None\",\n    \"x_title\": \"图2 统计图\",\n    \"y_title\": [\n        \"None\",\n        \"None\"\n    ],\n    \"values\": {\n        \"轻客出口（万辆）\": {\n\"2017\": \"2.0\",\n            \"2018\": \"1.9\",\n            \"2019\": \"2.0\",\n            \"2020\": \"1.2\",\n
```

```
\"2021\": \"1.1\",\n          \"2022\": \"1.4\",\n          \"2023Q1\": \"0.4\"\n      },\n      \"中客出口（万辆）\": {\n          \"2017\": \"0.4\",\n          \"2018\": \"0.2\",\n          \"2019\": \"0.4\",\n          \"2020\": \"0.4\",\n          \"2021\": \"0.3\",\n          \"2022\": \"0.6\",\n          \"2023Q1\": \"0.0\"\n      },\n      \"大客出口（万辆）\": {\n          \"2017\": \"3.3\",\n          \"2018\": \"3.6\",\n          \"2019\": \"4.1\",\n          \"2020\": \"2.5\",\n          \"2021\": \"2.3\",\n          \"2022\": \"2.8\",\n          \"2023Q1\": \"0.7\"\n      },\n      \"客车出口占客车总销量的比例（%）\": {\n          \"2017\": \"10.8\",\n          \"2018\": \"11.6\",\n          \"2019\": \"13.6\",\n          \"2020\": \"9.1\",\n          \"2021\": \"7.3\",\n          \"2022\": \"11.8\",\n          \"2023Q1\": \"12.0\"\n      }\n  }\n}"
            }
          ],
          "meta": {
            "page_width": 612,
            "page_height": 792,
            "is_scan": false,
            "page_angle": 0,
            "page_type": "text",
            "sheet_name": ""
          }
        ],
        "chunks": [
          {
            "chunk_id": "2aJLm2-chunk-0",
            "content": "甲、乙双方根据《中华人民共和国合同法》及其它相关法律、法规的规定，本着平等、自愿、互利的原则，经友好协商，订立本合同，以资共同信守：",
            "type": "text",
            "meta": {
              "title": [
                "买卖合同"
              ],
              "position": [
                {
                  "box": [
                    84,
                    160,
                    444,
                    31
                  ],
                  "page_num": 0
                }
              ]
            }
          },
          {
            "chunk_id": "2aJLm2-chunk-1",
            "content": "| 序号 | 商品名称 | 产品简称 | 单价 | 数量 | 总价 | 税率 | 备注 |\n| --- | --- | --- | --- | --- | --- | --- | --- |\n| 1 | 软件-AI | 中台内网-推理平台+训练平台 | 95,000 | 1 | 905,000 | 13% | 第一单元 |\n| 2 | 硬件【A】 | 服务器 | 30,250 | 37 | 1,11950 | 13% | 第一单元 |\n",
            "type": "table",
            "meta": {
              "title": [
                "买卖合同",
                "1.合同标的物信息"
              ],
              "position": [
                {
                  "box": [
                    82,
                    224,
                    452,
                    97
                  ],
                  "page_num": 0
```

```
            }
          ]
        }
      },
      {
        "chunk_id": "2aJLm2-chunk-2",
        "content": "本合同一式【2】份，经双方代表签字盖章生效。甲乙双方各执【3】份，具有同等法律效力。",
        "type": "text",
        "meta": {
          "title": [
            "买卖合同",
            "1.合同标的物信息"
          ],
          "position": [
            {
              "box": [
                79,
                348,
                456,
                31
              ],
              "page_num": 0
            }
          ]
        }
      }
    ]
}
```


失败返回示例（详细的错误码说明见API文档-错误码）：


```
{"log_id": "13665091038742503867108513247608",
"error_code": "282007",
"error_msg": "task not exist, please check task id",
"result": "null"}
```


卡证文字识别

身份证识别

接口描述

支持对二代居民身份证正反面所有8个字段进行结构化识别，包括姓名、性别、民族、出生日期、住址、身份证号、签发机关、有效期限，识别准确率超过99%；同时支持身份证正面头像检测，并返回头像切片的base64编码及位置信息。

同时，支持对用户上传的身份证图片进行图像质量和风险检测，是否存在正反颠倒、模糊、欠曝、过曝等质量问题，可识别图片是否为复印件或临时身份证，是否被翻拍或编辑，是否存在四角不完整、头像或关键字段被遮挡。

增值能力	详情
裁剪能力	头像检测与切片：返回头像切片的base64编码及位置信息
	身份证检测与切片：返回身份证的base64编码及位置信息（去掉证件外围多余背景、自动矫正拍摄角度）
质量检测	身份证图片模糊检测
	身份证关键字段反光或过曝光
	身份证图片较暗或欠曝光
	身份证边框/四角不完整告警
	身份证头像或关键字段被遮挡/马赛克告警
风险检测	身份证复印件告警
	临时身份证告警
	身份证翻拍告警
	身份证PS编辑告警
	身份证证号不合法告警
	身份证证号和出生日期、性别信息不一致告警

> 视频教程请参见 身份证识别API调用教程（视频版）

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/idcard`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
id_card_side	是	string	front/back	-front：身份证含照片的一面 -back：身份证带国徽的一面 自动检测身份证正反面，如果传参指定方向与图片相反，支持正常识别，返回参数image_status字段为"reversed_side"
detect_ps	否	string	true/false	是否检测上传的身份证被PS，默认不检测。可选值： -true：检测 - false：不检测
detect_risk	否	string	true/false	是否开启身份证风险类型（身份证复印件、临时身份证、身份证翻拍、修改过的身份证）检测功能，默认不开启，即：false。 - true：开启，请查看返回参数risk_type； - false：不开启
detect_quality	否	string	true/false	是否开启身份证质量类型（清晰模糊、边框/四角不完整、头像或关键字段被遮挡/马赛克）检测功能，默认不开启，即：false。 - true：开启，请查看返回参数card_quality； - false：不开启
detect_photo	否	string	true/false	是否检测头像内容，默认不检测。可选值：true-检测头像并返回头像的 base64 编码及位置信息
detect_card	否	string	true/false	是否检测身份证进行裁剪，默认不检测。可选值：true-检测身份证并返回证照的 base64 编码及位置信息
detect_direction	否	string	true/false	是否检测上传的身份证图片方向，默认不检测。可选值： -true：检测 - false：不检测

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/idcard?access_token=【调用鉴权接口获取的token】' --data 'id_card_side=front&image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	object	定位和识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
direction	否	int32	图像方向，输入参数 detect_direction= true 时返回。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
image_status	是	string	normal-识别正常 reversed_side-身份证正反面颠倒 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 other_type_card-其他类型证照 over_exposure-身份证关键字段反光或过曝 over_dark-身份证欠曝（亮度过低） unknown-未知状态
card_ps	否	string	输入参数 detect_ps = true 时，则返回该字段，判断身份证是否被PS，返回值： - 0：正常， - 1：PS， - -1：无效
risk_type	否	string	输入参数 detect_risk = true 时，则返回该字段识别身份证**风险类型**: normal-正常身份证； copy-复印件； temporary-临时身份证；

			screen-翻拍； unknown-其他未知情况
edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
card_quality	否	object	输入参数 detect_quality = true 时，则返回该字段识别身份证质量类型
+ IsClear	否	float	质量类型，是否清晰，返回值包括： - 1：清晰 - 0：不清晰
+ IsClear_propobility	否	float	“是否清晰”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 IsClear_propobility 超过0.5时，对应 IsClear 返回1，低于0.5，则返回0
+ IsComplete	否	float	质量类型，是否边框/四角完整，返回值包括： - 1：边框/四角完整 - 0：边框/四角不完整
+ IsComplete_propobility	否	float	“是否边框/四角完整”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 IsComplete_propobility 超过0.5时，对应 IsComplete 返回1，低于0.5，则返回0
+ IsNoCover	否	float	质量类型，是否头像、关键字段无遮挡/马赛克，返回值包括： - 1：头像、关键字段无遮挡/马赛克 - 0：头像、关键字段有遮挡/马赛克
+ IsNoCover_propobility	否	float	“是否头像、关键字段无遮挡/马赛克”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 IsNoCover_propobility 超过0.3时，对应IsNoCover 返回1，低于0.3，则返回0
photo	否	string	当请求参数 detect_photo = true时返回，头像切图的 base64 编码（无编码头，需自行处理）
photo_location	否	object	当请求参数 detect_photo = true时返回，头像的位置信息（坐标0点为左上角）
card_image	否	string	当请求参数 detect_card = true时返回，身份证裁剪切图的 base64 编码（无编码头，需自行处理）
card_location	否	object	当请求参数 detect_card = true时返回，身份证裁剪切图的位置信息（坐标0点为左上角）
idcard_number_type	是	int	用于校验身份证号码、性别、出生是否一致，输出结果及其对应关系如下： - **-1：** 身份证正面所有字段全为空 **0：** 身份证证号不合法，此情况下不返回身份证证号 **1：** 身份证证号和性别、出生信息一致 **2：** 身份证证号和性别、出生信息都不一致 **3：** 身份证证号和出生信息不一致 **4：** 身份证证号和性别信息不一致
+ location	是	array[]	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++	是	uint32	表示定位位置的长方形的宽度

width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	否	string	识别结果字符串

返回示例（身份证头像面）


```
{
  "log_id": "1559208562721579319",
  "direction": 0,
  "image_status": "normal",
  "photo": "/9j/4AAQSkZJRgABA.....",
  "photo_location": {
     "width": 1189,
     "top": 638,
     "left": 2248,
     "height": 1483
  },
  "card_image": "/9j/4AAQSkZJRgABA.....",
  "card_location": {
     "top": 328,
     "left": 275,
     "width": 1329,
     "height": 571
  },
  "words_result": {
     "住址": {
        "location": {
           "left": 267,
           "top": 453,
           "width": 459,
           "height": 99
        },
        "words": "南京市江宁区弘景大道3889号"
     },
     "公民身份号码": {
        "location": {
           "left": 443,
           "top": 681,
           "width": 589,
           "height": 45
        },
        "words": "330881199904173914"
     },
     "出生": {
        "location": {
           "left": 270,
           "top": 355,
           "width": 357,
           "height": 45
        },
        "words": "19990417"
     },
     "姓名": {
        "location": {
           "left": 267,
           "top": 176,
           "width": 152,
           "height": 50
        },
        "words": "伍云龙"
```

```
        },
        "性别": {
            "location": {
                "left": 269,
                "top": 262,
                "width": 33,
                "height": 52
            },
            "words": "男"
        },
        "民族": {
            "location": {
                "left": 492,
                "top": 279,
                "width": 30,
                "height": 37
            },
            "words": "汉"
        }
    },
    "words_result_num": 6
}
```


返回示例（身份证国徽面）


```
{
    "words_result": {
        "失效日期": {
            "words": "20390711",
            "location": {
                "top": 445,
                "left": 523,
                "width": 153,
                "height": 38
            }
        },
        "签发机关": {
            "words": "陆丰市公安局",
            "location": {
                "top": 377,
                "left": 339,
                "width": 195,
                "height": 38
            }
        },
        "签发日期": {
            "words": "20190606",
            "location": {
                "top": 445,
                "left": 343,
                "width": 152,
                "height": 38
            }
        }
    },
    "log_id": "1559208562721579328",
    "words_result_num": 3,
    "error_code": 0,
    "image_status": "normal"
}
```


AES加密

您可以选择使用AES加密来请求本身份证识别接口，支持对身份证图片及识别结果进行加密后传输，示意图如下：



1. 在百度云控制台「文字识别-应用列表-管理」中获取您的AES Key
2. 使用AES Key对将要识别的图片进行加密
3. 将加密后的图片传入接口，请求参数AESEncry设置为true
4. 接口返回加密后的识别结果，使用AES Key进行解密，得到明文识别结果

使用AES加密不影响身份证识别接口支持的质量检测、风险检测等其他能力，也不影响识别效果。 **请求说明 请求示例**

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/idcard`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	AES加密后的图像数据，请对加密图片进行base64编码后进行urlencode，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
AESEncry	是	string	true/false	使用AES加密请传true
id_card_side	是	string	front/back	-front：身份证含照片的一面 -back：身份证带国徽的一面 自动检测身份证正反面，如果传参指定方向与图片相反，支持正常识别，返回参数image_status字段为"reversed_side"
detect_direction	否	string	-	此参数新版本无需传，支持自动检测图像旋转角度
detect_risk	否	string	true/false	是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)功能，默认不开启，即：false。 - true：开启，请查看返回参数risk_type； - false：不开启
detect_photo	否	string	true/false	是否检测头像内容，默认不检测。可选值：true-检测头像并返回头像的 base64 编码及位置信息

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Python

JAVA

C++

PHP

C#

```
 ##### -*- coding: utf-8 -*-
#
from urllib import urlencode
import base64
from urllib import unquote
import requests
import json
from Crypto.Cipher import AES
import binascii

ak = "ak"
##### 文字识别应用的API Key
sk = "sk"
##### 文字识别应用的Secret Key

aes = 'res'
##### aes key 从控制台 文字识别-应用列表-应用管理 获取

class AESCipher:
```

返回说明 返回示例

```
{
  "log_id":"2648325511",
  "result":"密文"
}
```

请对result密文进行base64解码后得到byte流，再进行AES解密，得到识别结果的明文。

身份证混贴识别

接口描述

身份证混贴识别支持自动检测与识别身份证正反面在同一张图片上的场景，一次识别图片中身份证正反面所有字段。

支持对二代居民身份证正反面所有8个字段进行结构化识别，包括姓名、性别、民族、出生日期、住址、身份证号、签发机关、有效期限，识别准确率超过99%；同时支持身份证正面头像检测，并返回头像切片的base64编码及位置信息。

同时，支持对用户上传的身份证图片进行图像风险和质量检测，可识别图片是否为复印件或临时身份证，是否被翻拍或编辑，是否存在正反颠倒、模糊、欠曝、过曝等质量问题。

增值能力	详情
裁剪能力	头像检测与切片：返回头像切片的base64编码及位置信息
	身份证检测与切片：返回身份证的base64编码及位置信息（去掉证件外围多余背景、自动矫正拍摄角度）
质量检测	身份证图片模糊检测
	身份证关键字段反光或过曝光
	身份证图片较暗或欠曝光
	身份证边框/四角不完整告警
	身份证头像或关键字段被遮挡/马赛克告警
风险检测	身份证复印件告警
	临时身份证告警
	身份证翻拍告警
	身份证PS编辑告警
	身份证证号不合法告警
	身份证证号和姓名、出生日期、性别信息不一致告警

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/multi_idcard`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_risk	否	string	true/false	是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)功能，默认不开启，即：false。 - true：开启，请查看返回参数risk_type； - false：不开启
detect_quality	否	string	true/false	是否开启身份证质量类型(边框/四角不完整、头像或关键字段被遮挡/马赛克)检测功能，默认不开启，即：false。 - true：开启，请查看返回参数card_quality； - false：不开启
detect_photo	否	string	true/false	是否检测头像内容，默认不检测，即：false。 - true：检测头像并返回头像的 base64 编码及位置信息； - false：不检测
detect_card	否	string	true/false	是否检测身份证进行裁剪，默认不检测，默认不检测，即：false。 - true：检测身份证并返回证照的 base64 编码及位置信息； - false：不检测

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/multi_idcard?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
+ card_info	是	object	识别结果信息，包含身份证的位置信息、风险检测信息、质量检测信息、正反面信息、方向信息
++ card_location	是	object	身份证的位置信息（坐标0点为左上角）
++ card_type	是	string	- idcard_front：身份证正面（头像面） - idcard_back：身份证反面（国徽面）
direction	是	int32	图像方向。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
++ image_status	是	string	normal-识别正常 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 other_type_card-其他类型证照 over_exposure-身份证关键字段反光或过曝 over_dark-身份证欠曝（亮度过低） unknown-未知状态
++ risk_type	否	string	输入参数 detect_risk = true 时，则返回该字段识别身份证类型: normal-正常身份证；copy-复印件；temporary-临时身份证；screen-翻拍；unknown-其他未知情况
++ edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
			输入参数 detect_quality = true 时，则返回该字段识别身份证**质量类型**：

card_quality	否	object	输入参数 detect_quality = true 时，则返回该字段识别身份证质量类型： IsClear - 是否清晰； IsComplete - 是否边框/四角完整； IsNoCover - 是否头像、关键字段无遮挡/马赛克。 及对应的概率：IsComplete_propobility、IsClear_propobility、IsNoCover_propobility，值在0-1之间，值越大表示图像质量越好。 **默认阈值（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）**：当 IsClear_propobility、IsComplete_propobility 超过0.5时，对应IsClear、IsComplete 返回1，低于0.5，则返回0； 当 IsNoCover_propobility 超过0.3时，对应IsNoCover 返回1，低于0.3，则返回0
++ photo	否	string	当请求参数 detect_photo = true时返回，头像切图的 base64 编码（无编码头，需自行处理）
++ photo_location	否	array[]	当请求参数 detect_photo = true时返回，头像的位置信息（坐标0点为左上角）
++ card_image	否	string	当请求参数 detect_card = true时返回，身份证裁剪切图的 base64 编码（无编码头，需自行处理）
++ idcard_number_type	是	string	用于校验身份证号码、性别、出生是否一致，输出结果及其对应关系如下： - **-1**： 身份证正面所有字段全为空 **0**： 身份证证号不合法，此情况下不返回身份证证号 **1**： 身份证证号和性别、出生信息一致 **2**： 身份证证号和性别、出生信息都不一致 **3**： 身份证证号和出生信息不一致 **4**： 身份证证号和性别信息不一致
+ card_result	是	object	识别结果
++ words	否	string	识别结果字符串
++ location	否	object	位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位位置的长方形的宽度
+++ height	否	uint32	表示定位位置的长方形的高度

返回示例


```
{
  "words_result": [
    {
      "card_info": {
        "card_location": {
          "top": 121,
          "left": 687,
          "width": 501,
          "height": 323
        },
        "risk_type": "normal",
        "idcard_number_type": 1,
```

```
            "edit_tool": "",
            "image_status": "normal",
            "card_type": "idcard_front",
            "direction": 0
        },
        "card_result": {
            "姓名": {
                "words": "王媛",
                "location": {
                    "top": 158,
                    "left": 788,
                    "width": 67,
                    "height": 26
                }
            },
            "民族": {
                "words": "汉",
                "location": {
                    "top": 204,
                    "left": 884,
                    "width": 19,
                    "height": 21
                }
            },
            "住址": {
                "words": "北京市海淀区西北旺东路10号院",
                "location": {
                    "top": 282,
                    "left": 782,
                    "width": 205,
                    "height": 51
                }
            },
            "公民身份号码": {
                "words": "410433199510104518",
                "location": {
                    "top": 385,
                    "left": 863,
                    "width": 266,
                    "height": 29
                }
            },
            "出生": {
                "words": "19951010",
                "location": {
                    "top": 240,
                    "left": 784,
                    "width": 163,
                    "height": 25
                }
            },
            "性别": {
                "words": "女",
                "location": {
                    "top": 202,
                    "left": 789,
                    "width": 19,
                    "height": 22
                }
            }
        }
    },
```

```
        {
            "card_info": {
                "card_location": {
                    "top": 526,
                    "left": 66,
                    "width": 524,
                    "height": 323
                },
                "risk_type": "normal",
                "edit_tool": "",
                "image_status": "normal",
                "card_type": "idcard_back",
                "direction": 0
            },
            "card_result": {
                "失效日期": {
                    "words": "20350413",
                    "location": {
                        "top": 801,
                        "left": 394,
                        "width": 95,
                        "height": 19
                    }
                },
                "签发机关": {
                    "words": "北京市公安局海淀分局",
                    "location": {
                        "top": 760,
                        "left": 290,
                        "width": 180,
                        "height": 19
                    }
                },
                "签发日期": {
                    "words": "20150413",
                    "location": {
                        "top": 801,
                        "left": 289,
                        "width": 89,
                        "height": 19
                    }
                }
            }
        }
    ],
    "direction": 0,
    "log_id": 1397109704106180608
}
```


身份证识别（金融加密版）

能力介绍

根据人民银行在《个人金融信息保护技术规范》中指出，身份证证件信息传输过程的参与方应当保证信息传输过程中的保密性、完整性和可用性，信息通过公共网络传输时，应使用加密通道或数据加密的方式进行传输。

为保护个人身份证隐私信息不被窃取泄露，面向金融、支付机构等客户，推出身份证识别（金融加密版）接口，联手百度安全实验室，实现“一次一密”的金融级数据加密传输，能够有效防止信息泄露。

本身份证识别（加密版）接口的主要功能为：对用户身份证图片与识别结果进行加密，避免在数据传输过程中，身份证图片与其中的姓名、身份证号、住址等隐私数据被泄露，保护数据安全传输，免受恶意攻击。

接口说明

身份证识别（金融加密版）接口支持所有身份证识别接口的能力，并在此基础上，提供信息传输加密，身份证图片的加密与识别结果的解密操作需要在您的服务端完成。

支持对二代居民身份证正反面所有8个字段进行结构化识别，包括姓名、性别、民族、出生日期、住址、身份证号、签发机关、有效期限，识别准确率超过99%；同时支持身份证正面头像检测，并返回头像切片的base64编码及位置信息。

同时，支持对用户上传的身份证图片进行图像风险和质量检测，可识别图片是否为复印件或临时身份证，是否被翻拍或编辑，是否存在正反颠倒、模糊、欠曝、过曝等质量问题。

为使您更好的使用本接口，我们提供了接入工具文件及工具使用文档（工具仅支持*Java*开发语言），如有需要，请联系您的商务经理或 提交工单 联系我们。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/idcard_enc`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	加密后的图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
enc_aes_key	是	string	-	白盒加密后的AES key，请参照工具文件及工具使用文档，使用加密文件生成
id_card_side	是	string	front/back	- front：身份证含照片的一面 - back：身份证带国徽的一面 自动检测身份证正反面，如果传参指定方向与图片相反，支持正常识别，返回参数image_status字段为"reversed_side"
detect_risk	否	string	true/false	是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)功能，默认不开启，即：false。 - true：开启，请查看返回参数risk_type； - false：不开启
detect_quality	否	string	true/false	是否开启身份证质量类型(边框/四角不完整、头像或关键字段被遮挡/马赛克)检测功能，默认不开启，即：false。 - true：开启，请查看返回参数card_quality； - false：不开启
detect_photo	否	string	true/false	是否检测头像内容，默认不检测。可选值：true-检测头像并返回头像的 base64 编码及位置信息
detect_card	否	string	true/false	是否检测身份证进行裁剪，默认不检测。可选值：true-检测身份证并返回证照的 base64 编码及位置信息

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/idcard_enc?access_token=【调用鉴权接口获取的token】' --data 'id_card_side=front&image=【图片Base64编码，需UrlEncode】&enc_aes_key=enc_aes_key' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

对返回结果解密后的结果信息与身份证识别一致，如下：

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	是	int32	图像方向。 - - 1：未定义， - 0：正向， - 1: 逆时针90度， - 2：逆时针180度， - 3：逆时针270度
image_status	是	string	normal-识别正常 reversed_side-身份证正反面颠倒 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 other_type_card-其他类型证照 over_exposure-身份证关键字段反光或过曝 over_dark-身份证欠曝（亮度过低） unknown-未知状态
risk_type	否	string	输入参数 detect_risk = true 时，则返回该字段识别身份证类型: normal-正常身份证；copy-复印件；temporary-临时身份证；screen-翻拍；unknown-其他未知情况

edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
card_quality	否	object	输入参数 detect_quality = true 时，则返回该字段识别身份证**质量类型**: IsClear - 是否清晰； IsComplete - 是否边框/四角完整； IsNoCover - 是否头像、关键字段无遮挡/马赛克。 及对应的概率：IsComplete_propobility、IsClear_propobility、IsNoCover_propobility，值在0-1之间，值越大表示图像质量越好。 **默认阈值（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：**当 IsClear_propobility、IsComplete_propobility 超过0.5时，对应IsClear、IsComplete 返回1，低于0.5，则返回0； 当 IsNoCover_propobility 超过0.3时，对应IsNoCover 返回1，低于0.3，则返回0
photo	否	string	当请求参数 detect_photo = true时返回，头像切图的 base64 编码（无编码头，需自行处理）
photo_location	否	array[]	当请求参数 detect_photo = true时返回，头像的位置信息（坐标0点为左上角）
card_image	否	string	当请求参数 detect_card = true时返回，身份证裁剪切图的 base64 编码（无编码头，需自行处理）
card_location	否	object	当请求参数 detect_card = true时返回，身份证裁剪切图的位置信息（坐标0点为左上角）
idcard_number_type	是	string	用于校验身份证号码、性别、出生是否一致，输出结果及其对应关系如下： **-1：** 身份证正面所有字段全为空 **0：** 身份证证号不合法，此情况下不返回身份证证号 **1：** 身份证证号和性别、出生信息一致 **2：** 身份证证号和性别、出生信息都不一致 **3：** 身份证证号和出生信息不一致 **4：** 身份证证号和性别信息不一致
words_result	是	array[]	定位和识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ location	是	array[]	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	否	string	识别结果字符串

银行卡识别

接口描述

支持对主流银行卡的卡号、有效期、发卡行、卡片类型、持卡人、银行卡号位置 6 个关键字段进行结构化识别，识别准确率超过99%。同时支持返回银行卡号的字段位置坐标，及开启银行卡质量类型检测。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/bankcard`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	类型	是否必须	说明
image	string	和url二选一	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	string	和image二选一	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	string	否	是否返回银行卡号的字段位置坐标，默认为 false，即不返回。可选值： - true：返回 - false：不返回
detect_quality	string	否	是否开启银行卡质量类型（清晰模糊、边框/四角不完整）检测功能，默认不开启，即：false。 - true：开启，请查看返回参数card_quality； - false：不开启

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/bankcard?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
direction	int32	是	图像方向。 - - 1：未定义; - 0：正向; - 1：逆时针90度; - 2：逆时针180度; - 3：逆时针270度
result	object	是	返回结果
+ bank_card_number	string	是	银行卡卡号
+ valid_date	string	是	有效期
+ bank_card_type	uint32	是	银行卡类型，0：不能识别; 1：借记卡; 2：贷记卡（原信用卡大部分为贷记卡）; 3：准贷记卡; 4：预付费卡
+ bank_name	string	是	银行名，不能识别时为空
+ holder_name	string	是	持卡人姓名，不能识别时为空
+ bank_card_number_location	string	否	银行卡号的字段位置坐标，当输入参数 location = true 时返回
++ left	string	否	表示银行卡号定位位置的长方形左上顶点的水平坐标，当输入参数 location = true 时返回

++ top	string	否	表示银行卡号定位位置的长方形左上顶点的垂直坐标，当输入参数 location = true 时返回
++ width	string	否	表示银行卡号定位位置的长方形的宽度，当输入参数 location = true 时返回
++ height	string	否	表示银行卡号定位位置的长方形的高度，当输入参数 location = true 时返回
+ card_quality	否	object	输入参数 detect_quality = true 时，则返回该字段识别银行卡质量类型
++ IsClear	是	string	质量类型，是否清晰
++ IsClear_probability	是	string	“是否清晰”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 IsClear_probability 超过0.5时，对应 IsClear 返回1，低于0.5，则返回0
++ IsComplete	是	string	质量类型，是否边框/四角完整
++ IsComplete_probability	是	string	“是否边框/四角完整”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 IsComplete_probability 超过0.5时，对应 IsComplete 返回1，低于0.5，则返回0

返回示例


```
{
  "log_id": 144718895115129615,
  "direction": 0,
  "result": {
     "bank_card_number": "3568 8900 8000 0005",
     "valid_date": "07/21",
     "bank_card_type": 2,
     "bank_name": "招商银行",
     "holder_name": "MR.CHENTA"
  }
}
```


营业执照识别

接口描述

支持对不同版式营业执照的证件编号、社会信用代码、单位名称、地址、法人、类型、成立日期、有效日期、经营范围等关键字段进行结构化识别。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/business_license`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	类型	是否必须	说明
image	string	和url二选一	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px,支持jpg/jpeg/png/bmp格式
url	string	和image二选一	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
accuracy	string	否	此参数新版本无需传，可选值：normal,high
risk_warn	string	否	是否开启风险类型功能，默认不开启，即：false。 - false：不开启 - true：开启
detect_quality	string	否	是否开启质量类型（清晰模糊、边框/四角不完整）检测功能，默认不开启，即：false。 - true：开启，结果请查看返回参数card_quality - false：不开启
fullwidth_shift	string	否	是否开启全角符号转换，默认不开启，即：false。 - false：不开启，单位名称、类型、经营范围字段内括号以半角输出 - true：开启，单位名称、类型、经营范围字段内括号以全角输出

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/business_license?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

参数	是否必须	类型	说明
log_id	是	uint64	请求标识码，随机数，唯一。
direction	是	uint32	图像方向。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
risk_type	否	string	当输入参数 risk_warn=true 时返回识出的营业执照的类型：normal-正常营业执照；copy-复印件；screen-翻拍；scan-扫描
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
card_quality	否	object	输入参数 detect_quality = true 时，返回该字段，包含质量检测结果
+ is_clear	是	string	质量类型，是否清晰，返回值包括： - 1：清晰 - 0：不清晰
+ is_clear_propobility	是	string	“是否清晰”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 is_clear_propobility 超过0.5时，对应 is_clear 返回1，低于0.5，则返回0
+ is_complete	是	string	质量类型，是否边框/四角完整，返回值包括： - 1：边框/四角完整 - 0：边框/四角不完整
+ is_complete_propobility	是	string	“是否边框/四角完整”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 is_complete_propobility 超过0.5时，对应 is_complete 返回1，低于0.5，则返回0
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	否	string	识别结果字符串

返回示例


```
{
"words_result": {
 "经营范围": {
  "location": {
   "top": 589
```

```
    "top": 535,
    "left": 381,
    "width": 90,
    "height": 19
   },
   "words": "商务服务业"
  },
  "组成形式": {
   "location": {
    "top": -1,
    "left": -1,
    "width": 0,
    "height": 0
   },
   "words": "无"
  },
  "法人": {
   "location": {
    "top": 537,
    "left": 381,
    "width": 36,
    "height": 19
   },
   "words": "方平"
  },
  "证件编号": {
   "location": {
    "top": 218,
    "left": 302,
    "width": 140,
    "height": 15
   },
   "words": "921MA190538210301"
  },
  "注册资本": {
   "location": {
    "top": 431,
    "left": 1044,
    "width": 152,
    "height": 21
   },
   "words": "200万元"
  },
  "单位名称": {
   "location": {
    "top": 431,
    "left": 384,
    "width": 71,
    "height": 20
   },
   "words": "有限公司"
  },
  "有效期": {
   "location": {
    "top": 536,
    "left": 1044,
    "width": 198,
    "height": 20
   },
   "words": "长期"
  },
  "社会信用代码": {
   "location": {
```

```
    "top": 300,
    "left": 241,
    "width": 156,
    "height": 16
   },
   "words": "10440119MA06M85"
  },
  "实收资本": {
   "location": {
    "top": -1,
    "left": -1,
    "width": 0,
    "height": 0
   },
   "words": "无"
  },
  "有效期起始日期": {
   "location": {
    "top": 536,
    "left": 1044,
    "width": 198,
    "height": 20
   },
   "words": "2019年01月01日"
  },
  "核准日期": {
   "location": {
    "top": 884,
    "left": 1188,
    "width": 199,
    "height": 22
   },
   "words": "2019年01月01日"
  },
  "成立日期": {
   "location": {
    "top": 484,
    "left": 1043,
    "width": 126,
    "height": 19
   },
   "words": "2019年01月01日"
  },
  "税务登记号": {
   "location": {
    "top": -1,
    "left": -1,
    "width": 0,
    "height": 0
   },
   "words": "无"
  },
  "地址": {
   "location": {
    "top": 588,
    "left": 1043,
    "width": 55,
    "height": 22
   },
   "words": "广州市"
  },
  "登记机关": {
```

```
    "location": {
     "top": 0,
     "left": 0,
     "width": 0,
     "height": 0
    },
    "words": "无"
   },
   "类型": {
    "location": {
     "top": 484,
     "left": 382,
     "width": 258,
     "height": 20
    },
    "words": "有限责任公司(自然人投资或控股)"
   }
  },
  "direction": 0,
  "words_result_num": 16,
  "log_id": "3166723741167575145"
}
```


护照识别

接口描述

支持对中国大陆护照个人资料页所有15个字段进行结构化识别，包括国家码、护照号、姓名、姓名拼音、性别、出生地点、出生日期、签发地点（不支持境外签发地）、签发日期、有效期、签发机关、护照类型、国籍、MRZCode1、MRZCode2。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/passport`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/passport?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	int32	是	图像方向。 - - 1：未定义; - 0：正向; - 1：逆时针90度; - 2：逆时针180度; - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ location	是	uint32	水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ height	是	uint32	表示定位位置的长方形的高度
++ width	是	uint32	表示定位位置的长方形的宽度
+ words	是	string	识别内容

返回示例


```
{
  "log_id": 7377468409496932872,
  "direction": 0,
  "words_result_num": 15,
  "words_result": {
    "护照类型": {
        "location": {
            "width": 59,
            "top": 200,
            "left": 762,
            "height": 26
        },
        "words": "P"
    },
    "国家码": {
        "location": {
            "width": 59,
            "top": 200,
            "left": 762,
            "height": 26
        },
        "words": "CHN"
    },
    "护照签发地点": {
        "location": {
            "width": 236,
            "top": 505,
            "left": 558,
            "height": 43
        },
        "words": "山东/SHANDONG"
    },
    "MRZCode2": {
        "location": {
            "width": 1252,
            "top": 797,
```

```
            "left": 145,
            "height": 88
        },
        "words": "E898657303CHNSDM0702<<<<<<<<<<<<<"
    },
    "有效期至": {
        "location": {
            "width": 287,
            "top": 528,
            "left": 955,
            "height": 46
        },
        "words": "20261004"
    },
    "签发机关": {
        "location": {
            "width": 271,
            "top": 583,
            "left": 552,
            "height": 42
        },
        "words": "出入境管理局"
    },
    "MRZCode1": {
        "location": {
            "width": 1201,
            "top": 781,
            "left": 162,
            "height": 45
        },
        "words": "PONSUN<<JIAJIA<<<<<<<<<<<<<<<<<<<<<<<<<<"
    },
    "护照号码": {
        "location": {
            "width": 237,
            "top": 215,
            "left": 994,
            "height": 48
        },
        "words": "E89865730"
    },
    "签发日期": {
        "location": {
            "width": 279,
            "top": 453,
            "left": 955,
            "height": 46
        },
        "words": "20161005"
    },
    "出生地点": {
        "location": {
            "width": 216,
            "top": 429,
            "left": 564,
            "height": 43
        },
        "words": "山东/SHANDONG"
    },
    "姓名": {
        "location": {
            "width": 159,
            "top": 247
```

```
            "top": 247,
            "left": 581,
            "height": 34
        },
        "words": "孙嘉佳"
    },
    "姓名拼音": {
        "location": {
            "width": 229,
            "top": 279,
            "left": 578,
            "height": 41
        },
        "words": "SUN,JIAJIA"
    },
    "国籍": {
        "location": {
            "width": 209,
            "top": 366,
            "left": 695,
            "height": 42
        },
        "words": "中国/CHINESE"
    },
    "生日": {
        "location": {
            "width": 202,
            "top": 382,
            "left": 950,
            "height": 39
        },
        "words": "19950723"
    },
    "性别": {
        "location": {
            "width": 73,
            "top": 357,
            "left": 570,
            "height": 34
        },
        "words": "男/M"
    }
  }
}
```


护照识别（港澳台地区及境外）

接口描述

支持对港澳台地区及境外护照进行结构化识别，包括MRZCode1、MRZCode2、出生日期、国家码、国籍、姓名拼音、性别、护照号、护照类型、有效期，10 个关键字段。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/overseas_passport`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/overseas_passport?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ word	是	string	字段识别结果，对应 **MRZCode1、MRZCode2、出生日期、国家码、国籍、姓名拼音、性别、护照号、护照类型、有效期** 10 个字段的识别结果

返回示例

```
{
  "words_result": {
    "MRZCode2": [
      {
        "word": "XX000000<0FRA7501012F19073021234567890<<<<70"
      }
    ],
    "护照类型": [
      {
        "word": ""
      }
    ],
    "护照号": [
      {
        "word": "XX000000"
      }
    ],
    "出生日期": [
      {
        "word": "19750101"
      }
    ],
```

```
        "国家码": [
          {
            "word": ""
          }
        ],
        "国籍": [
          {
            "word": ""
          }
        ],
        "姓名拼音": [
          {
            "word": "SPECIMEN SAMPLE"
          }
        ],
        "MRZCode1": [
          {
            "word": "P<INPSPECIMEN<<SAMPLE<<<<<<<<<<<<<<<<<<<<<<"
          }
        ],
        "性别": [
          {
            "word": "女"
          }
        ],
        "有效期": [
          {
            "word": "20190730"
          }
        ]
    },
    "words_result_num": 10,
    "log_id": 1744966034361997990
}
```


社保卡识别

接口描述

支持识别全国各地社保卡，支持识别社会保障卡号、姓名、性别、出生日期、银行卡号、有效期限等8个字段。

在线调试
您可以在[示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=卡证OCR&api=rest/2.0/ocr/v1/social_security_card&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。
请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/social_security_card`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	类型	是否必须	说明
image	string	和url二选一	图像数据，base64编码后进行urlencode，需去掉编码头（）</br>要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	string	和image二选一	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/social_security_card?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
##### encoding:utf-8

import requests
import base64

'''
社保卡识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/social_security_card"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 社保卡识别
*/
public class SocialSecurityCard {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String securityCard() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/social_security_card";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        BankCard.bankCard();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/social_security_card";
static std::string securityCard_result;
/**
 * curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
 * @param 参数定义见libcurl文档
 * @return 返回值定义见libcurl文档
 */
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    bankCard_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
 * 银行卡识别
 * @return 调用成功返回0，发生错误返回其他错误码
 */
int bankCard(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = securityCard_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
 * 发起http post请求(REST API), 并获取REST请求的结果
 * @param string $url
 * @param string $param
 * @return - http response body if succeeds, else false.
 */
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/social_security_card?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class SocialSecurityCard
    {
        // 社保卡识别
        public static string bankCard()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/social_security_card?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("社保卡识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
direction	int32	否	图像方向，当 detect_direction = true 时，返回该参数。 </br>- -1：未定义;</br>- 0：正向;</br>- 1：逆时针90度;</br>- 2：逆时针180度;</br>- 3：逆时针270度
words_result	object	是	返回结果
+ card_number	string	是	卡号
+ name	string	是	姓名
+ sex	string	是	性别
+ social_security_number	string	是	社会保障卡号
+ birth_date	string	是	出生日期
+ issue_date	string	是	签发日期
+ bank_card_number	string	是	银行卡号

| + expiry_date | string | 是 | 有效期限 |

返回示例

```JSON
{
    "direction": "0",
    "words_result": {
        "sex": {
            "word": "男"
        },
        "birth_date": {
            "word": "1993年6月16日"
        },
        "issue_date": {
            "word": ""
        },
        "name": {
            "word": "徐乐"
        },
        "card_number": {
            "word": "122932663447"
        },
        "bank_card_number": {
            "word": ""
        },
        "expiry_date": {
            "word": ""
        },
        "social_security_number": {
            "word": "130201199306168223"
        }
    },
    "log_id": 1438866840412542540
}

```

户口本识别

接口描述

支持对户口本内常住人口登记卡的全部 22 个字段，包括户号、姓名、与户主关系、性别、出生地、民族、出生日期、身份证号、本市县其他住址、曾用名、籍贯、宗教信仰、身高、血型、文化程度、婚姻状况、兵役状况、服务处所、职业、何时由何地迁往本市、何时由何地迁往本址、登记日期。

支持对户口本内户主页的5个字段进行结构化识别，包括户别、户主姓名、户号、住址、户主页时间。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=卡证OCR&api=rest/2.0/ocr/v1/household_register&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/household_register`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
household_register_side	否	string	subpage/homepage	-subpage：**默认值**，常住人口登记卡（成员页）</br>-homepage：户主页 </br>

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/household_register?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
户口本识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/household_register"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
```
~~~

```
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 户口本识别
*/
public class HouseholdRegister {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String householdRegister() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/household_register";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        HouseholdRegister.householdRegister();
    }
}
```
```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
```

```
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/household_register";
static std::string householdRegister_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    householdRegister_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 户口本识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int householdRegister(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = householdRegister_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
```

```
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/household_register?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class HouseholdRegister
    {
        // 户口本识别
        public static string householdRegister()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/household_register?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
```

```
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("户口本识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```

返回说明

返回参数

常住人口登记卡（成员页）

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ Name	是	object{}	姓名
++ words	是	string	所属字段的具体内容，以下各字段均含有此元素
+ Relationship	是	object{}	户主或与户主关系
+ Sex	是	object{}	性别
+ BirthAddress	是	object{}	出生地
+ Nation	是	object{}	民族
+ Birthday	是	object{}	生日
+ CardNo	是	object{}	身份证号
+ HouseholdNum	是	object{}	户号
+ FormerName	是	object{}	曾用名
+ Hometown	是	object{}	籍贯
+ OtherAddress	是	object{}	本市（县）其他住址
+ Belief	是	object{}	宗教信仰
+ Height	是	object{}	身高
+ BloodType	是	object{}	血型
+ Education	是	object{}	文化程度
+ MaritalStatus	是	object{}	婚姻状况
+ VeteranStatus	是	object{}	兵役状况
+ WorkAddress	是	object{}	服务处所
+ Career	是	object{}	职业
+ WWToCity	是	object{}	何时由何地迁来本市(县)
+ WWHere	是	object{}	何时由何地迁往本址
+ Date	是	object{}	登记日期

户主页

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
direction	是	int32	图像方向，当图像旋转时，返回该参数。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result	是	object{}	识别结果
+ HouseholdType	是	object{}	户别
++ words	是	string	所属字段的具体内容，以下各字段均含有此元素
+ HouseholderName	是	object{}	户主姓名
+ HouseholdNum	是	object{}	户号
+ Address	是	object{}	住址
+ IssueDate	是	object{}	签发日期

返回示例


```
{
"words_result": {
 "Nation": {
  "words": "汉族"
 },
 "Sex": {
  "words": "女"
 },
 "Birthday": {
  "words": "1994年7月27日"
 },
 "Date": {
  "words": "2017年7月27日"
 },
 "BirthAddress": {
  "words": "四川省"
 },
 "Name": {
  "words": "王燕"
 },
      "FormerName ": {
  "words": "王佳"
 },
 "HouseholdNum": {
  "words": "000007670"
 },
 "WWToCity": {
  "words": "由久居"
 },
 "WWHere": {
  "words": "1994年07月27日因出生迁来"
 },
 "CardNo": {
  "words": "112102199407273123"
 },
 "Education": {
```

```
    "words": "初中毕业"
  },
  "Relationship": {
    "words": "独生女"
  },
  "Height": {
    "words": "170厘米"
  },
      "Career": {
    "words": "无"
  },
  "WorkAddress": {
    "words": "无"
  },
  "Hometown": {
    "words": "四川省"
  }
      "OtherAddress": {
    "words": "四川省乐山市"
  }
      "Belief": {
    "words": "佛教"
  }
      "BloodType": {
    "words": "A型"
  }
       "MaritalStatus": {
    "words": "未婚"
  }
      "VeteranStatus": {
    "words": "无"
  }
  },
  "log_id": "1407164137607266304",
  "words_result_num": 22
}
```


出生医学证明识别

接口描述

支持对出生医学证明的23个关键字段进行结构化识别，包括新生儿姓名、性别、出生时间、父亲姓名、母亲姓名、出生证编号等。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/birth_certificate`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/birth_certificate?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	int32	是	图像方向。 - - 1：未定义; - 0：正向; - 1：逆时针90度; - 2：逆时针180度; - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果数组
+ BabyBirthday	是	string	出生时间
+ BirthProvince	是	string	出生地点_省
+ BirthCity	是	string	出生地点_市
+ BirthCounty	是	string	出生地点_县（区）
+ BirthWeight	是	string	出生体重
+ BirthLength	是	string	出生身长
+ GestationalAge	是	string	出生孕周
+ BabyName	是	string	姓名
+ BabySex	是	string	性别
+ Code	是	string	出生证编号
+ Hospital	是	string	出生医院/医疗机构名称
+ FatherName	是	string	父亲姓名
+ FatherID	是	string	父亲身份证号
+ FatherNationality	是	string	父亲国籍
+ FatherEthnic	是	string	父亲民族
+ FatherAddress	是	string	父亲住址
+ FatherAge	是	string	父亲年龄
+ MotherName	是	string	母亲姓名
+ MotherID	是	string	母亲身份证号
+ MotherNationality	是	string	母亲国籍
+ MotherEthnic	是	string	母亲民族
+ MotherAddress	是	string	母亲地址
+ MotherAge	是	string	母亲年龄

返回示例

```
{
  "words_result": {
    "BirthProvince": {
      "words": "山东"
    },
    "BirthWeight": {
      "words": "3450克"
    },
    "MotherNationality": {
      "words": "中国"
```

```
        "words": "中国"
    },
    "FatherAddress": {
        "words": "山东省青岛市"
    },
    "MotherEthnic": {
        "words": "汉族"
    },
    "MotherAddress": {
        "words": "山东省青岛市"
    },
    "Code": {
        "words": "U370073333"
    },
    "BabySex": {
        "words": "女"
    },
    "BirthCounty": {
        "words": "黄岛区"
    },
    "BabyBirthday": {
        "words": "2020年06月19日17时00分"
    },
    "BabyName": {
        "words": "安琪"
    },
    "FatherAge": {
        "words": "30岁"
    },
    "FatherEthnic": {
        "words": "汉族"
    },
    "MotherName": {
        "words": "田田"
    },
    "MotherID": {
        "words": "370284199509086021"
    },
    "BirthLength": {
        "words": "50厘米"
    },
    "GestationalAge": {
        "words": "41+1周"
    },
    "BirthCity": {
        "words": "青岛"
    },
    "MotherAge": {
        "words": "25岁"
    },
    "Hospital": {
        "words": "青岛市黄岛区医院"
    },
    "FatherNationality": {
        "words": "中国"
    },
    "FatherID": {
        "words": "370284199207126034"
    },
    "FatherName": {
        "words": "王军"
    }
},
```

```
    "log_id": 1420219665606852609,
    "words_result_num": 23,
    "direction": 0
}
```


企业工商信息查询（标准版）

接口描述

传入企业名称、注册号、统一社会信用代码中的任意一种，即可查询企业的基本信息，包括法人、注册资本、信用代码、经营状态等20+字段。

注：目前支持工商注册的企业认证，事业单位、律师事务所等信息暂不支持验证，发生工商变更或刚注册的企业信息，预计在企业公示的核准日期T+3个工作日后可进行验证。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/businesslicense_verification_standard`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
verifynum	是	string	-	查询关键字段（企业名称、注册号、社会统一信用代码，可任意输入其中一种）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

OCR-企业工商信息查询（标准版）

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/businesslicense_verification_standard?access_token=【调用鉴权接口获取的token】' --data 'verifynum=查询关键字段（企业名称、注册号、社会统一信用代码，可任意输入其中一种）' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
companyname	是	string	企业名称
companytype	是	string	企业类型
legalperson	是	string	法定代表人
capital	是	string	注册资本
companycode	是	string	注册码
companyaddress	是	string	企业地址
businessscope	是	string	经营范围
authority	是	string	登记机关
companystatus	是	string	登记状态
establishdate	是	string	成立时间
creditno	是	string	统一社会信用代码
operationstartdate	是	string	营业日期
operationenddate	是	string	截止日期
issuedate	是	string	核准时间
province	是	string	所在省份
provincecode	是	string	所在省份-行政区号
city	是	string	所在市
citycode	是	string	所在市-行政区号
district	是	string	所在地区
districtcode	是	string	所在地区-行政区号
regcapcur	是	string	注册资本币种
orgcode	是	string	组织机构代码
licensedbusinessscope	是	string	许可经营范围
companyenglishname	是	string	企业英文名称
onceusedname	是	string	企业曾用名
orgcompanycode	是	string	原注册号
paidincapital	是	string	实收资本
revokedate	是	string	吊销日期
logoffdate	是	string	注销日期

返回示例

```
{
  "words_result_num": 29,
  "words_result": {
    "companyname": "百度在线网络技术（北京）有限公司",
    "companytype": "有限责任公司(外国法人独资)",
    "legalperson": "崔珊珊",
    "capital": "4520万",
    "companycode": "91110108717743469K",
    "companyaddress": "北京市海淀区上地十街10号百度大厦三层",
    "businessscope": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告；软件开发；技术开发、技术推广、技术转让；销售自产产品、医疗器械II类、电子产品、器件和元件、计算机、软件及辅助设备、灯具、五金交电、自行开发后的产品；计算机系统集成；委托加工生产通讯设备；销售第三类医疗器械。（市场主体依法自主选择经营项目，开展经营活动；销售第三类医疗器械以及依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动；不得从事国家和本市产业政策禁止和限制类项目的经营活动。）",
    "authority": "北京市海淀区市场监督管理局",
    "companystatus": "在营（开业）企业",
    "establishdate": "2000-01-18 00:00:00",
    "creditno": "91110108717743469K",
    "province": "北京市",
    "operationstartdate": "2000-01-18 00:00:00",
    "operationenddate": "长期",
    "issuedate": "2020-11-27 00:00:00",
    "provincecode": "110000",
    "city": "北京市",
    "citycode": "110000",
    "district": "海淀区",
    "districtcode": "110108",
    "regcapcur": "美元",
    "orgcode": "717743469",
    "licensedbusinessscope": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告；软件开发；技术开发、技术推广、技术转让；销售自产产品、医疗器械II类、电子产品、器件和元件、计算机、软件及辅助设备、灯具、五金交电、自行开发后的产品；计算机系统集成；委托加工生产通讯设备；销售第三类医疗器械。（市场主体依法自主选择经营项目，开展经营活动；销售第三类医疗器械以及依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动；不得从事国家和本市产业政策禁止和限制类项目的经营活动。）",
    "companyenglishname": "Baidu Online Network Technology(Beijing)Co.,Ltd.",
    "onceusedname": [],
    "orgcompanycode": "",
    "paidincapital": "4520.0万",
    "revokedate": "",
    "logoffdate": ""
  },
  "log_id": 1688899176144648671
}
```


企业工商信息查询（高级版）

接口描述

传入企业名称、注册号、统一社会信用代码中的任意一种，具体返回企业全维度信息，包括工商基本信息、分支机构信息、企业变更信息、纳税信息、联系信息、企业高管信息、经营异常信息、动产抵押信息、曾用名信息、股东信息、行政处罚信息、行政许可信息、股权出资信息、失信信息、被执行信息等。

注：目前支持工商注册的企业认证，事业单位、律师事务所等信息暂不支持验证，发生工商变更或刚注册的企业信息，预计在企业公示的核准日期T+3个工作日后可进行验证。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/ocr/v1/businesslicense_verification_detailed

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
verifynum	是	string	-	查询关键字段（企业名称、注册号、统一社会信用代码，可任意输入其中一种）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
 OCR-企业工商信息查询（高级版）
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/businesslicense_verification_detailed?access_token=【调用鉴权接口获取的token】' --data 'verifynum=查询关键字段（（企业名称、注册号、统一社会信用代码，可任意输入其中一种）' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数

words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
base	是	object	工商基本信息
+ legalperson	是	string	法人名
+ establishdate	是	string	成立日期
+ revokedate	是	string	吊销日期
+ companystatus	是	string	企业状态
+ province	是	string	省份
+ creditno	是	string	统一社会信用代码
+ capital	是	string	注册资本
+ companytype	是	string	企业类型
+ companyaddress	是	string	地址
+ businessscope	是	string	经营范围
+ businessdatefrom	是	string	营业开始日期
+ businessdateto	是	string	营业结束日期
+ issuedate	是	string	发照日期
+ orgcode	是	string	组织机构代码
+ isonstock	是	string	是否上市 (0为未上市，1为上市)
+ stocknumber	是	string	上市公司代码
+ stocktype	是	string	上市类型
+ keyno	是	string	内部keyno
+ companyname	是	string	企业名称
+ companycode	是	string	注册号
+ authority	是	string	登记机关
+ regcapcur	是	string	注册资本币种
branches	是	array	分支机构，每个数组可能包含多个object
+ companycode	否	string	注册号
+ companyname	否	string	名称
+ authority	否	string	登记机关
+ creditno	否	string	统一社会信用代码
+ legalperson	否	string	法人姓名
changes	是	array	企业变更
+ changefield	否	string	变更事项
+ changebefore	否	string	变更前内容
+ changeafter	否	string	变更后内容
+ changedate	否	string	变更日期
taxcreditems	是	array	纳税信息
+ taxpayerno	否	string	纳税人识别号
+ taxpayername	否	string	纳税人名称
+ year	否	string	评价年度

+ level	否	string	信用等级
contactinfo	是	object	联系信息
+ website	是	array	网站信息，每个数组可能包含多个object
+ + name	否	string	网站名称
+ + url	否	string	网站地址
+ phonenumber	是	string	联系电话
+ email	是	string	联系邮箱
employees	是	array	企业高管，每个数组可能包含多个object
+ employeename	否	string	姓名
+ position	否	string	职位
exceptions	是	array	经营异常
+ addreason	否	string	列入经营异常名录原因
+ adddate	否	string	列入日期
+ removereason	否	string	移出经营异常名录原因
+ removedate	否	string	移出日期
+ decisionoffice	否	string	作出决定机关
+ removedecisionoffice	否	string	移除决定机关
industry	是	array	行业信息
+ industry	否	string	国民经济行业分类大类名称
+ subindustry	否	string	国民经济行业分类小类名称
liquidation	是	array	公司清算信息
+ leader	否	string	清算组负责人
+ member	否	string	清算组成员
mpledges	是	array	动产抵押，每个数组可能包含多个object
+ registerno	否	string	登记编号
+ registerdate	否	string	登记时间
+ publicdate	否	string	公示时间
+ registeroffice	否	string	登记机关
+ debtsecuredamount	否	string	被担保债权数额
+ status	否	string	状态
originalname	是	array	曾用名，每个数组可能包含多个object
+ name	否	string	曾用名
+ changedate	否	string	变更日期
partners	是	array	股东信息，每个数组可能包含多个object
+ stockname	否	string	股东
+ stocktype	否	string	股东类型
+ stockpercent	否	string	出资比例
+ stockcapital	否	string	认缴出资额
+ shouddate	否	string	认缴出资时间
+ investtype	否	string	认缴出资方式

+ stockrealcapital	否	string	实缴出资额
+ capidate	否	string	实缴时间
+ investname	否	string	实际出资方式
+ concur	否	string	出资币种
penalties	是	array	行政处罚，每个数组可能包含多个object
+ docno	否	string	行政处罚决定书文号
+ penaltytype	否	string	违法行为类型
+ officename	否	string	行政处罚决定机关名称
+ content	否	string	行政处罚内容
+ penaltydate	否	string	作出行政处罚决定日期
+ publicdate	否	string	作出行政公示日期
+ remark	否	string	备注
permissions	是	array	行政许可，每个数组可能包含多个object
+ name	否	string	项目名称
+ province	否	string	地域
+ liandate	否	string	决定日期
+ caseno	否	string	决定文书号
pledges	是	array	股权出质，每个数组可能包含多个object
+ registno	否	string	质权登记编号
+ pledgor	否	string	出质人
+ pledgorno	否	string	出质人证照编号
+ pledgee	否	string	质权人
+ pledgeeno	否	string	质权人证照编号
+ pledgedamount	否	string	出质股权数额
+ regdate	否	string	股权出质设立登记日期
+ publicdate	否	string	公示时间
+ status	否	string	出质状态
spotchecks	是	array	企业抽查检查，每个数组可能包含多个object
+ no	否	string	登记编号
+ executiveorg	否	string	检查实施机关
+ type	否	string	类型
+ date	否	string	日期
+ consequence	否	string	结果
+ remark	否	string	备注
shixinitems	是	array	失信，每个数组可能包含多个object
+ iname	否	string	公司名称
+ regdate	否	string	立案日期
+ casecode	否	string	立案文书号
+ cardnum	否	string	组织机构代码
+ gistcid	否	string	执行依据文号

+ gistcid	否	string	执行依据文号
+ publishdate	否	string	发布时间
+ performance	否	string	被执行人的履约情况
+ disreputtypename	否	string	行为备注
+ courtname	否	string	执行法院
zhixingitems	是	array	被执行，每个数组可能包含多个object
+ casestate	否	string	状态
+ partycardnum	否	string	身份证号码/组织机构代码
+ zxid	否	string	官网系统ID
+ pname	否	string	名称
+ casecreatetime	否	string	立案时间
+ casecode	否	string	立案号
+ execcourtname	否	string	执行法院
+ execmoney	否	string	标的

返回示例


```
{
  "words_result_num": 18,
  "words_result": {
    "liquidation": {},
    "penalties": [
      {
        "officename": "国家市场监督管理总局",
        "penaltytype": "",
        "remark": "",
        "penaltydate": "2021-11-13 00:00:00",
        "docno": "国市监处〔2021〕116号",
        "publicdate": "",
        "content": ""
      }
    ],
    "shixinitems": [],
    "originalname": [],
    "changes": [
      {
        "changebefore": "21",
        "changeafter": "",
        "changefield": "营业期限",
        "changedate": "2019-12-05 00:00:00"
      },
      {
        "changebefore": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告。依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。",
        "changeafter": "销售第三类医疗器械。开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品、医疗器械II类；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告；软件开发；销售第三类医疗器械以及依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。",
        "changefield": "经营范围",
        "changedate": "2019-02-11 00:00:00"
      },
      {
        "changebefore": "1 向海龙 总经理",
        "changeafter": "None",
        "changefield": "董事（理事）、经理、监事"
```

```
            "changefield": "董事（理事）、经理、监事",
            "changedate": "2019-06-25 00:00:00"
        },
        {
            "changebefore": "崔珊珊*,韦方",
            "changeafter": "崔珊珊*,梁志祥",
            "changefield": "董事（理事）、经理、监事",
            "changedate": "2020-11-27 00:00:00"
        },
        {
            "changebefore": "沈皓瑜",
            "changeafter": "王湛",
            "changefield": "法定代表人",
            "changedate": "2011-07-28 00:00:00"
        },
        {
            "changebefore": "1 王湛 总经理",
            "changeafter": "1 向海龙 总经理",
            "changefield": "董事（理事）、经理、监事",
            "changedate": "2016-05-30 00:00:00"
        },
        {
            "changebefore": "向海龙*",
            "changeafter": "崔珊珊*,韦方",
            "changefield": "董事（理事）、经理、监事",
            "changedate": "2019-06-25 00:00:00"
        },
        {
            "changebefore": "销售第三类医疗器械。开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品、医疗器械II类；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告；软件开发；销售第三类医疗器械以及依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。",
            "changeafter": "销售第三类医疗器械。开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告；软件开发；技术开发、技术推广、技术转让；销售自产产品、医疗器械II类、电子产品、器件和元件、计算机、软件及辅助设备、灯具、五金交电、自行开发后的产品；计算机系统集成；委托加工生产通讯设备。销售第三类医疗器械以及依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。",
            "changefield": "经营范围",
            "changedate": "2019-12-05 00:00:00"
        },
        {
            "changebefore": "开发、生产计算机软件；自产产品的技术咨询、技术服务；承接计算机网络系统工程；销售自产产品；货物进出口；技术进出口；代理进出口。",
            "changeafter": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品；货物进出口；技术进出口；代理进出口。（未取得行政许可的项目除外）",
            "changefield": "经营范围",
            "changedate": "2012-07-11 00:00:00"
        },
        {
            "changebefore": "",
            "changeafter": "",
            "changefield": "营业期限",
            "changedate": "2019-12-05 00:00:00"
        },
        {
            "changebefore": "开发、生产计算机软件；自产产品的技术咨询、技术服务；承接计算机网络系统工程；销售自产产品。",
            "changeafter": "开发、生产计算机软件；自产产品的技术咨询、技术服务；承接计算机网络系统工程；销售自产产品；货物进出口；技术进出口；代理进出口。",
            "changefield": "经营范围",
            "changedate": "2011-07-28 00:00:00"
        },
        {
```

```
            "changebefore": "1 沈皓瑜 总经理",
            "changeafter": "1 王湛 总经理",
            "changefield": "董事（理事）、经理、监事",
            "changedate": "2011-07-28 00:00:00"
        },
        {
            "changebefore": "王湛*",
            "changeafter": "向海龙*",
            "changefield": "董事（理事）、经理、监事",
            "changedate": "2016-05-30 00:00:00"
        },
        {
            "changebefore": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告。（依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。）",
            "changeafter": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品、医疗器械II类；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告；软件开发；销售第三类医疗器械。（销售第三类医疗器械以及依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。）",
            "changefield": "经营范围",
            "changedate": "2019-02-11 00:00:00"
        },
        {
            "changebefore": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品；货物进出口；技术进出口；代理进出口。（依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。）",
            "changeafter": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告。（依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。）",
            "changefield": "经营范围",
            "changedate": "2016-08-26 00:00:00"
        },
        {
            "changebefore": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品、医疗器械II类；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告；软件开发；销售第三类医疗器械。（销售第三类医疗器械以及依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。）",
            "changeafter": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告；软件开发；技术开发、技术推广、技术转让；销售自产产品、医疗器械II类、电子产品、器件和元件、计算机、软件及辅助设备、灯具、五金交电、自行开发后的产品；计算机系统集成；委托加工生产通讯设备；销售第三类医疗器械。（销售第三类医疗器械以及依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。）",
            "changefield": "经营范围",
            "changedate": "2019-12-05 00:00:00"
        },
        {
            "changebefore": "王湛",
            "changeafter": "向海龙",
            "changefield": "法定代表人",
            "changedate": "2016-05-30 00:00:00"
        },
        {
            "changebefore": "1 向海龙 总经理",
            "changeafter": "None",
            "changefield": "董事（理事）、经理、监事",
            "changedate": "2019-06-25 00:00:00"
        },
        {
            "changebefore": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；销售自产产品；货物进出口；技术进出口；代理进出口。依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。",
            "changeafter": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工
```

```
程；销售自产产品；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告。依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动。",
            "changefield": "经营范围",
            "changedate": "2016-08-26 00:00:00"
        }
    ],
    "industry": {
        "industry": "信息传输、软件和信息技术服务业",
        "subindustry": "互联网和相关服务"
    },
    "branches": [
        {
            "companycode": "",
            "companyname": "百度在线网络技术（北京）有限公司天津和平区分公司",
            "authority": "天津市市场监督管理委员会",
            "legalperson": "郭丽霞",
            "creditno": "91120000MA06X3RTXX"
        },
        {
            "companycode": "",
            "companyname": "百度在线网络技术（北京）有限公司天津分公司",
            "authority": "天津市市场监督管理委员会",
            "legalperson": "郭丽霞",
            "creditno": "91120000MA06X2UT6T"
        },
        {
            "companycode": "",
            "companyname": "百度在线网络技术（北京）有限公司上海宝山分公司",
            "authority": "上海市市场监督管理局",
            "legalperson": "尚国斌",
            "creditno": "91310000MA1GNWYG37"
        },
        {
            "companycode": "310115500129605",
            "companyname": "百度在线网络技术（北京）有限公司上海第二分公司",
            "authority": "自由贸易试验区市场监督管理局",
            "legalperson": "沈抖",
            "creditno": "91310115093500281H"
        },
        {
            "companycode": "310000500149803",
            "companyname": "百度在线网络技术（北京）有限公司上海软件技术分公司",
            "authority": "上海市市场监督管理局",
            "legalperson": "张宏宇",
            "creditno": "91310000772120643P"
        },
        {
            "companycode": "310000500552158",
            "companyname": "百度在线网络技术（北京）有限公司上海第一分公司",
            "authority": "上海市市场监督管理局",
            "legalperson": "张宏宇",
            "creditno": "9131000031235406XP"
        },
        {
            "companycode": "800868",
            "companyname": "百度在线网络技术（北京）有限公司上海分公司",
            "authority": "",
            "legalperson": "何涌",
            "creditno": ""
        },
        {
            "companycode": "310115500011975",
```

```
            "companyname": "百度在线网络技术（北京）有限公司上海分事处",
            "authority": "",
            "legalperson": "任旭阳",
            "creditno": ""
        },
        {
            "companycode": "企独鄂武分字第005603号",
            "companyname": "百度在线网络技术（北京）有限公司武汉分公司",
            "authority": "",
            "legalperson": "周品",
            "creditno": ""
        },
        {
            "companycode": "",
            "companyname": "百度在线网络技术（北京）有限公司成都分公司",
            "authority": "成都市市场监督管理局",
            "legalperson": "喻友平",
            "creditno": "91510100MA65UY1Q8L"
        },
        {
            "companycode": "",
            "companyname": "百度在线网络技术（北京）有限公司邯郸分公司",
            "authority": "邯郸市市场监督管理局",
            "legalperson": "李飞",
            "creditno": "91130400MA7BMR8G5T"
        },
        {
            "companycode": "",
            "companyname": "百度在线网络技术（北京）有限公司广州分公司",
            "authority": "广州市市场监督管理局",
            "legalperson": "喻友平",
            "creditno": "91440100MAC0MJXR1P"
        }
    ],
    "exceptions": [],
    "mpledges": [],
    "spotchecks": [
        {
            "date": "2017-10-22 00:00:00",
            "no": "1",
            "consequence": "正常",
            "remark": "",
            "type": "抽查",
            "executiveorg": "北京市工商行政管理局海淀分局"
        }
    ],
    "partners": [
        {
            "shoulddate": "2019-12-05 00:00:00",
            "stockname": "百度控股有限公司",
            "investtype": "货币",
            "capidate": "",
            "stocktype": "企业法人",
            "stockrealcapital": "",
            "stockcapital": "4520.0",
            "investname": "",
            "stockpercent": "1.0000"
        }
    ],
    "zhixingitems": [],
    "permissions": [
        {
            "province": "",
```

```
            "province": "",
            "name": "高新技术企业认定证书",
            "liandate": "",
            "caseno": ""
        },
        {
            "province": "",
            "name": "高新技术企业认定证书",
            "liandate": "",
            "caseno": ""
        },
        {
            "province": "",
            "name": "高新技术企业认定证书",
            "liandate": "",
            "caseno": ""
        },
        {
            "province": "",
            "name": "软件企业认定证书",
            "liandate": "",
            "caseno": ""
        },
        {
            "province": "",
            "name": "开户许可证",
            "liandate": "",
            "caseno": ""
        },
        {
            "province": "",
            "name": "建设项目用地预审意见",
            "liandate": "",
            "caseno": ""
        },
        {
            "province": "",
            "name": "生产建设项目水土保持设施竣工验收",
            "liandate": "",
            "caseno": ""
        },
        {
            "province": "",
            "name": "建设项目用地预审意见",
            "liandate": "",
            "caseno": ""
        },
        {
            "province": "",
            "name": "行政许可决定书",
            "liandate": "2019-07-18 00:00:00",
            "caseno": "JY31108200076349"
        }
    ],
    "contactinfo": {
        "website": [],
        "phonenumber": "010-59928888",
        "email": "service-center@baidu.com"
    },
    "pledges": [],
    "employees": [
        {
            "position": "总经理,执行董事",
```

```
            "employeename": "崔珊珊"
        },
        {
            "position": "监事",
            "employeename": "梁志祥"
        }
    ],
    "base": {
        "isonstock": "0",
        "capital": "4520.000000万美元",
        "orgcode": "717743469",
        "stocknumber": "",
        "companystatus": "存续（在营、开业、在册）",
        "issuedate": "2020-11-27 00:00:00",
        "companyaddress": "北京市海淀区上地十街10号百度大厦三层",
        "keyno": "867bef5d172cbae117a5986883cf5c0e",
        "companycode": "110000410144104",
        "revokedate": "",
        "province": "BJ",
        "businessscope": "开发、生产计算机软件；提供相关技术咨询、技术服务、技术培训；承接计算机网络系统工程；货物进出口、技术进出口、代理进出口；设计、制作、代理、发布广告；软件开发；技术开发、技术推广、技术转让；销售自产产品、医疗器械II类、电子产品、器件和元件、计算机、软件及辅助设备、灯具、五金交电、自行开发后的产品；计算机系统集成；委托加工生产通讯设备；销售第三类医疗器械。（市场主体依法自主选择经营项目，开展经营活动；销售第三类医疗器械以及依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动；不得从事国家和本市产业政策禁止和限制类项目的经营活动。）",
        "businessdatefrom": "2000-01-18 00:00:00",
        "stocktype": "",
        "companyname": "百度在线网络技术（北京）有限公司",
        "companytype": "有限责任公司(外国法人独资)",
        "authority": "北京市海淀区市场监督管理局",
        "legalperson": "崔珊珊",
        "creditno": "91110108717743469K",
        "establishdate": "2000-01-18 00:00:00",
        "businessdateto": ""
    },
    "taxcredititems": [
        {
            "taxpayerno": "110108717743469",
            "year": "2015",
            "level": "A",
            "taxpayername": "百度在线网络技术（北京）有限公司"
        },
        {
            "taxpayerno": "91110108717743469K",
            "year": "2018",
            "level": "A",
            "taxpayername": "百度在线网络技术（北京）有限公司"
        },
        {
            "taxpayerno": "91110108717743469K",
            "year": "2019",
            "level": "A",
            "taxpayername": "百度在线网络技术（北京）有限公司"
        },
        {
            "taxpayerno": "91110108717743469K",
            "year": "2020",
            "level": "A",
            "taxpayername": "百度在线网络技术（北京）有限公司"
        }
    ]
},
```

```
    "log_id": 1585110490379986830
}
```

企业二要素核验

接口描述

通过核验企业名称、统一社会信用代码一致性，快速核验企业资质。

注：目前支持工商注册的企业认证，事业单位、律师事务所等信息暂不支持验证，发生工商变更或刚注册的企业信息，预计在企业公示的核准日期T+3个工作日后可进行验证。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/two_factors_verification`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
company	是	string	-	企业名称
regnum	是	string	-	社会统一信用代码

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

bash

python

java

c++

php

c#

OCR-企业二要素核验

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/two_factors_verification?access_token=【调用鉴权接口获取的token】' --data 'company=企业名称&regnum=社会统一信用代码' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
verifyresult	是	string	返回值为: -**1**：（二要素完全匹配） -**0**：（二要素不完全匹配）
companymatch	是	string	企业名称， 1：匹配 0：不匹配 2：无法验证
regnummatch	是	string	统一社会信用代码， 1：匹配 0：不匹配 2：无法验证

返回示例

```
{
  "words_result_num": 3,
  "words_result": {
     "verifyresult": "1",
     "companymatch": "1",
     "regnummatch": "1"
  },
  "log_id": 1585111579921862445
}
```

企业三要素核验

接口描述

通过核验企业名称、统一社会信用代码、法人姓名一致性，快速核验企业资质。

注：目前支持工商注册的企业认证，事业单位、律师事务所等信息暂不支持验证，发生工商变更或刚注册的企业信息，预计在企业公示的核准日期T+3个工作日后可进行验证。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/three_factors_verification`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
name	是	string	-	法人姓名
company	是	string	-	企业名称
regnum	是	string	-	社会统一信用代码

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
 OCR-企业三要素核验
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/three_factors_verification?access_token=【调用鉴权接口获取的token】' --data 'name=法人姓名&company=企业名称&regnum=社会统一信用代码' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
verifyresult	是	string	返回值为: **-1：**（三要素完全匹配） **-0：**（三要素不完全匹配）
namematch	是	string	法人姓名， 1：匹配 0：不匹配 2：无法验证
companymatch	是	string	企业名称， 1：匹配 0：不匹配 2：无法验证
regnummatch	是	string	统一社会信用代码， 1：匹配 0：不匹配 2：无法验证

返回示例

```
{
  "words_result_num": 4,
  "words_result": {
     "verifyresult": "1",
     "namematch": "1",
     "companymatch": "1",
     "regnummatch": "1"
  },
  "log_id": 1585111579921862445
}
```

企业四要素核验

接口描述

比对校验企业名称、统一社会信用代码、法人姓名、注册证件号的一致性，验证企业工商信息。

注：目前支持工商注册、组织机构代码中心登记的企业认证，包括事业单位、律师事务所等，发生工商变更或刚注册的企业信息，预计在企业公示的核准日期T+15个工作日后可进行验证。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/four_factors_verification`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
name	是	string	-	法人姓名
idcard	是	string	-	法人证件号码，如身份证等
company	是	string	-	企业名称
regnum	是	string	-	社会统一信用代码

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
 OCR-企业四要素核验
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/four_factors_verification?access_token=【调用鉴权接口获取的token】' --data 'idcard=注册证件号&name=法人姓名&company=企业名称&regnum=社会统一信用代码' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
verifyresult	是	string	返回值为: -**1**：（四要素完全匹配） -**0**：（四要素不完全匹配）
namematch	是	string	法人姓名， 1：匹配 0：不匹配 2：无法验证
idnummatch	是	string	注册证件号， 1：匹配 0：不匹配 2：无法验证
companymatch	是	string	企业名称， 1：匹配 0：不匹配 2：无法验证
regnummatch	是	string	统一社会信用代码， 1：匹配 0：不匹配 2：无法验证

返回示例


```
{
  "words_result_num": 5,
  "words_result": {
     "verifyresult": "0",
     "idnummatch": "0",
     "namematch": "1",
     "companymatch": "1",
     "regnummatch": "1"
  },
  "log_id": 1584898210595034428
}
```


港澳台证件识别

接口描述

支持识别4类港澳台出入境证件，包含港澳通行证正/反面、台湾通行证正/反面、台胞证（台湾居民来往大陆通行证）正/反面、返乡证（港澳居民来往内地通行证）正/反面，支持识别以上4类证件的全部字段信息。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/hk_macau_taiwan_exitentrypermit`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
exitentrypermit_type	是	string	hk_mc_passport_front：港澳通行证正面 hk_mc_passport_back：港澳通行证反面 tw_passport_front：台湾通行证正面 tw_passport_back：台湾通行证反面 tw_return_passport_front：台胞证正面 tw_return_passport_back：台胞证反面 hk_mc_return_passport_front：返乡证正面 hk_mc_return_passport_back：返乡证反面	出入境许可证件的具体类型
probability	否	string	true/false	是否返回字段置信度，默认为 false ，即不返回
location	否	string	true/false	是否返回字段位置坐标，默认为 false，即不返回

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

```
Bash
```

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/hk_macau_taiwan_exitentrypermit?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】&exitentrypermit_type=tw_return_passport_front' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

1、港澳通行证正面

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ card_number	是	array[]	证件号码
+ name_chn	是	array[]	姓名
+ name_eng	是	array[]	姓名（英文）
+ birthday	是	array[]	出生日期
+ sex	是	array[]	性别
+ valid_date	是	array[]	有效期限
+ issue_authority	是	array[]	签发机关
+ issue_place	是	array[]	签发地点
+ MRZCode	是	array[]	证件下方第一行
++ word	是	string	字段识别结果，以上字段均包含
++ location	否	object{}	字段位置信息，当请求参数 location=true 时返回该字段，以上字段均包含
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object{}	字段识别结果置信度，当请求参数 probability=true 时返回该字段，以上字段均包含
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

2、港澳通行证反面

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ hk_type	是	array[]	来往香港签注-种类
+ hk_valid_date	是	array[]	来往香港签注-有效期
+ hk_remarks	是	array[]	来往香港签注-备注
+ hk_round_trip_number	是	array[]	来往香港签注-往返有效
+ mc_type	是	array[]	来往澳门签注-种类
+ mc_valid_date	是	array[]	来往澳门签注-有效期
+ mc_remarks	是	array[]	来往澳门签注-备注
+ mc_round_trip_number	是	array[]	来往澳门签注-往返有效
++ word	是	string	字段识别结果，以上字段均包含
++ location	否	object{}	字段位置信息，当请求参数 location=true 时返回该字段，以上字段均包含
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object{}	字段识别结果置信度，当请求参数 probability=true 时返回该字段，以上字段均包含
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

3、台湾通行证正面

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ card_number	是	array[]	证件号码
+ name_chn	是	array[]	姓名
+ name_eng	是	array[]	姓名（英文）
+ birthday	是	array[]	出生日期
+ sex	是	array[]	性别
+ valid_date	是	array[]	有效期限
+ issue_authority	是	array[]	签发机关
+ issue_place	是	array[]	签发地点
+ MRZCode	是	array[]	证件下方第一行
++ word	是	string	字段识别结果，以上字段均包含
++ location	否	object{}	字段位置信息，当请求参数 location=true 时返回该字段，以上字段均包含
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object{}	字段识别结果置信度，当请求参数 probability=true 时返回该字段，以上字段均包含
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

4、台湾通行证反面

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ type	是	array[]	种类
+ valid_date	是	array[]	有效期
+ remarks	是	array[]	备注
+ round_trip_number	是	array[]	往返有效
++ word	是	string	字段识别结果，以上字段均包含
++ location	否	object{}	字段位置信息，当请求参数 location=true 时返回该字段，以上字段均包含
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object{}	字段识别结果置信度，当请求参数 probability=true 时返回该字段，以上字段均包含
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

5、返乡证正面

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ name_chn	是	array[]	姓名
+ name_eng	是	array[]	姓名（英文）
+ birthday	是	array[]	出生日期
+ sex	是	array[]	性别
+ valid_date	是	array[]	有效期限
+ issue_authority	是	array[]	签发机关
+ issue_times	是	array[]	签发次数
+ card_number	是	array[]	证件号码
++ word	是	string	字段识别结果，以上字段均包含
++ location	否	object{}	字段位置信息，当请求参数 location=true 时返回该字段，以上字段均包含
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object{}	字段识别结果置信度，当请求参数 probability=true 时返回该字段，以上字段均包含
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

6、返乡证反面

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ idcard_name	是	array[]	身份证姓名
+ idcard_number	是	array[]	身份证号码
+ MRZCode1	是	array[]	证件下方第一行
+ MRZCode2	是	array[]	证件下方第二行
++ word	是	string	字段识别结果，以上字段均包含
++ location	否	object{}	字段位置信息，当请求参数 location=true 时返回该字段，以上字段均包含
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object{}	字段识别结果置信度，当请求参数 probability=true 时返回该字段，以上字段均包含
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

7、台胞证正面

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ name_chn	是	array[]	姓名
+ name_eng	是	array[]	姓名（英文）
+ birthday	是	array[]	出生日期
+ sex	是	array[]	性别
+ valid_date	是	array[]	有效期限
+ issue_authority	是	array[]	签发机关
+ issue_place	是	array[]	签发地点
+ card_number	是	array[]	证件号码
+ issue_times	是	array[]	签发次数
++ word	是	string	字段识别结果，以上字段均包含
++ location	否	object{}	字段位置信息，当请求参数 location=true 时返回该字段，以上字段均包含
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object{}	字段识别结果置信度，当请求参数 probability=true 时返回该字段，以上字段均包含
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

8、台胞证反面

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ idcard_name	是	array[]	身份证姓名
+ idcard_number	是	array[]	身份证号码
+ MRZCode1	是	array[]	证件下方第一行
+ MRZCode2	是	array[]	证件下方第二行
++ word	是	string	字段识别结果，以上字段均包含
++ location	否	object{}	字段位置信息，当请求参数 location=true 时返回该字段，以上字段均包含
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object{}	字段识别结果置信度，当请求参数 probability=true 时返回该字段，以上字段均包含
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

返回示例

示例一：港澳通行证正面


```
{
  "words_result_num": 9,
  "words_result": {
     "card_number": [
        {
           "word": "C00000000"
        }
     ],
     "name_chn": [
        {
           "word": "证件样本"
        }
     ],
     "name_eng": [
        {
           "word": "ZHENGJIAN,YANGBEN"
        }
     ],
     "birthday": [
        {
           "word": "1981.08.03"
        }
     ],
     "sex": [
        {
           "word": "女"
        }
     ],
     "valid_date": [
        {
```

```
            "word": "2014.04.21-2024.04.20"
          }
        ],
        "issue_authority": [
          {
            "word": "公安部出入境管理局"
          }
        ],
        "issue_place": [
          {
            "word": "广东"
          }
        ],
        "MRZCode": [
          {
            "word": "CSC000000004<2404200<8108038<6"
          }
        ]
    },
    "direction": 0,
    "log_id": 1645241609425709013
}
```


结婚证识别

接口描述

支持对结婚证进行结构化识别，包括**姓名_男**、**身份证件号_男**、**出生日期_男**、**国籍_男**、**性别_男**、**姓名_女**、**身份证件号_女**、**出生日期_女**、**国籍_女**、**性别_女**、**结婚证字号**、**持证人**、**备注**、**登记日期**，全部 14 个字段。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/#/api?product=AI&project=文字识别&parent=卡证OCR&api=rest/2.0/ocr/v1/marriage_certificate&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/marriage_certificate`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

| 参数 | 是否必选 | 类型 | 可选值范围 | 说明 |

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	string	true/false	是否返回字段置信度，默认为 false ，即不返回
location	否	string	true/false	是否返回字段位置坐标，默认为 false，即不返回

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/marriage_certificate?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
##### encoding:utf-8

import requests
import base64

'''
结婚证识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/marriage_certificate"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 结婚证识别
*/
public class MarriageCertificate{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String marriageCertificate() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/marriage_certificate";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        marriageCertificate.marriageCertificate();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/marriage_certificate";
static std::string marriageCertificate_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    marriageCertificate_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 结婚证识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int marriageCertificate(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = marriageCertificate_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/marriage_certificate?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class marriageCertificate
    {
        // 结婚证识别
        public static string marriageCertificate()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/marriage_certificate?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("结婚证识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

```
 ###### 返回说明

**返回参数**

| 字段              | 是否必选 | 类型       | 说明
|
| ---------------- | ---- | -------- | ----------------------
```

_男**、**性别_男**、**姓名_女**、**身份证件号_女**、**出生日期_女**、**国籍_女**、**性别_女**、**结婚证字号、持证人、备注、登记日期** 14 个字段的识别结果 |
| + location | 否 | object{} | 字段位置信息，当请求参数 location=true 时返回该字段
|
| ++ top | 否 | uint32 | 字段的上边距
|
| ++ left | 否 | uint32 | 字段的左边距
|
| ++ height | 否 | uint32 | 字段的高度
|
| ++ width | 否 | uint32 | 字段的宽度
|
| + probability | 否 | object{} | 字段识别结果置信度，当请求参数 probability=true 时返回该字段
|
| ++ average | 否 | float | 字段识别结果中各字符的置信度平均值
|
| ++ min | 否 | float | 字段识别结果中各字符的置信度最小值
|

返回示例

```JSON
{
   "words_result_num": 14,
   "words_result": {
      "姓名_男": [
         {
            "word": "李佑"
         }
      ],
      "身份证件号_男": [
         {
            "word": "433024197905103103"
         }
      ],
      "出生日期_男": [
         {
            "word": "1979-05-10"
         }
      ],
      "国籍_男": [
         {
            "word": "中国"
         }
      ],
      "性别_男": [
         {
            "word": "男"
         }
      ],
      "姓名_女": [
         {
            "word": "刘美"
         }
      ],
      "身份证件号_女": [
         {
            "word": "433024197609160160"
         }
      ],
      "出生日期_女": [
```

```
          {
            "word": "1976-09-16"
          }
        ],
        "国籍_女": [
          {
            "word": "中国"
          }
        ],
        "性别_女": [
          {
            "word": "女"
          }
        ],
        "结婚证字号": [
          {
            "word": "怀溆结字011004100号"
          }
        ],
        "持证人": [
          {
            "word": "李佑"
          }
        ],
        "备注": [
          {
            "word": ""
          }
        ],
        "登记日期": [
          {
            "word": "2010-06-10"
          }
        ]
    },
    "log_id": 1645244176883578313
}
```


房产证识别

接口描述

支持对房产证进行结构化识别，包括权利人、坐落、权利类型、面积、字第号 、不动产单元号、共有情况、用途、使用期限 、登记日期、共有人，全部 11 个字段。

在线调试

您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=%E6%96%87%E5%AD%97%E8%AF%86%E5%88%AB&parent=%E5%8D%A1%E8%AF%81OCR&api=rest%) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/real_estate_certificate`

URL参数：

参数	值

| ------------ | -- |
| access_token | 通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)” |

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	string	true/false	是否返回字段置信度，默认为 false ，即不返回
location	否	string	true/false	是否返回字段位置坐标，默认为 false，即不返回

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/real_estate_certificate?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
房产证识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/real_estate_certificate"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
```
~~~

```
    print (response.json())

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 房产证识别
*/
public class RealEstateCertificate{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String realEstateCertificate() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/real_estate_certificate";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        realEstateCertificate.realEstateCertificate();
    }
}
```
```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
```

```
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/real_estate_certificate";
static std::string realEstateCertificate_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    realEstateCertificate_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 房产证识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int realEstateCertificate(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = realEstateCertificate_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
```

```
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/real_estate_certificate?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class realEstateCertificate
    {
        // 房产证识别
        public static string realEstateCertificate()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/real_estate_certificate?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
```

```
            Console.WriteLine("房产证识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ word	是	string	字段识别结果，对应 **权利人、坐落、权利类型、面积、字第号 、不动产单元号、共有情况、用途、使用期限 、登记日期、共有人** 11 个字段的识别结果
+ location	否	object{}	字段位置信息，当请求参数 location=true 时返回该字段
++ top	否	uint32	字段的上边距
++ left	否	uint32	字段的左边距
++ height	否	uint32	字段的高度
++ width	否	uint32	字段的宽度
+ probability	否	object{}	字段识别结果置信度，当请求参数 probability=true 时返回该字段
++ average	否	float	字段识别结果中各字符的置信度平均值
++ min	否	float	字段识别结果中各字符的置信度最小值

返回示例


```
{
  "words_result_num": 11,
  "words_result": {
    "权利人": [
      {
        "word": "阮兴武"
      }
```

```
        ],
        "坐落": [
          {
            "word": "吉水县乌江镇前江村丰山组"
          }
        ],
        "权利类型": [
          {
            "word": "宅基地使用权/房屋(构筑物)所有权"
          }
        ],
        "面积": [
          {
            "word": "土地使用权面积:115.720㎡/房屋建筑面积:298.520㎡"
          }
        ],
        "字第号": [
          {
            "word": "0042537"
          }
        ],
        "不动产单元号": [
          {
            "word": "360822"
          }
        ],
        "共有情况": [
          {
            "word": "家庭成员共有"
          }
        ],
        "用途": [
          {
            "word": "农村宅基地/住宅"
          }
        ],
        "使用期限": [
          {
            "word": ""
          }
        ],
        "登记日期": [
          {
            "word": ""
          }
        ],
        "共有人": [
          {
            "word": ""
          }
        ]
    },
    "log_id": 1739493844726379007
}
```


开户许可证识别

接口描述

支持对开户许可证进行结构化识别，包括公司名称、开户银行、核准号、法人、编号、账号，全部 6 个字段。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/account_opening`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/account_opening?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
开户许可证识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/account_opening"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 开户许可证识别
*/
public class AccountOpening{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String accountOpening() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/account_opening";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        accountOpening.accountOpening();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/account_opening";
static std::string accountOpening_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    accountOpening_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 开户许可证识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int accountOpening(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = accountOpening_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/account_opening?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class accountOpening
    {
        // 开户许可证识别
        public static string accountOpening()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/account_opening?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("开户许可证识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ word	是	string	字段识别结果，对应 **公司名称、开户银行、核准号、法人、编号、账号** 6 个字段的识别结果

返回示例

```JSON
{
   "words_result": {
      "账号": {
         "word": [
            "254653912299"
         ]
      },
      "公司名称": {
         "word": [
            "河南专开机械设备有限公司"
         ]
      },
      "核准号": {
         "word": [
            "J4910057237101"
         ]
      },
      "法人": {
         "word": [
            "郭娟"
         ]
      },
      "编号": {
         "word": [
            "4910-02691732"
         ]
      },
      "开户银行": {
         "word": [
            "中国银行股份有限公司郑州高新技术开发区支行"
         ]
      }
   },
   "words_result_num": 6,
   "log_id": 1754840007553369396
}
```

外国人永久居住证识别

接口描述

支持对外国人永久居住证进行结构化识别，识别字段包括 Name、Nationality、Sex、出生日期、国籍、失效日期、姓名、性别、签发日期、证件号码、证件版本，全部 11 个字段。

申请试用
该接口正在邀测中，在正式使用之前，请先提交[合作咨询](https://ai.baidu.com/consultation/cooperation?&referrerUrl=/tech/ocr_cards/idcard)，或者[提交工单](https://console.bce.baidu.com/ticket/?_=1710300394181#/ticket/create?productId=96)，提供公司名称、appid、应用场景，工作人员协助开通权限后方可使用。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/foreign_resident_id_card`

URL参数：

参数	值

| ---------- | ---------- |
| access_token | 通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)” |

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/foreign_resident_id_card?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
外国人永久居住证识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/foreign_resident_id_card"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 外国人永久居住证识别
*/
public class ForeignResidentIDCard{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String foreignResidentIDCard() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/foreign_resident_id_card";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        foreignResidentIDCard.foreignResidentIDCard();
    }
}
```
```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/foreign_resident_id_card";
static std::string foreignResidentIDCard_result;
```
```

```
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    foreignResidentIDCard_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 外国人永久居住证识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int foreignResidentIDCard(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = foreignResidentIDCard_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }
```

```
    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/foreign_resident_id_card?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class foreignResidentIDCard
    {
        // 外国人永久居住证识别
        public static string foreignResidentIDCard()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/foreign_resident_id_card?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("外国人永久居住证识别:");
            Console.WriteLine(result);
```

```
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ word	是	string	字段识别结果，对应 **Name、Nationality、Sex、出生日期、国籍、失效日期、姓名、性别、签发日期、证件号码、证件版本** 11 个字段的识别结果

返回示例


```
{
  "words_result": {
    "姓名": [
      {
        "word": "森田"
      }
    ],
    "证件版本": [
      {
        "word": "1"
      }
    ],
    "失效日期": [
      {
        "word": "2033.06.11"
      }
    ],
    "证件号码": [
      {
        "word": "123456789"
      }
    ],
    "Sex": [
      {
```

```
            "word": "M"
        }
      ],
      "出生日期": [
        {
          "word": "1990.01.01"
        }
      ],
      "签发日期": [
        {
          "word": "2023.06.12"
        }
      ],
      "国籍": [
        {
          "word": "日本"
        }
      ],
      "Nationality": [
        {
          "word": "JPN"
        }
      ],
      "Name": [
        {
          "word": "MORITA. SHUNSUKE"
        }
      ],
      "性别": [
        {
          "word": "男"
        }
      ]
   },
   "words_result_num": 11,
   "log_id": 1767755838708432889
}
```


食品经营许可证识别

接口描述

支持对食品经营许可证进行结构化识别，包括经营者名称、社会信用代码、法定代表人、住所、经营场所、主体业态、经营项目、有效期至、许可证编号、日常监督管理机构、日常监督管理人员、发证机关、签发人、签发日期，全部 14 个字段。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/food_business_license`

URL参数：

参数	值

| access_token | 通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)” |

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/food_business_license?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
##### encoding:utf-8

import requests
import base64

'''
食品经营许可证识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/food_business_license"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 食品经营许可证识别
*/
public class FoodBusinessLicense{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String foodBusinessLicense() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/food_business_license";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        foodBusinessLicense.foodBusinessLicense();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/food_business_license";
static std::string foodBusinessLicense_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    foodBusinessLicense_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 食品经营许可证识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int foodBusinessLicense(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = foodBusinessLicense_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/food_business_license?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class foodBusinessLicense
    {
        // 食品经营许可证识别
        public static string foodBusinessLicense()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/food_business_license?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("食品经营许可证识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ word	是	string	字段识别结果，对应 **经营者名称、社会信用代码、法定代表人、住所、经营场所、主体业态、经营项目、有效期至、许可证编号、日常监督管理机构、日常监督管理人员、发证机关、签发人、签发日期** 14 个字段的识别结果

返回示例

```JSON
{
    "words_result": {
        "经营者名称": [
            {
                "word": "张三"
            }
        ],
        "经营场所": [
            {
                "word": "中华大道201号"
            }
        ],
        "日常监督管理机构": [
            {
                "word": "吴中区市场监督管理局开发区分局"
            }
        ],
        "日常监督管理人员": [
            {
                "word": "顾李根沈丽芳"
            }
        ],
        "签发日期": [
            {
                "word": "2015年12月01日"
            }
        ],
        "经营项目": [
            {
                "word": "预包装食品（不含冷藏冷冻食品）销售"
            }
        ],
        "有效期至": [
            {
                "word": "2020年11月30日"
            }
        ],
        "许可证编号": [
            {
                "word": "JY132050600000"
            }
        ],
        "法定代表人": [
            {
                "word": "黄友江"
            }
        ],
        "社会信用代码": [
            {
                "word": "320506600957"
            }
        ],
        "住所": [
            {
                "word": "江苏省苏州市吴中区越溪街道中大道"
            }
        ],
        "发证机关": [
            {
                "word": "苏州市吴中区场监督管理局"
```

```
          }
        ],
        "主体业态": [
          {
            "word": "食品销售经营者"
          }
        ],
        "签发人": [
          {
            "word": "苏征宇"
          }
        ]
    },
    "words_result_num": 14,
    "log_id": 1770091234220759935
}
```


食品生产许可证识别

接口描述

支持对食品生产许可证进行结构化识别，包括生产者名称、社会信用代码、法定代表人、住所、生产地址、食品类别、有效期至、许可证编号、日常监督管理机构、日常监督管理人员、投诉举报电话等信息、发证机关、签发人、签发日期，全部 14 个字段。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/food_product_license`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内

| pdf_file_num | 否 | string | - | 需要识别的 PDF 文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面的内容，若不传入，则默认识别第 1 页|

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/food_product_license?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
食品生产许可证识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/food_product_license"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 食品生产许可证识别
*/
public class FoodProductLicense{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String foodProductLicense() {
~~~

```
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/food_product_license";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        foodProductLicense.foodProductLicense();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/food_product_license";
static std::string foodProductLicense_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    foodProductLicense_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 食品生产许可证识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int foodProductLicense(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
```

```
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = foodProductLicense_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/food_product_license?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
```

```
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class foodProductLicense
    {
        // 食品生产许可证识别
        public static string foodProductLicense()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/food_product_license?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("食品生产许可证识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ word	是	string	字段识别结果，对应 **生产者名称、社会信用代码、法定代表人、住所、生产地址、食品类别、有效期至、许可证编号、日常监督管理机构、日常监督管理人员、投诉举报电话等信息、发证机关、签发人、签发日期** 14 个字段的识别结果

返回示例


```
{
  "words_result": {
    "日常监督管理机构": [
      {
        "word": "北京市场监督管理局"
      }
    ],
    "日常监督管理人员": [
      {
        "word": "周梦"
      }
    ],
    "签发日期": [
      {
        "word": "2020年5月31日"
      }
      ],
    "投诉举报电话等信息": [
      {
        "word": "12331"
      }
    ],
    "有效期至": [
      {
        "word": "2032年5月31日"
      }
    ],
    "许可证编号": [
      {
        "word": "SC12345678901234"
      }
    ],
    "法定代表人": [
      {
        "word": "侯小柯"
      }
    ],
    "社会信用代码": [
      {
        "word": "11072919830312065Y"
      }
    ],
```

```
        ],
        "生产地址": [
            {
                "word": "北京市海淀区中关村大街"
            }
        ],
        "住所": [
            {
                "word": "北京市海淀区中关村大街"
            }
        ],
        "食品类别": [
            {
                "word": "冷冻食品"
            }
        ],
        "发证机关": [
            {
                "word": "北京市场监督管理局"
            }
        ],
        "签发人": [
            {
                "word": "王虎项"
            }
        ],
        "生产者名称": [
            {
                "word": "北京卡顿食品有限公司"
            }
        ]
    },
    "words_result_num": 14,
    "log_id": 1770092256672897447
}
```


离婚证识别

接口描述

支持对离婚证进行结构化识别，包括**姓名_男**、**身份证件号_男**、**出生日期_男**、**国籍_男**、**性别_男**、**姓名_女**、**身份证件号_女**、**出生日期_女**、**国籍_女**、**性别_女**、**离婚证字号**、**持证人**、**备注**、**登记日期**，全部 14 个字段。

在线调试

您可以在 [示例代码中心](https://console.bce.baidu.com/tools/#/api?product=AI&project=文字识别&parent=卡证OCR&api=rest/2.0/ocr/v1/divorce_certificate&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/divorce_certificate`

URL参数：

参数	值

| access_token | 通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)” |

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/divorce_certificate?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
离婚证识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/divorce_certificate"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 离婚证识别
*/
public class DivorceCertificate{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String divorceCertificate() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/divorce_certificate";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        DivorceCertificate.divorceCertificate();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/divorce_certificate";
static std::string divorceCertificate_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    divorceCertificate_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 离婚证识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int divorceCertificate(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s
",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = divorceCertificate_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/divorce_certificate?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class divorceCertificate
    {
        // 离婚证识别
        public static string divorceCertificate()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/divorce_certificate?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("离婚证识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ word	是	string	字段识别结果，对应**姓名_男**、**身份证件号_男**、**出生日期_男**、**国籍_男**、**性别_男**、**姓名_女**、**身份证件号_女**、**出生日期_女**、**国籍_女**、**性别_女**、**离婚证字号**、**持证人**、**备注**、**登记日期** 共 14 个字段的识别结果
+ location	否	object{}	字段位置信息
++ top	否	uint32	字段的上边距

++ left	否	uint32	字段的左边距
++ height	否	uint32	字段的高度
++ width	否	uint32	字段的宽度
+ probability	否	float	字段识别结果置信度

返回示例

```JSON
{
    "words_result_num": 14,
    "words_result": {
        "姓名_男": [
            {
                "word": "李佑",
                "probability": 2.998,
                "location": {
                    "top": 423,
                    "left": 114,
                    "width": 45,
                    "height": 16
                }
            }
        ],
        "身份证件号_男": [
            {
                "word": "433024197905103103",
                "probability": 17.964,
                "location": {
                    "top": 484,
                    "left": 152,
                    "width": 138,
                    "height": 15
                }
            }
        ],
        "出生日期_男": [
            {
                "word": "1979年05月10日",
                "probability": 10.982,
                "location": {
                    "top": 465,
                    "left": 368,
                    "width": 104,
                    "height": 15
                }
            }
        ],
        "国籍_男": [
            {
                "word": "中国",
                "probability": 1.998,
                "location": {
                    "top": 454,
                    "left": 111,
                    "width": 31,
                    "height": 17
                }
            }
        ],
        "性别_男": [
            {
```

```
            "word": "男",
            "probability": 1.0,
            "location": {
                "top": 424,
                "left": 350,
                "width": 15,
                "height": 16
            }
        }
    ],
    "姓名_女": [
        {
            "word": "刘美",
            "probability": 2.287,
            "location": {
                "top": 533,
                "left": 113,
                "width": 44,
                "height": 16
            }
        }
    ],
    "身份证件号_女": [
        {
            "word": "433024197609160160",
            "probability": 17.971,
            "location": {
                "top": 593,
                "left": 153,
                "width": 138,
                "height": 14
            }
        }
    ],
    "出生日期_女": [
        {
            "word": "1976年09月16日",
            "probability": 10.989,
            "location": {
                "top": 568,
                "left": 367,
                "width": 104,
                "height": 15
            }
        }
    ],
    "国籍_女": [
        {
            "word": "中国",
            "probability": 1.999,
            "location": {
                "top": 560,
                "left": 115,
                "width": 31,
                "height": 17
            }
        }
    ],
    "性别_女": [
        {
            "word": "女",
            "probability": 1.0,
            "location": {
```

```
            "location": {
                "top": 534,
                "left": 351,
                "width": 16,
                "height": 16
            }
        }
    ],
    "登记日期": [
        {
            "word": "2010年06月10日",
            "probability": 10.983,
            "location": {
                "top": 204,
                "left": 108,
                "width": 151,
                "height": 16
            }
        }
    ],
    "离婚证字号": [
        {
            "word": "L010101-1020-001111",
            "probability": 18.981,
            "location": {
                "top": 268,
                "left": 109,
                "width": 152,
                "height": 14
            }
        }
    ],
    "持证人": [
        {
            "word": "李佑",
            "probability": 2.998,
            "location": {
                "top": 145,
                "left": 109,
                "width": 49,
                "height": 16
            }
        }
    ],
    "备注": [
        {
            "word": "备注",
            "probability": 1.998,
            "location": {
                "top": 292,
                "left": 71,
                "width": 33,
                "height": 17
            }
        }
    ]
  },
  "log_id": 1873982128764157190
}
```


交通场景文字识别

行驶证识别

接口描述

对机动车行驶证主页及副页所有22个字段进行结构化识别，包括号牌号码、车辆类型、所有人、品牌型号、车辆识别代码、发动机号码、核定载人数、质量、尺寸、检验记录等。

同时支持识别交管12123 APP 发放的电子行驶证正页及副页，包括车辆类型、号牌号码、使用性质、状态、所有人、强制报废期止、车身颜色、能源种类、外廓尺寸等25个字段。
>视频教程请参见 [行驶证识别API调用教程（视频版）](https://cloud.baidu.com/video-center/video.html?id=743)

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=交通场景OCR&api=rest/2.0/ocr/v1/vehicle_license&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_license`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
detect_direction	否	string	true/false	- **false：默认值**不进行图像方向自动矫正</br> - **true:** 开启图像方向自动矫正功能，可对旋转 90/180/270 度的图片进行自动矫正并识别
vehicle_license_side	否	string	front/back	- **front：默认值**，识别行驶证主页、电子行驶证主页</br>- **back：**识别行驶证副页、电子行驶证副页
unified	否	string	true/false	- **false：默认值**，不进行归一化处理</br>- **true：**对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出
quality_warn	否	string	true/false	是否开启质量检测功能，仅在行驶证正页识别时生效，可选值如下 -**false：默认值**，不输出质量告警信息 -**true：**在 warn_infos 输出行驶证遮挡、不完整、模糊质量告警信息。同时，可在 quality_propobility 输出质量检测置信度信息

| risk_warn | 否 | string | true/false | 是否开启风险检测功能，仅在行驶证正页识别时生效，</br>**- false：默认值**，不输出风险告警信息 </br>**- true**：开启，输出行驶证复印、翻拍、PS等告警信息 |

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_license?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
行驶证识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_license"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 行驶证识别
*/
public class VehicleLicense {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
```
~~~

```
     */
    public static String vehicleLicense() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_license";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        VehicleLicense.vehicleLicense();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_license";
static std::string vehicleLicense_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    vehicleLicense_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 行驶证识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int vehicleLicense(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
```

```
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = vehicleLicense_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_license?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
```

```
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class VehicleLicense
    {
        // 行驶证识别
        public static string vehicleLicense()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_license?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("行驶证识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```
```

返回说明

返回参数
```

| 字段 | 必选 | 类型 | 说明 |
| --- | --- | --- | --- |
| log_id | 是 | uint64 | 唯一的log id，用于问题定位 |
| direction | 否 | int32 | 图像方向，当 detect_direction=true 时返回该字段。<br>- -1：未定义，<br>- 0：正向，<br>- 1：逆时针90度，<br>- 2：逆时针180度，<br>- 3：逆时针270度 |
| words_result_num | 是 | uint32 | 识别结果数，表示words_result的元素个数 |
| words_result | 是 | object | 识别结果 |
| + words | 否 | string | 识别结果字符串 |
| warn_infos | 否 | array[] | 质量告警信息，当请求参数 vehicle_license_side=front 且 quality_warn=true 时输出，<br>- shield：行驶证证照存在遮挡告警提示<br>- incomplete：行驶证证照边框不完整告警提示<br>- fuzzy：行驶证证照存在模糊告警提示 |
| quality_propobility | 否 | object | 质量检测置信度，当请求参数 vehicle_license_side=front 且 quality_warn=true 时输出 |
| + is_clear_propobility | 是 | string | “是否清晰”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 is_clear_propobility 超过 0.5 时，对应 warn_infos 返回 fuzzy，低于 0.5，则不返回 fuzzy |
| + is_complete_propobility | 是 | string | “是否边框/四角完整”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 is_complete_propobility 超过 0.5 时，对应 warn_infos 返回 incomplete，低于0.5，则不返回 incomplete |
| + is_noshield_propobility | 是 | string | “是否被遮挡”质量类型对应的概率，值取0或1，0代表图像被遮挡，1代表图像没有被遮挡 |
| risk_type | 否 | string | 当输入参数 risk_warn=true 时输出，<br>- normal：正常行驶证<br>- copy：复印件<br>- screen：翻拍 |
| edit_tool | 否 | string | 当输入参数 risk_warn=true 时返回，如果检测行驶证被编辑过，该字段指定编辑软件名称，如：Adobe Photoshop CC 2014 (Macintosh)，如果没有被编辑过则返回值为空 |

**返回示例（行驶证正页）**

```
{
"words_result": {
 "车辆识别代号": {
  "words": "SSVUDDTT2J2022558"
 },
 "住址": {
  "words": "中牟县三刘寨村"
 },
 "发证日期": {
  "words": "20180313"
 },
      "发证单位": {
  "words": "北京市公安局公安交通管理局"
 },
 "品牌型号": {
  "words": "大众汽车牌SVW6474DFD"
 },
 "车辆类型": {
  "words": "小型普通客车"
 },
 "所有人": {
  "words": "郑昆"
 },
 "使用性质": {
  "words": "非营运"
 },
 "发动机号码": {
  "words": "111533"
 },
 "号牌号码": {
  "words": "豫A99RR9"
 },
 "注册日期": {
  "words": "20180312"
 }
},
"log_id": "1290127183406170112",
"words_result_num": 11
}
```


返回示例（电子行驶证正页）

```
{
  "words_result": {
    "车辆识别代号": {
      "words": "L2C******LG395859"
    },
    "状态": {
      "words": "正常"
    },
    "品牌型号": {
      "words": "捷豹牌CJL7203J2A6"
    },
    "车辆类型": {
      "words": "小型轿车"
    },
    "所有人": {
      "words": "孙佳"
    },
    "使用性质": {
      "words": "非营运"
    },
    "发动机号码": {
      "words": "**********2C0365"
    },
    "证芯编号": {
      "words": "1310033548941"
    },
    "生成时间": {
      "words": "2025年04月10日"
    },
    "发证日期": {
      "words": "20201029"
    },
    "检验有效期": {
      "words": "2026年10月"
    },
    "号牌号码": {
      "words": "京BE11Z1"
    },
    "注册日期": {
      "words": "20201029"
    }
  },
  "direction": 0,
  "words_result_num": 13,
  "log_id": 1910254490149466440
}
```


返回示例（行驶证副页）

```
{
  "words_result": {
    "检验记录": {
      "words": "2022年09月豫N"
    },
    "核定载质量": {
      "words": ""
    },
    "整备质量": {
      "words": "8805kg"
    },
    "外廓尺寸": {
      "words": "6950X2490X3570mm"
    },
    "核定载人数": {
      "words": "2人"
    },
    "总质量": {
      "words": "0kg"
    },
    "燃油类型": {
      "words": "柴油"
    },
    "准牵引总质量": {
      "words": "40000kg"
    },
    "备注": {
      "words": "2032-09-01"
    },
    "档案编号": {
      "words": "411491011135"
    },
    "号牌号码": {
      "words": "豫NS2307"
    },
    "证芯编号": {
      "words": "4130036033493"
    }
  },
  "words_result_num": 12,
  "direction": 0,
  "log_id": 1910314944961222207
}
```


返回示例（电子行驶证副页）

```
{
  "words_result": {
    "强制报废期止": {
      "words": "20391025"
    },
    "车身颜色": {
      "words": "绿"
    },
    "能源种类": {
      "words": "柴油"
    },
    "住址": {
      "words": "河北省邯郸市邯山区马头工业城杜村九组"
    },
    "发证机关": {
      "words": "河北省邯郸市公安局交通巡逻警察支队"
    },
    "核定载质量": {
      "words": "9925kg"
    },
    "整备质量": {
      "words": "7880kg"
    },
    "外廓尺寸": {
      "words": "6180X2350X2850mm"
    },
    "核定载人数": {
      "words": ""
    },
    "总质量": {
      "words": "18000kg"
    },
    "准牵引总质量": {
      "words": ""
    },
    "备注": {
      "words": ""
    }
  },
  "words_result_num": 12,
  "direction": 0,
  "log_id": 1910314527349141281
}
```


驾驶证识别

接口描述

支持对机动车驾驶证正页及副页所有字段进行结构化识别。

驾驶证正页：包括证号、姓名、性别、国籍、出生日期、初次领证日期、准驾车型、有效起始日期、失效日期、住址、发证单位 11 个字段。

驾驶证副页：包括姓名、记录、证号、档案编号 4 个字段。

同时支持识别交管12123 APP 发放的电子驾驶证正页，包括证号、姓名、性别、国籍、出生日期、初次领证日期、准驾车型、有效起始日期、失效日期、累积记分、状态、档案编号、生成时间、当前时间、条形码下编号 15个字段。

> 视频教程请参见 驾驶证识别API调用教程（视频版）

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/driving_license`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	否	string	true/false	- **false：默认值，**不检测朝向，朝向是指输入图像是正常方向、逆时针旋转90/180/270度 - **true：**检测朝向
driving_license_side	否	string	front/back	- **front：默认值，**识别驾驶证正页、电子驾驶证正页 - **back：**识别驾驶证副页
unified_valid_period	否	string	true/false	- **false: 默认值**，不进行归一化处理 - **true:** 归一化格式输出，将驾驶证正页的「有效起始日期」+「有效期限」及「有效期限」+「至」，归一化为「有效起始日期」+「失效日期」格式输出
quality_warn	否	string	true/false	是否开启质量检测功能，仅在驾驶证正页识别时生效，可选值如下 -**false：默认值**，不输出质量告警信息 -**true：**在 warn_infos 输出驾驶证遮挡、不完整、模糊质量告警信息。同时，可在 quality_propobility 输出质量检测置信度信息
risk_warn	否	string	true/false	是否开启风险检测功能，仅在驾驶证正页识别时生效， - **false：默认值**，不输出风险告警信息 - **true**：开启，输出驾驶证复印、翻拍、PS等告警信息

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/driving_license?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ words	否	string	识别结果字符串
warn_infos	否	array[]	质量告警信息，当 driving_license_side=front 且 quality_warn=true 时输出 quality_warn=true 时输出， - shield：驾驶证证照存在遮挡告警提示 - incomplete：驾驶证证照边框不完整告警提示 - fuzzy：驾驶证证照存在模糊告警提示
quality_propobility	否	object	质量检测置信度，当 driving_license_side=front 且 quality_warn=true 时输出
+ is_clear_propobility	是	string	“是否清晰”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 is_clear_propobility 超过 0.5 时，对应 warn_infos 返回 fuzzy ，低于 0.5，则不返回 fuzzy
+ is_complete_propobility	是	string	“是否边框/四角完整”质量类型对应的概率，值在0-1之间，值越大表示图像质量越好。**默认阈值**（仅为推荐值，建议按照实际业务场景，基于图片返回的具体概率值，自定义设置判断阈值）：当 is_complete_propobility 超过 0.5 时，对应 warn_infos 返回 incomplete，低于 0.5，则不返回 incomplete
+ is_noshield_propobility	是	string	“是否被遮挡”质量类型对应的概率，值取0或1，0代表图像被遮挡，1代表图像没有被遮挡
risk_type	否	string	风险告警信息，当risk_warn=true 时输出， -normal：正常驾驶证 -copy：复印件 -screen：翻拍
edit_tool	否	string	当输入参数 risk_warn=true 时返回，如果检测驾驶证被编辑过，该字段指定编辑软件名称，如：Adobe Photoshop CC 2014 (Macintosh)，如果没有被编辑过则返回值为空

返回示例（驾驶证正页）

```
{
  "words_result": {
    "姓名": {
      "words": "王桃桃"
    },
    "出生日期": {
      "words": "19880929"
    },
    "证号": {
      "words": "210282198809294228"
    },
    "住址": {
      "words": "辽宁省大连市甘井子区"
    },
    "初次领证日期": {
      "words": "20150518"
    },
    "国籍": {
      "words": "中国"
    },
    "准驾车型": {
      "words": "C1"
    },
    "性别": {
      "words": "女"
    },
    "发证单位": {
  "words": "北京市公安局公安交通管理局"
    },
    "有效期限": {
      "words": "20150518"
    },
    "至": {
      "words": "20210518"
    }
  },
  "log_id": 1321746413993852928,
  "words_result_num": 11,
  "direction": -1
}
```


返回示例（电子驾驶证正页）

```
{
  "words_result": {
    "姓名": {
      "words": "王桃桃"
    },
    "出生日期": {
      "words": "19880929"
    },
    "证号": {
      "words": "210282198809294228"
    },
    "初次领证日期": {
      "words": "20150518"
    },
    "国籍": {
      "words": "中国"
    },
    "准驾车型": {
      "words": "C1"
    },
    "性别": {
      "words": "女"
    },
    "有效起始日期": {
      "words": "20150518"
    },
    "失效日期": {
      "words": "20250518"
    },
    "状态": {
      "words": "正常"
    },
    "档案编号": {
      "words": "429818304"
    },
    "生成时间": {
      "words": "2021年09月04日"
    },
    "当前时间": {
      "words": "2022年04月16日14:09:39"
    },
    "条形码下编号": {
      "words": "*4360028416376*"
    },
    "累积记分": {
  "words": "0分"
    }
  },
  "log_id": 1321746413993852928,
  "words_result_num": 15,
  "direction": -1
}
```


返回示例（驾驶证副页）

```
{
  "words_result": {
    "姓名": {
      "words": "万万"
    },
    "记录": {
      "words": "请于每个记分周期结束后三十日接受审验。无记分的，免予本次审验。"
    },
    "证号": {
      "words": "513601198209290000"
    },
    "档案编号": {
      "words": "511600001169"
    }
  },
  "direction": 0,
  "words_result_num": 4,
  "log_id": 1483000040398531214
}
```


车辆证照混贴识别

接口描述

车辆证照混贴识别接口支持自动检测与识别行驶证、驾驶证混贴图片，即识别机动车行驶证主页及副页、机动车驾驶证主页及副页在同一张图片上的场景，一次性识别图片中多个行驶证、驾驶证的所有字段。

支持对机动车行驶证主页及副页所有22个字段进行结构化识别，包括号牌号码、车辆类型、所有人、品牌型号、车辆识别代码、发动机号码、核定载人数、质量、尺寸、检验记录等；支持对机动车驾驶证正页及副页所有15个字段进行结构化识别，包括证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效期限、发证单位、档案编号等。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/mixed_multi_vehicle`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	否	string	true/false	- **false：默认值**不进行图像方向自动矫正 - **true:** 开启图像方向自动矫正功能，可对旋转 90/180/270 度的图片进行自动矫正并识别
unified	否	string	true/false	- **false：默认值**，不进行归一化处理 - **true：**对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出

请求代码示例 提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/mixed_multi_vehicle?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
card_type	是	string	证件类型，vehicle_front、vehicle_back、driving_front和driving_back，分别对应行驶证正、副页，驾驶证正、副页
direction	否	int32	图像方向，当图像旋转时，返回该参数。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
probability	是	uint32	检测到证件的置信度
location	是	object	证件位置数组（坐标0点为左上角）
+ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+ width	是	uint32	表示定位位置的长方形的宽度
+ height	是	uint32	表示定位位置的长方形的高度
license_info	是	object	识别结果信息，包含图片中多个行驶证、驾驶证的识别结果
+ word_name	是	string	字段名，如果license_info对应行驶证，字段名包含号牌号码、车辆类型、所有人、品牌型号、车辆识别代码、发动机号码、核定载人数、质量、尺寸、检验记录等；如果license_info对应驾驶证，字段名包含证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效期限、发证单位、档案编号等
+ word	是	string	word_name字段对应的识别结果

返回示例


```
{
  "words_result":[
     {
        "license_info":[
           {
              "word_name":"证号",
              "word":"320621196512031267"
           },
           {
              "word_name":"姓名",
              "word":"程成"
           },
           {
              "word_name":"性别",
              "word":"男"
           },
           {
              "word_name":"国籍",
```

```
            "word":"中国"
         },
         {
            "word_name":"住址",
            "word":"江苏省南通市"
         },
         {
            "word_name":"出生日期",
            "word":"19651203"
         },
         {
            "word_name":"初次领证日期",
            "word":"20110311"
         },
         {
            "word_name":"准驾车型",
            "word":"B2"
         },
         {
            "word_name":"有效起始日期",
            "word":"20170311"
         },
         {
            "word_name":"失效日期",
            "word":"20270311"
         },
         {
            "word_name":"发证单位",
            "word":"江苏省南通市公安局交通警察支队"
         }
      ],
      "probability":0.99496907,
      "location":{
         "top":14,
         "left":15,
         "width":701,
         "height":459
      },
      "card_type":"driving_front",
      "direction":0
   },
   {
      "license_info":[
         {
            "word_name":"证号",
            "word":"320621196512031267"
         },
         {
            "word_name":"姓名",
            "word":"程成"
         },
         {
            "word_name":"档案编号",
            "word":"320601650706"
         },
         {
            "word_name":"记录",
            "word":"自2017年03月06日至有效起始日期有效。请于每个记分周期结束后三十日接受审验。无记分的，免予本次审验。"
         }
      ],
      "probability":0.9815229177,
```

```
        "location":{
            "top":7,
            "left":731,
            "width":650,
            "height":456
        },
        "card_type":"driving_back",
        "direction":0
    },
    {
        "license_info":[
            {
                "word_name":"号牌号码",
                "word":"京A10D08"
            },
            {
                "word_name":"车辆类型",
                "word":"小型面包车"
            },
            {
                "word_name":"所有人",
                "word":"王京"
            },
            {
                "word_name":"住址",
                "word":"北京市石景山区"
            },
            {
                "word_name":"使用性质",
                "word":"非营运"
            },
            {
                "word_name":"品牌型号",
                "word":"东风牌EQ6456PF"
            },
            {
                "word_name":"车辆识别代号",
                "word":"LGK022K69A9240616"
            },
            {
                "word_name":"发动机号码",
                "word":"10066180"
            },
            {
                "word_name":"注册日期",
                "word":"20100707"
            },
            {
                "word_name":"发证日期",
                "word":"20191020"
            },
            {
                "word_name":"发证单位",
                "word":"北京市公安局交通警察支队"
            }
        ],
        "probability":0.9907341003,
        "location":{
            "top":532,
            "left":736,
            "width":622,
            "height":415
```

```
        },
        "card_type":"vehicle_front",
        "direction":0
    },
    {
        "license_info":[
            {
                "word_name":"号牌号码",
                "word":"京A10D08"
            },
            {
                "word_name":"备注",
                "word":""
            },
            {
                "word_name":"整备质量",
                "word":"985kg"
            },
            {
                "word_name":"核定载质量",
                "word":""
            },
            {
                "word_name":"外廓尺寸",
                "word":"3660X1560X1925mm"
            },
            {
                "word_name":"核定载人数",
                "word":"7人"
            },
            {
                "word_name":"总质量",
                "word":"1565kg"
            },
            {
                "word_name":"准牵引总质量",
                "word":""
            },
            {
                "word_name":"档案编号",
                "word":""
            },
            {
                "word_name":"检验记录",
                "word":"2020年07月京A(03)"
            },
            {
                "word_name":"燃油类型",
                "word":"汽油"
            }
        ],
        "probability":0.9925737381,
        "location":{
            "top":554,
            "left":12,
            "width":609,
            "height":416
        },
        "card_type":"vehicle_back",
        "direction":0
    }
],
"log_id":1420339917916065252
```

```
    "log_id":1420339917916065252,
    "words_result_num":4
}
```


车牌识别

接口描述

支持识别中国大陆机动车蓝牌、黄牌（单双行）、绿牌、大型新能源（黄绿）、领使馆车牌、警牌、武警牌（单双行）、军牌（单双行）、港澳出入境车牌、农用车牌、民航车牌、非机动车车牌（北京地区，不支持临时牌）的地域编号和车牌号，并能同时识别图像中的多张车牌。

> 视频教程请参见 车牌识别API调用教程（视频版）

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/license_plate`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
image	和url二选一	string	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效请注意关闭URL防盗链
multi_detect	否	string	是否检测多张车牌，默认为false，当置为true的时候可以对一张图片内的多张车牌进行识别
multi_scale	否	string	在高拍等车牌较小的场景下可开启，默认为false，当置为true时，能够提高对较小车牌的检测和识别。**提示：当前新版车牌识别能力无需开启此参数，即可支持在高拍场景下的准确识别，此参数已无效，即将下线**
detect_complete	否	string	是否开启车牌遮挡检测功能，默认为false，不开启； - true：开启遮挡检测 - false：不开启遮挡检测
detect_risk	否	string	是否开启车牌PS检测功能，默认为false，不开启； - true：开启PS检测 - false：不开启PS检测

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/license_plate?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

参数	是否必须	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	识别结果数组
+ color	是	string	车牌颜色：支持blue、green、yellow、white、black、yellow_green（新能源大型汽车黄绿车牌）、unknow（未知颜色）、penyin（大货车喷印车牌）
+ number	是	string	车牌号码
+ probability	是	string	7个数字分别为车牌中每个字符的置信度（从左往右），区间为0-1，如需平均置信度，将全部数值相加，计算平均值即可
+ vertexes_location	是	array[]	返回文字外接多边形顶点位置
++ x	是	uint32	水平坐标（坐标0点为左上角）
++ y	是	uint32	垂直坐标（坐标0点为左上角）
+ cover_info	否	string	判断车牌有没有被遮挡，当 detect_complete=true 时生效； - incomplete：被遮挡； - complete：没有被遮挡
+ edit_tool	否	string	判断车牌有没有被遮挡，当 detect_risk=true 时生效；如果检测车牌被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh)，如果没有被编辑过则返回值为空

返回示例

```
{
"words_result": [
 {
  "color": "blue",
  "number": "京KBT355",
  "probability": [
   0.9999992847,
   0.999999404,
   0.9999910593,
   0.9999765158,
   0.999994874,
   0.9998959303,
   0.9999984503
  ],
  "vertexes_location": [
   {
    "x": 495,
    "y": 589
   },
   {
    "x": 800,
    "y": 587
   },
   {
    "x": 800,
    "y": 676
   },
   {
    "x": 496,
    "y": 678
   }
  ]
 }
],
 "log_id": "6845817085824549137"
}
```


VIN码识别

接口描述

支持对车辆挡风玻璃、发动机铭牌处的车架号码进行识别。

VIN码示例图片：





在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/vin_code`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/vin_code?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	VIN码识别结果

返回示例

```
{
"log_id": 246589877,
"words_result": [
 {
  "location": {
   "left": 124,
   "top": 11,
   "width": 58,
   "height": 359
  },
  "words": "LFV2A11K8D4010942"
 }
],
"words_result_num": 1
}
```


机动车销售发票识别

接口描述

支持对机动车销售发票的26个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、机器编号、购买方名称、购买方身份证号码/组织机构代码、车辆类型、厂牌型号、产地、合格证号、发动机号码、车架号码、价税合计、价税合计小写、销货单位名称、电话、纳税人识别号、账号、地址、开户银行、税率、税额、主管税务机关及代码、不含税价格、限乘人数。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_invoice`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_invoice?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array	识别结果数组
+ InvoiceCode	是	string	发票代码
+ InvoiceNum	是	string	发票号码
+ PrintedCode	是	string	机打代码
+ PrintedNum	是	string	机打号码
+ InvoiceDate	是	string	开票日期
+ MachineCode	是	string	机器编号
+ Purchaser	是	string	购买方名称
+ PurchaserCode	是	string	购买方身份证号码/组织机构代码
+ VehicleType	是	string	车辆类型
+ ManuModel	是	string	厂牌型号
+ Origin	是	string	产地
+ CertificateNum	是	string	合格证号

+ EngineNum	是	string	发动机号码
+ VinNum	是	string	车架号码
+ PriceTax	是	string	价税合计
+ PriceTaxLow	是	string	价税合计小写
+ Saler	是	string	销货单位名称
+ SalerPhone	是	string	销货单位电话
+ SalerCode	是	string	销货单位纳税人识别号
+ SalerAccountNum	是	string	销货单位账号
+ SalerAddress	是	string	销货单位地址
+ SalerBank	是	string	销货单位开户银行
+ TaxRate	是	string	税率
+ Tax	是	string	税额
+ TaxAuthor	是	string	主管税务机关
+ TaxAuthorCode	是	string	主管税务机关代码
+ Price	是	string	不含税价格
+ LimitPassenger	是	string	限乘人数
+ toonage	是	string	吨位
+ sheet-num	是	string	联次
+ drawer	是	string	开票人
+ remarks	是	string	备注
+ import-certificate-num	是	string	进口证明书号
+ tax-payment-voucher-no	是	string	完整凭税编号
+ inspection-form-num	是	string	商检单号

返回示例

```
{
  "log_id": 283449393728149457,
  "words_result_num": 26,
  "words_result": {
    "InvoiceNum": "00875336",
    "Saler": "深圳市新能源汽车销售有限公司",
    "LimitPassenger": "5",
    "MachineCode": "669745967911",
    "VinNum": "LJLGTCRP1J4007581",
    "TaxRate": "16%",
    "PriceTaxLow": "106100.00",
    "InvoiceDate": "2018-11-29",
    "Price": "￥91465.52",
    "SalerBank": "中国工商银行股份有限公司深圳岭园支行",
    "TaxAuthor": "国家锐务总局深圳市龙岗区税务局第五税务所",
    "ManuModel": "江淮牌HFC7007EYBD6",
    "CertificateNum": "WCH0794J0976801",
    "Purchaser": "苏子潇",
    "VehicleType": "纯电动轿车",
    "InvoiceCode": "14975047560",
    "PriceTax": "壹拾万陆仟壹佰圆整",
    "SalerPhone": "0755-83489306",
    "SalerAddress": "深圳市龙岗区龙岗街道百世国际汽车城",
    "Origin": "安徽省合肥市",
    "EngineNum": "18958407",
    "Tax": "14634.48",
    "PurchaserCode": "5135934475603742222",
    "TaxAuthorCode": "14037589413",
    "SalerAccountNum": "中国工商银行股份有限公司深圳岭园支行",
    "SalerCode": "91449283464582922 78H"
  }
}
```


二手车销售发票识别

接口描述

支持对二手车销售发票的25个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、买方、卖方、车牌号、车辆类型、二手车市场等。

注：暂不支持数电二手车销售发票。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/used_vehicle_invoice`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/used_vehicle_invoice?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	是	int32	图像方向 - 1：未定义，0：正向，1：逆时针90度，2：逆时针180度，3：逆时针270度
word_result	是	object	识别结果数组
+ InvoiceCode	是	object	发票代码
+ InvoiceNum	是	object	发票号码
+ InvoiceDate	是	object	开票日期
+ TaxCode	是	object	税控码
+ Purchaser	是	object	买方
+ PurchaserCode	是	object	买方身份证号
+ PurchaserAddress	是	object	买方地址
+ PurchaserPhone	是	object	买方电话
+ Saler	是	object	卖方
+ SalerCode	是	object	卖方身份证号
+ SalerAddress	是	object	卖方地址
+ SalerPhone	是	object	卖方电话
+ LicensePlateNum	是	object	车牌号
+ RegistrationCode	是	object	登记证号
+ VehicleType	是	object	车辆类型
+ VinNum	是	object	车架号
+ ManuModel	是	object	厂牌型号
+ TransferVehicleManagementOffice	是	object	转入地车管所名称
+ TotalCarPrice	是	object	车价合计大写
+ TotalCarPriceLow	是	object	车价合计小写
+ UsedCarMarket	是	object	二手车市场
+ TaxNum	是	object	纳税人识别号
+ TaxAddress	是	object	纳税人地址
+ TaxPhone	是	object	纳税人电话
+ SheetNum	是	object	联次

返回示例


```
{
  "word_result": {
     "TransferVehicleManagementOffice": {
        "word": "车管所"
     },
     "ManuModel": {
        "word": "森林人JF1SH999"
     },
```

```
    "RegistrationCode": {
        "word": "320008998998"
    },
    "TotalCarPriceLow": {
        "word": "￥88000.00"
    },
    "TaxAddress": {
        "word": "苏州高新区"
    },
    "PurchaserCode": {
        "word": "313502198402043055"
    },
    "Saler": {
        "word": "春春"
    },
    "TaxCode": {
        "word": "74881478982966442866"
    },
    "Purchaser": {
        "word": "丽丽"
    },
    "InvoiceCode": {
        "word": "022001900539"
    },
    "InvoiceDate": {
        "word": "2020-05-13"
    },
    "SalerAddress": {
        "word": "江苏省苏州市工业园区"
    },
    "SalerCode": {
        "word": "4124524199001156900"
    },
    "TaxPhone": {
        "word": "12282680332"
    },
    "TaxNum": {
        "word": "91330505MW1NCQUQWE"
    },
    "PurchaserPhone": {
        "word": "0"
    },
    "VehicleType": {
        "word": "小型越野客车"
    },
    "LicensePlateNum": {
        "word": "苏U1A555"
    },
    "SheetNum": {
        "word": "二"
    },
    "VinNum": {
        "word": "JJ1SH95F4AG098838"
    },
    "TotalCarPrice": {
        "word": "捌万捌仟圆整"
    },
    "PurchaserAddress": {
        "word": "江苏省苏州市"
    },
    "SalerPhone": {
        "word": "0"
    }
```

```
        },
        "UsedCarMarket": {
            "word": "苏州二手车电子商务有限公司"
        },
        "InvoiceNum": {
            "word": "00354542"
        }
    },
    "log_id": 1384799789308182528,
    "direction": "0"
}
```


车辆合格证识别

接口描述

支持对车辆合格证的28个关键字段进行结构化识别，包括合格证编号、发证日期、车辆制造企业名、车辆品牌、车辆名称、车辆型号、车架号、车身颜色、发动机型号、发动机号、燃料种类、排量、功率、排放标准、轮胎数、轴距、轴数、转向形式、总质量、整备质量、驾驶室准乘人数、最高设计车速、车辆制造日期等。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_certificate`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash
Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_certificate?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	是	uint64	图像方向。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	识别结果数组
+ CertificationNo	是	string	合格证编号
+ CertificateDate	是	string	发证日期
+ Manufacturer	是	string	车辆制造企业名
+ CarBrand	是	string	车辆品牌
+ CarName	是	string	车辆名称
+ CarModel	是	string	车辆型号
+ VinNo	是	string	车架号
+ CarColor	是	string	车身颜色
+ EngineType	是	string	发动机型号
+ EngineNo	是	string	发动机号
+ FuelType	是	string	燃料种类
+ Displacement	是	string	排量
+ Power	是	string	功率
+ EmissionStandard	是	string	排放标准
+ TyreNum	是	string	轮胎数
+ Wheelbase	是	string	轴距
+ AxleNum	是	string	轴数
+ SteeringType	是	string	转向形式
+ TotalWeight	是	string	总质量
+ SaddleMass	是	string	整备质量
+ LimitPassenger	是	string	驾驶室准乘人数
+ SpeedLimit	是	string	最高设计车速
+ ManufactureDate	是	string	车辆制造日期
+ ChassisID	是	string	底盘ID
+ ChassisModel	是	string	底盘型号
+ SeatingCapacity	是	string	额定载客人数
+ QualifySeal	是	string	合格印章：1表示有，0表示无
+ CGSSeal	是	string	CGS印章：1表示有，0表示无

返回示例

```
{
  "log_id": 14814098736243057,
  "words_result_num": 28,
  "direction": 0,
  "words_result": {
    "ManufactureDate": "2016年10月13日",
    "CarColor": "红",
    "LimitPassenger": "2",
    "EngineType": "WP12.460E50",
    "TotalWeight": "25000",
    "Power": "338",
    "CertificationNo": "WEK29JX98645437",
    "FuelType": "汽油",
    "Manufacturer": "陕西汽车集团有限责任公司",
    "SteeringType": "方向盘",
    "Wheelbase": "3175+1350",
    "SpeedLimit": "105",
    "EngineNo": "1418K129178",
    "SaddleMass": "8600",
    "AxleNum": "3",
    "CarModel": "SX4250MC4",
    "VinNo": "LZGJHYD83JX197344",
    "CarBrand": "陕汽牌",
    "EmissionStandard": "GB17691-2005国V,GB3847-2005",
    "Displacement": "11596",
    "CertificateDate": "2018年11月28日",
    "CarName": "牵引汽车",
    "TyreNum": "10",
    "ChassisID": "",
    "ChassisModel": "",
    "SeatingCapacity": "5",
    "QualifySeal": "1",
    "CGSSeal": "0"
  }
}
```


机动车登记证书识别

接口描述

支持对机动车登记证书的15个关键字段进行结构化识别，包括编号、机动车所有人、登记机关、登记日期、登记编号、车辆类型等，同时支持检测发证机关章

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_registration_certificate`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_registration_certificate?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	是	int32	图像方向 - 1：未定义， 0：正向， 1：逆时针90度， 2：逆时针180度， 3：逆时针270度
words_result	是	object	识别结果数组
+ number	是	object	编号
+ name_idcard_no	是	object	机动车所有人/身份证明名称/号码
+ registration_authority	是	object	登记机关
+ registration_date	是	object	登记日期
+ registration_num	是	object	机动车登记编号
+ vehicle_model	是	object	车辆型号
+ vehicle_type	是	object	车辆类型
+ vin	是	object	车架号
+ engine_num	是	object	发动机号
+ seating_capacity	是	object	核定载客
+ body_color	是	object	车身颜色
+ nature_of_use	是	object	使用性质
+ date_of_production	是	object	出厂日期
+ date_of_issue	是	object	发证日期
+ seal_of_issue_authority	是	object	发证机关章，1表示有，0表示无

返回示例

```
{
  "words_result": {
    "registration_authority": {
      "words": "江苏省昆山市公安局交通巡逻警察大队"
    },
    "vehicle_model": {
      "words": "DHW6691R8CEE"
    },
    "vehicle_type": {
      "words": "小型普通客车"
    },
    "registration_num": {
      "words": "苏A88FF2"
    },
    "engine_num": {
      "words": "2005533"
    },
    "number": {
      "words": "32004574998"
    },
    "body_color": {
      "words": "白"
    },
    "registration_date": {
      "words": "2016-06-06"
    },
    "date_of_production": {
      "words": "2016-03-11"
    },
    "seating_capacity": {
      "words": "7"
    },
    "date_of_issue": {
      "words": "2016-06-06"
    },
    "nature_of_use": {
      "words": "非营运"
    },
    "vin": {
      "words": "LVHRR8836G5004527"
    },
    "seal_of_issue_authority": {
      "words": "1"
    },
    "name_idcard_no": {
      "words": "汽车服务有限公司/统一社会信用代码/91320583088874412F"
    }
  },
  "log_id": 1392089398849306624,
  "direction": "0"
}
```


磅单识别

接口描述

结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/weight_note`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/weight_note?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ PlateNum	是	object	车牌号
+ PrintTime	是	object	打印时间
+ CrossWeight	是	object	毛重
+ TareWeight	是	object	皮重
+ NetWeight	是	object	净重
+ SendingCompany	是	object	发货单位
+ ReceivingCompany	是	object	收货单位
+ DeliveryNumber	是	object	单号
++ word	是	string	字段识别结果，以上各字段均包含此参数
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "words_result": [
    {
      "TareWeight": [
      {
          "word": "14.20"
        }
```

```
        ],
        "CrossWeight": [
          {
            "word": "50.70"
          }
        ],
        "PlateNum": [
          {
            "word": "京A12356"
          }
        ],
        "SendingCompany": [
         {
            "word": "宣化县耿矿煤业有限公司"
          }
        ],
        "DeliveryNumber": [
         {
            "word": "278933000"
          }
        ],
        "ReceivingCompany": [
          {
            "word": "宁夏市京裕达实业公司"
          }
        ],
        "PrintTime": [
          {
            "word": "2020年1月1日"
          }
        ],
        "NetWeight": [
          {
            "word": "36.50"
          }
        ]
      }
    ],
    "words_result_num": 1,
    "log_id": 1428311410130160734
}
```


快递面单识别

接口描述

支持市面上常见版式的快递面单识别，包括申通/圆通/中通/百世汇通/韵达/顺丰/京东/邮政/极兔/天天等面单版式，结构化识别运单号、收/寄件人姓名、收/寄件人电话、收/寄件人地址等字段。同时支持识别隐私面单。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=交通场景OCR&api=rest/2.0/ocr/v1/waybill&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。
请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/waybill`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url 二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和 image 二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
is_identify_virtual_waybill	true/false	string	-	是否需要识别隐私面单，</br>- true：需要识别隐私面单，即此参数存在并填写为true的时候，会增加返回隐私面单的3个字段，详见下方返回参数列表；</br>- false：不识别隐私面单

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/waybill?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
快递面单识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/waybill"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 快递面单识别
*/
public class Waybill {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String waybill() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/waybill";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        Waybill.waybill();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/waybill";
static std::string waybill_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    waybill = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 快递面单识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int waybill(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = waybill_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/waybill?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class Waybill
    {
        // 快递面单识别
        public static string waybill()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/waybill?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("快递面单识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~

###### 返回说明

**返回参数**

| 字段 | 是否必输出 | 类型 | 说明 |
| --- | --- | --- | --- |
| log_id | 是 | uint64 | 调用日志id，用于问题定位 |
| words_result | 是 | object | 识别结果 |
| words_result_num | 是 | uint32 | 识别结果数，表示words_result的元素个数 |
| + image_info | 是 | object |图像四方向 |
| ++ direction | 是 | string | 图像四方向，0表示正向，1/2/3依次表示逆时针旋转90度，180度，270度 |
| + bar_code | 是 | object | 条形码 |
| + waybill_number | 是 | object | 快递运单号 |
| + three_segment_code | 是 | object | 三段码 |
| + recipient_name | 是 | object | 收件人姓名 |
| + sender_name | 是 | object | 寄件人姓名 |
| + recipient_addr | 是 | object | 收件人地址 |
| + sender_addr | 是 | object | 寄件人地址 |
| + recipient_phone | 是 | object | 收件人电话 |
| + sender_phone | 是 | object | 寄件人电话 |
| + virtual_number | 是 | object | 虚拟面单号。当请求参数 is_identify_virtual_waybill = true 时返回该字段 |
| + virtual_number_last | 是 | object | 隐私面单的4位转接号。当请求参数 is_identify_virtual_waybill = true 时返回该字段 |
| + is_virtual_waybill | 是 | object | 此张快递面单是否为隐私面单，“true”代表“是”，“false”代表“否”。当请求参数 is_identify_virtual_waybill = true 时返回该字段 |
| ++ word | 是 | string | 字段识别结果，以上各字段均包含此参数 |

**返回示例**

```JSON
{
    "words_result_num": 10,
    "words_result": [
```
```

```
        {
            "image_info": {
                "direction": "0"
            },
            "bar_code": [
                {
                    "word": ""
                }
            ],
            "waybill_number": [
                {
                    "word": "J420222-2013-013228"
                }
            ],
            "three_segment_code": [
                {
                    "word": ""
                }
            ],
            "recipient_name": [
                {
                    "word": ""
                }
            ],
            "sender_name": [
                {
                    "word": "中国"
                }
            ],
            "recipient_addr": [
                {
                    "word": ""
                }
            ],
            "sender_addr": [
                {
                    "word": ""
                }
            ],
            "recipient_phone": [
                {
                    "word": ""
                }
            ],
            "sender_phone": [
                {
                    "word": ""
                }
            ]
        }
    ],
    "log_id": 1654092794470239194
}
```


道路运输证识别

接口描述

结构化识别道路运输证的业户名称、地址、车辆号牌、经营许可证、经济类型、车辆类型、吨座位、车辆规格、经营范围、初领日期、备注、发证日期、道路运输证号等15个关键字段，支持识别横版及竖版两种道路运输证

在线调试

在线调试

您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=交通场景OCR&api=rest/2.0/ocr/v1/road_transport_certificate&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/road_transport_certificate`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大8192px</br>**优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/road_transport_certificate?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
道路运输证识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/road_transport_certificate"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 道路运输证识别
*/
public class RoadTransportCertificate{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String roadTransportCertificate() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/road_transport_certificate";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        RoadTransportCertificate.roadTransportCertificate();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/road_transport_certificate";
static std::string roadTransportCertificate_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    roadTransportCertificate_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 道路运输证识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int roadTransportCertificate(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = roadTransportCertificate_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/road_transport_certificate?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class RoadTransportCertificate
    {
        // 道路运输证识别
        public static string roadTransportCertificate()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/road_transport_certificate?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("道路运输证识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数
words_result	是	object{}	识别结果
++ word	否	string	识别结果字符串

返回示例

```JSON
{
    "words_result_num": 15,
    "words_result": {
        "道路运输证号": [
            {
                "word": "20219269"
            }
        ],
        "车辆号牌": [
            {
                "word": "皖CC2987"
            }
        ],
        "经济类型": [
            {
                "word": ""
            }
        ],
        "经营范围": [
            {
                "word": "普通货运"
            }
        ],
        "车辆类型": [
            {
                "word": "欧曼牌BJ42593780KB-CA"
            }
        ],
        "吨座位": [
            {
                "word": "0"
            }
        ],
        "备注": [
            {
                "word": ""
            }
        ],
        "经营许可证": [
            {
                "word": "340322220617"
            }
        ],
        "车辆毫米_高": [
            {
                "word": "3900"
            }
        ],
        "车辆毫米_宽": [
            {
                "word": "2490"
            }
        ],
        "发证日期": [
            {
                "word": "2020年11月27日"
            }
        ],
        "地址": [
            {
```

```
            "word": "安徽省蚌埠市五河县城关镇张庙村张庙46号"
          }
        ],
        "车辆毫米_长": [
          {
            "word": "7115"
          }
        ],
        "业户名称": [
          {
            "word": "五河东弘汽车运输有限公司"
          }
        ],
        "初领日期": [
          {
            "word": ""
          }
        ]
    },
    "log_id": 1498639858564778332
}
```


财务票据文字识别

智能财务票据识别

接口描述

支持财务场景中13种常见票据的分类及结构化识别，包括增值税发票、卷票、机打发票、定额发票、火车票（含电子发票铁路电子客票）、出租车票、网约车行程单、飞机行程单（含电子发票航空电子客票行程单）、汽车票、过路过桥费、船票、机动车/二手车销售发票。支持多张不同种类票据在同一张图片上的混贴场景，可返回每张票据的位置、种类及票面信息的结构化识别结果。

>视频教程请参见 [智能财务票据识别+增值税发票验真使用教程](https://cloud.baidu.com/video-center/video.html?id=752)

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668425998119#/api?product=AI&project=文字识别&parent=财务票据OCR&api=rest/2.0/ocr/v1/multiple_invoice&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。
请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/multiple_invoice`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
verify_parameter	否	string	true/false	是否开启验真，默认为 false，即不开启，**当为 true 时，返回匹配发票验真接口所需的6要素信息，具体返回信息详见末尾说明**
probability	否	string	true/false	是否返回字段置信度，默认为 false ，即不返回
location	否	string	true/false	是否返回字段位置坐标，默认为 false，即不返回

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/multiple_invoice?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
智能财务票据识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/multiple_invoice"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 智能财务票据识别
*/
public class MultipleInvoice {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String multipleInvoice() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/multiple_invoice";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MultipleInvoice.multipleInvoice();
    }
}
```
```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/multiple_invoice";
static std::string multipleInvoice_result;
```
```

```
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    multipleInvoice_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 智能财务票据识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int multipleInvoice(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = multipleInvoice_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }
```

```
    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/multiple_invoice?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MultipleInvoice
    {
        // 智能财务票据识别
        public static string multipleInvoice()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/multiple_invoice?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("智能财务票据识别:");
            Console.WriteLine(result);
```

```
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ probability	是	string	表示单张票据分类的置信度
+ left	是	string	表示单张票据定位位置的长方形左上顶点的水平坐标
+ top	是	string	表示单张票据定位位置的长方形左上顶点的垂直坐标
+ width	是	string	表示单张票据定位位置的长方形的宽度
+ height	是	string	表示单张票据定位位置的长方形的高度
+ type	是	string	每一张票据的种类
+ result	是	dict	单张票据的识别结果

type 字段会返回以下17种结果，每种结果对应的票据类型详见下表

type 返回结果	说明
vat_invoice	增值税发票
taxi_receipt	出租车票
train_ticket	火车票
quota_invoice	定额发票
air_ticket	飞机行程单
roll_normal_invoice	卷票
printed_invoice	机打发票
printed_elec_invoice	机打电子发票
bus_ticket	汽车票
toll_invoice	过路过桥费发票
ferry_ticket	船票
motor_vehicle_invoice	机动车销售发票
used_vehicle_invoice	二手车销售发票
taxi_online_ticket	网约车行程单
limit_invoice	限额发票
shopping_receipt	购物小票
pos_invoice	POS小票
others	其他

返回说明-增值税发票

type 的返回结果为 vat_invoice，即“增值税发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。**不同消费类型输出：**餐饮、电器设备、通讯、服务、日用品食品、医疗、交通、其他
++ InvoiceTypeOrg	是	array[]	发票名称
++ InvoiceType	是	array[]	增值税发票的细分类型。 **不同细分类型的发票输出：**普通发票、专用发票、电子普通发票、电子专用发票、通行费电子普票、区块链发票、通用机打电子发票、电子发票(专用发票)、电子发票(普通发票)
++ InvoiceTag	是	array[]	增值税发票左上角标志。 包含：通行费、销项负数、代开、收购、成品油、其他
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ InvoiceCodeConfirm	是	array[]	发票代码的辅助校验码，一般业务情景可忽略
++ InvoiceNum	是	array[]	发票号码的辅助校验码，一般业务情景可忽略

Confirm			
++ CheckCode	是	array[]	校验码。增值税专票无此参数
++ InvoiceNumDigit	是	array[]	数电票号码。密码区部分的数电票号码，仅在纸质的数电票上出现
++ InvoiceDate	是	array[]	开票日期
++ PurchaserName	是	array[]	购方名称
++ PurchaserRegisterNum	是	array[]	购方纳税人识别号
++ PurchaserAddress	是	array[]	购方地址及电话
++ PurchaserBank	是	array[]	购方开户行及账号
++ Password	是	array[]	密码区
++ Province	是	array[]	省
++ City	是	array[]	市
++ SheetNum	是	array[]	联次信息。**专票**第一联到第三联分别输出：第一联：记账联、第二联：抵扣联、第三联：发票联；**普通发票**第一联到第二联分别输出：第一联：记账联、第二联：发票联
++ Agent	是	array[]	是否代开
++ OnlinePay	是	String	电子支付标识。仅区块链发票含有此参数
++ SellerName	是	array[]	销售方名称
++ SellerRegisterNum	是	array[]	销售方纳税人识别号
++ SellerAddress	是	array[]	销售方地址及电话
++ SellerBank	是	array[]	销售方开户行及账号
++ TotalAmount	是	array[]	合计金额
++ TotalTax	是	array[]	合计税额
++ AmountInWords	是	array[]	价税合计(大写)

++ AmountInFiguers	是	array[]	价税合计(小写)
++ Payee	是	array[]	收款人
++ Checker	是	array[]	复核
++ NoteDrawer	是	array[]	开票人
++ Remarks	是	array[]	备注
+++ word	是	string	识别结果字符串，以上各字段均包含
++ CommodityName	是	array[]	货物名称
++ CommodityType	是	array[]	规格型号
++ CommodityUnit	是	array[]	单位
++ CommodityNum	是	array[]	数量
++ CommodityPrice	是	array[]	单价
++ CommodityAmount	是	array[]	金额
++ CommodityTaxRate	是	array[]	税率
++ CommodityTax	是	array[]	税额
++ CommodityPlateNum	是	array[]	车牌号。仅通行费增值税电子普通发票含有此参数，其余类型该参数返回为空
++ CommodityVehicleType	是	array[]	类型。仅通行费增值税电子普通发票含有此参数，其余类型该参数返回为空
++ CommodityStartDate	是	array[]	通行日期起。仅通行费增值税电子普通发票含有此参数，其余类型该参数返回为空
++ CommodityEndDate	是	array[]	通行日期止。仅通行费增值税电子普通发票含有此参数，其余类型该参数返回为空

++ PassengName	是	array[]	出行人。仅旅客运输类发票有此参数，其余类型该参数返回为空
++ PassengIdNum	是	array[]	有效身份证件号，仅旅客运输类发票有此参数，其余类型该参数返回为空
++ PassengDate	是	array[]	出行日期。仅旅客运输类发票有此参数，其余类型该参数返回为空
++ PassengOrigin	是	array[]	出发地。仅旅客运输类发票有此参数，其余类型该参数返回为空
++ PassengDestination	是	array[]	到达地。仅旅客运输类发票有此参数，其余类型该参数返回为空
++ PassengClass	是	array[]	等级。仅旅客运输类发票有此参数，其余类型该参数返回为空
++ PassengVehicleType	是	array[]	交通工具类型。仅旅客运输类发票有此参数，其余类型该参数返回为空
++ TransportType	是	array[]	运输工具种类。仅货物运输类发票有此参数，其余类型该参数返回为空
++ TransportPlateNum	是	array[]	运输工具牌号。仅货物运输类发票有此参数，其余类型该参数返回为空
++ TransportDeparture	是	array[]	起运地。仅货物运输类发票有此参数，其余类型该参数返回为空
++ TransportArrival	是	array[]	到达地。仅货物运输类发票有此参数，其余类型该参数返回为空
++ TransportCargoInfo	是	array[]	运输货物名称。仅货物运输类发票有此参数，其余类型该参数返回为空
+++ row	是	uint32	行号，以上各字段均包含
+++ word	是	string	内容，以上各字段均包含

返回说明-出租车票

type 的返回结果为 taxi_receipt，即“出租车票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。出租车票此字段固定输出：交通
++ InvoiceCode	是	array[]	发票代号
++ InvoiceNum	是	array[]	发票号码
++ TaxiNum	是	array[]	车牌号
++ Date	是	array[]	日期
++ Time	是	array[]	上下车时间
++ PickupTime	是	array[]	上车时间
++ DropoffTime	是	array[]	下车时间
++ Fare	是	array[]	金额
++ FuelOilSurcharge	是	array[]	燃油附加费
++ CallServiceSurcharge	是	array[]	叫车服务费
++ TotalFare	是	array[]	总金额
++ Location	是	array[]	开票城市
++ Province	是	array[]	省
++ City	是	array[]	市
++ PricePerkm	是	array[]	单价
++ Distance	是	array[]	里程
+++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-火车票

type 的返回结果为 train_ticket，即“火车票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。火车票此字段固定输出：交通
++ ticket_num	是	array[]	车票号
++ starting_station	是	array[]	始发站
++ train_num	是	array[]	车次号
++ destination_station	是	array[]	到达站
++ date	是	array[]	出发日期
++ ticket_rates	是	array[]	车票金额，当火车票为退票时，该字段表示退票费
++ seat_category	是	array[]	席别
++ name	是	array[]	乘客姓名
++ ID_card	是	array[]	身份证号
++ serial_number	是	array[]	序列号
++ sales_station	是	array[]	售站
++ time	是	array[]	时间
++ seat_num	是	array[]	座位号
++ Waiting_area	是	array[]	候检区
++ refund_flag	否	array[]	标识，仅在输入为铁路电子客票时返回值，包括“退票”、“换开”、“始发改签”等
++ invoice_num	否	array[]	发票号码
++ invoice_date	否	array[]	开票日期
++ fare	否	array[]	不含税金额
++ tax_rate	否	array[]	税率
++ tax	否	array[]	税额
++ elec_ticket_num	否	array[]	电子客票号
++purchaser_name	否	array[]	购买方名称
++purchaser_register_num	否	array[]	购买方统一社会信用代码
+++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-定额发票

type 的返回结果为 quota_invoice，即“定额发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。定额发票此字段固定输出：交通
++ invoice_code	是	array[]	发票代码
++ invoice_number	是	array[]	发票号码
++ invoice_rate	是	array[]	金额
++ invoice_rate_in_figure	是	array[]	金额小写
++ invoice_rate_in_word	是	array[]	金额大写
++ Province	是	array[]	省
++ City	是	array[]	市
++ Location	是	array[]	发票所在地
++ invoice_type	是	array[]	发票名称
+++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-飞机行程单

type 的返回结果为 air_ticket，即“飞机行程单”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。飞机行程单此字段固定输出：交通
++ name	是	array[]	姓名
++ starting_station	是	array[]	始发站
++ destination_station	是	array[]	目的站
++ flight	是	array[]	航班号
++ date	是	array[]	日期
++ ticket_number	是	array[]	电子客票号码
++ fare	是	array[]	票价
++ dev_fund	是	array[]	民航发展基金/机建费
++ oil_money	是	array[]	燃油附加费
++ other_tax	是	array[]	其他税费
++ ticket_rates	是	array[]	合计金额
++ start_date	是	array[]	填开日期
++ id_no	是	array[]	身份证号
++ carrier	是	array[]	承运人
++ time	是	array[]	时间
++ issued_by	是	array[]	填开单位
++ serial_number	是	array[]	印刷序号
++ insurance	是	array[]	保险费
++ fare_basis	是	array[]	客票级别
++ class	是	array[]	座位等级
++ agent_code	是	array[]	销售单位号
++ endorsement	是	array[]	签注
++ allow	是	array[]	免费行李
++ ck	是	array[]	验证码
++ effective_date	是	array[]	客票生效日期
++ expiration_date	是	array[]	有效期截止日期
++ invoice_type_org	是	array[]	发票名称
++ identification	是	array[]	国内国际标识
++ invoice_status	是	array[]	开票状态
++ invoice_num	是	array[]	发票号码
++ commodity_tax_rate	是	array[]	增值税税率
++ commodity_tax	是	array[]	增值税税额
++ purchaser_name	是	array[]	购买方名称
++ purchaser_register_num	是	array[]	统一社会信用代码/纳税人识别号
++ tip	是	array[]	提示信息
+++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-卷票

type 的返回结果为 roll_normal_invoice，即“卷票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceType	是	array[]	发票名称
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ MachineNum	是	array[]	机打号码。仅增值税卷票含有此参数
++ MachineCode	是	array[]	机器编号。仅增值税卷票含有此参数
++ InvoiceDate	是	array[]	开票日期
++ PurchaserName	是	array[]	购方名称
++ PurchaserRegisterNum	是	array[]	购方纳税人识别号
++ SellerName	是	array[]	销售方名称
++ SellerRegisterNum	是	array[]	销售方纳税人识别号
++ TotalTax	是	array[]	价税合计
++ AmountInWords	是	array[]	合计金额(大写)
++ AmountInFiguers	是	array[]	合计金额(小写)
++ Payee	是	array[]	收款人
++ CheckCode	是	array[]	校验码。增值税专票无此参数
++ Province	是	array[]	省
++ City	是	array[]	市
+++ word	是	string	识别结果字符串，以上各字段均包含
++ CommodityName	是	array[]	货物名称
++ CommodityNum	是	array[]	数量
++ CommodityPrice	是	array[]	单价
++ CommodityAmount	是	array[]	金额
+++ row	是	uint32	行号，以上各字段均包含
+++ word	是	string	内容，以上各字段均包含

返回说明-机打发票

type 的返回结果为 printed_invoice，即“机打发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。**不同消费类型输出**：餐饮、电器设备、通讯、服务、日用品食品、医疗、交通、其他
++ InvoiceType	是	array[]	发票类型
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ InvoiceDate	是	array[]	开票日期
++ AmountInFiguers	是	array[]	合计金额小写
++ AmountInWords	是	array[]	合计金额大写
++ MachineNum	是	array[]	机打号码
++ CheckCode	是	array[]	校验码
++ SellerName	是	array[]	销售方名称
++ SellerRegisterNum	是	array[]	销售方纳税人识别号
++ PurchaserName	是	array[]	购买方名称
++ PurchaserRegisterNum	是	array[]	购买方纳税人识别号
++ TotalTax	是	array[]	合计税额
++ Province	是	array[]	省
++ City	是	array[]	市
++ Time	是	array[]	时间
++ SheetNum	是	array[]	联次
+++ word	是	string	识别结果字符串，以上各字段均包含
++ CommodityName	是	array[]	商品名称
++ CommodityUnit	是	array[]	商品单位
++ CommodityPrice	是	array[]	商品单价
++ CommodityNum	是	array[]	商品数量
++ CommodityAmount	是	array[]	商品金额
+++ row	是	uint32	行号，以上各字段均包含
+++ word	是	string	内容，以上各字段均包含

返回说明-汽车票

type 的返回结果为 bus_ticket，即“汽车票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。汽车票此字段固定输出：交通
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ Date	是	array[]	日期
++ Time	是	array[]	时间
++ ExitStation	是	array[]	出发站
++ Amount	是	array[]	金额
++ IdCard	是	array[]	身份证号
++ ArrivalStation	是	array[]	到达站
++ Name	是	array[]	姓名
++ InvoiceTime	是	array[]	开票日期
+++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-过路过桥费

type 的返回结果为 toll_invoice，即“过路过桥费”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。过路过桥费此字段固定输出：交通
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ Entrance	是	array[]	入口
++ Exit	是	array[]	出口
++ OutDate	是	array[]	日期
++ OutTime	是	array[]	时间
++ TotalAmount	是	array[]	金额
++ Province	是	array[]	省
++ City	是	array[]	市
+++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-船票

type 的返回结果为 ferry_ticket，即“船票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。船票此字段固定输出：交通
++ InvoiceType	是	array[]	发票类型
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ ExitStation	是	array[]	出发地点
++ ArrivalStation	是	array[]	到达地点
++ Amount	是	array[]	总金额
++ Date	是	array[]	开票日期
++ MoneyType	是	array[]	金额类型
++ BarCode	是	array[]	条码
++ BarCodeNum	是	array[]	条码编号
++ City	是	array[]	市
++ Province	是	array[]	省
++ InvoiceTitle	是	array[]	发票抬头，这里指该张船票的运行公司名
++ QrCode	是	array[]	二维码
++ Time	是	array[]	出发时间
++ TicketTime	是	array[]	制票时间
++ TicketDate	是	array[]	制票日期
++ PassengerName	是	array[]	乘客姓名
++ IdCard	是	array[]	乘客身份证号
+++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-机动车销售发票

type 的返回结果为 motor_vehicle_invoice，即“机动车销售发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ date	是	array[]	开票日期
++ fapiao-daima	是	array[]	发票代码
++ fapiao-haoma	是	array[]	发票号码
++ printed-daima	是	array[]	机打代码
++ printed-haoma	是	array[]	机打号码
++ machine-num	是	array[]	机器编号
++ buyer-name	是	array[]	购买方名称
++ payer-tax-num	是	array[]	纳税人识别号/统一社会信用代码/身份证号码
++ car-class	是	array[]	车辆类型
++ car-model	是	array[]	厂牌型号
++ product-location	是	array[]	产地
++ certificate-num	是	array[]	合格证号
++ engine-num	是	array[]	发动机号码
++ vin-num	是	array[]	车辆识别代号/车架号码
++ price-tax-big	是	array[]	价税合计
++ price-tax-small	是	array[]	价税合计小写
++ saler	是	array[]	销货单位名称
++ saler-phone	是	array[]	销货单位电话
++ saler-tax-num	是	array[]	销货单位纳税人识别号
++ saler-bank-num	是	array[]	销货单位账号
++ saler-address	是	array[]	销货单位地址
++ saler-bank	是	array[]	销货单位开户银行
++ tax-rate	是	array[]	税率
++ tax	是	array[]	税额
++ tax-jiguan	是	array[]	主管税务机关
++ tax-jiguan-daima	是	array[]	主管税务机关代码
++ price	是	array[]	不含税价格
++ limit-mount	是	array[]	限乘人数
++ toonage	是	array[]	吨位
++ sheet-num	是	array[]	联次
++ drawer	是	array[]	开票人
++ remarks	是	array[]	备注
++ import-certificate-num	是	array[]	进口证明书号
++ tax-payment-voucher-no	是	array[]	完整凭税编号
++ inspection-form-num	是	array[]	商检单号
+++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-二手车销售发票

type 的返回结果为 used_vehicle_invoice，即“二手车销售发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ invoice_code	是	array[]	发票代码
++ invoice_num	是	array[]	发票号码
++ date	是	array[]	开票日期
++ tax_code	是	array[]	税控码
++ buyer	是	array[]	买方
++ buyer_id	是	array[]	买方身份证号
++ buyer_station	是	array[]	买方地址
++ buyer_tel	是	array[]	买方电话
++ saler	是	array[]	卖方
++ saler_id	是	array[]	卖方身份证号
++ saler_station	是	array[]	卖方地址
++ saler_tel	是	array[]	卖方电话
++ car_plate	是	array[]	车牌号
++ car_certificate	是	array[]	登记证号
++ car_class	是	array[]	车辆类型
++ vin_num	是	array[]	车架号
++ model	是	array[]	厂牌型号
++ to_station	是	array[]	转入地车管所名称
++ big_price	是	array[]	车价合计大写
++ small_price	是	array[]	车价合计小写
++ car_market	是	array[]	二手车市场
++ tax_num	是	array[]	纳税人识别号
++ tax_location	是	array[]	纳税人地址
++ tax_tel	是	array[]	纳税人电话
++ sheet_num	是	array[]	联次
+++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-网约车行程单

type 的返回结果为 taxi_online_ticket，即“网约车行程单”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ ServiceType	是	array[]	发票消费类型。网约车行程单此字段固定输出：交通
++ service_provider	是	array[]	服务商
++ start_time	是	array[]	行程开始时间
++ destination_time	是	array[]	行程结束时间
++ phone	是	array[]	行程人手机号
++ application_date	是	array[]	申请日期
++ total_fare	是	array[]	总金额
++ item_num	是	array[]	行程信息中包含的行程数量
+++ word	是	string	识别结果字符串，以上各字段均包含
++ items	是	array[]	行程信息
+++ item_id	是	array[]	行程信息的对应序号
+++ item_provider	是	array[]	行程信息的对应服务商
+++ pickup_time	是	array[]	上车时间
+++ pickup_date	是	array[]	上车日期
+++ car_type	是	array[]	车型
+++ distance	是	array[]	里程
+++ start_place	是	array[]	起点
+++ destination_place	是	array[]	终点
+++ city	是	array[]	城市
+++ fare	是	array[]	金额
++++ word	是	string	识别结果字符串，以上各字段均包含

返回说明-限额发票

type 的返回结果为 limit_invoice，即“限额发票”时，由于此类型仅支持检测分类，无法识别具体票据内容。识别结果的返回字段如下

字段	是否必选	类型	说明
words_result	是	object{}	识别结果
+ probability	是	string	表示单张票据分类的置信度
+ left	是	string	表示单张票据定位位置的长方形左上顶点的水平坐标
+ top	是	string	表示单张票据定位位置的长方形左上顶点的垂直坐标
+ width	是	string	表示单张票据定位位置的长方形的宽度
+ height	是	string	表示单张票据定位位置的长方形的高度
+ type	是	string	**limit_invoice**

返回说明-购物小票

type 的返回结果为 shopping_receipt，即“购物小票”时，由于此类型仅支持检测分类，无法识别具体票据内容。识别结果的返回字段如下

字段	是否必选	类型	说明
words_result	是	object{}	识别结果
+ probability	是	string	表示单张票据分类的置信度
+ left	是	string	表示单张票据定位位置的长方形左上顶点的水平坐标
+ top	是	string	表示单张票据定位位置的长方形左上顶点的垂直坐标
+ width	是	string	表示单张票据定位位置的长方形的宽度
+ height	是	string	表示单张票据定位位置的长方形的高度
+ type	是	string	**shopping_receipt**

返回说明-POS小票

type 的返回结果为 pos_invoice，即“POS小票”时，由于此类型仅支持检测分类，无法识别具体票据内容。识别结果的返回字段如下

字段	是否必选	类型	说明
words_result	是	object{}	识别结果
+ probability	是	string	表示单张票据分类的置信度
+ left	是	string	表示单张票据定位位置的长方形左上顶点的水平坐标
+ top	是	string	表示单张票据定位位置的长方形左上顶点的垂直坐标
+ width	是	string	表示单张票据定位位置的长方形的宽度
+ height	是	string	表示单张票据定位位置的长方形的高度
+ type	是	string	**pos_invoice**

返回说明-其他

type 的返回结果为 others，即“其他”时，由于此类型仅支持检测分类，无法识别具体票据内容。识别结果的返回字段如下

字段	是否必选	类型	说明
words_result	是	object{}	识别结果
+ probability	是	string	表示单张票据分类的置信度
+ left	是	string	表示单张票据定位位置的长方形左上顶点的水平坐标
+ top	是	string	表示单张票据定位位置的长方形左上顶点的垂直坐标
+ width	是	string	表示单张票据定位位置的长方形的宽度
+ height	是	string	表示单张票据定位位置的长方形的高度
+ type	是	string	**others**

返回说明-验真参数开启

当验真参数开启（即 verify_parameter=true 时），返回匹配发票验真接口所需的6要素信息

字段	是否必选	类型	说明
++ invoice_code	是	array[]	发票代码
++ invoice_num	是	array[]	发票号码
++ invoice_date	是	array[]	开票日期。返回格式为 YYYYMMDD，例：20210101
++ invoice_type	是	array[]	发票种类。**不同类型发票输出如下结果**： 增值税专用发票：special_vat_invoice 增值税电子专票：elec_special_vat_invoice 增值税普通发票：normal_invoice 增值税普通发票（电子）：elec_normal_invoice 增值税普通发票（卷式）：roll_normal_invoice 全电发票（专用发票）：elec_invoice_special 全电发票（普通发票）：elec_invoice_normal 通行费增值税电子普通发票：toll_elec_normal_invoice 货运运输业增值税专用发票：special_freight_transport_invoice 机动车销售发票：motor_vehicle_invoice 二手车销售发票：used_vehicle_invoice 区块链发票：blockchain_invoice 通用机打电子发票：printed_elec_invoice 电子发票（铁路电子客票）：elec_train_ticket_invoice 电子发票（航空运输电子客票行程单）：elec_flight_itinerary_invoice
++ total_amount	是	array[]	不含税金额
++ check_code	否	array[]	检验码。**如需使用百度的增值税发票验真接口，需提取返回值的后6位后，再传入验真接口**

返回示例


```
{
  "words_result": [
    {
      "type": "vat_invoice",
      "width": 0,
      "probability": 0.9980429411,
      "height": 649,
      "left": 154,
      "top": 177,
      "result": {
        "AmountInWords": [
          {
            "word": "叁佰陆拾圆整"
          }
        ],
        "InvoiceNumConfirm": [
          {
            "word": "07286261"
          }
        ],
        "CommodityEndDate": [],
        "CommodityVehicleType": [],
        "CommodityStartDate": [],
        "CommodityPrice": [
          {
```

```
            "row": "1",
            "word": "339.62"
        }
    ],
    "NoteDrawer": [
        {
            "word": "余佳燕"
        }
    ],
    "SellerAddress": [],
    "CommodityNum": [
        {
            "row": "1",
            "word": "1"
        }
    ],
    "SellerRegisterNum": [
        {
            "word": "91330106673959654P"
        }
    ],
    "MachineCode": [],
    "Remarks": [],
    "SellerBank": [
        {
            "word": "招商银行杭州高新支行502905023610702"
        }
    ],
    "CommodityTaxRate": [
        {
            "row": "1",
            "word": "6%"
        }
    ],
    "TotalTax": [
        {
            "word": "20.38"
        }
    ],
    "InvoiceCodeConfirm": [
        {
            "word": "3321192130"
        }
    ],
    "CheckCode": [],
    "InvoiceCode": [
        {
            "word": "3321192130"
        }
    ],
    "InvoiceDate": [
        {
            "word": "2019年08月28日"
        }
    ],
    "PurchaserRegisterNum": [
        {
            "word": "91110911717743469K"
        }
    ],
    "InvoiceTypeOrg": [
        {
            "word": "浙江增值税专用发票"
```

```
            "word": "浙江增值税专用发票"
        }
    ],
    "OnlinePay": [],
    "Password": [
        {
            "word": "508>3909>1*>01/-46709-6/3+*7+8>/1*19+7-0**>+58290-6>647-
+324865*9*1<*2191/7754/<0>2<838+//5-69--748*<251408<"
        }
    ],
    "Agent": [
        {
            "word": "否"
        }
    ],
    "AmountInFiguers": [
        {
            "word": "360.00"
        }
    ],
    "PurchaserBank": [
        {
            "word": "招商银行北京分行大电路支行866180100210002"
        }
    ],
    "Checker": [
        {
            "word": "柳余"
        }
    ],
    "City": [],
    "TotalAmount": [
        {
            "word": "339.62"
        }
    ],
    "CommodityAmount": [
        {
            "row": "1",
            "word": "339.62"
        }
    ],
    "PurchaserName": [
        {
            "word": "百度在线网络技术(北京)有限公司"
        }
    ],
    "CommodityType": [],
    "Province": [
        {
            "word": "浙江"
        }
    ],
    "InvoiceType": [
        {
            "word": "专用发票"
        }
    ],
    "SheetNum": [
        {
            "word": "第二联：抵扣联"
        }
    ],
```

```
            "PurchaserAddress": [],
            "CommodityTax": [
                {
                    "row": "1",
                    "word": "20.38"
                }
            ],
            "CommodityPlateNum": [],
            "CommodityUnit": [
                {
                    "row": "1",
                    "word": "套"
                }
            ],
            "Payee": [
                {
                    "word": "佳机"
                }
            ],
            "CommodityName": [
                {
                    "row": "1",
                    "word": "*信息技术服务*软件服务费"
                }
            ],
            "SellerName": [
                {
                    "word": "百度智能云"
                }
            ],
            "InvoiceNum": [
                {
                    "word": "07286261"
                }
            ]
        }
    },
    {
        "type": "taxi_receipt",
        "width": 0,
        "probability": 0.9858493805,
        "height": 615,
        "left": 1325,
        "top": 200,
        "result": {
            "PickupTime": [
                {
                    "word": "10:50"
                }
            ],
            "DropoffTime": [
                {
                    "word": "17:06"
                }
            ],
            "Time": [
                {
                    "word": "10:50-17:06"
                }
            ],
            "City": [
                {
```

```
            "word": ""
        }
    ],
    "FuelOilSurcharge": [
        {
            "word": "1.00"
        }
    ],
    "Date": [
        {
            "word": "2019-03-20"
        }
    ],
    "Province": [
        {
            "word": "陕西省"
        }
    ],
    "CallServiceSurcharge": [
        {
            "word": "0.00"
        }
    ],
    "Fare": [
        {
            "word": "21.10"
        }
    ],
    "TotalFare": [
        {
            "word": "22.00"
        }
    ],
    "TaxiNum": [
        {
            "word": "AQ6353"
        }
    ],
    "PricePerkm": [
        {
            "word": "2.30"
        }
    ],
    "InvoiceCode": [
        {
            "word": "161001881016"
        }
    ],
    "Distance": [
        {
            "word": "6.0"
        }
    ],
    "InvoiceNum": [
        {
            "word": "05070716"
        }
    ],
    "Location": [
        {
            "word": "陕西省"
        }
```

```
            ]
        }
    }
  ],
  "words_result_num": 2,
  "log_id": 1438382953545048984
}
```


增值税发票识别

接口描述

支持对增值税普票、专票、全电发票（新版全国统一电子发票，专票/普票）、卷票、区块链发票的所有字段进行结构化识别，包括发票基本信息、销售方及购买方信息、商品信息、价税信息等，其中五要素字段的识别准确率超过 99.9%； 同时，支持对增值税卷票的 21 个关键字段进行识别，包括发票类型、发票代码、发票号码、机打号码、机器编号、收款人、销售方名称、销售方纳税人识别号、开票日期、购买方名称、购买方纳税人识别号、项目、单价、数量、金额、税额、合计金额(小写)、合计金额(大写)、校验码、省、市，四要素字段的识别准确率可达95%。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/vat_invoice`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
type	否	string	normal/roll	进行识别的增值税发票类型，默认为 normal，可缺省 - **normal**：可识别增值税普票、专票、电子发票 - **roll**：可识别增值税卷票
seal_tag	否	string	true/false	是否开启印章判断功能，并返回印章内容的识别结果 - **true**：开启 - **false**：不开启

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/vat_invoice?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
ofd_file_size	否	string	传入OFD文件的总页数，当 ofd_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ ServiceType	是	string	发票消费类型。**不同消费类型输出：**餐饮、电器设备、通讯、服务、日用品食品、医疗、交通、其他
+ InvoiceType	是	string	发票种类。**不同类型发票输出**：普通发票、专用发票、电子普通发票、电子专用发票、通行费电子普票、区块链发票、通用机打电子发票、电子发票(专用发票)、电子发票(普通发票)
+ InvoiceTypeOrg	是	string	发票名称
+ InvoiceCode	是	string	发票代码
+ InvoiceNum	是	string	发票号码
+ InvoiceCodeConfirm	是	string	发票代码的辅助校验码，一般业务情景可忽略
+ InvoiceNumConfirm	是	string	发票号码的辅助校验码，一般业务情景可忽略
+ InvoiceNumDigit	是	string	数电票号，仅针对纸质的全电发票，在密码区有数电票号码的字段输出

+ InvoiceTag	是	string	增值税发票左上角标志。 包含：通行费、销项负数、代开、收购、成品油、其他
+ MachineNum	是	string	机打号码。仅增值税卷票含有此参数
+ MachineCode	是	string	机器编号。仅增值税卷票含有此参数
+ CheckCode	是	string	校验码
+ InvoiceDate	是	string	开票日期
+ PurchaserName	是	string	购方名称
+ PurchaserRegisterNum	是	string	购方纳税人识别号
+ PurchaserAddress	是	string	购方地址及电话
+ PurchaserBank	是	string	购方开户行及账号
+ Password	是	string	密码区
+ Province	是	string	省
+ City	是	string	市
+ SheetNum	是	string	联次信息。**专票**第一联到第三联分别输出：第一联：记账联、第二联：抵扣联、第三联：发票联；**普通发票**第一联到第二联分别输出：第一联：记账联、第二联：发票联
+ Agent	是	string	是否代开
+ CommodityName	是	array[]	货物名称
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityType	是	array[]	规格型号
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityUnit	是	array[]	单位
++ row	是	uint32	行号
++ word	是	string	内容

+ CommodityNum	是	array[]	数量
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityPrice	是	array[]	单价
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityAmount	是	array[]	金额
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTaxRate	是	array[]	税率
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTax	是	array[]	税额
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityPlateNum	是	array[]	车牌号。仅通行费增值税电子普通发票含有此参数
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityVehicleType	是	array[]	类型。仅通行费增值税电子普通发票含有此参数
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityStartDate	是	array[]	通行日期起。仅通行费增值税电子普通发票含有此参数
++ row	是	uint33	行号
++ word	是	string	内容
+ CommodityEndDate	是	array[]	通行日期止。仅通行费增值税电子普通发票含有此参数

++ row	是	uint33	行号
++ word	是	string	内容
+ OnlinePay	是	String	电子支付标识。仅区块链发票含有此参数
+ SellerName	是	string	销售方名称
+ SellerRegisterNum	是	string	销售方纳税人识别号
+ SellerAddress	是	string	销售方地址及电话
+ SellerBank	是	string	销售方开户行及账号
+ TotalAmount	是	uint32	合计金额
+ TotalTax	是	uint32	合计税额
+ AmountInWords	是	string	价税合计(大写)
+ AmountInFiguers	是	uint32	价税合计(小写)
+ Payee	是	string	收款人
+ Checker	是	string	复核
+ NoteDrawer	是	string	开票人
+ Remarks	是	string	备注
+ company_seal	否	string	判断是否存在印章。返回“0或1”，1代表存在印章，0代表不存在印章，当 seal_tag=true 时返回该字段
+ seal_info	否	string	印章识别结果内容。当 seal_tag=true 时返回该字段
+ PassengerName	是	array[]	出行人，仅旅客运输类发票有此参数，其余类型该参数返回为空
+ PassengerIdNum	是	array[]	有效身份证件号，仅旅客运输类发票有此参数，其余类型该参数返回为空
+ PassengerDate	是	array[]	出行日期，仅旅客运输类发票有此参数，其余类型该参数返回为空
+ PassengerDeparture	是	array[]	出发地，仅旅客运输类发票有此参数，其余类型该参数返回为空
+ PassengerArrival	是	array[]	到达地，仅旅客运输类发票有此参数，其余类型该参数返回为空

+ PassengerClass	是	array[]	等级，仅旅客运输类发票有此参数，其余类型该参数返回为空
+ PassengerVehicleType	是	array[]	交通工具类型，仅旅客运输类发票有此参数，其余类型该参数返回为空
+ TransportType	是	array[]	运输工具种类，仅货物运输类发票有此参数，其余类型该参数返回为空
+ TransportPlateNum	是	array[]	运输工具牌号，仅货物运输类发票有此参数，其余类型该参数返回为空
+ TransportDeparture	是	array[]	起运地，仅货物运输类发票有此参数，其余类型该参数返回为空
+ TransportArrival	是	array[]	到达地，仅货物运输类发票有此参数，其余类型该参数返回为空
+ TransportCargoInfo	是	array[]	运输货物名称，仅货物运输类发票有此参数，其余类型该参数返回为空

返回示例


```
{
 "log_id": "5425496231209218858",
"words_result_num": 36,
"words_result": {
      "InvoiceNumDigit": "123456",
      "ServiceType": "其他",
 "InvoiceNum": "14641426",
      "InvoiceNumConfirm": "14641426",
 "SellerName": "上海易火广告传媒有限公司",
 "CommodityTaxRate": [
  {
   "word": "6%",
   "row": "1"
  }
 ],
 "SellerBank": "中国银行南翔支行446863841354",
 "Checker": ":沈园园",
 "TotalAmount": "94339.62",
 "CommodityAmount": [
  {
   "word": "94339.62",
   "row": "1"
  }
 ],
 "InvoiceDate": "2016年06月02日",
 "CommodityTax": [
  {
   "word": "5660.38",
   "row": "1"
  }
 ],
```

```
  "PurchaserName": "百度时代网络技术(北京)有限公司",
  "CommodityNum": [
   {
    "word": "",
    "row": "1"
   }
  ],
        "Province": "上海",
        "City": "",
        "SheetNum": "第三联",
        "Agent": "否",
  "PurchaserBank": "招商银行北京分行大屯路支行8661820285100030",
  "Remarks": "告传",
  "Password": "074/45781873408>/6>8>65*887676033/51+<5415>9/32--852>1+29<65>641-
5>66<500>87/*-34<943359034>716905113*4242>",
  "SellerAddress": ":嘉定区胜辛南路500号15幢1161室55033753",
  "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二属A2010-59108001",
  "InvoiceCode": "3100153130",
        "InvoiceCodeConfirm": "3100153130",
  "CommodityUnit": [
   {
    "word": "",
    "row": "1"
   }
  ],
  "Payee": ":徐蓉",
  "PurchaserRegisterNum": "110108787751579",
  "CommodityPrice": [
   {
    "word": "",
    "row": "1"
   }
  ],
  "NoteDrawer": "沈园园",
  "AmountInWords": "壹拾万圆整",
  "AmountInFiguers": "100000.00",
  "TotalTax": "5660.38",
  "InvoiceType": "专用发票",
  "SellerRegisterNum": "913101140659591751",
  "CommodityName": [
   {
    "word": "信息服务费",
    "row": "1"
   }
  ],
  "CommodityType": [
   {
    "word": "",
    "row": "1"
   }
  ],
        "CommodityPlateNum": [],
        "CommodityVehicleType": [],
        "CommodityStartDate": [],
        "CommodityEndDate": [],
        "OnlinePay": ""
 }
}
```


增值税发票验真

接口描述

支持 14 种增值税发票的信息核验，包括增值税专票、电子专票、普票、电子普票、卷票、区块链发票（深圳地区）、全电发票（新版全国统一电子发票，专票/普票）、通行费增值税电子普通发票、货物运输业增值税专用发票、机动车销售发票、二手车销售发票、电子发票（航空运输电子客票行程单）、电子发票（铁路电子客票）等，支持返回票面的全部信息。同时可直接与同平台的发票识别能力对接，完成发票识别的同时进行自动化验真。

> 视频教程请参见 智能财务票据识别+增值税发票验真使用教程

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/vat_invoice_verification`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
invoice_code	是	string	-	发票代码。 全电发票（专用发票）、全电发票（普通发票）、全电发票（含通行费标识）、电子发票（机动车销售统一发票）、电子发票（二手车销售统一发票）此参数可为空，其他类型发票均不可为空
invoice_num	是	string	-	发票号码
invoice_date	是	string	-	开票日期。格式YYYYMMDD，例：20210101
			增值税专用发票：special_vat_invoice 增值税电子专用发票：elec_special_vat_invoice 增值税普通发票：normal_invoice 增值税普通发票（电子）：elec_normal_invoice 增值税普通发票（卷式）：roll_normal_invoice 通行费增值税电子普通发票：toll_elec_normal_invoice 区块链电子发票（目前仅支持深圳地区）：blockchain_invoice	

invoice_type	是	string	全电发票（专用发票）：elec_invoice_special 全电发票（普通发票）：elec_invoice_normal 货运运输业增值税专用发票：special_freight_transport_invoice 机动车销售发票/电子发票（纸质机动车销售统一发票）/电子发票（机动车销售统一发票）：motor_vehicle_invoice 二手车销售发票/电子发票（纸质二手车销售统一发票）/电子发票（二手车销售统一发票）：used_vehicle_invoice 电子发票（航空运输电子客票行程单）：elec_flight_itinerary_invoice 电子发票（铁路电子客票）：elec_train_ticket_invoice 全电发票（含通行费标识）：elec_toll_invoice	发票种类
check_code	是	string	-	校验码。填写发票校验码后6位。 增值税电子专票、普票、电子普票、卷票、区块链电子发票、通行费增值税电子普通发票**此参数必填**； 其他类型发票**此参数可为空**
total_amount	是	string	-	发票金额。 增值税专票、电子专票、区块链电子发票、机动车销售发票、电子发票（纸质机动车销售统一发票）、货运专票填写**不含税金额**； 二手车销售发票、电子发票（纸质二手车销售统一发票）、电子发票（二手车销售统一发票）填写**车价合计**； 全电发票（专用发票）、全电发票（普通发票）、铁路电子客票（电子发票）、电子发票（机动车销售统一发票）、全电发票（含通行费标识）填写**价税合计金额**，其他类型发票可为空

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash
Python
JAVA
PHP
C#
C++

OCR-增值税发票验真

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/vat_invoice_verification?access_token=【调用鉴权接口获取的token】' --data 'invoice_code=发票代码&invoice_num=发票号码&invoice_date=开票日期&check_code=校验码。填写发票校验码后6位&invoice_type=发票类型&total_amount=不含税金额' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
VerifyResult	是	string	查验结果。查验成功返回“0001”，查验失败返回对应查验结果错误码，详见末尾表格
VerifyMessage	是	string	查验结果信息。查验成功且发票为真返回“查验成功发票一致“，查验失败返回对应错误原因，详见末尾表格
VerifyFrequency	是	string	查验次数。为历史查验次数
InvalidSign	是	string	发票状态。Y：已作废；H：已冲红；N：未作废；BH：部分红冲；QH：全额红冲
InvoiceType	是	string	发票种类。即增值税专用发票、增值税电子专用发票、增值税普通发票、增值税普通发票（电子）、增值税普通发票（卷式）、通行费增值税电子普通发票、区块链电子发票、全电发票（专用发票）、全电发票（普通发票）、机动车销售发票、电子发票（机动车销售统一发票）、电子发票（纸质二手车销售统一发票）、二手车销售发票、电子发票（二手车销售统一发票）、货物运输业增值税专用发票、电子发票（航空运输电子客票行程单）、电子发票（铁路电子客票）、全电发票（含通行费标识）
InvoiceCode	是	string	发票代码
InvoiceNum	是	string	发票号码
CheckCode	是	string	校验码
InvoiceDate	是	string	开票日期
MachineCode	是	string	机器编号

增值税专票、电子专票、普票、电子普通发票、卷票、通行费增值税电子普通发票、货物运输业增值税专用发票、全电发票（含通行费标识）返回信息

字段	是否必选	类型	说明
+ PurchaserName	是	string	购方名称
+ PurchaserRegisterNum	是	string	购方纳税人识别号
+ PurchaserAddre	是	string	购方地址及电话

ss			
+ PurchaserBank	是	string	购方开户行及账号
+ CommodityName	是	array[]	货物名称/项目名称
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityType	是	array[]	规格型号
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityUnit	是	array[]	单位
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityNum	是	array[]	数量
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityPrice	是	array[]	单价
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityAmount	是	array[]	金额
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTaxRate	是	array[]	税率
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTax	是	array[]	税额
++ row	是	uint32	行号
++ word	是	string	内容
+ SellerName	是	string	销售方名称
+ SellerRegisterNum	是	string	销售方纳税人识别号
+ SellerAddress	是	string	销售方地址及电话
+ SellerBank	是	string	销售方开户行及账号

+ TotalAmount	是	string	合计金额
+ TotalTax	是	string	合计税额
+ AmountInFiguers	是	string	价税合计（小写）
+ TollSign	是	string	通行费标志。Y-可抵扣通行费，N-不可抵扣通行费。通行费增值税电子普通发票返回信息，其他类型发票可忽略
+ ZeroTaxRateIndicator	是	string	零税率标识。空：非零税率,1：税率栏位显示“免税”，2：税率栏位显示“不征税”，3：零税率。通行费增值税电子普通发票返回信息，其他类型发票可忽略
+ CommodityPlateNum	是	array[]	车牌号。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityVehicleType	是	array[]	类型。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityStartDate	是	array[]	通行日期起。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityEndDate	是	array[]	通行日期止。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ Carrier	是	string	承运人名称。货运专票返回信息，其他类型发票可忽略
+ CarrierCode	是	string	承运人识别号。货运专票返回信息，其他类型发票可忽略
+ Recipient	是	string	受票方名称。货运专票返回信息，其他类型发票可忽略
+ RecipientCode	是	string	受票方识别号。货运专票返回信息，其他类型发票可忽略
+ Receiver	是	string	收货人名称。货运专票返回信息，其他类型发票可忽略
+ ReceiverCode	是	string	收货人识别号。货运专票返回信息，其他类型发票可忽略
+ Sender	是	string	发货人名称。货运专票返回信息，其他类型发票可忽略
+ SenderCode	是	string	发货人识别号。货运专票返回信息，其他类型发票可忽略
+ TransportCargoInformation	是	string	运输货物信息。货运专票返回信息，其他类型发票可忽略
+ DepartureViaArrival	是	string	起运地、经由、到达地。货运专票返回信息，其他类型发票可忽略

