

# ISUDA 文档



#### 【版权声明】

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

#### 【商标声明】



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

#### 【免责声明】

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



| 目录         |     |
|------------|-----|
| 目录         | 2   |
| 功能更新记录     | 6   |
| 产品描述       | 13  |
| 产品简介       | 13  |
| 基本概念       | 14  |
| 应用场景       | 15  |
| 技术白皮书      | 16  |
| 爱速搭与amis   | 22  |
| 产品定价       | 22  |
| 快速入门       | 22  |
| 使用流程       | 22  |
| 创建应用       | 22  |
| 连接数据源      | 23  |
| 创建数据模型     | 23  |
| 页面设计       | 23  |
| 权限管理       | 23  |
| 发布应用       | 23  |
| 操作指南       | 23  |
| 组织管理       | 23  |
| 工作台管理      | 23  |
| 应用管理       | 23  |
| 组织架构及成员管理  | 25  |
| 角色管理       | 27  |
| 基本信息管理     | 30  |
| 页面设计       | 32  |
| 页面类型说明     | 32  |
| 普通页面设计     | 39  |
| 页面管理       | 107 |
| 应用管理       | 109 |
| 应用权限设置     | 109 |
| 应用设置       | 111 |
| 应用发布       | 121 |
| 组件模板导入     | 125 |
| 数据模型       | 129 |
| 外部数据源接入    | 129 |
| 外部数据源IP白名单 | 129 |
| 外部数据源使用    | 130 |
| 外部数据源接入    | 142 |
| 数据源编辑      | 145 |
| 模型其他设置     | 149 |

|               |     |
|---------------|-----|
| Baidu 百度智能云文档 | 目录  |
| 模型基本设置        | 150 |
| 模型字段配置        | 155 |
| 模型数据管理        | 178 |
| API对接         | 180 |
| API中心         | 180 |
| API编排         | 184 |
| 基础            | 193 |
| 本地开发调试        | 196 |
| 应用搭建          | 197 |
| 页面设计          | 197 |
| 页面设计器介绍       | 197 |
| 页面设计简介        | 204 |
| 应用发布          | 208 |
| 版本管理          | 208 |
| 1. 简介         | 208 |
| 1.1 功能简介      | 208 |
| 1.2 应用场景      | 208 |
| 2. 操作步骤       | 208 |
| 2.1 查看详情      | 208 |
| 2.2 版本对比      | 209 |
| 2.3 版本回滚      | 210 |
| 多环境           | 211 |
| 应用发布          | 211 |
| 1. 概述         | 211 |
| 2. 操作步骤       | 212 |
| 对象存储          | 213 |
| 对象存储管理        | 213 |
| 配置对象存储        | 216 |
| BPM引擎         | 219 |
| BPM概述         | 219 |
| 快速入门          | 222 |
| 流程设计          | 225 |
| 如何设计一个流程      | 225 |
| 节点描述          | 227 |
| 流程运行          | 254 |
| 资源管理          | 257 |
| 图片管理          | 257 |
| 音频管理          | 262 |
| 典型实践          | 267 |
| 配置一个列表        | 267 |

|                   |     |
|-------------------|-----|
| Baidu 百度智能云文档     | 目录  |
| 视频教程              | 268 |
| 教学视频大纲            | 268 |
| 平台快速入门            | 270 |
| 三分钟完成应用搭建         | 270 |
| 组织工作台             | 270 |
| 实体管理              | 270 |
| 页面管理              | 270 |
| 门户管理              | 270 |
| 页面设计器             | 270 |
| 应用管理              | 270 |
| 应用设置              | 270 |
| 应用发布              | 271 |
| API管理             | 271 |
| 移动应用常用容器          | 271 |
| AI原生应用            | 271 |
| 开发助手              | 271 |
| 业务助手              | 272 |
| AI原生应用快速搭建        | 273 |
| 学习资源获取            | 273 |
| 学习资源获取            | 273 |
| 学习资源获取            | 273 |
| 功能模块详解            | 273 |
| NPM组件包            | 273 |
| NPM扩展包高阶功能介绍      | 273 |
| 流程引擎              | 273 |
| 流程引擎6_排他网关功能介绍    | 274 |
| 流程引擎10_服务调用节点功能介绍 | 274 |
| 流程企微通知            | 275 |
| 服务编排              | 275 |
| 页面设计              | 275 |
| 高级功能              | 275 |
| 场景专题              | 276 |
| 阶段1:快速了解          | 276 |
| 快速创建一个应用          | 276 |
| 场景应用搭建            | 276 |
| 供应商管理-9.营业执照识别功能  | 277 |
| 应用市场模板操作讲解        | 278 |
| 采购供应商管理           | 278 |
| 采购供应商管理           | 278 |
| 博物馆预约系统           | 278 |
| 电子化访客登记           | 278 |

|               |        |
|---------------|--------|
| Baidu 百度智能云文档 | 功能更新记录 |
| 公司庆典答题操作讲解    | 278    |
| 运维软件一站式服务台    | 278    |
| 客户线索跟进系统      | 278    |
| 邀请函           | 278    |
| 预算报销管控系统      | 278    |
| 客户关系管理系统      | 278    |
| 员工打卡系统        | 279    |
| 活动抽奖模板        | 279    |
| 爱速搭考核与知识管理系统  | 279    |
| 零代码           | 279    |
| 零代码应用         | 279    |
| 常见问题          | 279    |
| 私有部署相关问题      | 279    |

## 功能更新记录

| 发布时间    | 功能概述                                                                                                                                                                                                                                                                                                                                                                                 |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2023-12 | <div><div><div></div></div><div>报表引擎</div><div><ul style="list-style-type: none"><li>X轴和Y轴的配置项得到了增强，包括联动和视觉关联、字体样式（包括颜色）的配置、轴线颜色和宽度配置，以及相关的主题样式设置</li><li>环形图、数值指标、仪表盘组件的样式设置，包括支持主题设置、宽高可视化配置、标题位置配置</li><li>Flex布局和容器组件的宽度设置，支持弹性、适配设置，并展示最大和最小宽度配置项</li><li>报表主题数据色支持实时切换，报表支持运行时主题切换</li><li>报表和数值指标的通用样式配置进行了调整，支持新版通用样式，包括内外边距配置、渐变启动配置项、渐变模式配置以及渐变透明度配置</li></ul></div></div> |
|         | <div><div><div></div></div><div>数据集</div><div><ul style="list-style-type: none"><li>编辑器，支持节点拖拽，数据模型节点和sql节点；</li><li>支持勾选基础字段，未勾选的字段不返回</li><li>自定义字段支持，分组、数据通、表达式</li><li>关联关系，支持左连接、右连接、内连接</li><li>支持将数据集作为图表数据来源</li></ul></div></div>                                                                                                                                           |

- AIGC
  - 对话式生成单实体、修改实体属性，包含属性配置、添加字段、字段配置、添加关系、添加索引
  - 对话式生产amis组件
  - 服务编排支持大模型服务调用
  - 服务编排大模型prompt模板百宝箱
  - 自然语言生成图表组件
  - 爱速搭文档/amis问答助手

- 弹窗优化升级，含：1) 从sub-editor弹出框模式改为页面内可视化；2) 新增弹窗设计视图，所见即所得设计弹窗；3) 新增弹窗大纲和弹窗列表；4) 优化弹窗和属性、外观配置
- 应用级组件模板/报表组件 支持使用数据模型、API中心和外部API数据（支持跨域）
- 报表/模型数据源/筛选条件和聚合筛选条件支持页面变量、页面参数和应用变量等
- 列表、卡片列表组件升级，含：1) 选项设计方式取消弹框模式，可编辑器内直接设计；2) 数据绑定可视化；3) cards支持行列间距设置
-

|        |                                                                                                                                                                                                                                                                                         |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2023-7 | <ul style="list-style-type: none"><li>• 弹窗容器支持数据可视化，包括弹窗内的数据和其直接父级的数据（注意弹窗内的数据不分层，同名即覆盖）</li><li>• 表达式编辑器升级为新样式，功能同前端模块表达式优化内容</li><li>• 支持根据汇报关系逐级审批，支持设置审批至指定人/指定级数</li><li>• 弹窗容器支持数据可视化，包括弹窗内的数据和其直接父级的数据（注意弹窗内的数据不分层，同名即覆盖）</li><li>• 支持OpenAPI，可获取流程待办、流程状态、流程处理人等信息</li></ul>    |
| 2023-5 | <ul style="list-style-type: none"><li>• 实体模型底层升级，优化底层构建逻辑</li><li>• 外部数据源扩展多种数据源类型（推荐优先使用MySQL和Oracle12+）</li><li>• 数据模型实体设计操作交互优化，新增模型右侧面板，提供更快捷的操作</li><li>• flex布局组件属性配置合并和交互优化</li><li>• 报表除数值指标组件外12个组件均支持事件动作，能通过事件动作满足简单的报表联动下钻场景</li><li>• 优化状态组件、表格编辑框、富文本编辑器组件配置项面板</li></ul> |
|        | <ul style="list-style-type: none"><li>• 支持用户使用应用时看到所有文本的多语言，包括但不限于常见的各类名称、描述及通知消息等</li><li>• 支持在应用管理的多语言管理界面中配置语言设置和语料设置，语料支持单条操作或批量上传</li><li>• 支持应用发布时控制发布应用支持的语言，并生成对应的语料数据，存放到发布产物中</li></ul>                                                                                       |

|         |                                                                                                                                                                                                                                                                                            |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2023-3  | <ul style="list-style-type: none"><li>● 报表组件增加通用外观样式设置，支持内外边距、边框、背景色等通用样式配置</li><li>● 报表除数值指标组件外12个组件均支持事件动作，能通过事件动作满足简单的报表联动下钻场景</li><li>● 支持自定义异常处理规则，如节点错误时自定义错误处理逻辑</li><li>● 数值指标和环形进度图组件支持「上下文」作为数据源</li></ul>                                                                       |
| 2022-12 | <ul style="list-style-type: none"><li>● API编排连接器节点扩展短信发送能力</li><li>● 流程引擎支持流程超期规则设计及自定义任务页</li><li>● 前端事件动作设计支持防抖功能、冒泡控制可视化配置</li><li>● 前端样式设计支持主题风格设计器包含布局类、表单类等17个组件的样式自定义</li><li>● 前端页面设计上线20+组件优化包括选人组件、布局容器、部分表单项组件等</li><li>● 平台集成模块支持对接飞书和企微的消息通知和事件订阅</li></ul>                   |
| 2022-11 | <ul style="list-style-type: none"><li>● 新增数据字典，支持组织/应用级范围内统一、规范字典项内容</li><li>● 新增基础风格设计器，解决平台/应用级的主题定义与复用，支持颜色/文字/边框/尺寸/阴影/图标可视化配置</li><li>● 新增表单项静态展示态，表单项支持在展示态、只读、禁用、编辑之间切换，使得表单查询、编辑页面一次设计完成</li><li>● 新增柱状图、折线图、饼图、漏斗图报表图表组件，新增报表组件外观可视化配置</li><li>● 新增平台国际化，支持平台级切换简体中文、英文</li></ul> |



|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | <ul style="list-style-type: none"><li>更新应用时新增支持保留部分应用基本数据，如应用名称、logo、环境变量、权限、用户认证、数据等</li><li>动作新增支持数据追加控制，解决前端组件联动的可视化配置的场景问题</li></ul>                                                                                                                                                                                                                                                                                                                                                         |
| 2022-7 | <div>1.</div> <div>组件功能与交互升级，提升组件可视化配置体验</div> <ul style="list-style-type: none"><li>优化15个系统组件，属性面板交互、组件设计交互体验提升，主要包括<ul style="list-style-type: none"><li>表单项：级联选择器、链式选择器、数字框、滑块、颜色框</li><li>展示类：超链接、头像</li><li>容器类：锚点导航、iFrame、容器、面板、web-component</li><li>按钮类：下拉按钮、按钮组、按钮点选</li></ul></li><li>事件动作配置面板升级，增强覆盖度，提升配置体验<ul style="list-style-type: none"><li>扩充更多组件支持事件监听和特性动作配置</li><li>优化动作分类，支持搜索，帮助快速定位所需动作</li><li>设置组件数据动作支持可视化选择字段名（针对表单），降低配置成本</li><li>修复可视化配置展示问题</li></ul></li></ul> |

|         |                                                                                                                                                                  |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         | <ul style="list-style-type: none"><li>支持30+组件默认值可视化配置，可通过公式编辑器快速关联数据域内其他变量并进行计算</li><li>组件列表icon升级</li></ul> <p>2. 应用独立部署Java版本请升级到 jdk11</p> <p>3. 优化部分已知问题</p> |
| .....   | .....                                                                                                                                                            |
| 2021-10 | <ul style="list-style-type: none"><li>开发环境支持全局搜索查找功能，可以一次性搜索应用内的页面、API、公共组件、自定义组件</li><li>增加组织移交功能</li><li>应用支持发布部分页面</li></ul>                                  |
| 2021-09 | <ul style="list-style-type: none"><li>JSSDK 支持直接下载文件，可以直接下载生成的文件、模型列表的预设条件支持枚举和日期类型字段</li><li>api 中心支持类似页面的历史版本功能，方便多人协作</li><li>编辑器右侧的代码编辑支持全屏模式</li></ul>      |
| 2021-08 | <ul style="list-style-type: none"><li>新增页面顶部访问历史功能，并将面包屑放入页面头</li><li>新增页面复用通用配置功能、数据模型字段支持自定义正则验证器、模型组件支持行列权限设置</li></ul>                                       |
| 2021-07 | <ul style="list-style-type: none"><li>工作台也支持按名称搜索</li></ul>                                                                                                      |
|         | <ul style="list-style-type: none"><li>自定义组件支持样式文件和编辑器插件扩充功能</li></ul>                                                                                            |

|         |                                                                                                                                                                                                                                                          |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2021-06 | <ul style="list-style-type: none"> <li>• 新增「API 编排」功能、输出「JSSDK」功能</li> <li>• API 中心支持 mock 功能</li> </ul>                                                                                                                                                 |
| 2021-05 | <ul style="list-style-type: none"> <li>• 增加存储导出功能，用于 MySQL 版本迁移</li> <li>• API 中心支持不标准的 json 格式、支持自定义脚本来进行结果转换、支持自定义标识、支持显示 api 使用情况</li> <li>• 角色中的用户支持批量删除</li> <li>• autoFill 支持 multiple 模式</li> <li>• 左侧导航支持自定义 icon</li> <li>• 页面添加水印开关</li> </ul> |
| 2021-04 | <ul style="list-style-type: none"> <li>• “简易页面” 更名为 “模型页面”</li> </ul>                                                                                                                                                                                    |
| 2021-03 | <ul style="list-style-type: none"> <li>• 支持应用级别关闭代理</li> <li>• 支持在应用内查看操作日志</li> <li>• 优化：自定义组件调试 schema 自动保存</li> </ul>                                                                                                                                 |
| 2021-03 | <ul style="list-style-type: none"> <li>• 支持 API 401 报错的时候跳转到对应的地址</li> <li>• 优化普通页面打开性能</li> <li>• 首页新增首次访问引导</li> <li>• 支持导入组织架构</li> <li>• 新增 API 接口，支持应用角色相关的控制</li> <li>• 私有部署版本支持平台级别后端自定义签名插件</li> <li>• 修复：页面资源权限问题</li> </ul>                    |
| 2021-03 | <ul style="list-style-type: none"> <li>• 支持 Redis 集群模式</li> <li>• 支持日志输出到控制台</li> <li>• 性能优化</li> <li>• 应用内支持创建应用级别角色</li> </ul>                                                                                                                         |

# 产品描述

## 产品简介

[爱速搭](#)是百度智能云推出的低代码开发平台，它灵活性强，对开发者友好，在百度内部大规模使用，有超过 4w 内部页面是基于它制作的，是百度内部中台系统的核心基础设施。

它具备以下功能：

### 1. 页面制作

1. 基于 [amis](#) 页面可视化编辑，具备超过 120 种内置组件，对于大部分中后台页面可以做到无需了解前端就能制作。
2. 支持自定义前端组件，可以通过代码扩充定制组件。
3. 支持导出前端项目，可以运行在任意前端框架中。
4. 支持页面模板，可以自定义模板，快速搭建公司内常见页面。
5. 支持自定义主题，具备 1000+ 细粒度展现控制，可自定义 CSS。
6. 自适应移动端布局。

### 2. 数据模型

1. 具备数据模型能力，可以直连外部数据库，做到对于简单增删改查需求无需后端。
2. 支持所有数据库基础字段，并扩展了十几种高级字段类型，并支持对字段的校验和索引设置。
3. 支持软删除、记录操作时间。
4. 支持设置数据字段验证。
5. 支持设置索引规则。
6. 支持批量导入导出数据。
7. 支持自动生成基于模型的增删改查页面。
8. 支持一对一、多对一、一对多、多对多关联关系。
9. 支持通过 SQL 的方式对数据进行处理，灵活实现各种数据操作功能。

### 3. API 对接

1. 具备 API 对接能力，可以直连外部接口。
2. 支持百度云 BOS、S3、阿里云 OSS 协议的对象存储对接。

### 4. 业务编排

1. 支持可视化编辑后端逻辑，可以对多个接口进行组合。
2. 支持 HTTP、SQL 方式获取数据。
3. 支持并行和串行调用。
4. 支持循环、分支进行流程控制。
5. 支持脚本进行自定义扩展。

## 5. 业务流程

1. 权限控制，可以控制每个节点能查看和编辑哪些数据。
2. 支持动态查找处理人，比如上级。
3. 流程流转判断、加签、会签。
4. 支持打回到上个节点或前面某个节点。
5. 流程通知。

## 6. 权限控制

1. 具备基于角色的细粒度资源控制。
2. 可以针对页面内组件粒度进行控制，比如某个按钮对某些用户不显示。
3. 可以对某个页面创建分享链接。
4. 所有操作日志，所有操作都会记录，便于后续审计。

## 7. 多环境

1. 支持多套环境，相互之间数据隔离，开发时不影响正式版本的使用。
2. 支持版本管理，可以通过回滚快速还原。
3. 平台升级不会影响应用已发布版本。

## 8. 私有部署

1. 支持私有部署，不限制实例数，环境只依赖 Docker、MySQL、Redis，安装简单、运维成本低。
2. 支持对接 oauth 2.0，也可以支持自定义登录协议。
3. 可以对接自己的组织架构。

## 9. 支持 OpenAPI 对系统进行操作。

爱速搭目前的客户有知名的银行及互联网公司，比如「[知乎](#)」，商业咨询请联系邮箱 [aisuda@baidu.com](mailto:aisuda@baidu.com)。

# 基本概念

### ● 组织

组织是爱速搭的主要入口，组织可以理解为一家公司，它可以有自己的组织架构，爱速搭中的每个用户都属于一个或多个组织。

在爱速搭 SaaS 版本中有多个组织，相互之间的用户是隔离的，而在私有部署版本中只有一个组织。

### ● 应用

在一个组织中能创建多个应用，每个应用就是一个独立的系统，比如一个 CRM 应用，其中包括页面及应用内角色。

### ● 页面

应用中可以创建多个页面，应用在实际使用时就是访问这些页面，爱速搭支持6种页面类型：

- 普通页面 (amis)
- 模型页面
- 富文本页面
- Markdown 页面

- 外部链接
- 嵌入外部页面

页面会在左侧导航中显示，有树形层级关系。

- **数据源**

数据源就是指数据库，可以是爱速搭内置的数据库，也可以是外部数据库，比如自己搭建的 MySQL 数据库。

- **数据模型**

数据模型就是数据库中的表，爱速搭可以在线设计数据库表结构，自动对数据库表结构进行变更，可以建立多个表的关联关系。

- **流程**

流程是指业务审批流程，可以控制每个节点的操作权限。

- **API 中心**

API 中心是用于管理后端 http 接口。

- **多接口聚合**

多接口聚合功能在 API 中心里，它可以对多个后端接口进行组装，前端只需要调用一个接口。

- **amis 组件**

爱速搭所有前端界面都是通过 amis 组件来渲染的，amis 是一种低代码渲染引擎，它支持 120+ 组件，可以基于它完成大部分中台页面的开发。

- **页面模板**

页面模板是事先创建好的页面，可以基于它来快速创建常见页面，可以自定义自己的专属模板。

## 应用场景

爱速搭平台适用于多种不同类型企业的多样化场景企业应用开发、系统项目实施与数字化建设。

### 信息采集、标注与审核

爱速搭可快速构建各类表单类型应用，实现对用户、业务数据的采集、加工处理和汇总分析。可应用于用户反馈、市场调研、信息征集、线索收集、信息登记、数据标注等场景。

### 产品管理后台配套建设

灵活的数据对接特性，可小时级构建复杂产品服务管理中心，满足产品服务运营管理与快速交付。可用于开发业务管理后台、运营平台、审核平台、CMS 等。

### 办公应用搭建与完善

可基于 BPM、数据模型等能力，根据实际企业办公场景诉求，搭建各类适合自身需求的办公应用，并支持与已有系统打通，可开发行政审批系统、客户管理、项目管理、HR 管理系统等应用。

### 企业 ERP 系统建设、迁移与扩展

可按需快速构建库管、生产、调度、客服等各业务系统，并接驳打通已有系统，形成架构统一、灵活升级的一体化 ERP 平台，支持库存管理、生产管理、销售管理、财务管理、客服管理等系统开发。

### IT 智能化改造

爱速搭融合有更多的 AI 集成特性，可帮助企业快速搭建或改造具备智能化能力的工业应用，实现业务升级，满足工业质检、智慧巡检、生产管理、产能分析等场景的需求。

## 客户项目定制实施

爱速搭可根据业务需求快速实现定制化开发，需求变更成本低，更可免除运维烦恼，实现高效交付，完成行业综合解决方案(如政务、教育、物联网、工业、能源等)快速实施交付。

## SaaS 软件研发

爱速搭基于 AI、数据集成、可视化和云服务能力，可快速研发输出具备 AI 特性的企业 SaaS 应用，实现智能化开发企业 SaaS 应用生产与交付中台。

## 数字化中台建设

爱速搭可与智能云数据中台、AI 中台和微服务产品深度融合，联合构建企业数字化中台，帮助企业轻松实现应用开发/管理中台、信息化管理中心。

# 技术白皮书

这里将介绍爱速搭设计理念及核心技术点，它在技术上和其它低代码平台有什么不同。

## 🔗 设计理念

爱速搭最初用户是百度内部开发者，因此它最重要的设计理念是「开发者优先」。

我们认为面向非开发者的零代码平台使用场景有限，只能做简单的办公应用，大部分时候还不如在线 Excel 简单方便，因此爱速搭不是零代码平台，它面向的是有一定开发经验的用户。

爱速搭低代码平台的目标不是取代开发者，而是辅助开发者更高效地完成工作。

因为重视开发者，爱速搭比起其它低代码平台有三个显著特点：

- **灵活性高**，我们重视灵活性和功能可扩展，让开发人员可以最大程度发挥，而不是为了易用性处处设限，这点在后续的前端和后端设计中都有体现。
- **开放和透明**，低代码平台最大的风险是技术锁定和黑盒，作为开发者我们自己也不喜欢封闭的低代码平台，因此爱速搭重视开放性，前端方面渲染器开源了，后端方面也选择了开发人员最熟悉的传统数据库技术，没有中间层，对开发者是透明的，并且爱速搭后端不依赖任何云厂商，可以部署到任意环境。
- **架构上松耦合**，比起大而全的集成系统，我们选择了松耦合的架构，这样能让开发者有更多选择，可以选择只用前端、只用可视化编辑器、只用后端，可以很好和现有开发结合。

接下来将介绍爱速搭中前后端及编辑器的核心设计。

## 🔗 前端

爱速搭前端使用了自主开发的开源项目 amis，它的 star 数量有 7.1k，并且一直在持续更新，目前大量互联网及金融公司使用，包括但不限于：腾讯云、滴滴、阅文集团、[贝壳](#)、光大银行等知名企业。

amis 渲染器的核心是将 JSON 配置渲染成可交互的页面，比如下面一段简单的配置：

```
{
  "type": "form",
  "api": "/saveForm",
  "body": [
    {
      "label": "Name",
      "type": "text",
      "name": "name"
    },
    {
      "label": "Email",
      "type": "email",
      "name": "email"
    }
  ]
}
```

使用 amis 渲染之后就成了一个表单：

| 表单            |                       |
|---------------|-----------------------|
| Name          | <input type="text"/>  |
| Email         | <input type="email"/> |
| <div>提交</div> |                       |

可以看到 amis 配置非常简洁，没有多余的内容，即便从来没学过也能看懂。

amis 可以支持多种场景使用，你可以用它来制作：

- 带导航的多页面应用。
- 单个页面。
- 已有页面（任意前端框架）的某个小区域，比如只用 amis 做表单。

amis 的核心技术点有：

- 支持无限层级嵌套，可以做出复杂的前端界面。
- 除了界面，还包括数据获取、表单验证、数据联动、显隐切换、交互的低代码配置。
- 大量内置组件（120+），开箱即用。
- 扩展容易，使用标准 React 组件扩展，同时爱速搭还支持使用 vue 和 jquery 扩展。

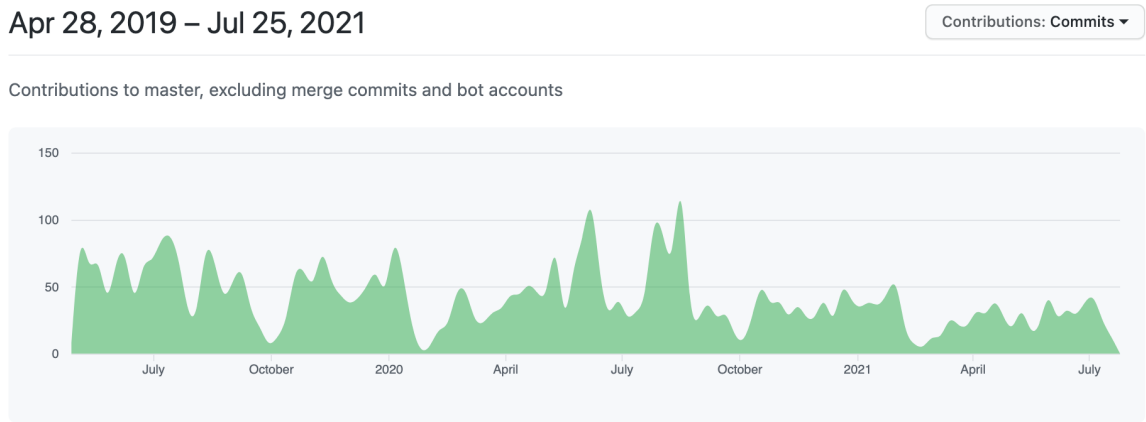
amis 低代码开发比起传统开发的主要优势：

- 节省前端人力，百度内部百家号分析过，一个季度内 94% 的中台页面无需前端。
- 统一前端技术栈，并且未来无需重构
  - 百度内部 5 年前的页面配置现在依然能渲染，完全不需要修改，体验还随着 amis 的升级更好了，我们将会一直坚持向下兼容，即便再过 5 年也没问题。



- 而传统方式每过两年就不得不重构，angular/vue/react 最新版本都不向下兼容的，不重构意味着得不到最新的功能，文档也不好找，重构意味着需要花费大量研发时间，而重构本身并不创造产品价值。
- 质量有保证，因为是配置，所以不管是谁写出来的都一样，哪怕完全交给外包开发，也不担心交付的代码质量，百度内部实际上已经大规模转成外包开发，让宝贵的研发资源花在面向用户的产品上。

amis 已经经历了 5 年的发展，至今依然在快速迭代，是有专人维护的持续项目。



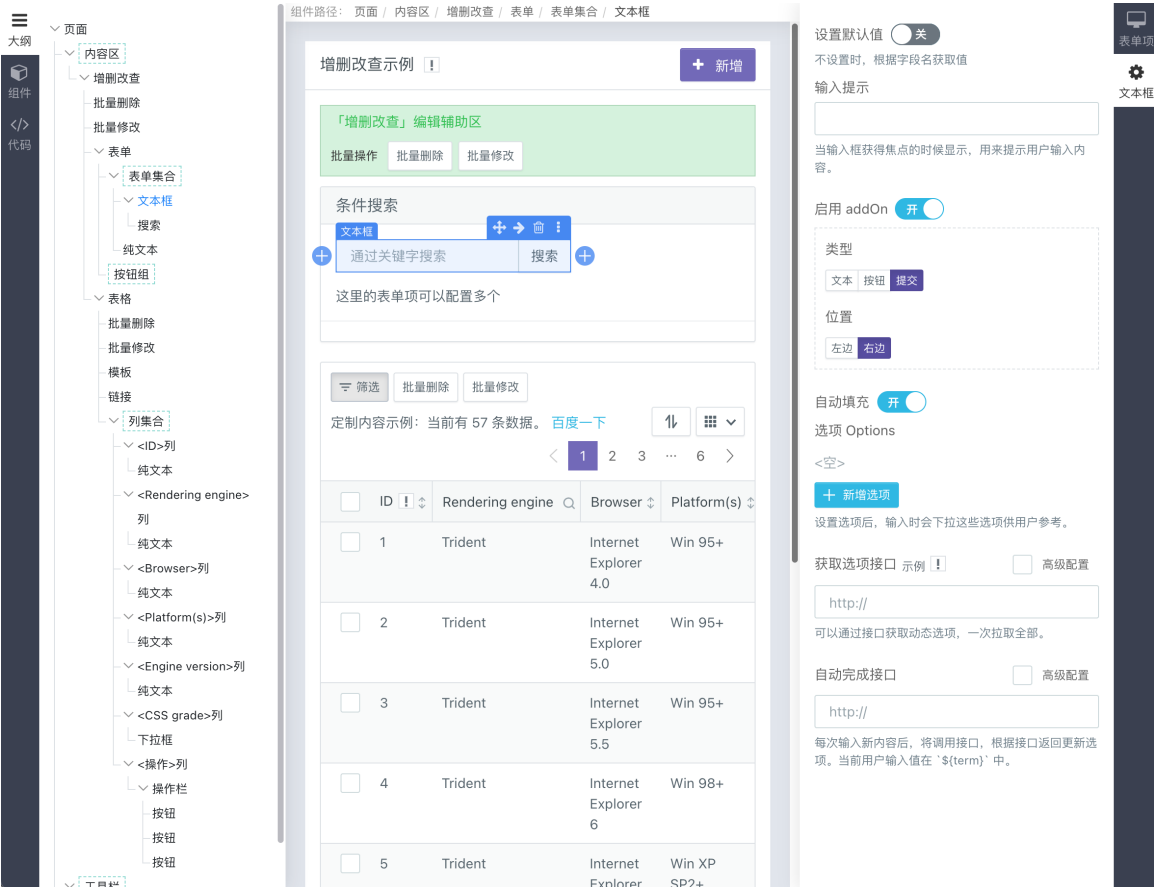
可视化编辑器

很多低代码平台的编辑器都只支持一个层级编辑，下图是某个开源表单设计器的截图：



可以看到它所有字段都是平铺的，只有通过栅格来进行简单的左右布局，无法支持复杂嵌套场景，比如表格列中嵌套表单、复杂的二维布局等。

而爱速搭的编辑器支持树形编辑，可以用它制作复杂页面。



编辑器的核心技术点有：

- 支持 amis 的无限层级编辑。
- 支持拖拽编辑、右键菜单。
- 支持代码模式和可视化双向编辑，满足初学者和熟练者的不同需求。
- 树形导航，代码可以聚焦到细粒度组件，只编辑选中组件的配置。
- 支持移动端编辑和预览。
- 灵活简单的扩展机制，扩展自定义组件的编辑面板，编辑面板是 amis 表单，因此只需要会 amis 就能制作编辑器插件。

后端数据模型

爱速搭后端的数据模型是基于传统数据库，这和业界低代码平台的常见做法不一样，业界常见有两种做法：

- 基于非关系型数据库，最常用的是 mongo，大量零代码平台都是基于它开发的。
- 基于已创建好的宽表，比如预先创建 500 列，然后分配这些列给不同字段。

但这两种做法都有明显缺点：

- 数据是孤岛，因为数据类型特殊，因此只能存储在平台中，难以打通现有应用数据，只能通过 api 的方式交换数据，这也意味着这些低代码平台都只适合做全新应用，无法用于现有核心系统的开发，无法直连现有系统的数据库。
- 开发人员不熟悉，mongo 在国内依然小众，无 schema 既是优点也是缺点，没有 schema 容易导致新老数据结构不一致而引起问题，目前未听说有大公司在核心系统中使用 mongo，并且它的查询语法也远不如 SQL 普及。
- 有大量限制，基于已创建的宽表使得无法使用所有 SQL 语句，通常只支持查询，无法用 `select *`、无法创建 view 视图等，数据库各种高级功能几乎都没法用。
- 性能难以预估，而且因为要经过中间层转换，性能将有损，对开发者来说是个黑盒，出现性能问题将不好排查。

爱速搭的数据模型采用了更传统的做法，它的原理是动态数据库 ORM 映射，可以理解为动态创建了 Entity 类，然后通过这些

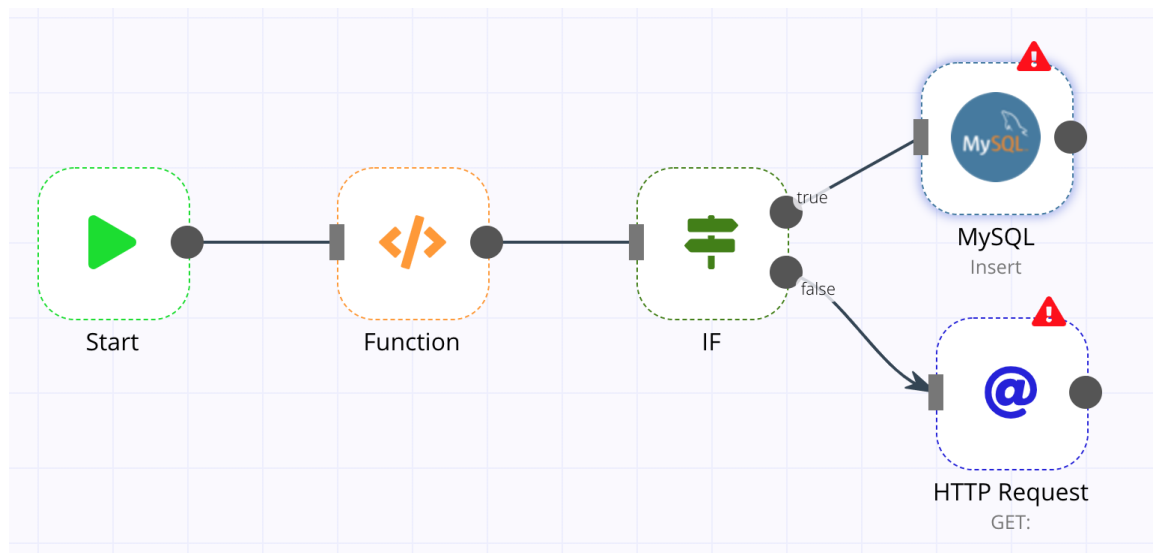
类去操作数据库。

这意味着在爱速搭中：

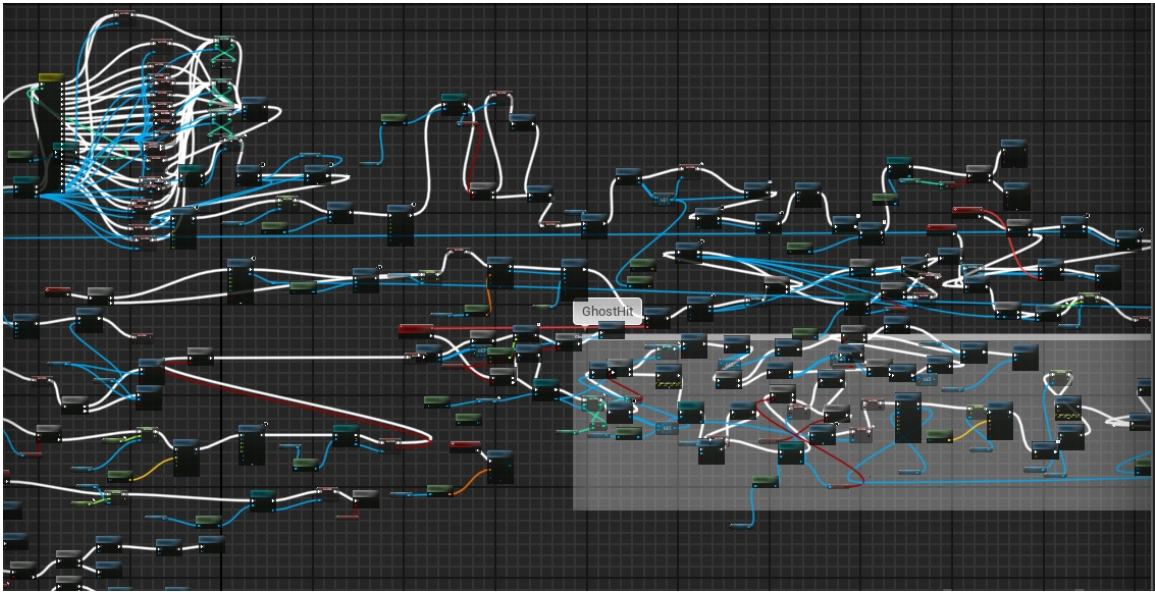
- 数据模型就是数据库表，其中的字段就是表中的字段，对字段的修改是自动生成数据库的表结构变更操作，这 and 传统开发是一样的，开发人员都熟悉。
- 可以直连已有数据库，数据不再是孤岛，爱速搭可以和传统开发混搭，可以用在现有成熟系统中，而不是只能做新应用。
- 既可以像操作对象那样便捷操作数据，也能使用任意 SQL 语句来实现复杂功能，实现和普通开发一样的灵活度。
- 性能和稳定性可预期，因为没有中间层转换，性能上限取决于数据库本身。
- 多环境独立，爱速搭应用后端支持多环境，每个环境完全独立。
- 所有页面信息、导航、API 中心等发布后就变成只读了，每个环境都是独立的。
- 数据模型将使用不同的库，数据不会相互影响。
- 发布之后所有前端静态文件都会保存当时的历史版本，即便平台做了不兼容升级也不会导致应用页面出问题。

## 业务编排

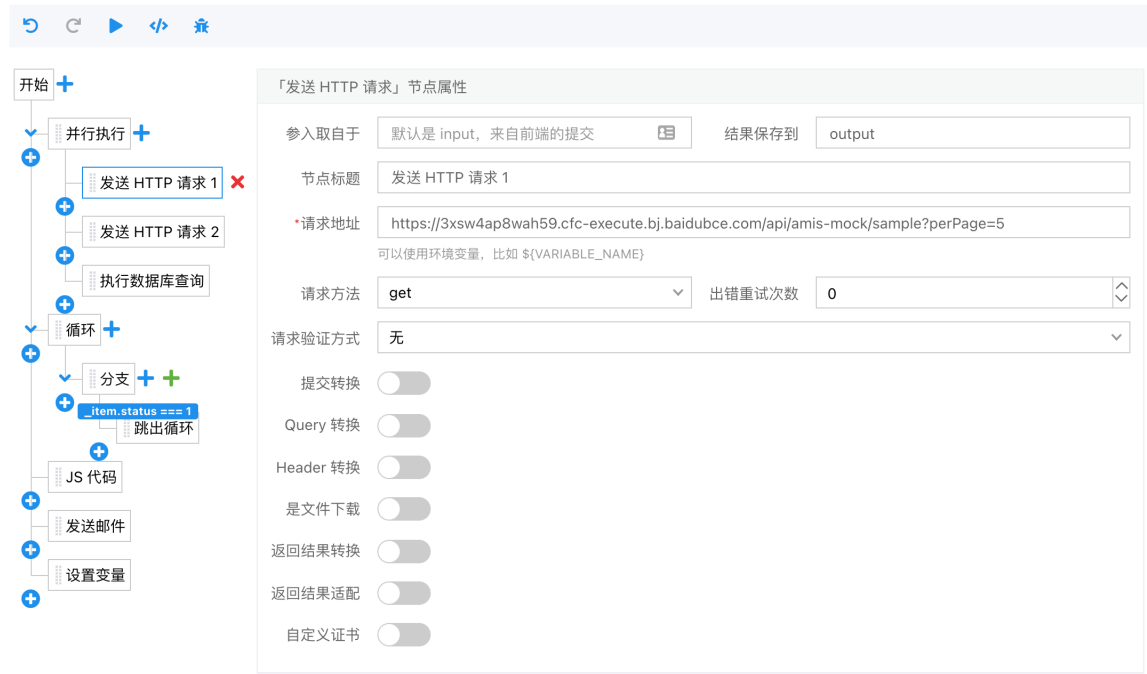
业务编排是一种可视化后端逻辑的能力，不少低代码产品使用的方式是基于图及连线的方式来编辑，比如下面是某个开源编排的截图：



使用图连接的方式虽然看起来漂亮，但实际使用体验不好，尤其是业务复杂的时候，连线会很乱，比如下面是一个复杂点的例子，可以发现这时已经很难看出执行逻辑了：



因此爱速搭选择了开发人员更为熟悉的树形结构来进行业务编排：



使用这种方式有如下优点：

- 1. 开发人员容易理解和看懂，它和写代码的思路是一样的，从上往下执行。
- 2. 同时爱速搭支持生成伪代码的能力，方便开发者检查。
- 3. 可以构建深层结构，并且能将它折叠起来方便一层层阅读，比起连线更简洁易懂。

爱速搭在百度

爱速搭在百度主要用户中后台前端开发，在百度所有核心部门都有使用，包括搜索、手机百度、社区、AI、自动驾驶、系统部、百度云等，更是用在了许多重点项目中，比如两年前百度春晚红包，对红包分配的实时调度控制界面就是基于爱速搭开发的。

爱速搭在百度的大规模普及并不是因为强制使用，而是百度产品线自主选择的，体现出了爱速是大量开发人员都认可的平台。

百度内部比较大的产品线基于爱速搭制作了上千个页面，最复杂的页面有超过 1 万行配置。

因为百度自己内部大规模使用，爱速搭的可靠性和长期维护都有保障。

## 爱速搭与amis

[amis](#) 是爱速搭团队开源的前端低代码框架，爱速搭应用中的页面都是基于 amis 渲染的，同时爱速搭平台自身的绝大部分页面也是基于 amis 开发。

amis 的定位是前端 UI 框架，类似于 Element、Ant Design，可以基于它快速开发前端页面，但需要自己整合到自己内部系统中，并且实现路由功能。

爱速搭是完整的低代码平台，它除了页面开发，还实现了如下功能：

- 对接可视化编辑器
- 页面左侧导航
- 页面顶部访问历史
- 页面多版本管理
- 自定义组件
- 主题定制
- 页面操作日志

同时爱速搭还具备后端能力，包括以下功能

- 应用多环境
- 页面外部 api 代理
- 业务编排
- 业务流程（暂时关闭）
- 对接对象存储
- 角色权限管理
- 数据模型
- OpenAPI

## 产品定价

爱速搭应用智能低代码搭建平台面向企业客户开通免费试用权益，请通过官网申请表单提交，会有工作人员进行审核并通过邮件告知审核结果。免费试用过程中请勿用作商业开发平台或应用于任何商业用途。如有商业需求，请联系爱速搭团队进行商业应用洽谈。

### 商业用途计费规则

商业用途收费标准为按套餐收费，支持公有云服务和私有化服务，不同套餐类型价格不同，详细信息请联系爱速搭团队进一步沟通。

商业合作咨询请联系邮箱 [aisuda@baidu.com](mailto:aisuda@baidu.com)。

## 快速入门

### 使用流程

### 创建应用

- 连接数据源
- 创建数据模型
- 页面设计
- 权限管理
- 发布应用
- 操作指南
- 组织管理
- 工作台管理

🔗 工作台管理

工作台是组织环境的操作台，会根据登录成员的权限和角色不同，展示不同的应用。



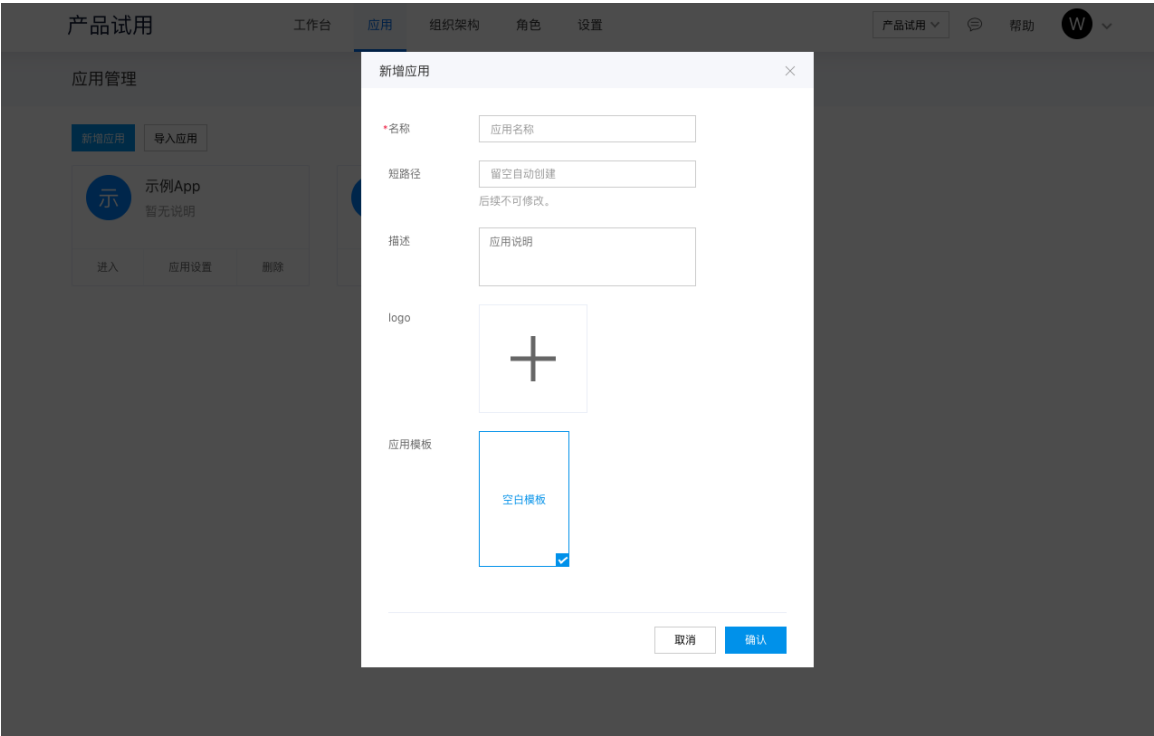
应用管理

🔗 应用管理

1. 点击导航中的【应用】，进入应用管理页面。



2. 应用管理页面支持通过新建的方式新建应用。



3. 或通过导入应用包完成新建，原应用包内的模型、页面等资源都会保留。



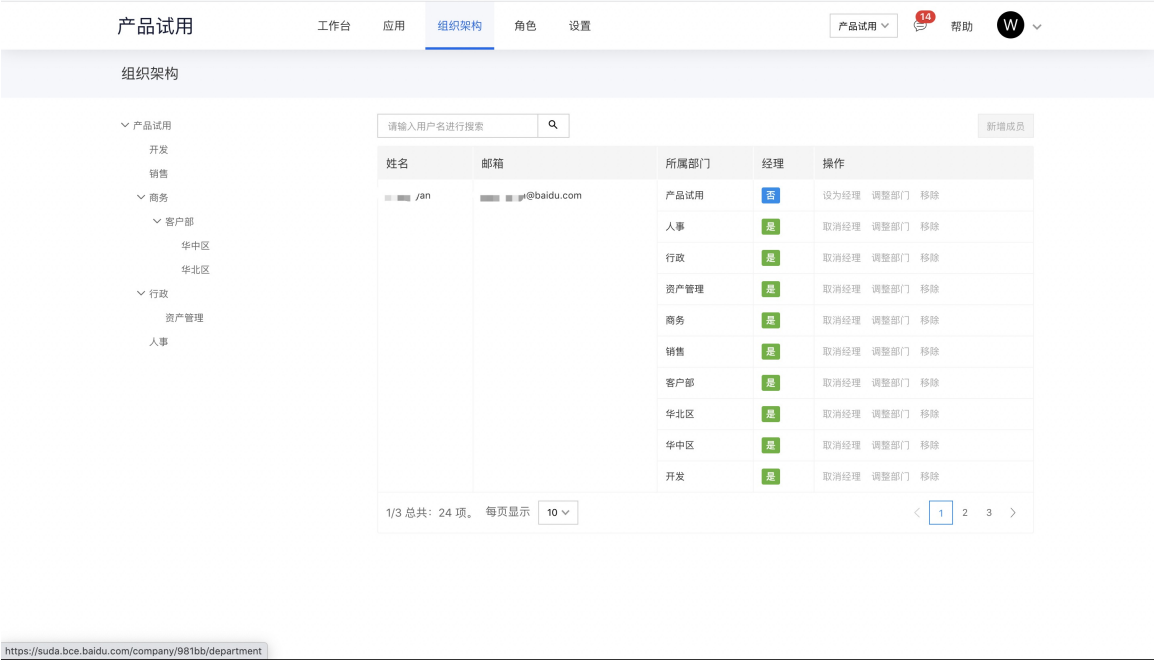
4. 每个应用都可以直接进入编辑环境或进入应用设置，也可以便捷进行删除操作。



组织架构及成员管理

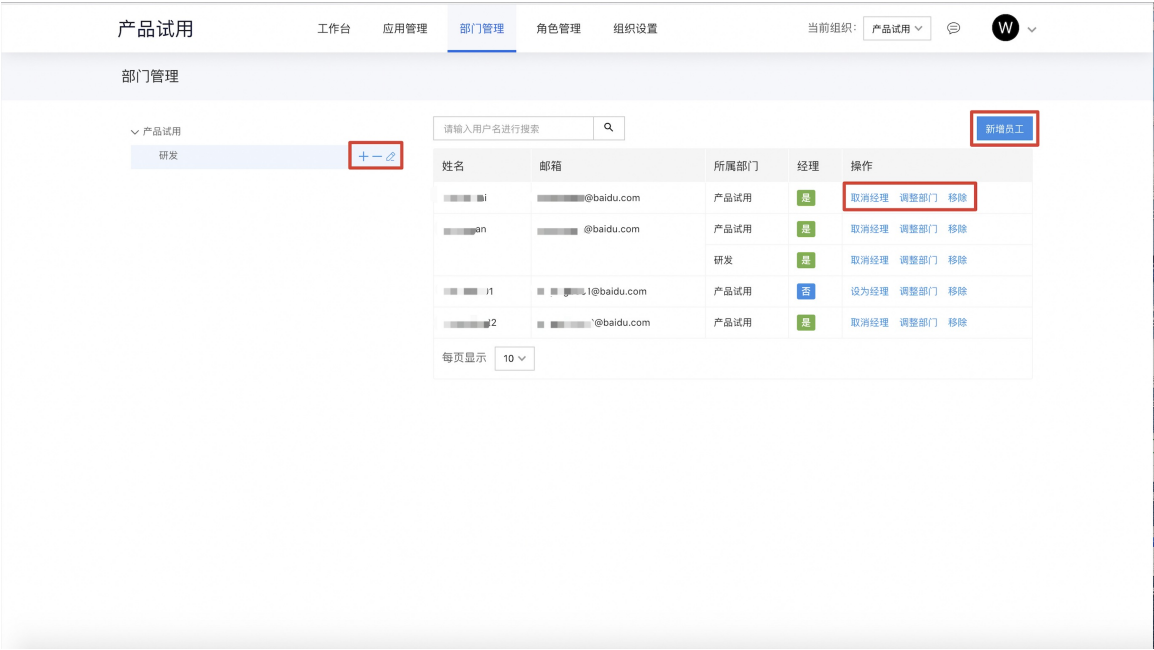
组织架构可以管理组织内的人员和部门信息，协助整体的项目管理工作





部门管理

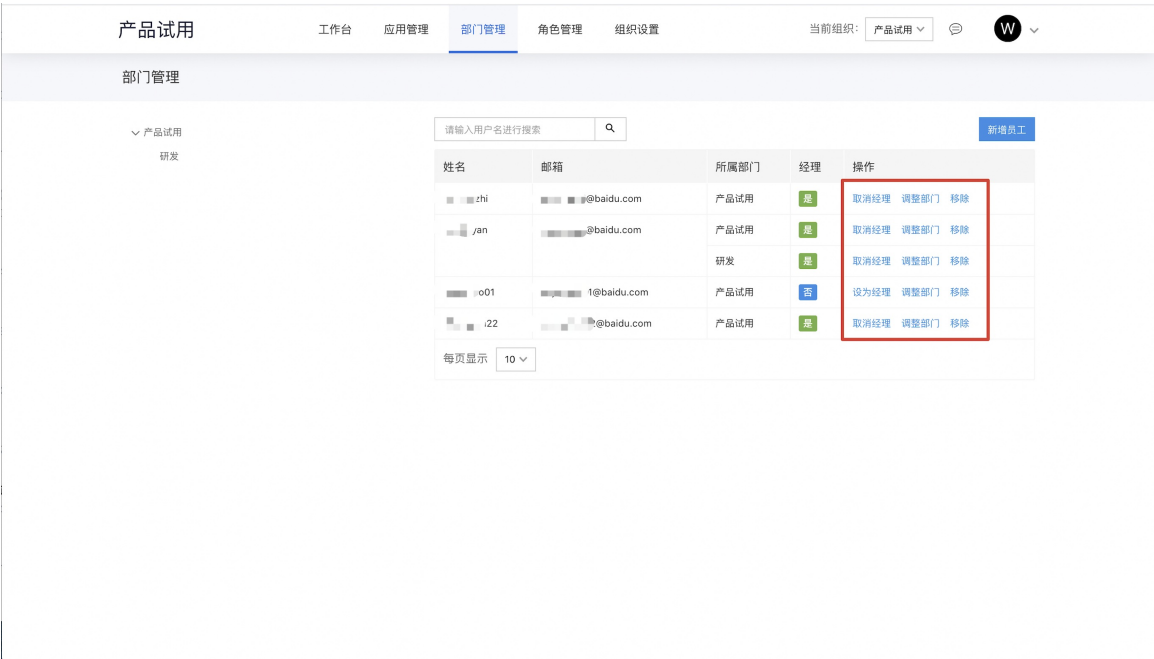
系统默认生成一个以组织名命名的部门，支持名称修改和下级部门增加。



所需支持的架构层级可以在组织设置中配置，默认最多支持 10 级。

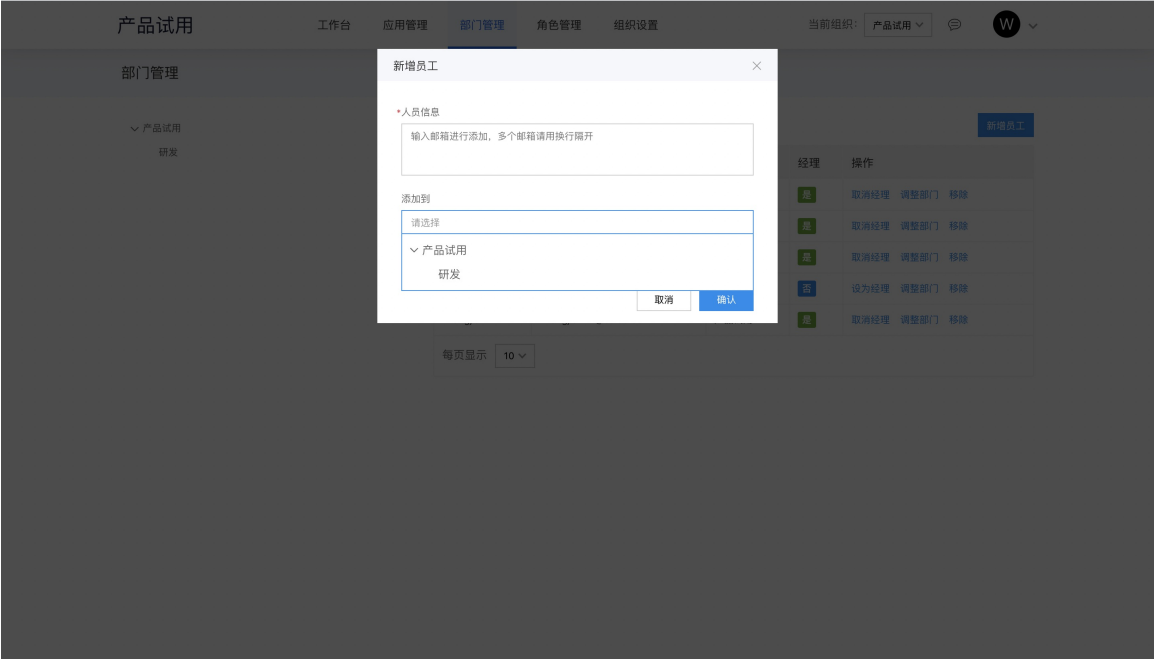
调整部门

组织创建人默认为所有部门的『经理』，『经理』是部门的负责人，具有修改部门员工信息的权限。如果员工的部门信息需要变更，可以通过【调整部门】修改其部门信息。层级编辑后需要刷新页面，员工的所属部门信息才会更新。



