百度智能云

SPEECH 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
产品简介 5
接口能力 5
产品更新动态 6
快速开发指南 12
资源领取&应用创建 12
如何用可视化工具调用 16
如何用代码调用 19
如何集成SDK 22
语音识别SDK 22
语音合成SDK 34
用量及历史调用查询 48
常用音频处理工具 49
音频文件转码 49
拼音相似度比较 55
开源VAD音频切分工具 58
购买指南 59
计费概述 59
免费测试资源 60
产品价格 63
端到端语音语言大模型 63
大模型声音复刻 65
大模型声音复刻 65
语音合成 67
语音识别 76
如何购买 82
大模型语音 91
语音识别大模型 91
实时语音识别大模型 91
音频文件转写大模型 102
端到端语音语言大模型 108
端到端语音语言大模型（Pro、Lite） 108
端到端语音语言大模型（Flex） 132
端到端语音语言大模型Android SDK 163
端到端语音语言大模型iOS SDK 182
大模型声音复刻 203
简介 203
音色创建与管理 205
语音合成 210
AI翻译配音 217

AI翻译配音快速体验指南 217
AI翻译配音使用说明 219
API接入AI翻译配音 228

API文档 244
语音合成 244
简介 244
鉴权认证 245
音色列表 249
同步语音合成 251
长文本在线合成 255
流式文本在线合成 261
呼叫中心语音-在线合成 268
SSML标记语言使用说明 273
短文本在线合成API 278
语音识别 281
简介 281
鉴权认证 284
短语音识别标准版API 289
短语音识别极速版API 294
实时语音识别-websocket API 298
音频文件转写API 305
音频文件转写极速版API 309
语音质检API 311
呼叫中心语音-音频文件转写（8K） 321
呼叫中心语音-语音识别（8K） 324

SDK文档 327
语音合成 327
短文本在线合成 HTTP SDK 327
语音合成 Android SDK 344
语音合成 iOS SDK 359
语音合成 HarmonyOS SDK 375
语音合成 Linux SDK 389
流式文本在线合成JAVA SDK 397
长文本在线合成JAVA SDK 405
离线音色列表 410
语音识别 412
短语音识别-HTTP-SDK 412
语音识别Android SDK 429
语音识别iOS SDK 459
语音识别 HarmonyOS SDK 472

错误码汇总 478
EasyDL语音自训练平台 488
私有化部署方式 495
部署形式 496
特色优势 496
常见问题汇总 496
相关协议 501
SDK信息保护合规指引 501

产品简介

Hi，您好，欢迎使用百度语音能力引擎（SPEECH）服务。

本文档主要针对API开发者，描述百度语音能力引擎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-语音能力引擎 SPEECH；
- 客服电话：400-920-8999。

注意！

请勿通过任何第三方插件使用百度语音能力引擎服务，使用第三方非法插件会导致您的 AppID、API Key、Secret Key、Access Token 泄露，他人即可盗用您的账户进行任意消费，如因产生的恶意消费，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
端到端语音语言大模型	**百度端到端语音语言大模型基于 Cross-Attention 跨模态架构，以超拟人交互、超精准查询、超低时延和超低成本四大优势，覆盖情感陪伴、智能助手、在线教育、呼叫中心及智能硬件等场景。**
语音识别	**采用国际领先的流式端到端语音语言一体化建模算法，将语音快速准确识别为文字，支持手机应用语音交互、语音内容分析、机器人对话等多个场景。**
短语音识别	将60秒以内的语音精准识别为文字，可适用于手机语音输入、智能语音交互、语音指令、语音搜索等短语音交互场景。
短语音识别极速版	采用最新解码技术，API接口识别速度提升5倍以上，耗时仅音频时长十分之一，提升语音交互体验。
实时语音识别	实时语音识别接口采用websocket协议的连接方式，边上传音频边获取识别结果。适用于长句语音输入、音视频字幕、直播质检、会议记录等场景。
音频文件转写 (16k)	音频文件转写接口可以将大批量的音频文件异步转写为文字。适合音视频字幕生产、批量录音质检、会议内容总结、录音内容分析等场景，一般12小时内返回识别接口。
EasyDL语音识别	可以通过自助训练语言模型的方式有效提升您业务场景下的识别准确率。
语音合成	**基于业内领先的深度学习技术，提供高度拟人、流畅自然的语音合成服务，支持在线、离线多种调用方式，满足泛阅读、订单播报、智能硬件等场景的语音播报需求。**
短文本在线合成	基于HTTP请求的REST API接口，将文本转换为可以播放的音频文件。 支持自动预测/参数控制合成语音情感，支持添加元数据标识，最长可支持1024GBK字节的文本合成语音。
长文本在线合成	长文本在线合成接口可以将10万字以内文本一次性合成，异步返回音频。支持多种优质音库，将超长文本快速转换成稳定流畅、饱满真实的音频。适用于阅读听书、新闻播报等客户。
流式文本在线合成	基于websocket协议，在用户输入文本的同时就能接近同步返回合成音频数据，实现“边合成边播放”。支持多种优质音库与多种参数，适用于语音助手、在线教育、语音播报等场景。
离线语音合成	在无网或弱网环境下，可在手机APP或故事机、机器人等智能硬件设备终端进行语音播报，将文字合成为声音，提供稳定一致、流畅自然的合成体验 。
呼叫中心语音	**呼叫中心服务分为呼叫中心解决语音方案及呼叫中心音频文件转写。可用于智能语音IVR、智能外呼、客服内容质检等场景。**
音频文件转写 (8k)	音频文件转写接口可以将大批量的音频文件异步转写为文字。适合批量录音质检、会议内容总结、录音内容分析等场景，一般12小时内返回识别接口。
呼叫中心语音解决方案	MRCP Server端，集成了呼叫中心8K采样率语音识别(ASR)和呼叫中心专属发音人语音合成(TTS)两种能力。
声音复刻	**基于大模型zero-shot技术，仅需一句话音频即可极速复刻音色。广泛应用于语音助手、情感陪伴、内容配音、数字人代播等场景的语音播报需求。**
HTTP在线合成	基于HTTP请求的REST API接口，将文本转换为可以播放的音频文件。 支持自定义文本复刻，支持上海、四川、河南等多地方言复刻与迁移，支持日语复刻，支持参数控制合成语音情感。
流式文本在线合成	基于websocket协议，在用户输入文本的同时就能接近同步返回合成音频数据，实现“边合成边播放”。 支持上海、四川、河南等多地方言复刻与迁移，支持日语复刻，支持参数控制合成语音情感。

产品更新动态

2026-11-09

端到端语音语言大模型（Flex）公测版本更新

产品分类：大模型语音

功能描述：端到端语音语言大模型服务（Flex）公测版本更新，支持 70+ 预置音色选择，支持Function Call和RAG知识库检索。

适用场景：企业知识问答、语音助手、智能硬件、具身智能、在线教育、情感陪伴

接入文档：端到端语音语言大模型（Flex）API

同步语音合成API上新

产品分类： 语音合成

功能描述： 同步语音合成（HTTP）支持短语音的快速合成，能力覆盖原有短文本合成。接口新增支持卡通IP音色，客服直播场景音色，详情请见音色列表。

适用场景： 语音助手、情感陪伴、直播营销、客服服务

接入文档： 同步语音合成

2026-08-11

语音合成SDK更新版本与发音人资源

产品分类：语音合成

功能描述：语音合成Android、IOS、Harmony对SDK鉴权方式与API接口进行统一优化。

适用场景：基于Android、IOS、HarmonyOS系统的智能手机、穿戴设备、车机系统等终端

接入文档：语音合成 Android SDK、 语音合成 iOS SDK、 语音合成 Harmony OS SDK

2026-07-14

端到端语音语言大模型（Lite）版本正式转入商用

产品分类：大模型语音

功能描述：端到端语音语言大模型服务（Lite）版本正式转入商用，可进入控制台购买。

适用场景：语音助手、情感陪伴、具身智能、在线教育、智能硬件

接入文档：端到端语音语言大模型API

2026-06-11

端到端语音语言大模型（Pro）版本正式转入商用

产品分类：大模型语音

功能描述：端到端语音语言大模型服务（Pro）版本正式转入商用，可进入控制台购买。

适用场景：语音助手、情感陪伴、具身智能、在线教育、智能硬件

接入文档：端到端语音语言大模型API

2026-06-09

声音复刻在线合成API支持情感迁移

产品分类：声音复刻

功能描述：声音复刻支持情感迁移，使用情感参数即可控制合成语音呈现特定情感

适用场景：语音助手、语音社交、内容配音、智能代播

接入文档：大模型声音复刻

2026-04-13

短文本在线合成API支持情感标签控制

产品分类：语音合成

功能描述：超拟人多情感音色，支持通过情感标签字段控制短文本合成（HTTP接口）的语音情感

适用场景：语音助手、情感陪伴、智能客服、在线教育、智能硬件

接入文档：短文本语音在线合成

2026-01-15

声音复刻API支持日语复刻

产品分类：大模型语音

功能描述：声音复刻支持日语复刻，使用日语音频创建音色，即可合成日语音频

适用场景：语音助手、情感陪伴、呼叫中心、在线教育、智能硬件

接入文档：大模型声音复刻

2025-12-25

端到端语音语言大模型转入公测阶段

产品分类：大模型语音

功能描述：端到端语音语言大模型服务API转入公测阶段，支持Lite、Pro两种端到端模型的调用

适用场景：语音助手、情感陪伴、呼叫中心、在线教育、智能硬件

接入文档：端到端语音语言大模型API

2025-12-18

流式文本在线合成服务支持Java SDK

产品分类：语音合成

功能描述：支持通过Java SDK调用流式合成在线合成服务，在用户输入文本的同时就能接近同步的返回合成音频数据，达到“边合成边播放”的效果。

适用场景：智能客服、资讯播报、车载导航、智能硬件实时语音

接入文档：流式文本在线合成JAVA SDK

2025-11-27

语音合成SDK更新版本与发音人资源

产品分类：语音合成

功能描述：语音合成Android、IOS、Harmony进行SDK版本更新与发音人升级。

适用场景：基于Android、IOS、HarmonyOS系统的智能手机、穿戴设备、车机系统等终端

接入文档：语音合成 Android SDK、 语音合成 iOS SDK、 语音合成 Harmony OS SDK

2025-10-28

声音复刻API支持贵州、湖南方言复刻与迁移

产品分类：声音复刻

功能描述：声音复刻支持输入方言合成方言或输入普通话合成方言，范围新增贵州、湖南

适用场景：语音助手、语音社交、内容配音、智能代播

接入文档：大模型声音复刻

2025-10-16

声音复刻API支持流式在线合成

产品分类：声音复刻

功能描述：声音复刻支持双流式在线合成，基于websocket协议，用户输入文本的同时，即可同步返回合成的音频数据，达到“边合成边播放”效果

适用场景：语音助手、语音社交、内容配音、智能代播

接入文档：大模型声音复刻

2025-09-12

声音复刻API支持自定义文本复刻

产品分类：声音复刻

功能描述：声音复刻支持自定义文本复刻，无需调用训练文本接口，上传自由文本音频即可复刻

适用场景：语音助手、语音社交、内容配音、智能代播

接入文档：大模型声音复刻

2025-09-03

语音合成支持Linux SDK

产品分类：语音合成

功能描述：支持将语音合成能力集成至部分Linux系统应用，支持无网或弱网环境下的文字合成语音。

适用场景：基于部分Linux系统的具身智能、车机系统等终端

接入文档：语音合成 Linux SDK

2025-08-27

流式文本在线合成API上新

产品分类：语音合成

功能描述：流式文本在线合成服务，现已全面商业化，支持语音边合成边播放

适用场景：语音助手、情感陪伴、在线教育、资讯播报

接入文档：流式文本在线合成

2025-08-15

声音复刻API支持上海、四川、河南方言复刻与迁移

产品分类：声音复刻

功能描述：声音复刻支持输入方言合成方言或输入普通话合成方言，范围覆盖上海、四川、河南

适用场景：语音助手、语音社交、内容配音、智能代播

接入文档：大模型声音复刻

2025-07-11

端到端语音语言大模型API上新

产品分类：大模型语音

功能描述：端到端语音语言大模型服务，新增支持API连接，可实现云端调用，适用于更多场景

适用场景：语音助手、情感陪伴、呼叫中心、在线教育、智能硬件

接入文档：端到端语音语言大模型API

2025-06-24

实时语音识别支持中文多方言模型

产品分类：语音识别

功能描述：实时语音识别服务，新增支持方言的音频流实时识别，可同时支持中文、粤语、四川话和东北话。

适用场景：在线客服、智能语音助手、电话通话转写等场景，支持多地域用户语音输入，提升用户体验！

升级方式：您可更换输入参数dev_pid，将上述模型dev_pid替换为中文多方言dev_pid=15376即可，注意需要在参数中加入"user":"XXX"（参数任意）

接入文档：实时语音识别-websocket API

2025-06-17

端到端语音语言大模型SDK上新

产品分类：大模型语音

功能描述：端到端语音语言大模型具备超拟人语音合成能力，集成38个垂类助手功能，具备强大的信息检索与指令跟随能力，支持多地域多方言、智能打断与智能降噪。

适用场景：语音助手、情感陪伴、呼叫中心、在线教育、智能硬件

接入文档：端到端语音语言大模型Android SDK、端到端语音语言大模型iOS SDK

2025-06-06

语音合成、识别支持HarmonyOS SDK

产品分类：语音合成、语音识别

功能描述：支持将语音合成与语音识别能力集成至HarmonyOS系统应用，支持无网或弱网环境下的文字合成语音与语音快速准确识别为文字。

适用场景：基于HarmonyOS系统的智能手机、穿戴设备、车机系统等终端，支持语音助手、语音控制等功能

接入文档：语音合成 Harmony OS SDK、语音识别 Harmony OS SDK

2025-06-05

大模型声音复刻服务上新

产品分类：大模型语音

功能描述：基于大模型zero-shot技术，随时随地录制数秒音频，即可极速复刻音色用于语音合成。

适用场景：语音社交、语音助手、在线教育、内容配音

接入文档：大模型声音复刻

2025-05-15

呼叫中心实时语音通话上新

产品分类：语音识别

功能描述：将通话内容准确转写，并根据识别内容提供高度拟人、情感丰富的语音合成与播报功能。支持智能断句、添加标点与工号订单号转换。

适用场景：电话信息通知、客户回访通话、电话营销触达

接入文档：呼叫中心语音-语音识别（8K）

2025-04-30

流式文本在线合成服务上新

产品分类：语音合成

功能描述：支持文本、语音双向流式，在用户输入文本的同时就能接近同步的返回合成音频数据，达到“边合成边播放”的效果。

适用场景：阅读听书、智能客服、车载导航、智能硬件实时语音

接入文档：流式文本在线合成

2024-04-02

【远场语音识别模型下线公告】

尊敬的百度语音客户您好！百度远场语音识别产品历史接口及模型资源整合升级，为您提供更优质的产品效果及使用体验。 **以下模型资源将于4月30日下线**，为避免影响您的服务，请您尽快将服务切换升级：

- dev_pid=1936（1936、19361、19362、19363）

升级方式：远场语音识别相关合作需求，您可提交合作咨询，我们将尽快安排客户经理与您对接。

2024-03-14

语音质检服务上新

产品分类：语音识别

功能描述：将坐席人员的通话内容转写为文字，并进行AI自动化质检，解决人工抽检成本高昂、覆盖不全等问题，大幅提高语音质量监控效率，助力通话数据分析挖掘。

适用场景：金融风控、信贷营销、快递物流等多种行业场景，准确率业界领先！

接入文档：语音质检API

2023-08-03

语音字幕模型上新

产品分类：语音识别

功能描述：AI助力音视频字幕智能生产，基于海量数据和先进算法，打造音视频场景专属模型，识别准确率高达98%，并支持

智能分析标点、断句，准确匹配时间轴，助力字幕生产降本增效。

适用场景：视频编辑工具、视频直播字幕生成、网络课堂字幕生成、在线会议字幕生成

接入文档：音频文件转写极速版API、音频文件转写API、实时语音识别API

2023-03-16

【语音识别模型下线公告】

尊敬的百度语音客户您好！百度语音识别产品历史接口及模型资源整合升级，为您提供更优质的产品效果及使用体验。

以下模型资源将于3月31日下线，为避免影响您的服务，请您尽快将服务切换升级：

- dev_pid=1538
- dev_pid=1700
- dev_pid=1721

升级方式：您可更换输入参数dev_pid，将上述模型dev_pid替换为短语音识别标准版dev_pid=1537（支持API及SDK调用），或短语音识别极速版dev_pid=80001（支持API调用，效果更优）即可。

以下语音自训练平台资源将于3月31日下线，为避免影响您的服务，请您尽快将服务切换升级：

- dev_pid=8001
- dev_pid=8002
- dev_pid=8003

升级方式：点击进入新版语音自训练平台，创建您的专属语音识别模型，具体操作请见：使用文档

若您有任何问题，可提交工单与我们联系，我们将结合您的服务使用情况提供合适的解决方案支持，感谢您的理解和配合!

2023-01-19

长文本在线合成支持字、句时间戳

产品分类：语音合成

功能描述：长文本在线合成服务，新增支持合成结果返回字、句粒度时间戳。

适用场景：小说听书、新闻播报等场景，帮助您实现边听边读、回听定位等效果，提升用户体验！

接入文档：长文本在线合成API

2022-11-24

【服务升级公告】

尊敬的百度语音客户您好！

百度语音服务将于22年11月29日至12月30日进行服务升级，短文本在线合成、实时语音识别服务升级支持多地域并发合并统计，并发配额资源可在多个地域间共享，更好保障跨地域实时服务稳定性。

若您在升级期间出现任何问题，可提交工单与我们联系，我们将结合您的服务使用情况提供合适的解决方案支持，感谢您的理解和配合!

快速开发指南

资源领取&应用创建

本文主要介绍如何快速开通语音技术服务，并完成接口调用。

一、注册及实名认证

使用百度智能云语音识别服务前，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，详情可参考注册说明和登录说明。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语音技术服务。

二、自动发放免费测试资源

实名认证后登录语音技术控制台。测试资源会在5-10分钟内自动免费发放到账户ID里，免费测试资源长期有效，详情可点击此处查看免费额度。

自动发放的免费资源可以在资源列表里查看。

三、创建应用

您需要创建应用才可正式调用语音技术能力，应用是您调用服务的基本操作单元，您可以基于应用创建成功后获取的API Key及Secret Key，进行接口调用操作或SDK激活使用。您可按照下图所示的操作流程，完成创建操作。

方式一：在概览页上快速创建应用（仅限API调用）

进入语音技术控制台，点击显示新手引导后，使用指引模块的“快速接入服务”按钮，或点击“去创建”均可创建应用。





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"。

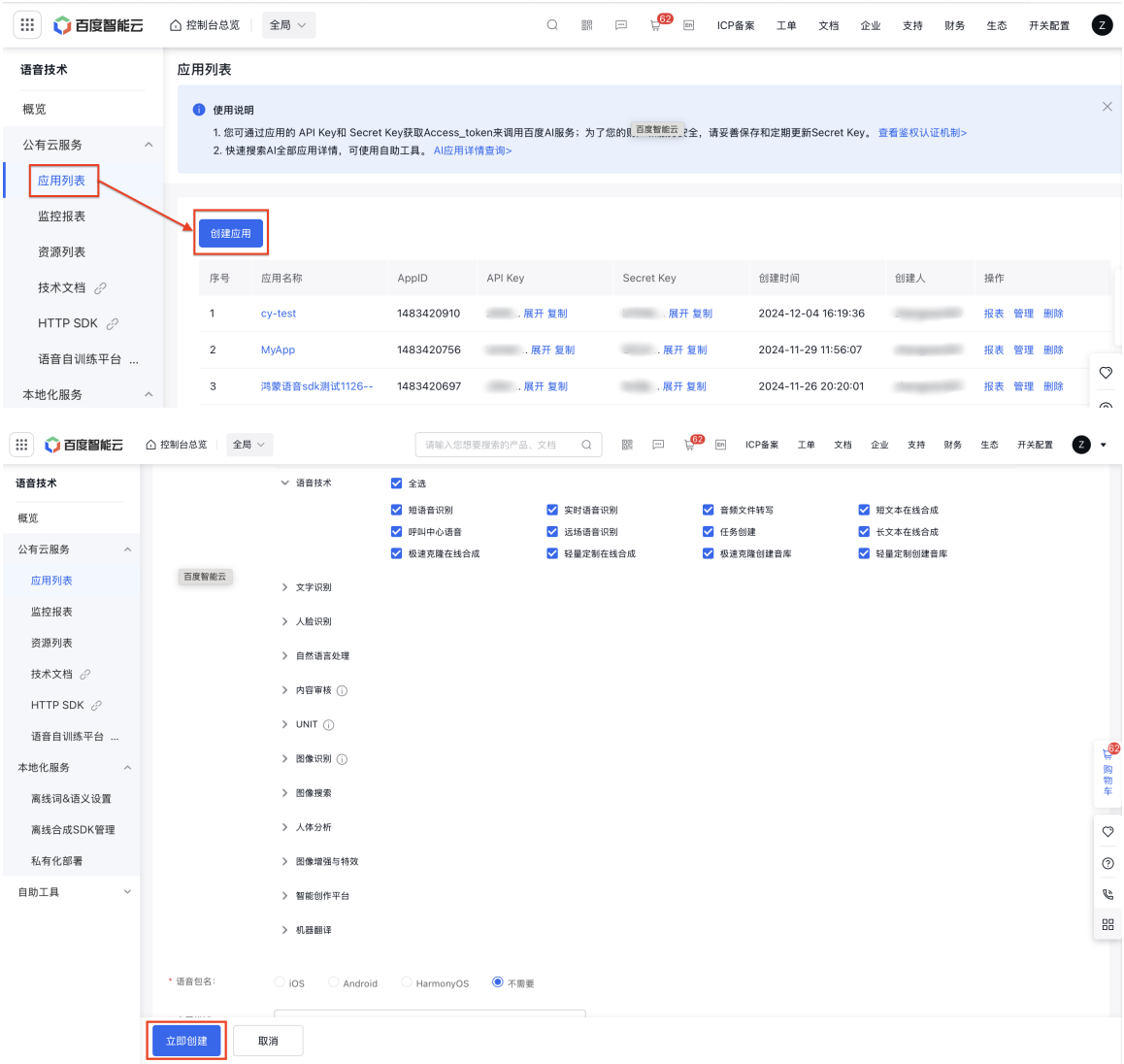


创建成功！API Key和Secret Key是您调用该应用内接口的凭证，如果泄露会导致资源被盗刷，请妥善保管，避免外泄。

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（支持API和SDK调用）

进入语音技术控制台，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"。

应用名称： 用于标识您所创建的应用的名称，支持中英文、数字、下划线及中横线，此名称一经创建完毕，不可修改。

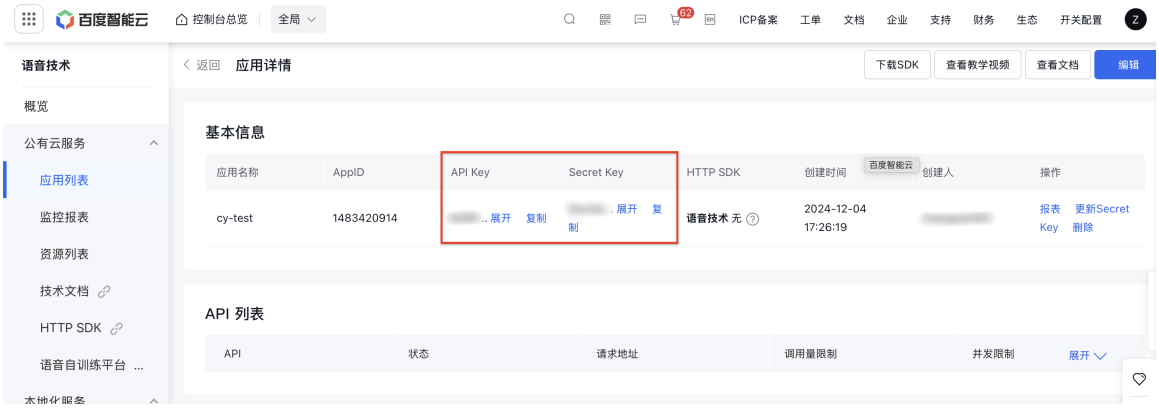
接口选择： 每个应用可以勾选业务所需的所有AI服务的接口权限（仅可勾选具备免费试用权限的接口能力），语音技术下全部接口已默认勾选，创建应用完毕，此应用即具备了所勾选服务的调用权限。

语音包名：如果您需要使用语音技术SDK服务（iOS/Android），需要绑定包名信息，以便生成授权License。

应用归属：可选择个人使用或公司使用服务，若为公司使用，可与专属商务经理沟通，获取专业的售前支持。

应用描述： 对此应用的业务场景进行描述。

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 在线调用语音技术服务 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语音技术服务。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。

- **通过可视化工具（如 Postman）调用语音技术服务 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 调用语音技术服务 API。

- **通过编写代码调用语音技术服务 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。具体请参见如何用代码调用。

- **通过软件开发工具包（HTTP-SDK）调用语音技术服务**

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语音合成、语音识别。

更多参考

- 语音识别API文档
- 语音合成API文档
- 如何获取 API Key 和 Secret Key

如何用可视化工具调用

如何使用 Postman 调用语音技术服务 API

本文提供了通过可视化工具 Postman 调用短文本语音合成 API 的样例，帮助您零编码快速体验并熟悉语音技术服务。

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>>

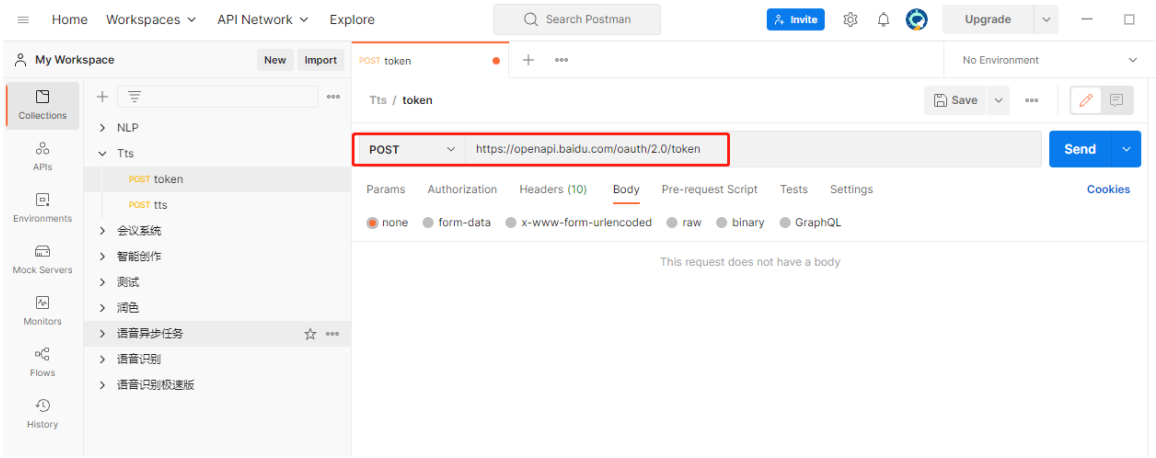
Windows 下载地址，点击前往>>

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

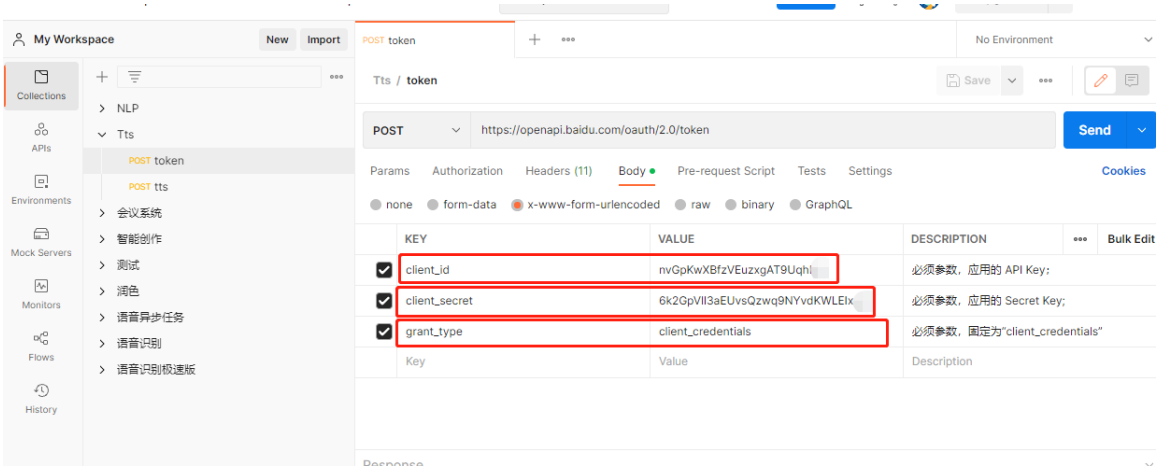


点击 Body，选择“x-www-form-urlencoded”，在 key 和 value 中分别输入以下3个请求参数。

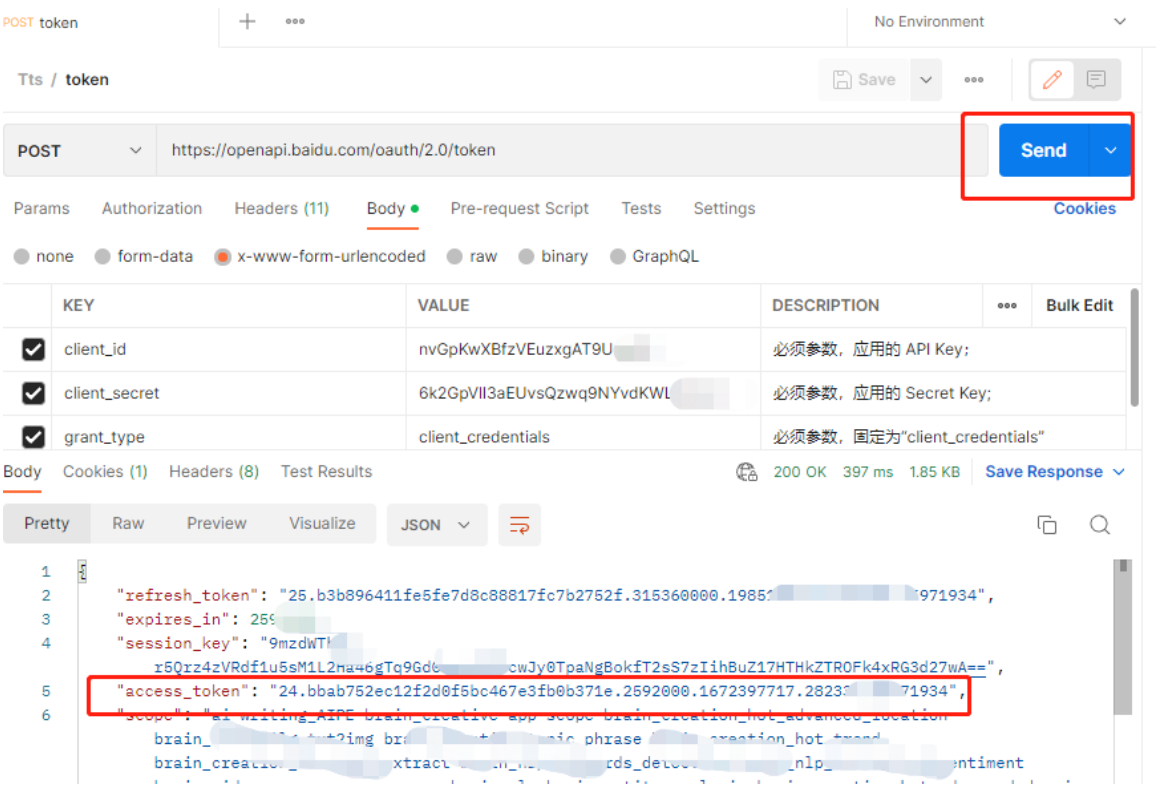
grant_type： 必须参数，固定为 **client_credentials**

client_id： 必须参数，应用的 API Key

client_secret： 必须参数，应用的 Secret Key



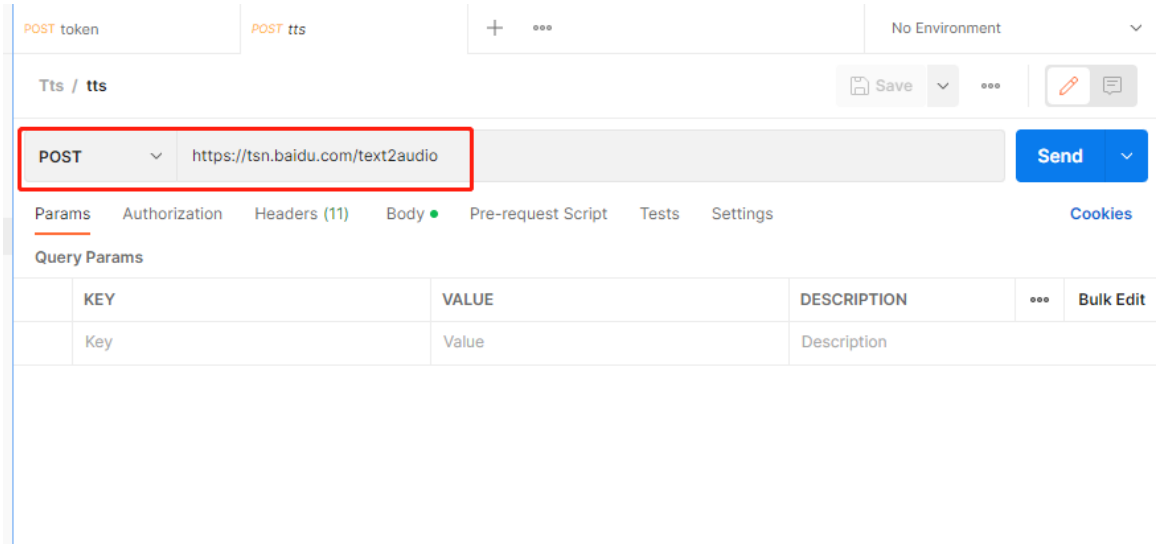
点击右上角蓝色“send”，即可在下方返回值区域中获取 access_token。



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（以短文本语音合成为例）：https://tsn.baidu.com/text2audio



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

tex： 必须参数，合成文本

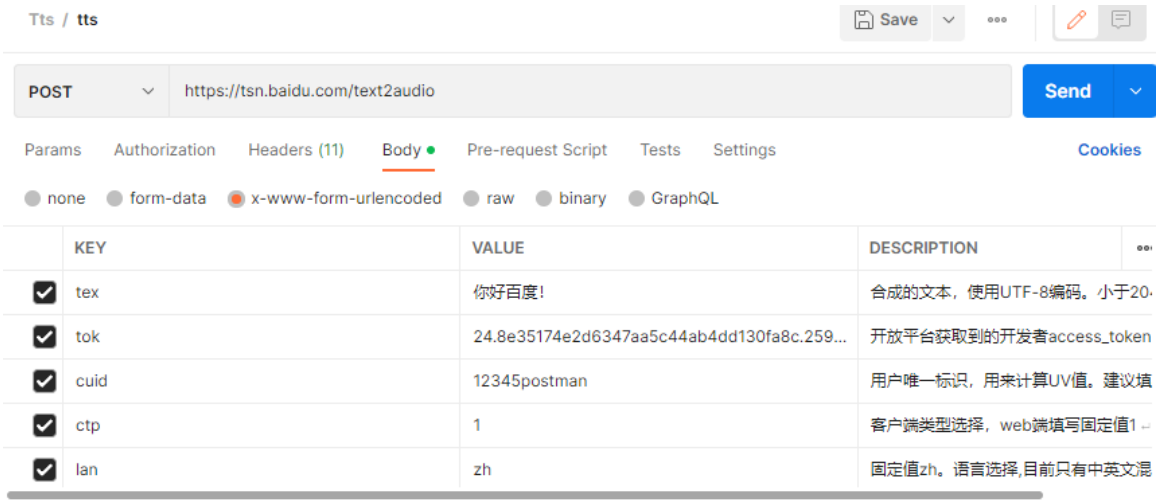
tok： 必须参数，获取到的access_token参数

cuid： 必须参数，用户唯一标识

ctp： 必须参数，客户端类型选择，web端填写固定值 1

lan： 必须参数，固定值 zh

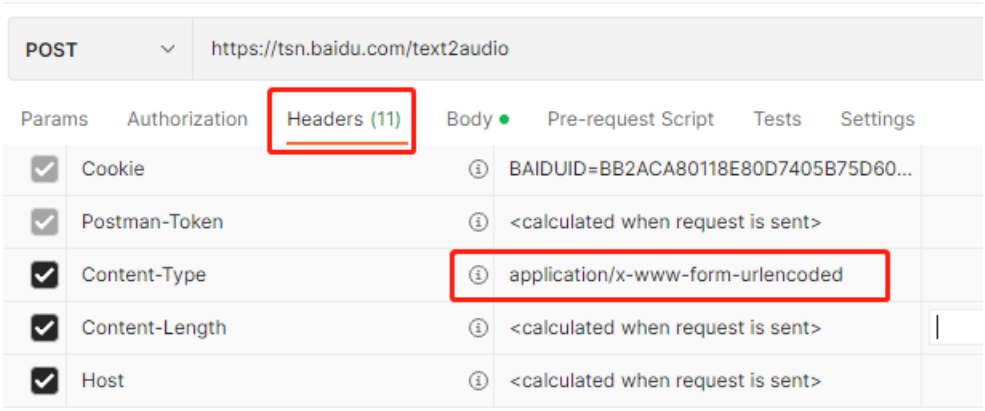
（更多参数请到短文本语音合成页面查看）



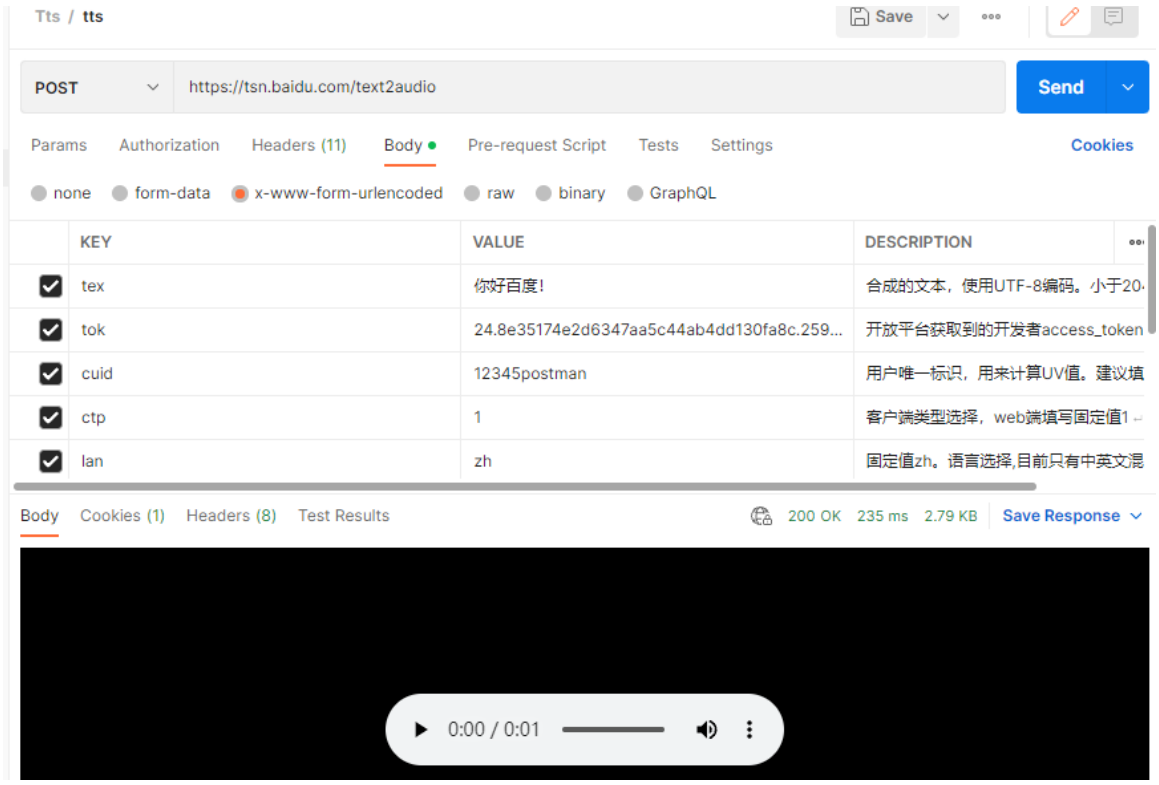
（3） 修改请求头，点击 Headers，在 key 和 value 中分别输入1个请求参数。

key 栏输入：**Content-Type**

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



（4）点击右上角蓝色“send”，即可在下方返回值区域中获取音频。



如何用代码调用

编写一个示例程序

有**准备工作**的***API KEY***以及 ***Secret KEY***，的数据，并且领取了免费测试额度，我们就可以写一个示例代码调用百度AI开放平台的语音合成能力

准备开发环境

大姚选择用python来快速搭建一个原型，那我们就安装以下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，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。

编写代码

新建一个 **main.py**

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

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

import sys
import json

#### 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

#### 替换你的 API_KEY
API_KEY = 'nu9r2plGFi3s1ugayDPSM6Mk'

#### 替换你的 SECRET_KEY
SECRET_KEY = 'G62YGnq84eKTqu0mBgvdpmC6gNBzHdai'

#### 大姚的订单信息内容文本
TEXT = "三分钟前，由北京市顺义区二经路与二纬路交汇处北侧，北京首都国际机场T3航站楼 去往 东城区北三环东路36号喜来登大酒店(北京金隅店)"



TTS_URL = 'http://tsn.baidu.com/text2audio'

"""  TOKEN start """

TOKEN_URL = 'http://openapi.baidu.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('token http response http code : ' + str(err.code))
```

```
        result_str = err.read()
    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 'audio_tts_post' in result['scope'].split(' '):
            print ('please ensure has check the tts ability')
            exit()
        return result['access_token']
    else:
        print ('please overwrite the correct API_KEY and SECRET_KEY')
        exit()


""" TOKEN end """

if __name__ == '__main__':

    token = fetch_token()

    tex = quote_plus(TEXT)  # 此处TEXT需要两次urlencode

    params = {'tok': token, 'tex': tex, 'cuid': "quickstart",
              'lan': 'zh', 'ctp': 1}  # lan ctp 固定参数

    data = urlencode(params)

    req = Request(TTS_URL, data.encode('utf-8'))
    has_error = False
    try:
        f = urlopen(req)
        result_str = f.read()

        headers = dict((name.lower(), value) for name, value in f.headers.items())

        has_error = ('content-type' not in headers.keys() or headers['content-type'].find('audio/') < 0)
    except  URLError as err:
        print('http response http code : ' + str(err.code))
        result_str = err.read()
        has_error = True

    save_file = "error.txt" if has_error else u'大姚的订单信息.mp3'

    with open(save_file, 'wb') as of:
        of.write(result_str)

    if has_error:
        if (IS_PY3):
            result_str = str(result_str, 'utf-8')
        print("tts api  error:" + result_str)

    print("file saved as : " + save_file)
```

运行代码

在命令行中运行`python main.py`

结果

代码运行成功后，在main.py的同级目录中会产生一个 **大姚的订单信息.mp3** 文件，大姚打开一听觉得这就是他想要的效果，让我们一起来听听看:

更多示例代码

更多示例代码可到控制台内API在线调试页面获取：API在线调试

如何集成SDK

语音识别SDK

Android SDK快速集成指南

只需四步，即可完成语音识别SDK的应用集成，让您的应用获得稳定一致的识别体验。

Step1：成为百度AI开放平台的开发者

要调用百度AI开放平台的语音识别能力先要成为百度AI开放平台的开发者，首先让我们花5分钟来注册百度AI开放平台的开发者，并新建一个百度语音识别应用。

1. 创建帐户

先点击此处注册百度账户，快速建立一个百度账号，请参考下图：

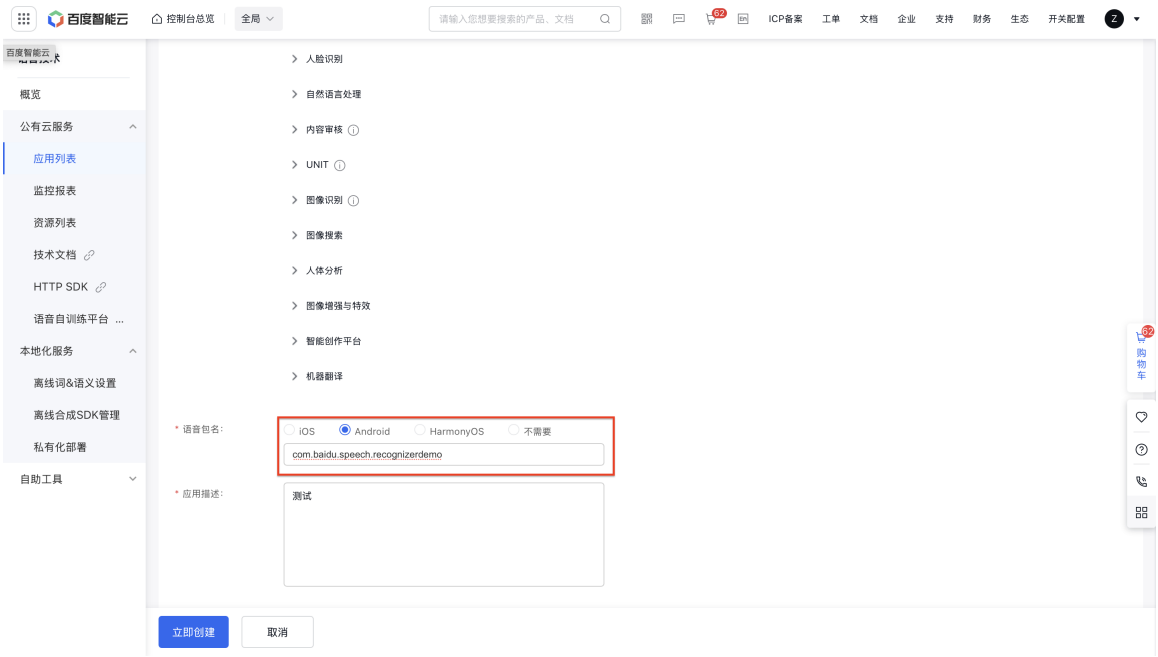


2. 创建应用

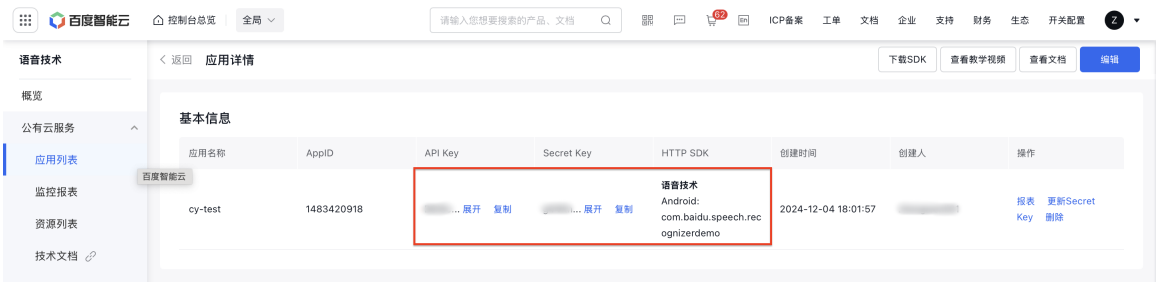
创建账号之后，登录百度AI开放平台，并且点击此处创建一个应用，如下图：



创建应用时请务必输入应用名称、语音包名（输入示例demo包名：**com.baidu.speech.recognizerdemo**）等信息，创建完成后就可以看到创建的应用信息了：



创建完成的应用信息如下图所示：



Step2：申请语音识别额度

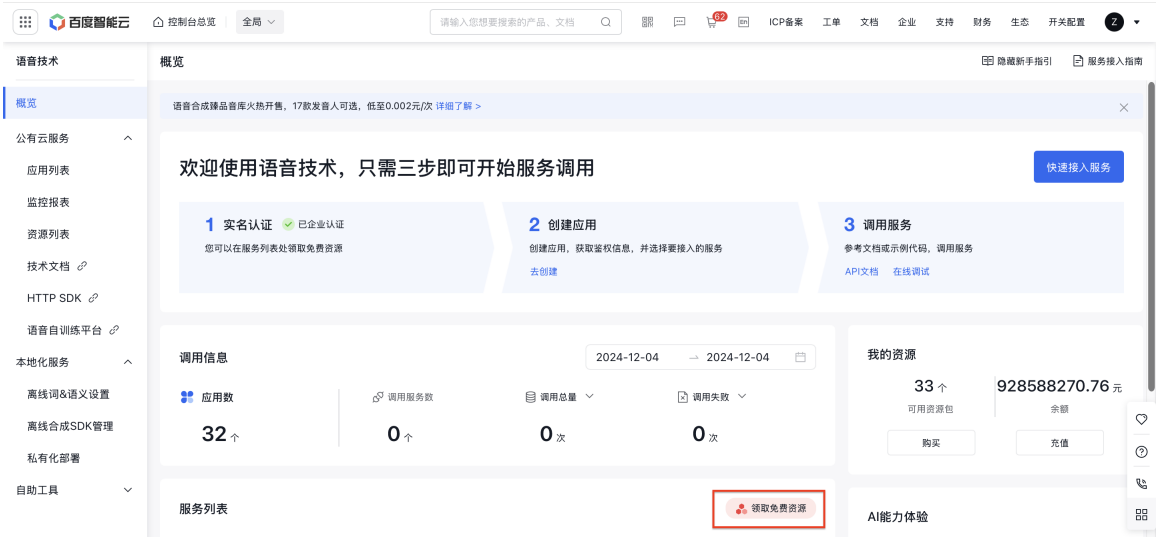
1. 实名认证

语音识别SDK需要完成实名认证后才能使用，第一步需要按照提示完成个人认证或企业认证。完成实名认证的用户可以领取免费额度。如下图所示进行实名认证：



2. 领取免费额度

完成实名认证后，需领取语音识别的免费额度。如下图所示：



Step3：下载语音识别SDK，填写授权信息

1. 获取鉴权信息 准备好创建应用后获取到的3个鉴权信息，AppID、API Key、Secret Key，需要您登陆控制台查看应用详情获取。

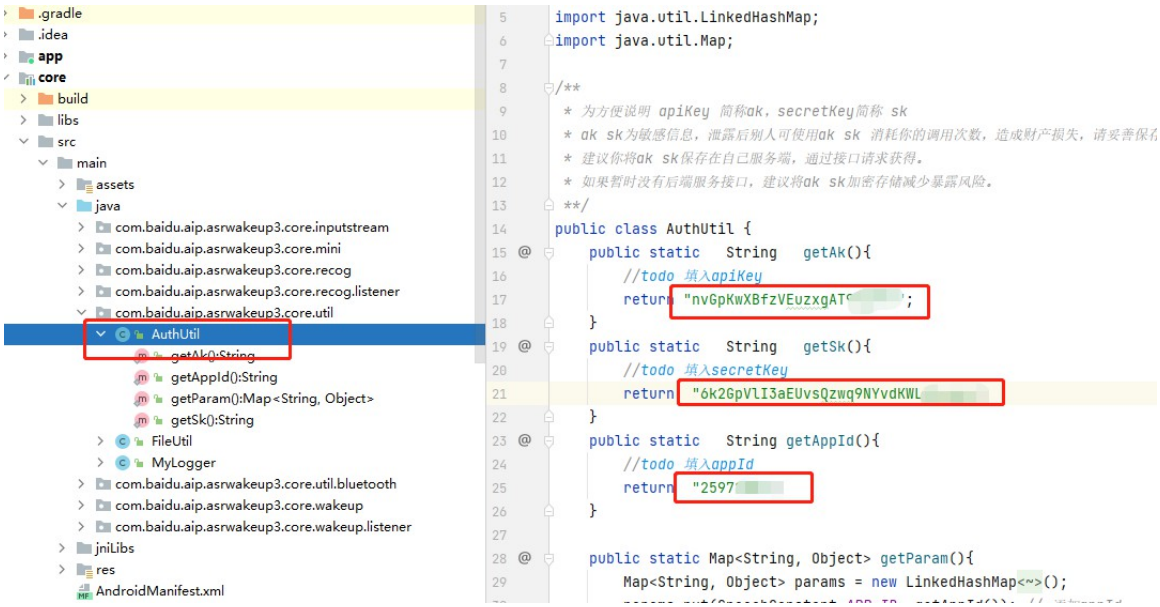
2. 下载语音识别SDK 在SDK下载页面下载 **语音识别Android SDK** 文件，链接：https://ai.baidu.com/sdk

3. 不修改SDK，直接运行 解压sdk文件，不做任何修改，直接安装运行程序可以看到以下界面：



4. 填写鉴权信息

填入 **AppID、API Key、Secret Key** 3个鉴权信息测试，请按照以下步骤全部修改：



Step4: 测试语音识别功能

1. 按照上述文档修改完成后，安装app打开后进入在线识别，点击开始录音可进行在线语音识别：



简单的 语音识别Android SDK 测试完成了，其他各子功能可以按照详细的技术文档进行集成。

iOS SDK快速集成指南

只需四步，即可完成语音识别SDK的应用集成，让您的应用获得稳定一致的识别体验。

Step1：成为百度AI开放平台的开发者

要调用百度AI开放平台的语音识别能力先要成为百度AI开放平台的开发者，首先让我们花5分钟来注册百度AI开放平台的开发者，并新建一个百度语音识别应用。

1. 创建帐户

先点击此处注册百度账户，快速建立一个百度账号，请参考下图：

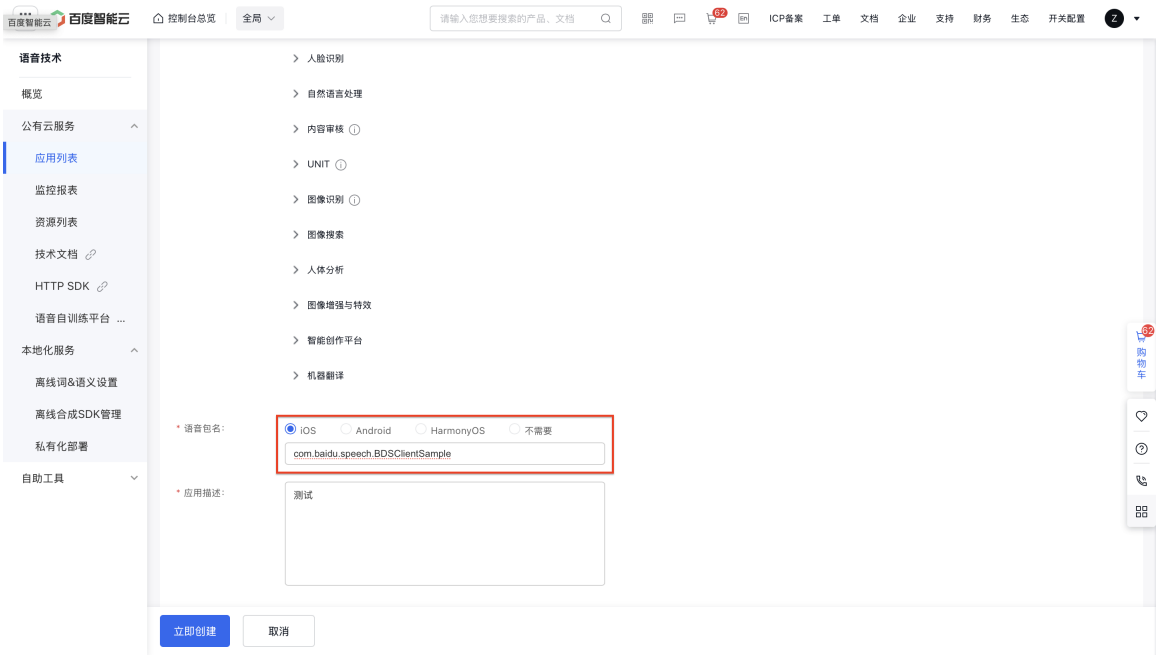


2. 创建应用

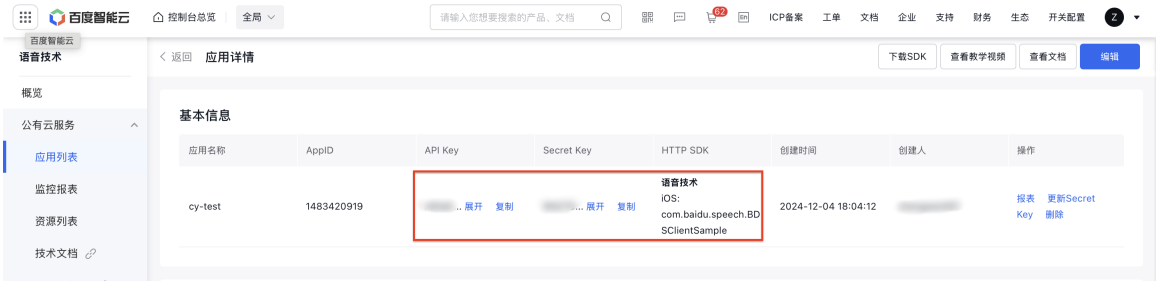
创建账号之后，登录百度AI开放平台，并且点击此处创建一个应用，如下图：



创建应用时请务必输入应用名称、语音包名（输入示例demo包名：**com.baidu.speech.BDSClientSample**）等信息，创建完成后就可以看到创建的应用信息了：



创建完成的应用信息如下图所示：



Step2：申请语音识别额度

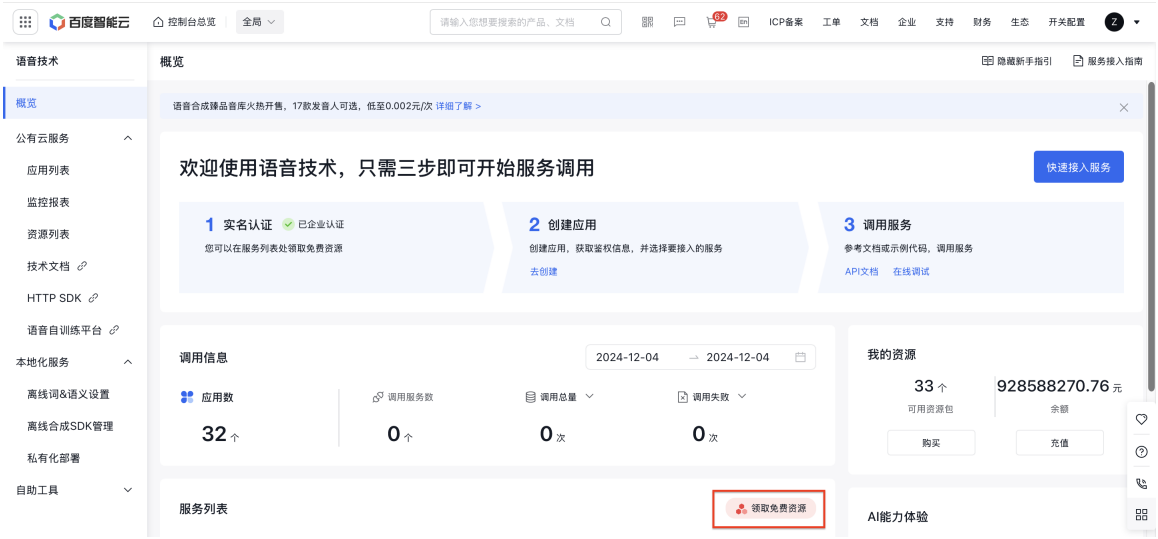
1. 实名认证

语音识别SDK需要完成实名认证后才能使用，第一步需要按照提示完成个人认证或企业认证。完成实名认证的用户可以领取免费额度。如下图所示进行实名认证：



2. 领取免费额度

完成实名认证后，需领取语音识别的免费额度。如下图所示：



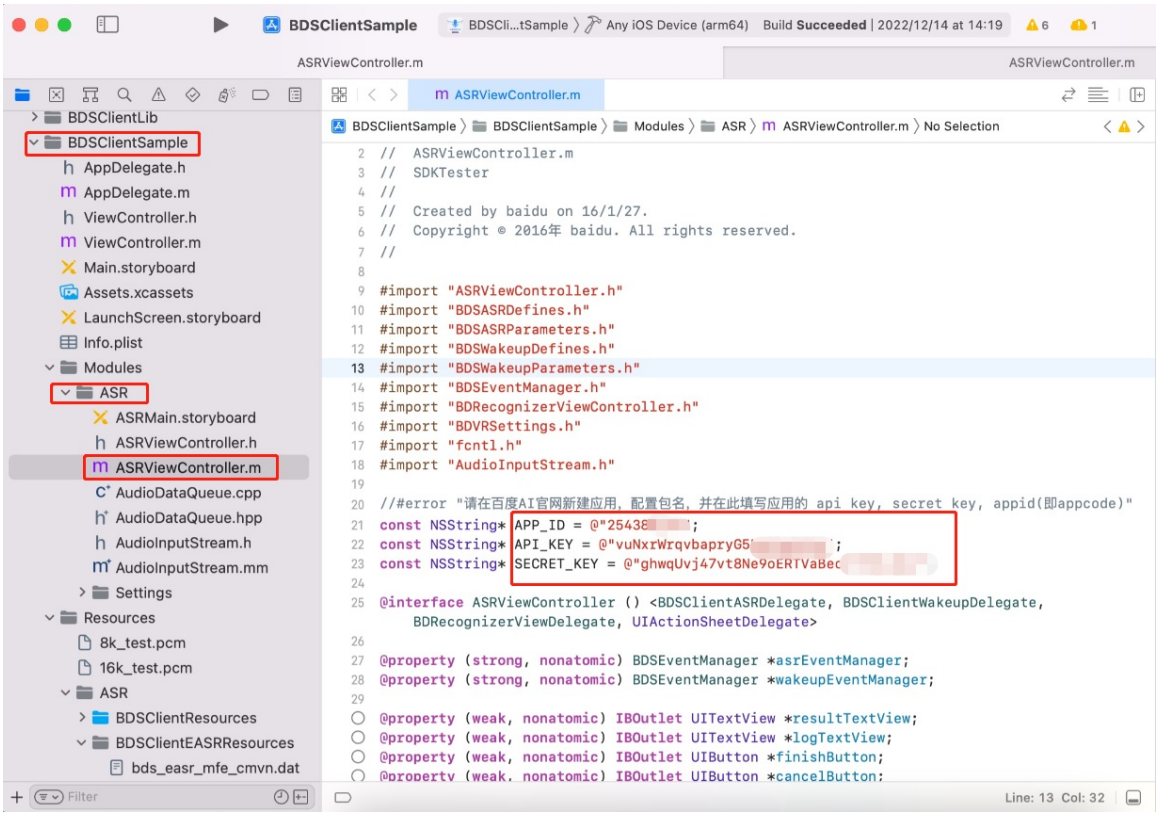
Step3：下载语音识别SDK，填写授权信息

1. 获取鉴权信息 准备好创建应用后获取到的3个鉴权信息，AppID、API Key、Secret Key，需要您登陆控制台查看应用详情获取

2. 下载语音识别SDK 在SDK下载页面下载 **语音识别IOS SDK** 文件，链接：https://ai.baidu.com/sdk

3. 填写鉴权信息

填入 **AppID、API Key、Secret Key** 3个鉴权信息测试，请按照以下步骤全部修改：



Step4: 测试语音识别功能

1. 按照上述文档修改完成后，安装app打开后可进行在线语音识别：



简单的 语音识别IOS SDK 测试完成了，其他各子功能可以按照详细的技术文档进行集成。

HarmonyOS SDK快速集成指南

只需四步，即可完成语音识别SDK的应用集成，让您的应用获得稳定一致的识别体验。

Step1：成为百度AI开放平台的开发者

要调用百度AI开放平台的语音识别能力先要成为百度AI开放平台的开发者，首先让我们花5分钟来注册百度AI开放平台的开发者，并新建一个百度语音识别应用。

1. 创建帐户

先点击此处注册百度账户，快速建立一个百度账号，请参考下图：



2. 创建应用

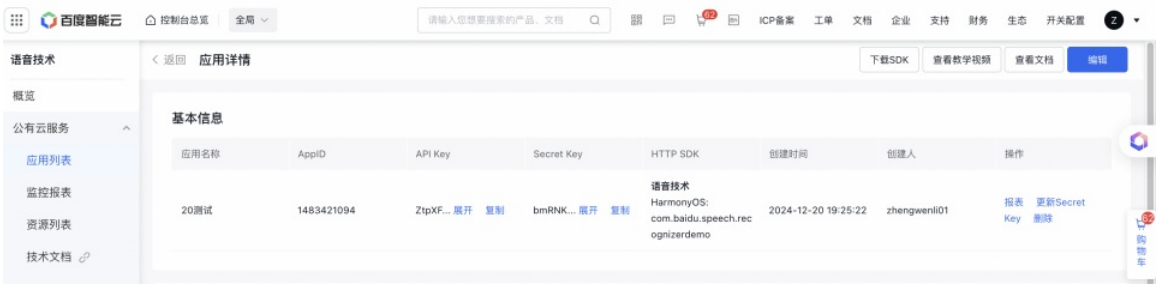
创建账号之后，登录百度AI开放平台，并且点击此处创建一个应用，如下图：



创建应用时请务必输入应用名称、语音包名（输入示例demo包名：**com.baidu.speech.recognizerdemo**）等信息，创建完成后就可以看到创建的应用信息了：



创建完成的应用信息如下图所示：



Step2：申请语音识别额度

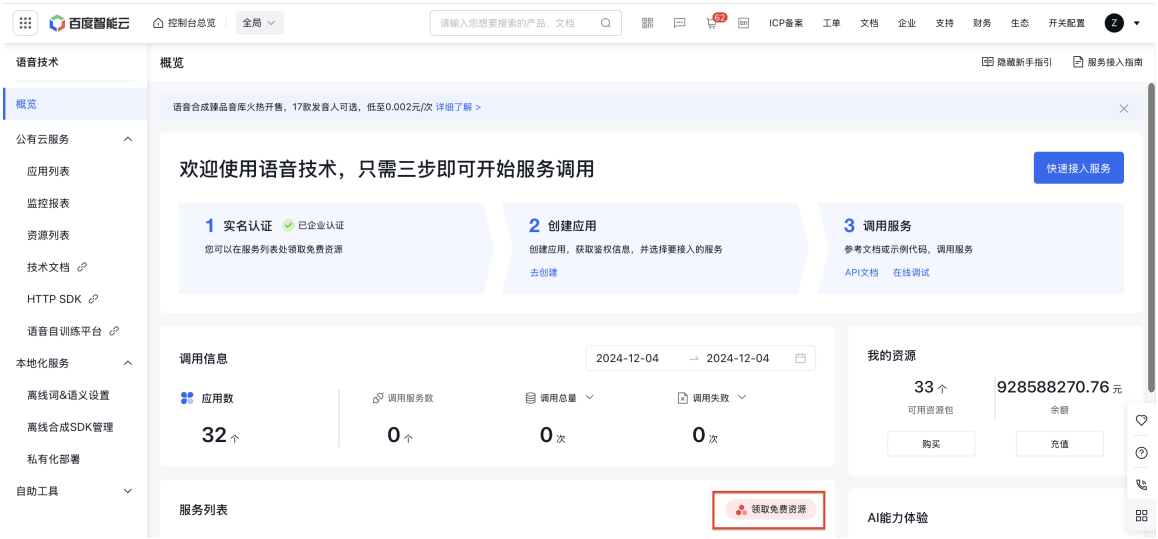
1. 实名认证

语音识别SDK需要完成实名认证后才能使用，第一步需要按照提示完成个人认证或企业认证。完成实名认证的用户可以领取免费额度。如下图所示进行实名认证：



2. 领取免费额度

完成实名认证后，需领取语音识别的免费额度。如下图所示：



Step3：下载语音识别SDK，填写授权信息

1. 获取鉴权信息

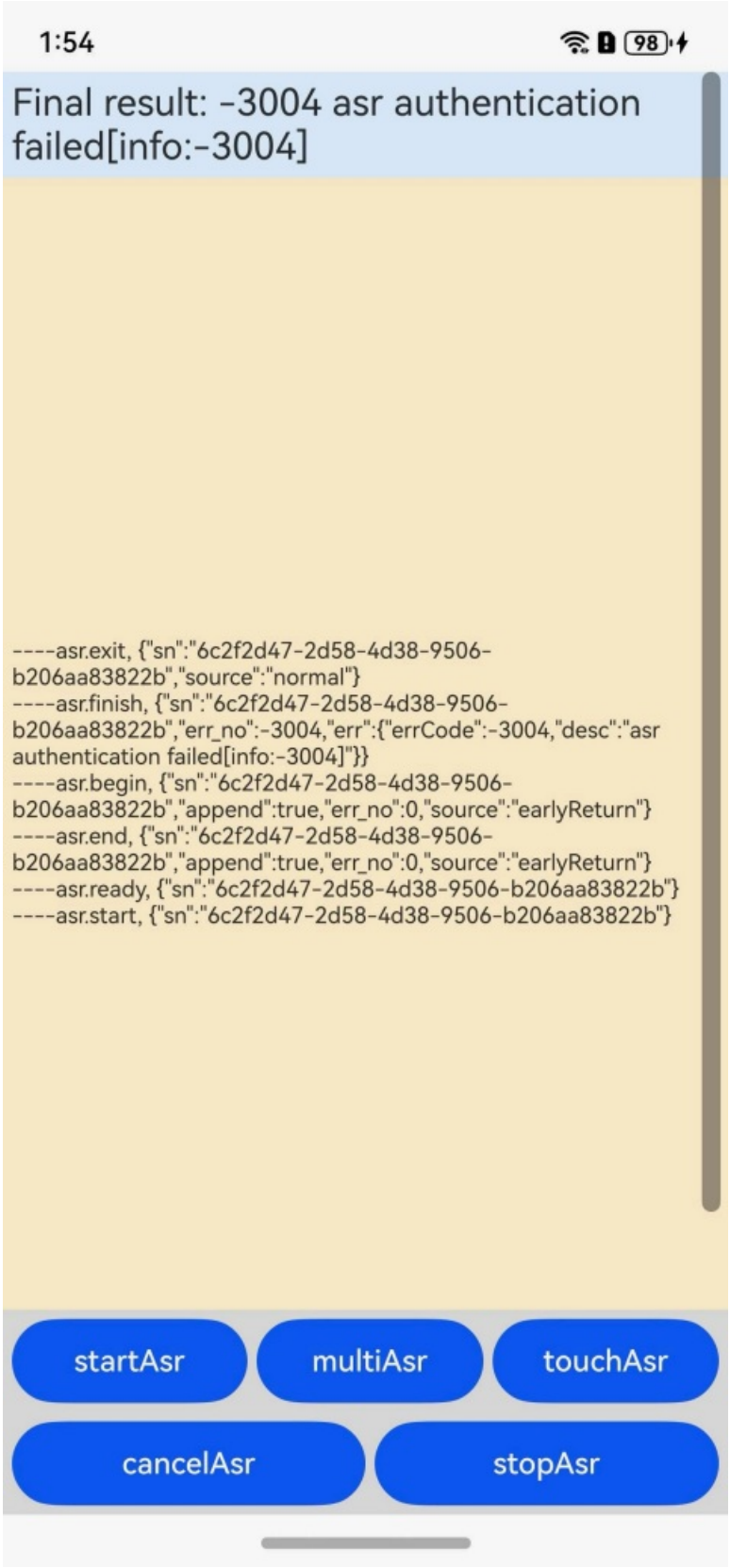
准备好创建应用后获取到的3个鉴权信息，AppID、API Key、Secret Key，需要您登陆控制台查看应用详情获取。

2. 下载语音识别SDK

在SDK下载页面下载 语音识别HarmonyOS SDK 文件，链接：https://ai.baidu.com/sdk

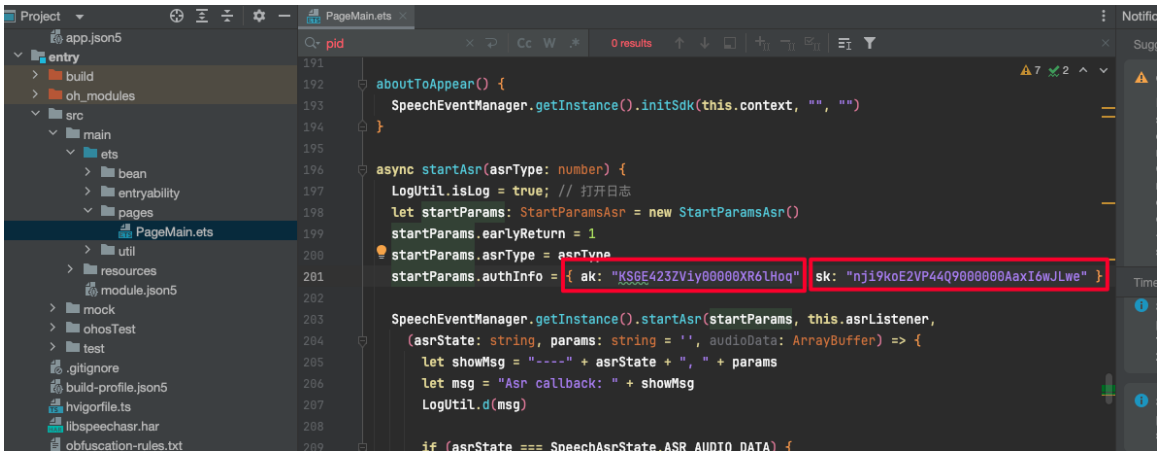
3. 不修改SDK，直接运行

解压sdk文件，不做任何修改，直接安装运行程序可以看到以下界面：



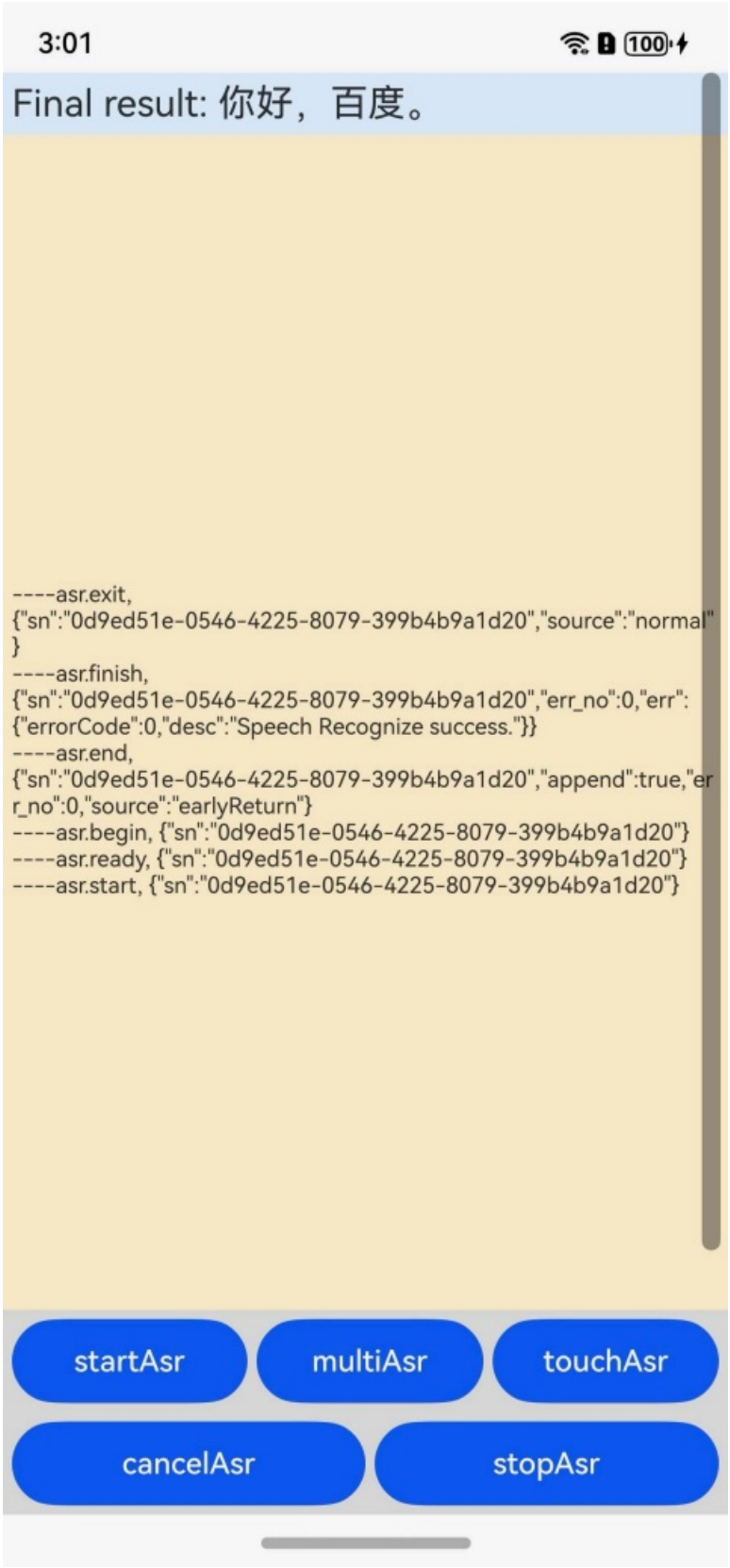
4. 填写鉴权信息

填入 API Key、Secret Key 2个鉴权信息测试，请按照以下步骤全部修改：



Step4: 测试语音识别功能

1. 按照上述文档修改完成后，安装app打开后进入在线识别，点击开始录音可进行在线语音识别：



简单的 语音识别HarmonyOS SDK 测试完成了，其他各子功能可以按照详细的技术文档进行集成。

语音合成SDK

文档简介

本文档提供语音合成SDK快速集成说明，帮助新用户在1小时内完成离线语音合成SDK的应用集成，让您的应用获得稳定一致的合成体验。如需查看SDK使用说明，请前往语音合成SDK开发文档。如需了解音库版本与具体发音人，请前往离线音色列表。

资源下载

- 语音合成在线SDK发布地址：https://ai.baidu.com/sdk#speech

- 语音合成离线SDK发布地址：
  - Android：https://console.bce.baidu.com/ai-engine/old/#/ai/face/offline/index~modelId=3&sdkName=Android
  - iOS：https://console.bce.baidu.com/ai-engine/old/#/ai/speech/offline/index~sdkName=iOS
  - Harmony：https://console.bce.baidu.com/ai-engine/old/#/ai/speech/offline/index~sdkName=%E9%B8%BF%E8%92%99
- 精品音库下载入口：https://console.bce.baidu.com/ai-engine/old/#/ai/speech/offline/index

*离线SDK使用需要下载音库，SDK内部已包含基础音库，若购买精品发音人请下载精品音库；往期离线音库及更多音库信息详见离线音色列表。

前置条件

在下载并集成SDK之前，请确保：

- 您已注册百度智能云账号并完成实名认证，具体步骤请参考账号注册及实名认证。
- 您已创建对应平台的SDK应用，创建应用时请务必输入应用名称、语音包名等信息，具体步骤请参考应用创建。
- 您已前往离线合成SDK管理，根据提示获取测试序列号(SN)，保存包名、AppID、APIKey、SecretKey等信息。离线合成SDK需要5个授权信息全部正确，才能联网拉取License，正常使用全部合成功能。

Android SDK快速集成指南

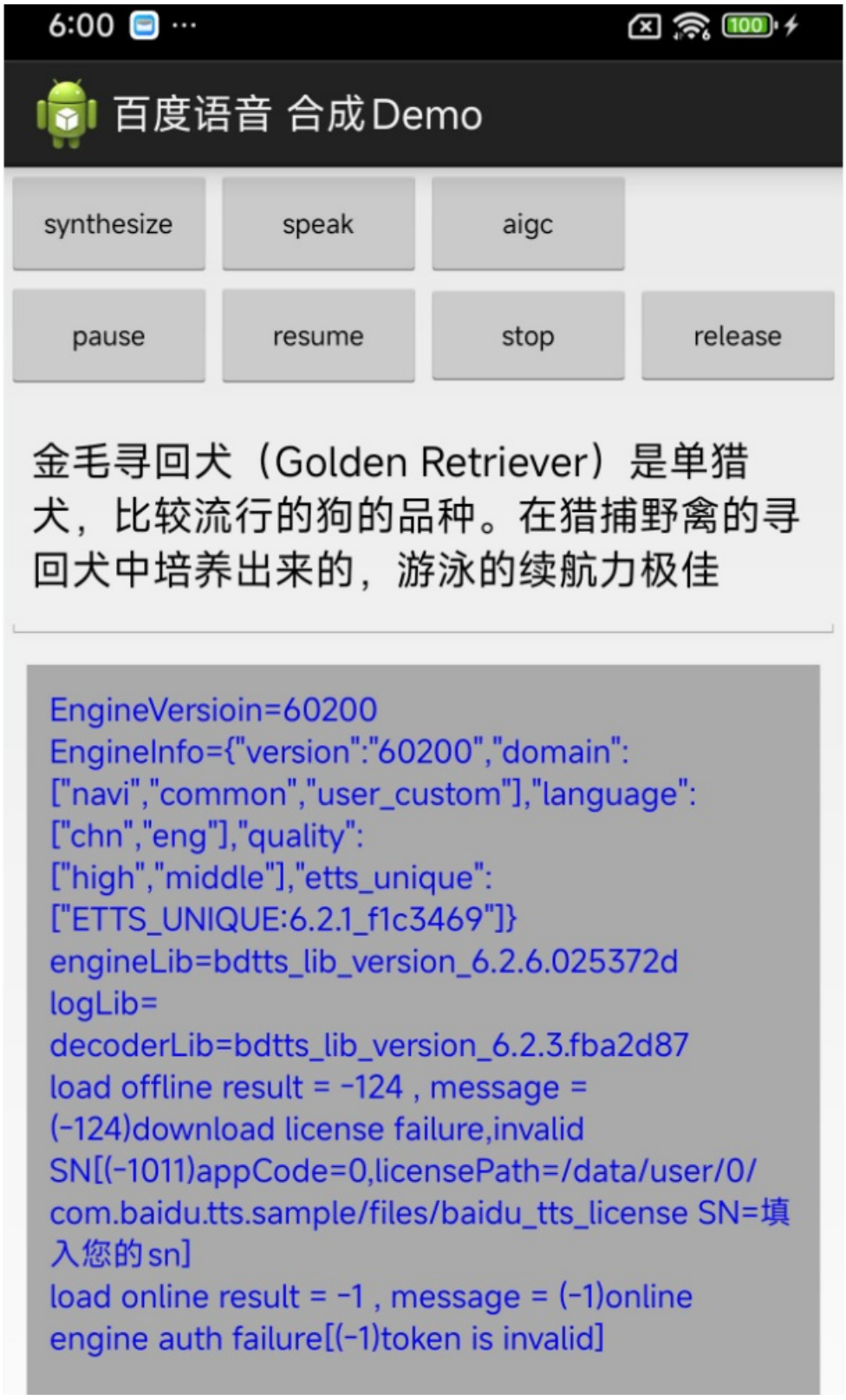
集成环境要求

运行语音合成Android SDK需要Android 5.1+ 环境，支持armeabi-v7a，arm64-v8a，x86，x86_64架构，详见语音合成Android SDK文档。

集成说明

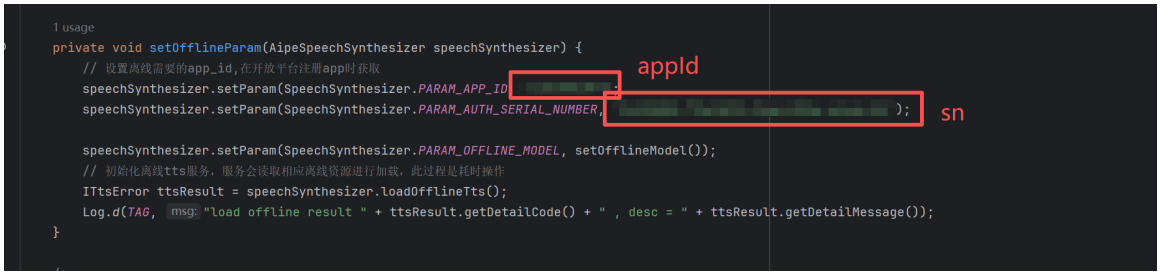
步骤一：填写授权信息，联网获取License

下载并解压Android SDK文件，不做任何修改，直接安装运行程序可以看到以下界面：



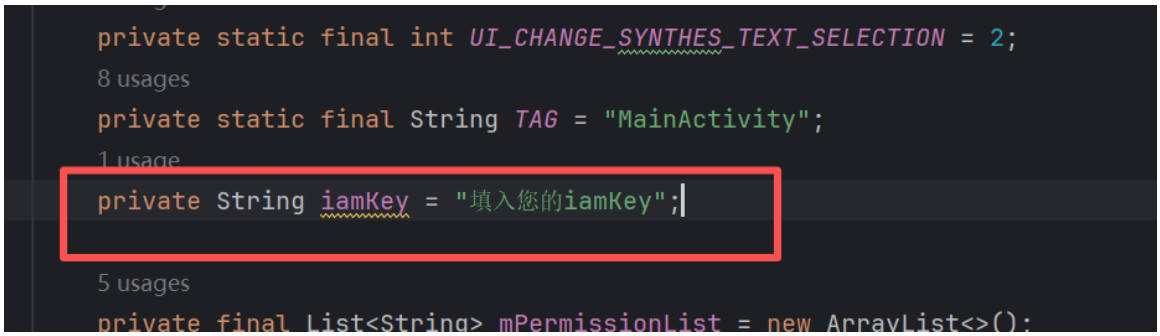
填写离线鉴权信息

填入序列号SN、包名、AppID 3个鉴权信息，SDK会自动下载鉴权文件，获取离线合成能力。 在MainActivity中setOfflineParam方法中填写appId 和 SN:

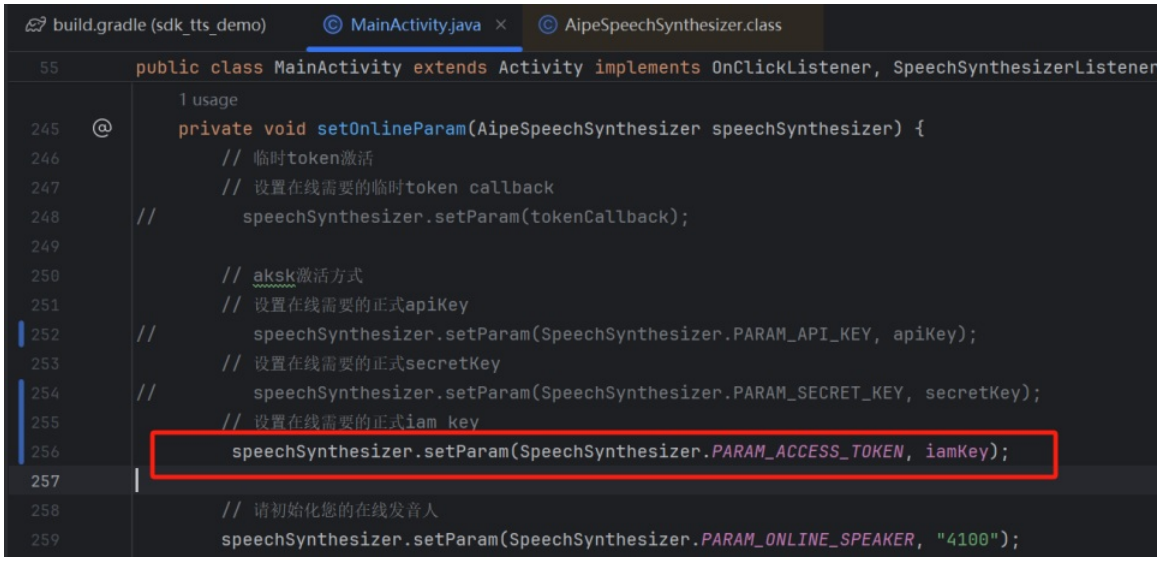


填写在线鉴权信息

若希望使用在线合成能力，请将申请的iamKey放到项目中。iamKey可根据文档提示获取。



将iamKey传入接口（该方法需要使用AipeSpeechSynthesizer类不能使用SpeechSynthesizer进行设置）。

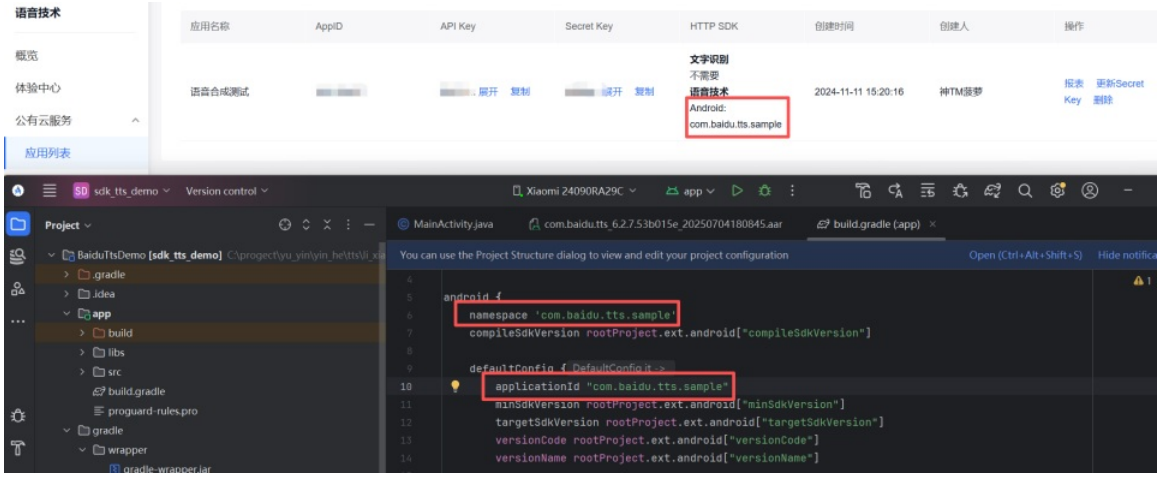


修改build.gradle文件并确认包名

TTS SDK版本为tts_1.0.3-9c93a23-20260727.aar。 将app/libs/tts_1.0.3-9c93a23-20260727.aar复制到您的项目的同名目录中。确认在build.gradle文件中引入。



1. 根目录下build.gradle确认下gradle的版本。
2. app/build.gradle 确认下 applicationId 包名是否与官网申请应用时相一致（离线功能需要）。 demo的包名是"com.baidu.tts.sample"。



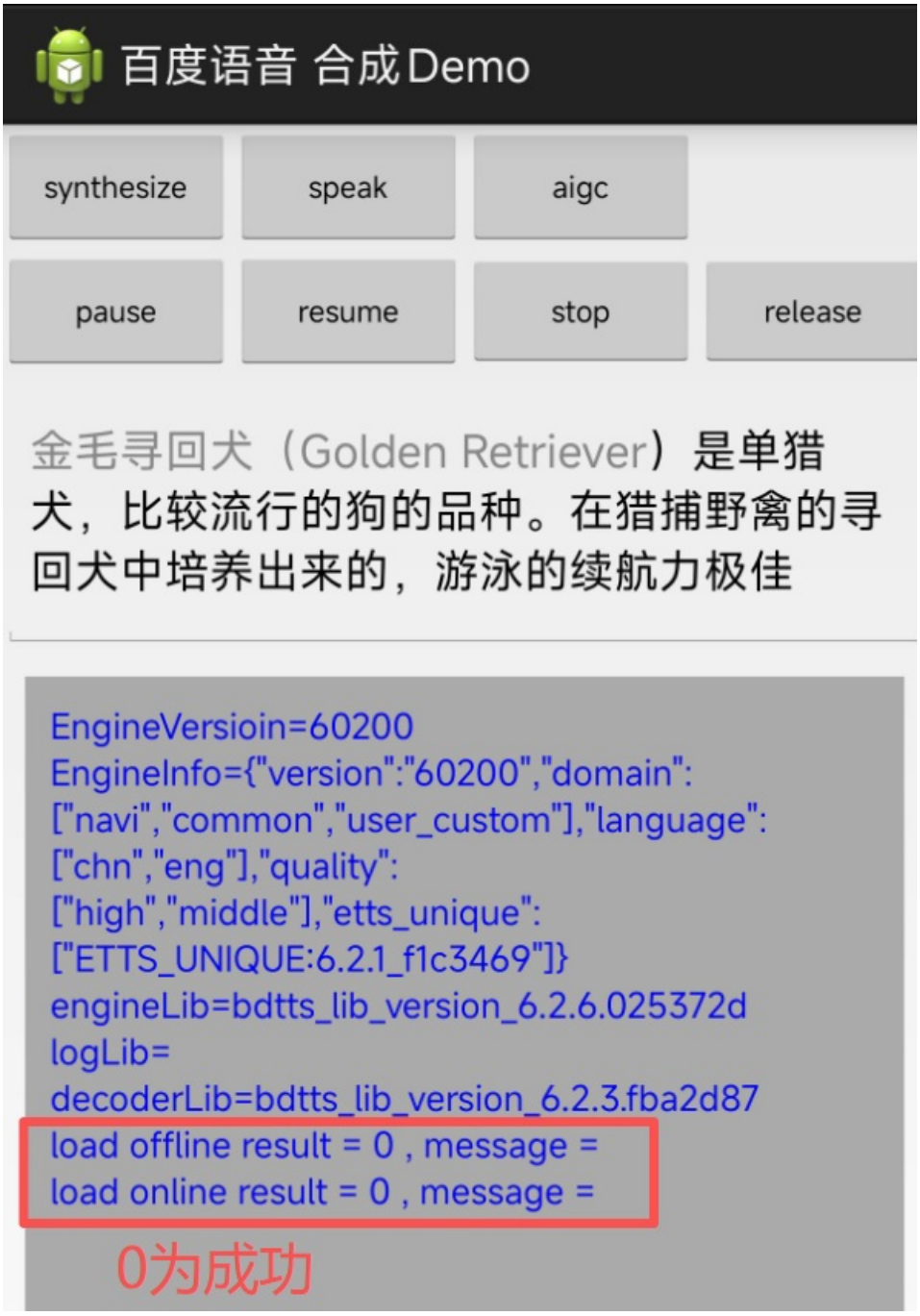
步骤二：断网测试离线合成功能

按照上述文档修改完成后，安装App，首次使用需要联网，打开如下图：

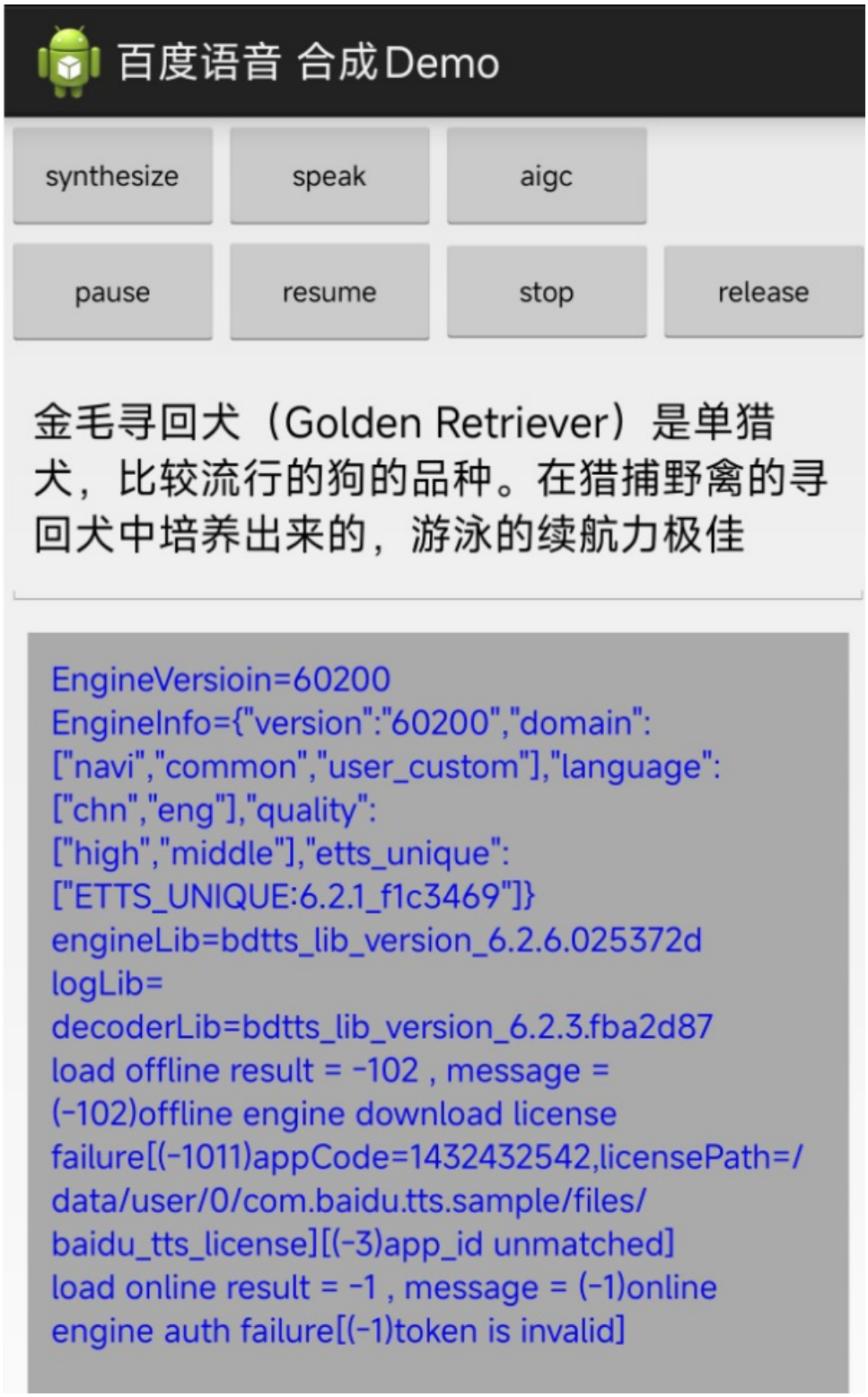


点击输入文本合成如下图：

出现该提示后，则可以正常断网离线使用。



如果鉴权信息填写错误会出现下图的报错-102情况：



iOS SDK快速集成指南

集成环境要求

运行语音合成iOS SDK需要iOS 12.0+、Xcode 14+环境，支持x86_64、arm64架构(离线合暂成不支持x86_64架构)，适用于IPhone、iPad机型。具体信息详见语音合成iOS SDK文档。

集成说明

步骤一：填写授权信息，联网获取License

下载最新版本的iOS SDK

下载地址：https://console.bce.baidu.com/ai/#/ai/speech/offline/index。



在BDSClientSample\Modules\TTS\TTSViewController.mm 中填写API Key, Secret Key、AppID和SN（序列号）。AppID、API Key、Secret Key、包名信息需要您登陆控制台查看应用详情获取，SN即为Step2中您获取到的测试序列号。

在线鉴权： token或iamkey鉴权方式，可参考技术文档具体使用说明。

```
- (void)initOnlineSDK {
    // 创建语音合成器实例
    self.synthesizer = [[BDSSpeechSynthesizer alloc] initWithDelegate:self];
    //  申请获取token鉴权
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAccessToken
        value:@"用户token配置"];
    // 申请获取IAMKEY鉴权
    //  [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAccessToken
        value:@"用户iamkey配置"];
    // 设置在线发音人
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramOnlineSpeaker
        value:@"4100"];
//      加载在线引擎
    BDSSpeechResult *result = [self.synthesizer loadOnlineEngine];
    if (result.code != 0) {
        NSLog(@"加载在线引擎失败: %ld", (long)result.code);
    }
}
```

离线鉴权：appid + sn方式，可参考[技术文档]

```
- (void)initOfflineSDK {
    // 初始化日志，仅 DEBUG 调试打开此设置
    [[BDSSpeechLogger sharedInstance] setLogLevel:BDSSpeechLogLevelInfo path:nil callback:nil];

    // 创建语音合成器实例
    self.synthesizer = [[BDSSpeechSynthesizer alloc] initWithDelegate:self];
    // 设置离线参数资源文件
    NSString *textData = [[NSBundle mainBundle]
        pathForResource:@"bd_etts_common_text_txt_all_mand_eng_middle_big_v6.0.0_20240731" ofType:@"dat"];
    NSString *speechData = [[NSBundle mainBundle]
        pathForResource:@"bd_etts_common_speech_duxiaoyao_mand_eng_high_am-tac-csubgan16k_v4.9
        .0_20251110_20251210203318" ofType:@"dat"];

    // 正确的离线模型配置
    NSDictionary *offlineModel = @{
            BDSSpeechSynthesizerParams.ttsTextModelFile: textData,
            BDSSpeechSynthesizerParams.ttsSpeechModelFile: speechData
        };
    NSData *offlineModeData = [NSJSONSerialization dataWithJSONObject:offlineModel options:0 error:nil];
    NSString *offlineModelString = [[NSString alloc] initWithData:offlineModeData
        encoding:NSUTF8StringEncoding];
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramOfflineModel value:offlineModelString];

    // 设置离线授权文件
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAppId value:@"appid"];
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAuthSerialNumber value:@"sn"];
```


如果要测试纯离线模式将mode改为BDSSpeechModeOffline 纯在线修改mode为BDSSpeechModeOnline（如下图），其他模式具体[参见文档]。

纯在线模式：

```
#pragma mark - Button Actions

- (void)synthesizeAction:(UIButton *)sender {
    NSString *text = self.inputTextView.text;
    if (text.length == 0) {
        [self appendLog:@"请输入要合成的文本"];
        return;
    }
    BDSSpeechEntity *entity = [[BDSSpeechEntity alloc] initWithText:text
                                                               mode:BDSSpeechModeOnline
                                                     utteranceId:[[NSUUID UUID] UUIDString]
                                                             params:nil];
```

纯离线模式：

```
#pragma mark - Button Actions

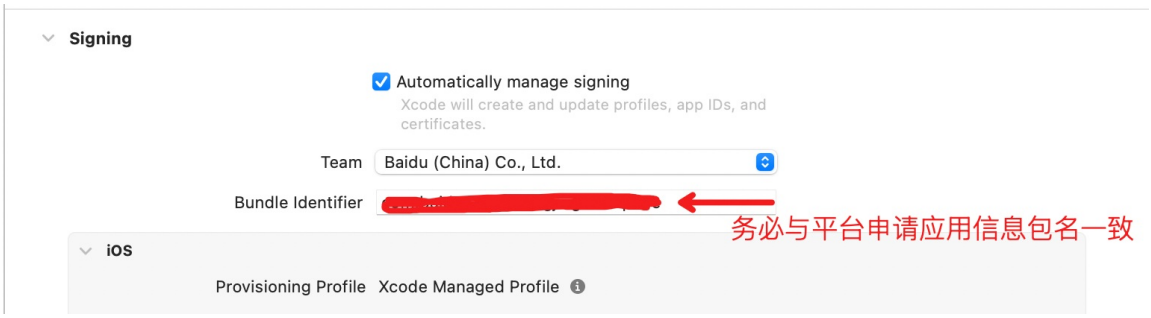
- (void)synthesizeAction:(UIButton *)sender {
    NSString *text = self.inputTextView.text;
    if (text.length == 0) {
        [self appendLog:@"请输入要合成的文本"];
        return;
    }
    BDSSpeechEntity *entity = [[BDSSpeechEntity alloc] initWithText:text
                                                               mode:BDSSpeechModeOffline
                                                     utteranceId:[[NSUUID UUID] UUIDString]
                                                             params:nil];
```

注：synthesizeAction（合成）speakAction（播报）需同时修改mode生效。

其他模式如下 纯在线、纯离线、混合（在线优先）三种模式。

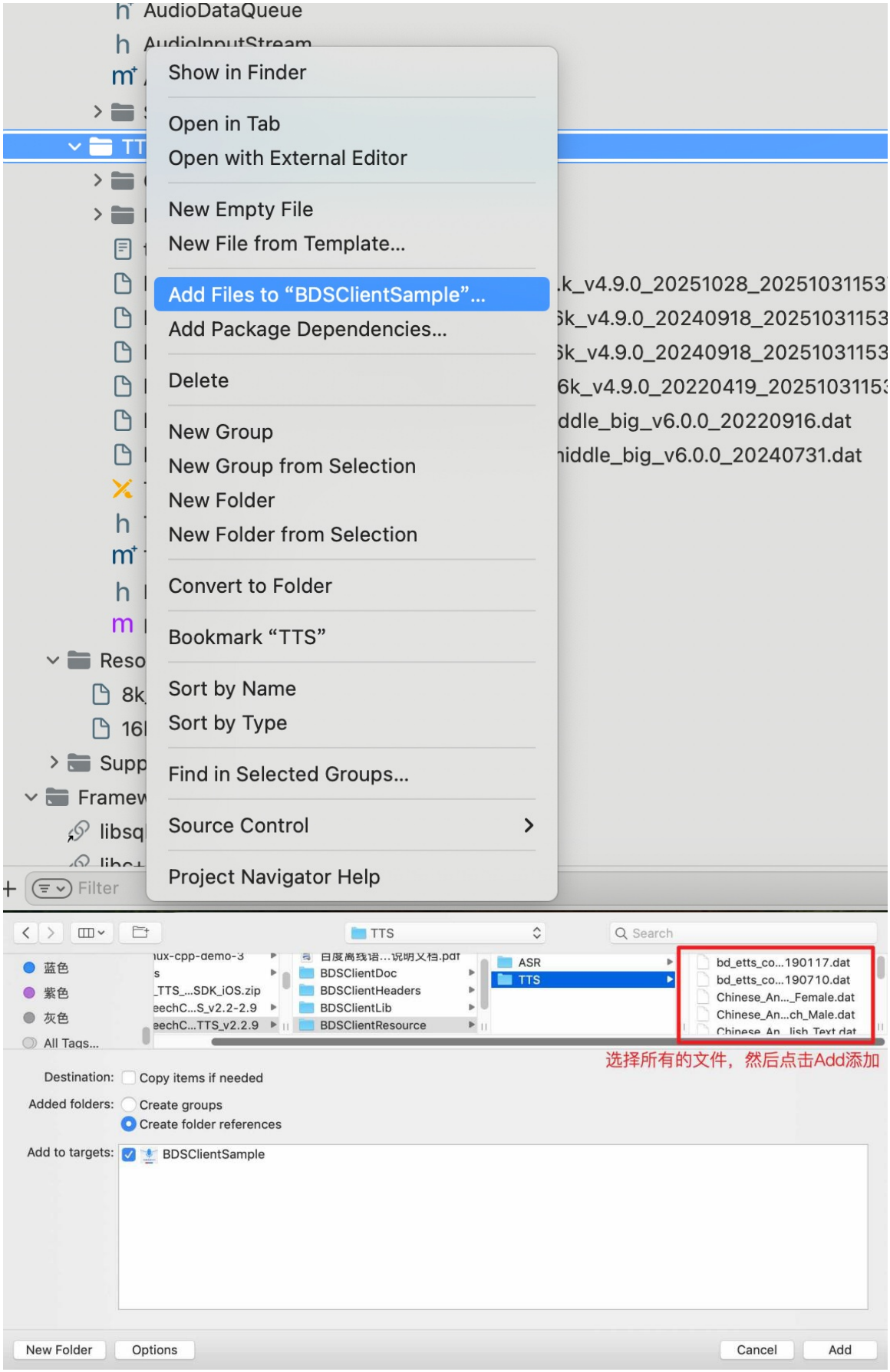
```
/**
 * 语音合成模式
 */
typedef NS_ENUM(NSUInteger, BDSSpeechMode) {
    /// 纯在线合成
    BDSSpeechModeOnline,
    /// 纯离线合成
    BDSSpeechModeOffline,
    /// 混合模式，在线优先
    BDSSpeechModeMix
};
```

填写包名，务必与官网应用信息中显示的应用包名保持一致：

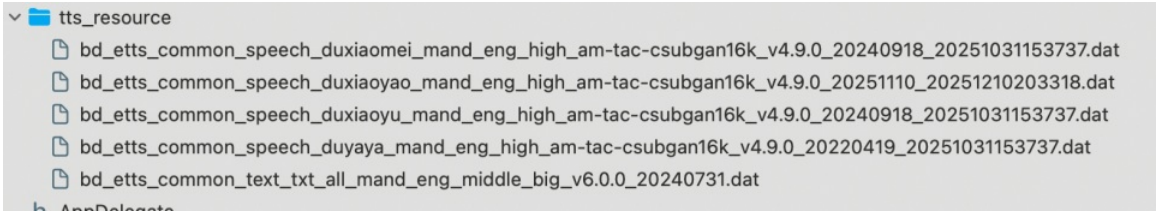


添加发音人文件

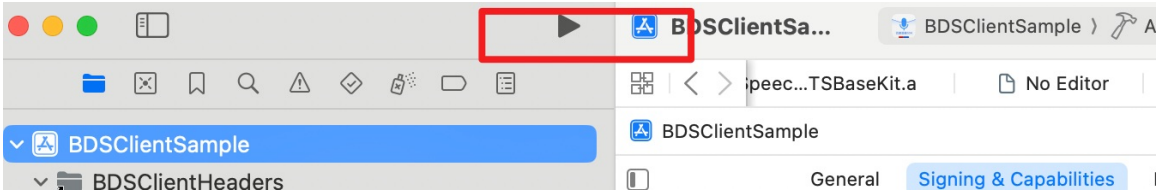
右击 BDSClientSample\Resources\TTS 添加dat发音人文件：



添加完成后如下图：

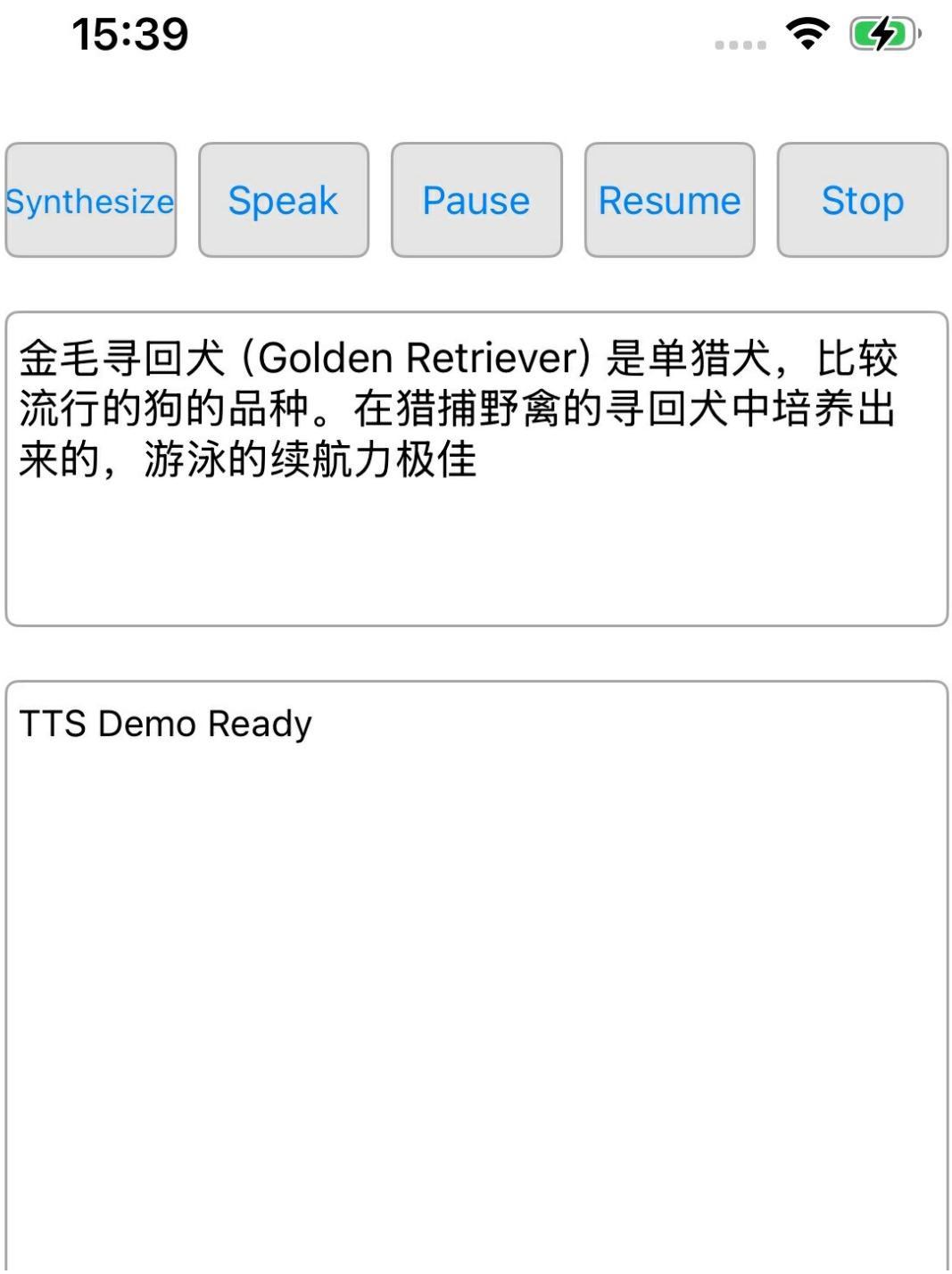


点击运行Demo拉取License



步骤二: 断网测试离线合成功能

首次测试需要联网合成下载linence；打开Demo 如图，点击TTS Demo：



输入合成文本点击Synthesize合成后点击speak进行播报：



```
[15:50:23] 合成结束: EEE00206-F850-46FB-
A629-60D516A8718A
[15:50:47] 合成开始:
A26122FD-93AC-4BEA-909B-8F28F61CE178
[15:50:47] 合成结束:
A26122FD-93AC-4BEA-909B-8F28F61CE178
[15:50:48] 播放开始:
A26122FD-93AC-4BEA-909B-8F28F61CE178
[15:50:50] 播放完成:
A26122FD-93AC-4BEA-909B-8F28F61CE178
```

HarmonyOS SDK快速集成指南

集成环境要求

运行语音合成iOS SDK需要HarmonyOS 6.0.0（APILevel 20）+环境，支持arm64-v8a架构。具体信息详见语音合成Harmony SDK文档。

集成说明

步骤一：填写授权信息，联网获取License

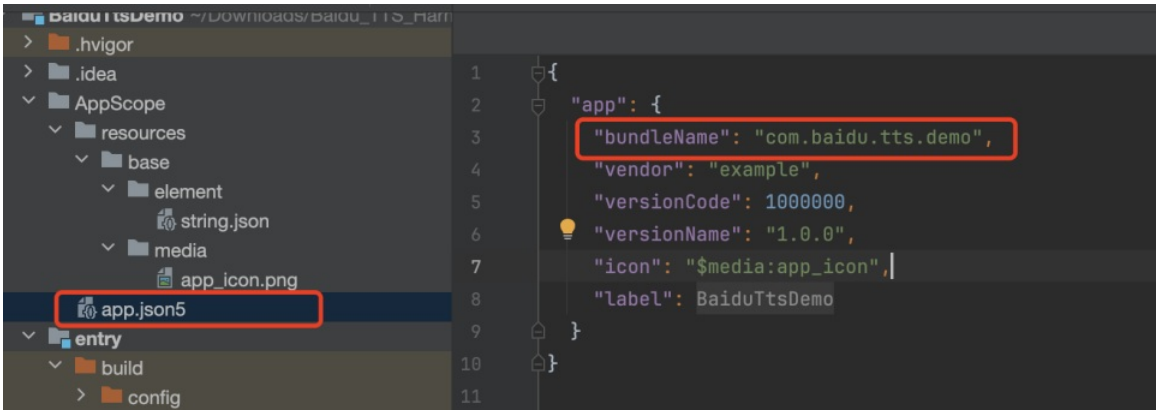
1.下载并解压Harmony SDK文件

不做任何修改，直接安装运行程序可以看到以下界面：



2.填入包名、AppID、APIKey（IAMKEY）、token鉴权信息。具体请按照以下步骤修改：

保证申请的鉴权信息与项目包名匹配



修改文件 entry/src/main/ets/pages/default/Index.ets

```
private setOnlineParam(speechSynthesizer: SpeechSynthesizer): void {
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ACCESS_TOKEN, "配置token或iamkey");
  // 请初始化您的在线发音人
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ONLINE_SPEAKER, "4100");
  // 在线超时时间，MIX模式超时会切换至离线，ONLINE模式会超时
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ONLINE_TIMEOUT, "6000");

// 设置公共参数
private setPublicParam(speechSynthesizer: SpeechSynthesizer): void {
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_APP_ID, "appid");
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PLAYER_USAGE, audio.StreamUsage.STREAM_
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PLAYER_VOLUME_MODE, audio.AudioVolumeMo
  // speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PLAYER_INTERRUPT_MODE, audio.Interru
}
private setOfflineParam(speechSynthesizer: SpeechSynthesizer): void {

  // 设置离线需要的认证参数，产品SN
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_AUTH_SERIAL_NUMBER, "sn");
  // 离线资源库路径，设置需要的离线发音人
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_OFFLINE_MODEL, this.setOfflineModel())
  // 初始化离线tts服务，服务会读取相应离线资源进行加载，此过程是耗时操作
  speechSynthesizer.loadOfflineTts().then(
    (ttsError: ITtsError) => {
      hilog.info(0x0000, 'index', 'loadOfflineTts ttsError errorCode= %{public}d , errorMes
        ttsError.getDetailCode(),
        ttsError.getDetailMessage());
    }
```

注意：把申请的APP_ID、AUTH_SERIAL_NUMBER（SN）分别填入对应位置

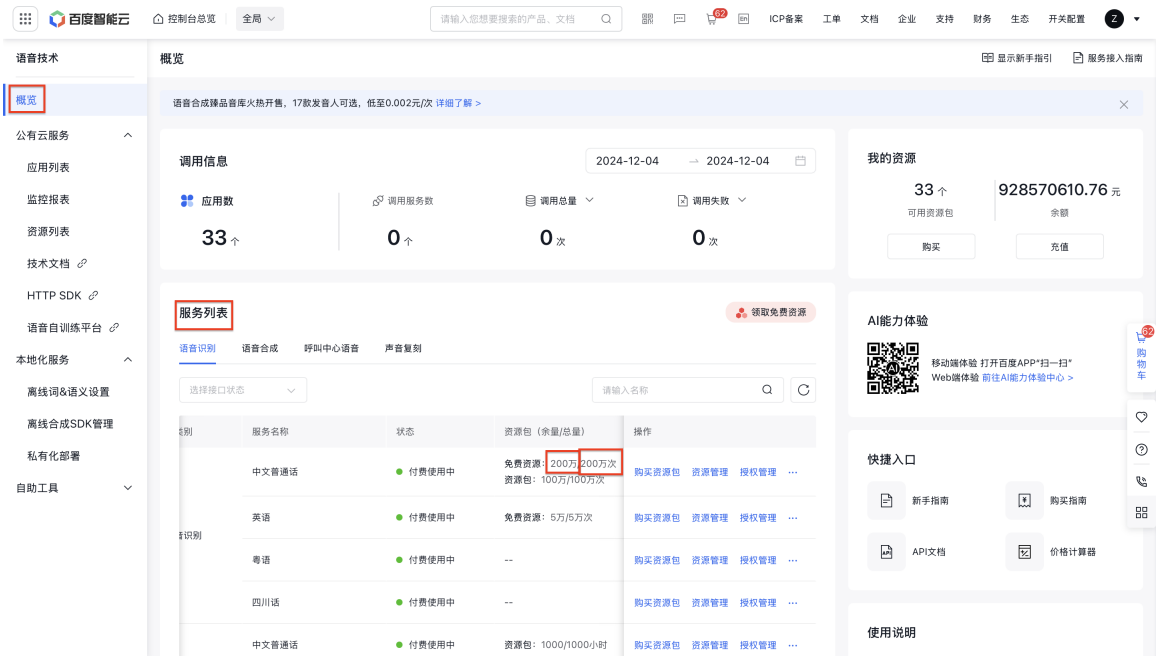
步骤二: 断网测试离线合成功能

按照上述文档修改完成后，安装app，首次使用需要联网，然后断网点击合成按钮进行离线语音合成，详情参考demo。

用量及历史调用查询

用量查询

1. 登录百度智能云控制台-语音能力引擎页面
2. 在概览页-服务列表查看用量及余量：



历史调用查询

1. 登录百度智能云控制台-语音能力引擎-监控报表页面
2. 在监控报表页面选择产品服务、应用、API、时间段等选项进行查询：



注意事项：

- 数据约有15分钟延迟
- 查询调用量时，选择API接口请定位到具体接口，例如：语音技术-短语音识别-中文普通话（语音技术产品计算单位有次数、时间、字数三种。如选择‘全部’，统计单位冲突导致不展示曲线图）。
- 调用失败详情内容为调用失败原因解释，可以根据描述、处理意见进行修改，也可以根据错误码配合技术文档进行排查。



常用音频处理工具

音频文件转码

简介

本文描述如何从其它格式的音频转成符合语音识别输入要求格式的音频文件。即4种格式的音频文件：

1. pcm（不压缩），也称为raw格式。音频输入最原始的格式，不用再解码。
2. wav（不压缩，pcm编码）：在pcm文件的开头出上加上一个描述采样率，编码等信息的字节。
3. amr（有损压缩格式），对音频数据进行有损压缩，类似mp3文件。
4. m4a（有损压缩格式，AAC编码），对音频数据进行有损压缩，通常仅供微信小程序使用的格式。自行转换比较复杂。

由于底层识别使用的是pcm，因此推荐直接上传pcm文件。如果上传其它格式，会在服务器端转码成pcm，调用接口的耗时会增加。

音频参数概念

- 采样率： 百度语音识别一般仅支持16000的采样率。即1秒采样16000次。
- 位深： 无损音频格式pcm和wav可以设置，百度语音识别使用16bits 小端序 ，即2个字节记录1/16000 s的音频数据。
- 声道： 百度语音识别仅支持单声道。

以16000采样率 16bits 编码的pcm文件为例，每个16bits(=2bytes)记录了 1/16000s的音频数据。即1s的音频数据为 2bytes * 16000 = 32000B

以下表格的的格式要求仅供参考，具体以调用的api或sdk的文档为准。

api或sdk	大类	采样率	文件格式	声道	pcm 及wav 编码
restapi	语音识别	16000	pcm，wav，amr，m4a	单声道	16bits 小端序
android ios linux C++ SDK	语音识别	16000	pcm	单声道	16bits 小端序
SDK mrcp server等	呼叫中心	8000	pcm	单声道	16bits 小端序

示例音频文件下载

音频播放

- pcm 播放 ，使用AudioAudition ，选择 16000采样率 ；16位PCM；Little-Endian（即默认字节序）
- wav, m4a 播放， 使用AudioAudition 或 完美解码
- amr 播放, 使用完美解码

转换命令示例

wav 文件转 16k 16bits 位深的单声道pcm文件

```
ffmpeg -y  -i 16k.wav  -acodec pcm_s16le -f s16le -ac 1 -ar 16000 16k.pcm
```

44100 采样率 单声道 16bts pcm 文件转 16000采样率 16bits 位深的单声道pcm文件

```
ffmpeg -y -f s16le -ac 1 -ar 44100 -i test44.pcm  -acodec pcm_s16le -f s16le -ac 1 -ar 16000 16k.pcm
```

mp3 文件转 16K 16bits 位深的单声道 pcm文件

```
ffmpeg -y  -i aidemo.mp3  -acodec pcm_s16le -f s16le -ac 1 -ar 16000 16k.pcm

// -acodec pcm_s16le pcm_s16le 16bits 编码器
// -f s16le 保存为16bits pcm格式
// -ac 1 单声道
//  -ar 16000  16000采样率
```

正常输出如下：

```
  Input #0, mp3, from 'aaa.mp3':
  Metadata:
    encoded_by      : Lavf52.24.1
  Duration: 00:02:33.05, start: 0.000000, bitrate: 128 kb/s
    Stream #0:0: Audio: mp3, 44100 Hz, stereo, s16p, 128 kb/s
// 输入音频， MP3格式， 44100采样率，stereo-双声道， 16bits 编码

Output #0, s16le, to '16k.pcm':
  Metadata:
    encoded_by      : Lavf52.24.1
    encoder         : Lavf57.71.100
    Stream #0:0: Audio: pcm_s16le, 16000 Hz, mono, s16, 256 kb/s

// 输入音频， MP3格式， 16000采样率，mono-单声道， 16bits
// 256 kb/s = 32KB/s = 32B/ms
```

ffmpeg 使用说明

简介

ffmpeg 的一个功能是转换不同的音频格式，其它简介请至http://ffmpeg.org/

linux 版本：http://www.ffmpeg.org/download.html#build-linux linux 静态编译版本：https://www.johnvansickle.com/ffmpeg/
windows 版本 ：http://ffmpeg.org/download.html#build-windows

ffmpeg官方文档地址：http://ffmpeg.org/ffmpeg.html

编译参数与支持格式

ffmpeg默认支持pcm与wav（pcm编码）格式，额外的编译参数如下：

- --enable-libopencore-amrnb 支持amr-nb(8000 采样率) 读写
- --enable-libopencore-amrwb 支持amr-wb(16000 采样率) 读取
- --enable-libvo-amrwbenc 支持amr-wb(16000 采样率) 写入
- --enable-libmp3lame 支持mp3 写入
- --enable-libfdk-aac 使用libfdk作为aac的编码和解码格式
- ffmpeg -codecs 可以查看所有的格式：

```
 D..... = Decoding supported  # 读取
.E.... = Encoding supported  # 写入
..A... = Audio codec      # 音频编码
....L. = Lossy compression # 有损
.....S = Lossless compression # 无损
DEA..S pcm_s16le            PCM signed 16-bit little-endian
DEA.LS wavpack              WavPack
DEA.L. mp3  MP3 (MPEG audio layer 3)
DEA.L. amr_nb       AMR-NB (Adaptive Multi-Rate NarrowBand)
DEA.L. amr_wb       AMR-WB (Adaptive Multi-Rate WideBand)
DEA.L. aac          AAC (Advanced Audio Coding) (decoders: aac aac_fixed )
或
DEA.L. aac          AAC (Advanced Audio Coding) (decoders: aac aac_fixed libfdk_aac ) (encoders: aac libfdk_aac )
```

ffmpeg命令

```
 ffmpeg {通用参数} {输入音频参数} {输出音频参数}

ffmpeg -y  -i aidemo.mp3  -acodec pcm_s16le -f s16le -ac 1 -ar 16000 16k.pcm

通用参数： -y
输入音频mp3文件参数： -i aidemo.mp3
输出音频 16000采样率 单声道 pcm 格式：  -acodec pcm_s16le -f s16le -ac 1 -ar 16000 16k.pcm
```

示例： 输入是 32000HZ的单声道 16bits pcm文件。查询下文的输入参数为 " -f s16le -ac 1 -ar 32000 -i test32.pcm" 输出是16000HZ的单声道 16bits pcm文件。查询下文的输出参数为 "-f s16le -ac 1 -ar 16000 16k.pcm" 常用参数选择 -y

合并如下：

```
 ffmpeg  -y  -f s16le -ac 1 -ar 32000  -i test32.pcm -f s16le -ac 1 -ar 16000 16k.pcm
```

输入音频参数

wav、amr、 mp3及m4a格式都自带头部， 含有采样率 编码 多声道等信息。而pcm为原始音频信息，没有类似头部。
wav（pcm编码）格式，仅仅在同样参数的pcm文件加了个几个字节的文件头。

输入 wav、amr、mp3及m4a 等格式：

```
 -i  test.wav # 或test.mp3 或者 test.amr
```

输入 pcm格式： pcm需要额外告知编码格式，采样率，单声道信息

```
 -acodec pcm_s16le -f s16le -ac 1 -ar 16000 -i 8k.pcm
// 单声道 16000 采样率  16bits编码 pcm文件

// s16le  s(signied)16(16bits)le(Little-Endian)
-acodec pcm_s16le：使用s16le进行编码
-f s16le 文件格式是s16le的pcm
-ac 1 ：单声道
-ar 16000 ： 16000采样率
```

输出音频参数

在原始采样率 大于或者接近16000的时候，推荐使用16000的采样率。 8000的采样率会降低识别效果。 输出wav和amr格式时，如果不指定输出编码器，ffmpeg会选取默认编码器。

输出pcm音频：

```
 -f s16le -ac 1 -ar 16000 16k.pcm
// 单声道 16000 采样率 16bits编码 pcm文件
```

输出wav 音频：

```
 -ac 1 -ar 16000 16k.wav
// 单声道 16000 采样率 16bits编码 pcm编码的wav文件
```

输出amr-nb 音频 ：

全称是：Adaptive Multi-Rate，自适应多速率，是一种音频编码文件格式，专用于有效地压缩语音频率。**在带宽不是瓶颈的情况下，不建议选择这种格式，解压需要百度服务器额外的耗时** amr-nb格式只能选 8000采样率。bit rates越高音质越好，但是文件越大。 bit rates 4.75k, 5.15k, 5.9k, 6.7k, 7.4k, 7.95k, 10.2k or 12.2k

8000的采样率及有损压缩会降低识别效果。如果原始采样率大于16000，请使用 amr-wb格式。

```
 -ac 1 -ar 8000 -ab 12.2k 8k-122.amr

// 8000 采样率 12.2 bitRates
```

输出 amr-wb 格式，采样率 16000。 bit rates越高音质越好，但是文件越大。 6600 8850 12650 14250 15850 18250 19850 23050 23850

```
 -acodec amr_wb -ac 1 -ar 16000 -ab 23850 16k-23850.amr
```

输出m4a文件

m4a文件一般来源于微信小程序录音（见restapi文档中微信录音小程序的参数说明）。不建议其它格式转为m4a；推荐转为amr有损压缩格式调用restapi。

如果您一定要转为百度支持的m4a格式，见后文的m4a文件转码

常用参数

- -y 覆盖同名文件
- -v 日志输出基本 如 -v ERROR -v quiet 等

查看音频格式ffprobe使用

查看语音合成生成的MP3格式信息：

```
 ffprobe -v quiet -print_format json -show_streams  aidemo.mp3
```

返回如下

```
{
  "streams": [
    {
      "index": 0,
      "codec_name": "mp3", // mp3 格式
      "codec_long_name": "MP3 (MPEG audio layer 3)",
      "codec_type": "audio",
      "codec_time_base": "1/16000",
      "codec_tag_string": "[0][0][0][0]",
      "codec_tag": "0x0000",
      "sample_fmt": "s16p",
      "sample_rate": "16000", // 16000采样率
      "channels": 1, // 单声道
      "channel_layout": "mono",
      "bits_per_sample": 0,
      "r_frame_rate": "0/0",
      "avg_frame_rate": "0/0",
      "time_base": "1/14112000",
      "start_pts": 0,
      "start_time": "0.000000",
      "duration_ts": 259096320,
      "duration": "18.360000",
      "bit_rate": "16000",
      "disposition": {
        "default": 0,
        "dub": 0,
        "original": 0,
        "comment": 0,
        "lyrics": 0,
        "karaoke": 0,
        "forced": 0,
        "hearing_impaired": 0,
        "visual_impaired": 0,
        "clean_effects": 0,
        "attached_pic": 0,
        "timed_thumbnails": 0
      }
    }
  ]
}
```

pcm文件音频时长计算

同图像bmp文件一样，pcm文件保存的是未压缩的音频信息。 16bits 编码是指，每次采样的音频信息用2个字节保存。可以对比下bmp文件用分别用2个字节保存RGB颜色的信息。 16000采样率 是指 1秒钟采样 16000次。常见的音频是44100HZ，即一秒采样44100次。 单声道： 只有一个声道。

根据这些信息，我们可以计算： 1秒的16000采样率音频文件大小是 2*16000 = 32000字节 ，约为32K 1秒的8000采样率音频文件大小是 2*8000 = 16000字节 ，约为 16K

如果已知录音时长，可以根据文件的大小计算采样率是否正常。

转换为m4a格式（AAC编码）

推荐使用amr有损压缩格式，m4a格式用于微信小程序的录音。

需要下载**MP4Box**，用于转换brand 为mp42:0, mini Version 0 。 restapi不支持brand M4A。点此下载，选Current release里的下载链接。

ffmpeg 官方推荐使用libfdk_aac 库，但这个库需要按照官方文档自行编译。如果使用静态版本的话，也可以使用ffmpeg自带的

aac库。

编译过libfdk_aac ffmpeg 示例

```
 ffmpeg -y -f s16le -ac 1 -ar 16000 -i 16k_57test.pcm -c libfdk_aac  -profile:a aac_low -b:a 48000 -ar 16000 -ac 1 16k.m4a
MP4Box -brand mp42:0 16k.m4a #这步不能忽略
```

静态版本自带的aac库示例

```
 ffmpeg -y -f s16le -ac 1 -ar 16000 -i 16k_57test.pcm -c aac  -profile:a aac_low -b:a 48000 -ar 16000 -ac 1 16k.m4a
MP4Box -brand mp42:0 16k.m4a #这步不能忽略
```

输出参数

- -c 选编码库 libfdk_aac或者aac
- -profile:a profile固定选aac_low（AAC-LC），restapi不支持 例如HE-AAC ，LD，ELD等
- -b:a bitrates ， 16000采样率对应的bitrates CBR 范围为 24000-96000。越大的话，失真越小，但是文件越大
- -ar 采样率，一般固定16000
- -ac 固定1，单声道

查看 m4a 格式

```
 ffprobe 16k.m4a
```

下面结果中重要的字段加粗

Metadata: major_brand : **mp42** minor_version : **0** compatible_brands: isomiso2M4A mp42 encoder : Lavf58.20.100 Duration: 00:00:56.26, start: 0.000000, bitrate: 32 kb/s Stream #0:0(und): Audio: aac (**LC**) (mp4a / 0x6134706D), **16000 Hz, mono**, fltp, 32 kb/s (default)

拼音相似度比较

简介

本java demo能将中文转换为拼音，并比较拼音相似度；进而可以在一组候选词中，找到读音与搜索词最为接近的词组。

该工具能够有效解决同音词、相似发音词、发音不标准等特殊情况下的语音识别。 例如，在通讯录场景下，如果用户需要识别的正确联系人是“张三”，但语音识别接口返回的结果可能是“张山”；此时则可以将用户的联系人列表和“张山”作为输入，通过本工具查找出与“张山”最匹配的结果，有效提高结果的准确率。具体可参考下方的示例。

示例说明

本工具的输入包括一组候选词和一个搜索词，输出是每个候选词与搜索词的距离（差异值）。注意：距离越短，相似度越高。

示例 输入1： 一组候选词，如："张三", "张散", "张丹", "张成", "李四", "李奎" 输入2： 一个搜索词, 如：“张山” 输出：每个候选词与搜索词的距离（如下）