+ TaxControlNum	是	string	税控盘号。货运专票返回信息，其他类型发票可忽略
+ VehicleType	是	string	车种车号。货运专票返回信息，其他类型发票可忽略
+ VehicleTonnage	是	string	车船吨位。货运专票返回信息，其他类型发票可忽略
+ CommodityExpenseItem	是	array[]	费用项目。货运专票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ NoteDrawer	是	string	开票人（因税局不返回此字段，故字段为空，暂做保留使用）
+ Checker	是	string	复核（因税局不返回此字段，故字段为空，暂做保留使用）
+ Payee	是	string	收款人（因税局不返回此字段，故字段为空，暂做保留使用）
+ Remarks	是	string	备注
+ ESVATURL	是	string	增值税电子专票（即 ofd 发票）的下载地址
+ ListLabel	是	string	清单标识，Y：带清单；N：无清单； 说明：只有当发票种类为：增值税专票，电子专票，普票，电子普通发票时返回此字段的值

机动车销售发票（包含电子发票（纸质机动车销售统一发票）、电子发票（机动车销售统一发票））返回信息

字段	是否必选	类型	说明
+ Purchaser	是	string	购买方名称
+ PurchaserCode	是	string	购买方身份证号/组织机构代码
+ VehicleType	是	string	车辆类型
+ ManuModel	是	string	厂牌型号
+ Origin	是	string	产地
+ CertificateNum	是	string	合格证号书
+ CommodityInspectionNum	是	string	商检单号
+ EngineNum	是	string	发动机号码
+ VinNum	是	string	车辆识别代号/车架号码
+ ImportCertificateNum	是	string	进口证明书号
+ TaxPaymentVoucherNum	是	string	完税凭证号码
+ LimitPassenger	是	string	限乘人数
+ TaxAuthor	是	string	主管税务机关名称
+ TaxAuthorCode	是	string	主管税务机关代码
+ Tonnage	是	string	吨位
+ Price	是	string	不含税价格
+ TaxRate	是	string	税率
+ Tax	是	string	税额
+ PriceTaxLow	是	string	价税合计
+ Saler	是	string	销货单位名称
+ SalerCode	是	string	销货单位纳税人识别号
+ SalerBank	是	string	销货单位开户银行
+ SalerAccountNum	是	string	销货单位账号
+ SalerPhone	是	string	销货单位电话

二手车销售发票（包含电子发票（纸质二手车销售统一发票）、电子发票（二手车销售统一发票））返回信息

字段	是否必选	类型	说明
+ Purchaser	是	string	买方单位/个人
+ PurchaserCode	是	string	买方单位代码/身份证号
+ PurchaserAddress	是	string	买方单位/个人住址
+ PurchaserPhone	是	string	买方电话
+ Saler	是	string	卖方单位/个人
+ SalerCode	是	string	卖方单位代码/身份证号
+ SalerAddress	是	string	卖方单位/个人住址
+ SalerPhone	是	string	卖方电话
+ LicensePlateNum	是	string	车牌照号
+ RegistrationCode	是	string	登记证号
+ TotalCarPrice	是	string	车价合计
+ TransferVehicleManagementOffice	是	string	转入地车辆车管所名称
+ VehicleType	是	string	车辆类型
+ ManuModel	是	string	厂牌型号
+ VinNum	是	string	车辆识别代号/车架号码
+ Operator	是	string	经营、拍卖单位
+ OperatorAddress	是	string	经营、拍卖单位地址
+ OperatorCode	是	string	经营、拍卖单位纳税人识别号
+ OperatorBank	是	string	开户银行及账号
+ OperatorPhone	是	string	经营、拍卖单位电话
+ UsedCarMarket	是	string	二手车市场
+ UsedCarMarketCode	是	string	二手车市场纳税人识别号
+ UsedCarMarketAddress	是	string	二手车市地址
+ UsedCarMarketBank	是	string	二手车市场开户银行及账号
+ UsedCarMarketPhone	是	string	二手车市场电话

航空运输电子客票行程单（电子发票）返回信息

字段	是否必选	类型	说明
+ purchaser_name	是	string	购方名称
+ purchaser_register_num	是	string	购方纳税人识别号
+ purchaser_address	是	string	购方地址及电话
+ purchaser_bank	是	string	购方开户行及账号
+ seller_name	是	string	销方名称
+ seller_register_num	是	string	销方纳税人识别号
+ seller_address	是	string	销方地址及电话
+ seller_bank	是	string	销方开户行及账号
+ price_tax_low	是	string	价税合计
+ total_tax	是	string	合计税额
+ name	是	string	旅客姓名
+ id_num	是	string	旅客身份证号
+ ticket_num	是	string	电子客票号
+ identification	是	string	国内国际标识
+ gp_num	是	string	GP 单号
+ flight_segment	是	array[]	航段
+ origin	是	array[]	始发地
+ destination	是	array[]	目的地
+ carrier	是	array[]	承运人
+ flight_num	是	array[]	航班号
+ class	是	array[]	座位等级
+ date	是	array[]	日期
+ time	是	array[]	起飞时间
+ fare_basis	是	array[]	客票级别/客票类别

铁路电子客票（电子发票）返回信息

字段	是否必选	类型	说明
+ purchaser_name	是	string	购方名称
+ purchaser_register_num	是	string	购方纳税人识别号
+ purchaser_address	是	string	购方地址及电话
+ purchaser_bank	是	string	购方开户行及账号
+ seller_name	是	string	销方名称
+ seller_register_num	是	string	销方纳税人识别号
+ seller_address	是	string	销方地址及电话
+ seller_bank	是	string	销方开户行及账号
+ commodity_tax	是	string	发票税额
+ commodity_tax_rate	是	string	税率
+ commodity_amount	是	string	发票金额
+ fare	是	string	票价
+ name	是	string	旅客姓名
+ id_num	是	string	旅客身份证号
+ sales_type	是	string	业务类型，可输出： 1-售； 2-退
+ starting_station	是	string	出发站
+ destination_station	是	string	到达站
+ train_num	是	string	车次
+ date	是	string	乘车日期
+ time	是	string	出发时间
+ seat_category	是	string	席别
+ carriage_num	是	string	车厢号
+ seat_num	是	string	席位号
+ ticket_num	是	string	电子客票号
+ air_condition	是	string	空调特征

查验结果码释义表

查验结果（VerifyResult）	查验结果信息（VerifyMessage）	描述
9999	查验失败	查验失败，业务出现异常，请提交工单咨询
0002	超过该张票当天查验次数	此发票今日查询次数已达上限（5次），请次日查询
0005	请求不合法	发票入参格式有误，请核对后再查询
0006	发票信息不一致	发票信息有误，请核对后再查询
0009	发票不存在	所查发票不存在
1004	已超过最大查验量	已超过最大查验量，请提交工单咨询
1005	查询发票不规范	信息有误，请核对后再查询
1006	查验异常	发票信息有误，请核对后再查询
1008	字段不能为空	发票请求参数不能为空
1009	参数长度不正确	参数长度不符合规范，确认参数，再次查验
1014	日期当天的不能查验	日期当天的不能查验，请隔天再查
1015	超过5年的不能查验	超过5年的不能查验
1020	没有查验权限	没有查验权限，请提交工单咨询
1021	网络超时	税局维护升级，暂时无法查验，请提交工单咨询

返回示例


```
 // 增值税专票、电子专票、普票、电子普通发票、卷票、通行费增值税电子普通发票、货物运输业增值税专用发票
{
    "words_result": {
        "log_id": 1394226734160674816,
        "words_result_num": 43,
        "VerifyFrequency": "3",
        "VerifyMessage": "查验成功发票一致",
        "InvalidSign": "N",
        "InvoiceType": "增值税普通发票（电子）",
        "MachineCode": "661616300747",
        "CheckCode": "67820461013285253079",
        "InvoiceCode": "043002000111",
        "InvoiceDate": "20210503",
        "VerifyResult": "0001",
        "InvoiceNum": "63509760"
        "TaxControlNum": "",
        "CommodityEndDate": [
            {
                "row": "1",
                "word": ""
            }
        ],
        "VehicleTonnage": "",
        "CommodityVehicleType": [
            {
                "row": "1"
            }
        ],
        "CommodityStartDate": [
            {
                "row": "1",
                "word": ""
            }
        ],
        "SellerAddress": "湖南省长沙市天心区芙蓉中路三段446号0731-83592079",
        "CommodityPrice": [
```

```
        {
            "row": "1",
            "word": "28.20000000"
        }
    ],
    "TransportCargoInformation": "",
    "NoteDrawer": "",
    "CommodityNum": [
        {
            "row": "1",
            "word": "1.00000000"
        }
    ],
    "SellerRegisterNum": "9143010071219848l2",
    "SellerBank": "建行长沙铁银支行营业部43001710661050003739",
    "Remarks": "账期:202104",
    "TotalTax": "0.00",
    "CommodityTaxRate": [
        {
            "row": "1",
            "word": "不征税"
        }
    ],
    "CommodityExpenseItem": [
        {
            "row": "1",
            "word": ""
        }
    ],
    "ZeroTaxRateIndicator": "",
    "Carrier": "",
    "SenderCode": "",
    "PurchaserRegisterNum": "911101087877515792",
    "ReceiverCode": "",
    "AmountInFiguers": "28.20",
    "PurchaserBank": "招商银行北京分行大屯路支行 866182028510003",
    "Checker": "",
    "TollSign": "",
    "VehicleTypeNum": "",
    "DepartureViaArrival": "",
    "Receiver": "",
    "Recipient": "",
    "TotalAmount": "28.20",
    "CommodityAmount": [
        {
            "row": "1",
            "word": "28.20"
        }
    ],
    "PurchaserName": "百度时代网络技术（北京）有限公司",
    "CommodityType": [
        {
            "row": "1",
            "word": ""
        }
    ],
    "Sender": "",
    "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二层A201059108001",
    "CommodityTax": [
        {
            "row": "1",
            "word": "***"
        }
```

```
        }
      ],
      "CarrierCode": "",
      "CommodityPlateNum": [
        {
          "row": "1",
          "word": ""
        }
      ],
      "CommodityUnit": [
        {
          "row": "1",
          "word": ""
        }
      ],
      "Payee": "",
      "RecipientCode": "",
      "CommodityName": [
        {
          "row": "1",
          "word": "*电信服务*通讯费服务费"
        }
      ],
      "SellerName": "中国移动通信集团湖南有限公司长沙分公司"
    },
}
// 机动车销售发票
{
 "words_result": {
  "log_id": 1394232842988290048,
  "words_result_num": 24,
  "VerifyFrequency": "1",
  "VerifyMessage": "查验成功发票一致",
  "InvalidSign": "N",
  "InvoiceType": "机动车销售统一发票",
  "MachineCode": "539927983",
  "CheckCode": "",
  "InvoiceCode": "13200378019836",
  "InvoiceDate": "20210128",
  "VerifyResult": "0001",
  "InvoiceNum": "00342061",
  "Origin": "中国",
  "ManuModel": "东风日产牌DFL8",
  "SalerBank": "工行支行",
  "VehicleType": "多用途乘用车",
  "Tax": "18238.29",
  "TaxPaymentVoucherNum": "",
  "CommodityInspectionNum": "",
  "TaxAuthorCode": "1332803841100",
  "VinNum": "LGBM464574",
  "SalerPhone": "0513-8237861",
  "LimitPassenger": "5",
  "PurchaserCode": "211402199410176136",
  "TaxAuthor": "国家税务总局海门市税务局三厂税务分局",
  "Tonnage": "",
  "ImportCertificateNum": "",
  "Saler": "海门市海通汽车销售服务有限公司",
  "SalerAccountNum": "1111527109002888833",
  "Price": "145840.71",
  "CertificateNum": "WAC224003769810",
  "TaxRate": "13%",
  "Purchaser": "郑如意",
  "SalerCode": "913206847828000000 7164",
```

```
  "EngineNum": "43380M",
  "PriceTaxLow": "1323800"
 },
// 二手车销售发票
{
 "words_result": {
  "log_id": 1394233936539811840,
  "words_result_num": 25,
  "VerifyFrequency": "1",
  "VerifyMessage": "查验成功发票一致",
  "InvalidSign": "N",
  "InvoiceType": "二手车销售统一发票",
  "MachineCode": "66173004789204",
  "CheckCode": "",
  "InvoiceCode": "0323789200007",
  "InvoiceDate": "20200509",
  "VerifyResult": "0001",
  "InvoiceNum": "002890341",
  "Operator": "",
  "TransferVehicleManagementOffice": "苏州市车管所",
  "ManuModel": "JF1SH95F",
  "RegistrationCode": "3200478903518",
  "OperatorPhone": "",
  "PurchaserCode": "320503782902308u425",
  "Saler": "张散文",
  "UsedCarMarketCode": "91320378038NCQUQXA",
  "Purchaser": "张丽",
  "OperatorCode": "",
  "UsedCarMarketBank": "中国农业银行股份有限公司苏州分行清算中心10549001040001493",
  "SalerAddress": "江苏省苏州市工业园区倪浜路3号",
  "SalerCode": "411524199001016511",
  "PurchaserPhone": "0",
  "LicensePlateNum": "苏U1A666",
  "VehicleType": "小型越野客车",
  "OperatorBank": "",
  "OperatorAddress": "",
  "VinNum": "JF1SH78006596636",
  "TotalCarPrice": "66000.00",
  "SalerPhone": "",
  "PurchaserAddress": "江苏省苏州市相城区元和莫阳村",
  "UsedCarMarketPhone": "13182680222",
  "UsedCarMarketAddress": "苏州高新区长江路668号(3号厂房)",
  "UsedCarMarket": "苏州车市界二手车电子商务有限公司"
 },
}
```


银行回单识别

接口描述

支持对不同版式银行回单进行结构化识别，包括标题、付款人户名、付款人开户银行、付款人账号、收款人户名、收款人开户银行、收款人账号、大写金额、小写金额、流水号、回单编号、交易日期、摘要、用途 14个关键字段。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/bank_receipt_new`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	string	true/false	是否返回字段置信度，默认为 false ，即不返回
location	否	string	true/false	是否返回字段位置坐标，默认为 false，即不返回

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/bank_receipt_new?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ word	是	string	字段识别结果，对应**标题、付款人户名、付款人开户银行、付款人账号、收款人户名、收款人开户银行、收款人账号、大写金额、小写金额、流水号、回单编号、交易日期、摘要、用途** 14个字段的识别结果
+ location	否	object	字段位置信息，当请求参数 location=true 时返回该字段
++ top	否	uint32	字段的上边距
++ left	否	uint32	字段的左边距
++ height	否	uint32	字段的高度
++ width	否	uint32	字段的宽度
+ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时返回该字段
++ average	否	float	字段识别结果中各字符的置信度平均值
++ min	否	float	字段识别结果中各字符的置信度最小值

返回示例


```
{
  "words_result_num": 14,
  "words_result": {
    "标题": [
      {
        "word": "中国工商银行网上银行电子回单"
      }
    ],
    "付款人户名": [
      {
        "word": "北京小度网络科技有限公司"
      }
    ],
    "付款人开户银行": [
      {
        "word": "北京开发区支行营业室"
      }
    ],
    "付款人账号": [
      {
        "word": "3208095919200208008"
      }
```

```
        ],
        "收款人户名": [
            {
                "word": "上海小度网络科技有限公司"
            }
        ],
        "收款人开户银行": [
            {
                "word": "工行前海支行"
            }
        ],
        "收款人账号": [
            {
                "word": "3707020802702372707"
            }
        ],
        "大写金额": [
            {
                "word": "人民币壹万陆仟贰佰叁拾玖元叁角"
            }
        ],
        "小写金额": [
            {
                "word": "￥16239.30元"
            }
        ],
        "流水号": [
            {
                "word": "27208770"
            }
        ],
        "回单编号": [
            {
                "word": "2085758711703264"
            }
        ],
        "交易日期": [
            {
                "word": ""
            }
        ],
        "摘要": [
            {
                "word": "货款"
            }
        ],
        "用途": [
            {
                "word": ""
            }
        ]
    },
    "log_id": 1630759834809123854
}
```


定额发票识别

接口描述

支持对各类定额发票的发票代码、发票号码、金额、发票所在地、发票金额小写、省、市7个关键字段进行结构化识

别。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=财务票据OCR&api=rest/2.0/ocr/v1/quota_invoice&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/quota_invoice`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/quota_invoice?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
定额发票识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/quota_invoice"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 定额发票识别
*/
public class QuotaInvoice {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String quotaInvoice() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/quota_invoice";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        QuotaInvoice.quotaInvoice();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/quota_invoice";
static std::string quotaInvoice_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    quotaInvoice_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 定额发票识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int quotaInvoice(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = quotaInvoice_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/quota_invoice?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class QuotaInvoice
    {
        // 定额发票识别
        public static string quotaInvoice()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/quota_invoice?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("定额发票识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果数组
+ invoice_code	否	string	发票代码
+ invoice_number	否	string	发票号码
+ invoice_rate	否	string	金额
+ location	否	string	发票所在地
+ invoice_rate_lowercase	否	string	发票金额小写
+ province	否	string	省
+ city	否	string	市
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

|

返回示例

```JSON
{
   "log_id": 2480896295,
   "words_result_num": 3,
   "words_result": {
      "invoice_code": "132081490320",
      "invoice_number": "01275486",
      "invoice_rate": "伍拾元整"
      "invoice_rate_lowercase": "100.00",
      "invoice_code": "161034127200",
      "province": "陕西",
   }
}
```

通用机打发票识别

接口描述

支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别，包括发票类型、发票号码、发票代码、开票日期、合计金额大写、合计金额小写、商品名称、商品单位、商品单价、商品数量、商品金额、机打代码、机打号码、校验码、销售方名称、销售方纳税人识别号、购买方名称、购买方纳税人识别号、合计税额等。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=财务票据OCR&api=rest/2.0/ocr/v1/invoice&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/invoice`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级

**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效 |
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/invoice?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
通用机打发票识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/invoice"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 通用机打发票识别
*/
public class Invoice {
~~~

```
public class Invoice {

    /**
     * 重要提示代码中所需工具类
     * FileUtil,Base64Util,HttpUtil,GsonUtils请从
     * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
     * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
     * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
     * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
     * 下载
     */
    public static String invoice() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/invoice";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        Invoice.invoice();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/invoice";
static std::string invoice_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    invoice_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 通用机打发票识别
* @return 调用成功返回0，发生错误返回其他错误码
```

```
*/
int invoice(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = invoice_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP

<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
```

```
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/invoice?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class Invoice
    {
        // 通用机打发票识别
        public static string invoice()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/invoice?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("通用机打发票识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

```
}


```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	是	int32	图像方向。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ InvoiceType	否	string	发票类型
+ InvoiceCode	否	string	发票代码
+ InvoiceNum	否	string	发票号码
+ InvoiceDate	否	string	开票日期
+ AmountInFiguers	否	string	合计金额小写
+ AmountInWords	否	string	合计金额大写
+ CommodityName	否	array[]	商品名称
++ row	否	unit32	行号
++ word	否	string	内容
+ CommodityUnit	否	array[]	商品单位
++ row	否	unit32	行号
++ word	否	string	内容
+ CommodityPrice	否	array[]	商品单价
++ row	否	unit32	行号
++ word	否	string	内容
+ CommodityNum	否	array[]	商品数量
++ row	否	unit32	行号
++ word	否	string	内容
+ CommodityAmount	否	array[]	商品金额
++ row	否	unit32	行号
++ word	否	string	内容
+ IndustrySort	否	string	行业分类
+ MachineNum	否	string	机打号码
+ CheckCode	否	string	校验码
+ SellerName	否	string	销售方名称
+ [illegible]	否	[illegible]	销售方纳税人识别号

+ SellerRegisterNum	否	string	销售方纳税人识别号
+ PurchaserName	否	string	购买方名称
+ PurchaserRegisterNum	否	string	购买方纳税人识别号
+ TotalTax	否	string	合计税额
+ Province	否	string	省
+ City	否	string	市
+ Time	否	string	时间
+ SheetNum	否	string	联次
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例


```
{
  "log_id": 4423022131715883558,
  "direction": 0,
  "words_result_num": 22,
  "words_result": {
     "City": "",
     "InvoiceNum": "01445096",
     "SellerName": "百度餐饮店",
     "IndustrySort": "生活服务",
     "Province": "广东省",
     "CommodityAmount": [
        {
           "word": "183.00",
           "row": "1"
        }
     ],
     "InvoiceDate": "2020年07月28日",
     "PurchaserName": "中信建投证券股份有限公司",
     "CommodityNum": [],
     "InvoiceCode": "144001901511",
     "CommodityUnit": [],
     "SheetNum": "",
     "PurchaserRegisterNum": "9144223008453480X9",
     "Time": "",
     "CommodityPrice": [],
     "AmountInFiguers": "183.00",
     "AmountInWords": "壹佰捌拾叁元整",
     "CheckCode": "61042119820421061301",
     "TotalTax": "183.00",
     "InvoiceType": "广东通用机打发票",
     "SellerRegisterNum": "61042119820421061301",
     "CommodityName": [
        {
           "word": "餐费",
           "row": "1"
        }
     ]
  }
}
```


火车票识别

接口描述

支持对红/蓝火车票、铁路电子客票的关键字段进行结构化识别，包括车票号码、始发站、目的站、车次、日期、票价、席

别、姓名、座位号、身份证号、售站、序列号、时间。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/train_ticket`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必须	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

```
Bash
```

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/train_ticket?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

参数	是否必须	类型	说明
log_id	是	uint64	请求标识码，唯一，用于调用失败后进行问题定位
direction	是	int32	图像方向 - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result	是	object{}	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ ticket_num	是	string	车票号
+ starting_station	是	string	始发站
+ train_num	是	string	车次号
+ destination_station	是	string	到达站
+ date	是	string	出发日期
+ ticket_rates	是	string	车票金额
+ seat_category	是	string	席别
+ name	是	string	乘客姓名
+ id_num	是	string	身份证号
+ serial_number	是	string	序列号
+ sales_station	是	string	售站
+ time	是	string	时间
+ seat_num	是	string	座位号
+ refund_flag	否	string	退票标识，仅在输入为电子火车票时返回该字段
+ invoice_num	否	string	发票号码，仅在输入为电子火车票时返回该字段
+ invoice_date	否	string	开票日期，仅在输入为电子火车票时返回该字段
+ fare	否	string	不含税金额，仅在输入为电子火车票时返回该字段
+ tax_rate	否	string	税率，仅在输入为电子火车票时返回该字段
+ tax	否	string	税额，仅在输入为电子火车票时返回该字段
+ elec_ticket_num	否	string	电子客票号，仅在输入为电子火车票时返回该字段
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
"log_id": "12317512659",
"direction": 1,
"words_result_num": 13,
"words_result": {
 "id_num": "2302051998****156X",
 "name": "裴一丽",
 "ticket_rates": "￥54.5元",
 "destination_station": "天津站",
 "seat_category": "二等座",
 "sales_station": "北京南",
 "ticket_num": "F05706",
 "seat_num": "02车03C号",
 "time": "09:36",
 "date": "2019年04月03日",
 "serial_number": "10010300067846",
 "train_num": "C255",
 "starting_station": "北京南站"
}
}
```


出租车票识别

接口描述

支持识别全国各大城市出租车票的 16 个关键字段，包括发票号码、代码、车号、日期、总金额、燃油附加费、叫车服务费、省、市、单价、里程、上车时间、下车时间等。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/taxi_receipt`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必须	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/taxi_receipt?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

参数	是否必须	类型	说明
log_id	是	uint64	请求标识码，随机数，唯一。
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果数组
+ InvoiceCode	是	string	发票代码
+ InvoiceNum	是	string	发票号码
+ TaxiNum	是	string	车牌号
+ Date	是	string	日期
+ Time	是	string	上下车时间
+ PickupTime	是	string	上车时间
+ DropoffTime	是	string	下车时间
+ Fare	是	string	金额
+ FuelOilSurcharge	是	string	燃油附加费
+ CallServiceSurcharge	是	string	叫车服务费
+ TotalFare	是	string	总金额
+ Location	是	string	开票城市
+ Province	是	string	省
+ City	是	string	市
+ PricePerkm	是	string	单价
+ Distance	是	string	里程
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
"log_id":2034039896,
"words_result_num":16,
"words_result":
{
 "Date":"2017-11-26",
 "Fare":"¥153.30元",
      "Location": "北京",
 "InvoiceCode":"111001681009",
 "InvoiceNum":"90769610",
 "TaxiNum":"BV2062",
 "Time":"20:42-21:07",
      "PickupTime":"20:42",
      "DropoffTime":"21:07",
 "FuelOilSurcharge": "¥1.00",
 "CallServiceSurcharge": "¥1.00",
      "TotalFare":"¥155.30元",
      "Province": "北京",
      "City": "北京市",
      "PricePerkm": "2.50元/KM",
      "Distance": "4.5KM"
 }
}
```


飞机行程单识别

接口描述

支持对飞机行程单的24个字段进行结构化识别，包括电子客票号、印刷序号、姓名、始发站、目的站、航班号、日期、时间、票价、身份证号、承运人、民航发展基金、保险费、燃油附加费、其他税费、合计金额、填开日期、订票渠道、客票级别、座位等级、销售单位号、签注、免费行李、验证码。 同时，支持单张行程单上的多航班信息识别。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/air_ticket`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
multi_detect	否	string	true/false	控制是否开启多航班信息识别功能，**默认值：false** - **true：开启多航班信息识别功能**，开启后返回结果中对应字段格式将改为数组类型 - **false：** 不开启，仅识别单一航班信息

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/air_ticket?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ name	是	string	姓名
+ starting_station	是	string	始发站
+ destination_station	是	string	目的站
+ flight	是	string	航班号
+ date	是	string	日期
+ ticket_number	是	string	电子客票号码
+ fare	是	string	票价
+ dev_fund	是	string	民航发展基金/机建费
+ fuel_surcharge	是	string	燃油附加费
+ other_tax	是	string	其他税费
+ ticket_rates	是	string	合计金额
+ issued_date	是	string	填开日期
+ id_num	是	string	身份证号
+ carrier	是	string	承运人
+ time	是	string	时间
+ issued_by	是	string	填开单位
+ serial_number	是	string	印刷序号
+ insurance	是	string	保险费
+ fare_basis	是	string	客票级别
+ class	是	string	座位等级
+ agent_code	是	string	销售单位号
+ endorsement	是	string	签注
+ allow	是	string	免费行李
+ ck	是	string	验证码
+ effective_date	是	string	客票生效日期
+ expiration_date	是	string	有效期截止日期
+ invoice_num	是	string	发票号码
+ commodity_tax_rate	是	string	增值税税率
+ commodity_tax	是	string	增值税税额
+ purchaser_name	是	string	购买方名称
+ purchaser_register_num	是	string	统一社会信用代码/纳税人识别号
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
  // 识别单航班信息（multi_detect=false，或参数缺省）
{
```

```
    "log_id": 7306800033425229106,
    "words_result_num": 18,
    "words_result": {
        "insurance": "20.00",
        "date": "2019-10-22",
        "allow": "20K",
        "flight": "CA6589",
        "issued_by": "中国国际航空服务有限公司",
        "starting_station": "武汉",
        "fare": "260.00",
        "endorsement": "不得签转改期退转",
        "ticket_rates": "350.00",
        "ck": "5866",
        "serial_number": "51523588676",
        "ticket_number": "7843708871196",
        "fuel_surcharge": "EXEMPT",
        "carrier": "南航",
        "issued_date": "2019-10-30",
        "other_tax": "",
        "fare_basis": "NREOW",
        "id_num": "411201123909020877",
        "destination_station": "合肥",
        "name": "郭达",
        "agent_code": "BJS19197300025",
        "time": "21:25",
        "class": "N",
        "dev_fund": "50.00"
    }
}

// 识别多航班信息（multi_detect=true)
{
    "words_result": {
        "log_id": "1280814270572920832",
        "words_result_num": 18
        "insurance": [
            {
                "word": "XXX"
            }
        ],
        "date": [
            {
                "word": "2019-10-18"
            },
            {
                "word": "2019-10-21"
            }
        ],
        "flight": [
            {
                "word": "CZ3565"
            },
            {
                "word": "CZ3566"
            }
        ],
        "issued_by": [
            {
                "word": "上海携程旅行社有限公司"
            }
        ],
        "starting_station": [
```

```
            {
                "word": "北京"
            }
        ],
        "fare": [
            {
                "word": "1080.00"
            }
        ],
        "ticket_rates": [
            {
                "word": "1420.00"
            }
        ],
        "serial_number": [
            {
                "word": "45956029770"
            }
        ],
        "ticket_number": [
            {
                "word": "7849648364314"
            }
        ],
        "fuel_surcharge": [
            {
                "word": "240.00"
            }
        ],
        "carrier": [
            {
                "word": "南航"
            },
            {
                "word": "南航"
            }
        ],
        "issued_date": [
            {
                "word": "2019-09-18"
            }
        ],
        "other_tax": [],
        "id_num": [
            {
                "word": "0789654700"
            }
        ],
        "destination_station": [
            {
                "word": "深圳"
            },
            {
                "word": "北京"
            }
        ],
        "name": [
            {
                "word": "姚佳"
            }
        ],
        "time": [
            {
```

```
            {
                "word": "13:55"
            },
            {
                "word": "16:30"
            }
        ],
        "dev_fund": [
            {
                "word": "100.00"
            }
        ]
    },
}
```


汽车票识别

接口描述

支持对全国范围不同版式汽车票的发票代码、发票号码、到达站、出发站、日期、时间、金额、身份证号、姓名、开票日期10个字段进行结构化识别。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/bus_ticket`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必须	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/bus_ticket?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

参数	是否必须	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ InvoiceCode	是	string	发票代码
+ InvoiceNum	是	string	发票号码
+ Date	是	string	日期
+ Time	是	string	时间
+ StartingStation	是	string	出发站
+ Fare	是	string	金额
+ IdNum	是	string	身份证号
+ DestinationStation	是	string	到达站
+ Name	是	string	姓名

返回示例


```
{
  "words_result": {
     "DestinationStation": "郑州西站",
     "IdNum": "424133********3014",
     "Time": "08:30",
     "InvoiceCode": "145021822522",
     "InvoiceNum": "1450374239",
     "StartingStation": "北京南站",
     "Date": "2020-01-24",
     "Name": "王文",
     "Fare": "69.00"
  },
  "log_id": "1274914257661591552",
  "words_result_num": 9
}
```

过路过桥费发票识别

接口描述

支持对全国范围不同版式过路、过桥费发票的发票代码、发票号码、入口、出口、日期、时间、金额、省、市9个字段进行结构化识别。

在线调试
您可以在 |示例代码中心|(https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=财务票据OCR&api=rest/2.0/ocr/v1/toll_invoice&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: ` https://aip.baidubce.com/rest/2.0/ocr/v1/toll_invoice`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必须	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。
~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/toll_invoice?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python
 ##### encoding:utf-8

import requests
import base64

'''
过路过桥费发票识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/toll_invoice"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 过路过桥费发票识别
*/
public class TollInvoice {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String tollInvoice() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/toll_invoice";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        TollInvoice.tollInvoice();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/toll_invoice";
static std::string tollInvoice_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    tollInvoice_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 过路过桥费发票识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int tollInvoice(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = tollInvoice_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/toll_invoice?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class TollInvoice
    {
        // 过路过桥费发票识别
        public static string tollInvoice()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/toll_invoice?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("过路过桥费发票识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

参数	是否必须	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ InvoiceCode	是	string	发票代码
+ InvoiceNum	是	string	发票号码
+ Entrance	是	string	入口
+ Exit	是	string	出口
+ Date	是	string	日期
+ Time	是	string	时间
+ Fare	是	string	金额
+ Province	是	string	省

|+ City | 是 |string | 市 |

返回示例

```JSON
    "words_result": {
        "Entrance": "保定北",
        "Time": "13:24",
        "InvoiceCode": "237001473201",
        "Date": "2020-02-18",
        "Exit": "邯郸",
        "Fare": "110.00",
        "InvoiceNum": "95876669"
        "City": "保定",
        "Province": "河北"
    },
    "log_id": "1274932270494384128",
    "words_result_num": 7
}
```

船票识别

接口描述
对全国范围内不同版式的客运船票、货运船票进行结构化识别，包括发票代码、发票号码、发票日期、发票类型、总金额、出发地点、到达地点等7个字段。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=财务票据OCR&api=rest/2.0/ocr/v1/ferry_ticket&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/ferry_ticket`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/ferry_ticket?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
船票识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/ferry_ticket"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 船票识别
```
~~~

```
 船票识别
*/
public class TaxiReceipt {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String ferryTicket() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/ferry_ticket";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        FerryTicket.ferryTicket();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/ferry_ticket";
static std::string ferryTicket_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    ferryTicket_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
```

```
* 船票识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int ferryTicket(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result =ferryTicket_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
```

```
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/ferry_ticket?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class FerryTicket
    {
        // 船票识别
        public static string ferryTicket()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/ferry_ticket?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("船票识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
```

```
    }
}

```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ InvoiceType	是	string	发票类型
+ InvoiceCode	是	string	发票代码
+ InvoiceNum	是	string	发票号码
+ StartingStation	是	string	出发地点
+ DestinationStation	是	string	到达地点
+ Fare	是	string	总金额
+ InvoiceDate	是	string	开票日期

返回示例

```
{
"log_id":2034039811,
"words_result_num":7,
"words_result":
{
 "InvoiceType":"客运船票",
 "InvoiceCode":"111001681009",
 "InvoiceNum":"90769610",
 "StartingStation":"烟台",
 "DestinationStation": "大连",
 "InvoiceDate": "2020-10-26",
      "Fare":"¥153.30元"
}
}
```

网约车行程单识别

接口描述

对各大主要服务商的网约车行程单进行结构化识别，包括滴滴打车、花小猪打车、高德地图、曹操出行、阳光出行，支持识别服务商、行程开始时间、行程结束时间、车型、总金额等16 个关键字段。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/online_taxi_itinerary`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/online_taxi_itinerary?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ ServiceProvider	是	string	服务商
+ StartTime	是	string	行程开始时间
+ EndTime	是	string	行程结束时间
+ Phone	是	string	行程人手机号
+ ApplicationDate	是	string	申请日期
+ TotalFare	是	string	总金额
+ ItemNum	是	array	行程信息中包含的行程数量
+ Items	是	array	行程信息
++ ItemId	是	string	行程信息的对应序号
++ PickupTime	是	string	上车时间
++ PickupDate	是	string	上车日期
++ CarType	是	string	车型
++ Distance	是	string	里程
++ StartPlace	是	string	起点
++ DestinationPlace	是	string	终点
++ City	是	string	城市
++ Fare	是	string	金额
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 1385196013945356288,
  "words_result_num": 7
```

```
    "words_result": {
        "TotalFare": "2316",
        "EndTime": "2020-07-30 19:00",
        "Phone": "13000000000",
        "ServiceProvider": "滴滴企业版",
        "StartTime": "2020-07-01 16:00",
        "ApplicationDate": "2017-12-08",
        "ItemId": "3"
        "items": [
            {
                "ItemId": "1",
                "StartPlace": "鱼化寨地铁-D口",
                "PickupTime": "16:00",
                "CarType": "快车",
                "City": "西安市",
                "Distance": "9.7",
                "PickupDate": "20-07-01",
                "DestinationPlace": "创新港",
                "Fare": "20.86"
            },
            {
                "ItemId": "2",
                "StartPlace": "科学园东门",
                "PickupTime": "14:56",
                "CarType": "快车",
                "City": "西安市",
                "Distance": "91",
                "PickupDate": "20-07-02",
                "DestinationPlace": "鱼化寨地铁站",
                "Fare": "18.58"
            },
            {
                "ItemId": "3",
                "StartPlace": "中俄丝路创新园东门",
                "PickupTime": "19:00",
                "CarType": "快车",
                "City": "西安市",
                "Distance": "9.1",
                "PickupDate": "20-07-30",
                "DestinationPlace": "新门地铁站",
                "Fare": "20.38"
            },
        ],
    },
}
```


购物小票识别

接口描述

支持识别各类商场、超市及药店的购物小票，包括店名、小票号码、机器编号、工号、消费日期、消费时间、总金额、找零、币种、实收金额、优惠金额、打印日期、打印时间、明细商品名称、单价、数量、小计金额等信息。

在线调试
您可以在[示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=财务票据OCR&api=rest/2.0/ocr/v1/shopping_receipt&method=post)中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。
请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/shopping_receipt`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和 image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和 image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px</br>**优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	string	true/false	是否返回字段置信度，默认为 false ，即不返回
location	否	string	true/false	是否返回字段位置坐标，默认为 false，即不返回

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/shopping_receipt?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
购物小票识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/shopping_receipt"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 购物小票识别
*/
public class ShoppingReceipt{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String shoppingReceipt() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/shopping_receipt";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        ShoppingReceipt.shoppingReceipt();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/shopping_receipt";
static std::string shoppingReceipt_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    shoppingReceipt_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 购物小票识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int shoppingReceipt(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = shoppingReceipt_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/shopping_receipt?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class ShoppingReceipt
    {
        // 购物小票识别
        public static string shoppingReceipt()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/shopping_receipt?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("购物小票识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

```
 ###### 返回说明

**返回参数**


| 字段                | 是否必选 | 类型   | 说明                                  |
| ------------------- | -------- | ------ | ------------------------------------- |
| log_id              | 是       | uint64 | 唯一的log id，用于问题定位            |
| pdf_file_size             | 否       | string | 传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
|
| words_result        | 是       | object{}  | 识别结果                          |
|+ shop_name |是 |string |店名/超市名字|
|+ receipt_num |是 |string |小票号码|
|+ machine_num |是 |string |机器编号|
|+ employee_num |是 |string |工号|
|+ consumption_date |是 |string |消费日期|
|+ consumption_time |是 |string |消费时间|
```

+ total_amount	是	string	总金额
+ change	是	string	找零
+ currency	是	string	币种
+ paid_amount	是	string	实收金额
+ discount	是	string	优惠/折扣
+ print_date	是	string	打印日期
+ print_time	是	string	打印时间
+ table_row_num	是	uint32	商品明细行数，表示Table中的object个数
+ table	是	array[]	消费明细区域
++ product	是	string	商品条码/名称
++ quantity	是	string	数量
++ unit_price	是	string	单价
++ subtotal_amount	是	string	小计金额
+++ word	是	string	字段识别结果，以上各字段均包含此参数
+++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
++++ top	否	uint32	字段的上边距
++++ left	否	uint32	字段的左边距
++++ height	否	uint32	字段的高度
++++ width	否	uint32	字段的宽度
+++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
++++ average	否	float	字段识别结果中各字符的置信度平均值
++++ min	否	float	字段识别结果中各字符的置信度最小值

返回示例

```JSON
{
    "words_result": [
        {
            "total_amount": [
                {
                    "word": "5.50"
                }
            ],
            "discount": [
                {
                    "word": "0.8"
                }
            ],
            "currency": [
                {
                    "word": "RMB"
                }
            ],
            "receipt_num": [
                {
                    "word": "22202010230414"
                }
            ],
            "shop_name": [
                {
                    "word": "盛玛特购物中心"
                }
            ],
            "consumption_time": [
                {
                    "word": "16:18"
                }
            ],
            "consumption_date": [
                {
                    "word": "2020-10-23"
```

```
        }
    ],
    "table_row_num": 2,
    "table": [
        {
            "product": {
                "word": "仔姜2100871003105"
            },
            "quantity": {
                "word": "0.199"
            },
            "unit_price": {
                "word": "15.60"
            },
            "subtotal_amount": {
                "word": "3.10"
            }
        },
        {
            "product": {
                "word": "绿豆芽2100741001200"
            },
            "quantity": {
                "word": "0.469"
            },
            "unit_price": {
                "word": "2.56"
            },
            "subtotal_amount": {
                "word": "1.20"
            }
        },
        {
            "product": {
                "word": "魔芋2100776001206"
            },
            "quantity": {
                "word": "0.612"
            },
            "unit_price": {
                "word": "1.96"
            },
            "subtotal_amount": {
                "word": "1.20"
            }
        }
    ],
    "paid_amount": [
        {
            "word": "5.50"
        }
    ],
    "print_date": [
        {
            "word": "2022-01-01"
        }
    ],
    "machine_num": [
        {
            "word": "22"
        }
    ],
    "change": [
```

```
            "change": [
              {
                "word": ""
              }
            ],
            "employee_num": [
              {
                "word": "4004"
              }
            ],
            "print_time": [
              {
                "word": ""
              }
            ]
          }
      ],
      "log_id": 1468109066360637315
}
```


医疗票据文字识别

医疗发票识别

接口描述

支持识别全国各地门诊/住院发票的全字段信息，包括业务流水号、发票号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付等关键字段，及收费项目明细、各省直辖市的专有信息，其中北京/广东/河北/河南/江苏/山东/上海/天津/浙江等地区票据识别效果较佳。
</br>支持识别收费项目明细，并可根据不同省市地区返回对应的识别信息，**同时支持收费项目信息的医保三目录信息核验**，支持包括北京、上海、广州、深圳等21 个城市的医保三目录信息核验。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=医疗票据OCR&api=rest/2.0/ocr/v1/medical_invoice&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_invoice`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	string	true/false	是否返回字段的位置信息，**默认为 false，可缺省**</br>- **false：**不返回字段位置信息</br>- **true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	string	true/false	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>- **false：**不返回字段识别结果的置信度</br>- **true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）
medi_query	否	string	100000→北京</br>200000→上海</br>510000→广州</br>518000→深圳</br>519000→珠海</br>410000→长沙</br>400000→重庆</br>710000→西安</br>300000→天津</br>528000→佛山</br>523000→东莞</br>310000→杭州</br>450000→郑州</br>550000→贵阳</br>116000→大连</br>215000→苏州</br>210000→南京</br>430000→武汉</br>230000→合肥</br>610000→成都</br>50000→石家庄	医保三目录查询，可选值为对应城市代码，可缺省，当此参数的输入值不为对应城市编码时，即进行全量查询，默认进行全量查询，**如不需要进行医保三目录查询，无需添加此参数**

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_invoice?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
医疗发票识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_invoice"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())

```
```java label=JAVA
package com.baidu.ai.aip;
```
~~~

```
import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 医疗发票识别
*/
public class Medicalinvoice {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalinvoice() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_invoice";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        Medicalinvoice.medicalinvoice();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_invoice";
static std::string medicalinvoice_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
```

```
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalinvoice_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 医疗发票识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalinvoice(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalinvoice_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```

```php label=PHP

<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
```

```
    $curl_post = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_invoice?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class Medicalinvoice
    {
        // 医疗发票识别
        public static string medicalinvoice()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_invoice?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("医疗发票识别:");
            Console.WriteLine(result);
            return result;
        }
```

```
        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
InvoiceType	是	string	票据种类
Province	是	string	省市：支持返回以下省市 北京/广东/河北/河南/江苏/山东/上海/天津/浙江等
words_result	是	object	识别结果
+ BusinessNum	是	object	业务流水号
+ InvoiceNum	是	object	发票号码
+ HospitalNum	是	object	住院号
+ HospitalName	是	object	医院名称
+ RecordNum	是	object	病例号
+ HospitalDay	是	object	住院天数
+ AdmissionDate	是	object	入院时间
+ DischargeDate	是	object	出院时间
+ Name	是	object	姓名
+ Sex	是	object	性别
+ HospitalType	是	object	医疗机构类型
+ SocialSecurityNum	是	object	社保卡号
+ InsuranceType	是	object	医保类型
+ ChargingUnit	是	object	收款单位
+ Payee	是	object	收款人
+ Date	是	object	开票日期
+ AmountInWords	是	object	大写合计金额
+ AmountInFiguers	是	object	小写合计金额
+ InsurancePayment	是	object	医保统筹支付
+ PersonalPayment	是	object	个人账户支付
+ PrepayAmount	是	object	预缴金额

+ PaymentAmount	是	object	补缴金额
+ RefundAmount	是	object	退费金额
+ ClinicNum	是	object	门诊号
++ word	是	string	字段识别结果，以上各字段均包含此参数
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+ CostCategories	是	array[]	项目大类：治疗费、检查费等项目大类
+ CostDetail	是	array[]	明细类别：药物/检查的明细类别
+ RegionSupplement	是	array[]	地区字段：根据省市返回改地区特有的字段

CostCategories字段包含多个array，每个数组包含多个object，见以下参数

字段	说明
++ name	字段名，包括：收费项目、金额
++ word	name字段对应的识别结果
++ medi_info	医保目录查询结果，**当 medi_query 参数存在时，仅在「收费项目」的数组里返回，**以下参数同此说明
+++ medi_check	查询是否成功，“1”表示成功，“0”表示不成功
+++ medi_name	药品名
+++ medi_type	医保类别
+++ medi_region	医保目录的城市
+++ medi_code	药品编码
+++ medi_register	药品注册号

CostDetail字段包含多个array，每个数组包含多个object，见以下参数

字段	说明
++ name	字段名，包括：编码、项目、规格、数量、单价、金额
++ word	name字段对应的识别结果
++ medi_info	**医保目录查询结果，当 medi_query 参数存在时，仅在「项目」的数组里返回，**以下参数同此说明
+++ medi_check	查询是否成功，“1”表示成功，“0”表示不成功
+++ medi_name	药品名
+++ medi_type	医保类别
+++ medi_region	医保目录的城市
+++ medi_code	药品编码
+++ medi_register	药品注册号

RegionSupplement字段包含多个object，不同省市返回字段不同，以实际票面上的信息为准，以下列表中字段仅为各省直辖

市可能包含的字段列举参考（不包含所有会返回的字段）

省市	返回参数（name）
军/警	票控码、公务员补助、机打票号、应收金额、商保赔付、合计、个人自理、比例自付、医师、票控盘号
北京	年度门诊大额累计支付、超封顶金额、门诊大额支付、交易流水号、退休补充支付、自付一、本次支付后个人账户余额、年度累计医保范围内金额、本次医保范围内金额、残军补助支付、实时结算、累计医保内范围金额、其他医保支付、单位补充险[原公疗]支付、年度居民基本医疗保险基金门诊累计支付
上海	分类自负、历年余额、本年余额、现金支付、自负、自费、附加支付、银联卡、可报金额、找零
西藏	统筹支付、现金支付、总费用
辽宁	其他医保支付、公务员补助、个人支付金额、实收金额、病种编码、挂号时间、可报销金额、大病支出、挂号级别、退款银行、负担较重、保外自费
山西	病人ID、现金、挂号科室、应收、优惠、医生、实收、个人支付金额、挂号时间、优惠、货币误差、实收扣卡金额
内蒙古	个人支付金额、就诊卡号、科室、现金、医疗机构、病区、申请人、经手人、床号、统筹段自付
安徽	银联卡、现金、卡号、退支票、就诊卡号、微信支付、科别、其他医保支付、账户支付、个人支付金额、卡支付
青海	其他医保支付、个人支付金额、现金、本页小计、操作员、就诊条码、费别、医院支付、就诊序号
湖南	优惠金额、现金、缴费日期、住院次数、大病、科室、打印时间、挂号日期、票控盘号、个人支付金额
宁夏	病人ID、医院名称、序号、现金支付、账户余额、账户支付、共付段自付、收费员、交费金额、统筹基金支付小计、特诊特治自付
重庆	科室、微信支付、结算情况、病历号、民政补助、住院预交金、疾病应急补助、医院补偿、冲预交、公务员、医保基金
吉林	入院时间、出院时间、自理费用、生育保险支付、住院天数、制单、医疗保险号、住院病志号、个人现金支付、保险对象补助支付、自付比例、各项补助支出
河北	就诊地点、民政、起付标准、统筹支付、现金支付、年龄、医院垫支、个人现金支付、自付、患者ID、本次起付额、个账支付、医生、医疗救助金额、基本统筹支付、救助支付金额
甘肃	住院科室、个人支付金额、发票号码、民政大病、公务员类型、医疗待遇类别、进入统筹、丙类自费、异地自负、开单医生、医师、刷卡后余额、公补助累计
陕西	其他医保支付、个人支付金额、统筹挂账、单据号、现、支、病区、余额
广西	病人ID、个人支付金额、就诊卡号、大额医疗支付、本年大额支付、残疾军人补助、补收现金、补交金额、转账、特检、单病种定额金额、医疗类别、住院附加累计、二次报销统筹支出
云南	个人账户支出、退、现金、预交款、统筹、民政补助、其中账户、个人自付总额、科室、现金支付、总额
广东	银联卡、基本医疗、病人ID、现金、补、医保卡、银行卡、医保统筹、医生、流水号、结算方式、科室、打印时间、预收诊金、发票调整费、病人卡号、欠款、个人支付金额
黑龙江	预交金额、科别、其他支付、医疗补助、补交金额、预交金、医保统筹、合计金额(大写)、医保类别、个人支付、现金支付、金额合计(大写)、账户支付、大病保险
浙江	印刷号、自费、实收现金、其中现金、结算流水号、预缴款、统筹基金、历年余额、票据号码、医保支付、优惠金额、自负、支票、基金支付、惠民
湖北	按比例自付、统筹支付2、置换材料自付、个人自费、统筹支付1、起付线、医疗总费用、公务员补助1、商保赔付2、找零、费用总额、补充医疗保险、基本费用、社区三免、财政兜底、科室、结算日期、应收费用、大额费用、个人账户余额、就诊记录号、慈善救助
江西	住院费用、伤残基金、本次报销、卡余额、其他医保支付、单位补充、挂账、企业补充、门慢起伏累计、重大疾病内补、重大疾病外补、政府兜底保障、建档大病、同级财政补助、扶贫基金累计
福建	结算时间、民政救助、公务员补助、现金转账、数字指纹、结算前余额、结算补交、医院减免金额、健康通、付款方式、高值类费用、商险支付、B卡支付、退还
新疆	其他医保支付、基本医疗、个人支付金额

新疆	其他医保支付、基本医疗、个人支付金额
贵州	自费、统筹、账户、大病保险、起付线、大病、住院科室、微信支付、支付方式、个人支付金额
江苏	本年住院次数、现金自付、医保政策范围内个人自付、个帐支付、共济支付、本次公补账户计入、慢病门诊医疗费累计、医疗救助、工会互助、报销金额、校验码、起付标准、医疗救助支付、本年医保累计、现金支付、其他医保支付
四川	床号、校验码、科室、现金支付、个人自付、医保编码、个人现金支付、现金、个人支付
山东	就诊日期、当年账户余额、统筹支付、微信支付、挂号日期、现金支付、其他医保支付、本次账户支付、本次总费用、医疗补助、打印时间、个人负担、住院科室、自费金额
河南	大病补充保险、大额支付、账户余额、公务员记账、个人支付金额、大病补充、挂账、欠费、超大额、大额
天津	挂号、个人支付金额、总计金额、印刷号、结算时间、大病保险支付、医保个人账户、科室、本次起始费金额、总费用

返回示例


```
{
  "log_id": 1397076899313745920,
  "words_result_num": 28,
  "Province": "北京",
  "InvoiceType": "门诊发票"
  "words_result": {
     "AmountInWords": {
        "word": "玖佰叁拾玖元肆角捌分"
     },
     "Sex": {
        "word": "男"
     },
     "InsuranceType": {
        "word": "城镇职工"
     },
     "Name": {
        "word": "王成"
     },
     "SocialSecurityNum": {
        "word": "T03419700077"
     },
     "DischargeDate": {
        "word": "2016-01-01"
     },
     "HospitalNum": {
        "word": "0937829032"
     },
     "HospitalName": {
        "word": "北京市房山区良乡医院"
     },
     "CostCategories": [
        [
           {
              "name": "收费项目",
              "word": "西药费"
           },
           {
              "name": "金额",
              "word": "426.70"
           }
        ],
        [
           {
              "name": "收费项目",
              "word": "中成药费"
```

```
            "word": "中成药费"
        },
        {
            "name": "金额",
            "word": "512.78"
        }
    ]
],
"RegionSupplement": [
    {
        "name": "其他医保支付",
        "word": "0.00"
    },
    {
        "name": "年度门诊大额累计支付",
        "word": "83.79"
    },
    {
        "name": "起付金额",
        "word": "777.11"
    },
    {
        "name": "基金支付",
        "word": "101.75"
    },
    {
        "name": "本次支付后个人账户余额",
        "word": "0.00"
    },
    {
        "name": "个人支付金额",
        "word": "837.73"
    },
    {
        "name": "交易流水号",
        "word": "111100030Z160517006328"
    },
    {
        "name": "自付一",
        "word": "795.06"
    },
    {
        "name": "自付二",
        "word": "42.67"
    },
    {
        "name": "累计医保内范围金额",
        "word": "1419.70"
    },
    {
        "name": "门诊大额支付",
        "word": "83.79"
    },
    {
        "name": "本次医保范围内金额",
        "word": "896.81"
    },
    {
        "name": "退休补充支付",
        "word": "17.96"
    },
    {
        "name": "超封顶金额",
```

```
            "word": "0.00"
        },
        {
            "name": "残军补助支付",
            "word": "0.00"
        },
        {
            "name": "单位补充险[原公疗]支付",
            "word": "0.00"
        },
        {
            "name": "自费",
            "word": "42.67"
        }
    ],
    "ClinicNum": {
        "word": "12169298"
    },
    "AmountInFiguers": {
        "word": "939.48"
    },
    "AdmissionDate": {
        "word": "2016-01-01"
    },
    "HospitalType": {
        "word": "综合医院"
    },
    "RefundAmount": {
        "word": "0.00"
    },
    "Date": {
        "word": "2016年05月17日"
    },
    "ChargingUnit": {
        "word": "房山区良乡医院"
    },
    "CostDetail": [
        [
            {
                "name": "编码",
                "word": "01"
            },
            {
                "name": "项目",
                "word": "阿托伐他汀钙胶囊"
            },
            {
                "name": "规格",
                "word": "10mg*7支"
            },
            {
                "name": "数量",
                "word": "10"
            },
            {
                "name": "单价",
                "word": "29.3200"
            },
            {
                "name": "金额",
                "word": "293.20"
            }
```

```
            ],
            [
                {
                    "name": "编码",
                    "word": "02"
                },
                {
                    "name": "项目",
                    "word": "替米沙坦胶囊"
                },
                {
                    "name": "规格",
                    "word": "40mg*12粒"
                },
                {
                    "name": "数量",
                    "word": "5"
                },
                {
                    "name": "单价",
                    "word": "26.7000"
                },
                {
                    "name": "金额",
                    "word": "133.50"
                }
            ],
        ],
        "PaymentAmount": {
            "word": "0.00"
        },
        "PrepayAmount": {
            "word": "0.00"
        },
        "PersonalPayment": {
            "word": "0.00"
        },
        "HospitalDay": {
            "word": "15"
        },
        "BusinessNum": {
            "word": "40091198916051710196"
        },
        "InsurancePayment": {
            "word": "0.00"
        },
        "Payee": {
            "word": "刘冰"
        },
        "RecordNum": {
            "word": "0001268129"
        },
        "InvoiceNum": {
            "word": "0103152099"
        }
    },
}
```


医疗费用明细识别

接口描述

接口描述

支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含项目类型、项目名称、单价、数量、规格、金额，其中北京地区识别效果最佳。

在线调试

您可以在 |示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=医疗票据OCR&api=rest/2.0/ocr/v1/medical_detail&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_detail`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省**</br>**- false：**不返回字段位置信息</br>**- true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>**- false：**不返回字段识别结果的置信度</br>**- true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_detail?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
医疗费用明细识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_detail"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 医疗费用明细识别
*/
public class MedicalDetail {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalDetail() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_detail";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalDetail.medicalDetail();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_detail";
static std::string medicalDetail_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalDetail_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 医疗费用明细识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalDetail(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalDetail_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_detail?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalDetail
    {
        // 医疗费用明细识别
        public static string medicalDetail()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_detail?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("医疗费用明细识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~


###### 返回说明

**返回参数**

| 字段                | 是否必输出 | 类型   | 说明 |
| ------------------- | -------- | ------ | ------------------------------ |
| log_id              | 是       | uint64 | 调用日志id，用于问题定位 |
| words_result        | 是       | object | 识别结果 |
| words_result_num    | 是       | uint32 | 识别结果数，表示words_result的元素个数 |
| + Name              | 是       | object | 姓名 |
| + Date              | 是       | object | 日期 |
| + PatientID         | 是       | object | 病人ID |
| + TotalAmount       | 是       | object | 总金额 |
| + word              | 是       | string | 字段识别结果，以上各字段均包含此参数 |
| ++  location | 否 | object | 字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数 |
| +++  top | 否 | uint32 | 字段的上边距 |
| +++  left | 否 | uint32 | 字段的左边距 |
| +++  height | 否 | uint32 | 字段的高度 |
| +++  width | 否 | uint32 | 字段的宽度 |
| ++ probability | 否 | object | 字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数 |
| +++ average | 否 | float | 字段识别结果中各字符的置信度平均值 |
| +++ min | 否 | float | 字段识别结果中各字符的置信度最小值 |
| + CostDetail | 是 | array[] | 项目明细 |

**CostDetail字段包含多个Array，每个数组包含多个object，见以下参数**

| 字段 | 说明 |
| --- | --- |
| ++ word_name | 字段名，包括：项目类型、项目名称、单价、数量、规格、金额 |
| ++ word | word_name字段对应的识别结果 |

**返回示例**
```

```JSON

{
    "log_id": 1397090241579319296,
    "words_result_num": 5,
    "words_result": {
      "PatientID": {
          "word": "23683829"
      },
      "TotalAmount": {
          "word": "600.00"
      },
      "Date": {
          "word": "2020年11月04日"
      },
      "Name": {
          "word": "范浩"
      },
      "CostDetail": [
          [
              {
                  "word_name": "清单项目名称",
                  "word": "普通过敏原(新)筛查"
              },
              {
                  "word_name": "单价",
                  "word": "580.00"
              },
              {
                  "word_name": "数量",
                  "word": "1.00"
              },
              {
                  "word_name": "金额",
                  "word": "580.00"
              },
              {
                  "word_name": "规格",
                  "word": "次"
              },
              {
                  "word_name": "项目类型",
                  "word": "化验费"
              }
          ],
          [
              {
                  "word_name": "清单项目名称",
                  "word": "总IgE测定"
              },
              {
                  "word_name": "单价",
                  "word": "20.00"
              },
              {
                  "word_name": "数量",
                  "word": "1.00"
              },
              {
                  "word_name": "金额",
                  "word": "20.00"
              },
              {
```

```
            {
               "word_name": "规格",
               "word": "次"
            },
            {
               "word_name": "项目类型",
               "word": "化验费"
            }
         ]
      ],
   },
}
```

医疗费用结算单识别

接口描述

支持识别全国医疗费用结算单的姓名、出/入院时间、发票总金额、自费金额、医保支付金额等 6 个关键字段，其中北京地区票据识别效果最佳。

在线调试

您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=医疗票据OCR&api=rest/2.0/ocr/v1/medical_statement&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL：`https://aip.baidubce.com/rest/2.0/ocr/v1/medical_statement`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	string	true/false	是否返回字段的位置信息，**默认为 false，可缺省**</br>- **false：**不返回字段位置信息</br>- **true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	string	true/false	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>- **false：**

不返回字段识别结果的置信度</br>- **true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min） |

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_statement?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
##### encoding:utf-8

import requests
import base64

'''
医疗费用结算单识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_statement"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 医疗费用结算单识别
*/
public class MedicalStatement {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalStatement() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_statement";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalStatement.medicalStatement();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_statement";
static std::string medicalStatement_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalStatement_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 医疗费用结算单识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalStatement(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalStatement_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_statement?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalStatement
    {
        // 医疗费用结算单识别
        public static string medicalStatement()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_statement?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("医疗费用结算单识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~


###### 返回说明

**返回参数**

| 字段                  | 是否必输出 | 类型   | 说明 |
| --------------------- | -------- | ------ | ------------------------------ |
| log_id                | 是       | uint64 | 调用日志id，用于问题定位 |
| words_result_num      | 是       | uint32 | 识别结果数，表示words_result的元素个数 |
| InvoiceType           | 是       | string | 票据种类 |
| words_result          | 是       | object  | 识别结果 |
| + AdmissionDate | 是 | object | 入院时间 |
| + DischargeDate | 是 | object | 出院时间 |
| + Name | 是 | object | 姓名 |
|+ AmountInFiguers|是| object|发票总金额|
|+ SelfPaymentAmount|是|object|全自费|
|+ MedicalInsuranceAmount|是|object|医保支付|
 ++ word | 是 | string | 字段识别结果，以上各字段均包含此参数 |
| ++ location | 否 | object | 字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数 |
| +++ top | 否 | uint32 | 字段的上边距 |
| +++ left | 否 | uint32 | 字段的左边距 |
| +++ height | 否 | uint32 | 字段的高度 |
| +++ width | 否 | uint32 | 字段的宽度 |
| ++ probability | 否 | object | 字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数 |
| +++ average | 否 | float | 字段识别结果中各字符的置信度平均值 |
| +++ min | 否 | float | 字段识别结果中各字符的置信度最小值 |

**返回示例**

```JSON
{
    "log_id": 1397086143811420160,
    "words_result_num": 6,
    "InvoiceType": "普通结算单"
```
```

```
    "word_result": {
        "AmountInFiguers": {
            "word": "10066.84"
        },
        "MedicalInsuranceAmount": {
            "word": "9066.84"
        },
        "SelfPaymentAmount": {
            "word": "1000.00"
        },
        "AdmissionDate": {
            "word": "2017-06-02"
        },
        "DischargeDate": {
            "word": "2017-06-13"
        },
        "Name": {
            "word": "王美花"
        }
    },
}
```


医疗检验报告单识别

接口描述

支持识别全国各地医疗检验报告单的姓名、性别、医院名称、报告单名称等关键字段，支持识别检查具体项目的项目名称、结果、单位、参考区间、结果提示等明细字段。

在线调试

您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=医疗票据OCR&api=rest/2.0/ocr/v1/medical_report_detection&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_report_detection`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明

| --- | --- | --- | --- | --- |
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省**</br>**- false：**不返回字段位置信息</br>**- true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>**- false：**不返回字段识别结果的置信度</br>**- true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_report_detection?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
##### encoding:utf-8

import requests
import base64

'''
医疗检验报告单识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_report_detection"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 医疗检验报告单识别
*/
public class MedicalReportDetection{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalReportDetection() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_report_detection";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalReportDetection.medicalReportDetection();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_report_detection";
static std::string medicalReportDetection_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalReportDetection_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 医疗检验报告单识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalReportDetection(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalReportDetection_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_report_detection?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalReportDetection
    {
        // 医疗检验报告单识别
        public static string medicalReportDetection()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_report_detection?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("医疗检验报告单识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
CommonData_result_num	是	uint32	患者具体信息的识别结果数，表示CommonData的元素个数
Item_row_num	是	uint32	检查项目的行数，表示Item中的数组个数
+ CommonData	是	array[]	患者具体信息
+ Item	是	array[]	检查项目
++ word_name	是	string	字段名，详见下方表格区说明
++ word	是	string	word_name字段对应的识别结果
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

CommonData字段包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：医院、报告单名称、姓名、性别、年龄、科室、标本种类、临床诊断、时间、临床症状、检查项目、标本情况、检查目的、建议、检查结果
++ word	word_name字段对应的识别结果

Item字段包含多个array，每个数组包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：项目名称、项目代号、结果、单位、参考区间、结果提示、测试方法、仪器类型
++ word	word_name字段对应的识别结果

返回示例

```JSON
{
    "Item_row_num": 5,
    "words_result": {
        "Item": [
            [
                {
                    "word_name": "仪器类型",
                    "word": ""
                },
                {
                    "word_name": "单位",
                    "word": "10^9/L"
                },
                {
                    "word_name": "参考区间",
                    "word": "3.97--9.15"
                },
                {
                    "word_name": "测试方法",
                    "word": ""
                },
                {
                    "word_name": "结果",
                    "word": "18.82"
                },
                {
                    "word_name": "结果提示",
                    "word": ""
                },
                {
                    "word_name": "项目代号",
                    "word": "WBC"
                },
                {
                    "word_name": "项目名称",
                    "word": "白细胞计数"
                }
            ],
            [
                {
                    "word_name": "仪器类型",
                    "word": ""
                },
                {
                    "word_name": "单位",
                    "word": "10^12/L"
                },
                {
                    "word_name": "参考区间",
                    "word": "3.68-574"
                },
                {
                    "word_name": "测试方法",
                    "word": ""
```

```
            "word":
        },
        {
            "word_name": "结果",
            "word": "5.13"
        },
        {
            "word_name": "结果提示",
            "word": ""
        },
        {
            "word_name": "项目代号",
            "word": "RBC"
        },
        {
            "word_name": "项目名称",
            "word": "红细胞计数"
        }
    ],
    [
        {
            "word_name": "仪器类型",
            "word": ""
        },
        {
            "word_name": "单位",
            "word": "g/L"
        },
        {
            "word_name": "参考区间",
            "word": "113--172"
        },
        {
            "word_name": "测试方法",
            "word": ""
        },
        {
            "word_name": "结果",
            "word": "132.00"
        },
        {
            "word_name": "结果提示",
            "word": ""
        },
        {
            "word_name": "项目代号",
            "word": "HGB"
        },
        {
            "word_name": "项目名称",
            "word": "血红蛋白浓度"
        }
    ],
    [
        {
            "word_name": "仪器类型",
            "word": ""
        },
        {
            "word_name": "单位",
            "word": "10^9/L"
        },
        {
```

```
            "word_name": "参考区间",
            "word": "101--320"
        },
        {
            "word_name": "测试方法",
            "word": ""
        },
        {
            "word_name": "结果",
            "word": "419.101"
        },
        {
            "word_name": "结果提示",
            "word": ""
        },
        {
            "word_name": "项目代号",
            "word": "PLT"
        },
        {
            "word_name": "项目名称",
            "word": "血小板计数"
        }
    ],
    [
        {
            "word_name": "仪器类型",
            "word": ""
        },
        {
            "word_name": "单位",
            "word": "%"
        },
        {
            "word_name": "参考区间",
            "word": "50-70"
        },
        {
            "word_name": "测试方法",
            "word": ""
        },
        {
            "word_name": "结果",
            "word": "68.54"
        },
        {
            "word_name": "结果提示",
            "word": ""
        },
        {
            "word_name": "项目代号",
            "word": "NEUT%"
        },
        {
            "word_name": "项目名称",
            "word": "中性粒细胞百分比"
        }
    ]
],
"CommonData": [
    {
        "word_name": "临床症状",
```

```
            "word": ""
        },
        {
            "word_name": "临床诊断",
            "word": ""
        },
        {
            "word_name": "医院",
            "word": ""
        },
        {
            "word_name": "姓名",
            "word": "刘安洋"
        },
        {
            "word_name": "年龄",
            "word": "3岁"
        },
        {
            "word_name": "建议",
            "word": ""
        },
        {
            "word_name": "性别",
            "word": "女"
        },
        {
            "word_name": "报告单名称",
            "word": "霍林郭勒市中蒙医医院检验报告单"
        },
        {
            "word_name": "时间",
            "word": "20201031"
        },
        {
            "word_name": "标本情况",
            "word": ""
        },
        {
            "word_name": "标本种类",
            "word": "血液"
        },
        {
            "word_name": "检查目的",
            "word": ""
        },
        {
            "word_name": "检查结果",
            "word": ""
        },
        {
            "word_name": "检查项目",
            "word": ""
        },
        {
            "word_name": "科室",
            "word": "儿科门诊"
        }
    ]
},
"CommonData_result_num": 15,
"log_id": 1455012594820522020
}
```

}

医疗诊断报告单识别

接口描述

支持识别全国各地各医院医疗诊断报告单，包括医院名称、报告名称、姓名、性别、年龄、科室、临床诊断、报告日期、检查部位、检查方法、检查所见、检查提示、建议、肉眼可见 14个字段。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=医疗票据OCR&api=rest/2.0/ocr/v1/health_report&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/health_report`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省**</br>**- false：**不返回字段位置信息</br>**- true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>**- false：**不返回字段识别结果的置信度</br>**- true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/health_report?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
 ##### encoding:utf-8

import requests
import base64

'''
医疗诊断报告单识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/health_report"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 医疗诊断报告单识别
*/
public class HealthReport{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String healthReport() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/health_report";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        HealthReport.healthReport();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/health_report";
static std::string healthReport_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    healthReport_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 医疗诊断报告单识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int healthReport(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = healthReport_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
 <?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/health_report?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class HealthReport
    {
        // 医疗诊断报告单识别
        public static string healthReport()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/health_report?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("医疗诊断报告单识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ word_name	是	string	字段名，详见下方表格区说明
+ word	是	string	word_name字段对应的识别结果
+ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
++ top	否	uint32	字段的上边距
++ left	否	uint32	字段的左边距
++ height	否	uint32	字段的高度
++ width	否	uint32	字段的宽度
+ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
++ average	否	float	字段识别结果中各字符的置信度平均值
++ min	否	float	字段识别结果中各字符的置信度最小值

words_result字段包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：医院名称、报告名称、姓名、性别、年龄、科室、临床诊断、报告日期、检查部位、检查方法、检查所见、检查提示、建议、肉眼可见
++ word	word_name字段对应的识别结果

返回示例

```JSON
```

```
{
    "words_result_num": 14,
    "words_result": [
        {
            "word": "陆军军医大学第一附属医院(西南医院)",
            "word_name": "医院名称"
        },
        {
            "word": "放射科CT诊断报告单",
            "word_name": "报告名称"
        },
        {
            "word": "杜凤一",
            "word_name": "姓名"
        },
        {
            "word": "女",
            "word_name": "性别"
        },
        {
            "word": "60岁",
            "word_name": "年龄"
        },
        {
            "word": "骨科新冠筛查门诊",
            "word_name": "科室"
        },
        {
            "word": "待查",
            "word_name": "临床诊断"
        },
        {
            "word": "CT胸椎三维重建检查",
            "word_name": "检查方法"
        },
        {
            "word": "胸椎椎序列正常,椎体缘可见骨质增生变尖,胸9椎体致密结节影,椎间隙无狭窄,椎管内未见异常密度,椎旁软组织未见异常。",
            "word_name": "检查所见"
        },
        {
            "word": "1.胸椎轻度退性行改变。2.胸9椎体骨岛可能。",
            "word_name": "检查提示"
        },
        {
            "word": "20201019",
            "word_name": "报告日期"
        },
        {
            "word": "",
            "word_name": "肉眼可见"
        },
        {
            "word": "",
            "word_name": "建议"
        },
        {
            "word": "",
            "word_name": "检查部位"
        }
    ],
    "log_id": 1547051630940329127
```

    log_id：1547031630940329127
}

病案首页识别

接口描述

支持识别全国各地病案首页的病案号、姓名、性别、出生日期、身份证号、出/入院科别、住院次数、药物过敏情况等15 个关键字段，其中北京地区票据识别效果最佳

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=财务票据OCR&api=rest/2.0/ocr/v1/shopping_receipt&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_record
`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	string	true/false	是否返回字段的位置信息，**默认为 false，可缺省**</br>- **false：**不返回字段位置信息</br>- **true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	string	true/false	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>- **false：**不返回字段识别结果的置信度</br>- **true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_record?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
 ##### encoding:utf-8

import requests
import base64

'''
病案首页识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_record"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 病案首页识别
*/
public class MedicalRecord {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalRecord() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_record";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalRecord.medicalRecord();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_record";
static std::string medicalRecord_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalRecord_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 病案首页识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalRecord(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalRecord_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_record?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalRecord
    {
        // 病案首页识别
        public static string medicalRecord()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_record?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("病案首页识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~


###### 返回说明

**返回参数**

| 字段                | 是否必输出 | 类型   | 说明 |
| ------------------- | -------- | ------ | ------------------------------ |
| log_id              | 是       | uint64 | 调用日志id，用于问题定位 |
| words_result_num    | 是       | uint32 | 识别结果数，表示words_result的元素个数 |
| InvoiceType         | 是       | string | 票据种类 |
| words_result        | 是       | object  | 识别结果 |
| + RecordNum | 是 | object | 病案号 |
| + Name | 是 | object | 姓名 |
| + Sex | 是 | object | 性别 |
|+ Birthday|是| object|出生日期|
|+ Age|是|object|年龄|
|+ Career|是|object|职业|
|+ MaritalStatus|是|object|婚姻|
|+ Nation|是|object|民族|
|+ ID|是|object|身份证号|
|+ Nationality|是|object|国籍|
|+ AdmissionDepartment|是|object|入院科别|
|+ DischargeDepartment|是|object|出院科别|
|+ HospitalDay|是|object|住院次数|
|+ Allergy|是|object|药物过敏|
|+ BloodType|是|object|血型、Rh血型、ABO血型|
| ++ word | 是 | string | 字段识别结果，以上各字段均包含此参数 |
| ++ location | 否 | object | 字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数 |
| +++ top | 否 |  uint32 | 字段的上边距 |
| +++ left | 否 | uint32 | 字段的左边距 |
| +++ height | 否 | uint32 | 字段的高度 |
| +++ width | 否 | uint32 | 字段的宽度 |
| ++ probability | 否 | object | 字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数 |
| +++ average | 否 | float | 字段识别结果中各字符的置信度平均值 |
```

| +++ min | 否 | float | 字段识别结果中各字符的置信度最小值 |

返回示例

```JSON
{
    "log_id": 1397084278038200320,
    "words_result_num": 15,
    "InvoiceType": "病案首页"
    "words_result": {
        "Nation": {
            "word": "汉族"
        },
        "Allergy": {
            "word": "2"
        },
        "Sex": {
            "word": "2"
        },
        "Birthday": {
            "word": "1968年01月10日"
        },
        "Nationality": {
            "word": "中国"
        },
        "Name": {
            "word": "毛丽"
        },
        "MaritalStatus": {
            "word": "已婚"
        },
        "HospitalDay": {
            "word": "第1次住院"
        },
        "AdmissionDepartment": {
            "word": "乳腺中心病房"
        },
        "ID": {
            "word": "1101"
        },
        "DischargeDepartment": {
            "word": "外科"
        },
        "Career": {
            "word": "其他"
        },
        "Age": {
            "word": "48岁"
        },
        "BloodType": {
            "word": "A型"
        },
        "RecordNum": {
            "word": "796968"
        }
    },
}
```

出院小结识别

接口描述

支持识别全国出院小结的科室、姓名、性别、年龄、入院日期、出院日期、住院天数、入院诊断、出院诊断、出院医嘱等关键字段。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=医疗票据OCR&api=rest/2.0/ocr/v1/medical_summary&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。
请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省**</br>**- false：**不返回字段位置信息</br>**- true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>**- false：**不返回字段识别结果的置信度</br>**- true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
出院小结识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 出院小结识别
*/
public class MedicalSummary {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalSummary() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalSummary.medicalSummary();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary";
static std::string medicalSummary_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalSummary_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 出院小结识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalSummary(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalSummary_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalSummary
    {
        // 出院小结识别
        public static string medicalSummary()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("出院小结识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ word_name	是	string	字段名，包括：科室、姓名、性别、年龄、入院日期、出院日期、住院天数、入院诊断、出院诊断、出院医嘱
+ word	是	string	字段识别结果，以上各字段均包含此参数
+ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
++ top	否	uint32	字段的上边距
++ left	否	uint32	字段的左边距
++ height	否	uint32	字段的高度
++ width	否	uint32	字段的宽度
+ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
++ average	否	float	字段识别结果中各字符的置信度平均值
++ min	否	float	字段识别结果中各字符的置信度最小值

返回示例

```JSON
{
    "words_result": [
        {
            "word_name": "住院天数",
            "word": "10"
        },
        {
```

```
            "word_name": "入院日期",
            "word": "2020-12-16"
        },
        {
            "word_name": "入院诊断",
            "word": "肠胃炎"
        },
        {
            "word_name": "出院医嘱",
            "word": ""
        },
        {
            "word_name": "出院日期",
            "word": "2020-12-26"
        },
        {
            "word_name": "出院诊断",
            "word": "已康复"
        },
        {
            "word_name": "姓名",
            "word": "曹根华"
        },
        {
            "word_name": "年龄",
            "word": "46"
        },
        {
            "word_name": "性别",
            "word": "男"
        },
        {
            "word_name": "科室",
            "word": "肠胃一科"
        }
    ],
    "words_result_num": 10,
    "log_id": 1468419876418830448
}
```


入院小结识别

接口描述

支持识别全国各地各医院入院小结的姓名、性别、年龄、入院时间、主诉、身份证号、联系人电话、病史采集日期、既往史、现病史、个人史、月经婚育史、工作单位、可靠程度 14个关键字段。

申请试用

该接口正在邀测中，在正式使用之前，请先提交[合作咨询](https://ai.baidu.com/consultation/cooperation?referrerUrl=/tech/ocr_medical)，或者[提交工单](https://ticket.bce.baidu.com/#/ticket/create~productId=96)，提供公司名称、appid、应用场景，工作人员协助开通权限后方可使用。

请求说明

请求示例

HTTP 方法：`POST`

HTTP 方法：POST

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_in_hospital`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省**</br>**- false：**不返回字段位置信息</br>**- true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>**- false：**不返回字段识别结果的置信度</br>**- true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_in_hospital?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
入院小结识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_in_hospital"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 入院小结识别
*/
public class MedicalSummaryInHospital{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalSummaryInHospital() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_in_hospital";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalSummaryInHospital.medicalSummaryInHospital();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_in_hospital";
static std::string medicalSummaryInHospital_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalSummaryInHospital_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 入院小结识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalSummaryInHospital(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalSummaryInHospital_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_in_hospital?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalSummaryInHospital
    {
        // 入院小结识别
        public static string medicalSummaryInHospital()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_in_hospital?access_token=" +
token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
```

```
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("入院小结识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

~~~

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ word_name	是	string	字段名，详见下方表格区说明
+ word	是	string	word_name字段对应的识别结果
+ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
++ top	否	uint32	字段的上边距
++ left	否	uint32	字段的左边距
++ height	否	uint32	字段的高度
++ width	否	uint32	字段的宽度
+ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
++ average	否	float	字段识别结果中各字符的置信度平均值
++ min	否	float	字段识别结果中各字符的置信度最小值

words_result字段包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：姓名、性别、年龄、入院时间、主诉、身份证号、联系人电话、病史采集日期、既往史、现病史、个人史、月经婚育史、工作单位、可靠程度
++ word	word_name字段对应的识别结果

返回示例
~~~

```JSON
{
    "words_result_num": 14,
    "words_result": [
        {
            "word": "生于原籍,无外地久居史。否认疫区、疫水接触史,否认特殊化学品及放射性物质接触史。无吸烟饮酒等不良嗜好。",
            "word_name": "个人史"
        },
        {
            "word": "发现双侧甲状腺多发结节2年半",
            "word_name": "主诉"
        },
        {
            "word": "2019年04月29日 14:17:00",
            "word_name": "入院时间"
        },
        {
            "word": "可靠",
            "word_name": "可靠程度"
        },
        {
            "word": "王月",
            "word_name": "姓名"
        },
        {
            "word": "",
            "word_name": "工作单位"
        },
        {
            "word": "31岁",
            "word_name": "年龄"
        },
        {
            "word": "女",
            "word_name": "性别"
        },
        {
            "word": "平素身体健康状况一般,近期出现血压升高并诊断为高血压,最高血压196/120mHg,目前长期口服傲坦20mgqd,倍他乐克47.5mgqd,目前血压可控制在125/75mHg。",
            "word_name": "既往史"
        },
        {
            "word": "初潮13岁,行经天数5-7天,月经周期35-40天,末次月经2018-05-08。",
            "word_name": "月经婚育史"
        },
        {
            "word": "患者2年半前因体检发现双侧甲状腺多发结节(具体报告未见),未见临床症状,否认头痛、心悸、多汗、声音哑、情绪改变等其他不适,就诊于外院,建议患者行穿刺活检术以进一步明确诊断,患者未予重视。患者自起病以来,精神、饮食、睡眠可,大小便如常,体重未见明显变化。",
            "word_name": "现病史"
        },
        {
            "word": "",
            "word_name": "病史采集日期"
        },
        {
            "word": "",
            "word_name": "联系人电话"
        },
        {
            "word": "",
            "word_name": "身份证号"
```

```
            word_name：身份证号
        }
    ],
    "log_id": 1556908548410403256
}
```

诊断证明识别

接口描述

支持识别全国各地各医院诊断证明的姓名、性别、年龄、科室、入院时间、出院时间、住院号、出院诊断、出院医嘱 9 个关键字段。

申请试用

该接口正在邀测中，在正式使用之前，请先提交[合作咨询](https://ai.baidu.com/consultation/cooperation?referrerUrl=/tech/ocr_medical)，或者[提交工单](https://ticket.bce.baidu.com/#/ticket/create~productId=96)，提供公司名称、appid、应用场景，工作人员协助开通权限后方可使用。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_diagnosis`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省**</br>**- false：**不返回字段位置信息</br>**- true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>**- false：**不返回字段识别结果的置信度</br>**- true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_diagnosis?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_diagnosis?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
 ##### encoding:utf-8

import requests
import base64

'''
诊断证明识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_diagnosis"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 诊断证明识别
*/
public class MedicalSummaryDiagnosis{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalSummaryDiagnosis() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_diagnosis";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalSummaryDiagnosis.medicalSummaryDiagnosis();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_diagnosis";
static std::string medicalSummaryDiagnosis_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalSummaryDiagnosis_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 诊断证明识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalSummaryDiagnosis(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalSummaryDiagnosis_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_diagnosis?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalSummaryDiagnosis
    {
        // 诊断证明识别
        public static string medicalSummaryDiagnosis()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_summary_diagnosis?access_token=" +
token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
```

```
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("诊断证明识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ word_name	是	string	字段名，详见下方表格区说明
+ word	是	string	word_name字段对应的识别结果
+ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
++ top	否	uint32	字段的上边距
++ left	否	uint32	字段的左边距
++ height	否	uint32	字段的高度
++ width	否	uint32	字段的宽度
+ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
++ average	否	float	字段识别结果中各字符的置信度平均值
++ min	否	float	字段识别结果中各字符的置信度最小值

words_result字段包含多个object，见以下参数

字段	说明
+ word_name	字段名，包括：姓名、性别、年龄、科室、入院时间、出院时间、住院号、出院诊断、出院医嘱
+ word	word_name字段对应的识别结果

返回示例

```JSON
```

```
{
   "words_result_num": 9,
   "words_result": [
      {
         "word": "473924782",
         "word_name": "住院号"
      },
      {
         "word": "2022-01-01",
         "word_name": "入院时间"
      },
      {
         "word": "无",
         "word_name": "出院医嘱"
      },
      {
         "word": "2022-01-07",
         "word_name": "出院时间"
      },
      {
         "word": "建议随访",
         "word_name": "出院诊断"
      },
      {
         "word": "张国立",
         "word_name": "姓名"
      },
      {
         "word": "56",
         "word_name": "年龄"
      },
      {
         "word": "男",
         "word_name": "性别"
      },
      {
         "word": "心脑外科",
         "word_name": "科室"
      }
   ],
   "log_id": 1545239263197406867
}
```


门诊病历识别

接口描述

支持识别全国各地各医院门诊病历的姓名、日期、诊断、检查、主诉、现病史 6个关键字段，及表格区清单项目名称、规格、单价、数量、金额、项目类型等字段。

申请试用

该接口正在邀测中，在正式使用之前，请先提交[合作咨询](https://ai.baidu.com/consultation/cooperation?referrerUrl=/tech/ocr_medical)，或者[提交工单](https://ticket.bce.baidu.com/#/ticket/create~productId=96)，提供公司名称、appid、应用场景，工作人员协助开通权限后方可使用。

请求说明

请求示例

HTTP 方法：`POST`

HTTP 方法：POST

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_outpatient`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省**</br>**- false：**不返回字段位置信息</br>**- true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>**- false：**不返回字段识别结果的置信度</br>**- true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_outpatient?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
门诊病历识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_outpatient"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 门诊病历识别
*/
public class MedicalOutpatient{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalOutpatient() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_outpatient";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalOutpatient.medicalOutpatient();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_outpatient";
static std::string medicalOutpatient_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalOutpatient_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 门诊病历识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalOutpatient(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalOutpatient_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_outpatient?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalOutpatient
    {
        // 门诊病历识别
        public static string medicalOutpatient()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_outpatient?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("门诊病历识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
CommonData_result_num	是	uint32	患者个人信息的识别结果数，表示CommonData的元素个数
CostDetail_row_num	是	uint32	具体项目行数，表示CostDetail中的数组个数
+ CommonData	是	array[]	患者个人信息
+ CostDetail	是	array[]	具体项目
++ word_name	是	string	字段名，详见下方表格区说明
++ word	是	string	word_name字段对应的识别结果
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

CommonData字段包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：姓名、日期、诊断、检查、主诉、现病史
++ word	word_name字段对应的识别结果

CostDetail字段包含多个array，每个数组包含多个object，见以下参数

| 字段 | 说明 |

字段	说明
++ word_name	字段名，包括：清单项目名称、规格、单价、数量、金额、项目类型
++ word	word_name字段对应的识别结果

返回示例

```JSON
{
    "CommonData_result_num": 6,
    "CostDetail_row_num": 2,
    "words_result": {
        "CommonData": [
            {
                "word_name": "姓名",
                "word": "马奕冉"
            },
            {
                "word_name": "主诉",
                "word": "鼻堵"
            },
            {
                "word_name": "检查",
                "word": "鼻粘膜肿胀、苍白,双下甲肿大。"
            },
            {
                "word_name": "日期",
                "word": "2019年05月10日"
            },
            {
                "word_name": "诊断",
                "word": ""
            },
            {
                "word_name": "现病史",
                "word": ""
            }
        ],
        "CostDetail": [
            [
                {
                    "word_name": "清单项目名称",
                    "word": "糖酸莫米松鼻喷雾剂"
                },
                {
                    "word_name": "规格",
                    "word": "50ug140喷/支"
                },
                {
                    "word_name": "单价",
                    "word": "58"
                },
                {
                    "word_name": "数量",
                    "word": "1"
                },
                {
                    "word_name": "金额",
                    "word": "58"
                },
                {
                    "word_name": "项目类型",
                    "word": ""
```

```
            "word":
          }
        ],
        [
          {
            "word_name": "清单项目名称",
            "word": "孟鲁司特钠咀嚼片"
          },
          {
            "word_name": "规格",
            "word": "4mg×5片/盒"
          },
          {
            "word_name": "单价",
            "word": "98"
          },
          {
            "word_name": "数量",
            "word": "1"
          },
          {
            "word_name": "金额",
            "word": "98"
          },
          {
            "word_name": "项目类型",
            "word": ""
          }
        ]
      ]
    },
    "log_id": 1545241096948571787
}
```


处方笺识别

接口描述

支持识别全国各地各医院处方笺的姓名、日期、病人ID、科别 4个关键字段，及表格区清单项目名称、规格、单价、数量、金额、频率、用量、用法等字段。

申请试用

该接口正在邀测中，在正式使用之前，请先提交[合作咨询](https://ai.baidu.com/consultation/cooperation?referrerUrl=/tech/ocr_medical)，或者[提交工单](https://ticket.bce.baidu.com/#/ticket/create~productId=96)，提供公司名称、appid、应用场景，工作人员协助开通权限后方可使用。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_prescription`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省**</br>**- false：**不返回字段位置信息</br>**- true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>**- false：**不返回字段识别结果的置信度</br>**- true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_prescription?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
##### encoding:utf-8

import requests
import base64

'''
处方笺识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_prescription"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 处方笺识别
*/
public class MedicalPrescription{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalPrescription() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_prescription";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalPrescription.medicalPrescription();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_prescription";
static std::string medicalPrescription_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalPrescription_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 处方笺识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalPrescription(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalPrescription_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_prescription?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalPrescription
    {
        // 处方笺识别
        public static string medicalPrescription()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_prescription?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("处方笺识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
CommonData_result_num	是	uint32	患者个人信息的识别结果数，表示CommonData的元素个数
CostDetail_row_num	是	uint32	具体项目行数，表示CostDetail中的数组个数
+ CommonData	是	array[]	患者个人信息
+ CostDetail	是	array[]	具体项目
++ word_name	是	string	字段名，详见下方表格区说明
++ word	是	string	word_name字段对应的识别结果
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值

CommonData字段包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：姓名、日期、病人ID、科别
++ word	word_name字段对应的识别结果

CostDetail字段包含多个array，每个数组包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：清单项目名称、规格、单价、数量、金额、频率、用量、用法
++ word	word_name字段对应的识别结果

返回示例

```JSON
{
    "CommonData_result_num": 4,
    "CostDetail_row_num": 3,
    "words_result": {
        "CommonData": [
            {
                "word_name": "病人ID",
                "word": "000823203000"
            },
            {
                "word_name": "姓名",
                "word": "石彩云"
            },
            {
                "word_name": "日期",
                "word": ""
            },
            {
                "word_name": "科别",
                "word": ""
            }
        ],
        "CostDetail": [
            [
                {
                    "word_name": "清单项目名称",
                    "word": "雷贝拉唑钠肠溶胶囊"
                },
                {
                    "word_name": "规格",
                    "word": "20mg×7粒/盒"
                },
                {
                    "word_name": "数量",
                    "word": "2.00"
                },
                {
                    "word_name": "频率",
                    "word": "2/日"
                },
                {
                    "word_name": "用量",
                    "word": "1粒/次"
                },
                {
                    "word_name": "用法",
                    "word": "口服"
                },
                {
                    "word_name": "单价",
                    "word": ""
                },
                {
                    "word_name": "金额"
```

```
            "word_name": "金额",
            "word": ""
        }
    ],
    [
        {
            "word_name": "清单项目名称",
            "word": "葛根芩连丸"
        },
        {
            "word_name": "规格",
            "word": "3g×6袋/盒"
        },
        {
            "word_name": "数量",
            "word": "4.00"
        },
        {
            "word_name": "频率",
            "word": "3/日"
        },
        {
            "word_name": "用量",
            "word": "1袋/次"
        },
        {
            "word_name": "用法",
            "word": "口服"
        },
        {
            "word_name": "单价",
            "word": ""
        },
        {
            "word_name": "金额",
            "word": ""
        }
    ],
    [
        {
            "word_name": "清单项目名称",
            "word": "洁白胶囊"
        },
        {
            "word_name": "规格",
            "word": "0.4g×36粒/盒"
        },
        {
            "word_name": "数量",
            "word": "2.00"
        },
        {
            "word_name": "频率",
            "word": "3/日"
        },
        {
            "word_name": "用量",
            "word": "3粒/次"
        },
        {
            "word_name": "用法",
            "word": "口服"
        },
```

```
            {
                "word_name": "单价",
                "word": ""
            },
            {
                "word_name": "金额",
                "word": ""
            }
        ]
    ]
  },
  "log_id": 1545241586768488669
}
```

手术记录识别

接口描述

支持识别全国各地各医院手术记录的姓名、性别、年龄、科室、麻醉方式、手术操作名称、手术步骤、手术结果、术中所见、术中诊断、术前诊断、术后诊断 12个关键字段。

申请试用

该接口正在邀测中，在正式使用之前，请先提交[合作咨询](https://ai.baidu.com/consultation/cooperation?referrerUrl=/tech/ocr_medical)，或者[提交工单](https://ticket.bce.baidu.com/#/ticket/create~productId=96)，提供公司名称、appid、应用场景，工作人员协助开通权限后方可使用。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_surgery`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>

请注意关闭URL防盗链 |
| location | 否 | true/false | - |是否返回字段的位置信息，**默认为 false，可缺省**</br>**- false：**不返回字段位置信息</br>**- true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）|
| probability | 否 | true/false | - |是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>**- false：**不返回字段识别结果的置信度</br>**- true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min） |

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_surgery?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

```python label=Python
##### encoding:utf-8

import requests
import base64

'''
手术记录识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_surgery"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
~~~

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 手术记录识别
*/
public class MedicalSurgery{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalSurgery() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_surgery";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalSurgery.medicalSurgery();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_surgery";
static std::string medicalSurgery_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalSurgery_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 手术记录识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalSurgery(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalSurgery_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_surgery?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalSurgery
    {
        // 手术记录识别
        public static string medicalSurgery()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_surgery?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("手术记录识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~
```

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ word_name	是	string	字段名，详见下方表格区说明
+ word	是	string	word_name字段对应的识别结果
+ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
++ top	否	uint32	字段的上边距
++ left	否	uint32	字段的左边距
++ height	否	uint32	字段的高度
++ width	否	uint32	字段的宽度
+ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
++ average	否	float	字段识别结果中各字符的置信度平均值
++ min	否	float	字段识别结果中各字符的置信度最小值

words_result字段包含多个object，见以下参数

字段	说明
+ word_name	字段名，包括：姓名、性别、年龄、科室、麻醉方式、手术操作名称、手术步骤、手术结果、术中所见、术中诊断、术前诊断、术后诊断
+ word	word_name字段对应的识别结果

返回示例

```JSON
```

```
{
    "words_result_num": 12,
    "words_result": [
        {
            "word": "郭一中",
            "word_name": "姓名"
        },
        {
            "word": "48岁",
            "word_name": "年龄"
        },
        {
            "word": "男",
            "word_name": "性别"
        },
        {
            "word": "肠粘连松解+胃部分切除术",
            "word_name": "手术操作名称"
        },
        {
            "word": "取平卧位,麻醉成功后,术野消毒铺单。沿有左侧腹手术痕作纵梭切#1,切除原有手术疤痕,逐层切开入腹腔,注意保护切口。部分小肠与原腹壁切口、部补小肠与大网膜、部分小肠及系膜广泛粘连呈团,予以锐性解剖开,仔细检查肠壁完整。肿瘤位于胃体上段大弯侧,直径约4.5cm,肿瘤与胃壁关系密切。游离胃大弯及肿物,注意保护膈肌及璃腺,距肿物边缘1.0cm用一次性线形切割吻合器闭合并切断胃壁,完整切除肿物,移走标本,新端消毒,远端浆肌层加强缝合满意。",
            "word_name": "手术步骤"
        },
        {
            "word": "",
            "word_name": "手术结果"
        },
        {
            "word": "",
            "word_name": "术中所见"
        },
        {
            "word": "",
            "word_name": "术中诊断"
        },
        {
            "word": "",
            "word_name": "术前诊断"
        },
        {
            "word": "",
            "word_name": "术后诊断"
        },
        {
            "word": "外科二病区",
            "word_name": "科室"
        },
        {
            "word": "",
            "word_name": "麻醉方式"
        }
    ],
    "log_id": 1556909218057240008
}
```


医疗票据类别检测

> **该接口已停止更新且即将下线，如需更丰富、效果更佳的分类功能，请使用 [文件检测分类](https://ai.baidu.com/ai-doc/OCR/qlor1ahik)，此服务支持超过 200 种票据、卡证、文档的检测与分类，包括医疗发票、医疗费用明细、医疗费用结算单、病案首页、出院小结、增值税发票等。**

接口描述

支持对医疗发票、医疗费用明细、医疗费用结算单、病案首页、出院小结、增值税发票6项医疗票据的分类，并返回对应票据的图片质量及位置方向信息。

申请试用

该接口正在邀测中，在正式使用之前，请先提交[合作咨询](https://ai.baidu.com/consultation/cooperation?referrerUrl=/tech/ocr/medical_detail)，或者[提交工单](https://ticket.bce.baidu.com/#/ticket/create~productId=96)，提供公司名称、appid、应用场景，工作人员协助开通权限后方可使用。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/medical_recipts_classify`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_recipts_classify?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
医疗票据类别检测
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_recipts_classify"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 医疗票据类别检测
*/
public class MedicalReciptsClassify{

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String medicalReciptsClassify() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_recipts_classify";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        MedicalReciptsClassify.medicalReciptsClassify();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_recipts_classify";
static std::string medicalReciptsClassify_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    medicalReciptsClassify_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 医疗票据类别检测
* @return 调用成功返回0，发生错误返回其他错误码
*/
int medicalReciptsClassify(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = medicalReciptsClassify_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/medical_recipts_classify?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class MedicalReciptsClassify
    {
        // 医疗票据类别检测
        public static string medicalReciptsClassify()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/medical_recipts_classify?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
```

```
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("医疗票据类别检测:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

~~~



###### 返回说明

**返回参数**

| 字段                | 是否必输出 | 类型   | 说明 |
| ------------------- | -------- | ------ | ---------------------------- |
| log_id              | 是       | uint64 | 调用日志id，用于问题定位 |
| words_result        | 是       | object | 识别结果 |
| words_result_num    | 是       | uint32 | 识别结果数，表示words_result的元素个数 |
| + quality    | 是       | int：0或者1 | **票据图像质量**</br>1：表示票据质量良好</br>0：表示票据质量不佳 |
| + class          | 是       | string| **票据图像的分类结果，具体返回值的中英文对照：**</br>medical_invoice    医疗发票</br>medical_detail    医疗费用明细</br>medical_statement  医疗费用结算单</br>medical_summary    出院小结</br>medical_record  病案首页</br>medical_vat        增值税发票</br>others             其他</br> |
| + direction          | 是       |  int：0,1,2,3 | **票据图像的方向**</br>- 0：正向，</br>- 1：逆时针90度，</br>- 2：逆时针180度，</br>- 3：逆时针270度 |
| +  location | 是 | object | **票据图像的位置信息** |
| ++  top | 是 | uint32 | 票据图像定位位置的长方形的上边距 |
| ++  left | 是 | uint32 | 票据图像定位位置的长方形的左边距 |
|++  height | 是 | uint32 | 票据图像定位位置的长方形的高度 |
| ++  width | 是 | uint32 | 票据图像定位位置的长方形的宽度 |



**返回示例**

```JSON
{
    "words_result": [
        {
            "direction": 0,
            "class": "medical_invoice",
            "location": {
```
```

```
            "height": 1000,
            "top": 69,
            "width": 1469,
            "left": 80
        },
        "quality": 0
    }
  ],
  "words_result_num": 1,
  "log_id": 1457967830499025185
}
```


教育场景文字识别

试卷分析与识别

接口描述

可对文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景，支持公式识别、手写竖式识别。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=教育场景OCR&api=rest/2.0/ocr/v1/doc_analysis&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px</br>**优先级**：image > url > pdf_file，当image、url字段存

小不超过10M，最短边至少15px，最长边最大8192px</br> 优先级：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效|
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
language_type	否	string	CHN_ENG/ ENG	识别语言类型，默认为CHN_ENG 可选值包括： = CHN_ENG：中英文 = ENG：英文
result_type	否	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 = big：返回行识别结果 = small：返回行识别结果之上还会返回单字结果
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	否	string	true/false	是否返回每行识别结果的置信度。默认为false
disp_line_poly	否	string	true/false	是否返回每行的四角点坐标。默认为false
words_type	否	string	handwring_only/ handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	否	string	true/false	是否分析文档版面：包括layout（图、表、标题、段落、目录）；attribute（栏、页眉、页脚、页码、脚注）的分析输出
recg_formula	否	string	true/false	是否检测并识别公式，默认为false，公式以 Latex 格式文本返回。 =true：检测并识别公式 =false：不检测识别公式
recg_long_division	否	string	true/false	是否检测并识别手写竖式，默认为false。 =true：检测并识别手写竖式 =false：不检测手写竖式
disp_underline_analysis	否	string	true/false	是否开启下划线识别功能，可选值如下： =true：开启，在返回参数 underline 内输出下划线信息 =false：关闭，默认值，不输出下划线信息
recg_alter	否	string	true/false	是否开启返回涂改识别结果功能，可选值如下： =true：开启检测，涂改部分统一用“〓”返回 =false：关闭，默认值，不输出涂改识别结果

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
##### 试卷分析与识别
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis?access_token=【调用鉴权接口获取的token】' --data 'language_type=CHN_ENG&result_type=big&image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
试卷分析与识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img,"language_type":"CHN_ENG","result_type":"big"}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 文档版面分析与识别
*/
public class DocAnalysis {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String docAnalysis() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "language_type=" + "CHN_ENG" + "&result_type=" + "big" + "&image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可
自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        DocAnalysis.docAnalysis();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis";
static std::string docAnalysis_result;
/**
 * curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
 * @param 参数定义见libcurl文档
 * @return 返回值定义见libcurl文档
 */
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    docAnalysis_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
 * 文档版面分析与识别
 * @return 调用成功返回0，发生错误返回其他错误码
 */
int docAnalysis(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "language_type", CURLFORM_COPYCONTENTS, "CHN_ENG",
CURLFORM_END);
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "result_type", CURLFORM_COPYCONTENTS, "big",
CURLFORM_END);
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = docAnalysis_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```php
 <?php
/**
 * 发起http post请求(REST API), 并获取REST请求的结果
 * @param string $url
 * @param string $param
 * @return - http response body if succeeds, else false.
 */
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'language_type' => "CHN_ENG",
    'result_type' => "big",
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class DocAnalysis
    {
        // 文档版面分析与识别
        public static string docAnalysis()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_analysis?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "language_type=" + "CHN_ENG" + "&result_type=" + "big" + "&image=" +
HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("文档版面分析与识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
img_direction	否	int32	detect_direction=true 时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	是	uint32	识别结果数，表示results的元素个数
results	是	array[]	识别结果数组
+ words_type	是	string	文字属性（手写、印刷），handwriting 手写，print 印刷
+ words	是	array[]	整行的识别结果数组。
++ line_probability	否	array[]	line_probability=true 时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++ average	否	float	行置信度

+++ min	否	float	整行中单字的最低置信度
++ word	是	string	整行的识别结果
++ poly_location	否	array[]	是否返回每行的四角点坐标，disp_line_poly=true时返回
++ words_location	是	array[]	整行的矩形框坐标。位置信息（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位定位位置的长方形的宽度
+++ height	是	uint32	表示位置的长方形的高度
+ chars	否	array[]	result_type=small 时返回。单字符结果数组
++ char	否	string	result_type=small 时返回。每个单字的内容
++ chars_location	否	object	每个单字的矩形框坐标。位置信息（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
formula_result	否	array[]	识别结果中的公式数组，包括公式位置和公式内容，</br>recg_formula=true时返回
+ form_location	否	array[]	识别结果中公式的矩形框坐标数组（坐标0点为左上角）
+ form_words	否	string	识别结果中公式的内容
words_result	否	array[]	将普通文字和公式融合后的识别结果数组，</br>recg_formula=true时返回
+ location	否	array[]	识别结果中整行的矩形框坐标数组（坐标0点为左上角）
+ words	否	string	识别结果中整行的内容
+ chars	否	array[]	单字符结果数组，公式整体作为一个单字，</br>result_type=small 时返回
++ char	否	string	每个单字的内容
++ chars_location	否	object	每个单字的矩形框坐标数组（坐标0点为左上角）
layouts_num	否	uint32	版面分析结果数，表示layout的元素个数
layouts	否	array[]	每个「栏：section」里面的文档版面模块数组，包含表格、图、段落文本、标题、目录等5个模块；每个模块的坐标位置；段落文本和表格内文本内容对应的行序号id。
+ layout	否	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title ，目录:contents
+ layout_location	否	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	水平坐标（坐标0点为左上角）
+ layout_idx	否	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）
sec_rows	否	uint32	将所有的版面中的「栏:section」内容表示成 M x N 的网格，sec_rows = M
sec_cols	否	uint32	将所有的版面中的「分栏」内容表示成 M x N 的网格，sec_cols = N
sections	否	array[]	一张图片中包含的5大版面属性，包含：栏，页眉，页脚，页码，脚注，该数组里有属性的标签、属性的位置、属性所包含文本内容的id序号。 其中，栏（section）里面包含5个模块内容，有：表格、图、段落文本、标题、目录（在返回参数layouts里输出）
+ attribute	否	string	版面分析的属性标签结果，栏:section, 页眉:header, 页脚:footer, 页码:number，脚注:footnote
+ attri_location	否	array[]	版面分析的属性所在位置，四个顶点: 左上，右上，右下，左下
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	水平坐标（坐标0点为左上角）
+ sec_idx	否	string	sections返回参数中的5个版面属性里，包含的内容序号标识
++ idx	否	string	sections返回参数中的5个版面属性里，每个属性下包含的文本行id序号
++ para_idx	否	string	当且仅当attribute=section时才会返回。表示，返回参数中的「栏：section」里面，所包含的表格、图、段落文本、标题、目录等5个模块返回的顺序号id（即layouts返回结果中，每个模块的返回顺序号）
++ row_idx	否	string	当且仅当attribute=section时才会返回。表示，将所有栏表示成 M xN 的网格，所属网格的行的id
++ col_idx	否	string	当且仅当attribute=section时才会返回。表示，将所有栏表示成 M xN 的网格，所属网格的列的id
+ long_division	否	array[]	手写竖式识别结果，当 recg_long_division=true 时返回
+ location	否	object	手写竖式的矩形框坐标数组（坐标0点为左上角）
+ words	否	object	按行输出手写竖式内文字结果
++ word	否	string	每行文字的内容
++ words_location	否	object	每行的矩形框坐标数组（坐标0点为左上角）
+ long_division_num	否	uint32	手写竖式识别结果数，表示 long_division 的元素个数，当recg_long_division=true 时返回
underline	否	array[]	识别到的下划线结果，当 disp_underline_analysis=true 时返回
+ points	否	object	下划线坐标信息
++ start_x	否	uint32	下划线起点 x 坐标

++ start_y	否	uint32	下划线起点 y 坐标
++ end_x	否	uint32	下划线终点 x 坐标
++ end_y	否	uint32	下划线终点 y 坐标
+ prob	否	uint32	下划线置信度，取值范围在 [0，1] 之间
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```JSON
{
 "results_num": 6,
 "log_id": "4488766695474114139",
 "img_direction": 0,
 "layouts_num": 0,
 "results": [
  {
   "words_type": "print",
   "words": {
    "words_location": {
     "top": 124,
     "left": 136,
     "width": 418,
     "height": 65
    },
    "word": "五默写(4分)"
   },
  },
  {
   "words_type": "print",
   "words": {
    "words_location": {
     "top": 246,
     "left": 136,
     "width": 37,
     "height": 45
    },
    "word": "1"
   },
  },
  {
   "words_type": "handwriting",
   "words": {
    "words_location": {
     "top": 195,
     "left": 237,
     "width": 469,
     "height": 104
    },
    "word": "采菊东篱下"
   },
  },
  {
   "words_type": "print",
   "words": {
    "words_location": {
     "top": 241,
     "left": 889,
     "width": 287,
     "height": 52
    },
    "word": "悠然见南山?"
   },
```

```
  },
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 415,
    "left": 134,
    "width": 472,
    "height": 52
   },
   "word": "2.商女不知亡国恨"
  },
 },
 {
  "words_type": "handwriting",
  "words": {
   "words_location": {
    "top": 377,
    "left": 607,
    "width": 556,
    "height": 93
   },
   "word": "隔江犹唱后庭花。"
  },
 },
],
 "formula_result": [
      {
          "form_location": {
              "top": 0,
              "left": 97,
              "width": 151,
              "height": 77
          },
          "form_words": " x = \\frac { 1 } { n - 1 } - 1 1 \\frac { \\frac { 5 } { 2 } } { 5 }"
      },
      {
          "form_location": {
              "top": 119,
              "left": 118,
              "width": 115,
              "height": 80
          },
          "form_words": " = \\sqrt { \\frac { x } { 2 } ( x - 1 ) ^ { 2 } }"
      },
      {
          "form_location": {
              "top": 196,
              "left": 78,
              "width": 17,
              "height": 24
          },
          "form_words": " x ^ { 2 }"
      },
      {
          "form_location": {
              "top": 244,
              "left": 79,
              "width": 103,
              "height": 70
          },
          "form_words": " s = \\frac { \\sum _ { i = 0 } { m } \\cdot i v } { - 1 }"
```

```
      }
   ],
   "words_result": [
      {
         "location": {
            "top": 164,
            "left": 255,
            "width": 111,
            "height": 16
         },
         "words": "其中m表示考生"
      },
      {
         "location": {
            "top": 198,
            "left": 24,
            "width": 341,
            "height": 18
         },
         "words": "的人数 x ^ { 2 } 表示的是滴个考上的第i题等分，"
      },
   ],
}
```

公式识别

> **该接口的公有云服务即将下线，若您仍需要使用公式识别，您可以选择[试卷分析与识别](https://ai.baidu.com/ai-doc/OCR/jk9m7mj1l)。** 试卷分析与识别也可支持公式识别，同时提供版面分析功能，可分版块输出OCR识别结果。

接口描述

支持对试卷中的数学公式及题目内容进行识别，可提取公式部分进行单独识别，也可对题目和公式进行混合识别，并返回Latex格式公式内容及位置信息，便于进行后续处理。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=教育场景OCR&api=rest/2.0/ocr/v1/formula&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/formula`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（）</br>要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置；small：定位单字符位置。默认值为big
detect_direction	否	bool	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
disp_formula	否	bool	true/false	是否分离输出公式识别结果，在words_result外单独输出公式结果，展示在“formula_result”中

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/formula?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
公式识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/formula"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())

```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;
```
~~~

```
/**
* 公式识别
*/
public class Formula {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String formula() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/formula";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        Formula.formula();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/formula";
static std::string formula_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    formula_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
```

```
}
/**
* 公式识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int formula(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = formula_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
```

```
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/formula?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class Formula
    {
        // 公式识别
        public static string formula()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/formula?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("公式识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
```

```
            return baser64;
        }
    }
}

```

返回说明

返回参数

字段	是否必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - - 1：未定义， - 0：正向， - 1： 逆时针90度， - 2：逆时针180度， - 3：逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
formula_result	否	bool	是否返回单独公式识别结果，默认false
formula_result_num	否	bool	识别结果中的公式个数，表示formula_result的元素个数
words_result	是	array[]	识别结果数组
+ location	是	object{}	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串

返回示例


```
{
  "log_id": 2671713289176456793,
  "direction": 0,
 "formula_result_num": 3,
  "formula_result": [
     {
        "location": {
           "width": 258,
           "top": 265,
           "left": 450,
           "height": 204
        },
        "words": "\\left\\{ \\begin{aligned} & x = - 1 1 \\\\ & y = 2 \\\\ \\end{aligned} \\right. "
     },
     {
        "location": {
           "width": 429,
           "top": 546,
           "left": 310,
           "height": 203
        },
        "words": "\\left\\{ \\begin{aligned} & 3 x + 2 y = m \\\\ & n x - y = 2 \\\\ \\end{aligned} \\right. "
     },
```

```
        },
        {
            "location": {
                "width": 142,
                "top": 613,
                "left": 1029,
                "height": 71
            },
            "words": "m - \\left[ 1 0 0 , - \\infty \\right) "
        }
    ],
    "words_result_num": 5,
    "words_result": [
        {
            "location": {
                "width": 168,
                "top": 313,
                "left": 292,
                "height": 110
            },
            "words": "已知"
        },
        {
            "location": {
                "width": 258,
                "top": 265,
                "left": 450,
                "height": 204
            },
            "words": "\\left\\{ \\begin{aligned} & x = - 1 1 \\\\ & y = 2 \\\\ \\end{aligned} \\right. "
        },
        {
            "location": {
                "width": 582,
                "top": 319,
                "left": 728,
                "height": 84
            },
            "words": "是二元一次方程组"
        },
        {
            "location": {
                "width": 429,
                "top": 546,
                "left": 310,
                "height": 203
            },
            "words": "\\left\\{ \\begin{aligned} & 3 x + 2 y = m \\\\ & n x - y = 2 \\\\ \\end{aligned} \\right."
        },
        {
            "location": {
                "width": 780,
                "top": 597,
                "left": 745,
                "height": 88
            },
            "words": "的解,则 m - \\left[ 1 0 0 , - \\infty \\right) 的值是()"
        }
    ]
}
```


词典笔文字识别

接口描述

面向词典笔应用场景，是词典笔的基本功能之一，主要用于扫描文字并识别，为字词查询和翻译提供文本信息支撑。支持中英文识别、印刷手写混排识别，满足多种复杂背景下的文字扫描识别。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/pen`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
probability	否	string	true/false	是否返回识别结果中每一行的置信度

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	识别结果数组
+ words	否	string	识别结果字符串
+ location	是	array[]	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时返回该字段
++ char	否	string	单字符识别结果，当 recognize_granularity=small 时返回该字段
++ location	否	array[]	位置数组（坐标0点为左上角），当 recognize_granularity=small 时返回该字段
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标，当 recognize_granularity=small 时返回该字段
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标，当 recognize_granularity=small 时返回该字段
+++ width	否	uint32	表示定位位置的长方形的宽度，当 recognize_granularity=small 时返回该字段
+++ height	否	uint32	表示定位位置的长方形的高度，当 recognize_granularity=small 时返回该字段
+ probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值，当 probability=true 时返回该字段
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例


```
{
  "words_result_num": 1,
  "words_result": [
    {
      "probability": {
        "average": 0.9977043867,
        "min": 0.974070549,
        "variance": 0.00003904080586
      },
      "words": "了文献研究法和调查问卷法。广泛收集和查阅国内外相关文",
      "location": {
        "top": 848,
        "left": 98,
        "width": 1087,
        "height": 47
      }
```

```
        }
    ],
    "direction": 0,
    "log_id": 1663011878006764485
}
```

试卷切题识别

接口描述

支持对图片/PDF格式文档内的题目自动切分与结构化识别，可按题输出题干、选项、答案等信息，适用于整页试卷、习题册、课本等，可广泛应用于拍照搜题、题库录入、智能判卷等场景

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/#/api?product=AI&project=文字识别&parent=教育场景OCR&api=rest/2.0/ocr/v1/paper_cut_edu&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

申请试用
该接口正在邀测中，**请您先提交 [合作咨询](https://ai.baidu.com/consultation/cooperation?referrerUrl=/ai-doc/OCR/k9m7mj1l) 或 [提交工单](https://console.bce.baidu.com/ticket/#/)**，提供公司名称、appid、应用场景等信息，工作人员协助开通权限后方可使用。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/paper_cut_edu`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 </br>**优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式</br>**优先级**：image > url > pdf_file，当image字段存在时，url字段失效</br>**请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px</br>**优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内

容，若不传入，则默认识别第 1 页|
language_type	否	string	CHN_ENG/ ENG	识别语言类型，默认为CHN_ENG。可选值包括： = CHN_ENG：中英文 = ENG：英文，纯英文场景下建议开启
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括： true ：检测朝向，输入非正向图片时建议开启 false：不检测朝向
words_type	否	string	handprint_mix/handwring_only	识别文字类型，默认为手写印刷混排识别，即：handprint_mix。可选值包括： = handprint_mix：手写印刷混排 = handwring_only：手写，纯手写场景下建议开启
splice_text	否	string	true/false	是否拼接题目元素内每行的文本信息后输出，默认不拼接，即：false。开启该参数后，处理耗时预计会增加 1s。可选值包括： true ：拼接题目元素每行的文本信息，在elem_text 内输出； false：不拼接，仅按行输出文本信息

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	检测到的图像朝向，当 detect_direction=true 时返回。0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
qus_result_num	是	int32	识别题目结果数，表示 qus_result 的元素个数
qus_figure	是	array[]	试卷内题目图片信息
+fig_location	是	array[]	题目图片位置的四角点坐标，坐标 0 点为左上角，顺时针返回
qus_result	是	array[]	试卷切题信息
+qus_type	是	int32	检测到的题目类型。0：选择题；1：判断题；2：填空题；3：问答题；4：其他
+qus_probability	是	float	题目置信度
+elem_text	否	object	题目各元素的完整文本信息，当 splice_text = true 时，拼接题目各元素内每行的文本信息后输出
++stem_text	否	string	题干文本信息
++subqus_text	否	string	子题文本信息
++answer_text	否	string	答案文本信息
++option_text	否	string	选项文本信息，仅在题目类型为选择题时输出
++interpretation_text	否	string	参考答案文本信息
+qus_location	是	array[]	题目位置四角点坐标，坐标 0 点为左上角，顺时针返回
+qus_element	是	array[]	题目元素信息
++elem_type	是	int32	题目元素类型。0：题干；1：子题；2：答案；3：选项；4：配图；5：参考答案
++elem_probability	是	float	题目元素置信度
++elem_location	是	array[]	题目元素位置四角点坐标，坐标 0 点为左上角，顺时针返回
++elem_word	是	array[]	题目元素的文本信息
+++word_location	是	array[]	按行返回文字位置信息，坐标 0 点为左上角，顺时针返回
+++word_type	是	string	按行返回文字属性信息。handwriting：手写；print：印刷
+++word	是	string	按行返回文字信息
pdf_file_size	否	int32	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

其他场景文字识别

仪器仪表盘读数识别

接口描述

适用于不同品牌、不同型号的仪器仪表盘读数识别，广泛适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号，支持液晶屏、字轮表等表型。

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=其他场景OCR&api=rest/2.0/ocr/v1/meter&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/meter`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px。支持jpg/jpeg/png/bmp格式.**注意：图片的base64编码是不包含图片头的，如（）**
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
probability	否	string	true/false	是否返回每行识别结果的置信度。默认为false
poly_location	否	string	true/false	位置信息返回形式，默认：false false：只给出识别结果所在长方形位置信息 true：除了默认的识别文字所在长方形的位置信息，还会给出文字所在区域的最小外接旋转矩形的4个点坐标信息

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
##### 仪器仪表盘读数识别
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/meter?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
~~~

```
 ##### encoding:utf-8

import requests
import base64

'''
仪器仪表盘读数识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/meter"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```

```
 package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 仪器仪表盘读数识别
*/
public class Meter {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String meter() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/meter";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        Meter.meter();
    }
}
```

```
 ##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/meter";
static std::string meter_result;
/**
 * curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
 * @param 参数定义见libcurl文档
 * @return 返回值定义见libcurl文档
 */
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    meter_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
 * 仪器仪表盘读数识别
 * @return 调用成功返回0，发生错误返回其他错误码
 */
int meter(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】",
CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = meter_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}
```

```
 <?php
/**
 * 发起http post请求(REST API), 并获取REST请求的结果
 * @param string $url
 * @param string $param
 * @return - http response body if succeeds, else false.
 */
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/meter?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
 using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class Meter
    {
        // 仪器仪表盘读数识别
        public static string meter()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/meter?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("仪器仪表盘读数识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	是	string	识别结果字符串
+ location	是	array[]	识别结果所在长方形位置信息
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ probability	否	string	probability=true 时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值

| + poly_location | 否 | array[] | poly_location=true 时存在。文字所在区域的外接四边形的4个点坐标信息 |

返回示例

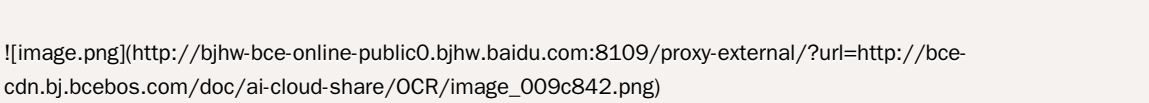
```JSON

{
    "log_id": "1392680790663364608",
 "words_result_num": 5
 "words_result": [
  {
   "words": "5.8",
   "location": {
    "top": 150,
    "left": 370,
    "width": 87,
    "height": 79
   }
  },
  {
   "words": "mmol/L",
   "location": {
    "top": 241,
    "left": 402,
    "width": 52,
    "height": 12
   }
  },
  {
   "words": "10:38",
   "location": {
    "top": 115,
    "left": 347,
    "width": 42,
    "height": 21
   }
  },
  {
   "words": "12-11",
   "location": {
    "top": 116,
    "left": 410,
    "width": 36,
    "height": 20
   }
  },
  {
   "words": "am",
   "location": {
    "top": 115,
    "left": 391,
    "width": 12,
    "height": 5
   }
  }
 ],
}

### 门脸文字识别
```

接口描述

针对含有门脸/门头的图片进行专项优化，支持识别门脸/门头上的文字内容。

在百度云控制台的位置

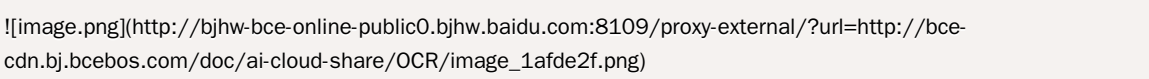
进入[文字识别的的百度云控制台概览页面](https://console.bce.baidu.com/ai/?_=1635220514682&fromai=1&locale=zh-cn#/ai/ocr/app/list)，门脸文字识别在如下图所示位置：



在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/?_=1668473684721#/api?product=AI&project=文字识别&parent=其他场景OCR&api=rest/2.0/ocr/v1/facade&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

接口鉴权

进入[文字识别的的百度云控制台应用列表页面](https://console.bce.baidu.com/ai/?_=1635220514682&fromai=1&locale=zh-cn#/ai/ocr/app/list)，如下图所示：



如果还未创建应用，请点击「创建应用」按钮进行创建。创建应用后，参考鉴权参考文档，使用API Key(AK)和Secret Key(SK)获取access_token

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/facade`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，base64编码去除编码头（data:image/jpeg;base64），要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式

请求代码示例

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

~~~codeset
```bash label=Bash
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/facade?access_token=【调用鉴权接口获取的token】' --data
'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```
```python label=Python
##### encoding:utf-8

import requests
import base64

'''
门脸文字识别
'''

request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/facade"
##### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image":img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=params, headers=headers)
if response:
    print (response.json())
```
```java label=JAVA
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.Base64Util;
import com.baidu.ai.aip.utils.FileUtil;
import com.baidu.ai.aip.utils.HttpUtil;

import java.net.URLEncoder;

/**
* 门脸文字识别
*/
public class facade {

    /**
    * 重要提示代码中所需工具类
    * FileUtil,Base64Util,HttpUtil,GsonUtils请从
    * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
    * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
    * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
    * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
    * 下载
    */
    public static String facade() {
        // 请求url
        String url = "https://aip.baidubce.com/rest/2.0/ocr/v1/facade";
        try {
            // 本地文件路径
            String filePath = "[本地文件路径]";
```
~~~

```
            byte[] imgData = FileUtil.readFileByBytes(filePath);
            String imgStr = Base64Util.encode(imgData);
            String imgParam = URLEncoder.encode(imgStr, "UTF-8");

            String param = "image=" + imgParam;

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间， 客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        WebImage.webImage();
    }
}
```

```cpp label=C++
##### include <iostream>
##### include <curl/curl.h>

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "https://aip.baidubce.com/rest/2.0/ocr/v1/facade";
static std::string webImage_result;
/**
* curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
* @param 参数定义见libcurl文档
* @return 返回值定义见libcurl文档
*/
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    webImage_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
* 门脸文字识别
* @return 调用成功返回0，发生错误返回其他错误码
*/
int webImage(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_httppost *post = NULL;
        curl_httppost *last = NULL;
        curl_formadd(&post, &last, CURLFORM_COPYNAME, "image", CURLFORM_COPYCONTENTS, "【base64_img】", CURLFORM_END);

        curl_easy_setopt(curl, CURLOPT_HTTPPOST, post);
```

```
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, callback);
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = webImage_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```
```php label=PHP
<?php
/**
* 发起http post请求(REST API), 并获取REST请求的结果
* @param string $url
* @param string $param
* @return - http response body if succeeds, else false.
*/
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = 'https://aip.baidubce.com/rest/2.0/ocr/v1/facade?access_token=' . $token;
$img = file_get_contents('[本地文件路径]');
$img = base64_encode($img);
$bodys = array(
    'image' => $img
);
$res = request_post($url, $bodys);

var_dump($res);
```

```
```csharp label=C#
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Web;

namespace com.baidu.ai
{
    public class WebImage
    {
        // 门脸文字识别
        public static string webImage()
        {
            string token = "[调用鉴权接口获取的token]";
            string host = "https://aip.baidubce.com/rest/2.0/ocr/v1/facade?access_token=" + token;
            Encoding encoding = Encoding.Default;
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(host);
            request.Method = "post";
            request.KeepAlive = true;
            // 图片的base64编码
            string base64 = getFileBase64("[本地图片文件]");
            String str = "image=" + HttpUtility.UrlEncode(base64);
            byte[] buffer = encoding.GetBytes(str);
            request.ContentLength = buffer.Length;
            request.GetRequestStream().Write(buffer, 0, buffer.Length);
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);
            string result = reader.ReadToEnd();
            Console.WriteLine("门脸文字识别:");
            Console.WriteLine(result);
            return result;
        }

        public static String getFileBase64(String fileName) {
            FileStream filestream = new FileStream(fileName, FileMode.Open);
            byte[] arr = new byte[filestream.Length];
            filestream.Read(arr, 0, (int)filestream.Length);
            string baser64 = Convert.ToBase64String(arr);
            filestream.Close();
            return baser64;
        }
    }
}

```
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	否	array[]	定位和识别结果数组
+ words	否	string	识别结果字符串
+ score	否	float	words返回为主门脸名称的置信度评分
+ brief	否	string	门脸副标题等周边描述

返回示例

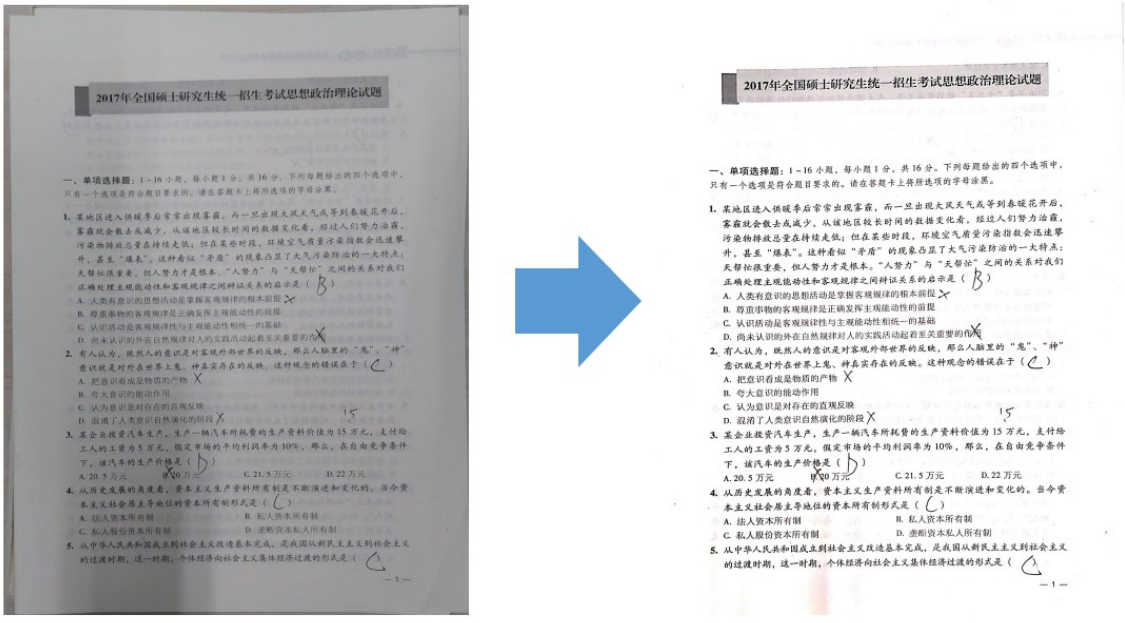

```
{
  "log_id": 341559937361307312,
  "words_result_num": 3,
  "words_result": [
        {
        "words": "生活超市",
        "brief": "家得利",
        "score": 0.80533
        },
        {
        "words": "火火火火锅",
        "score": 0.0112433
        },
        {
        "words": "天天好便利店",
        "score': 0.188124
        }
  ]
}
```


文档图像处理

文档矫正增强

接口描述

对图片中的文件、卡证、票据等内容进行四角点检测定位，提取主体内容并对其进行矫正，同时可选图片增强效果进一步提升图片清晰度，达到主体检测矫正并增强的目的，提升图片整体质量。示意图如下：



在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/doc_crop_enhance`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

参数	类型	是否必须	说明
image	string	和 url/pdf_file 三选一	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	string	和 image/pdf_file 三选一	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	string	和 image/url 三选一	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	string	否	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
scan_type	int32	否	选择是否对图片内主体内容进行四角点增强或矫正，可选值如下： - scan_type = 1：只做检测，不对主体进行矫正，返回主体四角点坐标，可用作前端页面展示 - scan_type = 2：只做矫正，需传入主体四角点坐标，使用传入的坐标值对主体进行扣取及矫正 - scan_type = 3：**默认值，**检测并矫正，返回主体在原图中的四角点坐标以及矫正后的图像
points	array	否	如 scan_type = 2，则需传入此参数，左上角起顺时针汇总四角点坐标为[{x1,y1},{x2,y2},{x3,y3},{x4,.y4}]
enhance_type	int32	否	选择是否开启图像增强功能，如开启可选择增强效果，可选值如下： - enhance_type =0：**默认值，**不开启增强功能 - enhance_type = 1：去阴影 - enhance_type = 2：增强并锐化 - enhance_type = 3：黑白滤镜

返回说明

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	唯一的log id，用于问题定位
image_processed	string	是	返回处理后的图片，base64编码，如请求参数 scan_type = 1&enhance_type =0，则返回原图
points	array[]	否	检测到的图片内主体在原图中的四角点坐标，scan_type = 2 时不返回此参数
pdf_file_size	string	否	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例


```
{
  "points": [
    {
      "x": 859,
      "y": 41
    },
    {
      "x": 854,
      "y": 546
    },
    {
      "x": 117,
      "y": 550
    },
    {
      "x": 104,
      "y": 49
    }
  ],
  "image_processed":"/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAYEBQYFBACUoKSj/2w",
  "log_id": 1540570874964208918
}
```


文档去手写

接口描述

去除图片中的手写内容，保留印刷体内容，可用于试卷去手写还原等场景。示意图如下：</br>
![3.JPG](https://bce.bdstatic.com/doc/ai-cloud-share/OCR/3_48a66d6.JPG)

在线调试
您可以在 [示例代码中心](https://console.bce.baidu.com/tools/#/api?product=AI&project=文字识别&parent=文档图像处理&api=/rest/2.0/ocr/v1/remove_handwriting&method=post) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/remove_handwriting`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	类型	是否必须	说明
image	string	和 url/pdf_file 三选一	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	string	和 image/pdf_file 三选一	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	string	和 image/url 三选一	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	string	否	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
enable_detect	string	否	是否去除文件边缘后再识别，默认为 true，可选值： - true：去除，适用于有背景/边缘的拍摄件 - false：不去除，适用于不含背景/边缘的扫描件

返回说明

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	唯一的log id，用于问题定位
image_processed	string	是	返回处理后的图片，base64编码
pdf_file_size	string	否	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

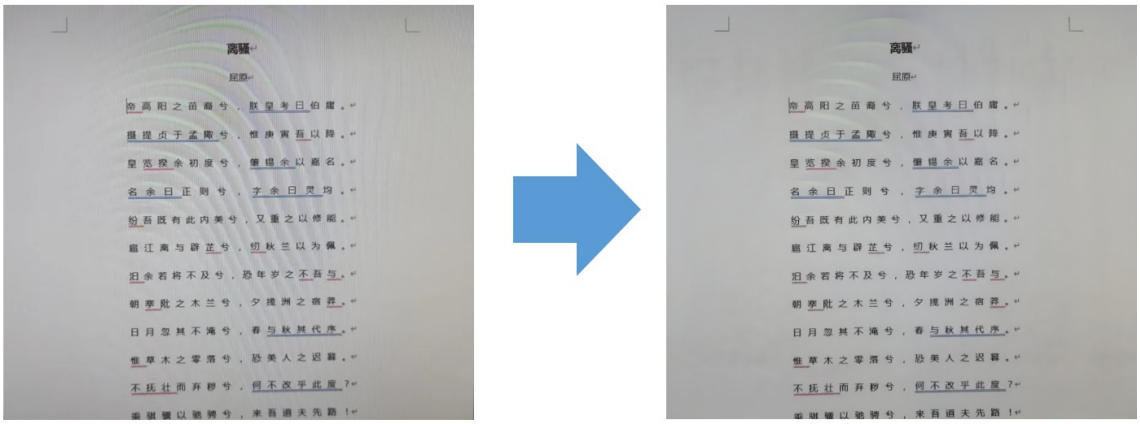
返回示例

```JSON
{
    "image_processed": "/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAYEBQYFB",
    "log_id": 1540571996316633631
}
```

图片去摩尔纹

接口描述

去除翻拍电脑、手机等显示屏照片中的摩尔纹，使图片更加清晰。示例图如下：



在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/image-process/v1/remove_moire`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	类型	是否必须	说明
image	string	和 url/pdf_file 三选一	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	string	和 image/pdf_file 三选一	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	string	和 image/url 三选一	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	string	否	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

返回说明

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	唯一的log id，用于问题定位
image_processed	string	是	返回处理后的图片，base64编码
pdf_file_size	string	否	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "image_processed": "/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAYEBQYFBAYG",
  "log_id": 1540571634473820843
}
```

文档图像去底纹

文件检测分类

接口描述

对图片中的文档、卡证、票据等含文字的主体进行检测、分类，可同时支持一张图片中多张主体的情况，返回每个主体的类别及位置信息。

请求说明

请求示例

HTTP 方法: `POST`

请求URL: `https://aip.baidubce.com/rest/2.0/ocr/v1/doc_classify`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	类型	是否必须	说明
image	string	和 url 二选一	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url，当image字段存在时，url字段失效
url	string	和 image 二选一	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url，当image字段存在时，url字段失效 **请注意关闭URL防盗链**

返回说明

返回参数

参数	类型	是否必输出	说明
log_id	uint64	是	唯一的日志id，用于问题定位
words_result	array[]	是	检测和分类结果数组，如图片中无文字内容，则此数组为空
+type	string	否	类别信息，当前可输出类别列表见下文「附录：类别列表」
+probablity	float	否	分类置信度
+location	object	否	位置数组（左上角为坐标0点）
++ left	uint32	否	表示定位位置的长方形左上顶点的水平坐标
++ top	uint32	否	表示定位位置的长方形左上顶点的垂直坐标
++ width	uint32	否	表示定位位置的长方形的宽度
++ height	uint32	否	表示定位位置的长方形的高度

返回示例


```
{
  "words_result": [
    {
      "probablity": 0.9999860525,
      "location": {
        "top": 1684,
        "left": 300,
        "width": 2710,
        "height": 1712
      },
      "type": "卡证_银行卡"
    }
  ],
  "log_id": "17778996938487203",
  "error_code": 0
}
```


附录：类别列表

序号	分类名称
1	通用场景
2	办公文档
3	教育文档
4	表格
5	卡证_不动产权证
6	卡证_车辆合格证
7	卡证_车牌
8	卡证_电子驾照
9	卡证_港澳返乡证_副
10	卡证_港澳返乡证_正
11	卡证_港澳通行证_副
12	卡证_港澳通行证_正
13	卡证_行驶证_副
14	卡证_行驶证_正
15	卡证_户口本_登记页

16	卡证_户口本_户主页
17	卡证_护照
18	卡证_护照_菲律宾
19	卡证_驾驶证_副
20	卡证_驾驶证_正
21	卡证_结婚证
22	卡证_军官证
23	卡证_离婚证
24	卡证_临时身份证_副
25	卡证_临时身份证_正
26	卡证_社保卡
27	卡证_身份证_副
28	卡证_身份证_正
29	卡证_士兵证
30	卡证_事业单位法人证
31	卡证_税务登记证
32	卡证_台湾返乡证_副
33	卡证_台湾返乡证_正
34	卡证_台湾身份证
35	卡证_台湾通行证_副
36	卡证_台湾通行证_正
37	卡证_统一社会信用代码证
38	卡证_文职干部证
39	卡证_香港居民身份证
40	卡证_香港永久性居民身份证
41	卡证_烟草专卖许可证
42	卡证_银行卡
43	卡证_营业执照
44	卡证_组织机构代码证
45	电子身份证
46	出生证明
47	安全生产许可证
48	二建证书
49	毕业证书
50	港澳台国外学历学位认证书
51	餐饮服务许可证
52	产品合格证
53	动物检疫合格证明
54	动物检疫合格证明_电子票
55	财务_POS小票

56	财务_餐费购物水单
57	财务_车辆通行费
58	财务_出租车票
59	财务_定额发票
60	财务_二手购车发票
61	财务_购车发票
62	财务_火车票
63	财务_机票行程单
64	财务_卷票
65	财务_轮船票
66	财务_汽车票
67	财务_通用机打发票
68	财务_网约车行程单
69	财务_限额发票
70	财务_增值税发票
71	财务_增值税发票_全电发票
72	财务_支付凭据截图
73	财务_住宿会议费水单
74	财务_行程信息提示单
75	财务_登机牌
76	报销单封面
77	医疗_CT检查报告
78	医疗_MRI检查报告
79	医疗_PET-CT检查报告
80	医疗_X线检查报告
81	医疗_病理报告
82	医疗_病历记录
83	医疗_超声检查报告
84	医疗_出院记录
85	医疗_出院小结
86	医疗_出院诊断证明
87	医疗_大便常规检查
88	医疗_骨髓穿刺检查
89	医疗_护理记录
90	医疗_甲状腺_肿瘤标志物
91	医疗_介入检查
92	医疗_救护车费票据
93	医疗_临时医嘱单
94	医疗_门诊病历
95	医疗_门诊处方笺

95	医疗_门诊处方笺
96	医疗_门诊发票
97	医疗_门诊费用明细_大票
98	医疗_门诊费用明细_小票
99	医疗_门诊挂号单
100	医疗_门诊收据
101	医疗_脑电图
102	医疗_内窥镜检查报告
103	医疗_尿常规检查
104	医疗_农合结算单
105	医疗_其他化验检查
106	医疗_其他检查
107	医疗_入院记录_住院记录
108	医疗_伤残证明
109	医疗_视力检查
110	医疗_手术记录
111	医疗_听力检查
112	医疗_心电图
113	医疗_血常规检查
114	医疗_血凝检查
115	医疗_血生化检查
116	医疗_血液透析单
117	医疗_羊水穿刺检查
118	医疗_医保结算单
119	医疗_医院结账单
120	医疗_乙肝五项
121	医疗_长期医嘱单
122	医疗_肿瘤穿刺检查
123	医疗_住院病案首页
124	医疗_住院病案首页附页
125	医疗_住院发票
126	医疗_住院费用明细
127	医疗_住院收据
128	医疗_住院证
129	医疗机构执业许可证
130	医疗_出院通知
131	医疗出院证
132	医疗器械经营许可证
133	保险_第三方分割单
134	保险_法定受益人确认表

135	保险_法定受益人约定书
136	保险_机动车交强险保单
137	保险_理赔申请书
138	保险_理赔须知
139	金融_银行承兑汇票
140	金融_银行回单
141	存折
142	车管所_安全技术检验合格证明
143	车管所_报废机动车回收证明
144	车管所_发动机编号铭牌照片
145	车管所_机动车查验记录表
146	车管所_机动车登记_注册
147	车管所_机动车登记_转移
148	车管所_机动车驾驶证申请表
149	车管所_机动车使用性质证明
150	车管所_机动车注册_转移_注销_登记_转入申请表
151	车管所_进口凭证
152	车管所_申请表_机动车牌证申请表
153	道路运输经营许可证
154	车辆电子信息单
155	车辆识别代码
156	车辆委托书
157	机动车信息查询结果单
158	交通_磅单
159	交通意外事故死亡证明
160	交通意外事故证明
161	进口车辆电子信息
162	进口机动车辆随车检验单
163	房地产经纪从业人员信息卡_经纪人
164	房地产经纪从业人员信息卡_经纪人协理
165	房地产经纪从业人员信息卡_业务员
166	报关单
167	海关进口关税
168	海关进口增值税
169	海关特别关税
170	彩票
171	法院判决书
172	公安司法等部门公布的死亡名单
173	公证书
174	关系声明

175	关系证明
176	国税清单
177	户口注销证明
178	火化证明
179	计算机软件著作权登记申请表
180	建筑企业资质证书
181	健康码
182	结业证书
183	居住证
184	开户许可证
185	民办学校办学许可证
186	名片
187	内河船舶适航证书
188	农药经营许可证
189	前往港澳通行证
190	入境货物检验检疫证明
191	入账清单
192	食品经营许可证
193	手写发票
194	手续合格通知书
195	输血费票据
196	司法鉴定书
197	司法鉴定许可证
198	死亡医学证明书_居民死亡殡葬证
199	特种设备安装改造维修告知书
200	特种设备使用登记表
201	特种设备型式试验证书
202	体检报告首页
203	调解书
204	完税证明
205	完税证明_电子版
206	危险化学品经营许可证
207	卫生证书
208	物流_道路运输证
209	物流_快递单
210	学籍在线验证报告
211	学历证书_移动版
212	学历证书电子注册备案表
213	学位证书
214	学位证书_移动版

214	学位证书_移动版
215	验证码
216	药品经营许可证
217	因公往来港澳通行证
218	执业药师证
219	专利证书
220	准考证
221	仪器仪表盘
222	设备铭牌

错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
 "log_id": "1803053124329552748",
 "error_code": 110,
 "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	未知错误，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如"-"，可手动输入重试
4	Open api request limit reached	集群超限额，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
6	No permission to access data	无接口调用权限，创建应用时未勾选相关文字识别接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口后重新调用，也可使用权限额度诊断工具完成自助诊断
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	免费测试资源使用完毕，每天请求量超限额，已支持计费的接口，您可以在控制台文字识别服务选择购买相关接口的次数包或开通按量后付费；邀测和未支持计费的接口，您可以在控制台提交工单申请提升限额，也可先使用权限额度诊断工具完成自助诊断
18	Open api qps request limit reached	QPS超限额，免费额度并发限制为2QPS，开通按量后付费或购买次数包后并发限制为10QPS，如您需要更多的并发量，可以选择购买QPS叠加包；邀测和未支持计费的接口，您可以在控制台提交工单申请提升限额，也可先使用权限额度诊断工具完成自助诊断
19	Open api total request limit reached	请求总量超限额，已支持计费的接口，您可以在控制台文字识别服务选择购买相关接口的次数包或开通按量后付费；邀测和未支持计费的接口，您可以在控制台提交工单申请提升限额

100	Invalid parameter	无效的access_token参数，token拉取失败，您可以参考“Access Token获取”文档重新获取
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，请注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，请注意需要定期更换，也可以每次请求都拉取新token
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，请根据调用服务的接口文档，查看请求参数image要求，重新上传图片
216205	input oversize	传入的请求体大小错误，现阶段我们支持的请求体最大上限为：base64编码后小于10M，请重新发送请求
216306	Upload file error	上传文件失败，请检查提交请求接口的请求参数
216308	Pdf_file_num exceeds the number of pdf pages	参数pdf_file_num大于PDF文件实际页数
216401	Create task failed	提交请求失败
216402	Query task failed	获取结果失败
216603	Check pdf page num failed	获取PDF文件页数失败，请检查PDF文件以及base64编码
216604	Insufficient available quota	请求总量超限额，您可以购买或申请更多限额
216630	recognize error	识别错误，请再次请求，请确保图片中包含对应卡证票据
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片、上传的银行卡正面图片不完整或模糊
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片、上传的身份证图片不完整或模糊
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
216600	business verify failed	企业核验相关服务请求失败，请再次请求，**仅适用于企业核验相关服务**：企业工商信息查询（标准版/高级版）、企业二/三/四要素核验。如果持续出现此类错误，请在控制台提交工单联系技术支持团队
216601	business verify	企业核验相关服务查询成功，但是无查询结果返回，请再次请求，**仅适用于企业核验相关服务**：企业工商信息查询（标准版/高级版）、企业二/三/四要素核验。如果持续出现此类错误，

	result empty	请在控制台提交工单联系技术支持团队
216602	business verify timeout	企业核验相关服务接口超时，请再次请求，**仅适用于企业核验相关服务**：企业工商信息查询（标准版/高级版）、企业二/三/四要素核验。如果持续出现此类错误，请在控制台提交工单联系技术支持团队
282000	internal error	服务器内部错误，如果您使用的是高精度接口，报这个错误码的原因可能是您上传的图片中文字过多，识别超时导致的，建议您对图片进行切割后再识别，其他情况请再次请求， 如果持续出现此类错误，请在控制台提交工单联系技术支持团队
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282100	image transcode error	图片压缩转码错误
282102	target detect error	未检测到图片中识别目标，请确保图片中包含对应卡证票据，出现此问题的原因一般为：您上传了非卡证图片、图片不完整或模糊
282103	recognize error, failed to match the template	图片目标识别错误，请确保图片中包含对应卡证票据，出现此问题的原因一般为：您上传了非卡证图片、图片不完整或模糊
282110	urls not exit	URL参数不存在，请核对URL后再次提交
282111	url format illegal	URL格式非法，请检查url格式是否符合相应接口的入参要求
282112	url download timeout	url下载超时，请检查url对应的图床/图片无法下载或链路状况不好，或图片大小大于3M，或图片存在防盗链，您可以重新尝试以下，如果多次尝试后仍不行，建议更换图片地址
282113	url response invalid	URL返回无效参数
282114	url size error	URL长度超过1024字节或为0
282134	officialWeb service exception	**仅适用于增值税发票验真接口**，国税局端网络超时（一般因地方税务局升级或系统调整造成，建议您第2日重试，如果持续出现此类错误，请在控制台提交工单联系技术支持团队）
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
282160	driving license backend resource overrun	后端资源超限，**仅适用于行驶证核验接口**，请在控制台提交工单联系技术支持团队
282161	driving license requests too frequently	请求过于频繁，**仅适用于行驶证核验接口**

常见问题

售前咨询

Q：如何测试产品的识别效果？

A：您可以前往百度AI开放平台，在对应产品页面的『功能演示』处，上传需测试的图片进行效果测试。



或者您也可以前往AI能力体验中心，体验产品效果。

Q：官网的产品都不匹配我的需求，应该怎么办？

A：如果您要识别的图片是固定版式的卡证/票据，建议尝试iOCR自定义模板文字识别或智能结构化识别；如果需识别非固定版式的卡证/票据/单据的固定字段，或希望基于数据进行模型的迭代优化，建议尝试使用EasyDL OCR自训练平台； 如果需识别的图片是文本段落，无固定版式，可尝试使用通用文字识别。

> iOCR自定义模板文字识别：针对固定版式的卡证/票据，可自助创建识别模板和分类器，实现图片自动分类并结构化输出识别结果。
> 智能结构化识别：结构化识别各类卡证/票据，无需配置结构化对应关系、无需提取关键词、无需定制开发，直接上传图片即可获得结构化识别信息。
> EasyDL OCR自训练平台：零算法门槛的文字识别模型生产平台，针对卡证/票据/单据，可自训练产出更高精度的结构化识别模型，并持续迭代优化；平台内置百度领先的 OCR 预训练模型，帮助企业/开发者低成本定制专属的 OCR 识别模型。
> 通用文字识别：提供多场景、多语种、高精度的整图文字检测和识别服务，多项ICDAR指标居世界第一，可识别20种语言。

如果以上产品均不是很符合您的需求，您可以填写合作咨询或拨打400-920-8999转1联系我们。

Q：云端服务、离线识别SDK、私有化部署、一体机是什么，应该如何选择？

A：这是四种文字识别产品的使用方式，您可以根据业务需求进行选择：

云端服务：即公有云API，提供各类文字识别的云端Paas服务接口，您可直接调用API或使用SDK对图片中的文字进行识别，适合网络通畅的B端或C端应用场景。

离线识别SDK：集成到移动设备中（Windows、Android、iOS），无需网络即可实现文字识别功能，适合网络环境不佳的各类场景，例如：地下停车场、封闭仓库等。详见离线识别SDK产品介绍。

私有化部署：部署至您的本地服务器，在内网中实现文字识别功能，保障数据私密性，适合银行、政府、公安等数据保密性较强的场景。详见私有化部署产品介绍。

一体机：软硬一体一站式交付，支持多种硬件配置，更有国产化配置可选，开箱即用。您可以填写合作咨询进行申请，我们会安排专人联系您。

如果您不是很明确应该如何选择，您可以填写合作咨询或拨打400-920-8999转1联系我们。

Q：API调用的次数包、按量后付费、QPS叠加包等付费方式，应该如何选择？

A：**次数包**：一次购买，全年使用，适用于调用量可预估的企业。

按量后付费：随开随停，适用于灵活付费的企业。

QPS叠加包：当赠送的QPS不足以满足您的业务需求时，您可以付费扩充QPS。

如果上述方式不能满足您的需求，或预估调用量极大，您可以填写合作咨询或拨打400-920-8999转1联系我们。

Q：文字识别服务的云端API应该如何接入?

A：您可以参考新手操作指引或API调用视频教程进行操作。如果您对操作流程还是存在疑惑，您可以填写提交工单或拨打400-920-8999转1联系我们。

Q：使用文字识别接口，担心所上传图片中的身份、财产信息泄露？

A：百度十分重视用户信息的保护，百度将会采取合理措施保护您（及您的用户）的隐私权。隐私政策详见：公有云API隐私政策、 SDK隐私政策 。

> 遇到其他相关问题，您可以提交工单寻求帮助，或填写合作咨询，或拨打400-920-8999转1，会有专人跟进处理。

产品使用问题

Q: 文字识别的并发量上限是多少？

A: 大部分文字识别接口在未开通付费时提供2QPS额度，开通付费后提升至10QPS，如果您有更高并发需求，可购买QPS叠加包进行扩充。对于已上线但没有价格的产品，暂属于测试状态，当测试额度不足时，您可以提交工单进行申请，您需要在工单提供您的appid、业务场景描述、需要的接口名称和申请的并发量。

Q：如何购买/提升调用次数？

A：已上线计费的接口，您可以直接在控制台购买次数包或开通按量后付费，计费价格参见产品价格；未上线计费的接口，您可以提交工单进行申请，您需要在工单提供您的appid、业务场景描述、需要的接口名称和申请的调用次数。

Q：文字识别对上传的文件格式和大小有哪些要求？

A：**图片格式**：支持 JPG、JPEG、BMP、TIF、WebP 等格式图片及PDF文档，暂不支持 GIF 类型的动图识别。具体以相应接口的API文档中的请求参数为准。

图片大小：一般情况下，图像base64编码后大小必须小于4M，建议不要超过1M；最小边长不小于15 px，最大边长不超过4096 px，建议不要超过1024（编码后大于1M或最大边长超过1024的图像会被等比压缩，建议控制输入图像大小，有助于减少网络传输及接口处理耗时）。但是不同的功能接口，对于图片大小的要求可能不同，具体以相应接口的API文档中的 Image 和 url 参数说明为准。

Q：什么是base64编码，如何提供？

A：图片的base64编码指将一副图片数据编码成一串字符串，各种编程语言均包含Base64编码函数，可直接调用使用。

注：图片base64编码后需去除图片头，如（），并进行urlencode后方可上传。

Q：怎么提高识别的准确率和识别速度？

A：文字识别的准确率跟拍摄光照、背景、清晰度等因素有关。推荐上传JPG图片格式，图片大小建议1M以内。可在图片采集端尽量扩大要识别文字的区域，并保证图片内文字清晰人眼可辨认、倾斜度不得小于30%。同时，适当压缩图片大小，可大幅缩短图片识别时间。

Q：文字识别支持的语言？

A：不同的功能接口，所支持的语言都不同。常见多语言识别接口如下：

通用文字识别（标准版）、通用文字识别（标准含位置版）：支持中文简体、中文繁体、英文、日语、韩语、法语、西班牙语、葡萄牙语、德语、意大利语、俄语。

通用文字识别（高精度版）、通用文字识别（高精度含位置版）：支持中文简体、中文繁体、英文、日语、韩语、法语、西班牙语、葡萄牙语、德语、意大利语、俄语、丹麦语、荷兰语、马来语、瑞典语、印尼语、波兰语、罗马尼亚语、土耳其语、希腊语、匈牙利语。

其他接口（除国内专用卡证票据外）基本均可支持中、英文内容识别。详细参见API文档。

如果您对于产品支持识别的语言有特殊需求，您可以提交合作咨询联系我们。

Q：文字识别是否支持方向旋转/不同朝向的图片文字识别？

A：文字识别大部分能力均已支持图像方向自动校正功能，可对旋转的图片进行正确的识别，也可通过控制参数

『direction』为true/false控制该功能是否开启。如果您在使用中存在部分旋转图片无法正确识别的问题，您可以提交工单告知我们

Q：文字识别有无区分卡证、票据原件和复印件的功能？

A：身份证识别含风险检测功能，可区分身份证原件及复印件。详情参考API文档。

如果您对其他文字识别服务有区分原件、复印件功能的需求，您可以提交合作咨询告知我们。

Q：文字识别有无区分卡证、票据真伪的功能？

A：身份证识别具备翻拍、PS、复印件告警功能，您也可以使用人脸核身的公安验证接口，用于校验姓名和身份证号的真实性和一致性。增值税发票验真可快速对接国家税务机关发票查验平台，支持全部9类增值税发票的信息核验。

如果您对其他文字识别服务有区分卡证、票据真伪功能的需求，您可以提交合作咨询告知我们。

Q：文字识别是否可以批量识别吗？

A：暂不支持，单次调用仅可识别单张图片，但您可在QPS允许范围进行多线程调用。

Q：可以识别PDF、Word、Excel等格式文件吗？

A：支持 JPG、JPEG、BMP、TIF、WebP 等格式图片及PDF文档，暂不支持 Word、Excel等格式，如果您对这部分文本格式存在识别需求，可以先将其转为图片格式或PDF文档后再进行上传。

Q：识别结果可以转化为Word或者TXT吗？

A：OCR提取之后返回的结果是JSON格式，需要您通过进行业务处理将结果保存为Word或者TXT格式。

Q：可否在文字识别的应用界面添加扫描框？

A：百度仅提供文字识别API接口，应用界面可根据您的需求进行自行开发。

Q：是否支持识别验证码？

A：对验证码进行识别涉及网络安全问题，百度不提供验证码识别专项服务。

Q: 文字识别的响应速度是多少？

A: 一般在1s内，识别时间会受图片大小、字数多少影响，但最长不超过7s，一旦超过将自动返回『超时』错误，相应调用不计费。

注：但由于数据传输的网络情况不同，且为百度不可控范围，实际您感受到的响应时间为百度模型识别时间+数据来往传输时间，如出现大量耗时过长情况，请您排查服务器网络状况，适当扩充带宽或对图片进行压缩后上传，如有需要也可提交工单联系我们。

Q：在国外，也可以在线调用文字识别的接口吗？

A：可以的，但延时会大些。

Q：为什么文字识别结果不准确？

A：有以下几个原因：

（1）图片尺寸过小，图片尺寸小于15px，无法进行识别。

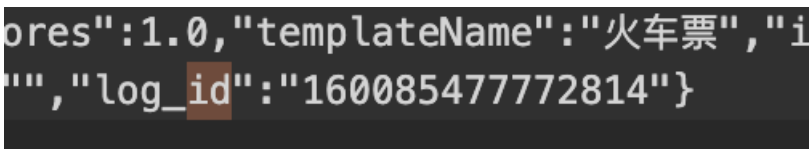
（2）图片画质太差，例如图片过暗，文字内容不可辨识。

（3）文字内容存在水印、印章、褶皱等遮挡。

（4）图片样式与接口支持类型不符。例如，身份证识别只支持识别二代居民身份证，不支持识别护照、银行卡等。

（5）如果有返回错误码，请参考错误码排查问题。

如果仍然无法确定原因和解决问题，您可以提交工单，您需要在工单提供误识别的原图及返回的log_id（log_id为调用接口返回时，在返回结果中出现的一长串数字）。举例如下：



Q：调用文字识别API服务失败时，应该怎么处理？

A：排查原因：

（1）根据API调用返回结果或错误码查找原因。

（2）检查API调用方法是否正确（您可以参考AI接入指南或OCR调用教学视频进行操作）。

如果仍然无法确定原因和解决问题，您可以提交工单联系我们。

> 遇到其他相关问题，您可提交工单寻求帮助，会有专人跟进处理。

计费问题

Q：如何购买云端API服务?

A：您可以参考「如何购买」技术文档或视频演示，在控制台进行开通、充值、购买等操作。

Q：如何获知产品价格？

A：您可以参考产品价格进行选购。对于已上线但没有价格的产品，暂属于免费测试状态，您可以直接进行测试。当无测试权限时，请在相应产品页面申请试用。当测试额度不足时，您可以提交工单进行提额申请，您需要在工单提供您的appid、业务场景描述、需要的接口名称和申请的日调用量。

Q：希望正式付费使用某产品，但目前仍不支持购买，什么时候会上线计费？

A：我们在持续提升产品成熟度，当成熟度满足商用要求时，产品才会上线计费。当产品正式商用时，我们会在官网资讯发布公告；您也可以加入文字识别QQ群（三群：1055623827），快速获取产品更新、功能升级、优惠活动等资讯。

如果您希望获知产品预计上线计费的时间，您可以提交工单联系我们。

Q：次数包用完以后将如何进行收费？

A：如果您已购买新的次数包，则自动转入新次数包进行扣减。如果您未购买次数包或者次数包已耗尽，将自动转入按量后付费结算。具体价格参见产品价格。

Q：免费测试资源的消耗是账号维度，还是应用维度进行计算？

A：免费测试资源是针对百度账号的，该账号内所有的应用共用。例如，1个账号的免费调用资源为500次，该账号下的10个应用的免费调用资源则共享500次。

Q：免费额度和次数包快用完时是否有通知？

A：当免费配额用尽时，无消息提醒，如果您没有开通服务/购买次数包，服务将自动停止。如果您购买了次数包，当次数包用量仅剩20%或即将到期时，会向您账号预留的手机号及邮箱发送提醒，请您关注。您可以在控制台，根据如何购买的指引，进行开通、充值、购买等操作。

Q：欠费是如何产生的？如何预防？

A：后付费服务的产品由于是先使用服务后扣费，扣费时账户余额不足以支付账单金额时即产生欠费，例如：使用身份证识别按量后付费服务，当发现您的账户余额不足以支付调用所产生的费用时，即欠费。欠费时系统会发送欠费通知。

> 为保证服务的正常使用，请参考如何购买，提前进行账户充值，保证账户中有足够的现金金额。

Q：欠费后该如何处理？

A：欠费后，系统会立即停止相关服务，为避免影响服务与业务的正常运行，请参考如何购买，及时在账户充值进行现金充值，充值金额将优先抵扣欠款。

Q：调用失败、识别不成功是否扣费？

A：调用失败，返回错误码时，不会对调用进行扣费。您可以根据错误码提示，及时进行原因排查。

Q: 文字识别服务升级后，原次数包未用完，如何计费？

A：升级不影响您的正常使用及计费。文字识别各能力均会不断升级，以提升识别准确率及性能。

Q: 购买QPS叠加包后，QPS会变成多少?

A：根据您账户当前已有的QPS，购买QPS叠加包后的QPS值是您账户当前的QPS数量加上购买的QPS数量。

Q：同时识别多张卡证/票据，如何收费？

A：一般情况下，按照调用接口的次数收费。但调用混贴识别类接口，如智能财务票据识别和iOCR财会版中的混贴票据识别，则按照识别对象的数量计费。例如，一张粘贴单上粘贴 4 张火车票，则计费次数计作 4 次。具体计费说明可参考各接口计费文档。

Q: 为什么两次账单中的调用次数基本相同，费用却差距很大？

A: 文字识别大部分接口采用分段式阶梯计价，因月度调用累积量不同而单价不同。两次小时级账单虽然显示调用次数基本相似，但是月度调用累积量可能已经落入不同的阶梯区间，这样导致调用次数的单价不同，因此费用差距比较大。这种差异多发生在月末与月初账单对比，或是跨价格阶梯的账单对比中。

Q: 多个应用均调用同个接口识别，可以部分应用作为测试，只使用免费配额，另一部分应用开通付费吗？

A: 不能。产品计费是按照账户维度进行计算的，单账户下的所有应用进行累积计费，暂时无法实现只有部分应用开通付费。如开通了银行卡识别的付费，那么当相关应用调用银行卡识别的日调用量总计超过了500次，则开始计费，不区分应用。

Q：次数包买错了可以退款或者更换产品类别吗？

A：次数包购买后，不支持更换、退款或延期。请付款前确认所需购买的次数包类型和数量，并保证在一年有效期内使用完毕。如果次数包一年内没有用完，剩余的调用次数在次年将失效。

Q：文字识别服务购买后，如何开具发票用于企业报销？

A：开通付费后，您可以进入控制台，再进入财务中心，在发票管理进行发票申请、发票信息管理等。更多发票相关问题可点击查看详情。

Q：完成实名认证后，文字识别免费测试资源，系统何时发放？

A：完成实名认证后，需再次登录并进入文字识别控制台 ，免费测试资源会在您进入控制台后约10分钟内发放，若领取接口长时间未在「资源列表」上生效显示，请提交工单咨询。

> 遇到其他相关问题，您可提交工单寻求帮助，会有专人跟进处理。

HTTP-SDK文档

Python语言

简介

Hi，您好，欢迎使用百度文字识别服务。

本文档主要针对Python开发者，描述百度文字识别接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 在百度智能云控制台内提交工单，咨询问题类型请选择**人工智能服务**；
- 如有疑问，进入AI社区交流：http://ai.baidu.com/forum/topic/list/164

接口能力

接口名称	接口能力简要描述
通用文字识别	识别图片中的文字信息
通用文字识别（高精度版）	更高精度地识别图片中的文字信息
通用文字识别（含位置信息版）	识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（高精度含位置版）	更高精度地识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（含生僻字版）	识别图片中的文字信息（包含对常见字和生僻字的识别）
网络图片文字识别	识别一些网络上背景复杂，特殊字体的文字
网络图片文字识别（含位置版）	识别网络图片中的文字内容（包含文字区域的坐标信息）

身份证识别	识别身份证正反面的文字信息
银行卡识别	识别银行卡的卡号并返回发卡行和卡片性质信息
驾驶证识别	识别机动车驾驶证所有关键字段
行驶证识别	识别机动车行驶证所有关键字段
车牌识别	识别中国大陆各类机动车车牌信息
营业执照识别	对营业执照进行识别
表格文字识别	自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容
通用票据识别	对各类票据图片（医疗票据，保险保单等）进行文字识别，并返回文字在图片中的位置信息
增值税发票识别	对增值税发票进行文字识别，并结构化返回字段信息，支持增值税专票、普票、电子发票
出租车票识别	针对全国各大城市出租车票的发票号码、发票代码、车号、日期、时间、金额等进行结构化识别
VIN码识别	对车辆车架、挡风玻璃上的VIN码进行识别
火车票识别	支持对大陆火车票的车票号、始发站、目的站、车次、日期、票价、席别、姓名进行结构化识别
飞机行程单识别	支持对飞机行程单的24个字段进行结构化识别
二维码识别	对图片中的二维码、条形码进行检测和识别，返回存储的文字信息
数字识别	识别图片中的数字，适用于手机号提取、快递单号提取、充值号码提取等场景
手写文字识别	支持对图片中的手写中文、手写数字进行检测和识别
护照识别	支持对中国大陆护照个人资料页所有15个字段进行结构化识别
户口本识别	对出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码字段进行识别
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果
通用机打发票	支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别
机动车销售发票	支持对机动车销售发票的26个关键字段进行结构化识别
车辆合格证	支持对车辆合格证的23个关键字段进行结构化识别
通用机打发票	对国家/地方税务局发行的横/竖版通用机打发票进行结构化识别
护照识别	支持对中国大陆护照个人资料页所有11个字段进行结构化识别
医疗费用明细识别	支持识别全国医疗费用明细识别
网约车行程单识别	对国家/地方税务局发行的横/对各大主要服务商的网约车行程单进行结构化识别
磅单识别	结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单
仪器仪表盘读数识别	适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号
自定义模板文字识别	针对固定版式卡证票据提供的 OCR 定制化产品，可由用户自助创建识别模板和分类器，实现对任意版式卡证票据进行自动分类并结构化输出识别结果
医疗费用明细识别	支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含项目类型、项目名称、单价、数量、规格、金额
办公文档识别	可对办公类文档的版面进行分析，输出图、表、标题、文本、目录、栏、页眉、页脚、页码和脚注的位置，并输出分版块内容的OCR识别结果

印章识别	检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别
机动车登记证书识别	对机动车登记证书的编号、机动车所有人、登记机关、车辆类型、发证机关章等15个关键字段进行结构化识别
智能财务票据识别	对增值税发票、卷票、火车票、出租车票、机票行程单等13类票据混贴的图片进行切分识别
增值税发票验真	支持9种增值税发票的真伪及字段信息准确性校验，包括增值税专票、电子专票、普票、电子普票、卷票、通行费增值税电子普票、货运专票、机动车销售发票、二手车销售发票，支持返回票面的全部信息
医疗发票识别	支持识别全国各地门诊/住院发票的 业务流水号、发票号、住院号、门诊号、病例号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付等关键字段。支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数
门脸文字识别	识别图片中的门脸文字信息，自动过滤非门脸文字内容，接口返回门脸名称、描述文字和置信度
车辆证照混贴识别	对机动车行驶证主页及副页、驾驶证主页及副页在同一张图片上的场景进行结构化识别
公式识别	对试卷中的数学公式及题目内容进行识别
图文转换器	可识别图片/PDF文档版面布局，提取文字内容，并转换为保留原文档版式的Word、Excel文档，方便二次编辑和复制，可支持含表格、印章、水印、手写等内容的文档

版本更新记录

上线日期	版本号	更新内容
2022.03.22	4.16.2	OCR 商用和公测接口新增支持 url 传图方式，并支持 pdf 文件传入识别
2022.03.22	4.16.1	新增： 智能财务票据识别，增值税发票验真，医疗发票识别，门脸文字识别，车辆证照混贴识别，公式识别
2022.03.15	4.15.16	新增： 办公文档识别，印章识别，机动车登记证书识别
2021.12.14	4.15.14	新增： 网约车行程单识别，磅单识别，医疗明细识别
2021.05.26	4.15.12	新增： 二维码，行程单，机动车销售发票，车辆合格证，试卷分析与识别，手写，护照，户口本，通用机打
2021.01.28	4.15.4	新增：增值税发票，出租车票，VIN码，火车票，数字识别
2020.08.06	4.15.1	新增：文档版面分析与识别，仪器仪表盘读数识别， 网络图片文字识别
2018.04.09	2.2.2	新增：表格识别同步接口
2018.01.12	2.1.0	新增：自定义OCR识别
2017.12.22	2.0.0	SDK代码重构
2017.08.25	1.6.4	OCR 新增营业执照识别
2017.05.11	1.0.0	OCR服务上线

快速入门

安装OCR Python SDK

OCR Python SDK目录结构

```
 ├─── README.md
├─── aip                  //SDK目录
│    ├─── __init__.py      //导出类
│    ├─── base.py          //aip基类
│    ├─── http.py          //http请求
│    └─── ocr.py //OCR
└─── setup.py             //setuptools安装
```

支持Python版本：2.7.+ ,3.+

安装使用Python SDK有如下方式：

- 如果已安装pip，执行`pip install baidu-aip`即可。
- 如果已安装setuptools，执行`python setup.py install`即可。

新建AipOcr

AipOcr是OCR的Python SDK客户端，为使用OCR的开发人员提供了一系列的交互方法。

参考如下代码新建一个AipOcr：

```
 from aip import AipOcr

""" 你的 APPID AK SK """
APP_ID = '你的 App ID'
API_KEY = '你的 Api Key'
SECRET_KEY = '你的 Secret Key'

client = AipOcr(APP_ID, API_KEY, SECRET_KEY)
```

在上面代码中，常量`APP_ID`可在百度智能云控制台应用列表中创建应用获得，常量`API_KEY`与`SECRET_KEY`在创建完毕应用后均可获得，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的**应用列表**中查看。

注意：如您以前是百度智能云的老用户，其中`API_KEY`对应百度智能云的“Access Key ID”，`SECRET_KEY`对应百度智能云的“Access Key Secret”。

配置AipOcr

如果用户需要配置AipOcr的网络请求参数(一般不需要配置)，可以在构造AipOcr之后调用接口设置参数，目前只支持以下参数：

接口	说明
setConnectionTimeoutInMillis	建立连接的超时时间（单位：毫秒
setSocketTimeoutInMillis	通过打开的连接传输数据的超时时间（单位：毫秒）

接口说明

通用文字识别（标准版）

用户向服务请求识别某张图中的所有文字。

```
    """ 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用通用文字识别（标准版）
res_image = client.basicGeneral(image)
res_url = client.basicGeneralUrl(url)
res_pdf = client.basicGeneralPdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options = {}
options["language_type"] = "CHN_ENG"
options["detect_direction"] = "true"
options["detect_language"] = "true"
options["probability"] = "true"
res_image = client.basicGeneral(image, options)
res_url = client.basicGeneralUrl(url, options)
res_pdf = client.basicGeneralPdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

通用文字识别 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
language_type	否	string	CHN_ENG ENG JAP KOR FRE SPA POR GER ITA RUS	识别语言类型，默认为CHN_ENG 可选值包括： - CHN_ENG：中英文混合 - ENG：英文 - JAP：日语 - KOR：韩语 - FRE：法语 - SPA：西班牙语 - POR：葡萄牙语 - GER：德语 - ITA：意大利语 - RUS：俄语
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true/false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
paragraph	否	string	true/false	是否输出段落信息
probability	否	string	true/false	是否返回识别结果中每一行的置信度

通用文字识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	识别结果数组
+ words	否	string	识别结果字符串
+ probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值，当 probability=true 时返回该字段
paragraphs_result	否	array[]	段落检测结果，当 paragraph=true 时返回该字段
+ words_result_idx	否	array[]	一个段落包含的行序号，当 paragraph=true 时返回该字段
language	否	int32	当 detect_language=true 时返回该字段
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

通用文字识别 返回示例

```
 {
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青島啤酒"}
   ]
}
```

通用文字识别（高精度版）

用户向服务请求识别某张图中的所有文字，相对于通用文字识别该产品精度更高，但是识别耗时会稍长。

```
    """ 读取文件 """
    def get_file_content(filePath):
        with open(filePath, "rb") as fp:
            return fp.read()

    image = get_file_content('文件路径')
    url = "https://www.x.com/sample.jpg"
    pdf_file = get_file_content('文件路径')

# 调用通用文字识别（高精度版）
    res_image = client.basicAccurate(image)
    res_url = client.basicAccurateUrl(url)
    res_pdf = client.basicAccuratePdf(pdf_file)
    print(res_image)
    print(res_url)
    print(res_pdf)

# 如果有可选参数
    options = {}
    options["detect_direction"] = "true"
    options["probability"] = "true"
    res_image = client.basicAccurate(image, options)
    res_url = client.basicAccurateUrl(url, options)
    res_pdf = client.basicAccuratePdf(pdf_file, options)
    print(res_image)
    print(res_url)
    print(res_pdf)
```

通用文字识别（高精度版） 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
			auto_detect CHN_ENG ENG JAP KOR FRE SPA	识别语言类型，默认为CHN_ENG 可选值包括： - auto_detect：自动检测语言，并识别 - CHN_ENG：中英文混合 - ENG：英文 - JAP：日语 - KOR：韩语 - FRE：法语 - SPA：西班牙语 - POR：葡萄牙语

language_type	否	string	SPA POR GER ITA RUS DAN DUT MAL SWE IND POL ROM TUR GRE HUN	- GER：德语 - ITA：意大利语 - RUS：俄语 - DAN：丹麦语 - DUT：荷兰语 - MAL：马来语 - SWE：瑞典语 - IND：印尼语 - POL：波兰语 - ROM：罗马尼亚语 - TUR：土耳其语 - GRE：希腊语 - HUN：匈牙利语 - THA：泰语 - VIE：越南语 - ARA：阿拉伯语 - HIN：印地语
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向
paragraph	否	string	true/false	是否输出段落信息
probability	否	string	true/false	是否返回识别结果中每一行的置信度

通用文字识别（高精度版） 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result	是	array[]	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	否	string	识别结果字符串
paragraphs_result	否	array[]	段落检测结果，当 paragraph=true 时返回该字段
+ words_result_idx	否	array[]	一个段落包含的行序号，当 paragraph=true 时返回该字段
+ probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值，当 probability=true 时返回该字段
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

通用文字识别（高精度版） 返回示例

参考通用文字识别（标准版）返回示例

通用文字识别（标准含位置版）

用户向服务请求识别某张图中的所有文字，并返回文字在图中的位置信息。

```
    """ 读取文件 """
    def get_file_content(filePath):
        with open(filePath, "rb") as fp:
            return fp.read()

    image = get_file_content('文件路径')
    url = "https://www.x.com/sample.jpg"
    pdf_file = get_file_content('文件路径')

# 调用通用文字识别（标准含位置信息版）
    res_image = client.general(image)
    res_url = client.generalUrl(url)
    res_pdf = client.generalPdf(pdf_file)
    print(res_image)
    print(res_url)
    print(res_pdf)

# 如果有可选参数
    options = {}
    options["recognize_granularity"] = "big"
    options["language_type"] = "CHN_ENG"
    options["detect_direction"] = "true"
    options["detect_language"] = "true"
    options["vertexes_location"] = "true"
    options["probability"] = "true"
    res_image = client.general(image, options)
    res_url = client.generalUrl(url, options)
    res_pdf = client.generalPdf(pdf_file, options)
    print(res_image)
    print(res_url)
    print(res_pdf)
```

通用文字识别（标准含位置版） 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
language_type	否	string	CHN_ENG ENG JAP KOR FRE SPA POR GER ITA RUS	识别语言类型，默认为CHN_ENG 可选值包括： - CHN_ENG：中英文混合 - ENG：英文 - JAP：日语 - KOR：韩语 - FRE：法语 - SPA：西班牙语 - POR：葡萄牙语 - GER：德语 - ITA：意大利语 - RUS：俄语
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true/false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
paragraph	否	string	true/false	是否输出段落信息
vertexes_location	否	string	true/false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	string	true/false	是否返回识别结果中每一行的置信度

通用文字识别（标准含位置版） 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	否	string	识别结果字符串
+ location	是	array[]	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
language	否	int32	当 detect_language=true 时返回该字段
paragraphs_result	否	array[]	段落检测结果，当 paragraph=true 时返回该字段
+ words_result_idx	否	array[]	一个段落包含的行序号，当 paragraph=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时返回该字段
++ char	否	string	单字符识别结果，当 recognize_granularity=small 时返回该字段
++ location	否	array[]	位置数组（坐标0点为左上角），当 recognize_granularity=small 时返回该字段
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标，当 recognize_granularity=small 时返回该字段
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标，当 recognize_granularity=small 时返回该字段
+++ width	否	uint32	表示定位定位位置的长方形的宽度，当 recognize_granularity=small 时返回该字段
+++ height	否	uint32	表示位置的长方形的高度，当 recognize_granularity=small 时返回该字段
+ probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值，当 probability=true 时返回该字段
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

通用文字识别（含位置信息版） 返回示例

```
{
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
    "location": {
      "left": 35,
      "top": 53,
      "width": 193,
      "height": 109
    },
    "words": "感动",
    "chars": [    //recognize_granularity=small时存在
      {
        "location": {
          "left": 56,
          "top": 65,
          "width": 69,
          "height": 88
        },
        "char": "感"
      },
      {
        "location": {
          "left": 140,
          "top": 65,
          "width": 70,
          "height": 88
        },
        "char": "动"
      }
    ]
  }
  ...
]
}
```

通用文字识别（高精度含位置版）

用户向服务请求识别某张图中的所有文字，并返回文字在图片中的坐标信息，相对于通用文字识别（含位置信息版）该产品精度更高，但是识别耗时会稍长。

```
    """ 读取文件 """
    def get_file_content(filePath):
        with open(filePath, "rb") as fp:
            return fp.read()

    image = get_file_content('文件路径')
    url = "https://www.x.com/sample.jpg"
    pdf_file = get_file_content('文件路径')

# 调用通用文字识别（高精度含位置版）
    res_image = client.accurate(image)
    res_url = client.accurateUrl(url)
    res_pdf = client.accuratePdf(pdf_file)
    print(res_image)
    print(res_url)
    print(res_pdf)

# 如果有可选参数
    options = {}
    options["recognize_granularity"] = "big"
    options["detect_direction"] = "true"
    options["vertexes_location"] = "true"
    options["probability"] = "true"
    res_image = client.accurate(image, options)
    res_url = client.accurateUrl(url, options)
    res_pdf = client.accuratePdf(pdf_file, options)
    print(res_image)
    print(res_url)
    print(res_pdf)
```

通用文字识别（高精度含位置版） 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
			auto_detect CHN_ENG ENG	识别语言类型，默认为CHN_ENG 可选值包括： - auto_detect：自动检测语言，并识别 - CHN_ENG：中英文混合 - ENG：英文 - JAP：日语 - KOR：韩语

language_type	否	string	JAP KOR FRE SPA POR GER ITA RUS DAN DUT MAL SWE IND POL ROM TUR GRE HUN	- FRE：法语 - SPA：西班牙语 - POR：葡萄牙语 - GER：德语 - ITA：意大利语 - RUS：俄语 - DAN：丹麦语 - DUT：荷兰语 - MAL：马来语 - SWE：瑞典语 - IND：印尼语 - POL：波兰语 - ROM：罗马尼亚语 - TUR：土耳其语 - GRE：希腊语 - HUN：匈牙利语 - THA：泰语 - VIE：越南语 - ARA：阿拉伯语 - HIN：印地语
eng_granularity	否	string	word/letter	表示识别语言类型为「中英文（CHN_ENG）」的情况下，英文的单字符结果是按照单词（word）维度输出还是字母（letter）维度输出，当recognize_granularity=small 时生效
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vertexes_location	否	string	true/false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
paragraph	否	string	true/false	是否输出段落信息
probability	否	string	true/false	是否返回识别结果中每一行的置信度

通用文字识别（高精度含位置版） 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result	是	array[]	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	否	string	识别结果字符串
+ location	是	array[]	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
paragraphs_result	否	array[]	段落检测结果，当 paragraph=true 时返回该字段
+ words_result_idx	否	array[]	一个段落包含的行序号，当 paragraph=true 时返回该字段
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时返回该字段
++ char	否	string	单字符识别结果，当 recognize_granularity=small 时返回该字段
++ location	否	array[]	位置数组（坐标0点为左上角），当 recognize_granularity=small 时返回该字段
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标，当 recognize_granularity=small 时返回该字段
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标，当 recognize_granularity=small 时返回该字段
+++ width	否	uint32	表示定位定位位置的长方形的宽度，当 recognize_granularity=small 时返回该字段
+++ height	否	uint32	表示定位位置的长方形的高度，当 recognize_granularity=small 时返回该字段
+ probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值，当 probability=true 时返回该字段
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

通用文字识别（含位置高精度版） 返回示例

```
{
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
    "location": {
      "left": 35,
      "top": 53,
      "width": 193,
      "height": 109
    },
    "words": "感动",
    "chars": [    //recognize_granularity=small时存在
      {
        "location": {
          "left": 56,
          "top": 65,
          "width": 69,
          "height": 88
        },
        "char": "感"
      },
      {
        "location": {
          "left": 140,
          "top": 65,
          "width": 70,
          "height": 88
        },
        "char": "动"
      }
    ]
  }
  ...
]
}
```

通用文字识别（含生僻字版）

【该服务已停止更新，如需更好的识别效果请使用通用文字识别（高精度版 / 高精度含位置版），此两项服务已扩充字库，可支持生僻字识别】字库范围更大，支持对图片中的生僻字进行识别

通用文字识别（含生僻字版） 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含生僻字版） 返回示例

```
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
  [
    {"words": " TSINGTAO"},
    {"words": "青島啤酒"}
  ]
}
```

网络图片文字识别

用户向服务请求识别一些网络上背景复杂，特殊字体的文字。

```
    """ 读取文件 """
    def get_file_content(filePath):
        with open(filePath, "rb") as fp:
            return fp.read()

    image = get_file_content('文件路径')
    url = "https://www.x.com/sample.jpg"
    pdf_file = get_file_content('文件路径')

    # 调用网络图片文字识别
    res_image = client.webImage(image)
    res_url = client.webImageUrl(url)
    res_pdf = client.webImagePdf(pdf_file)
    print(res_image)
    print(res_url)
    print(res_pdf)

# 如果有可选参数
    options = {}
    options["detect_direction"] = "true"
    options["detect_language"] = "true"
    res_image = client.webImage(image, options)
    res_url = client.webImageUrl(url, options)
    res_pdf = client.webImagePdf(pdf_file, options)
    print(res_image)
    print(res_url)
    print(res_pdf)
```

网络图片文字识别 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true/false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

网络图片文字识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

网络图片文字识别 返回示例

```
 {
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青島啤酒"}
   ]
}
```

身份证识别

用户向服务请求识别身份证，身份证识别包括正面和背面。

```python
    """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"

idCardSide = "back"

# 调用身份证识别
res_image = client.idcard(image, idCardSide)
res_url = client.idcardUrl(url, idCardSide)
print(res_image)
print(res_url)

# 如果有可选参数
options = {}
options["detect_direction"] = "true"
options["detect_risk"] = "false"
res_image = client.idcard(image, idCardSide, options)
res_url = client.idcardUrl(url, idCardSide, options)
print(res_image)
print(res_url)
```

身份证识别 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
id_card_side	是	string	front/back	-front：身份证含照片的一面 -back：身份证带国徽的一面 自动检测身份证正反面，如果传参指定方向与图片相反，支持正常识别，返回参数image_status字段为"reversed_side"
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向
detect_risk	否	string	true/false	是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)检测功能，默认不开启，即：false。 - true：开启，请查看返回参数risk_type； - false：不开启
detect_quality	否	string	true/false	是否开启身份证质量类型(边框/四角不完整、头像或关键字段被遮挡/马赛克)检测功能，默认不开启，即：false。 - true：开启，请查看返回参数card_quality； - false：不开启
detect_photo	否	string	true/false	是否检测头像内容，默认不检测。可选值：true-检测头像并返回头像的 base64 编码及位置信息
detect_card	否	string	true/false	是否检测身份证进行裁剪，默认不检测。可选值：true-检测身份证并返回证照的 base64 编码及位置信息

身份证识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
image_status	是	string	normal-识别正常 reversed_side-身份证正反面颠倒 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 other_type_card-其他类型证照 over_exposure-身份证关键字段反光或过曝 over_dark-身份证欠曝（亮度过低） unknown-未知状态
			输入参数 detect_risk = true 时，则返回该字段识别身份证**风险类型**: normal-正常身份证；

risk_type	否	string	normal-正常身份证； copy-复印件； temporary-临时身份证； screen-翻拍； unknown-其他未知情况
edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
card_quality	否	object	输入参数 detect_quality = true 时，则返回该字段识别身份证**质量类型**: IsClear - 是否清晰； IsComplete - 是否边框/四角完整； IsNoCover - 是否头像、关键字段无遮挡/马赛克。 及对应的概率：IsComplete_propobility、IsNoCover_propobility、IsClear_propobility，值在0-1之间，值越大表示图像质量越好。 **默认阈值：**当 IsComplete_propobility 超过0.5时，IsComplete返回1，低于0.5，则返回0。 IsNoCover_propobility、IsClear_propobility 同上
log_id	是	uint64	唯一的log id，用于问题定位
photo	否	string	当请求参数 detect_photo = true时返回，头像切图的 base64 编码（无编码头，需自行处理）
photo_location	否	object	当请求参数 detect_photo = true时返回，头像的位置信息（坐标0点为左上角）
card_image	否	string	当请求参数 detect_card = true时返回，身份证裁剪切图的 base64 编码（无编码头，需自行处理）
card_location	否	object	当请求参数 detect_card = true时返回，身份证裁剪切图的位置信息（坐标0点为左上角）
idcard_number_type	是	string	用于校验身份证号码、性别、出生是否一致，输出结果及其对应关系如下： - **1：** 身份证正面所有字段全为空 **0：** 身份证证号不合法，此情况下不返回身份证证号 **1：** 身份证证号和性别、出生信息一致 **2：** 身份证证号和性别、出生信息都不一致 **3：** 身份证证号和出生信息不一致 **4：** 身份证证号和性别信息不一致
words_result	是	array[]	定位和识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ location	是	array[]	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	否	string	识别结果字符串

身份证识别 返回示例

```
{
  "log_id": 2648325511,
  "direction": 0,
  "image_status": "normal",
```

```
    "idcard_type": "normal",
    "edit_tool": "Adobe Photoshop CS3 Windows",
    "words_result": {
        "住址": {
            "location": {
                "left": 267,
                "top": 453,
                "width": 459,
                "height": 99
            },
            "words": "南京市江宁区弘景大道3889号"
        },
        "公民身份号码": {
            "location": {
                "left": 443,
                "top": 681,
                "width": 589,
                "height": 45
            },
            "words": "330881199904173914"
        },
        "出生": {
            "location": {
                "left": 270,
                "top": 355,
                "width": 357,
                "height": 45
            },
            "words": "19990417"
        },
        "姓名": {
            "location": {
                "left": 267,
                "top": 176,
                "width": 152,
                "height": 50
            },
            "words": "伍云龙"
        },
        "性别": {
            "location": {
                "left": 269,
                "top": 262,
                "width": 33,
                "height": 52
            },
            "words": "男"
        },
        "民族": {
            "location": {
                "left": 492,
                "top": 279,
                "width": 30,
                "height": 37
            },
            "words": "汉"
        }
    },
    "words_result_num": 6
}
```


银行卡识别

识别银行卡并返回卡号和发卡行。

```
    """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"

# 调用银行卡识别
res_image = client.bankcard(image)
res_url = client.bankcardUrl(url)
print(res_image)
print(res_url)
```

银行卡识别 请求参数详情

参数	类型	是否必须	说明
image	string	和url二选一	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	string	和image二选一	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	string	true/false	是否检测图像朝向，默认检测，即：true。可选值包括true - 检测朝向；false - 不检测朝向。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。

银行卡识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
direction	int32	否	图像方向，当 detect_direction = true 时，返回该参数。 - - 1：未定义; - 0：正向; - 1：逆时针90度; - 2：逆时针180度; - 3：逆时针270度
result	object	是	返回结果
+ bank_card_number	string	是	银行卡卡号
+ valid_date	string	是	有效期
+ bank_card_type	uint32	是	银行卡类型，0：不能识别; 1：借记卡; 2：贷记卡（原信用卡大部分为贷记卡）; 3：准贷记卡; 4：预付费卡
+ bank_name	string	是	银行名，不能识别时为空
+ holder_name	string	是	持卡人姓名，不能识别时为空

银行卡识别 返回示例

```
{
  "log_id": 1447188951,
  "result": {
     "bank_card_number": "622500000000000",
     "bank_name": "招商银行",
     "bank_card_type": 1
  }
}
```

驾驶证识别

对机动车驾驶证所有关键字段进行识别。

```
    """ 读取图片 """
    def get_file_content(filePath):
        with open(filePath, 'rb') as fp:
            return fp.read()

    image = get_file_content('example.jpg')
    url = "https://www.x.com/sample.jpg"

    # 调用驾驶证识别
    res_image = client.drivingLicense(image)
    res_url = client.drivingLicenseUrl(url)
    print(res_image)
    print(res_url)

    # 如果有可选参数
    options = {}
    options["detect_direction"] = "true"
    res_image = client.drivingLicense(image, options)
    res_url = client.drivingLicenseUrl(url, options)
    print(res_image)
    print(res_url)
```