新增成员

可以通过邮箱快速添加多个新成员，支持批量增加，一行填写一个邮箱，或使用批量导入功能，并为成员指定部门。



角色管理

在组织的「角色」页面可以对组织中的所有角色进行管理，也可以创建全局角色。

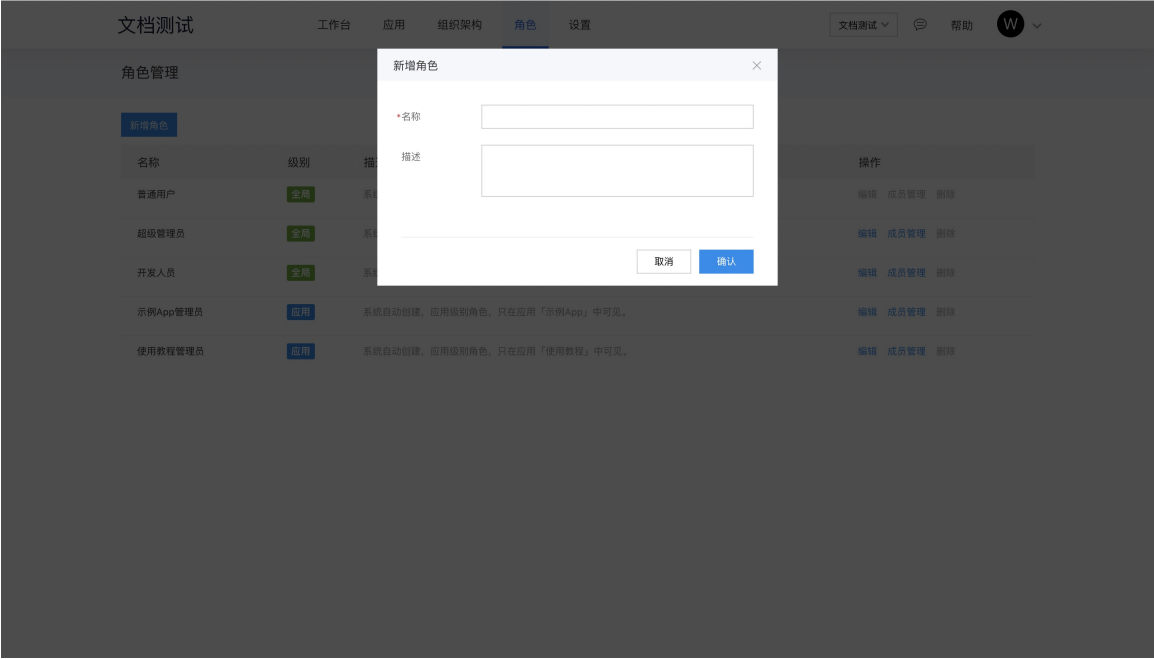
🔗 全局角色

默认有「所有用户」、「组织管理员」和「开发人员」三种全局角色，其中「所有用户」是一种特殊角色，所有组织内的用户都会属于这个角色。



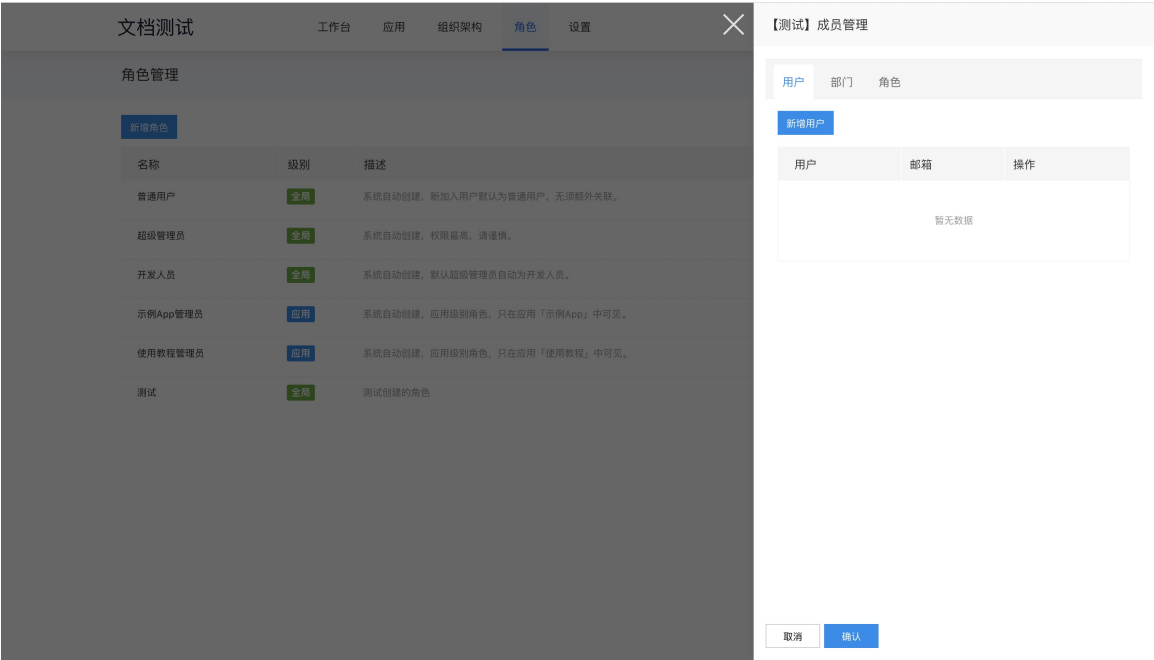
新增角色

全局角色会在所有应用中可见，可以根据需要添加新的全局角色。

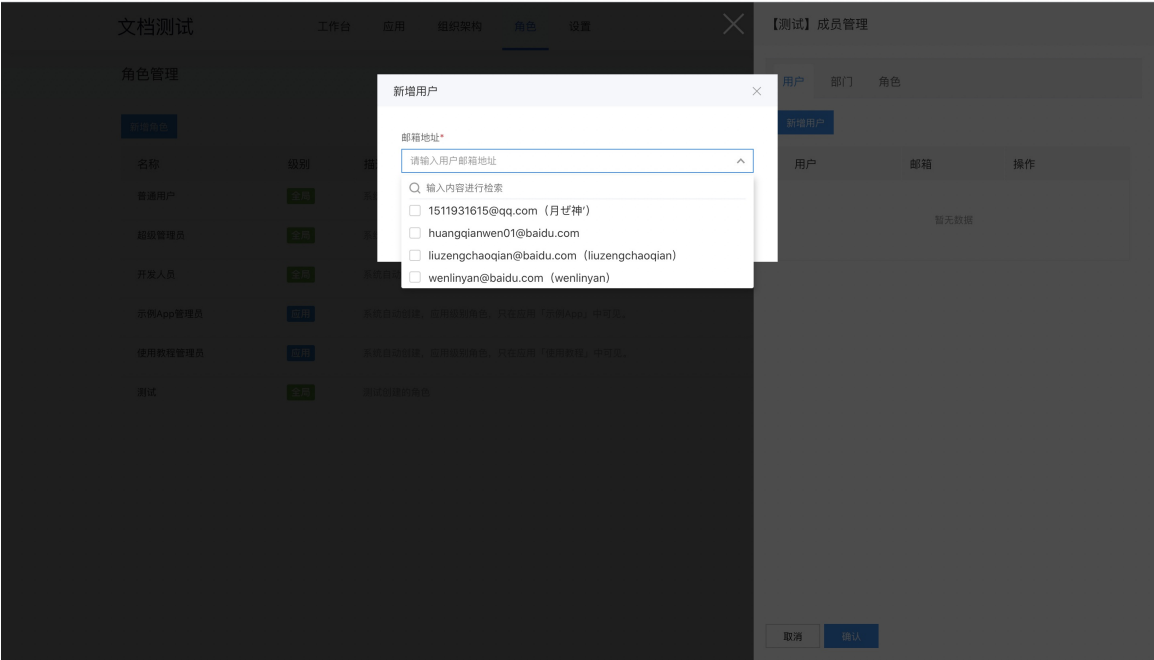


成员管理

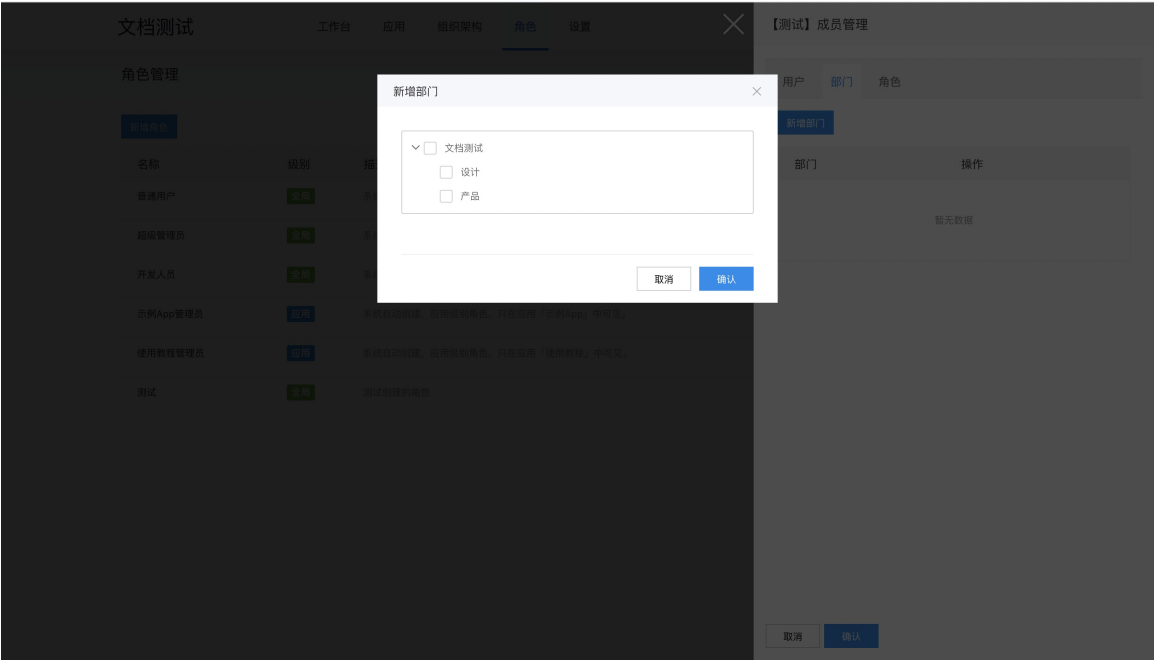
可以为角色新增成员、部门或关联其他角色，但新增和关联的对象只能是系统已有的人员、部门和角色。



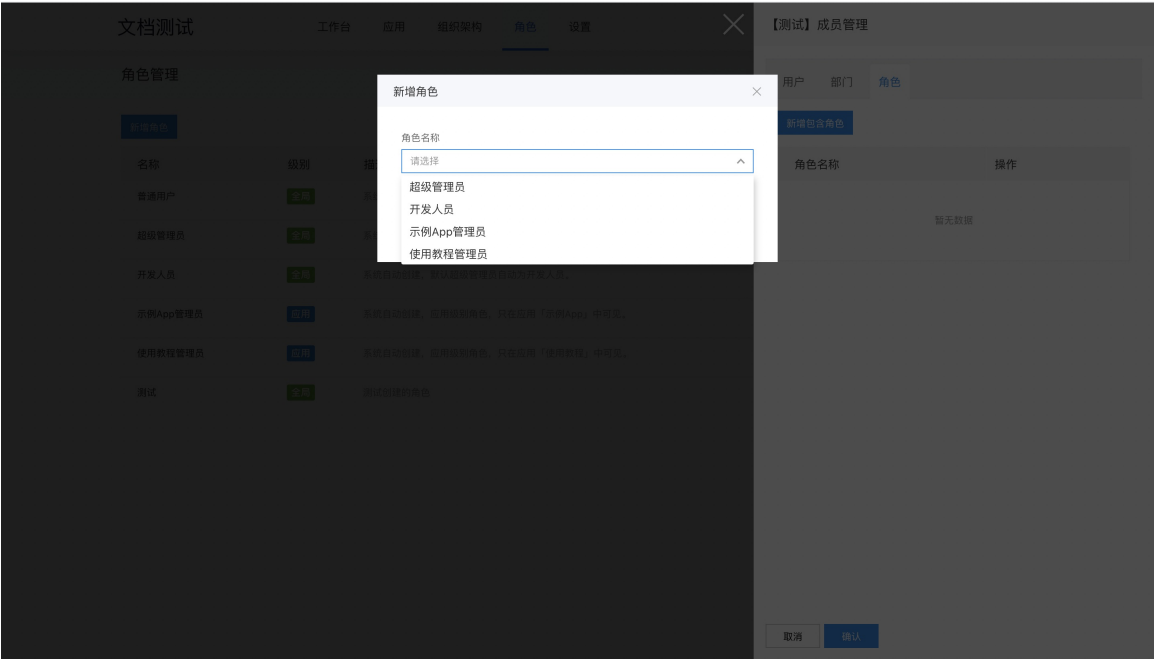
- 关联成员



- 关联部门



• 关联角色



角色本身不具备权限，作为标签，关联成员。在权限管理过程中，可以通过将权限和角色相关联来快速实现对成员的权限管理。

## 基本信息管理

### 🔗 基本信息管理

**设置**可以对组织基本信息和默认设置进行修改和维护，包括基本配置、部门设置、应用权限设置、部门权限设置、角色权限设置。

#### 基本信息

组织设置默认展示在创建组织时填写的信息，设置中可以添加组织标语并修改已有设置。

产品试用

工作台应用管理部门管理角色管理组织设置

当前组织：产品试用

W

基本配置

\*组织名称

产品试用

Logo

+

简称

标语

\*短名字

981bb

联系方式

13879475395

部门设置

部门设置可设置组织最高层级数，默认为 10，也可设置经理创建子部门的权限，默认为开。

部门设置

部门最高层级

10

经理可创建子部门 ☒

应用权限设置

配置应用操作权限和访问权限。

应用默认权限

\*访问开发环境

创建者

组织管理员

所有用户

开发人员

限制谁访问应用开发环境。这里只是默认设置，只会对后续创建的应用生效。

\*可写

创建者

组织管理员

所有用户

开发人员

限制谁能编辑应用，这里只是默认设置，只会对后续创建的应用生效。

\*可删

创建者

组织管理员

所有用户

开发人员

限制谁能删除应用，这里只是默认设置，只会对后续创建的应用生效。

\*可发布

创建者

组织管理员

所有用户

开发人员

限制谁能发布应用，这里只是默认设置，只会对后续创建的应用生效。

部门权限设置

配置部门操作权限，默认为超级管理员具有操作权限。

部门默认权限

\*可写

创建者

组织管理员

所有用户

开发人员

限制谁能编辑部门，这里只是默认设置，只会对后续创建的部门生效。

\*可删

创建者

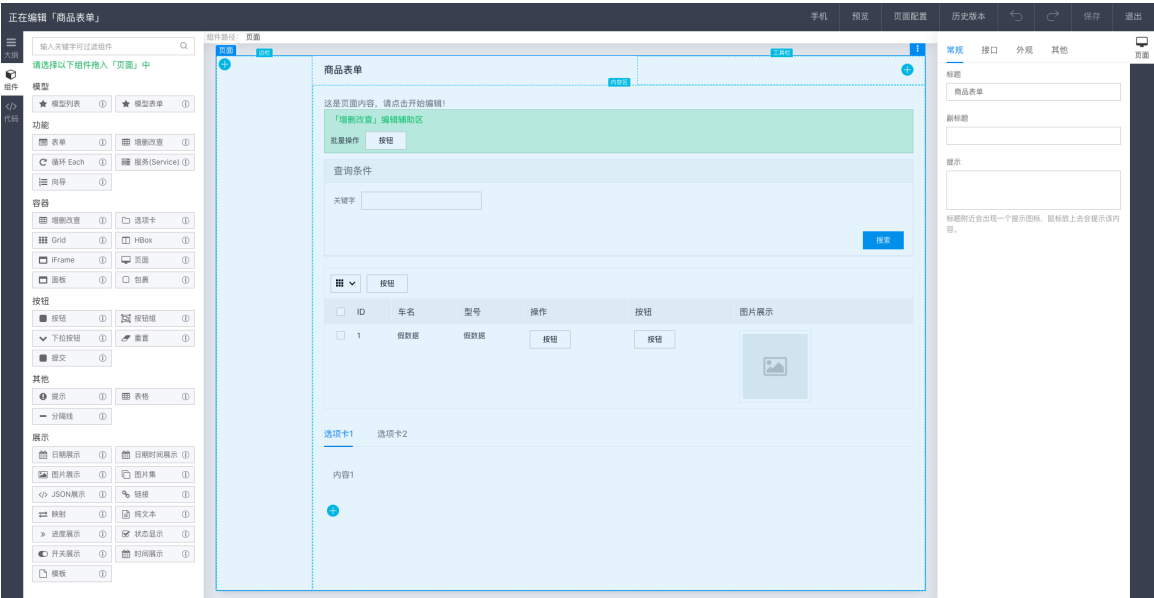
组织管理员

所有用户

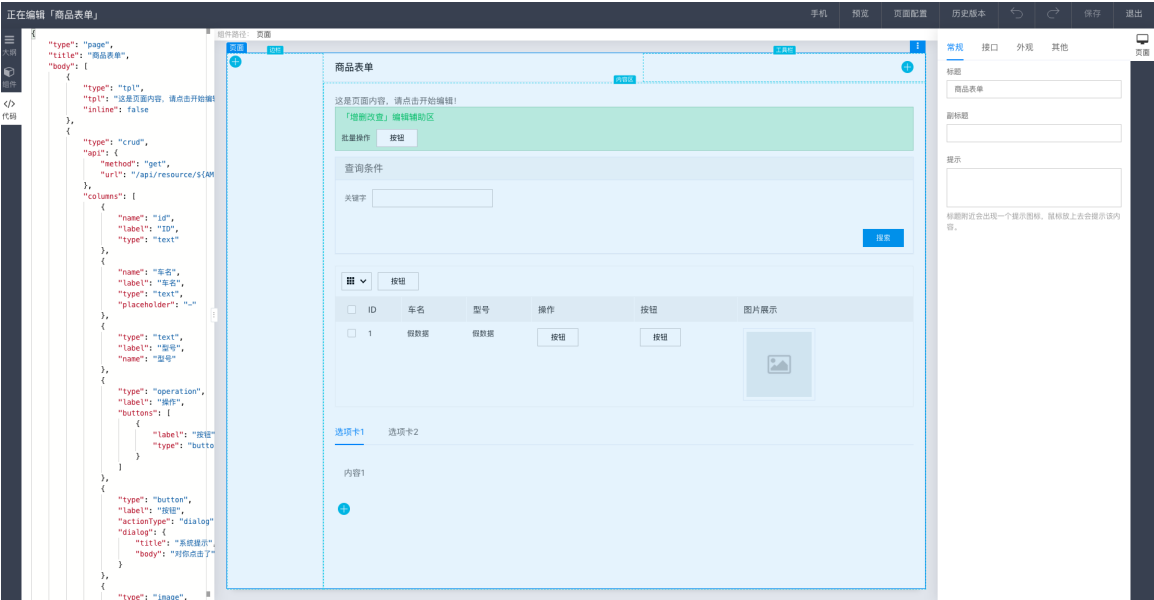
开发人员

限制谁能删除部门，这里只是默认设置，只会对后续创建的部门生效。





同时平台支持页面代码编辑，方便快速通过 json 配置实现页面设计。



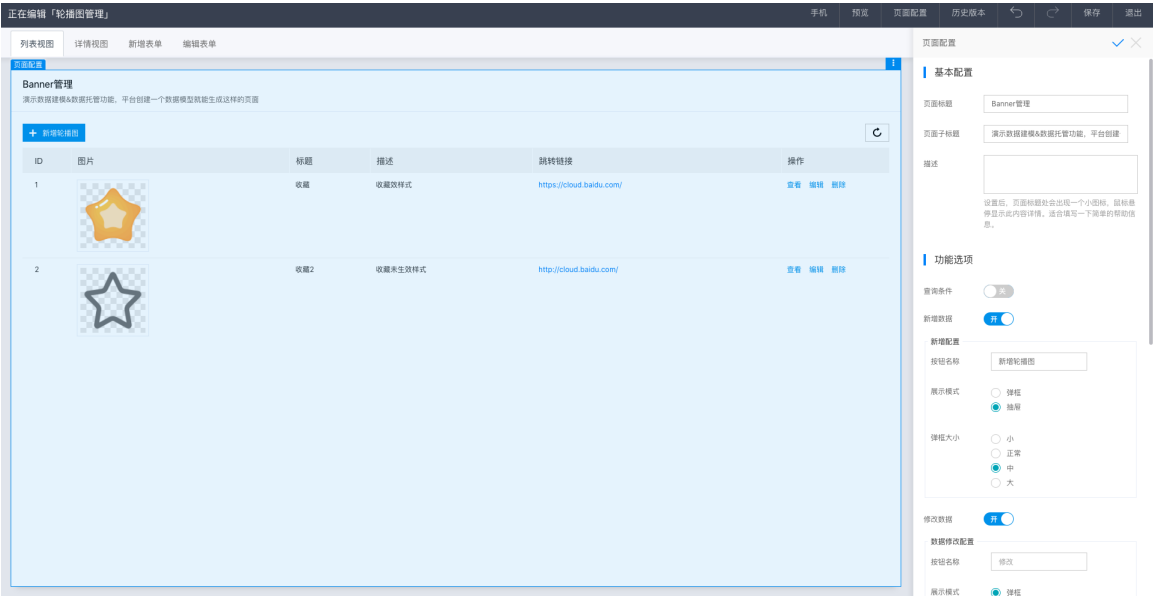
模型页面

模型页面由模型一键生成。根据类型的不同，可编辑的视图不同，通过右侧的页面配置面板，可以实现各视图的详细配置。

列表视图

可配置列表视图的显示字段、字段位置、页面功能包括：查询条件、新增数据功能、修改数据功能、删除数据功能、查看详情功能及操作效果，以及批量操作、数据导入导出功能。

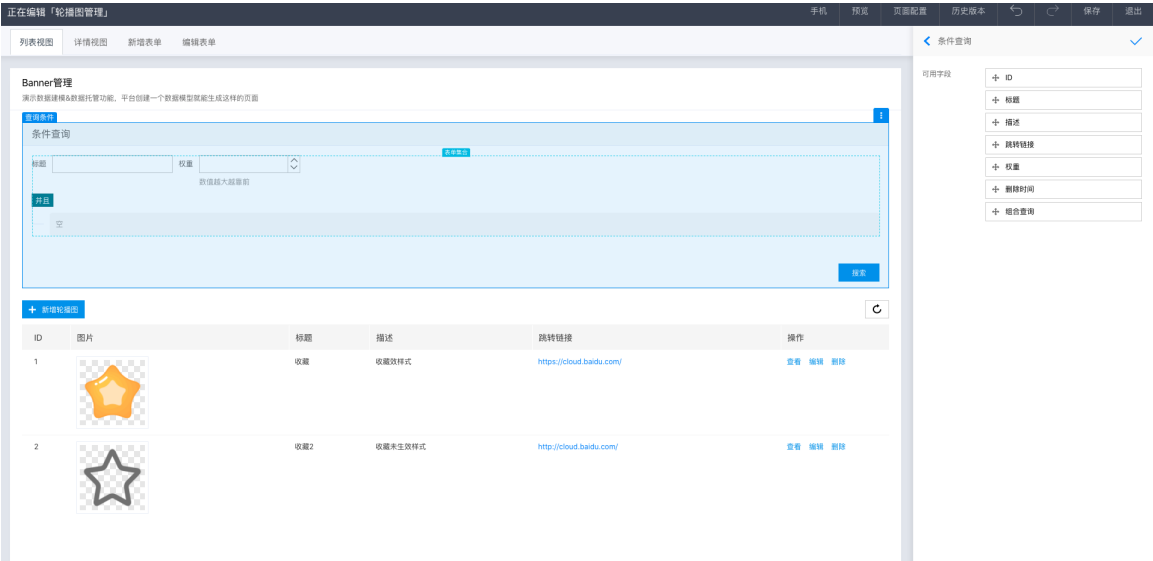




以新增数据功能为例，支持的配置项有：按钮名称、展现模式、弹窗大小。



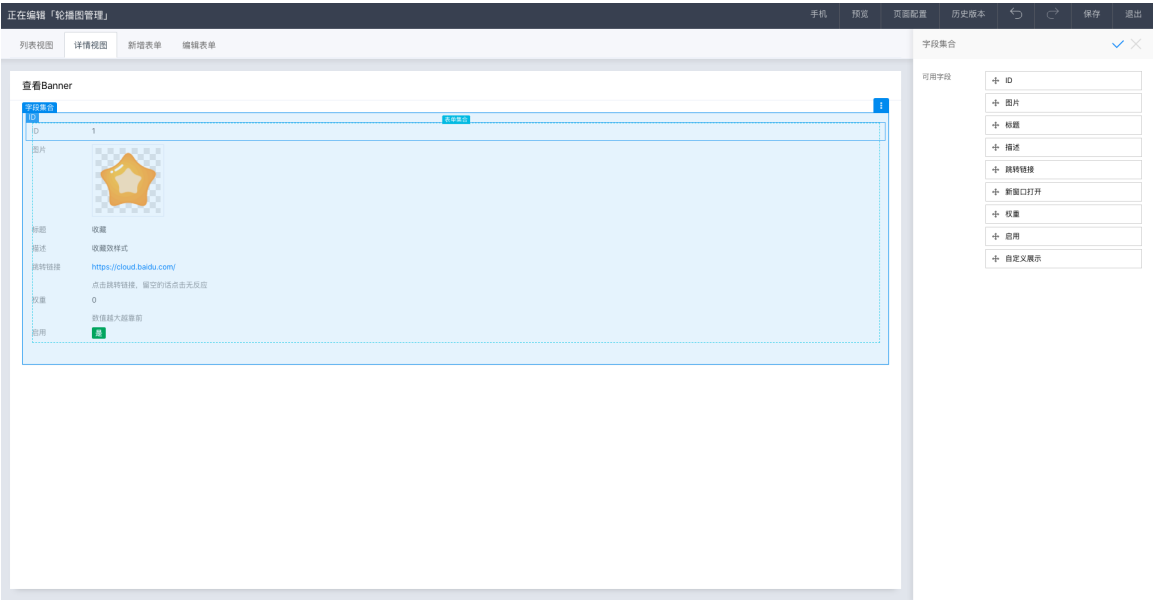
开启查询条件功能可在列表页新增条件查询模块，点击左侧条件查询区域，右侧展开条件配置面板。将字段拖拽到条件查询模块即可完成查询条件配置。同时平台支持组合条件查询，可实现两个以上条件组合结果进行查询。



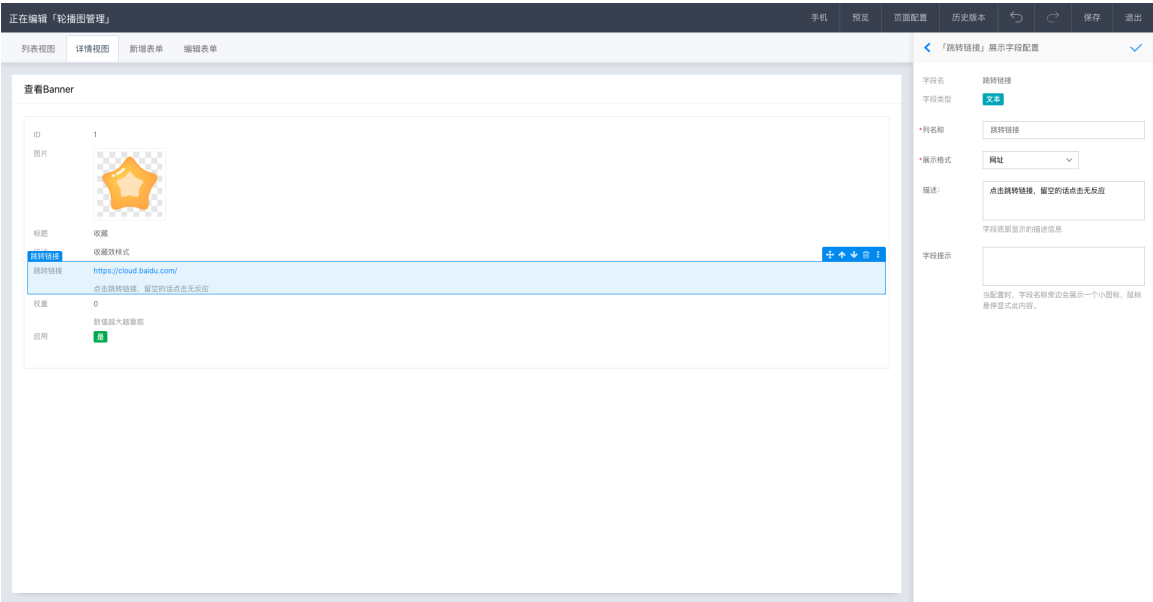
● 详情视图

因详情视图主要用于查看详情，因此配置项相对较少，页面默认包含数据模型其中的全部字段。首先是字段项的配置，可实现详情页面展示内容中包含哪些字段的配置。在某些场景下，一条数据包含较多的数据，但是查看时只需要关注其中的部分重要

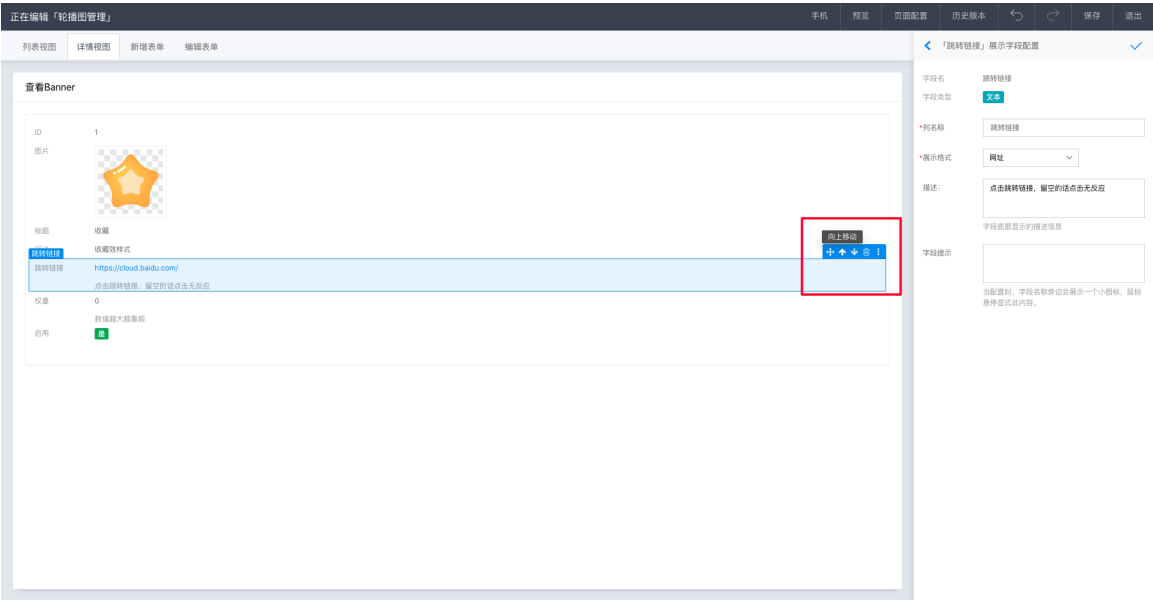
信息，就可以通过在详情视图中删除部分字段来实现。



详情视图也支持字段级的信息配置。不同的字段类型支持不同的配置项。



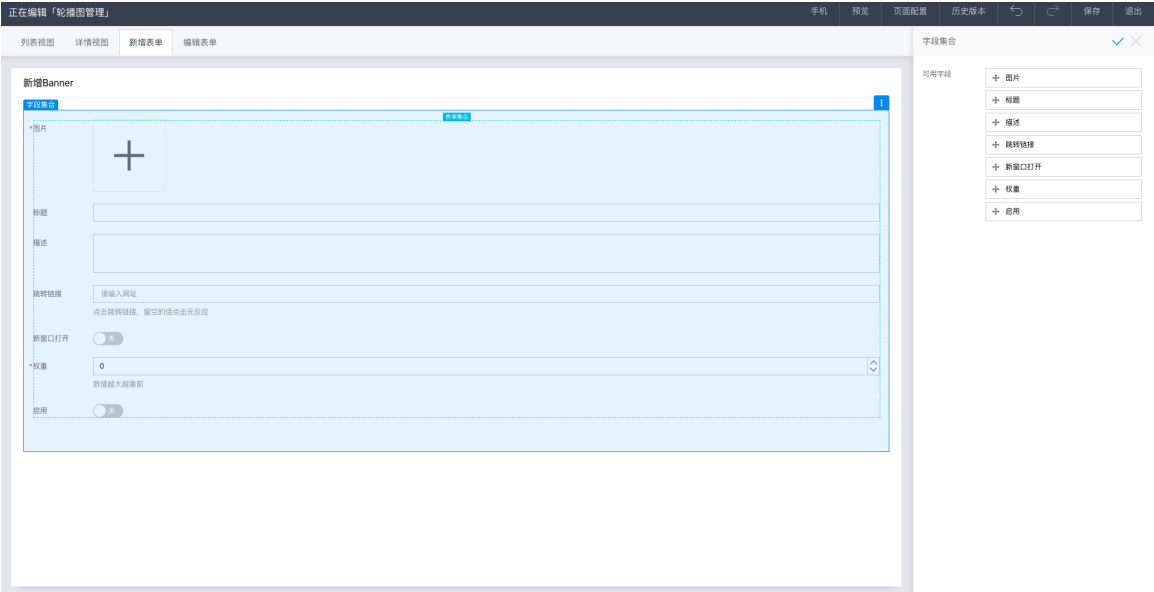
此外，也可以按需设置表单中字段的位置关系。



视图编辑可进行字段信息配置、可进行字段位置调整是所有视图都有的能力

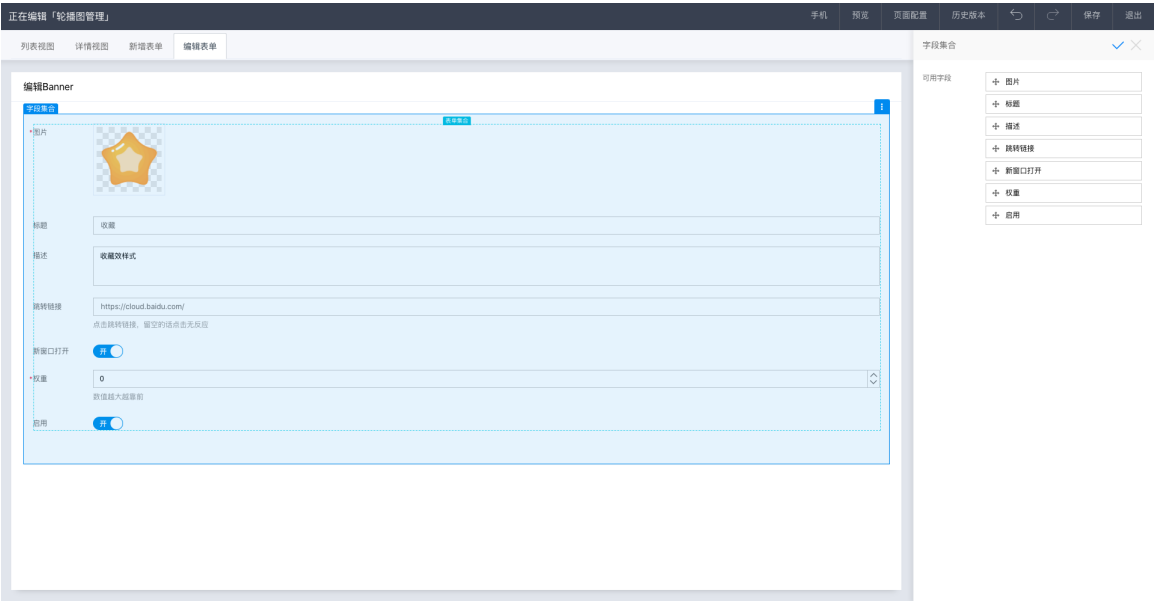
新增视图

新增视图的主要功能是信息收集和提交，因此支持配置项也相对较少。*列表页面的新增视图和模型页面的新增表单视图类似*，只是在实际应用环境中展现效果不一样。



编辑视图

编辑视图和新增视图类似，也是个表单页，也支持字段集合设置和字段信息配置。



需要注意的是，在进行任何页面修改操作后都需要点击右侧信息配置面板的保存图标，修改的配置项才会被暂时保存。

「跳转链接」展示字段配置

✓

✕

字段名

跳转链接

字段类型

文本

\*列名称

跳转链接

\*展示格式

网址

描述:

点击跳转链接, 留空的话点击无反应

字段底部显示的描述信息

字段提示

当配置时, 字段名称旁边会展示一个小图标, 鼠标悬停显示此内容。

最后, 编辑完成后要离开页面时, 请先点击页面右上方的【保存】按钮, 才能完成页面配置信息的保存。

预览

页面配置

历史版本

↶

↷

保存

退出

「跳转链接」展示字段配置

✓

✕

字段名

跳转链接

富文本页面

富文本页面类似一个富文本编辑器, 可以编辑任意需要的内容, 适合页面内容丰富的文本展示型页面。

正在编辑「应用首页」

手机

预览

页面配置

历史版本

↶

↷

保存

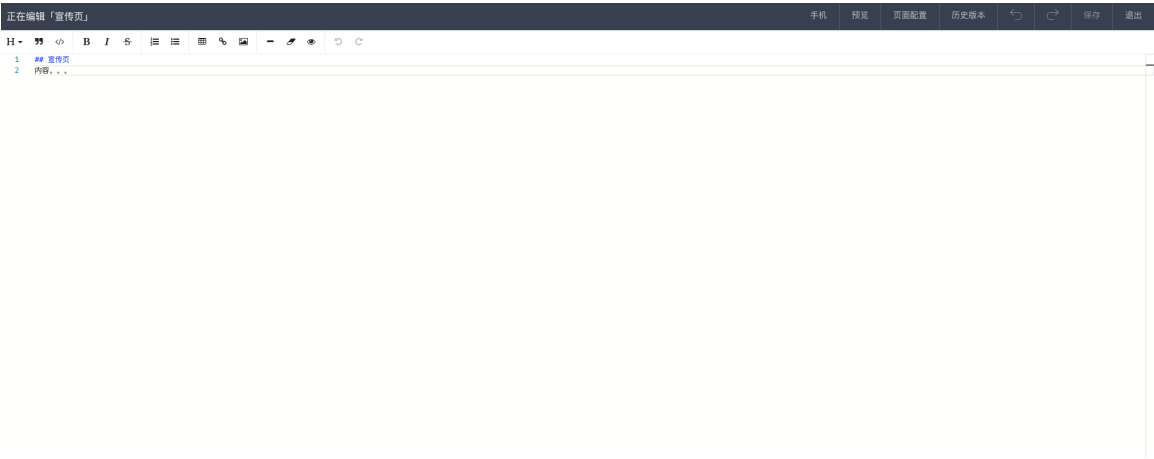
退出

应用首页

内容...

Markdown 页面

Markdown 页面类似一个 markdown 编辑器, 可以编辑任意需要的内容, 适合页面内容较多的文本型页面。



🔗 文件夹页面

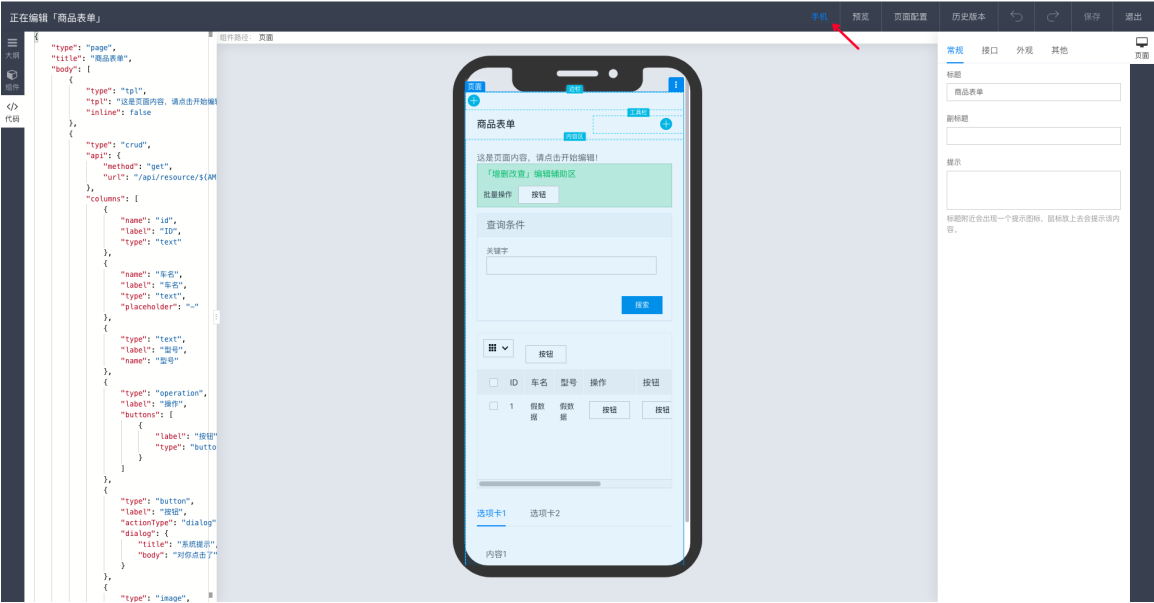
文件夹页面没有实际页面内容，因此不支持页面内容编辑。

🔗 移动端样式

目前爱速搭平台搭建的页面自动适配移动端，模型页面自动适配，组件页面同时支持移动端上做样式修改。

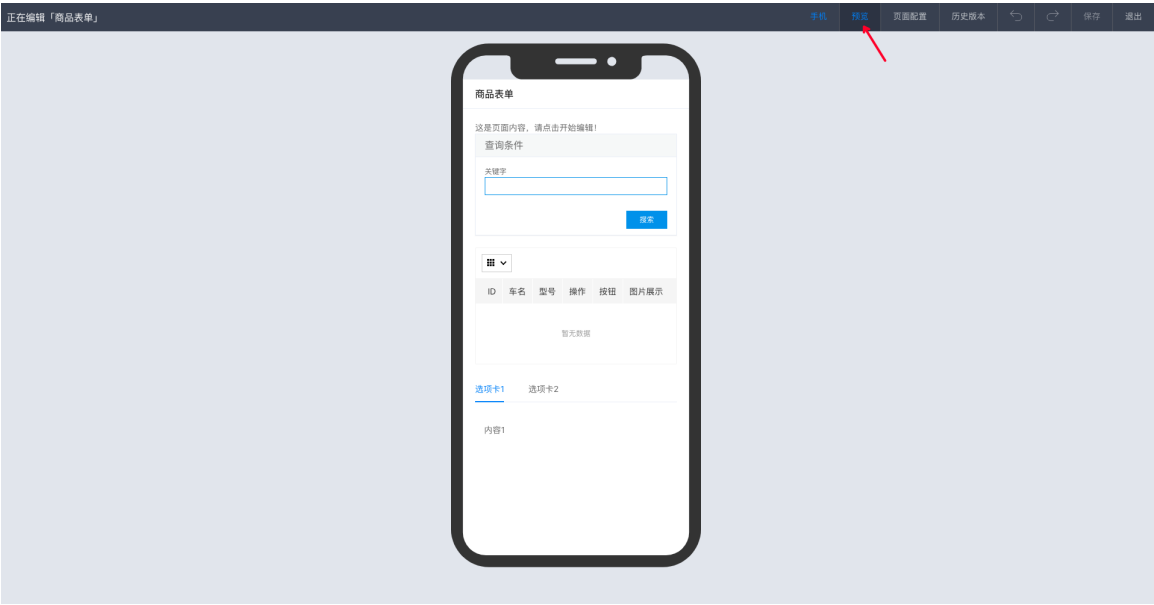
- 切换方式

点击编辑器环境顶栏的【手机】，切换到手机编辑模式。



- 预览方式

手机编辑模式下，点击顶栏【预览】进行效果预览。

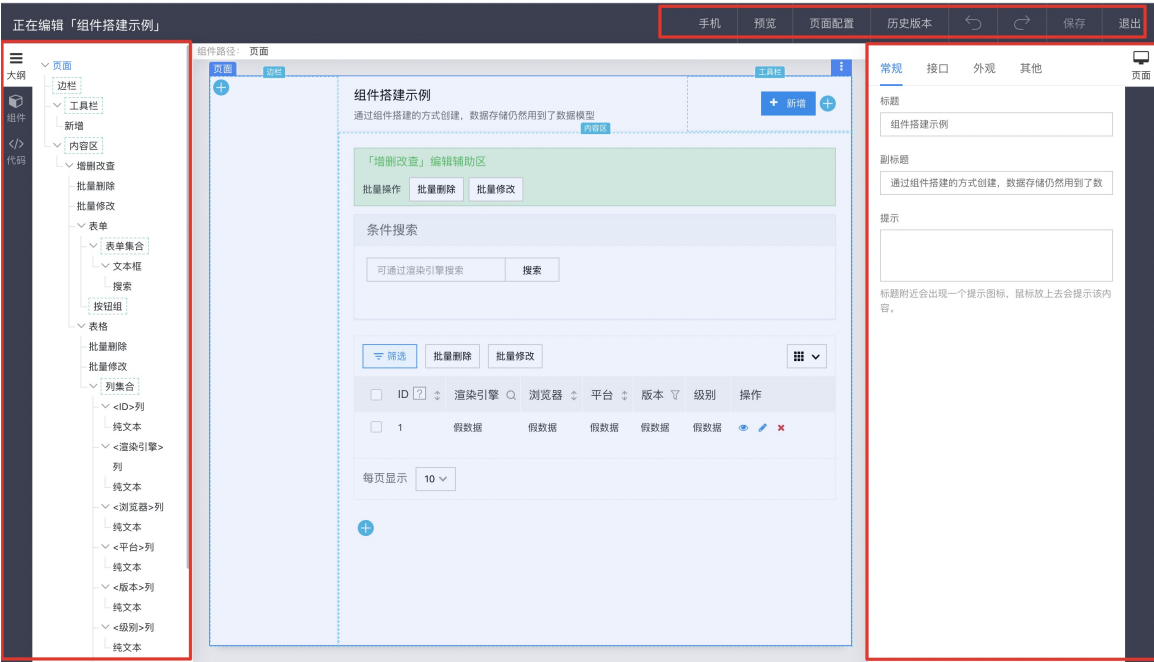


## 普通页面设计

### 普通页面说明

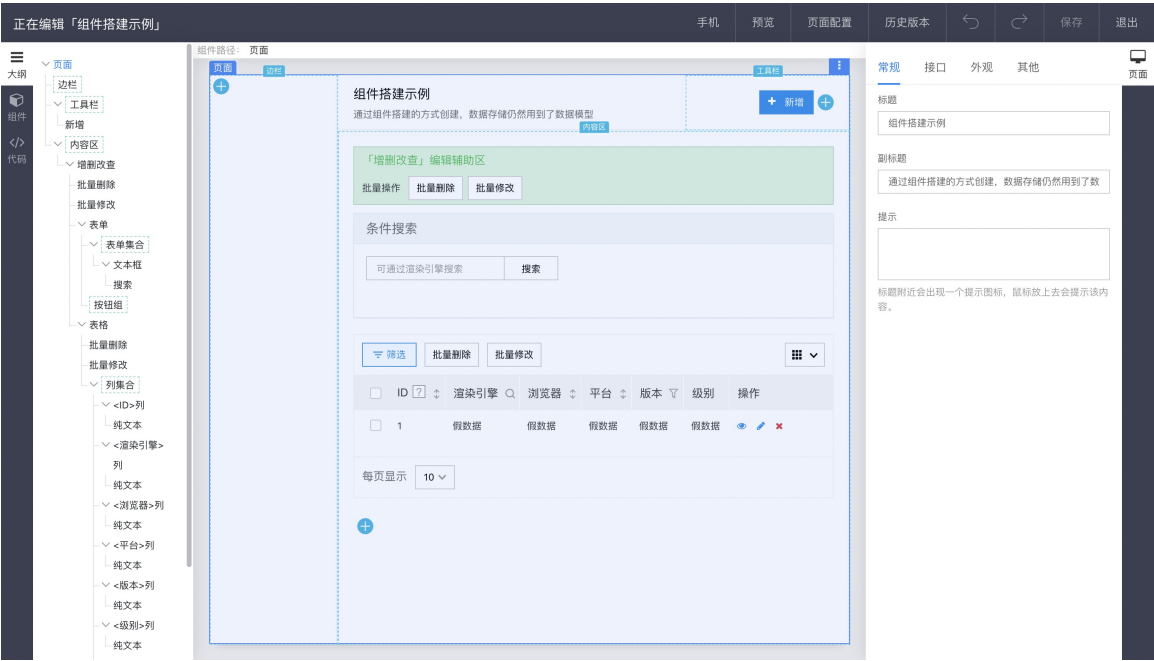
普通页面使用爱速搭开源的前端渲染器 [amis](https://baidu.gitee.io/amis/docs/index) 实现，因此具有 amis 所有的能力，相关文档：<https://baidu.gitee.io/amis/docs/index>

组件搭建页面由编辑区、全局工具、组件面板和配置面板和四部分组成。



### 编辑区

页面蓝色区域，显示当前状态页面内容，分为边栏、工具栏和内容区。编辑区可以通过鼠标选中、大纲树节点选择或右键功能选择想要选中的区域并进行编辑。



全局工具

全局工具支持手机模式切换，支持预览、撤销、保存、推出等基本操作，也可以通过【页面配置】修改页面配置信息。【历史版本】支持查看编辑保存日志，并恢复已保存版本实现页面配置回滚。



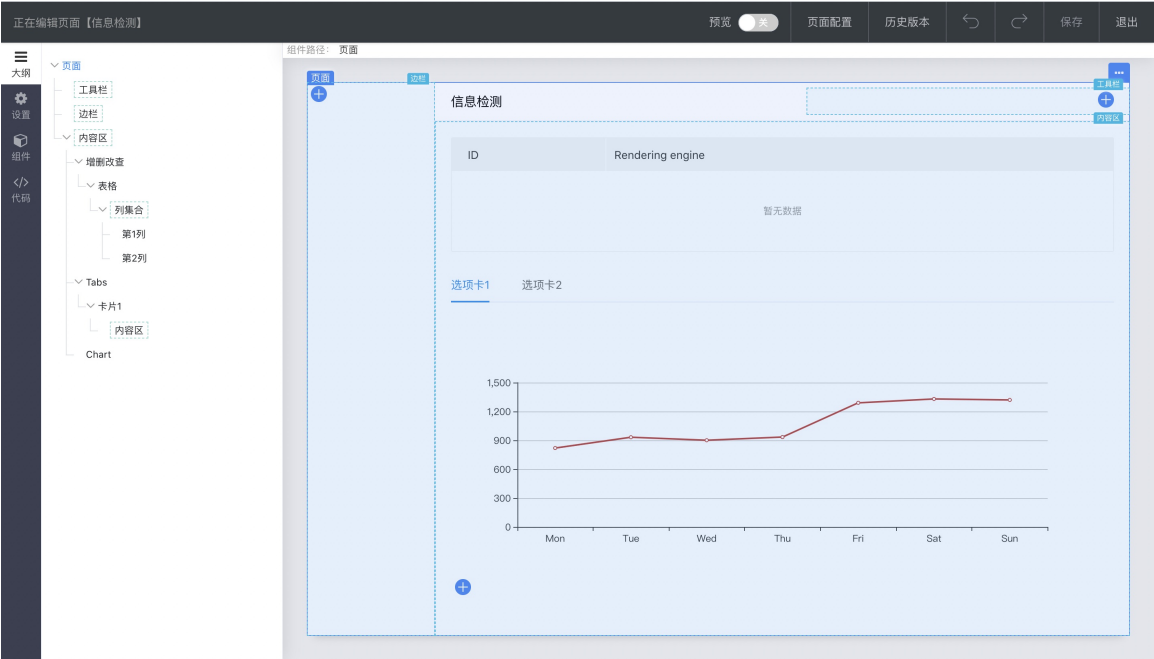
组件面板

组件面板分为大纲、组件和代码三个模块。



大纲

可以通过大纲，选中不同层级的组件，从而进行组件属性的设置



组件

展示组件的组件面板，可以从中选择要添加的组件拖拽到指定的区域中。



大纲

组件

代码

输入关键字可过滤组件

请选择以下组件拖入「页面」中

模型

★ 模型列表

★ 模型表单

功能

表单

增删改查

循环 Each

服务(Service)

向导

容器

增删改查

选项卡

Grid

HBox

iFrame

页面

面板

包裹

按钮

按钮

按钮组

下拉按钮

重置

提交

其他

提示

表格

分隔线



通过编辑区的添加图标也可以唤起组件列表，这里会有对组件的简要介绍，并且包含平台的所有组件。部分新增的组件也会优先在这里展示。

选中组件插入到 页面 > 内容区

输入关键字可过滤组件

Q

全部

模型

功能

容器

按钮

其他

展示

模型列表

模型列表

选定模型后，可根据模型信息生成列表数据。

模型表单

模型表单

可以用来给目标模型做新增、查询或者更新操作。

表单

文本

提交

表单

可用于新建、编辑或者展示数据，配置初始化接口可从远端加载数据，配置提交接口可将数据发送远端。另外也可以将数据提交给其他组件，与其他组件通信。[详情](#)

| A | B | 操作                                                                                                                                                                      |
|---|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | 2 |   |
| 3 | 4 |   |
| 5 | 6 |   |

增删改查

用来实现对数据的增删改查，支持三种模式展示：table、cards和list. 负责数据的拉取，分页，单条操作，批量操作，排序，快速编辑等等功能。集成查询条件。[详情](#)

按钮

按钮

用来展示一个按钮，你可以配置不同的展示样式，配置不同的点击行为。[详情](#)

选项卡1

选项卡2

内容1

选项卡

选项卡，可以将内容分组用选项卡的形式展示，降低用户使用成本。[详情](#)

组件

插入

取消

如果需要的组件在组件面板中未找到，可以尝试在组件列表中进行检索，如仍未找到可以使用自定义组件功能（近期即将推出）或通过文档留言或工单进行反馈。

## 代码

也可以通过代码模式，直接对 amis 配置代码进行编辑，具体 amis 配置文档可看：<https://baidu.gitee.io/amis/docs/index>

## 正在编辑「组件搭建示例」



大纲



组件



代码

```
{
  "type": "page",
  "title": "组件搭建示例",
  "body": [
    {
      "type": "crud",
      "draggable": false,
      "api": "/api/resource/${AMIS_C",
      "filter": {
        "title": "条件搜索",
        "submitText": "",
        "controls": [
          {
            "type": "text",
            "name": "engine",
            "placeholder": "可",
            "addOn": {
              "label": "搜索",
              "type": "submi"
            }
          }
        ]
      },
      "bulkActions": [
        {
          "label": "批量删除",
          "actionType": "ajax",
          "api": "delete:/api/re",
          "confirmText": "确定要批",
          "size": "sm"
        },
        {
          "label": "批量修改",
          "actionType": "dialog"
```



## 配置面板

主要实现组件属性的可视化配置，配置信息会随着组件的不同而变化。

以增删该查组件为例，需要配置的项包括：

### • 增删改查

- 常规：基础功能配置，如查询条件、批量操作、单条操作、同步地址栏、默认参数、保留条目选择、指定主键等
- 接口：可配置数据拉取接口、快速保存接口、快速保存单条接口、默认消息提示等
- 外观：可配置内容展现方式、顶部工具栏配置、底部工具栏配置、快捷工具配置和 CSS 设置等
- 其他：页面基础设置配置，如数据源、页面信息字段等