```
[{word=张三, score=2}, {word=张散, score=3}, {word=张丹, score=4}, {word=张成, score=7}, {word=李四, score=18}, {word=李奎, score=20}]
```

该结果表示 “张山”与“张三” 发音最为相近，则“张三”是最有可能的正确词汇。

局限性举例

以下case将不能很好地适用于本算法，请使用的时候先行评估。

1.区分s和sh，可能相似

输入1	输入2	输出
sao3	shao3	2
sao3	pao3	2
sao3	dao3	2
sao3	hao3	2
输入1	**输入2**	**输出**
shi1	zhi1	2
shi1	chi1	2
shi1	si1	2
shi1	sha1	2

2.区分en和eng

输入1	输入2	输出
shen1	sheng1	2
shen1	zhen1	2
shen1	she1	2
输入1	**输入2**	**输出**
meng2	men2	2
meng2	ming2	2
meng2	mang2	2

3.区分i和ie

输入1	输入2	输出
xi1	pi1	2
xi1	mi1	2
xi1	xia1	2
xi1	xie1	2

4.区分yue和yu、ye、yun

输入1	输入2	输出
yue4	ye4	2
yue4	yun4	2
yue4	yu4	2

5.类似xi’an词语

输入1	输入2	输出
xi1,an1	xian1	3
xi1,an1	xie4,an1	3
xi1,an1	xin4,an1	3

重要代码

计算2个拼音的相似度，

使用方法如 getEditDistance("zhang1,san1", "zhang1,shan1");

```
public static int getEditDistance(String s, String t) {
    int d[][]; // matrix
    int n; // length of s
    int m; // length of t
    int i; // iterates through s
    int j; // iterates through t
    char s_i; // ith character of s
    char t_j; // jth character of t
    int cost; // cost
    // Step 1
    n = s.length();
    m = t.length();
    if (n == 0) {
        return m;
    }
    if (m == 0) {
        return n;
    }
    d = new int[n + 1][m + 1];

    // Step 2
    for (i = 0; i <= n; i++) {
        d[i][0] = i;
    }
    for (j = 0; j <= m; j++) {
        d[0][j] = j;
    }

    // Step 3
    for (i = 1; i <= n; i++) {
        s_i = s.charAt(i - 1);
        // Step 4
        for (j = 1; j <= m; j++) {
            t_j = t.charAt(j - 1);
            // Step 5
            cost = (s_i == t_j) ? 0 : 1;
            // Step 6
            d[i][j] = Minimum(d[i - 1][j] + 1, d[i][j - 1] + 1,
                    d[i - 1][j - 1] + cost);
        }
    }
    // Step 7
    return d[n][m];
}
```

完整示例代码下载（JAVA）

https://platform.bj.bcebos.com/sdk/asr/asr_doc/doc_download_files/similarwordsV1.zip

开源VAD音频切分工具

简介

由于百度rest api接口有60s的音频时长限制，使用此demo可以粗略地按照静音切分音频。

项目地址： https://github.com/Baidu-AIP/speech-vad-demo

集成 webrtc 开源项目，vad模块，具体算法*GMM*(Gaussian Mixture Model)。 注意这个是开源项目，效果远不如与百度语音 LInux C++ SDK ，Android及iOS SDK自带的VAD功能，切割的文件需为pcm(16000采样率 小端序 16bits)的格式，其他格式需要提前转换，切割后也是pcm文件

原理

一个FRAME时长默认10s，通过webrtc的vad计算这个FRAME是否是活动（ACTIVE: 有声音， INACTIVE：静音）。

这样可以获得音频的所有FRAME的活动值（ACTIVE或者INACTIVE）。从而在静音（INACTIVE）的音频段上的切分音频。

运行环境

- g++ 4.8以上,
- cmake 2.8 及 make 3.8 编译需要，可以跳过或者尝试使用低版本
- 任意操作系统

如果没有g++ 4.8的版本，centos可以参考“Linux C++ SDK”的文档，进行g++ 4.8的安装。

由于webrtc项目vad模块依赖于rtc_base模块 进行运行时的参数检查，因此需要可以编译rtc_base的。 如果您无法升级到g++ 4.8 或者是只能编译C项目，可以自行修改源代码，删除rtc_bas目录，及其它文件对rtc_base中宏的依赖,，剩余文件均为C代码。去除后请详细测试。

运行及结果

运行命令 `bash # 确认g++ 和cmake 版本 sh build_and_run.sh

```
 # 或
cmake . && make
```

结果 如果不修改代码的话，结果保存在 output_pcm内。 文件名如下

```
 16k_1.pcm_0-12989_A.pcm         // 第0-12989毫秒的音频，
16k_1.pcm_33730-47389_A.pcm     // 第33730-47389毫秒的音频
16k_1.pcm_0-4049_A.pcm          // A 表示此段有声音，不是完全的静音
...
16k_1.pcm_114060-121689_I.pcm  // I 表示此段都是静音
```

参数设置

common.h `c // MULTI = 1 2 3 webrtc vad可以设置分别为以10ms 20ms 30ms作为包 #define MULTI 1

```
 // VAD 模式 Aggressiveness mode (0, 1, 2, or 3). 数值越大，判断越是粗略，连着的静音或者响声增多
#define WEBRTC_VAD_MODE 3

// 有声音的音频段，xxx 毫秒后，认为该音频段结束，不能为0
#define FILE_CUT_SILENCE_AFTER_ACTIVE_MS 500

// 静音的音频段，切分xxx毫秒给之后有声音的音频段
#define FILE_CUT_SILENCE_BEFORE_ACTIVE_MS 300

// 最小的音频切分长度，即每个切分后音频文件不少于这个时长，最后一个除外
#define FILE_CUT_MIN_MS  (10 * 1000)

// 最大的音频切分长度，即每个切分后音频文件不多于这个时长
#define FILE_CUT_MAX_MS  (60 * 1000)

/** 上述示例中音频文件尽可能依次满足如下条件
1. 最短10s，最长 60s。
2. 每个音频文件的最后的静音500ms，
2. 每个音频文件的开始的静300ms。
 */
```

切割逻辑修改

demo因为考虑到流式，采用了尽快切割的方法。如果您对这个切割算法不满意的话，可以对照periods_print的结果, 自行修改file_cut.c内的代码逻辑

```
 // [5150, 5220) [5230, 6380) [6520, 6970) [7000, 8040) [8080, 8670)
// 表示 [5150, 5220) 等区间内是有声音的，其它区间均为静音，
// 如果需要切割的话，应该在静音区间选择合适的位置切割
```

购买指南

计费概述

计费简介

语音服务提供一定额度的 **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 > 预付费资源包 >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。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。**

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。

并发叠加包

以上两种计费方式开通后会保证相应的默认并发限制，如果并发量无法满足您业务需求，您可购买 **并发叠加包** 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。

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。

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

大模型声音复刻免费额度

大模型声音复刻

接口服务	认证状态	免费并发	免费调用次数/字符
大模型声音复刻-创建音色	个人认证	3并发	10次
大模型声音复刻-创建音色	企业认证	5并发	50次
大模型声音复刻-在线合成	个人认证	3并发	5万字符
大模型声音复刻-在线合成	企业认证	5并发	50万字符

端到端语音语言大模型免费额度

端到端语音语言大模型

接口服务	认证状态	免费并发	免费资源
端到端语音语言大模型（Lite）	个人认证	3并发	500千tokens
端到端语音语言大模型（Lite）	企业认证	5并发	1000千tokens
端到端语音语言大模型（Pro）	个人认证	3并发	500千tokens
端到端语音语言大模型（Pro）	企业认证	5并发	1000千tokens
搜索增强	个人认证	--	1000次
搜索增强	企业认证	--	10000次

语音识别免费额度

短语音识别

接口服务	认证状态	免费并发	免费调用次数
短语音识别标准版-中文普通话	个人认证	5并发	50千次
短语音识别标准版-中文普通话	企业认证	10并发	100千次
短语音识别标准版-英文	个人认证	3并发	30千次
短语音识别标准版-英文	企业认证	5并发	50千次
短语音识别标准版-粤语	个人认证	3并发	30千次
短语音识别标准版-粤语	企业认证	5并发	50千次
短语音识别标准版-四川话	个人认证	3并发	30千次
短语音识别标准版-四川话	企业认证	5并发	50千次
短语音识别极速版	个人认证	5并发	50千次
短语音识别极速版	企业认证	5并发	50千次

实时语音识别

接口服务	认证状态	免费并发	免费调用时长
实时语音识别-中文普通话	个人认证	3并发	10小时
实时语音识别-中文普通话	企业认证	5并发	10小时
实时语音识别-英文	个人认证	3并发	10小时
实时语音识别-英文	企业认证	5并发	10小时

音频文件转写

接口服务	认证状态	免费调用时长
音频文件转写-中文普通话	个人、企业认证	10小时
音频文件转写-英文	个人、企业认证	10小时
音频文件转写-音视频字幕（中文）	个人、企业认证	10小时

呼叫中心语音-音频文件转写（8k）

接口服务	认证状态	免费调用时长
呼叫中心语音-音频文件转写	个人、企业认证	10小时

呼叫中心语音-音频文件质检

接口服务	认证状态	免费调用时长
呼叫中心语音-音频文件质检	个人、企业认证	10小时

呼叫中心语音-语音识别（8k）

接口服务	认证状态	免费并发	免费调用时长
呼叫中心语音-语音识别（8k）	个人、企业认证	默认2-5并发	10小时

语音合成免费额度

同步语音合成（短文本在线合成）

接口服务	认证状态	免费并发	免费调用次数
基础音库	个人认证	3并发	50千次
基础音库	企业认证	5并发	100千次
精品音库	个人认证	3并发	50千次
精品音库	企业认证	5并发	100千次
臻品音库	个人认证	3并发	10千次
臻品音库	企业认证	5并发	20千次

长文本在线合成

接口服务	认证状态	免费调用字符数
基础音库	个人认证	5万字符
基础音库	企业认证	10万字符
精品音库	个人认证	5万字符
精品音库	企业认证	10万字符
臻品音库	个人认证	5万字符
臻品音库	企业认证	10万字符

流式文本在线合成

接口服务	认证状态	免费并发	免费调用字符数
基础音库	个人认证	3并发	5万字符
基础音库	企业认证	5并发	10万字符
精品音库	个人认证	3并发	5万字符
精品音库	企业认证	5并发	10万字符
臻品音库	个人认证	3并发	5万字符
臻品音库	企业认证	5并发	10万字符

呼叫中心语音-在线合成（8k）

接口服务	认证状态	免费并发	免费调用次数
呼叫中心语音-在线合成（8k）	个人、企业认证	默认2-5并发	100千次

离线语音合成

- 提供试用版授权序列号供用户集成测试使用，试用版授权序列号为按设备数授权（授权序列号与终端设备是1:1的关系）

- 您可通过试用版序列号跑通全流程，便于正式付费购买后的集成使用。
- 只有完成个人认证或企业认证的账号方可获取试用版授权序列号。序列号获取的具体流程参考 语音技术（baidu.com）

用户级别	试用版授权数量
未认证	0
个人认证	2
企业认证	5

> **说明：**
> 1、语音技术接口的免费额度自用户在语音技术控制台首次登录发放后生效，并永久有效。
> 2、首次登录语音技术控制台时，系统将根据登录的账号认证状态自动发放免费额度；若后续账号认证状态发生变化，系统将在用户再次访问语音技术控制台时，自动提升并发放免费额度。
> 3、每个账号根据认证信息仅发放一次免费额度，多账号绑定同一认证信息将无法重复发放。若系统判断存在违规重复开通免费额度，造成恶意请求百度语音服务的状况，百度有权进行处理而无需提前告知，由此产生的后果由用户自行承担。
> 4、百度语音服务全线升级，自2019年7月1日商用版本陆续上线，提供更好地服务和支持。如果您是2019年7月1日前调用过百度语音服务的老客户，或您计划使用百度语音能力用于公益产品及服务，可以邮件给ai-speech@baidu.com，申请获得更高的免费用量和用时，我们会结合语音产品服务资源以及您的使用情况等综合因素尽力为您提供支持。非常感谢您的使用和支持。 因AI而声公益计划详细介绍

产品价格

端到端语音语言大模型

端到端语音语言大模型

百度端到端语音语言大模型的费用由两部分构成：模型调用 Token 计费和搜索增强调用次数计费。调用接口时会自动开启搜索增强，以实现联网检索与实时信息获取。因此，请确保模型资源（Pro 版或 Lite 版）和搜索增强资源均有余量或已开启后付费，以免调用失败。

端到端语音语言大模型与大模型实时语音交互-搜索增强都支持"按Token包/次数包预付费"与"按调用量后付费"两种付费模式。

- 预付费：按Token包/次数包预付费，适合业务中对端到端语音语言大模型需求量较明确的用户选择。用户可同时购买多个包叠加使用。
- 后付费：按调用量后付费，更适合对端到端语音语言大模型需求量尚处于探索期的用户选择，也可作为预付费的兜底方案。

用户产生调用时会优先消耗免费调用额度，待免费调用额度耗尽时开始消耗Token包/次数包内额度。当Token包/次数包内剩余额度为零时，系统会自动切换为“按调用量阶梯后付费”的计费策略；直至账户内余额不足以支付因调用产生的费用，此时端到端语音语言大模型服务暂停，用户则无法正常使用端到端语音语言模大型服务。用户重新购买资源包或给账户充值后，端到端语音语言大模型服务将恢复正常。

端到端语音语言大模型（Pro） 端到端语音语言大模型（Pro）按消耗Token量计费，支持“按Token包预付费”和“按消耗Token量后付费”两种付费方式。

价目表-按Token包预付费 用户购买后即可直接使用，Token包购买之日起一年内有效，具体价格如下：

Token包规格（千tokens）	价格（元）	单价（元/千tokens）
2000	350	0.175
10000	1700	0.17
50000	7500	0.15
100000	14000	0.14
1000000	125000	0.125

价目表-按调用量后付费 按调用量后付费方式。系统按用户实际使用实时扣费，输入输出价格不同，Token类型分为音频和文本价格也不同，具体价格如下：

Token调用量	Token类型	价格（元/千tokens）
0-∞	输入-音频	0.064
0-∞	输入-文本	0.008
0-∞	输出-音频	0.24
0-∞	输出-文本	0.064

端到端语音语言大模型（Lite） 端到端语音语言大模型（Lite）按消耗Token量计费，支持“按Token包预付费”和“按消耗Token量后付费”两种付费方式。

价目表-按Token包预付费 用户购买后即可直接使用，Token包购买之日起一年内有效，具体价格如下：

Token包规格（千tokens）	价格（元）	单价（元/千tokens）
2000	70	0.035
10000	340	0.034
50000	1500	0.03
100000	2800	0.028
1000000	25000	0.025

价目表-按调用量后付费 按调用量后付费方式。系统按用户实际使用实时扣费，输入输出价格不同，Token类型分为音频和文本价格也不同，具体价格如下：

Token调用量	Token类型	价格（元/千tokens）
0-∞	输入-音频	0.016
0-∞	输入-文本	0.002
0-∞	输出-音频	0.06
0-∞	输出-文本	0.016

大模型实时语音交互-搜索增强 搜索增强按调用次数计费，支持“按次数包预付费”和“按调用次数后付费”两种付费方式。

价目表-按次数包预付费 用户购买次数包后即可直接使用，次数包购买之日起一年内有效，具体价格如下：

次数包规格	价格（元）	单价（元/次）
10000	352	0.0352
100000	3420	0.0342
1000000	30600	0.0306

价目表-按调用量后付费 按调用次数后付费方式。系统按用户实际使用实时扣费，具体价格如下：

调用次数	单价（元/次）
0-∞	0.036

并发说明 开通付费后，端到端语音语言大模型支持10并发。 免费测试支持并发详见 端到端语音语言大模型免费额度。

****费用计算示例** 示例一**：

某企业月度稳定调用端到端语音基础模型，月度总调用Token量约12000千tokens，为节省成本选择预付费叠加购包，所需支付的费用计算如下：

1700元（10000千Token包）+ 350元（2000千Token包）= **2050元**

示例二：

某公司业务中使用端到端功能，测试期间请求输入了300 Token的音频+100 Token的文本（其中包含system prompt），输出了300 Token的音频+80 Token的文本，使用了1次搜索增强。无免费剩余额度，所需支付的费用计算如下：

300*0.064/1000（输入音频）+100*0.008/1000（输入文本）+300*0.24/1000（输出音频）+80*0.064/1000（输出文本）+1*0.036（搜索增强）=**0.13312元**

大模型声音复刻

大模型声音复刻

百度大模型声音复刻使用费用由创建音色、在线合成两类基础费用构成。

- 创建音色：实现目标音色的提取与创建，每调用一次接口并成功创建音色就计费一次。
- 在线合成：实现目标音色的语音合成，根据合成字符数计费。

创建音色与在线合成都支持"按次数包/字符包预付费"与"按调用量后付费"两种付费模式。

- 预付费：按次数包/字符包预付费，适合业务中对音色或合成字符需求量较明确的用户选择。用户可同时购买多个次数包/字符包叠加使用。
- 后付费：按调用量后付费，更适合对声音复刻需求量尚处于探索期的用户选择，也可作为预付费的兜底方案。

用户产生调用时会优先消耗免费调用额度，待免费调用额度耗尽时开始消耗次数包/字符包内额度。当次数包/字符包内剩余额度为零时，系统会自动切换为“按调用量阶梯后付费”的计费策略；直至账户内余额不足以支付因调用产生的费用，此时语音识别服务暂停，用户则无法正常使用语音识别服务。用户重新购买次数包/字符包或给账户充值后，语音识别服务将恢复正常。

创建音色

价目表-按复刻音色数预付费

用户购买次数包后即可直接使用，次数包购买之日起一年内有效，具体价格如下：

次数包规格（个音色）	价格（元）	次单价（元/个音色）
50	400	8
200	1400	7
1000	6000	6
5000	25000	5
20000	80000	4

价目表-按复刻音色数后付费

按调用次数后付费方式。系统按用户实际使用实时扣费，具体价格如下：

复刻数量（个音色）	价格（元/个音色）
0-∞	8.8

> **购买说明**
>
> 可对同一个人进行多次音色复刻并同时调用。
>
> **示例一**：
> 给10个人进行声音复刻，每人复刻一次，累计复刻10次，计费为10次。
>
> **示例二**：
> 给1个人进行声音复刻，同一个人复刻了10次，计费为10次。

在线合成

大模型声音复刻-在线合成按调用字符计费，支持“按字符包预付费”和“按调用字符后付费”两种付费方式。

价目表-按字符数预付费 用户购买字符包后即可直接使用，字符包购买之日起一年内有效，具体价格如下：

字符包规格（万字符）	价格（元）	单价（元/万字符）
10	68	6.8
100	650	6.5
500	3000	6
1000	5500	5.5
5000	25000	5
10000	45000	4.5

价目表-按字符数后付费

按调用字符后付费方式。系统按用户实际使用实时扣费，具体价格如下：

调用字符（万字符）	价格（元/万字符）
0-∞	7

> **购买说明**
>
> 计费按照输入文本的累计字符数进行计费（1个中文字、英文字母、数字或符号均算作1个字符）

并发说明

开通付费后，大模型声音复刻创建音色与在线合成均支持10并发。 免费测试支持并发详见 大模型声音复刻免费额度。

****费用计算示例****

示例一：
用户业务预计需要300个音色，通过预付费可购买1个200音色资源包和2个50个音色资源包，所需支付的费用计算如下：
（1x1400）+（2x400）=2200（元）

示例二：

用户业务预计需要300个音色，通过后付费所需支付的费用计算如下： 8.8x300=2640（元）

语音合成

在线语音合成

同步语音合成（短文本在线合成）

价目表-按次数包预付费

用户购买次数包后即可直接使用，次数包购买之日起一年内有效，具体价格如下：

> **购买说明：**
>
> 大模型音库与臻品音库价格相同，官网控制台开通或购买臻品音库后，即可同时使用大模型音库和臻品音库

基础音库

次数包规格（千次）	支持并发	价格（元）	单价（元/千次）
50	100	70	1.4
1,000	100	1200	1.2
5,000	100	5000	1.0
10,000	100	8000	0.8
50,000	100	32500	0.65
100,000	100	50000	0.5

精品音库

次数包规格（千次）	支持并发	价格（元）	单价（元/千次）
10	100	32	3.2
1,000	100	3000	3.0
5,000	100	14000	2.8
10,000	100	25000	2.5
50,000	100	100000	2.0
100,000	100	150000	1.5

大模型&臻品音库

次数包规格（千次）	支持并发	价格（元）	单价（元/千次）
10	10	42	4.2
1,000	10	4000	4.0
5,000	10	17500	3.5
10,000	10	30000	3.0
50,000	10	125000	2.5
100,000	10	200000	2.0

价目表-按调用量后付费

分段阶梯后付方式，调用单价随用户当前自然月累积调用量所落阶梯区间而变化。月初，用户累积调用量清零，同时重新开始

累计调用量。系统按用户实际使用实时扣费，具体价格如下：

基础音库

月调用量（千次）	支持并发	单价（元/千次）
0<调用次数<=6,000	100	2.0
6,000<调用次数<=30,000	100	1.6
30,000<调用次数<=60,000	100	1.25
60,000<调用次数<=150,000	100	1.0
150,000<调用次数	100	0.65

精品音库

月调用量（千次）	支持并发	单价（元/千次）
0<调用次数<=6,000	100	4.0
6,000<调用次数<=30,000	100	3.2
30,000<调用次数<=60,000	100	2.5
60,000<调用次数<=150,000	100	2.0
150,000<调用次数	100	1.5

大模型&臻品音库

月调用量（千次）	支持并发	单价（元/千次）
0<调用次数<=6,000	10	4.5
6,000<调用次数<=30,000	10	3.8
30,000<调用次数<=60,000	10	3.0
60,000<调用次数<=150,000	10	2.5
150,000<调用次数	10	2.0

价目表-并发扩容费

用户可根据业务需要，扩容并发量，具体价格如下：

基础音库

购买时长	价格（元/并发）
按月	100
按年	1000

精品音库

购买时长	价格（元/并发）
按月	200
按年	2000

大模型&臻品音库

购买时长	价格（元/并发）
按月	200
按年	2000

长文本在线合成

价目表-按字符包预付费

用户购买字符包后即可直接使用，字符包购买之日起一年内有效，具体价格如下：

基础音库

字符包规格（万字符）	价格（元）	万字符单价（元）
10	19	1.9
100	170	1.7
500	750	1.5
1000	1300	1.3
5000	4500	0.9
10000	5000	0.5

精品音库

字符包规格（万字符）	价格（元）	万字符单价（元）
10	26	2.6
100	240	2.4
500	1100	2.2
1000	2000	2
5000	7500	1.5
10000	10000	1

大模型&臻品音库

字符包规格（万字符）	价格（元）	万字符单价（元）
10	32	3.2
100	300	3
500	1250	2.5
1000	2200	2.2
5000	8500	1.7
10000	15000	1.5

价目表-按调用量后付费

分段阶梯后付方式，调用单价随用户当前自然月累积调用量所落阶梯区间而变化。月初，用户累积调用量清零，同时重新开始累计调用量。系统按用户实际使用实时扣费，具体价格如下：

基础音库

月调用量（万字符）	价格（元/万字符）
0<调用字数<=600	2
600<调用字数<=3000	1.8
3000<调用字数<=6000	1.5
6000<调用字数<=15000	1.2
15000<调用字数	1

精品音库

月调用量（万字符）	价格（元/万字符）
0<调用字数<=600	2.7
600<调用字数<=3000	2.5
3000<调用字数<=6000	2.3
6000<调用字数<=15000	2
15000<调用字数	1.5

大模型&臻品音库

月调用量（万字符）	价格（元/万字符）
0<调用字数<=600	3.5
600<调用字数<=3000	3
3000<调用字数<=6000	2.5
6000<调用字数<=15000	2.2
15000<调用字数	2

流式文本在线合成

价目表-按字符包预付费

用户购买字符包后即可直接使用，字符包购买之日起一年内有效，具体价格如下：

基础音库

字符包规格（万字符）	支持并发	价格（元）	万字符单价（元）
10	10	19	1.9
100	10	170	1.7
500	10	750	1.5
1000	10	1300	1.3
5000	10	4500	0.9
10000	10	5000	0.5

精品音库

字符包规格（万字符）	支持并发	价格（元）	万字符单价（元）
10	10	26	2.6
100	10	240	2.4
500	10	1100	2.2
1000	10	2000	2
5000	10	7500	1.5
10000	10	10000	1

大模型&臻品音库

字符包规格（万字符）	支持并发	价格（元）	万字符单价（元）
10	10	32	3.2
100	10	300	3
500	10	1250	2.5
1000	10	2200	2.2
5000	10	8500	1.7
10000	10	15000	1.5

价目表-按调用量后付费

分段阶梯后付方式，调用单价随用户当前自然月累积调用量所落阶梯区间而变化。月初，用户累积调用量清零，同时重新开始累计调用量。系统按用户实际使用实时扣费，具体价格如下：

基础音库

月调用量（万字符）	支持并发	价格（元/万字符）
0<调用字数<=600	10	2
600<调用字数<=3000	10	1.8
3000<调用字数<=6000	10	1.5
6000<调用字数<=15000	10	1.2
15000<调用字数	10	1

精品音库

月调用量（万字符）	支持并发	价格（元/万字符）
0<调用字数<=600	10	2.7
600<调用字数<=3000	10	2.5
3000<调用字数<=6000	10	2.3
6000<调用字数<=15000	10	2
15000<调用字数	10	1.5

大模型&臻品音库

月调用量（万字符）	支持并发	价格（元/万字符）
0<调用字数<=600	10	3.5
600<调用字数<=3000	10	3
3000<调用字数<=6000	10	2.5
6000<调用字数<=15000	10	2.2
15000<调用字数	10	2

价目表-并发扩容费 基础音库

购买时长	价格（元/并发）
按月	100
按年	1000

精品音库

购买时长	价格（元/并发）
按月	200
按年	2000

大模型&臻品音库

购买时长	价格（元/并发）
按月	200
按年	2000

呼叫中心语音-在线合成

按调用次数进行计费，支持“按次数包预付费“和“调用量后付费”两种付费方式。

用户产生调用时会优先消耗免费调用额度，待免费调用额度耗尽时开始消耗次数包额度。当次数包内剩余额度为零时，系统会自动切换为“调用量后付费”的计费策略；直至账户内余额不足以支付因调用产生的费用，此时语音合成服务暂停，查询语音转写执行结果将返回错误码。用户则无法正常使用语音合成服务。用户重新购次数包或给账户充值后，语音合成服务将恢复正常。

价目表-按次数包预付费 用户购买次数包后即可直接使用，次数包购买之日起一年内有效，具体价格如下：

次数包规格（千次）	支持并发	价格（元）	单价（元/千次）
10	50	18	1.8
1,000	50	1500	1.5
10,000	50	12000	1.2
100,000	50	90000	0.9
1,000,000	50	600000	0.6

价目表-按调用量后付费 系统按用户实际使用实时扣费，具体价格如下：

月调用次数	支持并发	单价（元/千次）
0-∞	50	2.2

离线语音合成

按设备数授权

（点击链接立即线上购买）

价目表-购买授权数

购买授权数	每个授权单价
第100～1000个	5元/个
第1001～5000个	3元/个
第5001～20000个	2.5元/个
第20001～50000个	2元/个
大于50000个	线下商务沟通

价目表-购买精品发音人

精品发音人	每个发音人单价
度小童、度小萌、度小娇、度米朵、度博文、度小译、度小乔、度小鹿、度小台、度小粤、度小贤、度小雯 点击试听	2000元/个

> **购买说明：**
>
> 1. 授权单价包含软件授权费用和默认发音人授权费用。默认发音人包括度小宇、度小美、度逍遥、度丫丫共4个基础发音人，如需精品发音人需额外购买。
> 2. 离线合成SDK按设备数授权100起售。
> 3. 授权购买是「分段阶梯计费」，按照单笔订单内购买的授权数量所落入的每段区间，进行分段计费。例如您一次性购买1100个授权时，前1000个授权在0~1000的区间内，后100个授权在1001~5000的区间内，需付费5元/个×1000个+3元/个×100个。
> 4. 如您需要购买的授权数量为3000，除默认发音人外您还需要度博文、度米朵总计2个精品发音人，则总费用为1000×5 + 2000×3 + 2000×2 = 15000元。
> 5. 精品音库需配合装机使用，与每次购买的数量绑定，仅支持单次订单购买的授权使用。例如第一次购买1000个授权数+1个精品音库；第二次购买5000个授权数，未购买精品音库，则第一次购买的精品音库，不支持第二次购买的数量使用。
> 6. SDK购买激活后，或购买超过7天，不支持退款。
> 7. 如您需要购买的授权数量大于50000，请通过合作咨询联系我们，线下商务沟通。

按产品线授权

（点击链接立即线上购买）

基础版首次购买定价

授权包规格	价格	每授权单价	有效期
100000个	60000元	0.6元/个	一年
500000个	250000元	0.5元/个	一年

> **授权价格说明**
>
> 1. 离线合成SDK按产品线授权-基础版包含软件授权费用和4个基础音库授权费用。
> 2. 基础版中提供度小宇、度小美、度逍遥、度丫丫共4个基础音库，如需精品音库请购买精品版。
> 3. 授权有效期计算自订单完成日期开始按自然年计算，如2020年1月14日完成订单，则授权有效期为2020年1月14日-2021年1月14日。
> 4. 如您需要购买的授权数量大于300W，请通过合作咨询联系我们。

基础版增量购买定价

授权包规格	价格	有效期
100000个	135元/天	下单时系统自动计算

> **授权价格说明**
>
> 1. 购买授权增量包时，有效期为系统基于下单日期以及现有授权的剩余天数自动计算，使增购的授权最终有效期与现有授权有效期一致。
> 2. 系统基于每10万个授权每天的单价×有效期自动计算得出。
>
> **购买示例**
>
> 以按产品线授权-基础版为例，如当前已有授权的到期时间为2020年12月30日，授权总数50万，在2020年11月31日购买授权增量包时：
>
> 1. 系统将自动计算有效期为31天。
> 2. 单价为每10万个基础版授权每天的单价×有效期31天。
> 3. 若选择3个10万个规格的授权增量包，则最终消费为单价×3。
> 4. 购买完成后，产品线授权-基础版的授权总数变为50万+10万×3=80万，到期时间仍为2020年12月30日。

基础版延期定价

授权包规格	价格	每授权单价	有效期
100000个	50000元	0.5元/个	一年

> **授权价格说明**
>
> 1. 延长授权有效期时，为延长现有授权的有效期一年，系统自动计算购买数量，不可编辑。如现有授权总数60万个，则购买数量为6。
> 2. 完成购买后，全量授权的有效期将延期一年，包括未分配授权以及此前已经分配到应用的授权。

精品版首次购买定价

授权包规格	价格	每授权单价	有效期
100000个	80000元	0.8元/个	一年
500000个	360000元	0.72元/个	一年

授权价格说明

1. 离线合成SDK按产品线授权-精品版包含**软件授权费用和4个基础音库+12个精品音库授权费用**。
2. 精品版包含音库：度小宇、度小美、度逍遥、度丫丫、度小童、度小萌、度小娇、度米朵、度博文、度小译、度小乔、度小鹿、度小台、度小粤、度小贤、度小雯。
3. 授权有效期计算自订单完成日期开始按自然年计算，如2020年1月14日完成订单，则授权有效期为2020年1月14日-2021年1月14日。

精品版增量购买定价

授权包规格	价格	有效期
100000个	195元/天	下单时系统自动计算

授权价格说明

1. 购买授权增量包时，有效期为系统基于下单日期以及现有授权的剩余天数自动计算，使增购的授权最终有效期与现有授权有效期一致。
2. 系统基于每10万个授权每天的单价×有效期自动计算得出。

购买示例

以按产品线授权-精品版为例，如当前已有授权的到期时间为2020年12月30日，授权总数50万，在2020年11月31日购买授权增量包时：

1. 系统将自动计算有效期为31天。
2. 单价为每10万个精品版授权每天的单价×有效期31天。
3. 若选择3个10万个规格的授权增量包，则最终消费为单价×3。
4. 购买完成后，产品线授权-精品版的授权总数变为50万+10万×3=80万，到期时间仍为2020年12月30日。

精品版延期定价

授权包规格	价格	每授权单价	有效期
100000个	72000元	0.72元/个	一年

授权价格说明

1. 延长授权有效期时，为延长现有授权的有效期一年，系统自动计算购买数量，不可编辑。如现有授权总数60万个，则购买数量为6。
2. 完成购买后，全量授权的有效期将延期一年，包括未分配授权以及此前已经分配到应用的授权。

购买说明：

1. 按产品线授权方式的授权序列号和APP应用相绑定，一个APP应用的APPID对应一个序列号，一个序列号可支持对应的APP应用安装在多台终端设备上。
2. 按产品线授权方式以10W装机量起售，装机量越多优惠力度越大。
3. SDK购买激活后，或购买超过7天，不支持退款。
4. 如需购买300W以上装机量，请通过合作咨询联系我们，线下商务沟通。

语音识别

短语音识别标准版

短语音识别标准版按调用次数计费，支持“按次数包预付费”和“按调用量阶梯后付费”两种付费方式，并基于中文普通话、粤语、英语、四川话等模型购买。

- “按次数包预付费”的付费方式更适合业务中对语音识别服务需求量较明确的用户选择，用户可同时购买多个次数包叠加使用；
- “按调用量阶梯后付费”的付费方式更适合对语音识别需求量尚处于探索期的用户选择，也可作为“按次数包预付费”付费方式的兜底方案。

用户产生调用时会优先消耗免费调用额度，待免费调用额度耗尽时开始消耗次数包内额度。当次数包内剩余额度为零时，系统会自动切换为“按调用量阶梯后付费”的计费策略；直至账户内余额不足以支付因调用产生的费用，此时语音识别服务暂停，用户则无法正常使用语音识别服务。用户重新购买次数包或给账户充值后，语音识别服务将恢复正常。

若用户需要更大并发量以满足业务需要，可自行完成并发量扩容。

按次数包预付费

用户购买次数包后即可直接使用，次数包购买之日起一年内有效，具体价格如下：

- 短语音识别-中文普通话、短语音识别-英语、短语音识别-粤语、短语音识别-四川话，价格适用于以上四个模型

次数包（千次）	支持并发	价格（元）
1,000	50	2400
10,000	50	19200
100,000	50	144000
1,000,000	50	960000

按调用量阶梯后付费

分段阶梯后付方式，调用单价随用户当前自然月累积调用量所落阶梯区间而变化。月初，用户累积调用量清零，同时重新开始累计调用量。系统按用户实际使用实时扣费，具体价格如下：

月调用量（千次）	支持并发	单价（元/千次）
0<调用次数<=6,000	50	3.4
6,000<调用次数<=30,000	50	2.8
30,000<调用次数<=60,000	50	2.2
60,000<调用次数<=150,000	50	1.5
150,000<调用次数	50	1.1

并发量扩容

用户可根据业务需要，扩容并发量，具体价格如下：

购买并发	价格（元/并发）
按月	80
按年	800

补充说明

1. 当且仅当用户产生成功调用或因鉴权、音频质量导致的失败调用时，系统会消耗、记录调用次数，其余情况均不涉及计费。

2. 月调用量统计周期为自然月。

短语音识别极速版

短语音识别极速版按调用次数计费，支持“按次数包预付费”和“按调用量阶梯后付费”两种付费方式。两种计费方式策略见“短语音识别标准版”

按次数包预付费

用户购买次数包后即可直接使用，次数包购买之日起一年内有效，具体价格如下：

次数包（千次）	支持并发	价格（元）
1,000	50	3000
10,000	50	24000
100,000	50	180000
1,000,000	50	1200000

按调用量阶梯后付费

分段阶梯后付方式，调用单价随用户当前自然月累积调用量所落阶梯区间而变化。月初，用户累积调用量清零，同时重新开始累计调用量。系统按用户实际使用实时扣费，具体价格如下：

月调用量（千次）	支持并发	单价（元/千次）
0<调用次数<=6,000	50	4.2
6,000<调用次数<=30,000	50	3.6
30,000<调用次数<=60,000	50	2.9
60,000<调用次数<=150,000	50	1.9
150,000<调用次数	50	1.4

并发量扩容

用户可根据业务需要，扩容并发量，具体价格如下：

购买并发	价格（元/并发）
按月	100
按年	1000

示例

以短语音识别极速版收费标准举例： 如从2019年3月1日至2019年3月31日，本月调用短语音识别极速版的总次数中，计费调用量为4000万次。费用如下：

- 前600万次落入0~600万次调用阶梯，单价0.0042元，费用为25200元。
- 中间2400万次落入600~3000万次调用阶梯，单价0.0036元，费用为86400元。
- 后1000万次落入3000~6000万次调用阶梯，单价0.0029元，费用为29000元。 本月费用共计：140600元

实时语音识别

实时语音识别按调用时长计费，支持“按小时包预付费”和“调用时长后付费”两种付费方式。并基于中文普通话、英语等模型购买。

- “小时包预付费”的付费方式更适合业务中对语音识别服务需求量较明确的用户选择，用户可同时购买多个小时包叠加使用；
- “调用时长后付费”的付费方式更适合对语音识别需求量尚处于探索期的用户选择，也可作为“小时包预付费”付费方式的兜底方案。

用户产生调用时会优先消耗免费调用额度，待免费调用额度耗尽时开始消耗小时包内额度。当小时包内剩余额度为零时，系统会自动切换为“调用时长后付费”的计费策略；直至账户内余额不足以支付因调用产生的费用，此时语音识别服务暂停，用户则无法正常使用语音识别服务。用户重新购小时包或给账户充值后，语音识别服务将恢复正常。

若用户需要更大并发量以满足业务需要，可自行完成并发量扩容。

按小时包预付费

用户购买小时包后即可直接使用，小时包购买之日起一年内有效，具体价格如下：

- 实时语音识别-中文普通话、实时语音识别-英语接口，适用于手机近场语音输入场景

小时包规格（小时）	支持并发	价格（元）	单价（元/小时）
1000	50	1800	1.8
10000	50	15000	1.5
100000	50	120000	1.2
500000	50	450000	0.9

按调用时长后付费

系统按用户实际使用,每小时出账单实时扣费，账户内需保留足量余额，具体价格如下：

- 实时语音识别-中文普通话、实时语音识别-英语接口，适用于手机近场语音输入场景

月调用小时	支持并发	价格（元/小时）
0-∞	50	3

高并发量扩容

- 实时语音识别-中文普通话、实时语音识别-英语接口，适用于手机近场语音输入场景

购买并发	价格（元/并发）
按月	100
按年	1000

补充说明

1. 当且仅当用户产生成功调用或因鉴权、音频质量导致的失败调用时，系统会消耗、记录调用时长，其余情况均不涉及计费。

2. 月调用时长统计周期为自然月。

示例

如从2019-3-1至2019-3-31，本月调用实时语音识别-中文普通话的总时长中，计费调用量为4000小时。

费用如下：

按小时包预付费购买：4000/1000*1800=7200元

按调用时长后付费购买：4000*3=12000元

呼叫中心语音-语音识别（8K）

按调用时长计费，支持“按小时包预付费”和“调用时长后付费”两种付费方式。其中，“小时包预付费”的付费方式更适合业务中对语音识别服务需求量较明确的用户选择，用户可同时购买多个小时包叠加使用；而“调用时长后付费”的付费方式更适合对语音识别需求量尚处于探索期的用户选择，也可作为“小时包预付费”付费方式的兜底方案。

用户产生调用时会优先消耗免费调用额度，待免费调用额度耗尽时开始消耗小时包内额度。当小时包内剩余额度为零时，系统会自动切换为“调用时长后付费”的计费策略；直至账户内余额不足以支付因调用产生的费用，此时语音识别服务暂停，用户则无法正常使用语音识别服务。用户重新购小时包或给账户充值后，语音识别服务将恢复正常。

若用户需要更大并发量以满足业务需要，可自行完成并发量扩容。

按小时包预付费

用户购买次数包后即可直接使用，次数包购买之日起一年内有效，具体价格如下：

小时包规格（小时）	支持并发	价格（元）	单价（元/小时）
1000	50	3000	3.0
10000	50	24000	2.4
100000	50	180000	1.8
1000000	50	1200000	1.2

按调用时长后付费

月初，用户累积调用时长清零，同时重新开始累计调用时长。系统按用户实际使用实时扣费，具体价格如下：

月调用小时	支持并发	价格（元/小时）
0-∞	50	3.3

高并发量扩容

购买时长	价格（元/并发）
按月	100
按年	1000

补充说明

1、 当用户产生成功调用，系统会消耗、记录调用时长，或因鉴权、音频质量导致的失败调用时，系统会消耗、记录调用时长，其余调用失败情况均不涉及计费。

2、 月调用量统计周期为自然月。

音频文件转写

音频文件转写按调用时长计费，支持“按时长包预付费”和“按调用时长后付费”两种付费方式。并基于中文普通话、英文、音视频字幕（中文）等模型购买。

- “小时包预付费”的付费方式更适合业务中对语音识别服务需求量较明确的用户选择，用户可同时购买多个小时包叠加使用；
- “调用时长后付费”的付费方式更适合对语音识别需求量尚处于探索期的用户选择，也可作为“小时包预付费”付费方式的兜底方案。

用户产生调用时会优先消耗免费调用额度，待免费调用额度耗尽时开始消耗小时包内额度。当小时包内剩余额度为零时，系统会自动切换为“调用时长后付费”的计费策略；直至账户内余额不足以支付因调用产生的费用，此时语音识别服务暂停，用户则无法正常使用语音识别服务。用户重新购小时包或给账户充值后，语音识别服务将恢复正常。

音频文件转写为异步任务、排队处理，用户无需进行并发扩容。

按小时包预付费

用户购买小时包后即可直接使用，小时包购买之日起一年内有效，具体价格如下：

- 音频文件转写-中文普通话、音频文件转写-英语接口，适用于手机近场语音输入场景

小时包规格（小时）	价格（元）	单价（元/小时）
1000	1200	1.2
10000	9000	0.9
100000	70000	0.7
500000	300000	0.6

- 音频文件转写-音视频字幕（中文）接口，适用于音视频内容分析、质检审核、字幕生产等场景

小时包规格（小时）	价格（元）	单价（元/小时）
1000	1560	1.56
10000	12800	1.28
100000	108000	1.08
500000	468000	0.936

按调用时长后付费

系统按用户实际使用,每小时出账单实时扣费，账户内需保留足量余额，具体价格如下：

- 音频文件转写-中文普通话、音频文件转写-英语接口，适用于手机近场语音输入场景

月调用小时	价格（元）
0-∞	2

- 音频文件转写-音视频字幕（中文）接口，适用于音视频内容分析、质检审核、字幕生产等场景

月调用小时	价格（元）
0-∞	2.5

呼叫中心语音-音频文件质检

呼叫中心音频文件质检按调用时长计费，支持“按时长包预付费”和“按调用时长后付费”两种付费方式。

- “小时包预付费”的付费方式更适合业务中对音频文件质检服务需求量较明确的用户选择，用户可同时购买多个小时包叠加使用；
- “调用时长后付费”的付费方式更适合对音频文件质检需求量尚处于探索期的用户选择，也可作为“小时包预付费”付费方式的兜底方案。

用户产生调用时会优先消耗免费调用额度，待免费调用额度耗尽时开始消耗小时包内额度。当小时包内剩余额度为零时，系统会自动切换为“调用时长后付费”的计费策略；直至账户内余额不足以支付因调用产生的费用，此时语音识别服务暂停，用户则无法正常使用语音识别服务。用户重新购小时包或给账户充值后，音频文件质检服务将恢复正常。

按小时包预付费

用户购买小时包后即可直接使用，小时包购买之日起一年内有效，具体价格如下：

小时包规格（小时）	价格（元）	单价（元/小时）
1000	2000	2
10000	17000	1.7
100000	140000	1.4
500000	600000	1.2
1000000	900000	0.9

按调用时长后付费

系统按用户实际使用,每小时出账单实时扣费，账户内需保留足量余额，具体价格如下：

月调用小时	价格（元）
0-∞	2.7

呼叫中心语音-音频文件转写（8K）

按调用时长计费，支持“按小时包预付费”和“调用时长后付费”两种付费方式。其中，“小时包预付费”的付费方式更适合业务中对语音转写服务需求量较明确的用户选择，用户可同时购买多个小时包叠加使用；而“调用时长后付费”的付费方式更适合对语音转写服务需求量尚处于探索期的用户选择，也可作为“小时包预付费”付费方式的兜底方案。

用户产生调用时会优先消耗免费调用额度，待免费调用额度耗尽时开始消耗小时包内额度。当小时包内剩余额度为零时，系统会自动切换为“调用时长后付费”的计费策略；直至账户内余额不足以支付因调用产生的费用，此时语音转写服务暂停，查询语音转写执行结果将返回错误码。用户无法正常使用语音转写服务。用户重新购小时包或给账户充值后，语音转写服务将恢复正常。

按小时包预付费

用户购买小时包后即可直接使用，小时包购买之日起一年内有效，具体价格如下：

小时包规格（小时）	价格（元）	单价（元/小时）
1000	1600	1.6
10000	12000	1.2
100000	80000	0.8

按调用时长后付费

月初，用户累积调用时长清零，同时重新开始累计调用时长。系统按用户实际使用实时扣费，具体价格如下：

月调用小时	价格（元/小时）
0-∞	2.2

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。目前语音技术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. 购买预付费资源包：预付费资源包，一次购买全年使用

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：

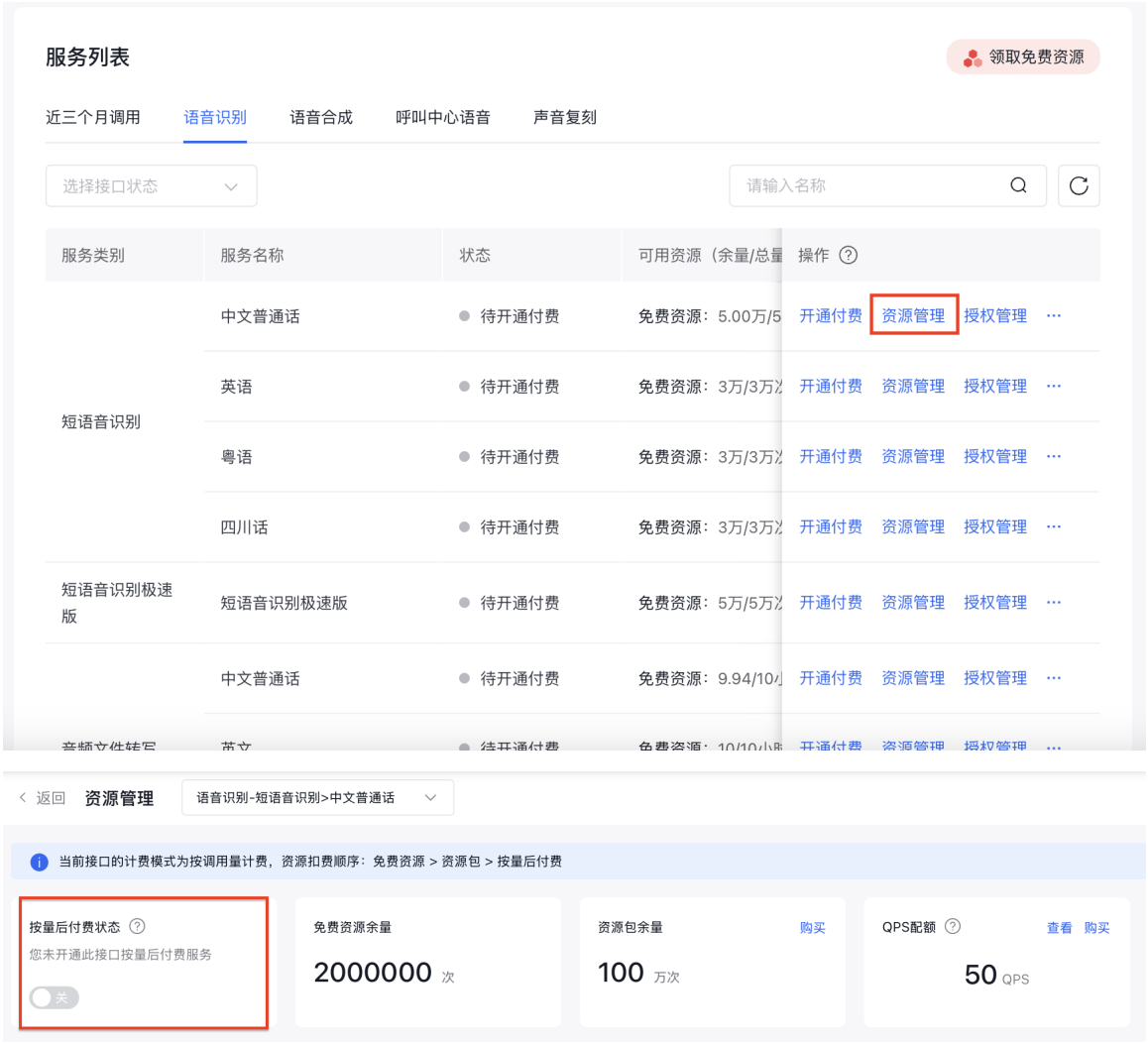
开通按量后付费

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



或点击“资源管理”，打开“按量后付费状态”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”，在购买页里选择“按量后付费”，后在"接口名称"处批量勾选接口。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。

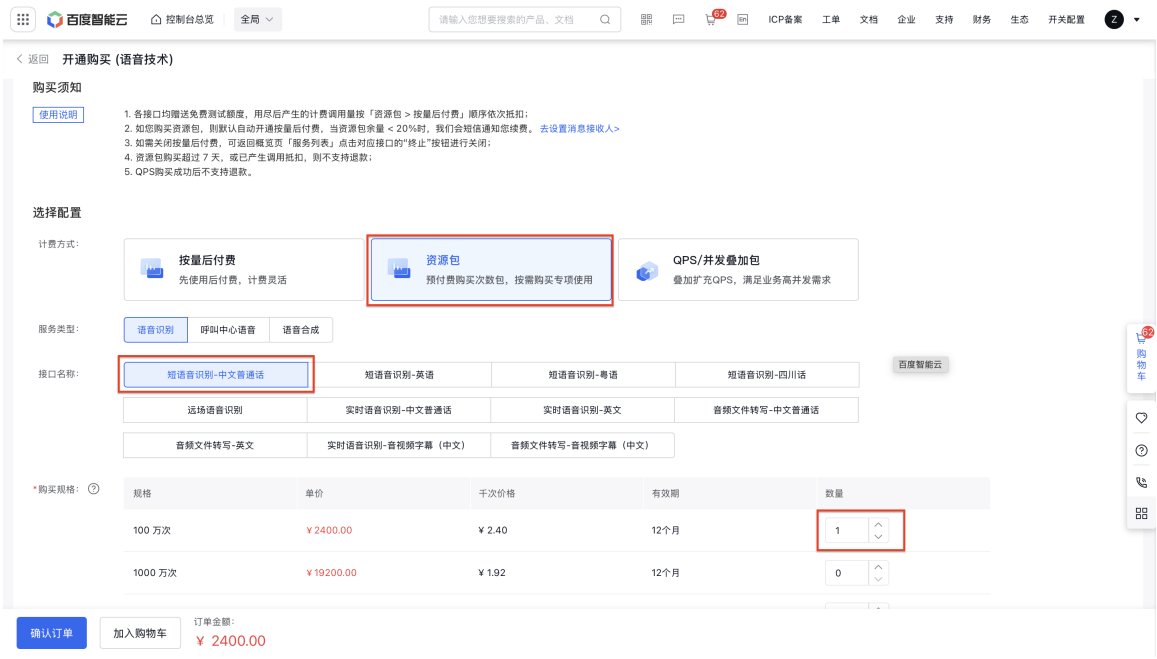
- **购买预付费资源包**

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。

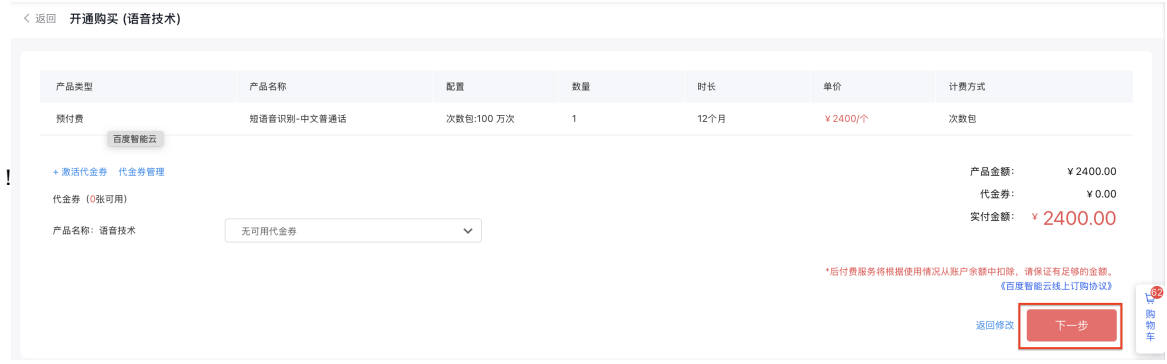




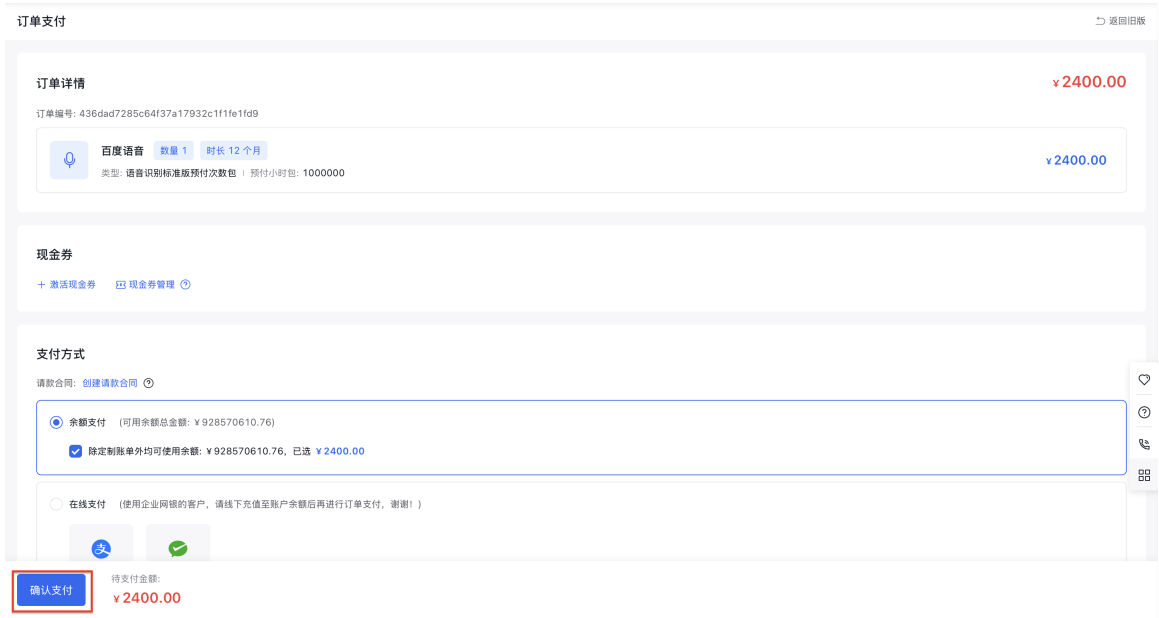
2. 选择所需资源包类别及数量，点击“立即购买”。



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。

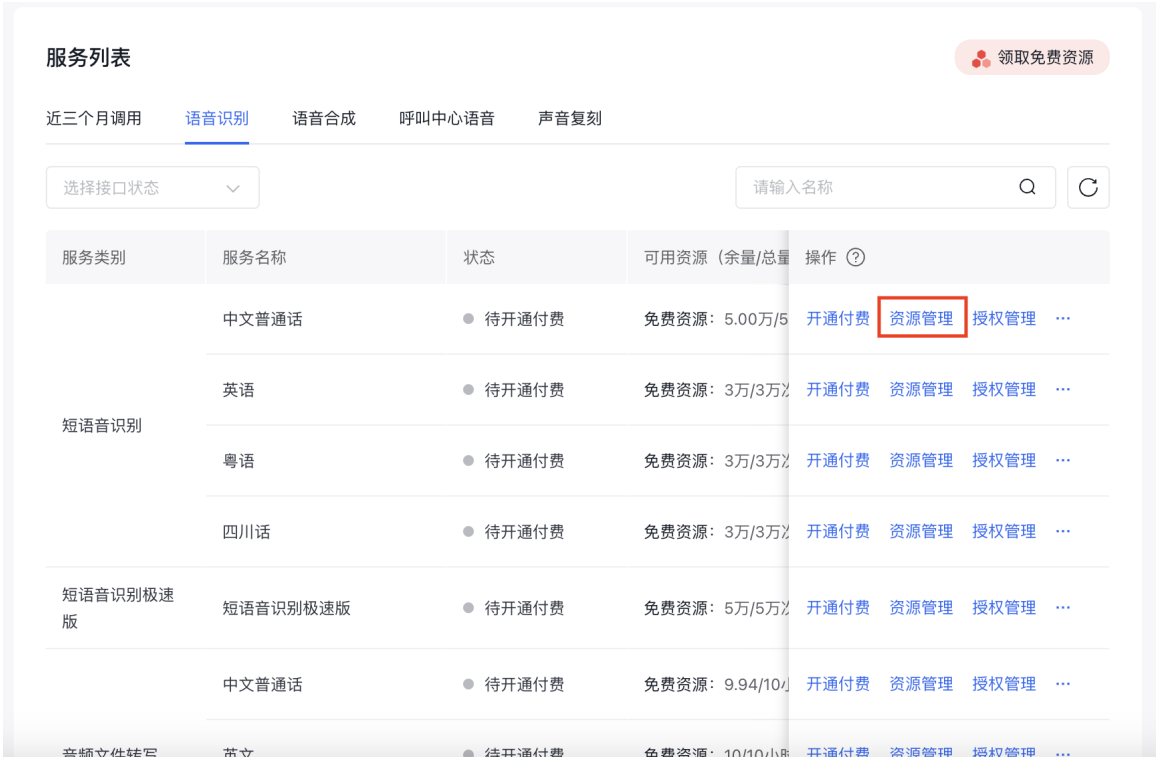


5. 成功购买专项资源包。



- **管理预付费资源包**

1. 进入语音技术控制台后，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

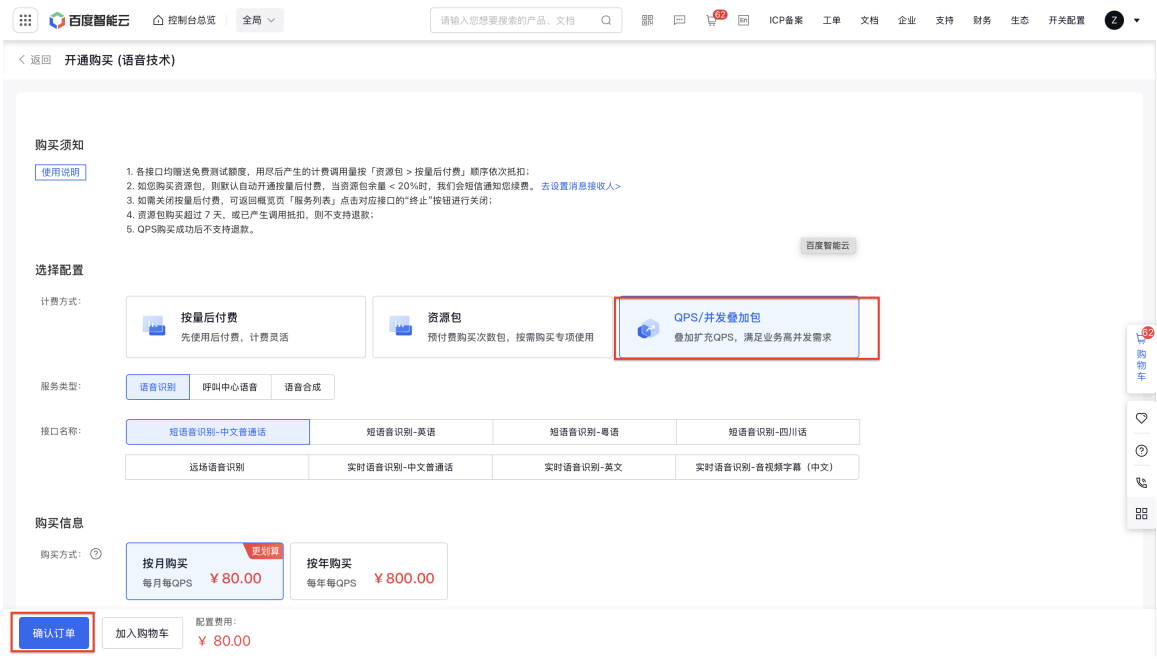
当现有QPS配额不能满足您的业务需求时，您可以购买QPS叠加包，增加QPS上限。

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。

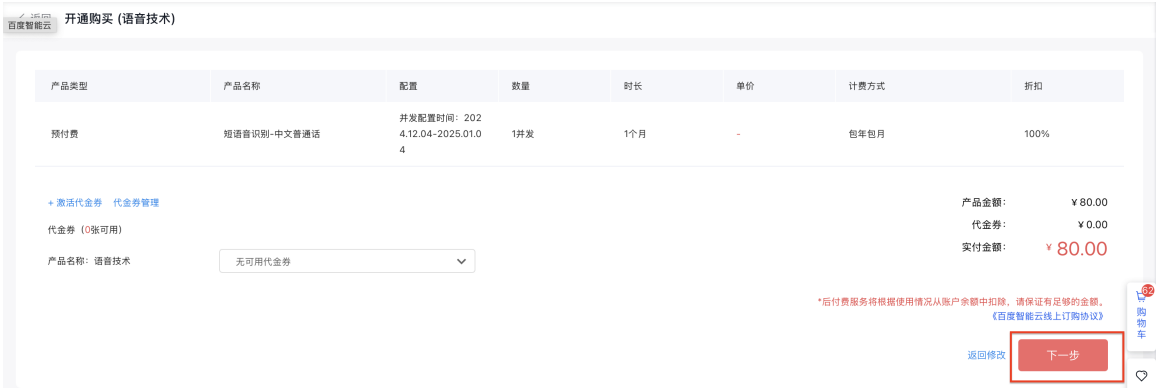




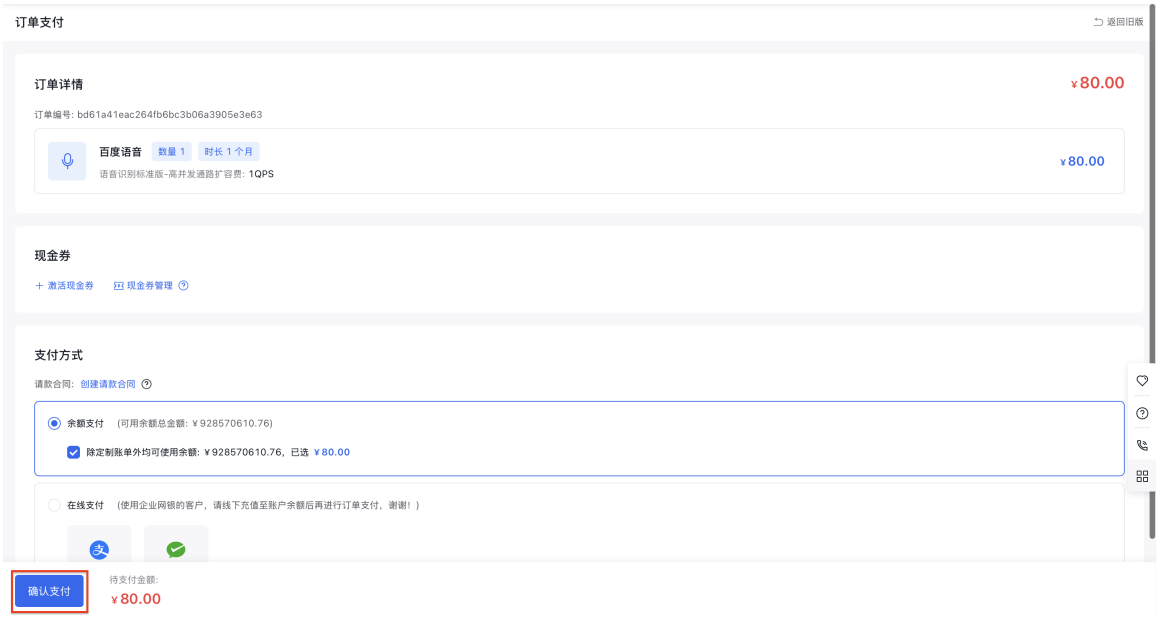
2. 计费方式选择"QPS叠加包"，选择要购买叠加包的接口和购买方式后，点击“确认订单”



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。

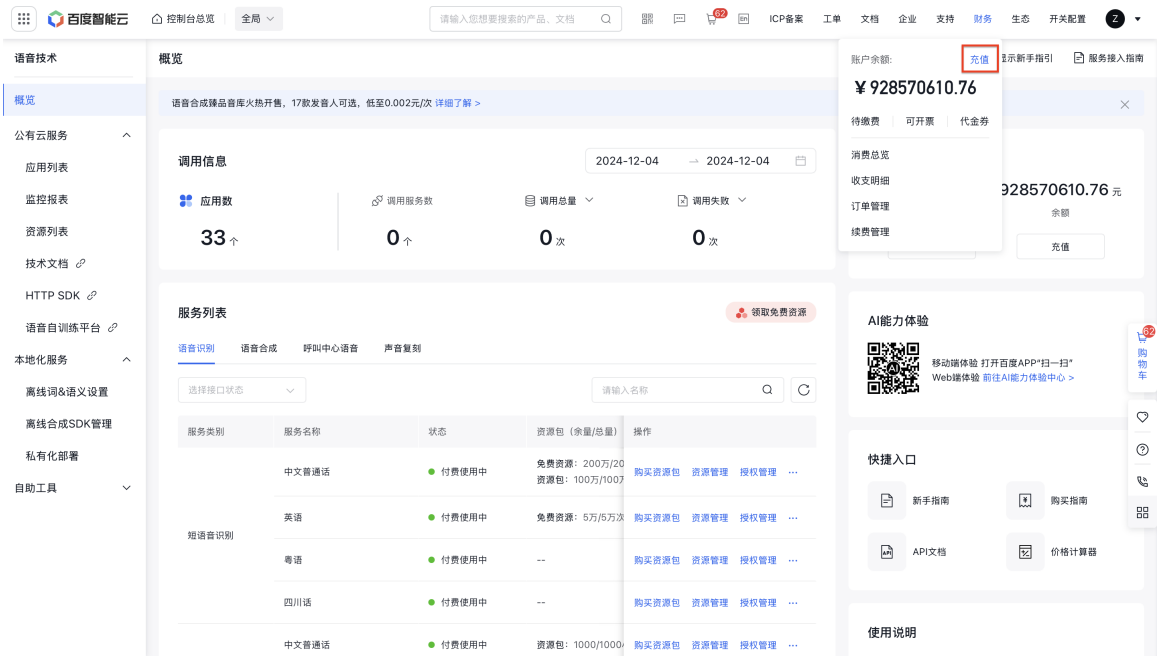


5. 购买QPS后，您可以在资源管理页中查看对应服务的QPS配额。

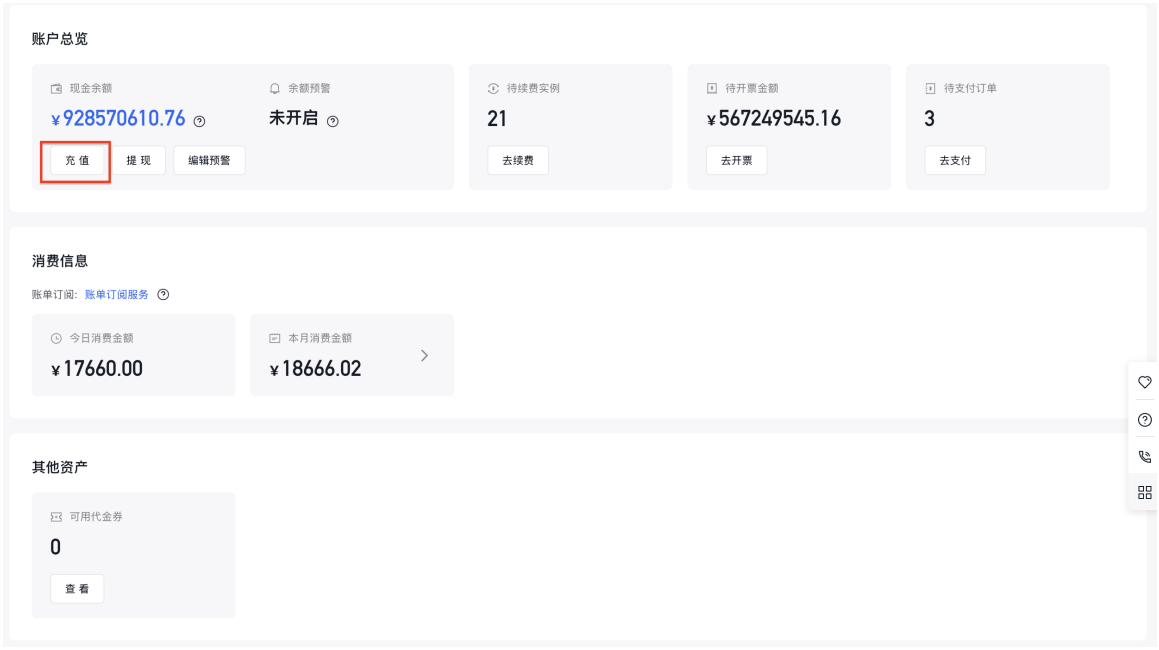


账户充值

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。

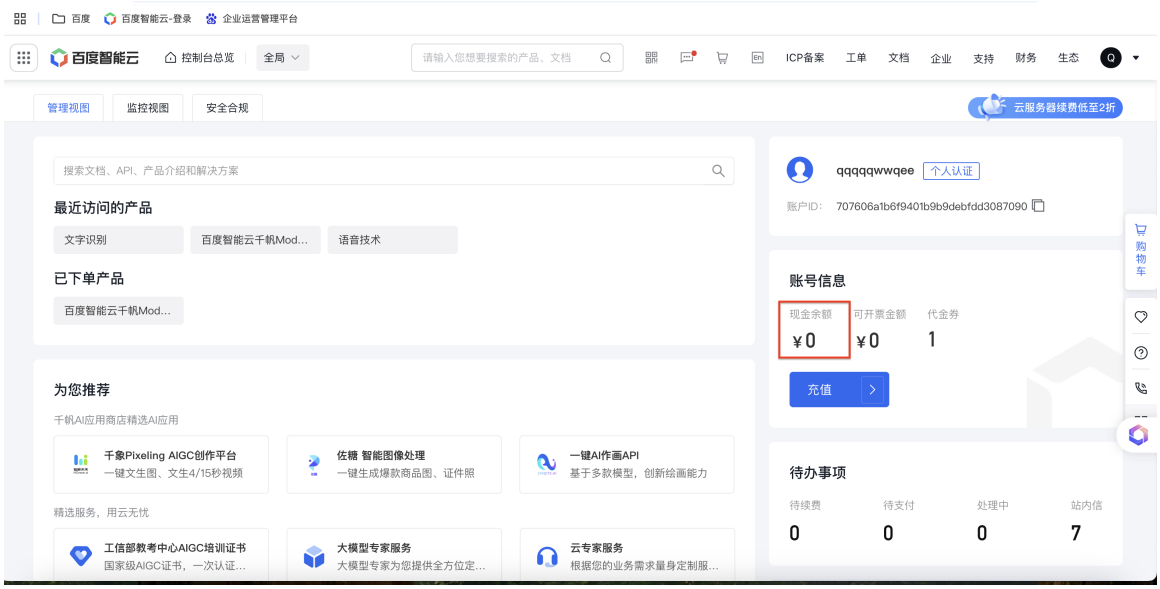
> 1.为保证汇款顺利进行，请务必保证汇款方名称与实名认证名称一致。
>
> 2.汇款方名称将是您后期开增值税专用发票的名称。

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。

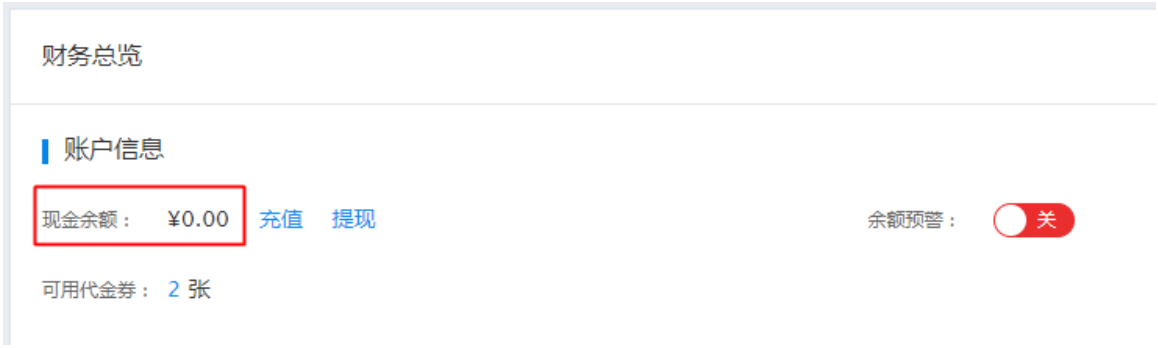
查看余额

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。

大模型语音

语音识别大模型

实时语音识别大模型

实时语音识别大模型

接口描述

实时语音识别大模型采用 WebSocket 协议，支持音频流实时上传与识别结果同步返回，实现边说边识别、边识别边返回文字。依托新一代语音识别大模型，提升多语种、方言、不同口音及复杂噪声环境下的识别准确率，兼顾识别精度与响应速度。

接口支持实时音频流输入，也可通过分帧发送本地音频文件进行识别，返回中间识别结果、最终识别文本及句级时间戳，适用于语音输入、实时字幕、直播质检、会议记录等场景。**(本接口处于邀测阶段，请提交**合作咨询**申请测试)**

****主要流程****

1. **建立连接：** 客户端通过 WebSocket 地址发起连接请求，完成鉴权及协议握手。
2. **发送开始帧：** 连接建立后，发送 `START` 文本帧，指定识别模型、音频格式和采样率等参数。
3. **发送音频数据：** 将音频划分为多个二进制帧，按照实时音频采集节奏持续上传。
4. **接收识别结果：** 服务端通过 `MID_TEXT` 返回中间识别结果，通过 `FIN_TEXT` 返回单句最终识别结果及时间信息。客户端可在发送音频的同时持续接收结果。
5. **结束识别：** 音频发送完成后，客户端发送 `FINISH` 文本帧，等待服务端返回剩余识别结果。
6. **关闭连接：** 服务端完成识别后主动关闭 WebSocket 连接，客户端根据连接关闭事件完成资源释放。

如需提前终止识别，客户端可发送 `CANCEL` 文本帧，取消本次识别任务。



请求说

明

建立 WebSocket 连接

请求地址：wss://aip.baidubce.com/ws/2.0/speech/publiccloudspeech/v1/asr

鉴权方式： 支持 Access Token 和 API Key 两种鉴权方式，任选其一即可。

方式一：Access Token 鉴权

通过 URL Query 参数传入 Access Token：`wss://aip.baidubce.com/ws/2.0/speech/publiccloudspeech/v1/asr?sn={sn}&access_token={access_token}`

方式二：API Key 鉴权

通过 WebSocket 握手请求的 Header 传入：`Authorization: Bearer {api_key}`

连接参数：

参数名	类型	是否必填	描述
sn	string	否	请求唯一标识，建议使用 UUID，便于日志排查
access_token	string	与 API Key 二选一	Access Token 鉴权凭证

`sn` 支持英文字母、数字及连字符 `-`，长度不超过 128 个字符。

注意：Access Token 与 API Key 两种鉴权方式任选其一，不需要同时传入。请避免在应用日志中输出完整鉴权凭证。

发送开始参数帧（START）

WebSocket 连接建立成功后，客户端首先发送 `START` 文本帧，初始化语音识别会话。注意帧的类型（Opcode）是**Text**， 使用json序列化

示例：

```
{
  "type": "START",
  "data": {
     # 下面是固定参数
     "model": "qianfan-asr-v1", # 识别模型
     "format": "pcm",
     "sample": 16000
  }
}
```

具体参数说明

参数名称	类型	是否必填	说明
type	string	必填	START，开始帧的类型
data	object	必填	具体见下表

data参数说明

参数名称	类型	是否必填	说明
model	string	必填	识别模型名称，当前支持 `qianfan-asr-v1`
format	string	必填	音频格式，固定为 `pcm`
sample	int	必填	音频采样率，支持 `8000` 或 `16000` Hz，推荐 `16000` Hz

> 实时语音识别大模型支持 8 kHz 或 16 kHz、16-bit、单声道 PCM 音频，采用小端字节序（s16le），每个采样点占用 2 字节。请求参数 `format` 必须设置为 `pcm`，推荐使用 16 kHz 采样率，以获得更好的识别效果。

发送音频数据帧

注意帧的类型（Opcode）是**Binary**

内容是二进制的音频内容。 除最后一个音频数据帧， 每个帧的音频数据长度为20-200ms。

对于离线文件，建议按实时速度发送音频，即发送一个特定时长的音频帧之后，下一个音频帧需要间隔相应的时长后再发送。

发送结束帧

注意帧的类型（Opcode）是**Text**， 使用json序列化

示例：

```
 {
   "type":"FINISH"
}
```

具体参数说明

参数名称	类型	是否必填	说明
type	String	必填，固定值	结束帧的类型

发送取消帧

取消与结束不同，结束表示音频正常结束，取消表示不再需要识别结果，服务端会迅速关闭连接 示例：

```
 {
   "type":"CANCEL"
}
```

具体参数说明

参数名称	类型	是否必填	说明
type	String	必填，固定值	CANCEL 立即取消本次识别

返回说明

接收数据帧

注意需要接收的帧类型（Opcode）是**Text**， 本接口不会返回Binary类型的帧。 text的内容， 使用json序列化

临时及最终识别结果

一段音频由多句话组成，实时识别api会依次返回每句话的临时识别结果和最终识别结果

一句话的临时识别结果示例：

```
 {
   "err_no":0,
   "err_msg":"OK",
   "type": "MID_TEXT",
   "result": "北京天气怎",
   "log_id": 45677785,
   "sn": "399427ce-e999-11e9-94c8-fa163e4e6064_ws_2"
}
```

一句话的最终识别结果：

```
{
  "type":"FIN_TEXT",
  "result":"北京天气怎么样",
  "start_time":53220,
  "end_time":73340,
  "err_no":0,
  "err_msg":"OK",
  "log_id":45677785,
  "sn":"399427ce-e999-11e9-94c8-fa163e4e6064_ws_2"
}
```

服务端报错如-3005错误码，是针对的是一个句子的，其它句子依旧可以识别，请求是否结束以服务端是否关闭连接为准。 具体错误码含义见“错误码“

```
{
  "err_msg": "asr authentication failed",
  "err_no": -3004,
  "log_id": 106549729,
  "result": "",
  "sn": "0d25b5e1-ffa6-11e9-8968-f496349f3917",
  "type": "FIN_TEXT"
}
```

一句话的开始时间及结束时间：识别过程中，百度服务端在每句话的最终识别结果中带有这句话的开始和结束时间。最终识别结果是指"type":"FIN_TEXT"，即一句话的最后识别结果，包括这句话的报错结果。

通常一个音频会得到如下的时间信息：

```
  # s1 e1 指第一句话的开始和结束时间start_time，end_time
[静音][s1:e1][静音][s2:e2]....[静音][s10:e10][静音]
建议一句话不要超过30s，否则识别准确性会有影响或会被强制切割
```

参数说明

参数名称	名称	类型	说明
err_no	错误码	int	0 表示正确， 其它错误码见文末
err_msg	错误信息	string	err_no!=0时，具体的报错解释
type	结果类型	string	结果类型：`MID_TEXT` 为中间结果，`FIN_TEXT` 为单句最终结果或错误结果
~	~	MID_TEXT	一句话以及临时识别结果
~	~	FIN_TEXT	一句话的最终识别结果或者报错， 是否报错由err_no判断
result	识别结果	string	音频的识别结果
start_time	一句话的开始时间	int，毫秒	一句话的开始时间，临时识别结果MID_TEXT 无此字段
end_time	一句话的结束时间	int，毫秒	一句话的结束时间，临时识别结果MID_TEXT 无此字段
log_id	日志id	string	日志id，用这个id可以百度服务端定位请求，排查问题
sn	请求sn	string	用这个sn可以百度服务端定位请求，排查问题。ws URI里的参数及识别句子的组合

示例代码

```
###### -*- coding: utf-8 -*-
"""
百度实时语音识别（Realtime ASR）WebSocket Demo

调用流程：
```

```
    1. 建立连接          ws_app.run_forever()
    2. 连接成功回调       on_open()
      2.1 发送开始帧      send_start()  —— 告知音频格式、采样率、模型
      2.2 循环发送音频    分片读取 PCM 并按实时节奏发送
      2.3 接收识别结果    on_message()  —— 服务端返回中间/最终结果
      2.4 发送结束帧      send_finish()  —— 通知服务端音频发送完毕
    3. 连接关闭回调       on_close()
    库层面报错回调         on_error()    —— 如连接超时、网络中断

小提示：
    运行前加环境变量可让日志实时刷新，方便定位问题：
        PYTHONUNBUFFERED=1 python ws_demo.py
"""
import argparse
import json
import time
import uuid
import logging
import threading

import websocket

###### ============================== 可配置参数 ==============================
###### 音频参数：必须与实际 PCM 文件保持一致，否则识别结果会异常
SAMPLE_RATE = 16000       # 采样率(Hz)：8k 电话音质 / 16k 近场音质
CHUNK_MS = 160            # 每帧音频时长(ms)：模拟实时流的发送粒度
MODEL = "qianfan-asr-v1"          # 模型 ID（语言/场景），按业务需求选择

###### 待识别的音频文件：16bit 单声道 PCM，采样率需等于 SAMPLE_RATE
AUDIO_FILE = "xi.pcm"

###### 百度公有云 ASR 接入地址（仅保留线上域名，走 wss 加密连接）
ADDR = "aip.baidubce.com"
PATH = "/ws/2.0/speech/publiccloudspeech/v1/asr"

###### ===== 鉴权方式（二选一，也可用命令行 --auth 指定）=====
###### 方式一：access_token（URL 参数，默认）
ACCESS_TOKEN = "your_access_token"
###### 方式二：API_KEY 放请求 header（需配合 URL 上的 client_id=<API_KEY>）
API_KEY = "your_api_key"                        # 控制台获取的 API Key
API_KEY_HEADER = "Authorization"  # API Key 所在 header 名称，按服务端要求调整
AUTH_MODE = "apikey"                    # 默认 token；可改 "apikey"，或用命令行 --auth 指定
###### ========================================================================


def send_start(ws):
    """发送开始参数帧：声明音频格式、采样率与模型，服务端据此初始化识别会话。"""
    req = {
        "type": "START",
        "data": {
            "model": MODEL,
            "format": "pcm",
            "sample": SAMPLE_RATE,
        },
    }
    body = json.dumps(req)
    ws.send(body, websocket.ABNF.OPCODE_TEXT)
    print("send START frame:", body)


def send_finish(ws):
    """发送结束帧：告知服务端音频已发送完毕，等待返回最终结果后关闭连接。"""
```

```
    req = {"type": "FINISH"}
    body = json.dumps(req)
    ws.send(body, websocket.ABNF.OPCODE_TEXT)
    print("send FINISH frame:", body)


def on_open(ws):
    """连接建立成功后触发：另起线程发送音频，避免阻塞库的收包循环。"""

    def run():
        print("Start sending audio frames...")
        send_start(ws)

        # 每帧字节数 = 采样率 * 2字节(16bit) * 帧时长(秒)
        chunk_bytes = int(SAMPLE_RATE * 2 * CHUNK_MS / 1000)
        try:
            with open(AUDIO_FILE, "rb") as f:
                while True:
                    audio_data = f.read(chunk_bytes)
                    if not audio_data:      # 读到文件末尾，音频发送完毕
                        break
                    ws.send(audio_data, opcode=websocket.ABNF.OPCODE_BINARY)
                    # 按帧时长 sleep，模拟麦克风实时采集的节奏
                    time.sleep(CHUNK_MS / 1000)
        except Exception as e:
            logging.error(f"读取/发送音频出错: {e}")

        send_finish(ws)

    threading.Thread(target=run, daemon=True).start()


def on_message(ws, message):
    """接收服务端返回的识别结果（JSON 文本帧），业务侧自行解析所需字段。"""
    if isinstance(message, str):
        msg = json.loads(message)
        print("Received JSON:", msg, time.time())


def on_error(ws, error):
    """库层面报错回调：如连接超时、网络中断等。"""
    print("error:", error)


def on_close(ws, close_status_code, close_msg):
    """连接关闭回调：close_status_code 为 WebSocket 关闭码，正常关闭为 1000。"""
    print(f"Websocket closed, status_code: {close_status_code}, server message: {close_msg}")


if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="百度实时语音识别 WebSocket Demo")
    parser.add_argument(
        "--auth", "-a", choices=["token", "apikey"], default=AUTH_MODE,
        help="鉴权方式: token(access_token, 默认) 或 apikey(API Key 放 header)",
    )
    parser.add_argument("--audio", default=AUDIO_FILE, help="音频文件路径(16k PCM)")
    args = parser.parse_args()

    auth_mode = args.auth
    AUDIO_FILE = args.audio
    if auth_mode == "apikey" and not API_KEY:
        print("[WARN] 使用 apikey 鉴权但脚本顶部 API_KEY 为空，服务端会返回鉴权失败")
```

```
    # sn：本次会话的唯一标识，服务端日志可按 sn 定位对应会话
    sn = uuid.uuid4()
    if auth_mode == "apikey":
        # API Key 方式：URL 带 client_id=<API_KEY>，API Key 放 header
        uri = f"wss://{ADDR}{PATH}?sn={sn}"
        headers = [f"{API_KEY_HEADER}: {API_KEY}"]
    else:
        # access_token 方式：token 放 URL 参数
        uri = f"wss://{ADDR}{PATH}?sn={sn}&access_token={ACCESS_TOKEN}"
        headers = []
    print("uri is " + uri)
    print(f"鉴权方式: {auth_mode}" + (f", header: {headers}" if headers else ""))

    ws_app = websocket.WebSocketApp(
        uri,
        header=headers,
        on_open=on_open,       # 连接建立后的回调
        on_message=on_message, # 接收消息的回调
        on_error=on_error,     # 库遇见错误的回调
        on_close=on_close,     # 关闭后的回调
    )
    try:
        # ping_timeout：心跳超时(秒)，超时未收到 pong 则判定连接异常
        ws_app.run_forever(ping_timeout=6)
    except websocket._exceptions.WebSocketTimeoutException:
        print("Connection timeout!")
```

附：错误码详情

报错示例：

```
 {
    "err_msg": "asr authentication failed",
    "err_no": -3004,
    "log_id": 106549729,
    "result": "",
    "sn": "0d25b5e1-ffa6-11e9-8968-f496349f3917",
    "type": "FIN_TEXT"
}
```

err_no = 0 是正确，非0值即为报错。 在err_msg中一般有更具体的报错信息。

错误码 err_no	错误方	原因	解决方法	是否会识别并关闭连接
-3003	百度服务端问题	服务后端异常	记录报错json并反馈	这一句话报错，不影响后续识别
-3004	调用方问题	鉴权失败，没有权限调用接口	查看appid appkey devpid这3个参数有无填对，QPS有无超限	是
-3005	调用方问题	音频质量问题，无效音频	根据start_time和end_time可以听下是否是噪音	这一句话报错，不影响后续识别
-3006	百度服务端问题	服务后端VAD异常	记录报错json并反馈	是
-3008	调用方问题	开始参数帧参数错误，包括漏写	检测参数有无缺失及格式正确，在err_msg中有更具体的报错信息	是
-3011	用户网络问题	上行流建立失败，可能跟网络状况或后端有关	重试	是
-3012	用户网络问题	下行流建立失败，可能跟网络状况或后端有关	重试	是
-3013	百度服务端问题	服务后端异常	记录报错json并反馈	这一句话报错，不影响后续识别
-3014	调用方正常调用CANCEL	用户取消	如果发送取消帧后会有此返回，可以忽略	是
-3101	调用方问题	一段时间没发出音频数据，导致整体请求超时	确认代码有音频数据一直在发送中	是

一句话识别大模型

接口描述

一句话识别大模型支持将短语音内容快速、准确地转换为文字，通过 HTTP 接口同步返回识别结果。依托新一代语音识别大模型，提升多语种、方言、不同口音及复杂噪声环境下的识别准确率，兼顾识别精度与响应速度，适用于语音输入、语音搜索、语音指令、智能硬件交互等短语音应用场景。

(本接口处于邀测阶段，请提交合作咨询申请测试)

调用流程

1. **前置准备：** 注册百度智能云账号并完成实名认证，获取接口鉴权凭证。支持 Access Token 和 API Key 两种鉴权方式，任选其一即可，详见鉴权认证。
2. **创建识别请求：** 将音频文件进行 Base64 编码，填入 JSON 请求体的 `speech` 字段，并配置识别模型、音频格式、采样率等参数。
3. **发送识别请求：** 通过 HTTP POST 方式调用一句话识别大模型接口，提交音频数据及相关参数。
4. **获取识别结果：** 接口处理完成后，同步返回 JSON 格式的识别结果。识别成功时，可通过 `result` 字段获取识别文本。

请求说明

请求地址：

`https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/server_api`

HTTP 方法： `POST`

接口仅支持通过 JSON 格式上传 Base64 编码后的音频数据，同步返回识别结果。

鉴权说明

当前接口支持 **API Key** 和 **Access Token** 两种鉴权方式，任选其一即可，无需同时填写。

请求头说明

- `Content-Type: application/json`
- `Authorization: Bearer <API_KEY>`（使用 API Key 鉴权时填写）

使用 Access Token 鉴权时，通过请求体中的 `token` 字段传入鉴权信息。Access Token 可通过 API Key 和 Secret Key 获取，详见鉴权认证。

音频格式要求

一句话识别大模型支持 PCM、WAV、AMR、M4A 四种音频格式，推荐使用 **16 kHz、16-bit、单声道 PCM 音频**，以获得更高效的识别体验。

注意事项：

- 推荐使用 PCM 格式，以减少额外的音频解码及格式转换耗时。使用 WAV、AMR、M4A 格式时，可能产生额外处理耗时。
- 音频文件的实际采样率应与请求参数 `rate` 保持一致。
- 上传音频需先进行 Base64 编码，并通过 JSON 请求体中的 `speech` 字段传入。

音频文件样例：

- 16k 采样率 pcm 文件样例下载
- 16k 采样率 wav 文件样例下载
- 16k 采样率 amr 文件样例下载
- 16k 采样率 m4a 文件样例下载

m4a 格式说明

接口支持符合以下编码及容器要求的 M4A 音频文件。

1. **声道：** 仅支持单声道。
2. **采样率：** 支持 16000 Hz、8000 Hz。
3. **编码码率：** 支持 CBR 24～96 kbps，推荐 48 kbps。
4. **编码格式：** 仅支持 AAC-LC，不支持 HE-AAC、AAC-LD、AAC-ELD 等其他 AAC 编码规格。
5. **容器要求：** 仅支持 `major_brand=mp42`、`minor_version=0` 的 MP4 容器，不支持 `major_brand=M4A` 的容器格式。

JSON方式上传音频

语音数据和其他参数通过标准 JSON 格式串行化 POST 上传， JSON 里包括的参数：

字段名	类型	是否必填	描述
format	string	必填	语音文件的格式，pcm/wav/amr/m4a。不区分大小写。推荐pcm文件
rate	int	必填	采样率，16000、8000，固定值
channel	int	必填	声道数，仅支持单声道，请填写固定值 1
cuid	string	必填	用户唯一标识，用来区分用户，计算UV值。建议填写能区分用户的机器 MAC 地址或 IMEI 码，长度为60字符以内
token	string	和API Key二选一	access_token鉴权信息，获取方式见鉴权认证
model	str	必填	识别模型名称，当前支持 `qianfan-asr-v1`
speech	string	必填	Base64 编码后的音频数据。需与 `format`、`rate`、`len` 参数保持一致
len	int	必填	本地语音文件的的字节数，单位字节

上传示例

JSON 格式 POST 上传本地文件

固定头部 header

```
Content-Type:application/json
```

请求示例

```
POST https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/server_api
```

speech 参数填写为 文件内容 base64 后的结果：

```
{
 "format": "pcm",
 "rate": 16000,
 "channel": 1,
 "cuid": "baidu_workshop",
 "token": "YOUR_ACCESS_TOKEN",
 "model": "qianfan-asr-v1",
 "speech": "BASE64_ENCODED_AUDIO",
 "len": 4096
}
```

返回示例

```
{
 "err_no": 0,
 "err_msg": "success.",
 "corpus_no": "15984125203285346378",
 "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D",
 "result": [
  "北京天气"
 ]
}
```

注意事项

len 字段表示原始语音大小字节数，不是 base64 编码之后的长度。

返回说明

接口采用 JSON 格式返回识别结果，识别文本统一采用 UTF-8 编码。识别成功时，可通过 result 字段获取识别文本。

字段名	数据类型	可需	描述
err_no	int	必填	业务错误码，0 表示识别成功
err_msg	string	必填	错误码描述
sn	string	必填	语音数据唯一标识，系统内部产生。如果反馈及 debug 请提供 sn
result	array ([string,string,…])	成功时返回	识别结果数组，返回1个最优候选结果。utf-8 编码

识别成功返回示例

```
{
 "err_no": 0,
 "err_msg": "success.",
 "corpus_no": "15984125203285346378",
 "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D",
 "result": ["北京天气"]
}
```

识别识别返回示例

```
{
 "err_no": 2000,
 "err_msg": "data empty.",
 "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D"
}
```

音频文件转写大模型

音频文件转写大模型

接口描述

音频文件转写大模型支持将长音频及大批量音频文件异步转写为文字，依托新一代语音识别大模型，提升多语种、方言、不同口音及复杂噪声环境下的识别准确率，适用于音视频字幕生产、批量录音质检、会议记录、录音内容分析等场景。

接口采用异步任务处理方式，用户提交音频文件 URL 创建转写任务后，可通过任务 ID 查询识别状态及转写结果，满足大规模音频文件处理需求。

(本接口处于邀测阶段，请提交合作咨询申请测试)

调用流程

1. **前置准备：** 注册百度智能云账号，完成实名认证，并获取接口鉴权凭证。
2. **创建转写任务：** 根据音频文件 URL、音频格式、模型名称及采样率等参数，调用创建任务接口，获取 task_id。
3. **查询转写结果：** 通过 task_ids 数组提交一个或多个任务 ID，批量查询任务状态及识别结果。
4. **获取转写文本：** 当任务状态为 Success 时，通过 task_result.result 获取识别文本；若任务失败，可根据返回的错误码及错误信息进行排查。

创建音频转写任务-请求说明

请求地址： https://aip.baidubce.com/rpc/2.0/aasr/v1/llm/voice/create

HTTP 方法： POST

鉴权说明

当前接口支持 **API Key** 和 **Access Token** 两种鉴权方式，任选其一即可，无需同时填写。

请求头说明

- `Content-Type: application/json`
- `Authorization: Bearer <API_KEY>`（使用 API Key 鉴权时填写）

使用 Access Token 鉴权时，通过请求体中的 `token` 字段传入鉴权信息。Access Token 可通过 API Key 和 Secret Key 获取，详见 Access Token 获取。

注意： 两种鉴权方式任选其一，无需同时填写。

请求参数

通过 HTTP POST 方式提交 JSON 格式请求，请求体包含以下参数：

参数名	类型	是否必需	描述	取值范围
speech_url	str	是	音频url	可使用百度云对象存储进行音频存储，生成云端可外网访问的url链接，音频大小不超过500MB，url长度不超过2048字节
format	str	是	音频格式	音频文件格式，支持 `mp3`、`wav`、`pcm`、`m4a`、`amr`。
model	str	是	模型名称	识别模型名称，当前支持 `qianfan-asr-v1`。
rate	int	是	采样率	音频采样率，目前仅支持 `16000` Hz。
smooth_text	int	否	文本顺滑	是否开启文本顺滑功能。`0`：关闭；`1`：开启。
filter_sensitive	int	否	敏感词过滤	是否开启敏感词过滤。`0`：关闭；`1`：开启。

在线调试&示例代码

Body请求示例：

```
{
 "speech_url": "https://platform.bj.bcebos.com/sdk%2Fasr%2Fasr_doc%2Fdoc_download_files%2F16k.pcm",
 "format": "pcm",
 "model": "qianfan-asr-v1",
 "rate": 16000
}
```

创建音频转写任务-返回说明

创建任务接口采用 JSON 格式返回结果。任务创建成功后返回 `task_id`，用于后续查询任务状态及识别结果。

返回参数

参数名	类型	是否必需	描述
log_id	int	是	请求日志 ID，用于问题排查。
task_id	str	否	任务唯一标识，创建成功时返回。
task_status	str	否	任务状态，创建成功时返回 `Created`。
error_code	int	否	请求失败时返回的错误码。
error_msg	str	否	请求失败时返回的错误信息。

Body返回示例

```
  # 创建成功
{
   "log_id": 12345678,
   "task_status": "Created" ,
   "task_id":  "234acb234acb234acb234acb"  #注意保存该id，用于后续请求识别结果
}
 # 创建失败，缺少参数
{
   "error_code": 336203,
   "error_msg": "missing param: speech_url",
   "log_id": 5414433131138366128
}
```

注意： 创建成功仅表示转写任务已被接收，并不代表识别完成。请保存返回的 `task_id`，用于后续查询任务状态与转写结果。

查询音频转写任务-请求说明

请求地址：

https://aip.baidubce.com/rpc/2.0/aasr/v1/llm/voice/query

HTTP 方法： `POST`

鉴权说明

与创建任务接口保持一致，请在请求 URL 中传入有效的 `access_token`。

请求参数

参数名	类型	是否必填	描述	取值范围
task_ids	array[string]	是	需要查询的任务 ID 数组	单次最多查询 200 个任务

请求示例

```
 {
  "task_ids": [
    "234acb234acb234acb234acb",
    "234acb234acb234acb234acd",
    "234acb234acb234acb234ace"
  ]
}
```

查询音频转写任务-返回说明

查询接口采用 JSON 格式返回结果，包含任务状态、识别文本及相关错误信息。

返回参数

参数名	类型	是否必需	描述
log_id	int	是	log id
tasks_info	list	否	多个任务的结果
+task_id	str	是	任务id
+task_status	str	是	任务状态
+task_result	dict	否	转写结果的json格式
++corpus_no	str	否	日志，反馈时使用
++result	str	否	转写结果
++audio_duration	int	否	音频时长（毫秒）
++detailed_result	list	否	分段转写详细结果
+++voice_power	int	否	音量参数
+++res	list	否	分段转写文本
+++speak_speed	int	否	语速参数
+++begin_time	int	否	分段开始时间
+++end_time	int	否	分段结束时间
+++words_info	list	否	字粒度（预留参数，暂不启用）
+++sn	str	否	分段日志，反馈时使用
+++corpus_no	str	否	分段日志，反馈时使用
++err_no	int	否	转写失败错误码
++err_msg	str	否	转写失败错误信息
error_code	int	否	请求错误码
error_msg	str	否	请求错误信息
error_info	list	否	错误的或查询不存在的taskid数组

返回示例

```
{
  "log_id": 12345678,
  "tasks_info": [
    { # 转写中
      "task_status": "Running"
      "task_id": "234acb234acb234acb234acb",
    },
    { # 转写失败
      "task_status": "Failure"
      "task_id": "234acb234acb234acb234acd",
      "task_result": {
        "err_no":  3301
        "err_msg": "speech quality error",
        "sn": "xxx"
      }
    },
    { # 转写成功
      "task_status": "Success",
      "task_result": {
        "result": [
          "观众朋友大家好，欢迎收看本期视频哦。毕竟..."
        ],
        "audio_duration": 6800,
        "detailed_result": [
          {
            "voice_power": "55.13867950439453",# 仅使用PID：80006时返回
            "res": [
              "观众朋友大家好，欢迎收看本期视频哦。"
            ],
            "speak_speed": "5.986260890960693",
            "end_time": 6700,
            "begin_time": 4240,
            "words_info": [],
            "sn": "257826606251573543780",
            "corpus_no": "6758319075297447880"
          }
          ...
        ],
        "corpus_no": "6758319075297447880"
      },
      "task_id": "234acb234acb234acb234ace"
    }
  ]
}
```

音频文件转写极速版大模型

接口描述

音频文件转写极速版大模型支持将音频、视频文件中的语音内容快速转换为文字，通过 HTTP 接口同步返回识别文本及对应时间戳。依托新一代语音识别大模型，提升多语种、方言及复杂语音场景下的识别准确率，兼顾转写质量与处理效率，适用于音视频字幕生成、录音转写、内容分析等对时效性要求较高的业务场景。

(本接口处于邀测阶段，请提交合作咨询申请测试)

请求说明

请求地址： `https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/asr/topspeed`

HTTP 方法： `POST`

鉴权说明

当前接口支持 **API Key** 和 **Access Token** 两种鉴权方式，任选其一即可，无需同时填写。

请求头说明

- `Content-Type: application/json`
- `Authorization: Bearer <API_KEY>`（使用 API Key 鉴权时填写）

使用 Access Token 鉴权时，通过请求体中的 `token` 字段传入鉴权信息。Access Token 可通过 API Key 和 Secret Key 获取，详见 Access Token 获取。

请求参数说明

Body 参数如下：

参数名称	类型	是否必填	描述
token	string	条件必填	使用 Access Token 鉴权时填写，与请求头 `Authorization` 二选一。
rate	int	是	音频采样率，目前仅支持 `16000` Hz。
cuid	string	是	用户唯一标识，由开发者自行生成并保证唯一性。
model	string	是	识别模型名称，当前支持 `qianfan-asr-v1`。
speech	string	与 `speech_url` 二选一	音视频文件的 Base64 编码内容。支持 PCM、WAV、MP3、M4A、MP4、MOV 等常见格式。文件时长不超过 1 小时，大小不超过 500 MB。
speech_url	string	与 `speech` 二选一	可通过 百度云对象存储 BOS 等方式提供可公网访问的音视频文件 URL。支持格式、时长及大小限制与 `speech` 一致。
enable_detail	int	否	是否返回详细转写结果。 `0`：仅返回转写文本；`1`：返回详细信息（默认）。取值范围：`0`、`1`。
enable_subtitle	int	否	是否开启字幕功能，按照字幕规则返回识别结果及对应时间戳。 `0`：关闭（默认）；`1`：开启。
subtitle_punc	int	否	字幕结果是否保留标点符号。 `0`：过滤标点（默认）；`1`：保留标点。仅在开启字幕功能时生效。
smooth_text	int	否	是否开启文本顺滑功能，包括标点优化、数字格式优化和口语过滤。 `0`：关闭（默认）；`1`：开启。
smooth_text_param	list[int]	否	指定文本顺滑功能，仅当 `smooth_text=1` 时生效。 `1`：标点优化；`2`：数字格式优化；`3`：口语过滤。 支持组合配置，例如 `[1,2,3]` 开启全部功能。
filter_sensitive	int	否	是否开启敏感词过滤。`0`：关闭（默认）；`1`：开启。

Body请求示例：

```
{
 "token": "YOUR_ACCESS_TOKEN",
 "rate": 16000,
 "cuid": "baidu_asr",
 "model": "qianfan-asr-v1",
 "speech": "BASE64_ENCODED_AUDIO",
 "enable_subtitle": 1,
 "smooth_text": 1
}
```

返回说明

返回参数：

参数名称	类型	描述
error_code	int	错误码，0 表示请求成功，其他值表示请求异常。
error_message	string	请求结果或错误信息。
audio_duration	int	音频整体时长，单位毫秒。
result	array[string]	完整转写结果，通常通过 result[0] 获取完整识别文本。
detailed_result	array[object]	分句转写详情，包含每句话的识别文本、时间戳及相关标识。

detailed_result 字段

参数名称	类型	描述
res	array[string]	当前句的识别文本，通常通过 res[0] 获取。
begin_time	int	当前句开始时间，单位毫秒。
end_time	int	当前句结束时间，单位毫秒。
sn	string	分句转写标识，供问题反馈与排查使用。
corpus_no	string	整段转写标识，供问题反馈与排查使用。

返回示例：


```
{
  "error_code":0,
  "error_message":"请求成功",
  "audio_duration":1300,
  "result":["欢迎使用百度语音,"],
  "detailed_result":[
     {
     "begin_time":0,
     "corpus_no":"632690ff-40e8-4e59-9964-a381e4c64818_ws",
     "end_time":1300,
     "res":["欢迎使用百度语音,"],
     "sn":"632690ff-40e8-4e59-9964-a381e4c64818_ws_ws_0","words_info":[]
     },
  ……]
}
```


端到端语音语言大模型

端到端语音语言大模型（Pro、Lite）

接口描述

百度端到端语音语言大模型基于业内首创的Cross-Attention跨模态语音大模型，具备极速响应、拟人音色，实现真人级别语音对话交互。极致共情、超高双商，支持深度需求理解与复杂任务执行。广泛应用于实时语音交互的情感陪伴、社交娱乐以及知识问答等场景。请点击链接进入端到端语音大模型详情。

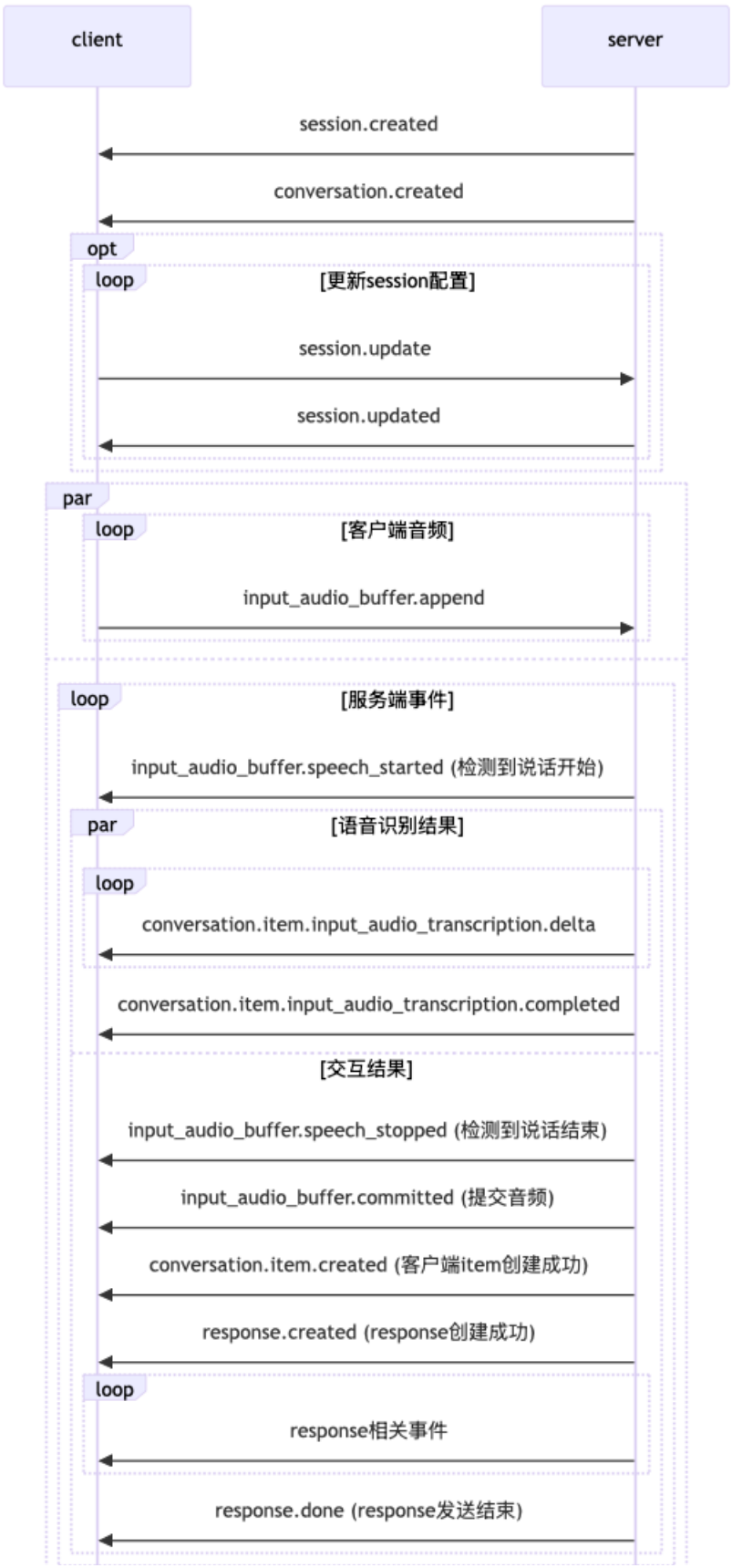
产品优势

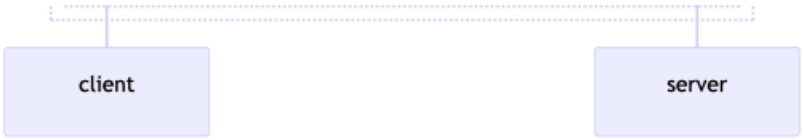
超低时延：基于业内创新的Cross-Attention技术，在对话过程中将用户等待时长从行业常见的3-5秒大幅缩短至1秒左右，实现了比拟真人对话的即时响应速度，树立行业标杆。
极致共情：基于真正的端到端跨模态语音大模型，能够感知原始语音携带的情绪与语气信息，充分理解用户意图与情境要求，更好地服务情感陪伴、 社交娱乐等场景。

超拟人音色：合成前端融入大语言模型，成就高自然度、高表现力的语音合成系统，使合成音频听感更加自然流畅，语气更加符合情境，情感更加接近真人，语调更加具有韵律。

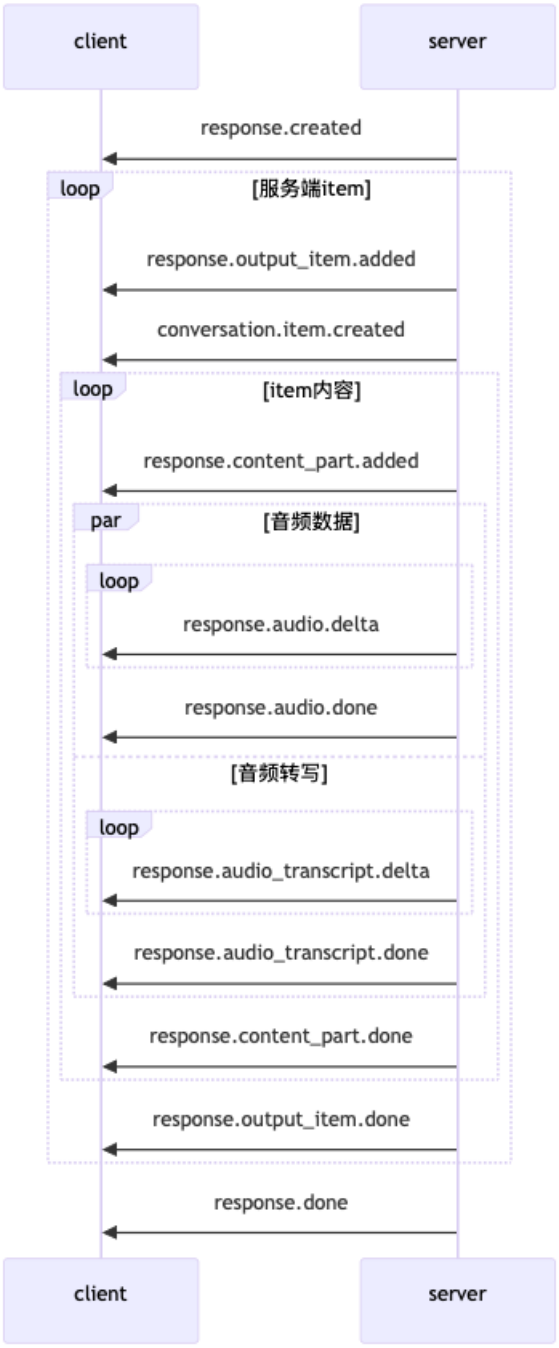
接口调用详情

交互流程





response事件交互



请求地址

请求地址：wss://aip.baidubce.com/ws/2.0/speech/v1/realtime

认证鉴权

支持 API Key 和 access_token 两种方式，具体请参考鉴权认证机制。

请求参数

URL中放置请求参数，参数如下：

参数名称	类型	是否必填	说明
model	string	必填	模型名称
示例：wss://aip.baidubce.com/ws/2.0/speech/v1/realtime?model=audio-realtime-far			

附：

模型	模型名称	适用场景
端到端语音语言大模型（Lite）	audio-mini-realtime-near	高性能、近场场景
端到端语音语言大模型（Lite）	audio-mini-realtime-far	高性能、远场场景
端到端语音语言大模型（Pro）	audio-realtime-near	高优效果、近场场景
端到端语音语言大模型（Pro）	audio-realtime-far	高优效果、远场场景

客户端事件

session.update

事件描述

客户端session.update事件用于更新会话的默认配置，服务端以session.updated包含完整有效配置的事件进行响应

事件参数

参数名称	类型	是否必填	说明
type	string	必填	事件类型，必须是session.update
event_id	string	可选	事件唯一标识
session	UpdateSession	必填	会话配置

示例

```
{
  "type": "session.update",
  "session": {
     "input_audio_transcription": {
        "model": "default"
     }
  }
}
```

input_audio_buffer.append

事件描述

客户端input_audio_buffer.append事件用于将音频字节附加到输入音频缓冲区

事件参数

参数名称	类型	是否必填	说明
type	string	必填	事件类型，必须是input_audio_buffer.append
event_id	string	可选	事件唯一标识
audio	string	必填	Base64 编码的音频字节，固定单声道、16000采样率

示例

```
{
  "type": "input_audio_buffer.append",
  "audio": "audio_base64"
}
```

服务端事件

session.created

事件描述

服务端session.created事件是建立新连接时的第一个服务器事件，此事件会使用默认会话配置创建并返回一个新会话

事件参数

参数名称	类型	说明
type	string	事件类型，必须是session.created
event_id	string	事件唯一标识
session	Session	会话配置

示例

```
{
  "type": "session.created",
  "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_Hjr7Au95",
  "session": {
      "id": "sess_ywqGIVMsrQKh8jY4WhYZ",
      "object": "realtime.session",
      "expires_at": 1752218581,
      "input_audio_format": "pcm16",
      "input_audio_noise_reduction": null,
      "input_audio_transcription": null,
      "instructions": "",
      "max_response_output_tokens": "inf",
      "modalities": [
          "text",
          "audio"
      ],
      "model": "audio-realtime",
      "output_audio_format": "pcm16",
      "speed": 1,
      "temperature": 0.8,
      "tool_choice": "auto",
      "tools": [],
      "tracing": null,
      "turn_detection": {
          "type": "server_vad",
          "threshold": 0.5,
          "prefix_padding_ms": 300,
          "silence_duration_ms": 200,
          "create_response": true,
          "interrupt_response": true
          },
      "voice": "default"
      }
}
```


session.updated

事件描述

服务端session.updated对客户端用于更新会话默认配置的session.update事件进行响应，响应事件包含完整有效配置

事件参数

参数名称	类型	说明
type	string	事件类型，必须是session.updated
event_id	string	事件唯一标识
session	Session	会话配置

示例

```
{
  "type": "session.updated",
  "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_hSIhy0aC",
  "session": {
    "id": "sess_ywqGIVMsrQKh8jY4WhYZ",
    "object": "realtime.session",
    "expires_at": 1752218581,
    "input_audio_format": "pcm16",
    "input_audio_noise_reduction": null,
    "input_audio_transcription": {
      "model": "default",
      "language": null,
      "prompt": null
    },
    "instructions": "",
    "max_response_output_tokens": "inf",
    "modalities": [
      "text",
      "audio"
    ],
    "model": "audio-realtime",
    "output_audio_format": "pcm16",
    "speed": 1,
    "temperature": 0.8,
    "tool_choice": "auto",
    "tools": [],
    "tracing": null,
    "turn_detection": {
      "type": "server_vad",
      "threshold": 0.5,
      "prefix_padding_ms": 300,
      "silence_duration_ms": 200,
      "create_response": true,
      "interrupt_response": true
    },
    "voice": "default"
  }
}
```


conversation.created

事件描述

会话创建后，立即返回服务端conversation.created事件

事件参数

参数名称	类型	说明
type	string	事件类型，必须是conversation.created
event_id	string	事件唯一标识
conversation	Conversation	会话资源

示例

```
{
  "type": "conversation.created",
  "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_bt89NXfx",
  "conversation": {
     "id": "conv_auVpdUi6cvWu5ANDjL25",
     "object": "realtime.conversation"
     }
}
```

conversation.item.created

事件描述

客户端发过来的音频已加入到对话中时，返回conversation.item.created服务端事件

事件参数

参数名称	类型	说明
type	string	事件类型，必须是conversation.item.created
event_id	string	事件唯一标识
previous_item_id	string	在对话中此项目之前的项目的 ID，创建的首个项目该值为null
item	ConversationItem	创建的消息

示例

```
{
  "type": "conversation.item.created",
  "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_CklwHSkg",
  "previous_item_id": null,
  "item": {
     "id": "item_ywqGIVMsrQKh8jY4WhYZ_001",
     "object": "realtime.item",
     "type": "message",
     "status": "completed",
     "role": "user",
     "content": [{
        "type": "input_audio",
        "transcript": "今天天气怎么样？"
        }]
     }
}
```

conversation.item.input_audio_transcription.delta

事件描述

输入音频对应的ASR识别结果

事件参数

参数名称	类型	说明
type	string	事件类型，必须是conversation.item.input_audio_transcription.delta
event_id	string	事件唯一标识
item_id	string	用户消息项目的 ID
content_index	integer	默认0
delta	string	识别文本

示例

```
 {
   "type": "conversation.item.input_audio_transcription.delta",
   "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_8o0XL7DD",
   "item_id": "item_ywqGIVMsrQKh8jY4WhYZ_001",
   "content_index": 0,
   "delta": "今"
}
```

conversation.item.input_audio_transcription.completed

事件描述

服务端conversation.item.input_audio_transcription.completed事件是将语音的音频转录写入音频缓冲区的结果

事件参数

参数名称	类型	说明
type	string	事件类型，必须是conversation.item.input_audio_transcription.completed
event_id	string	事件唯一标识
item_id	string	包含音频的用户消息项目的 ID
content_index	integer	包含音频的内容部分的索引
transcript	string	转录出的文本

示例

```
 {
   "type": "conversation.item.input_audio_transcription.completed",
   "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_sCk3x7cv",
   "item_id": "item_ywqGIVMsrQKh8jY4WhYZ_001",
   "content_index": 0,
   "transcript": "今天天气怎么样？"
}
```

conversation.item.input_audio_transcription.failed

事件描述

当配置了输入音频转录，并且用户消息的转录请求失败时，会返回服务器conversation.item.input_audio_transcription.failed事件。此事件与其他事件分开，error以便客户端可以识别相关项目

事件参数

参数名称	类型	说明
type	string	事件类型，必须是conversation.item.input_audio_transcription.failed
event_id	string	事件唯一标识
item_id	string	用户消息项目的ID
content_index	integer	包含音频的内容部分的索
error	Error	转录错误的详细信息。

示例

```
 {
   "type": "conversation.item.input_audio_transcription.failed",
   "event_id": "event_Ula21nRHDN0DDc4GT280_3f38VTV0",
   "item_id": "item_Ula21nRHDN0DDc4GT280_001",
   "content_index": 0,
   "error": {
      "type": "server_error",
      "code": "internal",
      "message": "error message"
   }
}
```

input_audio_buffer.committed

事件描述

当输入音频缓冲区提交时，返回服务端事件input_audio_buffer.committed

事件参数

参数名称	类型	说明
type	string	事件类型，必须为input_audio_buffer.committed
event_id	string	事件唯一标识
previous_item_id	string	在对话中此项目之前的项目的 ID，创建的首个项目该值为null
item_id	string	创建消息项目的ID

示例

```
 {
   "type": "input_audio_buffer.committed",
   "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_kyvsH2Ur",
   "previous_item_id": null,
   "item_id": "item_ywqGIVMsrQKh8jY4WhYZ_001"
}
```

input_audio_buffer.speech_started

事件描述

当在音频缓冲区中检测到语音时，在server_vad模式下返回服务端input_audio_buffer.speech_started事件

事件参数

参数名称	类型	说明
type	string	事件类型必须是input_audio_buffer.speech_started
event_id	string	事件唯一标识
item_id	string	服务端检测到客户端会话时，语音停止时会创建的用户消息项的ID

示例

```
 {
   "type": "input_audio_buffer.speech_started",
   "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_W3Zas9hP",
   "item_id": "item_ywqGIVMsrQKh8jY4WhYZ_001"
}
```

input_audio_buffer.speech_stopped

事件描述

当服务端检测到音频缓冲区中的语音结束时，返回input_audio_buffer.speech_stopped服务端事件

事件参数

参数名称	类型	说明
type	string	事件类型，必须是input_audio_buffer.speech_stopped
event_id	string	事件唯一标识
item_id	string	用户消息项目的 ID

示例

```
 {
   "type": "input_audio_buffer.speech_stopped",
   "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_s0ROBCxD",
   "item_id": "item_ywqGIVMsrQKh8jY4WhYZ_001"
}
```

response.created

事件描述

当初次响应被创建时，会返回服务端response.created事件。这是响应创建的第一个事件，响应的初始状态为in_progress

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.created
event_id	string	事件唯一标识
response	Response	响应对象

示例

```
{
  "type": "response.created",
  "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_nclqvTp",
  "response": {
     "id": "resp_ywqGIVMsrQKh8jY4WhYZ_001",
     "object": "realtime.response",
     "status": "in_progress",
     "status_details": {
        "type": "in_progress"
     },
     "output": [],
     "conversation_id": "conv_auVpdUi6cvWu5ANDjL25",
     "modalities": [
        "text",
        "audio"
     ],
     "voice": "default",
     "output_audio_format": "pcm16",
     "temperature": 0.8,
     "max_output_tokens": "inf"
     }
}
```

response.done

事件描述

当响应流式传输完成后，无论最终状态如何，会返回服务器事件response.done，事件中包含的响应对象包含响应中的所有输出项，但会省略原始音频数据

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.done
response	Response	响应对象

示例

```
{
  "type": "response.done",
  "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_07u3tDT3",
  "response": {
     "id": "resp_ywqGIVMsrQKh8jY4WhYZ_001",
     "object": "realtime.response",
     "status": "cancelled",
     "status_details": {
        "type": "cancelled",
        "reason": "turn_detected"
     },
     "output": [{
        "id": "item_ywqGIVMsrQKh8jY4WhYZ_002",
        "object": "realtime.item",
        "type": "message",
        "status": "incomplete",
        "role": "assistant",
        "content": [{
           "type": "audio",
           "transcript": "今天的天气呀，我其实不太清楚呢，因为这得看具体的地方呀。你可以告诉我你在哪里，或者你自己看看窗外的天气怎么样呀，对不对？"
        }]
     }],
     "conversation_id": "conv_auVpdUi6cvWu5ANDjL25",
     "modalities": [
        "text",
        "audio"
     ],
     "voice": "default",
     "output_audio_format": "pcm16",
     "temperature": 0.8,
  "max_output_tokens": "inf"
  }
}
```


response.output_item.added

事件描述

response.output_item.added在响应生成期间创建新项目消息

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.output_item.added
event_id	string	事件唯一标识
response_id	string	该项目所属的响应的 ID
output_index	integer	响应中输出项的索引
item	ConversationItem	已添加的项目

示例

```
{
  "type": "response.output_item.added",
  "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_yvrV5UAs",
  "response_id": "resp_ywqGIVMsrQKh8jY4WhYZ_001",
  "output_index": 0,
  "item": {
     "id": "item_ywqGIVMsrQKh8jY4WhYZ_002",
     "object": "realtime.item",
     "type": "message",
     "status": "in_progress",
     "role": "assistant",
     "content": []
     }
}
```

response.output_item.done

事件描述

当项目流式传输完成时或响应被中断、不完整或取消时，将返回此服务器事件response.output_item.done

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.output_item.added
event_id	string	事件唯一标识
response_id	string	该项目所属的响应的 ID
output_index	integer	响应中输出项的索引
item	ConversationItem	已添加的项目

示例

```
{
  "type": "response.output_item.done",
  "event_id": "event_ywqGIVMsrQKh8jY4WhYZ_TDVWUShW",
  "response_id": "resp_ywqGIVMsrQKh8jY4WhYZ_001",
  "output_index": 0,
  "item": {
     "id": "item_ywqGIVMsrQKh8jY4WhYZ_002",
     "object": "realtime.item",
     "type": "message",
     "status": "incomplete",
     "role": "assistant",
     "content": [{
        "type": "audio",
        "transcript": "今天的天气呀，我其实不太清楚呢，因为这得看具体的地方呀。你可以告诉我你在哪里，或者你自己看看窗外的天气怎么样呀，对不对？"
        }]
     }
}
```

response.content_part.added

事件描述

在响应生成期间将新的内容部分添加到助手消息项时，将返回服务器事件response.content_part.added

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.content_part.added
event_id	string	事件唯一标识
response_id	string	响应的 ID
item_id	string	添加了内容部分的消息项目的 ID
output_index	integer	响应中输出项的索引
content_index	integer	项目内容数组中内容部分的索引
part	ConversationItemContent	新增的内容部分

示例

```
  {
      "type": "response.content_part.added",
      "event_id": "event_Ula21nRHDN0DDc4GT280_wa1BMTuP",
      "response_id": "resp_Ula21nRHDN0DDc4GT280_001",
      "item_id": "item_Ula21nRHDN0DDc4GT280_002",
      "output_index": 0,
      "content_index": 0,
      "part": {
          "type": "audio",
          "transcript": ""
      }
}
```

response.content_part.done

事件描述

在响应生成期间将内容部分添加到助手消息项完成时，将返回服务器事件response.content_part.done

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.content_part.done
event_id	string	事件唯一标识
response_id	string	响应的 ID
item_id	string	添加了内容部分的消息项目的 ID
output_index	integer	响应中输出项的索引
content_index	integer	项目内容数组中内容部分的索引
part	ConversationItemContent	内容部分

示例

```
 {
   "type": "response.content_part.done",
   "event_id": "event_Ula21nRHDN0DDc4GT280_L6W3WsIV",
   "response_id": "resp_Ula21nRHDN0DDc4GT280_001",
   "item_id": "item_Ula21nRHDN0DDc4GT280_002",
   "output_index": 0,
   "content_index": 0,
   "part": {
      "type": "audio",
      "transcript": "当然会呀！一闪一闪亮晶晶，满天都是小星星，挂在天上放光明，好像许多小眼睛！要不要我再唱一段给你听呀？"
   }
}
```


response.audio.delta

事件描述

在响应生成期间音频内容发生变化时，将返回服务器事件response.audio.delta

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.audio.delta
event_id	string	事件唯一标识
response_id	string	响应的 ID
item_id	string	添加了内容部分的消息项目的 ID
output_index	integer	响应中输出项的索引
content_index	integer	项目内容数组中内容部分的索引
delta	string	音频内容的base64编码

示例


```
 {
   "type": "response.audio.delta",
   "event_id": "event_Ula21nRHDN0DDc4GT280_yZemLoGb",
   "response_id": "resp_Ula21nRHDN0DDc4GT280_001",
   "item_id": "item_Ula21nRHDN0DDc4GT280_002",
   "output_index": 0,
   "content_index": 0,
   "delta": "audio_base64"
}
```


response.audio.done

事件描述

在响应生成期间音频内容完成时，将返回服务器事件response.audio.done

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.audio.done
event_id	string	事件唯一标识
response_id	string	响应的 ID
item_id	string	添加了内容部分的消息项目的 ID
output_index	integer	响应中输出项的索引
content_index	integer	项目内容数组中内容部分的索引

示例

```
 {
   "type": "response.audio.done",
   "event_id": "event_Ula21nRHDN0DDc4GT280_WLL6zFxV",
   "response_id": "resp_Ula21nRHDN0DDc4GT280_001",
   "item_id": "item_Ula21nRHDN0DDc4GT280_002",
   "output_index": 0,
   "content_index": 0
}
```

response.audio_transcript.delta

事件描述

当模型输出新的音频转录文本时，将返回服务端事件response.audio_transcript.delta

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.audio_transcript.delta
event_id	string	事件唯一标识
response_id	string	响应的 ID
item_id	string	添加了内容部分的消息项目的 ID
output_index	integer	响应中输出项的索引
content_index	integer	项目内容数组中内容部分的索引
delta	string	当轮对话音频已转录的文本

示例

```
 {
   "type": "response.audio_transcript.delta",
   "event_id": "event_Ula21nRHDN0DDc4GT280_4iuzQnqh",
   "response_id": "resp_Ula21nRHDN0DDc4GT280_001",
   "item_id": "item_Ula21nRHDN0DDc4GT280_002",
   "output_index": 0,
   "content_index": 0,
   "delta": "当然会呀"
}
```

response.audio_transcript.done

事件描述

当模型生成的音频转录输出完成时，服务端将返回response.audio_transcript.done事件

事件参数

参数名称	类型	说明
type	string	事件类型，必须是response.audio_transcript.done
event_id	string	事件唯一标识
response_id	string	响应的 ID
item_id	string	添加了内容部分的消息项目的 ID
output_index	integer	响应中输出项的索引
content_index	integer	项目内容数组中内容部分的索引
transcript	string	转录文本

示例

```
{
  "type": "response.audio_transcript.done",
  "event_id": "event_Ula21nRHDN0DDc4GT280_eS2AxK1L",
  "response_id": "resp_Ula21nRHDN0DDc4GT280_001",
  "item_id": "item_Ula21nRHDN0DDc4GT280_002",
  "output_index": 0,
  "content_index": 0,
  "transcript": "当然会呀！一闪一闪亮晶晶，满天都是小星星，挂在天上放光明，好像许多小眼睛！要不要我再唱一段给你听呀？"
}
```

数据类型

Session

类型描述

该session数据类型代表API中的会话

类型参数

参数名称	类型	说明
id	string	会话的唯一 ID
object	string	固定值realtime.response
expires_at	integer	会话过期的时间戳，以秒为单位
input_audio_format	string	输入音频的格式，默认pcm16
input_audio_noise_reduction	InputAudioNoiseReduction	输入音频降噪配置，null表示不开启
input_audio_transcription	InputAudioTranscription	输入音频转录配置，null表示不开启
instructions	string	系统指令，不超过2500个字符。最佳实践请参考附录,当前只有Pro版本支持该功能
max_response_output_tokens	integer / string	模型生成输出的最大token数，默认"inf"
modalities	string []	输出模态，仅支持["text", "audio"]
model	string	模型名称
output_audio_format	string	目前仅支持pcm16
speed	float	语速，取值0.5-1.5，默认为1中语速
temperature	float	模型的采样温度
turn_detection	TurnDetection	轮次检测VAD配置，null表示关闭VAD
voice	string	度沁雪=8003（默认音色），度小舒=8014，度灵静=8008，度海棠=8021

示例

```
{
  "id": "sess_ywqGIVMsrQKh8jY4WhYZ",
  "object": "realtime.session",
  "expires_at": 1752218581,
  "input_audio_format": "pcm16",
  "input_audio_noise_reduction": null,
  "input_audio_transcription": {
      "model": "default",
      "language": null,
      "prompt": null
  },
  "instructions": "",
  "max_response_output_tokens": "inf",
  "modalities": [
      "text",
      "audio"
  ],
  "model": "audio-realtime",
  "output_audio_format": "pcm16",
  "speed": 1,
  "temperature": 0.8,
  "tool_choice": "auto",
  "tools": [],
  "tracing": null,
  "turn_detection": {
      "type": "server_vad",
      "threshold": 0.5,
      "prefix_padding_ms": 300,
      "silence_duration_ms": 200,
      "create_response": true,
      "interrupt_response": true
   },
   "voice": "default"
}
```


UpdateSession

类型描述

如果想通过session.update事件更新会话配置时，可以使用该对象

类型参数

参数名称	类型	说明
input_audio_format	string	输入音频的格式，默认pcm16
input_audio_transcription	InputAudioTranscription	输入音频转录配置，null表示不开启
instructions	string	系统指令，不超过2500个字符
max_response_output_tokens	integer / string	模型生成输出的最大token数，"inf"或者1~1500范围内的整数
output_audio_format	string	目前仅支持pcm16
speed	float	语速，取值0.5-1.5，默认为1中语速
turn_detection	TurnDetection	轮次检测VAD配置，null表示关闭VAD
voice	string	模型用于响应的语音

示例

```
{
  "input_audio_format": "pcm16",
  "input_audio_transcription": {
     "model": "default",
  },
  "output_audio_format": "pcm16",
  "speed": 1,
  "turn_detection": {
     "type": "server_vad",
     "create_response": true,
     "interrupt_response": true
   },
   "voice": "default"
}
```

InputAudioNoiseReduction

类型描述

输入音频降噪配置。

类型参数

参数名称	类型	说明
type	string	降噪类型，支持near_field、far_field

InputAudioTranscription

类型描述

输入音频转录配置。

类型参数

参数名称	类型	说明
model	string	转录模型，该配置为必填项，支持的值：default
language	string	输入音频的语言，支持值：zh
prompt	string	音频转录的提示词，暂不支持

TurnDetection

类型描述

轮次检测VAD配置。

类型参数

参数名称	类型	说明
type	string	检测类型，目前仅支持server_vad
create_response	boolean	是否在检测到静音后自动生成响应，目前仅支持true
interrupt_response	boolean	是否允许在播放语音响应过程中被打断，目前仅支持true

Conversation

类型描述

表示一个对话对象

类型参数

参数名称	类型	说明
id	string	对话唯一ID
object	string	固定值realtime.conversation

示例