驾驶证识别 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和image二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	否	string	true/false	- **false：默认值，**不检测朝向，朝向是指输入图像是正常方向、逆时针旋转90/180/270度 - **true：**检测朝向
driving_license_side	否	string	front/back	- **front：默认值，**识别驾驶证正页 - **back：**识别驾驶证副页
unified_valid_period	否	bool	true/false	- **false: 默认值**，不进行归一化处理 - **true:** 归一化格式输出驾驶证的「有效起始日期」+「有效期限」及「有效期限」+「至」两种输出格式归一化为「**有效起始日期**」+「**失效日期**」
quality_warn	否	string	true/false	是否开启质量检测功能，仅在驾驶证正页识别时生效， - **false：默认值**，不输出质量告警信息 - **true：** 输出驾驶证遮挡、不完整质量告警信息
risk_warn	否	string	true/false	是否开启风险检测功能， - **false：默认值**，不输出风险告警信息 - **true**：开启，输出驾驶证复印、翻拍、PS等告警信息

驾驶证识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ words	否	string	识别结果字符串
warn_infos	否	array[]	当输入参数 driving_license_side=front，且 quality_warn=true 时输出， - shield：驾驶证证照存在遮挡告警提示 - incomplete：驾驶证证照边框不完整告警提示
risk_type	否	string	当输入参数 risk_warn=true 时返回识出的驾驶证的类型：normal-正常驾驶证；copy-复印件；screen-翻拍
edit_tool	否	string	当输入参数 risk_warn=true 时返回，如果检测驾驶证被编辑过，该字段指定编辑软件名称，如：Adobe Photoshop CC 2014 (Macintosh)，如果没有被编辑过则返回值为空

返回示例（驾驶证正页）

```
{
  "words_result": {
    "姓名": {
      "words": "王桃桃"
    },
    "至": {
      "words": "20210518"
    },
    "出生日期": {
      "words": "19880929"
    },
    "证号": {
      "words": "210282198809294228"
    },
    "住址": {
      "words": "辽宁省大连市甘井子区"
    },
    "初次领证日期": {
      "words": "20150518"
    },
    "国籍": {
      "words": "中国"
    },
    "准驾车型": {
      "words": "C1"
    },
    "性别": {
      "words": "女"
    },
    "有效期限": {
      "words": "20150518"
    },
    "发证单位": {
  "words": "北京市公安局公安交通管理局"
    }
  "log_id": 1321746413993852928,
  "words_result_num": 11,
  "direction": -1
}
```


返回示例（驾驶证副页）

```
{
  "words_result": {
    "姓名": {
      "words": "万万"
    },
    "记录": {
      "words": "请于每个记分周期结束后三十日接受审验。无记分的，免予本次审验。"
    },
    "证号": {
      "words": "513601198209290000"
    },
    "档案编号": {
      "words": "511600001169"
    }
  },
  "direction": 0,
  "words_result_num": 4,
  "log_id": 1483000040398531214
}
```


行驶证识别

对机动车行驶证正本所有关键字段进行识别。

```
  """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"

# 调用行驶证识别
res_image = client.vehicleLicense(image)
res_url = client.vehicleLicenseUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options = {}
options["detect_direction"] = "true"
options["accuracy"] = "normal"
res_image = client.vehicleLicense(image, options)
res_url = client.vehicleLicenseUrl(url, options)
print(res_image)
print(res_url)
```

行驶证识别 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和image二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	否	string	true/false	- **false：默认值**不进行图像方向自动矫正 - **true:** 开启图像方向自动矫正功能，可对旋转 90/180/270 度的图片进行自动矫正并识别
vehicle_license_side	否	string	front/back	- **front：默认值**，识别行驶证主页 - **back：**识别行驶证副页
unified	否	string	true/false	- **false：默认值**，不进行归一化处理 - **true：**对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出
quality_warn	否	string	true/false	是否开启质量检测功能，仅在行驶证正页识别时生效， - **false：默认值**，不输出质量告警信息 - **true：** 输出行驶证遮挡、不完整质量告警信息
risk_warn	否	string	true/false	是否开启风险检测功能， - **false：默认值**，不输出风险告警信息 - **true**：开启，输出行驶证复印、翻拍、PS等告警信息

行驶证识别 返回数据参数详情

字段	必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ words	否	string	识别结果字符串
warn_infos	否	array[]	当输入参数 vehicle_license_side=front，且 quality_warn=true 时输出， - shield：行驶证证照存在遮挡告警提示 - incomplete：行驶证证照边框不完整告警提示
risk_type	否	string	当输入参数 risk_warn=true 时返回识出的行驶证的类型：normal-正常行驶证；copy-复印件；screen-翻拍
edit_tool	否	string	当输入参数 risk_warn=true 时返回，如果检测行驶证被编辑过，该字段指定编辑软件名称，如：Adobe Photoshop CC 2014 (Macintosh)，如果没有被编辑过则返回值为空

行驶证识别 返回示例


```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "品牌型号": {
    "words": "保时捷GT37182RUCRE"
   },
   "发证日期": {
    "words": "20160104"
   },
   "使用性质": {
    "words": "非营运"
   },
   "发动机号码": {
    "words": "20832"
   },
   "号牌号码": {
    "words": "苏A001"
   },
   "所有人": {
    "words": "圆圆"
   },
   "住址": {
    "words": "南京市江宁区弘景大道"
   },
   "注册日期": {
    "words": "20160104"
   },
   "车辆识别代号": {
    "words": "HCE58"
   },
   "车辆类型": {
    "words": "小型轿车"
   }
  }
 }
}
```


车牌识别

识别机动车车牌，并返回号牌号码和车牌颜色。

```
    """ 读取图片 """
    def get_file_content(filePath):
        with open(filePath, 'rb') as fp:
            return fp.read()

    image = get_file_content('example.jpg')
    url = "https://www.x.com/sample.jpg"

    # 调用车牌识别
    res_image = client.licensePlate(image)
    res_url = client.licensePlateUrl(url)
    print(res_image)
    print(res_url)

    # 如果有可选参数
    options = {}
    options["multi_detect"] = "true"
    res_image = client.licensePlate(image, options)
    res_url = client.licensePlateUrl(url, options)
    print(res_image)
    print(res_url)
```

车牌识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
multi_detect	否	string	true false	false	是否检测多张车牌，默认为false，当置为true的时候可以对一张图片内的多张车牌进行识别

车牌识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
Color	string	是	车牌颜色
number	string	是	车牌号码

车牌识别 返回示例

```
{
    "log_id": 3583925545,
    "words_result": {
        "color": "blue",
        "number": "苏HS7766"
    }
}
```

营业执照识别

识别营业执照，并返回关键字段的值，包括单位名称、法人、地址、有效期、证件编号、社会信用代码等。

```
    """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"

# 调用营业执照识别
res_image = client.businessLicense(image)
res_url = client.businessLicenseUrl(url)
print(res_image)
print(res_url)
```

营业执照识别 请求参数详情

参数	类型	是否必须	说明
image	string	和url二选一	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	string	和image二选一	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效请注意关闭URL防盗链
detect_direction	string	否	此参数新版本无需传，支持自动检测图像旋转角度；朝向是指输入图像是正常方向、逆时针旋转90/180/270度
accuracy	string	否	此参数新版本无需传，可选值：normal,high
risk_warn	string	否	是否开启风险类型功能，默认不开启，即：false。 - false：不开启 - true：开启

营业执照识别 返回数据参数详情

参数	是否必须	类型	说明
log_id	是	uint64	请求标识码，随机数，唯一。
direction	否	uint32	图像方向，当图像旋转时，返回该参数。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
risk_type	否	string	当输入参数 risk_warn=true 时返回识出的营业执照的类型：normal-正常营业执照；copy-复印件；screen-翻拍；scan-扫描
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	否	string	识别结果字符串

营业执照识别 返回示例

```
{
  "log_id": 490058765,
  "words_result": {
    "单位名称": {
      "location": {
        "left": 500,
        "top": 479,
        "width": 618,
        "height": 54
      },
      "words": "袁氏财团有限公司"
    },
    "法人": {
      "location": {
        "left": 938,
        "top": 557,
        "width": 94,
        "height": 46
      },
      "words": "袁运筹"
    },
    "地址": {
      "location": {
        "left": 503,
        "top": 644,
        "width": 574,
        "height": 57
      },
      "words": "江苏省南京市中山东路19号"
    },
    "有效期": {
      "location": {
        "left": 779,
        "top": 1108,
        "width": 271,
        "height": 49
      },
      "words": "2015年02月12日"
    },
    "证件编号": {
      "location": {
        "left": 1219,
        "top": 357,
        "width": 466,
        "height": 39
      },
      "words": "苏餐证字(2019)第666602666661号"
    },
    "社会信用代码": {
      "location": {
        "left": 0,
        "top": 0,
        "width": 0,
        "height": 0
      },
      "words": "无"
    }
  },
  "words_result_num": 6
}
```

通用票据识别

用户向服务请求识别医疗票据、增值税发票、出租车票、保险保单等票据类图片中的所有文字，并返回文字在图中的位置信息。

```
    """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"

# 调用通用票据识别
res_image = client.receipt(image)
res_url = client.receiptUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options = {}
options["recognize_granularity"] = "big"
options["probability"] = "true"
options["accuracy"] = "normal"
options["detect_direction"] = "true"
res_image = client.receipt(image, options)
res_url = client.receiptUrl(url, options)
print(res_image)
print(res_url)
```

通用票据识别 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true/false	是否返回识别结果中每一行的置信度
accuracy	否	string	normal/缺省	normal：使用快速服务；缺省或其它值：使用高精度服务
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。可选值包括： - true：检测朝向； - false：不检测朝向，朝向是指输入图像是正常方向、逆时针旋转90/180/270度

通用票据识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
+ location	是	object{}	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串
+ chars	否	array[]	单字符结果，recognize_granularity=small 时存在
++ char	否	string	单字符识别结果
++ location	否	object{}	位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
+ probability	否	float	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值

通用票据识别 返回示例

```
{
  "log_id": 2661573626,
  "words_result": [
    {
      "location": {
        "left": 10,
        "top": 3,
        "width": 121,
        "height": 24
      },
      "words": "姓名:小明明",
      "chars": [
        {
          "location": {
            "left": 16,
            "top": 6,
            "width": 17,
            "height": 20
          },
          "char": "姓"
        }
        ...
      ]
    },
    {
      "location": {
        "left": 212,
        "top": 3,
        "width": 738,
        "height": 24
      },
      "words": "卡号/病案号:105353990标本编号:150139071送检科室:血液透析门诊病房",
      "chars": [
        {
          "location": {
            "left": 218,
            "top": 6,
            "width": 18,
            "height": 21
          },
          "char": "卡"
        }
        ...
      ]
    }
  ],
  "words_result_num": 2
}
```


自定义模板文字识别

自定义模板文字识别，是针对百度官方没有推出相应的模板，但是当用户需要对某一类卡证/票据（如房产证、军官证、火车票等）进行结构化的提取内容时，可以使用该产品快速制作模板，进行识别。

```python
    # 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()
image = get_file_content('aa.jpg')

# 必填参数
options = {}
# key固定为templateSign 后面给页面提供的 模板ID（templateSign） 的值即可
options["templateSign"] = ""
# 调用自定义模板文字识别
res_image = client.custom(image, options)
print(res_image)
```

自定义模板文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
options	是	object	用于传入额外参数，如templateSign、classifierId
+ templateSign	否	string	您在自定义文字识别平台制作的模板的ID
+ classifierId	否	string	分类器Id。这个参数和templateSign至少存在一个，优先使用templateSign。存在templateSign时，表示使用指定模板；如果没有templateSign而有classifierId，表示使用分类器去判断使用哪个模板

自定义模板文字识别 返回数据参数详情

字段	是否必选	类型	说明
error_code	number	number	0代表成功，如果有错误码返回可以参考下方错误码列表排查问题
error_msg	是	string	具体的失败信息，可以参考下方错误码列表排查问题
data	jsonObject	识别返回的结果	

自定义模板文字识别 返回示例


```
{
  "isStructured": true,
  "ret": [
    {
      "charset": [
        {
          "rect": {
            "top": 183,
            "left": 72,
            "width": 14,
            "height": 28
          },
          "word": "5"
        },
        {
          "rect": {
            "top": 183,
            "left": 90,
            "width": 14,
            "height": 28
```

```
            },
            "word": "4"
        },
        {
            "rect": {
                "top": 183,
                "left": 103,
                "width": 15,
                "height": 28
            },
            "word": "."
        },
        {
            "rect": {
                "top": 183,
                "left": 116,
                "width": 14,
                "height": 28
            },
            "word": "5"
        },
        {
            "rect": {
                "top": 183,
                "left": 133,
                "width": 19,
                "height": 28
            },
            "word": "元"
        }
    ],
    "word_name": "票价",
    "word": "54.5元"
},
{
    "charset": [
        {
            "rect": {
                "top": 144,
                "left": 35,
                "width": 14,
                "height": 28
            },
            "word": "2"
        },
        {
            "rect": {
                "top": 144,
                "left": 53,
                "width": 14,
                "height": 28
            },
            "word": "0"
        },
        {
            "rect": {
                "top": 144,
                "left": 79,
                "width": 14,
                "height": 28
            },
            "word": "1"
        }
```

```
            },
            {
                "rect": {
                    "top": 144,
                    "left": 97,
                    "width": 14,
                    "height": 28
                },
                "word": "7"
            }
        ]
    ]
}
```

试卷分析与识别

可对文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景。

```python
  """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用试卷分析与识别
res_image = client.docAnalysis(image)
res_url = client.docAnalysisUrl(url)
res_pdf = client.docAnalysisPdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)
```

识别结果 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
language_type	否	string	CHN_ENG/ ENG	识别语言类型，默认为CHN_ENG 可选值包括： = CHN_ENG：中英文 = ENG：英文
result_type	否	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 = big：返回行识别结果 = small：返回行识别结果之上还会返回单字结果
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	否	string	true/false	是否返回每行识别结果的置信度。默认为false
words_type	否	string	handwring_only/ handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	否	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
img_direction	否	int32	detect_direction=true 时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	是	uint32	识别结果数，表示results的元素个数

results	是	array[]	识别结果数组
+ words_type	是	string	文字属性（手写、印刷），handwriting 手写，print 印刷
+ words	是	array[]	整行的识别结果数组。
++ line_probability	否	array[]	line_probability=true 时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++ average	否	float	行置信度
+++ min	否	float	整行中单字的最低置信度
++ word	是	float	整行的识别结果
++ words_location	是	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位定位位置的长方形的宽度
+++ height	是	uint32	表示位置的长方形的高度
+ chars	否	array[]	result_type=small 时返回。单字符结果数组
++ char	否	string	result_type=small 时返回。每个单字的内容
++ chars_location	否	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
layouts_num	否	uint32	版面分析结果数，表示layout的元素个数
layouts	否	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+ layout	否	string	版面分析的标签结果。表格：table, 图：figure, 文本：text, 标题：title
+ layout_location	否	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	水平坐标（坐标0点为左上角）
+ layout_idx	否	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

仪器仪表盘读数识别

适用于不同品牌、不同型号的仪器仪表盘读数识别，广泛适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号，支持液晶屏、字轮表等表型。

```
    """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"

# 调用仪器仪表盘读数识别
res_image = client.meter(image)
res_url = client.meterUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options={}
options['poly_location']='true'
res_image = client.meter(image, options)
res_url = client.meterUrl(url, options)
print(res_image)
print(res_url)
```

识别结果 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px。支持jpg/jpeg/png/bmp格式.**注意：图片的base64编码是不包含图片头的，如（）**
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
probability	否	string	true/false	是否返回每行识别结果的置信度。默认为false
poly_location	否	string	true/false	位置信息返回形式，默认：false false：只给出识别结果所在长方形位置信息 true：除了默认的识别文字所在长方形的位置信息，还会给出文字所在区域的最小外接旋转矩形的4个点坐标信息

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	是	string	识别结果字符串
+ location	是	array[]	识别结果所在长方形位置信息
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ probability	否	string	probability=true 时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ poly_location	否	array[]	poly_location=true 时存在。文字所在区域的外接四边形的4个点坐标信息

返回示例

```
{
  "log_id": "1392680790663364608",
"words_result_num": 5
"words_result": [
{
 "words": "5.8",
 "location": {
  "top": 150,
  "left": 370,
  "width": 87,
  "height": 79
 }
},
{
 "words": "mmol/L",
 "location": {
  "top": 241,
  "left": 402,
  "width": 52,
  "height": 12
 }
},
{
 "words": "10:38",
 "location": {
  "top": 115,
  "left": 347,
  "width": 42,
  "height": 21
 }
},
{
 "words": "12-11",
 "location": {
  "top": 116,
  "left": 410,
  "width": 36,
  "height": 20
 }
},
{
 "words": "am",
 "location": {
  "top": 115,
  "left": 391,
  "width": 12,
  "height": 5
 }
}
],
}
```


网络图片文字识别（含位置版）

支持识别艺术字体或背景复杂的文字内容，除文字信息外，还可返回每行文字的位置信息、行置信度，以及单字符内容和位置等。

```
  """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用网络图片文字识别（含位置版）
res_image = client.webimageLoc(image)
res_url = client.webimageLocUrl(url)
res_pdf = client.webimageLocPdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options={}
options['probability']='true'
res_image = client.webimageLoc(image, options)
res_url = client.webimageLocUrl(url, options)
res_pdf = client.webimageLocPdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

网络图片文字识别（含位置版） 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向
probability	否	string	true/false	是否返回每行识别结果的置信度。默认为false
poly_location	否	string	true/false	是否返回文字所在区域的外接四边形的4个点坐标信息。默认为false
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。检测到的图像朝向： - -1：未定义； - 0 ：正向； - 1：逆时针旋转90度； - 2：逆时针旋转180度； - 3：逆时针旋转270度
words_result	是	array[]	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	是	string	整行的识别结果
+ location	是	object	整行的矩形框坐标。位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ probability	否	string	当 probability=true 时返回该字段。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ poly_location	否	array[]	当 probability=true 时返回该字段。文字所在区域的外接矩形的4个点坐标信息
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	垂直坐标（坐标0点为左上角）
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时返回该字段
++ char	否	string	单字符识别结果
++ location	否	object	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示定位定位位置的长方形的高度
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
返回示例			

```
{
  "log_id": 1390656223866519552,
  "words_result_num": 3,
  "words_result": [
    {
      "words": "梦想起航",
      "location": {
        "top": 328,
        "left": 1079,
        "width": 56,
        "height": 262
      },
    },
    {
      "words": "前往下一个目的地",
      "location": {
        "top": 329,
        "left": 1160,
        "width": 63,
        "height": 446
      },
    },
    {
      "words": "开始新的旅程",
      "location": {
        "top": 455,
        "left": 1246,
        "width": 63,
        "height": 340
      },
    }
  ],
}
```


增值税发票识别

支持对增值税普票、专票、卷票、电子发票、区块链发票的所有字段进行结构化识别，包括发票基本信息、销售方及购买方信息、商品信息、价税信息等，其中五要素识别准确率超过 99.9%； 同时，支持对增值税卷票的 21 个关键字段进行识别，包括发票类型、发票代码、发票号码、机打号码、机器编号、收款人、销售方名称、销售方纳税人识别号、开票日期、购买方名称、购买方纳税人识别号、项目、单价、数量、金额、税额、合计金额(小写)、合计金额(大写)、校验码、省、市，四要素平均识别准确率可达95%以上。

```
    """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用增值税发票识别
res_image = client.vatInvoice(image)
res_url = client.vatInvoiceUrl(url)
res_pdf = client.vatInvoicePdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options={}
options['type']='roll'
res_image = client.vatInvoice(image, options)
res_url = client.vatInvoiceUrl(url, options)
res_pdf = client.vatInvoicePdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
type	否	string	normal /roll	进行识别的增值税发票类型，默认为 normal，可缺省 - **normal**：可识别增值税普票、专票、电子发票 - **roll**：可识别增值税卷票
返回参数				

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_	是		

num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ ServiceType	是	string	发票消费类型。**不同消费类型输出：**餐饮、电器设备、通讯、服务、日用品食品、医疗、交通、其他
+ InvoiceType	是	string	发票种类。**不同类型发票输出**：普通发票、专用发票、电子普通发票、电子专用发票、通行费电子普票、区块链发票、通用机打电子发票
+ InvoiceTypeOrg	是	string	发票名称
+ InvoiceCode	是	string	发票代码
+ InvoiceNum	是	string	发票号码
+ InvoiceCodeConfirm	是	string	发票代码的辅助校验码，一般业务情景可忽略
+ InvoiceNumConfirm	是	string	发票号码的辅助校验码，一般业务情景可忽略
+ MachineNum	是	string	机打号码。仅增值税卷票含有此参数
+ MachineCode	是	string	机器编号。仅增值税卷票含有此参数
+ CheckCode	是	string	校验码。增值税专票无此参数
+ InvoiceDate	是	string	开票日期
+ PurchaserName	是	string	购方名称
+ PurchaserRegisterNum	是	string	购方纳税人识别号
+ PurchaserAddress	是	string	购方地址及电话
+ PurchaserBank	是	string	购方开户行及账号
+ Password	是	string	密码区
+ Province	是	string	省
+ City	是	string	市
+ SheetNum	是	string	联次信息。**专票**第一联到第三联分别输出：第一联：记账联、第二联：抵扣联、第三联：发票联；**普通发票**第一联到第二联分别输出：第一联：记账联、第二联：发票联
+ Agent	是	string	是否代开
+ CommodityN	是	array[]	货物名称

ame			
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityType	是	array[]	规格型号
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityUnit	是	array[]	单位
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityNum	是	array[]	数量
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityPrice	是	array[]	单价
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityAmount	是	array[]	金额
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTaxRate	是	array[]	税率
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTax	是	array[]	税额
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityPlateNum	是	array[]	车牌号。仅通行费增值税电子普通发票含有此参数
++ row	是	uint32	行号
++ word	是	string	内容

+ CommodityVehicleType	是	array[]	类型。仅通行费增值税电子普通发票含有此参数
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityStartDate	是	array[]	通行日期起。仅通行费增值税电子普通发票含有此参数
++ row	是	uint33	行号
++ word	是	string	内容
+ CommodityEndDate	是	array[]	通行日期止。仅通行费增值税电子普通发票含有此参数
++ row	是	uint33	行号
++ word	是	string	内容
+ OnlinePay	是	String	电子支付标识。仅区块链发票含有此参数
+ SellerName	是	string	销售方名称
+ SellerRegisterNum	是	string	销售方纳税人识别号
+ SellerAddress	是	string	销售方地址及电话
+ SellerBank	是	string	销售方开户行及账号
+ TotalAmount	是	uint32	合计金额
+ TotalTax	是	uint32	合计税额
+ AmountInWords	是	string	价税合计(大写)
+ AmountInFigures	是	uint32	价税合计(小写)
+ Payee	是	string	收款人
+ Checker	是	string	复核
+ NoteDrawer	是	string	开票人
+ Remarks	是	string	备注

返回示例


```
{
 "log_id": "5425496231209218858",
"words_result_num": 29,
"words_result": {
 "InvoiceNum": "14641426",
 "SellerName": "上海易火广告传媒有限公司",
 "CommodityTaxRate": [
  {
```

```
    "word": "6%",
    "row": "1"
   }
  ],
  "SellerBank": "中国银行南翔支行446863841354",
  "Checker": ":沈园园",
  "TotalAmount": "94339.62",
  "CommodityAmount": [
   {
    "word": "94339.62",
    "row": "1"
   }
  ],
  "InvoiceDate": "2016年06月02日",
  "CommodityTax": [
   {
    "word": "5660.38",
    "row": "1"
   }
  ],
  "PurchaserName": "百度时代网络技术(北京)有限公司",
  "CommodityNum": [
   {
    "word": "",
    "row": "1"
   }
  ],
       "Province": "上海",
       "City": "",
       "SheetNum": "第三联",
       "Agent": "否",
  "PurchaserBank": "招商银行北京分行大屯路支行86618202851000 30",
  "Remarks": "告传",
  "Password": "074/45781873408>/6>8>65*887676033/51+<5415>9/32--852>1+29<65>641-
5>66<500>87/*-34<943359034>716905113*4242>",
  "SellerAddress": ":嘉定区胜辛南路500号15幢1161室55033753",
  "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二属A2010-59108001",
  "InvoiceCode": "3100153130",
  "CommodityUnit": [
   {
    "word": "",
    "row": "1"
   }
  ],
  "Payee": ":徐蓉",
  "PurchaserRegisterNum": "110108787751579",
  "CommodityPrice": [
   {
    "word": "",
    "row": "1"
   }
  ],
  "NoteDrawer": "沈园园",
  "AmountInWords": "壹拾万圆整",
  "AmountInFiguers": "100000.00",
  "TotalTax": "5660.38",
  "InvoiceType": "专用发票",
  "SellerRegisterNum": "913101140659591751",
  "CommodityName": [
   {
    "word": "信息服务费",
    "row": "1"
   }
```

```
  ],
  "CommodityType": [
   {
    "word": "",
    "row": "1"
   }
  ]
 }
}
```

出租车票识别

支持识别全国各大城市出租车票的 16 个关键字段，包括发票号码、代码、车号、日期、总金额、燃油附加费、叫车服务费、省、市、单价、里程、上车时间、下车时间等。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https//www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用出租车票识别
res_image = client.taxiReceipt(image)
res_url = client.taxiReceiptUrl(url)
res_pdf = client.taxiReceiptPdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options={}
options['pdf_file_num'] = '1'
res_image = client.taxiReceipt(image, options)
res_url = client.taxiReceiptUrl(url, options)
res_pdf = client.taxiReceiptPdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必须	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
返回参数				

参数	是否必须	类型	说明
log_id	是	uint64	请求标识码，随机数，唯一。
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果数组
+ InvoiceCode	是	string	发票代号
+ InvoiceNum	是	string	发票号码
+ TaxiNum	是	string	车牌号
+ Date	是	string	日期
+ Time	是	string	上下车时间
+ PickupTime	是	string	上车时间
+ DropoffTime	是	string	下车时间
+ Fare	是	string	金额
+ FuelOilSurcharge	是	string	燃油附加费
+ CallServiceSurcharge	是	string	叫车服务费
+ TotalFare	是	string	总金额
+ Location	是	string	开票城市
+ Province	是	string	省
+ City	是	string	市
+ PricePerkm	是	string	单价
+ Distance	是	string	里程
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
"log_id":2034039896,
"words_result_num":6,
"words_result":
{
 "Date":"2017-11-26",
 "Fare":"¥153.30元",
 "InvoiceCode":"111001681009",
 "InvoiceNum":"90769610",
 "TaxiNum":"BV2062",
 "Time":"20:42-21:07",
 "FuelOilSurcharge": "¥0.00",
 "CallServiceSurcharge": "¥0.00",
       "Province": "浙江省",
       "City": "杭州市",
       "PricePerkm": "2.50元/KM",
       "Distance": "4.5KM"
}
}
```


VIN码识别

支持对车辆挡风玻璃处的车架号码进行识别。

```
""" 读取文件 """
def get_file_content(filePath):
  with open(filePath, "rb") as fp:
    return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用VIN码识别
res_image = client.vinCode(image)
res_url = client.vinCodeUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options={}
res_image = client.vinCode(image, options)
res_url = client.vinCodeUrl(url, options)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	VIN码识别结果

返回示例


```
{
"log_id": 246589877,
"words_result": [
 {
  "location": {
   "left": 124,
   "top": 11,
   "width": 58,
   "height": 359
  },
  "words": "LFV2A11K8D4010942"
 }
],
"words_result_num": 1
}
```


火车票识别

支持对红、蓝火车票的13个关键字段进行结构化识别，包括车票号码、始发站、目的站、车次、日期、票价、席别、姓名、座位号、身份证号、售站、序列号、时间。

```
""" 读取文件 """
def get_file_content(filePath):
  with open(filePath, "rb") as fp:
    return fp.read()

image = get_file_content('文件路径')
url = "https//www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用火车票识别
res_image = client.trainTicket(image)
res_url = client.trainTicketUrl(url)
res_pdf = client.trainTicketPdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options={}
res_image = client.trainTicket(image, options)
res_url = client.trainTicketUrl(url, options)
res_pdf = client.trainTicketPdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必须	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
返回参数				

参数	是否必须	类型	说明
log_id	是	uint64	请求标识码，唯一，用于调用失败后进行问题定位
direction	是	int32	图像方向 - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result	是	object{}	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ ticket_num	是	string	车票号
+ starting_station	是	string	始发站
+ train_num	是	string	车次号
+ destination_station	是	string	到达站
+ date	是	string	出发日期
+ ticket_rates	是	string	车票金额
+ seat_category	是	string	席别
+ name	是	string	乘客姓名
+ id_num	是	string	身份证号
+ serial_number	是	string	序列号
+ sales_station	是	string	售站
+ time	是	string	时间
+ seat_num	是	string	座位号
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
返回示例			


```
{
"log_id": "12317512659",
"direction": 1,
"words_result_num": 13,
"words_result": {
 "id_num": "2302051998****156X",
 "name": "裴一丽",
 "ticket_rates": "￥54.5元",
 "destination_station": "天津站",
 "seat_category": "二等座",
 "sales_station": "北京南",
 "ticket_num": "F05706",
 "seat_num": "02车03C号",
 "time": "09:36",
 "date": "2019年04月03日",
 "serial_number": "10010300067846",
 "train_num": "C255",
 "starting_station": "北京南站"
 }
}
```


数字识别

对图片中的数字进行提取和识别，自动过滤非数字内容，仅返回数字内容及其位置信息，识别准确率超过99%。

```
""" 读取文件 """
def get_file_content(filePath):
  with open(filePath, "rb") as fp:
    return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用数字识别
res_image = client.numbers(image)
res_url = client.numbersUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options={}
options['recognize_granularity']='small'
res_image = client.numbers(image, options)
res_url = client.numbersUrl(url, options)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。 可选值包括 - true：检测朝向； - false：不检测朝向

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时返回该字段
++ char	否	string	单字符识别结果
++ location	否	object	位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度

返回示例


```
{
  "log_id": 620759800,
  "words_result": [
    {
      "location": {
        "left": 56,
        "top": 0,
        "width": 21,
        "height": 210
      },
      "words": "3"
    }
  ],
  "words_result_num": 1
}
```


飞机行程单识别

支持对飞机行程单的24个字段进行结构化识别，包括电子客票号、印刷序号、姓名、始发站、目的站、航班号、日期、时间、票价、身份证号、承运人、民航发展基金、保险费、燃油附加费、其他税费、合计金额、填开日期、订票渠道、客票级别、座位等级、销售单位号、签注、免费行李、验证码。 同时，支持单张行程单上的多航班信息识别。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https//www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用飞机行程单识别
res_image = client.airTicket(image)
res_url = client.airTicketUrl(url)
res_pdf = client.airTicketPdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options={}
options["multi_detect"] = "false"
res_image = client.airTicket(image, options)
res_url = client.airTicketUrl(url, options)
res_pdf = client.airTicketPdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
multi_detect	否	string	true/false	控制是否开启多航班信息识别功能，**默认值：false** - **true：开启多航班信息识别功能**，开启后返回结果中对应字段格式将改为数组类型 - **false：** 不开启，仅识别单一航班信息

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ name	是	string	姓名
+ starting_station	是	string	始发站
+ destination_station	是	string	目的站
+ flight	是	string	航班号
+ date	是	string	日期
+ ticket_number	是	string	电子客票号码
+ fare	是	string	票价
+ dev_fund	是	string	民航发展基金/基建费
+ fuel_surcharge	是	string	燃油附加费
+ other_tax	是	string	其他税费
+ ticket_rates	是	string	合计金额
+ issued_date	是	string	填开日期
+ id_num	是	string	身份证号
+ carrier	是	string	承运人
+ time	是	string	时间
+ issued_by	是	string	订票渠道
+ serial_number	是	string	印刷序号
+ insurance	是	string	保险费
+ fare_basis	是	string	客票级别
+ class	是	string	座位等级
+ agent_code	是	string	销售单位号
+ endorsement	是	string	签注
+ allow	是	string	免费行李
+ ck	是	string	验证码
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
 // 识别单航班信息（multi_detect=false，或参数缺省）
{
    "log_id": 7306800033425229106,
    "direction": 0,
    "words_result_num": 18,
    "words_result": {
        "insurance": "20.00",
        "date": "2019-10-22",
        "allow": "20K",
        "flight": "CA6589",
        "issued_by": "中国国际航空服务有限公司",
        "starting_station": "武汉",
        "fare": "260.00",
```

```
        "endorsement": "不得签转改期退转",
        "ticket_rates": "350.00",
        "ck": "5866",
        "serial_number": "51523588676",
        "ticket_number": "7843708871196",
        "fuel_surcharge": "EXEMPT",
        "carrier": "南航",
        "issued_date": "2019-10-30",
        "other_tax": "",
        "fare_basis": "NREOW",
        "id_num": "411201123909020877",
        "destination_station": "合肥",
        "name": "郭达",
        "agent_code": "BJS19197300025",
        "time": "21:25",
        "class": "N",
        "dev_fund": "50.00"
    }
}

// 识别多航班信息（multi_detect=true)
{
    "words_result": {
        "log_id": "1280814270572920832",
        "words_result_num": 18
        "insurance": [
            {
                "word": "XXX"
            }
        ],
        "date": [
            {
                "word": "2019-10-18"
            },
            {
                "word": "2019-10-21"
            }
        ],
        "flight": [
            {
                "word": "CZ3565"
            },
            {
                "word": "CZ3566"
            }
        ],
        "issued_by": [
            {
                "word": "上海携程旅行社有限公司"
            }
        ],
        "starting_station": [
            {
                "word": "北京"
            }
        ],
        "fare": [
            {
                "word": "1080.00"
            }
        ],
        "ticket_rates": [
```

```
        {
            "word": "1420.00"
        }
    ],
    "serial_number": [
        {
            "word": "45956029770"
        }
    ],
    "ticket_number": [
        {
            "word": "7849648364314"
        }
    ],
    "fuel_surcharge": [
        {
            "word": "240.00"
        }
    ],
    "carrier": [
        {
            "word": "南航"
        },
        {
            "word": "南航"
        }
    ],
    "issued_date": [
        {
            "word": "2019-09-18"
        }
    ],
    "other_tax": [],
    "id_num": [
        {
            "word": "0789654700"
        }
    ],
    "destination_station": [
        {
            "word": "深圳"
        },
        {
            "word": "北京"
        }
    ],
    "name": [
        {
            "word": "姚佳"
        }
    ],
    "time": [
        {
            "word": "13:55"
        },
        {
            "word": "16:30"
        }
    ],
    "dev_fund": [
        {
            "word": "100.00"
```

```
        ]
    },
}
```

二维码识别

对图片中的二维码、条形码进行检测和识别，返回存储的文字信息。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用二维码识别
res_image = client.qrcode(image)
res_url = client.qrcodeUrl(url)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
codes_result_num	是	uint32	识别结果数，表示codes_result的元素个数
codes_result	是	array[]	定位和识别结果数组
+ type	是	string	识别码类型条码类型包括：9种条形码（UPC_A、UPC_E、EAN_13、EAN_8、CODE_39、CODE_93、CODE_128、ITF、CODABAR），4种二维码（QR_CODE、DATA_MATRIX、AZTEC、PDF_417）
+ text	是	string	条形码识别内容,暂时只限于识别中英文结果

返回示例

```
{
  "log_id": 863402790,
  "codes_result": [
```

```
        {
            "type": "QR_CODE",
            "text": [
                "中国",
                "北京"
            ]
        }
    ],
    "codes_result_num": 1
}
```


示例2（多个图的情况）：


```
{
    "log_id": 1508509437,
    "codes_result": [
        {
            "type": "QR_CODE",
            "text": [
                "HTTP://Q8R.HK/YELZ0"
            ]
        },
        {
            "type": "PDF_417",
            "text": [
                "PDF417倢丄TL-30偱撉傒庢傝壜擻偱偡丅"
            ]
        },
        {
            "type": "CODABAR",
            "text": [
                "000800"
            ]
        },
        {
            "type": "CODE_39",
            "text": [
                "1234567890"
            ]
        },
        {
            "type": "AZTEC",
            "text": [
                "www.tec-it.com"
            ]
        },
        {
            "type": "DATA_MATRIX",
            "text": [
                "Wikipedia, the free encyclopedia"
            ]
        },
        {
            "type": "CODE_93",
            "text": [
                "123456789"
            ]
        },
        {
            "type": "CODE_128",
            "text": [
                "50090500019191"
            ]
        },
```

```
        {
            "type": "EAN_8",
            "text": [
                "12345670"
            ]
        },
        {
            "type": "EAN_13",
            "text": [
                "6901234567892"
            ]
        },
        {
            "type": "UPC_E",
            "text": [
                "01234565"
            ]
        }
    ],
    "codes_result_num": 11
}
```

手写文字识别

支持对图片中的手写中文、手写数字进行检测和识别，针对不规则的手写字体进行专项优化，识别准确率可达90%以上。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用手写文字识别
res_image = client.handwriting(image)
res_url = client.handwritingUrl(url)
res_pdf = client.handwritingPdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options={}
options["recognize_granularity"] = "true"
options["detect_direction"] = "true"
options["probability"] = "true"
res_image = client.handwriting(image, options)
res_url = client.handwritingUrl(url, options)
res_pdf = client.handwritingPdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true/false	是否返回识别结果中每一行的置信度，默认为false，不返回置信度
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: true：检测朝向； false：不检测朝向

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串
+ chars	否	array[]	单字符结果，当 recognize_granularity=small 时存返回该字段
++ char	否	string	单字符识别结果
++ candidates	否	array[]	单字符识别结果的候选词内容
+++ word	否	string	单字符识别结果的候选词文字
+++ prob	否	string	单字符识别结果的候选词置信度
++ location	否	object	位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
probability	否	object	当 probability=true 时返回该字段，表示识别结果中每一行的置信度值，包含： - **average**： 行置信度平均值 - **variance**：行置信度方差 - **min**：行置信度最小值
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
      "location": {
        "left": 56,
        "top": 0,
        "width": 21,
        "height": 210
      },
      "words": "3"
    }
  ],
  "words_result_num": 1
}
```

护照识别

支持对中国大陆护照个人资料页所有15个字段进行结构化识别，包括国家码、护照号、姓名、姓名拼音、性别、出生地点、出生日期、签发地点（不支持境外签发地）、签发日期、有效期、签发机关、护照类型、国籍、MRZCode1、MRZCode2。

```python
""" 读取文件 """
def get_file_content(filePath):
  with open(filePath, "rb") as fp:
    return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用护照识别
res_image = client.passport(image)
res_url = client.passportUrl(url)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ location	是	uint32	水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ height	是	uint32	表示定位位置的长方形的高度
++ width	是	uint32	表示定位位置的长方形的宽度
+ words	是	string	识别内容

返回示例


```
{
  "log_id": 7377468409496932872,
  "words_result_num": 14,
  "words_result": {
    "国家码": {
      "location": {
        "width": 59,
        "top": 200,
        "left": 762,
        "height": 26
      },
      "words": "CHN"
    },
    "护照签发地点": {
      "location": {
        "width": 236,
        "top": 505,
        "left": 558,
        "height": 43
      },
      "words": "山东/SHANDONG"
    },
    "MRZCode2": {
      "location": {
        "width": 1252,
        "top": 797,
        "left": 145,
        "height": 88
      },
      "words": "E898657303CHNSDM0702<<<<<<<<<<<<<"
    },
    "有效期至": {
      "location": {
        "width": 287,
        "top": 528,
        "left": 955,
        "height": 46
      },
      "words": "20261004"
    },
    "签发机关": {
      "location": {
        "width": 271,
        "top": 583,
```

```
            "left": 552,
            "height": 42
        },
        "words": "出入境管理局"
    },
    "MRZCode1": {
        "location": {
            "width": 1201,
            "top": 781,
            "left": 162,
            "height": 45
        },
        "words": "PONSUN<<JIAJIA<<<<<<<<<<<<<<<<<<<<<<<<<<"
    },
    "护照号码": {
        "location": {
            "width": 237,
            "top": 215,
            "left": 994,
            "height": 48
        },
        "words": "E89865730"
    },
    "签发日期": {
        "location": {
            "width": 279,
            "top": 453,
            "left": 955,
            "height": 46
        },
        "words": "20161005"
    },
    "出生地点": {
        "location": {
            "width": 216,
            "top": 429,
            "left": 564,
            "height": 43
        },
        "words": "山东/SHANDONG"
    },
    "姓名": {
        "location": {
            "width": 159,
            "top": 247,
            "left": 581,
            "height": 34
        },
        "words": "孙嘉佳"
    },
    "姓名拼音": {
        "location": {
            "width": 229,
            "top": 279,
            "left": 578,
            "height": 41
        },
        "words": "SUN,JIAJIA"
    },
    "国籍": {
        "location": {
            "width": 209,
```

```
            "top": 366,
            "left": 695,
            "height": 42
        },
        "words": "中国/CHINESE"
    },
    "生日": {
        "location": {
            "width": 202,
            "top": 382,
            "left": 950,
            "height": 39
        },
        "words": "19950723"
    },
    "性别": {
        "location": {
            "width": 73,
            "top": 357,
            "left": 570,
            "height": 34
        },
        "words": "男/M"
    }
  }
}
```

通用机打发票识别

支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别，包括发票类型、发票号码、发票代码、开票日期、合计金额大写、合计金额小写、商品名称、商品单位、商品单价、商品数量、商品金额、机打代码、机打号码、校验码、销售方名称、销售方纳税人识别号、购买方名称、购买方纳税人识别号、合计税额等。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用通用机打发票识别
res_image = client.invoice(image)
res_url = client.invoiceUrl(url)
res_pdf = client.invoicePdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options={}
options["location"] = "true"
res_image = client.invoice(image, options)
res_url = client.invoiceUrl(url, options)
res_pdf = client.invoicePdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	是	int32	图像方向。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ InvoiceType	否	string	发票类型
+ InvoiceCode	否	string	发票代码
+ InvoiceNum	否	string	发票号码
+ InvoiceDate	否	string	开票日期
+ AmountInFiguers	否	string	合计金额小写
+ AmountInWords	否	string	合计金额大写
+ CommodityName	否	array[]	商品名称
++ row	否	unit32	行号
++ word	否	string	内容
+ CommodityUnit	否	array[]	商品单位
++ row	否	unit32	行号
++ word	否	string	内容
+ CommodityPrice	否	array[]	商品单价
++ row	否	unit32	行号
++ word	否	string	内容
+ CommodityNum	否	array[]	商品数量
++ row	否	unit32	行号
++ word	否	string	内容

++ word	否	string	内容
+ CommodityAmount	否	array[]	商品金额
++ row	否	unit32	行号
++ word	否	string	内容
+ IndustrySort	否	string	行业分类
+ MachineNum	否	string	机打号码
+ CheckCode	否	string	校验码
+ SellerName	否	string	销售方名称
+ SellerRegisterNum	否	string	销售方纳税人识别号
+ PurchaserName	否	string	购买方名称
+ PurchaserRegisterNum	否	string	购买方纳税人识别号
+ TotalTax	否	string	合计税额
+ Province	否	string	省
+ City	否	string	市
+ Time	否	string	时间
+ SheetNum	否	string	联次
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 4423022131715883558,
  "direction": 0,
  "words_result_num": 22,
  "words_result": {
     "City": "",
     "InvoiceNum": "01445096",
     "SellerName": "百度餐饮店",
     "IndustrSort": "生活服务",
     "Province": "广东省",
     "CommodityAmount": [
        {
           "word": "183.00",
           "row": "1"
        }
     ],
     "InvoiceDate": "2020年07月28日",
     "PurchaserName": "中信建投证券股份有限公司",
     "CommodityNum": [],
     "InvoiceCode": "144001901511",
     "CommodityUnit": [],
     "SheetNum": "",
     "PurchaserRegisterNum": "91442300845348OX9",
     "Time": "",
     "CommodityPrice": [],
     "AmountInFiguers": "183.00",
     "AmountInWords": "壹佰捌拾叁元整",
     "CheckCode": "61042119820421061301",
     "TotalTax": "183.00",
     "InvoiceType": "广东通用机打发票",
     "SellerRegisterNum": "61042119820421061301",
     "CommodityName": [
        {
           "word": "餐费",
           "row": "1"
        }
     ]
  }
}
```


户口本识别

支持对户口本内常住人口登记卡的全部 22 个字段进行结构化识别，包括户号、姓名、与户主关系、性别、出生地、民族、出生日期、身份证号、本市县其他住址、曾用名、籍贯、宗教信仰、身高、血型、文化程度、婚姻状况、兵役状况、服务处所、职业、何时由何地迁往本市、何时由何地迁往本址、登记日期。

```
    """ 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用户口本识别
res_image = client.householdRegister(image)
res_url = client.householdRegisterUrl(url)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和image二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
Name	是	object{}	姓名
+ words	是	string	所属字段的具体内容，以下各字段均含有此元素
Relationship	是	object{}	户主或与户主关系
Sex	是	object{}	性别
BirthAddress	是	object{}	出生地
Nation	是	object{}	民族
Birthday	是	object{}	生日
CardNo	是	object{}	身份证号
HouseholdNum	是	object{}	户号
FormerName	是	object{}	曾用名
Hometown	是	object{}	籍贯
OtherAddress	是	object{}	本市（县）其他住址
Belief	是	object{}	宗教信仰
Height	是	object{}	身高
BloodType	是	object{}	血型
Education	是	object{}	文化程度
MaritalStatus	是	object{}	婚姻状况
VeteranStatus	是	object{}	兵役状况
WorkAddress	是	object{}	服务处所
Career	是	object{}	职业
WWToCity	是	object{}	何时由何地迁来本市(县)
WWHere	是	object{}	何时由何地迁往本址
Date	是	object{}	登记日期

返回示例

```
{
 "log_id": 1301870459,
 "words_result": {
    "BirthAddress": {
       "words": "河南洛阳市郊区"
    },
    "Birthday": {
       "words": "2016-07-28"
    },
    "CardNo": {
       "words": "410311201607282825"
    },
    "Name": {
       "words": "孙翌晨"
    },
    "Nation": {
       "words": "汉族"
    },
    "Relationship": {
       "words": "户主"
    },
    "Sex": {
       "words": "男"
    },
 },
 "words_result_num": 7
}
```


机动车销售发票识别

支持对机动车销售发票的26个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、机器编号、购买方名称、购买方身份证号码/组织机构代码、车辆类型、厂牌型号、产地、合格证号、发动机号码、车架号码、价税合计、价税合计小写、销货单位名称、电话、纳税人识别号、账号、地址、开户银行、税率、税额、主管税务机关及代码、不含税价格、限乘人数。

```
 """ 读取文件 """
def get_file_content(filePath):
  with open(filePath, "rb") as fp:
    return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用机动车销售发票识别
res_image = client.vehicleInvoice(image)
res_url = client.vehicleInvoiceUrl(url)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	识别结果数组
+ InvoiceCode	是	string	发票代码/机打代码
+ InvoiceNum	是	string	发票号码/机打号码
+ InvoiceDate	是	string	开票日期
+ MachineCode	是	string	机器编号
+ Purchaser	是	string	购买方名称
+ PurchaserCode	是	string	购买方身份证号码/组织机构代码
+ VehicleType	是	string	车辆类型
+ ManuModel	是	string	厂牌型号
+ Origin	是	string	产地
+ CertificateNum	是	string	合格证号
+ EngineNum	是	string	发动机号码
+ VinNum	是	string	车架号码
+ PriceTax	是	string	价税合计
+ PriceTaxLow	是	string	价税合计小写
+ Saler	是	string	销货单位名称
+ SalerPhone	是	string	销货单位电话
+ SalerCode	是	string	销货单位纳税人识别号
+ SalerAccountNum	是	string	销货单位账号
+ SalerAddress	是	string	销货单位地址
+ SalerBank	是	string	销货单位开户银行
+ TaxRate	是	string	税率
+ Tax	是	string	税额
+ TaxAuthor	是	string	主管税务机关
+ TaxAuthorCode	是	string	主管税务机关代码
+ Price	是	string	不含税价格
+ LimitPassenger	是	string	限乘人数

返回示例

```
{
  "log_id": 283449393728149457,
  "words_result_num": 26,
  "words_result": {
     "InvoiceNum": "00875336",
     "Saler": "深圳市新能源汽车销售有限公司",
     "LimitPassenger": "5",
     "MachineCode": "669745967911",
     "VinNum": "LJLGTCRP1J4007581",
     "TaxRate": "16%",
     "PriceTaxLow": "106100.00",
     "InvoiceDate": "2018-11-29",
     "Price": "￥91465.52",
     "SalerBank": "中国工商银行股份有限公司深圳岭园支行",
     "TaxAuthor": "国家锐务总局深圳市龙岗区税务局第五税务所",
     "ManuModel": "江淮牌HFC7007EYBD6",
     "CertificateNum": "WCH0794J0976801",
     "Purchaser": "苏子潇",
     "VehicleType": "纯电动轿车",
     "InvoiceCode": "14975047560",
     "PriceTax": "壹拾万陆仟壹佰圆整",
     "SalerPhone": "0755-83489306",
     "SalerAddress": "深圳市龙岗区龙岗街道百世国际汽车城",
     "Origin": "安徽省合肥市",
     "EngineNum": "18958407",
     "Tax": "14634.48",
     "PurchaserCode": "5135934475603742222",
     "TaxAuthorCode": "14037589413",
     "SalerAccountNum": "中国工商银行股份有限公司深圳岭园支行",
     "SalerCode": "91449283464582922 78H"
  }
}
```


车辆合格证识别

支持对车辆合格证的23个关键字段进行结构化识别，包括合格证编号、发证日期、车辆制造企业名、车辆品牌、车辆名称、车辆型号、车架号、车身颜色、发动机型号、发动机号、燃料种类、排量、功率、排放标准、轮胎数、轴距、轴数、转向形式、总质量、整备质量、驾驶室准乘人数、最高设计车速、车辆制造日期。

```
    """ 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用车辆合格证
res_image = client.vehicleCertificate(image)
res_url = client.vehicleCertificateUrl(url)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	识别结果数组
+ CertificationNo	是	string	合格证编号
+ CertificateDate	是	string	发证日期
+ Manufacturer	是	string	车辆制造企业名
+ CarBrand	是	string	车辆品牌
+ CarName	是	string	车辆名称
+ CarModel	是	string	车辆型号
+ VinNo	是	string	车架号
+ CarColor	是	string	车身颜色
+ EngineType	是	string	发动机型号
+ EngineNo	是	string	发动机号
+ FuelType	是	string	燃料种类
+ Displacement	是	string	排量
+ Power	是	string	功率
+ EmissionStandard	是	string	排放标准
+ TyreNum	是	string	轮胎数
+ Wheelbase	是	string	轴距
+ AxleNum	是	string	轴数
+ SteeringType	是	string	转向形式
+ TotalWeight	是	string	总质量
+ SaddleMass	是	string	整备质量
+ LimitPassenger	是	string	驾驶室准乘人数
+ SpeedLimit	是	string	最高设计车速
+ ManufactureDate	是	string	车辆制造日期

返回示例

```
{
  "log_id": 14814098736243057,
  "words_result_num": 23,
  "words_result": {
     "ManufactureDate": "2016年10月13日",
     "CarColor": "红",
     "LimitPassenger": "2",
     "EngineType": "WP12.460E50",
     "TotalWeight": "25000",
     "Power": "338",
     "CertificationNo": "WEK29JX98645437",
     "FuelType": "汽油",
     "Manufacturer": "陕西汽车集团有限责任公司",
     "SteeringType": "方向盘",
     "Wheelbase": "3175+1350",
     "SpeedLimit": "105",
     "EngineNo": "1418K129178",
     "SaddleMass": "8600",
     "AxleNum": "3",
     "CarModel": "SX4250MC4",
     "VinNo": "LZGJHYD83JX197344",
     "CarBrand": "陕汽牌",
     "EmissionStandard": "GB17691-2005国V,GB3847-2005",
     "Displacement": "11596",
     "CertificateDate": "2018年11月28日",
     "CarName": "牵引汽车",
     "TyreNum": "10"
  }
}
```


网约车行程单识别

对各大主要服务商的网约车行程单进行结构化识别，包括滴滴打车、花小猪打车、高德地图、曹操出行、阳光出行，支持识别服务商、行程开始时间、行程结束时间、车型、总金额等16 个关键字段。

```
""" 读取文件 """
def get_file_content(filePath):
  with open(filePath, "rb") as fp:
    return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用网约车行程单识别
res_image = client.onlineTaxiItinerary(image)
res_url = client.onlineTaxiItineraryUrl(url)
res_pdf = client.onlineTaxiItineraryPdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options={}
options["pdf_file_num"] = "1"
res_pdf = client.onlineTaxiItineraryPdf(image, options)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	说明
image	和url二选一	string	图像数据，base64编码后进行urlencode，base64编码去除编码头（data:image/jpeg;base64），要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效，请注意关闭URL防盗链
pdf_file	和image/url 三选一	string	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
返回参数详情			

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ ServiceProvider	是	string	服务商
+ StartTime	是	string	行程开始时间
+ EndTime	是	string	行程结束时间
+ Phone	是	string	行程人手机号
+ ApplicationDate	是	string	申请日期
+ TotalFare	是	string	总金额
+ ItemNum	是	array	行程信息中包含的行程数量
+ Items	是	array	行程信息
++ ItemId	是	string	行程信息的对应序号
++ PickupTime	是	string	上车时间
++ PickupDate	是	string	上车日期
++ CarType	是	string	车型
++ Distance	是	string	里程
++ StartPlace	是	string	起点
++ DestinationPlace	是	string	终点
++ City	是	string	城市
++ Fare	是	string	金额
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 1385196013945356288,
  "words_result_num": 7
  "words_result": {
     "TotalFare": "2316",
     "EndTime": "2020-07-30 19:00",
     "Phone": "13000000000",
     "ServiceProvider": "滴滴企业版",
     "StartTime": "2020-07-01 16:00",
     "ApplicationDate": "2017-12-08",
     "ItemId": "3"
     "items": [
        {
           "ItemId": "1",
           "StartPlace": "鱼化寨地铁-D口",
           "PickupTime": "16:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.7",
           "PickupDate": "20-07-01",
           "DestinationPlace": "创新港",
           "Fare": "20.86"
        },
        {
           "ItemId": "2",
           "StartPlace": "科学园东门",
           "PickupTime": "14:56",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "91",
           "PickupDate": "20-07-02",
           "DestinationPlace": "鱼化寨地铁站",
           "Fare": "18.58"
        },
        {
           "ItemId": "3",
           "StartPlace": "中俄丝路创新园东门",
           "PickupTime": "19:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.1",
           "PickupDate": "20-07-30",
           "DestinationPlace": "新门地铁站",
           "Fare": "20.38"
        },
     ],
  },
}
```


磅单识别

结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单。

```
""" 读取文件 """
def get_file_content(filePath):
  with open(filePath, "rb") as fp:
    return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用磅单识别
res_image = client.weightNote(image)
res_url = client.weightNoteUrl(url)
res_pdf = client.weightNotePdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options={}
options["pdf_file_num"] = "1"
options["probability"] = "false"
res_image = client.weightNote(image, options)
res_url = client.weightNoteUrl(url, options)
res_pdf = client.weightNotePdf(image, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ PlateNum	是	object	车牌号
+ PrintTime	是	object	打印时间
+ CrossWeight	是	object	毛重
+ TareWeight	是	object	皮重
+ NetWeight	是	object	净重
+ SendingCompany	是	object	发货单位
+ ReceivingCompany	是	object	收货单位
+ DeliveryNumber	是	object	单号
++ word	是	string	字段识别结果，以上各字段均包含此参数
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "words_result": [
    {
      "TareWeight": [
      {
            "word": "14.20"
        }
      ],
      "CrossWeight": [
        {
          "word": "50.70"
        }
      ],
      "PlateNum": [
        {
          "word": "京A12356"
        }
      ],
      "SendingCompany": [
       {
            "word": "宣化县耿矿煤业有限公司"
        }
      ],
      "DeliveryNumber": [
       {
            "word": "278933000"
        }
      ],
      "ReceivingCompany": [
        {
          "word": "宁夏市京裕达实业公司"
        }
      ],
      "PrintTime": [
        {
          "word": "2020年1月1日"
        }
      ],
      "NetWeight": [
        {
          "word": "36.50"
        }
      ]
    }
  ],
  "words_result_num": 1,
  "log_id": 1428311410130160734
}
```


医疗费用明细识别

支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含项目类型、项目名称、单价、数量、规格、金额，其中北京地区识别效果最佳。

```
""" 读取文件 """
def get_file_content(filePath):
  with open(filePath, "rb") as fp:
    return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用医疗费用明细识别
res_image = client.medicalDetail(image)
res_url = client.medicalDetailUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options={}
options["location"] = "1"
options["probability"] = "false"
res_image = client.medicalDetail(image, options)
res_url = client.medicalDetailUrl(url, options)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省** - **false**：不返回字段位置信息 - **true**：返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ Name	是	object	姓名
+ Date	是	object	日期
+ PatientID	是	object	病人ID
+ TotalAmount	是	object	总金额
+ word	是	string	字段识别结果，以上各字段均包含此参数
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+ CostDetail	是	array[]	项目明细

CostDetail字段包含多个Array，每个数组包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：项目类型、项目名称、单价、数量、规格、金额
++ word	word_name字段对应的识别结果

返回示例


```
{
  "log_id": 1397090241579319296,
  "words_result_num": 5,
  "words_result": {
    "PatientID": {
        "word": "23683829"
    },
    "TotalAmount": {
        "word": "600.00"
    },
    "Date": {
        "word": "2020年11月04日"
    },
    "Name": {
        "word": "范浩"
    },
    "CostDetail": [
        [
            {
                "word_name": "清单项目名称",
                "word": "普通过敏原(新)筛查"
            },
            {
                "word_name": "单价",
                "word": "580.00"
```

```
            },
            {
                "word_name": "数量",
                "word": "1.00"
            },
            {
                "word_name": "金额",
                "word": "580.00"
            },
            {
                "word_name": "规格",
                "word": "次"
            },
            {
                "word_name": "项目类型",
                "word": "化验费"
            }
        ],
        [
            {
                "word_name": "清单项目名称",
                "word": "总IgE测定"
            },
            {
                "word_name": "单价",
                "word": "20.00"
            },
            {
                "word_name": "数量",
                "word": "1.00"
            },
            {
                "word_name": "金额",
                "word": "20.00"
            },
            {
                "word_name": "规格",
                "word": "次"
            },
            {
                "word_name": "项目类型",
                "word": "化验费"
            }
        ]
    ],
  },
}
```


办公文档识别

可对办公类文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用办公文档识别
res_image = client.doc_analysis_office(image)
res_url = client.doc_analysis_officeUrl(url)
res_pdf = client.doc_analysis_officePdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options = {}
options["probability"] = "true"
options["result_type"] = "small"
res_image = client.doc_analysis_office(image, options)
res_url = client.doc_analysis_officeUrl(url, options)
res_pdf = client.doc_analysis_officePdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
language_type	否	string	CHN_ENG/ ENG	识别语言类型，默认为CHN_ENG 可选值包括： =CHN_ENG：中英文 =ENG：英文
result_type	否	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	否	string	true/false	是否返回每行识别结果的置信度。默认为false
disp_line_poly	否	string	true/false	是否返回每行的四角点坐标。默认为false
words_type	否	string	handwring_only/ handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	否	string	true/false	是否分析文档版面：包括layout（图、表、标题、段落、目录）；attribute（栏、页眉、页脚、页码、脚注）的分析输出
erase_seal	否	string	true/false	是否先擦除水印、印章后再识别文档

返回参数详情

字段	是否必选	类型	说明

log_id	是	uint64	唯一的log id，用于问题定位
img_direction	否	int32	detect_direction=true 时返回该字段。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	是	uint32	识别结果数，表示results的元素个数
results	是	array[]	识别结果数组
+ words_type	是	string	文字属性（手写、印刷），handwriting 手写，print 印刷
+ words	是	array[]	整行的识别结果数组
++ line_probability	否	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++ average	否	float	行置信度
+++ min	否	float	整行中单字的最低置信度
++ word	是	float	整行的识别结果
++ poly_location	否	array[]	是否返回每行的四角点坐标，disp_line_poly=true时返回
++ words_location	是	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位定位位置的长方形的宽度
+++ height	是	uint32	表示位置的长方形的高度
+ chars	否	array[]	result_type=small时返回。单字符结果数组
++ char	否	string	result_type=small时返回。每个单字的内容
++ chars_location	否	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
layouts_num	否	uint32	版面分析结果数，表示layout的元素个数
layouts	否	array[]	每个「栏：section」里面的文档版面模块数组，包含表格、图、段落文本、标题、目录等5个模块；每个模块的坐标位置；段落文本和表格内文本内容对应的行序号id。

+ layout	否	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title ，目录:contents
+ layout_location	否	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	水平坐标（坐标0点为左上角）
+ layout_idx	否	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
sec_rows	否	uint32	将所有的版面中的「栏:section」内容表示成 M x N 的网格，sec_rows = M
sec_cols	否	uint32	将所有的版面中的「分栏」内容表示成 M x N 的网格，sec_cols = N
sections	否	array[]	一张图片中包含的5大版面属性，包含：栏，页眉，页脚，页码，脚注，该数组里有属性的标签、属性的位置、属性所包含文本内容的id序号。 其中，栏（section）里面包含5个模块内容，有：表格、图、段落文本、标题、目录（在返回参数layouts里输出）。
+ attribute	否	string	版面分析的属性标签结果，栏:section, 页眉:header, 页脚:footer, 页码:number，脚注:footnote。
+ attri_location	否	array[]	版面分析的属性所在位置，四个顶点: 左上，右上，右下，左下
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	水平坐标（坐标0点为左上角）
+ sec_idx	否	string	sections返回参数中的5个版面属性里，包含的内容序号标识
++ idx	否	string	sections返回参数中的5个版面属性里，每个属性下包含的文本行id序号
++ para_idx	否	string	当且仅当attribute=section时才会返回。表示，返回参数中的「栏：section」里面，所包含的表格、图、段落文本、标题、目录等5个模块返回的顺序号id（即layouts返回结果中，每个模块的返回顺序号）
++ row_idx	否	string	当且仅当attribute=section时才会返回。表示，将所有栏表示成 M xN 的网格，所属网格的行的id。
++ col_idx	否	string	当且仅当attribute=section时才会返回。表示，将所有栏表示成 M xN 的网格，所属网格的列的id。

返回示例


```
{
"results_num": 5,
"log_id": "1410491260247950412",
"results": [
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 88,
    "left": 442,
```

```
    "width": 142,
    "height": 49
   },
   "word": "行程单"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 241,
    "left": 439,
    "width": 393,
    "height": 37
   },
   "word": "美国东海岸名校8天7晚"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 318,
    "left": 436,
    "width": 774,
    "height": 31
   },
   "word": "国会大厦位于华盛顿25米高的国会山上,是美国的心脏建筑。"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 374,
    "left": 434,
    "width": 805,
    "height": 31
   },
   "word": "中央顶楼上的大圆顶上立有一尊6米高的自由女神青铜雕像。"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 431,
    "left": 436,
    "width": 556,
    "height": 31
   },
   "word": "东面的大草坪是历届总统举行就职典礼的地方。"
  }
 }
]
}
```


印章识别

检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用印章识别
res_image = client.seal(image)
res_url = client.sealUrl(url)
res_pdf = client.sealPdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options = {}
options["pdf_file_num"] = "1"
res_image = client.seal(image, options)
res_url = client.sealUrl(url, options)
res_pdf = client.sealPdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
result_num	是	uint32	识别结果数，表示results的元素个数
result	是	array[]	定位结果数组
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ probability	是	float	每一个识别结果的置信度值
+ type	是	string	印章的类别，共有circle（圆章），ellipse（椭圆章），rectangle（方章）三种
+ major	是	object	主字段内容
++ words	是	string	主字段识别内容，即章内上环弯曲文字结果
++ probability	是	float	主字段识别内容的置信度
+ minor	是	array[]	其他字段内容，即除主字段外的文字识别内容均放置于该参数中返回，若章内不存在其他字段文字，则该参数为空
++ words	是	string	其他字段识别内容
++ probability	是	float	其他字段识别内容的置信度
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
"result": [
 {
  "major": {
   "probability": 0.99759155511856,
   "words": "峨眉山旅游股份有限公司成都峨眉山雪芽大酒店分公司"
  },
  "minor": [
   {
    "probability": 0.99994027614594,
    "words": "前厅部"
   }
  ],
  "probability": 0.9936261177063,
  "location": {
   "top": 594,
   "left": 918,
   "width": 150,
   "height": 142
  },
  "type": "circle"
 }
],
"log_id": "1349006147834609664",
"result_num": 1
}
```


机动车登记证书识别

支持对机动车登记证书的15个关键字段进行结构化识别，包括编号、机动车所有人、登记机关、登记日期、登记编号、车辆类型等，同时支持检测发证机关章。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用机动车登记证书识别
res_image = client.vehicle_registration_certificate(image)
res_url = client.vehicle_registration_certificateUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options = {}
res_image = client.vehicle_registration_certificate(image, options)
res_url = client.vehicle_registration_certificateUrl(url, options)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	是	int32	图像方向 - 1：未定义， 0：正向， 1：逆时针90度， 2：逆时针180度， 3：逆时针270度
words_result	是	object	识别结果数组
+ number	是	object	编号
+ name_idcard_no	是	object	机动车所有人/身份证明名称/号码
+ registration_authority	是	object	登记机关
+ registration_date	是	object	登记日期
+ registration_num	是	object	机动车登记编号
+ vehicle_model	是	object	车辆型号
+ vehicle_type	是	object	车辆类型
+ vin	是	object	车架号
+ engine_num	是	object	发动机号
+ seating_capacity	是	object	核定载客
+ body_color	是	object	车身颜色
+ nature_of_use	是	object	使用性质
+ date_of_production	是	object	出厂日期
+ date_of_issue	是	object	发证日期
+ seal_of_issue_authority	是	object	发证机关章，1表示有，0表示无

返回示例

```
{
  "words_result": {
    "registration_authority": {
      "words": "江苏省昆山市公安局交通巡逻警察大队"
    },
    "vehicle_model": {
      "words": "DHW6691R8CEE"
    },
    "vehicle_type": {
      "words": "小型普通客车"
    },
    "registration_num": {
      "words": "苏A88FF2"
    },
    "engine_num": {
      "words": "2005533"
    },
    "number": {
      "words": "32004574998"
    },
    "body_color": {
      "words": "白"
    },
    "registration_date": {
      "words": "2016-06-06"
    },
    "date_of_production": {
      "words": "2016-03-11"
    },
    "seating_capacity": {
      "words": "7"
    },
    "date_of_issue": {
      "words": "2016-06-06"
    },
    "nature_of_use": {
      "words": "非营运"
    },
    "vin": {
      "words": "LVHRR8836G5004527"
    },
    "seal_of_issue_authority": {
      "words": "1"
    },
    "name_idcard_no": {
      "words": "汽车服务有限公司/统一社会信用代码/91320583088874412F"
    }
  },
  "log_id": 1392089398849306624,
  "direction": "0"
}
```


智能财务票据识别

支持财务场景中13种常见票据的分类及结构化识别，包括增值税发票、卷票、机打发票、定额发票、火车票、出租车票、网约车行程单、飞机行程单、汽车票、过路过桥费、船票、机动车/二手车销售发票。支持多张不同种类票据在同一张图片上的混贴场景，可返回每张票据的位置、种类及票面信息的结构化识别结果。

```python
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 调用智能财务票据识别
res_image = client.multipleInvoice(image)
res_url = client.multipleInvoiceUrl(url)
res_pdf = client.multipleInvoicePdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)

# 如果有可选参数
options = {}
options["probability"] = "true"
options["location"] = "true"
res_image = client.multipleInvoice(image, options)
res_url = client.multipleInvoiceUrl(url, options)
res_pdf = client.multipleInvoicePdf(pdf_file, options)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
verify_parameter	否	string	true/false	是否开启验真，默认为 false，即不开启，**当为 true 时，返回匹配发票验真接口所需的6要素信息，具体返回信息详见末尾说明**
probability	否	string	true/false	是否返回字段置信度，默认为 false ，即不返回
location	否	string	true/false	是否返回字段位置坐标，默认为 false，即不返回

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ probability	是	string	表示单张票据分类的置信度
+ left	是	string	表示单张票据定位位置的长方形左上顶点的水平坐标
+ top	是	string	表示单张票据定位位置的长方形左上顶点的垂直坐标
+ width	是	string	表示单张票据定位位置的长方形的宽度
+ height	是	string	表示单张票据定位位置的长方形的高度
+ type	是	string	每一张票据的种类
+ result	是	array[]	单张票据的识别结果数组
type 字段会返回以下17种结果，每种结果对应的票据类型详见下表			

type 返回结果	说明
vat_invoice	增值税发票
taxi_receipt	出租车票
train_ticket	火车票
quota_invoice	定额发票
air_ticket	机票行程单
roll_normal_invoice	卷票
printed_invoice	机打发票
bus_ticket	汽车票
toll_invoice	过路过桥费发票
ferry_ticket	船票
motor_vehicle_invoice	机动车销售发票
used_vehicle_invoice	二手车发票
taxi_online_ticket	网约车行程单
limit_invoice	限额发票
shopping_receipt	购物小票
pos_invoice	POS小票
others	其他

type 的返回结果为 vat_invoice，即“增值税发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceTypeOr	是	array[]	发票名称

g			
++ InvoiceType	是	array[]	增值税发票的细分类型。不同细分类型的增值税发票发票输出：普通发票、专用发票、电子普通发票、电子专用发票、通行费电子普票、区块链发票、通用机打电子发票
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ InvoiceCodeConfirm	是	array[]	发票代码的辅助校验码，一般业务情景可忽略
++ InvoiceNumConfirm	是	array[]	发票号码的辅助校验码，一般业务情景可忽略
++ CheckCode	是	array[]	校验码。增值税专票无此参数
++ InvoiceDate	是	array[]	开票日期
++ PurchaserName	是	array[]	购方名称
++ PurchaserRegisterNum	是	array[]	购方纳税人识别号
++ PurchaserAddress	是	array[]	购方地址及电话
++ PurchaserBank	是	array[]	购方开户行及账号
++ Password	是	array[]	密码区
++ Province	是	array[]	省
++ City	是	array[]	市
++ SheetNum	是	array[]	联次信息。**专票**第一联到第三联分别输出：第一联：记账联、第二联：抵扣联、第三联：发票联；**普通发票**第一联到第二联分别输出：第一联：记账联、第二联：发票联
++ Agent	是	array[]	是否代开
++ OnlinePay	是	String	电子支付标识。仅区块链发票含有此参数
++ SellerName	是	array[]	销售方名称
++ SellerRegisterNum	是	array[]	销售方纳税人识别号
++ SellerAddress	是	array[]	销售方地址及电话
++ SellerBank	是	array[]	销售方开户行及账号
++			

++ TotalAmount	是	array[]	合计金额
++ TotalTax	是	array[]	合计税额
++ AmountInWords	是	array[]	价税合计(大写)
++ AmountInFigures	是	array[]	价税合计(小写)
++ Payee	是	array[]	收款人
++ Checker	是	array[]	复核
++ NoteDrawer	是	array[]	开票人
++ Remarks	是	array[]	备注
+++ word	是	string	识别结果字符串，以上各字段均包含
++ CommodityName	是	array[]	货物名称
++ CommodityType	是	array[]	规格型号
++ CommodityUnit	是	array[]	单位
++ CommodityNum	是	array[]	数量
++ CommodityPrice	是	array[]	单价
++ CommodityAmount	是	array[]	金额
++ CommodityTaxRate	是	array[]	税率
++ CommodityTax	是	array[]	税额
++ CommodityPlateNum	是	array[]	车牌号。仅通行费增值税电子普通发票含有此参数
++ CommodityVehicleType	是	array[]	类型。仅通行费增值税电子普通发票含有此参数
++			

CommodityStartDate	是	array[]	通行日期起。仅通行费增值税电子普通发票含有此参数
++ CommodityEndDate	是	array[]	通行日期止。仅通行费增值税电子普通发票含有此参数
+++ row	是	uint32	行号，以上各字段均包含
+++ word	是	string	内容，以上各字段均包含

type 的返回结果为 taxi_receipt，即“出租车票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceCode	是	array[]	发票代号
++ InvoiceNum	是	array[]	发票号码
++ TaxiNum	是	array[]	车牌号
++ Date	是	array[]	日期
++ Time	是	array[]	上下车时间
++ PickupTime	是	array[]	上车时间
++ DropoffTime	是	array[]	下车时间
++ Fare	是	array[]	金额
++ FuelOilSurcharge	是	array[]	燃油附加费
++ CallServiceSurcharge	是	array[]	叫车服务费
++ TotalFare	是	array[]	总金额
++ Location	是	array[]	开票城市
++ Province	是	array[]	省
++ City	是	array[]	市
++ PricePerkm	是	array[]	单价
++ Distance	是	array[]	里程
+++ word	是	string	识别结果字符串，以上各字段均包含
type 的返回结果为 train_ticket，即“火车票”时，识别结果的返回字段如下			

字段	是否必选	类型	说明
++ ticket_num	是	array[]	车票号
++ starting_station	是	array[]	始发站
++ train_num	是	array[]	车次号
++ destination_station	是	array[]	到达站
++ date	是	array[]	出发日期
++ ticket_rates	是	array[]	车票金额
++ seat_category	是	array[]	席别
++ name	是	array[]	乘客姓名
++ ID_card	是	array[]	身份证号
++ serial_number	是	array[]	序列号
++ sales_station	是	array[]	售站
++ time	是	array[]	时间
++ seat_num	是	array[]	座位号
++ Waiting_area	是	array[]	候检区
+++ word	是	string	识别结果字符串，以上各字段均包含
type 的返回结果为 quota_invoice，即“定额发票”时，识别结果的返回字段如下			

字段	是否必选	类型	说明
++ invoice_code	是	array[]	发票代码
++ invoice_number	是	array[]	发票号码
++ invoice_rate	是	array[]	金额
++ invoice_rate_in_figure	是	array[]	金额小写
++ invoice_rate_in_word	是	array[]	金额大写
++ Province	是	array[]	省
++ City	是	array[]	市
++ Location	是	array[]	发票所在地
++ invoice_type	是	array[]	发票名称
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 air_ticket，即“飞机行程单”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ name	是	array[]	姓名
++ starting_station	是	array[]	始发站
++ destination_station	是	array[]	目的站
++ flight	是	array[]	航班号
++ date	是	array[]	日期
++ ticket_number	是	array[]	电子客票号码
++ fare	是	array[]	票价
++ dev_fund	是	array[]	民航发展基金/基建费
++ oil_money	是	array[]	燃油附加费
++ other_tax	是	array[]	其他税费
++ ticket_rates	是	array[]	合计金额
++ start_date	是	array[]	填开日期
++ id_no	是	array[]	身份证号
++ carrier	是	array[]	承运人
++ time	是	array[]	时间
++ issued_by	是	array[]	订票渠道
++ serial_number	是	array[]	印刷序号
++ insurance	是	array[]	保险费
++ fare_basis	是	array[]	客票级别
++ class	是	array[]	座位等级
++ agent_code	是	array[]	销售单位号
++ endorsement	是	array[]	签注
++ allow	是	array[]	免费行李
++ ck	是	array[]	验证码
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 roll_normal_invoice，即“卷票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceType	是	array[]	发票名称
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ MachineNum	是	array[]	机打号码。仅增值税卷票含有此参数
++ MachineCode	是	array[]	机器编号。仅增值税卷票含有此参数
++ InvoiceDate	是	array[]	开票日期
++ PurchaserName	是	array[]	购方名称
++ PurchaserRegisterNum	是	array[]	购方纳税人识别号
++ SellerName	是	array[]	销售方名称
++ SellerRegisterNum	是	array[]	销售方纳税人识别号
++ TotalTax	是	array[]	价税合计
++ AmountInWords	是	array[]	合计金额(大写)
++ AmountInFiguers	是	array[]	合计金额(小写)
++ Payee	是	array[]	收款人
++ CheckCode	是	array[]	校验码。增值税专票无此参数
++ Province	是	array[]	省
++ City	是	array[]	市
+++ word	是	string	识别结果字符串，以上各字段均包含
++ CommodityName	是	array[]	货物名称
++ CommodityNum	是	array[]	数量
++ CommodityPrice	是	array[]	单价
++ CommodityAmount	是	array[]	金额
+++ row	是	uint32	行号，以上各字段均包含
+++ word	是	string	内容，以上各字段均包含

type 的返回结果为 printed_invoice，即“机打发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceType	是	array[]	发票类型
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ InvoiceDate	是	array[]	开票日期
++ AmountInFiguers	是	array[]	合计金额小写
++ AmountInWords	是	array[]	合计金额大写
++ MachineNum	是	array[]	机打号码
++ CheckCode	是	array[]	校验码
++ SellerName	是	array[]	销售方名称
++ SellerRegisterNum	是	array[]	销售方纳税人识别号
++ PurchaserName	是	array[]	购买方名称
++ PurchaserRegisterNum	是	array[]	购买方纳税人识别号
++ TotalTax	是	array[]	合计税额
++ Province	是	array[]	省
++ City	是	array[]	市
++ Time	是	array[]	时间
++ SheetNum	是	array[]	联次
+++ word	是	string	识别结果字符串，以上各字段均包含
++ CommodityName	是	array[]	商品名称
++ CommodityUnit	是	array[]	商品单位
++ CommodityPrice	是	array[]	商品单价
++ CommodityNum	是	array[]	商品数量
++ CommodityAmount	是	array[]	商品金额
+++ row	是	uint32	行号，以上各字段均包含
+++ word	是	string	内容，以上各字段均包含

type 的返回结果为 bus_ticket，即“汽车票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ Date	是	array[]	日期
++ Time	是	array[]	时间
++ ExitStation	是	array[]	出发站
++ Amount	是	array[]	金额
++ IdCard	是	array[]	身份证号
++ ArrivalStation	是	array[]	到达站
++ Name	是	array[]	姓名
++ InvoiceTime	是	array[]	开票日期
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 toll_invoice，即“过路过桥费”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ Entrance	是	array[]	入口
++ Exit	是	array[]	出口
++ OutDate	是	array[]	日期
++ OutTime	是	array[]	时间
++ TotalAmount	是	array[]	金额
++ Province	是	array[]	省
++ City	是	array[]	市
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 ferry_ticket，即“船票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceType	是	array[]	发票类型
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ ExitStation	是	array[]	出发地点
++ ArrivalStation	是	array[]	到达地点
++ Amount	是	array[]	总金额
++ Date	是	array[]	开票日期
++ MoneyType	是	array[]	金额类型
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 motor_vehicle_invoice，即“机动车销售发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ date	是	array[]	开票日期
++ fapiao-daima	是	array[]	发票代码/机打代码
++ fapiao-haoma	是	array[]	发票号码/机打号码
++ printed-daima	是	array[]	机打代码
++ printed-haoma	是	array[]	机打号码
++ machine-num	是	array[]	机器编号
++ buyer-name	是	array[]	购买方名称
++ payer-tax-num	是	array[]	购买方身份证号码/组织机构代码
++ car-class	是	array[]	车辆类型
++ car-model	是	array[]	厂牌型号
++ product-location	是	array[]	产地
++ certificate-num	是	array[]	合格证号
++ engine-num	是	array[]	发动机号码
++ vin-num	是	array[]	车架号码
++ price-tax-big	是	array[]	价税合计
++ price-tax-small	是	array[]	价税合计小写
++ saler	是	array[]	销货单位名称
++ saler-phone	是	array[]	销货单位电话
++ saler-tax-num	是	array[]	销货单位纳税人识别号
++ saler-bank-num	是	array[]	销货单位账号
++ saler-address	是	array[]	销货单位地址
++ saler-bank	是	array[]	销货单位开户银行
++ tax-rate	是	array[]	税率
++ tax	是	array[]	税额
++ tax-jiguan	是	array[]	主管税务机关
++ tax-jiguan-daima	是	array[]	主管税务机关代码
++ price	是	array[]	不含税价格
++ limit-mount	是	array[]	限乘人数
++ toonage	是	array[]	吨位
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 used_vehicle_invoice，即“二手车销售发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ invoice_code	是	array[]	发票代码
++ invoice_num	是	array[]	发票号码
++ date	是	array[]	开票日期
++ tax_code	是	array[]	税控码
++ buyer	是	array[]	买方
++ buyer_id	是	array[]	买方身份证号
++ buyer_station	是	array[]	买方地址
++ buyer_tel	是	array[]	买方电话
++ saler	是	array[]	卖方
++ saler_id	是	array[]	卖方身份证号
++ saler_station	是	array[]	卖方地址
++ saler_tel	是	array[]	卖方电话
++ car_plate	是	array[]	车牌号
++ car_certificate	是	array[]	登记证号
++ car_class	是	array[]	车辆类型
++ vin_num	是	array[]	车架号
++ model	是	array[]	厂牌型号
++ to_station	是	array[]	转入地车管所名称
++ big_price	是	array[]	车价合计大写
++ small_price	是	array[]	车价合计小写
++ car_market	是	array[]	二手车市场
++ tax_num	是	array[]	纳税人识别号
++ tax_location	是	array[]	纳税人地址
++ tax_tel	是	array[]	纳税人电话
++ sheet_num	是	array[]	联次
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 taxi_online_ticket，即“网约车行程单”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ service_provider	是	array[]	服务商
++ start_time	是	array[]	行程开始时间
++ destination_time	是	array[]	行程结束时间
++ phone	是	array[]	行程人手机号
++ application_date	是	array[]	申请日期
++ total_fare	是	array[]	总金额
++ item_num	是	array[]	行程信息中包含的行程数量
+++ word	是	string	识别结果字符串，以上各字段均包含
++ items	是	array[]	行程信息
++++ item_id	是	array[]	行程信息的对应序号
++++ pickup_time	是	array[]	上车时间
++++ pickup_date	是	array[]	上车日期
++++ car_type	是	array[]	车型
++++ distance	是	array[]	里程
++++ start_place	是	array[]	起点
++++ destination_place	是	array[]	终点
++++ city	是	array[]	城市
++++ fare	是	array[]	金额

当验真参数开启（即 verify_parameter=true 时），返回匹配发票验真接口所需的6要素信息

字段	是否必选	类型	说明
++ invoice_code	是	array[]	发票代码
++ invoice_num	是	array[]	发票号码
++ invoice_date	是	array[]	开票日期。返回格式为 YYYYMMDD，例：20210101
++ invoice_type	是	array[]	发票种类。**不同类型发票输出如下结果**： 增值税专用发票：special_vat_invoice 增值税电子专票：elec_special_vat_invoice 增值税普通发票：normal_invoice 增值税普通发票（电子）：elec_normal_invoice 增值税普通发票（卷式）：roll_normal_invoice 通行费增值税电子普通发票： toll_elec_normal_invoice 货运运输业增值税专用发票：special_freight_transport_invoice 机动车销售发票：motor_vehicle_invoice 二手车销售发票：used_vehicle_invoice 区块链发票：blockchain_invoice 通用机打电子发票：printed_elec_invoice
++ total_amount	是	array[]	不含税金额
++ check_code	否	array[]	检验码。**如需使用百度的增值税发票验真接口，需提取返回值的后6位后，再传入验真接口**

返回示例


```
{
  "words_result": [
    {
      "type": "vat_invoice",
      "width": 0,
      "probability": 0.9980429411,
      "height": 649,
      "left": 154,
      "top": 177,
      "result": {
        "AmountInWords": [
          {
            "word": "叁佰陆拾圆整"
          }
        ],
        "InvoiceNumConfirm": [
          {
            "word": "07286261"
          }
        ],
        "CommodityEndDate": [],
        "CommodityVehicleType": [],
        "CommodityStartDate": [],
        "CommodityPrice": [
          {
            "row": "1",
            "word": "339.62"
          }
        ],
```

```
        "NoteDrawer": [
            {
                "word": "余佳燕"
            }
        ],
        "SellerAddress": [],
        "CommodityNum": [
            {
                "row": "1",
                "word": "1"
            }
        ],
        "SellerRegisterNum": [
            {
                "word": "91330106673959654P"
            }
        ],
        "MachineCode": [],
        "Remarks": [],
        "SellerBank": [
            {
                "word": "招商银行杭州高新支行502905023610702"
            }
        ],
        "CommodityTaxRate": [
            {
                "row": "1",
                "word": "6%"
            }
        ],
        "TotalTax": [
            {
                "word": "20.38"
            }
        ],
        "InvoiceCodeConfirm": [
            {
                "word": "3321192130"
            }
        ],
        "CheckCode": [],
        "InvoiceCode": [
            {
                "word": "3321192130"
            }
        ],
        "InvoiceDate": [
            {
                "word": "2019年08月28日"
            }
        ],
        "PurchaserRegisterNum": [
            {
                "word": "91110911717743469K"
            }
        ],
        "InvoiceTypeOrg": [
            {
                "word": "浙江增值税专用发票"
            }
        ],
        "OnlinePay": [],
```

```
            "Password": [
                {
                    "word": "508>3909>1*>01/-46709-6/3+*7+8>/1*19+7-0**>+58290-6>647-
+324865*9*1<*2191/7754/<0>2<838+//5-69--748*<251408<"
                }
            ],
            "Agent": [
                {
                    "word": "否"
                }
            ],
            "AmountInFiguers": [
                {
                    "word": "360.00"
                }
            ],
            "PurchaserBank": [
                {
                    "word": "招商银行北京分行大电路支行866180100210002"
                }
            ],
            "Checker": [
                {
                    "word": "柳余"
                }
            ],
            "City": [],
            "TotalAmount": [
                {
                    "word": "339.62"
                }
            ],
            "CommodityAmount": [
                {
                    "row": "1",
                    "word": "339.62"
                }
            ],
            "PurchaserName": [
                {
                    "word": "百度在线网络技术(北京)有限公司"
                }
            ],
            "CommodityType": [],
            "Province": [
                {
                    "word": "浙江"
                }
            ],
            "InvoiceType": [
                {
                    "word": "专用发票"
                }
            ],
            "SheetNum": [
                {
                    "word": "第二联：抵扣联"
                }
            ],
            "PurchaserAddress": [],
            "CommodityTax": [
                {
                    "row": "1"
```

```
                "row": "1",
                "word": "20.38"
            }
        ],
        "CommodityPlateNum": [],
        "CommodityUnit": [
            {
                "row": "1",
                "word": "套"
            }
        ],
        "Payee": [
            {
                "word": "佳机"
            }
        ],
        "CommodityName": [
            {
                "row": "1",
                "word": "*信息技术服务*软件服务费"
            }
        ],
        "SellerName": [
            {
                "word": "阿里云计算有限公司"
            }
        ],
        "InvoiceNum": [
            {
                "word": "07286261"
            }
        ]
    }
},
{
    "type": "taxi_receipt",
    "width": 0,
    "probability": 0.9858493805,
    "height": 615,
    "left": 1325,
    "top": 200,
    "result": {
        "PickupTime": [
            {
                "word": "10:50"
            }
        ],
        "DropoffTime": [
            {
                "word": "17:06"
            }
        ],
        "Time": [
            {
                "word": "10:50-17:06"
            }
        ],
        "City": [
            {
                "word": ""
            }
        ],
        "FuelOilSurcharge": [
```

```
            {
                "word": "1.00"
            }
        ],
        "Date": [
            {
                "word": "2019-03-20"
            }
        ],
        "Province": [
            {
                "word": "陕西省"
            }
        ],
        "CallServiceSurcharge": [
            {
                "word": "0.00"
            }
        ],
        "Fare": [
            {
                "word": "21.10"
            }
        ],
        "TotalFare": [
            {
                "word": "22.00"
            }
        ],
        "TaxiNum": [
            {
                "word": "AQ6353"
            }
        ],
        "PricePerkm": [
            {
                "word": "2.30"
            }
        ],
        "InvoiceCode": [
            {
                "word": "161001881016"
            }
        ],
        "Distance": [
            {
                "word": "6.0"
            }
        ],
        "InvoiceNum": [
            {
                "word": "05070716"
            }
        ],
        "Location": [
            {
                "word": "陕西省"
            }
        ]
    }
  }
],
```

```
    "words_result_num": 2,
    "log_id": 1438382953545048984
}
```

增值税发票验真

支持 12 种增值税发票的信息核验，包括增值税专票、电子专票、普票、电子普票、卷票、区块链发票（深圳地区）、全电发票（增值税专用发票）、全电发票（普通发票）、通行费增值税电子普通发票、货物运输业增值税专用发票、机动车销售发票、二手车销售发票等，支持返回票面的全部信息。同时可直接与同平台的发票识别能力对接，完成发票识别的同时进行自动化验真。

```
# 调用增值税发票验真
options = {}
options["invoice_code"] = "011002000000"
options["invoice_num"] = "93705563"
options["invoice_date"] = "20210000"
options["invoice_type"] = "elec_normal_invoice"
options["check_code"] = "000000"
options["total_amount"] = "00.00"
result = client.vat_invoice_verification(options)
print(result)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
invoice_code	是	string	-	发票代码。全电发票（专用发票）、全电发票（普通发票）此参数可为空，其他类型发票均不可为空
invoice_num	是	string	-	发票号码
invoice_date	是	string	-	开票日期。格式YYYYMMDD，例：20210101
invoice_type	是	string	增值税专用发票：special_vat_invoice 增值税电子专用发票：elec_special_vat_invoice 增值税普通发票：normal_invoice 增值税普通发票（电子）：elec_normal_invoice 增值税普通发票（卷式）：roll_normal_invoice 通行费增值税电子普通发票：toll_elec_normal_invoice 区块链电子发票（目前仅支持深圳地区）：blockchain_invoice 全电发票（专用发票）：elec_invoice_special 全电发票（普通发票）：elec_invoice_normal 货运运输业增值税专用发票：special_freight_transport_invoice 机动车销售发票：motor_vehicle_invoice 二手车销售发票：used_vehicle_invoice	发票种类
check_code	是	string	-	校验码。填写发票校验码后6位，增值税电子专票、普票、电子普票、卷票、区块链电子发票、通行费增值税电子普通发票此参数不可为空，其他类型发票可为空
total_amount	是	string	-	发票金额。增值税专票、电子专票、货运专票、机动车销售发票填写**不含税金额**； 二手车销售发票填写**车价合计**； 全电发票（专用发票）、全电发票（普通发票）填写**价税合计金额**，其他类型发票可为空

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
VerifyResult	是	string	查验结果。查验成功返回“0001”，查验失败返回对应查验结果错误码，详见末尾表格
VerifyMessage	是	string	查验结果信息。查验成功且发票为真返回“查验成功发票一致“，查验失败返回对应错误原因，详见末尾表格
VerifyFrequency	是	string	查验次数。为历史查验次数
InvalidSign	是	string	是否作废（冲红）。Y：已作废；H：已冲红；N：未作废
InvoiceType	是	string	发票种类
InvoiceCode	是	string	发票代码
InvoiceNum	是	string	发票号码
CheckCode	是	string	校验码
InvoiceDate	是	string	开票日期
MachineCode	是	string	机器编号

增值税专票、电子专票、普票、电子普通发票、卷票、通行费增值税电子普通发票、货物运输业增值税专用发票返回信息

字段	是否必选	类型	说明
+ PurchaserName	是	string	购方名称
+ PurchaserRegisterNum	是	string	购方纳税人识别号
+ PurchaserAddress	是	string	购方地址及电话
+ PurchaserBank	是	string	购方开户行及账号
+ CommodityName	是	array[]	货物名称/项目名称
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityType	是	array[]	规格型号
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityUnit	是	array[]	单位

++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityNum	是	array[]	数量
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityPrice	是	array[]	单价
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityAmount	是	array[]	金额
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTaxRate	是	array[]	税率
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTax	是	array[]	税额
++ row	是	uint32	行号
++ word	是	string	内容
+ SellerName	是	string	销售方名称
+ SellerRegisterNum	是	string	销售方纳税人识别号
+ SellerAddress	是	string	销售方地址及电话
+ SellerBank	是	string	销售方开户行及账号
+ TotalAmount	是	string	合计金额
+ TotalTax	是	string	合计税额
+ AmountInFiguers	是	string	价税合计（小写）
+ TollSign	是	string	通行费标志。Y-可抵扣通行费，N-不可抵扣通行费。通行费增值税电子普通发票返回信息，其他类型发票可忽略
+ ZeroTaxRateIndicator	是	string	零税率标识。空：非零税率,1：税率栏位显示“免税”，2：税率栏位显示“不征税”，3：零税率。通行费增值税电子普通发票返回信息，其他类型发票可忽略
+ CommodityPlateNum	是	array[]	车牌号。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号

++ word	是	string	内容
+ CommodityVehicleType	是	array[]	类型。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityStartDate	是	array[]	通行日期起。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityEndDate	是	array[]	通行日期止。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ Carrier	是	string	承运人名称。货运专票返回信息，其他类型发票可忽略
+ CarrierCode	是	string	承运人识别号。货运专票返回信息，其他类型发票可忽略
+ Recipient	是	string	受票方名称。货运专票返回信息，其他类型发票可忽略
+ RecipientCode	是	string	受票方识别号。货运专票返回信息，其他类型发票可忽略
+ Receiver	是	string	收货人名称。货运专票返回信息，其他类型发票可忽略
+ ReceiverCode	是	string	收货人识别号。货运专票返回信息，其他类型发票可忽略
+ Sender	是	string	发货人名称。货运专票返回信息，其他类型发票可忽略
+ SenderCode	是	string	发货人识别号。货运专票返回信息，其他类型发票可忽略
+ TransportCargoInformation	是	string	运输货物信息。货运专票返回信息，其他类型发票可忽略
+ DepartureViaArrival	是	string	起运地、经由、到达地。货运专票返回信息，其他类型发票可忽略
+ TaxControlNum	是	string	税控盘号。货运专票返回信息，其他类型发票可忽略
+ VehicleType	是	string	车种车号。货运专票返回信息，其他类型发票可忽略
+ VehicleTonnage	是	string	车船吨位。货运专票返回信息，其他类型发票可忽略
+ CommodityExpenseItem	是	array[]	费用项目。货运专票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ NoteDrawer	是	string	开票人
+ Checker	是	string	复核
+ Payee	是	string	收款人

+ Remarks	是	string	备注
+ ESVATURL	是	string	增值税电子专票（即 ofd 发票）的下载地址
+ ListLabel	是	string	清单标识，Y：带清单；N：无清单； 说明：只有当发票种类为：增值税专票，电子专票，普票，电子普通发票时返回此字段的值

机动车销售发票返回信息

字段	是否必选	类型	说明
+ Purchaser	是	string	购买方名称
+ PurchaserCode	是	string	购买方身份证号/组织机构代码
+ VehicleType	是	string	车辆类型
+ ManuModel	是	string	厂牌型号
+ Origin	是	string	产地
+ CertificateNum	是	string	合格证号书
+ CommodityInspectionNum	是	string	商检单号
+ EngineNum	是	string	发动机号码
+ VinNum	是	string	车辆识别代号/车架号码
+ ImportCertificateNum	是	string	进口证明书号
+ TaxPaymentVoucherNum	是	string	完税凭证号码
+ LimitPassenger	是	string	限乘人数
+ TaxAuthor	是	string	主管税务机关名称
+ TaxAuthorCode	是	string	主管税务机关代码
+ Tonnage	是	string	吨位
+ Price	是	string	不含税价格
+ TaxRate	是	string	税率
+ Tax	是	string	税额
+ PriceTaxLow	是	string	价税合计
+ Saler	是	string	销货单位名称
+ SalerCode	是	string	销货单位纳税人识别号
+ SalerBank	是	string	销货单位开户银行
+ SalerAccountNum	是	string	销货单位账号
+ SalerPhone	是	string	销货单位电话

二手车销售发票返回信息

字段	是否必选	类型	说明
+ Purchaser	是	string	买方单位/个人
+ PurchaserCode	是	string	买方单位代码/身份证号
+ PurchaserAddress	是	string	买方单位/个人住址
+ PurchaserPhone	是	string	买方电话
+ Saler	是	string	卖方单位/个人
+ SalerCode	是	string	卖方单位代码/身份证号
+ SalerAddress	是	string	卖方单位/个人住址
+ SalerPhone	是	string	卖方电话
+ LicensePlateNum	是	string	车牌照号
+ RegistrationCode	是	string	登记证号
+ TotalCarPrice	是	string	车价合计
+ TransferVehicleManagementOffice	是	string	转入地车辆车管所名称
+ VehicleType	是	string	车辆类型
+ ManuModel	是	string	厂牌型号
+ VinNum	是	string	车辆识别代号/车架号码
+ Operator	是	string	经营、拍卖单位
+ OperatorAddress	是	string	经营、拍卖单位地址
+ OperatorCode	是	string	经营、拍卖单位纳税人识别号
+ OperatorBank	是	string	开户银行及账号
+ OperatorPhone	是	string	经营、拍卖单位电话
+ UsedCarMarket	是	string	二手车市场
+ UsedCarMarketCode	是	string	二手车市场纳税人识别号
+ UsedCarMarketAddress	是	string	二手车市地址
+ UsedCarMarketBank	是	string	二手车市场开户银行及账号
+ UsedCarMarketPhone	是	string	二手车市场电话

查验结果错误码

查验结果（VerifyResult）	查验结果信息（VerifyMessage）	描述
9999	查验失败	查验失败，业务出现异常，请提交工单咨询
0002	超过该张票当天查验次数	此发票今日查询次数已达上限（5次），请次日查询
0005	请求不合法	发票信息有误，请核对后再查询
0006	发票信息不一致	发票信息有误，请核对后再查询
0009	发票不存在	所查发票不存在
1004	已超过最大查验量	已超过最大查验量，请提交工单咨询
1005	查询发票不规范	信息有误，请核对后再查询
1006	查验异常	发票信息有误，请核对后再查询
1007	该批次已过期，请重新更换批次号查验	该批次已过期，请重新更换批次查验
1008	字段不能为空	发票请求参数不能为空
1009	参数长度不正确	参数长度不符合规范，确认参数，再次查验
1014	日期当天的不能查验	日期当天的不能查验，请隔天再查
1015	超过5年的不能查验	超过5年的不能查验
1020	没有查验权限	没有查验权限，请提交工单咨询
1021	网络超时	税局维护升级，暂时无法查验，请提交工单咨询

返回示例

```
 // 增值税专票、电子专票、普票、电子普通发票、卷票、通行费增值税电子普通发票、货物运输业增值税专用发票
{
    "words_result": {
        "log_id": 1394226734160674816,
        "words_result_num": 43,
        "VerifyFrequency": "3",
        "VerifyMessage": "查验成功发票一致",
        "InvalidSign": "N",
        "InvoiceType": "增值税普通发票（电子）",
        "MachineCode": "661616300747",
        "CheckCode": "67820461013285253079",
        "InvoiceCode": "043002000111",
        "InvoiceDate": "20210503",
        "VerifyResult": "0001",
        "InvoiceNum": "63509760"
        "TaxControlNum": "",
        "CommodityEndDate": [
            {
                "row": "1",
                "word": ""
            }
        ],
        "VehicleTonnage": "",
        "CommodityVehicleType": [
            {
                "row": "1"
            }
        ],
        "CommodityStartDate": [
            {
                "row": "1",
                "word": ""
            }
        ],
        "SellerAddress": "湖南省长沙市天心区芙蓉中路三段446号0731-83592079",
```

```
        "CommodityPrice": [
          {
            "row": "1",
            "word": "28.20000000"
          }
        ],
        "TransportCargoInformation": "",
        "NoteDrawer": "",
        "CommodityNum": [
          {
            "row": "1",
            "word": "1.00000000"
          }
        ],
        "SellerRegisterNum": "914301007121984812",
        "SellerBank": "建行长沙铁银支行营业部43001710661050003739",
        "Remarks": "账期:202104",
        "TotalTax": "0.00",
        "CommodityTaxRate": [
          {
            "row": "1",
            "word": "不征税"
          }
        ],
        "CommodityExpenseItem": [
          {
            "row": "1",
            "word": ""
          }
        ],
        "ZeroTaxRateIndicator": "",
        "Carrier": "",
        "SenderCode": "",
        "PurchaserRegisterNum": "911101087877515792",
        "ReceiverCode": "",
        "AmountInFiguers": "28.20",
        "PurchaserBank": "招商银行北京分行大屯路支行 866182028510003",
        "Checker": "",
        "TollSign": "",
        "VehicleTypeNum": "",
        "DepartureViaArrival": "",
        "Receiver": "",
        "Recipient": "",
        "TotalAmount": "28.20",
        "CommodityAmount": [
          {
            "row": "1",
            "word": "28.20"
          }
        ],
        "PurchaserName": "百度时代网络技术（北京）有限公司",
        "CommodityType": [
          {
            "row": "1",
            "word": ""
          }
        ],
        "Sender": "",
        "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二层A201059108001",
        "CommodityTax": [
          {
            "row": "1",
```

```
            "word": "***"
          }
        ],
        "CarrierCode": "",
        "CommodityPlateNum": [
          {
            "row": "1",
            "word": ""
          }
        ],
        "CommodityUnit": [
          {
            "row": "1",
            "word": ""
          }
        ],
        "Payee": "",
        "RecipientCode": "",
        "CommodityName": [
          {
            "row": "1",
            "word": "*电信服务*通讯费服务费"
          }
        ],
        "SellerName": "中国移动通信集团湖南有限公司长沙分公司"
    },
}
// 机动车销售发票
{
    "words_result": {
        "log_id": 1394232842988290048,
        "words_result_num": 24,
        "VerifyFrequency": "1",
        "VerifyMessage": "查验成功发票一致",
        "InvalidSign": "N",
        "InvoiceType": "机动车销售统一发票",
        "MachineCode": "539927983",
        "CheckCode": "",
        "InvoiceCode": "13200378019836",
        "InvoiceDate": "20210128",
        "VerifyResult": "0001",
        "InvoiceNum": "00342061"
        "Origin": "中国",
        "ManuModel": "东风日产牌DFL8",
        "SalerBank": "工行支行",
        "VehicleType": "多用途乘用车",
        "Tax": "18238.29",
        "TaxPaymentVoucherNum": "",
        "CommodityInspectionNum": "",
        "TaxAuthorCode": "1332803841100",
        "VinNum": "LGBM464574",
        "SalerPhone": "0513-8237861",
        "LimitPassenger": "5",
        "PurchaserCode": "211402199410176136",
        "TaxAuthor": "国家税务总局海门市税务局三厂税务分局",
        "Tonnage": "",
        "ImportCertificateNum": "",
        "Saler": "海门市海通汽车销售服务有限公司",
        "SalerAccountNum": "1111527109002888833",
        "Price": "145840.71",
        "CertificateNum": "WAC224003769810",
        "TaxRate": "13%",
```

```
        "Purchaser": "郑如意",
        "SalerCode": "9132068478280000007164",
        "EngineNum": "43380M",
        "PriceTaxLow": "1323800"
    },
}
// 二手车销售发票
{
    "words_result": {
        "log_id": 1394233936539811840,
        "words_result_num": 25,
        "VerifyFrequency": "1",
        "VerifyMessage": "查验成功发票一致",
        "InvalidSign": "N",
        "InvoiceType": "二手车销售统一发票",
        "MachineCode": "66173004789204",
        "CheckCode": "",
        "InvoiceCode": "0323789200007",
        "InvoiceDate": "20200509",
        "VerifyResult": "0001",
        "InvoiceNum": "002890341"
        "Operator": "",
        "TransferVehicleManagementOffice": "苏州市车管所",
        "ManuModel": "JF1SH95F",
        "RegistrationCode": "3200478903518",
        "OperatorPhone": "",
        "PurchaserCode": "320503782902308u425",
        "Saler": "张散文",
        "UsedCarMarketCode": "91320378038NCQUQXA",
        "Purchaser": "张丽",
        "OperatorCode": "",
        "UsedCarMarketBank": "中国农业银行股份有限公司苏州分行清算中心10549001040001493",
        "SalerAddress": "江苏省苏州市工业园区倪浜路3号",
        "SalerCode": "411524199001016511",
        "PurchaserPhone": "0",
        "LicensePlateNum": "苏U1A666",
        "VehicleType": "小型越野客车",
        "OperatorBank": "",
        "OperatorAddress": "",
        "VinNum": "JF1SH78006596636",
        "TotalCarPrice": "66000.00",
        "SalerPhone": "",
        "PurchaserAddress": "江苏省苏州市相城区元和莫阳村",
        "UsedCarMarketPhone": "13182680222",
        "UsedCarMarketAddress": "苏州高新区长江路668号(3号厂房)",
        "UsedCarMarket": "苏州车市界二手车电子商务有限公司"
    },
}
```


医疗发票识别

支持识别全国各地门诊/住院发票的 业务流水号、发票号、住院号、门诊号、病例号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付 等关键字段，其中北京/广东/河北/河南/江苏/山东/上海/天津/浙江等地区票据识别效果较佳。支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用医疗发票识别
res_image = client.medicalInvoice(image)
res_url = client.medicalInvoiceUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options = {}
options["location"] = "true"
options["probability"] = "true"
res_image = client.medicalInvoice(image, options)
res_url = client.medicalInvoiceUrl(url, options)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	string	true/false	是否返回字段的位置信息，**默认为 false，可缺省** - **false**：不返回字段位置信息 - **true**：返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	string	true/false	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
InvoiceType	是	string	票据种类
Province	是	string	省市：支持返回以下省市 北京/广东/河北/河南/江苏/山东/上海/天津/浙江等
words_result	是	object	识别结果
+ BusinessNum	是	object	业务流水号

+ InvoiceNum	是	object	发票号码
+ HospitalNum	是	object	住院号
+ HospitalName	是	object	医院名称
+ RecordNum	是	object	病例号
+ HospitalDay	是	object	住院天数
+ AdmissionDate	是	object	入院时间
+ DischargeDate	是	object	出院时间
+ Name	是	object	姓名
+ Sex	是	object	性别
+ HospitalType	是	object	医疗机构类型
+ SocialSecurityNum	是	object	社保卡号
+ InsuranceType	是	object	医保类型
+ ChargingUnit	是	object	收款单位
+ Payee	是	object	收款人
+ Date	是	object	开票日期
+ AmountInWords	是	object	大写合计金额
+ AmountInFiguers	是	object	小写合计金额
+ InsurancePayment	是	object	医保统筹支付
+ PersonalPayment	是	object	个人账户支付
+ PrepayAmount	是	object	预缴金额
+ PaymentAmount	是	object	补缴金额
+ RefundAmount	是	object	退费金额
+ ClinicNum	是	object	门诊号
++ word	是	string	字段识别结果，以上各字段均包含此参数
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+ CostCategories	是	array[]	项目大类：治疗费、检查费等项目大类
+ CostDetail	是	array[]	明细类别：药物/检查的明细类别
+ RegionSupplement	是	array[]	地区字段：根据省市返回改地区特有的字段
CostCategories字段包含多个array，每个数组包含多个object，见以下参数			

字段	说明
++ name	字段名，包括：收费项目、金额
++ word	name字段对应的识别结果
CostDetail字段包含多个array，每个数组包含多个object，见以下参数	

字段	说明
++ name	字段名，包括：编码、项目、规格、数量、单价、金额
++ word	name字段对应的识别结果
RegionSupplement字段包含多个object，不同省市返回字段不同，见以下参数	

省市	返回参数（name）
北京	个人支付金额、其他医保支付、交易流水号、基金支付、单位补充险[原公疗]支付、年度门诊大额累计支付、本次医保范围内金额、本次支付后个人账户余额、残军补助支付、累计医保内范围金额、自付一、自付二、自费、起付金额、超封顶金额、退休补充支付、门诊大额支付
广东	个人支付金额、其他医保支付
河北	个人账户余额、统筹累计支付、自负、自费、起付标准
河南	个人支付金额、其他医保支付
江苏	个人支付金额、其他医保支付
山东	个人支付金额、其他医保支付
上海	分类自负、历年余额、本年余额、现金支付、自负、自费、附加支付
天津	个人支付金额、其他医保支付
浙江	历年余额、历年支付、基金支付、本年余额、本年支付、现金支付

返回示例


```
{
  "log_id": 1397076899313745920,
  "words_result_num": 28,
  "Province": "北京",
  "InvoiceType": "门诊发票"
  "words_result": {
     "AmountInWords": {
        "word": "玖佰叁拾玖元肆角捌分"
     },
     "Sex": {
        "word": "男"
     },
     "InsuranceType": {
        "word": "城镇职工"
     },
     "Name": {
        "word": "王成"
     },
     "SocialSecurityNum": {
        "word": "T03419700077"
     },
     "DischargeDate": {
        "word": "2016-01-01"
     },
     "HospitalNum": {
        "word": "0937829032"
     }
```

```
    },
    "HospitalName": {
      "word": "北京市房山区良乡医院"
    },
    "CostCategories": [
      [
        {
          "name": "收费项目",
          "word": "西药费"
        },
        {
          "name": "金额",
          "word": "426.70"
        }
      ],
      [
        {
          "name": "收费项目",
          "word": "中成药费"
        },
        {
          "name": "金额",
          "word": "512.78"
        }
      ]
    ],
    "RegionSupplement": [
      {
        "name": "其他医保支付",
        "word": "0.00"
      },
      {
        "name": "年度门诊大额累计支付",
        "word": "83.79"
      },
      {
        "name": "起付金额",
        "word": "777.11"
      },
      {
        "name": "基金支付",
        "word": "101.75"
      },
      {
        "name": "本次支付后个人账户余额",
        "word": "0.00"
      },
      {
        "name": "个人支付金额",
        "word": "837.73"
      },
      {
        "name": "交易流水号",
        "word": "111100030Z160517006328"
      },
      {
        "name": "自付一",
        "word": "795.06"
      },
      {
        "name": "自付二",
        "word": "42.67"
      },
```

```
        {
            "name": "累计医保内范围金额",
            "word": "1419.70"
        },
        {
            "name": "门诊大额支付",
            "word": "83.79"
        },
        {
            "name": "本次医保范围内金额",
            "word": "896.81"
        },
        {
            "name": "退休补充支付",
            "word": "17.96"
        },
        {
            "name": "超封顶金额",
            "word": "0.00"
        },
        {
            "name": "残军补助支付",
            "word": "0.00"
        },
        {
            "name": "单位补充险[原公疗]支付",
            "word": "0.00"
        },
        {
            "name": "自费",
            "word": "42.67"
        }
    ],
    "ClinicNum": {
        "word": "12169298"
    },
    "AmountInFiguers": {
        "word": "939.48"
    },
    "AdmissionDate": {
        "word": "2016-01-01"
    },
    "HospitalType": {
        "word": "综合医院"
    },
    "RefundAmount": {
        "word": "0.00"
    },
    "Date": {
        "word": "2016年05月17日"
    },
    "ChargingUnit": {
        "word": "房山区良乡医院"
    },
    "CostDetail": [
        [
            {
                "name": "编码",
                "word": "01"
            },
            {
                "name": "项目",
```

```
                "word": "阿托伐他汀钙胶囊"
            },
            {
                "name": "规格",
                "word": "10mg*7支"
            },
            {
                "name": "数量",
                "word": "10"
            },
            {
                "name": "单价",
                "word": "29.3200"
            },
            {
                "name": "金额",
                "word": "293.20"
            }
        ],
        [
            {
                "name": "编码",
                "word": "02"
            },
            {
                "name": "项目",
                "word": "替米沙坦胶囊"
            },
            {
                "name": "规格",
                "word": "40mg*12粒"
            },
            {
                "name": "数量",
                "word": "5"
            },
            {
                "name": "单价",
                "word": "26.7000"
            },
            {
                "name": "金额",
                "word": "133.50"
            }
        ],
    ],
    "PaymentAmount": {
        "word": "0.00"
    },
    "PrepayAmount": {
        "word": "0.00"
    },
    "PersonalPayment": {
        "word": "0.00"
    },
    "HospitalDay": {
        "word": "15"
    },
    "BusinessNum": {
        "word": "40091198916051710196"
    },
    "InsurancePayment": {
```

```
            "word": "0.00"
        },
        "Payee": {
            "word": "刘冰"
        },
        "RecordNum": {
            "word": "0001268129"
        },
        "InvoiceNum": {
            "word": "0103152099"
        }
    },
}
```

门脸文字识别

针对含有门脸/门头的图片进行专项优化，支持识别门脸/门头上的文字内容。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')

# 调用门脸文字识别
res_image = client.facade(image)
print(res_image)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，base64编码去除编码头（data:image/jpeg;base64），要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	否	array[]	定位和识别结果数组
+ words	否	string	识别结果字符串
+ score	否	float	words返回为主门脸名称的置信度评分
+ brief	否	string	门脸副标题等周边描述

返回示例

```
{
  "log_id": 341559937361307312,
  "words_result_num": 3,
  "words_result": [
        {
        "words": "生活超市",
        "brief": "家得利",
        "score": 0.80533
        },
        {
        "words": "火火火火锅",
        "score": 0.0112433
        },
        {
        "words": "天天好便利店",
        "score': 0.188124
        }
  ]
}
```


车辆证照混贴识别

支持自动检测与识别行驶证、驾驶证混贴图片，即识别机动车行驶证主页及副页、机动车驾驶证主页及副页在同一张图片上的场景，一次性识别图片中多个行驶证、驾驶证的所有字段。

支持对机动车行驶证主页及副页所有22个字段进行结构化识别，包括号牌号码、车辆类型、所有人、品牌型号、车辆识别代码、发动机号码、核定载人数、质量、尺寸、检验记录等；支持对机动车驾驶证正页及副页所有15个字段进行结构化识别，包括证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效期限、发证单位、档案编号等。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用车辆证照混贴识别
res_image = client.mixed_multi_vehicle(image)
res_url = client.mixed_multi_vehicleUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options = {}
options["detect_direction"] = "true"
options["unified"] = "true"
res_image = client.mixed_multi_vehicle(image, options)
res_url = client.mixed_multi_vehicleUrl(url, options)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	否	string	true/false	- **false：默认值**不进行图像方向自动矫正 - **true:** 开启图像方向自动矫正功能，可对旋转 90/180/270 度的图片进行自动矫正并识别
unified	否	string	true/false	- **false：默认值**，不进行归一化处理 - **true：**对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出

返回参数详情

字段	必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
card_type	是	string	证件类型，vehicle_front、vehicle_back、driving_front和driving_back，分别对应行驶证正、副页，驾驶证正、副页
direction	否	int32	图像方向，当图像旋转时，返回该参数。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
probability	是	uint32	检测到证件的置信度
location	是	object	证件位置数组（坐标0点为左上角）
+ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+ width	是	uint32	表示定位位置的长方形的宽度
+ height	是	uint32	表示定位位置的长方形的高度
license_info	是	object	识别结果信息，包含图片中多个行驶证、驾驶证的识别结果
+ word_name	是	string	字段名，如果license_info对应行驶证，字段名包含号牌号码、车辆类型、所有人、品牌型号、车辆识别代码、发动机号码、核定载人数、质量、尺寸、检验记录等；如果license_info对应驾驶证，字段名包含证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效期限、发证单位、档案编号等
+ word	是	string	word_name字段对应的识别结果

返回示例


```
{
  "words_result":[
     {
        "license_info":[
           {
              "word_name":"证号",
              "word":"320621196512031267"
           },
           {
              "word_name":"姓名",
              "word":"程成"
           },
           {
              "word_name":"性别",
              "word":"男"
           },
           {
              "word_name":"国籍",
```

```
            "word_name":"国籍",
            "word":"中国"
         },
         {
            "word_name":"住址",
            "word":"江苏省南通市"
         },
         {
            "word_name":"出生日期",
            "word":"19651203"
         },
         {
            "word_name":"初次领证日期",
            "word":"20110311"
         },
         {
            "word_name":"准驾车型",
            "word":"B2"
         },
         {
            "word_name":"有效起始日期",
            "word":"20170311"
         },
         {
            "word_name":"失效日期",
            "word":"20270311"
         },
         {
            "word_name":"发证单位",
            "word":"江苏省南通市公安局交通警察支队"
         }
      ],
      "probability":0.99496907,
      "location":{
         "top":14,
         "left":15,
         "width":701,
         "height":459
      },
      "card_type":"driving_front",
      "direction":0
   },
   {
      "license_info":[
         {
            "word_name":"证号",
            "word":"320621196512031267"
         },
         {
            "word_name":"姓名",
            "word":"程成"
         },
         {
            "word_name":"档案编号",
            "word":"320601650706"
         },
         {
            "word_name":"记录",
            "word":"自2017年03月06日至有效起始日期有效。请于每个记分周期结束后三十日接受审验。无记分的，免予本次审验。"
         }
      ],
      "probability":0.9815229177,
```

```
        "location":{
            "top":7,
            "left":731,
            "width":650,
            "height":456
        },
        "card_type":"driving_back",
        "direction":0
    },
    {
        "license_info":[
            {
                "word_name":"号牌号码",
                "word":"京A10D08"
            },
            {
                "word_name":"车辆类型",
                "word":"小型面包车"
            },
            {
                "word_name":"所有人",
                "word":"王京"
            },
            {
                "word_name":"住址",
                "word":"北京市石景山区"
            },
            {
                "word_name":"使用性质",
                "word":"非营运"
            },
            {
                "word_name":"品牌型号",
                "word":"东风牌EQ6456PF"
            },
            {
                "word_name":"车辆识别代号",
                "word":"LGK022K69A9240616"
            },
            {
                "word_name":"发动机号码",
                "word":"10066180"
            },
            {
                "word_name":"注册日期",
                "word":"20100707"
            },
            {
                "word_name":"发证日期",
                "word":"20191020"
            },
            {
                "word_name":"发证单位",
                "word":"北京市公安局交通警察支队"
            }
        ],
        "probability":0.9907341003,
        "location":{
            "top":532,
            "left":736,
            "width":622,
            "height":415
```

```
        },
        "card_type":"vehicle_front",
        "direction":0
    },
    {
        "license_info":[
            {
                "word_name":"号牌号码",
                "word":"京A10D08"
            },
            {
                "word_name":"备注",
                "word":""
            },
            {
                "word_name":"整备质量",
                "word":"985kg"
            },
            {
                "word_name":"核定载质量",
                "word":""
            },
            {
                "word_name":"外廓尺寸",
                "word":"3660X1560X1925mm"
            },
            {
                "word_name":"核定载人数",
                "word":"7人"
            },
            {
                "word_name":"总质量",
                "word":"1565kg"
            },
            {
                "word_name":"准牵引总质量",
                "word":""
            },
            {
                "word_name":"档案编号",
                "word":""
            },
            {
                "word_name":"检验记录",
                "word":"2020年07月京A(03)"
            },
            {
                "word_name":"燃油类型",
                "word":"汽油"
            }
        ],
        "probability":0.9925737381,
        "location":{
            "top":554,
            "left":12,
            "width":609,
            "height":416
        },
        "card_type":"vehicle_back",
        "direction":0
    }
],
"log_id":1420339917916065252
```

```
    "log_id":1420339917916065252,
    "words_result_num":4
}
```

公式识别

支持对试卷中的数学公式及题目内容进行识别，可提取公式部分进行单独识别，也可对题目和公式进行混合识别，并返回Latex格式公式内容及位置信息，便于进行后续处理。

```
""" 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"

# 调用公式识别
res_image = client.formula(image)
res_url = client.formulaUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options = {}
options["recognize_granularity"] = "big"
options["detect_direction"] = "true"
options["disp_formula"] = "true"
res_image = client.formula(image, options)
res_url = client.formulaUrl(url, options)
print(res_image)
print(res_url)
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置；small：定位单字符位置。默认值为big
detect_direction	否	bool	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
disp_formula	否	bool	true/false	是否分离输出公式识别结果，在words_result外单独输出公式结果，展示在“formula_result”中

返回参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - - 1：未定义， - 0：正向， - 1： 逆时针90度， - 2：逆时针180度， - 3：逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
formula_result	否	bool	是否返回单独公式识别结果，默认false
formula_result_num	否	bool	识别结果中的公式个数，表示formula_result的元素个数
words_result	是	array[]	识别结果数组
+ location	是	object{}	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串

返回示例


```
{
  "log_id": 2671713289176456793,
  "direction": 0,
 "formula_result_num": 3,
  "formula_result": [
     {
        "location": {
           "width": 258,
           "top": 265,
           "left": 450,
           "height": 204
        },
        "words": "\\left\\{ \\begin{aligned} & x = - 1 1 \\\\ & y = 2 \\\\ \\end{aligned} \\right. "
     },
     {
        "location": {
           "width": 429,
           "top": 546,
           "left": 310,
           "height": 203
        },
        "words": "\\left\\{ \\begin{aligned} & 3 x + 2 y = m \\\\ & n x - y = 2 \\\\ \\end{aligned} \\right. "
     },
     {
        "location": {
           "width": 142,
           "top": 613,
           "left": 1029,
           "height": 71
        },
        "words": "m - \\left[ 1 0 0 , - \\infty \\right) "
     }
  ],
```

```
    "words_result_num": 5,
    "words_result": [
        {
            "location": {
                "width": 168,
                "top": 313,
                "left": 292,
                "height": 110
            },
            "words": "已知"
        },
        {
            "location": {
                "width": 258,
                "top": 265,
                "left": 450,
                "height": 204
            },
            "words": "\\left\\{ \\begin{aligned} & x = - 1 1 \\\\ & y = 2 \\\\ \\end{aligned} \\right. "
        },
        {
            "location": {
                "width": 582,
                "top": 319,
                "left": 728,
                "height": 84
            },
            "words": "是二元一次方程组"
        },
        {
            "location": {
                "width": 429,
                "top": 546,
                "left": 310,
                "height": 203
            },
            "words": "\\left\\{ \\begin{aligned} & 3 x + 2 y = m \\\\ & n x - y = 2 \\\\ \\end{aligned} \\right."
        },
        {
            "location": {
                "width": 780,
                "top": 597,
                "left": 745,
                "height": 88
            },
            "words": "的解,则 m - \\left[ 1 0 0 , - \\infty \\right) 的值是()"
        }
    ]
}
```


图文转换器（接口版）--提交请求

图文转换器对应的接口版产品，可识别图片/PDF文件中的文本内容，进行智能版式分析，并转换为保留原文档版式的Word、Excel文档，返回文档下载连接，支持含表格、印章、手写等内容的文档。满足文档版式还原、企业档案电子化等信息管理需求。如需直接在线使用轻应用，可到控制台-图文转换器使用。

```
    """ 读取文件 """
def get_file_content(filePath):
    with open(filePath, "rb") as fp:
        return fp.read()

image = get_file_content('文件路径')
url = "https://www.x.com/sample.jpg"
pdf_file = get_file_content('文件路径')

# 图文转换器（接口版）--提交请求
res_image = client.docConvertRequestV1(image)
res_url = client.docConvertRequestV1Url(url)
res_pdf = client.docConvertRequestV1Pdf(pdf_file)
print(res_image)
print(res_url)
print(res_pdf)
```

请求参数详情

参数	是否必选	类型	说明
image	和 url/pdf_file 三选一	string	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **优先级：** image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **优先级：** image > url > pdf_file，当image字段存在时url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不不超过10M **优先级：** image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容；**若不传入，默认识别文件所有页，页码从1开始**

返回参数详情

字段	类型	说明
success	bool	当前请求状态； true 表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该请求生成的task_id，后续使用该task_id获取识别结果
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": 12345,
  "result":{
     "task_id":"task-xxxxxxx",
  },
  "code":1001,
  "message": "Create task successfully!"
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
  "success":false,
  "log_id": 12345,
  "error_code": 216401,
  "error_msg": "Create task failed!"
}
```

图文转换器（接口版）--获取结果

```
task_id = "xxxxxx"

# 调用图文转换器（接口版）--获取结果
result = client.docConvertResultV1(task_id)
print(result)
```

请求参数详情

参数	是否必选	类型	说明
task_id	是	string	发送提交请求时返回的task_id

返回参数详情

字段	类型	说明
success	bool	当前请求状态； true表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该文件对应请求的task_id
+ ret_code	int	识别状态，1：任务未开始；2：进行中；3：已完成
+ ret_msg	string	识别状态信息：任务未开始；进行中；已完成
+ percent	int	文档转换进度（百分比）
+ result_data	dict	识别结果字符串，返回word、excel的文件分别的下载地址
+ +word	string	还原后的word文件的下载地址，**文件识别失败时返回""**
+ +excel	string	还原后的Excel文件的下载地址，**若文档中没有表格则返回""**
+ create_time	datetime	任务创建时间
+ start_time	datetime	任务开始时间
+ end_time	datetime	任务结束时间
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": "xxxxxx",
  "result":{
     "task_id":"task-xxxxxxx",
     "ret_code": 3,
     "ret_msg": "已完成",
     "percent": 100,
     "result_data": {
        "word": "word_download_url",
        "excel": "",
     },
     "create_time": "2023-01-17 11:06:12",
     "start_time": "2023-01-17 11:06:13",
     "end_time": "2023-01-17 11:06:15"
  },
  "code":1001,
  "message": "Query task successfully!""
}
```

若查询的task_id不存在，返回result为{}。 请求失败响应体示例如下：

```
{
  "code":1001,

  "log_id":1635891796603052032,

  "message":"Query task successfully!",

  "result":{},

  "success":true
}
```

表格文字识别（同步接口）

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别（同步接口）。

表格文字识别(异步接口)--提交请求

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别(异步接口)--提交请求。

表格文字识别(异步接口)--获取结果

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别(异步接口)--获取结果。

表格识别接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别接口。

错误信息

错误返回格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

错误码

错误码	错误信息	描述
4	Open api request limit	集群超限额

	request limit reached	某群超限额
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天流量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
282000	internal error	服务器内部错误，如果您使用的是高精度接口，报这个错误码的原因可能是您上传的图片中文字过多，识别超时导致的，建议您对图片进行切割后再识别，其他情况请再次请求， 如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096 px，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整

216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282110	urls not exit	URL参数不存在，请核对URL后再次提交
282111	url format illegal	URL格式非法，请检查url格式是否符合相应接口的入参要求
282112	url download timeout	url下载超时，请检查url对应的图床/图片无法下载或链路状况不好，您可以重新尝试以下，如果多次尝试后仍不行，建议更换图片地址
282113	url response invalid	URL返回无效参数
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误

Java语言

简介

Hi，您好，欢迎使用百度文字识别服务。

本文档主要针对Java开发者，描述百度文字识别接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 在百度智能云控制台内提交工单，咨询问题类型请选择**人工智能服务**；
- 如有疑问，进入AI社区交流：http://ai.baidu.com/forum/topic/list/164

接口能力

接口名称	接口能力简要描述
通用文字识别	识别图片中的文字信息
通用文字识别（高精度版）	更高精度地识别图片中的文字信息
通用文字识别（含位置信息版）	识别图片中的文字信息（包含文字区域的坐标信息）

通用文字识别（高精度含位置版）	更高精度地识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（含生僻字版）	识别图片中的文字信息（包含对常见字和生僻字的识别）
网络图片文字识别	识别一些网络上背景复杂，特殊字体的文字
网络图片文字识别（含位置版）	识别网络图片中的文字内容（包含文字区域的坐标信息）
身份证识别	识别身份证正反面的文字信息
银行卡识别	识别银行卡的卡号并返回发卡行和卡片性质信息
驾驶证识别	识别机动车驾驶证所有关键字段
行驶证识别	识别机动车行驶证所有关键字段
车牌识别	识别中国大陆各类机动车车牌信息
营业执照识别	对营业执照进行识别
表格文字识别	自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容
通用票据识别	对各类票据图片（医疗票据，保险保单等）进行文字识别，并返回文字在图片中的位置信息
增值税发票识别	对增值税发票进行文字识别，并结构化返回字段信息，支持增值税专票、普票、电子发票
出租车票识别	针对全国各大城市出租车票的发票号码、发票代码、车号、日期、时间、金额等进行结构化识别
VIN码识别	对车辆车架、挡风玻璃上的VIN码进行识别
火车票识别	支持对大陆火车票的车票号、始发站、目的站、车次、日期、票价、席别、姓名进行结构化识别
飞机行程单识别	支持对飞机行程单的24个字段进行结构化识别
二维码识别	对图片中的二维码、条形码进行检测和识别，返回存储的文字信息
数字识别	识别图片中的数字，适用于手机号提取、快递单号提取、充值号码提取等场景
手写文字识别	支持对图片中的手写中文、手写数字进行检测和识别
护照识别	支持对中国大陆护照个人资料页所有15个字段进行结构化识别
户口本识别	对出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码字段进行识别
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果
通用机打发票	支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别
机动车销售发票	支持对机动车销售发票的26个关键字段进行结构化识别
车辆合格证	支持对车辆合格证的23个关键字段进行结构化识别
通用机打发票	对国家/地方税务局发行的横/竖版通用机打发票进行结构化识别
护照识别	支持对中国大陆护照个人资料页所有11个字段进行结构化识别
医疗费用明细识别	支持识别全国医疗费用明细识别
网约车行程单	对国家/地方税务局发行的横/对各大主要服务商的网约车行程单进行结构化识别

识别	对国家/地方税务局发行的横/对各人主要服务商的网约车行程单进行结构化识别
磅单识别	结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单
仪器仪表盘读数识别	适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号
自定义模板文字识别	针对固定版式卡证票据提供的 OCR 定制化产品，可由用户自助创建识别模板和分类器，实现对任意版式卡证票据进行自动分类并结构化输出识别结果
医疗费用明细识别	支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含项目类型、项目名称、单价、数量、规格、金额
办公文档识别	可对办公类文档的版面进行分析，输出图、表、标题、文本、目录、栏、页眉、页脚、页码和脚注的位置，并输出分版块内容的OCR识别结果
印章识别	检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别
机动车登记证书识别	对机动车登记证书的编号、机动车所有人、登记机关、车辆类型、发证机关章等15个关键字段进行结构化识别
智能财务票据识别	对增值税发票、卷票、火车票、出租车票、机票行程单等13类票据混贴的图片进行切分识别
增值税发票验真	支持9种增值税发票的真伪及字段信息准确性校验，包括增值税专票、电子专票、普票、电子普票、卷票、通行费增值税电子普票、货运专票、机动车销售发票、二手车销售发票，支持返回票面的全部信息
医疗发票识别	支持识别全国各地门诊/住院发票的 业务流水号、发票号、住院号、门诊号、病例号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付等关键字段。支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数
门脸文字识别	识别图片中的门脸文字信息，自动过滤非门脸文字内容，接口返回门脸名称、描述文字和置信度
车辆证照混贴识别	对机动车行驶证主页及副页、驾驶证主页及副页在同一张图片上的场景进行结构化识别
公式识别	对试卷中的数学公式及题目内容进行识别
图文转换器	可识别图片/PDF文档版面布局，提取文字内容，并转换为保留原文档版式的Word、Excel文档，方便二次编辑和复制，可支持含表格、印章、水印、手写等内容的文档

版本更新记录

上线日期	版本号	更新内容
2022.08.23	4.16.11	新增通信行程卡识别、健康码识别、核酸证明接口
2021.12.11	4.16.3	新增： 网约车行程单识别，磅单识别，医疗明细识别
2021.05.26	4.15.8	新增 二维码、行程单、机动车销售发票、车辆合格证、试卷分析与识别、手写、护照、户口本、通用机打
2021.01.28	4.15.4	新增 增值税发票、出租车票、VIN码、火车票、数字识别
2020.08.06	4.15.1	新增 文档版面分析与识别，仪器仪表盘读数识别， 网络图片文字识别
2018.4.3	4.2.0	新增表格识别同步接口
2018.1.11	4.1.0	新增自定义模板ocr接口
2017.12.22	4.0.0	接口统一升级
2017.10.18	3.2.1	使用proxy问题修复
2017.8.25	3.0.0	新增营业执照识别接口，更新sdk打包方式：所有AI服务集成一个SDK
2017.8.10	1.3.9	新增票据识别接口
2017.7.28	1.3.8	新增通用文字识别高精度版（包括带位置信息版），通用文字识别、通用文字识别（带位置信息版）、通用文字识别（生僻字版）支持参数为图片url
2017.7.14	1.3.7	新增车牌识别接口，更新sdk打包方式
2017.6.30	1.3.6	新增表格识别系列接口
2017.6.16	1.3.5	新增驾驶证、行驶证识别接口
2017.4.13	1.3.2	新增通用文字识别（含生僻字版）和网图识别接口
2017.3.31	1.3.1	新增通用文字识别（含位置信息版）
2017.3.23	1.3	兼容Android环境
2017.3.2	1.2	上线对图片参数要求限制，增加设置超时接口
2017.1.20	1.1	对部分云用户调用不成功的错误修复
2017.1.6	1.0	初始版本，上线身份证识别、银行卡识别和通用文字识别接口

快速入门

安装OCR Java SDK

OCR Java SDK目录结构

```
com.baidu.aip
    ├─── auth                    //签名相关类
    ├─── http                    //Http通信相关类
    ├─── client                  //公用类
    ├─── exception                 //exception类
    ├─── ocr
    │      └─── AipOcr             //AipOcr类
    └─── util                    //工具类
```

支持 JAVA版本：1.7+

使用maven依赖：

添加以下依赖即可，其中版本号可在maven官网查询

```
<dependency>
  <groupId>com.baidu.aip</groupId>
  <artifactId>java-sdk</artifactId>
  <version>${version}</version>
</dependency>
```

直接使用JAR包步骤如下：

1.在官方网站下载Java SDK压缩工具包。

2.将下载的`aip-java-sdk-version.zip`解压后，复制到工程文件夹中。

3.在Eclipse右键“工程 -> Properties -> Java Build Path -> Add JARs”。

4.添加SDK工具包`aip-java-sdk-version.jar`和第三方依赖工具包`json-20160810.jar` `slf4j-simple-1.7.25.jar`。

其中，`version`为版本号，添加完成后，用户就可以在工程中使用OCR Java SDK。

新建AipOcr

AipOcr是Optical Character Recognition的Java客户端，为使用Optical Character Recognition的开发人员提供了一系列的交互方法。

用户可以参考如下代码新建一个AipOcr,初始化完成后建议**单例使用**,避免重复获取access_token：

```
import java.util.*;
import org.json.JSONObject;
import com.baidu.aip.ocr.AipOcr;
public class Sample {
    //设置APPID/AK/SK
    public static final String APP_ID = "你的 App ID";
    public static final String API_KEY = "你的 Api Key";
    public static final String SECRET_KEY = "你的 Secret Key";

    public static void main(String[] args) {
        // 初始化一个AipOcr
        AipOcr client = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

        // 可选：设置网络连接参数
        client.setConnectionTimeoutInMillis(2000);
        client.setSocketTimeoutInMillis(60000);

        // 可选：设置代理服务器地址, http和socket二选一，或者均不设置
        client.setHttpProxy("proxy_host", proxy_port);  // 设置http代理
        client.setSocketProxy("proxy_host", proxy_port);  // 设置socket代理

        // 可选：设置log4j日志输出格式，若不设置，则使用默认配置
        // 也可以直接通过jvm启动参数设置此环境变量
        System.setProperty("aip.log4j.conf", "path/to/your/log4j.properties");

        // 调用接口
        String path = "test.jpg";
        JSONObject res = client.basicGeneral(path, new HashMap<String, String>());
        System.out.println(res.toString(2));

    }
}
```

其中示例的log4j.properties文件内容如下：

```
#可以设置级别：debug>info>error
#debug：显示debug、info、error
#info：显示info、error
#error：只error
log4j.rootLogger=debug,appender1
#log4j.rootLogger=info,appender1
#log4j.rootLogger=error,appender1

#输出到控制台
log4j.appender.appender1=org.apache.log4j.ConsoleAppender
#样式为TTCCLayout
log4j.appender.appender1.layout=org.apache.log4j.PatternLayout

#自定义样式
# %r 时间 0
# %t 方法名 main
# %p 优先级 DEBUG/INFO/ERROR
# %c 所属类的全名(包括包名)
# %l 发生的位置，在某个类的某行
# %m 输出代码中指定的讯息，如log(message)中的message
# %n 输出一个换行

log4j.appender.appender1.layout.ConversionPattern=[%d{yy/MM/dd HH:mm:ss:SSS}][%t][%p] -%l %m%n
```

在上面代码中，常量`APP_ID`在百度智能云控制台中创建，常量`API_KEY`与`SECRET_KEY`是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的**应用列表**中查看。

注意：如您以前是百度智能云的老用户，其中`API_KEY`对应百度智能云的“Access Key ID”，`SECRET_KEY`对应百度智能云的“Access Key Secret”。

配置AipOcr

如果用户需要配置AipOcr的一些细节参数，可以在构造AipOcr之后调用接口设置参数，目前只支持以下参数：

接口	说明
setConnectionTimeoutInMillis	建立连接的超时时间（单位：毫秒）
setSocketTimeoutInMillis	通过打开的连接传输数据的超时时间（单位：毫秒）
setHttpProxy	设置http代理服务器
setSocketProxy	设置socket代理服务器 （http和socket类型代理服务器只能二选一）

接口说明

通用文字识别

用户向服务请求识别某张图中的所有文字。

```
public void sample(AipOcr client) {
  // 传入可选参数调用接口
  HashMap<String, String> options = new HashMap<String, String>();
  options.put("language_type", "CHN_ENG");
  options.put("detect_direction", "true");
  options.put("detect_language", "true");
  options.put("probability", "true");


  // 参数为本地图片路径
  String image = "test.jpg";
  JSONObject res = client.basicGeneral(image, options);
  System.out.println(res.toString(2));

  // 参数为本地图片二进制数组
  byte[] file = readImageFile(image);
  res = client.basicGeneral(file, options);
  System.out.println(res.toString(2));


  // 通用文字识别, 图片参数为远程url图片
  JSONObject res = client.basicGeneralUrl(url, options);
  System.out.println(res.toString(2));

}
```

通用文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
url	是	String			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	String	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	String	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	String	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	String	true false		是否返回识别结果中每一行的置信度

通用文字识别 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别 返回示例

```
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青岛啤酒"}
   ]
}
```

通用文字识别（高精度版）

用户向服务请求识别某张图中的所有文字，相对于通用文字识别该产品精度更高，但是识别耗时会稍长。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("detect_direction", "true");
      options.put("probability", "true");


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.basicAccurateGeneral(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.basicAccurateGeneral(file, options);
      System.out.println(res.toString(2));

   }
```

通用文字识别（高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
detect_direction	否	String	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
probability	否	String	true false		是否返回识别结果中每一行的置信度

通用文字识别（高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别（高精度版） 返回示例

参考通用文字识别返回示例

通用文字识别（含位置信息版）

用户向服务请求识别某张图中的所有文字，并返回文字在图中的位置信息。

```
public void sample(AipOcr client) {
  // 传入可选参数调用接口
  HashMap<String, String> options = new HashMap<String, String>();
  options.put("recognize_granularity", "big");
  options.put("language_type", "CHN_ENG");
  options.put("detect_direction", "true");
  options.put("detect_language", "true");
  options.put("vertexes_location", "true");
  options.put("probability", "true");


  // 参数为本地图片路径
  String image = "test.jpg";
  JSONObject res = client.general(image, options);
  System.out.println(res.toString(2));

  // 参数为本地图片二进制数组
  byte[] file = readImageFile(image);
  res = client.general(file, options);
  System.out.println(res.toString(2));


  // 通用文字识别（含位置信息版），图片参数为远程url图片
  JSONObject res = client.generalUrl(url, options);
  System.out.println(res.toString(2));

}
```

通用文字识别（含位置信息版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
url	是	String			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
recognize_granularity	否	String	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值； small：定位单字符位置
language_type	否	String	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	String	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	String	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
vertexes_location	否	String	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	String	true false		是否返回识别结果中每一行的置信度
paragraph	否	String	true false		是否输出段落信息
通用文字识别（含位置信息版） 返回数据参数详情					

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置信息版） 返回示例

```
 {
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
    "location": {
      "left": 35,
      "top": 53,
      "width": 193,
      "height": 109
    },
    "words": "感动",
    "chars": [    //recognize_granularity=small时存在
      {
        "location": {
          "left": 56,
          "top": 65,
          "width": 69,
          "height": 88
        },
        "char": "感"
      },
      {
        "location": {
          "left": 140,
          "top": 65,
          "width": 70,
          "height": 88
        },
        "char": "动"
      }
    ]
  }
  ...
]
}
```

通用文字识别（含位置高精度版）

用户向服务请求识别某张图中的所有文字，并返回文字在图片中的坐标信息，相对于通用文字识别（含位置信息版）该产品精度更高，但是识别耗时会稍长。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("recognize_granularity", "big");
      options.put("detect_direction", "true");
      options.put("vertexes_location", "true");
      options.put("probability", "true");


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.accurateGeneral(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.accurateGeneral(file, options);
      System.out.println(res.toString(2));

    }
```

通用文字识别（含位置高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
recognize_granularity	否	String	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	String	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vertexes_location	否	String	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	String	true false		是否返回识别结果中每一行的置信度

通用文字识别（含位置高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置高精度版） 返回示例

```
{
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
    "location": {
      "left": 35,
      "top": 53,
      "width": 193,
      "height": 109
    },
    "words": "感动",
    "chars": [    //recognize_granularity=small时存在
      {
        "location": {
          "left": 56,
          "top": 65,
          "width": 69,
          "height": 88
        },
        "char": "感"
      },
      {
        "location": {
          "left": 140,
          "top": 65,
          "width": 70,
          "height": 88
        },
        "char": "动"
      }
    ]
  }
  ...
]
}
```


通用文字识别（含生僻字版）

【该服务已停止更新，如需更好的识别效果请使用通用文字识别（高精度版 / 高精度含位置版），此两项服务已扩充字库，可支持生僻字识别】字库范围更大，支持对图片中的生僻字进行识别

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("language_type", "CHN_ENG");
      options.put("detect_direction", "true");
      options.put("detect_language", "true");
      options.put("probability", "true");


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.enhancedGeneral(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.enhancedGeneral(file, options);
      System.out.println(res.toString(2));


      // 通用文字识别（含生僻字版），图片参数为远程url图片
      JSONObject res = client.enhancedGeneralUrl(url, options);
      System.out.println(res.toString(2));

    }
```

通用文字识别（含生僻字版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
url	是	String			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	String	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	String	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	String	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	String	true false		是否返回识别结果中每一行的置信度

通用文字识别（含生僻字版） 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含生僻字版） 返回示例

```
 {
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青島啤酒"}
   ]
}
```

网络图片文字识别

用户向服务请求识别一些网络上背景复杂，特殊字体的文字。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("detect_direction", "true");
      options.put("detect_language", "true");


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.webImage(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.webImage(file, options);
      System.out.println(res.toString(2));


      // 网络图片文字识别, 图片参数为远程url图片
      JSONObject res = client.webImageUrl(url, options);
      System.out.println(res.toString(2));

    }
```

网络图片文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
url	是	String			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
detect_direction	否	String	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	String	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

网络图片文字识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

网络图片文字识别 返回示例

```
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
  [
    {"words": " TSINGTAO"},
    {"words": "青岛啤酒"}
  ]
}
```

身份证识别

用户向服务请求识别身份证，身份证识别包括正面和背面。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("detect_direction", "true");
      options.put("detect_risk", "false");

      String idCardSide = "back";

      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.idcard(image, idCardSide, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.idcard(file, idCardSide, options);
      System.out.println(res.toString(2));

    }
```

身份证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
id_card_side	是	String	front - 身份证含照片的一面 back - 身份证带国徽的一面		front：身份证含照片的一面；back：身份证带国徽的一面
detect_direction	否	String	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_risk	否	String	true - 开启 false - 不开启	false	是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)功能，默认不开启，即：false。可选值:true-开启；false-不开启
detect_photo	否	String	true - 开启 false - 不开启	false	是否检测头像内容，默认不检测。可选值：true-检测头像并返回头像的base64 编码及位置信息
detect_card	否	String	true - 开启 false - 不开启	false	是否检测身份证进行裁剪，默认不检测。可选值：true-检测身份证并返回证照的 base64 编码及位置信息

身份证识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
image_status	是	string	normal-识别正常 reversed_side-未摆正身份证 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 over_exposure-身份证关键字段反光或过曝 unknown-未知状态
risk_type	否	string	输入参数 detect_risk = true 时，则返回该字段识别身份证类型: normal-正常身份证；copy-复印件；temporary-临时身份证；screen-翻拍；unknow-其他未知情况
edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array(object)	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+location	是	array(object)	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	string	识别结果字符串

身份证识别 返回示例


```
{
  "log_id": 2648325511,
  "direction": 0,
  "image_status": "normal",
  "idcard_type": "normal",
  "edit_tool": "Adobe Photoshop CS3 Windows",
  "words_result": {
    "住址": {
      "location": {
        "left": 267,
        "top": 453,
        "width": 459,
        "height": 99
      },
      "words": "南京市江宁区弘景大道3889号"
    },
    "公民身份号码": {
      "location": {
```

```
            "left": 443,
            "top": 681,
            "width": 589,
            "height": 45
        },
        "words": "330881199904173914"
    },
    "出生": {
        "location": {
            "left": 270,
            "top": 355,
            "width": 357,
            "height": 45
        },
        "words": "19990417"
    },
    "姓名": {
        "location": {
            "left": 267,
            "top": 176,
            "width": 152,
            "height": 50
        },
        "words": "伍云龙"
    },
    "性别": {
        "location": {
            "left": 269,
            "top": 262,
            "width": 33,
            "height": 52
        },
        "words": "男"
    },
    "民族": {
        "location": {
            "left": 492,
            "top": 279,
            "width": 30,
            "height": 37
        },
        "words": "汉"
    }
  },
  "words_result_num": 6
}
```


银行卡识别

识别银行卡并返回卡号和发卡行。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.bankcard(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.bankcard(file, options);
      System.out.println(res.toString(2));

    }
```

银行卡识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据

银行卡识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	请求标识码，随机数，唯一。
result	object	是	返回结果
+bank_card_number	string	是	银行卡卡号
+bank_name	string	是	银行名，不能识别时为空
+bank_card_type	number	是	银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡

银行卡识别 返回示例

```
{
  "log_id": 1447188951,
  "result": {
     "bank_card_number": "622500000000000",
     "bank_name": "招商银行",
     "bank_card_type": 1
  }
}
```

驾驶证识别

对机动车驾驶证所有关键字段进行识别。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("detect_direction", "true");


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.drivingLicense(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.drivingLicense(file, options);
      System.out.println(res.toString(2));

    }
```

驾驶证识别 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和image二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	否	string	true/false	- **false：默认值，**不检测朝向，朝向是指输入图像是正常方向、逆时针旋转90/180/270度 - **true：**检测朝向
driving_license_side	否	string	front/back	- **front：默认值，**识别驾驶证正页 - **back：**识别驾驶证副页
unified_valid_period	否	bool	true/false	- **false: 默认值**，不进行归一化处理 - **true:** 归一化格式输出驾驶证的「有效起始日期」+「有效期限」及「有效期限」+「至」两种输出格式归一化为「**有效起始日期**」+「**失效日期**」
quality_warn	否	string	true/false	是否开启质量检测功能，仅在驾驶证正页识别时生效， - **false：默认值**，不输出质量告警信息 - **true：** 输出驾驶证遮挡、不完整质量告警信息
risk_warn	否	string	true/false	是否开启风险检测功能， - **false：默认值**，不输出风险告警信息 - **true**：开启，输出驾驶证复印、翻拍、PS等告警信息

驾驶证识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ words	否	string	识别结果字符串
warn_infos	否	array[]	当输入参数 driving_license_side=front，且 quality_warn=true 时输出， - shield：驾驶证证照存在遮挡告警提示 - incomplete：驾驶证证照边框不完整告警提示
risk_type	否	string	当输入参数 risk_warn=true 时返回识出的驾驶证的类型：normal-正常驾驶证；copy-复印件；screen-翻拍
edit_tool	否	string	当输入参数 risk_warn=true 时返回，如果检测驾驶证被编辑过，该字段指定编辑软件名称，如：Adobe Photoshop CC 2014 (Macintosh)，如果没有被编辑过则返回值为空

驾驶证识别 返回示例

```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "证号": {
    "words": "3208231999053090"
   },
   "有效期限": {
    "words": "6年"
   },
   "准驾车型": {
    "words": "B2"
   },
   "有效起始日期": {
    "words": "20101125"
   },
   "住址": {
    "words": "江苏省南通市海门镇秀山新城"
   },
   "姓名": {
    "words": "小欧欧"
   },
   "国籍": {
    "words": "中国"
   },
   "出生日期": {
    "words": "19990530"
   },
   "性别": {
    "words": "男"
   },
   "初次领证日期": {
    "words": "20100125"
   }
  }
 }
}
```


行驶证识别

对机动车行驶证所有关键字段进行识别。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("detect_direction", "true");
      options.put("accuracy", "normal");


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.vehicleLicense(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.vehicleLicense(file, options);
      System.out.println(res.toString(2));

    }
```

行驶证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
detect_direction	否	String	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vehicle_license_side	否	string	front/back	front	- **front**：识别行驶证主页 - **back**：识别行驶证副页
unified	否	string	true/false	false	- **false**：不进行归一化处理 - **true**：对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出

行驶证识别 返回数据参数详情

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
+words	否	string	识别结果字符串

行驶证识别 返回示例

```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "品牌型号": {
    "words": "保时捷GT37182RUCRE"
   },
   "发证日期": {
    "words": "20160104"
   },
   "使用性质": {
    "words": "非营运"
   },
   "发动机号码": {
    "words": "20832"
   },
   "号牌号码": {
    "words": "苏A001"
   },
   "所有人": {
    "words": "圆圆"
   },
   "住址": {
    "words": "南京市江宁区弘景大道"
   },
   "注册日期": {
    "words": "20160104"
   },
   "车辆识别代号": {
    "words": "HCE58"
   },
   "车辆类型": {
    "words": "小型轿车"
   }
  }
 }
}
```


车牌识别

识别机动车车牌，并返回号牌号码和车牌颜色。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("multi_detect", "true");


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.plateLicense(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.plateLicense(file, options);
      System.out.println(res.toString(2));

    }
```

车牌识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
multi_detect	否	String	true false	false	是否检测多张车牌，默认为false，当置为true的时候可以对一张图片内的多张车牌进行识别

车牌识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一
Color	string	是	车牌颜色
number	string	是	车牌号码

车牌识别 返回示例

```
{
  "log_id": 3583925545,
  "words_result": {
     "color": "blue",
     "number": "苏HS7766"
  }
}
```

营业执照识别

识别营业执照，并返回关键字段的值，包括单位名称、法人、地址、有效期、证件编号、社会信用代码等。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.businessLicense(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.businessLicense(file, options);
      System.out.println(res.toString(2));

    }
```

营业执照识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据

营业执照识别 返回数据参数详情

参数	是否必须	类型	说明
log_id	是	number	请求标识码，随机数，唯一。
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	否	string	识别结果字符串

营业执照识别 返回示例

```
{
  "log_id": 490058765,
  "words_result": {
    "单位名称": {
      "location": {
        "left": 500,
        "top": 479,
        "width": 618,
        "height": 54
      },
      "words": "袁氏财团有限公司"
    },
    "法人": {
      "location": {
        "left": 938,
        "top": 557,
        "width": 94,
        "height": 46
      },
      "words": "袁运筹"
    },
    "地址": {
      "location": {
        "left": 503,
        "top": 644,
        "width": 574,
        "height": 57
      },
      "words": "江苏省南京市中山东路19号"
    },
    "有效期": {
      "location": {
        "left": 779,
        "top": 1108,
        "width": 271,
        "height": 49
      },
      "words": "2015年02月12日"
    },
    "证件编号": {
      "location": {
        "left": 1219,
        "top": 357,
        "width": 466,
        "height": 39
      },
      "words": "苏餐证字(2019)第666602666661号"
    },
    "社会信用代码": {
      "location": {
        "left": 0,
        "top": 0,
        "width": 0,
        "height": 0
      },
      "words": "无"
    }
  },
  "words_result_num": 6
}
```

通用票据识别

用户向服务请求识别医疗票据、增值税发票、出租车票、保险保单等票据类图片中的所有文字，并返回文字在图中的位置信息。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("recognize_granularity", "big");
      options.put("probability", "true");
      options.put("accuracy", "normal");
      options.put("detect_direction", "true");


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.receipt(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.receipt(file, options);
      System.out.println(res.toString(2));

    }
```

通用票据识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
recognize_granularity	否	String	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	String	true false		是否返回识别结果中每一行的置信度
accuracy	否	String	normal - 使用快速服务		normal 使用快速服务，1200ms左右时延；缺省或其它值使用高精度服务，1600ms左右时延
detect_direction	否	String	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

通用票据识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array()	单字符结果，recognize_granularity=small时存在
location	是	array()	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位定位位置的长方形的宽度
height	是	number	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值

通用票据识别 返回示例

```
{
  "log_id": 2661573626,
  "words_result": [
    {
      "location": {
        "left": 10,
        "top": 3,
        "width": 121,
        "height": 24
      },
      "words": "姓名:小明明",
      "chars": [
        {
          "location": {
            "left": 16,
            "top": 6,
            "width": 17,
            "height": 20
          },
          "char": "姓"
        }
        ...
      ]
    },
    {
      "location": {
        "left": 212,
        "top": 3,
        "width": 738,
        "height": 24
      },
      "words": "卡号/病案号:105353990标本编号:150139071送检科室:血液透析门诊病房",
      "chars": [
        {
          "location": {
            "left": 218,
            "top": 6,
            "width": 18,
            "height": 21
          },
          "char": "卡"
        }
        ...
      ]
    }
  ],
  "words_result_num": 2
}
```


自定义模板文字识别

自定义模板文字识别，是针对百度官方没有推出相应的模板，但是当用户需要对某一类卡证/票据（如房产证、军官证、火车票等）进行结构化的提取内容时，可以使用该产品快速制作模板，进行识别。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();

      options.put("templateSign", "Nsdax2424asaAS791823112");

      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.custom(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.custom(file, options);
      System.out.println(res.toString(2));
    }
```

自定义模板文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据
templateSign	否	String	您在自定义文字识别平台制作的模板的ID
classifierId	否	string	分类器Id。这个参数和templateSign至少存在一个，优先使用templateSign。存在templateSign时，表示使用指定模板；如果没有templateSign而有classifierId，表示使用分类器去判断使用哪个模板

自定义模板文字识别 返回数据参数详情

字段	是否必选	类型	说明
error_code	number	number	0代表成功，如果有错误码返回可以参考下方错误码列表排查问题
error_msg	是	string	具体的失败信息，可以参考下方错误码列表排查问题
data	jsonObject	识别返回的结果	

自定义模板文字识别 返回示例

```
{
  "isStructured": true,
  "ret": [
    {
      "charset": [
        {
          "rect": {
            "top": 183,
            "left": 72,
            "width": 14,
            "height": 28
          },
          "word": "5"
        },
        {
          "rect": {
            "top": 183,
            "left": 90,
            "width": 14,
            "height": 28
          },
```

```
            "word": "4"
        },
        {
            "rect": {
                "top": 183,
                "left": 103,
                "width": 15,
                "height": 28
            },
            "word": "."
        },
        {
            "rect": {
                "top": 183,
                "left": 116,
                "width": 14,
                "height": 28
            },
            "word": "5"
        },
        {
            "rect": {
                "top": 183,
                "left": 133,
                "width": 19,
                "height": 28
            },
            "word": "元"
        }
    ],
    "word_name": "票价",
    "word": "54.5元"
},
{
    "charset": [
        {
            "rect": {
                "top": 144,
                "left": 35,
                "width": 14,
                "height": 28
            },
            "word": "2"
        },
        {
            "rect": {
                "top": 144,
                "left": 53,
                "width": 14,
                "height": 28
            },
            "word": "0"
        },
        {
            "rect": {
                "top": 144,
                "left": 79,
                "width": 14,
                "height": 28
            },
            "word": "1"
        },
        {
```

```
            {
                "rect": {
                    "top": 144,
                    "left": 97,
                    "width": 14,
                    "height": 28
                },
                "word": "7"
            }
        ]
    ]
}
```

试卷分析与识别

可对文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景。

```
public void sample(AipOcr client) {
  // 传入可选参数调用接口
  HashMap<String, String> options = new HashMap<String, String>();

  // 参数为本地图片路径
  String image = "test.jpg";
  JSONObject res = client.docAnalysis(image, options);
  System.out.println(res.toString(2));

  // 参数为本地图片二进制数组
  byte[] file = readImageFile(image);
  res = client.docAnalysis(file, options);
  System.out.println(res.toString(2));
}
```

识别结果请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少64px，最长边最大4096px。注意：图片的base64编码是不包含图片头的，如（）	
language_type	false	string	CHN_ENG / ENG	识别语言类型，默认为CHN_ENG 可选值包括：=CHN_ENG：中英文 =ENG：英文	
result_type	false	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果	
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度	
line_probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false	
words_type	false	string	handwring_only/ handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别	
layout_analysis	false	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出	

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
img_direction	false	int32	detect_direction=true时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	true	uint32	识别结果数，表示results的元素个数
results	true	array[]	识别结果数组
+words_type	true	string	文字属性（手写、印刷）,handwriting 手写，print 印刷
+words	true	array[]	整行的识别结果数组。
++line_probability	false	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++average	false	float	行置信度
+++min	false	float	整行中单字的最低置信度
++word	true	float	整行的识别结果
++words_location	true	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	true	uint32	表示定位定位位置的长方形的宽度
+++height	true	uint32	表示位置的长方形的高度
+chars	false	array[]	result_type=small时返回。单字符结果数组。
++char	false	string	result_type=small时返回。每个单字的内容。
++chars_location	false	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示位置的长方形的高度
layouts_num	false	uint32	版面分析结果数，表示layout的元素个数
layouts	false	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+layout	false	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title
+layout_location	false	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++x	false	uint32	水平坐标（坐标0点为左上角）
++y	false	uint32	水平坐标（坐标0点为左上角）
+layout_idx	false	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）

仪器仪表盘读数识别

适用于不同品牌、不同型号的仪器仪表盘读数识别，广泛适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号，支持液晶屏、字轮表等表型。

```
public void sample(AipOcr client) {
  // 传入可选参数调用接口
  HashMap<String, String> options = new HashMap<String, String>();

  // 参数为本地图片路径
  String image = "test.jpg";
  JSONObject res = client.meter(image, options);
  System.out.println(res.toString(2));

  // 参数为本地图片二进制数组
  byte[] file = readImageFile(image);
  res = client.meter(file, options);
  System.out.println(res.toString(2));
}
```

识别结果请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px。支持jpg/jpeg/png/bmp格式.注意：图片的base64编码是不包含图片头的，如（）	
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false	
poly_location	false	string	true/false	位置信息返回形式，默认：false false：只给出识别结果所在长方形位置信息 true：除了默认的识别文字所在长方形的位置信息，还会给出文字所在区域的最小外接旋转矩形的4个点坐标信息	

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	识别结果字符串
+location	true	array[]	识别结果所在长方形位置信息
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	false	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	false	array[]	poly_location=true时存在。文字所在区域的外接四边形的4个点坐标信息

网络图片文字识别（含位置版）

支持识别艺术字体或背景复杂的文字内容，除文字信息外，还可返回每行文字的位置信息、行置信度，以及单字符内容和位置等。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();

      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.webimageLoc(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.webimageLoc(file, options);
      System.out.println(res.toString(2));
    }
```

网络图片文字识别（含位置版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，像素乘积不超过2048*2048（1024*1024以内图像处理效果最佳）。注意：图片的base64编码是不包含图片头的，如（）	
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向	
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false	
poly_location	false	string	true/false	是否返回文字所在区域的外接四边形的4个点坐标信息。默认为false	
recognize_granularity	false	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置	

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
direction	false	int32	图像方向，当detect_direction=true时存在。检测到的图像朝向： 0 ：正向； 1：逆时针旋转90度； 2：逆时针旋转180度； 3：逆时针旋转270度
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	整行的识别结果
+location	true	object	整行的矩形框坐标。位置数组（坐标0点为左上角）
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	true	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	true	array[]	poly_location=true时存在。文字所在区域的外接矩形的4个点坐标信息
++x	true	uint32	水平坐标（坐标0点为左上角）
++y	true	uint32	垂直坐标（坐标0点为左上角）
+chars	false	array[]	单字符结果，recognize_granularity=small时存在
++char	false	string	单字符识别结果
++location	false	object	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示定位定位位置的长方形的高度

增值税发票识别

支持对增值税普票、专票、卷票、电子发票、区块链发票的所有字段进行结构化识别，包括发票基本信息、销售方及购买方信息、商品信息、价税信息等，其中五要素字段的识别准确率超过 99.9%； 同时，支持对增值税卷票的 21 个关键字段进行识别，包括发票类型、发票代码、发票号码、机打号码、机器编号、收款人、销售方名称、销售方纳税人识别号、开票日期、购买方名称、购买方纳税人识别号、项目、单价、数量、金额、税额、合计金额(小写)、合计金额(大写)、校验码、省、市，四要素字段的识别准确率可达95%。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();

      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.vatInvoice(image,EImgType.FILE, options);
      System.out.println(res.toString(2));
      // 参数为url
      String image = "http://test.jpg";
      JSONObject res = client.vatInvoice( image, EImgType.URL, options);
      System.out.println(res.toString(2));
      // 参数为本地pdf
      String image = "test.pdf";
      SONObject res = client.vatInvoice( image, EImgType.PDF, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.vatInvoice(file, options);
      System.out.println(res.toString(2));
    }
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url或者pdf文件
imageType	是	EImgType	FILE,URL,PDF 表示image 类型
type	否	String	可选参数，进行识别的增值税发票类型，默认为 normal，可缺省normal：可识别增值税普票、专票、电子发票roll：可识别增值税卷票
返回参数			

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	是	string	发票种类
InvoiceTypeOrg	是	string	发票名称
InvoiceCode	是	string	发票代码
InvoiceNum	是	string	发票号码
MachineNum	是	string	机打号码。仅增值税卷票含有此参数
MachineCode	是	string	机器编号。仅增值税卷票含有此参数
CheckCode	否	string	校验码。增值税专票无此参数
InvoiceDate	是	string	开票日期
PurchaserName	是	string	购方名称
PurchaserRegisterNum	是	string	购方纳税人识别号
PurchaserAddress	是	string	购方地址及电话

PurchaserBank	是	string	购方开户行及账号
Password	是	string	密码区
Province	是	string	省
City	是	string	市
SheetNum	是	string	联次
Agent	是	string	是否代开
CommodityName	是	array[]	货物名称
- row	是	uint32	行号
- word	是	string	内容
CommodityType	是	array[]	规格型号
- row	是	uint32	行号
- word	是	string	内容
CommodityUnit	是	array[]	单位
- row	是	uint32	行号
- word	是	string	内容
CommodityNum	是	array[]	数量
- row	是	uint32	行号
- word	是	string	内容
CommodityPrice	是	array[]	单价
- row	是	uint32	行号
- word	是	string	内容
CommodityAmount	是	array[]	金额
- row	是	uint32	行号
- word	是	string	内容
CommodityTaxRate	是	array[]	税率
- row	是	uint32	行号
- word	是	string	内容
CommodityTax	是	array[]	税额
- row	是	uint32	行号
- word	是	string	内容
SellerName	是	string	销售方名称
SellerRegisterNum	是	string	销售方纳税人识别号
SellerAddress	是	string	销售方地址及电话
SellerBank	是	string	销售方开户行及账号
TotalAmount	是	uint32	合计金额
TotalTax	是	uint32	合计税额
AmountInWords	是	string	价税合计(大写)
AmountInFiguers	是	uint32	价税合计(小写)
Payee	是	string	收款人
Checker	是	string	复核

NoteDrawer	是	string	开票人
Remarks	是	string	备注

返回示例


```
{
 "log_id": "5425496231209218858",
"words_result_num": 29,
"words_result": {
 "InvoiceNum": "14641426",
 "SellerName": "上海易火广告传媒有限公司",
 "CommodityTaxRate": [
  {
   "word": "6%",
   "row": "1"
  }
 ],
 "SellerBank": "中国银行南翔支行446863841354",
 "Checker": ":沈园园",
 "TotalAmount": "94339.62",
 "CommodityAmount": [
  {
   "word": "94339.62",
   "row": "1"
  }
 ],
 "InvoiceDate": "2016年06月02日",
 "CommodityTax": [
  {
   "word": "5660.38",
   "row": "1"
  }
 ],
 "PurchaserName": "百度时代网络技术(北京)有限公司",
 "CommodityNum": [
  {
   "word": "",
   "row": "1"
  }
 ],
       "Province": "上海",
       "City": "",
       "SheetNum": "第三联",
       "Agent": "否",
 "PurchaserBank": "招商银行北京分行大屯路支行86618202851000 30",
 "Remarks": "告传",
 "Password": "074/45781873408>/6>8>65*887676033/51+<5415>9/32--852>1+29<65>641-
5>66<500>87/*-34<943359034>716905113*4242>",
 "SellerAddress": ":嘉定区胜辛南路500号15幢1161室55033753",
 "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二属A2010-59108001",
 "InvoiceCode": "3100153130",
 "CommodityUnit": [
  {
   "word": "",
   "row": "1"
  }
 ],
 "Payee": ":徐蓉",
 "PurchaserRegisterNum": "110108787751579",
 "CommodityPrice": [
  {
   "word": "",
```

```
    "row": "1"
   }
  ],
  "NoteDrawer": "沈园园",
  "AmountInWords": "壹拾万圆整",
  "AmountInFiguers": "100000.00",
  "TotalTax": "5660.38",
  "InvoiceType": "专用发票",
  "SellerRegisterNum": "913101140659591751",
  "CommodityName": [
   {
    "word": "信息服务费",
    "row": "1"
   }
  ],
  "CommodityType": [
   {
    "word": "",
    "row": "1"
   }
  ]
 }
}
```


出租车票识别

支持识别全国各大城市出租车票的 16 个关键字段，包括发票号码、代码、车号、日期、总金额、燃油附加费、叫车服务费、省、市、单价、里程、上车时间、下车时间等。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();

      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.taxiReceipt(image,EImgType.FILE, options);
      System.out.println(res.toString(2));

      // 参数为url
      String image = "http://test.jpg";
      JSONObject res = client.taxiReceipt(image,EImgType.URL, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.taxiReceipt(file, options);
      System.out.println(res.toString(2));
    }
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url
imageType	是	EImgType	FILE,URL 表示image 类型
返回参数			

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
words_result_num	uint32	是	识别结果数，表示words_result的元素个数
InvoiceCode	string	是	发票代号
InvoiceNum	string	是	发票号码
TaxiNum	string	是	车牌号
Date	string	是	日期
Time	string	是	上下车时间
Fare	string	是	总金额
FuelOilSurcharge	string	是	燃油附加费
CallServiceSurcharge	string	是	叫车服务费
Province	string	是	省
City	string	是	市
PricePerkm	string	是	单价
Distance	string	是	里程

返回示例


```
{
"log_id":2034039896,
"words_result_num":6,
"words_result":
{
 "Date":"2017-11-26",
 "Fare":"¥153.30元",
 "InvoiceCode":"111001681009",
 "InvoiceNum":"90769610",
 "TaxiNum":"BV2062",
 "Time":"20:42-21:07",
 "FuelOilSurcharge": "¥0.00",
 "CallServiceSurcharge": "¥0.00",
       "Province": "浙江省",
       "City": "杭州市",
       "PricePerkm": "2.50元/KM",
       "Distance": "4.5KM"
}
}
```


VIN码识别

支持对车辆挡风玻璃处的车架号码进行识别。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();

      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.vinCode(image,EImgType.FILE, options);
      System.out.println(res.toString(2));

      // 参数为url
      String image = "http://test.jpg";
      JSONObject res = client.vinCode(image,EImgType.URL, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.vinCode(file, options);
      System.out.println(res.toString(2));
    }
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url
imageType	是	EImgType	FILE,URL 表示image 类型

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
location	是	object{}	识别结果
words	是	string	VIN码识别结果
words_result_num	是	int	识别结果数，表示words_result的元素个数

返回示例

```
{
"log_id": 246589877,
"words_result": [
 {
  "location": {
   "left": 124,
   "top": 11,
   "width": 58,
   "height": 359
  },
  "words": "LFV2A11K8D4010942"
 }
],
"words_result_num": 1
}
```

火车票识别

支持对红、蓝火车票的13个关键字段进行结构化识别，包括车票号码、始发站、目的站、车次、日期、票价、席别、姓名、座位号、身份证号、售站、序列号、时间。

```
    public void sample(AipOcr client) {

        // 传入可选参数调用接口
        HashMap<String, String> options = new HashMap<String, String>();

        // 参数为本地图片路径
        String image = "test.jpg";
        JSONObject res = client.trainTicket(image,EImgType.FILE, options);
        System.out.println(res.toString(2));

        // 参数为url
        String image = "http://test.jpg";
        JSONObject res = client.trainTicket(image,EImgType.URL, options);
        System.out.println(res.toString(2));

        // 参数为本地图片二进制数组
        byte[] file = readImageFile(image);
        res = client.trainTicket(file, options);
        System.out.println(res.toString(2));
    }
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url
imageType	是	EImgType	FILE,URL 表示image 类型
返回参数			

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
ticket_num	string	是	车票号
starting_station	string	是	始发站
train_num	string	是	车次号
destination_station	string	是	到达站
date	string	是	出发日期
ticket_rates	string	是	车票金额
seat_category	string	是	席别
name	string	是	乘客姓名
id_num	string	是	身份证号
serial_number	string	是	序列号
sales_station	string	是	售站
time	string	是	时间
seat_num	string	是	座位号
返回示例			

```
{
"log_id": "12317512659",
"direction": 1,
"words_result_num": 13,
"words_result": {
 "id_num": "2302051998****156X",
 "name": "裴一丽",
 "ticket_rates": "￥54.5元",
 "destination_station": "天津站",
 "seat_category": "二等座",
 "sales_station": "北京南",
 "ticket_num": "F05706",
 "seat_num": "02车03C号",
 "time": "09:36",
 "date": "2019年04月03日",
 "serial_number": "10010300067846",
 "train_num": "C255",
 "starting_station": "北京南站"
}
}
```


数字识别

对图片中的数字进行提取和识别，自动过滤非数字内容，仅返回数字内容及其位置信息，识别准确率超过99%。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("recognize_granularity","big");
      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.numbers(image,options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.numbers(file, options);
      System.out.println(res.toString(2));
    }
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据
recognize_granularity	false	string	big、small
detect_direction	false	string	true、false

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
      "location": {
        "left": 56,
        "top": 0,
        "width": 21,
        "height": 210
      },
      "words": "3"
    }
  ],
  "words_result_num": 1
}
```

二维码识别

对图片中的二维码、条形码进行检测和识别，返回存储的文字信息。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      // 参数为二进制数组
      byte[] file = readFile("test.jpg");
      res = client.qrcode(file, options);
      System.out.println(res.toString(2));

      // 参数图片url
      String url = "http://localhost/test.jpg"
      res = client.qrcodeUrl(url, options);
      System.out.println(res.toString(2));
    }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
codes_result_num	是	uint32	识别结果数，表示codes_result的元素个数
codes_result	是	array[]	定位和识别结果数组
-type	是	string	识别码类型条码类型包括：9种条形码（UPC_A、UPC_E、EAN_13、EAN_8、CODE_39、CODE_93、CODE_128、ITF、CODABAR），4种二维码（QR_CODE、DATA_MATRIX、AZTEC、PDF_417）
-text	是	string	条形码识别内容,暂时只限于识别中英文结果

返回示例

```
{
  "log_id": 863402790,
  "codes_result": [
     {
        "type": "QR_CODE",
        "text": [
           "中国",
           "北京"
        ]
     }
  ],
  "codes_result_num": 1
```

```
    "codes_result_num": 1
}
```


示例2（多个图的情况）：


```
{
    "log_id": 1508509437,
    "codes_result": [
        {
            "type": "QR_CODE",
            "text": [
                "HTTP://Q8R.HK/YELZ0"
            ]
        },
        {
            "type": "PDF_417",
            "text": [
                "PDF417倢丄TL-30偱撉傒庢傝壜擻偱偡丅"
            ]
        },
        {
            "type": "CODABAR",
            "text": [
                "000800"
            ]
        },
        {
            "type": "CODE_39",
            "text": [
                "1234567890"
            ]
        },
        {
            "type": "AZTEC",
            "text": [
                "www.tec-it.com"
            ]
        },
        {
            "type": "DATA_MATRIX",
            "text": [
                "Wikipedia, the free encyclopedia"
            ]
        },
        {
            "type": "CODE_93",
            "text": [
                "123456789"
            ]
        },
        {
            "type": "CODE_128",
            "text": [
                "50090500019191"
            ]
        },
        {
            "type": "EAN_8",
            "text": [
                "12345670"
            ]
        },
        {
            "type": "EAN_13",
            "text": [
```

```
            "6901234567892"
         ]
      },
      {
         "type": "UPC_E",
         "text": [
            "01234565"
         ]
      }
   ],
   "codes_result_num": 11
}
```

机动车销售发票

支持对机动车销售发票的26个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、机器编号、购买方名称、购买方身份证号码/组织机构代码、车辆类型、厂牌型号、产地、合格证号、发动机号码、车架号码、价税合计、价税合计小写、销货单位名称、电话、纳税人识别号、账号、地址、开户银行、税率、税额、主管税务机关及代码、不含税价格、限乘人数。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      // 参数为二进制数组
      byte[] file = readFile("test.jpg");
      res = client.vehicleInvoice(file, options);
      System.out.println(res.toString(2));

      // 参数图片url
      String url = "http://localhost/test.jpg"
      res = client.vehicleInvoiceUrl(url, options);
      System.out.println(res.toString(2));
   }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
InvoiceCode	是	string	发票代码/机打代码
InvoiceNum	是	string	发票号码/机打号码
InvoiceDate	是	string	开票日期
MachineCode	是	string	机器编号
Purchaser	是	string	购买方名称
PurchaserCode	是	string	购买方身份证号码/组织机构代码
VehicleType	是	string	车辆类型
ManuModel	是	string	厂牌型号
Origin	是	string	产地
CertificateNum	是	string	合格证号
EngineNum	是	string	发动机号码
VinNum	是	string	车架号码
PriceTax	是	string	价税合计
PriceTaxLow	是	string	价税合计小写
Saler	是	string	销货单位名称
SalerPhone	是	string	销货单位电话
SalerCode	是	string	销货单位纳税人识别号
SalerAccountNum	是	string	销货单位账号
SalerAddress	是	string	销货单位地址
SalerBank	是	string	销货单位开户银行
TaxRate	是	string	税率
Tax	是	string	税额
TaxAuthor	是	string	主管税务机关
TaxAuthorCode	是	string	主管税务机关代码
Price	是	string	不含税价格
LimitPassenger ｜ 是	string	限乘人数	

返回示例

```
{
  "log_id": 283449393728149457,
  "words_result_num": 26,
  "words_result": {
     "InvoiceNum": "00875336",
     "Saler": "深圳市新能源汽车销售有限公司",
     "LimitPassenger": "5",
     "MachineCode": "669745967911",
     "VinNum": "LJLGTCRP1J4007581",
     "TaxRate": "16%",
     "PriceTaxLow": "106100.00",
     "InvoiceDate": "2018-11-29",
     "Price": "￥91465.52",
     "SalerBank": "中国工商银行股份有限公司深圳岭园支行",
     "TaxAuthor": "国家锐务总局深圳市龙岗区税务局第五税务所",
     "ManuModel": "江淮牌HFC7007EYBD6",
     "CertificateNum": "WCH0794J0976801",
     "Purchaser": "苏子潇",
     "VehicleType": "纯电动轿车",
     "InvoiceCode": "14975047560",
     "PriceTax": "壹拾万陆仟壹佰圆整",
     "SalerPhone": "0755-83489306",
     "SalerAddress": "深圳市龙岗区龙岗街道百世国际汽车城",
     "Origin": "安徽省合肥市",
     "EngineNum": "18958407",
     "Tax": "14634.48",
     "PurchaserCode": "5135934475603742222",
     "TaxAuthorCode": "14037589413",
     "SalerAccountNum": "中国工商银行股份有限公司深圳岭园支行",
     "SalerCode": "91449283464582922 78H"
  }
}
```


车辆合格证

支持对车辆合格证的23个关键字段进行结构化识别，包括合格证编号、发证日期、车辆制造企业名、车辆品牌、车辆名称、车辆型号、车架号、车身颜色、发动机型号、发动机号、燃料种类、排量、功率、排放标准、轮胎数、轴距、轴数、转向形式、总质量、整备质量、驾驶室准乘人数、最高设计车速、车辆制造日期。

```
 public void sample(AipOcr client) {
  // 传入可选参数调用接口
  HashMap<String, String> options = new HashMap<String, String>();
  // 参数为二进制数组
  byte[] file = readFile("test.jpg");
  res = client.vehicleCertificate(file, options);
  System.out.println(res.toString(2));

  // 参数图片url
  String url = "http://localhost/test.jpg"
  res = client.vehicleCertificateUrl(url, options);
  System.out.println(res.toString(2));
 }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
CertificationNo	是	string	合格证编号
CertificateDate	是	string	发证日期
Manufacturer	是	string	车辆制造企业名
CarBrand	是	string	车辆品牌
CarName	是	string	车辆名称
CarModel	是	string	车辆型号
VinNo	是	string	车架号
CarColor	是	string	车身颜色
EngineType	是	string	发动机型号
EngineNo	是	string	发动机号
FuelType	是	string	燃料种类
Displacement	是	string	排量
Power	是	string	功率
EmissionStandard	是	string	排放标准
TyreNum	是	string	轮胎数
Wheelbase	是	string	轴距
AxleNum	是	string	轴数
SteeringType	是	string	转向形式
TotalWeight	是	string	总质量
SaddleMass	是	string	整备质量
LimitPassenger	是	string	驾驶室准乘人数
SpeedLimit	是	string	最高设计车速
ManufactureDate	是	string	车辆制造日期

返回示例

```
{
  "log_id": 14814098736243057,
  "words_result_num": 23,
  "words_result": {
     "ManufactureDate": "2016年10月13日",
     "CarColor": "红",
     "LimitPassenger": "2",
     "EngineType": "WP12.460E50",
     "TotalWeight": "25000",
     "Power": "338",
     "CertificationNo": "WEK29JX98645437",
     "FuelType": "汽油",
     "Manufacturer": "陕西汽车集团有限责任公司",
     "SteeringType": "方向盘",
     "Wheelbase": "3175+1350",
     "SpeedLimit": "105",
     "EngineNo": "1418K129178",
     "SaddleMass": "8600",
     "AxleNum": "3",
     "CarModel": "SX4250MC4",
     "VinNo": "LZGJHYD83JX197344",
     "CarBrand": "陕汽牌",
     "EmissionStandard": "GB17691-2005国V,GB3847-2005",
     "Displacement": "11596",
     "CertificateDate": "2018年11月28日",
     "CarName": "牵引汽车",
     "TyreNum": "10"
  }
}
```


户口本识别

支持对户口本内常住人口登记卡的全部 22 个字段进行结构化识别，包括户号、姓名、与户主关系、性别、出生地、民族、出生日期、身份证号、本市县其他住址、曾用名、籍贯、宗教信仰、身高、血型、文化程度、婚姻状况、兵役状况、服务处所、职业、何时由何地迁往本市、何时由何地迁往本址、登记日期。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      // 参数为二进制数组
      byte[] file = readFile("test.jpg");
      res = client.householdRegister(file, options);
      System.out.println(res.toString(2));

      // 参数图片url
      String url = "http://localhost/test.jpg"
      res = client.householdRegisterUrl(url, options);
      System.out.println(res.toString(2));
    }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和image二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
Name	是	object{}	姓名
+ words	是	string	所属字段的具体内容，以下各字段均含有此元素
Relationship	是	object{}	户主或与户主关系
Sex	是	object{}	性别
BirthAddress	是	object{}	出生地
Nation	是	object{}	民族
Birthday	是	object{}	生日
CardNo	是	object{}	身份证号
HouseholdNum	是	object{}	户号
FormerName	是	object{}	曾用名
Hometown	是	object{}	籍贯
OtherAddress	是	object{}	本市（县）其他住址
Belief	是	object{}	宗教信仰
Height	是	object{}	身高
BloodType	是	object{}	血型
Education	是	object{}	文化程度
MaritalStatus	是	object{}	婚姻状况
VeteranStatus	是	object{}	兵役状况
WorkAddress	是	object{}	服务处所
Career	是	object{}	职业
WWToCity	是	object{}	何时由何地迁来本市(县)
WWHere	是	object{}	何时由何地迁往本址
Date	是	object{}	登记日期

返回示例

```
{
 "log_id": 1301870459,
 "words_result": {
    "BirthAddress": {
       "words": "河南洛阳市郊区"
    },
    "Birthday": {
       "words": "2016-07-28"
    },
    "CardNo": {
       "words": "410311201607282825"
    },
    "Name": {
       "words": "孙翌晨"
    },
    "Nation": {
       "words": "汉族"
    },
    "Relationship": {
       "words": "户主"
    },
    "Sex": {
       "words": "男"
    },
 },
 "words_result_num": 7
}
```


手写文字识别

支持对图片中的手写中文、手写数字进行检测和识别，针对不规则的手写字体进行专项优化，识别准确率可达90%以上。

```
  public void sample(AipOcr client) {
    // 传入可选参数调用接口
    HashMap<String, Object> options = new HashMap<String, Object>();
    options.put("recognize_granularity","big");
    options.put("probability",false);
    options.put("detect_direction",true);
    // 参数为二进制数组
    byte[] file = readFile("test.jpg");
    res = client.handwriting(file, options);
    System.out.println(res.toString(2));

    // 参数图片url
    String url = "http://localhost/test.jpg"
    res = client.handwritingUrl(url, options);
    System.out.println(res.toString(2));
  }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
recognize_granularity	否	string	big、small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true/false	是否返回识别结果中每一行的置信度，默认为false，不返回置信度
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: true：检测朝向； false：不检测朝向

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	float	当请求参数 **probability=true** 时返回该字段，表示识别结果中每一行的置信度值，包含： - **average**： 行置信度平均值 - **variance**：行置信度方差 - **min**：行置信度最小值
direction	否	int32	图像方向，当detect_direction=true时存在 -1:未定义， 0:正向， 1: 逆时针90度， 2:逆时针180度， 3:逆时针270度

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
        "location": {
          "left": 56,
          "top": 0,
          "width": 21,
          "height": 210
        },
        "words": "3"
    }
  ],
  "words_result_num": 1
}
```

飞机行程单识别

支持对飞机行程单的24个字段进行结构化识别，包括电子客票号、印刷序号、姓名、始发站、目的站、航班号、日期、时间、票价、身份证号、承运人、民航发展基金、保险费、燃油附加费、其他税费、合计金额、填开日期、订票渠道、客票级别、座位等级、销售单位号、签注、免费行李、验证码。同时，支持单张行程单上的多航班信息识别。

```
public void sample(AipOcr client) {
  // 传入可选参数调用接口
  HashMap<String, Object> options = new HashMap<String, Object>();
  options.put("multi_detect",true)
  // 参数为二进制数组
  byte[] file = readFile("test.jpg");
  res = client.airTicket(file, options);
  System.out.println(res.toString(2));

  // 参数图片url
  String url = "http://localhost/test.jpg"
  res = client.airTicketUrl(url, options);
  System.out.println(res.toString(2));
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
multi_detect	否	bool	true/false	控制是否开启多航班信息识别功能，**默认值：false** - **true：开启多航班信息识别功能**，开启后返回结果中对应字段格式将改为数组类型 - **false：** 不开启，仅识别单一航班信息

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
name	是	string	姓名
starting_station	是	string	始发站
destination_station	是	string	目的站
flight	是	string	航班号
date	是	string	日期
ticket_number	是	string	电子客票号码
fare	是	string	票价
dev_fund	是	string	民航发展基金/基建费
fuel_surcharge	是	string	燃油附加费
other_tax	是	string	其他税费
ticket_rates	是	string	合计金额
issued_date	是	string	填开日期
id_num	是	string	身份证号
carrier	是	string	承运人
time	是	string	时间
issued_by	是	string	订票渠道
serial_number	是	string	印刷序号
insurance	是	string	保险费
fare_basis	是	string	客票级别
class	是	string	座位等级
agent_code	是	string	销售单位号
endorsement	是	string	签注
allow	是	string	免费行李
ck	是	string	验证码

返回示例

```
 // 识别单航班信息（multi_detect=false，或参数缺省）
{
    "log_id": 7306800033425229106,
    "direction": 0,
    "words_result_num": 18,
    "words_result": {
        "insurance": "20.00",
        "date": "2019-10-22",
        "allow": "20K",
        "flight": "CA6589",
        "issued_by": "中国国际航空服务有限公司",
        "starting_station": "武汉",
        "fare": "260.00",
        "endorsement": "不得签转改期退转",
        "ticket_rates": "350.00",
        "ck": "5866",
```

```
        "serial_number": "51523588676",
        "ticket_number": "7843708871196",
        "fuel_surcharge": "EXEMPT",
        "carrier": "南航",
        "issued_date": "2019-10-30",
        "other_tax": "",
        "fare_basis": "NREOW",
        "id_num": "411201123909020877",
        "destination_station": "合肥",
        "name": "郭达",
        "agent_code": "BJS19197300025",
        "time": "21:25",
        "class": "N",
        "dev_fund": "50.00"
    }
}

// 识别多航班信息（multi_detect=true)
{
    "words_result": {
        "insurance": [
            {
                "word": "XXX"
            }
        ],
        "date": [
            {
                "word": "2019-10-18"
            },
            {
                "word": "2019-10-21"
            }
        ],
        "flight": [
            {
                "word": "CZ3565"
            },
            {
                "word": "CZ3566"
            }
        ],
        "issued_by": [
            {
                "word": "上海携程旅行社有限公司"
            }
        ],
        "starting_station": [
            {
                "word": "北京"
            }
        ],
        "fare": [
            {
                "word": "1080.00"
            }
        ],
        "ticket_rates": [
            {
                "word": "1420.00"
            }
        ],
        "serial_number": [
```

```
            {
                "word": "45956029770"
            }
        ],
        "ticket_number": [
            {
                "word": "7849648364314"
            }
        ],
        "fuel_surcharge": [
            {
                "word": "240.00"
            }
        ],
        "carrier": [
            {
                "word": "南航"
            },
            {
                "word": "南航"
            }
        ],
        "issued_date": [
            {
                "word": "2019-09-18"
            }
        ],
        "other_tax": [],
        "id_num": [
            {
                "word": "0789654700"
            }
        ],
        "destination_station": [
            {
                "word": "深圳"
            },
            {
                "word": "北京"
            }
        ],
        "name": [
            {
                "word": "姚佳"
            }
        ],
        "time": [
            {
                "word": "13:55"
            },
            {
                "word": "16:30"
            }
        ],
        "dev_fund": [
            {
                "word": "100.00"
            }
        ]
    },
    "log_id": "1280814270572920832",
    "words_result_num": 18
}
```