### • 表格

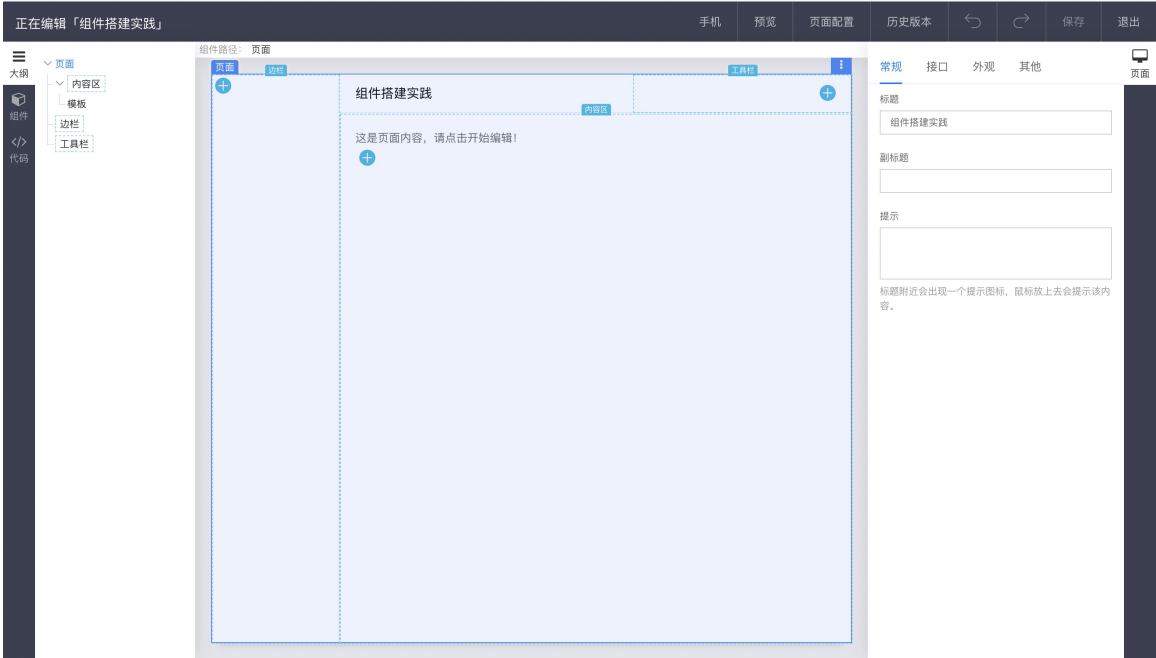
- 常规：标题和表格列设置
- 外观：表格展现样式
- 其他：数据显隐配置

## 🔗 页面搭建流程

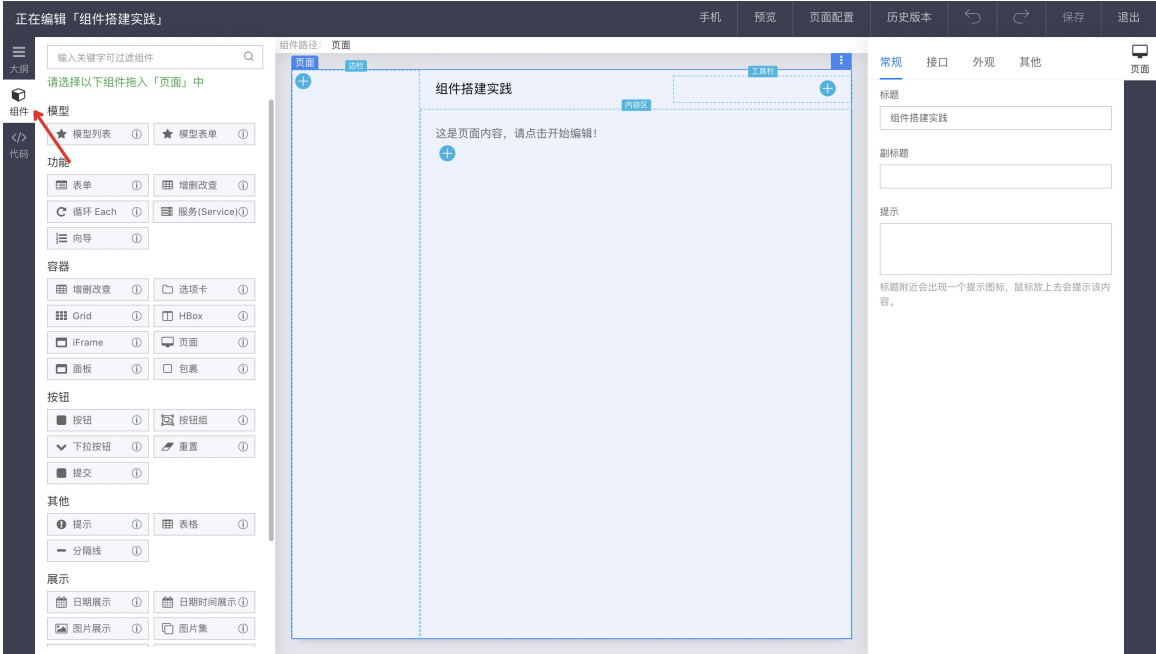
通过组件搭建完成的页面主要包括两部分：添加组件和配置信息。

### 添加组价

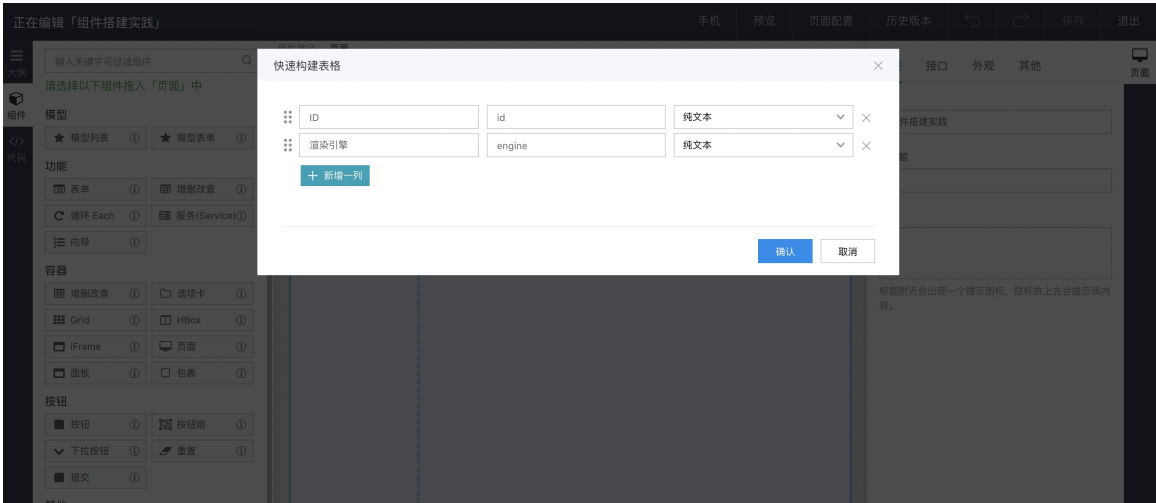
1. 新建一个普通页面，进入页面编辑环境。



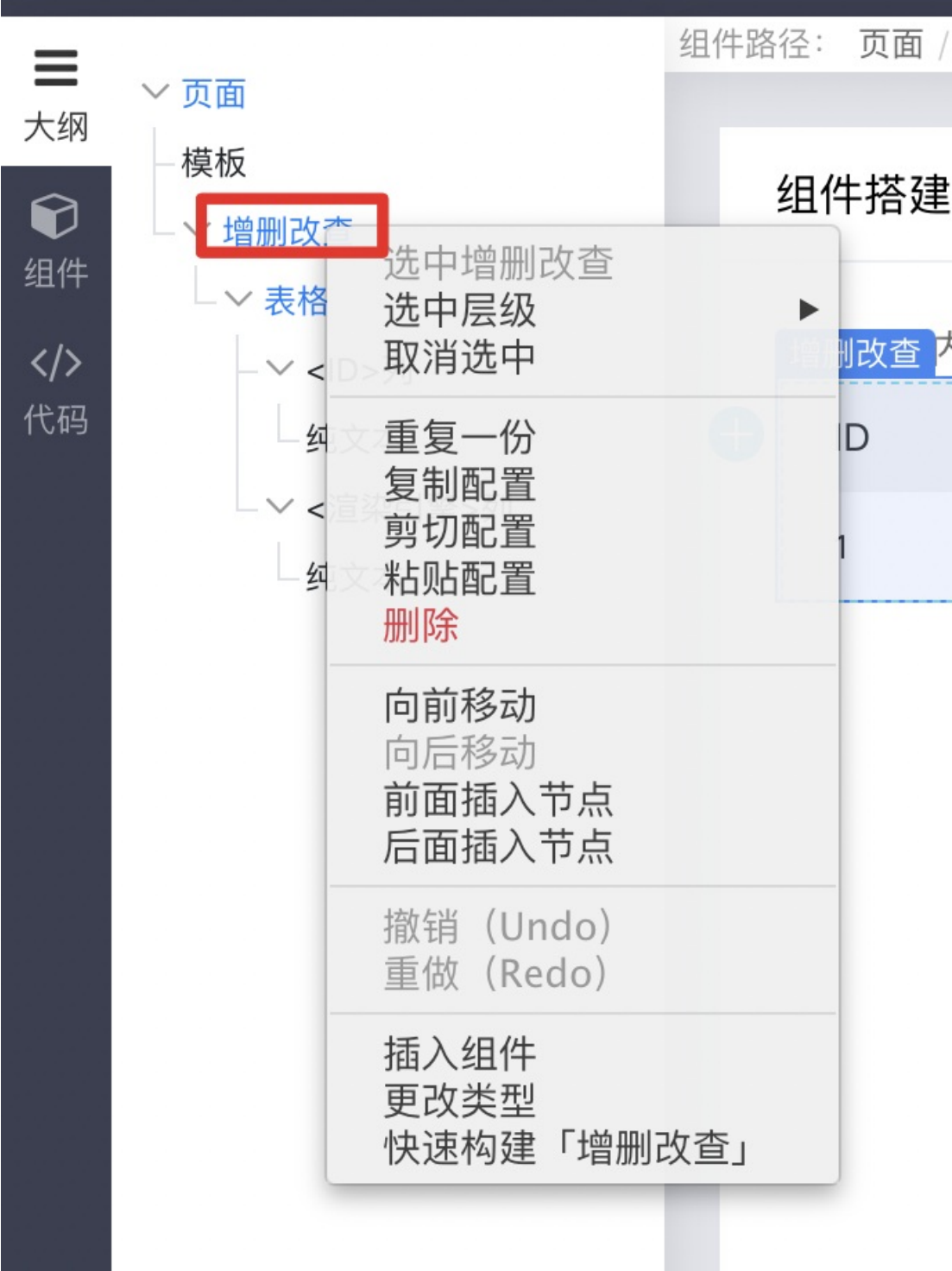
2、点击左侧【组件】打开组件面板。



3. 选择一个增删改查组件，并拖拽到页面内容区域。在弹出的页面中配置数据信息：ID、渲染引擎、浏览器、平台、版本、级别。

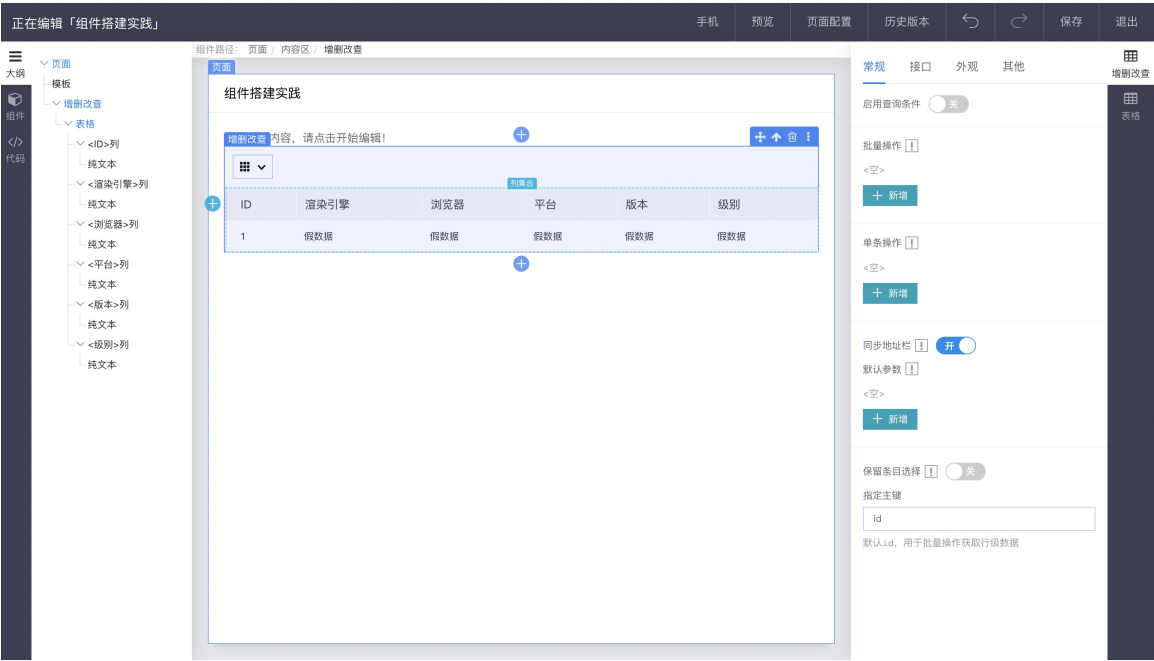


如果初始状态下未完成字段配置，可以通过点击大纲树中的组件【增删改查】然后右键打开快捷操作。

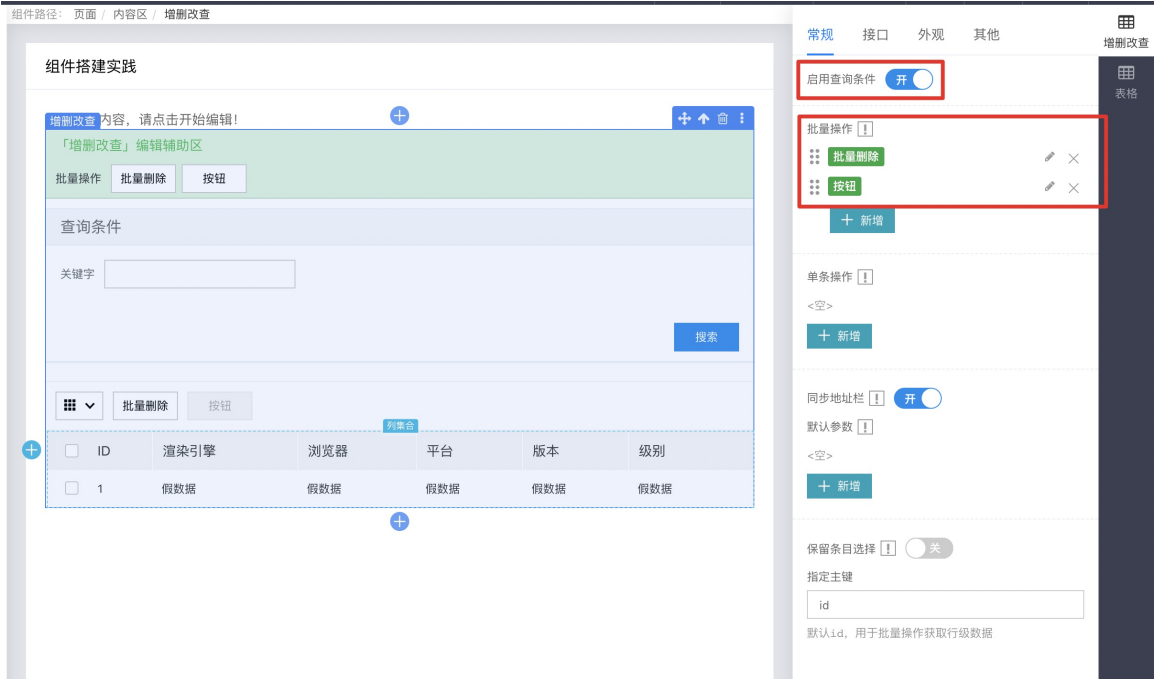


说明：右键快捷操作在编辑器环境下都可以使用，选中的层级和组件不同，支持的操作项也有不同。

4. 配置字段后可以看到如下效果。

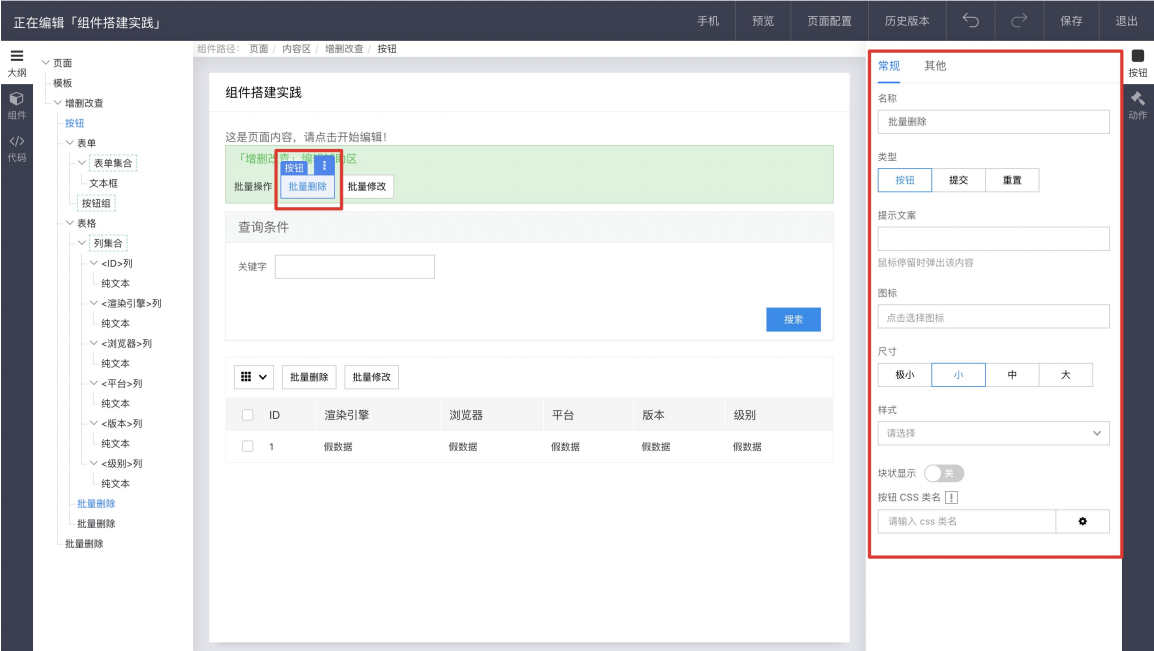


5. 然后为页面增加一些基础功能。打开查询条件，添加两个批量操作按钮。

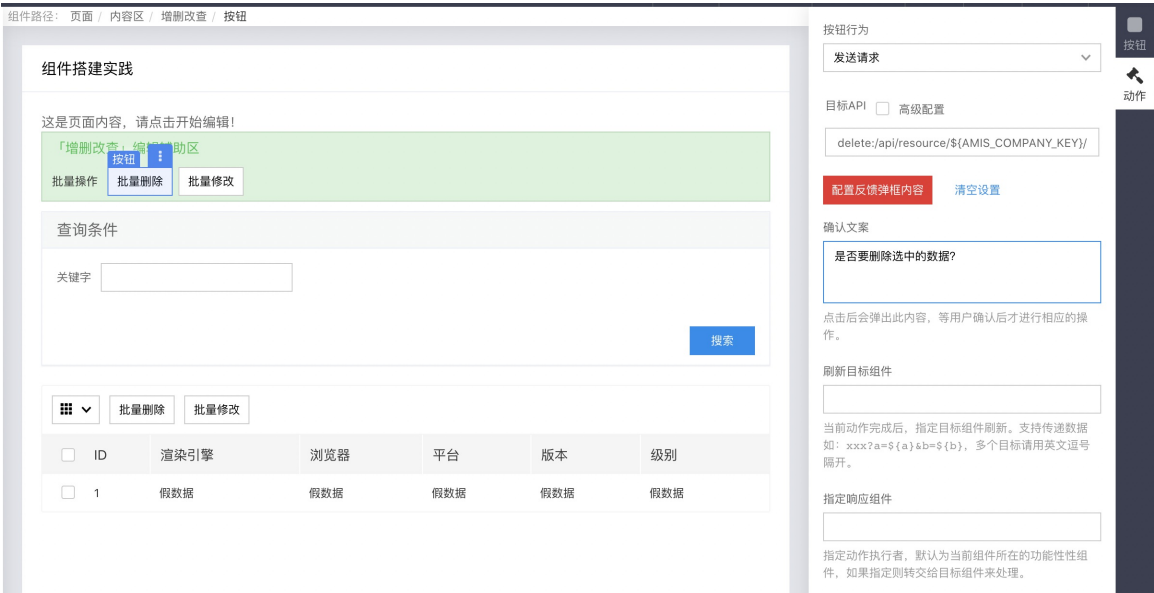


6. 点击左侧编辑区的按钮，在右侧的配置面板中进行操作配置。





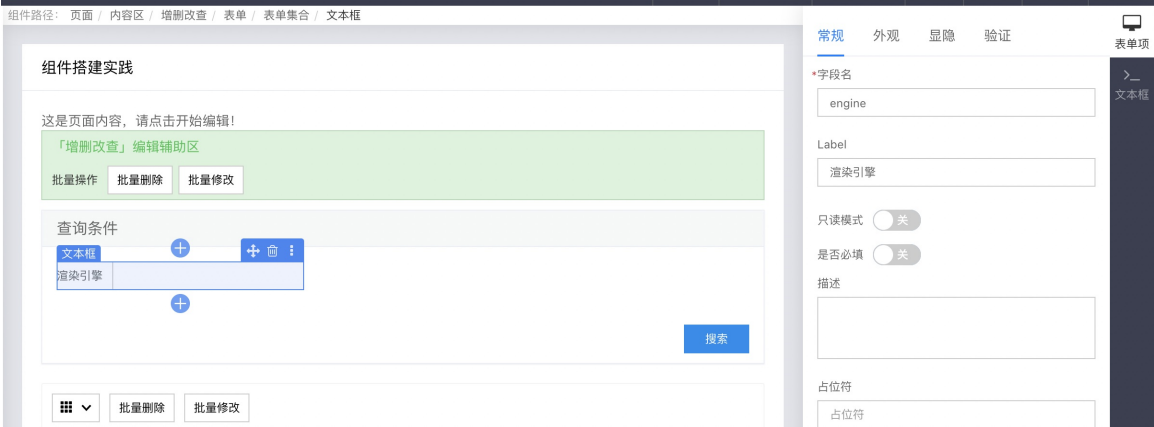
7. 然后配置按钮的动作为“发送请求”，并且填入目标 API，并完善确认文案。类似的方法完成批量修改按钮的配置，平台支持 13 种动作类型，能满足主要按钮场景的需求。



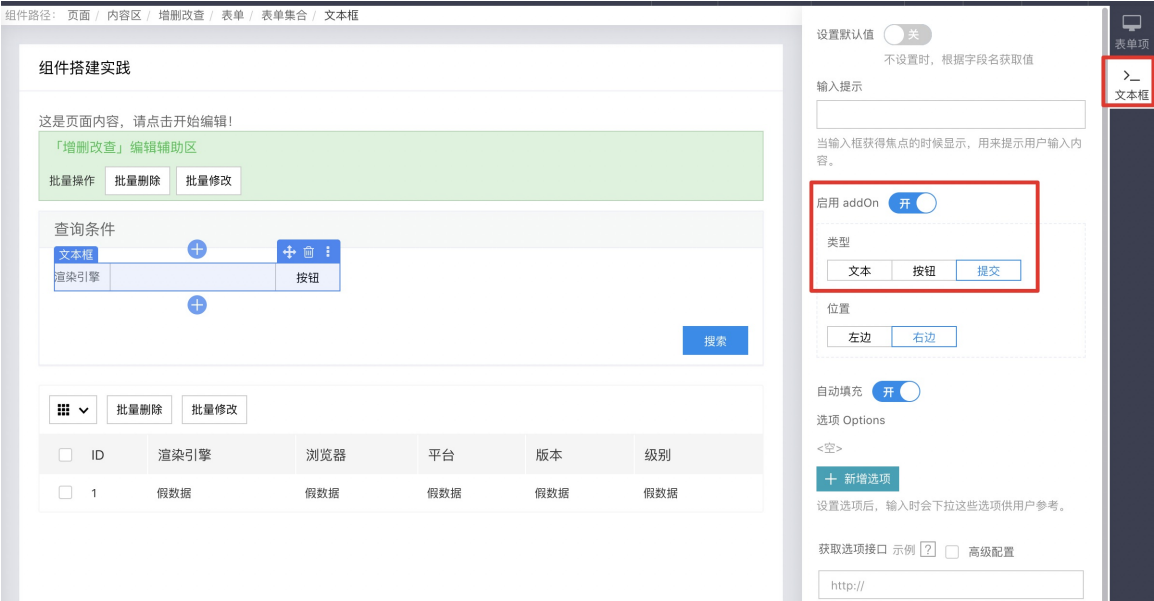
8. 下一步配置查询条件，选中查询表单组件，打开右侧配置面板。



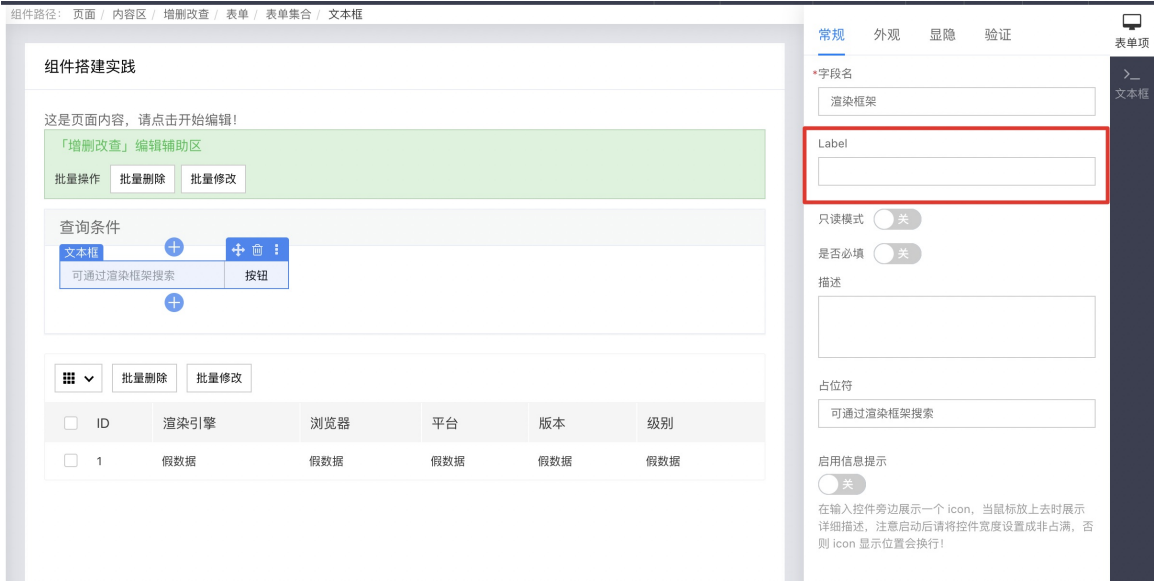
9. 完成字段配置和字段名绑定，配置筛选条件为：渲染引擎，字段名为 engine。



10. 在文本框配置中，打开 addOn，按钮类型改为【提交】。



11. 删除 label 就可实现文本框名称显示。



12. 然后保存，这个页面的配置过程就完成了。退出后可看到效果。



说明：上述示例中配置的接口是没有数据的，因此表中无数据信息。

组件说明

组件是爱速搭平台的核心能力之一，平台支持上百种组件，并实现了根据选中组件的不同进行分组展示，让组件选择和拖拽更加方便。

页面设计器默认展示的可选组件主要分为：

- 模型组件

模型组件是组件化的数据模型，往页面中添加数据模型即完成了数据模型表单在页面中的嵌入，功能上保留了数据模型的主要特性如数据收集、提交、管理等。便捷完成组件的数据管理配置。

目前平台支持了模型列表组件和模型表单组件，分别对应数据模型的列表展示类型和表单提交类型和当前用户信息。



- 功能组件

功能组件主要用于实现一定的页面逻辑功能。如自定义代码，二维码，异步任务等。

功能

|                          |                      |
|--------------------------|----------------------|
| <div>增删改查 (API) ⓘ</div>  |                      |
| <div>表单 (API) ⓘ</div>    | <div>公式 ⓘ</div>      |
| <div>隐藏域 ⓘ</div>         | <div>地理位置选择 ⓘ</div>  |
| <div>UUID ⓘ</div>        | <div>音频 ⓘ</div>      |
| <div>自定义代码 ⓘ</div>       | <div>循环 Each ⓘ</div> |
| <div>导航 ⓘ</div>          | <div>二维码 ⓘ</div>     |
| <div>服务(Service) ⓘ</div> | <div>异步任务 ⓘ</div>    |
| <div>视频 ⓘ</div>          | <div>向导 ⓘ</div>      |

- 容器组件

容器组件主要是在页面中添加一个容器，以进行局部页面更多的配置和设计。

容器

|                            |                   |
|----------------------------|-------------------|
| <div>增删改查 (API) ⓘ</div>    |                   |
| <div>表单项容器 ⓘ</div>         | <div>容器 ⓘ</div>   |
| <div>Flex 布局 ⓘ</div>       | <div>两栏 ⓘ</div>   |
| <div>三栏 ⓘ</div>            | <div>HBox ⓘ</div> |
| <div>iFrame ⓘ</div>        | <div>面板 ⓘ</div>   |
| <div>选项卡 ⓘ</div>           | <div>锚点导航 ⓘ</div> |
| <div>包裹 ⓘ</div>            | <div>表格视图 ⓘ</div> |
| <div>Web Component ⓘ</div> |                   |

• 按钮组件

顾名思义是按钮类的组件，可以配置一些按钮类型的事件和交互。

按钮



• 其他组件

其他组件是指没有明确分类的组件，如提示组件，表格组件、分隔符组件等。以提示组件为例，可以配置特定触发条件下的提示效果。

其他



• 展示组件

顾名思义是主要实现展示效果的组件，如日期展示、图片集、图表、JSON 展示等。

展示

|                                                                                                                                                                                   |                                                                                                                                                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  卡片             |  卡片列表        |
|  轮播图            |  图表          |
|  折叠器            |  日期展示        |
|  日期时间展示         |  图片展示        |
|  图片集            |  JSON展示      |
|  链接             |  列表          |
|  日志             |  映射          |
|  Markdown       |  纯文本         |
|  进度展示       |  状态显示    |
|  Steps 步骤条  |  走势图     |
|  时间展示       |  文字      |
|  代码高亮       |  当前用户信息  |

- 部分组件未在上述图片中展示，更多组件样式请在组件列表中获取。如图表组件、轮播图等。
- 如需在某个组件内添加组件，可在编辑区选中目标组件，左侧组件面板会列出所有可选组件。

Angular组件

React、Vue、jQuery 组件都可以通过自定义组件的方式引入，但 Angular 不行，因为 Angular 本身是一个框架，它除了 UI 以外还内置了路由和数据请求等功能，导致 Angular 组件并不能直接嵌入到其它框架中。

目前要嵌入 Angular 组件，需要进行三步操作：

1. 对 Angular 组件进行改造，支持打包为 Web Component
2. 在爱速搭应用设置中引入外部 JS/CSS
3. 在页面编辑器中引入 Web Component 组件

下面具体说明每一步的操作

## Angular 组件改造

改造细节请参考[官方文档](#)，改造完后的 Angular 组件就能嵌入到任意页面中。

除了编译为 Web Component，在还有个需要注意的地方是 service 的调用，在普通 Angular 项目中数据请求一般是通过 service 实现的，UI 组件不关心，但打包成 Web Component 组件后无法修改内部实现，因此需要将 service 外部接口所需的配置也放在组件属性配置中，比如下面这个组件示例

```
@Component({
  selector: "my-select",
  templateUrl: "./select.component.html",
})
export class SelectComponent {
  users: User[] = [];

  constructor(
    private userService: UserService
  ) {}

  ngOnInit() {
    this.users = this.userService.getItems();
  }
}
```

数据获取接口是通过外部 `UserService`，组件本身并不知道数据是从哪获取的，这个组件打包成 Web Component 后，对外使用时就是下面的形式

```
<my-select></my-select>
```

内部 api 地址被封装起来了，导致无法更改，也无法使用爱速搭平台的后端代理功能，因此我们需要做如下改造

```
@Component({
  selector: "my-select",
  templateUrl: "./select.component.html",
})
export class SelectComponent {
  @Input() api: string;

  users: User[] = [];

  constructor(
    private http: HttpClient
  ) {}

  ngOnInit() {
    this.users = this.http.get(this.api);
  }
}
```

上面改造的要点是将 api 变成组件属性，这时使用组件就能通过如下形式，是的组件可以控制 api 来源

```
<my-select api="http://xxx"></my-select>
```

改造完成后续需要使用 `npm run build` 打包成 js 文件。

打包完后，需要测试一下页面中放入该组件是否能正常运行，测试页面类似如下

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Elements</title>
  </head>
  <body>
    <my-select></my-select>
    <script src="runtime-es5.3d14b3ad980d82585080.js"></script>
    <script src="polyfills-es5.31618c6906826ad6deb8.js"></script>
    <script src="main-es5.3745a270d0c7c1fecdf6.js"></script>
  </body>
</html>
```

确认没问题后就可以进入下一步了

## 应用引入外部依赖

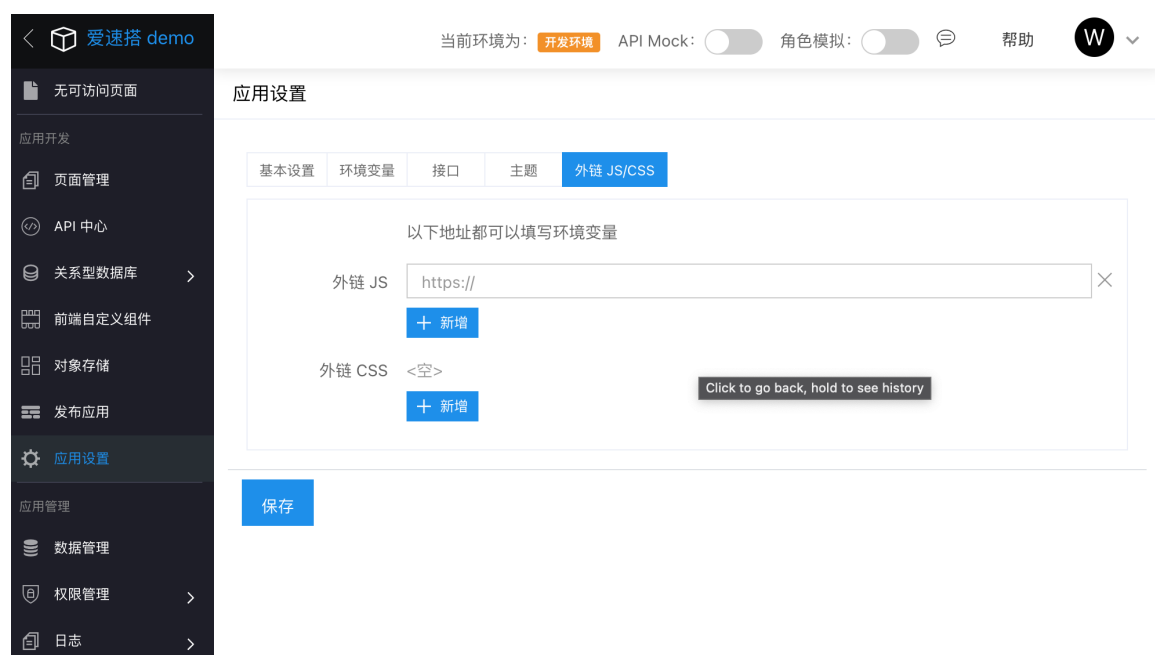
经过前面的改造和编译后，一般会有如下文件产出（不同版本的 Angular 可能会有区别）：

```
3rdpartylicenses.txt
favicon.ico
index.html
main-es2017.1a27f1fc3b52e01f2273.js
main-es5.1a27f1fc3b52e01f2273.js
polyfills-es2017.79d8baa3acf5316d4e1a.js
polyfills-es5.31618c6906826ad6deb8.js
runtime-es2017.3d14b3ad980d82585080.js
runtime-es5.3d14b3ad980d82585080.js
styles.5abc20b315a4b266fdb5.css
```

其中我们只需要 es5 相关的 `runtime-es5.xx.js` `polyfills-es5.xx.js` 和 `main-es5.xx.js`。

将这几个文件上传到用户浏览器可以访问的 CDN 上。

进入「应用设置」-「外链 JS/CSS」中，将这前面上传的文件地址添加进来，如图所示



添加完后需要刷新页面才会生效。

## 在页面编辑器中引入 Web Component



进入页面编辑器，在左侧组件列表中找到「Web Component」，添加到页面中，通过右侧属性面板修改它的标签及属性。

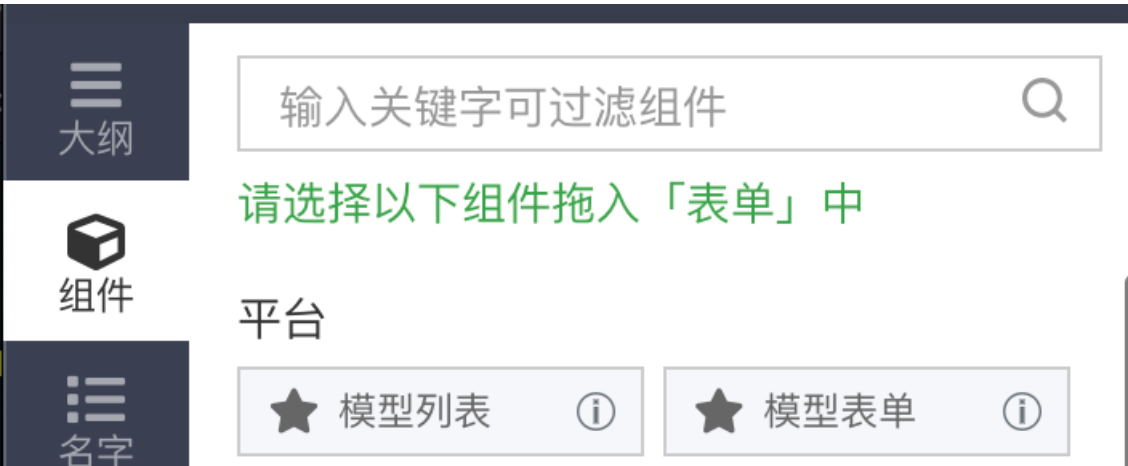


这样就完成了 Angular 组件在页面中的显示。

模型组件

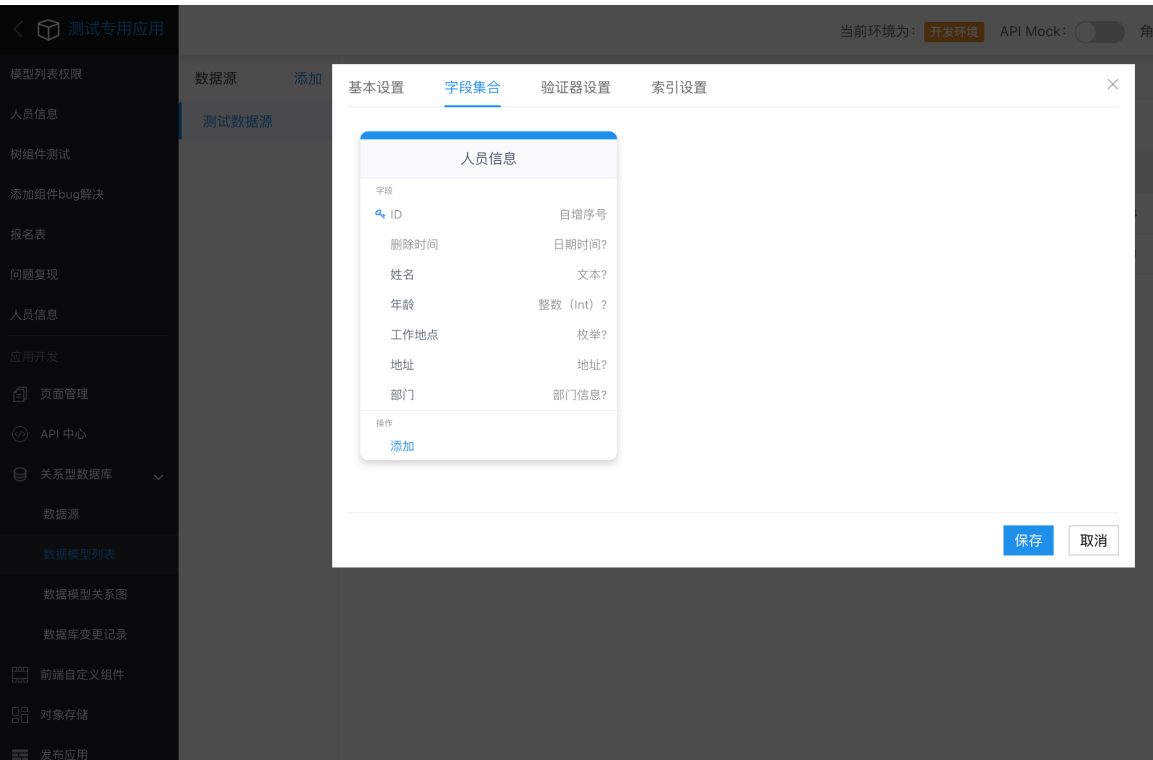
模型组件简介

目前爱速搭平台支持的模型组件有两种，模型列表和模型表单, 分别对应增删改查列表和单条数据的填写/查看/更新等功能



模型组件的使用方法

- 在关系型数据库中新建一个数据模型



- 往页面中添加组件时，可以选择组件绑定的数据模型

快速构建模型列表

模型选择

测试数据源

请选择

Q 输入内容进行检索

人员信息

树状模型

确认

取消

模型列表

模型列表对应的是对数据模型的增/删/改/查操作

展示字段

- 默认展示字段

模型列表可以通过设置默认展示字段来限制一般用户可见的字段 如下图配置默认展示字段为 ID 姓名 年龄

默认展示字段

ID

启用

姓名

启用

年龄

启用

身份证号

启用

国籍

启用

则运行态页面中的模型组件只展示 ID 姓名 年龄 三个字段

默认展示字段示例

新增

上传 Excel 数据

导出数据

批量编辑

批量删除

| <input type="checkbox"/> | ID | 姓名 | 年龄 | 操作                                                       |
|--------------------------|----|----|----|----------------------------------------------------------|
| <input type="checkbox"/> | 7  | 张三 | 1  | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |
| <input type="checkbox"/> | 8  | 张三 | 1  | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |

设置/更改默认展示字段的方式有三种：

- 模型组件创建阶段左侧的“列表展示”标签页、创建后右侧面板的“更改”按钮，以及直接在右侧面板更改保存
- 选中组件点击右侧面板的“更改”按钮
- 直接在右侧面板更改

- 不同角色区别展示字段

某些时候，我们希望不同的角色在进入页面时根据角色权限看到不同字段，那么可以在设置面板中开启“不同角色区别展示字

段”，并为不同角色设置不同的默认展示字段

如下图为不同角色设置了不同的展示字段

默认展示字段

ID

姓名

年龄

身份证号

国籍

启用

启用

启用

启用

启用

不同角色区别展示字段

选择角色

数据模型权限测试应用-管理员

展示字段

ID

姓名

年龄

身份证号

国籍

启用

启用

启用

启用

启用

+ 新增

当前用户为普通用户时：

首页 / 默认展示字段示例

当前环境为：开发环境

API Mock：API Mock

角色模拟：角色模拟

数据模型权限测试应用-用户

帮助

默认展示字段示例

新增

上传 Excel 数据

导出数据

批量编辑

批量删除

| ID | 姓名 | 年龄 | 操作       |
|----|----|----|----------|
| 7  | 张三 | 30 | 详情 编辑 删除 |
| 8  | 李四 | 20 | 详情 编辑 删除 |

当前用户为应用管理员时：

61

首页 / 默认展示字段示例

当前环境为：

开发环境

 API Mock： 角色模拟： 数据模型权限测试应用-管理员  帮助 

P

首页

默认展示字段示例

默认展示字段示例

新增

上传 Excel 数据

导出数据

批量编辑

批量删除

| <input type="checkbox"/> | ID | 姓名 | 年龄 | 身份证号               | 国籍 | 操作                                                       |
|--------------------------|----|----|----|--------------------|----|----------------------------------------------------------|
| <input type="checkbox"/> | 7  | 张三 | 30 | 440982199002121234 | 中国 | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |
| <input type="checkbox"/> | 8  | 李四 | 20 | 440982199002121234 | 中国 | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |

如果一个用户分属不同角色，展示的字段是这个用户所属的所有角色可见字段的并集

预设查询条件

默认预设查询条件

默认预设查询条件实现的是数据行的筛选，数据接口会按预设查询条件过滤掉不希望给用户呈现的数据行

如下面的例子，当预设查询条件设置为 在职时间 小于或等于 5 时，进入页面的用户就只能看到在职时间小于等于5年的人员数据

预设查询条件配置：

模型选择

测试数据源

人员信息

功能

☐ 条件查询

☒ 新增

☒ 编辑

☒ 查看详情

☒ 删除

☐ 批量删除

☐ 批量编辑

☐ 数据导入

☐ 数据导出

列表展示

新增表单

编辑表单

查看详情

默认展示字段

ID

姓名

在职时间

工作地点

地址

部门

启用

启用

启用

启用

启用

启用

不同角色区别展示字段

默认预设查询条件

并且

在职时间

小于或等于

5

默认查询条件，无须用户指定，当前列表默认就会应用这些条件。

不同角色区别预设查询条件

确认

取消

页面显示：

| ID | 姓名    | 在职时间 | 工作地点 | 地址             | 部门     | 操作                                                       |
|----|-------|------|------|----------------|--------|----------------------------------------------------------|
| 1  | 张三二55 | 1    | 中国   | 天津市 天津市市辖区 河东区 | 测试专用组织 | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |
| 2  | 李四    | 1    | 中国   | 天津市 天津市市辖区 河东区 | 测试专用组织 | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |
| 3  | 王五    | 1    | 海外   | 天津市 天津市市辖区 河北区 | 测试专用组织 | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |
| 4  | 赵六    | 5    | 海外   |                |        | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |
| 5  | 王某某   | 2    | 海外   |                |        | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |

● 不同角色区别预设查询条件

预设查询条件同样也支持按不同角色设置，如下图的配置(留空为不筛选)：

默认预设查询条件

并且

在职时间

小于或等于

5

默认的查询条件，无须用户指定，当前列表默认就会应用这些条件。

不同角色区别预设查询条件



选择角色

测试专用应用-管理员

预设查询条件

并且

空

预设的查询条件，无须用户指定，当前列表默认就会应用这些条件。

+ 新增

当前用户为普通用户时：



当前用户为应用管理员时：



条件查询

- 模糊查询 模糊查询是对文本类型字段的模糊匹配，开启之后模型列表头部工具栏会出现相应的关键词输入框

配置：



页面展示效果：



简单查询

简单查询开启后会在模型列表组件的条件查询工具栏生成与字段类型对应的搜索选项框，

如下图开启了三个字段的简单查询

条件查询

列表展示

新增表单

编辑表单

组合查询

模糊查询

name

配置模糊查询覆盖哪些字段

启用

条件查询框里就会生成三个对应的输入组件

新增

上传 Excel 数据

导出数据

批量编辑

批量删除

四

| ID | 姓名 | 在职时间 | 工作地点 | 地址             | 部门     | 操作                                                       |
|----|----|------|------|----------------|--------|----------------------------------------------------------|
| 2  | 李四 | 1    | 中国   | 天津市 天津市市辖区 河东区 | 测试专用组织 | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |

组合查询

模型列表功能

模型列表目前具备以下能力，可以在组件的创建和编辑阶段勾选启用能力，默认勾选的能力为编辑、批量编辑、查看详情、删除

模型选择

测试数据源

人员信息

功能

条件查询

新增

编辑

查看详情

删除

批量删除

批量编辑

数据导入

数据导出

条件查询

列表展示

组合查询

模糊查询

- 新增 选中模型列表组件的“新增”按钮，在右侧面板点击“配置弹框内容”

组件路径: 页面 / 内容区 / 模型列表 / 表格 / 新增

模型组件演示页面

「增删改查」编辑辅助区

批量操作

批量编辑

批量删除

条件查询

姓名

在职时间

工作地点

请选择

搜索

新增

新增

上传 Excel 数据

导出数据

批量编辑

批量删除

请输入

| ID | 姓名    | 在职时间 | 工作地点 | 地址             | 部门     | 操作                                                       |
|----|-------|------|------|----------------|--------|----------------------------------------------------------|
| 1  | 张三二55 | 1    | 中国   | 天津市 天津市市辖区 河东区 | 测试专用组织 | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |

按钮行为

弹框

配置弹框内容

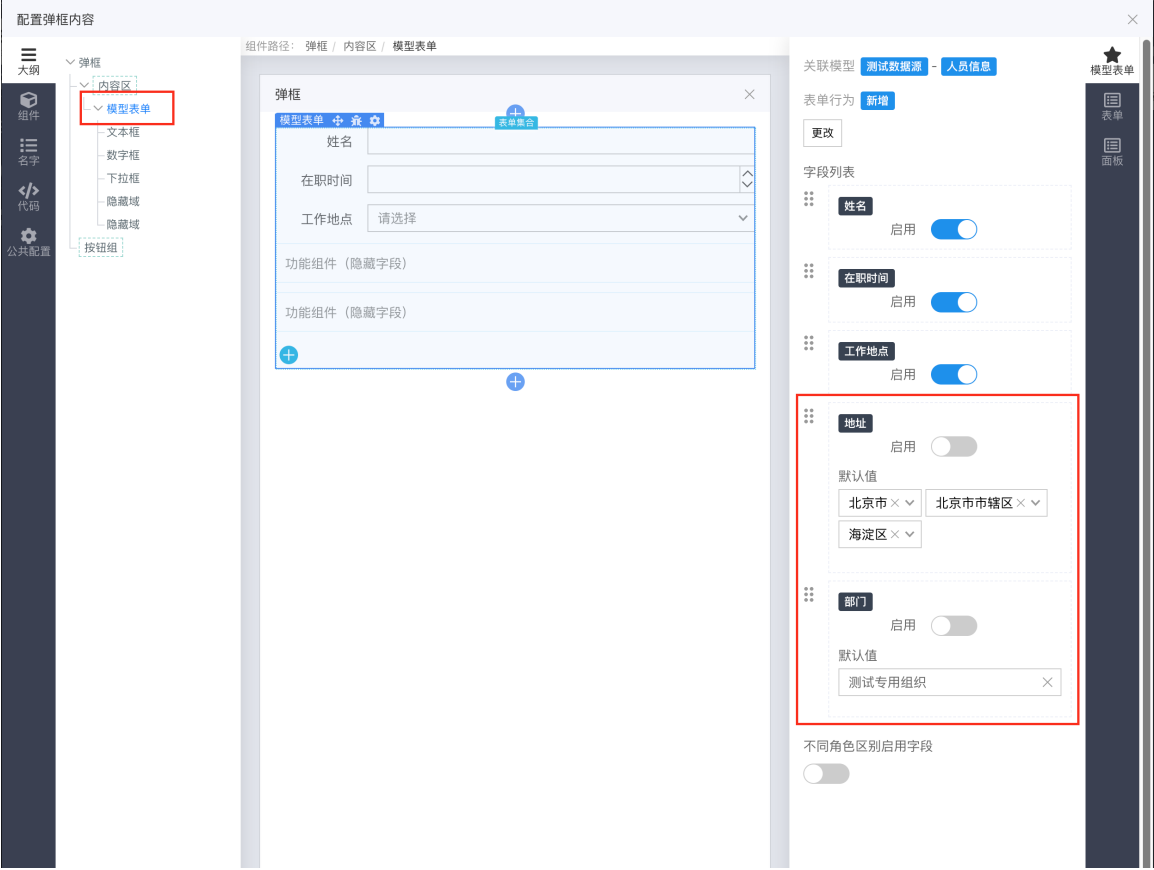
确认文案

刷新目标组件

指定响应组件

在弹框内容编辑界面中选中模型表单组件，可以配置启用字段





新增界面会根据配置的启用字段生成

弹框

姓名

七七

在职时间

1

工作地点

海外

取消

确认

没有开放输入的字段在插入时会用设置的默认值填充

|                          |    |    |   |    |                |        |                                                          |
|--------------------------|----|----|---|----|----------------|--------|----------------------------------------------------------|
| <input type="checkbox"/> | 11 | 七七 | 1 | 海外 | 北京市 北京市市辖区 丰台区 | 测试专用组织 | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |
|--------------------------|----|----|---|----|----------------|--------|----------------------------------------------------------|

新增表单同样支持根据角色配置启用字段，如下图，新增表单本质上是附属于列表的模型表单，具体行为及配置见后续的【模型表单】章节

快速构建模型表单

字段列表

姓名

在职时间

工作地点

地址

部门

启用

启用

启用

启用

启用

默认值

默认值

请选择 ×

请选择

不同角色区别

启用字段

选择角色

测试专用应用-管理员

启用字段

姓名

在职时间

工作地点

地址

部门

启用

启用

启用

启用

启用

新增

确认

取消

• 编辑

编辑是与单行数据绑定的更新表单，支持自定义可编辑字段，其编辑入口为每一行的“编辑”操作按钮

新增 新增 上传 Excel 数据 导出数据 批量编辑 批量删除 请输入

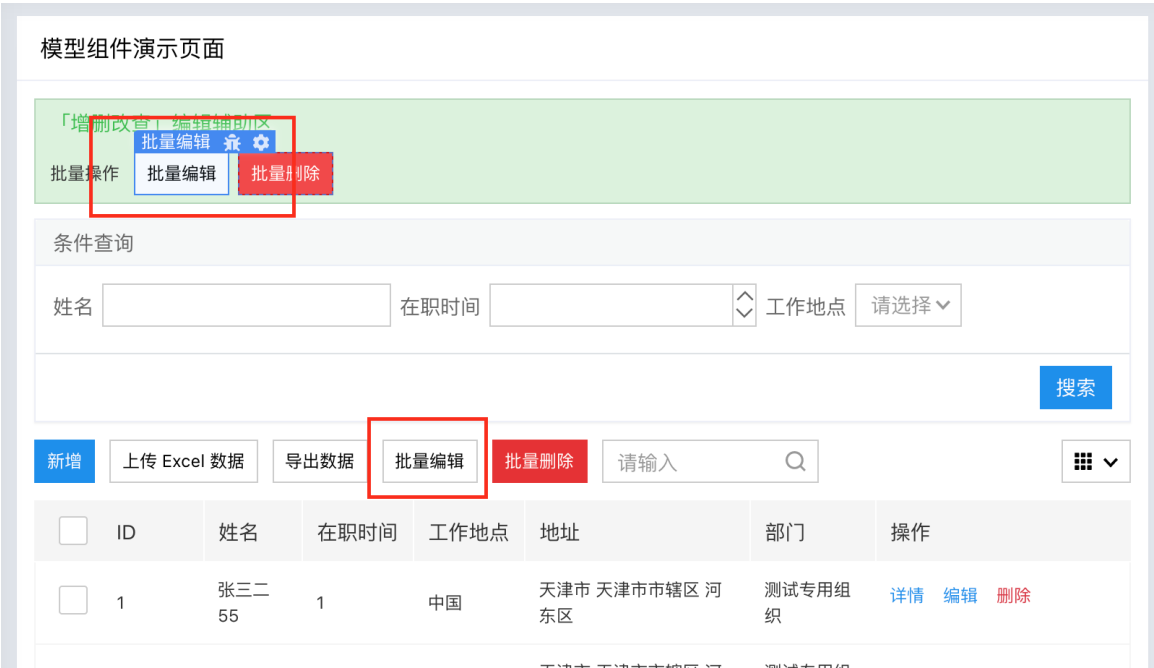
| <input type="checkbox"/> | ID | 姓名    | 在职时间 | 工作地点 | 地址             | 部门     | 操作       |
|--------------------------|----|-------|------|------|----------------|--------|----------|
| <input type="checkbox"/> | 1  | 张三二55 | 1    | 中国   | 天津市 天津市市辖区 河东区 | 测试专用组织 | 详情 编辑 删除 |
| <input type="checkbox"/> | 2  | 李四    | 1    | 中国   | 天津市 天津市市辖区 河东区 | 测试专用组织 | 详情 编辑 删除 |

其余参考【新增】

• 批量编辑

批量编辑指同时更新选中的多条数据，支持自定义可编辑字段，其编辑入口为“增删改查”编辑辅助区的“批量编辑”按钮

68



其余参考【新增】

• 查看详情

查看详情指查看每一行的详情，支持自定义可见字段（与列表解耦，需要单独配置），其编辑入口为每一行的“详情”操作按钮

| <input type="checkbox"/> | ID | 姓名    | 在职时间 | 工作地点 | 地址             | 部门     | 操作                                                       |
|--------------------------|----|-------|------|------|----------------|--------|----------------------------------------------------------|
| <input type="checkbox"/> | 1  | 张三二55 | 1    | 中国   | 天津市 天津市市辖区 河东区 | 测试专用组织 | <a href="#">详情</a> <a href="#">编辑</a> <a href="#">删除</a> |

• 删除

删除单条数据，删除权限对应“预设查询条件”的权限，用户可删除当前账号可访问的数据行，可配置反馈弹窗内容



• 批量删除

删除选中的多条数据，删除权限对应“预设查询条件”的权限，用户可删除当前账号可访问的数据行

• 数据导入

按照要求的格式填写数据并上传数据，目前只支持具有数据管理权限的角色批量导入数据

• 数据导出

导出数据为 Excel，导出的字段与当前用户可见字段相同

以树形结构展示

创建数据模型时，可以在模型的【基本设置】面板开启“树形结构”

基本设置

字段集合

验证器设置

索引设置

\*模型名称

树状模型

\*表名

tree

描述

标题字段

请选择

标题模板

{{ID}} {{字段名}}

软删除

使用标记删除而不是物理删除，方便找回数据

树形结构

树存储方式

物化路径

记录时间

开启后，数据会记录创建时间、更新时间。

记录操作人

开启后，数据会记录创建人、更新人。

手动排序

开启后，数据将按用户指定的顺序展示。

保存

取消

绑定的模型列表会自动以树形模式展示

| √ ID | 父级节点 | 分组名称   | 分组说明 | 联系人手机       | 操作                                                       |
|------|------|--------|------|-------------|----------------------------------------------------------|
| √ 1  |      | 测试分组一号 | 111  | 2222        | <a href="#">查看</a> <a href="#">编辑</a> <a href="#">删除</a> |
| 2    | 1    | 测试分组二号 | 222  | 3333        | <a href="#">查看</a> <a href="#">编辑</a> <a href="#">删除</a> |
| 3    | 1    | 测试分组三号 | 333  | 18612345678 | <a href="#">查看</a> <a href="#">编辑</a> <a href="#">删除</a> |

模型表单

模型表单对应的是数据模型的单条数据管理操作，模型表单支持配置为新增、查看、查看或新增、更新、更新或新增五种模式。

以下图的数据模型为例，数据模型中有 报名人（人员信息）- 姓名（文本）- 照片（图片）- 个人简介（多行文本）四个字段

活动报名表

字段

ID

自增序号

删除时间

日期时间?

报名人

人员信息

姓名

文本

照片

图片?

个人简介

多行文本

操作

添加

在编辑界面可通过拖拽或添加将模型表单添加到页面中

快速构建模型表单

模型选择

测试数据源

活动报名表

表单行为

新增

查看

查看或新增

更新

更新或新增

字段列表

报名人

启用

姓名

启用

照片

启用

个人简介

启用

不同角色区别

启用字段

确认

取消

预设条件

模型表单的预设条件制定了表单预加载的数据内容

如果有多条满足预设条件的数据，则模型表单组件默认展示 / 编辑第一条匹配的记录

如下图的预设查询条件配置为“报名人”“等于”“当前用户”

快速构建模型表单

模型选择

测试数据源

活动报名表

表单行为

新增

查看

查看或新增

更新

更新或新增

字段列表

ID

启用

报名人

启用

姓名

启用

照片

启用

个人简介

启用

不同角色区别

启用字段

预设条件

并且

报名人

等于

当前用户

无数据提示

数据加载失败!

确认

取消

则页面加载时会自动用报名人等于当前用户的第一条数据填充表单

首页 / 我的报名信息

当前环境为：开发环境

API Mock：API Mock

角色模拟：角色模拟

帮助

P

panzheng

个人中心

退出

首页

我的报名信息

我的报名信息

查看活动报名表

ID 2

报名人

panzheng

姓名

某某

照片

个人简介

这个人很懒，什么也没有留下

提交

启用字段

• 默认启用字段

和模型列表一样，模型表单也支持配置启用字段

如下面的配置：

快速构建模型表单

模型选择

测试数据源

活动报名表

表单行为

新增

查看

查看或新增

更新

更新或新增

字段列表

ID

启用

报名人

启用

姓名

启用

照片

启用

个人简介

启用

不同角色区别

启用字段

预设条件

并且

报名人

等于

当前用户

无数据提示

数据加载失败！

确认

取消

页面运行时只展示启用的字段：

我的报名信息

查看活动报名表

ID

2

报名人

panzheng

姓名

某某

提交

不同角色区别启用字段

模型表单支持按角色进行启用字段的配置，

如下面配置：

快速构建模型表单

模型选择

测试数据源

活动报名表

表单行为

新增

查看

查看或新增

更新

更新或新增

字段列表

ID

启用

报名人

启用

姓名

启用

照片

启用

个人简介

启用

不同角色区别

启用字段

选择角色

测试专用应用-管理员

启用字段

ID

启用

报名人

启用

姓名

启用

照片

启用

个人简介

启用

新增

预设条件

并且

报名人

等于

当前用户

无数据提示

数据加载失败!

确认

取消

当前角色为普通用户：

首页 / 我的报名信息

当前环境为：开发环境

API Mock：

角色模拟：

测试专用应用-用户

帮助

P

首页

我的报名信息

我的报名信息

查看活动报名表

ID 2

报名人 panzheng

姓名 某某

提交

当前角色为应用管理员：

74



首页 / 我的报名信息

当前环境为：

开发环境

API Mock：角色模拟：测试专用应用-管理员

帮助

我的报名信息

查看活动报名表

ID2

报名人panzheng

姓名某某

照片

个人简介这个人很懒，什么也没有留下

提交

模型表单行为

- 新增

模型表单默认行为为新增，其行为是往目标数据模型中新增一条数据

以一个简单的模型表单为例

编辑状态

大纲

组件

名字

代码

公共组件

页面

模型表单

表单集合

人员选择

文本框

图片上传

多行文本框

按钮组

组件路径：页面 / 内容区 / 模型表单

报名表-正式版

新增活动报名表

报名人panzheng

姓名

照片

个人简介

按钮组

关联模型测试数据库 - 活动报名表

表单行为新增

更改

表单行为新增查看查看或新增更新更新或新增

字段列表

报名人

启用

姓名

启用

照片

启用

个人简介

启用

不同角色区别启用字段

页面运行状态

报名表-正式版

新增活动报名表

\* 报名人

panzheng ×

\* 姓名

某某

照片



\* 个人简介

这个人很懒，什么也没有留下

提交

点击提交之后前往数据管理页面，可以看到相应的数据模型下多了一条数据

测试数据源

人员信息

树状模型

有不允许为空字段的模型

活动报名表

活动报名表列表

+ 新增「活动报名表」

上传 Excel 数据

导出数据

刷新

更多

清空数据

| ID | 报名人      | 姓名 | 照片                                                                                            | 个人简介          | 操作                                                 |
|----|----------|----|-----------------------------------------------------------------------------------------------|---------------|----------------------------------------------------|
| 2  | panzheng | 某某 | <div></div> | 这个人很懒，什么也没有留下 | <div><div>查看</div><div>编辑</div><div>删除</div></div> |

查看

对应查看单条数据详情的行为

查看或新增

“查看”和“新增”的复合表单，根据既定的筛选条件，如果数据库中已有数据，则展示已有数据，否则展示新增表单 编辑时需要分别对“查看”和“新增”的表单项进行配置

更新

对应单条数据编辑行为

更新或新增

“更新”和“新增”的复合表单，根据既定的筛选条件，如果数据库中已有数据，则展示已有数据的编辑界面，否则展示新增表单 编辑时需要分别对“更新”和“新增”的表单项进行配置

编辑器说明

多端编辑模式

页面编辑器默认打开 PC 页面编辑环境，如需配置手机端页面需要先切换到手机页面模式。点击顶部全局工具的【手机】即可进行切换，再次点击为返回 PC 模式。

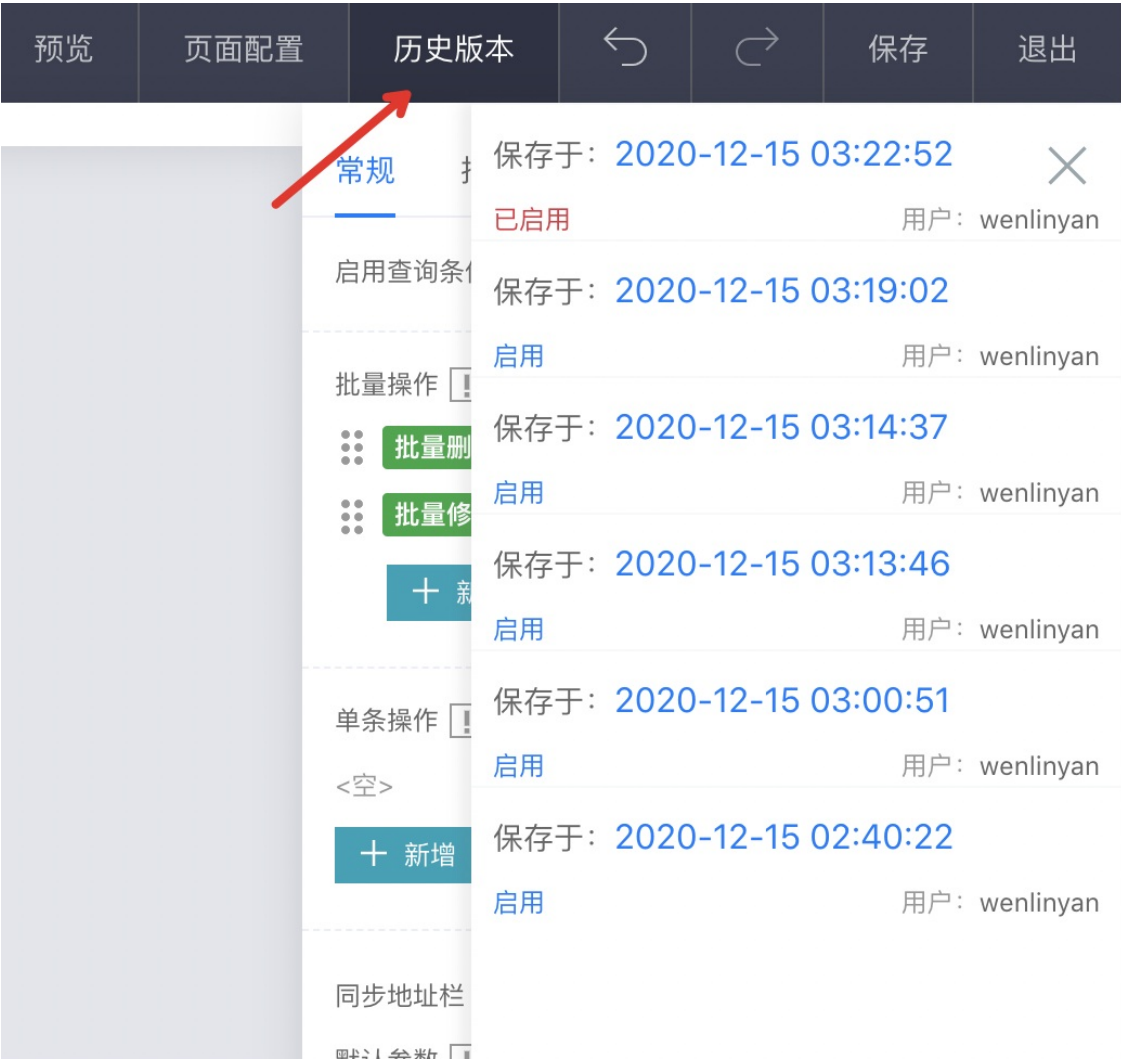


编辑器支持的组件配置规则同 PC 页面编辑模式，配置完成后点击预览可以查看效果。



● 页面回滚

点击全局工具中的【页面历史】可以查看页面历史保存记录，并支持状态回滚。



• 页面配置

页面配置支持对页面创建时填写的配置信息进行修改。

预览

页面配置

历史版本

保存

退出

编辑页面属性

\* 页面类型

普通页面

Markdown

富文本

文件夹

简易页面

使用外部 API，实现各种表单及表格页面。

\* 标题

组件搭建实践

描述

\* 菜单名称

组件搭建实践

菜单图标

访问地址

/company/123456/app/52f40c21f439/

组件搭建实

导航中显示

开

父级页面

请选择父级页面

自定义组件

除了使用爱速搭内置的组件以外，还可以通过创建自定义组价，扩充可使用的组件

前提条件

- 1. 需要对 amis 整体概念有一定的了解，可以查看 [这里](#)
- 2. 需要有一定的 JavaScript 基础
- 3. 至少能够熟练使用 [React](#)、[Vue](#)、[jQuery](#) 中任意一门框架

基本流程

现在说明一下如何创建和使用一个自定义组件

组件创建

进入自定义组件模块，点击左上角「新增组件」按钮，填写如下组件配置：

新增组件

\*组件名称

组件类型 ☒ 展示类 ☐ 普通表单项 ☐ 选择器表单项 ☐ 其他  
主要用于展示

\*组件key    
防止与amis内置组件冲突，默认添加 custom- 前缀用以区分

框架 ☒ React ☐ Vue ☐ jQuery

是否启用 ☒ 开

说明

- 组件名称：设置该自定义组件的名称
- 组件类型：选择新建组件的类型，主要有「展示类」、「普通表单项」、「选择器表单项」、「其他」4类，后面文档会详细介绍
- 组件 key：在 amis schema 配置中的 type 值
- 框架：可选 React、Vue 和 jQuery 作为开发框架
- 是否启用：可以选择组件是否启用
- 说明：组件的说明，描述组件

如上图填写完组件配置后，点击确认，进入组件编辑界面

### 组件代码编辑

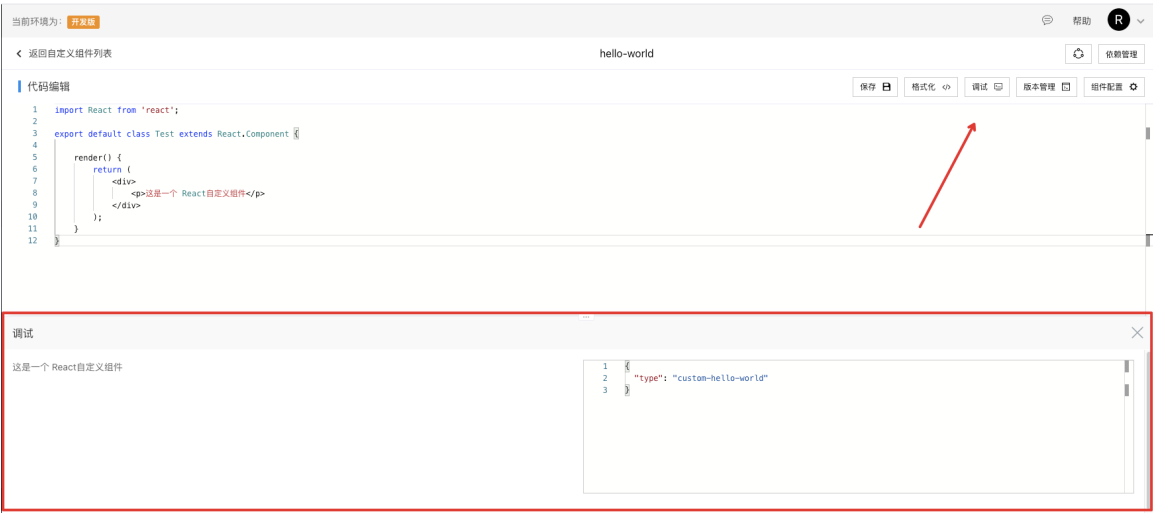
自定义组件代码核心是用 `export default` 来导出编写好的自定义组件，如下：