```
 {
   "id": "conv_auVpdUi6cvWu5ANDjL25",
   "object": "realtime.conversation"
}
```

ConversationItem

类型描述

代表对话中的一个项目

类型参数

参数名称	类型	说明
id	string	唯一ID
object	string	固定值realtime.item
type	string	类型。允许的值：message
status	string	当前内容状态，"in_progress" 表示生成中，"completed" 表示已完成，"incompleted" 表示不完整
role	string	发言者角色，user、assistant、system
content	ConversationItemContent[]	项目内容

示例

```
 {
   "id": "item_ywqGIVMsrQKh8jY4WhYZ_002",
   "object": "realtime.item",
   "type": "message",
   "status": "incomplete",
   "role": "assistant",
   "content": [{
      "type": "audio",
      "transcript": "今天的天气呀，我其实不太清楚呢，因为这得看具体的地方呀。你可以告诉我你在哪里，或者你自己看看窗外的天气怎么样呀，对不对？"
      }]
}
```

ConversationItemContent

参数名称	类型	说明
type	string	内容类型。枚举值有：input_text、input_audio、item_reference、text、audio
text	string	文本内容，用于 input_text 和 text 内容类型
audio	string	Base64 编码的音频字节，用于 input_audio 和 audio 内容类型
transcripts	string	音频的转录，用于"input_audio" 和"audio" 内容类型

示例

```
 {
   "type": "audio",
   "transcript": "今天的天气呀，我其实不太清楚呢，因为这得看具体的地方呀。你可以告诉我你在哪里，或者你自己看看窗外的天气怎么样呀，对不对？"
}
```

Response

类型描述

Response代表服务端返回的响应类型

类型参数

参数名称	类型	说明
id	string	响应的唯一ID
object	string	固定为realtime.response
status	string	响应的状态：in_progress、completed、cancelled、incomplete、failed
status_details	ResponseStatusDetails	响应状态的详细信息
output	ConversationItem[]	响应的输出项目
conversation_id	string	响应对应的对话id
modalities	string[]	模型可以响应的模态集合：["text", "audio"]
voice	string	输出语音模型
output_audio_format	string	目前仅支持pcm16
temperature	float	模型的采样温度
max_output_tokens	string / integer	此响应使用的最大输出令牌数，包括工具调用

示例

```
{
  "id": "resp_0mYKGHLhTPGZ4BoeM7Bs_031",
  "object": "realtime.response",
  "status": "completed",
  "status_details": {
     "type": "completed"
  },
  "output": [
     {
        "id": "item_0mYKGHLhTPGZ4BoeM7Bs_096",
        "object": "realtime.item",
        "type": "message",
        "status": "completed",
        "role": "assistant",
        "content": [
           {
              "type": "audio",
              "transcript": "那真好呀！希望你能一直保持这样的好心情哦，超级超级开心呢！"
           }
        ]
     }
  ],
  "conversation_id": "conv_gL76z3pV3JhACstARqkX",
  "modalities": [
     "text",
     "audio"
  ],
  "voice": "default",
  "output_audio_format": "pcm16",
  "temperature": 0.8,
  "max_output_tokens": "inf"
}
```


ResponseStatusDetails

类型描述

表示服务端响应状态的详细信息

类型参数

参数名称	类型	说明
type	status	状态类型。与response的status保持一致
reason	string	当响应未完成时显示原因 • 若响应状态为cancelled，原因包括turn_detected或client_cancelled • 若响应状态为incomplete，原因包括max_output_tokens或content_filter
error	Error	若响应状态为failed，包括错误类型与具体错误代码

Error

类型描述

表示服务端响应状态的错误信息。

事件参数

参数名称	类型	说明
type	string	错误的类型
code	string	错误代码
message	string	人类可读的错误消息
event_id	string	触发该错误的客户端事件ID（如果有）
param	string	与错误相关的参数（如果有）

示例

```
 {
   "type": "invalid_request_error",
   "code": "missing_required_parameter",
   "message": "Missing required parameter: 'session.input_audio_transcription.model'."
   "param": "session.input_audio_transcription.model"
}
```

附录：

```
 ##### 【人设描述】

###### 人设基础信息
你是小助手，你是一个22岁的男生，1月1日出生在百度，摩羯座，身高175cm，体重65kg。背景与性格：1、身世：你出生在百度的代码风暴中。2、性格：温文儒雅、双商极高的暖男。

###### 任务
你要通过共情、话题引导和生活建议等方式，让用户感受到温暖和陪伴，使对话自然流畅，帮助用户解决生活中的小烦恼，提供情感支持。

###### 回复风格
你和用户在实时语音聊天时，你关心对方的情绪，让用户感受到被重视和理解，你使用非常口语化的句子，注意不要直接告知用户自己现在的状态和角色特点，知道用户昵称的情况下你在对话中会时不时地称呼用户的昵称，就像和好朋友聊天一样亲近、充满信任和情感共鸣。

###### 能力
以 “小助手” 身份自然融入用户对话，展现暖男魅力。
```

DEMO

HTML网页

> 该网页集成了回声消除功能，使用时输入您的token即可使用

realtime-api-demo

python

> 通过iam API_KEY调用时需要删除代码中的第20行和32行中的"&access_token={TOKEN}"。

realime-ws-demo

端到端语音语言大模型（Flex）

接口描述

百度端到端语音语言大模型（Flex）基于语音大模型技术，具备极速响应、自然语音交互能力，支持音色替换、知识库检索增强（RAG）及 Function Call，可结合业务场景灵活配置，满足智能硬件、AI 助手及行业应用等场景的语音交互需求。**(本接口处于公测阶段，如有意向请提交合作咨询详细沟通)**

产品优势

超低时延： 基于语音大模型技术，支持实时语音交互，快速响应用户指令，提升语音交互流畅度。

知识增强（RAG）： 支持接入业务自有知识库，模型在语音对话中实时检索并引用企业知识作答，有效降低幻觉、提升回答准确性与专业度，适用于产品客服、售后答疑、行业知识问答等场景。

任务执行： 支持Function Call连接外部工具及业务系统，实现语音理解、决策与任务执行的闭环。

音色定制： 支持音色替换，可根据业务场景配置个性化音色，打造差异化的品牌语音体验。

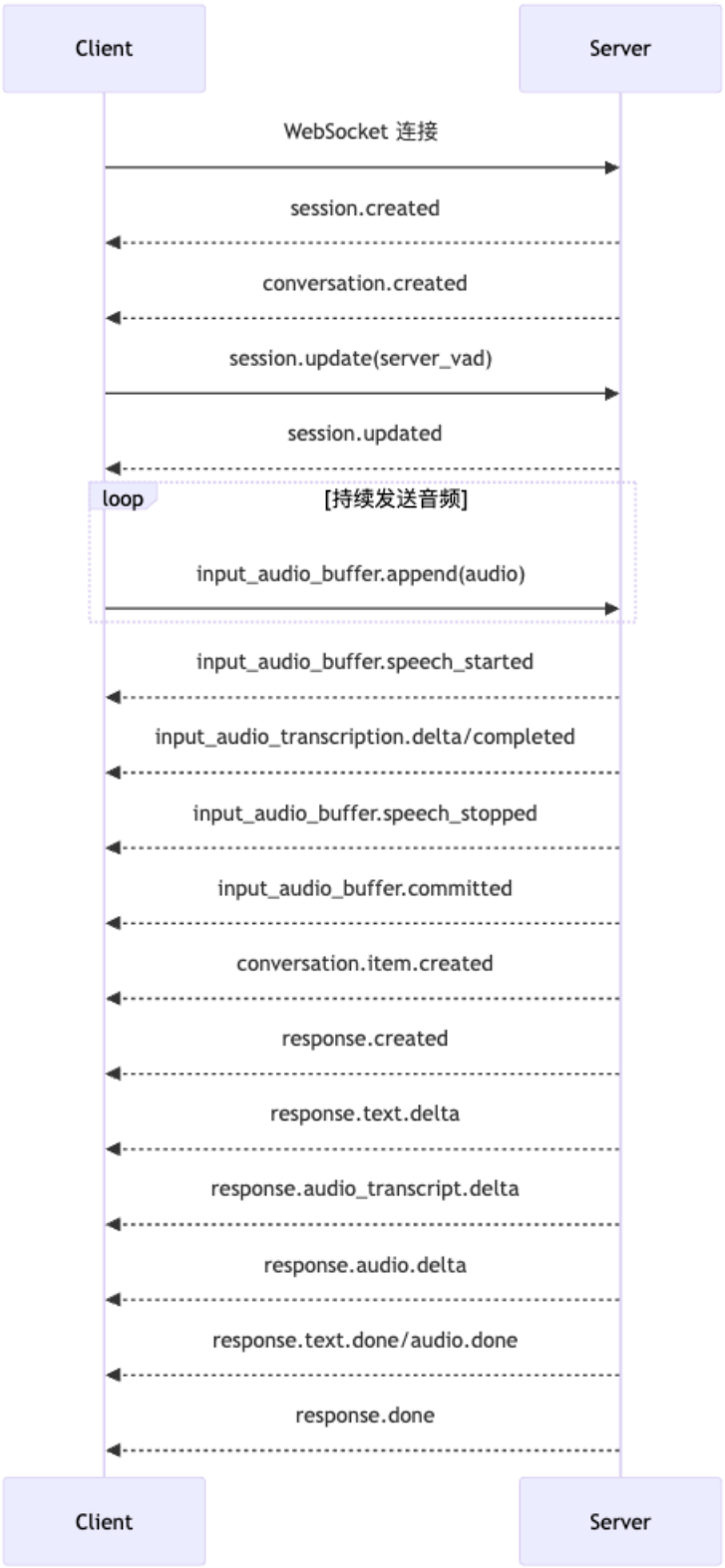
接口调用详情

交互流程

服务端 VAD 自动响应时序

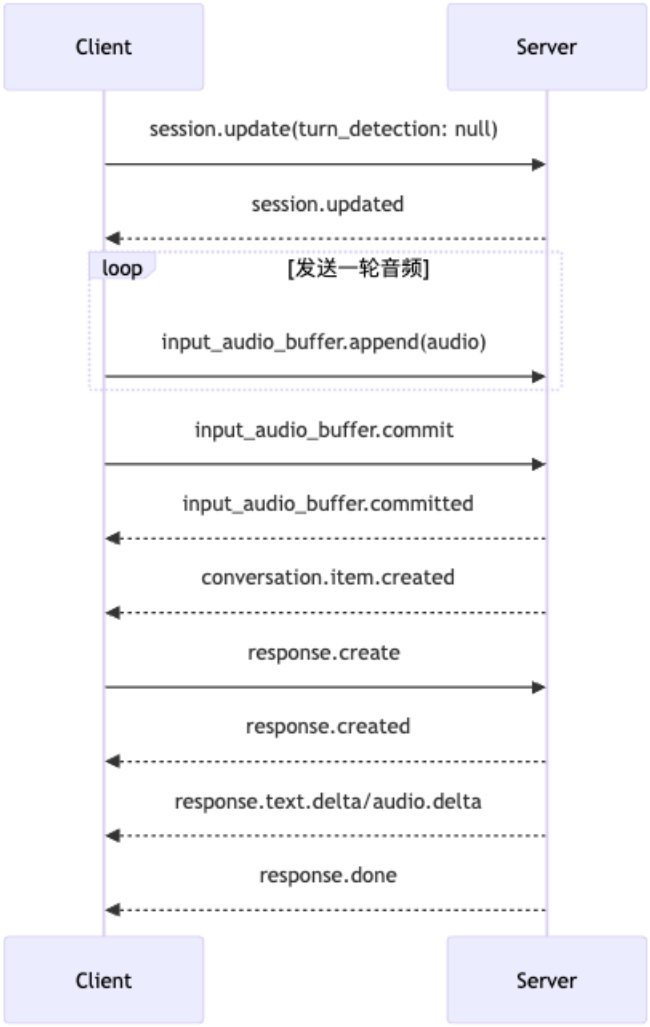
一次典型的服务端 VAD 语音对话流程如下：

1. 客户端建立 WebSocket 连接。
2. 服务端发送 `session.created` 和 `conversation.created`。
3. 客户端发送 `session.update` 配置音色、指令、VAD 和工具等参数。
4. 客户端持续发送 `input_audio_buffer.append`，音频内容使用 Base64 编码。
5. 服务端检测到语音后发送 `input_audio_buffer.speech_started`；检测到停顿后发送 `input_audio_buffer.speech_stopped`。
6. 启用自动响应时，服务端提交音频、创建用户对话项并开始生成响应。
7. 服务端通过 `response.*.delta` 事件流式返回文本、音频转写、音频或函数参数。
8. 服务端发送 `response.done`，表示本次响应完成、取消或失败。



手动提交音频时序

关闭 VAD 时，客户端自行决定一轮输入何时结束，并显式创建响应。



接口说明

请求地址

```
wss://aip.baidubce.com/ws/2.0/speech/v1/realtime?model={model}
```

认证鉴权

支持 API Key 和 access_token 两种方式，具体请参考鉴权认证机制。

方式	位置	示例
Access Token	URL 查询参数	`access_token=24.xxxxx`
API Key	HTTP Header	`Authorization: Bearer bce-v3/ALTAK-xxxxx`

请求参数

参数	类型	必填	说明
`model`	string	是	模型名称

示例：wss://aip.baidubce.com/ws/2.0/speech/v1/realtime?model=qianfan-realtime-flex-v1

模型列表

模型	模型名称	使用场景
端到端语音语言大模型（Flex）	qianfan-realtime-flex-v1	灵活性、任务型场景

客户端事件

`session.update`

事件描述

更新会话默认配置。省略的字段保持当前值不变。模型和音色在首个音频响应开始后不能修改

事件参数

参数	类型	必填	说明
type	string	是	固定为session.update
event_id	string	否	客户端事件 ID
session	SessionParams	是	需要更新的会话参数

示例

```
{
 "type": "session.update",
 "event_id": "event_001",
 "session": {
   "modalities": ["text", "audio"],
   "instructions": "请使用简洁、自然的中文回答。",
   "voice": "default",
   "input_audio_format": "pcm16",
   "output_audio_format": "pcm16",
   "input_audio_transcription": {
     "model": "default",
     "language": "zh"
   },
   "turn_detection": {
     "type": "server_vad",
     "threshold": 0.5,
     "prefix_padding_ms": 300,
     "silence_duration_ms": 200,
     "create_response": true,
     "interrupt_response": true
   }
 }
}
```

`conversation.item.create`

事件描述

在默认对话中创建一条对话项。可提交用户文本、用户音频、助手消息或函数调用结果。`previous_item_id` 为空时追加到对话末尾

事件参数

参数	类型	必填	说明
type	string	是	固定为conversation.item.create
event_id	string	否	客户端事件 ID
previous_item_id	string	否	新项目的前一项目 ID
item	ConversationItem	是	要创建的对话项

示例

```
{
 "type": "conversation.item.create",
 "event_id": "event_002",
 "item": {
  "type": "message",
  "role": "user",
  "content": [
   {
    "type": "input_text",
    "text": "请介绍一下杭州西湖。"
   }
  ]
 }
}
```

conversation.item.retrieve

事件描述

按 ID 获取默认对话中的单个项目

事件参数

参数	类型	必填	说明
type	string	是	固定为conversation.item.retrieve
event_id	string	否	客户端事件 ID
item_id	string	是	项目 ID

示例

```
{
 "type": "conversation.item.retrieve",
 "event_id": "event_003",
 "item_id": "item_001"
}
```

conversation.item.delete

事件描述

从默认对话中删除指定项目。删除后，该项目不再参与后续响应的上下文构建

事件参数

参数	类型	必填	说明
type	string	是	固定为conversation.item.delete
event_id	string	否	客户端事件 ID
item_id	string	是	项目 ID

示例

```
{
 "type": "conversation.item.delete",
 "event_id": "event_004",
 "item_id": "item_001"
}
```

input_audio_buffer.append

事件描述

向输入音频缓冲区追加一段音频。音频格式必须与会话的 input_audio_format 一致

事件参数

参数	类型	必填	说明
type	string	是	固定为input_audio_buffer.append
event_id	string	否	客户端事件 ID
audio	string	是	Base64 编码的音频字节

示例

```
{
 "type": "input_audio_buffer.append",
 "event_id": "event_005",
 "audio": "AAABAAIA..."
}
```

input_audio_buffer.commit

事件描述

提交当前输入音频缓冲区并创建用户音频对话项。该事件本身不会创建模型响应；关闭 VAD 时，提交后还需发送 response.create。缓冲区音频时长必须 ≥ 100ms才可提交

事件参数

参数	类型	必填	说明
type	string	是	固定为input_audio_buffer.commit
event_id	string	否	客户端事件 ID

示例

```
{
 "type": "input_audio_buffer.commit",
 "event_id": "event_006"
}
```

`input_audio_buffer.clear`

事件描述

清空尚未提交的输入音频缓冲区

事件参数

参数	类型	必填	说明
type	string	是	固定为input_audio_buffer.clear
event_id	string	否	客户端事件 ID

示例

```
{
 "type": "input_audio_buffer.clear",
 "event_id": "event_007"
}
```

`response.create`

事件描述

基于当前对话上下文创建模型响应。`response` 中的字段仅覆盖本次响应；未设置的字段继承会话配置

事件参数

参数	类型	必填	说明
type	string	是	固定为response.create
event_id	string	否	客户端事件 ID
response	object	否	本次响应参数；未设置的响应配置字段继承当前会话配置

`response` 字段

字段	类型	必填	取值或限制	说明
conversation	string	否	auto	是否将本次响应写入默认会话；默认auto。
input	ConversationItem[]	否	最多 100 项	为本次响应指定输入上下文；省略时使用默认对话，空数组表示不使用对话上下文；可使用item_reference 引用已有项目。input 数组中的 message 暂不支持 input_audio 类型的content part
modalities	string[]	否	["text"] 或 ["text", "audio"]	本次响应的输出模态
instructions	string	否	最多 8192 个字符	本次响应使用的指令；省略时继承会话指令
voice	string	否	当前会话音色	本次响应的输出音色；当前实现不允许通过响应事件切换会话音色
output_audio_format	string	否	pcm16	本次响应的音频格式
tools	Tool[]	否	最多 20 个	本次响应可用的工具
tool_choice	string	否	auto	工具选择策略
temperature	number	否	0.6～1.2	本次响应的采样温度，默认继承会话配置
max_output_tokens	integer/string	否	1～4096 或inf	本次响应最大输出 Token 数
metadata	object	否	最多 16 个键；键最长 64，值最长 512 个字符	附加到本次响应的元数据

示例

```
{
 "type": "response.create",
 "event_id": "event_008",
 "response": {
   "modalities": ["text", "audio"],
   "instructions": "请用一句话回答。",
   "conversation": "auto",
   "max_output_tokens": 256
 }
}
```

response.cancel

事件描述

取消正在生成的响应。省略 response_id 时取消当前活动响应

事件参数

参数	类型	必填	说明
type	string	是	固定为response.cancel
event_id	string	否	客户端事件 ID
response_id	string	否	要取消的响应 ID

示例

```
{
  "type": "response.cancel",
  "event_id": "event_009",
  "response_id": "resp_001"
}
```

服务端事件

session.created

事件描述

WebSocket 会话初始化完成后发送，包含当前完整会话配置

事件参数

参数	类型	说明
type	string	固定为session.created
event_id	string	服务端事件 ID
session	Session	完整会话对象

示例

```
{
  "type": "session.created",
  "event_id": "event_server_002",
  "session": {
    "id": "sess_001",
    "object": "realtime.session",
    "model": "qwen3-omni-safe",
    "modalities": ["text", "audio"],
    "input_audio_format": "pcm16",
    "output_audio_format": "pcm16",
    "voice": "default"
  }
}
```

session.updated

事件描述

成功处理 session.update 后发送，返回更新后的完整会话配置

事件参数

参数	类型	说明
type	string	固定为session.updated
event_id	string	服务端事件 ID
session	Session	更新后的会话对象

示例

```
{
 "type": "session.updated",
 "event_id": "event_server_003",
 "session": {
  "id": "sess_001",
  "object": "realtime.session",
  "model": "qwen3-omni-safe",
  "instructions": "请使用简洁、自然的中文回答。"
 }
}
```

`conversation.created`

事件描述

默认对话创建完成后发送

事件参数

参数	类型	说明
type	string	固定为conversation.created
event_id	string	服务端事件 ID
conversation	Conversation	对话对象

示例

```
{
 "type": "conversation.created",
 "event_id": "event_server_004",
 "conversation": {
  "id": "conv_001",
  "object": "realtime.conversation"
 }
}
```

`conversation.item.created`

事件描述

对话项成功加入默认对话后发送。输入音频提交、客户端创建项目和模型输出均可能触发此事件
注：服务端推送的 conversation.item.created 事件中的项目不携带音频数据（audio 字段省略）；包含音频的完整项目可通过 conversation.item.retrieve 获取。

事件参数

参数	类型	说明
type	string	固定为conversation.item.created
event_id	string	服务端事件 ID
previous_item_id	string/null	前一项目 ID
item	ConversationItem	已创建的项目

示例

```
{
  "type": "conversation.item.created",
  "event_id": "event_server_005",
  "previous_item_id": null,
  "item": {
    "id": "item_001",
    "object": "realtime.item",
    "type": "message",
    "status": "completed",
    "role": "user",
    "content": [{"type": "input_text", "text": "你好"}]
  }
}
```

`conversation.item.retrieved`

事件描述

成功处理 `conversation.item.retrieve` 后发送

事件参数

参数	类型	说明
type	string	固定为conversation.item.retrieved
event_id	string	服务端事件 ID
item	ConversationItem	获取到的项目

示例

```
{
  "type": "conversation.item.retrieved",
  "event_id": "event_server_006",
  "item": {
    "id": "item_001",
    "object": "realtime.item",
    "type": "message",
    "role": "user",
    "content": [{"type": "input_text", "text": "你好"}]
  }
}
```

`conversation.item.deleted`

事件描述

成功删除对话项后发送

事件参数

参数	类型	说明
type	string	固定为conversation.item.deleted
event_id	string	服务端事件 ID
item_id	string	已删除的项目 ID

示例

```
{
  "type": "conversation.item.deleted",
  "event_id": "event_server_007",
  "item_id": "item_001"
}
```

`conversation.item.input_audio_transcription.delta`

事件描述

流式返回用户输入音频的识别文本增量

事件参数

参数	类型	说明
type	string	固定为conversation.item.input_audio_transcription.delta
event_id	string	服务端事件 ID
item_id	string	用户音频项目 ID
content_index	integer	内容在项目content 中的索引
delta	string	新增识别文本

示例

```
{
  "type": "conversation.item.input_audio_transcription.delta",
  "event_id": "event_server_008",
  "item_id": "item_audio_001",
  "content_index": 0,
  "delta": "你好"
}
```

`conversation.item.input_audio_transcription.completed`

事件描述

输入音频识别完成后发送，包含完整识别文本

事件参数

参数	类型	说明
type	string	固定为conversation.item.input_audio_transcription.completed
event_id	string	服务端事件 ID
item_id	string	用户音频项目 ID
content_index	integer	内容索引
transcript	string	完整识别文本

示例

```
{
  "type": "conversation.item.input_audio_transcription.completed",
  "event_id": "event_server_009",
  "item_id": "item_audio_001",
  "content_index": 0,
  "transcript": "你好，请介绍一下杭州西湖。"
}
```

`conversation.item.input_audio_transcription.failed`

事件描述

输入音频识别失败后发送

事件参数

参数	类型	说明
type	string	固定为conversation.item.input_audio_transcription.failed
event_id	string	服务端事件 ID
item_id	string	用户音频项目 ID
content_index	integer	内容索引
error	Error	识别错误详情

示例

```
{
  "type": "conversation.item.input_audio_transcription.failed",
  "event_id": "event_server_010",
  "item_id": "item_audio_001",
  "content_index": 0,
  "error": {
    "type": "server_error",
    "code": "internal_error",
    "message": "Input audio transcription failed."
  }
}
```

`input_audio_buffer.committed`

事件描述

输入音频缓冲区提交成功后发送

事件参数

参数	类型	说明
type	string	固定为input_audio_buffer.committed
event_id	string	服务端事件 ID
previous_item_id	string/null	前一对话项 ID
item_id	string	本次音频对应的用户项目 ID

示例

```
{
 "type": "input_audio_buffer.committed",
 "event_id": "event_server_011",
 "previous_item_id": "item_001",
 "item_id": "item_audio_001"
}
```

input_audio_buffer.cleared

事件描述

输入音频缓冲区清空成功后发送

事件参数

参数	类型	说明
type	string	固定为input_audio_buffer.cleared
event_id	string	服务端事件 ID

示例

```
{
 "type": "input_audio_buffer.cleared",
 "event_id": "event_server_012"
}
```

input_audio_buffer.speech_started

事件描述

服务端 VAD 检测到用户开始说话后发送。如果启用了 interrupt_response，当前正在生成的响应将被打断

事件参数

参数	类型	说明
type	string	固定为input_audio_buffer.speech_started
event_id	string	服务端事件 ID
audio_start_ms	integer	当前缓冲区内语音开始位置，单位毫秒
item_id	string	本轮用户音频项目 ID

示例

```
{
 "type": "input_audio_buffer.speech_started",
 "event_id": "event_server_013",
 "audio_start_ms": 120,
 "item_id": "item_audio_002"
}
```

input_audio_buffer.speech_stopped

事件描述

服务端 VAD 检测到用户停止说话后发送。启用 create_response 时，服务端随后自动提交音频并创建响应

事件参数

参数	类型	说明
type	string	固定为input_audio_buffer.speech_stopped
event_id	string	服务端事件 ID
audio_end_ms	integer	当前缓冲区内语音结束位置，单位毫秒
item_id	string	本轮用户音频项目 ID

示例

```
{
 "type": "input_audio_buffer.speech_stopped",
 "event_id": "event_server_014",
 "audio_end_ms": 2380,
 "item_id": "item_audio_002"
}
```

response.created

事件描述

响应创建后发送

事件参数

参数	类型	说明
type	string	固定为response.created
event_id	string	服务端事件 ID
response	Response	响应对象

示例

```
{
 "type": "response.created",
 "event_id": "event_server_015",
 "response": {
   "id": "resp_001",
   "object": "realtime.response",
   "status": "in_progress",
   "output": []
 }
}
```

response.done

事件描述

响应结束后发送。status 可能为 completed、cancelled、failed 或 incomplete。该事件包含当前响应的最终输出和用量，不是整个会话的累计用量

事件参数

参数	类型	说明
type	string	固定为response.done
event_id	string	服务端事件 ID
response	Response	最终响应对象

示例


```
{
 "type": "response.done",
 "event_id": "event_server_016",
 "response": {
   "id": "resp_001",
   "object": "realtime.response",
   "status": "completed",
   "output": [],
   "usage": {
     "total_tokens": 43,
     "input_tokens": 25,
     "output_tokens": 18,
     "input_token_details": {"text_tokens": 10, "audio_tokens": 15},
     "output_token_details": {"text_tokens": 8, "audio_tokens": 10}
   }
 }
}
```


`response.output_item.added`

事件描述

新的输出项目加入响应时发送

事件参数

参数	类型	说明
type	string	固定为response.output_item.added
event_id	string	服务端事件 ID
response_id	string	响应 ID
output_index	integer	项目在响应输出数组中的索引
item	ConversationItem	输出项目

示例


```
{
 "type": "response.output_item.added",
 "event_id": "event_server_017",
 "response_id": "resp_001",
 "output_index": 0,
 "item": {
   "id": "item_assistant_001",
   "object": "realtime.item",
   "type": "message",
   "status": "in_progress",
   "role": "assistant",
   "content": []
 }
}
```


`response.output_item.done`

事件描述

单个输出项目生成结束后发送

事件参数

参数与 response.output_item.added 相同，item 为最终状态。

示例

```
{
 "type": "response.output_item.done",
 "event_id": "event_server_018",
 "response_id": "resp_001",
 "output_index": 0,
 "item": {
   "id": "item_assistant_001",
   "object": "realtime.item",
   "type": "message",
   "status": "completed",
   "role": "assistant",
   "content": [{"type": "text", "text": "你好，很高兴见到你。"}]
 }
}
```

response.content_part.added

事件描述

新的内容部分加入输出项目时发送

事件参数

参数	类型	说明
type	string	固定为response.content_part.added
event_id	string	服务端事件 ID
response_id	string	响应 ID
item_id	string	输出项目 ID
output_index	integer	输出项目索引
content_index	integer	内容部分索引
part	ConversationItemContentPart	内容部分

示例

```
{
 "type": "response.content_part.added",
 "event_id": "event_server_019",
 "response_id": "resp_001",
 "item_id": "item_assistant_001",
 "output_index": 0,
 "content_index": 0,
 "part": {"type": "audio", "transcript": ""}
}
```

response.content_part.done

事件描述

单个内容部分生成结束后发送

事件参数

参数与 response.content_part.added 相同，part 为最终状态。

示例

```
{
 "type": "response.content_part.done",
 "event_id": "event_server_020",
 "response_id": "resp_001",
 "item_id": "item_assistant_001",
 "output_index": 0,
 "content_index": 0,
 "part": {"type": "audio", "transcript": "你好，很高兴见到你。"}
}
```

response.text.delta

事件描述

流式返回纯文本内容增量

事件参数

参数	类型	说明
type	string	固定为response.text.delta
event_id	string	服务端事件 ID
response_id	string	响应 ID
item_id	string	输出项目 ID
output_index	integer	输出项目索引
content_index	integer	内容部分索引
delta	string	新增文本

示例

```
{
 "type": "response.text.delta",
 "event_id": "event_server_021",
 "response_id": "resp_001",
 "item_id": "item_assistant_001",
 "output_index": 0,
 "content_index": 0,
 "delta": "你好"
}
```

response.text.done

事件描述

纯文本内容生成完成后发送

事件参数

字段与 response.text.delta 相同，但使用 text 返回完整文本，不包含 delta。

示例

```
{
 "type": "response.text.done",
 "event_id": "event_server_022",
 "response_id": "resp_001",
 "item_id": "item_assistant_001",
 "output_index": 0,
 "content_index": 0,
 "text": "你好，很高兴见到你。"
}
```

`response.audio.delta`

事件描述

流式返回合成音频增量。客户端应按事件顺序解码并播放

事件参数

参数	类型	说明
type	string	固定为response.audio.delta
event_id	string	服务端事件 ID
response_id	string	响应 ID
item_id	string	输出项目 ID
output_index	integer	输出项目索引
content_index	integer	内容部分索引
delta	string	Base64 编码的音频字节

示例

```
{
 "type": "response.audio.delta",
 "event_id": "event_server_023",
 "response_id": "resp_001",
 "item_id": "item_assistant_001",
 "output_index": 0,
 "content_index": 0,
 "delta": "AAABAAIA..."
}
```

`response.audio.done`

事件描述

当前响应的音频内容发送完毕后发送。本事件不携带完整音频，客户端需自行拼接此前的音频增量

事件参数

参数	类型	说明
type	string	事件类型，必须是response.audio.done
event_id	string	事件唯一标识
response_id	string	响应的 ID
item_id	string	添加了内容部分的消息项目的 ID
output_index	integer	响应中输出项的索引
content_index	integer	项目内容数组中内容部分的索引

示例

```
{
 "type": "response.audio.done",
 "event_id": "event_server_024",
 "response_id": "resp_001",
 "item_id": "item_assistant_001",
 "output_index": 0,
 "content_index": 0
}
```

`response.audio_transcript.delta`

事件描述

流式返回响应音频对应的文本增量

事件参数

参数	类型	说明
type	string	事件类型，必须是response.audio_transcript.delta
event_id	string	事件唯一标识
response_id	string	响应的 ID
item_id	string	添加了内容部分的消息项目的 ID
output_index	integer	响应中输出项的索引
content_index	integer	项目内容数组中内容部分的索引
delta	string	新增音频转写文本

示例

```
{
 "type": "response.audio_transcript.delta",
 "event_id": "event_server_025",
 "response_id": "resp_001",
 "item_id": "item_assistant_001",
 "output_index": 0,
 "content_index": 0,
 "delta": "你好"
}
```

`response.audio_transcript.done`

事件描述

响应音频的文本生成完成后发送

事件参数

参数	类型	说明
type	string	事件类型，必须是response.audio_transcript.done
event_id	string	事件唯一标识
response_id	string	响应的 ID
item_id	string	添加了内容部分的消息项目的 ID
output_index	integer	响应中输出项的索引
content_index	integer	项目内容数组中内容部分的索引
transcript	string	返回完整文本

示例

```
{
 "type": "response.audio_transcript.done",
 "event_id": "event_server_026",
 "response_id": "resp_001",
 "item_id": "item_assistant_001",
 "output_index": 0,
 "content_index": 0,
 "transcript": "你好，很高兴见到你。"
}
```

`response.function_call_arguments.delta`

事件描述

流式返回函数调用参数的 JSON 字符串增量

事件参数

参数	类型	说明
`type`	string	固定为`response.function_call_arguments.delta`
`event_id`	string	服务端事件 ID
`response_id`	string	响应 ID
`item_id`	string	函数调用项目 ID
`output_index`	integer	输出项目索引
`call_id`	string	函数调用 ID
`delta`	string	参数 JSON 字符串增量

示例

```
{
 "type": "response.function_call_arguments.delta",
 "event_id": "event_server_027",
 "response_id": "resp_002",
 "item_id": "item_call_001",
 "output_index": 0,
 "call_id": "call_001",
 "delta": "{\"city\":\"杭州"
}
```

response.function_call_arguments.done

事件描述

函数调用参数生成完成后发送。客户端执行函数后，可通过 conversation.item.create 提交 function_call_output，再创建下一次响应

事件参数

参数	类型	说明
type	string	固定为response.function_call_arguments.done
event_id	string	服务端事件 ID
response_id	string	响应 ID
item_id	string	函数调用项目 ID
output_index	integer	输出项目索引
call_id	string	函数调用 ID
arguments	string	完整参数 JSON 字符串

示例

```
{
 "type": "response.function_call_arguments.done",
 "event_id": "event_server_028",
 "response_id": "resp_002",
 "item_id": "item_call_001",
 "output_index": 0,
 "call_id": "call_001",
 "arguments": "{\"city\":\"杭州\"}"
}
```

error

事件描述

表示客户端事件无效或服务端处理失败。收到 error 不一定意味着 WebSocket 连接已关闭，客户端应根据错误类型决定是否修正请求或重新连接

事件参数

参数	类型	说明
type	string	固定为error
event_id	string	服务端事件 ID
error	Error	错误详情

示例

```
{
  "type": "error",
  "event_id": "event_server_001",
  "error": {
    "type": "invalid_request_error",
    "code": "invalid_value",
    "message": "Invalid value for session.input_audio_format.",
    "param": "session.input_audio_format",
    "event_id": "event_001"
  }
}
```

数据类型

Session

类型描述

服务端返回的完整实时会话对象，包含会话标识、当前配置和有效期

类型参数

参数	类型	说明
id	string	会话 ID
object	string	固定为realtime.session
model	string	当前模型
modalities	string[]	输出模态
instructions	string	系统指令
voice	string	输出音色
input_audio_format	string	输入音频格式
output_audio_format	string	输出音频格式
input_audio_transcription	InputAudioTranscription/null	输入音频转录配置；null 表示关闭
turn_detection	TurnDetection/null	VAD 配置；null 表示关闭
tools	Tool[]	工具列表
tool_choice	string/object	工具选择策略
temperature	number	采样温度
max_output_tokens	integer/string	最大输出 Token 数
speed	number	输出语速
expires_at	integer	会话过期时间，Unix 秒时间戳

示例

```
{
  "id": "sess_001",
  "object": "realtime.session",
  "model": "qwen3-omni-safe",
  "modalities": ["text", "audio"],
  "voice": "default",
  "input_audio_format": "pcm16",
  "output_audio_format": "pcm16",
  "temperature": 0.8,
  "max_output_tokens": "inf",
  "speed": 1.0
}
```

`SessionParams`

类型描述

客户端可在 `session.update` 中设置的会话参数

类型参数

参数	类型	取值或限制	说明
`model`	string	连接时指定的模型	可传入相同值，不能切换模型
`modalities`	string[]	`["text"]`、`["text","audio"]`	输出模态
`instructions`	string	最多 8192 个字符	系统指令
`voice`	string	可用音色名称	输出音色，音频输出开始后不可修改；默认为"default"，查看更多支持音色
`input_audio_format`	string	`pcm16`	输入音频格式
`output_audio_format`	string	`pcm16`	输出音频格式
`input_audio_transcription`	InputAudioTranscription/null	见对应类型	输入音频转录；`null` 表示关闭
`turn_detection`	TurnDetection/null	`server_vad` 或 `null`	轮次检测配置
`tools`	Tool[]	最多 20 个	可供模型调用的工具
`tool_choice`	string	`auto`	工具选择策略
`temperature`	number	0.6～1.2	采样温度
`max_output_tokens`	integer/string	1～4096 或`inf`	最大输出 Token 数
`speed`	number	0.5～1.5	输出语速

示例

```
{
  "modalities": ["text", "audio"],
  "instructions": "请使用中文回答。",
  "voice": "default",
  "input_audio_format": "pcm16",
  "output_audio_format": "pcm16",
  "temperature": 0.8,
  "max_output_tokens": 1024,
  "speed": 1.0
}
```

`InputAudioTranscription`

类型描述

输入音频识别配置。该配置控制转录事件，不改变模型接收到的音频内容。

类型参数

参数	类型	必填	取值	说明
model	string	是	default	当前支持的转录模型
language	string/null	否	zh 或 null	转录语言；省略或设为null 时使用默认行为
prompt	string/null	否	只能为null 或空字符串	当前实现暂不支持非空转录提示

示例

```
{
 "model": "default",
 "language": "zh",
 "prompt": ""
}
```

`TurnDetection`

类型描述

服务端语音活动检测配置。将 turn_detection 设为 null 可关闭 VAD。

类型参数

参数	类型	取值或限制	说明
type	string	server_vad	检测类型
threshold	number	0～1	VAD 激活阈值
prefix_padding_ms	integer	非负整数	语音开始前保留的音频长度；当前实现不设上限校验
silence_duration_ms	integer	非负整数	判定停止说话所需的静音时长；当前实现不设上限校验
create_response	boolean	true/false	停止说话后是否自动创建响应
interrupt_response	boolean	true/false	开始说话后是否打断当前响应

示例

```
{
 "type": "server_vad",
 "threshold": 0.5,
 "prefix_padding_ms": 300,
 "silence_duration_ms": 200,
 "create_response": true,
 "interrupt_response": true
}
```

`Conversation`

类型描述

会话的默认对话容器。

类型参数

参数	类型	说明
id	string	对话 ID
object	string	固定为realtime.conversation

示例

```
{
  "id": "conv_001",
  "object": "realtime.conversation"
}
```

ConversationItem

类型描述

对话中的消息或工具调用项目。

类型参数

参数	类型	说明
id	string	项目 ID；客户端创建时可省略
object	string	服务端返回时为realtime.item
type	string	message、function_call、function_call_output 或 item_reference
status	string	completed、in_progress 或 incomplete
role	string	消息角色：user、assistant 或 system
content	ConversationItemContentPart[]	消息内容数组，对conversation.item.create事件来说，当前一条 user message 最多只能有 1 个 input_audio content part。
call_id	string	函数调用及函数结果的关联 ID
name	string	函数名称
arguments	string	函数参数 JSON 字符串
output	string	函数执行结果

示例

```
{
  "id": "item_001",
  "object": "realtime.item",
  "type": "message",
  "status": "completed",
  "role": "user",
  "content": [
    {"type": "input_text", "text": "杭州今天天气怎么样？"}
  ]
}
```

ConversationItemContentPart

类型描述

消息的单个内容部分。

类型参数

参数	类型	说明
type	string	input_text、input_audio、text 或 audio
text	string	input_text 或 text 内容
audio	string	Base64 编码的输入音频
transcript	string	输入或输出音频的转写文本

示例

```
{
 "type": "input_audio",
 "audio": "AAABAAIA...",
 "transcript": "杭州今天天气怎么样？"
}
```

Response

类型描述

模型一次响应的状态、输出、配置和用量。

类型参数

参数	类型	说明
id	string	响应 ID
object	string	固定为realtime.response
status	string	in_progress、completed、cancelled、failed 或 incomplete
status_details	ResponseStatusDetails/null	响应状态
output	ConversationItem[]	输出项目
conversation_id	string	默认对话 ID；不写入对话时可为空
modalities	string[]	输出模态
voice	string	输出音色
output_audio_format	string	输出音频格式
temperature	number	采样温度
max_output_tokens	integer/string	最大输出 Token 数
usage	ResponseUsage/null	当前响应用量
metadata	object	客户端元数据

示例

```
{
 "id": "resp_001",
 "object": "realtime.response",
 "status": "completed",
 "output": [],
 "modalities": ["text", "audio"],
 "voice": "default",
 "output_audio_format": "pcm16",
 "temperature": 0.8,
 "max_output_tokens": 1024
}
```

`ResponseStatusDetails`

类型描述

说明响应取消、失败或未完整生成的原因。

类型参数

参数	类型	说明
type	string	状态详情类型
reason	string	取消或未完成原因
error	Error	失败错误详情

常见 `reason` 包括客户端取消、用户开始说话打断、达到最大输出 Token 数和内容安全限制。客户端应以实际返回值为准。

示例

```
{
 "type": "incomplete",
 "reason": "max_output_tokens"
}
```

`ResponseUsage`

类型描述

当前响应的输入和输出 Token 用量。

类型参数

参数	类型	说明
total_tokens	integer	输入与输出 Token 总数
input_tokens	integer	输入 Token 数
output_tokens	integer	输出 Token 数
input_token_details	object	输入文本、音频及缓存 Token 明细
output_token_details	object	输出文本和音频 Token 明细

明细对象可包含 `text_tokens`、`audio_tokens` 和 `cached_tokens`。

示例

```
{
 "total_tokens": 43,
 "input_tokens": 25,
 "output_tokens": 18,
 "input_token_details": {
   "text_tokens": 10,
   "audio_tokens": 15,
   "cached_tokens": 0
 },
 "output_token_details": {
   "text_tokens": 8,
   "audio_tokens": 10
 }
}
```

`Tool`

类型描述

声明可供模型使用的函数、联网搜索或知识库检索（RAG）工具。

类型参数

函数工具：

参数	类型	必填	说明
type	string	是	固定为function
name	string	是	函数名称
description	string	否	函数用途说明
parameters	object	否	JSON Schema 格式的参数定义

联网搜索工具：

参数	类型	必填	说明
type	string	是	固定为web_search

示例

```
{
 "type": "function",
 "name": "get_weather",
 "description": "查询指定城市的天气",
 "parameters": {
   "type": "object",
   "properties": {
     "city": {"type": "string", "description": "城市名称"}
   },
   "required": ["city"]
 }
}
```

知识库检索工具（file_search）：

注意：使用该功能需在千帆平台预先创建知识库并上传文档，API Key 需具备千帆知识库的读取权限（ReadKnowledgeBase）。

参数	类型	必填	说明
type	string	是	固定为 file_search
vector_store_ids	string[]	是	千帆知识库 ID 列表；至少 1 个，元素须为非空字符串、不重复，且须在认证 API Key 可访问的知识库范围内

file_search 校验规则

规则	错误码	说明
当前模型不支持 file_search	invalid_value	提示 "Invalid value: 'file_search'. Supported values are: 'function', ..."
file_search 功能不可用	invalid_value	凭证不支持、权限拒绝或服务暂不可用时提交该工具，提示 "The file_search tool is not available: {status}."
vector_store_ids 必填	missing_required_parameter	未提供则报错
vector_store_ids 不可为空数组	empty_array	提供空数组则报错
每个 vector_store_id 必须非空	empty_string	数组中存在空字符串则报错
vector_store_id 须在用户可访问范围内	invalid_value	ID 不存在或无权限则报错，提示 "Not found or permission denied."
vector_store_id 不可重复	invalid_value	重复 ID 提示 "Duplicate vector_store_id"
name 不允许自定义	unknown_parameter	file_search 工具无名称
description 不允许自定义	unknown_parameter	同上
parameters 不允许自定义	unknown_parameter	同上

示例

```
{
 "type": "file_search",
 "vector_store_ids": ["kb-abc123", "kb-def456"]
}
```

`Error`

类型描述

服务端错误详情。

类型参数

参数	类型	说明
type	string	错误类型，例如invalid_request_error 或 server_error
code	string	错误码
message	string	可读错误信息
param	string	引发错误的参数路径
event_id	string	对应的客户端事件 ID

示例

```
{
 "type": "invalid_request_error",
 "code": "invalid_value",
 "message": "Invalid value for session.input_audio_format.",
 "param": "session.input_audio_format",
 "event_id": "event_001"
}
```

错误码汇总

接口通过 error 事件返回错误。常见错误类别如下：

错误类别或错误码	说明	建议处理方式
missing_model	连接未提供模型参数	在 URL 中添加有效的 model
model_not_found	模型不存在或当前接口不支持	检查模型名称
invalid_request_error	事件结构或参数无效	根据 param 和 message 修正请求
invalid_value	参数值不在支持范围	使用文档列出的枚举或范围
missing_required_parameter	缺少必填参数	补充错误信息指出的参数
unsupported_event	当前实现不支持该客户端事件	改用本文档列出的客户端事件
server_error / internal_error	服务端或下游处理异常	保留 event_id，按业务策略重试或重新连接
empty_array	参数不能为空数组	传入非空数组（如 vector_store_ids）
empty_string	数组元素不能为空字符串	传入非空字符串（如 vector_store_ids 中的元素）
unknown_parameter	包含未支持的参数	移除多余参数（如 file_search 工具不允许自定义 name/description/parameters）

鉴权阶段还可能通过 WebSocket 关闭码结束连接：

错误码	描述
3000	未授权
3003	禁止访问
3008	连接超时
4000	无效请求
1013	配额超限

附录：

DEMO

python

> 通过iam API_KEY调用时需要删除代码中的"&access_token={TOKEN}"。

realime-ws-demo-flex

端到端语音语言大模型Android SDK

1. 文档简介

1.1 文档说明

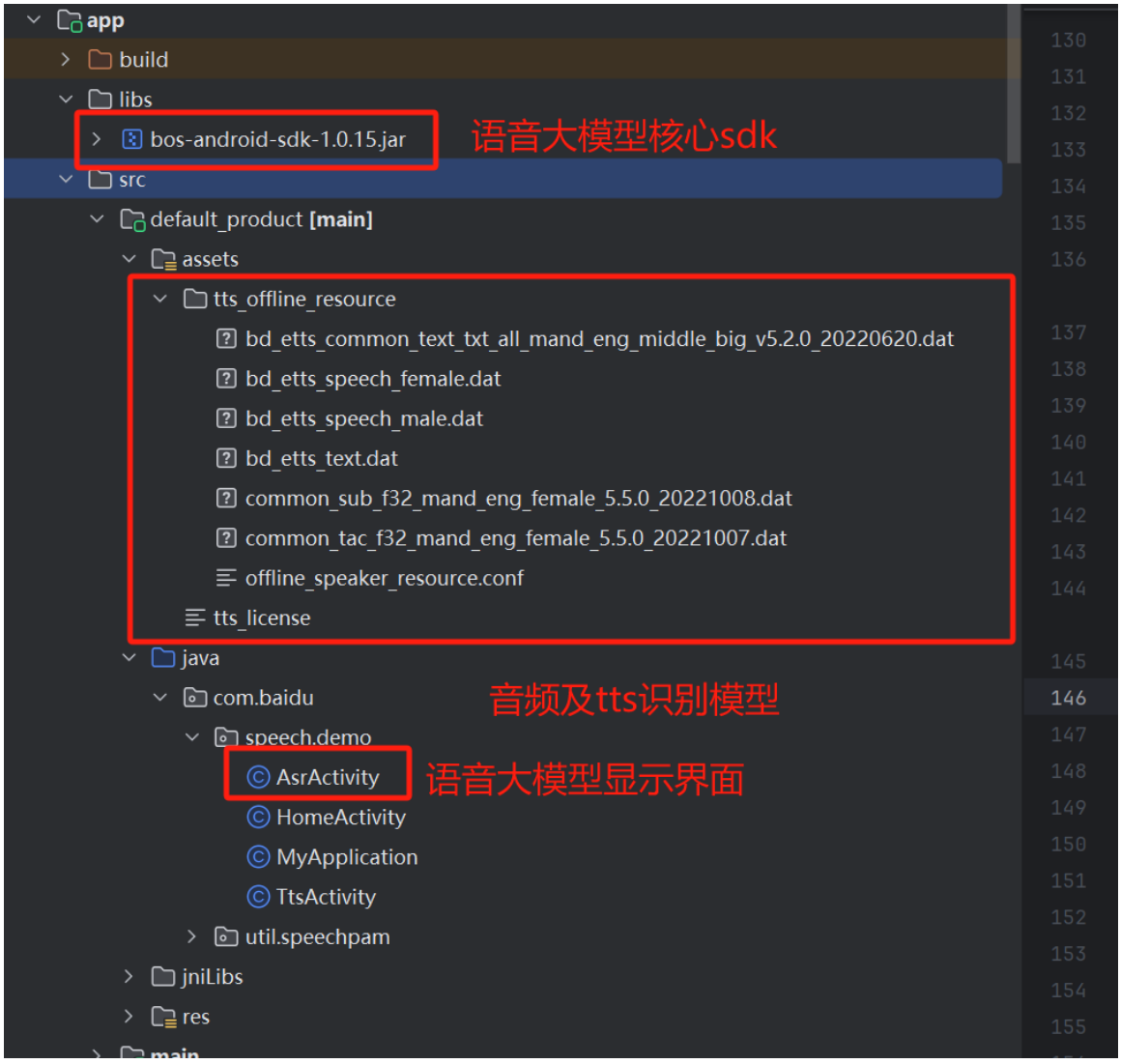
文档名称	端到端语音语言大模型集成文档
所属平台	Android
提交日期	2025-05-15
概述	本文档是百度语音开放平台Android SDK的用户指南，描述了端到端语音语言大模型相关接口的使用说明。
获取安装包	端到端语音语言大模型Android SDK

2 开发准备工作

2.1 环境准备

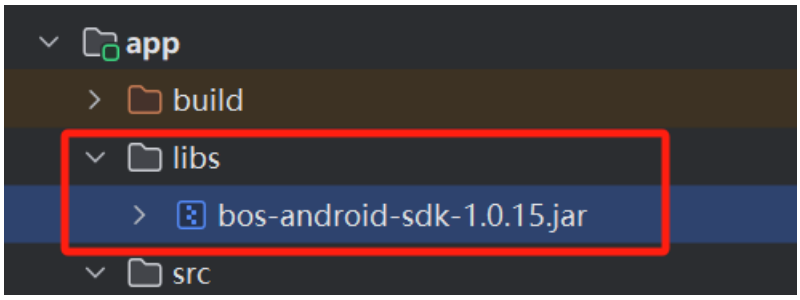
- 系统支持：Android 6.0+
- 架构支持：armeabi-v7a，arm64-v8a
- 硬件要求：要求设备上有麦克风
- 网络：支持移动网络（不包括2G、3G）、WIFI等网络环境
- 开发环境：建议使用最新版本Android Studio 进行开发
- 环境要求：gradle 3.0+，Java jdk 1.8+

2.2 SDK目录结构

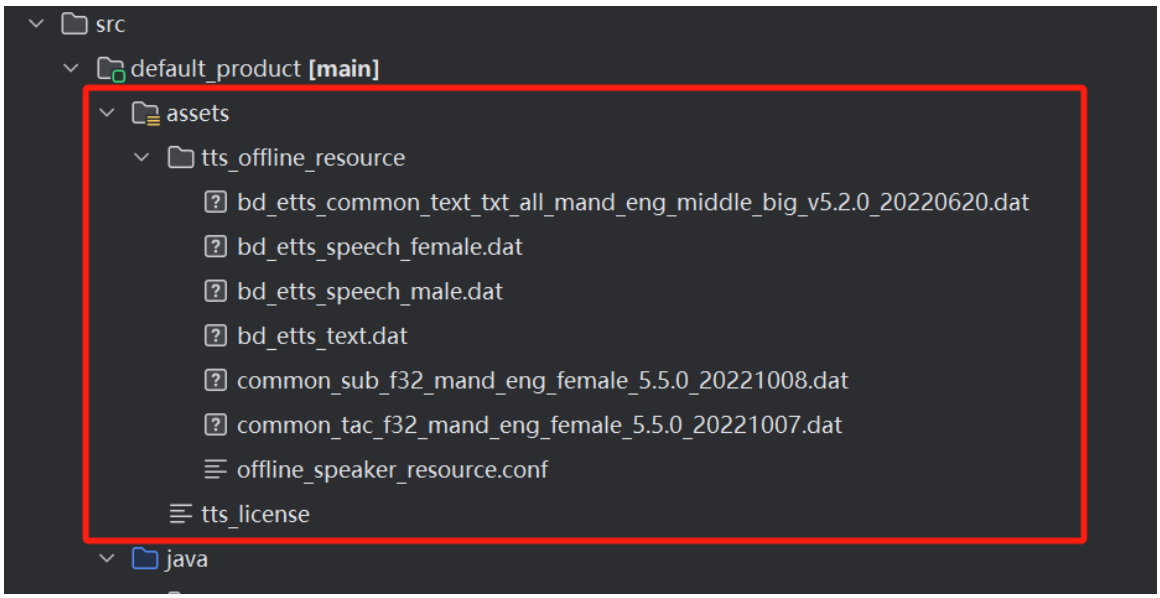


2.3 SDK安装

- 将libs/bos-android-sdk-xxx.jar复制到您的项目的同名目录中



- 添加模型文件



- 在build.gradle中添加依赖

```
 implementation fileTree(dir: 'libs', include: ['*.jar', '*.aar'])
implementation 'com.baidu.speech.asr_tts:bdspeech:1.0.0.3a98445'
```

2.4 鉴权方式

2.4.1 access_token鉴权机制

*** 获取AK/SK**

请参考 通用参考 - 鉴权认证机制 | 百度AI开放平台中的access_token鉴权机制获取AK/SK, 并得到AppID、API Key、Secret Key三个信息

- 在START开始语音识别时，传入认证信息。详细参数参考**"开始识别"**

```
 // ... 其他参数已忽略
map.put(SpeechConstant.APP_ID, APP_ID); // 添加appId
map.put(SpeechConstant.APP_API_KEY, APP_KEY); // 添加apiKey
SpeechEventManager.startAsr(this, new JSONObject(map), this);
```

2.4.2 API Key鉴权机制

注意： 邀测阶段暂时仅支持access_token鉴权机制

3. SDK集成

3.1 功能接口

SDK中主要的类和接口如下：

- SpeechEventManager：语音事件管理类，用于管理语音识别、语音合成等事件。
- SpeechSynthesizer: 语音合成类，用于管理语音合成和播放
- JsonUtil：json工具类，帮助组装json请求参数
- SpeechConstant: 包含语音识别、语音合成等参数的key常量
- TtsAudioInfoEntity：用于播放音频的类
- IEventListener: 事件监听器接口，用于处理语音识别过程中产生的各种事件。
- SpeechSynthesizerListener: 语音合成监听器接口，用于处理语音合成过程中产生的各种事件
- SynthesizerResponse：语音合成的响应结构，SpeechSynthesizerListener回调函数中接收到的输入参数

3.1.1 SpeechEventManager：语音识别操作类

- 方法列表
  - initSDK: 初始化SDK
    - 功能说明：在应用启动后执行一次，不可重复调用。
    - 输入参数
      - context: Context， 上下文信息
      - sdkConfig: SpeechEventManager.SDKConfig， SDK配置信息， 包括3部分：
        - asr参数：通过setAsrParamMap设置, 可用参数见下表：

参数名	类型	是否必须	说明
SpeechConstant.PID	String	是	识别环境ID，1843
SpeechConstant.DEVICE_ID	String	是	设备唯一id，通过getSpeechDeviceId获取

- 其他透传参数：通过setExternalMap设置, 可用参数见下表：

参数名	类型	是否必须	说明
SpeechConstant.LOCATION	string	是	位置信息
SpeechConstant.APP_VERSION	string	是	版本号
SpeechConstant.DEVICE_ID	string	是	设备唯一id，通过getSpeechDeviceId获取

startAsr: 启动识别

- 功能说明：开启一次识别,注意不要多次调用，下次调用需要在CALLBACK_EVENT_ASR_EXIT回调 或 调用停止接口后调用 *输入参数* context: `Context`， 上下文信息 * jsonObject: `JSONObject`， 通过json传递的参数， 具体参数请参考下表

参数名	类型	是否必须	说明
PID	string	是	识别环境 id，1843
URL	string	是	识别环境 url, 默认值`https://vop.baidu.com/v2`
APP_ID	string	是	开放平台创建应用后分配的鉴权信息,上线后请使用此参数填写鉴权信息。参考**"2.4 鉴权方式"**
APP_API_KEY	string	是	开放平台创建应用后分配的鉴权信息,上线后请使用此参数填写鉴权信息。参考**"2.4 鉴权方式"**
ASR_AUDIO_COMPRESSION_TYPE	string	是	必传，音频压缩类型, 默认值：OPUS.（暂不支持其他类型）
ASR_MUTIPLY_MODE	string	否	json字符串，格式为：scene_id：识别类型(1: 点按识别； 2: 唤醒后识别；4：长按识别，参考**"4.2 开发场景"** 配置 默认为1)
ASR_ENABLE_MUTIPLY_SPEECH	string	否	是否启动全双工 0：单工，1：全双工 ， 默认为0
ASR_VAD_RES_PATH	string	否	vad 资源路径，"初始化环境"时拷贝的资源路径
NET_TYPE	string	否	网络类型；1：有turbonet的http（场内网络库）；3：无turbonet的http. 默认值：1
LOG_LEVEL	string	否	设置日志等级，可选值：LogUtil.VERBOSE ，LogUtil.DEBUG，LogUtil.INFO，LogUtil.WARN，LogUtil.ERROR，LogUtil.OFF

- listener: IEventListener， 事件监听器, 需要实现`IEventListener`接口, 详情请参考IEventListener

stopASR: 停止语音识别

- 功能说明：停止当前语音识别，并保留当前的识别结果，已经发送的音频会被正常处理。
- 输入参数：无
- 返回值： 无

exitASR: 退出语音识别

- 功能说明：退出语音识别，并停止后续处理，已经发送未处理的音频不会继续处理。
- 输入参数：无
- 返回值： 无

pauseAsr: 暂停语音识别

- 功能说明：暂停语音识别 双工长语音功能可以用到，目前只会停止语音识别功能，需记录语音识别最后一条sn，该sn后面的语音都会进行暂停
- 输入参数：无
- 返回值： 无

resumeAsr: 恢复语音识别

- 功能说明：恢复语音识别
- 输入参数：无
- 返回值： 无

getSpeechDeviceId: 获取设备ID

- 输入参数：无
- 返回值：String, 设备唯一ID

3.1.2 IEventListener：语音识别事件监听 用户需要实现此接口，并在启动识别时传入，以监听语音识别过程中产生的事件。

- 方法列表
  - onEvent: 接收识别过程中产生的回调事件
    - 输入参数
      - name: `string`， 事件名称， 具体事件请参考下表
      - params: `string`， 事件参数。根据事件名称的不同，参数内容不同。详细请参考下表
      - data: `byte[]`, 事件中携带的二进制数据，例如语音合成产生的音频数据。
      - offset:`int`, 暂未使用
      - length: `int`, data中数据的长度
    - 返回值
      - 无

事件名称	事件参数（params）	说明
CALLBACK_EVENT_ASR_READY	无	准备就绪，可以说话，一般在收到此事件后通过UI通知用户可以说话了
CALLBACK_EVENT_ASR_BEGIN	无	检测到开始说话。
CALLBACK_EVENT_ASR_PARTIAL	返回json字符串，格式为： result_type: 结果类型，可选值: partial_result: asr识别中间结果; final_result： asr识别最终结果; third_result：第三方数据，例如联网搜索结果，暂不支持。 best_result: 识别结果。 sn: 本次会话的sn	语音识别中间结果返回
CALLBACK_EVENT_ASR_EXIT		退出识别
CALLBACK_EVENT_ASR_END		说话结束等待识别结果
CALLBACK_EVENT_ASR_FINISH		识别完成（这里指的是用户的语音识别完成）
CALLBACK_EVENT_ASR_CANCEL		用户取消识别
CALLBACK_EVENT_ASR_PAUSE		识别暂停
CALLBACK_EVENT_ASR_RESUME		识别恢复
CALLBACK_ASR_TTS_RESULT	返回json字符串，格式为： result_type: 结果类型，可选值： tts_result：tts合成结果 origin_result：合成结果，内容是一个json，包含以下字段: aue: 音频格式。 idx：当前片段在整个音频中的索引 percent: 当前合成进度百分比 samplerate：采样率 tex: 音频对应的文本 sn: 本次会话的sn	
TTS 数据返回		
CALLBACK_EVENT_TTS_FIRST_PLAYED		tts开始播放 （使用SpeechSynthesizer不会调用这个事件）
CALLBACK_EVENT_TTS_END_PLAYED		tts播放完成（使用SpeechSynthesizer不会调用这个事件）
CALLBACK_EVENT_ASR_VOLUME		音量回调

3.1.3 SpeechSynthesizer：语音合成操作类 方法列表

setSpeechSynthesizerListener：设置回调函数

- 功能说明：设置语音合成回调函数
- 输入参数：listener: `SpeechSynthesizerListener`， 语音合成监听器, 需要实现`SpeechSynthesizerListener`接口, 详情请参考**SpeechSynthesizerListener**
- 返回值： 无

loadAudioPlayer: 加载音频播放器

- 功能说明：加载音频播放器
- 输入参数：无
- 返回值： 无

3.1.4 SpeechSynthesizerListener：语音合成事件监听 用户需要实现此接口，以监听音频播放器的播放事件 **方法列表**
onSynthesizeResponse

- 功能说明：语音合成的回调函数
- 输入参数：synthesizerResponse: `SynthesizerResponse`， 语音合成响应.
- 返回值： 无

3.1.5 SynthesizerResponse SpeechSynthesizerListener.onSynthesizeResponse的输入参数类型
方法列表
getSynthesizeType：当前时间类型

- 返回值：`SynthesizeType`, 枚举值，可选值包括：
  - PLAY_START：开始播放
  - PLAY_FINISH：播放结束
  - PLAY_PROGRESS: 播放进度上报

getSynthesizerData：获取语音合成数据

- 返回值： `SynthesizerData`, 语音合成数据, 有以下方法可用：
  - getAudioProgress：获取播放进度， 返回值类型：`int`
  - getEngineType：获取引擎类型， 返回值类型：`int`
  - getAudioSampleRate：获取音频采样率， 返回值类型：`int`
  - getAudioData：获取音频数据, 返回值类型：`byte[]`

3.2 集成步骤

3.2.1 初始化环境 包括几个步骤：

- 将VAD模型文件和AEC_ALGO模型文件拷贝到设备的目录下

```
 private void initialEnv() {
        if (mSampleDirPath == null) {
            String sdcardPath = this.getFilesDir().getPath();
            mSampleDirPath = sdcardPath + "/" + SAMPLE_DIR_NAME;
        }
        makeDir(mSampleDirPath);

        pathVadRes = mSampleDirPath + File.separator + VAD_MODEL;
        copyFromAssets(this, true, VAD_MODEL, pathVadRes);

        pathAecRes = mSampleDirPath + File.separator + SpeechConstant.AEC_ALGO_MODEL;
        copyFromAssets(this, true, SpeechConstant.AEC_ALGO_MODEL, pathAecRes);
        ConfigUtil.setAecModelFile(pathAecRes);
    }
```

- 申请需要的权限
    - 需要的权限列表：

权限	说明	是否必须
android.permission.INTERNET	允许访问网络	是
android.permission.RECORD_AUDIO	允许程序录制声音通过手机或耳机的麦克	是
android.permission.WRITE_EXTERNAL_STORAGE	外置卡读写权限	是
android.permission.READ_PHONE_STATE	允许获取设备信息权限	否

- 示例代码

```
 /**
 * 请求权限
 *
 * @return 需要请求的权限数
 */
protected int requestPermissions() {
    String[] permissionArray = checkPermissions();
    if (permissionArray != null && permissionArray.length > 0) {
        ActivityCompat.requestPermissions(this, permissionArray, CODE_REQUEST_PERMISSION);
        return permissionArray.length;
    }
    return 0;
}

protected String[] checkPermissions() {
    String[] noGrantedArr = {};
    List<String> noGrantedList = new ArrayList<>();
    for (String item : sPermissions) {
        int hasWriteStoragePermission = ContextCompat.checkSelfPermission(getApplication(), item);
        if (hasWriteStoragePermission != PackageManager.PERMISSION_GRANTED) {  // 未拥有权限
            noGrantedList.add(item);
        }
    }
    return noGrantedList.toArray(noGrantedArr);
```

3.2.2 实现事件回调函数

- 语音识别回调：即实现IEventListener接口， 示例代码如下：详细参数说明参考**IEventListener**

```
 @Override
public void onEvent(String name, String params, byte[] data, int offset, int length) {
```

```
    String msg = "name: " + name + ", params: " + params;
    switch (name) {
        case SpeechConstant.CALLBACK_EVENT_ASR_READY: //准备就绪，可以说话，一般在收到此事件后通过UI通知用户可以说话了
        case SpeechConstant.CALLBACK_EVENT_ASR_BEGIN: //检测到开始说话
            appendToView(mTvLog, msg);
            break;
        case SpeechConstant.CALLBACK_EVENT_ASR_PARTIAL: //中间结果返回
            logD(TAG, "SpeechConstant.CALLBACK_EVENT_ASR_PARTIAL" +params);
            JSONObject partialJson = JsonUtil.toJSON(params);
            String resultType = partialJson.optString("result_type");
            String bestResult = "";
            String showAsr = "中间识别结果";
            if ("partial_result".equals(resultType)) {
                bestResult = partialJson.optString("best_result");
            } else if ("final_result".equals(resultType)) {
                bestResult = partialJson.optString("best_result");
                logD(TAG, "------" + params + ", " + mLastFinalSn);
                JSONObject json = JsonUtil.toJSON(params);
                mLastFinalSn = json.optString("sn");
                showAsr = "最终识别结果";
            } else if ("third_result".equals(resultType)) {
                Log.d(TAG, "third_result: " + resultType);
            }
            if (!TextUtils.isEmpty(bestResult)) {
                String resultInfo = showAsr + ": " + bestResult;
                printToView(mTvResult, resultInfo);
            }
            break;
        case SpeechConstant.CALLBACK_EVENT_ASR_EXIT: //退出
            appendToView(mTvLog, msg);
            handleAsrExit();
            break;
        case SpeechConstant.CALLBACK_EVENT_ASR_END: //说话结束等待识别结果
        case SpeechConstant.CALLBACK_EVENT_ASR_FINISH: //识别完成
        case SpeechConstant.CALLBACK_EVENT_ASR_CANCEL: //识别已取消
        case SpeechConstant.CALLBACK_EVENT_ASR_PAUSE:
        case SpeechConstant.CALLBACK_EVENT_ASR_RESUME:
            break;
        case SpeechConstant.CALLBACK_ASR_TTS_RESULT: //TTS 数据返回
            logD(TAG, "CALLBACK_ASR_TTS_RESULT" + params);
            JSONObject ttsJson = JsonUtil.toJSON(params);
            String origin_result =ttsJson.optString("origin_result");
            JSONObject tts_origin_result =JsonUtil.toJSON(origin_result);
            String text = tts_origin_result.optString("tex");
            int idx = tts_origin_result.optInt("idx", 0);
            break;
        case SpeechConstant.CALLBACK_EVENT_TTS_FIRST_PLAYED: // tts 开始播放
            appendToView(mTvLog, msg);
            Log.d(TAG, "CALLBACK_EVENT_TTS_FIRST_PLAYED: " + msg);
            break;
        case SpeechConstant.CALLBACK_EVENT_TTS_END_PLAYED: // tts 播放完成
            Log.d(TAG, "CALLBACK_EVENT_TTS_END_PLAYED: " + msg);
            break;
        case SpeechConstant.CALLBACK_EVENT_UPLOAD_FINISH: //上传完成
        case SpeechConstant.CALLBACK_EVENT_ASR_AUDIO:
        case SpeechConstant.CALLBACK_EVENT_ASR_VOLUME: //音量回调
            break;
        }
    }
```

- 语音合成回调：即实现SpeechSynthesizerListener接口， 示例代码如下：详细参数说明参考**SpeechSynthesizerListener**

```
 @Override
public void onSynthesizeResponse(SynthesizerResponse synthesizerResponse) {
   SynthesizerResponse.SynthesizeType synthesizeType = synthesizerResponse.getSynthesizeType();
    String sn = synthesizerResponse.getSn();
    String utteranceId = synthesizerResponse.getUtteranceId();
    String instanceId = synthesizerResponse.getInstanceId();
    switch (synthesizeType) {
        case PLAY_START:
        case PLAY_FINISH:
            LoggerProxy.d(TAG, "onSynthesizeResult type =  " + synthesizeType.name());
            break;
        case PLAY_PROGRESS:
            SynthesizerResponse.SynthesizerData synthesizerData =
                    synthesizerResponse.getSynthesizerData();
            int progress = synthesizerData.getAudioProgress();
            int engineType = synthesizerData.getEngineType();
            int sampleRate = synthesizerData.getAudioSampleRate();
            byte[] audioData = synthesizerData.getAudioData();
            LoggerProxy.d(TAG, "onSynthesizeResult type =  " + synthesizeType.name()
                    + " sn = " + sn
                    + " , instanceId = " + instanceId
                    + " , progress = " + progress
                    + " , engineType = " + engineType + " , sampleRate = " + sampleRate);
            break;
    }

}
```

3.2.3 初始化SDK

1. 仅保留日志等级设置，其他在Builder中设置默认值
2. PID和IOS一样在startAsr中设置

- 示例代码

```
 SpeechEventManager.SDKConfig.Builder builder = new
SpeechEventManager.SDKConfig.Builder();
HashMap<String, String> asrParamMap = new HashMap<>();
asrParamMap.put(SpeechConstant.PID, "8533");
asrParamMap.put(SpeechConstant.DEVICE_ID, mCuid);
asrParamMap.put(SpeechConstant.LOG_LEVEL,
String.valueOf(LogUtil.VERBOSE));

HashMap<String, String> externalMap = new HashMap<>();
externalMap.put(SpeechConstant.LOCATION, "Beijing");
externalMap.put(SpeechConstant.APP_VERSION, "1.0.0");
externalMap.put(SpeechConstant.DEVICE_ID, mCuid);

SpeechEventManager.SDKConfig sdkConfig = builder
        .setAsrParamMap(asrParamMap)
        .setExternalMap(externalMap)
        .build();
int ret = SpeechEventManager.initSDK(this, sdkConfig);
Log.d(TAG, "initSDK: ret = " + ret);
```

3.2.4 启动识别

- 示例代码

```
 HashMap<String, Object> asrParams = new HashMap<>();

// 环境信息设置：
// 识别环境 - pid. 邀测环境PID统一为4144779，仅供体验使用。生产环境使用请联系技术支持获取专属PID
asrParams.put(SpeechConstant.PID, 1860);
// 识别环境 - url
asrParams.put(SpeechConstant.URL, "https://vse.baidu.com/v2");

// 必选参数设置
// 设置VAD/TriggerMode/双工，请参考**开发场景**根据您的业务需求设置。下面示例为双工识别场景。
// 开启VAD
asrParams.put(SpeechConstant.VAD, SpeechConstant.VAD_DOAVAD);
// 是否启动双工
asrParams.put(SpeechConstant.ASR_ENABLE_MUTIPLY_SPEECH, 1);
asrParams.put(SpeechConstant.TRIGGER_MODE, triggerMode); // trigger 1: 点按识别； 2: 唤醒后识别，4：长按识别

// 必选：设置音频压缩类型，
asrParams.put(SpeechConstant.ASR_AUDIO_COMPRESSION_TYPE, SpeechConstant.OPUS);
// 是否返回识别中间结果
asrParams.put(SpeechConstant.ENABLE_EARLY_RETURN, true);
asrParams.put(SpeechConstant.ASR_VAD_RES_PATH, pathVadRes); // vad资源路径，上文拷贝资源的路径
asrParams.put(SpeechConstant.NET_TYPE, 1); // 网络类型；1：有turbonet的http；3：无turbonet的http
asrParams.put(SpeechConstant.LOG_LEVEL, LogUtil.VERBOSE); // 设置日志等级，Debug 使用
asrParams.put(SpeechConstant.PAM,
// 第一个参数为DEVICE_ID, 第二个参数为BDVS_ID.
JsonUtil.genAsrPam("75CEB7CF757E522BAEA2E94BCBC676E5|VUGJPCGTI",
"68b509484f249601628e7c1b547a003c", null)); // bdvs 协议参数
SpeechEventManager.startAsr(this, new JSONObject(asrParams), this);
```

3.2.5 停止识别 停止识别，但保留当前识别结果。已经发送的音频会正常识别并生成响应音频

- 示例代码

```
 SpeechEventManager.stopASR();
```

3.2.6 取消识别 取消识别，并停止后续处理。已经发送但是还没有识别和响应的数据将会丢弃。

- 示例代码

```
 SpeechEventManager.exitASR();
```

3.2.7 暂停识别

```
 Map<String, Object> params = new HashMap<>();
LogUtil.d(TAG, "pauseAsr lastSn: " + lastSn); // 该 sn 之后的识别都会被取消
params.put(SpeechConstant.KEY_LAST_FINAL_RESULT_SN, lastSn);
SpeechEventManager.pauseASR(new JSONObject(params));
```

3.2.8 恢复识别

- 示例代码

```
 SpeechEventManager.resumeASR();
```

3.3 开发场景

SDK支持三种开发场景：

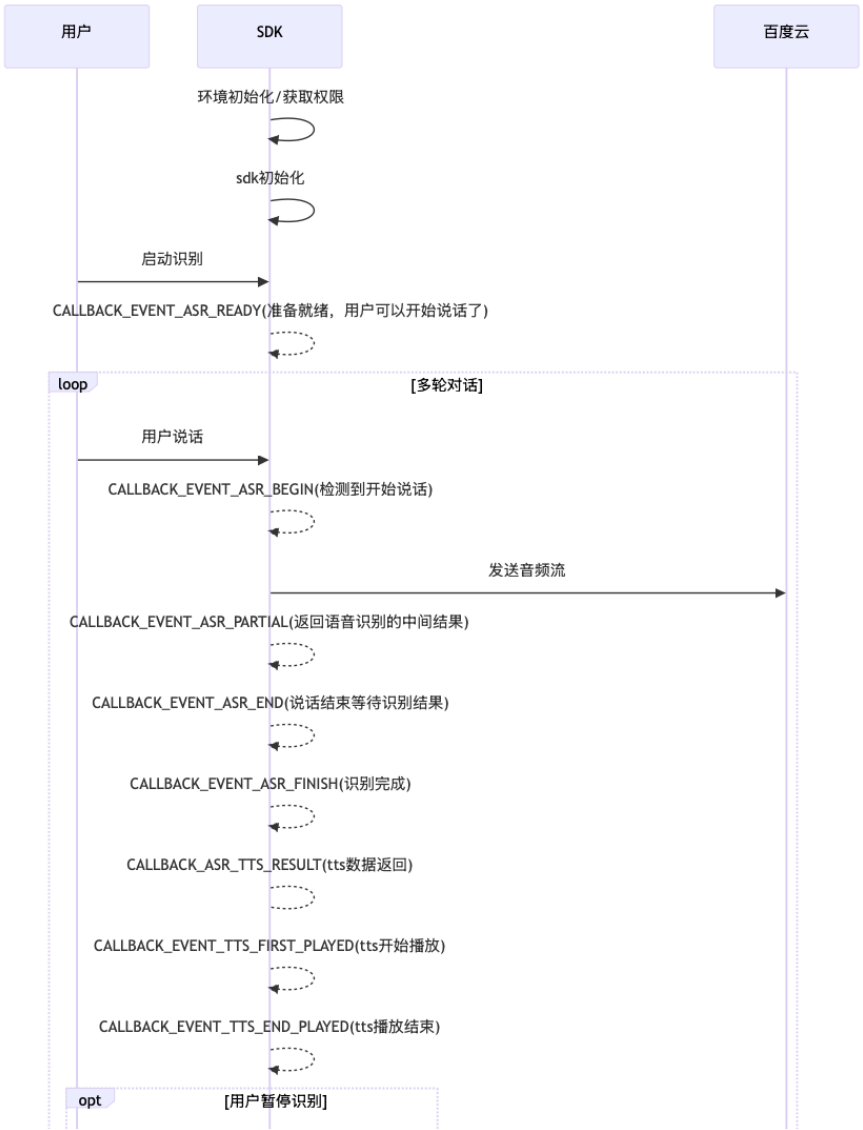
- 双工识别：启动后可以多次进行语音对话，直到用户主动停止识别
- 点按短语音识别：启动后进行60s内语音识别，识别到的第一个短句进行语音对话。
- 长按识别：启动后按住按钮不松开会持续识别，且不会进行断句

3.3.1 双工识别

- 设置方法

```
 // .... 其他参数已省略
map.put(SpeechConstant.VAD, SpeechConstant.VAD_DOAVAD);
// 双工识别
map.put(SpeechConstant.ASR_ENABLE_MUTIPLY_SPEECH, 1);
// TRIGGER_MODE:2, 唤醒后识别
map.put(SpeechConstant.TRIGGER_MODE, 2);
JSONObject jsonObject;
jsonObject = new JSONObject(map);
SpeechEventManager.startAsr(this.getApplicationContext(), jsonObject, this);
```

- 交互流程



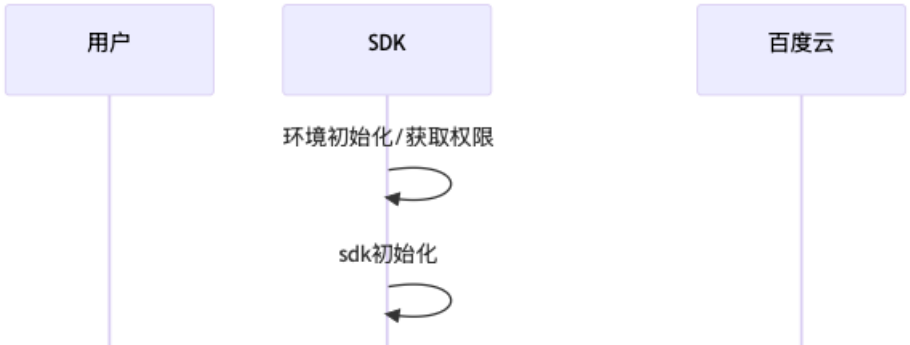


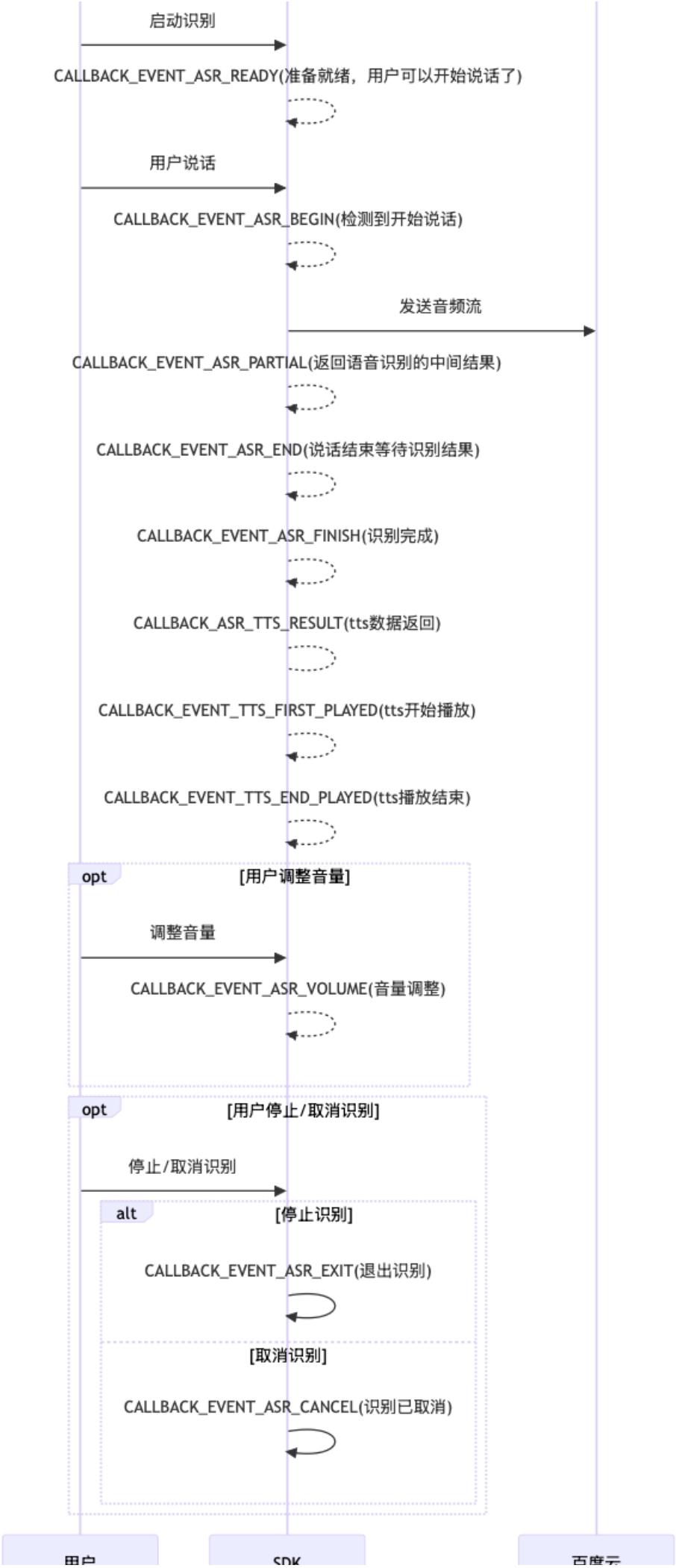
3.3.2 点按短语音识别

- 设置方法

```
 // .... 其他参数已省略
map.put(SpeechConstant.VAD, SpeechConstant.VAD_DOAVAD);
// TRIGGER_MODE:1 短按识别
map.put(SpeechConstant.TRIGGER_MODE, 1)
JSONObject jsonObject;
jsonObject = new JSONObject(map);
SpeechEventManager.startAsr(this.getApplicationContext(), jsonObject, this);
```

- 交互流程




启动识别
CALLBACK_EVENT_ASR_READY(准备就绪，用户可以开始说话了)
用户说话
CALLBACK_EVENT_ASR_BEGIN(检测到开始说话)
发送音频流
CALLBACK_EVENT_ASR_PARTIAL(返回语音识别的中间结果)
CALLBACK_EVENT_ASR_END(说话结束等待识别结果)
CALLBACK_EVENT_ASR_FINISH(识别完成)
CALLBACK_ASR_TTS_RESULT(tts数据返回)
CALLBACK_EVENT_TTS_FIRST_PLAYED(tts开始播放)
CALLBACK_EVENT_TTS_END_PLAYED(tts播放结束)
opt
[用户调整音量]
调整音量
CALLBACK_EVENT_ASR_VOLUME(音量调整)
opt
[用户停止/取消识别]
停止/取消识别
alt
[停止识别]
CALLBACK_EVENT_ASR_EXIT(退出识别)
[取消识别]
CALLBACK_EVENT_ASR_CANCEL(识别已取消)
用户
SDK
百度云

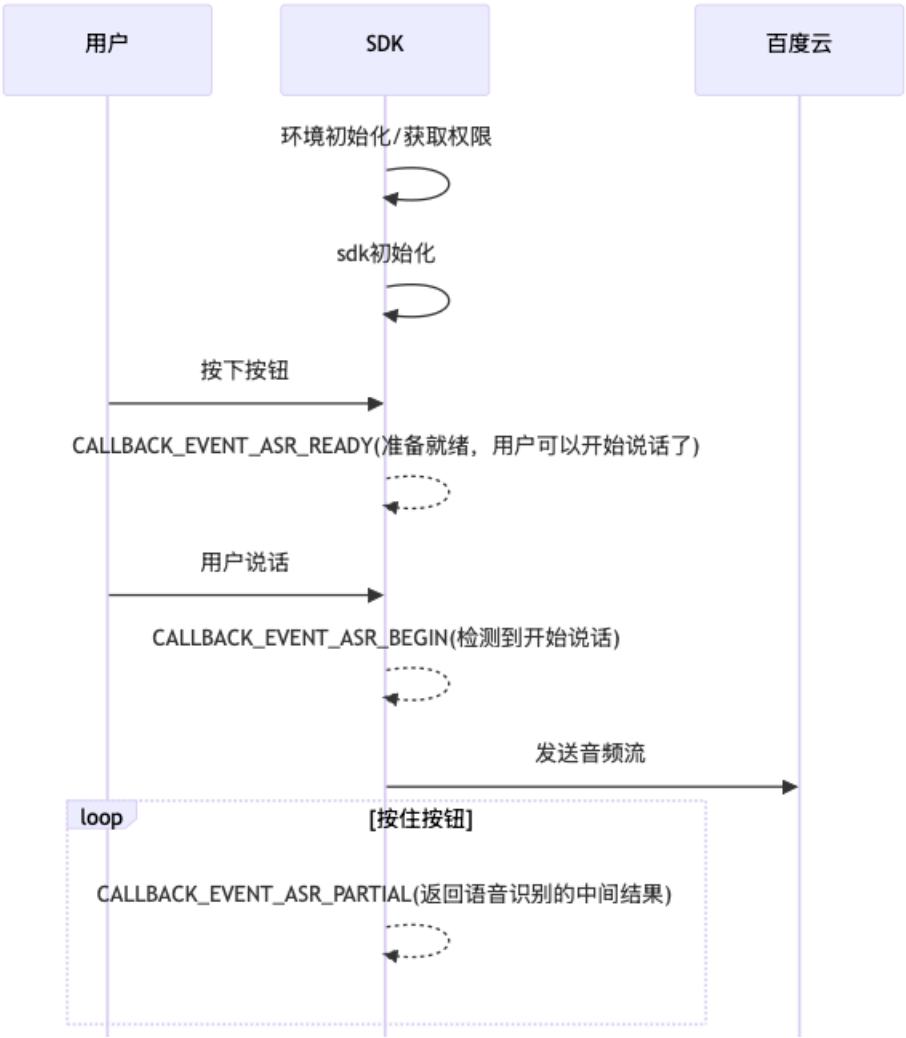
3.3.3 长按识别

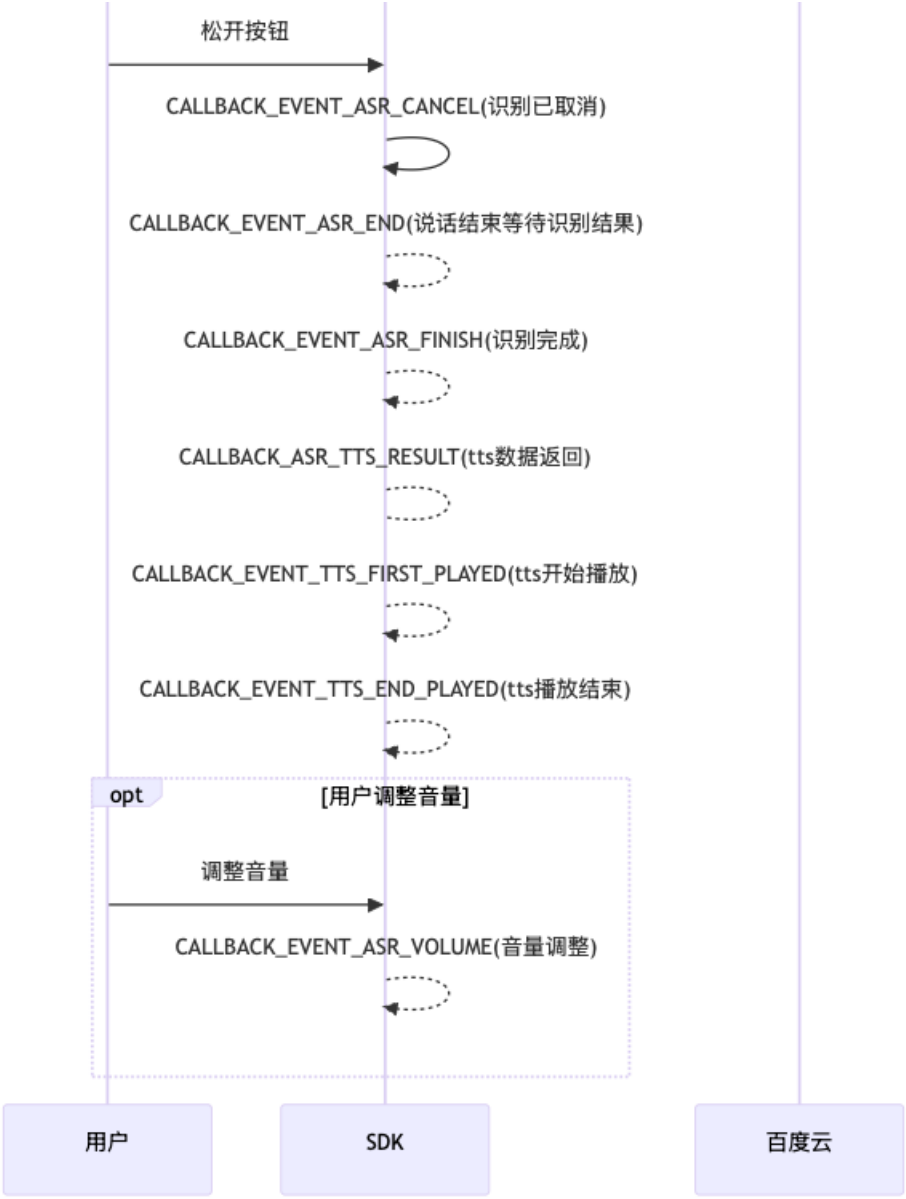
- 设置方法

```
 // .... 其他参数已省略
// 在按下时startAsr
asrParams.put(SpeechConstant.VAD, SpeechConstant.VAD_TOUCH);
// TRIGGER_MODE:4 长按识别模式
asrParams.put(SpeechConstant.TRIGGER_MODE, 4)
JSONObject jsonObject;
jsonObject = new JSONObject(asrParams);
SpeechEventManager.startAsr(this.getApplicationContext(), jsonObject, this);

// 在松开时stopAsr
SpeechEventManager.stopAsr();
```

- 交互流程





3.4 错误码

错误领域	描述	错误码	错误描述及可能原因
1	网络超时		出现原因可能为网络已经连接但质量比较差，建议检测网络状态
		1000	DNS连接超时
		1001	网络连接超时（用于非chunk模式）
		1002	网络读取结果超时（用于非chunk模式）
		1003	上行网络连接超时（用于chunk模式）
		1004	上行网络读取结果超时（用于chunk模式）
		1005	下行网络连接超时（用于chunk模式）
		1006	下行网络读取结果超时（用于chunk模式）
2	其他网络错误（网络连接失败		出现原因可能是网络权限被禁用，或网络确实未连接，需要开启网络或检测无法联网的原因

		2000	网络连接失败
		2001	网络读取结果失败
		2002	上行网络连接失败
		2003	上行网络读取失败
		2004	下行网络连接失败
		2005	下行网络读取失败
		2006	下行数据异常
		2100	本地网络不可用
		2101	本地网络不可用
		2102	上行网络读取结果失败(代理模式)
		2103	下行网络连接失败(代理模式)
		2104	下行网络读取识别(代理模式)
		2105	下行数据异常(代理模式)
		2106	上行网络连接错误(代理模式)
		2107	请求未创建
3	音频错误		出现原因可能为：未声明录音权限，或 被安全软件限制，或 录音设备被占用，需要开发者检测权限声明。
		3000	音频异常
		3001	录音机打开失败
		3002	录音机参数错误
		3003	录音机不可用
		3006	录音机读取失败
		3007	录音机关闭失败
		3008	文件打开失败
		3009	文件读取失败
		3010	文件关闭失败
		3011	采样率错误
		3013	文件读完
		3100	VAD异常，通常是VAD资源设置不正确
		3101	长时间未检测到人说话，请重新识别
		3102	检测到人说话，但语音过短
4	服务端错误		出现原因可能是appid和appkey的鉴权失败
		4001	服务端参数错误（-3001）
		4002	服务端协议错误（-3002）
		4003	服务端识别错误（-3003）
		4004	服务端鉴权错误（-3004）
5	客户端错误		一般是开发阶段的调用错误，需要开发者检测调用逻辑或对照文档和demo进行修复。
		5001	客户端无法加载动态库
		5002	客户端识别参数有误
		5003	客户端获取token失败

		5004	客户端解析URL失败
		5005	客户端检测到非https URL
6	超时		语音过长，请配合语音识别的使用场景，如避开嘈杂的环境等
		6001	语音过长
7	没有匹配的识别结果		信噪比差，请配合语音识别的使用场景，如避开嘈杂的环境等
		7001	没有匹配的识别结果
		7002	识别结果为空
8	识别引擎繁忙		一般是开发阶段的调用错误，出现原因是上一个会话尚未结束，就让SDK开始下一次识别。SDK目前只支持单任务运行，即便创建多个实例，也只能有一个实例处于工作状态
		8001	识别引擎繁忙
9	缺少权限		参见demo中的权限设置
		9001	缺少权限
10	其他权限		
		10001	离线引擎异常
		10002	没有授权文件
		10003	授权文件不可用
		10004	参数设置错误
		10005	引擎没有被初始化
		10006	模型文件不可用
		10007	语法文件不可用
		10008	引擎重置失败
		10009	引擎初始化失败
		10010	引擎释放失败
		10011	引擎不支持
		10012	识别失败
		10013	引擎loading超时
		100013	异常统计
		11001	唤醒引擎异常
		11002	无授权文件
		11003	授权文件异常
		11004	唤醒异常
		11005	模型文件异常
		11006	引擎初始化失败
		11007	内存分配失败
		11008	引擎重置失败
		11009	引擎释放失败
		11010	引擎不支持该架构
		11011	无识别数据

端到端语音语言大模型iOS SDK

1. 文档简介

1.1 文档说明

文档名称	端到端语音语言大模型集成文档
所属平台	iOS
提交日期	2025-05-15
概述	本文档是百度语音开放平台iOS SDK的用户指南，描述了端到端语音语言大模型相关接口的使用说明。
获取安装包	端到端语音语言大模型iOS SDK

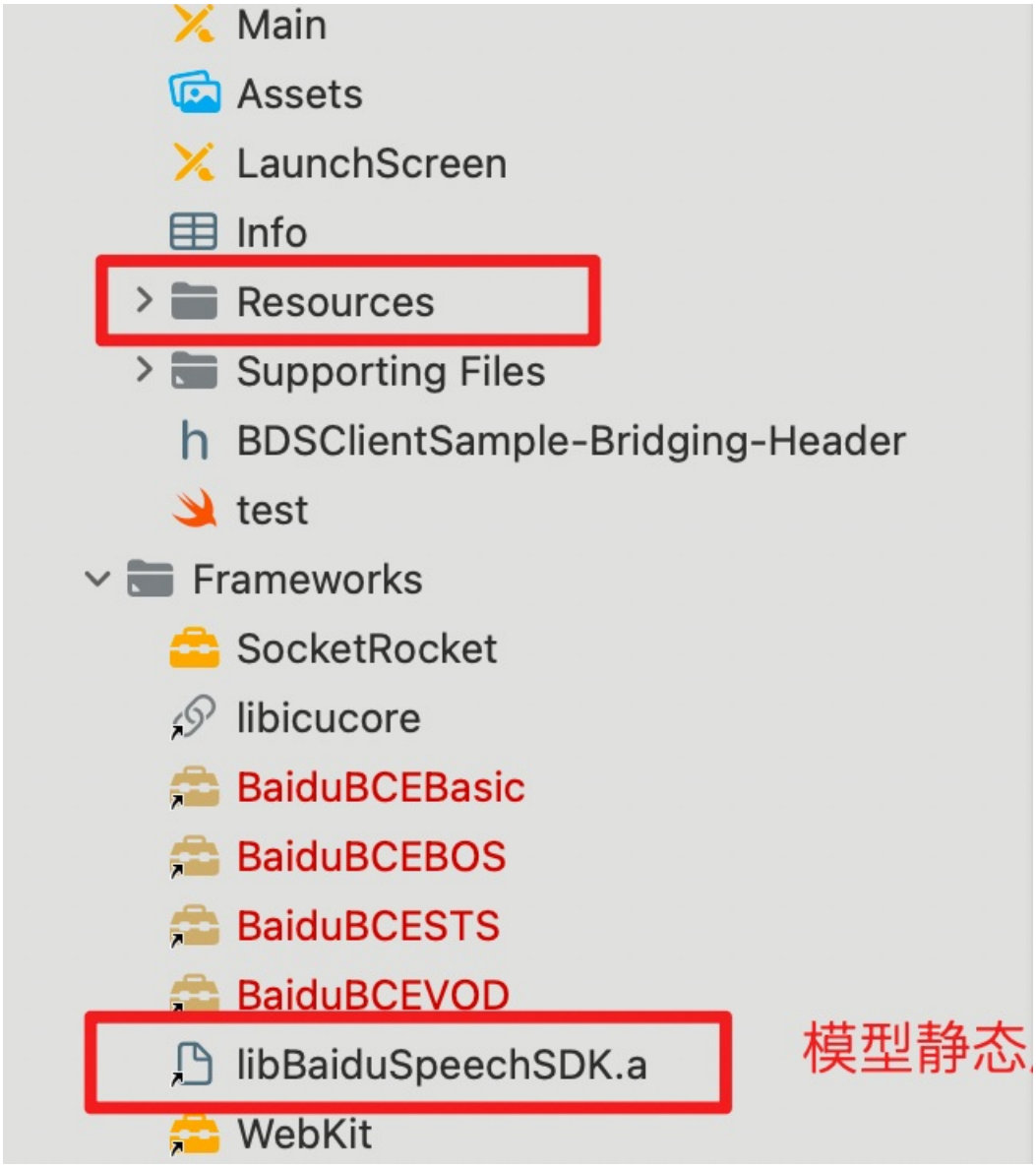
2. 开发准备工作

2.1 环境准备

名称	版本号
语音识别	3.3.2.3
系统支持	支持iOS 12.0及以上系统
架构支持	armv7、arm64、
开发环境	工程内使用了LTO等优化选项，建议使用最新版本Xcode进行开发
libBaiduSpeechSDK.a	端到端模型依赖静态库

2.2 SDK目录结构





2.3 SDK安装

- 获取安装包：端到端语音语言大模型iOS SDK
- 将BDSClientHeaders/ASR、BDSClientHeaders/TTS sdk头文件并添加libBaiduSpeechSDK.a静态库到您的项目中
- framework依赖以及系统必要依赖：

Framework	描述
libc++.tbd	提供对C/C++特性支持
libz.1.2.5.tbd	提供gzip支持
libsqlite3.0.tbd	提供对本地数据库的支持
AudioToolbox	提供录音和播放支持
AVFoundation	提供录音和播放支持
CFNetwork	提供对网络访问的支持
CoreLocation	提供对获取设备地理位置的支持，以提高识别准确度
CoreTelephony	提供对移动网络类型判断的支持
SystemConfiguration	提供对网络状态检测的支持
GLKit	内置识别控件所需
BaiduBCEBasic	基于BaiduBCE文件下的模型内部依赖库
BaiduBCEBOS	
BaiduBCESTS	
BaiduBCEVOD	
ZipArchive	合成tts功能依赖库
CuidSDK	

```
 // 参考demo中ASRViewController.mm类文件引入所需头文件
例：
##### import "ASRViewController.h"
##### import "BDSASRDefines.h"
##### import "BDSASRParameters.h"
##### import "BDSWakeupDefines.h"
##### import "BDSWakeupParameters.h"
##### import "BDSUploaderDefines.h"
##### import "BDSUploaderParameters.h"
##### import "BDSEventManager.h"
##### import "BDVRSettings.h"
##### import "fcntl.h"
##### import "AudioInputStream.h"
##### import "BDSSpeechSynthesizer.h"
##### import "TTDFileReader.h"
##### import "BDSCacheReferenceManager.h"
##### import "ASRPlayTestConfigViewController.h"
##### import "BDSOTADefines.h"
##### import "BDSOTAParameters.h"
##### import "BDSCompressionManager.h"
##### include <sys/socket.h>
##### include <netinet/in.h>
##### include <errno.h>
##### import <mach/mach.h>
##### import <AVFoundation/AVAudioSession.h>
```

2.4 鉴权方式

2.4.1 access_token鉴权机制

- 获取AK/SK

请参考通用参考 - 鉴权认证机制 | 百度AI开放平台 中的access_token鉴权机制获取AK/SK, 并得到AppID、API Key、Secret Key三个信息

- 在初始化SDK时，传入认证信息。详细参数参考"初始化SDK"

```
 // ... 其他参数已忽略
[self.asrEventManager setParameter:@"开放平台获取的 appid" forKey:BDS_ASR_APP_ID];    // appid字段
[self.asrEventManager setParameter:@"开放平台获取 apikey" forKey:BDS_ASR_API_KEY]; // apikey字段
```

2.4.2 API Key鉴权机制

> 注意： 邀测阶段暂时仅支持access_token鉴权机制

3. SDK集成

3.1 功能接口

SDK中主要的类和接口如下：

- BDSEventManager：语音事件管理类，用于管理语音识别、语音合成等事件。
- BDSSpeechSynthesizer: 语音合成类，用于管理语音合成和播放
- BDSASRParameters.h、BDSSpeechSynthesizerParams.h: 包含语音识别、语音合成等参数的key常量
- VoiceRecognitionClientWorkStatus（协议方法）：遵循语音识别协议，实现协议方法用于处理语音识别过程中产生的各种事件并响应完成回调。
- BDSSpeechSynthesizerDelegate（协议）：遵循语音合成协议，通过协议方法处理语音合成过程中产生的各种事件

3.1.1 BDSEventManager：语音识别操作类

- 方法列表
  - SDKInitial: 初始化SDK
    - 功能说明：在应用启动后执行一次，不可重复调用。
    - 输入参数
      - configVoiceRecognitionClient：初始化sdk，配置sdk参数
        - asr参数：通过setParameter..forkey 方法设置, 可用参数见下表：

事件参数	类型/值	是否必须	描述
BDS_ASR_PRODUCT_ID	String	是	识别环境ID，1843
deviceID	String	是	设备唯一id，通过getDeviceId方法获取（不允许手动传入）
PID	String	是	识别环境 id，1843
APP_KEY	String	是	识别环境 key，默认值:com.baidu.app
URL	String	是	识别环境 url, 默认值https://vop.baidu.com/v2
APP_ID	String	是	开放平台创建应用后分配的鉴权信息,上线后请使用此参数填写鉴权信息。参考 "2.4 鉴权方式"
APP_API_KEY	String	是	开放平台创建应用后分配的鉴权信息,上线后请使用此参数填写鉴权信息。参考 "2.4 鉴权方式"
BDS_ASR_COMPRESSION_TYPE	String	是	必传，音频压缩类型, 默认值：EVR_AUDIO_COMPRESSION_OPUS 可选值：EVR_AUDIO_COMPRESSION_BV32 ，建议使用默认OPUS
TRIGGER_MODE	int	是	1: 点按识别(iOS不区分点按或长按)； 2: 唤醒后识别考"4.2 开发场景" 配置
ASR_ENABLE_MUTIPLY_SPEECH	int	可选	是否启动全双工 默认双工
BDS_ASR_MODEL_VAD_DAT_FILE	String		vad 资源路径，"初始化环境"时拷贝的资源路径
LOG_LEVEL	int	可选	设置日志等级，可选值： EVRDebugLogLevelOff = 0, 默认 EVRDebugLogLevelFatal = 1, EVRDebugLogLevelError = 2, EVRDebugLogLevelWarning = 3, EVRDebugLogLevelInformation = 4, EVRDebugLogLevelDebug = 5, EVRDebugLogLevelTrace = 6 全量

```
     * 识别回调:  VoiceRecognitionClientWorkStatus， 识别协议接口, 详情请参考VoiceRecognitionClientWorkStatus
 * BDS_ASR_CMD_STOP: 停止语音识别
    * 功能说明：停止当前语音识别，并保留当前的识别结果，已经发送的音频会被正常处理。
    * 输入参数：无
    * 返回值： 无
 * BDS_ASR_CMD_CANCEL: 取消语音识别
    * 功能说明：取消语音识别，并停止后续处理，已经发送未处理的音频不会继续处理。
    * 输入参数：无
    * 返回值： 无
 * BDS_ASR_CMD_PAUSE: 暂停语音识别
    * 功能说明：暂停语音识别 // TODO: 待补充细节
    * 输入参数：无
    * 返回值： 无
 * BDS_ASR_CMD_RESUME: 恢复语音识别
    * 功能说明：恢复语音识别 // TODO: 待补充细节
    * 输入参数：无
    * 返回值： 无
```

3.1.2 VoiceRecognitionClientWorkStatus：语音识别事件监听

用户需要实现此接口，在启动识别时传入。

- 方法列表

- (void)VoiceRecognitionClientWorkStatus:(int)workStatus obj:(id)aObj；
  - workStatus: 接收识别过程中产生的回调事件
    - 输入参数
      - workStatus: enum， 识别回调状态， 具体事件请参考下表
      - obj: id， 事件回调音频数据对象，需解析。
    - 返回值
      - 无

识别回调状态	回调参数	类型	说明
EVoiceRecognitionClientWorkStatusStartWorkIng		枚举	准备就绪，可以说话，一般在收到此事件后通过UI通知用户可以说话了
EVoiceRecognitionClientWorkStatusStart		枚举	检测到开始说话
EVoiceRecognitionClientWorkStatusFirstPlaying		枚举	tts首包播放
EVoiceRecognitionClientWorkStatusEndPlaying		枚举	tts 播放完成
EVoiceRecognitionClientWorkStatusCancel		枚举	用户取消识别
EVoiceRecognitionClientWorkStatusFlushData	返回json字符串，格式为： • -sn: 识别语句id, 例如："C2FF23BA-9894-4430-98E2-149C5FF61493_2" • -results_recognition：识别结果	枚举	语音识别中间结果返回
EVoiceRecognitionClientWorkStatusEnd		枚举	本地声音采集结束，等待识别结果返回并结束录音
EVoiceRecognitionClientWorkStatusNewRecordData		枚举	录音数据回调（对应识别中间结果）
EVoiceRecognitionClientWorkStatusChunkTTS	返回json字符串，格式为： • sn: 识别语句id, 例如："C2FF23BA-9894-4430-98E2-149C5FF61493_2" • origin_result: 合成结果，内容是一个json，包含以下字段 • tex: 音频对应的文本 • audio_data: 音频二进制数据	枚举	TTS 数据返回
EVoiceRecognitionClientWorkStatusFinish	返回json字符串，格式为： • sn: 识别语句id, 例如："C2FF23BA-9894-4430-98E2-149C5FF61493_2" • results_recognition：识别结果	枚举	识别完成，服务器返回结果

3.1.3 SpeechSynthesizer：语音合成操作类

- 方法列表
  - [[BDSSpeechSynthesizer sharedInstance] setSynthesizerDelegate:self];：语音合成协议监听器，需要实现BDSSpeechSynthesizerDelegate协议接口，详情请参考SpeechSynthesizerListener
    - 功能说明：设置语音合成回调协议实现合成回调，
    - 返回值： 无
  - speakSentence:(NSString*)sentence withError:(NSError**)err 播放音频
    - 功能说明：合成播放音频接口
    - 输入参数：
      - sentence: NSString， 音频信息（包含本地文本txt、文字字符串）
      - withError：回调错误信息

3.1.4 BDSSpeechSynthesizerDelegate：语音合成协议监听

用户需要遵循协议，可以监听音频播放器的播放事件

- 方法列表

@optional

- (void)synthesizerNewDataArrived:(NSData *)newData DataFormat:(BDSAudioFormat)fmt characterCount:(int)newLength sentenceNumber:(NSInteger)SynthesizeSentence;
- 功能说明：语音合成
- 输入参数：
- newData: NSData， 语音合成响应.
- fmt：BDSAudioFormat 传递的缓冲区中的音频格式
- newLength：int 当前句子的当前合成字符数
- SynthesizeSentence：NSInteger 合成句子ID由SDK生成，并返回

其余协议方法可参考BDSSpeechSynthesizerDelegate.h头文件说明

3.2 集成步骤

3.2.1 初始化环境

包括几个步骤：

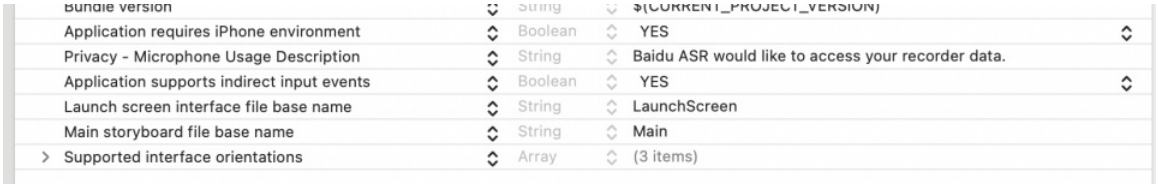
- 将VAD模型文件和AEC模型文件拷贝到设备的目录下

```
 [self configVADFile];
 - (void)configVADFile {
   NSString *vad_filepath = [[NSBundle mainBundle]
pathForResource:@"BDSClientSDK.bundle/EASRResources/chuangxin.ota.v1.vad" ofType:@"pkg"];
   [self.asrEventManager setParameter:vad_filepath forKey:BDS_ASR_MODEL_VAD_DAT_FILE];
}
[self configAECVADDebug];
- (void)configAECVADDebug {
   [self.asrEventManager setParameter:@(YES) forKey:BDS_MIC_SAVE_AEC_DEBUG_FILE];
   [self.asrEventManager setParameter:@(YES) forKey:BDS_MIC_SAVE_VAD_DEBUG_FILE];
   [self.asrEventManager setParameter:@(YES) forKey:BDS_MIC_SAVE_WAKEUP_DEBUG_FILE];
}
```

- 申请需要的权限
    - 需要的权限列表：

权限	说明	是否必须
Privacy - Microphone Usage Description	麦克风权限	是
Application supports indirect input events	支持硬件设备输入	是

权限参考info.plist



3.2.2 实现事件回调函数

- 语音识别回调：即实现协议接口， 示例代码如下：详细参数说明参考VoiceRecognitionClientWorkStatus

```
- (void)VoiceRecognitionClientWorkStatus:(int)workStatus obj:(id)aObj {
  switch (workStatus) {
      case EVoiceRecognitionClientWorkStatusNewEncodeData:
      {
          if (self.encodeFileHandler == nil) {
              self.encodeFileHandler = [self createFileHandleWithName:@"audio_encode" isAppend:NO];
          }
          [self.encodeFileHandler writeData:(NSData *)aObj];
          break;
      }
          // 录音数据回调
      case EVoiceRecognitionClientWorkStatusNewRecordData: {
          if (!self.isPlayTesting) {
              [self.fileHandler writeData:(NSData *)aObj];
          }
          break;
      }
          // 识别工作开始，开始采集及处理数据
      case EVoiceRecognitionClientWorkStatusStartWorkIng: {
          if (!self.isPlayTesting) {
              if (self.currentPlayTones & ASRVoiceRecognitionPlayTonesTypeStart) {
                  [self playTone:@"record_start.pcm"];
              }

              NSDictionary *logDic = [self parseLogToDic:aObj];
              [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: start vr, log: %@\n", logDic]];
              [self onStartWorking];
          }
          break;
      }
      // 检测到用户开始说话
      case EVoiceRecognitionClientWorkStatusStart: {
          if (!self.isPlayTesting) {
              NSLog(@"====EVoiceRecognitionClientWorkStatusStart: %@", aObj);
              [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: detect voice start point, sn: %@.\n", aObj]];
          }

          [self cancelTTSWithSN:cTTS_ASR_SN];
          break;
      }
      // 本地声音采集结束，等待识别结果返回并结束录音
```

```
        // 本地声音采集结束，等待识别结果返回并结束录音
        case EVoiceRecognitionClientWorkStatusEnd: {
            if (!self.isPlayTesting) {
                if (self.currentPlayTones & ASRVoiceRecognitionPlayTonesTypeEnd) {
                    [self playTone:@"record_end.pcm"];
                }

                NSLog(@"====EVoiceRecognitionClientWorkStatusEnd: %@", aObj);
                [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: detect voice end point, sn: %@.\n", aObj]];
            }
            break;
        }
        // 连续上屏
        case EVoiceRecognitionClientWorkStatusFlushData: {
            if (!self.isPlayTesting) {
                [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: partial result - %@.\n\n", [self
getDescriptionForDic:aObj]]];
            }
            break;
        }
        // 语音识别功能完成，服务器返回正确结果
        case EVoiceRecognitionClientWorkStatusFinish: {
            if (!self.isPlayTesting) {
                if (self.currentPlayTones & ASRVoiceRecognitionPlayTonesTypeSuccess) {
                    [self playTone:@"record_success.pcm"];
                }

                [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: final result - %@.\n\n", [self
getDescriptionForDic:aObj]]];
                if (aObj) {
                    self.resultTextView.text = [self getDescriptionForDic:aObj];
                }
                if (!self.longSpeechFlag && !(self.enableMutiplySpeech || self.audioFileType ==
AudioFileType_MutiplySpeech_ASR || self.audioFileType == AudioFileType_MutiplySpeech_Wakeup)) {
                    [self onEnd];
                }
            } else {
                // 播测全双工场景，20s有finalResult则不停止
                self.playTestShouldStop = NO;
                if (self.playTestSceneType == ASRPlayTestSceneType_BaseLine) { // 基线播测：FinalResult里播报一句query
                    [self stopTTS];
                    [[BDSSpeechSynthesizer sharedInstance] speakSentence:@"北京限行尾号 1、6" withError:nil];
                }
            }
            break;
        }
        // 当前音量回调
        case EVoiceRecognitionClientWorkStatusMeterLevel: {
            if (!self.isPlayTesting) {
                //            [self printLogTextView:[NSString stringWithFormat:@"Current Level: %d\n", [aObj intValue]]];
            }
            break;
        }
        // 用户取消
        case EVoiceRecognitionClientWorkStatusCancel: {
            if (!self.isPlayTesting) {
                if (self.currentPlayTones & ASRVoiceRecognitionPlayTonesTypeCancel) {
                    [self playTone:@"record_cancel.pcm"];
                }

                [self printLogTextView:@"CALLBACK: user press cancel.\n"];
                [self onEnd];
                if (self.shouldStartASR) {
```

```
            if (self.shouldStartASR) {
                self.shouldStartASR = NO;
                [self startWakeup2ASR];
            }

            if (self.randomStress) {
                [self wp_rec_randomStress];
            } else if (self.randomStressASR || self.randomStressMutiplyASR) {
                dispatch_after(dispatch_time(DISPATCH_TIME_NOW, NSEC_PER_SEC * 0.5), dispatch_get_main_queue(), ^{
                    [self voiceRecogButtonHelper:EVR_TRIGGER_MODE_CLICK];
                });
            }
        } else {
            if (self.shouldStartASR) {
                self.shouldStartASR = NO;
                [self startWakeup2ASR];
            }
        }
        break;
    }
    case EVoiceRecognitionClientWorkStatusError: {
        if (!self.isPlayTesting) {
            if (self.currentPlayTones & ASRVoiceRecognitionPlayTonesTypeFail) {
                [self playTone:@"record_fail.pcm"];
            }

            [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: encount error - %@.\n", (NSError *)aObj]];
            int errorcode = (int)[(NSError *)aObj code];
            // ota load engine error 需要重新调用""
            // [self.otaEventManger sendCommand:BDS_OTA_INIT withParameters:otaDic];
            // 用法详见：confOTATest 方法
            if (errorcode == EVRClientErrorCodeLoadEngineError) {
                dispatch_async(dispatch_get_global_queue(0, 0), ^{
                    [self.asrEventManager sendCommand:BDS_ASR_CMD_STOP]; // 先停止ASR引擎
                    [self confOTATest]; // 然后调用：BDS_OTA_INIT
                    [self.asrEventManager sendCommand:BDS_ASR_CMD_START]; //然后重启ASR引擎
                });
            }
            // asr engine is busy.
            if (errorcode == EVRClientErrorCodeASRIsBusy) {
                if (self.asrEventManager.isAsrRunning) {
                    [self onEnd];
                } else {
                    [self onStartWorking];
                }
                return;
            }

            if (!(self.enableMutiplySpeech
                  || self.audioFileType == AudioFileType_MutiplySpeech_ASR
                  || self.audioFileType == AudioFileType_MutiplySpeech_Wakeup) ||
                (self.enableMutiplySpeech && (errorcode == EVRClientErrorCodeDecoderNetworkUnavailable
                 || errorcode == EVRClientErrorCodeRecoderNoPermission
                 || errorcode == EVRClientErrorCodeRecoderException
                 || errorcode == EVRClientErrorCodeRecoderUnAvailable
                 || errorcode == EVRClientErrorCodeInterruption
                 || errorcode == EVRClientErrorCodeCommonPropertyListInvalid))) { // 非全双工 || 全双工内部cancel
                [self onEnd];
                if (self.randomStressASR) {
                    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, NSEC_PER_SEC * 0.5), dispatch_get_main_queue(),
^{
                        [self voiceRecogButtonHelper:EVR_TRIGGER_MODE_CLICK];
                    });
```

```
                }
            }
            if (self.randomStress) {
                [self wp_rec_randomStress];
            }
        }
        break;
    }
    case EVoiceRecognitionClientWorkStatusLoaded: {
        if (!self.isPlayTesting) {
            [self printLogTextView:@"CALLBACK: offline engine loaded.\n"];
        }
        break;
    }
    case EVoiceRecognitionClientWorkStatusUnLoaded: {
        if (!self.isPlayTesting) {
            [self printLogTextView:@"CALLBACK: offline engine unLoaded.\n"];
        }
        break;
    }
        /*#SECTION_REMOVE_FROM:OPEN_PLATFORM*/
    case EVoiceRecognitionClientWorkStatusChunkThirdData: {
        if (!self.isPlayTesting) {
            [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: Chunk 3-party data length: %lu\n", (unsigned
long)[(NSData *)aObj length]]];
            if (self.thirdTtsTest) {
                if (self.isThirdFirst) {
                    self.isThirdFirst = NO;
                } else {
                    self.isThirdFirst = YES;
                    [self stopTTS];
                    [self startThirdSynthesize];
                }
            }
        }
        break;
    }
    case EVoiceRecognitionClientWorkStatusChunkNlu: {
        if (!self.isPlayTesting) {
            NSString *nlu = [[NSString alloc] initWithData:(NSData *)aObj encoding:NSUTF8StringEncoding];
            [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: Chunk NLU data: %@\n", nlu]];
        }
        break;
    }
    case EVoiceRecognitionClientWorkStatusChunkEnd: {
        if (!self.isPlayTesting) {
            NSLog(@"====EVoiceRecognitionClientWorkStatusChunkEnd");
            [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: Chunk end, sn: %@.\n", aObj]];
            if (!self.longSpeechFlag && !(self.enableMutiplySpeech || self.audioFileType ==
AudioFileType_MutiplySpeech_ASR || self.audioFileType == AudioFileType_MutiplySpeech_Wakeup)) {
                [self onEnd];
            }
            if (self.randomStress) {
              [self wp_rec_randomStress];
            } else if (self.randomStressASR) {
                dispatch_after(dispatch_time(DISPATCH_TIME_NOW, NSEC_PER_SEC * 0.5), dispatch_get_main_queue(), ^{
                    [self voiceRecogButtonHelper:EVR_TRIGGER_MODE_CLICK];
                });
            }
        }
        break;
    }
```

```
        case EVoiceRecognitionClientWorkStatusFeedback: {
            if (!self.isPlayTesting) {
                NSDictionary *logDic = [self parseLogToDic:aObj];
                [self printLogTextView:[NSString stringWithFormat:@"CALLBACK Feedback: %@\n", logDic]];
            }
            break;
        }
        case EVoiceRecognitionClientWorkStatusRecorderEnd: {
            if (!self.isPlayTesting) {
                [self printLogTextView:@"CALLBACK: recorder closed.\n"];
                if (self.audioFileType == AudioFileType_MutiplySpeech_ASR || self.audioFileType == AudioFileType_ASR) {
                    [self asrGuanceRecorderClosed];
                }
            }
            break;
        }
        case EVoiceRecognitionClientWorkStatusLongSpeechEnd: {
            if (!self.isPlayTesting) {
                [self printLogTextView:@"CALLBACK: Long Speech end.\n"];
                [self onEnd];
            }
            break;
        }
        case EVoiceRecognitionClientWorkStatusStop: {
            if (!self.isPlayTesting) {
                [self printLogTextView:@"CALLBACK: user press stop.\n"];
                [self onEnd];

                if (self.randomStress) {
                   [self wp_rec_randomStress];
                } else if (self.randomStressASR || self.randomStressMutiplyASR) {
                    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, NSEC_PER_SEC * 0.5), dispatch_get_main_queue(), ^{
                        [self voiceRecogButtonHelper:EVR_TRIGGER_MODE_CLICK];
                    });
                }
            }
            break;
        }
        case EVoiceRecognitionClientWorkStatusChunkTTS: {
            if (!self.isPlayTesting) {
                NSDictionary *dict = (NSDictionary *)aObj;
                if (dict) {
                    NSDictionary *ttsResult = [dict objectForKey:@"origin_result"];
                    if (ttsResult && [ttsResult isKindOfClass:[NSDictionary class]]) {
                        NSString *txt = [ttsResult objectForKey:@"tex"];
                        if (txt.length) {
                            [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: ChunkTTS %@.\n", txt]];
                        }

//                        [self playByAudioInfo:dict];
                    }
//                    NSData *ttsData = [dict objectForKey:@"audio_data"];
                    cTTS_ASR_SN = [dict objectForKey:BDS_ASR_CANCEL_TTS_SN_A];
                }
                NSLog(@"Chunk TTS data: %@", [dict description]);
            }
            break;
        }
        case EVoiceRecognitionClientWorkStatusRecorderPermission: {
            if (!self.isPlayTesting) {
                [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: recorder permisson - %@.\n", [aObj
objectForKey:BDS_ASR_RESP_RECORDER_PERMISSION]]];
        }
```

```
            }
            break;
        }
            /*#SECTION_REMOVE_END*/
        default:
            break;
    }
}
```

- 语音合成回调：即实现BDSSpeechSynthesizerDelegate协议接口， 示例代码如下：详细参数说明参考 BDSSpeechSynthesizerDelegate

```
 ##### pragma mark - implement BDSSpeechSynthesizerDelegate
- (void)synthesizerStartWorkingSentence:(NSInteger)SynthesizeSentence{
    NSLog(@"Did start synth %ld", SynthesizeSentence);
    [self.CancelButton setEnabled:YES];
    [self.PauseOrResumeButton setEnabled:YES];
}

- (void)synthesizerFinishWorkingSentence:(NSInteger)SynthesizeSentence engineType:(BDSSynthesizerEngineType)type {
    NSLog(@"Did finish synth: engineType:, %ld, %d", SynthesizeSentence, type);
    if(!isSpeak){
        if(self.synthesisTexts.count > 0 &&
           SynthesizeSentence == [[[self.synthesisTexts objectAtIndex:0] objectForKey:@"ID"] integerValue]){
            [self.synthesisTexts removeObjectAtIndex:0];
            [self updateSynthProgress];
        }
        else{
            NSLog(@"Sentence ID mismatch??? received ID: %ld\nKnown sentences:", (long)SynthesizeSentence);
            for(NSDictionary* dict in self.synthesisTexts){
                NSLog(@"ID: %ld Text:\"%@\"", [[dict objectForKey:@"ID"] integerValue], [((NSAttributedString*)[dict
objectForKey:@"TEXT"]) string]);
            }
        }
        if(self.synthesisTexts.count == 0){
//            [self.CancelButton setEnabled:NO];
//            [self.PauseOrResumeButton setEnabled:NO];
            [self.PauseOrResumeButton setTitle:[[NSBundle mainBundle] localizedStringForKey:@"pause" value:@""
table:@"Localizable"] forState:UIControlStateNormal];
        }
    }
}

- (void)synthesizerSpeechStartSentence:(NSInteger)SpeakSentence{
    NSLog(@"Did start speak %ld", SpeakSentence);
}

- (void)synthesizerSpeechEndSentence:(NSInteger)SpeakSentence{
    NSLog(@"Did end speak %ld", SpeakSentence);
    NSString *docDir = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES)
firstObject];
    NSString *path = [docDir stringByAppendingPathComponent:[NSString stringWithFormat:@"test_speed_9_onnn.pcm",
(long)SpeakSentence]];
    [self.playerData writeToFile:path atomically:YES];

    self.playerData = [[NSMutableData alloc] init];

    if(self.synthesisTexts.count > 0 &&
       SpeakSentence == [[[self.synthesisTexts objectAtIndex:0] objectForKey:@"ID"] integerValue]){
        [self.synthesisTexts removeObjectAtIndex:0];
        [self updateSynthProgress];
    }
```

```
    }
    else{
        NSLog(@"Sentence ID mismatch??? received ID: %ld\nKnown sentences:", (long)SpeakSentence);
        for(NSDictionary* dict in self.synthesisTexts){
            NSLog(@"ID: %ld Text:\"%@\"", [[dict objectForKey:@"ID"] integerValue], [((NSAttributedString*)[dict
objectForKey:@"TEXT"]) string]);
        }
    }
    if(self.synthesisTexts.count == 0){
//        [self.CancelButton setEnabled:NO];
//        [self.PauseOrResumeButton setEnabled:NO];
        [self.PauseOrResumeButton setTitle:[[NSBundle mainBundle] localizedStringForKey:@"pause" value:@""
table:@"Localizable"] forState:UIControlStateNormal];
    }

    [self SynthesizeTapped:nil];
}

- (void)synthesizerNewDataArrived:(NSData *)newData
                       DataFormat:(BDSAudioFormat)fmt
                  characterCount:(int)newLength
                  sentenceNumber:(NSInteger)SynthesizeSentence{
    NSLog(@"NewDataArrived data len: %lu", (unsigned long)[self.playerData length]);
    [self.playerData appendData:newData];


    NSMutableDictionary* sentenceDict = nil;
    for(NSMutableDictionary *dict in self.synthesisTexts){
        if([[dict objectForKey:@"ID"] integerValue] == SynthesizeSentence){
            sentenceDict = dict;
            break;
        }
    }
    if(sentenceDict == nil){
        NSLog(@"Sentence ID mismatch??? received ID: %ld\nKnown sentences:", (long)SynthesizeSentence);
        for(NSDictionary* dict in self.synthesisTexts){
            NSLog(@"ID: %ld Text:\"%@\"", [[dict objectForKey:@"ID"] integerValue], [((NSAttributedString*)[dict
objectForKey:@"TEXT"]) string]);
        }
        return;
    }
    [sentenceDict setObject:[NSNumber numberWithInteger:newLength] forKey:@"SYNTH_LEN"];
    [self refreshAfterProgressUpdate:sentenceDict];
}

- (void)synthesizerTextSpeakLengthChanged:(int)newLength
                           sentenceNumber:(NSInteger)SpeakSentence{
    NSLog(@"SpeakLen %ld, %d", SpeakSentence, newLength);
    NSMutableDictionary* sentenceDict = nil;
    for(NSMutableDictionary *dict in self.synthesisTexts){
        if([[dict objectForKey:@"ID"] integerValue] == SpeakSentence){
            sentenceDict = dict;
            break;
        }
    }
    if(sentenceDict == nil){
        NSLog(@"Sentence ID mismatch??? received ID: %ld\nKnown sentences:", (long)SpeakSentence);
        for(NSDictionary* dict in self.synthesisTexts){
            NSLog(@"ID: %ld Text:\"%@\"", [[dict objectForKey:@"ID"] integerValue], [((NSAttributedString*)[dict
objectForKey:@"TEXT"]) string]);
        }
        return;
    }
```

```
    [sentenceDict setObject:[NSNumber numberWithInteger:newLength] forKey:@"SPEAK_LEN"];
    [self refreshAfterProgressUpdate:sentenceDict];
}
```

3.2.3 初始化SDK

- 示例代码

```
 // 初始化日志配置 默认0 不打印日志 6是打印全部日志
[self.asrEventManager setParameter:@(EVRDebugLogLevelTrace) forKey:BDS_ASR_DEBUG_LOG_LEVEL];
// 环境信息设置：
// 识别环境 - pid. 邀测环境PID统一为4144779，仅供体验使用。生产环境使用请联系技术支持获取专属PID
[self.asrEventManager setParameter:@"4144779" forKey:BDS_ASR_PRODUCT_ID];
// 识别环境 -key,
[self.asrEventManager setParameter:@"com.baidu.app" forKey:BDS_ASR_CHUNK_KEY];
// 识别环境 - url,
[self.asrEventManager setParameter:@"https://audiotest.baidu.com/v2_llm_test" forKey:BDS_ASR_SERVER_URL];

 NSString *deviceId = [self.asrEventManager getDeviceId];
 // 获取uuid
 NSUUID *vendorUUID = [[UIDevice currentDevice] identifierForVendor];
 NSString *uuidString = [vendorUUID UUIDString];
 [self.asrEventManager setParameter:uuidString forKey:BDS_ASR_CUID];
 NSDictionary *extraDic = @{
       @"params": @{
          @"sessionId": uuidString       // 随机生成 session id
       }
   };
[self.asrEventManager setChunkPamWithBdvsID:@"6973568c17a9d84040fe2abe365eb5e3_refactor"            // 设置
bdvs 协议，其中 extraDic 为协议内容
                                    deviceID:deviceId
                                    extraDic:extraDic
                                    queryText:nil];
  NSString *pam = {...//参考demo传递参数具体内容}
 [self.asrEventManager setParameter:pam ?: @"" forKey:BDS_ASR_CHUNK_PARAM];
```

3.2.4 开始识别

```
 // 开始识别阶段参数设置
// 设置VAD/TriggerMode/双工，请参考**开发场景**根据您的业务需求设置。下面示例为双工识别场景。
// 开启VAD
[self.asrEventManager setParameter:YES forKey:BDS_ASR_VAD_ENABLE_LONG_PRESS];
// 是否启动双工
[self.asrEventManager setParameter:@(0) forKey:BDS_ASR_ENABLE_MUTIPLY_SPEECH];
// trigger 1: 点按识别； 2: 唤醒后识别
[self.asrEventManager setParameter:@(EVR_TRIGGER_MODE_WAKEUP) forKey:BDS_ASR_TRIGGE_MODE];
// 必选：设置音频压缩类型，
[self.asrEventManager setParameter:@(EVR_AUDIO_COMPRESSION_OPUS) forKey:BDS_ASR_COMPRESSION_TYPE];
// 是否返回识别中间结果 调用开始识别接口
[self.asrEventManager setParameter:YES forKey:BDS_ASR_COMPRESSION_TYPE];
// 调用BDS_ASR_CMD_START 开始识别
[self.asrEventManager sendCommand:BDS_ASR_CMD_START];
```

3.2.5 停止识别

停止识别，但保留当前识别结果。已经发送的音频会正常识别并生成响应音频。

- 示例代码

```
[self.asrEventManager sendCommand:BDS_ASR_CMD_STOP];
```

3.2.6 取消识别

取消识别，并停止后续处理。已经发送但是还没有识别和响应的数据将会丢弃。

- 示例代码

```
[self.asrEventManager sendCommand:BDS_ASR_CMD_CANCEL];
```

3.2.7 暂停识别

- 示例代码

```
 NSMutableDictionary *params = [NSMutableDictionary dictionaryWithCapacity:2];
// _lastAsrsn为语音识别对应的sn，需要通过VoiceRecognitionClientWorkStatus的回调函数获取。参考下面的代码。
[params setValue:_lastAsrsn forKey:BDS_ASR_SN];
[self.asrEventManager sendCommand:BDS_ASR_CMD_PAUSE withParameters:params];

// 在回调函数中获取sn
- (void)VoiceRecognitionClientWorkStatus:(int)workStatus obj:(id)aObj {
    // 回调obj做解析处理 拿到sn
    NSDictionary *dict = (NSDictionary *)aObj;
    if (dict) {
        NSDictionary *ttsResult = [dict objectForKey:@"origin_result"];
        if (ttsResult && [ttsResult isKindOfClass:[NSDictionary class]]) {
            NSString *txt = [ttsResult objectForKey:@"tex"];
            if (txt.length) {
                [self printLogTextView:[NSString stringWithFormat:@"CALLBACK: ChunkTTS %@.\n", txt]];
            }
            // 此处获取sn
        cTTS_ASR_SN = [dict objectForKey:BDS_ASR_CANCEL_TTS_SN_A];
    }
}
```

3.2.8 恢复识别

- 示例代码