```
}
```

通用机打发票

支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别，包括发票类型、发票号码、发票代码、开票日期、合计金额大写、合计金额小写、商品名称、商品单位、商品单价、商品数量、商品金额、机打代码、机打号码、校验码、销售方名称、销售方纳税人识别号、购买方名称、购买方纳税人识别号、合计税额等。

```
public void sample(AipOcr client) {
  // 传入可选参数调用接口
  HashMap<String, Object> options = new HashMap<String, Object>();
  options.put("location",false);
  // 参数为二进制数组
  byte[] file = readFile("test.jpg");
  res = client.invoice(file, options);
  System.out.println(res.toString(2));

  // 参数图片url
  String url = "http://localhost/test.jpg"
  res = client.invoiceUrl(url, options);
  System.out.println(res.toString(2));
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	string	true/false	是否输出位置信息，true：输出位置信息，false：不输出位置信息，默认false

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	否	string	发票类型
InvoiceCode	否	string	发票代码
InvoiceNum	否	string	发票号码
InvoiceDate	否	string	开票日期
AmountInFiguers	否	string	合计金额小写
AmountInWords	否	string	合计金额大写
CommodityName	否	string	商品名称
CommodityUnit	否	string	商品单位
CommodityPrice	否	string	商品单价
CommodityNum	否	string	商品数量
CommodityAmount	否	string	商品金额
MachineCode	否	string	机打代码
MachineNum	否	string	机打号码
CheckCode	否	string	校验码
SellerName	否	string	销售方名称
SellerRegisterNum	否	string	销售方纳税人识别号
PurchaserName	否	string	购买方名称
PurchaserRegisterNum	否	string	购买方纳税人识别号
TotalTax	否	string	合计税额
Province	否	string	省
City	否	string	市
Time	否	string	时间
SheetNum	否	string	联次

返回示例

```
{
  "log_id": 4423022131715883558,
  "direction": 0,
  "words_result_num": 22,
  "words_result": {
     "City": "",
     "InvoiceNum": "01445096",
     "SellerName": "百度餐饮店",
     "IndustrSort": "生活服务",
     "Province": "广东省",
     "CommodityAmount": [
        {
           "word": "183.00",
           "row": "1"
        }
     ],
     "InvoiceDate": "2020年07月28日",
     "PurchaserName": "中信建投证券股份有限公司",
     "CommodityNum": [],
     "InvoiceCode": "144001901511",
     "CommodityUnit": [],
     "SheetNum": "",
     "PurchaserRegisterNum": "9144223008453480X9",
     "Time": "",
     "CommodityPrice": [],
     "AmountInFiguers": "183.00",
     "AmountInWords": "壹佰捌拾叁元整",
     "CheckCode": "61042119820421061301",
     "TotalTax": "183.00",
     "InvoiceType": "广东通用机打发票",
     "SellerRegisterNum": "61042119820421061301",
     "CommodityName": [
        {
           "word": "餐费",
           "row": "1"
        }
     ]
  }
}
```


护照识别

支持对中国大陆护照个人资料页所有15个字段进行结构化识别，包括国家码、护照号、姓名、姓名拼音、性别、出生地点、出生日期、签发地点（不支持境外签发地）、签发日期、有效期、签发机关、护照类型、国籍、MRZCode1、MRZCode2。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      // 参数为二进制数组
      byte[] file = readFile("test.jpg");
      res = client.passport(file, options);
      System.out.println(res.toString(2));

      // 参数图片url
      String url = "http://localhost/test.jpg"
      res = client.passportUrl(url, options);
      System.out.println(res.toString(2));
    }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
-location	是	uint32	水平坐标
-words	是	string	识别内容

返回示例


```
{
  "log_id": 7377468409496932872,
  "words_result_num": 14,
  "words_result": {
     "国家码": {
        "location": {
           "width": 59,
           "top": 200,
           "left": 762,
           "height": 26
        },
        "words": "CHN"
     },
     "护照签发地点": {
        "location": {
           "width": 236,
           "top": 505,
           "left": 558,
           "height": 43
        },
        "words": "山东/SHANDONG"
     },
     "MRZCode2": {
        "location": {
           "width": 1252,
           "top": 797,
           "left": 145,
           "height": 88
        },
        "words": "E898657303CHNSDM0702<<<<<<<<<<<<<<"
     },
     "有效期至": {
        "location": {
           "width": 287,
           "top": 528,
```

```
            "left": 955,
            "height": 46
        },
        "words": "20261004"
    },
    "签发机关": {
        "location": {
            "width": 271,
            "top": 583,
            "left": 552,
            "height": 42
        },
        "words": "出入境管理局"
    },
    "MRZCode1": {
        "location": {
            "width": 1201,
            "top": 781,
            "left": 162,
            "height": 45
        },
        "words": "PONSUN<<JIAJIA<<<<<<<<<<<<<<<<<<<<<<<<<<<"
    },
    "护照号码": {
        "location": {
            "width": 237,
            "top": 215,
            "left": 994,
            "height": 48
        },
        "words": "E89865730"
    },
    "签发日期": {
        "location": {
            "width": 279,
            "top": 453,
            "left": 955,
            "height": 46
        },
        "words": "20161005"
    },
    "出生地点": {
        "location": {
            "width": 216,
            "top": 429,
            "left": 564,
            "height": 43
        },
        "words": "山东/SHANDONG"
    },
    "姓名": {
        "location": {
            "width": 159,
            "top": 247,
            "left": 581,
            "height": 34
        },
        "words": "孙嘉佳"
    },
    "姓名拼音": {
        "location": {
            "width": 229,
            "top": 279
```

```
            "top": 279,
            "left": 578,
            "height": 41
        },
        "words": "SUN,JIAJIA"
    },
    "国籍": {
        "location": {
            "width": 209,
            "top": 366,
            "left": 695,
            "height": 42
        },
        "words": "中国/CHINESE"
    },
    "生日": {
        "location": {
            "width": 202,
            "top": 382,
            "left": 950,
            "height": 39
        },
        "words": "19950723"
    },
    "性别": {
        "location": {
            "width": 73,
            "top": 357,
            "left": 570,
            "height": 34
        },
        "words": "男/M"
    }
  }
}
```


网约车行程单识别

对各大主要服务商的网约车行程单进行结构化识别，包括滴滴打车、花小猪打车、高德地图、曹操出行、阳光出行，支持识别服务商、行程开始时间、行程结束时间、车型、总金额等16 个关键字段。

```
 public void Demo() {
AipOcr client=new AipOcr("appid","ak","sk");
HashMap<String, Object> options = new HashMap<>();
string url = "https//www.x.com/sample.jpg";
byte[] image = readFile("文件或图片路径");
   byte[] pdf_file =readFile("pdf路径") ;

JSONObject res = null;
  // 调用网约车行程单识别
res = client.OnlineTaxiItinerary(image);
res = client.OnlineTaxiItineraryUrl(url);
res = client.OnlineTaxiItineraryPdf(pdf_file,null);

// 如果有可选参数
options.put("pdf_file_num", 1);
res = client.onlineTaxiItineraryPdf(pdf_file, options);
System.out.println(res.toString(2));
}
```

请求参数详情

参数	是否必选	类型	说明
image	和url二选一	string	图像数据，base64编码后进行urlencode，base64编码去除编码头（data:image/jpeg;base64），要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效，请注意关闭URL防盗链
pdf_file	和image/url 三选一	string	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
返回参数详情			

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ ServiceProvider	是	string	服务商
+ StartTime	是	string	行程开始时间
+ EndTime	是	string	行程结束时间
+ Phone	是	string	行程人手机号
+ ApplicationDate	是	string	申请日期
+ TotalFare	是	string	总金额
+ ItemNum	是	array	行程信息中包含的行程数量
+ Items	是	array	行程信息
++ ItemId	是	string	行程信息的对应序号
++ PickupTime	是	string	上车时间
++ PickupDate	是	string	上车日期
++ CarType	是	string	车型
++ Distance	是	string	里程
++ StartPlace	是	string	起点
++ DestinationPlace	是	string	终点
++ City	是	string	城市
++ Fare	是	string	金额
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 1385196013945356288,
  "words_result_num": 7
  "words_result": {
     "TotalFare": "2316",
     "EndTime": "2020-07-30 19:00",
     "Phone": "13000000000",
     "ServiceProvider": "滴滴企业版",
     "StartTime": "2020-07-01 16:00",
     "ApplicationDate": "2017-12-08",
     "ItemId": "3"
     "items": [
        {
           "ItemId": "1",
           "StartPlace": "鱼化寨地铁-D口",
           "PickupTime": "16:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.7",
           "PickupDate": "20-07-01",
           "DestinationPlace": "创新港",
           "Fare": "20.86"
        },
        {
           "ItemId": "2",
           "StartPlace": "科学园东门",
           "PickupTime": "14:56",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "91",
           "PickupDate": "20-07-02",
           "DestinationPlace": "鱼化寨地铁站",
           "Fare": "18.58"
        },
        {
           "ItemId": "3",
           "StartPlace": "中俄丝路创新园东门",
           "PickupTime": "19:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.1",
           "PickupDate": "20-07-30",
           "DestinationPlace": "新门地铁站",
           "Fare": "20.38"
        },
     ],
  },
}
```


磅单识别

结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单。

```
 public void Demo() {
AipOcr client=new AipOcr("appid","ak","sk");
HashMap<String, Object> options = new HashMap<>();
string url = "https//www.x.com/sample.jpg";
byte[] pdf_file = ;
byte[] image = readFile("文件或图片路径");
JSONObject res = null; // 调用磅单识别
res = client.WeightNote(image,null);
res = client.WeightNoteUrl(url,null);
res = client.WeightNotePdf(pdf_file,null);

// 如果有可选参数
options.put("pdf_file_num", 1);
options.put("probability", false);

res = client.weightNote(image, options);
System.out.println(res.toString(2));

res = client.weightNoteUrl(url, options);
System.out.println(res.toString(2));

res = client.weightNotePdf(pdf_file, options);
System.out.println(res.toString(2));
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ PlateNum	是	object	车牌号
+ PrintTime	是	object	打印时间
+ CrossWeight	是	object	毛重
+ TareWeight	是	object	皮重
+ NetWeight	是	object	净重
+ SendingCompany	是	object	发货单位
+ ReceivingCompany	是	object	收货单位
+ DeliveryNumber	是	object	单号
++ word	是	string	字段识别结果，以上各字段均包含此参数
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
返回示例			

```
{
  "words_result": [
    {
      "TareWeight": [
      {
            "word": "14.20"
        }
      ],
      "CrossWeight": [
        {
          "word": "50.70"
        }
      ],
      "PlateNum": [
        {
          "word": "京A12356"
        }
      ],
      "SendingCompany": [
       {
          "word": "宣化县耿矿煤业有限公司"
        }
      ],
      "DeliveryNumber": [
       {
          "word": "278933000"
        }
      ],
      "ReceivingCompany": [
        {
          "word": "宁夏市京裕达实业公司"
        }
      ],
      "PrintTime": [
        {
          "word": "2020年1月1日"
        }
      ],
      "NetWeight": [
        {
          "word": "36.50"
        }
      ]
    }
  ],
  "words_result_num": 1,
  "log_id": 1428311410130160734
}
```


医疗费用明细识别

支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含项目类型、项目名称、单价、数量、规格、金额，其中北京地区识别效果最佳。

```
 public void Demo() {
AipOcr client=new AipOcr("appid","ak","sk");
HashMap<String, Object> options = new HashMap<>();
byte[] image = readFile("文件或图片路径");
   string url = "https//www.x.com/sample.jpg";

JSONObject res = null;
  // 调用医疗费用明细识别
res = client.MedicalDetail(image,null);
res = client.MedicalDetailUrl(url,null);

// 如果有可选参数
options.put("probability", false);

res = client.medicalDetail(image, options);
System.out.println(res.toString(2));

res = client.medicalDetailUrl(url, options);
System.out.println(res.toString(2));
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省** - **false**：不返回字段位置信息 - **true**：返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ Name	是	object	姓名
+ Date	是	object	日期
+ PatientID	是	object	病人ID
+ TotalAmount	是	object	总金额
+ word	是	string	字段识别结果，以上各字段均包含此参数
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+ CostDetail	是	array[]	项目明细

CostDetail字段包含多个Array，每个数组包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：项目类型、项目名称、单价、数量、规格、金额
++ word	word_name字段对应的识别结果

返回示例


```
{
  "log_id": 1397090241579319296,
  "words_result_num": 5,
  "words_result": {
    "PatientID": {
        "word": "23683829"
    },
    "TotalAmount": {
        "word": "600.00"
    },
    "Date": {
        "word": "2020年11月04日"
    },
    "Name": {
        "word": "范浩"
    },
    "CostDetail": [
        [
            {
                "word_name": "清单项目名称",
                "word": "普通过敏原(新)筛查"
            },
            {
                "word_name": "单价",
                "word": "580.00"
```

```
            },
            {
                "word_name": "数量",
                "word": "1.00"
            },
            {
                "word_name": "金额",
                "word": "580.00"
            },
            {
                "word_name": "规格",
                "word": "次"
            },
            {
                "word_name": "项目类型",
                "word": "化验费"
            }
        ],
        [
            {
                "word_name": "清单项目名称",
                "word": "总IgE测定"
            },
            {
                "word_name": "单价",
                "word": "20.00"
            },
            {
                "word_name": "数量",
                "word": "1.00"
            },
            {
                "word_name": "金额",
                "word": "20.00"
            },
            {
                "word_name": "规格",
                "word": "次"
            },
            {
                "word_name": "项目类型",
                "word": "化验费"
            }
        ]
    ],
  },
}
```


办公文档识别

可对办公类文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景。

```
public void docAnalysisOffice() {
byte[] image = readFile("文件或图片路径");
String url = "https://www.x.com/sample.jpg";
byte[] pdf_file =readFile("pdf路径") ;

// 调用办公文档识别
      JSONObject fileResult = client.docAnalysisOffice(image, null);
      JSONObject urlResult = client.docAnalysisOfficeUrl(url,null);
      JSONObject pdfResult = client.docAnalysisOfficePdf(pdf_file, 1, null);
      System.out.println("fileResult:"+fileResult);
      System.out.println("urlResult:"+urlResult);
      System.out.println("pdfResult:"+pdfResult);

// 如果有可选参数
HashMap<String, Object> option = new HashMap<>();
      option.put("result_type", "small");
      JSONObject fileResultOption = client.docAnalysisOffice(image, option);
      JSONObject urlResultOption = client.docAnalysisOfficeUrl(url, option);
      JSONObject pdfResultOption = client.docAnalysisOfficePdf(pdf_file, 1, option);
      System.out.println("urlResultOption:"+urlResultOption);
      System.out.println("fileResultOption:"+fileResultOption);
      System.out.println("pdfResultOption:"+pdfResultOption);
  }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
language_type	否	string	CHN_ENG/ ENG	识别语言类型，默认为CHN_ENG 可选值包括： =CHN_ENG：中英文 =ENG：英文
result_type	否	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	否	string	true/false	是否返回每行识别结果的置信度。默认为false
disp_line_poly	否	string	true/false	是否返回每行的四角点坐标。默认为false
words_type	否	string	handwring_only/ handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	否	string	true/false	是否分析文档版面：包括layout（图、表、标题、段落、目录）；attribute（栏、页眉、页脚、页码、脚注）的分析输出
erase_seal	否	string	true/false	是否先擦除水印、印章后再识别文档

返回参数详情

字段	是否必选	类型	说明

log_id	是	uint64	唯一的log id，用于问题定位
img_direction	否	int32	detect_direction=true 时返回该字段。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	是	uint32	识别结果数，表示results的元素个数
results	是	array[]	识别结果数组
+ words_type	是	string	文字属性（手写、印刷），handwriting 手写，print 印刷
+ words	是	array[]	整行的识别结果数组
++ line_probability	否	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++ average	否	float	行置信度
+++ min	否	float	整行中单字的最低置信度
++ word	是	float	整行的识别结果
++ poly_location	否	array[]	是否返回每行的四角点坐标，disp_line_poly=true时返回
++ words_location	是	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位定位位置的长方形的宽度
+++ height	是	uint32	表示位置的长方形的高度
+ chars	否	array[]	result_type=small时返回。单字符结果数组
++ char	否	string	result_type=small时返回。每个单字的内容
++ chars_location	否	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
layouts_num	否	uint32	版面分析结果数，表示layout的元素个数
layouts	否	array[]	每个「栏：section」里面的文档版面模块数组，包含表格、图、段落文本、标题、目录等5个模块；每个模块的坐标位置；段落文本和表格内文本内容对应的行序号id。

+ layout	否	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title ，目录:contents
+ layout_location	否	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	水平坐标（坐标0点为左上角）
+ layout_idx	否	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
sec_rows	否	uint32	将所有的版面中的「栏:section」内容表示成 M x N 的网格，sec_rows = M
sec_cols	否	uint32	将所有的版面中的「分栏」内容表示成 M x N 的网格，sec_cols = N
sections	否	array[]	一张图片中包含的5大版面属性，包含：栏，页眉，页脚，页码，脚注，该数组里有属性的标签、属性的位置、属性所包含文本内容的id序号。 其中，栏（section）里面包含5个模块内容，有：表格、图、段落文本、标题、目录（在返回参数layouts里输出）。
+ attribute	否	string	版面分析的属性标签结果，栏:section, 页眉:header, 页脚:footer, 页码:number，脚注:footnote。
+ attri_location	否	array[]	版面分析的属性所在位置，四个顶点: 左上，右上，右下，左下
++ x	否	uint32	水平坐标（坐标0点为左上角）
++ y	否	uint32	水平坐标（坐标0点为左上角）
+ sec_idx	否	string	sections返回参数中的5个版面属性里，包含的内容序号标识
++ idx	否	string	sections返回参数中的5个版面属性里，每个属性下包含的文本行id序号
++ para_idx	否	string	当且仅当attribute=section时才会返回。表示，返回参数中的「栏：section」里面，所包含的表格、图、段落文本、标题、目录等5个模块返回的顺序号id（即layouts返回结果中，每个模块的返回顺序号）
++ row_idx	否	string	当且仅当attribute=section时才会返回。表示，将所有栏表示成 M xN 的网格，所属网格的行的id。
++ col_idx	否	string	当且仅当attribute=section时才会返回。表示，将所有栏表示成 M xN 的网格，所属网格的列的id。

返回示例


```
{
"results_num": 5,
"log_id": "1410491260247950412",
"results": [
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 88,
    "left": 442,
```

```
    "width": 142,
    "height": 49
   },
   "word": "行程单"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 241,
    "left": 439,
    "width": 393,
    "height": 37
   },
   "word": "美国东海岸名校8天7晚"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 318,
    "left": 436,
    "width": 774,
    "height": 31
   },
   "word": "国会大厦位于华盛顿25米高的国会山上,是美国的心脏建筑。"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 374,
    "left": 434,
    "width": 805,
    "height": 31
   },
   "word": "中央顶楼上的大圆顶上立有一尊6米高的自由女神青铜雕像。"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 431,
    "left": 436,
    "width": 556,
    "height": 31
   },
   "word": "东面的大草坪是历届总统举行就职典礼的地方。"
  }
 }
]
}
```


印章识别

检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别。

```
 public void seal() {
byte[] image = readFile("文件或图片路径");
String url = "https://www.x.com/sample.jpg";
byte[] pdf_file =readFile("pdf路径") ;

// 调用印章识别
      JSONObject fileResult = client.seal(image);
      JSONObject urlResult = client.sealUrl(url);
      JSONObject pdfResult = client.sealPdf(pdf_file, 1);
      System.out.println("fileResult:"+fileResult);
      System.out.println("urlResult:"+urlResult);
      System.out.println("pdfResult:"+pdfResult);
  }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
result_num	是	uint32	识别结果数，表示results的元素个数
result	是	array[]	定位结果数组
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ probability	是	float	每一个识别结果的置信度值
+ type	是	string	印章的类别，共有circle（圆章），ellipse（椭圆章），rectangle（方章）三种
+ major	是	object	主字段内容
++ words	是	string	主字段识别内容，即章内上环弯曲文字结果
++ probability	是	float	主字段识别内容的置信度
+ minor	是	array[]	其他字段内容，即除主字段外的文字识别内容均放置于该参数中返回，若章内不存在其他字段文字，则该参数为空
++ words	是	string	其他字段识别内容
++ probability	是	float	其他字段识别内容的置信度
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
"result": [
 {
  "major": {
   "probability": 0.99759155511856,
   "words": "峨眉山旅游股份有限公司成都峨眉山雪芽大酒店分公司"
  },
  "minor": [
   {
    "probability": 0.99994027614594,
    "words": "前厅部"
   }
  ],
  "probability": 0.9936261177063,
  "location": {
   "top": 594,
   "left": 918,
   "width": 150,
   "height": 142
  },
  "type": "circle"
 }
],
"log_id": "1349006147834609664",
"result_num": 1
}
```


身份证混贴识别

身份证混贴识别支持自动检测与识别身份证正反面在同一张图片上的场景，一次识别图片中身份证正反面所有字段。

支持对二代居民身份证正反面所有8个字段进行结构化识别，包括姓名、性别、民族、出生日期、住址、身份证号、签发机关、有效期限，识别准确率超过99%；同时支持身份证正面头像检测，并返回头像切片的base64编码及位置信息。

同时，支持对用户上传的身份证图片进行图像风险和质量检测，可识别图片是否为复印件或临时身份证，是否被翻拍或编辑，是否存在正反颠倒、模糊、欠曝、过曝等质量问题。

```
public void multiIdcard() {
byte[] image = readFile("文件或图片路径");
String url = "https://www.x.com/sample.jpg";

// 调用身份证混贴识别
      JSONObject fileResult = client.multiIdcard(image, null);
      JSONObject urlResult = client.multiIdcardUrl(url, null);
      System.out.println("fileResult:"+fileResult);
      System.out.println("urlResult:"+urlResult);

// 如果有可选参数
HashMap<String, Object> option = new HashMap<>();
option.put("detect_risk", "true");
      JSONObject fileResultOption = client.multiIdcard(image, option);
      JSONObject urlResultOption = client.multiIdcardUrl(url, option);
      System.out.println("fileResultOption:"+fileResultOption);
      System.out.println("urlResultOption:"+urlResultOption);
  }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_risk	否	string	true/false	是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)功能，默认不开启，即：false。 - true：开启，请查看返回参数risk_type； - false：不开启
detect_quality	否	string	true/false	是否开启身份证质量类型(边框/四角不完整、头像或关键字段被遮挡/马赛克)检测功能，默认不开启，即：false。 - true：开启，请查看返回参数card_quality； - false：不开启
detect_photo	否	string	true/false	是否检测头像内容，默认不检测。可选值：true-检测头像并返回头像的 base64 编码及位置信息
detect_card	否	string	true/false	是否检测身份证进行裁剪，默认不检测。可选值：true-检测身份证并返回证照的 base64 编码及位置信息
返回参数详情				

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
+ card_info	是	object	识别结果信息，包含身份证的位置信息、风险检测信息、质量检测信息、正反面信息、方向信息
++ card_location	是	object	身份证的位置信息（坐标0点为左上角）
++ card_type	是	string	- idcard_front：身份证正面（头像面） - idcard_back：身份证反面（国徽面）
direction	是	int32	图像方向。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
++ image_status	是	string	normal-识别正常 reversed_side-身份证正反面颠倒 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 other_type_card-其他类型证照 over_exposure-身份证关键字段反光或过曝

			over_dark-身份证欠曝（亮度过低） unknown-未知状态
++ risk_type	否	string	输入参数 detect_risk = true 时，则返回该字段识别身份证类型: normal-正常身份证；copy-复印件；temporary-临时身份证；screen-翻拍；unknown-其他未知情况
++ edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
card_quality	否	object	输入参数 detect_quality = true 时，则返回该字段识别身份证**质量类型**: IsClear - 是否清晰； IsComplete - 是否边框/四角完整； IsNoCover - 是否头像、关键字段无遮挡/马赛克。 及对应的概率：IsComplete_propobility、IsNoCover_propobility、IsClear_propobility，值在0-1之间，值越大表示图像质量越好。 **默认阈值：**当 IsComplete_propobility 超过0.5时，IsComplete返回1，低于0.5，则返回0。IsNoCover_propobility、IsClear_propobility 同上
++ photo	否	string	当请求参数 detect_photo = true时返回，头像切图的 base64 编码（无编码头，需自行处理）
++ photo_location	否	array[]	当请求参数 detect_photo = true时返回，头像的位置信息（坐标0点为左上角）
++ card_image	否	string	当请求参数 detect_card = true时返回，身份证裁剪切图的 base64 编码（无编码头，需自行处理）
++ idcard_number_type	是	string	用于校验身份证号码、性别、出生是否一致，输出结果及其对应关系如下： - **-1：** 身份证正面所有字段全为空 **0：** 身份证证号不合法，此情况下不返回身份证证号 **1：** 身份证证号和性别、出生信息一致 **2：** 身份证证号和性别、出生信息都不一致 **3：** 身份证证号和出生信息不一致 **4：** 身份证证号和性别信息不一致
+ card_result	是	object	识别结果
++ words	否	string	识别结果字符串
++ location	否	object	位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位位置的长方形的宽度
+++ height	否	uint32	表示定位位置的长方形的高度

返回示例


```
{
  "words_result": [
    {
      "card_info": {
        "card_location": {
          "top": 121,
          "left": 687,
          "width": 501,
          "height": 323
```

```
        },
        "risk_type": "normal",
        "idcard_number_type": 1,
        "edit_tool": "",
        "image_status": "normal",
        "card_type": "idcard_front",
        "direction": 0
    },
    "card_result": {
        "姓名": {
            "words": "王媛",
            "location": {
                "top": 158,
                "left": 788,
                "width": 67,
                "height": 26
            }
        },
        "民族": {
            "words": "汉",
            "location": {
                "top": 204,
                "left": 884,
                "width": 19,
                "height": 21
            }
        },
        "住址": {
            "words": "北京市海淀区西北旺东路10号院",
            "location": {
                "top": 282,
                "left": 782,
                "width": 205,
                "height": 51
            }
        },
        "公民身份号码": {
            "words": "410433199510104518",
            "location": {
                "top": 385,
                "left": 863,
                "width": 266,
                "height": 29
            }
        },
        "出生": {
            "words": "19951010",
            "location": {
                "top": 240,
                "left": 784,
                "width": 163,
                "height": 25
            }
        },
        "性别": {
            "words": "女",
            "location": {
                "top": 202,
                "left": 789,
                "width": 19,
                "height": 22
            }
        }
```

```
            }
        },
        {
            "card_info": {
                "card_location": {
                    "top": 526,
                    "left": 66,
                    "width": 524,
                    "height": 323
                },
                "risk_type": "normal",
                "edit_tool": "",
                "image_status": "normal",
                "card_type": "idcard_back",
                "direction": 0
            },
            "card_result": {
                "失效日期": {
                    "words": "20350413",
                    "location": {
                        "top": 801,
                        "left": 394,
                        "width": 95,
                        "height": 19
                    }
                },
                "签发机关": {
                    "words": "北京市公安局海淀分局",
                    "location": {
                        "top": 760,
                        "left": 290,
                        "width": 180,
                        "height": 19
                    }
                },
                "签发日期": {
                    "words": "20150413",
                    "location": {
                        "top": 801,
                        "left": 289,
                        "width": 89,
                        "height": 19
                    }
                }
            }
        }
    ],
    "direction": 0,
    "log_id": 1397109704106180608
}
```


车辆证照混贴识别

车辆证照混贴识别接口支持自动检测与识别行驶证、驾驶证混贴图片，即识别机动车行驶证主页及副页、机动车驾驶证主页及副页在同一张图片上的场景，一次性识别图片中多个行驶证、驾驶证的所有字段。

支持对机动车行驶证主页及副页所有22个字段进行结构化识别，包括号牌号码、车辆类型、所有人、品牌型号、车辆识别代码、发动机号码、核定载人数、质量、尺寸、检验记录等；支持对机动车驾驶证正页及副页所有15个字段进行结构化识别，包括证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效期限、发证单位、档案编号等。

```
 public void mixedMultiVehicle() {
byte[] image = readFile("文件或图片路径");
String url = "https://www.x.com/sample.jpg";

// 调用车辆证照混贴识别
      JSONObject fileResult = client.mixedMultiVehicle(image, null);
      JSONObject urlResult = client.mixedMultiVehicleUrl(url, null);
      System.out.println("fileResult:"+fileResult);
      System.out.println("urlResult:"+urlResult);

// 如果有可选参数
HashMap<String, Object> option = new HashMap<>();
      option.put("unified", "true");
      JSONObject fileResultOption = client.mixedMultiVehicle(image, option);
      JSONObject urlResultOption = client.mixedMultiVehicleUrl(url, option);
      System.out.println("fileResultOption:"+fileResultOption);
      System.out.println("urlResultOption:"+urlResultOption);
  }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	否	string	true/false	- **false：默认值**不进行图像方向自动矫正 - **true:** 开启图像方向自动矫正功能，可对旋转 90/180/270 度的图片进行自动矫正并识别
unified	否	string	true/false	- **false：默认值**，不进行归一化处理 - **true：**对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出
返回参数详情				

字段	必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
card_type	是	string	证件类型，vehicle_front、vehicle_back、driving_front和driving_back，分别对应行驶证正、副页，驾驶证正、副页
direction	否	int32	图像方向，当图像旋转时，返回该参数。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
probability	是	uint32	检测到证件的置信度
location	是	object	证件位置数组（坐标0点为左上角）
+ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+ width	是	uint32	表示定位位置的长方形的宽度
+ height	是	uint32	表示定位位置的长方形的高度
license_info	是	object	识别结果信息，包含图片中多个行驶证、驾驶证的识别结果
+ word_name	是	string	字段名，如果license_info对应行驶证，字段名包含号牌号码、车辆类型、所有人、品牌型号、车辆识别代码、发动机号码、核定载人数、质量、尺寸、检验记录等；如果license_info对应驾驶证，字段名包含证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效期限、发证单位、档案编号等
+ word	是	string	word_name字段对应的识别结果

返回示例


```
{
  "words_result":[
     {
        "license_info":[
           {
              "word_name":"证号",
              "word":"320621196512031267"
           },
           {
              "word_name":"姓名",
              "word":"程成"
           },
           {
              "word_name":"性别",
              "word":"男"
           },
           {
              "word_name":"国籍",
```

```
            "word_name":"国籍",
            "word":"中国"
         },
         {
            "word_name":"住址",
            "word":"江苏省南通市"
         },
         {
            "word_name":"出生日期",
            "word":"19651203"
         },
         {
            "word_name":"初次领证日期",
            "word":"20110311"
         },
         {
            "word_name":"准驾车型",
            "word":"B2"
         },
         {
            "word_name":"有效起始日期",
            "word":"20170311"
         },
         {
            "word_name":"失效日期",
            "word":"20270311"
         },
         {
            "word_name":"发证单位",
            "word":"江苏省南通市公安局交通警察支队"
         }
      ],
      "probability":0.99496907,
      "location":{
         "top":14,
         "left":15,
         "width":701,
         "height":459
      },
      "card_type":"driving_front",
      "direction":0
   },
   {
      "license_info":[
         {
            "word_name":"证号",
            "word":"320621196512031267"
         },
         {
            "word_name":"姓名",
            "word":"程成"
         },
         {
            "word_name":"档案编号",
            "word":"320601650706"
         },
         {
            "word_name":"记录",
            "word":"自2017年03月06日至有效起始日期有效。请于每个记分周期结束后三十日接受审验。无记分的，免予本次审验。"
         }
      ],
      "probability":0.9815229177,
```

```
        "location":{
            "top":7,
            "left":731,
            "width":650,
            "height":456
        },
        "card_type":"driving_back",
        "direction":0
    },
    {
        "license_info":[
            {
                "word_name":"号牌号码",
                "word":"京A10D08"
            },
            {
                "word_name":"车辆类型",
                "word":"小型面包车"
            },
            {
                "word_name":"所有人",
                "word":"王京"
            },
            {
                "word_name":"住址",
                "word":"北京市石景山区"
            },
            {
                "word_name":"使用性质",
                "word":"非营运"
            },
            {
                "word_name":"品牌型号",
                "word":"东风牌EQ6456PF"
            },
            {
                "word_name":"车辆识别代号",
                "word":"LGK022K69A9240616"
            },
            {
                "word_name":"发动机号码",
                "word":"10066180"
            },
            {
                "word_name":"注册日期",
                "word":"20100707"
            },
            {
                "word_name":"发证日期",
                "word":"20191020"
            },
            {
                "word_name":"发证单位",
                "word":"北京市公安局交通警察支队"
            }
        ],
        "probability":0.9907341003,
        "location":{
            "top":532,
            "left":736,
            "width":622,
            "height":415
```

```
        },
        "card_type":"vehicle_front",
        "direction":0
    },
    {
        "license_info":[
            {
                "word_name":"号牌号码",
                "word":"京A10D08"
            },
            {
                "word_name":"备注",
                "word":""
            },
            {
                "word_name":"整备质量",
                "word":"985kg"
            },
            {
                "word_name":"核定载质量",
                "word":""
            },
            {
                "word_name":"外廓尺寸",
                "word":"3660X1560X1925mm"
            },
            {
                "word_name":"核定载人数",
                "word":"7人"
            },
            {
                "word_name":"总质量",
                "word":"1565kg"
            },
            {
                "word_name":"准牵引总质量",
                "word":""
            },
            {
                "word_name":"档案编号",
                "word":""
            },
            {
                "word_name":"检验记录",
                "word":"2020年07月京A(03)"
            },
            {
                "word_name":"燃油类型",
                "word":"汽油"
            }
        ],
        "probability":0.9925737381,
        "location":{
            "top":554,
            "left":12,
            "width":609,
            "height":416
        },
        "card_type":"vehicle_back",
        "direction":0
    }
],
"log_id":1420339917916065252
```

```
    "log_id":1420339917916065252,
    "words_result_num":4
}
```

机动车登记证书识别

支持对机动车登记证书的15个关键字段进行结构化识别，包括编号、机动车所有人、登记机关、登记日期、登记编号、车辆类型等，同时支持检测发证机关章。

```
 public void vehicleRegistrationCertificate() {
byte[] image = readFile("文件或图片路径");
String url = "https://www.x.com/sample.jpg";

// 调用机动车登记证书识别
      JSONObject fileResult = client.vehicleRegistrationCertificate(image);
      JSONObject urlResult = client.vehicleRegistrationCertificateUrl(url);
      System.out.println("fileResult:"+fileResult);
      System.out.println("urlResult:"+urlResult);
  }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	是	int32	图像方向 - 1：未定义， 0：正向， 1：逆时针90度， 2：逆时针180度， 3：逆时针270度
words_result	是	object	识别结果数组
+ number	是	object	编号
+ name_idcard_no	是	object	机动车所有人/身份证明名称/号码
+ registration_authority	是	object	登记机关
+ registration_date	是	object	登记日期
+ registration_num	是	object	机动车登记编号
+ vehicle_model	是	object	车辆型号
+ vehicle_type	是	object	车辆类型
+ vin	是	object	车架号
+ engine_num	是	object	发动机号
+ seating_capacity	是	object	核定载客
+ body_color	是	object	车身颜色
+ nature_of_use	是	object	使用性质
+ date_of_production	是	object	出厂日期
+ date_of_issue	是	object	发证日期
+ seal_of_issue_authority	是	object	发证机关章，1表示有，0表示无

返回示例

```
{
  "words_result": {
    "registration_authority": {
      "words": "江苏省昆山市公安局交通巡逻警察大队"
    },
    "vehicle_model": {
      "words": "DHW6691R8CEE"
    },
    "vehicle_type": {
      "words": "小型普通客车"
    },
    "registration_num": {
      "words": "苏A88FF2"
    },
    "engine_num": {
      "words": "2005533"
    },
    "number": {
      "words": "32004574998"
    },
    "body_color": {
      "words": "白"
    },
    "registration_date": {
      "words": "2016-06-06"
    },
    "date_of_production": {
      "words": "2016-03-11"
    },
    "seating_capacity": {
      "words": "7"
    },
    "date_of_issue": {
      "words": "2016-06-06"
    },
    "nature_of_use": {
      "words": "非营运"
    },
    "vin": {
      "words": "LVHRR8836G5004527"
    },
    "seal_of_issue_authority": {
      "words": "1"
    },
    "name_idcard_no": {
      "words": "汽车服务有限公司/统一社会信用代码/91320583088874412F"
    }
  },
  "log_id": 1392089398849306624,
  "direction": "0"
}
```


智能财务票据识别

支持财务场景中13种常见票据的分类及结构化识别，包括增值税发票、卷票、机打发票、定额发票、火车票、出租车票、网约车行程单、飞机行程单、汽车票、过路过桥费、船票、机动车/二手车销售发票。支持多张不同种类票据在同一张图片上的混贴场景，可返回每张票据的位置、种类及票面信息的结构化识别结果。

```
    public void multipleInvoice() {
byte[] image = readFile("文件或图片路径");
String url = "https://www.x.com/sample.jpg";
byte[] pdf_file =readFile("pdf路径") ;

// 调用智能财务票据识别
      JSONObject fileResult = client.multipleInvoice(image null);
      JSONObject urlResult = client.multipleInvoiceUrl(url, null);
      JSONObject pdfResult = client.multipleInvoicePdf(pdf_file, 1, null);
      System.out.println("fileResult:"+fileResult);
      System.out.println("urlResult:"+urlResult);
      System.out.println("pdfResult:"+pdfResult);

// 如果有可选参数
HashMap<String, Object> option = new HashMap<>();
      option.put("verify_parameter", "true");
      JSONObject fileResultOption = client.multipleInvoice(image, option);
      JSONObject urlResultOption = client.multipleInvoiceUrl(url, option);
      JSONObject pdfResultOption = client.multipleInvoicePdf(pdf_file, 1, option);
      System.out.println("fileResultOption:"+fileResultOption);
      System.out.println("urlResultOption:"+urlResultOption);
      System.out.println("pdfResultOption:"+pdfResultOption);
  }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和 url/pdf_file 三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
verify_parameter	否	string	true/false	是否开启验真，默认为 false，即不开启，**当为 true 时，返回匹配发票验真接口所需的6要素信息，具体返回信息详见末尾说明**
probability	否	string	true/false	是否返回字段置信度，默认为 false ，即不返回
location	否	string	true/false	是否返回字段位置坐标，默认为 false，即不返回

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ probability	是	string	表示单张票据分类的置信度
+ left	是	string	表示单张票据定位位置的长方形左上顶点的水平坐标
+ top	是	string	表示单张票据定位位置的长方形左上顶点的垂直坐标
+ width	是	string	表示单张票据定位位置的长方形的宽度
+ height	是	string	表示单张票据定位位置的长方形的高度
+ type	是	string	每一张票据的种类
+ result	是	array[]	单张票据的识别结果数组
type 字段会返回以下17种结果，每种结果对应的票据类型详见下表			

type 返回结果	说明
vat_invoice	增值税发票
taxi_receipt	出租车票
train_ticket	火车票
quota_invoice	定额发票
air_ticket	机票行程单
roll_normal_invoice	卷票
printed_invoice	机打发票
bus_ticket	汽车票
toll_invoice	过路过桥费发票
ferry_ticket	船票
motor_vehicle_invoice	机动车销售发票
used_vehicle_invoice	二手车发票
taxi_online_ticket	网约车行程单
limit_invoice	限额发票
shopping_receipt	购物小票
pos_invoice	POS小票
others	其他

type 的返回结果为 vat_invoice，即“增值税发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceTypeOr	是	array[]	发票名称

g			
++ InvoiceType	是	array[]	增值税发票的细分类型。不同细分类型的增值税发票发票输出：普通发票、专用发票、电子普通发票、电子专用发票、通行费电子普票、区块链发票、通用机打电子发票
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ InvoiceCodeConfirm	是	array[]	发票代码的辅助校验码，一般业务情景可忽略
++ InvoiceNumConfirm	是	array[]	发票号码的辅助校验码，一般业务情景可忽略
++ CheckCode	是	array[]	校验码。增值税专票无此参数
++ InvoiceDate	是	array[]	开票日期
++ PurchaserName	是	array[]	购方名称
++ PurchaserRegisterNum	是	array[]	购方纳税人识别号
++ PurchaserAddress	是	array[]	购方地址及电话
++ PurchaserBank	是	array[]	购方开户行及账号
++ Password	是	array[]	密码区
++ Province	是	array[]	省
++ City	是	array[]	市
++ SheetNum	是	array[]	联次信息。**专票**第一联到第三联分别输出：第一联：记账联、第二联：抵扣联、第三联：发票联；**普通发票**第一联到第二联分别输出：第一联：记账联、第二联：发票联
++ Agent	是	array[]	是否代开
++ OnlinePay	是	String	电子支付标识。仅区块链发票含有此参数
++ SellerName	是	array[]	销售方名称
++ SellerRegisterNum	是	array[]	销售方纳税人识别号
++ SellerAddress	是	array[]	销售方地址及电话
++ SellerBank	是	array[]	销售方开户行及账号
++			

++ TotalAmount	是	array[]	合计金额
++ TotalTax	是	array[]	合计税额
++ AmountInWords	是	array[]	价税合计(大写)
++ AmountInFigures	是	array[]	价税合计(小写)
++ Payee	是	array[]	收款人
++ Checker	是	array[]	复核
++ NoteDrawer	是	array[]	开票人
++ Remarks	是	array[]	备注
+++ word	是	string	识别结果字符串，以上各字段均包含
++ CommodityName	是	array[]	货物名称
++ CommodityType	是	array[]	规格型号
++ CommodityUnit	是	array[]	单位
++ CommodityNum	是	array[]	数量
++ CommodityPrice	是	array[]	单价
++ CommodityAmount	是	array[]	金额
++ CommodityTaxRate	是	array[]	税率
++ CommodityTax	是	array[]	税额
++ CommodityPlateNum	是	array[]	车牌号。仅通行费增值税电子普通发票含有此参数
++ CommodityVehicleType	是	array[]	类型。仅通行费增值税电子普通发票含有此参数
++			

CommodityStartDate	是	array[]	通行日期起。仅通行费增值税电子普通发票含有此参数
++ CommodityEndDate	是	array[]	通行日期止。仅通行费增值税电子普通发票含有此参数
+++ row	是	uint32	行号，以上各字段均包含
+++ word	是	string	内容，以上各字段均包含

type 的返回结果为 taxi_receipt，即“出租车票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceCode	是	array[]	发票代号
++ InvoiceNum	是	array[]	发票号码
++ TaxiNum	是	array[]	车牌号
++ Date	是	array[]	日期
++ Time	是	array[]	上下车时间
++ PickupTime	是	array[]	上车时间
++ DropoffTime	是	array[]	下车时间
++ Fare	是	array[]	金额
++ FuelOilSurcharge	是	array[]	燃油附加费
++ CallServiceSurcharge	是	array[]	叫车服务费
++ TotalFare	是	array[]	总金额
++ Location	是	array[]	开票城市
++ Province	是	array[]	省
++ City	是	array[]	市
++ PricePerkm	是	array[]	单价
++ Distance	是	array[]	里程
+++ word	是	string	识别结果字符串，以上各字段均包含
type 的返回结果为 train_ticket，即“火车票”时，识别结果的返回字段如下			

字段	是否必选	类型	说明
++ ticket_num	是	array[]	车票号
++ starting_station	是	array[]	始发站
++ train_num	是	array[]	车次号
++ destination_station	是	array[]	到达站
++ date	是	array[]	出发日期
++ ticket_rates	是	array[]	车票金额
++ seat_category	是	array[]	席别
++ name	是	array[]	乘客姓名
++ ID_card	是	array[]	身份证号
++ serial_number	是	array[]	序列号
++ sales_station	是	array[]	售站
++ time	是	array[]	时间
++ seat_num	是	array[]	座位号
++ Waiting_area	是	array[]	候检区
+++ word	是	string	识别结果字符串，以上各字段均包含
type 的返回结果为 quota_invoice，即“定额发票”时，识别结果的返回字段如下			

字段	是否必选	类型	说明
++ invoice_code	是	array[]	发票代码
++ invoice_number	是	array[]	发票号码
++ invoice_rate	是	array[]	金额
++ invoice_rate_in_figure	是	array[]	金额小写
++ invoice_rate_in_word	是	array[]	金额大写
++ Province	是	array[]	省
++ City	是	array[]	市
++ Location	是	array[]	发票所在地
++ invoice_type	是	array[]	发票名称
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 air_ticket，即“飞机行程单”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ name	是	array[]	姓名
++ starting_station	是	array[]	始发站
++ destination_station	是	array[]	目的站
++ flight	是	array[]	航班号
++ date	是	array[]	日期
++ ticket_number	是	array[]	电子客票号码
++ fare	是	array[]	票价
++ dev_fund	是	array[]	民航发展基金/基建费
++ oil_money	是	array[]	燃油附加费
++ other_tax	是	array[]	其他税费
++ ticket_rates	是	array[]	合计金额
++ start_date	是	array[]	填开日期
++ id_no	是	array[]	身份证号
++ carrier	是	array[]	承运人
++ time	是	array[]	时间
++ issued_by	是	array[]	订票渠道
++ serial_number	是	array[]	印刷序号
++ insurance	是	array[]	保险费
++ fare_basis	是	array[]	客票级别
++ class	是	array[]	座位等级
++ agent_code	是	array[]	销售单位号
++ endorsement	是	array[]	签注
++ allow	是	array[]	免费行李
++ ck	是	array[]	验证码
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 roll_normal_invoice，即“卷票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceType	是	array[]	发票名称
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ MachineNum	是	array[]	机打号码。仅增值税卷票含有此参数
++ MachineCode	是	array[]	机器编号。仅增值税卷票含有此参数
++ InvoiceDate	是	array[]	开票日期
++ PurchaserName	是	array[]	购方名称
++ PurchaserRegisterNum	是	array[]	购方纳税人识别号
++ SellerName	是	array[]	销售方名称
++ SellerRegisterNum	是	array[]	销售方纳税人识别号
++ TotalTax	是	array[]	价税合计
++ AmountInWords	是	array[]	合计金额(大写)
++ AmountInFiguers	是	array[]	合计金额(小写)
++ Payee	是	array[]	收款人
++ CheckCode	是	array[]	校验码。增值税专票无此参数
++ Province	是	array[]	省
++ City	是	array[]	市
+++ word	是	string	识别结果字符串，以上各字段均包含
++ CommodityName	是	array[]	货物名称
++ CommodityNum	是	array[]	数量
++ CommodityPrice	是	array[]	单价
++ CommodityAmount	是	array[]	金额
+++ row	是	uint32	行号，以上各字段均包含
+++ word	是	string	内容，以上各字段均包含

type 的返回结果为 printed_invoice，即“机打发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceType	是	array[]	发票类型
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ InvoiceDate	是	array[]	开票日期
++ AmountInFiguers	是	array[]	合计金额小写
++ AmountInWords	是	array[]	合计金额大写
++ MachineNum	是	array[]	机打号码
++ CheckCode	是	array[]	校验码
++ SellerName	是	array[]	销售方名称
++ SellerRegisterNum	是	array[]	销售方纳税人识别号
++ PurchaserName	是	array[]	购买方名称
++ PurchaserRegisterNum	是	array[]	购买方纳税人识别号
++ TotalTax	是	array[]	合计税额
++ Province	是	array[]	省
++ City	是	array[]	市
++ Time	是	array[]	时间
++ SheetNum	是	array[]	联次
+++ word	是	string	识别结果字符串，以上各字段均包含
++ CommodityName	是	array[]	商品名称
++ CommodityUnit	是	array[]	商品单位
++ CommodityPrice	是	array[]	商品单价
++ CommodityNum	是	array[]	商品数量
++ CommodityAmount	是	array[]	商品金额
+++ row	是	uint32	行号，以上各字段均包含
+++ word	是	string	内容，以上各字段均包含

type 的返回结果为 bus_ticket，即“汽车票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ Date	是	array[]	日期
++ Time	是	array[]	时间
++ ExitStation	是	array[]	出发站
++ Amount	是	array[]	金额
++ IdCard	是	array[]	身份证号
++ ArrivalStation	是	array[]	到达站
++ Name	是	array[]	姓名
++ InvoiceTime	是	array[]	开票日期
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 toll_invoice，即“过路过桥费”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ Entrance	是	array[]	入口
++ Exit	是	array[]	出口
++ OutDate	是	array[]	日期
++ OutTime	是	array[]	时间
++ TotalAmount	是	array[]	金额
++ Province	是	array[]	省
++ City	是	array[]	市
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 ferry_ticket，即“船票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ InvoiceType	是	array[]	发票类型
++ InvoiceCode	是	array[]	发票代码
++ InvoiceNum	是	array[]	发票号码
++ ExitStation	是	array[]	出发地点
++ ArrivalStation	是	array[]	到达地点
++ Amount	是	array[]	总金额
++ Date	是	array[]	开票日期
++ MoneyType	是	array[]	金额类型
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 motor_vehicle_invoice，即“机动车销售发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ date	是	array[]	开票日期
++ fapiao-daima	是	array[]	发票代码/机打代码
++ fapiao-haoma	是	array[]	发票号码/机打号码
++ printed-daima	是	array[]	机打代码
++ printed-haoma	是	array[]	机打号码
++ machine-num	是	array[]	机器编号
++ buyer-name	是	array[]	购买方名称
++ payer-tax-num	是	array[]	购买方身份证号码/组织机构代码
++ car-class	是	array[]	车辆类型
++ car-model	是	array[]	厂牌型号
++ product-location	是	array[]	产地
++ certificate-num	是	array[]	合格证号
++ engine-num	是	array[]	发动机号码
++ vin-num	是	array[]	车架号码
++ price-tax-big	是	array[]	价税合计
++ price-tax-small	是	array[]	价税合计小写
++ saler	是	array[]	销货单位名称
++ saler-phone	是	array[]	销货单位电话
++ saler-tax-num	是	array[]	销货单位纳税人识别号
++ saler-bank-num	是	array[]	销货单位账号
++ saler-address	是	array[]	销货单位地址
++ saler-bank	是	array[]	销货单位开户银行
++ tax-rate	是	array[]	税率
++ tax	是	array[]	税额
++ tax-jiguan	是	array[]	主管税务机关
++ tax-jiguan-daima	是	array[]	主管税务机关代码
++ price	是	array[]	不含税价格
++ limit-mount	是	array[]	限乘人数
++ toonage	是	array[]	吨位
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 used_vehicle_invoice，即“二手车销售发票”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ invoice_code	是	array[]	发票代码
++ invoice_num	是	array[]	发票号码
++ date	是	array[]	开票日期
++ tax_code	是	array[]	税控码
++ buyer	是	array[]	买方
++ buyer_id	是	array[]	买方身份证号
++ buyer_station	是	array[]	买方地址
++ buyer_tel	是	array[]	买方电话
++ saler	是	array[]	卖方
++ saler_id	是	array[]	卖方身份证号
++ saler_station	是	array[]	卖方地址
++ saler_tel	是	array[]	卖方电话
++ car_plate	是	array[]	车牌号
++ car_certificate	是	array[]	登记证号
++ car_class	是	array[]	车辆类型
++ vin_num	是	array[]	车架号
++ model	是	array[]	厂牌型号
++ to_station	是	array[]	转入地车管所名称
++ big_price	是	array[]	车价合计大写
++ small_price	是	array[]	车价合计小写
++ car_market	是	array[]	二手车市场
++ tax_num	是	array[]	纳税人识别号
++ tax_location	是	array[]	纳税人地址
++ tax_tel	是	array[]	纳税人电话
++ sheet_num	是	array[]	联次
+++ word	是	string	识别结果字符串，以上各字段均包含

type 的返回结果为 taxi_online_ticket，即“网约车行程单”时，识别结果的返回字段如下

字段	是否必选	类型	说明
++ service_provider	是	array[]	服务商
++ start_time	是	array[]	行程开始时间
++ destination_time	是	array[]	行程结束时间
++ phone	是	array[]	行程人手机号
++ application_date	是	array[]	申请日期
++ total_fare	是	array[]	总金额
++ item_num	是	array[]	行程信息中包含的行程数量
+++ word	是	string	识别结果字符串，以上各字段均包含
++ items	是	array[]	行程信息
++++ item_id	是	array[]	行程信息的对应序号
++++ pickup_time	是	array[]	上车时间
++++ pickup_date	是	array[]	上车日期
++++ car_type	是	array[]	车型
++++ distance	是	array[]	里程
++++ start_place	是	array[]	起点
++++ destination_place	是	array[]	终点
++++ city	是	array[]	城市
++++ fare	是	array[]	金额

当验真参数开启（即 verify_parameter=true 时），返回匹配发票验真接口所需的6要素信息

字段	是否必选	类型	说明
++ invoice_code	是	array[]	发票代码
++ invoice_num	是	array[]	发票号码
++ invoice_date	是	array[]	开票日期。返回格式为 YYYYMMDD，例：20210101
++ invoice_type	是	array[]	发票种类。**不同类型发票输出如下结果**： 增值税专用发票：special_vat_invoice 增值税电子专票：elec_special_vat_invoice 增值税普通发票：normal_invoice 增值税普通发票（电子）：elec_normal_invoice 增值税普通发票（卷式）：roll_normal_invoice 通行费增值税电子普通发票： toll_elec_normal_invoice 货运运输业增值税专用发票：special_freight_transport_invoice 机动车销售发票：motor_vehicle_invoice 二手车销售发票：used_vehicle_invoice 区块链发票：blockchain_invoice 通用机打电子发票：printed_elec_invoice
++ total_amount	是	array[]	不含税金额
++ check_code	否	array[]	检验码。**如需使用百度的增值税发票验真接口，需提取返回值的后6位后，再传入验真接口**

返回示例


```
{
  "words_result": [
    {
      "type": "vat_invoice",
      "width": 0,
      "probability": 0.9980429411,
      "height": 649,
      "left": 154,
      "top": 177,
      "result": {
        "AmountInWords": [
          {
            "word": "叁佰陆拾圆整"
          }
        ],
        "InvoiceNumConfirm": [
          {
            "word": "07286261"
          }
        ],
        "CommodityEndDate": [],
        "CommodityVehicleType": [],
        "CommodityStartDate": [],
        "CommodityPrice": [
          {
            "row": "1",
            "word": "339.62"
          }
        ],
```

```
"NoteDrawer": [
  {
    "word": "余佳燕"
  }
],
"SellerAddress": [],
"CommodityNum": [
  {
    "row": "1",
    "word": "1"
  }
],
"SellerRegisterNum": [
  {
    "word": "91330106673959654P"
  }
],
"MachineCode": [],
"Remarks": [],
"SellerBank": [
  {
    "word": "招商银行杭州高新支行502905023610702"
  }
],
"CommodityTaxRate": [
  {
    "row": "1",
    "word": "6%"
  }
],
"TotalTax": [
  {
    "word": "20.38"
  }
],
"InvoiceCodeConfirm": [
  {
    "word": "3321192130"
  }
],
"CheckCode": [],
"InvoiceCode": [
  {
    "word": "3321192130"
  }
],
"InvoiceDate": [
  {
    "word": "2019年08月28日"
  }
],
"PurchaserRegisterNum": [
  {
    "word": "91110911717743469K"
  }
],
"InvoiceTypeOrg": [
  {
    "word": "浙江增值税专用发票"
  }
],
"OnlinePay": [],
```

```
            "Password": [
                {
                    "word": "508>3909>1*>01/-46709-6/3+*7+8>/1*19+7-0**>+58290-6>647-
+324865*9*1<*2191/7754/<0>2<838+//5-69--748*<251408<"
                }
            ],
            "Agent": [
                {
                    "word": "否"
                }
            ],
            "AmountInFiguers": [
                {
                    "word": "360.00"
                }
            ],
            "PurchaserBank": [
                {
                    "word": "招商银行北京分行大电路支行866180100210002"
                }
            ],
            "Checker": [
                {
                    "word": "柳余"
                }
            ],
            "City": [],
            "TotalAmount": [
                {
                    "word": "339.62"
                }
            ],
            "CommodityAmount": [
                {
                    "row": "1",
                    "word": "339.62"
                }
            ],
            "PurchaserName": [
                {
                    "word": "百度在线网络技术(北京)有限公司"
                }
            ],
            "CommodityType": [],
            "Province": [
                {
                    "word": "浙江"
                }
            ],
            "InvoiceType": [
                {
                    "word": "专用发票"
                }
            ],
            "SheetNum": [
                {
                    "word": "第二联：抵扣联"
                }
            ],
            "PurchaserAddress": [],
            "CommodityTax": [
                {
                    "row": "1"
```

```
                "row": "1",
                "word": "20.38"
            }
        ],
        "CommodityPlateNum": [],
        "CommodityUnit": [
            {
                "row": "1",
                "word": "套"
            }
        ],
        "Payee": [
            {
                "word": "佳机"
            }
        ],
        "CommodityName": [
            {
                "row": "1",
                "word": "*信息技术服务*软件服务费"
            }
        ],
        "SellerName": [
            {
                "word": "阿里云计算有限公司"
            }
        ],
        "InvoiceNum": [
            {
                "word": "07286261"
            }
        ]
    }
},
{
    "type": "taxi_receipt",
    "width": 0,
    "probability": 0.9858493805,
    "height": 615,
    "left": 1325,
    "top": 200,
    "result": {
        "PickupTime": [
            {
                "word": "10:50"
            }
        ],
        "DropoffTime": [
            {
                "word": "17:06"
            }
        ],
        "Time": [
            {
                "word": "10:50-17:06"
            }
        ],
        "City": [
            {
                "word": ""
            }
        ],
        "FuelOilSurcharge": [
```

```
            {
                "word": "1.00"
            }
        ],
        "Date": [
            {
                "word": "2019-03-20"
            }
        ],
        "Province": [
            {
                "word": "陕西省"
            }
        ],
        "CallServiceSurcharge": [
            {
                "word": "0.00"
            }
        ],
        "Fare": [
            {
                "word": "21.10"
            }
        ],
        "TotalFare": [
            {
                "word": "22.00"
            }
        ],
        "TaxiNum": [
            {
                "word": "AQ6353"
            }
        ],
        "PricePerkm": [
            {
                "word": "2.30"
            }
        ],
        "InvoiceCode": [
            {
                "word": "161001881016"
            }
        ],
        "Distance": [
            {
                "word": "6.0"
            }
        ],
        "InvoiceNum": [
            {
                "word": "05070716"
            }
        ],
        "Location": [
            {
                "word": "陕西省"
            }
        ]
    }
  }
],
```

```
    "words_result_num": 2,
    "log_id": 1438382953545048984
}
```

增值税发票验真

支持 12 种增值税发票的信息核验，包括增值税专票、电子专票、普票、电子普票、卷票、区块链发票（深圳地区）、全电发票（增值税专用发票）、全电发票（普通发票）、通行费增值税电子普通发票、货物运输业增值税专用发票、机动车销售发票、二手车销售发票等，支持返回票面的全部信息。同时可直接与同平台的发票识别能力对接，完成发票识别的同时进行自动化验真。

```
    public void vatInvoiceVerification() {
// 调用增值税发票验真
String invoice_code = "011002000000";
String invoice_num = "93705563";
String invoice_date = "20210000";
String invoice_type = "elec_normal_invoice";
String check_code = "000000";
String total_amount = "00.00";
      JSONObject result = client.vatInvoiceVerification(invoice_code, invoice_num, invoice_date,
            invoice_type, check_code, total_amount);
      System.out.println("result:"+result);
  }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
invoice_code	是	string	-	发票代码。全电发票（专用发票）、全电发票（普通发票）此参数可为空，其他类型发票均不可为空
invoice_num	是	string	-	发票号码
invoice_date	是	string	-	开票日期。格式YYYYMMDD，例：20210101
invoice_type	是	string	增值税专用发票：special_vat_invoice 增值税电子专用发票：elec_special_vat_invoice 增值税普通发票：normal_invoice 增值税普通发票（电子）：elec_normal_invoice 增值税普通发票（卷式）：roll_normal_invoice 通行费增值税电子普通发票：toll_elec_normal_invoice 区块链电子发票（目前仅支持深圳地区）：blockchain_invoice 全电发票（专用发票）：elec_invoice_special 全电发票（普通发票）：elec_invoice_normal 货运运输业增值税专用发票：special_freight_transport_invoice 机动车销售发票：motor_vehicle_invoice 二手车销售发票：used_vehicle_invoice	发票种类
check_code	是	string	-	校验码。填写发票校验码后6位，增值税电子专票、普票、电子普票、卷票、区块链电子发票、通行费增值税电子普通发票此参数不可为空，其他类型发票可为空
total_amount	是	string	-	发票金额。增值税专票、电子专票、货运专票、机动车销售发票填写**不含税金额**； 二手车销售发票填写**车价合计**； 全电发票（专用发票）、全电发票（普通发票）填写**价税合计金额**，其他类型发票可为空

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
VerifyResult	是	string	查验结果。查验成功返回“0001”，查验失败返回对应查验结果错误码，详见末尾表格
VerifyMessage	是	string	查验结果信息。查验成功且发票为真返回“查验成功发票一致“，查验失败返回对应错误原因，详见末尾表格
VerifyFrequency	是	string	查验次数。为历史查验次数
InvalidSign	是	string	是否作废（冲红）。Y：已作废；H：已冲红；N：未作废
InvoiceType	是	string	发票种类
InvoiceCode	是	string	发票代码
InvoiceNum	是	string	发票号码
CheckCode	是	string	校验码
InvoiceDate	是	string	开票日期
MachineCode	是	string	机器编号

增值税专票、电子专票、普票、电子普通发票、卷票、通行费增值税电子普通发票、货物运输业增值税专用发票返回信息

字段	是否必选	类型	说明
+ PurchaserName	是	string	购方名称
+ PurchaserRegisterNum	是	string	购方纳税人识别号
+ PurchaserAddress	是	string	购方地址及电话
+ PurchaserBank	是	string	购方开户行及账号
+ CommodityName	是	array[]	货物名称/项目名称
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityType	是	array[]	规格型号
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityUnit	是	array[]	单位

++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityNum	是	array[]	数量
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityPrice	是	array[]	单价
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityAmount	是	array[]	金额
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTaxRate	是	array[]	税率
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityTax	是	array[]	税额
++ row	是	uint32	行号
++ word	是	string	内容
+ SellerName	是	string	销售方名称
+ SellerRegisterNum	是	string	销售方纳税人识别号
+ SellerAddress	是	string	销售方地址及电话
+ SellerBank	是	string	销售方开户行及账号
+ TotalAmount	是	string	合计金额
+ TotalTax	是	string	合计税额
+ AmountInFiguers	是	string	价税合计（小写）
+ TollSign	是	string	通行费标志。Y-可抵扣通行费，N-不可抵扣通行费。通行费增值税电子普通发票返回信息，其他类型发票可忽略
+ ZeroTaxRateIndicator	是	string	零税率标识。空：非零税率,1：税率栏位显示“免税”，2：税率栏位显示“不征税”，3：零税率。通行费增值税电子普通发票返回信息，其他类型发票可忽略
+ CommodityPlateNum	是	array[]	车牌号。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号

++ word	是	string	内容
+ CommodityVehicleType	是	array[]	类型。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityStartDate	是	array[]	通行日期起。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ CommodityEndDate	是	array[]	通行日期止。通行费增值税电子普通发票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ Carrier	是	string	承运人名称。货运专票返回信息，其他类型发票可忽略
+ CarrierCode	是	string	承运人识别号。货运专票返回信息，其他类型发票可忽略
+ Recipient	是	string	受票方名称。货运专票返回信息，其他类型发票可忽略
+ RecipientCode	是	string	受票方识别号。货运专票返回信息，其他类型发票可忽略
+ Receiver	是	string	收货人名称。货运专票返回信息，其他类型发票可忽略
+ ReceiverCode	是	string	收货人识别号。货运专票返回信息，其他类型发票可忽略
+ Sender	是	string	发货人名称。货运专票返回信息，其他类型发票可忽略
+ SenderCode	是	string	发货人识别号。货运专票返回信息，其他类型发票可忽略
+ TransportCargoInformation	是	string	运输货物信息。货运专票返回信息，其他类型发票可忽略
+ DepartureViaArrival	是	string	起运地、经由、到达地。货运专票返回信息，其他类型发票可忽略
+ TaxControlNum	是	string	税控盘号。货运专票返回信息，其他类型发票可忽略
+ VehicleType	是	string	车种车号。货运专票返回信息，其他类型发票可忽略
+ VehicleTonnage	是	string	车船吨位。货运专票返回信息，其他类型发票可忽略
+ CommodityExpenseItem	是	array[]	费用项目。货运专票返回信息，其他类型发票可忽略
++ row	是	uint32	行号
++ word	是	string	内容
+ NoteDrawer	是	string	开票人
+ Checker	是	string	复核
+ Payee	是	string	收款人

+ Remarks	是	string	备注
+ ESVATURL	是	string	增值税电子专票（即 ofd 发票）的下载地址
+ ListLabel	是	string	清单标识，Y：带清单；N：无清单； 说明：只有当发票种类为：增值税专票，电子专票，普票，电子普通发票时返回此字段的值

机动车销售发票返回信息

字段	是否必选	类型	说明
+ Purchaser	是	string	购买方名称
+ PurchaserCode	是	string	购买方身份证号/组织机构代码
+ VehicleType	是	string	车辆类型
+ ManuModel	是	string	厂牌型号
+ Origin	是	string	产地
+ CertificateNum	是	string	合格证号书
+ CommodityInspectionNum	是	string	商检单号
+ EngineNum	是	string	发动机号码
+ VinNum	是	string	车辆识别代号/车架号码
+ ImportCertificateNum	是	string	进口证明书号
+ TaxPaymentVoucherNum	是	string	完税凭证号码
+ LimitPassenger	是	string	限乘人数
+ TaxAuthor	是	string	主管税务机关名称
+ TaxAuthorCode	是	string	主管税务机关代码
+ Tonnage	是	string	吨位
+ Price	是	string	不含税价格
+ TaxRate	是	string	税率
+ Tax	是	string	税额
+ PriceTaxLow	是	string	价税合计
+ Saler	是	string	销货单位名称
+ SalerCode	是	string	销货单位纳税人识别号
+ SalerBank	是	string	销货单位开户银行
+ SalerAccountNum	是	string	销货单位账号
+ SalerPhone	是	string	销货单位电话

二手车销售发票返回信息

字段	是否必选	类型	说明
+ Purchaser	是	string	买方单位/个人
+ PurchaserCode	是	string	买方单位代码/身份证号
+ PurchaserAddress	是	string	买方单位/个人住址
+ PurchaserPhone	是	string	买方电话
+ Saler	是	string	卖方单位/个人
+ SalerCode	是	string	卖方单位代码/身份证号
+ SalerAddress	是	string	卖方单位/个人住址
+ SalerPhone	是	string	卖方电话
+ LicensePlateNum	是	string	车牌照号
+ RegistrationCode	是	string	登记证号
+ TotalCarPrice	是	string	车价合计
+ TransferVehicleManagementOffice	是	string	转入地车辆车管所名称
+ VehicleType	是	string	车辆类型
+ ManuModel	是	string	厂牌型号
+ VinNum	是	string	车辆识别代号/车架号码
+ Operator	是	string	经营、拍卖单位
+ OperatorAddress	是	string	经营、拍卖单位地址
+ OperatorCode	是	string	经营、拍卖单位纳税人识别号
+ OperatorBank	是	string	开户银行及账号
+ OperatorPhone	是	string	经营、拍卖单位电话
+ UsedCarMarket	是	string	二手车市场
+ UsedCarMarketCode	是	string	二手车市场纳税人识别号
+ UsedCarMarketAddress	是	string	二手车市地址
+ UsedCarMarketBank	是	string	二手车市场开户银行及账号
+ UsedCarMarketPhone	是	string	二手车市场电话

查验结果错误码

查验结果（VerifyResult）	查验结果信息（VerifyMessage）	描述
9999	查验失败	查验失败，业务出现异常，请提交工单咨询
0002	超过该张票当天查验次数	此发票今日查询次数已达上限（5次），请次日查询
0005	请求不合法	发票信息有误，请核对后再查询
0006	发票信息不一致	发票信息有误，请核对后再查询
0009	发票不存在	所查发票不存在
1004	已超过最大查验量	已超过最大查验量，请提交工单咨询
1005	查询发票不规范	信息有误，请核对后再查询
1006	查验异常	发票信息有误，请核对后再查询
1007	该批次已过期，请重新更换批次号查验	该批次已过期，请重新更换批次查验
1008	字段不能为空	发票请求参数不能为空
1009	参数长度不正确	参数长度不符合规范，确认参数，再次查验
1014	日期当天的不能查验	日期当天的不能查验，请隔天再查
1015	超过5年的不能查验	超过5年的不能查验
1020	没有查验权限	没有查验权限，请提交工单咨询
1021	网络超时	税局维护升级，暂时无法查验，请提交工单咨询

返回示例


```
// 增值税专票、电子专票、普票、电子普通发票、卷票、通行费增值税电子普通发票、货物运输业增值税专用发票
{
    "words_result": {
        "log_id": 1394226734160674816,
        "words_result_num": 43,
        "VerifyFrequency": "3",
        "VerifyMessage": "查验成功发票一致",
        "InvalidSign": "N",
        "InvoiceType": "增值税普通发票（电子）",
        "MachineCode": "661616300747",
        "CheckCode": "67820461013285253079",
        "InvoiceCode": "043002000111",
        "InvoiceDate": "20210503",
        "VerifyResult": "0001",
        "InvoiceNum": "63509760"
        "TaxControlNum": "",
        "CommodityEndDate": [
            {
                "row": "1",
                "word": ""
            }
        ],
        "VehicleTonnage": "",
        "CommodityVehicleType": [
            {
                "row": "1"
            }
        ],
        "CommodityStartDate": [
            {
                "row": "1",
                "word": ""
            }
        ],
        "SellerAddress": "湖南省长沙市天心区芙蓉中路三段446号0731-83592079",
```

```
        "CommodityPrice": [
          {
            "row": "1",
            "word": "28.20000000"
          }
        ],
        "TransportCargoInformation": "",
        "NoteDrawer": "",
        "CommodityNum": [
          {
            "row": "1",
            "word": "1.00000000"
          }
        ],
        "SellerRegisterNum": "914301007121984812",
        "SellerBank": "建行长沙铁银支行营业部43001710661050003739",
        "Remarks": "账期:202104",
        "TotalTax": "0.00",
        "CommodityTaxRate": [
          {
            "row": "1",
            "word": "不征税"
          }
        ],
        "CommodityExpenseItem": [
          {
            "row": "1",
            "word": ""
          }
        ],
        "ZeroTaxRateIndicator": "",
        "Carrier": "",
        "SenderCode": "",
        "PurchaserRegisterNum": "911101087877515792",
        "ReceiverCode": "",
        "AmountInFiguers": "28.20",
        "PurchaserBank": "招商银行北京分行大屯路支行 866182028510003",
        "Checker": "",
        "TollSign": "",
        "VehicleTypeNum": "",
        "DepartureViaArrival": "",
        "Receiver": "",
        "Recipient": "",
        "TotalAmount": "28.20",
        "CommodityAmount": [
          {
            "row": "1",
            "word": "28.20"
          }
        ],
        "PurchaserName": "百度时代网络技术（北京）有限公司",
        "CommodityType": [
          {
            "row": "1",
            "word": ""
          }
        ],
        "Sender": "",
        "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二层A201059108001",
        "CommodityTax": [
          {
            "row": "1",
```

```
            "word": "***"
          }
        ],
        "CarrierCode": "",
        "CommodityPlateNum": [
          {
            "row": "1",
            "word": ""
          }
        ],
        "CommodityUnit": [
          {
            "row": "1",
            "word": ""
          }
        ],
        "Payee": "",
        "RecipientCode": "",
        "CommodityName": [
          {
            "row": "1",
            "word": "*电信服务*通讯费服务费"
          }
        ],
        "SellerName": "中国移动通信集团湖南有限公司长沙分公司"
    },
}
// 机动车销售发票
{
    "words_result": {
        "log_id": 1394232842988290048,
        "words_result_num": 24,
        "VerifyFrequency": "1",
        "VerifyMessage": "查验成功发票一致",
        "InvalidSign": "N",
        "InvoiceType": "机动车销售统一发票",
        "MachineCode": "539927983",
        "CheckCode": "",
        "InvoiceCode": "13200378019836",
        "InvoiceDate": "20210128",
        "VerifyResult": "0001",
        "InvoiceNum": "00342061"
        "Origin": "中国",
        "ManuModel": "东风日产牌DFL8",
        "SalerBank": "工行支行",
        "VehicleType": "多用途乘用车",
        "Tax": "18238.29",
        "TaxPaymentVoucherNum": "",
        "CommodityInspectionNum": "",
        "TaxAuthorCode": "1332803841100",
        "VinNum": "LGBM464574",
        "SalerPhone": "0513-8237861",
        "LimitPassenger": "5",
        "PurchaserCode": "211402199410176136",
        "TaxAuthor": "国家税务总局海门市税务局三厂税务分局",
        "Tonnage": "",
        "ImportCertificateNum": "",
        "Saler": "海门市海通汽车销售服务有限公司",
        "SalerAccountNum": "1111527109002888833",
        "Price": "145840.71",
        "CertificateNum": "WAC224003769810",
        "TaxRate": "13%",
```

```
        "Purchaser": "郑如意",
        "SalerCode": "9132068478280000007164",
        "EngineNum": "43380M",
        "PriceTaxLow": "1323800"
    },
}
// 二手车销售发票
{
    "words_result": {
        "log_id": 1394233936539811840,
        "words_result_num": 25,
        "VerifyFrequency": "1",
        "VerifyMessage": "查验成功发票一致",
        "InvalidSign": "N",
        "InvoiceType": "二手车销售统一发票",
        "MachineCode": "66173004789204",
        "CheckCode": "",
        "InvoiceCode": "0323789200007",
        "InvoiceDate": "20200509",
        "VerifyResult": "0001",
        "InvoiceNum": "002890341"
        "Operator": "",
        "TransferVehicleManagementOffice": "苏州市车管所",
        "ManuModel": "JF1SH95F",
        "RegistrationCode": "3200478903518",
        "OperatorPhone": "",
        "PurchaserCode": "320503782902308u425",
        "Saler": "张散文",
        "UsedCarMarketCode": "91320378038NCQUQXA",
        "Purchaser": "张丽",
        "OperatorCode": "",
        "UsedCarMarketBank": "中国农业银行股份有限公司苏州分行清算中心10549001040001493",
        "SalerAddress": "江苏省苏州市工业园区倪浜路3号",
        "SalerCode": "411524199001016511",
        "PurchaserPhone": "0",
        "LicensePlateNum": "苏U1A666",
        "VehicleType": "小型越野客车",
        "OperatorBank": "",
        "OperatorAddress": "",
        "VinNum": "JF1SH78006596636",
        "TotalCarPrice": "66000.00",
        "SalerPhone": "",
        "PurchaserAddress": "江苏省苏州市相城区元和莫阳村",
        "UsedCarMarketPhone": "13182680222",
        "UsedCarMarketAddress": "苏州高新区长江路668号(3号厂房)",
        "UsedCarMarket": "苏州车市界二手车电子商务有限公司"
    },
}
```


医疗发票识别

支持识别全国各地门诊/住院发票的 业务流水号、发票号、住院号、门诊号、病例号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付 等关键字段，其中北京/广东/河北/河南/江苏/山东/上海/天津/浙江等地区票据识别效果较佳。支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数。

```
    public void medicalInvoice() {
byte[] image = readFile("文件或图片路径");
String url = "https://www.x.com/sample.jpg";

        // 调用医疗发票识别
        JSONObject fileResult = client.medicalInvoice(image, null);
        JSONObject urlResult = client.medicalInvoiceUrl(url, null);
        System.out.println("fileResult:"+fileResult);
        System.out.println("urlResult:"+urlResult);

// 如果有可选参数
HashMap<String, Object> option = new HashMap<>();
        option.put("probability", "true");
JSONObject fileResultOption = client.medicalInvoice(image, option);
        JSONObject urlResultOption = client.medicalInvoiceUrl(url, option);
        System.out.println("fileResultOption:"+fileResultOption);
        System.out.println("urlResultOption:"+urlResultOption);
    }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	string	true/false	是否返回字段的位置信息，**默认为 false，可缺省** - **false**：不返回字段位置信息 - **true**：返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	string	true/false	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
InvoiceType	是	string	票据种类
Province	是	string	省市：支持返回以下省市 北京/广东/河北/河南/江苏/山东/上海/天津/浙江等
words_result	是	object	识别结果
+ BusinessNum	是	object	业务流水号
+ InvoiceNum	是	object	发票号码
+ HospitalNum	是	object	住院号
+ HospitalName	是	object	医院名称

+ RecordNum	是	object	病例号
+ HospitalDay	是	object	住院天数
+ AdmissionDate	是	object	入院时间
+ DischargeDate	是	object	出院时间
+ Name	是	object	姓名
+ Sex	是	object	性别
+ HospitalType	是	object	医疗机构类型
+ SocialSecurityNum	是	object	社保卡号
+ InsuranceType	是	object	医保类型
+ ChargingUnit	是	object	收款单位
+ Payee	是	object	收款人
+ Date	是	object	开票日期
+ AmountInWords	是	object	大写合计金额
+ AmountInFiguers	是	object	小写合计金额
+ InsurancePayment	是	object	医保统筹支付
+ PersonalPayment	是	object	个人账户支付
+ PrepayAmount	是	object	预缴金额
+ PaymentAmount	是	object	补缴金额
+ RefundAmount	是	object	退费金额
+ ClinicNum	是	object	门诊号
++ word	是	string	字段识别结果，以上各字段均包含此参数
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+ CostCategories	是	array[]	项目大类：治疗费、检查费等项目大类
+ CostDetail	是	array[]	明细类别：药物/检查的明细类别
+ RegionSupplement	是	array[]	地区字段：根据省市返回改地区特有的字段
CostCategories字段包含多个array，每个数组包含多个object，见以下参数			

字段	说明
++ name	字段名，包括：收费项目、金额
++ word	name字段对应的识别结果
CostDetail字段包含多个array，每个数组包含多个object，见以下参数	

字段	说明
++ name	字段名，包括：编码、项目、规格、数量、单价、金额
++ word	name字段对应的识别结果
RegionSupplement字段包含多个object，不同省市返回字段不同，见以下参数	

省市	返回参数（name）
北京	个人支付金额、其他医保支付、交易流水号、基金支付、单位补充险[原公疗]支付、年度门诊大额累计支付、本次医保范围内金额、本次支付后个人账户余额、残军补助支付、累计医保内范围金额、自付一、自付二、自费、起付金额、超封顶金额、退休补充支付、门诊大额支付
广东	个人支付金额、其他医保支付
河北	个人账户余额、统筹累计支付、自负、自费、起付标准
河南	个人支付金额、其他医保支付
江苏	个人支付金额、其他医保支付
山东	个人支付金额、其他医保支付
上海	分类自负、历年余额、本年余额、现金支付、自负、自费、附加支付
天津	个人支付金额、其他医保支付
浙江	历年余额、历年支付、基金支付、本年余额、本年支付、现金支付

返回示例


```
{
  "log_id": 1397076899313745920,
  "words_result_num": 28,
  "Province": "北京",
  "InvoiceType": "门诊发票"
  "words_result": {
     "AmountInWords": {
        "word": "玖佰叁拾玖元肆角捌分"
     },
     "Sex": {
        "word": "男"
     },
     "InsuranceType": {
        "word": "城镇职工"
     },
     "Name": {
        "word": "王成"
     },
     "SocialSecurityNum": {
        "word": "T03419700077"
     },
     "DischargeDate": {
        "word": "2016-01-01"
     },
     "HospitalNum": {
        "word": "0937829032"
```

```
    },
    "HospitalName": {
      "word": "北京市房山区良乡医院"
    },
    "CostCategories": [
      [
        {
          "name": "收费项目",
          "word": "西药费"
        },
        {
          "name": "金额",
          "word": "426.70"
        }
      ],
      [
        {
          "name": "收费项目",
          "word": "中成药费"
        },
        {
          "name": "金额",
          "word": "512.78"
        }
      ]
    ],
    "RegionSupplement": [
      {
        "name": "其他医保支付",
        "word": "0.00"
      },
      {
        "name": "年度门诊大额累计支付",
        "word": "83.79"
      },
      {
        "name": "起付金额",
        "word": "777.11"
      },
      {
        "name": "基金支付",
        "word": "101.75"
      },
      {
        "name": "本次支付后个人账户余额",
        "word": "0.00"
      },
      {
        "name": "个人支付金额",
        "word": "837.73"
      },
      {
        "name": "交易流水号",
        "word": "111100030Z160517006328"
      },
      {
        "name": "自付一",
        "word": "795.06"
      },
      {
        "name": "自付二",
        "word": "42.67"
      },
```

```
        {
            "name": "累计医保内范围金额",
            "word": "1419.70"
        },
        {
            "name": "门诊大额支付",
            "word": "83.79"
        },
        {
            "name": "本次医保范围内金额",
            "word": "896.81"
        },
        {
            "name": "退休补充支付",
            "word": "17.96"
        },
        {
            "name": "超封顶金额",
            "word": "0.00"
        },
        {
            "name": "残军补助支付",
            "word": "0.00"
        },
        {
            "name": "单位补充险[原公疗]支付",
            "word": "0.00"
        },
        {
            "name": "自费",
            "word": "42.67"
        }
    ],
    "ClinicNum": {
        "word": "12169298"
    },
    "AmountInFiguers": {
        "word": "939.48"
    },
    "AdmissionDate": {
        "word": "2016-01-01"
    },
    "HospitalType": {
        "word": "综合医院"
    },
    "RefundAmount": {
        "word": "0.00"
    },
    "Date": {
        "word": "2016年05月17日"
    },
    "ChargingUnit": {
        "word": "房山区良乡医院"
    },
    "CostDetail": [
        [
            {
                "name": "编码",
                "word": "01"
            },
            {
                "name": "项目",
```

```
            "word": "阿托伐他汀钙胶囊"
        },
        {
            "name": "规格",
            "word": "10mg*7支"
        },
        {
            "name": "数量",
            "word": "10"
        },
        {
            "name": "单价",
            "word": "29.3200"
        },
        {
            "name": "金额",
            "word": "293.20"
        }
    ],
    [
        {
            "name": "编码",
            "word": "02"
        },
        {
            "name": "项目",
            "word": "替米沙坦胶囊"
        },
        {
            "name": "规格",
            "word": "40mg*12粒"
        },
        {
            "name": "数量",
            "word": "5"
        },
        {
            "name": "单价",
            "word": "26.7000"
        },
        {
            "name": "金额",
            "word": "133.50"
        }
    ],
],
"PaymentAmount": {
    "word": "0.00"
},
"PrepayAmount": {
    "word": "0.00"
},
"PersonalPayment": {
    "word": "0.00"
},
"HospitalDay": {
    "word": "15"
},
"BusinessNum": {
    "word": "40091198916051710196"
},
"InsurancePayment": {
```

```
            "word": "0.00"
        },
        "Payee": {
            "word": "刘冰"
        },
        "RecordNum": {
            "word": "0001268129"
        },
        "InvoiceNum": {
            "word": "0103152099"
        }
    },
}
```

公式识别

支持对试卷中的数学公式及题目内容进行识别，可提取公式部分进行单独识别，也可对题目和公式进行混合识别，并返回Latex格式公式内容及位置信息，便于进行后续处理。

```
    public void formula() {
byte[] image = readFile("文件或图片路径");
String url = "https://www.x.com/sample.jpg";

    // 调用公式识别
    JSONObject fileResult = client.formula(image, null);
    JSONObject urlResult = client.formulaUrl(url, null);
    System.out.println("fileResult:"+fileResult);
    System.out.println("urlResult:"+urlResult);

// 如果有可选参数
    HashMap<String, Object> option = new HashMap<>();
    option.put("recognize_granularity", "small");
    JSONObject fileResultOption = client.formula(image, option);
    JSONObject urlResultOption = client.formulaUrl(url, option);
    System.out.println("fileResultOption:"+fileResultOption);
    System.out.println("urlResultOption:"+urlResultOption);
  }
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置；small：定位单字符位置。默认值为big
detect_direction	否	bool	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
disp_formula	否	bool	true/false	是否分离输出公式识别结果，在words_result外单独输出公式结果，展示在“formula_result”中
返回参数详情				

字段	是否必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - - 1：未定义， - 0：正向， - 1： 逆时针90度， - 2：逆时针180度， - 3：逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
formula_result	否	bool	是否返回单独公式识别结果，默认false
formula_result_num	否	bool	识别结果中的公式个数，表示formula_result的元素个数
words_result	是	array[]	识别结果数组
+ location	是	object{}	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串

返回示例

```
{
  "log_id": 2671713289176456793,
  "direction": 0,
 "formula_result_num": 3,
  "formula_result": [
```

```
        {
            "location": {
                "width": 258,
                "top": 265,
                "left": 450,
                "height": 204
            },
            "words": "\\left\\{ \\begin{aligned} & x = - 1 1 \\\\ & y = 2 \\\\ \\end{aligned} \\right. "
        },
        {
            "location": {
                "width": 429,
                "top": 546,
                "left": 310,
                "height": 203
            },
            "words": "\\left\\{ \\begin{aligned} & 3 x + 2 y = m \\\\ & n x - y = 2 \\\\ \\end{aligned} \\right. "
        },
        {
            "location": {
                "width": 142,
                "top": 613,
                "left": 1029,
                "height": 71
            },
            "words": "m - \\left[ 1 0 0 , - \\infty \\right) "
        }
    ],
    "words_result_num": 5,
    "words_result": [
        {
            "location": {
                "width": 168,
                "top": 313,
                "left": 292,
                "height": 110
            },
            "words": "已知"
        },
        {
            "location": {
                "width": 258,
                "top": 265,
                "left": 450,
                "height": 204
            },
            "words": "\\left\\{ \\begin{aligned} & x = - 1 1 \\\\ & y = 2 \\\\ \\end{aligned} \\right. "
        },
        {
            "location": {
                "width": 582,
                "top": 319,
                "left": 728,
                "height": 84
            },
            "words": "是二元一次方程组"
        },
        {
            "location": {
                "width": 429,
                "top": 546,
                "left": 310,
```

```
            "height": 203
        },
        "words": "\\left\\{ \\begin{aligned} & 3 x + 2 y = m \\\\ & n x - y = 2 \\\\ \\end{aligned} \\right."
    },
    {
        "location": {
            "width": 780,
            "top": 597,
            "left": 745,
            "height": 88
        },
        "words": "的解,则 m - \\left[ 1 0 0 , - \\infty \\right) 的值是()"
    }
  ]
}
```


门脸文字识别

针对含有门脸/门头的图片进行专项优化，支持识别门脸/门头上的文字内容。

```
public void facade() {
byte[] image = readFile("文件或图片路径");

// 调用门脸文字识别
JSONObject fileResult = client.facade(image);
System.out.println("fileResult:"+fileResult);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，base64编码去除编码头（data:image/jpeg;base64），要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	否	array[]	定位和识别结果数组
+ words	否	string	识别结果字符串
+ score	否	float	words返回为主门脸名称的置信度评分
+ brief	否	string	门脸副标题等周边描述

返回示例

```
{
  "log_id": 341559937361307312,
  "words_result_num": 3,
  "words_result": [
        {
        "words": "生活超市",
        "brief": "家得利",
        "score": 0.80533
        },
        {
        "words": "火火火火锅",
        "score": 0.0112433
        },
        {
        "words": "天天好便利店",
        "score': 0.188124
        }
  ]
}
```


通信行程卡识别

可识别国内通信大数据行程卡页面截图中的手机号码、更新日期、途径地三个字段内容，并返回途径地的风险情况。

```
public void sample(AipOcr client) {
  // 参数为本地图片路径
  String image = "test.jpg";
  JSONObject res = client.travelCard(image);
  System.out.println(res.toString(2));

  // 参数为本地图片二进制数组
  byte[] file = readImageFile(image);
  res = client.travelCard(file);
  System.out.println(res.toString(2));
}
```

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的日志id，用于问题定位
result	是	object	识别结果
+ 手机号	是	array	手机号
+ 途径地	是	array	到达或途径的城市信息
+ 更新时间	是	array	该通行信息的更新时间，可用于与当前时间比对查验时效性
+ 风险性	是	bool	返回值范围：true/false，若途径城市存在中高风险地区（带*号），则该值返回为true，因新版行程卡取消*号展示，该字段将固定返回false
++ words	是	string	字段识别内容
++ location	是	object	位置数组（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位位置的长方形的宽度
+++ height	是	uint32	表示定位位置的长方形的高度
error_code	是	uint64	错误码，error_code=0 表明识别无误，其他返回结果表示识别出错，详情请查看 https://ai.baidu.com/ai-doc/OCR/dk3h7y5vr
error_msg	是	string	错误信息

返回示例

```
{
  "error_code":0,
  "error_msg":"",
  "log_id":"146756929847144448",
  "result":{
    "手机号":[
      {
        "location":{
          "top":443,
          "left":156,
          "width":219,
          "height":53
        },
        "word":[
          "132****6231"
        ]
      }
    ],
    "途经地":[
      {
        "location":{
          "top":1004,
          "left":76,
          "width":581,
          "height":223
        },
        "word":[
          "上海市",
          "浙江省杭州市",
          "福建省"
        ]
      }
    ],
    "风险性":true,
    "更新时间":[
      {
        "location":{
          "top":515,
          "left":282,
          "width":330,
          "height":52
        },
        "word":[
          "2021.08.04 21:17:31"
        ]
      }
    ]
  }
}
```


健康码识别

可识别国内各省市健康码截图或拍摄照片中的姓名、更新时间、健康状态三个字段内容。

```
  public void sample(AipOcr client) {
    // 参数为本地图片路径
    String image = "test.jpg";
    JSONObject res = client.healthCode(image);
    System.out.println(res.toString(2));

    // 参数为本地图片二进制数组
    byte[] file = readImageFile(image);
    res = client.healthCode(file);
    System.out.println(res.toString(2));
  }
```

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的日志id，用于问题定位
result	是	object	识别结果
+ 姓名	是	array	健康码姓名
+ 更新时间	是	array	健康码更新时间
+ 状态	是	array	
++ words	是	string	字段识别内容
++ location	是	object	位置数组（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位位置的长方形的宽度
+++ height	是	uint32	表示定位位置的长方形的高度
error_code	是	uint64	错误码，error_code=0 表明识别无误，其他返回结果表示识别出错，详情请查看 https://ai.baidu.com/ai-doc/OCR/dk3h7y5vr
error_msg	是	string	错误信息

返回示例

```
{
  "姓名": [
    {
      "word": "**佳",
      "location": {
        "height": 58,
        "top": 164,
        "width": 98,
        "left": 329
      }
    }
  ],
  "状态": [
    {
      "word": "绿码",
      "location": {
        "height": 38,
        "top": 690,
        "width": 69,
        "left": 380
      }
    }
  ],
  "更新时间": [
    {
      "word": "2022-07-03 16:43:09",
      "location": {
        "height": 52,
        "top": 998,
        "width": 486,
        "left": 34
      }
    }
  ],
  "log_id": 1543887780872961511
}
```


核酸证明识别

可识别国内各省市电子核酸证明截图或拍摄照片中的姓名、检测时间、检测结果三个字段内容，暂不支持纸质核酸证明识别。

```
public void sample(AipOcr client) {
  // 参数为本地图片路径
  String image = "test.jpg";
  JSONObject res = client.covidTest(image);
  System.out.println(res.toString(2));

  // 参数为本地图片二进制数组
  byte[] file = readImageFile(image);
  res = client.covidTest(file);
  System.out.println(res.toString(2));
}
```

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的日志id，用于问题定位
result	是	object	识别结果
+ 姓名	是	array	核酸证明姓名
+ 检测时间	是	array	该核酸结果的检测时间，可用于与当前时间比对检测时效性
+ 检测结果	是	array	核酸检测结果，返回结果为“阴性/阳性/其他”，建议返回结果非“阴性”则进行人工校验
++ words	是	string	字段识别内容
++ location	是	object	位置数组（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位位置的长方形的宽度
+++ height	是	uint32	表示定位位置的长方形的高度

返回示例


```
{
  "姓名": [
    {
      "word": "*佳",
      "location": {
        "height": 28,
        "top": 754,
        "width": 45,
        "left": 580
      }
    }
  ],
  "检测结果": [
    {
      "word": "阴性",
      "location": {
        "height": 73,
        "top": 441,
        "width": 178,
        "left": 304
      }
    }
  ],
  "检测时间": [
    {
      "word": "2022-06-30 18:31:30",
      "location": {
        "height": 24,
        "top": 1037,
        "width": 226,
        "left": 404
      }
    }
  ],
  "log_id": 1543890667962890449
}
```


图文转换器（接口版）--提交请求

图文转换器对应的接口版产品，可识别图片/PDF文件中的文本内容，进行智能版式分析，并转换为保留原文档版式的Word、Excel文档，返回文档下载连接，支持含表格、印章、手写等内容的文档。满足文档版式还原、企业档案电子化等信息管理需求。如需直接在线使用轻应用，可到控制台-图文转换器使用。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();

      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.docConvertRequestV1(image, options);
      System.out.println(res.toString(2));
      // 参数为url
      String url = "http://test.jpg";
      JSONObject res = client.docConvertRequestV1Url(url, options);
      System.out.println(res.toString(2));
      // 参数为本地pdf
      String pdf_file = "test.pdf";
      JSONObject res = client.docConvertRequestV1Pdf(pdf_file, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.docConvertRequestV1(file, options);
      System.out.println(res.toString(2));
    }
```

请求参数

参数	是否必选	类型	说明
image	和 url/pdf_file 三选一	string	图像数据，base64编码后进行urlencode，需去掉编码头（）要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **优先级：** image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **优先级：** image > url > pdf_file，当image字段存在时url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不不超过10M **优先级：** image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容；**若不传入，默认识别文件所有页，页码从1开始**

返回参数

字段	类型	说明
success	bool	当前请求状态； true 表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该请求生成的task_id，后续使用该task_id获取识别结果
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": 12345,
  "result":{
     "task_id":"task-xxxxxxx",
  },
  "code":1001,
  "message": "Create task successfully!"
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
  "success":false,
  "log_id": 12345,
  "error_code": 216401,
  "error_msg": "Create task failed!"
}
```

图文转换器（接口版）--获取结果

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();

      String task_id = "xxxxx";

      // 获取结果
      JSONObject res = client.docConvertResultV1(task_id, options);
      System.out.println(res.toString(2));

    }
```

请求参数

参数	是否必选	类型	说明
task_id	是	string	发送提交请求时返回的task_id

返回参数

字段	类型	说明
success	bool	当前请求状态； true表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该文件对应请求的task_id
+ ret_code	int	识别状态，1：任务未开始；2：进行中；3：已完成
+ ret_msg	string	识别状态信息：任务未开始；进行中；已完成
+ percent	int	文档转换进度（百分比）
+ result_data	dict	识别结果字符串，返回word、excel的文件分别的下载地址
+ +word	string	还原后的word文件的下载地址，**文件识别失败时返回""**
+ +excel	string	还原后的Excel文件的下载地址，**若文档中没有表格则返回""**
+ create_time	datetime	任务创建时间
+ start_time	datetime	任务开始时间
+ end_time	datetime	任务结束时间
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": "xxxxxx",
  "result":{
     "task_id":"task-xxxxxxx",
     "ret_code": 3,
     "ret_msg": "已完成",
     "percent": 100,
     "result_data": {
        "word": "word_download_url",
        "excel": "",
     },
     "create_time": "2023-01-17 11:06:12",
     "start_time": "2023-01-17 11:06:13",
     "end_time": "2023-01-17 11:06:15"
  },
  "code":1001,
  "message": "Query task successfully!""
}
```

若查询的task_id不存在，返回result为{}。 请求失败响应体示例如下：

```
{
  "code":1001,
  "log_id":1635891796603052032,
  "message":"Query task successfully!",
  "result":{},
  "success":true
}
```

表格文字识别同步接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别同步接口。

表格文字识别

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别。

表格识别结果

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别结果。

表格识别轮询接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别轮询接口。

错误信息

错误返回格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

错误码

SDK本地检测参数返回的错误码：

error_code	error_msg	描述
SDK100	image size error	图片大小超限
SDK101	image length error	图片边长不符合要求
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时
SDK109	unsupported image format	不支持的图片格式

服务端返回的错误码：

错误码	错误信息	描述
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
	Open api	

17	daily request limit reached	每天流量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
282000	internal error	服务器内部错误，如果您使用的是高精度接口，报这个错误码的原因可能是您上传的图片中文字过多，识别超时导致的，建议您对图片进行切割后再识别，其他情况请再次请求， 如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
282003	missing parameters: {参数名}	请求参数缺失
	batch	

282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282110	urls not exit	URL参数不存在，请核对URL后再次提交
282111	url format illegal	URL格式非法，请检查url格式是否符合相应接口的入参要求
282112	url download timeout	url下载超时，请检查url对应的图床/图片无法下载或链路状况不好，您可以重新尝试以下，如果多次尝试后仍不行，建议更换图片地址
282113	url response invalid	URL返回无效参数
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误

PHP语言

简介

Hi，您好，欢迎使用百度文字识别服务。

本文档主要针对PHP开发者，描述百度文字识别接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 在百度智能云控制台内提交工单，咨询问题类型请选择**人工智能服务**；
- 如有疑问，进入AI社区交流：http://ai.baidu.com/forum/topic/list/164

接口能力

接口名称	接口能力简要描述
通用文字识别	识别图片中的文字信息
通用文字识别（高精度版）	更高精度地识别图片中的文字信息
通用文字识别（含位置信息版）	识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（高精度含位置版）	更高精度地识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（含生僻字版）	识别图片中的文字信息（包含对常见字和生僻字的识别）
网络图片文字识别	识别一些网络上背景复杂，特殊字体的文字

网络图片文字识别（含位置版）	识别网络图片中的文字内容（包含文字区域的坐标信息）
身份证识别	识别身份证正反面的文字信息
银行卡识别	识别银行卡的卡号并返回发卡行和卡片性质信息
驾驶证识别	识别机动车驾驶证所有关键字段
行驶证识别	识别机动车行驶证所有关键字段
车牌识别	识别中国大陆各类机动车车牌信息
营业执照识别	对营业执照进行识别
表格文字识别	自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容
通用票据识别	对各类票据图片（医疗票据，保险保单等）进行文字识别，并返回文字在图片中的位置信息
增值税发票识别	对增值税发票进行文字识别，并结构化返回字段信息，支持增值税专票、普票、电子发票
出租车票识别	针对全国各大城市出租车票的发票号码、发票代码、车号、日期、时间、金额等进行结构化识别
VIN码识别	对车辆车架、挡风玻璃上的VIN码进行识别
火车票识别	支持对大陆火车票的车票号、始发站、目的站、车次、日期、票价、席别、姓名进行结构化识别
飞机行程单识别	支持对飞机行程单的24个字段进行结构化识别
二维码识别	对图片中的二维码、条形码进行检测和识别，返回存储的文字信息
数字识别	识别图片中的数字，适用于手机号提取、快递单号提取、充值号码提取等场景
手写文字识别	支持对图片中的手写中文、手写数字进行检测和识别
护照识别	支持对中国大陆护照个人资料页所有15个字段进行结构化识别
户口本识别	对出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码字段进行识别
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果
通用机打发票	支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别
机动车销售发票	支持对机动车销售发票的26个关键字段进行结构化识别
车辆合格证	支持对车辆合格证的23个关键字段进行结构化识别
通用机打发票	对国家/地方税务局发行的横/竖版通用机打发票进行结构化识别
护照识别	支持对中国大陆护照个人资料页所有11个字段进行结构化识别
医疗费用明细识别	支持识别全国医疗费用明细识别
网约车行程单识别	对国家/地方税务局发行的横/对各大主要服务商的网约车行程单进行结构化识别
磅单识别	结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单
仪器仪表盘读数识别	适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号
自定义模板文字识别	针对固定版式卡证票据提供的 OCR 定制化产品，可由用户自助创建识别模板和分类器，实现对任意版式卡证票据进行自动分类并结构化输出识别结果
医疗费用明细	支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含

识别	项目类型、项目名称、单价、数量、规格、金额
办公文档识别	可对办公类文档的版面进行分析，输出图、表、标题、文本、目录、栏、页眉、页脚、页码和脚注的位置，并输出分版块内容的OCR识别结果
印章识别	检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别
机动车登记证书识别	对机动车登记证书的编号、机动车所有人、登记机关、车辆类型、发证机关章等15个关键字段进行结构化识别
智能财务票据识别	对增值税发票、卷票、火车票、出租车票、机票行程单等13类票据混贴的图片进行切分识别
增值税发票验真	支持9种增值税发票的真伪及字段信息准确性校验，包括增值税专票、电子专票、普票、电子普票、卷票、通行费增值税电子普票、货运专票、机动车销售发票、二手车销售发票，支持返回票面的全部信息
医疗发票识别	支持识别全国各地门诊/住院发票的 业务流水号、发票号、住院号、门诊号、病例号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付等关键字段。支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数
门脸文字识别	识别图片中的门脸文字信息，自动过滤非门脸文字内容，接口返回门脸名称、描述文字和置信度
车辆证照混贴识别	对机动车行驶证主页及副页、驾驶证主页及副页在同一张图片上的场景进行结构化识别
公式识别	对试卷中的数学公式及题目内容进行识别
图文转换器	可识别图片/PDF文档版面布局，提取文字内容，并转换为保留原文档版式的Word、Excel文档，方便二次编辑和复制，可支持含表格、印章、水印、手写等内容的文档

版本更新记录

上线日期	版本号	更新内容
2021.12.11	4.16.1	新增： 网约车行程单识别，磅单识别，医疗明细识别
2021.05.27	4.15.9	新增： 二维码、行程单、机动车销售发票、车辆合格证、试卷分析与识别、手写、护照、户口本、通用机打（均为商用接口）
2021.01.28	4.15.4	新增 增值税发票、出租车票、VIN码、火车票、数字识别
2020.08.06	4.15.1	新增 文档版面分析与识别，仪器仪表盘读数识别， 网络图片文字识别
2018.4.9	2.2.2	新增表格识别同步接口
2018.01.12	2.1.0	新增自定义OCR识别
2017.12.22	2.0.0	SDK代码重构
2017.8.25	1.6.4	OCR 新增营业执照识别
2017.5.11	1.0.0	OCR服务上线

快速入门

安装OCR PHP SDK

OCR PHP SDK目录结构

```
├─── AipOcr.php          //OCR
└─── lib
   ├─── AipHttpClient.php      //内部http请求类
   ├─── AipBCEUtil.php         //内部工具类
   └─── AipBase               //Aip基类
```

支持PHP版本：5.3+

使用PHP SDK开发骤如下：

1.在官方网站下载php SDK压缩包

2.将下载的`aip-php-sdk-version.zip`解压后，复制AipOcr.php以及lib/*到工程文件夹中

3.引入AipOcr.php

新建AipOcr

AipOcr是OCR的PHP SDK客户端，为使用OCR的开发人员提供了一系列的交互方法。

参考如下代码新建一个AipOcr：

```
 require_once 'AipOcr.php';

// 你的 APPID AK SK
const APP_ID = '你的 App ID';
const API_KEY = '你的 Api Key';
const SECRET_KEY = '你的 Secret Key';

$client = new AipOcr(APP_ID, API_KEY, SECRET_KEY);
```

在上面代码中，常量`APP_ID`在百度智能云控制台中创建，常量`API_KEY`与`SECRET_KEY`是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的**应用列表**中查看。

注意：如您以前是百度智能云的老用户，其中`API_KEY`对应百度智能云的“Access Key ID”，`SECRET_KEY`对应百度智能云的“Access Key Secret”。

配置AipOcr

如果用户需要配置AipOcr的网络请求参数(一般不需要配置)，可以在构造AipOcr之后调用接口设置参数，目前只支持以下参数：

接口	说明
setConnectionTimeoutInMillis	建立连接的超时时间（单位：毫秒)
setSocketTimeoutInMillis	通过打开的连接传输数据的超时时间（单位：毫秒）

接口说明

通用文字识别

用户向服务请求识别某张图中的所有文字。

```
 $image = file_get_contents('example.jpg');

// 调用通用文字识别, 图片参数为本地图片
$client->basicGeneral($image);

// 如果有可选参数
$options = array();
$options["language_type"] = "CHN_ENG";
$options["detect_direction"] = "true";
$options["detect_language"] = "true";
$options["probability"] = "true";

// 带参数调用通用文字识别, 图片参数为本地图片
$client->basicGeneral($image, $options);
$url = "https//www.x.com/sample.jpg";

// 调用通用文字识别, 图片参数为远程url图片
$client->basicGeneralUrl($url);

// 如果有可选参数
$options = array();
$options["language_type"] = "CHN_ENG";
$options["detect_direction"] = "true";
$options["detect_language"] = "true";
$options["probability"] = "true";

// 带参数调用通用文字识别, 图片参数为远程url图片
$client->basicGeneralUrl($url, $options);
```

通用文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别 返回示例

```
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青島啤酒"}
   ]
}
```

通用文字识别（高精度版）

用户向服务请求识别某张图中的所有文字，相对于通用文字识别该产品精度更高，但是识别耗时会稍长。

```
$image = file_get_contents('example.jpg');

// 调用通用文字识别（高精度版）
$client->basicAccurate($image);

// 如果有可选参数
$options = array();
$options["detect_direction"] = "true";
$options["probability"] = "true";

// 带参数调用通用文字识别（高精度版）
$client->basicAccurate($image, $options);
```

通用文字识别（高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别（高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别（高精度版） 返回示例

参考通用文字识别返回示例

通用文字识别（含位置信息版）

用户向服务请求识别某张图中的所有文字，并返回文字在图中的位置信息。

```
 $image = file_get_contents('example.jpg');

// 调用通用文字识别（含位置信息版），图片参数为本地图片
$client->general($image);

// 如果有可选参数
$options = array();
$options["recognize_granularity"] = "big";
$options["language_type"] = "CHN_ENG";
$options["detect_direction"] = "true";
$options["detect_language"] = "true";
$options["vertexes_location"] = "true";
$options["probability"] = "true";

// 带参数调用通用文字识别（含位置信息版），图片参数为本地图片
$client->general($image, $options);
$url = "https//www.x.com/sample.jpg";

// 调用通用文字识别（含位置信息版），图片参数为远程url图片
$client->generalUrl($url);

// 如果有可选参数
$options = array();
$options["recognize_granularity"] = "big";
$options["language_type"] = "CHN_ENG";
$options["detect_direction"] = "true";
$options["detect_language"] = "true";
$options["vertexes_location"] = "true";
$options["probability"] = "true";

// 带参数调用通用文字识别（含位置信息版），图片参数为远程url图片
$client->generalUrl($url, $options);
```

通用文字识别（含位置信息版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
recognize_granularity	否	string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
language_type	否	string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
vertexes_location	否	string	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	string	true false		是否返回识别结果中每一行的置信度
paragraph	否	string	true false		是否输出段落信息

通用文字识别（含位置信息版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置信息版） 返回示例

```
{
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
    "location": {
      "left": 35,
      "top": 53,
      "width": 193,
      "height": 109
    },
    "words": "感动",
    "chars": [    //recognize_granularity=small时存在
      {
        "location": {
          "left": 56,
          "top": 65,
          "width": 69,
          "height": 88
        },
        "char": "感"
      },
      {
        "location": {
          "left": 140,
          "top": 65,
          "width": 70,
          "height": 88
        },
        "char": "动"
      }
    ]
  }
  ...
]
}
```

通用文字识别（含位置高精度版）

用户向服务请求识别某张图中的所有文字，并返回文字在图片中的坐标信息，相对于通用文字识别（含位置信息版）该产品精度更高，但是识别耗时会稍长。

```php
$image = file_get_contents('example.jpg');

// 调用通用文字识别（含位置高精度版）
$client->accurate($image);

// 如果有可选参数
$options = array();
$options["recognize_granularity"] = "big";
$options["detect_direction"] = "true";
$options["vertexes_location"] = "true";
$options["probability"] = "true";

// 带参数调用通用文字识别（含位置高精度版）
$client->accurate($image, $options);
```

通用文字识别（含位置高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
recognize_granularity	否	string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vertexes_location	否	string	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别（含位置高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置高精度版） 返回示例

```
 {
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
     "location": {
        "left": 35,
        "top": 53,
        "width": 193,
        "height": 109
     },
     "words": "感动",
     "chars": [    //recognize_granularity=small时存在
        {
           "location": {
              "left": 56,
              "top": 65,
              "width": 69,
              "height": 88
           },
           "char": "感"
        },
        {
           "location": {
              "left": 140,
              "top": 65,
              "width": 70,
              "height": 88
           },
           "char": "动"
        }
     ]
  }
  ...
]
}
```

通用文字识别（含生僻字版）

某些场景中，图片中的中文不光有常用字，还包含了生僻字，这时用户需要对该图进行文字识别，应使用通用文字识别（含生僻字版）。

```
 $image = file_get_contents('example.jpg');

// 调用通用文字识别（含生僻字版），图片参数为本地图片
$client->enhancedGeneral($image);

// 如果有可选参数
$options = array();
$options["language_type"] = "CHN_ENG";
$options["detect_direction"] = "true";
$options["detect_language"] = "true";
$options["probability"] = "true";

// 带参数调用通用文字识别（含生僻字版），图片参数为本地图片
$client->enhancedGeneral($image, $options);
$url = "https//www.x.com/sample.jpg";

// 调用通用文字识别（含生僻字版），图片参数为远程url图片
$client->enhancedGeneralUrl($url);

// 如果有可选参数
$options = array();
$options["language_type"] = "CHN_ENG";
$options["detect_direction"] = "true";
$options["detect_language"] = "true";
$options["probability"] = "true";

// 带参数调用通用文字识别（含生僻字版），图片参数为远程url图片
$client->enhancedGeneralUrl($url, $options);
```

通用文字识别（含生僻字版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别（含生僻字版） 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含生僻字版） 返回示例


```
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
  [
    {"words": " TSINGTAO"},
    {"words": "青島啤酒"}
  ]
}
```


网络图片文字识别

用户向服务请求识别一些网络上背景复杂，特殊字体的文字。

```
$image = file_get_contents('example.jpg');

// 调用网络图片文字识别, 图片参数为本地图片
$client->webImage($image);

// 如果有可选参数
$options = array();
$options["detect_direction"] = "true";
$options["detect_language"] = "true";

// 带参数调用网络图片文字识别, 图片参数为本地图片
$client->webImage($image, $options);
$url = "https//www.x.com/sample.jpg";

// 调用网络图片文字识别, 图片参数为远程url图片
$client->webImageUrl($url);

// 如果有可选参数
$options = array();
$options["detect_direction"] = "true";
$options["detect_language"] = "true";

// 带参数调用网络图片文字识别, 图片参数为远程url图片
$client->webImageUrl($url, $options);
```

网络图片文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

网络图片文字识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

网络图片文字识别 返回示例

```
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
  [
    {"words": " TSINGTAO"},
    {"words": "青岛啤酒"}
  ]
}
```

身份证识别

用户向服务请求识别身份证，身份证识别包括正面和背面。

```
$image = file_get_contents('example.jpg');
$idCardSide = "back";

// 调用身份证识别
$client->idcard($image, $idCardSide);

// 如果有可选参数
$options = array();
$options["detect_direction"] = "true";
$options["detect_risk"] = "false";

// 带参数调用身份证识别
$client->idcard($image, $idCardSide, $options);
```

身份证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
id_card_side	是	string	front - 身份证含照片的一面 back - 身份证带国徽的一面		front：身份证含照片的一面；back：身份证带国徽的一面
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_risk	否	string	true - 开启 false - 不开启		是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)功能，默认不开启，即：false。可选值:true-开启；false-不开启

身份证识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
image_status	是	string	normal-识别正常 reversed_side-未摆正身份证 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 over_exposure-身份证关键字段反光或过曝 unknown-未知状态
risk_type	否	string	输入参数 detect_risk = true 时，则返回该字段识别身份证类型: normal-正常身份证；copy-复印件；temporary-临时身份证；screen-翻拍；unknow-其他未知情况
edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array(object)	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+location	是	array(object)	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	string	识别结果字符串

身份证识别 返回示例


```
{
  "log_id": 2648325511,
  "direction": 0,
  "image_status": "normal",
  "idcard_type": "normal",
  "edit_tool": "Adobe Photoshop CS3 Windows",
  "words_result": {
    "住址": {
      "location": {
        "left": 267,
        "top": 453,
        "width": 459,
        "height": 99
      },
      "words": "南京市江宁区弘景大道3889号"
    },
    "公民身份号码": {
      "location": {
```

```
            "left": 443,
            "top": 681,
            "width": 589,
            "height": 45
        },
        "words": "330881199904173914"
    },
    "出生": {
        "location": {
            "left": 270,
            "top": 355,
            "width": 357,
            "height": 45
        },
        "words": "19990417"
    },
    "姓名": {
        "location": {
            "left": 267,
            "top": 176,
            "width": 152,
            "height": 50
        },
        "words": "伍云龙"
    },
    "性别": {
        "location": {
            "left": 269,
            "top": 262,
            "width": 33,
            "height": 52
        },
        "words": "男"
    },
    "民族": {
        "location": {
            "left": 492,
            "top": 279,
            "width": 30,
            "height": 37
        },
        "words": "汉"
    }
},
"words_result_num": 6
}
```


银行卡识别

识别银行卡并返回卡号和发卡行。

```
 $image = file_get_contents('example.jpg');

// 调用银行卡识别
$client->bankcard($image);
```

银行卡识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。

银行卡识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	请求标识码，随机数，唯一。
result	object	是	返回结果
+bank_card_number	string	是	银行卡卡号
+bank_name	string	是	银行名，不能识别时为空
+bank_card_type	number	是	银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡

银行卡识别 返回示例

```
{
  "log_id": 1447188951,
  "result": {
     "bank_card_number": "622500000000000",
     "bank_name": "招商银行",
     "bank_card_type": 1
  }
}
```

驾驶证识别

对机动车驾驶证所有关键字段进行识别。

```
 $image = file_get_contents('example.jpg');

// 调用驾驶证识别
$client->drivingLicense($image);

// 如果有可选参数
$options = array();
$options["detect_direction"] = "true";

// 带参数调用驾驶证识别
$client->drivingLicense($image, $options);
```

驾驶证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

驾驶证识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
+words	否	string	识别结果字符串

驾驶证识别 返回示例


```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "证号": {
    "words": "3208231999053090"
   },
   "有效期限": {
    "words": "6年"
   },
   "准驾车型": {
    "words": "B2"
   },
   "有效起始日期": {
    "words": "20101125"
   },
   "住址": {
    "words": "江苏省南通市海门镇秀山新城"
   },
   "姓名": {
    "words": "小欧欧"
   },
   "国籍": {
    "words": "中国"
   },
   "出生日期": {
    "words": "19990530"
   },
   "性别": {
    "words": "男"
   },
   "初次领证日期": {
    "words": "20100125"
   }
  }
 }
}
```


行驶证识别

对机动车行驶证正本所有关键字段进行识别。

```
$image = file_get_contents('example.jpg');

// 调用行驶证识别
$client->vehicleLicense($image);

// 如果有可选参数
$options = array();
$options["detect_direction"] = "true";
$options["accuracy"] = "normal";

// 带参数调用行驶证识别
$client->vehicleLicense($image, $options);
```

行驶证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
detect_direction	否	string	true/false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vehicle_license_side	否	string	front/back	front	- **front**：识别行驶证主页 - **back**：识别行驶证副页
unified	否	string	true/false	false	- **false**：不进行归一化处理 - **true**：对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出

行驶证识别 返回数据参数详情

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
+words	否	string	识别结果字符串

行驶证识别 返回示例

```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "品牌型号": {
    "words": "保时捷GT37182RUCRE"
   },
   "发证日期": {
    "words": "20160104"
   },
   "使用性质": {
    "words": "非营运"
   },
   "发动机号码": {
    "words": "20832"
   },
   "号牌号码": {
    "words": "苏A001"
   },
   "所有人": {
    "words": "圆圆"
   },
   "住址": {
    "words": "南京市江宁区弘景大道"
   },
   "注册日期": {
    "words": "20160104"
   },
   "车辆识别代号": {
    "words": "HCE58"
   },
   "车辆类型": {
    "words": "小型轿车"
   }
  }
 }
}
```


车牌识别

识别机动车车牌，并返回号牌号码和车牌颜色。

```
    $image = file_get_contents('example.jpg');

    // 调用车牌识别
    $client->licensePlate($image);

    // 如果有可选参数
    $options = array();
    $options["multi_detect"] = "true";

    // 带参数调用车牌识别
    $client->licensePlate($image, $options);
```

车牌识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
multi_detect	否	string	true false	false	是否检测多张车牌，默认为false，当置为true的时候可以对一张图片内的多张车牌进行识别

车牌识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一
Color	string	是	车牌颜色
number	string	是	车牌号码

车牌识别 返回示例

```
{
  "log_id": 3583925545,
  "words_result": {
     "color": "blue",
     "number": "苏HS7766"
  }
}
```

营业执照识别

识别营业执照，并返回关键字段的值，包括单位名称、法人、地址、有效期、证件编号、社会信用代码等。

```
 $image = file_get_contents('example.jpg');

// 调用营业执照识别
$client->businessLicense($image);
```

营业执照识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。

营业执照识别 返回数据参数详情

参数	是否必须	类型	说明
log_id	是	number	请求标识码，随机数，唯一。
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	否	string	识别结果字符串

营业执照识别 返回示例

```
{
  "log_id": 490058765,
  "words_result": {
    "单位名称": {
      "location": {
        "left": 500,
        "top": 479,
        "width": 618,
        "height": 54
      },
      "words": "袁氏财团有限公司"
    },
    "法人": {
      "location": {
        "left": 938,
        "top": 557,
        "width": 94,
        "height": 46
      },
      "words": "袁运筹"
    },
    "地址": {
      "location": {
        "left": 503,
        "top": 644,
        "width": 574,
        "height": 57
      },
      "words": "江苏省南京市中山东路19号"
    },
    "有效期": {
      "location": {
        "left": 779,
        "top": 1108,
        "width": 271,
        "height": 49
      },
      "words": "2015年02月12日"
    },
    "证件编号": {
      "location": {
        "left": 1219,
        "top": 357,
        "width": 466,
        "height": 39
      },
      "words": "苏餐证字(2019)第666602666661号"
    },
    "社会信用代码": {
      "location": {
        "left": 0,
        "top": 0,
        "width": 0,
        "height": 0
      },
      "words": "无"
    }
  },
  "words_result_num": 6
}
```

通用票据识别

用户向服务请求识别医疗票据、增值税发票、出租车票、保险保单等票据类图片中的所有文字，并返回文字在图中的位置信息。

```
 $image = file_get_contents('example.jpg');

// 调用通用票据识别
$client->receipt($image);

// 如果有可选参数
$options = array();
$options["recognize_granularity"] = "big";
$options["probability"] = "true";
$options["accuracy"] = "normal";
$options["detect_direction"] = "true";

// 带参数调用通用票据识别
$client->receipt($image, $options);
```

通用票据识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
recognize_granularity	否	string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true false		是否返回识别结果中每一行的置信度
accuracy	否	string	normal - 使用快速服务		normal 使用快速服务，1200ms左右时延；缺省或其它值使用高精度服务，1600ms左右时延
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

通用票据识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array()	单字符结果，recognize_granularity=small时存在
location	是	array()	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位定位位置的长方形的宽度
height	是	number	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值

通用票据识别 返回示例

```
{
  "log_id": 2661573626,
  "words_result": [
    {
      "location": {
        "left": 10,
        "top": 3,
        "width": 121,
        "height": 24
      },
      "words": "姓名:小明明",
      "chars": [
        {
          "location": {
            "left": 16,
            "top": 6,
            "width": 17,
            "height": 20
          },
          "char": "姓"
        }
        ...
      ]
    },
    {
      "location": {
        "left": 212,
        "top": 3,
        "width": 738,
        "height": 24
      },
      "words": "卡号/病案号:105353990标本编号:150139071送检科室:血液透析门诊病房",
      "chars": [
        {
          "location": {
            "left": 218,
            "top": 6,
            "width": 18,
            "height": 21
          },
          "char": "卡"
        }
        ...
      ]
    }
  ],
  "words_result_num": 2
}
```


自定义模板文字识别

自定义模板文字识别，是针对百度官方没有推出相应的模板，但是当用户需要对某一类卡证/票据（如房产证、军官证、火车票等）进行结构化的提取内容时，可以使用该产品快速制作模板，进行识别。

```php
$image = file_get_contents('example.jpg');

// 设置可选参数（templateSign和classifierId至少存在一个）
$options = array();
$options["templateSign"] = "Nsdax2424asaAS791823112";

// 调用自定义模板文字识别
$client->custom($image, $options);
```

自定义模板文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
templateSign	否	string	您在自定义文字识别平台制作的模版的ID
classifierId	否	string	分类器Id。这个参数和templateSign至少存在一个，优先使用templateSign。存在templateSign时，表示使用指定模板；如果没有templateSign而有classifierId，表示使用分类器去判断使用哪个模板

自定义模板文字识别 返回数据参数详情

字段	是否必选	类型	说明
error_code	number	number	0代表成功，如果有错误码返回可以参考下方错误码列表排查问题
error_msg	是	string	具体的失败信息，可以参考下方错误码列表排查问题
data	jsonObject	识别返回的结果	

自定义模板文字识别 返回示例


```
{
  "isStructured": true,
  "ret": [
    {
      "charset": [
        {
          "rect": {
            "top": 183,
            "left": 72,
            "width": 14,
            "height": 28
          },
          "word": "5"
        },
        {
          "rect": {
            "top": 183,
            "left": 90,
            "width": 14,
            "height": 28
          },
          "word": "4"
        },
        {
          "rect": {
            "top": 183
```

```
                "top": 183,
                "left": 103,
                "width": 15,
                "height": 28
            },
            "word": "."
        },
        {
            "rect": {
                "top": 183,
                "left": 116,
                "width": 14,
                "height": 28
            },
            "word": "5"
        },
        {
            "rect": {
                "top": 183,
                "left": 133,
                "width": 19,
                "height": 28
            },
            "word": "元"
        }
    ],
    "word_name": "票价",
    "word": "54.5元"
},
{
    "charset": [
        {
            "rect": {
                "top": 144,
                "left": 35,
                "width": 14,
                "height": 28
            },
            "word": "2"
        },
        {
            "rect": {
                "top": 144,
                "left": 53,
                "width": 14,
                "height": 28
            },
            "word": "0"
        },
        {
            "rect": {
                "top": 144,
                "left": 79,
                "width": 14,
                "height": 28
            },
            "word": "1"
        },
        {
            "rect": {
                "top": 144,
                "left": 97,
                "width": 14,
```

```
                "height": 28
            },
            "word": "7"
        }
    ]
  ]
}
```

文档版面分析与识别

可对文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景。

```
$image = file_get_contents('example.jpg');
$client->docAnalysis($image);
```

识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少64px，最长边最大4096px。注意：图片的base64编码是不包含图片头的，如（） **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。	
language_type	false	string	CHN_ENG / ENG	识别语言类型，默认为CHN_ENG 可选值包括：=CHN_ENG：中英文 =ENG：英文	
result_type	false	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果	
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度	
line_probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false	
words_type	false	string	handwring_only/ handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别	
layout_analysis	false	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出	

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
img_direction	false	int32	detect_direction=true时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	true	uint32	识别结果数，表示results的元素个数
results	true	array[]	识别结果数组
+words_type	true	string	文字属性（手写、印刷）,handwriting 手写，print 印刷
+words	true	array[]	整行的识别结果数组。
++line_probability	false	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++average	false	float	行置信度
+++min	false	float	整行中单字的最低置信度
++word	true	float	整行的识别结果
++words_location	true	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	true	uint32	表示定位定位位置的长方形的宽度
+++height	true	uint32	表示位置的长方形的高度
+chars	false	array[]	result_type=small时返回。单字符结果数组。
++char	false	string	result_type=small时返回。每个单字的内容。
++chars_location	false	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示位置的长方形的高度
layouts_num	false	uint32	版面分析结果数，表示layout的元素个数
layouts	false	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+layout	false	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title
+layout_location	false	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++x	false	uint32	水平坐标（坐标0点为左上角）
++y	false	uint32	水平坐标（坐标0点为左上角）
+layout_idx	false	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）

仪器仪表盘读数识别

适用于不同品牌、不同型号的仪器仪表盘读数识别，广泛适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号，支持液晶屏、字轮表等表型。

```
$image = file_get_contents('example.jpg');
$client->meter($image);
```

识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px。支持jpg/jpeg/png/bmp格式，图片的base64编码是不包含图片头的，如（）。 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。	
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false	
poly_location	false	string	true/false	位置信息返回形式，默认：false false：只给出识别结果所在长方形位置信息 true：除了默认的识别文字所在长方形的位置信息，还会给出文字所在区域的最小外接旋转矩形的4个点坐标信息	

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	识别结果字符串
+location	true	array[]	识别结果所在长方形位置信息
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	false	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	false	array[]	poly_location=true时存在。文字所在区域的外接四边形的4个点坐标信息

网络图片文字识别（含位置版）

支持识别艺术字体或背景复杂的文字内容，除文字信息外，还可返回每行文字的位置信息、行置信度，以及单字符内容和位置等。

```
$image = file_get_contents('example.jpg');
$client->webimageLoc($image);
```

网络图片文字识别（含位置版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，像素乘积不超过2048*2048（1024*1024以内图像处理效果最佳），图片的base64编码是不包含图片头的，如（）。 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。	
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向	
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false	
poly_location	false	string	true/false	是否返回文字所在区域的外接四边形的4个点坐标信息。默认为false	
recognize_granularity	false	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置	

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
direction	false	int32	图像方向，当detect_direction=true时存在。检测到的图像朝向： 0 ：正向； 1：逆时针旋转90度； 2：逆时针旋转180度； 3：逆时针旋转270度
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	整行的识别结果
+location	true	object	整行的矩形框坐标。位置数组（坐标0点为左上角）
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	true	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	true	array[]	poly_location=true时存在。文字所在区域的外接矩形的4个点坐标信息
++x	true	uint32	水平坐标（坐标0点为左上角）
++y	true	uint32	垂直坐标（坐标0点为左上角）
+chars	false	array[]	单字符结果，recognize_granularity=small时存在
++char	false	string	单字符识别结果
++location	false	object	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示定位定位位置的长方形的高度

增值税发票

```
 $client = new AipOcr(APP_ID, API_KEY, SECRET_KEY);
$image = file_get_contents('example.jpg');
//可选参数
$options=array();
$options['type']='normal';
//本地文件识别
$client->vatInvoice($image,$options);
//url 识别
$client->vatInvoiceUrl('htpp:/test.jpeg',$options);
//本地pdf文件识别
$image = file_get_contents('example.pdf');
$client->vatInvoicePdf($image,$options);
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url或者pdf文件 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
type	否	String	可选参数，进行识别的增值税发票类型，默认为 normal，可缺省normal：可识别增值税普票、专票、电子发票roll：可识别增值税卷票

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	是	string	发票种类
InvoiceTypeOrg	是	string	发票名称
InvoiceCode	是	string	发票代码
InvoiceNum	是	string	发票号码
MachineNum	是	string	机打号码。仅增值税卷票含有此参数
MachineCode	是	string	机器编号。仅增值税卷票含有此参数
CheckCode	否	string	校验码。增值税专票无此参数
InvoiceDate	是	string	开票日期
PurchaserName	是	string	购方名称
PurchaserRegisterNum	是	string	购方纳税人识别号
PurchaserAddress	是	string	购方地址及电话
PurchaserBank	是	string	购方开户行及账号
Password	是	string	密码区
Province	是	string	省
City	是	string	市
SheetNum	是	string	联次
Agent	是	string	是否代开
CommodityName	是	array[]	货物名称
- row	是	uint32	行号
- word	是	string	内容
CommodityType	是	array[]	规格型号
- row	是	uint32	行号
- word	是	string	内容
CommodityUnit	是	array[]	单位
- row	是	uint32	行号
- word	是	string	内容
CommodityNum	是	array[]	数量
- row	是	uint32	行号

- word	是	string	内容
CommodityPrice	是	array[]	单价
- row	是	uint32	行号
- word	是	string	内容
CommodityAmount	是	array[]	金额
- row	是	uint32	行号
- word	是	string	内容
CommodityTaxRate	是	array[]	税率
- row	是	uint32	行号
- word	是	string	内容
CommodityTax	是	array[]	税额
- row	是	uint32	行号
- word	是	string	内容
SellerName	是	string	销售方名称
SellerRegisterNum	是	string	销售方纳税人识别号
SellerAddress	是	string	销售方地址及电话
SellerBank	是	string	销售方开户行及账号
TotalAmount	是	uint32	合计金额
TotalTax	是	uint32	合计税额
AmountInWords	是	string	价税合计(大写)
AmountInFiguers	是	uint32	价税合计(小写)
Payee	是	string	收款人
Checker	是	string	复核
NoteDrawer	是	string	开票人
Remarks	是	string	备注

返回示例


```
{
 "log_id": "5425496231209218858",
"words_result_num": 29,
"words_result": {
 "InvoiceNum": "14641426",
 "SellerName": "上海易火广告传媒有限公司",
 "CommodityTaxRate": [
  {
   "word": "6%",
   "row": "1"
  }
 ],
 "SellerBank": "中国银行南翔支行446863841354",
 "Checker": ":沈园园",
 "TotalAmount": "94339.62",
 "CommodityAmount": [
  {
   "word": "94339.62",
   "row": "1"
  }
 ],
```

```
  "InvoiceDate": "2016年06月02日",
  "CommodityTax": [
   {
    "word": "5660.38",
    "row": "1"
   }
  ],
  "PurchaserName": "百度时代网络技术(北京)有限公司",
  "CommodityNum": [
   {
    "word": "",
    "row": "1"
   }
  ],
        "Province": "上海",
        "City": "",
        "SheetNum": "第三联",
        "Agent": "否",
  "PurchaserBank": "招商银行北京分行大屯路支行8661820285100030",
  "Remarks": "告传",
  "Password": "074/45781873408>/6>8>65*887676033/51+<5415>9/32--852>1+29<65>641-
5>66<500>87/*-34<943359034>716905113*4242>",
  "SellerAddress": ":嘉定区胜辛南路500号15幢1161室55033753",
  "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二属A2010-59108001",
  "InvoiceCode": "3100153130",
  "CommodityUnit": [
   {
    "word": "",
    "row": "1"
   }
  ],
  "Payee": ":徐蓉",
  "PurchaserRegisterNum": "110108787751579",
  "CommodityPrice": [
   {
    "word": "",
    "row": "1"
   }
  ],
  "NoteDrawer": "沈园园",
  "AmountInWords": "壹拾万圆整",
  "AmountInFiguers": "100000.00",
  "TotalTax": "5660.38",
  "InvoiceType": "专用发票",
  "SellerRegisterNum": "913101140659591751",
  "CommodityName": [
   {
    "word": "信息服务费",
    "row": "1"
   }
  ],
  "CommodityType": [
   {
    "word": "",
    "row": "1"
   }
  ]
 }
}
```


出租车票识别

```
 $client = new AipOcr(APP_ID, API_KEY, SECRET_KEY);
$image = file_get_contents('example.jpg');
//可选参数
$options=array();
//本地文件识别
$client->taxiReceipt($image,$options);
//url 识别
$client->taxiReceiptUrl('htpp:/test.jpeg',$options);
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
words_result_num	uint32	是	识别结果数，表示words_result的元素个数
InvoiceCode	string	是	发票代号
InvoiceNum	string	是	发票号码
TaxiNum	string	是	车牌号
Date	string	是	日期
Time	string	是	上下车时间
Fare	string	是	总金额
FuelOilSurcharge	string	是	燃油附加费
CallServiceSurcharge	string	是	叫车服务费
Province	string	是	省
City	string	是	市
PricePerkm	string	是	单价
Distance	string	是	里程

返回示例

```
{
"log_id":2034039896,
"words_result_num":6,
"words_result":
{
 "Date":"2017-11-26",
 "Fare":"¥153.30元",
 "InvoiceCode":"111001681009",
 "InvoiceNum":"90769610",
 "TaxiNum":"BV2062",
 "Time":"20:42-21:07",
 "FuelOilSurcharge": "¥0.00",
 "CallServiceSurcharge": "¥0.00",
      "Province": "浙江省",
      "City": "杭州市",
      "PricePerkm": "2.50元/KM",
      "Distance": "4.5KM"
}
}
```


VIN码识别

```
 $client = new AipOcr(APP_ID, API_KEY, SECRET_KEY);
$image = file_get_contents('example.jpg');
//可选参数
$options=array();
//本地文件识别
$client->vinCode($image,$options);
//url 识别
$client->vinCodeUrl('htpp:/test.jpeg',$options);
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
location	是	object{}	识别结果
words	是	string	VIN码识别结果
words_result_num	是	int	识别结果数，表示words_result的元素个数

返回示例

```
{
"log_id": 246589877,
"words_result": [
 {
  "location": {
   "left": 124,
   "top": 11,
   "width": 58,
   "height": 359
  },
  "words": "LFV2A11K8D4010942"
 }
],
"words_result_num": 1
}
```


火车票识别

```
 $client = new AipOcr(APP_ID, API_KEY, SECRET_KEY);
$image = file_get_contents('example.jpg');
//可选参数
$options=array();
//本地文件识别
$client->trainTicket($image,$options);
//url 识别
$client->trainTicketUrl('htpp:/test.jpeg',$options);
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url， **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
ticket_num	string	是	车票号
starting_station	string	是	始发站
train_num	string	是	车次号
destination_station	string	是	到达站
date	string	是	出发日期
ticket_rates	string	是	车票金额
seat_category	string	是	席别
name	string	是	乘客姓名
id_num	string	是	身份证号
serial_number	string	是	序列号
sales_station	string	是	售站
time	string	是	时间
seat_num	string	是	座位号

返回示例


```
{
"log_id": "12317512659",
"direction": 1,
"words_result_num": 13,
"words_result": {
 "id_num": "2302051998****156X",
 "name": "裴一丽",
 "ticket_rates": "￥54.5元",
 "destination_station": "天津站",
 "seat_category": "二等座",
 "sales_station": "北京南",
 "ticket_num": "F05706",
 "seat_num": "02车03C号",
 "time": "09:36",
 "date": "2019年04月03日",
 "serial_number": "10010300067846",
 "train_num": "C255",
 "starting_station": "北京南站"
}
}
```


数字识别

```
 $client = new AipOcr(APP_ID, API_KEY, SECRET_KEY);
$image = file_get_contents('example.jpg');
//可选参数
$options=array();
//本地文件识别
$client->numbers($image,$options);
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据， **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
recognize_granularity	false	string	big、small
detect_direction	false	string	true、false

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
      "location": {
        "left": 56,
        "top": 0,
        "width": 21,
        "height": 210
      },
      "words": "3"
    }
  ],
  "words_result_num": 1
}
```

二维码识别

对图片中的二维码、条形码进行检测和识别，返回存储的文字信息。

```
 $image = file_get_contents('example.jpg');

// 二维码识别
$client->qrcode($image);

// 如果有可选参数
$options = array();

// 带参数调用车型识别
$client->qrcode($image, $options);
 // 参数图片url
$url = "https//www.x.com/sample.jpg"
client.qrcodeUrl($url, $options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
codes_result_num	是	uint32	识别结果数，表示codes_result的元素个数
codes_result	是	array[]	定位和识别结果数组
-type	是	string	识别码类型条码类型包括：9种条形码（UPC_A、UPC_E、EAN_13、EAN_8、CODE_39、CODE_93、CODE_128、ITF、CODABAR），4种二维码（QR_CODE、DATA_MATRIX、AZTEC、PDF_417）
-text	是	string	条形码识别内容,暂时只限于识别中英文结果

返回示例

```
{
  "log_id": 863402790,
  "codes_result": [
     {
        "type": "QR_CODE",
        "text": [
           "中国",
           "北京"
        ]
```

```
        }
    ],
    "codes_result_num": 1
}
```


示例2（多个图的情况）：


```
{
    "log_id": 1508509437,
    "codes_result": [
        {
            "type": "QR_CODE",
            "text": [
                "HTTP://Q8R.HK/YELZ0"
            ]
        },
        {
            "type": "PDF_417",
            "text": [
                "PDF417使上TL-30循擎傒座傝壜撽循堪丁"
            ]
        },
        {
            "type": "CODABAR",
            "text": [
                "000800"
            ]
        },
        {
            "type": "CODE_39",
            "text": [
                "1234567890"
            ]
        },
        {
            "type": "AZTEC",
            "text": [
                "www.tec-it.com"
            ]
        },
        {
            "type": "DATA_MATRIX",
            "text": [
                "Wikipedia, the free encyclopedia"
            ]
        },
        {
            "type": "CODE_93",
            "text": [
                "123456789"
            ]
        },
        {
            "type": "CODE_128",
            "text": [
                "50090500019191"
            ]
        },
        {
            "type": "EAN_8",
            "text": [
                "12345670"
            ]
        },
```

```
        {
            "type": "EAN_13",
            "text": [
                "6901234567892"
            ]
        },
        {
            "type": "UPC_E",
            "text": [
                "01234565"
            ]
        }
    ],
    "codes_result_num": 11
}
```


试卷分析与识别

可对文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景。

```php
 $image = file_get_contents('example.jpg');

// 试卷分析与识别
$client->docAnalysis($image);

// 如果有可选参数
$options = array();

// 带参数调用车型识别
$client->docAnalysis($image, $options);
 // 参数图片url
$url = "https//www.x.com/sample.jpg"
client.docAnalysisUrl($url, $options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少64px，最长边最大4096px，注意：图片的base64编码是不包含图片头的，如（）。 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
language_type	false	string	CHN_ENG/ENG	识别语言类型，默认为CHN_ENG 可选值包括： =CHN_ENG：中英文 =ENG：英文
result_type	false	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
words_type	false	string	handwring_only/handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	false	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
img_direction	false	int32	detect_direction=true时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	true	uint32	识别结果数，表示results的元素个数
results	true	array[]	识别结果数组
+words_type	true	string	文字属性（手写、印刷）,handwriting 手写，print 印刷
+words	true	array[]	整行的识别结果数组。
++line_probability	false	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++average	false	float	行置信度
+++min	false	float	整行中单字的最低置信度
++word	true	float	整行的识别结果
++words_location	true	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	true	uint32	表示定位定位位置的长方形的宽度
+++height	true	uint32	表示位置的长方形的高度
+chars	false	array[]	result_type=small时返回。单字符结果数组。
++char	false	string	result_type=small时返回。每个单字的内容。
++chars_location	false	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示位置的长方形的高度
layouts_num	false	uint32	版面分析结果数，表示layout的元素个数
layouts	false	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+layout	false	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title
+layout_location	false	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++x	false	uint32	水平坐标（坐标0点为左上角）
++y	false	uint32	水平坐标（坐标0点为左上角）
+layout_idx	false	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）

返回示例

请参考通用文字识别（标准含位置版）返回示例

机动车销售发票

支持对机动车销售发票的26个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、机器编号、购买方名称、购买方身份证号码/组织机构代码、车辆类型、厂牌型号、产地、合格证号、发动机号码、车架号码、价税合计、价税合计小写、销货单位名称、电话、纳税人识别号、账号、地址、开户银行、税率、税额、主管税务机关及代码、不含税价格、限乘人数。

```
 $image = file_get_contents('example.jpg');

// 机动车销售发票
$client->vehicleInvoice($image);

// 如果有可选参数
$options = array();

// 带参数调用车型识别
$client->vehicleInvoice($image, $options);
 // 参数图片url
$url = "https//www.x.com/sample.jpg"
client.vehicleInvoiceUrl($url, $options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
InvoiceCode	是	string	发票代码/机打代码
InvoiceNum	是	string	发票号码/机打号码
InvoiceDate	是	string	开票日期
MachineCode	是	string	机器编号
Purchaser	是	string	购买方名称
PurchaserCode	是	string	购买方身份证号码/组织机构代码
VehicleType	是	string	车辆类型
ManuModel	是	string	厂牌型号
Origin	是	string	产地
CertificateNum	是	string	合格证号
EngineNum	是	string	发动机号码
VinNum	是	string	车架号码
PriceTax	是	string	价税合计
PriceTaxLow	是	string	价税合计小写
Saler	是	string	销货单位名称
SalerPhone	是	string	销货单位电话
SalerCode	是	string	销货单位纳税人识别号
SalerAccountNum	是	string	销货单位账号
SalerAddress	是	string	销货单位地址
SalerBank	是	string	销货单位开户银行
TaxRate	是	string	税率
Tax	是	string	税额
TaxAuthor	是	string	主管税务机关
TaxAuthorCode	是	string	主管税务机关代码
Price	是	string	不含税价格
LimitPassenger ｜ 是	string	限乘人数	

返回示例

```
{
  "log_id": 283449393728149457,
  "words_result_num": 26,
  "words_result": {
     "InvoiceNum": "00875336",
     "Saler": "深圳市新能源汽车销售有限公司",
     "LimitPassenger": "5",
     "MachineCode": "669745967911",
     "VinNum": "LJLGTCRP1J4007581",
     "TaxRate": "16%",
     "PriceTaxLow": "106100.00",
     "InvoiceDate": "2018-11-29",
     "Price": "￥91465.52",
     "SalerBank": "中国工商银行股份有限公司深圳岭园支行",
     "TaxAuthor": "国家锐务总局深圳市龙岗区税务局第五税务所",
     "ManuModel": "江淮牌HFC7007EYBD6",
     "CertificateNum": "WCH0794J0976801",
     "Purchaser": "苏子潇",
     "VehicleType": "纯电动轿车",
     "InvoiceCode": "14975047560",
     "PriceTax": "壹拾万陆仟壹佰圆整",
     "SalerPhone": "0755-83489306",
     "SalerAddress": "深圳市龙岗区龙岗街道百世国际汽车城",
     "Origin": "安徽省合肥市",
     "EngineNum": "18958407",
     "Tax": "14634.48",
     "PurchaserCode": "5135934475603742222",
     "TaxAuthorCode": "14037589413",
     "SalerAccountNum": "中国工商银行股份有限公司深圳岭园支行",
     "SalerCode": "91449283464582922 78H"
  }
}
```


车辆合格证

支持对车辆合格证的23个关键字段进行结构化识别，包括合格证编号、发证日期、车辆制造企业名、车辆品牌、车辆名称、车辆型号、车架号、车身颜色、发动机型号、发动机号、燃料种类、排量、功率、排放标准、轮胎数、轴距、轴数、转向形式、总质量、整备质量、驾驶室准乘人数、最高设计车速、车辆制造日期。

```
 $image = file_get_contents('example.jpg');

// 车辆合格证
$client->vehicleCertificate($image);

// 如果有可选参数
$options = array();

// 带参数调用车型识别
$client->vehicleCertificate($image, $options);
 // 参数图片url
$url = "https//www.x.com/sample.jpg"
client.vehicleCertificateUrl($url, $options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
CertificationNo	是	string	合格证编号
CertificateDate	是	string	发证日期
Manufacturer	是	string	车辆制造企业名
CarBrand	是	string	车辆品牌
CarName	是	string	车辆名称
CarModel	是	string	车辆型号
VinNo	是	string	车架号
CarColor	是	string	车身颜色
EngineType	是	string	发动机型号
EngineNo	是	string	发动机号
FuelType	是	string	燃料种类
Displacement	是	string	排量
Power	是	string	功率
EmissionStandard	是	string	排放标准
TyreNum	是	string	轮胎数
Wheelbase	是	string	轴距
AxleNum	是	string	轴数
SteeringType	是	string	转向形式
TotalWeight	是	string	总质量
SaddleMass	是	string	整备质量
LimitPassenger	是	string	驾驶室准乘人数
SpeedLimit	是	string	最高设计车速
ManufactureDate	是	string	车辆制造日期

返回示例

```
{
  "log_id": 14814098736243057,
  "words_result_num": 23,
  "words_result": {
     "ManufactureDate": "2016年10月13日",
     "CarColor": "红",
     "LimitPassenger": "2",
     "EngineType": "WP12.460E50",
     "TotalWeight": "25000",
     "Power": "338",
     "CertificationNo": "WEK29JX98645437",
     "FuelType": "汽油",
     "Manufacturer": "陕西汽车集团有限责任公司",
     "SteeringType": "方向盘",
     "Wheelbase": "3175+1350",
     "SpeedLimit": "105",
     "EngineNo": "1418K129178",
     "SaddleMass": "8600",
     "AxleNum": "3",
     "CarModel": "SX4250MC4",
     "VinNo": "LZGJHYD83JX197344",
     "CarBrand": "陕汽牌",
     "EmissionStandard": "GB17691-2005国V,GB3847-2005",
     "Displacement": "11596",
     "CertificateDate": "2018年11月28日",
     "CarName": "牵引汽车",
     "TyreNum": "10"
  }
}
```


户口本识别

支持对户口本内常住人口登记卡的全部 22 个字段进行结构化识别，包括户号、姓名、与户主关系、性别、出生地、民族、出生日期、身份证号、本市县其他住址、曾用名、籍贯、宗教信仰、身高、血型、文化程度、婚姻状况、兵役状况、服务处所、职业、何时由何地迁往本市、何时由何地迁往本址、登记日期。

```
 $image = file_get_contents('example.jpg');

// 户口本识别
$client->householdRegister($image);

// 如果有可选参数
$options = array();

// 带参数调用车型识别
$client->householdRegister($image, $options);
 // 参数图片url
$url = "https//www.x.com/sample.jpg"
$client->householdRegisterUrl($url, $options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和image二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
Name	是	object{}	姓名
+ words	是	string	所属字段的具体内容，以下各字段均含有此元素
Relationship	是	object{}	户主或与户主关系
Sex	是	object{}	性别
BirthAddress	是	object{}	出生地
Nation	是	object{}	民族
Birthday	是	object{}	生日
CardNo	是	object{}	身份证号
HouseholdNum	是	object{}	户号
FormerName	是	object{}	曾用名
Hometown	是	object{}	籍贯
OtherAddress	是	object{}	本市（县）其他住址
Belief	是	object{}	宗教信仰
Height	是	object{}	身高
BloodType	是	object{}	血型
Education	是	object{}	文化程度
MaritalStatus	是	object{}	婚姻状况
VeteranStatus	是	object{}	兵役状况
WorkAddress	是	object{}	服务处所
Career	是	object{}	职业
WWToCity	是	object{}	何时由何地迁来本市(县)
WWHere	是	object{}	何时由何地迁往本址
Date	是	object{}	登记日期

返回示例

```
{
  "log_id": 1301870459,
  "words_result": {
      "BirthAddress": {
          "words": "河南洛阳市郊区"
      },
      "Birthday": {
          "words": "2016-07-28"
      },
      "CardNo": {
          "words": "410311201607282825"
      },
      "Name": {
          "words": "孙翌晨"
      },
      "Nation": {
          "words": "汉族"
      },
      "Relationship": {
          "words": "户主"
      },
      "Sex": {
          "words": "男"
      },
  },
  "words_result_num": 7
}
```


手写文字识别

支持对图片中的手写中文、手写数字进行检测和识别，针对不规则的手写字体进行专项优化，识别准确率可达90%以上。

```
 $image = file_get_contents('example.jpg');

// 手写文字识别
$client->handwriting($image);

// 如果有可选参数
$options = array();

// 带参数调用车型识别
$client->handwriting($image, $options);
 // 参数图片url
$url = "https//www.x.com/sample.jpg"
client.handwritingUrl($url, $options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
recognize_granularity	否	string	big、small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true/false	是否返回识别结果中每一行的置信度，默认为false，不返回置信度
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: true：检测朝向； false：不检测朝向

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	float	当请求参数 **probability=true** 时返回该字段，表示识别结果中每一行的置信度值，包含： - **average**： 行置信度平均值 - **variance**：行置信度方差 - **min**：行置信度最小值
direction	否	int32	图像方向，当detect_direction=true时存在 -1:未定义， 0:正向， 1: 逆时针90度， 2:逆时针180度， 3:逆时针270度

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
        "location": {
            "left": 56,
            "top": 0,
            "width": 21,
            "height": 210
        },
        "words": "3"
    }
  ],
  "words_result_num": 1
}
```

飞机行程单识别

支持对飞机行程单的24个字段进行结构化识别，包括电子客票号、印刷序号、姓名、始发站、目的站、航班号、日期、时间、票价、身份证号、承运人、民航发展基金、保险费、燃油附加费、其他税费、合计金额、填开日期、订票渠道、客票级别、座位等级、销售单位号、签注、免费行李、验证码。 同时，支持单张行程单上的多航班信息识别。

```
 $image = file_get_contents('example.jpg');

// 飞机行程单识别
$client->airTicket($image);

// 如果有可选参数
$options = array();

// 带参数调用车型识别
$client->airTicket($image, $options);
 // 参数图片url
$url = "https//www.x.com/sample.jpg"
client.airTicketUrl($url, $options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
multi_detect	否	bool	true/false	控制是否开启多航班信息识别功能，**默认值：false** - **true：开启多航班信息识别功能**，开启后返回结果中对应字段格式将改为数组类型 - **false：** 不开启，仅识别单一航班信息

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
name	是	string	姓名
starting_station	是	string	始发站
destination_station	是	string	目的站
flight	是	string	航班号
date	是	string	日期
ticket_number	是	string	电子客票号码
fare	是	string	票价
dev_fund	是	string	民航发展基金/基建费
fuel_surcharge	是	string	燃油附加费
other_tax	是	string	其他税费
ticket_rates	是	string	合计金额
issued_date	是	string	填开日期
id_num	是	string	身份证号
carrier	是	string	承运人
time	是	string	时间
issued_by	是	string	订票渠道
serial_number	是	string	印刷序号
insurance	是	string	保险费
fare_basis	是	string	客票级别
class	是	string	座位等级
agent_code	是	string	销售单位号
endorsement	是	string	签注
allow	是	string	免费行李
ck	是	string	验证码

返回示例

```
 // 识别单航班信息（multi_detect=false，或参数缺省）
{
    "log_id": 7306800033425229106,
    "direction": 0,
    "words_result_num": 18,
    "words_result": {
        "insurance": "20.00",
        "date": "2019-10-22",
        "allow": "20K",
        "flight": "CA6589",
        "issued_by": "中国国际航空服务有限公司",
        "starting_station": "武汉",
        "fare": "260.00",
        "endorsement": "不得签转改期退转",
        "ticket_rates": "350.00",
        "ck": "5866",
```

```
        "serial_number": "51523588676",
        "ticket_number": "7843708871196",
        "fuel_surcharge": "EXEMPT",
        "carrier": "南航",
        "issued_date": "2019-10-30",
        "other_tax": "",
        "fare_basis": "NREOW",
        "id_num": "411201123909020877",
        "destination_station": "合肥",
        "name": "郭达",
        "agent_code": "BJS19197300025",
        "time": "21:25",
        "class": "N",
        "dev_fund": "50.00"
    }
}

// 识别多航班信息（multi_detect=true)
{
    "words_result": {
        "insurance": [
            {
                "word": "XXX"
            }
        ],
        "date": [
            {
                "word": "2019-10-18"
            },
            {
                "word": "2019-10-21"
            }
        ],
        "flight": [
            {
                "word": "CZ3565"
            },
            {
                "word": "CZ3566"
            }
        ],
        "issued_by": [
            {
                "word": "上海携程旅行社有限公司"
            }
        ],
        "starting_station": [
            {
                "word": "北京"
            }
        ],
        "fare": [
            {
                "word": "1080.00"
            }
        ],
        "ticket_rates": [
            {
                "word": "1420.00"
            }
        ],
        "serial_number": [
```

```
            {
                "word": "45956029770"
            }
        ],
        "ticket_number": [
            {
                "word": "7849648364314"
            }
        ],
        "fuel_surcharge": [
            {
                "word": "240.00"
            }
        ],
        "carrier": [
            {
                "word": "南航"
            },
            {
                "word": "南航"
            }
        ],
        "issued_date": [
            {
                "word": "2019-09-18"
            }
        ],
        "other_tax": [],
        "id_num": [
            {
                "word": "0789654700"
            }
        ],
        "destination_station": [
            {
                "word": "深圳"
            },
            {
                "word": "北京"
            }
        ],
        "name": [
            {
                "word": "姚佳"
            }
        ],
        "time": [
            {
                "word": "13:55"
            },
            {
                "word": "16:30"
            }
        ],
        "dev_fund": [
            {
                "word": "100.00"
            }
        ]
    },
    "log_id": "1280814270572920832",
    "words_result_num": 18
}
```

```
}
```

通用机打发票

支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别，包括发票类型、发票号码、发票代码、开票日期、合计金额大写、合计金额小写、商品名称、商品单位、商品单价、商品数量、商品金额、机打代码、机打号码、校验码、销售方名称、销售方纳税人识别号、购买方名称、购买方纳税人识别号、合计税额等。

```
 $image = file_get_contents('example.jpg');

// 通用机打发票
$client->invoice($image);

// 如果有可选参数
$options = array();

// 带参数调用车型识别
$client->invoice($image, $options);
 // 参数图片url
$url = "https//www.x.com/sample.jpg"
client.invoiceUrl($url, $options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	string	true/false	是否输出位置信息，true：输出位置信息，false：不输出位置信息，默认false

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	否	string	发票类型
InvoiceCode	否	string	发票代码
InvoiceNum	否	string	发票号码
InvoiceDate	否	string	开票日期
AmountInFiguers	否	string	合计金额小写
AmountInWords	否	string	合计金额大写
CommodityName	否	string	商品名称
CommodityUnit	否	string	商品单位
CommodityPrice	否	string	商品单价
CommodityNum	否	string	商品数量
CommodityAmount	否	string	商品金额
MachineCode	否	string	机打代码
MachineNum	否	string	机打号码
CheckCode	否	string	校验码
SellerName	否	string	销售方名称
SellerRegisterNum	否	string	销售方纳税人识别号
PurchaserName	否	string	购买方名称
PurchaserRegisterNum	否	string	购买方纳税人识别号
TotalTax	否	string	合计税额
Province	否	string	省
City	否	string	市
Time	否	string	时间
SheetNum	否	string	联次

返回示例

```
{
  "log_id": 4423022131715883558,
  "direction": 0,
  "words_result_num": 22,
  "words_result": {
    "City": "",
    "InvoiceNum": "01445096",
    "SellerName": "百度餐饮店",
    "IndustrSort": "生活服务",
    "Province": "广东省",
    "CommodityAmount": [
      {
        "word": "183.00",
        "row": "1"
      }
    ],
    "InvoiceDate": "2020年07月28日",
    "PurchaserName": "中信建投证券股份有限公司",
    "CommodityNum": [],
    "InvoiceCode": "144001901511",
    "CommodityUnit": [],
    "SheetNum": "",
    "PurchaserRegisterNum": "91442300845348OX9",
    "Time": "",
    "CommodityPrice": [],
    "AmountInFiguers": "183.00",
    "AmountInWords": "壹佰捌拾叁元整",
    "CheckCode": "61042119820421061301",
    "TotalTax": "183.00",
    "InvoiceType": "广东通用机打发票",
    "SellerRegisterNum": "61042119820421061301",
    "CommodityName": [
      {
        "word": "餐费",
        "row": "1"
      }
    ]
  }
}
```


护照识别

支持对中国大陆护照个人资料页所有15个字段进行结构化识别，包括国家码、护照号、姓名、姓名拼音、性别、出生地点、出生日期、签发地点（不支持境外签发地）、签发日期、有效期、签发机关、护照类型、国籍、MRZCode1、MRZCode2。

```
 $image = file_get_contents('example.jpg');

// 护照识别
$client->passport($image);

// 如果有可选参数
$options = array();

// 带参数调用车型识别
$client->passport($image, $options);
 // 参数图片url
$url = "https//www.x.com/sample.jpg"
$client->passportUrl($url, $options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
-location	是	uint32	水平坐标
-words	是	string	识别内容

返回示例


```
{
  "log_id": 7377468409496932872,
  "words_result_num": 14,
  "words_result": {
     "国家码": {
        "location": {
           "width": 59,
           "top": 200,
           "left": 762,
           "height": 26
        },
        "words": "CHN"
     },
     "护照签发地点": {
        "location": {
           "width": 236,
           "top": 505,
           "left": 558,
           "height": 43
        },
        "words": "山东/SHANDONG"
     },
     "MRZCode2": {
        "location": {
           "width": 1252,
           "top": 797,
           "left": 145,
           "height": 88
        },
        "words": "E898657303CHNSDM0702<<<<<<<<<<<<<<"
     },
     "有效期至": {
        "location": {
```

```
            "width": 287,
            "top": 528,
            "left": 955,
            "height": 46
        },
        "words": "20261004"
    },
    "签发机关": {
        "location": {
            "width": 271,
            "top": 583,
            "left": 552,
            "height": 42
        },
        "words": "出入境管理局"
    },
    "MRZCode1": {
        "location": {
            "width": 1201,
            "top": 781,
            "left": 162,
            "height": 45
        },
        "words": "PONSUN<<JIAJIA<<<<<<<<<<<<<<<<<<<<<<<<<<"
    },
    "护照号码": {
        "location": {
            "width": 237,
            "top": 215,
            "left": 994,
            "height": 48
        },
        "words": "E89865730"
    },
    "签发日期": {
        "location": {
            "width": 279,
            "top": 453,
            "left": 955,
            "height": 46
        },
        "words": "20161005"
    },
    "出生地点": {
        "location": {
            "width": 216,
            "top": 429,
            "left": 564,
            "height": 43
        },
        "words": "山东/SHANDONG"
    },
    "姓名": {
        "location": {
            "width": 159,
            "top": 247,
            "left": 581,
            "height": 34
        },
        "words": "孙嘉佳"
    },
    "姓名拼音": {
```

```
            "location": {
                "width": 229,
                "top": 279,
                "left": 578,
                "height": 41
            },
            "words": "SUN,JIAJIA"
        },
        "国籍": {
            "location": {
                "width": 209,
                "top": 366,
                "left": 695,
                "height": 42
            },
            "words": "中国/CHINESE"
        },
        "生日": {
            "location": {
                "width": 202,
                "top": 382,
                "left": 950,
                "height": 39
            },
            "words": "19950723"
        },
        "性别": {
            "location": {
                "width": 73,
                "top": 357,
                "left": 570,
                "height": 34
            },
            "words": "男/M"
        }
    }
}
```


网约车行程单识别

对各大主要服务商的网约车行程单进行结构化识别，包括滴滴打车、花小猪打车、高德地图、曹操出行、阳光出行，支持识别服务商、行程开始时间、行程结束时间、车型、总金额等16 个关键字段。

```php
 $client= new AipOcr("APP_ID", "API_KEY", "SECRET_KEY");
$pdf_file = file_get_contents('文件路径');
$url = "https//www.x.com/sample.jpg";
$image = file_get_contents('文件路径');
  $options = array();
// 调用网约车行程单识别
var_dump(client->onlineTaxiItinerary($image));
var_dump(client->onlineTaxiItineraryUrl($url));
var_dump(client->onlineTaxiItineraryPdf($pdf_file,$options));

// 如果有可选参数
$options["pdf_file_num", 1];
var_dump(client->onlineTaxiItineraryPdf($pdf_file, $options));
```

请求参数详情

参数	是否必选	类型	说明
image	和url二选一	string	图像数据，base64编码后进行urlencode，base64编码去除编码头（data:image/jpeg;base64），要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式。 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效，请注意关闭URL防盗链
pdf_file	和image/url 三选一	string	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ ServiceProvider	是	string	服务商
+ StartTime	是	string	行程开始时间
+ EndTime	是	string	行程结束时间
+ Phone	是	string	行程人手机号
+ ApplicationDate	是	string	申请日期
+ TotalFare	是	string	总金额
+ ItemNum	是	array	行程信息中包含的行程数量
+ Items	是	array	行程信息
++ ItemId	是	string	行程信息的对应序号
++ PickupTime	是	string	上车时间
++ PickupDate	是	string	上车日期
++ CarType	是	string	车型
++ Distance	是	string	里程
++ StartPlace	是	string	起点
++ DestinationPlace	是	string	终点
++ City	是	string	城市
++ Fare	是	string	金额
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 1385196013945356288,
  "words_result_num": 7
  "words_result": {
     "TotalFare": "2316",
     "EndTime": "2020-07-30 19:00",
     "Phone": "13000000000",
     "ServiceProvider": "滴滴企业版",
     "StartTime": "2020-07-01 16:00",
     "ApplicationDate": "2017-12-08",
     "ItemId": "3"
     "items": [
        {
           "ItemId": "1",
           "StartPlace": "鱼化寨地铁-D口",
           "PickupTime": "16:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.7",
           "PickupDate": "20-07-01",
           "DestinationPlace": "创新港",
           "Fare": "20.86"
        },
        {
           "ItemId": "2",
           "StartPlace": "科学园东门",
           "PickupTime": "14:56",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "91",
           "PickupDate": "20-07-02",
           "DestinationPlace": "鱼化寨地铁站",
           "Fare": "18.58"
        },
        {
           "ItemId": "3",
           "StartPlace": "中俄丝路创新园东门",
           "PickupTime": "19:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.1",
           "PickupDate": "20-07-30",
           "DestinationPlace": "新门地铁站",
           "Fare": "20.38"
        },
     ],
  },
}
```


磅单识别

结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单。

```
 $client= new AipOcr("APP_ID", "API_KEY", "SECRET_KEY");
$pdf_file = file_get_contents('文件路径');
$url = "https//www.x.com/sample.jpg";
$image = file_get_contents('文件路径');
// 调用磅单识别
var_dump(client->weightNote($image));
var_dump(client->weightNoteUrl($url));
var_dump(client->weightNotePdf($pdf_file));
//  如果有可选参数
$options = array();
$options["pdf_file_num", 1];
$options["probability", false];
var_dump(client->weightNote($image, $options));
var_dump(client->weightNoteUrl($url, $options));
var_dump(client->weightNotePdf($pdf_file, $options));
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式。 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ PlateNum	是	object	车牌号
+ PrintTime	是	object	打印时间
+ CrossWeight	是	object	毛重
+ TareWeight	是	object	皮重
+ NetWeight	是	object	净重
+ SendingCompany	是	object	发货单位
+ ReceivingCompany	是	object	收货单位
+ DeliveryNumber	是	object	单号
++ word	是	string	字段识别结果，以上各字段均包含此参数
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "words_result": [
    {
      "TareWeight": [
      {
            "word": "14.20"
        }
      ],
      "CrossWeight": [
        {
          "word": "50.70"
        }
      ],
      "PlateNum": [
        {
          "word": "京A12356"
        }
      ],
      "SendingCompany": [
       {
          "word": "宣化县耿矿煤业有限公司"
        }
      ],
      "DeliveryNumber": [
       {
          "word": "278933000"
        }
      ],
      "ReceivingCompany": [
        {
          "word": "宁夏市京裕达实业公司"
        }
      ],
      "PrintTime": [
        {
          "word": "2020年1月1日"
        }
      ],
      "NetWeight": [
        {
          "word": "36.50"
        }
      ]
    }
  ],
  "words_result_num": 1,
  "log_id": 1428311410130160734
}
```


医疗费用明细识别

SDK 调用示例

```
 $client= new AipOcr("APP_ID", "API_KEY", "SECRET_KEY");
$url = "https//www.x.com/sample.jpg";
$image = file_get_contents('文件路径');

 // 调用医疗费用明细识别
var_dump(client->medicalDetail($image))
var_dump(client->medicalDetailUrl($url))
```

接口详情

可参考API文档：医疗费用明细识别

印章识别

SDK 调用示例

```
 $client= new AipOcr("APP_ID", "API_KEY", "SECRET_KEY");
$pdf_file = file_get_contents('文件路径');
$url = "https//www.x.com/sample.jpg";
$image = file_get_contents('文件路径');

 // 调用印章识别
var_dump(client->sealRecognize($image));
var_dump(client->sealRecognizeUrl($url));
var_dump(client->sealRecognizePdf($pdf_file));
```

接口详情

可参考API文档：印章识别

办公文档识别

SDK 调用示例

```
  $client= new AipOcr("APP_ID", "API_KEY", "SECRET_KEY");
  $pdf_file = file_get_contents('文件路径');
  $url = "https//www.x.com/sample.jpg";
  $image = file_get_contents('文件路径');

 // 调用办公文档识别
  var_dump(client->docAnalysisOffice($image));
  var_dump(client->docAnalysisOfficeUrl($url));
  var_dump(client->docAnalysisOfficePdf($pdf_file));
```

接口详情

可参考API文档：办公文档识别

医疗费用明细识别

```
 $client= new AipOcr("APP_ID", "API_KEY", "SECRET_KEY");
$url = "https//www.x.com/sample.jpg";
$image = file_get_contents('文件路径');
// 调用医疗费用明细识别
var_dump(client->medicalDetail($image))
var_dump(client->medicalDetailUrl($url))
//  如果有可选参数
$options = array();
$options["location", ];
$options["probability", false];
var_dump(client->medicalDetail($image, $options))
var_dump(client->medicalDetailUrl($url, $options))
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式。 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省** - **false：**不返回字段位置信息 - **true：**返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false：**不返回字段识别结果的置信度 - **true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ Name	是	object	姓名
+ Date	是	object	日期
+ PatientID	是	object	病人ID
+ TotalAmount	是	object	总金额
+ word	是	string	字段识别结果，以上各字段均包含此参数
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+ CostDetail	是	array[]	项目明细

CostDetail字段包含多个Array，每个数组包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：项目类型、项目名称、单价、数量、规格、金额
++ word	word_name字段对应的识别结果

返回示例

```
{
  "log_id": 1397090241579319296,
  "words_result_num": 5,
  "words_result": {
    "PatientID": {
        "word": "23683829"
    },
    "TotalAmount": {
        "word": "600.00"
    },
    "Date": {
        "word": "2020年11月04日"
    },
    "Name": {
        "word": "范浩"
    },
    "CostDetail": [
        [
            {
                "word_name": "清单项目名称",
                "word": "普通过敏原(新)筛查"
            },
            {
                "word_name": "单价",
                "word": "580.00"
            },
            {
                "word_name": "数量",
                "word": "1.00"
            },
            {
                "word_name": "金额",
                "word": "580.00"
            },
            {
                "word_name": "规格",
                "word": "次"
            },
            {
                "word_name": "项目类型",
                "word": "化验费"
            }
        ],
        [
            {
                "word_name": "清单项目名称",
                "word": "总IgE测定"
            },
            {
                "word_name": "单价",
                "word": "20.00"
            },
            {
                "word_name": "数量",
                "word": "1.00"
            },
            {
                "word_name": "金额",
                "word": "20.00"
            },
            {
                "word_name": "规格",
```

```
                "word": "次"
            },
            {
                "word_name": "项目类型",
                "word": "化验费"
            }
        ]
    ],
  },
}
```

图文转换器（接口版）--提交请求

图文转换器对应的接口版产品，可识别图片/PDF文件中的文本内容，进行智能版式分析，并转换为保留原文档版式的Word、Excel文档，返回文档下载连接，支持含表格、印章、手写等内容的文档。满足文档版式还原、企业档案电子化等信息管理需求。如需直接在线使用轻应用，可到控制台-图文转换器使用。

```php
 $client= new AipOcr("APP_ID", "API_KEY", "SECRET_KEY");
$pdf_file = file_get_contents('文件路径');
$url = "https//www.x.com/sample.jpg";
$image = file_get_contents('文件路径');

 // 调用图文转换器（接口版）--提交请求
var_dump(client->docConvertRequestV1($image));
var_dump(client->docConvertRequestV1Url($url));
var_dump(client->docConvertRequestV1Pdf($pdf_file));
```

请求参数详情

参数	是否必选	类型	说明
image	和 url/pdf_file 三选一	string	图像数据，base64编码后进行urlencode，需去掉编码头（）要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **优先级：** image > url > pdf_file，当image字段存在时，url、pdf_file字段失效。 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。
url	和 image/pdf_file 三选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **优先级：** image > url > pdf_file，当image字段存在时url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不不超过10M **优先级：** image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容；**若不传入，默认识别文件所有页，页码从1开始**

返回参数详情

字段	类型	说明
success	bool	当前请求状态； true 表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该请求生成的task_id，后续使用该task_id获取识别结果
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": 12345,
  "result":{
     "task_id":"task-xxxxxxx",
  },
  "code":1001,
  "message": "Create task successfully!"
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
  "success":false,
  "log_id": 12345,
  "error_code": 216401,
  "error_msg": "Create task failed!"
}
```

图文转换器（接口版）--获取结果

```
 $client= new AipOcr("APP_ID", "API_KEY", "SECRET_KEY");
$task_id = "xxxxxxxx";

 // 调用图文转换器（接口版）--获取结果
var_dump(client->docConvertResultV1($task_id));
```

请求参数详情

参数	是否必选	类型	说明
task_id	是	string	发送提交请求时返回的task_id

返回参数详情

字段	类型	说明
success	bool	当前请求状态； true表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该文件对应请求的task_id
+ ret_code	int	识别状态，1：任务未开始；2：进行中；3：已完成
+ ret_msg	string	识别状态信息：任务未开始；进行中；已完成
+ percent	int	文档转换进度（百分比）
+ result_data	dict	识别结果字符串，返回word、excel的文件分别的下载地址
+ +word	string	还原后的word文件的下载地址，**文件识别失败时返回""**
+ +excel	string	还原后的Excel文件的下载地址，**若文档中没有表格则返回""**
+ create_time	datetime	任务创建时间
+ start_time	datetime	任务开始时间
+ end_time	datetime	任务结束时间
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": "xxxxxx",
  "result":{
      "task_id":"task-xxxxxxx",
      "ret_code": 3,
      "ret_msg": "已完成",
      "percent": 100,
      "result_data": {
          "word": "word_download_url",
          "excel": "",
      },
      "create_time": "2023-01-17 11:06:12",
      "start_time": "2023-01-17 11:06:13",
      "end_time": "2023-01-17 11:06:15"
  },
  "code":1001,
  "message": "Query task successfully!""
}
```

若查询的task_id不存在，返回result为{}。 请求失败响应体示例如下：

```
{
  "code":1001,
  "log_id":1635891796603052032,
  "message":"Query task successfully!",
  "result":{},
  "success":true
}
```


表格文字识别同步接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别同步接口。

表格文字识别

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别。

表格识别结果

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别结果。

表格识别接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别接口。

错误信息

错误返回格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

错误码

SDK本地检测参数返回的错误码：

error_code	error_msg	描述
SDK100	image size error	图片大小超限
SDK101	image length error	图片边长不符合要求
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时
SDK109	unsupported image format	不支持的图片格式

服务端返回的错误码：

错误码	错误信息	描述
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
	Open api	

17	daily request limit reached	每天流量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
282000	internal error	服务器内部错误，如果您使用的是高精度接口，报这个错误码的原因可能是您上传的图片中文字过多，识别超时导致的，建议您对图片进行切割后再识别，其他情况请再次请求， 如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
282003	missing parameters: {参数名}	请求参数缺失
	batch	

282005	processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282110	urls not exit	URL参数不存在，请核对URL后再次提交
282111	url format illegal	URL格式非法，请检查url格式是否符合相应接口的入参要求
282112	url download timeout	url下载超时，请检查url对应的图床/图片无法下载或链路状况不好，您可以重新尝试以下，如果多次尝试后仍不行，建议更换图片地址
282113	url response invalid	URL返回无效参数
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误

C#语言

简介

Hi，您好，欢迎使用百度文字识别服务。

本文档主要针对C#开发者，描述百度文字识别接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 在百度智能云控制台内提交工单，咨询问题类型请选择**人工智能服务**；
- 如有疑问，进入AI社区交流：http://ai.baidu.com/forum/topic/list/164

接口能力

接口名称	接口能力简要描述
通用文字识别	识别图片中的文字信息
通用文字识别（高精度版）	更高精度地识别图片中的文字信息
通用文字识别（含位置信息版）	识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（高精度含位置版）	更高精度地识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（含生僻字版）	识别图片中的文字信息（包含对常见字和生僻字的识别）
网络图片文字识别	识别一些网络上背景复杂，特殊字体的文字

网络图片文字识别（含位置版）	识别网络图片中的文字内容（包含文字区域的坐标信息）
身份证识别	识别身份证正反面的文字信息
银行卡识别	识别银行卡的卡号并返回发卡行和卡片性质信息
驾驶证识别	识别机动车驾驶证所有关键字段
行驶证识别	识别机动车行驶证所有关键字段
车牌识别	识别中国大陆各类机动车车牌信息
营业执照识别	对营业执照进行识别
表格文字识别	自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容
通用票据识别	对各类票据图片（医疗票据，保险保单等）进行文字识别，并返回文字在图片中的位置信息
增值税发票识别	对增值税发票进行文字识别，并结构化返回字段信息，支持增值税专票、普票、电子发票
出租车票识别	针对全国各大城市出租车票的发票号码、发票代码、车号、日期、时间、金额等进行结构化识别
VIN码识别	对车辆车架、挡风玻璃上的VIN码进行识别
火车票识别	支持对大陆火车票的车票号、始发站、目的站、车次、日期、票价、席别、姓名进行结构化识别
飞机行程单识别	支持对飞机行程单的24个字段进行结构化识别
二维码识别	对图片中的二维码、条形码进行检测和识别，返回存储的文字信息
数字识别	识别图片中的数字，适用于手机号提取、快递单号提取、充值号码提取等场景
手写文字识别	支持对图片中的手写中文、手写数字进行检测和识别
护照识别	支持对中国大陆护照个人资料页所有15个字段进行结构化识别
户口本识别	对出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码字段进行识别
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果
通用机打发票	支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别
机动车销售发票	支持对机动车销售发票的26个关键字段进行结构化识别
车辆合格证	支持对车辆合格证的23个关键字段进行结构化识别
通用机打发票	对国家/地方税务局发行的横/竖版通用机打发票进行结构化识别
护照识别	支持对中国大陆护照个人资料页所有11个字段进行结构化识别
医疗费用明细识别	支持识别全国医疗费用明细识别
网约车行程单识别	对国家/地方税务局发行的横/对各大主要服务商的网约车行程单进行结构化识别
磅单识别	结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单
仪器仪表盘读数识别	适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号
自定义模板文字识别	针对固定版式卡证票据提供的 OCR 定制化产品，可由用户自助创建识别模板和分类器，实现对任意版式卡证票据进行自动分类并结构化输出识别结果
医疗费用明细	支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含

识别	项目类型、项目名称、单价、数量、规格、金额
办公文档识别	可对办公类文档的版面进行分析，输出图、表、标题、文本、目录、栏、页眉、页脚、页码和脚注的位置，并输出分版块内容的OCR识别结果
印章识别	检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别
机动车登记证书识别	对机动车登记证书的编号、机动车所有人、登记机关、车辆类型、发证机关章等15个关键字段进行结构化识别
智能财务票据识别	对增值税发票、卷票、火车票、出租车票、机票行程单等13类票据混贴的图片进行切分识别
增值税发票验真	支持9种增值税发票的真伪及字段信息准确性校验，包括增值税专票、电子专票、普票、电子普票、卷票、通行费增值税电子普票、货运专票、机动车销售发票、二手车销售发票，支持返回票面的全部信息
医疗发票识别	支持识别全国各地门诊/住院发票的 业务流水号、发票号、住院号、门诊号、病例号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付等关键字段。支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数
门脸文字识别	识别图片中的门脸文字信息，自动过滤非门脸文字内容，接口返回门脸名称、描述文字和置信度
车辆证照混贴识别	对机动车行驶证主页及副页、驾驶证主页及副页在同一张图片上的场景进行结构化识别
公式识别	对试卷中的数学公式及题目内容进行识别
图文转换器	可识别图片/PDF文档版面布局，提取文字内容，并转换为保留原文档版式的Word、Excel文档，方便二次编辑和复制，可支持含表格、印章、水印、手写等内容的文档

版本更新记录

上线日期	版本号	更新内容
2021.12.11	4.15.8	新增： 网约车行程单识别，磅单识别，医疗明细识别
2021.6.3	4.15.7	二维码、行程单、机动车销售发票、车辆合格证、试卷分析与识别、手写、护照、户口本、通用机打
2021.3.18	4.15.5	手写识别、二维码识别、试卷分析与识别、数字识别、办公文档识别、印章识别、仪器仪表盘读数识别、网络图片文字识别（含位置版）
2021.1.4	4.15.4	增加增值税发票识别，出租车票，vin码，火车票，数字识别
2018.4.2	3.4.0	新增.Net Core支持；新增表格识别同步接口
2018.1.11	3.3.1	新增自定义文字识别接口
2017.12.21	3.3.0	接口升级
2017.9.12	3.0.0	更新SDK打包方式：所有AI服务集成一个SDK
2017.8.24	1.6.0	OCR 新增营业执照识别
2017.8.11	1.5.0	OCR 新增通用票据识别
2017.7.28	1.4.0	OCR 新增图片url识别；新增高精度文字识别
2017.7.13	1.3.0	OCR加入车牌识别；Bug fix
2017.6.30	1.2.0	OCR加入表格识别
2017.6.16	1.1.0	新增驾驶证、行驶证识别接口
2017.5.4	1.0.0	第一版！

快速入门

安装文字识别 C# SDK

C# SDK 现已开源! https://github.com/Baidu-AIP/dotnet-sdk

支持平台：.Net Framework 3.5 4.0 4.5， .Net Core 2.0

方法一：使用Nuget管理依赖 （推荐）

在NuGet中搜索 `Baidu.AI`，安装最新版即可。

packet地址 https://www.nuget.org/packages/Baidu.AI/

方法二：下载安装

文字识别 C# SDK目录结构

```
Baidu.Aip
  ├─── net35
  │    ├─── AipSdk.dll            // 百度AI服务 windows 动态库
  │    ├─── AipSdk.xml            // 注释文件
  │    └─── Newtonsoft.Json.dll   // 第三方依赖
  ├─── net40
  ├─── net45
  └─── netstandard2.0
       ├─── AipSdk.deps.json
       └─── AipSdk.dll
```

> 如果需要在 Unity 平台使用，可引用工程源码自行编译。

安装

1.在官方网站下载C# SDK压缩工具包。

2.解压后，将 `AipSdk.dll` 和 `Newtonsoft.Json.dll` 中添加为引用。

新建交互类

Baidu.Aip.Ocr.Ocr是文字识别的交互类，为使用文字识别的开发人员提供了一系列的交互方法。

用户可以参考如下代码新建一个交互类：

```
 // 设置APPID/AK/SK
var APP_ID = "你的 App ID";
var API_KEY = "你的 Api Key";
var SECRET_KEY = "你的 Secret Key";

var client = new Baidu.Aip.Ocr.Ocr(API_KEY, SECRET_KEY);
client.Timeout = 60000;  // 修改超时时间
```

在上面代码中，常量`APP_ID`在百度智能云控制台中创建，常量`API_KEY`与`SECRET_KEY`是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的**应用列表**中查看。

注意：如您以前是百度智能云的老用户，其中`API_KEY`对应百度智能云的“Access Key ID”，`SECRET_KEY`对应百度智能云的“Access Key Secret”。

接口说明

通用文字识别

用户向服务请求识别某张图中的所有文字

```
 public void GeneralBasicDemo() {
 var image = File.ReadAllBytes("图片文件路径");
 // 调用通用文字识别, 图片参数为本地图片，可能会抛出网络等异常，请使用try/catch捕获
 var result = client.GeneralBasic(image);
 Console.WriteLine(result);
 // 如果有可选参数
 var options = new Dictionary<string, object>{
    {"language_type", "CHN_ENG"},
    {"detect_direction", "true"},
    {"detect_language", "true"},
    {"probability", "true"}
 };
 // 带参数调用通用文字识别, 图片参数为本地图片
 result = client.GeneralBasic(image, options);
 Console.WriteLine(result);
}
public void GeneralBasicUrlDemo() {
 var url = "https//www.x.com/sample.jpg";

 // 调用通用文字识别, 图片参数为远程url图片，可能会抛出网络等异常，请使用try/catch捕获
 var result = client.GeneralBasicUrl(url);
 Console.WriteLine(result);
 // 如果有可选参数
 var options = new Dictionary<string, object>{
    {"language_type", "CHN_ENG"},
    {"detect_direction", "true"},
    {"detect_language", "true"},
    {"probability", "true"}
 };
 // 带参数调用通用文字识别, 图片参数为远程url图片
 result = client.GeneralBasicUrl(url, options);
 Console.WriteLine(result);
}
```

通用文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别 返回示例

```
 {
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青岛啤酒"}
   ]
}
```

通用文字识别（高精度版）

用户向服务请求识别某张图中的所有文字，相对于通用文字识别该产品精度更高，但是识别耗时会稍长。

```
 public void AccurateBasicDemo() {
 var image = File.ReadAllBytes("图片文件路径");
 // 调用通用文字识别（高精度版），可能会抛出网络等异常，请使用try/catch捕获
 var result = client.AccurateBasic(image);
 Console.WriteLine(result);
 // 如果有可选参数
 var options = new Dictionary<string, object>{
    {"detect_direction", "true"},
    {"probability", "true"}
 };
 // 带参数调用通用文字识别（高精度版）
 result = client.AccurateBasic(image, options);
 Console.WriteLine(result);
}
```

通用文字识别（高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别（高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别（高精度版） 返回示例

参考通用文字识别返回示例

通用文字识别（含位置信息版）

用户向服务请求识别某张图中的所有文字，并返回文字在图中的位置信息。

```
 public void GeneralDemo() {
 var image = File.ReadAllBytes("图片文件路径");
 // 调用通用文字识别（含位置信息版），图片参数为本地图片，可能会抛出网络等异常，请使用try/catch捕获
 var result = client.General(image);
 Console.WriteLine(result);
 // 如果有可选参数
 var options = new Dictionary<string, object>{
     {"recognize_granularity", "big"},
     {"language_type", "CHN_ENG"},
     {"detect_direction", "true"},
     {"detect_language", "true"},
     {"vertexes_location", "true"},
     {"probability", "true"}
 };
 // 带参数调用通用文字识别（含位置信息版），图片参数为本地图片
 result = client.General(image, options);
 Console.WriteLine(result);
}
public void GeneralUrlDemo() {
 var url = "https//www.x.com/sample.jpg";

 // 调用通用文字识别（含位置信息版），图片参数为远程url图片，可能会抛出网络等异常，请使用try/catch捕获
 var result = client.GeneralUrl(url);
 Console.WriteLine(result);
 // 如果有可选参数
 var options = new Dictionary<string, object>{
     {"recognize_granularity", "big"},
     {"language_type", "CHN_ENG"},
     {"detect_direction", "true"},
     {"detect_language", "true"},
     {"vertexes_location", "true"},
     {"probability", "true"}
 };
 // 带参数调用通用文字识别（含位置信息版），图片参数为远程url图片
 result = client.GeneralUrl(url, options);
 Console.WriteLine(result);
}
```

通用文字识别（含位置信息版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
recognize_granularity	否	string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
language_type	否	string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
vertexes_location	否	string	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	string	true false		是否返回识别结果中每一行的置信度
paragraph	否	string	true false		是否输出段落信息

通用文字识别（含位置信息版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置信息版） 返回示例

```
 {
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
     "location": {
        "left": 35,
        "top": 53,
        "width": 193,
        "height": 109
     },
     "words": "感动",
     "chars": [    //recognize_granularity=small时存在
        {
           "location": {
              "left": 56,
              "top": 65,
              "width": 69,
              "height": 88
           },
           "char": "感"
        },
        {
           "location": {
              "left": 140,
              "top": 65,
              "width": 70,
              "height": 88
           },
           "char": "动"
        }
     ]
  }
  ...
]
}
```

通用文字识别（含位置高精度版）

用户向服务请求识别某张图中的所有文字，并返回文字在图片中的坐标信息，相对于通用文字识别（含位置信息版）该产品精度更高，但是识别耗时会稍长。

```
 public void AccurateDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用通用文字识别（含位置高精度版），可能会抛出网络等异常，请使用try/catch捕获
var result = client.Accurate(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{
    {"recognize_granularity", "big"},
    {"detect_direction", "true"},
    {"vertexes_location", "true"},
    {"probability", "true"}
};
// 带参数调用通用文字识别（含位置高精度版）
result = client.Accurate(image, options);
Console.WriteLine(result);
}
```

通用文字识别（含位置高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
recognize_granularity	否	string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vertexes_location	否	string	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别（含位置高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置高精度版） 返回示例

```
{
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
    "location": {
      "left": 35,
      "top": 53,
      "width": 193,
      "height": 109
    },
    "words": "感动",
    "chars": [    //recognize_granularity=small时存在
      {
        "location": {
          "left": 56,
          "top": 65,
          "width": 69,
          "height": 88
        },
        "char": "感"
      },
      {
        "location": {
          "left": 140,
          "top": 65,
          "width": 70,
          "height": 88
        },
        "char": "动"
      }
    ]
  }
  ...
]
}
```

通用文字识别（含生僻字版）

某些场景中，图片中的中文不光有常用字，还包含了生僻字，这时用户需要对该图进行文字识别，应使用通用文字识别（含生僻字版）。

```
 public void GeneralEnhancedDemo() {
 var image = File.ReadAllBytes("图片文件路径");
 // 调用通用文字识别（含生僻字版），图片参数为本地图片，可能会抛出网络等异常，请使用try/catch捕获
 var result = client.GeneralEnhanced(image);
 Console.WriteLine(result);
 // 如果有可选参数
 var options = new Dictionary<string, object>{
    {"language_type", "CHN_ENG"},
    {"detect_direction", "true"},
    {"detect_language", "true"},
    {"probability", "true"}
 };
 // 带参数调用通用文字识别（含生僻字版），图片参数为本地图片
 result = client.GeneralEnhanced(image, options);
 Console.WriteLine(result);
}
public void GeneralEnhancedUrlDemo() {
 var url = "https//www.x.com/sample.jpg";

 // 调用通用文字识别（含生僻字版），图片参数为远程url图片，可能会抛出网络等异常，请使用try/catch捕获
 var result = client.GeneralEnhancedUrl(url);
 Console.WriteLine(result);
 // 如果有可选参数
 var options = new Dictionary<string, object>{
    {"language_type", "CHN_ENG"},
    {"detect_direction", "true"},
    {"detect_language", "true"},
    {"probability", "true"}
 };
 // 带参数调用通用文字识别（含生僻字版），图片参数为远程url图片
 result = client.GeneralEnhancedUrl(url, options);
 Console.WriteLine(result);
}
```

通用文字识别（含生僻字版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别（含生僻字版） 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含生僻字版） 返回示例


```
 {
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青島睥酒"}
   ]
}
```