```
import React from 'react';

export default class Test extends React.Component {
  render() {
    return <div>这是一个 React 自定义组件</div>;
  }
}
```

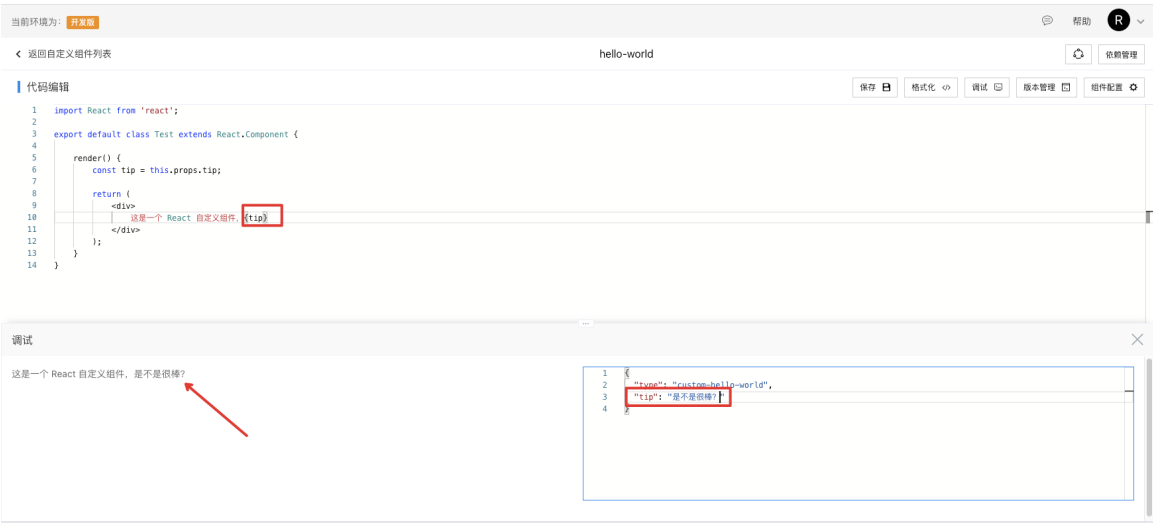
### 调试

组件保存后会自动进入构建状态，调试按钮将禁用，当构建成功后，即可点击调试按钮，进行组件调试操作：



调试面板左侧为组件渲染后效果，右侧为 amis schema，可以调整右侧 schema，动态调试组件效果。

例如我们修改代码如下：



我们获取 props 中的 tip 属性，然后我们在调试面板右侧，修改 amis schema，添加 tip 属性，可以发现组件渲染出该文本

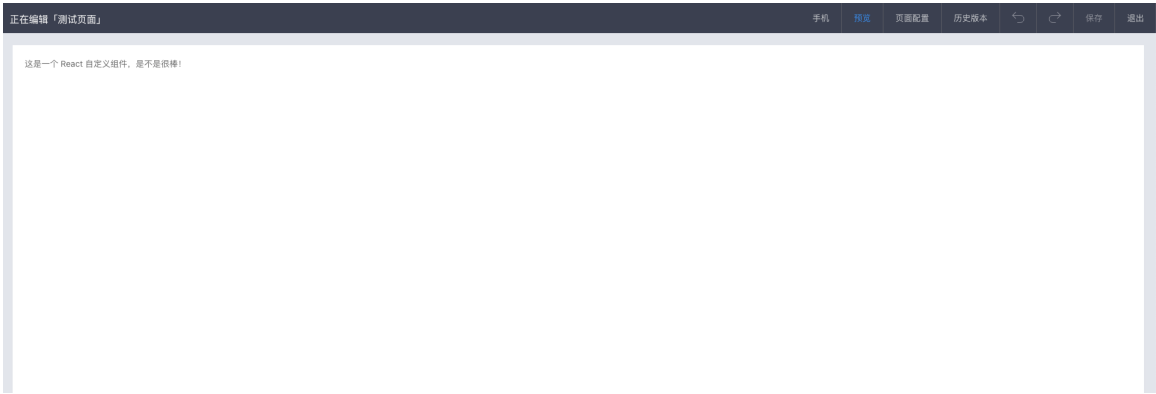
修改完代码后，需要保存并构建完成后，才可以看到最新的组件效果

组件使用

在 amis schema 中，可以通过 type 来使用自定义组件，例如上例中我们新建的组件 key 为 custom-hello-wolrd，且是普通展示类，则可以通过 "type": "custom-hello-wolrd" 使用该自定义组件

我们新建一个测试页面，并编辑页面 schema 如下：

```
{
  "type": "page",
  "body": {
    "type": "custom-hello-world",
    "tip": "是不是很棒！"
  }
}
```

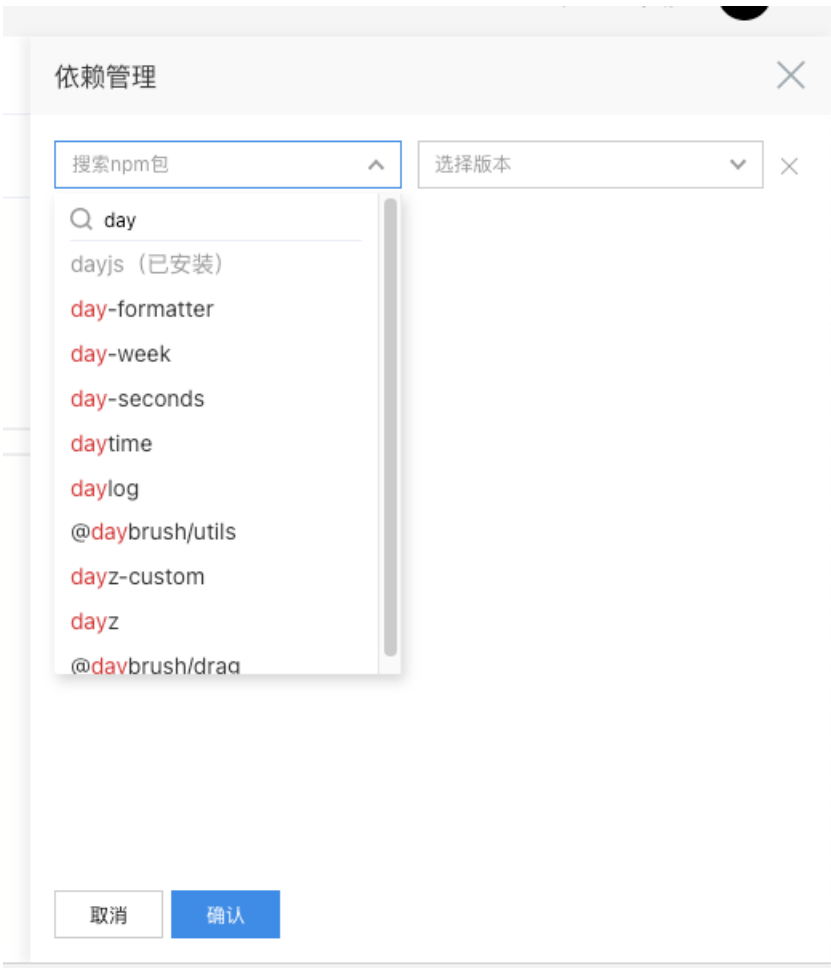


可以看到上图中自定义组件已经可以正常使用了。

添加 npm 依赖

平台支持引入在线第三方 npm 库进行辅助开发

首先点击右上角依赖管理，搜索并添加最新版本的 day.js，该插件已内置安装，因此可以直接使用：



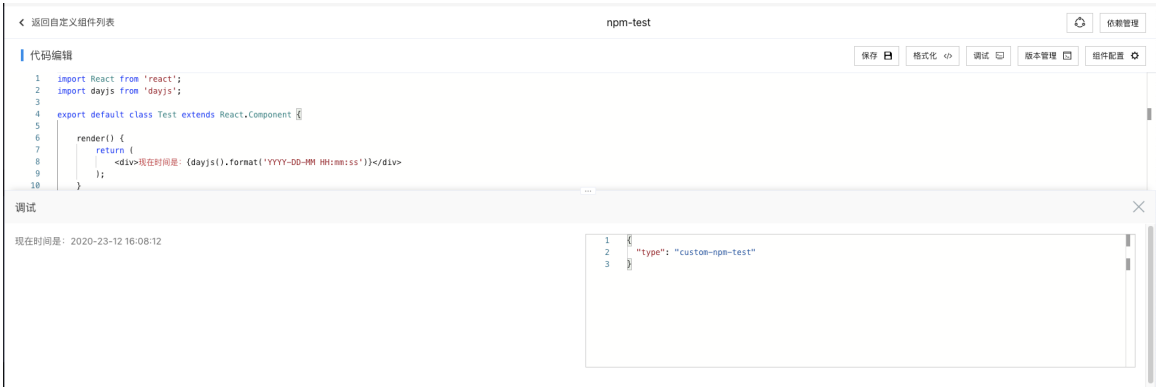
我们使用展示类，并选中 React 作为开发框架，下面我们引用 day.js 并打印一下当前时间，代码如下：

```
import React from 'react';
import dayjs from 'dayjs';

export default class Test extends React.Component {
  render() {
    return <div>现在是时间是：{dayjs().format('YYYY-DD-MM HH:mm:ss')}</div>;
  }
}
```

效果如下：





内置已安装的插件，都可以直接使用

开发框架

平台支持 React、Vue、jQuery 作为组件的开发框架

React（推荐）

平台以及 amis 本身是基于 React 开发的，因此使用 React 可以更好地整合开发组件

导出组件类

核心方式是通过 `export default` 来导出组件类，从而使组件生效

```
import React from 'react';

export default class Test extends React.Component {
  render() {
    return <div>这是一个 React 自定义组件</div>;
  }
}
```

使用 UI 组件库

可以使用 npm UI 组件库，例如 [Ant Design](#)。

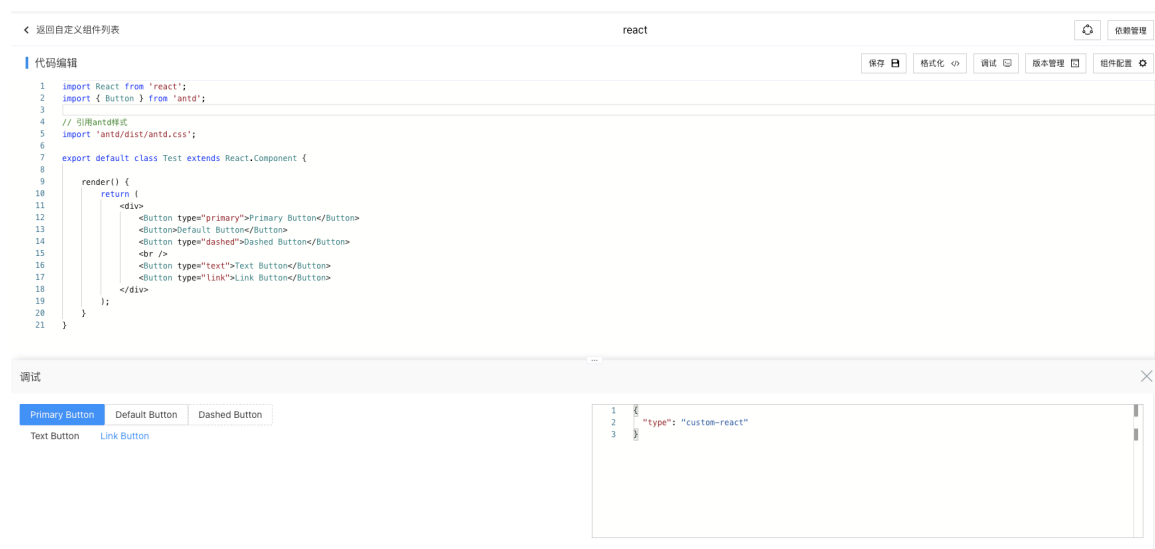
首先依赖管理中添加 antd 组件库，然后选择展示类，和 React 作为开发框架，然后编写代码如下：

```
import React from 'react';
import { Button } from 'antd';

// 引用 antd 样式
import 'antd/dist/antd.css';

export default class Test extends React.Component {
  render() {
    return (
      <div>
        <Button type="primary">Primary Button</Button>
        <Button>Default Button</Button>
        <Button type="dashed">Dashed Button</Button>
        <br />
        <Button type="text">Text Button</Button>
        <Button type="link">Link Button</Button>
      </div>
    );
  }
}
```

效果如下：



上例中我们展示了 antd 的 Button 组件

## Vue 2

我们下面开发一个简单的自定义组价，打印一行文本：

```
export default {
  template: '<div>这是一个 {{name}} 自定义组件</div>',

  data: {
    name: 'Vue'
  },

  methods: {
    foo() {
      console.log('foo');
    }
  },

  created() {
    this.foo();
  }
};
```

## 使用第三方 UI 组件库

可以使用第三方组件库，例如 [Element-UI](#)

首先依赖管理里添加 element-ui 组件库，然后新建组件，选择展示类组件，勾选 Vue 作为开发框架，代码如下：

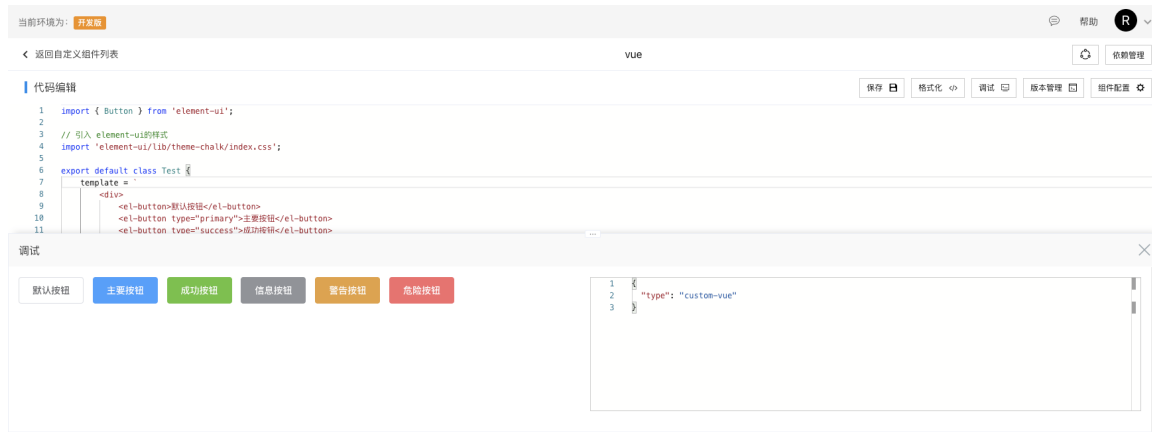
```
import {Button} from 'element-ui';

// 引入 element-ui 的样式
import 'element-ui/lib/theme-chalk/index.css';

export default class Test {
  template = `
    <div>
      <el-button>默认按钮</el-button>
      <el-button type="primary">主要按钮</el-button>
      <el-button type="success">成功按钮</el-button>
      <el-button type="info">信息按钮</el-button>
      <el-button type="warning">警告按钮</el-button>
      <el-button type="danger">危险按钮</el-button>
    </div>
  `;

  components = {
    'el-button': Button
  };
}
```

效果如下：



### Vue 3

请注意，在一个应用中不能混用 Vue 2 和 Vue 3 组件，因为 vue 只有一个 package 名称，无法同时存在多个版本

Vue 3 在接口层面做了很大改动，它的使用方式和 Vue 2 不同。

默认展现的示例是：

```
export default ({createApp, props, funcs}) => {
  const app = createApp({
    template: `
      <div id="event-handling">
        <p>{{ message }}</p>
        <button v-on:click="reverseMessage">反转 Message</button>
      </div>
    `,
    data() {
      return {
        message: 'Hello Vue.js!'
      };
    },
    methods: {
      reverseMessage() {
        this.message = this.message.split('').reverse().join('');
      }
    }
  });
  return app;
};
```

爱速搭会传递 createApp 方法，而插件需要返回构建好的 app 对象。

### 使用第三方 UI 库

首先在依赖中添加，然后通过 app.use 方法来添加这个第三方 UI 库的插件

```
import './style.scss'; // 自定义样式，文件名称不可修改

import ElementPlus from 'element-plus';
import 'element-plus/dist/index.css';

export default ({createApp, props, funcs}) => {
  const app = createApp({
    template: `
<el-row>
  <el-button>默认按钮</el-button>
  <el-button type="primary">主要按钮</el-button>
  <el-button type="success">成功按钮</el-button>
  <el-button type="info">信息按钮</el-button>
  <el-button type="warning">警告按钮</el-button>
  <el-button type="danger">危险按钮</el-button>
</el-row>
`
  });

  app.use(ElementPlus);

  return app;
};
```

## jQuery

平台内部支持用 jQuery 进行开发，且进行了一层简单的封装，方便用户更好的操作。下面我们创建一个简单的自定义组件，打印一行文本：

```
import $ from 'jquery';

export default {
  template: `这是个`,

  /**
   * 组件挂载的时候调用
   */
  onMount(props) {
    this.foo();
  },

  /**
   * amis props 更新的时候调用
   */
  onUpdate(props, prevProps) {},

  /**
   * 组件销毁的时候调用
   */
  onUnmout(props) {},

  foo() {
    // this.$root 获取当前顶级dom
    $(this.$root).append(' jQuery 自定义组件');
  }
};
```

效果如下：



### 获取 schema 配置的属性

可以在 schema 中配置属性，然后在组件中获取，例如：

React 中：

```
import React from 'react';

export default class Test extends React.Component {
  render() {
    const tip = this.props.tip;
    return <div>提示文本: {tip}</div>;
  }
}
```

使用组件时可以配置 schema 如下：

```
{
  "type": "xxx", // 组件的 key 值
  "tip": "这是一段提示"
}
```

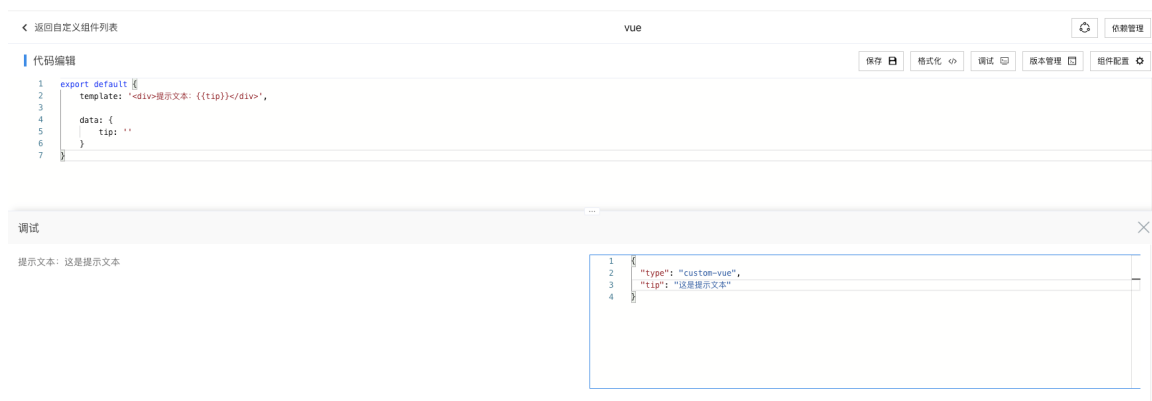
Vue 中：

```
export default {
  template: '<div>提示文本: {{tip}}</div>',

  data: {
    tip: '' // 需要声明一个空的值，否则可能会报错
  }
};
```

由于该 [Vue 特性](#)，需要在使用前声明一个空的默认值

效果如下：



jQuery 中

```
import $ from 'jquery';

export default {
  template: `提示文本：<span id="tip"></span>`,

  onMount(props) {
    $('#tip').text(props.tip);
  },

  onUpdate(props) {
    $('#tip').text(props.tip);
  }
};
```

效果如下：



暴露的变量和方法

render

渲染器方法，可以在自定义组件中渲染已有的 amis 组件

适用条件

- 开发框架：React
- 组件类型：展示类、表单项、选择器表单项

函数签名

```
(region, node, subProps) => JSX.Element;
```

- region：给当前组件设置一个 key
- node：amis 组件的配置项
- subProps：可以不填，额外的一些配置项

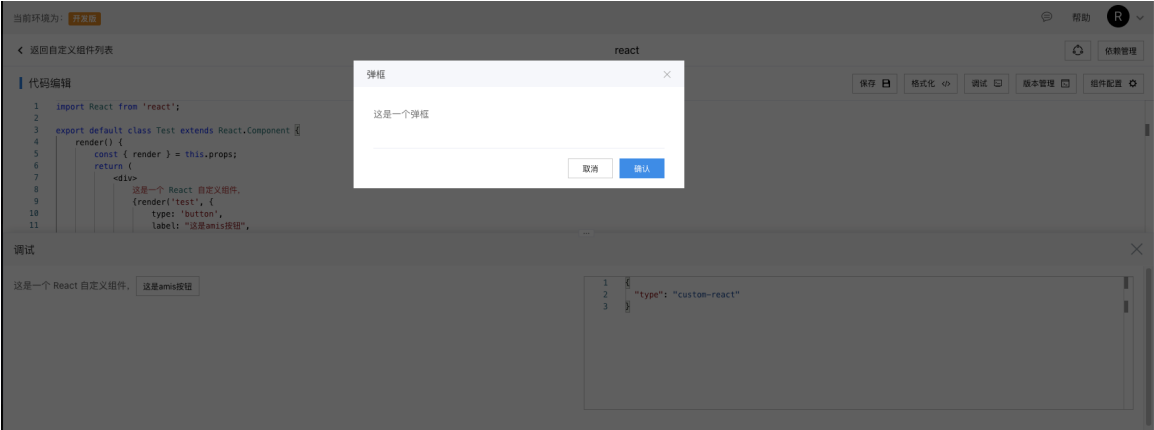
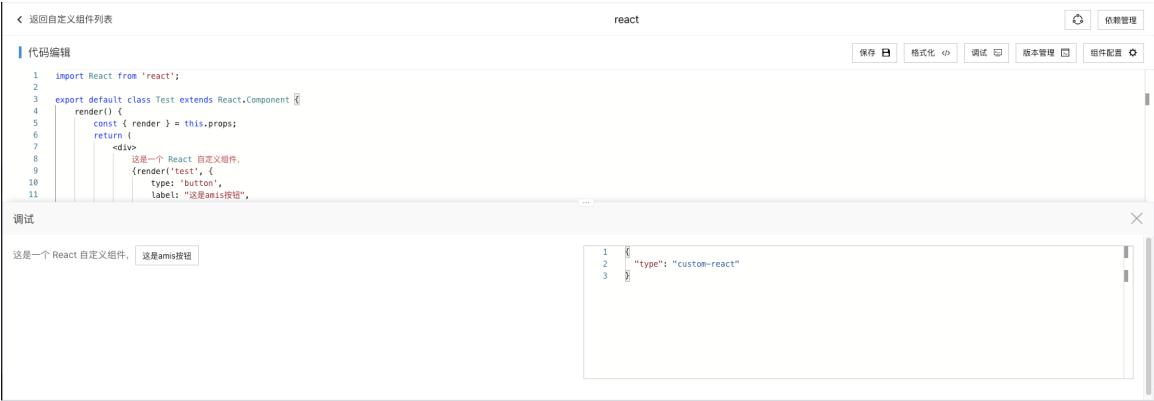
使用方法

React 中：

```
import React from 'react';

export default class Test extends React.Component {
  render() {
    const {render} = this.props;
    return (
      <div>
        这是一个 React 自定义组件，
        {render('test', {
          type: 'button',
          label: '这是amis按钮',
          actionType: 'dialog',
          dialog: {
            title: '弹框',
            body: '这是一个弹框'
          }
        })}
      </div>
    );
  }
}
```

效果如下：



onAction

调用 amis 内置的行为，可参考 [行为](#)

适用条件

- 开发框架：React、Vue、jQuery
- 组件类型：展示类、表单项、选择器表单项

函数签名



```
(event, action, ctx) => void;
```

- `event`：可忽略，传入 `null` 即可；
- `action`：传入需要执行的行为对象配置，参考 [行为](#)；
- `ctx`：给当前行为内传入一些数据。如果没有则传入空对象 `{}`，否则会报错

## 使用方法

我们来通过该方法实现：点击按钮，然后调起一个 amis 弹框。

### React 中

```
import React from 'react';

export default class Test extends React.Component {
  constructor() {
    super();
    this.handleClick = this.handleClick.bind(this);
  }

  handleClick() {
    const onAction = this.props.onAction;

    onAction(
      null,
      {
        actionType: 'dialog',
        dialog: {
          title: '弹框',
          body: '这是一个amis弹框'
        }
      },
      {}
    );
  }

  render() {
    return <button onClick={this.handleClick}>调起 amis 弹框</button>;
  }
}
```

### Vue 中

```
export default {
  template: '<button v-on:click="handleClick">调起 amis 弹框</button>',

  data: {},

  methods: {
    handleClick() {
      this.$emit('onAction', [
        null,
        {
          actionType: 'dialog',
          dialog: {
            title: '弹框',
            body: '这是一个amis弹框'
          }
        },
        {}
      ]);
    }
  }
};
```

### jQuery 中

```
import $ from 'jquery';

export default {
  template: `<button id="btn">调起 amis 弹框</button>`,

  onMount(props) {
    $('#btn').click(() => {
      props.onAction(
        null,
        {
          actionType: 'dialog',
          dialog: {
            title: '弹框',
            body: '这是一个amis弹框'
          }
        },
        {}
      );
    });
  }
};
```

### value 和 onChange

当编写表单类型的自定义组件时，最重要的是与 Form 数据域的通信，而实现该通信的核心就是 value 属性和 onChange 方法

### 适用条件

- 开发框架：React、Vue、jQuery
- 组件类型：普通表单项、选择器表单项

### 使用方法

核心思路是：

- 拿到 props 中的 value 属性，并赋值给自定义组件内输入框
- 监听自定义组件内输入框 change 事件或 value 值变化，如果有变化，通过 onChange 事件，将新的 value 值同步给 amis 层

下面来演示不同框架下的使用示例：

## React 中

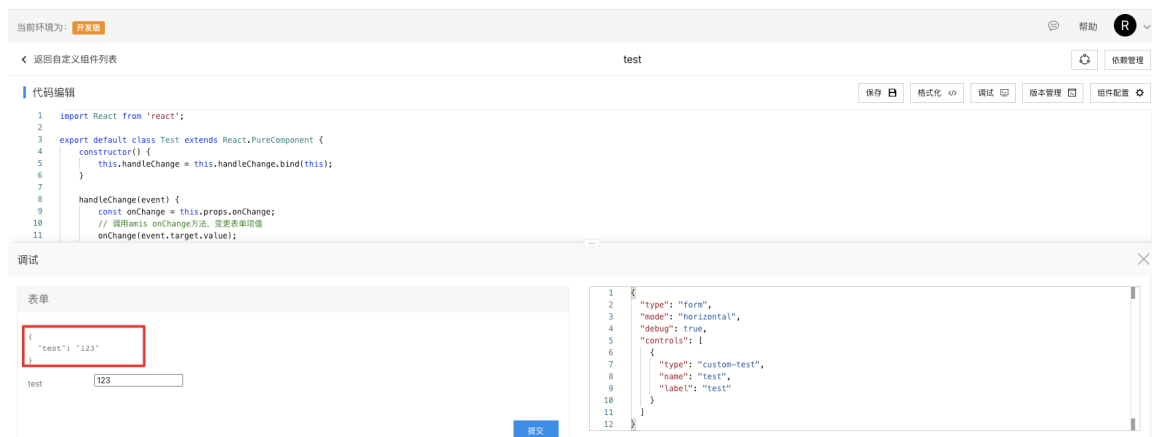
```
import React from 'react';

export default class Test extends React.PureComponent {
  constructor() {
    this.handleChange = this.handleChange.bind(this);
  }

  // 监听 input 的 change 事件，并同步value值
  handleChange(event) {
    const onChange = this.props.onChange;
    // 调用amis onChange方法，同步最新的value值
    onChange(event.target.value);
  }

  render() {
    // 获取 props 中的 value 属性，并赋值给input
    const {value} = this.props;
    return <input type="text" value={value} onChange={this.handleChange} />;
  }
}
```

效果如下：



更改 input 值，可以观察表单数据域的变化。

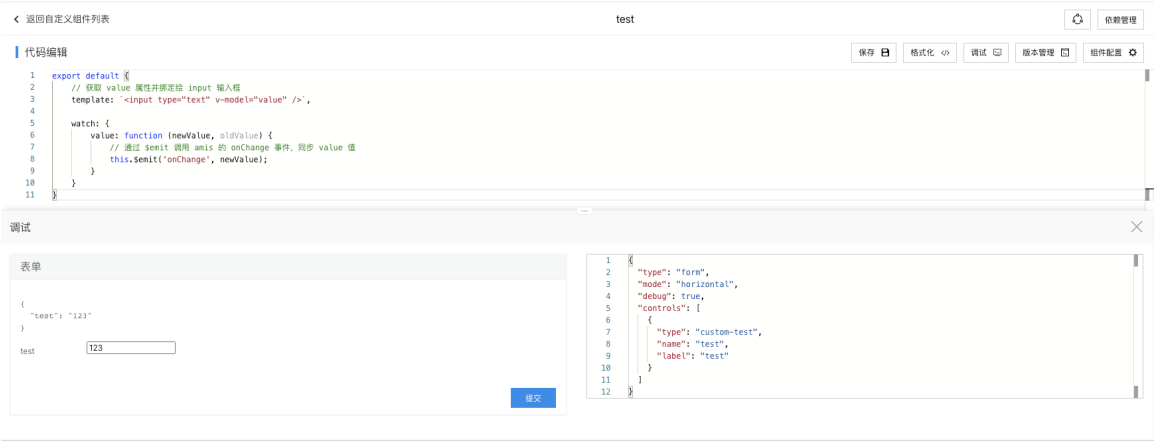
## Vue 中

```
export default {
  // 获取 value 属性并绑定给 input 输入框
  template: `<input type="text" v-model="value" />`,

  watch: {
    value: function (newValue, oldValue) {
      // 通过 $emit 调用 amis 的 onChange 事件，同步 value 值
      this.$emit('onChange', newValue);
    }
  }
};
```

vue 中可以利用 watch 监听器来实现 value 同步逻辑。

效果如下：



更改 input 值，可以观察表单数据域的变化。

jQuery 中

```

import $ from 'jquery';

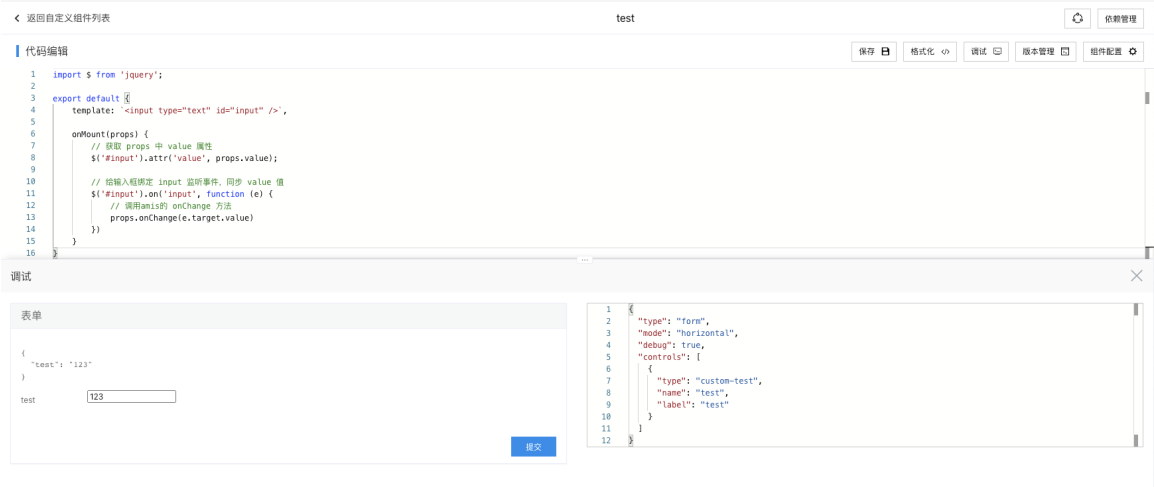
export default {
  template: `<input type="text" id="input" />`,

  onMount(props) {
    // 获取 props 中 value 属性
    $('#input').attr('value', props.value);

    // 给输入框绑定 input 监听事件，同步 value 值
    $('#input').on('input', function (e) {
      // 调用amis的 onChange 方法
      props.onChange(e.target.value);
    });
  }
};

```

效果如下：



更改 input 值，可以观察表单数据域的变化。

onBulkChange

该方法类似于 onChange，不同点在于，它可以批量修改表单项的值。

```
import React from 'react';

export default class Test extends React.PureComponent {
  constructor() {
    this.handleChange = this.handleChange.bind(this);
  }

  handleChange(event) {
    const {name, other, onBulkChange} = this.props;
    const value = event.target.value;
    // 调用amis onChange方法，变更表单项值
    onBulkChange({
      [name]: value,
      [other]: value
    });
  }

  render() {
    // 获取表单项 value 属性
    const {value} = this.props;

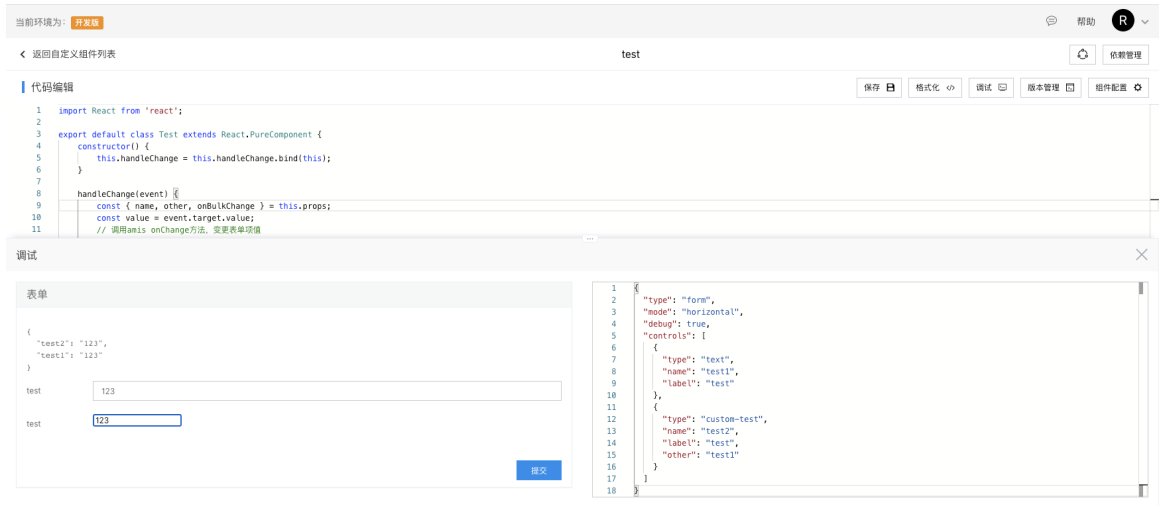
    return <input type="text" value={value} onChange={this.handleChange} />;
  }
}
```

例如上例中，我们实现一个：修改当前输入框，会同步修改另外一个数据框的值，具体 schema 如下

```
{
  "type": "form",
  "mode": "horizontal",
  "debug": true,
  "controls": [
    {
      "type": "custom-test",
      "name": "text1",
      "label": "text1",
      "other": "text2"
    },
    {
      "type": "text",
      "name": "text1",
      "label": "test2"
    }
  ]
}
```

当 text1 变化时，获取 other 属性，该属性配置的是要同步修改的另外一个数据框的 name 值，即 text2，然后调用 onBulkChange，批量修改数据域值

效果如下：



vue 和 jquery 使用逻辑相同，具体使用语法方法见 onChange

options

选择器表单项特有的属性，一个选择器组件总有一组选项，可以提供给用户勾选一项或多项，默认该属性为空数组

适用范围

- 开发框架：React、Vue、jQuery
- 组件类型：选择器表单项

手动配置

你可以手动在 schema 中配置 options 吗，然后在组件内获取并使用，例如：

```

import React from 'react';

export default class Test extends React.PureComponent {
  constructor() {
    this.handleChange = this.handleChange.bind(this);
  }

  handleChange(event) {
    // 调用amis onToggle 方法，变更选择器表单项值
    const {onToggle, options} = this.props;
    const option = options.find(o => o.value === event.target.value);
    onToggle(option);
  }

  render() {
    // 获取表单项 value 和 options 属性
    const {value, options} = this.props;

    return (
      <select value={value} onChange={this.handleChange}>
        {options.map(option => (
          <option key={option.value} value={option.value}>
            {option.label}
          </option>
        ))}
      </select>
    );
  }
}

```

编辑 schema 如下：

```

{
  "type": "form",
  "mode": "horizontal",
  "debug": true,
  "controls": [
    {
      "type": "custom-test",
      "name": "test",
      "label": "test",
      "options": [
        {
          "label": "Option A",
          "value": "a"
        },
        {
          "label": "Option B",
          "value": "b"
        },
        {
          "label": "Option C",
          "value": "c"
        }
      ]
    }
  ]
}

```

这种方式和普通获取 schema 属性无区别。

## 动态加载

选择器表单项可以通过配置 `source` 属性，来动态拉取远程选项，然后在组件更新钩子函数中，获取拉取到的新的 `options`

## setOptions

可以手动调用该方法动态设置 `options`

## 适用范围

- 开发框架：React、Vue、jQuery
- 组件类型：选择器表单项

## 函数签名

```
(options) => void;
```

- `options`：一个对象数组，里面有若干选项

## 使用方法



```

import React from 'react';

export default class Test extends React.PureComponent {
  constructor() {
    this.handleChange = this.handleChange.bind(this);
  }

  // 组件挂载时，调用 setOptions，设置 options
  componentDidMount() {
    const {setOptions} = this.props;
    setOptions([
      {
        label: 'Option A',
        value: 'a'
      },
      {
        label: 'Option B',
        value: 'b'
      },
      {
        label: 'Option C',
        value: 'c'
      }
    ]);
  }

  handleChange(event) {
    // 调用 amis onToggle 方法，变更选择器表单项值
    const {onToggle, options} = this.props;
    const option = options.find(o => o.value === event.target.value);
    onToggle(option);
  }

  render() {
    // 获取表单项 value 和 options 属性
    const {value, options} = this.props;

    return (
      <select value={value} onChange={this.handleChange}>
        {options.map(option => (
          <option key={option.value} value={option.value}>
            {option.label}
          </option>
        ))}
      </select>
    );
  }
}

```

### selectedOptions

由于 [拼接符-delimiter](#)、[拼接值-joinvalues](#)、[提取多选值-extractvalue](#) 的存在，value 值格式可以是多种格式。

selectedOptions 则永远为当前选中的值的数组形式，方便逻辑操作。

### 适用范围

- 开发框架：React、Vue、jQuery
- 组件类型：选择器表单项

## onToggle

选择器表单项专用属性，类似于 onChange，不同的是，该方法需要传入一个完整的选项对象，可以设置选项切换勾选。

## 适用范围

- 开发框架：React、Vue、jQuery
- 组件类型：选择器表单项

## 函数签名

```
(option) => void;
```

- option：一个选项对象

## 使用方法

```
import React from 'react';

export default class Test extends React.PureComponent {
  constructor() {
    this.handleChange = this.handleChange.bind(this);
  }

  handleChange(event) {
    // 调用amis onToggle 方法，变更选择器表单项值
    const {onToggle, options} = this.props;
    const option = options.find(o => o.value === event.target.value);
    onToggle(option);
  }

  render() {
    // 获取表单项 value 和 options 属性
    const {value, options} = this.props;

    return (
      <select value={value} onChange={this.handleChange}>
        {options.map(option => (
          <option key={option.value} value={option.value}>
            {option.label}
          </option>
        ))}
      </select>
    );
  }
}
```

## 与 onChange 有什么不同

例如有如下 options：

```
{
  "options": [
    {
      "label": "Option A",
      "value": "a"
    },
    {
      "label": "Option B",
      "value": "b"
    },
    {
      "label": "Option B",
      "value": "b"
    }
  ]
}
```

- 例如使用 `onChange`，当第一次勾选了 `Option A` 后，调用 `onChange`，同步表单项 `value` 值为 `"a"`，当用户再次点击 `Option A`，该表单项的值仍然是 `"a"`，因为又一次重复设置了 `value`；
- 而使用 `onToggle`，当第一次勾选了 `Option A` 后，用户再次点击 `Option A`后，将会取消勾选 `Option A` 选项，`value` 值将为空字符串 `""`

在配置 `multiple:true`，选择器支持多选时：

第一次选中 `Option A`，`value` 值为 `"a"`，点击 `Option C` 时，`value` 会变成 `"a,c"`，再次点击 `Option C`，则 `value` 会变为 `"a"`

#### `onToggleAll`

切换全选和全不选

#### 适用范围

- 开发框架：React、Vue、jQuery
- 组件类型：选择器表单项
- 表达项 `schema` 需配置 `multiple: true`

#### 使用方法

```
import React from 'react';

export default class Test extends React.PureComponent {
  constructor() {
    this.handleChange = this.handleChange.bind(this);
    this.handleSelectAll = this.handleSelectAll.bind(this);
  }