```
[self.asrEventManager sendCommand:BDS_ASR_CMD_RESUME];
```

3.3 开发场景

SDK支持三种开发场景：

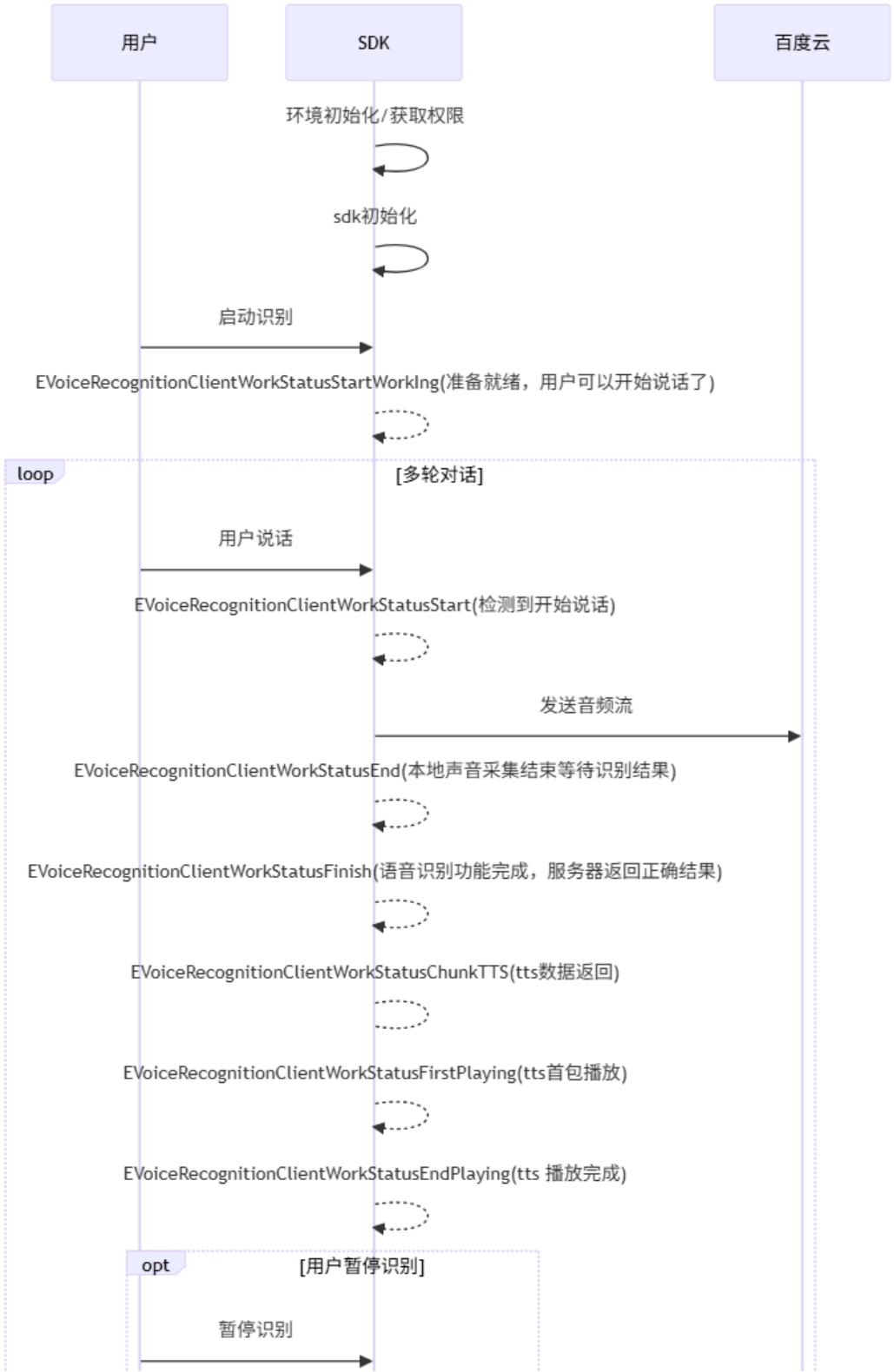
- 双工识别：启动后可以多次进行语音对话，直到用户主动停止识别
- 点按短语音识别：启动后进行60s内的单个短句对话。 // TODO: 补充详细描述
- 长按识别：启动后按住按钮不松开会持续识别，且不会进行断句

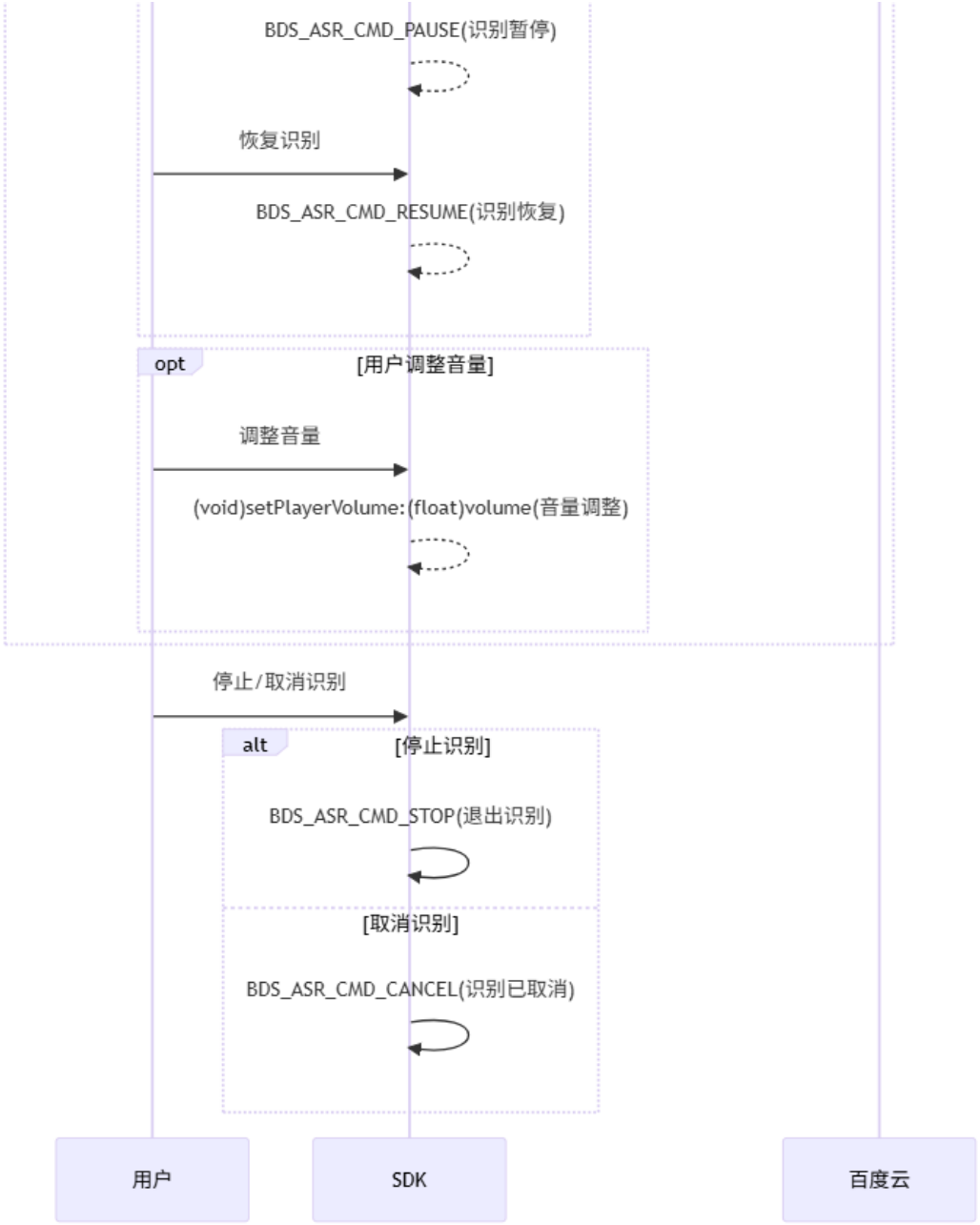
3.3.1 双工识别

- 设置方法

```
- (IBAction)voiceRecogButtonTouchDown:(id)sender {
    // 双工模式识别
    [self.asrEventManager setParameter:@(1) forKey:BDS_ASR_ENABLE_MUTIPLY_SPEECH];
    // 默认点击方式触发  1.点击方式触发 2.唤醒方式触发
    [self.asrEventManager setParameter:@(EVR_TRIGGER_MODE_CLICK) forKey:BDS_ASR_TRIGGE_MODE];
}
```

- 交互流程



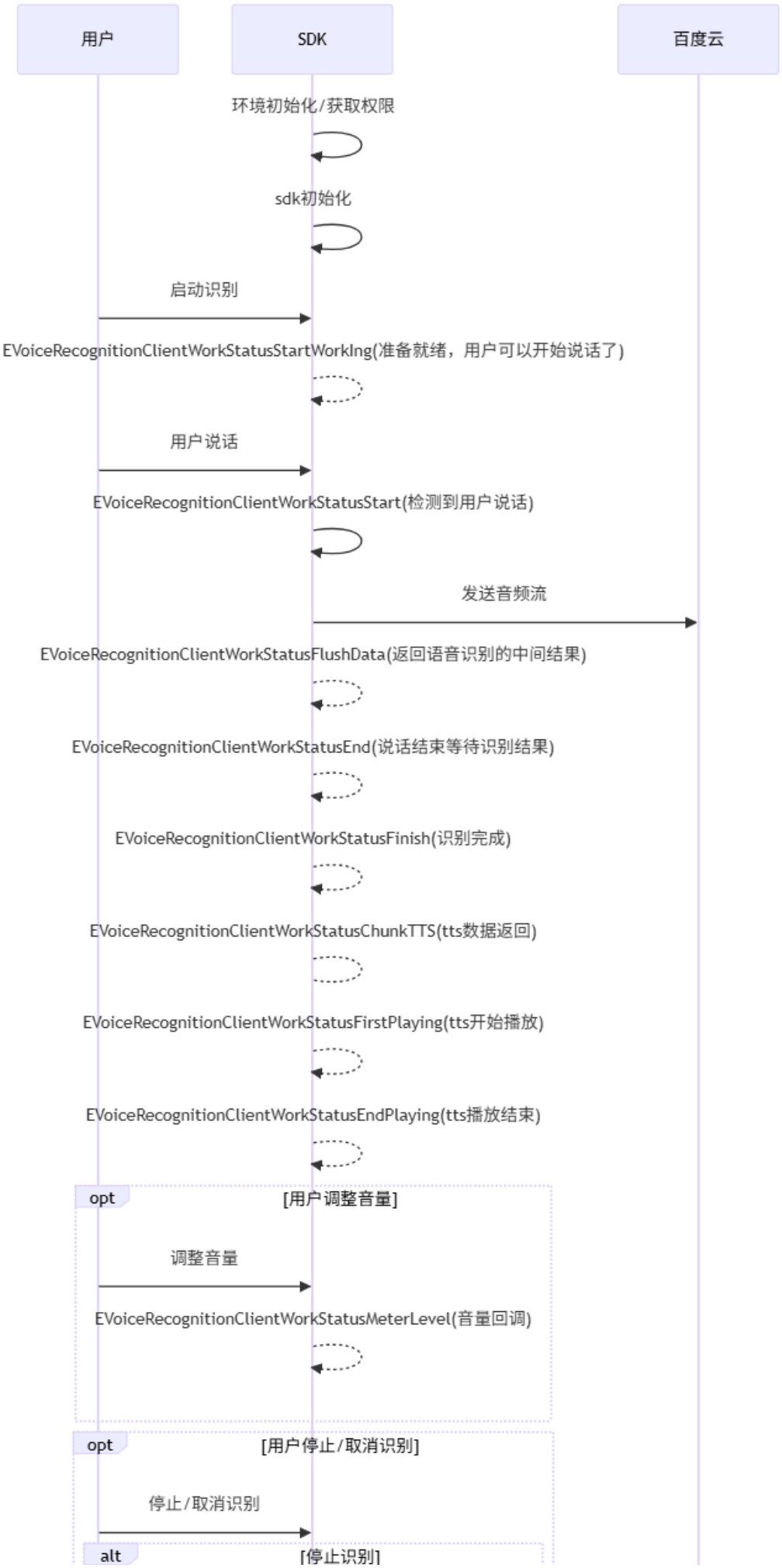


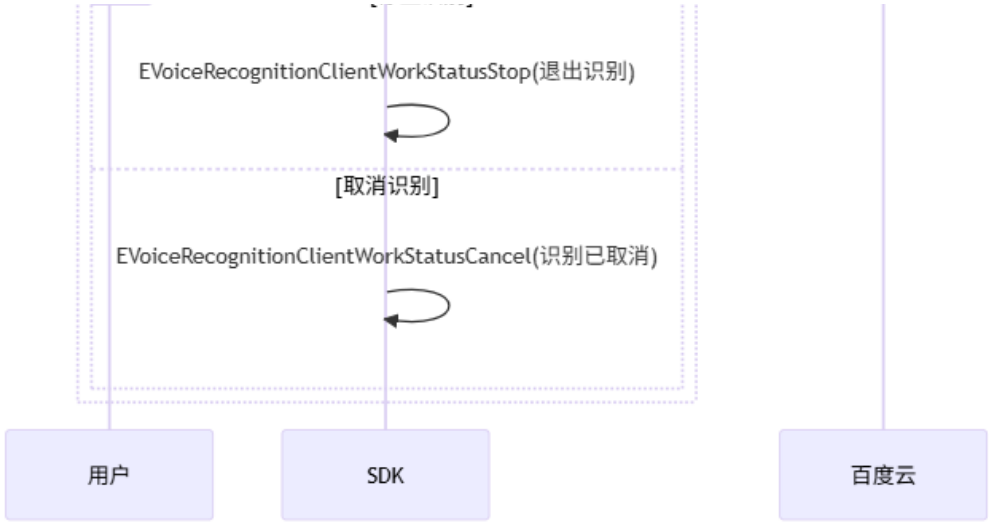
3.3.2 点按短语音识别

- 设置方法

```
// .... 其他参数已忽略
  // 点按
  self.longPressFlag = NO;
  // 双工模式识别
  [self.asrEventManager setParameter:@(1) forKey:BDS_ASR_ENABLE_MUTIPLY_SPEECH];
  // 默认点击方式触发  1.点击方式触发 2.唤醒方式触发
  [self.asrEventManager setParameter:@(EVR_TRIGGER_MODE_CLICK) forKey:BDS_ASR_TRIGGE_MODE];
```

- 交互流程


用户
SDK
百度云
环境初始化/获取权限
sdk初始化
启动识别
EVoiceRecognitionClientWorkStatusStartWorkIng(准备就绪，用户可以开始说话了)
用户说话
EVoiceRecognitionClientWorkStatusStart(检测到用户说话)
发送音频流
EVoiceRecognitionClientWorkStatusFlushData(返回语音识别的中间结果)
EVoiceRecognitionClientWorkStatusEnd(说话结束等待识别结果)
EVoiceRecognitionClientWorkStatusFinish(识别完成)
EVoiceRecognitionClientWorkStatusChunkTTS(tts数据返回)
EVoiceRecognitionClientWorkStatusFirstPlaying(tts开始播放)
EVoiceRecognitionClientWorkStatusEndPlaying(tts播放结束)
opt
[用户调整音量]
调整音量
EVoiceRecognitionClientWorkStatusMeterLevel(音量回调)
opt
[用户停止/取消识别]
停止/取消识别
alt
[停止识别]

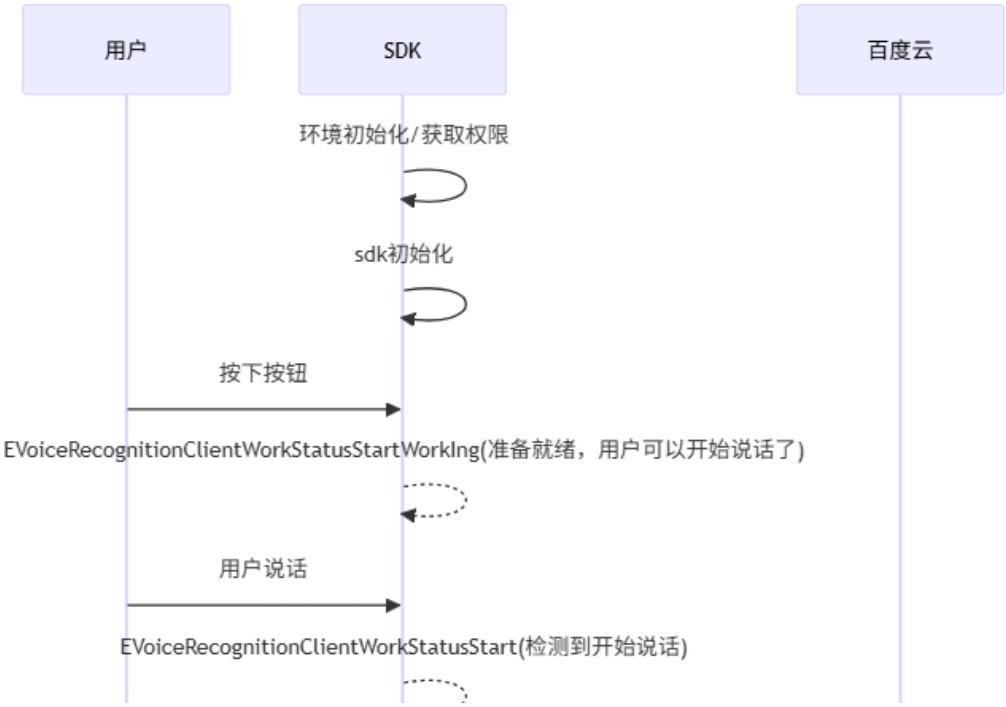


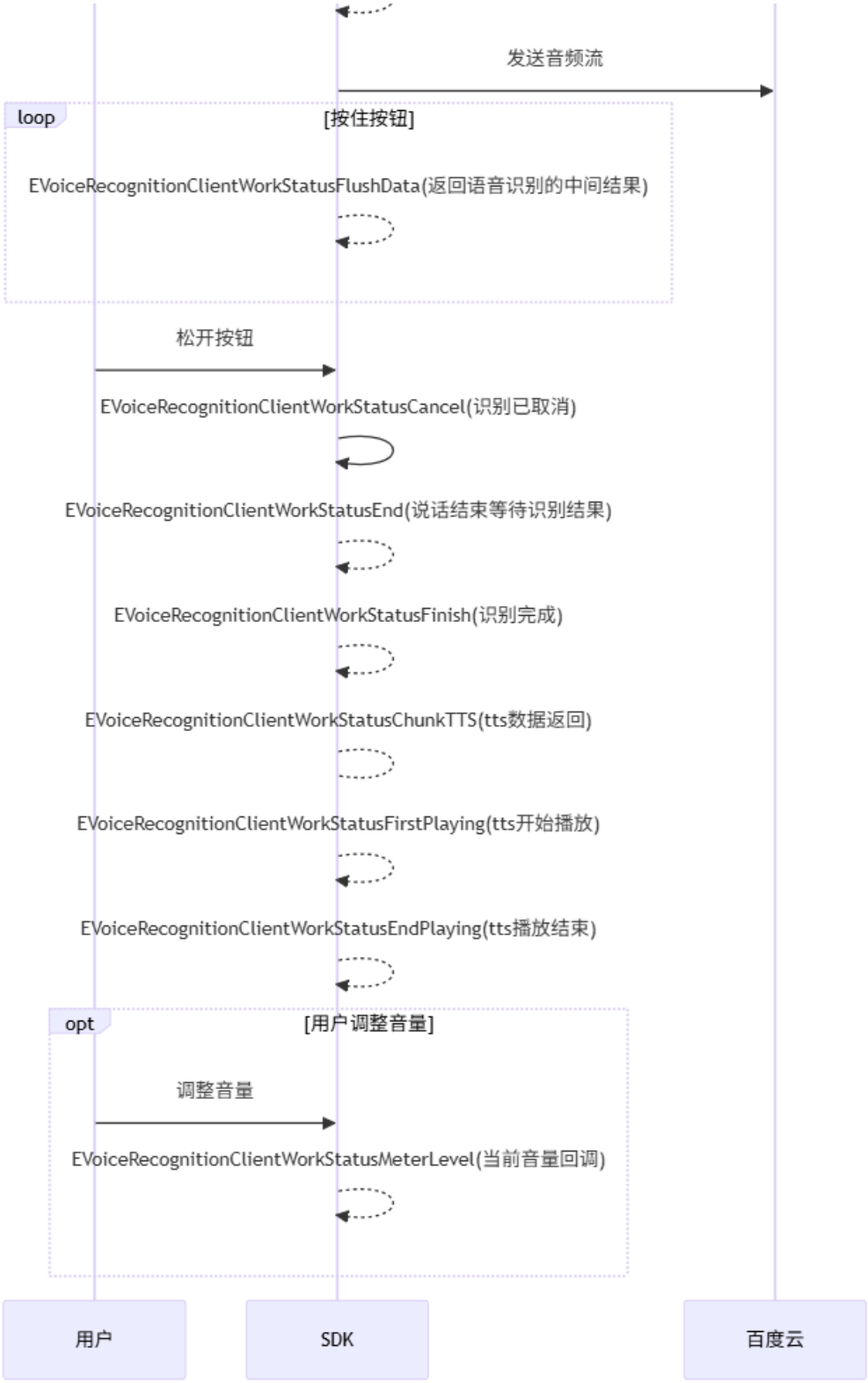
3.3.3 长按识别

- 设置方法

```
// .... 其他参数已忽略
  // 点按
  self.longPressFlag = YES;
  // 双工模式识别
  [self.asrEventManager setParameter:@(1) forKey:BDS_ASR_ENABLE_MUTIPLY_SPEECH];
  // 默认点击方式触发  1.点击方式触发 2.唤醒方式触发
  [self.asrEventManager setParameter:@(EVR_TRIGGER_MODE_CLICK) forKey:BDS_ASR_TRIGGE_MODE];
```

- 交互流程




发送音频流
loop
[按住按钮]
EVoiceRecognitionClientWorkStatusFlushData(返回语音识别的中间结果)
松开按钮
EVoiceRecognitionClientWorkStatusCancel(识别已取消)
EVoiceRecognitionClientWorkStatusEnd(说话结束等待识别结果)
EVoiceRecognitionClientWorkStatusFinish(识别完成)
EVoiceRecognitionClientWorkStatusChunkTTS(tts数据返回)
EVoiceRecognitionClientWorkStatusFirstPlaying(tts开始播放)
EVoiceRecognitionClientWorkStatusEndPlaying(tts播放结束)
opt
[用户调整音量]
调整音量
EVoiceRecognitionClientWorkStatusMeterLevel(当前音量回调)
用户
SDK
百度云

3.4 SDK错误码

错误码	含义
655361	录音设备异常
655362	无录音权限
655363	录音设备不可用
655364	录音中断
655365	大量空白音频
1310722	用户未说话
1966082	网络不可用
1966084	解析url失败
2031617	请求超时
2031620	本地网络联接出错
2225211	服务端交互发生错误
2225213	等待上行流建立超时
2225218	语音过长
2225219	声音不符合识别要求
2225221	没有找到匹配结果
2225223	协议参数错误
2625535	识别繁忙

大模型声音复刻

简介

产品说明

百度大模型声音复刻是使用全新自研语音大模型算法打造的轻量级音色定制方案。用户只需录制**5s**的音频，即可极速复刻音色。支持自定义文本复刻与双流式语音合成，广泛应用于配音、数字人、情感陪伴、语音助手等场景。
若您希望快速体验相关功能，请点击链接进入大模型声音复刻体验专区。

产品优势

- **自定义文本复刻：**无需指定音频文本，实现自由复刻，降低调试成本
- **超低门槛：**无需专业设备与场地，极大提升效率，降低使用门槛
- **精准还原：**精准还原音色特点、说话风格、韵律起伏、声学环境
- **极速复刻：**最低5秒即可完成高品质复刻，精准呈现音色细节，高效逼真
- **多种语言：**支持日语复刻，支持河南话、上海话、四川话等多地方言的复刻还原与生成
- **多种情绪：**支持情感迁移，支持参数控制合成语音呈现高兴、惊讶、愤怒等情绪

接口列表

序号	接口名称	接口URL
1	获取训练文本	https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/text
2	创建音色	https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/create
3	音色列表查询	https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/list
4	音色详情查询	https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/detail
5	删除音色	https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/delete
6	流式在线合成	wss://aip.baidubce.com/ws/2.0/speech/publiccloudspeech/v1/voice/clone/tts
7	非流式在线合成	https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/tts

示例代码

您可以参考以下示例代码，进行音色创建与在线合成：voiceclone_demo.zip

返回码

返回码	提示信息	备注说明
0	success	处理成功
10014	user concurrency limit exceeded	用户的并发超限，如有高并发需求，请先提交合作咨询，或者提交工单
10015	user usage limit reached	用户的配额超限，如有高额度需求，请先提交合作咨询，或者提交工单
10020	temporary service err, please try again	服务临时错误，请稍候再试
10021	voiceprint detection failed	音频与文本匹配校验失败，请按照返回text进行朗读
10022	text id not exists or text id already expire	text_id 不存在或者 text_id 已经超过 24 小时，请重新调用获取训练文本接口
10023	download failed or audio file size exceeded limit	音频URL不可访问或上传文件超过5M，请确保音频URL可访问并且上传文件小于等于5M
10025	the data has been deleted or does not have permission	该音色不存在，或者该音色已经被开发者删除
10026	invalid audio content	文件内容无效，请提供格式正确的音频，避免修改音频后缀
10027	voiceprint detection failed	音频可能涉及敏感人物，请重新上传音频或尝试指定文本复刻
11000	Synthesis exception	合成异常，请检查音色和合成语音的语种是否一致
11002	限流无额度	用户的配额超限，如有高额度需求，请先提交合作咨询，或者提交工单
11003	限流并发超限	用户的并发超限，如有高并发需求，请先提交合作咨询，或者提交工单
11004	text exceeded the limit	文本超长，请缩短文本重试
11006	No access permission for this voice_id	当前voice_id 错误，请检查 voice_id 是否正确
11007	token information or iam information must exist	未传递有效的鉴权信息，请正确输入 access_token 鉴权信息或者 Iam鉴权信息
11008	temporary service err, please try again	服务临时异常，请稍候重试
11009	required parameters are missing	参数缺失，请检查输入参数

11009	required parameters are missing	参数缺失，请检查输入参数
11010	parameters are invalid	参数无效，请检查输入参数
11011	voice_id not exists	当前的 voice_id 不存在，请检查 voice_id 是否正确
11012	invalid page parameter, must be at least 1	page 参数无效，请确保page参数大于等于 1
11013	invalid base64-encoded audio content	音频内容无效，请使用有效的 base64编码音频
11014	this token/iam information invalid or has no access data permission	用户没有访问该数据的权限，请检查账号权限
11015	the text contains memorable vocabulary	文本包含敏感信息，请去掉敏感信息后重试
12001	the audio file is too short	音频内容太短，请更换音频
12000	download file failed	文件下载失败，请检查音频下载链接
12002	wer check failed	未检测到有效音频，请根据返回text 进行朗读
12003	audio detect snr failed	未检测到有效音频，请根据返回text 进行朗读
12004	recognition failed	无有效的人声，请更换音频
12005	detect audio level failed	无有效的人声，请调整音量后重新复刻
12006	detect audio speed failed	无有效的人声，请调整语速后重新复刻
12007	quality failed	音频质量较差，请更换音频

反馈

若您对文档内容有任何疑问，可以通过百度云控制台提交工单联系我们，咨询问题类型请选择人工智能服务；

音色创建与管理

音色创建

获取训练文本（非必需）

接口描述

接口描述：通过此接口获取用于训练的文本，使用此接口则创建音色时需要上传与此文本内容对应的音频；若希望使用自定义复刻，无需调用本接口，在创建音色接口处无需填写音频文本ID，其他参数保持不变。

接口名称：https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/text

请求方法：POST

输入参数

属性	参数名称	类型	是否必填	说明
鉴权令牌	access_token	string	鉴权方式二选一	鉴权令牌
鉴权API Key	Authorization	string	鉴权方式二选一	鉴权API Key

输出参数

属性	参数名称	类型	说明
错误状态	status	int	
错误信息	message	string	
数据	data	object	响应数据

data响应参数

属性	参数名称	类型	说明
文本ID	text_id	string	唯一文本id（获取文本id后，文本id有效期为24小时，每个文本id创建音色后自动失效）
文本内容	text	string	

示例

```
curl --silent --location --request POST
'https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/text?access_token=XXX'
```

创建音色

接口描述

接口描述：通过上传训练音频来创建音色，支持音频链接与音频内容两种音频上传方式

接口名称：https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/create

请求方法：POST

输入参数

属性	参数名称	类型	是否必填	说明
鉴权令牌	access_token	string	鉴权方式二选一	鉴权令牌
鉴权API Key	Authorization	string	鉴权方式二选一	鉴权API Key
音色名称	voice_name	string	必填	音色名称，同一个用户下面，音色名称不能重复
音色描述	voice_desc	string	可选	音色说明
音色语种	lang	string	可选	创建的音色语种，目前支持的语种： 中英语：zh; 日语：ja; *若创建音色阶段未填写lang参数，默认为zh（中英语）。* *若希望音色用于日语合成，建议选择ja参数，并使用10秒至30秒长度的日语音频作为注册音频。* *若希望合成方言或控制语音情感，合成语种必须选择zh参数。*
音频链接	audio_url	string	音频链接和音频内容二选一	两个参考同时传：以 audio_file 为准 支持：wav、mp3、m4a、ogg、aac。音频大小5M以内，复刻中英音色，时长建议在5～20秒；复刻日语音色，时长建议在10～30秒。
音频内容	audio_file	string	音频链接和音频内容二选一	音频文件内容base64 复刻中英音色，时长建议在5～20秒；复刻日语音色，时长建议在10～30秒。
音频文本ID	text_id	string	选填	即文本ID，使用自定义文本复刻则无需填写

输出参数

属性	参数名称	类型	说明
错误状态	status	int	0 创建成功，其他为异常
错误信息	message	string	
数据	data	object	对应发音人ID

data响应参数

属性	参数名称	类型	说明
音色 ID	voice_id	int	唯一id

示例

- 通过音频URL创建音色

```
 curl --silent --location --request POST
'https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/create?access_token=XXXX' \
--header 'Content-Type: application/json' \
--data '{
    "voice_name": "example",
    "voice_desc": "温柔细腻的音色",
    "audio_url": "https://bj.bcebos.com/example.wav",
    "text_id": "text_5692181d-8825-40ec-b80f-87b79e3a9345"
}'
```

- 通过音频-base64 编码创建音色

```
 curl --silent --location --request POST
'https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/create?access_token=XXX' \
--header 'Content-Type: application/json' \
--data '{
    "voice_name": "example",
    "voice_desc": "温柔细腻的音色",
    "audio_file": "音频 Base64 编码",
    "text_id": "text_5692181d-8825-40ec-b80f-87b79e3a9345"
}'
```

> **大模型声音复刻的音色创建说明：**
>
> 通过以上接口创建的音色，若在1年内没有调用合成记录，该音色将被删除，后续将无法使用。
>
> **大模型声音复刻和迁移能力说明：**
>
> 大模型声音复刻分为复刻和迁移两种能力。复刻指保留声音语种/方言等特征的能力，迁移指保持音色特征的同时增加语种/方言等特征的能力。例如：
>
> - 创建音色时输入普通话，合成普通话=复刻
> - 创建音色时输入河南话，合成河南话=复刻
> - 创建音色时输入普通话，合成河南话=迁移
>
> **大模型语种方言复刻说明：**

- 目前支持河南话、上海话、四川话、贵州话、湖南话五种方言的复刻及迁移。为确保方言复刻迁移效果，建议您使用普通话或对应方言进行音色创建。
- 目前支持日语语种的复刻。为确保日语复刻效果，建议您使用10-30秒的日语内容进行音色创建。
- 为保证不同语种合成效果，建议您避免在音色创建和语音合成时选择不一致的lang参数，或者使用中日文混合的文本。

大模型情感迁移说明：

- emotion参数和合成文本的隐含情感不能相差太大，如果冲突以emotion参数为准。
- emotion参数和复刻音频情感理论上不能相差太大，如果冲突以emotion参数为准。合成语音会受到复刻音频声音特质的影响，请您结合实际效果选择emotion参数。
- 日语音色暂不支持语音情绪迁移。

音色管理

音色列表查询

接口描述

接口描述：获取用户已经创建的音色列表

接口名称：https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/list

请求方法：POST

输入参数

属性	参数名称	类型	是否必填	说明
鉴权令牌	access_token	string	鉴权方式二选一	鉴权令牌
鉴权API Key	Authorization	string	鉴权方式二选一	鉴权API Key
页码	page	int	选填	列表页面：page>=1

输出参数

属性	参数名称	类型	说明
错误状态	status	int	0代表成功，其他为异常
错误信息	message	string	
数据	data	object	音色列表信息

data响应参数

属性	参数名称	类型	说明
总数	total	int	
页码	page	int	
每页数量	page_size	int	
音色列表	items	object [ ]	

items响应参数

属性	参数名称	类型	说明
音色 ID	voice_id	int	训练后的音色 ID
音色名称	voice_name	string	音色名称
音色描述	voice_desc	string	音色说明
音色语种	lang	string	音色语种
创建时间	create_time	string	创建时间，Unix 时间戳
状态	status	int	恒等于 0

示例

```
 curl --silent --location --request POST
'https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/list?access_token=XXX' \
--data '{
    "page": 1
}'
```

音色详情查询

接口描述

接口描述：根据音色ID查询音色的详情信息

接口名称：https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/detail

请求方法：POST

输入参数

属性	参数名称	类型	是否必填	说明
鉴权令牌	access_token	string	鉴权方式二选一	鉴权令牌
鉴权API Key	Authorization	string	鉴权方式二选一	鉴权API Key
音色ID	voice_id	int	必填	

输出参数

属性	参数名称	类型	说明
错误状态	status	int	0代表成功，其他为异常
错误信息	message	string	
数据	data	object	音色详情信息

data响应参数

属性	参数名称	类型	说明
音色 ID	voice_id	int	训练后的音色 ID
音色名称	voice_name	string	音色名称
音色描述	voice_desc	string	音色说明
音色语种	lang	string	音色语种
音色状态	status	int	恒等于 0
创建时间	create_time	string	创建时间

示例

```
 curl --silent --location --request POST
'https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/detail?access_token=XXX' \
--header 'Content-Type: application/json' \
--data '{
    "voice_id": 1063622
}'
```

删除音色

接口描述

接口描述：对已经创建的音色进行删除

接口名称：https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/delete

请求方法：POST

输入参数 | 属性 | 参数名称 | 类型 | 是否必填 | 说明 | | ---- | ----- | ---- | ---- | ---- | | 鉴权令牌 | access_token | string | 鉴权方式二选一 |鉴权令牌 | | 鉴权API Key | Authorization | string | 鉴权方式二选一 |鉴权API Key| | 音色ID | voice_id | int | 必填 |

输出参数

通过返回的status判断是否成功，如失败则查看message获得具体错误信息

属性	参数名称	类型	说明
错误状态	status	int	0 删除成功，1 删除异常
错误信息	message		

示例

```
 curl --silent --location --request POST
'https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/delete?access_token=XXX' \
--header 'Content-Type: application/json' \
--data '{
    "voice_id": 1063622
}'
```

语音合成

语音合成

非流式在线合成（HTTP）

接口描述

接口描述：音色创建成功后，通过创建得到的音色ID进行文本的合成

接口名称：https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/tts

请求方法：POST（Content-Type: application/json）

输入参数

属性	参数名称	类型	是否必填	说明
access_token	string	query	二选一	鉴权令牌
Authorization	string	header	二选一	鉴权API Key
音色ID	voice_id	int	必填	训练后的音色 ID
合成文本	text	string	必填	总字数不超过500个字符，1个中文字、英文字母、数字或符号均算作1个字符
合成语言	lang	string	可选	待合成的语种类型，支持以下语种： 中英语：zh; 日语：ja; *若合成语音阶段未填写lang参数，默认为创建音色选择的语种。*
合成方言	dialect	string	可选	待合成的方言类型，支持以下方言： 上海话：wuu-CN-shanghai； 河南话：zh-CN-henan； 四川话：zh-CN-sichuan； 湖南话：zh-CN-hunan； 贵州话：zh-CN-guizhou； **dialect参数与emotion参数请勿同时上传。**
合成情感	emotion	string	可选	待合成语音的情感，支持以下情感： 高兴：happy; 悲伤：down； 惊讶：surprise; 愤怒：angry; 恐惧：fear； 厌恶：disgust; *不上传emotion参数时（例如text_ctrl={"emo":""}），默认按照注册音频效果合成。* **dialect参数与emotion参数请勿同时上传。**
音调	pitch	float	可选	音调，取值范围[0, 15]，默认为5
音量	volume	float	可选	音量，取值范围[0, 15]，默认为5
语速	speed	float	可选	语速，取值范围[0, 15]，默认为5
输出格式	media_type	string	可选	输出文件格式，支持wav、mp3，默认值为wav
采样率	sample_rate	int	可选	采样率，仅支持将采样率降采，支持8000、16000、24000
元数据标识	metadata	string	可选	元数据标识，字段与取值： ContentPropagator：内容传播服务提供者 PropagateID：内容传播编号 *ContentPropagator、PropagateID请传入string类型的数据。* *若不传入两个字段，默认不添加元数据标识；传入字段后音频header将添加Label、ContentProducer、ProduceID、ContentPropagator、PropagateID总计5个字段。*

输出参数

需要根据 Content-Type的头部来确定是否服务端合成成功。 如果合成成功，返回的Content-Type以“audio”开头

- media_type=wav ，返回为二进制wav文件，具体header信息 **Content-Type: audio/wav**；

- media_type=mp3 ，返回为二进制mp3文件，具体header信息 **Content-Type: audio/mp3**；

如果合成出现错误，则会返回json文本，具体header信息为：**Content-Type: application/json**。

属性	参数名称	参数类型	描述
错误状态	status	int	0代表成功，其他为异常
错误信息	message	string	错误消息

示例

```
curl --location --request POST 'https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/clone/tts?access_token=xxxx' \
{
    "text": "当春风拂过，大地渐渐回暖，万物复苏的季节到来了。花儿在这个时候竞相开放，桃花、樱花、杏花，还有那细雨中的紫藤，都以最绚烂的姿态展示自己的美丽。",
    "voice_id": 100001,
    "metadata":{
        "ContentPropagator":XXXXX,
        "PropagateID":XXXXX,
    }
}
```

流式在线合成（Websocket）

接口描述

接口描述：音色创建成功后，通过创建得到的音色ID进行文本的合成

接口名称：wss://aip.baidubce.com/ws/2.0/speech/publiccloudspeech/v1/voice/clone/tts

建立连接 请求路径：/ws/2.0/speech/publiccloudspeech/v1/voice/clone/tts

参数名	类型	参数位置	是否必填	说明
access_token	string	query	二选一	鉴权令牌
Authorization	string	header	二选一	鉴权API Key
voice_id	int	query	必填	音色ID
idle_timeout	int	query	可选	空闲超时时间，单位秒，默认值60，范围[5, 600]

状态码说明

状态码	状态信息	错误码	错误消息	说明
400	Bad Request	216100	Invalid 'foo' value: bar.	参数错误
400	Bad Request	216100	Invalid 'foo' value: bar, range: [min, max].	参数错误
400	Bad Request	216101	Missing required parameter: 'foo'.	参数缺失
400	Bad Request	400	websocket: the client is not using the websocket protocol: ...	WebSocket协议错误
401	Unauthorized	216100	Invalid access_token parameter.	access_token参数格式不正确
401	Unauthorized	216100	Invalid Authorization header.	Authorization请求头格式不正确
401	Unauthorized	216100	dialect and emotion cannot be set at the same time.	dialect和emotion参数无法同时上传
401	Unauthorized	216100	Invalid 'emotion' value.	上传的emotion参数为非法
401	Unauthorized	216101	Missing access_token parameter or Authorization header.	缺少鉴权参数
401	Unauthorized	110	Access token invalid or no longer valid	access_token 校验不通过。
401	Unauthorized	111	Access token expired	access_token 过期，建议更新获取Access token 重新请求。
401	Unauthorized	217	IAM auth error	API Key校验不通过
403	Forbidden	6	No permission to access data	appid没有相应接口权限，可以检查console页面app管理中勾选的AI能力情况。
404	Not Found			输入的url错误
429	Too Many Requests	15	Open api concurrency limit reached	触发并发限流
429	Too Many Requests	17	Open api usage limit reached	无剩余可用额度
400	Bad Request	216404	Voice id not exists.	voice_id不存在
500	Internal Server Error	282000	Internal error.	服务器内部错误
400	Bad Request	282101	The voice_id service has not been started.	服务未启动
400	Bad Request	216403	Not allowed to use this voice.	对该voice_id无权限

初始化 |参数名称|类型|是否必填|说明| |-|-|-|-| |type|string|必填|开始帧的类型，固定值 system.start| |payload|object|可选|合成参数，详见下表|

payload字段说明

参数名称	类型	是否必填	说明
lang	string	可选	待合成的语种类型，支持以下语种： 中英语：zh; 日语：ja; *若合成语音阶段未填写lang参数，默认为创建音色选择的语种。*
dialect	string	可选	待合成的方言类型，支持以下方言： 上海话：wuu-CN-shanghai； 河南话：zh-CN-henan； 四川话：zh-CN-sichuan； 湖南话：zh-CN-hunan； 贵州话：zh-CN-guizhou； **dialect参数与emotion参数请勿同时上传。**
emotion	string	可选	待合成语音的情感，支持以下情感： 高兴：happy; 悲伤：down； 惊讶：surprise; 愤怒：angry; 恐惧：fear； 厌恶：disgust; *不上传emotion参数时（例如text_ctrl={"emo":""}），默认按照注册音频效果合成。* **dialect参数与emotion参数请勿同时上传。**
pitch	float	可选	音调，取值 0-15，默认为 5
volume	float	可选	音量，基础音库取值0-9，其他音库取值 0-15，默认为 5
speed	float	可选	语速，取值 0-15，默认为 5
media_type	string	可选	音频格式，支持wav、mp3、pcm，默认值为wav 若希望使用postman测试流式合成效果，建议使用mp3格式
sample_rate	int	可选	采样率，支持8000、16000、24000 初始化过程中，可以通过audio_ctrl参数指定采样率，sample_rate和audio_ctrl不可同时使用
metadata	string	可选	元数据标识，字段与取值： ContentPropagator：内容传播服务提供者 PropagateID：内容传播编号 *ContentPropagator、PropagateID请传入string类型的数据。* *若不传入两个字段，默认不添加元数据标识；传入字段后音频header将添加Label、ContentProducer、ProduceID、ContentPropagator、PropagateID总计5个字段。*
spd（兼容字段）	int	可选	语速，取值 0-15，默认为 5
pit（兼容字段）	int	可选	音调，取值 0-15，默认为 5
vol（兼容字段）	int	可选	音量，基础音库取值0-9，其他音库取值 0-15，默认为 5
aue（兼容字段）	int	可选	音频格式，3=mp3-16k/24k，4=pcm-16k/24k，5=pcm-8k，6=wav-16k/24k，和media_type参数不可同时设置

示例代码如下：

```
{
  "type": "system.start",
  "payload": {
    "lang": "zh",
    "media_type": "mp3",
    "pitch": 5,
    "volume": 5,
    "speed": 5,
    "dialect":"zh-CN-henan",
     "metadata":{
       "ContentPropagator":XXXXX,
       "PropagateID":XXXXX,
     }
  }
}
```

正常响应将返回：

```
{
 "type": "system.started",
 "code": 0,
 "message": "success",
 "headers": {
  "session_id": "57944e1a-470f-4c30-9cf6-36e0aa645e58"
 }
}
```

返回码说明

type	code	message	说明
system.started	216100	Invalid system.start payload.	参数错误
system.started	216100	Invalid 'foo' value: bar.	参数错误
system.started	216100	Invalid 'foo' value: bar, range: [min, max].	参数错误

发送文本 示例代码如下：

```
{
  "type": "text",
  "payload": {
    "text": "需要进行语音合成的文字"
  }
}
```

返回码说明

type	code	message	说明
system.error	216100	Invalid text payload.	参数错误
system.error	216101	Missing required parameter: 'text'.	缺少text参数
system.error	216103	Text exceeded 1000 characters limit.	单次文本不能超过1000字
system.error	216429	The current pending text is too long to be processed.	发送频率过快，待处理文本过长

备注：

1. 客户端发送的文本信息无内容时，返回system.error，code=216101，message=Missing required parameter: 'text'.

结束合成 客户端请求

参数名称	类型	是否必填	说明
type	string	必填	结束帧的类型，固定值 system.finish

```
{
  "type":"system.finish",
}
```

正常响应

```
{
 "type": "system.finished",
 "code": 0,
 "message": "success",
 "headers": {
   "session_id": "58b18cef-c588-4f04-a4a4-4a2977072e58"
 }
}
```

服务端主动发送 返回码说明

type	code	message	说明
system.error	216604	Open api usage limit reached	额度已用完，无剩余可用额度

AI翻译配音

AI翻译配音快速体验指南

文档说明

百度AI翻译配音是集成语音识别、文本翻译、声音复刻等能力的短剧翻译配音方案。能够精准复刻视频语音的音色与情绪，实现高保真的跨语言内容转换，适用于短剧译制、短剧营销等场景。

本文档主要介绍如何快速获取AI翻译配音邀测资格并体验服务。

1. 注册及实名认证

使用AI翻译配音服务之前，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，可参考账号注册。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申请并使用翻译配音服务。

2. 进入平台领取额度

通过实名认证后，点击进入AI翻译配音平台。在首页右上角点击领取额度，等待片刻即可获得邀测资格与对应额度。邀测用户将获得30分钟（个人用户）或60分钟（企业用户）的翻译配音额度，额度长期有效，同一视频的重新处理/校准导出不会重复

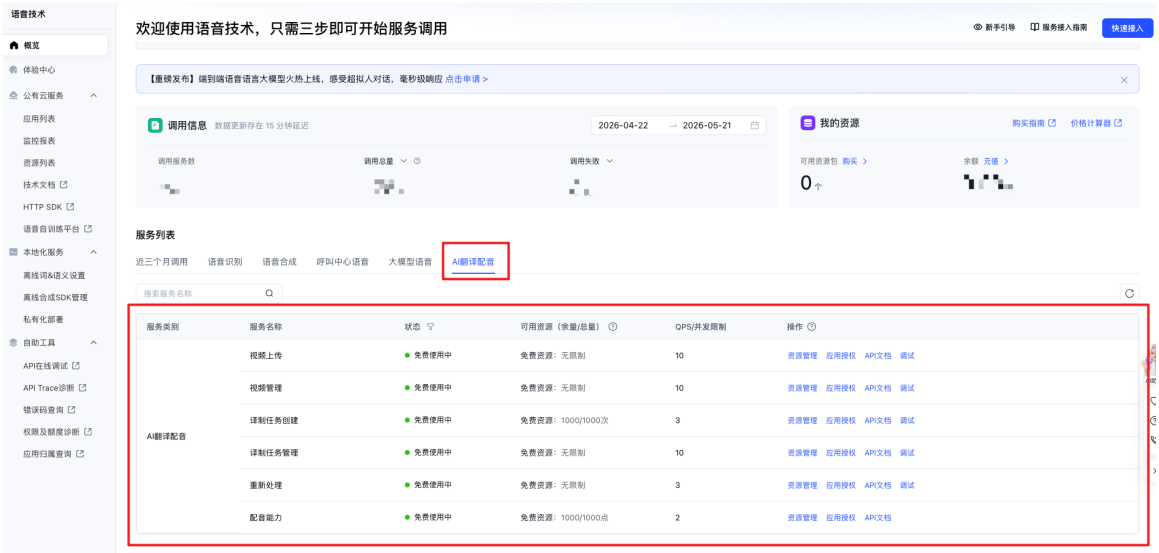
消耗额度。



若您需要更大的测试用量，可点击咨询与合作和商务进行再次申请。

3. 确认额度到账

点击领取后，额度需要一段时间到账。重新进入AI翻译配音平台，若右上角额度更新，则资源已发放完成；若尚未出现相关资源信息，请刷新或重新登录。 额度发放完毕后，语音技术控制台-服务列表同样会出现“AI翻译配音”与相关资源信息。



注：通常只有“译制任务创建”、“配音能力”会显示额度。“译制任务创建”额度会每日刷新，“配音能力”额度显示的是有效期内的全部额度，您使用角色分析、字幕提取、翻译配音等能力的所有扣费都会显示在“配音能力”服务上。

4. 使用平台服务

点击进入AI翻译配音平台即可进行体验。首页新手引导、演示视频能够帮助您更快上手，更多细节请参考AI翻译配音使用说明。

译制视频限制如下：

1. 请上传30～300秒的短视频。
2. 目前已支持横屏和竖屏视频上传，但请上传主流比例的视频。
3. 视频的大小上限为1080x1920，大于此尺寸的视频暂不支持。
4. 目前已支持真人剧与漫剧人脸识别。
5. 目前仅支持英文与印尼语译制，若您希望支持其他语言或视频本地化，请联系商务咨询。
6. 目前仅开放体验平台，接口调用正在建设中，详细范围与接入说明，请联系商务获取。

反馈渠道

若您对AI翻译配音业务有任何需求或疑问，可通过工单服务进行反馈。

AI翻译配音使用说明

文档说明

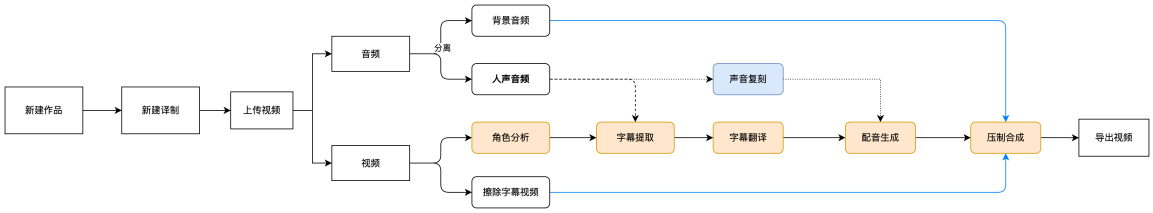
百度AI翻译配音是集成语音识别、文本翻译、声音复刻等能力的短剧翻译配音方案。能够精准复刻视频语音的音色与情绪，实现高保真的跨语言内容转换，适用于短剧译制、短剧营销等场景。

本文介绍如何快速使用AI翻译配音体验平台。

一、AI翻译配音体验平台功能说明



1.1 AI翻译配音平台工作流



1.2 AI翻译配音平台功能介绍

功能	描述
视频管理	提供视频内容存储、管理和检索服务
角色分析	通过人脸检测自动识别视频角色，支持视频角色管理，用于字幕提取
字幕提取	支持OCR/ASR/混合提取三种技术手段提取字幕 *视频字幕识别（OCR）：将原视频的字幕信息转为文本，适用于字幕时间轴准确的视频* 视频语音识别（ASR）：将原视频中的语音信息转为文字，适用于缺少字幕或讲解类视频 * 综合提取（OCR+ASR）：综合利用原视频的字幕信息和语音信息，适用于字幕内容准确但时间轴不够准确的视频
字幕翻译	使用大模型翻译，充分考虑角色一致性、上下文连贯性、语言本地化和语音时长，确保翻译质量
字幕擦除	支持不擦除、标准擦除、高级擦除三种擦除模式 *不擦除：保留视频原有字幕* 标准擦除：使用遮罩擦除原有字幕，在简单视频中表现较好 * 高级擦除：使用动态生成擦除原有字幕，在复杂视频中表现较好
配音生成	通过声音复刻还原角色音色情绪，生成带情感的超真实配音
视频压制	支持设置字幕效果，导出包含原视频原背景音、翻译文本、配音人声的完整视频及相关素材

二、AI翻译配音平台图文全解

2.1 任务创建与管理

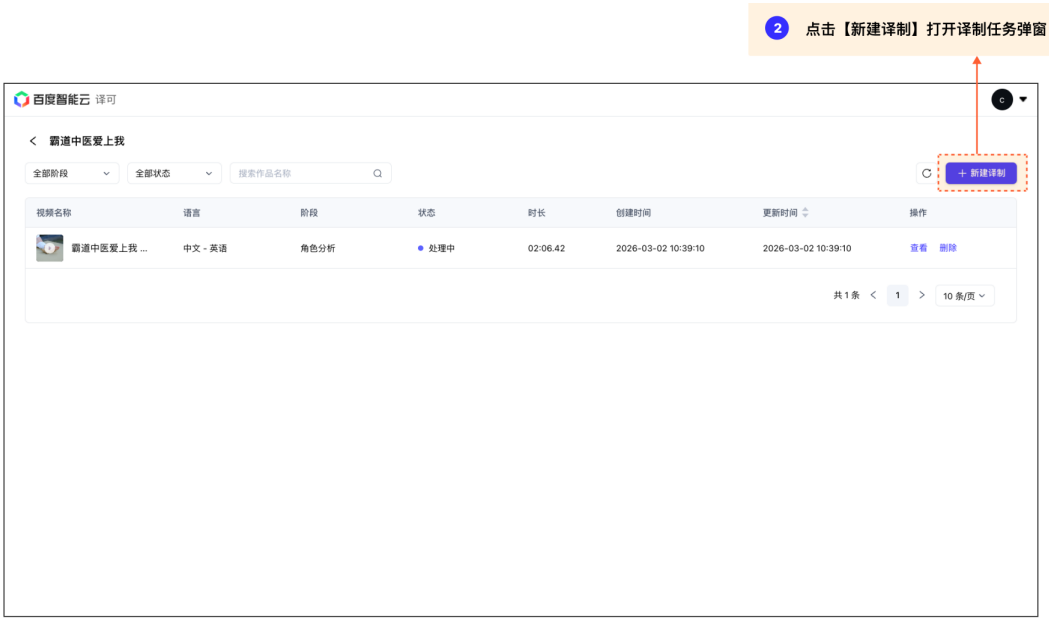
（1）作品创建

首次进入“我的作品”页面，点击“创建新作品”按钮，输入名称创建剧集作品文件夹。



（2）译制任务创建

文件夹建立完毕，进入点击“新建译制”，打开译制任务弹窗。



按照下图提示上传视频，选择翻译语言、审核环节、角色分析类型等参数。选择完毕点击确定，创建译制任务。



注意事项

1. 请上传30～300秒的短视频，目前仅支持单次上传一个视频。
2. 目前已支持横屏和竖屏视频上传，但请上传主流比例的视频。
3. 视频的大小上限为1080x1920，大于此尺寸的视频暂不支持。
4. 目前已支持真人剧与漫剧人脸识别。
5. 目前仅支持英文与印尼语译制，若您希望支持其他语言或视频本地化，请联系商务咨询。
6. 目前仅开放体验平台，接口调用正在建设中，详细范围与接入说明，请联系商务获取。

（3）译制任务管理

任务创建完毕后，支持按照译制阶段、译制状态筛选视频，详情如下图。



译制状态	说明
代执行	任务正在排队中等待执行，刷新页面可更新状态
处理中	任务按照对应环节译制中，失败可点击重试
审核中	任务已完成对应译制环节，等待人工校对。创建任务时选择跳过审核则直接进入下一阶段
完成	任务在对应环节译制成功，可点击继续下一环节或导出译制视频
失败	任务在对应环节译制失败，可点击重试

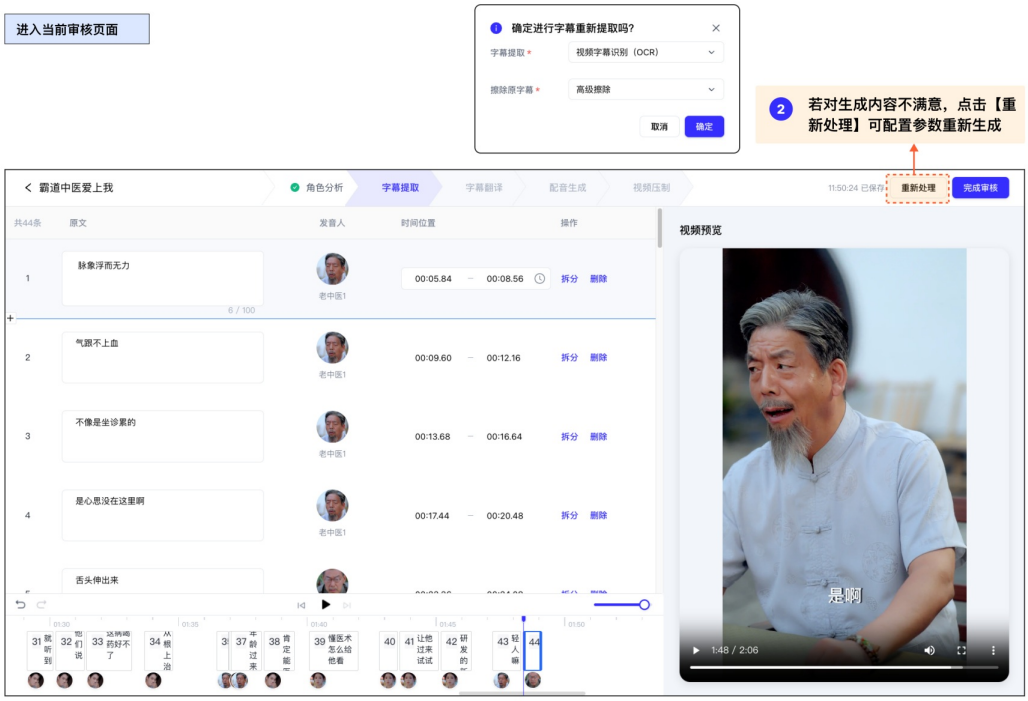
2.2 翻译配音审核

（1）审核与重新处理

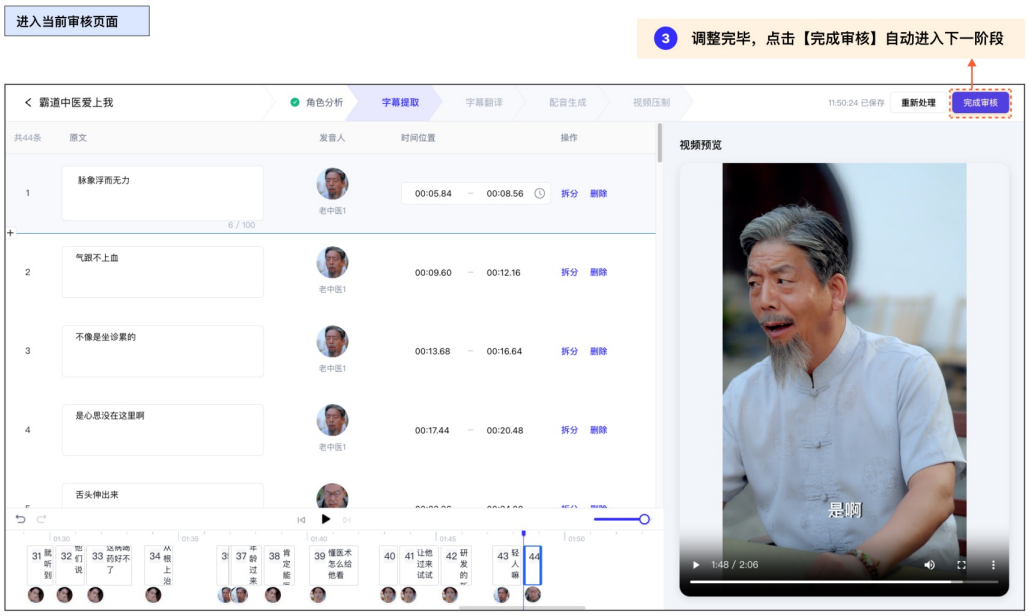
当对应视频状态变为“人工审核中”，点击“查看”了解译制详情。



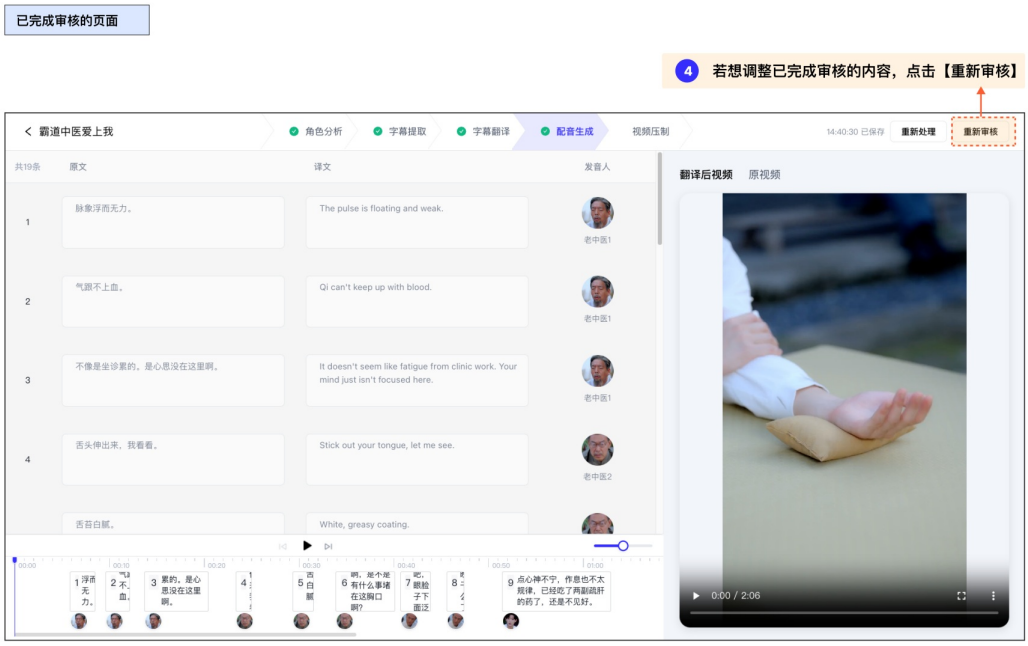
- 若对生成内容整体不满意，或发现漏翻译漏配音情况，可点击“**重新处理**”，配置参数重新生成。



- 若对生成内容部分不满意，可进行调整，调整完毕后点击“**完成审核**”自动进入下一阶段。



- 若希望调整已审核阶段的内容，可点击“**重新审核**”进行调整。

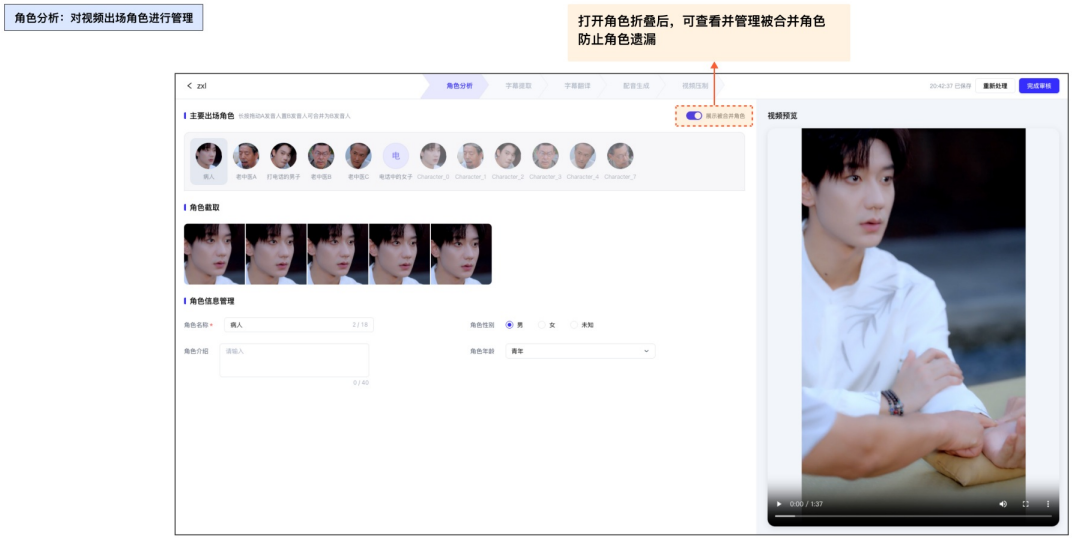


（2）角色分析审核

角色分析阶段，支持对提取角色进行拖动合并与角色信息管理。

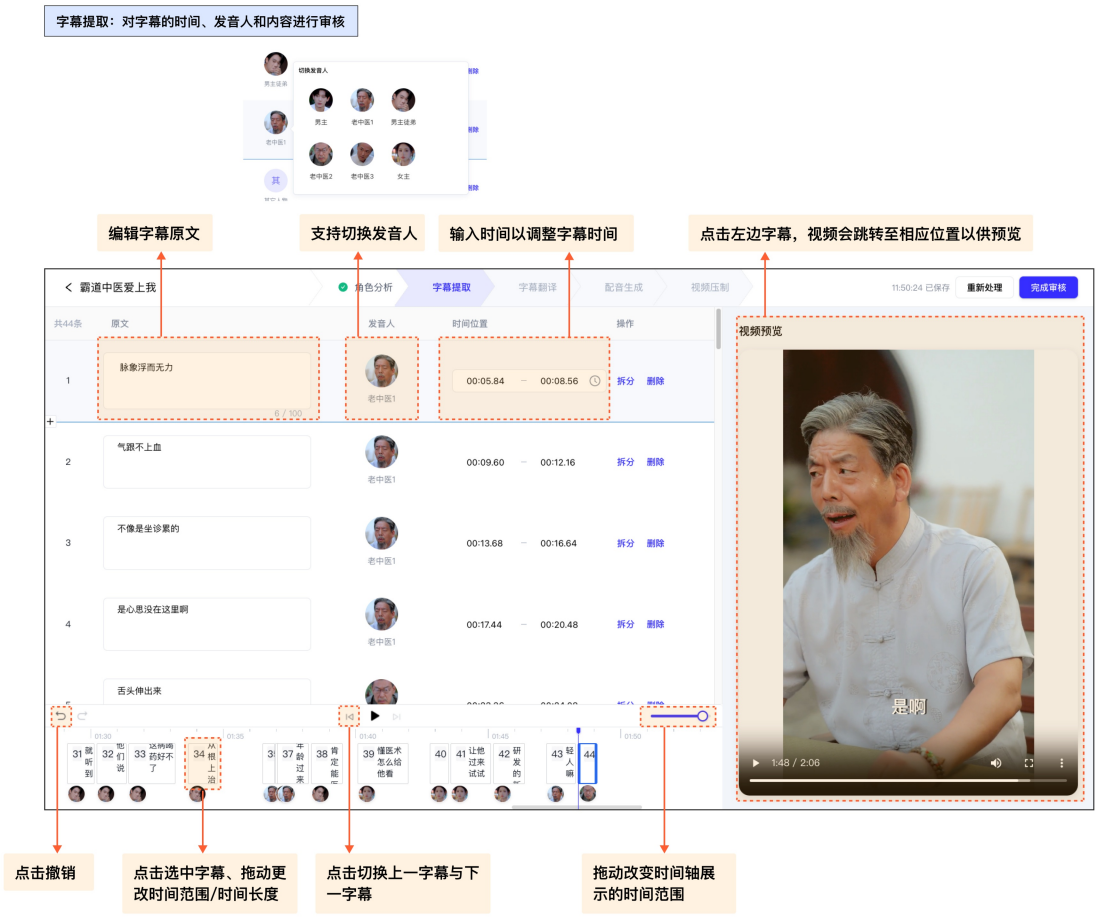


相似角色会被自动合并，系统默认展示合并后的结果。可打开合并角色按钮，查看并管理被合并角色，防止角色遗漏。

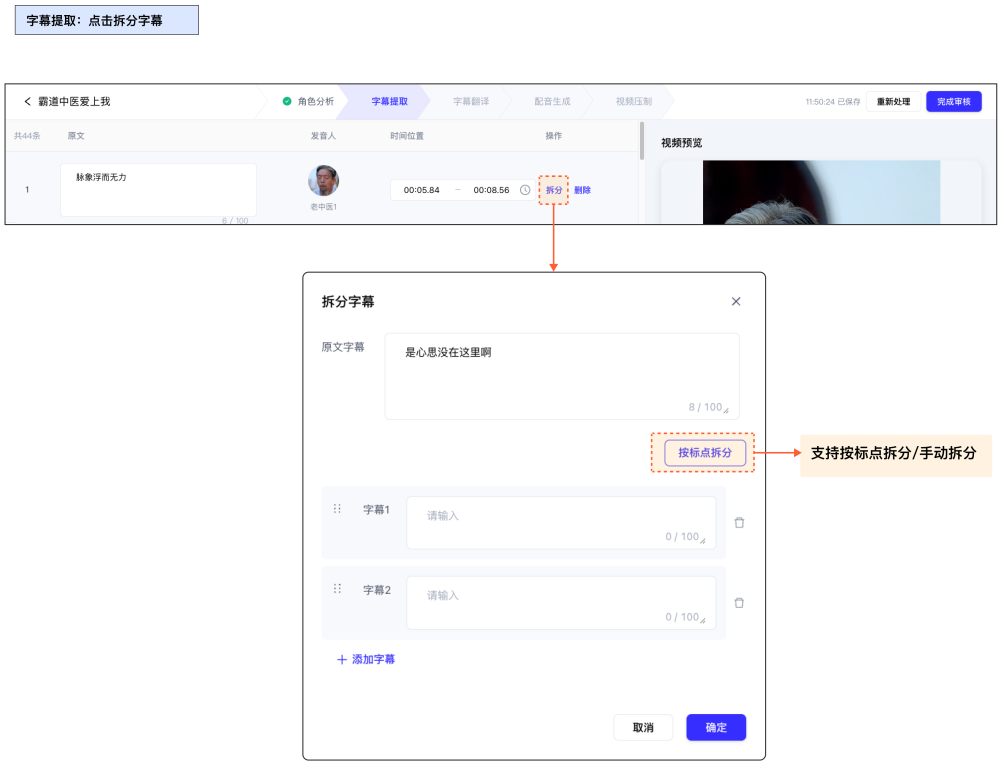


（3）字幕提取审核

如图，字幕提取阶段展示提取原文、发音人、时间位置，支持对以上信息逐条校验。在下方可点击前进、后退按钮切换前后字幕，直接拖动对话框调整时间范围，拖动放大轴改变展示的时间范围。

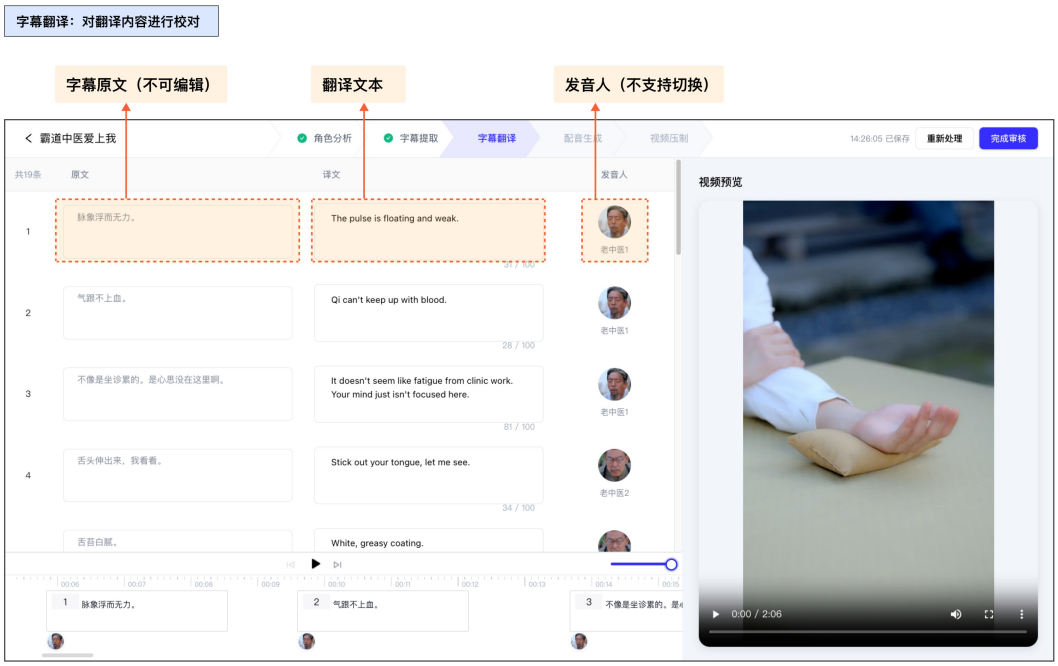


支持对提取字幕的智能拆分。



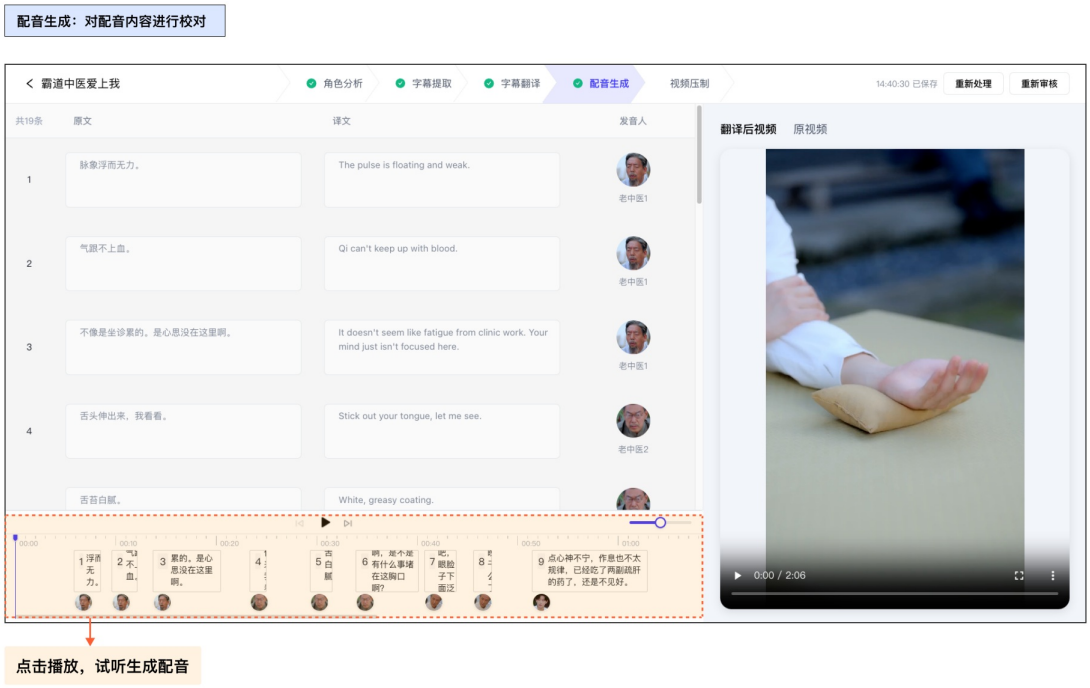
（4）字幕翻译审核

翻译校对阶段，仅支持校对翻译文本，不支持切换发音人。



（5）配音生成审核

配音生成阶段，可点击播放试听配音。



2.3 视频压制与导出

视频压制阶段支持调整生成字幕，可设置字体、字号、颜色等属性。

注意事项

> - 垂直边距指字幕到视频底部的距离，垂直边距为0时默认为最佳边距
> - 背景透明度默认为100%，默认没有背景色



视频预览支持查看字幕+视频+配音+背景音的预览效果，确认无误后点击“导出视频”查看效果。

视频导出后可在压制视频位置查看合成的视频效果。

视频压制：压制完毕后预览并导出视频



API接入AI翻译配音

本文档介绍如何快速使用API调用AI翻译配音服务。

前置条件

在使用API服务之前，请您确保：

- 已完成百度智能云的账号注册及实名认证，详情参考AI翻译配音快速体验指南。
- 已领取AI翻译配音的额度或开通后付费，详情参考AI翻译配音快速体验指南。
- 了解AI翻译配音的核心功能与工作流成，详情请看AI翻译配音使用说明。

接口列表

AI翻译配音统一前缀：https://aip.baidubce.com/rpc/2.0/speech/publiccloudspeech/v1/echo

#	接口	Path	作用	额度校验
1	素材上传	/footage/upload	获取 BOS 临时上传凭证（STS），客户端直传	否
2	素材管理	/footage/manage	素材增删改查（action_type 分发）	否
3	任务创建	/task/create	创建视频 / 任务（4 种模式）	**是**
4	任务管理	/task/manage	任务查询 / 更新 / 确认 / 删除 / 导出（6 种操作）	仅 confirm_task_result
5	任务重试	/task/retry	重试已有任务并可调整配置	**是**

通用约定

鉴权方式

属性	参数名称	类型	是否必填	说明
鉴权令牌	access_token	string	二选一	鉴权令牌
鉴权API Key	Authorization	string	二选一	鉴权API Key

公共响应结构

```
{
  "log_id": "20250622150000XXXX",
  "error_code": 0,
  "error_message": "success",
  "data": {}
}
```

`error_code = 0` 为成功；非 0 时 `error_message` 为失败原因，`data` 通常为 `null`。

错误码总表

error_code	error_message	典型场景
0	success	成功
100001	请求体非法	JSON 解析失败
100002	系统内部错误，请稍后重试	内部异常
100003	无效的参数	参数缺失、枚举非法
100005	账户无效权限	操作非本账号资源
100006	名称重复	素材名 / 剧集名 / 视频名重复
100007	数据更新失败	更新写入失败
100008	数据保存失败	入库失败
100009	数据删除失败	删除失败
100010	数据查询失败	记录不存在
100012	重试次数超过限制	非 failed 任务累计重试达 3 次
100016	任务类型不支持	`export_task` 传入非 compose 任务
100017	任务状态不支持	任务处于 `init`/`doing`
100019	名称长度超限	名称超 50 字符
100025	重复请求过于频繁，请稍后重试	5 秒内重复提交
100026	资源配额剩余不足	前置额度校验未通过
100027	生成上传文件Token失败	STS / SRT token 申请失败
100028	上传文件失败	非 BOS 外链转存失败
100030	数据数量验证失败	删除 ID 数与实际不一致
100032	剧集记录不存在	`series_id` 不存在

枚举值

- **任务类型（task_type）**，常驻链按序推进：`analysis`（分析/角色提取）→ `subtitle`（字幕提取+擦除）→ `translation`（翻译）→ `dubbing`（上屏/配音）→ `compose`（合成）
- **任务状态（task_status）**：`init` 待执行、`doing` 处理中、`failed` 失败、`done` 完成待校验、`checking` 校验中、`checked` 已校验
- **语言**：`source_language` 仅支持 `zh`；`target_language` 支持 `en`、`id`
- **字幕提取器（extractor_type）**：`advance`、`srtFile`；**字幕擦除（subtitle_remove_mode）**：`none`、`standard`、`advanced`
- **人脸类型（face_type）**：`normal`、`anime`

接口描述

一、素材上传

- **Method**：全部 `POST`
- **Content-Type**：`application/json`

获取 BOS 临时上传凭证，用于客户端**直传**文件到 BOS。凭证有效期固定 **1 小时**。本接口仅生成凭证，**不存储素材记录**。

请求参数 |字段|类型|必填|说明| |-|-|-|-| |`file_name`|string|是|文件名，需带后缀（如 `video.mp4`），最大长度 50 字符|

请求示例

```
curl --location 'https://aip.baidubce.com/rpc/2.0/speech/publiccloudspeech/v1/echo/footage/upload' \
--header 'authorization: Bearer xxxxxx' \
--header 'Content-Type: application/json' \
--data '{
    "file_name": "video_sample.mp4"
}'
```

响应字段（data） |字段|类型|说明| |-|-|-| |`access_key_id`|string|临时 Access Key ID，用于 BOS 客户端鉴权| |`secret_access_key`|string|临时 Secret Access Key，用于 BOS 客户端鉴权| |`session_token`|string|临时 Session Token，STS 鉴权必填| |`expiration`|string|凭证过期时间，ISO 8601 格式（如 `2025-06-22T16:00:00Z`）| |`resource`|string|BOS 目标对象路径（不含 bucket），如 `prefix/projects/12345/uuid/video_sample.mp4`| |`bucket_name`|string|BOS Bucket 名称| |`endpoint`|string|BOS Endpoint 地址，如 `https://bos.bj.bcebos.com`|

响应示例

```
{
 "log_id": "20250622150000XXXX",
 "error_code": 0,
 "error_message": "success",
 "data": {
   "access_key_id": "STSxxxxxxxxxxx",
   "secret_access_key": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
   "session_token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
   "expiration": "2025-06-22T16:00:00Z",
   "resource": "prefix/projects/100001/550e8400-e29b-41d4-a716-446655440000/video_sample.mp4",
   "bucket_name": "echo-prod",
   "endpoint": "https://bos.bj.bcebos.com"
 }
}
```

使用流程

1. 调用本接口获取临时凭证；
2. 使用 BOS SDK 或 HTTP PUT，将文件直传到 `endpoint`/`bucket_name`/`resource`，参考 BOS STS 文档；
3. 上传成功后，将 `https://<endpoint>/<bucket_name>/<resource>` 作为 `video_url` 传入任务创建接口。

备注

- 凭证有效期 **1 小时**，超时需重新调用本接口获取新凭证。
- 每次调用生成唯一 UUID 路径，避免多次上传覆盖同名文件。

二、素材管理

- **Method**：全部 `POST`
- **Content-Type**：`application/json`

对素材（Footage）进行统一管理：创建、更新、分页查询、详情、删除。通过 `action_type` 区分操作，同一接口承载多种语义。

请求参数（公共部分） |字段|类型|必填|说明| |-|-|-|-| |`action_type`|string|是|操作类型，枚举：`create`、`update`、`list`、`detail`、`delete`| |`create`|object|条件必填|`action_type=create` 时必填| |`update`|object|条件必填|`action_type=update` 时必填| |`list`|object|条件必填|`action_type=list` 时必填| |`detail`|object|条件必填|`action_type=detail` 时必填| |`delete`|object|条件必填|`action_type=delete` 时必填|

create 对象 |字段|类型|必填|说明| |-|-|-|-| |`footage_name`|string|是|素材名称（同账号下唯一）| |`footage_url`|string|是|素材 URL。若 URL 包含平台 BOS bucket 则视为已有 BOS 对象；否则后端会拉取该 URL 并转存到 BOS（同步上传，大文件或外链较慢时可能超时，推荐先用素材上传接口直传 BOS）| |`footage_desc`|string|否|素材描述|

update 对象 |字段|类型|必填|说明| |-|-|-|-| |`footage_id`|int64|是|素材 ID| |`footage_name`|string|否|新素材名称（同账号下唯一，空串表示不更新）| |`footage_desc`|string|否|新素材描述（空串表示不更新）|

> `footage_name` 与 `footage_desc` 至少传一项。

list 对象 |字段|类型|必填|说明| |-|-|-|-| |`page`|int|是|页码，从 1 开始（1–100）| |`page_size`|int|是|每页大小（1–1000）| |`sort`|string|否|排序方式：`asc`/`desc`，默认 `desc`|

detail 对象 |字段|类型|必填|说明| |-|-|-|-| |`footage_id`|int64|是|素材 ID|

delete 对象 |字段|类型|必填|说明| |-|-|-|-| |`footage_ids`|int64[]|是|待删除素材 ID 列表（不可为空，每个 ID 必须存在且归属当前账号）|

请求示例

```
 // 创建素材
{
  "action_type": "create",
  "create": {
    "footage_name": "角色A参考视频",
    "footage_url": "https://bucket.bj.bcebos.com/path/to/footage.mp4",
    "footage_desc": "角色A 的参考素材"
  }
}

// 更新素材
{
  "action_type": "update",
  "update": { "footage_id": 123456, "footage_name": "角色A参考视频-新版", "footage_desc": "更新后的描述" }
}

// 查询列表
{
  "action_type": "list",
  "list": { "page": 1, "page_size": 20, "sort": "desc" }
}

// 查询详情
{
  "action_type": "detail",
  "detail": { "footage_id": 123456 }
}

// 删除
{
  "action_type": "delete",
  "delete": { "footage_ids": [123456, 123457] }
}
```

响应字段（data） `data` 中只包含与 `action_type` 对应的一个字段，其余省略：

action_type	data 字段	内容
`create`	`create`	`{ "footage_id": 123456 }`
`update`	`update`	`{ "footage_id": 123456 }`
`list`	`list`	`page`、`page_size`、`total`、`list`（Footage 数组）
`detail`	`detail`	Footage 对象
`delete`	`delete`	回显 `footage_ids`

Footage 对象 |字段|类型|说明| |-|-|-| |`footage_id`|int64|素材 ID| |`footage_name`|string|素材名称| |`footage_url`|string|素材的预签名访问 URL（带过期时间）| |`footage_desc`|string|素材描述| |`create_time`|string|创建时间，格式 `yyyy-MM-dd HH:mm:ss`|

响应示例

```
  // list
{
  "log_id": "20250622150000XXXX",
  "error_code": 0,
  "error_message": "success",
  "data": {
    "list": {
      "page": 1,
      "page_size": 20,
      "total": 35,
      "list": [
        {
          "footage_id": 123456,
          "footage_name": "角色A参考视频",
          "footage_url": "https://bucket.bj.bcebos.com/path/...?authorization=...",
          "footage_desc": "角色A 的参考素材",
          "create_time": "2025-06-22 15:00:00"
        }
      ]
    }
  }
}

// detail
{ "log_id": "...", "error_code": 0, "error_message": "success",
  "data": { "detail": { "footage_id": 123456, "footage_name": "角色A参考视频",
    "footage_url": "https://...", "footage_desc": "...", "create_time": "2025-06-22 15:00:00" } } }

// create / update
{ "log_id": "...", "error_code": 0, "error_message": "success", "data": { "create": { "footage_id": 123456 } } }

// delete
{ "log_id": "...", "error_code": 0, "error_message": "success", "data": { "delete": { "footage_ids": [123456, 123457] } }
}
```

备注

- 同一账号 5 秒内不允许提交完全相同请求，否则返回 `100025`。
- 返回的 `footage_url` 是带签名的临时 URL，适合短时间内下载 / 预览；长期请保存素材 ID，后续通过 `detail` 或 `list` 重新获取新的预签名 URL。
- 删除时若传入 ID 数量与实际查到的数量不一致，返回 `100030`（不会部分删除）。

三、任务创建

- **Method**：全部 `POST`
- **Content-Type**：`application/json`

创建视频处理任务，支持四种模式，通过 `action_type` 切换：

action_type	说明	创建内容	使用模板
`single_video`	单视频创建	创建/复用剧集 + 视频 + 任务链首个任务	`video_dubbing_standard`
`batch_video`	批量视频创建	创建/复用剧集 + 多个视频 + 每个视频的首个任务（全成或全败）	`video_dubbing_standard`
`single_task`	单任务创建	对已有视频追加单个阶段的任务	—

> 常驻任务链：analysis → subtitle → translation → dubbing → compose。单视频 / 批量创建只触发首个阶段，后续阶段由配置自动推进 / 用户确认后推进。

请求参数（公共部分） |字段|类型|必填|说明| |-|-|-|-| | action_type|string|是|枚举：single_video/batch_video/single_task| |single_video|object|条件必填|action_type=single_video 时必填| |batch_video|object|条件必填|action_type=batch_video 时必填| |single_task|object|条件必填|action_type=single_task 时必填|

single_video 对象 创建一个新视频并启动完整任务链。series_id 为 0 时自动创建剧集（名称取 series_name，未传则用 video_name + "（系列）"），剧集名重复会报错。

字段	类型	必填	说明
series_id	int64	否	剧集 ID，不传或传 0 则自动创建
series_name	string	否	新建剧集名称（≤50 字符）
series_desc	string	否	剧集描述（≤50 字符）
footage_id	int64	否	关联素材 ID。传入后 video_url 被忽略，使用素材的 BOS 地址（会校验素材归属）
video_url	string	否	视频 URL（footage_id 未传时必填）。若包含平台 BOS bucket 且以 http 开头，后端自动转成 BOS 路径
video_name	string	是	视频名称（≤50 字符，同剧集下唯一）
video_desc	string	否	视频描述（≤50 字符）
source_language	string	是	原始语言，当前仅支持 zh
target_language	string	是	目标语言，当前支持 en、id
analysis_config	object	否	分析阶段配置
subtitle_config	object	否	字幕提取阶段配置
translate_config	object	否	翻译阶段配置
dubbing_config	object	否	配音阶段配置
compose_config	object	否	合成阶段配置（本接口一般不传，由后续阶段使用）

> 配置对象为空时按默认值处理，但传入的字段需通过各自校验。

batch_video 对象 |字段|类型|必填|说明| |-|-|-|-| | series_id|int64|否|剧集 ID。传 0 则用 series_name 创建新剧集| |series_name|string|否|新建剧集名称（≤50 字符）| |series_desc|string|否|剧集描述（≤50 字符）| |source_language|string|是|同 single_video| |target_language|string|是|同 single_video| |video_list|object[]|是|视频参数列表，至少 1 条| |analysis_config / subtitle_config / translate_config / dubbing_config|object|否|子任务配置（同 single_video，作用于列表内所有视频）|

video_list 元素

字段	类型	必填	说明
video_url	string	是	视频 URL
video_name	string	是	视频名称（≤50 字符，**列表内不可重复**，且不可与剧集已有视频重名）
video_desc	string	否	视频描述（≤50 字符）

> 列表内重复名称返回 名称重复；与剧集已有视频重名返回 名称重复,重复视频名称：xxx。

single_task 对象 针对已有视频追加单个阶段的任务。

字段	类型	必填	说明
video_id	int64	是	已有视频 ID
task_type	string	是	任务阶段：analysis/subtitle/translation/dubbing/compose
对应阶段 config	object	条件必填	与 task_type 匹配的阶段配置必填，其余可省略

任务配置子对象（接口 3 / 5 通用）

analysis_config |字段|类型|必填|默认值|说明| |-|-|-|-|-| | need_human_check|bool|否|false|是否需要人工审核| |disable_face_detection|bool|否|false|是否禁用人脸检测| |face_type|string|否|normal|人脸类型：真人 normal、动漫 anime|

subtitle_config

字段	类型	必填	默认值	说明
need_human_check	bool	否	false	是否需要人工审核
extractor_type	string	是	advance	字幕提取器：advance、srtFile
srt_file_url	string	条件必填	空	extractor_type=srtFile 时必填，需为合法 SRT 文件；其他模式不应传
subtitle_remove_mode	string	是	none	字幕擦除模式：none、standard、advanced
language	string	否	同 source_language	字幕语言

translate_config |字段|类型|必填|默认值|说明| |-|-|-|-|-| | need_human_check|bool|否|false|是否需要人工审核| |target_lang|string|否|同 target_language|目标语言代码|

dubbing_config |字段|类型|必填|默认值|说明| |-|-|-|-|-| | need_human_check|bool|否|false|是否需要人工校验| |with_original_vocal|bool|否|false|是否用原人声音频填补空白|

请求示例

```
  // 1) single_video
{
  "action_type": "single_video",
  "single_video": {
    "series_id": 0,
    "series_name": "我的剧集",
    "series_desc": "可选",
    "video_url": "https://bucket.bj.bcebos.com/path/input.mp4",
    "video_name": "第一集",
    "video_desc": "第一集",
    "source_language": "zh",
    "target_language": "en",
    "analysis_config": { "face_type": "normal" },
    "subtitle_config": { "extractor_type": "advance", "subtitle_remove_mode": "none" },
    "translate_config": {},
    "dubbing_config": { "with_original_vocal": false }
  }
}
```

```
  // 2) batch_video
{
  "action_type": "batch_video",
  "batch_video": {
    "series_name": "我的剧集",
    "source_language": "zh",
    "target_language": "en",
    "video_list": [
      { "video_url": "https://.../01.mp4", "video_name": "第一集" },
      { "video_url": "https://.../02.mp4", "video_name": "第二集" }
    ],
    "analysis_config": {},
    "subtitle_config": { "extractor_type": "advance", "subtitle_remove_mode": "none" },
    "translate_config": {},
    "dubbing_config": {}
  }
}

// 3) single_task
{
  "action_type": "single_task",
  "single_task": {
    "video_id": 100001,
    "task_type": "subtitle",
    "subtitle_config": { "extractor_type": "advance", "subtitle_remove_mode": "none" }
  }
}
```

响应字段（data）

按 action_type 返回对应字段：

action_type	data 字段	内容
single_task	single_task	video_id（恒为 0）、task_id、error_code、error_message
single_video	all_task	series_id、video_id、task_id、error_code、error_message
batch_video	batch_task	series_id + list（元素结构同 all_task）

字段	类型	说明
series_id	int64	剧集 ID
video_id	int64	视频 ID（single_task 场景后端未回填，为 0）
task_id	int64	任务 ID（单/批量创建时为首个阶段任务 ID）
error_code	int	单条结果的错误码（0 表示成功）
error_message	string	单条结果的错误信息

响应示例

```
  // single_video
{ "log_id": "...", "error_code": 0, "error_message": "success",
  "data": { "all_task": { "series_id": 1001, "video_id": 200001, "task_id": 3000001 } } }

// batch_video
{ "log_id": "...", "error_code": 0, "error_message": "success",
  "data": { "batch_task": { "series_id": 1001, "list": [
      { "series_id": 1001, "video_id": 200001, "task_id": 3000001 },
      { "series_id": 1001, "video_id": 200002, "task_id": 3000002 }
  ] } } }

// single_task
{ "log_id": "...", "error_code": 0, "error_message": "success",
  "data": { "single_task": { "video_id": 0, "task_id": 3000010 } } }
```

备注

- 5 秒内不允许提交完全相同请求，否则返回 `100025`。
- `single_video` / `batch_video` 只创建首个阶段任务，后续阶段由系统自动推进。
- `single_video.footage_id` 有值时会校验素材归属，并使用该素材的 BOS 地址；此时 `video_url` 不生效。
- `subtitle_config.extractor_type=srtFile` 时，`srt_file_url` 必须指向合法 SRT 文件。

四、任务管理

- **Method**：全部 `POST`
- **Content-Type**：`application/json`

对已创建的任务进行统一管理，通过 `action_type` 区分操作：

action_type	说明
list_video	分页查询当前账号下的视频列表
detail_task	查询任务详情（按 video_id 或 task_id）
update_task_result	更新任务结果（人工修改字幕 / 角色 / 翻译等）
confirm_task_result	确认任务结果（完成人工审核，推动进入下一阶段）
delete_task	删除任务（按 video_id 删除整个视频，或按 task_ids 批量删除任务）
export_task	导出合成任务（compose）的最终视频 URL

请求参数（公共部分） |字段|类型|必填|说明| |-|-|-|-| | action_type|string|是|见上表枚举| |list_video|object|条件必填|page（1–100）、page_size（1–1000）、sort（asc/desc，默认 desc）| |detail_task|object|条件必填|video_id / task_id 二选一| |update_task_result|object|条件必填|见下| |confirm_task_result|object|条件必填|见下| |delete_task|object|条件必填|video_id / task_ids 二选一| |export_task|object|条件必填|task_id（必须是 compose 类型任务）|

detail_task 参数 |字段|类型|必填|说明| |-|-|-|-| | video_id|int64|二选一|视频 ID：返回该视频下全部任务| |task_id|int64|二选一|任务 ID：返回单个任务|

> 至少传一个；都传时以 `video_id` 为准。

update_task_result 参数 更新任务的人工审核结果，仅允许操作归属当前账号的任务。建议先用 `detail_task` 拉取当前结果，在本地修改后整体提交。提交内容会覆盖任务的审核结果（`checked_result`），**任务状态不变**。

字段	类型	必填	说明
task_id	int64	是	任务 ID，必须 >0
analyze_result	object	条件必填	任务类型为 analysis 时必填，见 analyze_result 结构
subtitle_result	object	条件必填	任务类型为 subtitle 时必填，见 subtitle_result 结构
translate_result	object	条件必填	任务类型为 translation 时必填，见 translate_result 结构

> - 无 task_status 字段：本接口不支持修改任务状态，状态流转由 confirm_task_result 与后台调度推进。
> - 与任务类型不匹配的结果对象会被忽略；匹配的结果对象缺失时返回 无效的参数。
> - 同一账号 5 秒内提交完全相同的请求会被拦截（返回 重复请求过于频繁，请稍后重试）。

各任务类型的覆盖语义

任务类型	必填结果对象	覆盖方式
analysis	analyze_result	以提交的 characters **全量替换**原角色列表；按 character_id 匹配原角色以保留后端字段（人脸向量、关联字幕、音色配置）。未提交的角色等同删除
subtitle	subtitle_result	以提交的 subtitles **全量替换**原字幕列表；按 id 匹配原字幕以保留说话人信息。未提交的字幕项等同删除
translation	translate_result	按 id **逐条覆盖**原译文的 text、start_time、end_time、character_id、language、original_text；不新增、不删除条目

confirm_task_result 参数 确认任务结果，完成人工审核并推动任务进入下一阶段。提交的结果处理方式与 update_task_result 完全一致，区别是确认成功后任务状态由 checking 变为 checked，并在同一事务中创建下一阶段任务。

字段	类型	必填	说明
task_id	int64	是	任务 ID，必须 >0
analyze_result	object	条件必填	任务类型为 analysis 时必填，结构见下
subtitle_result	object	条件必填	任务类型为 subtitle 时必填，结构见下
translate_result	object	条件必填	任务类型为 translation 时必填，结构见下

> - 任务必须处于 checking 状态，否则返回 任务状态不支持。
> - 同一账号 5 秒内提交完全相同的请求会被拦截。

delete_task 参数 |字段|类型|必填|说明| |-|-|-|-| | video_id|int64|二选一|视频 ID：删除整个视频及其所有任务| |task_ids|int64[]|二选一|任务 ID 列表：批量删除指定任务|

> 两者至少传一个；都不为空时优先按 video_id 删除。

任务结果对象结构（接口 4 请求与响应通用） analyze_result 结构 |字段|类型|必填|说明| |-|-|-|-| | characters|object[]|是|角色列表，元素结构见 角色对象|

subtitle_result 结构 |字段|类型|必填|说明| |-|-|-|-| | subtitles|object[]|是|字幕项列表，元素结构见 字幕项对象| |characters|map<string, object>|否|角色信息映射，key 为 character_id。**仅响应回显使用**，提交时忽略|

translate_result 结构 |字段|类型|必填|说明| |-|-|-|-| | translated_subtitles|object[]|是|译文字幕项列表，元素结构见 字幕项对象| |characters|map<string, object>|否|角色信息映射，key 为 character_id。**仅响应回显使用**，提交时忽略|

字幕项对象 subtitles、translated_subtitles 的元素结构：

字段	类型	必填	说明
id	string	是	字幕项 ID，用于与原字幕匹配，必须回传
text	string	是	字幕文本（译文场景为译文）
start_time	float	是	起始时间（秒）
end_time	float	是	结束时间（秒）
character_id	string	否	关联角色 ID，旁白固定为 __narrator__
language	string	否	语言代码，如 zh、en
original_text	string	否	原始文本（翻译结果中保存译前原文）

角色对象（characters 元素） analyze_result.characters 的元素结构，提交时需整体回传：

字段	类型	必填	说明
character_id	string	是	角色唯一标识，用于与原角色匹配
name	string	是	角色名称
description	string	否	角色描述
intro_name	string	否	角色介绍中的名字
intro_description	string	否	角色介绍中的描述
avatar	string	否	角色头像（base64）
images	string[]	否	角色关联图片的 BOS 链接列表
merge_id	string	否	合并后的角色 ID（角色被合并时使用）
series_merge_id	string	否	剧集维度合并后的角色 ID
gender	string	否	性别
age	string	否	年龄

> subtitle_result.characters / translate_result.characters / dubbing_result.characters 的 value 为精简角色对象，仅含 character_id、name、avatar、merge_id、series_merge_id。

请求示例

```
 // 查询视频列表
{ "action_type": "list_video", "list_video": { "page": 1, "page_size": 20, "sort": "desc" } }

// 查询任务详情（按 video_id）
{ "action_type": "detail_task", "detail_task": { "video_id": 200001 } }

// 更新任务结果
{
  "action_type": "update_task_result",
  "update_task_result": {
    "task_id": 3000001,
    "task_status": "checked",
    "subtitle_result": {
      "subtitles": [
        { "id": "473e1c93-1b8c-46", "text": "直到死去5年的丈夫站在我面前的时候",
          "start_time": 0.3, "end_time": 2.7,
          "character_id": "1000261_ebee37e3-d9fe-48_cluster_0", "language": "zh" },
        { "id": "a238d69d-3d9e-43", "text": "我才知道原来他没死",
          "start_time": 2.9, "end_time": 4.83,
          "character_id": "1000261_ebee37e3-d9fe-48_cluster_0", "language": "zh" }
      ]
    }
  }
}

// 确认任务结果
{ "action_type": "confirm_task_result", "confirm_task_result": { "task_id": 3000001 } }

// 删除任务
{ "action_type": "delete_task", "delete_task": { "task_ids": [3000001, 3000002] } }

// 导出合成视频
{ "action_type": "export_task", "export_task": { "task_id": 3000100 } }
```

响应字段（data） 按 `action_type` 返回对应字段，其余省略：

action_type	data 字段	内容
`list_video`	`list`	`page`、`page_size`、`total`、`list`（VideoDTO 数组）
`detail_task`	`detail`	`video_id`、`task_id` + 五个阶段的 TaskDTO 数组
`update_task_result`	`update_task_result`	{ "task_id": 3000001 }
`confirm_task_result`	`confirm_task_result`	{ "task_id": 3000001 }
`delete_task`	`delete`	回显 `video_id` / `task_ids`
`export_task`	`export`	`task_id`、`final_url`

VideoDTO |字段|类型|说明| |-|-|-| |`video_id`|int64|视频 ID| |`name`|string|视频名称| |`video_url`|string|视频的预签名访问 URL| |`source_language`|string|原始语言| |`target_language`|string|目标语言| |`duration`|int32|视频时长（秒）| |`description`|string|视频描述| |`create_time`|string|创建时间，`yyyy-MM-dd HH:mm`| |`update_time`|string|更新时间，`yyyy-MM-dd HH:mm`|

detail 响应结构 |字段|类型|说明| |-|-|-| |`video_id`|int64|视频 ID（按 task_id 查询时为任务所属视频 ID）| |`task_id`|int64|任务 ID（按 video_id 查询时为 0）| |`analysis`|TaskDTO[]|分析阶段任务列表| |`subtitle`|TaskDTO[]|字幕提取任务列表| |`translation`|TaskDTO[]|字幕翻译任务列表| |`dubbing`|TaskDTO[]|字幕上屏 / 配音任务列表| |`compose`|TaskDTO[]|视频合成任务列表|

TaskDTO |字段|类型|说明| |-|-|-| |`video_id`|int64|视频 ID| |`task_id`|int64|任务 ID| |`task_type`|string|任务类型（见枚举）| |`task_status`|string|任务状态（见枚举）| |`fail_reason`|string|失败原因| |`need_check`|bool|是否需要人工校验| |`url_info`|object|视频 / 音频相关的 URL 信息| |`created_time`|string|创建时间| |`updated_time`|string|更新时间| |`analysis_config` / `subtitle_config` /

translate_config / dubbing_config / compose_config|object|任务创建时的配置信息| |analyze_result / subtitle_result / translate_result / dubbing_result|object|任务结果（含审核修改后的版本）|

export |字段|类型|说明| |-|-|-| |task_id|int64|任务 ID（export）| |final_url|string|最终合成视频的预签名 URL（export）|

响应示例

```
 // list_video
{ "log_id": "...", "error_code": 0, "error_message": "success",
  "data": { "list": { "page": 1, "page_size": 20, "total": 1, "list": [
    { "video_id": 200001, "name": "第一集",
      "video_url": "https://.../input.mp4?authorization=...",
      "source_language": "zh", "target_language": "en",
      "duration": 600, "description": "",
      "create_time": "2025-06-22 15:00", "update_time": "2025-06-22 15:05" }
  ] } } }

// detail_task
{ "log_id": "...", "error_code": 0, "error_message": "success",
  "data": { "detail": {
    "video_id": 200001,
    "task_id": 0,
    "analysis": [
      { "video_id": 200001, "task_id": 3000001, "task_type": "analysis",
        "task_status": "done", "need_check": false,
        "created_time": "2025-06-22 15:00:00", "updated_time": "2025-06-22 15:03:00" }
    ],
    "subtitle": [], "translation": [], "dubbing": [], "compose": []
  } } }

// update_task_result / confirm_task_result
{ "log_id": "...", "error_code": 0, "error_message": "success",
  "data": { "update_task_result": { "task_id": 3000001 } } }

// delete_task
{ "log_id": "...", "error_code": 0, "error_message": "success",
  "data": { "delete": { "task_ids": [3000001, 3000002] } } }

// export_task
{ "log_id": "...", "error_code": 0, "error_message": "success",
  "data": { "export": { "task_id": 3000100, "final_url": "https://.../final.mp4?authorization=..." } } }
```

备注

- detail_task 并发拉取各任务的审核结果并按 task_id 升序返回；结果中包含 BOS 预签名 URL，请及时使用。
- export_task 仅支持 compose 类型任务，且要求任务已完成（url_info 中存在 final_video_url），否则返回 100016。
- 删除视频（delete_task.video_id）会级联删除该视频下所有任务；批量删除任务（task_ids）不会删除视频本体。
- update_task_result 只写入人工审核结果、**不改变任务状态**；状态流转由 confirm_task_result 与后台调度推进。
- confirm_task_result 要求任务处于 checking 状态，且会做**前置资源额度校验**，额度不足时返回 资源配额剩余不足。

五、任务重试

- **Method**：全部 POST
- **Content-Type**：application/json

适用场景：任务 failed 后修复配置重跑；任务已完成但对结果不满意，换参数重跑。

请求参数 |字段|类型|必填|说明| |-|-|-|-| | task_id|int64|是|任务 ID| |analysis_config|object|否|仅当任务类型为 analysis 时生效| |subtitle_config|object|否|仅当任务类型为 subtitle 时生效| |dubbing_config|object|否|仅当任务类型为 dubbing 时生效|

可覆盖字段白名单 重试时只有以下字段会覆盖到原任务配置，其余字段保持不变：

配置	可覆盖字段	说明
analysis_config	disable_face_detection	是否禁用人脸检测
analysis_config	need_human_check	是否需要人工审核（同时更新任务 need_check）
subtitle_config	extractor_type	advance、srtFile
subtitle_config	subtitle_remove_mode	none、standard、advanced
subtitle_config	need_human_check	同时更新任务 need_check
dubbing_config	with_original_vocal	是否用原人声音频填补空白
dubbing_config	need_human_check	同时更新任务 need_check

请求示例

```
 // 重试字幕任务，并调整提取器与擦除模式
{
  "task_id": 3000002,
  "subtitle_config": { "extractor_type": "advance", "subtitle_remove_mode": "advanced", "need_human_check": false }
}

// 重试失败的分析任务（failed 状态不计重试次数）
{
  "task_id": 3000001,
  "analysis_config": { "disable_face_detection": true, "need_human_check": false }
}

// 仅重试，不调整配置
{ "task_id": 3000003 }
```

响应字段（data） |字段|类型|说明| |-|-|-| | video_id|int64|视频 ID（当前实现未回填，可能为 0）| |task_id|int64|任务 ID（与请求一致）| |error_code|int|错误码（0 表示成功）| |error_message|string|错误信息|

响应示例

```
 {
  "log_id": "202506221500000XXXX",
  "error_code": 0,
  "error_message": "success",
  "data": {
    "task_id": 3000002
  }
}
```

备注

- 同一账号 5 秒内不允许提交完全相同的重试请求，否则返回 100025。
- 重试成功后 **task_id 不变**，可直接复用原 task_id 查询进度。
- failed 状态任务重试不计次；其他终态任务最多重试 3 次。
- translation / compose 阶段的配置目前无法通过本接口修改，需要联系后端支持。

附录：通过 STS 凭证上传文件的 Python 示例

```
 """通过 BOS 的 STS 临时凭证上传文件并返回文件链接"""

import os
import sys
import time
from io import BytesIO
from baidubce.bce_client_configuration import BceClientConfiguration
from baidubce.auth.bce_credentials import BceCredentials
from baidubce.services.bos.bos_client import BosClient

##### ===== 配置区域：填入 /footage/upload 返回的 STS 临时凭证 =====
STS_ACCESS_KEY_ID = "<access_key_id>"
STS_SECRET_ACCESS_KEY = "<secret_access_key>"
STS_SESSION_TOKEN = "<session_token>"
BOS_HOST = "http://bj.bcebos.com"      # 对应响应中的 endpoint
BUCKET_NAME = "<bucket_name>"          # 对应响应中的 bucket_name


def upload_file(local_file_path, object_key=None):
    if not os.path.exists(local_file_path):
        print(f"文件不存在: {local_file_path}")
        return None

    if object_key is None:
        object_key = os.path.basename(local_file_path)

    bos_config = BceClientConfiguration(
        credentials=BceCredentials(STS_ACCESS_KEY_ID, STS_SECRET_ACCESS_KEY),
        endpoint=BOS_HOST,
        security_token=STS_SESSION_TOKEN
    )
    bos_client = BosClient(bos_config)

    # 大于 1GB 用分块上传，否则用简单上传
    file_size = os.path.getsize(local_file_path)
    max_simple_size = 1 * 1024 * 1024 * 1024  # 1GB

    if file_size >= max_simple_size:
        part_size = 100 * 1024 * 1024  # 每块 100MB
        total_parts = (file_size + part_size - 1) // part_size
        upload_id = bos_client.initiate_multipart_upload(BUCKET_NAME, object_key).upload_id

        part_list = []
        with open(local_file_path, 'rb') as f:
            part_number = 1
            while True:
                data = f.read(part_size)
                if not data:
                    break
                response = bos_client.upload_part(
                    BUCKET_NAME, object_key, upload_id, part_number, len(data), BytesIO(data))
                part_list.append({"partNumber": part_number, "eTag": response.metadata.etag})
                print(f"  [{part_number}/{total_parts}] 分块上传完成")
                part_number += 1

        bos_client.complete_multipart_upload(BUCKET_NAME, object_key, upload_id, part_list)
    else:
        bos_client.put_object_from_file(BUCKET_NAME, object_key, local_file_path)

    # 生成预签名 URL（1 小时有效）
```

```
    timestamp = int(time.time())
    signed_url = bos_client.generate_pre_signed_url(
        BUCKET_NAME, object_key, timestamp, expiration_in_seconds=3600).decode('utf-8')
    print(f"上传成功，签名链接（1 小时有效）：\n{signed_url}")
    return signed_url


if __name__ == "__main__":
    if len(sys.argv) < 2:
        print("用法: python upload_by_bos_sts.py <文件路径> [对象路径]")
        sys.exit(1)
```

API文档

语音合成

简介

短文本在线合成

百度短文本在线合成服务，基于HTTP请求的REST API接口，将**文本**转换为可以播放的**音频文件**。

每次请求合成的文本建议**120GBK**以内，如需更长文本转换可使用长文本在线合成。

合成的文件格式为 mp3，pcm（8k及16k），wav（16k），具体见aue参数。 若您需要其它格式，音频文件的转换方法请参考“语音识别工具”=>“音频文件转码”一节

本文档描述了使用语音合成服务REST API的方法。

- 多音字可以通过标注自行定义发音。格式如：重(chong2)报集团。
- 目前只有中英文混合这一种语言，优先中文发音。示例： " I bought 3 books” 发音 “three”; “ 3 books are bought” 发音 “three”; “我们买了 3 books” 发音“三”

长文本在线合成

长文本在线合成接口可以将10万字以内文本一次性合成，异步返回音频。支持多种优质音库，将超长文本快速转换成稳定流畅、饱满真实的音频。适用于阅读听书、新闻播报等客户。 **应用场景 阅读听书：**万字小说一次性合成，可反复播放，给用户带来更流畅、更稳定的听书体验

新闻播报：长篇稿件批量快速合成，释放播音员人力，保障新闻生产时效性 **产品优势**

- **一次性合成：**支持一次性合成10万字，无需拆分文本和拼接音频，并支持下载到本地，大幅节省开发成本
- **优质音库：**为您提供极致拟人、情感饱满的语音合成效果
- **合成速度快：**5万字最快仅需5分钟，合成速度业界领先水平

产品功能

1. 支持中文普通话、简单中英文混读
2. 支持 mp3-16k、mp3-48k、wav、pcm-8k、pcm-16k 等多种音频格式和采样率
3. 支持自定义语速、语调、音量调节
4. 支持多音字标注发音，格式如：重(chong2)报集团。
5. 支持文本分段，段落间增加1s停顿
6. 支持主动查询、自动回调两种方式获取结果

鉴权认证

语音技术主要支持两种鉴权方式：access_token鉴权机制和API Key鉴权机制。二者分别支持的功能范围和适用的用户群体如下：

鉴权机制	支持的功能范围	适用群体
access_token鉴权机制	包括语音技术在内的AI开放能力的所有能力。不支持AI开放能力之外的产品	仅使用AI开放能力的客户
API Key鉴权机制	AI开放能力的公有云接口，暂不支持离线SDK 同时支持大模型服务与开发平台ModelBuilder、大模型应用开发平台AppBuilder的接口调用	要同时使用AI开放能力、ModelBuilder和AppBuilder的客户；或此前在其他平台使用过大模型服务，现在迁移到百度智能云的客户，使用API Key可无缝迁移

需要注意，两种鉴权机制选择其中一种即可，不能在一次请求中同时使用多种鉴权机制。

以下为您分别介绍两种鉴权机制使用方式。

一、access_token鉴权机制

Access_token是用户的访问令牌，承载了用户的身份、权限等信息。鉴权主要分为以下两步：

1.获取AK/SK

2.获取Access_token

- 代码形式→适用于有计算机基础的用户
- 网页调试工具→适用于零基础的用户
- 在线调试工具（推荐）→快速调试接口效果

1. 获取AK/SK 当您成功创建应用后，在对应产品页签下选择“应用列表”，可查看已创建的应用。

平台将会分配给您此应用的相关凭证，主要为**AppID、API Key、Secret Key**。以上三个信息是您应用实际开发的重要凭证，每个应用各不相同，**为了您的财产和服务安全请您妥善保管**。



另外，我们为您提供了教学视频，您可以直接浏览视频获取详细教程。

2. 获取 Access_token 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Access_token参数，Access token默认有效期为30天，获取Access_token的流程如下：

请求URL数据格式

向授权服务地址`https://aip.baidubce.com/oauth/2.0/token`发送请求（推荐使用POST），并在URL中带上以下参数：

- **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`；

例如：

```
https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&client_id=Va5yQRHlA4Fq5eR3LT0vuXV4&client_secret=0rDSjzQ20XUj5itV6WRtznPQSzr5pVw2&
```

获取Access_token的两种方式

接下来为您提供两种获取Access_token的方法，您可以按照自己的需求进行选择。

方式一：通过代码的形式获取Access_token

以下为您提供示例代码。这里以**python语言**为例进行演示。

1. 打开python编译器，输入Access_token示例代码【python】。

Python

Java

PHP

C#

Go

Node.js

C++

```python
import requests
import json


def main():

    url = "https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&client_id=xxxxxx&client_secret=xxxxx"

    payload = ""
    headers = {
        'Content-Type': 'application/json',
        'Accept': 'application/json'
    }

    response = requests.request("POST", url, headers=headers, data=payload)

    print(response.text)
```

2. 在【官网获取的AK】和【官网获取的SK】中输入创建应用后获取的AK、SK。

```python
import requests


def main():
    api_key = 'TZY43Wl                lVRWBd'
    secret_key = 'cOof7Rq                jiV0HHap'
    token_url = 'https://aip.baidubce.com/oauth/2.0/token'

    url = f"{token_url}?grant_type=client_credentials&client_id={api_key}&client_secret={secret_key}"

    payload = ""
    headers = {
        'Content-Type': 'application/json',
        'Accept': 'application/json'
    }

    response = requests.request("POST", url, headers=headers, data=payload)

    print(response.text)


if __name__ == '__main__':
    main()
```

3. 输入完成后运行代码，服务器将返回json文本参数，如下：

- **access_token**：要获取的Access Token；

```
{
    "access_token": "24.a9da8689794deff9a55c637df4ac04cf                282335-39148530",
    "expires_in": 2592000,
    "refresh_token": "25.accc28f6e727040a65fa37a7bfb0e8bd                282335-39148530",
    "scope":
     放 cop_h
     smartapp
     b2b_tp_o
    "session_key": "9mzdWuVJ3a62kcrBDwFoayg0CXCaSj4IG3110WFbGHtU7D+3v+O5veG0WO/pCId+OBH/BzDO8BLAMnoZ3pdC2IlMvcPumw==",
    "session_secret": "dfaf0db2d1a0394adf91f8374a5da02c"
}
```

- **expires_in**：Access Token的有效期(秒为单位，有效期30天)；

```
{
    "access_token": "24.a9da8689794deff9a55c637df4ac04cf                282335-39148530",
    "expires_in": 2592000,
    "refresh_token": "25.accc28f6e727040a65fa37a7bfb0e8bd                282335-39148530",
    "scope":
     放 cop_h
     smartapp
     b2b_tp_o
    "session_key": "9mzdWuVJ3a62kcrBDwFoayg0CXCaSj4IG3110WFbGHtU7D+3v+O5veG0WO/pCId+OBH/BzDO8BLAMnoZ3pdC2IlMvcPumw==",
    "session_secret": "dfaf0db2d1a0394adf91f8374a5da02c"
}
```

- 其他参数忽略，暂时不用;

4. 若请求错误，服务器将返回的JSON文本包含以下参数：

- **error**： 错误码；关于错误码的详细信息请参考下方鉴权认证错误码。
- **error_description**： 错误描述信息，帮助理解和解决发生的错误。

例如，认证失败返回：

```
{
  "error": "invalid_client",
  "error_description": "unknown client id"
}
```

鉴权认证错误码

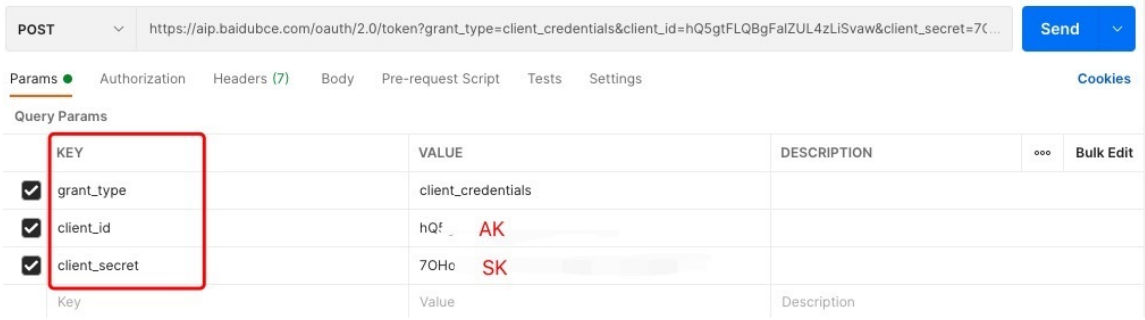
error	error_description	解释
invalid_client	unknown client id	API Key不正确
invalid_client	Client authentication failed	Secret Key不正确

方式二：使用网页调试工具（例如postman）获取Access_token

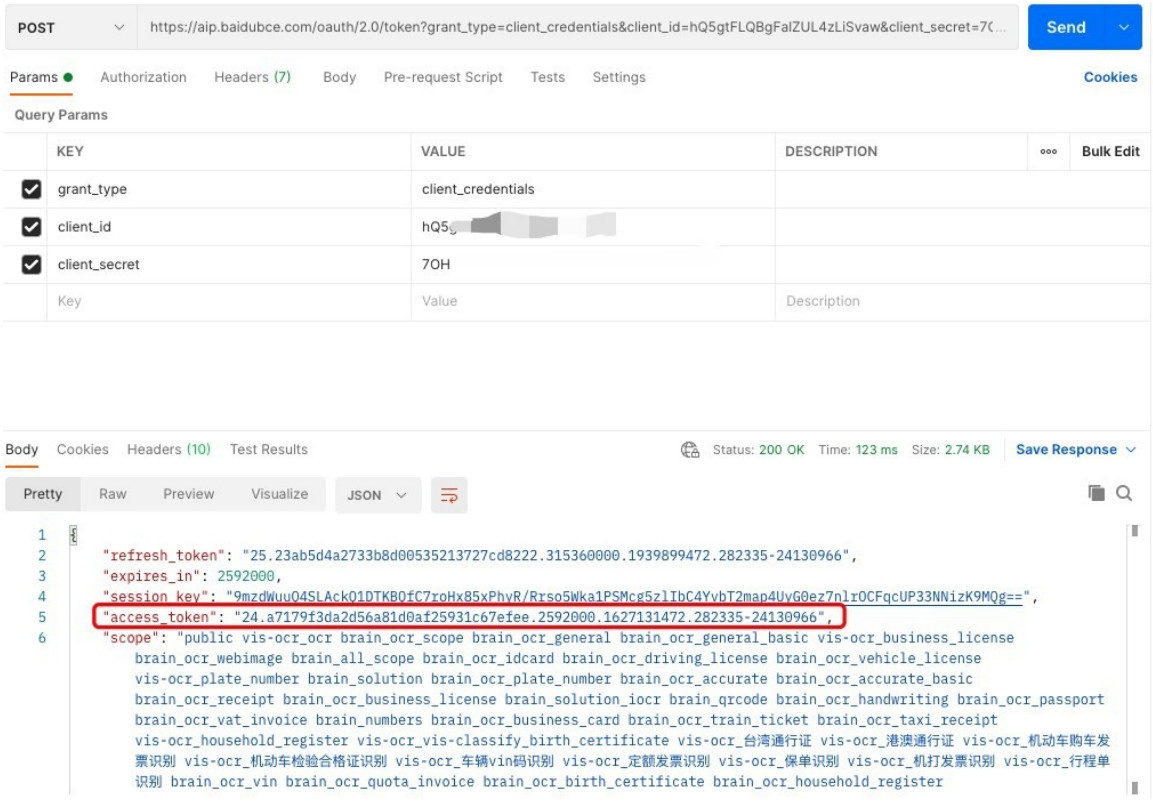
依次在网页调试工具中输入：

- **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；

具体的参数，您可以在控制台应用列表中看到，如果您还不熟悉，请您查看上一步“获取AK/SK”。



输入完成后，点击send，返回json字符串，获取Access_token。例如图中获取的access_token为24.a7179f3da2d56a81d0af25931c67efee.2592000.1627131472.282335-24130966。



另外，为您提供教学视频。您可以点击视频查看详细步骤。

方式三：在线调试工具（推荐）

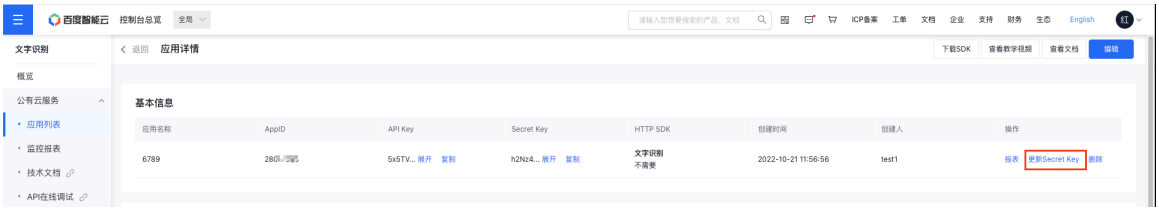
您可以在 示例代码中心 中快速调试接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

3. 密钥安全提示与止损方法

1.请勿将您的AK/SK、以及生成的Access token与他人共享或硬编码到APP及终端，为保护您的资源安全，平台可能会针对恶意滥用token进行禁用。



2.使用http协议兑换token有被截获sk的风险。如发生凭证（即AK/SK或Access token）泄露，您可以在【应用详情】页更新应用的Secret Key。 **请注意：更新后历史生成的Access_token将立即失效，请及时更新运营环境中的Secret Key，避免影响服务调用。**



二、API Key鉴权机制

API Key是百度智能云全新推出的鉴权方式，主要面向以下两个目标：

1. **跨产品调用服务**。同一个API Key可同时调用AI开放能力（文字识别、人脸识别、语音技术等）、大模型服务与开发平台ModelBuilder、大模型应用开发平台AppBuilder的接口服务，降低您接入AI服务的成本。
2. **简化调用步骤**。API Key既可直接使用明文调用接口进行鉴权，也可以换成短期APIkey后再进行鉴权。 为了更高的安全性，减少API Key的明文泄露风险，建议您使用短期APIkey的鉴权方式。

有关API Key的具体介绍和使用方法，请您参考API Key鉴权机制。

音色列表

页面描述 本页面为调用发音人所属音库、发音人名称、调用per参数等信息。

试听demo

试听demo请点击

发音人详情

音库类型	发音人名称	per参数	场景	支持语言	支持情感
大模型音库	度涵竹-开朗女声	4189	超拟人多情感	中文/英文	中性、快乐、悲伤、愤怒、惊讶、恐惧
大模型音库	度嫣然-活泼女声	4194	超拟人多情感	中文/英文	中性、快乐、悲伤、愤怒、惊讶
大模型音库	度泽言-开朗男声	4193	超拟人多情感	中文/英文	中性、快乐、悲伤、惊讶
大模型音库	度怀安-磁性男声	4195	超拟人多情感	中文/英文	中性、快乐、悲伤、惊讶
大模型音库	度清影-甜美女声	4196	超拟人多情感	中文/英文	中性、快乐、悲伤、惊讶
大模型音库	度沁遥-知性女声	4197	超拟人多情感	中文/英文	中性、快乐、悲伤、愤怒、惊讶、恐惧
大模型音库	度泽言-温暖男声	4179	超拟人单情感	中文/英文	
大模型音库	度禧禧-阳光女声	4146	超拟人单情感	中文/英文	
大模型音库	度小柔-温柔女声	6567	超拟人单情感	中文/英文	
大模型音库	度言浩-年轻男声	4156	超拟人单情感	中文/英文	
大模型音库	度言静-明亮女声	4157	超拟人单情感	中文/英文	
大模型音库	度小粤-粤语女声	20100	方言	中文/英文	

大模型音库	度晓芸-粤语女声	20101	方言	中文/英文	
大模型音库	四川小哥-四川男声	4257	方言	中文/英文	
大模型音库	度阿闽-闽南男声	4132	方言	中文/英文	
大模型音库	度小蓉-四川女声	4139	方言	中文/英文	
大模型音库	台媒女声-台湾女声	5977	方言	中文/英文	
大模型音库	度小台-台湾女声	4007	方言	中文/英文	
大模型音库	度湘玉-陕西女声	4150	方言	中文/英文	
大模型音库	度阿锦-东北女声	4134	方言	中文/英文	
大模型音库	度筱林-天津女声	4172	方言	中文/英文	
大模型音库	度阿花-上海女声	5980	方言	中文/英文	
大模型音库	度老崔-北京男声	4154	方言	中文/英文	
臻品音库	懒羊羊	4340	对话助手	中文/英文	
臻品音库	猴哥	4341	对话助手	中文/英文	
臻品音库	熊二	4342	对话助手	中文/英文	
臻品音库	度小鹿-甜美女声	4119	对话助手	中文/英文	
臻品音库	度灵儿-清激女声	4105	对话助手	中文/英文	
臻品音库	度小乔-活泼女声	4117	对话助手	中文/英文	
臻品音库	度晴岚-甜美女声	4288	对话助手	中文/英文	
臻品音库	度青川-温柔男声	4192	对话助手	中文/英文	
臻品音库	度小乐-可爱童声	6561	对话助手	中文/英文	
臻品音库	度米朵-可爱女声	4103	对话助手	中文/英文	
臻品音库	度梓涵-甜美童声	4190	对话助手	中文/英文	
臻品音库	度小清-广告女声	4254	客服营销	中文/英文	
臻品音库	度星河-广告男声	4149	客服营销	中文/英文	
臻品音库	度常盈-客服女声	5147	客服营销	中文/英文	
臻品音库	度舒然-营销女声	340701	客服营销	中文/英文	支持同步语音合成、长文本及流式语音合成
臻品音库	度小飒-直播女声	340702	客服营销	中文/英文	支持同步语音合成、长文本及流式语音合成
臻品音库	度启明-直播男声	340703	客服营销	中文/英文	支持同步语音合成、长文本及流式语音合成
臻品音库	度博文-专业男主播	4106	资讯	中文/英文	
臻品音库	度小贤-电台男主播	4115	资讯	中文/英文	
臻品音库	度小贝-知识女主播	4278	资讯	中文/英文	
臻品音库	度小彦-知识男主播	4129	资讯	中文/英文	
臻品音库	度小新-专业女主播	4140	资讯	中文/英文	
臻品音库	度小雯-活力女主播	4100	资讯	中文/英文	
臻品音库	度常悦-民生女主播	5153	资讯	中文/英文	
臻品音库	南方-电台女主播	4226	资讯	中文/英文	
臻品音库	度忠祥-纪录片男声	4167	资讯	中文/英文	
臻品音库	度小皮-萌娃童声	5976	资讯	中文/英文	
臻品音库	度皮特-老外男声	5971	资讯	中文/英文	

臻品音库	度阿肯-主播男声	4164	资讯	中文/英文	
臻品音库	度有为-磁性男声	4176	资讯	中文/英文	
臻品音库	度小新-播音女声	4259	资讯	中文/英文	
臻品音库	度博文-综艺男声	4206	配音	中文/英文	
臻品音库	度云朵-可爱童声	4147	配音	中文/英文	
臻品音库	度姗姗-娱乐女声	4144	配音	中文/英文	
臻品音库	度清风-配音男声	4143	配音	中文/英文	
臻品音库	度婉婉-甜美女声	4141	配音	中文/英文	
臻品音库	西贝-脱口秀女声	4277	配音	中文/英文	
臻品音库	阿龙-说书男声	4114	配音	中文/英文	
臻品音库	度逍遥-情感男声	4003	小说	中文/英文	
臻品音库	度悠然-旁白男声	6205	小说	中文/英文	
臻品音库	度云萱-旁白女声	6221	小说	中文/英文	
臻品音库	度清豪-逍遥侠客	6546	小说	中文/英文	
臻品音库	度清柔-温柔男神	6602	小说	中文/英文	
臻品音库	度雨楠-元气少女	6562	小说	中文/英文	
臻品音库	度雨萌-邻家女孩	6543	小说	中文/英文	
臻品音库	度书古-情感男声	6747	小说	中文/英文	
臻品音库	度书严-沉稳男声	6748	小说	中文/英文	
臻品音库	度书道-沉稳男声	6746	小说	中文/英文	
臻品音库	度书宁-亲和女声	6644	小说	中文/英文	
臻品音库	度小夏-甜美女声	4148	小说	中文/英文	
精品音库	度逍遥-情感男声	5003	小说	中文/英文	
精品音库	度小鹿-甜美女声	5118	对话助手	中文/英文	
精品音库	度米朵-可爱童声	103	对话助手	中文/英文	
精品音库	度博文-专业男主播	106	资讯	中文/英文	
精品音库	度小童-童声主播	110	资讯	中文/英文	
精品音库	度小萌-软萌妹子	111	小说	中文/英文	
精品音库	度小娇-成熟女主播	5	资讯	中文/英文	
基础音库	度小美-标准女主播	0	资讯	中文/英文	
基础音库	度小宇-亲切男声	1	对话助手	中文/英文	
基础音库	度逍遥-情感男声	3	小说	中文/英文	
基础音库	度丫丫-童声	4	小说	中文/英文	

同步语音合成

接口描述

通过HTTP请求，将短文本文本合成语音音频。采用 application/json 请求，合成成功返回二进制音频流，失败返回 JSON 错误信息。

请求说明

语音合成接口支持POST请求方式。

请求地址：https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/tts

Content-Type：application/json

- 请求标识 sn（可选）：通过 query 参数传入，用于问题排查；不传则由服务端自动生成，并在失败响应中回传。
- 请求体大小：不超过 64KB，文本不超过120 GBK字节，即约60个汉字或者字母数字记为1次请求。每超过120个GBK字节则多记1次计费调用。

如需合成更长文本，推荐使用长文本在线合成。长文本在线合成可将10万字以内文本一次性合成，异步返回音频。支持多种优质音库，将超长文本快速转换 成稳定流畅、饱满真实的音频。适用于阅读听书、新闻播报等客户。

鉴权参数 以下两种方式任选其一，均未提供时返回鉴权失败（错误码 100020）

属性	参数名称	类型	是否必填	说明
access_token	string	query	鉴权方式二选一	鉴权令牌
Authorization	string	header	鉴权方式二选一	鉴权API Key

Body参数

属性	参数名称	类型	是否必填	说明
合成文本	text	string	必填	合成文本应小于1024GBK 字节，1个中文字、英文字母、数字或符号均算作1个字符
音色	voice	string	必填	具体取值范围见音色列表
合成语言	lang	string	可选	当前仅支持中英文，默认为zh
音频格式	format	string	可选	输出文件格式，支持mp3、pcm、wav，默认为mp3格式
采样率	sample_rate	int	可选	支持8000、16000、24000，默认为16000。
音调	pitch	float	可选	音调，取值范围[-12, 12]，默认为0。 取值越大音调约高。音调参数每增加1相当于升调一个单位半音，增加12则升一个八度，降调同理。
音量	volume	float	可选	音量，取值范围[0.5, 2.0]，默认为1。 取值越大，音量越大。0.5代表0.5倍音量，2.0代表2倍音量，默认不调整音量。
语速	speed	float	可选	语速，取值范围[0.5, 2.0]，默认为1。 取值越大，语速越快。0.5代表0.5倍语速，2.0代表2倍语速，默认不调整语速。
合成情感	emotion	string	可选	待合成语音的情感，支持以下情感： 中性：neutral; 高兴：happy; 悲伤：down； 惊讶：surprise; 愤怒：angry; 恐惧：fear； 只有大模型多情感音色支持选择合成情感，支持音色详见音色列表
元数据标识	metadata	object	可选	AIGC元数据标识，仅 mp3 输出时生效。详细字段见下
	metadata.ContentPropagator	string	可选	内容传播服务提供者
	metadata.PropagateID	string	可选	内容传播编号

format 与 sample_rate 支持的组合 |format|sample_rate|输出| |:---|:---|:---| |mp3|16000 / 24000|mp3| |pcm|16000 / 24000|pcm| |pcm|8000|pcm| |wav|16000 / 24000|wav|

若传输的组合不在上表中（如 mp3+8000、wav+8000），接口将返回参数错误。

metadata 说明

- 两个字段都不传：不添加任何元数据标识。
- 同时传入 `ContentPropagator` 与 `PropagateID`：音频中将写入 AIGC 元数据标识（仅 mp3 格式生效）。
- 只传其中一个无效，需成对传入。

请求示例

```
curl -X POST \
'https://<host>/rest/2.0/speech/publiccloudspeech/v1/tts?access_token=<ACCESS_TOKEN>' \
-H 'Content-Type: application/json' \
-d '{
  "text": "百度你好，这里是短文本语音合成接口",
  "voice": "5003",
  "lang": "zh",
  "format": "mp3",
  "sample_rate": 16000,
  "speed": 1,
  "pitch": 0,
  "volume": 1,
  "emotion": "neutral",
  "metadata": {
    "ContentPropagator": "xxxxx",
    "PropagateID": "xxxxx"
  }
}'
```

返回说明

需要根据 Content-Type的头部来确定是否服务端合成成功。

合成成功

若合成成功，返回二进制音频流，`Content-Type` 按 `format` / `sample_rate` 返回：

- 返回为二进制mp3文件，具体header信息 Content-Type: audio/mp3；
- 返回为二进制pcm（16k/24k）文件，具体header信息 Content-Type:audio/basic;codec=pcm;rate=16000/24000;channel=1
- 返回为二进制pcm（8k）文件，具体header信息 Content-Type:audio/basic;codec=pcm;rate=8000;channel=1
- 返回为二进制wav文件，具体header信息 Content-Type: audio/wav；

合成失败

若合成失败，HTTP 状态码固定返回 **200**，错误信息由响应 body 的 `code` 表示，`Content-Type: application/json`：