网络图片文字识别

用户向服务请求识别一些网络上背景复杂，特殊字体的文字。

```
 public void WebImageDemo() {
 var image = File.ReadAllBytes("图片文件路径");
 // 调用网络图片文字识别, 图片参数为本地图片，可能会抛出网络等异常，请使用try/catch捕获
 var result = client.WebImage(image);
 Console.WriteLine(result);
 // 如果有可选参数
 var options = new Dictionary<string, object>{
    {"detect_direction", "true"},
    {"detect_language", "true"}
 };
 // 带参数调用网络图片文字识别, 图片参数为本地图片
 result = client.WebImage(image, options);
 Console.WriteLine(result);
}
public void WebImageUrlDemo() {
 var url = "https//www.x.com/sample.jpg";

 // 调用网络图片文字识别, 图片参数为远程url图片，可能会抛出网络等异常，请使用try/catch捕获
 var result = client.WebImageUrl(url);
 Console.WriteLine(result);
 // 如果有可选参数
 var options = new Dictionary<string, object>{
    {"detect_direction", "true"},
    {"detect_language", "true"}
 };
 // 带参数调用网络图片文字识别, 图片参数为远程url图片
 result = client.WebImageUrl(url, options);
 Console.WriteLine(result);
}
```

网络图片文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

网络图片文字识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

网络图片文字识别 返回示例

```
 {
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青島睥酒"}
   ]
}
```

身份证识别

用户向服务请求识别身份证，身份证识别包括正面和背面。

```
 public void IdcardDemo() {
var image = File.ReadAllBytes("图片文件路径");
var idCardSide = "back";

// 调用身份证识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.Idcard(image, idCardSide);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{
    {"detect_direction", "true"},
    {"detect_risk", "false"}
};
// 带参数调用身份证识别
result = client.Idcard(image, idCardSide, options);
Console.WriteLine(result);
}
```

身份证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
id_card_side	是	string	front - 身份证含照片的一面 back - 身份证带国徽的一面		front：身份证含照片的一面；back：身份证带国徽的一面
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_risk	否	string	true - 开启 false - 不开启	false	是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)功能，默认不开启，即：false。可选值:true-开启；false-不开启
detect_photo	否	string	true - 开启 false - 不开启	false	是否检测头像内容，默认不检测。可选值：true-检测头像并返回头像的base64 编码及位置信息
detect_card	否	string	true - 开启 false - 不开启	false	是否检测身份证进行裁剪，默认不检测。可选值：true-检测身份证并返回证照的 base64 编码及位置信息

身份证识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
image_status	是	string	normal-识别正常 reversed_side-未摆正身份证 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 over_exposure-身份证关键字段反光或过曝 unknown-未知状态
risk_type	否	string	输入参数 detect_risk = true 时，则返回该字段识别身份证类型: normal-正常身份证；copy-复印件；temporary-临时身份证；screen-翻拍；unknow-其他未知情况
edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array(object)	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+location	是	array(object)	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	string	识别结果字符串

身份证识别 返回示例


```
{
  "log_id": 2648325511,
  "direction": 0,
  "image_status": "normal",
  "idcard_type": "normal",
  "edit_tool": "Adobe Photoshop CS3 Windows",
  "words_result": {
    "住址": {
      "location": {
        "left": 267,
        "top": 453,
        "width": 459,
        "height": 99
      },
      "words": "南京市江宁区弘景大道3889号"
    },
    "公民身份号码": {
      "location": {
```

```
            "left": 443,
            "top": 681,
            "width": 589,
            "height": 45
        },
        "words": "330881199904173914"
    },
    "出生": {
        "location": {
            "left": 270,
            "top": 355,
            "width": 357,
            "height": 45
        },
        "words": "19990417"
    },
    "姓名": {
        "location": {
            "left": 267,
            "top": 176,
            "width": 152,
            "height": 50
        },
        "words": "伍云龙"
    },
    "性别": {
        "location": {
            "left": 269,
            "top": 262,
            "width": 33,
            "height": 52
        },
        "words": "男"
    },
    "民族": {
        "location": {
            "left": 492,
            "top": 279,
            "width": 30,
            "height": 37
        },
        "words": "汉"
    }
},
"words_result_num": 6
}
```


银行卡识别

识别银行卡并返回卡号和发卡行。

```
 public void BankcardDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用银行卡识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.Bankcard(image);
Console.WriteLine(result);
}
```

银行卡识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据

银行卡识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	请求标识码，随机数，唯一。
result	object	是	返回结果
+bank_card_number	string	是	银行卡卡号
+bank_name	string	是	银行名，不能识别时为空
+bank_card_type	number	是	银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡

银行卡识别 返回示例

```
{
  "log_id": 1447188951,
  "result":{
     "bank_card_number": "622500000000000",
     "bank_name": "招商银行",
     "bank_card_type": 1
  }
}
```

驾驶证识别

对机动车驾驶证所有关键字段进行识别

```
 public void DrivingLicenseDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用驾驶证识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.DrivingLicense(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{
   {"detect_direction", "true"}
};
// 带参数调用驾驶证识别
result = client.DrivingLicense(image, options);
Console.WriteLine(result);
}
```

驾驶证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

驾驶证识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
+words	否	string	识别结果字符串

驾驶证识别 返回示例


```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "证号": {
    "words": "3208231999053090"
   },
   "有效期限": {
    "words": "6年"
   },
   "准驾车型": {
    "words": "B2"
   },
   "有效起始日期": {
    "words": "20101125"
   },
   "住址": {
    "words": "江苏省南通市海门镇秀山新城"
   },
   "姓名": {
    "words": "小欧欧"
   },
   "国籍": {
    "words": "中国"
   },
   "出生日期": {
    "words": "19990530"
   },
   "性别": {
    "words": "男"
   },
   "初次领证日期": {
    "words": "20100125"
   }
  }
 }
}
```


行驶证识别

对机动车行驶证所有关键字段进行识别

```
 public void VehicleLicenseDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用行驶证识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.VehicleLicense(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{
    {"detect_direction", "true"},
    {"accuracy", "normal"}
};
// 带参数调用行驶证识别
result = client.VehicleLicense(image, options);
Console.WriteLine(result);
}
```

行驶证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
detect_direction	否	string	true/false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vehicle_license_side	否	string	front/back	front	- **front**：识别行驶证主页 - **back**：识别行驶证副页
unified	否	string	true/false	false	- **false**：不进行归一化处理 - **true**：对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出

行驶证识别 返回数据参数详情

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
+words	否	string	识别结果字符串

行驶证识别 返回示例

```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "品牌型号": {
    "words": "保时捷GT37182RUCRE"
   },
   "发证日期": {
    "words": "20160104"
   },
   "使用性质": {
    "words": "非营运"
   },
   "发动机号码": {
    "words": "20832"
   },
   "号牌号码": {
    "words": "苏A001"
   },
   "所有人": {
    "words": "圆圆"
   },
   "住址": {
    "words": "南京市江宁区弘景大道"
   },
   "注册日期": {
    "words": "20160104"
   },
   "车辆识别代号": {
    "words": "HCE58"
   },
   "车辆类型": {
    "words": "小型轿车"
   }
  }
 }
}
```


车牌识别

识别机动车车牌，并返回签发地和号牌。

```
 public void LicensePlateDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用车牌识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.LicensePlate(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{
    {"multi_detect", "true"}
};
// 带参数调用车牌识别
result = client.LicensePlate(image, options);
Console.WriteLine(result);
}
```

车牌识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
multi_detect	否	string	true false	false	是否检测多张车牌，默认为false，当置为true的时候可以对一张图片内的多张车牌进行识别

车牌识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
Color	string	是	车牌颜色
number	string	是	车牌号码

车牌识别 返回示例

```
{
  "log_id": 3583925545,
  "words_result":{
     "color": "blue",
     "number": "苏HS7766"
  }
}
```

营业执照识别

识别营业执照，并返回关键字段的值，包括单位名称、法人、地址、有效期、证件编号、社会信用代码等。

```
 public void BusinessLicenseDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用营业执照识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.BusinessLicense(image);
Console.WriteLine(result);
}
```

营业执照识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据

营业执照识别 返回数据参数详情

参数	是否必须	类型	说明
log_id	是	number	请求标识码，随机数，唯一。
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	否	string	识别结果字符串

营业执照识别 返回示例

```
{
  "log_id": 490058765,
  "words_result": {
    "单位名称": {
      "location": {
        "left": 500,
        "top": 479,
        "width": 618,
        "height": 54
      },
      "words": "袁氏财团有限公司"
    },
    "法人": {
      "location": {
        "left": 938,
        "top": 557,
        "width": 94,
        "height": 46
      },
      "words": "袁运筹"
    },
    "地址": {
      "location": {
        "left": 503,
        "top": 644,
        "width": 574,
        "height": 57
      },
      "words": "江苏省南京市中山东路19号"
    },
    "有效期": {
      "location": {
        "left": 779,
        "top": 1108,
        "width": 271,
        "height": 49
      },
      "words": "2015年02月12日"
    },
    "证件编号": {
      "location": {
        "left": 1219,
        "top": 357,
        "width": 466,
        "height": 39
      },
      "words": "苏餐证字(2019)第666602666661号"
    },
    "社会信用代码": {
      "location": {
        "left": 0,
        "top": 0,
        "width": 0,
        "height": 0
      },
      "words": "无"
    }
  },
  "words_result_num": 6
}
```

通用票据识别

用户向服务请求识别医疗票据、发票、的士票、保险保单等票据类图片中的所有文字，并返回文字在图中的位置信息。

```
 public void ReceiptDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用通用票据识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.Receipt(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{
    {"recognize_granularity", "big"},
    {"probability", "true"},
    {"accuracy", "normal"},
    {"detect_direction", "true"}
};
// 带参数调用通用票据识别
result = client.Receipt(image, options);
Console.WriteLine(result);
}
```

通用票据识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
recognize_granularity	否	string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true false		是否返回识别结果中每一行的置信度
accuracy	否	string	normal - 使用快速服务		normal 使用快速服务，1200ms左右时延；缺省或其它值使用高精度服务，1600ms左右时延
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

通用票据识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array()	单字符结果，recognize_granularity=small时存在
location	是	array()	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位定位位置的长方形的宽度
height	是	number	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值

通用票据识别 返回示例

```
{
  "log_id": 2661573626,
  "words_result": [
    {
      "location": {
        "left": 10,
        "top": 3,
        "width": 121,
        "height": 24
      },
      "words": "姓名:小明明",
      "chars": [
        {
          "location": {
            "left": 16,
            "top": 6,
            "width": 17,
            "height": 20
          },
          "char": "姓"
        }
        ...
      ]
    },
    {
      "location": {
        "left": 212,
        "top": 3,
        "width": 738,
        "height": 24
      },
      "words": "卡号/病案号:105353990标本编号:150139071送检科室:血液透析门诊病房",
      "chars": [
        {
          "location": {
            "left": 218,
            "top": 6,
            "width": 18,
            "height": 21
          },
          "char": "卡"
        }
        ...
      ]
    }
  ],
  "words_result_num": 2
}
```


自定义模板文字识别

自定义模板文字识别，是针对百度官方没有推出相应的模板，但是当用户需要对某一类卡证/票据（如房产证、军官证、火车票等）进行结构化的提取内容时，可以使用该产品快速制作模板，进行识别。

```
 public void CustomDemo() {
 var image = File.ReadAllBytes("图片文件路径");
 var templateSign = "Nsdax2424asaAS791823112";

 // 调用自定义模板文字识别，可能会抛出网络等异常，请使用try/catch捕获
 var result = client.Custom(image, templateSign);
 Console.WriteLine(result);
}
```

自定义模板文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据
templateSign	否	String	您在自定义文字识别平台制作的模板的ID
classifierId	否	string	分类器Id。这个参数和templateSign至少存在一个，优先使用templateSign。存在templateSign时，表示使用指定模板；如果没有templateSign而有classifierId，表示使用分类器去判断使用哪个模板

自定义模板文字识别 返回数据参数详情

字段	是否必选	类型	说明
error_code	number	number	0代表成功，如果有错误码返回可以参考下方错误码列表排查问题
error_msg	是	string	具体的失败信息，可以参考下方错误码列表排查问题
data	jsonObject	识别返回的结果	

自定义模板文字识别 返回示例


```
{
  "isStructured": true,
  "ret": [
    {
      "charset": [
        {
          "rect": {
            "top": 183,
            "left": 72,
            "width": 14,
            "height": 28
          },
          "word": "5"
        },
        {
          "rect": {
            "top": 183,
            "left": 90,
            "width": 14,
            "height": 28
          },
          "word": "4"
        },
        {
          "rect": {
            "top": 183,
            "left": 103,
            "width": 15,
            "height": 28
```

```
            },
            "word": "."
        },
        {
            "rect": {
                "top": 183,
                "left": 116,
                "width": 14,
                "height": 28
            },
            "word": "5"
        },
        {
            "rect": {
                "top": 183,
                "left": 133,
                "width": 19,
                "height": 28
            },
            "word": "元"
        }
    ],
    "word_name": "票价",
    "word": "54.5元"
},
{
    "charset": [
        {
            "rect": {
                "top": 144,
                "left": 35,
                "width": 14,
                "height": 28
            },
            "word": "2"
        },
        {
            "rect": {
                "top": 144,
                "left": 53,
                "width": 14,
                "height": 28
            },
            "word": "0"
        },
        {
            "rect": {
                "top": 144,
                "left": 79,
                "width": 14,
                "height": 28
            },
            "word": "1"
        },
        {
            "rect": {
                "top": 144,
                "left": 97,
                "width": 14,
                "height": 28
            },
            "word": "7"
        }
```

```
        ]
    ]
}
```

增值税发票识别

```
 var APP_ID = "你的 App ID";
var API_KEY = "你的 Api Key";
var SECRET_KEY = "你的 Secret Key";

var client = new Baidu.Aip.Ocr.Ocr(API_KEY, SECRET_KEY);
public void Demo() {
    // 参数为本地图片
      var image = File.ReadAllBytes("图片文件路径");
    result = client.VatInvoice(image, options);
    Console.WriteLine(result);
      // 参数为url
    result = client.VatInvoiceUrl("http:/test.jpg", options);
      Console.WriteLine(result);
      // 参数为pdf
      var pdf = File.ReadAllBytes("pdf文件");
    result = client.VatInvoicePdf(pdf, options);
      Console.WriteLine(result);
   }
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url或者pdf文件
type	否	String	可选参数，进行识别的增值税发票类型，默认为 normal，可缺省normal：可识别增值税普票、专票、电子发票roll：可识别增值税卷票

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	是	string	发票种类
InvoiceTypeOrg	是	string	发票名称
InvoiceCode	是	string	发票代码
InvoiceNum	是	string	发票号码
MachineNum	是	string	机打号码。仅增值税卷票含有此参数
MachineCode	是	string	机器编号。仅增值税卷票含有此参数
CheckCode	否	string	校验码。增值税专票无此参数
InvoiceDate	是	string	开票日期
PurchaserName	是	string	购方名称
PurchaserRegisterNum	是	string	购方纳税人识别号
PurchaserAddress	是	string	购方地址及电话
PurchaserBank	是	string	购方开户行及账号
Password	是	string	密码区

Province	是	string	省
City	是	string	市
SheetNum	是	string	联次
Agent	是	string	是否代开
CommodityName	是	array[]	货物名称
- row	是	uint32	行号
- word	是	string	内容
CommodityType	是	array[]	规格型号
- row	是	uint32	行号
- word	是	string	内容
CommodityUnit	是	array[]	单位
- row	是	uint32	行号
- word	是	string	内容
CommodityNum	是	array[]	数量
- row	是	uint32	行号
- word	是	string	内容
CommodityPrice	是	array[]	单价
- row	是	uint32	行号
- word	是	string	内容
CommodityAmount	是	array[]	金额
- row	是	uint32	行号
- word	是	string	内容
CommodityTaxRate	是	array[]	税率
- row	是	uint32	行号
- word	是	string	内容
CommodityTax	是	array[]	税额
- row	是	uint32	行号
- word	是	string	内容
SellerName	是	string	销售方名称
SellerRegisterNum	是	string	销售方纳税人识别号
SellerAddress	是	string	销售方地址及电话
SellerBank	是	string	销售方开户行及账号
TotalAmount	是	uint32	合计金额
TotalTax	是	uint32	合计税额
AmountInWords	是	string	价税合计(大写)
AmountInFiguers	是	uint32	价税合计(小写)
Payee	是	string	收款人
Checker	是	string	复核
NoteDrawer	是	string	开票人

Remarks	是	string	备注

返回示例


```
{
 "log_id": "5425496231209218858",
"words_result_num": 29,
"words_result": {
 "InvoiceNum": "14641426",
 "SellerName": "上海易火广告传媒有限公司",
 "CommodityTaxRate": [
  {
   "word": "6%",
   "row": "1"
  }
 ],
 "SellerBank": "中国银行南翔支行446863841354",
 "Checker": ":沈园园",
 "TotalAmount": "94339.62",
 "CommodityAmount": [
  {
   "word": "94339.62",
   "row": "1"
  }
 ],
 "InvoiceDate": "2016年06月02日",
 "CommodityTax": [
  {
   "word": "5660.38",
   "row": "1"
  }
 ],
 "PurchaserName": "百度时代网络技术(北京)有限公司",
 "CommodityNum": [
  {
   "word": "",
   "row": "1"
  }
 ],
      "Province": "上海",
      "City": "",
      "SheetNum": "第三联",
      "Agent": "否",
 "PurchaserBank": "招商银行北京分行大屯路支行8661820285100030",
 "Remarks": "告传",
 "Password": "074/45781873408>/6>8>65*887676033/51+<5415>9/32--852>1+29<65>641-
5>66<500>87/*-34<943359034>716905113*4242>",
 "SellerAddress": ":嘉定区胜辛南路500号15幢1161室55033753",
 "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二属A2010-59108001",
 "InvoiceCode": "3100153130",
 "CommodityUnit": [
  {
   "word": "",
   "row": "1"
  }
 ],
 "Payee": ":徐蓉",
 "PurchaserRegisterNum": "110108787751579",
 "CommodityPrice": [
  {
   "word": "",
   "row": "1"
  }
```

```
  ],
  "NoteDrawer": "沈园园",
  "AmountInWords": "壹拾万圆整",
  "AmountInFiguers": "100000.00",
  "TotalTax": "5660.38",
  "InvoiceType": "专用发票",
  "SellerRegisterNum": "913101140659591751",
  "CommodityName": [
   {
    "word": "信息服务费",
    "row": "1"
   }
  ],
  "CommodityType": [
   {
    "word": "",
    "row": "1"
   }
  ]
 }
}
```


出租车票识别

```
 public void CustomDemo() {
var image = File.ReadAllBytes("图片文件路径");
var templateSign = "Nsdax2424asaAS791823112";
// 如果有可选参数
   var options = new Dictionary<string, object>{ };
// 参数为本地图片
   var image = File.ReadAllBytes("图片文件路径");
result = client.taxiReceipt(image, options);
Console.WriteLine(result);
   // 参数为url
result = client.taxiReceiptUrl("http:/test.jpg", options);
   Console.WriteLine(result);
}
**请求参数详情**
```

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
words_result_num	uint32	是	识别结果数，表示words_result的元素个数
InvoiceCode	string	是	发票代号
InvoiceNum	string	是	发票号码
TaxiNum	string	是	车牌号
Date	string	是	日期
Time	string	是	上下车时间
Fare	string	是	总金额
FuelOilSurcharge	string	是	燃油附加费
CallServiceSurcharge	string	是	叫车服务费
Province	string	是	省
City	string	是	市
PricePerkm	string	是	单价
Distance	string	是	里程

返回示例

```
{
"log_id":2034039896,
"words_result_num":6,
"words_result":
{
 "Date":"2017-11-26",
 "Fare":"¥153.30元",
 "InvoiceCode":"111001681009",
 "InvoiceNum":"90769610",
 "TaxiNum":"BV2062",
 "Time":"20:42-21:07",
 "FuelOilSurcharge": "¥0.00",
 "CallServiceSurcharge": "¥0.00",
      "Province": "浙江省",
      "City": "杭州市",
      "PricePerkm": "2.50元/KM",
      "Distance": "4.5KM"
}
}
```

VIN码识别

```
    public void CustomDemo() {
    // 如果有可选参数
      var options = new Dictionary<string, object>{ };
    // 参数为本地图片
      var image = File.ReadAllBytes("图片文件路径");
    result = client.vinCode(image, options);
    Console.WriteLine(result);
      // 参数为url
    result = client.vinCodeUrl("http:/test.jpg", options);
      Console.WriteLine(result);
  }
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
location	是	object{}	识别结果
words	是	string	VIN码识别结果
words_result_num	是	int	识别结果数，表示words_result的元素个数

返回示例

```
{
"log_id": 246589877,
"words_result": [
 {
  "location": {
   "left": 124,
   "top": 11,
   "width": 58,
   "height": 359
  },
  "words": "LFV2A11K8D4010942"
 }
],
"words_result_num": 1
}
```

火车票识别

```
    public void CustomDemo() {
        var options = new Dictionary<string, object>{ };
    // 参数为本地图片
        var image = File.ReadAllBytes("图片文件路径");
    result = client.trainTicket(image, options);
    Console.WriteLine(result);
        // 参数为url
    result = client.trainTicketUrl("http:/test.jpg", options);
        Console.WriteLine(result);
    }
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
ticket_num	string	是	车票号
starting_station	string	是	始发站
train_num	string	是	车次号
destination_station	string	是	到达站
date	string	是	出发日期
ticket_rates	string	是	车票金额
seat_category	string	是	席别
name	string	是	乘客姓名
id_num	string	是	身份证号
serial_number	string	是	序列号
sales_station	string	是	售站
time	string	是	时间
seat_num	string	是	座位号

返回示例


```
{
"log_id": "12317512659",
"direction": 1,
"words_result_num": 13,
"words_result": {
 "id_num": "2302051998****156X",
 "name": "裴一丽",
 "ticket_rates": "￥54.5元",
 "destination_station": "天津站",
 "seat_category": "二等座",
 "sales_station": "北京南",
 "ticket_num": "F05706",
 "seat_num": "02车03C号",
 "time": "09:36",
 "date": "2019年04月03日",
 "serial_number": "10010300067846",
 "train_num": "C255",
 "starting_station": "北京南站"
}
}
```


数字识别

```
    public void CustomDemo() {
        var options = new Dictionary<string, object>{ };
    // 参数为本地图片
        var image = File.ReadAllBytes("图片文件路径");
    result = client.numbers(image, options);
    Console.WriteLine(result);

    }
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据
recognize_granularity	false	string	big、small
detect_direction	false	string	true、false

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
     {
        "location": {
           "left": 56,
           "top": 0,
           "width": 21,
           "height": 210
        },
        "words": "3"
     }
  ],
  "words_result_num": 1
}
```

印章识别

检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别

```
 public void SealDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 印章识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.Seal(image);
Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
result_num	是	uint32	识别结果数，表示results的元素个数
result	是	array[]	定位结果数组
+location	是	object{}	位置数组（坐标0点为左上角）
++left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	是	uint32	表示定位位置的长方形的宽度
++height	是	uint32	表示定位位置的长方形的高度
+probability	是	float	每一个识别结果的置信度值
+type	是	string	印章的类别，共有circle（圆章），ellipse（椭圆章），rectangle（方章）三种
+major	是	object{}	主字段内容
++words	是	string	主字段识别内容，即章内上环弯曲文字结果
++probability	是	float	主字段识别内容的置信度
+minor	是	array[]	其他字段内容，即除主字段外的文字识别内容均放置于该参数中返回，若章内不存在其他字段文字，则该参数为空
++words	是	string	其他字段识别内容
++probability	是	float	其他字段识别内容的置信度

返回示例

```
{
"result": [
 {
  "major": {
   "probability": 0.99759155511856,
   "words": "峨眉山旅游股份有限公司成都峨眉山雪芽大酒店分公司"
  },
  "minor": [
   {
    "probability": 0.99994027614594,
    "words": "前厅部"
   }
  ],
  "probability": 0.9936261177063,
  "location": {
   "top": 594,
   "left": 918,
   "width": 150,
   "height": 142
  },
  "type": "circle"
 }
],
"log_id": "1349006147834609664",
"result_num": 1
}
```


网络图片文字识别（含位置版）

支持识别艺术字体或背景复杂的文字内容，除文字信息外，还可返回每行文字的位置信息、行置信度，以及单字符内容和位置等。

```
 public void WebimageLocDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 网络图片文字识别（含位置版），可能会抛出网络等异常，请使用try/catch捕获
var result = client.WebimageLoc(image);
Console.WriteLine(result);
// 文件url
var url = "http://host/test.jpeg"
result = client.WebimageLocUrl(url);
// 如果有可选参数
var options = new Dictionary<string, object>{}
// 带参数调用网络图片文字识别（含位置版）
result = client.WebimageLoc(image, options);
Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，像素乘积不超过2048*2048（1024*1024以内图像处理效果最佳）。**注意：图片的base64编码是不包含图片头的，如（）**
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
poly_location	false	string	true/false	是否返回文字所在区域的外接四边形的4个点坐标信息。默认为false
recognize_granularity	false	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
direction	false	int32	图像方向，当detect_direction=true时存在。检测到的图像朝向： 0 ：正向； 1：逆时针旋转90度； 2：逆时针旋转180度； 3：逆时针旋转270度
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	整行的识别结果
+location	true	object	整行的矩形框坐标。位置数组（坐标0点为左上角）
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	true	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	true	array[]	poly_location=true时存在。文字所在区域的外接矩形的4个点坐标信息
++x	true	uint32	水平坐标（坐标0点为左上角）
++y	true	uint32	垂直坐标（坐标0点为左上角）
+chars	false	array[]	单字符结果，recognize_granularity=small时存在
++char	false	string	单字符识别结果
++location	false	object	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示定位定位位置的长方形的高度

返回示例

请参考通用文字识别（高精度含位置版）返回示例

仪器仪表盘读数识别

适用于不同品牌、不同型号的仪器仪表盘读数识别，广泛适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号，支持液晶屏、字轮表等表型。

```
 public void MeterDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 仪器仪表盘读数识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.Meter(image);
Console.WriteLine(result);
// 文件url
var url = "http://host/test.jpeg"
result = client.MeterUrl(url);
// 如果有可选参数
var options = new Dictionary<string, object>{}
// 带参数调用仪器仪表盘读数识别
result = client.Meter(image, options);
Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px。支持jpg/jpeg/png/bmp格式.**注意：图片的base64编码是不包含图片头的，如（）**
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
poly_location	false	string	true/false	位置信息返回形式，默认：false false：只给出识别结果所在长方形位置信息 true：除了默认的识别文字所在长方形的位置信息，还会给出文字所在区域的最小外接旋转矩形的4个点坐标信息

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	识别结果字符串
+location	true	array[]	识别结果所在长方形位置信息
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	false	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	false	array[]	poly_location=true时存在。文字所在区域的外接四边形的4个点坐标信息

返回示例

请参考通用文字识别（高精度含位置版）返回示例

试卷分析与识别

可对文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景

```
 public void DocAnalysisDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 试卷分析与识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.DocAnalysis(image);
Console.WriteLine(result);
// 文件url
var url = "http://host/test.jpeg"
result = client.DocAnalysisUrl(url);
// 如果有可选参数
var options = new Dictionary<string, object>{}
// 带参数调用试卷分析与识别
result = client.DocAnalysis(image, options);
Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少64px，最长边最大4096px。**注意：图片的base64编码是不包含图片头的，如（）**
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
language_type	false	string	CHN_ENG/ENG	识别语言类型，默认为CHN_ENG 可选值包括： =CHN_ENG：中英文 =ENG：英文
result_type	false	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
words_type	false	string	handwring_only/handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	false	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
img_direction	false	int32	detect_direction=true时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	true	uint32	识别结果数，表示results的元素个数
results	true	array[]	识别结果数组
+words_type	true	string	文字属性（手写、印刷）,handwriting 手写，print 印刷
+words	true	array[]	整行的识别结果数组。
++line_probability	false	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++average	false	float	行置信度
+++min	false	float	整行中单字的最低置信度
++word	true	float	整行的识别结果
++words_location	true	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	true	uint32	表示定位定位位置的长方形的宽度
+++height	true	uint32	表示位置的长方形的高度
+chars	false	array[]	result_type=small时返回。单字符结果数组。
++char	false	string	result_type=small时返回。每个单字的内容。
++chars_location	false	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示位置的长方形的高度
layouts_num	false	uint32	版面分析结果数，表示layout的元素个数
layouts	false	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+layout	false	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title
+layout_location	false	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++x	false	uint32	水平坐标（坐标0点为左上角）
++y	false	uint32	水平坐标（坐标0点为左上角）
+layout_idx	false	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）

返回示例

请参考通用文字识别（标准含位置版）返回示例

手写文字识别

支持对图片中的手写中文、手写数字进行检测和识别，针对不规则的手写字体进行专项优化，识别准确率可达90%以上。

```
 public void HandwritingDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 手写文字识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.Handwriting(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{}
// 带参数调用手写文字识别
result = client.Handwriting(image, options);
Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
recognize_granularity	否	string	big、small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true/false	是否返回识别结果中每一行的置信度，默认为false，不返回置信度
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: true：检测朝向； false：不检测朝向

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	float	当请求参数 **probability=true** 时返回该字段，表示识别结果中每一行的置信度值，包含： - **average**： 行置信度平均值 - **variance**：行置信度方差 - **min**：行置信度最小值
direction	否	int32	图像方向，当detect_direction=true时存在 -1:未定义， 0:正向， 1: 逆时针90度， 2:逆时针180度， 3:逆时针270度

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
        "location": {
            "left": 56,
            "top": 0,
            "width": 21,
            "height": 210
        },
        "words": "3"
    }
  ],
  "words_result_num": 1
}
```

办公文档识别

可对办公类文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景。

```
 public void DocAnalysisOfficeDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 办公文档识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.DocAnalysisOffice(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{}
// 带参数调用办公文档识别
result = client.DocAnalysisOffice(image, options);
Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少64px，最长边最大4096px。**注意：图片的base64编码是不包含图片头的，如（）**
language_type	false	string	CHN_ENG/ENG	识别语言类型，默认为CHN_ENG 可选值包括： =CHN_ENG：中英文 =ENG：英文
result_type	false	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
words_type	false	string	handwring_only/handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	false	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出
erase_seal	false	string	true/false	是否先擦除水印、印章后再识别文档

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
img_direction	false	int32	detect_direction=true时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	true	uint32	识别结果数，表示results的元素个数
results	true	array[]	识别结果数组
+words_type	true	string	文字属性（手写、印刷）,handwriting 手写，print 印刷
+words	true	array[]	整行的识别结果数组。
++line_probability	false	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++average	false	float	行置信度
+++min	false	float	整行中单字的最低置信度
++word	true	float	整行的识别结果
++words_location	true	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	true	uint32	表示定位定位位置的长方形的宽度
+++height	true	uint32	表示位置的长方形的高度
+chars	false	array[]	result_type=small时返回。单字符结果数组。
++char	false	string	result_type=small时返回。每个单字的内容。
++chars_location	false	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示位置的长方形的高度
layouts_num	false	uint32	版面分析结果数，表示layout的元素个数
layouts	false	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+layout	false	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title
+layout_location	false	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++x	false	uint32	水平坐标（坐标0点为左上角）
++y	false	uint32	水平坐标（坐标0点为左上角）
+layout_idx	false	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）

返回示例

请参考通用文字识别（标准含位置版）返回示例

二维码识别

对图片中的二维码、条形码进行检测和识别，返回存储的文字信息

```
 public void QrcodeDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 二维码识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.Qrcode(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{}
// 带参数调用二维码识别
result = client.Qrcode(image, options);
Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
codes_result_num	是	uint32	识别结果数，表示codes_result的元素个数
codes_result	是	array[]	定位和识别结果数组
-type	是	string	识别码类型条码类型包括：9种条形码（UPC_A、UPC_E、EAN_13、EAN_8、CODE_39、CODE_93、CODE_128、ITF、CODABAR），4种二维码（QR_CODE、DATA_MATRIX、AZTEC、PDF_417）
-text	是	string	条形码识别内容,暂时只限于识别中英文结果

返回示例

```
{
  "log_id": 863402790,
  "codes_result": [
    {
      "type": "QR_CODE",
      "text": [
        "中国",
        "北京"
      ]
    }
  ],
  "codes_result_num": 1
}
```

示例2（多个图的情况）：

```
{
  "log_id": 1508509437,
  "codes_result": [
```

```
    {
      "type": "QR_CODE",
      "text": [
        "HTTP://Q8R.HK/YELZ0"
      ]
    },
    {
      "type": "PDF_417",
      "text": [
        "PDF417倢丄TL-30偱撉傒庢傝壜擻偱偡丅"
      ]
    },
    {
      "type": "CODABAR",
      "text": [
        "000800"
      ]
    },
    {
      "type": "CODE_39",
      "text": [
        "1234567890"
      ]
    },
    {
      "type": "AZTEC",
      "text": [
        "www.tec-it.com"
      ]
    },
    {
      "type": "DATA_MATRIX",
      "text": [
        "Wikipedia, the free encyclopedia"
      ]
    },
    {
      "type": "CODE_93",
      "text": [
        "123456789"
      ]
    },
    {
      "type": "CODE_128",
      "text": [
        "50090500019191"
      ]
    },
    {
      "type": "EAN_8",
      "text": [
        "12345670"
      ]
    },
    {
      "type": "EAN_13",
      "text": [
        "6901234567892"
      ]
    },
    {
      "type": "UPC_E",
```

```
            "text": [
                "01234565"
            ]
        }
    ],
    "codes_result_num": 11
}
```

机动车销售发票

支持对机动车销售发票的26个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、机器编号、购买方名称、购买方身份证号码/组织机构代码、车辆类型、厂牌型号、产地、合格证号、发动机号码、车架号码、价税合计、价税合计小写、销货单位名称、电话、纳税人识别号、账号、地址、开户银行、税率、税额、主管税务机关及代码、不含税价格、限乘人数

```
 public void VehicleInvoiceDemo() {
var image = File.ReadAllBytes("图片文件路径");// 调用机动车销售发票
var result = client.VehicleInvoice(image);
var url = "https//www.x.com/sample.jpg"
result = client.VehicleInvoiceUrl(url);
   Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
InvoiceCode	是	string	发票代码/机打代码
InvoiceNum	是	string	发票号码/机打号码
InvoiceDate	是	string	开票日期
MachineCode	是	string	机器编号
Purchaser	是	string	购买方名称
PurchaserCode	是	string	购买方身份证号码/组织机构代码
VehicleType	是	string	车辆类型
ManuModel	是	string	厂牌型号
Origin	是	string	产地
CertificateNum	是	string	合格证号
EngineNum	是	string	发动机号码
VinNum	是	string	车架号码
PriceTax	是	string	价税合计
PriceTaxLow	是	string	价税合计小写
Saler	是	string	销货单位名称
SalerPhone	是	string	销货单位电话
SalerCode	是	string	销货单位纳税人识别号
SalerAccountNum	是	string	销货单位账号
SalerAddress	是	string	销货单位地址
SalerBank	是	string	销货单位开户银行
TaxRate	是	string	税率
Tax	是	string	税额
TaxAuthor	是	string	主管税务机关
TaxAuthorCode	是	string	主管税务机关代码
Price	是	string	不含税价格
LimitPassenger ｜ 是	string	限乘人数	

返回示例

```
{
  "log_id": 283449393728149457,
  "words_result_num": 26,
  "words_result": {
     "InvoiceNum": "00875336",
     "Saler": "深圳市新能源汽车销售有限公司",
     "LimitPassenger": "5",
     "MachineCode": "669745967911",
     "VinNum": "LJLGTCRP1J4007581",
     "TaxRate": "16%",
     "PriceTaxLow": "106100.00",
     "InvoiceDate": "2018-11-29",
     "Price": "￥91465.52",
     "SalerBank": "中国工商银行股份有限公司深圳岭园支行",
     "TaxAuthor": "国家锐务总局深圳市龙岗区税务局第五税务所",
     "ManuModel": "江淮牌HFC7007EYBD6",
     "CertificateNum": "WCH0794J0976801",
     "Purchaser": "苏子潇",
     "VehicleType": "纯电动轿车",
     "InvoiceCode": "14975047560",
     "PriceTax": "壹拾万陆仟壹佰圆整",
     "SalerPhone": "0755-83489306",
     "SalerAddress": "深圳市龙岗区龙岗街道百世国际汽车城",
     "Origin": "安徽省合肥市",
     "EngineNum": "18958407",
     "Tax": "14634.48",
     "PurchaserCode": "5135934475603742222",
     "TaxAuthorCode": "14037589413",
     "SalerAccountNum": "中国工商银行股份有限公司深圳岭园支行",
     "SalerCode": "91449283464582922 78H"
  }
}
```


车辆合格证

支持对车辆合格证的23个关键字段进行结构化识别，包括合格证编号、发证日期、车辆制造企业名、车辆品牌、车辆名称、车辆型号、车架号、车身颜色、发动机型号、发动机号、燃料种类、排量、功率、排放标准、轮胎数、轴距、轴数、转向形式、总质量、整备质量、驾驶室准乘人数、最高设计车速、车辆制造日期

```
 public void VehicleCertificateDemo() {
 var image = File.ReadAllBytes("图片文件路径");// 调用车辆合格证
 var result = client.VehicleCertificate(image);
 var url = "https//www.x.com/sample.jpg"
 result = client.VehicleCertificateUrl(url);
    Console.WriteLine(result);
//  如果有可选参数
 var options = new Dictionary<string, object>{}
 options.Add("language_type", ); options.Add("result_type", ); options.Add("detect_direction", );
options.Add("line_probability", ); options.Add("words_type", );
 result = client.VehicleCertificate(image, options);
    Console.WriteLine(result);
 result = client.VehicleCertificateUrl(url, options);
    Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
CertificationNo	是	string	合格证编号
CertificateDate	是	string	发证日期
Manufacturer	是	string	车辆制造企业名
CarBrand	是	string	车辆品牌
CarName	是	string	车辆名称
CarModel	是	string	车辆型号
VinNo	是	string	车架号
CarColor	是	string	车身颜色
EngineType	是	string	发动机型号
EngineNo	是	string	发动机号
FuelType	是	string	燃料种类
Displacement	是	string	排量
Power	是	string	功率
EmissionStandard	是	string	排放标准
TyreNum	是	string	轮胎数
Wheelbase	是	string	轴距
AxleNum	是	string	轴数
SteeringType	是	string	转向形式
TotalWeight	是	string	总质量
SaddleMass	是	string	整备质量
LimitPassenger	是	string	驾驶室准乘人数
SpeedLimit	是	string	最高设计车速
ManufactureDate	是	string	车辆制造日期

返回示例

```
{
  "log_id": 14814098736243057,
  "words_result_num": 23,
  "words_result": {
     "ManufactureDate": "2016年10月13日",
     "CarColor": "红",
     "LimitPassenger": "2",
     "EngineType": "WP12.460E50",
     "TotalWeight": "25000",
     "Power": "338",
     "CertificationNo": "WEK29JX98645437",
     "FuelType": "汽油",
     "Manufacturer": "陕西汽车集团有限责任公司",
     "SteeringType": "方向盘",
     "Wheelbase": "3175+1350",
     "SpeedLimit": "105",
     "EngineNo": "1418K129178",
     "SaddleMass": "8600",
     "AxleNum": "3",
     "CarModel": "SX4250MC4",
     "VinNo": "LZGJHYD83JX197344",
     "CarBrand": "陕汽牌",
     "EmissionStandard": "GB17691-2005国V,GB3847-2005",
     "Displacement": "11596",
     "CertificateDate": "2018年11月28日",
     "CarName": "牵引汽车",
     "TyreNum": "10"
  }
}
```


户口本识别

支持对户口本内常住人口登记卡的全部 22 个字段进行结构化识别，包括户号、姓名、与户主关系、性别、出生地、民族、出生日期、身份证号、本市县其他住址、曾用名、籍贯、宗教信仰、身高、血型、文化程度、婚姻状况、兵役状况、服务处所、职业、何时由何地迁往本市、何时由何地迁往本址、登记日期

```
 public void HouseholdRegisterDemo() {
var image = File.ReadAllBytes("图片文件路径");// 调用户口本识别
var result = client.HouseholdRegister(image);
var url = "https//www.x.com/sample.jpg"
result = client.HouseholdRegisterUrl(url);
   Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和image二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
Name	是	object{}	姓名
+ words	是	string	所属字段的具体内容，以下各字段均含有此元素
Relationship	是	object{}	户主或与户主关系
Sex	是	object{}	性别
BirthAddress	是	object{}	出生地
Nation	是	object{}	民族
Birthday	是	object{}	生日
CardNo	是	object{}	身份证号
HouseholdNum	是	object{}	户号
FormerName	是	object{}	曾用名
Hometown	是	object{}	籍贯
OtherAddress	是	object{}	本市（县）其他住址
Belief	是	object{}	宗教信仰
Height	是	object{}	身高
BloodType	是	object{}	血型
Education	是	object{}	文化程度
MaritalStatus	是	object{}	婚姻状况
VeteranStatus	是	object{}	兵役状况
WorkAddress	是	object{}	服务处所
Career	是	object{}	职业
WWToCity	是	object{}	何时由何地迁来本市(县)
WWHere	是	object{}	何时由何地迁往本址
Date	是	object{}	登记日期

返回示例

```
{
  "log_id": 1301870459,
  "words_result": {
     "BirthAddress": {
        "words": "河南洛阳市郊区"
     },
     "Birthday": {
        "words": "2016-07-28"
     },
     "CardNo": {
        "words": "410311201607282825"
     },
     "Name": {
        "words": "孙翌晨"
     },
     "Nation": {
        "words": "汉族"
     },
     "Relationship": {
        "words": "户主"
     },
     "Sex": {
        "words": "男"
     },
  },
  "words_result_num": 7
}
```


飞机行程单识别

支持对飞机行程单的24个字段进行结构化识别，包括电子客票号、印刷序号、姓名、始发站、目的站、航班号、日期、时间、票价、身份证号、承运人、民航发展基金、保险费、燃油附加费、其他税费、合计金额、填开日期、订票渠道、客票级别、座位等级、销售单位号、签注、免费行李、验证码。 同时，支持单张行程单上的多航班信息识别。

```
 public void AirTicketDemo() {
 var image = File.ReadAllBytes("图片文件路径");// 调用飞机行程单识别
 var result = client.AirTicket(image);
 var url = "https//www.x.com/sample.jpg"
 result = client.AirTicketUrl(url);
   Console.WriteLine(result);
// 如果有可选参数
 var options = new Dictionary<string, object>{}
 options.Add("multi_detect", );
 result = client.AirTicket(image, options);
   Console.WriteLine(result);
 result = client.AirTicketUrl(url, options);
   Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
multi_detect	否	bool	true/false	控制是否开启多航班信息识别功能，**默认值：false** - **true：开启多航班信息识别功能**，开启后返回结果中对应字段格式将改为数组类型 - **false：** 不开启，仅识别单一航班信息

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
name	是	string	姓名
starting_station	是	string	始发站
destination_station	是	string	目的站
flight	是	string	航班号
date	是	string	日期
ticket_number	是	string	电子客票号码
fare	是	string	票价
dev_fund	是	string	民航发展基金/基建费
fuel_surcharge	是	string	燃油附加费
other_tax	是	string	其他税费
ticket_rates	是	string	合计金额
issued_date	是	string	填开日期
id_num	是	string	身份证号
carrier	是	string	承运人
time	是	string	时间
issued_by	是	string	订票渠道
serial_number	是	string	印刷序号
insurance	是	string	保险费
fare_basis	是	string	客票级别
class	是	string	座位等级
agent_code	是	string	销售单位号
endorsement	是	string	签注
allow	是	string	免费行李
ck	是	string	验证码

返回示例

```
 // 识别单航班信息（multi_detect=false，或参数缺省）
{
    "log_id": 7306800033425229106,
    "direction": 0,
    "words_result_num": 18,
    "words_result": {
        "insurance": "20.00",
        "date": "2019-10-22",
        "allow": "20K",
        "flight": "CA6589",
        "issued_by": "中国国际航空服务有限公司",
        "starting_station": "武汉",
        "fare": "260.00",
        "endorsement": "不得签转改期退转",
        "ticket_rates": "350.00",
        "ck": "5866",
        "serial_number": "51523588676",
        "ticket_number": "7843708871196",
        "fuel_surcharge": "EXEMPT",
        "carrier": "南航",
        "issued_date": "2019-10-30",
        "other_tax": "",
        "fare_basis": "NREOW",
        "id_num": "411201123909020877",
        "destination_station": "合肥",
        "name": "郭达",
        "agent_code": "BJS19197300025",
        "time": "21:25",
        "class": "N",
        "dev_fund": "50.00"
    }
}

// 识别多航班信息（multi_detect=true)
{
    "words_result": {
        "insurance": [
            {
                "word": "XXX"
            }
        ],
        "date": [
            {
                "word": "2019-10-18"
            },
            {
                "word": "2019-10-21"
            }
        ],
        "flight": [
            {
                "word": "CZ3565"
            },
            {
                "word": "CZ3566"
            }
        ],
        "issued_by": [
            {
                "word": "上海携程旅行社有限公司"
            }
        ],
```

```
    ],
    "starting_station": [
      {
        "word": "北京"
      }
    ],
    "fare": [
      {
        "word": "1080.00"
      }
    ],
    "ticket_rates": [
      {
        "word": "1420.00"
      }
    ],
    "serial_number": [
      {
        "word": "45956029770"
      }
    ],
    "ticket_number": [
      {
        "word": "7849648364314"
      }
    ],
    "fuel_surcharge": [
      {
        "word": "240.00"
      }
    ],
    "carrier": [
      {
        "word": "南航"
      },
      {
        "word": "南航"
      }
    ],
    "issued_date": [
      {
        "word": "2019-09-18"
      }
    ],
    "other_tax": [],
    "id_num": [
      {
        "word": "0789654700"
      }
    ],
    "destination_station": [
      {
        "word": "深圳"
      },
      {
        "word": "北京"
      }
    ],
    "name": [
      {
        "word": "姚佳"
      }
    ],
```

```
        "time": [
          {
            "word": "13:55"
          },
          {
            "word": "16:30"
          }
        ],
        "dev_fund": [
          {
            "word": "100.00"
          }
        ]
    },
    "log_id": "1280814270572920832",
    "words_result_num": 18
}
```

通用机打发票

支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别，包括发票类型、发票号码、发票代码、开票日期、合计金额大写、合计金额小写、商品名称、商品单位、商品单价、商品数量、商品金额、机打代码、机打号码、校验码、销售方名称、销售方纳税人识别号、购买方名称、购买方纳税人识别号、合计税额等。

```
 public void InvoiceDemo() {
 var image = File.ReadAllBytes("图片文件路径");// 调用通用机打发票
 var result = client.Invoice(image);
 var url = "https//www.x.com/sample.jpg"
 result = client.InvoiceUrl(url);
   Console.WriteLine(result);
// 如果有可选参数
 var options = new Dictionary<string, object>{}
 options.Add("location", true);
 result = client.Invoice(image, options);
   Console.WriteLine(result);
 result = client.InvoiceUrl(url, options);
   Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	string	true/false	是否输出位置信息，true：输出位置信息，false：不输出位置信息，默认false

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	否	string	发票类型
InvoiceCode	否	string	发票代码
InvoiceNum	否	string	发票号码
InvoiceDate	否	string	开票日期
AmountInFiguers	否	string	合计金额小写
AmountInWords	否	string	合计金额大写
CommodityName	否	string	商品名称
CommodityUnit	否	string	商品单位
CommodityPrice	否	string	商品单价
CommodityNum	否	string	商品数量
CommodityAmount	否	string	商品金额
MachineCode	否	string	机打代码
MachineNum	否	string	机打号码
CheckCode	否	string	校验码
SellerName	否	string	销售方名称
SellerRegisterNum	否	string	销售方纳税人识别号
PurchaserName	否	string	购买方名称
PurchaserRegisterNum	否	string	购买方纳税人识别号
TotalTax	否	string	合计税额
Province	否	string	省
City	否	string	市
Time	否	string	时间
SheetNum	否	string	联次

返回示例

```
{
  "log_id": 4423022131715883558,
  "direction": 0,
  "words_result_num": 22,
  "words_result": {
     "City": "",
     "InvoiceNum": "01445096",
     "SellerName": "百度餐饮店",
     "IndustrSort": "生活服务",
     "Province": "广东省",
     "CommodityAmount": [
        {
           "word": "183.00",
           "row": "1"
        }
     ],
     "InvoiceDate": "2020年07月28日",
     "PurchaserName": "中信建投证券股份有限公司",
     "CommodityNum": [],
     "InvoiceCode": "144001901511",
     "CommodityUnit": [],
     "SheetNum": "",
     "PurchaserRegisterNum": "91442300845348OX9",
     "Time": "",
     "CommodityPrice": [],
     "AmountInFiguers": "183.00",
     "AmountInWords": "壹佰捌拾叁元整",
     "CheckCode": "61042119820421061301",
     "TotalTax": "183.00",
     "InvoiceType": "广东通用机打发票",
     "SellerRegisterNum": "61042119820421061301",
     "CommodityName": [
        {
           "word": "餐费",
           "row": "1"
        }
     ]
  }
}
```


护照识别

支持对中国大陆护照个人资料页所有15个字段进行结构化识别，包括国家码、护照号、姓名、姓名拼音、性别、出生地点、出生日期、签发地点（不支持境外签发地）、签发日期、有效期、签发机关、护照类型、国籍、MRZCode1、MRZCode2。

```
 public void PassportDemo() {
var image = File.ReadAllBytes("图片文件路径");// 调用护照识别
var result = client.Passport(image);
var url = "https//www.x.com/sample.jpg"
result = client.PassportUrl(url);
   Console.WriteLine(result);
}
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
-location	是	uint32	水平坐标
-words	是	string	识别内容

返回示例

```
 护照OCR:
耗时:00:00:00.7168000:
{
    "log_id": 7377468409496932872,
    "words_result_num": 14,
    "words_result": {
        "国家码": {
            "location": {
                "width": 59,
                "top": 200,
                "left": 762,
                "height": 26
            },
            "words": "CHN"
        },
        "护照签发地点": {
            "location": {
                "width": 236,
                "top": 505,
                "left": 558,
                "height": 43
            },
            "words": "山东/SHANDONG"
        },
        "MRZCode2": {
            "location": {
                "width": 1252,
                "top": 797,
                "left": 145,
                "height": 88
            },
            "words": "E898657303CHNSDM0702<<<<<<<<<<<<<<"
        },
        "有效期至": {
            "location": {
```

```
            "width": 287,
            "top": 528,
            "left": 955,
            "height": 46
        },
        "words": "20261004"
    },
    "签发机关": {
        "location": {
            "width": 271,
            "top": 583,
            "left": 552,
            "height": 42
        },
        "words": "出入境管理局"
    },
    "MRZCode1": {
        "location": {
            "width": 1201,
            "top": 781,
            "left": 162,
            "height": 45
        },
        "words": "PONSUN<<JIAJIA<<<<<<<<<<<<<<<<<<<<<<<<<<"
    },
    "护照号码": {
        "location": {
            "width": 237,
            "top": 215,
            "left": 994,
            "height": 48
        },
        "words": "E89865730"
    },
    "签发日期": {
        "location": {
            "width": 279,
            "top": 453,
            "left": 955,
            "height": 46
        },
        "words": "20161005"
    },
    "出生地点": {
        "location": {
            "width": 216,
            "top": 429,
            "left": 564,
            "height": 43
        },
        "words": "山东/SHANDONG"
    },
    "姓名": {
        "location": {
            "width": 159,
            "top": 247,
            "left": 581,
            "height": 34
        },
        "words": "孙嘉佳"
    },
    "姓名拼音": {
        "location": {
```

```
            "location": {
                "width": 229,
                "top": 279,
                "left": 578,
                "height": 41
            },
            "words": "SUN,JIAJIA"
        },
        "国籍": {
            "location": {
                "width": 209,
                "top": 366,
                "left": 695,
                "height": 42
            },
            "words": "中国/CHINESE"
        },
        "生日": {
            "location": {
                "width": 202,
                "top": 382,
                "left": 950,
                "height": 39
            },
            "words": "19950723"
        },
        "性别": {
            "location": {
                "width": 73,
                "top": 357,
                "left": 570,
                "height": 34
            },
            "words": "男/M"
        }
    }
}
```


图文转换器（接口板）

代码调用示例

```
 // 图片
var imagePath = "图片路径";
var image = File.ReadAllBytes(imagePath);
// url
var url = "url地址";
// pdfPath = "pdf路径";
var pdf = File.ReadAllBytes(pdfPath);

Dictionary<string, object> options = new Dictionary<string, object> { };

var result = _ocr.DocConvertRequestV1(image, options);
var result = _ocr.DocConvertRequestV1Url(url, options);
var result = _ocr.DocConvertRequestV1Pdf(pdf, options);

// 获取最终结果
var task_id = "任务ID";
var result = _ocr.DocConvertResultV1(task_id);
```

接口详情 可参考API文档：图文转换器

表格文字识别同步接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别同步接口。

表格文字识别

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别。

表格识别结果

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别结果。

表格识别轮询接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别轮询接口。

FAQ

1. throw exception "fail to fetch token: 基础连接已关闭"

- 检查网络连接
- 检查网络是否有代理

2. SSL报错 "The authentication or decryption has failed"

可能由于网络代理等原因导致证书不正确，属于常见的网络问题，可以参考这个答案

3. 调用接口，等待一段时间后出现超时

可以设置Timeout参数。

4. 在.Net Core下运行报错 'GBK' is not a supported encoding name'

添加System.Text.Encoding.CodePages引用，并注册。 `Encoding.RegisterProvider(CodePagesEncodingProvider.Instance)`

错误信息

错误返回格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

错误码

错误码	错误信息	描述
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天流量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额

100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
282000	internal error	服务器内部错误，如果您使用的是高精度接口，报这个错误码的原因可能是您上传的图片中文字过多，识别超时导致的，建议您对图片进行切割后再识别，其他情况请再次请求， 如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282110	urls not exit	URL参数不存在，请核对URL后再次提交
282111	url format illegal	URL格式非法，请检查url格式是否符合相应接口的入参要求
	url download	url下载超时，请检查url对应的图床/图片无法下载或链路状况不好，您可以重新尝试以下，如果多

282112	url download timeout	url下载超时，请检查url对应的图床/图片无法下载或链路状况不好，您可以重新尝试以下，如果多次尝试后仍不行，建议更换图片地址
282113	url response invalid	URL返回无效参数
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误

Node.js语言

简介

Hi，您好，欢迎使用百度文字识别服务。

本文档主要针对Nodejs开发者，描述百度文字识别接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 在百度智能云控制台内提交工单，咨询问题类型请选择**人工智能服务**；
- 如有疑问，进入AI社区交流：http://ai.baidu.com/forum/topic/list/164

接口能力

接口名称	接口能力简要描述
通用文字识别	识别图片中的文字信息
通用文字识别（高精度版）	更高精度地识别图片中的文字信息
通用文字识别（含位置信息版）	识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（高精度含位置版）	更高精度地识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（含生僻字版）	识别图片中的文字信息（包含对常见字和生僻字的识别）
网络图片文字识别	识别一些网络上背景复杂，特殊字体的文字
网络图片文字识别（含位置版）	识别网络图片中的文字内容（包含文字区域的坐标信息）
身份证识别	识别身份证正反面的文字信息
银行卡识别	识别银行卡的卡号并返回发卡行和卡片性质信息
驾驶证识别	识别机动车驾驶证所有关键字段
行驶证识别	识别机动车行驶证所有关键字段

车牌识别	识别中国大陆各类机动车车牌信息
营业执照识别	对营业执照进行识别
表格文字识别	自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容
通用票据识别	对各类票据图片（医疗票据，保险保单等）进行文字识别，并返回文字在图片中的位置信息
增值税发票识别	对增值税发票进行文字识别，并结构化返回字段信息，支持增值税专票、普票、电子发票
出租车票识别	针对全国各大城市出租车票的发票号码、发票代码、车号、日期、时间、金额等进行结构化识别
VIN码识别	对车辆车架、挡风玻璃上的VIN码进行识别
火车票识别	支持对大陆火车票的车票号、始发站、目的站、车次、日期、票价、席别、姓名进行结构化识别
飞机行程单识别	支持对飞机行程单的24个字段进行结构化识别
二维码识别	对图片中的二维码、条形码进行检测和识别，返回存储的文字信息
数字识别	识别图片中的数字，适用于手机号提取、快递单号提取、充值号码提取等场景
手写文字识别	支持对图片中的手写中文、手写数字进行检测和识别
护照识别	支持对中国大陆护照个人资料页所有15个字段进行结构化识别
户口本识别	对出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码字段进行识别
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果
通用机打发票	支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别
机动车销售发票	支持对机动车销售发票的26个关键字段进行结构化识别
车辆合格证	支持对车辆合格证的23个关键字段进行结构化识别
通用机打发票	对国家/地方税务局发行的横/竖版通用机打发票进行结构化识别
护照识别	支持对中国大陆护照个人资料页所有11个字段进行结构化识别
医疗费用明细识别	支持识别全国医疗费用明细识别
网约车行程单识别	对国家/地方税务局发行的横/对各大主要服务商的网约车行程单进行结构化识别
磅单识别	结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单
仪器仪表盘读数识别	适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号
自定义模板文字识别	针对固定版式卡证票据提供的 OCR 定制化产品，可由用户自助创建识别模板和分类器，实现对任意版式卡证票据进行自动分类并结构化输出识别结果
医疗费用明细识别	支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含项目类型、项目名称、单价、数量、规格、金额
办公文档识别	可对办公类文档的版面进行分析，输出图、表、标题、文本、目录、栏、页眉、页脚、页码和脚注的位置，并输出分版块内容的OCR识别结果
印章识别	检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别
机动车登记证书识别	对机动车登记证书的编号、机动车所有人、登记机关、车辆类型、发证机关章等15个关键字段进行结构化识别
智能财务票据	

识别	对增值税发票、卷票、火车票、出租车票、机票行程单等13类票据混贴的图片进行切分识别
增值税发票验真	支持9种增值税发票的真伪及字段信息准确性校验，包括增值税专票、电子专票、普票、电子普票、卷票、通行费增值税电子普票、货运专票、机动车销售发票、二手车销售发票，支持返回票面的全部信息
医疗发票识别	支持识别全国各地门诊/住院发票的 业务流水号、发票号、住院号、门诊号、病例号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付等关键字段。支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数
门脸文字识别	识别图片中的门脸文字信息，自动过滤非门脸文字内容，接口返回门脸名称、描述文字和置信度
车辆证照混贴识别	对机动车行驶证主页及副页、驾驶证主页及副页在同一张图片上的场景进行结构化识别
公式识别	对试卷中的数学公式及题目内容进行识别
图文转换器	可识别图片/PDF文档版面布局，提取文字内容，并转换为保留原文档版式的Word、Excel文档，方便二次编辑和复制，可支持含表格、印章、水印、手写等内容的文档

版本更新记录

上线日期	版本号	更新内容
2021.12.11	4.15.6	新增： 网约车行程单识别，磅单识别，医疗明细识别
2021.05.27	4.15.5	新增： 二维码、行程单、机动车销售发票、车辆合格证、试卷分析与识别、手写、护照、户口本、通用机打（均为商用接口）
2021.01.28	4.15.4	新增 增值税发票、出租车票、VIN码、火车票、数字识别
2020.08.06	4.15.1	新增 文档版面分析与识别，仪器仪表盘读数识别， 网络图片文字识别
2018.4.9	2.2.0	新增表格识别同步接口
2018.1.12	2.1.0	新增自定义文字识别接口
2017.12.21	2.0.0	实现代码重构，接口返回标准promise对象
2017.8.10	1.2.3	增加通用票据识别接口
2017.7.28	1.2.2	新增通用文字识别高精度接口，通用文字识别部分支持url调用
2017.7.14	1.2.1	OCR加入车牌识别接口
2017.6.30	1.2.0	OCR加入表格识别
2017.6.15	1.1.0	加入行驶证，驾驶证
2017.4.13	1.0.0	初版

快速入门

安装通用文字识别 Node SDK

通用文字识别 Node SDK目录结构

```
├─── src
│   ├─── auth                    //授权相关类
│   ├─── http                    //Http通信相关类
│   ├─── client                  //公用类
│   ├─── util                    //工具类
│   └─── const                   //常量类
├─── AipOcr.js                 //通用文字识别交互类
├─── index.js                  //入口文件
└─── package.json                //npm包描述文件
```

支持 node 版本 4.0+

查看源码 Nodejs SDK代码已开源，您可以查看代码、或者在License范围内修改和编译SDK以适配您的环境。 github链接：https://github.com/Baidu-AIP/nodejs-sdk

直接使用node开发包步骤如下：

1.在官方网站下载node SDK压缩包。

2.将下载的`aip-node-sdk-version.zip`解压后，复制到工程文件夹中。

3.进入目录，运行npm install安装sdk依赖库。

4.把目录当做模块依赖。

其中，`version`为版本号，添加完成后，用户就可以在工程中使用通用文字识别 Node SDK。

直接使用npm安装依赖：

`npm install baidu-aip-sdk`

新建AipOcrClient

AipOcrClient是Optical Character Recognition的node客户端，为使用Optical Character Recognition的开发人员提供了一系列的交互方法。

用户可以参考如下代码新建一个AipOcrClient：

```
 var AipOcrClient = require("baidu-aip-sdk").ocr;

// 设置APPID/AK/SK
var APP_ID = "你的 App ID";
var API_KEY = "你的 Api Key";
var SECRET_KEY = "你的 Secret Key";

// 新建一个对象，建议只保存一个对象调用服务接口
var client = new AipOcrClient(APP_ID, API_KEY, SECRET_KEY);
```

为了使开发者更灵活的控制请求，模块提供了设置全局参数和全局请求拦截器的方法；本库发送网络请求依赖的是request模块，因此参数格式与request模块的参数相同，更多参数细节您可以参考request官方参数文档。

```
 var HttpClient = require("baidu-aip-sdk").HttpClient;

// 设置request库的一些参数，例如代理服务地址，超时时间等
// request参数请参考 https://github.com/request/request#requestoptions-callback
HttpClient.setRequestOptions({timeout: 5000});

// 也可以设置拦截每次请求（设置拦截后，调用的setRequestOptions设置的参数将不生效），
// 可以按需修改request参数（无论是否修改，必须返回函数调用参数）
// request参数请参考 https://github.com/request/request#requestoptions-callback
HttpClient.setRequestInterceptor(function(requestOptions) {
    // 查看参数
    console.log(requestOptions)
    // 修改参数
    requestOptions.timeout = 5000;
    // 返回参数
    return requestOptions;
});
```

在上面代码中，常量APP_ID在百度智能云控制台中创建，常量API_KEY与SECRET_KEY是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的**应用列表**中查看。

注意：如您以前是百度智能云的老用户，其中API_KEY对应百度智能云的“Access Key ID”，SECRET_KEY对应百度智能云的“Access Key Secret”。

接口说明

通用文字识别

用户向服务请求识别某张图中的所有文字。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用通用文字识别, 图片参数为本地图片
client.generalBasic(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别, 图片参数为本地图片
client.generalBasic(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;

var url = "https//www.x.com/sample.jpg";

// 调用通用文字识别, 图片参数为远程url图片
client.generalBasicUrl(url).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别, 图片参数为远程url图片
client.generalBasicUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

通用文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别 返回示例

```
 {
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青島啤酒"}
   ]
}
```

通用文字识别（高精度版）

用户向服务请求识别某张图中的所有文字，相对于通用文字识别该产品精度更高，但是识别耗时会稍长。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用通用文字识别（高精度版）
client.accurateBasic(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["detect_direction"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（高精度版）
client.accurateBasic(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

通用文字识别（高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别（高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别（高精度版） 返回示例

参考通用文字识别返回示例

通用文字识别（含位置信息版）

用户向服务请求识别某张图中的所有文字，并返回文字在图中的位置信息。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用通用文字识别（含位置信息版），图片参数为本地图片
client.general(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["recognize_granularity"] = "big";
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["vertexes_location"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含位置信息版），图片参数为本地图片
client.general(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;

var url = "https//www.x.com/sample.jpg";

// 调用通用文字识别（含位置信息版），图片参数为远程url图片
client.generalUrl(url).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["recognize_granularity"] = "big";
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["vertexes_location"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含位置信息版），图片参数为远程url图片
client.generalUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

通用文字识别（含位置信息版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
recognize_granularity	否	string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
language_type	否	string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
vertexes_location	否	string	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	string	true false		是否返回识别结果中每一行的置信度
paragraph	否	string	true false		是否输出段落信息

通用文字识别（含位置信息版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置信息版） 返回示例

```
 {
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
     "location": {
        "left": 35,
        "top": 53,
        "width": 193,
        "height": 109
     },
     "words": "感动",
     "chars": [    //recognize_granularity=small时存在
        {
           "location": {
              "left": 56,
              "top": 65,
              "width": 69,
              "height": 88
           },
           "char": "感"
        },
        {
           "location": {
              "left": 140,
              "top": 65,
              "width": 70,
              "height": 88
           },
           "char": "动"
        }
     ]
  }
  ...
]
}
```

通用文字识别（含位置高精度版）

用户向服务请求识别某张图中的所有文字，并返回文字在图片中的坐标信息，相对于通用文字识别（含位置信息版）该产品精度更高，但是识别耗时会稍长。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用通用文字识别（含位置高精度版）
client.accurate(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["recognize_granularity"] = "big";
options["detect_direction"] = "true";
options["vertexes_location"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含位置高精度版）
client.accurate(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

通用文字识别（含位置高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
recognize_granularity	否	string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vertexes_location	否	string	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别（含位置高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置高精度版） 返回示例

```
 {
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
     "location": {
        "left": 35,
        "top": 53,
        "width": 193,
        "height": 109
     },
     "words": "感动",
     "chars": [    //recognize_granularity=small时存在
        {
           "location": {
              "left": 56,
              "top": 65,
              "width": 69,
              "height": 88
           },
           "char": "感"
        },
        {
           "location": {
              "left": 140,
              "top": 65,
              "width": 70,
              "height": 88
           },
           "char": "动"
        }
     ]
  }
  ...
]
}
```

通用文字识别（含生僻字版）

某些场景中，图片中的中文不光有常用字，还包含了生僻字，这时用户需要对该图进行文字识别，应使用通用文字识别（含生僻字版）。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用通用文字识别（含生僻字版），图片参数为本地图片
client.generalEnhance(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含生僻字版），图片参数为本地图片
client.generalEnhance(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;

var url = "https//www.x.com/sample.jpg";

// 调用通用文字识别（含生僻字版），图片参数为远程url图片
client.generalEnhanceUrl(url).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含生僻字版），图片参数为远程url图片
client.generalEnhanceUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

通用文字识别（含生僻字版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	string	true false		是否返回识别结果中每一行的置信度

通用文字识别（含生僻字版） 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含生僻字版） 返回示例

```
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
  [
    {"words": " TSINGTAO"},
    {"words": "青島啤酒"}
  ]
}
```

网络图片文字识别

用户向服务请求识别一些网络上背景复杂，特殊字体的文字。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用网络图片文字识别, 图片参数为本地图片
client.webImage(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["detect_direction"] = "true";
options["detect_language"] = "true";

// 带参数调用网络图片文字识别, 图片参数为本地图片
client.webImage(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;

var url = "https//www.x.com/sample.jpg";

// 调用网络图片文字识别, 图片参数为远程url图片
client.webImageUrl(url).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["detect_direction"] = "true";
options["detect_language"] = "true";

// 带参数调用网络图片文字识别, 图片参数为远程url图片
client.webImageUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

网络图片文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
url	是	string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

网络图片文字识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

网络图片文字识别 返回示例

```
 {
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青岛啤酒"}
   ]
}
```

身份证识别

用户向服务请求识别身份证，身份证识别包括正面和背面。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");
var idCardSide = "back";

// 调用身份证识别
client.idcard(image, idCardSide).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["detect_direction"] = "true";
options["detect_risk"] = "false";

// 带参数调用身份证识别
client.idcard(image, idCardSide, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

身份证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
id_card_side	是	string	front - 身份证含照片的一面 back - 身份证带国徽的一面		front：身份证含照片的一面；back：身份证带国徽的一面
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_risk	否	string	true - 开启 false - 不开启		是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)功能，默认不开启，即：false。可选值:true-开启；false-不开启

身份证识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
image_status	是	string	normal-识别正常 reversed_side-未摆正身份证 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 over_exposure-身份证关键字段反光或过曝 unknown-未知状态
risk_type	否	string	输入参数 detect_risk = true 时，则返回该字段识别身份证类型: normal-正常身份证；copy-复印件；temporary-临时身份证；screen-翻拍；unknow-其他未知情况
edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array(object)	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+location	是	array(object)	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	string	识别结果字符串

身份证识别 返回示例


```
{
  "log_id": 2648325511,
  "direction": 0,
  "image_status": "normal",
  "idcard_type": "normal",
  "edit_tool": "Adobe Photoshop CS3 Windows",
  "words_result": {
    "住址": {
      "location": {
        "left": 267,
        "top": 453,
        "width": 459,
        "height": 99
      },
      "words": "南京市江宁区弘景大道3889号"
    },
    "公民身份号码": {
      "location": {
```

```
            "left": 443,
            "top": 681,
            "width": 589,
            "height": 45
        },
        "words": "330881199904173914"
    },
    "出生": {
        "location": {
            "left": 270,
            "top": 355,
            "width": 357,
            "height": 45
        },
        "words": "19990417"
    },
    "姓名": {
        "location": {
            "left": 267,
            "top": 176,
            "width": 152,
            "height": 50
        },
        "words": "伍云龙"
    },
    "性别": {
        "location": {
            "left": 269,
            "top": 262,
            "width": 33,
            "height": 52
        },
        "words": "男"
    },
    "民族": {
        "location": {
            "left": 492,
            "top": 279,
            "width": 30,
            "height": 37
        },
        "words": "汉"
    }
},
"words_result_num": 6
}
```


银行卡识别

识别银行卡并返回卡号和发卡行。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用银行卡识别
client.bankcard(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

银行卡识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

银行卡识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	请求标识码，随机数，唯一。
result	object	是	返回结果
+bank_card_number	string	是	银行卡卡号
+bank_name	string	是	银行名，不能识别时为空
+bank_card_type	number	是	银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡

银行卡识别 返回示例

```
{
  "log_id": 1447188951,
  "result": {
     "bank_card_number": "622500000000000",
     "bank_name": "招商银行",
     "bank_card_type": 1
  }
}
```

驾驶证识别

对机动车驾驶证所有关键字段进行识别。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用驾驶证识别
client.drivingLicense(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["detect_direction"] = "true";

// 带参数调用驾驶证识别
client.drivingLicense(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

驾驶证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

驾驶证识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
+words	否	string	识别结果字符串

驾驶证识别 返回示例

```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "证号": {
    "words": "3208231999053090"
   },
   "有效期限": {
    "words": "6年"
   },
   "准驾车型": {
    "words": "B2"
   },
   "有效起始日期": {
    "words": "20101125"
   },
   "住址": {
    "words": "江苏省南通市海门镇秀山新城"
   },
   "姓名": {
    "words": "小欧欧"
   },
   "国籍": {
    "words": "中国"
   },
   "出生日期": {
    "words": "19990530"
   },
   "性别": {
    "words": "男"
   },
   "初次领证日期": {
    "words": "20100125"
   }
  }
 }
}
```


行驶证识别

对机动车行驶证所有关键字段进行识别。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用行驶证识别
client.vehicleLicense(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["detect_direction"] = "true";
options["accuracy"] = "normal";

// 带参数调用行驶证识别
client.vehicleLicense(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

行驶证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
detect_direction	否	string	true/false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vehicle_license_side	否	string	front/back	front	- **front**：识别行驶证主页 - **back**：识别行驶证副页
unified	否	string	true/false	false	- **false**：不进行归一化处理 - **true**：对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出

行驶证识别 返回数据参数详情

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
+words	否	string	识别结果字符串

行驶证识别 返回示例

```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "品牌型号": {
    "words": "保时捷GT37182RUCRE"
   },
   "发证日期": {
    "words": "20160104"
   },
   "使用性质": {
    "words": "非营运"
   },
   "发动机号码": {
    "words": "20832"
   },
   "号牌号码": {
    "words": "苏A001"
   },
   "所有人": {
    "words": "圆圆"
   },
   "住址": {
    "words": "南京市江宁区弘景大道"
   },
   "注册日期": {
    "words": "20160104"
   },
   "车辆识别代号": {
    "words": "HCE58"
   },
   "车辆类型": {
    "words": "小型轿车"
   }
  }
 }
}
```


车牌识别

识别机动车车牌，并返回号牌号码和车牌颜色。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用车牌识别
client.licensePlate(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["multi_detect"] = "true";

// 带参数调用车牌识别
client.licensePlate(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

车牌识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
multi_detect	否	string	true false	false	是否检测多张车牌，默认为false，当置为true的时候可以对一张图片内的多张车牌进行识别

车牌识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
Color	string	是	车牌颜色
number	string	是	车牌号码

车牌识别 返回示例

```
{
  "log_id": 3583925545,
  "words_result": {
     "color": "blue",
     "number": "苏HS7766"
  }
}
```

营业执照识别

识别营业执照，并返回关键字段的值，包括单位名称、法人、地址、有效期、证件编号、社会信用代码等。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用营业执照识别
client.businessLicense(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

营业执照识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

营业执照识别 返回数据参数详情

参数	是否必须	类型	说明
log_id	是	number	请求标识码，随机数，唯一。
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	否	string	识别结果字符串

营业执照识别 返回示例

```
{
  "log_id": 490058765,
  "words_result": {
    "单位名称": {
      "location": {
        "left": 500,
        "top": 479,
        "width": 618,
        "height": 54
      },
      "words": "袁氏财团有限公司"
    },
    "法人": {
      "location": {
        "left": 938,
        "top": 557,
        "width": 94,
        "height": 46
      },
      "words": "袁运筹"
    },
    "地址": {
      "location": {
        "left": 503,
        "top": 644,
        "width": 574,
        "height": 57
      },
      "words": "江苏省南京市中山东路19号"
    },
    "有效期": {
      "location": {
        "left": 779,
        "top": 1108,
        "width": 271,
        "height": 49
      },
      "words": "2015年02月12日"
    },
    "证件编号": {
      "location": {
        "left": 1219,
        "top": 357,
        "width": 466,
        "height": 39
      },
      "words": "苏餐证字(2019)第666602666661号"
    },
    "社会信用代码": {
      "location": {
        "left": 0,
        "top": 0,
        "width": 0,
        "height": 0
      },
      "words": "无"
    }
  },
  "words_result_num": 6
}
```

通用票据识别

用户向服务请求识别医疗票据、增值税发票、出租车票、保险保单等票据类图片中的所有文字，并返回文字在图中的位置信息。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用通用票据识别
client.receipt(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["recognize_granularity"] = "big";
options["probability"] = "true";
options["accuracy"] = "normal";
options["detect_direction"] = "true";

// 带参数调用通用票据识别
client.receipt(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

通用票据识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	string			图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
recognize_granularity	否	string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true false		是否返回识别结果中每一行的置信度
accuracy	否	string	normal - 使用快速服务		normal 使用快速服务，1200ms左右时延；缺省或其它值使用高精度服务，1600ms左右时延
detect_direction	否	string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

通用票据识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array()	单字符结果，recognize_granularity=small时存在
location	是	array()	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位定位位置的长方形的宽度
height	是	number	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值

通用票据识别 返回示例

```
{
  "log_id": 2661573626,
  "words_result": [
    {
      "location": {
        "left": 10,
        "top": 3,
        "width": 121,
        "height": 24
      },
      "words": "姓名:小明明",
      "chars": [
        {
          "location": {
            "left": 16,
            "top": 6,
            "width": 17,
            "height": 20
          },
          "char": "姓"
        }
        ...
      ]
    },
    {
      "location": {
        "left": 212,
        "top": 3,
        "width": 738,
        "height": 24
      },
      "words": "卡号/病案号:105353990标本编号:150139071送检科室:血液透析门诊病房",
      "chars": [
        {
          "location": {
            "left": 218,
            "top": 6,
            "width": 18,
            "height": 21
          },
          "char": "卡"
        }
        ...
      ]
    }
  ],
  "words_result_num": 2
}
```


自定义模板文字识别

自定义模板文字识别，是针对百度官方没有推出相应的模版，但是当用户需要对某一类卡证/票据（如房产证、军官证、火车票等）进行结构化的提取内容时，可以使用该产品快速制作模板，进行识别。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");
var templateSign = "Nsdax2424asaAS791823112";

// 调用自定义模板文字识别
client.custom(image, {templateSign}).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

自定义模板文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
templateSign	否	String	您在自定义文字识别平台制作的模板的ID
classifierId	否	string	分类器Id。这个参数和templateSign至少存在一个，优先使用templateSign。存在templateSign时，表示使用指定模板；如果没有templateSign而有classifierId，表示使用分类器去判断使用哪个模板

自定义模板文字识别 返回数据参数详情

字段	是否必选	类型	说明
error_code	number	number	0代表成功，如果有错误码返回可以参考下方错误码列表排查问题
error_msg	是	string	具体的失败信息，可以参考下方错误码列表排查问题
data	jsonObject	识别返回的结果	

自定义模板文字识别 返回示例


```
{
  "isStructured": true,
  "ret": [
    {
      "charset": [
        {
          "rect": {
            "top": 183,
            "left": 72,
            "width": 14,
            "height": 28
          },
          "word": "5"
        },
        {
          "rect": {
            "top": 183,
            "left": 90,
            "width": 14,
            "height": 28
          },
          "word": "4"
        },
        {
```

```
            "rect": {
               "top": 183,
               "left": 103,
               "width": 15,
               "height": 28
            },
            "word": "."
         },
         {
            "rect": {
               "top": 183,
               "left": 116,
               "width": 14,
               "height": 28
            },
            "word": "5"
         },
         {
            "rect": {
               "top": 183,
               "left": 133,
               "width": 19,
               "height": 28
            },
            "word": "元"
         }
      ],
      "word_name": "票价",
      "word": "54.5元"
   },
   {
      "charset": [
         {
            "rect": {
               "top": 144,
               "left": 35,
               "width": 14,
               "height": 28
            },
            "word": "2"
         },
         {
            "rect": {
               "top": 144,
               "left": 53,
               "width": 14,
               "height": 28
            },
            "word": "0"
         },
         {
            "rect": {
               "top": 144,
               "left": 79,
               "width": 14,
               "height": 28
            },
            "word": "1"
         },
         {
            "rect": {
               "top": 144,
```

```
                    "left": 97,
                    "width": 14,
                    "height": 28
                },
                "word": "7"
            }
        ]
    ]
}
```

试卷分析与识别

可对文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景。

```
 var fs = require('fs');
var image = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');

client.docAnalysis(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["result_type"] = "json";

client.docAnalysis(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少64px，最长边最大4096px。注意：图片的base64编码是不包含图片头的，如（）	
language_type	false	string	CHN_ENG / ENG	识别语言类型，默认为CHN_ENG 可选值包括：=CHN_ENG：中英文 =ENG：英文	
result_type	false	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果	
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度	
line_probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false	
words_type	false	string	handwring_only/ handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别	
layout_analysis	false	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出	

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
img_direction	false	int32	detect_direction=true时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	true	uint32	识别结果数，表示results的元素个数
results	true	array[]	识别结果数组
+words_type	true	string	文字属性（手写、印刷）,handwriting 手写，print 印刷
+words	true	array[]	整行的识别结果数组。
++line_probability	false	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++average	false	float	行置信度
+++min	false	float	整行中单字的最低置信度
++word	true	float	整行的识别结果
++words_location	true	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	true	uint32	表示定位定位位置的长方形的宽度
+++height	true	uint32	表示位置的长方形的高度
+chars	false	array[]	result_type=small时返回。单字符结果数组。
++char	false	string	result_type=small时返回。每个单字的内容。
++chars_location	false	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示位置的长方形的高度
layouts_num	false	uint32	版面分析结果数，表示layout的元素个数
layouts	false	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+layout	false	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title
+layout_location	false	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++x	false	uint32	水平坐标（坐标0点为左上角）
++y	false	uint32	水平坐标（坐标0点为左上角）
+layout_idx	false	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）

仪器仪表盘读数识别

适用于不同品牌、不同型号的仪器仪表盘读数识别，广泛适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号，支持液晶屏、字轮表等表型。

```
 var fs = require('fs');
var image = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');

client.meter(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["probability"] = "true";

client.meter(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px。支持jpg/jpeg/png/bmp格式.注意：图片的base64编码是不包含图片头的，如（）	
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false	
poly_location	false	string	true/false	位置信息返回形式，默认：false false：只给出识别结果所在长方形位置信息 true：除了默认的识别文字所在长方形的位置信息，还会给出文字所在区域的最小外接旋转矩形的4个点坐标信息	

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	识别结果字符串
+location	true	array[]	识别结果所在长方形位置信息
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	false	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	false	array[]	poly_location=true时存在。文字所在区域的外接四边形的4个点坐标信息

网络图片文字识别（含位置版）

支持识别艺术字体或背景复杂的文字内容，除文字信息外，还可返回每行文字的位置信息、行置信度，以及单字符内容和位置等。

```
 var fs = require('fs');
var image = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');
client.webimageLoc(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["probability"] = "true";
client.webimageLoc(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

网络图片文字识别（含位置版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，像素乘积不超过2048*2048（1024*1024以内图像处理效果最佳）。注意：图片的base64编码是不包含图片头的，如（）	
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向	
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false	
poly_location	false	string	true/false	是否返回文字所在区域的外接四边形的4个点坐标信息。默认为false	
recognize_granularity	false	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置	

识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
direction	false	int32	图像方向，当detect_direction=true时存在。检测到的图像朝向： 0 ：正向； 1：逆时针旋转90度； 2：逆时针旋转180度； 3：逆时针旋转270度
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	整行的识别结果
+location	true	object	整行的矩形框坐标。位置数组（坐标0点为左上角）
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	true	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	true	array[]	poly_location=true时存在。文字所在区域的外接矩形的4个点坐标信息
++x	true	uint32	水平坐标（坐标0点为左上角）
++y	true	uint32	垂直坐标（坐标0点为左上角）
+chars	false	array[]	单字符结果，recognize_granularity=small时存在
++char	false	string	单字符识别结果
++location	false	object	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示定位定位位置的长方形的高度

增值税发票

```
 //本地文件识别
var fs = require('fs');
var image = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');
client.vatInvoice(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["type"] = "normal";
let image= new Buffer(fs.readFileSync('./vin.jpeg')).toString('base64');
client.vatInvoice(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
//url识别
var url="http:/test.jpg";
client.vatInvoiceUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
//本地pdf文件识别
var pdf="./test.pdf";
client.vatInvoicePdf(pdf, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url或者pdf文件
type	否	String	可选参数，进行识别的增值税发票类型，默认为 normal，可缺省normal：可识别增值税普票、专票、电子发票roll：可识别增值税卷票

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	是	string	发票种类
InvoiceTypeOrg	是	string	发票名称
InvoiceCode	是	string	发票代码
InvoiceNum	是	string	发票号码
MachineNum	是	string	机打号码。仅增值税卷票含有此参数

MachineCode	是	string	机器编号。仅增值税卷票含有此参数
CheckCode	否	string	校验码。增值税专票无此参数
InvoiceDate	是	string	开票日期
PurchaserName	是	string	购方名称
PurchaserRegisterNum	是	string	购方纳税人识别号
PurchaserAddress	是	string	购方地址及电话
PurchaserBank	是	string	购方开户行及账号
Password	是	string	密码区
Province	是	string	省
City	是	string	市
SheetNum	是	string	联次
Agent	是	string	是否代开
CommodityName	是	array[]	货物名称
- row	是	uint32	行号
- word	是	string	内容
CommodityType	是	array[]	规格型号
- row	是	uint32	行号
- word	是	string	内容
CommodityUnit	是	array[]	单位
- row	是	uint32	行号
- word	是	string	内容
CommodityNum	是	array[]	数量
- row	是	uint32	行号
- word	是	string	内容
CommodityPrice	是	array[]	单价
- row	是	uint32	行号
- word	是	string	内容
CommodityAmount	是	array[]	金额
- row	是	uint32	行号
- word	是	string	内容
CommodityTaxRate	是	array[]	税率
- row	是	uint32	行号
- word	是	string	内容
CommodityTax	是	array[]	税额
- row	是	uint32	行号
- word	是	string	内容
SellerName	是	string	销售方名称
SellerRegisterNum	是	string	销售方纳税人识别号
SellerAddress	是	string	销售方地址及电话
SellerBank	是	string	销售方开户行及账号

TotalAmount	是	uint32	合计金额
TotalTax	是	uint32	合计税额
AmountInWords	是	string	价税合计(大写)
AmountInFiguers	是	uint32	价税合计(小写)
Payee	是	string	收款人
Checker	是	string	复核
NoteDrawer	是	string	开票人
Remarks	是	string	备注

返回示例


```
{
 "log_id": "5425496231209218858",
"words_result_num": 29,
"words_result": {
 "InvoiceNum": "14641426",
 "SellerName": "上海易火广告传媒有限公司",
 "CommodityTaxRate": [
  {
   "word": "6%",
   "row": "1"
  }
 ],
 "SellerBank": "中国银行南翔支行446863841354",
 "Checker": ":沈园园",
 "TotalAmount": "94339.62",
 "CommodityAmount": [
  {
   "word": "94339.62",
   "row": "1"
  }
 ],
 "InvoiceDate": "2016年06月02日",
 "CommodityTax": [
  {
   "word": "5660.38",
   "row": "1"
  }
 ],
 "PurchaserName": "百度时代网络技术(北京)有限公司",
 "CommodityNum": [
  {
   "word": "",
   "row": "1"
  }
 ],
       "Province": "上海",
       "City": "",
       "SheetNum": "第三联",
       "Agent": "否",
 "PurchaserBank": "招商银行北京分行大屯路支行8661820285100030",
 "Remarks": "告传",
 "Password": "074/45781873408>/6>8>65*887676033/51+<5415>9/32--852>1+29<65>641-
5>66<500>87/*-34<943359034>716905113*4242>",
 "SellerAddress": ":嘉定区胜辛南路500号15幢1161室55033753",
 "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二属A2010-59108001",
 "InvoiceCode": "3100153130",
 "CommodityUnit": [
```

```
  "CommodityUnit": [
   {
    "word": "",
    "row": "1"
   }
  ],
  "Payee": ":徐蓉",
  "PurchaserRegisterNum": "110108787751579",
  "CommodityPrice": [
   {
    "word": "",
    "row": "1"
   }
  ],
  "NoteDrawer": "沈园园",
  "AmountInWords": "壹拾万圆整",
  "AmountInFiguers": "100000.00",
  "TotalTax": "5660.38",
  "InvoiceType": "专用发票",
  "SellerRegisterNum": "913101140659591751",
  "CommodityName": [
   {
    "word": "信息服务费",
    "row": "1"
   }
  ],
  "CommodityType": [
   {
    "word": "",
    "row": "1"
   }
  ]
 }
}
```


出租车票识别

```
 //本地文件识别
var fs = require('fs');
var image = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');
client.taxiReceipt(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
//url识别
var url="http://test.jpg";
client.taxiReceiptUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
words_result_num	uint32	是	识别结果数，表示words_result的元素个数
InvoiceCode	string	是	发票代号
InvoiceNum	string	是	发票号码
TaxiNum	string	是	车牌号
Date	string	是	日期
Time	string	是	上下车时间
Fare	string	是	总金额
FuelOilSurcharge	string	是	燃油附加费
CallServiceSurcharge	string	是	叫车服务费
Province	string	是	省
City	string	是	市
PricePerkm	string	是	单价
Distance	string	是	里程

返回示例


```
{
"log_id":2034039896,
"words_result_num":6,
"words_result":
{
 "Date":"2017-11-26",
 "Fare":"¥153.30元",
 "InvoiceCode":"111001681009",
 "InvoiceNum":"90769610",
 "TaxiNum":"BV2062",
 "Time":"20:42-21:07",
 "FuelOilSurcharge": "¥0.00",
 "CallServiceSurcharge": "¥0.00",
      "Province": "浙江省",
      "City": "杭州市",
      "PricePerkm": "2.50元/KM",
      "Distance": "4.5KM"
}
}
```


VIN码识别

```
 //本地文件识别
var fs = require('fs');
var image = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');
client.vinCode(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
//url识别
var url="http://test.jpg";
client.vinCodeUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
location	是	object{}	识别结果
words	是	string	VIN码识别结果
words_result_num	是	int	识别结果数，表示words_result的元素个数

返回示例

```
 {
"log_id": 246589877,
"words_result": [
 {
  "location": {
   "left": 124,
   "top": 11,
   "width": 58,
   "height": 359
  },
  "words": "LFV2A11K8D4010942"
 }
],
"words_result_num": 1
}
```

火车票识别

```
 //本地文件识别
var fs = require('fs');
var image = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');
client.trainTicket(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
//url识别
var url="http://test.jpg";
client.trainTicketUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
ticket_num	string	是	车票号
starting_station	string	是	始发站
train_num	string	是	车次号
destination_station	string	是	到达站
date	string	是	出发日期
ticket_rates	string	是	车票金额
seat_category	string	是	席别
name	string	是	乘客姓名
id_num	string	是	身份证号
serial_number	string	是	序列号
sales_station	string	是	售站
time	string	是	时间
seat_num	string	是	座位号

返回示例

```
{
 "log_id": "12317512659",
 "direction": 1,
 "words_result_num": 13,
 "words_result": {
  "id_num": "2302051998****156X",
  "name": "裴一丽",
  "ticket_rates": "￥54.5元",
  "destination_station": "天津站",
  "seat_category": "二等座",
  "sales_station": "北京南",
  "ticket_num": "F05706",
  "seat_num": "02车03C号",
  "time": "09:36",
  "date": "2019年04月03日",
  "serial_number": "10010300067846",
  "train_num": "C255",
  "starting_station": "北京南站"
 }
}
```


数字识别

```
 //本地文件识别
var fs = require('fs');
var image = new
Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');
var options= {};
options.recognize_granularity='big';
client.numbers(image,options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据
recognize_granularity	false	string	big、small
detect_direction	false	string	true、false

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
      "location": {
        "left": 56,
        "top": 0,
        "width": 21,
        "height": 210
      },
      "words": "3"
    }
  ],
  "words_result_num": 1
}
```

二维码识别

对图片中的二维码、条形码进行检测和识别，返回存储的文字信息。

```
 var fs = require('fs');

var image = new Buffer(fs.readFileSync('example.jpg')).toString('base64');// 二维码识别
$client.qrcode(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};

client.qrcode(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
 // 参数图片url
var url = "https//www.x.com/sample.jpg"
client.qrcodeUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
codes_result_num	是	uint32	识别结果数，表示codes_result的元素个数
codes_result	是	array[]	定位和识别结果数组
-type	是	string	识别码类型条码类型包括：9种条形码（UPC_A、UPC_E、EAN_13、EAN_8、CODE_39、CODE_93、CODE_128、ITF、CODABAR），4种二维码（QR_CODE、DATA_MATRIX、AZTEC、PDF_417）
-text	是	string	条形码识别内容,暂时只限于识别中英文结果

返回示例


```
{
  "log_id": 863402790,
  "codes_result": [
    {
      "type": "QR_CODE",
      "text": [
        "中国",
        "北京"
      ]
    }
  ],
  "codes_result_num": 1
}
```


示例2（多个图的情况）：


```
{
  "log_id": 1508509437,
  "codes_result": [
    {
      "type": "QR_CODE",
      "text": [
        "HTTP://Q8R.HK/YELZ0"
      ]
    },
    {
      "type": "PDF_417",
      "text": [
        "PDF417倢丄TL-30偱撉傒庢傝壜擻偱偡丅"
      ]
    },
    {
      "type": "CODABAR",
      "text": [
        "000800"
      ]
    },
    {
      "type": "CODE_39",
      "text": [
        "1234567890"
      ]
    },
    {
      "type": "AZTEC",
      "text": [
        "www.tec-it.com"
      ]
    },
    {
      "type": "DATA_MATRIX",
      "text": [
        "Wikipedia, the free encyclopedia"
      ]
    },
    {
      "type": "CODE_93",
      "text": [
        "123456789"
      ]
    },
    {
```

```
            "type": "CODE_128",
            "text": [
                "50090500019191"
            ]
        },
        {
            "type": "EAN_8",
            "text": [
                "12345670"
            ]
        },
        {
            "type": "EAN_13",
            "text": [
                "6901234567892"
            ]
        },
        {
            "type": "UPC_E",
            "text": [
                "01234565"
            ]
        }
    ],
    "codes_result_num": 11
}
```

机动车销售发票

支持对机动车销售发票的26个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、机器编号、购买方名称、购买方身份证号码/组织机构代码、车辆类型、厂牌型号、产地、合格证号、发动机号码、车架号码、价税合计、价税合计小写、销货单位名称、电话、纳税人识别号、账号、地址、开户银行、税率、税额、主管税务机关及代码、不含税价格、限乘人数。

```
 var fs = require('fs');

var image = new Buffer(fs.readFileSync('example.jpg')).toString('base64');// 机动车销售发票
$client.vehicleInvoice(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};

client.vehicleInvoice(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
 // 参数图片url
var url = "https//www.x.com/sample.jpg"
client.vehicleInvoiceUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
InvoiceCode	是	string	发票代码/机打代码
InvoiceNum	是	string	发票号码/机打号码
InvoiceDate	是	string	开票日期
MachineCode	是	string	机器编号
Purchaser	是	string	购买方名称
PurchaserCode	是	string	购买方身份证号码/组织机构代码
VehicleType	是	string	车辆类型
ManuModel	是	string	厂牌型号
Origin	是	string	产地
CertificateNum	是	string	合格证号
EngineNum	是	string	发动机号码
VinNum	是	string	车架号码
PriceTax	是	string	价税合计
PriceTaxLow	是	string	价税合计小写
Saler	是	string	销货单位名称
SalerPhone	是	string	销货单位电话
SalerCode	是	string	销货单位纳税人识别号
SalerAccountNum	是	string	销货单位账号
SalerAddress	是	string	销货单位地址
SalerBank	是	string	销货单位开户银行
TaxRate	是	string	税率
Tax	是	string	税额
TaxAuthor	是	string	主管税务机关
TaxAuthorCode	是	string	主管税务机关代码
Price	是	string	不含税价格
LimitPassenger ｜ 是	string	限乘人数	

返回示例

```
{
  "log_id": 283449393728149457,
  "words_result_num": 26,
  "words_result": {
    "InvoiceNum": "00875336",
    "Saler": "深圳市新能源汽车销售有限公司",
    "LimitPassenger": "5",
    "MachineCode": "669745967911",
    "VinNum": "LJLGTCRP1J4007581",
    "TaxRate": "16%",
    "PriceTaxLow": "106100.00",
    "InvoiceDate": "2018-11-29",
    "Price": "￥91465.52",
    "SalerBank": "中国工商银行股份有限公司深圳岭园支行",
    "TaxAuthor": "国家锐务总局深圳市龙岗区税务局第五税务所",
    "ManuModel": "江淮牌HFC7007EYBD6",
    "CertificateNum": "WCH0794J0976801",
    "Purchaser": "苏子潇",
    "VehicleType": "纯电动轿车",
    "InvoiceCode": "14975047560",
    "PriceTax": "壹拾万陆仟壹佰圆整",
    "SalerPhone": "0755-83489306",
    "SalerAddress": "深圳市龙岗区龙岗街道百世国际汽车城",
    "Origin": "安徽省合肥市",
    "EngineNum": "18958407",
    "Tax": "14634.48",
    "PurchaserCode": "5135934475603742222",
    "TaxAuthorCode": "14037589413",
    "SalerAccountNum": "中国工商银行股份有限公司深圳岭园支行",
    "SalerCode": "91449283464582922 78H"
  }
}
```


车辆合格证

支持对车辆合格证的23个关键字段进行结构化识别，包括合格证编号、发证日期、车辆制造企业名、车辆品牌、车辆名称、车辆型号、车架号、车身颜色、发动机型号、发动机号、燃料种类、排量、功率、排放标准、轮胎数、轴距、轴数、转向形式、总质量、整备质量、驾驶室准乘人数、最高设计车速、车辆制造日期。

```
 var fs = require('fs');

var image = new Buffer(fs.readFileSync('example.jpg')).toString('base64');// 车辆合格证
$client.vehicleCertificate(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};

client.vehicleCertificate(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
 // 参数图片url
var url = "https//www.x.com/sample.jpg"
client.vehicleCertificateUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
CertificationNo	是	string	合格证编号
CertificateDate	是	string	发证日期
Manufacturer	是	string	车辆制造企业名
CarBrand	是	string	车辆品牌
CarName	是	string	车辆名称
CarModel	是	string	车辆型号
VinNo	是	string	车架号
CarColor	是	string	车身颜色
EngineType	是	string	发动机型号
EngineNo	是	string	发动机号
FuelType	是	string	燃料种类
Displacement	是	string	排量
Power	是	string	功率
EmissionStandard	是	string	排放标准
TyreNum	是	string	轮胎数
Wheelbase	是	string	轴距
AxleNum	是	string	轴数
SteeringType	是	string	转向形式
TotalWeight	是	string	总质量
SaddleMass	是	string	整备质量
LimitPassenger	是	string	驾驶室准乘人数
SpeedLimit	是	string	最高设计车速
ManufactureDate	是	string	车辆制造日期

返回示例

```
{
  "log_id": 14814098736243057,
  "words_result_num": 23,
  "words_result": {
     "ManufactureDate": "2016年10月13日",
     "CarColor": "红",
     "LimitPassenger": "2",
     "EngineType": "WP12.460E50",
     "TotalWeight": "25000",
     "Power": "338",
     "CertificationNo": "WEK29JX98645437",
     "FuelType": "汽油",
     "Manufacturer": "陕西汽车集团有限责任公司",
     "SteeringType": "方向盘",
     "Wheelbase": "3175+1350",
     "SpeedLimit": "105",
     "EngineNo": "1418K129178",
     "SaddleMass": "8600",
     "AxleNum": "3",
     "CarModel": "SX4250MC4",
     "VinNo": "LZGJHYD83JX197344",
     "CarBrand": "陕汽牌",
     "EmissionStandard": "GB17691-2005国V,GB3847-2005",
     "Displacement": "11596",
     "CertificateDate": "2018年11月28日",
     "CarName": "牵引汽车",
     "TyreNum": "10"
  }
}
```


户口本识别

支持对户口本内常住人口登记卡的全部 22 个字段进行结构化识别，包括户号、姓名、与户主关系、性别、出生地、民族、出生日期、身份证号、本市县其他住址、曾用名、籍贯、宗教信仰、身高、血型、文化程度、婚姻状况、兵役状况、服务处所、职业、何时由何地迁往本市、何时由何地迁往本址、登记日期。

```
 var fs = require('fs');

var image = new Buffer(fs.readFileSync('example.jpg')).toString('base64');// 户口本识别
$client.householdRegister(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};

client.householdRegister(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
 // 参数图片url
var url = "https//www.x.com/sample.jpg"
client.householdRegisterUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和image二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
Name	是	object{}	姓名
+ words	是	string	所属字段的具体内容，以下各字段均含有此元素
Relationship	是	object{}	户主或与户主关系
Sex	是	object{}	性别
BirthAddress	是	object{}	出生地
Nation	是	object{}	民族
Birthday	是	object{}	生日
CardNo	是	object{}	身份证号
HouseholdNum	是	object{}	户号
FormerName	是	object{}	曾用名
Hometown	是	object{}	籍贯
OtherAddress	是	object{}	本市（县）其他住址
Belief	是	object{}	宗教信仰
Height	是	object{}	身高
BloodType	是	object{}	血型
Education	是	object{}	文化程度
MaritalStatus	是	object{}	婚姻状况
VeteranStatus	是	object{}	兵役状况
WorkAddress	是	object{}	服务处所
Career	是	object{}	职业
WWToCity	是	object{}	何时由何地迁来本市(县)
WWHere	是	object{}	何时由何地迁往本址
Date	是	object{}	登记日期

返回示例

```
{
  "log_id": 1301870459,
  "words_result": {
     "BirthAddress": {
        "words": "河南洛阳市郊区"
     },
     "Birthday": {
        "words": "2016-07-28"
     },
     "CardNo": {
        "words": "410311201607282825"
     },
     "Name": {
        "words": "孙翌晨"
     },
     "Nation": {
        "words": "汉族"
     },
     "Relationship": {
        "words": "户主"
     },
     "Sex": {
        "words": "男"
     },
  },
  "words_result_num": 7
}
```


手写文字识别

支持对图片中的手写中文、手写数字进行检测和识别，针对不规则的手写字体进行专项优化，识别准确率可达90%以上。

```
 var fs = require('fs');

var image = new Buffer(fs.readFileSync('example.jpg')).toString('base64');// 手写文字识别
$client.handwriting(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};

client.handwriting(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
 // 参数图片url
var url = "https//www.x.com/sample.jpg"
client.handwritingUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
recognize_granularity	否	string	big、small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true/false	是否返回识别结果中每一行的置信度，默认为false，不返回置信度
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: true：检测朝向； false：不检测朝向

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	float	当请求参数 **probability=true** 时返回该字段，表示识别结果中每一行的置信度值，包含： - **average**： 行置信度平均值 - **variance**：行置信度方差 - **min**：行置信度最小值
direction	否	int32	图像方向，当detect_direction=true时存在 -1:未定义， 0:正向， 1: 逆时针90度， 2:逆时针180度， 3:逆时针270度

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
    {
        "location": {
            "left": 56,
            "top": 0,
            "width": 21,
            "height": 210
        },
        "words": "3"
    }
  ],
  "words_result_num": 1
}
```

飞机行程单识别

支持对飞机行程单的24个字段进行结构化识别，包括电子客票号、印刷序号、姓名、始发站、目的站、航班号、日期、时间、票价、身份证号、承运人、民航发展基金、保险费、燃油附加费、其他税费、合计金额、填开日期、订票渠道、客票级别、座位等级、销售单位号、签注、免费行李、验证码。 同时，支持单张行程单上的多航班信息识别。

```
 var fs = require('fs');

var image = new Buffer(fs.readFileSync('example.jpg')).toString('base64');// 飞机行程单识别
$client.airTicket(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};

client.airTicket(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
 // 参数图片url
var url = "https//www.x.com/sample.jpg"
client.airTicketUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
multi_detect	否	bool	true/false	控制是否开启多航班信息识别功能，**默认值：false** - **true：开启多航班信息识别功能**，开启后返回结果中对应字段格式将改为数组类型 - **false：** 不开启，仅识别单一航班信息

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
name	是	string	姓名
starting_station	是	string	始发站
destination_station	是	string	目的站
flight	是	string	航班号
date	是	string	日期
ticket_number	是	string	电子客票号码
fare	是	string	票价
dev_fund	是	string	民航发展基金/基建费
fuel_surcharge	是	string	燃油附加费
other_tax	是	string	其他税费
ticket_rates	是	string	合计金额
issued_date	是	string	填开日期
id_num	是	string	身份证号
carrier	是	string	承运人
time	是	string	时间
issued_by	是	string	订票渠道
serial_number	是	string	印刷序号
insurance	是	string	保险费
fare_basis	是	string	客票级别
class	是	string	座位等级
agent_code	是	string	销售单位号
endorsement	是	string	签注
allow	是	string	免费行李
ck	是	string	验证码

返回示例

```
 // 识别单航班信息（multi_detect=false，或参数缺省）
{
    "log_id": 7306800033425229106,
    "direction": 0,
    "words_result_num": 18,
    "words_result": {
        "insurance": "20.00",
        "date": "2019-10-22",
        "allow": "20K",
        "flight": "CA6589",
        "issued_by": "中国国际航空服务有限公司",
        "starting_station": "武汉",
        "fare": "260.00",
        "endorsement": "不得签转改期退转",
        "ticket_rates": "350.00",
        "ck": "5866",
```

```
        "serial_number": "51523588676",
        "ticket_number": "7843708871196",
        "fuel_surcharge": "EXEMPT",
        "carrier": "南航",
        "issued_date": "2019-10-30",
        "other_tax": "",
        "fare_basis": "NREOW",
        "id_num": "411201123909020877",
        "destination_station": "合肥",
        "name": "郭达",
        "agent_code": "BJS19197300025",
        "time": "21:25",
        "class": "N",
        "dev_fund": "50.00"
    }
}

// 识别多航班信息（multi_detect=true)
{
    "words_result": {
        "insurance": [
            {
                "word": "XXX"
            }
        ],
        "date": [
            {
                "word": "2019-10-18"
            },
            {
                "word": "2019-10-21"
            }
        ],
        "flight": [
            {
                "word": "CZ3565"
            },
            {
                "word": "CZ3566"
            }
        ],
        "issued_by": [
            {
                "word": "上海携程旅行社有限公司"
            }
        ],
        "starting_station": [
            {
                "word": "北京"
            }
        ],
        "fare": [
            {
                "word": "1080.00"
            }
        ],
        "ticket_rates": [
            {
                "word": "1420.00"
            }
        ],
        "serial_number": [
```

```
            {
                "word": "45956029770"
            }
        ],
        "ticket_number": [
            {
                "word": "7849648364314"
            }
        ],
        "fuel_surcharge": [
            {
                "word": "240.00"
            }
        ],
        "carrier": [
            {
                "word": "南航"
            },
            {
                "word": "南航"
            }
        ],
        "issued_date": [
            {
                "word": "2019-09-18"
            }
        ],
        "other_tax": [],
        "id_num": [
            {
                "word": "0789654700"
            }
        ],
        "destination_station": [
            {
                "word": "深圳"
            },
            {
                "word": "北京"
            }
        ],
        "name": [
            {
                "word": "姚佳"
            }
        ],
        "time": [
            {
                "word": "13:55"
            },
            {
                "word": "16:30"
            }
        ],
        "dev_fund": [
            {
                "word": "100.00"
            }
        ]
    },
    "log_id": "1280814270572920832",
    "words_result_num": 18
}
```

```
}
```

通用机打发票

支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别，包括发票类型、发票号码、发票代码、开票日期、合计金额大写、合计金额小写、商品名称、商品单位、商品单价、商品数量、商品金额、机打代码、机打号码、校验码、销售方名称、销售方纳税人识别号、购买方名称、购买方纳税人识别号、合计税额等。

```
 var fs = require('fs');

var image = new Buffer(fs.readFileSync('example.jpg')).toString('base64');// 通用机打发票
$client.invoice(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};

client.invoice(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
 // 参数图片url
var url = "https//www.x.com/sample.jpg"
client.invoiceUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	string	true/false	是否输出位置信息，true：输出位置信息，false：不输出位置信息，默认false

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	否	string	发票类型
InvoiceCode	否	string	发票代码
InvoiceNum	否	string	发票号码
InvoiceDate	否	string	开票日期
AmountInFiguers	否	string	合计金额小写
AmountInWords	否	string	合计金额大写
CommodityName	否	string	商品名称
CommodityUnit	否	string	商品单位
CommodityPrice	否	string	商品单价
CommodityNum	否	string	商品数量
CommodityAmount	否	string	商品金额
MachineCode	否	string	机打代码
MachineNum	否	string	机打号码
CheckCode	否	string	校验码
SellerName	否	string	销售方名称
SellerRegisterNum	否	string	销售方纳税人识别号
PurchaserName	否	string	购买方名称
PurchaserRegisterNum	否	string	购买方纳税人识别号
TotalTax	否	string	合计税额
Province	否	string	省
City	否	string	市
Time	否	string	时间
SheetNum	否	string	联次

返回示例

```
{
  "log_id": 4423022131715883558,
  "direction": 0,
  "words_result_num": 22,
  "words_result": {
     "City": "",
     "InvoiceNum": "01445096",
     "SellerName": "百度餐饮店",
     "IndustrSort": "生活服务",
     "Province": "广东省",
     "CommodityAmount": [
        {
           "word": "183.00",
           "row": "1"
        }
     ],
     "InvoiceDate": "2020年07月28日",
     "PurchaserName": "中信建投证券股份有限公司",
     "CommodityNum": [],
     "InvoiceCode": "144001901511",
     "CommodityUnit": [],
     "SheetNum": "",
     "PurchaserRegisterNum": "91442300845348OX9",
     "Time": "",
     "CommodityPrice": [],
     "AmountInFiguers": "183.00",
     "AmountInWords": "壹佰捌拾叁元整",
     "CheckCode": "61042119820421061301",
     "TotalTax": "183.00",
     "InvoiceType": "广东通用机打发票",
     "SellerRegisterNum": "61042119820421061301",
     "CommodityName": [
        {
           "word": "餐费",
           "row": "1"
        }
     ]
  }
}
```


护照识别

支持对中国大陆护照个人资料页所有15个字段进行结构化识别，包括国家码、护照号、姓名、姓名拼音、性别、出生地点、出生日期、签发地点（不支持境外签发地）、签发日期、有效期、签发机关、护照类型、国籍、MRZCode1、MRZCode2。

```
 var fs = require('fs');

var image = new Buffer(fs.readFileSync('example.jpg')).toString('base64');// 护照识别
$client.passport(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};

client.passport(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
 // 参数图片url
var url = "https//www.x.com/sample.jpg"
client.passportUrl(url, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
-location	是	uint32	水平坐标
-words	是	string	识别内容

返回示例

```
{
  "log_id": 7377468409496932872,
  "words_result_num": 14,
  "words_result": {
     "国家码": {
```

```
        "location": {
            "width": 59,
            "top": 200,
            "left": 762,
            "height": 26
        },
        "words": "CHN"
    },
    "护照签发地点": {
        "location": {
            "width": 236,
            "top": 505,
            "left": 558,
            "height": 43
        },
        "words": "山东/SHANDONG"
    },
    "MRZCode2": {
        "location": {
            "width": 1252,
            "top": 797,
            "left": 145,
            "height": 88
        },
        "words": "E898657303CHNSDM0702<<<<<<<<<<<<<<"
    },
    "有效期至": {
        "location": {
            "width": 287,
            "top": 528,
            "left": 955,
            "height": 46
        },
        "words": "20261004"
    },
    "签发机关": {
        "location": {
            "width": 271,
            "top": 583,
            "left": 552,
            "height": 42
        },
        "words": "出入境管理局"
    },
    "MRZCode1": {
        "location": {
            "width": 1201,
            "top": 781,
            "left": 162,
            "height": 45
        },
        "words": "PONSUN<<JIAJIA<<<<<<<<<<<<<<<<<<<<<<<<<<"
    },
    "护照号码": {
        "location": {
            "width": 237,
            "top": 215,
            "left": 994,
            "height": 48
        },
        "words": "E89865730"
    },
```

```
        "签发日期": {
            "location": {
                "width": 279,
                "top": 453,
                "left": 955,
                "height": 46
            },
            "words": "20161005"
        },
        "出生地点": {
            "location": {
                "width": 216,
                "top": 429,
                "left": 564,
                "height": 43
            },
            "words": "山东/SHANDONG"
        },
        "姓名": {
            "location": {
                "width": 159,
                "top": 247,
                "left": 581,
                "height": 34
            },
            "words": "孙嘉佳"
        },
        "姓名拼音": {
            "location": {
                "width": 229,
                "top": 279,
                "left": 578,
                "height": 41
            },
            "words": "SUN,JIAJIA"
        },
        "国籍": {
            "location": {
                "width": 209,
                "top": 366,
                "left": 695,
                "height": 42
            },
            "words": "中国/CHINESE"
        },
        "生日": {
            "location": {
                "width": 202,
                "top": 382,
                "left": 950,
                "height": 39
            },
            "words": "19950723"
        },
        "性别": {
            "location": {
                "width": 73,
                "top": 357,
                "left": 570,
                "height": 34
            },
            "words": "男/M"
        }
```

```
    }
  }
```

网约车行程单识别

对各大主要服务商的网约车行程单进行结构化识别，包括滴滴打车、花小猪打车、高德地图、曹操出行、阳光出行，支持识别服务商、行程开始时间、行程结束时间、车型、总金额等16 个关键字段。

```
 let fs = require('fs');
let AipOcrClient = require("../src/AipOcr");
let client=AipOcr new AipOcrClient("APPID", 'AK', 'Sk');

var image = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var pdf_file = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var url = "https//www.x.com/sample.jpg";
// 调用网约车行程单识别
client.onlineTaxiItinerary(image).then(function(result) {
 console.log(JSON.stringify(result));
});
client.onlineTaxiItineraryUrl(url).then(function(result) {
 console.log(JSON.stringify(result));
});
client.onlineTaxiItineraryPdf(pdf_file).then(function(result) {
 console.log(JSON.stringify(result));
});
//  如果有可选参数
var options = {};
options["pdf_file_num", 1];
client.onlineTaxiItineraryPdf(pdf_file, options).then(function(result) {
 console.log(JSON.stringify(result));
});
```

请求参数详情

参数	是否必选	类型	说明
image	和url二选一	string	图像数据，base64编码后进行urlencode，base64编码去除编码头（data:image/jpeg;base64），要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效，请注意关闭URL防盗链
pdf_file	和image/url 三选一	string	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ ServiceProvider	是	string	服务商
+ StartTime	是	string	行程开始时间
+ EndTime	是	string	行程结束时间
+ Phone	是	string	行程人手机号
+ ApplicationDate	是	string	申请日期
+ TotalFare	是	string	总金额
+ ItemNum	是	array	行程信息中包含的行程数量
+ Items	是	array	行程信息
++ ItemId	是	string	行程信息的对应序号
++ PickupTime	是	string	上车时间
++ PickupDate	是	string	上车日期
++ CarType	是	string	车型
++ Distance	是	string	里程
++ StartPlace	是	string	起点
++ DestinationPlace	是	string	终点
++ City	是	string	城市
++ Fare	是	string	金额
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 1385196013945356288,
  "words_result_num": 7
  "words_result": {
     "TotalFare": "2316",
     "EndTime": "2020-07-30 19:00",
     "Phone": "13000000000",
     "ServiceProvider": "滴滴企业版",
     "StartTime": "2020-07-01 16:00",
     "ApplicationDate": "2017-12-08",
     "ItemId": "3"
     "items": [
        {
           "ItemId": "1",
           "StartPlace": "鱼化寨地铁-D口",
           "PickupTime": "16:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.7",
           "PickupDate": "20-07-01",
           "DestinationPlace": "创新港",
           "Fare": "20.86"
        },
        {
           "ItemId": "2",
           "StartPlace": "科学园东门",
           "PickupTime": "14:56",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "91",
           "PickupDate": "20-07-02",
           "DestinationPlace": "鱼化寨地铁站",
           "Fare": "18.58"
        },
        {
           "ItemId": "3",
           "StartPlace": "中俄丝路创新园东门",
           "PickupTime": "19:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.1",
           "PickupDate": "20-07-30",
           "DestinationPlace": "新门地铁站",
           "Fare": "20.38"
        },
     ],
  },
}
```


磅单识别

结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单。

```
 let fs = require('fs');
let AipOcrClient = require("../src/AipOcr");
let client=AipOcr new AipOcrClient("APPID", 'AK', 'Sk');

var url = "https//www.x.com/sample.jpg";
var pdf_file = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var image = new Buffer(fs.readFileSync('文件路径')).toString('base64');
// 调用磅单识别
client.weightNote(image).then(function(result) {
 console.log(JSON.stringify(result));
});
client.weightNoteUrl(url).then(function(result) {
 console.log(JSON.stringify(result));
});
client.weightNotePdf(pdf_file).then(function(result) {
 console.log(JSON.stringify(result));
});
//  如果有可选参数
var options = {};
options["pdf_file_num", 1];
options["probability", false];
client.weightNote(image, options).then(function(result) {
 console.log(JSON.stringify(result));
});
client.weightNoteUrl(url, options).then(function(result) {
 console.log(JSON.stringify(result));
});
client.weightNotePdf(pdf_file, options).then(function(result) {
 console.log(JSON.stringify(result));
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ PlateNum	是	object	车牌号
+ PrintTime	是	object	打印时间
+ CrossWeight	是	object	毛重
+ TareWeight	是	object	皮重
+ NetWeight	是	object	净重
+ SendingCompany	是	object	发货单位
+ ReceivingCompany	是	object	收货单位
+ DeliveryNumber	是	object	单号
++ word	是	string	字段识别结果，以上各字段均包含此参数
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "words_result": [
    {
      "TareWeight": [
      {
            "word": "14.20"
        }
      ],
      "CrossWeight": [
        {
          "word": "50.70"
        }
      ],
      "PlateNum": [
        {
          "word": "京A12356"
        }
      ],
      "SendingCompany": [
       {
          "word": "宣化县耿矿煤业有限公司"
        }
      ],
      "DeliveryNumber": [
       {
          "word": "278933000"
        }
      ],
      "ReceivingCompany": [
        {
          "word": "宁夏市京裕达实业公司"
        }
      ],
      "PrintTime": [
        {
          "word": "2020年1月1日"
        }
      ],
      "NetWeight": [
        {
          "word": "36.50"
        }
      ]
    }
  ],
  "words_result_num": 1,
  "log_id": 1428311410130160734
}
```


医疗费用明细识别

SDK 调用示例

```
 let fs = require('fs');
let AipOcrClient = require("../src/AipOcr");
let client = AipOcr new AipOcrClient("APPID", 'AK', 'Sk');

 var image = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var url = "https//www.x.com/sample.jpg";

 // 调用医疗费用明细识别
client.medicalDetail(image).then(function(result) {
 console.log(JSON.stringify(result));
});
client.medicalDetailUrl(url).then(function(result) {
 console.log(JSON.stringify(result));
});
```

接口详情

可参考API文档：医疗费用明细识别

出生证明识别

SDK 调用示例

```
 let fs = require('fs');
let AipOcrClient = require("../src/AipOcr");
let client = AipOcr new AipOcrClient("APPID", 'AK', 'Sk');

 var image = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var url = "https//www.x.com/sample.jpg";

 // 调用出生证明识别
client.birthCertificateV1(image).then(function(result) {
 console.log(JSON.stringify(result));
});
client.birthCertificateV1Url(url).then(function(result) {
 console.log(JSON.stringify(result));
});
```

接口详情

可参考API文档：出生证明识别

定额发票识别_旧

SDK 调用示例

```
 let fs = require('fs');
let AipOcrClient = require("../src/AipOcr");
let client=AipOcr new AipOcrClient("APPID", 'AK', 'Sk');

 var image = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var url = "https//www.x.com/sample.jpg";
var pdf_file = new Buffer(fs.readFileSync('文件路径')).toString('base64');

 // 调用定额发票识别_旧
client.quotaInvoiceV1(image).then(function(result) {
 console.log(JSON.stringify(result));
});
client.quotaInvoiceV1Url(url).then(function(result) {
 console.log(JSON.stringify(result));
});
client.quotaInvoiceV1Pdf(pdf_file).then(function(result) {
 console.log(JSON.stringify(result));
});
```

接口详情

可参考API文档：定额发票识别_旧

印章识别

SDK 调用示例

```
 let fs = require('fs');
let AipOcrClient = require("../src/AipOcr");
let client=AipOcr new AipOcrClient("APPID", 'AK', 'Sk');

 var image = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var url = "https//www.x.com/sample.jpg";
var pdf_file = new Buffer(fs.readFileSync('文件路径')).toString('base64');

 // 调用印章识别
client.sealV1(image).then(function(result) {
 console.log(JSON.stringify(result));
});
client.sealV1Url(url).then(function(result) {
 console.log(JSON.stringify(result));
});
client.sealV1Pdf(pdf_file).then(function(result) {
 console.log(JSON.stringify(result));
});
```

接口详情

可参考API文档：印章识别

办公文档识别

SDK 调用示例

```
 let fs = require('fs');
let AipOcrClient = require("../src/AipOcr");
let client=AipOcr new AipOcrClient("APPID", 'AK', 'Sk');

 var image = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var url = "https//www.x.com/sample.jpg";
var pdf_file = new Buffer(fs.readFileSync('文件路径')).toString('base64');

 // 调用办公文档识别
client.docAnalysisOfficeV1(image).then(function(result) {
 console.log(JSON.stringify(result));
});
client.docAnalysisOfficeV1Url(url).then(function(result) {
 console.log(JSON.stringify(result));
});
client.docAnalysisOfficeV1Pdf(pdf_file).then(function(result) {
 console.log(JSON.stringify(result));
});
```

接口详情

可参考API文档：办公文档识别

医疗费用明细识别

```
 let fs = require('fs');
let AipOcrClient = require("../src/AipOcr");
let client=AipOcr new AipOcrClient("APPID", 'AK', 'Sk');

var image = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var url = "https//www.x.com/sample.jpg";
// 调用医疗费用明细识别
client.medicalDetail(image).then(function(result) {
 console.log(JSON.stringify(result));
});
client.medicalDetailUrl(url).then(function(result) {
 console.log(JSON.stringify(result));
});
//  如果有可选参数
var options = {};
options["location", ];
options["probability", false];
client.medicalDetail(image, options).then(function(result) {
 console.log(JSON.stringify(result));
});
client.medicalDetailUrl(url, options).then(function(result) {
 console.log(JSON.stringify(result));
});
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省** - **false**：不返回字段位置信息 - **true**：返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ Name	是	object	姓名
+ Date	是	object	日期
+ PatientID	是	object	病人ID
+ TotalAmount	是	object	总金额
+ word	是	string	字段识别结果，以上各字段均包含此参数
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+ CostDetail	是	array[]	项目明细

CostDetail字段包含多个Array，每个数组包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：项目类型、项目名称、单价、数量、规格、金额
++ word	word_name字段对应的识别结果

返回示例

```
{
  "log_id": 1397090241579319296,
```

```
"words_result_num": 5,
"words_result": {
  "PatientID": {
     "word": "23683829"
  },
  "TotalAmount": {
     "word": "600.00"
  },
  "Date": {
     "word": "2020年11月04日"
  },
  "Name": {
     "word": "范浩"
  },
  "CostDetail": [
    [
      {
        "word_name": "清单项目名称",
        "word": "普通过敏原(新)筛查"
      },
      {
        "word_name": "单价",
        "word": "580.00"
      },
      {
        "word_name": "数量",
        "word": "1.00"
      },
      {
        "word_name": "金额",
        "word": "580.00"
      },
      {
        "word_name": "规格",
        "word": "次"
      },
      {
        "word_name": "项目类型",
        "word": "化验费"
      }
    ],
    [
      {
        "word_name": "清单项目名称",
        "word": "总IgE测定"
      },
      {
        "word_name": "单价",
        "word": "20.00"
      },
      {
        "word_name": "数量",
        "word": "1.00"
      },
      {
        "word_name": "金额",
        "word": "20.00"
      },
      {
        "word_name": "规格",
        "word": "次"
      },
```

```
            {
                "word_name": "项目类型",
                "word": "化验费"
            }
        ]
    ],
  },
}
```

图文转换器（接口版）--提交请求

图文转换器对应的接口版产品，可识别图片/PDF文件中的文本内容，进行智能版式分析，并转换为保留原文档版式的Word、Excel文档，返回文档下载连接，支持含表格、印章、手写等内容的文档。满足文档版式还原、企业档案电子化等信息管理需求。如需直接在线使用轻应用，可到控制台-图文转换器使用。

```
 let fs = require('fs');
let AipOcrClient = require("../src/AipOcr");
let client=AipOcr new AipOcrClient("APPID", 'AK', 'Sk');

 var image = new Buffer(fs.readFileSync('文件路径')).toString('base64');
var url = "https//www.x.com/sample.jpg";
var pdf_file = new Buffer(fs.readFileSync('文件路径')).toString('base64');

 // 调用图文转换器（接口版）--提交请求
client.docConvertRequestV1(image).then(function(result) {
 console.log(JSON.stringify(result));
});
client.docConvertRequestV1Url(url).then(function(result) {
 console.log(JSON.stringify(result));
});
client.docConvertRequestV1Pdf(pdf_file).then(function(result) {
 console.log(JSON.stringify(result));
});
```

请求参数详情

参数	是否必选	类型	说明
image	和 url/pdf_file 三选一	string	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **优先级：** image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式 **优先级：** image > url > pdf_file，当image字段存在时url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不不超过10M **优先级：** image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容；**若不传入，默认识别文件所有页，页码从1开始**

返回参数详情

字段	类型	说明
success	bool	当前请求状态； true 表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该请求生成的task_id，后续使用该task_id获取识别结果
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": 12345,
  "result":{
     "task_id":"task-xxxxxxx",
  },
  "code":1001,
  "message": "Create task successfully!"
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
  "success":false,
  "log_id": 12345,
  "error_code": 216401,
  "error_msg": "Create task failed!"
}
```

图文转换器（接口版）--获取结果

```
  var task_id = "xxxxx";

// 调用图文转换器（接口版）--获取结果
client.tableGetresult(task_id).then(function(result) {
    console.log(JSON.stringify(result));
});
```

请求参数详情

参数	是否必选	类型	说明
task_id	是	string	发送提交请求时返回的task_id

返回参数详情

字段	类型	说明
success	bool	当前请求状态； true表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该文件对应请求的task_id
+ ret_code	int	识别状态，1：任务未开始；2：进行中；3：已完成
+ ret_msg	string	识别状态信息：任务未开始；进行中；已完成
+ percent	int	文档转换进度（百分比）
+ result_data	dict	识别结果字符串，返回word、excel的文件分别的下载地址
+ +word	string	还原后的word文件的下载地址，**文件识别失败时返回""**
+ +excel	string	还原后的Excel文件的下载地址，**若文档中没有表格则返回""**
+ create_time	datetime	任务创建时间
+ start_time	datetime	任务开始时间
+ end_time	datetime	任务结束时间
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": "xxxxxx",
  "result":{
     "task_id":"task-xxxxxxx",
     "ret_code": 3,
     "ret_msg": "已完成",
     "percent": 100,
     "result_data": {
        "word": "word_download_url",
        "excel": "",
     },
     "create_time": "2023-01-17 11:06:12",
     "start_time": "2023-01-17 11:06:13",
     "end_time": "2023-01-17 11:06:15"
  },
  "code":1001,
  "message": "Query task successfully!""
}
```

若查询的task_id不存在, 返回result为{}。 请求失败响应体示例如下：

```
{
  "code":1001,
  "log_id":1635891796603052032,
  "message":"Query task successfully!",
  "result":{},
  "success":true
}
```

表格文字识别同步接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别同步接口。

表格文字识别

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别。

表格识别结果

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别结果。

表格识别接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别接口。

错误信息

错误返回格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

错误码

SDK本地检测参数返回的错误码：

error_code	error_msg	描述
SDK100	image size error	图片大小超限
SDK101	image length error	图片边长不符合要求
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时
SDK109	unsupported image format	不支持的图片格式

服务端返回的错误码:

错误码	错误信息	描述
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
	Open api	

17	daily request limit reached	每天流量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
282000	internal error	服务器内部错误，如果您使用的是高精度接口，报这个错误码的原因可能是您上传的图片中文字过多，识别超时导致的，建议您对图片进行切割后再识别，其他情况请再次请求， 如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
282003	missing parameters: {参数名}	请求参数缺失
	batch	

282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282110	urls not exit	URL参数不存在，请核对URL后再次提交
282111	url format illegal	URL格式非法，请检查url格式是否符合相应接口的入参要求
282112	url download timeout	url下载超时，请检查url对应的图床/图片无法下载或链路状况不好，您可以重新尝试以下，如果多次尝试后仍不行，建议更换图片地址
282113	url response invalid	URL返回无效参数
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误

C++语言

简介

Hi，您好，欢迎使用百度文字识别服务。

本文档主要针对C++开发者，描述百度文字识别接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 在百度智能云控制台内提交工单，咨询问题类型请选择**人工智能服务**；
- 如有疑问，进入AI社区交流：http://ai.baidu.com/forum/topic/list/164

接口能力

接口名称	接口能力简要描述
通用文字识别	识别图片中的文字信息
通用文字识别（高精度版）	更高精度地识别图片中的文字信息
通用文字识别（含位置信息版）	识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（高精度含位置版）	更高精度地识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（含生僻字版）	识别图片中的文字信息（包含对常见字和生僻字的识别）
网络图片文字识别	识别一些网络上背景复杂，特殊字体的文字

网络图片文字识别（含位置版）	识别网络图片中的文字内容（包含文字区域的坐标信息）
身份证识别	识别身份证正反面的文字信息
银行卡识别	识别银行卡的卡号并返回发卡行和卡片性质信息
驾驶证识别	识别机动车驾驶证所有关键字段
行驶证识别	识别机动车行驶证所有关键字段
车牌识别	识别中国大陆各类机动车车牌信息
营业执照识别	对营业执照进行识别
表格文字识别	自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容
通用票据识别	对各类票据图片（医疗票据，保险保单等）进行文字识别，并返回文字在图片中的位置信息
增值税发票识别	对增值税发票进行文字识别，并结构化返回字段信息，支持增值税专票、普票、电子发票
出租车票识别	针对全国各大城市出租车票的发票号码、发票代码、车号、日期、时间、金额等进行结构化识别
VIN码识别	对车辆车架、挡风玻璃上的VIN码进行识别
火车票识别	支持对大陆火车票的车票号、始发站、目的站、车次、日期、票价、席别、姓名进行结构化识别
飞机行程单识别	支持对飞机行程单的24个字段进行结构化识别
二维码识别	对图片中的二维码、条形码进行检测和识别，返回存储的文字信息
数字识别	识别图片中的数字，适用于手机号提取、快递单号提取、充值号码提取等场景
手写文字识别	支持对图片中的手写中文、手写数字进行检测和识别
护照识别	支持对中国大陆护照个人资料页所有15个字段进行结构化识别
户口本识别	对出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码字段进行识别
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果
通用机打发票	支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别
机动车销售发票	支持对机动车销售发票的26个关键字段进行结构化识别
车辆合格证	支持对车辆合格证的23个关键字段进行结构化识别
通用机打发票	对国家/地方税务局发行的横/竖版通用机打发票进行结构化识别
护照识别	支持对中国大陆护照个人资料页所有11个字段进行结构化识别
医疗费用明细识别	支持识别全国医疗费用明细识别
网约车行程单识别	对国家/地方税务局发行的横/对各大主要服务商的网约车行程单进行结构化识别
磅单识别	结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单
仪器仪表盘读数识别	适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号
自定义模板文字识别	针对固定版式卡证票据提供的 OCR 定制化产品，可由用户自助创建识别模板和分类器，实现对任意版式卡证票据进行自动分类并结构化输出识别结果
医疗费用明细	支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含

识别	项目类型、项目名称、单价、数量、规格、金额
办公文档识别	可对办公类文档的版面进行分析，输出图、表、标题、文本、目录、栏、页眉、页脚、页码和脚注的位置，并输出分版块内容的OCR识别结果
印章识别	检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别
机动车登记证书识别	对机动车登记证书的编号、机动车所有人、登记机关、车辆类型、发证机关章等15个关键字段进行结构化识别
智能财务票据识别	对增值税发票、卷票、火车票、出租车票、机票行程单等13类票据混贴的图片进行切分识别
增值税发票验真	支持9种增值税发票的真伪及字段信息准确性校验，包括增值税专票、电子专票、普票、电子普票、卷票、通行费增值税电子普票、货运专票、机动车销售发票、二手车销售发票，支持返回票面的全部信息
医疗发票识别	支持识别全国各地门诊/住院发票的 业务流水号、发票号、住院号、门诊号、病例号、姓名、性别、社保卡号、金额大/小写、收款单位、省市、医保统筹支付、个人账户支付等关键字段。支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数
门脸文字识别	识别图片中的门脸文字信息，自动过滤非门脸文字内容，接口返回门脸名称、描述文字和置信度
车辆证照混贴识别	对机动车行驶证主页及副页、驾驶证主页及副页在同一张图片上的场景进行结构化识别
公式识别	对试卷中的数学公式及题目内容进行识别
图文转换器	可识别图片/PDF文档版面布局，提取文字内容，并转换为保留原文档版式的Word、Excel文档，方便二次编辑和复制，可支持含表格、印章、水印、手写等内容的文档

版本更新记录

上线日期	版本号	更新内容
2021.12.11	4.15.8	新增： 网约车行程单识别，磅单识别，医疗明细识别
2021.6.3	4.15.7	二维码、行程单、机动车销售发票、车辆合格证、试卷分析与识别、手写、护照、户口本、通用机打
2021.3.18	4.15.5	手写识别、二维码识别、试卷分析与识别、数字识别、办公文档识别、印章识别、仪器仪表盘读数识别、网络图片文字识别（含位置版）
2021.1.4	4.15.4	增加增值税发票识别，出租车票，vin码，火车票，数字识别
2018.4.9	0.6.0	新增表格识别同步接口
2018.1.12	0.5.0	新增自定义文字识别接口
2017.12.21	0.4.0	更改了网络图片文字识别的接口名称
2017.11.24	0.3.2	修复windows平台VC环境的编译错误
2017.11.9	0.3.0	初始化参数修改
2017.10.31	0.1.0	文字识别第一版

快速入门

安装文字识别 C++ SDK

文字识别 C++ SDK目录结构

```
├─── base
│   ├─── base.h                    // 请求客户端基类
│   ├─── base64.h                   // base64加密相关类
│   ├─── http.h                    // http请求封装类
│   └─── utils.h                   // 工具类
└─── ocr.h                  // 文字识别 交互类
```

最低支持 C++ 11+

直接使用开发包步骤如下：

1.在官方网站下载C++ SDK压缩包。

2.将下载的`aip-cpp-sdk-version.zip`解压, 其中文件为包含实现代码的头文件。

3.安装依赖库libcurl（需要支持https） openssl jsoncpp(>1.6.2版本，0.x版本将不被支持)。

4.编译工程时添加 C++11 支持 (gcc/clang 添加编译参数 -std=c++11), 添加第三方库链接参数 lcurl, lcrypto, ljsoncpp。

5.在源码中include ocr.h ，引入压缩包中的头文件以使用aip命名空间下的类和方法。

新建client

client是文字识别的C++客户端，为使用文字识别的开发人员提供了一系列的交互方法。当您引入了相应头文件后就可以新建一个client对象。

用户可以参考如下代码新建一个client：

```
 #include "ocr.h"

// 设置APPID/AK/SK
std::string app_id = "你的 App ID";
std::string api_key = "你的 Api key";
std::string secret_key = "你的 Secret Key";

aip::Ocr client(app_id, api_key, secret_key);
```

在上面代码中，常量`APP_ID`在百度智能云控制台中创建，常量`API_KEY`与`SECRET_KEY`是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的**应用列表**中查看。

接口说明

通用文字识别

用户向服务请求识别某张图中的所有文字

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用通用文字识别, 图片参数为本地图片
result = client.general_basic(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别, 图片参数为本地图片
result = client.general_basic(image, options);

Json::Value result;

std::string url = "https//www.x.com/sample.jpg";

// 调用通用文字识别, 图片参数为远程url图片
result = client.general_basic_url(url, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别, 图片参数为远程url图片
result = client.general_basic_url(url, options);
```

通用文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
url	是	std::string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	std::string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	std::string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	std::string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	std::string	true false		是否返回识别结果中每一行的置信度

通用文字识别 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别 返回示例

```
 {
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青岛啤酒"}
   ]
}
```

通用文字识别（高精度版）

用户向服务请求识别某张图中的所有文字，相对于通用文字识别该产品精度更高，但是识别耗时会稍长。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用通用文字识别（高精度版）
result = client.accurate_basic(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["detect_direction"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（高精度版）
result = client.accurate_basic(image, options);
```

通用文字识别（高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
detect_direction	否	std::string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
probability	否	std::string	true false		是否返回识别结果中每一行的置信度

通用文字识别（高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	定位和识别结果数组
+words	否	string	识别结果字符串
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+average	否	number	行置信度平均值
+variance	否	number	行置信度方差
+min	否	number	行置信度最小值

通用文字识别（高精度版） 返回示例

```
{
  "log_id": 1390582998516105216,
  "words_result_num": 2
  "words_result": [
     {
        "words": " OCR"
     },
     {
        "words": "百度通用文字识别高精度版"
     }
  ]
}
```

通用文字识别（含位置信息版）

用户向服务请求识别某张图中的所有文字，并返回文字在图中的位置信息。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用通用文字识别（含位置信息版），图片参数为本地图片
result = client.general(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["recognize_granularity"] = "big";
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["vertexes_location"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含位置信息版），图片参数为本地图片
result = client.general(image, options);

Json::Value result;

std::string url = "https//www.x.com/sample.jpg";

// 调用通用文字识别（含位置信息版），图片参数为远程url图片
result = client.general_url(url, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["recognize_granularity"] = "big";
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["vertexes_location"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含位置信息版），图片参数为远程url图片
result = client.general_url(url, options);
```

通用文字识别（含位置信息版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
url	是	std::string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
recognize_granularity	否	std::string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
language_type	否	std::string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	std::string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	std::string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
vertexes_location	否	std::string	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	std::string	true false		是否返回识别结果中每一行的置信度
paragraph	否	std::string	true false		是否输出段落信息

通用文字识别（含位置信息版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置信息版） 返回示例

```
 {
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
     "location": {
        "left": 35,
        "top": 53,
        "width": 193,
        "height": 109
     },
     "words": "感动",
     "chars": [    //recognize_granularity=small时存在
        {
           "location": {
              "left": 56,
              "top": 65,
              "width": 69,
              "height": 88
           },
           "char": "感"
        },
        {
           "location": {
              "left": 140,
              "top": 65,
              "width": 70,
              "height": 88
           },
           "char": "动"
        }
     ]
  }
  ...
]
}
```

通用文字识别（含位置高精度版）

用户向服务请求识别某张图中的所有文字，并返回文字在图片中的坐标信息，相对于通用文字识别（含位置信息版）该产品精度更高，但是识别耗时会稍长。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用通用文字识别（含位置高精度版）
result = client.accurate(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["recognize_granularity"] = "big";
options["detect_direction"] = "true";
options["vertexes_location"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含位置高精度版）
result = client.accurate(image, options);
```

通用文字识别（含位置高精度版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
recognize_granularity	否	std::string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	否	std::string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vertexes_location	否	std::string	true false	false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
probability	否	std::string	true false		是否返回识别结果中每一行的置信度

通用文字识别（含位置高精度版） 返回数据参数详情

字段	必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array	定位和识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+vertexes_location	否	array	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	number	水平坐标（坐标0点为左上角）
++y	是	number	垂直坐标（坐标0点为左上角）
+location	是	array	位置数组（坐标0点为左上角）
++left	是	number	表示定位位置的长方形左上顶点的水平坐标
++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
++width	是	number	表示定位位置的长方形的宽度
++height	是	number	表示定位位置的长方形的高度
+words	否	number	识别结果字符串
+chars	否	array	单字符结果，recognize_granularity=small时存在
++location	是	array	位置数组（坐标0点为左上角）
+++left	是	number	表示定位位置的长方形左上顶点的水平坐标
+++top	是	number	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	number	表示定位定位位置的长方形的宽度
+++height	是	number	表示位置的长方形的高度
++char	是	string	单字符识别结果
probability	否	object	行置信度信息；如果输入参数 probability = true 则输出
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含位置高精度版） 返回示例

```
{
"log_id": 3523983603,
"direction": 0, //detect_direction=true时存在
"words_result_num": 2,
"words_result": [
  {
    "location": {
      "left": 35,
      "top": 53,
      "width": 193,
      "height": 109
    },
    "words": "感动",
    "chars": [    //recognize_granularity=small时存在
      {
        "location": {
          "left": 56,
          "top": 65,
          "width": 69,
          "height": 88
        },
        "char": "感"
      },
      {
        "location": {
          "left": 140,
          "top": 65,
          "width": 70,
          "height": 88
        },
        "char": "动"
      }
    ]
  }
  ...
]
}
```

通用文字识别（含生僻字版）

【该服务已停止更新，如需更好的识别效果请使用通用文字识别（高精度版 / 高精度含位置版），此两项服务已扩充字库，可支持生僻字识别】字库范围更大，支持对图片中的生僻字进行识别

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用通用文字识别（含生僻字版），图片参数为本地图片
result = client.general_enhanced(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含生僻字版），图片参数为本地图片
result = client.general_enhanced(image, options);

Json::Value result;

std::string url = "https//www.x.com/sample.jpg";

// 调用通用文字识别（含生僻字版），图片参数为远程url图片
result = client.general_enhanced_url(url, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["language_type"] = "CHN_ENG";
options["detect_direction"] = "true";
options["detect_language"] = "true";
options["probability"] = "true";

// 带参数调用通用文字识别（含生僻字版），图片参数为远程url图片
result = client.general_enhanced_url(url, options);
```

通用文字识别（含生僻字版） 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
url	是	std::string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
language_type	否	std::string	CHN_ENG ENG POR FRE GER ITA SPA RUS JAP KOR	CHN_ENG	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语； - KOR：韩语；
detect_direction	否	std::string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	std::string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
probability	否	std::string	true false		是否返回识别结果中每一行的置信度

通用文字识别（含生僻字版） 返回数据参数详情

字段	是否必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

通用文字识别（含生僻字版） 返回示例

```
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
  [
    {"words": " TSINGTAO"},
    {"words": "青島啤酒"}
  ]
}
```

网络图片文字识别

用户向服务请求识别一些网络上背景复杂，特殊字体的文字。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用网络图片文字识别, 图片参数为本地图片
result = client.webimage(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["detect_direction"] = "true";
options["detect_language"] = "true";

// 带参数调用网络图片文字识别, 图片参数为本地图片
result = client.webimage(image, options);

Json::Value result;

std::string url = "https//www.x.com/sample.jpg";

// 调用网络图片文字识别, 图片参数为远程url图片
result = client.webimage_url(url, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["detect_direction"] = "true";
options["detect_language"] = "true";

// 带参数调用网络图片文字识别, 图片参数为远程url图片
result = client.webimage_url(url, options);
```

网络图片文字识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
url	是	std::string			图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，当image字段存在时url字段失效
detect_direction	否	std::string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	否	std::string	true false	false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

网络图片文字识别 返回数据参数详情

字段	是否必选	类型	说明
direction	否	number	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	number	唯一的log id，用于问题定位
words_result	是	array()	识别结果数组
words_result_num	是	number	识别结果数，表示words_result的元素个数
+words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+ average	否	number	行置信度平均值
+ variance	否	number	行置信度方差
+ min	否	number	行置信度最小值

网络图片文字识别 返回示例

```
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
  [
    {"words": " TSINGTAO"},
    {"words": "青岛啤酒"}
  ]
}
```

身份证识别

用户向服务请求识别身份证，身份证识别包括正面和背面。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

std::string id_card_side = "back";

// 调用身份证识别
result = client.idcard(image, id_card_side, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["detect_direction"] = "true";
options["detect_risk"] = "false";

// 带参数调用身份证识别
result = client.idcard(image, id_card_side, options);
```

身份证识别 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
id_card_side	是	string	front/back	-front：身份证含照片的一面 -back：身份证带国徽的一面 自动检测身份证正反面，如果传参指定方向与图片相反，支持正常识别，返回参数image_status字段为"reversed_side"
detect_risk	否	string	true/false	是否开启身份证风险类型(身份证复印件、临时身份证、身份证翻拍、修改过的身份证)检测功能，默认不开启，即：false。 - true：开启，请查看返回参数risk_type； - false：不开启
detect_quality	否	string	true/false	是否开启身份证质量类型(边框/四角不完整、头像或关键字段被遮挡/马赛克)检测功能，默认不开启，即：false。 - true：开启，请查看返回参数card_quality； - false：不开启
detect_photo	否	string	true/false	是否检测头像内容，默认不检测。可选值：true-检测头像并返回头像的 base64 编码及位置信息
detect_card	否	string	true/false	是否检测身份证进行裁剪，默认不检测。可选值：true-检测身份证并返回证照的 base64 编码及位置信息

身份证识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
direction	是	int32	图像方向。 - - 1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
image_status	是	string	normal-识别正常 reversed_side-身份证正反面颠倒 non_idcard-上传的图片中不包含身份证 blurred-身份证模糊 other_type_card-其他类型证照 over_exposure-身份证关键字段反光或过曝 over_dark-身份证欠曝（亮度过低） unknown-未知状态

risk_type	否	string	输入参数 detect_risk = true 时，则返回该字段识别身份证**风险类型**: normal-正常身份证； copy-复印件； temporary-临时身份证； screen-翻拍； unknown-其他未知情况
edit_tool	否	string	如果参数 detect_risk = true 时，则返回此字段。如果检测身份证被编辑过，该字段指定编辑软件名称，如:Adobe Photoshop CC 2014 (Macintosh),如果没有被编辑过则返回值无此参数
card_quality	否	object	输入参数 detect_quality = true 时，则返回该字段识别身份证**质量类型**: IsClear - 是否清晰； IsComplete - 是否边框/四角完整； IsNoCover - 是否头像、关键字段无遮挡/马赛克。 及对应的概率：IsComplete_propobility、IsNoCover_propobility、IsClear_propobility，值在0-1之间，值越大表示图像质量越好。 **默认阈值：**当 IsComplete_propobility 超过0.5时，IsComplete返回1，低于0.5，则返回0。 IsNoCover_propobility、IsClear_propobility 同上
photo	否	string	当请求参数 detect_photo = true时返回，头像切图的 base64 编码（无编码头，需自行处理）
photo_location	否	object	当请求参数 detect_photo = true时返回，头像的位置信息（坐标0点为左上角）
card_image	否	string	当请求参数 detect_card = true时返回，身份证裁剪切图的 base64 编码（无编码头，需自行处理）
card_location	否	object	当请求参数 detect_card = true时返回，身份证裁剪切图的位置信息（坐标0点为左上角）
idcard_number_type	是	int	用于校验身份证号码、性别、出生是否一致，输出结果及其对应关系如下： - **-1：** 身份证正面所有字段全为空 **0：** 身份证证号不合法，此情况下不返回身份证证号 **1：** 身份证证号和性别、出生信息一致 **2：** 身份证证号和性别、出生信息都不一致 **3：** 身份证证号和出生信息不一致 **4：** 身份证证号和性别信息不一致
+ location	是	array[]	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	否	string	识别结果字符串

身份证识别 返回示例


```
{
  "log_id": 2648325511,
  "direction": 0,
  "image_status": "normal",
  "idcard_type": "normal",
  "edit_tool": "Adobe Photoshop CS3 Windows",
  "words_result": {
    "住址": {
      "location": {
        "left": 267
```

```
            "left": 267,
            "top": 453,
            "width": 459,
            "height": 99
        },
        "words": "南京市江宁区弘景大道3889号"
    },
    "公民身份号码": {
        "location": {
            "left": 443,
            "top": 681,
            "width": 589,
            "height": 45
        },
        "words": "330881199904173914"
    },
    "出生": {
        "location": {
            "left": 270,
            "top": 355,
            "width": 357,
            "height": 45
        },
        "words": "19990417"
    },
    "姓名": {
        "location": {
            "left": 267,
            "top": 176,
            "width": 152,
            "height": 50
        },
        "words": "伍云龙"
    },
    "性别": {
        "location": {
            "left": 269,
            "top": 262,
            "width": 33,
            "height": 52
        },
        "words": "男"
    },
    "民族": {
        "location": {
            "left": 492,
            "top": 279,
            "width": 30,
            "height": 37
        },
        "words": "汉"
    }
},
"words_result_num": 6
}
```


银行卡识别

识别银行卡并返回卡号和发卡行。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用银行卡识别
result = client.bankcard(image, aip::null);
```

银行卡识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取

银行卡识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	请求标识码，随机数，唯一。
result	object	是	返回结果
+bank_card_number	string	是	银行卡卡号
+bank_name	string	是	银行名，不能识别时为空
+bank_card_type	number	是	银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡

银行卡识别 返回示例

```
{
  "log_id": 1447188951,
  "result": {
     "bank_card_number": "622500000000000",
     "bank_name": "招商银行",
     "bank_card_type": 1
  }
}
```

驾驶证识别

对机动车驾驶证所有关键字段进行识别

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用驾驶证识别
result = client.driving_license(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["detect_direction"] = "true";

// 带参数调用驾驶证识别
result = client.driving_license(image, options);
```

驾驶证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
detect_direction	否	std::string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

驾驶证识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
+words	否	string	识别结果字符串

驾驶证识别 返回示例

```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "证号": {
    "words": "3208231999053090"
   },
   "有效期限": {
    "words": "6年"
   },
   "准驾车型": {
    "words": "B2"
   },
   "有效起始日期": {
    "words": "20101125"
   },
   "住址": {
    "words": "江苏省南通市海门镇秀山新城"
   },
   "姓名": {
    "words": "小欧欧"
   },
   "国籍": {
    "words": "中国"
   },
   "出生日期": {
    "words": "19990530"
   },
   "性别": {
    "words": "男"
   },
   "初次领证日期": {
    "words": "20100125"
   }
  }
 }
}
```


行驶证识别

对机动车行驶证所有关键字段进行识别

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用行驶证识别
result = client.vehicle_license(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["detect_direction"] = "true";
options["accuracy"] = "normal";

// 带参数调用行驶证识别
result = client.vehicle_license(image, options);
```

行驶证识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
detect_direction	否	std::string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
vehicle_license_side	否	string	front/back	front	- **front**：识别行驶证主页 - **back**：识别行驶证副页
unified	否	string	true/false	false	- **false**：不进行归一化处理 - **true**：对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出

行驶证识别 返回数据参数详情

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
+words	否	string	识别结果字符串

行驶证识别 返回示例

```
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "品牌型号": {
    "words": "保时捷GT37182RUCRE"
   },
   "发证日期": {
    "words": "20160104"
   },
   "使用性质": {
    "words": "非营运"
   },
   "发动机号码": {
    "words": "20832"
   },
   "号牌号码": {
    "words": "苏A001"
   },
   "所有人": {
    "words": "圆圆"
   },
   "住址": {
    "words": "南京市江宁区弘景大道"
   },
   "注册日期": {
    "words": "20160104"
   },
   "车辆识别代号": {
    "words": "HCE58"
   },
   "车辆类型": {
    "words": "小型轿车"
   }
  }
 }
}
```


车牌识别

识别机动车车牌，并返回签发地和号牌。

```
  Json::Value result;

 std::string image;
 aip::get_file_content("/assets/sample.jpg", &image);

 // 调用车牌识别
 result = client.license_plate(image, aip::null);

 // 如果有可选参数
 std::map<std::string, std::string> options;
 options["multi_detect"] = "true";

 // 带参数调用车牌识别
 result = client.license_plate(image, options);
```

车牌识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
multi_detect	否	std::string	true false	false	是否检测多张车牌，默认为false，当置为true的时候可以对一张图片内的多张车牌进行识别

车牌识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
Color	string	是	车牌颜色
number	string	是	车牌号码

车牌识别 返回示例

```
{
  "log_id": 3583925545,
  "words_result": {
     "color": "blue",
     "number": "苏HS7766"
  }
}
```

营业执照识别

识别营业执照，并返回关键字段的值，包括单位名称、法人、地址、有效期、证件编号、社会信用代码等。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用营业执照识别
result = client.business_license(image, aip::null);
```

营业执照识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取
detect_direction	否	std::string	此参数新版本无需传，支持自动检测图像旋转角度；朝向是指输入图像是正常方向、逆时针旋转90/180/270度
accuracy	否	std::string	此参数新版本无需传，可选值：normal,high
risk_warn	否	std::string	是否开启风险类型功能，默认不开启，即：false。false：不开启 true：开启

营业执照识别 返回数据参数详情

参数	是否必须	类型	说明
log_id	是	number	请求标识码，随机数，唯一。
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array(object)	识别结果数组
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	否	string	识别结果字符串

营业执照识别 返回示例

```
{
  "log_id": 490058765,
  "words_result": {
    "单位名称": {
      "location": {
        "left": 500,
        "top": 479,
        "width": 618,
        "height": 54
      },
      "words": "袁氏财团有限公司"
    },
    "法人": {
      "location": {
        "left": 938,
        "top": 557,
        "width": 94,
        "height": 46
      },
      "words": "袁运筹"
    },
    "地址": {
      "location": {
        "left": 503,
        "top": 644,
        "width": 574,
        "height": 57
      },
      "words": "江苏省南京市中山东路19号"
    },
    "有效期": {
      "location": {
        "left": 779,
        "top": 1108,
        "width": 271,
        "height": 49
      },
      "words": "2015年02月12日"
    },
    "证件编号": {
      "location": {
        "left": 1219,
        "top": 357,
        "width": 466,
        "height": 39
      },
      "words": "苏餐证字(2019)第666602666661号"
    },
    "社会信用代码": {
      "location": {
        "left": 0,
        "top": 0,
        "width": 0,
        "height": 0
      },
      "words": "无"
    }
  },
  "words_result_num": 6
}
```

通用票据识别

用户向服务请求识别医疗票据、发票、的士票、保险保单等票据类图片中的所有文字，并返回文字在图中的位置信息。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用通用票据识别
result = client.receipt(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["recognize_granularity"] = "big";
options["probability"] = "true";
options["accuracy"] = "normal";
options["detect_direction"] = "true";

// 带参数调用通用票据识别
result = client.receipt(image, options);
```

通用票据识别 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
recognize_granularity	否	std::string	big - 不定位单字符位置 small - 定位单字符位置	small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	std::string	true false		是否返回识别结果中每一行的置信度
accuracy	否	std::string	normal - 使用快速服务		normal 使用快速服务，1200ms左右时延；缺省或其它值使用高精度服务，1600ms左右时延
detect_direction	否	std::string	true false	false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

通用票据识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位位置的长方形的宽度
height	是	number	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array()	单字符结果，recognize_granularity=small时存在
location	是	array()	位置数组（坐标0点为左上角）
left	是	number	表示定位位置的长方形左上顶点的水平坐标
top	是	number	表示定位位置的长方形左上顶点的垂直坐标
width	是	number	表示定位定位位置的长方形的宽度
height	是	number	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值

通用票据识别 返回示例

```
{
  "log_id": 2661573626,
  "words_result": [
    {
      "location": {
        "left": 10,
        "top": 3,
        "width": 121,
        "height": 24
      },
      "words": "姓名:小明明",
      "chars": [
        {
          "location": {
            "left": 16,
            "top": 6,
            "width": 17,
            "height": 20
          },
          "char": "姓"
        }
        ...
      ]
    },
    {
      "location": {
        "left": 212,
        "top": 3,
        "width": 738,
        "height": 24
      },
      "words": "卡号/病案号:105353990标本编号:150139071送检科室:血液透析门诊病房",
      "chars": [
        {
          "location": {
            "left": 218,
            "top": 6,
            "width": 18,
            "height": 21
          },
          "char": "卡"
        }
        ...
      ]
    }
  ],
  "words_result_num": 2
}
```


自定义模板文字识别

自定义模板文字识别，是针对百度官方没有推出相应的模板，但是当用户需要对某一类卡证/票据（如房产证、军官证、火车票等）进行结构化的提取内容时，可以使用该产品快速制作模板，进行识别。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

std::string templateSign = "Nsdax2424asaAS791823112";

// 调用自定义模板文字识别
result = client.custom(image, templateSign, aip::null);
```

自定义模板文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取
templateSign	否	std::string	您在自定义文字识别平台制作的模板的ID
classifierId	否	std::string	分类器Id。这个参数和templateSign至少存在一个，优先使用templateSign。存在templateSign时，表示使用指定模板；如果没有templateSign而有classifierId，表示使用分类器去判断使用哪个模板

自定义模板文字识别 返回数据参数详情

字段	是否必选	类型	说明
error_code	number	number	0代表成功，如果有错误码返回可以参考下方错误码列表排查问题
error_msg	是	string	具体的失败信息，可以参考下方错误码列表排查问题
data	jsonObject	识别返回的结果	

自定义模板文字识别 返回示例

```
{
  "isStructured": true,
  "ret": [
    {
      "charset": [
        {
          "rect": {
            "top": 183,
            "left": 72,
            "width": 14,
            "height": 28
          },
          "word": "5"
        },
        {
          "rect": {
            "top": 183,
            "left": 90,
            "width": 14,
            "height": 28
          },
          "word": "4"
        },
        {
          "rect": {
            "top": 183,
            "left": 103,
```

```
                "width": 15,
                "height": 28
            },
            "word": "."
        },
        {
            "rect": {
                "top": 183,
                "left": 116,
                "width": 14,
                "height": 28
            },
            "word": "5"
        },
        {
            "rect": {
                "top": 183,
                "left": 133,
                "width": 19,
                "height": 28
            },
            "word": "元"
        }
    ],
    "word_name": "票价",
    "word": "54.5元"
},
{
    "charset": [
        {
            "rect": {
                "top": 144,
                "left": 35,
                "width": 14,
                "height": 28
            },
            "word": "2"
        },
        {
            "rect": {
                "top": 144,
                "left": 53,
                "width": 14,
                "height": 28
            },
            "word": "0"
        },
        {
            "rect": {
                "top": 144,
                "left": 79,
                "width": 14,
                "height": 28
            },
            "word": "1"
        },
        {
            "rect": {
                "top": 144,
                "left": 97,
                "width": 14,
                "height": 28
```

```
            },
            "word": "7"
        }
    ]
  ]
}
```

增值税发票识别

支持对增值税普票、专票、卷票、电子发票、区块链发票的所有字段进行结构化识别，包括发票基本信息、销售方及购买方信息、商品信息、价税信息等，其中五要素字段的识别准确率超过 99.9%； 同时，支持对增值税卷票的 21 个关键字段进行识别，包括发票类型、发票代码、发票号码、机打号码、机器编号、收款人、销售方名称、销售方纳税人识别号、开票日期、购买方名称、购买方纳税人识别号、项目、单价、数量、金额、税额、合计金额(小写)、合计金额(大写)、校验码、省、市，四要素字段的识别准确率可达95%。

```
#include "ocr.h"

// 设置APPID/AK/SK
std::string app_id = "你的 App ID";
std::string api_key = "你的 Api key";
std::string secret_key = "你的 Secret Key";

aip::Ocr client(app_id, api_key, secret_key);
Json::Value result;
// 如果有可选参数
std::map<std::string, std::string> options;
std::string image;
//图片识别
aip::get_file_content("./sample.jpg", &image);
result = client.vatInvoice(image, options);
//url识别
result = client.vatInvoiceUrl("htpp://test.jpg", options);
 //pdf识别
aip::get_file_content("./pdf.jpg", &image);
result = client.vatInvoicePdf(image, options);
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url或者pdf文件
type	否	String	可选参数，进行识别的增值税发票类型，默认为 normal，可缺省normal：可识别增值税普票、专票、电子发票roll：可识别增值税卷票

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	是	string	发票种类
InvoiceTypeOrg	是	string	发票名称
InvoiceCode	是	string	发票代码
InvoiceNum	是	string	发票号码
MachineNum	是	string	机打号码。仅增值税卷票含有此参数
MachineCode	是	string	机器编号。仅增值税卷票含有此参数

CheckCode	否	string	校验码。增值税专票无此参数
InvoiceDate	是	string	开票日期
PurchaserName	是	string	购方名称
PurchaserRegisterNum	是	string	购方纳税人识别号
PurchaserAddress	是	string	购方地址及电话
PurchaserBank	是	string	购方开户行及账号
Password	是	string	密码区
Province	是	string	省
City	是	string	市
SheetNum	是	string	联次
Agent	是	string	是否代开
CommodityName	是	array[]	货物名称
- row	是	uint32	行号
- word	是	string	内容
CommodityType	是	array[]	规格型号
- row	是	uint32	行号
- word	是	string	内容
CommodityUnit	是	array[]	单位
- row	是	uint32	行号
- word	是	string	内容
CommodityNum	是	array[]	数量
- row	是	uint32	行号
- word	是	string	内容
CommodityPrice	是	array[]	单价
- row	是	uint32	行号
- word	是	string	内容
CommodityAmount	是	array[]	金额
- row	是	uint32	行号
- word	是	string	内容
CommodityTaxRate	是	array[]	税率
- row	是	uint32	行号
- word	是	string	内容
CommodityTax	是	array[]	税额
- row	是	uint32	行号
- word	是	string	内容
SellerName	是	string	销售方名称
SellerRegisterNum	是	string	销售方纳税人识别号
SellerAddress	是	string	销售方地址及电话
SellerBank	是	string	销售方开户行及账号
TotalAmount	是	uint32	合计金额

TotalAmount	是	uint32	合计金额
TotalTax	是	uint32	合计税额
AmountInWords	是	string	价税合计(大写)
AmountInFiguers	是	uint32	价税合计(小写)
Payee	是	string	收款人
Checker	是	string	复核
NoteDrawer	是	string	开票人
Remarks	是	string	备注

返回示例


```
{
 "log_id": "5425496231209218858",
"words_result_num": 29,
"words_result": {
 "InvoiceNum": "14641426",
 "SellerName": "上海易火广告传媒有限公司",
 "CommodityTaxRate": [
  {
   "word": "6%",
   "row": "1"
  }
 ],
 "SellerBank": "中国银行南翔支行446863841354",
 "Checker": ":沈园园",
 "TotalAmount": "94339.62",
 "CommodityAmount": [
  {
   "word": "94339.62",
   "row": "1"
  }
 ],
 "InvoiceDate": "2016年06月02日",
 "CommodityTax": [
  {
   "word": "5660.38",
   "row": "1"
  }
 ],
 "PurchaserName": "百度时代网络技术(北京)有限公司",
 "CommodityNum": [
  {
   "word": "",
   "row": "1"
  }
 ],
      "Province": "上海",
      "City": "",
      "SheetNum": "第三联",
      "Agent": "否",
 "PurchaserBank": "招商银行北京分行大屯路支行8661820285100030",
 "Remarks": "告传",
 "Password": "074/45781873408>/6>8>65*887676033/51+<5415>9/32--852>1+29<65>641-
5>66<500>87/*-34<943359034>716905113*4242>",
 "SellerAddress": ":嘉定区胜辛南路500号15幢1161室55033753",
 "PurchaserAddress": "北京市海淀区东北旺西路8号中关村软件园17号楼二属A2010-59108001",
 "InvoiceCode": "3100153130",
 "CommodityUnit": [
  {
```

```
    "word": "",
    "row": "1"
   }
  ],
  "Payee": ":徐蓉",
  "PurchaserRegisterNum": "110108787751579",
  "CommodityPrice": [
   {
    "word": "",
    "row": "1"
   }
  ],
  "NoteDrawer": "沈园园",
  "AmountInWords": "壹拾万圆整",
  "AmountInFiguers": "100000.00",
  "TotalTax": "5660.38",
  "InvoiceType": "专用发票",
  "SellerRegisterNum": "913101140659591751",
  "CommodityName": [
   {
    "word": "信息服务费",
    "row": "1"
   }
  ],
  "CommodityType": [
   {
    "word": "",
    "row": "1"
   }
  ]
 }
}
```

出租车票识别

支持识别全国各大城市出租车票的 16 个关键字段，包括发票号码、代码、车号、日期、总金额、燃油附加费、叫车服务费、省、市、单价、里程、上车时间、下车时间等。

```
    #include "ocr.h"

  // 设置APPID/AK/SK
  std::string app_id = "你的 App ID";
  std::string api_key = "你的 Api key";
  std::string secret_key = "你的 Secret Key";

  aip::Ocr client(app_id, api_key, secret_key);
  Json::Value result;
     // 如果有可选参数
  std::map<std::string, std::string> options;

  std::string image;
  //图片识别
  aip::get_file_content("./sample.jpg", &image);
  result = client.taxiReceipt(image, options);
  //url识别
  result = client.taxiReceiptUrl("htpp://test.jpg", options);
  **请求参数详情**
```

参数名称	是否必选	类型	说明
image	是	mixed	本地图片二进制数据或url

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
words_result_num	uint32	是	识别结果数，表示words_result的元素个数
InvoiceCode	string	是	发票代号
InvoiceNum	string	是	发票号码
TaxiNum	string	是	车牌号
Date	string	是	日期
Time	string	是	上下车时间
Fare	string	是	总金额
FuelOilSurcharge	string	是	燃油附加费
CallServiceSurcharge	string	是	叫车服务费
Province	string	是	省
City	string	是	市
PricePerkm	string	是	单价
Distance	string	是	里程

返回示例


```
{
"log_id":2034039896,
"words_result_num":6,
"words_result":
{
 "Date":"2017-11-26",
 "Fare":"¥153.30元",
 "InvoiceCode":"111001681009",
 "InvoiceNum":"90769610",
 "TaxiNum":"BV2062",
 "Time":"20:42-21:07",
 "FuelOilSurcharge": "¥0.00",
 "CallServiceSurcharge": "¥0.00",
      "Province": "浙江省",
      "City": "杭州市",
      "PricePerkm": "2.50元/KM",
      "Distance": "4.5KM"
}
}
```


VIN码识别

支持对车辆挡风玻璃处的车架号码进行识别。

```
    #include "ocr.h"

  // 设置APPID/AK/SK
  std::string app_id = "你的 App ID";
  std::string api_key = "你的 Api key";
  std::string secret_key = "你的 Secret Key";

  aip::Ocr client(app_id, api_key, secret_key);
  Json::Value result;
     // 如果有可选参数
  std::map<std::string, std::string> options;
  std::string image;
  //图片识别
  aip::get_file_content("./sample.jpg", &image);
  result = client.vinCode(image, options);
  //url识别
  result = client.vinCodeUrl("htpp://test.jpg", options);
  **请求参数详情**
```

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array[]	定位和识别结果数组
location	是	object{}	识别结果
words	是	string	VIN码识别结果
words_result_num	是	int	识别结果数，表示words_result的元素个数

返回示例

```
{
"log_id": 246589877,
"words_result": [
 {
  "location": {
   "left": 124,
   "top": 11,
   "width": 58,
   "height": 359
  },
  "words": "LFV2A11K8D4010942"
 }
],
"words_result_num": 1
}
```

火车票识别

支持对红、蓝火车票的13个关键字段进行结构化识别，包括车票号码、始发站、目的站、车次、日期、票价、席别、姓名、座位号、身份证号、售站、序列号、时间。

```
    #include "ocr.h"

// 设置APPID/AK/SK
std::string app_id = "你的 App ID";
std::string api_key = "你的 Api key";
std::string secret_key = "你的 Secret Key";

aip::Ocr client(app_id, api_key, secret_key);
Json::Value result;
    // 如果有可选参数
std::map<std::string, std::string> options;
std::string image;
//图片识别
aip::get_file_content("./sample.jpg", &image);
result = client.trainTicket(image, options);
//url识别
result = client.trainTicketUrl("htpp://test.jpg", options);
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据或url

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	请求标识码，随机数，唯一。
ticket_num	string	是	车票号
starting_station	string	是	始发站
train_num	string	是	车次号
destination_station	string	是	到达站
date	string	是	出发日期
ticket_rates	string	是	车票金额
seat_category	string	是	席别
name	string	是	乘客姓名
id_num	string	是	身份证号
serial_number	string	是	序列号
sales_station	string	是	售站
time	string	是	时间
seat_num	string	是	座位号

返回示例

```
{
"log_id": "12317512659",
"direction": 1,
"words_result_num": 13,
"words_result": {
 "id_num": "2302051998****156X",
 "name": "裴一丽",
 "ticket_rates": "￥54.5元",
 "destination_station": "天津站",
 "seat_category": "二等座",
 "sales_station": "北京南",
 "ticket_num": "F05706",
 "seat_num": "02车03C号",
 "time": "09:36",
 "date": "2019年04月03日",
 "serial_number": "10010300067846",
 "train_num": "C255",
 "starting_station": "北京南站"
}
}
```


数字识别

对图片中的数字进行提取和识别，自动过滤非数字内容，仅返回数字内容及其位置信息，识别准确率超过99%。

```
    #include "ocr.h"

  // 设置APPID/AK/SK
  std::string app_id = "你的 App ID";
  std::string api_key = "你的 Api key";
  std::string secret_key = "你的 Secret Key";

  aip::Ocr client(app_id, api_key, secret_key);
  Json::Value result;
     // 如果有可选参数
  std::map<std::string, std::string> options;
  std::string image;
  //图片识别
  aip::get_file_content("./sample.jpg", &image);
  result = client.numbers(image, options);
```

请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据
recognize_granularity	false	string	big、small
detect_direction	false	string	true、false

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果

返回示例

```
{
  "log_id": 620759800,
  "words_result": [
     {
        "location": {
           "left": 56,
           "top": 0,
           "width": 21,
           "height": 210
        },
        "words": "3"
     }
  ],
  "words_result_num": 1
}
```

印章识别

检测并识别合同文件或常用票据中的印章，输出文字内容、印章位置信息以及相关置信度，已支持圆形章、椭圆形章、方形章等常见印章检测与识别

```
  Json::Value result;;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 印章识别
result = client.seal(image,aip::null);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
result_num	是	uint32	识别结果数，表示results的元素个数
result	是	array[]	定位结果数组
+location	是	object{}	位置数组（坐标0点为左上角）
++left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	是	uint32	表示定位位置的长方形的宽度
++height	是	uint32	表示定位位置的长方形的高度
+probability	是	float	每一个识别结果的置信度值
+type	是	string	印章的类别，共有circle（圆章），ellipse（椭圆章），rectangle（方章）三种
+major	是	object{}	主字段内容
++words	是	string	主字段识别内容，即章内上环弯曲文字结果
++probability	是	float	主字段识别内容的置信度
+minor	是	array[]	其他字段内容，即除主字段外的文字识别内容均放置于该参数中返回，若章内不存在其他字段文字，则该参数为空
++words	是	string	其他字段识别内容
++probability	是	float	其他字段识别内容的置信度

返回示例

```
{
"result": [
{
 "major": {
  "probability": 0.99759155511856,
  "words": "峨眉山旅游股份有限公司成都峨眉山雪芽大酒店分公司"
 },
 "minor": [
  {
   "probability": 0.99994027614594,
   "words": "前厅部"
  }
 ],
 "probability": 0.9936261177063,
 "location": {
  "top": 594,
  "left": 918,
  "width": 150,
  "height": 142
 },
 "type": "circle"
}
],
"log_id": "1349006147834609664",
"result_num": 1
}
```


网络图片文字识别（含位置版）

支持识别艺术字体或背景复杂的文字内容，除文字信息外，还可返回每行文字的位置信息、行置信度，以及单字符内容和位置等。

```
  Json::Value result;;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 网络图片文字识别（含位置版）
result = client.webimageloc(image,aip::null);
// 文件url
std::string = "http://host/test.jpeg"
result = client.webimagelocurl(url,aip::null);
// 如果有可选参数
std::map<std::string, std::string> options;
// 带参数调用网络图片文字识别（含位置版）
result = client.webimageloc(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，像素乘积不超过2048*2048（1024*1024以内图像处理效果最佳）。**注意：图片的base64编码是不包含图片头的，如（）**
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
poly_location	false	string	true/false	是否返回文字所在区域的外接四边形的4个点坐标信息。默认为false
recognize_granularity	false	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
direction	false	int32	图像方向，当detect_direction=true时存在。检测到的图像朝向： 0 ：正向； 1：逆时针旋转90度； 2：逆时针旋转180度； 3：逆时针旋转270度
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	整行的识别结果
+location	true	object	整行的矩形框坐标。位置数组（坐标0点为左上角）
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	true	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	true	array[]	poly_location=true时存在。文字所在区域的外接矩形的4个点坐标信息
++x	true	uint32	水平坐标（坐标0点为左上角）
++y	true	uint32	垂直坐标（坐标0点为左上角）
+chars	false	array[]	单字符结果，recognize_granularity=small时存在
++char	false	string	单字符识别结果
++location	false	object	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示定位定位位置的长方形的高度

返回示例

```
{
  "log_id": 1390656223866519552,
  "words_result_num": 3,
  "words_result": [
    {
      "words": "梦想起航",
      "location": {
        "top": 328,
        "left": 1079,
        "width": 56,
        "height": 262
      },
    },
    {
      "words": "前往下一个目的地",
      "location": {
        "top": 329,
        "left": 1160,
        "width": 63,
        "height": 446
      },
    },
    {
      "words": "开始新的旅程",
      "location": {
        "top": 455,
        "left": 1246,
        "width": 63,
        "height": 340
      },
    }
  ],
}
```


仪器仪表盘读数识别

适用于不同品牌、不同型号的仪器仪表盘读数识别，广泛适用于各类血糖仪、血压仪、燃气表、电表等，可识别表盘上的数字、英文、符号，支持液晶屏、字轮表等表型。

```
 Json::Value result;;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 仪器仪表盘读数识别
result = client.meter(image,aip::null);
// 文件url
std::string = "http://host/test.jpeg"
result = client.meterurl(url,aip::null);
// 如果有可选参数
std::map<std::string, std::string> options;
// 带参数调用仪器仪表盘读数识别
result = client.meter(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px。支持jpg/jpeg/png/bmp格式.**注意：图片的base64编码是不包含图片头的，如（）**
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
poly_location	false	string	true/false	位置信息返回形式，默认：false false：只给出识别结果所在长方形位置信息 true：除了默认的识别文字所在长方形的位置信息，还会给出文字所在区域的最小外接旋转矩形的4个点坐标信息

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
words_result	true	array[]	识别结果数组
words_result_num	true	uint32	识别结果数，表示words_result的元素个数
+words	true	string	识别结果字符串
+location	true	array[]	识别结果所在长方形位置信息
++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	true	uint32	表示定位位置的长方形的宽度
++height	true	uint32	表示定位位置的长方形的高度
+probability	false	string	probability=true时存在。识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值
+poly_location	false	array[]	poly_location=true时存在。文字所在区域的外接四边形的4个点坐标信息

返回示例

```
{
  "log_id": "1392680790663364608",
"words_result_num": 5
"words_result": [
{
 "words": "5.8",
 "location": {
  "top": 150,
  "left": 370,
  "width": 87,
  "height": 79
 }
},
{
 "words": "mmol/L",
 "location": {
  "top": 241,
  "left": 402,
  "width": 52,
  "height": 12
 }
},
{
 "words": "10:38",
 "location": {
  "top": 115,
  "left": 347,
  "width": 42,
  "height": 21
 }
},
{
 "words": "12-11",
 "location": {
  "top": 116,
  "left": 410,
  "width": 36,
  "height": 20
 }
},
{
 "words": "am",
 "location": {
  "top": 115,
  "left": 391,
  "width": 12,
  "height": 5
 }
}
],
}
```


试卷分析与识别

可对文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景

```
 Json::Value result;;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 试卷分析与识别
result = client.docanalysis(image,aip::null);
// 文件url
std::string = "http://host/test.jpeg"
result = client.docanalysisurl(url,aip::null);
// 如果有可选参数
std::map<std::string, std::string> options;
// 带参数调用试卷分析与识别
result = client.docanalysis(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少64px，最长边最大4096px。**注意：图片的base64编码是不包含图片头的，如（）**
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
language_type	false	string	CHN_ENG/ENG	识别语言类型，默认为CHN_ENG 可选值包括： =CHN_ENG：中英文 =ENG：英文
result_type	false	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
words_type	false	string	handwring_only/handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	false	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
img_direction	false	int32	detect_direction=true时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	true	uint32	识别结果数，表示results的元素个数
results	true	array[]	识别结果数组
+words_type	true	string	文字属性（手写、印刷）,handwriting 手写，print 印刷
+words	true	array[]	整行的识别结果数组。
++line_probability	false	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++average	false	float	行置信度
+++min	false	float	整行中单字的最低置信度
++word	true	float	整行的识别结果
++words_location	true	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	true	uint32	表示定位定位位置的长方形的宽度
+++height	true	uint32	表示位置的长方形的高度
+chars	false	array[]	result_type=small时返回。单字符结果数组。
++char	false	string	result_type=small时返回。每个单字的内容。
++chars_location	false	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示位置的长方形的高度
layouts_num	false	uint32	版面分析结果数，表示layout的元素个数
layouts	false	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+layout	false	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title
+layout_location	false	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++x	false	uint32	水平坐标（坐标0点为左上角）
++y	false	uint32	水平坐标（坐标0点为左上角）
+layout_idx	false	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）

返回示例

```
{
"results_num": 6,
"log_id": "4488766695474114139",
"img_direction": 0,
"layouts_num": 0,
"results": [
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 124,
    "left": 136,
    "width": 418,
    "height": 65
   },
   "word": "五默写(4分)"
  },
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 246,
    "left": 136,
    "width": 37,
    "height": 45
   },
   "word": "1"
  },
 },
 {
  "words_type": "handwriting",
  "words": {
   "words_location": {
    "top": 195,
    "left": 237,
    "width": 469,
    "height": 104
   },
   "word": "采菊东篱下"
  },
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 241,
    "left": 889,
    "width": 287,
    "height": 52
   },
   "word": "悠然见南山?"
  },
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 415,
    "left": 134,
    "width": 472,
    "height": 52
   },
```

```
    },
    "word": "2.商女不知亡国恨"
   },
  },
  {
   "words_type": "handwriting",
   "words": {
    "words_location": {
     "top": 377,
     "left": 607,
     "width": 556,
     "height": 93
    },
    "word": "隔江犹唱后庭花。"
   },
  },
 ]
}
```

手写文字识别

支持对图片中的手写中文、手写数字进行检测和识别，针对不规则的手写字体进行专项优化，识别准确率可达90%以上。

```
  Json::Value result;;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 手写文字识别
result = client.handwriting(image,aip::null);
// 如果有可选参数
std::map<std::string, std::string> options;
// 带参数调用手写文字识别
result = client.handwriting(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
recognize_granularity	否	string	big、small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true/false	是否返回识别结果中每一行的置信度，默认为false，不返回置信度
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: true：检测朝向； false：不检测朝向

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array[]	单字符结果，recognize_granularity=small时存在
location	是	object{}	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	float	当请求参数 **probability=true** 时返回该字段，表示识别结果中每一行的置信度值，包含： - **average**： 行置信度平均值 - **variance**：行置信度方差 - **min**：行置信度最小值
direction	否	int32	图像方向，当detect_direction=true时存在 -1:未定义， 0:正向， 1: 逆时针90度， 2:逆时针180度， 3:逆时针270度

返回示例


```
{
  "log_id": 620759800,
  "words_result": [
    {
        "location": {
          "left": 56,
          "top": 0,
          "width": 21,
          "height": 210
        },
        "words": "3"
    }
  ],
  "words_result_num": 1
}
```

办公文档识别

可对办公类文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景。

```
 Json::Value result;;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 办公文档识别
result = client.docanalysisoffice(image,aip::null);
// 如果有可选参数
std::map<std::string, std::string> options;
// 带参数调用办公文档识别
result = client.docanalysisoffice(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少64px，最长边最大4096px。**注意：图片的base64编码是不包含图片头的，如（）**
language_type	false	string	CHN_ENG/ENG	识别语言类型，默认为CHN_ENG 可选值包括： =CHN_ENG：中英文 =ENG：英文
result_type	false	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
words_type	false	string	handwring_only/handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	false	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出
erase_seal	false	string	true/false	是否先擦除水印、印章后再识别文档

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
img_direction	false	int32	detect_direction=true时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	true	uint32	识别结果数，表示results的元素个数
results	true	array[]	识别结果数组
+words_type	true	string	文字属性（手写、印刷）,handwriting 手写，print 印刷
+words	true	array[]	整行的识别结果数组。
++line_probability	false	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++average	false	float	行置信度
+++min	false	float	整行中单字的最低置信度
++word	true	float	整行的识别结果
++words_location	true	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	true	uint32	表示定位定位位置的长方形的宽度
+++height	true	uint32	表示位置的长方形的高度
+chars	false	array[]	result_type=small时返回。单字符结果数组。
++char	false	string	result_type=small时返回。每个单字的内容。
++chars_location	false	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示位置的长方形的高度
layouts_num	false	uint32	版面分析结果数，表示layout的元素个数
layouts	false	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+layout	false	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title
+layout_location	false	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++x	false	uint32	水平坐标（坐标0点为左上角）
++y	false	uint32	水平坐标（坐标0点为左上角）
+layout_idx	false	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）

返回示例

```
{
"results_num": 5,
"log_id": "1410491260247950412",
"results": [
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 88,
    "left": 442,
    "width": 142,
    "height": 49
   },
   "word": "行程单"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 241,
    "left": 439,
    "width": 393,
    "height": 37
   },
   "word": "美国东海岸名校8天7晚"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 318,
    "left": 436,
    "width": 774,
    "height": 31
   },
   "word": "国会大厦位于华盛顿25米高的国会山上,是美国的心脏建筑。"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 374,
    "left": 434,
    "width": 805,
    "height": 31
   },
   "word": "中央顶楼上的大圆顶上立有一尊6米高的自由女神青铜雕像。"
  }
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 431,
    "left": 436,
    "width": 556,
    "height": 31
   },
   "word": "东面的大草坪是历届总统举行就职典礼的地方。"
  }
```

```
  }
 ]
}
```

二维码识别

对图片中的二维码、条形码进行检测和识别，返回存储的文字信息

```
 Json::Value result;;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 二维码识别
result = client.qrcode(image,aip::null);
// 如果有可选参数
std::map<std::string, std::string> options;
// 带参数调用二维码识别
result = client.qrcode(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
codes_result_num	是	uint32	识别结果数，表示codes_result的元素个数
codes_result	是	array[]	定位和识别结果数组
-type	是	string	识别码类型条码类型包括：9种条形码（UPC_A、UPC_E、EAN_13、EAN_8、CODE_39、CODE_93、CODE_128、ITF、CODABAR），4种二维码（QR_CODE、DATA_MATRIX、AZTEC、PDF_417）
-text	是	string	条形码识别内容,暂时只限于识别中英文结果

返回示例

```
{
  "log_id": 863402790,
  "codes_result": [
     {
        "type": "QR_CODE",
        "text": [
           "中国",
           "北京"
        ]
     }
  ],
  "codes_result_num": 1
}
示例2（多个图的情况）：
{
```

```
    "log_id": 1508509437,
    "codes_result": [
      {
        "type": "QR_CODE",
        "text": [
          "HTTP://Q8R.HK/YELZ0"
        ]
      },
      {
        "type": "PDF_417",
        "text": [
          "PDF417倢丄TL-30偱撉傒庢傝壜擻偱偡丅"
        ]
      },
      {
        "type": "CODABAR",
        "text": [
          "000800"
        ]
      },
      {
        "type": "CODE_39",
        "text": [
          "1234567890"
        ]
      },
      {
        "type": "AZTEC",
        "text": [
          "www.tec-it.com"
        ]
      },
      {
        "type": "DATA_MATRIX",
        "text": [
          "Wikipedia, the free encyclopedia"
        ]
      },
      {
        "type": "CODE_93",
        "text": [
          "123456789"
        ]
      },
      {
        "type": "CODE_128",
        "text": [
          "50090500019191"
        ]
      },
      {
        "type": "EAN_8",
        "text": [
          "12345670"
        ]
      },
      {
        "type": "EAN_13",
        "text": [
          "6901234567892"
        ]
      },
```

```
        {
            "type": "UPC_E",
            "text": [
                "01234565"
            ]
        }
    ],
    "codes_result_num": 11
}
```

试卷分析与识别

可对文档版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果，支持中、英两种语言，手写、印刷体混排多种场景

```
 Json::Value result;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 调用试卷分析与识别
result = client.docAnalysis(image, aip::null);
std::string url =
result = client.docAnalysisUrl(url, aip::null);
//  如果有可选参数
std::map<std::string, std::string> options;
options["multi_detect"] = ""
result = client.docAnalysis(image, options);
result = client.docAnalysisUrl(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少64px，最长边最大4096px。**注意：图片的base64编码是不包含图片头的，如（）**
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
language_type	false	string	CHN_ENG/ENG	识别语言类型，默认为CHN_ENG 可选值包括： =CHN_ENG：中英文 =ENG：英文
result_type	false	string	big/small	返回识别结果是按单行结果返回，还是按单字结果返回，默认为big。 =big：返回行识别结果 =small：返回行识别结果之上还会返回单字结果
detect_direction	false	string	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。其中， 0 ：正向 1：逆时针旋转90度 2：逆时针旋转180度 3：逆时针旋转270度
line_probability	false	string	true/false	是否返回每行识别结果的置信度。默认为false
words_type	false	string	handwring_only/handprint_mix	文字类型。 默认：印刷文字识别 = handwring_only：手写文字识别 = handprint_mix： 手写印刷混排识别
layout_analysis	false	string	true/false	是否分析文档版面：包括图、表、标题、段落的分析输出

返回参数详情

字段	是否必选	类型	说明
log_id	true	uint64	唯一的log id，用于问题定位
img_direction	false	int32	detect_direction=true时返回。检测到的图像朝向，0 ：正向； 1：逆时针旋转90度；2：逆时针旋转180度；3：逆时针旋转270度
results_num	true	uint32	识别结果数，表示results的元素个数
results	true	array[]	识别结果数组
+words_type	true	string	文字属性（手写、印刷）,handwriting 手写，print 印刷
+words	true	array[]	整行的识别结果数组。
++line_probability	false	array[]	line_probability=true时返回。识别结果中每一行的置信度值，包含average：行置信度平均值，min：行置信度最小值
+++average	false	float	行置信度
+++min	false	float	整行中单字的最低置信度
++word	true	float	整行的识别结果
++words_location	true	array[]	整行的矩形框坐标。位置数组（坐标0点为左上角）
+++left	true	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	true	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	true	uint32	表示定位定位位置的长方形的宽度
+++height	true	uint32	表示位置的长方形的高度
+chars	false	array[]	result_type=small时返回。单字符结果数组。
++char	false	string	result_type=small时返回。每个单字的内容。
++chars_location	false	array[]	每个单字的矩形框坐标。位置数组（坐标0点为左上角）
+++left	false	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	false	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	false	uint32	表示定位定位位置的长方形的宽度
+++height	false	uint32	表示位置的长方形的高度
layouts_num	false	uint32	版面分析结果数，表示layout的元素个数
layouts	false	array[]	文档版面信息数组，包含表格、图、段落文本、标题等标签；标签的坐标位置；段落文本和表格内文本内容对应的行序号ID
+layout	false	string	版面分析的标签结果。表格:table, 图:figure, 文本:text, 标题:title
+layout_location	false	array[]	文档版面信息标签的位置，四个顶点: 左上，右上，右下，左下
++x	false	uint32	水平坐标（坐标0点为左上角）
++y	false	uint32	水平坐标（坐标0点为左上角）
+layout_idx	false	array[]	文档版面信息中的文本在results结果中的位置：版面文本标签对应的行序号ID为n，则此标签中的文本在results结果中第n+1条展示）

返回示例

```
{
"results_num": 6,
"log_id": "4488766695474114139",
"img_direction": 0,
"layouts_num": 0,
"results": [
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 124,
    "left": 136,
    "width": 418,
    "height": 65
   },
   "word": "五默写(4分)"
  },
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 246,
    "left": 136,
    "width": 37,
    "height": 45
   },
   "word": "1"
  },
 },
 {
  "words_type": "handwriting",
  "words": {
   "words_location": {
    "top": 195,
    "left": 237,
    "width": 469,
    "height": 104
   },
   "word": "采菊东篱下"
  },
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 241,
    "left": 889,
    "width": 287,
    "height": 52
   },
   "word": "悠然见南山?"
  },
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 415,
    "left": 134,
    "width": 472,
    "height": 52
   },
```

```
    },
    "word": "2.商女不知亡国恨"
   },
  },
  {
   "words_type": "handwriting",
   "words": {
    "words_location": {
     "top": 377,
     "left": 607,
     "width": 556,
     "height": 93
    },
    "word": "隔江犹唱后庭花。"
   },
  },
 ]
}
```