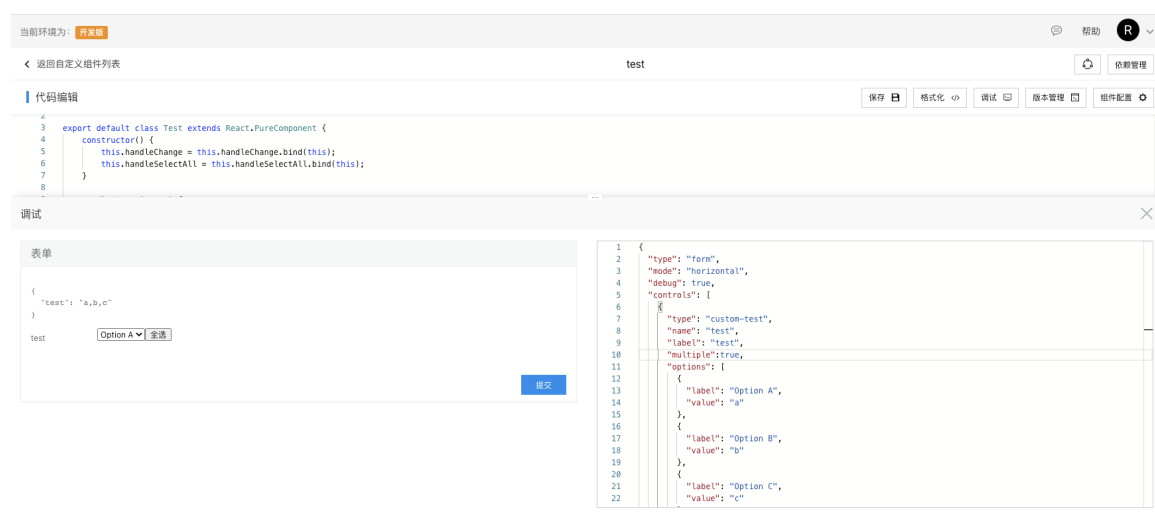
  handleChange(event) {
    // 调用amis onToggle 方法，变更选择器表单项值
    const {onToggle, options} = this.props;
    const option = options.find(o => o.value === event.target.value);
    onToggle(option);
  }

  handleSelectAll() {
    const {onToggleAll} = this.props;
    onToggleAll();
  }

  render() {
    // 获取表单项 value 和 options 属性
    const {value, options} = this.props;

    return (
      <>
        <select value={value} onChange={this.handleChange}>
          {options.map(option => (
            <option key={option.value} value={option.value}>
              {option.label}
            </option>
          ))}
        </select>
        <button onClick={this.handleSelectAll}>全选</button>
      </>
    );
  }
}
```

效果如下：



## 页面外部嵌入

此功能可以在爱速搭制作的页面，快速嵌入到已有业务系统中。采用的是 jssdk 模式，相比 iframe，有更好的用户体验，比

如弹框是直接弹出在整个页面而不是只能在 iframe 内部。此功能整合成本较低，适用于所有前端框架（包括微前端），引入一段 js 代码片段即可完成指定容器（div）的页面渲染。（如果是 React，还可以当成内部组件一样使用。）

js 代码由爱速搭自动生成的，能一键上传到您指定的对象存储（bos、s3 或者 oss）中，然后可以通过 cdn 地址引用。这样您的业务系统在运行时不依赖爱速搭平台渲染页面，稳定性能得到保障。（如果页面里面只是用您业务系统中的内部接口，这个时候运行时是完全脱离了对爱速搭平台的依赖。但是如果页面里面使用了 api 编排，或者数据模型相关接口，则还是会有依赖，不过内置了登录功能，依然可以联通）

### 简要说明

JSSDK 会生成如下代码，可以看出引入一段 js，然后执行一下 mount 即可完成页面渲染到指定 div 容器。

- script 代码

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>test</title>
  </head>
  <body>
    // 将『{{loderUrl}}』替换为导出的loader文件地址
    <script src="{{loderUrl}}"></script>

    <div id="root"></div>
    <
    <script>
      // 将『{{pageUrl}}』替换为导出的page文件地址
      amisLoader.load('{{pageUrl}}', page => {
        const data = {key: 'value'};
        page.mount('#root', page.schema, data);
      });
    </script>
  </body>
</html>
```

如果是嵌入到 React 框架中，则可以直接使用我们封装好的 React Component，然后当普通组件一样使用即可，如：

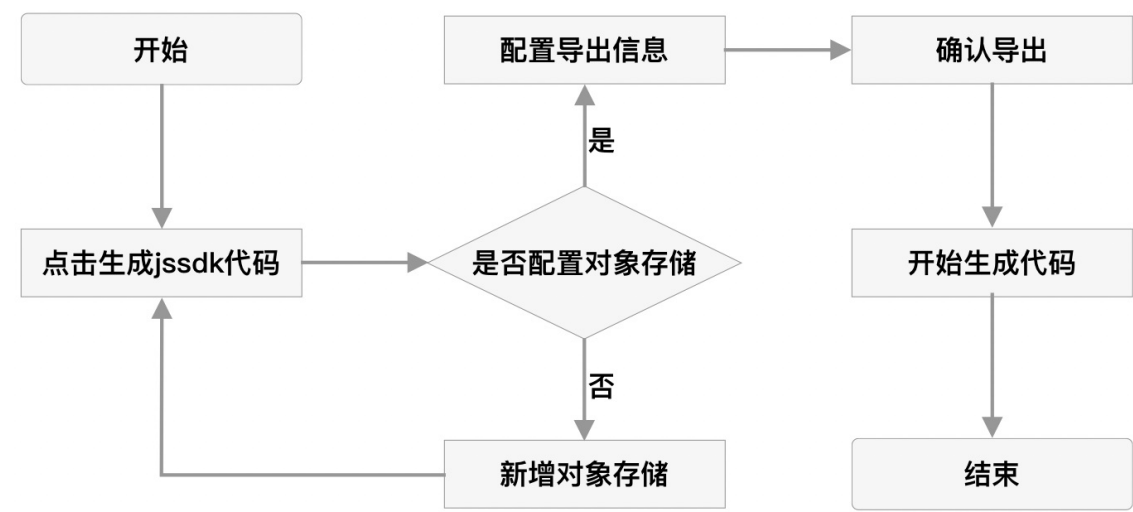
```
function myComponent() {
  const pageUrl = '';
  const data = {};

  return <AisudaPage pageUrl={pageUrl} data={data} />;
}
```

AisudaPage 内部会自动会加载所需要的资源。

### 功能使用详情

由于 jssdk 是自动上传到对象存储的，所以需要提前设置对象存储信息，大致流程如图。



功能使用详细介绍：

- 配置对象存储，配置入口如图所示：



- 功能入口：功能的入口在页面的右侧，如图所示：



- 点击导出后，您可以选择当前页面导出的环境类型，路由模式以及请求域名。
  - 环境类型：应用的环境类型，默认当前环境。可以选择其他环境，但需要确保当前页面已发布。
  - 路由模式：hash 或 browser 模式，属于浏览器的特性，详细信息可查询文档。
  - 请求域名：如果在 API 中心配置了请求，可以在这里配置请求域名，修改请求的转发地址，不配置默认爱速搭的域名。

页面成功导出后，会自动生成前端代码，包括 script 代码以及 react 组件示例

- script 代码

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>test</title>
  </head>
  <body>
    // 将『{{loaderUrl}}』替换为导出的loader文件地址
    <script src="{{loaderUrl}}"></script>

    <div id="root"></div>
    <
    <script>
      // 将『{{pageUrl}}』替换为导出的page文件地址
      amisLoader.load('{{pageUrl}}', page => {
        const data = {key: 'value'};
        page.mount('#root', page.schema, data);
      });
    </script>
  </body>
</html>

```

- react 公共组件代码

```

/**
 * @file common/aisudaPage.tsx
 */

import * as React from 'react';

interface AisudaPageProps {
  /**
   * 加载器 Url
   */
  loaderUrl: string;

  /** 页面 Url */
  pageUrl: string;

  data?: any;
}

export class AisudaPage extends React.Component<AisudaPageProps> {
  static defaultProps: Pick<AisudaPageProps, 'loaderUrl'> = {
    // 将『{{loaderUrl}}』替换为导出的loader文件地址
    loaderUrl: '{{loaderUrl}}'
  };

  rootDom = React.createRef<HTMLDivElement>();
  unmounted = false;
  currentPage: any = null;

  componentDidMount() {
    this.loadAMISLoader(this.props.loaderUrl, loader => {
      if (this.unmounted) {
        return;
      }

      loader.load(this.props.pageUrl, (page: any) => {

```

```
// 替换请求域名
const data = {
  AISUDA_HOST: 'http://localhost.aisuda.bce.baidu.com:8089',
  ...this.props.data
};
this.currentPage = page;
this.rootDom.current &&
!this.unmounted &&
page.mount(this.rootDom.current, page.schema, data);
});
});
}

componentDidUpdate() {
  this.currentPage?.updateProps({
    data: this.props.data
  });
}

componentWillUnmount() {
  this.unmounted = true;
  this.currentPage?.unmount();
}

loadAMISLoader(url: string, callback?: (loader: any) => void) {
  if ((window as any).amisLoader) {
    callback?.((window as any).amisLoader);
    return;
  }

  const script = document.createElement('script');
  script.setAttribute('type', 'text/javascript');
  script.onload = () => {
    callback?.((window as any).amisLoader);
  };
  script.setAttribute('src', url);

  document.body.appendChild(script);
}

render() {
  return <div ref={this.rootDom}>Loading...</div>;
}
}
```

- react 组件示例代码



```
/**
 * @file jssdkReactComp.tsx
 *
 */

import * as React from 'react';
import {AisudaPage} from '@common/aisudaPage';

export default function TestComp() {
  // 将『{{pageUrl}}』替换为导出的page文件地址
  const pageUrl = '{{pageUrl}}';
  const data = {customData: 'this is custom data'};

  return (
    <>
      <AisudaPage pageUrl={pageUrl} data={data} />
    </>
  );
}
```

- 其他框架使用

该功能提供了两个核心文件：loaderUrl 和 pageUrl，加载 loaderUrl 后会在全局挂载一个 amisLoader 对象，然后就可以通过 amisLoader.load 加载对应的页面。

## 使用文档

amisLoader.load 方法接受两个参数，参数说明如下：

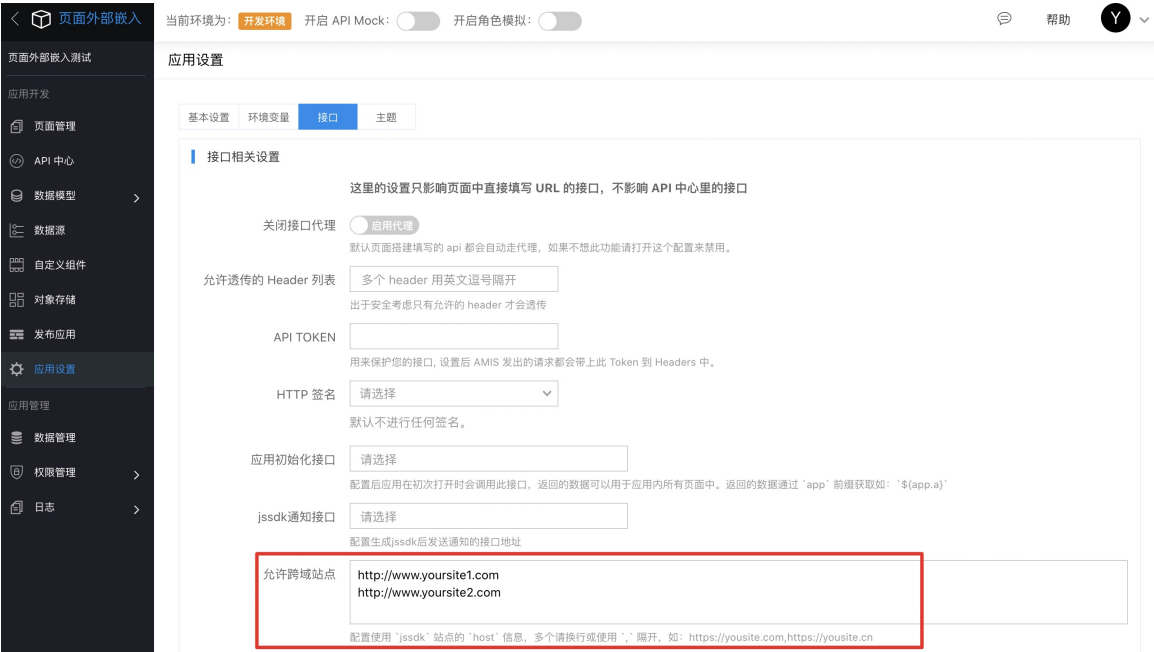
- pageSrc: 类型为 string，页面信息存储的地址，如：<http://bucket.endpoint/key.js>
- 回调函数：回调函数接受一个参数 page，类型为 PageProsp，具体内容见 PageProsp

PageProsp 中的参数以及方法说明如下：

- schema：页面的 schema 信息
- mount：挂载页面的方法，第一个参数可以是 ID 选择器，也可以是 dom 节点(ref)；第二个参数是页面的 schema，既 PageProsp.schema；第三个参数是页面需要的数据
- unmount：卸载页面的方法，第一个参数可以是 ID 选择器，也可以是 dom 节点(ref)。
- updateProps: 更新页面数据的方法。

## 其他说明

- 如果您使用了 API 中心的请求，且导出后请求域名为爱速搭平台，既爱速搭平台代理您的请求，需要进行以下设置：
  - 您的应用必须是 https，并且确保有 header X-Forwarded-Proto: https 透传到实例
  - 配置 docker env ISUDA\_COOKIE\_SAMESITE 为 none
- 应用设置/接口 中将使用 jssdk 的站点 host 信息填入到允许跨域站点列表中



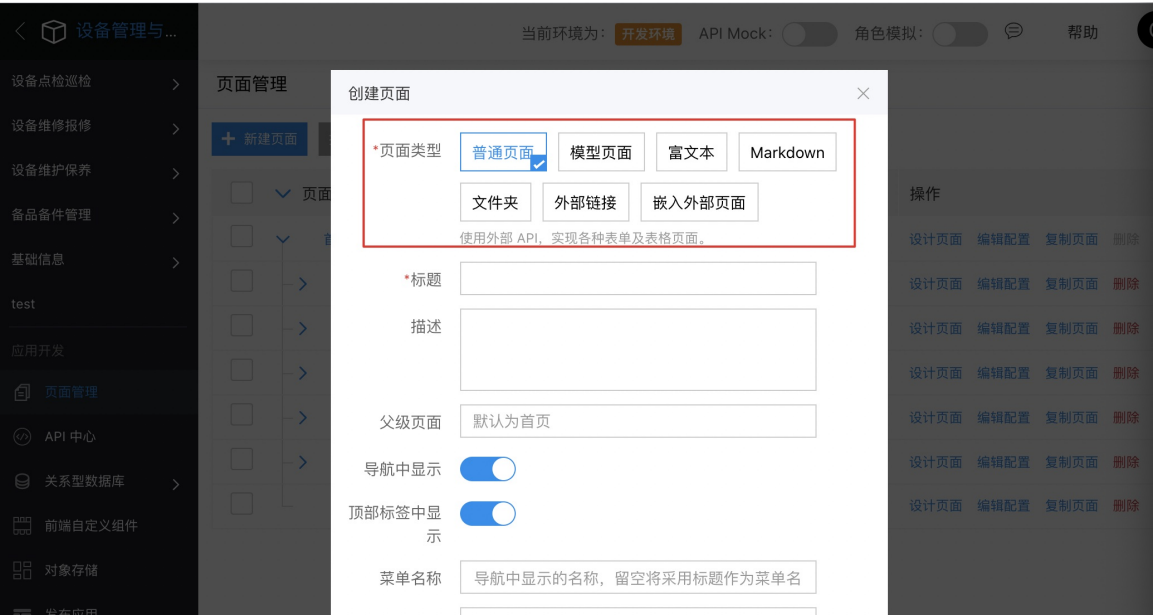
- 我们提供了一个通知功能，您可以在 API 中心新增一个接口，然后在 应用设置/接口 中配置 jssdk 通知接口，配置之后，当您导出页面之后，点击 发送消息通知，我们会将您当前导出的页面 pageurl 、 loaderUrl 等信息发送给您配置的接口。



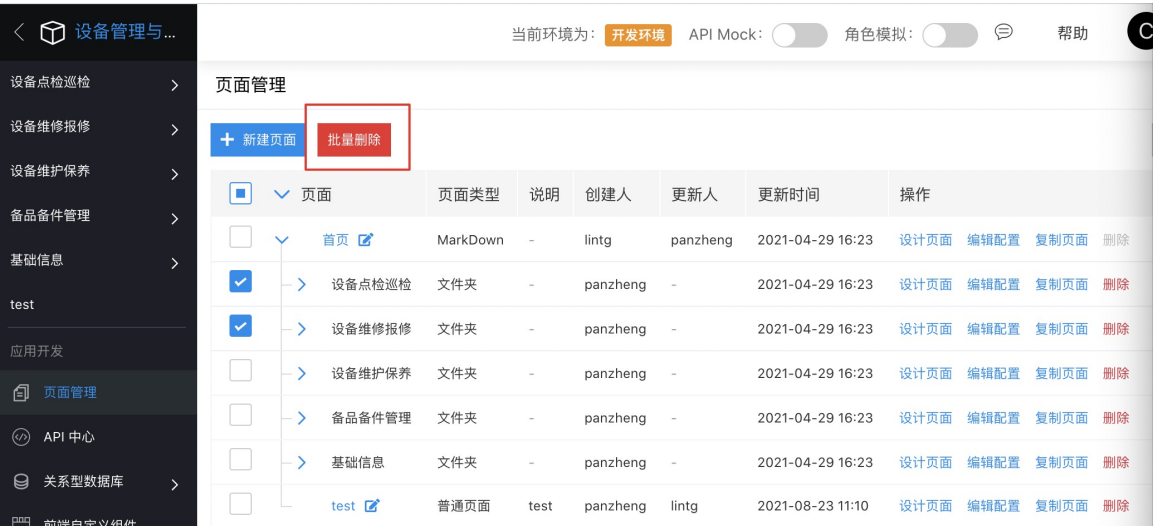
## 页面管理

爱速搭支持丰富的页面管理功能，支持页面便捷预览和编辑。平台支持 7 中页面类型：

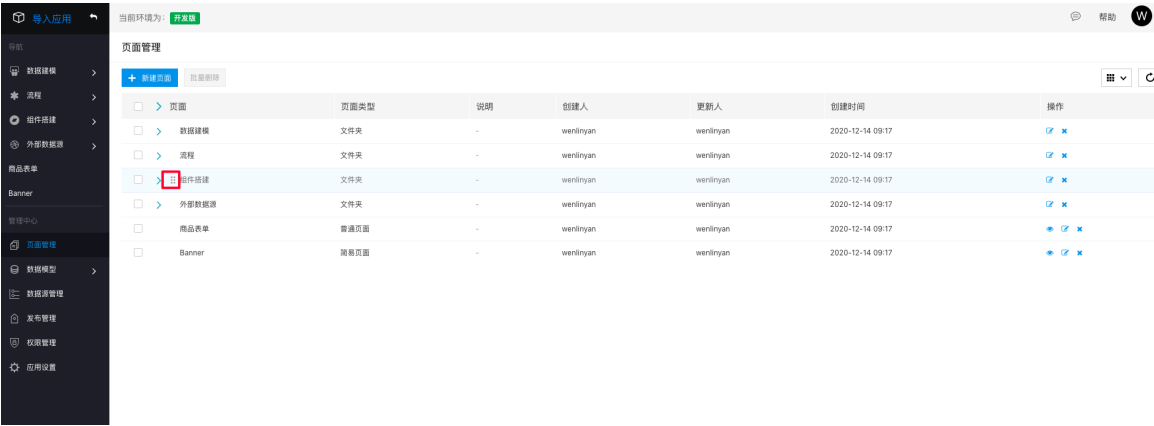
- 普通页面：支持自定义，拖拽组件完成配置，适合所有页面场景，同时支持自定义前端组件能力满足个性化需求
- 模型页面：数据模型一键生成，支持列表视图、新增视图、编辑石头、详情视图，每个页面可以单独配置，适合数据收集、数据展示等页面场景
- 富文本页面：内置富文本编辑，可快速编辑内容，适合内容展示页面
- Markdown 页面：内置 Markdown 编辑器，可快速编辑内容，适合内容展示页面
- 文件夹页面：没有实际的页面内容，在页面结构树中起占位作用。
- 外部链接：通过外链的方式放在导航中，点击导航时会跳转到外部页面。
- 嵌入外部页面：通过 iframe 的方式将外部页面集成到平台中，点击导航时会直接显示这个外部页面，用于将之前页面集成进来。



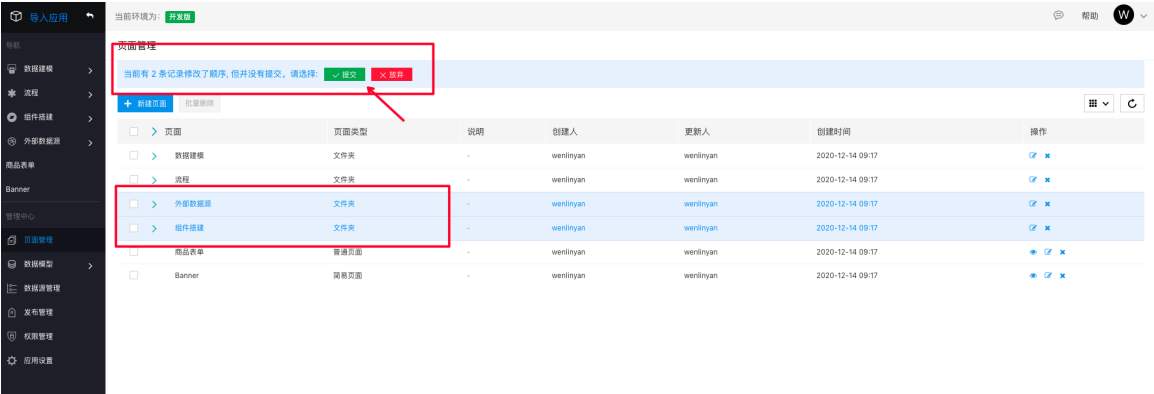
如果页面需要删除，不仅支持单个页面删除，也支持批量删除。



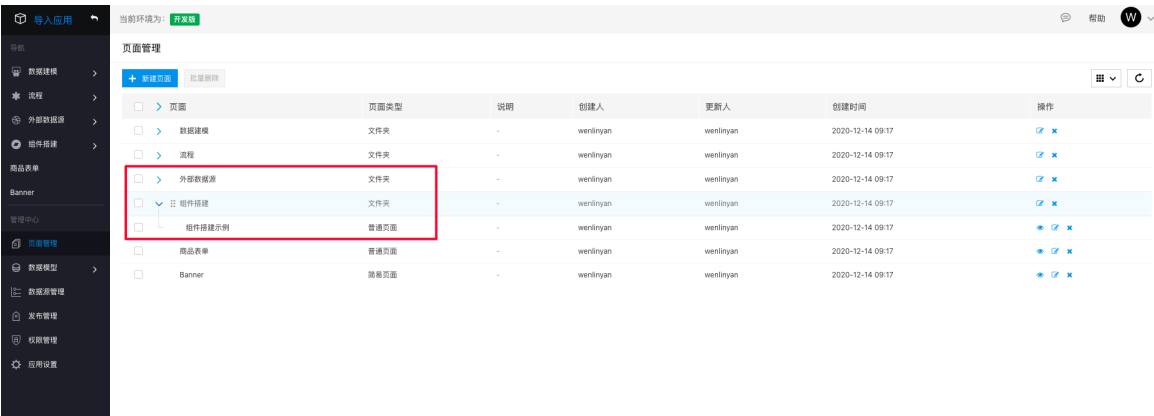
如果需要调整页面结构，可以直拖拽完成。首先鼠标悬浮在要移动的页面上，找到并长按位置拖拽 icon。



然后进行操作确认，完成拖拽操作。



如果对包含子页面的页面进行结构调整，子页面也会一起被调整。

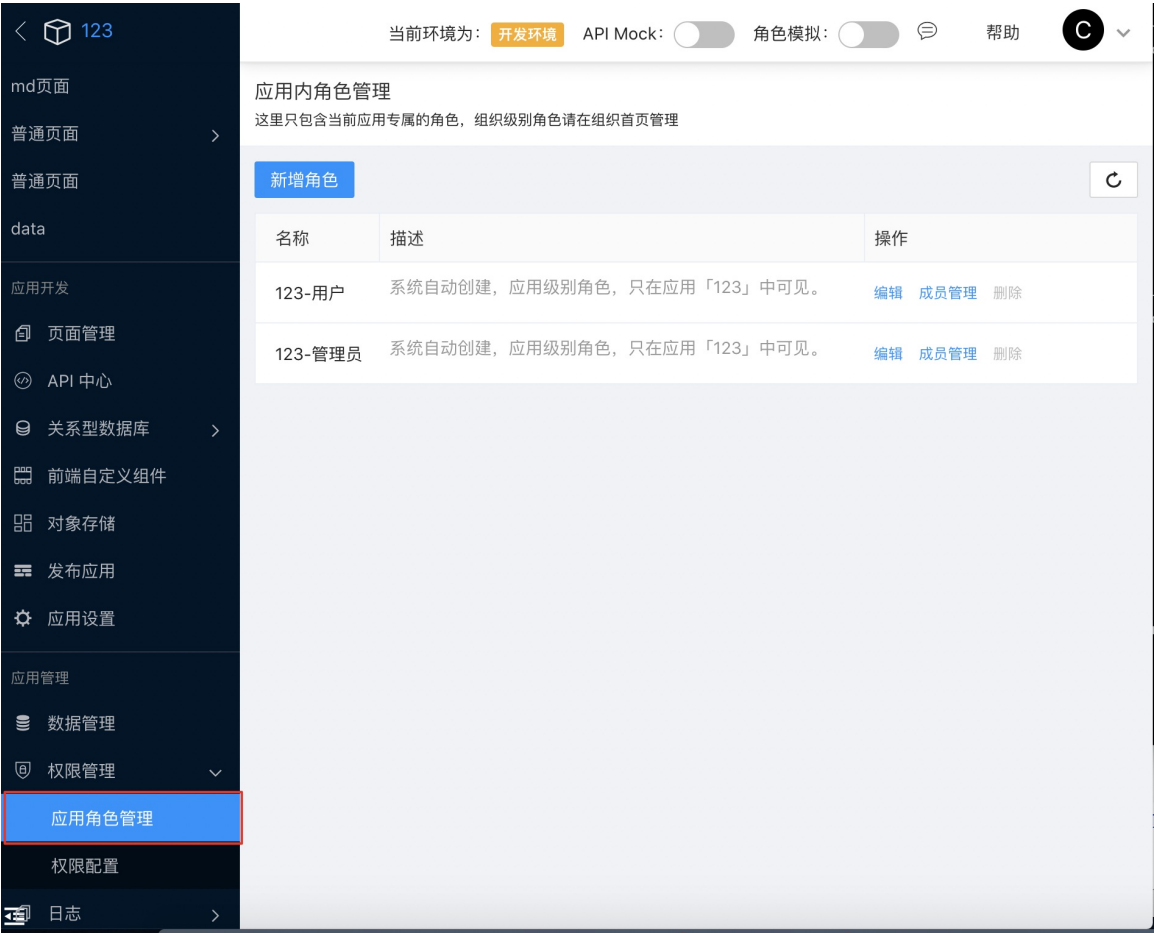


## 应用管理

### 应用权限设置

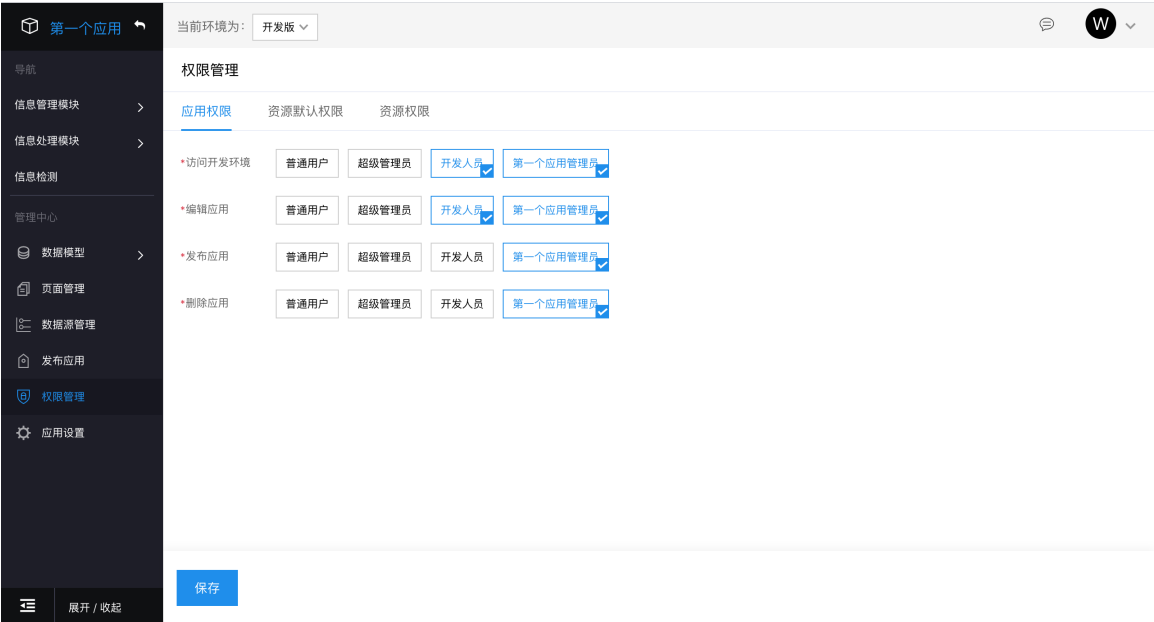
应用权限管理模块可以快速实现应用权限设置和资源权限设置。

**应用角色管理** 管理应用内专属角色，非系统级角色



应用基本权限

应用权限设置可对应用级操作权限的默认值进行修改，修改设置项后点击【保存】即可保存修改内容。



资源默认权限

资源默认权限是指应用内的各种资源的初始权限配置，可以根据业务需求进行修改。目前资源类型有页面资源、数据模型资源和数据源资源。



资源权限

资源权限是指应用中的角色与资源操作权限的对应关系，可对应用内的所有资源设置操作权限。

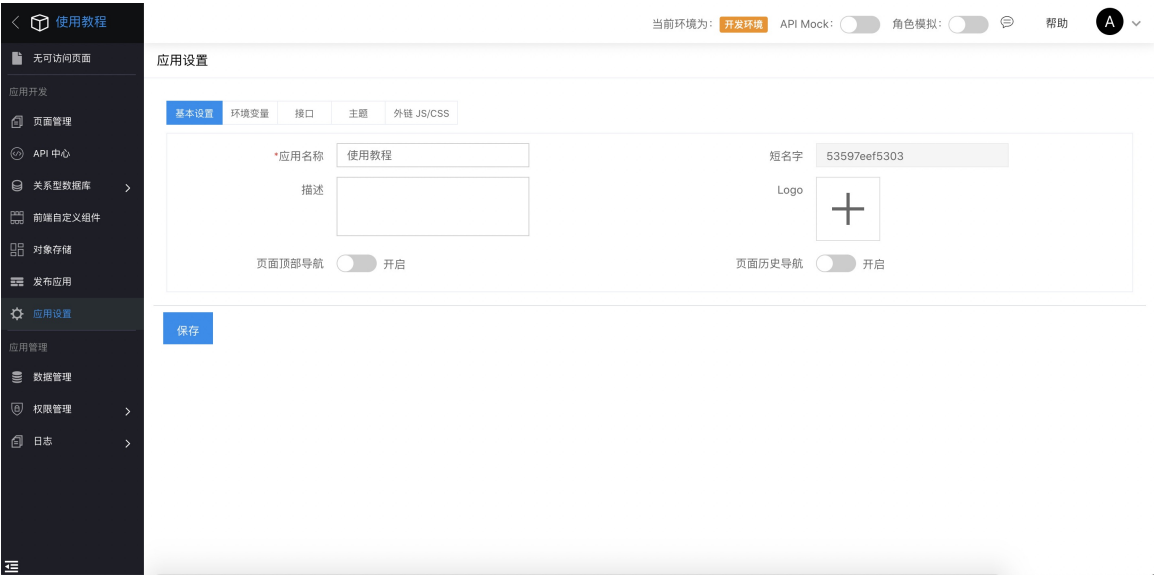


应用设置

应用的相关设置可以在【应用设置】模块进行完善和修改，包括基本配置、接口相关设置及环境变量配置。接口相关配置可以配置默认发送格式、API TOKEN 以及适配器等。

基本配置

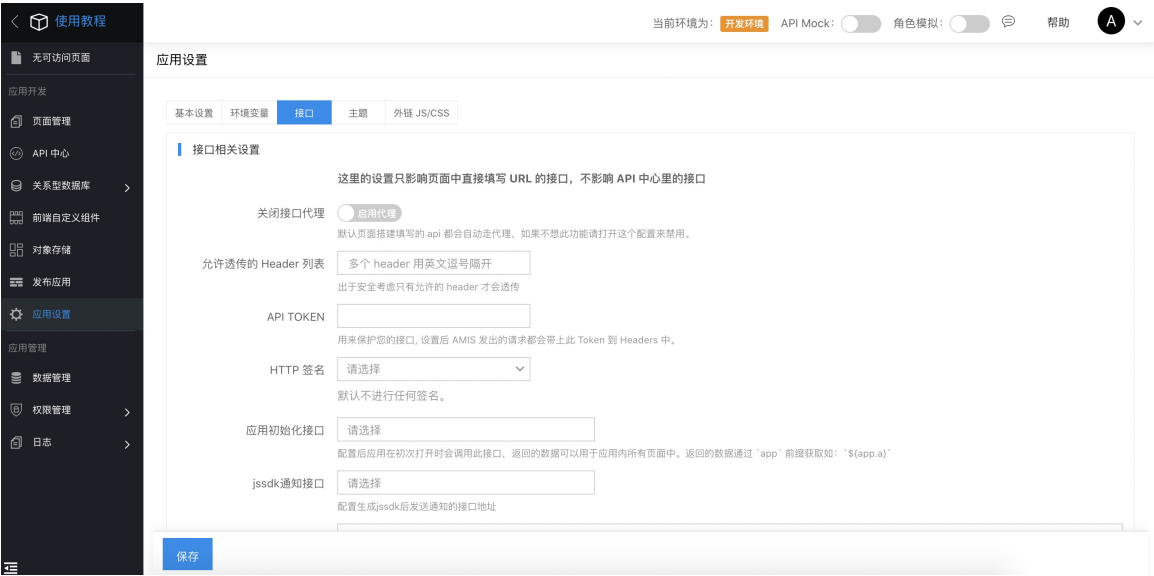
应用的基本配置支持应用名称、短名字、描述以及 logo 等信息的管理。



环境变量是方便管理不同环境下应用的一些资源。

接口相关设置

接口相关设置主要是对应用中针对接口有关的功能设置。



默认发送格式

- JSON: 发送数据体格式为 application/json
- Form: 发送数据体格式为 application/x-www-form-urlencoded
- FormData: 发送数据体格式为 multipart/form-data

当发送内容中有文件时，会自动使用 multipart/form-data 格式

API TOKEN

用来保护开发接口, 设置后爱速搭发出的请求都会带上此 Token 到 Headers 中。

HTTP 签名

选择要使用的 HTTP 签名，目前支持两种。

## HTTP 签名

请选择



HTTP基本认证

百度云iam签名 (v1)

### 全局发送适配器

发送适配器是指在接口请求前，对请求进行一些自定义处理，例如修改发送数据体、添加请求头、等等，而全局适配器将应用到该群组中所有的接口上。

设置后所有的请求都会经过此适配器，接口本身可以单独设置适配器，此适配器会在接口本身发送适配器之前执行。

函数签名：(api) => api，数据在 api.data 中，修改后返回 api 对象。