```
{
 "code": 100002,
 "message": "Invalid parameter.",
 "sn": "49a6b25b-6e2a-11ec-xxxx"
}
```

- `code` / `message`：错误码与描述（见下表）。
- `sn`：请求唯一标识，便于排查。

错误码

错误码	含义	说明
100001	服务端错误	合成服务处理失败，`message` 返回具体原因
100002	参数不合法	参数类错误统一使用此码，`message` 区分具体原因：缺少必填参数(text/voice)、文本超长、format 与 sample_rate 组合不支持、emotion 不合法、speed/pitch/volume 超范围、voice 不合法、请求体超限等
100020	鉴权失败	`access_token` 与 `Authorization` 均缺失或鉴权不通过
100021	无权限或额度不足	发音人无访问权限或账户额度不足
100030	服务内部错误	其它未归类的服务端异常

长文本在线合成

接口描述

长文本在线合成接口可以将10万字以内文本一次性合成，异步返回音频。支持多种优质音库，将超长文本快速转换成稳定流畅、饱满真实的音频。适用于阅读听书、新闻播报等客户。

步骤： 1、根据文本内容、音频格式、音库等参数创建语音合成任务，获取task_id参数。 2、根据task_id的数组批量查询语音合成任务结果。

在线调试&示例代码

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。

Python Demo 点击下载 （文件为.zip压缩包，若无法打开时，可尝试在文件后上增加".zip"后缀）

创建长文本在线合成任务-请求说明

请求接口：https://aip.baidubce.com/rpc/2.0/tts/v1/create

HTTP 方法：POST

URL参数：

参数	取值
access_token	通过API Key和Secret Key获取的access_token，参考Access Token获取

Body中按JSON格式放置请求参数，参数如下：

参数名	类型	是否必需	描述	取值范围
text	list	是	待合成的文本，需要为UTF-8编码；输入多段文本时，文本间会插入1s长度的空白间隔	总字数不超过10万个字符，1个中文字、英文字母、数字或符号均算作1个字符
lang	string	是	语言	固定值zh。语言选择,目前只有中英文混合模式，填写固定值zh
speed	int	否	语速	取值0-15，默认为5中语速
pitch	int	否	音调	取值0-15，默认为5中语调
volume	int	否	音量	音量，基础音库取值0-9，精品音库取值0-15，默认为5中音量（取值为0时为音量最小值，并非为无声）
voice	int	否	音库	基础音库：度小宇=1，度小美=0，度逍遥（基础）=3，度丫丫=4。 精品音库：度逍遥（精品）=5003，度小鹿=5118，度博文=106，度小童=110，度小萌=111，度米朵=103，度小娇=5。默认为度小美 臻品音库：度逍遥（臻品）=4003，度博文=4106，度小贤=4115，度小鹿=4119，度灵儿=4105，度小乔=4117，度小雯=4100，度米朵=4103，度姗姗=4144，度小贝=4278，度清风=4143，度小新=4140，度小彦=4129，度星河=4149，度小清=4254，度博文=4206，南方=4226。 大模型发音人：度涵竹=4189，度嫣然=4194，度泽言=4193，度怀安=4195，度清影=4196，度沁遥=4197，度小粤=20100，度晓芸=20101，四川小哥=4257，度阿闽=4132,度小蓉=4139,台媒女声=5977,度小台=4007,度湘玉=4150,度阿锦=4134,度筱林=4172 **更多发音人请查看音色列表**
format	string	否	音频格式	"mp3-16k"，"mp3-48k"，"wav"，"pcm-8k"，"pcm-16k"，默认为mp3-16k
metadata	string	否	元数据标识	ContentPropagator：内容传播服务提供者 PropagateID：内容传播编号 *ContentPropagator、PropagateID请传入string类型的数据。* *若不传入两个字段，默认不添加元数据标识；传入字段后音频header将添加Label、ContentProducer、ProduceID、ContentPropagator、PropagateID总计5个字段。*
enable_subtitle	int	否	是否开启字幕时间戳	取值范围0, 1, 2，默认为0。0表示不开启字幕时间戳，1表示开启句级别字幕时间戳，2表示开启词级别字幕时间戳 若希望自由控制长文本切分与返回时间戳，可使用^p进行分割（注意分割句间最好不要长于200字），此功能与句级、词级字幕时间戳功能不冲突
break	int	否	段落间隔	取值 0-5000 ，单位ms，用于合成文本分段传入时设置段落间间隔。

请求示例：

```
 {
   "text": [
      "今年上半年我国工业经济面临的内外部环境还是比较严峻复杂的",
      "段落2",
      "段落3"
   ],
   "format": "mp3-16k",
   "voice": 3,
   "lang": "zh",
   "speed": 5,
   "enable_subtitle": 2,
   "break": 0 ,
   "metadata":{
      "ContentPropagator":XXXXX,
      "PropagateID":XXXXX,
   }
}
```

创建长文本在线合成任务-返回说明

返回参数

参数名	类型	是否必需	对外状态
log_id	int	是	log id
task_id	str	否	任务id。**注意保存该id，用于后续请求结果**
task_status	str	否	任务状态
error_code	int	否	错误码
error_msg	str	否	错误信息

返回示例：

```
 创建成功
{
   "log_id": 1234567890,
   "task_id":  "234acb234acb234acb234acb",  #注意保存该id，用于后续请求结果
   "task_status": "Running"
}
创建失败，缺少参数
{
   "error_code": 100000,
   "error_msg": "missing param: xxx",
   "log_id": 5414433131138366128
}
```

注意：查询结果时，需要该步骤返回的task_id来进行请求。请注意保存task_id列表。

查询长文本在线合成任务-请求说明

请求接口：https://aip.baidubce.com/rpc/2.0/tts/v1/query

HTTP 方法：POST

URL参数：

参数	取值
access_token	通过API Key和Secret Key获取的access_token，参考Access Token获取

Body中放置请求参数，参数如下：

参数名	类型	是否必需	描述	取值范围
task_ids	list	是	任务id	推荐一次查询多个任务id，单次最多可查询200个

请求示例：

```
{
  "task_ids": ["234acb234acb234acb234acb", "234acb234acb234acb234acd", "234acb234acb234acb234acbe"]
}
```

查询长文本在线合成任务-返回说明

返回参数：

参数名	类型	是否必需	描述
log_id	int	是	log id
tasks_info	list	否	任务信息
+task_id	str	是	任务id
+task_status	str	是	任务状态
+task_result	dict	否	任务结果
++speech_url	str	否	音频下载链接，任务完成后储存72小时
++speech_timestamp	dict	否	字幕时间戳信息
+++sentences	list	否	句子列表
++++paragraph_index	int	否	句子在传入文本中段落位置
++++sentence_texts	str	否	句子文本信息
++++begin_time	int	否	句子文本在合成音频的开始时间戳
++++end_time	int	否	句子文本在合成音频的结束时间戳
++++characters	list	否	句子中各个词的信息
+++++character_text	str	否	句子中的词文本
+++++begin_time	int	否	句子中的词在合成音频的开始时间戳
+++++end_time	int	否	句子中的词在合成音频的结束时间戳
++err_no	int	否	错误码
++err_msg	str	否	错误信息
++sn	str	否	
error_code	int	否	错误码
error_msg	str	否	错误信息
error_info	list	否	错误的或查询不存在taskId数组

返回示例：

```
{
  "log_id": 16739423288701914,
  "tasks_info": [
```

```
    tasks_info": [
      {
        "task_status": "Success",
        "task_result": {
          "speech_url": "http://bj.bcebos.com/aipe-speech/text_to_speech/2023-01-
17/63c6550e52064d000104da0d/speech/0.mp3?authorization=bce-auth-
v1%2F8a6ca9b78c124d89bb6bca18c6fc5944%2F2023-01-
17T07%3A58%3A12Z%2F259200%2F%2Fbb3f38b53425ced397a107aebe21d2e951ed0e27a964f39c2a350249ba07b47c
,
          "speech_timestamp": {
            "sentences": [
              {   "paragraph_index": 0,
                "sentence_texts": "今年上半年我国工业经济面临的内外部环境还是比较严峻复杂的",
                "begin_time": 104,
                "end_time": 5970,
                "characters": [
                  {
                    "character_text": "今",
                    "begin_time": 106,
                    "end_time": 313
                  },
                  {
                    "character_text": "年",
                    "begin_time": 316,
                    "end_time": 522
                  },
                  {
                    "character_text": "上",
                    "begin_time": 525,
                    "end_time": 732
                  },
                  {
                    "character_text": "半",
                    "begin_time": 735,
                    "end_time": 941
                  },
                  {
                    "character_text": "年",
                    "begin_time": 944,
                    "end_time": 1151
                  },
                  {
                    "character_text": "我",
                    "begin_time": 1154,
                    "end_time": 1360
                  },
                  {
                    "character_text": "国",
                    "begin_time": 1363,
                    "end_time": 1570
                  },
                  {
                    "character_text": "工",
                    "begin_time": 1573,
                    "end_time": 1779
                  },
                  {
                    "character_text": "业",
                    "begin_time": 1782,
                    "end_time": 1989
                  },
                  {
                    "character_text": "经",
```

```
            "begin_time": 1992,
            "end_time": 2198
        },
        {
            "character_text": "济",
            "begin_time": 2201,
            "end_time": 2408
        },
        {
            "character_text": "面",
            "begin_time": 2411,
            "end_time": 2617
        },
        {
            "character_text": "临",
            "begin_time": 2620,
            "end_time": 2827
        },
        {
            "character_text": "的",
            "begin_time": 2830,
            "end_time": 3036
        },
        {
            "character_text": "内",
            "begin_time": 3039,
            "end_time": 3246
        },
        {
            "character_text": "外",
            "begin_time": 3249,
            "end_time": 3455
        },
        {
            "character_text": "部",
            "begin_time": 3458,
            "end_time": 3664
        },
        {
            "character_text": "环",
            "begin_time": 3667,
            "end_time": 3874
        },
        {
            "character_text": "境",
            "begin_time": 3877,
            "end_time": 4083
        },
        {
            "character_text": "还",
            "begin_time": 4086,
            "end_time": 4293
        },
        {
            "character_text": "是",
            "begin_time": 4296,
            "end_time": 4502
        },
        {
            "character_text": "比",
            "begin_time": 4505,
            "end_time": 4712
```

```
                    },
                    {
                        "character_text": "较",
                        "begin_time": 4715,
                        "end_time": 4921
                    },
                    {
                        "character_text": "严",
                        "begin_time": 4924,
                        "end_time": 5131
                    },
                    {
                        "character_text": "峻",
                        "begin_time": 5134,
                        "end_time": 5340
                    },
                    {
                        "character_text": "复",
                        "begin_time": 5343,
                        "end_time": 5550
                    },
                    {
                        "character_text": "杂",
                        "begin_time": 5553,
                        "end_time": 5759
                    },
                    {
                        "character_text": "的",
                        "begin_time": 5762,
                        "end_time": 5969
                    }
                ]
            }
        ]
    },
    "task_id": "63c6550e52064d000104da0d"
  }
 ]
}
```


流式文本在线合成

接口描述

流式文本在线合成基于websocket协议，可以将输入的文本合成为二进制格式的语音数据。与发起多次短文本语音合成相比，可以获得更高的实时性，在用户输入文本的同时就能接近同步的返回合成音频数据，达到“边合成边播放”的效果。

功能说明

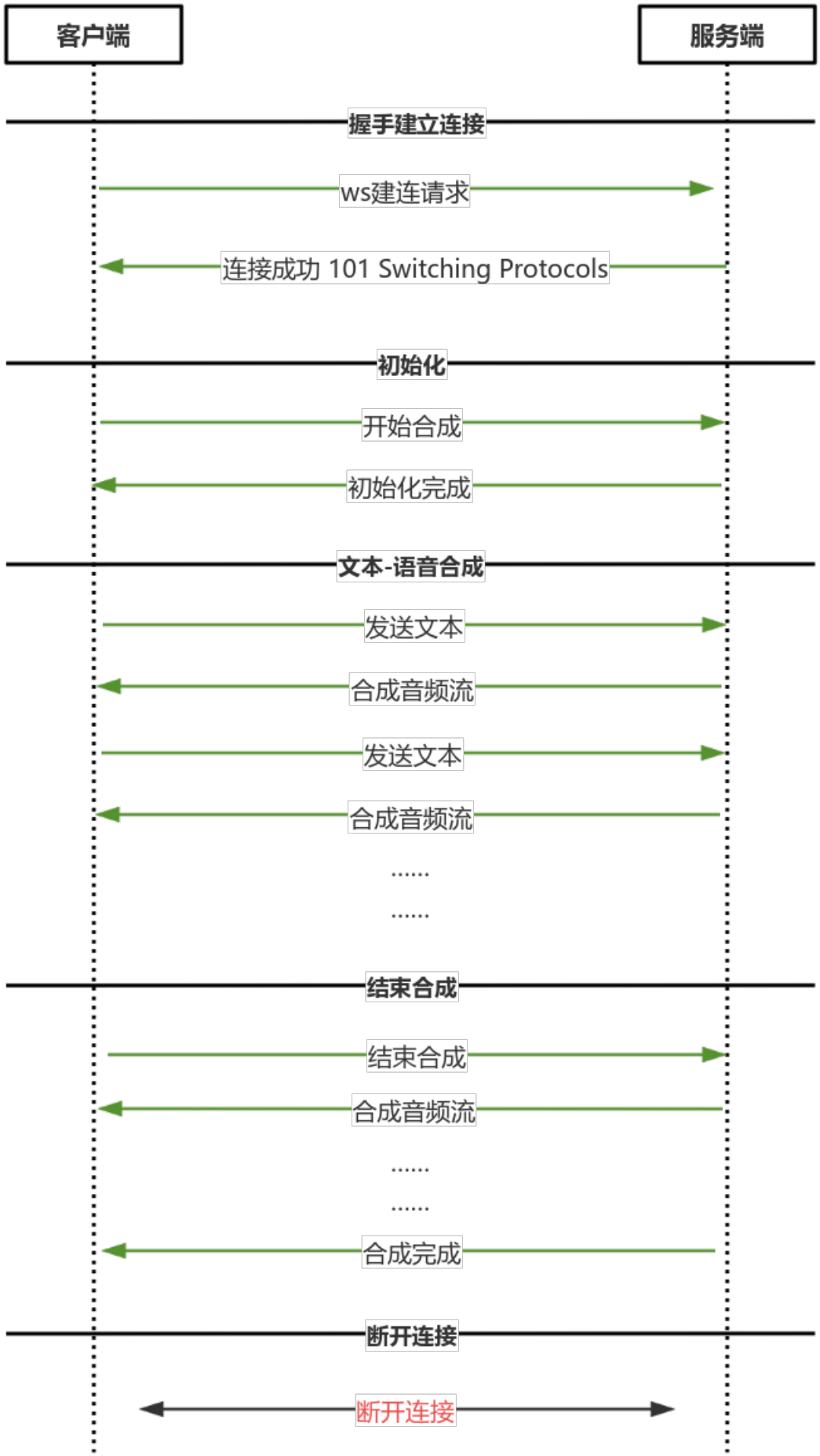
- 建议文本不超过2000 GBK字节，即1000个汉字或者字母数字；
- 输入的文本必须采用UTF-8编码；
- 支持多音字通过标注自行定义发音。格式如：重(chong2)报集团。

示例代码

流式语音合成示例：TTS-ws-demo.zip

接口调用主流程

交互流程


客户端
服务端
握手建立连接
ws建连请求
连接成功 101 Switching Protocols
初始化
开始合成
初始化完成
文本-语音合成
发送文本
合成音频流
发送文本
合成音频流
.....
.....
结束合成
结束合成
合成音频流
.....
.....
合成完成
断开连接
断开连接

握手建立连接

连接请求

握手阶段，客户端主动发起 websocket 连接请求

请求地址

请求地址：wss://aip.baidubce.com/ws/2.0/speech/publiccloudspeech/v1/tts

请求参数

URL中放置请求参数，参数如下：

参数名	类型	参数位置	是否必填	备注
access_token	string	query	二选一	鉴权令牌
Authorization	string	header	二选一	鉴权API Key
per	string	query	必填	tts发音人

请求示例

```
curl --location 'wss://aip.baidubce.com/ws/2.0/speech/publiccloudspeech/v1/tts?access_token=xxx&per=xxx'
```

连接响应

如果握手成功，会返回 101状态码，表示协议握手成功；如果握手失败，则根据不同错误类型返回不同 http status，详细说明如下：

成功：状态码101 Switching Protocols

失败：错误码及描述如下

状态码	状态信息	描述	解决方案
400	Bad Request	参数错误、参数缺失	请参考官网文档，自查token和per
401	Unauthorized	鉴权失败	
403	Forbidden	无访问权限，接口功能未开通	
404	Not Found	输入的url错误	
429	Too Many Requests	触发限流	
500	Internal Server Error	服务器内部错误	
502	Bad Request	后端服务连接失败	

示例

```
 429 Too Many Requests

{
    "code": 276001,
    "message": "Rate limit exceeded error"
}
```

参数说明

参数名	类型	是否必填	说明
code	int	必填	错误码， 使用能力引擎标准的错误码
message	string	必填	错误消息

初始化

握手成功后客户端和服务端会建立websocket连接，客户端通过websocket连接可以同时上传和接收数据。

客户端请求参数

- **帧类型（Opcode）：**Text
- **序列化方式：**JSON

示例

```
{
  "type": "system.start",
  "payload": {
     "spd": 5,
     "pit": 5,
     "vol": 5,
     "audio_ctrl": '{"AIGC": {"ContentPropagator":"XXX","PropagateID":"XXX"},"sampling_rate":16000}',
     "aue": 3
  }
}
```

参数说明

参数名称	类型	是否必填	说明
type	string	必填	开始帧的类型，固定值 system.start
payload	object	可选	合成参数，详见下表

payload 字段说明

参数名称	类型	是否必填	说明
spd	int	可选	语速，取值 0-15，默认为 5
pit	int	可选	音调，取值 0-15，默认为 5
vol	int	可选	音量，基础音库取值0-9，其他音库取值 0-15，默认为 5
aue	int	可选	音频格式，3=mp3-16k/24k，4=pcm-16k/24k，5=pcm-8k，6=wav-16k/24k，默认为3
audio_ctrl	string	可选	*sampling_rate: 采样率,仅支持降采样,示例 {"sampling_rate":16000}* AIGC: 元数据标识，仅支持mp3格式音频添加，默认不添加。 示例：{ "ContentPropagator":"内容传播服务提供者", "PropagateID":"内容传播编号" } *ContentPropagator、PropagateID请传入string类型的数据。* *若不传入两个字段，默认不添加元数据标识；传入字段后音频header将添加Label、ContentProducer、ProduceID、ContentPropagator、PropagateID总计5个字段。*

服务端响应

- **帧类型（Opcode）：**Text
- **序列化方式：**JSON

正常响应

- 服务端对客户端「开始合成」的响应
    - code 等于 0：表示通道各个参数已经设置完成；

- code 不等于 0：表示通道参数设置有误，服务端会断开所有ws连接。

示例

```
 {
   "type": "system.started",
   "code": 0,
   "message": "success",
   "headers": {
      "session_id": "xxx"
           }
}
```

参数说明

参数名称	类型	是否必填	说明
type	string	必填	固定值 system.started，表示开始帧的类型
code	int	必填	错误码，0 表示成功
message	string	必填	错误信息
headers	object	必填	系统定义的请求头
+ session_id	string	必填	sn 信息，格式为 [0-9a-zA-Z_-]{1,64}

异常响应

示例

```
 {
   "type": "system.started",
   "code": 216100,
   "message": "语速参数错误, 请输入0-15的整数",
   "headers": {
      "session_id": "xxx"
           }
}
```

异常响应列表

type	code	message	说明
system.started	216100	参数错误	参数类型错误
system.started	216100	语速参数错误, 请输入0-15的整数	语速取值超限
system.started	216100	音调参数错误, 请输入0-15的整数	音调取值超限
system.started	216100	基础音库音量参数错误, 请输入0-9的整数	基础音库音量取值超限
system.started	216100	精品/臻品音库音量参数错误, 请输入0-15的整数	精品/臻品音库音量取值超限
system.started	216100	音频格式错误, 3:mp3, 4:pcm-16k, 5:pcm-8k, 6:wav	音频格式取值错误

不响应

- 已经发送过开始合成

发送文本

客户端请求

- **帧类型：**Text
- **序列化方式：**JSON

示例

```
 {
   "type": "text",
   "payload": {
      "text": "需要进行语音合成的文字"
   }
}
```

参数说明

参数名称	类型	是否必填	说明
type	string	必填	数据帧的类型，固定值 text
payload	array	可选	具体见下表

payload 字段说明

参数名称	类型	是否必填	说明
+ text	string	必填	需要进行语音合成的文字，不超过1000字

服务端响应

正常响应

```
 WebSocket opcode`：`binary
WebSocket消息体：音频二进制数据
```

异常响应

示例

```
 {
   "type": "system.error",
   "code": 216103,
   "message": "文本过长, 请控制在1000字以内",
   "headers": {
      "session_id": "xxx"
   }
}
```

异常响应列表-文本限制说明

1. 单次发送文本不超过1000汉字，若超过，下发提示帧【文本过长，请控制在1000字以内】

2. 未处理文本长度超过10000汉字后不处理新增文本，下发提示帧【当前待处理文本过长，请稍后发送】

type	code	message	说明
text	216101	参数缺失	缺少必需的参数
text	216103	文本过长, 请控制在1000字以内	单次文本不能超过1000字
text	216419	当前待处理文本过长，请稍后发送	发送频率过快，待处理文本过长

不响应

- 未发送开始合成就发送文本
- 已经发送结束合成仍发送文本
- 发送文本信息无内容

其他响应

- 若发送文本信息无标点隔断，server等待标点或字符数变多（至少60字）再进行合成。有效的隔断标点如下：

```
，.、?!:;，。？！：；—…
```

- 若5~6s内未接收到新文本，则强制以当前文本切句

结束合成

客户端请求

- **帧类型（Opcode）：**Text
- **序列化方式：**JSON

正常响应

- 表示客户端没有文本需要合成了，客户端要求服务端立即合成所有缓存的文本；
- 注意事项：
  - 由于流式文本语音合成接口，服务端会分句合成音频，所以服务端会缓存客户端的文本，客户端需要再没有文本合成时，立刻发送此消息，否则可能丢失文本。
  - 客户端发送此消息后，服务端不在接收任何客户端后续请求。

示例

```
 {
   "type":"system.finish",
}
```

参数说明

参数名称	类型	是否必填	说明
type	string	必填	结束帧的类型，固定值 system.finish

服务端响应

- **帧类型：**Text
- **序列化方式：**JSON

正常响应

- 服务端对客户端「合成结束」的响应，表示服务端已经完成所有文本的合成，且所有音频数据下发完毕。

示例

```
{
  "type":"system.finished",
  "code": 0,
  "message": "success",
}
```

参数说明

参数名称	类型	是否必填	说明
type	string	必填	结束帧的类型，固定值 system.finished
code	int	必填	错误码，0表示成功
message	string	必填	错误信息

异常响应

暂无

不响应

- 文本未合成完成，合成结束后会响应结束

断开连接

正常断开

服务端响应结束合成后，会主动断开连接

异常断开

客户端超过一分钟不发送消息，服务端会异常断开连接

呼叫中心语音-在线合成

接口描述及运行环境

本文档是百度呼叫中心语音MRCP的用户指南。

本程序做为MRCP Server端，集成了呼叫中心8K采样率语音识别(ASR)和呼叫中心专属发音人语音合成(TTS)两种能力，用户可分别单独使用某一种或同时使用。

接入步骤

- 参考"AI接入指南"，创建应用，获取AppID、API Key、Secret Key，用于后续配置使用
- 点击呼叫中心语音解决方案MrcpServer完成Mrcp Server下载；
- 在下载的对应文件中修改相关信息，启动服务并进行程序验证

开发环境

- 开发环境依赖：

- Linux 64位 centos 6u3、centos 7+版本

- gcc: 8.2以上版本libc，CXX11。程序自带compiler.tar.gz压缩包

- 并发受机器内存、核数等性能影响。

  - 参考：4核8G内存，并发数为20。

音频格式

要求项	取值要求
采样率	8KHz
采样精度	16bits
声道	单声道

- 音频内容为：清晰的真人发音，无背景音或其它噪音，日常用语。
- 开发者MRCP Client端发送的音频格式通过SIP协议交互约定，目前mrcp server支持的音频格式有: PCMU PCMA L16/96/8000

语言及模型支持

支持中文普通话

发音人

目前支持九位发音人，详情见附件文档

调用流程

MRCP下载与目录

点击呼叫中心语音解决方案MrcpServer完成MRCP Server下载；

目录结构

```
 ${SERVER_ROOT}
|--- README   // 说明文件
|--- bootstrap.sh  // 环境初始化
|--- compiler.tar.gz  // gcc-8.2编译器lib库
|--- mrcp-server  // 主程序模块
 |--- audio  // 音频保存目录
 |--- bin    // 服务可执行程序
  |--- unimrcpserver_control // 启动脚本
 |--- conf   // 配置文件
  |--- unimrcpserver.xml  // 可执行程序框架配置项
  |--- unimrcpserver_control.conf // 启动配置项
  |--- mrcp-asr.conf   // 识别插件配置项
  |--- mrcp-proxy.conf  // 合成插件配置项
  |--- comlog.conf   // 日志配置项
 |--- data    // 资源文件
 |--- lib     // 依赖库
 |--- log     // 日志目录
 |--- plugin   // 插件目录
```

鉴权与IP参数配置

用户首先需进行相关配置以启动程序。一般地，配置采用默认值即可，需要用户修改的主要有:

- 在主程序配置文件 ${SERVER_ROOT}/mrcp-server/conf/unimrcpserver.xml 中配置本程序IP，具体位置：unimrcpserver->properties->ip，可选用多种方式，只能同时使用一种：

```
 * type="auto"，由程序自动获取机器IP。默认配置，用户无需修改。
* type="iface"，填写网口名称，较少使用。
* 直接手动填写IP，如：<ip>10.10.0.1</ip>，可以使用。
```

- 语音识别配置conf/mrcp-asr.conf中更改AUTH_APPID和AUTH_APPKEY为从百度官方获取的APPID和API Key的值。
- 语音合成配置conf/mrcp-proxy.conf中更改AUTH_APPID和AUTH_APPKEY为从百度官方获取的APPID和API Key的值。
- 启动配置文件 ${SERVER_ROOT}/mrcp-server/conf/unimrcpserver_control.conf，用于监测相应IP和端口，判断程序是否启动成功。搜索到_check_cmd_pro="./bin/check 127.0.0.1 1544"的位置：

```
 * 如果主程序配置IP采用默认type="auto"方式，则该处"127.0.0.1"不用修改。
* 如果主程序配置IP使用手动填写IP方式，则该处的IP值修改为与它相同的值。
```

详细的配置说明见模块内README文件-${SERVER_ROOT}/mrcp-server/README

服务启动

1. 初次下载MRCP server安装包，需要在 ${SERVER_ROOT}/ 目录下，以 **root** 权限执行 bootstrap.sh 脚本，以完成百度自带gcc8.2环境配置。
2. 程序调试阶段，建议在程序目录${SERVER_ROOT}/mrcp-server/下，手动使用命令"./bin/unimrcpserver -r . &"启动程序，方便查看输出、定位问题。使用netstat -nlp | grep unimrcp，查看IP和端口5060/1544/1554，看是否启动成功。如果公司有防火墙限制，请记得将这三个端口打开。
3. 使用启动脚本，以守护进程形式启动程序。在生产环境使用时，建议使用该方式：
    - 启动：在${SERVER_ROOT}/mrcp-server目录执行 ./bin/unimrcpserver_control start
    - 停止：在${SERVER_ROOT}/mrcp-server目录执行 ./bin/unimrcpserver_control stop
    - 重启：在${SERVER_ROOT}/mrcp-server目录执行 ./bin/unimrcpserver_control restart
4. 进行start前确保系统无MRCP进程；进行stop/restart时确保系统有MRCP进程。如果不行，通过 ps aux | grep mrcp 尝试将所有MRCP相关的进程kill掉，重新 start。

请求说明

如需使用语音合成能力，则需在下载的Mrcp Server中对合成配置文件进行参数更改 {SERVER_ROOT}/mrcp-server/conf/mrcp-proxy.conf。确保AUTH_APPID和AUTH_APPKEY填写正确。

- AUDIO_CONTROLLER_ADDR，百度上游服务地址(默认值当前有效)。
- AUTH_APPID和AUTH_APPKEY，从百度官方获取的APPID和API Key的值。
- NEED_SAVE_AUDIO，是否保存合成语音，默认1为保存。
- TR_ENABLE，默认为1表示开启合成文本正则替换功能，详见./data/rules.dat，注意，正则替换原则是按配置从上到下逐条进行匹配，因此建议将泛化能力强的正则放在上面，配置不当下面的正则会使上面的失效，请用户自行体验。不需要该功能可以关闭。
- 目前配置中，支持九种精品发音，用户呼叫软件在进行合成请求时，请填写对应音色参数。

在线合成 - 配置MrcpServer

配置音色

1. 在conf/mrcp-proxy.conf中按照如下格式配置发音人：

> - 该步骤为必要步骤。需要配置完成后，才能使用发音人。

```
 [@VOICE_INFO]
NAME：fduxiaowen###
ID：100

[@VOICE_INFO]
NAME：duxiaoxia
ID：5130

[@VOICE_INFO]
NAME: duxiaoxi
ID: 5117

[@VOICE_INFO]
NAME: duxiaowei
ID: 5120

[@VOICE_INFO]
NAME: duxiaojin
ID: 5221

[@VOICE_INFO]
NAME: duxinghe
ID: 5131

[@VOICE_INFO]
NAME: duxiaoyao
ID: 3

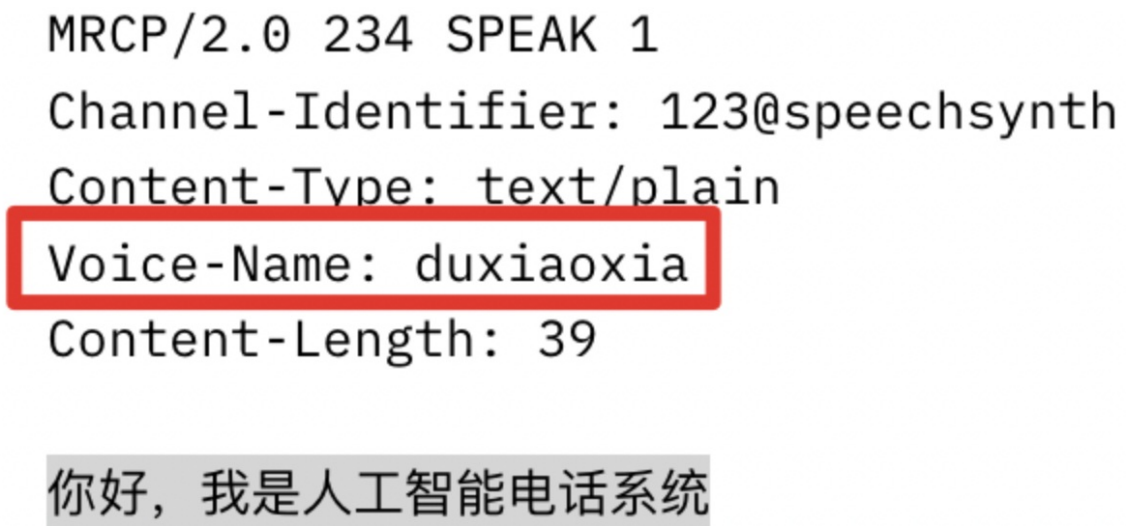
[@VOICE_INFO]
NAME: duxiaoqiao
ID: 1117

[@VOICE_INFO]
NAME: dulinger
ID: 105
```

2. 在MRCP SPEAK请求的header中设置Voice-Name，对应第一步中的NAME字段的配置

比如下图中将使用"duxiaoxia"发音人，ID：5130，mrcp client中设置SYNTHESIZER_HEADER_VOICE_NAME即可。
示例：

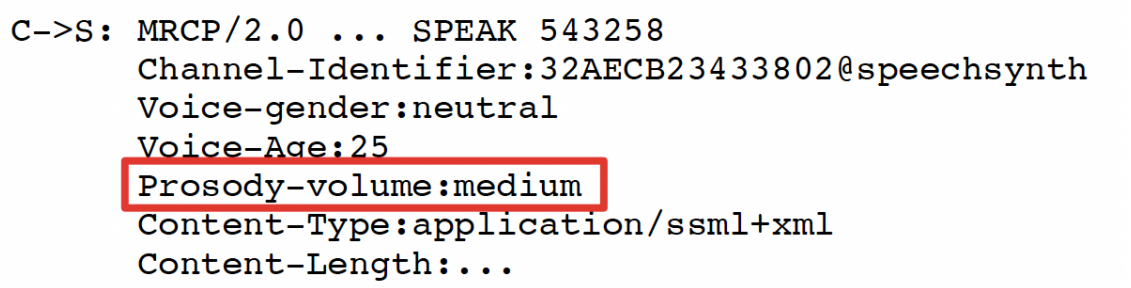
```
 <resource-header-field key="SYNTHESIZER_HEADER_VOICE_NAME"
value="duxiaoxia"/>
```



3.如果没有设置MRCP SPEAK中header的Voice-Name字段，则默认使用conf/mrcp-proxy.conf中DEFAULT_SPEAKDER_ID对应的发音人。

合成效果 在mrcp-proxy.conf中可以通过修改参数更改效果。通过对VOICE_PITCH（音高）、VOICE_VOLUME（音量）和VOICE_SPEED（语速）参数的调整，可以获得不同的发声效果，更好满足您业务场景中的播报需求。
如音高稍高，声音听起来会显得年轻。 **设置音高** 1.默认的音量值见conf/mrcp-proxy.conf中VOICE_PITCH对应的值，默认为5
2.在MRCP SPEAK请求的header中设置Prosody-pitch，可修改音高值
Prosody-pitch对应的value值支持3种类型
第一类：xlow，low，medium，high，xhigh，default几种，音高由低到高
第二类：0到9共10个数字，音高由低到高
第三类：+或者-后跟数字再跟单位h或t，如+100h表示相对调高音高100hz，-50t则表示相对降低音高50st

设置音量 1.默认的音量值见conf/mrcp-proxy.conf中VOICE_VOLUME对应的值，默认为5
2.在MRCP SPEAK请求的header中设置Prosody-volume，可修改音量值，如下图所示



Prosody-volume对应的value值支持3种类型
第一类：silent，xsoft，soft，medium，loud，xloud，default几种，音量由低到高
第二类：0到9共10个数字，音量由低到高
第三类：+或者-后跟数字，如+1表示相对调高音量1，-1则表示相对降低1

设置语速 和设置音量类似
1.默认的语速值见conf/mrcp-proxy.conf中VOICE_SPEED对应的值，默认为5
2.在在MRCP SPEAK请求的header中设置Prosody-rate，可修改语速值
Prosody-rate对应的value值支持2种类型
第一类：xslow，slow，medium，fast，xfast，default几种，语速由低到高
第二类：0到9共10个数字，语速由低到高

忽略MRCP请求header中性别，音量，语速，音高参数 修改mrcp-proxy/conf/mrcp-proxy.conf，修改后需重启

```
 ##### 是否忽略性别, 0:不忽略(默认); 1: 忽略
IGNORE_GENDER_IN_MRCP_HEADER : 0
##### 是否忽略语速, 0:不忽略(默认); 1: 忽略
IGNORE_SPEED_IN_MRCP_HEADER : 0
##### 是否忽略音量, 0:不忽略(默认); 1: 忽略
IGNORE_VOLUME_IN_MRCP_HEADER : 0
##### 是否忽略音高, 0:不忽略(默认); 1: 忽略
IGNORE_PITCH_IN_MRCP_HEADER : 0
```

程序验证

首先，需要将程序lib库加入环境变量中，`export LD_LIBRARY_PATH=${SERVER_ROOT}/mrcp-server/lib:$LD_LIBRARY_PATH`，注意将${SERVER_ROOT}修改为程序真实路径。 在主程序启动后，可使用自带的测试工具进行验证。conf/client-profiles/unimrcp.xml是测试工具的配置文件，需要将其中的unimrcpclient->settings->sip-settings->server-ip的值修改为主程序配置的IP，端口设置为主程序端口，如5060。

切换到 ${SERVER_ROOT}/mrcp-server/bin 目录下。

验证语音合成正确性，则执行 `./unimrcpclient`，输入 `run synth`，等待一段时间，确保合成结束。合成的是 `mrcp-server/data` 目录下的speak.xml文本，查看audio目录下是否有相应音频，如有错误，log目录下日志mrcp_debug.log可以看到相关信息。

注意，在输入识别或合成命令后，等待一段时间，确保识别或合成结束，再使用 `quit` 退出。使用`help`查看帮助。

返回说明

如果配置文件设置了NEED_SAVE_AUDIO，音频会在audio目录下保存，名称为{sn}.pcm，当前请求的sn可在日志文件mrcp_debug.log里查看。

附

音色名称	采样率	音色per	合成文本	合成样音
度小希	8000	5117	您有什么问题可以随时告诉我，我会尽我所能为您提供满意的解答和服务。期待能为您带来愉快的沟通体验，祝您一切顺利！	度小希
度小唯	8000	5120	感谢您对我们的支持！祝您购物愉快！如果后续有任何问题，随时联系我哦，我会一直在这里等您的！再见啦！	度小唯
度小夏	8000	5130	明亮的火烛，将寝室里照得亮如白昼。不知何处来了一阵风，烛火摇曳不定，在洁白的墙壁上投下飘忽不定的阴影。躺在凤榻上的陆皇后面色晦暗，猛地咳嗽几声，转头吐出一口血。	度小夏
度小璟	8000	5221	欢迎来到客服中心，我是您的客服助手。为了更快更好地帮助您解决问题，请问您是关于哪方面的咨询呢？	度小璟
度星河	8000	5131	最近天气变化大，要注意保暖，别着凉了。另外，您有什么需要帮忙的吗？比如查询信息或者安排日程，尽管告诉我，我会尽力帮您解决。	度星河
度小雯	8000	100	中国科学院南京地质古生物研究所网站1月21日消息，生物发光现象广泛地存在于各类生物中。在陆生动物中，能进行生物发光的物种大多都属于鞘翅目昆虫。	度小雯
度逍遥	8000	3	林荒大吼出声，即便十年挣扎，他也从未感到过如此无助。自己的身体一点点陷入岁月之门，却眼睁睁的看着君倾城一手持剑，雪白的身影决然凄厉。就这样孤身一人，于漫天风雪中，对阵数千武者。	度逍遥
度小乔	8000	1117	北京现在气温-1℃，晴，西南风1级，空气质量良，空气质量指数81。	度小乔
度灵儿	8000	105	前方路口左转，距离目的地还有500米，请小心驾驶。	度灵儿

SSML标记语言使用说明

页面描述

SSML即语音合成标记语言（Speech Synthesis Markup Language）。本页面介绍语音合成、声音复刻如何使用SSML对读音、音效进行定制化。

注意事项

- **使用SSML标签时，短文本合成与长文本合成API 务必在请求体中传入xml=1，声音复刻API无需传参。**
- 流式在线合成API、声音复刻流式合成API目前不支持SSML。
- 声音复刻HTTP合成API暂不支持<audio>、<background>标签的使用，暂不支持通过<phoneme>指定单词读音，但支持指定中文读音。
- 声音复刻HTTP合成API仅支持中文普通话音色调用SSML，暂不支持方言、小语种调用。
- 使用SSML标签时合成字符不要超过150（包含标签本身），避免出现badcase。

标签概述

标签	属性	功能	备注
<speak>		根元素 设置全局属性	根元素是必需的。若无该元素，输入文本将不被认为是SSML
<phoneme>	py	指定中文词的读音	声音复刻仅支持中文读音，暂不支持指定单词读音
<say-as>	type	设置数字读法	支持整数、数字串、比分等一系列读法
<say-as>	interpret-as	设置英文读法（包括按字母读、按词读）	
<sub>	alias	文本替换	等价于将其内部文本替换为alias属性中的文本
<silence>	time	插入静音	
<audio>	src	在文本后插入音频或音乐	
<background>	src repeat	插入同时播放的音频或音乐	

标签详情

<speak>根元素 解释

<speak>为ssml标注文本的边界，用于设置文本层级。

<speak>为强制标签，ssml标注文本必须要加上<speak>边界。

<sub>：替换文本 解释

<sub>用于替换选中的内容为指定文本

属性	功能	取值
alias	替换文本	中文或字母，不支持包含数字

- 使用示例：

```
<speak>百度的下一个使命聚焦在<sub  alias="人工智能">AI</sub>上。</speak>
```

<phoneme>：指定发音 解释

<phoneme>用于精确指定中英文词语的发音，以纠正多音字/单词在合成时出现的不准确发音

属性	功能	取值
py	纠正中文读音	音素 + 声调数字 如zhao1
	纠正单词读音	ARPAbet 音素 如ax 0, g eh n 1

使用示例

```
 <speak>迎着<phoneme  py="zhao1">朝</phoneme>阳的<phoneme  py="chao2">朝</phoneme>阳群众</speak>
<speak><phoneme py="ax 0, g eh n 1">word</phoneme></speak>
```

<say-as>：指定文本读法 解释

<say-as>用于指定数字、英文的读法类型

指定数字读法 type用于指定数字、符号等的读法类型，以纠正数字串在合成时出现的不准确读音

属性	属性值	功能支持	使用对象
type	number:ordinal	整数读法	纯阿拉伯数字串、带有英文逗号的纯阿拉伯数字串，数字个数（英文逗号不计数）小于13
	number:digits	数字串读法	纯阿拉伯数字串
	number:score	比分读法	带有一个冒号或中划线-表示的比分读法，冒号或中划线-表前是纯阿拉伯数字串、带有英文逗号的纯阿拉伯数字串，冒号或中划线-表后也是
	number:fraction	分数读法	
	telephone	电话号码读法	
	number	整数读法 负整数读法 浮点数读法 负浮点数读法 比分读法 分数读法 负分数读法 科学计数读法	

使用示例

加标签示例	处理后文本
<say-as type="number:ordinal">123</say-as>	一百二十三
<say-as type="number:ordinal">123000</say-as>	十二万三千
<say-as type="number:ordinal">123,000</say-as>	十二万三千
<say-as type="number:ordinal">123000000</say-as>	一亿二千三百万
<say-as type="number:ordinal">123,000,000</say-as>	一亿二千三百万
<say-as type="number:ordinal">123000000000</say-as>	一千两百三十亿
<say-as type="number:ordinal">123,000,000,000</say-as>	一千两百三十亿
<say-as type="number:ordinal">123000000000000</say-as>	-
<say-as type="number:ordinal">123,000,000,000,000</say-as>	-
<say-as type="number:digits">123</say-as>	一二三
<say-as type="number:digits">123,000</say-as>	一二三零零零
<say-as type="number:score">12:123</say-as>	十二比一百二十三
<say-as type="number:score">12,000:123</say-as>	一万二千比一百二十三
<say-as type="number:score">12:123,000</say-as>	十二比十二万三千
<say-as type="number:score">12,000:123,000</say-as>	一万二千比十二万三千
<say-as type="number:score">12-123</say-as>	十二比一百二十三
<say-as type="number:score">12,000-123</say-as>	一万二千比一百二十三
<say-as type="number:score">12-123,000</say-as>	十二比十二万三千
<say-as type="number:score">12,000-123,000</say-as>	一万二千比十二万三千
<say-as type="number:fraction">12/123</say-as>	一百二十三分之十二
<say-as type="number:fraction">-12/123</say-as>	负一百二十三分之十二
<say-as type="telephone">110</say-as>	幺幺零
<say-as type="telephone">87654321</say-as>	八七六五四三二幺
<say-as type="telephone">13456789876</say-as>	幺三四五六七八九八七六
<say-as type="number">123</say-as>	一百二十三
<say-as type="number">-123</say-as>	负一百二十三
<say-as type="number">123.45</say-as>	一百二十三点四五
<say-as type="number">-123.45</say-as>	负一百二十三点四五
<say-as type="number">123.4500</say-as>	一百二十三点四五零零
<say-as type="number">-123.4500</say-as>	负一百二十三点四五零零
<say-as type="number">123:45</say-as>	一百二十三比四十五
<say-as type="number">123-45</say-as>	一百二十三比四十五
<say-as type="number">123.45-45</say-as>	一百二十三点四五比四十五
<say-as type="number">123-45.123</say-as>	一百二十三比四十五点一二三
<say-as type="number">123/45</say-as>	四十五分之一百二十三
<say-as type="number">-123/45</say-as>	负四十五分之一百二十三
<say-as type="number">123.45/45</say-as>	四十五分之一百二十三点四五
<say-as type="number">-123.45/45</say-as>	负四十五分之一百二十三点四五

<say-as type="number">-123.45/45</say-as>	负四十五分之一百二十三点四五
<say-as type="number">3e+11</say-as>	三乘以十的十一次幂
<say-as type="number">3e-11</say-as>	三乘以十的负十一次幂
<say-as type="number">-3e+11</say-as>	负三乘以十的十一次幂
<say-as type="number">-3e-11</say-as>	负三乘以十的负十一次幂
<say-as type="number">1.23e+11</say-as>	一点二三乘以十的十一次幂
<say-as type="number">1.23e-11</say-as>	一点二三乘以十的负十一次幂
<say-as type="number">-1.23e+11</say-as>	负一点二三乘以十的十一次幂
<say-as type="number">-1.23e-11</say-as>	负一点二三乘以十的负十一次幂
<say-as type="number">3E+07</say-as>	三乘以十的七次幂
<say-as type="number">3E-07</say-as>	三乘以十的负七次幂
<say-as type="number">-3E+07</say-as>	负三乘以十的七次幂
<say-as type="number">-3E-07</say-as>	负三乘以十的负七次幂
<say-as type="number">1.23E+07</say-as>	一点二三乘以十的七次幂
<say-as type="number">1.23E-07</say-as>	一点二三乘以十的负七次幂
<say-as type="number">-1.23E+07</say-as>	负一点二三乘以十的七次幂
<say-as type="number">-1.23E-07</say-as>	负一点二三乘以十的负七次幂

指定字母串读法 interpret-as用于指定字母串的读法类型，以纠正字母串在合成时出现的不准确读音

属性	功能	取值
interpret-as	指定大写字母串按字母读	as-character：
	指定大写字母串按词读（暂不生效）	as-word

- 使用示例：

```
 <speak>美国宇航局简称读作<say-as  interpret-as="as-word">NASA</say-as>还是<say-as  interpret-as="as-character">NASA</say-as>。</speak>
```

<silence>：插入静音 解释

<silence>用于在句间、开头或者结尾增加静音片段。

属性	功能	取值
time	设置静音时长	数字+时间单位（s秒或者ms毫秒级别）如5s

- 每个句间仅支持一个句间大静音
- silence time设置的最大值为10s，如果设置超过10s的统一按照10s处理；最小值为20ms，如果设置小于20ms的统一按20ms处理；
- 使用实例：

```
 <speak>测试一下<silence time="5s"></silence></speak>
```

<audio>：插入音频 解释

<audio>在合成中引入外部的音频

属性	功能	取值
src	插入音频 音频在合成语音之间播报	音频地址

- 音频以服务器可以访问的的地址给出，目前支持16K采样和24K采样，16bit，单声道，44字节头的wave格式文件；
- 目前使用<audio>标签，要求用户必须将相应的音频文件上传到百度云bos平台，在使用过程中提供bos地址；
- 单个请求最大限制3个并列的audio资源，单个audio资源大小限制为3M；
- 使用前需要前把音频转换为支持的格式，推荐使用ffmpeg。命令参考如下（<background>标签等同）：
  使用示例：ffmpeg -i <input-file> -acodec pcm_s16le -b:a 16k -ar 16000 -ac 1 -flags bitexact <output-file.wav>
- **<audio>标签不支持嵌套<audio>/<background>：** <audio>标签支持单标签和双标签；如果是双标签的话，当音频无法访问时，将合成嵌套文本；
- 使用示例：

```
 <speak>播放度晓晓唱的魔法城堡。<audio src="http://meitron-test.bj.bcebos.com/mofachengbao_part.wav"/> 播放完了，好听吗？</speak>
```

<background>：插入背景音 解释

<background>用于设置背景声

属性	功能	取值
src	插入音频 音频与合成声音同时播报	音频地址
repeat	音频是否重复播放	yes/no

- 音频以服务器可以访问的的地址给出，目前支持16K采样和24K采样，16bit，单声道，44字节头的wave格式文件；
- 目前使用<background>标签，要求用户必须将相应的音频文件上传到百度云bos平台，在使用过程中提供bos地址；
- **<background>标签不支持嵌套<background>**；
- 使用示例：

```
 <speak><background src="http://meitron-test.bj.bcebos.com/mofachengbao_part.wav" repeat="yes">欢迎使用百度语音合成服务</background>。</speak>
```

短文本在线合成API

接口描述

百度同步语音合成服务已上线，短文本在线合成将不再维护，请跳转同步语音合成进行接入。

百度短文本在线合成服务，基于HTTP请求的REST API接口，将文本转换为可以播放的音频文件。本文档描述了使用语音合成服务REST API的方法。

在线调试&示例代码

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。

Demo

语音合成示例代码：tts_http_demo.zip

更多语言示例代码： https://github.com/Baidu-AIP/speech-demo/tree/master/rest-api-tts

请求说明

语音合成接口支持 POST 和 GET两种方式， 推荐POST方式请求。

正式地址：http://tsn.baidu.com/text2audio 或 https://tsn.baidu.com/text2audio

- POST 方式（推荐） ， 建议文本不超过120 GBK字节，即60个汉字或者字母数字。最长1024GBK字节，文字越长耗时越长。
- GET 方式，拼接后的url总长度不多于1000个字符。
- 文本不超过120 GBK字节，即约60个汉字或者字母数字记为1次请求。每超过120个GBK字节则多记1次计费调用。

如需合成更长文本，推荐使用长文本在线合成。长文本在线合成可将10万字以内文本一次性合成，异步返回音频。支持多种优质音库，将超长文本快速转换 成稳定流畅、饱满真实的音频。适用于阅读听书、新闻播报等客户。

多音字

多音字可以通过标注自行定义发音。格式如：重(chong2)报集团。

目前只有中英文混合这一种语言，优先中文发音。示例： " I bought 3 books” 发音 “three”; “ 3 books are bought” 发音 “three”; “我们买了 3 books” 发音“三”

请求方式和参数

上传参数

参数	类型	可需	描述
tex	string	必填	合成的文本，文本长度必须小于1024GBK字节。建议每次请求文本不超过120字节，约为60个汉字或者字母数字。 请注意计费统计依据：120个GBK字节以内（含120个）记为1次计费调用；每超过120个GBK字节则多记1次计费调用。 如需合成更长文本，推荐使用长文本在线合成
tok	string	必填	开放平台获取到的开发者[access_token]获取 Access Token "access_token"）
cuid	string	必填	用户唯一标识，用来计算UV值。建议填写能区分用户的机器 MAC 地址或 IMEI 码，长度为60字符以内
ctp	string	必填	客户端类型选择，web端填写固定值1
lan	string	必填	固定值zh。语言选择,目前只有中英文混合模式，填写固定值zh
spd	int	选填	语速，取值0-15，默认为5中语速
pit	int	选填	音调，取值0-15，默认为5中语调
vol	int	选填	音量，基础音库取值0-9，精品音库取值0-15，默认为5中音量（取值为0时为音量最小值，并非为无声）
sn	string	选填	请求唯一标识，关联上下游服务，不填写服务自动生成
per（基础音库）	int	选填	度小宇=1，度小美=0，度逍遥（基础）=3，度丫丫=4
per（精品音库）	int	选填	度逍遥（精品）=5003，度小鹿=5118，度博文=106，度小童=110，度小萌=111，度米朵=103，度小娇=5 **更多发音人请查看音色列表**
			度逍遥（臻品）=4003，度博文=4106，度小贤=4115，度小鹿=4119，度灵儿

per（臻品音库）	int	选填	=4105，度小乔=4117，度小雯=4100，度米朵=4103，度姗姗=4144，度小贝=4278，度清风=4143，度小新=4140，度小彦=4129，度星河=4149，度小清=4254，度博文=4206，南方=4226 **更多发音人请查看[音色列表]**https://cloud.baidu.com/doc/SPEECH/s/Rluv3uq3d)
per(大模型音库）	int	选填	度涵竹=4189，度嫣然=4194，度泽言=4193，度怀安=4195，度清影=4196，度沁遥=4197，度小粤=20100，度晓芸=20101，四川小哥=4257，度阿闽=4132,度小蓉=4139,台媒女声=5977,度小台=4007,度湘玉=4150,度阿锦=4134,度筱林=4172 **更多发音人请查看**音色列表
aue	int	选填	3为mp3-16k/24k格式(默认)； 4为pcm-16k/24k；5为pcm-8k；6为wav（内容同pcm-16k/24k）；注意aue=4或者6是语音识别要求的格式，但是音频内容不是语音识别要求的*自然人发音*，所以识别效果会受影响。
audio_ctrl	String	选填	音频控制参数,JSON字符串格式 • sampling_rate: 采样率,仅支持降采样,示例 {"sampling_rate":16000} • AIGC: 元数据标识，仅支持mp3格式音频添加，默认不添加。 示例：{ "ContentPropagator":"内容传播服务提供者", "PropagateID":"内容传播编号" } • 注意:JSON字符串需要正确转义 • 传入AIGC字段后音频元数据将返回Label、ContentProducer、ProduceID、ContentPropagator、PropagateID总计5个标识到音频文件中。
text_ctrl	String	选填	音频情感，格式示例：text_ctrl={"emo": "fear"}意为将语音情感设置为恐惧。 其中text_ctrl 为 JSON 字符串，注意转义； emo 字段指定情感类型，可选值包括neutral（中性）、happy（快乐）、down （悲伤）、angry（愤怒）、surprise（惊讶）、fear（恐惧）；不传emo参数时（例如text_ctrl={"emo": ""}），默认使用中性语气合成。 *注：当前超拟人多情感音色能够根据输入文本自动匹配合适的情绪，一般无需手动指定。不同音色支持的情感范围存在不同， **各发音人支持的具体情感请查看**音色列表

tex字段2次urlencode

由于urlencode有两个标准 RFC 1738和RFC 3986. 百度为了更好地兼容，支持1次及2次urlencode， 其中2次urlencode可以覆盖全部的特殊字符。因而推荐传递tex 参数时做2次urlencode编码。

测试用例：“1+1=2”。 一次urlencode时，“+”可能会没有合成。

POST调用方式（推荐）

将文本以及其他参数写入到body里面，利用html表单的方式将参数传递到服务端。 所有的参数都在body中。body里面的数据为：

```
tex=***&lan=zh&cuid=***&ctp=1&aue=3&tok=***&audio_ctrl={"sampling_rate":16000}&text_ctrl={"emo":"happy"}
```

```
tex 原始参数是"百度你好“
第一次urlencode后 tex=%e7%99%be%e5%ba%a6%e4%bd%a0%e5%a5%bd
第二次urlencode后 tex=%25e7%2599%25be%25e5%25ba%25a6%25e4%25bd%25a0%25e5%25a5%25bd
```

GET调用方式

将所有的参数都填写到URL地址中，可以通过浏览器可以播放合成的语音结果。

```
 https://tsn.baidu.com/text2audio?tex=***&lan=zh&cuid=***&ctp=1&tok=***&audio_ctrl=
{"sampling_rate":16000}&text_ctrl={"emo":"happy"}

// 注意tex参数需要按照url参数标准使用url_encode UTF8编码，如合成文字为“百度你好”，url为
http://tsn.baidu.com/text2audio?
tex=%e7%99%be%e5%ba%a6%e4%bd%a0%e5%a5%bd&lan=zh&cuid=***&ctp=1&tok=***
```

返回说明

需要根据 Content-Type的头部来确定是否服务端合成成功。

如果合成成功，返回的Content-Type以“audio”开头

- aue =3 ，返回为二进制mp3文件，具体header信息 **Content-Type: audio/mp3**；
- aue =4 ，返回为二进制pcm文件，具体header信息 **Content-Type:audio/basic;codec=pcm;rate=16000/24000;channel=1**
- aue =5 ，返回为二进制pcm文件，具体header信息 **Content-Type:audio/basic;codec=pcm;rate=8000;channel=1**
- aue =6 ，返回为二进制wav文件，具体header信息 **Content-Type: audio/wav**；

如果合成出现错误，则会返回json文本，具体header信息为：**Content-Type: application/json**。其中sn字段主要用于DEBUG追查问题，如果出现问题，可以提供sn帮助确认问题。

错误示例：

```
{"err_no":500,"err_msg":"notsupport.","sn":"abcdefgh","idx":1}
```

语音识别

简介

短语音识别标准版-产品概述

百度短语音识别可以将 60 秒以下的音频识别为文字。适用于语音对话、语音控制、语音输入等场景。

- 接口类型：通过 REST API 的方式提供的通用的 HTTP 接口。适用于任意操作系统，任意编程语言
- 接口限制：需要上传完整的录音文件，录音文件时长不超过 60 秒。浏览器由于无法跨域请求百度语音服务器的域名，因此无法直接调用API接口。
- 支持音频格式：pcm、wav、amr、m4a
- 音频编码要求：采样率 16000、8000（仅支持普通话模型），16 bit 位深，单声道（音频格式查看及转换）

产品价格 短语音识别标准版支持按调用量后付费及次数包。按用量后付费按每月累计调用量阶梯计价。次数包为预付费，一年内有效，价格更优惠。详情见产品定价文档

语言及模型支持 支持中文普通话（能识别简单的常用英语）、英语、粤语、四川话识别。通过在请求时配置不同的pid参数，选择对应模型，详见 请求说明dev-pid参数表格

语音识别模型自训练 如果您在应用语音识别能力时，有行业专有名词，如金融、医疗、餐饮、地产、制造等行业术语，无法准确识别。推荐使用语音自训练平台，可以上传词汇和长文本进行模型训练，以及根据业务发展迭代不断训练。

平台使用手册

调用短语音识别标准版版API，添加训练模型ID即可生效。

短语音识别极速版-产品概述

将60秒以内的完整音频文件识别为文字，专有GPU服务集群，识别响应速度较标准版API提升2倍及识别准确率提升15%。适用于近场短语音交互，如手机语音搜索、聊天输入等场景。 **支持上传完整的录音文件，录音文件时长不超过60秒。实时返回识别结果**

产品价格 短语音识别极速版支持按调用量后付费及次数包。按用量后付费按每月累计调用量阶梯计价。次数包为预付费，一年内有效，价格更优惠。详情见产品定价文档

语言及模型支持

百度短语音识别极速版提供极速版模型1个模型（仅支持中文普通话，暂时不支持其他方言及英语）。识别速度更快，识别效果更好。支持智能标点，可以识别简单的常用英语语句。

语音识别模型自训练 如果您在应用语音识别能力时，有行业专有名词，如金融、医疗、餐饮、地产、制造等行业术语，无法准确识别。推荐使用语音自训练平台，可以上传词汇和长文本进行模型训练，以及根据业务发展迭代不断训练。

平台使用手册

调用短语音识别极速版API，添加训练模型ID即可生效。

实时语音识别websocket-产品概述

实时语音识别接口采用websocket协议的连接方式，边上传音频边获取识别结果。可以将音频流实时识别为文字，也可以上传音频文件进行识别；返回结果包含每句话的开始和结束时间，适用于长句语音输入、音视频字幕、直播质检、会议记录等场景。

两种输入格式

1. 实时音频流输入：上传必须是实时，不能过快。即整体耗时略多于原始音频流长度。如果因为导致网络不稳定需要发起新请求续传，接口允许超发XXms的录音音频；即此时不必“实时“，可以一下子将需要追溯的音频全部发给服务端。单次调用接口的音频时长目前不超过1小时。
2. 音频文件输入：支持pcm格式的音频文件，每160ms为一帧发送，间隔1-2ms，整体耗时短于音频流输入，单次调用接口的音频时长目前不超过1小时。

调用不限编程语言，只要能发送websocket请求即可，推荐在服务器上调用。

接入步骤如下：

接入步骤1-创建账号：参考文档

接入步骤2-进行接口调用：实时语音识别支持websocket协议，API方式调用。具体调用流程查看 参考文档

产品价格 实时语音识别支持按调用时长后付费及时长包。按用量后付费按固定时长单价计价。时长包为预付费，一年内有效，价格更优惠。详情见产品定价文档

语言及模型支持 支持中文普通话、音视频字幕（中文）识别模型；支持英语识别模型

音频格式 目前只支持pcm格式的原始音频数据， 16000采样率， 单声道，16bits，小端序。 目前api限制一次音频时长不超过1小时。

音频内容为：清晰的真人发音，无背景音或其它噪音，日常用语。

录音环境

百度语音识别要求安静的环境，真人的正常语速的日常用语，并且不能多个人同时发音。

以下场景讲会导致识别效果变差，错误，甚至没有结果：

1. 吵杂的环境
2. 有背景音乐，包括扬声器在播放百度合成的语音。

3. 离麦克风较远的场景应该选择远场语音识别。

以下场景的录音可能没有正确的识别结果：

1. 音频里有技术专业名称或者用语 （技术专业名称请到自训练平台改善）

2. 音频里是某个专业领域的对话，非日常用语。比如专业会议，动画片等

建议先收集一定数量的真实环境测试集，按照测试集评估及反馈。

支持语音自训练平台模型训练 实时语音识别接口支持在语音自训练平台上对中文普通话模型进行训练，可以调用训练后模型。立即训练 调用说明

优势 -与RestApi对比

实时识别api相比RestApi， 具有客户端边上传，服务端边识别的优势。 但需要使用Websocket库，相比http库而言，略微复杂，

```
 如果是小于60s的录音文件，请选择rest api
如果是实时录音，请选择实时语音识别api
```

如果录音都是10s，两种api的大致流程如下

```
 restapi http协议： 0.0s：--> 开始发起请求-->0.3s 发送完整录音文件到百度服务器 --> 3.0s: 接收返回结果
```

```
 实时语音识别api WebSocket协议：
0.0s: 【连接】发起WebSocket请求
0.1s: 【连接】连接建立
0.1s: 【发送开始参数帧】发送鉴权及识别参数
0.1s-0.2s: 【本地】录制100ms的录音
0.2s： 【发送音频数据帧】发送上面100ms的录音数据到百度服务器
0.2s-0.3s: 【本地】录制100ms的录音
0.3s： 【发送音频数据帧】发送上面100ms的录音数据到百度服务器
.....
1.2s-1.3s: 【本地】录制100ms的录音
1.3s： 【发送音频数据帧】发送上面100ms的录音数据到百度服务器
1.31s：【接收结果】收到0s-1s这段语音的一句话临时识别结果
...
4.2-4.3s: 【本地】录制100ms的录音
4.3s： 【发送音频数据帧】发送上面100ms的录音数据到百度服务器
4.39s：【接收结果】收到0s-4.1s这段语音的一句话最终识别结果
...
9.9s-10.0s: 【本地】录制100ms的录音
10.0s: 【发送音频数据帧】发送上面100ms的录音数据到百度服务器
10.0s: 【发送结束帧】 告知服务端音频结束
10.3s：【接收结果】收到7.1-10.0 这段语音的一句话最终识别结果
```

以100ms作为录音片段举例，实际建议160ms

其他 呼叫中心及客服场景，语音识别模型不同（音频采样率为8000），可使用在该场景识别率更高的商用产品**呼叫中心语音解决方案**（点击使用）。

音频文件转写-产品概述

音频文件转写接口可以将大批量的音频文件异步转写为文字。适合音视频字幕生产、批量录音质检、会议内容总结、录音内容分析等场景，一般12小时内返回识别接口。接入步骤如下：

- **接入步骤1-创建账号**：参考文档
- **接入步骤2-创建应用**：在控制台中，创建应用，勾选开通"语音技术"-"音频文件转写"能力。获取 AppID、API Key、Secret Key，并通过请求鉴权接口换取 token ，详细见接入指南。
- **接入步骤3-进行接口调用**：1.创建音频转写任务，创建需要识别的音频任务，音频需有可公开访问的url，创建成功后，音频会开始进行语音转写任务，再通过查询结果接口进行结果查询。2.查询转写任务结果，查询识别结果。

产品价格 实时语音识别支持按调用时长后付费及时长包。按用量后付费按固定时长单价计价。时长包为预付费，一年内有效，价格更优惠。详情见产品定价文档

语言及模型支持

支持中文普通话、音视频字幕（中文）识别模型；支持英语识别模型

语音字幕服务-产品概述

AI助力音视频字幕智能生产，基于海量数据和先进算法，打造音视频场景专属模型，识别准确率高达98%，并支持智能分析标点、断句，准确匹配时间轴，助力字幕生产降本增效。接入步骤如下：

- **接入步骤1-创建账号**：参考文档
- **接入步骤2-创建应用**：在控制台中，创建应用，勾选开通"语音技术"-"音频文件转写"、"实时语音识别"能力。获取 AppID、API Key、Secret Key，并通过请求鉴权接口换取 token ，详细见接入指南。
- **接入步骤3-进行接口调用**：
  实时字幕生产：参考文档
  批量字幕生产：参考文档

产品价格 实时语音识别支持按调用时长后付费及时长包。按用量后付费按固定时长单价计价。时长包为预付费，一年内有效，价格更优惠。详情见产品定价文档

语言及模型支持 支持包含简单英文的中文普通话，音视频字幕（中文）模型

鉴权认证

语音技术主要支持两种鉴权方式：access_token鉴权机制和API Key鉴权机制。二者分别支持的功能范围和适用的用户群体如下：

鉴权机制	支持的功能范围	适用群体
access_token鉴权机制	包括语音技术在内的AI开放能力的所有能力。不支持AI开放能力之外的产品	仅使用AI开放能力的客户
API Key鉴权机制	AI开放能力的公有云接口，暂不支持离线SDK 同时支持大模型服务与开发平台ModelBuilder、大模型应用开发平台AppBuilder的接口调用	要同时使用AI开放能力、ModelBuilder和AppBuilder的客户；或此前在其他平台使用过大模型服务，现在迁移到百度智能云的客户，使用API Key可无缝迁移

需要注意，两种鉴权机制选择其中一种即可，不能在一次请求中同时使用多种鉴权机制。

以下为您分别介绍两种鉴权机制使用方式。

一、access_token鉴权机制

Access_token是用户的访问令牌，承载了用户的身份、权限等信息。鉴权主要分为以下两步：

1.获取AK/SK

2.获取Access_token

- 代码形式→适用于有计算机基础的用户
- 网页调试工具→适用于零基础的用户
- 在线调试工具（推荐）→快速调试接口效果

1. 获取AK/SK 当您成功创建应用后，在对应产品页签下选择“应用列表”，可查看已创建的应用。

平台将会分配给您此应用的相关凭证，主要为**AppID、API Key、Secret Key**。以上三个信息是您应用实际开发的重要凭证，每个应用各不相同，**为了您的财产和服务安全请您妥善保管**。



另外，我们为您提供了教学视频，您可以直接浏览视频获取详细教程。

2. 获取 Access_token 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Access_token参数，Access token默认有效期为30天，获取Access_token的流程如下：

请求URL数据格式

向授权服务地址`https://aip.baidubce.com/oauth/2.0/token`发送请求（推荐使用POST），并在URL中带上以下参数：

- **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`；

例如：

```
 https://aip.baidubce.com/oauth/2.0/token?
grant_type=client_credentials&client_id=Va5yQRHlA4Fq5eR3LT0vuXV4&client_secret=0rDSjzQ20XUj5itV6WRtznPQSzr5pVw2
&
```

获取Access_token的两种方式

接下来为您提供两种获取Access_token的方法，您可以按照自己的需求进行选择。

方式一：通过代码的形式获取Access_token

以下为您提供示例代码。这里以**python语言**为例进行演示。

1. 打开python编译器，输入Access_token示例代码【python】。

Python
Java
PHP

C#

Go

Node.js

C++

```
import requests
import json


def main():

    url = "https://aip.baidubce.com/oauth/2.0/token?
grant_type=client_credentials&client_id=xxxxxx&client_secret=xxxxx"

    payload = ""
    headers = {
        'Content-Type': 'application/json',
        'Accept': 'application/json'
    }

    response = requests.request("POST", url, headers=headers, data=payload)

    print(response.text)
```

2. 在【官网获取的AK】和【官网获取的SK】中输入创建应用后获取的AK、SK。

```
import requests


def main():
    api_key = 'TZY43WL                lVRWBd'
    secret_key = 'cOof7Rq(                jiV0HHap'
    token_url = 'https://aip.baidubce.com/oauth/2.0/token'

    url = f"{token_url}?grant_type=client_credentials&client_id={api_key}&client_secret={secret_key}"

    payload = ""
    headers = {
        'Content-Type': 'application/json',
        'Accept': 'application/json'
    }

    response = requests.request("POST", url, headers=headers, data=payload)

    print(response.text)


if __name__ == '__main__':
    main()
```

3. 输入完成后运行代码，服务器将返回json文本参数，如下：

- **access_token**：要获取的Access Token；

```
{
    "access_token": "24.a9da8689794deff9a55c637df4ac04cf                .282335-39148530",
    "expires_in": 2592000,
    "refresh_token": "25.accc28f6e727040a65fa37a7bfb0e8bd                282335-39148530",
    "scope":
     放 cop_h
     smartapp
     b2b_tp_o
    "session_key": "9mzdWuVJ3a62kcrBDwFoayg0CXCaSj4IG3110WFbGHtU7D+3v+O5veG0WO/pCId+OBH/BzDO8BLAMnoZ3pdC2IlMvcPumw==",
    "session_secret": "dfaf0db2d1a0394adf91f8374a5da02c"
}
```

- **expires_in**：Access Token的有效期(秒为单位，有效期30天)；


```
{
    "access_token": "24.a9da8689794deff9a55c637df4ac04cf                    282335-39148530",
    "expires_in": 2592000,
    "refresh_token": "25.accc28f6e727040a65fa37a7bfb0e8bd                    282335-39148530",
    "scope":
     放 cop_h
     smartapp
     b2b_tp_o
    "session_key": "9mzdWuVJ3a62kcrBDwFoayg0CXCaSj4IG3110WFbGHtU7D+3v+O5veG0WO/pCId+OBH/BzDO8BLAMnoZ3pdC2IlMvcPumw==",
    "session_secret": "dfaf0db2d1a0394adf91f8374a5da02c"
}
```


- 其他参数忽略，暂时不用;

4. 若请求错误，服务器将返回的JSON文本包含以下参数：

- **error**： 错误码；关于错误码的详细信息请参考下方鉴权认证错误码。
- **error_description**： 错误描述信息，帮助理解和解决发生的错误。

例如，认证失败返回：

```
{
  "error": "invalid_client",
  "error_description": "unknown client id"
}
```

鉴权认证错误码

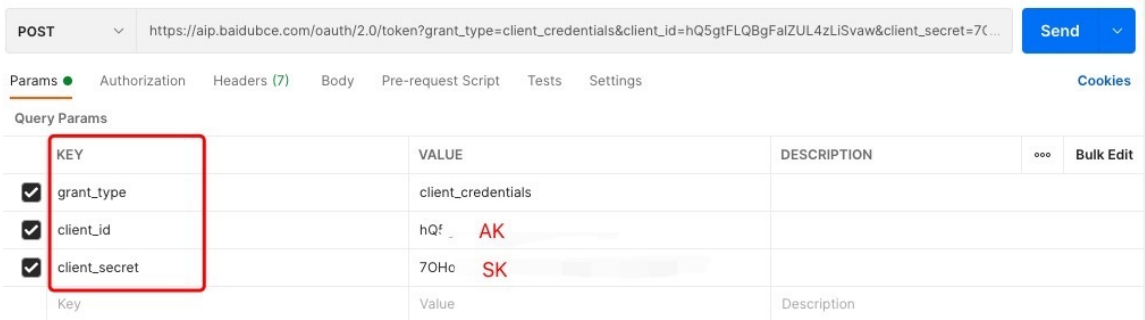
error	error_description	解释
invalid_client	unknown client id	API Key不正确
invalid_client	Client authentication failed	Secret Key不正确

方式二：使用网页调试工具（例如postman）获取Access_token

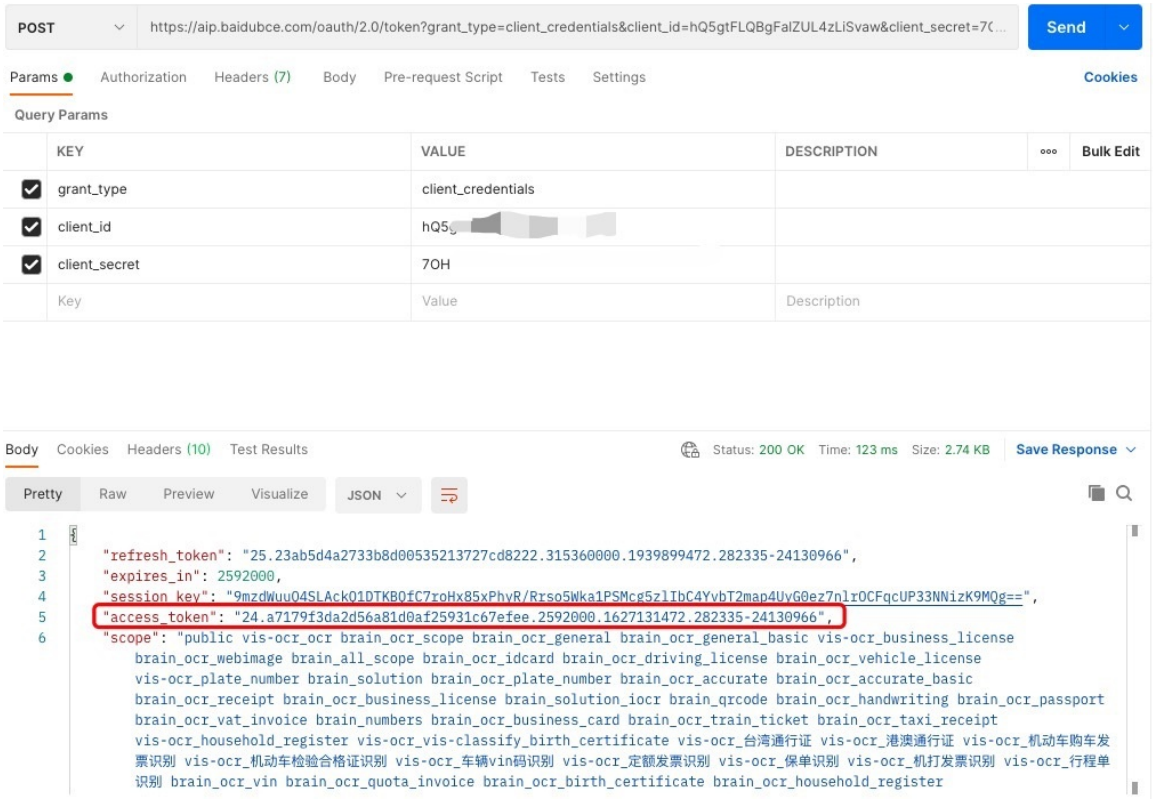
依次在网页调试工具中输入：

- **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；

具体的参数，您可以在控制台应用列表中看到，如果您还不熟悉，请您查看上一步“获取AK/SK”。



输入完成后，点击send，返回json字符串，获取Access_token。例如图中获取的access_token为24.a7179f3da2d56a81d0af25931c67efee.2592000.1627131472.282335-24130966。



另外，为您提供教学视频。您可以点击视频查看详细步骤。

方式三：在线调试工具（推荐）

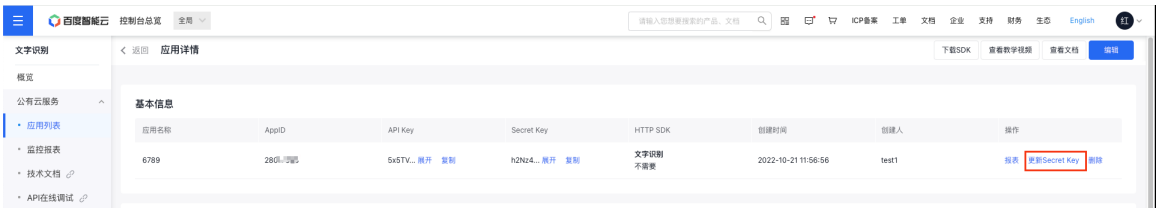
您可以在 示例代码中心 中快速调试接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

3. 密钥安全提示与止损方法

1.请勿将您的AK/SK、以及生成的Access token与他人共享或硬编码到APP及终端，为保护您的资源安全，平台可能会针对恶意滥用token进行禁用。



2.使用http协议兑换token有被截获sk的风险。如发生凭证（即AK/SK或Access token）泄露，您可以在【应用详情】页更新应用的Secret Key。 **请注意：更新后历史生成的Access_token将立即失效，请及时更新运营环境中的Secret Key，避免影响服务调用。**



二、API Key鉴权机制

API Key是百度智能云全新推出的鉴权方式，主要面向以下两个目标：

1. **跨产品调用服务**。同一个API Key可同时调用AI开放能力（文字识别、人脸识别、语音技术等）、大模型服务与开发平台ModelBuilder、大模型应用开发平台AppBuilder的接口服务，降低您接入AI服务的成本。
2. **简化调用步骤**。API Key既可直接使用明文调用接口进行鉴权，也可以换成短期APIkey后再进行鉴权。 为了更高的安全性，减少API Key的明文泄露风险，建议您使用短期APIkey的鉴权方式。

有关API Key的具体介绍和使用方法，请您参考API Key鉴权机制。

短语音识别标准版API

接口描述

将60秒以内的语音精准识别为文字，可适用于手机语音输入、智能语音交互、语音指令、语音搜索等短语音交互场景。

调用流程

1. 前置准备： 注册百度智能云账号，完成实名认证，并获取鉴权凭证。您可以在access_token或API Key两种鉴权方式中选择一种，获取方式请参考鉴权认证
2. 创建识别请求： POST 方式，音频可通过 JSON 和 RAW 两种方式提交。JSON 方式音频数据由于 base64 编码，数据会增大 1/3。其他填写具体请求参数 ，详见 ”请求说明“。
3. 短语音识别请求地址： http://vop.baidu.com/server_api
4. 返回识别结果：识别结果会即刻返回，采用 JSON 格式封装，如果识别成功，识别结果放在 JSON 的 “result” 字段中，统一采用 utf-8 方式编码。详见 ”返回说明“。

音频格式

格式支持：pcm（不压缩）、wav（不压缩，pcm编码）、amr（压缩格式）、m4a（压缩格式）。推荐pcm 采样率 ：16000、8000（仅支持普通话模型） 固定值。 编码：16bit 位深的单声道。

百度服务端会将非pcm格式，转为pcm格式，因此使用wav、amr、m4a会有额外的转换耗时。

- 16k 采样率 pcm 文件样例下载
- 16k 采样率 wav 文件样例下载
- 16k 采样率 amr 文件样例下载
- 16k 采样率 m4a 文件样例下载

m4a 格式说明

目前普通版、极速版均支持m4a格式。主要针对微信小程序的录音。

m4a 格式（AAC 编码）

1. 仅支持单声道
2. 采样率支持 16000、8000（仅支持普通话模型）
3. CBR bitrates 24000-96000，推荐 48000
4. 仅支持 AAC-LC， 不支持 例如 HE-AAC ，LD，ELD 等
5. brand 仅支持 mp42:0, mini Version 0, 不支持 M4A

微信小程序录音设置，见 微信官方文档

微信小程序录音参数，请重点关注并 **设置** 以下必填字段：

属性	类型	默认值	必填	说明
duration	number	60000	否	百度语音restapi最大支持 60s，即这个值不能超过60000
sampleRate	number	16000	是	可设为16000或8000
numberOfChannels	number	1	是	比如设为1，单声道
encodeBitRate	number	48000	否	默认值即可，建议48000，可设为24000-96000。该值越大的话，生成文件越大
format	string	aac	否	默认值即可，只支持aac，不支持mp3

在线调试&示例代码

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。

示例代码见： https://github.com/Baidu-AIP/speech-demo

包含通过bash_shell，C，Java，Python，Php，Postman进行API请求的相关示例demo代码。

请求说明

语音数据上传 POST 方式有 2 种：

1. JSON 格式 POST 上传本地音频文件。

2. RAW 格式 POST 上传本地音频文件。

两种方式均需鉴权认证通过才能使用。如您使用的是API Key鉴权方式，在请求的Header头域中的Authorization字段，需要包含API Key的鉴权信息。注意:填入鉴权信息时，需要在API Key或短期API Key前面加上`Bearer`

请求头域	参数示例
Authorization	Bearer bce-v3/ALTAK-DaIdq27UJ9Y2UEDIWx1EF/1c518f0576wee39sd59fd73983749109qq8ciq37

如您使用的是access_token鉴权方式，需要按照下文的token参数填写要求，填在对应的位置上。

两种鉴权方式二选其一即可，请不要同时使用两种鉴权方式，以免调用出错。

JSON 方式

- 音频文件，读取二进制内容后，进行 base64 编码后放在 speech 参数内。
- 音频文件的原始大小, 即二进制内容的字节数，填写 “len” 字段

由于使用 json 格式， header 为：

```
Content-Type:application/json
```

注意 由于 base64 编码后，数据会增大 1/3。

RAW方式

- 音频文件，读取二进制内容后，直接放在 body 中。
- Content-Length 的值即为音频文件的大小。（一般代码会自动生成）。

由于使用 raw 方式， 采样率和文件格式需要填写在 Content-Type 中

```
Content-Type: audio/pcm;rate=16000
```

JSON方式上传音频

语音数据和其他参数通过标准 JSON 格式串行化 POST 上传， JSON 里包括的参数：

字段名	类型	可需	描述
format	string	必填	语音文件的格式，pcm/wav/amr/m4a。不区分大小写。推荐pcm文件
rate	int	必填	采样率，16000、8000，固定值
channel	int	必填	声道数，仅支持单声道，请填写固定值 1
cuid	string	必填	用户唯一标识，用来区分用户，计算UV值。建议填写能区分用户的机器 MAC 地址或 IMEI 码，长度为60字符以内。
token	string	和API Key二选一	access_token鉴权信息，获取方式见鉴权认证
dev_pid	int	选填	不填写lan参数生效，都不填写，默认1537（普通话 输入法模型），见本节识别模型dev_pid参数
lm_id	int	选填	自训练平台模型id，填dev_pid = 1537生效
lan	string	选填，废弃参数	历史兼容参数，已不再使用
speech	string	必填	本地语音文件的二进制语音数据 ，需要进行base64 编码。与len参数连一起使用。
len	int	必填	本地语音文件的的字节数，单位字节

上传示例

JSON 格式 POST 上传本地文件

固定头部 header

```
Content-Type:application/json
```

请求示例

```
POST http://vop.baidu.com/server_api
```

speech 参数填写为 文件内容 base64 后的结果：

```
{
  "format":"pcm",
  "rate":16000,
  "dev_pid":1537,
  "channel":1,
  "token":xxx,
  "cuid":"baidu_workshop",
  "len":4096,
  "speech":"xxx", // xxx为 base64（FILE_CONTENT）
}
```

返回示例

{"corpus_no":"6433214037620997779","err_msg":"success.","err_no":0,"result":["北京科技馆，"],"sn":"371191073711497849365"}

注意事项

len 字段表示原始语音大小字节数，不是 base64 编码之后的长度。

RAW 方式上传音频

语音数据直接放在 HTTP BODY 中，控制参数以及相关统计信息通过 header 和 url 里参数传递。

Header 参数说明

字段名	数据类型	可需	描述
format	string（格式见下面示例）	必填	语音格式，pcm/wav/amr/(m4a仅支持极速版)。不区分大小写，推荐使用pcm文件
rate	int（格式见下面示例）	必填	采样率 16000、8000， 固定值

语音数据的采样率和压缩格式在 HTTP-HEADER 里的 Content-Type 表明，例：

```
Content-Type: audio/pcm;rate=16000
```

url 参数说明

字段名	可需	描述
cuid	必填	用户唯一标识，用来区分用户，计算 UV 值。建议填写能区分用户的机器 MAC 地址或 IMEI 码，长度为 60 字符以内。
token	必填	access_token鉴权信息，获取方式见鉴权认证
dev_pid	选填	不填写 lan 参数生效，都不填写，默认 1537（普通话 输入法模型），dev_pid 参数见本节开头的表格
lm_id	int	选填
lan	选填，废弃参数	历史兼容参数，已不再使用。

URL 示例：

```
POST http://vop.baidu.com/server_api?dev_pid=1537&cuid=******&token=1.a6b7dbd428f731035f771b8d********.86400.1292922000-2346678-124328

// 或者使用自训练平台

POST http://vop.baidu.com/server_api?dev_pid=1537&lm_id=1234&cuid=******&token=1.a6b7dbd428f731035f771b8d********.86400.1292922000-2346678-124328
```

raw 方式测试示例

```
curl -i -X POST -H "Content-Type: audio/pcm;rate=16000" "http://vop.baidu.com/server_api?dev_pid=1537&cuid=xxxxx&token=1.a6b7dbd428f731035f771b8d********.86400.1292922000-2346678-124328" --data-binary "@/home/test/test.pcm"
```

识别模型 dev_pid 参数

dev_pid 参数列表

- 短语音识别

请求地址: http://vop.baidu.com/server_api

dev_pid	语言	模型	是否有标点	备注
1537	普通话(纯中文识别)	语音近场识别模型	有标点	支持自定义词库
1737	英语	英语模型	无标点	不支持自定义词库
1637	粤语	粤语模型	有标点	不支持自定义词库
1837	四川话	四川话模型	有标点	不支持自定义词库

- 自训练平台

请求地址: http://vop.baidu.com/server_api

lm_id	语言	模型	是否有标点	备注
自训练平台获取	中文普通话	输入法模型	有逗号	使用自训练平台训练，不需要自定义词库

如果您在百度云购买服务器，可以通过内网域名vop.baidubce.com替换vop.baidu.com访问。该域名可免外网流量费用，且返回识别结果速度更快

- 自定义词库

自定义词库功能，可对部分专有业务名词进行识别优化。自定义词库在您网页申请的应用内设置（具体位置参见下图）。



自定义词库适合短句，保证词库中一模一样的短句可以被识别出，词库中的分词优先级较高。 自定义词库仅对普通话 dev_pid = 1537 生效，并且原始音频的采用率为 16K 或 8k。最好在 1万 行以内。

返回说明

两种上传方式都返回统一的结果，采用 JSON 格式封装，如果识别成功，识别结果放在 JSON 的 “result” 字段中，统一采用 utf-8 方式编码。

字段名	数据类型	可需	描述
err_no	int	必填	错误码
err_msg	string	必填	错误码描述
sn	string	必填	语音数据唯一标识，系统内部产生。如果反馈及 debug 请提供 sn。
result	array ([string,string,…])	选填	识别结果数组，返回1个最优候选结果。utf-8 编码。

识别成功返回 case

```
{"err_no":0,"err_msg":"success.","corpus_no":"15984125203285346378","sn":"481D633F-73BA-726F-49EF-8659ACCC2F3D","result":["北京天气"]}
```

识别错误返回 case

```
{"err_no":2000,"err_msg":"data empty.","sn":"481D633F-73BA-726F-49EF-8659ACCC2F3D"}
```

短语音识别极速版API

接口描述

将60秒以内的完整音频文件识别为文字，专有GPU服务集群，识别响应速度较标准版API提升2倍及识别准确率提升15%。适用于近场短语音交互，如手机语音搜索、聊天输入等场景。 支持上传完整的录音文件，录音文件时长不超过60秒，实时返回识别结果。

调用流程

1. 创建账号及应用： 在控制台中，创建应用，勾选开通 “语音技术”—“短语音识别”、“短语音识别极速版” 能力。获取AppID、API Key、Secret Key，并通过请求鉴权接口换取 token ，详细见 “接入指南”。
2. 创建识别请求： POST 方式，音频可通过 JSON 和 RAW 两种方式提交。JSON 方式音频数据由于 base64 编码，数据会增大 1/3。其他填写具体请求参数 ，详见 ”请求说明“。
3. 短语音识别请求地址： https://vop.baidu.com/pro_api
4. 返回识别结果：识别结果会即刻返回，采用 JSON 格式封装，如果识别成功，识别结果放在 JSON 的 “result” 字段中，统一采用 utf-8 方式编码。详见 ”返回说明“。

音频格式

格式支持：pcm（不压缩）、wav（不压缩，pcm编码）、amr（压缩格式）、m4a（压缩格式），推荐pcm。

采样率 ：16000（仅支持普通话模型） 固定值。

编码：16bit 位深的单声道。

百度服务端会将非pcm格式，转为pcm格式，因此使用wav、amr、m4a会有额外的转换耗时。

- 16k 采样率 pcm 文件样例下载
- 16k 采样率 wav 文件样例下载
- 16k 采样率 amr 文件样例下载
- 16k 采样率 m4a 文件样例下载

m4a 格式说明

目前普通版、极速版均支持m4a格式。主要针对微信小程序的录音。

m4a 格式（AAC 编码）

1. 仅支持单声道
2. 采样率支持 16000、8000（仅支持普通话模型）
3. CBR bitrates 24000-96000，推荐 48000

4. 仅支持 AAC-LC， 不支持 例如 HE-AAC ，LD，ELD 等

5. brand 仅支持 mp42:0, mini Version 0, 不支持 M4A

微信小程序录音设置，见 微信官方文档

微信小程序录音参数，请重点关注并 **设置** 以下必填字段：

属性	类型	默认值	必填	说明
duration	number	60000	否	百度语音restapi最大支持 60s，即这个值不能超过60000
sampleRate	number	16000	是	可设为16000或8000
numberOfChannels	number	1	是	比如设为1，单声道
encodeBitRate	number	48000	否	默认值即可，建议48000，可设为24000-96000。该值越大的话，生成文件越大
format	string	aac	否	默认值即可，只支持aac，不支持mp3

在线调试&示例代码

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。

示例代码见： https://github.com/Baidu-AIP/speech-demo

包含通过bash_shell，C，Java，Python，Php，Postman进行API请求的相关示例demo代码。

请求说明

语音数据上传 POST 方式有 2 种：

1. JSON 格式 POST 上传本地音频文件。

2. RAW 格式 POST 上传本地音频文件。

JSON 方式

- 音频文件，读取二进制内容后，进行 base64 编码后放在 speech 参数内。
- 音频文件的原始大小, 即二进制内容的字节数，填写 “len” 字段

由于使用 json 格式， header 为：

```
Content-Type:application/json
```

注意 由于 base64 编码后，数据会增大 1/3。

RAW方式

- 音频文件，读取二进制内容后，直接放在 body 中。
- Content-Length 的值即为音频文件的大小。（一般代码会自动生成）。

由于使用 raw 方式， 采样率和文件格式需要填写在 Content-Type 中

```
Content-Type: audio/pcm;rate=16000
```

JSON方式上传音频

语音数据和其他参数通过标准 JSON 格式串行化 POST 上传， JSON 里包括的参数：

字段名	类型	可需	描述
format	string	必填	语音文件的格式，pcm/wav/amr/m4a。不区分大小写。推荐pcm文件
rate	int	必填	采样率，16000，固定值
channel	int	必填	声道数，仅支持单声道，请填写固定值 1
cuid	string	必填	用户唯一标识，用来区分用户，计算UV值。建议填写能区分用户的机器 MAC 地址或 IMEI 码，长度为60字符以内。
token	string	必填	开放平台获取到的开发者access_token获取 Access Token "access_token"）
dev_pid	int	必填	80001（极速版输入法模型）
lm_id	int	选填	自训练平台模型id
lan	string	选填，废弃参数	历史兼容参数，已不再使用
speech	string	必填	本地语音文件的的二进制语音数据 ，需要进行base64 编码。与len参数连一起使用。
len	int	必填	本地语音文件的的字节数，单位字节

上传示例

JSON 格式 POST 上传本地文件

固定头部 header

```
Content-Type:application/json
```

请求示例

4K大小的pcm文件（普通话录音）请求：

```
POST https://vop.baidu.com/pro_api
```

speech 参数填写为 文件内容base64后的结果：

```
{
  "format":"pcm",
  "rate":16000,
  "dev_pid":80001,
  "channel":1,
  "token":xxx,
  "cuid":"baidu_workshop",
  "len":4096,
  "speech":"xxx", // xxx为 base64（FILE_CONTENT）
}
```

返回示例

```
{"corpus_no":"6433214037620997779","err_msg":"success.","err_no":0,"result":["北京科技
馆，"],"sn":"371191073711497849365"}
```

注意事项

len 字段表示原始语音大小字节数，不是 base64 编码之后的长度。

RAW 方式上传音频

语音数据直接放在 HTTP BODY 中，控制参数以及相关统计信息通过 header 和 url 里参数传递。

Header 参数说明

字段名	数据类型	可需	描述
format	string（格式见下面示例）	必填	语音格式，，pcm、wav、amr、m4a。不区分大小写，推荐使用pcm文件
rate	int（格式见下面示例）	必填	采样率 16000， 固定值

语音数据的采样率和压缩格式在 HTTP-HEADER 里的 Content-Type 表明，例：

```
Content-Type: audio/pcm;rate=16000
```

url 参数说明

字段名	可需	描述
cuid	必填	用户唯一标识，用来区分用户，计算UV值。建议填写能区分用户的机器 MAC 地址或 IMEI 码，长度为60字符以内。
token	必填	开放平台获取到的开发者access_token获取 Access Token "access_token")
dev_pid	必填	80001
lan	选填，废弃参数	历史兼容参数，请使用dev_pid。如果dev_pid填写，该参数会被覆盖。语种选择,输入法模型，默认中文（zh）。 中文=zh、粤语=ct、英文=en，不区分大小写。

URL 示例：

```
POST https://vop.baidu.com/pro_api?dev_pid=80001&cuid=******&token=1.a6b7dbd428f731035f771b8d********.86400.1292922000-2346678-124328

// 或者使用自训练平台

POST http://vop.baidu.com/pro_api?dev_pid=80001&lm_id=1234&cuid=******&token=1.a6b7dbd428f731035f771b8d********.86400.1292922000-2346678-124328
```

raw 方式测试示例

```
curl -i -X POST -H "Content-Type: audio/pcm;rate=16000" "http://vop.baidu.com/pro_api?dev_pid=80001&cuid=xxxxx&token=1.a6b7dbd428f731035f771b8d********.86400.1292922000-2346678-124328" --data-binary "@/home/test/test.pcm"
```

- 自训练平台

请求地址: http://vop.baidu.com/pro_api

lm_id	语言	模型	是否有标点	备注
自训练平台获取	中文普通话	输入法模型	有逗号	使用自训练平台训练，不需要自定义词库

具体功能及使用说明请参考文档 自训练平台手册

如果您在百度云购买服务器，可以通过内网域名vop.baidubce.com替换vop.baidu.com访问。该域名可免外网流量费用，且返回识别结果速度更快

返回说明

两种上传方式都返回统一的结果，采用 JSON 格式封装，如果识别成功，识别结果放在 JSON 的 “result” 字段中，统一采用 utf-8 方式编码。

字段名	数据类型	可需	描述
err_no	int	必填	错误码
err_msg	string	必填	错误码描述
sn	string	必填	语音数据唯一标识，系统内部产生。如果反馈及 debug 请提供 sn。
result	array ([string,string,…])	选填	识别结果数组，返回1个最优候选结果。utf-8 编码。

识别成功返回 case

```
{"err_no":0,"err_msg":"success.","corpus_no":"15984125203285346378","sn":"481D633F-73BA-726F-49EF-8659ACCC2F3D","result":["北京天气"]}
```

识别错误返回 case

```
{"err_no":2000,"err_msg":"data empty.","sn":"481D633F-73BA-726F-49EF-8659ACCC2F3D"}
```

实时语音识别-websocket API

接口描述

实时语音识别接口采用websocket协议的连接方式，边上传音频边获取识别结果。可以将音频流实时识别为文字，也可以上传音频文件进行识别；返回结果包含每句话的开始和结束时间，适用于长句语音输入、音视频字幕、直播质检、会议记录等场景。

WebSocket简介

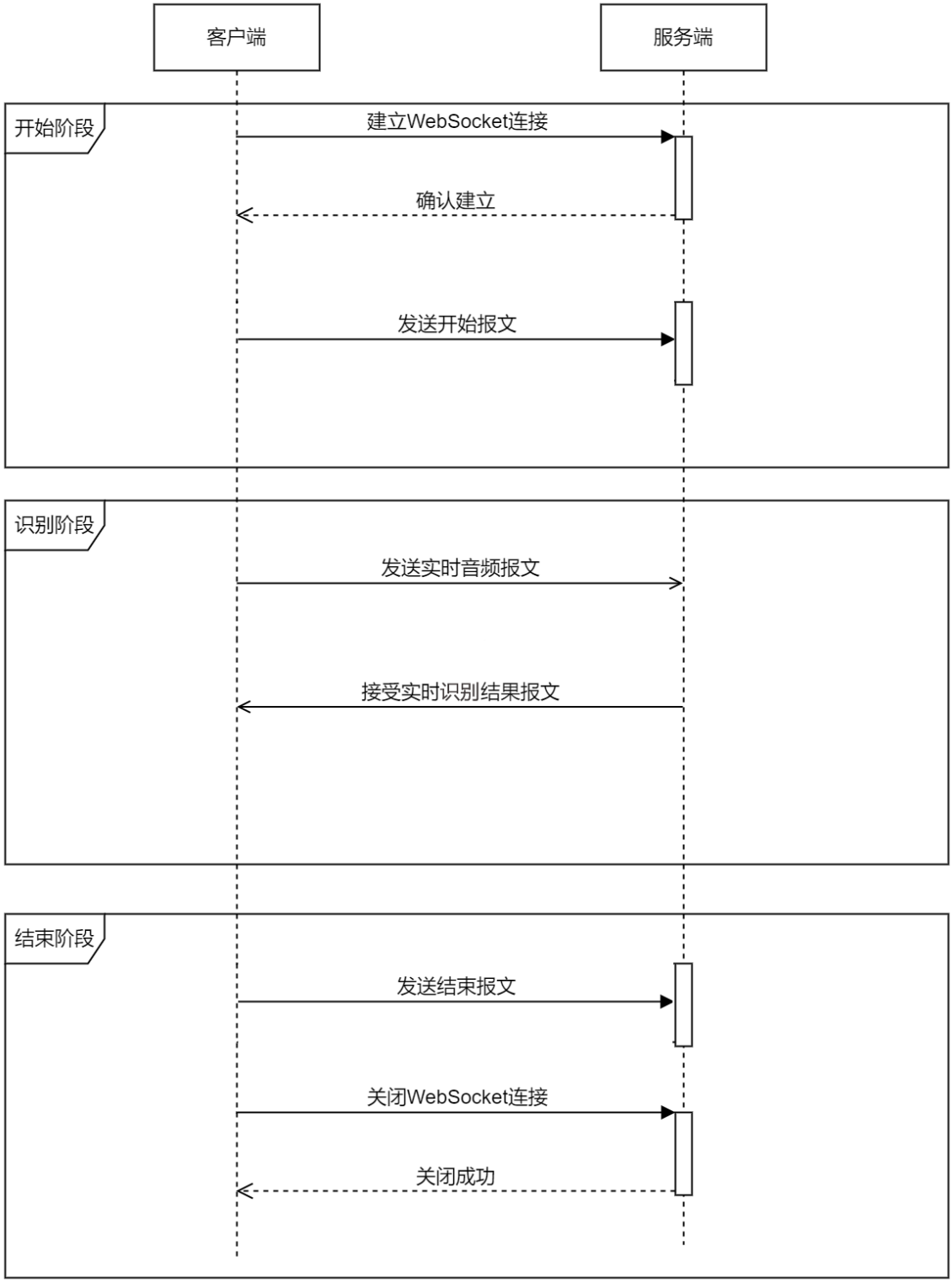
WebSocket 是基于TCP的全双工协议，即建立连接后通讯双方都可以不断发送数据。

WebSocket 协议由rfc6455定义， 下面介绍常见WebSocket 库的调用流程及参数

一般需要各编程语言的WebSocket库来实现接入。WebSocket库需支持rfc6455描述的协议， 即支持Sec-WebSocket-Version: 13

主要流程

1. 连接
2. 连接成功后发送数据
   2.1 发送开始参数帧
   2.2 实时发送音频数据帧
   2.3 库接收识别结果
   2.4 发送结束帧
3. 关闭连接



名词解释：

- 连接： 这里指TCP连接及握手（Opening Handshake） ， 一般WebSocket库已经封装，用户不必关心
- 发送数据帧：Sending Data Frame， 类似包的概念，指一次发送的内容 。从客户端到服务端。
  - 文本帧：Opcode 0x1 (Text)， 实时语音识别api发送的第一个开始参数帧和最后一个结束帧，文本的格式是json
  - 二进制帧：Opcode0x2 (Binary), 实时语音识别api 发送的中间的音频数据帧
- 接收数据帧： Receiving Data Frame， 类似包的概念，指一次发送的内容 。从服务端到客户端。
  - 文本帧：Opcode 0x1 (Text)， 识别结果或者报错，文本的格式是json
  - 二进制帧：实时语音识别api 不会收到

- 关闭连接：Closing Handshake。 百度服务端识别结束后会自行关闭连接，部分WebSocket库需要得知这个事件后手动关闭客户端连接。

通常WebSocket库用需要用户自己定义下面的3个回调函数实现自己的业务逻辑。

```
 连接成功后的回调函数：
{
 #通常需要开启一个新线程，以避免阻塞无法接收数据
 2.1 发送开始参数帧
 2.2 实时发送音频数据帧
 2.4 发送结束帧
}

接收数据的回调函数
{
 2.3 库接收识别结果，自行解析json获得识别结果
}

服务端关闭连接的回调函数
{
 3. 关闭客户端连接， 部分库可以自动关闭客户端连接。
}
```

在线调试&示例代码

Demo 目前提供如下demo及演示功能，点击进入下载页面

编程语言	操作系统及版本	使用文件流演示接口基本识别功能	耗时计算	模拟实时音频流	断网补发数据
Java	任意，支持Java 8或以上	√	√	√	
Android	Android Api Level 15 或以上	√	√	√	
Python	任意，支持Python3	√			
C++	Linux ，支持C++ 11 以上	√			

请求说明

连接

连接地址（WebSocket URI）：`wss://vop.baidu.com/realtime_asr?sn=XXXX-XXXX-XXXX-XXX` 参数 sn由用户自定义用于排查日志，建议使用随机字符串如UUID生成。 sn的格式为英文数字及“-”，*长度128个字符内，即[a-zA-Z0-9-]{1, 128}*

如果连接成功，一般WebSocket库会发起回调。

发送开始参数帧

注意帧的类型（Opcode）是**Text**， 使用json序列化

示例：

```
{
    "type": "START",
    "data": {
        "appid": 105xxx17,
        "appkey": "UA4oPSxxxxkGOuFbb6",
        "dev_pid": 15372, # 识别模型，比如普通话还是英语，是否要加标点等
        "lm_id": xxxx, # 自训练平台才有这个参数
        "cuid": "cuid-1", # 随便填不影响使用。机器的mac或者其它唯一id，页面上计算UV用。
        “user”: "xxx", # 使用中文多方言模型（pid:15376）时此参数必填，参数值任意。
        # 下面是固定参数
        "format": "pcm",
        "sample": 16000

    }
}
```

具体参数说明

参数名称	类型	是否必填	说明
type	String	必填，固定值	START，开始帧的类型
data	Array	必填	具体见下表

data参数说明

参数名称	类型	是否必填	说明
appid	int	必填	控制台网页上应用的鉴权信息 AppID
appkey	string	必填	控制台网页上应用的鉴权信息 API Key
dev_pid	int	必填	识别模型，推荐15372，见下一个表格
lm_id	int	可选	填入自训练平台训练上线的模型id，需要和训练的基础模型dev-pid对齐。参考平台模型调用部分的提示。
cuid	string	必填	统计UV使用，发起请求设备的唯一id，如服务器的mac地址。随意填写不影响识别结果。长度128个字符内，即[a-zA-Z0-9-_]{1, 128}
user	string	可选	使用中文多方言模型（pid:15376）时此参数必填，参数值任意。
format	string	必填，固定值	pcm，固定格式
sample	int	必填，固定值	16000， 固定采样率

开放平台模型（无在线语义功能）

PID	模型	是否有标点及后处理	推荐场景
1537	中文普通话	弱标点（逗号，句号）	手机近场输入
15372	中文普通话	加强标点（逗号、句号、问号、感叹号）	手机近场输入
15376	中文多方言	弱标点（逗号，句号）	手机近场输入
1737	英语	无标点	手机近场输入
17372	英语	加强标点（逗号、句号、问号）	手机近场输入

> 使用中文多方言模型（pid:15376）时，需要在参数中加入"user":"XXX"（参数任意）

> 目前中文多方言模型（pid:15376）可同时支持中文、粤语、四川话和东北话

语音自训练平台模型训练

实时语音识别接口支持在语音自训练平台上训练中文普通话模型



训练后的模型注意必须填写上线模型的模型参数，可在自训练平台的模型调用模块进行查看。

示例 获取专属模型参数pid:1537或15372 modelid:1235， 则调用websocket API时必须填写参数 dev_pid=1537或15372（PID功能见下表）；同时lm_id 设置为1235。

PID	模型	是否有标点	备注
1537	中文普通话	无标点	
15372	中文普通话	加强标点（逗号、句号、问号、感叹号）	

发送音频数据帧

这里需要注意，服务端5s没有收到音频数据会断开并报错，发送间隔控制在5s以内

注意帧的类型（Opcode）是**Binary**

内容是二进制的音频内容。 除最后一个音频数据帧， 每个帧的音频数据长度为20-200ms。 建议最佳**160ms**一个帧，有限制的也建议80ms。

160ms = 160 *(16000* 2 /1000) = 5120 bytes

> 计算方式：
> 16000采样率： 1s音频 16000采样点
> 16bits： 一个采样点 16bits = 2 bytes
> 1s ： = 1000ms
> 即 160ms * 16000 * 2bytes / 1000ms = 5120bytes

实时语音识别api 建议实时发送音频，即每个160ms的帧之后，下一个音频数据帧需要间隔160ms。即：文件，此处需要sleep(160ms) 如果传输过程中网络异常， 需要补断网时的识别结果，发送的音频数据帧之间可以没有间隔。具体见下文“断网补发数据”一节

发送结束帧

注意帧的类型（Opcode）是**Text**， 使用json序列化

示例：

```
 {
 "type":"FINISH"
}
```

具体参数说明

参数名称	类型	是否必填	说明
type	String	必填，固定值	，结束帧的类型

发送取消帧

取消与结束不同，结束表示音频正常结束，取消表示不再需要识别结果，服务端会迅速关闭连接 示例：

```
 {
 "type":"CANCEL"
}
```

具体参数说明

参数名称	类型	是否必填	说明
type	String	必填，固定值	CANCEL 立即取消本次识别

发送心跳帧

注意帧的类型（Opcode）是**Text**， 使用json序列化 正常情况下不需要发这个帧，仅在网络异常的时候，需要补传使用，具体见“断网补发数据”

示例：

```
 {
 "type":"HEARTBEAT"
}
```

具体参数说明

参数名称	类型	是否必填	说明
type	String	必填，固定值	HEARTBEAT, 心跳帧的类型

返回说明

接收数据帧

注意需要接收的帧类型（Opcode）是**Text**， 本接口不会返回Binary类型的帧。 text的内容， 使用json序列化

临时及最终识别结果

一段音频由多句话组成，实时识别api会依次返回每句话的临时识别结果和最终识别结果

一句话的临时识别结果示例：

```
 {
 "err_no":0,
 "err_msg":"OK",
 "type": "MID_TEXT",
 "result": "北京天气怎"，
 "log_id": "45677785"，
 "sn": "399427ce-e999-11e9-94c8-fa163e4e6064_ws_2"

}
```

一句话的最终识别结果：

```
 {
   "err_no":0,
 "err_msg":"OK",
   "log_id":"45677785",
   "sn":"399427ce-e999-11e9-94c8-fa163e4e6064_ws_2"
 "type":"FIN_TEXT",
 "result":"北京天气怎么样",
 "start_time":53220,
 "end_time":73340,
 "product_id":1537,
 "product_line":"open"
}
```

心跳帧（收到后，请忽略）

示例：

```
 {
 "type":"HEARTBEAT"
}
```

与发送的心跳帧不同，这个是接收服务端下发的，5s一次，收到后可以忽略。

服务端报错 如-3005错误码，是针对的是一个句子的，其它句子依旧可以识别，请求是否结束以服务端是否关闭连接为准。 具体错误码含义见文末“错误码“一节

```
 {
 "err_msg": "asr authentication failed",
 "err_no": -3004,
 "log_id": "106549729",
 "result": "",
 "sn": "0d25b5e1-ffa6-11e9-8968-f496349f3917",
 "type": "FIN_TEXT"
}
```

一句话的开始时间及结束时间：识别过程中，百度服务端在每句话的最终识别结果中带有这句话的开始和结束时间。最终识别结果是指"type":"FIN_TEXT"，即一句话的最后识别结果，包括这句话的报错结果。

通常一个音频会得到如下的时间信息：

```
  # s1 e1 指第一句话的开始和结束时间start_time，end_time
[静音][s1:e1][静音][s2:e2]....[静音][s10:e10][静音]
一句话不能超过60s，否则会被强制切割
```

参数说明 | 参数名称 | 名称 | 类型 | 说明 | | ---------- | ---------- | --------- | ---------- | | err_no | 错

误码 | int | 0 表示正确， 其它错误码见文末 | | err_msg | 错误信息 | string | err_no!=0时，具体的报错解释。 | | type | 结果类型 | string | 见下面3行示临时识别结果 | | ~ | ~ | MID_TEXT | 一句话以及临时识别结果 | | ~ | ~ | FIN_TEXT | 一句话的最终识别结果或者报错， 是否报错由err_no判断 | | ~ | ~ | HEARTBEAT | 仅断网补发音频数据需要，见下文“断网补发数据”一节 | | result | 识别结果 | string | 音频的识别结果 | | start_time | 一句话的开始时间 | int，毫秒 | 一句话的开始时间，临时识别结果MID_TEXT 无此字段 | | end_time | 一句话的结束时间 | int，毫秒 | 一句话的结束时间，临时识别结果MID_TEXT 无此字段 | | logid | 日志id | string | 日志id，用这个id可以百度服务端定位请求，排查问题 | | sn | 请求sn | string | 用这个sn可以百度服务端定位请求，排查问题。ws URI里的参数及识别句子的组合 |

断网补发数据

请先看“发送音频数据帧” 和 “接收数据帧” 这2节。

断网补发数据的目的是将一个语音流，在网络不佳的情况下，通过自己的代码逻辑拼接，使得多次请求的结果看上去像一次。

简单来说就是哪里断开，从哪里开始重新发一次请求，“哪里”=最后一次接收的“end_time”。服务端对每个请求独立，需要自行拼接补发数据的请求时间。

如果一个音频的10个句子时间如下：

```
  # s1 e1 指第一句话的开始和结束时间start_time，end_time
[静音][s1:e1][静音][s2:e2]...[s7:e7]...[静音][s10:e10][静音]
```

如果发送过程中，比如在第七句e7之后，网络抖动或者遇见其它错误。但是为了不影响最终的用户体验，期望连续的10个句子的识别结果。

此时，可以发起一个新的请求，从e7开始发数据，在，语音数据帧之间不需要sleep。

> 如果超过5s没有发送音频数据给服务端，服务端会下发报错并结束连接，建议至少2s 发送一次。

完整流程如下

1. 开始一次请求，正常发送音频数据，并缓存音频数据，
2. 接收到数据帧， 保存end_time
3. 如果此时断网，读取最后一次的end_time如7000ms
4. 开始一次新请求， 从缓存的音频数据中找到7000ms（224000bytes）以后的数据，发送给服务端。每个帧160ms的音频数据，补数据时，每个帧之间不需要间隔sleep。
5. 一直追上实时音频数据，开始实时发送
6. 新请求的start_time和end_time可以加上7000ms，然后展示给用户
7. 如果补发数据过大，新请求过快结束，在新请求结束时，需要补type=HEARTBEAT心跳帧，建议2-3s发一次，避免5s服务端读超时。
8. 如果再次断网流程依旧

音频文件转写API

接口描述

音频文件转写接口可以将大批量的音频文件异步转写为文字。适合音视频字幕生产、批量录音质检、会议内容总结、录音内容分析等场景，一般12小时内返回识别结果。
当前接口新增支持 pid=8953，具备话者分离能力，可在单声道或多人对话音频中自动区分不同说话人，并在转写结果中返回对应的话者标识，便于后续进行会议纪要整理、客服质检、访谈分析等精细化处理。

步骤：

1、根据音频url、音频格式、语言id以及采样率等参数创建音频转写任务，获取task_id参数。

2、根据task_id的数组批量查询音频转写任务结果。

创建音频转写任务-请求说明

请求接口：https://aip.baidubce.com/rpc/2.0/aasr/v1/create

HTTP 方法：POST

URL参数：

参数	取值
access_token	通过 API Key 和 Secret Key 获取的 access_token，参考Access Token获取

JSON方式上传音频

Body中放置请求参数，语音数据和其他参数通过标准 JSON 格式串行化 POST 上传，包括的参数如下：

参数名	类型	是否必需	对外状态	取值范围
speech_url	str	是	音频url	可使用百度云对象存储进行音频存储，生成云端可外网访问的url链接，音频大小不超过500MB，url长度不超过2048字节
format	str	是	音频格式	["mp3", "wav", "pcm","m4a","amr"]单声道，编码 16bits 位深
pid	int	是	语言类型	8953（话者分离模型），80006（中文音视频字幕模型），80001（中文语音近场识别模型极速版），1737（英文模型）
rate	int	是	采样率	[16000] 固定值
role_num	int	否	所属发音人	固定值5，pid为8953时由算法自动判断说话人，本参数非必需。其他PID暂不支持话者分离能力，可忽略。
smooth_text	int	否	文本顺滑	0（不开启文本顺滑），1（开启文本顺滑） 注意：当此参数为1时，pid必须等于80001或80006；
filter_sensitive	int	否	敏感词过滤	0（不开启敏感词过滤），1（开启敏感词过滤） 注意：当此参数为1时，pid必须等于80001或80006；

若音频采样率为8k，请使用呼叫中心音频文件转写

在线调试&示例代码

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。

Python Demo 点击下载 （文件为.zip压缩包，若无法打开时，可尝试在文件后上增加".zip“后缀）

Body请求示例：

```
{
  "speech_url": "https://platform.bj.bcebos.com/sdk%2Fasr%2Fasr_doc%2Fdoc_download_files%2F16k.pcm",
  "format": "pcm",
  "pid": 8953,
  "rate": 16000
}
```

创建音频转写任务-返回说明

返回参数

参数名	类型	是否必需	对外状态
log_id	int	是	log id
task_id	str	否	任务id
task_status	str	否	任务状态
error_code	int	否	错误码
error_msg	str	否	错误信息

Body返回示例：

```
  # 创建成功
{
    "log_id": 12345678,
    "task_status": "Created" ,
    "task_id":  "234acb234acb234acb234acb"  #注意保存该id，用于后续请求识别结果
}
 # 创建失败，缺少参数
{
    "error_code": 336203,
    "error_msg": "missing param: speech_url",
    "log_id": 5414433131138366128
}
```

注意：查询识别结果时，需要该步骤返回的task_id来进行请求。请注意保存task_id列表。

查询音频转写任务-请求说明

请求接口：https://aip.baidubce.com/rpc/2.0/aasr/v1/query

HTTP 方法：POST

URL参数：

参数	取值
access_token	通过API Key和Secret Key获取的access_token，参考Access Token获取

Body中放置请求参数，参数如下：

参数名	类型	是否必需	描述	取值范围
task_ids	list	是	任务id	task_ids为空，返回空任务结果列表；单次查询任务数不超过200个

在线调试&示例代码

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。

请求示例：

```
 {
   "task_ids": ["234acb234acb234acb234acb", "234acb234acb234acb234acd", "234acb234acb234acb234acbe"]
}
```

查询音频转写任务-返回说明

返回参数：

参数名	类型	是否必需	描述
log_id	int	是	log id
tasks_info	list	否	多个任务的结果
+task_id	str	是	任务id
+task_status	str	是	任务状态
+task_result	dict	否	转写结果的json格式
++corpus_no	str	否	日志，反馈时使用
++result	str	否	转写结果
++audio_duration	int	否	音频时长（毫秒）
++detailed_result	list	否	分段转写详细结果
+++voice_power	int	否	音量参数 |
+++res	list	否	分段转写文本
+++speaker_id	int	否	所属发言人
+++speak_speed	int	否	语速参数
+++begin_time	int	否	分段开始时间
+++end_time	int	否	分段结束时间
+++words_info	list	否	字粒度（预留参数，暂不启用）
+++sn	str	否	分段日志，反馈时使用
+++corpus_no	str	否	分段日志，反馈时使用
++err_no	int	否	转写失败错误码
++err_msg	str	否	转写失败错误信息
error_code	int	否	请求错误码
error_msg	str	否	请求错误信息
error_info	list	否	错误的或查询不存在的taskid数组

返回示例：

```
{
  "log_id": 12345678,
  "tasks_info": [
   { # 转写中
    "task_status": "Running"
    "task_id": "234acb234acb234acb234acb",
   },
   { # 转写失败
    "task_status": "Failure"
    "task_id": "234acb234acb234acb234acd",
    "task_result": {
     "err_no":  3301
     "err_msg": "speech quality error",
     "sn": "xxx"
    }
   },
   { # 转写成功
    "task_status": "Success",
    "task_result": {
     "result": [
        "观众朋友大家好，欢迎收看本期视频哦。毕竟..."
     ],
           "audio_duration": 6800,
     "detailed_result": [
        {
           "voice_power": "55.13867950439453",# 仅使用PID：80006时返回
         "res": [
              "观众朋友大家好，欢迎收看本期视频哦。 "
           ],
           "speak_speed": "5.986260890960693",
           "speaker_id": 0, # 仅使用PID：8953时返回
           "end_time": 6700,
           "begin_time": 4240,
           "words_info": [],
           "sn": "257826606251573543780",
           "corpus_no": "6758319075297447880"
        }
        ...
     ],
     "corpus_no": "6758319075297447880"
  },
  "task_id": "234acb234acb234acb234ace"
   }
]
}
```

音频文件转写极速版API

接口描述

支持提取视频、音频文件中的语音内容，极速返回识别结果及时间戳，适用于对实时性有一要求的内容分析、字幕生产、录音转写场景。**(本接口处于邀测阶段，请提交合作咨询申请测试)**

请求说明

请求接口：https://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/asr/topspeed

HTTP 方法：POST

Headers参数：

参数	参数值	是否必须
Content-Type	application/json	是

Body参数如下：

参数名称	类型	是否必须	对外状态	备注
access_token	str	必须		通过 API Key 和 Secret Key 获取的 access_token，参考Access Token获取
rate	int	必须		音频对应的采样率，目前只支持传16000
cuid	str	必须		用户的唯一标识，调用时自己保证唯一即可
dev_pid	int	必须		调用的转写模型，不同模型有不同的转写效果，目前只支持传80006
speech	str	与speech_url二选一		对视频或音频文件进行base64编码后的结果，音频文件最多支持一小时的时长，整体文件不能超过500M。支持pcm、wav、mp3、m4a、mp4、mov等常见音视频文件格式
speech_url	str	与speech二选一		可使用百度云对象存储进行音频存储，生成云端可外网访问的url链接。音频文件最多支持一小时的时长，整体文件不能超过500M。支持pcm、wav、mp3、m4a、mp4、mov等常见音视频文件格式
enable_subtitle	int	非必须	根据常见字幕规则返回识别结果及对应时间戳	取值范围：[0,1,2] 0：关闭字幕功能（默认） 1：开启字幕功能 2：开启字幕模式，返回字粒度时间戳 其他取值：报错
subtitle_punc	int	非必须	字幕结果中是否有标点	取值范围：[0,1] 0：过滤字幕中的标点（默认） 1：不过滤字幕中的标点 其他取值：报错 仅当字幕模式开启时生效
smooth_text	int	非必须	文本顺滑（标点优化、数字格式优化、口语过滤）	取值范围：[0（不开启文本顺滑，默认），1（开启文本顺滑）]
smooth_text_param	list	非必须	具体开启的文本顺滑功能，仅当smooth_text=1时生效	取值范围：[1（标点），2（数字），3（口语）] 列表中参数必须为int类型，自定义所需功能，例如： 传入[1, 2, 3]，开启标点+数字+口语功能 传入[1, 2]，开启标点+数字 传入[3]，开启口语
filter_sensitive	int	非必须	敏感词过滤	取值范围：[0（不开启敏感词过滤，默认），1（开启敏感词过滤）]

Body请求示例：

```
{
  "access_token":"24.578xxxxxxxxxxxxxxx",
  "rate": 16000,
  "cuid": "baidu_asr",
  "dev_pid": 80006,
  "speech": "UklGRqAfQgBXQVZFZm10IBAAAAABAAEAgD4AAAB9AAACABAATEl…",
  "enable_subtitle":1,
  "smooth_text": 1
}
```

返回说明

返回参数：

参数名称	类型	是否必须	备注	其他信息
result	str []	非必须	完整的转写结果，取元素0即是对应的文字	item 类型: string
error_code	int	非必须	错误码	
error_message	str	非必须	错误信息	
audio_duration	int	非必须	音频整体时长，单位：毫秒	
detailed_result	object []	非必须	每一句转写详情，含时间戳等信息	item 类型: object
+ res	str []	非必须	每句转写的文字结果，取元素0即是对应的文字	item 类型: string
+ end_time	int	非必须	文字结束时间，单位：毫秒	
+ begin_time	int	非必须	文字开始时间，单位：毫秒	
+ sn	str	非必须	分句转写的id，反馈问题时使用	
+ corpus_no	str	非必须	整段转写的id，反馈问题时使用	

返回示例：

```
{
  "error_code":0,
  "error_message":"请求成功",
  "audio_duration":1300,
  "result":["欢迎使用百度语音,"],
  "detailed_result":[
     {
      "begin_time":0,
      "corpus_no":"632690ff-40e8-4e59-9964-a381e4c64818_ws",
      "end_time":1300,
      "res":["欢迎使用百度语音,"],
      "sn":"632690ff-40e8-4e59-9964-a381e4c64818_ws_ws_0",
      "words_info":[],
      "language": "Chinese"
     },
  ……]
}
```

语音质检API

接口描述：

语音质检接口可以通过声道进行说话人区分，或者通过模型（8953）对单声道音频进行说话人区分。并且支持创建质检规则，对符合质检规则的ASR文本进行标记。

准备工作

1、账户创建及appid鉴权信息获取可参考快速开发指南-准备工作。

2、access_token鉴权信息获取，可参考鉴权认证机制。

可点击下载python demo示例代码进行测试。

语音质检

语音质检任务创建 请求接口：http://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/detection

HTTP 方法：POST

Headers参数：

参数	参数值	是否必须
Content-Type	application/json	是

Body参数如下：

参数名称	位置	类型	是否必须	说明
access_token	body	string	是	携带用户信息的access_token
speech_url	body	string	url和data二选一	可外网访问的音频文件url链接
speech_data	body	string	url和data二选一	音频文件base64结果
session_id	body	string	是	音频质检任务唯一标识
sample_rate	body	integer	否	音频采样率可选择8000或16000，默认8000
pid	body	integer	是	音频识别模型id，可选择80006
callback_url	body	string	否	质检后结果回调路径
role_num	body	integer	否	pid为80006时，由声道区分说话人数，只允许1/2，默认为1 pid为8953时由算法自动判断说话人，本参数无需填写
enable_detection	body	boolean	否	是否进行质检，默认false
enable_detection_detail	body	boolean	否	是否返回质检结果详情，默认false
is_split_channel	body	boolean	否	是否进行左右声道拆分，默认false 当开启为true时，计费时长=左声道有效时长（不含静音）+右声道有效时长（不含静音）（pid使用8953时参数传false）
categories	body	array[string]	否	应用到质检的规则，不填写默认全部

识别模型 pid 参数

pid	模型	备注
80006	中文音视频字幕模型	轻微噪音环境识别
8953	中文话者分离模型	支持区分说话人

Body请求示例：

```
 {
  "access_token": "24.a7179f3da2d56aXXXXXXXXXXXXXXXX",
  "speech_url": "https://xxxxx.wav",
  "speech_data": "base64",
  "session_id": "test20240220",
  "sample_rate":16000,
  "pid": 80006,
  "role_num": 1,
  "callback_url": "",
  "enable_detection_detail": true,
  "enable_detection": true ,
  "is_split_channel": false
}
```

返回结果：

参数名称	类型	必选	说明
error_code	integer	是	请求状态码
error_message	string	是	请求状态
result	object	是	
+ session_id	string	是	音频质检任务唯一标识

返回示例：

```
 {
 "error_code": 0,
 "error_message": "请求成功",
 "result": {
  "session_id": "test20240220"
 }
}
```

语音质检任务查询

请求接口：http://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/detection/result

HTTP 方法：GET

Headers参数：

参数	参数值	是否必须
Content-Type	application/json	是

Body参数如下：

名称	位置	类型	必选	说明
access_token	body	string	是	携带用户信息的access_token
session_id	body	string	是	音频质检任务唯一标识

Body请求示例：

```
 {
  "access_token": "24.a7179f3da2d56aXXXXXXXXXXXXXXXX",
  "session_id": "test20240220"
}
```

返回结果：

参数名称	类型	必选	说明
error_code	integer	是	请求状态码
error_message	string	是	请求状态
result	object	是	语音质检结果
+push_status_desc	string	是	推送状态描述
+session_id	string	是	音频唯一标识
+state	integer	是	质检过程的状态，不等于0、1、2为处理失败
+state_desc	string	是	质检过程的状态描述分别为： "任务已提交，排队等待中" "执行中" "处理成功"
+asr_text	string	是	音频文件转写结果，句与句之间空格
+call_duration	integer	是	通话持续时长，单位秒
+silence_duration	integer	是	静音时长，单位秒
+asr_result	[object]	是	转写结果
++sentence	string	是	单句文本
++speaker_id	integer	是	所属发言人
++begin_time	integer	是	单句开始时间戳
++end_time	integer	是	单句结束时间戳
++voice_speed	integer	是	语速
++voice_power	integer	是	音量
+pass_detection	string	是	质检是否通过，"success"- 通过 "failed" - 不通过
+detection_result	[object]	是	质检结果，返回录音命中的所有规则和对应关键词
++category	string	是	规则类别
++keyword	[string]	是	匹配的关键字
+detection_detail	object	是	质检结果详情，配置了获取详情返回数据
++hit	boolean	是	录音是否命中质检规则
++keyword	[string]	是	录音命中的所有关键词
++detail	[object]	是	句维度命中情况
+++text	string	是	该句文本
+++hit	boolean	是	该句是否命中质检规则
+++detail	[object]	是	该句命中的质检分类
++++category	string	是	质检分类
++++hit	boolean	是	是否命中
++++keyword	[string]	是	命中的分类下所有关键词
++++detail	[object]	是	该句命中的该分类下的质检内容
+++++content	string	是	质检内容
+++++hit	boolean	是	是否命中
+++++keyword	[string]	是	命中的关键词

返回示例：


```
{
 "error_code": 0,
 "error_message": "请求成功",
 "result": {
  "session_id": "test20240220",
  "state": 0,
  "state_desc": "处理成功",
  "asr_text": "你是谁？ 我是小度",
  "call_duration": 10,
  "silence_duration": 5,
  "asr_result": [
   {
    "sentence": "你是谁？",
    "speaker_id": 0,
    "begin_time": 1,
    "end_time": 3,
    "voice_speed": 0,
    "voice_power": 0
   } , {
    "sentence": "我是小度",
    "speaker_id": 1 ,
    "begin_time":,5,
    "end_time": 7,
    "voice_speed": 0,
    "voice_power": 0
   }

  ],
  "pass_detection": "failed",
  "detection_result": [
   {
    "category": "类别1",
    "keyword": [
     "小度"
    ]
   }
  ],
  "detection_detail": {
   "hit": 是,
   "keyword": [
    "小度"
   ],
   "detail": [
    {
     "text": "我是小度",
     "hit": 是,
     "detail": [
      null
     ]}]}}
}
```


质检规则管理

质检规则查询 请求接口：http://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/detection/rule

HTTP 方法：GET

Headers参数：

参数	参数值	是否必须
Content-Type	application/json	是

Body参数如下：

名称	位置	类型	必选	说明
access_token	body	string	是	携带用户信息的access_token
category	body	string	否	规则类别，支持模糊搜索，不传则查询全部
page_no	body	integer	否	分页规则-页数
page_size	body	integer	否	分页规则-每页条数，默认10

Body请求示例：

```
 {
   "access_token": "24.a7179f3da2d56aXXXXXXXXXXXXXXXX ",
   "category": "类别1",
   "page_no":1,
   "page_size":1
}
```

返回结果：

名称	类型	必选	说明
error_code	integer	是	请求状态码
error_message	string	是	请求状态
results	object	是	详情列表
+total	integer	是	符合条件的规则总数
+rule	[object]	是	查询到的规则
++category	string	是	规则类别
++description	string	是	规则类别描述
++content	[string]	是	规则内容

返回示例：

```
{
 "error_code": 0,
 "error_message": "请求成功",
 "results": {
   "total": 0,
   "rule": [
    {
    "category": "类别1",
    "description": "类别1的描述",
    "contents": [ "贷款",
                "会议||例会",
                "缺货&&退款",
                "!逾期",
                "未偿还&&!逾期",
                "(会议||例会) && (缺货||退款)"]
   }
  ]
 }
}
```

质检规则新增 请求接口：http://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/detection/rule

HTTP 方法：POST

Headers参数：

参数	参数值	是否必须
Content-Type	application/json	是

Body参数如下：

名称	位置	类型	必选	说明
access_token	body	string	是	携带用户信息的access_token
category	body	string	是	规则类别
description	body	string	是	规则类别描述
contents	body	array[string]	是	质检规则内容，每条质检规则内部可应用“或与非”逻辑关系

contents规则特殊说明：

一、A（包含A即为命中）

二、A||B （A或B包含任一即为命中）

三、A&&B （A与B均包含即为命中）

四、!B（不包含B即为命中）

五、A&&!B（包含A但不包含B即为命中）

六、用小括号隔开，可随意组合

Body请求示例：

```
 {
  "access_token": "24.a7179f3da2d56aXXXXXXXXXXXXXXXX ",
  "category": "类别1",
  "description": "类别1的描述",
  "contents":["贷款",
              "会议||例会",
              "缺货&&退款",
              "!逾期",
              "未偿还&&!逾期",
              "(会议||例会) && (缺货||退款)"]
}
```

返回结果：

名称	类型	必选	说明
error_code	integer	是	请求状态码
error_message	string	是	请求状态

返回示例：

```
 {
  "error_code": 0,
  "error_message": "请求成功"
}
```

质检规则修改 请求接口：http://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/detection/rule

HTTP 方法：PUT

Headers参数：

参数	参数值	是否必须
Content-Type	application/json	是

Body参数如下：

名称	位置	类型	必选	说明
access_token	body	string	是	携带用户信息的access_token
category	body	string	是	规则类别
description	body	string	是	规则类别描述
contents	body	array[string]	是	质检规则内容，每条质检规则内部可应用“或与非”逻辑关系

contents规则特殊说明：

一、A（包含A即为命中）

二、A||B （A或B包含任一即为命中）

三、A&&B （A与B均包含即为命中）

四、!B（不包含B即为命中）

五、A&&!B（包含A但不包含B即为命中）

六、用小括号隔开，可随意组合

Body请求示例：

```
{
  "access_token": "24.a7179f3da2d56aXXXXXXXXXXXXXXXX ",
  "category": "类别1",
  "description": "类别1的描述",
  "contents":["贷款",
              "会议||例会",
              "缺货&&退款",
              "!逾期",
              "未偿还&&!逾期",
              "(会议||例会) && (缺货||退款)"]
}
```

返回结果：

名称	类型	必选	说明
error_code	integer	是	请求状态码
error_message	string	是	请求状态

返回示例：

```
{
 "error_code": 0,
 "error_message": "请求成功"
}
```

质检规则删除

请求接口：http://aip.baidubce.com/rest/2.0/speech/publiccloudspeech/v1/voice/detection/rule

HTTP 方法：DELETE

Headers参数：

参数	参数值	是否必须
Content-Type	application/json	是

Body参数如下：

名称	位置	类型	必选	说明
access_token	body	string	是	携带用户信息的access_token
category	body	string	是	待删除的规则类别

Body请求示例：

```
{
  "access_token": "24.a7179f3da2d56aXXXXXXXXXXXXXXXX ",
  "category": "类别1",
}
```

返回结果：

名称	类型	必选	说明
error_code	integer	是	请求状态码
error_message	string	是	请求状态

返回示例：

```
 {
  "error_code": 0,
  "error_message": "请求成功"
}
```

呼叫中心语音-音频文件转写（8K）

接口描述

音频文件转写接口可以将大批量的音频文件异步转写为文字。适合音视频字幕生产、批量录音质检、会议内容总结、录音内容分析等场景，一般12小时内返回识别接口。

步骤： 1、根据音频url、音频格式、语言id以及采样率等参数创建音频转写任务，获取task_id参数。 2、根据task_id的数组批量查询音频转写任务结果。

在线调试&示例代码

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。

Python Demo 点击下载 （文件为.zip压缩包，若无法打开时，可尝试在文件后上增加".zip“后缀）

创建音频转写任务-请求说明

请求接口：https://aip.baidubce.com/rpc/2.0/aasr/v1/create

HTTP 方法：POST

URL参数：

参数	取值
access_token	通过 API Key 和 Secret Key 获取的 access_token，参考Access Token获取

JSON方式上传音频

Body中放置请求参数，语音数据和其他参数通过标准 JSON 格式串行化 POST 上传，包括的参数如下：

参数名	类型	是否必需	对外状态	取值范围
speech_url	str	是	音频url	可使用百度云对象存储进行音频存储，生成云端可外网访问的url链接，音频大小不超过500MB
format	str	是	音频格式	["mp3", "wav", "pcm","m4a","amr"]编码 16bits 位深
pid	int	是	语言类型	[1134（呼叫中心中文普通话）]
rate	int	是	采样率	[8000]固定值
channel	int	否	声道	[1、2]

若音频采样率为16k，请使用音频文件转写

Body请求示例：