机动车销售发票

支持对机动车销售发票的26个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、机器编号、购买方名称、购买方身份证号码/组织机构代码、车辆类型、厂牌型号、产地、合格证号、发动机号码、车架号码、价税合计、价税合计小写、销货单位名称、电话、纳税人识别号、账号、地址、开户银行、税率、税额、主管税务机关及代码、不含税价格、限乘人数

```
 Json::Value result;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 调用机动车销售发票
result = client.vehicleInvoice(image, aip::null);
std::string url =
result = client.vehicleInvoiceUrl(url, aip::null);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
InvoiceCode	是	string	发票代码/机打代码
InvoiceNum	是	string	发票号码/机打号码
InvoiceDate	是	string	开票日期
MachineCode	是	string	机器编号
Purchaser	是	string	购买方名称
PurchaserCode	是	string	购买方身份证号码/组织机构代码
VehicleType	是	string	车辆类型
ManuModel	是	string	厂牌型号
Origin	是	string	产地
CertificateNum	是	string	合格证号
EngineNum	是	string	发动机号码
VinNum	是	string	车架号码
PriceTax	是	string	价税合计
PriceTaxLow	是	string	价税合计小写
Saler	是	string	销货单位名称
SalerPhone	是	string	销货单位电话
SalerCode	是	string	销货单位纳税人识别号
SalerAccountNum	是	string	销货单位账号
SalerAddress	是	string	销货单位地址
SalerBank	是	string	销货单位开户银行
TaxRate	是	string	税率
Tax	是	string	税额
TaxAuthor	是	string	主管税务机关
TaxAuthorCode	是	string	主管税务机关代码
Price	是	string	不含税价格
LimitPassenger ｜ 是	string	限乘人数	

返回示例

```
{
  "log_id": 283449393728149457,
  "words_result_num": 26,
  "words_result": {
     "InvoiceNum": "00875336",
     "Saler": "深圳市新能源汽车销售有限公司",
     "LimitPassenger": "5",
     "MachineCode": "669745967911",
     "VinNum": "LJLGTCRP1J4007581",
     "TaxRate": "16%",
     "PriceTaxLow": "106100.00",
     "InvoiceDate": "2018-11-29",
     "Price": "￥91465.52",
     "SalerBank": "中国工商银行股份有限公司深圳岭园支行",
     "TaxAuthor": "国家锐务总局深圳市龙岗区税务局第五税务所",
     "ManuModel": "江淮牌HFC7007EYBD6",
     "CertificateNum": "WCH0794J0976801",
     "Purchaser": "苏子潇",
     "VehicleType": "纯电动轿车",
     "InvoiceCode": "14975047560",
     "PriceTax": "壹拾万陆仟壹佰圆整",
     "SalerPhone": "0755-83489306",
     "SalerAddress": "深圳市龙岗区龙岗街道百世国际汽车城",
     "Origin": "安徽省合肥市",
     "EngineNum": "18958407",
     "Tax": "14634.48",
     "PurchaserCode": "5135934475603742222",
     "TaxAuthorCode": "14037589413",
     "SalerAccountNum": "中国工商银行股份有限公司深圳岭园支行",
     "SalerCode": "91449283464582922 78H"
  }
}
```


车辆合格证

支持对车辆合格证的23个关键字段进行结构化识别，包括合格证编号、发证日期、车辆制造企业名、车辆品牌、车辆名称、车辆型号、车架号、车身颜色、发动机型号、发动机号、燃料种类、排量、功率、排放标准、轮胎数、轴距、轴数、转向形式、总质量、整备质量、驾驶室准乘人数、最高设计车速、车辆制造日期

```
 Json::Value result;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 调用车辆合格证
result = client.vehicleCertificate(image, aip::null);
std::string url =
result = client.vehicleCertificateUrl(url, aip::null);
//  如果有可选参数
std::map<std::string, std::string> options;
options["language_type"] = ""
options["result_type"] = ""
options["detect_direction"] = ""
options["line_probability"] = ""
options["words_type"] = ""
result = client.vehicleCertificate(image, options);
result = client.vehicleCertificateUrl(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	识别结果数组
CertificationNo	是	string	合格证编号
CertificateDate	是	string	发证日期
Manufacturer	是	string	车辆制造企业名
CarBrand	是	string	车辆品牌
CarName	是	string	车辆名称
CarModel	是	string	车辆型号
VinNo	是	string	车架号
CarColor	是	string	车身颜色
EngineType	是	string	发动机型号
EngineNo	是	string	发动机号
FuelType	是	string	燃料种类
Displacement	是	string	排量
Power	是	string	功率
EmissionStandard	是	string	排放标准
TyreNum	是	string	轮胎数
Wheelbase	是	string	轴距
AxleNum	是	string	轴数
SteeringType	是	string	转向形式
TotalWeight	是	string	总质量
SaddleMass	是	string	整备质量
LimitPassenger	是	string	驾驶室准乘人数
SpeedLimit	是	string	最高设计车速
ManufactureDate	是	string	车辆制造日期

返回示例

```
{
  "log_id": 14814098736243057,
  "words_result_num": 23,
  "words_result": {
     "ManufactureDate": "2016年10月13日",
     "CarColor": "红",
     "LimitPassenger": "2",
     "EngineType": "WP12.460E50",
     "TotalWeight": "25000",
     "Power": "338",
     "CertificationNo": "WEK29JX98645437",
     "FuelType": "汽油",
     "Manufacturer": "陕西汽车集团有限责任公司",
     "SteeringType": "方向盘",
     "Wheelbase": "3175+1350",
     "SpeedLimit": "105",
     "EngineNo": "1418K129178",
     "SaddleMass": "8600",
     "AxleNum": "3",
     "CarModel": "SX4250MC4",
     "VinNo": "LZGJHYD83JX197344",
     "CarBrand": "陕汽牌",
     "EmissionStandard": "GB17691-2005国V,GB3847-2005",
     "Displacement": "11596",
     "CertificateDate": "2018年11月28日",
     "CarName": "牵引汽车",
     "TyreNum": "10"
  }
}
```


户口本识别

支持对户口本内常住人口登记卡的全部 22 个字段进行结构化识别，包括户号、姓名、与户主关系、性别、出生地、民族、出生日期、身份证号、本市县其他住址、曾用名、籍贯、宗教信仰、身高、血型、文化程度、婚姻状况、兵役状况、服务处所、职业、何时由何地迁往本市、何时由何地迁往本址、登记日期

```
 Json::Value result;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 调用户口本识别
result = client.householdRegister(image, aip::null);
std::string url =
result = client.householdRegisterUrl(url, aip::null);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和image二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	int	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
Name	是	object{}	姓名
+ words	是	string	所属字段的具体内容，以下各字段均含有此元素
Relationship	是	object{}	户主或与户主关系
Sex	是	object{}	性别
BirthAddress	是	object{}	出生地
Nation	是	object{}	民族
Birthday	是	object{}	生日
CardNo	是	object{}	身份证号
HouseholdNum	是	object{}	户号
FormerName	是	object{}	曾用名
Hometown	是	object{}	籍贯
OtherAddress	是	object{}	本市（县）其他住址
Belief	是	object{}	宗教信仰
Height	是	object{}	身高
BloodType	是	object{}	血型
Education	是	object{}	文化程度
MaritalStatus	是	object{}	婚姻状况
VeteranStatus	是	object{}	兵役状况
WorkAddress	是	object{}	服务处所
Career	是	object{}	职业
WWToCity	是	object{}	何时由何地迁来本市(县)
WWHere	是	object{}	何时由何地迁往本址
Date	是	object{}	登记日期

返回示例

```
{
  "log_id": 1301870459,
  "words_result": {
      "BirthAddress": {
          "words": "河南洛阳市郊区"
      },
      "Birthday": {
          "words": "2016-07-28"
      },
      "CardNo": {
          "words": "410311201607282825"
      },
      "Name": {
          "words": "孙翌晨"
      },
      "Nation": {
          "words": "汉族"
      },
      "Relationship": {
          "words": "户主"
      },
      "Sex": {
          "words": "男"
      },
  },
  "words_result_num": 7
}
```


飞机行程单识别

支持对飞机行程单的24个字段进行结构化识别，包括电子客票号、印刷序号、姓名、始发站、目的站、航班号、日期、时间、票价、身份证号、承运人、民航发展基金、保险费、燃油附加费、其他税费、合计金额、填开日期、订票渠道、客票级别、座位等级、销售单位号、签注、免费行李、验证码。 同时，支持单张行程单上的多航班信息识别。

```
 Json::Value result;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 调用飞机行程单识别
result = client.airTicket(image, aip::null);
std::string url =
result = client.airTicketUrl(url, aip::null);
//  如果有可选参数
std::map<std::string, std::string> options;
options["multi_detect"] = ""
result = client.airTicket(image, options);
result = client.airTicketUrl(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
multi_detect	否	bool	true/false	控制是否开启多航班信息识别功能，**默认值：false** - **true：开启多航班信息识别功能**，开启后返回结果中对应字段格式将改为数组类型 - **false：** 不开启，仅识别单一航班信息

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
name	是	string	姓名
starting_station	是	string	始发站
destination_station	是	string	目的站
flight	是	string	航班号
date	是	string	日期
ticket_number	是	string	电子客票号码
fare	是	string	票价
dev_fund	是	string	民航发展基金/基建费
fuel_surcharge	是	string	燃油附加费
other_tax	是	string	其他税费
ticket_rates	是	string	合计金额
issued_date	是	string	填开日期
id_num	是	string	身份证号
carrier	是	string	承运人
time	是	string	时间
issued_by	是	string	订票渠道
serial_number	是	string	印刷序号
insurance	是	string	保险费
fare_basis	是	string	客票级别
class	是	string	座位等级
agent_code	是	string	销售单位号
endorsement	是	string	签注
allow	是	string	免费行李
ck	是	string	验证码

返回示例

```
 // 识别单航班信息（multi_detect=false，或参数缺省）
{
    "log_id": 7306800033425229106,
    "direction": 0,
    "words_result_num": 18,
    "words_result": {
        "insurance": "20.00",
        "date": "2019-10-22",
        "allow": "20K",
        "flight": "CA6589",
        "issued_by": "中国国际航空服务有限公司",
        "starting_station": "武汉",
        "fare": "260.00",
        "endorsement": "不得签转改期退转",
        "ticket_rates": "350.00",
        "ck": "5866",
        "serial_number": "51523588676",
        "ticket_number": "7843708871196",
        "fuel_surcharge": "EXEMPT",
        "carrier": "南航",
        "issued_date": "2019-10-30",
        "other_tax": "",
        "fare_basis": "NREOW",
        "id_num": "411201123909020877",
        "destination_station": "合肥",
        "name": "郭达",
        "agent_code": "BJS19197300025",
        "time": "21:25",
        "class": "N",
        "dev_fund": "50.00"
    }
}

// 识别多航班信息（multi_detect=true)
{
    "words_result": {
        "insurance": [
            {
                "word": "XXX"
            }
        ],
        "date": [
            {
                "word": "2019-10-18"
            },
            {
                "word": "2019-10-21"
            }
        ],
        "flight": [
            {
                "word": "CZ3565"
            },
            {
                "word": "CZ3566"
            }
        ],
        "issued_by": [
            {
                "word": "上海携程旅行社有限公司"
            }
        ],
```

```
    ],
    "starting_station": [
      {
        "word": "北京"
      }
    ],
    "fare": [
      {
        "word": "1080.00"
      }
    ],
    "ticket_rates": [
      {
        "word": "1420.00"
      }
    ],
    "serial_number": [
      {
        "word": "45956029770"
      }
    ],
    "ticket_number": [
      {
        "word": "7849648364314"
      }
    ],
    "fuel_surcharge": [
      {
        "word": "240.00"
      }
    ],
    "carrier": [
      {
        "word": "南航"
      },
      {
        "word": "南航"
      }
    ],
    "issued_date": [
      {
        "word": "2019-09-18"
      }
    ],
    "other_tax": [],
    "id_num": [
      {
        "word": "0789654700"
      }
    ],
    "destination_station": [
      {
        "word": "深圳"
      },
      {
        "word": "北京"
      }
    ],
    "name": [
      {
        "word": "姚佳"
      }
    ],
```

```
        "time": [
          {
            "word": "13:55"
          },
          {
            "word": "16:30"
          }
        ],
        "dev_fund": [
          {
            "word": "100.00"
          }
        ]
    },
    "log_id": "1280814270572920832",
    "words_result_num": 18
}
```

通用机打发票

支持对国家/地方税务局发行的横/竖版通用机打发票的23个关键字段进行结构化识别，包括发票类型、发票号码、发票代码、开票日期、合计金额大写、合计金额小写、商品名称、商品单位、商品单价、商品数量、商品金额、机打代码、机打号码、校验码、销售方名称、销售方纳税人识别号、购买方名称、购买方纳税人识别号、合计税额等。

```
 Json::Value result;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 调用通用机打发票
result = client.invoice(image, aip::null);
std::string url =
result = client.invoiceUrl(url, aip::null);
//  如果有可选参数
std::map<std::string, std::string> options;
options["location"] = ""
result = client.invoice(image, options);
result = client.invoiceUrl(image, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	string	true/false	是否输出位置信息，true：输出位置信息，false：不输出位置信息，默认false

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
InvoiceType	否	string	发票类型
InvoiceCode	否	string	发票代码
InvoiceNum	否	string	发票号码
InvoiceDate	否	string	开票日期
AmountInFiguers	否	string	合计金额小写
AmountInWords	否	string	合计金额大写
CommodityName	否	string	商品名称
CommodityUnit	否	string	商品单位
CommodityPrice	否	string	商品单价
CommodityNum	否	string	商品数量
CommodityAmount	否	string	商品金额
MachineCode	否	string	机打代码
MachineNum	否	string	机打号码
CheckCode	否	string	校验码
SellerName	否	string	销售方名称
SellerRegisterNum	否	string	销售方纳税人识别号
PurchaserName	否	string	购买方名称
PurchaserRegisterNum	否	string	购买方纳税人识别号
TotalTax	否	string	合计税额
Province	否	string	省
City	否	string	市
Time	否	string	时间
SheetNum	否	string	联次

返回示例

```
{
  "log_id": 4423022131715883558,
  "direction": 0,
  "words_result_num": 22,
  "words_result": {
     "City": "",
     "InvoiceNum": "01445096",
     "SellerName": "百度餐饮店",
     "IndustrSort": "生活服务",
     "Province": "广东省",
     "CommodityAmount": [
        {
           "word": "183.00",
           "row": "1"
        }
     ],
     "InvoiceDate": "2020年07月28日",
     "PurchaserName": "中信建投证券股份有限公司",
     "CommodityNum": [],
     "InvoiceCode": "144001901511",
     "CommodityUnit": [],
     "SheetNum": "",
     "PurchaserRegisterNum": "91442300845348OX9",
     "Time": "",
     "CommodityPrice": [],
     "AmountInFiguers": "183.00",
     "AmountInWords": "壹佰捌拾叁元整",
     "CheckCode": "61042119820421061301",
     "TotalTax": "183.00",
     "InvoiceType": "广东通用机打发票",
     "SellerRegisterNum": "61042119820421061301",
     "CommodityName": [
        {
           "word": "餐费",
           "row": "1"
        }
     ]
  }
}
```


护照识别

支持对中国大陆护照个人资料页所有15个字段进行结构化识别，包括国家码、护照号、姓名、姓名拼音、性别、出生地点、出生日期、签发地点（不支持境外签发地）、签发日期、有效期、签发机关、护照类型、国籍、MRZCode1、MRZCode2。

```
 Json::Value result;
std::string image;
aip::get_file_content("/assets/sample.jpg", &image);
// 调用护照识别
result = client.passport(image, aip::null);
std::string url =
result = client.passportUrl(url, aip::null);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
-location	是	uint32	水平坐标
-words	是	string	识别内容

返回示例


```
{
  "log_id": 7377468409496932872,
  "words_result_num": 14,
  "words_result": {
     "国家码": {
        "location": {
           "width": 59,
           "top": 200,
           "left": 762,
           "height": 26
        },
        "words": "CHN"
     },
     "护照签发地点": {
        "location": {
           "width": 236,
           "top": 505,
           "left": 558,
           "height": 43
        },
        "words": "山东/SHANDONG"
     },
     "MRZCode2": {
        "location": {
           "width": 1252,
           "top": 797,
           "left": 145,
           "height": 88
        },
        "words": "E898657303CHNSDM0702<<<<<<<<<<<<<<"
     },
     "有效期至": {
        "location": {
           "width": 287,
           "top": 528,
```

```
            "left": 955,
            "height": 46
        },
        "words": "20261004"
    },
    "签发机关": {
        "location": {
            "width": 271,
            "top": 583,
            "left": 552,
            "height": 42
        },
        "words": "出入境管理局"
    },
    "MRZCode1": {
        "location": {
            "width": 1201,
            "top": 781,
            "left": 162,
            "height": 45
        },
        "words": "PONSUN<<JIAJIA<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
    },
    "护照号码": {
        "location": {
            "width": 237,
            "top": 215,
            "left": 994,
            "height": 48
        },
        "words": "E89865730"
    },
    "签发日期": {
        "location": {
            "width": 279,
            "top": 453,
            "left": 955,
            "height": 46
        },
        "words": "20161005"
    },
    "出生地点": {
        "location": {
            "width": 216,
            "top": 429,
            "left": 564,
            "height": 43
        },
        "words": "山东/SHANDONG"
    },
    "姓名": {
        "location": {
            "width": 159,
            "top": 247,
            "left": 581,
            "height": 34
        },
        "words": "孙嘉佳"
    },
    "姓名拼音": {
        "location": {
            "width": 229,
            "top": 279
```

```
            "top": 279,
            "left": 578,
            "height": 41
        },
        "words": "SUN,JIAJIA"
    },
    "国籍": {
        "location": {
            "width": 209,
            "top": 366,
            "left": 695,
            "height": 42
        },
        "words": "中国/CHINESE"
    },
    "生日": {
        "location": {
            "width": 202,
            "top": 382,
            "left": 950,
            "height": 39
        },
        "words": "19950723"
    },
    "性别": {
        "location": {
            "width": 73,
            "top": 357,
            "left": 570,
            "height": 34
        },
        "words": "男/M"
    }
  }
}
```


网约车行程单识别

对各大主要服务商的网约车行程单进行结构化识别，包括滴滴打车、花小猪打车、高德地图、曹操出行、阳光出行，支持识别服务商、行程开始时间、行程结束时间、车型、总金额等16 个关键字段。

```
 Json::Value result;
std::string image;
aip::get_file_content("文件或图片路径", &image);
std::string pdf_file;
aip::get_file_content("文件或图片路径", &pdf_file);
std::string url = "https//www.x.com/sample.jpg"
// 调用网约车行程单识别
result = client.onlineTaxiItinerary(image);
result = client.onlineTaxiItineraryUrl(url);
result = client.onlineTaxiItineraryPdf(pdf_file,aip::null);
//  如果有可选参数
std::map<std::string, std::string> options;
options["pdf_file_num"] = 1
result = client.onlineTaxiItineraryPdf(pdf_file, options);
```

请求参数详情

参数	是否必选	类型	说明
image	和url二选一	string	图像数据，base64编码后进行urlencode，base64编码去除编码头（data:image/jpeg;base64），要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效，请注意关闭URL防盗链
pdf_file	和image/url 三选一	string	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页

返回参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ ServiceProvider	是	string	服务商
+ StartTime	是	string	行程开始时间
+ EndTime	是	string	行程结束时间
+ Phone	是	string	行程人手机号
+ ApplicationDate	是	string	申请日期
+ TotalFare	是	string	总金额
+ ItemNum	是	array	行程信息中包含的行程数量
+ Items	是	array	行程信息
++ ItemId	是	string	行程信息的对应序号
++ PickupTime	是	string	上车时间
++ PickupDate	是	string	上车日期
++ CarType	是	string	车型
++ Distance	是	string	里程
++ StartPlace	是	string	起点
++ DestinationPlace	是	string	终点
++ City	是	string	城市
++ Fare	是	string	金额
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "log_id": 1385196013945356288,
  "words_result_num": 7
  "words_result": {
     "TotalFare": "2316",
     "EndTime": "2020-07-30 19:00",
     "Phone": "13000000000",
     "ServiceProvider": "滴滴企业版",
     "StartTime": "2020-07-01 16:00",
     "ApplicationDate": "2017-12-08",
     "ItemId": "3"
     "items": [
        {
           "ItemId": "1",
           "StartPlace": "鱼化寨地铁-D口",
           "PickupTime": "16:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.7",
           "PickupDate": "20-07-01",
           "DestinationPlace": "创新港",
           "Fare": "20.86"
        },
        {
           "ItemId": "2",
           "StartPlace": "科学园东门",
           "PickupTime": "14:56",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "91",
           "PickupDate": "20-07-02",
           "DestinationPlace": "鱼化寨地铁站",
           "Fare": "18.58"
        },
        {
           "ItemId": "3",
           "StartPlace": "中俄丝路创新园东门",
           "PickupTime": "19:00",
           "CarType": "快车",
           "City": "西安市",
           "Distance": "9.1",
           "PickupDate": "20-07-30",
           "DestinationPlace": "新门地铁站",
           "Fare": "20.38"
        },
     ],
  },
}
```


磅单识别

结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单。

```
 Json::Value result;
std::string image;
aip::get_file_content("文件或图片路径", &image);
std::string url = "https//www.x.com/sample.jpg"
std::string pdf_file;
aip::get_file_content("文件或图片路径", &image);
// 调用磅单识别
result = client.weightNote(image,aip::null);
result = client.weightNoteUrl(url,aip::null);
result = client.weightNotePdf(pdf_file,aip::null);
//  如果有可选参数
std::map<std::string, std::string> options;
options["pdf_file_num"] = 1
options["probability"] = false
result = client.weightNote(image, options);
result = client.weightNoteUrl(url, options);
result = client.weightNotePdf(pdf_file, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image/pdf_file 三选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ PlateNum	是	object	车牌号
+ PrintTime	是	object	打印时间
+ CrossWeight	是	object	毛重
+ TareWeight	是	object	皮重
+ NetWeight	是	object	净重
+ SendingCompany	是	object	发货单位
+ ReceivingCompany	是	object	收货单位
+ DeliveryNumber	是	object	单号
++ word	是	string	字段识别结果，以上各字段均包含此参数
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
pdf_file_size	否	string	传入PDF文件的总页数，当 pdf_file 参数有效时返回该字段

返回示例

```
{
  "words_result": [
    {
      "TareWeight": [
      {
            "word": "14.20"
        }
      ],
      "CrossWeight": [
        {
          "word": "50.70"
        }
      ],
      "PlateNum": [
        {
          "word": "京A12356"
        }
      ],
      "SendingCompany": [
       {
            "word": "宣化县耿矿煤业有限公司"
        }
      ],
      "DeliveryNumber": [
       {
            "word": "278933000"
        }
      ],
      "ReceivingCompany": [
        {
          "word": "宁夏市京裕达实业公司"
        }
      ],
      "PrintTime": [
        {
          "word": "2020年1月1日"
        }
      ],
      "NetWeight": [
        {
          "word": "36.50"
        }
      ]
    }
  ],
  "words_result_num": 1,
  "log_id": 1428311410130160734
}
```


医疗费用明细识别

支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，包含项目类型、项目名称、单价、数量、规格、金额，其中北京地区识别效果最佳。

```
 Json::Value result;
std::string url = "https//www.x.com/sample.jpg"
std::string image;
aip::get_file_content("文件或图片路径", &image);
// 调用医疗费用明细识别
result = client.medicalDetail(image,aip::null);
result = client.medicalDetailUrl(url,aip::null);
//  如果有可选参数
std::map<std::string, std::string> options;
options["location"] = true
options["probability"] = false
result = client.medicalDetail(image, options);
result = client.medicalDetailUrl(url, options);
```

请求参数详情

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
location	否	true/false	-	是否返回字段的位置信息，**默认为 false，可缺省** - **false**：不返回字段位置信息 - **true**：返回字段的位置信息，包括上边距（top）、左边距（left）、宽度（width）、高度（height）
probability	否	true/false	-	是否返回字段识别结果的置信度，**默认为 false，可缺省** - **false**：不返回字段识别结果的置信度 - **true**：返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）和最小值（min）

返回参数详情

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
words_result	是	object	识别结果
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ Name	是	object	姓名
+ Date	是	object	日期
+ PatientID	是	object	病人ID
+ TotalAmount	是	object	总金额
+ word	是	string	字段识别结果，以上各字段均包含此参数
++ location	否	object	字段位置信息，当请求参数 location=true 时，以上各字段均包含此参数
+++ top	否	uint32	字段的上边距
+++ left	否	uint32	字段的左边距
+++ height	否	uint32	字段的高度
+++ width	否	uint32	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+ CostDetail	是	array[]	项目明细

CostDetail字段包含多个Array，每个数组包含多个object，见以下参数

字段	说明
++ word_name	字段名，包括：项目类型、项目名称、单价、数量、规格、金额
++ word	word_name字段对应的识别结果

返回示例


```
{
  "log_id": 1397090241579319296,
  "words_result_num": 5,
  "words_result": {
    "PatientID": {
        "word": "23683829"
    },
    "TotalAmount": {
        "word": "600.00"
    },
    "Date": {
        "word": "2020年11月04日"
    },
    "Name": {
        "word": "范浩"
    },
    "CostDetail": [
        [
            {
                "word_name": "清单项目名称",
                "word": "普通过敏原(新)筛查"
            },
            {
                "word_name": "单价",
                "word": "580.00"
```

```
            },
            {
                "word_name": "数量",
                "word": "1.00"
            },
            {
                "word_name": "金额",
                "word": "580.00"
            },
            {
                "word_name": "规格",
                "word": "次"
            },
            {
                "word_name": "项目类型",
                "word": "化验费"
            }
        ],
        [
            {
                "word_name": "清单项目名称",
                "word": "总IgE测定"
            },
            {
                "word_name": "单价",
                "word": "20.00"
            },
            {
                "word_name": "数量",
                "word": "1.00"
            },
            {
                "word_name": "金额",
                "word": "20.00"
            },
            {
                "word_name": "规格",
                "word": "次"
            },
            {
                "word_name": "项目类型",
                "word": "化验费"
            }
        ]
    ],
  },
}
```


彩票识别

SDK 调用示例

```
 Json::Value result;
std::string image;
aip::get_file_content("文件或图片路径", &image);
std::string url = "https//www.x.com/sample.jpg"

// 调用彩票识别
result = client.lottery_v1(image, options);
result = client.lottery_v1_url(url, options);
```

接口详情

可参考API文档：彩票识别

iOCR通用版

SDK 调用示例

```
 Json::Value result;
std::string image;
std::string pdf_file;
aip::get_file_content("文件或图片路径", &image);
std::string url = "https//www.x.com/sample.jpg"
aip::get_file_content("文件或图片路径", &pdf_file);

 // 调用iOCR通用版
result = client.recognise_iocr_v1(image, aip::null);
result = client.recognise_iocr_v1_url(url, aip::null);
result = client.recognise_iocr_v1_pdf(pdf_file, aip::null);
```

接口详情

可参考API文档：iOCR通用版

出生证明识别

SDK 调用示例

```
 Json::Value result;
std::string image;
aip::get_file_content("文件或图片路径", &image);
std::string url = "https//www.x.com/sample.jpg"

 // 调用出生证明识别
result = client.birth_certificate_v1(image, aip::null);
result = client.birth_certificate_v1_url(url, aip::null);
```

接口详情

可参考API文档：出生证明识别

定额发票识别

SDK 调用示例

```
 Json::Value result;
std::string image;
std::string pdf_file;
aip::get_file_content("文件或图片路径", &image);
std::string url = "https//www.x.com/sample.jpg"
aip::get_file_content("文件或图片路径", &pdf_file);

 // 定额发票识别
result = client.quota_invoice_v1(image, aip::null);
result = client.quota_invoice_v1_url(url, aip::null);
result = client.quota_invoice_v1_pdf(pdf_file, aip::null);
```

接口详情

可参考API文档：定额发票识别

图文转换器

SDK 调用示例

```
    // 图片文件
std::string img_file_content;
  aip::get_file_content("图片文件路径", &img_file_content);
  // url
  std::string url = "https//www.x.com/sample.jpg"
  // pdf文件
  std::string pdf_file_content;
  aip::get_file_content("pdf文件路径", &pdf_file_content);

  Json::Value options;

  Json::Value result = ocr->doc_convert_request_v1(img_file_content, options);
  Json::Value result = ocr->doc_convert_request_v1_url(url.c_str(), options);
  Json::Value result = ocr->doc_convert_request_v1_pdf(pdf_file_content, options);

  // 获取结果
  std::string task_id = "任务ID";
  Json::Value result = ocr->doc_convert_result_v1(task_id);
```

接口详情

可参考API文档：图文转换器

表格文字识别同步接口

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别同步接口。

表格文字识别

接口已下线，请使用表格文字识别V2，历史版本可查看表格文字识别。

表格识别结果

接口已下线，请使用表格文字识别V2，历史版本可查看表格识别结果。

错误信息

错误返回格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

错误码

错误码	错误信息	描述
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天流量超限额
18	Open api qps request limit	QPS超限额

	reached	
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
282000	internal error	服务器内部错误，如果您使用的是高精度接口，报这个错误码的原因可能是您上传的图片中文字过多，识别超时导致的，建议您对图片进行切割后再识别，其他情况请再次请求， 如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下

282110	urls not exit	URL参数不存在，请核对URL后再次提交
282111	url format illegal	URL格式非法，请检查url格式是否符合相应接口的入参要求
282112	url download timeout	url下载超时，请检查url对应的图床/图片无法下载或链路状况不好，您可以重新尝试以下，如果多次尝试后仍不行，建议更换图片地址
282113	url response invalid	URL返回无效参数
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误

Android SDK

简介

本文档主要介绍OCR Android SDK的安装和使用。在使用本文档前，您需要先了解Optical Character Recognition(OCR)的基础知识，并已经开通了OCR服务。视频教程请参见 OCR 在线 Android SDK 使用教程 。

接口能力

远程API能力

SDK提供了下列百度AI开放平台RESTful接口的封装：

接口名称	接口能力简要描述
通用文字识别	识别图片中的文字信息
通用文字识别（高精度版）	更高精度地识别图片中的文字信息
通用文字识别（含位置信息版）	识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（高精度含位置版）	更高精度地识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（含生僻字版）	识别图片中的文字信息（包含对常见字和生僻字的识别）
网络图片文字识别	识别一些网络上背景复杂，特殊字体的文字
身份证识别	识别身份证正反面的文字信息，并支持端上数据加密
银行卡识别	识别银行卡的卡号并返回发卡行和卡片性质信息，并支持端上数据加密
驾驶证识别	识别机动车驾驶证所有关键字段
行驶证识别	识别机动车行驶证所有关键字段
车牌识别	对小客车的车牌进行识别
营业执照识别	对营业执照进行识别
通用票据识别	对各类票据图片（医疗票据，保险保单等）进行文字识别，并返回文字在图片中的位置信息
增值税发票识别	对增值税普票、专票、卷票、电子发票进行识别
出租车发票识别	识别全国各大城市出租车票
VIN码识别	对车辆挡风玻璃处的车架号码进行识别
火车票识别	对红、蓝火车票进行识别
数字识别	对图片中的数字进行提取和识别
二维码识别	对二维码、条形码中对应的文字内容进行识别
飞机行程单识别	对飞机行程单中的姓名、始发站、目的站、航班号、日期、票价字段进行结构化识别
机动车销售发票识别	对机动车销售发票的号码、代码、日期、价税合计等字段进行结构化识别
车辆合格证识别	对车辆合格证的编号、车架号、排放标准、发动机编号等字段进行结构化识别
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果
手写文字识别	对手写汉字或手写数字进行识别
护照识别	支持对中国大陆居民护照的资料页进行结构化识别
户口本识别	对户口本的出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码等字段进行识别别
通用机打发票识别	对国家/地方税务局发行的横/竖版通用机打发票的号码、代码、日期、合计金额、类型等字段进行结构化识别
医疗费用明细识别	支持识别全国医疗费用明细识别
网约车行程单识别	对国家/地方税务局发行的横/对各大主要服务商的网约车行程单进行结构化识别
磅单识别	结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单

本地质量控制能力

除了包含远程API调用能力外，安卓SDK中还集成了身份证识别的本地质量控制能力，提供给开发者本地检测身份证的功能。SDK可以先行在本地完成身份证的预判断，然后上传至服务端识别，以达成“自动扫描识别”的功能，使用时可实时检测

取景框中是否包含身份证，是否存在模糊、欠/过曝等情况，并提示用户矫正，提高图片采集质量，提升识别准确率。安卓SDK下载

版本更新记录

上线日期	版本号	更新内容
2022.6.9	2.0.1	修复身份证、银行卡的采集质控模块鉴权问题
2022.2.17	2.0.0	升级安全策略，更新鉴权校验规则，需绑定应用签名 MD5 生成鉴权文件进行账号鉴权。使用此版本 SDK 建议前往 文字识别控制台-应用管理 编辑添加 Android 签名 MD5 信息
2022.1.7	1.4.9	更新身份证、银行卡数据加密功能，新增磅单识别、网约车行程单、医疗费用明细识别功能
2021.8.5	1.4.7	新增二维码、飞机行程单、机动车销售发票、车辆合格证、试卷分析与识别、手写识别、护照、户口本、通用机打发票识别功能
2021.2.2	1.4.6	新增增值税发票、出租车发票、VIN码、火车票和数字识别功能
2020.12.3	1.4.5	修复安卓10环境下的闪退问题
2018.2.8	1.4.2	修复高精度通用文字识别调用api错误的问题
2018.2.1	1.4.1	优化和修复了一些引起崩溃的问题；身份证本地扫描新增一个用户手动加和释放模型的类，强烈推荐用户参照demo中手动初始化和释放模型
2017.11.23	1.4.0	新增高精度版通用文字，营业执照，通用票据接口
2017.11.2	1.3.3	修复一个本地代码内存泄露问题，优化代码结构
2017.10.17	1.3.2	修复token对象expireTime时间异常的问题
2017.9.21	1.3.1	修复了一些机型下autofocus fail的错误；添加了请求接口token过期前10秒自动获取新token的逻辑；对demo界面文案做了微调
2017.8.15	1.3.0	增加驾驶证，行驶证，车牌识别功能；修复了一个潜在内存泄露问题；身份证本地质量控制模型升级，加入完整性保证
2017.8.1	1.2.3	ui库输出格式RGB565压缩，身份证识别参数加入压缩质量，对焦实现改为间隔自动对焦，修复了一些问题
2017.7.14	1.2.2	配合添加身份证本地能力升级SDK的安全性，身份证识别支持自动质量控制扫描模式以及默认的拍照识别模式
2017.6.30	1.2.1	1.对SDK的安全性作出优化 2.对本地身份证输入校验功能进行升级，该功能暂时不可用
2017.6.20	1.2.0	ocr_ui库身份证识别升级，交互修改为基于本地模型实现实时扫描判断后自动上传识别身份证
2017.5.18	1.1.0	增加通用文字识别基础版，生僻字，网图接口的SDK接口和demo演示；移除okhttp依赖；支持x86架构CPU；略微优化了demo的交互
2017.4.13	1.0.2	修复部分用户使用ak，sk方式无法获取token的问题
2017.3.23	1.0.1	更新demo获取token失败的错误提示的交互
2017.3.16	1.0.0	在线OCR第一版！

快速入门

支持的系统和硬件版本：

- 系统：支持 Android 4.1（API Level 16）到Android 12（API Level 31）系统。需要开发者通过minSdkVersion来保证支持系统的检测。
- CPU架构：armeabi，arm64-v8a，armeabi-v7a，x86。
- 机型：手机和平板皆可。
- 硬件要求：要求设备上有相机模块。
- 网络：支持WIFI及移动网络，移动网络支持使用NET网关及WAP网关（CMWAP、CTWAP、UNIWAP、3GWAP）。

开发包说明

```
aip-ocr-android-sdk.zip             // OCR SDK包，包括文档，demo工程，SDK核心库
|- OCRDemo                        // demo示例工程
  |- libs                        // lib 库，包括aar包。
  |- ocr-ui                      // ocr UI模块
```

sdk的包含的UI部分和demo工程以Android Studio方式提供，sdk部分则可以较方便的集成到eclipse工程中。

1. 前往SDK下载页面下载Android SDK压缩包。
2. (必须)将下载包libs目录中的ocr-sdk.aar文件拷贝到工程libs目录中，并加入工程依赖。
3. (可选)如果需要使用UI模块，请在Android studio中以模块方式导入下载包中的ocr-ui文件夹。

为您自己的工程添加必要的权限

如果您在自己的工程中集成SDK，请确保已经在工程AndroidManifest.xml文件中添加如下权限：

```
 <uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

各个权限的用途说明见下表：

名称	用途
INTERNET	应用联网，发送请求数据至服务器，获得识别结果。
CAMERA	调用相机进行拍照（仅UI部分需要）
WRITE_EXTERNAL_STORAGE	图片裁剪临时存储
READ_EXTERNAL_STORAGE	图片裁剪临时存储

Proguard配置

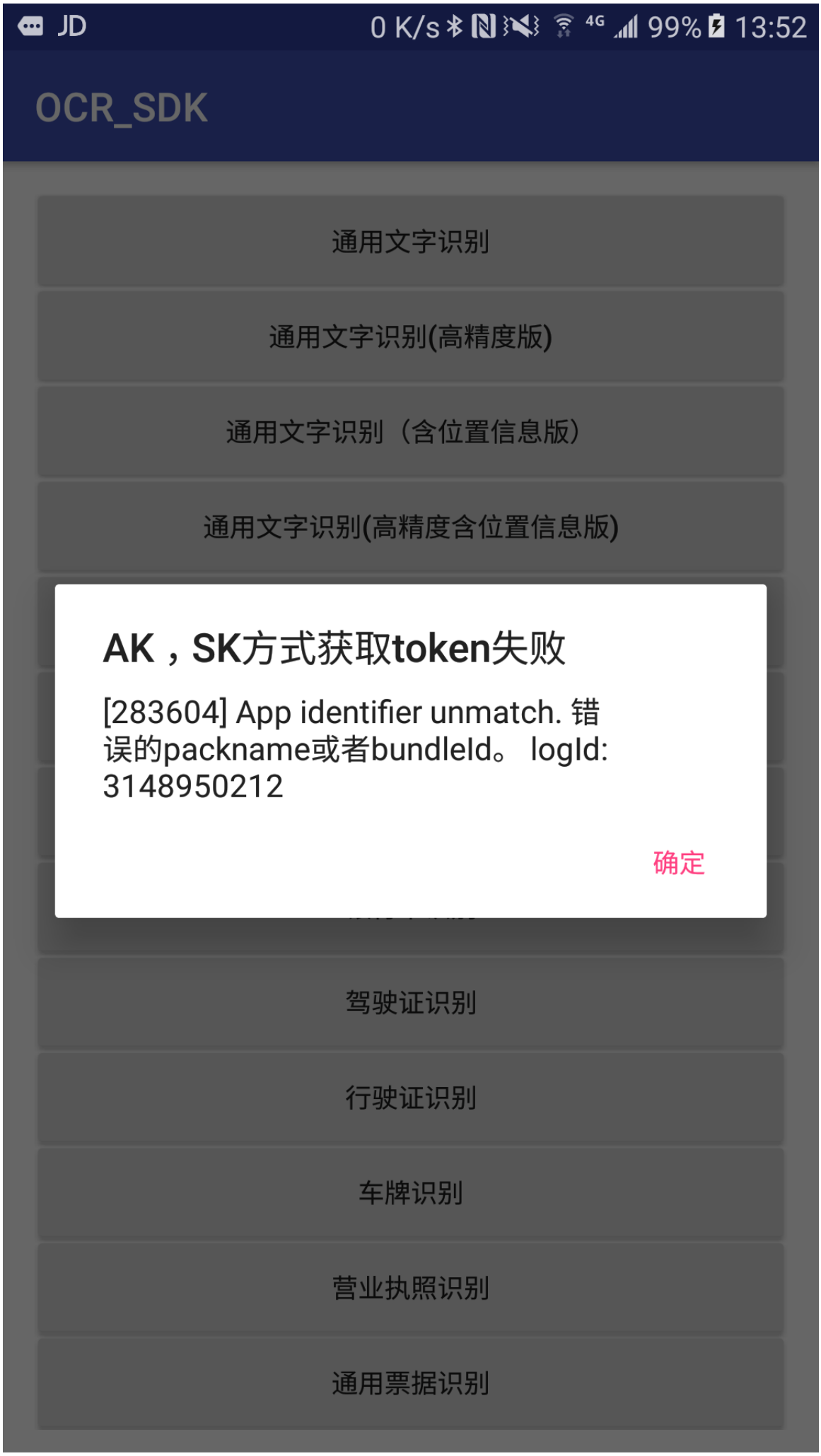
如果您在自己的工程中集成SDK，请在Proguard配置文件中增加，防止release发布时打包报错：

```
 -keep class com.baidu.ocr.sdk.**{*;}
-dontwarn com.baidu.ocr.**
```

DEMO使用说明

安卓SDK包中提供了一个可快速运行的demo工程，该工程已经集成了sdk，UI库，您只需直接在Android Studio中导入开发包OCRDemo目录即可运行。

若运行提示"身份验证错误"，可能是您还未填写正确的Api Key和Secret Key，或者是还未绑定您安卓应用的包名，如何绑定包名请参考下一章节：***身份验证与安全*** 章节


JD
0 K/s 4G 99% 13:52
OCR_SDK
通用文字识别
通用文字识别(高精度版)
通用文字识别（含位置信息版）
通用文字识别(高精度含位置信息版)
AK，SK方式获取token失败
[283604] App identifier unmatch. 错误的packname或者bundleId。 logId: 3148950212
确定
驾驶证识别
行驶证识别
车牌识别
营业执照识别
通用票据识别

百度AI开放平台使用OAuth2.0授权调用开放API，调用API时必须在URL中带上accesss_token参数。**AccessToken可用AK/SK或者授权文件的方式获得**。安卓SDK中已经为您做了封装，当初始化完毕后，所有API请求会自动带上accesss_token参数，您也可以通过initAccessTokenWithAkSk，initAccessToken这两个函数的回调中查看。

OCR Android SDK提供了以下2种AccessToken管理方法。

通过API Key / Secret Key获取AccessToken

此种身份验证方案使用AK/SK获得AccessToken。

虽然SDK对网络传输的敏感数据进行了二次加密，但由于AK/SK是明文填写在代码中，在移动设备中可能会存在AK/SK被盗取的风险。有安全考虑的开发者可使用第二种授权方案。

使用步骤：

1. 在管理控制台中新建一个OCR应用，并且请填写正确的包名和签名MD5



2. 在***应用详情***页面查看并复制应用的Api Key（简称AK） 和 Secret Key（简称SK），初始化OCR单例：

   ```
   OCR.getInstance().initAccessTokenWithAkSk(new OnResultListener() { @Override public void onResult(AccessToken result) { // 调用成功，返回AccessToken对象 String token = result.getAccessToken(); } @Override public void onError(OCRError error) { // 调用失败，返回OCRError子类SDKError对象 } }, getApplicationContext(), "您的应用AK", "您的应用SK");
   ```

由于AK/SK是明文填写在代码中，在移动设备中可能会存在AK/SK被盗取的风险。有安全考虑的开发者可使用第二种授权方案。

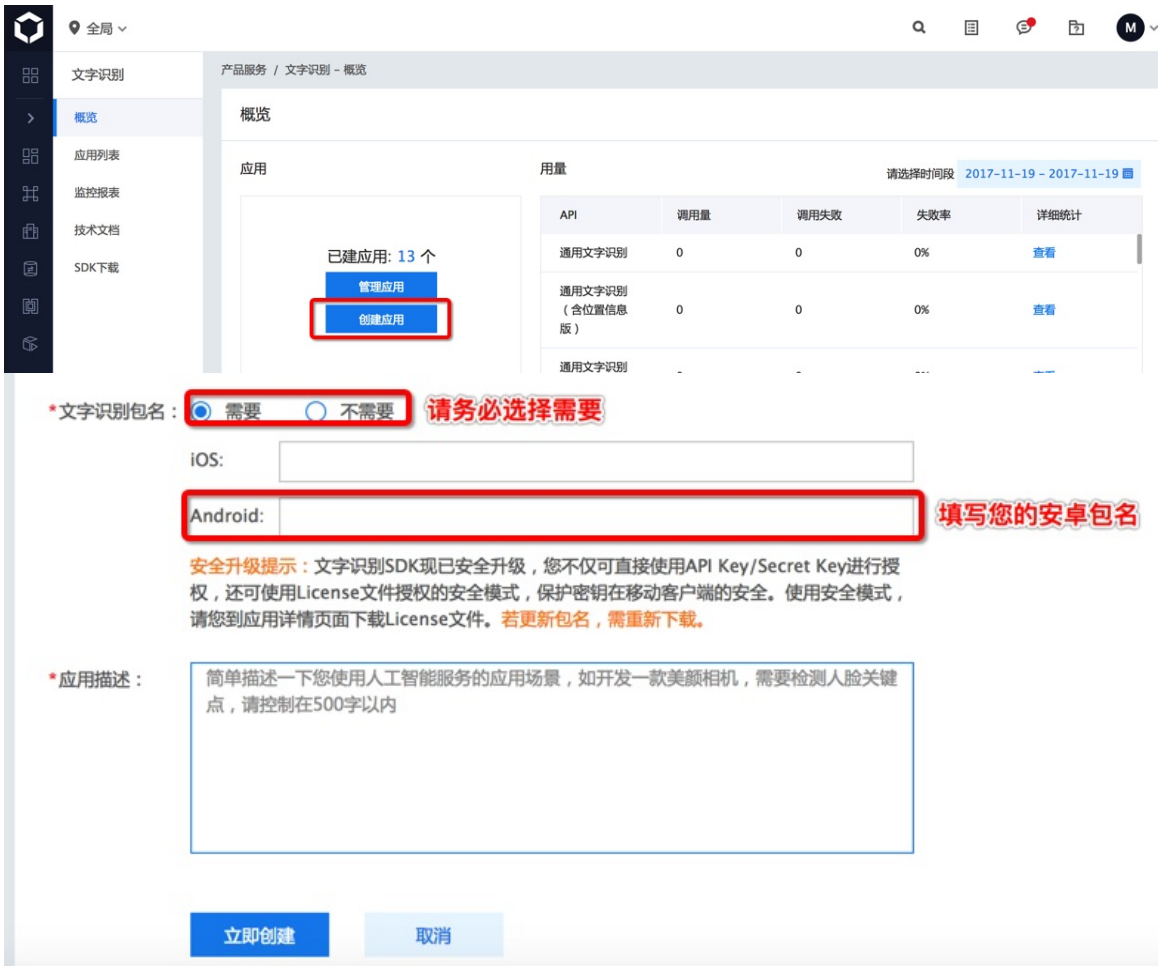
授权文件（安全模式）获取AccessToken

此种身份验证方案使用授权文件获得AccessToken，缓存在本地。***建议有安全考虑的开发者使用此种身份验证方式。***

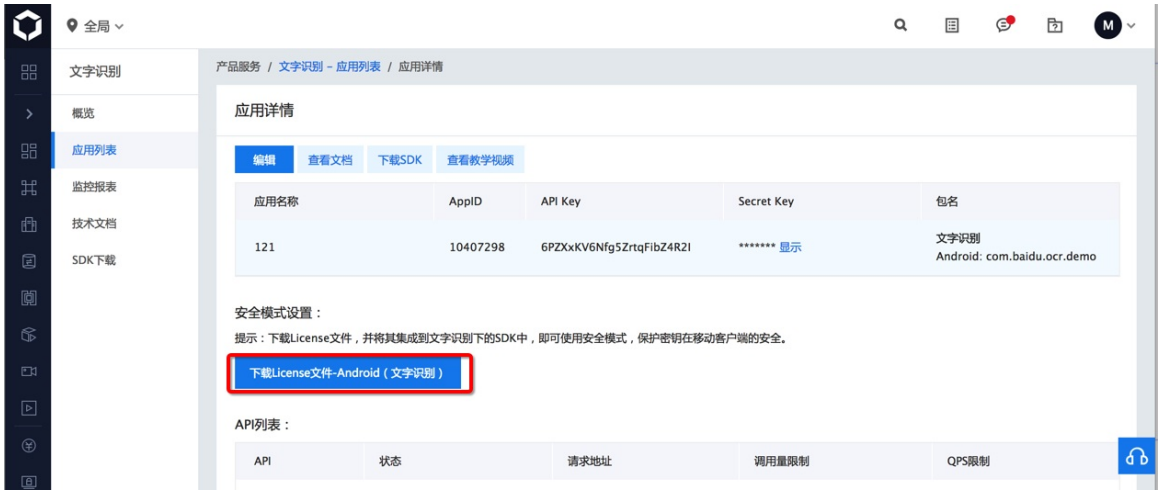
在您的移动APP分发出去之后，APP存在被反编译的可能，所以直接将AK / SK 置于APP源码之中，存在被盗取的风险。采用授权文件的身份验证方法，可有效保护AK/SK在移动设备中的安全。攻击者即使拦截了流量，盗取了授权文件，也难以盗用您的配额。

使用步骤：

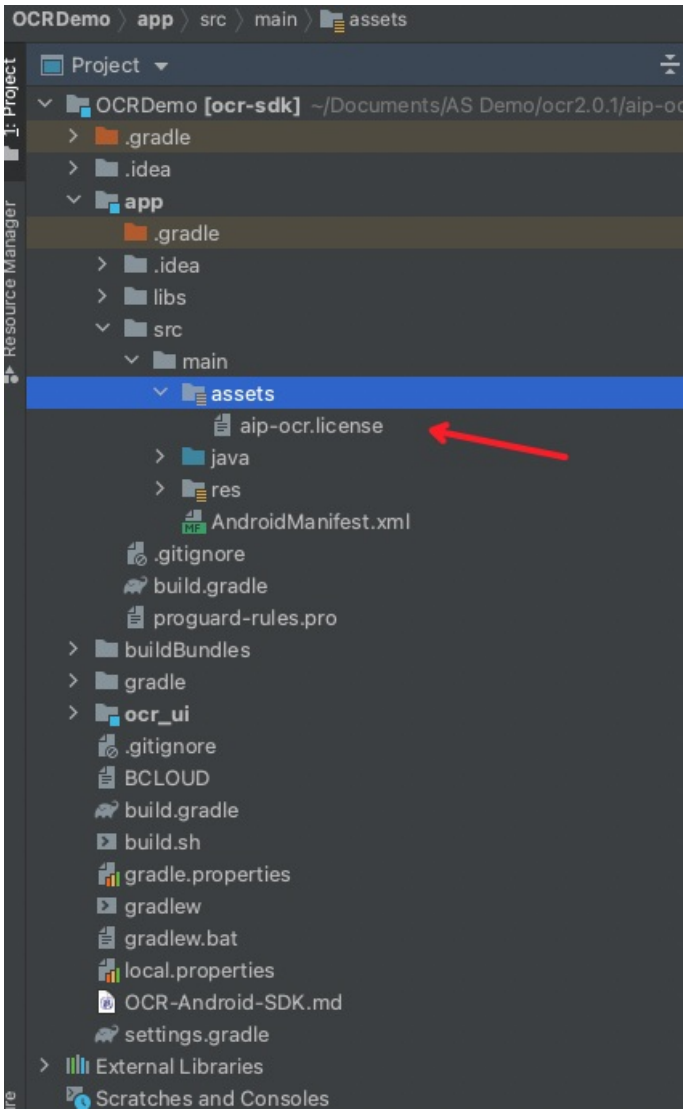
1. 在官网中配置应用



2. 在**应用详情**页面下载对应应用的授权文件



3. 将授权文件添加至工程assets文件夹，文件名必须为**aip-ocr.license**



4. 调用initAccessToken方法，初始化OCR单例：

   OCR.getInstance().initAccessToken(new OnResultListener() { @Override public void onResult(AccessToken result) { // 调用成功，返回AccessToken对象 String token = result.getAccessToken(); } @Override public void onError(OCRError error) { // 调用失败，返回OCRError子类SDKError对象 } }, getApplicationContext());

接口调用说明

OCR-UI模块

OCR-UI模块提供了一套默认的UI。如需使用，请将ocr_ui模块包含到您的工程，具体使用可参考SDK包中附带的示例工程。

OCR-UI模块调用示例

调用拍摄activity，更详细的类别请参考demo工程。

```
 // 生成intent对象
Intent intent = new Intent(IDCardActivity.this, CameraActivity.class);

// 设置临时存储
intent.putExtra(CameraActivity.KEY_OUTPUT_FILE_PATH,
FileUtil.getSaveFile(getApplication()).getAbsolutePath());

// 调用除银行卡，身份证等识别的activity
intent.putExtra(CameraActivity.KEY_CONTENT_TYPE, CameraActivity.CONTENT_TYPE_GENERAL);
startActivityForResult(intent, REQUEST_CODE_CAMERA);
// 通过参数确定接口类型
startActivityForResult(intent, REQUEST_CODE_GENERAL_BASIC);

// 调用拍摄银行卡的activity
intent.putExtra(CameraActivity.KEY_CONTENT_TYPE, CameraActivity.CONTENT_TYPE_BANK_CARD);
startActivityForResult(intent, REQUEST_CODE_CAMERA);

// 调用拍摄身份证正面（不带本地质量控制）activity
intent.putExtra(CameraActivity.KEY_CONTENT_TYPE, CameraActivity.CONTENT_TYPE_ID_CARD_FRONT);
startActivityForResult(intent, REQUEST_CODE_CAMERA);

// 调用身份证本识别（带本地质量控制）activity
 Intent intent = new Intent(IDCardActivity.this, CameraActivity.class);
// 使用本地质量控制能力需要授权，需要在OCR调用initAccessToken或者
// initAccessTokenWithAkSk成功返回后才能获取License授权本地质量控制能力
intent.putExtra(CameraActivity.KEY_NATIVE_TOKEN,
                    OCR.getInstance().getLicense());
// 使用本地质量控制能力需要设置开启
intent.putExtra(CameraActivity.KEY_NATIVE_ENABLE,
                    true);
// 开启身份证正面本地识别
intent.putExtra(CameraActivity.KEY_CONTENT_TYPE, CameraActivity.CONTENT_TYPE_ID_CARD_FRONT);
```

通过onActivityResult获取拍摄结果，更详细的类别请参考demo工程。

```
 @Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // 获取调用参数
    String contentType = data.getStringExtra(CameraActivity.KEY_CONTENT_TYPE);
    // 通过临时文件获取拍摄的图片
    String filePath = FileUtil.getSaveFile(getApplicationContext()).getAbsolutePath();
    // 判断拍摄类型（通用，身份证，银行卡等）
    if (requestCode == REQUEST_CODE_GENERAL && resultCode == Activity.RESULT_OK) {
    // 判断是否是身份证正面
        if (CameraActivity.CONTENT_TYPE_ID_CARD_FRONT.equals(contentType)){
        // 获取图片文件调用sdk数据接口，见数据接口说明
        }
    }
}
```

数据接口

通用文字识别

- 调用示例

  // 通用文字识别参数设置 GeneralBasicParams param = new GeneralBasicParams(); param.setDetectDirection(true); param.setImageFile(new File(filePath));

  // 调用通用文字识别服务 OCR.getInstance().recognizeGeneralBasic(param, new OnResultListener() { @Override public void onResult(GeneralResult result) { // 调用成功，返回GeneralResult对象 for (WordSimple wordSimple : result.getWordList()) { // wordSimple不包含位置信息 wordSimple word = wordSimple; sb.append(word.getWords());

sb.append("\n"); } // json格式返回字符串 listener.onResult(result.getJsonRes()); } @Override public void onError(OCRError error) { // 调用失败，返回OCRError对象 } });

options参数详情

参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
language_type	false	string	CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语
detect_direction	false	boolean	true、false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	FALSE	string	true、false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

- 结果返回

字段	必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
+words	否	string	识别结果字符串

```
// 示例
返回格式参考通用文字识别
```

通用文字识别（高精度版）

- 调用示例

// 通用文字识别参数设置 GeneralBasicParams param = new GeneralBasicParams(); param.setDetectDirection(true); param.setImageFile(new File(filePath));

// 调用通用文字识别服务 OCR.getInstance().recognizeAccurateBasic(param, new OnResultListener() { @Override public void onResult(GeneralResult result) { // 调用成功，返回GeneralResult对象 for (WordSimple wordSimple :

result.getWordList()) { // wordSimple不包含位置信息 wordSimple word = wordSimple; sb.append(word.getWords()); sb.append("\n"); } // json格式返回字符串 listener.onResult(result.getJsonRes()); } @Override public void onError(OCRError error) { // 调用失败，返回OCRError对象 } });

options参数详情

参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
detect_direction	false	boolean	true、false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

- 结果返回

字段	必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
+words	否	string	识别结果字符串

```
 // 示例
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
   [
      {"words": " TSINGTAO"},
      {"words": "青島啤酒"}
   ]
}
```

通用文字识别（含位置信息版）

- 调用示例

  // 通用文字识别参数设置 GeneralParams param = new GeneralParams(); param.setDetectDirection(true); param.setImageFile(new File(filePath));

  // 调用通用文字识别服务（含位置信息版） OCR.getInstance().recognizeGeneral(param, new OnResultListener() { @Override public void onResult(GeneralResult result) { StringBuilder sb = new StringBuilder(); for (WordSimple wordSimple : result.getWordList()) { // word包含位置 Word word = (Word) wordSimple; sb.append(word.getWords()); sb.append("\n"); } // 调用成功，返回GeneralResult对象，通过getJsonRes方法获取API返回字符串 listener.onResult(result.getJsonRes()); } @Override public void onError(OCRError error) { // 调用失败，返回OCRError对象 } });

options参数详情

参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据，base64编码，要求base64编码后大小不超过1M，最短边至少15px，最长边最大2048px,支持jpg/png/bmp格式
recognize_granularity	false	string	big、small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
language_type	false	string	CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语
detect_direction	false	boolean	true、false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	false	string	true、false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
vertexes_location	false	string	true、false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false

- 结果返回

字段	必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array()	定位和识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+vertexes_location	否	array()	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	uint32	水平坐标（坐标0点为左上角）
++y	是	uint32	垂直坐标（坐标0点为左上角）
+location	是	array()	位置数组（坐标0点为左上角）
++left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	是	uint32	表示定位位置的长方形的宽度
++height	是	uint32	表示定位位置的长方形的高度
+words	否	string	识别结果字符串
+chars	否	array()	单字符结果，recognize_granularity=small时存在
++location	是	array()	位置数组（坐标0点为左上角）
+++left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	uint32	表示定位定位位置的长方形的宽度
+++height	是	uint32	表示位置的长方形的高度
++char	是	string	单字符识别结果

```
 // 示例
{
    direction : 2,
    log_id : 676709620,
    words_result : [ {
            location : {
                height : 20;
                left : 86;
                top : 387;
                width : 22;
            };
            words : "N";
        },
    ],
    words_result_num : 1;
}
```

通用文字识别（高精度含位置信息版）

- 调用示例

  // 通用文字识别参数设置 GeneralParams param = new GeneralParams(); param.setDetectDirection(true);

param.setImageFile(new File(filePath));

// 调用通用文字识别服务（含位置信息版） OCR.getInstance().recognizeAccurate(param, new OnResultListener() { @Override public void onResult(GeneralResult result) { StringBuilder sb = new StringBuilder(); for (WordSimple wordSimple : result.getWordList()) { // word包含位置 Word word = (Word) wordSimple; sb.append(word.getWords()); sb.append("\n"); } // 调用成功，返回GeneralResult对象，通过getJsonRes方法获取API返回字符串 listener.onResult(result.getJsonRes()); } @Override public void onError(OCRError error) { // 调用失败，返回OCRError对象 } });

options参数详情

参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据，base64编码，要求base64编码后大小不超过1M，最短边至少15px，最长边最大2048px,支持jpg/png/bmp格式
vertexes_location	false	string	true、false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false
recognize_granularity	false	string	big、small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
detect_direction	false	boolean	true、false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。

- 结果返回

字段	必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array()	定位和识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+vertexes_location	否	array()	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	uint32	水平坐标（坐标0点为左上角）
++y	是	uint32	垂直坐标（坐标0点为左上角）
+location	是	array()	位置数组（坐标0点为左上角）
++left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	是	uint32	表示定位位置的长方形的宽度
++height	是	uint32	表示定位位置的长方形的高度
+words	否	string	识别结果字符串
+chars	否	array()	单字符结果，recognize_granularity=small时存在
++location	是	array()	位置数组（坐标0点为左上角）
+++left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	uint32	表示定位定位位置的长方形的宽度
+++height	是	uint32	表示位置的长方形的高度
++char	是	string	单字符识别结果

```
// 返回结果参考通用文字识别（含位置信息版）
```

通用文字识别(含生僻字版)

- 调用示例

  // 通用文字识别(含生僻字版)参数设置 GeneralBasicParams param = new GeneralBasicParams(); param.setDetectDirection(true); param.setImageFile(new File(filePath));

  // 调用通用文字识别服务 OCR.getInstance().recognizeGeneralEnhanced(param, new OnResultListener() { @Override public void onResult(GeneralResult result) { // 调用成功，返回GeneralResult对象 for (WordSimple wordSimple : result.getWordList()) { // wordSimple不包含位置信息 wordSimple word = wordSimple; sb.append(word.getWords()); sb.append("\n"); } } @Override public void onError(OCRError error) { // 调用失败，返回OCRError对象 } });

options参数详情

参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
language_type	false	string	CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语
detect_direction	false	boolean	true、false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	FALSE	string	true、false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

- 结果返回

字段	必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
+words	否	string	识别结果字符串

```
// 示例
```
参考通用文字识别接口

网络图片文字识别

- 调用示例

```
// 网络图片文字识别参数设置 GeneralBasicParams param = new GeneralBasicParams(); param.setDetectDirection(true); param.setImageFile(new File(filePath));

// 调用通用文字识别服务 OCR.getInstance().recognizeWebimage(param, new OnResultListener() { @Override public void onResult(GeneralResult result) { // 调用成功，返回GeneralResult对象 for (WordSimple wordSimple : result.getWordList()) { // wordSimple不包含位置信息 wordSimple word = wordSimple; sb.append(word.getWords()); sb.append("\n"); } } @Override public void onError(OCRError error) { // 调用失败，返回OCRError对象 } });
```

options参数详情

参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
detect_direction	false	boolean	true、false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	false	string	true、false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）

- 结果返回

字段	必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
+words	否	string	识别结果字符串

```
// 示例
参考通用文字识别接口
```

银行卡识别

- 调用示例

  // 银行卡识别参数设置 BankCardParams param = new BankCardParams(); param.setImageFile(new File(filePath));

  // 调用银行卡识别服务 OCR.getInstance().recognizeBankCard(param, new OnResultListener() { @Override public void onResult(BankCardResult result) { // 调用成功，返回BankCardResult对象 } @Override public void onError(OCRError error) { // 调用失败，返回OCRError对象 } });

- 结果返回

参数	类型	描述
log_id	Uint64	唯一的log id，用于问题定位
result	Object	定位和识别结果数组
+bank_card_number	String	银行卡识别结果
+bank_name	String	银行名，不能识别时为空
+bank_card_type	uint32	银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡

```
 // 示例
{
  "log_id": 3207866271;
  result: {
     "bank_card_number": "6226 2288 8888 8888",
     "bank_card_type": 1,
     "bank_name": "\U5de5\U5546\U94f6\U884c"
  };
}
```

身份证识别

- 调用示例

  // 身份证识别参数设置 IDCardParams param = new IDCardParams(); param.setImageFile(new File(filePath));

  // 调用身份证识别服务 OCR.getInstance().recognizeIDCard(param, new OnResultListener() { @Override public void onResult(IDCardResult result) { // 调用成功，返回IDCardResult对象 } @Override public void onError(OCRError error) { // 调用失败，返回OCRError对象 } });

options参数

参数	必选	范围	类型	说明
image	true		String	图像数据，支持本地图像文件路径，图像文件二进制数据，要求base64编码后大小不超过1M，最短边至少15px，最长边最大2048px,支持jpg/png/bmp格式
isFront	true	true、false	Boolean	true：身份证正面，false：身份证背面
detect_direction	false	true、false	string	是否检测图像朝向，默认不检测，即：false。可选值为：true - 检测图像朝向；false - 不检测图像朝向。朝向是指输入图像是正常方向、逆时针旋转90/180/270度
accuracy	false	auto、normal、high	string	精准度，精度越高，速度越慢。default：auto

- 结果返回

参数	类型	描述
direction	Int32	图像方向，当detect_direction=true时存在。-1:未定义，0:正向，1: 逆时针90度， 2:逆时针180度， 3:逆时针270度
log_id	Uint64	唯一的log id，用于问题定位
words_result	Array	定位和识别结果数组，数组元素的key是身份证的主体字段（正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发机关、签发日期、失效日期）。只返回识别出的字段。若身份证号码校验不通过，则不返回
words_result_num	Uint32	识别结果数，表示words_result的元素个数
+location	Array	位置数组（坐标0点为左上角）
++left	Uint32	表示定位位置的长方形左上顶点的水平坐标
++top	Uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	Uint32	表示定位位置的长方形的宽度
++height	Uint32	表示定位位置的长方形的高度
+words	String	识别结果字符串

```
 //示例
{
   "log_id": 7037721,
   "direction": 0,
   "words_result_num": 2,
   "words_result": {
      "住址": {
         "location": {
            "left": 227,
            "top": 235,
            "width": 229,
            "height": 51
         },
         "words": "湖北省天门市渔薪镇杨咀村一组2号",
      }
      ...
   }
}
```

行驶证识别

- 调用示例

  // 行驶证识别参数设置 OcrRequestParams param = new OcrRequestParams();

  // 设置image参数 param.setImageFile(new File(filePath)); // 设置其他参数 param.putParam("detect_direction", true); // 调用行驶证识别服务 OCR.getInstance().recognizeVehicleLicense(param, new OnResultListener() { @Override public void onResult(OcrResponseResult result) { // 调用成功，返回OcrResponseResult对象 listener.onResult(result.getJsonRes()); }

```
 @Override
public void onError(OCRError error) {
   // 调用失败，返回OCRError对象
}
```

```
});
```

options参数

参数	是否必选	类型	可选值范围	说明
image	是	File对象	-	图像数据, 要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
detect_direction	否	boolean	true/false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括:- true：检测朝向；- false：不检测朝向。
vehicle_license_side	否	string	front/back	- **front：默认值，**识别行驶证主页 - **back：**识别行驶证副页
unified	否	string	true/false	- **false：默认值，**不进行归一化处理 - **true：**对输出字段进行归一化处理，将新/老版行驶证的“注册登记日期/注册日期”统一为”注册日期“进行输出

- 结果返回

字段	说明	是否必选	类型
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	识别结果数组
+words	否	string	识别结果字符串

```
 //示例
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "品牌型号": {
    "words": "保时捷GT37182RUCRE"
   },
   "发证日期": {
    "words": "20160104"
   },
   "使用性质": {
    "words": "非营运"
   },
   "发动机号码": {
    "words": "20832"
   },
   "号牌号码": {
    "words": "苏A001"
   },
   "所有人": {
    "words": "圆圆"
   },
   "住址": {
    "words": "南京市江宁区弘景大道"
   },
   "注册日期": {
    "words": "20160104"
   },
   "车辆识别代号": {
    "words": "HCE58"
   },
   "车辆类型": {
    "words": "小型轿车"
   }
  }
 }
}
```

驾驶证识别

- 调用示例

  // 驾驶证识别参数设置 OcrRequestParams param = new OcrRequestParams();

  // 设置image参数 param.setImageFile(new File(filePath)); // 设置其他参数 param.putParam("detect_direction", true); // 调用驾驶证识别服务 OCR.getInstance().recognizeDrivingLicense(param, new OnResultListener() { @Override public void onResult(OcrResponseResult result) { // 调用成功，返回OcrResponseResult对象 listener.onResult(result.getJsonRes()); }

  ```
    @Override
  public void onError(OCRError error) {
      // 调用失败，返回OCRError对象
  }
  ```

  });

options参数

参数	必选	类型	可选值范围	说明
image	是	File对象	-	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
detect_direction	否	boolean	true、false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括:- true：检测朝向；- false：不检测朝向。

- 结果返回

字段	必选	类型	
log_id	是	number	唯一的log id，用于问题定位
words_result_num	是	number	识别结果数，表示words_result的元素个数
words_result	是	array	识别结果数组
+words	否	string	识别结果字符串

```
 //示例
{
 "errno": 0,
 "msg": "success",
 "data": {
  "words_result_num": 10,
  "words_result": {
   "证号": {
    "words": "3208231999053090"
   },
   "有效期限": {
    "words": "6年"
   },
   "准驾车型": {
    "words": "B2"
   },
   "有效起始日期": {
    "words": "20101125"
   },
   "住址": {
    "words": "江苏省南通市海门镇秀山新城"
   },
   "姓名": {
    "words": "小欧欧"
   },
   "国籍": {
    "words": "中国"
   },
   "出生日期": {
    "words": "19990530"
   },
   "性别": {
    "words": "男"
   },
   "初次领证日期": {
    "words": "20100125"
   }
  }
 }
}
```

车牌识别

- 调用示例

// 车牌识别参数设置 OcrRequestParams param = new OcrRequestParams();

// 设置image参数 param.setImageFile(new File(filePath));

// 调用车牌识别服务 OCR.getInstance().recognizeLicensePlate(param, new OnResultListener() { @Override public void onResult(OcrResponseResult result) { // 调用成功，返回OcrResponseResult对象 listener.onResult(result.getJsonRes()); }

```
 @Override
public void onError(OCRError error) {
    // 调用失败，返回OCRError对象
}
```

});

options参数

参数	必选	类型	可选值范围	说明
image	是	File对象	-	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

参数	是否必须	类型	说明
log_id	是	number	请求标识码，随机数，唯一
words_result	是	object	暴恐结果置信度
+color	是	string	车牌颜色，如"blue"
+number	是	string	车牌号码，示例："苏HS7766"

```
 //示例
{
 "words_result":{
    "color":"blue",
    "number":"粤FQ0000"
 },
 "log_id":2783673432
}
```

营业执照识别

- 调用示例

// 营业执照识别参数设置 OcrRequestParams param = new OcrRequestParams();

// 设置image参数 param.setImageFile(new File(filePath));

// 调用营业执照识别服务 OCR.getInstance().recognizeBusinessLicense(param, new OnResultListener() { @Override public void onResult(OcrResponseResult result) { listener.onResult(result.getJsonRes()); }

```
 @Override
public void onError(OCRError error) {
    listener.onResult(error.getMessage());
}
```

});

options参数

参数	必选	类型	可选值范围	说明
image	是	File对象	-	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

参数	是否必须	类型	说明
log_id	是	uint64	请求标识码，随机数，唯一。
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	array()	识别结果数组	
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	否	string	识别结果字符串

```
 //示例
{
    "log_id": 490058765,
    "words_result": {
        "单位名称": {
            "location": {
                "left": 500,
                "top": 479,
                "width": 618,
                "height": 54
            },
            "words": "袁氏财团有限公司"
        },
        "法人": {
            "location": {
                "left": 938,
                "top": 557,
                "width": 94,
                "height": 46
            },
            "words": "袁运筹"
        },
        "地址": {
            "location": {
                "left": 503,
                "top": 644,
                "width": 574,
                "height": 57
            },
            "words": "江苏省南京市中山东路19号"
        },
        "有效期": {
            "location": {
                "left": 779,
                "top": 1108,
                "width": 271,
                "height": 49
            },
            "words": "2015年02月12日"
```

```
        },
        "证件编号": {
            "location": {
                "left": 1219,
                "top": 357,
                "width": 466,
                "height": 39
            },
            "words": "苏餐证字(2019)第666602666661号"
        },
        "社会信用代码": {
            "location": {
                "left": 0,
                "top": 0,
                "width": 0,
                "height": 0
            },
            "words": "无"
        }
    },
    "words_result_num": 6
}
```

通用票据识别

- 调用示例

  // 通用票据识别参数设置 OcrRequestParams param = new OcrRequestParams();

  // 设置image参数 param.setImageFile(new File(filePath));

  // 设置额外参数 param.putParam("detect_direction", "true");

  // 调用通用票据识别服务 OCR.getInstance().recognizeReceipt(param, new OnResultListener() { @Override public void onResult(OcrResponseResult result) { listener.onResult(result.getJsonRes()); }

```
  @Override
 public void onError(OCRError error) {
     listener.onResult(error.getMessage());
 }
```

  });

options参数

参数	必选	类型	可选值范围	说明
image	是	File对象	-	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
recognize_granularity	false	string	big、small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	false	string	true、false	是否返回识别结果中每一行的置信度
accuracy	false	string	normal,缺省	normal 使用快速服务;缺省或其它值使用高精度服务
detect_direction	false	string	true、false	是否检测图像朝向，默认不检测，即：false。可选值包括true - 检测朝向；false - 不检测朝向。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。

- 结果返回

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array()	定位和识别结果数组
location	是	object	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位位置的长方形的宽度
height	是	uint32	表示定位位置的长方形的高度
words	是	string	识别结果字符串
chars	否	array()	单字符结果，recognize_granularity=small时存在
location	是	array()	位置数组（坐标0点为左上角）
left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
width	是	uint32	表示定位定位位置的长方形的宽度
height	是	uint32	表示位置的长方形的高度
char	是	string	单字符识别结果
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值


```
{
  "log_id": 2661573626,
  "words_result": [
    {
      "location": {
        "left": 10,
        "top": 3,
        "width": 121,
        "height": 24
      },
      "words": "姓名:小明明",
      "chars": [
        {
          "location": {
            "left": 16,
            "top": 6,
            "width": 17,
            "height": 20
          },
          "char": "姓"
        },
        {
          "location": {
            "left": 35,
            "top": 6,
            "width": 17,
            "height": 20
          },
          "char": "名"
```

```
            },
            {
                "location": {
                    "left": 55,
                    "top": 6,
                    "width": 11,
                    "height": 20
                },
                "char": ":"
            },
            {
                "location": {
                    "left": 68,
                    "top": 6,
                    "width": 17,
                    "height": 20
                },
                "char": "小"
            },
            {
                "location": {
                    "left": 87,
                    "top": 6,
                    "width": 17,
                    "height": 20
                },
                "char": "明"
            },
            {
                "location": {
                    "left": 107,
                    "top": 6,
                    "width": 17,
                    "height": 20
                },
                "char": "明"
            }
        ]
    }
  ],
  "words_result_num": 2
}
```


增值税发票识别

- 调用示例

```
 // 增值税发票识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
OCR.getInstance(ctx).recognizeVatInvoice(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
参数
Image 是 File对象 图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
type	否	string	**可选值：**normal/roll 进行识别的增值税发票类型，默认为 normal，可缺省 - **normal：**可识别增值税普票、专票、电子发票 - **roll：**可识别增值税卷票
seal_tag	否	string	**可选值：**true/false 是否开启印章判断功能，并返回印章内容的识别结果 - **true：**开启 - **false：**不开启

- 结果返回

```
{
 "words_result": {
  "AmountInWords": "",
  "InvoiceNumConfirm": "16486500",
  "CommodityPrice": [],
  "NoteDrawer": "",
  "SellerAddress": "无北司列1",
  "CommodityNum": [],
  "SellerRegisterNum": "",
  "MachineCode": "",
  "Remarks": "",
  "SellerBank": "",
  "CommodityTaxRate": [],
  "TotalTax": "",
  "InvoiceCodeConfirm": "3200092140",
  "CheckCode": "",
  "InvoiceCode": "3200092140",
  "InvoiceDate": "",
  "PurchaserRegisterNum": "",
  "InvoiceTypeOrg": "江苏增值税专用发票",
  "Password": "北兴57号:仟5其江市无无3福南林东道街路12338411889949",
  "Agent": "否",
  "AmountInFiguers": "",
  "PurchaserBank": "",
  "Checker": "",
  "City": "",
  "TotalAmount": "",
  "CommodityAmount": [],
  "PurchaserName": "北温雄件科技电路规高有限公司",
  "CommodityType": [],
  "Province": "",
  "InvoiceType": "专用发票",
  "SheetNum": "",
  "PurchaserAddress": "",
  "CommodityTax": [],
  "CommodityUnit": [],
  "Payee": "",
  "CommodityName": [],
  "SellerName": "",
  "InvoiceNum": "16486500"
 },
 "log_id": 1356821167419162624,
 "words_result_num": 38
}
```


出租车票

```
 // 出租车票识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用出租车发票识别服务
OCR.getInstance(ctx).recognizeTaxireceipt(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
    "words_result": {
     "Time": "",
     "FuelOilSurcharge": "0.00",
     "City": "",
     "Date": "",
     "Province": "",
     "Fare": "",
     "CallServiceSurcharge": "0.00",
     "TotalFare": "",
     "TaxiNum": "",
     "PricePerkm": "",
     "InvoiceCode": "",
     "Distance": "",
     "Location": "",
     "InvoiceNum": ""
   },
   "log_id": 1356822535122976768,
   "words_result_num": 14
 }
```

VIN码

- 调用示例

```
 // VIN码识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用VIN码识别服务
OCR.getInstance(ctx).recognizeVincode(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
 {
  "words_result": [],
  "log_id": 1356823413598978048,
  "words_result_num": 0
}
```

火车票

- 调用示例

```
 // 火车票识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用火车票识别服务
OCR.getInstance(ctx).recognizeTrainticket(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
 {
  "words_result": {
    "date": "",
    "starting_station": "",
    "ticket_num": "",
    "train_num": "",
    "ticket_rates": "",
    "serial_number": "",
    "seat_category": "",
    "id_num": "",
    "name": "",
    "destination_station": "",
    "time": "",
    "sales_station": "",
    "seat_num": ""
  },
  "log_id": 1356824154413727744,
  "words_result_num": 13,
  "direction": 3
}
```

数字识别

- 调用示例

```
 // 数字识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用数字识别服务
OCR.getInstance(ctx).recognizeNumbers(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
 {
  "log_id": 7693179126190465891,
  "words_result_num": 2,
  "words_result": [
    {
      "location": {
        "width": 122,
        "top": 17,
        "left": 102,
        "height": 28
      },
      "words": "0"
    },
    {
      "location": {
        "width": 430,
        "top": 41,
        "left": 2,
        "height": 64
      },
      "words": "105"
    }
  ]
}
```

二维码识别

- 调用示例

```
 // 二维码识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用二维码识别服务
OCR.getInstance(ctx).recognizeQrcode(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
 {
  "codes_result": [],
  "codes_result_num": 0,
  "log_id": 1423221291786076983
}
```

飞机行程单

- 调用示例

```
 // 行程单识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用行程单识别服务
OCR.getInstance(ctx).recognizeTripTicket(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
 {
    "log_id": 7306800033425229106,
    "words_result_num": 18,
    "words_result": {
        "insurance": "20.00",
        "date": "2019-10-22",
        "allow": "20K",
        "flight": "CA6589",
        "issued_by": "中国国际航空服务有限公司",
        "starting_station": "武汉",
        "fare": "260.00",
        "endorsement": "不得签转改期退转",
        "ticket_rates": "350.00",
        "ck": "5866",
        "serial_number": "51523588676",
        "ticket_number": "7843708871196",
        "fuel_surcharge": "EXEMPT",
        "carrier": "南航",
        "issued_date": "2019-10-30",
        "other_tax": "",
        "fare_basis": "NREOW",
        "id_num": "411201123909020877",
        "destination_station": "合肥",
        "name": "郭达",
        "agent_code": "BJS19197300025",
        "time": "21:25",
        "class": "N",
        "dev_fund": "50.00"
    }
}
```

机动车销售发票

- 调用示例

```
 // 机动车销售发票识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用机动车销售发票识别服务
OCR.getInstance(ctx).recognizeVehicleSellInvoice(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
 {
    "log_id": 283449393728149457,
    "words_result_num": 26,
    "words_result": {
        "InvoiceNum": "00875336",
        "Saler": "深圳市新能源汽车销售有限公司",
        "LimitPassenger": "5",
        "MachineCode": "669745967911",
        "VinNum": "LJLGTCRP1J4007581",
        "TaxRate": "16%",
        "PriceTaxLow": "106100.00",
        "InvoiceDate": "2018-11-29",
        "Price": "￥91465.52",
        "SalerBank": "中国工商银行股份有限公司深圳岭园支行",
        "TaxAuthor": "国家锐务总局深圳市龙岗区税务局第五税务所",
        "ManuModel": "江淮牌HFC7007EYBD6",
        "CertificateNum": "WCH0794J0976801",
        "Purchaser": "苏子潇",
        "VehicleType": "纯电动轿车",
        "InvoiceCode": "14975047560",
        "PriceTax": "壹拾万陆仟壹佰圆整",
        "SalerPhone": "0755-83489306",
        "SalerAddress": "深圳市龙岗区龙岗街道百世国际汽车城",
        "Origin": "安徽省合肥市",
        "EngineNum": "18958407",
        "Tax": "14634.48",
        "PurchaserCode": "5135934475603742222",
        "TaxAuthorCode": "14037589413",
        "SalerAccountNum": "中国工商银行股份有限公司深圳岭园支行",
        "SalerCode": "91449283464582922 78H"
    }
}
```

车辆合格证

- 调用示例

```
 // 车辆合格证识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用车辆合格证识别服务
OCR.getInstance(ctx).recoginzeVehicleCertificate(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数 | 字段 | 是否必选 | 类型 | 说明 | | ---------- | ---- | ------ | ---------- | |Image|是 | File 对象 | 图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式|

- 结果返回

```
{
    "log_id": 14814098736243057,
    "words_result_num": 28,
    "direction": 0,
    "words_result": {
        "ManufactureDate": "2016年10月13日",
        "CarColor": "红",
        "LimitPassenger": "2",
        "EngineType": "WP12.460E50",
        "TotalWeight": "25000",
        "Power": "338",
        "CertificationNo": "WEK29JX98645437",
        "FuelType": "汽油",
        "Manufacturer": "陕西汽车集团有限责任公司",
        "SteeringType": "方向盘",
        "Wheelbase": "3175+1350",
        "SpeedLimit": "105",
        "EngineNo": "1418K129178",
        "SaddleMass": "8600",
        "AxleNum": "3",
        "CarModel": "SX4250MC4",
        "VinNo": "LZGJHYD83JX197344",
        "CarBrand": "陕汽牌",
        "EmissionStandard": "GB17691-2005国V,GB3847-2005",
        "Displacement": "11596",
        "CertificateDate": "2018年11月28日",
        "CarName": "牵引汽车",
        "TyreNum": "10",
        "ChassisID": "",
        "ChassisModel": "",
        "SeatingCapacity": "5",
        "QualifySeal": "1",
        "CGSSeal": "0"
    }
}
```

试卷分析与识别

- 调用示例

```
 // 试卷分析与识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用试卷分析与识别服务
OCR.getInstance(ctx).recognizeExampleDoc(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
{
"results_num": 6,
"log_id": "4488766695474114139",
"img_direction": 0,
"layouts_num": 0,
"results": [
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 124,
    "left": 136,
    "width": 418,
    "height": 65
   },
   "word": "五默写(4分)"
  },
 },
 {
  "words_type": "handwriting",
  "words": {
   "words_location": {
    "top": 195,
    "left": 237,
    "width": 469,
    "height": 104
   },
   "word": "采菊东篱下"
  },
 },
 {
  "words_type": "print",
  "words": {
   "words_location": {
    "top": 241,
    "left": 889,
    "width": 287,
    "height": 52
   },
   "word": "悠然见南山"
  },
 }
]
}
```


手写文字识别

- 调用示例

```
 // 手写文字识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用手写文字识别服务
OCR.getInstance(ctx).recognizeWrittenText(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
 {
    "log_id": 620759800,
    "words_result": [
        {
            "location": {
                "left": 56,
                "top": 0,
                "width": 21,
                "height": 210
            },
            "words": "3"
        }
    ],
    "words_result_num": 1
}
```

护照

- 调用示例

```
 // 护照识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用护照识别服务
OCR.getInstance(ctx).recognizePassport(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回


```
{
  "log_id": 7377468409496932872,
  "words_result_num": 14,
  "words_result": {
    "国家码": {
      "location": {
        "width": 59,
        "top": 200,
        "left": 762,
        "height": 26
      },
      "words": "CHN"
    },
    "护照签发地点": {
      "location": {
        "width": 236,
        "top": 505,
        "left": 558,
        "height": 43
      },
      "words": "山东/SHANDONG"
    },
    "MRZCode2": {
      "location": {
        "width": 1252,
        "top": 797,
        "left": 145,
        "height": 88
      },
      "words": "E898657303CHNSDM0702<<<<<<<<<<<<<<"
    },
    "有效期至": {
      "location": {
        "width": 287,
        "top": 528,
        "left": 955,
        "height": 46
      },
      "words": "20261004"
    },
    "签发机关": {
      "location": {
        "width": 271,
        "top": 583,
        "left": 552,
        "height": 42
      },
      "words": "出入境管理局"
    },
    "MRZCode1": {
      "location": {
        "width": 1201,
```

```
            "top": 781,
            "left": 162,
            "height": 45
        },
        "words": "PONSUN<<JIAJIA<<<<<<<<<<<<<<<<<<<<<<<<<<"
    },
    "护照号码": {
        "location": {
            "width": 237,
            "top": 215,
            "left": 994,
            "height": 48
        },
        "words": "E89865730"
    },
    "签发日期": {
        "location": {
            "width": 279,
            "top": 453,
            "left": 955,
            "height": 46
        },
        "words": "20161005"
    },
    "出生地点": {
        "location": {
            "width": 216,
            "top": 429,
            "left": 564,
            "height": 43
        },
        "words": "山东/SHANDONG"
    },
    "姓名": {
        "location": {
            "width": 159,
            "top": 247,
            "left": 581,
            "height": 34
        },
        "words": "孙嘉佳"
    },
    "姓名拼音": {
        "location": {
            "width": 229,
            "top": 279,
            "left": 578,
            "height": 41
        },
        "words": "SUN,JIAJIA"
    },
    "国籍": {
        "location": {
            "width": 209,
            "top": 366,
            "left": 695,
            "height": 42
        },
        "words": "中国/CHINESE"
    },
    "生日": {
        "location": {
```

```
            "width": 202,
            "top": 382,
            "left": 950,
            "height": 39
        },
        "words": "19950723"
    },
    "性别": {
        "location": {
            "width": 73,
            "top": 357,
            "left": 570,
            "height": 34
        },
        "words": "男/M"
    }
  }
}
```

户口本

- 调用示例

```
 // 户口本识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用户口本识别服务
OCR.getInstance(ctx).recognizehousehold(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
 {
 "words_result": {
  "Nation": {
   "words": "汉族"
  },
  "Sex": {
   "words": "女"
  },
  "Birthday": {
   "words": "1994年7月27日"
  },
  "Date": {
   "words": "2017年7月27日"
```

```
  },
  "BirthAddress": {
   "words": "四川省"
  },
  "Name": {
   "words": "王燕"
  },
      "FormerName ": {
   "words": "王佳"
  },
  "HouseholdNum": {
   "words": "000007670"
  },
  "WWToCity": {
   "words": "由久居"
  },
  "WWHere": {
   "words": "1994年07月27日因出生迁来"
  },
  "CardNo": {
   "words": "112102199407273123"
  },
  "Education": {
   "words": "初中毕业"
  },
  "Relationship": {
   "words": "独生女"
  },
  "Height": {
   "words": "170厘米"
  },
      "Career": {
   "words": "无"
  },
  "WorkAddress": {
   "words": "无"
  },
  "Hometown": {
   "words": "四川省"
  }
      "OtherAddress": {
   "words": "四川省乐山市"
  }
      "Belief": {
   "words": "佛教"
  }
      "BloodType": {
   "words": "A型"
  }
       "MaritalStatus": {
   "words": "未婚"
  }
      "VeteranStatus": {
   "words": "无"
  }
 },
 "log_id": "1407164137607266304",
 "words_result_num": 22
}
```


通用机打发票

- 调用示例

```
 // 通用机打发票识别参数设置 OcrRequestParams param = new OcrRequestParams();
// 设置image参数 param.setImageFile(new File(filePath));
// 调用通用机打发票识别服务
OCR.getInstance(ctx).recognizeMachineInvoice(param, new OnResultListener<OcrResponseResult>() {
    @Override
    public void onResult(OcrResponseResult result) {
        listener.onResult(result.getJsonRes());
    }

    @Override
    public void onError(OCRError error) {
        listener.onResult(error.getMessage());
    }
});
```

- 参数

字段	是否必选	类型	说明
Image	是	File对象	图像数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

- 结果返回

```
{
  "log_id": 4423022131715883558,
  "direction": 0,
  "words_result_num": 22,
  "words_result": {
     "City": "",
     "InvoiceNum": "01445096",
     "SellerName": "百度餐饮店",
     "IndustrySort": "生活服务",
     "Province": "广东省",
     "CommodityAmount": [
        {
           "word": "183.00",
           "row": "1"
        }
     ],
     "InvoiceDate": "2020年07月28日",
     "PurchaserName": "中信建投证券股份有限公司",
     "CommodityNum": [],
     "InvoiceCode": "144001901511",
     "CommodityUnit": [],
     "SheetNum": "",
     "PurchaserRegisterNum": "9144223008453480X9",
     "Time": "",
     "CommodityPrice": [],
     "AmountInFiguers": "183.00",
     "AmountInWords": "壹佰捌拾叁元整",
     "CheckCode": "610421198204210613O1",
     "TotalTax": "183.00",
     "InvoiceType": "广东通用机打发票",
     "SellerRegisterNum": "610421198204210613O1",
     "CommodityName": [
        {
           "word": "餐费",
           "row": "1"
        }
     ]
  }
}
```


FAQ

Q：使用demo为何发生 App identifier unmatch 错误？

A：使用demo工程请在百度智能云控制台中绑定demo工程的包名（com.baidu.ocr.demo），然后选择license或者ak，sk方式授权。

Q: demo工程为何提示token还未获取成功，无法使用识别功能？

A：识别需要token作为参数，token需要通过网络获取，网络环境较差的情况下返回较慢。

Q：关于身份证识别的两种模式？

A：身份证识别依赖本地库，如果您不需要本地能力可以在传入activity的参数中选择关闭，并且移除ui模块中的本地so文件和ui模块中assets下的模型文件。

错误码表

验证错误

错误码	错误信息	说明	备注
110	Access token invalid or no longer valid	Access Token过期失效	请重新获得有效的Token
283501	License file check error	授权文件不匹配	请在控制台中配置正确的包名，并确认使用了正确的授权文件
283502	App identifier unmatch	BundleId不匹配	请在控制台中配置正确的包名，并确认使用了正确的授权文件
283503	License file not exists	请确认aip-ocr.license文件存在于assets文件夹中	
283504	Network error	网络请求失败	请授权App网络权限并保证网络通畅
283505	Server illegal response	服务器返回数据异常	
283506	Load jni so library error	JNI加载异常	请确认开发包中的so库被正确加载
283507	App signsure md5 unmatch	签名MD5不匹配	请在控制台中配置正确的签名MD5，并确认使用了正确的授权文件
283601	Server authentication error	身份验证错误。	请在控制台中配置应用，并确认填写了正确的AK/SK，或使用了正确的授权文件
283602	Authentication time error	时间戳不正确，可能是设备时间异常。	请确保不要改变调用设备的本地时间
283604	App identifier unmatch	错误的PackageName或者BundleId	请在控制台中配置正确的包名，并确认使用了正确的授权文件
283700	Server internal error	服务器内部错误	您可以在工单系统中提交错误信息中的logId，我们将尝试帮您排查错误原因

服务错误

错误码	错误信息	描述
216015	module closed	模块关闭
216100	invalid param	非法参数
216101	not enough param	参数数量不够
216102	service not support	业务不支持
216103	param too long	参数太长
216110	appid not exist	APP ID不存在
216111	invalid userid	非法用户ID
216200	empty image	空的图片
216201	image format error	图片格式错误
216202	image size error	图片大小错误
216300	db error	DB错误
216400	backend error	后端系统错误
216401	internal error	内部错误
216500	unknown error	未知错误
216600	id number format error	身份证的ID格式错误
216601	id number and name not match	身份证的ID和名字不匹配
216630	recognize error	识别错误
216631	recognize bank card error	识别银行卡错误（通常为检测不到银行卡）
216632	ocr	unknown error
216633	recognize idcard error	识别身份证错误（通常为检测不到身份证）
216634	detect error	检测错误
216635	get mask error	获取mask图片错误
282000	logic internal error	业务逻辑层内部错误
282001	logic backend error	业务逻辑层后端服务错误
282100	image transcode error	图片压缩转码错误

SDK合规使用指南

前言

2023年2月工信部发布的**《工业和信息化部关于进一步提升移动互联网应用服务能力的通知》**对SDK、个人信息保护、服务体验等方面提出了具体要求，同时为了帮助开发者向最终用户提供相应功能及服务，特此起草本SDK合规使用指导。 您作为开发者为最终用户提供服务，需知悉并确认将遵守适用的法律法规和相关的标准规范，履行个人信息保护义务，并遵循合法、正当、必要和诚信的原则处理用户个人信息，包括但不限于《中华人民共和国个人信息保护法》、《中华人民共和国网络安全法》、《中华人民共和国数据安全法》以及其他适用的法律法规和相关的标准规范。 此文档用于帮助您更好地了解百度OCR采集SDK并合规使用本SDK服务，仅适用于开发者的业务区域为中国大陆地区的场景。

特别提示

本**《SDK合规指南》**是对本 SDK 的合规性和安全性描述与要求仅为我们向您提供的服务说明及使用建议，不构成也不应被视为我们对于任何法律法规及政策文件的及时的、完整的、全面的、甚至完全准确的分析，亦不构成我们对百度OCR采集SDK产品和/或服务的承诺和保证。本《指南》不能作为判断您与您所开发、运营的 App 是否满足合规与安全要求的可依赖的标准，亦不构成我们对前述事宜作出的任何担保或保证，您应当自行、独立地对前述 App 承担合规与安全责任。

目录

- SDK收集个人信息的频次、精度、使用目的、场景及对应选择的配置方式、示例

- SDK所需系统权限与功能关系，以及权限申请时机
- SDK扩展业务功能介绍及对应关闭的配置方式、示例
- SDK初始化及各项业务功能接口合规调用时机
- 联系我们

1. SDK收集个人信息不同频次、精度使用目的、场景及对应选择的配置方式、示例

为了帮助开发者向最终用户提供相应功能及服务，当最终用户使用相应功能及服务时，我们会通过开发者应用向系统申请最终用户设备的相应权限。开发者应确保最终用户可以随时通过取消系统授权开发者应用获取相应设备权限或其他开发者应用提供的授权设置，停止我们对最终用户个人信息的收集，之后最终用户可能将无法使用基于相应个人信息而提供的相关服务或功能，或者无法达到基于相应个人信息提供的相关服务拟达到的效果，但不会影响最终用户正常使用OCR采集SDK的其他不基于相应个人信息即可实现的业务功能。各项功能及服务涉及的个人信息包括：

序号	功能服务	个人信息类型	收集方式	适用系统版本
1.	为帮助开发者通过对身份证照片进行文字识别的方式，获取最终用户的姓名、身份证件号码作为待核验数据，以便实现无需最终用户手动输入即可向最终用户提供权威数据源身份核验功能	身份证照片	SDK直接采集	iOS及Android

注：更多个人信息收集、使用、处理的细节请参考OCR采集SDK隐私政策

2. SDK所需系统权限与功能关系，以及权限申请时机

为了保证最终用户能正常使用OCR采集SDK相应功能及服务，我们会通过开发者应用向系统申请最终用户设备的以下系统设备权限，申请前我们会征询最终用户的同意，最终用户可以选择“允许”或“禁止”权限申请。经过最终用户的授权后我们会开启相关权限，最终用户可以随时在系统中取消授权，最终用户取消授权会导致最终用户无法使用相关的业务功能，但不会导致最终用户无法使用其他业务功能。各项功能及功能对设备权限的调用情况如下：

Android系统版本 |设备权限 |功能及服务 |权限授权方式| | --- | --- | --- | |读取/写入外部存储卡| 用于存储OCR采集SDK的模型信息、配置文件、以及保存安全风险诊断信息；从相册中读取身份证照片，以便识别身份证号和姓名，以实现权威数据源身份核验功能| 授权方式由设备系统开发方以及开发者应用决定；当最终用户同意向开发者应用授予该权限时开启| |访问网络 |用于连接网络，进行数据传输，用作活体及身份核验| 授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启| |获取网络状态 |用于获取当前网络信息，进行安全风险诊断，并保护网络传输 |授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启| |打开相机/摄像头| 实现OCR图片采集、身份证照片采集，以便进行活体及身份校验| 授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启| |访问相册 |从相册中读取身份证照片，以便识别身份证号和姓名，以实现权威数据源身份核验功能| 授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启|

iOS系统版本 |设备权限 |功能及服务 |权限授权方式| | --- | --- | --- | |打开相机/摄像头| 实现OCR图片采集、身份证照片采集功能，以便进行活体及身份校验 |授权方式由设备系统开发方以及开发者应用决定；当最终用户同意向开发者应用授予该权限时开启| |访问相册 |从相册中读取身份证照片，以便识别身份证号和姓名，以实现权威数据源身份核验功能 |授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启| 注：在不同设备中，权限显示方式及关闭方式可能有所不同，具体请最终用户参考设备及系统开发方说明或指引。

3. SDK初始化及各项业务功能接口合规调用时机

为了避免您的应用在未获取用户的同意前SDK提前处理用户的个人信息，我们提供了SDK初始化的接口，请保证您的应用获取用户同意后才能调用此接口初始化SDK。

- **iOS：**[[AipOcrService shardService]getTokenSuccessHandler:^(NSString token) { //获取到身份证质量控制token// } failHandler:^(NSError error) {}]; // 利用获取到的token 完成 IdcardQualityAdaptor的初始化 IdcardQualityAdaptor *idcard = [[IdcardQualityAdaptor alloc]init]; [idcard initWithToken:token];
- **Android**：OCR.getInstance().initAccessToken(new OnResultListener() { @Override public void onResult(AccessToken result) { // 调用成功，返回AccessToken对象 String token = result.getAccessToken(); } @Override public void onError(OCRError error) { // 调用失败，返回OCRError子类SDKError对象 } }, getApplicationContext());

4. 联系我们

OCR采集SDK的成长离不开各方开发者及最终用户的共同努力，我们非常感谢开发者及最终用户对OCR采集SDK数据更新、使用反馈方面的贡献。开发者及最终用户可以通过百度云工单反馈开发者及最终用户对OCR采集SDK产品和服务的建议以及在使用过程中遇到的问题，以帮助我们优化产品功能及服务，使更多用户更加便捷的使用我们的产品和服务。

iOS SDK

简介

本文档主要介绍OCR iOS SDK的安装和使用。在使用本文档前，您需要先了解Optical Character Recognition(OCR)的基础知识，并已经开通了OCR服务。

支持的系统和硬件版本：

- iOS: 8.0 以上
- 架构：armv7 armv7s arm64

> 其中 身份证本地扫描 IdcardQuality.framework不支持模拟器，如需使用模拟器调试，可自行剔除此模块。

```
 #if !TARGET_IPHONE_SIMULATOR
IdcardQualityAdaptor *idcard = [[IdcardQualityAdaptor alloc]init];
[idcard initWithToken:token];
#endif
```

接口能力

远程API能力

接口名称	接口能力简要描述
通用文字识别	识别图片中的文字信息
通用文字识别（高精度版）	更高精度地识别图片中的文字信息
通用文字识别（含位置信息版）	识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（高精度含位置版）	更高精度地识别图片中的文字信息（包含文字区域的坐标信息）
通用文字识别（含生僻字版）	识别图片中的文字信息（包含对常见字和生僻字的识别）
网络图片文字识别	识别一些网络上背景复杂，特殊字体的文字
身份证识别	识别身份证正反面的文字信息，并支持端上数据加密
银行卡识别	识别银行卡的卡号并返回发卡行和卡片性质信息，并支持端上数据加密
驾驶证识别	识别机动车驾驶证所有关键字段
行驶证识别	识别机动车行驶证所有关键字段
车牌识别	对小客车的车牌进行识别
营业执照识别	对营业执照进行识别
通用票据识别	对各类票据图片（医疗票据，保险保单等）进行文字识别，并返回文字在图片中的位置信息
增值税发票识别	对增值税普票、专票、卷票、电子发票进行识别
出租车发票识别	识别全国各大城市出租车票
VIN码识别	对车辆挡风玻璃处的车架号码进行识别
火车票识别	对红、蓝火车票进行识别
数字识别	对图片中的数字进行提取和识别
二维码识别	对二维码、条形码中对应的文字内容进行识别
飞机行程单识别	对飞机行程单中的姓名、始发站、目的站、航班号、日期、票价字段进行结构化识别
机动车销售发票识别	对机动车销售发票的号码、代码、日期、价税合计等字段进行结构化识别
车辆合格证识别	对车辆合格证的编号、车架号、排放标准、发动机编号等字段进行结构化识别
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果
手写文字识别	对手写汉字或手写数字进行识别
护照识别	支持对中国大陆居民护照的资料页进行结构化识别
户口本识别	对户口本的出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码等字段进行识别别
通用机打发票识别	对国家/地方税务局发行的横/竖版通用机打发票的号码、代码、日期、合计金额、类型等字段进行结构化识别
医疗费用明细识别	支持识别全国医疗费用明细识别
网约车行程单识别	对国家/地方税务局发行的横/对各大主要服务商的网约车行程单进行结构化识别
磅单识别	结构化识别磅单的车牌号、打印时间、毛重、皮重、净重、发货单位、收货单位、单号8个关键字段，现阶段仅支持识别印刷体磅单

本地质量控制能力

除了包含远程API调用能力外，iOS SDK中还集成了身份证识别的本地质量控制能力，提供给开发者本地检测身份证的功能。SDK可以先行在本地完成身份证的预判断，然后上传至服务端识别，以达成“自动扫描识别”的功能，使用时可实时检测取景

框中是否包含身份证，是否存在模糊、欠/过曝等情况，并提示用户矫正，提高图片采集质量，提升识别准确率。SDK下载

版本更新记录

上线日期	版本号	更新内容
2025.5.13	3.0.9	通过移除Bitcode，修复 App Store上线审核报错问题
2023.7.14	3.0.8	更新身份证的云端识别模型返回字段，新增返回“risk_type"字段，返回风险类型
2022.6.9	3.0.7	修复身份证、银行卡的采集质控模块鉴权问题
2022.1.7	3.0.6	更新了身份证，银行卡数据加密功能，新增磅单识别、网约车行程单、医疗费用明细识别
2021.8.5	3.0.5	新增二维码、飞机行程单、机动车销售发票、车辆合格证、试卷分析与识别、手写识别、护照、户口本、通用机打发票识别功能
2021.2.2	3.0.4	新增识别增值税发票、出租车发票、VIN码、火车票和数字识别功能
2018.1.15	3.0.1	修复身份证扫描页面没有初始化完成，快速点击左上角返回按钮可能会闪退的问题
2017.11.23	3.0.0	新增高精度、票据、营业执照识别；优化工程结构，打包方式
2017.9.21	2.1.1	身份证扫描提示文案调整
2017.8.15	2.1.0	新增行驶证、驾驶证、车牌识别
2017.7.14	2.0.0	新增bitcode支持；安全性更新；新增身份证质量控制 IdcardQuality.framework
2017.6.30	1.1.1	1.对SDK的安全性作出优化 2.对端上身份证输入校验功能进行升级，该功能暂时不可用
2017.6.20	1.2.0	新增身份证质量控制 IdcardQuality.framework
2017.5.18	1.1.0	新增网图、生僻字接口
2017.4.27	1.0.1	新增模拟器支持
2017.3.16	1.0.0	在线OCR第一版！

快速入门

SDK工程结构

```
aip-ocr-ios-sdk-$version
  lib                          // 需要引用的框架
    |-AipBase.framework          // 鉴权等基础框架包，必须引入
    |-AipOcrSdk.framework        // OCR服务，UI框架包，必须引入
    |-IdcardQuality.framework    // 身份证本地质量控制框架包，必须引入
  AipOcrDemo                   // Demo 工程
  AipOcrSdk                    // AipOcrSdk.framework源码
```

AipOcrDemo工程Link了AipBase.framework、AipOcrSdk.framework、IdcardQuality.framework三个框架。

AipOcrSdk工程Link了AipBase.framework、IdcardQuality.framework两个框架

注意 !!! 以上框架为Dynamic框架，请务必使用Embedded Binary方式嵌入!!!

其中

- AipBase.framework中包含了鉴权相关服务，必须引入。
- AipOcrSdk.framework中包含OCR服务、UI，必须引入。
- IdcardQuality.framework包含了身份证本地质量控制的功能，必须引入。
- AipOcrSdk目录下的工程为AipOcrSdk.framework的源码；开发者可自行根据需求修改并引用。

如何运行Demo工程

iOS SDK提供了一个可快速运行的Demo工程，建议首先运行以下Demo工程，其中包含了使用SDK的主要步骤。

运行步骤如下：

1. 在官网下载iOS SDK 压缩包
2. 解压缩，双击打开 AipOcrDemo/AipOcrDemo.xcodeproj
3. 在管理控制台中新建文字识别应用，配置BundleId为AipOcrDemo的BunldeId(默认为`com.baidu.AipOcrDemo`)
4. 在AipOcrDemo工程中`AipOcrDemo/ViewController.m` `viewDidLoad`方法中配置相应管理控制台中新建的应用的Api Key, Secret Key
5. 运行AipOcrDemo
6. 上传至AppStore前，必须使用lipo命令移除模拟器架构。请参考 FAQ-2。

身份验证与安全

百度AIP开放平台使用OAuth2.0授权调用开放API，调用API时必须在URL中带上accesss_token参数。AccessToken可用AK/SK或者授权文件的方式获得。

OCR iOS SDK提供了以下两种AccessToken管理方法。

API Key / Secret Key

此种身份验证方案使用AK/SK获得AccessToken，缓存在本地。

虽然SDK对网络传输的敏感数据进行了二次加密，但由于AK/SK是明文填写在代码中，在移动设备中可能会存在AK/SK被盗取的风险。有安全考虑的开发者可使用第二种授权方案。

使用步骤：

1. 在管理控制台中配置OCR应用
2. 复制应用的Api Key（简称AK） 和 Secret Key（简称SK），初始化`AipOcrService`单例：

   [[AipOcrService shardService] authWithAK:@"Api Key" andSK:@"Secret Key"];

授权文件（安全模式）

此种身份验证方案使用授权文件获得AccessToken，缓存在本地。***建议有安全考虑的开发者使用此种身份验证方式。***

在您的移动APP分发出去之后，APP存在被反编译的可能，所以直接将AK / SK 置于APP源码之中，存在被盗取的风险。采用授权文件的身份验证方法，可有效保护AK/SK在移动设备中的安全。攻击者即使拦截了流量，盗取了授权文件，也难以盗用您的配额。

使用步骤：

1. 在官网中配置应用
2. 下载对应应用的授权文件（默认名字为aip.license）
3. 将授权文件添加至XCode工程（配置为资源并拷贝，Target -> Build Phases -> Copy Bundle Resource 中添加该文件）
4. 读取授权文件原始字节，`NSData`格式，初始化`AipOcrService`单例：

// 示例 // 若未添加至主工程，则[NSBundle mainBundle]修改为对应bundle NSString *licenseFile = [[NSBundle mainBundle] pathForResource:@"aip" ofType:@"license"]; NSData licenseFileData = [NSData dataWithContentsOfFile:licenseFile]; [[AipOcrService shardService] authWithLicenseFileData:licenseFileData];

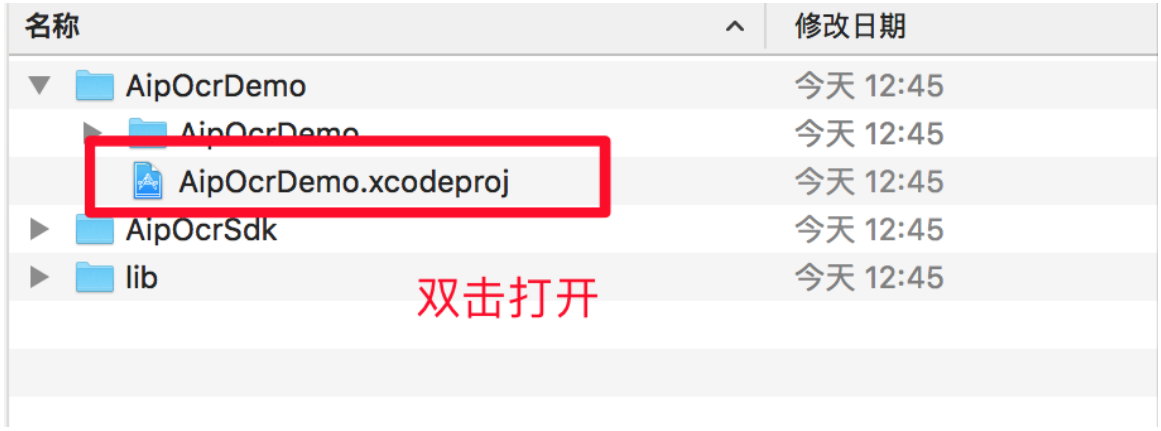
SDK集成图文教程

如何运行demo工程

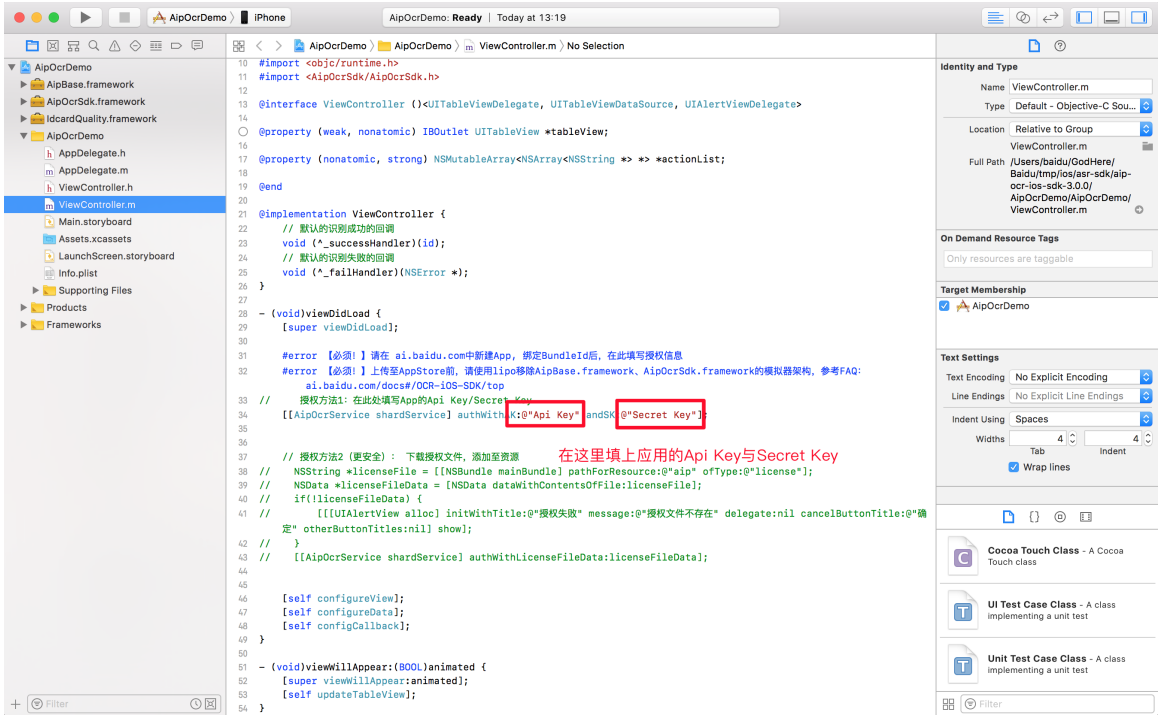
下载最新版本的SDK，打开Demo工程；

下载地址：http://ai.baidu.com/sdk#ocr



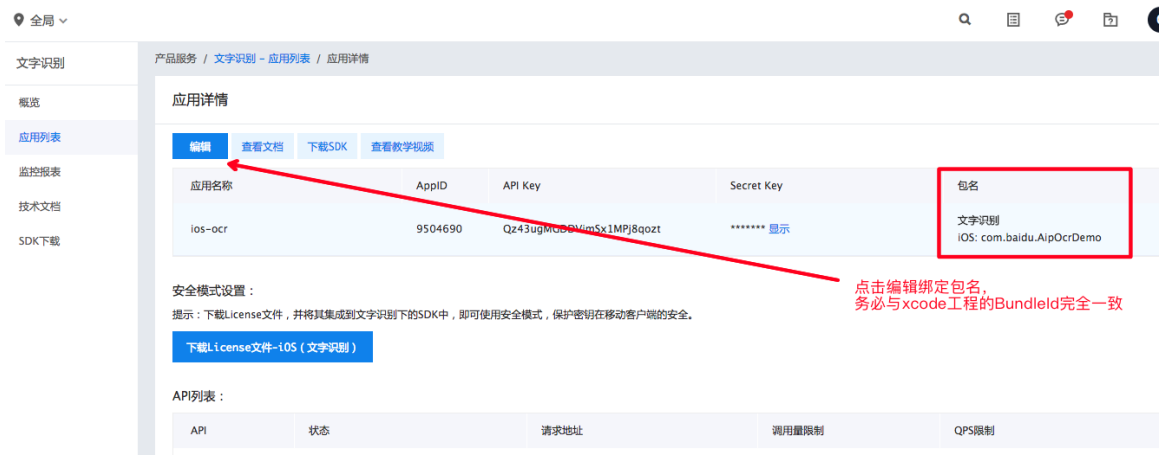


在AipOcrDemo/AipOcrDemo/ViewController.m中填写Api Key, Secret Key。注释去两个人工定义error：

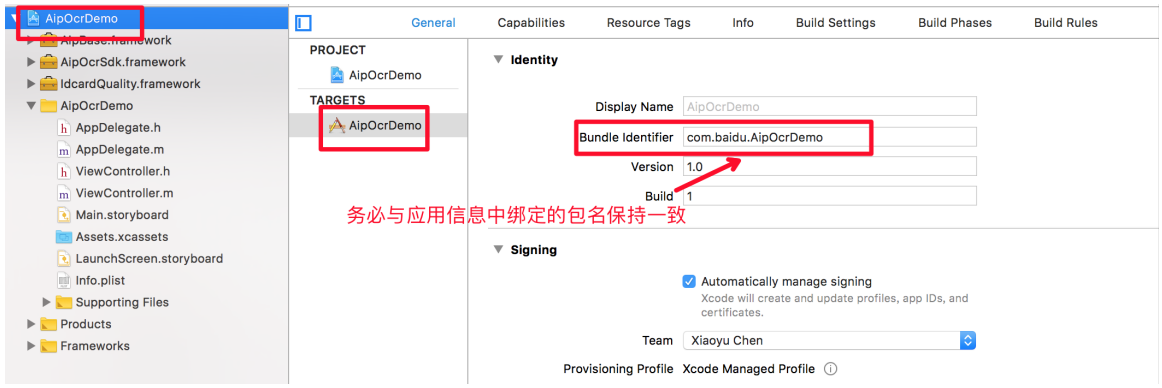


在官网新建文字识别应用，获得Api Key, Secret Key。这步可参考图文教程http://ai.baidu.com/docs#/Begin/top

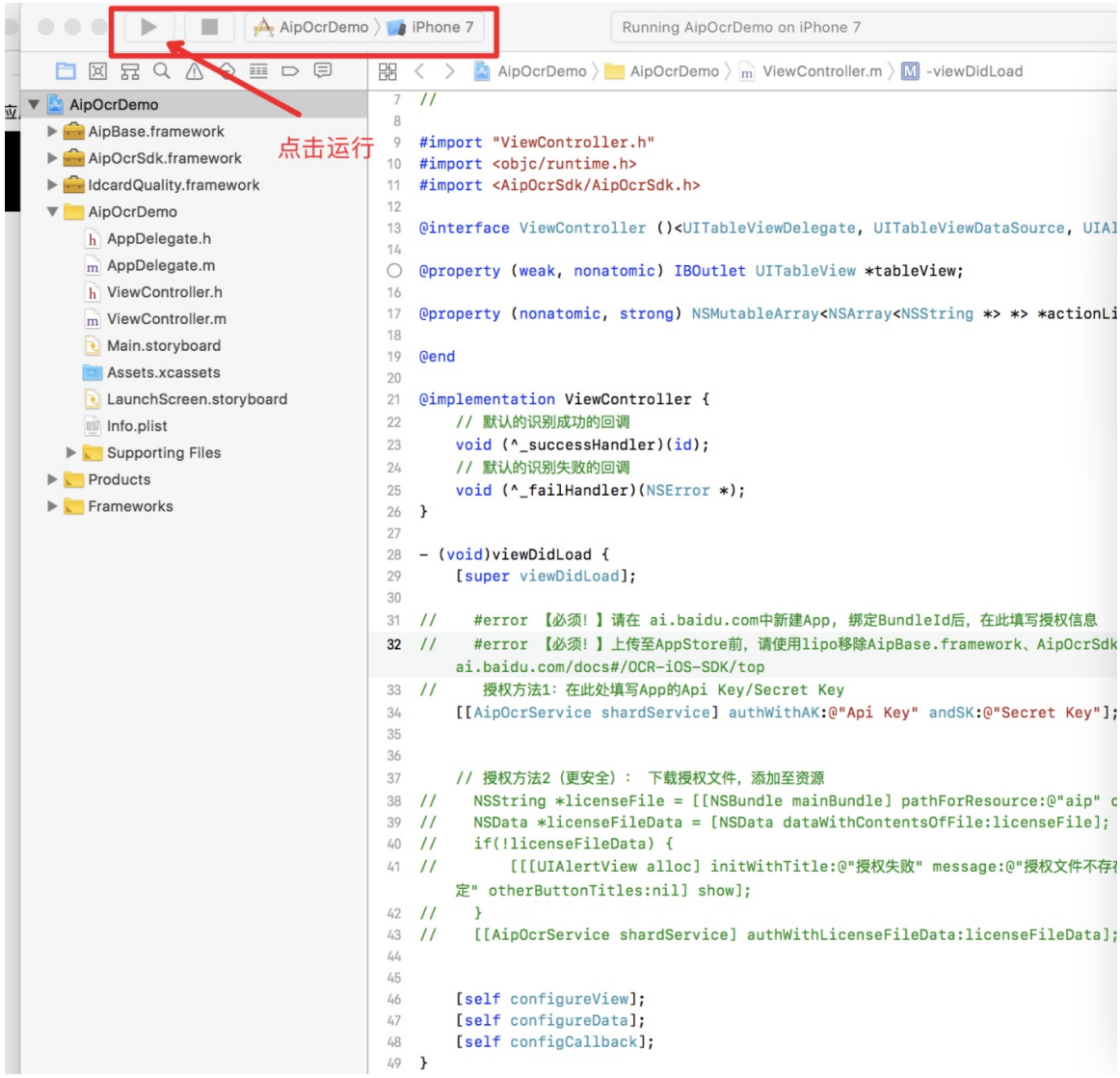
绑定包名



务必与官网应用信息中显示的包名保持一致

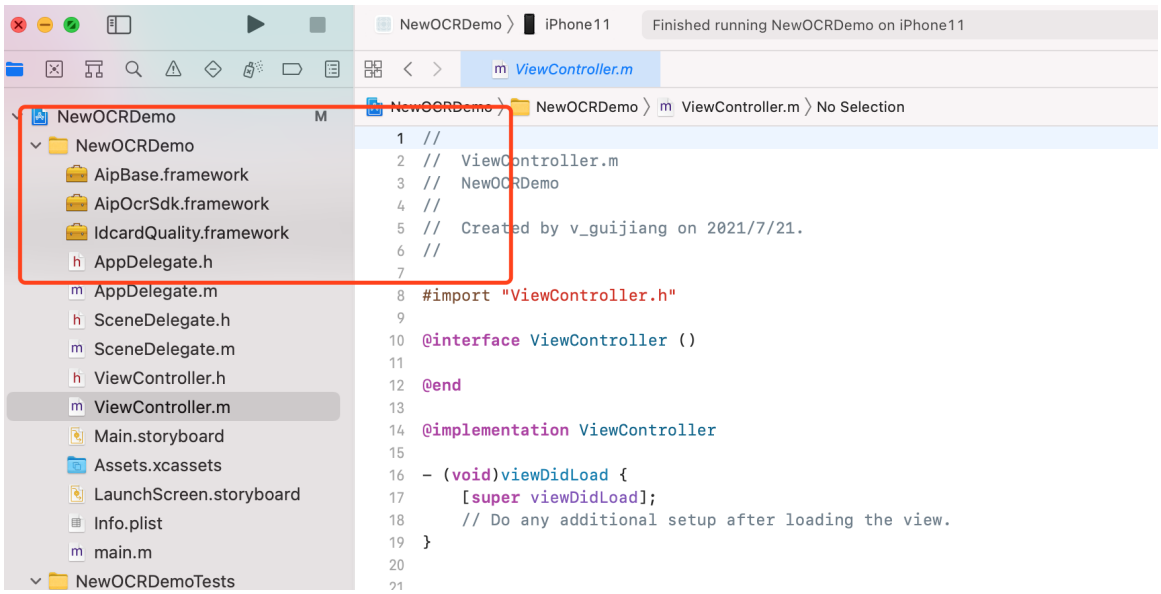


点击运行，搞定！

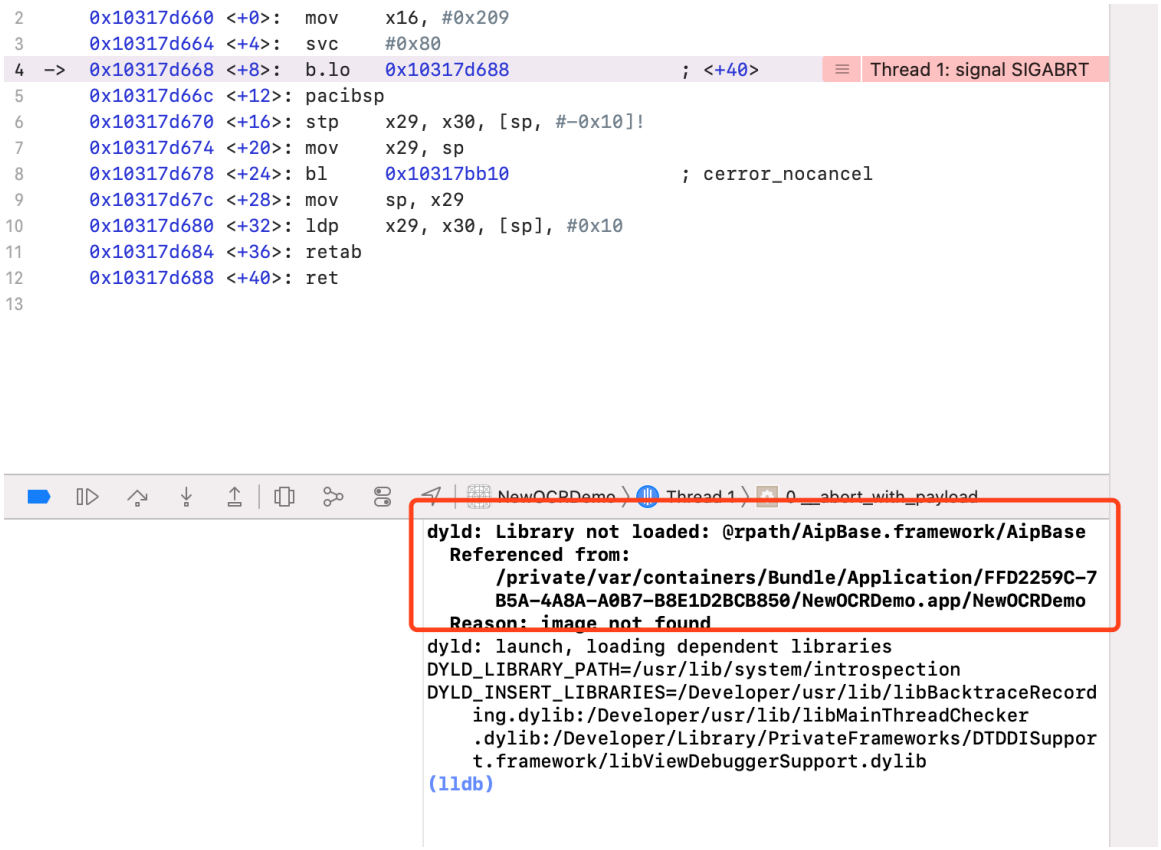


如何集成到自己的工程中

1.这三个库拖入到目标项目中（AipBase.framework、AipOcrSdk.framework、IdcardQuality.framework）

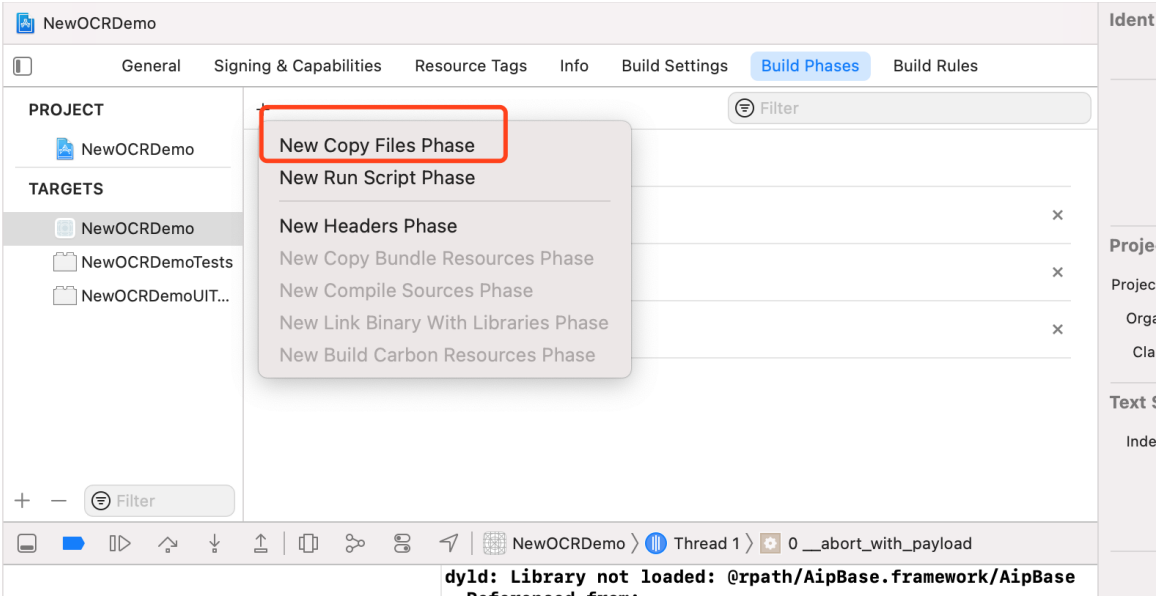


2.运行报以下错误

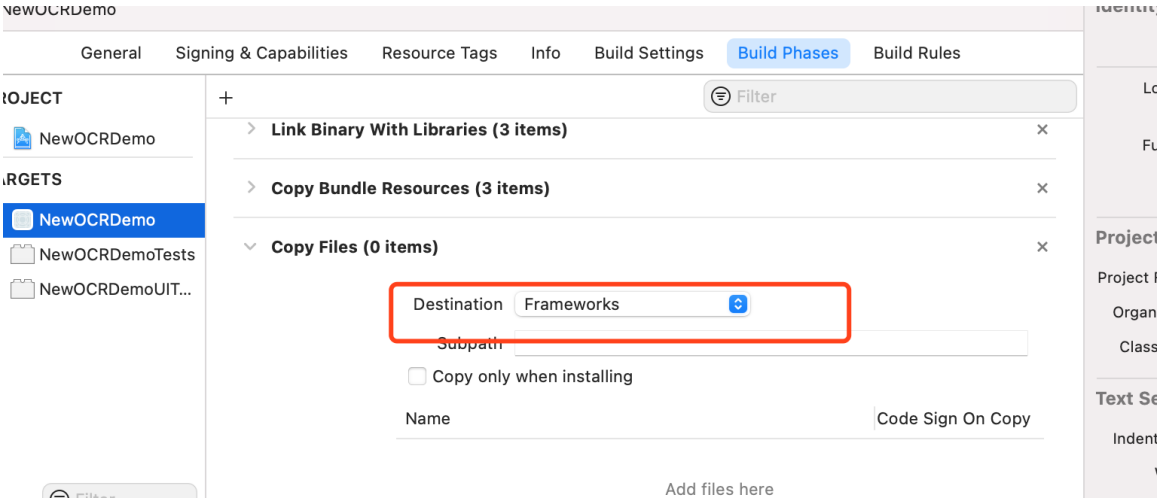


3.安装以下步骤解决以上错误

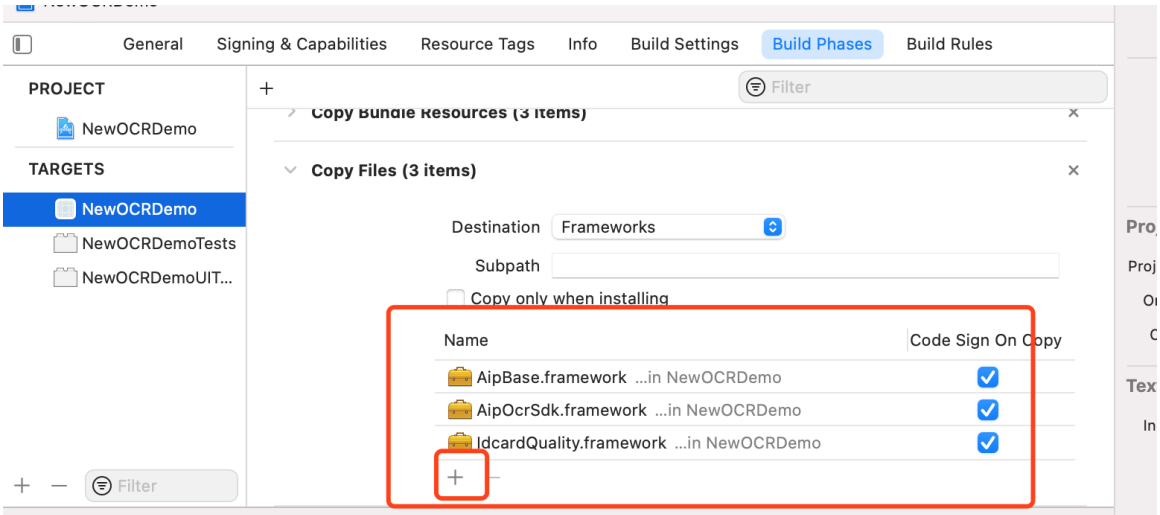
a. 添加New Copy Files Phase



b.修改destination 为frameworks



c.点击加号添加所加的库

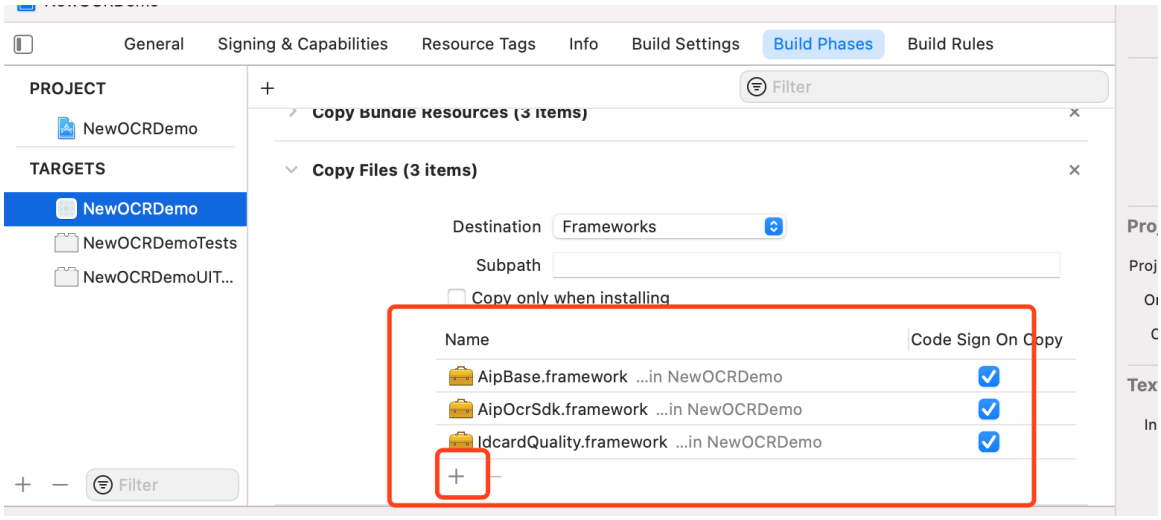


d.运行如果遇到下面错误

Building for iOS Simulator, but the linked and embedded framework '****.framework' was built for iOS + iOS Simulator.

解决方法Build Setting > Build Options > Validate WorkSpace设置为 true；

运行之后再改回去也可以，这个问题是xcode11以后有的问题



4.注意事项

a.bundleID 和 Api Key 和 Secret Key 一定要对应

b.info.plist文件添加相机权限 Privacy - Camera Usage Description

c.参考ViewController.m 添加需要识别的项目

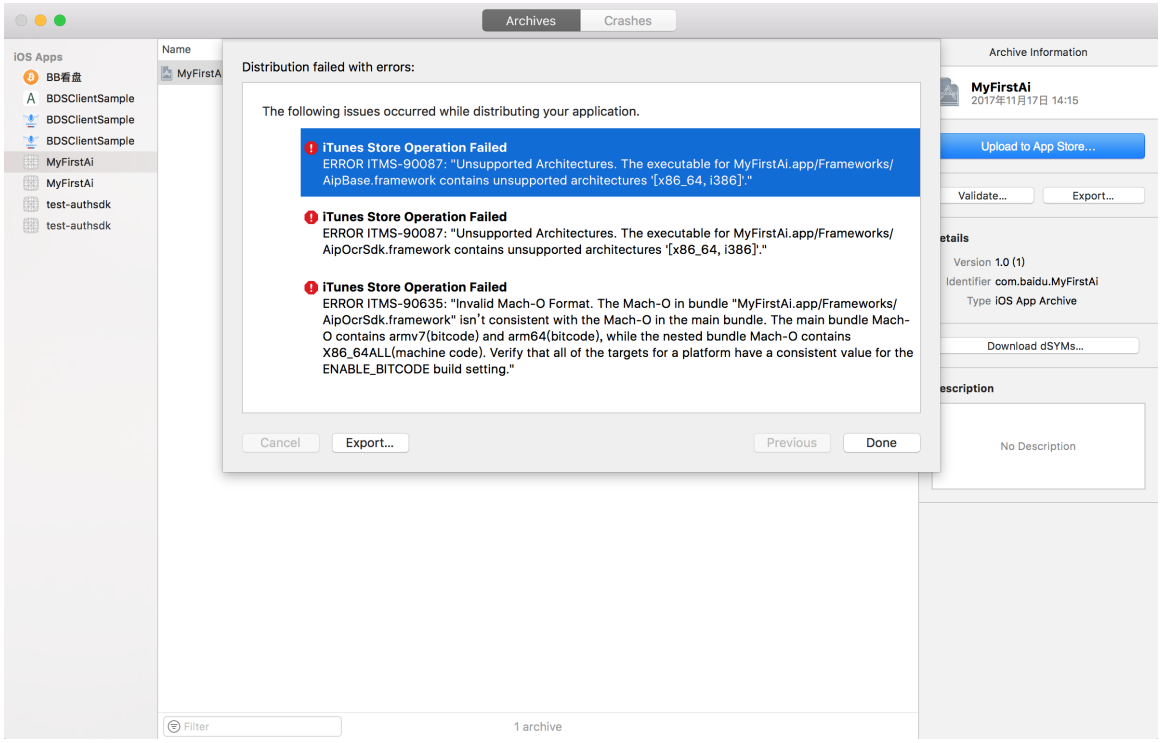
d.'UIAlertView已弃用，无法用于基于UIScene的应用程序，请使用UIAlertController！'

如何上传到AppStore

> 使用动态库有诸多优势，但若在动态库包含多个架构，在上传AppStore前需要删除模拟器架构

为了方便开发者调试，我们使用lipo工具合并了模拟器和真机的架构。所以在上传前AppStore前，必须使用lipo移除模拟器的架构，否则会报错：

> ERROR ITMS-90087: "Unsupported Architectures. The executable for MyFirstAi.app/Frameworks/AipBase.framework contains unsupported architectures '[x86_64, i386]'."



我们使用lipo工具移除模拟器架构即可。详细的操作方法已经在文档最后的FAQ中列出。

```
➜ lib lipo -info AipBase.framework/AipBase
Architectures in the fat file: AipBase.framework/AipBase are: i386 x86_64 armv7 arm64
➜ lib lipo -remove x86_64  AipBase.framework/AipBase -o AipBase.framework/AipBase
➜ lib lipo -remove i386  AipBase.framework/AipBase -o AipBase.framework/AipBase
➜ lib lipo -remove x86_64 AipOcrSdk.framework/AipOcrSdk -o AipOcrSdk.framework/AipOcrSdk
➜ lib lipo -remove i386 AipOcrSdk.framework/AipOcrSdk -o AipOcrSdk.framework/AipOcrSdk
```

当然开发者可以根据自己的需求先行把需要的库分成多个架构，按需使用。

SDK接口调用

相机接口

具体返回格式见 数据接口

操作步骤：

1. 身份验证：调用 `[[AipOcrService shardService] authWithAK:SK` 或其他验证方法；
2. 初始化对应的ViewController，设置对应的block。 如 身份证本地扫描识别：

   UIViewController *vc = [AipCaptureCardVC ViewControllerWithCardType:CardTypeLocalIdCardFont andImageHandler:^(UIImage* image) { // 成功扫描出身份证 [[AipOcrService shardService] detectIdCardFrontFromImage:image withOptions:nil successHandler:^(id result){ // 在成功回调中，保存图片到系统相册 UIImageWriteToSavedPhotosAlbum(image, nil, nil, (__bridge void *)self); // 打印出识别结果 NSLog(@"%@", result); } failHandler:_failHandler]; }]; // 展示ViewController [self presentViewController:vc animated:YES completion:nil];
3. 在合适的地方启动ViewController: 如`[self presentViewController:vc animated:YES completion:nil]`

数据接口

该调用方法传入需要识别的UIImage，异步识别，识别完成之后，回调返回识别结果。

主要类为AipOcrService类，使用单例`[AipOcrService sharedService]`来调用相关接口即可。

操作步骤：

1. 身份验证：调用 `[[AipOcrService shardService] authWithAK:SK` 或其他验证方法；
2. 调用相应接口

- 通用文字识别（基础版、不含位置信息） `detectTextBasicFromImage`
- 通用文字识别（含位置信息） `detectTextFromImage`
- 通用文字识别（高精度、不含位置信息）`detectTextAccurateBasicFromImage`
- 通用文字识别（高精度、含位置信息）`detectTextAccurateFromImage`
- 通用文字识别 （含生僻字）`detectTextEnhancedFromImage`
- 网图识别 `detectWebImageFromImage`
- 身份证正面识别`detectIdCardFrontFromImage`
- 身份证背面识别`detectIdCardBackFromImage`
- 银行卡识别`detectBankCardFromImage`
- 驾驶证识别 `detectDrivingLicenseFromImage`
- 行驶证识别 `detectVehicleLicenseFromImage`
- 车牌识别 `detectPlateNumberFromImage`
- 营业执照识别 `detectBusinessLicenseFromImage`
- 通用票据识别 `detectReceiptFromImage`

- 增值税发票识别 `detectValueAddedTaxImage`
- 出租车票识别 `detectTaxiReceiptImage`
- VIN码识别 `detectVinCodeImage`
- 火车票识别 `detectTrainTicketImage`
- 数字识别 `detectNumbersImage`

所有回调函数均在后台线程中被调用，如需在主线程中操作，请使用`[[NSOperationQueue mainQueue] addOperationWithBlock]`patch到主线程中，示例参考demo工程。

以下以通用文字识别调用为例，其他接口的参数与返回可参考API文档

```
 NSDictionary *options = @{@"language_type": @"CHN_ENG", @"detect_direction": @"true"};
[[AipOcrService shardService] detectTextFromImage:finalImage withOptions:options successHandler:^(id result) {
 // 成功识别的后续逻辑
} failHandler:^(NSError *err) {
 // 失败的后续逻辑
}];
```

options参数详情

参数	是否必选	类型	可选值范围	说明
image（已由参数替代）	true	string	-	图像数据，base64编码，要求base64编码后大小不超过1M，最短边至少15px，最长边最大2048px,支持jpg/png/bmp格式
recognize_granularity	false	string	big、small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
mask	false	string	-	表示mask区域的黑白灰度图片，白色代表选中，base64编码
language_type	false	string	CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP	识别语言类型，默认为CHN_ENG。可选值包括： - CHN_ENG：中英文混合； - ENG：英文； - POR：葡萄牙语； - FRE：法语； - GER：德语； - ITA：意大利语； - SPA：西班牙语； - RUS：俄语； - JAP：日语
detect_direction	false	boolean	true、false	是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括: - true：检测朝向； - false：不检测朝向。
detect_language	false	string	true、false	是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）
classify_dimension	false	string	lottery	分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。 lottery：彩票分类，设置detect_direction有助于提升精度
vertexes_location	false	string	true、false	是否返回文字外接多边形顶点位置，不支持单字位置。默认为false

- 结果返回

字段	必选	类型	说明
direction	否	int32	图像方向，当detect_direction=true时存在。 - -1:未定义， - 0:正向， - 1: 逆时针90度， - 2:逆时针180度， - 3:逆时针270度
log_id	是	uint64	唯一的log id，用于问题定位
words_result	是	array()	定位和识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+vertexes_location	否	array()	当前为四个顶点: 左上，右上，右下，左下。当vertexes_location=true时存在
++x	是	uint32	水平坐标（坐标0点为左上角）
++y	是	uint32	垂直坐标（坐标0点为左上角）
+location	是	array()	位置数组（坐标0点为左上角）
++left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++width	是	uint32	表示定位位置的长方形的宽度
++height	是	uint32	表示定位位置的长方形的高度
+words	否	string	识别结果字符串
+chars	否	array()	单字符结果，recognize_granularity=small时存在
++location	是	array()	位置数组（坐标0点为左上角）
+++left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++width	是	uint32	表示定位定位位置的长方形的宽度
+++height	是	uint32	表示位置的长方形的高度
++char	是	string	单字符识别结果

```
 // 示例
{
    direction : 2,
    log_id : 676709620,
    words_result : [ {
            location : {
                height : 20;
                left : 86;
                top : 387;
                width : 22;
            };
            words : "N";
        },
    ],
    words_result_num : 1;
}
```

身份证质量控制

身份证本地质量控制已包含在[AipCaptureCardVC ViewControllerWithCardType:CardTypeLocalIdCardFont ....]之中。

相关的使用方法已在AipOcrSdk工程中开源，用户可根据自己的需要自行修改。

- 初始化

  [[AipOcrService shardService]getTokenSuccessHandler:^(NSString *token) { //获取到身份证质量控制token// } failHandler:^(NSError* error) {

  }]; // 利用获取到的token 完成 IdcardQualityAdaptor的初始化 IdcardQualityAdaptor *idcard = [[IdcardQualityAdaptor alloc]init];

  [idcard initWithToken:token];

- 检测图片
  - (IdcardQualityModel *)process:(UIImage *)image width:(int)width height:(int)height channel:(int)channel cardType:(idcard_quality::IdCardType)type;

options参数

参数	类型	说明
image（已由image参数代替）	String	图像数据，支持本地图像文件路径，图像文件二进制数据，要求base64编码后大小不超过1M，最短边至少15px，最长边最大2048px,支持jpg/png/bmp格式 长宽比应与身份证规格(1.6：1)接近。当身份证(1)正方向完整位于输入图像内，占比80%-90% (2)正面姓名，号码，反面失效期 清晰，无反光 (3)无明显倾斜 时接口返回IDCARD_NORMAL, 其他情况返回细分错误码用于引导用户扫描到符合要求的图像。
Width，height，channel	Int	Uiimage的宽、高 、channel （RGB彩图设置为3）
type	idcard_quality::IdCardType	IDCARD_FRONT_SIDE为检测身份证头像面，IDCARD_BACK_SIDE为检测国徽面
IdcardQualityModel 中ImageStatusType属性	Int	IDCARD_NORMAL = 0 图像包含占比合适，清晰，无反光的身份证 IDCARD_WRONG_LOCATION = 1 图像不包含占比合适的身份证。可能的情况包括非身份证，过于倾斜等 IDCARD_BLURRED = 2 图像包含占比合适的身份证，但关键字段模糊 IDCARD_OVER_EXPOSURE = 3 图像包含占比合适的身份证，但关键字段反光 IDCARD_REVERSED_SIDE = 4 输入图像与输入参数设置的身份证国徽/人脸面不匹配 IDCARD_MOVING = 5 连续输入算法的两帧之间差异过大，可能是镜头或身份证在晃动 IDCARD_TOO_SMALL = 6 图像包含身份证，但占比过小

- 释放资源
  - (void)releaseIdcardQuality;

特殊配置

截图分辨率系数

该系数影响到拍摄照片之后截取的图片大小

在AipOcrSdk/AipOcrSdk/View/AipCutImageView.m 中

```
 //截图的分辨率系数 开发者可自行配置
static CGFloat const scale = 1.0;
```

图片放大/缩小系数

在AipOcrSdk/AipOcrSdk/View/AipCutImageView.m 中

```
 //捏合操作最大/最小系数
static CGFloat const pinchMaxscale = 10.0;
static CGFloat const pinchMinscale = 0.5;
```

FAQ

1. 报错 App identifier unmatch. 错误的BundleId或PackageName，如何解决？

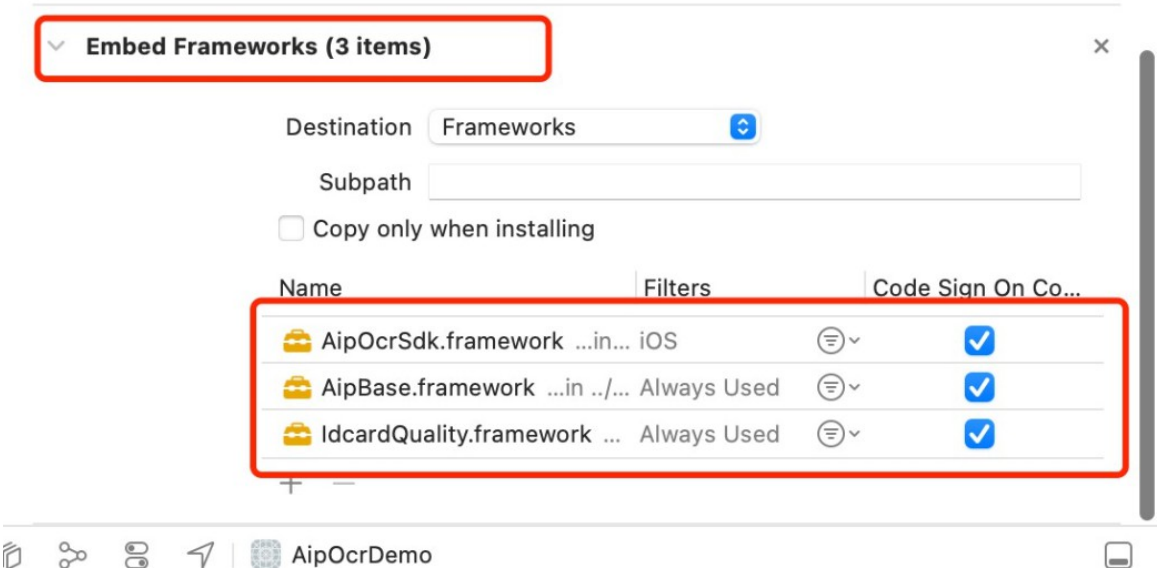
请在官网应用信息中绑定包名，具体步骤可参考本文档的图文教程。

2. 直接运行无问题，但Archive/IPA/Upload AppStore 时报错"Unsupported Architecture. Your executable contains unsupported architecture '[x86_64, i386]..."

为了方便开发者调试，AipBase.framework合并了模拟器和真机架构，上线前，使用lipo工具移除相关架构即可，参考：

```
 cd lib
# 使用lipo -info 可以查看包含的架构
lipo -info AipBase.framework/AipBase  # Architectures in the fat file: AipBase are: i386 x86_64 armv7 armv7s arm64
# 移除x86_64, i386
lipo -remove x86_64 AipBase.framework/AipBase -o AipBase.framework/AipBase
lipo -remove i386 AipBase.framework/AipBase -o AipBase.framework/AipBase
lipo -remove x86_64 AipOcrSdk.framework/AipOcrSdk -o AipOcrSdk.framework/AipOcrSdk
lipo -remove i386 AipOcrSdk.framework/AipOcrSdk -o AipOcrSdk.framework/AipOcrSdk
# 再次查看
lipo -info AipBase.framework/AipBase # Architectures in the fat file: AipBase are: armv7 armv7s arm64
```

3. Xcode12及以上版本上报错dyld: Library not loaded: XXXX，Reason: image not found 如何处理？ 此报错是因为framework是动态库，重新引入即可。可参考下图添加：



错误码表

IdcardQuality错误

错误信息	描述
IDCQ_NO_AUTHORITY	授权失败，Token无效
IDCQ_PATH_ERROR	模型地址错误，请保证idcardquality.framwork完整性
IDCQ_MODEL_INIT_ERROR	模型初始化错误，请保证idcardquality.framwork完整性
IDCQ_IMAGE_ERROR	image错误，请保证image参数合规
IDCQ_MEMORY_ERROR	MEMORY错误，请保证正确调用

验证错误

错误码	错误信息	说明	备注
110	Access token invalid or no longer valid	Access Token过期失效	请重新获得有效的Token
283501	License file check error	授权文件不匹配	请在控制台中配置正确的包名，并确认使用了正确的授权文件
283502	App identifier unmatch	BundleId不匹配	请在控制台中配置正确的包名，并确认使用了正确的授权文件
283504	Network error	网络请求失败	请授权App网络权限并保证网络通畅
283505	Server illegal response	服务器返回数据异常	
283601	Server authentication error	身份验证错误。	请在控制台中配置应用，并确认填写了正确的AK/SK，或使用了正确的授权文件
283602	Authentication time error	时间戳不正确，可能是设备时间异常。	
283604	App identifier unmatch	错误的PackageName或者BundleId	请在控制台中配置正确的包名，并确认使用了正确的授权文件
283700	Server internal error	服务器内部错误	您可以在工单系统中提交错误信息中的logId，我们将尝试帮您排查原因

服务错误

错误码	错误信息	描述
216015	module closed	模块关闭
216100	invalid param	非法参数
216101	not enough param	参数数量不够
216102	service not support	业务不支持
216103	param too long	参数太长
216110	appid not exist	APP ID不存在
216111	invalid userid	非法用户ID
216200	empty image	空的图片
216201	image format error	图片格式错误
216202	image size error	图片大小错误
216300	db error	DB错误
216400	backend error	后端系统错误
216401	internal error	内部错误
216500	unknown error	未知错误
216600	id number format error	身份证的ID格式错误
216601	id number and name not match	身份证的ID和名字不匹配
216630	recognize error	识别错误
216631	recognize bank card error	识别银行卡错误（通常为检测不到银行卡）
216632	ocr	unknown error
216633	recognize idcard error	识别身份证错误（通常为检测不到身份证）
216634	detect error	检测错误
216635	get mask error	获取mask图片错误
282000	logic internal error	业务逻辑层内部错误
282001	logic backend error	业务逻辑层后端服务错误
282002	input encoding error	请求参数编码错误
282100	image transcode error	图片压缩转码错误

SDK合规使用指南

前言

2023年2月工信部发布的**《工业和信息化部关于进一步提升移动互联网应用服务能力的通知》**对SDK、个人信息保护、服务体验等方面提出了具体要求，同时为了帮助开发者向最终用户提供相应功能及服务，特此起草本SDK合规使用指导。 您作为开发者为最终用户提供服务，需知悉并确认将遵守适用的法律法规和相关的标准规范，履行个人信息保护义务，并遵循合法、正当、必要和诚信的原则处理用户个人信息，包括但不限于《中华人民共和国个人信息保护法》、《中华人民共和国网络安全法》、《中华人民共和国数据安全法》以及其他适用的法律法规和相关的标准规范。 此文档用于帮助您更好地了解百度OCR采集SDK并合规使用本SDK服务，仅适用于开发者的业务区域为中国大陆地区的场景。

特别提示

本**《SDK合规指南》**是对本 SDK 的合规性和安全性描述与要求仅为我们向您提供的服务说明及使用建议，不构成也不应被视为我们对于任何法律法规及政策文件的及时的、完整的、全面的、甚至完全准确的分析，亦不构成我们对百度OCR采集SDK产品和/或服务的承诺和保证。本《指南》不能作为判断您与您所开发、运营的 App 是否满足合规与安全要求的可依赖的标准，亦不构成我们对前述事宜作出的任何担保或保证，您应当自行、独立地对前述 App 承担合规与安全责任。

目录

- SDK收集个人信息的频次、精度、使用目的、场景及对应选择的配置方式、示例
- SDK所需系统权限与功能关系，以及权限申请时机
- SDK扩展业务功能介绍及对应关闭的配置方式、示例
- SDK初始化及各项业务功能接口合规调用时机
- 联系我们

1. SDK收集个人信息不同频次、精度使用目的、场景及对应选择的配置方式、示例

为了帮助开发者向最终用户提供相应功能及服务，当最终用户使用相应功能及服务时，我们会通过开发者应用向系统申请最终用户设备的相应权限。开发者应确保最终用户可以随时通过取消系统授权开发者应用获取相应设备权限或其他开发者应用提供的授权设置，停止我们对最终用户个人信息的收集，之后最终用户可能将无法使用基于相应个人信息而提供的相关服务或功能，或者无法达到基于相应个人信息提供的相关服务拟达到的效果，但不会影响最终用户正常使用OCR采集SDK的其他不基于相应个人信息即可实现的业务功能。各项功能及服务涉及的个人信息包括：

序号	功能服务	个人信息类型	收集方式	适用系统版本
1.	为帮助开发者通过对身份证照片进行文字识别的方式，获取最终用户的姓名、身份证件号码作为待核验数据，以便实现无需最终用户手动输入即可向最终用户提供权威数据源身份核验功能	身份证照片	SDK直接采集	iOS及Android

注：更多个人信息收集、使用、处理的细节请参考OCR采集SDK隐私政策

2. SDK所需系统权限与功能关系，以及权限申请时机

为了保证最终用户能正常使用OCR采集SDK相应功能及服务，我们会通过开发者应用向系统申请最终用户设备的以下系统设备权限，申请前我们会征询最终用户的同意，最终用户可以选择“允许”或“禁止”权限申请。经过最终用户的授权后我们会开启相关权限，最终用户可以随时在系统中取消授权，最终用户取消授权会导致最终用户无法使用相关的业务功能，但不会导致最终用户无法使用其他业务功能。各项功能及功能对设备权限的调用情况如下：

Android系统版本 |设备权限 |功能及服务 |权限授权方式| | --- | --- | --- | |读取/写入外部存储卡| 用于存储OCR采集SDK的模型信息、配置文件、以及保存安全风险诊断信息；从相册中读取身份证照片，以便识别身份证号和姓名，以实现权威数据源身份核验功能| 授权方式由设备系统开发方以及开发者应用决定；当最终用户同意向开发者应用授予该权限时开启| |访问网络 |用于连接网络，进行数据传输，用作活体及身份核验| 授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启| |获取网络状态 |用于获取当前网络信息，进行安全风险诊断，并保护网络传输 |授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启| |打开相机/摄像头| 实现OCR图片采集、身份证照片采集，以便进行活体及身份校验| 授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启| |访问相册 |从相册中读取身份证照片，以便识别身份证号和姓名，以实现权威数据源身份核验功能| 授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启|

iOS系统版本 |设备权限 |功能及服务 |权限授权方式| | --- | --- | --- | |打开相机/摄像头| 实现OCR图片采集、身份证照片采集功能，以便进行活体及身份校验 |授权方式由设备系统开发方以及开发者应用决定；当最终用户同意向开发者应用授予该权限时开启| |访问相册 |从相册中读取身份证照片，以便识别身份证号和姓名，以实现权威数据源身份核验功能 |授权方式由设备系统开发方以及开发者产品决定；当最终用户同意向开发者产品授予该权限时开启| 注：在不同设备中，权限显示方式及关闭方式可能有所不同，具体请最终用户参考设备及系统开发方说明或指引。

3. SDK初始化及各项业务功能接口合规调用时机

为了避免您的应用在未获取用户的同意前SDK提前处理用户的个人信息，我们提供了SDK初始化的接口，请保证您的应用获取用户同意后才能调用此接口初始化SDK。

- **iOS：**[[AipOcrService shardService]getTokenSuccessHandler:^(NSString token) { //获取到身份证质量控制token// } failHandler:^(NSError error) {}]; // 利用获取到的token 完成 IdcardQualityAdaptor的初始化 IdcardQualityAdaptor *idcard = [[IdcardQualityAdaptor alloc]init]; [idcard initWithToken:token];
- **Android**：OCR.getInstance().initAccessToken(new OnResultListener() { @Override public void onResult(AccessToken result) { // 调用成功，返回AccessToken对象 String token = result.getAccessToken(); } @Override public void

onError(OCRError error) { // 调用失败，返回OCRError子类SDKError对象 } }, getApplicationContext());

4. 联系我们

OCR采集SDK的成长离不开各方开发者及最终用户的共同努力，我们非常感谢开发者及最终用户对OCR采集SDK数据更新、使用反馈方面的贡献。开发者及最终用户可以通过百度云工单反馈开发者及最终用户对OCR采集SDK产品和服务的建议以及在使用过程中遇到的问题，以帮助我们优化产品功能及服务，使更多用户更加便捷的使用我们的产品和服务。

离线SDK

概览

试用或购买授权

您申请的试用版授权或购买的正式授权都默认在授权总数池里，授权池里的授权可以自由分配，用于单台设备授权或批量设备授权两种授权方式（单台设备授权也称为按序列号授权）。

申请测试

申请试用版授权，您可以用于产品测试和效果验证，申请测试授权通过后，序列号有效期为默认为**自激活日期后的30天**，有效期内完全免费，您可以用于进行产品试用；批量设备授权应用有效期默认为**自申请日期后的30天**。

选择文字识别模型首次申请测试时，需要选择模型配置并填写项目信息，为了更全面地了解您的需求，优化产品效果，请您具体描述业务场景。提交信息后，我们将在**1个工作日**内进行审核，审核通过后，相应数量的授权将被分配到您的授权总数池中，**默认测试授权数量为2个**。

如您已经申请过相应模型的试用版授权，但数量不够时，可以选择「**申请更多**」，填写申请理由即可，审批通过后，默认下发对应模型2个试用版授权。

正式购买

正式购买的授权，有效期为**永久有效**，此「永久」是指绑定到具体的设备维度，但如已绑定的硬件设备变更后，授权则可能会失效。

产品价格

离线SDK的授权基于设备维度，每个序列号仅可以授权给一台设备。每个账号单次购买的序列号会累计计算，累计购买量越多，单价越便宜。购买授权并激活后，即可下载授权文件，永久有效。

通用文字识别、数字字母识别、身份证识别、车牌识别、vin码识别等SDK价格：

累计购买授权数量	每个授权单价
第3~1000个	299元/个
第1001~5000个	249元/个
第5001个及以上	199元/个

办公文档识别、快递单识别（windows）等SDK价格：

累计购买授权数量	每个授权单价
第3~1000个	799元/个
第1001~5000个	669元/个
第5001个及以上	499元/个

如有其他SDK需求，或需要大批量采购，可提交工单联系我们，或直接联系您的客户经理

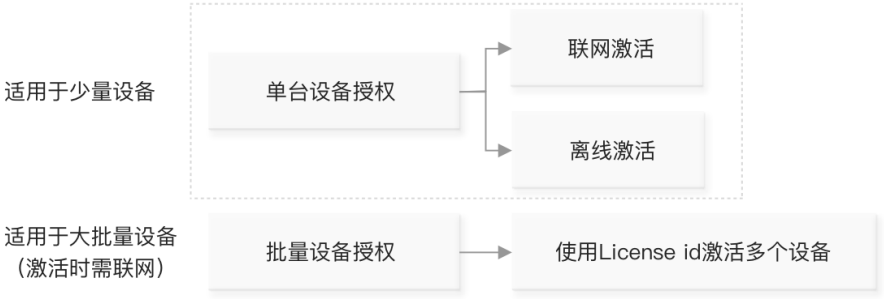
购买说明

1. 序列号按照设备维度授权（依据于硬件指纹信息），每个序列号仅可激活一台设备,激活后授权**永久有效**，但如果硬件信息发生变更（如拆卸网卡或后续增加网卡），或者硬件损坏，则授权不可恢复。
2. 序列号不限制开发平台，序列号仅绑定于指定的设备。无论是使用安卓还是Windows，序列号都可以激活。

3. 序列号的购买是「累计阶梯计费」，例如您购买2000个车牌SDK授权，应付费1000x299+1000x249。
4. SDK购买激活后，或购买超过7天，不支持退款。
   立即购买

授权方式

离线识别SDK授权方式为按设备授权，分为单台设备授权（适用少量设备）和批量设备授权（适用大量设备），每台硬件设备需要一个独立的授权，此授权的校验是基于设备的硬件指纹（指纹的获取在SDK初始化时会自动读取并展示），被授权的设备，将支持在有效期内运行SDK。视频教程请参见 OCR离线SDK使用教程（视频版）



以下需重新拉取授权的情况：设备授权不变，您只需要重新激活而已

- 删除SDK或基于SDK开发的应用
- 安卓系统升级

以下授权失败的情况：您需要重新购买序列号，之前的序列号失效

- 设备激活后，刷机、更换硬件导致指纹发生变化
- 硬件损坏

单台设备授权

序列号为管理授权的依据，每台被授权的设备都应对应一个序列号，用于标识对应的设备信息及授权记录。序列号的形式为16位随机英文数字组合，如：*CG56-GDGD-ZXVK-F7CR*。您在 管理后台 购买SDK授权后，选择添加序列号，系统将会发放您所选择数量的序列号。序列号不限制平台版本，相同模型任何版本的离线SDK，都可以使用此序列号激活，序列号不限制账号，可供任何设备激活使用。

点击「添加序列号」，选择模型，选择测试版或正式版授权，填写添加个数，点击确定后，后台将分批执行任务，预计**1~2分钟以内生效**，生效后，您即可在单台设备授权列表中查看已添加的序列号。

激活

已购买的序列号，是用于激活的唯一凭证，激活流程主要是将序列号与具体的硬件进行绑定（硬件指纹，即device_id），从而生成对应硬件设备的授权文件（License.zip），SDK运行前，将会校验授权文件是否和实际硬件信息相匹配。

联网激活

此种激活方式，适用于设备激活时可联网的情况，优势在于激活方便，使用序列号随时可以在设备端一键激活，满足业务灵活使用。

Windows版本联网激活

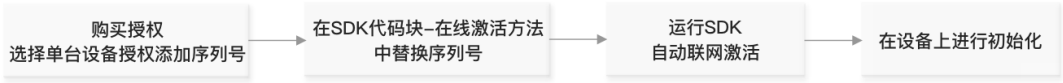


1. 获取序列号：从 管理后台 购买授权后，选择添加序列号即可获取
2. 在SDK配置界面中填写序列号：将SDK置于设备上，运行LicenseTool激活工具，在配置界面中填写序列号
3. 启动激活：在LicenseTool激活工具配置界面中，点击激活按钮

4. SDK自动联网，激活完毕：界面将提醒“激活成功”，并自动下载授权文件

> 视频教程请参见 OCR离线SDK使用教程（Windows版本）

安卓版本联网激活



1. 获取序列号：从 管理后台 购买授权后，选择添加序列号即可获取
2. 在SDK代码块中替换序列号：将序列号替换至以下函数中：

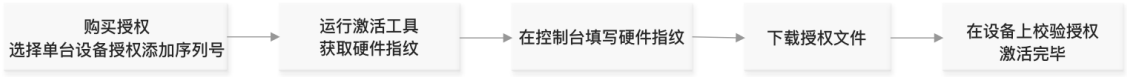
```
int ret = BDLicenseActivator.initLicenseOnLine(acticity, licenseID:"xxxx-xxxx-xxxx-xxxx", filename:"",
Predictor.getAlgorithmID());
```

3. 运行SDK，自动联网激活
4. 在设备上进行初始化：在设备上运行demo进行初始化

> 视频教程请参见 OCR 离线 Android SDK 使用教程（单台设备授权）

离线激活

此种激活方式，适用于设备完全不可联网的情况，优势在于可避免联网激活，满足业务对网络及安全性的严格要求，以及设备批量注册需求。您需要在后台配置好硬件指纹并完成和序列号的绑定，然后将授权文件放到SDK的指定位置。 **Windows版本离线激活**



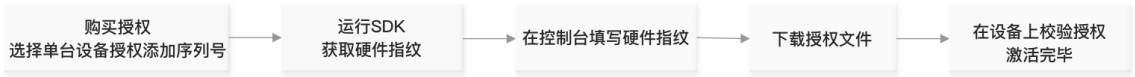
1. 获取序列号：从 管理后台 购买授权后，选择添加序列号即可获取
2. 采集硬件指纹：将SDK置于设备上，运行LicenseTool激活程序，获取硬件指纹
3. 配置授权：在 管理后台 单台设备授权列表对应的序列号下，选择离线激活，将硬件指纹填写到对应序列号上



4. 下载授权文件：绑定成功后下载授权文件，解压授权文件放置在SDK的License文件夹下（调试开发环境，请将授权文件，即License.zip文件中的license.ini 和license.key文件，放置在License文件夹中，即vcxproj文件同级目录下；实际运行环境，将授权文件放置在exe同级目录下即可）
5. 设备激活：运行程序，并初始化SDK完成授权

> 视频教程请参见 OCR离线SDK使用教程（Windows版本）

安卓版本离线激活



1. 获取序列号：从 管理后台 购买授权后，选择添加序列号即可获取
2. 采集硬件指纹：将SDK置于设备上，在开发工具中运行demo，在logcat中获取deviceid
3. 配置授权：在管理后台 单台设备授权列表对应的序列号下，选择离线激活，将硬件指纹填写到对应序列号上



4. 下载授权文件：绑定成功后下载授权文件，解压授权文件放置在SDK的assets目录下

5. 使用SDK中authFromFile函数进行离线激活

```
AndroidLicenser.ErrorCode ret =Predictor.getInstance().authFromFile(activity,licensekey（序列号）,licensename（license.ini）,true，algorithmId（鉴权id：例OcrgveEnginePlate.getAlgorithmId()）);
```

6. 设备激活：运行程序，并初始化SDK完成授权

批量设备授权

批量设备授权支持大量设备批量授权，License ID为管理授权的依据，多台设备可对应一个License ID，用于标识您应用的license授权信息，每个应用只能有唯一的标识，一经创建无法修改。批量激活需要设备支持联网。

点击批量设备授权Tab下的新建应用（当您申请过试用版或正式版授权后可见），填写以下信息：

- 应用名称：英文或数字，可根据业务情况自定义填写。
- License ID：用于标识授权文件的ID，每个应用的唯一标识，请自定义填写，创建应用后系统将在您填写的License ID后自动添加-offlinesdk-app后缀。**如系统提示License ID已存在，请修改后重新提交**。
- 包名：安卓工程的包名，是安卓应用的唯一标识，请填写与您真实的安卓包名一致。
- 签名的md5：安卓包签名的keystore文件中私钥的数据摘要，用于生成Android SDK生成对应License时的主要依据。
- 选择模型：选择模型后，可填写对应模型下您可分配的授权个数。一个应用可支持分配多个授权，即您的License ID可用于多个设备的授权。

填写完成后创建应用，创建成功您即可在批量设备授权列表中查看到您创建的应用。您可以用License ID放置到SDK代码中来完成批量激活。

License ID放置位置：请放置在initLicenseOnLine函数中，filename可不填

```
int ret = BDLicenseActivator.initLicenseOnLine(acticity, licenseID:"ocrplatenumberdemo-offlinesdk-app", filename:"", Predictor.getAlgorithmID());
```

目前通用文字识别SDK支持Android、Windows批量鉴权，车牌、身份证、数字字母和vin码SDK支持Android批量鉴权。

延长有效期

批量授权的方式，支持延长试用版授权的有效期，当您的授权池中有对应模型的授权，且存在该模型下的应用时，您可以选择延长应用有效期，默认延长**30天**。

具体为：当您的批量设备列表中存在应用时，您可以选择延长应用有效期，注意选择右上角对应的模型后，再申请延长有效期，比如选择车牌识别试用版有效期，我们将为您延长车牌识别应用的有效期。

> 视频教程请参见 OCR 离线 Android SDK 使用教程（批量设备授权）

错误码

鉴权相关错误码:

ErrorCode	常量值	说明
SUCCESS	0	成功
LICENSE_NOT_INIT_ERROR	1	license未初始化
LICENSE_DECRYPT_ERROR	2	license数据解密失败
LICENSE_INFO_FORMAT_ERROR	3	license数据格式错误
LICENSE_KEY_CHECK_ERROR	4	license-key(api-key)校验错误
LICENSE_ALGORITHM_CHECK_ERROR	5	算法ID校验错误
LICENSE_MD5_CHECK_ERRO	6	MD5校验错误
LICENSE_DEVICE_ID_CHECK_ERROR	7	设备ID校验错误
LICENSE_PACKAGE_NAME_CHECK_ERROR	8	包名（应用名）校验错误
LICENSE_EXPIRED_TIME_CHECK_ERROR	9	过期时间不正确
LICENSE_FUNCTION_CHECK_ERROR	10	功能未授权
LICENSE_TIME_EXPIRED	11	授权已过期
LICENSE_LOCAL_FILE_ERRO	12	本地文件读取失败
LICENSE_REMOTE_DATA_ERROR	13	远程数据拉取失败
LICENSE_LOCAL_TIME_ERROR	14	本地时间校验错误
OTHER_ERROR	0xff	其他错误

常见问题

安卓—常见问题

Q：同一台设备可以被多个序列号多次激活么？

A：可以。一台设备可以被多个序列号激活，没有限制。

Q：哪些情况可能导致指纹变化？

A：刷机、更换硬件设备将会导致指纹变化。但安卓系统升级、APP卸载重装、恢复系统出厂值并不会导致硬件指纹变化。

Q：安卓SDK百度界面可以修改吗？比如去除裁剪框以及修改识别界面的的从相册中选择图片等按钮，以及界面布局的背景

A：可以的，代码开源。

Q：离线SDK的授权时间是多久？

试用版授权时间默认是1个月，正式授权时间默认永久。

Q：离线激活如何获取硬件指纹（device_id）？

运行SDK示例工程，在logcat里搜索 BDLicenseLocalInfo 就能拿到device_id；如果使用Windows版本的通用SDK，运行LicenseTool.exe采集工具会自动读取硬件指纹。

Q：授权文件要存放在什么位置？

授权文件一般存放在assets资源文件夹下，便于管理。但是有需要存放在其他文件夹下，也是可以的。

Q：控制台显示已激活状态，但初始化还是报local auth faile 4

根据报错信息显示是本地授权失败，此报错一般是序列号填写有误、序列号已过期，或者使用方式有误，您先自行排查以下输入序列号时是否有空格等无关字符，以及设备的时间是否为当前时间，如均正确，可以在控制台提交工单，由技术支持同学为您进一步排查

Windows—常见问题

Q：离线sdk支持Windows的哪些版本?