```
// 进行一些操作
let data = api.data;
data.foo = 'bar'; // // 新添加数据
console.log(api); // 可以打日志，打开浏览器开发者工具查看日志
// 将修改后的 api return 出去
return {
  ...api,
  data: data
}
```

上例中，我们获取暴露的 api.data 参数，并且为其添加了一个新的字段 foo，并且一起返回出去就可以了，这样我们的请求数据体中就会加上我们这个新的字段。

api 中一般会包含下面几个变量：

- url：当前接口 Api 地址
- method：当前请求的方式
- data：请求的数据体
- headers：请求的头部信息

操作符为 es6 扩展操作符，js 相关知识请自行学习

### 全局接收适配器

如果后端返回跟平台要求的数据格式不一致时,可以通过配置自定义 js function 转换。

函数签名：(data, response, api) => data 修改完后返回 data

参数说明：

- data: 接口返回的数据体
- response: 接口返回的对象，包含状态值等。
- api: 接口信息，包含发送地址、方式、数据体。

示例：

例如：我们接口返回的 data 直接是一个数组，而我们的 crud 组件，需要包装一个 items 字段，才可以正常显示数据，那么可



以：

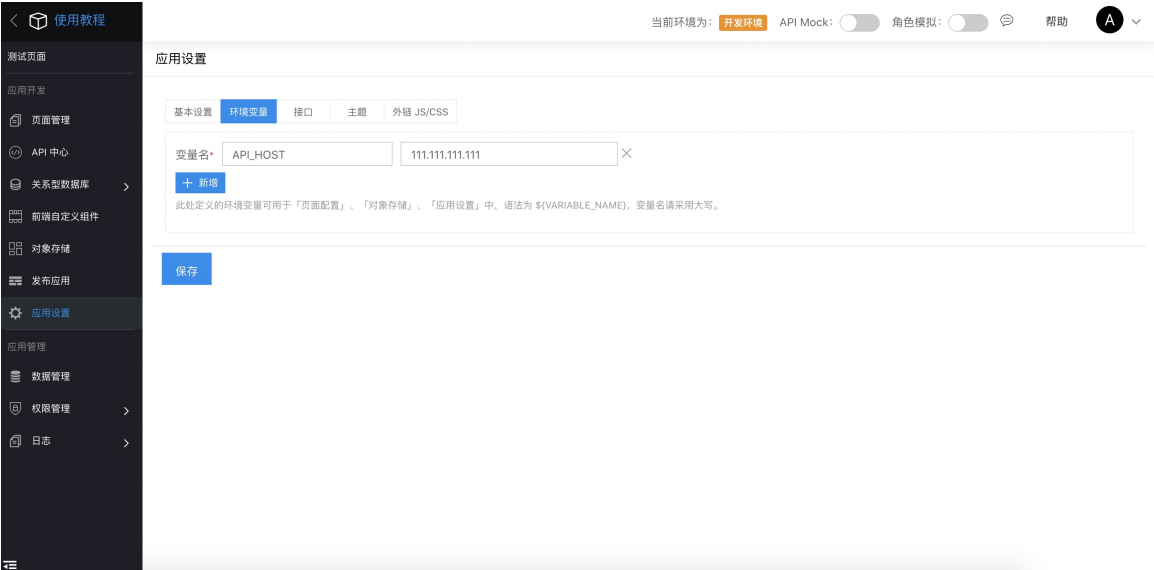
```
// 进行一些操作
const items = payload.data;
console.log(items); // 可以打日志，打开浏览器开发者工具查看日志
// 将修改后的 data return 出去
return {
  ...payload,
  data: {
    items: items
  }
}
```

我们获取 payload.data，并且将该值赋给 items 字段，并且放到要 return 出去的 data 中，payload 其他参数原样返回，这样就可以实现上述效果

设置环境变量

通过 key-value 的形式设置环境变量

开发环境



其他环境

发布应用时，可以不同环境设置不同的环境变量值，这样发布之后，就可以针对不同环境，使用不同的资源。

发布版本

版本号

1.0.0

@

当前开发版

选择当前开发版本，或者选择已发布过的版本。

\*标题

版本标题

描述

简单描述，比如更新日志。

类型

正式环境

沙盒

测试环境

不同的类型，数据独立。

\*访问权限

普通用户

超级管理员

开发人员

使用教程管理员

测试

环境变量

API\_HOST

222.222.222.222

发布镜像

关

发布成 Docker 镜像，可以将此镜像进行私有化部署。

取消

确认

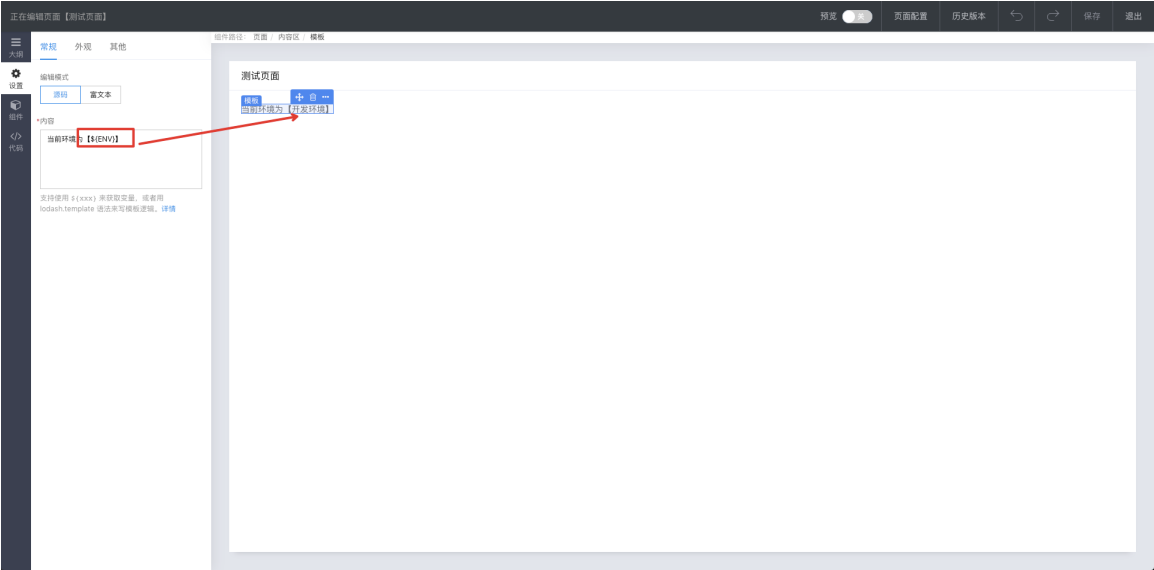
上例中，我们本地使用API\_HOST为111.111.111.111，我们线上环境使用API\_HOST时就是222.222.222.222

使用环境变量

可以在应用中使用环境变量

在页面配置中

在组件搭建页面中，可以通过\${xxx}获取当前环境变量，例如：



上图中我们在页面直接显示一段文字，并使用\${xxx}获取ENV变量。然后我们发布时，更改这个变量

发布版本

版本号

1.0.1

@

当前开发版

选择当前开发版本，或者选择已发布过的版本。

\*标题

版本标题

描述

简单描述，比如更新日志。

类型

正式环境

沙盒

测试环境

不同的类型，数据独立。

\*访问权限

普通用户

超级管理员

开发人员

测试公司管理员

环境变量

ENV

正式环境

发布镜像

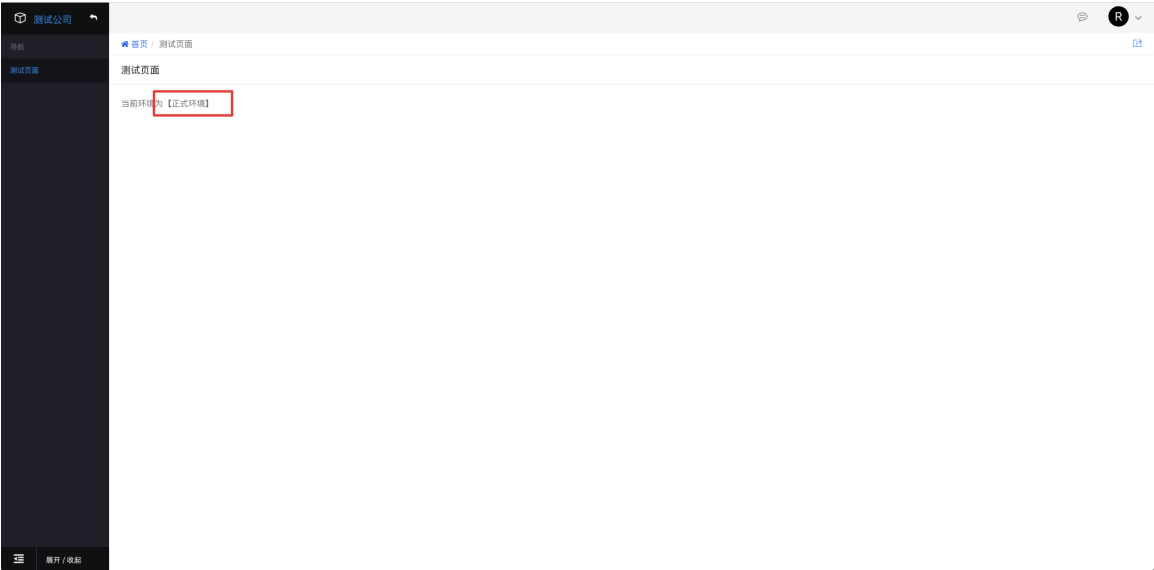
关

发布成 Docker 镜像，可以将此镜像进行私有化部署。

取消

确认

上图中我们将正式环境中的ENV变量为正式环境。发布完打开页面我们可以看到文字已经变为"当前环境为【正式环境】"



你几乎可以在任何地方使用\${xxx}来获取，你也可以在配置api接口的地方使用环境变量：



创建数据源中

创建外部数据源也支持填写环境变量，实现不同环境使用不同数据库资源地址的能力。

我们只以数据库地址为例：

首先先添加环境变量 `DB_HOST`：

新增「数据源」

1 数据源连接 > 2 映射配置 > 3 完成 >

\*数据源名称

请输入数据源名称

\*数据库类型

MySQL

\*数据库地址

`\${DB\_HOST}`

支持使用环境变量，使用语法为`\${VARIABLE\_NAME}`

\*端口

3306

支持使用环境变量，使用语法为`\${VARIABLE\_NAME}`

\*数据库

请输入数据库 或 环境变量

支持使用环境变量，使用语法为`\${VARIABLE\_NAME}`

\*用户名

请输入用户名 或 环境变量

支持使用环境变量，使用语法为`\${VARIABLE\_NAME}`

\*密码

请输入密码 或 环境变量

支持使用环境变量，使用语法为`\${VARIABLE\_NAME}`

自动同步

开

是否自动同步已映射的数据源模型。如果关闭则为手动同步，建议开启。

上一步

测试连通性并下一步

然后在创建数据源的时候，数据库地址填入`\${DB\_HOST}`

119

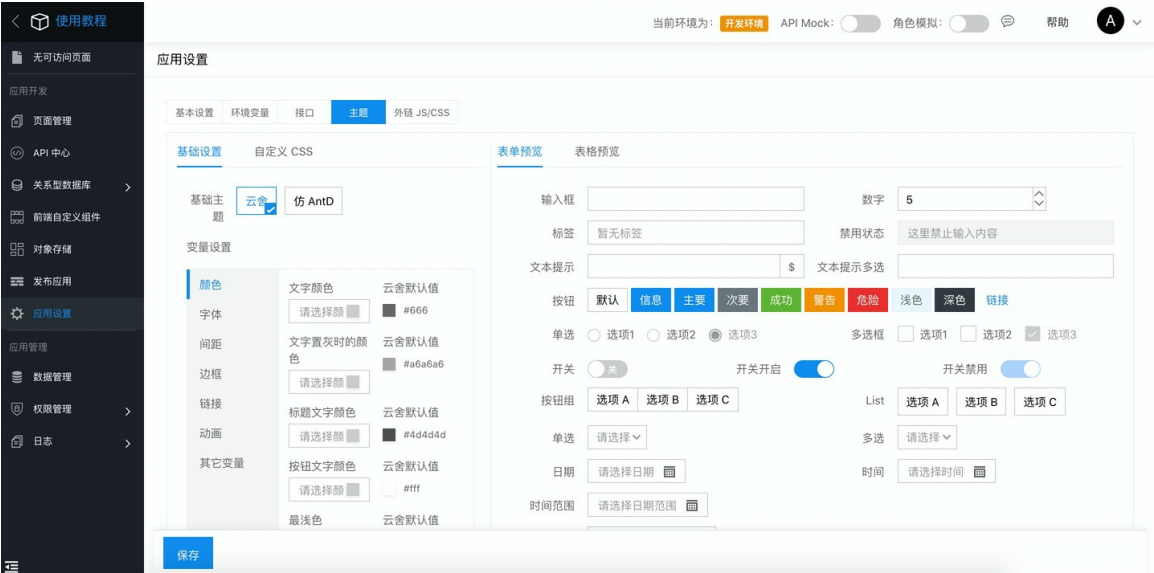


这样点击下一步测试联动性的时候会获取 111.111.111.111 作为数据库地址进行联通

请保证不同资源地址的外部数据库结构保持一致，否则可能会有问题。

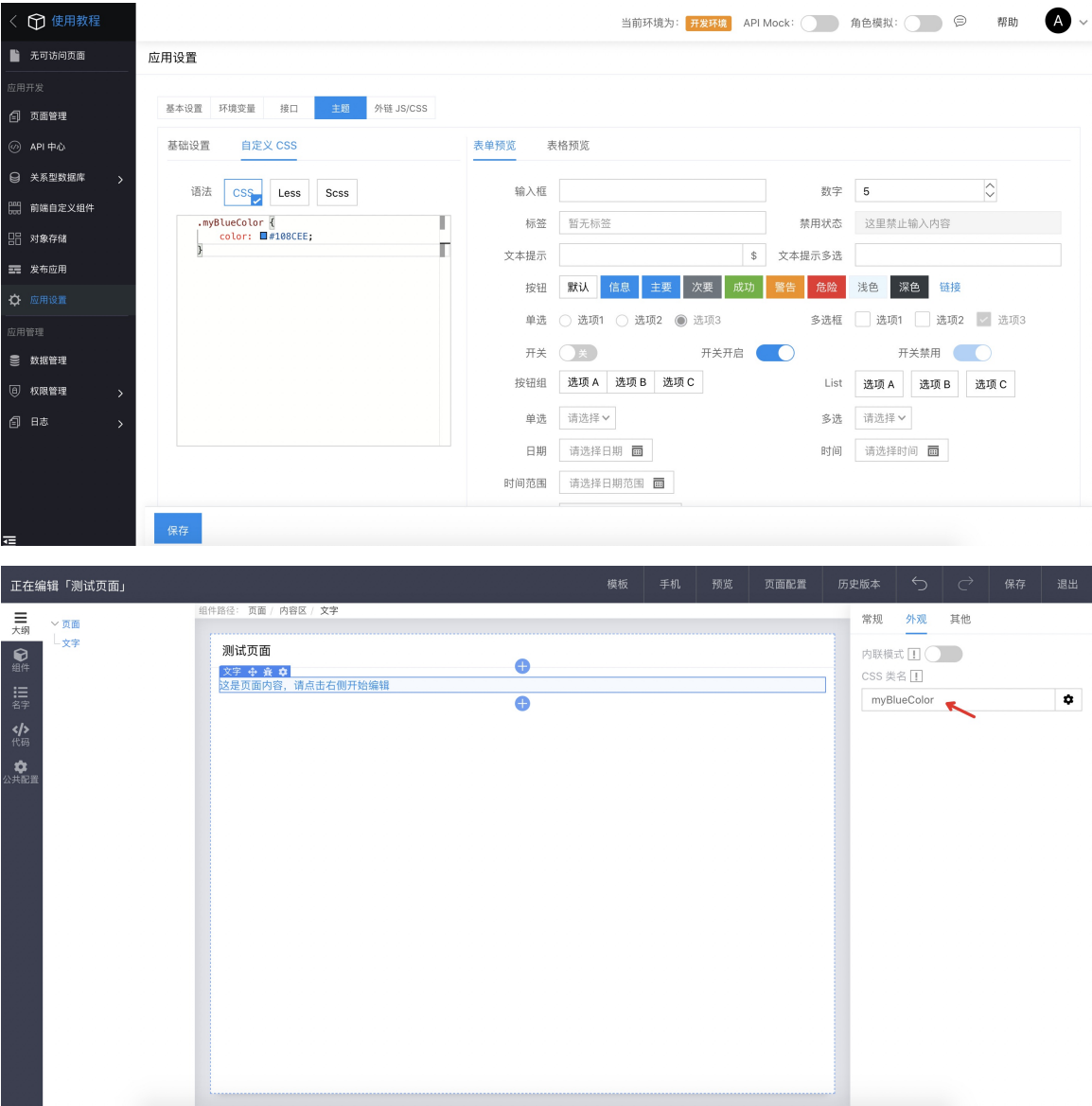
主题设置

主题设置主要是对应用主题风格的设置，其中包括基础设置和自定义CSS。



**基础设置** 基础设置主要是对基础主题和变量的设置，主题支持云舍和仿 AntD，变量包括颜色、字体、间距、边框、链接、动画、其他变量。

**自定义CSS** 自定义CSS主要应用于页面元素的样式配置，例如：在这里定义CSS样式的类名，在页面可视化配置中就可以引用这个样式类名。



**外链 JS/CSS** 这个配置主要是为嵌入 Angular 组件提供的外部 JS 和 CSS 链接地址配置，具体可以参考页面设计章节。

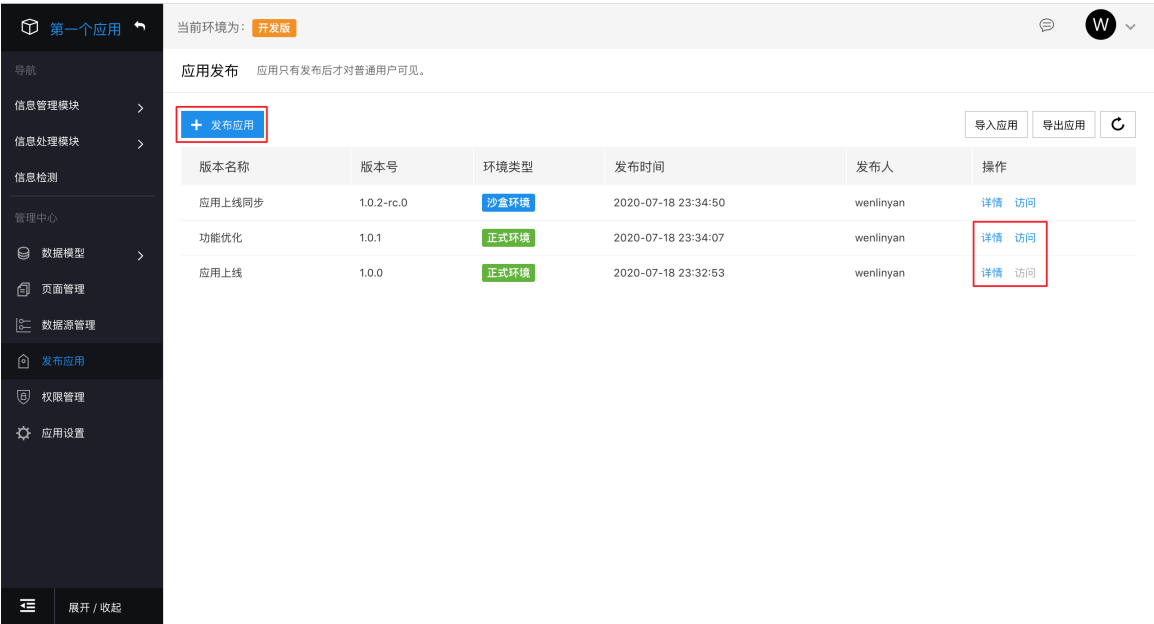


应用发布

应用可以通过发布完成上线流程，爱速搭支持强大的版本管理能力，可以轻松实现版本迭代和发布控制。一键发布及完成上线，自带版本发布详情，可快速溯源。如果线上版本可快速回滚。

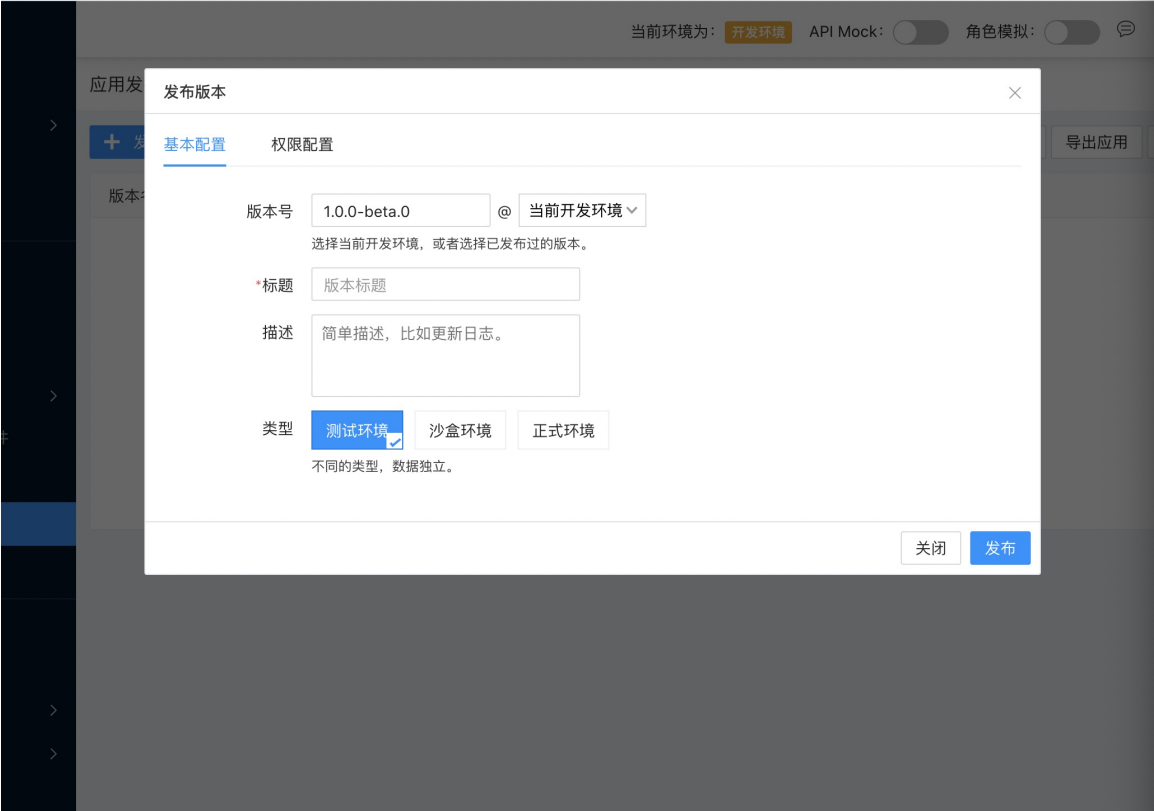
应用发布

编辑完成的应用可以通过【发布应用】进行发布，已发布版本可以快速访问和查看详情，版本发布后不支持版本信息修改。

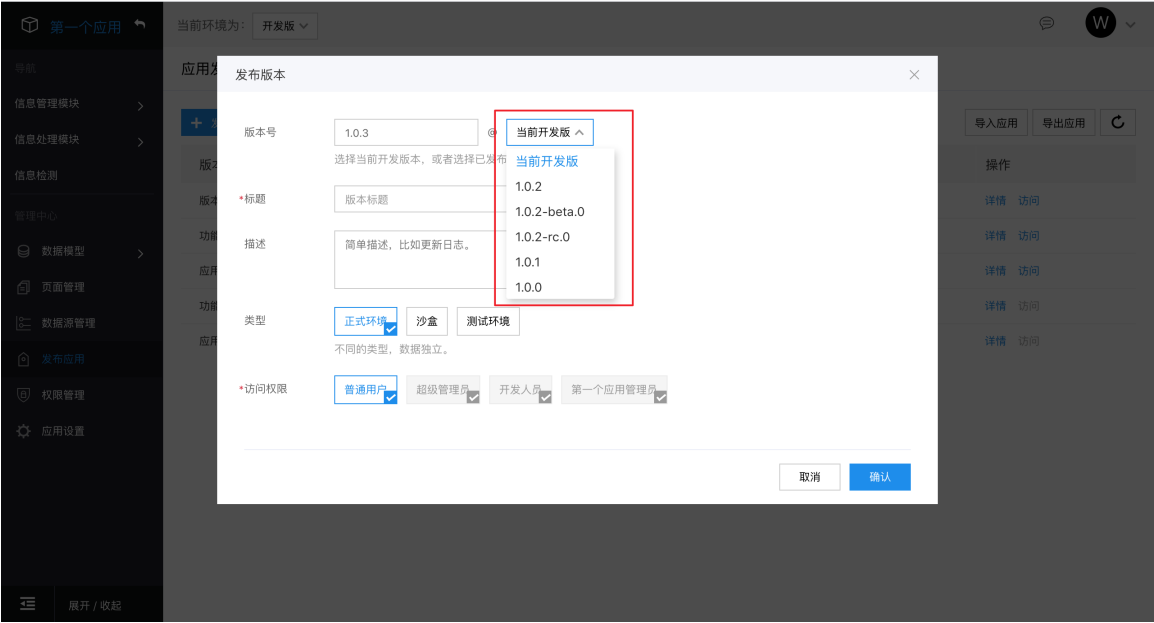


应用发布时可以配置环境、权限等，支持线上版本、沙盒版本及测试版本，并可以轻松实现版本控制。

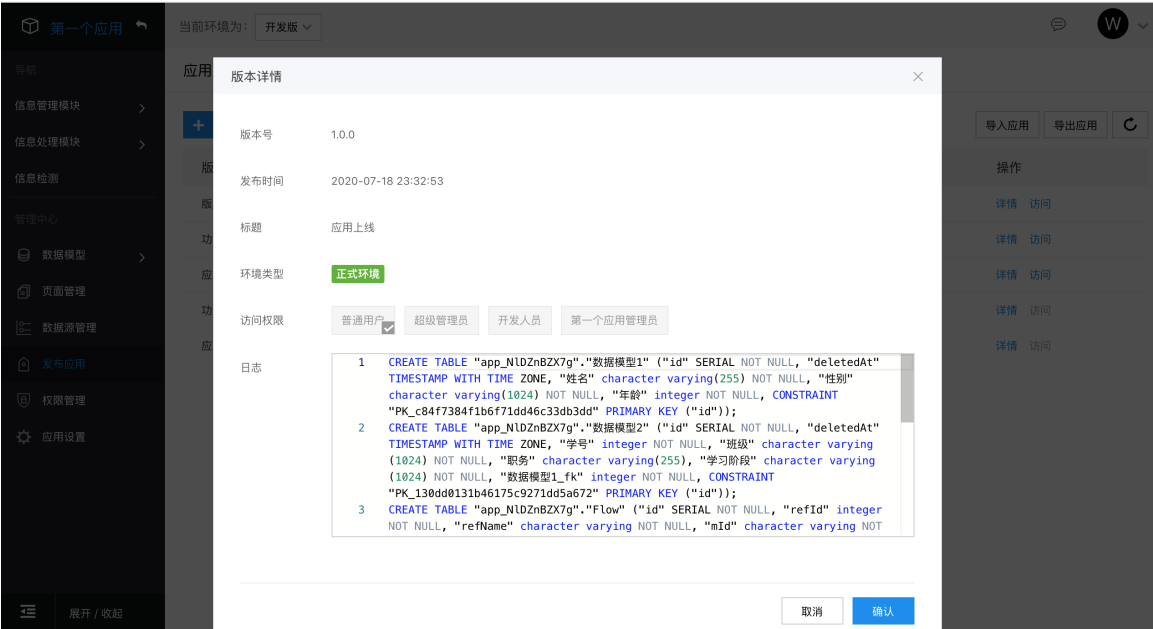




平台支持版本回滚，只需选择要回滚的版本和发布环境重新发布。



版本详情中可以查看本次版本的基本信息和变更日志。



应用版本权限管理 包括访问权限，管理权限，数据管理和资源权限。



环境类型

爱速搭平台支持了线上环境、沙盒环境和测试环境，方便开发过程中的测试和上线。



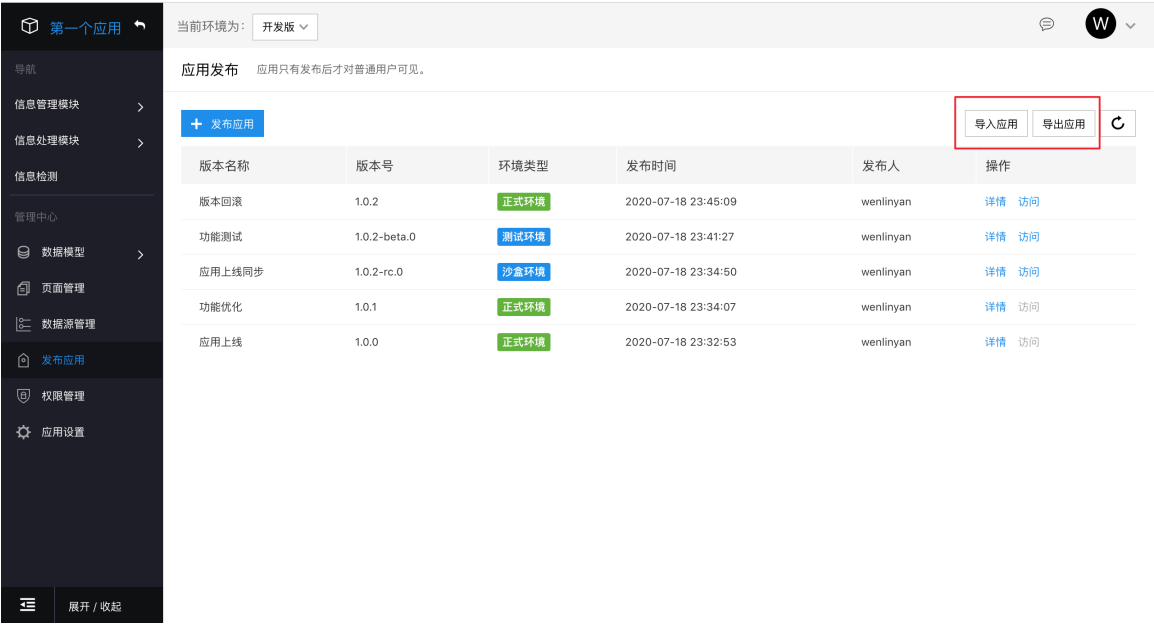
发版时可选择任意环境下的已发布版本或当前开发版，不同环境类型下的数据独立。

🔗 版本回滚

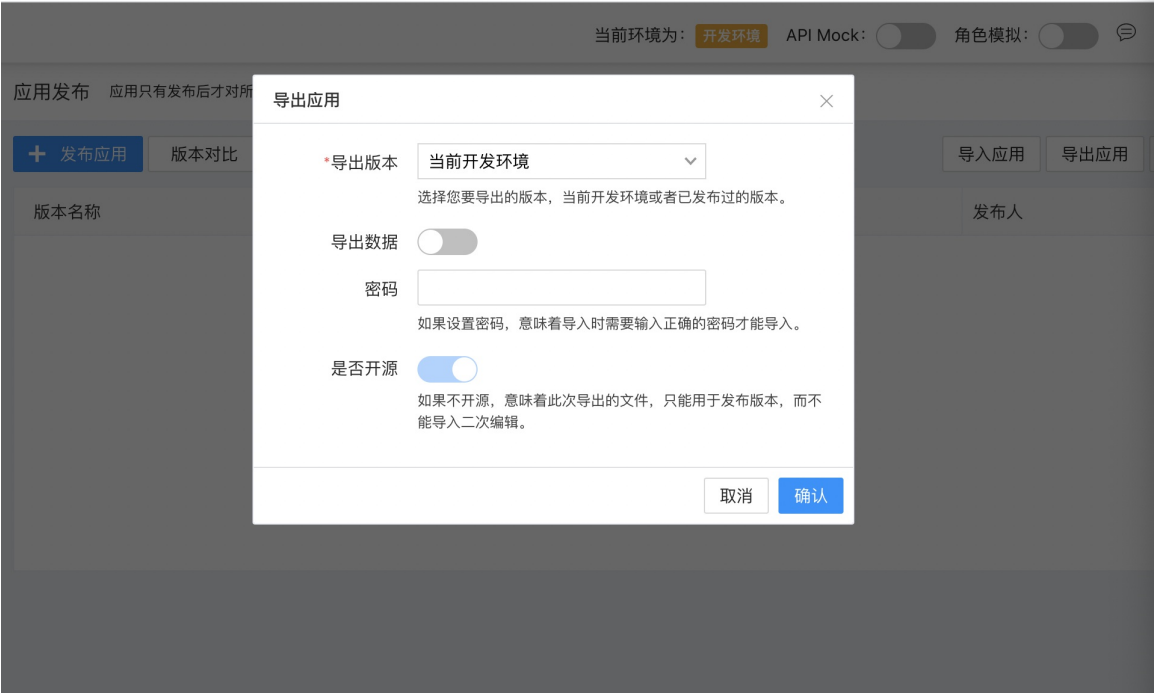
版本回滚的状态非常简单，选择正确的要回滚的版本，基于选中的版本重新发布一个版本即可。

🔗 导出应用

为了方便应用生成模板和在公司间的复用，平台支持应用导入导出功能。



应用导出支持选择要操作的版本，需要设置密码以保证应用导出后的安全性，并且支持控制导出的应用支持的二次操作权限。



🔗 导入应用

在组织管理部分有介绍到，应用管理模块也可以实现应用导入。在应用发布模块同样支持应用导入，导入的应用包在当前操作的应用上生效。

应用导入需要上传应用文件并输入密码才能实现应用的安装，需要注意的是，目前仅支持在不包含自建资源的应用中执行导入应用的操作。



## 组件模板导入

为了方便您快速使用，我们准备了常见的组件模板文件。您可以通过以下步骤自行下载并操作导入

1. 下载文件

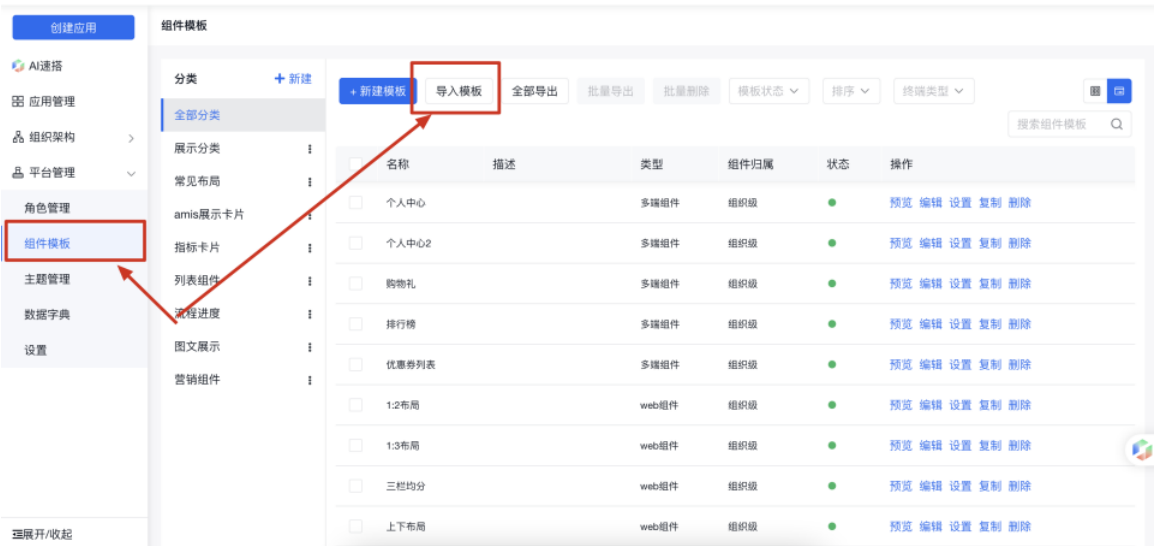
点击下方链接下载文件至您的电脑

[amis组件模板文件包](#)

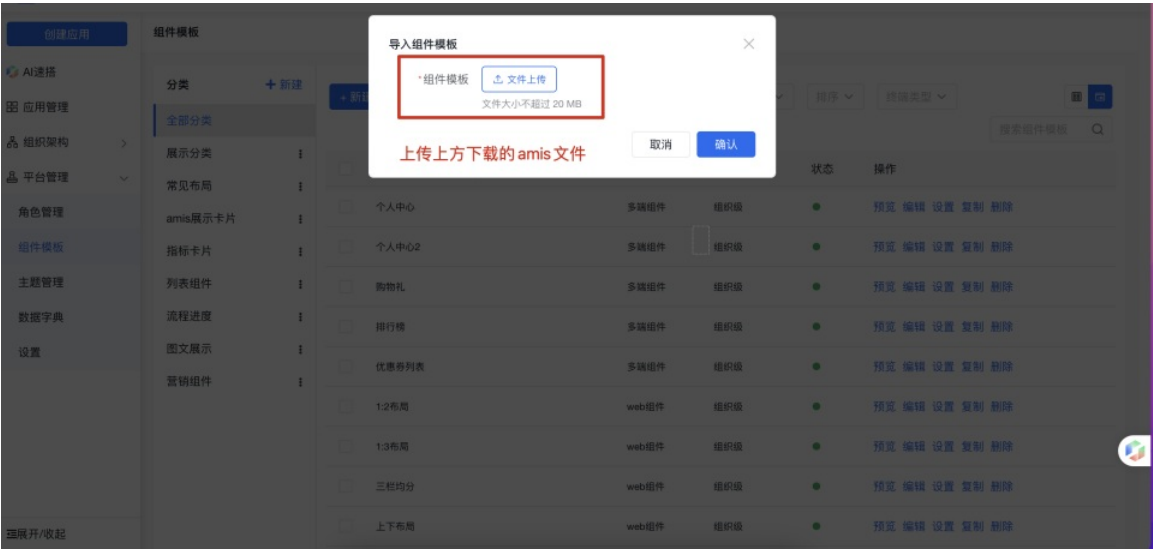
2. 导入指引

下载完成后，请参考我们的简易操作指南完成导入

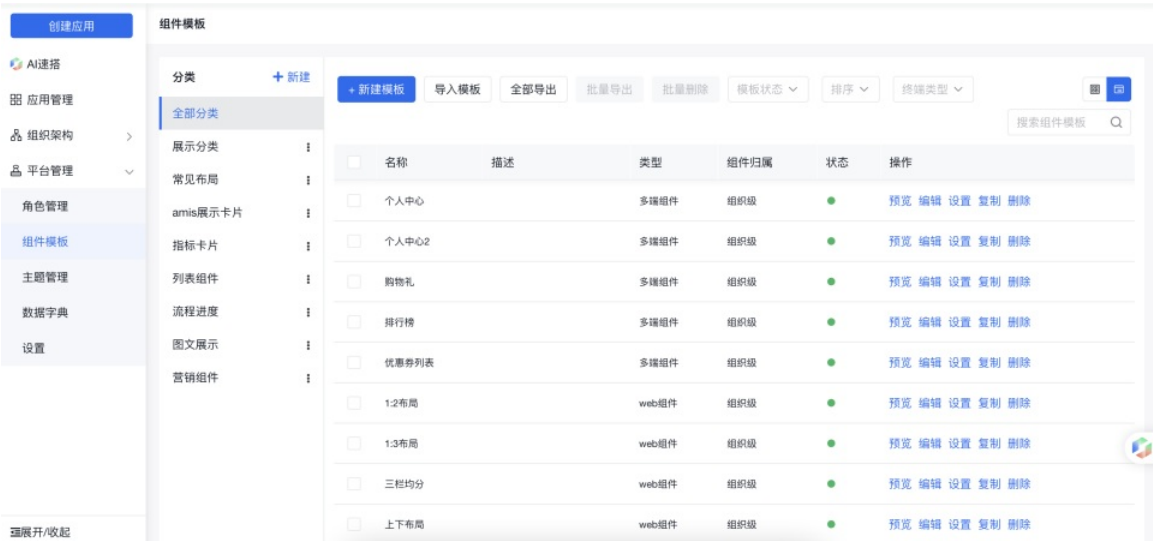
- 进入到组织选择组件模板，选择导入模板



- 上传刚刚下载的文件，点击确认



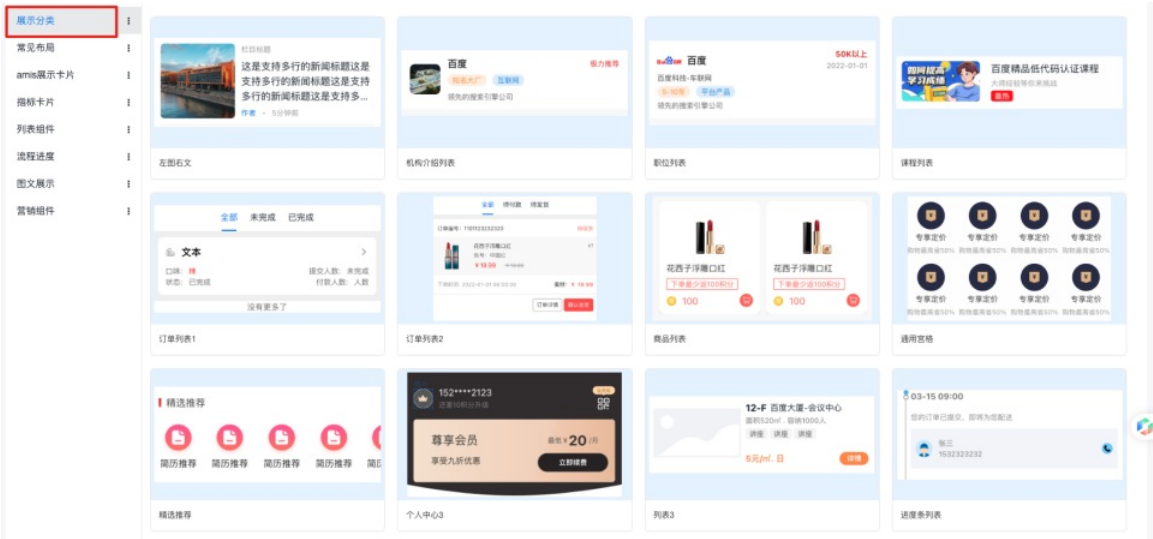
- 页面提示导入成功后，文件内容会变成一个组件。此时可以对组件进行预览、编辑等操作，并可以用于补充各类页面设计中的组件物料。

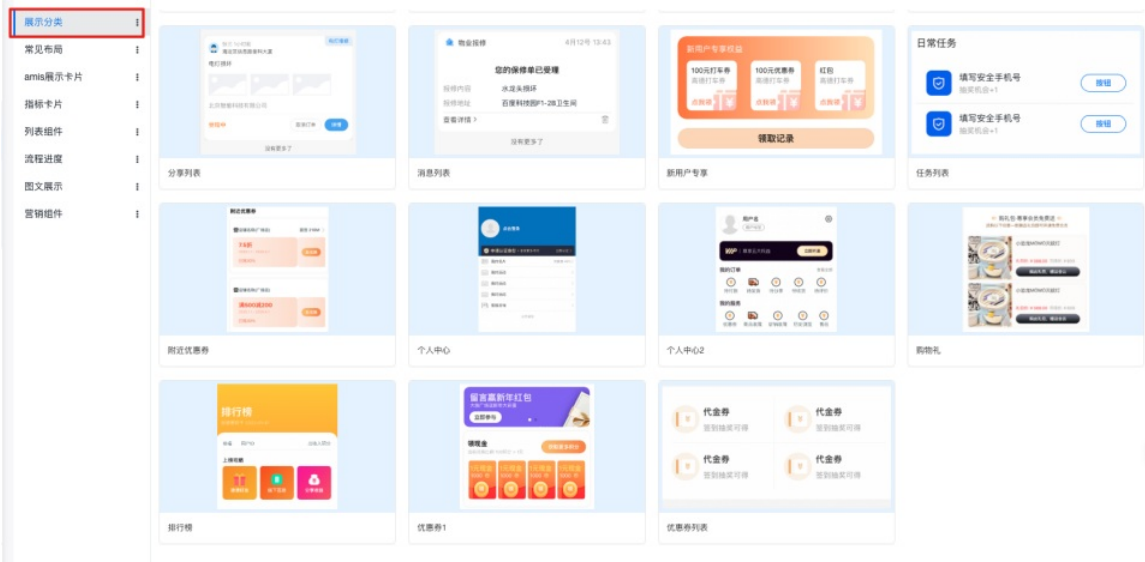


3. 组件包内模板类型

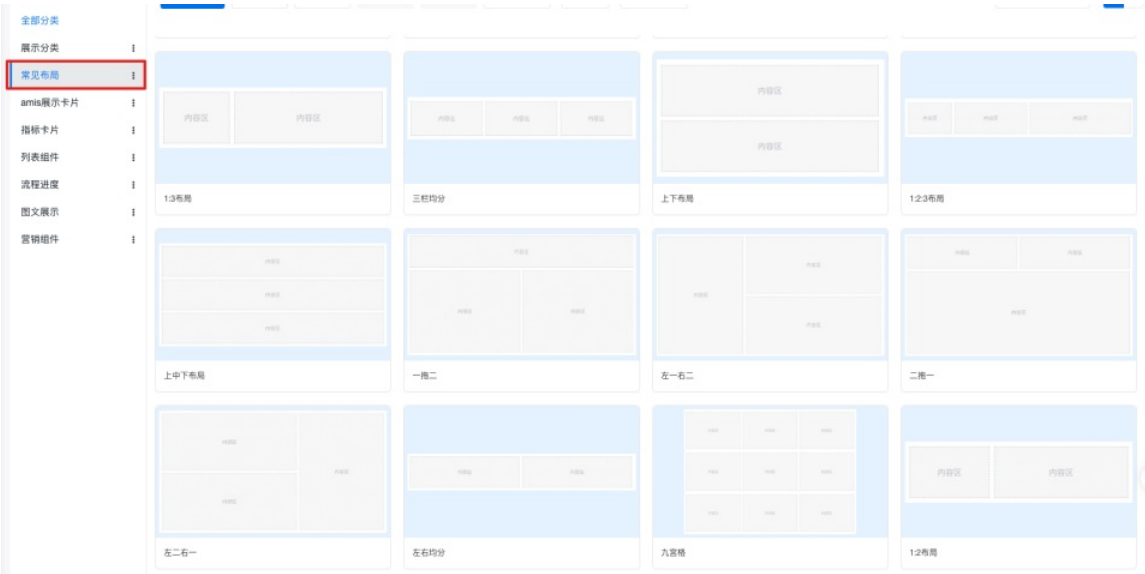
- 以下是组件包内的模板类型和相关样式。

展示分类

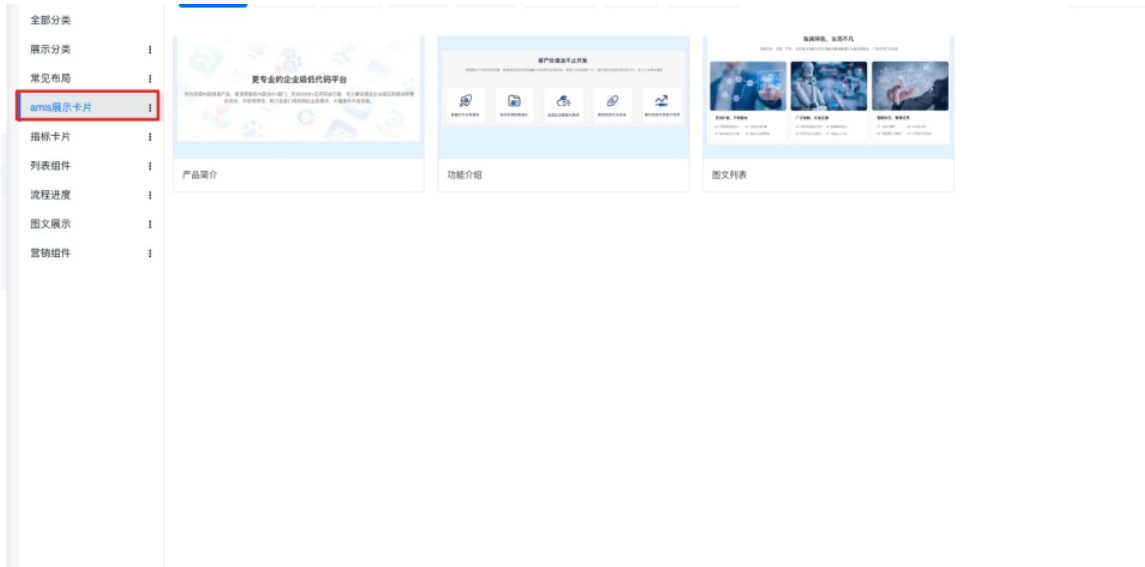




常见布局



amis展示卡片



指标卡片

展示分类

常见布局

amis展示卡片

指标卡片

列表组件

流程进度

图文展示

营销组件

展示分类

常见布局

amis展示卡片

指标卡片

列表组件

流程进度

图文展示

营销组件

全部分类

展示分类

常见布局

amis展示卡片

指标卡片

列表组件

流程进度

图文展示

营销组件

\$ 45,021

总收入

展示卡片

总收入

\$ 26,35,262

上周业绩

指数卡片1

公司业绩

12088 万

营业收入

+12.1%

指数卡片2

总订单量

12088

增长+12.1%

指数卡片3

总订单量

12088

近一个月增长情况

+20%

指数卡片4

总订单量

12088

+12.1% 增长

指数卡片5

总订单量

12088

查看详情

指数卡片6

总订单量

12088

+12.1%

指数卡片7

总订单量

12088

+12.1%

指数卡片8

总订单量

12088

+12.1%

指数卡片9

产品名称

负责人: 张三

+12.1%

指数卡片10

278

QA总数

指数卡片11

峰值带宽

20 GB

较昨日↓ 18%

指数卡片12

花西子浮耀口红

下单量: 100

商品列表

12-F 百度大厦-会议中心

面积1000㎡, 容纳1000人

演讲: 演讲: 演讲

5元/㎡, 日

列表3

03-15 09:00

您的订单已提交, 即将为您配送

张三

1532232323

进度条列表

花西子浮耀口红

下单量: 100

分享列表

您的保单已受理

保单内容: 水险火险

保单地址: 西康科技园P-28J室

查看详情?

消息列表

点击跳转组件模板编辑页

新闻标题这是支持多行的新闻标题这是支持多行的新闻标题这是支持多...

作者: 5分钟前

左图右文

百度

强力推荐

机构介绍列表

百度

50K以上

2022-01-01

职位列表

全部 未完成 已完成

文本

状态: 已完成

提交人数: 未提交

付款人数: 未付款

订单列表1

订单编号: 100123232323

订单金额: ¥100.00

订单时间: 2022-01-01 10:00:00

实际: ¥100.00

订单列表2

产品名称

产品型号

¥201.00

列表组件1

会员权益

会员权益: 会员权益

会员权益: 会员权益

卡片列表1

订单列表1

订单列表2

列表组件1

卡片列表1

数据看板

数据看板: 数据看板

数据看板: 数据看板

卡片列表2

百度自动驾驶出行服务平台

快速正式落地程度超预期

文章列表2

15

12:00 am - 03:00 pm

与客户在会议室开会

事件列表

项目进度

项目进度: 项目进度

项目进度: 项目进度

卡片列表3

机房设备名称

3200

安全防护指数: 800

安全防护指数: 800

卡片列表4

表格

表格: 表格

表格: 表格

table风格列表

会员权益

会员权益: 会员权益

会员权益: 会员权益

卡片列表5

列表组

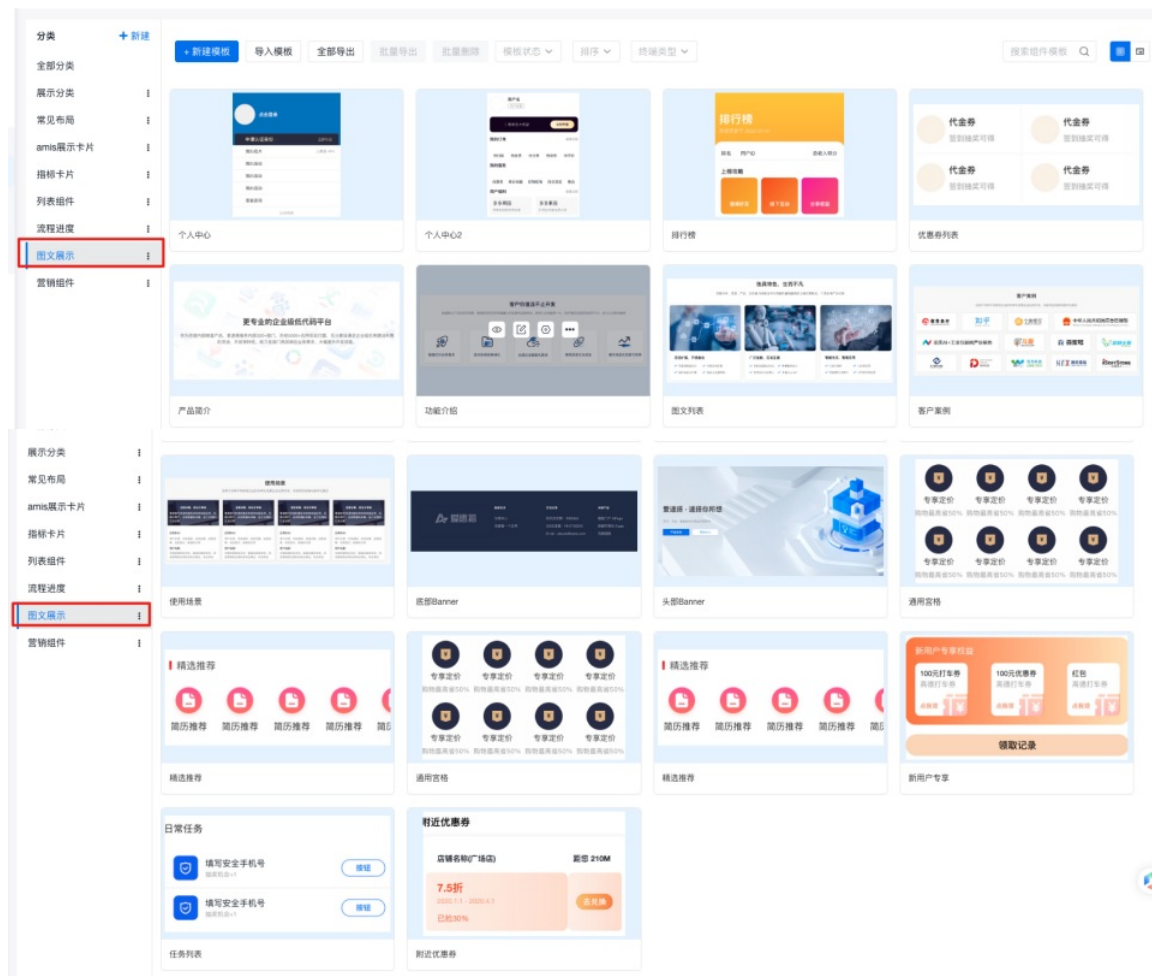
件

流程进度





### 图文展示



营销组件



## 数据模型

## 外部数据源接入

### 外部数据源IP白名单



为了数据库安全，建议设置 IP 白名单来只允许爱速搭访问，爱速搭出口 IP 是 14.215.188.0/25 和 111.45.0.0/22，如果您的云服务厂商不支持这种 CIDR 设置方法，可以将以下所有 IP 加入白名单中：

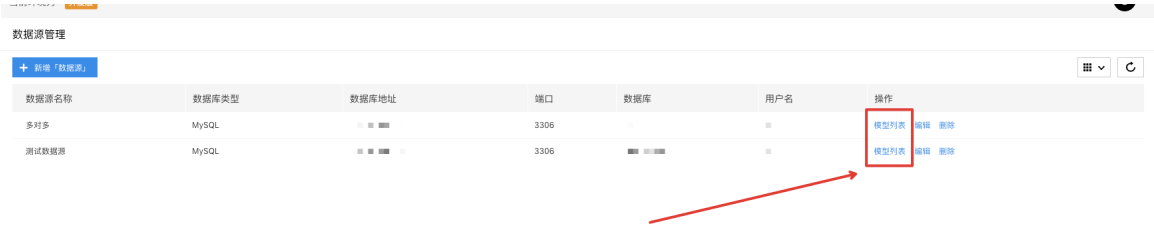
```
14.215.188.4
14.215.188.5
14.215.188.6
14.215.188.7
14.215.188.8
14.215.188.9
14.215.188.10
14.215.188.11
14.215.188.12
14.215.188.13
14.215.188.14
14.215.188.15
14.215.188.16
14.215.188.17
14.215.188.18
14.215.188.19
14.215.188.68
14.215.188.69
14.215.188.70
14.215.188.71
14.215.188.72
14.215.188.73
14.215.188.74
14.215.188.75
14.215.188.76
14.215.188.77
14.215.188.78
14.215.188.79
14.215.188.80
14.215.188.81
14.215.188.82
14.215.188.83
111.45.1.9
111.45.1.137
111.45.1.201
111.45.2.9
111.45.2.137
111.45.2.201
111.45.1.73
111.45.2.73
```

外部数据源使用

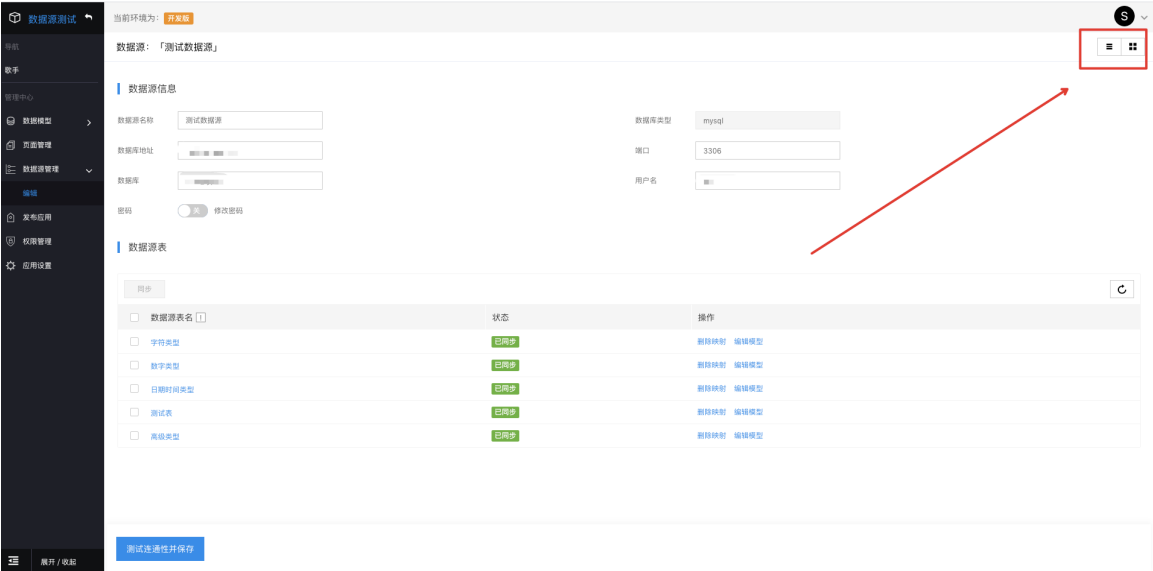
数据源的使用主要是通过数据源模型

查看数据源模型

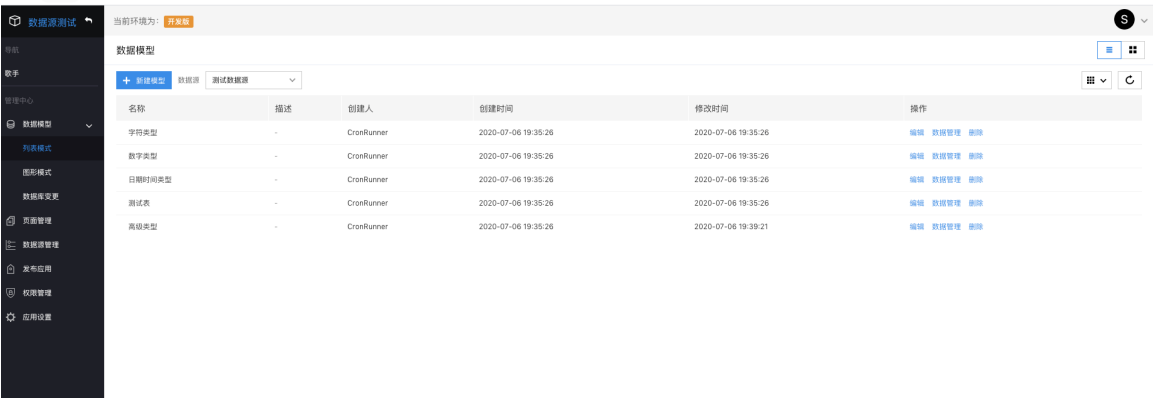
可以在数据源列表操作列中，点击模型列表，查看已映射的数据源模型列表



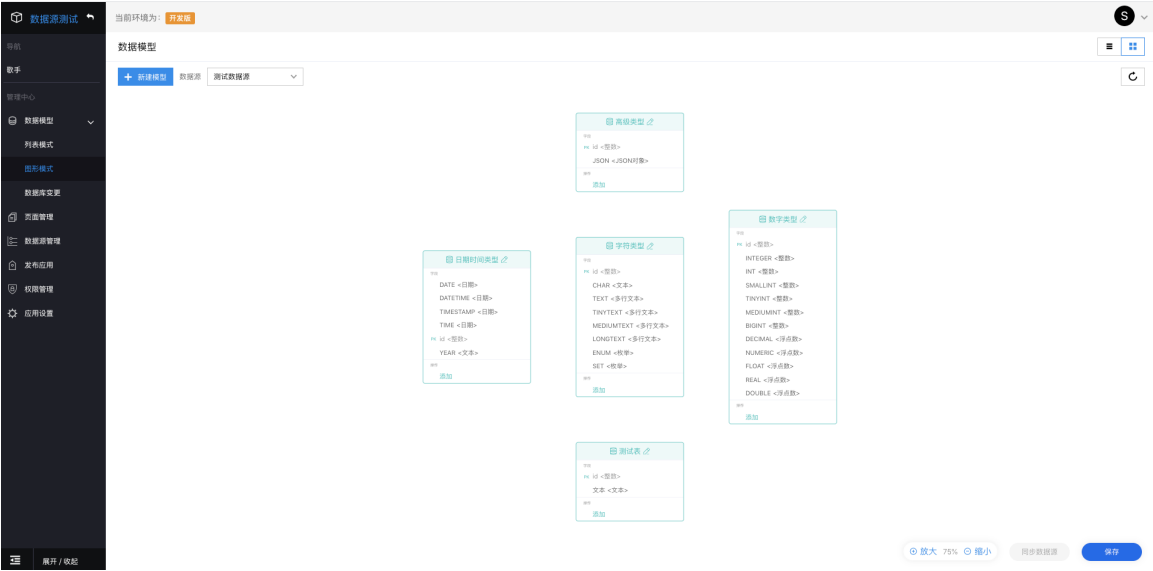
也可以在数据源编辑界面，点击右上角图标，查看数据源模型列表和图表



数据源模型列表

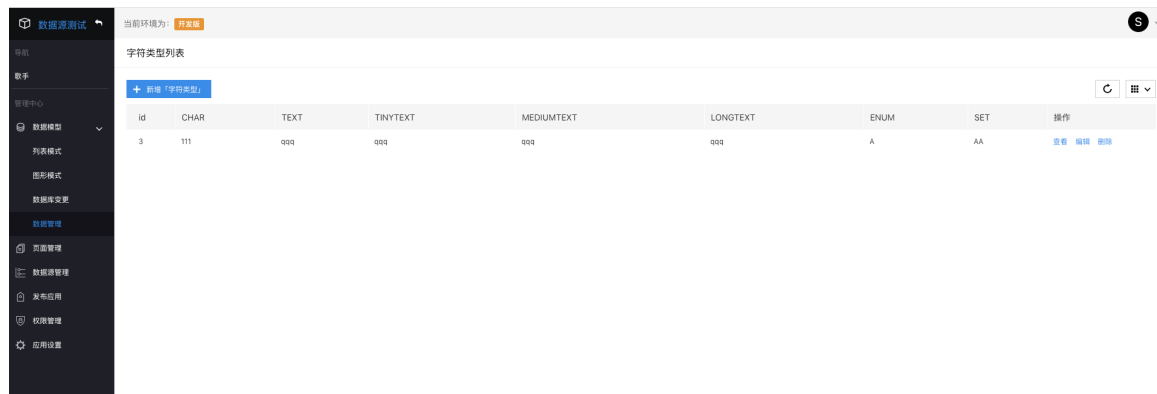


数据源模型图表



数据管理

在数据源模型列表操作列中，点击数据管理，可以快速对数据源数据进行增删改查操作。

[查看数据](#)

### 新增数据

新增「字符类型」

CHAR

TEXT

TINYTEXT

MEDIUMTEXT

LONGTEXT

ENUM

请选择

SET

请选择

取消

确认

修改数据

编辑字符类型数据「3」

CHAR

111

TEXT

qqq

TINYTEXT

qqq

MEDIUMTEXT

qqq

LONGTEXT

qqq

ENUM

A

SET

AA

取消

确认

删除数据

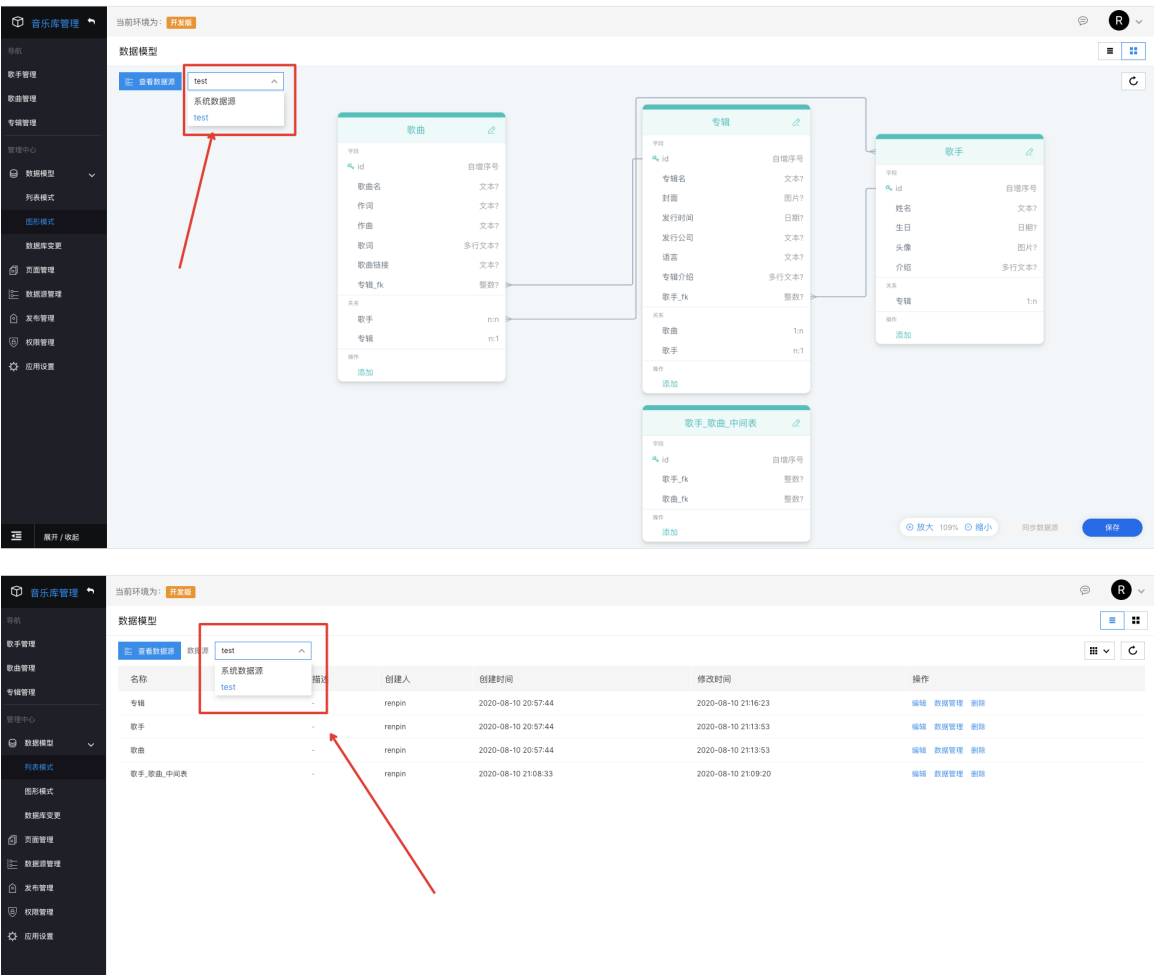
请注意！！默认删除为物理删除，测试前请注意数据备份。

更改字段类型

数据源模型支持所有的基本字段类型和绝大部分高级字段类型，可以通过更改默认映射的字段类型来实现高级字段类型使用

例如：我们有一个表【歌手】，有一个字段为【头像】，我们想保存头像图片地址，字段类型为varchar，步骤如下

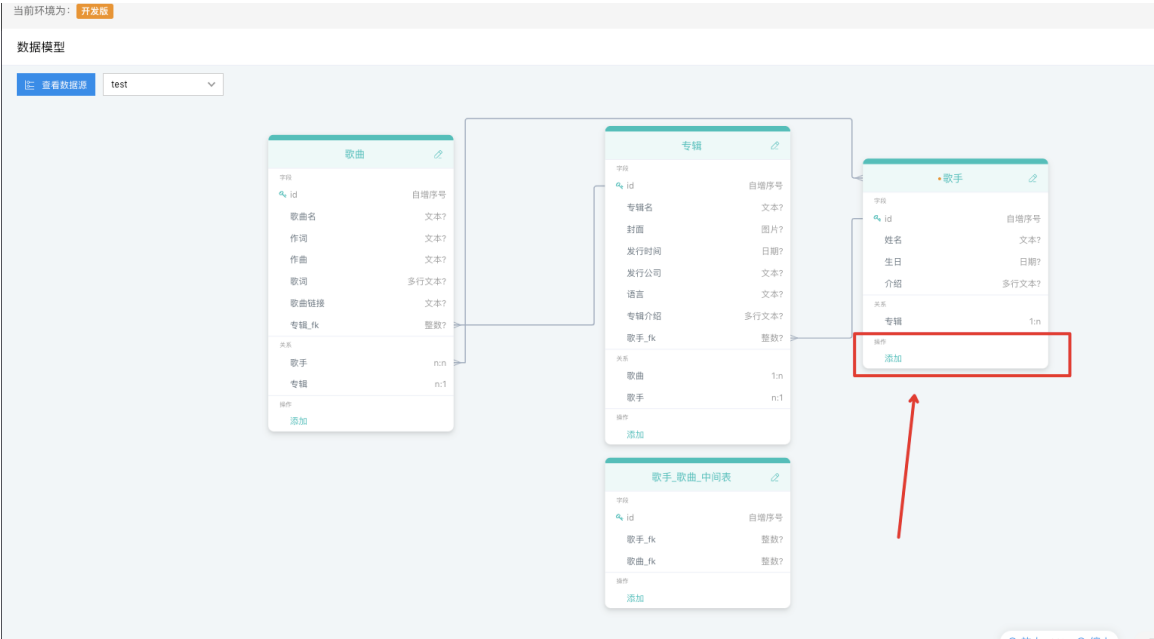
- 1. 在数据模型 -> 列表模式/图形模式 -> 选择数据源 -> 查看响应数据源模型



- 2. 编辑数据源字段

在图形模式下可以直接编辑，列表模式下需要进入编辑界面，并切换到【字段集合】面板

- 3. 删除默认映射的【头像】字段，然后点击添加



4. 选择高级字段中的【图片】类型，并下一步



5. 勾选之前删除的字段【头像】，并调整配置



6. 点击保存，可以在数据管理中预览效果

新增歌手会渲染一个图片选择器组件，可以直接选择图片文件进行上传



新增「歌手」

姓名

生日

请选择日期

头像

介绍

专辑

<空>

+ 新增

取消

确认

选择成功后点击保存，可以在列表中预览图片

歌手列表

+ 新增「歌手」

| id | 姓名  | 生日                  | 头像                                                                                             | 介绍                                                                                                       | 专辑          | 操作       |
|----|-----|---------------------|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|-------------|----------|
| 1  | 周杰伦 | 1979-01-18 00:00:00 | <div></div> | 周杰伦 (Jay Chou) ，1979年1月18日出生于台湾省新北市，中国台湾流行乐男歌手、音乐人、演员、导演、编剧、监制、商人。 2000年发行首张个人专辑《Jay》，2001年发行的专辑《范特西》... | 查看关联的「专辑」列表 | 查看 编辑 删除 |

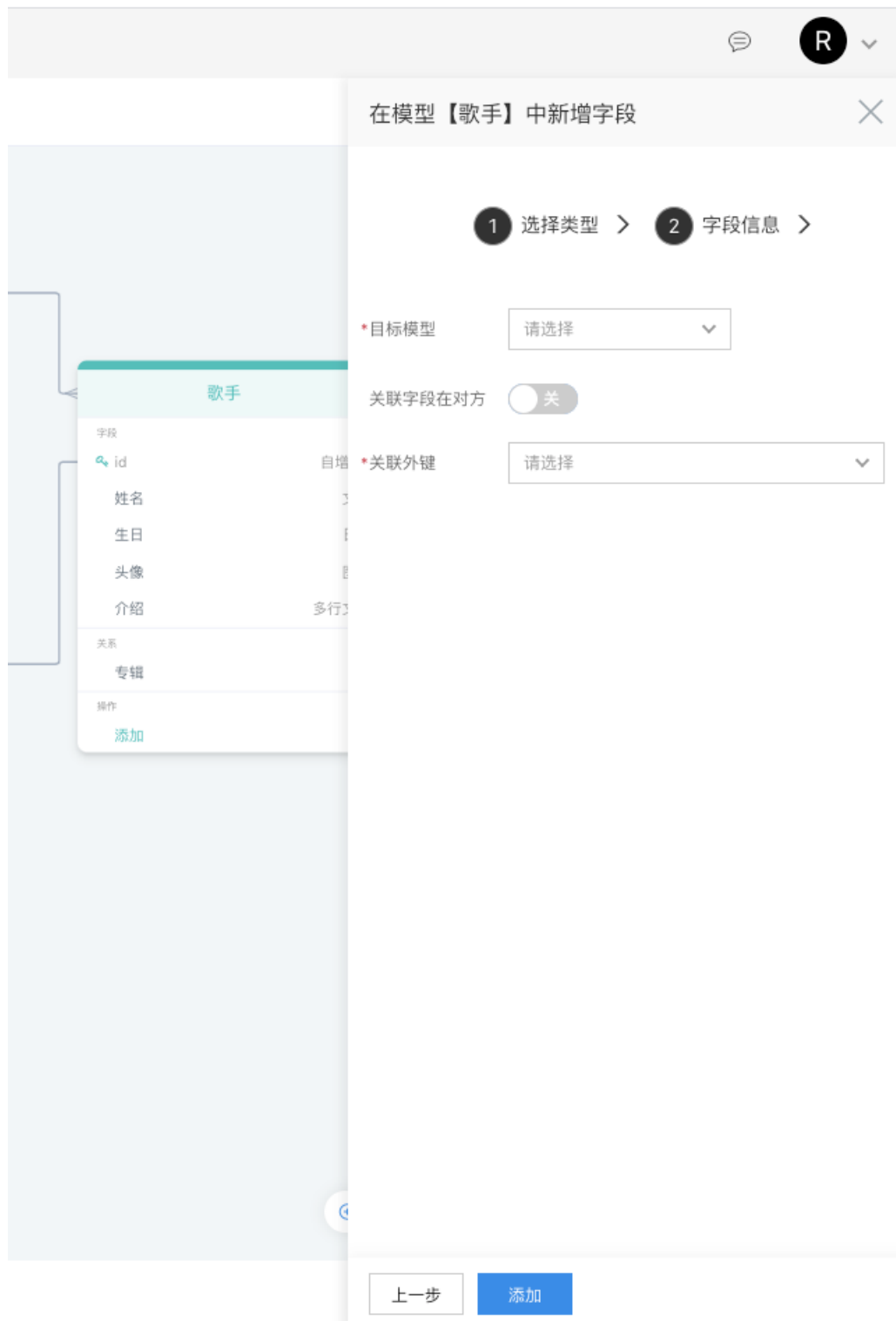
其他字段类型请查看数据模型相应介绍。

添加关系

数据源模型也支持添加关系，支持【一对一】、【一对多】、【多对一】、【多对多】

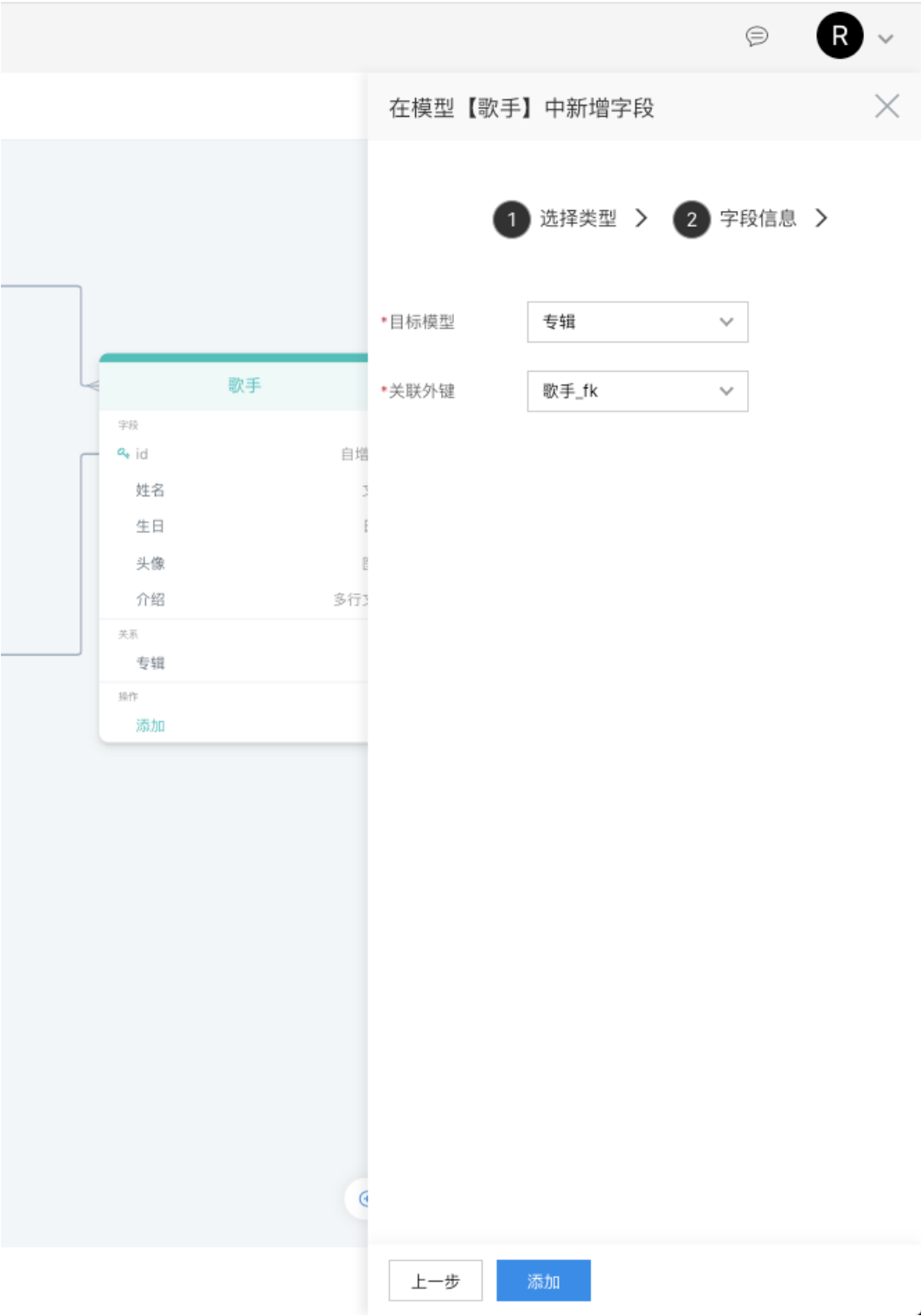


一对一



- 目标模型：选中需要建立关系的目标模型
- 关联字段在对方：关联的外键是否在对方表中
- 关联外键：选择关联的外键

一对多/多对一



- 目标模型：选择需要建立关系的目标模型
- 关联外键：选择关联的外键字段

多对多

多对多需要提前提供一张中间表



- 目标模型：选中需要建立关系的目标模型；
- 中间表：选中多对多关系需要的中间表模型
- 「xx」外键：分别选中关联两个表的外键

外部数据源接入

请确保数据源可通过外网访问。

数据源接入操作步骤

1. 进入应用 -> 左侧点击数据源管理
2. 点击列表左上角的新增「数据源」按钮



3. 弹框中，填写下面配置



4. 点击测试连通性并下一步，然后选择需要接入的数据源表，并调整配置

选中需要接入的数据源表，并进行初始化配置，平台会自动生成同名的数据模型，以供后续操作

新增「数据源」

1 数据源连接 >

2 映射配置 >

3 完成 >

选择数据库表

x 歌手

x 歌手\_歌曲\_中间表

x 专辑

x 歌曲

选择需要进行操作的远端数据库表，同步映射数据源模型

数据模型配置

歌手

歌手\_歌曲\_中间表

专辑

歌曲

\*主键字段

id

选择当前表的主键，需要设置自动递增

标题字段

请选择

请选择一个字段，当在关系视图下，用来展示并帮助用户选择该模型

开启软删除

关

开启并选中 删除时间字段，删除数据后不会进行物理删除，而是会将删除时间记录在选中字段上

开启记录时间

关

开启并选中 创建时间、更新时间字段，操作数据后会记录操作时间在选中字段上

记录操作人

关

开启该开关，并选中 新建人、更新人、删除人 字段，操作数据后，会记录当前操作人在选中字段上

上一步

下一步

- 主键字段：必填，必须是主键且是开启自增的字段
- 标题字段：用于关系中展示字段，方便选择
- 开启软删除：开启并绑定记录删除时间的字段，实现软删除功能（即非物理删除）
- 开启记录时间：开启并绑定时间字段（新建时间、更新时间），可以实现记录数据操作时间功能
- 开启操作人：开启并绑定操作人字段（新建人、更新人、删除人），可以实现记录数据操作人功能

模型创建成功后，也可以在数据模型编辑中修改这些配置

5. 点击下一步，进行创建前预览

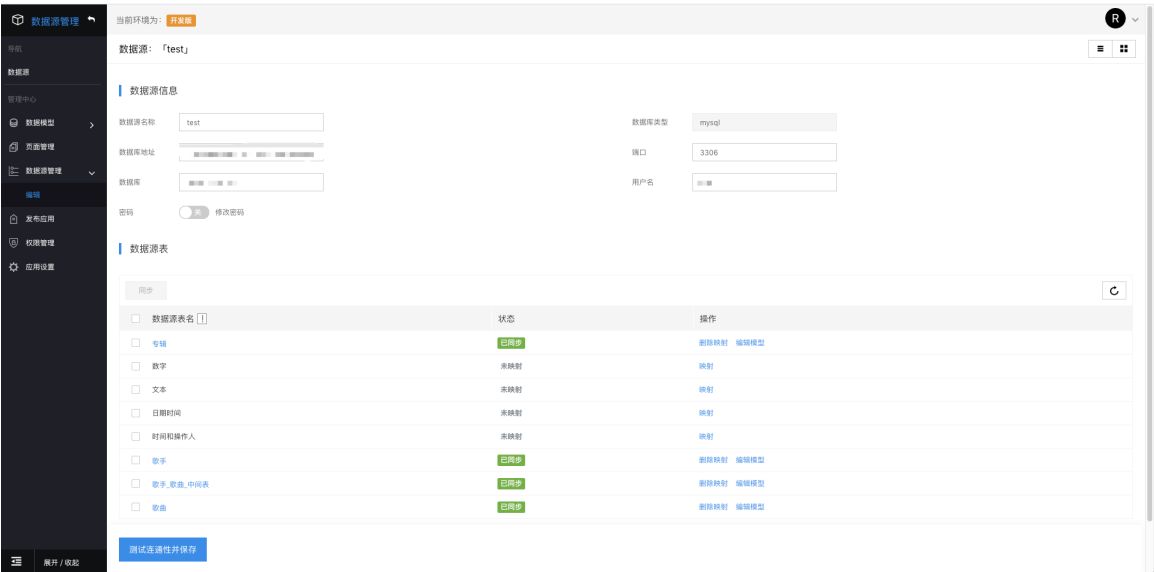


6. 点击完成，创建成功，可以进行后续的操作啦



数据源编辑

- 1. 进入应用 -> 左侧点击数据源管理
- 2. 查看数据源列表操作列，点击编辑，会看到如下界面



编辑数据源信息

可以修改数据信息，点击测试连通性并保存。

查看和编辑 数据源表信息

查看映射详情



已映射的数据源表，可以点击表名查看映射详情

数据源表

|                                    |     |              |  |
|------------------------------------|-----|--------------|--|
| 同步                                 |     |              |  |
| <input type="checkbox"/> 数据源表名     | 状态  | 操作           |  |
| <input type="checkbox"/> 专辑        | 已同步 | 删除映射 编辑模型    |  |
| <input type="checkbox"/> 数字        | 已同步 | 删除映射 编辑模型    |  |
| <input type="checkbox"/> 文本        | 待同步 | 同步 删除映射 编辑模型 |  |
| <input type="checkbox"/> 日期时间      | 未映射 | 映射           |  |
| <input type="checkbox"/> 时间和操作人    | 未映射 | 映射           |  |
| <input type="checkbox"/> 歌手        | 已同步 | 删除映射 编辑模型    |  |
| <input type="checkbox"/> 歌手_歌曲_中间表 | 已同步 | 删除映射 编辑模型    |  |
| <input type="checkbox"/> 歌曲        | 已同步 | 删除映射 编辑模型    |  |

详情弹框

「文本」表结构

| 字段名        | 字段类型       | 长度  | 无符号 | 可否为NULL | 主键 | 是否唯一 | 默认值 | 注释 | 已映射类型 | 状态   |
|------------|------------|-----|-----|---------|----|------|-----|----|-------|------|
| id         | int        | -   | ×   | ×       | ✓  | ×    | -   | -  | 整数    | 已同步  |
| CHAR       | char       | 200 | ×   | ✓       | ×  | ×    | -   | -  | 文本    | 已同步  |
| VARCHAR    | varchar    | 200 | ×   | ✓       | ×  | ×    | -   | -  | 文本    | 已同步  |
| TINYTEXT   | tinytext   | -   | ×   | ✓       | ×  | ×    | -   | -  | 多行文本  | 已同步  |
| TEXT       | text       | -   | ×   | ✓       | ×  | ×    | -   | -  | 多行文本  | 已同步  |
| MEDIUMTEXT | mediumtext | -   | ×   | ✓       | ×  | ×    | -   | -  | 多行文本  | 已同步  |
| LONGTEXT   | longtext   | -   | ×   | ✓       | ×  | ×    | -   | -  | 多行文本  | 已同步  |
| BINARY     | binary     | 200 | ×   | ✓       | ×  | ×    | -   | -  | -     | 暂不支持 |
| VARBINARY  | varbinary  | 100 | ×   | ✓       | ×  | ×    | -   | -  | -     | 暂不支持 |
| TINYBLOB   | tinyblob   | -   | ×   | ✓       | ×  | ×    | -   | -  | -     | 暂不支持 |
| MEDIUMBLOB | mediumblob | -   | ×   | ✓       | ×  | ×    | -   | -  | -     | 暂不支持 |
| BLOB       | blob       | -   | ×   | ✓       | ×  | ×    | -   | -  | -     | 暂不支持 |
| LOB        | longblob   | -   | ×   | ✓       | ×  | ×    | -   | -  | -     | 暂不支持 |
| ENUM       | enum       | -   | ×   | ✓       | ×  | ×    | -   | -  | 枚举    | 已同步  |
| SET        | set        | -   | ×   | ✓       | ×  | ×    | -   | -  | 枚举    | 已同步  |

取消 确认

添加数据源模型

如果想先添加数据源模型，可以点击对应映射按钮，或选中未映射的数据源表，点击左上角【同步】按钮，来映射尚未映射的数据源表：

数据源表

|                                    |     |           |  |
|------------------------------------|-----|-----------|--|
| 同步                                 |     |           |  |
| <input type="checkbox"/> 数据源表名     | 状态  | 操作        |  |
| <input type="checkbox"/> 专辑        | 已同步 | 删除映射 编辑模型 |  |
| <input type="checkbox"/> 数字        | 已同步 | 删除映射 编辑模型 |  |
| <input type="checkbox"/> 文本        | 未映射 | 映射        |  |
| <input type="checkbox"/> 日期时间      | 未映射 | 映射        |  |
| <input type="checkbox"/> 时间和操作人    | 未映射 | 映射        |  |
| <input type="checkbox"/> 歌手        | 已同步 | 删除映射 编辑模型 |  |
| <input type="checkbox"/> 歌手_歌曲_中间表 | 已同步 | 删除映射 编辑模型 |  |
| <input type="checkbox"/> 歌曲        | 已同步 | 删除映射 编辑模型 |  |



添加新的数据源模型同样需要进行初始化配置



同步数据源模型

自动同步

默认开启自动同步策略，远端数据库更新后，在使用数据源的时候，便会自动同步响应表结构

手动同步

如果关闭的自动同步，远端数据源表有更新，需要点击对应数据源表操作列中的同步按钮，进行同步

数据源表

同步

C

| 数据源表名                                  | 状态  | 操作           |
|----------------------------------------|-----|--------------|
| <input type="checkbox"/> 专辑            | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 数字            | 已同步 | 删除映射 编辑模型    |
| <input checked="" type="checkbox"/> 文本 | 待同步 | 同步 删除映射 编辑模型 |
| <input type="checkbox"/> 日期时间          | 未映射 | 映射           |
| <input type="checkbox"/> 时间和操作人        | 未映射 | 映射           |
| <input type="checkbox"/> 歌手            | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 歌手_歌曲_中间表     | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 歌曲            | 已同步 | 删除映射 编辑模型    |

如果有多个数据源表需同步可以选中多个，点击左上角同步按钮

数据源表

同步

C

| 数据源表名                                         | 状态  | 操作           |
|-----------------------------------------------|-----|--------------|
| <input type="checkbox"/> 专辑                   | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 数字                   | 已同步 | 删除映射 编辑模型    |
| <input checked="" type="checkbox"/> 文本        | 待同步 | 同步 删除映射 编辑模型 |
| <input type="checkbox"/> 日期时间                 | 未映射 | 映射           |
| <input type="checkbox"/> 时间和操作人               | 未映射 | 映射           |
| <input checked="" type="checkbox"/> 歌手        | 已同步 | 删除映射 编辑模型    |
| <input checked="" type="checkbox"/> 歌手_歌曲_中间表 | 已同步 | 删除映射 编辑模型    |
| <input checked="" type="checkbox"/> 歌曲        | 已同步 | 删除映射 编辑模型    |

如果批量选中未映射的数据源表，平台会自动创建响应数据源模型

删除数据源模型

在对应的数据源表操作列中，点击删除映射

数据源表

同步

C

| 数据源表名                              | 状态  | 操作           |
|------------------------------------|-----|--------------|
| <input type="checkbox"/> 专辑        | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 数字        | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 文本        | 待同步 | 同步 删除映射 编辑模型 |
| <input type="checkbox"/> 日期时间      | 未映射 | 映射           |
| <input type="checkbox"/> 时间和操作人    | 未映射 | 映射           |