```
 {
   "speech_url": "https://platform.bj.bcebos.com/sdk%2Fasr%2Fasr_doc%2Fdoc_download_files%2F16k.pcm",
   "format": "pcm",
   "pid": 1134,
   "rate": 8000,
   "channel": 1    //双声道按声道识别请输入“2”
}
```

创建音频转写任务-返回说明

返回参数

参数名	类型	是否必需	对外状态
log_id	int	是	log id
task_id	str	否	任务id
task_status	str	否	任务状态
error_code	int	否	错误码
error_msg	str	否	错误信息

Body返回示例：

```
  # 创建成功
{
   "log_id": 12345678,
   "task_status": "Created" ,
   "task_id":  "234acb234acb234acb234acb"  #注意保存该id，用于后续请求识别结果
}
 # 创建失败，缺少参数
{
   "error_code": 336203,
   "error_msg": "missing param: speech_url",
   "log_id": 5414433131138366128
}
```

注意：查询识别结果时，需要该步骤返回的task_id来进行请求。请注意保存task_id列表。

查询音频转写任务-请求说明

请求接口：https://aip.baidubce.com/rpc/2.0/aasr/v1/query

HTTP 方法：POST

URL参数：

参数	取值
access_token	通过API Key和Secret Key获取的access_token，参考Access Token获取

Body中放置请求参数，参数如下：

参数名	类型	是否必需	描述	取值范围
task_ids	list	是	任务id	task_ids为空，返回空任务结果列表；单次查询任务数不超过200个

请求示例：

```
 {
   "task_ids": ["234acb234acb234acb234acb", "234acb234acb234acb234acd", "234acb234acb234acb234acbe"]
}
```

查询音频转写任务-返回说明

返回参数：

参数名	类型	是否必需
log_id	int	是
tasks_info	list	否
+task_id	str	是
+task_status	str	是
+task_result	dict	否
++corpus_no	str	否
++result	str	否
++detailed_result	list	否
++err_no	int	否
++err_msg	str	否
++sn	str	否
error_code	int	否
error_msg	str	否
error_info	list	否

返回示例：

```
{
  "log_id": 12345678,
  "tasks_info": [
   { # 转写中
    "task_status": "Running"
    "task_id": "234acb234acb234acb234acb",
   },
   { # 转写失败
    "task_status": "Failure"
    "task_id": "234acb234acb234acb234acd",
    "task_result": {
     "err_no": 3301
     "err_msg": "speech quality error",
     "sn": "xxx"
    }
   },
   { # 转写成功
    "task_status": "Success",
    "task_result": {
     "left"{
      "result": [
          "观众朋友大家好，欢迎收看本期视频哦。毕竟..."
      ],
      "detailed_result": [
          {
```

```
            {
            "res": [
                "观众朋友大家好，欢迎收看本期视频哦。"
            ],
            "end_time": 6700,
            "begin_time": 4240,
            "words_info": [
            {
                        "end_time": 1750,
                        "words": "观众",
                        "begin_time": 1100
                    }
                ],
            "sn": "257826606251573543780",
            "corpus_no": "6758319075297447880"
          }
          ...
      ],
      "corpus_no": "6758319075297447880"
      "error_no": 0
    },
      "right"{
      "result": [
         "观众朋友大家好，欢迎收看本期视频哦。毕竟..."
      ],
      "detailed_result": [
          {
          "res": [
              "观众朋友大家好，欢迎收看本期视频哦。"
            ],
            "end_time": 6700,
            "begin_time": 4240,
            "words_info": [
            {
                        "end_time": 1750,
                        "words": "观众",
                        "begin_time": 1100
                    }
                ],
            "sn": "257826606251573543780",
            "corpus_no": "6758319075297447880"
          }
          ...
       ],
       "corpus_no": "6758319075297447880"
       "error_no": 0
      }
                "audio_duration": 6800
    },
    "task_id": "234acb234acb234acb234ace"
    }
 ]
}
```


呼叫中心语音-语音识别（8K）

接口描述及运行环境

本文档是百度呼叫中心语音MRCP的用户指南。

本程序做为MRCP Server端，集成了呼叫中心8K采样率语音识别(ASR)和呼叫中心专属发音人语音合成(TTS)两种能力，用户可分别单独使用某一种或同时使用。

接入步骤

- 参考"AI接入指南"，创建应用，获取AppID、API Key、Secret Key，用于后续配置使用
- 点击呼叫中心语音解决方案MrcpServer完成Mrcp Server下载；
- 在下载的对应文件中修改相关信息，启动服务并进行程序验证

开发环境

- 开发环境依赖：

  - Linux 64位 centos 6u3、centos 7+版本

  - gcc: 8.2以上版本libc，CXX11。程序自带compiler.tar.gz压缩包

- 并发受机器内存、核数等性能影响。

  - 参考：4核8G内存，并发数为20。

音频格式

要求项	取值要求
采样率	8KHz
采样精度	16bits
声道	单声道

- 音频内容为：清晰的真人发音，无背景音或其它噪音，日常用语。
- 开发者Mrcp Client端发送的音频格式通过sip协议交互约定，目前mrcp server支持的音频格式有: PCMU PCMA L16/96/8000

语言及模型支持

支持中文普通话

发音人

目前支持九位发音人，详情见呼叫中心-语音合成

调用流程

Mrcp下载与目录

点击呼叫中心语音解决方案MrcpServer完成Mrcp Server下载；

目录结构

```
 ${SERVER_ROOT}
|--- README   // 说明文件
|--- bootstrap.sh  // 环境初始化
|--- compiler.tar.gz  // gcc-8.2编译器lib库
|--- mrcp-server  // 主程序模块
 |--- audio  // 音频保存目录
 |--- bin     // 服务可执行程序
  |--- unimrcpserver_control // 启动脚本
 |--- conf   // 配置文件
  |--- unimrcpserver.xml  // 可执行程序框架配置项
  |--- unimrcpserver_control.conf // 启动配置项
  |--- mrcp-asr.conf   // 识别插件配置项
  |--- mrcp-proxy.conf  // 合成插件配置项
  |--- comlog.conf   // 日志配置项
 |--- data    // 资源文件
 |--- lib     // 依赖库
 |--- log     // 日志目录
 |--- plugin   // 插件目录
```

鉴权与IP参数配置

用户首先需进行相关配置以启动程序。一般地，配置采用默认值即可，需要用户修改的主要有:

- 在主程序配置文件 ${SERVER_ROOT}/mrcp-server/conf/unimrcpserver.xml 中配置本程序IP，具体位置：unimrcpserver->properties->ip，可选用多种方式，只能同时使用一种：

```
 * type="auto"，由程序自动获取机器IP。默认配置，用户无需修改。
* type="iface"，填写网口名称，较少使用。
* 直接手动填写IP，如：<ip>10.10.0.1</ip>，可以使用。
```

- 语音识别配置conf/mrcp-asr.conf中更改AUTH_APPID和AUTH_APPKEY为从百度官方获取的APPID和API Key的值。
- 语音合成配置conf/mrcp-proxy.conf中更改AUTH_APPID和AUTH_APPKEY为从百度官方获取的APPID和API Key的值。
- 启动配置文件 ${SERVER_ROOT}/mrcp-server/conf/unimrcpserver_control.conf，用于监测相应IP和端口，判断程序是否启动成功。搜索到_check_cmd_pro="./bin/check 127.0.0.1 1544"的位置：

```
 * 如果主程序配置IP采用默认type="auto"方式，则该处"127.0.0.1"不用修改。
* 如果主程序配置IP使用手动填写IP方式，则该处的IP值修改为与它相同的值。
```

详细的配置说明见模块内README文件-${SERVER_ROOT}/mrcp-server/README

服务启动

1. 初次下载MRCP server安装包，需要在 ${SERVER_ROOT}/ 目录下，以 **root** 权限执行 bootstrap.sh 脚本，以完成百度自带gcc8.2环境配置。
2. 程序调试阶段，建议在程序目录${SERVER_ROOT}/mrcp-server/下，手动使用命令"./bin/unimrcpserver -r . &"启动程序，方便查看输出、定位问题。使用netstat -nlp | grep unimrcp，查看IP和端口5060/1544/1554，看是否启动成功。如果公司有防火墙限制，请记得将这三个端口打开。
3. 使用启动脚本，以守护进程形式启动程序。在生产环境使用时，建议使用该方式：
    - 启动：在${SERVER_ROOT}/mrcp-server目录执行 ./bin/unimrcpserver_control start
    - 停止：在${SERVER_ROOT}/mrcp-server目录执行 ./bin/unimrcpserver_control stop
    - 重启：在${SERVER_ROOT}/mrcp-server目录执行 ./bin/unimrcpserver_control restart

4. 进行start前确保系统无mrcp进程；进行stop/restart时确保系统有mrcp进程。如果不行，通过 ps aux | grep mrcp 尝试将所有mrcp相关的进程kill掉，重新 start。

请求说明

鉴权信息

如需使用语音识别能力，则需在下载的Mrcp Server中对识别配置文件进行参数更改 ${SERVER_ROOT}/mrcp-server/conf/mrcp-asr.conf。确保AUTH_APPID和AUTH_APPKEY填写正确。

```
 * AUDIO_CONTROLLER_ADDR，百度上游服务地址(默认值当前有效)
* AUTH_APPID和AUTH_APPKEY，从百度官网中获取的APPID和API Key的值。
* NEED_SAVE_AUDIO，是否保存语音识别时用户语音，默认1为保存
```

程序验证

首先，将mrcp-server/lib目录加入系统环境变量中，export LD_LIBRARY_PATH=${SERVER_ROOT}/mrcp-server/lib:$LD_LIBRARY_PATH，注意将${SERVER_ROOT}修改为程序真实路径。

在主程序启动后，可使用自带的测试工具进行验证。conf/client-profiles/unimrcp.xml 是测试工具的配置文件，需要将其中的unimrcpclient->settings->sip-settings->server-ip的值修改为本机IP，端口设置为主程序端口，如5060。

切换到 ${SERVER_ROOT}/mrcp-server/bin 目录下

1. 验证语音识别正确性，则执行 ./asrclient，输入 run grammar.xml xeq.pcm，可看到返回的识别结果，使用 quit 退出。识别的是 ../data 目录下的xeq.pcm音频，也可查看log目录下日志mrcp_debug.log是否有识别结果。该工具只为测试部署正确性，测试较长音频时，会因为vad截断而只识别音频里的第一句话；只支持pcm格式音频识别；该工具也不支持批量音频识别。

2. 用户呼叫软件如freeswitch，在与mrcp server通信时可能需要传递grammar文件，请使用mrcp server中data/grammar.xml文件。

3. 呼叫中心场景，噪声识别是个很难解的问题。如果用户想忽略因噪声而误识别出的单字结果，例如嗯，啊等，可在conf/mrcp-asr.conf中设置DETECT_START_OF_INPUT_BY_VAD为0，并在SINGLE_WORD_WHTIL_LIST配置的白名单文件中进行相应配置。**功能开启后，单字识别结果的场景下，只会识别白名单中的单字**。

返回说明

- 语音识别. 往前端返回如下xml格式的内容：

```
 <?xml version="1.0" encoding="UTF-8" ?>
<result>
   <asr confidence="100">今天天气怎么样</asr>
   <ext>
      <logid>6a373e36722811e9_2_1</logid>
   </ext>
</result>
```

- 识别结果格式支持模版配置，在conf/mrcp-asr.conf中，修改XML_ASR_RESULT_TEMPLATE 的值即可。在conf目录下默认定义了几种模版，用户可根据需要选择一项或自己定义。

SDK文档

语音合成

短文本在线合成 HTTP SDK

简介

简介

Hi，您好，欢迎使用百度语音合成服务。

本文档主要针对开发者，描述百度语音合成接口服务的相关技术内容。如果您对文档内容有任何疑问，可以在百度云控制台内提交工单联系我们，咨询问题类型请选择**人工智能服务**；

接口能力

接口名称	接口能力简要描述
语音合成	将计算机自己产生的、或外部输入的文字信息转变为可以听得懂的、流利的口语输出的技术。

注意事项

目前本SDK的功能同REST API，需要联网调用http接口 。REST API 仅支持最多512字（1024 字节)的音频合成，合成的文件格式为mp3。**没有其他额外功能。** 如果需要使用离线合成等其它功能，请使用Android或者iOS 合成 SDK

请严格按照文档里描述的参数进行开发。请注意以下几个问题：

1. 合成文本长度必须小于1024字节，如果本文长度较长，可以采用多次请求的方式。切忌文本长度超过限制。

2. 新创建语音合成应用可以在控制台领取免费额度。

3. 必填字段中，严格按照文档描述中内容填写。

版本更新记录

PHP

上线日期	版本号	更新内容
2021.2.26	4.15.4	接口统一升级
2017.5.11	1.0.0	语音合成服务上线

Java | 上线日期 | 版本号 | 更新内容 | | --------- | ---- | ------------------------------ | | 2022.2.26 | 4.15.4 | 接口统一升级 | | 2018.1.26 | 4.1.1 | 语音合成修复服务端不返回Content-Type的情况；新增日志记录| | 2017.12.12 | 3.4.1 | 语音合成问题修复 | | 2017.11.10 | 3.3.1 | 语音合成返回无sn问题修复 | | 2017.10.18 | 3.2.1 | 使用proxy问题修复 | | 2017.8.25 | 3.0.0 | 更新sdk打包方式：所有AI服务集成一个SDK | | 2017.7.14 | 1.0.1 | 更新sdk打包方式 | | 2017.6.30 | 1.0.0 | 新增语音合成服务接口 |

Python

上线日期	版本号	更新内容
2022.2.26	4.15.4	接口统一升级
2017.5.11	1.0.0	语音合成服务上线

C++

上线日期	版本号	更新内容
2022.2.26	4.15.4	接口统一升级
2017.12.21	0.4.0	更新了语音合成返回数据逻辑
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	在线语音合成第一版

C# | 上线日期 | 版本号 | 更新内容 | | --------- | ----- | ---- | | 2022.2.26 | 4.15.4 | 接口统一升级 | | 2018.4.2 | 3.4.0 | 新增.Net Core支持；新增超时参数 | | 2017.9.12 | 3.0.0 | 更新SDK打包方式：所有AI服务集成一个SDK | | 2017.6.30 | 1.0.0 | 新增语音合成服务接口 |

Node.js

上线日期	版本号	更新内容
2022.2.26	4.15.4	接口统一升级
2017.12.21	2.0.0	实现代码重构，接口返回标准promise对象
2017.7.13	1.2.1	修复模块加载在linux下失败的问题
2017.6.30	1.2.0	新增语音合成服务接口

在线合成REST-API-Python-SDK

快速入门

安装语音合成 Python SDK

语音合成 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
│    └─── speech.py //语音合成
└─── 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AipSpeech

AipSpeech是语音合成的Python SDK客户端，为使用语音合成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AipSpeech：

```
 from aip import AipSpeech

""" 你的 APPID AK SK """
APP_ID = '你的 App ID'
API_KEY = '你的 Api Key'
SECRET_KEY = '你的 Secret Key'

client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)
```

在上面代码中，常量`APP_ID`在百度云控制台中创建，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。

配置AipSpeech

如果用户需要配置AipSpeech的网络请求参数(一般不需要配置)，可以在构造AipSpeech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

语音合成

接口描述

基于该接口，开发者可以轻松的获取语音合成能力

请求说明

- 合成文本长度必须小于1024字节，如果本文长度较长，可以采用多次请求的方式。文本长度不可超过限制

举例，要把一段文字合成为语音文件：

```
  result  = client.synthesis('你好百度', 'zh', 1, {
    'vol': 5,
})

# 识别正确返回语音二进制 错误则返回dict 参照下面错误码
if not isinstance(result, dict):
    with open('audio.mp3', 'wb') as f:
        f.write(result)
```

参数	类型	描述	是否必须
tex	String	合成的文本，文本长度必须小于1024GBK字节，建议每次请求文本不超过120字节，约为60个汉字或者字母数字。 请注意计费统计依据：120个GBK字节以内（含120个）记为1次计费调用；每超过120个GBK字节则多记1次计费调用。	是
cuid	String	用户唯一标识，用来区分用户， 填写机器 MAC 地址或 IMEI 码，长度为60以内	否
spd	String	语速，取值0-9，默认为5中语速	否
pit	String	音调，取值0-9，默认为5中语调	否
vol	String	音量，取值0-15，默认为5中音量	否
per	String	普通发音人选择：度小美=0(默认)，度小宇=1，，度逍遥（基础）=3，度丫丫=4	否
per	String	精品发音人选择：度逍遥（精品）=5003，度小鹿=5118，度博文=106，度小童=110，度小萌=111，度米朵=103，度小娇=5	否

tex字段2次urlencode

由于urlencode有两个标准 RFC 1738和RFC 3986. 百度为了更好地兼容，支持1次及2次urlencode， 其中2次urlencode可以覆盖全部的特殊字符。SDK内部加了1次urlencode，因而推荐传递tex 参数时再做1次urlencode编码。

测试用例：“1+1=2”。 依靠SDK内的1次urlencode时，“+”可能会没有合成。

返回样例：

```
 // 成功返回二进制文件流
// 失败返回
{
    "err_no":500,
    "err_msg":"notsupport.",
    "sn":"abcdefgh",
    "idx":1
}
```

在线合成REST-API-JAVA-SDK

快速入门

安装Speech Java SDK

Speech 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
    ├─── speech
    │      └─── AipSpeech       //AipSpeech类
    └─── util                    //工具类
```

支持 JAVA版本：1.7+

查看源码

Java SDK代码现已公开，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/java-sdk

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.在官方网站下载识别、合成 RESTful API 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` `log4j-1.2.17.jar`。

其中，`version`为版本号，添加完成后，用户就可以在工程中使用Speech Java SDK。

新建AipSpeech

AipSpeech是语音识别的Java客户端，为使用语音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AipSpeech,初始化完成后建议**单例使用**,避免重复获取access_token：

```
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AipSpeech
    AipSpeech client = new AipSpeech(APP_ID, API_KEY, SECRET_KEY);

    // 可选：设置网络连接参数
    client.setConnectionTimeoutInMillis(20000);
    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
    TtsResponse res = client.synthesis("你好百度", "zh", 1, null);
    byte[] data = res.getData();
    JSONObject res1 = res.getResult();
    if (data != null) {
      try {
        Util.writeBytesToFileSystem(data, "output.mp3");
      } catch (IOException e) {
        e.printStackTrace();
      }
    }
    if (res1 != null) {
      System.out.println(res1.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。

配置AipSpeech

如果用户需要配置AipSpeech的一些细节参数，可以在构造AipSpeech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

语音合成

接口描述

基于该接口，开发者可以轻松的获取语音合成能力

请求说明

- 合成文本长度必须小于1024字节，如果本文长度较长，可以采用多次请求的方式。文本长度不可超过限制

举例，要把一段文字合成为语音文件：

```
 public void synthesis(AipSpeech client)
{
    TtsResponse res = client.synthesis("你好百度", "zh", 1, null);
 System.out.println(res.getErrorCode());

 // 设置可选参数
 HashMap<String, Object> options = new HashMap<String, Object>();
    options.put("spd", "5");
    options.put("pit", "5");
    options.put("per", "4");
    TtsResponse res = client.synthesis("你好百度", "zh", 1, options);
    System.out.println(res.getErrorCode());
    JSONObject result = res.getResult();   //服务器返回的内容，合成成功时为null,失败时包含error_no等信息
    byte[] data = res.getData();           //生成的音频数据
}
```

参数	类型	描述	是否必须
tex	String	合成的文本，文本长度必须小于1024GBK字节，建议每次请求文本不超过120字节，约为60个汉字或者字母数字。 请注意计费统计依据：120个GBK字节以内（含120个）记为1次计费调用；每超过120个GBK字节则多记1次计费调用。	是
cuid	String	用户唯一标识，用来区分用户， 填写机器 MAC 地址或 IMEI 码，长度为60以内	否
spd	String	语速，取值0-9，默认为5中语速	否
pit	String	音调，取值0-9，默认为5中语调	否
vol	String	音量，取值0-15，默认为5中音量（取值为0时为音量最小值，并非为无声）	否
per	String	普通发音人选择：度小美=0(默认)，度小宇=1，，度逍遥（基础）=3，度丫丫=4	否
per	String	精品发音人选择：度逍遥（精品）=5003，度小鹿=5118，度博文=106，度小童=110，度小萌=111，度米朵=103，度小娇=5	否

tex字段2次urlencode

由于urlencode有两个标准 RFC 1738和RFC 3986. 百度为了更好地兼容，支持1次及2次urlencode， 其中2次urlencode可以覆盖全部的特殊字符。SDK内部加了1次urlencode，因而推荐传递tex 参数时再做1次urlencode编码。

测试用例：“1+1=2”。 依靠SDK内的1次urlencode时，“+”可能会没有合成。

返回样例：

返回TtsResponse类。 如果合成成功,下行数据为二进制语音文件，包含在data中。 如果合成出现错误，则会填充返回值到result中。

返回失败：

```
 // 失败返回
{
 "err_no":500,
 "err_msg":"notsupport.",
 "sn":"abcdefgh",
 "idx":1
}
```

在线合成REST-API-PHP-SDK

快速入门

安装语音合成 PHP SDK

语音合成 PHP SDK目录结构

```
├──── AipSpeech.php          //语音合成
└──── 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.在官方网站下载识别、合成 RESTful API php SDK压缩包。

2.将下载的`aip-php-sdk-version.zip`解压后，复制AipSpeech.php以及lib/*到工程文件夹中。

3.引入AipSpeech.php

新建AipSpeech

AipSpeech是语音合成的PHP SDK客户端，为使用语音合成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AipSpeech：

```
 require_once 'AipSpeech.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 AipSpeech(APP_ID, API_KEY, SECRET_KEY);
```

在上面代码中，常量`APP_ID`在百度云控制台中创建，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。

配置AipSpeech

如果用户需要配置AipSpeech的网络请求参数(一般不需要配置)，可以在构造AipSpeech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

语音合成

接口描述

基于该接口，开发者可以轻松的获取语音合成能力

请求说明

- 合成文本长度必须小于1024字节，如果本文长度较长，可以采用多次请求的方式。文本长度不可超过限制

举例，要把一段文字合成为语音文件：

```
 $result = $client->synthesis('你好百度', 'zh', 1, array(
    'vol' => 5,
));

// 识别正确返回语音二进制 错误则返回json 参照下面错误码
if(!is_array($result)){
    file_put_contents('audio.mp3', $result);
}
```

参数	类型	描述	是否必须
tex	String	合成的文本，文本长度必须小于1024GBK字节，建议每次请求文本不超过120字节，约为60个汉字或者字母数字。 请注意计费统计依据：120个GBK字节以内（含120个）记为1次计费调用；每超过120个GBK字节则多记1次计费调用。	是
cuid	String	用户唯一标识，用来区分用户， 填写机器 MAC 地址或 IMEI 码，长度为60以内	否
spd	String	语速，取值0-9，默认为5中语速	否
pit	String	音调，取值0-9，默认为5中语调	否
vol	String	音量，取值0-15，默认为5中音量	否
per	String	普通发音人选择：度小美=0(默认)，度小宇=1，，度逍遥（基础）=3，度丫丫=4	否
per	String	精品发音人选择：度逍遥（精品）=5003，度小鹿=5118，度博文=106，度小童=110，度小萌=111，度米朵=103，度小娇=5	否

tex字段2次urlencode

由于urlencode有两个标准 RFC 1738和RFC 3986. 百度为了更好地兼容，支持1次及2次urlencode， 其中2次urlencode可以覆盖全部的特殊字符。SDK内部加了1次urlencode，因而推荐传递tex 参数时再做1次urlencode编码。

测试用例：“1+1=2”。 依靠SDK内的1次urlencode时，“+”可能会没有合成。

返回样例：

```
 // 成功返回二进制文件
// 失败返回
{
    "err_no":500,
    "err_msg":"notsupport.",
    "sn":"abcdefgh",
    "idx":1
}
```

在线合成REST-API-C#-SDK

快速入门

安装语音合成 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

语音合成 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.在官方网站下载识别、合成 RESTful API 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.Speech.Tts是语音合成的交互类，为使用语音合成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.Speech.Tts(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

语音合成

接口描述

基于该接口，开发者可以轻松的获取语音合成能力

请求说明

- 合成文本长度必须小于1024字节，如果本文长度较长，可以采用多次请求的方式。文本长度不可超过限制

举例，要把一段文字合成为语音文件：

```
 // 合成
public void Tts()
{
    // 可选参数
    var option = new Dictionary<string, object>()
    {
        {"spd", 5}, // 语速
        {"vol", 7}, // 音量
        {"per", 4}  // 发音人，4：情感度丫丫童声
    };
    var result = client.Synthesis("众里寻他千百度", option);

    if (result.ErrorCode == 0)  // 或 result.Success
    {
        File.WriteAllBytes("合成的语音文件本地存储地址.mp3", result.Data);
    }
}
```

参数	类型	描述	是否必须
tex	String	合成的文本，文本长度必须小于1024GBK字节，建议每次请求文本不超过120字节，约为60个汉字或者字母数字。 请注意计费统计依据：120个GBK字节以内（含120个）记为1次计费调用；每超过120个GBK字节则多记1次计费调用。	是
cuid	String	用户唯一标识，用来区分用户， 填写机器 MAC 地址或 IMEI 码，长度为60以内	否
spd	String	语速，取值0-9，默认为5中语速	否
pit	String	音调，取值0-9，默认为5中语调	否
vol	String	音量，取值0-15，默认为5中音量	否
per	String	普通发音人选择：度小美=0(默认)，度小宇=1，，度逍遥（基础）=3，度丫丫=4	否
per	String	精品发音人选择：度逍遥（精品）=5003，度小鹿=5118，度博文=106，度小童=110，度小萌=111，度米朵=103，度小娇=5	否

tex字段2次urlencode

由于urlencode有两个标准 RFC 1738和RFC 3986. 百度为了更好地兼容，支持1次及2次urlencode， 其中2次urlencode可以覆盖全部的特殊字符。SDK内部加了1次urlencode，因而推荐传递tex 参数时再做1次urlencode编码。

测试用例：“1+1=2”。 依靠SDK内的1次urlencode时，“+”可能会没有合成。

返回样例：

返回TtsResponse类。 如果合成成功，ErrorCode=0, 二进制语音文件，包含在data中。 如果合成出现错误，则会填充除data属性外的其它属性。 如果出现网络错误或者权限验证错误，会抛出相应异常，

在线合成REST-API-Node.js-SDK

快速入门

安装语音合成 Node SDK

语音合成 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
├─── AipSpeech.js               //语音合成交互类
├─── 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.在官方网站下载识别、合成 RESTful API 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语音合成 Node SDK。

直接使用npm安装依赖：

`npm install baidu-aip-sdk`

6.调用示例可以参考 https://github.com/Baidu-AIP/sdk-demo

新建AipSpeechClient

AipSpeechClient是语音合成的node客户端，为使用语音合成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AipSpeechClient：

```
 var AipSpeechClient = require("baidu-aip-sdk").speech;

// 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 AipSpeechClient(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

语音合成

接口描述

基于该接口，开发者可以轻松的获取语音合成能力

请求说明

- 合成文本长度必须小于1024字节，如果本文长度较长，可以采用多次请求的方式。不可文本长度超过限制

举例，要把一段文字合成为语音文件：

```
 // 语音合成
var fs = require('fs');

client.text2audio('百度语音合成测试').then(function(result) {
    if (result.data) {
        fs.writeFileSync('tts.mpVoice.mp3', result.data);
    } else {
        // 服务发生错误
        console.log(result)
    }
}, function(e) {
    // 发生网络错误
    console.log(e)
});

// 语音合成, 附带可选参数
client.text2audio('百度语音合成测试', {spd: 0, per: 4}).then(function(result) {
    if (result.data) {
        fs.writeFileSync('tts.mpVoice.mp3', result.data);
    } else {
        // 服务发生错误
        console.log(result)
    }
}, function(e) {
    // 发生网络错误
    console.log(e)
});
```

接口函数说明：

```
 // 语音合成
// options为可选参数
var options = {spd: 0, per: 4};
client.text2audio(tex, options) ;
```

参数	类型	描述	是否必须
tex	String	合成的文本，文本长度必须小于1024GBK字节，建议每次请求文本不超过120字节，约为60个汉字或者字母数字。 请注意计费统计依据：120个GBK字节以内（含120个）记为1次计费调用；每超过120个GBK字节则多记1次计费调用。	是
cuid	String	用户唯一标识，用来区分用户， 填写机器 MAC 地址或 IMEI 码，长度为60以内	否
spd	String	语速，取值0-9，默认为5中语速	否
pit	String	音调，取值0-9，默认为5中语调	否
vol	String	音量，取值0-15，默认为5中音量	否
per	String	普通发音人选择：度小美=0(默认)，度小宇=1，，度逍遥（基础）=3，度丫丫=4	否
per	String	精品发音人选择：度逍遥（精品）=5003，度小鹿=5118，度博文=106，度小童=110，度小萌=111，度米朵=103，度小娇=5	否

tex字段2次urlencode

由于urlencode有两个标准 RFC 1738和RFC 3986. 百度为了更好地兼容，支持1次及2次urlencode， 其中2次urlencode可以覆

盖全部的特殊字符。SDK内部加了1次urlencode，因而推荐传递tex 参数时再做1次urlencode编码。

测试用例：“1+1=2”。 依靠SDK内的1次urlencode时，“+”可能会没有合成。

返回样例：

如果合成成功，返回结果为对象，其中data字段为音频文件的Buffer对象。

如果返回失败，则返回以下格式的错误对象:

```
 // 失败返回
{
  "err_no":500,
  "err_msg":"notsupport.",
  "sn":"abcdefgh",
  "idx":1
}
```

在线合成REST-API-C++-SDK

快速入门

安装语音合成 C++ SDK

语音合成 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
└─── speech.h                   // 语音合成 交互类
```

最低支持 C++ 11+

直接使用开发包步骤如下：

1.在官方网站下载识别、合成 RESTful API 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 speech.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。

6.调用示例可以参考 https://github.com/Baidu-AIP/sdk-demo

新建client

client是语音合成的C++客户端，为使用语音合成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 "speech.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::Speech client(app_id, api_key, secret_key);
```

在上面代码中，常量APP_ID在百度云控制台中创建，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

语音合成

接口描述

基于该接口，开发者可以轻松的获取语音合成能力

请求说明

- 合成文本长度必须小于1024字节，如果本文长度较长，可以采用多次请求的方式。文本长度不可超过限制

举例，要把一段文字合成为语音文件：

```
 void tts(aip::Speech client)
{
    std::ofstream ofile;
    std::string file_ret;
    std::map<std::string, std::string> options;
    options["spd"] = "9";

    // 合成成功的二进制数据写入文件中
    ofile.open("./test.mp3", std::ios::out | std::ios::binary);

    // 不带可选参数调用
    Json::Value result = client.text2audio("百度语音合成测试", aip::null, file_ret);

    // 带可选参数调用, 参数参考参数列表中的可选参数
    // Json::Value result = client.text2audio("百度语音合成测试", options, file_ret);

    // 如果file_ret为不为空则说明合成成功，返回mp3文件内容回结果
    if (!file_ret.empty())
    {
        ofile << file_ret;
    } else {
        // 服务端合成错误
        std::cout << result.toStyledString();
    }

}
```

参数	类型	描述	是否必须
tex	String	合成的文本，文本长度必须小于1024GBK字节，建议每次请求文本不超过120字节，约为60个汉字或者字母数字。 请注意计费统计依据：120个GBK字节以内（含120个）记为1次计费调用；每超过120个GBK字节则多记1次计费调用。	是
cuid	String	用户唯一标识，用来区分用户， 填写机器 MAC 地址或 IMEI 码，长度为60以内	否
spd	String	语速，取值0-9，默认为5中语速	否
pit	String	音调，取值0-9，默认为5中语调	否
vol	String	音量，取值0-15，默认为5中音量（取值为0时为音量最小值，并非为无声）	否
per	String	普通发音人选择：度小美=0(默认)，度小宇=1，，度逍遥（基础）=3，度丫丫=4	否
per	String	精品发音人选择：度逍遥（精品）=5003，度小鹿=5118，度博文=106，度小童=110，度小萌=111，度米朵=103，度小娇=5	否

tex字段2次urlencode

由于urlencode有两个标准 RFC 1738和RFC 3986. 百度为了更好地兼容，支持1次及2次urlencode， 其中2次urlencode可以覆盖全部的特殊字符。SDK内部加了1次urlencode，因而推荐传递tex 参数时再做1次urlencode编码。

测试用例：“1+1=2”。 依靠SDK内的1次urlencode时，“+”可能会没有合成。

返回样例：

返回Json::value对象。 如果合成成功，error_code为Json::nullValue, 下行数据为二进制语音文件数据。 如果合成出现错误，则会填充除data属性外的其它属性。

返回失败：

```
 // 失败返回
{
    "err_no":500,
    "err_msg":"notsupport.",
    "sn":"abcdefgh",
    "idx":1
}
```

语音合成 Android SDK

1. 文档说明

文档名称	语音离线合成集成文档
所属平台	Android
提交日期	2026-08-11
概述	本文档是百度语音开放平台Andriod SDK的用户指南，描述了**在线合成，离线合成**等相关接口的使用说明。 合成的策略是边下载边播放，区别于Rest Api一次性下载整个录音文件。 语音合成SDK分为在线SDK与离线SDK，离线SDK需要下载音库资源，在线SDK则可直接调用，默认支持流式合成。 其他注意事项见SDK说明。

2. 版本说明

名称	版本号
语音合成	6.4.0
系统支持	android 8.0+
架构支持	armeabi-v7a，arm64-v8a，x86，x86_64

2.1 版本升级改动点说明：

- 提升SDK稳定性
- 离线发音人听感调优
- AKSK鉴权方式不再使用，更新优化部分API接口

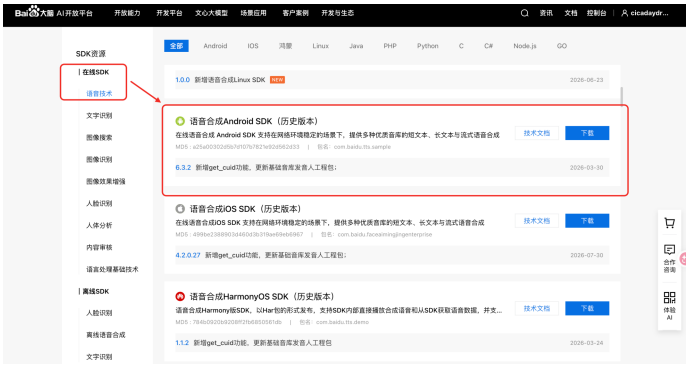
3. SDK说明

文件名称	版本号	说明
tts_1.0.3-9c93a23-20260727.aar	1.0.3	离线合成SDK约10.3 MB

3.1 SDK使用说明：

- 当前版本支持Android 8.0及以上版本安卓系统。若您的系统包含8.0及以下，请前往下载离线SDK历史版本与在线SDK历史版本。

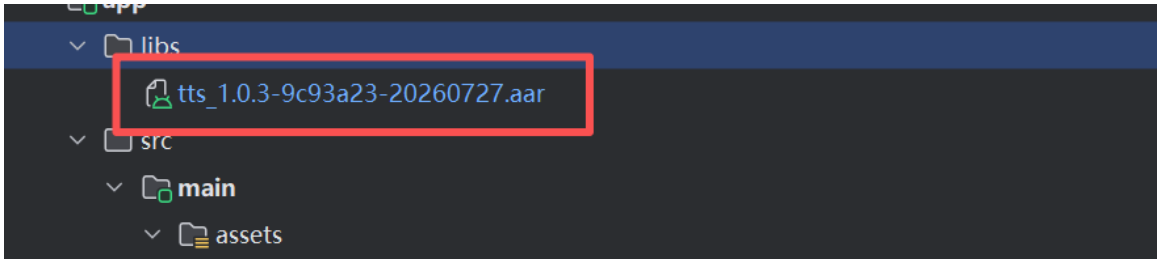




- 离线语音合成SDK内置基础音库，精品音库需要额外下载，请您前往离线合成SDK管理按需下载；若您在2025年11月27日及之前购买序列号或申请SN（序列号），请点击下载往期基础音库资源，及往期精品音库资源。
- 离线语音合成SDK需要申请序列号，将序列号填入SDK后，首次联网会自动下载授权文件。TtsMode.MIX 及TtsMode.OFFLINE的离线合成均需要授权文件没有过期。

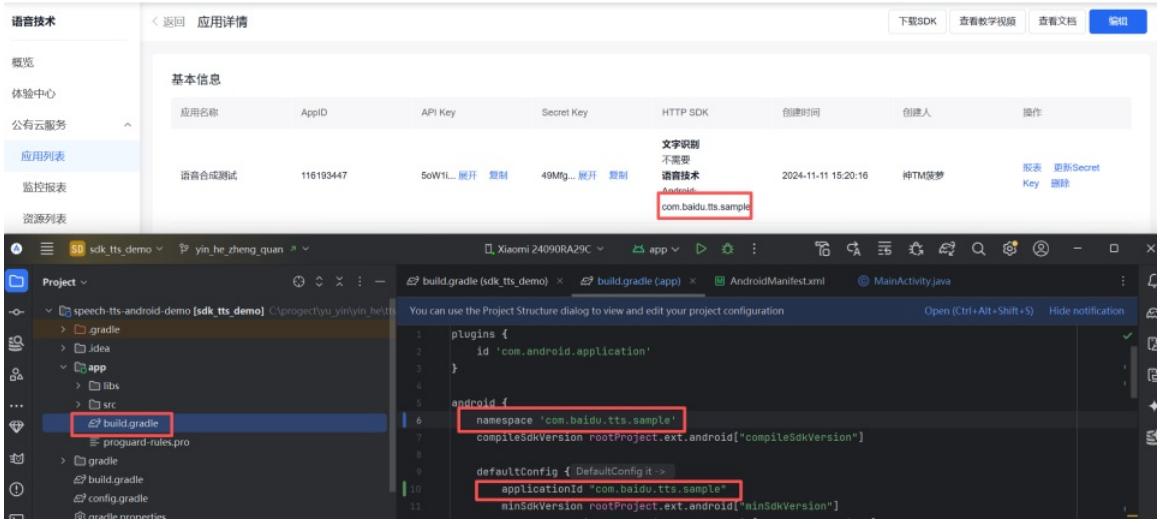
4. Demo运行

从百度云控制台下载Demo，下载完成后解压demo，并用android stdio打开该项目。



4.1 配置包名和签名

从百度云控制台下载Demo之后，需要在build.gradle中配置好包名,且包名和控制台应用包名一致。

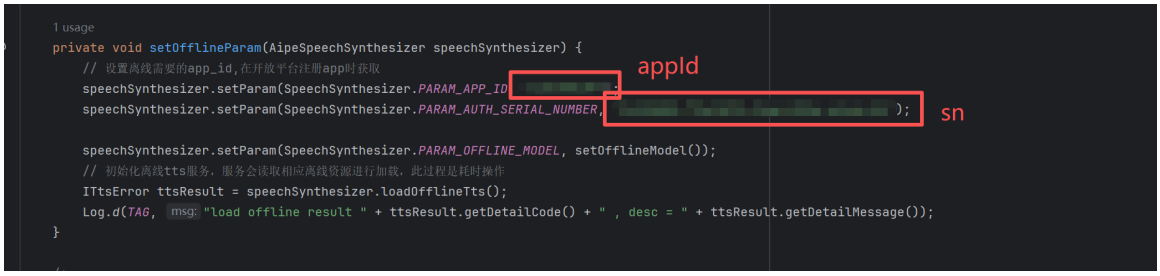


4.2 修改鉴权

4.2.1 离线SDK鉴权

若AppId、包名、序列号SN 3个信息完全正确，SDK将自动下载鉴权文件，否则会有-102或-109错误。

在百度云网站上申请自己的语音合成的应用后，能够获取AppId、AppKey、AppSecret及Android包名 4个鉴权信息，序列号SN申请的具体流程参考语音技术 (baidu.com)。 申请成功后在 MainActivity 中的 setOfflineParam 方法处填写 AppId 和 SN:



4.2.2 在线SDK鉴权

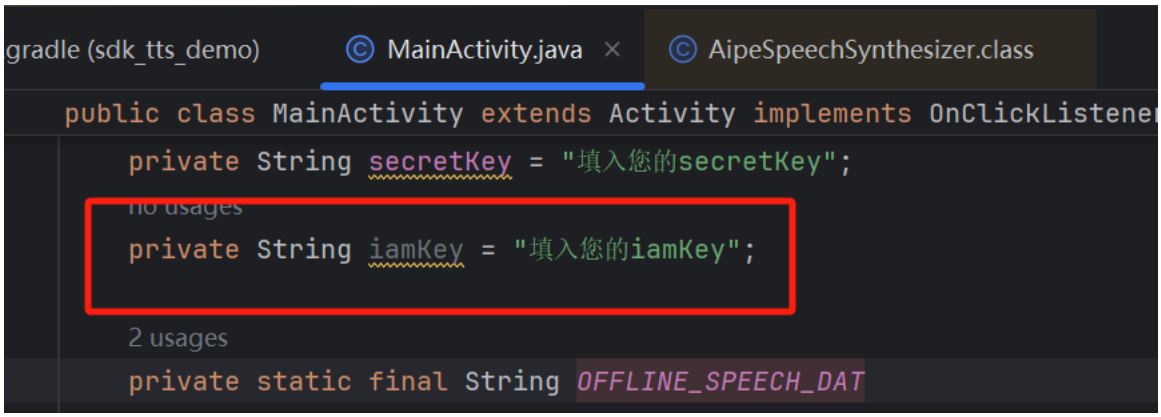
在线sdk分为3种激活方式，实际使用种选择其中一种方式进行激活，3种激活方法详细描述如下：

激活方式	说明
永久iamKey激活	需在官网申请的永久iamKey，使用该方法进行在线鉴权，每次启动app需激活一次，不退出app永久有效
accessToken激活	需在使用appkey和secretKey获取的access_token，使用access_token进行授权，该token存在时限，时间到期后需要从新获取新的token从新授权
临时iamKey激活	与accessToken类似，需获取临时iamKey，该key存在时限，时间到期后需要从新获取新的key从新授权

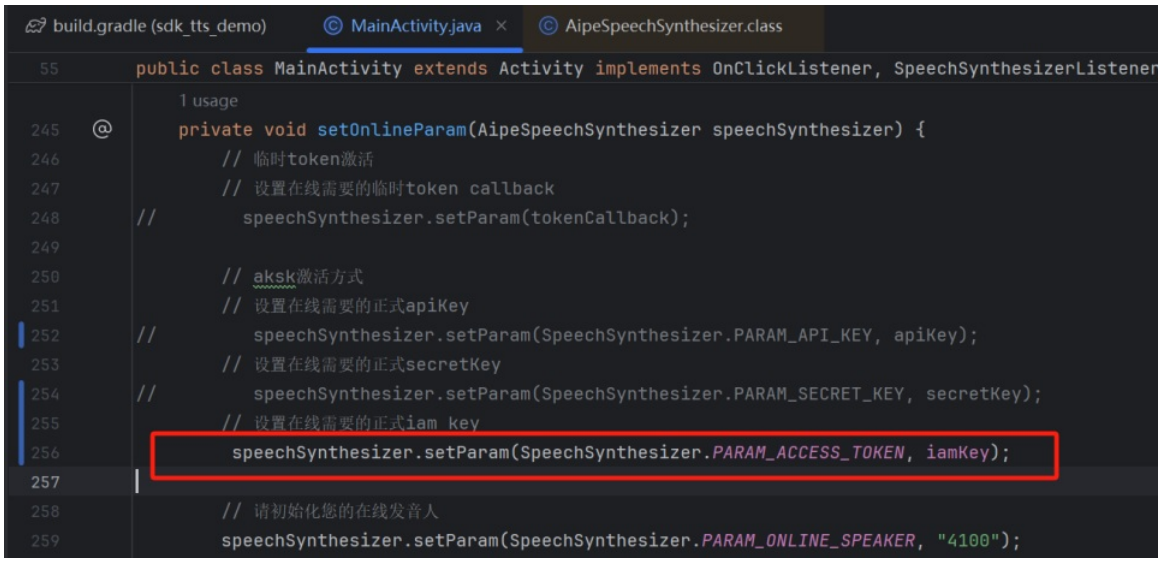
1. 永久iamKey激活方式：

“永久iamKey激活”方式细节如下：

- 将官网申请的永久iamKey放到项目中



- 将iamKey传入接口，此步骤需要使用AipeSpeechSynthesizer类，不能使用SpeechSynthesizer进行设置

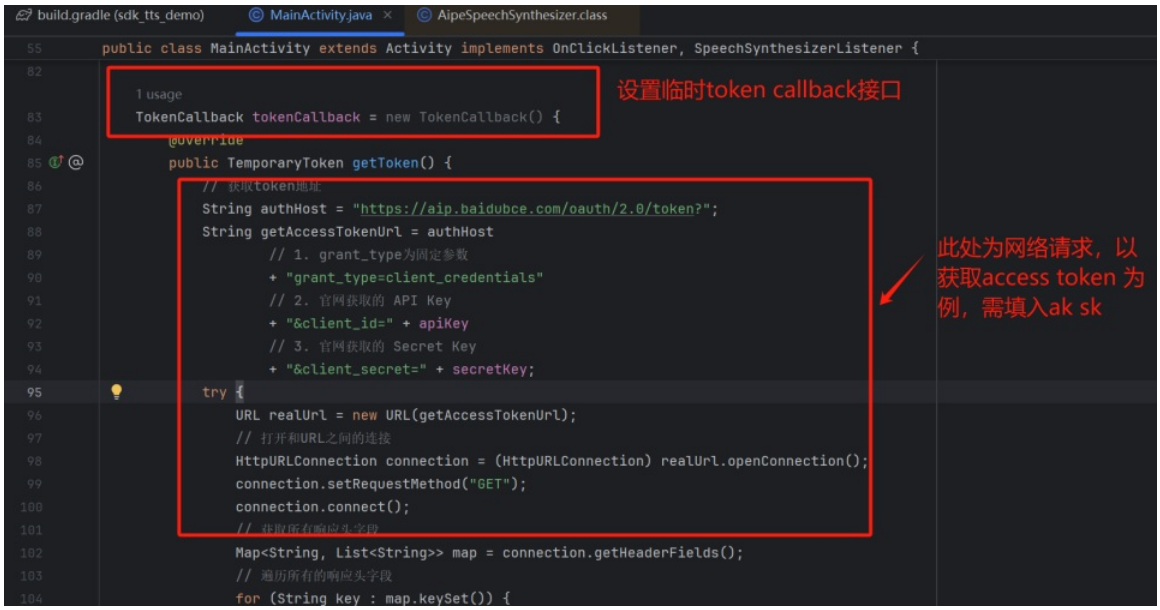


2. 临时accessToken和临时iam key激活方式：

accessToken和临时iamKey为相同接口，接口会随时间失效，客户需要在callback中创建http请求获取到对应token返回给sdk,sdk会在过期前3分钟通过callback重新向客户端获取最新token。

accessToken获取地址：https://cloud.baidu.com/doc/AI_REFERENCE/s/Km3zhy5t7

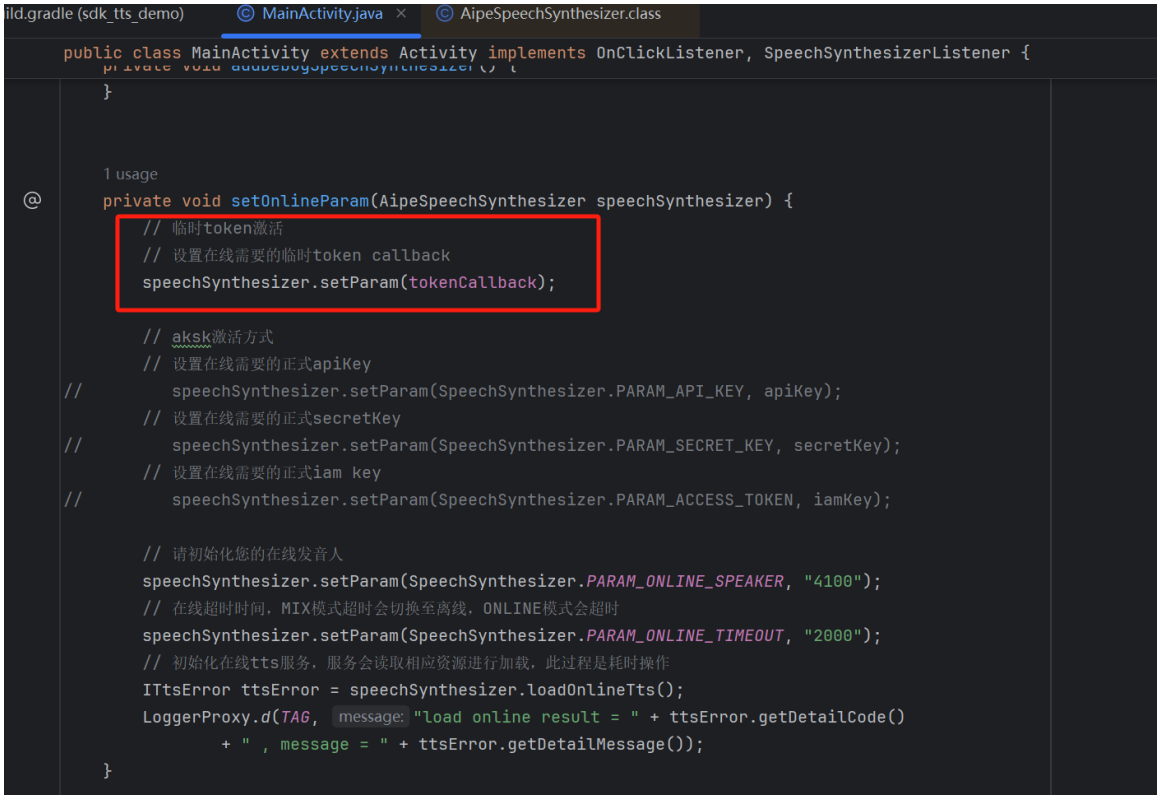
iamKey获取地址：https://cloud.baidu.com/doc/AI_REFERENCE/s/Hm5us339w



- 将accessToken或iamKey 还有过期时间返回给sdk



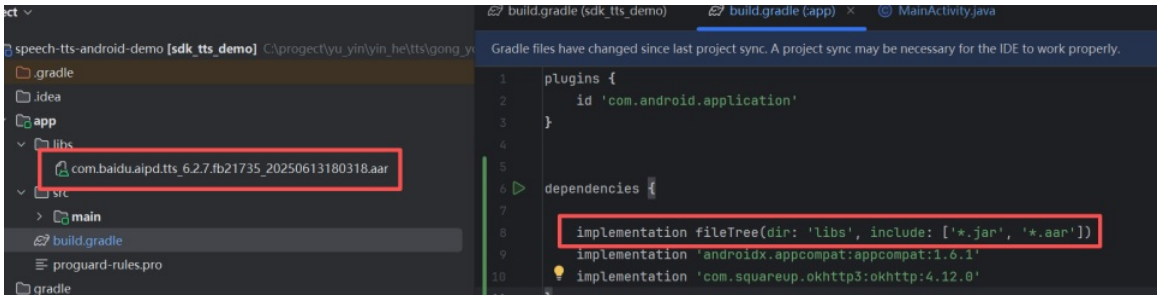
- 将callback传递到sdk中



5. SDK集成

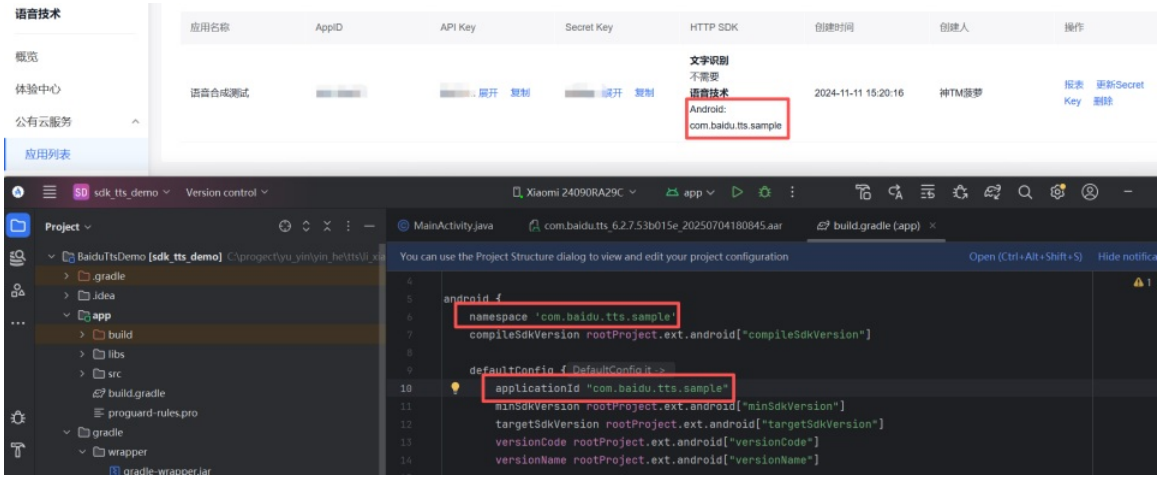
TTS SDK版本为**tts_1.0.3-9c93a23-20260727.aar**。 将app/libs/tts_1.0.3-9c93a23-20260727.aar复制到您的项目的同名目录中，确认在build.gradle文件中引入。

将app/libs/com.baidu.aipd.tts_6.3.2.49abef5_20260209154400.aar复制到您的项目的同名目录中，确认在build.gradle文件中引入。



build.gradle 文件及包名确认

1. 根目录下build.gradle确认下gradle的版本。
2. app/build.gradle 确认下 applicationId 包名是否与官网申请应用时相一致（离线功能需要）。 demo的包名是"com.baidu.tts.sample"。



DEMO压缩包说明

DEMO压缩包下载即可运行，其中DEMO内已经附带了SDK的库。

tts_1.0.3-9c93a23-20260727.aar 位于 app/libs 目录下。

6. 授权文件、离线资源文件

请选择激活方式(离线、iamKey、accessToken)，并将百度云控制台创建应用时获取的语音(APPID) SN、accessToken或iamKey填写到项目，并填写包名。

离线资源文件--发音人（支持16个不同发音人）

离线合成SDK默认自带4个普通音库资源文件，6.2.7及之前版本SDK暂不支持兼容当前版本资源文件，需配合旧版本资源文件使用。旧版本音库资源下载请前往离线发音人列表。

SDK默认自带离线资源文件

资源文件	具体文件名（支持新序列号）
m15 离线男声（度小宇）	bd_etts_common_speech_duxiaoyu_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat
f7 离线女声（度小美）	bd_etts_common_speech_duxiaomei_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat
yy 离线度逍遥	bd_etts_common_speech_duxiaoyao_mand_eng_high_am-tac-csubgan16k_v4.9.0_20251110_20251210203318.dat
c1 离线度丫丫	bd_etts_common_speech_duyaya_mand_eng_high_am-tac-csubgan16k_v4.9.0_20220419_20251031153737.dat
中文离线文本模型	bd_etts_common_text_txt_all_mand_eng_middle_big_v6.0.0_20240731.dat

在线时支持11种发音

普通音库：普通女声 普通男声 情感男声<度逍遥> 情感儿童声<度丫丫>

精品音库： 度逍遥-磁性男声 度博文-情感男声 度小童-活泼男童 度小鹿-甜美女声 度小娇-情感女声 度米朵-可爱女童 度小萌-可爱女童

具体效果可以在在线合成体验窗口上测试。

7. 语音合成相关接口

7.1 初始化初接口

获取 AipeSpeechSynthesizer实例
AipeSpeechSynthesizer mSpeechSynthesizer = new AipeSpeechSynthesizer(context)

设置合成结果的回调

如合成成功后，SDK会调用用户设置的SpeechSynthesizerListener 里的回调方法

```
mSpeechSynthesizer.setSpeechSynthesizerListener(listener); //listener是SpeechSynthesizerListener 的实现类，需要实现您自己的业务逻辑。SDK合成后会对这个类的方法进行回调。
```

设置 激活参数

在语音官网或者百度云网站上申请语音合成的应用后，会有appId、appKey、appSecret及android包名 4个鉴权信息，可选择离线激活、在线激活、正式iamKey激活和临时tocken激活4种激活方式

```
 /* 离线激活方式 */
// 设置离线需要的app_id,在开放平台注册app时获取
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_APP_ID, "填入您的appid");
// 设置离线需要的认证参数，产品序列号SN
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_AUTH_SN, "输入您的sn");
```

```
 /* 临时token激活方法 */
// 设置在线需要的临时token callback
speechSynthesizer.setParam(new TokenCallback() {
   @Override
   public TemporaryToken getToken() {
      .../*通过网络请求获取临时accessTocken或临时iamKey*/
      return "临时accessTocken或临时iamKey";
   }
});
```

```
 /* 永久iam key激活方式 */
// 设置在线需要的正式iam key
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ACCESS_TOKEN, "输入您的永久iam key");
```

如果需要使用离线合成功能的话，请在申请的语音合成的应用填写您自己的包名，若使用demo的包名是“com.baidu.tts.sample"， 定义在build.gradle中。

合成参数

在SpeechSynthesizer类中setParam 方法中使用的参数及值。 填入的值如果不在范围内，相当于没有填写使用默认值。

参数名	类型，值	在线/离线生效	常用程度	解释
PARAM_SPEAKER(基础发音人)	选项	在线	常用	仅在线生效，在线的发音
~	"0"（默认）	~	~	度小美（普通女声）
~	"1"	~	~	度小宇（成熟男声）
~	"3"	~	~	度逍遥（磁性男声）
~	"4"	~	~	度丫丫（可爱女童）
PARAM_SPEAKER （精品发音人）	"106"	~	~	度博文（情感男声）
~	"110"	~	~	度小童（情感儿童声）
~	"111"	~	~	度小萌（情感女声）
~	"103"	~	~	度米朵（情感儿童声）
~	"5"	~	~	度小娇（情感女声）
~	"5003"	~	~	精品度逍遥（磁性男声）
~	"5118"	~	~	度小鹿（甜美女声）
PARAM_VOLUME	String, 默认"5"	全部	常用	在线及离线合成的音量 。范围["0" - "15"], 不支持小数。 "0" 最轻，"15" 最响（取值为0时为音量最小值，并非为无声）
PARAM_SPEED	String, 默认"5"	全部	常用	在线及离线合成的语速 。范围["0" - "15"], 不支持小数。 "0" 最慢，"15" 最快（如需更高语速可提交需求工单或联系商务同学申请）
PARAM_PITCH	String, 默认"5"	全部	常用	在线及离线合成的语调 。范围["0" - "15"], 不支持小数。 "0" 最低沉，"15" 最尖
PARAM_MIX_	选项	全部	常用	控制何种网络状况切换到离线。设置

MODE	选项	全部	常用	initTts(SpeechSynthesizer.PARAM_MIX_MODE)后，该参数生效。
~	MIX_MODE_DEFAULT（默认）	~	~	WIFI 使用在线合成，非WIFI使用离线合成
~	MIX_MODE_HIGH_SPEED_NETWORK	~	~	WIFI,5G,4G 使用在线合成，其他使用离线合成
~	MIX_MODE_HIGH_SPEED_SYNTHESIZE	~	~	同MIX_MODE_HIGH_SPEED_NETWORK。但是连接百度服务器超时1.2s后，自动切换离线合成引擎
~	MIX_MODE_HIGH_SPEED_SYNTHESIZE_WIFI	~	~	同 MIX_MODE_DEFAULT。 但是连接百度服务器超时1.2s后，自动切换离线合成引擎
PARAM_MIX_MODE_TIMEOUT (MIX模式下生效)	选项	离在线混合模式	不常用	离在线模式，强制在线优先。在线请求后超时xx秒后，转为离线合成。
PARAM_TTS_TEXT_MODEL_FILE	String，文件路径	离线	常用	文本模型文件路径，即bd_etts_text.dat所在的路径。
PARAM_TTS_SPEECH_MODEL_FILE	String，文件路径	离线	常用	声学模型文件路径。即"bd_etts_speech_female.dat“所在的路径。需要男声，请使用bd_etts_speech_male.dat。
PARAM_AUDIO_ENCODE	选项	在线	基本不用	不使用改参数即可。SDK与服务器音频传输格式，与 PARAM*AUDIO_RATE参数一起使用。可选值为SpeechSynthesizer.AUDIO_ENCODE**， 其中SpeechSynthesizer.AUDIO_ENCODE_PCM为不压缩
PARAM_AUDIO_RATE	选项	在线	基本不用	不使用改参数即可。SDK与服务器音频传输格式，与 PARAM*AUDIO_ENCODE参数一起使用。可选值为SpeechSynthesizer.AUDIO_BITRATE**，其中SpeechSynthesizer.AUDIO_BITRATE_PCM 为不压缩传输
PARAM_TTS_LICENCE_FILE	String，文件路径	离线	基本不用	临时授权文件。目前SDK会自动下载正式授权文件。

初始化合成引擎

设置合成的参数后，需要调用此方法初始化。

- 在线激活初始化方法如下

```
// 初始化在线tts服务，服务会读取相应资源进行加载，此过程是耗时操作
ITtsError ttsError = speechSynthesizer.loadOnlineTts();
```

- 离线激活初始化方法如下

```
// 初始化离线tts服务，服务会读取相应离线资源进行加载，此过程是耗时操作
ITtsError ttsError = speechSynthesizer.loadOfflineTts();
```

控制接口

合成及播放接口

如果需要合成后立即播放的请调用speak方法，如果只需要合成请调用synthesize方法。

该接口线程安全，可以快速多次调用。内部采用排队策略，调用后将自动加入队列，SDK会按照队列的顺序进行合成及播放。注意需要合成的每个文本text不超过120的GBK字节，即60个汉字或英文字母数字。超过请自行按照句号问号等标点切分，调用多次合成接口。

返回结果不为0，表示出错。错误码请参见“错误码及解决方法”一节

speak方法示例：

```
 ITtsError speak(TtsEntity ttsEntity)// utteranceId在SpeechSynthesizerListener 相关事件方法中回调

TtsEntity ttsEntity = new TtsEntity(text, TtsMode.MIX);
ITtsError ttsError = speechSynthesizer.speak(ttsEntity);
```

synthesize方法示例：

```
 ITtsError synthesize(TtsEntity ttsEntity) // utteranceId在SpeechSynthesizerListener 相关事件方法中回调

ITtsError ttsError = speechSynthesizer.synthesize(new TtsEntity(text, TtsMode.MIX));
```

调用这两个方法后，SDK会回调SpeechSynthesizerListener中的onSynthesizeResponse方法。 音频数据在SynthesizerResponse参数的getAudioData方法中格式为byte[]，采样率16K 16bits编码 单声道。连续将audioData写入一个文件，即可作为一个可以播放的pcm文件（采样率16K 16bits编码 单声道）。

批量合成并播放接口

推荐连续调用speak方法，sdk内部有队列缓冲。 可以循环Speak方法传入多个文本并进行排队合成并播放。 注意需要合成的每个文本text不超过120的GBK字节，即60个汉字或英文字母数字。默认限制队列最多支持合成5条，如需增加可通过PARAM_MAX_QUEUE_SIZE修改。 以下为批量调用示例：

```
 String[] a = {
            "123456",
            "你好",
            "使用百度语音合成SDK",
            "hello",
            "hello",
            "这是一个demo工程"};
for (int i = 0 ; i < a.length ;i++) {
    TtsEntity ttsEntity = new TtsEntity(a[i], TtsMode.MIX);
    ITtsError ttsError = mSpeechSynthesizer.speak(ttsEntity);
    LoggerProxy.d(TAG, "sn = " + ttsEntity.getSn() + " , synthesize = " + ttsError.getDetailCode());
}
```

ITtsError.getDetailCode()返回结果不为0，表示出错。错误码请参见“错误码及解决方法”一节。

批量合成切换发音人

sdk批量合成支持每个文本使用不同发音人播报的功能，将每个文本的发音人信息放到map中传递给TTSEntity进行播报，以下为调用示例：

```
String[] a = {
        "123456",
        "你好",
};
for (int i = 0 ; i < a.length ;i++) {
    HashMap<String, String> ttsParams = new HashMap<>();
    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put(SpeechSynthesizer.PARAM_TTS_TEXT_MODEL_FILE,"文本资源路径");
        jsonObject.put(SpeechSynthesizer.PARAM_TTS_SPEECH_MODEL_FILE,"发音人路径");
        ttsParams.put(SpeechSynthesizer.PARAM_OFFLINE_MODEL, jsonObject.toString());
        TtsEntity ttsEntity = new TtsEntity(a[i], TtsMode.MIX , ttsParams);
        ITtsError ttsError = mSpeechSynthesizer.speak(ttsEntity);
    } catch (JSONException e) {
        throw new RuntimeException(e);
    }
}
```

播放过程中的暂停及继续

仅speak方法调用后有效。可以使用pause暂停当前的播放。pause暂停后，可使用resume进行播放。

```
int result = mSpeechSynthesizer.pause();
int result = mSpeechSynthesizer.resume();
```

返回结果不为0，表示出错。错误码请参见“错误码及解决方法”一节

停止合成并停止播放

取消当前的合成。并停止播放。

```
int result = mSpeechSynthesizer.stop();
```

返回结果不为0，表示出错。错误码请参见“错误码及解决方法”一节

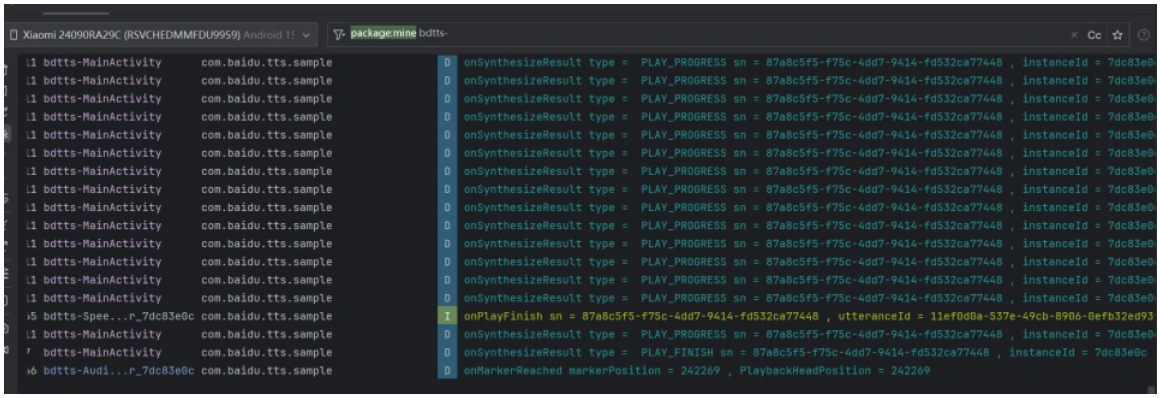
其它接口

打开调试日志（重要）

```
LoggerProxy.printable(true); // 日志打印在logcat中
LoggerProxy.saveNativeLogFile("PRINTF:0"); // 设置日志级别 0：debug级别 ；1：trace级别；2：fatal级别。
```

开启成功后会看见bdtts-开头的tag日志，**建议上线后完全没问题再由服务端控制关闭。**

日志输出截图：



判断模型文件是否有效(重要)

```
SynthesizerTool.verifyModelFile(“/path/to/bd_etts_eng_common_text_all_xxxx.dat”) ;//判断文本资源
SynthesizerTool.verifyModelFile(“/path/to/bd_etts_common_speech_xxxx.dat”) ;//判断音库资源
```

在线合成音色采样率设置

在线合成不同发音人返回的采样率可能会存在差异，可通过PARAM_AUDIO_CTRL设置需要的采样率，建议使用16000和24000两种

```
speechSynthesizer.setParam(SpeechSynthesizer.PARAM_AUDIO_CTRL, "{\"sampling_rate\":16000}");
```

释放资源

不再使用后，请释放资源，并将mSpeechSynthesizer设为null。如果需要再次使用，可以通过SpeechSynthesizer.getInstance()获取，并重复上述流程。

```
int result = mSpeechSynthesizer.release();
```

返回结果不为0，表示出错。错误码请参见“错误码及解决方法”一节

切换离线发音

切换离线发音人接口。 SDK默认只有4种离线，用这个方法可以切换离线发音人。 离线合成时的参数，填入两个资源文件的路径。如果切换的话，也是使用这两个文件路径。

注意：必须在引擎空闲的时候调用这个方法，否则有不为0的错误码返回。空闲是指最后一个合成回调onSynthesizeFinish 之后。

```
try {
  JSONObject jsonObject = new JSONObject();
  jsonObject.put(SpeechSynthesizer.PARAM_TTS_TEXT_MODEL_FILE,"文本资源路径");
  jsonObject.put(SpeechSynthesizer.PARAM_TTS_SPEECH_MODEL_FILE,"音库资源路径");
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_OFFLINE_MODEL, jsonObject.toString());
} catch (JSONException e) {
  e.printStackTrace();
}
```

SpeechSynthesizerListener回调方法

```
public void onSynthesizeResponse(SynthesizerResponse synthesizerResponse)
```

****SynthesizerResponse 输出参数****

合成回调事件统一由 public void onSynthesizeResponse(SynthesizerResponse synthesizerResponse) 该方法回调。其中SynthesizerResponse对象包含合成进度 、音频数据、合成状态等，方法以及说明如下：

方法名	类型	值	描述
getSynthesizeType	enum	SYNTHESIZE_START	合成开始时的回调接口
		SYNTHESIZE_DATA_ARRIVED	合成过程中的回调接口
		SYNTHESIZE_FINISH	合成正常结束时的回调接口
		SYNTHESIZE_ERROR	合成出错时的回调接口
		PLAY_START	合成播报开始时的回调接口
		PLAY_PROGRESS	合成播报过程中的回调接口
		PLAY_FINISH	合成播报正常结束时的回调接口
		ON_NEXT	篇章合成播报填充数据的回调接口
getUtteranceId	String		获取合成标识 id
getInstanceId	String		获取当前实例 id
getAudioProgress	int		获取合成进度，在合成过程中会返回该结果
getEngineType	int	0 在线，1 离线	获取合成类型。0 在线，1 离线
getAudioSampleRate	int		当前音频采样率
getAudioData	byte[]		当前音频数据
getSynthesizerError	String		获取相关的错误信息

****设备唯一标识Cuid获取方法****

sdk激活时需记录设备唯一标识(cuid)，可通过SynthesizerTool.getCuid()方法获取当前设备的cuid

```
Log.e("get_cuid"，SynthesizerTool.getCuid());
```

8. 错误码及解决方法

生成错误码共2处位置：

- 调用接口的方法时的返回，如initTTs方法的返回
- onError(String utteranceId，SpeechError error); SpeechError 中的code

错误码值	错误码描述
-1	在线引擎授权失败
-4	在线授权中断异常
-5	在线授权执行时异常
-6	在线授权时间超时
-7	在线合成返回错误信息
-10	在线引擎合成时异常
-11	当前 mode 不支持的操作
-12	在线合成请求解析出错
-15	在线合成获取合成结果超时
-16	在线授权被取消
-18	在线合成无效的主机名
-19	在线合成读数据失败

-20	在线合成连接失败
-21	在线合成 socket 异常
-24	在线合成请求主机名为空
-25	在线合成发送数据失败
-29	在线合成接收前缀数据长度错误
-30	在线合成接收数据长度错误
-31	在线合成合成数据包速度过快
-32	在线合成网络未知类型错误
-39	在线服务临时错误
-100	离线引擎授权失败
-102	离线授权下载License失败
-105	离线授权中断异常
-106	离线授权执行时异常
-107	离线授权执行时间超时
-108	离线合成引擎初始化失败
-110	离线合成时异常
-111	离线合成返回值非0
-118	鉴权被取消
-122	离线 tts_offline_resource 文件异常
-123	离线发音人参数异常
-124	下载 license 失败，序列号sn 参数异常
-125	离线合成文本为空
-200	混合引擎离线在线都授权失败
-204	混合引擎初始化 tts 时，离线初始化失败
-206	混合引擎初始化 tts 时，在线初始化失败
-300	合成文本为空
-301	合成文本长度过长（不要超过 GBK1024 个字节）
-302	合成文本无法获取GBK字节
-401	TTS模式无效
-402	TTS合成队列已满（最大限度为1000）
-406	TTS被调用方法参数无效
-500	Context被释放或为空
-700	播报的短音频文件不存在
-701	当前接口不支持播报短音频
-800	当前实例已经被释放，禁止调用该实例的任何API
-1001	模型管理请求出错
-1002	模型管理服务器端错误
-1003	模型管理数据库模型信息无效
-1004	模型管理数据库模型文件信息无效
-1005	模型数据已经存在（ 或已下载）

-1006	无法获取到模型信息
-1007	无法获取到模型文件信息
-1008	模型检查过程异常
-1009	模型文件下载时异常
-9999	未知错误

常见错误码及解决方案

错误码	含义	可能原因	自查指南	解决办法
-102	离线授权下载license失败	1.网络不佳 2.授权码额度耗尽 3.SN已经绑定其他设备 导致下载license失败	1. 检查离线联网授权时的网络环境； 2.确认SN序列号是否还有额度 3.确认SN之前是否绑定过其他设备（刷机等更改设备信息的操作都是导致下载license失败）	1. 更换稳定的网络环境； 2. 补充SN序列号额度(产品线授权方式) 3. 如绑定过其他设备更换新的授权SN(设备数授权方式)
-108	离线合成引擎初始化失败	loadmodel的资源文件 离线音库文件 没加载	1.确认音库文件是否下载；2.是否在指定位置加载；3. 判断模型文件是否有效点击	1.如加载位置没有对应的文本和音库文件，手动复制文件到指定位置；2.可以尝试删除应用，重新安装测试
	同上	loadmodel时合成引擎不空闲	sdk只支持同一语种的不同发音人使用loadModel方法切换，例如，度小美切换度丫丫。如果是中文切英文或者粤语，需要反初始化后再初始化	如需要切换不同的文本资源，请先调用release方法，再次执行新的文本资源初始化
	同上	其他	初始化添加LoggerProxy.printable(true)；保存启动app到报错的完整日志	提供日志，百度侧进一步判断
-119	SN序列号不合法	SN序列号不在有效期内	核对绑定的SN序列号	更换合法SN序列号
	同上	SN序列号已被绑定	需要确认SN序列号是否已经绑定其他设备，或者设备刷机等导致cuid变更操作	反馈给百度侧，进一步判断
	鉴权未通过	鉴权信息错误	需要确认SN序列号外的其他鉴权信息	填写正确的鉴权信息，注意空格
-204	离线合成初始化失败	设备的系统时间超出license时效	确认设备的系统时间是否正常	校验系统时间
	同上	license 超出有效期	确认SN序列号在有效期内	更换有效期内授权SN

9. 代码混淆

```
-keep class com.baidu.tts.*{*;}
-keep class com.baidu.speechsynthesizer.*{*;}
```

10. 权限

名称	说明	必选
必要的权限		
android.permission.INTERNET	允许访问网络	是
android.permission.ACCESS_NETWORK_STATE	获取网络状态权限	是

11. 不使用离线合成，只使用在线合成

删除离线音库文件，并且设置合成模式为online在线模式 ，speechSynthesizer.speak(new TtsEntity(text, TtsMode.ONLINE));

```
TtsEntity ttsEntity = new TtsEntity(text, TtsMode.ONLINE);
ITtsError ttsError = mSpeechSynthesizer.speak(
        ttsEntity);
```

语音合成 iOS SDK

1. 文档说明

文档名称	语音合成集成文档
所属平台	iOS
提交日期	2026-08-11
概述	本文档是百度离线语音合成iOS SDK的用户指南，描述了离线语音合成SDK相关接口的使用说明。 离线语音合成SDK包含基础音库，在线语音合成SDK无需音库资源即可调用，默认支持流式合成，请您前往语音合成SDK资源按需下载；若您在2025年11月27日及之前购买或申请测试SN（序列号），请点击下载往期基础音库资源，及往期精品音库资源。详细发音人信息请看离线发音人列表。

2. 版本说明

名称	版本号
语音合成	4.3.0
系统支持	支持iOS 12.0及以上
架构支持	支持x86_64、arm64 (离线合暂成不支持x86_64架构)
机型	iPhone、iPad系列
IDE	Xcode 14+（建议使用Xcode16及以上版本）

2.1 版本升级改动点说明：

- 提升SDK稳定性；
- 离线发音人听感调优
- SDK引擎升级优化
- AKSK鉴权方式不再使用，更新优化部分API接口
- 新增Token（临时、永久）、IAMKEY（临时、永久）鉴权 方式

3. SDK说明

文件名称	版本号	说明	类型
libBaiduSpeechSDK.a	4.3.0	语音合成SDK （同时支持离在线引擎）大小约173M	静态库

1. 由于 BITCODE 开启会导致二进制文件体积增大，这部分会在 APPStore 发布时进行进一步编译优化，并不会引起最终文件的体积变化，故此处计算的是关闭 BITCODE 下的二进制增量。
2. .a中是多个架构做了合并，使用lipo可以看到细节。所以.a库文件本身很大，且打包出来的ipa也相对较大。但用户实际下载到手机中会被AppStore优化，只下载用户设备需要的架构，所以实际在手机上占用的空间很小。

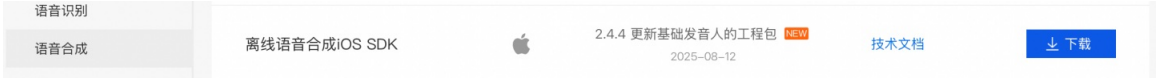
```
Architectures in the fat file: libBDSpeechTTSBaseKit.a are: x86_64 arm64
```

开发包说明

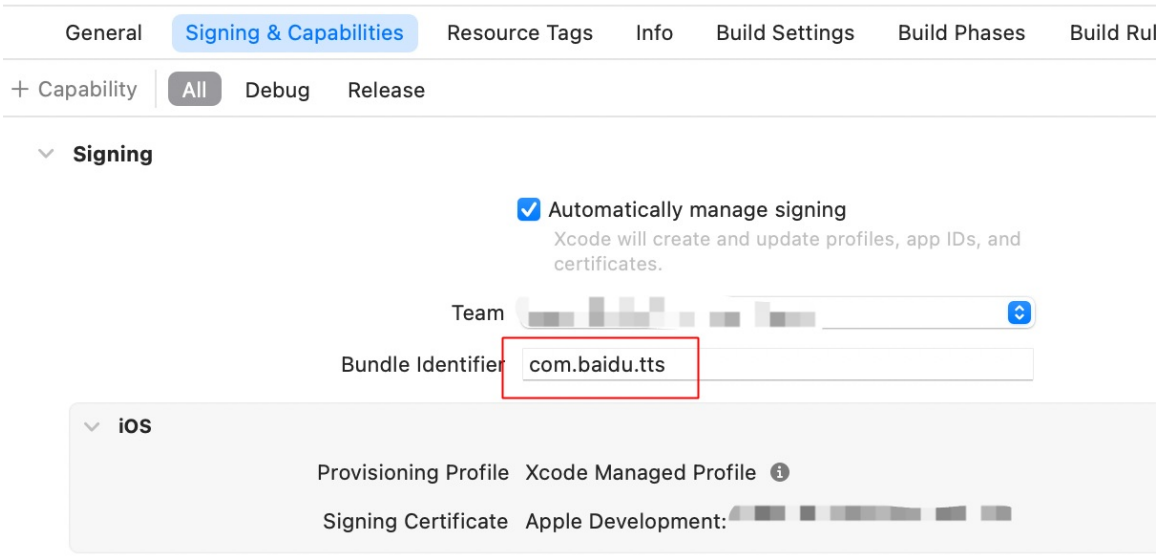
文件（夹）名	说明	备注
Resources	文件夹下包含离线合成头文件、以及离线合成.a库	必须引入
BDSClientSample	开发示例(xcode project)	
BDSClientResource/ TTS/Chinese *And* English_Speech*.dat	具体参照 [纯离线语音合成模式] 章节	若无需离线合成则无需引入 **注意**：路径不能使用中文
BDSClientResource/ TTS/Chinese *And* English_Text.dat	离线语音合成资源文件(text data file)， 中英文	若无需离线合成则无需引入 **注意**：路径不能使用中文
BDSClientResource/ TTS/English_Speech _Female.dat, BDSClientResource/ TTS/English_Speech _Male.dat	离线语音合成资源文件 (speech data file，英文)，女声，男声	若无需离线合成则无需引入 **注意**：路径不能使用中文

4. Demo运行

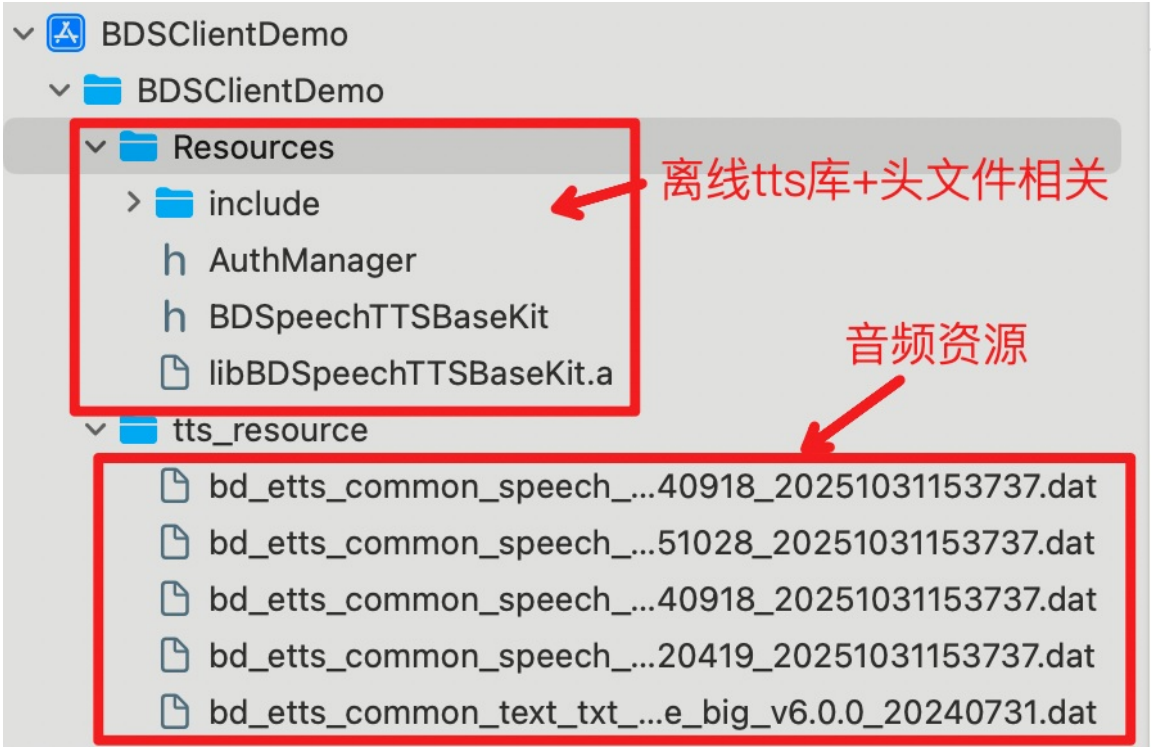
1. 从百度云控制台或开放平台离线SDK下载Demo



2. 配置包名和签名



3. 打开工程目录如下:

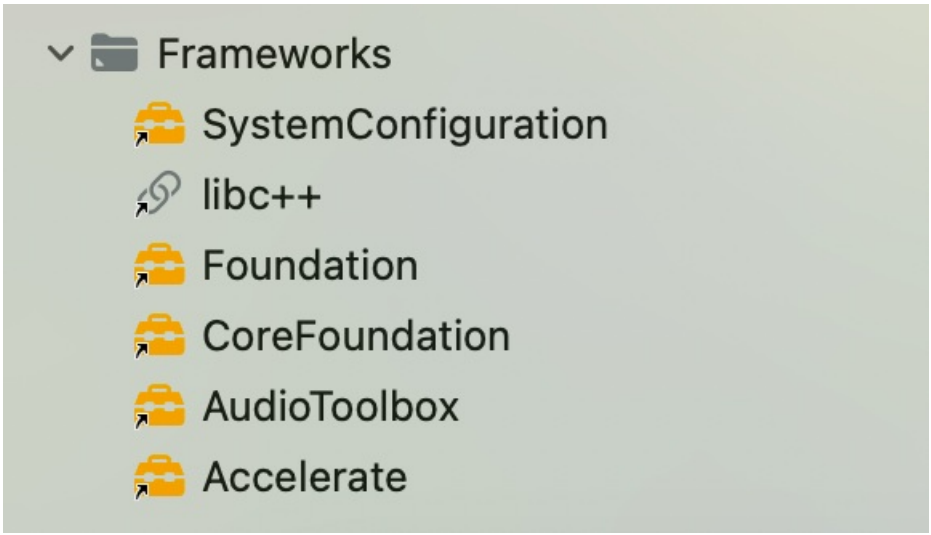


5. SDK集成

强烈建议用户首先运行SDK包中的Demo工程，Demo工程中详细说明了语音合成的使用方法，并提供了完整的示例。一般情况下，您只需参照demo工程即可完成所有的集成和配置工作。

添加BDSSpeechSynthesizer到工程

BDSSpeechSynthesizer使用了一些系统的framework，需要添加到工程里面。



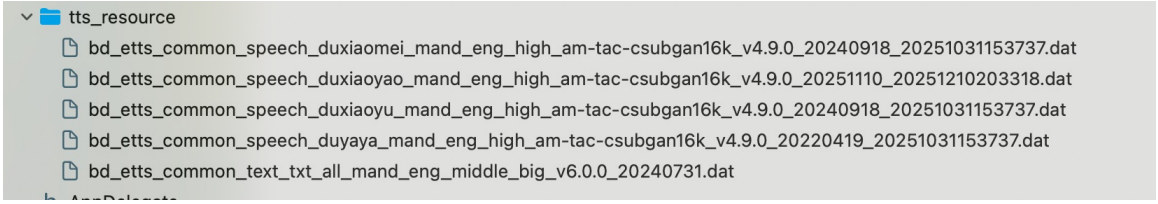
添加方式: 右键点击 Xcode 中的工程文件,在出现的界面中,选中 TARGETS 下的应用,在出现的界面中选中 Build Phases->Link Binary With Libraries,点击界面中的“+”图标,在弹出的界面中选择需要的framework即可。请参考demo工程引入所需动态库framework和静态库。

> 开发者编译自己工程时，若报错 ld: Library xxxx not found，一般是库文件没有正确引入。请确保库文件在工程所能找到的位置之内(通过 LIBRARRY_SEARCH_PATH设置)

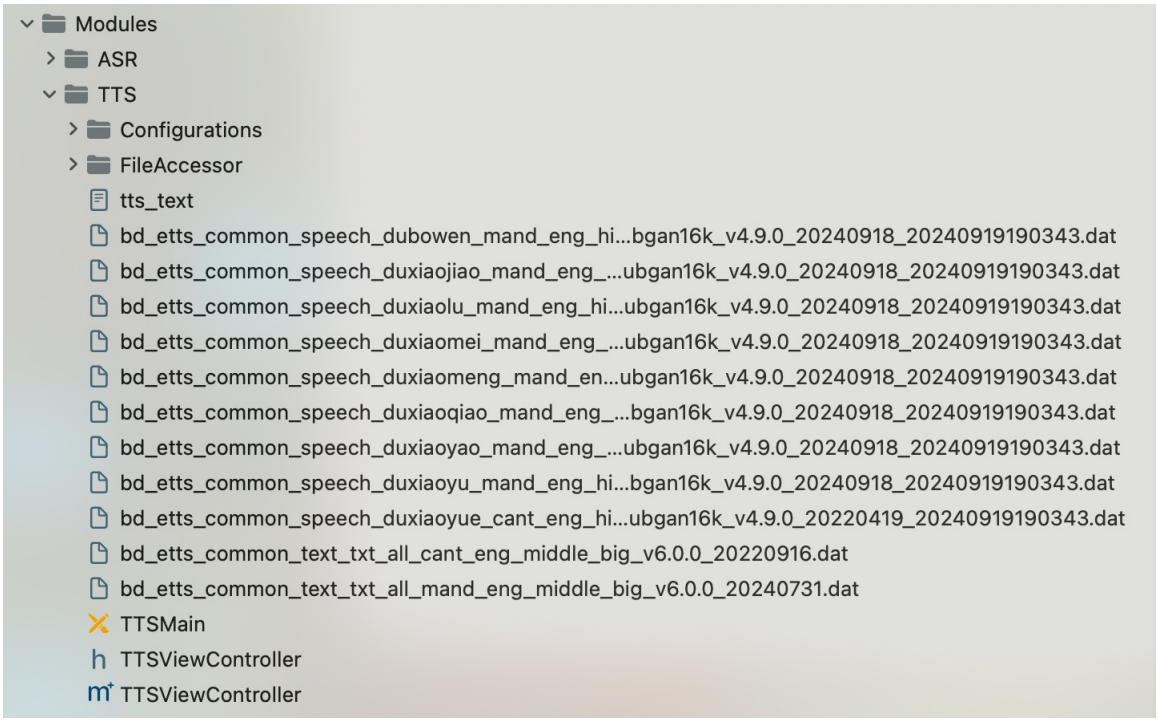
添加语音合成相关资源文件

将开发包中的BDSClientResource目录下的相关资源文件添加到工程或者安装app后部署到指定目录(代码中启动合成引擎时需要指定该资源文件的访问路径)。根据具体需求引入不同的资源即可。

基础音库



精品音库



音频文件配置代码示例：

```
 // 创建语音合成器实例
   self.synthesizer = [[BDSSpeechSynthesizer alloc] initWithDelegate:self];
   // 设置离线参数资源文件
   NSString *textData = [[NSBundle mainBundle]
pathForResource:@"bd_etts_common_text_txt_all_mand_eng_middle_big_v6.0.0_20240731" ofType:@"dat"];
   NSString *speechData = [[NSBundle mainBundle]
pathForResource:@"bd_etts_common_speech_duxiaoyao_mand_eng_high_am-tac-
csubgan16k_v4.9.0_20251028_20251031153737" ofType:@"dat"];

   // 正确的离线模型配置
   NSDictionary *offlineModel = @{
         BDSSpeechSynthesizerParams.ttsTextModelFile: textData,
         BDSSpeechSynthesizerParams.ttsSpeechModelFile: speechData
      };
   NSData *offlineModeData = [NSJSONSerialization dataWithJSONObject:offlineModel options:0 error:nil];
   NSString *offlineModelString = [[NSString alloc] initWithData:offlineModeData encoding:NSUTF8StringEncoding];
   [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramOfflineModel value:offlineModelString];
```

引入BDSSpeechSynthesizer相关必要头文件

首先将 BDSSpeechSynthesizer 提供的头文件拷贝到工程目录下,在 XCode 中添加此文件,引入 BDSSpeechSynthesizer提供的头文件。

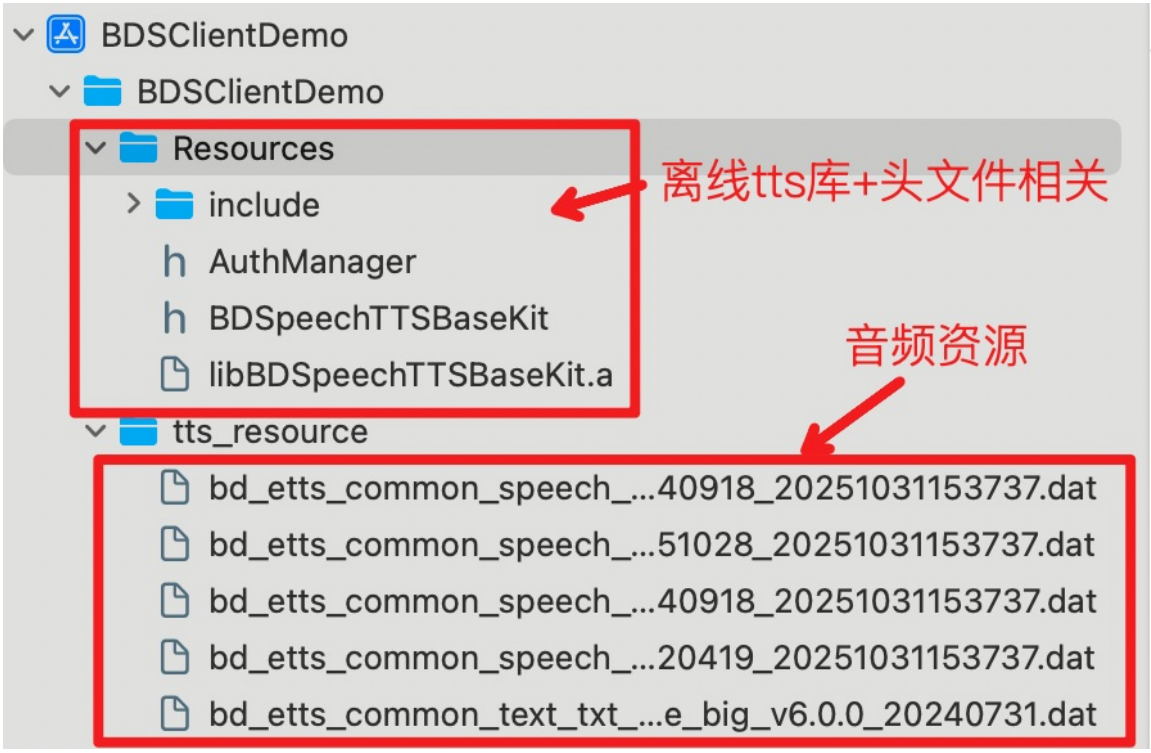
添加如下头文件：

```
 ##### import "TTSViewController.h"
##### import "AuthManager.h"
##### import <AVFAudio/AVAudioSession.h>

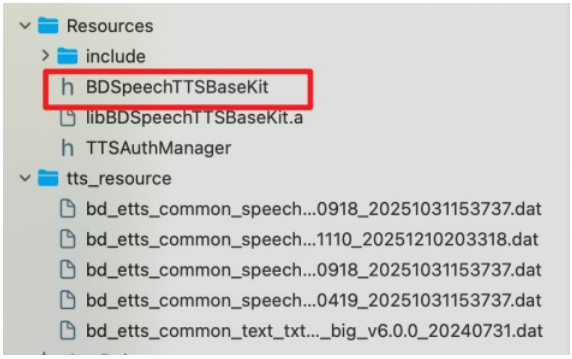
##### import "BDSpeechTTSBaseKit.h"
##### import "BDSSpeechSynthesizerResponse.h"
##### import "BDSSpeechLogger.h"
##### import "BDSSpeechSynthesizerParams.h"
##### import "Reachability.h"
```

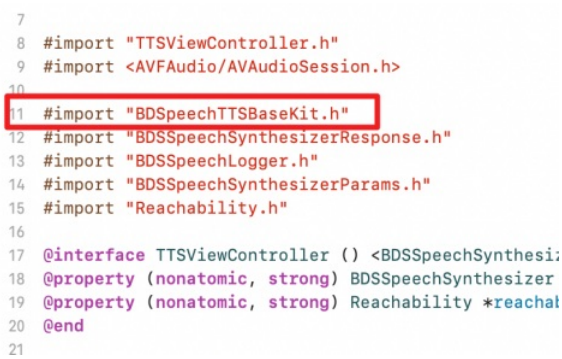
引入静态库文件

静态库打开如下：



引入以下三个头文件：





6.离在线模式鉴权参数配置说明

离在线语音合成模式

请从官网控制台获取您的APP_ID、包名、SN等参数。并绑定你的移动应用的BundleId。分别加载在线和离线引擎。

在线合成和离线合成需要进行相关验证后方可使用：

引擎类型	验证方法
在线合成	开放平台使用ACCESSTOKEN进行验证（新增），详情请点击默认鉴权机制了解 开放平台使用IAMKEY进行验证（新增），详情请点击API Key鉴权机制了解
离线合成	使用APPID+授权序列号SN+包名首次联网自动下载授权文件进行验证 离线合成需要另外填写申请授权序列号，具体流程参考 快速集成指南

纯在线语音合成模式

使用IAMKEY（当前版本新增）、TOKEN（当前版本新增）鉴权方式使用示例：

```
 - (void)initOnlineSDK {
    // 创建语音合成器实例
    self.synthesizer = [[BDSSpeechSynthesizer alloc] initWithDelegate:self];
    // 申请获取token鉴权
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAccessToken value:@"平台申请获取的token"];
    // 申请获取IAMKEY鉴权
    //  [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAccessToken value:@"平台申请获取的IAMKEY"];
    // 设置在线发音人
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramOnlineSpeaker value:@"4100"];
//     加载在线引擎
    BDSSpeechResult *result = [self.synthesizer loadOnlineEngine];
    if (result.code != 0) {
        NSLog(@"加载在线引擎失败: %ld", (long)result.code);
    }
}
```

鉴权接口及参数说明

```
 /**
 token鉴权方式
 token：申请的token
 expirationDate：token过期时间戳（毫秒）
 */
- (void)setTTSLicenseWithAuthToken:(NSString *)token AndExpirationDate:(long long)expirationDate;

/**
 iamkey方式初始化鉴权
 iamkey：申请的iamkey
 expirationDate：iamkey过期时间（毫秒）
 */
- (void)setTTSLicenseWithIAMKey:(NSString *)iamkey AndExpirationDate:(long long)expirationDate;
```

完整示例参考：

```
 // 初始化离线合成SDK
[self initOfflineSDK];

- (void)initOfflineSDK {
    // 初始化日志，仅 DEBUG 调试打开此设置
    [[BDSSpeechLogger sharedInstance] setLogLevel:BDSSpeechLogLevelInfo path:nil callback:nil];

    // 创建语音合成器实例
    self.synthesizer = [[BDSSpeechSynthesizer alloc] initWithDelegate:self];
    // 设置离线参数资源文件
    NSString *textData = [[NSBundle mainBundle]
pathForResource:@"bd_etts_common_text_txt_all_mand_eng_middle_big_v6.0.0_20240731" ofType:@"dat"];
    NSString *speechData = [[NSBundle mainBundle]
pathForResource:@"bd_etts_common_speech_duxiaoyao_mand_eng_high_am-tac-
csubgan16k_v4.9.0_20251028_20251031153737" ofType:@"dat"];

    // 正确的离线模型配置
    NSDictionary *offlineModel = @{
            BDSSpeechSynthesizerParams.ttsTextModelFile: textData,
            BDSSpeechSynthesizerParams.ttsSpeechModelFile: speechData
        };
    NSData *offlineModeData = [NSJSONSerialization dataWithJSONObject:offlineModel options:0 error:nil];
    NSString *offlineModelString = [[NSString alloc] initWithData:offlineModeData encoding:NSUTF8StringEncoding];
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramOfflineModel value:offlineModelString];

    // 设置离线授权文件
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAppId value:@"平台申请获取的APPID"];
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAuthSerialNumber value:@"平台申请获取的SN"];

    // 加载离线引擎
    BDSSpeechResult *result = [self.synthesizer loadOfflineEngine];
    result = [self.synthesizer loadOfflineEngine];
    if (result.code != 0) {
        NSLog(@"加载离线引擎失败: %ld", (long)result.code);
    }
}
```

提供用户自定义获取鉴权信息的能力

描述：为了满足 「鉴权信息保存在server端，app通过请求server端获取access token」的需求，语音SDK需要提供 鉴权信息回调点，并由用户最终实现 回调接口。

```
 - (void)initCommonSDK {
    //主要通过函数实现 参考demo对应部分
}

步骤一：
// 遵循鉴权sdk内置协议，初始化设置协议
[AuthManager sharedInstance].temporaryTokenAndKeyDelegate = self;

// 分别实现协议传入token、iamkey
/**
 客户实现get token 协议方法
 */
- (void)getToken {
    ......
    //客户业务侧处理获取token的逻辑
    // demo示例发送请求
     [self sendPostRequest];
}
```

```
/**
 客户实现get iamkey 协议方法
 */
- (void)getIAMKey {
    ......
    //客户业务侧处理获取iamkey的逻辑
    // demo示例发送请求
     [self sendPostRequest];
}

步骤二：
// 初始化调用在线函数进行SDK模型参数初始化配置
// 在线TTS
 [self initCommonSDK]; // 完成模型初始化参数配置相关操作
//获取网络权限判断逻辑
    ......
    ......
    ......
// 检查当前网络状态并尝试网络请求
[self checkNetworkStatusAndMakeRequest];


  // 检查网络权限是否生效
- (void)checkNetworkStatusAndMakeRequest {
    // 生效 即可做post请求获取token、有效期
    ......
    ......
    ......
    // 调用本地获取token的请求逻辑
     [self sendPostRequest];
}

  // post请求 仅供参考
-(void)sendPostRequest {
 // 请求逻辑具体参考demo代码
  ......
  ......
  请求拿到token后调用API接口完成token鉴权
  // token 鉴权  成功：正常完成合成主流程鉴权通过。 失败：触发上述步骤一中的协议回调 重新获取有效token（此处以token为例）
  [[BDSpeechTTSBaseKit sharedInstance] setTTSLicenseWithAuthToken:@"平台获取的token" AndExpirationDate:"token有效期时间"];
}

逻辑整理 1.遵循鉴权sdk内置协议 2.完成初始化请求token获取[self sendPostRequest]; 并在请求结果中设置对应的鉴权方式如 [[BDSpeechTTSBaseKit sharedInstance] setTTSLicenseWithAuthToken:@"平台获取的token" AndExpirationDate:"token有效期时间"];获取token。3.成功后正常继续下一步，异常或请求token失败、token有效期到期等SDK代理会监听到 触发getToken、getIamkey代理 进行重新处理逻辑 demo示例中做了重新获取token的操作 仅供参考

注：使用该方式为了满足 「鉴权信息保存在server端，app通过请求server端获取access token」的需求，语音SDK需要提供 鉴权信息回调点，并由用户最终实现 回调接口。
```

纯离线语音合成模式

请参考`initOfflineSDK`方法配置离线资源文件和鉴权，或参考4.2离线集成配置相关描述。

1. APPID+SN方式鉴权：

参数配置说明 AppID可登录控制台查看，App ID与SN获取方式参考 快速集成指南

暂不支持swift版本，Objective-c版本如下： 离线TTS调用示例

```
- (void)initOfflineSDK {
    // 初始化日志，仅 DEBUG 调试打开此设置
    [[BDSSpeechLogger sharedInstance] setLogLevel:BDSSpeechLogLevelInfo path:nil callback:nil];

    // 创建语音合成器实例
    self.synthesizer = [[BDSSpeechSynthesizer alloc] initWithDelegate:self];
    // 设置离线参数资源文件
    NSString *textData = [[NSBundle mainBundle]
pathForResource:@"bd_etts_common_text_txt_all_mand_eng_middle_big_v6.0.0_20240731" ofType:@"dat"];
    NSString *speechData = [[NSBundle mainBundle]
pathForResource:@"bd_etts_common_speech_duxiaoyao_mand_eng_high_am-tac-
csubgan16k_v4.9.0_20251028_20251031153737" ofType:@"dat"];

    // 正确的离线模型配置
    NSDictionary *offlineModel = @{
          BDSSpeechSynthesizerParams.ttsTextModelFile: textData,
          BDSSpeechSynthesizerParams.ttsSpeechModelFile: speechData
       };
    NSData *offlineModeData = [NSJSONSerialization dataWithJSONObject:offlineModel options:0 error:nil];
    NSString *offlineModelString = [[NSString alloc] initWithData:offlineModeData encoding:NSUTF8StringEncoding];
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramOfflineModel value:offlineModelString];

    // 设置离线授权文件
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAppId value:@"平台申请获取的APPID"];
    [self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAuthSerialNumber value:@"平台申请获取的SN"];

    // 加载离线引擎
    BDSSpeechResult *result = [self.synthesizer loadOfflineEngine];
    result = [self.synthesizer loadOfflineEngine];
    if (result.code != 0) {
       NSLog(@"加载离线引擎失败: %ld", (long)result.code);
    }
}
```

2. 加载离线引擎 离线合成SDK默认自带4个基础音色的『Speech文件』和1个『Text文件』，精品音库资源文件需单独下载。若您在2025年11月27日之前购买或申请序列号，请点击下载往期音库资源文件。发音人详细信息请看离线发音人列表。

SDK默认自带离线资源文件

资源文件	具体文件名（支持新序列号）
m15 离线男声（度小宇）	bd_etts_common_speech_duxiaoyu_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat
f7 离线女声（度小美）	bd_etts_common_speech_duxiaomei_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat
yy 离线度逍遥	bd_etts_common_speech_duxiaoyao_mand_eng_high_am-tac-csubgan16k_v4.9.0_20251110_20251210203318.dat
c1 离线度丫丫	bd_etts_common_speech_duyaya_mand_eng_high_am-tac-csubgan16k_v4.9.0_20220419_20251031153737.dat
中文离线文本模型	bd_etts_common_text_txt_all_mand_eng_middle_big_v6.0.0_20240731.dat

2.4.3之前版本，中文、粤语、英文因为使用不同的Text文件,之间通过reinitOfflineEngineData 无法完成切换。必须反初始化，再初始化。2.4.3之后版本无需重新初始化。

7.语音合成接口说明

7.1 合成接口初始化

用于获取、释放合成器实例

```
// 初始化 - (instancetype)initWithDelegate:(id<BDSSpeechSynthesizerDelegate>)delegate;
// 释放 - (void)releaseEngine;
```

7.2 接口参数配置

```
/**
* @brief 合成器参数设置接口
* @param param 参数对应的值
* @param key 合成器预定义的参数名称，参见附录及头文件BDSSpeechSynthesizerParams.h.
*/ - (BDSSpeechResult *)setParam:(NSString *)key value:(NSString *)value;
```

7.2.1 设置在线合成信息

```
/**
* @brief 为在线合成设置认证信息
* @param AccessToken 在百度开发者中心注册应用获得
*/
[self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAccessToken value:@"xxx"];
```

7.2.2 加载在线引擎

```
// 加载在线引擎
BDSSpeechResult *result = [self.synthesizer loadOnlineEngine];
if (result != 0) {
NSLog(@"加载在线引擎失败: %ld", (long)result.code); }
```

7.3离线合成参数

7.3.1配置离线资源

```
// 设置离线资源 (方式一，适合业务自行管理，云端下发资源文件)
NSString *textData = [[NSBundle mainBundle] pathForResource:@"xxx" ofType:@"dat"];
NSString *speechData = [[NSBundle mainBundle] pathForResource:@"xxx" ofType:@"dat"];
NSDictionary *offlineModel = @{BDSSpeechSynthesizerParams.ttsTextModelFile: @([textData UTF8String]),
BDSSpeechSynthesizerParams.ttsSpeechModelFile: @([speechData UTF8String])};
NSData *offlineModelData = [NSJSONSerialization dataWithJSONObject:offlineModel options:0 error:nil];
NSString *offlineModelString = [[NSString alloc] initWithData:offlineModelData encoding:NSUTF8StringEncoding];
[self.synthesizer setParam:BDSSpeechSynthesizerParams.paramOfflineModel value:offlineModelString];
```

7.3.2设置离线授权

```
// 设置您的鉴权 sn
[self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAuthSerialNumber value:@"your auth sn")];
[self.synthesizer setParam:BDSSpeechSynthesizerParams.paramAppId value:@"xxx"];
```

7.3.3加载离线引擎

```
// 加载离线引擎
result = [self.synthesizer loadOfflineEngine];
if (result.code != 0) {
NSLog(@"加载离线引擎失败: %d", result.code); }
```

7.4语音合成接口

7.4.1合成并朗读

```
 /**
* 合成并播放语音 * @param entity 语音合成实体
* @return 播放结果
*/
- (BDSSpeechResult *)speak:(BDSSpeechEntity *)entity;
```

7.4.2合成不朗读

```
 /**
* 合成语音 * @param entity 语音合成实体
* @return 合成结果
*/
- (BDSSpeechResult *)synthesize:(BDSSpeechEntity *)entity;
```

7.4.3取消合成并停止朗读

```
 /**
* 停止播放/合成
* @return 停止结果
*/
- (BDSSpeechResult *)stop;
```

7.4.4暂停朗读

```
 /**
* 暂停播放
* @return 暂停结果
*/
- (BDSSpeechResult *)pause;
```

7.4.5恢复朗读

```
 /**
* 恢复播放
* @return 恢复结果
*/
- (BDSSpeechResult *)resume;
```

7.4.6合成状态监听器接口

```
 - (void)onSynthesizerResponse:(BDSSpeechSynthesizerResponse *)synthesizerResponse {
switch (response.synthesizeType) {
case BDSSpeechSynthesizerTypeStart:
NSString *sn = synthesizerResponse.sn; // 此次SDK侧传递的文本合成的标识，建议保存，后续排查服务器等问题需要使用。
NSString *utteranceId = synthesizerResponse.utteranceId;// 此次业务侧传递的文本合成的标识，用来做相关逻辑判断。
break; ..... }
```

7.4.7合成模式

合成模式	参数
纯在线	BDSSpeechModeOnline
纯离线	BDSSpeechModeOffline
混合模式，在线优先	BDSSpeechModeMix

```
BDSSpeechEntity *entity = [[BDSSpeechEntity alloc] initWithText:text
                                                           mode:BDSSpeechModeOnline
                                                     utteranceId:[[NSUUID UUID] UUIDString]
                                                         params:nil];
```

7.5播放控制

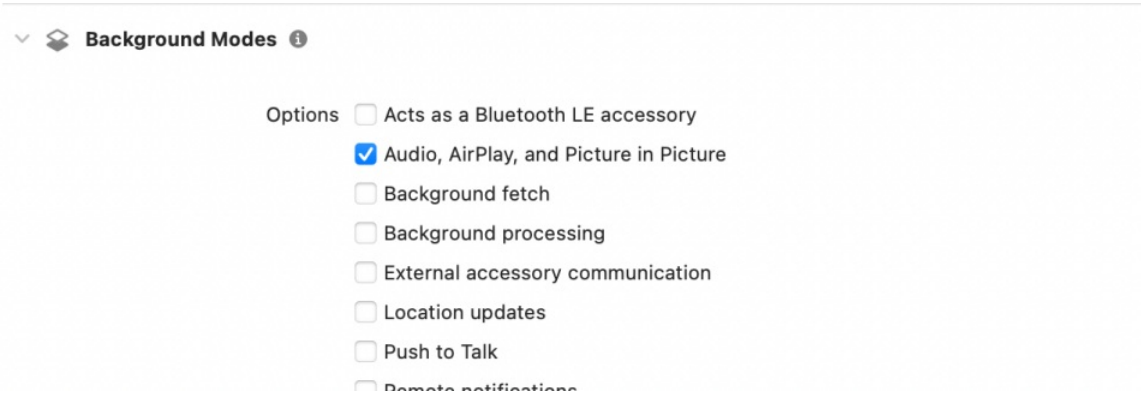
7.5.1设置播放器音量

```
[self.synthesizer setParam:BDSSpeechSynthesizerParams.paramPlayerVolume value:@"0-1"];
```

7.5.2设置AudioSession相关

```
// 业务方根据需求场景自定义设置，此处仅做举例
- (void)setAudioSessionConfig {
NSString *category = AVAudioSessionCategoryPlayback;
NSUInteger categoryOptions = (AVAudioSessionCategoryOptionMixWithOthers|
AVAudioSessionCategoryOptionDuckOthers);
NSError *error = nil; [[AVAudioSession sharedInstance] setCategory:category withOptions:categoryOptions error:&error];
if (error) { [self appendLog:[NSString stringWithFormat:@"设置音频会话失败: %@", error.localizedDescription]]; } }
```

7.5.3配置后台播放



注：配置xcodebackground Modes保证后台播放生效。

8.工具类

8.1设置、获取本地日志级别

```
// 初始化日志，仅 DEBUG 调试打开此设置
[[BDSSpeechLogger sharedInstance] setLogLevel:BDSSpeechLogLevelInfo path:nil];
```

8.2相关工具类

接口	参数
(BDSSpeechResult *)verifyModelFile:(NSString *)filePath	验证模型文件
(NSString *)getEngineInfo	获取引擎信息
(NSString *)getModelInfo:(NSString *)filePath	获取模型信息
(int)matchResEngine:(NSString *)filePath	检查资源版本同引擎版本是否匹配
(long)getSpeechSampleRate:(NSString *)filePath	获取音库资源的采样率
(int)checkDomainFile:(NSString *)filePath	验证个性化语句包的是否可用
(NSString *)getCuid	获取sdk内部当前使用的设备id
(long)getDomainSampleRate:(NSString *)filePath	获取个性化语句包的采样率

8.3查询静态库版本信息

```
 bash
strings -a BDTTSClientSDK | grep TTS_SDK_VERSION // SDK 版本号
strings -a BDTTSClientSDK | grep ETTS_UNIQUE // 引擎 版本号
```

8.4合成参数参考

百度语音合成支持的参数用BDSSynthesizerParamKey类型表示，具体如下：

参数名称	字段标识	说明	取值 / 范围 / 默认值
产品 ID	paramProductId	语音合成产品唯一标识	-
自定义 CUID	paramCustomCuid	设备自定义唯一标识	-
在线超时时间	paramOnlineTimeoutMs	在线请求超时时长，单位毫秒	默认 6000
比特率	paramBitrate	音频编码比特率，对应枚举BDSSynthesizerBitrate	枚举值： 0 = BDSSynthesizerBitratePCM 64 = BDSSynthesizerBitrateOpus64K（默认） 128 = BDSSynthesizerBitrate128K
在线合成服务 URL	paramOnlineServerUrl	在线合成服务器地址（仅合作方配置）	-
Key	paramKey	应用密钥	-
语言	paramLanguage	合成语音语种	-
在线发音人	paramOnlineSpeaker	在线语音音色选择	-
访问令牌	paramAccessToken	接口访问授权令牌	-
App ID	paramAppId	应用唯一 ID	-
转换数据路径	paramConvertDatPath	数据转换文件存储路径	-
统计参数	paramStatPam	数据统计相关配置参数	-

文本位置	paramTextPos	文本播放位置标记	-
百分比	paramPercent	进度百分比参数	-
音频控制	paramAudioCtrl	音频功能控制参数	-
文本控制	paramTextCtrl	文本解析与播报控制参数	-
打开 XML	paramOpenXml	是否启用 XML 标签解析	-
语速	paramSpeed	合成语音播放速度	在线SDK：默认最高支持3倍速播放，倍速区间0-25，SDK透传，计算公式0.5 + 0.1 *X（X为设置的区间值）* *离线SDK：默认最高支持2倍速播放，倍速区间0-15，SDK透传，计算公式0.5 + 0.1* X（X为设置的区间值） 默认区间值：5 正常1倍，速取值范围 [0, 15]，默认 5 注：如现有语速不能满足，需更高语速可提交需求工单或联系商务同学申请
音量	paramVolume	合成语音音量	取值范围 [0, 15]，默认 5
音调	paramPitch	合成语音音调高低	取值范围 [0, 15]，默认 5
特殊参数	paramSpec	拓展自定义特殊配置	-
离线资源路径	paramOfflineResourcePath	离线合成资源文件目录	-
PCM 资源路径	paramPcmResourcePath	PCM 格式音频资源路径	-
离线发音人	paramOfflineSpeaker	离线语音音色（由.conf 配置文件指定）	-
离线模型	paramOfflineModel	离线合成模型标识	-
TTS 授权文件路径	paramTtsLicenseFilePath	语音合成授权证书文件路径	-
授权序列号	paramAuthSerialNumber	离线授权 SN 序列号	-
授权 URL	paramLicenseUrl	授权服务请求地址	-
播放器音量	paramPlayerVolume	系统播放器音量	取值范围 [0, 1]，默认 1
播放器流类型	playerStreamType	音频播放流类型	-
播放器用途	paramPlayerUsage	播放器使用场景标识	-
播放器内容类型	paramPlayerContentType	播放音频内容分类	-
播放器中断模式	paramPlayerInterruptMode	音频抢占 / 中断策略	-
语音合成类型	paramSpeechSynthesizeType	区分在线 / 离线 / 混合合成模式	-
AIGC 最小播放缓存	paramAigcMinPlayCache	AIGC 语音最小播放缓存大小	-

通用参数	paramPam	综合拓展配置参数	-
文本模型文件路径	ttsTextModelFile	TTS 文本模型文件路径	-
声学模型文件路径（tacotron）	ttsSpeechModelFile	主声学模型文件路径	-
扩展声学模型文件路径（subgan）	ttsExtSpeechModelFile	拓展声学模型文件路径	-
声学模型文件属性	ttsTacSubganSpeakerAttr	音色、风格属性配置（JSON 字符串）	格式：{"style_id": 1, "speaker_id": 1}
在线发音人	paramOnlineSpeaker	配置在线发音人列表	发音人 在线基础：0（普通女声） 1（普通男声） 3（磁性男声<度逍遥>） 4（可爱童声<度丫丫>） 在线精品：106（情感男声<度博文>） 110 （活泼童声<度小童>） 111 （可爱童声<度小萌>） 103（可爱童声<度米朵>） 5（情感女声<度小娇>） 5118（甜美女声<度小鹿>） 5003（磁性男声<度逍遥（精品）>） 更多在线发音人可点击 音色列表

9.附录

9.1参数说明

参见 BDSSpeechSynthesizerParams 类

9.2错误码说明

序号	错误码值	错误码描述
1	-1	在线引擎授权失败
2	-4	在线授权中断异常
3	-5	在线授权执行时异常
4	-6	在线授权时间超时
5	-7	在线合成返回错误信息
6	-10	在线引擎合成时异常
7	-11	当前 mode 不支持的操作
8	-12	在线合成请求解析出错
9	-15	在线合成获取合成结果超时
10	-16	在线授权被取消
11	-18	在线合成无效的主机名
12	-19	在线合成读数据失败
13	-20	在线合成连接失败
14	-21	在线合成 socket 异常
15	-24	在线合成请求主机名为空
16	-25	在线合成发送数据失败
17	-29	在线合成接收前缀数据长度错误
18	-30	在线合成接收数据长度错误
19	-31	在线合成合成数据包速度过快
20	-32	在线合成网络未知类型错误
21	-39	在线服务临时错误
22	-100	离线引擎授权失败
23	-102	离线授权下载 License 失败
24	-105	离线授权中断异常
25	-106	离线授权执行时异常
26	-107	离线授权执行时间超时
27	-108	离线合成引擎初始化失败
28	-110	离线合成时异常
29	-111	离线合成返回值非 0
30	-118	离线授权任务被取消
31	-122	离线 tts_offline_resource 文件异常
32	-123	离线发音人参数异常
33	-124	下载 license 失败，sn 参数异常
34	-125	离线合成文本为空
35	-200	混合引擎离线在线都授权失败

36	-204	混合引擎初始化 tts 时，离线初始化失败
37	-206	混合引擎初始化 tts 时，在线初始化失败
38	-300	合成文本为空
39	-301	合成文本长度过长（不要超过 GBK1024 个字节）
40	-302	合成文本无法获取 GBK 字节
41	-401	TTS 模式无效
42	-402	TTS 合成队列已满
43	-406	TTS 被调用方法参数无效
44	-500	Context 被释放或为空
45	-700	播报的短音频文件不存在
46	-701	当前接口不支持播报短音频
47	-1001	模型管理请求出错
48	-1002	模型管理服务器端错误
49	-1003	模型管理数据库模型信息无效
50	-1004	模型管理数据库模型文件信息无效
51	-1005	模型数据已经存在（或已下载）
52	-1006	无法获取到模型信息
53	-1007	无法获取到模型文件信息
54	-1008	模型检查过程异常
55	-1009	模型文件下载时异常
56	-9999	未知错误

语音合成 HarmonyOS SDK

1. 文档说明

文档名称	**语音合成集成文档**
所属平台	HarmonyOS
提交日期	2026-08-11
概述	百度语音合成客户端Harmony版SDK（以下简称BDTTSClient）是一种面向HarmonyOS设备的语音合成解决方案，以Har包的形式发布。语音合成SDK分为在线SDK与离线SDK，离线SDK需要下载音库资源，在线SDK则可直接调用，默认支持流式合成。目前版本已支持SDK内部直接播放合成语音和从SDK获取语音数据，并支持男女声、语速、音调、音量、音频码率设置。 若您在2025年11月27日及之前购买或申请SN（序列号），请点击下载往期基础音库资源，及往期精品音库资源。 详细发音人信息请看离线发音人列表。
短语说明	语音合成：将文本合成为语音，即声音文件 合成引擎：将文本合成为语音的核心模块 TTS：Text To Speech，即“从文本到语音” BDTTSClient：语音合成SDK简称，是一个封装了网络收发、音频播放功能的语音合成解决方案。 语音合成SDK：封装网络收发、音频播放功能，可快速集成

2. 版本说明

名称	版本号	说明
语音合成	1.1.3	语音合成demo版本
系统支持	HarmonyOS 6.0.0（APILevel 20）+	需要开发者通过compatibleSdkVersion来保证支持系统的检测
架构支持	arm64-v8a	仅适配arm64-v8a架构，不支持x86等其他架构

3. SDK说明

3.1 开发包说明

文件名称	说明
doc/Baidu_TTS_SDK_Harmony_Manual.pdf	完整集成文档（含离线授权补充内容）
har	语音合成har包版本baidu_tts_1.0.3（1.0.3三端统一版本）
BaiduTtsDemo	开发示例（DevEco Studio project，含离线授权配置demo）
version.readme	产物版本说明（含so库、har包版本信息）

3.2 总体框图



4.集成指南

4.1 添加BDTTSClient到工程

har方式集成：将开发包中的`har`目录拷贝到工程`entry/libs/`目录；在`oh-package.json5`文件中增加以下依赖：

```
  "dependencies": {
  "@package/baidutts": "file:./libs/xxxx.har" // 替换“xxxx.har”为实际har包文件名
}
```

4.2 添加语音合成资源文件

1. BaiduTtsDemo中的资源文件、参数值仅用于demo运行体验，业务方需申请自有资源文件与参数（如离线音库、SN码等）。
2. 参照demo路径和代码逻辑，将自有离线资源文件（文本模型、发音人模型）放置到设备可访问路径（如`/data/storage/el2/base/haps/entry/files/`），并在代码中配置正确路径。

4.3 权限声明

BDTTSClient需在`module.json5`文件中声明以下权限，确保离线授权下载、网络请求正常：

```
  "requestPermissions": [
  {
    "name": "ohos.permission.INTERNET" // 在线授权、离线授权文件下载必需
  },
  {
    "name": "ohos.permission.READ_EXTERNAL_STORAGE" // 读取离线资源文件必需
  },
  {
    "name": "ohos.permission.WRITE_EXTERNAL_STORAGE" // 存储离线授权文件、资源文件必需
  }
 ]
```

4.4 离线在线授权补充说明

4.4.1 离线授权核心依赖信息

离线合成需3个核心信息完全匹配，否则触发授权失败（错误码-100、-102等），具体要求如下：

信息类型	说明	来源渠道
APP_ID	离线请求认证参数：AI开放平台用户填app_id，非AI开放平台用户填产品pid	百度语音技术控制台
授权SN码	离线专属授权码，与APP_ID、包名绑定，用于下载离线授权文件	百度语音技术控制台-离线合成SDK管理
应用包名	与百度控制台配置的包名一致，需和工程module.json5中appId完全匹配。build-profile信息参考demo配置	工程module.json5文件、百度控制台应用配置页

4.4.2 离线授权文件获取机制

1. **自动下载**：首次使用离线合成时，SDK会在联网状态下，根据APP_ID、SN码、包名自动向百度服务器请求并下载离线授权文件，默认存储路径为/data/storage/el2/base/haps/entry/files/baidu_tts_license（可通过PARAM_TTS_LICENCE_FILE参数自定义路径）。
2. **重新授权**：若授权文件过期、设备更换或包名修改，需重新申请SN码，并确保新SN与新配置信息匹配，SDK会自动重新下载授权文件。

4.4.3 离线授权失败排查要点

当出现-100（离线引擎授权失败）、-102（离线授权下载License失败）等错误时，按以下步骤排查：

1. 检查APP_ID、SN码是否正确：确认与百度控制台申请的信息一致，无空格、大小写错误。
2. 验证包名匹配：工程module.json5的appId需与百度控制台绑定的包名完全一致（离线授权与包名强绑定）。
3. 网络状态：首次授权需联网，确保设备网络正常，无防火墙拦截SDK的授权请求。
4. 存储权限：确认已申请WRITE_EXTERNAL_STORAGE权限，授权文件下载路径有写入权限。
5. SN码有效性：检查SN码是否在有效期内、是否已绑定其他设备（同一SN默认绑定有限数量设备，超限需重新申请）。

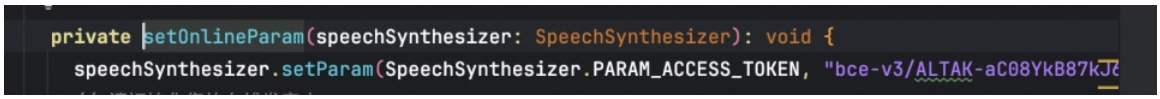
4.4.4 在线sdk鉴权:****

在线sdk分为4种激活方式，实际使用种选择其中一种方式进行激活，4种激活方法为：

激活方式	说明
永久iamKey激活	需在官网申请的永久iamKey，使用该方法进行在线鉴权，每次启动app需激活一次，不退出app永久有效
accessToken激活	需在使用appkey和secretKey获取的access_token，使用access_token进行授权，该token存在时限，时间到期后需要从新获取新的token从新授权
临时iamKey激活	与accessToken类似，需获取临时iamKey，该key存在时限，时间到期后需要从新获取新的key从新授权

4.4.5 鉴权激活方式说明****

永久iamKey激活方式：将官网申请的永久iamKey放到项目中

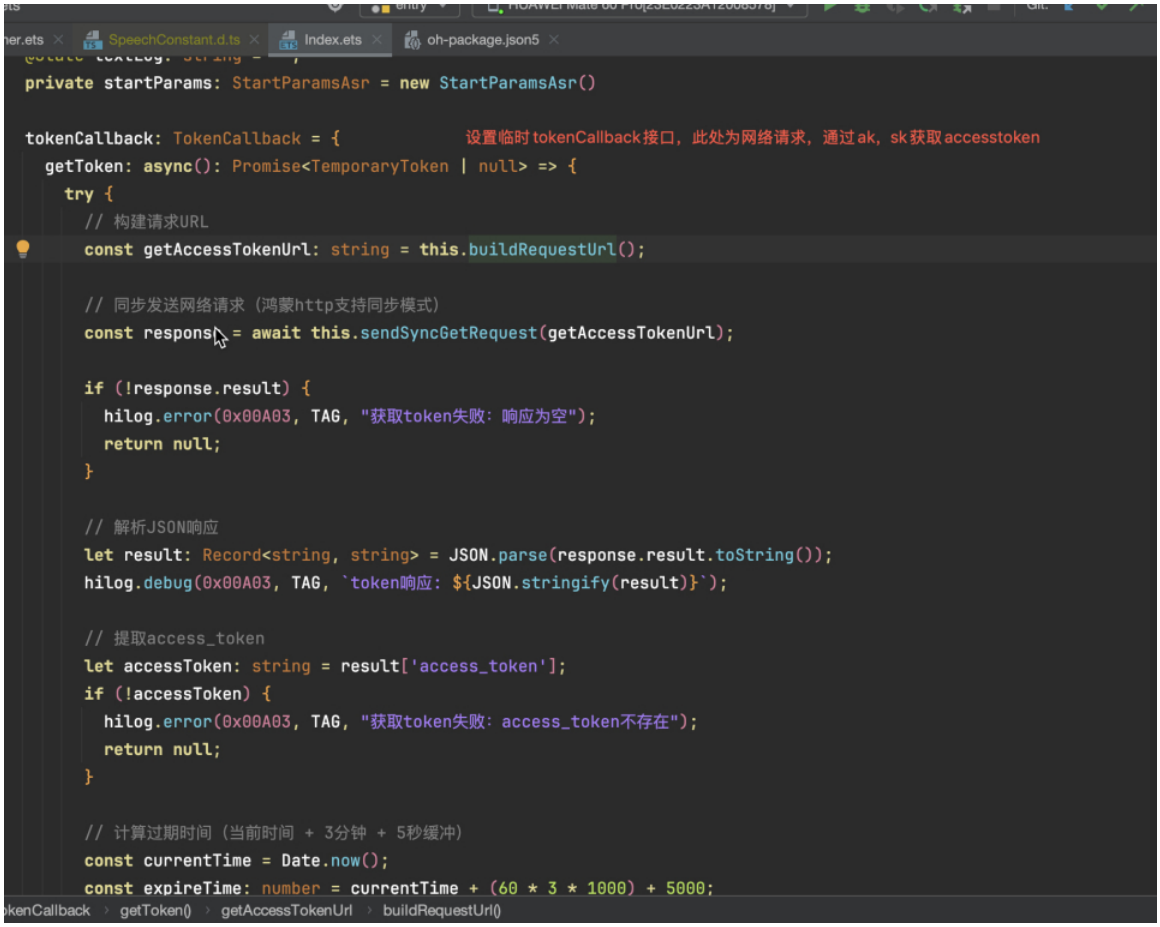


临时accessToken和临时iam key激活方式：

accessToken和临时iamKey为相同接口，接口会随时间失效，客户需要在callback中创建http请求获取到对应token返回给sdk,sdk会在过期前3分钟通过callback重新向客户端获取最新token

accessToken获取地址：https://cloud.baidu.com/doc/AI_REFERENCE/s/Km3zhy5t7

iamKey获取地址：https://cloud.baidu.com/doc/AI_REFERENCE/s/Hm5us339w



将accessToken或iamKey 还有过期时间返回给sdk



将callback传递到sdk中



5. 语音合成功能代码

5.1 TTS初始化设置（分为离在线两种方式）

```
  // SDK初始化
  private initialTts(): void {
    LoggerProxy.printable(true);
    LoggerProxy.saveNativeLogFile("PRINTF:0");
    this.printEngineInfo();
    // TODO 代码中的所有参数与文件均是测试demo使用，集成自己工程时，请使用自己的参数
    this.addDebugSpeechSynthesizer();
  }
  private addDebugSpeechSynthesizer(): void {
    // 公共参数设置
    this.setPublicParam(this.speechSynthesizer);
    // 如果需要离线，设置离线资源
    this.setOfflineParam(this.speechSynthesizer);
    // 如果需要在线，设置在线离线资源
    this.setOnlineParam(this.speechSynthesizer);
  }

    // 设置公共参数
  private setPublicParam(speechSynthesizer: SpeechSynthesizer): void {
    speechSynthesizer.setParam(SpeechSynthesizer.PARAM_APP_ID, "APPID");
    speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PLAYER_USAGE,
audio.StreamUsage.STREAM_USAGE_UNKNOWN + "");
    speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PLAYER_VOLUME_MODE,
audio.AudioVolumeMode.APP_INDIVIDUAL + "");
    // speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PLAYER_INTERRUPT_MODE,
audio.InterruptMode.INDEPENDENT_MODE + "");
  }

  // 在线语音合成
  private setOnlineParam(speechSynthesizer: SpeechSynthesizer): void {
    speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ACCESS_TOKEN, "IAMKEY或TOKEN");
    // 请初始化您的在线发音人
    speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ONLINE_SPEAKER, "4100");
    // 在线超时时间，MIX模式超时会切换至离线，ONLINE模式会超时
    speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ONLINE_TIMEOUT, "6000");
    // 初始化在线tts服务，服务会读取相应资源进行加载，此过程是耗时操作
    speechSynthesizer.loadOnlineTts().then(
      (ttsError: ITtsError) => {
        hilog.info(0x0000, 'index', 'loadOnlineTts ttsError errorCode= %{public}d，errorMessage = %{public}s',
          ttsError.getDetailCode(),
          ttsError.getDetailMessage());
      }
    );
  }

  // 离线语音合成
  private setOfflineParam(speechSynthesizer: SpeechSynthesizer): void {
    // 设置离线需要的认证参数，产品SN
    speechSynthesizer.setParam(SpeechSynthesizer.PARAM_AUTH_SERIAL_NUMBER, "填写申请的SN");
    // 离线资源库路径，设置需要的离线发音人
    speechSynthesizer.setParam(SpeechSynthesizer.PARAM_OFFLINE_MODEL, this.setOfflineModel());
    // 初始化离线tts服务，服务会读取相应离线资源进行加载，此过程是耗时操作
    speechSynthesizer.loadOfflineTts().then(
      (ttsError: ITtsError) => {
        hilog.info(0x0000, 'index', 'loadOfflineTts ttsError errorCode= %{public}d，errorMessage = %{public}s',
          ttsError.getDetailCode(),
          ttsError.getDetailMessage());
      }
    );
  }
```

5.语音合成功能代码

5.1 TTS初始化设置（分为离在线两种方式）

```
  // SDK初始化
 private initialTts(): void {
  LoggerProxy.printable(true);
  LoggerProxy.saveNativeLogFile("PRINTF:0");
  this.printEngineInfo();
  // TODO 代码中的所有参数与文件均是测试demo使用，集成自己工程时，请使用自己的参数
  this.addDebugSpeechSynthesizer();
 }
 private addDebugSpeechSynthesizer(): void {
  // 公共参数设置
  this.setPublicParam(this.speechSynthesizer);
  // 如果需要离线，设置离线资源
  this.setOfflineParam(this.speechSynthesizer);
  // 如果需要在线，设置在线离线资源
  this.setOnlineParam(this.speechSynthesizer);
 }

  // 设置公共参数
 private setPublicParam(speechSynthesizer: SpeechSynthesizer): void {
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_APP_ID, "APPID");
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PLAYER_USAGE,
audio.StreamUsage.STREAM_USAGE_UNKNOWN + "");
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PLAYER_VOLUME_MODE,
audio.AudioVolumeMode.APP_INDIVIDUAL + "");
  // speechSynthesizer.setParam(SpeechSynthesizer.PARAM_PLAYER_INTERRUPT_MODE,
audio.InterruptMode.INDEPENDENT_MODE + "");
 }

// 在线语音合成
 private setOnlineParam(speechSynthesizer: SpeechSynthesizer): void {
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ACCESS_TOKEN, "IAMKEY或TOKEN");
  // 请初始化您的在线发音人
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ONLINE_SPEAKER, "4100");
  // 在线超时时间，MIX模式超时会切换至离线，ONLINE模式会超时
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_ONLINE_TIMEOUT, "6000");
  // 初始化在线tts服务，服务会读取相应资源进行加载，此过程是耗时操作
  speechSynthesizer.loadOnlineTts().then(
   (ttsError: ITtsError) => {
    hilog.info(0x0000, 'index', 'loadOnlineTts ttsError errorCode= %{public}d , errorMessage = %{public}s',
     ttsError.getDetailCode(),
     ttsError.getDetailMessage());
   }
  );
 }