支持Win7、Win10主流平台，不支持Win XP平台，不支持Windows Server，硬件需求是intel的cpu，需支持AVX指令集。

Q：车牌识别离线SDK与通用识别离线SDK一起集成出现冲突

相同的so库或jar包共用一个即可，不需要在集成中导入多次或重复引用。

Q：在线SDK与离线SDK共用是否会有冲突

不会有冲突，可共同使用。

智能文档分析平台

产品简介

产品简介

基于百度**文心大模型4.0**打造的一站式文档处理智能助手，涵盖合同审查、文档抽取、文档格式转换、文档比对、文档解析等多项功能，为用户提供高效、便捷的文档处理体验，助力企业轻松应对海量文档，有效规避潜在风险。

使用方式

支持在线工具和API服务两种使用方式。

如希望快速可视化体验效果，可登录智能文档分析平台，一键上传文档，在线测试；如希望使用API服务，可参见API文档。在线工具和API服务的额度共享互通。

核心功能

合同审查

文心大模型4.0驱动的合同审查，可精准定位条款原文，提供专业的风险判断、风险分析及修改示例，加速合同审查流程，可用于合规性审查场景：

- 自动预审风险，提效减漏，聚焦高价值任务
- 辅助合规性管理，质量效率双向提升

文档抽取

文心大模型4.0驱动的文档抽取，支持自定义字段，无需训练即可实现短语、长段落、表格等文本信息抽取，适用于票据、表单、合同等多类文档，可用于文档智能化场景：

- 确保从复杂文档中提取关键信息的准确性和完整性
- 表格内容精确提取，便捷挖掘数据价值

文档格式转换

识别图片/PDF/OFD 文档版面布局，提取文字内容，并转换为保留原文档版式的 Word、Excel 、OFD文档，方便二次编辑和复制，可用于文档电子化场景：

- 高效录入文档内容，实现纸质档案电子化
- 快速处理多种文档，提示信息处理效率

文档比对

精准比对文档的增删改差异，快速定位并高亮显示差异原文，支持导出完整的比对报告，大幅提升比对准确性和效率，可用于文档防篡改场景：

- 确保签署重要文档的准确性和完整性
- 快速定位文档版本差异，提升审核效率

文档解析

支持doc、xlsx和pdf等16种格式文档进行解析，输出文档的版面、表格、阅读顺序、标题层级、文本切分块等信息，返回Markdown格式文本内容，识别准确率可达 90% 以上，可用于文档结构化场景：

- 解析切分文档，丰富RAG上下文，提高大模型问答系统的效率与精准度
- 将非结构化文本转换为结构化信息，以提升文档分类、归档和索引效率

产品优势

文心大模型4.0驱动

- 基于大模型对海量文本的深度学习，精准理解文本上下文信息，高效完成审查、抽取等文档分析任务

文档格式全面

- 支持10+类主流文档格式，轻松实现多类文档文本内容解析，准确识别各式版面布局

使用方式灵活

- 支持 SaaS 在线使用、公有云 API 调用以及私有化本地部署，满足企业各场景使用需求

服务安全可靠

- 数据安全已通过国家信息安全等级保护三级、ISO27001等多项国内外权威信息安全管理认证，提供国际化安全保障能力

客户案例

智联招聘

- 融合“法律大模型+法务知识模型”的前置审查，大幅提升风险透明度，审核准确率达85-90%，驱动合同审查迈入新纪元

石化盈科

- 打造行业内领先的能源领域合同智能化应用平台，完善合同审查体系，审批效率提升 68%

深圳证券

- 构建面向金融文本的标注和训练系统，实现对文件信息的精准抽取，协助业务人员提升对文本和图片分析与处理效率，赋能上层业务

快速入门

智能文档分析平台是一款基于百度文心大模型4.0赋能的智能文档助手，集合同审查、文档抽取、文档格式转换、文档比对等功能于一体，致力于打造高效便捷的文档处理方案。它助力企业轻松驾驭海量文档，不仅提升工作效率，更精准规避潜在风险，是企业文档管理的得力助手。本文档将为新用户提供轻松入门的流程以及相关注意事项的介绍，帮助您快速上手智能文档分析平台。

一、注册与认证

（一）账号注册

请参考文档并注册百度智能云账号；若您已经注册百度智能云账号，可忽略此步骤。



（二）实名认证

请参考文档进行实名认证。完成个人实名或企业实名认证后，方可获得免费额度赠送；若您已经在百度智能云完成实名认证，可忽略此步骤。



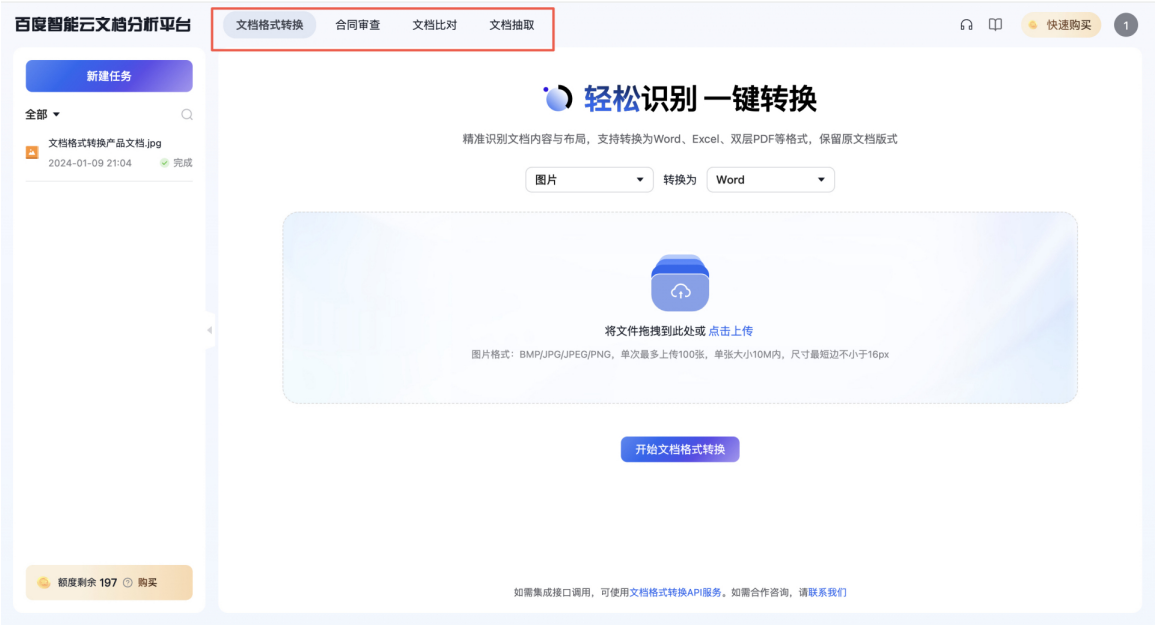
二、获取免费资源

注意：

- 每个智能文档分析功能均可领取200页免费测试资源
- **在线工具和API服务的额度为共享互通**，在一个平台上领取免费额度后，无法在另外一个平台上重复领取；
- 免费测试资源使用完毕后，可开通付费或申请免费调额。

（一） 在线工具领取

1. 登录智能文档分析平台，根据您的需求**选择相应的功能**。



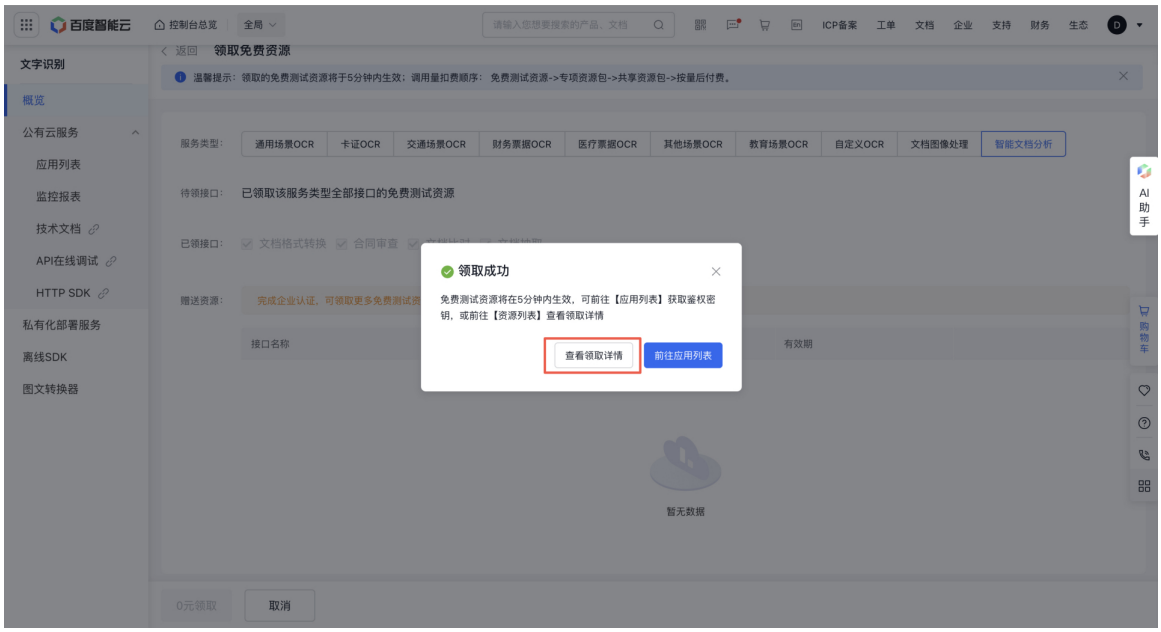
2. 新用户首次进入功能页面后，即自动领取**200页免费配额**。每个功能的额度是相互独立的。



（二）API服务领取

1. 登录 文字识别控制台。如果您已经完成了实名认证，系统将自动发放免费测试资源到您的账户上。

2. 领取成功的免费测试资源将会显示在资源列表的「已领取资源」中。您可以选择「查看领取记录」去往「资源列表」查看。**刚领取的资源大约5分钟生效，若领取接口长时间未在「资源列表」上生效显示，可提交工单咨询。**



三、功能操作指南

接下来将引导您快速熟悉在线工具的各项功能，带您快速入门。API版的操作指南请查看API文档

（一）文档格式转换

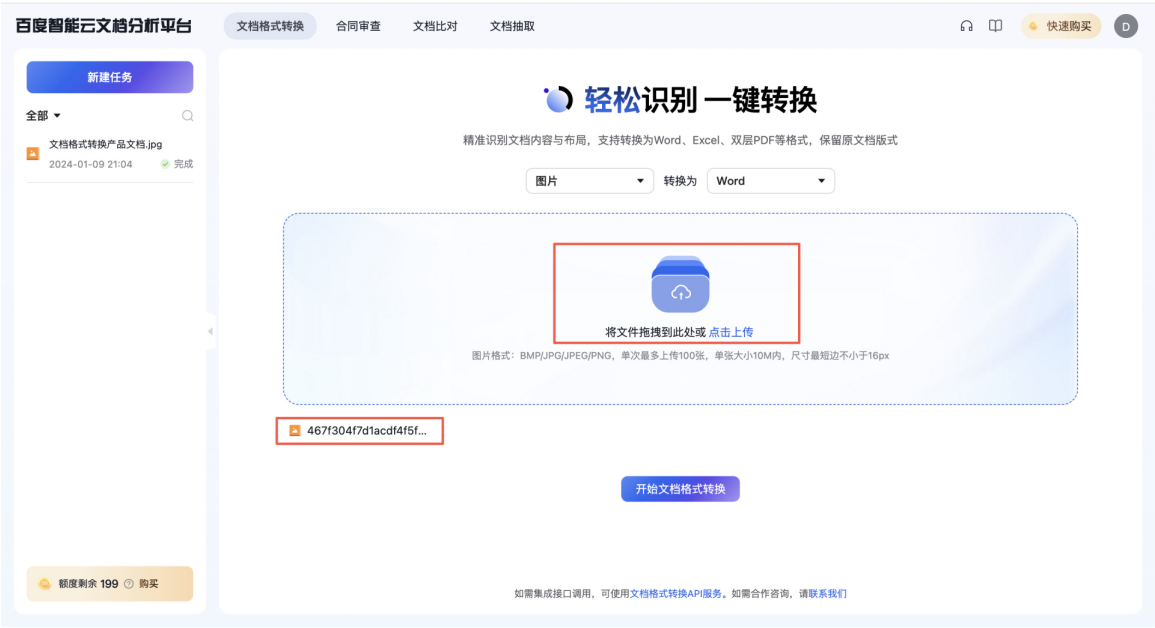
1. 进入文档格式转换页面后，系统会展示一份内置的「图片转Word」示例，供您参考。



2. 点击「审查清单」选项框，选择您所需的转换类型。



3.拖拽文件至上传区域或点击上传，上传文件后，点击「开启文档格式转换」按钮提交任务。



4. 任务处理完后，页面左侧展示源文件，页面右侧展示识别结果。鼠标悬停在识别结果上时，左侧对应文本会有高亮显示，点击「复制文本」按钮即可复制转换后的结果，点击「下载结果」按钮即可下载转换后的格式文件。

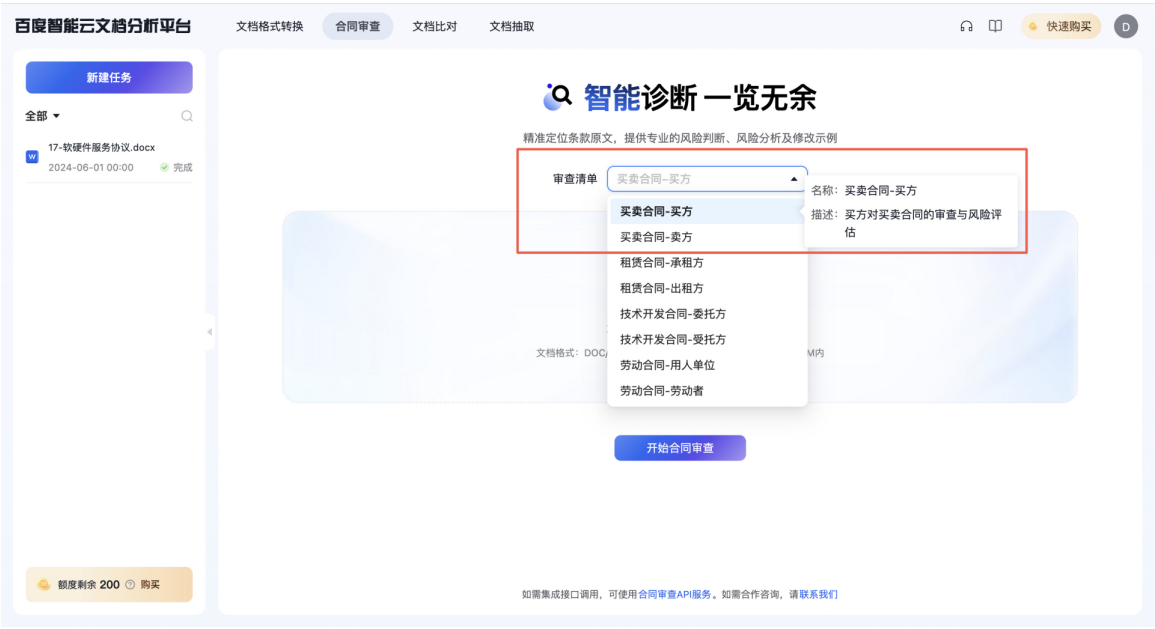


（二）合同审查

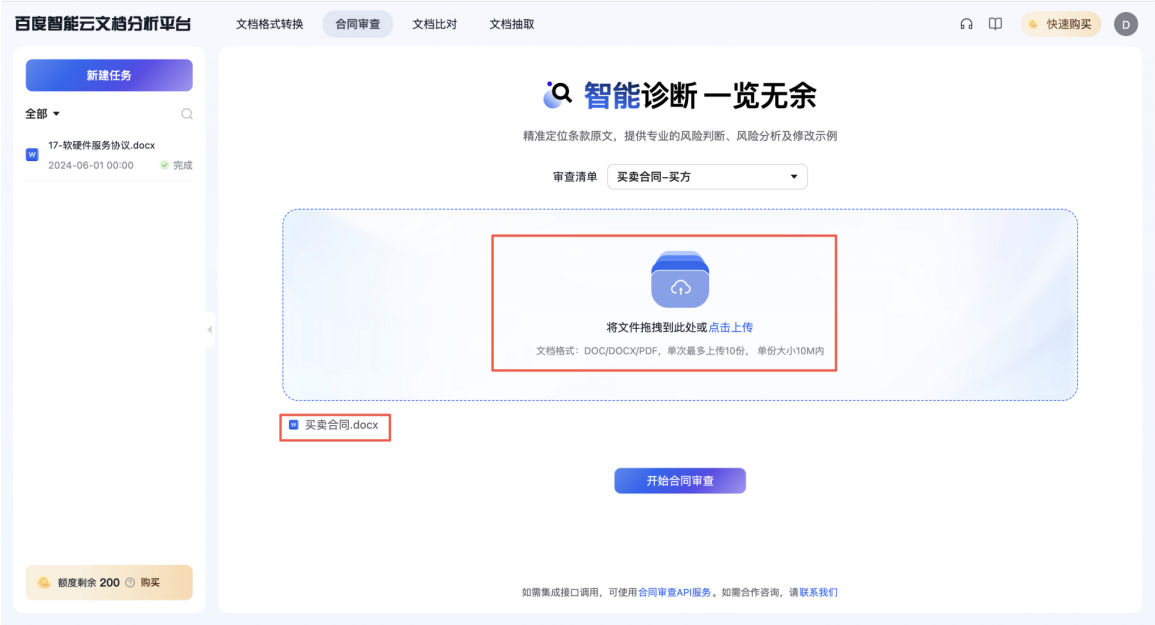
1. 进入合同审查页面后，系统会展示一份内置的「买卖合同-卖方」的审查示例，供您参考。



2. 点击「审查清单」选项框，选择您所需的合同审查类型。



4. 拖拽文件至上传区域或点击上传，单次支持上传最多10份文件；上传文件后，点击「开始合同审查」按钮提交任务。



5. 任务处理完后，页面左侧展示源文件，页面右侧展示各个条款的审查结果。您可以查看每个条款的审查结果以及原文中相应的位置。



6.您可以点击风险等级和隐藏功能按钮，对任务文件进行条款筛选管理。



7.点击「下载结果」按钮即可将审查结果导出成带批注的Word文件。



（三）文档比对

1. 进入文档比对页面后，系统会展示一份内置的文档比对示例，供您参考。



2. 选择您希望保留的特殊差异识别，拖拽上传主版和副版文件至上传区域或点击上传。上传文件后，点击「开始文档比对」按钮提交任务。



3. 任务处理完后，页面左侧展示源文件，页面右侧展示主版和副版之间的差异项。总共有「新增」、「删除」、「修改」三个类别：

a. 「新增」是指相比主版，副版新增的内容

b. 「删除」是指相比主版，副版删除的内容

c. 「修改」是指相比主版，副版修改的内容

- 您可以点击页面右侧卡片或中心的图标来定位原文中差异的位置。



- 您可以点击「新增」「删除」「修改」按钮，对任务文件进行差异筛选管理。



4. 点击「下载结果」按钮即可下载比对差异结果的报告。



汽车买卖合同.doc（汽车买卖合同2.doc）比对报告

一、汇总信息

副版相较于主版有 6 处差异，其中包含：

- 0 项文字新增
- 4 项文字删除
- 2 项文字修改

二、比对差异列表

比对差异列表

类型说明：
- 新增：相比主版，副版新增的内容
- 修改：相比主版，副版修改的内容
- 删除：相比主版，副版删除的内容

序号	差异类型	比对内容类型	主版	副版	主版位置	副版位置
1	删除	文本内容	出卖人:张三	出卖人:	第1页	第1页
2	删除	文本内容	买受人:李四	买受人:	第1页	第1页

（四）文档抽取

1. 进入文档抽取页面后，系统会展示一份内置的文档抽取示例，供您参考。



2. 拖拽文件至上传区域或点击上传；上传文件后，点击「开始文档抽取」按钮提交文件。



3. 完成文件解析后，配置需要抽取的字段，配置完成后点击「发起抽取」按钮提交抽取任务。

其中，单个字段是指独立的值，例如姓名；组合字段是指一组相关联字段的值，例如商品的详细信息，包含商品编码、价格、重量等。



4. 任务处理完后，页面左侧展示源文件，页面右侧【查看抽取结果】选项下展示最终抽取结果。点击「下载结果」按钮可以下载json格式文件。



5. 您可以点击页面右侧抽取结果中的字段来定位原文中差异的位置。



API参考

功能发布记录

发布时间	更新日志
2025-06-24	**文档解析新增mhtml格式解析**：可对 mhtml 格式的流式文档进行解析，输出文档的版面等信息 **文档解析新增多语种识别：**支持识别中、英、日、韩、法等 20+ 语言类型
2024-10-17	**文档抽取正式商用：**可开通按量后付费或购买预付费资源包，按页计费，详情参见价格文档 **文档抽取新增字段描述和字段值去重**：可对抽取字段进行描述，提升模型抽取效果；可对单个字段重复抽取的值进行去重。可访问智能文档分析平台-文档抽取即可体验 **文档解析新增公式识别：**可识别并返回文档内的公式
2024-08-15	**文档解析正式商用：**可开通按量后付费或购买预付费资源包，按页计费，详情参见价格文档
2024-08-08	**文档抽取开放公测：**文心大模型驱动的文档抽取，支持自定义字段，无需训练即可实现短语、长段落、表格等文本信息抽取，支持在线工具和API服务两种使用方式。如需使用在线工具，请访问智能文档分析平台-文档抽取；如需使用API服务，请参见API文档
2024-08-06	**文档解析新增Markdown格式返回结果：**可返回Markdown格式的解析结果，详情参见API文档
2024-07-25	**文档解析新增文档内容切分能力：**支持将长文档内容按照标题层级、标点符号或者字符数的维度切分成更易于处理的文本块，详情参见API文档
2024-07-08	**合同审查、文档比对新增前端SDK渲染：**可辅助用户在网页中便捷地调用合同审查和文档比对服务，实现与当前智能文档分析平台一致的前端渲染和交互界面，详情参见API文档
2024-06-24	**文档解析开放公测：**可支持对doc、pdf、图片、xlsx等16种格式文档进行解析，输出文档的版面、表格、阅读顺序、标题层级、旋转角度等信息，详情参见API文档
2024-03-27	**合同审查、文档比对正式商用：**可开通按量后付费或购买预付费资源包，按页计费，详情参见价格文档
2024-03-12	**合同审查新增批注下载功能：**可支持将审查结果以批注形式导出至Word文件，详情参见API文档
2023-12-19	**合同审查开放公测**：文心大模型驱动的合同审查，可精准定位条款原文，提供专业的风险判断、风险分析及修改示例，支持在线工具和API服务两种使用方式。如需使用在线工具，请访问智能文档分析平台-合同审查；如需使用API服务，请参见API文档 **文档比对开放公测：** 可精准比对文档增删改查，高亮定位差异信息，支持导出完整比对报告，支持在线工具和API服务两种使用方式。如需使用在线工具，请访问智能文档分析平台-文档比对；如需使用API服务，请参见API文档
2023-08-16	**文档格式转换正式商用**：可开通按量后付费或购买预付费资源包，按页计费，详情参见价格文档
2023-05-26	**文档格式转换开放公测**：可识别图片/PDF文档版面布局，提取文字内容，并转换为保留原文档版式的Word、Excel文档，方便二次编辑和复制，支持在线工具和API服务两种使用方式。如需使用在线工具，请访问智能文档分析平台-文档格式转换；如需使用API服务，请参见API文档

API文档

文档格式转换

接口描述

原「图文转换器」现已更名为「文档格式转换」。

可识别图片/PDF文档版面布局，提取文字内容，并转换为保留原文档版式的Word、Excel文档，方便二次编辑和复制，可支持含表格、印章、水印、手写等内容的文档。满足文档格式转换、企业档案电子化等信息管理需求。如希望快速可视化体验效果，可登录智能文档分析平台，一键上传文档，在线测试；**在线工具和API服务的额度共享互通**。

在线工具已支持「PDF -> 双层PDF/双层OFD」、「OFD -> 双层OFD/Word/Excel」 转换类型，API服务尚不支持上述转换类型，正在开发中，敬请期待。

文档格式转换API服务为**异步接口**，需要先调用**提交请求接口**获取 task_id，然后调用**获取结果接口**进行结果轮询，建议提交

请求后 5～10 秒轮询。**提交请求接口**QPS为2，**获取结果接口**QPS为10。

提交请求接口

在线调试 您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ocr/v1/doc_convert/request`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

注意：要求使用 JSON 格式的结构体来描述一个请求的具体内容。

参数	是否必选	类型	说明
image	和 url/pdf_file 三选一	string	图像的base64编码，需去掉编码头（） 支持的文件类型：jpg/jpeg/png/bmp 支持的文件大小：base64编码后大小不超过4M，最短边至少15px，最长边最大4096px **优先级：** image > url > pdf_file，当image字段存在时，url、pdf_file字段失效
url	和 image/pdf_file 三选一	string	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码 支持的文件类型：jpg/jpeg/png/bmp 支持的文件大小：base64编码后大小不超过4M，最短边至少15px，最长边最大4096px **优先级：** image > url > pdf_file，当image字段存在时url字段失效 **请注意关闭URL防盗链**
pdf_file	和 image/url 三选一	string	pdf文件的base64编码，base64编码后大小不超过10M **优先级：** image > url > pdf_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容；**若不传入，默认识别文件所有页，页码从1开始**

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、文档地址或Base64信息。

提示二：目前仅提供Python语言，如需其他语言示例可参考 示例代码中心 。

```
Python
```

```
 import base64
import requests

'''
文档格式转换-提交请求
'''


request_host = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_convert/request"
###### 二进制方式打开图片文件
f = open('[本地文件]', 'rb')
img = base64.b64encode(f.read())

params = {"image": img}
access_token = '[调用鉴权接口获取的token]'
request_url = request_host + "?access_token=" + access_token
headers = {'Content-Type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, headers=headers, data=params)
```

返回说明

返回参数

字段	类型	说明
success	bool	当前请求状态； true 表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该请求生成的task_id，后续使用该task_id获取识别结果
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": 12345,
  "result":{
     "task_id":"task-xxxxxxx",
  },
  "code":1001,
  "message": "Create task successfully!"
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
  "success":false,
  "log_id": 12345,
  "error_code": 216401,
  "error_msg": "Create task failed!"
}
```

获取结果接口

在线调试 您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：POST

请求URL： https://aip.baidubce.com/rest/2.0/ocr/v1/doc_convert/get_request_result

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
task_id	是	string	发送提交请求时返回的task_id

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、task_id。

提示二：目前仅提供Python语言，如需其他语言示例可参考 **示例代码中心** 。

Python

```python
 import requests

'''
文档格式转换-获取结果
'''


request_host = "https://aip.baidubce.com/rest/2.0/ocr/v1/doc_convert/get_request_result"
params = {"task_id": "[调用提交请求接口获取的task_id]"}

access_token = '[调用鉴权接口获取的token]'
request_url = request_host + "?access_token=" + access_token
headers = {'Content-Type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, headers=headers, data=params)
if response:
 print(response.json())
```

返回说明

返回参数

字段	类型	说明
success	bool	当前请求状态； true表示请求成功，false表示请求异常
log_id	uint64	唯一的log id，用于问题定位
result	dict	返回的结果列表
+ task_id	string	该文件对应请求的task_id
+ ret_code	int	识别状态，1：任务未开始；2：进行中；3：已完成
+ ret_msg	string	识别状态信息：任务未开始；进行中；已完成
+ percent	int	文档转换进度（百分比）
+ result_data	dict	识别结果字符串，返回word、excel的文件分别的下载地址
+ +word	string	还原后的word文件的下载地址，**下载地址有效期为30天，文件识别失败时返回""**
+ +excel	string	还原后的Excel文件的下载地址，**下载地址有效期为30天，若文档中没有表格则返回""**
+ create_time	datetime	任务创建时间
+ start_time	datetime	任务开始时间
+ end_time	datetime	任务结束时间
code	int	成功状态码
message	string	详情

返回示例

成功返回示例：

```
{
  "success":true,
  "log_id": "xxxxxx",
  "result":{
     "task_id":"task-xxxxxxx",
     "ret_code": 3,
     "ret_msg": "已完成",
     "percent": 100,
     "result_data": {
        "word": "word_download_url",
        "excel": "",
     },
     "create_time": "2023-01-17 11:06:12",
     "start_time": "2023-01-17 11:06:13",
     "end_time": "2023-01-17 11:06:15"
  },
  "code":1001,
  "message": "Query task successfully!""
}
```

若查询的task_id不存在，返回result为{}。 请求失败响应体示例如下：

```
{
  "code":1001,
  "log_id":1635891796603052032,
  "message":"Query task successfully!",
  "result":{},
  "success":true
}
```


合同审查

接口描述

合同审查支持多种合同场景审查，精准定位关键审查点，提供专业的风险评估、判断依据及修改建议，加速合同审查流程。如希望快速可视化体验效果，可登录智能文档分析平台，一键上传文档，在线测试；**在线工具和API服务的额度共享互通**。

合同审查API服务为**异步接口**，需要先调用**提交请求接口**获取 taskID，然后调用**获取结果接口**进行结果轮询，建议提交请求后1～2分钟开始轮询。**提交请求接口**QPS为2，**获取结果接口**QPS为10。

提交请求接口

在线调试 您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/file/2.0/brain/online/v1/textreview/task`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	multipart/form-data

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
file	file/fileURLList 二选一	file	-	文件数据 支持的文件类型：doc/docx/pdf 支持的文件大小：仅支持上传一篇文件，文件大小不超过10M **优先级：** file > fileURLList，当file字段存在时，fileURLList字段失效
fileURLList	file/fileURLList 二选一	string	-	文件完整URL，仅支持北京区域的BOS公网访问，URL长度不超过1024字节，支持PDF/doc/docx格式，仅支持上传1篇文件，文件大小不超过10M **优先级：** file > fileURLList，当file字段存在时，fileURLList字段失效 **请注意关闭URL防盗链**
			Sales_Party A_V2/ Sales_Party B_V2/	

templateName	templateName/templateId 二选一	string	Lease_PartyA_V2/ Lease_PartyB_V2/ TechDev_PartyA_V2/ TechDev_PartyB_V2/ Labor_PartyA_V2/ Labor_PartyB_V2/ Entrustment_PartyA_V2/ Entrustment_PartyB_V2/ Work-for-hire_PartyA_V2/ Work-for-hire_PartyB_V2/ LaborDispatch_PartyA_V2/ LaborDispatch_PartyB_V2/ RealtySvcs_PartyA_V2/ RealtySvcs_PartyB_V2/ EquipPur_PartyA_V2/ EquipPur_PartyB_V2/ FinLease_PartyA_V2/ FinLease_PartyB_V2/ DebtAssign_PartyA_V2/ DebtAssign_PartyB_V2/ CISG_PartyA_V2/ CISG_PartyB_V2 / GUAR_PartyA_V2 /	该清单类型为系统预置清单。每次审查任务仅可选择单一清单。具体清单选项如下： - **Sales_PartyA_V2：**买卖合同-买方； - **Sales_PartyB_V2：**买卖合同-卖方； - **Lease_PartyA_V2：**租赁合同-出租方； - **Lease_PartyB_V2：**租赁合同-承租方； - **TechDev_PartyA_V2：**技术开发合同-委托方； - **TechDev_PartyB_V2：**技术开发合同-受托方； - **Labor_PartyA_V2：**劳动合同-用人单位； - **Labor_PartyB_V2：**劳动合同-劳动者； - **Entrustment_PartyA_V2：**委托合同-委托方； - **Entrustment_PartyB_V2：**委托合同-受托方； - **Work-for-hire_PartyA_V2：**承揽合同-定作人； - **Work-for-hire_PartyB_V2：**承揽合同-承揽人； - **LaborDispatch_PartyA_V2：**劳务派遣合同-用工单位； - **LaborDispatch_PartyB_V2：**劳务派遣合同-劳务派遣单位； - **RealtySvcs_PartyA_V2：**物业服务合同-业主； - **RealtySvcs_PartyB_V2：**物业服务合同-物业服务人； - **EquipPur_PartyA_V2：**设备采购合同-买方； - **EquipPur_PartyB_V2：**设备采购合同-卖方； - **FinLease_PartyA_V2：**融资租赁合同-出租方； - **FinLease_PartyB_V2：**融资租赁合同-承租方； - **DebtAssign_PartyA_V2：**债权转让合同-转让方； - **DebtAssign_PartyB_V2：**债权转让合同-受让方； - **CISG_PartyA_V2：**国际货物贸易合同-买方； - **CISG_PartyB_V2：**国际货物贸易合同-卖方 ； - **GUAR_PartyA_V2：**保证合同-保证方 ； - **GUAR_PartyA_V2：**保证合同-债权方 ； - **CG_PartyA_V2：**货运合同-承运方 ； - **CG_PartyB_V2：**货运合同-托运方 ；

			GUAR_PartyB_V2 / CG_PartyA_V2 / CG_PartyB_V2 / Factoring_PartyA_V2 / Factoring_PartyB_V2 / Brokerage_PartyA_V2 / Brokerage_PartyB_V2 / TradingTrust_PartyA_V2 / TradingTrust_PartyB_V2 / PNRship_V2 / PT/PAT_PartyA_V2 / PT/PAT_PartyB_V2/ TST_PartyA_V2/ TST_PartyB_V2 / TechLic_PartyA_V2 / TechLic_PartyB_V2/ EquipLea_PartyA_V2/ EquipLea_PartyB_V2/ ConstCtrl_PartyA_V2/ ConstCtrl_PartyB_V2	- **Factoring_PartyA_V2**：保理合同-保理商 ； - **Factoring_PartyB_V2**：保理合同-卖方 ； - **Brokerage_PartyA_V2**：中介合同-委托人 ； - **Brokerage_PartyB_V2**：中介合同-中介人 ； - **TradingTrust_PartyA_V2**：行纪合同-委托人 ； - **TradingTrust_PartyB_V2**：行纪合同-行纪人 ； - **PNRship_V2**：合伙合同-合伙人 ； - **PT/PAT_PartyA_V2**：专利（申请）权转让合同-受让方 ； - **PT/PAT_PartyB_V2**：专利（申请）权转让合同-转让方； - **TST_PartyA_V2**：技术秘密转让合同-受让方； - **TST_PartyB_V2**：技术秘密转让合同-转让方； - **TST_PartyB_V2**：技术秘密转让合同-转让方； -**TechLic_PartyA_V2**：技术许可合同-许可方； -**TechLic_PartyB_V2**：技术许可合同-被许可方； - **EquipLea_PartyA_V2**：设备租赁合同-出租方； - **EquipLea_PartyB_V2**：设备租赁合同-承租方； - **ConstCtrl_PartyA_V2**：建设工程施工合同-发包方； - **ConstCtrl_PartyB_V2**：建设工程施工合同-承包方
templateId	templateName/templateId 二选一	string	-	该清单类型为用户在**合同审查清单管理** 创建的我的清单。每次审查任务仅可选择单一清单。
commentRiskLevel	否	string	normal/major/all	筛选下载的批注结果中展示的风险等级条款。默认为"major"，即仅展示重大风险条款。 - **normal**：一般风险； - **major**：重大风险； - **all**：一般风险和重大风险

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、文档地址或Base64信息。

提示二：目前仅提供Python语言，如需其他语言示例可参考 **示例代码中心** 。

Python

```python
import requests
import os

'''
合同审查-提交请求
'''


request_host = "https://aip.baidubce.com/file/2.0/brain/online/v1/textreview/task"
###### 二进制方式打开文档文件
f = '[本地文件]'
file = {"file": (os.path.basename(f), open(f, 'rb'), "multipart/form-data")}

###### 将 templateName 放在表单数据中
data = {"templateName": "Sales_PartyA"}
access_token = '[调用鉴权接口获取的token]'
request_url = f"{request_host}?access_token={access_token}"
response = requests.post(request_url, data=data, files=file)
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ taskId	string	该请求生成的taskId，后续使用该taskId获取审查结果

返回示例

成功返回示例：

```
{
  "error_code": 0,
  "error_msg": "",
  "log_id": "259575694341050368",
  "result": {
     "taskId": "textreview-task-xnejhkwvcz5qpr3c"
  }
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
  "error_code": 282003,
  "error_msg": "missing parameters",
  "log_id": "259909120864665600"
}
```

获取结果接口 在线调试 您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示

例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/file/2.0/brain/online/v1/textreview/task/query`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	multipart/form-data

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
taskId	是	string	发送提交请求时返回的taskId

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、taskId。

提示二：目前仅提供Python语言，如需其他语言示例可参考 **示例代码中心** 。

Python

```python
import requests

'''
合同审查-获取结果
'''


request_host = "https://aip.baidubce.com/file/2.0/brain/online/v1/textreview/task/query"
params = {"taskId": "[调用提交请求接口获取的taskId]"}

access_token = '[调用鉴权接口获取的token]'
request_url = f"{request_host}?access_token={access_token}"
response = requests.post(request_url, params=params, files=params)
if response:
    print(response.json())
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ taskId	string	任务ID

+ status	string	任务状态，pending：排队中；processing：运行中；success：成功；failed：失败
+ reason	string	任务失败描述信息
+ createdAt	string	任务创建时间
+ startedAt	string	任务开始时间
+ finishedAt	string	任务结束时间
+ duration	string	任务执行时长
+ textreviewResult	dict	合同审查结果列表
++ docId	string	文档ID
++ docName	string	文档名称
++ status	string	文件状态，success：成功；failed：失败
++ reason	string	文件失败描述信息
++ fileURL	string	解析后的含位置信息的PDF文件地址，**地址有效期30天**
++ reportURL	string	包含重大风险条款批注的Word文档地址，**地址有效期30天**。若上传的源文件为PDF格式，则无法支持批注下载
++ reportStatus	string	包含批注的Word文档状态，success：成功；failed：失败
++ chatContents	dict	条款审查结果列表
+++ ruleId	string	条款ID
+++ ruleName	string	条款名称
+++ riskName	string	条款的风险等级
+++ predictResult	string	条款的原始审查结果
+++ markdownResult	string	条款的markdown格式审查结果
+++ riskReviewResults	dict	条款风险点审查结果列表
++++ riskPoint	string	风险点名称
++++ riskAnalysis	string	风险分析
++++ modifyExample	string	修改示例
++++ originalContract	string	合同原文
++++ positions	dict	重大风险条款的风险点原文位置信息列表
+++++ pageNum	int	条款的原文位置页码
+++++ box	[4]int	条款的原文四角点坐标，每个框选范围包含左上角x坐标、左上角y坐标、矩形框宽度、矩形框高度，共四个数值，如[89,74,61,12]
+++ tags	dict	条款的原文和位置信息列表
++++ value	string	条款的原文
++++ positions	dict	条款的原文位置信息列表

+++++ pageNum	int	条款的原文位置页码
+++++ box	[4]int	条款的原文四角点坐标，每个框选范围包含左上角x坐标、左上角y坐标、矩形框宽度、矩形框高度，共四个数值，如[89,74,61,12]

返回示例

成功返回示例：


```
{
{
  "error_code": 0,
  "error_msg": "",
  "log_id": "",
  "result": {
    "taskId": "textreview-task-bey18jc6j14bz92e",
    "status": "success",
    "createdAt": "2024-12-06 09:13:19 +0000 UTC",
    "startedAt": "2024-12-06 09:13:19 +0000 UTC",
    "finishedAt": "2024-12-06 09:14:59 +0000 UTC",
    "duration": "1分40秒",
    "reason": "",
    "textreviewResult": [
      {
        "docId": "doc-1n9d180uds3enztm",
        "docName": "011.docx",
        "status": "success",
        "reason": "",
        "chatContents": [
          {
            "ruleId": "rule-b7416vvakjtps4u7",
            "ruleName": "主体信息条款",
            "tags": [
              {
                "value": "甲方（转让方）\n名称：\n统一社会信用代码：\n乙方（受让方）\n名称：\n统一社会信用代码：\n本合同各方经平等自愿协商，根据《中华人民共和国民法典》及相关法律法规，就甲方向乙方转让标的账号事宜，签订本合同以共同遵守。",
                "positions": [
                  {
                    "pageNum": 1,
                    "box": [
                      90,
                      105,
                      396,
                      176
                    ]
                  }
                ]
              },
              {
                "value": "甲方（盖章）：\n法定代表人或授权代表：\n乙方（盖章）：\n法定代表人或授权代表：",
                "positions": [
                  {
                    "pageNum": 6,
                    "box": [
                      90,
                      190,
                      132,
                      106
                    ]
                  }
```

```
                ]
              }
            ],
            "riskReviewResults": [
              {
                "originalContract": "甲方（转让方）\n名称：\n统一社会信用代码：\n乙方（受让方）\n名称：\n统一社会信用代码：\n本合同各方经平等自愿协商，根据《中华人民共和国民法典》及相关法律法规，就甲方向乙方转让标的账号事宜，签订本合同以共同遵守",
                "riskPoint": "合同主体信息缺失",
                "riskAnalysis": "甲方（转让方）信息不完整，存在重大风险点：缺少名称和统一社会信用代码，建议补全甲方信息。乙方（受让方）信息不完整，存在重大风险点：缺少名称和统一社会信用代码，建议补全乙方信息。",
                "modifyExample": "甲方（转让方）\n名称：杭州大栅栏科技有限公司\n统一社会信用代码：687654567AAx5678\n乙方（受让方）\n名称：北京百度网讯科技有限公司\n统一社会信用代码：688765167AAx5678\n本合同各方经平等自愿协商，根据《中华人民共和国民法典》及相关法律法规，就甲方向乙方转让标的账号事宜，签订本合同以共同遵守。\n甲方（盖章）：\n法定代表人或授权代表：\n乙方（盖章）：\n法定代表人或授权代表：",
                "positions": [
                  {
                    "pageNum": 1,
                    "box": [
                      90,
                      105,
                      396,
                      176
                    ]
                  }
                ]
              }
            ],
            "predictResult": "```json\n{\n    \"风险等级\": \"重大风险\",\n    \"风险审查结果\": [\n        {\n            \"风险点\": \"合同主体信息缺失\",\n            \"风险分析\": \"甲方（转让方）信息不完整，存在重大风险点：缺少名称和统一社会信用代码，建议补全甲方信息。乙方（受让方）信息不完整，存在重大风险点：缺少名称和统一社会信用代码，建议补全乙方信息。\",\n            \"修改示例\": \"甲方（转让方）\\n名称：杭州大栅栏科技有限公司\\n统一社会信用代码：687654567AAx5678\\n乙方（受让方）\\n名称：北京百度网讯科技有限公司\\n统一社会信用代码：688765167AAx5678\\n本合同各方经平等自愿协商，根据《中华人民共和国民法典》及相关法律法规，就甲方向乙方转让标的账号事宜，签订本合同以共同遵守。\\n甲方（盖章）：\\n法定代表人或授权代表：\\n乙方（盖章）：\\n法定代表人或授权代表：\"\n        }\n    ]\n}\n```",
            "markdownResult": "风险等级：\n 重大风险\n\n风险审查结果：\n [\n        {\n            \"风险点\": \"合同主体信息缺失\",\n            \"风险分析\": \"甲方（转让方）信息不完整，存在重大风险点：缺少名称和统一社会信用代码，建议补全甲方信息。乙方（受让方）信息不完整，存在重大风险点：缺少名称和统一社会信用代码，建议补全乙方信息。\",\n            \"修改示例\": \"甲方（转让方）\\n名称：杭州大栅栏科技有限公司\\n统一社会信用代码：687654567AAx5678\\n乙方（受让方）\\n名称：北京百度网讯科技有限公司\\n统一社会信用代码：688765167AAx5678\\n本合同各方经平等自愿协商，根据《中华人民共和国民法典》及相关法律法规，就甲方向乙方转让标的账号事宜，签订本合同以共同遵守。\\n甲方（盖章）：\\n法定代表人或授权代表：\\n乙方（盖章）：\\n法定代表人或授权代表：\"\n        }\n    ]\n\n",
            "riskName": "重大风险"
          }
        ],
        "fileURL": "https://textmind.bj.bcebos.com//data/mnt/text_flow/parser/doc-1n9d180uds3enztm/pdf/doc-1n9d180uds3enztm.pdf?authorization=bce-auth-v1%2FALTAKt3jZIQbn3oTvNTnKh35nX%2F2024-12-06T09%3A13%3A27Z%2F2592000%2Fhost%2F41be8bf22501e7a5113bdead74aa10d9fd9696cb27cfccb5011d40174c

        "reportURL": "https://textmind.bj.bcebos.com//data/mnt/text_flow/paas_textreview/textreview-task-bey18jc6j14bz92e/document_comment_op/doc-1n9d180uds3enztm/011.docx?authorization=bce-auth-v1%2FALTAKt3jZIQbn3oTvNTnKh35nX%2F2024-12-06T09%3A14%3A58Z%2F2592000%2Fhost%2Fdb074c3732b9051d26075b8a1e71ce5e37e3705733a584f6ff2a8a47c

        "reportStatus": "success"
      }
    ]
  }
}
```

前端SDK渲染 辅助用户在网页中便捷地调用合同审查服务，实现与当前智能文档分析平台-合同审查在线工具一致的前端渲染和交互界面。

使用说明

示例URL： `https://textmind-sdk.bce.baidu.com/textmind/sdk/textreview/{taskId}?access_token={access_token}`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”
taskId	发送提交请求时返回的taskId

iframe引入方式

提示一：使用示例代码前，请记得替换其中的示例Token、taskId。

```
<iframe
  src="https://textmind-sdk.bce.baidu.com/textmind/sdk/textreview/{taskId}?access_token={access_token}"
/>
```

文档比对

接口描述

文档比对支持精准比对文档的增删改差异，快速定位并高亮显示差异原文，支持导出完整的比对报告，大幅提升比对准确性和效率。如希望快速可视化体验效果，可登录智能文档分析平台，一键上传文档，在线测试；**在线工具和API服务的额度共享互通**。

文档比对API服务为**异步接口**，需要先调用**提交请求接口**获取 taskID，然后调用**获取结果接口**进行结果轮询，建议提交请求后5～10秒开始轮询。**提交请求接口**QPS为2，**获取结果接口**QPS为10。

提交请求接口

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/file/2.0/brain/online/v1/textdiff/create_task`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	multipart/form-data

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
baseFile	和 baseFile/baseFileURL 二选一	file	-	文件数据，主版比对文档 支持的文件类型： -图片：bmp/jpg/jpeg/png/tif/tiff -文档：doc/docx/wps/pdf/ofd 支持的文件大小：仅支持上传一篇文件，文件大小不超过50M。图像最短边至少15px，最长边最大4096px

				优先级： baseFile > baseFileURL，当baseFile字段存在时，baseFileURL字段失效
compareFile	和compareFile/compareFileURL 二选一	file	-	文件数据，副版比对文档 支持的文件类型： -图片：bmp/jpg/jpeg/png/tif/tiff -文档：doc/docx/wps/pdf/ofd 支持的文件大小：仅支持上传一篇文件，文件大小不超过50M。图像最短边至少15px，最长边最大4096px **优先级：** compareFile > compareFileURL，当compareFile字段存在时，compareFileURL字段失效
baseFileURL	和baseFile/baseFileURL 二选一	string	-	文件完整URL，仅支持BOS公网访问，URL长度不超过1024字节 支持的文件类型： -图片：bmp/jpg/jpeg/png/tif/tiff -文档：doc/docx/wps/pdf/ofd 支持的文件大小：仅支持上传一篇文件，文件大小不超过50M。图像最短边至少15px，最长边最大4096px **优先级：** baseFile > baseFileURL，当baseFile字段存在时，baseFileURL字段失效 **请注意关闭URL防盗链**
compareFileURL	和compareFile/compareFileURL 二选一	string	-	文件完整URL，仅支持BOS公网访问，URL长度不超过1024字节 支持的文件类型： -图片：bmp/jpg/jpeg/png/tif/tiff -文档：doc/docx/wps/pdf/ofd 支持的文件大小：仅支持上传一篇文件，文件大小不超过50M。图像最短边至少15px，最长边最大4096px **优先级：** compareFile > compareFileURL，当compareFile字段存在时，compareFileURL字段失效 **请注意关闭URL防盗链**
param	是	dict	-	文档比对特殊差异识别参数。默认为false。将指定参数以json格式传递。例如，传递参数{"sealRecognition":true,"handWritingRecognition":true}将分别启用印章识别与手写体识别功能
+ sealRecognition	否	bool	true/false	是否识别印章信息（识别印章信息有额外的时间开销），支持电子签章差异比对。默认为false
+ fullWidthHalfWidthRecognition	否	bool	true/false	是否识别中英文符号差异。默认为false
+ fontFamilyRecognition	否	bool	true/false	是否识别字体差异。默认为false
+ fontSizeRecognition	否	bool	true/false	是否识别字号差异。默认为false
+ handWritingRecognition	否	bool	true/false	是否识别手写体差异。默认为false

请求代码示例

提示：使用示例代码前，请记得替换其中的示例Token、文档地址或Base64信息。

Python

```python
import requests
import os

'''
文档比对-提交请求
'''


request_host = "https://aip.baidubce.com/file/2.0/brain/online/v1/textdiff/create_task"
###### 二进制方式打开图片文件
f1 = '[本地文件-主版比对文档]'
f2 = '[本地文件-副版比对文档]'
body = {
    "baseFile": (os.path.basename(f1), open(f1, 'rb'), "multipart/form-data"),
    "compareFile": (os.path.basename(f2), open(f2, 'rb'), "multipart/form-data"),
}

data = {
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ taskId	string	该请求生成的taskId，后续使用该taskId获取比对结果

返回示例

成功返回示例：

```
{
  "error_code": 0,
  "error_msg": "",
  "log_id": "259575694341050368",
  "result": {
     "taskId": "textreview-task-xnejhkwvcz5qpr3c"
  }
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
  "error_code": 282003,
  "error_msg": "missing parameters",
  "log_id": "259909120864665600"
}
```

获取结果接口

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/file/2.0/brain/online/v1/textdiff/query_task`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	multipart/form-data

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
taskId	是	string	发送提交请求时返回的taskId

请求代码示例

提示：使用示例代码前，请记得替换其中的示例Token、taskId。

Python

```
import requests

'''
文档比对-获取结果
'''


request_host = "https://aip.baidubce.com/file/2.0/brain/online/v1/textdiff/query_task"
params = {"taskId": "[调用提交请求接口获取的taskId]"}

access_token = '[调用鉴权接口获取的token]'
request_url = f"{request_host}?access_token={access_token}"
response = requests.post(request_url, params=params, files=params)
if response:
    print(response.json())
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ taskId	string	任务ID
+ status	string	任务状态，pending：排队中；processing：运行中；success：成功；failed：失败
+ duration	string	任务持续时长
+ errorType	string	任务错误类型，quotaError：配额超限；taskError：任务失败，失败详情可见

		compareMessage
+ param	dict	文档比对特殊差异识别参数
++ sealRecognition	bool	识别印章差异
++ fullWidthHalfWidthRecognition	bool	识别中英文符号差异
++ fontFamilyRecognition	bool	识别字体差异
++ fontSizeRecognition	bool	识别字号差异
++ handWritingRecognition	bool	识别手写体差异
+ subTaskList	dict	文档比对结果列表
++ similarity	string	两份文档的相似度
++ totalDiff	int	两份文档的差异点数量
++ baseDocId	string	主版文档ID
++ baseDocName	string	主版文档名称
++ baseDocOssURL	string	主版文档的对象存储下载链接，**有效期30天**，为解析后的主版文档PDF地址，用于SDK渲染
++ compareDocId	string	副版文档ID
++ compareDocName	string	副版文档名称
++ compareDocOssURL	string	副版文档的对象存储下载链接，**有效期30天**，为解析后的副版文档PDF地址，用于SDK渲染
++ compareStatus	string	比对任务状态，success：成功；failed：失败
++ compareMessage	string	比对任务失败的描述信息
++ reportStatus	string	比对报告状态，success：成功；failed：失败
++ reportMessage	string	比对报告失败的描述信息
++ reportOssURL	string	比对报告的对象存储下载链接，为两份文档的差异项总结报告
++ createdAt	string	任务创建时间
++ finishedAt	string	任务结束时间
++ diffItemList	dict	任务的比对结果
+++ id	string	比对差异点ID
+++ basePageNum	int	主版文档页码
+++ baseDiffType	string	主版差异类型，insert：新增；delete：删除；replace：替换
+++ baseBoxArea	[4]int	主版文档差异点的原文四角点坐标，每个框选范围包含左上角x坐标、左上角y坐标、矩形框宽度、矩形框高度，共四个数值，如[89,74,61,12]
+++ baseDiffBoxes	string	主版文档差异点框的位置，二维数组的字符串，当差异项出现换行时，为多个字符串
+++ baseDiffContent	string	主版差异点内容
+++ baseDiffContext	string	主版差异点上下文

+++ baseDiffContentType	array	主版文档比对差异类型，seal：印章；figure：图片；size：字号；font：字体；full_half_width：中英文符号
+++ comparePageNum	string	副版文档页码
+++ compareDiffType	string	副版差异类型，insert：新增；delete：删除；replace：替换
+++ compareBoxArea	string	副版文档差异点的原文四角点坐标，每个框选范围包含左上角x坐标、左上角y坐标、矩形框宽度、矩形框高度，共四个数值，如[89,74,61,12]
+++ compareDiffBoxes	string	副版文档差异点框的位置，二维数组的字符串，因为一个差异点可能有多个框（换行情况）
+++ compareDiffContent	string	副版差异点内容
+++ compareDiffContext	string	副版差异点上下文
+++ compareDiffContentType	array	副版文档比对差异类型，seal：印章；figure：图片；size：字号；font：字体；full_half_width：中英文符号

返回示例

成功返回示例：


```
{
  "error_code": 0,
  "error_msg": "",
  "log_id": "262826631337504768",
  "result": {
    "taskId": "task-affq81x73t23pqmk",
    "subTaskList": [
      {
        "baseDocId": "doc-egdc1iziztwmech2",
        "baseDocName": "软硬件采购合同主版.docx",
        "baseDocOssURL": "http://ai-textmind.bj.bcebos.com//data/mnt/text_flow/parser/doc-egdc1iziztwmech2/pdf/doc-egdc1iziztwmech2.pdf?authorization=bce-auth-v1%2FALTAKsXn1egl7w69bHDlfE3SOD%2F2023-12-27T05%3A16%3A40Z%2F2592000%2Fhost%2Fe922fe2ac5060e0819f5bcda64139be322a733cdcc357ddea189fa1484

        "compareDocId": "doc-r8xbre9f5r3i8v85",
        "compareDocName": "软硬件采购合同副版.docx",
        "compareDocOssURL": "http://ai-textmind.bj.bcebos.com//data/mnt/text_flow/parser/doc-r8xbre9f5r3i8v85/pdf/doc-r8xbre9f5r3i8v85.pdf?authorization=bce-auth-v1%2FALTAKsXn1egl7w69bHDlfE3SOD%2F2023-12-27T05%3A16%3A40Z%2F2592000%2Fhost%2Fb48b118b7c6360143a7b8096bfa2353956ba918513dfcf432a5b040b1

        "similarity": "84.69%",
        "totalDiff": 30,
        "compareStatus": "success",
        "reportStatus": "success",
        "createdAt": "2023-12-27T05:16:14Z",
        "finishedAt": "2023-12-27T05:17:00Z",
        "reportOssURL": "http://ai-textmind.bj.bcebos.com//data/mnt/text_flow/paas_textdiff/task-affq81x73t23pqmk/textdiff_report_op/subtask-6625h1civfetzdau/subtask-6625h1civfetzdau.pdf?authorization=bce-auth-v1%2FALTAKsXn1egl7w69bHDlfE3SOD%2F2023-12-27T05%3A17%3A00Z%2F2592000%2Fhost%2Fe847aa97b8bc2387f8b635a68f2b727d3705e176fc13abfc02deed9e55

        "diffItemList": [
          {
            "id": "item-uy6tbygc11uqrc3x",
```

```
            "basePageNum": 1,
            "baseDiffType": "replace",
            "baseBoxArea": [
              138,
              210,
              13,
              14
            ],
            "baseDiffBoxes": "[[138,210,13,14]]",
            "baseDiffContent": "西",
            "baseDiffContext": "地址:天津市河西区幸福街**号",
            "comparePageNum": 1,
            "compareDiffType": "replace",
            "compareBoxArea": [
              138,
              210,
              13,
              14
            ],
            "compareDiffBoxes": "[[138,210,13,14]]",
            "compareDiffContent": "东",
            "compareDiffContext": "地址:天津市河东区幸福街**号",
            "baseDiffContentType": [],
            "compareDiffContentType": []
          },
          {
            "id": "item-nhhkrhhj55qduap0",
            "basePageNum": 10,
            "baseDiffType": "replace",
            "baseBoxArea": [
              327,
              490,
              11,
              12
            ],
            "baseDiffBoxes": "[[327,490,11,12]]",
            "baseDiffContent": "百",
            "baseDiffContext": "乙方:北京百度网讯科技有限公司",
            "comparePageNum": 10,
            "compareDiffType": "replace",
            "compareBoxArea": [
              327,
              333,
              11,
              12
            ],
            "compareDiffBoxes": "[[327,333,11,12]]",
            "compareDiffContent": "千",
            "compareDiffContext": "乙方:北京千度网讯科技有限公司",
            "baseDiffContentType": [],
            "compareDiffContentType": []
          }
        ]
      }
    ],
    "status": "success",
    "param": {
      "sealRecognition": false,
      "fullWidthHalfWidthRecognition": false,
      "fontFamilyRecognition": false,
      "fontSizeRecognition": false,
      "handWritingRecognition": false
    }
```

```
        "duration": "46秒"
    }
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
    "error_code": 0,
    "error_msg": "",
    "log_id": "262791511146655744",
    "result": {
        "taskId": "task-x41hqi5se8bkec4u",
        "subTaskList": [
            {
                "baseDocId": "doc-9ek1pqekgqsz2gpk",
                "baseDocName": "软硬件采购合同主版.docx",
                "baseDocOssURL": "",
                "compareDocId": "doc-qqwuga6st7hzshq0",
                "compareDocName": "软硬件采购合同副版.docx",
                "compareDocOssURL": "",
                "similarity": "",
                "totalDiff": 0,
                "compareStatus": "failed",
                "reportStatus": "failed",
                "createdAt": "2023-12-27T03:38:16Z",
                "finishedAt": "2023-12-27T03:38:16Z",
                "reportOssURL": ""
            }
        ],
        "status": "failed",
        "errorType": "quotaError",
        "param": {
            "sealRecognition": false,
            "fullWidthHalfWidthRecognition": false,
            "fontFamilyRecognition": false,
            "fontSizeRecognition": false,
            "handWritingRecognition": false
        },
        "duration": ""
    }
}
```

前端SDK渲染 辅助用户在网页中便捷地调用文档比对服务，实现与当前智能文档分析平台-文档比对在线工具一致的前端渲染和交互界面。

使用说明

示例URL： `https://textmind-sdk.bce.baidu.com/textmind/sdk/textdiff/{taskId}?access_token={access_token}`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”
taskId	发送提交请求时返回的taskId

iframe引入方式

提示一：使用示例代码前，请记得替换其中的示例Token、taskId。

```
<iframe
  src="https://textmind-sdk.bce.baidu.com/textmind/sdk/textdiff/{taskId}?access_token={access_token}"
/>
```

文档抽取

接口描述 文档抽取支持自定义配置字段，无需训练即可抽取文档字段信息，精准定位字段值，适用于合同、票据、订单等各类文档场景。如希望快速可视化体验效果，可登录智能文档分析平台，一键上传文档，在线测试；**在线工具和API服务的额度共享互通**。

文档抽取API服务为**异步接口**，需要先调用**提交请求接口**获取taskId，然后调用**获取结果接口**进行结果轮询，建议提交请求后30秒开始轮询。**提交请求接口**QPS为2，**获取结果接口**QPS为10。

提交请求接口 请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/brain/online/v1/extract/task`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
file	和 fileURLs 二选一	string	文件的 base64编码，需去掉编码头，例如图片类型需去掉编码头（） 支持的文件类型： -图片：jpg/jpeg/png/bmp/tif/tiff -文档：doc/docx/pdf/xlsx/xls/ofd 支持的文件大小：仅支持上传一篇文件，文件大小不超过50M。图像最短边至少15px，最长边最大4096px **优先级：**file>fileURLs，当file字段存在时，fileURLs字段失效
fileName	file不为空时必传	string	文档名称，当传入file参数时，该字段必传，例如test.docx
fileURLs	和file二选一	string	文件数据URL，URL长度不超过1024字节，支持单个URL传入，仅支持bos公网访问。其余文件准入标准与file一致。 **优先级：**file>fileURLs，当file字段存在时，fileURLs字段失效
manifestVersionId	和 manifest 二选一	string	用户在智能文档分析平台配置的清单版本id **优先级：**manifestVersionId>manifest，当manifestVersionId字段存在时，manifest字段失效
manifest	和 manifestVersionId 二选一	string	抽取字段配置，每个抽取字段包含key、parentKey、description三个参数： • key为抽取字段名称，是必传参数； • parentKey为抽取字段的主字段，是非必传参数，不存在时传root或为空； • description为抽取字段的补充说明，用于辅助大模型提升抽取效果，是非必传参数，不存在时为空。 以上三个参数支持中英文、数字、下划线、中划线、斜杠、冒号和括号，其中中横线、下划线、斜杠和冒号不能作为开头和结尾。key和parentKey字符数不超过30，description字符数不超过100。 key的数量不能超过100。 **优先级：**manifestVersionId>manifest，当manifestVersionId字段存在时，manifest字段失效
removeDuplicates	否	bool	是否开启字段值去重。开启后，对单个字段内重复抽取的相同值进行去重，并默认输出首个值的位置信息
pageRange	否	string	是否开启指定页抽取。开启后，可输入指定页码进行抽取，扣费额度也依据指定页码范围 **输入格式**：页码从1开始，使用英文逗号分隔单个页码，用连字符表示页码范围。例如，输入1,5-10,15，表示抽取第1页、第5至10页和第15页，共计扣减8页的额度 **注意**： 鉴于流式文档的特性，dox/docx/wps/xls/xlsx文件的页码解析不固定，建议使用pdf/ofd和图片等版式文档以确保页码解析的准确性
extractSeal	否	bool	是否开启印章抽取。开启后，将抽取文档内的印章信息，但会相应增加任务耗时
eraseWatermark	否	bool	是否开启水印去除。开启后，将去除文档内的水印、底纹、印章等视觉干扰，优化模型抽取效果，但会相应增加任务耗时
docCorrect	否	bool	是否开启图像矫正。开启后，将矫正倾斜图片，但会相应增加任务耗时

manifest参数示例：

```
 [{
 "parentKey": "",
   "key": "生效范围",
 "description": "指国家范围而非省份、城市或更细分的范围"
}, {
 "parentKey": "商品信息",
   "key": "商品编号",
 "description": ""
}, {
 "parentKey": "商品信息",
   "key": "商品价格",
 "description": ""
}]
```

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、文档地址和清单。

提示二：目前仅提供Python语言，如需其他语言示例可参考 **示例代码中心** 。

Python

```python
 import base64
import requests
import os
'''
文档抽取-提交请求
'''
file_path = '[本地文件]'
manifest_version_id = '[清单id]'
request_host = "https://aip.baidubce.com/rest/2.0/brain/online/v1/extract/task"
encoded_string = ''
with open(file_path, 'rb') as file:
    file_base64 = base64.b64encode(file.read()).decode('utf-8')
###### 优先级：manifestVersionId>manifest，当manifestVersionId字段存在时，manifest字段失效
data = {
    'file': file_base64,
    'fileName': os.path.basename(file_path),
    # 'manifestVersionId': manifest_version_id,
    'manifest': """
```

返回说明

返回参数

字段	类型	说明
log_id	string	唯一的 log_id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ taskId	string	该请求生成的taskId，后续使用该taskId获取抽取结果

返回示例 成功返回示例：

```
{
  "error_code": 0,
  "error_msg": "",
  "log_id": "088d6639-bafd-4007-be27-dcddbb651322",
  "result": {
     "taskId": "task-6tb7mgduz9rqaxzi"
  }
}
```


失败返回示例（详细的错误码说明见API文档-错误码）：


```
{
  "error_code": 283016,
  "error_msg": "parameters value error:清单字段名称格式错误",
  "log_id": "debb76a0-8015-4647-be19-f76c5a0f2892",
  "result": None
}
```


获取结果接口 请求说明

请求示例

HTTP 方法：POST 请求URL：https://aip.baidubce.com/rest/2.0/brain/online/v1/extract/query_task

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
taskId	是	string	发送提交请求时返回的taskId

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、taskId。

提示二：目前仅提供Python语言，如需其他语言示例可参考 **示例代码中心** 。

Python

```
 import requests
'''
文档抽取-获取结果
'''

request_host = "https://aip.baidubce.com/rest/2.0/brain/online/v1/extract/query_task"
data = {
    "taskId": "[调用提交请求接口获取的task_id]"
}
access_token = "[调用鉴权接口获取的token]"
request_url = request_host + "?access_token=" + access_token

headers = {'Content-Type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=data, headers=headers)
if response:
 print(response.json())
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ taskId	string	任务ID
+ status	string	任务状态，Pending：排队中；Running：运行中；Success：成功；Failed：失败
+ reason	string	任务失败描述信息
+ createdAt	string	任务创建时间
+ startedAt	string	任务开始时间
+ finishedAt	string	任务结束时间
+ duration	string	任务执行时长
+ extendInfos	dict	高级选项配置
++ removeDuplicates	bool	是否开启字段值去重
++ extractSeal	bool	是否开启印章抽取
++ eraseWatermark	bool	是否开启水印去处
++ docCorrect	bool	是否开启图像矫正
++ pageRange	bool	是否开启指定页抽取
+ extractLabelInfo	[]dict	清单字段配置信息，status为Success时返回
++ key	string	抽取字段名称
++ parentKey	string	抽取字段的主字段
++ description	string	抽取字段的补充说明
+ extractResult	[]dict	文档抽取结果列表
++ docId	string	文档ID
++ docName	string	文档名称

++ pdfUrl	string	源文件转换为PDF后的文档bos下载链接，status为Success时返回，有效期为30分钟
++ data	dict	抽取字段和结果信息
+++singleKey	dict	单个字段抽取结果列表
++++{字段名称}	[]dict	单个字段的字段名称
+++++word	string	单个字段的抽取结果
+++++valuePositions	[]dict	抽取内容位置信息
++++++box	[4] [2]int	抽取结果的四角点坐标框，[[x1,y1],[x2,y2],[,x3,y3],[x4,y4]]，分别对应坐标框四个点的坐标
++++++cbox	[4]int	抽取结果的坐标框，「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点）
++++++pageNo	int	页号
+++comboKey	dict	组合字段抽取结果列表
++++{组合名称}	[]dict	组合字段的组合名称
+++++{字段名称}	dict	组合字段的字段名称
++++++valuePositions	[]dict	抽取内容位置信息
+++++++box	[4] [2]int	四角点坐标框，[[x1,y1],[x2,y2],[,x3,y3],[x4,y4]]，分别对应坐标框四个点的坐标
+++++++cbox	[4]int	坐标框，「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点）
+++++++pageNo	int	页号
+++seal	[]dict	印章字段抽取结果
++++major	string	印章主字段识别内容
++++minor	string	印章子字段识别内容
++++valuePositions	[]dict	抽取内容位置信息
+++++box	[4] [2]int	四角点坐标框，[[x1,y1],[x2,y2],[,x3,y3],[x4,y4]]，分别对应坐标框四个点的坐标
+++++cbox	[4]int	坐标框，「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点）
+++++pageNo	int	页号

返回示例

成功返回示例：


```
{
"error_code": 0,
"error_msg": "",
"log_id": "8ca3c6cf-f9d3-485c-96ee-dedcd9fb9a54",
"result": {
 "taskId": "task-vrs7269yugakeg3g",
 "status": "Success",
 "createdAt": "2024-11-13 02:49:58 +0000 UTC",
 "startedAt": "2024-11-13 02:50:01 +0000 UTC",
 "finishedAt": "2024-11-13 02:50:38 +0000 UTC",
 "duration": 37,
 "reason": "",
 "extractLabelInfo": [{
  "key": "案号",
```

```
  "parentKey": "root",
  "description": ""
 }, {
  "key": "保全金额",
  "parentKey": "root",
  "description": ""
 }, {
  "key": "申请人地址",
  "parentKey": "申请人",
  "description": ""
 }, {
  "key": "申请人联系方式",
  "parentKey": "申请人",
  "description": ""
 }, {
  "key": "被申请人",
  "parentKey": "root",
  "description": ""
 }, {
  "key": "被申请人联系方式",
  "parentKey": "被申请人",
  "description": ""
 }, {
  "key": "申请人",
  "parentKey": "root",
  "description": ""
 }],
 "extractResult": [{
  "docId": "doc-pzvd73huishsjp29",
  "docName": "跨页楼地址.pdf",
  "pdfUrl": "https://model-extract-dev-bj.bj.bcebos.com/paas_extract_doc/50042791/doc-pzvd73huishsjp29/doc-
pzvd73huishsjp29.pdf?authorization=bce-auth-v1%2FALTAK7IDj758EUbA1igu04rHAh%2F2024-11-
13T02%3A50%3A39Z%2F1800%2Fhost%2F3b892fe167f095a5cc63b82b3d69cc9c72d7ee0ae65fde511d1c2a719456e

  "data": {
   "singleKey": {
    "保全金额": [{
     "valuePositions": [{
      "box": [
       [128, 423],
       [186, 423],
       [186, 442],
       [128, 442]
      ],
      "cbox": [128, 423, 58, 19],
      "pageNo": 4
     }, {
      "box": [
       [390, 238],
       [446, 238],
       [446, 256],
       [390, 256]
      ],
      "cbox": [390, 238, 56, 18],
      "pageNo": 4
     }],
     "word": "45000元"
    }]
   },
   "comboKey": {
    "申请人": [{
     "申请人地址": {
      "valuePositions": [{
```

```
        "valuePositions": [{
          "box": [
            [378, 107],
            [501, 107],
            [501, 126],
            [378, 126]
          ],
          "cbox": [378, 107, 123, 19],
          "pageNo": 1
        }, {
          "box": [
            [73, 140],
            [189, 140],
            [189, 158],
            [73, 158]
          ],
          "cbox": [73, 140, 116, 18],
          "pageNo": 1
        }],
        "word": "烟台市莱山区润华大厦1号楼14层"
      },
      "申请人联系方式": {
        "valuePositions": [{
          "box": [
            [149, 172],
            [236, 172],
            [236, 190],
            [149, 190]
          ],
          "cbox": [149, 172, 87, 18],
          "pageNo": 1
        }],
        "word": "13589767069"
      }
    }],
    "被申请人": [{
      "被申请人联系方式": {
        "valuePositions": [{
          "box": [
            [335, 300],
            [421, 300],
            [421, 317],
            [335, 317]
          ],
          "cbox": [335, 300, 86, 17],
          "pageNo": 1
        }, {
          "box": [
            [75, 526],
            [159, 526],
            [159, 542],
            [75, 542]
          ],
          "cbox": [75, 526, 84, 16],
          "pageNo": 1
        }, {
          "box": [
            [335, 654],
            [424, 654],
            [424, 672],
            [335, 672]
          ],
          "cbox": [335, 654, 89, 18],
```

```
    "pageNo": 1
   }, {
    "box": [
     [297, 80],
     [382, 80],
     [382, 96],
     [297, 96]
    ],
    "cbox": [297, 80, 85, 16],
    "pageNo": 2
   }, {
    "box": [
     [65, 241],
     [153, 241],
     [153, 256],
     [65, 256]
    ],
    "cbox": [65, 241, 88, 15],
    "pageNo": 2
   }, {
    "box": [
     [362, 368],
     [449, 368],
     [449, 386],
     [362, 386]
    ],
    "cbox": [362, 368, 87, 18],
    "pageNo": 2
   }, {
    "box": [
     [301, 498],
     [388, 498],
     [388, 515],
     [301, 515]
    ],
    "cbox": [301, 498, 87, 17],
    "pageNo": 2
   }, {
    "box": [
     [69, 664],
     [156, 664],
     [156, 680],
     [69, 680]
    ],
    "cbox": [69, 664, 87, 16],
    "pageNo": 2
   }, {
    "box": [
     [78, 145],
     [161, 145],
     [161, 160],
     [78, 160]
    ],
    "cbox": [78, 145, 83, 15],
    "pageNo": 3
   }, {
    "box": [
     [79, 299],
     [163, 299],
     [163, 314],
     [79, 314]
    ],
```

```
        "cbox": [79, 299, 84, 15],
        "pageNo": 3
       }, {
        "box": [
         [362, 421],
         [448, 421],
         [448, 439],
         [362, 439]
        ],
        "cbox": [362, 421, 86, 18],
        "pageNo": 3
       }, {
        "box": [
         [306, 547],
         [389, 547],
         [389, 564],
         [306, 564]
        ],
        "cbox": [306, 547, 83, 17],
        "pageNo": 3
       }, {
        "box": [
         [306, 676],
         [392, 676],
         [392, 694],
         [306, 694]
        ],
        "cbox": [306, 676, 86, 18],
        "pageNo": 3
       }, {
        "box": [
         [378, 114],
         [462, 114],
         [462, 132],
         [378, 132]
        ],
        "cbox": [378, 114, 84, 18],
        "pageNo": 4
       }],
       "word": "18866655005"
      }
     }, {
      "被申请人联系方式": {
       "valuePositions": [{
        "box": [
         [304, 397],
         [390, 397],
         [390, 413],
         [304, 413]
        ],
        "cbox": [304, 397, 86, 16],
        "pageNo": 1
       }],
       "word": "18615970607"
      }
     }]
    }
   }
  }],
  "quota": 1
 }
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：


```
{
  "error_code": 0,
  "error_msg": "",
  "log_id": "7ea4d878-ed81-49de-88cc-5f910c6b8d1b",
  "result": {
    "taskId": "task-utbgdcp6hqx96d3b",
    "status": "Failed",
    "createdAt": "2024-07-29 07:09:41 +0000 UTC",
    "startedAt": "",
    "finishedAt": "",
    "duration": 0,
    "reason": "no valid doc",
    "extractResult": [
      {
        "docId": "doc-g34h4tsyv7zpsz05",
        "docName": "4b252fcf49f9a7a7042e63e6b6fefbcc1000007.png",
        "data": {}
      }
    ]
  }
}
```


前端SDK渲染 辅助用户在网页中便捷地调用文档比对服务，实现与当前智能文档分析平台-文档抽取在线工具一致的前端渲染和交互界面。

使用说明

示例URL： `https://textmind-sdk.bce.baidu.com/textmind/sdk/textExtract/{taskId}?access_token={access_token}`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”
taskId	发送提交请求时返回的taskId

iframe引入方式

提示一：使用示例代码前，请记得替换其中的示例Token、taskId。

```
<iframe
  src="https://textmind-sdk.bce.baidu.com/textmind/sdk/textExtract/{taskId}?access_token={access_token}"
/>
```

文档解析

接口描述

文档解析支持对doc、pdf、图片、xlsx等18种格式文档进行解析，输出文档的版面、表格、阅读顺序、标题层级、旋转角度等信息，支持中、英、日、韩、法等20余种语言类型，可返回Markdown格式内容，将非结构化数据转化为易于处理的结构化数据，识别准确率可达 90% 以上。

文档解析API服务为**异步接口**，需要先调用**提交请求接口**获取 task_id，然后调用**获取结果接口**进行结果轮询，建议提交请求后 5～10 秒轮询。**提交请求接口**QPS为2，**获取结果接口**QPS为10。

提交请求接口

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/brain/online/v2/parser/task`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
file_data	和file_url二选一	string	-	文件的base64编码数据： -版式文档：pdf、jpg、jpeg、png、bmp、tif、tiff、ofd、ppt、pptx -流式文档：doc、docx、txt、xls、xlsx、wps、html、mhtml PDF文档大小不超过300M，非PDF文档大小不超过50M，文档页数不超过2000页（流式文档按2000字算一页） **优先级：** file_data > file_url，当file_data字段存在时，file_url字段失效
file_url	和file_data二选一	string	-	文件数据URL，URL长度不超过1024字节，支持单个URL传入，若文件大小超过50M，须通过该方式上传。其余文件准入标准与file_data一致。 **优先级：** file_data > file_url，当file_data字段存在时，file_url字段失效 **请注意关闭URL防盗链**
file_name	是	string	-	文件名，请保证文件名后缀正确，例如 "1.pdf "
return_doc_chunks	否	string	-	JSON 字符串，文档切分的传参结构，Python中，json字符串示例 json.dumps({"switch": True, "chunk_size": -1 })，详细结构详见下方说明，具体调用方式详见下方请求示例
+ switch	否	bool	True/False	是否进行文档内容切分，「default=False」
+ chunk_size	否	int	(0,∞)	切分块的大小，按照字符数统计。若段落字符数小于或等于chunk_size，则按chunk_size切分；若大于chunk_size，则整段作为一个切分块，不因chunk_size限制而额外切分。 「default=-1」，表示不限制切分块的大小，按照文档的段落进行切分
+ split_type	否	string	chunk/mark	切分方式。 chunk:按照字符数「chunk_size」切分; mark:按照「separators」中的标点符号切分; 「default=chunk」
+ separators	否	list	。/ ；/ ！/ ？/ ; / ! / ?	切分标点，仅包含表示一句话结束的标点。 「default=["。", "；", "！", "？", ";", "!", "?"]」
recognize_formula	否	bool	True/False	是否对版式类型文档进行公式识别
analysis_chart	否	bool	True/False	是否对统计图表进行解析
				识别语种类型，默认为 CHN_ENG，可选值如下： -CHN_ENG：中英文

language_type	否	string	-	- JAP：日语 - KOR：韩语 - FRE：法语 - SPA：西班牙语 - POR：葡萄牙语 - GER：德语 - ITA：意大利语 - RUS：俄语 - DAN：丹麦语 - DUT：荷兰语 - MAL：马来语 - SWE：瑞典语 - IND：印尼语 - POL：波兰语 - ROM：罗马尼亚语 - TUR：土耳其语 - GRE：希腊语 - HUN：匈牙利语 - THA：泰语 - VIE：越南语 - ARA：阿拉伯语 - HIN：印地语
switch_digital_width	否	string	-	是否对符号进行全半角转换，默认为 auto，可选值如下： -auto：不转换，按模型识别结果输出 - half：将所有的符号转换为半角输出 - full：将所有的符号转换为全角输出

请求代码示例

提示：使用示例代码前，请记得替换其中的示例Token、文档地址或Base64信息。

Python

```
 import requests
import os
import base64


def create_task(url, file_path, file_url):
    """
    Args:
        url: string, 服务请求链接
        file_path: 本地文件路径
        file_url: 文件链接
    Returns: 响应
    """
    # 文件请求
    with open(file_path, "rb") as f:
        file_data = base64.b64encode(f.read())
    data = {
        "file_data": file_data,
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ task_id	string	该请求生成的task_id，后续使用该task_id获取审查结果

返回示例

成功返回示例：

```
{
  "error_code": 0,
  "error_msg": "",
  "log_id": "101385981311373626852735056654 33",
  "result": {
     "task_id": "task-3zy9Bg8CHt1M4pPOcX2q5bg28j26801S"
  }
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
  "error_code": 282003,
  "error_msg": "missing parameters",
  "log_id": "375076310335855445079832539241 41",
  "result": "null"
}
```

获取结果接口

请求说明

请求示例

HTTP 方法：`POST`

请求URL：`https://aip.baidubce.com/rest/2.0/brain/online/v2/parser/task/query`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
task_id	是	string	发送提交请求时返回的task_id

请求代码示例

提示：使用示例代码前，请记得替换其中的示例Token、task_id。

Python

```python
import requests

def query_task(url, task_id):
    """
    Args:
        url: string, 请求链接
        task_id: string, task id
    Returns: 响应
    """
    data = {
        "task_id": task_id
    }
    headers = {'Content-Type': 'application/x-www-form-urlencoded'}
    print(url)
    response = requests.post(url, headers=headers, data=data)
    return response
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ task_id	string	任务ID
+ status	string	任务状态，pending：排队中；processing：运行中；success：成功；failed：失败
+ task_error	string	解析报错信息，包含任务失败、额度不够
+ markdown_url	string	文档解析结果的markdown格式链接，**链接有效期30天**
+ parse_result_url	string	文档解析结果的bos链接，**链接有效期30天**

可通过parse_result_url下载解析结果的JSON文件，parse_result_url的返回参数如下：

字段	类型	说明
file_name	string	文档名称
file_id	string	文档ID
+ pages	list	文件单页解析内容
++ page_id	string	页码ID
++ page_num	int	页码数
++ text	string	当前页的所有纯文字内容
++ layouts	list	页面内容版式分析的结果
+++ layout_id	string	layout ID，layout元素唯一标志，以"xxxxx-layout-{global_layout_index}"形式，global_layout_index为layout元素整个文档的全局索引
+++ text	string	layout对应的文本内容。注：当type为table, image时该字段为空, 需要根据type和layout_id分别到tables, images字段里找到对应的内容
+++ position	list	layout元素在页面中的位置，[x, y, w, h] box框，左上角和宽高
		layout元素类型, 当前可取值：

+++ type	string	• para：段落 • table：表格 • head_tail：页面顶部 • image：文档中的插图 • contents：目录 • seal：印章 • title：标题 • formula：公式
+++ sub_type	string	layout元素子类型，当type为title、image时，subtype有值。 title类的 subtype 包含： • title_{n}，代表n级标题，比如title_2代表二级标题 • image_title：图标题 • table_title：表标题 image类的 subtype 包含： • chart：统计图表 • figure：普通插图 • QR_code：二维码 • Bar_code：条形码
+++ parent	string	标题层级树中父节点的layout ID，若当前layout为一级标题，其parent为 "root"。在table和image的内嵌版面信息中暂时都为空
+++ children	list	标题层级树中子节点的layout ID。在table和image的内嵌版面信息中暂时都为空
++ tables	list	页面表格解析结果
+++ layout_id	string	layout ID，与layouts中的元素type为table的元素的layout ID对应
+++ markdown	string	表格内容的markdown形式
+++ table_title_id	list	表格标题对应的layout_id，默认为null
+++ position	list	边框数据 「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点），版式格式时有效
+++ cells	list	单元格的内嵌版面信息，layout类型为表格时有值
+++ matrix	list	二位数组 表示表格内布局位置信息，每个元素对应cells列表中元素的索引
+++ merge_table	string	「begin」- 跨页表格开始、「inner」- 跨页表格中间表格(表格跨页超过两页)、「end」- 跨页表格结束；非跨页表格该字段为空
++ images	list	页面中图片解析结果
+++ layout_id	string	layout ID，与layouts中的元素type为image的元素的layout ID对应
+++ image_title_id	list	图片标题对应的layout_id，默认为null
+++ position	list	边框数据 「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点），版式格式时有效
+++ content_layouts	list	图片的内嵌版面信息
+++ data_url	string	图片存储链接
+++ image_description	string	对统计图表进行内容解析和描述，输出结果为json字符串，可通过json.loads结构化为json格式
++ meta	dict	页面元信息

+++ page_width	int	页面宽度
+++ page_height	int	页面高度
+++ is_scan	bool	是否扫描件
+++ page_angle	int	页面倾斜角度
+++ page_type	string	页面属性「text」- 正文、「contents」- 目录、「appendix」- 附录、「others」- 其他
+++ sheet_name	string	excel的sheet名
+ chunks	list	文件内容切分结果，return_doc_chunks中switch为True时有值
++ chunk_id	string	切片的ID
++ content	string	切片的内容
++ type	string	切片类型, 为text或者table
++ meta	dict	chunk元信息
+++ title	list	chunk所属的多级标题内容
+++ position	list	chunk的位置，根据分块算法有可能chunk跨多个页
+++ box	list	chunk的位置坐标
+++ page_num	int	chunk内容所在页数

表格解析结构说明

以下图为例：

<table>
<tr><td>0 序号</td><td>1 客户名称</td><td>2 金额</td></tr>
<tr><td>3 1</td><td>4 百度在线</td><td>5 100</td></tr>
<tr><td>6 2</td><td>7 百度网讯</td><td>8 100</td></tr>
<tr><td>9 3</td><td>10 百度时代</td><td>11 100</td></tr>
<tr><td colspan="2">12 合同专用章 12
12 公章 12</td><td>13
14</td></tr>
</table>

```
{
###### cells列表包含14个元素，matrix中的每个数字表示一个单元格在cells列表中的索引。
  "cells": [
    {"layout_id": "layout-xxxx",
     "position": [90, 376, 21, 10],
     "text": "序号"
     ...
    }, # ... 其他单元格信息
  ],
  "matrix": [
    [0, 1, 2],
    [3, 4, 5],
    [6, 7, 8],
    [9, 10, 11],
    [12, 12, 13],
    [12, 12, 14]

}
```

返回示例

成功返回示例：


```
{
  "log_id": "2359659789928692176157936558237 3",
  "error_code": 0,
  "error_msg": "",
  "result":
  {
    "task_id": "task-UnvGsgbYZp9pS3BZRHn11ifzjNvKzTgf",
    "status": "success",
    "task_error": null,
    "duration": 902.0,
    "parse_result_url": "https:xxxxxxxxxxxxxxxxxx"
  }
}
```


解析结果示例：


```
{
  "file_name": "示例文件1（文字+表格）更新-改-1.pdf",
  "file_id": "file-u9kVDu6dtwMyNrizbejMIF8A852aJLm2",
  "pages": [
    {
      "page_id": "2aJLm2-page-0",
      "page_num": 0,
      "text": "买卖合同\n甲、乙双方根据《中华人民共和国合同法》及其它相关法律、法规的规定，本着平等、自愿、互利的原则，经友好协商，订立本合同，以资共同信守：\n1.合同标的物信息\n| 序号 | 商品名称 | 产品简称 | 单价 | 数量 | 总价 | 税率 | 备注 |\n| --- | --- | --- | --- | --- | --- | --- | --- |\n| 1 | 软件-AI | 中台内网-推理平台+训练平台 | 95,000 | 1 | 905,000 | 13% | 第一单元 |\n| 2 | 硬件【A】 | 服务器 | 30,250 | 37 | 1,11950 | 13% | 第一单元 |\n\n本合同一式【2】份，经双方代表签字盖章生效。甲乙双方各执【3】份，具有同等法律效力。\n1 \n",
      "layouts": [
        {
          "layout_id": "2aJLm2-layout-1",
          "text": "买卖合同",
          "position": [
            263,
            109,
            103,
            28
          ],
          "type": "title"
```

```
            "type": "title",
            "sub_type": "title_1",
            "parent": "root",
            "children": [
                "2aJLm2-layout-2",
                "2aJLm2-layout-3"
            ]
        },
        {
            "layout_id": "2aJLm2-layout-2",
            "text": "甲、乙双方根据《中华人民共和国合同法》及其它相关法律、法规的规定，本着平等、自愿、互利的原则，经友好协商，订立本合同，以资共同信守：",
            "position": [
                84,
                160,
                444,
                31
            ],
            "type": "text",
            "sub_type": "",
            "parent": "2aJLm2-layout-1",
            "children": [

            ]
        },
        {
            "layout_id": "2aJLm2-layout-3",
            "text": "1.合同标的物信息",
            "position": [
                79,
                206,
                110,
                14
            ],
            "type": "title",
            "sub_type": "title_2",
            "parent": "2aJLm2-layout-1",
            "children": [
                "2aJLm2-layout-4",
                "2aJLm2-layout-5",
                "2aJLm2-layout-6"
            ]
        },
        {
            "layout_id": "2aJLm2-layout-4",
            "text": "",
            "position": [
                82,
                224,
                452,
                97
            ],
            "type": "table",
            "sub_type": "",
            "parent": "2aJLm2-layout-3",
            "children": [

            ]
        },
        {
            "layout_id": "2aJLm2-layout-5",
            "text": "本合同一式【2】份，经双方代表签字盖章生效。甲乙双方各执【3】份，具有同等法律效力。",
            "position": [
```

```
                79,
                348,
                456,
                31
            ],
            "type": "text",
            "sub_type": "",
            "parent": "2aJLm2-layout-3",
            "children": [

            ]
        },
        {
            "layout_id": "2aJLm2-layout-6",
            "text": "1",
            "position": [
                305,
                745,
                11,
                12
            ],
            "type": "head_tail",
            "sub_type": "",
            "parent": "2aJLm2-layout-3",
            "children": [

            ]
        }
    ],
    "tables": [
        {
            "layout_id": "2aJLm2-layout-4",
            "markdown": "| 序号 | 商品名称 | 产品简称 | 单价 | 数量 | 总价 | 税率 | 备注 |\n| --- | --- | --- | --- | --- | --- | --- | --- |\n| 1 | 软件-AI | 中台内网-推理平台+训练平台 | 95,000 | 1 | 905,000 | 13% | 第一单元 |\n| 2 | 硬件【A】 | 服务器 | 30,250 | 37 | 1,11950 | 13% | 第一单元 |\n",
            "position": [
                82,
                224,
                452,
                97
            ],
            "cells": [
                {
                    "layout_id": "2aJLm2-layout-4-0",
                    "text": "序号",
                    "position": [
                        82,
                        224,
                        36,
                        28
                    ],
                    "type": "text",
                    "sub_type": "",
                    "parent": "",
                    "children": null
                },
                {
                    "layout_id": "2aJLm2-layout-4-1",
                    "text": "商品名称",
                    "position": [
                        118,
                        224,
```

```
            78,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-2",
        "text": "产品简称",
        "position": [
            196,
            224,
            92,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-3",
        "text": "单价",
        "position": [
            288,
            224,
            57,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-4",
        "text": "数量",
        "position": [
            345,
            224,
            43,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-5",
        "text": "总价",
        "position": [
            387,
            224,
            61,
            28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
```

```
        "children": null
      },
      {
        "layout_id": "2aJLm2-layout-4-6",
        "text": "税率",
        "position": [
          448,
          224,
          46,
          28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
      },
      {
        "layout_id": "2aJLm2-layout-4-7",
        "text": "备注",
        "position": [
          494,
          224,
          41,
          28
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
      },
      {
        "layout_id": "2aJLm2-layout-4-8",
        "text": "1",
        "position": [
          82,
          252,
          36,
          44
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
      },
      {
        "layout_id": "2aJLm2-layout-4-9",
        "text": "软件-AI ",
        "position": [
          118,
          252,
          78,
          44
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
      },
      {
        "layout_id": "2aJLm2-layout-4-10",
        "text": "中台内网-推理平台+训练平台",
        "position": [
          196,
```

```
            252,
            92,
            44
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-11",
        "text": "95,000",
        "position": [
            288,
            252,
            57,
            44
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-12",
        "text": "1",
        "position": [
            345,
            252,
            43,
            44
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-13",
        "text": "905,000",
        "position": [
            387,
            252,
            61,
            44
        ],
        "type": "text",
        "sub_type": "",
        "parent": "",
        "children": null
    },
    {
        "layout_id": "2aJLm2-layout-4-14",
        "text": "13% ",
        "position": [
            448,
            252,
            46,
            44
        ],
        "type": "text",
        "sub_type": "",
```

```
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-15",
            "text": "第一单元",
            "position": [
                494,
                252,
                41,
                71
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-16",
            "text": "2",
            "position": [
                82,
                295,
                36,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-17",
            "text": "硬件【A】",
            "position": [
                118,
                295,
                78,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-18",
            "text": "服务器",
            "position": [
                196,
                295,
                92,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-19",
            "text": "30,250",
```

```
            "position": [
                288,
                295,
                57,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-20",
            "text": "37",
            "position": [
                345,
                295,
                43,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-21",
            "text": "1,11950",
            "position": [
                387,
                295,
                61,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        },
        {
            "layout_id": "2aJLm2-layout-4-22",
            "text": "13% ",
            "position": [
                448,
                295,
                46,
                28
            ],
            "type": "text",
            "sub_type": "",
            "parent": "",
            "children": null
        }
    ],
    "matrix": [
        [
            0,
            1,
            2,
            3,
            4,
            5,
            6
```

```
                    7
                ],
                [
                    8,
                    9,
                    10,
                    11,
                    12,
                    13,
                    14,
                    15
                ],
                [
                    16,
                    17,
                    18,
                    19,
                    20,
                    21,
                    22,
                    15
                ]
            ],
            "merge_table": ""
        }
    ],
    "images": [
        {
            "layout_id": "Kr9RM7-layout-10",
            "image_title_id": null,
            "position":
            [
                90,
                549,
                422,
                221
            ],
            "data_url": " ",
            "image_description": " {\n    \"title\": \"None\",\n    \"source\": \"None\",\n    \"x_title\": \"图2 统计图\",\n    \"y_title\": [\n        \"None\",\n        \"None\"\n    ],\n    \"values\": {\n        \"轻客出口（万辆）\": {\n\"2017\": \"2.0\",\n            \"2018\": \"1.9\",\n            \"2019\": \"2.0\",\n            \"2020\": \"1.2\",\n\"2021\": \"1.1\",\n            \"2022\": \"1.4\",\n            \"2023Q1\": \"0.4\"\n        },\n        \"中客出口（万辆）\": {\n            \"2017\": \"0.4\",\n            \"2018\": \"0.2\",\n            \"2019\": \"0.4\",\n\"2020\": \"0.4\",\n            \"2021\": \"0.3\",\n            \"2022\": \"0.6\",\n            \"2023Q1\": \"0.0\"\n},\n        \"大客出口（万辆）\": {\n            \"2017\": \"3.3\",\n            \"2018\": \"3.6\",\n            \"2019\": \"4.1\",\n            \"2020\": \"2.5\",\n            \"2021\": \"2.3\",\n            \"2022\": \"2.8\",\n\"2023Q1\": \"0.7\"\n        },\n        \"客车出口占客车总销量的比例（%）\": {\n            \"2017\": \"10.8\",\n\"2018\": \"11.6\",\n            \"2019\": \"13.6\",\n            \"2020\": \"9.1\",\n            \"2021\": \"7.3\",\n\"2022\": \"11.8\",\n            \"2023Q1\": \"12.0\"\n        }\n    }\n}"
        }
    ],
    "meta": {
        "page_width": 612,
        "page_height": 792,
        "is_scan": false,
        "page_angle": 0,
        "page_type": "text",
        "sheet_name": ""
    }
],
"chunks": [
    {
```

```
        "chunk_id": "2aJLm2-chunk-0",
        "content": "甲、乙双方根据《中华人民共和国合同法》及其它相关法律、法规的规定，本着平等、自愿、互利的原则，经友好协商，订立本合同，以资共同信守：",
        "type": "text",
        "meta": {
          "title": [
            "买卖合同"
          ],
          "position": [
            {
              "box": [
                84,
                160,
                444,
                31
              ],
              "page_num": 0
            }
          ]
        }
      },
      {
        "chunk_id": "2aJLm2-chunk-1",
        "content": "| 序号 | 商品名称 | 产品简称 | 单价 | 数量 | 总价 | 税率 | 备注 |\n| --- | --- | --- | --- | --- | --- | --- | --- |\n| 1 | 软件-AI | 中台内网-推理平台+训练平台 | 95,000 | 1 | 905,000 | 13% | 第一单元 |\n| 2 | 硬件【A】 | 服务器 | 30,250 | 37 | 1,11950 | 13% | 第一单元 |\n",
        "type": "table",
        "meta": {
          "title": [
            "买卖合同",
            "1.合同标的物信息"
          ],
          "position": [
            {
              "box": [
                82,
                224,
                452,
                97
              ],
              "page_num": 0
            }
          ]
        }
      },
      {
        "chunk_id": "2aJLm2-chunk-2",
        "content": "本合同一式【2】份，经双方代表签字盖章生效。甲乙双方各执【3】份，具有同等法律效力。",
        "type": "text",
        "meta": {
          "title": [
            "买卖合同",
            "1.合同标的物信息"
          ],
          "position": [
            {
              "box": [
                79,
                348,
                456,
                31
              ],
```

```
                "page_num": 0
            }
        ]
    }
}
]
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{"log_id": "13665091038742503867108513247608",
"error_code": "282007",
"error_msg": "task not exist, please check task id",
"result": "null"}
```

错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
 "error_code": 110,
 "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	未知错误，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如"-"，可手动输入重试
4	Open api request limit reached	集群超限额，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
6	No permission to access data	无接口调用权限，创建应用时未勾选相关文字识别接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口后重新调用，也可使用权限额度诊断工具完成自助诊断
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	免费测试资源使用完毕，每天请求量超限额，已支持计费的接口，您可以在控制台文字识别服务选择购买相关接口的次数包或开通按量后付费；邀测和未支持计费的接口，您可以在控制台提交工单申请提升限额，也可先使用权限额度诊断工具完成自助诊断
	Open api qps	QPS超限额，免费额度并发限制为2QPS，开通按量后付费或购买次数包后并发限制为10QPS，如

18	request limit reached	您需要更多的并发量，可以选择购买QPS叠加包；邀测和未支持计费的接口，您可以在控制台提交工单申请提升限额，也可先使用权限额度诊断工具完成自助诊断
19	Open api total request limit reached	请求总量超限额，已支持计费的接口，您可以在控制台文字识别服务选择购买相关接口的次数包或开通按量后付费；邀测和未支持计费的接口，您可以在控制台提交工单申请提升限额
100	Invalid parameter	无效的access_token参数，token拉取失败，您可以参考“Access Token获取”文档重新获取
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，请注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，请注意需要定期更换，也可以每次请求都拉取新token
216200	empty file or fileurl	文件或文件路径为空
216201	file format error	文件格式错误，现阶段我们支持的文件格式为：PDF、doc、docx
216202	file size error	文件大小异常，现阶段我们支持文件大小为10M
282000	internal error	任务处理失败，请重新提交任务，若持续出现此类错误，请通过工单联系技术支持
282001	template not found	合同类型不存在，请检查传递的合同类型名称
282003	missing parameters	请求参数缺失
282005	quota exceed error	配额超限，请重新申请配置
282006	check user auth error	用户权限校验失败，请检查用户是否拥有该权限
282007	task not exist, please check task id	任务不存在，请检查任务ID
282111	url format illegal	URL格式非法，请检查URL格式是否符合相应接口的入参要求
282112	url download timeout	URL下载超时，请检查URL对应的文件是否无法下载或链路状况不好
282113	url response invalid	URL返回无效参数
282114	url size error	URL长度超过1024字节或为0
283016	parameters value error	请求参数不合法

购买指南

产品价格

文档格式转换

支持文档格式转换的在线工具和接口调用两种使用方式。 首次进入在线工具-文档格式转换 页面即可自动领取免费资源，或者访问控制台-概览页 实名用户也可以自动领取免费测试资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或 开通按量后付费。 支持整份多页PDF/图

片/OFD转换，**按文件页数计费**，转换一页PDF/一张图片扣除一页。 **预付费资源包** | 规格（页） | 价格（元） | | ---------- | ---------- | | 100 | 18 | | 1000 | 180 | | 5000 | 850 | | 1万 | 1600 | | 5万 | 7500 | | 10万 | 14000 | | 20万 | 26000 | | 50万 | 55000 | | 100万 | 90000 | | 500万 | 350000 |

> **说明：**
>
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 文档格式转换（元/页） | | ---------- | ---------- | | 不限量 | 0.18 | **说明：** “调用次数”只包括成功调用，调用失败不计费

合同审查

支持合同审查的在线工具和接口调用两种使用方式。 首次进入在线工具-合同审查 页面即可自动领取免费资源，或者在控制台-免费资源领取页 领取免费资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。 免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或开通按量后付费。 **预付费资源包** | 规格（页） | 价格（元） | | ---------- | ---------- | | 100 | 200 | | 1000 | 1800 | | 1万 | 16000 | | 5万 | 75000 | | 10万 | 140000 | | 50万 | 550000 | | 100万 | 900000 |

> **说明：**
>
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 合同审查（元/页） | | ---------- | ---------- | | 不限量 | 2 | **说明：** “调用次数”只包括成功调用，调用失败不计费

文档比对

支持文档比对的在线工具和接口调用两种使用方式。 首次进入在线工具-文档比对 页面即可自动领取免费资源，或者在控制台-免费资源领取页 领取免费资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或开通按量后付费。 **预付费资源包** | 规格（页） | 价格（元） | | ---------- | ---------- | | 100 | 20 | | 1000 | 180 | | 1万 | 1600 | | 5万 | 7500 | | 10万 | 14000 | | 50万 | 55000 | | 100万 | 90000 |

> **说明：**
>
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 文档比对（元/页） | | ---------- | ---------- | | 不限量 | 0.2 | **说明：** “调用次数”只包括成功调用，调用失败不计费

文档抽取

支持文档抽取的在线工具和接口调用两种使用方式。 首次进入在线工具-文档抽取 页面即可自动领取免费资源，或者在控制台-免费资源领取页 领取免费资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。 免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或开通按量后付费。

预付费资源包 | 规格（页） | 价格（元） | | ---------- | ---------- | | 1000 | 200 | | 5000 | 950 | | 1万 | 1800 | | 5万 | 8500

| | 10万 | 16000 | | 20万 | 30000 | | 50万 | 65000 | | 100万 | 110000 | | 500万 | 450000 |

> **说明：**
>
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 文档抽取（元/页） | | ---------- | ---------- | | 不限量 | 0.2 | **说明：** “调用次数”只包括成功调用，调用失败不计费

文档解析

文档解析支持接口调用的使用方式。 首次使用可在控制台-免费资源领取页领取免费资源。**个人认证 200 页，企业认证 200 页**，有效期均为 365 天。免费测试资源用尽后按照如下价格进行计费。如需付费使用，可购买资源包或开通按量后付费。

预付费资源包 | 规格（页） | 价格（元） | | ---------- | ---------- | | 1000 | 180 | | 5000 | 850 | | 1万 | 1600 | | 5万 | 7500 | | 10万 | 14000 | | 20万 | 26000 | | 50万 | 55000 | | 100万 | 90000 | | 500万 | 350000 |

> **说明：**
>
> - 资源包购买后一年内有效，有效期内产生计费调用量会抵扣资源包额度，超出有效期未抵扣额度自动失效，无法继续使用，请根据实际业务需求酌情购买
> - 资源包购买后7天内若未产生调用可前往退订管理页面进行自助退订，详细退订规则见退订说明 **按量后付费** | 月调用量 | 文档解析（元/页） | | ---------- | ---------- | | 不限量 | 0.18 | **说明：** “调用次数”只包括成功调用，调用失败不计费

iOCR自定义文字识别

更新记录

更新记录

时间	更新说明
2020-03-27	iOCR通用版新增 QPS 叠加包售卖，详情参见价格文档
2020-03-26	iOCR通用版及iOCR财会版整体升级，新增预置模板功能及关键词辅助分类功能 - 预置模板无需制作或编辑即可直接在“模板管理”中查看或进行 API 调用 - 关键词辅助分类功能可支持基于关键词的分类器训练，将图片中的特异性文字作为分类依据
2019-11-07	iOCR通用版增加 次数包售卖方式
2019.07.25	视觉效果及交互流程全面升级: 1. 增加模板搜索及按发布状态筛选功能，提升检索效率 2. 扩大模板编辑区面积，优化模板编辑流程，提升模板制作效率
2019.06.05	**iOCR财会版上线**，提供财会场景专项解决方案
2019.04.25	**正式商用**，接入计费系统，开通付费即可享受 10QPS 无限量调用
2019.02.27	新增模板矩形裁切、透视裁切功能，让模板更规范，后期识别率更高；新增支持模板替换功能；模板匹配算法优化，识别准确率提升；同时对产品易用性做了多项提升
2019.02.21	优化了浏览器兼容性，新增对Firefox、Safari、IE11浏览器的支持
2019.02.11	分类器中加入增值税发票、出租车票、火车票、定额发票、行程单、大陆护照、机动车销售发票、车辆合格证、户口本的预置分类，用户直接勾选后即可实现对应票据的自动分类并结构化
2019.01.25	模板列表增加模板图片缩略图，查找模板更方便直观
2018.12.27	在识别区字段类型中新增手写数字识别类型，针对手写数字识别率更高
2018.10.31	新增自定义字段类型功能，在“字段类型管理”中针对返回值是有限集合的字段，用户可以上传词典，对该字段输出值进行限定
2018.10.10	分类模型升级，百度自有测试集上，准确率由95%左右全部提升到100%（视具体情况准确率可能有所波动）
2018.08.30	检测模型升级，识别耗时降低20%
2018.08.16	识别模型升级，漏识别率降低15.15%，准确率提升46.68%
2018.07.19	1.新增识别区属性选择，能针对特定类型的识别区提高识别率 2.新增返回字段置信度 3.表格识别区新增表格线判断功能，识别准确率提升 4.新增参照字段框选引导
2018.06.04	新增票据自动分类功能
2018.03.16	产品易用性提升，优化使用流程
2017.12.25	**产品上线**

概览

功能介绍

iOCR自定义模板文字识别是一款针对固定版式卡证票据提供的 OCR 定制化产品，可由用户自助创建识别模板和分类器，实现对任意版式卡证票据进行自动分类并结构化输出识别结果。

该产品提供三大自定义功能，分别为：

自定义模板

针对需要识别的图片版式，上传一张模板图片，通过框选参照字段和识别区即可自助制作一个识别模板，并建立图片中文字的 Key-Value 对应关系，实现对相同版式图片的结构化识别。

> **说明：**
>
> **参照字段：** 相同版式的不同图片中位置和内容固定不变的字段，可框选做为图片的锚点，用做对后续传入的图片进行模板匹配和矫正
>
> **识别区**：图片中需要进行识别的字段，可通过框选及命名构建 「字段名称：识别区内容」 的 Key：Value 对应关系，用于对后续传入的相同版式图片的相同位置内容进行结构化识别

自定义分类器

针对已发布的多个识别模板，只需每类上传 30 张训练集图片或填写分类关键词即可创建分类器，实现对不同版式图片的自动分类，省去人工分类成本，一步实现图片的自动分类和结构化识别

同时，还可选择系统预置模板，无需上传训练集或填写分类关键词即可对预置模板同版式图片进行分类识别

训练完成后的分类器有以下三种分类形式：

- **纯图片特征分类器：**通过图片训练集训练的分类模型，对上传的图片进行图像特征值匹配，将相同版式图片分类到对应的识别模板。如需使用此种分类器，需保证所有模板均上传了 30 张以上相同版式训练集图片，但均未填写分类关键词
- **纯文字特征分类器：**针对分类器中的各模板填写分类关键词，对上传的图片进行全版面文字识别，如识别出的文字信息包含所填的关键词则分类到对应的识别模板。如需使用此种分类器，无需上传训练集图片，但需保证所有模板均填写了分类关键词
- **图文协同分类器：**结合关键词文字信息及图片特征进行协同分类，对上传的图片先进行全版面文字识别匹配关键词，如一张图片匹配到不同模板的分类关键词，则再依据图片特征进行区分。如需使用此种分类器，需保证所有模板均填写了分类关键词，且存在部分模板上传了 30 张以上相同版式训练集图片

> **说明：**
>
> **图片训练集：** 针对已发布的识别模板上传的相同版式的图片集合，用于训练分类器对后续上传的图片进行自动分类；为了最佳的分类效果，训练集至少包含 30 张以上不重复的、版式相同的图片
>
> **分类关键词：**图片中存在的独有的文字内容，用作模板分类的文字依据，需保证填写的关键词在该版式图片中均有出现，可根据填写的关键词唯一确定图片所属模板类别。*如身份证人像面的“姓名”、“性别”、“民族”、“出生”、“公民身份号码”五个字段在每一张身份证人像面均会出现，且 5 个字段全部出现时基本可确定该图片为身份证人像面，则此 5 个字段即可作为身份证人像面的分类关键词*

自定义字段类型

针对输出值为有限集的字段，用户可将可能的输出值汇总为字段词典进行上传，在框选识别区后选择该字段类型，系统则会对识别结果进行智能匹配或纠正，用于规范识别结果，并提高识别准确率。

产品分类及对比

目前 iOCR 共分为 **通用版** 和 **财会版** 两个版本，分别适用于不同的使用场景。

产品分类

- iOCR通用版

针对任意固定版式的卡证票据，可自助创建识别模板和分类器，实现图片的自动分类及结构化识别

- **iOCR财会版**

  针对财会场景常用的各类发票及银行单据进行专项优化及整合，预置多种票据、单据模板及分类器，无需制作或训练即可直接使用；并提供混贴票据识别功能，可对粘贴在一张报销单上的多张不同种类发票进行切分识别；同时支持对未预置的固定版式票据、单据定制结构化识别模板和分类器

5分钟制作自定义模板

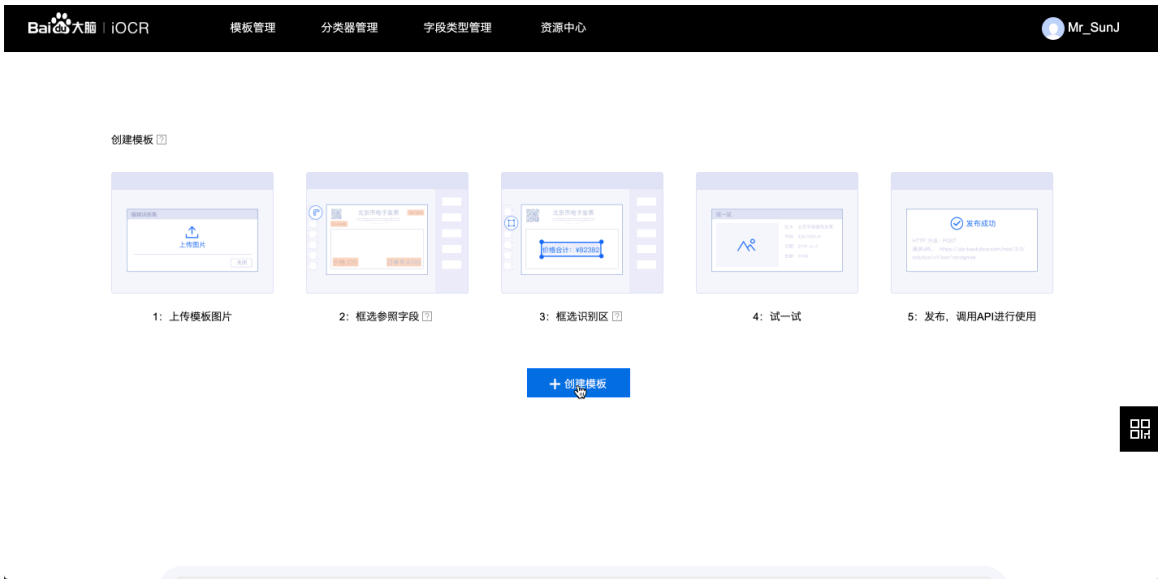
> 以 **iOCR通用版** 为例

创建自定义模板的基本流程如下图所示，仅需上传一张规范的模板图片，通过简单框选，5 分钟即可完成结构化识别模板的制作。视频教程请参见 iOCR通用版使用教程（视频版）。



Step 1: 上传模板图片

在iOCR - 模板管理页面点击下方 「**创建模板**」按钮，在弹出框中上传一张**字迹清晰且摆放端正**的模板图片（大小不超过4M，最长边不超过4096像素），并对模板进行命名。



Step 2: 框选参照字段

进入模板编辑页面，右侧操作步骤中选择 「**第1步：框选参照字段**」 标签，左侧工具栏选择 「**框选参照字段**」 按钮，使用鼠标在模板图片中框选位置和内容都固定不变的文字，如下图所示**橘色矩形**框选区域。



注意： 「参照字段」为相同版式的不同图片中位置和内容固定不变的字段，可做为图片的锚点，用做对后续传入的图片进行模板匹配和矫正

框选Tips：

1. 参照字段个数需保证在4个以上（推荐8个以上），并尽量分散在四角
2. 单个参照字段不可跨行，推荐字数在4个以内
3. 参照字段文字内容在上下文中不会重复出现
4. 仅支持框选中英文、数字，不可包含符号、图案

Step 3: 框选识别区

右侧操作步骤中点击 「**第2步：框选识别区**」 标签，左侧工具栏选择 「**框选识别区**」 按钮，使用鼠标在模板图片上框选业务场景需要进行识别的字段，如下图所示**蓝色矩形**框选区域；同时，填写「**字段名称**」，并选择合适的「**字段类型**」以提高识别准确率。



注意：「识别区」为图片中需要进行识别的字段，可通过框选及命名构建 「字段名称：识别区内容」 的 Key：Value 对应关系，用于对后续传入的相同版式图片的相同位置内容进行结构化识别

框选Tips：

1. 尽量扩大识别区框选范围，保证后续传入图片的对应字段内容可被完全覆盖，但同时也需保证不框选到其他字段内容
2. 选择合适的字段类型有助于提升字段识别效果，也可自定义字段类型进行使用
3. 如需识别图片内列宽固定的表格，可点击工具栏中「插入表格」按钮框选表格识别区

Step 4: 试一试

参照字段和识别区全部框选完毕后，可点击页面右上角的「**试一试**」按钮进行识别效果测试，在弹出框中上传任意一张相同版式的图片即可，如下图所示。可点击图片下方「更换图片」按钮更换测试图片，如多次测试效果满意即可进行发布；如效果不满意可返回继续编辑。



注意：

1. 如试一试结果出现图片无法匹配模板的情况，需确认上传的测试图片与模板图片是否为同一版式，如确认无误可调节参照字段框选范围或更换参照字段，以提升模板匹配准确率
2. 如试一试结果中出现识别结果错误的情况，可调整识别区框选范围或更换识别区字段类型，以提升识别准确率

Step 5: 发布模板，调用API进行使用

如测试效果满意，可点击试一试弹出框右下角的「**立即发布**」按钮或模板编辑页面右上角的「**发布**」按钮进行发布，发布成功后即可通过模板ID调用该模板，调用方式可查看API文档。



注意：只有发布后的模板才能通过线上接口进行调用，如果编辑未发布，那么仅仅是生成了一个新的版本，此时对模板的任何修改都不会影响线上调用。

如您在操作过程中出现上述内容未说明的问题，可参考 iOCR 常见问题，或在论坛发布您的问题。您也可加入百度 iOCR 交流群（群号:570832882）与更多开发者进行交流。

iOCR全场景识别

简介

概述

iOCR 全场景识别平台是一款基于 MoE 混合专家方案打造的一站式文档智能识别平台，依托于百度 OCR 模型与文心大模型 3.5，无需定制训练、无需分类后调用，致力于提供更加**全面、精准、易用的文档结构化服务**，有效提升企业内部费用报销、保险理赔、贷款审批等业务处理效率。

平台预置意图识别大模型，可根据任务内不同文档类型，自动推荐并分发至对应专家模型，包括 10 余种 OCR 模型（支持结构化识别 20 多类文件）与通用抽取大模型（支持自动抽取任意文档的关键字段），从繁琐的人工分类、复杂的多接口调用升级为**一站式全自动处理**。此外，可在平台自定义文档抽取模型，不限类型与版式，只需上传一张图片、输入待抽取的字段，**仅需 1 分钟，即可构建新模型**，从此告别采集、标注、训练等传统模型构建的漫长流程。

使用方式

支持在线平台和API服务两种使用方式。

如需快速体验效果，可进入示例项目空间 ，一键上传文档，即可完成测试；如需使用API服务，可参见API文档。在线工具和API服务的额度共享互通。

核心功能

iOCR 全场景识别提供以下两大自定义功能：**自定义项目空间、自定义抽取模型**，结合使用，可实现对全场景不同类型文档的自动分类与结构化识别，分类准确率可达 99%，识别准确率可达 95%，详细介绍如下。

自定义项目空间

可基于业务需求，自主创建项目空间并配置识别规则，不同空间下，任务数据完全隔离，有效保障数据隐私性。

- **一键勾选所需模型**：平台已预置丰富的预置模型，同时支持用户自定义我的模型，只需一键勾选，即可快速开启使用；
- **灵活选择兜底策略**：若出现已选模型未覆盖的文件类型，可从全文识别、结构化识别、拒识别中任选一种作为兜底策略，避免业务流中断。



自定义抽取模型

若预置模型中未包含您想要的模型，可通过自定义抽取模型，快速扩展可识别的文件类型：只需上传一张样本图片、配置待识别字段，无需训练即可精准抽取字段值。

- **基础字段**：指单个字段，例如单位名称、住所；

- **组合字段**：指一组存在关联关系的字段，例如项目的详细信息，包含项目名称、子项目、项目参数。更直观的组名有利于模型抽取效果。



产品优势

- **开箱即用**：MoE 混合专家方案驱动，预置丰富的 OCR 模型及文心大模型3.5，无需定制训练、无需分类后再调用，一个接口全识别，分钟级完成业务上线。
- **积木式组合**：可按照实际业务场景灵活配置所需模型，依托意图识别模型自动分至对应专家模型，高效处理、提高资源利用率。
- **适应复杂场景**：基于百度沉淀多年的 OCR 模型，以及文心大模型对海量文本的深度学习，准确识别文字并理解上下文信息，精准完成结构化识别。
- **使用方式灵活**：支持 SaaS 平台在线使用、公有云 API 调用以及私有化本地部署，满足企业各场景使用需求。

API文档

申请试用

该接口正在邀测中，**请您先提交 合作咨询 或 提交工单**，提供公司名称、appid、应用场景等信息，工作人员协助开通权限后方可使用。

接口描述

iOCR 全场景识别采用 MoE 混合专家架构，**无需分类调用、无需定制训练，一个接口实现全场景结构化识别**。基于意图识别模型，自动将同一任务中的不同类型文件分发至对应专家模型，已预置 10 余种 OCR 识别模型与通用抽取大模型，同时支持自定义文档抽取模型，不限类型与版式，轻松提取复杂文档关键字段。

登录 iOCR 全场景识别平台，可在示例项目空间快速体验，也可创建您专属的项目空间与抽取模型，平台和接口服务的额度共享互通。

本服务为异步接口，包含提交请求、获取结果两个子接口，详情如下：

- **iOCR 全场景识别-提交请求**：传入图片/PDF、项目空间ID等参数，创建识别任务并获取任务ID，该接口支持2QPS；
- **iOCR 全场景识别-获取结果**：成功创建任务后，传入任务ID获取识别结果，该接口支持10QPS。可调用获取结果接口进行结果轮询，建议提交请求后30秒开始轮询。

iOCR 全场景识别-提交请求

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise/universal/request`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
file_name	是	string	-	文件名称，例如：通过pdf_file传入一份名为test的文件，该字段需传入test.pdf。 **注意：** 文件名称后缀需与文件实际类型一致
image	和url/pdf_file/ofd_file四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px，支持jpg/jpeg/png/bmp格式 **优先级：** image>url>pdf_file>ofd_file，当image字段存在时，url、pdf_file、ofd_file字段失效
url	和image/pdf_file/ofd_file 四选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px **优先级：** image>url>pdf_file>ofd_file，当image字段存在时，url字段失效 **注意：** 请关闭URL防盗链
pdf_file	和image/url/ofd_file四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px。若PDF存在多页，**默认识别全部页** **优先级：** image>url>pdf_file>ofd_file，当image、url字段存在时，pdf_file字段失效
ofd_file	和image/url/pdf_file四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过10M，最短边至少15px，最长边最大8192px。若OFD存在多页，**默认识别全部页** **优先级：** image>url>pdf_file>ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
templateSign	和 classifierId 二选一	string	-	文件类型 ID，自定义模型或预置模型的唯一标示，可用于调用指定的识别模型进行结构化识别。 **自定义模型 ID：** 可在「模型管理」页查看并复制使用 **预置模型 ID：** 可在「预置模型清单」查看并复制使用
classifierId	和 templateSign 二选一	string	-	项目空间 ID，项目空间识别规则的唯一标示，可用于调用指定项目空间的识别规则，对传入的图片自动分类及识别，可在「项目空间」查看并复制使用。 **优先级：** templateSign>classfierId，当templateSign字段存在时，classfierId字段失效

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、文件地址等信息

提示二：目前仅提供Python语言

```
 import base64
import requests
import os
'''
iOCR 全场景识别-提交请求
'''
file_path = '[本地文件]'
request_host = "https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise/universal/request"
encoded_string = ''
with open(file_path, 'rb') as file:
    file_base64 = base64.b64encode(file.read()).decode('utf-8')
##### 优先级：templateSign>classifierId，当templateSign字段存在时，classifierId字段失效
data = {
    'image': file_base64,
    'fileName': os.path.basename(file_path),
    "classifierId":"project-yht3ia44fyixcwm2",
    "templateSign":"id_card_font"

}
access_token = '[调用鉴权接口获取的token]'
request_url = request_host + "?access_token=" + access_token
headers = {'Content-Type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, headers=headers, data=data)
if response:
    print(response.json())
```

返回说明

返回参数

字段	类型	说明
log_id	string	唯一的 log_id，用于问题定位
error_code	int	错误码
error_msg	string	错误码描述信息
error_explain	string	错误码详细中文描述
result	dict	返回的结果列表
+ task_id	string	该请求生成的task_id，可使用该task_id获取识别结果

返回示例

成功返回示例

```
 {
    "error_code": 0,
    "error_msg": "",
    "log_id": "3599798266",
    "task_result": {
        "task_id": "task-2m8g3dhqmrv9cxfe"
    }
}
```

失败返回示例（错误码说明详见API文档-错误码）：

```
{
  "error_code": 283016,
  "error_msg": "parameters value error",
  "error_explain": "classifier_id和template_sign不能同时为空",
  "log_id": "3599798268",
  "data": null
}
```

iOCR 全场景识别-获取结果

请求说明

请求示例

HTTP 方法：`POST` 请求URL：`https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise/universal/get_result`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
task_id	是	string	发送提交请求时返回的 task_id

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、task_id信息

提示二：目前仅提供Python语言

```
import requests
'''
iOCR 全场景识别-获取结果
'''

request_host = "https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise/universal/get_result"
task_id = "task-xxxx"
access_token = "[调用鉴权接口获取的token]"
request_url = request_host + "?access_token=" + access_token
data={
  "task_id":"task-111"
}
headers = {'Content-Type': 'application/x-www-form-urlencoded'}
response = requests.post(request_url, data=data, headers=headers)
if response:
    print(response.json())
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
task_result	dict	任务详情列表
+ task_id	string	任务id
+ status	string	任务状态，Success：成功；Running：处理中；Failed：失败
+ created	int	创建时间，格式为：YYYY-MM-DD nn:nn:nn（UTC/GMT+08:00 东八区）
+ started	int	开始时间，格式为：YYYY-MM-DD nn:nn:nn（UTC/GMT+08:00 东八区）
+ finished	int	完成时间，格式为：YYYY-MM-DD nn:nn:nn（UTC/GMT+08:00 东八区）
+ duration	int	处理时长，单位为秒（s）
words_result_num	int	识别结果数量，表示word_result结果数量
words_result	list	识别结果列表
+ doc_type	string	文件类型
+ page_num	int	页码信息，页码从1开始
+ doc_location	dict	文件位置信息
++ box	[4][2]int	四角点坐标框，[[x1,y1],[x2,y2],[,x3,y3],[x4,y4]]，分别对应坐标框四个点的坐标
++ cbox	array	坐标框，「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点）
+ kv_result	dict	结构化识别结果，包含抽取字段和结果信息
++ single_key	dict	单个字段抽取结果
+++ {key}	[]dict	单个字段名称
++++ words	string	单个字段抽取结果
++++ location	[]dict	抽取内容信息
+++++ box	[4][2]int	四角点坐标框，[[x1,y1],[x2,y2],[,x3,y3],[x4,y4]]，分别对应坐标框四个点的坐标
+++++ cbox	[4]int	坐标框，「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点）
++ combo_key	dict	组合字段抽取结果
+++ {root_key}	[]dict	父字段名称
++++ {leaf_key}	[]dict	子字段名称
+++++ words	string	子字段抽取结果
+++++ location	[]dict	抽取内容信息
++++++ box	[4][2]int	四角点坐标框，[[x1,y1],[x2,y2],[,x3,y3],[x4,y4]]，分别对应坐标框四个点的坐标
++++++ cbox	[4]int	坐标框，「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点）
++ full_result	dict	全文识别结果，仅在doc_type为"其他"，且识别规则兜底策略选择“全文识别”时返回
++ other_result	dict	其他识别结果，如卡证质量、风险检测结果

返回示例

识别成功示例

```
{
```

```
"error_code": 0,
"error_msg": "",
"log_id": "2761737701",
"task_result": {
    "task_id": "task-2m8g3dhqmrv9cxfe",
    "status": "Success",
    "created": "2025-03-17 15:37:55",
    "started": "2025-03-17 15:37:55",
    "finished": "2025-03-17 15:38:02",
    "duration": 7
},
"words_result_num": 1,
"words_result": [
    {
        "page_num": 1,
        "doc_type": "其他",
        "doc_location": null,
        "kv_result": {
            "single_key": {
                "价税合计（大写）": [
                    {
                        "words": "叁佰陆拾圆整",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    }
                ],
                "价税合计（小写）": [
                    {
                        "words": "￥360.00",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    }
                ],
                "发票号码": [
                    {
                        "words": "3321192130",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    },
                    {
                        "words": "No 07285251",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    }
                ],
                "名称": [
                    {
                        "words": "百度在线网络技术（北京）有限公司",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    },
                    {
                        "words": "阿里云计算有限公司"
```

```
            "words": "阿里云计算有限公司",
            "location": {
                "box": null,
                "cbox": null
            }
        }
    ],
    "地址、电话": [
        {
            "words": "北京市海淀区上地十街10号百度大厦三层",
            "location": {
                "box": null,
                "cbox": null
            }
        },
        {
            "words": "杭州市转塘科技经济区块16号8幢 0571-85022088",
            "location": {
                "box": null,
                "cbox": null
            }
        }
    ],
    "开户行及账号": [
        {
            "words": "招商银行北京分行大屯路支行866180100210002",
            "location": {
                "box": null,
                "cbox": null
            }
        },
        {
            "words": "招商银行杭州高新支行502905023610702",
            "location": {
                "box": null,
                "cbox": null
            }
        }
    ],
    "开票日期": [
        {
            "words": "2019年08月28日",
            "location": {
                "box": null,
                "cbox": null
            }
        }
    ],
    "纳税人识别号": [
        {
            "words": "91110911717743469K",
            "location": {
                "box": null,
                "cbox": null
            }
        },
        {
            "words": "91330106673959654P",
            "location": {
                "box": null,
                "cbox": null
            }
        }
```

```
            ]
        },
        "combo_key": {
            "货物或应税劳务、服务信息": [
                {
                    "单价": {
                        "words": "339. 62",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    },
                    "单位": {
                        "words": "",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    },
                    "数量": {
                        "words": "1",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    },
                    "税率": {
                        "words": "",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    },
                    "税额": {
                        "words": "20. 38",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    },
                    "规格型号": {
                        "words": "",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    },
                    "货物或应税劳务、服务名称": {
                        "words": "*信息技术服务*软件服务费",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    },
                    "金额": {
                        "words": "339. 62",
                        "location": {
                            "box": null,
                            "cbox": null
                        }
                    }
                }
```

```
        ],
        "购买方扣税凭证信息": [
          {
            "单价": {
              "words": "2.30",
              "location": {
                "box": null,
                "cbox": null
              }
            },
            "合计": {
              "words": "￥339.62",
              "location": {
                "box": null,
                "cbox": null
              }
            },
            "时间": {
              "words": "18:50-17:06",
              "location": {
                "box": null,
                "cbox": null
              }
            },
            "油附加费": {
              "words": "￥1.00",
              "location": {
                "box": null,
                "cbox": null
              }
            },
            "税额": {
              "words": "Y20.38",
              "location": {
                "box": null,
                "cbox": null
              }
            },
            "证号": {
              "words": "244926",
              "location": {
                "box": null,
                "cbox": null
              }
            },
            "车号": {
              "words": "京B.26353",
              "location": {
                "box": null,
                "cbox": null
              }
            },
            "里程": {
              "words": "6.0",
              "location": {
                "box": null,
                "cbox": null
              }
            }
          }
        ]
      }
```

```
        }
      }
    ]
}
```

任务运行中示例：

```
{
  "error_code": 0,
  "error_msg": "",
  "log_id": "3739485273",
  "task_result": {
    "task_id": "task-q0zaegy82ujqe6qc",
    "status": "Running",
    "reason": "",
    "created": "2025-03-17 17:29:38",
    "started": "2025-03-17 17:29:38",
    "finished": "",
    "duration": 5
  },
  "words_result": null
}
```

预置模型清单

序号	文件类型	文件类型 ID（通过templateSign传入）	默认请求参数	备注
1	身份证正面	id_card_front	id_card_side = front detect_ps = true detect_risk = true detect_quality = true detect_photo = true detect_card = false detect_direction = true	即人像面，返参含义详见 API文档-身份证识别
2	身份证反面	id_card_back	id_card_side = back detect_ps = true detect_risk = true detect_quality = true detect_photo = true detect_card = false detect_direction = true	即国徽面，返参含义详见 API文档-身份证识别
3	行驶证正页	vehicle_license	detect_direction = true vehicle_license_side = front unified = false quality_warn = true risk_warn = true	返参含义详见 API文档-行驶证识别
4	行驶证副页	vehicle_license_back	detect_direction = true vehicle_license_side = back unified = false quality_warn = true risk_warn = true	返参含义详见 API文档-行驶证识别
			detect_direction = true	

5	驾驶证正页	driver_license	driving_license_side = front unified_valid_period = false quality_warn = true risk_warn = true	支持电子驾驶证正页，返参含义详见 API文档-驾驶证识别
6	驾驶证副页	driver_license_back	detect_direction = true driving_license_side = back unified_valid_period = false quality_warn = true risk_warn = true	返参含义详见 API文档-驾驶证识别
7	银行卡	bank_card	location = true detect_quality = true	返参含义详见 API文档-银行卡识别
8	大陆护照	chinese_passport	/	返参含义详见 API文档-护照识别
9	户口本登记页	household_register_subpage	household_register_side = subpage	返参含义详见 API文档-户口本识别
10	户口本户主页	household_register_homepage	household_register_side = homepage	返参含义详见 API文档-户口本识别
11	社保卡	social_security_card	/	返参含义详见 API文档-社保卡识别
12	房产证	real_estate_certificate	probability = true location = true	返参含义详见 API文档-房产证识别
13	财务票据	multiple_invoice	probability = true location = true verify_parameter = false	支持增值税发票、卷票、机打发票等13类票据，返参含义详见 API文档-智能财务票据识别
14	营业执照	biz_license	risk_warn= true detect_quality = true	返参含义详见 API文档-营业执照
15	港澳居民来往内地通行证正面	hk_mc_return_passport_front	exitentrypermit_type = hk_mc_return_passport_front probability = true location = true	即人像面，返参含义详见 API文档-港澳台证件识别
16	港澳居民来往内地通行证反面	hk_mc_return_passport_back	exitentrypermit_type = hk_mc_return_passport_back probability = true location = true	返参含义详见 API文档-港澳台证件识别
17	港澳通行证正面	hk_mc_passport_front	exitentrypermit_type = hk_mc_passport_front probability = true location = true	即人像面，返参含义详见 API文档-港澳台证件识别
18	港澳通行证反面	hk_mc_passport_back	exitentrypermit_type = hk_mc_passport_back probability = true location = true	返参含义详见 API文档-港澳台证件识别
19	台湾居民来往大陆通行证正面	tw_return_passport_front	exitentrypermit_type = tw_return_passport_front probability = true	即人像面，返参含义详见 API文档-港澳台证件识别

			location = true	
20	台湾居民来往大陆通行证反面	tw_return_passport_back	exitentrypermit_type = tw_return_passport_back probability = true location = true	返参含义详见 API文档-港澳台证件识别
21	台湾通行证正面	tw_passport_front	exitentrypermit_type = tw_passport_front probability = true location = true	即人像面，返参含义详见 API文档-港澳台证件识别
22	台湾通行证反面	tw_passport_back	exitentrypermit_type = tw_passport_back probability = true location = true	返参含义详见 API文档-港澳台证件识别
23	车辆合格证	vehicle_certificate	/	返参详见 API文档-车辆合格证识别
24	快递面单	waybill	is_identify_virtual_waybill = true	返参详见 API文档-快递面单识别
25	车牌	license_plate	multi_detect = true multi_scale = false detect_complete = true detect_risk = true	返参详见 API文档-车牌识别

iOCR通用版

简介

Hi，欢迎您使用百度 iOCR 通用版。

iOCR 通用版是 iOCR 自定义模板文字识别针对通用场景下固定版式的卡证票据、文件资料提供的一款 OCR 定制化产品，您仅需上传一张模板图片，即可通过框选参照字段及识别区快速制作结构化识别模型；同时，还可针对制作的多个模板训练自定义分类器，一步完成图片的自动分类和结构化识别。

iOCR 通用版提供三大自定义功能：

- **自定义模板：** 针对需要识别的图片版式，仅需上传一张模板图片，即可通过框选参照字段和识别区自助制作一个识别模板，并建立图片中文字的 Key-Value 对应关系，实现对相同版式图片的结构化识别
- **自定义分类器：** 针对已发布的多个识别模板，只需每类上传 30 张训练集图片或填写分类关键词即可创建分类器，实现对不同版式图片的自动分类，省去人工分类成本，一步实现图片的自动分类和结构化识别
- **自定义字段类型：** 针对输出值为有限集的字段，用户可将可能的输出值汇总为字段词典进行上传，在框选识别区后选择该字段类型，系统则会对识别结果进行智能匹配或纠正，用于规范识别结果，并提高识别准确率

以上三大自定义功能可结合使用实现对不同版式图片进行自动分类并结构化识别的需求，分类准确率可达 99% 以上，识别准确率可达 95% 以上。如需了解更多，可访问 iOCR 通用版产品介绍页

名词解释

- **固定版式：** 指不同图片中的对应字段文字虽内容不同、长短不同，但可能出现的位置及范围固定不变的一类图片，如身份证、学生证、结婚证、火车票等；若图片中存在前文内容长度影响后文位置、表格行列不固定等情况的需针对不同情况制作多个模板
- **参照字段：** 相同版式的不同图片中 **位置和内容固定不变的字段**，可框选做为图片的锚点，用做对后续传入的图片进行模板匹配和矫正；建议选取 8 个以上、分散在图片四周、且不重复无换行的参照字段，效果最佳

- **识别区：** 图片中需要进行识别的字段，可通过框选及命名构建 「字段名称：识别区内容」 的 Key：Value 对应关系，用于对后续传入的相同版式图片的相同位置内容进行结构化识别；识别区的框选直接影响最终识别效果，框选区域需完全覆盖文字可能出现的范围
- **字段类型：** 针对不同识别区内容类型进行专项优化的切片识别模型，如小写金额、日期、纯数字等，可根据需求选择合适的字段类型以提升识别准确率也可通过穷举可能的输出值范围，自定义字段类型，对识别结果进行智能纠正和规范
- **训练集图片：** 针对已发布的识别模板上传的相同版式的图片集合，用于提取该版式图片的视觉特征，对后续上传的图片进行自动分类。为了最佳的分类效果，训练集至少包含 30 张以上不重复的、版式相同的图片
- **分类关键词：**图片中存在的独有的文字内容，用作模板分类的文字依据，需保证填写的关键词在该版式图片中均有出现，可根据填写的关键词唯一确定图片所属模板类别
- **templateSign：** 模板ID，用于指定后期上传的图片用哪个模板来进行识别
- **classifierId：** 分类器ID，用于指定使用具体某个分类器，传入本参数后不用再传templateSign参数

预置能力介绍

预置模板

- **大陆身份证正面：**支持对中国大陆二代居民身份证人像面所有 6 个字段进行结构化识别，包括姓名、性别、民族、出生日期、住址、身份证号
- **大陆身份证背面：**支持对中国大陆二代居民身份证国徽面的签发日期、失效日期、签发机关 3 个字段进行结构化识别
- **行驶证正页：**支持对机动车行驶证正页所有 10 个字段进行结构化识别，包括号牌号码、车辆类型、所有人、住址、使用性质、品牌型号、车辆识别代号、发动机号码、注册日期、发证日期
- **行驶证副页：**支持对机动车行驶证副页所有 11 个字段进行结构化识别，包括号牌号码、档案编号、核定载人数、总质量、整备质量、核定载质量、外廓尺寸、准牵引总质量、备注、检验记录、燃油类型
- **驾驶证：**支持对机动车驾驶证正本所有 9 个字段进行结构化识别，包括证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效起始日期、失效日期
- **银行卡：**支持对国内主流银行卡的卡号、有效期、发卡行、卡片类型 4 个关键字段进行结构化识别
- **营业执照（可选）：**支持对不同版式营业执照所有 11 个字段进行结构化识别，包括证件编号、社会信用代码、单位名称、地址、法人、类型、组成形式、注册资本、成立日期、有效日期、经营范围
- **大陆护照（可选）：**支持对中国大陆护照个人资料页所有 11 个字段进行结构化识别，包括国家码、护照号、姓名、姓名拼音、性别、出生地点、出生日期、签发地点、签发日期、有效期、签发机关
- **港澳通行证（可选）：**支持对大陆居民往来港澳通行证的 7 个关键字段进行结构化识别，包括证件号码、姓名、姓名拼音、出生日期、性别、有效期限、签发地点
- **台湾通行证（可选）：**支持对大陆居民往来台湾通行证的 7 个关键字段进行结构化识别，包括证件号码、姓名、姓名拼音、出生日期、性别、有效期限、签发地点
- **车辆合格证（可选）：**支持对车辆合格证的 23 个关键字段进行结构化识别，包括合格证编号、发证日期及制造企业名、品牌、名称、型号等车辆信息
- **增值税发票（可选）：**支持对增值税普票或专票所有 31 个字段进行结构化识别，包括发票基本信息、销售方及购买方信息、商品信息、价税信息等，其中四要素识别准确率超过99.9%
- **增值税卷票（可选）：**支持对增值税卷票的 16 个关键字段进行识别，包括发票类型、发票代码、发票号码、机打号

码、机器编号、销售方纳税人识别号、开票日期、购买方纳税人识别号、项目、单价、数量、金额、税额、合计金额(小写)、合计金额(大写)、校验码

- **定额发票（可选）**：支持对各类定额发票的发票代码、发票号码、金额 3 个关键字段进行结构化识别
- **通用机打发票（可选）**：支持对国家/地方税务局发行的横/竖版通用机打发票的 6 个关键字段进行结构化识别，包括发票类型、发票号码、发票代码、开票日期、商品名称、合计金额
- **火车票（可选）**：支持对红、蓝火车票的 8 个关键字段进行结构化识别，包括车票号码、始发站、目的站、车次、日期、票价、席别、姓名
- **出租车票（可选）**：支持识别全国各大城市出租车票的 6 个关键字段，包括发票号码、代码、车号、日期、时间、金额
- **行程单（可选）**：支持对飞机行程单的 12 个关键字段进行结构化识别，包括印刷序号、姓名、始发站、目的站、航班号、日期、票价、民航发展基金、燃油附加费、其他税费、合计金额、填开日期

> **注意：**
>
> - 预置模板默认已发布，可直接复制对应的 templateSign 使用API进行调用
> - 备注「可选」字样的预置模板未默认展示在「模板管理 - 预置模板」标签页，如需使用需点击页面右上角的「添加更多」按钮按需进行勾选添加

联系我们

本文档主要说明如何使用 iOCR 自定义模板文字识别完成文字识别模型及分类器的制作，以及发布后如何通过 API 进行调用，如文档内容无法解决您的问题，可以通过以下方式寻求帮助：

- 在百度云控制台内 提交工单 联系专业的技术支持人员进行解答
- 进入 OCR 论坛 发布您的问题参与互动
- 加入百度 iOCR 交流群（群号:570832882）与更多开发者进行交流

使用流程

使用说明

本文档主要说明如何在 iOCR 通用版的可视化界面中完成模板、分类器及字段类型的自定义，同时对模板图片预处理、框选表格识别区等高级功能进行说明。视频教程请参见 iOCR通用版使用教程。

自定义模板

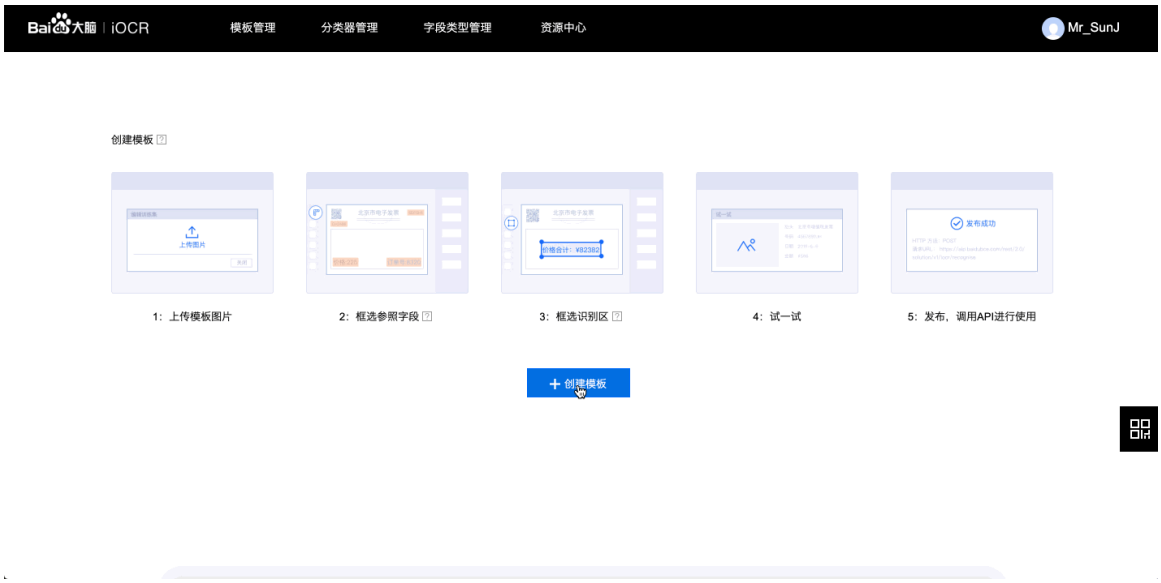
创建自定义模板的基本流程如下图所示，仅需上传一张规范的模板图片，通过简单框选，5 分钟即可完成结构化识别模板的制作。



1：上传模板图片　2：框选参照字段　3：框选识别区　4：试一试　5：发布，调用API进行使用

Step 1: 上传模板图片

在 iOCR通用版 - 模板管理页面 点击「自定义模板」标签页下方 「**创建模板**」 按钮，在弹出框中上传一张 **字迹清晰且摆放端正** 的模板图片（大小不超过4M，最长边不超过4096像素），并对模板进行命名。



Step 2: 框选参照字段

进入模板编辑页面，右侧操作步骤中选择 「**第1步：框选参照字段**」 标签，左侧工具栏选择 「**框选参照字段**」 按钮，使用鼠标在模板图片中框选位置和内容都固定不变的文字，如下图所示 **橘色矩形** 框选区域。



> **注意：** 「参照字段」为相同版式的不同图片中位置和内容固定不变的字段，可做为图片的锚点，用做对后续传入的图片进行模板匹配和矫正

> **框选Tips：**
>
> 1. 参照字段个数需保证在4个以上（推荐8个以上），并尽量分散在四角
> 2. 单个参照字段不可跨行，推荐字数在4个以内
> 3. 参照字段文字内容在上下文中不会重复出现
> 4. 仅支持框选中英文、数字，不可包含符号、图案

Step 3: 框选识别区

右侧操作步骤中点击 「**第2步：框选识别区**」 标签，左侧工具栏选择 「**框选识别区**」 按钮，使用鼠标在模板图片上框选业务场景需要进行识别的字段，如下图所示 **蓝色矩形** 框选区域；同时，填写 「**字段名称**」，并选择合适的 「**字段类型**」 以提高识别准确率。



注意：「识别区」为图片中需要进行识别的字段，可通过框选及命名构建 「字段名称：识别区内容」 的 Key：Value 对应关系，用于对后续传入的相同版式图片的相同位置内容进行结构化识别

框选Tips：

1. 尽量扩大识别区框选范围，保证后续传入图片的对应字段内容可被完全覆盖，但同时也需保证不框选到其他字段内容
2. 选择合适的字段类型有助于提升字段识别效果
3. 如需识别图片内列宽固定的表格，可点击工具栏中「插入表格」按钮框选表格识别区，可参考 表格识别区使用说明

Step 4: 试一试

参照字段和识别区全部框选完毕后，可点击页面右上角的 「**试一试**」 按钮进行识别效果测试，在弹出框中上传任意一张相同版式的图片即可，如下图所示。可点击图片下方「更换图片」按钮更换测试图片，如多次测试效果满意即可进行发布；如效果不满意可返回继续编辑。



注意：

1. 如试一试结果出现图片无法匹配模板的情况，需确认上传的测试图片与模板图片是否为同一版式，如确认无误可调节参照字段框选范围或更换参照字段，以提升模板匹配准确率
2. 如试一试结果中出现识别结果错误的情况，可调整识别区框选范围或更换识别区字段类型，以提升识别准确率

Step 5: 发布模板，调用API进行使用

如测试效果满意，可点击试一试弹出框右下角的 「**立即发布**」 按钮或模板编辑页面右上角的 「**发布**」 按钮进行发布，发布成功后即可通过模板ID调用该模板，调用方式可查看 API文档。



> **注意：** 只有发布后的模板才能通过线上接口进行调用，如果编辑未发布，那么仅仅是生成了一个新的版本，此时对模板的任何修改都不会影响线上调用。

自定义分类器

创建自定义分类器的基本流程如下图所示，仅需针对每个模板上传 30 张相同版式的训练集图片或填写分类关键词，即可自动训练一个能够对所选模板进行自动分类的分类器。

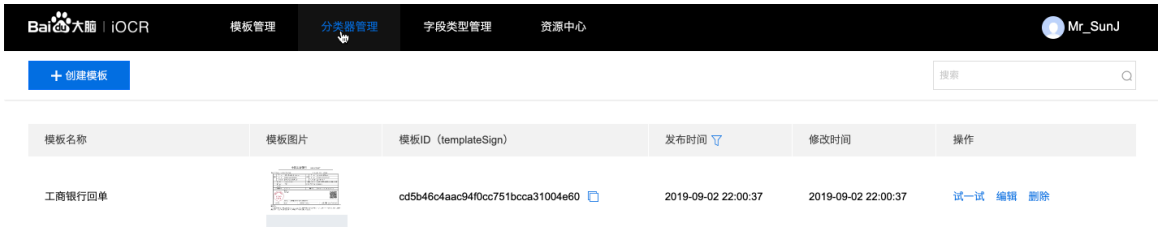


1：选择模板　2：上传训练集　3：训练　4：测试　5：发布

Step 1: 选择模板

在 iOCR通用版 - 分类器管理页面 点击下方 「**创建分类器**」 按钮，在弹出框中填写「分类器名称」及「功能描述」，填写完毕点击 「**确认**」 即可进入分类器编辑页面。

在分类器编辑页面中点击 「**添加预置模板**」 或 「**添加自定义模板**」，选择 **已发布的** 自定义模板或系统预置模板加入训练队列。



> **说明：**
>
> **预置模板：** 系统预置的常用卡证、票据模板，已上传100张训练集图片并已填写分类关键词信息，可直接勾选用于对应图片的自动分类和结构化识别
>
> **自定义模板：** 已发布的自定义模板

Step 2: 上传训练集/填写分类关键词

点击选中的自定义模板所在行右侧的 「**编辑训练集**」 按钮，在弹出框中上传一个包含 30 张以上相同版式图片、不超过 200MB 的 ZIP 格式压缩包。



> **注意：** 「图片训练集」为针对已发布的识别模板上传的相同版式的图片集合，用于训练分类器对后续上传的图片进行自动分类；为了最佳的分类效果，训练集至少包含 30 张以上不重复的、版式相同的图片；系统模版已预置 100 张训练集图片，无需上传

也可在「**分类关键词**」列填写对应模板图片中的关键文字内容，需保证填写的关键词在该版式图片中均有出现，且具有特异性，可根据填写的关键词唯一确定图片所属模板类别

如身份证人像面的“姓名”、“性别”、“民族”、“出生”、“公民身份号码”五个字段在每一张身份证人像面均会出现，且 5 个字段全部出现时基本可确定该图片为身份证人像面，则此 5 个字段即可作为身份证人像面的分类关键词

> **注意：**每个模板最多可设置 5 个分类关键词，每个关键词最多包含 15 个中英文字符，不可包含其他特殊符号，关键词之间使用（中/英文）逗号进行间隔

Step 3: 训练

参与分类的模板及其训练集图片数量或分类关键词符合标准后，即可点击页面左下角的 「**开始训练**」 按钮进行训练，等待训练完毕后即可在分类器列表中进行查看，训练完成后的分类器有以下三种分类形式：

- **纯图片特征分类器：**如所有模板各上传了 30 张以上相同版式训练集图片，但均未填写分类关键词，则训练出的分类器将完全依据图片特征进行分类；
- **纯文字特征分类器：**如所有模板均填写了分类关键词，但部分或全部模板未上传训练集图片，则训练出的分类器将完全依据关键词的文字信息进行分类；
- **图文协同分类器：**如所有模板各上传了 30 张以上相同版式训练集图片，且部分或全部模板填写了分类关键词，则训练出的分类器将先进行关键词匹配，如匹配到相同关键词则再根据图片特征进行区分。



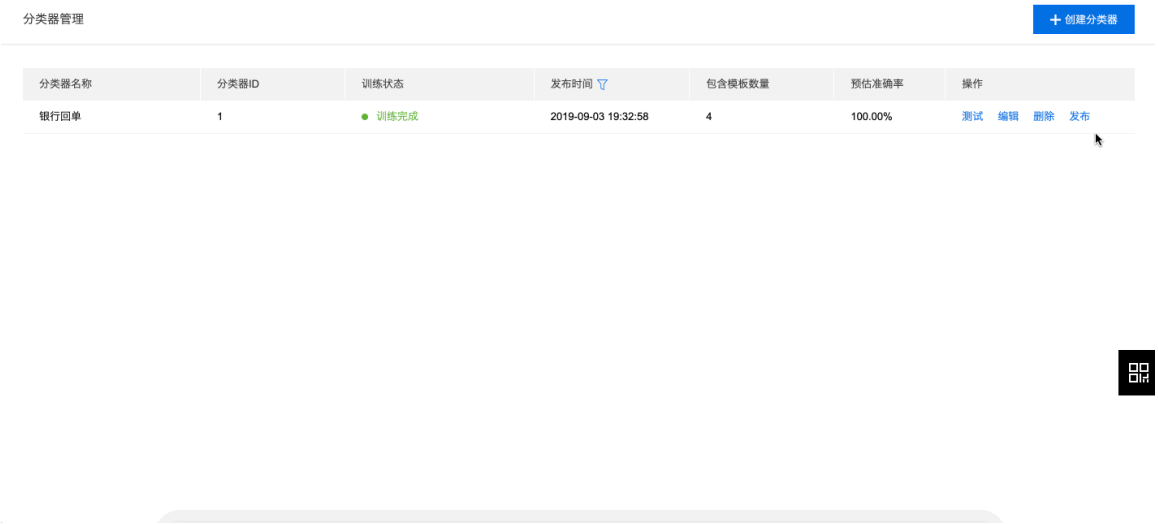
Step 4: 测试

在分类器管理页面的分类器列表中点击训练完毕的分类器右侧的 「**测试**」 按钮进行效果测试，在弹出框中上传任意一张图片即可，如下图所示，测试结果中包含分类结果及结构化识别结果。可点击图片下方「更换图片」按钮更换测试图片，如多次测试效果满意即可进行发布；如效果不满意可返回继续编辑。



Step 5: 发布

如测试效果满意，可点击测试弹出框右下角的 「**立即发布**」 按钮或分类器列表右侧的 「**发布**」 按钮进行发布，发布成功后即可通过分类器ID调用该分类器，实现图片的自动分类及结构化识别，调用方式可查看 API文档。



自定义字段类型

创建自定义字段类型的基本流程如下图所示，仅需输入全部可能出现的输出值，即可创建一个自定义字段类型在制作模板时进行使用，用于字段识别结果的规范和智能纠正。



1：编辑字段词典 2：开始使用

Step 1: 编辑字段词典

在 iOCR通用版 - 字段类型管理页面 点击下方 「**创建字段类型**」 按钮，在弹出框中填写「字段类型名称」及「词典内容」，在词典内容中列举全部可能出现的输出值内容，填写完毕点击 「**立即创建**」 即可完成创建；也可点击 「**添加预置类型**」 按钮直接添加预置的常用字段类型，无需自行编辑词典内容。



Step 2: 开始使用

创建成功的字段类型即可在制作自定义模板选择字段类型时，点击 「**我的字段类型**」 进行勾选使用，使用后即可对该字段的识别结果进行规范或智能纠正。



高级功能

模板图片预处理

为保证模板的识别效果，建议上传的模板图片 **字迹清晰且摆放端正**，以保证后续传入的同版式图片能被匹配、矫正并准确识别。

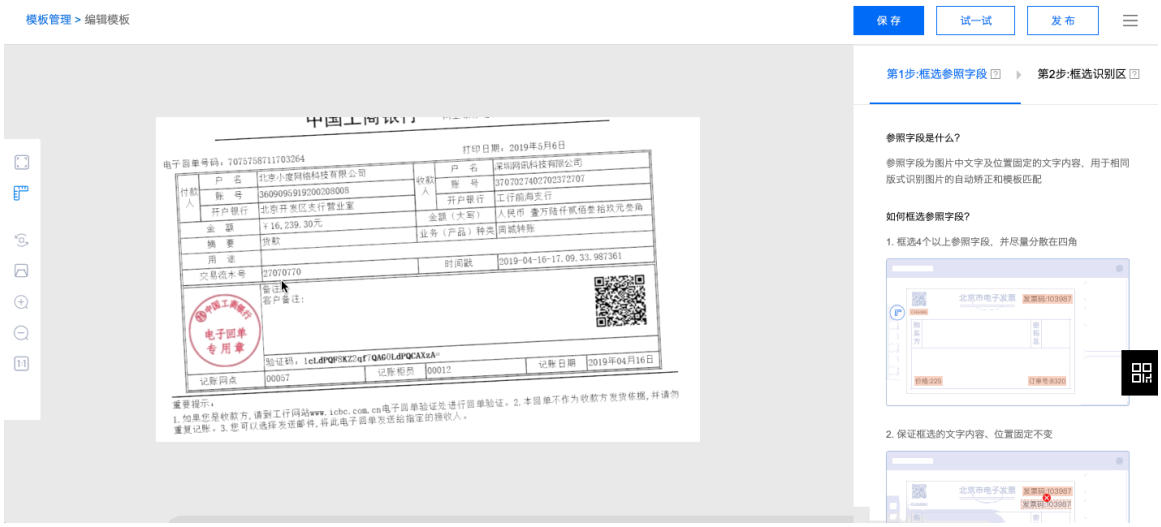
如您无法保证上传的模板图片摆放端正，可 **在框选参照字段/识别区前** 通过我们提供的模板图片预处理功能对模板图片进行 **旋转/透视裁剪** 操作纠正倾斜图片。

图片旋转

点击编辑区左侧工具栏中的 「**图片旋转**」 按钮，即可进入旋转处理页面。

在旋转处理页面，使用鼠标按住上方旋转按钮拖动图片顺时针/逆时针旋转，旋转效果满意后松开鼠标点击 「**保存**」 按钮即

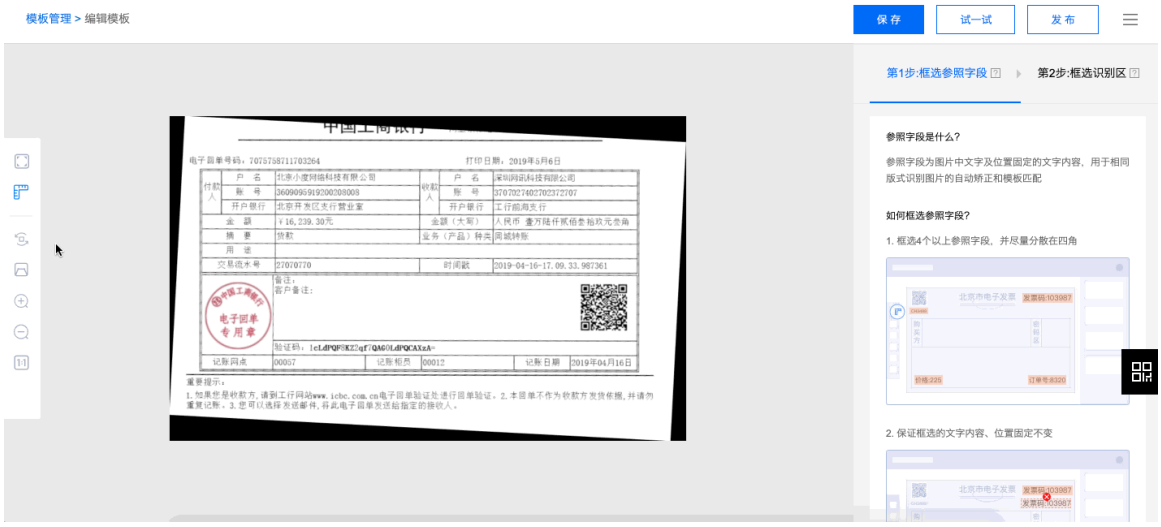
可。



图片透视裁剪

点击编辑区左侧工具栏中的 「**透视裁剪**」 按钮，即可进入裁剪处理页面。

在裁剪处理页面，使用鼠标按住四角点小圆点进行拖拽，使四边紧贴卡证或票据边缘，拖拽完毕后点击 「**保存**」 按钮即可。



框选表格识别区

如果要识别的图片中存在行列固定的表格，可点击工具栏中的 「**插入表格**」 按钮添加指定列数的表格识别区，通过拖拽表格四角及列分隔线的小白点使识别区覆盖表格区域，并使各列识别区与表格各列对应，即可对表格内容进行结构化识别。

> **注意：** 表格识别区仅需覆盖表格内容，无需将表头内容进行框选



如您在操作过程中出现上述内容未说明的问题，可参考 iOCR常见问题，或在OCR论坛发布您的问题，也可加入百度iOCR交流群（群号:570832882）与更多开发者进行交流。

API文档

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（data:image/jpeg;base64，） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image/pdf_file 三选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，需去掉编码头（data:application/pdf;base64，） 要求base64编码和urlencode后大小不超过4M **注：**目前仅支持单页PDF识别，如上传的为多页PDF，仅识别第一页
templateSign	和classifierId二选一	string	-	模板 ID，自定义模板或预置模板的唯一标示，可用于调用指定的识别模板进行结构化识别，可在「模板管理」页查看并复制使用
classifierId	和templateSign 二选一	string	-	分类器Id，分类器的唯一标示，可用于调用指定的分类器对传入的图片进行自动分类及识别 **与 templateSign 至少存在一个，如同时存在，则优先级 templateSign > classfierId**

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

Objective-C

```
##### 请求模板id
curl -i -k 'https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise?access_token=【调用鉴权接口获取的token】' --data 'templateSign=xxx&image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
##### 请求分类器id
curl -i -k 'https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise?access_token=【调用鉴权接口获取的token】' --data 'classifierId=xxx&image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
logid	是	uint64	唯一的log id，用于问题定位
error_code	是	int	0代表成功，如果有错误码返回可以参考错误码列表排查问题
error_msg	是	string	如果error_code具体的失败信息，可以参考下方错误码列表排查问题
data	是	jsonObject	识别返回的结果
+ isStructured	是	string	表示是否结构化成功，true为成功，false为失败；成功时，返回结构化的识别结果；失败时，如果能识别，按行返回识别结果，如果不能识别，返回空
+ templateSign	否	string	图片分类结果对应的模板id或指定使用的模版id
+ scores	否	float	分类置信度，如果指定templateSign，则该值为1
+ ret	否	jsonArray	识别出来的字段数组，每一个单元里包含以下几个元素
++ word_name	否	string	isStructured 为 true 时存在，表示字段的名字；如果 isStructured 为 false 时，不存在
++ word	否	string	识别的字符串或单字
++ location	否	jsonObject	字段在原图上对应的矩形框
++ probability	否	jsonObject	字段的置信度，包括平均、最小和方差

返回示例

```
{
  "data": {
    "ret": [
      {
        "probability": {
          "average": 0.998482,
          "min": 0.9957,
          "variance": 0.000002
        },
        "location": {
          "height": 88,
          "left": 1202,
          "top": 437,
          "width": 267
        },
        "word_name": "终点站",
        "word": "天津"
      },
      {
        "probability": {
          "average": 0.994316,
          "min": 0.629856,
          "variance": 0.000281
        },
        "location": {
          "height": 82,
          "left": 359,
          "top": 593,
          "width": 660
        },
        "word_name": "发车时间",
        "word": "201706092107"
      },
      {
        "probability": {
          "average": 0.998482,
          "min": 0.9957,
          "variance": 0.000002
        },
        "location": {
          "height": 90,
          "left": 432,
          "top": 432,
          "width": 261
        },
        "word_name": "始发站",
        "word": "北京南"
      },
      {
        "probability": {
          "average": 0.952242,
          "min": 0.77037,
          "variance": 0.008272
        },
        "location": {
          "height": 79,
          "left": 879,
          "top": 464,
          "width": 252
        },
        "word_name": "车次",
        "word": "C2097"
```

```
            },
            {
                "probability": {
                    "average": 0.980604,
                    "min": 0.932502,
                    "variance": 0.000352
                },
                "location": {
                    "height": 74,
                    "left": 982,
                    "top": 877,
                    "width": 206
                },
                "word_name": "乘车人",
                "word": "向宇波"
            },
            {
                "probability": {
                    "average": 0.994155,
                    "min": 0.903164,
                    "variance": 0.000396
                },
                "location": {
                    "height": 65,
                    "left": 1171,
                    "top": 593,
                    "width": 248
                },
                "word_name": "座位号",
                "word": "07车无座"
            },
            {
                "probability": {
                    "average": 0.993914,
                    "min": 1.2888,
                    "variance": 0.000009
                },
                "location": {
                    "height": 67,
                    "left": 429,
                    "top": 674,
                    "width": 193
                },
                "word_name": "价格",
                "word": "54.50"
            }
        ],
        "templateSign": "1c65a67f151df56ba4e29c4dddace5ee",
        "scores": 1,
        "isStructured": true,
        "logId": "153206517722624"
    },
    "error_code": 0,
    "error_msg": ""
}
```


iOCR财会版

简介

Hi，欢迎您使用百度 iOCR 财会版。

iOCR 财会版是 iOCR 自定义模板文字识别针对财会报销场景提出的专项解决方案，预置多种财务场景常用识别模板及财务票

据分类器，无需制作或训练即可直接使用；并提供混贴票据识别功能，可对粘贴在一张报销单上的多张不同种类发票进行切分识别；同时支持对未预置的固定版式票据可定制结构化识别模板和分类器。详情如下：

- **预置模板及分类器：** 预置财会场景常用的 8 类报销发票及银行回单、汇票、支票等 3 类银行单据识别模板，同时预置财会票据分类器，无需制作或训练，即可直接调用实现常用票据/单据的识别及财务票据分类能力
- **混贴票据识别：** 针对粘贴在一张图上的多张不同种类发票可进行自动检测分类及结构化识别，返回票据种类、位置及结构化识别结果
- **自定义模板/分类器：** 针对未预置且版式固定的票据、单据，仅需上传一张模板图片，即可通过框选参照字段和识别区自助制作一个识别模板，并建立图片中文字的 Key-Value 对应关系，实现对相同版式图片的结构化识别；同时，支持对已发布的多个识别模板自定义分类器，一步实现对不同版式图片的自动分类和结构化识别

iOCR 财会版针对财会场景进行深度定制，对常用的各类发票及银行单据进行专项优化及整合，同时提供对长尾票据的模板自定义功能，一站式解决财会报销的自动化识别问题，大幅度提高票据、单据的处理效率。如需了解更多，可访问 iOCR 财会版产品介绍页。

名词解释

- **票据：** 报销场景常用的各类票据凭证，如各类发票、交通票据等，通常打印在专用纸张，且各字段有固定打印位置
- **单据：** 财税场景常用的各类单据，如银行对账单、银行回单、银行汇票、银行支票等，通常由银行开具，用于内部财税核对
- **固定版式：** 指不同图片中的对应字段文字虽内容不同、长短不同，但可能出现的位置及范围固定不变的一类图片，如身份证、学生证、结婚证、火车票等；若图片中存在前文内容长度影响后文位置、表格行列不固定等情况的需针对不同情况制作多个模板
- **参照字段：** 相同版式的不同图片中 **位置和内容固定不变的字段**，可框选做为图片的锚点，用做对后续传入的图片进行模板匹配和矫正；建议选取 8 个以上、分散在图片四周、且不重复无换行的参照字段，效果最佳
- **识别区：** 图片中需要进行识别的字段，可通过框选及命名构建 「字段名称：识别区内容」 的 Key：Value 对应关系，用于对后续传入的相同版式图片的相同位置内容进行结构化识别；识别区的框选直接影响最终识别效果，框选区域需完全覆盖文字可能出现的范围
- **字段类型：** 针对不同识别区内容类型进行专项优化的切片识别模型，如小写金额、日期、纯数字等，可根据需求选择合适的字段类型以提升识别准确率；也可通过穷举可能的输出值范围，自定义字段类型，对识别结果进行智能纠正和规范
- **训练集图片：** 针对已发布的识别模板上传的相同版式的图片集合，用于提取该版式图片的视觉特征，对后续上传的图片进行自动分类。为了最佳的分类效果，训练集至少包含 30 张以上不重复的、版式相同的图片
- **分类关键词：**图片中存在的独有的文字内容，用作模板分类的文字依据，需保证填写的关键词在该版式图片中均有出现，可根据填写的关键词唯一确定图片所属模板类别
- **templateSign：** 模板ID，用于指定后期上传的图片用哪个模板来进行识别
- **classifierId：** 分类器ID，用于指定使用具体某个分类器，传入本参数后不用再传templateSign参数

预置能力介绍

预置模板

- **混贴票据：**支持对粘贴在同一张图上的多张不同种类发票进行自动检测分类并结构化识别，返回每张发票的种类、位置及结构化识别结果，目前支持对增值税发票、增值税卷票、定额发票、机打发票、火车票、出租车票、行程单、通行费发票、汽车票等 9 类发票进行检测识别，可点击 iOCR 财会版 - 功能演示 进行体验。



- **增值税发票：** 支持对增值税普票或专票所有30个字段进行结构化识别，包括发票基本信息、销售方及购买方信息、商品信息、价税信息等，其中四要素识别准确率超过99.9%
- **增值税卷票：**支持对增值税卷票的 16 个关键字段进行识别，包括发票类型、发票代码、发票号码、机打号码、机器编号、销售方纳税人识别号、开票日期、购买方纳税人识别号、项目、单价、数量、金额、税额、合计金额(小写)、合计金额(大写)、校验码
- **定额发票：** 支持对各类定额发票的发票代码、发票号码、金额3个关键字段进行结构化识别
- **通用机打发票：** 支持对国家/地方税务局发行的横/竖版通用机打发票的6个关键字段进行结构化识别，包括发票类型、发票号码、发票代码、开票日期、商品名称、合计金额
- **火车票：** 支持对红、蓝火车票的8个关键字段进行结构化识别，包括车票号码、始发站、目的站、车次、日期、票价、席别、姓名
- **出租车票：** 支持识别全国各大城市出租车票的6个关键字段，包括发票号码、代码、车号、日期、时间、金额
- **行程单：** 支持对飞机行程单的6个关键字段进行结构化识别，包括姓名、始发站、目的站、航班号、日期、票价
- **机动车销售发票：** 支持对机动车销售发票的14个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、机器编号、厂牌型号、发动机号码、车架号码、价税合计（大/小写）、销货单位名称、纳税人识别号、税率、税额、不含税价
- **银行回单：** 支持对各大银行回单的记账日期、出/入账户信息、交易金额等信息进行结构化识别，仅支持对单张图片上的单张回单进行识别
- **银行汇票：**支持对普通商业承兑汇票、普通银行承兑汇票、电子商业承兑汇票等多类汇票的 21 个关键字段进行结构化识别，包括收/付款人账号、出票金额、出票日期、承兑人信息等
- **银行支票：**支持对转账支票、现金支票、普通支票等多类银行支票的 7 个关键字段进行结构化识别，包括银行名称、出票日期、出票人账号、收款人、 付款行名称、金额、用途
- **银行回单（多张）（可选）：**支持对单张图片上的多张回单进行识别，可结构化输出每张银行回单的记账日期、出/入账户信息、交易金额等信息
- **大陆身份证正面（可选）：**支持对中国大陆二代居民身份证人像面所有 6 个字段进行结构化识别，包括姓名、性别、民族、出生日期、住址、身份证号
- **大陆身份证背面（可选）：**支持对中国大陆二代居民身份证国徽面的签发日期、失效日期、签发机关 3 个字段进行结构化识别
- **行驶证正页（可选）：**支持对机动车行驶证正页所有 10 个字段进行结构化识别，包括号牌号码、车辆类型、所有人、住址、使用性质、品牌型号、车辆识别代号、发动机号码、注册日期、发证日期
- **行驶证副页（可选）：**支持对机动车行驶证副页所有 11 个字段进行结构化识别，包括号牌号码、档案编号、核定载人数、总质量、整备质量、核定载质量、外廓尺寸、准牵引总质量、备注、检验记录、燃油类型
- **驾驶证（可选）：**支持对机动车驾驶证正本所有 9 个字段进行结构化识别，包括证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效起始日期、失效日期

- **银行卡（可选）**：支持对国内主流银行卡的卡号、有效期、发卡行、卡片类型 4 个关键字段进行结构化识别
- **营业执照（可选）**：支持对不同版式营业执照所有 11 个字段进行结构化识别，包括证件编号、社会信用代码、单位名称、地址、法人、类型、组成形式、注册资本、成立日期、有效日期、经营范围

> **注意：**
>
> - 预置模板默认已发布，可直接复制对应的 templateSign 使用API进行调用
> - 备注「可选」字样的预置模板未默认展示在「模板管理 - 预置模板」标签页，如需使用需点击页面右上角的「添加更多」按钮按需进行勾选添加

预置分类器

- **财会票据分类器：** 可对增值税发票、机打发票、定额发票、火车票、出租车票、行程单、机动车销售发票等 7 类财务票据进行自动分类并结构化识别

> **注意：** 预置分类器默认已发布，可直接复制对应的 classifierId（财会票据分类器：10001） 使用API进行调用

联系我们

本文档主要说明如何使用 iOCR财会版完成文字识别模型及分类器的制作，以及发布后如何通过 API 进行调用，如文档内容无法解决您的问题，可以通过以下方式寻求帮助：

- 在百度云控制台内 提交工单 联系专业的技术支持人员进行解答
- 进入 OCR 论坛 发布您的问题参与互动
- 加入百度 iOCR 交流群（群号:570832882）与更多开发者进行交流

使用流程

使用说明

本文档主要说明如何在 iOCR 财会版的可视化界面中完成模板、分类器及字段类型的自定义，同时对模板图片预处理、框选表格识别区等高级功能进行说明。

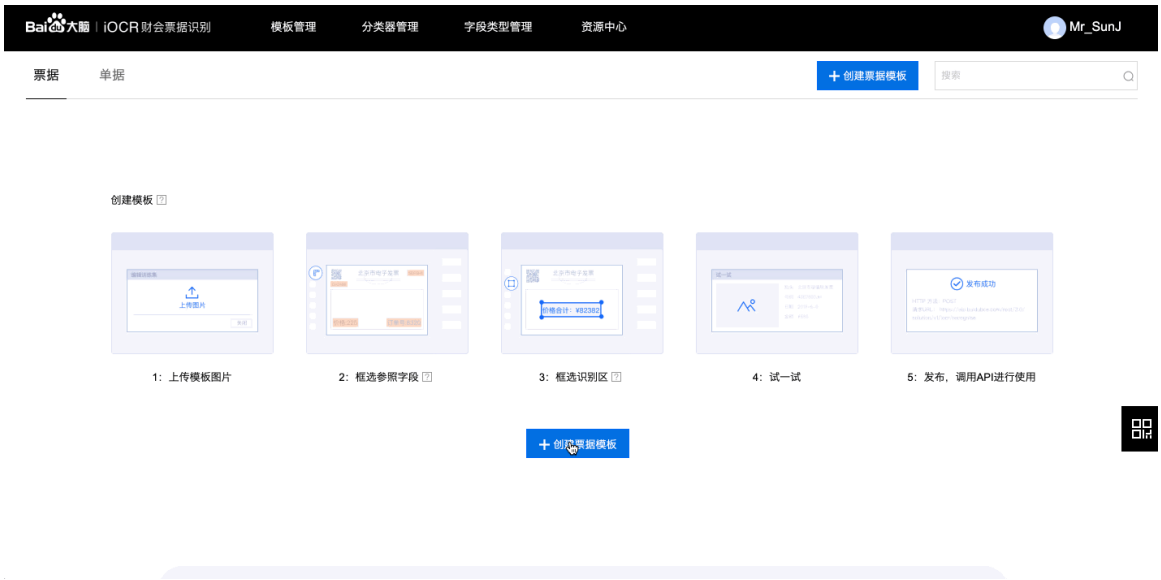
自定义模板

创建自定义模板的基本流程如下图所示，仅需上传一张规范的模板图片，通过简单框选，5 分钟即可完成结构化识别模板的制作。



Step 1: 上传模板图片

在 iOCR 财会版 - 模板管理页面 点击「自定义模板」标签页下方 「**创建模板**」 按钮，在弹出框中上传一张 **字迹清晰且摆放端正** 的模板图片（大小不超过4M，最长边不超过4096像素），并对模板进行命名。



Step 2: 框选参照字段

进入模板编辑页面，右侧操作步骤中选择 「**第1步：框选参照字段**」 标签，左侧工具栏选择 「**框选参照字段**」 按钮，使用鼠标在模板图片中框选位置和内容都固定不变的文字，如下图所示**橘色矩形**框选区域。



> **注意：** 「参照字段」为相同版式的不同图片中位置和内容固定不变的字段，可做为图片的锚点，用做对后续传入的图片进行模板匹配和矫正

> **框选Tips：**
>
> 1. 参照字段个数需保证在4个以上（推荐8个以上），并尽量分散在四角
> 2. 单个参照字段不可跨行，推荐字数在4个以内
> 3. 参照字段文字内容在上下文中不会重复出现
> 4. 仅支持框选中英文、数字，不可包含符号、图案

Step 3: 框选识别区

右侧操作步骤中点击 「**第2步：框选识别区**」 标签，左侧工具栏选择 「**框选识别区**」 按钮，使用鼠标在模板图片上框选业务场景需要进行识别的字段，如下图所示 **蓝色矩形** 框选区域；同时，填写 「**字段名称**」，并选择合适的 「**字段类型**」 以提高识别准确率。



注意： 「识别区」为图片中需要进行识别的字段，可通过框选及命名构建 「字段名称：识别区内容」 的 Key：Value 对应关系，用于对后续传入的相同版式图片的相同位置内容进行结构化识别

框选Tips：

1. 尽量扩大识别区框选范围，保证后续传入图片的对应字段内容可被完全覆盖，但同时也需保证不框选到其他字段内容
2. 选择合适的字段类型有助于提升字段识别效果
3. 如需识别图片内列宽固定的表格，可点击工具栏中「插入表格」按钮框选表格识别区，可参考表格识别区使用说明

Step 4: 试一试

参照字段和识别区全部框选完毕后，可点击页面右上角的 「**试一试**」 按钮进行识别效果测试，在弹出框中上传任意一张相同版式的图片即可，如下图所示。可点击图片下方「更换图片」按钮更换测试图片，如多次测试效果满意即可进行发布；如效果不满意可返回继续编辑。



注意：

1. 如试一试结果出现图片无法匹配模板的情况，需确认上传的测试图片与模板图片是否为同一版式，如确认无误可调节参照字段框选范围或更换参照字段，以提升模板匹配准确率
2. 如试一试结果中出现识别结果错误的情况，可调整识别区框选范围或更换识别区字段类型，以提升识别准确率

Step 5: 发布模板，调用API进行使用

如测试效果满意，可点击试一试弹出框右下角的 「**立即发布**」 按钮或模板编辑页面右上角的 「**发布**」 按钮进行发布，发布成功后即可通过模板ID调用该模板，调用方式可查看 API文档。



> **注意：** 只有发布后的模板才能通过线上接口进行调用，如果编辑未发布，那么仅仅是生成了一个新的版本，此时对模板的任何修改都不会影响线上调用。

自定义分类器

创建自定义分类器仅需针对每个已发布的模板上传 30 张相同版式的训练集图片或填写分类关键词，即可自动训练一个能够对所选模板进行自动分类的分类器。

预置分类器默认已添加并发布，可直接复制对应的 classifierId（财会票据分类器：10001） 使用API进行调用

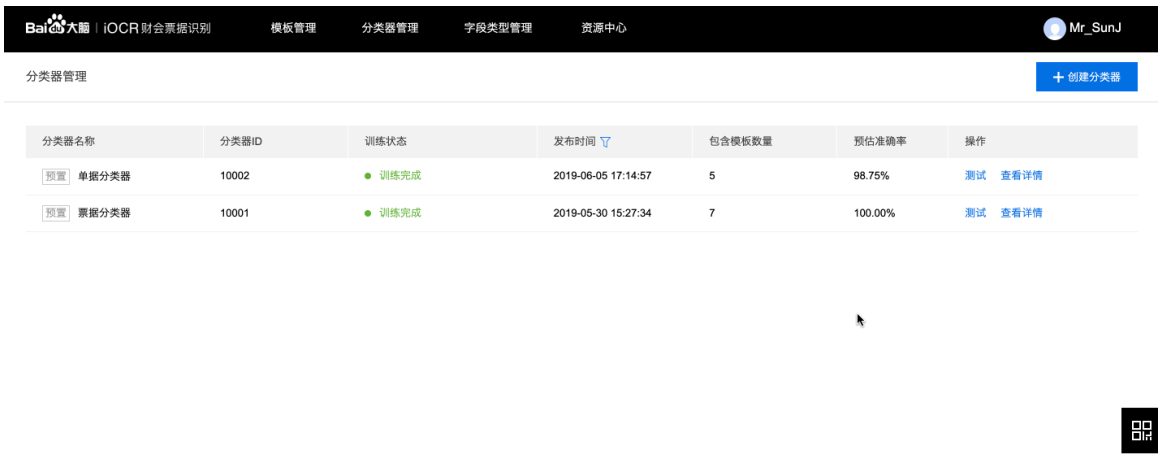


1：选择模板　2：上传训练集　3：训练　4：测试　5：发布

Step 1: 选择模板

在 iOCR 财会版 - 分类器管理页面 点击右上角 「**创建分类器**」 按钮，在弹出框中填写「分类器名称」及「功能描述」，填写完毕点击 「**确认**」 即可进入分类器编辑页面。

在分类器编辑页面中点击 「**添加自定义模板**」 或 「**添加预置模板**」，选择 **已发布的** 自定义模板或预置模板加入训练队列。



> **说明：**
>
> **预置模板：** 系统预置的常用卡证、票据模板，已上传100张训练集图片并已填写分类关键词信息，可直接勾选用于对应图片的自动分类和结构化识别
>
> **自定义模板：** 已发布的自定义模板

Step 2: 上传训练集/填写分类关键词

点击选中的自定义模板所在行右侧的 「**编辑训练集**」 按钮，在弹出框中上传一个包含 30 张以上相同版式图片、不超过 200MB 的 ZIP 格式压缩包。



> **注意：**「训练集」为针对已发布的识别模板上传的相同版式的图片集合，用于训练分类器对后续上传的图片进行自动分类；为了最佳的分类效果，训练集至少包含 30 张以上不重复的、版式相同的图片；系统模版已预置 100 张训练集图片，无需上传

也可在「**分类关键词**」列填写对应模板图片中的关键文字内容，需保证填写的关键词在该版式图片中均有出现，且具有特异性，可根据填写的关键词唯一确定图片所属模板类别

如身份证人像面的“姓名”、“性别”、“民族”、“出生”、“公民身份号码”五个字段在每一张身份证人像面均会出现，且 5 个字段全部出现时基本可确定该图片为身份证人像面，则此 5 个字段即可作为身份证人像面的分类关键词

> **注意：**每个模板最多可设置 5 个分类关键词，每个关键词最多包含 15 个中英文字符，不可包含其他特殊符号，关键词之间使用（中/英文）逗号进行间隔

Step 3: 训练

参与分类的模板及其训练集图片数量或分类关键词符合标准后，即可点击页面左下角的 「**开始训练**」 按钮进行训练，等待训练完毕后即可在分类器列表中进行查看，训练完成后的分类器有以下三种分类形式：

- **纯图片特征分类器：**如所有模板各上传了 30 张以上相同版式训练集图片，但均未填写分类关键词，则训练出的分类器将完全依据图片特征进行分类；
- **纯文字特征分类器：**如所有模板均填写了分类关键词，但部分或全部模板未上传训练集图片，则训练出的分类器将完全依据关键词的文字信息进行分类；
- **图文协同分类器：**如所有模板各上传了 30 张以上相同版式训练集图片，且部分或全部模板填写了分类关键词，则训练出的分类器将先进行关键词匹配，如匹配到相同关键词则再根据图片特征进行区分。



Step 4: 测试

在分类器管理页面的分类器列表中点击训练完毕的分类器右侧的 「**测试**」 按钮进行效果测试，在弹出框中上传任意一张图片即可，如下图所示，测试结果中包含分类结果及结构化识别结果。可点击图片下方「更换图片」按钮更换测试图片，如多次测试效果满意即可进行发布；如效果不满意可返回继续编辑。



Step 5: 发布

如测试效果满意，可点击测试弹出框右下角的 「**立即发布**」 按钮或分类器列表右侧的 「**发布**」 按钮进行发布，发布成功后即可通过分类器ID调用该分类器，实现图片的自动分类及结构化识别，调用方式可查看API文档。



自定义字段类型

创建自定义字段类型的基本流程如下图所示，仅需输入全部可能出现的输出值，即可创建一个自定义字段类型在制作模板时进行使用，用于字段识别结果的规范和智能纠正。



1：编辑字段词典 2：开始使用

Step 1: 编辑字段词典

在 iOCR 财会版 - 字段类型管理页面 点击下方 「**创建字段类型**」 按钮，在弹出框中填写「字段类型名称」及「词典内容」，在词典内容中列举全部可能出现的输出值内容，填写完毕点击 「**立即创建**」 即可完成创建；也可点击 「**添加预置类型**」 按钮直接添加预置的常用字段类型，无需自行编辑词典内容。



Step 2: 开始使用

创建成功的字段类型即可在制作自定义模板选择字段类型时，点击 「**我的字段类型**」 进行勾选使用，使用后即可对该字段的识别结果进行规范或智能纠正。



高级功能

模板图片预处理

为保证模板的识别效果，建议上传的模板图片 **字迹清晰且摆放端正**，以保证后续传入的同版式图片能被匹配、矫正并准确识别。

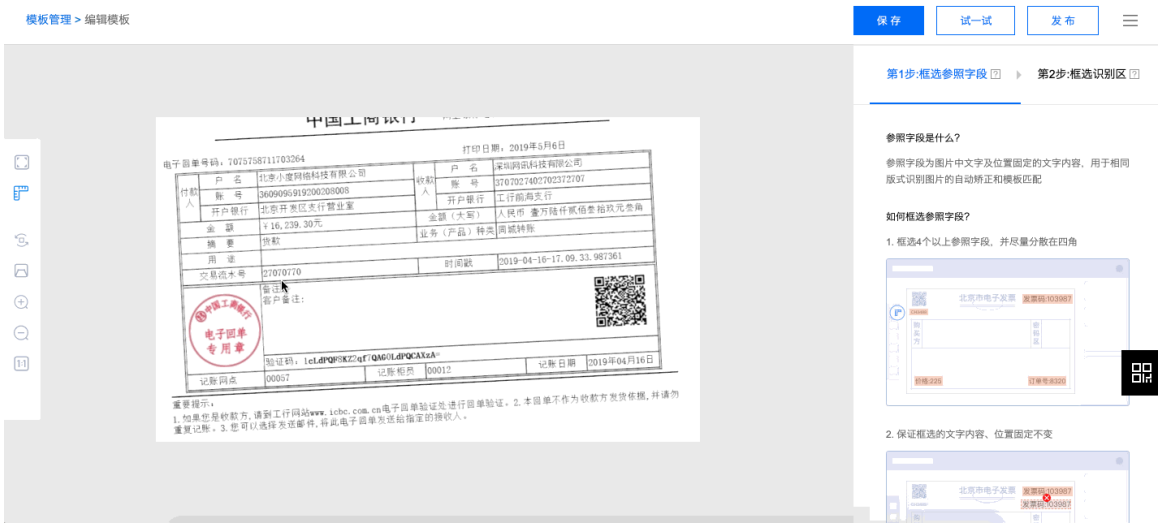
如您无法保证上传的模板图片摆放端正，可 **在框选参照字段/识别区前** 通过我们提供的模板图片预处理功能对模板图片进行 **旋转/透视裁剪** 操作纠正倾斜图片。

图片旋转

点击编辑区左侧工具栏中的 「**图片旋转**」 按钮，即可进入旋转处理页面。

在旋转处理页面，使用鼠标按住上方旋转按钮拖动图片顺时针/逆时针旋转，旋转效果满意后松开鼠标点击 「**保存**」 按钮即

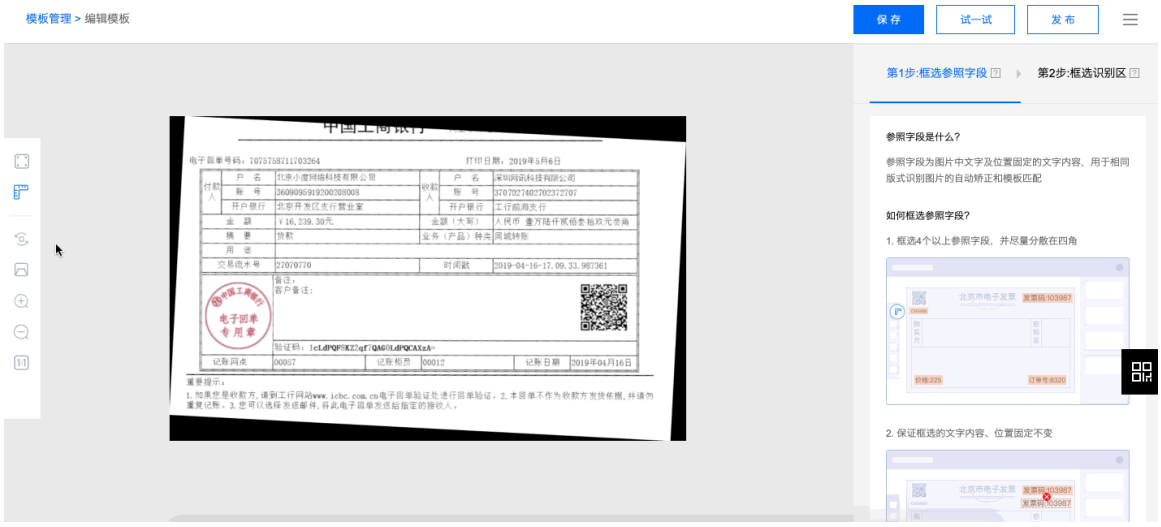
可。



图片透视裁剪

点击编辑区左侧工具栏中的 「**透视裁剪**」 按钮，即可进入裁剪处理页面。

在裁剪处理页面，使用鼠标按住四角点小圆点进行拖拽，使四边紧贴卡证或票据边缘，拖拽完毕后点击 「**保存**」 按钮即可。



框选表格识别区

如果要识别的图片中存在行列固定的表格，可点击工具栏中的 「**插入表格**」 按钮添加指定列数的表格识别区，通过拖拽表格四角及列分隔线的小白点使识别区覆盖表格区域，并使各列识别区与表格各列对应，即可对表格内容进行结构化识别。

> **注意：** 表格识别区仅需覆盖表格内容，无需将表头内容进行框选



如您在操作过程中出现上述内容未说明的问题，可参考 iOCR常见问题，或在 OCR论坛 发布您的问题，也可加入百度iOCR交流群（群号:570832882）与更多开发者进行交流。

API文档

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise/finance`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file三选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（data:image/jpeg;base64，） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式
url	和image/pdf_file 三选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
pdf_file	和image/url三选一	string	-	PDF文件，base64编码后进行urlencode，需去掉编码头（data:application/pdf;base64，） 要求base64编码和urlencode后大小不超过4M **注：**目前仅支持单页PDF识别，如上传的为多页PDF，仅识别第一页
templateSign	否	string	-	模板 ID，自定义模板或预置模板的唯一标示，可用于调用指定的识别模板进行结构化识别，可在「模板管理」页查看并复制使用
classifierId	否	string	-	分类器Id，分类器的唯一标示，可用于调用指定的分类器对传入的图片进行自动分类及识别 **与 templateSign 至少存在一个，如同时存在，则优先级 templateSign > classfierId**

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
##### 请求模板id
curl -i -k 'https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise/finance?access_token=【调用鉴权接口获取的token】' --data 'templateSign=xxx&image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
##### 请求分类器id
curl -i -k 'https://aip.baidubce.com/rest/2.0/solution/v1/iocr/recognise/finance?access_token=【调用鉴权接口获取的token】' --data 'classifierId=xxx&image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
logid	是	uint64	唯一的log id，用于问题定位
error_code	是	int	0代表成功，如果有错误码返回可以参考下方错误码列表排查问题
error_msg	是	string	如果error_code具体的失败信息，可以参考下方错误码列表排查问题
data	是	jsonObject	识别返回的结果
+ ret	是	jsonArray	识别出来的字段数组，每一个单元里包含以下几个元素
++ word_name	否	string	isStructured 为 true 时存在，表示字段的名字；如果 isStructured 为 false 时，不存在
++ word	否	string	识别的字符串或单字
++ location	否	jsonObject	字段在原图上对应的矩形框位置，通过上边距、左边距、宽度、高度表示
++ probability	否	jsonObject	字段的置信度，包括平均、最小和方差
+ templateSign	否	string	图片分类结果对应的模板id或指定使用的模版id。templateSign的对应关系为： - mixed_receipt：混贴发票，可对粘贴单中的多张不同票据进行检测分类，返回每张发票的类别及识别结果； - vat_invoice：增值税发票； - taxi_receipt：出租车票； - roll_ticket：卷票； - train_ticket：火车票； - quota_invoice：定额发票； - air_ticket：行程单； - car_invoice：汽车票； - toll_invoice：通行费发票； - printed_invoice：机打发票。
+ scores	否	float	分类置信度，如果指定templateSign，则该值为1
+ isStructured	否	string	表示是否结构化成功，true为成功，false为失败；成功时，返回结构化的识别结果；失败时，如果能识别，按行返回结果，如果不能识别，返回空

返回示例

- 使用自定义模板及自定义分类器功能时，返回结果可参考 iOCR通用版-返回示例；
- templateSign = mixed_receipt 时，返回结果如下所示：

```
{
"data": {
 "ret": [
  {
```

```
"ret": [
 {
  "rect": {
   "top": 277,
   "left": 237,
   "width": 61,
   "height": 10
  },
  "probability": {
   "average": 0.98831981420517,
   "min": 0.96548694372177
  },
  "word_name": "AmountInWords",
  "word": "叁佰陆拾圆整"
 },
 {
  "rect": {
   "top": 29,
   "left": 482,
   "width": 85,
   "height": 18
  },
  "probability": {
   "average": 0.99745708703995,
   "min": 0.99514311552048
  },
  "word_name": "InvoiceNumConfirm",
  "word": "07286261"
 },
 {
  "rect": {
   "top": 352,
   "left": 393,
   "width": 32,
   "height": 12
  },
  "probability": {
   "average": 0.99022936820984,
   "min": 0.98398983478546
  },
  "word_name": "NoteDrawer",
  "word": "余佳燕"
 },
 {
  "rect": {
   "top": 326,
   "left": 158,
   "width": 214,
   "height": 10
  },
  "probability": {
   "average": 0.94039279222488,
   "min": 0.39105615019798
  },
  "word_name": "SellerAddress",
  "word": "杭州市转塘科技经济区块16号8幢0571-85022088"
 },
 {
  "rect": {
   "top": 311,
   "left": 171,
   "width": 146,
```

```
      "height": 12
     },
     "probability": {
      "average": 0.99681425094604,
      "min": 0.98468536138535
     },
     "word_name": "SellerRegisterNum",
     "word": "91330106673959654P"
    },
    {
     "rect": {
      "top": 0,
      "left": 0,
      "width": -1,
      "height": -1
     },
     "probability": {
      "average": 0,
      "min": 0
     },
     "word_name": "MachineCode",
     "word": ""
    },
    {
     "rect": {
      "top": 0,
      "left": 0,
      "width": -1,
      "height": -1
     },
     "probability": {
      "average": 0,
      "min": 0
     },
     "word_name": "Remarks",
     "word": ""
    },
    {
     "rect": {
      "top": 339,
      "left": 158,
      "width": 181,
      "height": 11
     },
     "probability": {
      "average": 0.99247741699219,
      "min": 0.8911309838295
     },
     "word_name": "SellerBank",
     "word": "招商银行杭州高新支行502905023610702"
    },
    {
     "rect": {
      "top": 259,
      "left": 576,
      "width": 43,
      "height": 10
     },
     "probability": {
      "average": 0.97683322429657,
      "min": 0.89436012506485
     },
     "word_name": "TotalTax"
```

```
    "word_name": "TotalTax",
    "word": "20.38"
   },
   {
    "rect": {
     "top": 32,
     "left": 124,
     "width": 101,
     "height": 16
    },
    "probability": {
     "average": 0.99661940336227,
     "min": 0.99573355913162
    },
    "word_name": "InvoiceCodeConfirm",
    "word": "3321192130"
   },
   {
    "rect": {
     "top": 0,
     "left": 0,
     "width": -1,
     "height": -1
    },
    "probability": {
     "average": 0,
     "min": 0
    },
    "word_name": "CheckCode",
    "word": ""
   },
   {
    "rect": {
     "top": 32,
     "left": 124,
     "width": 101,
     "height": 16
    },
    "probability": {
     "average": 0.99661940336227,
     "min": 0.99573355913162
    },
    "word_name": "InvoiceCode",
    "word": "3321192130"
   },
   {
    "rect": {
     "top": 65,
     "left": 534,
     "width": 73,
     "height": 12
    },
    "probability": {
     "average": 0.99508810043335,
     "min": 0.97497177124023
    },
    "word_name": "InvoiceDate",
    "word": "2019年08月28日"
   },
   {
    "rect": {
     "top": 104,
     "left": 168,
```

```
      "width": 147,
      "height": 12
     },
     "probability": {
      "average": 0.9933996796608,
      "min": 0.96598559617996
     },
     "word_name": "PurchaserRegisterNum",
     "word": "91110911717743469K"
    },
    {
     "rect": {
      "top": 18,
      "left": 257,
      "width": 164,
      "height": 19
     },
     "probability": {
      "average": 0.99611341953278,
      "min": 0.98104286193848
     },
     "word_name": "InvoiceTypeOrg",
     "word": "浙江增值税专用发票"
    },
    {
     "rect": {
      "top": 93,
      "left": 405,
      "width": 191,
      "height": 45
     },
     "probability": {
      "average": 0.97755342721939,
      "min": 0.82740485668182
     },
     "word_name": "Password",
     "word": "508>3909>1*>01/-46709-6/3+*7+8>/1*19+7-0**>+58290-6>647-
+324865*9*1<*2191/7754/<0>2<838+//5-69--748*<251408<"
    },
    {
     "rect": {
      "top": 0,
      "left": 0,
      "width": -1,
      "height": -1
     },
     "probability": {
      "average": 0,
      "min": 0
     },
     "word_name": "Agent",
     "word": "否"
    },
    {
     "rect": {
      "top": 278,
      "left": 511,
      "width": 54,
      "height": 10
     },
     "probability": {
      "average": 0.95414996147156,
```

```
    "min": 0.68566131591797
   },
   "word_name": "AmountInFiguers",
   "word": "360.00"
  },
  {
   "rect": {
    "top": 134,
    "left": 159,
    "width": 204,
    "height": 11
   },
   "probability": {
    "average": 0.97773444652557,
    "min": 0.61343103647232
   },
   "word_name": "PurchaserBank",
   "word": "招商银行北京分行大电路支行866180100210002"
  },
  {
   "rect": {
    "top": 352,
    "left": 259,
    "width": 26,
    "height": 12
   },
   "probability": {
    "average": 0.98384791612625,
    "min": 0.97088402509689
   },
   "word_name": "Checker",
   "word": "柳余"
  },
  {
   "rect": {
    "top": 0,
    "left": 0,
    "width": -1,
    "height": -1
   },
   "probability": {
    "average": 0,
    "min": 0
   },
   "word_name": "City",
   "word": ""
  },
  {
   "rect": {
    "top": 258,
    "left": 460,
    "width": 49,
    "height": 11
   },
   "probability": {
    "average": 0.98758614063263,
    "min": 0.9416212439537
   },
   "word_name": "TotalAmount",
   "word": "339.62"
  },
  {
   "rect": {
```

```
    "rect": {
     "top": 90,
     "left": 159,
     "width": 150,
     "height": 12
    },
    "probability": {
     "average": 0.96976244449615,
     "min": 0.70321601629257
    },
    "word_name": "PurchaserName",
    "word": "百度在线网络技术(北京)有限公司"
   },
   {
    "rect": {
     "top": 0,
     "left": 0,
     "width": -1,
     "height": -1
    },
    "probability": {
     "average": 0,
     "min": 0
    },
    "word_name": "Province",
    "word": "浙江"
   },
   {
    "rect": {
     "top": 18,
     "left": 257,
     "width": 164,
     "height": 19
    },
    "probability": {
     "average": 0.99611341953278,
     "min": 0.98104286193848
    },
    "word_name": "InvoiceType",
    "word": "专用发票"
   },
   {
    "rect": {
     "top": 145,
     "left": 626,
     "width": 9,
     "height": 28
    },
    "probability": {
     "average": 0.99723851680756,
     "min": 0.99662339687347
    },
    "word_name": "SheetNum",
    "word": "第二联"
   },
   {
    "rect": {
     "top": 119,
     "left": 159,
     "width": 158,
     "height": 12
    },
    "probability": {
```

```
    "average": 0.89263164997101,
    "min": 0.21246993541718
   },
   "word_name": "PurchaserAddress",
   "word": "北京市海淀区上地十侧10号百厘大厦三座"
  },
  {
   "rect": {
    "top": 353,
    "left": 113,
    "width": 22,
    "height": 10
   },
   "probability": {
    "average": 0.84802502393723,
    "min": 0.560218334198
   },
   "word_name": "Payee",
   "word": "佳机"
  },
  {
   "rect": {
    "top": 298,
    "left": 158,
    "width": 85,
    "height": 11
   },
   "probability": {
    "average": 0.96288979053497,
    "min": 0.8344641327858
   },
   "word_name": "SellerName",
   "word": "阿里云计算有限公司"
  },
  {
   "rect": {
    "top": 29,
    "left": 482,
    "width": 85,
    "height": 18
   },
   "probability": {
    "average": 0.99745708703995,
    "min": 0.99514311552048
   },
   "word_name": "InvoiceNum",
   "word": "07286261"
  },
  {
   "rect": {
    "top": 163,
    "left": 71,
    "width": 116,
    "height": 11
   },
   "probability": {
    "average": 0.9905007481575,
    "min": 0.98428183794022
   },
   "word_name": "DetailsOfTax#1#CommodityName",
   "word": "*信息技术服务*软件服务费"
  },
```

```
    {
     "word_name": "DetailsOfTax#1#CommodityType",
     "word": ""
    },
    {
     "rect": {
      "top": 164,
      "left": 292,
      "width": 10,
      "height": 10
     },
     "probability": {
      "average": 0.89159053564072,
      "min": 0.80279469490051
     },
     "word_name": "DetailsOfTax#1#CommodityUnit",
     "word": "套"
    },
    {
     "rect": {
      "top": 166,
      "left": 360,
      "width": 7,
      "height": 8
     },
     "probability": {
      "average": 0.95917397737503,
      "min": 0.95113134384155
     },
     "word_name": "DetailsOfTax#1#CommodityNum",
     "word": "1"
    },
    {
     "rect": {
      "top": 165,
      "left": 397,
      "width": 28,
      "height": 9
     },
     "probability": {
      "average": 0.97336208820343,
      "min": 0.89550644159317
     },
     "word_name": "DetailsOfTax#1#CommodityPrice",
     "word": "339.62"
    },
    {
     "rect": {
      "top": 165,
      "left": 480,
      "width": 28,
      "height": 9
     },
     "probability": {
      "average": 0.98559606075287,
      "min": 0.96932607889175
     },
     "word_name": "DetailsOfTax#1#CommodityAmount",
     "word": "339.62"
    },
    {
     "rect": {
```

```
      "top": 165,
      "left": 522,
      "width": 11,
      "height": 9
     },
     "probability": {
      "average": 0.92075562477112,
      "min": 0.8382123708725
     },
     "word_name": "DetailsOfTax#1#CommodityTaxRate",
     "word": "6%"
    },
    {
     "rect": {
      "top": 165,
      "left": 595,
      "width": 24,
      "height": 9
     },
     "probability": {
      "average": 0.97626549005508,
      "min": 0.9149768948555
     },
     "word_name": "DetailsOfTax#1#CommodityTax",
     "word": "20.38"
    }
   ],
   "receiptCoordinate": "{\"height\":642,\"left\":155,\"top\":178,\"width\":1106}",
   "error_msg": "success",
   "templateSign": "vat_invoice",
   "scores": 1,
   "templateName": "增值税发票",
   "isStructured": true,
   "error_code": 0
  },
  {
   "ret": [
    {
     "probability": {
      "average": 0,
      "min": 0
     },
     "word_name": "PickupTime",
     "word": "16:50"
    },
    {
     "probability": {
      "average": 0,
      "min": 0
     },
     "word_name": "DropoffTime",
     "word": "17:06"
    },
    {
     "rect": {
      "top": 212,
      "left": 48,
      "width": 66,
      "height": 9
     },
     "probability": {
      "average": 0.98931306600571,
      "min": 0.92181307077408
```