| <input type="checkbox"/> 歌手        | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 歌手_歌曲_中间表 | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 歌曲        | 已同步 | 删除映射 编辑模型    |

删除成功后，数据源表会回复未映射的状态，已映射的数据源模型会被删除

数据源表

同步

C

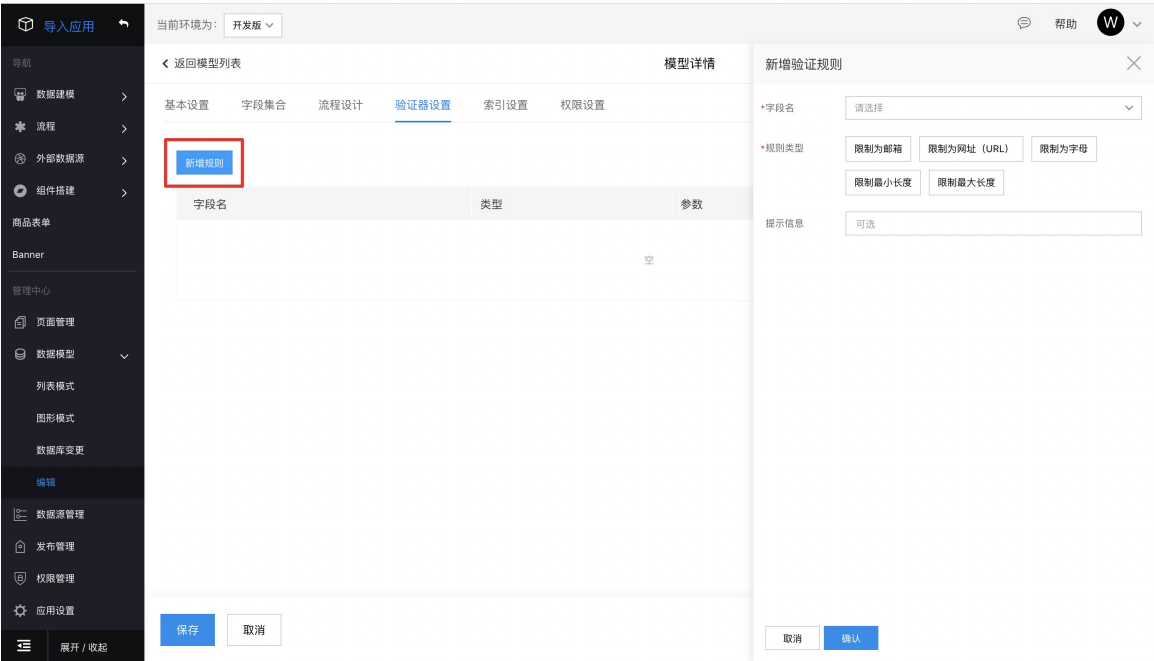
| 数据源表名                              | 状态  | 操作           |
|------------------------------------|-----|--------------|
| <input type="checkbox"/> 专辑        | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 数字        | 未映射 | 映射           |
| <input type="checkbox"/> 文本        | 待同步 | 同步 删除映射 编辑模型 |
| <input type="checkbox"/> 日期时间      | 未映射 | 映射           |
| <input type="checkbox"/> 时间和操作人    | 未映射 | 映射           |
| <input type="checkbox"/> 歌手        | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 歌手_歌曲_中间表 | 已同步 | 删除映射 编辑模型    |
| <input type="checkbox"/> 歌曲        | 已同步 | 删除映射 编辑模型    |

也可以在数据模型页面进行删除

# 模型其他设置

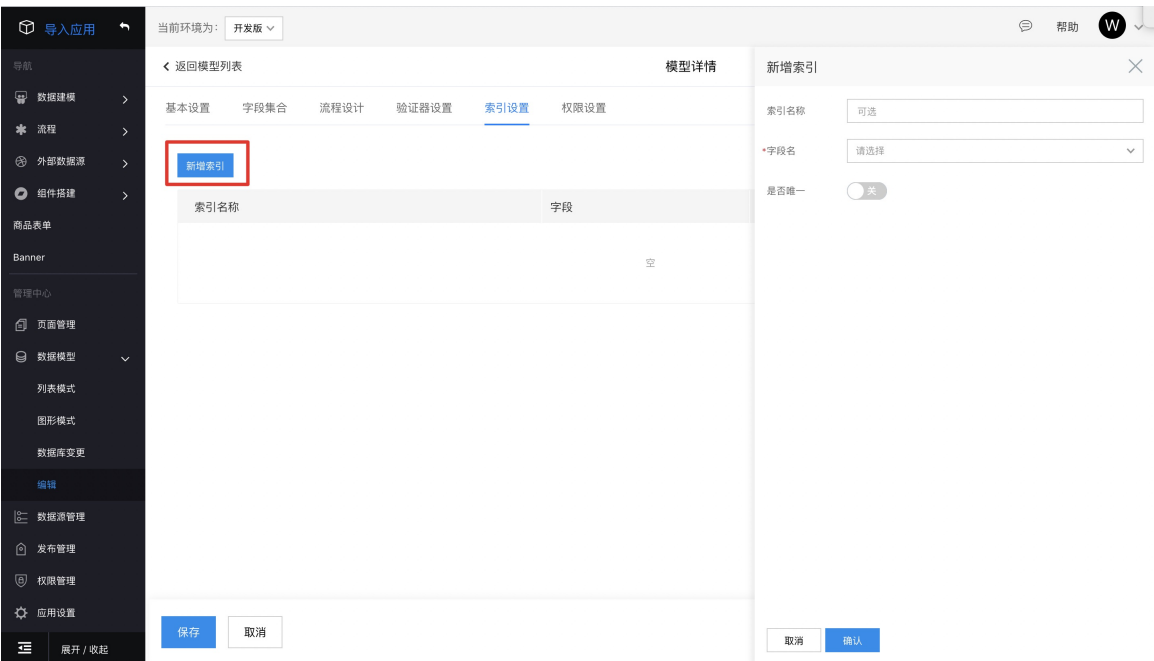
## 验证器设置

验证器设置支持基于平台已支持的规则类型建立验证规则。



## 索引设置

索引设置支持选择模型中的字段建立索引。



## 权限设置

权限设置主要实现数据相关的操作权限设置，默认无限制。

当前环境为： 开发版 ▾

< 返回模型列表

模型详情

基本设置

字段集合

流程设计

验证器设置

索引设置

权限设置

数据权限

☐ 限制删除

数据创建者可删除，其他人可编辑，可查看，不可删除。

☐ 限制写入

数据创建者可编辑、可删除，其他人可读、不可删除，不可修改。

☐ 限制所有

只有数据创建者可见、可编辑、可删除。

☒ 无限制

即所有人都可以查看、修改、删除数据。

☐ 自定义

没有符合你需求的？自己来定义吧

保存

取消

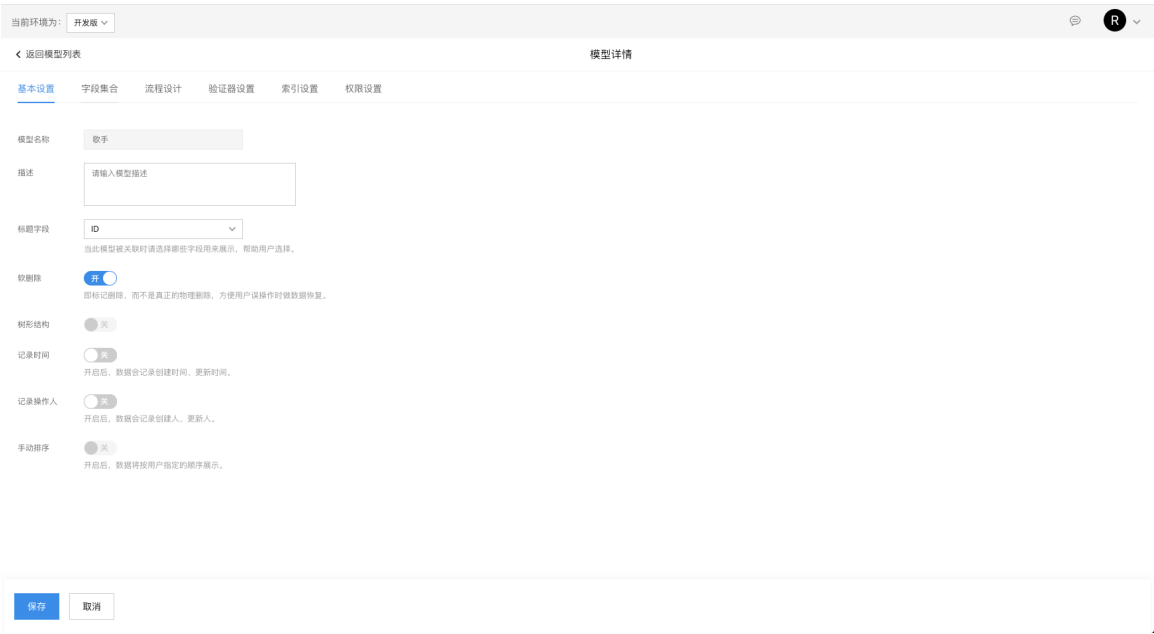
模型详情配置完成后要点击页面下方的【确定】按钮，才能对编辑内容进行保存。

## 模型基本设置

### 🔗 数据信息设置

模型编辑主要由 6 个模块构成：基本设置、字段集合、流程设计、验证器设置、索引设置、权限设置。本节主要对基本设置部分进行介绍，其他部分在后续都有详细的介绍。

下面是模型相关的一些基本设置。



🔗 模型名称

为创建时设置的模型名称，不可修改。

🔗 描述

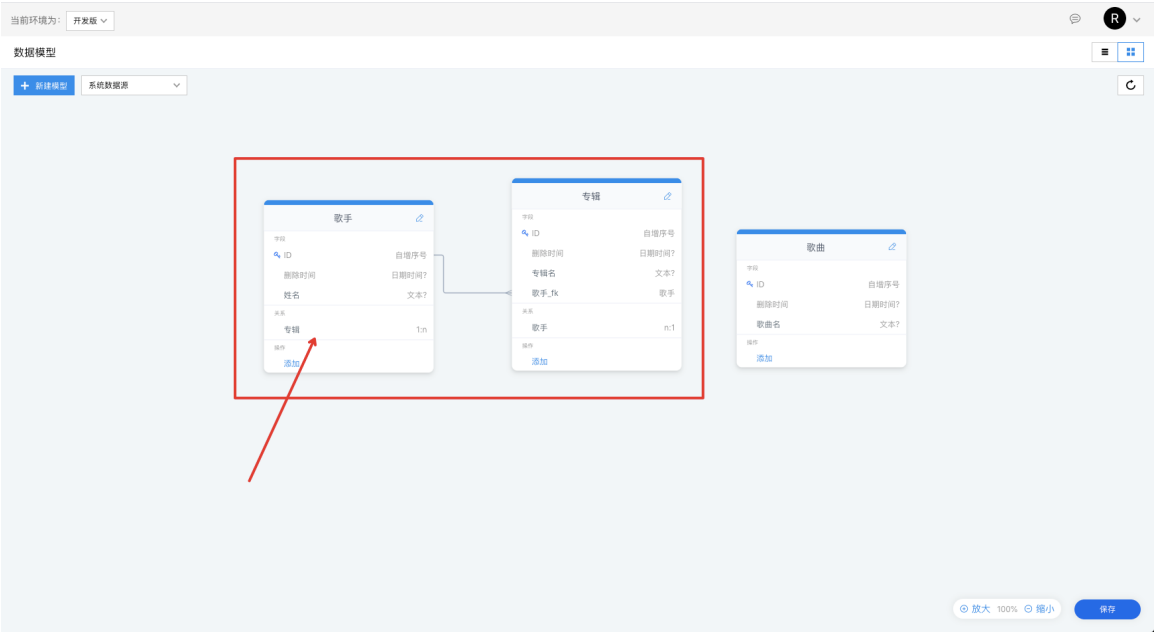
模型的描述信息。

🔗 标题字段

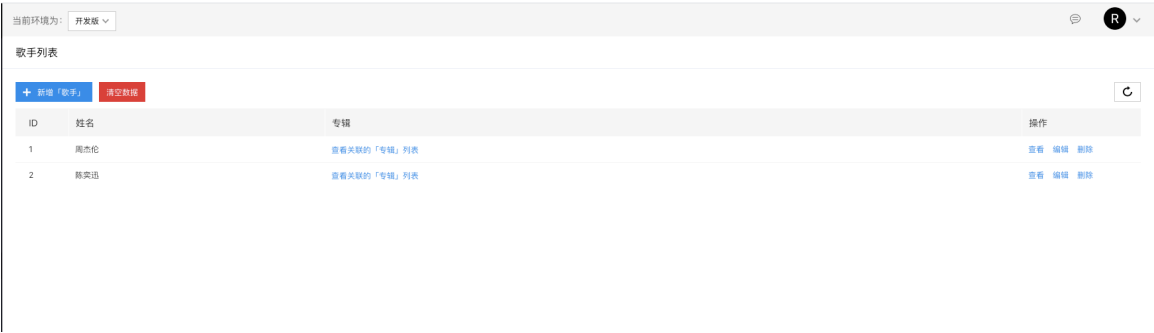
当模型添加关系后，编辑界面中需要勾选互相有关系的模型时，会显示设置的该字段，否则默认就显示 id。



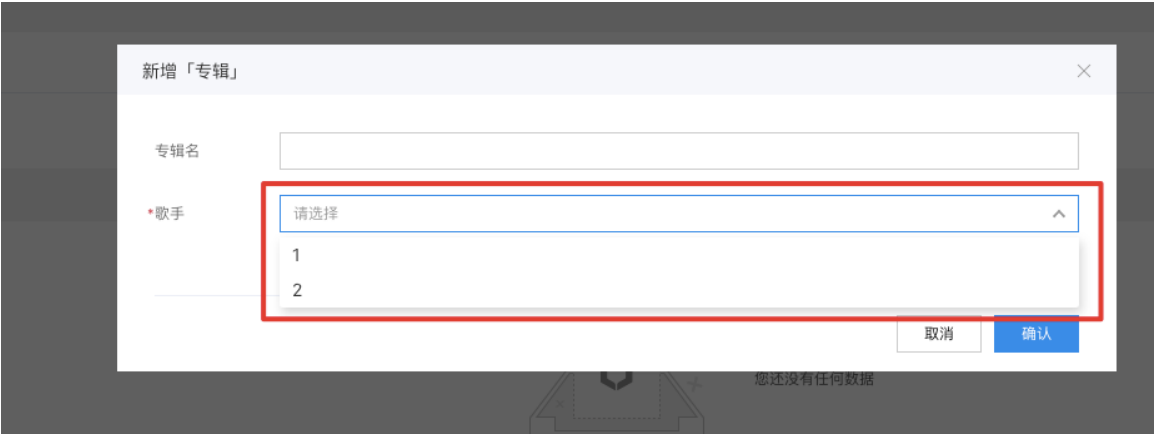
例如下图中，我们给【歌手】添加了一对多关系，目标模型为【专辑】



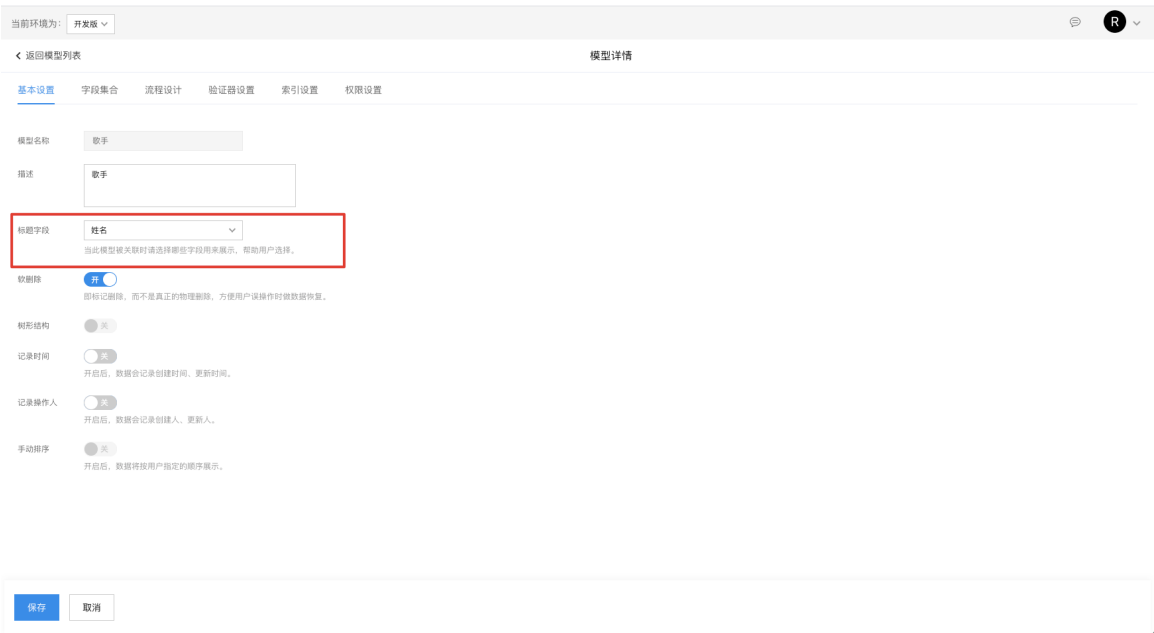
然后我们给添加两个歌手数据：



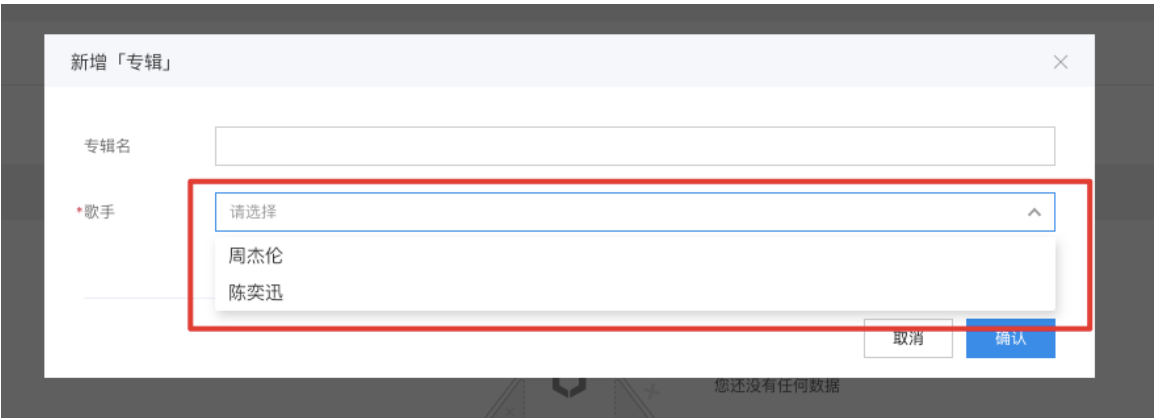
然后我们来到【专辑】数据管理界面，点击新增，我们发现选择歌手显示的数据 id，不方便用户选择



来到【歌手】模型编辑界面，调整【歌手】的模型标题字段为姓名：



然后再回到【专辑】数据管理界面，点击新增，我们发现选择歌手显示的数据为姓名了：



🔗 软删除

当前模型是否开启软删除功能。

开启软删除后，会自动给当前模型添加删除时间字段，删除数据时并不会真正删除掉，而是会记录当前时间到删除时间字段，查询数据时，隐藏有删除时间的数据。



建议开启，方便数据恢复。

🔗 树形结构

近期即将支持，敬请期待。

🔗 记录时间

当前模型是否开启记录时间功能。

开启记录时间后，会自动给当前模型添加创建时间和更新时间字段。新增和修改数据时会记录当前时间到创建时间和更新时间字段。



[返回模型列表](#)

基本设置    **字段集合**    流程设计    验证器设置

| 歌手     |       |
|--------|-------|
| 字段     |       |
| ID     | 自增序号  |
| 删除时间   | 日期时间? |
| 姓名     | 文本?   |
| • 创建时间 | 日期时间  |
| • 更新时间 | 日期时间  |
| 关系     |       |
| 专辑     | 1:n   |
| 操作     |       |
| 添加     |       |

记录操作人

当前模型是否开启记录操作人功能。

开启记录操作人后，会自动给当前模型添加创建人、更新人和删除人字段。新增、修改和删除时，会记录当前操作人到创建人、更新人和删除人字段。

[返回模型列表](#)

基本设置    **字段集合**    流程设计    验证器设置

| 歌手    |       |
|-------|-------|
| 字段    |       |
| ID    | 自增序号  |
| 删除时间  | 日期时间? |
| 姓名    | 文本?   |
| • 创建人 | 人员信息  |
| • 更新人 | 人员信息  |
| • 删除人 | 人员信息? |
| 关系    |       |
| 专辑    | 1:n   |
| 操作    |       |
| 添加    |       |

手动排序

近期即将支持，敬请期待。

## 模型字段配置

### 🔗 模型字段

字段集合可以新增字段和关系。字段和数据库的表的字段类似，具备属性；关系是表与表之间的关联，和 E-R 图中表之间的关系类似。

### 🔗 添加字段

模型字段共分为两大类，基础字段类型和高级字段类型。

#### 基础字段

##### ☐ 文本

用来存储小段文本信息，比如：名称、邮箱、网址、身份证号等等

##### ☐ 多行文本

用来存储大片文本信息，长度不限，可以用来存储文章内容。

##### ☐ 整数 (Int)

用来存储整型数据，比如：年龄、长度、距离等等

##### ☐ 浮点数 (Float)

用来存储带小数点的数字，比如：经度、纬度等等

##### ☐ 日期

用来存储日期格式，包含：日期时间、日期、时间和时间戳格式。

##### ☐ 日期范围

用来存储日期范围格式。

##### ☐ 枚举

用来存储固定的某几个值，常用来存储状态。

##### ☐ 布尔 (开关)

用来存储是与否。

##### ☐ JSON

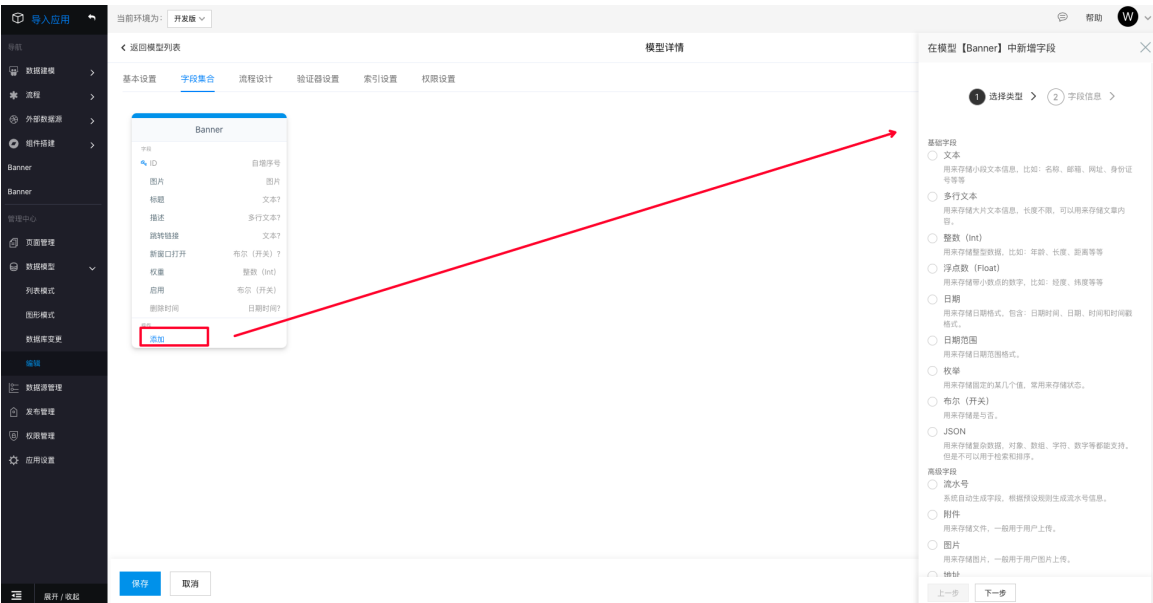
用来存储复杂数据，对象、数组、字符、数字等都能支持。但是不可以用于检索和排序。

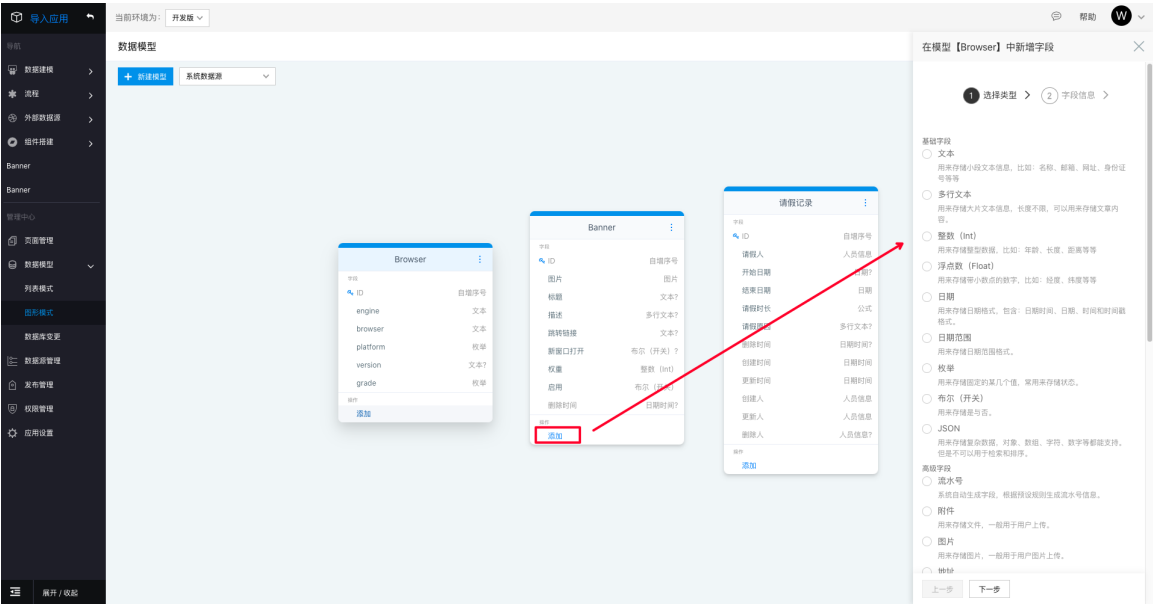
高级字段

- ☐ 流水号  
系统自动生成字段，根据预设规则生成流水号信息。
- ☐ 附件  
用来存储文件，一般用于用户上传。
- ☐ 图片  
用来存储图片，一般用于用户上传。
- ☐ 地址  
用来存储地址，包含省份、城市、地区街道信息。
- ☐ 位置  
用来存储地理位置，包含经纬度信息。
- ☐ 密码  
用来存储密文，只能用于结果比对，不可反解。
- ☐ 密文  
用来存储加密文本，可反解。
- ☐ 金额  
用来存储金额类型字段，单位为 分
- ☐ 人员信息  
可用来存储人员信息，与平台用户表关联。
- ☐ 拥有者  
用来存储数据所属人员信息，可用于权限设置。
- ☐ 所属部门  
用来存储数据所属部门信息，可用于权限设置。
- ☐ 公式  
不存储数据，根据公式自动计算结果

- 基础字段：文本、多行文本、整数、浮点数、日期、日期范围、枚举、布尔（开关）、JSON
- 高级字段：流水号、附件、图片、地址、位置、密码、密文、金额、人员信息、拥有者、所属部门、公式

在模型图的底部，点击添加即可显示选择字段类型抽屉。





公共属性

- 字段名：模型的字段名，支持中文与英文；
- 描述：描述当前字段作用，一般会在对应表单项下展示；

1 选择类型 >

2 字段信息 >

\*字段名：

姓名

描述：

歌手的姓名

\*格式：

普通文本

邮箱

网址

身份证号

手机号

电话号码

邮编

颜色

年份

允许多选：

关

允许空值：

开

长度：

255

新增「歌手」

姓名

歌手的姓名

取消

确认

- **允许多选**：部分字段不支持该属性，配置字段是否支持以多选模式进行选值
- **允许空值**：配置字段是否可以空

文本

用来存储小段文本信息，比如：名称、邮箱、网址、身份证号等等。

1 选择类型 >

2 字段信息 >

\*字段名：

允许多选：

关

允许空值：

开

长度：

256

^

v

\*格式：

普通文本

邮箱

网址

身份证号

手机号

电话号码

邮编

颜色

年份

- **格式**：表单项和静态展示中，该字段的输出格式
- **长度**：该字段可存的最大长度

多行文本

用来存储大片文本信息，长度不限，可以用来存储文章内容。

1 选择类型 >

2 字段信息 >

\*字段名：

允许空值：

开

- **文字识别**：该字段输入时是否支持图片识别文字

整数

用来存储整型数据，比如：年龄、长度、距离等等。

1 选择类型 > 2 字段信息 >

\*字段名：

存储类型

小整型

整型

大整型

允许多选：

关

允许空值：

开

- 存储类型：字段存储类型，请根据需求选择

浮点数

用来存储带小数点的数字，比如：经度、纬度等等。

1 选择类型 > 2 字段信息 >

\*字段名：

存储类型

带精度浮点数

✓

实数

双精度浮点数

精确小数，可用于货币计算

精度：

^

v

精度是指整个数字里全部位的数目，也就是小数点两边的位数。显示指定类型精度时的最大允许精度为 1000

比例：

^

v

比例是指小数点后边的位数

允许多选：

关

允许空值：

开

- 存储类型：字段存储类型，请根据需求选择
- 精度：精度是指整个数字里全部位的数目，也就是小数点两边的位数。显示指定类型精度时的最大允许精度为 1000
- 比例：比例是指小数点后数字的位数

日期

用来存储日期格式，包含：日期时间、日期、时间和时间戳格式。

1 选择类型 >

2 字段信息 >

\*字段名:

存储类型

日期时间

日期时间 (带时区) ☒

日期

时间

时间 (带时区)

时间日期，包含时期和时间信息，有时区信息。

允许多选:

☐ 关

允许空值:

☒ 开

- 存储类型：字段存储类型，请根据需求选择

日期范围

用来存储日期范围格式。

1 选择类型 >

2 字段信息 >

\*字段名:

存储类型

日期时间

日期时间 (带时区) ☒

日期

时间戳

时间日期，包含时期和时间信息，有时区信息。

允许多选:

☐ 关

允许空值:

☒ 开

- 存储类型：字段存储类型，请根据需求选择

枚举

用来存储固定的某几个值，常用来存储状态。

1 选择类型 >

2 字段信息 >

\*字段名:

描述:

选项集合

描述\*

+ 新增

选项键值分离

关

允许多选:

关

允许空值:

开

- 选项集合：配置枚举的选项组；
- 选项键值分离：选项组内单个选项是否是键值对分离的格式

布尔（开关）

用来存储是与否。

1 选择类型 >

2 字段信息 >

\*字段名:

允许空值:

开

JSON

用来存储复杂数据，对象、数组、字符、数字等都能支持。但是不可以用于检索和排序。



1 选择类型 >

2 字段信息 >

\*字段名:

描述:

允许空值:

开

流水号

系统自动生成字段，根据预设规则生成流水号信息。

1 选择类型 >

2 字段信息 >

\*字段名:

描述:

\*前置字符

\*日期格式

年

年月

年月日

自增序号位数

8

<^

>v

- 前置字符：生成流水号中的前置字符；
- 日期格式：流水号中日期的格式；
- 自增序号位数：自增序号的数字位数

附件

用来存储文件，一般用于用户上传。

1 选择类型 >

2 字段信息 >

\*字段名:

描述:

允许空值:

开

允许的格式

请填写允许文件的 MIME-TYPE，多个类型用逗号隔开。

允许多选:

关

- 允许的格式：填写限制上传的文件格式，为文件后缀名，例如：.mp3。配置多个可以用逗号分隔，例如：.mp3,.mp4

图片

用来存储图片，一般用于用户图片上传。

1 选择类型 >

2 字段信息 >

\*字段名:

描述:

允许多选:

关

允许空值:

开

允许的图片类型

JPEG

PNG

GIF

SVG

限制图片比率

4:3

16:9

1:1

其他

图像安全审核

关

由百度AI自动完成内容审核，包含：色情识别、暴恐识别、政治敏感人物识别、广告识别、图像垃圾文本识别（反作弊）、恶心图像识别等功能

- 允许图片类型：勾选允许的图片类型；
- 限制图片比率：配置是否限制图片比率
- 图像安全审核：是否开启图片安全审核功能，由百度 AI 自动完成内容审核，包含：色情识别、暴恐识别、政治敏感人物识别、广告识别、图像垃圾文本识别（反作弊）、恶心图像识别等功能

地址

用来存储地址，包含省份、城市、地区街道信息。

1 选择类型 >

2 字段信息 >

\*字段名:

描述:

允许多选:

关

允许空值:

开

记录城市

开

记录地区

开

记录街道

关

- **记录城市**：是否记录城市名
- **记录地区**：是否记录地区名
- **记录街道**：是否记录街道名

位置

用来存储地理位置，包含经纬度信息。

1 选择类型 > 2 字段信息 >

\*字段名：

描述：

允许多选：

关

允许空值：

开

记录城市

关

采用geometry

关

目前还不支持 Gemometry 类型，后续 db 安装相应插件再启用。

地图类型

百度地图

▼

存储的经纬度信息将统一存储为国测局坐标（火星坐标，GCJ02）

- 记录城市：是否记录城市；
- 采用 geometry：尚未支持，敬请期待；
- 地图类型：选择地图插件类型，目前只支持百度地图

密码

用来存储密文，只能用于结果比对，不可反解。

1 选择类型 > 2 字段信息 >

\*字段名：

描述：

加盐

设置一段字符, 在原有密文的基础上再次加盐，一旦创建将不可修改。支持\${xxx}取值

允许空值：

开

- 加盐：密码加盐。在密码学中，是指通过在密码任意固定位置插入特定的字符串，让散列后的结果和使用原始密码的散列结果不相符，这种过程称之为“加盐”。

密文

用来存储加密文本，可反解。

1 选择类型 >

2 字段信息 >

\*字段名:

描述:

\*密钥

请输入密钥

数据存储使用aes-256-cbc加密算法；  
64位密钥基于输入的文本，使用SHA-256算法生成，用于内容的加密和解码。

允许空值:

开

- 密钥：数据存储使用 aes-256-cbc 加密算法；64 位密钥基于输入的文本，使用 SHA-256 算法生成，用于内容的加密和解码。

金额

用来存储金额类型字段，单位为分

1 选择类型 >

2 字段信息 >

\*字段名:

描述:

允许空值:

开

币种

人民币

默认值:

^  
v

- 币种：选择币种

人员信息

可用来存储人员信息，与平台用户信息关联。

1 选择类型 > 2 字段信息 >

\*字段名

描述:

允许输入

关

是否允许输入，如果不允许则意味着会自动填写。

允许空值:

开

- **允许输入**：配置是否允许输入，如果不允许则意味着会自动设置当前操作人。

拥有者

用来存储数据所属人员信息，可用于权限设置。

1 选择类型 > 2 字段信息 >

\*字段名

拥有者

描述:

允许输入

关

是否允许输入，如果不允许则意味着会自动填写。

允许空值:

开

- **允许输入**：配置是否允许输入，如果不允许则意味着会自动设置当前操作人。

所属部门

用来存储数据所属部门信息，可用于权限设置。

1 选择类型 >

2 字段信息 >

\*字段名

所属部门

描述:

允许多选:

关

允许空值:

开

公式

不存储数据，根据公式自动计算结果

1 选择类型 >

2 字段信息 >

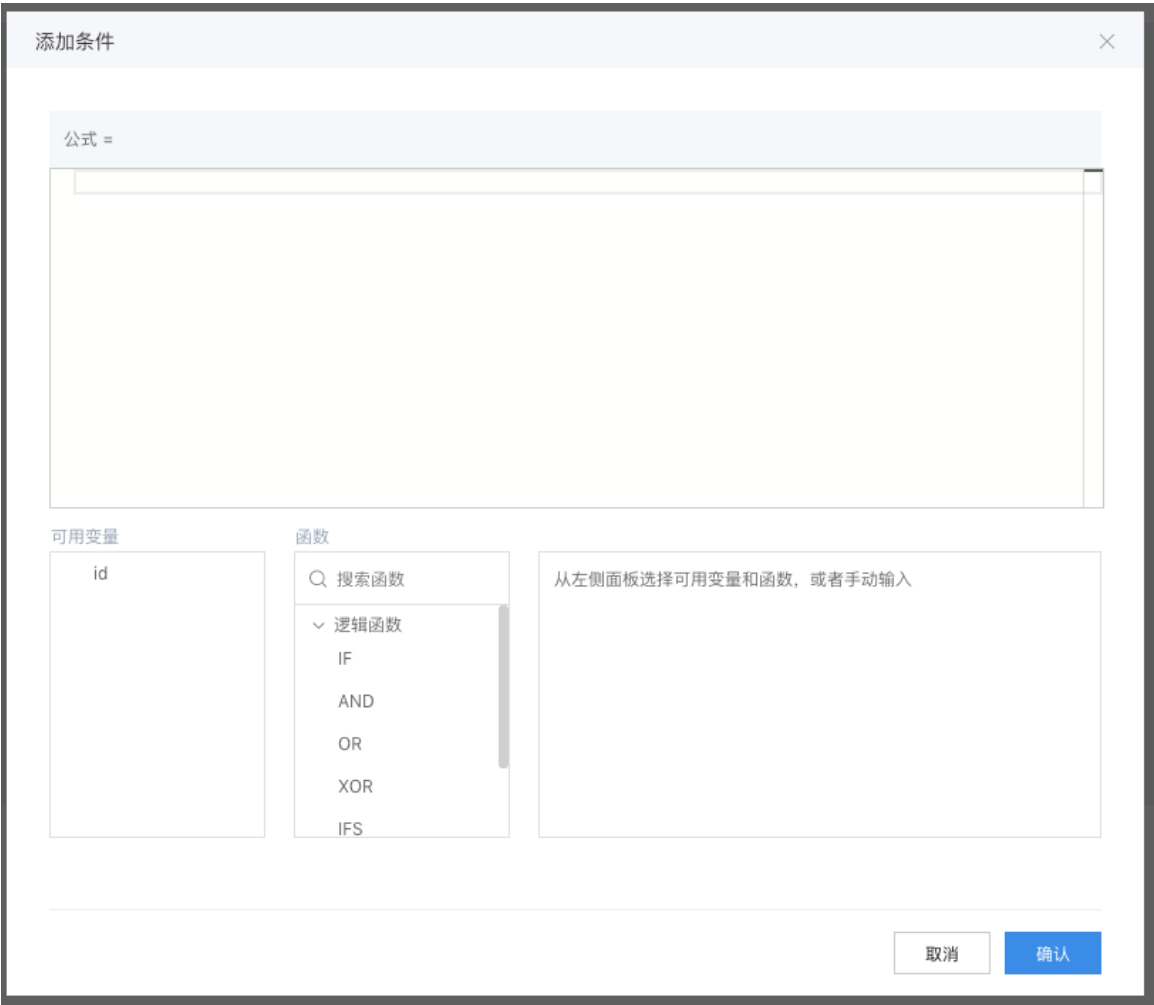
\*字段名:

描述:

\*公式

点击开始编辑

公式编辑：



编辑字段

可以在模型图中的字段右侧，鼠标悬浮在字段上，可以点击“铅笔”符号编辑字段属性。



删除字段

可以在模型图中的字段右侧，鼠标悬浮在字段上，可以点击“减号”符号删除字段。





添加关系

可以给模型添加关系，从而关联其他模型，更加方便实现数据间的联通。

关系

☐ 一对一

当前模型一条数据仅对应目标模型一条数据。比如「用户」与「资料」之间的关系。关系字段将创建在当前模型上。

☐ 多对一

当前模型多条数据对应目标模型一条数据，可以理解为属于关系，比如多篇「文章」属于同一个「作者」。关系字段将在目标模型中创建。

☐ 一对多

当前模型一条数据对应目标模型多条数据，可以理解为拥有关系，比如「作者」拥有很多「文章」。关系字段将在当前模型中创建。

☐ 多对多

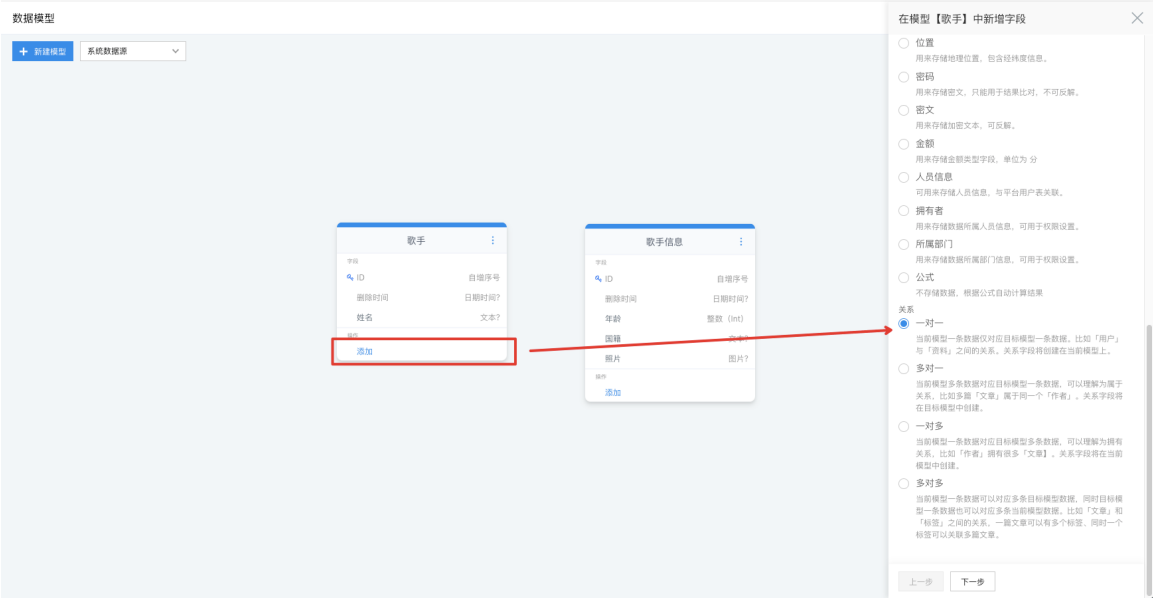
当前模型一条数据可以对应多条目标模型数据，同时目标模型一条数据也可以对应多条当前模型数据。比如「文章」和「标签」之间的关系，一篇文章可以有多个标签、同时一个标签可以关联多篇文章。

一对一

首先我们准备两个模型，分别为：【歌手】和【歌手信息】



然后我们为【歌手】添加一对一关系，关联【歌手信息】



- **目标模型**：选择要关联的模型，这里我们选择【歌手信息】
- **关联字段在对方**：配置标记关联关系的字段是否在对方，这里我们暂时不开启
- **允许空值**：该关系字段是否可以空

点击确认添加成功后，我们可以看到关系已经成功关联。



歌手信息\_fk 是 歌手信息外键的意思

这时候我们来到数据管理，点击新建歌手时，可以看到可以同步添加歌手信息了：

新增「歌手」

姓名

歌手信息

\*年龄

国籍

照片

取消

确认

我们添加一条数据：

新增「歌手」

姓名

周杰伦

歌手信息

\*年龄

41

国籍

中国

照片

清空数据

取消

确认

添加成功后，我们去看歌手信息数据管理页面，可以看到也同步添加了一条信息：

173

歌手信息列表

清空数据

| ID | 年龄 | 国籍 | 照片                                                                                | 操作                                                       |
|----|----|----|-----------------------------------------------------------------------------------|----------------------------------------------------------|
| 1  | 41 | 中国 |  | <a href="#">查看</a> <a href="#">编辑</a> <a href="#">删除</a> |

多对一

首先我们准备两个模型，分别是：【歌手】和【专辑】

歌手

字段

ID自增序号

删除时间日期时间?

姓名文本?

操作

添加

专辑

字段

ID自增序号

删除时间日期时间?

专辑名文本?

操作

添加

因为一个【歌手】可以对应多个专辑，因此我们在【专辑】上添加多对一关系：

数据模型

+ 新建模型

系统数据源

歌手

字段

ID自增序号

删除时间日期时间?

姓名文本?

操作

添加

专辑

字段

ID自增序号

删除时间日期时间?

专辑名文本?

操作

添加

在模型【专辑】中新增字段

位置

用来存储地理位置，包含经纬度信息。

密码

用来存储密码，只能用于结果比对，不可反解。

密文

用来存储加密文本，可反解。

金额

用来存储金额类型字段，单位为 分

人员信息

用来存储人员信息，与平台用户表关联。

所有者

用来存储数据所属人员信息，可用于权限设置。

所属部门

用来存储数据所属部门信息，可用于权限设置。

公式

不存储数据，根据公式自动计算结果

关系

一对一

当前模型一条数据仅对应目标模型一条数据。比如「用户」与「资料」之间的关系。关系字段将创建在当前模型上。

多对一

当前模型多条数据对应目标模型一条数据。可以理解为属于关系。比如多篇「文章」属于同一个「作者」，关系字段将在目标模型中创建。

一对多

当前模型一条数据对应目标模型多条数据。可以理解为拥有关系。比如「作者」拥有很多「文章」。关系字段将在当前模型中创建。

多对多

当前模型一条数据可以对应多条目标模型数据。同时目标模型一条数据也可以对应多条当前模型数据。比如「文章」和「标签」之间的关系。一篇文章可以有多个标签，同时一个标签可以关联多篇文章。

上一步

下一步

在模型【专辑】中新增字段

1 选择类型 >

2 字段信息 >

\*目标模型

专辑

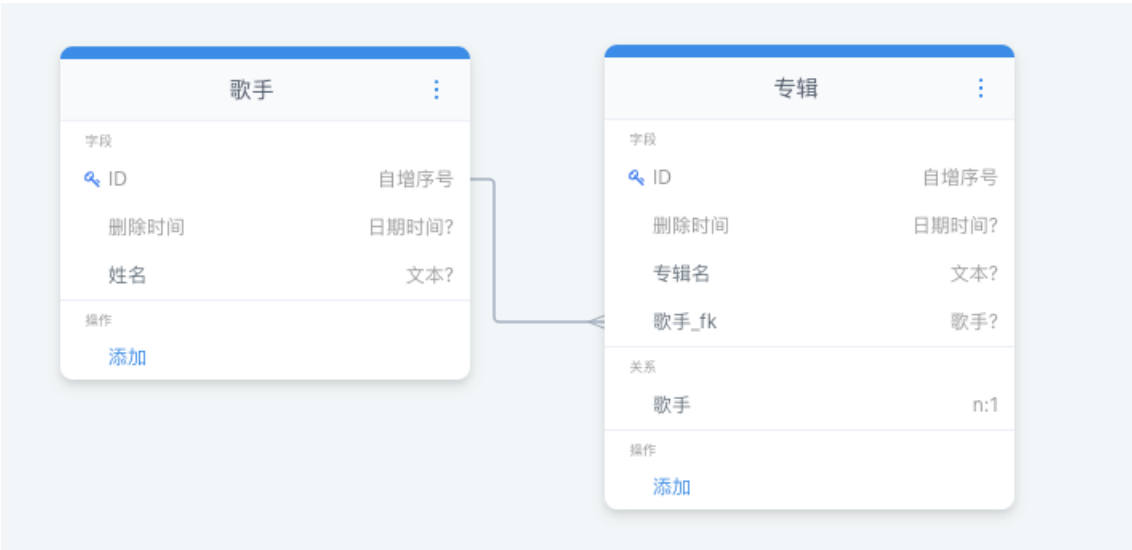
可否为空值

开

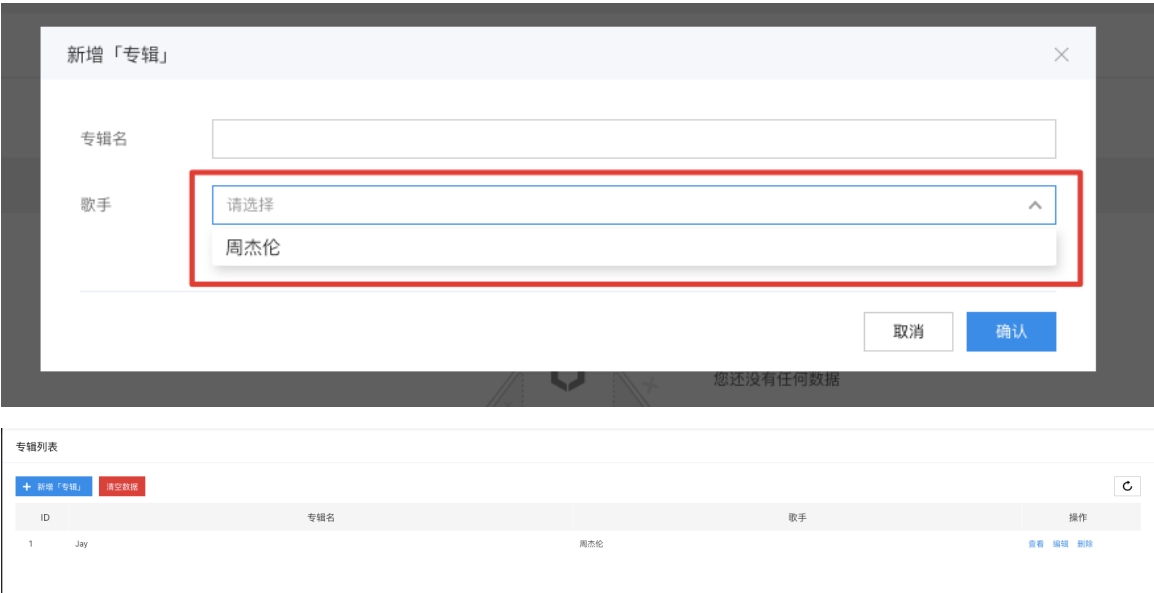
- 目标模型：选中需要关联的目标模型，这里我们选择【歌手】

- 可否为空值：关系字段是否可以为空值；

确认后，添加成功

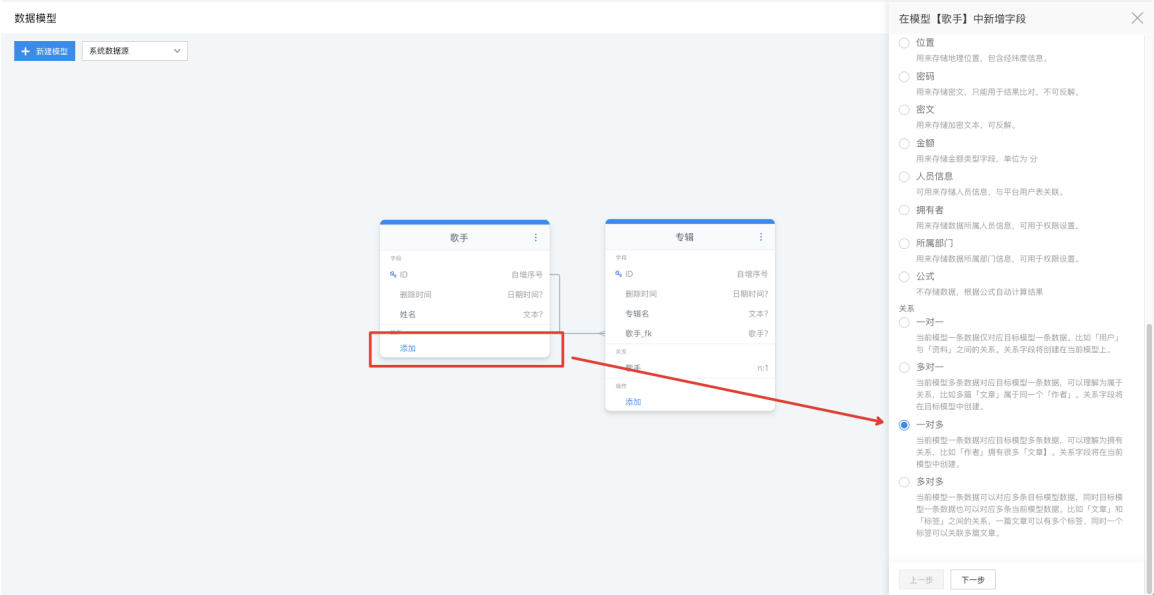


来到专辑的数据管理中，我们点击新增专辑，可以看到出现了勾选歌手的表单项：



一对多

同样因为一个歌手对应多个专辑，我们也可以给歌手模型上添加一对多关系：



在模型【歌手】中新增字段

1 选择类型 >

2 字段信息 >

\*目标模型

专辑

可否为空值

开

- **目标模型**：选中需要关联的目标模型，这里我们选择【专辑】
- **可否为空值**：关系字段是否可以为空值；

确认后，添加成功



这时我们来到【歌手】的数据管理页面，点击新增歌手，可以看到出现了可以新增【专辑】的表单项

新增「歌手」

姓名

专辑

专辑名

+ 新增

取消

确认

多对多

因为一首【歌曲】有可能有多个【歌手】合作，而一名【歌手】也会有很多首【歌曲】作品，因此我们来添加一个多对多关系。

首先准备【歌手】和【歌曲】模型：

歌手

字段

ID

自增序号

删除时间

日期时间?

姓名

文本?

操作

添加

歌曲

字段

ID

自增序号

删除时间

日期时间?

歌曲名

文本?

操作

添加

我们先选择在歌手上添加多对多关系

数据模型

+ 新建模型

系统数据源

歌手

字段

ID

自增序号

删除时间

日期时间?

姓名

文本?

操作

添加

歌曲

字段

ID

自增序号

删除时间

日期时间?

歌曲名

文本?

操作

添加

在模型【歌手】中新增字段

☐ 位置

用来存储地理位置，包含经纬度信息。

☐ 密码

用来存储密码，只能用于结果比对，不可反解。

☐ 密文

用来存储加密文本，可反解。

☐ 金额

用来存储金额类型字段，单位为 分

☐ 人员信息

用来存储人员信息，与平台用户表关联。

☐ 拥有者

用来存储数据所属人员信息，可用于权限设置。

☐ 所属部门

用来存储数据所属部门信息，可用于权限设置。

☐ 公式

不存储数据，根据公式自动计算结果

关系

☐ 一对一

当前模型一条数据对应目标模型一条数据。比如「用户」与「资料」之间的关系。关系字段将创建在当前模型上。

☐ 多对一

当前模型多条数据对应目标模型一条数据。可以理解属于关系。比如多篇「文章」属于同一个「作者」。关系字段将在目标模型中创建。

☐ 一对多

当前模型一条数据对应目标模型多条数据。可以理解拥有关系。比如「作者」拥有很多「文章」。关系字段将在当前模型中创建。

☒ 多对多

当前模型一条数据可以对应多条目标模型数据。同时目标模型一条数据也可以对应多条当前模型数据。比如「文章」和「标签」之间的关系。一篇文章可以有多个标签，同时一个标签可以关联多篇文章。

上一步

下一步

在模型【歌手】中新增字段

1 选择类型 >

2 字段信息 >

\*目标模型

歌曲

可自定义属性

关

除了单纯的关系信息外，如果你想额外的保存一些其他信息比如，关系建立时间、是否为特殊关系等等，请勾选此选项，平台将自动创建关系模型，在创建的关系模型中添加字段即可。

- 目标模型：选中需要关联的目标模型，这里我们选择【歌曲】
- 可自定义属性：除了单纯的关系信息外，如果你想额外的保存一些其他信息比如，关系建立时间、是否为特殊关系等等，请勾选此选项，平台将自动创建关系模型，在创建的关系模型中添加字段即可。

确认后，添加成功



然后我们来到歌手的数据管理界面，点击新增歌手，可以看到出现了勾选歌曲的表单项（需提前添加歌曲数据）：

新增「歌手」

姓名

歌曲

☐ ID

歌曲名

☐ 1

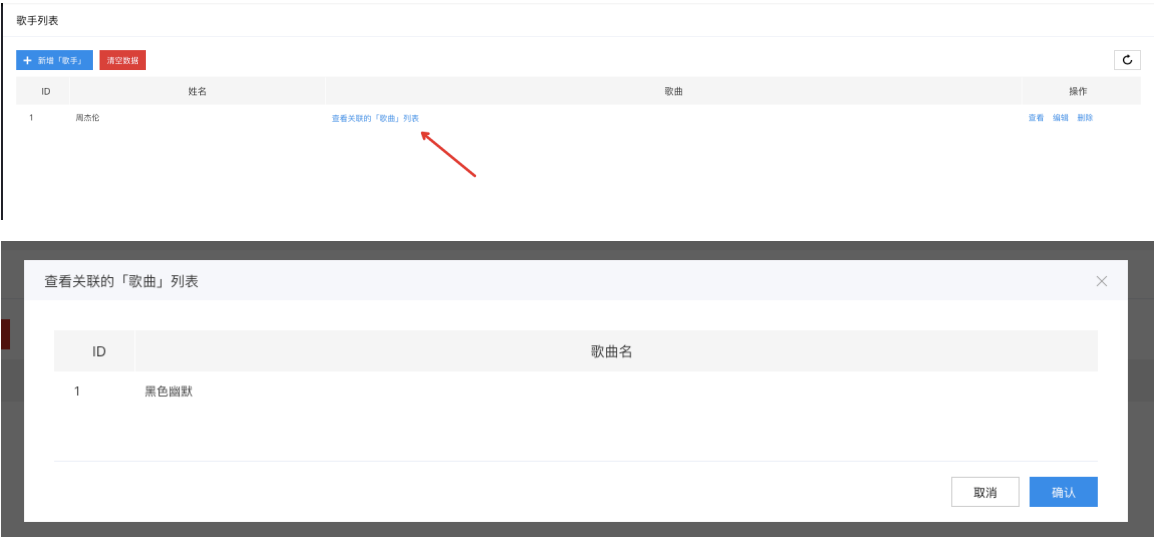
黑色幽默

取消

确认

添加数据后，也可以查看关系数据：





自己来试一试为【歌曲】添加多对多关系吧

编辑关系

可以在模型图中的关系右侧，鼠标悬浮在关系上，可以点击“铅笔”符号编辑关系属性。



删除关系

可以在模型图中的关系右侧，鼠标悬浮在关系上，可以点击“减号”符号删除关系。

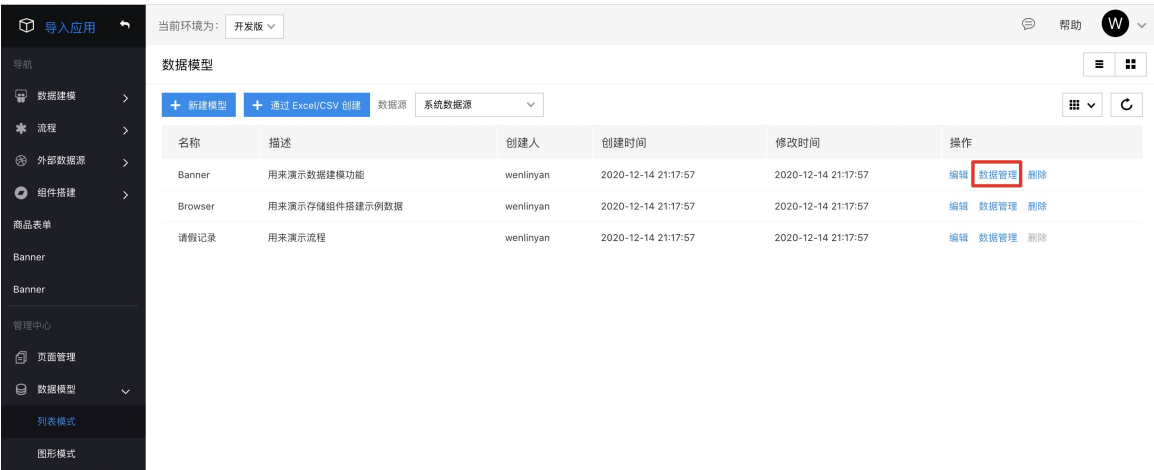


模型数据管理

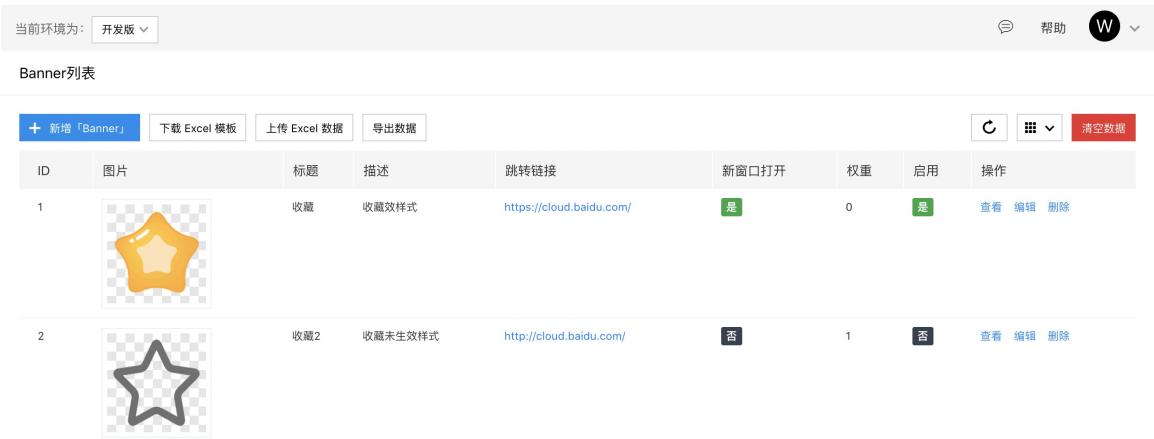
数据管理

由于模型页面是由数据模型一键生成的，因此页面产生的数据也与数据模型直接关联。页面数据管理直接通过模型的数据管理操作完成。

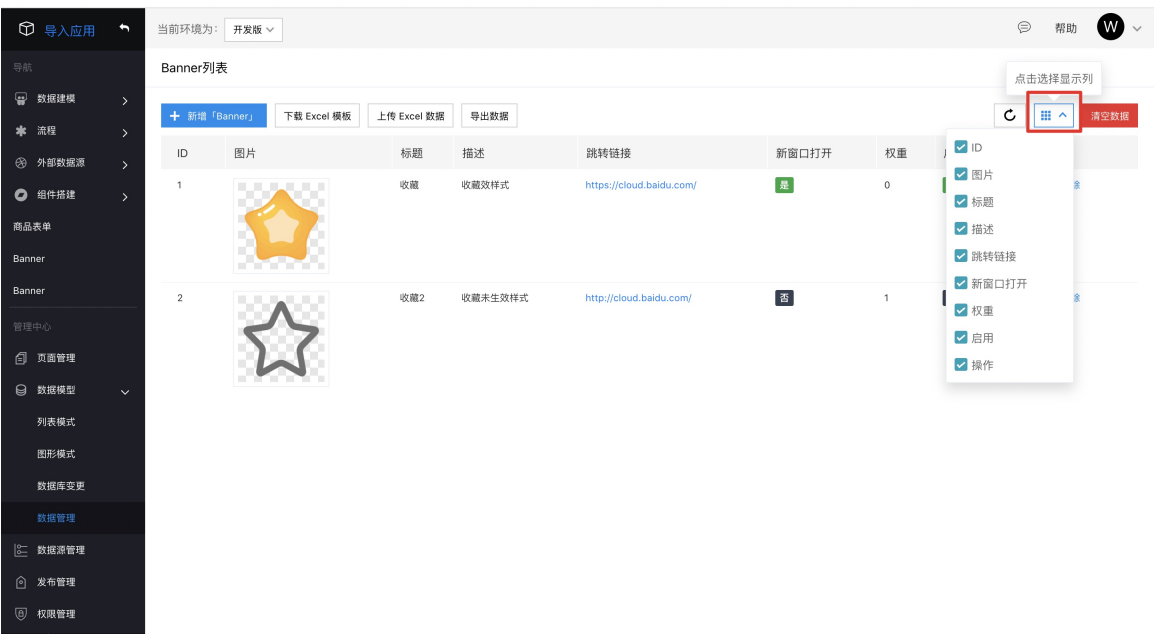
1. 进入数据模型【列表模式】，点击数据模型操作栏【数据管理】，进入数据管理视图。



2. 数据管理视图支持、新增、Excel 导入导出、数据清空操作。



3. 平台上所有字段无法全不展现的列表，都支持表单展示字段的配置。数据列表也支持列展示规则配置。



4. 以及操作栏的快捷操作入口。

Banner列表

+ 新增「Banner」

下载 Excel 模板

上传 Excel 数据

导出数据

刷新

列表

清空数据

| ID | 图片 | 标题  | 描述      | 跳转链接                                                            | 新窗口打开 | 权重 | 启用 | 操作                  |
|----|----|-----|---------|-----------------------------------------------------------------|-------|----|----|---------------------|
| 1  |    | 收藏  | 收藏效样式   | <a href="https://cloud.baidu.com/">https://cloud.baidu.com/</a> | 是     | 0  | 是  | <div>查看 编辑 删除</div> |
| 2  |    | 收藏2 | 收藏未生效样式 | <a href="http://cloud.baidu.com/">http://cloud.baidu.com/</a>   | 否     | 1  | 否  | <div>查看 编辑 删除</div> |

# API对接

## API中心

API 中心用于集中管理页面内的 API，比起直接填入 API 的方式，它增加了如下功能：

- 1. 方便对 API 进行调试
- 2. 每个 API 可以使用不同的签名方式
- 3. 支持对提交参数进行转换
- 4. 支持对返回结果进行转换
- 5. 自定义证书

### 使用方法

在开发版中左侧进入

app

导航

page

应用开发

页面管理

API 中心

API 列表

分组管理

数据模型

数据源

数据管理

当前环境为：开发版

帮助

W

API 列表

查找 API

标题关键词

请输入标题关键词

所属分组

请选择

+ 新增 API

刷新

| 标识                     | API 标题 | API 说明 | 所属分组 | 操作                           |
|------------------------|--------|--------|------|------------------------------|
| nfGweoKqustdSGDtAFdJcq | api    | -      | 默认分组 | <div>查看 编辑 复制 调试 删除 禁用</div> |

每页显示

10

### 签名方式

目前支持三种签名方式：

- HTTP 账号密码。
- JWT，将会对所有 query 及 header 及进行签名，写入 header 的 Authorization 字段中。
- 百度云，将会对提交数据及 header 都做签名，写入 header 的 Authorization 字段中

对于私有部署版本，还支持自定义签名方式。

### URL 地址替换

「请求地址」支持变量替换，比如这个例子 `http://1.1.1.1/blog/{id}`，其中的 `id` 是动态的，而在实际前端调用的时候，请求地址是类似 `/api/center/nicaVyNdbGmHicn5b3EKkU`，如何将它映射到目标地址的路径中？

方法是首先在前端请求的时候映射 query，比如 `post:api://nicaVyNdbGmHicn5b3EKkU?id=${id}`，然后在 api 中的 url 地址填写 `http://1.1.1.1/blog/{{query.id}}`

编辑 API 信息

API 标题

api

API 说明

API 分组

默认分组

+ 新增分组

API 详情

+ 请求地址

http://1.1.1.1/blog/{{query.id}}

除了 `query` 来获取 query，还有以下其它变量：

- `env`，环境变量
- `input`，提交参数
- `headers`，header

提交参数映射

API 中心里的提交参数、Query（URL 参数）、Header 都可以进行修改，转成对应接口所需的格式。

目前参数映射有以下几种方式：

1. 固定值，这个用于固定参数，比如接口需要 `type=1` 这个 query，使用如下方式：

Query 转换

提交 URL 参

数映射

参数映射方法

固定值

值

1

映射为

type

+ 新增

2. 字段名，用于取另一个字段的值，比如将环境变量中的 `TYPE` 值当作 query 中 `type` 的值，使用如下方式：

Query 转换

提交 URL 参

数映射

参数映射方法

字段名

来源字段名

env.TYPE

映射为

type

+ 新增

比如获取环境变量是 env.VARIABLE\_NAME，提交参数是 input.NAME，当前用户邮箱是 userEmail

3. 模板引擎转换，使用 `Handlebars` 模板的结果作为值，比如提交参数中有个 `type`，值为 `1`，但对应的接口需要的值是 `1s`，需要加上个 `s`，这时就可以使用模板转换的方式，如下所示：

Query 转换

提交 URL 参

数映射

参数映射方法

模板引擎转换

模板

{{input.type}}s

映射为

type

+ 新增

4. 公式计算，使用爱速搭中的公式能力进行计算，将结果作为值，比如常见的四则运算。

Query 转换

提交 URL 参

数映射

参数映射方法

公式计算

公式

input.type + input.type2

映射为

type

+ 新增

注意这个功能和页面中接口的「数据映射」功能类似，大部分情况下更推荐使用页面中的「数据映射」，因为它能够，但不同的是可以对 header 进行映射，并且这是后端实现的。

返回结果转换

爱速搭接口所需的返回格式是：