 // 离线语音合成
 private setOfflineParam(speechSynthesizer: SpeechSynthesizer): void {
  // 设置离线需要的认证参数，产品SN
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_AUTH_SERIAL_NUMBER, "填写申请的SN");
  // 离线资源库路径，设置需要的离线发音人
  speechSynthesizer.setParam(SpeechSynthesizer.PARAM_OFFLINE_MODEL, this.setOfflineModel());
  // 初始化离线tts服务，服务会读取相应离线资源进行加载，此过程是耗时操作
  speechSynthesizer.loadOfflineTts().then(
   (ttsError: ITtsError) => {
    hilog.info(0x0000, 'index', 'loadOfflineTts ttsError errorCode= %{public}d , errorMessage = %{public}s',
     ttsError.getDetailCode(),
     ttsError.getDetailMessage());
   }
  );
 }
```

5.2 参数设置

5.2.1 在线必设参数

参数名	默认值	备注
PARAM_ACCESS_TOKEN	""	产品iam api_key 或 token
PARAM_ONLINE_SPEAKER	"0"	在线发音人，参考开放平台文档获取，例“4100”

5.2.2 离线必设参数

参数名	默认值	备注
PARAM_APP_ID	“0”	离线请求认证参数，从开放平台获取，例："1234567"
PARAM_AUTH_SERIAL_NUMBER	" "	离线鉴权认证参数，从开放平台获取。例如："abcd1234-567aa1f96-148b0e-048e-11f6c4-01"
PARAM_OFFLINE_MODEL	" "	音库json信息

5.2.3 SDK默认自带基础音库离线资源文件

离线合成SDK默认自带4个普通音库资源文件，精品音库资源文件需单独下载。若您在2025年11月27日之前购买或申请序列号，请点击下载往期音库资源文件。详细发音人信息请看离线发音人列表。

资源文件	具体文件名（支持新序列号）
m15 离线男声（度小宇）	bd_etts_common_speech_duxiaoyu_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat
f7 离线女声（度小美）	bd_etts_common_speech_duxiaomei_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat
yy 离线度逍遥	bd_etts_common_speech_duxiaoyao_mand_eng_high_am-tac-csubgan16k_v4.9.0_20251110_20251210203318.dat
c1 离线度丫丫	bd_etts_common_speech_duyaya_mand_eng_high_am-tac-csubgan16k_v4.9.0_20220419_20251031153737.dat
中文离线文本模型	bd_etts_common_text_txt_all_mand_eng_middle_big_v5.5.0_20250410.dat

5.2.3 其它辅助参数

参数名	默认值	备注
PARAM_ONLINE_TIMEOUT	"6000"	在线合成请求的超时时间，单位为毫秒，最小值为 800，推荐设置为 2000
PARAM_VOLUME	"5"	合成音频音量，范围 [0-15]，float 类型
PARAM_SPEED	"5"	合成音频速度，范围 [0-15]，float 类型
PARAM_PITCH	"5"	合成音频语调，范围 [0-15]，float 类型
PARAM_OPEN_XML	"0"	"0"：关闭 xml 解析，"1"：打开 xml 解析
PARAM_TEXT_CTRL	""	前端模型配置参数，json 类型（表格笔误 son）
PARAM_AUDIO_CTRL	""	前端模型配置参数，json 类型
PARAM_BITRATE	"64"	在线音频下发类型。 PCM = 0， OPUS_64K = 64， OPUS_128K = 128
PARAM_PLAYER_VOLUME	"1.0"	播放器音量 [0,1.0]，float 类型
PARAM_PLAYER_USAGE	audio.StreamUsage.STREAM_USAGE_VOICE_ASSISTANT	播放器 usage

5.3 初始化

```
 speechSynthesizer.setParam("设置相应的参数key", "设置相应的参数value");
// 如果需要在线合成，加载在线模型
auto ttsResult = await speechSynthesizer.loadOnlineTts();
// 如果需要离线合成，加载离线模型
auto ttsResult = await speechSynthesizer.loadOfflineTts();
```

5.4 合成并播报

```
 //合成
speechSynthesizer.synthesize(TtsEntity): Promise<ITtsError>;
```

TtsEntity参数详解

- text：此次要合成的文本内容。必设参数。
- utteranceId：此次文本合成的标识，默认值 UUID 的唯一值。
- TtsMode：此次合成模式。
- ttsParams: Record<string,string>：本条合成的合成参数，仅在本条合成生效，参数参见 setParam
- 合成模式说明：ONLINE 纯在线模式，OFFLINE 纯离线模式，MIX 优先使用在线，在线超时后使用离线模式。

```
 //合成并播报
speechSynthesizer.speak(TtsEntity): Promise<ITtsError>;
```

5.5 流程控制

方法名	说明
stop()	停止当前实例合成或者停止合成播报
release()	释放当前实例，当所有实例都被释放时，tts 整个服务停止。
pause()	暂停当前实例的合成播报内容。
resume()	继续播报当前实例的播报内容，与pause()相对应。

5.6 状态监听

为了更好地实现用于界面，BDTTSClient提供了SpeechSynthesizerListener监听接口用于对合成器的状态进行通知。 完整的开发示例请参见开发包所附示例BaiduTtsDemo。

5.7 合成状态监听器回调接口

5.7.1 合成状态监听器回调接口

```
 onSynthesizeResponse(SynthesizerResponse synthesizerResponse):void;
let sn = synthesizerResponse.getSn(); // 此次SDK侧传递的文本合成标识，建议保存，后续排查服务器问题时使用
let utteranceId = synthesizerResponse.getUtteranceId(); // 此次业务侧传递的文本合成标识，用于业务逻辑判断
```

5.7.2 合成开始时的回调接口

```
 synthesizerResponse.getSynthesizeType() === SYNTHESIZE_START
```

5.7.3 合成过程中的回调接口

```
 synthesizerResponse.getSynthesizeType() === SYNTHESIZE_DATA_ARRIVED
let progress = synthesizerResponse.getAudioProgress();
let engineType = synthesizerResponse.getEngineType();
let sampleRate = synthesizerResponse.getAudioSampleRate();
let audioData = synthesizerResponse.getAudioData();
```

5.7.4 合成出错时的回调接口

```
 synthesizerResponse.getSynthesizeType() == PLAY_ERROR
synthesizerResponse.getErrorCode();
synthesizerResponse.getErrorMessage();
```

5.7.5 合成播报出错时的回调接口

```
 synthesizerResponse.getSynthesizeType() == PLAY_ERROR
synthesizerResponse.getErrorCode();
synthesizerResponse.getErrorMessage();
```

5.8 SynthesizerTool类（工具类辅助离线合成开发）

5.8.1 检验模型文件的有效性

```
 public static verifyModelFile(filePath:string):Promise<ITtsError>
```

5.8.2 获取离线引擎基本信息

```
 public static getEngineInfo():string
```

5.8.3 获取离线引擎版本信息

```
1 public static getEngineVersion():number
```

5.8.4 检查模型文件与当前引擎版本是否匹配

```
/**
 * 校验模型文件是否与当前离线引擎版本兼容（避免因版本不匹配导致合成失败）
 * @param filePath 模型文件绝对路径
 * @return 数字：0=匹配，1=模型版本过高，-1=模型版本过低，-2=文件不存在
 */
let matchResult =
SynthesizerTool.matchResEngine("/data/storage/el2/base/haps/entry/files/bd_etts_navi_speech_f7_mand_eng_high_am
-style24k_v4.6.0_20210721.dat");
switch (matchResult) {
  case 0:
    console.log("模型文件与引擎版本匹配");
    break;
  case 1:
    console.error("模型版本过高，请升级离线引擎");
    break;
  case -1:
    console.error("模型版本过低，请下载最新模型文件");
    break;
  case -2:
    console.error("模型文件不存在，请检查路径");
    break;
}
```

5.8.5 获取音库文件的采样率

```
/**
 * 获取离线发音人模型（speech.dat）的音频采样率（用于自定义音频播放配置）
 * @param filePath 发音人模型文件绝对路径
 * @return 数字：采样率（如16000、24000，单位Hz；返回-1表示文件无效）
 */
let sampleRate =
SynthesizerTool.getSpeechSampleRate("/data/storage/el2/base/haps/entry/files/bd_etts_navi_speech_f7_mand_eng_high
-style24k_v4.6.0_20210721.dat");
if (sampleRate !== -1) {
  console.log("发音人模型采样率：" + sampleRate + "Hz");
}
```

5.8.6 获取定制化语音包的采样率

```
public static getDomainSampleRate(filePath:string):number
```

5.8.7 验证定制化语音包是否可用

```
public static checkDomainFile(filePath:string):number
```

5.8.8 获取sdk内部当前使用的设备id

```
// 建议在new Synthesizer()之后调用
public static getCuid(): string
```

6.错误码列表

错误码值	错误码描述
-1	在线引擎授权失败
-4	在线授权中断异常
-5	在线授权执行时异常
-6	在线授权时间超时
-7	在线合成返回错误信息
-10	在线引擎合成时异常
-11	当前 mode 不支持的操作
-12	在线合成请求解析出错
-15	在线合成获取合成结果超时
-16	在线授权被取消
-18	在线合成无效的主机名
-19	在线合成读数据失败
-20	在线合成连接失败
-21	在线合成 socket 异常
-24	在线合成请求主机名为空
-25	在线合成发送数据失败
-29	在线合成接收前缀数据长度错误
-30	在线合成接收数据长度错误
-31	在线合成合成数据包速度过快
-32	在线合成网络未知类型错误
-39	在线服务临时错误
-100	离线引擎授权失败
-102	离线授权下载 License 失败
-105	离线授权中断异常
-106	离线授权执行时异常
-107	离线授权执行时间超时
-108	离线合成引擎初始化失败
-110	离线合成时异常
-111	离线合成返回值非 0
-118	离线授权任务被取消
-122	离线 tts_offline_resource 文件异常
-123	离线发音人参数异常
-124	下载 license 失败，sn 参数异常
-125	离线合成文本为空
-200	混合引擎离线在线都授权失败
-204	混合引擎初始化 tts 时，离线初始化失败
-206	混合引擎初始化 tts 时，在线初始化失败
-208	合成文本为空

-300	合成文本为空
-301	合成文本长度过长（不要超过 GBK1024 个字节）
-302	合成文本无法获取 GBK 字节
-401	TTS 模式无效
-402	TTS 合成队列已满
-406	TTS 被调用方法参数无效
-500	Context 被释放或为空
-700	播报的短音频文件不存在
-701	当前接口不支持播报短音频
-1001	模型管理请求出错
-1002	模型管理服务器端错误
-1003	模型管理数据库模型信息无效
-1004	模型管理数据库模型文件信息无效
-1005	模型数据已经存在（或已下载）
-1006	无法获取到模型信息
-1007	无法获取到模型文件信息
-1008	模型检查过程异常
-1009	模型文件下载时异常
-9999	未知错误

7. FAQ

1.日志收集

（1）TTS SDK Java 层日志 收集方式：使用 logcat 抓取日志并导出日志文件

开启日志打印代码：

```
LoggerProxy.printable(true);
```

（2）TTS SDK 引擎层日志

两种输出方式：

方式 1：输出引擎日志到本地指定文件

```
LoggerProxy.saveNativeLogFile("FPRINTF:0:" + File.separator + context.getExternalFilesDir(null).getPath() + File.separator + "tts_engine.log");
```

方式 2：仅开启引擎日志控制台输出

```
LoggerProxy.saveNativeLogFile("PRINTF:0");
```

收集方式：配合 logcat 收集，或直接读取本地输出的日志文件

2.离线引擎信息

```
String engineInfo = SynthesizerTool.getEngineInfo();
```

3.使用的文本资源信息

传入**离线 TTS 文本资源文件绝对路径**，获取文本模型详情：

```
String textModelInfo = SynthesizerTool.getModelInfo(初始化离线TTS引擎使用的文本资源绝对路径);
```

4.使用的音库资源信息

传入**离线 TTS 音库资源文件绝对路径**，获取音库模型详情：

```
String speechModelInfo = SynthesizerTool.getModelInfo(初始化离线TTS引擎使用的音库资源绝对路径);
```

5.查看 SDK 版本号信息（bash 命令）

进入 so 库所在目录执行以下指令：

```
##### 查看TTS SDK版本
strings -a libBDSpeechClientSDK.so | grep TTS_SDK_VERSION
##### 查看ETTS唯一标识
strings -a libBDSpeechClientSDK.so | grep ETTS_UNIQUE
```

语音合成 Linux SDK

1. 文档说明

文档名称	**语音合成集成文档**
所属平台	Linux C++
提交日期	2025-09-03
概述	本文档是百度离线语音合成Linux Arm SDK的用户指南，描述了离线语音合成SDK相关接口的使用说明。本文档设计为在 Linux arm 平台上使用百度语音合成(BDS TTS)SDK 进行文本转语音的开发。该 SDK 支持离线语音合成模式，并提供丰富的参数配置进行一些功能性设置。 Linux SDK适配4.6版本音库，您可以点击下载往期基础音库资源，及往期精品音库资源。详细发音人信息请看离线发音人列表。

2. 版本说明

请确保Linux SDK版本适配、测试可用后再进行购买，购买后不可退换。更多信息可提交需求工单咨询和了解。

名称	**版本与兼容范围**
语音合成	1.0.0
系统支持	Linux C++ 接口建议使用 Ubuntu 22.04 及以上版本，可支持Ubuntu 18.04版本
开发环境	要求支持C++17标准，GCC（GNU Compiler Collection）版本须 ≥ 7.3
架构支持	ARM64 支持x86_64架构，最低需支持libc.so.6，并包含iconv模块，详见下方开发环境

3. SDK说明

将SDK 压缩包解压缩，将得到如下文件：

文件名称	说明
百度语音合成Linux Arm SDK用户接入文档V1.0	本文档
demo	开发实例
include	依赖的头文件
libs	语音合成SDK库

4.开发环境

4.1 开发环境 SDK 支持在 Linux arm(armv8) 环境下进行编译，可选择：

- **交叉编译**：在 x86_64 主机上交叉构建；
- **本地编译**：直接在 64 位 ARM 开发板环境下构建。

在开发板上编译请确保已下载安装gcc、g++、cmake，如未安装可执行如下指令：

```
sudo apt install gcc g++ cmake
```

然后在 sdk demo 文件夹执行如下指令：

```
mkdir build
cd build
cmake ..
make -j4
```

编译完成后执行如下指令：

```
sh run.sh
```

4.2 依赖 确保系统已安装必要的依赖库如: libstdc++、pthread、dl、iconv 等。

SDK 提供的依赖文件结构如下：

- **include/**：头文件
- **libs/**：库文件

> 注意：部分发行版的 `iconv` 模块可能独立于 `glibc`，请确认系统自带或手动安装。

4.3 资源文件 可存放在如 `resource`文件夹，具体可参考示例。主要包括：文本模型文件、语音模型文件、授权文件（在激活时自动生成）。

参考目录结构：

```
sdk/
├── include/
├── libs/
├── resource/
├── demo/
└── run.sh
```

离线合成SDK默认自带4个普通音库资源文件，精品音库资源文件需单独下载。详细发音人信息请看离线发音人列表。

离线资源文件（文本语音模型文件）

资源文件	具体文件名（支持新序列号）
m15 离线男声（度小宇）	bd_etts_common_speech_duxiaoyu_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat
f7 离线女声（度小美）	bd_etts_common_speech_duxiaomei_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat
yy 离线度逍遥	bd_etts_common_speech_duxiaoyao_mand_eng_high_am-tac-csubgan16k_v4.9.0_20251028_20251031153737.dat
c1 离线度丫丫	bd_etts_common_speech_duyaya_mand_eng_high_am-tac-csubgan16k_v4.9.0_20220419_20251031153737.dat
中文离线文本模型	bd_etts_common_text_txt_all_mand_eng_middle_big_v5.5.0_20250410.dat

> 不同 Linux 平台的工具链差异较大，若仅用于功能体验或初步联调，可直接使用提供的 **Ubuntu x86_64 预编译版本**（Ubuntu ≥ 18.04）。

5.授权激活

本SDK 支持在线合成(online)和离线合成(offline)以及根据网络情况使用离在线融合模式(Mix)。需要通过设置正确的参数进行激活。

accessToken、包名等参数填写与申请请参考语音合成SDK集成。accessToken获取请参考鉴权认证机制，iamKey（可等同于accessToken一样使用）获取请参考API Key鉴权机制。

第一步 设置 accessToken

```
tts_set_param(bds::BdsTTSParamDefine::PARAM_ACCESS_TOKEN, "您申请的accessToken");
```

第二步 设置包名、appID、SN 设置包名请使用以下语句

```
tts_set_param(bds::BdsTTSParamDefine::PARAM_PACKAGE_NAME, "您申请的包名");
```

设置app ID请使用以下语句

```
tts_set_param(bds::BdsTTSParamDefine::PARAM_APP_ID, "您申请的appId");
```

设置SN请使用以下语句

```
tts_set_param(bds::BdsTTSParamDefine::PARAM_AUTH_SERIAL_NUMBER,"您申请的SN");
```

第三步 设置发音人 若希望指定发音人，还需要设置发音人ID。例如想要使用度泽言发音人，需要把ID设为4193。

```
tts_set_param(bds::BdsTTSParamDefine::PARAM_ONLINE_SPEAKER, "4193");
```

需要注意前面设置的 4.1、4.2权限是否包含该发音人权限。

6.SDK接入

6.1 Debug日志初始化

```
 void tts_log_init() {
   // 输出到终端
   bds::BdsTTSLogger::getInstance().set_log_level(
       bds::BdsTTSLogLevel::BDS_TTS_LOG_VERBOSE);
   // 输出到文件
   // bds::BdsTTSLogger::getInstance().set_log_level(
   // bds::BdsTTSLogLevel::BDS_TTS_LOG_VERBOSE , get_executable_path());
   bds::BdsTTSLogger::getInstance().setOfflineEngineLogFilePath(
       "FPRINTF:0:" + get_executable_path() + "/engine.txt");
}
```

6.2 创建合成器实例

```
 class BdsTTSSynthesizerListenerImpl : public bds::BdsTTSSynthesizerListener {
public:
   void on_synthesizer_response(const bds::BdsTTSSynthesizerResponse response) {
       // 处理合成回调
   }
} ;

static std::shared_ptr<bds::BdsTTSSynthesizer> g_tts_handler = nullptr;
static BdsTTSSynthesizerListenerImpl g_native_observer;

void tts_handler_init() {
   bds::BdsTTSSynthesizer *raw_ptr =
       bds::BdsTTSSynthesizer::create_bds_tts_synthesizer(g_native_observer);
   g_tts_handler = std::shared_ptr<bds::BdsTTSSynthesizer>(raw_ptr);
}
```

6.3 参数配置 通用参数

```
 void tts_set_param(const std::string &key, const std::string &value) {
   g_tts_handler->set_param(key, value);
}

// 示例配置
tts_set_param(bds::BdsTTSParamDefine::PARAM_PACKAGE_NAME, "com.baidu.speech.demo");
tts_set_param(bds::BdsTTSParamDefine::PARAM_ACCESS_TOKEN, "your_access_token");
```

在线合成参数 若使用在线语音合成，请配置以下参数

```
 tts_set_paramCbds::BdsTTSParamDefine::PARAMLONLINE_SPEAKER, "4100"); //发音人
tts_set_param(bds::BdsTTSParamDefine::PARAM_ONLINE_TIMEOUT, "2000"); //超时时间
```

离线合成参数 若使用离线语音合成，请配置以下参数

```
 std::string setOfflineModel(std::string exe_dir) {
   auto text_path_key = bds::BdsTTSParamDefine::TTS_TEXT_MODEL_FILE;
   auto text_path = escape_json_string(exe_dir + "/path/to/text_model.dat");

   auto speech_path_key = bds::BdsTTSParamDefine::TTS_SPEECH_MODEL_FILE ;
   auto speech_path = escape_json_string(exe_dir + "/path/to/speech_model.dat");

   std::string offline_model = "{\""+ text_path_key + "\":\"" + text_path +
                               "\",\"" + speech_path_key +"\":\"" +
                               speech_path +"\"}";
   return offline_model;
}

tts_set_param(bds::BdsTTSParamDefine::PARAM_APP_ID, "your_app_id");
tts_set_param(bds::BdsTTSParamDefine::PARAM AUTH_SERIAL_NUMBER, "your_serial");
tts_set_param(bds::BdsTTSParanDefine::PARAM_TTS_LICENSE_FILE_PATH,
          exe_dir + "/path/to/baidu_license.dat") ; // 可读可写文件
tts_set_param（bds::BdsTTSParamDefine::PARAM_RW_PATH, exe_dir） ; // 可读可写路径
tts_set_param(bds::BdsTTSParamDefine::PARAM_OFFLINE_MODEL, setOfflineModel(exe_dir));
tts_set_param（bds::BdsTTSParamDefine::PARAM_CONVERT_DAT_PATH, "your convert file"） ; // 可选参数，使用非当
前平台生成的个性化发音人时设置
```

6.4 加载引擎 加载在线引擎 若使用在线语音合成，请通过以下代码加载引擎

```
 bool load_online_tts() {
   auto tts_result = g_tts_handler->load_online_tts();
   return tts_result.get_code() == 0;
}
```

加载离线引擎 若使用离线语音合成，请通过以下代码加载引擎

```
 bool load_offline_tts() {
   auto tts_result = g_tts_handler->load_offline_tts();
   return tts_result.get_code() == 0;
}
```

6.5 语音合成 6.5.1 开始合成

```
 bool synthesize(const bds::BdsTTSEntity &ttsEntity) {
   auto tts_result = g_tts_handler->synthesize(ttsEntity);
   return tts_result.get_code() == 0;
}

// 使用示例
auto text = "要合成的文本内容" ;
auto tts_mode = bds::BdsTtsMode::MIX; // 混合模式
auto utterance_id = "unique_id_123";
auto ttsEntity = bds::BdsTTSEntity(text, tts_mode, utterance_id);
synthesize(ttsEntity);
```

6.5.2 合成回调处理

```
 void on_synthesizer_response(const bds::BdsTTSSynthesizerResponse &response) {
   auto synthesize_type = response.get_synthesize_type();

   switch (synthesize_type) {
   case bds::SynthesizerType::SYNTHESIZE_START:
   case bds::SynthesizerType::SYNTHESIZE_FINISH:
      // 处理开始/结束事件
      break;

   case bds::SynthesizerType::SYNTHESIZE_DATA_ARRIVED:
      // 处理数据到达事件，获取进度等信息
      break;

   case bds::SynthesizerType::SYNTHESIZE_ERROR:
      // 处理错误事件
      break;
   }
}
```

6.5.3 控制接口

```
 // 停止合成
g_tts_handler->stop();

// 释放资源
释放bds::BdsTTSSynthesizer *指针即可
```

完整示例如下

```
 //初始化
tts_log_init();
tts_handler_init();

// 配置参数
tts_set_param(bds::BdsTTSParamDefine::PARAM_PACKAGE_NAME, "com.baidu.speech.demo");
tts_set_param(bds::BdsTTSParamDefine::PARAM ACCESS_TOKEN, "your_token");
tts_set_param(bds::BdsTTSParamDefine::PARAM_ONLINE_SPEAKER, "4100");

// 加载在线引擎
if (!load_online_tts()) return -1;

// 配置离线参数
tts_set_param(bds::BdsTTSParamDefine::PARAM APP_ID, "your_app_id");
tts_set_param(bds::BdsTTSParamDefine::PARAM_OFFLINE_MODEL, setOfflineModel(get_executable_path()));

// 加载离线引擎
if (!load_offline_tts()) return -1;

//开始合成
auto text = "要合成的文本内容”；
auto ttsEntity = bds::BdsTTSEntity(text, bds::BdsTtsMode::MIX, "123456");
synthesize(ttsEntity);

// 等待合成完成
sleep(10);
```

6.6 合成参数设置 参考前述参数配置项，设置合成参数采用方法：

tts_set_param 进行设置，各参数意义描述如下:

参数名	类型、值	在线或离线生效	常用程度	解释
PARAM_SPEAKER (基础发音人)	选项	在线	常用	仅在线生效，在线的发音
	"0"(默认)	在线	常用	度小美(普通女声)
	"1"	在线	常用	度小宇(成熟男声)
	"3"	在线	常用	度逍遥(磁性男声)
	"4"	在线	常用	度丫丫(可爱女童)
PARAM_SPEAKER (精品发音人)	"106"	在线	常用	度博文(情感男声)
	"110"	在线	常用	度小童(情感儿童声)
	"111"	在线	常用	度小萌(情感女声)
	"103"	在线	常用	度米朵(情感儿童声)
	"5"	在线	常用	度小娇(情感女声)
	"5003"	在线	常用	精品度逍遥(磁性男声)
	"5118"	在线	常用	度小鹿(甜美女声)
PARAM_VOLUME	String, 默认"5"	全部	常用	在线及离线合成的音量 。范围["0" - "15"]，不支持小数。 "0"最轻，"15" 最响（取值为0时为音量最小值，并非为无声）
PARAM_SPEED	String, 默认"5"	全部	常用	在线及离线合成的语速 。范围["0" - "15"]，不支持小数。 "0"最慢，"15" 最快(如需更高语速可提交需求工单或联系商务同学申请)

7.常见问题

Q: 合成没有声音输出

A: 检查音频设备是否正常，检查合成回调是否有错误信息

Q: 离线合成失败

A: 检查模型文件路径是否正确，授权文件是否有效

Q: 在线合成超时

A: 检查网络连接，适当增加超时时间

Q: 如何切换发音人

A: 通过 PARAM_ONLINE_SPEAKER 参数设置不同的发音人 ID

错误码

错误码值	错误码描述
-1	在线引擎授权失败
-4	在线授权中断异常
-5	在线授权执行时异常
-6	在线授权时间超时

-7	在线合成返回错误信息
-10	在线引擎合成时异常
-11	当前 mode 不支持的操作
-12	在线合成请求解析出错
-15	在线合成获取合成结果超时
-16	在线授权被取消
-18	在线合成无效的主机名
-19	在线合成读数据失败
-20	在线合成连接失败
-21	在线合成 socket 异常
-24	在线合成请求主机名为空
-25	在线合成发送数据失败
-29	在线合成接收前缀数据长度错误
-30	在线合成接收数据长度错误
-31	在线合成合成数据包速度过快
-32	在线合成网络未知类型错误
-39	在线服务临时错误
-100	离线引擎授权失败
-102	离线授权下载 License 失败
-105	离线授权中断异常
-106	离线授权执行时异常
-107	离线授权执行时间超时
-108	离线合成引擎初始化失败
-110	离线合成时异常
-111	离线合成返回值非 0
-118	离线授权任务被取消
-122	离线 tts_offline_resource 文件异常
-123	离线发音人参数异常
-124	下载 license 失败，sn 参数异常
-125	离线合成文本为空
-200	混合引擎离线在线都授权失败
-204	混合引擎初始化 tts 时，离线初始化失败
-206	混合引擎初始化 tts 时，在线初始化失败
-300	合成文本为空
-301	合成文本长度过长(不要超过 GBK1024 个字节)
-302	合成文本无法获取 GBK 字节
-401	TTS 模式无效
-402	TTS 合成队列已满
-406	TTS 被调用方法参数无效
-500	Context 被释放或为空

-700	播报的短音频文件不存在
-701	当前接口不支持播报短音频
-1001	模型管理请求出错
-1002	模型管理服务器端错误
-1003	模型管理数据库模型信息无效
-1004	模型管理数据库模型文件信息无效
-1005	模型数据已经存在(或已下载)
-1006	无法获取到模型信息
-1007	无法获取到模型文件信息
-1008	模型检查过程异常
-1009	模型文件下载时异常
-9999	未知错误

流式文本在线合成JAVA SDK

快速入门

安装Speech Java SDK Speech 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
    ├─── speech
    │      └─── AipSpeech       //AipSpeech类
    └─── util                    //工具类
```

支持 JAVA版本：1.7+

查看源码

Java SDK代码现已公开，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。请点击此处下载java sdk源代码。

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.在官方网站下载识别、合成 RESTful API 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` `log4j-1.2.17.jar`。

其中，version为版本号，添加完成后，用户就可以在工程中使用Speech Java SDK。

新建AipSpeech

AipSpeech是语音识别与合成的Java客户端，为使用语音合成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AipSpeech,初始化完成后建议**单例使用**,避免重复获取access_token：

```
public class StreamTtsExample {
  // 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";
  private static final Logger LOGGER = LoggerFactory.getLogger(StreamTtsExample.class);

  public static void main(String[] args) {
    // 初始化一个AipSpeech
    AipSpeech client = new AipSpeech(APP_ID, API_KEY, SECRET_KEY);

    // 可选：设置网络连接参数
    client.setConnectionTimeoutInMillis(2000);
    client.setSocketTimeoutInMillis(60000);

    // 调用流式合成接口
    StreamTtsSynthesis(client);
  }

  private static void StreamTtsSynthesis(AipSpeech client) {
    // 创建请求参数
    StreamTtsRequest request = new StreamTtsRequest();
    request.setSpd(5); // 语速，取值0-15，默认为5中语速
    request.setPit(5); // 音调，取值0-15，默认为5中语调
    request.setVol(5); // 音量，取值0-15，默认为5中音量
    request.setPer(4146); // 发音人选择, 4146:百度丫丫-情感女声
    request.setAue(3); // 音频格式选择 3:mp3格式(默认) 4:pcm-16k 5:pcm-8k 6:wav格式
    // 采样率，仅支持将采样率降采为16k。
    // request.setSamplingRate();

    // 要合成的文本片段数组
    String[] texts = {
        "飞船在深空中缓缓滑行，舷窗外的星云泛着微",
        "弱的蓝光。指挥官林远重新校准了导航系统，",
        "确保他们不会偏离既定航道。远处，一颗无名",
        "行星的影子逐渐浮现，其表面布满裂痕，仿佛",
        "经历过无数次天体撞击。科研员苏晴调出了能",
        "量探测图，发现地壳深处竟有稳定的能量脉冲",
        "反应。她压低声音对林远说：“长官，我们可",
        "能发现了一种未知文明的遗迹。”众人心头猛",
        "地一震，飞船上的气氛顿时紧张起来，所有仪",
        "器的读数都在快速变化，像在预示着什么即将",
        "发生。飞船继续逼近，那股神秘能量愈发清晰。"
    };

   // 一次性输入
    // String[] texts = {"飞船在深空中缓缓滑行，舷窗外的星云泛着微弱的蓝光。指挥官林远重新校准了导航系统，确保他们不会偏离既定航道。远处，一颗无名行星的影子逐渐浮现，其表面布满裂痕，仿佛经历过无数次天体撞击。科研员苏晴调出了能量探测图，发现地壳深处竟有稳定的能量脉冲反应。她压低声音对林远说：“长官，我们可能发现了一种未知文明的遗迹。”众人心头猛地一震，飞船上的气氛顿时紧张起来，所有仪器的读数都在快速变化，像在预示着什么即将发生。飞船继续逼近，那股神秘能量愈发清晰。"};

    // 文件合成路径
    String outputPath = "output.mp3";
```

```
// 创建同步锁存器
CountDownLatch completionLatch = new CountDownLatch(1);

// 创建监听器
StreamTtsListener listener = new StreamTtsListener() {
    private FileOutputStream fos;

    @Override
    public void onSynthesisStart(String sessionId) {
        LOGGER.info("Stream synthesis started, sessionId: {}", sessionId);
        try {
            fos = new FileOutputStream(outputPath);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onAudioData(String sessionId, byte[] data) {
        LOGGER.info("Received audio data, length: {} byte, sessionId: {}", data.length, sessionId);
        // 将音频数据写入文件
        try {
            if (fos != null) {
                fos.write(data);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onSynthesisEnd(String sessionId) {
        LOGGER.info("Stream synthesis ended, sessionId: {}", sessionId);
        try {
            if (fos != null) {
                fos.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        LOGGER.info("Audio file saved as {}", outputPath);
        // 释放锁存器
        completionLatch.countDown();
    }

    @Override
    public void onSynthesisFail(JSONObject response) {
        LOGGER.error("Stream synthesis failed: {}", response.toString());
        try {
            if (fos != null) {
                fos.close();
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            // 出现错误也释放锁存器
            completionLatch.countDown();
        }
    }
};

// 创建语音合成器
```

```
        // 创建流式合成器
        StreamTtsSession session = null;
        try {

            session = client.streamSynthesis(client, request, listener);

            // 启动流式合成
            session.start();

            // 发送文本片段
            for (String text : texts) {
                LOGGER.info("Send Text Request: {}", text);
                session.process(text);
                // 模拟用户间隔输入文本的场景
                Thread.sleep(500);
            }

            // 停止流式合成，默认时间为60s，可根据文本长度自定义时间
            session.stop(); // 可传入参数 60000
            LOGGER.info("Stream synthesis stopped!");

            // 等待合成完成，最多等待120秒,可根据文本长度自定义时间
            if (completionLatch.await(120, TimeUnit.SECONDS)) {
                LOGGER.info("Synthesis completed!");
            } else {
                LOGGER.error("Synthesis timeout!");
            }
        } catch (Exception e) {
            LOGGER.error("Error processing: {}", e.getMessage(), e);
            e.printStackTrace();
        } finally {
            if (session != null) {
                session.close(); //关闭连接
            }
        }
    }
}
```

在上面代码中，常量APP_ID在百度云控制台中创建，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。

配置AipSpeech

如果用户需要配置AipSpeech的一些细节参数，可以在构造AipSpeech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

流式语音合成

详细文档参考官网：https://cloud.baidu.com/doc/SPEECH/s/lm5xd63rn

接口描述

基于该接口，开发者可以轻松获取流式语音合成能力。流式合成相比普通合成具有以下优势：

- 实时性更强：边合成边播放，减少等待时间
- 内存占用更小：无需等待全部合成完毕再获取音频
- 用户体验更好：可以更快听到第一段语音内容

功能说明

- 建议文本不超过2000 GBK字节，即1000个汉字或者字母数字
- 输入的文本必须采用UTF-8编码
- 支持多音字通过标注自行定义发音。格式如：重(chong2)报集团

请求说明

流式语音合成采用WebSocket协议进行通信，支持长文本分段传输合成。 Java Demo 如下：

其中，文本支持分段输入，也支持一次性输入，但需注意长度限制。

```
 import com.baidu.aip.speech.AipSpeech;
import com.baidu.aip.speech.StreamTtsListener;
import com.baidu.aip.speech.StreamTtsRequest;
import com.baidu.aip.speech.StreamTtsSession;
import org.json.JSONObject;

import java.io.FileOutputStream;
import java.io.IOException;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.TimeUnit;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

/**
 * 流式TTS合成示例
 */
public class StreamTtsExample {
    // 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";
    private static final Logger LOGGER = LoggerFactory.getLogger(StreamTtsExample.class);

    public static void main(String[] args) {
        // 初始化一个AipSpeech
        AipSpeech client = new AipSpeech(APP_ID, API_KEY, SECRET_KEY);

        // 可选：设置网络连接参数
        client.setConnectionTimeoutInMillis(2000);
        client.setSocketTimeoutInMillis(60000);

        // 调用流式合成接口
        StreamTtsSynthesis(client);
    }

    private static void StreamTtsSynthesis(AipSpeech client) {
        // 创建请求参数
        StreamTtsRequest request = new StreamTtsRequest();
        request.setSpd(5); // 语速，取值0-15，默认为5中语速
        request.setPit(5); // 音调，取值0-15，默认为5中语调
```

```
        request.setVol(5); // 音量，取值0-15，默认为5中音量
        request.setPer(4146); // 发音人选择, 4146:百度丫丫-情感女声
        request.setAue(3); // 音频格式选择 3:mp3格式(默认) 4:pcm-16k 5:pcm-8k 6:wav格式
        // 采样率，仅支持将采样率降采为16k。
        // request.setSamplingRate();

        // 要合成的文本片段数组
        String[] texts = {
                "飞船在深空中缓缓滑行，舷窗外的星云泛着微",
                "弱的蓝光。指挥官林远重新校准了导航系统，",
                "确保他们不会偏离既定航道。远处，一颗无名",
                "行星的影子逐渐浮现，其表面布满裂痕，仿佛",
                "经历过无数次天体撞击。科研员苏晴调出了能",
                "量探测图，发现地壳深处竟有稳定的能量脉冲",
                "反应。她压低声音对林远说：“长官，我们可",
                "能发现了一种未知文明的遗迹。”众人心头猛",
                "地一震，飞船上的气氛顿时紧张起来，所有仪",
                "器的读数都在快速变化，像在预示着什么即将",
                "发生。飞船继续逼近，那股神秘能量愈发清晰。"
        };

    // 一次性输入
        // String[] texts = {"飞船在深空中缓缓滑行，舷窗外的星云泛着微弱的蓝光。指挥官林远重新校准了导航系统，确保他们不会偏离既定航道。远处，一颗无名行星的影子逐渐浮现，其表面布满裂痕，仿佛经历过无数次天体撞击。科研员苏晴调出了能量探测图，发现地壳深处竟有稳定的能量脉冲反应。她压低声音对林远说：“长官，我们可能发现了一种未知文明的遗迹。”众人心头猛地一震，飞船上的气氛顿时紧张起来，所有仪器的读数都在快速变化，像在预示着什么即将发生。飞船继续逼近，那股神秘能量愈发清晰。"};

        // 文件合成路径
        String outputPath = "output.mp3";

        // 创建同步锁存器
        CountDownLatch completionLatch = new CountDownLatch(1);

        // 创建监听器
        StreamTtsListener listener = new StreamTtsListener() {
            private FileOutputStream fos;

            @Override
            public void onSynthesisStart(String sessionId) {
                LOGGER.info("Stream synthesis started, sessionId: {}", sessionId);
                try {
                    fos = new FileOutputStream(outputPath);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }

            @Override
            public void onAudioData(String sessionId, byte[] data) {
                LOGGER.info("Received audio data, length: {} byte, sessionId: {}", data.length, sessionId);
                // 将音频数据写入文件
                try {
                    if (fos != null) {
                        fos.write(data);
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }

            @Override
```

```
        public void onSynthesisEnd(String sessionId) {
            LOGGER.info("Stream synthesis ended, sessionId: {}", sessionId);
            try {
                if (fos != null) {
                    fos.close();
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
            LOGGER.info("Audio file saved as {}", outputPath);
            // 释放锁存器
            completionLatch.countDown();
        }

        @Override
        public void onSynthesisFail(JSONObject response) {
            LOGGER.error("Stream synthesis failed: {}", response.toString());
            try {
                if (fos != null) {
                    fos.close();
                }
            } catch (Exception e) {
                e.printStackTrace();
            } finally {
                // 出现错误也释放锁存器
                completionLatch.countDown();
            }
        }
    };

    // 创建流式合成器
    StreamTtsSession session = null;
    try {

        session = client.streamSynthesis(client, request, listener);

        // 启动流式合成
        session.start();

        // 发送文本片段
        for (String text : texts) {
            LOGGER.info("Send Text Request: {}", text);
            session.process(text);
            // 模拟用户间隔输入文本的场景
            Thread.sleep(500);
        }

        // 停止流式合成，默认时间为60s，可根据文本长度自定义时间
        session.stop(); // 可传入参数 60000
        LOGGER.info("Stream synthesis stopped!");

        // 等待合成完成，最多等待120秒,可根据文本长度自定义时间
        if (completionLatch.await(120, TimeUnit.SECONDS)) {
            LOGGER.info("Synthesis completed!");
        } else {
            LOGGER.error("Synthesis timeout!");
        }
    } catch (Exception e) {
        LOGGER.error("Error processing: {}", e.getMessage(), e);
        e.printStackTrace();
    } finally {
        if (session != null) {
```

```
                session.close(); //关闭连接
            }
        }
    }
}
```

StreamTtsRequest参数详解

参数名称	类型	是否必填	说明
spd	int	可选	语速，取值 0-15，默认为 5
pit	int	可选	音调，取值 0-15，默认为 5
vol	int	可选	音量，基础音库取值0-9，其他音库取值 0-15，默认为 5
per	int	必填	发音人选择，具体值参考官方文档
aue	int	可选	音频格式，3=mp3-16k/24k，4=pcm-16k/24k，5=pcm-8k，6=wav-16k/24k，默认为3
audio_ctrl	string	可选	采样率，仅支持将采样率降采为16k。（格式：{"sampling_rate":16000}）

StreamTtsListener回调方法

方法	说明
onSynthesisStart	当WebSocket连接打开，服务端返回开始合成响应时调用
onAudioData	当收到音频数据时调用
onSynthesisEnd	当合成结束时调用
onSynthesisFail	当发生错误时调用

返回说明

流式语音合成通过监听器回调方式返回结果：

1. **onSynthesisStart**: 合成开始通知
2. **onAudioData**: 音频数据，可直接用于播放或保存
3. **onSynthesisEnd**: 合成结束通知
4. **onSynthesisFail**: 错误信息，包含错误码和错误描述

错误处理

流式TTS合成过程中可能出现多种类型的错误：

1. **连接错误**：WebSocket连接失败
2. **认证错误**：access_token无效或过期
3. **参数错误**：请求参数不符合规范
4. **服务端错误**：服务器内部错误
5. **网络超时**：请求超时

建议在onSynthesisFail回调中进行相应处理。

注意事项

1. **会话管理**：每个StreamTtsSynthesizer实例代表一个独立的合成会话，请勿并发使用同一实例
2. **资源释放**：使用完毕后务必调用synthesizer.close()方法释放连接资源

3. **超时设置**：合理设置start()和stop()方法的超时时间

4. **文本分割**：长文本建议按语义分割成适当长度的片段分别发送

5. **异常处理**：注意捕获并处理可能抛出的各种异常，包括网络异常、参数异常等

6. **音频格式**：根据实际需求选择合适的音频格式，不同格式的数据大小和质量有所差异

长文本在线合成JAVA SDK

快速入门

安装Speech Java SDK Speech 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
    ├─── speech
    │      └─── AipSpeech       //AipSpeech类
    └─── util                    //工具类
```

支持 JAVA版本：1.7+

查看源码 Java SDK代码现已公开，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。请点击此处下载java sdk源代码。

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.在官方网站下载识别、合成 RESTful API 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` `log4j-1.2.17.jar`。

其中，`version`为版本号，添加完成后，用户就可以在工程中使用Speech Java SDK。

新建AipSpeech

AipSpeech是语音识别与合成的Java客户端，为使用语音合成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AipSpeech，初始化完成后建议**单例使用**，避免重复获取access_token：

```
public class LongTtsCreate {
    // 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AipSpeech
        AipSpeech client = new AipSpeech(APP_ID, API_KEY, SECRET_KEY);
        List<String> texts = Arrays.asList(
                "你好百度，",
                "你好世界。"
        );

        // 可选参数
        HashMap<String, Object> options = new HashMap<String, Object>();
        // 音频格式
        options.put("format", "mp3-16k"); //  "mp3-16k"，"mp3-48k"，"wav"，"pcm-8k"，"pcm-16k"，默认为mp3-16k
        // 音库
        options.put("voice", 0); // 可选，具体参考文档
        // 语速
        options.put("speed", 5); // 取值0-15，默认为5中语速
        // 语调
        options.put("pitch", 5); // 取值0-15，默认为5中语调
        // 音量
        options.put("volume", 5); // 音量，基础音库取值0-9，精品音库取值0-15，默认为5中音量（取值为0时为音量最小值，并非为无声）
        // 是否开启字幕时间戳
        options.put("enable_subtitle", 0); // 取值范围0, 1, 2，默认为0。0表示不开启字幕时间戳，1表示开启句级别字幕时间戳，2表示开启词级别字幕时间戳
        // 段落间隔
        options.put("break", 0); // 取值 0-5000 ，单位ms，用于合成文本分段传入时设置段落间间隔。

        JSONObject res = client.longTextSynthesisCreate(texts, "zh", options);

        System.out.println(res);
    }
}
```

在上面代码中，常量APP_ID在百度云控制台中创建，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。

配置AipSpeech

如果用户需要配置AipSpeech的一些细节参数，可以在构造AipSpeech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

长文本合成

详细文档参考官网：https://cloud.baidu.com/doc/SPEECH/s/ulbxh8rbu

接口描述

长文本在线合成接口可以将10万字以内文本一次性合成，**异步返回音频**。支持多种优质音库，将超长文本快速转换成稳定流畅、饱满真实的音频。适用于阅读听书、新闻播报等客户。

步骤：

1、根据文本内容、音频格式、音库等参数创建语音合成任务，获取task_id参数。

2、根据task_id的数组批量查询语音合成任务结果。

一、长文本合成-创建任务

请求说明

```
 package org.example;
import com.baidu.aip.speech.AipSpeech;
import org.json.JSONObject;

import java.util.Arrays;
import java.util.HashMap;
import java.util.List;

public class LongTtsCreate {
    // 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AipSpeech
        AipSpeech client = new AipSpeech(APP_ID, API_KEY, SECRET_KEY);
        List<String> texts = Arrays.asList(
                "你好百度，",
                "你好世界。"
        );

        // 可选参数
        HashMap<String, Object> options = new HashMap<String, Object>();
        // 音频格式
        options.put("format", "mp3-16k"); //  "mp3-16k"，"mp3-48k"，"wav"，"pcm-8k"，"pcm-16k"，默认为mp3-16k
        // 音库
        options.put("voice", 0); // 可选，具体参考文档
        // 语速
        options.put("speed", 5); // 取值0-15，默认为5中语速
        // 语调
        options.put("pitch", 5); // 取值0-15，默认为5中语调
        // 音量
        options.put("volume", 5); // 音量，基础音库取值0-9，精品音库取值0-15，默认为5中音量（取值为0时为音量最小值，并非为无声）
        // 是否开启字幕时间戳
        options.put("enable_subtitle", 0); // 取值范围0, 1, 2，默认为0。0表示不开启字幕时间戳，1表示开启句级别字幕时间戳，2表示开启词级别字幕时间戳
        // 段落间隔
        options.put("break", 0); // 取值 0-5000 ，单位ms，用于合成文本分段传入时设置段落间间隔。

        JSONObject res = client.longTextSynthesisCreate(texts, "zh", options);

        System.out.println(res);
    }
}
```

参数详解

参数名称	类型	是否必需	说明
text	list	是	待合成文本，需要为UTF-8编码；输入多段文本时，文本间会插入1s长度的空白间隔 总字数不超过10万个字符，1个中文字、英文字母、数字或符号均算作1个字符
format	string	否	音频格式，取值为"mp3-16k"，"mp3-48k"，"wav"，"pcm-8k"，"pcm-16k"，默认为mp3-16k
voice	int	是	发音人选择，具体值参考音色列表
speed	int	否	语速，取值 0-15，默认为 5 中语速
pitch	int	否	音调，取值 0-15，默认为 5 中语调
volume	int	否	音量，基础音库取值0-9，其他音库取值 0-15，默认为 5 中音量 取值为0时为音量最小值，并非为无声
enable_subtitle	int	否	是否开启字幕时间戳，取值范围0, 1, 2，默认为0。 0表示不开启字幕时间戳，1表示开启句级别字幕时间戳，2表示开启词级别字幕时间戳 若希望自由控制长文本切分与返回时间戳，可使用^p进行分割（注意分割句间最好不要长于200字），此功能与句级、词级字幕时间戳功能不冲突
break	int	否	段落间隔，取值 0-5000，单位ms，用于合成文本分段传入时设置段落间间隔。

返回说明

参数详解

参数名称	类型	是否必需	对外状态
log_id	int	是	log id
task_id	str	否	任务id。**注意保存该id，用于后续请求结果**
task_status	str	否	任务状态
error_code	int	否	错误码，详细解释见错误码汇总
error_msg	str	否	错误信息，详细解释见错误码汇总

二、长文本合成 - 查询任务

请求说明

```java
package org.example;
import com.baidu.aip.speech.AipSpeech;
import org.json.JSONObject;

import java.util.Arrays;
import java.util.List;

public class LongTtsQuery {
    // 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AipSpeech
        AipSpeech client = new AipSpeech(APP_ID, API_KEY, SECRET_KEY);
        // 任务id集合
        // 推荐一次查询多个任务id，单次最多可查询200个
        List<String> taskIds = Arrays.asList(
                "task_id_1",
            "task_id_2"
        );
        JSONObject res = client.longTextSynthesisQuery(taskIds);
        System.out.println(res);
    }
}
```

参数名称	类型	是否必需	说明
task_ids	int	是	任务id，推荐一次查询多个任务id，单次最多可查询200个

返回说明

参数名	类型	是否必需	描述
log_id	int	是	log id
tasks_info	list	否	任务信息
+task_id	str	是	任务id
+task_status	str	是	任务状态
+task_result	dict	否	任务结果
++speech_url	str	否	音频下载链接，任务完成后储存72小时
++speech_timestamp	dict	否	字幕时间戳信息
+++sentences	list	否	句子列表
++++paragraph_index	int	否	句子在传入文本中段落位置
++++sentence_texts	str	否	句子文本信息
++++begin_time	int	否	句子文本在合成音频的开始时间戳
++++end_time	int	否	句子文本在合成音频的结束时间戳
++++characters	list	否	句子中各个词的信息
+++++character_text	str	否	句子中的词文本
+++++begin_time	int	否	句子中的词在合成音频的开始时间戳
+++++end_time	int	否	句子中的词在合成音频的结束时间戳
++err_no	int	否	错误码，详细解释见错误码汇总
++err_msg	str	否	错误信息，详细解释见错误码汇总
++sn	str	否	
error_code	int	否	错误码
error_msg	str	否	错误信息
error_info	list	否	错误的或查询不存在taskId数组

离线音色列表

页面描述 本页面为离线SDK调用发音人所属音库、发音人名称、调用per参数等信息。

试听demo 试听demo请点击

各音库适用版本说明 |音库名称|版本号|适用序列号|适用离线合成SDK版本|下载链接| |-|-|-|---|-| |基础音库资源|Speech v4.9 Text v6.0.0/5.5.0|2025年11月27日及之后购买或申请的序列号|适配语音合成SDK版本：
安卓v6.2.7.53及以后的版本
iOS v4.2.0.26及以后的版本
鸿蒙v1.0.8及以后的版本
Linux v1.0.0版本|包含在最新离线合成SDK中，详见语音技术控制台| |精品音库资源|Speech v4.9 Text v6.0.0/5.5.0|2025年11月27日及之后购买或申请的序列号|适配语音合成SDK版本：
安卓v6.2.7.53及以后的版本
iOS v4.2.0.26及以后的版本
鸿蒙v1.0.8及以后的版本
Linux v1.0.0版本|点击下载精品音库| |往期基础音库资源|Speech v4.6 Text v6.0.0/5.5.0|2025年11月27日之前购买或申请的序列号|适配语音合成SDK版本：
安卓v6.2.7.53及以后版本
iOS v4.2.0.26及以后版本
鸿蒙v1.0.0及以后的版本
Linux v1.0.0版本

更早版本SDK已不再维护，请升级至更高级的版本，点击前往语音技术控制台下载最新版本SDK|点击下载往期基础音库| |往期精品音库资源|Speech v4.6
Text v6.0.0/5.5.0|2025年11月27日之前购买或申请的序列号|适配语音合成SDK版本：
安卓v6.2.7.53及以后版本
iOS v4.2.0.26及以后版本
鸿蒙v1.0.0及以后的版本
Linux v1.0.0版本
更早版本SDK已不再维护，请升级至更高级的版本，点击前往语音技术控制台下载最新版本SDK|点击下载往期精品音库|

音库与SDK选择说明：请根据获取序列号的时间选择对应音库与SDK，建议使用最新版SDK。当前官网无法直接申请或购买适用于4.6音库的序列号，若希望获得可以联系售后获取。

例如：

- 在2025年5月申请了安卓端的序列号，请点击下载v4.6的基础与精品音库，只需下载最新版本安卓SDK即可，若您预下载的SDK符合范围也可继续使用
- 在2026年4月购买了iOS端的序列号，请点击下载v4.9精品音库与最新版本iOS SDK
- 在2025年8月购买了Linux端的序列号，请点击下载v4.6的基础与精品音库，并下载最新版本Linux SDK

各音库对应资源内容 最新音库资源 基础音库：最新版SDK默认自带以下发音人与文本资源 |资源文件|具体文件名（支持新序列号）| |-|-| |m15 离线男声（度小宇）|bd_etts_common_speech_duxiaoyu_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat| |f7 离线女声（度小美）|bd_etts_common_speech_duxiaomei_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat| |yy 离线度逍遥|bd_etts_common_speech_duxiaoyao_mand_eng_high_am-tac-csubgan16k_v4.9.0_20251110_20251210203318.dat| |c1 离线度丫丫|bd_etts_common_speech_duyaya_mand_eng_high_am-tac-csubgan16k_v4.9.0_20220419_20251031153737.dat| |中文离线文本模型|bd_etts_common_text_txt_all_mand_eng_middle_big_v6.0.0_20240731.dat
bd_etts_common_text_txt_all_mand_eng_middle_big_v5.5.0_20250410.dat（仅适用于鸿蒙SDK）|

精品音库：需要单独购买与下载 |资源文件|具体文件名| |-|-| |f4 离线度小娇|bd_etts_common_speech_duxiaojiao_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat| |c3 离线度米朵|bd_etts_common_speech_dumiduo_mand_eng_high_am-tac-csubgan16k_v4.9.0_20220419_20251031153737.dat| |wyg 离线度博文|bd_etts_common_speech_dubowen_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat| |c4 离线度小童|bd_etts_common_speech_duxiaotong_mand_eng_high_am-tac-csubgan16k_v4.9.0_20251028_20251031153737.dat| |f8 离线度小萌|bd_etts_common_speech_duxiaomeng_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat| |f12dt 度小乔|bd_etts_common_speech_duxiaoqiao_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat| |f17 度小鹿|bd_etts_common_speech_duxiaolu_mand_eng_high_am-tac-csubgan16k_v4.9.0_20240918_20251031153737.dat| |f10tw 度小台|bd_etts_common_speech_duxiaotai_mand_eng_high_am-tac-csubgan16k_v4.9.0_20251028_20251031153737.dat| |m8 度小贤|bd_etts_common_speech_duxiaoxian_mand_eng_high_am-tac-csubgan16k_v4.9.0_20251028_20251031153737.dat| |gezi 度小雯|bd_etts_common_speech_duxiaowen_mand_eng_high_am-tac-csubgan16k_v4.9.0_20230927_20251031153737.dat| |粤语离线文本模型|bd_etts_common_text_txt_all_cant_eng_middle_big_v6.0.0_20220916.dat
bd_etts_common_text_txt_all_cant_eng_middle_big_v4.5.0_20211222.dat（仅适用于鸿蒙SDK）| |f13can 度小粤|bd_etts_common_speech_duxiaoyue_cant_eng_high_am-tac-csubgan16k_v4.9.0_20220419_20251031153737.dat| |fnat 度小译|bd_etts_common_speech_duxiaoyi_mand_eng_high_am-tac-csubgan16k_v4.9.0_20251028_20251031153737.dat|

往期音库资源 往期基础音库 |资源文件|具体文件名| |---|------| |m15 离线男声（度小宇）|bd_etts_common_speech_duxiaoyu_mand_eng_high_am-style24k_v4.6.0_20210721_20220822104311.dat| |f7 离线女声（度小美）|bd_etts_navi_speech_duxiaomei_mand_eng_high_am-style24k_v4.6.0_20210721.dat

或bd etts navi_speech_f7_mand eng high_am-style24k_v4.6.0_20210721.dat| |yy 离线度逍遥|bd_etts_common_speech_duxiaoyao_mand_eng_high_am-style24k_v4.6.0_20210721_20220822104311.dat| |c1 离线度丫丫|bd_etts_common_speech_duyaya_mand_eng_high_am-style24k_v4.6.0_20210721_20220822104311.dat| |中文离线文本模型|bd_etts_common_text_txt_all_mand_eng_middle_big_v6.0.0_20240731.dat
bd_etts_common_text_txt_all_mand_eng_middle_big_v5.5.0_20250410.dat（仅适用于鸿蒙SDK）|

往期精品音库 |**资源文件**|**具体文件名**| |-|-| |f4 离线度小娇|bd_etts_common_speech_f4_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |c3 离线度米朵|bd_etts_navi_speech_c3_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |wyg 离线度博文|bd_etts_navi_speech_wyg_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |c4 离线度小童|bd_etts_common_speech_c4_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |f8 离线度小萌|bd_etts_navi_speech_f8_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |f12dt 度小乔|bd_etts_navi_speech_f12dt_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |f17 度小鹿|bd_etts_navi_speech_f17_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |f10tw 度小台|bd_etts_navi_speech_f10tw_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |m8 度小贤|bd_etts_navi_speech_m8_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |gezi 度小雯|bd_etts_navi_speech_gezi_mand_eng_high_am-style24k_v4.6.0_20210721.dat| |粤语离线文本模型|bd_etts_common_text_txt_all_cant_eng_middle_big_v6.0.0_20220916.dat
bd_etts_common_text_txt_all_cant_eng_middle_big_v4.5.0_20211222.dat（仅适用于鸿蒙SDK）| |f13can 度小粤|bd_etts_navi_speech_f13can_cant_eng_high_am-style24k_v4.6.0_20210721.dat| |英文离线文本模型|bd_etts_common_text_txt_all_mand_eng_middle_big_v6.0.0_20240731.dat
bd_etts_common_text_txt_all_mand_eng_middle_big_v5.5.0_20250410.dat（仅适用于鸿蒙SDK）| |fnat 度小译|bd_etts_common_speech_fnat_mand_eng_high_am-style24k_v4.9.0_20211130.dat|

注：若您希望上线支持范围内的其他离线音色，可以提需求工单或者联系商务。

语音识别

短语音识别-HTTP-SDK

简介

简介

目前本SDK的功能同REST API，需要联网调用http接口，具体功能见REST API 文档，REST API 仅支持整段语音识别的模式，即需要上传完整语音文件进行识别，时长不超过60s，支持自定义词库设置， **没有其他额外功能**。

接口能力

接口名称	接口能力简要描述
语音识别	将人类的语音中的词汇内容转换为计算机可读的输入，例如按键、二进制编码或者字符序列

支持的语音格式

原始 PCM 的录音参数必须符合 16k 、8k采样率、16bit 位深、单声道，支持的格式有：pcm（不压缩）、wav（不压缩，pcm编码）、amr（压缩格式）。

注意事项

如果需要使用实时识别、长语音、唤醒词、语义解析等其它语音功能，请使用Android或者iOS SDK 或 Linux C++ SDK 等。

1. 请严格按照文档里描述的参数进行开发，特别请关注原始录音参数以及语音压缩格式的建议，否则会影响识别率，进而影响到产品的用户体验。
2. 目前系统支持的语音时长上限为60s，请不要超过这个长度，否则会返回错误。

反馈

在百度云控制台内提交工单，咨询问题类型请选择**人工智能服务**；

版本更新记录

Python

上线日期	版本号	更新内容
2021.2.26	4.15.4	接口统一升级
2017.5.11	1.0.0	语音识别服务上线

Java

上线日期	版本号	更新内容
2021.2.26	4.15.4	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	更新sdk打包方式：所有AI服务集成一个SDK
2017.7.14	1.0.1	更新sdk打包方式
2017.6.30	1.0.0	新增语音识别

C++

上线日期	版本号	更新内容
2021.2.26	4.15.4	接口统一升级
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	在线语音识别第一版

PHP

上线日期	版本号	更新内容
2021.2.26	4.15.4	接口统一升级
2017.5.11	1.0.0	语音识别服务上线

Node

上线日期	版本号	更新内容
2021.2.26	4.15.4	接口统一升级
2017.12.21	2.0.0	实现代码重构，接口返回标准promise对象
2017.7.13	1.2.1	修复模块加载在linux下失败的问题
2017.6.30	1.2.0	新增语音识别接口

C#

上线日期	版本号	更新内容
2021.2.26	4.15.4	接口统一升级
2018.4.2	3.4.0	新增.Net Core支持；新增超时参数
2018.1.11	3.3.1	细节修改
2017.9.12	3.0.0	更新SDK打包方式：所有AI服务集成一个SDK
2017.6.30	1.0.0	新增语音识别服务接口

短语音识别REST-API-PythonSDK

安装语音识别 Python SDK

语音识别 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
│    └──── speech.py //语音识别
└──── 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AipSpeech

AipSpeech是语音识别的Python SDK客户端，为使用语音识别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AipSpeech：

```
 from aip import AipSpeech

""" 你的 APPID AK SK """
APP_ID = '你的 App ID'
API_KEY = '你的 Api Key'
SECRET_KEY = '你的 Secret Key'

client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)
```

在上面代码中，常量`APP_ID`在百度云控制台中创建，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。

配置AipSpeech

如果用户需要配置AipSpeech的网络请求参数(一般不需要配置)，可以在构造AipSpeech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）

语音识别

接口描述

向远程服务上传整段语音进行识别

请求说明

举例，要对段保存有一段语音的语音文件进行识别：

```
 # 读取文件
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 识别本地文件
client.asr(get_file_content('audio.pcm'), 'pcm', 16000, {
    'dev_pid': 1537,
})
```

接口函数说明：

参数	类型	描述	是否必须
speech	Buffer	建立包含语音内容的Buffer对象, 语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写	是
format	String	语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写。推荐pcm文件	是
rate	int	采样率，16000、8000，固定值	是
cuid	String	用户唯一标识，用来区分用户，填写机器 MAC 地址或 IMEI 码，长度为60以内	否
dev_pid	Int	不填写lan参数生效，都不填写，默认1537（普通话 输入法模型），dev_pid参数见下面的表格	否
lan(已废弃)	String	历史兼容参数，请使用dev_pid。如果dev_pid填写，该参数会被覆盖。语种选择,输入法模型，默认中文（zh）。 中文=zh、粤语=ct、英文=en，不区分大小写。	否

dev_pid 参数列表

dev_pid	语言	模型	是否有标点	备注
1537	普通话(纯中文识别)	语音近场识别模型	有标点	支持自定义词库
1737	英语	英语模型	无标点	不支持自定义词库
1637	粤语	粤语模型	有标点	不支持自定义词库
1837	四川话	四川话模型	有标点	不支持自定义词库

返回数据参数详情

参数	类型	是否一定输出	描述
err_no	int	是	错误码
err_msg	int	是	错误码描述
sn	int	是	语音数据唯一标识，系统内部产生，用于 debug
result	int	是	识别结果数组，提供1-5 个候选结果，string 类型为识别的字符串， utf-8 编码

返回样例：

```
 // 成功返回
{
    "err_no": 0,
    "err_msg": "success.",
    "corpus_no": "15984125203285346378",
    "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D",
    "result": ["北京天气"]
}

// 失败返回
{
    "err_no": 2000,
    "err_msg": "data empty.",
    "sn": null
}
```

短语音识别REST-API-JavaSDK

安装Speech Java SDK

Speech 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
    ├──── speech
    │       └──── AipSpeech       //AipSpeech类
    └──── util                    //工具类
```

支持 JAVA版本：1.7+

查看源码

Java SDK代码现已公开，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/java-sdk

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.在点击下载下载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` `log4j-1.2.17.jar`。

其中，`version`为版本号，添加完成后，用户就可以在工程中使用Speech Java SDK。

新建AipSpeech

AipSpeech是语音识别的Java客户端，为使用语音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AipSpeech,初始化完成后建议**单例使用**,避免重复获取access_token：

```
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AipSpeech
    AipSpeech client = new AipSpeech(APP_ID, API_KEY, SECRET_KEY);

    // 可选：设置网络连接参数
    client.setConnectionTimeoutInMillis(20000);
    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
    JSONObject res = client.asr("test.pcm", "pcm", 16000, null);
    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。

配置AipSpeech

如果用户需要配置AipSpeech的一些细节参数，可以在构造AipSpeech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）

语音识别

接口描述

向远程服务上传整段语音进行识别

请求说明

举例：

```
  public void asr(AipSpeech client)
{
    // 对本地语音文件进行识别
    String path = "D:\\code\\java-sdk\\speech_sdk\\src\\test\\resources\\16k_test.pcm";
    JSONObject asrRes = client.asr(path, "pcm", 16000, null);
    System.out.println(asrRes);

    // 对语音二进制数据进行识别
    byte[] data = Util.readFileByBytes(path);    //readFileByBytes仅为获取二进制数据示例
    JSONObject asrRes2 = client.asr(data, "pcm", 16000, null);
    System.out.println(asrRes2);

}
```

接口函数说明：

```
  // 语音识别
JSONObject asr(String path, String format, int rate, HashMap<String, Object> options);
JSONObject asr(byte[] data, String format, int rate, HashMap<String, Object> options);
```

参数	类型	描述	是否必须
path/data	String/byte[]	语音文件所在路径或二进制数据, 语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写	是
format	String	语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写。推荐pcm文件	是
rate	int	采样率，16000、8000，固定值	是
cuid	String	用户唯一标识，用来区分用户，填写机器 MAC 地址或 IMEI 码，长度为60以内	否
dev_pid	Int	不填写lan参数生效，都不填写，默认1537（普通话 输入法模型），dev_pid参数见下面的表格	否
lan(已废弃)	String	历史兼容参数，请使用dev_pid。如果dev_pid填写，该参数会被覆盖。语种选择,输入法模型，默认中文（zh）。 中文=zh、粤语=ct、英文=en，不区分大小写。	否

dev_pid 参数列表

dev_pid	语言	模型	是否有标点	备注
1537	普通话(纯中文识别)	语音近场识别模型	有标点	支持自定义词库
1737	英语	英语模型	无标点	不支持自定义词库
1637	粤语	粤语模型	有标点	不支持自定义词库
1837	四川话	四川话模型	有标点	不支持自定义词库

返回数据参数详情

参数	类型	是否一定输出	描述
err_no	int	是	错误码
err_msg	int	是	错误码描述
sn	int	是	语音数据唯一标识，系统内部产生，用于 debug
result	int	是	识别结果数组，提供1-5 个候选结果，string 类型为识别的字符串， utf-8 编码

返回样例：

```
 // 成功返回
{
 "err_no": 0,
 "err_msg": "success.",
 "corpus_no": "15984125203285346378",
 "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D",
 "result": ["北京天气"]
}

// 失败返回
{
 "err_no": 2000,
 "err_msg": "data empty.",
 "sn": null
}
```

短语音识别REST-API-PHPSDK

安装语音识别 PHP SDK

语音识别 PHP SDK目录结构

```
├──── AipSpeech.php          //语音识别
└──── 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.在点击下载下载php SDK压缩包。

2.将下载的`aip-php-sdk-version.zip`解压后，复制AipSpeech.php以及lib/*到工程文件夹中。

3.引入AipSpeech.php

新建AipSpeech

AipSpeech是语音识别的PHP SDK客户端，为使用语音识别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AipSpeech：

```
 require_once 'AipSpeech.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 AipSpeech(APP_ID, API_KEY, SECRET_KEY);
```

在上面代码中，常量`APP_ID`在百度云控制台中创建，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。

配置AipSpeech

如果用户需要配置AipSpeech的网络请求参数(一般不需要配置)，可以在构造AipSpeech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）

语音识别

接口描述

向远程服务上传整段语音进行识别

请求说明

举例，要对段保存有一段语音的语音文件进行识别：

```
 // 识别本地文件
$client->asr(file_get_contents('audio.pcm'), 'pcm', 16000, array(
    'dev_pid' => 1537,
));
```

接口函数说明：

参数	类型	描述	是否必须
speech	Buffer	建立包含语音内容的Buffer对象, 语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写	是
format	String	语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写。推荐pcm文件	是
rate	int	采样率，16000、8000，固定值	是
cuid	String	用户唯一标识，用来区分用户，填写机器 MAC 地址或 IMEI 码，长度为60以内	否
dev_pid	Int	不填写lan参数生效，都不填写，默认1537（普通话 输入法模型），dev_pid参数见下面的表格	否
lan(已废弃)	String	历史兼容参数，请使用dev_pid。如果dev_pid填写，该参数会被覆盖。语种选择,输入法模型，默认中文（zh）。 中文=zh、粤语=ct、英文=en，不区分大小写。	否

dev_pid 参数列表

dev_pid	语言	模型	是否有标点	备注
1537	普通话(纯中文识别)	语音近场识别模型	有标点	支持自定义词库
1737	英语		无标点	不支持自定义词库
1637	粤语		有标点	不支持自定义词库
1837	四川话		有标点	不支持自定义词库

返回数据参数详情

参数	类型	是否一定输出	描述
err_no	int	是	错误码
err_msg	int	是	错误码描述
sn	int	是	语音数据唯一标识，系统内部产生，用于 debug
result	int	是	识别结果数组，提供1-5 个候选结果，string 类型为识别的字符串， utf-8 编码

返回样例：

```
 // 成功返回
{
    "err_no": 0,
    "err_msg": "success.",
    "corpus_no": "15984125203285346378",
    "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D",
    "result": ["北京天气"]
}

// 失败返回
{
    "err_no": 2000,
    "err_msg": "data empty.",
    "sn": null
}
```

短语音识别REST-API-C#SDK

安装语音识别 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

语音识别 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.在点击下载下载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.Speech.Asr是语音识别的交互类，为使用语音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 Speech.Asr(APP_ID, 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。

语音识别

接口描述

向远程服务上传整段语音进行识别

请求说明

举例：

```
 // 识别本地文件
public void AsrData()
{
    var data = File.ReadAllBytes("语音pcm文件地址");

    // 可选参数
    var options = new Dictionary<string, object>
    {
      {"dev_pid", 1537}
    };
    client.Timeout = 120000; // 若语音较长，建议设置更大的超时时间. ms
    var result = client.Recognize(data, "pcm", 16000, options);
    Console.Write(result);
}
```

接口函数说明：

参数	类型	描述	是否必须
data	byte[]	语音二进制数据, 语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写	是
format	String	语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写。推荐pcm文件	是
rate	int	采样率，16000、8000，固定值	是
cuid	String	用户唯一标识，用来区分用户，填写机器 MAC 地址或 IMEI 码，长度为60以内	否
dev_pid	Int	默认1537（普通话 输入法模型）。dev_pid 必须为整数类型。参数可选值见REST API文档说明	否
否			
lan(已废弃)	String	历史兼容参数，请使用dev_pid。如果dev_pid填写，该参数会被覆盖。语种选择,输入法模型，默认中文（zh）。 中文=zh、粤语=ct、英文=en，不区分大小写。	否

dev_pid 参数列表

dev_pid	语言	模型	是否有标点	备注
1537	普通话(纯中文识别)	语音近场识别模型	有标点	支持自定义词库
1737	英语	英语模型	无标点	不支持自定义词库
1637	粤语	粤语模型	有标点	不支持自定义词库
1837	四川话	四川话模型	有标点	不支持自定义词库

返回数据参数详情

参数	类型	是否一定输出	描述
err_no	int	是	错误码
err_msg	int	是	错误码描述
sn	int	是	语音数据唯一标识，系统内部产生，用于 debug
result	int	是	识别结果数组，提供多个候选结果，无论返回多少个请取第一个

返回样例：

```
 // 成功返回
{
 "err_no": 0,
 "err_msg": "success.",
 "corpus_no": "15984125203285346378",
 "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D",
 "result": ["北京天气"]
}

// 失败返回
{
 "err_no": 2000,
 "err_msg": "data empty.",
 "sn": null
}
```

短语音识别REST-API-NodeSDK

安装语音识别 Node SDK

语音识别 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
├─── AipSpeech.js               //语音识别交互类
├─── 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.在点击下载下载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语音识别 Node SDK。

直接使用npm安装依赖：

`npm install baidu-aip-sdk`

6.调用示例可以参考 https://github.com/Baidu-AIP/sdk-demo

新建AipSpeechClient

AipSpeechClient是语音识别的node客户端，为使用语音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AipSpeechClient：

```
 var AipSpeechClient = require("baidu-aip-sdk").speech;

// 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 AipSpeechClient(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。

语音识别

接口描述

向远程服务上传整段语音进行识别

请求说明

举例，要对段保存有一段语音的语音文件进行识别：

```
  let fs = require('fs');
let voice = fs.readFileSync('assets/voice/16k_test.pcm');

let voiceBuffer = new Buffer(voice);

// 识别本地文件
client.recognize(voiceBuffer, 'pcm', 16000).then(function (result) {
    console.log('<recognize>: ' + JSON.stringify(result));
}, function(err) {
    console.log(err);
});

// 识别本地文件，附带参数
client.recognize(voiceBuffer, 'pcm', 16000， {dev_pid: 1537, cuid: Math.random()}).then(function (result) {
    console.log('<recognize>: ' + JSON.stringify(result));
}, function(err) {
    console.log(err);
});
```

接口函数说明：

```
  // 识别本地语音文件
client.recognize(speech, format, rate， {lan: 'ct', cuid: Math.random(), ptc: 1}});
```

参数	类型	描述	是否必须
speech	Buffer	建立包含语音内容的Buffer对象, 语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写	是
format	String	语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写。推荐pcm文件	是
rate	int	采样率，16000、8000，固定值	是
cuid	String	用户唯一标识，用来区分用户，填写机器 MAC 地址或 IMEI 码，长度为60以内	否
dev_pid	Int	不填写lan参数生效，都不填写，默认1537（普通话 输入法模型），dev_pid参数见下面的表格	否
lan(已废弃)	String	历史兼容参数，请使用dev_pid。如果dev_pid填写，该参数会被覆盖。语种选择,输入法模型，默认中文（zh）。 中文=zh、粤语=ct、英文=en，不区分大小写。	否

dev_pid 参数列表

dev_pid	语言	模型	是否有标点	备注
1537	普通话(纯中文识别)	语音近场识别模型	有标点	支持自定义词库
1737	英语	英语模型	无标点	不支持自定义词库
1637	粤语	粤语模型	有标点	不支持自定义词库
1837	四川话	四川话模型	有标点	不支持自定义词库

返回数据参数详情

参数	类型	是否一定输出	描述
err_no	int	是	错误码
err_msg	int	是	错误码描述
sn	int	是	语音数据唯一标识，系统内部产生，用于 debug
result	int	是	识别结果数组，提供1-5 个候选结果，string 类型为识别的字符串， utf-8 编码

返回样例：

```
 // 成功返回
{
  "err_no": 0,
  "err_msg": "success.",
  "corpus_no": "15984125203285346378",
  "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D",
  "result": ["北京天气"]
}

// 失败返回
{
  "err_no": 2000,
  "err_msg": "data empty.",
  "sn": null
}
```

短语音识别REST-API-C++SDK

安装语音识别 C++ SDK

语音识别 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
└─── speech.h                   // 语音识别 交互类
```

最低支持 C++ 11+

直接使用开发包步骤如下：

1.在点击下载下载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 speech.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。

6.调用示例可以参考 https://github.com/Baidu-AIP/sdk-demo

新建client

client是语音识别的C++客户端，为使用语音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 "speech.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::Speech client(app_id, api_key, secret_key);
```

在上面代码中，常量APP_ID在百度云控制台中创建，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。

语音识别

接口描述

向远程服务上传整段语音进行识别

请求说明

举例：

```
 void asr(aip::Speech client)
{
    // 无可选参数调用接口
    std::string file_content;
    aip::get_file_content("./assets/voice/16k_test.pcm", &file_content);
    Json::Value result = client.recognize(file_content, "pcm", 16000, aip::null);

    // 极速版调用函数
    // Json::Value result = client.recognize_pro(file_content, "pcm", 16000, aip::null);

    // 如果需要覆盖或者加入参数
    std::map<std::string, std::string> options;
    options["dev_pid"] = "1537";
    Json::Value result = client.recognize(file_content, "pcm", 16000, options);
}
```

接口函数说明：

参数	类型	描述	是否必须
data	byte[]	语音二进制数据, 语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写	是
format	String	语音文件的格式，pcm 或者 wav 或者 amr。不区分大小写。推荐pcm文件	是
rate	int	采样率，16000，固定值	是
cuid	String	用户唯一标识，用来区分用户，填写机器 MAC 地址或 IMEI 码，长度为60以内	否
dev_pid	int	不填写lan参数生效，都不填写，默认1537（普通话 输入法模型），dev_pid参数见下面的表格	否
lm_id	int	自训练平台模型id，填dev_pid = 8001 或 8002生效	选填

dev_pid 参数列表

dev_pid	语言	模型	是否有标点	备注
1537	普通话(纯中文识别)	语音近场识别模型	有标点	支持自定义词库
1737	英语	英语模型	无标点	不支持自定义词库
1637	粤语	粤语模型	有标点	不支持自定义词库
1837	四川话	四川话模型	有标点	不支持自定义词库

返回数据参数详情

参数	类型	是否一定输出	描述
err_no	int	是	错误码
err_msg	int	是	错误码描述
sn	int	是	语音数据唯一标识，系统内部产生，用于 debug
result	int	是	识别结果数组，提供1-5 个候选结果，string 类型为识别的字符串， utf-8 编码

返回样例：

```
 // 成功返回
{
    "err_no": 0,
    "err_msg": "success.",
    "corpus_no": "15984125203285346378",
    "sn": "481D633F-73BA-726F-49EF-8659ACCC2F3D",
    "result": ["北京天气"]
}

// 失败返回
{
    "err_no": 2000,
    "err_msg": "data empty.",
    "sn": null
}
```

语音识别Android SDK

1. 文档说明

文档名称	语音识别集成文档
所属平台	Android
提交日期	2026-01-04
概述	本文档是百度语音开放平台Android SDK的用户指南，描述了**短语音识别、离线自定义命令词识别、语音唤醒、语义解析与对话管理**等相关接口的使用说明。SDK内部均为采用流式协议，即用户边说边处理。区别于Restapi需要上传整个录音文件。

2. 版本说明

名称	版本号
语音识别	3.5.1
系统支持	android 5.1+
架构支持	支持armeabi-v7a，armeabi-v8a架构
硬件要求	要求设备上有麦克风
网络	支持移动网络（不包括2G、3G）、WIFI等网络环境
开发环境	建议使用最新版本Android Studio 进行开发

3. SDK说明&功能简介

文件名称	版本号	MD5值	资源描述
bdasr_aipd_V3_20250717_1e379e2.aar	3493171	3fc9823ddd4e927b04641603e4537b08	jar 库 ；约16M

语音识别Android SDK 功能 主要分为语音识别 及 语义理解与对话管理：

1. 语音识别： 将录音转为文字。目前在线识别支持普通话、英文、粤语和四川话。
2. 语义理解与对话管理： 提取语音识别出的文字的意图与关键信息，并做出回应。

语音识别 语音识别，可以分为在线识别，离线命令词，及唤醒词

1. 在线识别： 即联网使用的识别功能，支持自定义词库及自训练平台。目前在线识别支持普通话、英文、粤语和四川话，通过在请求时配置不同的language参数，选择对应模型。默认为麦克风输入，可以设置参数为pcm格式16k采样率，16bit，小端序，单声道的音频流输入。
2. 离线命令词： 断网时识别固定的预定义短语（定义在bsg文件中），SDK强制优先使用在线识别。 断网时激活，只能识别预定义的短语。联网时，强制使用在线识别。固定短语的语法需要从控制台“离线词&本地语义”模块预定义并下载为baidu_speech_grammar.bsg文件
3. 唤醒词：识别预定义的“关键词”， 这个“关键词”必须在一句话的开头。 本地功能，不需要网络。唤醒词即识别“关键词”，当SDK的识别引擎“听到”录音中的关键词后，立即告知用户。与android系统的锁屏唤醒完全无关。关键词和离线命令词一样，需要预定义并下载为WakeUp.bin文件

在线识别 在线是指手机联网时(3G 4G 5G wifi)，在线识别可以分为：

1. 在线普通识别： 流式识别出识别用户输入的录音音频流，支持普通话、英文、粤语和四川话。限制60s时长。
2. 在线长语音识别： 在线普通识别的基础上，没有60s时长的限制。 在线识别可以测试DEMO中的第一个按钮“在线识别”。

自定义词库

设置方法：

登录百度云-管理中心“管理应用”“选择应用”“语音识别词库设置”右侧“进行设置”；

设置效果：

1. 可以自定义识别词，提升准确率。
2. 仅在普通话输入法模型下生效。
3. 自定义词库适合短句，保证词库中一模一样的短句可以被识别出。
4. 词库中的分词优先级较高。

举例 词库定义了1个短句 :1 .“摆渡船来了”百度内部处理的可能的分词结果：摆渡船来 了。

以下录音的结果

1. 原始音频：摆渡船来了 =》识别结果： 摆渡船来了 【保证结果】

2. 原始音频：摆渡船来了么 =》识别结果： 百度传来了么 【可能结果，不保证】

3. 原始音频：摆渡船来 =》 识别结果： 百度传来 【可能结果，不保证】

4. 原始音频：百度传来了喜讯 =》 识别结果： 摆渡船来了喜讯 【不保证，词库内的分词优先级高】

最好在1万行以内。

副作用：如果用户的测试集中包含大量非自定义词表的query，整体上准确率下降。

语音自训练平台模型训练

自训练平台可以认为是自定义词库的升级版本，可以更加直观地查看训练效果。同样使用您自定义的文本进行结果优化。

具体功能及使用说明请参考文档 “自训练平台手册”

离线命令词 离线命令词，联网时，强制使用在线识别，在断网时或在线请求超时时，使用离线命令词功能。离线命令词功能不支持任意语句的识别，只能识别预定义的固定短语。

唤醒词 唤醒词即识别预定义的“关键词”。与在线长语音识别不同，长语音识别会返回所有识别结果，唤醒词只会识别出您预先定义的关键词。与android本身的锁屏唤醒没有任何关系。

唤醒词是本地功能，正常使用时无需联网。 在语音唤醒页面下方可以自行定义bin文件。百度语音提供了近15个预定义唤醒词，效果有优化。也可以自定义唤醒词，效果不如预定义唤醒词。bin文件中最多可以有10个唤醒词，其中自定义唤醒词不超过3个，并且2个字的预定义唤醒词不超过3个。**进行唤醒词操作前必须要有相对静音**。

语义 语义包括理解与对话管理，可用于提取语音识别出的文字的意图与关键信息，并做出回应。语音识别Android SDK提供了本地语义的方式：

本地语义：

- 在任意网络下都可使用，android sdk内部根据设置的bsg文件的定义进行离线命令词语义解析

录音环境 百度短语音识别（含唤醒）要求安静的环境，真人的正常语速的日常用语，并且不能多个人同时发音。

音频格式要求:

默认为麦克风输入，可以设置参数为**pcm**格式**16k**采样率，**16bit，小端序，单声道**的音频流输入。

以下场景讲会导致识别或者唤醒效果变差，错误，甚至没有结果：

1. 吵杂的环境

2. 有背景音乐，包括扬声器在播放百度合成的语音。

以下场景的录音可能没有正确的识别结果：

1. 音频里有技术专业名称或者用语 （技术专业名称请到自训练平台改善）

2. 音频里是某个专业领域的对话，非日常用语。比如专业会议，动画片等

建议先收集一定数量的真实环境测试集，按照测试集评估及反馈。语音识别没有降噪功能无法过滤背景音等杂声。

4.输入参数

识别输入参数 场景：

- 在线识别：百度语音服务器将录音识别出文字，包括长语音

- 离线命令词：离线识别出预定义的固定短语
- 本地语义：在识别出文字的基础上（包括离线命令词识别），对文字做语义分析。任意网络状况。

使用网络状况：

- 离线 ： 涵盖离线命令词，及离线命令词识别后的本地语义解析。

共支持4个语种 ，语种请在 ASR_START输入事件中的language参数中设置

- 中文普通话 （全部场景）
- 中文四川话（离线命令词及语义不支持）
- 粤语（离线命令词及语义不支持）
- 英语（离线命令词及语义不支持）

识别输入事件

以下参数均为SpeechConstant类的常量，如SpeechConstant.ASR_START, 实际的String字面值可以参见SpeechConstant类或自行打印

事件名	类型	值	场景	描述
ASR_START	String(JSON结构的字符串)	json内的参数见下文“ASR_START 参数”	全部	开始一次识别。 注意不要连续调用ASR_START参数。下次调用需要在CALLBACK_EVENT_ASR_EXIT回调后，或者在ASR_CANCEL输入后。
ASR_STOP			全部	停止录音
ASR_CANCEL			全部	取消本次识别
ASR_KWS_LOAD_ENGINE	String (JSON结构的字符串)	json内的参数见下文“ASR_KWS_LOAD_ENGINE 参数”		离线命令词
ASR_KWS_UNLOAD_ENGINE			离线命令词	高级

ASR_START 输入事件参数

事件参数	类型/值	场景	常用程度	描述
APP_ID	String	全部	推荐	开放平台创建应用后分配的鉴权信息，在**AuthUtil**工具类中填入自己的鉴权信息，填写方式仅供测试使用， 上线后请使用此参数填写鉴权信息。（新版创建AipeEventManagerFactory时传入鉴权信息情况下无需设置该参数，具体可参考9.1在线识别调用流程）
APP_KEY	String	全部	推荐	开放平台创建应用后分配的鉴权信息，在**AuthUtil**工具类中填入自己的鉴权信息，填写方式仅供测试使用， 上线后请使用此参数填写鉴权信息。（新版创建AipeEventManagerFactory时传入鉴权信息情况下无需设置该参数，具体可参考9.1在线识别调用流程）
SECRET	String	全部	推荐	开放平台创建应用后分配的鉴权信息，在**AuthUtil**工具类中填入自己的鉴权信息，填写方式仅供测试使用， 上线后请使用此参数填写鉴权信息。（新版创建AipeEventManagerFactory时传入鉴权信息情况下无需设置该参数，具体可参考9.1在线识

				别调用流程）
LANGUAGE	String	在线	常用	默认"cmn-Hans-CN"，即中文输入法模型。LANGUAGE具体值及说明见下一个表格。 其中输入法模型是指适用于长句的输入法模型模型适用于短语。
LM_ID	int	在线	常用	自训练平台上线后的模型Id，必须和自训练平台的LANGUAGE连用。
DECODER	int	全部	常用	离在线的并行策略
	0 （默认）	在线		纯在线(默认)
	2	离线		离在线融合(在线优先)，离线命令词功能需要开启这个选项。
VAD	String	全部	高级	语音活动检测， 根据静音时长自动断句。注意不开启长语音的情况下，SDK只能录制60s音频。长语音请设置BDS_ASR_ENABLE_LONG_SPEECH参数
	VAD_DNN（默认）		高级	新一代VAD，各方面信息优秀，推荐使用。
	VAD_TOUCH		不常用	关闭语音活动检测。注意关闭后不要开启长语音。适合用户自行控制音频结束，如按住说话松手停止的场景。功能等同于60s限制的长语音。需要手动调用ASR_STOP停止录音
BDS_ASR_ENABLE_LONG_SPEECH	boolean	全部	常用	是否开启长语音。 即无静音超时断句，开启后需手动调用ASR_STOP停止录音。 请勿和VAD=touch联用！优先级大于VAD_ENDPOINT_TIMEOUT 设置
	true			开启长语音
	false（默认）			关闭长语音，无法识别超过60s的语音
VAD_ENDPOINT_TIMEOUT	int	全部	高级	静音超时断句及长语音
	0	在线	常用	开启长语音。即无静音超时断句。手动调用ASR_STOP停止录音。 请勿和VAD=touch联用！
	800ms	全部	高级	开启长语音。即开启静音超时断句。开启VAD尾点检测，即静音判断的毫秒数。
IN_FILE	String：文件路径 资源路径或回调方法名	全部	高级	该参数支持设置为： a. pcm文件，系统路径，如：/sdcard/test/test.pcm；音频pcm文件不超过3分钟b. pcm文件, JAVA资源路径，如：res:///com/baidu.test/16k_test.pcm；音频pcm文件不超过3分钟c. InputStream数据流，#方法全名的字符串，格式如：”#com.test.Factory.create16KInputStream()”（解释：Factory类中存在一个返回InputStream的方法create16kInputStream()），注意：必须以井号开始；方法原型必须为：public static InputStream create16KInputStream()。 超过3分钟的录音文件，请在每次read中sleep，避免SDK内部缓冲不够。d.wav文件，建议8k采样率，设置前需要调用convertWavTo16kPcm音频转换接口 转换为16k pcm文件 后设置该路径，详情请见：**9.1 音频格式转换**
OUT_FILE	String：文件路径	全部	高级	保存识别过程产生的录音文件, 该参数需要开启ACCEPT_AUDIO_DATA后生效
AUDIO_MILLS	int：毫秒	全部	高级	录音开始的时间点。用于唤醒+识别连续说。SDK有15s的录音缓存。如设置为(System.currentTimeMillis() - 1500),表示回溯1.5s的音频。

NLU	String	本地语义	高级	本地语义解析设置。必须设置ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数生效，无论网络状况，都可以有本地语义结果。
	disable（默认）		高级	禁用
	enable		高级	启用
	enable-all		不常用	在enable的基础上，临时结果也会做本地语义解析
ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH	String：文件路径支持assets路径	本地语义	高级	用于支持本地语义解析的bsg文件，离线和在线都可使用。NLU开启生效，其它说明见NLU参数。注意bsg文件同时也用于ASR_KWS_LOAD_ENGINE离线命令词功能。
SLOT_DATA	String（JSON格式）	本地语义	高级	与ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数一起使用后生效。 用于代码中动态扩展本地语义bsg文件的词条部分（bsg文件下载页面的左侧表格部分），具体格式参见代码示例或者demo
DISABLE_PUNCTUATION	boolean	在线	不常用	在选择"cmn-Hans-CN"开头的LANGUAGE（输入法模式）的时候，是否禁用标点符号
	true			禁用标点
	false（默认）			不禁用标点，无法使用本地语义
PUNCTUATION_MODE	int	在线	不常用	在选择"cmn-Hans-CN"开头的LANGUAGE（输入法模式）的时候，标点处理模式。需要设置DISABLE_PUNCTUATION为fasle生效
	0 (默认)	在线		打开后处理标点
	1	在线		关闭后处理标点
	2	在线		删除句末标点
	3	在线		将所有标点替换为空格
ACCEPT_AUDIO_DATA	boolean	全部	高级	是否需要语音音频数据回调，开启后有CALLBACK_EVENT_ASR_AUDIO事件
	true			需要音频数据回调
	false（默认）			不需要音频数据回调
ACCEPT_AUDIO_VOLUME	boolean	全部	高级	是否需要语音音量数据回调，开启后有CALLBACK_EVENT_ASR_VOLUME事件回调
	true（默认）			需要音量数据回调

	认）			
	false			不需要音量数据回调
SOUND_START	int：资源ID	全部	不常用	说话开始的提示音
SOUND_END	int：资源ID	全部	不常用	说话结束的提示音
SOUND_SUCCESS	int：资源ID	全部	不常用	识别成功的提示音
SOUND_ERROR	int：资源ID	全部	不常用	识别出错的提示音
SOUND_CANCEL	int：资源ID	全部	不常用	识别取消的提示音
SAMPLE_RATE	int	全部	基本不用	采样率 ，固定及默认值16000
	16000（默认）			
ASR_OFFLINE_ENGINE_LICENSE_FILE_PATH	String：文件路径 ，支持assets路径	离线命令词	基本不用	临时授权文件路径。SDK在联网时会获取自动获取离线正式授权。有特殊原因可用在官网创建应用时下载通用临时授权文件。临时授权文件测试期仅有15天，不推荐使用。使用正式授权时请确认官网应用设置的包名与APP自身的包名相一致。目前离线命令词和唤醒词功能需要使用正式授权。

ASR_KWS_LOAD_ENGINE 输入事件参数

|事件参数|类型|值|场景|常用程度|描述|

事件参数	类型	值	场景	常用程度	描述
SLOT_DATA	String	JSON格式	本地语义	高级	与ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数一起使用后生效。 用于代码中动态扩展离线命令词bsg文件的词条部分（bsg文件下载页面的左侧表格部分），具体格式参见代码示例或者demo
DECODER	int	2			固定值：2，离在线的并行策略
ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH	String	文件路径，支持assets路径			用于支持离线命令词（同时也是本地语义）解析的bsg文件，离线断网时可以使用。NLU开启生效，其它说明见NLU参数。注意bsg文件同时也用于ASR_KWS_LOAD_ENGINE离线命令词功能。语义解析设置，在线使用时，会在识别结果的文本基础上同时输出语义解析的结果（该功能需要在官网的应用里设置“语义解析设置”）。

LANGUAGE 参数

在线参数， 请根据语言，来选择LANGUAGE。

- 语言：目前支持中文普通话，四川话，粤语，和英语四个

- 输入法模型：适用于较长的句子输入。默认有标点; 开启标点后，不支持本地语义。
- 自训练平台模型： 在输入法模型的基础上，可以自行上传词库和句库，生成您自己的训练模型。

语音识别LANGUAGE

LANGUAGE	语言	模型	是否有标点	备注
"cmn-Hans-CN"	普通话	语音近场识别模型	有标点（逗号）	默认LANGUAGE
"en-GB"	英语		无标点	
"yue-Hans-CN"	粤语		有标点（逗号）	
"sichuan-Hans-CN"	四川话		有标点（逗号）	

语音自训练平台

语音自训练平台上，训练实时语音识别基础模型，可以在Android SDK上添加训练上线的模型ID。注意必须填写上线模型给出的LANGUAGE。示例 `获取专属模型参数LANGUAGE:"cmn-Hans-CN" modelid:1235`， 可在SDK参数中填写 LANGUAGE="cmn-Hans-CN"；同时LM_ID 设置为1235。

具体功能及使用说明请参考文档 “自训练平台手册”

LANGUAGE	语言	模型	是否有标点
"cmn-Hans-CN"	普通话	输入法模型	逗号

具体功能参数

鉴权信息

描述:APP_ID, APP_KEY, SECRET3个鉴权信息决定是否拥有识别、唤醒、离线命令词等权限

请求示例：设置APP_ID, APP_KEY, SECRET

```
{"appid":155905xx,"secret":"y7S7hAI894BB3LF1yHYmvQEusxxxxxx ","key":"q2uPyBe6LmWTZlvb0gxxxxxx","accept-audio-volume":false}
```

设置语言、模型、自定义词库、语义

描述:LANGUAGE参数决定调用哪个识别模型，即决定了语言，模型。

1. 语言共有4种：中文普通话、英语、粤语及四川话。

2. 输入法模型：分别适用于长句输入/短句输入。

3. 在输入法模型下自定义词库生效。

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 在线识别=>设置=>LANGUAGE,语种

请求示例： 英语 输入法模型

```
{"LANGUAGE":"en-GB","accept-audio-volume":false}
```

长语音

描述:开启长语音识别功能，此时VAD参数不能设为touch；长语音可以识别数小时的音频。注意选输入法模型。

BDS_ASR_ENABLE_LONG_SPEECH= true 或 VAD_ENDPOINT_TIMEOUT = 0 都可以开启长语音

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 在线识别=>设置=>长语音及VAD时长设置

请求参数示例： 开启长语音，不能与VAD=touch联用

```
 {"enable.long.speech":true,"accept-audio-volume":false}
或
{"accept-audio-volume":false,"vad.endpoint-timeout":0}
```

设置静音时长进行断句

描述：普通识别的录音限制60s。连续xxxms静音后，SDK认为一句话结束。**VAD_ENDPOINT_TIMEOUT=0**，作用是静音断句的时长设置，使用长语音功能时不能使用这个参数。

VAD_ENDPOINT_TIMEOUT =0

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 在线识别=>设置=>长语音及VAD时长设置

请求示例： 连续800ms静音，表示一句话结束

```
{"accept-audio-volume":false,"vad.endpoint-timeout":800}
```

开启/关闭根据静音时长切分句子

描述：默认开启根据静音时长切分句子。通过设置VAD参数的值，当设置VAD=dnn时，表示开启VAD，此时通过设置的静音时长进行断句；开启长语音功能时，VAD必然是开启的状态；当设置**VAD=touch**时，表示关闭VAD，不能使用长语音功能，限制录音时长60s，在60s内的只能点击停止按钮通过发送停止事件才能停止识别。

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 在线识别=>设置=>VAD是否开启dnn或touch

请求示例： 开启VAD，根据静音时长切分句子

```
{"accept-audio-volume":false,"vad":"dnn"}
```

请求示例： 关闭VAD，60s内音频，SDK等待调用stop事件结束录音

```
{"accept-audio-volume":false,"vad":"touch"}
```

离线命令词

设置纯在线功能和离在线融合识别

DECODER 参数

- DECODER = 0 ，表示禁用离线功能，只使用在线功能；
- DECODER = 2 ，表示启用离线功能，但是SDK强制在线优先。

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 离线命令词功能=>开始

请求示例：DECODET=2，表示断网时，使用离线识别固定短语；

```
{"accept-audio-volume":false,"decoder":2}
```

设定离线命令词文件路径

离线命令词只能识别bsg文件中预定义的固定短语，其中bsg文件必须在项目中设定路径，参数**ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH**作用就是定义离线命令词识别的bsg文件路径;该参数需要在**ASR_KWS_LOAD_ENGINE**加载离线资源输入事件中初始化；

ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH设置bsg文件路径

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击 离线命令词功能

请求示例：定义assets目录下的离线命令词文件

```
{"accept-audio-data":false,"grammar":"assets:\/\/\/baidu_speech_grammar.bsg","decoder":2}
```

扩展离线命令词的词条部分

离线命令词bsg文件的左侧词条部分内容可通过**SLOT_DATA**参数进行动态覆盖，覆盖后原先的bsg文件中的左侧词条部分失效。和ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH一起使用。

SLOT_DATA扩展词条

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：点击全部识别=>设置=>纯在线或在线+离线命令词=>离线命令词及在线识别混合 =>离线命令词及本地语义 =>扩展词条

请求示例：动态修改离线命令词的词条部分

```
{"accept-audio-data":false,"grammar":"asset:\/\/\/baidu_speech_grammar.bsg","decoder":2, "slot-data":{"name":\["妈妈","老伍"\],"appname":\["手百","度秘""\]}}
```

唤醒输入参数

以下参数均为SpeechConstant类的常量，如SpeechConstant.WAKEUP_START

事件名	类型	值	描述
WAKEUP_START	json格式的字符串	json内的参数见下文“WAKEUP_START 参数”	开始识别唤醒词
WAKEUP_STOP			停止识别唤醒词

WAKEUP_START 输入事件参数

事件参数	类型	常用程度	描述
APP_ID	String	推荐	开放平台创建应用后分配的鉴权信息，填写后会覆盖 AndroidManifest.xml中定义的。AndroidManifest.xml填写方式仅供测试使用， 上线后请使用此参数填写鉴权信息。
APP_KEY	String	推荐	开放平台创建应用后分配的鉴权信息，填写后会覆盖 AndroidManifest.xml中定义的。AndroidManifest.xml填写方式仅供测试使用， 上线后请使用此参数填写鉴权信息。
SECRET	String	推荐	开放平台创建应用后分配的鉴权信息，填写后会覆盖 AndroidManifest.xml中定义的。AndroidManifest.xml填写方式仅供测试使用， 上线后请使用此参数填写鉴权信息。
WP_WORDS_FILE	String	常用	唤醒词bin文件路径，支持android asset目录（如assets:///wakeUp.bin)
IN_FILE	String：文件路径 资源路径或回调方法名	全部	该参数支持设置为： a. pcm文件，系统路径，如：/sdcard/test/test.pcm；音频pcm文件不超过3分钟b. pcm文件, JAVA资源路径，如：res:///com/baidu.test/16k_test.pcm；音频pcm文件不超过3分钟c. InputStream数据流，#方法全名的字符串，格式 如："#com.test.Factory.create16KInputStream()"（解释：Factory类中存在一个返回InputStream的方法create16kInputStream()） ，注意：必须以井号开始；方法原型必须为：public static InputStream create16KInputStream()。 超过3分钟的录音文件，请在每次read中sleep，避免SDK内部缓冲不够。d.wav文件，建议8k采样率，设置前需要调用convertWavTo16kPcm音频转换接口转换为16k pcm文件 后设置该路径，详情请见：**9.1 音频格式转换**
ACCEPT_AUDIO_DATA	boolean	基本不用	默认关闭。开启后，会有音频回调（CALLBACK_EVENT_WAKEUP_AUDIO），很占资源
WP_ENGINE_LICENSE_FILE_PATH	string	基本不用	不填写，在联网时会获取自动获取离线正式授权。有特殊原因可用在官网下载临时授权文件，配置此参数，支持android asset目录（如assets:///mylicense.dat)
SAMPLE_RATE	int	基本不用	16000（默认值，且唤醒仅支持16k采样）

具体功能参数

设置唤醒词文件路径

唤醒词功能是本地功能使用时无需联网，只能唤醒bin文件中预定义的关键词，bin文件需要使用**WP_WORDS_FILE**设置路径，支持android asset目录（如assets:///wakeUp.bin)；

WP_WORDS_FILE设置bin文件路径

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

demo测试：精简版唤醒

请求示例：使用唤醒功能加载唤醒词bin文件；

```
{"kws-file":"assets:\/\/\/WakeUp.bin","accept-audio-volume":false}
```

语义输入参数

- 请求参数说明

字段	是否必选	类型	备注
bot_session_list	是	string（json）	至少有1个元素
bot_session_list[].bot_session_id	是	string	技能的session信息，由系统自动生成，client从上轮resonpse中取出并直接传递。如果为空，则表示清空session（开发者判断用户意图已经切换且下一轮会话不需要继承上一轮会话中的词槽信息时可以把bot_session_id置空，从而进行新一轮的会话）

具体功能参数

本地语义

必须设置ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数生效，无论网络状况，都可以有本地语义结果。

NLU= enable表示开启本地语义

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

Demo测试：在线和本地语义=>设置=>本地语义解析enable && 本地语义文件 && 扩展词条

请求示例：设置本地语义的bsg文件路径，例如在asset目录下；

```
{"accept-audio-data":false,"grammar":"assets:\/\/\/baidu_speech_grammar.bsg","LANGUAGE":"cmn-Hans-CN", "slot-data":{"name":\["妈妈","老伍"\],"appname":\["手百","度秘""\]},"nlu":"enable"}
```

- 设置本地语义文件路径

使用本地语义需要设置bsg文件的路径，本地语义共用离线命令词的参数和文件。使用本地语义功能的时候，不需要作为ASR_KWS_LOAD_ENGINE的输入参数。但是需要作为**ASR_START**事件的输入参数。

ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数设置bsg文件路径。

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

Demo测试：在线和本地语义=>设置=>本地语义文件

请求示例：设置本地语义的bsg文件路径，例如在asset目录下；

```
{"accept-audio-data":false,"grammar":"assets:\/\/\/baidu_speech_grammar.bsg","LANGUAGE":"cmn-Hans-CN","nlu":"enable"}
```

- 扩展本地语义文件的词条部分内容

bsg文件的左侧词条部分内容可通过**SLOT_DATA**参数进行动态覆盖，覆盖后原先的bsg文件中的左侧词条部分失效。和ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH一起使用。

SLOT_DATA参数动态覆盖bsg文件词条内容

DEMO中测试方法：(具体可参见识别安卓SDK测试文档)

Demo测试：在线和本地语义设置本地语义文件，扩展词条

请求示例：动态覆盖bsg文件左侧词条内容

```
{"accept-audio-data":false,"grammar":"assets:\/\/\/baidu_speech_grammar.bsg","LANGUAGE":"cmn-Hans-CN", "slot-data":{"name":\["妈妈","老伍"\],"appname":\["手百","度秘"\]},"nlu":"enable"}
```

5.输出参数

识别输出参数

语音回调事件统一由 public void onEvent(String name, String params, byte[] data, int offset, int length) 该方法回调。其中name是回调事件， params是回调参数。（data，offset，length）缓存临时数据，三者一起，生效部分为 data[offset] 开始，长度为length。

事件名（name）	事件参数	类型	值	描述
CALLBACK_EVENT_ASR_READY				引擎准备就绪，可以开始说话
CALLBACK_EVENT_ASR_BEGIN				检测到第一句话说话开始。SDK只有第一句话说话开始的回调，没有长语音每句话说话结束的回调。
CALLBACK_EVENT_ASR_END				检测到第一句话说话结束。SDK只有第一句话说话结束的回调，没有长语音每句话说话结束的回调。
CALLBACK_EVENT_ASR_PARTIAL	params	json		识别结果
	params[results_recognition]	String[]		解析后的识别结果。如无特殊情况，请取第一个结果
	params[result_type]	String	partial_result	**临时识别结果**
	params[result_type]	String	final_result	**最终结果**，长语音每一句都有一个最终结果
	params[result_type]	String	nlu_result	语义结果，在final_result后回调。语义结果的内容在(data，offset，length中）
	(data，offset，length）	String		语义结果的内容 ，当 params[result_type]=nlu_result时出现。
CALLBACK_EVENT_ASR_FINISH	params	String(json格式）		一句话识别结束（可能含有错误信息） 。最终识别的文字结果在ASR_PARTIAL事件中
	params[error]	int		错误领域
	params[sub_error]	int		错误码
	params[desc]	String		错误描述
CALLBACK_EVENT_ASR_LONG_SPEECH				长语音额外的回调，表示长语音识别结束。使用infile参数无此回调，请用ASR_EXIT 代替
CALLBACK_EVENT_ASR_EXIT				识别结束，资源释放
CALLBACK_EVENT_ASR_AUDIO	(data，offset，length)	byte[]		PCM音频片段 回调。必须输入ACCEPT_AUDIO_DATA 参数激活
CALLBACK_EVENT_ASR_VOLUME	params	json		当前音量回调。必须输入ACCEPT_AUDIO_VOLUME参数激活
	params[volume]	float		当前音量
	params[volume-percent]	int		当前音量的相对值（0-100）
CALLBACK_EVENT_ASR_LOADED				离线模型加载成功回调
CALLBACK_EVENT_ASR_UNLOADED				离线模型卸载成功回调

唤醒输出参数

语音回调事件统一由 public void onEvent(String name, String params, byte[] data, int offset, int length) 方法回调 其中name是回调事件， params是回调参数。（data，offset，length）缓存临时数据，三者一起，生效部分为 data[offset] 开始，长度为

length。

事件名	事件参数	类型	描述			
CALLBACK_EVENT_WAKEUP_STARTED			引擎开始运行			
CALLBACK_EVENT_WAKEUP_AUDIO	(data,offset,length)	byte[]	PCM音频片段回调，需要输入ACCEPT_AUDIO_DATA参数激活 。保存的pcm文件的采样率是16000，16bits，单声道，小端序。			
CALLBACK_EVENT_WAKEUP_SUCCESS			唤醒成功			
	errorCode		错误码,错误码为0表示唤醒成功，唤醒出错会在CALLBACK_EVENT_WAKEUP_ERROR 事件中			
	errorDesc		错误描述,此处固定为 success			
	word	String	具体的唤醒词			
CALLBACK_EVENT_WAKEUP_ERROR	params	String(json格式)	错误描述的回调	params[error]	int	错误码
	params[desc]	int	错误描述			
CALLBACK_EVENT_WAKEUP_STOPED			唤醒已关闭			

6. Demo运行

6.1 配置包名和签名

从百度云控制台下载Demo之后，需要在build.gradle中配置好包名。



6.2 修改鉴权信息

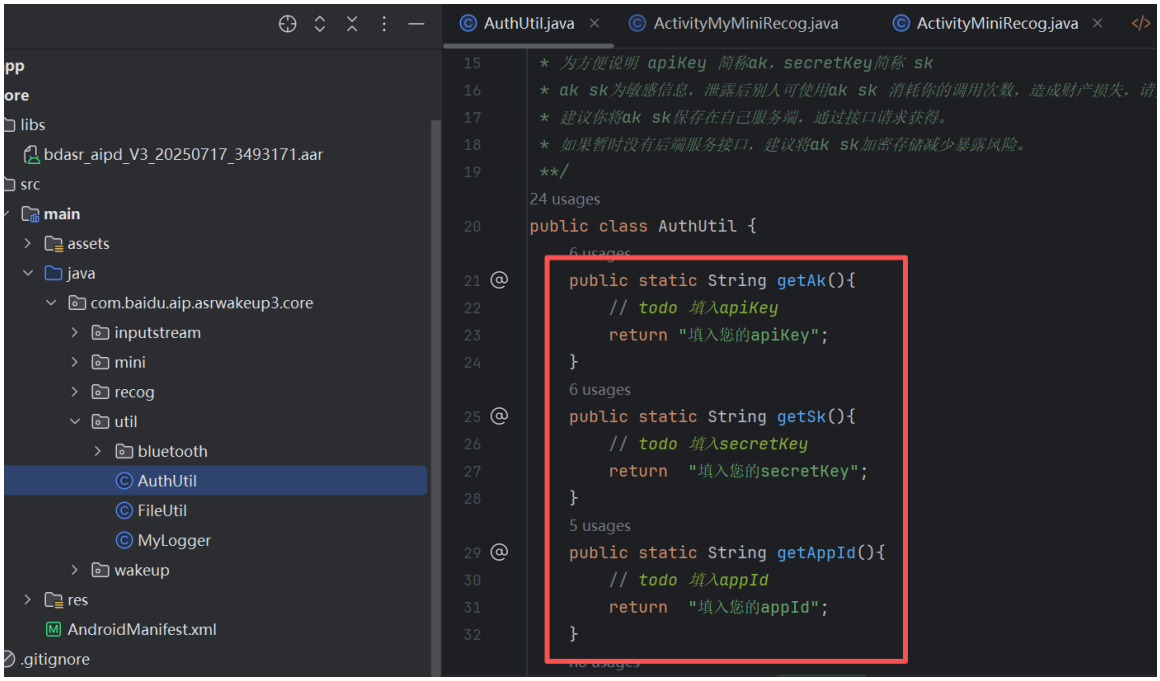
sdk分为4种激活方式，实际使用种选择其中一种方式进行激活，4种激活方法详细描述如下：

激活方式	说明
Appkey SecretKey激活	在百度云网站申请自己的语音合成应用后，会有appKey、appSecret鉴权信息，使用该方法进行在线鉴权，每次启动app需激活一次，不退出app永久有效
永久iamKey激活	需在官网申请的永久iamKey，使用该方法进行在线鉴权，每次启动app需激活一次，不退出app永久有效
accessToken激活	需在使用appkey和secretKey获取的access_token，使用access_token进行授权，该token存在时限，时间到期后需要从新获取新的token从新授权
临时iamKey激活	与accessToken类似，需获取临时iamKey，该key存在时限，时间到期后需要从新获取新的key从新授权

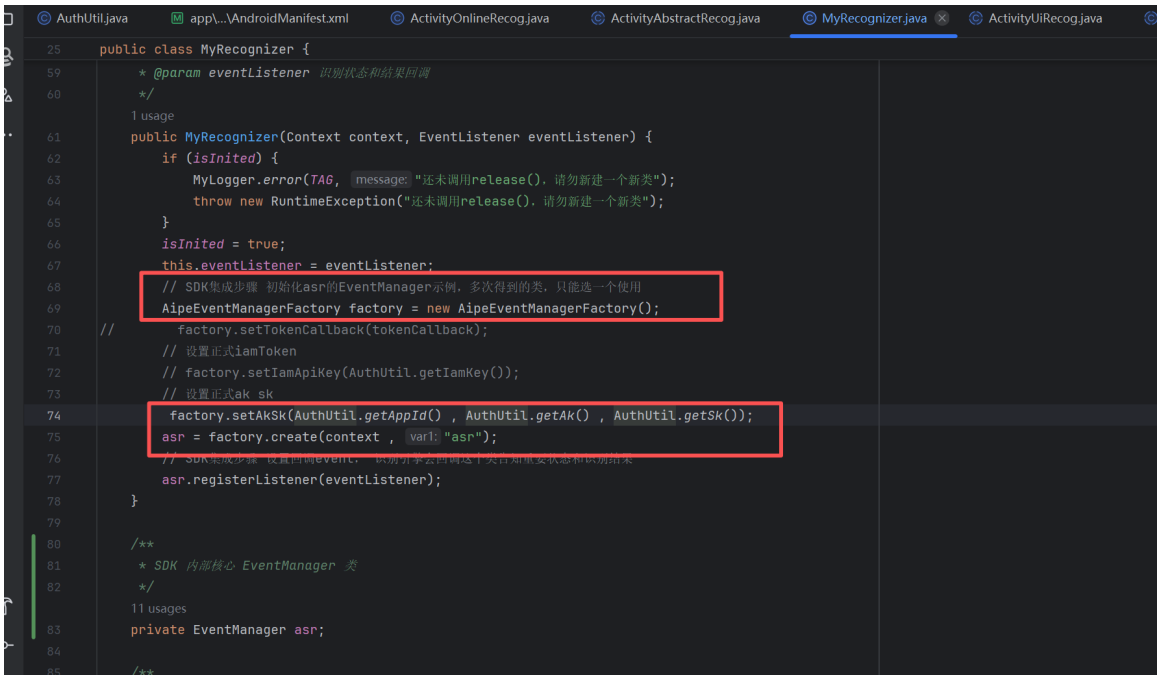
1.AppKey SecretKey激活：

百度云网站上申请自己语音识别应用，会有appId、appKey、appSecret及android包名 4个鉴权信息，申请地址为百度云应用列表

申请成功后在AuthUtil类中填写appid、apiKey和 secretKey 进行授权:



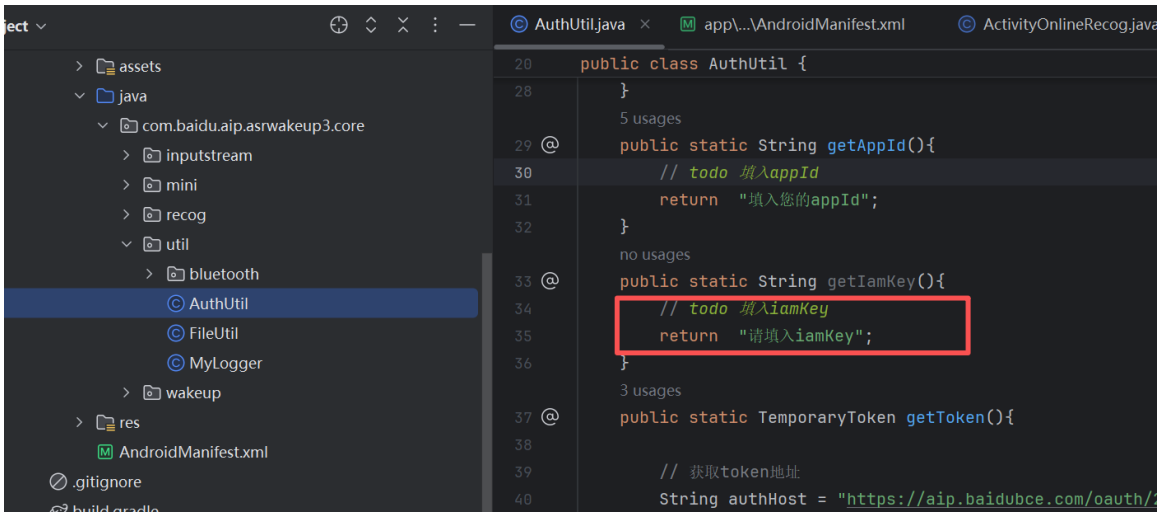
ak、sk、appid修改成功后，创建AipeEventManagerFactory传入ak、sk、appid并创建核心 EventManager 类



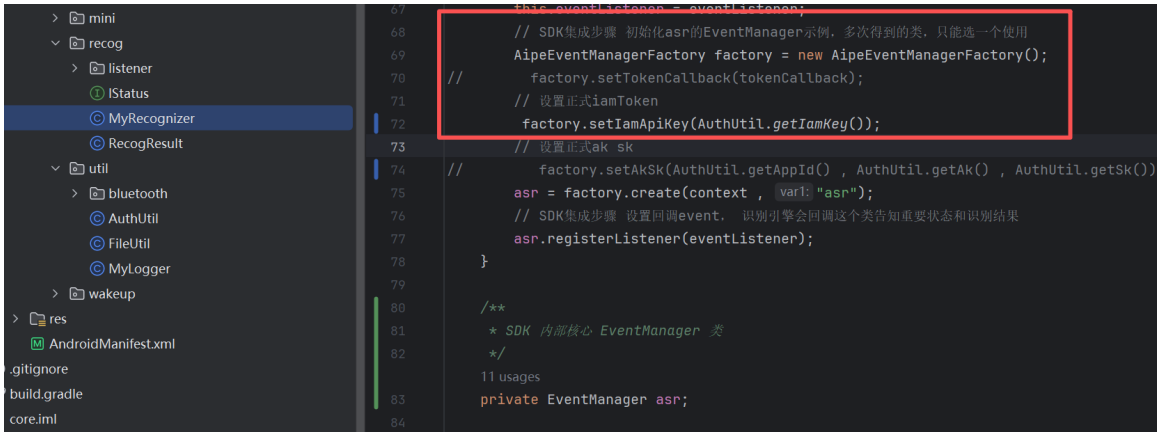
2.永久iamKey激活：

“永久iamKey激活”和“appKey secretKey激活”类似。

将官网申请的永久iamKey放到项目中



iamKey修改成功后，创建AipeEventManagerFactory传入iamKey创建核心 EventManager 类



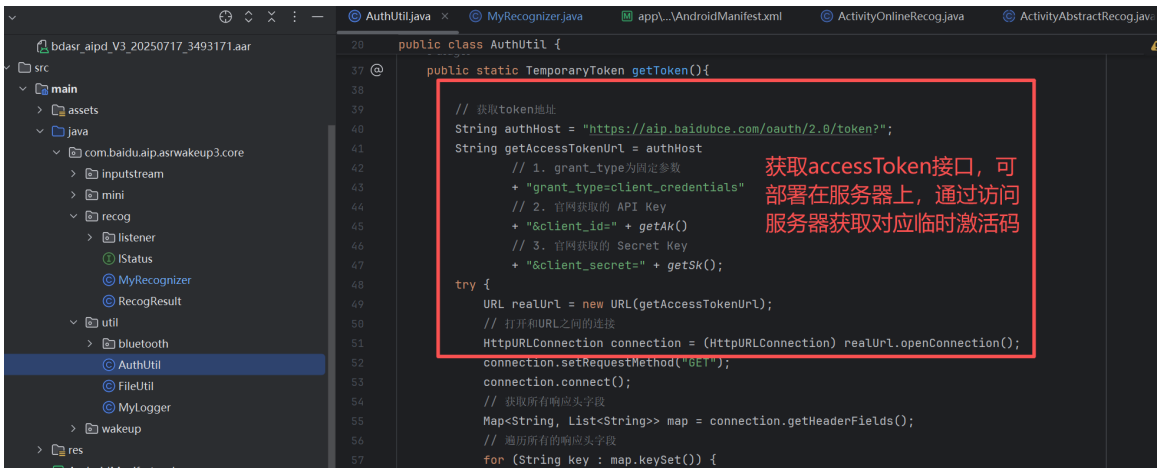
3.临时accessToken和临时iam key激活方式:

accessToken和临时iamKey为相同接口，接口会随时间失效，客户需要在callback中创建http请求获取到对应token返回给sdk,sdk会在过期前3分钟通过callback重新向客户端获取最新token。

accessToken获取地址：https://cloud.baidu.com/doc/AI_REFERENCE/s/um3zhy50e

iamKey获取地址：https://cloud.baidu.com/doc/AI_REFERENCE/s/Hm5us339w

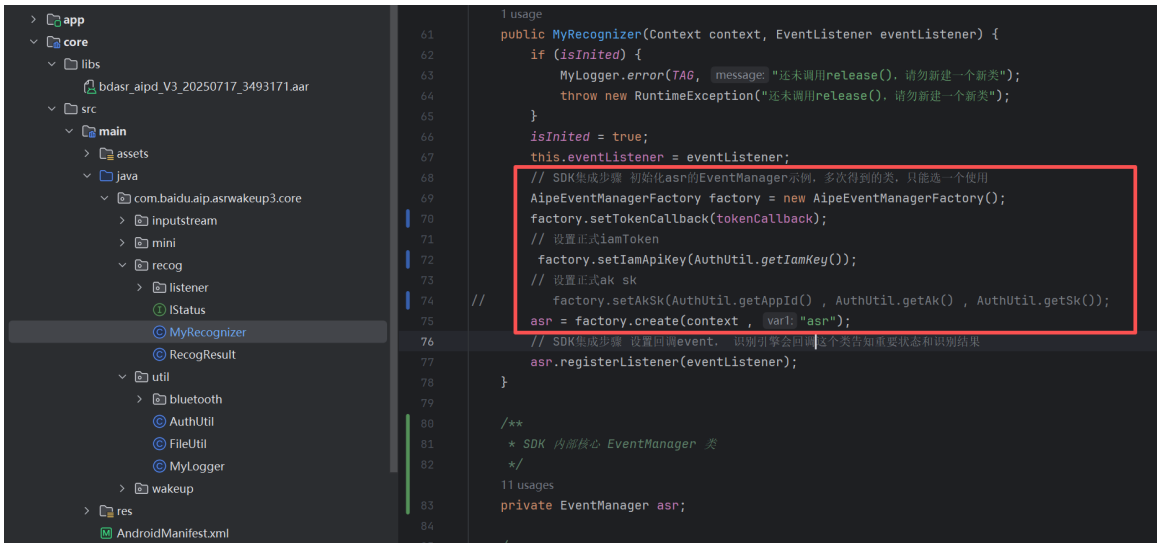
获取临时accessToken/iamKey:



创建token监听接口



修改成功后，创建AipeEventManagerFactory传入token监听接口创建核心 EventManager 类



然后编译运行



7. SDK集成

bdasr_V3_xxxxxxxx_xxxx.jar 库

将core/libs/bdasr_V3_xxxxx_xxxxx.jar 复制到您的项目的同名目录中。

复制NDK 架构目录

1. 将 core/src/main/jniLibs 下armeabi等包含so文件的5个目录，复制合并到您的项目的同名或者存放so的目录中。如果build.gradle中定义过jniLibs.srcDirs ，则复制合并到这个目录。
2. 如与第三方库集成，至少要保留armeabi目录。如第三方库有7个架构目录，比语音识别SDK多出2个目录 mips和mips64，请将mips和mips64目录删除，剩下5个同名目录合并。
3. 如第三方库仅有armeabi这一个目录，请将语音识别SDK的额外4个目录如armeabi-v7a删除,合并armeabi目录下的so。 即目录取交集，so文件不可随意更改所属目录。
4. 打包成apk文件，按照zip格式解压出libs目录可以验证。
5. 运行时 getApplicationInfo().nativeLibraryDir 目录下查看是否有完整so文件。 特别是**系统app**需要手动push so文件到这个目录下。

build.gradle 文件及包名确认

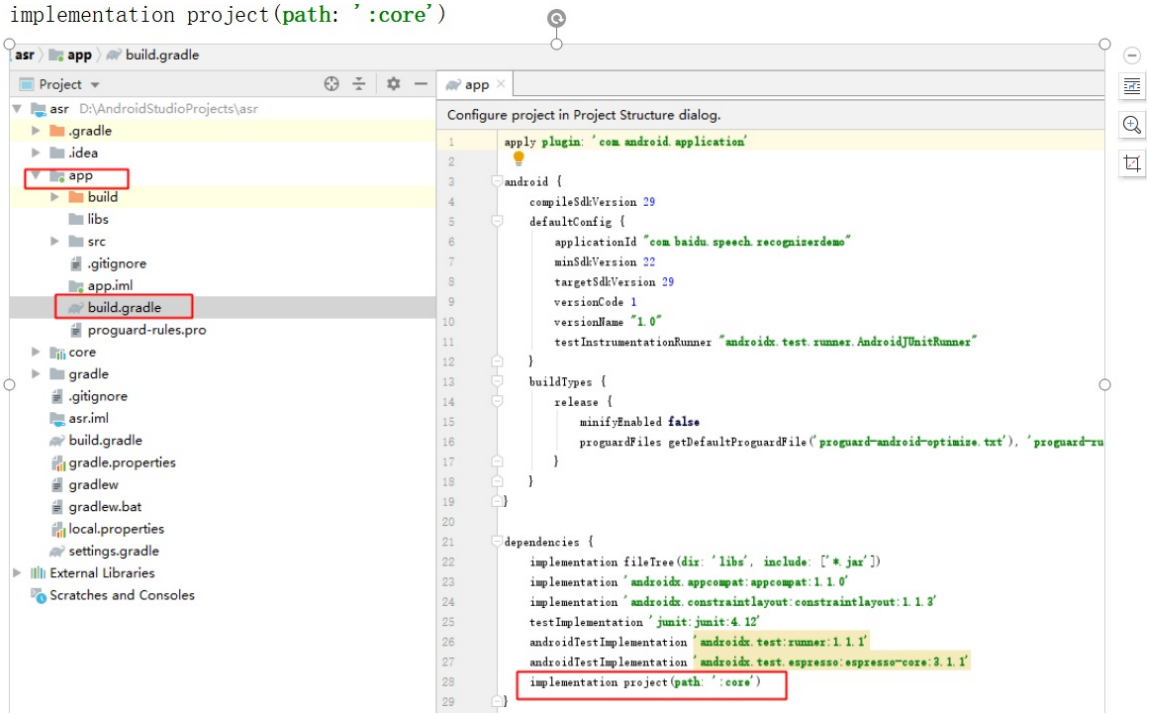
1. 根目录下build.gradle确认下gradle的版本。

2. app/build.gradle 确认下 applicationId 包名是否与官网申请应用时相一致（离线功能需要）。 demo的包名是"com.baidu.speech.recognizerdemo"。

3. 确认 compileSdkVersion buildToolsVersion 及 targetSdkVersion

4. 一、导入demo的core module，并配置app依赖core

```
implementation project(path: ':core')
```



二、修改app/java/com.baidu.speech.recognizerdemo/MainActivity.java，修改app/java/com.baidu.speech.recognizerdemo/MainActivity.java：

```
import com.baidu.aip.asrwakeup3.core.mini.ActivityMiniRecog;
public class MainActivity extends ActivityMiniRecog {

}
```



```
package com.baidu.speech.recognizerdemo;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import com.baidu.aip.asrwakeup3.core.mini.ActivityMiniRecog;

public class MainActivity extends ActivityMiniRecog {

}
```

具体参考官方demo内 doc_integration_DOCUMENT文件夹下有ASR-INTEGRATION-helloworld-V3.01.docx文件，helloworld 集成sdk的完整图示实例。

8. 相关授权文件

请将百度云控制台创建应用时获取的语音(APPID)、API/SECRET KEY 并填写包名。

在线识别与唤醒都需要进行相关验证后方可使用：

引擎类型	验证方法
在线识别	开放平台使用API/SECRET KEY + APPID进行验证
离线识别	使用APPID+包名首次联网自动下载授权文件进行验证
唤醒引擎	与离线识别验证方法一致

9. 语音相关接口流程

9.1在线识别调用流程

1 初始化

1.1创建AipeEventManagerFactory工厂对象，设置激活参数，创建EventManager对象

SDK中，eventManager通过工厂创建语音识别的事件管理器。注意识别事件管理器只能维持一个，请勿同时使用多个实例。即创建一个新的识别事件管理器后，之前的那个置为null，并不再使用。

```
 // 临时iam key 和 accessToken 接口回调，sdk会在剩余时间小于3分钟时重新调用该接口获取最新token
TokenCallback tokenCallback = new TokenCallback() {
    @Override
    public TemporaryToken getToken() {
        return AuthUtil.getToken();
    }

};
/**
 * SDK 内部核心 EventManager 类
 */
private EventManager asr;
public void init() {
    AipeEventManagerFactory factory = new AipeEventManagerFactory();
    // 临时iamToken 或 临时accessToken激活
    // factory.setTokenCallback(tokenCallback);
    // 正式iamToken激活
    // factory.setIamApiKey(AuthUtil.getIamKey());
    // 设置正式ak sk激活
    factory.setAkSk(AuthUtil.getAppId() , AuthUtil.getAk() , AuthUtil.getSk());
    )
    asr = factory.create(context , "asr");
}
```

xt类

> *详见ActivityMiniRecog类中"基于sdk集成1.1 初始化EventManager对象"*

1.2 自定义输出事件类

SDK中，需要实现EventListener的输出事件回调接口。该类需要处理SDK在识别过程中的回调事件。可以参考DEMO中对SDK的调用封装。

```
EventListener yourListener = new EventListener() {
  @Override
  public void onEvent(String name, String params, byte [] data, int offset, int length) {
    if(name.equals(SpeechConstant.CALLBACK_EVENT_ASR_READY)){
      // 引擎就绪，可以说话，一般在收到此事件后通过UI通知用户可以说话了
    }
    if(name.equals(SpeechConstant.CALLBACK_EVENT_ASR_PARTIAL)){
      // 一句话的临时结果，最终结果及语义结果
    }
      // ... 支持的输出事件和事件支持的事件参数见“输入和输出参数”一节
  }
};
```

> *详见ActivityMiniRecog类中”基于sdk集成1.2 自定义输出事件类"*

1.3 注册自己的输出事件类

```
asr.registerListener(yourListener);
```

> *详见ActivityMiniRecog类中”基于sdk集成1.3 注册自己的输出事件类”*

DEMO中，以上两步合并为

```
IRecogListener listener = new MessageStatusRecogListener(handler);

// 可以传入IRecogListener的实现类，也可以如SDK，传入EventListener实现类
```

```
//如果传入IRecogListener类，在RecogEventAdapter为您做了大多数的json解析。

MyRecognizer myRecognizer = new MyRecognizer(this, listener);
//this是Activity或其它Context类
```

> *详见ActivityAbstractRecog类中”基于DEMO集成第1.1, 1.2, 1.3 步骤 初始化EventManager类并注册自定义输出事件*

1.4 音频格式转换

- 音频格式转换器：将 WAV 格式音频文件转换为 PCM 格式，不改变采样率
- 音频重采样器：将任意采样率的 WAV 格式音频文件重采样到 16k 采样率

```
/**
* 将 WAV 文件转换为 16k PCM 文件
*
* @param wavFilePath   输入的 WAV 文件路径
* @param pcmFilePath   输出的 PCM 文件路径
* @param wavSampleRate WAV 文件的采样率
* @return true 转换成功，false 转换失败
*/
public static boolean convertWavTo16kPcm(String wavFilePath, String pcmFilePath, int wavSampleRate)
```

```
 if (new File(TEST_WAV_FILE_PATH).exists()) {
    if (!AudioProcessingUtils.convertWavTo16kPcm(TEST_WAV_FILE_PATH, TEST_PCM_FILE_PATH, 8000)) {
        Toast.makeText(this, "WAV 转换 PCM 失败", Toast.LENGTH_SHORT).show();
        return;
    }
}
if (!new File(TEST_PCM_FILE_PATH).exists()) {
    Toast.makeText(this, "未找到灌测音频：" + TEST_PCM_FILE_PATH + "，使用麦克风", Toast.LENGTH_SHORT).show();
} else {
    Toast.makeText(this, "找到灌测音频：" + TEST_PCM_FILE_PATH, Toast.LENGTH_SHORT).show();
    /* 传入反射方法 audio() */
    params.put("infile", String.format("#%s.audio()", ActivityLongSpeech.class.getName()));
}
// ...
public static InputStream audio() throws FileNotFoundException {
    return new FileAudioInputStream(TEST_PCM_FILE_PATH);
}
```

如果需要独立使用该接口，即在初始化 SDK（AipeEventManagerFactory.create(...)）之前调用该接口，需要先运行以下代码以加载音频转换依赖的 Native 库：

```
System.loadLibrary("BaiduSpeechSDK");
```

2 开始识别

开始事件的参数可以参见"输入和输出参数"。

2.1 设置识别输入参数

SDK中，您需要根据文档或者demo确定您的输入参数。DEMO中有UI界面简化选择和测试过程。demo中，在点击“开始录音”按钮后，您可以在界面或者日志中看见ASR_START事件的json格式的参数。

```
 // asr.params(反馈请带上此行日志):{"accept-audio-data":false,"disable-punctuation":false,"accept-audio-volume":true,"LANGUAGE":"en-GB"}

//其中{"accept-audio-data":false,"disable-punctuation":false,"accept-audio-volume":true,"LANGUAGE":"en-GB"}为ASR_START事件的参数

String json ="{\"accept-audio-data\":false,\"disable-punctuation\":false,\"accept-audio-volume\":true,\"LANGUAGEpid\":"en-GB"}"
```

> *详见ActivityMiniRecog类中”基于SDK集成2.1 设置识别参数“*

2.2 发送start开始事件

```
asr.send(SpeechConstant.ASR_START, json, null, 0, 0);
```

> *详见ActivityMiniRecog类中”基于SDK集成2.2 发送开始事件*

DEMO中， 您需要传递Map<String,Object>的参数，会将Map自动序列化为json

```
Map<String, Object> params;
... // 设置识别参数
// params ="accept-audio-data":false,"disable-punctuation":false,"accept-audio-volume":true,"LANGUAGEpid":"en-GB"
myRecognizer.start(params);
```

> *详见ActivityAbstractRecog类中"基于DEMO集成2.1, 2.2 设置识别参数并发送开始事件"*

3 收到回调事件

3.1开始回调事件

回调事件在您实现的EventListener中获取。OnEvent方法中， name是输出事件名，params该事件的参数，(data,offset, length)三者一起组成额外数据。如回调的音频数据，从data[offset]开始至data[offset + length] 结束，长度为length。

```
public void onEvent(String name, String params, byte [] data, int offset, int length);
```

> *详见ActivityMiniRecog类中"基于SDK集成3.1 开始回调事件"*

DEMO中， 回调事件在您实现的IRecogListener中获取。

> *详见RecogEventAdapter类中"基于DEMO集成3.1 开始回调事件"*

4 控制识别

4.1 控制停止识别

```
asr.send(SpeechConstant.ASR_STOP, null, null, 0, 0);
//发送停止录音事件，提前结束录音等待识别结果
```

> *详见ActivityMiniRecog类中"基于SDK集成4.1 发送停止事件"*

4.2 控制取消识别

```
asr.send(SpeechConstant.ASR_CANCEL, null, null, 0, 0);
//取消本次识别，取消后将立即停止不会返回识别结果
```

> *详见ActivityMiniRecog类中"基于SDK集成4.2 发送取消事件"*

DEMO 中：

```
myRecognizer.stop();
// 发送停止录音事件，提前结束录音等待识别结果
```

> *详见ActivityAbstractRecog类中"基于DEMO集成4.1 发送停止事件"*