```
    "min": 0.92181307077408
  },
  "word_name": "Time",
  "word": "16:50-17:06"
 },
 {
  "probability": {
   "average": 0,
   "min": 0
  },
  "word_name": "City",
  "word": ""
 },
 {
  "rect": {
   "top": 288,
   "left": 84,
   "width": 30,
   "height": 9
  },
  "probability": {
   "average": 0.99606895446777,
   "min": 0.99255055189133
  },
  "word_name": "FuelOilSurcharge",
  "word": "1.00"
 },
 {
  "rect": {
   "top": 198,
   "left": 53,
   "width": 61,
   "height": 9
  },
  "probability": {
   "average": 0.99483448266983,
   "min": 0.98498445749283
  },
  "word_name": "Date",
  "word": "2019-03-20"
 },
 {
  "probability": {
   "average": 0,
   "min": 0
  },
  "word_name": "Province",
  "word": "陕西省"
 },
 {
  "probability": {
   "average": 0,
   "min": 0
  },
  "word_name": "CallServiceSurcharge",
  "word": "0.00"
 },
 {
  "rect": {
   "top": 275,
   "left": 76,
   "width": 38,
   "height": 9
```

```
    },
    "probability": {
     "average": 0.98517167568207,
     "min": 0.97685235738754
    },
    "word_name": "Fare",
    "word": "21.10"
   },
   {
    "rect": {
     "top": 314,
     "left": 76,
     "width": 39,
     "height": 9
    },
    "probability": {
     "average": 0.97668653726578,
     "min": 0.93554848432541
    },
    "word_name": "TotalFare",
    "word": "2.00"
   },
   {
    "rect": {
     "top": 173,
     "left": 72,
     "width": 42,
     "height": 8
    },
    "probability": {
     "average": 0.98336416482925,
     "min": 0.88234621286392
    },
    "word_name": "TaxiNum",
    "word": "BQ6353"
   },
   {
    "rect": {
     "top": 225,
     "left": 89,
     "width": 25,
     "height": 8
    },
    "probability": {
     "average": 0.99482887983322,
     "min": 0.99453765153885
    },
    "word_name": "PricePerkm",
    "word": "2.30"
   },
   {
    "rect": {
     "top": 124,
     "left": 14,
     "width": 90,
     "height": 11
    },
    "probability": {
     "average": 0.99889290332794,
     "min": 0.99876511096954
    },
    "word_name": "InvoiceCode",
```

```
      "word": "161001881016"
     },
     {
      "rect": {
       "top": 238,
       "left": 96,
       "width": 18,
       "height": 8
      },
      "probability": {
       "average": 0.99126571416855,
       "min": 0.98285579681396
      },
      "word_name": "Distance",
      "word": "6.0"
     },
     {
      "rect": {
       "top": 137,
       "left": 14,
       "width": 60,
       "height": 10
      },
      "probability": {
       "average": 0.99211621284485,
       "min": 0.94001615047455
      },
      "word_name": "InvoiceNum",
      "word": "05070716"
     },
     {
      "probability": {
       "average": 0,
       "min": 0
      },
      "word_name": "Location",
      "word": "陕西省"
     }
    ],
    "receiptCoordinate": "{\"height\":618,\"left\":1325,\"top\":200,\"width\":215}",
    "error_msg": "success",
    "templateSign": "taxi",
    "scores": 1,
    "templateName": "出租车发票",
    "isStructured": true,
    "error_code": 0
   }
  ],
  "templateSign": "mixed_receipt",
  "templateName": "混贴票据",
  "scores": 1,
  "isStructured": true,
  "logId": "164196999300761",
  "version": 1
 },
 "error_code": 0,
 "error_msg": "",
 "log_id": "164196999300761"
}
```


常见问题

Q：什么是自定义模板文字识别？什么场景下我该使用这个产品？

A：自定义模板文字识别是一款您可以针对各种票据、卡证实现字段名和字段值对应提取的OCR产品；举例：当您需要识别某一种证件（比如房产证），但是百度官方还没有针对这种类型的证件推出具体的模板识别接口，而使用通用文字识别无法实现字段名和字段值对应化的提取，这种情况下您可以使用自定义模板文字识别产品，实现结构化的数据提取。

Q：产品实现的原理是什么？

A：基本原理：如果您要识别特定的一类具有固定格式的文档，首先上传一张票据、卡证的图片作为模板（*用于制作模板的图片要求摆放端正、平整，拍摄时避免过曝，阴影等不良情况*），然后在模板上框选一些固定的字段作为【参照字段】。后续调用识别接口时，会将新上传的图片以【参照字段】为锚点扭正到和模板图片一致。最后框选需要识别的区域作为【识别区】，框选后在右侧给该识别区命名，点击保存，这便完成了一个模板的制作。

Q：自定义模板文字识别支持哪些浏览器？

A：推荐使用Chrome（版本58及以上），暂时不支持Safari。

Q：上传的图片有大小限制吗？上传什么样的图片效果会比较好？

A：上传用于模板制作的图片，最大：小于等于4M，且分辨率小于等于4096像素乘4096像素，最小：大于等于15像素乘15像素且大于等于1KB，后期上传识别的图片最大：大小不超过4M，且分辨率小于等于4096像素乘4096像素，最小：大于等于15像素乘15像素且大于等于1KB。 为了保证更好的效果，建议模板图片：

1. 模板图片清晰平整，摆放端正
2. 模板图片格式以 **.jpg 为最佳**，png、bmp格式识别效果欠佳
3. 模板图片尽量突出需要识别的部分，请先手动剪裁掉不需要的部分，提高识别率
4. 模板图片大小建议为：转为base64编码后不超过1M，不宜过大或过小
5. 模板图片中，大多数汉字的大小保持在 32*32像素 左右，不符合的整体缩放调整

Q：我该怎么使用自定义模板文字识别？

A：首先您需要制作您的模板，在Chrome中打开 ai.baidu.com/iocr 进入模板管理界面，此时需要您首先登录百度账号（和您的百度网盘、百度贴吧、百度文库等百度系产品通用），进入后点击创建模板，进入模板编辑界面，首先您需要给您的模板进行命名，然后点击左侧编辑框中的按钮上传模板图片（模板图片要求端正、清晰），然后框选字段值，框选后在右侧对应位置填写字段名，全部框选完后点击右侧“参照字段”标签，在图中框选参照字段（要求参见下一条Q&A），完成后点击保存，则您已经制作完您的模板，此时您可以点击“发布”按钮，把次模板发布到线上环境（保存只是保存修改记录，不会实时生效，发布后您的所有操作才会生效），然后您可以参照文档中的“请求说明”上传图片，并制定templateSign（模板标识），来指定上传的图片使用该模板。

> 视频教程请参见 iOCR通用版使用教程（视频版）

Q：模板制作过程中怎么进行图片的缩放？

A：可以使用工具栏中的放大缩小工具，或使用鼠标滚轮，或使用触摸板（如果您的设备具有触摸板）进行双指缩放。

Q：什么是参照字段，选取时有什么注意点？

A：为了将您后期上传的图片矫正成和您模板图片以在同样的位置区间寻找关键值，您需要在制作模板的时候在“参照字段”标签页下框选至少4个（推荐框选8个以上）的参照字段，参照字段的选取需要点击编辑模板界面右上角工具栏中的“设置参照字段”工具，然后在图上拖动选取固定文字。 框选时有一些注意点：

- 同一参照字段的文字**必须**在同一行；
- 参照字段**必须**是模板图片和后期上传的图片中**共同拥有并且内容和位置都不变**的文字；
- 参照字段尽量**四散**在图片的边缘，尤其是四角；
- 参照字段尽量在模板图片上**唯一**，在图片中多次出现的文字段效果较差；

- 参照字段尽量**多**，至少4个，强烈推荐标注**8个**以上的参照字段，参照字段越多越分散识别效果越好；
- *如果后期测试时显示“未匹配到模板”则是因为参照字段选取和识别的问题，请按照上述要求检查核对并重新选取参照字段。*

Q：框选参照字段的时候发现识别错了，可以纠正吗？

A：可以的，您可以点击参照字段后面的编辑按钮，对参照字段进行人工纠正，输入正确的文字内容。修改正确参照字段的内容有助于提升后期模板匹配效果。纠正的规则是：

- 不能添加/删除超过两个字符，并且如果您框选的参照字段范围比较小，无法放下新增的两个字符，则您需要适当扩大该参照字段的框选范围
- 跨行的参照字段无法编辑，请先改为框选单行文字
- 不能将参照字段内容删除为空

Q：框选识别区时有什么办法可以提高准确率？

A：如果您选择的识别区内容正好为以下表格中的某一项，您可以选择对应的字段类型来提升识别效果：

字段属性	**适用范围**	**输出结果**
常规	适用于全场景识别， 如果该切片属于下列属性中任意一个， 建议使用下面的切片属性来提高准确率	识别区所有内容
小写数字金额	各类票据中金额数字	结果只返回至少包含小数点后两位的数值（不满两位则默认补充为##.00）并且会**忽略所有的非数字以外的文字和符号（也会丢弃¥、$）**
日期	**单个**日期如2018年7月19日	结果会做归一化处理统一以20180719格式返回
长串数字	如运单号、票号	结果只返回长串数字、英文组合
手写汉字	手写中文汉字	该识别区进行全量识别，但是对手写汉字有较高的准确率
手写数字	手写阿拉伯数字	该识别区进行全量识别，但是对手写数字有较高的准确率
数字/英文/符号混合	发票密码区	识别区所有内容，相比于“常规”识别率更高
我的字段类型	您可以在【字段类型管理】 中为字段值是有限集合的字段上传词典， 限定输出范围	智能匹配后的词典值

Q：在框选字段值/框选参照字段的时候不小心多点击增加了一个错误的选择框，应该怎么删除？

A：可以点击右侧的“识别区”/“参照字段”下面对应字段后的X按钮进行删除。

Q：有些识别区容易漏字、识别不准怎么办？

A：在对应识别区的“字段类型”中选择“数字/英文/符号混合”可以提高该字段的识别效果。

Q：保存和发布是什么关系/发布是用来干嘛的？

A：考虑到很多用户会把自己制作的模板使用到业务中去，所以为了尽可能的保证您业务的连续性，我们的模板编辑完后点击保存时只是把您的编辑操作保存到云端，此时，您线上使用的模板还是您之前的模板，直至您对刚才的模板进行发布操作。举例：您在2月1日生成了A模板，并点击发布，此时您调用这个接口时是使用的2月1日的A模板，然后您在3月1日对模板进行了修改，点击了保存，此时您在3月1日的所有编辑操作都已经保存在云端，但是您线上使用的模板仍然是2月1日的A模板，如果您需要使用3月1日的模板去替换2月1日的版本，您需要对3月1日编辑过的A模板进行发布操作，当您点击发布以后，您调用这个接口使用的将会是3月1日编辑的A模板。

Q：修改历史是什么？/我能回退到之前某个版本吗？

A：点击“修改历史”右侧的小箭头即可展开这个模板的版本记录，版本记录从新到旧列出了您针对这个模板修改的各个版本，您可以点击对应版本右侧的“退回到此刻”来将模板回退到当时那个版本，您框选的取值范围、参照字段都会回退到当时版本的设置，但是您的模板名字不会因此回退。

Q：制作完模板以后我可以给模板改名字吗？回退到其他版本的时候名字也会回退吗？

A：您可以在模板编辑页面随时修改您的名字，修改完名字以后需要您点击保存，此时会生成一个新的版本；在您回退到过去的某个版本的时候模板名字不会回退。

Q：分类时是否数据越多越好？

A：不一定，我们建议您每个模板提供30张同模板的训练集，如果您训练图片较丰富可以提供100张以下的图片，每张图片建议不超过500kb，选择的图片要尽量覆盖到使用的场景。同时更多的图片会导致训练时间加长。

Q：分类的细粒度大概是什么样的？

A：现在分类的细粒度为视觉元素层面有较明显的区分的图片，如身份证、银行卡、户口本这些人类能快速区分开的卡证、票据。但是无法做到需要根据文字内容来进行区分的地步，如：无法区分北京增值税专票和天津增值税专票。

Q：分类时训练数据不很是很多，能否用同一张照片PS处理成多张不同的图片来进行训练？

A：非常不建议您这么做。这样拟合出来的分类器模型没有很强的泛化能力，分类的准确性会大幅下降。我们还是建议您使用真实场景中需要分类的数据进行训练。

Q：为什么训练以后显示准确率100%，但是还会有分类错误的情况呢？

A：界面上显示的分类器的准确率预估是基于少量测试图片得出的结论，只代表在该测试集下的准确率。

Q：训练完的分类器预测的准确率不高是什么原因？

A：主要是训练数据的问题，包括： 1.某个模板的训练集中混杂了其他类型的图片； 2.训练集数据较少或过于单一没有很好覆盖全实际场景； 3.需要分类的不同模板区别不明显，如北京增值税专票和天津增值税专票 针对上述情况的解决方案如下：

- 1.检查每个模板的训练集，确保训练集中的图片属于同一个模板；
- 2.增加训练集中的图片数量，尽量覆盖实际使用时可能会遇到的场景；
- 3.只是文字内容级别的不同模板建议使用通用文字识别的结果作为参考进行分类。

错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
 "error_code": 110,
 "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	未知错误，请再次请求， 如果持续出现此类错误，请在控制台提交工单联系技术支持团队
2	Service temporarily unavailable	服务暂不可用，请再次请求， 如果持续出现此类错误，请在控制台提交工单联系技术支持团队
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额

4	Open api request limit reached	集群超限额
6	No permission to access data	无接口调用权限，创建应用时未勾选iOCR相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口后重新调用
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	免费测试资源使用完毕，每天请求量超限额，您可以在控制台文字识别服务选择购买iOCR相关接口的次数包或开通按量后付费
18	Open api qps request limit reached	QPS超限额，免费额度并发限制为2QPS，开通按量后付费或购买次数包后并发限制为10QPS，如您需要更多的并发量，可以选择购买QPS叠加包
19	Open api total request limit reached	请求总量超限额，您可以在控制台文字识别服务选择购买iOCR相关接口的次数包或开通按量后付费
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216102	service not support	请求了不支持的服务，请检查调用的url
216110	appid not exist	appid不存在，请确保调用接口所使用的应用归属于创建模板的百度云账号，且已开通该接口权限
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096x4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
272000	structure failed	未进行分类，未能匹配模板，请检查参照字段的设置是否符合规范，并重新选取或增加更多的参照字段
272001	classify failed	未能成功分类，请检查识别图片所属类别分类器中是否存在
272002	structure failed	分类成功，但未能匹配模板。请检查参照字段的设置是否符合规范，并重新选取或增加更多的参照字段
272003	unsuccessfully detected receipt	未能成功检测到票据，请检查识别图片是否为票据粘贴单
272004	classifier does not exist	分类器不存在
272005	keyword classification failed	关键词分类失败
282000	internal error	服务器内部错误，可能是图片尺寸过大文字太多识别超时，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
282004	invalid parameter, appId doesn't own this template nor not launch	您指定的模板暂未发布，请先发布该模板，再调用
282102	target detect error	未检测到图片中识别目标，请确保图片中包含对应卡证票据
282103	target recognize error	图片目标识别错误，请确保图片中包含对应卡证票据，如果持续出现此类错误，请在控制台提交工单联系技术支持团队

EasyDL OCR

产品介绍

功能介绍

EasyDL文字识别，可定制识别图片中的文字信息，结构化输出关键字段内容，极大提升OCR模型训练效率，满足个性化卡证票据识别需求

- **数据标注**

  创建数据集并上传真实图片，定义数据识别字段作为标注标签，在图片中框选对应的 Key/Value 内容区域，自动识别框选区域内容完成转写，标注人员对识别结果进行查验纠正即可完成标注

- **数据生成**

  基于已标注数据，将图中已框选 Value 区内容进行抹除，选择对应的字体、字号、颜色，并根据该字段的内容选择相匹配的语料库，即可完成虚拟数据生成底板的创建，并基于此底板生成任意张版式相同内容不同的虚拟数据，快速扩充数据集规模，结合真实数据一同用作模型训练集

- **模型训练与管理**

  支持根据使用场景需求创建多个的识别模型，选择包含已标注数据及虚拟数据的数据集进行训练，即可自动排队完成训练，同时输出预测准确率供参考；也可扩充数据集对现有模型进行迭代训练，产出新版本

- **服务部署**

  对训练完成的模型可上传真实数据进行模型校验，效果满意后即可一键发布上线，自动分配机器资源完成部署，并生成标准API接口供业务调用

特色优势

零门槛操作

提供一站式流程化训练，并预置最佳预训练模型及训练参数，无需算法基础、无需关注算法细节即可完成模型训练

高精度效果

基于百度丰富的商用模型实训经验，预置最佳实践产出的预训练模型，并基于百度自研的 EnDet 实体检测模型进行训练，模型平均准确率可达 90% 以上

低成本数据

提供可视化数据管理平台，对上传图片进行智能预标注，仅需核对修改即可完成标注，并可基于一张标注图片批量生成虚拟数据，快速扩充训练集，启动模型训练

超灵活部署

支持多种部署方式，公有云服务可一键部署，即刻生成 Restful API，毫秒级调用响应，高并发承载；同时，完整平台支持私有化部署，可用于搭建企业内部 AI 中台；也可支持产出模型容器化打包进行本地部署，快速完成项目交付

应用场景

- **证照电子化审批：**对政府、金融、企业等审批流程中涉及到的各种证照，如食品/药品经营许可证、特种设备审批证等，进行定制训练，快速提取关键信息完成线上审批，实现 7*24 小时无间断服务
- **财税报销电子化：**对不同金融或税务机构发行的各类财务发票、银行单据进行定制训练，快速实现财税凭证的录入，大幅度节约凭证邮寄、录入成本，实现线上电子化报税报销
- **保险智能理赔：**对不同版式的保单或不同地区、不同医疗系统开具的医疗票据进行定制训练，实现保险理赔相关材料的快速录入，降低人力成本，提升保险理赔的业务安全性及快捷性

API文档

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/rest/2.0/ai_custom/v1/ocr`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url 二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和 image 二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效，不支持https的图片链接。当image字段存在时 url 字段失效
modelId	是	string	-	模型 ID，自训练产出模型的唯一标示，可用于调用指定的已发布模型进行结构化识别，可在「我的模型」页查看并复制使用
detect_direction	是	string	true/false	是否开启图像方向矫正功能，可选值有： - **true**：开启图像方向矫正功能，可自动矫正不同旋转角度的图片进行识别； - **false**：关闭图像矫正功能，如要识别的内容均为正向图片，建议可关闭此功能避免误矫正。

返回说明

返回参数

字段	类型	说明
log_id	int	调用请求的唯一日志id，如需技术支持进行问题排查请反馈此id以快速进行问题定位
error_code	int	0代表成功（一般不返回），如果有错误码返回可以参考错误码列表排查问题
error_msg	string	如果error_code具体的失败信息，可以参考下方错误码列表排查问题
result	Object	识别返回的结果，每一个key代表识别字段名称，对应的value为该字段的识别结果
+ “key”	Array	识别内容，“key”为数据标注时创建的字段名称，将不同版式的内容进行归一化输出
++ probability	Object	字段的置信度，包括平均和最小置信度
+++ average	int	字段的平均置信度
+++ min	int	字段的最小置信度
++ location	Object	字段在原图上对应的矩形框位置
+++ top	int	字段文本框左上角点的上边距
+++ left	int	字段文本框左上角点的左边距
+++ width	int	字段文本框的宽度
+++ height	int	字段文本框的高度
++ word	string	字段识别结果

返回示例

```
{
  "log_id":"161648117706127",
  "result":{
    "收款卡号":[
      {
        "probability":{
          "average":0.9972677230834961,
          "min":0.6964160077398716
        },
        "location":{
          "top":404,
          "left":451,
          "width":303,
          "height":61
        },
        "word":"119086830765501"
      }
    ],
    "日期":[
      {
        "probability":{
          "average":0,
          "min":0
        },
        "location":{
          "top":-1,
          "left":-1,
          "width":0,
          "height":0
        },
        "word":"20180330"
      }
    ]
  }
}
```


错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error_code**：错误码。
- **error_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：


```
{
 "error_code": 110,
 "error_msg": "Access token invalid or no longer valid"
}
```


需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求， 如果持续出现此类错误，请在控制台提交工单联系技术支持团队
2	Service temporarily unavailable	服务暂不可用，请再次请求， 如果持续出现此类错误，请在控制台提交工单联系技术支持团队
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天请求量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216102	service not support	请求了不支持的服务，请检查调用的url
216110	appid not exist	appid不存在，请确保调用接口所使用的应用归属于创建模型的百度云账号，且已开通该接口权限
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
282000	internal error	服务器内部错误，可能是图片尺寸过大文字太多识别超时，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
336100	model temporarily unavailable	模型长期未调用需激活，遇到该错误码请等待5分钟后再次请求，即可恢复正常，若反复重试依然报错或有疑问请在百度智能云控制台内提交工单反馈

私有化部署服务

产品介绍

简介

OCR 私有化部署服务支持将 OCR 识别模型部署至本地服务器或私有云环境，为企业提供 **高隐私性** 和 **强实时性** 的内网 API 服务，可供局域网内的设备进行调用。**70+** 款标准模型及自定义平台可供选择，可满足 **手写/印刷文本、各类卡证票据** 等不

同类型文字识别需求，同时可提供 **自定义/自训练平台** 的本地化部署，助您快速搭建企业 AI 中台。

所有模型及平台均可提供 **容器化软件部署包**，可部署于企业的本地服务器或私有云上，CPU/GPU 环境均可部署，主流显卡均可兼容。同时，支持单机部署、多机部署、集群部署等方式，并提供一键部署工具，可快速安装运行环境、容器及模型服务，最快半小时即可完成安装部署。

部署后可以实现与对应云端 API 能力相同的功能及效果（参数有少量区别，具体请参考相应技术方向的接口文档），适用于政务审批、金融、保险、财务等 **数据需与公网隔离** 的场景，以及交通车牌识别等 **高并发高实时性要求** 的场景。

申请使用前也**请提前准备好鉴权物理机和服务器资源**，这是您运行 OCR 应用服务的基础，详细硬件推荐请点击[了解详情]。同时，强烈建议您**调用对应云端接口进行初步测试，确认效果及功能基本可满足需求后再发起私有化申请**，以免耽误您的部署时间。

> 视频教程请参见：OCR 私有化部署操作教程

方案优势

- **纯离线：**满足无网、弱网、专网等多种网络需求，满足内部数据与公网隔离的私密性需求
- **纯软件：**不与硬件进行捆绑，价格优惠，无需硬件入场审批，充分利用现有机器资源，且可以快速测试、快速交付
- **能力全面：**70+ 款标准能力及 2 款自定义/自训练平台的均可支持私有化部署，满足不同业务场景的文档、卡证、票据识别需求
- **识别效果领先：**基于业界领先的深度学习技术，提供多场景、多语种、高精度的整图文字检测和识别服务，多项 ICDAR 指标居世界第一
- **毫秒级响应：**具备高并发、高吞吐、低时延等能力，且算法卓越，性能强劲，识别速度业界领先，可应对各种实时性业务需求
- **选型灵活：**可自由选择不同 QPS 配置，灵活适应各种业务量级的调用需求
- **服务专业：**首次部署可提供专属交流群进行远程部署及答疑支持，正式购买后一年内提供免费线上技术支持及模型更新服务

常用能力列表

模型名称	模型功能	支持部署环境	对应云端能力
通用场景文字识别			
通用文字识别	识别图片中的文字信息及文字区域的坐标信息，支持中文简体、中文繁体、英文识别	CPU/GPU	通用文字识别
通用文字识别（多语种版）	识别图片中的文字信息及文字区域的坐标信息，支持 20 种语种识别，包含 中英文混合、英文、日语、韩语、法语、西班牙语、葡萄牙语、德语、意大利语、俄语、丹麦语、荷兰语、马来语、瑞典语、印尼语、波兰语、罗马尼亚语、土耳其语、希腊语、匈牙利语	GPU	通用文字识别
网络图片文字识别	针对网络图片进行专项优化，对艺术字体或背景复杂的文字内容具有更优的识别效果	GPU	网络图片文字识别
办公文档识别	支持对办公类文档版面进行分析，输出图、表、标题、文本的位置，及各版块识别结果，支持中、英两种语言的手写、印刷体混排场景	GPU	办公文档识别
手写文字识别	对手写汉字、英文、数字内容进行识别	GPU	手写文字识别

表格文字识别	对单据或报表中的表格内容进行结构化识别，识别结果默认以 JSON 形式返回，如需返回 Excel，可提供调用脚本进行结果转化	CPU/GPU	表格文字识别（同步接口
卡证文字识别			
身份证识别	对中国大陆二代居民身份证正反面所有8个字段进行结构化识别，支持身份证正反面自动检测及多身份证混贴识别	CPU/GPU	身份证识别
银行卡识别	对银行卡的卡号、有效期进行结构化识别	CPU/GPU	银行卡识别
营业执照识别	对各类版式的营业执照的全部字段进行结构化识别，包含证件编号、社会信用代码、单位名称、地址、法人、类型、成立日期、有效日期、经营范围	CPU/GPU	营业执照识别
护照识别	对中国大陆居民护照的资料页的全部 15 个字段进行结构化识别，包含国家码、姓名、性别、护照号、出生日期、签发日期、有效期至、签发地点等	CPU/GPU	护照识别
户口本识别	对户口本内常住人口登记卡的全部 22 个字段进行结构化识别，包含出生地、出生日期、姓名、民族、与户主关系、性别、身份证号码等	CPU/GPU	户口本识别
出生医学证明识别	对出生证明的全部 23 个字段进行结构化识别，包含出生时间、姓名、性别、出生证编号、父亲姓名、母亲姓名等	GPU	出生医学证明识别
财务票据文字识别			
混贴票据识别	对粘贴在同一张A4纸上的多张不同种类票据进行自动切分并识别，可返回每张票据的位置、种类及票面的结构化识别结果。已支持增值税发票、定额发票、卷票、火车票、出租车票、行程单、机动车销售发票、汽车票、通行费发票、购车发票、二手车发票、网约车行程单、船票等 13 类票据的混贴识别	CPU/GPU	混贴票据识别
银行回单识别	对各大银行的收/付款人户名、账号、开户银行、金额、日期等关键字段进行结构化识别	CPU	银行回单识别
增值税发票识别	对增值税普通发票、专用发票、电子发票的全部关键字段进行结构化识别，发票基本信息、销售方及购买方信息、商品信息、价税信息等；增值税卷票识别需部署单独模型	CPU/GPU	增值税发票识别
增值税卷票识别	对增值税普通卷票的全部关键字段进行结构化识别，发票基本信息、销售方及购买方信息、商品信息、价税信息等	CPU/GPU	增值税发票识别
火车票识别	支持对红、蓝火车票的 13 个关键字段进行结构化识别，包括车票号码、始发站、目的站、车次、日期、票价、席别、姓名、座位号、身份证号、售站、序列号、时间	CPU/GPU	火车票识别
出租车票识别	支持识别全国各大城市出租车票的 16 个关键字段，包括发票号码、代码、车号、日期、总金额、燃油附加费、叫车服务费、省、市、单价、里程、上车时间、下车时间等	CPU/GPU	出租车票识别
飞机行程单识别	支持对飞机行程单的 24 个字段进行结构化识别，包括电子客票号、姓名、始发站、目的站、航班号、日期、时间、票价、身份证号、金额、客票级别、座位等级等。 同时，支持单张行程单上的多航班信息识别	CPU/GPU	飞机行程单识别
汽车场景文字识别			
车牌识	支持识别中国大陆机动车蓝牌、黄牌（单双行）、绿牌、大型新能源（黄绿）、领使馆车牌、	CPU/G	车牌识

别	警牌、武警牌（单双行）、车牌（单双行）、港澳出入境车牌、农用车牌、民航车牌的地域编号和车牌号。同时，支持识别图像中的多张车牌，并可提供高拍场景专用识别模型	PU	别
驾驶证识别	支持对机动车驾驶证正页及副页所有15个字段进行结构化识别，包括证号、姓名、性别、国籍、住址、出生日期、初次领证日期、准驾车型、有效期限、发证单位、档案编号等	CPU/GPU	驾驶证识别
行驶证识别	对机动车行驶证主页及副页所有22个字段进行结构化识别，包括号牌号码、车辆类型、所有人、品牌型号、车辆识别代码、发动机号码、核定载人数、质量、尺寸、检验记录等	CPU/GPU	行驶证识别
机动车销售发票识别	支持对机动车销售发票的26个关键字段进行结构化识别，包括发票代码、发票号码、开票日期、购买方信息、销售方信息、车辆信息、价格、税额等	CPU	机动车销售发票
医疗票据文字识别			
医疗发票识别	支持识别全国各地门诊/住院发票的业务流水号、发票号、住院号、病例号、姓名、性别、社保卡号、金额大/小写等关键字段，其中北京/广东/河北/河南/江苏/山东/上海/天津/浙江等地区票据识别效果较佳；支持识别收费项目明细，并可根据不同省市地区返回对应的识别参数	GPU	医疗发票识别
医疗费用结算单识别	支持识别全国医疗费用结算单的姓名、出/入院时间、发票总金额、自费金额、医保支付金额等6 个关键字段，其中北京地区票据识别效果最佳	GPU	医疗费用结算单识别
病案首页识别	支持识别全国各地病案首页的病案号、姓名、性别、出生日期、身份证号、出/入院科别、住院次数、药物过敏情况等 15 个关键字段，其中北京地区票据识别效果最佳	GPU	病案首页识别
医疗费用明细识别	支持识别全国医疗费用明细的姓名、日期、病人ID、总金额等关键字段，支持识别费用明细项目清单，其中北京地区识别效果最佳	GPU	医疗费用明细识别
教育场景文字识别			
试卷分析与识别	可对作业、试卷的版面进行分析，输出图、表、标题、文本的位置，并输出分版块内容的OCR识别结果,支持中、英两种语言的手写、印刷体混排场景	GPU	试卷分析与识别

服务授权

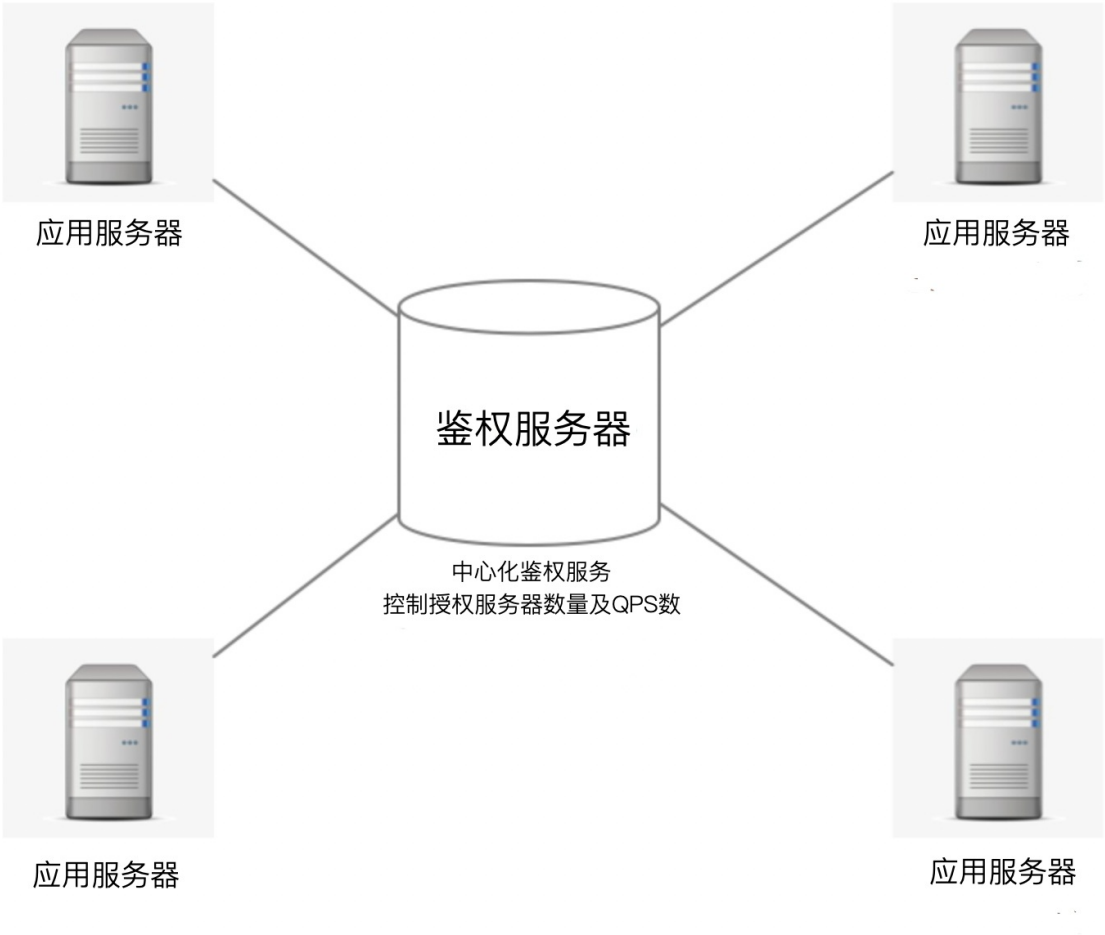
1、授权方式

私有化部署需获取百度授权后方可使用，百度授权 License 需绑定指定服务器，因此需要您在规划用于部署授权服务的物理机/虚拟机上提取机器指纹，并在申请 OCR 私有化部署包时进行上传方可提交模型打包申请，**指纹提取工具**的下载和使用可点击查看详情。

百度授权 License 需根据业务需求量进行授权，您可根据实际业务中 **每秒需识别的图片数量** 申请或购买相应 QPS 的服务授权，如服务部署在多台应用服务器上，各应用服务器均分总授权 QPS 数。

鉴权服务与应用服务可部署在同一机器上，也可分离部署，但需保证被采集了指纹的硬件不被替换，则在 License 允许的实例数范围内，可保证多台连通的服务器运行 OCR 模型，**在鉴权服务器不变的情况下可更换应用服务器**。

> 注：如鉴权服务部署在虚拟机上，也需保证对应的物理资源不会被动态调整。



2、授权有效期

如您已通过企业认证，即可申请所需 OCR 服务**一个月的免费试用期**，在 OCR 控制台 — 私有部署管理 页面发起申请即可。

如您需正式购买，则可选择 **按年授权** 或 **永久授权** 两种方式：

- **按年授权**：自私有化部署包交付日起 365 天内有效；
- **永久授权**：自私有化部署包交付日起至 2099 年 12 月 31 日有效。

部署流程

部署前环境检查（必看）

本文档介绍了鉴权服务及文字识别（OCR）服务部署的硬件、网络、及软件环境要求，请您在部署前**务必**参考此文档进行硬件、网络、及软件环境检查，以避免在安装部署过程中出现问题。

硬件环境要求

若您仍然不确定硬件选型，请提交工单联系百度的工作人员

鉴权服务节点

名称	推荐	说明
服务器	物理机、独享云服务器、虚拟机。	CPU，网卡MAC，内存，显卡、磁盘等发生变化会影响指纹，进而影响鉴权。如果是共享云服务器，机器指纹可能会发生变化。如果您不得不使用云服务器，建议您使用独享云服务器。
CPU	AMD/Intel x86_64，核数8核（推荐）	指令集必须支持bmi2和AVX ,可以通过该命令检查：lscpu | grep -E 'avx|bmi2'

应用服务节点

名称	推荐	说明
服务器	物理机、云服务器、虚拟机。	
linux桌面环境	部署GPU版本模型，需确保系统禁用linux桌面环境（包括但不限于"lightdm", "gdm", "kdm" 等）	如未提前关闭，部署期间linux桌面环境将不可用
CPU	AMD/Intel x86_64，核数8核（推荐）	指令集必须支持bmi2和AVX,可以通过该命令检查：lscpu | grep -E 'avx|bmi2'
GPU	目前支持型号包括： T4, A10, A100, A800, RTX2080, RTX3060, RTX3070, RTX3080, RTX3090 等 Turing/Ampere 架构 NVIDIA 显卡	显存>10G
硬盘	测试建议>50G,生产建议>500G	
内存	建议大于16G	如果模型较多，需要增加内存。

网络环境要求

1、为了缩减包体积，控制台为您提供的私有化部署包为精简包，未包含应用镜像等必要文件。如果您的服务器可以联通外网，可以在服务器侧执行 `bash download.sh` 下载完整的安装包；如果不可联通外网，您可以在办公电脑提前下载好完整的部署包后，上传至服务器端。

2、当应用服务和鉴权服务分离部署时，鉴权服务器和应用服务器之前需相互开放8443端口；如果鉴权服务器有多台的话，鉴权服务器之间需相互开放 8443，7091，7092端口

3、静态IP地址已分配。假如网卡为eth0, 可以通过`more /etc/sysconfig/network-scripts/ifcfg-eth0 |grep "BOOTPROTO="` 命令查看，如果返回 `BOOTPROTO=DHCP`,表示动态ip

软件环境要求

名称	说明
操作系统	Centos 7/8、Ubuntu 14/16/18/20、 Redhat 7.2、 Suse 12 可以进入python解释器执行命令查询 import platform print(platform.dist())
Linux 内核	>=3.10，可以通过uname -r命令查看
SELinux	确保系统禁用SELinux ，可以通过getenforce命令查询，返回Disabled 表示禁用
CPU指令集	支持avx/avx2指令集以及bmi2指令集（必要条件）
Python	==2.7.5
curl	确保机器存在curl命令
GLIBC	>=2.17，可以通过命令 ldd --version 查看
GLIBCXX	>=3.4.19 查询命令： ubuntu下执行 strings /usr/lib/x86_64-linux-gnu/libstdc++.so.* |grep GLIBCXX 其他操作系统执行 strings /usr/lib64/libstdc++.so.*|grep LIBCXX
GCC	>=4.8.2,可以通过 gcc --version 查看
root权限	需要批准可使用root权限用户来部署
Docker	系统是否自带Docker，如自带，需将docker版本号信息提供给技术支持同学评估
nvidia 驱动	系统是否已经安装nvidia驱动，可通过nvidia-smi查看，如已安装，请将该信息反馈给技术支持同学
nvidia-docker	系统是否已经安装nvidia-docker，可通过nvidia-docker version查看，如已安装，请将该信息反馈给技术支持同学

机器指纹提取

指纹提取工具获取

在私有化部署包的申请页面下载指纹提取工具后，参考此文档进行指纹提取操作。点击下载指纹提取工具 **注意**

- 上传指纹文件后才能发起私有化部署包申请。
- 单机部署，在需要部署的机器上提取指纹申请部署包即可。
- 集群部署，需要3或5（需要奇数机器并且>=3）台机器的指纹打包成一个tar的压缩包申请部署包即可（具体提取方式可参考指纹提取部分）。**注意：部署时需传入所有指纹对应机器的IP地址，并以逗号分隔；为保障鉴权服务正常运行，请确保网络互通且已在半数以上的机器中部署鉴权服务。**
- **建议以root用户运行指纹采集工具。注意：运行工具用户需和鉴权服务运行用户保持一致**（鉴权服务以及部署脚本运行用户默认为root，如有非root运行需求，请提交工单联系百度的工作人员）

指纹采集环境要求

硬件要求（物理机）

- CPU架构:AMD 64/x86_64，支持avx/avx2、bmi2指令集；如有ARM架构需求，请提交工单联系百度的工作人员
- 内存:>=32G（推荐,不强制）
- 硬盘:>=512G （推荐）
- 网络环境:机器需要在局域网内，且能获取到ip地址

操作系统要求

- 基于安全的角度考虑，虚拟机部署需与百度工作人员单独沟通申请，包括但不限于Virtual Box、VMware等。
- 支持的Linux发行版列表 *Ubuntu 14*、*16*、*18*；CentOS 7系列及 8.0； *RedHat 7.2*； SUSE 12
- Linux内核要求 >=3.10
- GLIBC >=2.17
- GLIBCXX >=3.4.19
- Python 2.7

指纹采集操作流程

1、服务的部署需要一台物理机做鉴权server，我们需要提取的就是这台机器的指纹，将指纹工具上传到服务器，然后使用unzip {文件名}命令解压工具包；

2、进入解压目录，执行chmod +x get_machine_finger_en && chmod +x phosphor 命令增加执行权限；

3、用root用户执行 ./get_machine_finger_en 命令提取指纹，命令执行结果在在/tmp目录下，文件名为secfile_xxx格式的文件即为指纹文件；

4、单指纹文件不必进行压缩，可直接上传提交申请。

5、两个及两个以上的指纹文件，需要将所有的指纹文件存档到一个文件目录下，该目录下只能有指纹文件；创建压缩包，进入指纹文件存储的目录，执行命令 tar cvzf ../finger.tar.gz ./*，在上一级目录生成了名字为finger.tar.gz的压缩包，注意检查该压缩包的目录结构，解压后直接就是指纹文件，没有多余的文件目录；

指纹采集常见问题说明

1、执行命令提取指纹后，出现报错提示“/dev/mapper/centos-root doesn't seem to be an new sg device”，拿不到硬盘序列号。

解决方案：检查/tmp文件夹下是否生成指纹文件，如已生成指纹文件，则忽略该报错。如未生成指纹文件，请发起工单处理。

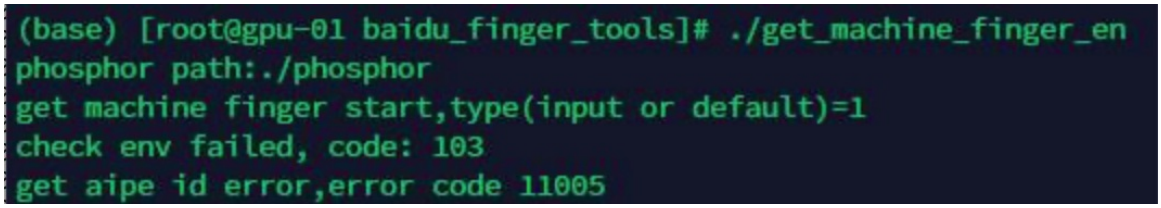
2、指纹提取工具必须使用root用户运行吗？

解决方案：建议以root用户运行。指纹工具运行用户和鉴权服务运行用户必须保持一致，鉴权服务运行用户默认root。

3、指纹提取时包报错："/usr/lib64/libstdc++.so.6: version 'GLIBCXX_3.4.14' not found"

解决方案：需要升级GLIBC到3.4.14版本或者更高。

4、虚拟机采集指纹报错，截图如下：



解决方案：原因是phosphor文件没有执行权限，执行chmod +x get_machine_finger_en && chmod +x phosphor命令增加执行权限；再执行./get_machine_finger_en采集指纹就可以了。

基于docker容器化部署说明

标准模型部署

一、资源规划

资源规划文档旨在系统资源使用层对真实交付场景做产品层面的通用指导

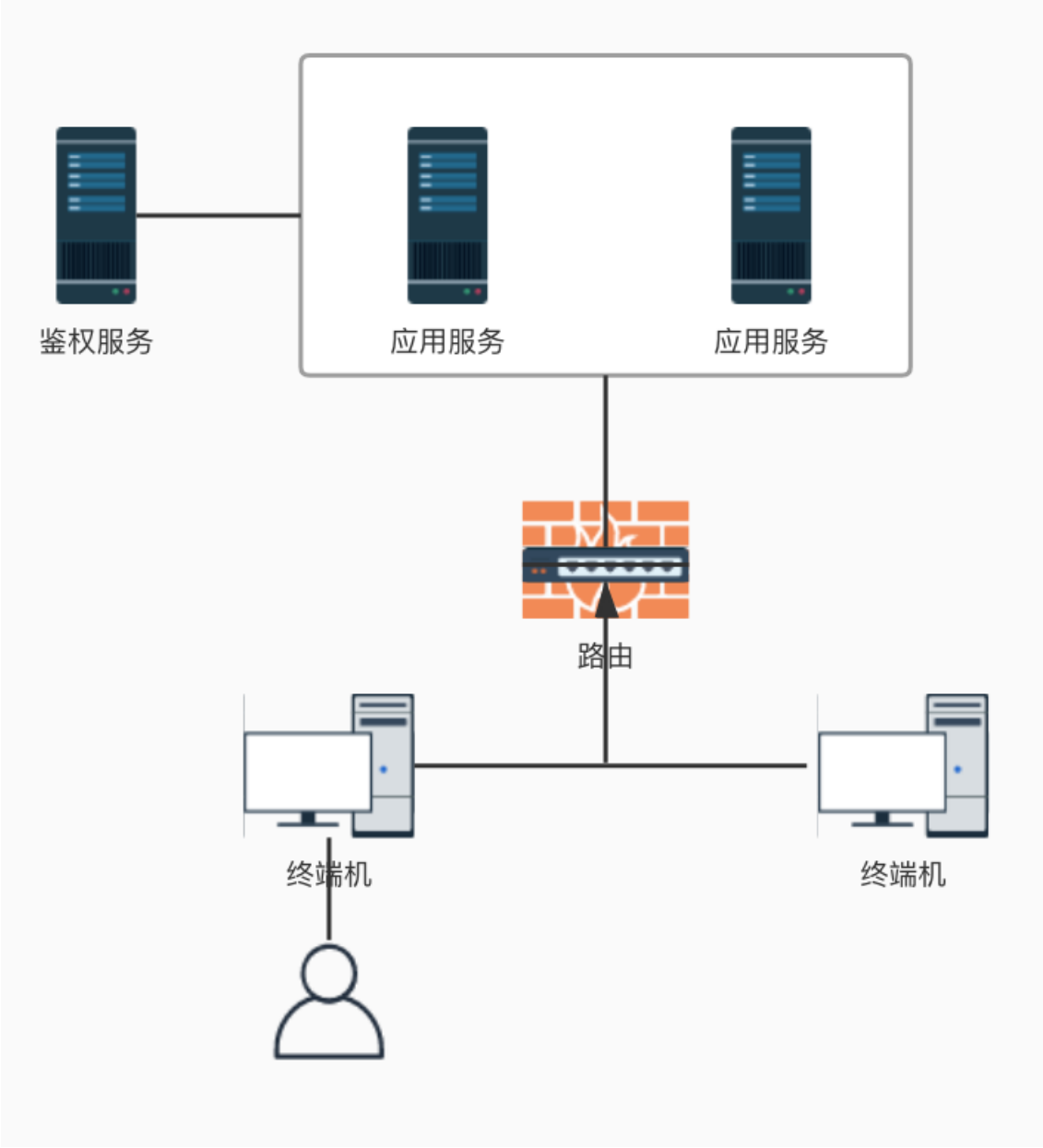
文档涵盖了构建百度OCR算子服务所需要的服务器主机、 CPU、内存、 GPU、网络等内容方面的规划。 本文档所列出的所有数量数字方面的值均需针对实际场景需求和限制做针对性调整。

名词定义

名称	含义
鉴权服务	鉴权服务包含百度发布的服务授权证书，如果不安装鉴权服务，后续的应用服务也将无法启动，目前支持加密狗（硬件）和提取机器指纹（软件）两种鉴权方式
应用服务	包含Docker等基础环境以及相关技术方向的算法模型，是私有化产品的的核心。部署应用服务的前提是部署鉴权服务，应用服务在运行时会实时请求鉴权服务，需要保障两个服务之间能够顺利通信
单机一键部署	适用于鉴权服务、应用服务部署在一台物理机上的场景。即执行一条命令将鉴权服务、应用服务安装完成。
多机分离部署	适用于鉴权服务、应用服务不在一台机器上部署的场景。首先部署鉴权服务、然后部署应用服务
测试环境	客户方提供的进行产品、场景测试的环境，一般与生产环境隔离，开发、测试能够方便接触到的环境，对部署的可用性要求较低
生产环境	客户的生产环境，一般的业务系统真实提供服务的环境，具备较高的可用性要求，包括异地多活的灾备要求。一般只有运维人员有该环境的超管权限

物理架构拓扑图 文字识别私有化部署产品包含鉴权服务和应用服务，其中

- 鉴权服务通过客户网关系统连接到应用服务器为应用服务提供鉴权认证
- 应用服务部分直接或通过生产级网关被客户业务场景直接使用



鉴权应用

规划原则：

鉴权服务是运行百度文字识别应用服务的基础，如果鉴权异常，将直接导致模型的API接口不可用。鉴权服务的高可用通过Raft算法实现。 鉴于Raft算法的特性，我们建议您选择“奇数”个节点来部署鉴权集群。

鉴权集群不同节点的容错能力如下。

集群节点数量	最大容错
1	0
2	0
3	1
4	1
5	2
6	2

规划流程：

一般建议鉴权节点数量为1或3个。

模型算子应用

规划原则：

- 单模型应用实例承担流量，根据业务逻辑复杂度（请求报文）、机器节点硬件条件（CPU、内存、网络、显存）强相关
- 单模型应用实例的内存分配依赖模型的大小
- 单模型应用实例的CPU资源与请求报文相关
- 单模型应用实例占用网路IO与请求报文和返回结果相关
- 单模型应用实例占用磁盘空间大小与请求报文、日志输出量等特性相关
- 场景需要的模型应用服务资源与整体业务QPS、单模型实例性能、高可用方案相关

规划流程：

根据实际场景进行性能测试，得出单模型实例性能指标（QPS、响应延时、内存占用、显存占用），结合场景高峰流量预估和高可用要求，以及服务器实际显卡数量等计算需要模型应用实例数，根据机器节点硬件资源指标，最终确定硬件节点数。

资源规划示例

模块清单

模块名	**模块角色说明**
openresty	基础依赖服务，提供负载均衡能力
docker	基础依赖服务
nvidia	基础依赖服务
redis	基础依赖服务
……	基础依赖服务
c-offline-security-server	鉴权服务
auth-manage-service	鉴权服务，用于鉴权管理，提供web页面查看鉴权信息。
ocr-x x x x x	模型应用服务
……	模型应用服务

以上模块清单针对不同的模型部署包稍有差异，可以通过下面的命令查询当前部署包内详细的模块清单。

```
###### se :表示search,根据模块名称查询安装包里是否包括某个模块,不区分大小写,如果没有指定名称则输出安装包里所有模块信息
cd origin/package/Install && python install.py se
```

您可以根据模型的数量适当调整cpu，内存，显存，存储等资源，下面以在单台服务器部署1个模型为例说明：

资源需求	节点	部署模块	单节点CPU	单节点内存	单节点显卡数	单节点存储	单节点网络
测试环境	节点A	基础依赖服务	4核	8G	视模型情况而定	50G	>1000Mbps
		鉴权服务和鉴权管理服务					
		模型应用服务					
生产环境	节点A	基础依赖服务	8核	16G	视模型情况而定	500G	>1000Mbps
		模型应用服务					
	节点B	鉴权服务	1核	2G	不需要	10G	>1000Mbps
生产环境	节点A	鉴权服务	8核	16G	视模型情况而定	500G	>1000Mbps
		基础依赖服务					
		模型应用服务					
	节点B	基础依赖服务	8核	16G	视模型情况而定	500G	>1000Mbps
		模型应用服务					

以上资源数值为推荐参考值，可根据实际情况做测试和调整。

如果您有多个服务器节点，可以参考以上示例，选择每台服务器节点需要部署的服务。

二、场景与名词

场景说明

- （无环境）全新部署：服务器环境为第一次部署，该服务器之前没有部署过百度文字识别产品的任何模型。
- （有环境）升级模型：是指用户之前部署过老版本的百度文字识别产品模型，需要对模型进行升级操作。
- （有环境）新增模型：是指用户当前服务器或者其他服务器已经部署过百度文字识别产品部分模型，需要新增其他模型
- 回滚：回滚到最近一次升级前的版本
- 卸载： 删除已经安装的模块

名词解释

名词	说明	示例
package_dir	存放部署包、升级包的路径,包体积较大，尽量不要放在/下	如 /mnt/disk0/baidu_ocr_install_20111009
work_dir	应用程序文件存储地址，默认为/home/baidu/work	如 /home/baidu/work

三 、准备工作

环境检查

部署之前请**务必**参考此文档部署前环境检查必看进行硬件、网络、及软件环境检查，以避免在安装部署过程中出现问题。

获取部署包

1、申请正式模型部署包安装文件下载链接，下载模型部署包并重命名。

```
###### 示例如下，-O --output-document=FILE  对文件重名名，O为大写英文字母
###### 请将示例中的9C20XXXXXXXX.tar.gz替换为真实的文件名
wget -O 9C20XXXXXXXX.tar.gz https://bj.bcebos.com/v1/private-ai-online/9C20XXXXXXX.tar.gz?authorization=bce-authXXXXXXX132187fcf81
```

2、将9C20XXXXXXXX.tar.gz上传到待部署的服务器，为方便区分不同的部署包，建议以【*baidu_ocr_install + 日期*】命名，如 baidu_ocr_install_20111009，该目录我们将其命名为package_dir

3、进入package_dir 执行以下命令解压部署包

```
###### baidu_ocr_install_20111009 部署包的根目录，次处仅为示例，实际操作时换成您上一步命令的路径
cd baidu_ocr_install_20111009 && tar zxvf 9C20XXXXXXXX.tar.gz
```

4、解压后进入original目录执行`bash download.sh`命令获取全部安装文件，执行脚本后会自动下载以下安装文件：数据库服务安装包、鉴权服务安装包、应用服务安装包以及docker安装包等基础依赖环境。如果已经提前在办公网络下载完毕，请忽略该步骤。

```
cd original && bash download.sh
```

执行结束后，会在download.sh同级目录下生成download.log日志记录下载详情。

若您在此过程出现问题，请提交工单联系百度的工作人员

接下来您可以根据实际使用场景和前文的资源规划，从下文选择一种部署方式来部署。

四、全新部署

指服务器环境为第一次部署，该服务器之前没有部署过任何百度文字识别产品的任何模型。

您可以进入模型部署包的存储路径，执行下面的命令查看程序help信息：

```
cd original/package/Install
python install.py
```

返回结果如下：

```
 install.py usage:
inall: 安装所有的产品以及鉴权服务和基础服务,适用于在单台物理机上安装所有模块的场景
in, install: 安装一个模块,名称不区分大小写,适用于产品模块和鉴权服务分机器部署的场景
se, search: 根据模块名称查询安装包里是否包括某个模块,不区分大小写,如果没有指定名称则输出安装包里所有模块信息
li, list: 根据模块名称查询某个模块是否已经安装,如果没有指定名称则输出所有已经安装的模块
rm, remove: 根据模块名称删除某个已经安装的模块,如果有其他模块依赖这个模块，则不允许删除
rmall: 删除所有已经安装的模块
lu, licenseupdate: 更新license文件，适用于授权延期、实例数扩容、增加产品授权
up, upgrade: 升级指定模块，不区分大小写，不指定参数时输出所有可升级模块信息
rb, rollback: 回滚指定模块，不区分大小写，回滚到最近一次升级前的版本
du, safestoredataupdate: 安全存储数据更新，包括敏感数据、模型解密密钥等相关文件的更新
```

请根据前文的资源规划和实际的使用场景从【单机一键部署】和【多机分离部署】选择一种方式部署。

单机一键部署

如果您只有一台服务器，可以使用`inall`参数来部署,`inall` 表示`install all`，安装所有的产品以及鉴权服务和基础服务,适用于在单台服务器上安装所有模块的场景，容易和 `install` 混淆，在使用时请注意。

1、使用root权限启动一键部署脚本进行安装：

```
###### inall: 安装所有的产品以及鉴权服务和基础服务,适用于在单台物理机上安装所有模块的场景
python install.py inall
```

执行安装后，首先进行环境检查

```
 未通过的检查项详情如下：

+--------------------------+------------+-------------+----------+-------------------+
|          模块            |   检查项   |    含义     | 指标要求 | 实际参数或报错信息 |
+--------------------------+------------+-------------+----------+-------------------+
|         default          | * disk_home | HOME磁盘空间 | >=512 GB |       201 GB       |
| c-offline-security-server |   machine   |  宿主机环境  |          |  当前环境为虚拟机  |
+--------------------------+------------+-------------+----------+-------------------+
带*的检查项非强制，可以选择跳过
2021-08-27 21:44:37,414 - 2712 - install - ERROR - 环境检查失败！请修复未通过的检查项后重新安装，如有问题请联系技术支持。
按任意键结束。输入continue后回车可跳过环境检查，跳过后不保证安装成功。
```

其中带 * 项 （如 disk_home等）非强制要求，可以忽略。确认无误后请输入 **continue** 英文字符，继续下一步

之后会要求手动输入IP地址和选用的显卡编号,示例如下：

```
 请输入鉴权集群的IP列表，逗号分隔，列入：192.168.1.101
###### 参考上文资源规划，如鉴权节点唯一，只需要输入真实机器IP即可，如106.12.141.217
###### 如果鉴权节点有多个，请以英文逗号分隔，如106.12.141.217,106.12.141.218,106.12.141.219
106.12.141.217
```

```
 ###### 输入您要使用的显卡编号，仅输入1个。可通过nvidia-smi查看显卡编号
enter value for gpu index numbers used by this application,separated by comma, e.g. 0,1,2:
```

注：建议填写docker0网卡的ip地址，如后续服务器IP地址发生变化，不会影响服务。

2、确认本次安装模块是否完整：

执行`docker ps -a | grep baidu`查看相关容器是否包含以下且状态均正常

以【通用文字识别模型（CPU）版本】为例：

```
[root@instance-wch0lkwp Install]# docker ps -a|grep baidu
CONTAINER ID        IMAGE                                                    COMMAND                  CREATED             STATUS
PORTS                    NAMES
c1643ee21c44        registry.baidu.com/aipe/openresty:1.11.2.3-trusty                    "/usr/local/openre..."   40
seconds ago      Restarting (1) 6 seconds ago                              nginx-1
c766cdbbdb73        iregistry.baidu-int.com/aipe/public/centos:cuda10.0-cudnn7-c7-gcc8-ocr   "sh start.sh"            40
seconds ago      Up 3 seconds                0.0.0.0:8138->8256/tcp   ocr-finance-gpu-1
```

如果只有一个模型应用实例，则不需要部署负载均衡服务，可以执行以下命令移除nginx容器

```
docker rm -f nginx-1
```

▶ 如何单独部署某个模块？

接下来请直接进入【九、验证服务可用性】章节。

多机分离部署

如果您有多台服务器需要部署服务，可以参考第一章节的资源规划示例，尝试规划每台服务器需要部署的服务类型

1、查询当前部署包内包含的模块清单

```
###### se :表示search,根据模块名称查询安装包里是否包括某个模块,不区分大小写,如果没有指定名称则输出安装包里所有模块信息
python install.py se
```

2、选择您要安装的模块部署

```
###### in :install, 安装某个模块,名称不区分大小写
python install.py in 模型名
```

接下来请直接进入【九、验证服务可用性】章节。

五、升级模型

升级模型部署，是指您之前部署过老版本的百度文字识别产品模型，需要对模型进行升级操作。

方法一：

1、首先查询新的部署包内包含的模块清单

```
###### se :表示search,根据模块名称查询安装包里是否包括某个模块,不区分大小写,如果没有指定名称则输出安装包里所有模块信息
python install.py se
```

2、选择您要升级的模型模块并更新授权证书

```
###### up, upgrade: 升级指定模块，不区分大小写，不指定参数时输出所有可升级模块信息
python install.py up 模块名（这里指模型）

###### lu, licenseupdate: 更新license文件，适用于授权延期、实例数扩容、增加产品授权
python install.py lu
```

方法二：

进入旧模型部署包（可以参考上文约束的命令规范：baidu_ocr_install_日期找到历史部署包）卸载当前版本的模型，之后进入新申请的部署包选择要升级的模型安装。

示例：

```
ll
**返回结果如下：**
###### drwxr-xr-x 2 root root 4096 10月 26 16:56 baidu_ocr_install_20211025
###### drwxr-xr-x 2 root root 4096 10月 26 16:56 baidu_ocr_install_20211026

###### 进入旧部署包
cd baidu_ocr_install_20211025/original/package/Install
###### 查看当前部署包内置的模块
python install.py se
###### 卸载指定模型
python install.py rm 模型模块名

###### 进入新申请的部署包
cd baidu_ocr_install_20211026/original/package/Install
###### 查看当前部署包内置的模块，找到该模型的模块
python install.py se
###### 安装同名模型
python install.py in 该模型的模块名
```

接下来请直接进入【九、验证服务可用性】章节。

六、新增模型

1、查询新的部署包内包含的模块清单，依次部署所需要的依赖

```
###### 进入新申请的部署包
cd original/package/Install

###### se :表示search,根据模块名称查询安装包里是否包括某个模块,不区分大小写,如果没有指定名称则输出安装包里所有模块信息
python install.py se

###### in :install, 安装某个模块,名称不区分大小写
python install.py in docker
```

如果有其他依赖需要安装的话（如您需要使用GPU等，需要额外安装nvidia模块），步骤同上。此处不再提供示例。

2、部署指定模型模块并更新授权证书

```
# in :install, 安装某个模块,名称不区分大小写
python install.py in 模型名(模型模块)

# lu, licenseupdate: 更新license文件，适用于授权延期、实例数扩容、增加产品授权
python install.py lu
```

接下来请直接进入【九、验证服务可用性】章节。

七、回滚

license回滚

在新申请获取的部署包内执行license更新操作后，如果发现证书异常可以通过下面的方法回滚，

找到旧部署包（可以参考上文约束的命令规范：baidu_ocr_install_日期），执行license update操作

```
###### 进入旧的部署安装包，执行如下命令替换当前的license文件
###### lu，表示 license update
cd original/package/Install && python install.py lu
```

应用回滚 1、查询部署包内包含的模块清单

```
###### se :表示search,根据模块名称查询安装包里是否包括某个模块,不区分大小写,如果没有指定名称则输出安装包里所有模块信息
python install.py se
```

2、选择您要回滚的模块

```
###### rb, rollback: 回滚指定模块，不区分大小写，回滚到最近一次升级前的版本
python install.py rb 模块名
```

若您在此过程出现问题，请提交工单联系百度的工作人员

八、 销毁

支持一键卸载所有模块

```
cd original/package/Install
###### rmall: 删除所有已经安装的模块
python install.py rmall
```

或者卸载指定模块

```
###### rm, remove: 根据模块名称删除某个已经安装的模块;如果有其他模块依赖这个模块，则不允许删除
python install.py rm 模块名
```

九、验证服务可用性

健康检查 百度OCR产品为您提供了私有化应用健康检查（或故障排查）脚本：trouble_shooting.tar，保障您在局域网内部署完毕后进行服务可用性检查或者在遇到突发故障时进行故障排查，方便又快捷，祝您使用愉快！

使用方法: 将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```
###### 解压
tar vxf trouble_shooting.tar
###### 执行
bash trouble_shooting.sh
```

接口测试

接口文档存储路径： 私有化部署包内 `original/docs`目录，如该目录不存在或者目录为空请联系对应AM(商务经理）线下获取对应模型接口文档。

您可根据接口说明文档中的接口地址（GeneralClassifyService/classify 或 GeneralClassifyService/process）来选择对应的代码示例。

GeneralClassifyService/classify

如果接口路径为“/GeneralClassifyService/classify”，请参考如下代码示例：

Python2

Python3

PHP

Node.js

Java

C#

Go

```
###### !/usr/bin/python2
###### -*- coding:UTF-8 -*-
import SimpleHTTPServer
import SocketServer
import sys
import urllib2,urllib
import base64
import hashlib
import json

url = "http://127.0.0.1:<模型应用端口号>/GeneralClassifyService/classify"  #此处ip需填写部署ocr的机器ip,端口填写ocr服务端口，端口号可参考troulbe_shooting.sh返回结果
with open('./test.jpg', "rb") as f:   #此处需填写请求图片的地址
    img = f.read()
base64Data = base64.b64encode(img) # 第一层base64转换
request_str = '<请参考对应接口文档填写对应的请求参数request_str>'   #此处需按照接口文档中的参数填写
data = request_str + "&image=" + base64Data
postData = {'data':base64.b64encode(data)} # 第二层base64转换
```

GeneralClassifyService/process

如果接口地址为“/GeneralClassifyService/process”，请参考如下代码示例：

Python2

```
###### !/usr/bin/python2
###### -*- coding:UTF-8 -*-
import SimpleHTTPServer
import SocketServer
import sys
import urllib2,urllib
import base64
import hashlib
import json

url = "http://127.0.0.1:<模型应用端口号>/GeneralClassifyService/process"  #此处ip需填写部署ocr的机器ip,端口填写ocr服务端口
with open('./test.jpg', "rb") as f:   #此处需填写请求图片的地址
    img = f.read()
base64Data = base64.b64encode(img)  #base64转换

###### 请参考对应接口文档修改下面的请求参数
input={
```

十、运维

您可以前往「私有化部署服务」/「常见问题」查找常见问题排查思路

您可以前往 「私有化部署服务」/「运维手册」查看常用运维文档

iOCR自定义平台部署

一、资源规划

资源规划文档旨在系统资源使用层对真实交付场景做产品层面的通用指导

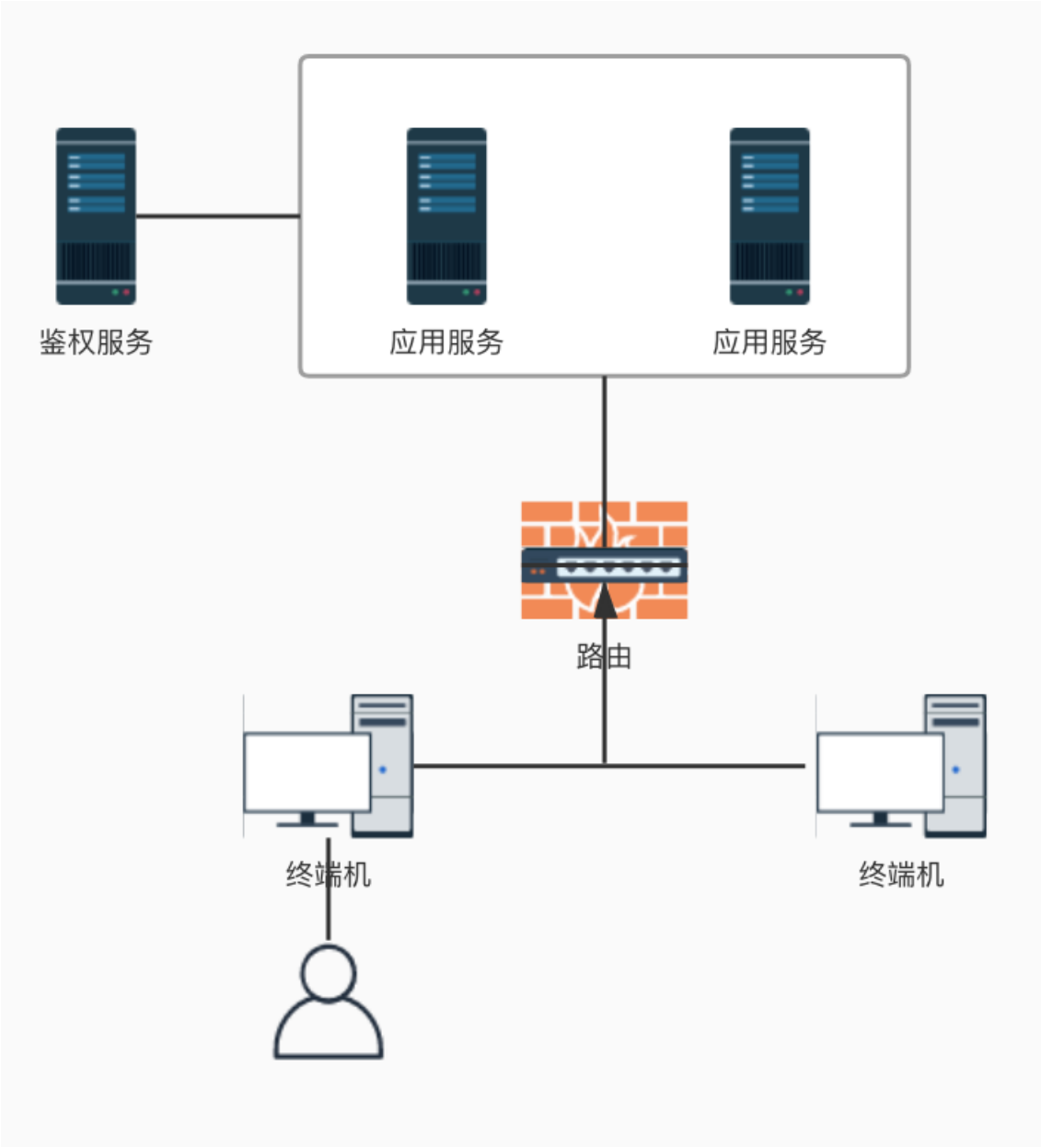
本文档涵盖了构建百度iOCR算子服务所需要的服务器主机、 CPU、内存、 GPU、网络等内容方面的规划。 本文档所列出的所有数量数字方面的值均需针对实际场景需求和限制做针对性调整。

名词定义

名称	含义
鉴权服务	鉴权服务包含百度发布的服务授权证书，如果没有安装鉴权服务，后续的应用服务也将无法启动，目前支持加密狗和提取机器指纹两种鉴权方式
应用服务	包含Docker等基础环境以及相关技术方向的算法模型，是私有化产品的的核心。部署应用服务的前提是部署鉴权服务，应用服务在运行时会实时请求鉴权服务，需要保障两个服务之间能够顺利通信
单机一键部署	适用于鉴权服务、应用服务部署在一台服务器上的场景。即执行一条命令将鉴权服务、应用服务安装完成。
多机一键部署	适用于鉴权服务、应用服务部署在多台服务器上的场景。即执行一条命令将鉴权服务、应用服务安装完成。
测试环境	客户方提供的进行产品、场景测试的环境，一般与生产环境隔离，开发、测试能够方便接触到的环境，对部署的可用性要求较低
生产环境	客户的生产环境，一般的业务系统真实提供服务的环境，具备较高的可用性要求，包括异地多活的灾备要求。一般只有运维人员有该环境的超管权限

物理架构拓扑图 文字识别私有化部署产品包含鉴权服务和应用服务，其中

- 鉴权服务通过客户网关系统连接到应用服务器为应用服务提供鉴权认证
- 应用服务部分直接或通过生产级网关被客户业务场景直接使用



鉴权应用

规划原则

鉴权服务是是您运行文字识别应用服务的基础，如果鉴权异常，将直接导致模型的API接口不可用。鉴权服务健康节点数需满足大于等于N/2 + 1个 (N表示鉴权服务节点总数，并向下取整)，如N=2,需保证2个节点鉴权服务都正常才能保证整体鉴权服务可用；如N=3,需保证存在2个节点鉴权服务正常才能保证整体鉴权服务可用，否则直接影响模型应用可用性。

规划流程

一般建议鉴权节点数量为1或3个。

模型算子应用

规划原则

- 单模型应用实例承担流量，根据业务逻辑复杂度（请求报文）、机器节点硬件条件（CPU、内存、网络、显存）强相关
- 单模型应用实例的内存分配依赖模型的大小
- 单模型应用实例的CPU资源与请求报文相关
- 单模型应用实例占用网路IO与请求报文和返回结果相关
- 单模型应用实例占用磁盘空间大小与请求报文、日志输出量等特性相关
- 场景需要的模型应用服务资源与整体业务QPS、单模型实例性能、高可用方案相关

规划流程

根据实际场景进行性能测试，得出单模型实例性能指标（QPS、响应延时、内存占用、显存占用），结合场景高峰流量预估和高可用要求，以及服务器实际显卡数量等计算需要模型应用实例数，根据机器节点硬件资源指标，最终确定硬件节点数。

应用服务资源规划示例

资源需求	部署模块	节点数	单节点CPU	单节点内存	单节点存储	单节点网络
测试环境	Mysql Redis iocr-api iocr-task iocr-train general-ocr-* Docker	1	4核	8G	50G	>1000Mbps
生产环境	Mysql Redis iocr-api iocr-task iocr-train general-ocr-* Docker	N	8核	16G	500G	>1000Mbps

以上资源数值为推荐参考值，可根据实际情况做测试和调整。

二、场景与名词

场景说明

- （无环境）全新部署：服务器环境为第一次部署，该服务器之前没有部署过任何百度iOCR自定义模板识别产品。

名词解释

名词	说明	示例
package_dir	存放部署包的路径、包体积较大，尽量不要放在/下	如 /mnt/disk0/baidu_ocr_install_20111009,以日期命名
work_dir	应用程序文件存储地址，默认为/home/baidu/work	如 /home/baidu/work

三 、准备工作

请您在部署前**务必**参考此文档部署前环境检查必看进行硬件、网络、及软件环境检查，以避免在安装部署过程中出现问题。

四、全新部署：

获取部署包

1、申请正式模型部署包安装文件下载链接，下载模型部署包。具体可参考：

```
###### 示例如下，-O --output-document=FILE  对文件重名名，O为大写英文字母
###### 请将示例中的9C20XXXXXXXX.tar.gz替换为真实的文件名
wget -O 9C20XXXXXXXX.tar.gz https://bj.bcebos.com/v1/private-ai-online/9C20XXXXXXX.tar.gz?authorization=bce-authXXXXXXX132187fcf81
```

2、将9C20XXXXXXXX.tar.gz上传到待部署的服务器中,建议以【*baiduocr_install + 日期】命名*，如baidu_ocr_install_20111009，该目录我们称之为package_dir

3、进入package_dir 执行以下命令解压部署包

```
cd baidu_ocr_install_20111009 && tar zxvf 9C20XXXXXXXX.tar.gz
```

4、解压后进入original目录执行bash download.sh命令获取全部安装文件，执行脚本后会自动下载以下安装文件：数据库服务安装包、鉴权服务安装包、应用服务安装包以及docker安装包等基础依赖环境。如果已经提前下载完毕，请忽略该步骤。

```
cd original && bash download.sh
```

同时会在download.sh同级目录下生成download.log日志记录下载详情。

若您在此过程出现问题，请提交工单联系百度的工作人员

5、下载iOCR私有化部署工具包

包含初始化环境变量脚本和算子服务验证代码demo，下载地址：iOCR私有化部署工具包，将下载的zip包上传至待部署的服务器并解压

初始化环境变量

找到上一步下载的iOCR私有化部署工具包文件存储路径，进入解压后的“初始化环境变量脚本”目录

1、修改env_local.config， 压缩包里有初始化环境变量的shell脚本env_local.sh,可根据需要修改env_local.config里的配置，该config默认是本地ip和服务默认端口。

脚本配置修改说明：

> a) 默认shell脚本配置为env_local.config，里面的预设的ip都是127.0.0.1，端口都是微服务默认的端口。
>
> b) 如果需要修改ip，打开env_local.config根据实际情况修改即可。
>
> c) 如果服务器ip可能变动，可以将ip设置为docker网桥ip，默认为172.17.0.1

ps：脚本里的各个参数说明见下文：

▶ 点击查看参数说明

2、执行shell脚本导入环境变量。

```
source env_local.sh
```

3、查看环境变量是否成功导入

```
sh print_env.sh
```

主要确认ip端口是否输入错误

正式安装

找到模型部署包的存储路径 package_dir，然后按照如下步骤操作：

1、进入package_dir开始进行鉴权服务和iOCR服务的一键部署。首先进入以下文件路径；

```
###### 进入上一步以日期命名的部署包目录
cd baidu_ocr_install_20111009 && cd original/package/Install
```

使用root权限启动一键部署脚本进行安装：

如果您只有一台服务器，可以使用inall参数来部署,inall 表示install all，安装所有的产品以及鉴权服务和基础服务,适用于在单台服务器上安装所有模块的场景，容易和 install混淆，在使用时请注意。

```
###### inall: 安装所有的产品以及鉴权服务和基础服务,适用于在单台物理机上安装所有模块的场景
python install.py inall
```

安装过程中可能会遇到以下情况：

a)

```
Press any key to quit. Press 'continue' to continue the installation without guarantee of succe
ss.
continue
```

此处尝试输入continue，如果continue以后部署失败，请联系支持人员排查问题。

b) 选择是否初始化数据库：

initialize database for iocr-web ? [y/n]

如果是第一次安装部署，选择是，输入：y，如果不是第一次安装部署，选择否，输入n。

c) 选择模型占用的gpu显卡。

输入显卡号即可。（注意如果需要多个模型记得一个模型需要独占一张卡）

2、确认本次安装模块是否完整：

执行docker ps -a 查看正在运行的docker容器否包含以下且状态均正常：

容器名	说明
iocr-web	页面服务
iocr-api	接口服务
iocr-task	任务服务
iocr-train	训练服务
mysql	数据库
redis	redis服务
general-ocr-*	ocr模型服务
content-ocr-s2	财税模型服务，需要使用预置财务模板时会有该服务，否则不需要该服务

如果您想单独卸载并重新安装某个模块，可以先输入se参数

```
###### se :表示search,根据模块名称查询安装包里是否包括某个模块,不区分大小写,如果没有指定名称则输出安装包里所有模块信息
python install.py se
```

来检索当前模型部署包内包含哪些模块，如：

```
 ......
模块名: openresty， 内置版本 7, 依赖模块 []
模块名: docker， 版本号：1.0， 内置版本 5, 依赖模块 []
......
```

如卸载openresty，可以执行如下命令（此处仅为示例，实际操作中根据需要执行）

```
 ###### rm :remove, 根据模块名称删除某个已经安装的模块;如果有其他模块依赖这个模块，则不允许删除
python install.py rm openresty
```

重新安装openresty，则可以执行（此处仅为示例，实际操作中根据需要执行）

```
 ###### in :install, 安装某个模块,名称不区分大小写
python install.py in openresty
```

五、web页面验证

验证步骤：

1、页面地址：http://ip:8084/iocr

2、默认的登录用户名和密码请联系技术支持同学

3、上传一张合法模板图（base64以后小于4M并且最长边不大于4096）（IOCR辅助脚本压缩包里附了一张模板图），

上传成功后，框选合适的参照字段。

参照字段如何框选？

1）框选4个以上参照字段，并尽量分散在四角

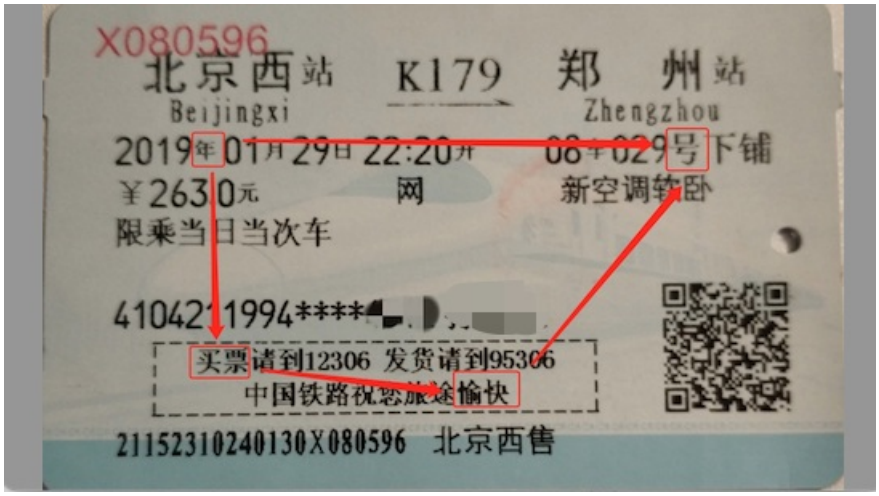
2）保证框选的文字内容、位置固定不变

3）单个参照字段不可跨行

4）选取图片中不会重复出现的文字

5）仅支持中英文、数字，不可包含符号、图案

如下图示例，参照字段不会重复出现，位置固定，且较为分散，将参照字段连起来可以构成饱满的多边形



如上传失败排查步骤见下方：

▶ 上传图片失败排查步骤

4、模板是按用户隔离的，如果需要新增修改删除用户，详细操作如下：

▶ 新增、删除、修改用户操作步骤

操作教程参考线上文档：

https://ai.baidu.com/docs#/iOCR-General-Step/top

六、API接口调用

请求示例：

HTTP 方法：POST

请求URL：

http://ip_address:8085/api/v1/solution/iocr/recognise

URL参数：

参数	说明
ip_address	机器IP地址

Header设置：

参数	说明
Content-Type	application/x-www-form-urlencoded

请求参数

参数	说明
appid	用户id，对应账户表里的id字段,默认账户的appid为1
templateSign	模板id，非必填
classifierId	分类器id，非必填,每次请求模板id和分类器id至少存在一个
image	图片的base64码

返回结构同线上，参考：https://ai.baidu.com/ai-doc/OCR/Ek3h7y961

错误码说明：

同线上，参考：https://ai.baidu.com/ai-doc/OCR/Ek3h7y961

python语言请求demo使用说明：

验证操作步骤：

1. 前面页面验证时创建编辑并发布的模板的templateSign复制粘贴到脚本python2_private_demo.py里的变量template_sign。
2. ip修改：默认为127.0.0.1，根据实际情况修改。
3. 图片修改，如果页面服务验证时使用的是压缩包内的图片，此处可以填压缩包内另一张图片的路径。
4. 运行脚本，python python2_private_demo.py
5. 返回json的errorCode等于0即为成功。

更多语言的demo参考线上文档：

https://ai.baidu.com/ai-doc/OCR/yk3h7y9u3

七、运维

您可以前往「私有化部署服务」/「常见问题」查找常见问题排查思路

您可以前往 「私有化部署服务」/「运维手册」查看常用运维文档

运维手册

修改模型服务端口号

下文将以通用文字识别GPU版为例，详细展开说明修改模型服务端口号的步骤。

1、首先进入部署包目录，检索模型模块命名

```
 cd original/package/Install
##### 根据模型名称检索模块名
python install.py se
```

如【通用文字识别GPU】的模块名为ocr-general-gpu

```
模块名: ocr-general-gpu， 中文名：通用文字识别GPU， 版本号：V2.2， 内置版本 2, 依赖模块 []
```

进入work_dir目录下算子应用目录

```
cd /home/baidu/work/ocr-general-gpu/start/
```

2、修改启动脚本start-1.sh

- 使用bridge网桥模式启动容器

当前百度OCR产品模型默认使用bridge网桥模式启动容器。

hostPort:containerPort：映射本机的指定端口到容器内的指定端口

如：8888:8256，映射本机的8888端口到容器内的8256端口

只需要修改“：”前的端口号，即可实现自定义算子服务端口。

```
docker run -ti -d --restart=always --security-opt seccomp:unconfined --cap-add S
YS_PTRACE -p 8888:8256 --name ocr-general-cpu-1 $DOCKER_ENV $DOCKER_VOLUMES $DOC
KER_IMAGE sh start.sh
-- 插入 --                                                    42,98         底端
```

- 使用host网络模式启动容器

host模式类似于Vmware的桥接模式，与宿主机在同一个网络中，但没有独立IP地址。一个Docker容器一般会分配一个独立的Network Namespace。但如果启动容器的时候使用host模式，那么这个容器将不会获得一个独立的Network Namespace，而是和宿主机共用一个Network Namespace。容器将不会虚拟出自己的网卡，配置自己的IP等，而是使用宿主机的IP和端口。

如果是通过这种方式启动的容器，修改端口号会比较复杂，请联系技术支持同学。

3、重启容器

```
##### 检索相关容器
docker ps -a |grep baidu
##### 停止容器，以ocr-general-gpu为例，实际操作中换成对应的模型容器名
docker stop ocr-general-gpu-1
##### 过10s后删除该容器
docker rm ocr-general-gpu-1
##### 重新启动该容器
sh /home/baidu/work/ocr-general-gpu/start/start-1.sh
```

扩充实例步骤

下文将以通用文字识别GPU版为例，详细展开说明应用服务实例扩容步骤。

修改容器启动脚本

1、首先进入部署包目录，检索模型模块命名

```
cd original/package/Install
##### 根据模型名称检索模块名
python install.py se
```

如【通用文字识别GPU】的模块名为ocr-general-gpu

```
模块名: ocr-general-gpu， 中文名：通用文字识别GPU， 版本号：V2.2， 内置版本 2, 依赖模块 []
```

进入work_dir目录下算子应用目录

```
cd /home/baidu/work/ocr-general-gpu/start/
```

如果要额外扩充9个实例,执行如下shell命令复制start-1.sh脚本

注：实例的数量不要超过license中限制的实例数

```
for i in $(seq 2 10); do cp start-1.sh start-$i.sh; done
```

2、依次修改克隆的start-*.sh脚本，修改docker run的启动参数，包括

1） 映射到宿主机的端口号

2） docker容器的Name

3） 为每一个docker容器指定一块显卡 (CPU模板不需要配置)

请提前规划每个应用容器绑定的显卡编号，如下

启动脚本名字	显卡编号
start-1.sh	0
start-2.sh	1
……	……
start-10.sh	9



3、依次执行stat-*.sh脚本启动算子应用容器

openresty 负载均衡配置

1、前往work_dir下的openresty目录,可以看到以模型模块名称命名的.conf配置文件

```
cd /home/baidu/work/openresty/conf/upstream && ll
```

返回如下

```
-rw-r--r-- 1 root root 81 10月 13 13:54 ocr-general-gpu.conf
-rw-r--r-- 1 root root 24 10月 12 16:27 upstream-template
```

2、修改ocr-general-gpu.conf文件配置参数

文件内容如下，其中ip和port 为本机IP和算子容器映射在宿主机的端口号，假如通用文字识别GPU版本一共启动了10个实例

```
 upstream ocr-general-gpu {
server ip:port1 max_fails=3 fail_timeout=30;
server ip:port2 max_fails=3 fail_timeout=30;
server ip:port3 max_fails=3 fail_timeout=30;
server ip:port4 max_fails=3 fail_timeout=30;
server ip:port5 max_fails=3 fail_timeout=30;
server ip:port6 max_fails=3 fail_timeout=30;
server ip:port7 max_fails=3 fail_timeout=30;
server ip:port8 max_fails=3 fail_timeout=30;
server ip:port9 max_fails=3 fail_timeout=30;
server ip:port10 max_fails=3 fail_timeout=30;
}
```

3、修改/home/baidu/work/openresty/conf/vhost/vhost.conf

内容如下：

```
 server {
listen 8666; #负载均衡（LB）的端口号，可自定义修改。流量统一发送给该端口号，然后通过openresty将流量转发给后端算子应用端口号
server_name 127.0.0.1; #本机IP地址,默认值
port_in_redirect off;
proxy_http_version 1.1;
proxy_set_header Connection "";

location / {
 set $group_name '模型名称'; # 修改模型名称为模型模块名，如ocr-general-gpu，与第2步定义的upstream名一致
 proxy_pass http://模型名称/; # 修改模型名称为模型模块名，如ocr-general-gpu，与第2步定义的upstream名一致
 proxy_read_timeout 10;
 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-for $remote_addr;
 }
}
```

4、重启使其生效

```
 ##### 强制删除现有的nginx-1 容器
docker rm -f nginx-1

##### 启动nginx容器
cd /home/baidu/work/openresty && bash start/start-1.sh
```

更新license

如果您的实例数超过了旧license中实例数限制，需要重新申请license后对license做更新操作。

```
 ##### 找到新申请的license升级包的路径，建议部署包以 baidu_ocr_日期命名
cd original/package/Install

##### lu, licenseupdate: 更新license文件，适用于授权延期、实例数扩容、增加产品授权
python install.py lu
```

验证

1、首先验证负载均衡容器状态可用

```
[root@instance-wch0lkwp Install]# docker ps
CONTAINER ID        IMAGE                                                COMMAND                  CREATED             STATUS              PORTS               NAMES
c1643ee21c44        registry.baidu.com/aipe/openresty:1.11.2.3-trusty    "/usr/local/openre..."   40 seconds ago      Up 3 seconds                            nginx-1
```

2、测试发往LB端口的流量可以被正常处理。具体可参考标准模型部署文档中的测试代码用例。

鉴权服务重启

停止服务：

```
cd /home/baidu/work/c-offline-security-server/ && bash start/c-offline-security-server-stop.sh
```

启动服务：

```
cd /home/baidu/work/c-offline-security-server/ && nohup bash start/c-offline-security-server-start.sh &
```

修改容器网络模式

下文将以通用文字识别GPU版为例，详细展开说明修改应用服务容器网络模式的步骤。

1、首先进入部署包目录，检索模型模块命名

```
cd original/package/Install
##### 根据模型名称检索模块名
python install.py se
```

如【通用文字识别GPU】的模块名为ocr-general-gpu

```
模块名: ocr-general-gpu， 中文名：通用文字识别GPU， 版本号：V2.2， 内置版本 2, 依赖模块 []
```

进入work_dir目录下算子应用目录

```
cd /home/baidu/work/ocr-general-gpu/start/
```

2、修改启动脚本start-1.sh，添加 --net=host 参数（**如果已经有--net参数，请将其值修改为host**）来使用主机网络模式，如果启动容器的时候使用host模式，那么这个容器将不会获得一个独立的Network Namespace，而是和宿主机共用一个Network Namespace。容器将不会虚拟出自己的网卡，配置自己的IP等，而是使用宿主机的IP和端口。



3、重启容器

```
##### 检索相关容器
docker ps -a |grep baidu
##### 停止容器，以ocr-general-gpu为例，实际操作中换成对应的模型容器名
docker stop ocr-general-gpu-1
##### 过10s后删除该容器
docker rm ocr-general-gpu-1
##### 重新启动该容器
sh /home/baidu/work/ocr-general-gpu/start/start-1.sh
```

License更新步骤

服务License更新说明

适用场景

适用于已经完成私有化部署的客户对License延期、License扩容、License新增产品授权的场景。

1、License延期：延长私有化部署包使用时间。

2、License扩容：增加私有化部署的应用实例数，使私有化部署包可在更多的设备上部署。

3、License新增产品授权：如部署完A产品后还需要在当前环境（同一台机器或同样的局域网环境）进行B产品的部署，A产品和新增的B产品使用同一套鉴权服务，需进行License变更。

如何获取更新后的License

1、License延期：在console端管理控制台**产品服务/XXX-本地部署包管理/查看详情**页面点击延期申请，发起License延期，审核通过后即可获得更新的License证书文件。或线下沟通后获取新的License证书文件。

2、License扩容：请线下商务沟通后获取新的License证书文件。

3、License新增产品授权：请线下商务沟通后获取新的License证书文件。 **更新流程**

EasyPack一键部署工具支持对私有化部署包的License证书进行一键更新，一键更新时**不需要执行download.sh 或 download.bat,并且不需要安装任何模块，在新申请的部署包中执行更新操作即可**。具体请参考以下流程进行License证书更新。

1、获取到包含License的部署包后，解压缩，进入以下文件路径.

```
cd original/package/Install
```

2、执行以下命令完成License更新。

```
python install.py lu
```

3、操作成功后如下图所示，提示License update successfully即更新完成。

```
[root@localhost Install]# python install.py lu

 _____       _     _                 ____                        _
| ___ \     (_)   | |               |  __|                      | |
| |_/ /  __ _  __| | |_   _  ___   | |__   __ _ ___  _   _  | |_/ /  __ _  ___| | __
| ___ \ / _` |/ _` | | | | |/ _ \  |  __| / _` / __|| | | | |  __/ / _` |/ __| |/ /
| |_/ /| (_| | (_| | | |_| |  __/  | |___| (_| \__ \| |_| | | |   | (_| | (__|   <
\____/  \__,_|\__,_|_|\__,_|\___|  \____/ \__,_|___/ \__, |\_|    \__,_|\___|_|\_\
                                                      __/ |
                                                     |___/

--------------------------------------------------------------------------------
2019-06-05 16:55:49,031 - 29135 - install - INFO - start to prepare work directory...
2019-06-05 16:55:49,033 - 29135 - install - INFO - install option: lu, user name: baidu
2019-06-05 16:55:49,039 - 29135 - install - INFO - subprocess start,cmd : cp -rf /home/baolong/original/lc//license/license /home/baidu/work/c-offline-security-server/license
2019-06-05 16:55:49,044 - 29135 - install - INFO - subprocess finished,cmd : cp -rf /home/baolong/original/lc//license/license /home/baidu/work/c-offline-security-server/license
Please wait a moment to complete license update...
2019-06-05 16:56:19,073 - 29135 - install - INFO - License update successfully.
```

修改docker的默认存储路径

Docker 默认安装的情况下，会使用 /var/lib/docker/ 目录作为存储目录，用以存放拉取的镜像和创建的容器等。不过由于此目录一般都位于系统盘，遇到系统盘比较小，而镜像和容器多的情况会影响服务的健壮性，这里说明以下如何修改 Docker 的存储目录。

输入

```
docker info|grep -i root
```

可以查看当前的docker存储目录

```
Docker Root Dir: /var/lib/docker
```

详细操作步骤如下：

1、停止docker服务

```
systemctl stop docker
```

2、创建新的存储路径，并迁移历史数据

```
mkdir -p /mnt/disk0/docker
mv /var/lib/docker /mnt/disk0/docker
```

3、编辑 /etc/docker/daemon.json 文件添加如下参数：

```
{
 "data-root": "/mnt/disk0/docker"
}
```

4、保存退出，然后重启 docker 服务：

```
systemctl restart docker
```

5、检查是否生效

```
docker info|grep -i root
docker ps
```

机器指纹鉴权切换加密狗硬件鉴权

该文档用于介绍如何将物理机器部署的c-offline-sercurity-server离线鉴权服务从机器指纹版本切换为加密狗硬件版本。该文档仅适用于物理机部署场景。

加密狗硬件鉴权包获取请您线下联系商务经理。

两者的目录结构是一致的

```
[root@yq01-aip-3e21e.yq01.host test]# tree .
.
├── new_hardware_package (加密狗硬件鉴权部署包)
│   └── original
│       ├── download.sh
│       ├── package
│       └── refs.txt
└── old_software_package （机器指纹软件鉴权）
    └── original
        ├── download.sh
        ├── package
        └── refs.txt
```

如何区分离线鉴权服务不同版本？

```
cd original/package/Install/
python install.py search
##### 或 python install.py se
```

如果c-offline-security-server 版本号 返回 `with-dog` 表示 **加密狗硬件鉴权**

```
模块名: c-offline-security-server， 版本号：with-dog， 内置版本 xxx，依赖模块 []
```

如果c-offline-security-server 版本号 `virtual` 表示 **虚拟机版本-机器指纹方式鉴权**

```
模块名: c-offline-security-server， 版本号：virtual， 内置版本 x, 依赖模块 []
```

如果c-offline-security-server 版本号 `no-dog` 表示 **物理机版本-机器指纹方式鉴权**

```
模块名: c-offline-security-server， 版本号：no-dog， 内置版本 x, 依赖模块 []
```

替换步骤

1、将部署包解压后进入original目录执行 `bash download.sh`命令获取全部安装文件

```
cd original && bash download.sh
```

2、将旧的机器指纹方式鉴权服务卸载

```
##### 进入新的部署包 （加密狗硬件鉴权部署包）
cd package/Install
python install.py remove c-offline-security-server
##### 或 python install.py rm c-offline-security-server
##### 检查/home/baidu/work/c-offline-security-server 是否存在，如存在将其删除
rm -rf /home/baidu/work/c-offline-security-server
```

3、安装加密狗硬件离线鉴权服务

```
python install.py install c-offline-security-server
##### 或 python install.py in c-offline-security-server
```

4、耐心等待几分钟后，执行私有化应用健康检查（或故障排查）脚本：trouble_shooting.tar 来验证 c-offline-sercurity-server 服务

```
##### 解压
tar vxf trouble_shooting.tar
##### 执行
bash trouble_shooting.sh
```

检查加密狗驱动是否运行，正常情况下返回 CodeMeter Server is running.

```
service codemeter status
```

检查加密狗硬件是否被机器识别

```
cmu -x
```

识别成功的结果如下

```
cmu - CodeMeter Universal Support Tool.
Version 6.70 of 2018-Jul-19 (Build 3152) for Linux
Copyright (C) 2007-2018 by WIBU-SYSTEMS AG. All rights reserved.
List all locally connected CmContainers:
- CmContainer with Serial Number 3-4512221 and version 4.10
...
Result: 1 CmContainer(s) listed.
```

识别失败的结果如下：

```
cmu - CodeMeter Universal Support Tool.Version 6.70 of 2018-Jul-19 (Build 3152) for LinuxCopyright (C) 2007-2018 by WIBU-SYSTEMS AG. All rights reserved.List all locally connected CmContainers:Result: 0 CmContainer(s) listed.
```

常见问题

安装部署问题排查

在私有化部署过程中遇到的部署相关问题，可以查看此文档进行解决。

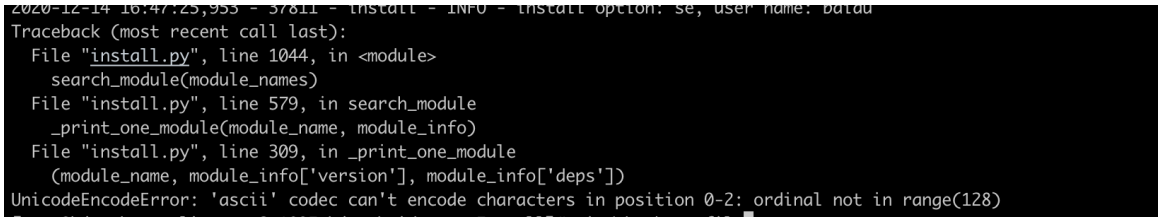
若文档仍未解决您的问题，请提交工单联系百度的工作人员

FAQ

1.容器日志报错：cudaErrorNoDevice: no CUDA-capable device is detected at

查看/home/baidu/work/模型名称/start/start-1.sh文件中指定显卡ID的地方 是否是对的，显卡ID从序号0开始

2.'ascii' codec can't encode characters in position 0-2: ordinal not in range(128)



机器系统环境编码有问题，PYTHONIOENCODING=utf-8 python install.py inall用这个命令安装就可以。

3.安装docker报错no such file or directory: '/etc/sysconfig/network-scripts/ifcfg-docker0'文件不存在

需要手动添加以下/etc/sysconfig/network-scripts/ifcfg-docker0这个文件，内容如下，然后重新执行安装

```
 STP=no
TYPE=Bridge
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=172.17.0.1
PREFIX=16
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV4_DNS_PRIORITY=100
IPV6INIT=no
NAME=docker0
UUID=a519039a-27f3-40a0-b571-e2aefc39d20a
DEVICE=docker0
ONBOOT=yes
ZONE=trusted
```

4、安装docker-ce，依赖包冲突 出现需要安装依赖的版本比系统自带版本低的情况，导致安装失败，详细报错如下

```
 --> 解决依赖关系完成
错误：软件包：libsemanage-python-2.5-8.el7.x86_64 (Local_yum)
        需要：libsemanage = 2.5-8.el7
        已安装: libsemanage-2.5-14.el7.x86_64 (@anaconda)
            libsemanage = 2.5-14.el7
        可用: libsemanage-2.5-8.el7.x86_64 (Local_yum)
            libsemanage = 2.5-8.el7
错误：软件包：audit-libs-python-2.7.6-3.el7.x86_64 (Local_yum)
        需要：audit-libs(x86-64) = 2.7.6-3.el7
        已安装: audit-libs-2.8.5-4.el7.x86_64 (@anaconda)
            audit-libs(x86-64) = 2.8.5-4.el7
        可用: audit-libs-2.7.6-3.el7.x86_64 (Local_yum)
            audit-libs(x86-64) = 2.7.6-3.el7
错误：软件包：policycoreutils-python-2.5-17.1.el7.x86_64 (Local_yum)
        需要：policycoreutils = 2.5-17.1.el7
        已安装: policycoreutils-2.5-34.el7.x86_64 (@anaconda)
            policycoreutils = 2.5-34.el7
        可用: policycoreutils-2.5-17.1.el7.x86_64 (Local_yum)
            policycoreutils = 2.5-17.1.el7
 您可以尝试添加 --skip-broken 选项来解决该问题
 您可以尝试执行：rpm -Va --nofiles --nodigest

2021-10-21 13:58:19,993 - 7100 - install - INFO - subprocess finished,cmd : ['yum', 'install', '-y', 'docker-ce']
```

解决方案： 服务器联网条件下，则可以通过在线安装解决。

```
 yum -y install docker-ce
```

如果yum安装 docker-ce 返回 `no package docker-ce available`，请使用如下方法解决：

```
 ##### 安装yum管理工具
yum install -y yum-utils

##### yum添加软件源
yum-config-manager \
--add-repo \
https://mirrors.ustc.edu.cn/docker-ce/linux/centos/docker-ce.repo

##### 刷新缓存
yum makecache fast

##### 安装docker-ce
yum install docker-ce
```

5、nvidia-docker2 安装失败

如果当前服务器已经安装>17.06.2 版本的docker， 通过`python2 install.py inall` 或 `python2 install.py in nvidia` 安装nvidia-docker2 的话 一般会遇到与现有docker版本冲突的问题。

解决方案： 如果服务器可以联网的话,可以yum来在线安装

```
 yum install -y nvidia-docker2
```

如果yum安装 nvidia-docker2 返回 `no package nvidia-docker2 available`，请使用如下方式解决：

```
 distribution=$(source /etc/os-release;echo $ID$VERSION_ID) && curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.repo | sudo tee /etc/yum.repos.d/nvidia-docker.repo

yum clean expire-cache
yum install -y nvidia-docker2
systemctl restart docker
```

6、鉴权服务安装或启动失败，日志报错too many open files

原因：句柄数超出系统限制

首先查看当前全部进程占用句柄数总和：

```
 lsof|awk '{print $2}'|wc -l
```

然后执行 `ulimit -a` 查看当前系统设置的最大句柄数是多少，如下图 open files即是最大句柄数设置

```
[root@localhost bin]# ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 127044
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files                      (-n) 4096
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 10240
cpu time               (seconds, -t) unlimited
max user processes              (-u) 4096
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

如果当前总和超过最大句柄限制，则修改最大句柄数即可

修改方法如下：

```
echo "* soft nofile 65536" >> /etc/security/limits.conf
echo "* hard nofile 65536" >> /etc/security/limits.conf
```

如果/etc/security/limits.conf里已经做过如上调整，修改其阈值即可。

退出当然用户，重新ssh登录使其生效。再次执行 `ulimit -a`验证是否生效

如果修改后，程序运行一段时间之后继续出现Too many open files异常，那么就应该查看句柄信息，进一步分析是什么句柄占用最多

```
cat lsof.log | awk '{print $8}' | sort | uniq -c | sort -rn | head -n 10
```

然后再进行分析，句柄问题解决后，重新安装或启动鉴权服务即可。

鉴权服务排查

机器指纹方式（软件）鉴权

该文档用于帮助您快速定位鉴权服务异常原因。

前面部署章节有提到私有化应用健康检查（或故障排查）脚本：trouble_shooting.tar，该工具会自动检查鉴权服务是否正常。

▶ 点击展开》如何手动检查鉴权服务进程、端口号以及相关日志？

1） c-offline-security-server 应用常见鉴权报错日志和其对应含义如下：

鉴权报错日志	含义	解决办法
verify finger is invalid	license里的指纹错误	请确认运行鉴权服务的服务器是否有变化， 如硬盘、网卡等，如有变化， 请重新提取指纹、申请授权
verify product has not foud	产品未授权	此服务器未获得授权，无法运行OCR模型 请重新提取指纹、申请授权
verify product lc is expired	产品授权过期	请重新提取指纹、申请授权
instance_check reg fail,r_list full	实例池已满 运行模型的服务器数量 （或实例数量）超出授权上限	请减少运行模型的服务器数量、 或实例数量
Environment is unsafe	运行环境不安全，可能是license的版本不对或者人为修改了license。	
cache client get context error: Connection refused	鉴权服务连接本机缓存失败，可能原因：7091和7092缓存端口没有启动，或者部署时填写的鉴权节点IP有误	

2）OCR算子容器日志错误码解读

通过执行`docker logs -f 容器名或容器ID` 或 根据私有化应用健康检查（或故障排查）脚本：trouble_shooting.tar 输出的容器日志结果发现有类似如下报错内容：

```
aipe auth failed, ret = 501

aipe auth failed, ret = 504
```

相关错误码解读如下：

错误码	含义	备注
0	鉴权成功	
1	curl初始化报错curl不支持的协议。离线鉴权通过https做了双向认证，如果鉴权server不是https，会报这个错	更多<100的可能未列出的错误码请直接参考：https://curl.haxx.se/libcurl/c/libcurl-errors.html
7	无法连接到鉴权服务器，一般是网络不通或者鉴权服务没起来（8443端口没开）	
28	请求鉴权服务超时。目前设置的connect timeout为1s，socket timeout为3s	
35	ssl双向认证失败。离线鉴权通过https做了双向认证，如果客户端/服务端证书不对，会报这个错	
501	鉴权失败，实例数超过上限	
502	鉴权失败，老版本状态码未拆分，产品未授权或指纹错误时触发，SDK三位版本1-0-12后，产品未授权错误码为508，指纹错误为509	
503	鉴权失败，产品授权过期了	
504	鉴权失败，申请的license中开启了qps控制，但是运行时却未开启异步鉴权	需要将容器启动脚本start-1.sh 中EASYPACK_ENABLE_ASYNC_AUTH环境变量值设置为1
505	鉴权失败，qps超限	
506	鉴权失败，异步鉴权模式下，异步鉴权的线程挂掉了	
507	鉴权失败，异步鉴权模式下，缓存的鉴权结果太长时间（10min）没有刷新。比如鉴权线程被挂起了，导致无法更新鉴权结果。	
508	鉴权失败，产品未授权	
509	鉴权失败，机器指纹有误	
2012	鉴权失败，鉴权服务器全部不可用。（AIPE_SECURITY_SERVER_HOST如果配置了多个ip，会自动开启到多个鉴权服务的探活，如果探活全部失败了，则报这个错）	
-1	未知错误	

加密狗方式（硬件）鉴权

如果使用了加密狗硬件辅助鉴权，除了按照上文【机器指纹方式（软件）鉴权】提到的排查思路依次进行排查以外， 需要额外检查以下内容

> 如以下指令无法执行，请参照部署文档执行 `bash download.sh`命令，该脚本会自动下载加密狗相关驱动文件，之后按照部署文档一键安装或单独安装即可。

1）加密狗驱动是否运行

```
service codemeter status
```

正常情况下返回 CodeMeter Server is running.

2）加密狗硬件是否被机器识别

```
cmu -x
```

识别成功的结果如下

```
 cmu - CodeMeter Universal Support Tool.
Version 6.70 of 2018-Jul-19 (Build 3152) for Linux
Copyright (C) 2007-2018 by WIBU-SYSTEMS AG. All rights reserved.
List all locally connected CmContainers:
- CmContainer with Serial Number 3-4512221 and version 4.10
...
Result: 1 CmContainer(s) listed.
```

识别失败的结果如下：

```
 cmu - CodeMeter Universal Support Tool.Version 6.70 of 2018-Jul-19 (Build 3152) for LinuxCopyright (C) 2007-2018 by WIBU-SYSTEMS AG. All rights reserved.List all locally connected CmContainers:Result: 0 CmContainer(s) listed.
```

查看license信息

```
curl 0.0.0.0:8443/security/license
```

返回示例如下

```
{"dogCount":0,"fingerType":3,"holder":"15322","info":"license","issued":"2021-10-12 16:27:36","issuer":"baidu","productLicenses":[{"functions":[],"instanceSize":1,"notAfter":"2021-11-26 00:00:00","notBefore":"2021-10-11 00:00:00","qps":0,"subject":"general_nor"}],"safeType":1,"securityType":2}
```

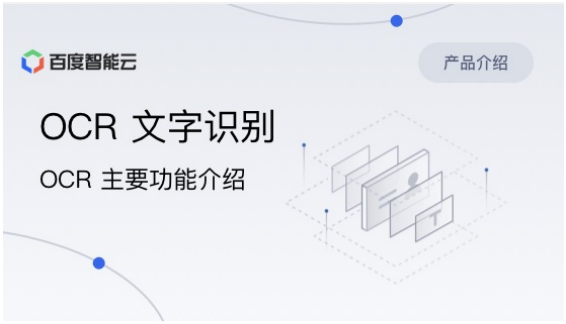
license有效期为 notBefore ~ notAfter

视频专区

产品介绍

文字识别产品介绍

- 2分钟带您认识百度智能云文字识别



操作指南

公有云API

通用类

API服务快速接入教程 （示例代码中心）



- API服务调用接入教程 （Postman/代码调用）



- 标准能力类
  - 身份证识别调用教程



  - 智能财务票据识别+增值税发票验真调用教程



  - 行驶证识别调用教程



  - 驾驶证识别调用教程



  - 车牌识别调用教程



iOCR通用版调用教程



表格文字识别V2 调用教程



表格文字识别（异步）调用教程



在线SDK

在线Android SDK使用教程



离线SDK

离线Android SDK使用教程（批量设备授权）



离线Android SDK使用教程（单台设备授权）



OCR离线SDK使用方法（Windows 版本）



OCR离线SDK使用方法（Linux 版本）



私有化

私有化部署包申请流程



私有化部署操作教程



常见问题

前期准备

- 领取免费额度的方法



调用操作

- 获取access_token的方法



- 单应用使用不同技术方向API接口的方法



查询调用情况

- 查看API接口的资源消耗情况



- 查询调用失败的错误码



账户管理

- 账户充值、查看余额和开通付费



- 查看账单/收支明细



- 设置账户余额提醒（避免扣费）



- 账号设置多联系人



历史版本

表格文字识别(同步接口)

> **该接口已停止更新且即将下线，如需更好的识别效果，请使用 表格文字识别V2，此服务支持识别更多类型表格，包括有线表格、无线表格、合并单元格表格等。**

接口描述

支持识别表格线齐全的常规表格和无框线表格的单元格内容，结构化输出表头、表尾及每个单元格的文字内容。 本接口为同步接口，相比于异步接口，本接口在请求后会实时返回请求结果。

提交请求接口

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
table_border	否	string	normal/none	识别表格对象是否有框线。缺省或 table_border = normal，可识别框线齐全的常规表格，table_border = none，可识别无框线表格。默认为normal

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/form?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
forms_result_num	是	uint32	识别结果元素个数
forms_result	是	array[]	识别结果
+ body	是	array[]	表格主体区域
+ footer	是	array[]	表格尾部区域信息
header	是	array[]	表格头部区域信息
vertexes_location	是	array[]	表格边界顶点

返回示例 参见表格识别（异步接口）

表格文字识别(异步接口)

> **该接口已停止更新且即将下线，如需更好的识别效果，请使用 表格文字识别V2，此服务支持识别更多类型表格，包括有线表格、无线表格、合并单元格表格等。**

接口描述

对图片中的表格文字内容进行提取和识别，结构化输出表头、表尾及每个单元格的文字内容。支持识别常规表格及含合并单

元格表格，并可选择以JSON或Excel形式进行返回。 本接口为异步接口，分为两个API：提交请求接口、获取结果接口。下面分别描述两个接口的使用方法。

提交请求接口

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
is_sync	否	string	true/false	是否同步返回识别结果。取值为“false”，需通过获取结果接口获取识别结果；取值为“true”，同步返回识别结果，无需调用获取结果接口。默认取值为“false”
request_type	否	string	json/excel	当 is_sync=true 时，需在提交请求时即传入此参数指定获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串。当 is_sync=false 时，需在获取结果时指定此参数。

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/solution/v1/form_ocr/request?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	list	返回的结果列表
+ request_id	是	string	该请求生成的request_id，后续使用该request_id获取识别结果

返回示例

成功返回示例：

```
{
  "result" : [
     {
        "request_id" : "1234_6789"
     }
  ],
  "log_id":149689853984104
}
```

失败返回示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

获取结果接口

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
request_id	是	string	-	发送表格文字识别请求时返回的request id
result_type	否	string	json/excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	object{}	返回的结果
+ result_data	是	string	识别结果字符串，如果request_type是excel，则返回excel的文件下载地址，如果request_type是json，则返回json格式的字符串
+ percent	是	int	表格识别进度（百分比）
+ request_id	是	string	该图片对应请求的request_id
+ ret_code	是	int	识别状态，1：任务未开始，2：进行中,3：已完成
+ ret_msg	是	string	识别状态信息，任务未开始，进行中,已完成

返回示例

成功返回示例：

```
{
  "result" : {
     "result_data" : "",
     "percent":100,
     "request_id": "149691317905102",
     "ret_code": 3
     "ret_msg": "已完成",
  },
  "log_id":149689853984104
}
```

当request_type为excel时，result_data格式样例为：

```
{
  "file_url":"https://ai.baidu.com/file/xxxfffddd"
}
```

当request_type为json时，result_data格式样例为：

```
{
"result": {
 "result_data":"{
   "form_num": 1,
   "forms": [
   {
    "footer": [],
    "header": [
    {
     "column": [
      1,
      2
      ],
     "probability":0.925165,
     "rect":{"left":1138.0,"top":127.0},
     "row": [
      1
      ],
     "word": "表头信息1",
     }
     ],
    "body": [
    {
     "column": [
      1,
      2
      ],
     "probability":0.999275,
     "rect":{"left":171.0,"top":26.0},
     "row": [
      1
      ],
     "word": "单元格文字",
     }
     ],
    }
   ]
}
}
```


其中各个参数的说明(json方式返回结果时)：

字段	是否必选	类型	说明
form_num	是	int	表格数量（可能一张图片中包含多个表格）
forms	是	list	表格内容信息的列表
+ header	是	list	每个表格中，表头数据的相关信息
+ footer	是	list	表尾的相关信息
+ body	是	list	表格主体部分的数据
++ row	是	list	该单元格占据的行号
++ column	是	list	该单元格占据的列号
++ word	是	string	该单元格中的文字信息

失败返回示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

多卡证类别检测

> **该接口的公有云服务即将下线，若您仍需要使用多卡证类别检测，您可以选择私有化部署，提交合作咨询联系我们。**

接口描述

对同一张图片中的多种卡证进行类别检测和定位，支持身份证正反面、行驶证正副页、驾驶证正副页、银行卡、营业执照5类常见卡证、8种版式；可结合卡证识别能力，应用在身份证复印件录入、车辆审查、驾照审核等场景，提升业务处理效率。

请求说明

请求示例

HTTP 方法: `POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	类型	是否必须	说明
image	string	和url二选一	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	string	和image二选一	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效请注意关闭URL防盗链

返回说明

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	唯一的log id，用于问题定位
result	array[]	是	识别结果数组
+ card_type	string	是	分类类型
+ probability	string	是	分类置信度
+ location	object	是	位置数组（坐标0点为左上角）
++ width	uint32	是	表示定位位置的长方形的宽度
++ height	uint32	是	表示定位位置的长方形的高度
++ top	uint32	是	表示定位位置的长方形左上顶点的垂直坐标
++ left	uint32	是	表示定位位置的长方形左上顶点的水平坐标

返回示例


```
{
  "result": [
    {
      "probability": 0.9982099533081055,
      "location": {
        "top": 2.8788100183010097,
        "left": 22.013554573059082,
        "width": 965.344873666763318,
        "height": 606.3181607425212903
      },
      "card_type": "idcard_back"
    },
    {
      "probability": 0.5510760545730591,
      "location": {
        "top": 692.9076057672501,
        "left": 0,
        "width": 968.4838762879372083,
        "height": 553.1868463754660
      },
      "card_type": "idcard_front"
    }
  ],
  "log_id": "1274939764931297280"
}
```


智能结构化识别

> **该接口已停止更新且即将下线，为避免影响您目前的业务使用，请您及时进行相关迁移工作**

接口描述

结构化识别各类卡证、票据，无需配置结构化对应关系、无需提取关键词、无需定制开发，直接上传图片即可获得结构化识别信息。

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“[Access Token获取](https://ai.baidu.com/ai-doc/REFERENCE/Ck3dwjhhu)”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效</br>请注意关闭URL防盗链
detect_direction	否	string	true/false	是否进行方向检测及矫正，**默认为 false，可缺省**</br>- **false：**不进行方向矫正，返回参数 direction 固定为 0</br>- **true：**开启自动方向矫正功能，可识别旋转90/180/270度的图片，并返回 direction 检测数值
detect_null_word	否	string	true/false	是否返回未成功匹配的Key/Value值，缺少的对应 Value/Key 置为 NULL，**默认为 false，可缺省**</br>- **false：**不返回未匹配成功的单独 Key/Value，仅返回成功匹配的 KV 结果组</br>- **true：**返回未匹配成功的单独 Key/Value，缺少的对应 Value/Key 置为 NULL
probability	否	string	true/false	是否返回字段识别结果的置信度，**默认为 false，可缺省**</br>- **false：**不返回字段识别结果的置信度</br>- **true：**返回字段识别结果的置信度，包括字段识别结果中各字符置信度的平均值（average）、最小值（min）和方差（variance）

返回说明

返回参数

字段	是否必输出	类型	说明
log_id	是	uint64	调用日志id，用于问题定位
direction	是	int	图片旋转角度，当请求参数 detect_direction=true 时，返回图片方向检测结果 0/1/2/3 分别代表 不旋转、逆时针旋转 90/180/270度
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array	识别结果数组
+ key	是	object	关键词信息，与 value 一一对应，形成 key：value 结构化识别结果组
++ word	是	string	识别结果
++ location	是	object	字段位置信息
+++ top	是	int	字段的上边距
+++ left	是	int	字段的左边距
+++ height	是	int	字段的高度
+++ width	是	int	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+++ variance	否	float	字段识别结果中各字符的置信度方差
+ value	是	object	字段内容，与 key 一一对应，形成 key：value 结构化识别结果组
++ word	是	string	识别结果
++ location	是	object	字段位置信息
+++ top	是	int	字段的上边距
+++ left	是	int	字段的左边距
+++ height	是	int	字段的高度
+++ width	是	int	字段的宽度
++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数

++ probability	否	object	字段识别结果置信度，当请求参数 probability=true 时，以上各字段均包含此参数
+++ average	否	float	字段识别结果中各字符的置信度平均值
+++ min	否	float	字段识别结果中各字符的置信度最小值
+++ variance	否	float	字段识别结果中各字符的置信度方差

返回示例

```json
{
 "words_result": [
  {
   "value": {
    "probability": {
     "average": 0.99977076053619,
     "min": 0.99970018863678,
     "variance": 4.9803929869086e-9
    },
    "location": {
     "top": 146,
     "left": 365,
     "width": 50,
     "height": 25
    },
    "word": "丛齐"
   },
   "key": {
    "probability": {
     "average": 0.99997997283936,
     "min": 0.99997985363007,
     "variance": 1.4210854715202e-14
    },
    "location": {
     "top": 149,
     "left": 311,
     "width": 40,
     "height": 20
    },
    "word": "姓名"
   }
  },
  {
   "value": {
    "probability": {
     "average": 0.9999588727951,
     "min": 0.9999588727951,
     "variance": 0
    },
    "location": {
     "top": 190,
     "left": 366,
     "width": 20,
     "height": 21
    },
    "word": "男"
   },
   "key": {
    "probability": {
     "average": 0.99983507394791,
     "min": 0.99969410896301,
     "variance": 1.9871126966109e-8
    },
    "location": {
     "top": 192,
     "left": 312,
```

```
    "width": 39,
    "height": 19
   },
   "word": "性别"
  }
 },
 {
  "value": {
   "probability": {
    "average": 0.99964165687561,
    "min": 0.99964165687561,
    "variance": 0
   },
   "location": {
    "top": 228,
    "left": 366,
    "width": 22,
    "height": 20
   },
   "word": "汉"
  },
  "key": {
   "probability": {
    "average": 0.99994975328445,
    "min": 0.99993216991425,
    "variance": 3.0917490789761e-10
   },
   "location": {
    "top": 228,
    "left": 310,
    "width": 41,
    "height": 20
   },
   "word": "民族"
  }
 },
 {
  "value": {
   "probability": {
    "average": 0.99989211559296,
    "min": 0.9996235370636,
    "variance": 1.0301564046244e-8
   },
   "location": {
    "top": 262,
    "left": 356,
    "width": 158,
    "height": 21
   },
   "word": "1989年7月28日"
  },
  "key": {
   "probability": {
    "average": 0.99828881025314,
    "min": 0.99683433771133,
    "variance": 0.0000021154903606657
   },
   "location": {
    "top": 264,
    "left": 312,
    "width": 40,
    "height": 19
```

```
    },
    "word": "出生"
   }
  },
  {
   "value": {
    "probability": {
     "average": 0.99985313415527,
     "min": 0.99945932626724,
     "variance": 1.7040544975089e-8
    },
    "location": {
     "top": 315,
     "left": 343,
     "width": 297,
     "height": 25
    },
    "word": "370441198907287001"
   },
   "key": {
    "probability": {
     "average": 0.99995613098145,
     "min": 0.99990141391754,
     "variance": 1.089595969006e-9
    },
    "location": {
     "top": 320,
     "left": 191,
     "width": 140,
     "height": 21
    },
    "word": "社会保障号码"
   }
  }
 ],
 "log_id": "8733452781125821952",
 "words_result_num": 5,
 "direction": 0
}
```


彩票识别

> **该接口的公有云服务即将下线，为避免影响您目前的业务使用，请您及时进行相关迁移工作。**

接口描述

支持对大乐透、双色球彩票票面文字内容进行识别，并按行返回结果。

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/lottery?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
+ location	是	object	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串
+ chars	否	array[]	单字符结果，recognize_granularity=small 时返回该字段
++ location	否	object	单字符结果的位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
++ char	否	string	单字符识别结果

返回示例


```
{
  "log_id": 3523983603,
  "words_result_num": 2,
  "words_result": [
    {
      "location": {
        "left": 35,
        "top": 53,
        "width": 193,
        "height": 109
      },
      "words": "感动",
      "chars": [
        {
          "location": {
            "left": 56,
            "top": 65,
            "width": 69,
            "height": 88
          },
          "char": "感"
        },
        {
          "location": {
            "left": 140,
            "top": 65,
            "width": 70,
            "height": 88
          },
          "char": "动"
        }
      ]
```

```
      }
    ]
}
```

通用文字识别（含生僻字版）

接口描述

【**该服务已停止更新且即将下线，如需更好的识别效果请使用通用文字识别（高精度版 / 高精度含位置版）**，此两项服务已扩充字库，可支持生僻字识别】字库范围更大，支持对图片中的生僻字进行识别

请求说明

请参考 [通用文字识别（高精度版）](https://ai.baidu.com/docs#/OCR-API-AccurateBasic/top) 或 [通用文字识别（高精度含位置版）](https://ai.baidu.com/docs#/OCR-API-Accurate/top)

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
direction	否	int32	图像方向，当 detect_direction=true 时返回该字段。 - -1：未定义， - 0：正向， - 1：逆时针90度， - 2：逆时针180度， - 3：逆时针270度
words_result	是	array[]	识别结果数组
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
+ words	否	string	识别结果字符串
probability	否	object	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值

返回示例

```JSON
{
"log_id": 2471272194,
"words_result_num": 2,
"words_result":
 [
  {"words": " TSINGTAO"},
  {"words": "青島啤酒"}
 ]
}
```

保险单识别

> **该接口已停止更新且即将下线，为避免影响您目前的业务使用，请您及时进行相关迁移工作**

接口描述

支持对保险单中的投保人、被保人、受益人的各项信息及保费、保险种类等字段进行识别，暂支持识别各类人身保险保单。

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链
kv_business	否	string	true/false	是否进行商业逻辑处理，true：进行商业逻辑处理，false：不进行商业逻辑处理，默认true

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/insurance_documents?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object{}	识别结果
+ InsBilCom	否	string	公司名称
+ InsBilNo	否	string	保险单号码
+ InsBilTim	否	string	保单生效日期
+ InsTolAmt	否	string	投保金额
+ InsCltGd1	否	string	投保人性别
+ InsCltNa1	否	string	投保人
+ InsBthDa1	否	string	投保人出生日期
+ InsIdcNb1	否	string	投保人证件号码
+ InsIdcTy1	否	string	投保人证件类型
+ InsPerLst	否	array[]	被保人信息
++ InsCltGd2	否	string	被保人性别
++ InsCltNa2	否	string	被保险人
++ InsBthDa2	否	string	被保险人出生日期
++ InsIdcNb2	否	string	被保险人证件号码
++ InsIdcTy2	否	string	被保险人证件类型
+ InsPrdList	否	array[]	保险信息
++ InsCovDur	否	string	保险期限
++ InsIcvAmt	否	string	基本保险金额
++ InsPayDur	否	string	交费期间
++ InsPayFeq	否	string	缴费频率
++ InsPerAmt	否	string	每期交费金额
++ InsPrdNam	否	string	产品名称
+ BenPerLst	否	array[]	受益人信息
++ BenCltNa	否	string	受益人姓名
++ BenPerPro	否	string	受益比例
++ BenPerOrd	否	string	受益顺序
++ BenPerTyp	否	string	受益人类型

返回示例

```
{
  "log_id": 5095868850706015333,
  "words_result_num": 11,
  "words_result": {
     "InsBilTim": "2015-02-16",
     "InsBthDa1": "1984-12-05",
     "InsTolAmt": "5000.00",
     "InsBilNo": "802600004350",
     "InsIdcTy1": "身份证",
     "InsIdcNb1": "43028119841205",
     "InsPerLst": [
        {
           "InsIdcTy2": "身份证",
           "InsCltNa2": "孙小宝",
           "InsIdcNb2": "430281199411054326",
           "InsBthDa2": "1994-11-05"
        }
     ],
     "InsCltNa1": "孙小宝",
     "InsPrdList": [
        {
           "InsPayFeq": "年交",
           "InsPerAmt": "5000.00",
           "InsPayDur": "5年",
           "InsCovDur": "20年",
           "InsPrdNam": "阳光人寿阳光康瑞年金保险",
           "InsIcvNum": "5"
        },
        {
           "InsCovDur": "1年",
           "InsPayFeq": "趸交",
           "InsIcvAmt": "375000.00",
           "InsPayDur": "1年",
           "InsPrdNam": "阳光人寿附加阳光康瑞重大疾病保险"
        }
     ],
     "BenPerLst": [
        {
           "BenPerTyp": "身故保险金受益人",
           "BenPerPro": "100%",
           "BenCltNa": "陈"
        }
     ],
     "InsBilCom": "阳光人寿保险股份有限公司"
  }
}
```


台湾通行证识别

> **该接口的公有云服务即将下线，若您仍需要识别台湾通行证，您可以选择使用港澳台证件识别接口。此接口功能和效果更佳，支持识别4类港澳台出入境证件识别，包含港澳通行证正/反面、台湾通行证正/反面、台胞证（台湾居民来往大陆通行证）正/反面、返乡证（港澳居民来往内地通行证）正/反面，同时支持识别以上4类证件的全部字段信息。**

接口描述

支持对大陆居民往来台湾通行证的证件号码、姓名、姓名拼音、出生日期、性别、有效期限、签发地点7个关键字段进行结构化识别。

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/taiwan_exitentrypermit?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果数组
+ Address	是	object	签发地点
+ Birthday	是	object	出生日期
+ CardNum	是	object	卡号
+ NameChn	是	object	姓名
+ NameEng	是	object	姓名拼音
+ Sex	是	object	性别
+ ValidDate	是	object	有效期限

返回示例

```
{
  "log_id": 424957212,
  "words_result": {
      "Address": {
          "words": "北京"
      },
      "Birthday": {
          "words": "19870405"
      },
      "CardNum": {
          "words": "L07379776"
      },
      "NameChn": {
          },
          "words": "陈孟"
      },
      "NameEng": {
          "words": "CHENMENG"
      },
      "Sex": {
          "words": "女"
      },
      "ValidDate": {
          "words": "20160116-20260115"
      },
  },
  "words_result_num": 7
}
```


港澳通行证识别

> **该接口的公有云服务即将下线，若您仍需要识别港澳通行证，您可以选择使用港澳台证件识别接口。此接口功能和效果更佳，支持识别4类港澳台出入境证件识别，包含港澳通行证正/反面、台湾通行证正/反面、台胞证（台湾居民来往大陆通行证）正/反面、返乡证（港澳居民来往内地通行证）正/反面，同时支持识别以上4类证件的全部字段信息。**

接口描述

支持对大陆居民往来港澳通行证的证件号码、姓名、姓名拼音、出生日期、性别、有效期限、签发地点7个关键字段进行结构化识别。

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，需去掉编码头（） 要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/HK_Macau_exitentrypermit?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ Address	是	object	签发地点
+ Birthday	是	object	出生日期
+ CardNum	是	object	卡号
+ NameChn	是	object	姓名
+ NameEng	是	object	姓名拼音
+ Sex	是	object	性别
+ ValidDate	是	object	有效期限

返回示例


```
{
  "log_id": 490699656,
  "words_result": {
      "Address": {
          "words": "山东"
      },
      "Birthday": {
          "words": "19900201"
      },
      "CardNum": {
          "words": "c10563465"
      },
      "NameChn": {
          "words": "陈露露"
      },
      "NameEng": {
          "words": "CHENLULU"
      },
      "Sex": {
          "words": "女"
      },
  "ValidDate": {
          "words": "20150116-20250115"
      },
  },
  "words_result_num": 7
}
```


名片识别

> **该接口已停止更新且即将下线，为避免影响您目前的业务使用，请您及时进行相关迁移工作**

接口描述

支持对各类名片的9个关键字段进行结构化识别，包括姓名、公司、职位、邮编、传真、电话、网址、地址、手机号。

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和image二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效 请注意关闭URL防盗链

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/business_card?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	object	识别结果
+ ADDR	是	array[]	地址
+ FAX	是	array[]	传真
+ MOBILE	是	array[]	手机
+ NAME	是	array[]	姓名
+ PC	是	array[]	邮编
+ URL	是	array[]	网址
+ TEL	是	array[]	电话
+ COMPANY	是	array[]	公司
+ TITLE	是	array[]	职位

返回示例


```
{
  "logid": "14815156840",
  "words_result_num": 7,
  "words_result": {
        "ADDR": ["中国·北京东城区朝阳门北大街4848号"],
        "FAX": ["010-818480043"],
        "MOBILE": ["18284584483"],
        "NAME": ["陈圆圆"],
        "PC": ["100010"],
        "URL": ["www.baidu.comwww.baidu.com"],
        "TEL": ["010-89184841"],
        "COMPANY": ["宝力机械有限公司"],
        "TITLE": ["总经理"]
  }
}
```


健康码识别

> **该接口已停止更新且即将下线，为避免影响您目前的业务使用，请您及时进行相关迁移工作**

接口描述

可识别国内各省市健康码截图或拍摄照片中的姓名、更新时间、健康状态、身份证号、核酸检测结果、核酸检测时间、核酸时间间隔、疫苗接种情况 8个字段内容。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/health_code?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的日志id，用于问题定位
result	是	object	识别结果
+ 姓名	是	array	健康码姓名
+ 更新时间	是	array	健康码更新时间
+ 状态	是	array	健康码状态
+ 身份证号	是	array	健康码身份证号
+ 核酸检测结果	是	array	健康码核酸检测结果
+ 核酸检测时间	是	array	健康码核酸检测时间
+ 核酸时间间隔	是	array	健康码时间间隔
+ 疫苗接种情况	是	array	健康码疫苗接种情况
++ words	是	string	字段识别内容
++ location	是	object	位置数组（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位位置的长方形的宽度
+++ height	是	uint32	表示定位位置的长方形的高度
error_code	是	uint64	错误码，error_code=0 表明识别无误，其他返回结果表示识别出错，详情请查看 https://ai.baidu.com/ai-doc/OCR/dk3h7y5vr
error_msg	是	string	错误信息

返回示例


```
{
  "姓名": [
    {
      "word": "**佳",
      "location": {
        "height": 58,
        "top": 164,
        "width": 98,
        "left": 329
      }
    }
  ],
  "状态": [
    {
      "word": "绿码",
      "location": {
        "height": 38,
        "top": 690,
        "width": 69,
        "left": 380
      }
    }
```

```
    ],
    "更新时间": [
        {
            "word": "2022-07-03 16:43:09",
            "location": {
                "height": 52,
                "top": 998,
                "width": 486,
                "left": 34
            }
        }
    ],
    "身份证号": [
        {
            "word": "411267198003157819",
            "location": {
                "height": 72,
                "top": 998,
                "width": 486,
                "left": 34
            }
        }
    ],
    "核酸检测结果": [
        {
            "word": "阴性",
            "location": {
                "height": 52,
                "top": 998,
                "width": 486,
                "left": 34
            }
        }
    ],
    "核酸检测时间": [
        {
            "word": "2022-07-03 16:43:09",
            "location": {
                "height": 52,
                "top": 998,
                "width": 486,
                "left": 34
            }
        }
    ],
    "核酸时间间隔": [
        {
            "word": "3天",
            "location": {
                "height": 52,
                "top": 998,
                "width": 486,
                "left": 34
            }
        }
    ],
    "疫苗接种情况": [
        {
            "word": "",
            "location": {
                "height": ,
                "top": ,
                "width": ,
```

```
            "left":
        }
    }
  ],
  "log_id": 1543887780872961511
}
```

核酸证明识别

> **该接口已停止更新且即将下线，为避免影响您目前的业务使用，请您及时进行相关迁移工作**

接口描述

可识别国内各省市电子核酸证明截图或拍摄照片中的姓名、检测时间、检测结果三个字段内容，暂不支持纸质核酸证明识别。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/covid_test?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的日志id，用于问题定位
result	是	object	识别结果
+ 姓名	是	array	核酸证明姓名
+ 检测时间	是	array	该核酸结果的检测时间，可用于与当前时间比对检测时效性
+ 检测结果	是	array	核酸检测结果，返回结果为“阴性/阳性/其他”，建议返回结果非“阴性”则进行人工校验
++ words	是	string	字段识别内容
++ location	是	object	位置数组（坐标0点为左上角）
+++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	是	uint32	表示定位位置的长方形的宽度
+++ height	是	uint32	表示定位位置的长方形的高度

返回示例

```
{
  "姓名": [
    {
      "word": "*佳",
      "location": {
        "height": 28,
        "top": 754,
        "width": 45,
        "left": 580
      }
    }
  ],
  "检测结果": [
    {
      "word": "阴性",
      "location": {
        "height": 73,
        "top": 441,
        "width": 178,
        "left": 304
      }
    }
  ],
  "检测时间": [
    {
      "word": "2022-06-30 18:31:30",
      "location": {
        "height": 24,
        "top": 1037,
        "width": 226,
        "left": 404
      }
    }
  ],
  "log_id": 1543890667962890449
}
```


通用票据识别

> **该接口已停止更新且即将下线，如需更好的识别效果，请使用 智能财务票据识别，此服务支持财务场景中13种常见票据的分类及结构化识别，支持多张不同种类票据在同一张图片上的混贴场景，可返回每张票据的位置、种类及票面信息的结构化识别结果。您也可以选择使用通用文字识别（高精度版），提供更高精度的识别服务，同时支持多语种识别。**

接口描述

针对票据字体做了专项优化的通用文字识别版本，支持对医疗票据、银行兑票、购物小票等各类票据的票面内容进行识别，并按行返回结果。

在线调试

您可以在 示例代码中心 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

请求说明

请求示例

HTTP 方法：`POST`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	和url/pdf_file/ofd_file 四选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file ，当image字段存在时，url、pdf_file、ofd_file 字段失效
url	和image/pdf_file/ofd_file 四选一	string	-	图片完整url，url长度不超过1024字节，url对应的图片base64编码后大小不超过8M，最短边至少15px，最长边最大4096px，支持jpg/jpeg/png/bmp格式 **优先级**：image > url > pdf_file > ofd_file，当image字段存在时，url字段失效 **请注意关闭URL防盗链**
pdf_file	和image/url/ofd_file 四选一	string	-	PDF文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url字段存在时，pdf_file字段失效
pdf_file_num	否	string	-	需要识别的PDF文件的对应页码，当 pdf_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
ofd_file	和image/url/pdf_file 四选一	string	-	OFD文件，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过8M，最短边至少15px，最长边最大4096px **优先级**：image > url > pdf_file > ofd_file，当image、url、pdf_file字段存在时，ofd_file字段失效
ofd_file_num	否	string	-	需要识别的OFD文件的对应页码，当 ofd_file 参数有效时，识别传入页码的对应页面内容，若不传入，则默认识别第 1 页
recognize_granularity	否	string	big/small	是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置
probability	否	string	true/false	是否返回识别结果中每一行的置信度
accuracy	否	string	normal/缺省	normal：使用快速服务；缺省或其它值：使用高精度服务
detect_direction	否	string	true/false	是否检测图像朝向，默认不检测，即：false。可选值包括： - true：检测朝向； - false：不检测朝向，朝向是指输入图像是正常方向、逆时针旋转90/180/270度

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

Bash

Python

JAVA

C++

PHP

C#

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/ocr/v1/receipt?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
words_result_num	是	uint32	识别结果数，表示words_result的元素个数
words_result	是	array[]	定位和识别结果数组
+ location	是	object{}	位置数组（坐标0点为左上角）
++ left	是	uint32	表示定位位置的长方形左上顶点的水平坐标
++ top	是	uint32	表示定位位置的长方形左上顶点的垂直坐标
++ width	是	uint32	表示定位位置的长方形的宽度
++ height	是	uint32	表示定位位置的长方形的高度
+ words	是	string	识别结果字符串
+ chars	否	array[]	单字符结果，recognize_granularity=small 时存在
++ char	否	string	单字符识别结果
++ location	否	object{}	位置数组（坐标0点为左上角）
+++ left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+++ top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+++ width	否	uint32	表示定位定位位置的长方形的宽度
+++ height	否	uint32	表示位置的长方形的高度
+ probability	否	float	识别结果中每一行的置信度值，包含average：行置信度平均值，variance：行置信度方差，min：行置信度最小值

返回示例


```
{
 "log_id": 2661573626,
"words_result_num": 1，
  "words_result": [
     {
        "location": {
           "left": 212,
           "top": 3,
           "width": 738,
           "height": 24
        },
        "words": "卡号/病案号:105353990标本编号:150139071送检科室:血液透析门诊病房",
     }
  ],
}
```


HTTP-SDK文档

Python语言

表格文字识别（同步接口）

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。

```python
    """ 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
url = "https://www.x.com/sample.jpg"

# 调用表格文字识别（同步接口）
res_image = client.form(image)
res_url = client.formUrl(url)
print(res_image)
print(res_url)

# 如果有可选参数
options = {}
options["table_border"] = "none"
res_image = client.form(image, options)
res_url = client.formUrl(url, options)
print(res_image)
print(res_url)
```

表格文字识别（同步接口） 请求参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
forms_result_num	是	uint32	识别结果元素个数
forms_result	是	array[]	识别结果
+ body	是	array[]	表格主体区域
+ footer	是	array[]	表格尾部区域信息
header	是	array[]	表格头部区域信息
vertexes_location	是	array[]	表格边界顶点

表格文字识别（同步接口） 返回示例

```
{
    "log_id": 3445697108,
    "forms_result_num": 1,
    "forms_result": [
        {
            "body": [
                {
                    "column": 0,
                    "probability": 0.99855202436447,
                    "row": 0,
                    "vertexes_location": [
                        {
                            "x": -2,
                            "y": 260
                        },
                        {
                            "x": 21,
                            "y": 244
                        },
                        {
                            "x": 35,
                            "y": 266
                        },
```

```
            {
                "x": 12,
                "y": 282
            }
        ],
        "words": "目"
    },
    {
        "column": 3,
        "probability": 0.99960500001907,
        "row": 5,
        "vertexes_location": [
            {
                "x": 603,
                "y": 52
            },
            {
                "x": 634,
                "y": 32
            },
            {
                "x": 646,
                "y": 50
            },
            {
                "x": 615,
                "y": 71
            }
        ],
        "words": "66"
    },
    {
        "column": 3,
        "probability": 0.99756097793579,
        "row": 6,
        "vertexes_location": [
            {
                "x": 634,
                "y": 73
            },
            {
                "x": 648,
                "y": 63
            },
            {
                "x": 657,
                "y": 77
            },
            {
                "x": 643,
                "y": 86
            }
        ],
        "words": "4"
    },
    {
        "column": 3,
        "probability": 0.96489900350571,
        "row": 10,
        "vertexes_location": [
            {
                "x": 699,
```

```
                "y": 178
            },
            {
                "x": 717,
                "y": 167
            },
            {
                "x": 727,
                "y": 183
            },
            {
                "x": 710,
                "y": 194
            }
        ],
        "words": "3,"
    },
    {
        "column": 3,
        "probability": 0.99809801578522,
        "row": 14,
        "vertexes_location": [
            {
                "x": 751,
                "y": 296
            },
            {
                "x": 786,
                "y": 273
            },
            {
                "x": 797,
                "y": 289
            },
            {
                "x": 761,
                "y": 312
            }
        ],
        "words": "206"
    }
],
"footer": [
    {
        "column": 0,
        "probability": 0.99853301048279,
        "row": 0,
        "vertexes_location": [
            {
                "x": 605,
                "y": 698
            },
            {
                "x": 632,
                "y": 680
            },
            {
                "x": 643,
                "y": 696
            },
            {
                "x": 616,
                "y": 714
```

```
                    "y": 714
                }
            ],
            "words": "22"
        }
    ],
    "header": [
        {
            "column": 0,
            "probability": 0.94802802801132,
            "row": 0,
            "vertexes_location": [
                {
                    "x": 183,
                    "y": 96
                },
                {
                    "x": 286,
                    "y": 29
                },
                {
                    "x": 301,
                    "y": 52
                },
                {
                    "x": 199,
                    "y": 120
                }
            ],
            "words": "29月"
        }
    ],
    "vertexes_location": [
        {
            "x": -154,
            "y": 286
        },
        {
            "x": 512,
            "y": -153
        },
        {
            "x": 953,
            "y": 513
        },
        {
            "x": 286,
            "y": 953
        }
    ]
}
]
}
```


表格文字识别(异步接口)--提交请求

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。表格文字识别接口为异步接口，分为两个API：提交请求接口、获取结果接口。

```
""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

# 调用表格文字识别(异步接口)--提交请求
res_image = client.tableRecognitionAsync(image)
print(res_image)
```

表格文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

表格文字识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	list	返回的结果列表
+request_id	是	string	该请求生成的request_id，后续使用该request_id获取识别结果

表格文字识别 返回示例

```
{
  "result" : [
    {
       "request_id" : "1234_6789"
    }
  ],
  "log_id":149689853984104
}
```

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

表格文字识别(异步接口)--获取结果

获取表格文字识别结果。

```
requestId = "23454320-23255"

# 调用表格文字识别(异步接口)--获取结果
res_image = client.getTableRecognitionResult(requestId)
print(res_image)

# 如果有可选参数
options = {}
options["result_type"] = "json"
res_image = client.getTableRecognitionResult(requestId, options)
print(res_image)
```

表格识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
request_id	是	string			发送表格文字识别请求时返回的request id
result_type	否	string	json excel	excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”

表格识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	object	返回的结果
+result_data	是	string	识别结果字符串，如果request_type是excel，则返回excel的文件下载地址，如果request_type是json，则返回json格式的字符串
+percent	是	int	表格识别进度（百分比）
+request_id	是	string	该图片对应请求的request_id
+ret_code	是	int	识别状态，1：任务未开始，2：进行中,3:已完成
+ret_msg	是	string	识别状态信息，任务未开始，进行中,已完成

表格识别结果 返回示例

成功应答示例：

```
{
  "result" :{
     "result_data" : "",
     "persent":100,
     "request_id": "149691317905102",
     "ret_code": 3
     "ret_msg": "已完成",
  },
  "log_id":149689853984104
}
```

当request_type为excel时，result_data格式样例为：

```
{
  "file_url":"https://ai.baidu.com/file/xxxfffddd"
}
```

当request_type为json时，result_data格式样例为：

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表头信息1",
      }
    ],
    "footer": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表尾信息1",
      }
    ],
    "body": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "单元格文字",
      }
    ]
  }
]
}
```

其中各个参数的说明(json方式返回结果时)：

字段	是否必选	类型	说明
form_num	是	int	表格数量（可能一张图片中包含多个表格）
forms	是	list	表格内容信息的列表
+header	是	list	每个表格中，表头数据的相关信息
+footer	是	list	表尾的相关信息
+body	是	list	表格主体部分的数据
++row	是	list	该单元格占据的行号
++column	是	list	该单元格占据的列号
++word	是	string	该单元格中的文字信息

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

表格识别接口

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口。

```
""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

# 调用表格识别
options = {}
options["result_type"] = "json"
res_image = client.tableRecognition(image, options)
print(res_image)
```

请求参数

tableRecognition(image, option, timeout)

参数名称	是否必选	类型	可选值范围	默认值	说明		
image	是	string			图片base64编码数据		
+result_type	是	string			json excel	excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”
timeout	是	number		10000	轮询tableGetresult接口获取数据的超时时间，单位毫秒		

返回参数表格识别结果接口返回相同

Java语言

表格文字识别同步接口

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.form(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.form(file, options);
      System.out.println(res.toString(2));

    }
```

表格文字识别同步接口 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据

表格文字识别同步接口 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
forms_result_num	是	number	
forms_result	是	array(object)	识别结果

表格文字识别同步接口 返回示例


```
{
    "log_id": 3445697108,
    "forms_result_num": 1,
    "forms_result": [
        {
            "body": [
                {
                    "column": 0,
                    "probability": 0.99855202436447,
                    "row": 0,
                    "vertexes_location": [
                        {
                            "x": -2,
                            "y": 260
                        },
                        {
                            "x": 21,
                            "y": 244
                        },
                        {
                            "x": 35,
                            "y": 266
                        },
                        {
                            "x": 12,
                            "y": 282
                        }
                    ],
                    "words": "目"
```

```
        },
        {
            "column": 3,
            "probability": 0.99960500001907,
            "row": 5,
            "vertexes_location": [
                {
                    "x": 603,
                    "y": 52
                },
                {
                    "x": 634,
                    "y": 32
                },
                {
                    "x": 646,
                    "y": 50
                },
                {
                    "x": 615,
                    "y": 71
                }
            ],
            "words": "66"
        },
        {
            "column": 3,
            "probability": 0.99756097793579,
            "row": 6,
            "vertexes_location": [
                {
                    "x": 634,
                    "y": 73
                },
                {
                    "x": 648,
                    "y": 63
                },
                {
                    "x": 657,
                    "y": 77
                },
                {
                    "x": 643,
                    "y": 86
                }
            ],
            "words": "4"
        },
        {
            "column": 3,
            "probability": 0.96489900350571,
            "row": 10,
            "vertexes_location": [
                {
                    "x": 699,
                    "y": 178
                },
                {
                    "x": 717,
                    "y": 167
                },
                {
```

```
                {
                    "x": 727,
                    "y": 183
                },
                {
                    "x": 710,
                    "y": 194
                }
            ],
            "words": "3,"
        },
        {
            "column": 3,
            "probability": 0.99809801578522,
            "row": 14,
            "vertexes_location": [
                {
                    "x": 751,
                    "y": 296
                },
                {
                    "x": 786,
                    "y": 273
                },
                {
                    "x": 797,
                    "y": 289
                },
                {
                    "x": 761,
                    "y": 312
                }
            ],
            "words": "206"
        }
    ],
    "footer": [
        {
            "column": 0,
            "probability": 0.99853301048279,
            "row": 0,
            "vertexes_location": [
                {
                    "x": 605,
                    "y": 698
                },
                {
                    "x": 632,
                    "y": 680
                },
                {
                    "x": 643,
                    "y": 696
                },
                {
                    "x": 616,
                    "y": 714
                }
            ],
            "words": "22"
        }
    ],
    "header": [
```

```
            {
                "column": 0,
                "probability": 0.94802802801132,
                "row": 0,
                "vertexes_location": [
                    {
                        "x": 183,
                        "y": 96
                    },
                    {
                        "x": 286,
                        "y": 29
                    },
                    {
                        "x": 301,
                        "y": 52
                    },
                    {
                        "x": 199,
                        "y": 120
                    }
                ],
                "words": "29月"
            }
        ],
        "vertexes_location": [
            {
                "x": -154,
                "y": 286
            },
            {
                "x": 512,
                "y": -153
            },
            {
                "x": 953,
                "y": 513
            },
            {
                "x": 286,
                "y": 953
            }
        ]
    }
]
}
```


表格文字识别

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。表格文字识别接口为异步接口，分为两个API：提交请求接口、获取结果接口。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();


      // 参数为本地图片路径
      String image = "test.jpg";
      JSONObject res = client.tableRecognitionAsync(image, options);
      System.out.println(res.toString(2));

      // 参数为本地图片二进制数组
      byte[] file = readImageFile(image);
      res = client.tableRecognitionAsync(file, options);
      System.out.println(res.toString(2));

    }
```

表格文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据

表格文字识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	list	返回的结果列表
+request_id	是	string	该请求生成的request_id，后续使用该request_id获取识别结果

表格文字识别 返回示例

```
{
  "result" : [
     {
        "request_id" : "1234_6789"
     }
  ],
  "log_id":149689853984104
}
```

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

表格识别结果

获取表格文字识别结果。

```
    public void sample(AipOcr client) {
      // 传入可选参数调用接口
      HashMap<String, String> options = new HashMap<String, String>();
      options.put("result_type", "json");

      String requestId = "23454320-23255";

      // 表格识别结果
      JSONObject res = client.tableResultGet(requestId, options);
      System.out.println(res.toString(2));

    }
```

表格识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
request_id	是	String			发送表格文字识别请求时返回的request id
result_type	否	String	json excel	excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”

表格识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	object	返回的结果
+result_data	是	string	识别结果字符串，如果request_type是excel，则返回excel的文件下载地址，如果request_type是json，则返回json格式的字符串
+percent	是	int	表格识别进度（百分比）
+request_id	是	string	该图片对应请求的request_id
+ret_code	是	int	识别状态，1：任务未开始，2：进行中,3:已完成
+ret_msg	是	string	识别状态信息，任务未开始，进行中,已完成

表格识别结果 返回示例

成功应答示例：

```
{
  "result" : {
     "result_data" : "",
     "persent":100,
     "request_id": "149691317905102",
     "ret_code": 3
     "ret_msg": "已完成",
  },
  "log_id":149689853984104
}
```

当request_type为excel时，result_data格式样例为：

```
 {
   "file_url":"https://ai.baidu.com/file/xxxfffddd"
}
```

当request_type为json时，result_data格式样例为：

```
{
   "form_num": 1,
   "forms": [
      {
         "header": [
            {
            "row": [
               1
            ],
            "column": [
               1,
               2
            ],
            "word": "表头信息1",
         }
      ],
      "footer": [
         {
            "row": [
               1
            ],
            "column": [
               1,
               2
            ],
            "word": "表尾信息1",
         }
      ],
      "body": [
         {
            "row": [
               1
            ],
            "column": [
               1,
               2
            ],
            "word": "单元格文字",
         }
      ]
   }
]
}
```

其中各个参数的说明(json方式返回结果时)：

字段	是否必选	类型	说明
form_num	是	int	表格数量（可能一张图片中包含多个表格）
forms	是	list	表格内容信息的列表
+header	是	list	每个表格中，表头数据的相关信息
+footer	是	list	表尾的相关信息
+body	是	list	表格主体部分的数据
++row	是	list	该单元格占据的行号
++column	是	list	该单元格占据的列号
++word	是	string	该单元格中的文字信息

失败应答示例（详细的错误码说明见本文档底部）：

```
{
    "log_id": 149319909347709,
    "error_code": 282000
    "error_msg":"internal error"
}
```

表格识别轮询接口

代码示例

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口。

```
public void tableRecognition(AipOcr client) {
  //异步接口

  //使用封装的同步轮询接口
  JSONObject jsonres = client.tableRecognizeToJson(file, 20000);
  System.out.println(jsonres.toString(2));

  JSONObject excelres = client.tableRecognizeToExcelUrl(file, 20000);
  System.out.println(excelres.toString(2));

}
```

请求参数

参数	类型	描述	是否必须
imgPath/imgData	byte[]	图像文件路径或二进制数据	是
timeoutMiliseconds	long	最长等待时间，超时将返回错误，一般任务在20s完成	是

返回参数与表格识别结果接口返回相同

PHP文档

表格文字识别同步接口

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。

```
 $image = file_get_contents('example.jpg');

// 调用表格文字识别同步接口
$client->form($image);
```

表格文字识别同步接口 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。

表格文字识别同步接口 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
forms_result_num	是	number	
forms_result	是	array(object)	识别结果

表格文字识别同步接口 返回示例


```
{
  "log_id": 3445697108,
  "forms_result_num": 1,
  "forms_result": [
    {
      "body": [
        {
          "column": 0,
          "probability": 0.99855202436447,
          "row": 0,
          "vertexes_location": [
            {
              "x": -2,
              "y": 260
            },
            {
              "x": 21,
              "y": 244
            },
            {
              "x": 35,
              "y": 266
            },
            {
              "x": 12,
              "y": 282
            }
          ],
          "words": "目"
        },
        {
          "column": 3,
          "probability": 0.99960500001907,
          "row": 5,
          "vertexes_location": [
            {
              "x": 603,
              "y": 52
            },
            {
              "x": 634,
              "y": 32
            },
            {
```

```
                {
                    "x": 646,
                    "y": 50
                },
                {
                    "x": 615,
                    "y": 71
                }
            ],
            "words": "66"
        },
        {
            "column": 3,
            "probability": 0.99756097793579,
            "row": 6,
            "vertexes_location": [
                {
                    "x": 634,
                    "y": 73
                },
                {
                    "x": 648,
                    "y": 63
                },
                {
                    "x": 657,
                    "y": 77
                },
                {
                    "x": 643,
                    "y": 86
                }
            ],
            "words": "4"
        },
        {
            "column": 3,
            "probability": 0.96489900350571,
            "row": 10,
            "vertexes_location": [
                {
                    "x": 699,
                    "y": 178
                },
                {
                    "x": 717,
                    "y": 167
                },
                {
                    "x": 727,
                    "y": 183
                },
                {
                    "x": 710,
                    "y": 194
                }
            ],
            "words": "3,"
        },
        {
            "column": 3,
            "probability": 0.99809801578522,
            "row": 14,
```

```
            "vertexes_location": [
              {
                "x": 751,
                "y": 296
              },
              {
                "x": 786,
                "y": 273
              },
              {
                "x": 797,
                "y": 289
              },
              {
                "x": 761,
                "y": 312
              }
            ],
            "words": "206"
          }
        ],
        "footer": [
          {
            "column": 0,
            "probability": 0.99853301048279,
            "row": 0,
            "vertexes_location": [
              {
                "x": 605,
                "y": 698
              },
              {
                "x": 632,
                "y": 680
              },
              {
                "x": 643,
                "y": 696
              },
              {
                "x": 616,
                "y": 714
              }
            ],
            "words": "22"
          }
        ],
        "header": [
          {
            "column": 0,
            "probability": 0.94802802801132,
            "row": 0,
            "vertexes_location": [
              {
                "x": 183,
                "y": 96
              },
              {
                "x": 286,
                "y": 29
              },
              {
```

```
                    "x": 301,
                    "y": 52
                },
                {
                    "x": 199,
                    "y": 120
                }
            ],
            "words": "29月"
        }
    ],
    "vertexes_location": [
        {
            "x": -154,
            "y": 286
        },
        {
            "x": 512,
            "y": -153
        },
        {
            "x": 953,
            "y": 513
        },
        {
            "x": 286,
            "y": 953
        }
    ]
}
}
```


表格文字识别

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。表格文字识别接口为异步接口，分为两个API：提交请求接口、获取结果接口。

```
$image = file_get_contents('example.jpg');

// 调用表格文字识别
$client->tableRecognitionAsync($image);
```

表格文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。

表格文字识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	list	返回的结果列表
+request_id	是	string	该请求生成的request_id，后续使用该request_id获取识别结果

表格文字识别 返回示例

```
{
  "result" : [
    {
       "request_id" : "1234_6789"
    }
  ],
  "log_id":149689853984104
}
```

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

表格识别结果

获取表格文字识别结果。

```
$requestId = "23454320-23255";

// 调用表格识别结果
$client->getTableRecognitionResult($requestId);

// 如果有可选参数
$options = array();
$options["result_type"] = "json";

// 带参数调用表格识别结果
$client->getTableRecognitionResult($requestId, $options);
```

表格识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
request_id	是	string			发送表格文字识别请求时返回的request id
result_type	否	string	json excel	excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”

表格识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	object	返回的结果
+result_data	是	string	识别结果字符串，如果request_type是excel，则返回excel的文件下载地址，如果request_type是json，则返回json格式的字符串
+percent	是	int	表格识别进度（百分比）
+request_id	是	string	该图片对应请求的request_id
+ret_code	是	int	识别状态，1：任务未开始，2：进行中,3:已完成
+ret_msg	是	string	识别状态信息，任务未开始，进行中,已完成

表格识别结果 返回示例

成功应答示例：

```
{
  "result" :{
     "result_data" : "",
     "persent":100,
     "request_id": "149691317905102",
     "ret_code": 3
     "ret_msg": "已完成",
  },
  "log_id":149689853984104
}
```

当request_type为excel时，result_data格式样例为：

```
{
  "file_url":"https://ai.baidu.com/file/xxxfffddd"
}
```

当request_type为json时，result_data格式样例为：

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表头信息1",
      }
    ],
    "footer": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表尾信息1",
      }
    ],
    "body": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "单元格文字",
      }
    ]
  }
]
}
```

其中各个参数的说明(json方式返回结果时)：

字段	是否必选	类型	说明
form_num	是	int	表格数量（可能一张图片中包含多个表格）
forms	是	list	表格内容信息的列表
+header	是	list	每个表格中，表头数据的相关信息
+footer	是	list	表尾的相关信息
+body	是	list	表格主体部分的数据
++row	是	list	该单元格占据的行号
++column	是	list	该单元格占据的列号
++word	是	string	该单元格中的文字信息

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

表格识别接口

代码示例

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口

```
// 同步表格识别 返回识别结果
$client->tableRecognition(
    file_get_contents('table.jpg'),
    array(
        'result_type' => 'json',
    )
);
```

请求参数

参数名称	是否必选	类型	可选值范围	默认值	说明		
image	是	string			图片base64编码数据 **注：** SDK已经将API接口封装，可按照示例直接传入本地图片路径，如需传值，请传入图片的二进制数据，SDK会自行base64编码。		
+result_type	是	string			json excel	excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”
timeout	是	number		10000	轮询tableGetresult接口获取数据的超时时间，单位毫秒		

返回参数与表格识别结果接口返回相同

C#语言

表格文字识别同步接口

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。

```
public void FormDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用表格文字识别同步接口，可能会抛出网络等异常，请使用try/catch捕获
var result = client.Form(image);
Console.WriteLine(result);
}
```

表格文字识别同步接口 请求参数详情

参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据

表格文字识别同步接口 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
forms_result_num	是	number	
forms_result	是	array(object)	识别结果

表格文字识别同步接口 返回示例


```
{
  "log_id": 3445697108,
  "forms_result_num": 1,
  "forms_result": [
    {
      "body": [
        {
          "column": 0,
          "probability": 0.99855202436447,
          "row": 0,
          "vertexes_location": [
            {
              "x": -2,
              "y": 260
            },
            {
              "x": 21,
              "y": 244
            },
            {
              "x": 35,
              "y": 266
            },
            {
              "x": 12,
              "y": 282
            }
          ],
          "words": "目"
        },
        {
          "column": 3,
          "probability": 0.99960500001907,
          "row": 5,
          "vertexes_location": [
            {
              "x": 603,
              "y": 52
            },
            {
              "x": 634,
              "y": 32
            },
            {
              "x": 646,
              "y": 50
            },
            {
              "x": 615,
              "y": 71
            }
          ],
          "words": "66"
        }
```

```
        },
        {
            "column": 3,
            "probability": 0.99756097793579,
            "row": 6,
            "vertexes_location": [
                {
                    "x": 634,
                    "y": 73
                },
                {
                    "x": 648,
                    "y": 63
                },
                {
                    "x": 657,
                    "y": 77
                },
                {
                    "x": 643,
                    "y": 86
                }
            ],
            "words": "4"
        },
        {
            "column": 3,
            "probability": 0.96489900350571,
            "row": 10,
            "vertexes_location": [
                {
                    "x": 699,
                    "y": 178
                },
                {
                    "x": 717,
                    "y": 167
                },
                {
                    "x": 727,
                    "y": 183
                },
                {
                    "x": 710,
                    "y": 194
                }
            ],
            "words": "3,"
        },
        {
            "column": 3,
            "probability": 0.99809801578522,
            "row": 14,
            "vertexes_location": [
                {
                    "x": 751,
                    "y": 296
                },
                {
                    "x": 786,
                    "y": 273
                },
                {
```

```
                {
                    "x": 797,
                    "y": 289
                },
                {
                    "x": 761,
                    "y": 312
                }
            ],
            "words": "206"
        }
    ],
    "footer": [
        {
            "column": 0,
            "probability": 0.99853301048279,
            "row": 0,
            "vertexes_location": [
                {
                    "x": 605,
                    "y": 698
                },
                {
                    "x": 632,
                    "y": 680
                },
                {
                    "x": 643,
                    "y": 696
                },
                {
                    "x": 616,
                    "y": 714
                }
            ],
            "words": "22"
        }
    ],
    "header": [
        {
            "column": 0,
            "probability": 0.94802802801132,
            "row": 0,
            "vertexes_location": [
                {
                    "x": 183,
                    "y": 96
                },
                {
                    "x": 286,
                    "y": 29
                },
                {
                    "x": 301,
                    "y": 52
                },
                {
                    "x": 199,
                    "y": 120
                }
            ],
            "words": "29月"
        }
```

```
        ],
        "vertexes_location": [
          {
            "x": -154,
            "y": 286
          },
          {
            "x": 512,
            "y": -153
          },
          {
            "x": 953,
            "y": 513
          },
          {
            "x": 286,
            "y": 953
          }
        ]
      }
    ]
}
```

表格文字识别

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。表格文字识别接口为异步接口，分为两个API：提交请求接口、获取结果接口。

```
 public void TableRecognitionRequestDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用表格文字识别，可能会抛出网络等异常，请使用try/catch捕获
var result = client.TableRecognitionRequest(image);
Console.WriteLine(result);
}
```

表格文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据

表格文字识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	list	返回的结果列表
+request_id	是	string	该请求生成的request_id，后续使用该request_id获取识别结果

表格文字识别 返回示例

```
 {
   "result" : [
      {
         "request_id" : "1234_6789"
      }
   ],
   "log_id":149689853984104
}
```

失败应答示例（详细的错误码说明见本文档底部）：

```
 {
   "log_id": 149319909347709,
   "error_code": 282000
   "error_msg":"internal error"
}
```

表格识别结果

获取表格文字识别结果

```
 public void TableRecognitionGetResultDemo() {
var requestId = "23454320-23255";

// 调用表格识别结果，可能会抛出网络等异常，请使用try/catch捕获
var result = client.TableRecognitionGetResult(requestId);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{
   {"result_type", "json"}
};
// 带参数调用表格识别结果
result = client.TableRecognitionGetResult(requestId, options);
Console.WriteLine(result);
}
```

表格识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
request_id	是	string			发送表格文字识别请求时返回的request id
result_type	否	string	json excel	excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”

表格识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	object	返回的结果
+result_data	是	string	识别结果字符串，如果request_type是excel，则返回excel的文件下载地址，如果request_type是json，则返回json格式的字符串
+percent	是	int	表格识别进度（百分比）
+request_id	是	string	该图片对应请求的request_id
+ret_code	是	int	识别状态，1：任务未开始，2：进行中,3:已完成
+ret_msg	是	string	识别状态信息，任务未开始，进行中,已完成

表格识别结果 返回示例

成功应答示例：

```
 {
   "result" : {
      "result_data" : "",
      "persent":100,
      "request_id": "149691317905102",
      "ret_code": 3
      "ret_msg": "已完成",
   },
   "log_id":149689853984104
}
```

当request_type为excel时，result_data格式样例为：

```
 {
   "file_url":"https://ai.baidu.com/file/xxxfffddd"
}
```

当request_type为json时，result_data格式样例为：

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表头信息1",
      }
    ],
    "footer": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表尾信息1",
      }
    ],
    "body": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "单元格文字",
      }
    ]
  }
]
}
```

其中各个参数的说明(json方式返回结果时)：

字段	是否必选	类型	说明
form_num	是	int	表格数量（可能一张图片中包含多个表格）
forms	是	list	表格内容信息的列表
+header	是	list	每个表格中，表头数据的相关信息
+footer	是	list	表尾的相关信息
+body	是	list	表格主体部分的数据
++row	是	list	该单元格占据的行号
++column	是	list	该单元格占据的列号
++word	是	string	该单元格中的文字信息

失败应答示例（详细的错误码说明见本文档底部）：

```
 {
   "log_id": 149319909347709,
   "error_code": 282000
   "error_msg":"internal error"
}
```

表格识别轮询接口

代码示例

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口

```
 var image = File.ReadAllBytes("图片文件路径");
// 识别为excel文件
var result = client.TableRecognitionToExcel(image, 30000);
Console.Write(result);

// 识别为json结果
result = client.TableRecognitionToJson(image, 30000);
Console.Write(result);
```

请求参数

参数	类型	描述	是否必须
image	byte[]	图像二进制数据	是
timeoutMiliseconds	long	最长等待时间，超时将返回错误, 一般任务在20s完成	是

返回参数

与**表格识别结果**接口返回相同

Node.js语言

表格文字识别同步接口

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。

```
 var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用表格文字识别同步接口
client.form(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

表格文字识别同步接口 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

表格文字识别同步接口 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
forms_result_num	是	number	
forms_result	是	array(object)	识别结果

表格文字识别同步接口 返回示例


```
{
  "log_id": 3445697108,
  "forms_result_num": 1,
  "forms_result": [
    {
      "body": [
        {
          "column": 0,
          "probability": 0.99855202436447,
          "row": 0,
          "vertexes_location": [
            {
              "x": -2,
              "y": 260
            },
            {
              "x": 21,
              "y": 244
            },
            {
              "x": 35,
              "y": 266
            },
            {
              "x": 12,
              "y": 282
            }
          ],
          "words": "目"
        },
        {
          "column": 3,
          "probability": 0.99960500001907,
          "row": 5,
          "vertexes_location": [
            {
              "x": 603,
              "y": 52
            },
            {
              "x": 634,
              "y": 32
            },
            {
              "x": 646,
              "y": 50
            },
            {
              "x": 615,
              "y": 71
            }
          ],
          "words": "66"
        }
```

```
        },
        {
            "column": 3,
            "probability": 0.99756097793579,
            "row": 6,
            "vertexes_location": [
                {
                    "x": 634,
                    "y": 73
                },
                {
                    "x": 648,
                    "y": 63
                },
                {
                    "x": 657,
                    "y": 77
                },
                {
                    "x": 643,
                    "y": 86
                }
            ],
            "words": "4"
        },
        {
            "column": 3,
            "probability": 0.96489900350571,
            "row": 10,
            "vertexes_location": [
                {
                    "x": 699,
                    "y": 178
                },
                {
                    "x": 717,
                    "y": 167
                },
                {
                    "x": 727,
                    "y": 183
                },
                {
                    "x": 710,
                    "y": 194
                }
            ],
            "words": "3,"
        },
        {
            "column": 3,
            "probability": 0.99809801578522,
            "row": 14,
            "vertexes_location": [
                {
                    "x": 751,
                    "y": 296
                },
                {
                    "x": 786,
                    "y": 273
                },
                {
```

```
                    "x": 797,
                    "y": 289
                },
                {
                    "x": 761,
                    "y": 312
                }
            ],
            "words": "206"
        }
    ],
    "footer": [
        {
            "column": 0,
            "probability": 0.99853301048279,
            "row": 0,
            "vertexes_location": [
                {
                    "x": 605,
                    "y": 698
                },
                {
                    "x": 632,
                    "y": 680
                },
                {
                    "x": 643,
                    "y": 696
                },
                {
                    "x": 616,
                    "y": 714
                }
            ],
            "words": "22"
        }
    ],
    "header": [
        {
            "column": 0,
            "probability": 0.94802802801132,
            "row": 0,
            "vertexes_location": [
                {
                    "x": 183,
                    "y": 96
                },
                {
                    "x": 286,
                    "y": 29
                },
                {
                    "x": 301,
                    "y": 52
                },
                {
                    "x": 199,
                    "y": 120
                }
            ],
            "words": "29月"
        }
```

```
            ],
            "vertexes_location": [
                {
                    "x": -154,
                    "y": 286
                },
                {
                    "x": 512,
                    "y": -153
                },
                {
                    "x": 953,
                    "y": 513
                },
                {
                    "x": 286,
                    "y": 953
                }
            ]
        }
    ]
}
```

表格文字识别

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。表格文字识别接口为异步接口，分为两个API：提交请求接口、获取结果接口。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用表格文字识别
client.tableBegin(image).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

表格文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式

表格文字识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	list	返回的结果列表
+request_id	是	string	该请求生成的request_id，后续使用该request_id获取识别结果

表格文字识别 返回示例

```
{
  "result" : [
    {
       "request_id" : "1234_6789"
    }
  ],
  "log_id":149689853984104
}
```

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

表格识别结果

获取表格文字识别结果

```
var requestId = "23454320-23255";

// 调用表格识别结果
client.tableGetresult(requestId).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});

// 如果有可选参数
var options = {};
options["result_type"] = "json";

// 带参数调用表格识别结果
client.tableGetresult(requestId, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

表格识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
request_id	是	string			发送表格文字识别请求时返回的request id
result_type	否	string	json excel	excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”

表格识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	object	返回的结果
+result_data	是	string	识别结果字符串，如果request_type是excel，则返回excel的文件下载地址，如果request_type是json，则返回json格式的字符串
+percent	是	int	表格识别进度（百分比）
+request_id	是	string	该图片对应请求的request_id
+ret_code	是	int	识别状态，1：任务未开始，2：进行中,3:已完成
+ret_msg	是	string	识别状态信息，任务未开始，进行中,已完成

表格识别结果 返回示例

成功应答示例：

```
{
  "result" :{
     "result_data" : "",
     "persent":100,
     "request_id": "149691317905102",
     "ret_code": 3
     "ret_msg": "已完成",
  },
  "log_id":149689853984104
}
```

当request_type为excel时，result_data格式样例为：

```
{
  "file_url":"https://ai.baidu.com/file/xxxfffddd"
}
```

当request_type为json时，result_data格式样例为：

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表头信息1",
      }
    ],
    "footer": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表尾信息1",
      }
    ],
    "body": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "单元格文字",
      }
    ]
  }
]
}
```

其中各个参数的说明(json方式返回结果时)：

字段	是否必选	类型	说明
form_num	是	int	表格数量（可能一张图片中包含多个表格）
forms	是	list	表格内容信息的列表
+header	是	list	每个表格中，表头数据的相关信息
+footer	是	list	表尾的相关信息
+body	是	list	表格主体部分的数据
++row	是	list	该单元格占据的行号
++column	是	list	该单元格占据的列号
++word	是	string	该单元格中的文字信息

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

表格识别接口

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口。

```
// 图片文件base64编码
var fs = require('fs');
var base64 = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');

// 以json格式获取表格识别结果，10秒的超时限制
client.tableRecorgnize(base64, 'json', 10000).then(function(result) {
    console.log('<tableRecorgnize>: ' + JSON.stringify(result));
}).catch(function(e){
    console.log(e)
});

// 以excel格式获取表格识别结果，10秒的超时限制，还可以设置每次轮询间隔1秒
client.tableRecorgnize(base64, 'excel', 10000, 1000).then(function(result) {
    console.log('<tableRecorgnize>: ' + JSON.stringify(result));
}).catch(function(e){
    console.log(e)
});
```

请求参数

tableRecorgnize(image, type, timeout, interval)

参数名称	是否必选	类型	可选值范围	默认值	说明		
image	是	string			图片base64编码数据		
type	是	string			json excel	excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”
timeout	是	number		20000	轮询tableGetresult接口获取数据的超时时间，单位毫秒		
interval	是	number		2000	轮询tableGetresult接口获取数据的间隔，单位毫秒		

返回参数与表格识别结果接口返回相同

C++语言

表格文字识别同步接口

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用表格文字识别同步接口
result = client.form(image, aip::null);
```

表格文字识别同步接口 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取

表格文字识别同步接口 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
forms_result_num	是	number	
forms_result	是	array(object)	识别结果

表格文字识别同步接口 返回示例


```
{
  "log_id": 3445697108,
  "forms_result_num": 1,
  "forms_result": [
    {
      "body": [
        {
          "column": 0,
          "probability": 0.99855202436447,
          "row": 0,
          "vertexes_location": [
            {
              "x": -2,
              "y": 260
            },
            {
              "x": 21,
              "y": 244
            },
            {
              "x": 35,
              "y": 266
            },
            {
              "x": 12,
              "y": 282
            }
          ],
          "words": "目"
        },
        {
          "column": 3,
          "probability": 0.99960500001907,
          "row": 5,
          "vertexes_location": [
            {
              "x": 603,
              "y": 52
```

```
            },
            {
                "x": 634,
                "y": 32
            },
            {
                "x": 646,
                "y": 50
            },
            {
                "x": 615,
                "y": 71
            }
        ],
        "words": "66"
    },
    {
        "column": 3,
        "probability": 0.99756097793579,
        "row": 6,
        "vertexes_location": [
            {
                "x": 634,
                "y": 73
            },
            {
                "x": 648,
                "y": 63
            },
            {
                "x": 657,
                "y": 77
            },
            {
                "x": 643,
                "y": 86
            }
        ],
        "words": "4"
    },
    {
        "column": 3,
        "probability": 0.96489900350571,
        "row": 10,
        "vertexes_location": [
            {
                "x": 699,
                "y": 178
            },
            {
                "x": 717,
                "y": 167
            },
            {
                "x": 727,
                "y": 183
            },
            {
                "x": 710,
                "y": 194
            }
        ],
        "words": "3."
```

```
            "words": 3,
        },
        {
            "column": 3,
            "probability": 0.99809801578522,
            "row": 14,
            "vertexes_location": [
                {
                    "x": 751,
                    "y": 296
                },
                {
                    "x": 786,
                    "y": 273
                },
                {
                    "x": 797,
                    "y": 289
                },
                {
                    "x": 761,
                    "y": 312
                }
            ],
            "words": "206"
        }
    ],
    "footer": [
        {
            "column": 0,
            "probability": 0.99853301048279,
            "row": 0,
            "vertexes_location": [
                {
                    "x": 605,
                    "y": 698
                },
                {
                    "x": 632,
                    "y": 680
                },
                {
                    "x": 643,
                    "y": 696
                },
                {
                    "x": 616,
                    "y": 714
                }
            ],
            "words": "22"
        }
    ],
    "header": [
        {
            "column": 0,
            "probability": 0.94802802801132,
            "row": 0,
            "vertexes_location": [
                {
                    "x": 183,
                    "y": 96
                },
```

```
                {
                    "x": 286,
                    "y": 29
                },
                {
                    "x": 301,
                    "y": 52
                },
                {
                    "x": 199,
                    "y": 120
                }
            ],
            "words": "29月"
        }
    ],
    "vertexes_location": [
        {
            "x": -154,
            "y": 286
        },
        {
            "x": 512,
            "y": -153
        },
        {
            "x": 953,
            "y": 513
        },
        {
            "x": 286,
            "y": 953
        }
    ]
  }
]
}
```


表格文字识别

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。表格文字识别接口为异步接口，分为两个API：提交请求接口、获取结果接口。

```
 Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用表格文字识别
result = client.table_recognize(image, aip::null);
```

表格文字识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取

表格文字识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	list	返回的结果列表
+request_id	是	string	该请求生成的request_id，后续使用该request_id获取识别结果

表格文字识别 返回示例

```
{
  "result" : [
    {
       "request_id" : "1234_6789"
    }
  ],
  "log_id":149689853984104
}
```

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

表格识别结果

获取表格文字识别结果

```
 Json::Value result;

std::string request_id = "23454320-23255";

// 调用表格识别结果
result = client.table_result_get(request_id, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["result_type"] = "json";

// 带参数调用表格识别结果
result = client.table_result_get(request_id, options);
```

表格识别结果 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
request_id	是	std::string			发送表格文字识别请求时返回的request id
result_type	否	std::string	json excel	excel	期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为”excel”

表格识别结果 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	long	唯一的log id，用于问题定位
result	是	object	返回的结果
+result_data	是	string	识别结果字符串，如果request_type是excel，则返回excel的文件下载地址，如果request_type是json，则返回json格式的字符串
+percent	是	int	表格识别进度（百分比）
+request_id	是	string	该图片对应请求的request_id
+ret_code	是	int	识别状态，1：任务未开始，2：进行中,3:已完成
+ret_msg	是	string	识别状态信息，任务未开始，进行中,已完成

表格识别结果 返回示例

成功应答示例：

```
{
  "result" :{
     "result_data" : "",
     "persent":100,
     "request_id": "149691317905102",
     "ret_code": 3
     "ret_msg": "已完成",
  },
  "log_id":149689853984104
}
```

当request_type为excel时，result_data格式样例为：

```
{
  "file_url":"https://ai.baidu.com/file/xxxfffddd"
}
```

当request_type为json时，result_data格式样例为：

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表头信息1",
        }
      ],
      "footer": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "表尾信息1",
        }
      ],
      "body": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "单元格文字",
        }
      ]
    }
]
}
```


其中各个参数的说明(json方式返回结果时)：

字段	是否必选	类型	说明
form_num	是	int	表格数量（可能一张图片中包含多个表格）
forms	是	list	表格内容信息的列表
+header	是	list	每个表格中，表头数据的相关信息
+footer	是	list	表尾的相关信息
+body	是	list	表格主体部分的数据
++row	是	list	该单元格占据的行号
++column	是	list	该单元格占据的列号
++word	是	string	该单元格中的文字信息

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

文档解析（旧接口）

接口描述

支持对doc、pdf、图片、xlsx等16种格式文档进行解析，输出文档的版面、表格、阅读顺序、标题层级、旋转角度等信息，将非结构化数据转化为易于处理的结构化数据，识别准确率可达 90% 以上。

该接口正在公测中，完成个人/企业认证的用户可领取200页免费额度，如需申请更多额度或者QPS，请合作咨询，或者提交工单。

文档解析为**异步接口**，需要先调用**提交请求接口**获取 task_id，然后调用**获取结果接口**进行结果轮询，建议提交请求后 5～10 秒轮询。**提交请求接口**QPS为2，**获取结果接口**QPS为10。

提交请求接口

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/file/2.0/brain/online/v1/parser/task`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	multipart/form-data

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
file	和file_url二选一	file	-	文件数据，支持的文件类型： -版式文档：pdf、jpg、jpeg、png、bmp、tif、tiff、ofd、ppt、pptx -流式文档：doc、docx、txt、xls、xlsx、wps 文档大小不超过50M，文档页数不超过1000页（流式文档按2000字算一页） **优先级：** file > file_url，当file字段存在时，file_url字段失效
file_url	和file二选一	string	-	文件完整URL，仅支持北京区域的BOS公网访问，URL长度不超过1024字节，支持格式与file一致，仅支持上传1篇文件，文件大小不超过50M **优先级：** file > file_url，当file字段存在时，file_url字段失效 **请注意关闭URL防盗链**
file_name	是	string	-	文件名，请保证文件名后缀正确，例如 "1.pdf "
return_para_nodes	否	bool	true/false	是否返回标题层级段落树。默认为false

请求代码示例

提示：使用示例代码前，请记得替换其中的示例Token、文档地址或Base64信息。

Python

```
import requests
import os


def create_task(url, file_path, file_url):
    """
    Args:
        url: string, 服务请求链接
        file_path: 本地文件路径
        file_url: 文件链接
    Returns: 响应
    """
    # 文件请求
    body = {
        "file": (os.path.basename(file_path), open(file_path, 'rb'), "multipart/form-data"),
    }

    # 文件链接请求
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ task_id	string	该请求生成的task_id，后续使用该task_id获取审查结果

返回示例

成功返回示例：

```
{
    "error_code": 0,
    "error_msg": "",
    "log_id": "10138598131137362685273505665433",
    "result": {
        "task_id": "task-3zy9Bg8CHt1M4pPOcX2q5bg28j26801S"
    }
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
{
    "error_code": 282003,
    "error_msg": "missing parameters",
    "log_id": "375076310335855445079832539241 41",
    "result": "null"
}
```

获取结果接口

请求说明

请求示例

HTTP 方法：`POST`

请求URL： `https://aip.baidubce.com/file/2.0/brain/online/v1/parser/task/query`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	multipart/form-data

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
task_id	是	string	发送提交请求时返回的task_id

请求代码示例

提示：使用示例代码前，请记得替换其中的示例Token、task_id。

Python

```
import requests

def query_task(url, task_id):
    """
    Args:
        url: string, 请求链接
        task_id: string, task id
    Returns: 响应
    """
    data = {
        "task_id": task_id
    }

    response = requests.post(url, data=data, files=data)
    return response


if __name__ == '__main__':
```

返回说明

返回参数

字段	类型	说明
log_id	uint64	唯一的log id，用于问题定位
error_code	int	错误码
error_msg	string	错误描述信息
result	dict	返回的结果列表
+ task_id	string	任务ID
+ status	string	任务状态，pending：排队中；running：运行中；success：成功；failed：失败
+ task_error	string	解析报错信息，包含任务失败、额度不够
+ duration	string	任务执行时长
+ parse_result_url	string	文档解析结果的bos链接

可通过parse_result_url下载解析结果的JSON文件，parse_result_url的返回参数如下：

字段	类型	说明
file_name	string	文档名称
file_content	list	文档解析的内容
+ page_num	int	页码，从0开始
+ page_size	dict	页面大小，版式格式时有效
++ width	float	页面宽度，版式格式时有效
++ height	float	页面高度，版式格式时有效
+ page_angle	int	页面旋转角度，版式格式时有效
+ is_scan	bool	是否为扫描件
+ page_content	dict	文档单页的解析内容
++ layout	string	页面内layout布局数据
+++ node_id	int	和para_node中的node_id对应
+++ box	list	边框数据 「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点），版式格式时有效
+++ type	string	布局类型 「text」- 段落、「image」- 图片、「table」- 表格、「head_tail」- 页眉页脚、「contents」- 目录、「cell」- 单元格（仅表格内才有）
+++ text	string	布局内文字信息
+++ children	list	布局嵌套数据, 当前layout type为table时有值，列表内部数据结构同layout ，为单元格的内容
++++ box	list	边框数据 「x, y, w, h」(x, y)为坐标点坐标，w为box宽度，h为box高度（以页面坐标为原点），版式格式时有效
++++ type	string	「cell」- 单元格（仅表格内才有）
++++ text	string	表格单元格内文字
+++ matrix	list	二维数组 表示表格内部合并单元格信息，「table类型layout返回，见table layout示例」
+++ merge_table	string	表格结构才有该字段，「begin」- 跨页表格开始、「inner」- 跨页表格中间表格(表格跨页超过两页)、「end」- 跨页表格结束
++ sheet_name	string	excel sheet表名
++ type	string	页面属性 「text」- 正文、「contents」- 目录、「appendix」- 附录、「others」- 其他
para_nodes	list	文章段落标题层级结构树，「return_para_nodes=True」时返回
+ node_id	int	节点id（从1开始计数，存在root节点id为0）
+ text	string	节点对应文本内容
+ node_type	string	节点类型 样举值「root、title、text、image、table、head_tail、contents」
+ parent	int	最近父节点node_id

+ children	list	子节点 node_id 数组
+ para_type	string	标题类型，当node*type为title时，固定格式 title*{int}（如title_1、title_2）对应标题层数，其余情况同node_type
+ position	list	节点对应在文档中的位置信息，包含layout的位置信息。列表形式，每个元素为一个layout的位置信息
++ pageno	int	对应文档页码
++ layout_index	int	layout在文档当前页中的索引
++ box	list	layout在文档当前页中的外接矩形的坐标[x, y, w, h]

返回示例

成功返回示例：


```
{
  "log_id": "2359659789928692176157936558237 3",
  "error_code": 0,
  "error_msg": "",
  "result":
  {
     "task_id": "task-UnvGsgbYZp9pS3BZRHn11ifzjNvKzTgf",
     "status": "success",
     "task_error": null,
     "duration": 902.0,
     "parse_result_url": "https:xxxxxxxxxxxxxxxxxx"
  }
}
```


解析结果示例：


```
{
  "file_name": "示例文件.pdf",
  "para_nodes": [
     {
        "node_id": 0,
        "text": "",
        "node_type": "root",
        "parent": null,
        "children": [1],
        "para_type": "root",
        "position": []
     },
     {
        "node_id": 1,
        "text": "建构大模型智能审查方案",
        "node_type": "title",
        "parent": 0,
        "children": [2],
        "para_type": "title_1",
        "position": [
           {
              "pageno": 0,
              "layout_index": 0,
              "box": [164, 115, 288, 28]
           }
        ]
     },
```

```
        {
            "node_id": 2,
            "text": "阿德勒的主张",
            "node_type": "title",
            "parent": 1,
            "children": [3],
            "para_type": "title_2",
            "position": [
                {
                    "pageno": 0,
                    "layout_index": 1,
                    "box": [79, 175, 271, 15]
                }
            ]
        },
        {
            "node_id": 3,
            "text": "阿德勒是个体心理学的创始人，他的理论强调个体在社会关系中的自我提升和归属感。",
            "node_type": "text",
            "parent": 2,
            "children": [],
            "para_type": "text",
            "position": [
                {
                    "pageno": 0,
                    "layout_index": 2,
                    "box": [79, 224, 441, 44]
                }
            ]
        }
    ],
    "file_content": [
        {
            "page_num": 0,
            "page_size": {
                "width": 612,
                "height": 792
            },
            "page_angle": 0,
            "is_scan": false,
            "page_content": {
                "layout": [
                    {
                        "box": [164, 115, 288, 28],
                        "type": "text",
                        "text": "建构大模型智能审查方案",
                        "node_id": 1
                    },
                    {
                        "box": [79, 175, 271, 15],
                        "type": "text",
                        "text": "阿德勒的主张",
                        "node_id": 2
                    },
                    {
                        "box": [79, 224, 441, 44],
                        "type": "text",
                        "text": "阿德勒是个体心理学的创始人，他的理论强调个体在社会关系中的自我提升和归属感。",
                        "node_id": 3
                    }
                ]
            }
        }
```

```
    ]
}
```

失败返回示例（详细的错误码说明见API文档-错误码）：

```
 {"log_id": "136650910387425038671085132476 08",
"error_code": "282007",
"error_msg": "task not exist, please check task id",
"result": "null"}
```