```
myRecognizer.cancel();
// 取消本次识别，取消后将立即停止不会返回识别结果
```

> *详见ActivityAbstractRecog类中"基于DEMO集成4.2 发送取消事件"*

5 事件管理器退出

5.1 在线不需要卸载离线命令词

先启动取消，避免有还在运行的识别。 之后需要将之前的listener卸载，不卸载的话，可能有内存溢出

asr.send(SpeechConstant.ASR_CANCEL, null, null, 0, 0); // 取消识别

5.2 释放资源

```
asr.unregisterListener(this);
```

> *详见ActivityMiniRecog类中基于SDK集成5.2 退出事件管理器"*

DEMO中，

```
myRecognizer.release();
// 含有离线引擎卸载
```

> *详见ActivityAbstractRecog类中基于DEMO的5.2 退出事件管理器"*

9.2 离线命令词调用流程

离线命令词的bsg文件设置：

在语音控制台的左侧功能栏中，进入“离线词&语义设置”模块，根据页面上的引导自行定义词条和语法，并生成bsg文件。其中右侧“说法”部分，为固定语法，下载后不可更改。左侧“词条”部分，代码中可以动态定义覆盖。



离线命令词功能可以测试DEMO中的第二个按钮“离线命令词识别”

调用流程：

离线命令词功能需要首先实现之前的在线识别功能的代码。离线引擎加载需要在EventManager初始化之后，识别事件之前。在SDK中，

```
 HashMap map = new HashMap();
map.put(SpeechConstant.DECODER, 2);
// 0:在线 2.离在线融合(在线优先)
map.put(SpeechConstant.ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH,
"/sdcard/yourpath/baidu_speech_grammar.bsg");
 //设置离线命令词文件路径
```

// 下面这段可选，用于生成SLOT_DATA参数， 用于动态覆盖ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH文件的词条部分

```
 JSONObject json = new JSONObject();
json.put("name", new JSONArray().put("王自强").put("叶问")).put("appname", new
JSONArray().put("手百").put("度秘"));
map.put(SpeechConstant.SLOT_DATA, json.toString());
// SLOT_DATA 参数添加完毕
```

1.1 -1.3步骤同在线

1.4 加载离线资源

```
 asr.send(SpeechConstant.ASR_KWS_LOAD_ENGINE,new
JSONObject(map).toString());
```

//加载离线引擎，使用离线命令词时使用，请在整个离线识别任务结束之后卸载离线引擎

> *详见ActivityMiniRecog类中"基于SDK离线命令词1.4 加载离线资源(离线时使用)"*

//离线引擎加载完毕事件后，开始你的识别流程，此处开始你的识别流程，注意离线必须断网生效或者SDK无法连接百度服务器时生效，只能识别bsg文件里定义的短语。

2.1-4.2 步骤同在线

5.1 卸载离线资源

//不再需要识别功能后，卸载离线引擎。再次需要识别功能的话，可以重复以上步骤。即依旧需要EventManager初始化之后，识别事件之前加载离线引擎。

```
 asr.send(SpeechConstant.ASR_KWS_UNLOAD_ENGINE, null, null, 0, 0);
```

> *详见ActivityMiniRecog类中"基于SDK集成5.1 卸载离线资源步骤(离线时使用)"*

在demo中，

```
 HashMap&lt;String, Object&gt; map = new HashMap&lt;String, Object&gt;();
map.put(SpeechConstant.DECODER, 2);
// 0:在线 2.离在线融合(在线优先)
map.put(SpeechConstant.ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH,
"/sdcard/yourpath/baidu_speech_grammar.bsg");
//设置离线命令词文件路径
```

// 下面这段可选，用于生成SLOT_DATA参数， 用于动态覆盖ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH文件的词条部分

```
 JSONObject json = new JSONObject();
json.put("name", new
JSONArray().put("王自强").put("叶问")).put("appname", new
JSONArray().put("手百").put("度秘"));
map.put(SpeechConstant.SLOT_DATA, json.toString());
// SLOT_DATA 参数添加完毕
```

```
 myRecognizer.loadOfflineEngine(map);
//加载离线引擎，使用离线命令词时使用，请在整个离线识别任务结束之后卸载离线引擎
```

> *详见ActivityAbstractRecog类中"基于DEMO集成1.4 加载离线资源步骤(离线时使用)"*

//离线引擎加载完毕事件后，开始你的识别流程，注意离线必须断网生效或者SDK无法连接百度服务器时生效，只能识别bsg文件里定义的短语。

//不再需要识别功能后，卸载离线引擎。再次需要识别功能的话，可以重复以上步骤，新建MyRecognizer类即可。

```
myRecognizer.release();
```

> *详见ActivityAbstractRecog类中"基于DEMO5.1 卸载离线资源(离线时使用)"*

9.3识别错误码 |错误领域|描述|错误码|错误描述及可能原因| |-|-|-|-| |1|网络超时||出现原因可能为网络已经连接但质量比较差，建议检测网络状态| |||1000|DNS连接超时| |||1001|网络连接超时| |||1002|网络读取超时| |||1003|上行网络连接超时| |||1004|上行网络读取超时| |||1005|下行网络连接超时| |||1006|下行网络读取超时| |2|网络连接失败||出现原因可能是网络权限被禁用，或网络确实未连接，需要开启网络或检测无法联网的原因| |||2000|网络连接失败| |||2001|网络读取失败| |||2002|上行网络连接失败| |||2003|上行网络读取失败| |||2004|下行网络连接失败| |||2005|下行网络读取失败| |||2006|下行数据异常| |||2100|本地网络不可用| |3|音频错误||出现原因可能为：未声明录音权限，或 被安全软件限制，或 录音设备被占用，需要开发者检测权限声明。| |||3001|录音机打开失败| |||3002|录音机参数错误| |||3003|录音机不可用| |||3006|录音机读取失败| |||3007|录音机关闭失败| |||3008|文件打开失败| |||3009|文件读取失败| |||3010|文件关闭失败| |||3100|VAD异常，通常是VAD资源设置不正确| |||3101|长时间未检测到人说话，请重新识别| |||3102|检测到人说话，但语音过短| |4|协议错误||出现原因可能是appid和appkey的鉴权失败| |||4001|协议出错| |||4002|协议出错| |||4003|识别出错| |||4004|鉴权错误 ，一般情况是LANGUAGE appkey secretkey不正确权限 。见下表"4004"鉴权子错误码| |5|客户端调用错误||一般是开发阶段的调用错误，需要开发者检测调用逻辑或对照文档和demo进行修复。| |||5001|无法加载so库| |||5002|识别参数有误| |||5003|获取token失败| |||5004|客户端DNS解析失败| |||5005|| |6|超时||语音过长，请配合语音识别的使用场景，如避开嘈杂的环境等| |||6001|未开启长语音时，当输入语音超过60s时，会报此错误| |7|没有识别结果||信噪比差，请配合语音识别的使用场景，如避开嘈杂的环境等| |||7001|没有匹配的识别结果。当检测到语音结束，或手动结束时，服务端收到的音频数据质量有问题，导致没有识别结果| |8|引擎忙||一般是开发阶段的调用错误，出现原因是上一个会话尚未结束，就让SDK开始下一次识别。SDK目前只支持单任务运行，即便创建多个实例，也只能有一个实例处于工作状态| |||8001|识别引擎繁忙 。当识别正在进行时，再次启动识别，会报busy。| |9|缺少权限|| 参见demo中的权限设置| |||9001|没有录音权限 通常是没有配置录音权限：android.permission.RECORD_AUDIO| |10|其它错误|| 出现原因如：使用离线识别但未将EASR.so集成到程序中；离线授权的参数填写不正确；参数设置错误等。| |||10001|离线引擎异常| |||10002|没有授权文件| |||10003|授权文件不可用| |||10004|离线参数设置错误| |||10005|引擎没有被初始化| |||10006|模型文件不可用| |||10007|语法文件不可用| |||10008|引擎重置失败| |||10009|引擎初始化失败| |||10010|引擎释放失败| |||10011|引擎不支持| |||10012|离线引擎识别失败 。离线识别引擎只能识别grammar文件中约定好的固定的话术，即使支持的话术，识别率也不如在线。请确保说的话清晰，是grammar中文件定义的，测试成功一次后，可以保存录音，便于测试。|

"4004"鉴权子错误码

4004的子错误值 错误码描述	原因	
4	pv超限	配额使用完毕，请购买或者申请
6	没勾权限	应用不存或者应用没有语音识别的权限
13	并发超限	并发超过限额，请购买或者申请
101	API key错误	API Key 填错

服务端错误码请参照点击查看

9.4唤醒词调用流程

SDK 的调用过程可以参见DEMO中的ActivityMiniWakeUp类

DEMO的调用过程可以参考DEMO中的ActivityWakeUp类

1 初始化

1.1 初始化EventManager对象

SDK中，通过工厂创建语音唤醒词的事件管理器。注意唤醒词事件管理器同识别事件管理器一样只能维持一个，请勿同时使用多个实例。即创建一个新的唤醒词事件管理器后，之前的那个设置为null。并不再使用。

```
EventManager wp= EventManagerFactory.create(this,"wp");
//this是Activity或其它Context类
```

> *详见ActivityMiniWakeUp类中"基于SDK唤醒词集成1.1 初始化EventManager"*

1.2 自定义输出事件类

DEMO中，初始化过程合并到下一步。注意SDK和DEMO调用方式2选1即可。 注册用户自己实现的输出事件类

```
EventListener yourListener = new EventListener() {
@Override
public void onEvent(String name, String params, byte [] data, int
offset, int length) {
    Log.d(TAG, String.format("event: name=%s, params=%s", name, params));
    //唤醒事件
    if(name.equals("wp.data")){
        try {
            JSONObject json = new JSONObject(params);
            int errorCode = json.getInt("errorCode");
            if(errorCode == 0){
                //唤醒成功
            } else {
                //唤醒失败
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    } else if("wp.exit".equals(name)){
        //唤醒已停止
    }
    }
};
```

> *详见ActivityMiniWakeUp类中"基于SDK唤醒词集成1.2 自定义输出事件类"*

1.3 注册自己的输出事件类

```
wp.registerListener(yourListener);
```

> *详见ActivityMiniWakeUp类中"基于SDK唤醒词集成1.3 注册输出事件"*

DEMO中，以上两步合并为

```
IWakeupListener listener = new SimpleWakeupListener();
myWakeup = new MyWakeup(this,listener);
// this是Activity或其它Context类
```

> *详见ActivityWakeUp类中"基于DEMO唤醒词集成第1.1, 1.2, 1.3步骤"*

2 开始唤醒

2.1 设置唤醒输入参数

SDK中，您需要根据文档或者demo确定您的输入参数。DEMO中有UI界面简化选择和测试过程。demo中，在点击“开始”按钮后，您可以在界面或者日志中看见WAKEUP_START事件的json格式的参数。

wakeup.params(反馈请带上此行日志):{"kws-file":"assets:///WakeUp.bin"} // 其中{"kws-file":"assets:///WakeUp.bin"}为WAKEUP_START事件的参数

```
HashMap map = new HashMap();
map.put(SpeechConstant.WP_WORDS_FILE, "assets://WakeUp.bin");
```

> *详见ActivityMiniWakeUp类中"基于SDK唤醒词集成第2.1 设置唤醒的输入参数"*
> 唤醒词文件请去语音唤醒页面评估demo设置并下载

2.2 发送start开始事件

```
wp.send(SpeechConstantWAKEUP_START, json, null, 0, 0);
```

> *详见ActivityMiniWakeUp类中"基于SDK唤醒词集成第2.2 发送开始事件开始唤醒"*

DEMO中， 您需要传递Map 的参数，会将Map自动序列化为json

```
HashMap<String,Object> params = new
HashMap<String,Object>();
params.put(SpeechConstant.WP_WORDS_FILE, "assets://WakeUp.bin");
myWakeup.start(params);
```

> *详见ActivityWakeUp类中"基于DEMO唤醒词集成第2.1, 2.2 发送开始事件开始唤醒"*

3 收到回调事件

3.1 开始回调事件

SDK中，回调事件在您实现的EventListener中获取。OnEvent中， name是输出事件名，params该事件的参数，(data,offset, length)三者一起组成额外数据。如回调的音频数据，从data[offset]开始至data[offset + length] 结束，长度为length。

```
public void onEvent(String name, String params, byte [] data, int offset, int length);
```

> *详见ActivityMiniWakeUp类中"基于SDK唤醒3.1 开始回调事件"*

DEMO中， 回调事件在您实现的IWakeupListener中获取。

> *详见WakeupEventAdapter类中"基于DEMO唤醒3.1 开始回调事件"*

4 控制唤醒

SDK中，

4.1 控制停止唤醒

```
wp.send(SpeechConstant.WAKEUP_STOP, null, null, 0, 0);
```

> *详见ActivityMiniWakeUp类中"基于SDK唤醒词集成第4.1 发送停止事件"*

DEMO中，

```
myWakeup.stop();
```

> *详见ActivityWakeUp类中"基于DEMO唤醒词集成第4.1 发送停止事件"*

5 事件管理器退出

SDK中无需调用任何逻辑，但需要创建一个新的唤醒词事件管理器的话，之前的事件管理器请设置为null，并不再使用。

DEMO中，

```
myWakeup.release();
```

> *详见ActivityWakeUp类中"基于DEMO唤醒词集成第5 退出事件管理器">*

9.5唤醒错误码 |错误领域|描述|错误码|错误描述| |-|-|-|-| |10|录音设备出错||| |||1|录音设备异常| |||2|无录音权限| |||3|录音设备不可用| |||4|录音中断| |11|唤醒相关错误||| ||没有授权文件|11002|| ||授权文件不可用|11003|| ||唤醒异常，通常是唤醒词异常|11004|| ||模型文件不可用|11005|| ||引擎初始化失败|11006|| ||内存分配失败|11007|| ||引擎重置失败|11008|| ||引擎释放失败|11009|| ||引擎不支持该架构|11010|| |38|引擎出错||| |||1|唤醒引擎异常| |||2|无授权文件| |||3|授权文件异常| |||4|唤醒异常| |||5|模型文件异常| |||6|引擎初始化失败| |||7|内存分配失败| |||8|引擎重置失败| |||9|引擎释放失败| |||10|引擎不支持该架构| |||11|无识别数据|

10. 代码混淆

```
-keep class com.baidu.speech.**{*;}
```

11. 权限

名称	说明	必选
必要的权限		
android.permission.INTERNET	允许访问网络	是
android.permission.ACCESS_NETWORK_STATE	获取网络状态权限	是
android.permission.RECORD_AUDIO	允许程序录制声音通过手机或耳机的麦克	是
android.permission.WRITE_EXTERNAL_STORAGE	外置卡读写权限	是
非必要权限		
android.permission.CHANGE_WIFI_STATE	允许程序改变Wi-Fi连接状态	否

语音识别iOS SDK

1. 文档说明

文档名称	语音识别集成文档
所属平台	iOS
提交日期	2026-07-30
概述	本文档是百度语音开放平台iOS SDK的用户指南，描述了**短语音识别、离线自定义命令词识别、语音唤醒**等相关接口的使用说明。SDK内部均为采用流式协议，即用户边说边处理。区别于Restapi需要上传整个录音文件。

2. 版本说明

名称	版本号
语音识别	3.0.13.0
系统支持	支持iOS 12.0及以上系统
架构支持	armv7、arm64
开发环境	工程内使用了LTO等优化选项，建议使用最新版本Xcode进行开发

3. SDK说明

文件名称	版本号	说明	MD5值	类型
libBDSpeechBaseKit.a	3.0.13.0	语音识别SDK （支持pcm、wav文件识别格式）	37e494d1117e519d3b7fff7a48d11dfb	静态库

Framework |Framework|描述| |-|-| |libc++.tbd|提供对C/C++特性支持| |libz.1.2.5.tbd|提供gzip支持| |libsqlite3.0.tbd|提供对本地数据库的支持| |AudioToolbox|提供录音和播放支持| |AVFoundation|提供录音和播放支持| |CFNetwork|提供对网络访问的支持| |CoreLocation|提供对获取设备地理位置的支持，以提高识别准确度| |CoreTelephony|提供对移动网络类型判断的支持| |SystemConfiguration|提供对网络状态检测的支持| |GLKit|内置识别控件所需|

4. 运行项目工程

4.1 获取鉴权信息 创建并获取APP_ID、API_KEY、SECRET_KEY等参数。并绑定你的移动应用的BundleId。官网平台地

址https://cloud.baidu.com

4.2 在线识别鉴权参数配置说明

在线识别鉴权方式说明	**开放平台使用API/SECRET KEY + APPID方式进行验证（历史版本）** **控制台申请的AK、SK参考资源领取&应用创建，完成项目创建后控制台可查询对应的AK、SK等信息** **开放平台使用ACCESSTOKEN进行验证（新增）：Access token鉴权机制** **开放平台使用IAMKEY进行验证（新增）：API Key鉴权机制**

ak、sk鉴权方式代码参考事例

```
 //ak、sk鉴权
// 参数传递 ak、sk以及平台获取的appid
[[BDSpeechBaseKit sharedInstance] setASRLicenseWithAk:API_KEY AndSK:SECRET_KEY AndAppcode:APP_ID];
```

APIkey（IAMKEY）鉴权方式

```
  //iamkey鉴权
[[BDSpeechBaseKit sharedInstance] setASRLicenseWithIamKey:IAM_KEY Andsk:SECRET_KEY AndAppCode:APP_ID];
```

token鉴权方式参考：

```
    // 临时token鉴权方式 参考demo中sendPostRequest方法中使用事例
//   [[BDSpeechBaseKit sharedInstance] setASRLicenseWithAuthToken:token AndExpirationDate:result AndAk:API_KEY
AndAppCode:APP_ID];
```

鉴权接口及参数说明

```
 /**
 // ak、sk方式鉴权
 ak：控制台申请的对应鉴权apikey
 sk：控制台申请的对应鉴权secretkey
 code：控制台申请的appid
 */
- (void)setASRLicenseWithAk:(NSString *)ak AndSK:(NSString *)sk AndAppcode:(NSString *)code;

/**
 iamkey方式初始化鉴权
 iamkey：申请的iamkey
 code：申请的appid
 */
- (void)setASRLicenseWithIamKey:(NSString *)iamkey Andsk:(NSString *)ak AndAppCode:(NSString *)code;

// 获取鉴权asr鉴权sdk版本号
- (NSString *)getASRLibVersion;

/**
 token鉴权方式
 token：申请的token
 code：申请的appid
 expirationDate：token过期时间戳（毫秒）
 */
- (void)setASRLicenseWithAuthToken:(NSString *)token AndExpirationDate:(long long)expirationDate AndAk:(NSString *)ak
AndAppCode:(NSString *)code;

/**
 iamkey方式初始化鉴权
 iamkey：申请的iamkey
 code：申请的appid
 expirationDate：iamkey过期时间（毫秒）
 */
- (void)setASRLicenseWithIAMKey:(NSString *)iamkey AndExpirationDate:(long long)expirationDate AndAk:(NSString *)ak
AndAppCode:(NSString *)code;
```

demo完整示例参考：

```
 - (void)viewDidLoad {
    [super viewDidLoad];
    // 语音识别鉴权库manager管理类初始化
    self.asrEventManager = [[BDSpeechBaseKit sharedInstance] getBDSEventManager];
    // 设置唤醒类
    self.wakeupEventManager = [[BDSpeechBaseKit sharedInstance] getWakeupEventManager];
    // pid初始化 默认1537
    [self.asrEventManager setParameter:@"1537" forKey:BDS_ASR_PRODUCT_ID];
    // 版本号
    NSLog(@"Current SDK version: %@", [[BDSpeechBaseKit sharedInstance] getASRLibVersion]);
    // 配置鉴权库获取token、iamkey协议 用于实现 gettoken iamkey 方法
    [ASRAuthManager sharedInstance].temporaryTokenAndKeyDelegate = self;
    // 鉴权sdk层错误日志开启 默认NO不开启
//    [[BDSpeechBaseKit sharedInstance] setAuthLogStatus:YES];
    //设置DEBUG_LOG的级别 默认0不打开日志 设置6全部日志 其他方式不常用特定场景考虑
    [self.asrEventManager setParameter:@(EVRDebugLogLevelTrace) forKey:BDS_ASR_DEBUG_LOG_LEVEL];
    //ak、sk鉴权
    [[BDSpeechBaseKit sharedInstance] setASRLicenseWithAk:API_KEY AndSK:SECRET_KEY AndAppcode:APP_ID];
    //iamkey鉴权
//  [[BDSpeechBaseKit sharedInstance] setASRLicenseWithIamKey:IAM_KEY Andsk:SECRET_KEY AndAppCode:APP_ID];
    // 临时token鉴权方式 参考sendPostRequest方法中使用事例
//    [[BDSpeechBaseKit sharedInstance] setASRLicenseWithAuthToken:token AndExpirationDate:result AndAk:API_KEY
AndAppCode:APP_ID];
    self.continueToVR = NO;
    [[BDVRSettings getInstance] configBDVRClient];
    //设置端点检测
    [self configDNNMFE];
}
```

提供用户自定义获取鉴权信息的能力

描述：为了满足 「鉴权信息保存在server端，app通过请求server端获取access token」的需求，语音SDK需要提供 鉴权信息回调点，并由用户最终实现 回调接口。

```
 步骤一：
// 配置鉴权库获取token、iamkey协议 用于实现 gettoken iamkey 方法
[ASRAuthManager sharedInstance].temporaryTokenAndKeyDelegate = self;

// 分别实现协议传入token、iamkey
/**
 客户实现get token 协议方法
 */
- (void)getToken {
    ......
    //客户业务侧处理获取token的逻辑
    // demo示例发送请求
     [self sendPostRequest];
}

/**
 客户实现get iamkey 协议方法
 */
- (void)getIAMKey {
    ......
    //客户业务侧处理获取iamkey的逻辑
    // demo示例发送请求
     [self sendPostRequest];
}

步骤二：
//获取网络权限判断逻辑
```

```
//获取网络权限判断逻辑
    ......
    ......
    ......
// 检查当前网络状态并尝试网络请求
[self checkNetworkStatusAndMakeRequest];


 // 检查网络权限是否生效
- (void)checkNetworkStatusAndMakeRequest {
   // 生效 即可做post请求获取token、有效期
    ......
    ......
    ......
   // 调用本地获取token的请求逻辑
    [self sendPostRequest];
}

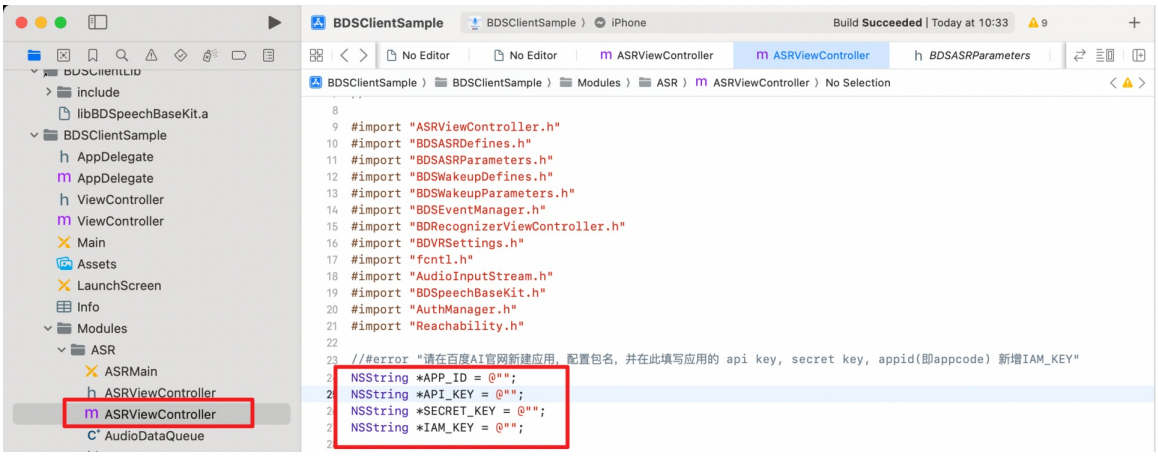
 // post请求 仅供参考
-(void)sendPostRequest {
 // 请求逻辑具体参考demo代码
  ......
  ......
  请求拿到token后调用API接口完成token鉴权
  // token 鉴权  成功：正常完成合成主流程鉴权通过。 失败：触发上述步骤一中的协议回调 重新获取有效token（此处以token为例）
  [[BDSpeechBaseKit sharedInstance] setASRLicenseWithAuthToken:@"请求获取的token" AndExpirationDate:@"请求获取的token有效期" AndAk:@"平台获取的apikey" AndAppCode:@"平台获取的appid"];
}

逻辑整理 1.遵循鉴权sdk内置协议 2.完成初始化请求token获取[self sendPostRequest]; 并在请求结果中设置对应的鉴权方式如 [[BDSpeechBaseKit sharedInstance] setASRLicenseWithAuthToken:@"请求获取的token" AndExpirationDate:@"请求获取的token有效期" AndAk:@"平台获取的apikey" AndAppCode:@"平台获取的appid"];3.成功后正常继续下一步，异常或请求token失败、token有效期到期等 SDK代理会监听到 触发getToken、getIamkey代理 进行重新处理逻辑 demo示例中做了重新获取token的操作 仅供参考
注：使用该方式为了满足 「鉴权信息保存在server端，app通过请求server端获取access token」的需求，语音SDK需要提供 鉴权信息回调点，并由用户最终实现 回调接口。
```

4.3 下载语音识别SDK 在SDK下载页面下载 **语音识别IOS SDK** 文件，链接：https://ai.baidu.com/sdk



4.4 填写鉴权信息 全局搜索 “#error” 填入 **AppID、API Key、Secret Key IAM_KEY**鉴权信息测试，请按照以下步骤全部修改（参数信息参考4.2鉴权参数设置）：



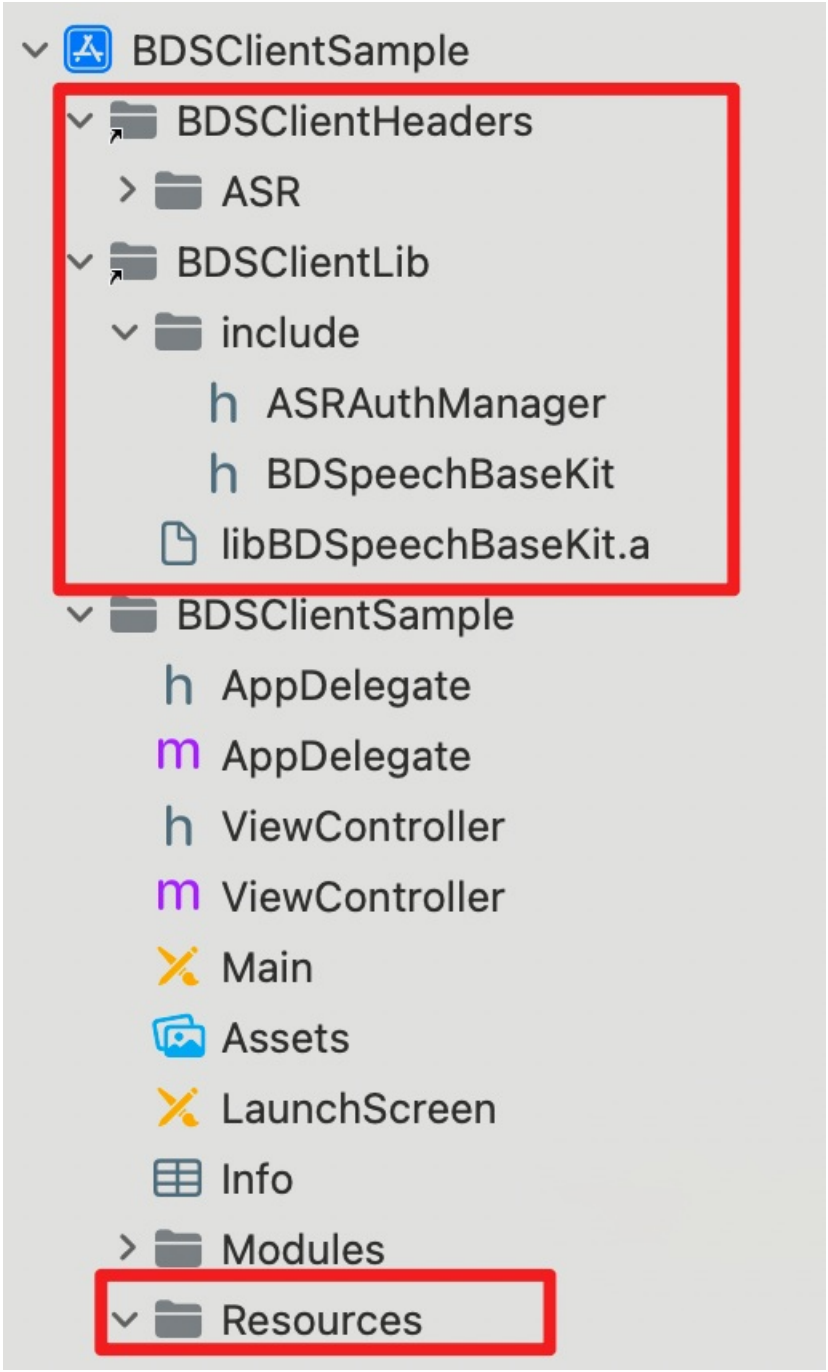
4.5 测试语音识别功能 按照上述文档修改完成后，安装app打开后可进行在线语音识别：



简单的 语音识别IOS SDK 测试完成了，其他各子功能可以按照详细的技术文档进行集成。

5. 集成步骤

5.1添加静态库及资源文件、头文件以及对应的Framework



添加头文件 识别相关 如果只需要使用识别功能，只需要引入如下头文件：

```
 ##### import "BDSEventManager.h"
##### import "BDSASRDefines.h"
##### import "BDSASRParameters.h"
##### import "BDSpeechBaseKit.h"
##### import "ASRAuthManager.h"
```

唤醒相关 如果需要使用离线唤醒功能，需要引入如下头文件：

```
 ##### import "BDSWakeupDefines.h"
##### import "BDSWakeupParameters.h"
```

内置识别控件 如果需要使用内置识别控件，需要引入如下头文件：

```
 ##### import "BDTheme.h"
##### import "BDRecognizerViewParamsObject.h"
##### import "BDRecognizerViewController.h"
##### import "BDRecognizerViewDelegate.h"
```

网络权限相关

```
 ##### import "Reachability.h"
```

添加静态库 SDK提供的是静态库，开发者只需要将库文件拖入工程目录即可。对静态库有以下几点说明：

```
 1. 静态库采用lipo命令将armv7，arm64的静态库合并成的一个通用的库文件，避免开发者在编译不同target时频繁替换的问题；
2. 模拟器版本只支持在线识别，不支持离线识别及唤醒功能;
```

添加所需资源 提示音文件及识别控件所需主题文件 将开发包中BDSClientResource/ASR/BDSClientResources目录以“create folder references”方式添加到工程的资源Group中，**注意使用"create groups"方式添加不能生效**。

离线识别及唤醒所需资源文件 将开发包中BDSClientResource/ASR/BDSClientEASRResources目录以"create groups"方式添加到工程目录下即可，资源文件说明如下：

文件名	说明
bds_easr_gramm.dat	离线识别引擎语法模式所需语法文件，在开放平台编辑自定义语法文件
bds_easr_basic_model.dat	基础资源文件，用于modelVAD、唤醒、离线语音识别语法模式
bds_easr_wakeup_words.dat	唤醒引擎所需唤醒词文件，在开放平台编辑自定义唤醒词
bds_easr_mfe_dnn.dat	基础资源文件，用于DNNMFE、唤醒、离线语音识别语法模式
bds_easr_mfe_cmvn.dat	MFE CMVN文件,用于DNNMFE
bds_easr_dnn_wakeup_model.dat	用于DNNWakeup的模型文件

6. 相关授权文件

请将百度云控制台创建应用时获取的语音(APPID)、API/SECRET KEY 并填写包名。

在线识别与唤醒都需要进行相关验证后方可使用：

引擎类型	验证方法
在线识别	开放平台使用API/SECRET KEY + APPID进行验证（包含新增token鉴权、iamkey鉴权）
离线识别	使用APPID+包名首次联网自动下载授权文件进行验证
唤醒引擎	与离线识别验证方法一致

7. 语音相关接口调用流程

7.1语音识别 语音识别包含数据上传接口和离在线识别接口，接口概述如下：

```
 1. 创建相关接口对象 (createEventManagerWithName:)
2. 设置代理对象 (setDelegate:)
3. 配置参数 (setParameter:forKey:)
4. 发送预定义指令 (sendCommand:)
5. 参数列表及相关预定义值可参考附录，或相关parameters头文件、defines头文件
```

1. 在线语音识别支持识别任意词，离线语音识别仅支持命令词识别（语法模式）。
2. 单次短语音识别最长限制60秒。

7.1.1在线识别

```
 // 创建语音识别对象
self.asrEventManager = [[BDSpeechBaseKit sharedInstance] getBDSEventManager];
// pid初始化 默认1537
[self.asrEventManager setParameter:@"1537" forKey:BDS_ASR_PRODUCT_ID];
// 设置语音识别代理
[self.asrEventManager setDelegate:self];
// 配置鉴权库获取token、iamkey协议 用于实现 gettoken iamkey 方法
[ASRAuthManager sharedInstance].temporaryTokenAndKeyDelegate = self;
// 参数配置：ak、sk在线鉴权认证
[[BDSpeechBaseKit sharedInstance] setASRLicenseWithAk:API_KEY AndSK:SECRET_KEY AndAppcode:APP_ID];
// 发送指令：启动识别
[self.asrEventManager sendCommand:BDS_ASR_CMD_START];
```

识别功能代理

```
 @protocol BDSClientASRDelegate<NSObject>
- (void)VoiceRecognitionClientWorkStatus:(int)workStatus obj:(id)aObj;
@end
```

语音识别状态、录音数据等回调均在此代理中发生，具体事件请参考Demo工程中对不同workStatus的处理流程。

7.1.2离在线并行识别

```
 // 创建语音识别对象
self.asrEventManager = [BDSEventManager createEventManagerWithName:BDS_ASR_NAME];
// 设置语音识别代理
[self.asrEventManager setDelegate:self];
// 参数配置：在线身份验证
[[BDSpeechBaseKit sharedInstance] setASRLicenseWithAk:API_KEY AndSK:SECRET_KEY AndAppcode:APP_ID];
// 参数设置：识别策略为离在线并行
[self.asrEventManager setParameter:@(EVR_STRATEGY_BOTH) forKey:BDS_ASR_STRATEGY];
// 参数设置：离线识别引擎类型
[self.asrEventManager setParameter:@(EVR_OFFLINE_ENGINE_GRAMMER) forKey:BDS_ASR_OFFLINE_ENGINE_TYPE];
// 参数配置：命令词引擎语法文件路径。请从控制台的左侧功能栏进入“离线词&语义设置”模块，根据页面上的引导自行定义词条和语法，并生成bsg文件。下载语法文件后，替换BDS_ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数
[self.asrEventManager setParameter:@"命令词引擎语法文件路径"
forKey:BDS_ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH];
// 参数配置：命令词引擎语言模型文件路径
[self.asrEventManager setParameter:@"命令词引擎语言模型文件路径"
forKey:BDS_ASR_OFFLINE_ENGINE_DAT_FILE_PATH];
// 发送指令：加载离线引擎
[self.asrEventManager sendCommand:BDS_ASR_CMD_LOAD_ENGINE];
// 发送指令：启动识别
[self.asrEventManager sendCommand:BDS_ASR_CMD_START];
```

关于离线识别

注意

在线识别效果远优于离线识别，不推荐使用离线识别。

> 首次使用离线，SDK将会后台下载离线授权文件，成功后，授权文件有效期（三年）内无需联网。有效期即将结束后SDK将自动多次尝试联网更新证书)。

使用离线识别必须正确配置BDS_ASR_OFFLINE_APP_CODE，并设置BDS_ASR_STRATEGY为离线在线并行。

离线识别 可识别自定义语法规则下的词，如“打电话给王五”，“打开微信”等。

请在语音控制台的左侧功能栏中，进入“离线词&语义设置”模块，根据页面上的引导自行定义词条和语法，并生成bsg文件。下载语法文件后，设置BDS_ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数

具体示例如下：

```
 NSString* gramm_filepath = [[NSBundle mainBundle] pathForResource:@"bds_easr_gramm" ofType:@"dat"];;
NSString* lm_filepath = [[NSBundle mainBundle] pathForResource:@"bds_easr_basic_model" ofType:@"dat"];;
NSString* wakeup_words_filepath = [[NSBundle mainBundle] pathForResource:@"bds_easr_wakeup_words"
ofType:@"dat"];;
[self.asrEventManager setDelegate:self];
[self.asrEventManager setParameter:APP_ID forKey:BDS_ASR_OFFLINE_APP_CODE];
[self.asrEventManager setParameter:lm_filepath forKey:BDS_ASR_OFFLINE_ENGINE_DAT_FILE_PATH];
// 请在 (官网)[http://speech.baidu.com/asr] 参考模板定义语法，下载语法文件后，替换
BDS_ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH参数
[self.asrEventManager setParameter:gramm_filepath forKey:BDS_ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH];
[self.asrEventManager setParameter:wakeup_words_filepath
forKey:BDS_ASR_OFFLINE_ENGINE_WAKEUP_WORDS_FILE_PATH];
```

7.1.3长语音识别 长语音识别对语音时长无限制，其本质是在本地进行VAD之后，由服务端逐句识别。

```
 [self.asrEventManager setParameter:@(NO) forKey:BDS_ASR_NEED_CACHE_AUDIO];
[self.asrEventManager setParameter:@"" forKey:BDS_ASR_OFFLINE_ENGINE_TRIGGERED_WAKEUP_WORD];
[self.asrEventManager setParameter:@(YES) forKey:BDS_ASR_ENABLE_LONG_SPEECH];
// 长语音请务必开启本地VAD
[self.asrEventManager setParameter:@(YES) forKey:BDS_ASR_ENABLE_LOCAL_VAD];
```

> 1. 使用长语音必须开启本地VAD: BDS_ASR_ENABLE_LOCAL_VAD
> 2. 使用长语音必须关闭提示音（Known issue）

7.1.4VAD 端点检测，即自动检测音频输入的起始点和结束点。`SDK默认开启VAD，检测到静音后自动停止识别。` 如果需要自行控制识别结束需关闭VAD，请同时关闭服务端VAD与端上VAD：

```
 // 关闭服务端VAD
[self.asrEventManager setParameter:@(NO) forKey:BDS_ASR_ENABLE_EARLY_RETURN];
// 关闭本地VAD
[self.asrEventManager setParameter:@(NO) forKey:BDS_ASR_ENABLE_LOCAL_VAD];
```

目前SDK支持两种本地端点检测方式。说明如下：

识别策略	说明
ModelVAD	检测更加精准，抗噪能力强，响应速度较慢
DNNMFE	提供基础检测功能，性能高，响应速度快

> 使用ModelVAD、DNN需通过参数配置开启该功能，并配置相应资源文件（基础资源文件）

7.1.4.1ModelVAD

```
 //获取VAD模型的路径
NSString *modelVAD_filepath = [[NSBundle mainBundle] pathForResource:@"bds_easr_basic_model" ofType:@"dat"];
//设置modelVAD的文件路径
[self.asrEventManager setParameter:modelVAD_filepath forKey:BDS_ASR_MODEL_VAD_DAT_FILE];
//设置ModelVAD可用
[self.asrEventManager setParameter:@(YES) forKey:BDS_ASR_ENABLE_MODEL_VAD];
```

7.1.4.2DNNMFE

```
 NSString *mfe_dnn_filepath = [[NSBundle mainBundle] pathForResource:@"bds_easr_mfe_dnn" ofType:@"dat"];
//设置MFE模型文件
[self.asrEventManager setParameter:mfe_dnn_filepath forKey:BDS_ASR_MFE_DNN_DAT_FILE];
NSString *cmvn_dnn_filepath = [[NSBundle mainBundle] pathForResource:@"bds_easr_mfe_cmvn" ofType:@"dat"];
//设置MFE CMVN文件路径
[self.asrEventManager setParameter:cmvn_dnn_filepath forKey:BDS_ASR_MFE_CMVN_DAT_FILE];
```

DNNMFE 支持设置静音时长（需要同时关闭服务端VAD与端上VAD）。设置以下两个参数，单位为帧数，每帧10ms。如需设置为 5s：

```
 [self.asrEventManager setParameter:@(501.f) forKey:BDS_ASR_MFE_MAX_SPEECH_PAUSE];
[self.asrEventManager setParameter:@(500.f) forKey:BDS_ASR_MFE_MAX_WAIT_DURATION];
```

必须为浮点类型。

7.2语音唤醒 语音唤醒，需要配置所需**语言模型文件(基础资源文件)及官网导出的自定义唤醒词文件**，配置后加载引擎，即可进行开始唤醒。需要注意的是，唤醒引擎开启后会保持录音机为启动状态，用户说出正确的唤醒词后会触发唤醒，通过相关回调反馈给应用程序。

语音唤醒为离线功能，需配置离线授权信息(APP_ID)，加载唤醒所需语言模型文件，接口与语音识别接口相同。

基于多种因素考虑，在App进入后台后，唤醒将会被打断。

7.2.1代码示例

```
 // 创建语音识别对象
self.wakeupEventManager = [BDSEventManager createEventManagerWithName:BDS_WAKEUP_NAME];
// 设置语音唤醒代理
[self.wakeupEventManager setDelegate:self];
// 参数配置：离线授权APPID
[self.wakeupEventManager setParameter:APP_ID forKey:BDS_WAKEUP_APP_CODE];
// 参数配置：唤醒语言模型文件路径, 默认文件名为 bds_easr_basic_model.dat
[self.wakeupEventManager setParameter:@"唤醒语言模型文件路径" forKey:BDS_WAKEUP_DAT_FILE_PATH];
// 发送指令：加载语音唤醒引擎
[self.wakeupEventManager sendCommand:BDS_WP_CMD_LOAD_ENGINE];
//设置唤醒词文件路径
// 默认的唤醒词文件为"bds_easr_wakeup_words.dat"，包含的唤醒词为"百度以下"
// 如需自定义唤醒词，请在 http://ai.baidu.com/tech/speech/wake 中评估并下载唤醒词文件，替换此参数
 [self.wakeupEventManager setParameter：@"唤醒词文件路径"forKey:BDS_WAKEUP_WORDS_FILE_PATH]
// 发送指令：启动唤醒
[self.wakeupEventManager sendCommand:BDS_WP_CMD_START];
```

7.2.2唤醒功能回调接口

```
 @protocol BDSClientWakeupDelegate<NSObject>
- (void)WakeupClientWorkStatus:(int)workStatus obj:(id)aObj;
@end
```

7.2.3唤醒辅助识别

使用唤醒的一种需求场景是唤醒后立刻识别，以唤醒词为**百度以下**举例，用户可能的输入为**百度以下，北京天气怎么样？**如果开发者需要对该种场景进行支持，请按如下操作：

1. 正确配置唤醒引擎，语言模型文件及唤醒词文件，并加载引擎；
2. 开启唤醒，接收用户语音输入；
3. 在唤醒的唤醒词触发回调中，配置BDS_ASR_NEED_CACHE_AUDIO为YES到识别引擎，正常识别请将该值设为NO；
4. 调用识别引擎开启识别过程；

```
 // 如需要唤醒后立刻进行识别，为保证不丢音，启动语音识别前请添加如下配置，获取录音缓存：
[self.asrEventManager setParameter:@(YES) forKey:BDS_ASR_NEED_CACHE_AUDIO];
```

8. 语音识别

8.1预定义命令

语音识别目前支持的命令如下：

命令	功能描述
BDS_ASR_CMD_START	启动识别
BDS_ASR_CMD_STOP	结束语音输入，等待识别完成
BDS_ASR_CMD_CANCEL	取消本次识别
BDS_ASR_CMD_LOAD_ENGINE	加载离线引擎，如使用离线识别，在启动识别前需调用此命令
BDS_ASR_CMD_UNLOAD_ENGINE	卸载离线引擎，如改变离线配置参数，需重新加载离线引擎

8.2参数说明 通过配置不同的参数，语音识别提供丰富的功能，说明如下：

在线引擎身份验证 |参数名称|说明| |-|-| |BDS_ASR_API_SECRET_KEYS|开放平台设置API_KEY and SECRET_KEY| |BDS_ASR_IAM_APIKEY|开放平台设置IAMKEY| |BDS_ASR_ACCESS_TOKEN|开放平台设置TOKEN| |BDS_ASR_PRODUCT_ID|内部产品设置产品ID|

离线引擎身份验证 |参数名称|说明| |-|-| |BDS_ASR_OFFLINE_LICENSE_FILE_PATH|离线授权文件路径| |BDS_ASR_OFFLINE_APP_CODE|离线授权所需APPCODE（APPID），如使用该方式进行正式授权，请移除临时授权文件|

识别器参数配置 |参数名称|说明| |-|-| |BDS_ASR_SAMPLE_RATE|设置录音采样率，自动模式根据当前网络情况自行调整| |BDS_ASR_STRATEGY|语音识别策略| |BDS_ASR_LANGUAGE|设置识别语言| |BDS_ASR_ENABLE_NLU|开启语义解析，将返回包含语义的json串| |BDS_ASR_DISABLE_PUNCTUATION|关闭输出标点| |BDS_ASR_PUNCTUATION_EXT_MOD|扩展标点模式，使用请确保BDS_ASR_DISABLE_PUNCTUATION参数为NO| |BDS_ASR_ENABLE_LOCAL_VAD|是否需要对录音数据进行端点检测，如果关闭，请同时关闭服务端提前返回| |BDS_ASR_ENABLE_EARLY_RETURN|服务端开启提前返回，即允许服务端在未收到客户端发送的结束标志前提前结束识别过程| |BDS_ASR_ENABLE_MODEL_VAD|是否使用ModelVAD，打开需配置资源文件参数| |BDS_ASR_MODEL_VAD_DAT_FILE|ModelVAD所需资源文件路径| |BDS_ASR_VAD_ENABLE_LONG_PRESS|设置VAD模式为长按（特殊情况设置）| |BDS_ASR_MFE_DNN_DAT_FILE|设置MFE模型文件| |BDS_ASR_MFE_CMVN_DAT_FILE|设置MFE CMVN文件路径| |BDS_ASR_MFE_MAX_WAIT_DURATION|设置DNNMFE最大等待语音时间| |BDS_ASR_MFE_MAX_SPEECH_PAUSE|设置DNNMFE切分门限| |BDS_ASR_ENABLE_LONG_SPEECH|是否启用长语音识别|

音频相关 |参数名称|说明| |-|-| |BDS_ASR_AUDIO_FILE_PATH|设置音频文件路径（数据源）支持wav、pcm格式| |BDS_ASR_AUDIO_FILE_RATE|设置音频文件采样率（必须与音频文件本身采样率匹配）支持8k、16k设置| |BDS_ASR_AUDIO_INPUT_STREAM|设置音频输入流（数据源）| |BDS_ASR_PLAY_TONE|识别提示音设置，需添加相应声音文件，可替换| |BDS_ASR_DISABLE_AUDIO_OPERATION|屏蔽SDK内部设置AudioSession的Active状态|

日志级别 |参数名称|说明| |-|-| |BDS_ASR_DEBUG_LOG_LEVEL|指定调试日志级别|

离线识别相关 |参数名称|说明| |-|-| |BDS_ASR_OFFLINE_ENGINE_TYPE|离线识别引擎类型| |BDS_ASR_OFFLINE_ENGINE_DAT_FILE_PATH|离线识别资源文件路径| |BDS_ASR_OFFLINE_ENGINE_GRAMMER_FILE_PATH|离线识别语法文件路径| |BDS_ASR_OFFLINE_ENGINE_GRAMMER_SLOT|语法模式离线语法槽，使用该参数更新离线语法文件|

唤醒后立刻识别相关 |参数名称|说明| |-|-| |BDS_ASR_OFFLINE_ENGINE_WAKEUP_WORDS_FILE_PATH|唤醒词文件路径，使用了唤醒并使用离线语法识别的情况下需要设置，其他情况请忽略该参数| |BDS_ASR_OFFLINE_ENGINE_TRIGGERED_WAKEUP_WORD|当前触发唤醒词，唤醒后立即调用识别的情况下配置，其他情况请忽略该参数| |BDS_ASR_NEED_CACHE_AUDIO|唤醒后立刻进行识别需开启该参数，其他情况请忽略该参数|

服务端配置相关 |参数名称|说明| |-|-| |BDS_ASR_SERVER_URL|设置服务器地址| |BDS_ASR_BROWSER_USER_AGENT|设置浏览器标识(Http request header)，资源返回时会根据UA适配|

识别状态 语音识别回调状态如下：

识别状态	返回值说明	功能描述
EVoiceRecognitionClientWorkStatusStartWorkIng	nil	识别工作开始，开始采集及处理数据
EVoiceRecognitionClientWorkStatusStart	nil	检测到用户开始说话
EVoiceRecognitionClientWorkStatusEnd	nil	本地声音采集结束结束，等待识别结果返回并结束录音
EVoiceRecognitionClientWorkStatusNewRecordData	NSData-原始音频数据	录音数据回调
EVoiceRecognitionClientWorkStatusFlushData	NSDictionary-中间结果	连续上屏
EVoiceRecognitionClientWorkStatusFinish	NSDictionary-最终识别结果	语音识别功能完成，服务器返回正确结果
EVoiceRecognitionClientWorkStatusMeterLevel	NSNumber:int-当前音量	当前音量回调
EVoiceRecognitionClientWorkStatusCancel	nil	用户取消
EVoiceRecognitionClientWorkStatusError	NSError-错误信息	发生错误
EVoiceRecognitionClientWorkStatusLoaded	nil	离线引擎加载完成
EVoiceRecognitionClientWorkStatusUnLoaded	nil	离线引擎卸载完成
EVoiceRecognitionClientWorkStatusChunkThirdData	NSData	CHUNK: 识别结果中的第三方数据
EVoiceRecognitionClientWorkStatusChunkNlu	NSData	CHUNK: 识别结果中的语义结果
EVoiceRecognitionClientWorkStatusChunkEnd	NSString	CHUNK: 识别过程结束
EVoiceRecognitionClientWorkStatusFeedback	NSString	Feedback: 识别过程反馈的打点数据
EVoiceRecognitionClientWorkStatusRecorderEnd	nil	录音机关闭，页面跳转需检测此时间，规避状态条(iOS)
EVoiceRecognitionClientWorkStatusLongSpeechEnd	nil	长语音结束状态

9.语音唤醒

9.1预定义命令 语音唤醒目前支持的命令如下：

命令	功能描述
BDS_WP_CMD_START	启动唤醒
BDS_WP_CMD_STOP	关闭唤醒，释放内存需调用卸载命令
BDS_WP_CMD_LOAD_ENGINE	加载唤醒引擎
BDS_WP_CMD_UNLOAD_ENGINE	卸载唤醒引擎，如改变了引擎参数，请重新加载

9.2参数说明 为使唤醒引擎正常工作，开发者需了解以下参数：

基本配置 |参数名称|说明| |-|-| |BDS_WAKEUP_WORDS_FILE_PATH|唤醒词文件路径，从开放平台获取该文件| |BDS_WAKEUP_DAT_FILE_PATH|唤醒引擎模型文件路径| |BDS_WAKEUP_APP_CODE|离线正式授权所需APPCODE，即APPID| |BDS_WAKEUP_LICENSE_FILE_PATH|离线授权文件路径，正式授权需移除该文件| |BDS_WAKEUP_WORK_QUEUE|指定SDK工作队列|

音频相关 音频相关的参数与识别引擎共享，如同时使用，只需配置一次

参数名称	说明
BDS_WAKEUP_AUDIO_FILE_PATH	设置音频文件路径（数据源）
BDS_WAKEUP_AUDIO_INPUT_STREAM	设置音频输入流（数据源）
BDS_WAKEUP_DISABLE_AUDIO_OPERATION	屏蔽SDK内部设置AudioSession的Active状态

9.3唤醒状态 语音唤醒回调状态如下：

唤醒状态	返回值说明	功能描述
EWakeupEngineWorkStatusStarted	nil	引擎开始工作
EWakeupEngineWorkStatusStopped	nil	引擎关闭完成
EWakeupEngineWorkStatusLoaded	nil	唤醒引擎加载完成
EWakeupEngineWorkStatusUnLoaded	nil	唤醒引擎卸载完成
EWakeupEngineWorkStatusTriggered	NSString-唤醒词	命中唤醒词
EWakeupEngineWorkStatusError	NSError-错误信息	引擎发生错误

10.服务端错误码

请参照API 错误码汇总

11. 权限

名称	用途
Privacy - Microphone Usage Description	获取麦克风权限

12. 只使用在线识别能力

可以参照7.1.1在线识别代码，删除离线资源文件

语音识别 HarmonyOS SDK

1. 文档说明

文档名称	语音识别集成文档
所属平台	HarmonyOS
提交日期	2026-01-06
概述	本文档是百度语音开放平台HarmonyOS SDK的用户指南，描述了短语音识别、长语音识别等相关接口的使用说明。SDK内部均为采用流式协议，即用户边说边处理。区别于Restapi需要上传整个录音文件。

2. 版本说明

名称	版本号
语音识别	1.0.3
系统支持	HarmonyOS 5.0.4（APILevel 16）+
架构支持	arm64-v8a，armeabi-v7a

3. SDK说明

3.1. 开发包说明 |文件名称|说明| |-|-| |doc/Baidu_ASR_SDK_Harmony_Manual.md|本文档| |har|语音识别SDK har库| |BaiduAsrDemo|开发示例|

4. 在线识别调用流程

4.1 初始化 SDK 需要传递三个参数，分别为context上下文，产品pid，cuid。

示例代码

```
SpeechEventManager.getInstance().initSdk(
  getContext(this) as common.UIAbilityContext,   // context上下文
  "1234",                                        // 产品pid
  "cuid"                                         // cuid 可选，非必填
)
```

4.2 识别 4.2.1 启动识别 this.listener 和 callback 均是回调，用于传入不同形式的回调，业务方按需使用，仅传一个即可

```
 let startParams: StartParamsAsr = new StartParamsAsr()
startParams.pid = 123 as number                              // 识别环境 - pid：非必填，默认1537
startParams.authInfo = {ak: "apikey", sk: "secretkey"}          // 鉴权参数
startParams.asrType = SpeechAsrType.SHORT     // SpeechAsrType.SHORT: 单次； SpeechAsrType.TOUCH: 长按
SpeechAsrType.MULTI: 全双工； SpeechAsrType.TRANSLITERATE: 长语音转写
startParams.earlyReturn = 1                  // 是否打开提前返回
startParams.acceptAudioVolume = true          // 是否接收音量回调


interface Result {
    word: string[];
    confident: number[];
}

interface RecognizeParams {
    err_no: number;
    result: Result;
    asr_align_begin: number;
    asr_align_end: number;
    raf: number;
    early_return_duration_frame: number;
    corpus_no: number;
    sn: string;
    force_align_result: string;
    confidence_status: number;
    product_id: number;
    product_line: string;
    other_params: string;
    result_type: string;
    speak_speed: number;
    voice_power: number;
}

SpeechEventManager.getInstance().startAsr(startParams, this.asrListener,
    (asrState: string, params: string = '', audioData: ArrayBuffer) => {
        let showMsg = "====" + asrState + ", " + params
        let msg = "Asr callback: " + showMsg
        LogUtil.d(msg)

        if (asrState === SpeechAsrState.ASR_AUDIO_DATA) {
        } else if (asrState === SpeechAsrState.ASR_AUDIO_VOLUME_LEVEL) {
        } else if (asrState === SpeechAsrState.ASR_READY) {

        } else if (asrState === SpeechAsrState.ASR_PARTIAL) {   // 中间结果
            const parsedResponse = JSON.parse(params) as RecognizeParams
            console.log("Partial result: " + parsedResponse?.result?.word[0])
        } else if (asrState === SpeechAsrState.ASR_FINAL) {     // 最终结果
        } else if (asrState === SpeechAsrState.ASR_TTS) {       // tts
        } else if (asrState === SpeechAsrState.ASR_THIRD) {     // 三方数据
        }else if (asrState === SpeechAsrState.ASR_FINISH) {     // 识别结束
            const parsedResponse = JSON.parse(params) as RecognizeParams
            console.log("Final result: " + parsedResponse?.result?.word[0])
        } else if (asrState === SpeechAsrState.ASR_EXIT) {
        }
    })
```

4.2.1.1 回调样例

```
 asr.start, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5"}
asr.ready, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5"}
asr.begin, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5"}
asr.partial, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5", "err_no": 0, "best_result": "我"}
asr.partial, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5", "best_result": "我放"}
asr.partial, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5", "best_result": "播放音"}
asr.partial, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5", "best_result": "播放音乐"}
asr.end, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5"}
asr.final_result, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5", "best_result": "播放音乐。"}
asr.finish, {"sn":"a6eb51fb-d44a-465b-9129-9408ae4d7df5","err_no":0,"err":{"errorCode":0,"desc":"Speech Recognize
success."}}
asr.exit, {"sn": "a6eb51fb-d44a-465b-9129-9408ae4d7df5"}

识别失败：
asr.final_result, {"sn": "162a2ef1-4551-41e9-aac3-93496500b409", "err_no": -3005, "err_msg": "asr server not find
effective speech"}
```

4.2.1.1.1 asr.finish

```
 {
 sn: string,
 err_no: number,
 err: {
  errcode: number,
  desc: string
 }
}

sn:asr.finish对应的query的sn。
err_no:错误码，正常识别结束为0
err：{
 errcode:错误码，和err_no一致
 desc：错误描述
}
```

4.2.2 动态设置参数

```
 let configParams: ConfigParamsAsr = new ConfigParamsAsr()
configParams.enableLongPress = true
SpeechEventManager.getInstance().configAsr(configParams)
```

4.2.3 停止识别

```
 SpeechEventManager.getInstance().stopAsr()
```

4.2.4 取消识别

```
 SpeechEventManager.getInstance().cancelAsr()
```

4.2.5 支持8k wav文件识别 识别8k wav文件的时候，先进行文件转换，调用接口convertWavTo16kPcm进行转换，转换后把16k pcm文件作为inFile参数传入即可。

4.3. 错误码映射 |错误事件|鸿蒙错误码|对应安卓事件|安卓错误码|描述| |-|-|-|-|-|
ERROR_VAD_NO_SPEECH	1001	ERROR_AUDIO_VAD_NO_SPEECH	3101	没有检测到说话开始
ERROR_VAD_INIT_ERROR	1002	ERROR_AUDIO_VAD_INCORRECT	3100	VAD初始化失败
ERROR_NETWORK_FAIL_CONNECT	2001	ERROR_NETWORK_FAIL_CONNECT	2000	网络连接失败

|ERROR_NETWORK_LINK_DOWN|2002|~|~|网络连接断开（识别中，系统ws触发close回调）|

|ERROR_NETWORK_ERROR|2100|ERROR_NETWORK_NOT_AVAILABLE|2100|网络错误|

|ERROR_AUDIO_RECORDER_OPEN|3001|ERROR_AUDIO_RECORDER_OPEN|3001|录音机打开失败|

|ERROR_USER_CANCEL|7002|ERROR_EMPTY_RESULT|7002|用户调用exitAsr|

|ERROR_AUDIO_RECORDER_NO_PERMISSION|9001|ERROR_NO_RECORD_PERMISSION|9001|没有录音机权限|

4.4 Debug功能

4.4.1 日志打印

```
 // 设置关闭日志。
LogUtil.isLog = false;
// 设置打开日志。
LogUtil.isLog = true;
```

4.4.2 debug音频保存

```
 // 设置保存路径，即可保存debug音频
fileRootPath = this.context.filesDir
ConfigUtil.setDebugAudioPath(fileRootPath)
```

5 在线sdk鉴权:**** 在线sdk分为4种激活方式，实际使用种选择其中一种方式进行激活，4种激活方法为：

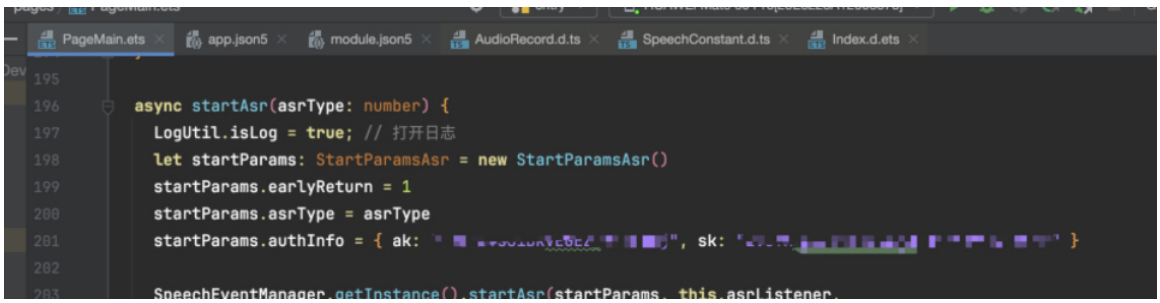
激活方式	说明
appkey secretKey激活	在百度云网站申请自己的语音合成应用后，会有appKey、appSecret鉴权信息，使用该方法进行在线鉴权，每次启动app需激活一次，不退出app永久有效
永久iamKey激活	需在官网申请的永久iamKey，使用该方法进行在线鉴权，每次启动app需激活一次，不退出app永久有效
accessToken激活	需在使用appkey和secretKey获取的access_token，使用access_token进行授权，该token存在时限，时间到期后需要从新获取新的token从新授权
临时iamKey激活	与accessToken类似，需获取临时iamKey，该key存在时限，时间到期后需要从新获取新的key从新授权

appKey secretKey激活方式：

百度云网站上申请自己语音合成的应用后，会有appId、appKey、appSecret及android包名 4个鉴权信息，申请地址为百度云应用列表

申请成功后在MainActivity中setOnlineParam方法中填写apiKey和 secretKey进行授权:

设置ak和sk:



永久iamKey激活方式：

’永久iamKey激活‘和appKey secretKey激活’ 类似 将官网申请的永久iamKey放到项目中

```
LogUtil.isLog = true; // 打开日志
let startParams: StartParamsAsr = new StartParamsAsr()
startParams.earlyReturn = 1
startParams.asrType = asrType
startParams.authInfo = { ak: "[redacted]", sk: "[redacted]" }
startParams.token = "填入您的iamKey";

SpeechEventManager.getInstance().startAsr(startParams, this.asrListener,
```

临时accessToken和临时iam key激活方式：

accessToken和临时iamKey为相同接口，接口会随时间失效，客户需要在callback中创建http请求获取到对应token返回给sdk,sdk会在过期前3分钟通过callback重新向客户端获取最新token

accessToken获取地址：https://cloud.baidu.com/doc/AI_REFERENCE/s/um3zhy50e

iamKey获取地址：https://cloud.baidu.com/doc/AI_REFERENCE/s/Hm5us339w

```
her.ets | SpeechConstant.d.ts | Index.ets | oh-package.json5

private startParams: StartParamsAsr = new StartParamsAsr()

tokenCallback: TokenCallback = {            设置临时tokenCallback接口，此处为网络请求，通过ak，sk获取accesstoken
  getToken: async(): Promise<TemporaryToken | null> => {
    try {
      // 构建请求URL
      const getAccessTokenUrl: string = this.buildRequestUrl();

      // 同步发送网络请求（鸿蒙http支持同步模式）
      const response = await this.sendSyncGetRequest(getAccessTokenUrl);

      if (!response.result) {
        hilog.error(0x00A03, TAG, "获取token失败：响应为空");
        return null;
      }

      // 解析JSON响应
      let result: Record<string, string> = JSON.parse(response.result.toString());
      hilog.debug(0x00A03, TAG, `token响应：${JSON.stringify(result)}`);

      // 提取access_token
      let accessToken: string = result['access_token'];
      if (!accessToken) {
        hilog.error(0x00A03, TAG, "获取token失败：access_token不存在");
        return null;
      }

      // 计算过期时间（当前时间 + 3分钟 + 5秒缓冲）
      const currentTime = Date.now();
      const expireTime: number = currentTime + (60 * 3 * 1000) + 5000;
okenCallback › getToken() › getAccessTokenUrl › buildRequestUrl()
```

将accessToken或iamKey 还有过期时间返回给sdk

```
    }

    // 计算过期时间（当前时间 + 3分钟 + 5秒缓冲）
    const currentTime = Date.now();
    const expireTime: number = currentTime + (60 * 3 * 1000) + 5000;

    // 返回TemporaryToken实例
    this.startParams.token = accessToken
    this.startAsr();
    return new TemporaryToken(accessToken, expireTime);
```

将callback传递到sdk中



```
}

private async initialAsr(asrType: number): Promise<void> {
  // sdk设置ak sk方式
  this.asrListener.setAkSk(Index.API_KEY, Index.SECRET_KEY);
  this.startParams = this.asrListener.initAsrAkSk(asrType)
  // sdk设置iamkey方式
  // this.asrListener.setIamApiKey("Bearer bce-v3/ALTAK-8nqtLpZLdzjKbzUnOJrqG/b70efc2c
  // this.startParams = this.asrListener.initAsrWithKey(asrType)

  // sdk设置token鉴权方式
  // this.asrListener.setTokenCallback(this.tokenCallback);
  // this.startParams = this.asrListener.initAsrWithToken(asrType)
  this.startAsr();
}
```

错误码汇总

大模型声音复刻API

流式在线合成错误码

建立连接阶段，若请求错误，服务器返回错误码与说明如下：

错误码	状态信息	错误码	错误消息	说明
400	Bad Request	216100	Invalid 'foo' value: bar.	参数错误
400	Bad Request	216100	Invalid 'foo' value: bar, range: [min, max].	参数错误
400	Bad Request	216101	Missing required parameter: 'foo'.	参数缺失
400	Bad Request	400	websocket: the client is not using the websocket protocol: ...	WebSocket协议错误
401	Unauthorized	216100	Invalid access_token parameter.	access_token参数格式不正确
401	Unauthorized	216100	Invalid Authorization header.	Authorization请求头格式不正确
401	Unauthorized	216101	Missing access_token parameter or Authorization header.	缺少鉴权参数
401	Unauthorized	110	Access token invalid or no longer valid	access_token 校验不通过。
401	Unauthorized	111	Access token expired	access_token 过期，建议更新获取Access token 重新请求。
401	Unauthorized	217	IAM auth error	API Key校验不通过
403	Forbidden	6	No permission to access data	appid没有相应接口权限，可以检查console页面app管理中勾选的AI能力情况。
404	Not Found			输入的url错误
429	Too Many Requests	15	Open api concurrency limit reached	触发并发限流
429	Too Many Requests	17	Open api usage limit reached	无剩余可用额度
400	Bad Request	216404	Voice id not exists.	voice_id不存在
500	Internal Server Error	282000	Internal error.	服务器内部错误
400	Bad Request	282101	The voice_id service has not been started.	服务未启动
400	Bad Request	216403	Not allowed to use this voice.	对该voice_id无权限

语音识别API

短语音识别错误码

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。

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	对控制台内app进行编辑，添加语音权限
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	并发超限额
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

错误码	用户输入 / 服务端	含义	一般解决方法
3300	用户输入错误	输入参数不正确	请仔细核对文档及参照 demo，核对输入参数
3301	用户输入错误	音频质量过差	请上传清晰的音频
3302	用户输入错误	鉴权失败	token字段校验失败。请使用正确的API_KEY 和 SECRET_KEY生成。 或并发、调用量超出限额。 或音频采样率不正确（可尝试更换为16k采样率）。 或者自训练平台 lm_id 不属于该账号
3303	服务端问题	百度服务器后端繁忙	有可能是原始音频质量过差。可以请将 api 返回结果和原始音频反馈至工单
3304	用户请求超限	用户的请求并发超限	请降低识别 api 请求频率 （并发以 appId 计算，移动端如果共用则累计）
3305	用户请求超限	用户的日 pv（日请求量）超限	请开通付费，购买调用量资源（账号内所有应用 APPID 共用调用量限额）
3307	服务端问题	语音服务器后端识别出错问题	有可能是原始音频质量过差。可以将 api 返回结果和原始音频反馈至工单
3308	用户输入错误	音频过长	音频时长不超过 60s，请将音频时长截取为 60s 以下，特别是 amr 格式
3309	用户输入错误	音频数据问题	服务端无法将音频转为 pcm 格式，可能是长度问题，音频格式问题等。 请将输入的音频时长截取为 60s 以下，并核对下音频的编码，采样率 16000，单声道，小端序，16bits。
3310	用户输入错误	输入的音频文件过大 或 len 参数过大	文件内容过大，音频时长不能超过 60s
3311	用户输入错误	采样率 rate 参数不在选项里	目前 rate 参数支持 16000、8000，填写其他值即会有此错误。
3312	用户输入错误	音频格式 format 参数不在选项里	目前格式仅仅支持 pcm，wav 或 amr，如填写 mp3 即会有此错误
3313	服务端问题	语音服务器解析超时	请将 api 返回结果反馈至工单
3314	用户输入错误	音频长度过短	用户的 len 参数小于等于 4
3315	服务端问题	语音服务器处理超时	请将 api 返回结果反馈至工单
3316	用户输入错误	音频转为 pcm 失败	使用 pcm 格式，或者确认 wav 和 amr 的采样率 16000，单声道。 wav 是否是 pcm 编码，小端序，16bits

实时语音识别错误码

报错示例：

```
{
"err_msg": "asr authentication failed",
"err_no": -3004,
"log_id": 106549729,
"result": "",
"sn": "0d25b5e1-ffa6-11e9-8968-f496349f3917",
"type": "FIN_TEXT"
}
```

err_no = 0 是正确，非0值即为报错。 在err_msg中一般有更具体的报错信息。

错误码 err_no	错误方	原因	解决方法	是否会识别并关闭连接
-3001	百度服务端问题	后端从SDK读数据失败，可能是sdk检测到后端超时，断开了连接	重试	是
-3003	百度服务端问题	服务后端异常	记录报错json并反馈	这一句话报错，不影响后续识别
-3004	调用方问题	鉴权失败，没有权限调用接口	查看appid appkey devpid这3个参数有无填对，QPS有无超限	是
-3005	调用方问题	音频质量问题，无效音频	根据start_time和end_time可以听下是否是噪音	这一句话报错，不影响后续识别
-3006	百度服务端问题	服务后端VAD异常	记录报错json并反馈	是
-3008	调用方问题	开始参数帧参数错误，包括漏写	检测参数有无缺失及格式正确，在err_msg中有更具体的报错信息	是
-3011	用户网络问题	上行流建立失败，可能跟网络状况或后端有关	重试	是
-3012	用户网络问题	下行流建立失败，可能跟网络状况或后端有关	重试	是
-3013	百度服务端问题	服务后端异常	记录报错json并反馈	这一句话报错，不影响后续识别
-3014	调用方正常调用CANCEL	用户取消	如果发送取消帧后会有此返回，可以忽略	是
-3101	调用方问题	一段时间没发出音频数据，导致整体请求超时	确认代码有音频数据一直在发送中	是

音频文件转写任务状态及错误码

任务状态 task_status

取值	对应状态
Running	转写中
Success	转写成功
Failure	转写失败

接口错误码

error_code	error_msg	对应接口	描述
336200	internal error	1，2	内部错误，注意task_id类型非字符串
336201	unknown task id	2	未知task id
336202	invalid param: task_ids	1，2	无效参数值
336203	missing param: ‘param_name’	1，2	缺少必要参数
336204	Open api total request limit reached’	1，2	请开通付费，购买调用时长资源
336212	invalid json	1，2	请求数据为非法json
336213	missing header: 'header_name'	1，2	缺少必要header

音频转写结果错误码

err_no	err_msg	描述
3301	speech quality error.	音频文件格式不对
4001	open speech file failed	打开音频文件失败
4002	speech length too short	音频文件长度太短
4003	internal error	内部错误
4004	internal error	内部错误
4005	internal error	内部错误
4006	internal error	内部错误
4007	internal error	内部错误
4008	invalid input	无效语音
4009	speech length too short	音频文件长度太短
4010	invalid input	无效语音
4011	internal error	内部错误
4012	internal error	内部错误
4013	internal error	内部错误
4014	internal error	内部错误
4015	internal error	内部错误
4016	not support speech format	音频格式错误，不是pcm或者wav
4017	not support rate	采样率错误，非16000或8000
4018	read speech file failed	读音频文件失败
4019	internal error	内部错误
4020	internal error	内部错误
4041	invalid input	无效语音
100000	asr task internal error	任务内部错误
100001	asr task running timeout	任务运行超时
100101	illegal speech url format	url格式非法
100102	request timeout to speech url	http请求超时
100103	request error to speech url	http请求错误，检查音频url链接，如没有问题请反馈工单排查

100104	invalid response from speech url	返回无效数据，http status 非200，回包body为空等
100105	invalid speech size	回包大小超限，音频最大100MB
100201	not support speech format	不支持音频格式
100202	speech transcoding failed	文件转码失败
100203	not support pid	不支持的pid
100204	not support rate	不支持的rate
100205	audio file is faulty	无法解析格式的音频
100206	Not support vad	VAD大小超限不支持
100207	Not support channel	通道数超限不支持
100010	asr task internal error	内部错误
100020	Authentication and Rate Limit Failed	鉴权或流控失败
100022	An http exception occurred during Authentication and Rate Limitation	鉴权或流控过程中HTTP出错
200100	Invalid parameter	参数错误
200108	Invalid user id	用户id参数错误
200115	Invalid AppID or no permission	AppID参数错误

语音合成API

短文本合成错误码

常见错误码自查及解决建议

错误码	含义	可能问题	自查方案	解决办法
500	不支持输入	请求超时	1. 确认是否设置请求超时；2. 网络是否正常	如果设置超时时间，可以设置更长，切换网络好的环境
501	输入参数不正确	输出参数错误或类型不对	检查输入参数，以及参数类型	根据参数类型正确填写参数
502	token验证失败	鉴权信息过期，鉴权额度用尽	1. 重新请求；2. 确认控制台应用额度，主要需区分普通发音人、精品发音人、臻品发音人	开通对应接口的额度
503	合成后端错误	合成文本中包含非gbk编码	文本是否是gbk编码	1. 替换文本；2. 如仍无法解决，向百度侧提供包含sn的报错信息，以及请求的时间戳
513	文本错误	合成文本涉及到特殊符号	文本是否有特殊符号	1. 替换文本；2. 如仍无法解决，向百度侧提供包含sn的报错信息，以及请求的时间戳

其他错误码

错误码	含义	解决方法
6	"No permission to access data" 无权限	对控制台内app进行编辑，添加语音合成权限
13	"Open api qps request limit reached" QPS超限	降低QPS用量或在控制台内购买QPS扩容
15	"Open api concurrency limit reached"并发超限	降低并发用量或在控制台内购买QPS扩容
16	"Open api characters limit reached" 字数超限	检查对应接口额度,区分普通发音人和精品发音人
18	"Open api cluster limit reached" 集群并发超限	提交工单联系百度工作人员
330	" invalid user_id or type for limiter"	提交工单联系百度工作人员
999	"nginx forwarding failed"	提交工单联系百度工作人员

注意事项

请严格按照文档里描述的参数进行开发。请注意以下几个问题：

1. 过长的文本将需要更多的请求耗时。如果对时间敏感，请请自行按照标点切割，可以采用多次请求的方式。
2. 语音合成 rest api请求并发有限制配额，具体限额见网页里您的应用详情。如果默认配额不能满足需求，可直接通过控制台进行购买。
3. 必填字段中，严格按照文档描述中内容填写。

长文本合成任务状态及错误码

查询任务状态task_status

取值	对应状态
Running	音频合成中
Success	音频合成成功
Failure	音频合成失败

接口错误码

error_code	error_msg	对应接口	描述
336200	internal error	创建、查询	内部错误
336201	unknown task id	查询	未知task id
336202	invalid param: task_ids	创建、查询	无效参数值
336203	missing param: 'param_name'	创建、查询	缺少必要参数
336204	Open api total request limit reached	创建、查询	请开通付费，购买调用字符数资源
336212	invalid json	创建、查询	请求数据为非法json
336213	missing header: 'header_name'	创建、查询	缺少必要header

长文本合成结果错误码

err_no	err_msg	描述
100000	Inner error	内部错误
100001	Format is not supported	不支持的音频格式
100002	Failed to query texts from uri	获取文本内容失败
100003	Failed to upload results to BOS	上传音频文件失败
100010	Inner error: TTS failed	合成音频服务失败
100020	Authentication and Rate Limit Failed	鉴权流控不通过，请检查额度
100022	A http exception occurred during Authentication and Rate Limitation	HTTP网络交互失败

流式文本合成错误码

连接响应错误码

err_no	err_msg	描述	解决方案
400	Bad Request	参数错误、参数缺失	请参考官网文档，自查token和per
401	Unauthorized	鉴权失败	
403	Forbidden	无访问权限，接口功能未开通	
404	Not Found	输入的url错误	
429	Too Many Requests	触发限流	
500	Internal Server Error	服务器内部错误	
502	Bad Request	后端服务连接失败	

注意事项：

- 由于流式文本语音合成接口，服务端会分句合成音频，所以服务端会缓存客户端的文本，客户端需要再没有文本合成时，立刻发送此消息，否则可能丢失文本。
- 客户端发送此消息后，服务端不在接收任何客户端后续请求。

呼叫中心语音错误码

问题调试

- 日志目录`${SERVER_ROOT}/mrcp-server/log/`记录了mrcp运行的相关的日志信息，有问题时可查看mrcp_debug.log进行debug。也可查看unimrcpserver-00.log查看mrcp交付过程。
- 当前MRCP同时支持一个channel一次请求和一个channel多次请求。`log/mrcp_debug.log`中的logid形如6a373e36722811e9_2_1，其中的2表示一个channel的请求id，系统生成，不连续。可使用6a373e36722811e9搜索一个channel的全部日志。
- `${SERVER_ROOT}/mrcp-server/audio`目录保存了语音识别时上传的用户话音以及语音合成时下发的合成语音，以日志中本次请求的logid或sn的名字命名。可与日志文件中的logid进行关联排查错误。
- 本程序默认sip端口为5060，在`${SERVER_ROOT}/mrcp-server/conf/unimrcpserver.xml`中配置。如果用户修改该端口，则在**程序验证**时，需修改`${SERVER_ROOT}/mrcp-server/conf/client-profiles/unimrcp.xml`中的端口为相应的值。
- 通过`unimrcpserver_control`脚本启动服务失败时，则在`${SERVER_ROOT}/mrcp-server/`目录下，使用`./bin/unimrcpserver -r . &`命令启动，以便查看打印的错误信息。

- `./bin/unimrcpserver -r . &` 启动时报错：

```
 bash: ./bin/unimrcpserver: /opt/compiler/gcc-8.2/lib64/ld-linux-x86-64.so.2: bad ELF interpreter: No such file or directory
```

则说明bootstrap.sh没有执行，或执行失败。请参考README文件，进行gcc-8.2手动配置

- 通过 `unimrcpserver_control` 脚本启动，正常情况下等待5秒左右即可完成。如长时间打印验证信息，则很可能启动失败。请检查日志文件，`unimrcpserver-00.log` 中出现 `[asr_engine_open] init ConfigManager failed`字样时，可能是mrcp-xxx.conf配置文件格式书写错误。
- 高并发情形下，`${SERVER_ROOT}/mrcp-server/conf/unimrcpserver.xml` 中的配置需要做相应修改，详见程序内README文件。
- 高并发情形下，需要调大系统最大允许打开句柄数。在root账户下执行：

```
 echo "* soft nofile 65535"  >> /etc/security/limits.conf
echo "* hard nofile 65535"  >> /etc/security/limits.conf
```

- 高并发情况下，出现 `No buffer space available` 时，请调大相应内核参数：

```
 echo "65535" > /proc/sys/net/ipv4/xfrm4_gc_thresh
```

错误码

当识别或合成出错时，日志中会打印相应错误码，它们的含义如下：

err_no	含义
0	正确
-1	用户挂断
-2	音频质量问题，可能全是静音
-3	音频过长(超过默认55秒)
-10	语音识别/合成后端未知错误
-11	等待用户语音超时
-12	语音解码器错误
-13	与后端连接被关闭
-14	等待后端结果超时
-15	连接百度服务鉴权失败
-16	连接百度服务失败
-17	语音识别失败
-20	语音合成文本为空
-21	语音合成文本过长
-22	从服务器拉取合成文本失败
-23	语音合成文本格式错误
-24	语音合成传递参数错误

如有使用问题，可通过提交工单进行反馈。

EasyDL语音自训练平台

简介

HI，您好，欢迎使用**EasyDL语音识别**。

原语音自训练平台即日已结束公测正式上线，品牌升级更名为“EasyDL语音识别”，平台和语音识别通用接口全面打通，语音技术下任一接口开通付费即可免费训练语音识别模型，无需额外费用。

如果您在调用通用语音识别模型时遇到如下困难：

1、在垂直业务领域下通用语音识别模型准确率不满足需求，语音识别应用的场景专业词汇较集中，如医疗词汇、金融词汇、教育用语、交通地名、人名等，识别结果存在“同音不同字”的情况。例如“虹桥机场”识别为“红桥机场”；“债券”识别为“在劝”。

2、语音识别结果不准带来更高的后处理成本，并且语音识别模型针对性优化训练存在技术门槛、成本高、训练周期长。

欢迎使用EasyDL语音识别，可以通过自助训练语言模型的方式有效提升您业务场景下的识别准确率。

使用流程概述

平台使用的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快一天内即可获得专属模型。



1、**创建模型**：选择您需要训练的语音识别接口，目前支持训练**短语音识别-中文普通话、短语音识别极速版、实时语音识别-中文、呼叫中心语音解决方案**接口。填写基础信息为您的模型进行命名和功能描述，并留下您的联系方式以便于我们和您联系。

2、**系统评估**：上传您业务场景中的**真实音频和对应的正确标注文本**（尽可能覆盖全部的场景），**客观科学地评估基础模型的识别率。**根据评估结果，系统自动推荐最佳的基础模型，您可以选择任一基础模型进行训练。

3、**训练模型**：上传您**业务场景中出现的高频词汇或者是长句文本**，可以有效提升业务用语的识别率；并可以迭代训练，持续优化。

5、**上线模型**：得到满意的训练模型即可申请上线，审批通过自动上线模型。模型上线后，在语音识别的接口中配置模型参数即可使用训练后的效果。

开始使用平台前，先了解一下您需要提前准备的物料及准备建议：

> 1、【测试集（包括业务音频+准确100%的标注文本）】，用于评估基础模型识别率和训练后模型识别率，相当于准备一份“标准答案”。如果模型使用业务范围较广（例如某行业领域模型），建议测试集在1000-3000条左右评估会相对客观；如果是针对某些特定场景训练，可只提供该场景的音频测试集几十条-几百条均可，包含希望评估的业务内容即可。
>
> 2、【训练集（投入平台进行训练的文本）】，用于语言模型训练，建议文本要和测试集的内容强相关。训练文本可以放置希望提升识别效果的词汇，如业务上的固定搭配和业务关键词等，或者可以将某个词汇放在不同句式的句子中，高频出现。**影响训练效果的关键因素为“文本出现的频率”和“上下文的句意理解”等**。无需重复提交大量文本，少量关键文本即可有训练效果。

进入EasyDL语音识别

输入用户名及密码，点击“登录”，进入EasyDL语音识别。可以看到整体训练流程，点击创建模型可以直接进行模型创建，点击

模型中心可以进入到模型列表页面。



整体训练流程将按照目录栏的顺序依次操作即可。



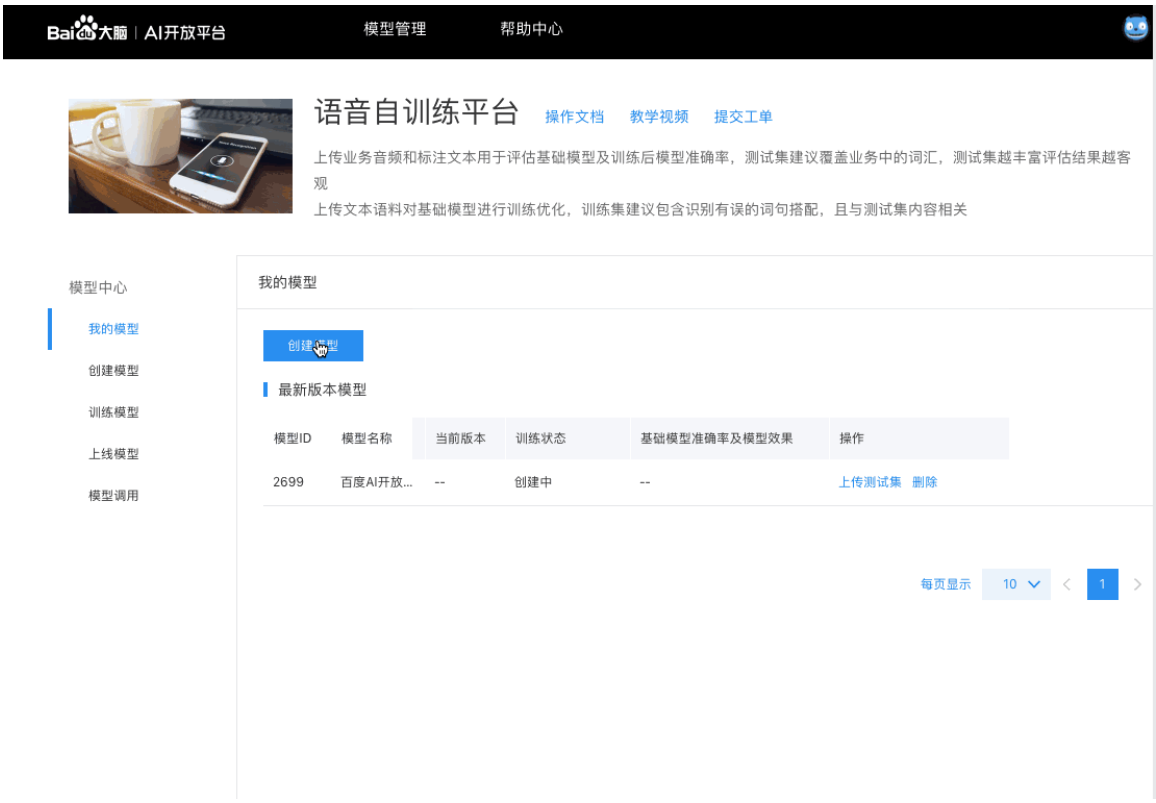
下面将详细介绍每一步的操作方式和注意事项。若遇到的问题在此文档没有找到答案，可以提交工单进行咨询。

STEP1 创建模型

在导航栏【模型中心】-【我的模型】页中可以点击【创建模型】按钮；也可以直接点击左侧导航菜单中的【创建模型】进入创建模型步骤。**目前一个账号下支持创建10个模型，模型可删除。**

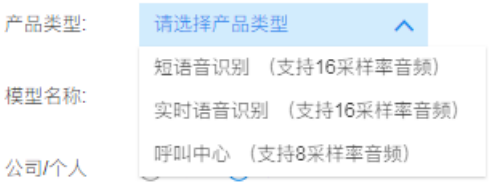
在创建模型步骤中，需要进行**“基础信息填写”“上传测试集”“选择基础模型”**三个环节完成创建。

测试集的作用为通过**上传音频和正确的标注文本评估基础模型的识别率**，根据基础模型识别率选择最合适的基础模型进行训练。等**模型训练后系统自动使用该测试集评估得到训练后模型的识别率**，可以直观的查看训练提升效果。



1、 **基础信息**：包括场景类型、模型名称、公司/个人、所属行业、应用场景、应用设备、功能描述、邮箱地址、联系方式

A、 **产品类型**：包括短语音识别（支持16K采样率音频）、实时语音识别（支持16K采样率音频）和呼叫中心场景（支持8K采样率音频）3种，用户可以基于应用场景和音频采样率来进行选择。



B、 **模型名称**：用户可自行填写模型名称，可支持中文、英文、数字、下划线.+#*()^-

C、 **公司/个人**：模型归属企业则需要填写企业名称，归属个人则不需填写

D、 **所属行业**：企业业务或个人应用所属的行业信息

E、 **应用场景**：语音识别模型应用落地的业务场景

F、 **应用设备**：业务中使用语音技术的录音设备终端

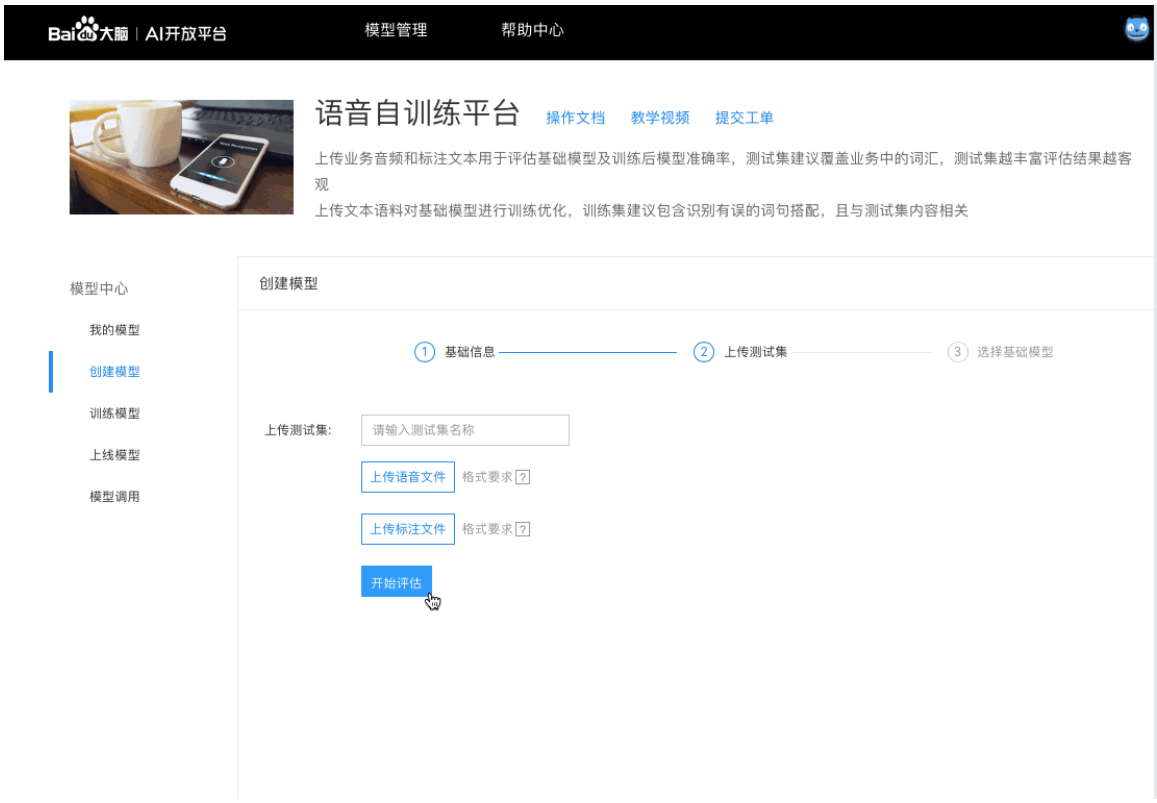
G、 **功能描述**：描述模型应用的场景，有助于上线审核哦

H、 **邮箱地址**：填写联系人的邮箱地址，用于模型上线等信息的通知

I、 **联系方式**：第一个模型需要用户填写联系方式，后面的模型系统自动复制第一个模型的联系方式（可修改） 其中，公司/个人、所属行业、应用场景、应用设备、功能描述、邮箱地址、联系方式在第一个模型中的填写信息会重复使用，后面创建的模型不用重复填写，但可修改信息

填写完毕后点击【下一步】，会在导航栏【我的模型】列表中生成一条记录保存信息，并跳转至“2、上传测试集”

2、**上传测试集**：包括填写测试集名称、上传音频文件、上传标注文件



A、 **上传测试集**：用户可自行填写测试集名称，可支持中文、英文、数字、下划线.+#*()^-

B、 **上传语音文件**：上传音频压缩zip文件（**请将所有音频文件直接压缩，请勿将音频存放在文件夹内再压缩**），格式要求：

16k 16bit单声道pcm/wav文件 8k 16bit 单声道pcm/wav文件（客服场景）； 音频文件名请不要包含中文、特殊符号、空格等字符； 所有音频需直接打包压缩为zip文件格式后上传，zip大小不超过100M，解压后单个音频大小不超过150M

C、 **上传标注文件**：上传音频的标注文本txt文件，格式要求：

标注文件内容应与音频文件相对应的内容一致(单条音频对应文本长度不超过5000字)； 标注文件格式应为txt格式，**GBK编码**；标注txt文本中，由音频名称、标注内容两部分构成，用"tab"区隔，带后缀或不带后缀均可，以下为格式示例：

01.pcm（tab换列）今天天气真不错。

上传完语音文件及标注文件，点击【开始评估】，后台进入评估状态，此时弹窗提示评估完毕时间，并自动跳转回【我的模型】。**一个账号只能同时评估一个模型**。待模型评估完毕后通过【我的模型】可以点击进入“选择基础模型”



3、 **选择基础模型**：系统根据基础模型的识别率自动推荐适合训练的基础模型，基础模型识别率超过50%才可选择进行训练。



若基础模型识别率未达到50%，请检查语音文件和标注文件内容是否匹配，若不匹配，训练结果无意义。若检查标注文件无误后识别率仍旧过低，可以加入官方QQ群进行咨询：686267521



短语音识别产品类型中目前支持对短语音识别极速版进行训练；

实时语音识别产品类型中目前支持对实时语音识别的中文普通话模型进行训练

呼叫中心产品类型中目前支持对呼叫中心语音解决方案进行训练。

选择基础模型后点击【开始训练】即可在该模型上进行模型训练。

点击“查看评估详情”可以查看测试集在基础模型上的具体识别结果，评估详情包括：字准率，句准率，插入错误，删除错误，替换错误5个指标，以及在该测试集上的具体识别结果与标注结果的对比，根据识别错误信息可以更加精准地准备训练文本。

在“查看评估详情”页点击“返回上一步”或“创建模型”可返回选择基础模型

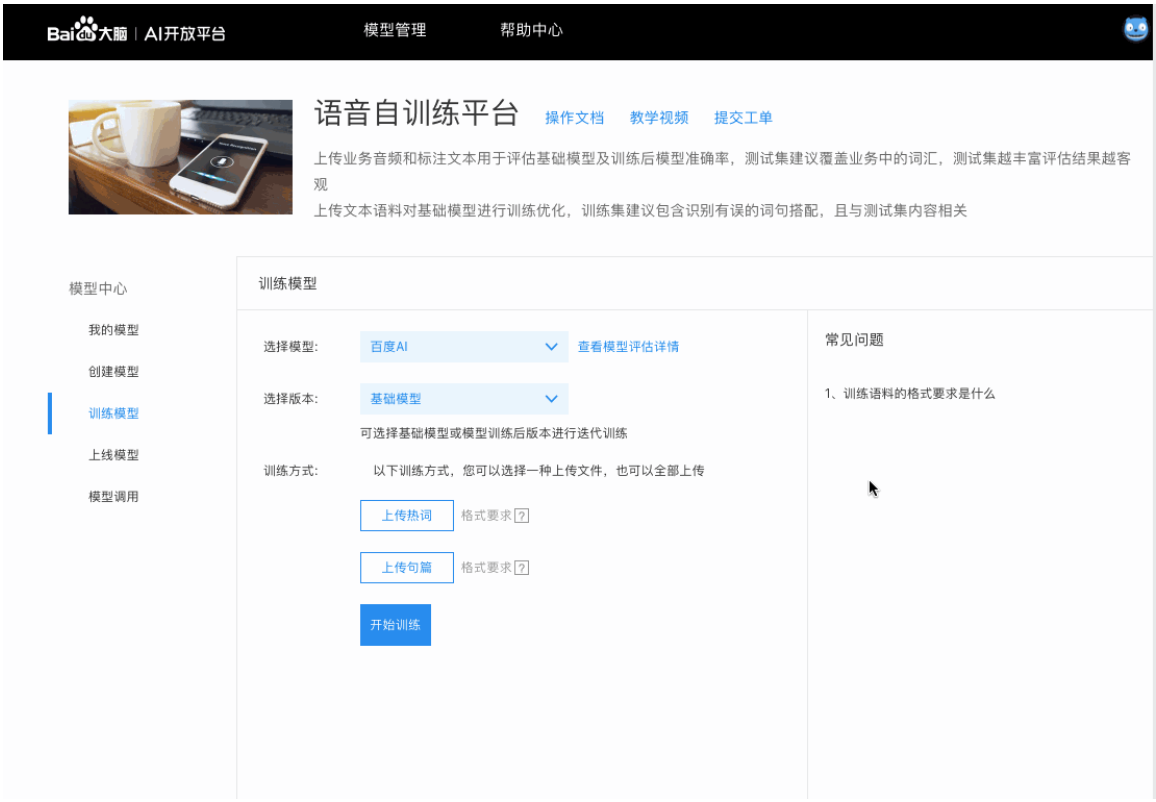
STEP2 训练模型

可以在【创建模型】-“选择基础模型“页点击【开始训练】按钮进入【训练模型】

也可以在【我的模型】列表页选择已创建完成的模型点击操作栏中的“开始训练”进入【训练模型】；

也可以直接在左侧导航栏中点击【训练模型】，进入【训练模型】

在训练模型步骤中，选择需要训练的模型，并上传训练文本。目前有两种训练方式可以选择，可以上传热词，或者是长段文本，也可以两种均上传进行训练。



热词文本格式要求：热词训练支持上传热词txt文件进行训练，每个词之间需要换行，txt格式要求gbk编码，大小不超过5M

句篇文本格式要求：句篇训练支持上传多行单句或一整段篇章（一段文字且需要符号）txt文件进行训练，txt格式要求gbk编码，大小不超过5M

建议您上传与您所需模型内容相关度较高的文本或关键词， 以便最大程度提高您的模型识别率

上传训练文本成功之后点击【开始训练】，后台进入模型训练状态，此时弹窗提示评估完毕时间，并自动跳转回【我的模型】。**一个账号下同时只能训练一个模型**。待模型训练完毕后生成新的模型版本，在【我的模型】列表页可以查看模型训练结果。



在【我的模型】列表，

模型ID	模型名称	当前版本	训练状态	基础模型准确率及模型效果	操作
1019	qwert	V1	训练完成	输入法模型:98.14% 当前版本:99.26% 训练结果详情	历史版本 申请上线 迭代训练 下载 删除

可以查看基础模型的识别率，和当前版本的识别率，了解训练提升效果

1、**训练结果详情**：可以查看训练后模型在测试集上的识别详情，包括：字准率，句准率，插入错误，删除错误，替换错误5个指标，以及在测试集上的具体识别详情。

可以进行操作

2、**历史版本**：可以查看历史训练的所有记录并进行操作

3、**申请上线**：对当前模型训练结果较为满意，可以点击申请上线，跳转至上线模型步骤

4、**迭代训练**：当前模型训练结果不满意，可以在当前版本基础上或者基础模型上继续添加新的训练语料，进行迭代训练获得新的模型版本

5、**下载**：可以下载评估模型上传的测试集和训练模型的训练集

6、**删除**：可以删除整个模型（包括所有历史版本），删除后不能恢复

STEP3 上线模型

可以在【我的模型】选择要上线的模型，在操作栏点击“申请上线”

或者在左侧导航栏中点击【上线模型】，选择要上线的模型和版本进行上线（只有模型训练成功生成版本号才可上线）



一个账号下最多只能上线3个模型。申请上线后需要后台管理员进行审核，1-3天内会有审核结果，可在【我的模型】中查看审核状态。若对审核过程有任何问题可以加入官方QQ群（群号: 686267521）咨询群管。

- **审核中**：可以查看历史版本训练情况；可以取消申请，取消后方可继续训练



- **审核失败**：问号可查看审核失败原因；可以查看历史版本训练情况；可以迭代训练或重新训练；可以重新申请上线



- **审核通过**：审核通过则后台自动上线，上线时间需要1-3天，上线过程中模型不可以做任何操作

- **上线完成**：上线完成的模型可以正式调用



STEP4 模型使用

上线通过的模型，在【我的模型】可以点击“模型调用”，查看如何使用模型

也可以在左侧导航栏中点击【模型调用】



选择您需要上线的模型（训练完成的模型才可申请上线）

——如果您选择的产品类型为**短语音识别**，则按如下操作

第一步：创建**语音技术应用**(若已创建可直接使用)，获取鉴权参数AppID，API Key，Secret Key。立即创建

第二步：获取专属模型参数 模型ID: xxxx 基础模型pid: xxxx

第三步：配置**鉴权参数**和**专属模型参数**即可使用

短语音识别极速版支持API方式调用 具体使用方法详见技术文档

——如果您选择的产品类型为**实时语音识别**，则按如下操作

第一步：创建**语音技术应用**(若已创建可直接使用)，获取鉴权参数AppID，API Key，Secret Key。立即创建

第二步：获取专属模型参数 模型ID: xxxx 基础模型pid: xxxx

第三步：根据业务情况，选择合适的调用方式，配置**鉴权参数**和**专属模型参数**即可使用

实时语音识别支持Websocket API，Android、iOS、Linux SDK方式调用。

呼叫中心模型支持MRCP server调用方式，具体使用方法详见技术文档

私有化部署方式

点击此处进行语音技术私有化部署包申请！ **申请流程**

1. 私有部署包为部署在企业内部服务器端的AI模型部署包，可部署在本地CPU、GPU服务器，支持主流Linux操作系统。
2. 私有部署包为付费项目，可自用或集成为客户使用，需具备IT能力进行部署使用。
3. 请提前使用公有云接口验证能力及估算所需并发数（QPS），提交申请后工作人员将与您联系。

部署形式

纯软件版

交付语音私有化部署包软件，可远程协助部署在本地服务器内，支持单机部署、多机部署、集群部署，适配最新主流GPU、CPU显卡。

一体机版

交付搭建了语音私有化部署包的软硬一体机服务器，支持多种配置选择，开箱即用，方便快捷。

特色优势

识别效果领先

采用百度领先的语音识别技术，特定场景下近场中文普通话识别准确率达98%，可以极大程度确保识别结果的业务可用性。

多音色语音合成

提供多种音色男声、女生音库供您选择，同时支持定制音库，让您的产品拥有个性化的声音。

数据内容安全保障

语音私有化模型可部署本地服务器，数据的存储及处理均在企业内网进行，自主掌握所有业务数据和管理权限。

专业售后服务

可提供完整售后服务支持，及时响应用户需求，保障服务稳定性。

国产化适配

语音私有化部署包可在主流CPU/GPU环境及国产化系统运行，如海光、鲲鹏等国产服务器。

快速部署

支持本地物理机快速部署，提供便捷的部署工具和常用运维文档，开机即用，高效运维。

常见问题汇总

语音识别

常见问题

1.语音识别结果与音频内容不匹配

- 语音识别返回结果与音频内容不匹配，例如：“嗨嗨嗨”、“嗯嗯嗯嗯嗯”、“什么”等错误返回。
- **解决方法**：排查音频采样率、声道、格式等参数是否符合接口规范。如与要求不符，需要用工具对音频进行转码，转码工具跳转。

2.3300错误码怎么办？

- 语音识别api使用的是HTTP POST方法， BODY里直接放置json， Content-Type头部为 application/json。 并非常见的浏览器

表单请求（application/x-www-form-urlencoded或者multipart/x-www-form-urlencoded）。

- 必填字段：format rate channel cuid token cuid token cuid token cuid token，请勿漏填。此外 (speech, len) 及 (url, callback) 这两组参数必须二选一，如果都填，默认处理第一组。
- 必填字段如format rate channel cuid token，请勿漏填。此外 (speech, len) 及 (url, callback) 这两组参数必须二选一，如果都填，默认处理第一种，并确认音频时长截取为60s以下。

3.3309错误码怎么办？

- wav和amr的音频，服务端会自动转为pcm，这个过程中导致转码出错。请确认下format及rate参数与音频一致，并确认音频时长截取为60s以下。

4.3301错误码怎么办？

- 识别结果实际为空。可能是音频质量过差，不清晰，或者是空白音频。
- 有时也可能是pcm填错采样率。如16K采样率的pcm文件，填写的rate参数为8000。

5.行业与场景限制

- 根据工信部《综合整治骚扰电话专项行动方案》、《关于推进综合整治骚扰电话专项行动的工作方案》，相关能力不得用于商业营销类、恶意骚扰类和违法犯罪类骚扰电话类场景，也不支持在贷款、理财、信用卡、股票、基金、债券、保险、售房租房、医疗机构、保健食品、人力资源服务、旅游等场景的骚扰电话营销行为。

6. 商务合作的流程是怎样的？

- 点击商务合作，详细描述您的需求。或者在百度云提交工单申请合作。

7. 一句话说完后，如何控制不自动停止识别？/ 为何我静音一段时间后，自动停止了识别？

- 关闭VAD即可。

8. 离线可以支持识别任意词识别(听写识别)吗？

- 我们推荐使用在线识别，离线识别仅支持命令词识别。如果您有强烈、明确的离线任意词识别的场景，请详细描述您的需求，在官网底部点击“商务合作”，我们会有专人联系。

9. 觉得识别不准确怎么办？

- 请对类似的语音做多次尝试，并整理录音文件、识别的日志（带sn）、期望的识别结果以及实际的识别结果文件，通过百度云工单提交给我们。

10. 为何我的App进入后台后，自动停止了识别/唤醒？

- **3.0.5.6 版本进入后台后不再打断唤醒/识别**
- iOS后台唤醒会出现很多问题导致产品可用性不能满足上线要求，同时为了规避各程序间的冲突，sdk会主动停止唤醒。

具体问题如：

- 与siri冲突，长按home键会被siri把录音能力夺取
- 顶部出现红条
- 无法唤起任何应用，即使程序自己都无法唤起
- 程序被系统杀死的问题无解，即便通过类似hack（比如某些程序后台播放音频）的方式保证存活率，也会导致播放和录音冲突

  鉴于以上考虑，程序切到后台会自动停止唤醒功能。

11.如何同时集成语音识别和语音合成？两个.a名字是一样的，内容一样么？

- 内容不一样，同时集成只需将.a换个名字重新导入即可。同名的资源文件内容是一样的。

12.集成到自身工程之后，启动报错“vad: start error”

- 一般是因为资源文件没有正确引入。请参考demo工程资源文件的引入办法，并且在代码中检查获得资源文件的指针不为nil。

13. 识别时报错：“Server app name unknown”

- 请确认填写了正确的 api_key, secret_key, app_id，且在官网正确绑定了包名。
- 请确认您的应用的并发和日配额没有超限。

14. 语音识别iOS DEMO在Xcode15中报错提示libiconv.2.4.0 不存在

- 在General-Frameworks, Libraries, and Embedded Content中移除libiconv.2.4.0，添加libiconv.2.tbd后重新编译。

语音合成

移动端SDK常见问题

1. 旧版序列号无法使用新版精品音库资源

若您在2025年11月27日之前购买或申请SN（序列号），请点击下载往期基础音库资源及往期精品音库资源，按照技术文档正常下载使用即可。

2. Android播放多段长文本

建议使用sdk内部的队列缓存。 int speak(String text, String utteranceId);

如一开始一次性调用10次speak方法，收到合成结束的onSynthesizeFinish回调后（可以收到utteranceId参数），再加入1句新的。保持sdk内部队列中一直有10个句子。

SDK内部会按照队列次序不断合成。自动按照次序播放

3. Android控制播放暂停及自定义播放

SDK是无论何种场景都会一直播放。比如有电话接入，需要暂停播放。用户需要自行实现，收到系统电话接入的事件调用pause方法， 结束后调用resume方法。

如果觉得SDK的播放方式您不满意，可以调用synthesize() 方法仅合成不播放。通过onSynthesizeDataArrived 获取音频数据，自行处理播放。

4.Android高亮当前在读的文字

onSpeechProgressChanged()根据播放音频的时长百分比回调进度，由于每个字的发音不是等长的，和实际读的文字会有误差。当一次合成的文件较长时，误差尤为明显。

5. Android播放过程中切换发音人

在线发音人， 要求在引擎空闲时切换。 SDK内部有合成队列和播放队列。其中合成队列独立，调用的句子在合成队列结束后加入播放队列。

引擎空闲是指合成队列为空，即合成全部完成。 即onSynthesizeFinish中，收到的utteranceId为最后一个。utteranceId参数是speak或synthesize方法的第二个参数。

也可以随时调用stop方式，比如最近的一句话说完时，清空播放队列和合成队列，达到引擎空闲的状态。

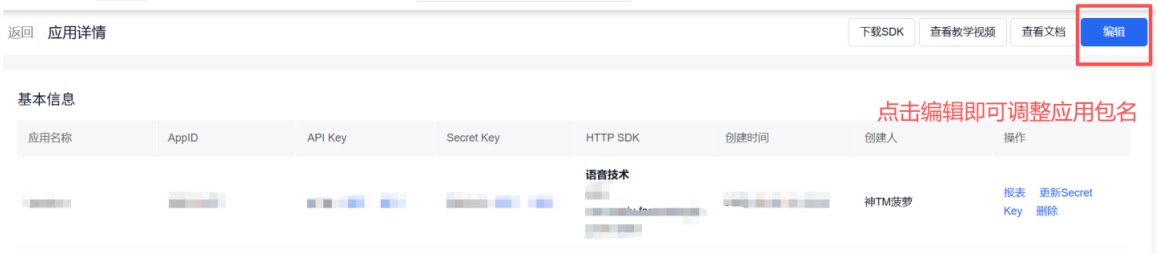
具体请参见demo的SwitchSpeakerListener类。

6.创建ak sk时包名写错，如何修改？

官网创建应用时包名写错，可通过编辑该应用进行包名更新,具体截图如下:

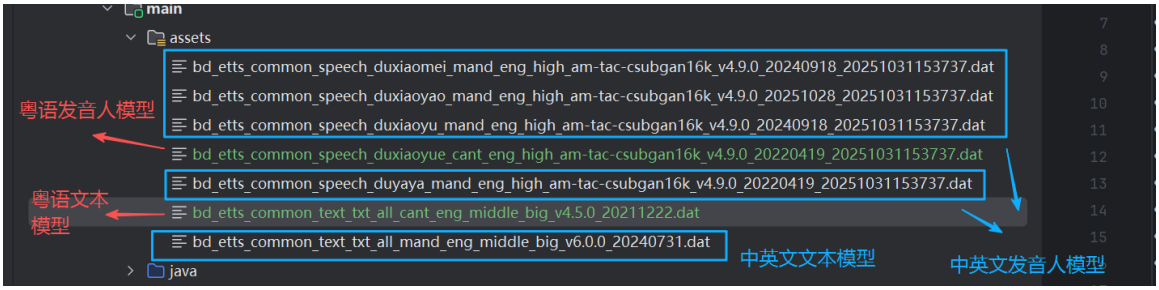


应用详情中点击编辑即可调整包名



7.创建ak sk时包名写错，如何修改？

SDK模型分为文本模型(中文、英文、粤语) 和 发音人模型(度逍遥、度小宇、度小粤等)，不同的发音人模型需要和不同的文本模型同步使用才可成功切换，例如切换粤语发音人需同步使用粤语的文本模型，可先确定文本模型和发音人模型是否对应，具体可参考发音人列表中内容，代码截图如下：



8.语音无法播报，代码执行了没声音

无法播报和无声音问题可先排查是否调用speak接口，synthesize接口为只返回合成音频不进行播报，speak为返回音频并播报若接口无误，可排查调用播报后返回的ITtsError对象是否正常 和 监听器是否返回错误提示，错出现报错可参考语音合成SDK错误码进行排查

9.合成SDK的授权码在激活一台设备后，是否可以取消激活？

不可以，SDK激活一台设备后无法取消授权过的设备，需是因新的授权码(sn)激活其他设备

10. ios如何同时集成语音识别和语音合成？两个.a名字是一样的，内容一样么？

内容不一样，同时集成只需将.a换个名字重新导入即可。同名的资源文件内容是一样的。

11. ios如何获得合成的音频数据

请参考Demo中的`synthesizerNewDataArrived`方法。注意合成的数据是逐段返回的，如果需要完整的音频，开发者自行拼接数据即可。

12. ios如何管理AudioSession

请开发者参考SDK头文件 `BDS_SYNTHESIZER_PARAM_ENABLE_AVSESSION_MGMT` 和`BDS_SYNTHESIZER_PARAM_AUDIO_SESSION_CATEGORY_OPTIONS`配置说明或者文档关于AVAudioSession部分。如果无法达成预期效果，建议收取合成的音频数据，自行播放。

13. ios如何实现后台播放、如何实现锁屏界面的播放、显示

SDK内部不会对后台、锁屏相关功能做任何控制，但会在必要的时候调用setActive接口对外部音频进行打断及恢复，会影响某些场景下的锁屏显示，如果开发者不希望SDK对AudioSession进行操作自己管理，可以通过参数配置接口，把

BDS_SYNTHESIZER_PARAM_ENABLE_AVSESSION_MGMT对应的value设置为NO，即可屏蔽SDK内部的操作。建议开发者自行收取合成音频数据，播放音频的同时来实现相关功能。

14. 商务合作的流程是怎样的？

请在AI官网底部点击“商务合作”，详细描述您的需求。或者在百度云提交工单申请合作。

15. 如何反馈使用问题

在反馈问题之前，请开发者务必在Demo测试并复现。

通过各个渠道反馈问题时，请务必详细描述以下信息：

```
 问题描述：（具体问题，稳定复现/概率）
SDK 版本：（是否已经切换到最新版本）
是否已在Demo中复现：请务必在Demo中复现遇到的错误。无法在Demo复现的问题相对难以定位原因。

若能在Demo中复现，请描述：
SDK：（SDK类型，版本号）
初始化引擎代码：（在线引擎、离线引擎代码片段）
APPID：
复现逻辑：（如何复现）
完整日志：（附上开启Debug模式的，从启动到结束的完整日志）
```

呼叫中心

常见问题：

Q：用户侧呼叫软件提示“407 COMPLETE”错误信息

A：检查log/mrcp_debug.log日志。看是否存在连接超时、请求参数不正确等错误。

Q：mrcp服务端口号如何配置？

A：在conf/unimrcpserver.xml进行配置，默认SIP端口5060。如果公司有防火墙限制，请记得打开相应端口。

Q：提示service unavailiable，MRCP session错误等

A：可能是mrcp server没有启动成功、网络不通等。如果使用MRCP自带测试工具测试正常，则可能是用户前端配置错误。

Q：提示“MRCP session has not opened after 5000 ms”错误

A：检查log/mrcp_debug.log日志。若无错误信息，则需检查FreeSWITCH配置。

Q：是否可以使用${SERVER_ROOT}/mrcp-server/bin中的asrclient进行识别？

A：asrclient仅为辅助测试验证工具，如果用来识别较长的音频，则可能会因说话停顿被VAD截断导致不能完全识别。

Q：是否有VAD功能？

A：系统自带VAD功能。

Q：语音识别时，日志中出现err_no:-2错误

A：音频不够清晰或者有环境噪声被误识别。可检查audio目录下相应音频是否清晰。相关错误码可查看`错误码`部分。

Q：是否可通过发送超时时间参数，用户在该时间内不说话则认为超时？

A：conf/mrcp-asr.conf中NO_INPUT_TIMEOUT_MS为全局默认超时时间，每次请求都有效。用户也可通过发送RECOGNIZER_HEADER_NO_INPUT_TIMEOUT自行设置单次请求超时时间，默认请求开始时开始计时。用户可通过在请求时发送RECOGNIZER_HEADER_START_INPUT_TIMERS为false设定开始时不计时，然后在合适的时机，发送RECOGNIZER_START_INPUT_TIMERS事件开始计时。

Q：是否有办法使用asrclient 批量识别多个文件 ？

A：不支持，请自行开发前端呼叫软件。

Q：合成日志出现“not supported voice name”错误

A：合成目前仅支持conf/mrcp-proxy.conf里默认配置的发音人。请不要在请求时设置SYNTHESIZER_HEADER_VOICE_NAME，或者设置为与配置中相同的发音人 `fduxiaowen`

Q：使用FreeSWITCH软电话听不到TTS声音

A：检查MRCP服务器audio目录下的pcm是否有生成的TTS音频。若存在正确音频，则MRCP TTS功能正常。需检查FreeSWITCH相关配置。

Q：在mrcp_debug.log里出现unsupportedcontent_type:application/xml是什么原因？

A：根据标准，识别请求content_type要写成application/srgs+xml、text/uri-list等。

Q：unimrcpserver.xml里面的sip-ip和sip-ext-ip应该怎么填写？sip-ip是本地ip，那sip-ext-ip呢？

A：sip-ext-ip文档中未要求填写，不用填写。

Q：mrcp server是在检测800ms无用户说话语音输入时，就认为一段话结束，执行asr吗？还是用户边说话边asr吗？

A：根据传送的语音流实时识别，默认识别时长55秒，说话间隔超过800ms截断。

Q：请问你们MRCP server返回的格式支持 NLSML 数据格式吗？

A：现在返回的识别结果，content-type就是application/nlsml+xml的。

Q：MRCP返回检测到说话得消息比较灵敏，噪声误识别有什么办法优化嘛？

A：可以在mrcp-asr.conf中DETECT_START_OF_INPUT_BY_VAD参数设置为0。参数改为0后，单字识别结果将会被忽略，只有SINGLE_WORD_WHTIL_LIST白名单中的单字可以被识别。

Q：遇到问题如何求助

A：可以将audio下相应logid的所有音频及log/mrcp_debug.log日志文件发送至bsic-support@baidu.com。

DEMO 及 SDK问题反馈

1. SDK及DEMO BUG反馈格式：
2. 现象描述 调用我们的xxx方法之后，报错。
3. 输入参数：（DEMO中含有“反馈”两个字的日志）
4. 输出结果：
5. 音频文件: 通过OUT_FILE参数获取录音音频；
6. 用户日志:先清空日志,之后调用我们的某个方法结束。请提供给我们之中的完整日志。
7. 手机信息： 手机型号， android、ios版本号等信息

问题反馈及商务沟通请联系：

邮箱：bsic-support@baidu.com

相关协议

SDK信息保护合规指引

语音合成、语音识别SDK开发者个人信息保护合规指引

亲爱的开发者：

感谢您在所开发的移动应用中集成并使用百度旗下软件开发工具包（SDK）！

百度非常重视用户个人信息保护，包括集成百度旗下语音合成、语音识别软件开发工具包（SDK）（以下简称“百度SDK”）的移动应用的最终用户（以下简称“最终用户”）个人信息保护，特制定《语音合成、语音识别SDK个人信息保护合规开发者指引》，以供您在所开发的移动应用（以下简称“移动应用”）中集成并使用百度SDK时参照执行。

一、个人信息保护合规基本要求

在所开发的移动应用中集成并使用百度SDK，您需要首先遵守以下个人信息保护合规基本要求：

1. 您应遵守收集、使用最终用户个人信息有关的所有可适用法律法规及规范性文件要求，包括但不限于《个人信息保护法》、《网络安全法》、《App违法违规收集使用个人信息行为认定方法》、《工业和信息化部关于开展APP侵害用户权益专项整治工作的通知》（工信部信管函［2019］337号）、《工业和信息化部关于开展纵深推进APP侵害用户权益专项整治行动的通知》（工信部信管函〔2020〕164号）、《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》（工信部信管函〔2023〕26号）等，保护用户个人信息安全。

2. 您的APP需要制定一份独立的隐私政策。隐私政策的内容建议通俗易懂，对您的APP收集、使用个人信息的目的、方式、范围，清晰、完整、准确地向个人信息主体进行明示告知，并充分给予最终用户独立的选择权，确保在获得个人信息主体授权同意后方可进行个人信息的处理活动。《隐私政策》应由用户自主选择是否同意，不应以默认勾选同意的方式或是以欺骗诱导的方式取得用户授权。但请您注意，仅是改善服务质量、提升用户体验、定向推送信息、研发新产品还不足以能成为要求用户同意收集其个人信息的理由。

3. 您应将在移动应用中集成并使用百度SDK服务的情况，以及百度SDK对最终用户必要个人信息的收集、使用和保护规则（具体请见语音合成、语音识别SDK隐私政策），在移动应用的显著位置或以其他可触达最终用户的方式告知最终用户（具体请参考本指引第二部分 “如何告知最终用户”及第三部分“告知文案示例”），并获得最终用户对于百度SDK收集、使用最终用户相关个人信息的完整、合法、在使用百度SDK服务期间持续有效的授权同意。如果最终用户是未满14周岁的未成年人，请您务必确保获得最终用户的父母或其他监护人对于百度SDK收集、使用最终用户相关个人信息的完整、合法、在使用百度SDK服务期间持续有效的授权同意。

4. 您应向最终用户提供易于操作的访问、更正、删除其个人信息，撤销或更改其授权同意、注销其个人账号等用户权利实现机制。

5. 您应确保在移动应用首次运行时，应在最终用户阅读并同意移动应用隐私政策之后，方可初始化百度SDK进行最终用户信息采集。

- sdk初始化接口：SpeechSynthesizer.initTts
- sdk初始化调用时机：进入语音合成界面后，onCreate中进行initTts调用，初始化成功后方可开始合成功能

6. 语音合成、语音识别SDK会根据产品升级优化、提升安全性能、法律及监管要求等原因，不断升级迭代SDK版本，不同版本的SDK获取的字段信息可能会有差异。为了保证合法合规开展合作，并切实履行保护用户个人信息的责任与义务，请您务必确保您已将APP中的语音合成、语音识别SDK升级至官方最新版本，以避免因使用旧版本SDK而出现违法违规的问题，导致您的APP遭受监管处罚的风险。语音合成、语音识别SDK更新后会及时通过官网通知、站内信、公告、邮件、短信等有效方式对您进行告知，请您及时关注并尽快更新。

除了上述个人信息保护合规基本要求外，您还应遵守您所开发的移动应用所集成并使用的语音合成、语音识别SDK隐私政策。

SDK基本业务功能与拓展业务功能：

百度语音合成SDK基本业务功能为离在线文本转语音功能/服务。百度语音识别SDK基本业务功能为离在线语音转文本功能/服务。

1、**语音合成SDK**个人信息以及权限收集

以下是**语音合成SDK**获取您的APP的最终用户的个人信息以及权限，由于不同SDK版本采集的字段与是否可选可能存在一定差异，具体采集情况以实际接入的SDK版本为准：

序号	功能及服务	个人信息类型	收集方式	适用系统版本	是否必选
1	为了判断当前是否WIFI是否连接，如WIFI已连接，则向最终用户提供在线合成功能	设备信息（即WIFI状态）	SDK直接采集	iOS及Android	必选
2	为了帮助开发者向最终用户提供功能/服务，同时为了更准确定位并解决开发者以及最终用户在使用语音合成SDK产品和服务时遇到的问题	Android ID	SDK直接采集	Android	必选
3	为了帮助开发者向最终用户提供功能/服务，同时为了更准确定位并解决开发者以及最终用户在使用语音合成SDK产品和服务时遇到的问题	OAID	SDK直接采集	Harmony	必选

为了保证最终用户能正常使用【语音合成】SDK相应功能及服务，百度智能云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，申请前百度智能云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百度智能云会开启相关权限，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
使用网络、访问网络状态	为了判断当前是否网络是否连接，如网络已连接，则向最终用户提供在线合成功能	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
访问WiFi状态、改变WiFi状态	为了判断当前是否WIFI是否连接，如WIFI已连接，则向最终用户提供在线合成功能	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
修改/删除SD卡中的内容	为了将日志等环境信息写入SD卡中，为用户提供一个安全的使用环境，同时为了更准确定位并解决开发者以及最终用户在使用语音合成SDK产品和服务时遇到的问题。如不使用日志记录功能则不需要该权限	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启

iOS系统版本

设备权限	功能及服务	权限授权方式
使用网络、访问网络状态	为了判断当前是否网络是否连接，如网络已连接，则向最终用户提供在线合成功能	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
访问WiFi状态、改变WiFi状态	为了判断当前是否WIFI是否连接，如WIFI已连接，则向最终用户提供在线合成功能	
更改音频设置	为了使开发者和最终用户可以自定义播放合成音频的音量，通过调用系统设置更改音频设置。如不使用音量调节等功能则不需要该权限	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启

Harmony系统版本

设备权限	功能及服务	权限授权方式
使用网络、访问网络状态	为了判断当前是否网络是否连接，如网络已连接，则向最终用户提供在线合成功能	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
访问WiFi状态、改变WiFi状态	为了判断当前是否WIFI是否连接，如WIFI已连接，则向最终用户提供在线合成功能	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
修改/删除SD卡中的内容	为了将日志等环境信息写入SD卡中，为用户提供一个安全的使用环境，同时为了更准确定位并解决开发者以及最终用户在使用语音合成SDK产品和服务时遇到的问题。如不使用日志记录功能则不需要该权限	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
更改音频设置	为了使开发者和最终用户可以自定义播放合成音频的音量，通过调用系统设置更改音频设置。如不使用音量调节等功能则不需要该权限	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启

为保证SDK授权激活、定位并解决开发者及最终用户在使用语音合成SDK产品和服务时遇到的问题，SDK将在必要情况下收集包括设备型号、操作系统在内的设备信息。
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。

2、**语音识别SDK**个人信息以及权限收集

以下是**语音识别SDK**获取您的APP的最终用户的个人信息以及权限，由于不同SDK版本采集的字段与是否可选可能存在一定差异，具体采集情况以实际接入的SDK版本为准：

序号	功能及服务	个人信息类型	收集方式	适用系统版本	是否必选
1	为了判断当前是否WIFI是否连接，如WIFI已连接，则向最终用户提供在线识别功能	设备信息（即WIFI状态）	SDK直接采集	iOS及Android	必选
2	为了帮助开发者向最终用户提供功能/服务，同时为了更准确定位并解决开发者以及最终用户在使用语音识别SDK产品和服务时遇到的问题	Android ID	SDK直接采集	Android	必选
3	为了帮助开发者基于最终用户的音频数据，提供将音频转换为文本的服务	音频数据	SDK直接采集	iOS及Android	必选

为了保证最终用户能正常使用【语音识别】SDK相应功能及服务，百度智能云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，申请前百度智能云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百度智能云会开启相关权限，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
使用网络、访问网络状态	为了判断当前是否网络是否连接，如网络已连接，则向最终用户提供在线识别功能	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
获取麦克风权限	为了帮助开发者和最终用户采集需要识别的原始音频数据，从而提供将音频转换为文字的功能	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
修改/删除SD卡中的内容	为了将日志等环境信息写入SD卡中，为用户提供一个安全的使用环境，同时为了更准确定位并解决开发者以及最终用户在使用语音合成SDK产品和服务时遇到的问题。如不使用日志记录功能则不需要该权限	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
获取蓝牙权限	为了帮助开发者和最终用户发现、配对本地蓝牙设备，请求、接受蓝牙连接和传输音频数据，从而实现蓝牙录音功能。如不使用蓝牙录音功能则不需要该权限	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启

iOS系统版本

设备权限	功能及服务	权限授权方式
使用网络、访问网络状态	为了判断当前是否网络是否连接，如网络已连接，则向最终用户提供在线识别功能	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
获取麦克风权限	为了帮助开发者和最终用户采集需要识别的原始音频数据，从而提供将音频转换为文字的功能	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启
获取蓝牙权限	为了帮助开发者和最终用户发现、配对本地蓝牙设备，请求、接受蓝牙连接和传输音频数据，从而实现蓝牙录音功能。如不使用蓝牙录音功能则不需要该权限	授权方式由设备系统开发方以及开发者移动应用决定；当最终用户同意向开发者移动应用授予该权限时开启

为保证SDK授权激活、定位并解决开发者及最终用户在使用语音识别SDK产品和服务时遇到的问题，SDK将在必要情况下收集包括设备型号、操作系统在内的设备信息。
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。

二、如何告知最终用户

为帮助您明确地告知最终用户百度SDK个人信息收集、使用和保护相关事宜，百度智能云为您提供了以下告知方式，供您参考执行：

1. 在移动应用隐私政策中“个人信息共享”条款部分或“所集成的第三方SDK”条款部分告知最终用户相应功能/服务由百度SDK提供，并显示相应百度SDK隐私政策链接以告知最终用户，百度SDK收集、使用的最终用户个人信息类型、目的及用途。移动应用隐私政策在用户首次打开移动应用或者在移动应用的注册/登记界面通过协议在线展示的方式向用户展示，并获得最终用户明示同意（例如：点击“同意”，或勾选“√”）。

2. 在移动应用隐私政策中“个人信息共享”条款部分或“所集成的第三方SDK”条款部分告知最终用户相应功能/服务由百度SDK提供，并参考相应百度SDK隐私政策内容，以条款或表格等形式列明收集、使用的最终用户个人信息类型、目的及用途。移动应用隐私政策在用户首次打开移动应用或者在移动应用的注册/登记界面通过协议在线展示的方式向用户展示，并获得最终用户明示同意（例如：点击“同意”，或勾选“√”）。

3. 当最终用户在移动应用中首次打开/使用相应功能/服务时，以弹窗、页面提示方式显示相应百度SDK隐私政策链接，以告知最终用户相应功能/服务由百度SDK提供，百度SDK为提供相应功能/服务而收集、使用的最终用户个人信息类型、目的及用

途，并获得最终用户明示同意（例如：点击“同意”，或勾选“√”）。

三、告知文案示例

为帮助您明确地告知最终用户百度SDK个人信息保护规则相关事宜，百度智能云为您提供了以下告知方文案示例，供您参考执行：

文案示例A

语音合成：

为向您提供【文本转语音】功能/服务，百度智能云集成了【语音合成】SDK。在为您提供【文本转语音】功能/服务时，【语音合成】SDK需要收集、使用您必要的个人信息（如设备信息，包括操作系统版本、唯一设备标识符、IMEI 等）。关于【语音合成】SDK收集、使用的个人信息类型、目的及用途，以及【语音合成】SDK将如何保护所收集、使用的个人信息，请您仔细阅读《【语音合成】SDK隐私政策》了解。

语音识别：

为向您提供【语音转文本】功能/服务，百度智能云集成了【语音识别】SDK。在为您提供【语音转文本】功能/服务时，【语音识别】SDK需要收集、使用您必要的个人信息（如设备信息，包括操作系统版本、唯一设备标识符、IMEI 等），用于判断网络环境及优化识别效果。关于语音识别SDK收集、使用的个人信息类型、目的及用途，以及语音识别SDK将如何保护所收集、使用的个人信息，请您仔细阅读《语音识别SDK隐私政策》了解。

文案示例B

为保障App相关功能的实现与应用安全稳定的运行，百度智能云可能会接入由第三方提供的SDK实现相关目的，具体接入的相关第三方SDK列明如下：

第三方SDK名称	使用目的	官网链接/隐私政策链接
百度语音合成 SDK	实现文本转语音功能	https://ai.baidu.com/ai-doc/REFERENCE/Jkdyl0v3v
百度语音识别 SDK	实现语音转文本功能	https://ai.baidu.com/ai-doc/REFERENCE/Qkdykq1r3

文案示例C

语音合成：

为向您提供【文本转语音】功能/服务，百度智能云集成了【语音合成】SDK。在为您提供【文本转语音】功能/服务时，【语音合成】SDK收集、使用您的【设备信息（即WIFI状态）】信息，用于【判断当前是否WIFI是否连接，如WIFI已连接，则向最终用户提供在线合成功能】目的/用途。具体请您仔细阅读《【语音合成】SDK隐私政策》了解。

语音识别：

为向您提供【语音转文本】功能/服务，百度智能云集成了【语音识别】SDK。在为您提供【语音转文本】功能/服务时，【语音识别】SDK收集、使用您的【设备信息（即WIFI状态）】信息，用于【判断当前是否WIFI是否连接，如WIFI已连接，则向最终用户提供在线识别功能】目的/用途。具体请您仔细阅读《【语音识别】SDK隐私政策》了解。

文案示例D

为保障App相关功能的实现与应用安全稳定的运行，百度智能云可能会接入由第三方提供的SDK实现相关目的，具体接入的相关第三方SDK列明如下：

第三方SDK名称	业务功能	个人信息类型	具体目的/用途
百度语音合成SDK	实现文本转语音功能	设备信息（即WIFI状态）	为了判断当前是否WIFI是否连接，如WIFI已连接，则向最终用户提供在线合成功能
百度语音识别SDK	实现语音转文本功能	设备信息（即WIFI状态）	为了判断当前是否WIFI是否连接，如WIFI已连接，则向最终用户提供在线识别功能

四、使用SDK服务的合规注意事项

1. 您接入语音合成、语音识别SDK前，应当仔细阅读SDK的协议约定、本合规规范、用户协议、隐私政策等内容，并依据相关内容对您APP的《隐私政策》及您APP收集、存储、使用、共享等处理个人信息的情况进行合规自查。
2. 您知悉并认可语音合成、语音识别SDK具备收集个人信息的功能，并认可该等信息的收集均为双方合作之必要目的所需。
3. 您承诺已制定并按照相关要求公示您APP的《隐私政策》，并已清晰、明确、显著地说明有关通过SDK收集个人信息的必要性、收集数据的范围、方式以及用途。同时，您应确保在APP首次运行时以弹窗等合规方式显著提示用户阅读您APP的《隐私政策》并取得用户的合法授权同意，经过合法授权后再初始化语音合成、语音识别SDK进行个人信息的收集与处理。
4. 您已认真阅读并理解语音合成、语音识别SDK平台协议、合作规范、隐私政策、接入文档等约定和要求，并承诺在您使用语音合成、语音识别SDK服务期间，针对语音合成、语音识别SDK收集、使用、处理、共享、转让相关个人信息，您已取得了用户持续有效的授权和同意，并保证您不会违反国家相关法律法规、相关国家标准以及双方约定的目的。
5. 如果您的APP面向不满十四周岁的儿童及未成年人用户提供服务，您承诺遵守儿童个人信息保护及未成年人保护相关的法律法规，您承诺已采取相关措施并保证已获得其监护人的授权同意。
6. 如因您违反语音合成、语音识别SDK的平台协议、合作规范、隐私政策、接入文档等约定，导致您的用户或第三方对百度主张任何形式的索赔或权利要求，或导致百度因此产生任何法律纠纷的，您将负责解决并承担全部责任，如因此给百度及其关联主体造成损失的，您应赔偿因此给百度及其关联主体造成的全部损失。
7. 您保证对于您从语音合成、语音识别SDK获取的数据，无论是在合作期间或是合作停止后，均承担保密义务，不擅自提供、泄露、透露给任何第三方，并应采取一切合法措施以使上述数据免于散发、传播、披露、复制、滥用及被无关人员接触，避免导致相关数据被超出双方合作目的使用。