```
{
  "status": 0,
  "msg": "",
  "data": {
    ...其他字段
  }
}
```

- **status**: 状态码，返回 0 表示当前接口正确返回，否则按错误请求处理
- **msg**: 返回接口处理信息，主要用于表单提交或请求失败时的 toast 显示
- **data**: 返回的具体数据

同时也兼容以下几种放回格式

1. errorCode 作为 status、errorMessage 作为 msg
2. errno 作为 status、errmsg/errstr 作为 msg
3. error 作为 status、errmsg 作为 msg
4. error.code 作为 status、error.message 作为 msg
5. message 作为 msg

如果返回的内容不是这些格式，或者返回内容不是 JSON，则需要对返回结果进行转换。

转换方法有以下几种：

1. 复制到，用于将所有数据复制到某个字段下
2. 取字段，用于将数据复制到返回结果的某个字段下
3. 删除字段
4. JSON 路径查询
5. 模板，使用 Handlebars 模板对结果进行转换
6. 字符串拆分数组，比如返回结果是 "1,2,3"，可以将其转成 [1,2,3]
7. 正则，抽取第一个组，通常用于 xml 中直接获取某个数据

这里介绍最常见的「取字段」方法，假设接口返回格式是：

```
{
  "error": 0,
  "error_msg": "",
  "result": {}
}
```

我们希望转成如下格式

```
{
  "status": 0,
  "msg": "",
  "data": {}
}
```

就需要通过「取字段」的方式提取这 3 个值到不同位置，如下图所示：

返回结果转换

原始数据类型

json

转换方法

取字段

字段名

error

比如获取环境变量是 env.VARIABLE\_NAME，提交参数是 input.NAME，当前用户邮箱是 userEmail

映射为

status

如果不填就是整个返回结果

原始数据类型

json

转换方法

取字段

字段名

error\_msg

比如获取环境变量是 env.VARIABLE\_NAME，提交参数是 input.NAME，当前用户邮箱是 userEmail

映射为

msg

如果不填就是整个返回结果

原始数据类型

json

转换方法

取字段

字段名

result

比如获取环境变量是 env.VARIABLE\_NAME，提交参数是 input.NAME，当前用户邮箱是 userEmail

映射为

data

如果不填就是整个返回结果

+ 新增

文件上传

如果是文件上传，请在「请求格式」里设置为「文件上传」。

新增 API

API 标题

API 分组

默认分组

API 说明

+ 新增分组

请求地址

http://example.com/api

可以使用环境变量，比如 \${VARIABLE\_NAME}

请求方法

post

请求格式

json

出错重试次数

0

请求验证方式

无

json

表单

文件上传

文件下载

如果是文件下载接口，请开启左下角的「是文件下载」

## 新增 API

\*API 标题

API 说明

\*请求地址

可以使用环境变量，比如 \${VARIABLE\_NAME}

请求方法

post

▼

请求验证方式

无

提交转换

☐

Query 转换

☐

Header 转换

☐

是文件下载

☒

自定义证书

☐

## API编排

API 编排功能可以通过组合的方式整合处理多个 API / 业务节点的数据，输出业务终端需要的数据结果

🔗 API 编排入门

🔗 基础

API 编排本质上是一种用可视化的方式来编写后端代码，它可以用于不复杂的接口聚合及增删改查场景。

API 编排是树形接口，它的执行顺序是从上至下执行，比如下图：



这个 API 编排首先运行「并行执行」节点，这里面的子节点会并行运行，接着是循环节点，可以对前面的输出结果进行循环处理，然后在这个循环里面有个分支判断，如果成功就运行「设置变量」的节点，最后再运行「发送邮件」节点，如果熟悉写代码，上面这个树等价于下面的代码：

```
// 这个是为了处理并行执行
Promise.all([
  httpResult = await sendHTTPAction(...),
  sqlResult = await runSQL(...)
]);
// 对 http 返回结果进行二次处理
for (const item of httpResult) {
  // 分支
  if (item.status == 1) {
    // 设置变量
    item.status = '成功';
  }
}
// 发送邮件
sendMail(...)
```

各个节点之间如何协作？方法是通过变量，比如发起一个 HTTP 接口，其中有个配置是「存入变量」



执行完后，下面的节点就能通过这个变量来获取到内容，比如后面接一个「设置变量」的节点，在这个节点中，可以通过 `{{}}` 来获取到，比如下面这个例子获取到前面 `output` 变量中的 `name` 值，然后又将它写入变量 `res`。



多接口聚合设计



当所有节点执行完后，output 变量的值就作为最终结果返回前端。

创建和编辑

在【API 中心】列表中点击【新增 API】，弹出面板选择【多接口聚合模式】，即可新增一个 API 编排

新增 API

\*API 标题

API 标识

API 说明

API 标识

不填将随机创建

API 说明

API 分组

默认分组

+ 新增分组

API 类型

单个接口

多接口聚合

多接口聚合细节需要稍后单独编辑

取消

确认

保存后在列表中点击【多接口聚合设计】，即可进行整个 API 编排的设计

+ 新增 API

批量导入

筛选

刷新

| 标识                     | API 标题   | API 说明 | 所属分组 | 操作                                                                                                                                          |
|------------------------|----------|--------|------|---------------------------------------------------------------------------------------------------------------------------------------------|
| v7H4dq4vcQN3Dea2SB16te | 多接口聚合API | -      | 默认分组 | <a href="#">查看</a> <a href="#">编辑</a> <a href="#">多接口聚合设计</a> <a href="#">复制</a> <a href="#">使用情况</a> <a href="#">删除</a> <a href="#">禁用</a> |

每页显示 10

点击连线上的 + 号，可添加连线所在层级的平级节点，点击节点右侧的 + 号，可添加该节点的子节点

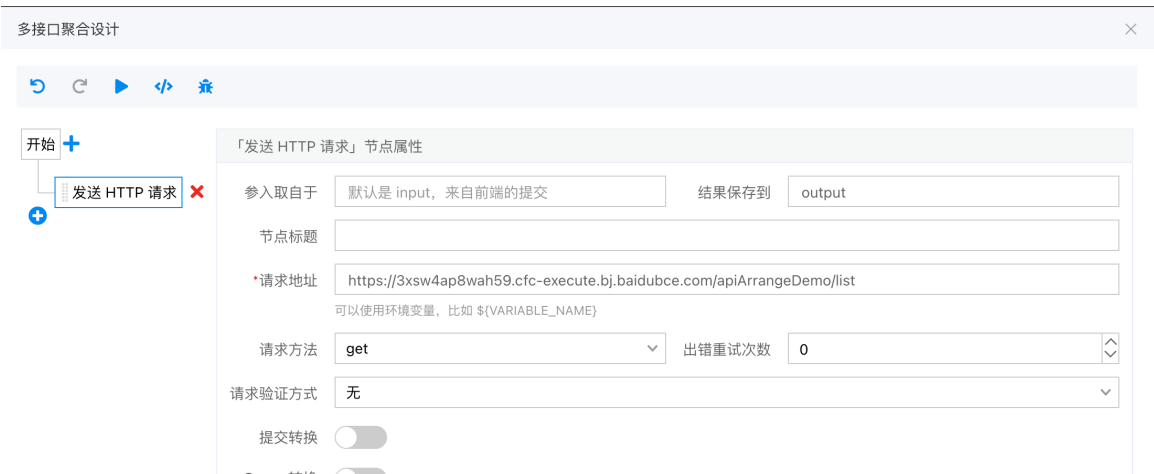


一个简单的例子

本例子将实现一个从服务端获取文章列表并对日期进行格式化后返回到前端的 API 编排

添加一个“发送 HTTP 请求”类型的节点，在“请求地址”输入框中填写以下 url

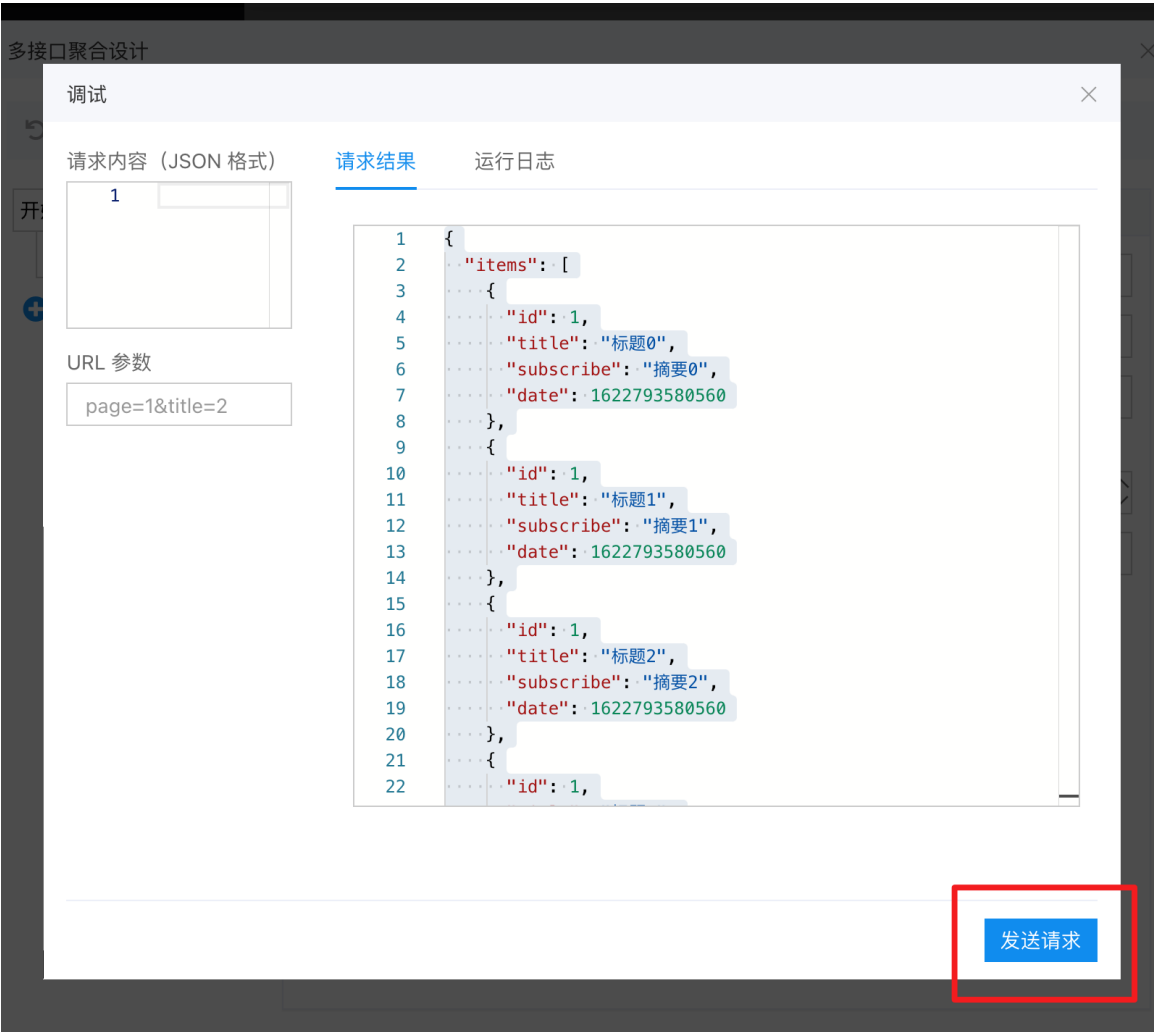
https://3xsw4ap8wah59.cfc-execute.bj.baidubce.com/apiArrangeDemo/list



点击工具栏中的“调试接口”按钮



在弹出框中点击右下角的“发送请求”按钮，等待请求执行完毕后，可以看到请求结果和运行日志



可以看到当前返回的数据是包含 items 和 total 两个字段的 json 格式数据，items 列表单项中的 date 字段是以毫秒时间戳格式返回的日期

```
{
  "items": [
    {
      "id": 1,
      "title": "标题0",
      "subscribe": "摘要0",
      "date": 1622793580560
    },
    ...
  ],
  "total": 200
}
```

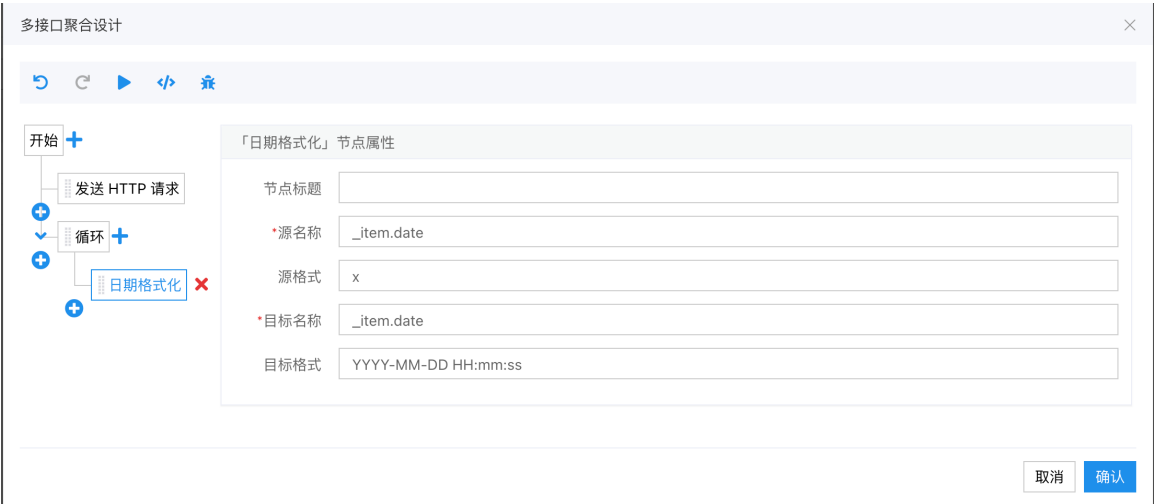
这里我们希望将 date 字段转换为能够友好展示的日期字符串

为“发送 HTTP 请求”节点添加一个平级“循环”类型的节点，并为“循环”节点添加一个“日期格式化”子节点

“循环”节点配置如下，接口返回的数据会被统一放在全局的 output 对象中，通过 output.xxx 可以取到相关字段



“日期格式化”节点配置如下，循环节点的行为是循环执行自身子节点的逻辑



点击发送请求，可以看到返回数据中的 date 被格式化成了 YYYY-MM-DD HH:mm:ss 格式

```
{
  "items": [
    {
      "id": 1,
      "title": "标题0",
      "subscribe": "摘要0",
      "date": "2021-06-04 17:23:41"
    },
    ...
  ],
  "total": 200
}
```

调试完之后需要点右下角的“确认”按钮保存，编辑界面中的内容才会被保存到数据库中 关于如何在页面中使用 API 编排获取数据，请看下一节

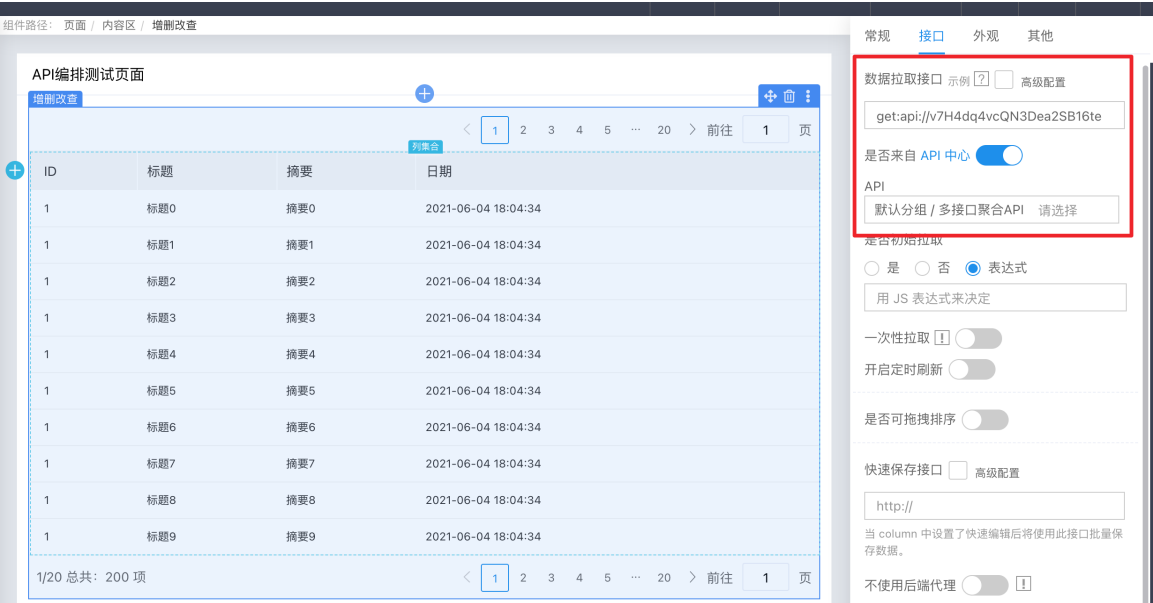
🔗 在页面中使用 API 编排

在页面中使用 API 编排的方式，和使用 API 中心 API 的方式完全相同，下面同样以一个简单的例子来说明

新建一个普通页面，在页面中拖入一个增删改查组件

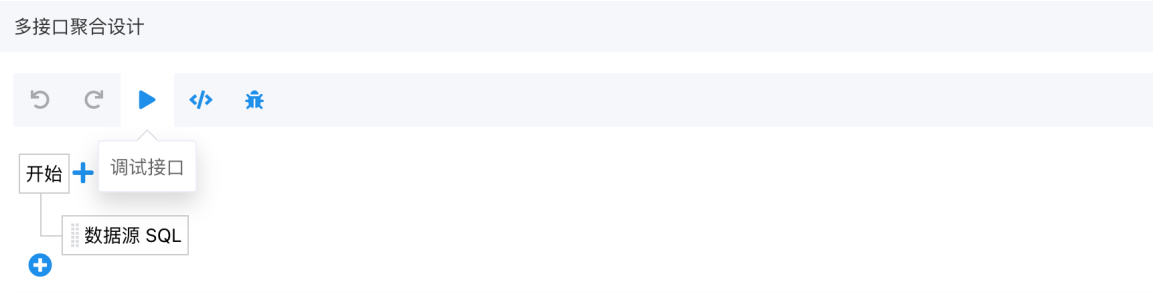
勾选增删改查组件的数据拉取接口的“来自 API 中心”选项，在列表中选择之前创建的 API 编排（和单个接口 API 在同一个列表中）

配置字段映射，可以看到 API 编排处理返回的数据被正常展示在列表中



调试接口

点击顶部的「调试接口」按钮进入接口调试功能。



在接口调试弹窗中，可以输入提交参数，然后点击「发送请求」进行测试，除了查看输出结果，最有用的功能是查看执行过程的记录，如下图所示：

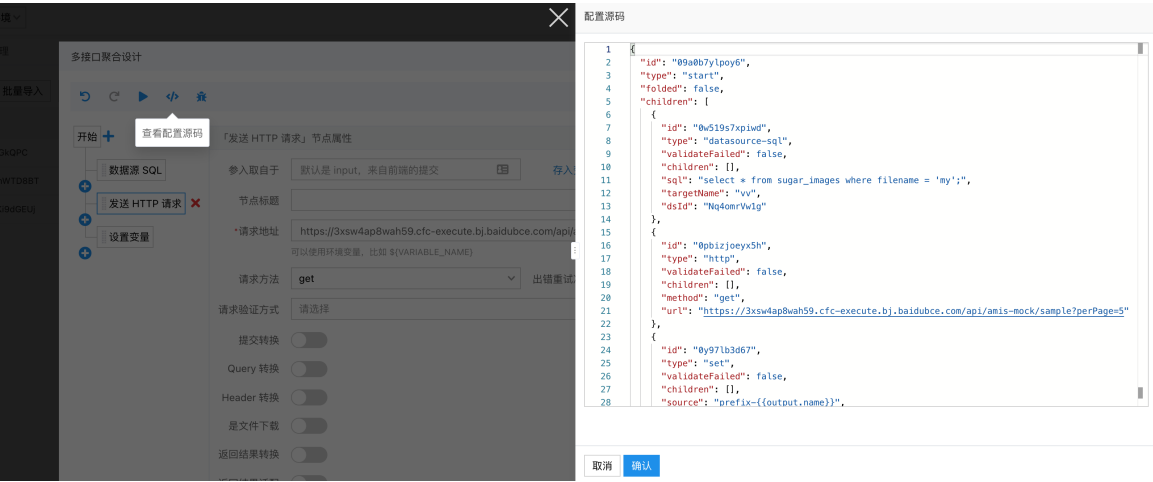


对于初始节点和有数据变更的节点，还能点击旁边的加号来查看当时的数据情况，这样就能知道每一步操作是否正确对数据进行了处理。



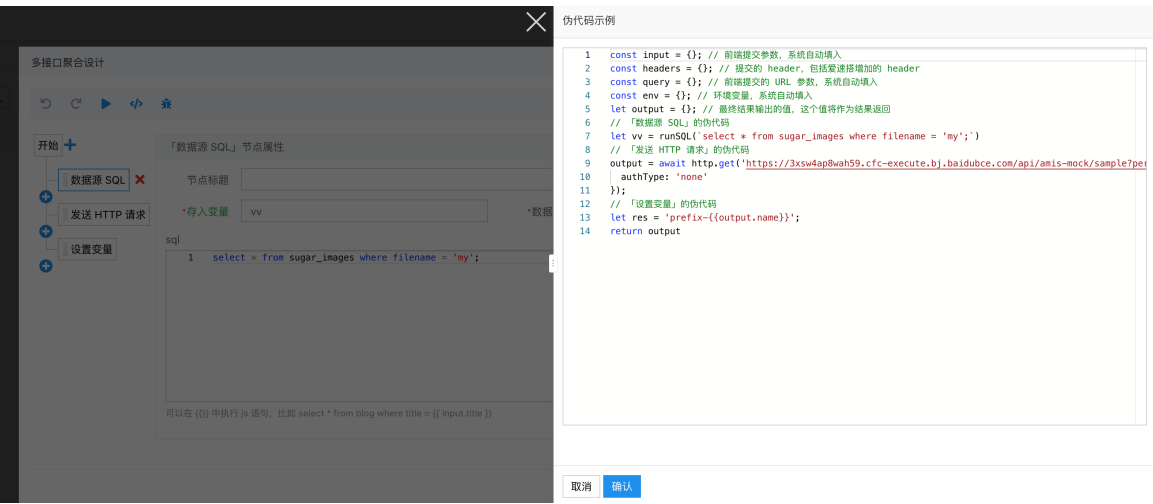
查看配置源码

点击顶部的「查看配置源码」可以看到最终生成的树形结构配置，这是 API 编排在实际运行是所使用的格式，这个功能主要是用于结果预期不一致时进行调试，可以将这个配置发给爱速搭的管理员协助分析。



生成的伪代码示例

点击顶部的「生成的伪代码示例」可以生成伪代码示例。



这个功能主要是给习惯看源码的研发，可以方便看出执行流程，不过需要注意的是爱速搭实际执行并不是使用这个生成的代码，因此这里仅供参考，输出的伪代码并不保证能运行。

节点分类介绍

接下来详细介绍各个节点的功能

数据接口

发送 HTTP 请求

最基本的 HTTP 请求发送节点，使用方法同【单个接口 API】

数据源 SQL

在数据源上执行 SQL 操作。

在 SQL 语句里用 {{}} 包裹的部分可以执行 JS 语句，实现灵活的处理，比如：

```
select * from blog where title = {{ input.title }}
```

它的实现原理是最终转成 `select * from blog where title = ?`，然后将 `input.title` 的值作为后续参数填入，因此默认会进行变量

转义，防止 sql 注入。

如果要实现有参数时才查询，可以用如下写法

```
select * from blog where ( {{ !input.title }} OR title = {{ input.title }} )
```

这个语句在输入参数有 title 的时候，会变成

```
select * from blog where (FALSE OR title = ?)
```

如果输入参数里没有 title，则会变成下面的语句，可以看到前面的条件是 true，所以 or 里的条件不会生效。

```
select * from blog where (TRUE OR title = ?)
```

## 🔗 执行控制

### 分支

相当于代码中的 if 语句。

### 循环

循环用于对数组进行遍历，它的每个子节点用于执行每次循环的中的语句。

### 跳出循环

用于跳出循环，不再执行循环的后续所有操作，相当于代码中的 break 语句。

### 继续循环

用于跳过当前循环的后续操作，进入下一个循环，相当于代码中的 continue 语句。

### 退出

用于退出整个接口，后续不再执行任何操作

## 🔗 并行/串行

### 顺序执行

按顺序执行子节点

### 并行执行

这个节点下的子节点会并行执行，用于独立获取各种数据的场景。

## 🔗 数据处理

### 设置变量

赋值语句，用于对变量进行赋值。

### 日期格式化

对日期格式进行转换，比如将时间戳转成可读的日期格式。

### JS 代码

通过 js 代码的方式来对数据进行处理，用于处理一些比较复杂的数据转换场景。

最简单的示例如下

```
module.exports = async function(event, state) {  
  return state; //  
}
```

其中 `state` 是 `api` 编排中的全局数据，比如前面如果有个 HTTP 节点将返回结果「存入变量」到 `data` 下，在 JS 代码中就能通过 `state.data` 获取到这个值，然后进行二次转换。

同时还能写入 `state.output` 来影响 `api` 聚合最终的输出结果。

这段 JS 函数的返回值会替换全局数据，因此上面的例子中直接返回 `state`。

## 其他

### 发送邮件

用于发送邮件，比如进行邮件通知

### session

有一个特殊的全局变量 `session`，它会在整个用户会话中存储，如果写入这个变量，下次再次请求的时候还能读取到。

这个变量的值可以是字符串，也可以是对象，但整体大小限制在 2k 字节。

`session` 是应用级别隔离，不同应用在不同环境下是独立的。

## 基础

普通页面支持 API 的方式对接外部接口，实现灵活扩展，API 详情请参考 [amis 文档](#)。

在爱速搭中有两个特殊功能：

## API 代理

普通页面中的 `api` 在最终渲染时会进行 url 替换，比如在页面设计中的配置中可能这样写的：

```
{  
  "api": "https://3xsw4ap8wah59.cfc-execute.bj.baidubce.com/api/amis-mock/sample"  
}
```

在实际运行中，爱速搭会将这个地址进行转换，实际在页面中会是类似下面的地址

```
{  
  "api": "/api/proxy/34fbe7"  
}
```

这样做有以下几个好处：

1. 解决前端跨域问题，因为第三方 `api` 肯定不在爱速搭域名下
2. 避免接口地址暴露
3. 传递一些爱速搭特有的信息，比如当前用户名、环境、应用 key 等

前面说到的特有信息可以通过下面这些 header 获取：

- `x-isuda-token`，在「应用设置」中的 token
- `x-isuda-appkey`，应用短路径
- `x-isuda-env`，应用环境，开发环境是 `dev`，也可能是 `qa`、`sandbox` 或者 `latest`



- `x-isuda-userid`，用户在爱速搭中的 id
- `x-isuda-username`，用户名
- `x-isuda-oauth-id`，oauth id，如果启用了 oauth 登录的话
- `x-isuda-oauth-access-token`，oauth 的 access token 如果启用了 oauth 登录的话
- `x-isuda-roles`，用户所属的角色名列表
- `x-isuda-profile-access-token` 获取用户信息的 access token，具体请看下方说明

如果不放心直接根据 header 下发的 `x-isuda-username` 完成用户认证，还可以通过环境变量关掉这个功能

```
ISUDA_DISABLE_PROXY_USER_ID: true
```

这样就只会下发以下 header 了。

- `x-isuda-token`
- `x-isuda-appkey`
- `x-isuda-env`
- `x-isuda-profile-access-token`
- `x-isuda-oauth-id`，oauth id，如果开启了 oauth 登录认证的话
- `x-isuda-oauth-access-token`，oauth 的 access token，如果开启了 oauth 登录认证的话

然后通过请求 `/openapi/profile?token={{这里用 x-isuda-profile-access-token 返回的 token 值}}` 就会获取以下信息。

```
{
  "username": "xxxx",
  "roles": ["xxx", "xxx"],
  "userid": "xxxx",
  "email": "xxx@xx.com"
}
```

#### 🔗 如果不想经过内置代理

可以在 url 前加上 `raw`，就不会走代理。

需要解决跨域问题，比如返回跨域 header：示例如下

- `Access-Control-Allow-Origin`: `https://suda.bce.baidu.com`，或者爱速搭部署的域名
- `Access-Control-Allow-Credentials`: `true`
- `Access-Control-Allow-Headers`: `Origin, X-Requested-With, Content-Type, Accept`
- `Access-Control-Allow-Methods`: `POST, GET, OPTIONS`

示例代码 (express)：

```
// 放在所有路由前面
app.use(function (req, res, next) {
  res.header('Access-Control-Allow-Origin', 'https://suda.bce.baidu.com');
  res.header('Access-Control-Allow-Credentials', 'true');
  res.header(
    'Access-Control-Allow-Headers',
    'Origin, X-Requested-With, Content-Type, Accept'
  );
  res.header('Access-Control-Allow-Methods', 'POST, GET, OPTIONS');
  next();
});
```

如果接口需要登录，可以在未登录的时候返回 401，内容是：

```
{
  "location": "http://跳转登录的页面地址"
}
```

需要注意在新浏览器下默认会限制 cookie 跨域，所以登录后保存状态的 cookie 必须加上 `SameSite=None; Secure`，类似如下

```
Set-Cookie: session=123; SameSite=None; Secure
```

如果想本地开发接口请查看[这里](#)。

🔗 在各种语言下获取参数的方法

🔗 Node

以 Express 为例

```
const express = require('express');
const app = express();
const bodyParser = require('body-parser');
app.use(bodyParser.json());
app.post('/api', function(req, res) {
  console.log(req.body); // 参数 json
  res.json(...);
})
app.listen(3000)
```

🔗 Java

以 Spring 为例

```
@RequestMapping(value = "/api", method = POST, consumes = MediaType.ALL_VALUE, produces =
MediaType.APPLICATION_JSON_UTF8_VALUE)
@ResponseBody
public String api(@RequestBody String body) {
  // body 就是字符串形式的 json，需要使用 Jackson 等库解析为对象来使用
}
```

🔗 PHP

```
<?PHP
$json = file_get_contents('php://input');
$data = json_decode($json);
```

## 🔗 .NET

```

using Microsoft.AspNetCore.Mvc;
using System.IO;
using Newtonsoft.Json;

namespace MvcMovie.Controllers
{
    public class HelloWorldController : Controller
    {
        [HttpPost]
        public JsonResult Index()
        {
            using (var reader = new StreamReader(HttpContext.Request.Body))
            {
                dynamic json = JsonConvert.DeserializeObject(reader.ReadToEnd());
                var conditions = json.conditions;
                var data = new { status = 0, msg = "" };
                return Json(data);
            }
        }
    }
}

```

## 🔗 go

```

func api(w http.ResponseWriter, r *http.Request) {
    body, _ := ioutil.ReadAll(r.Body)
    var result map[string]interface{} // 这里简化了，最好参考对应的参数说明文档创建 struct
    json.Unmarshal(body, &result)
}

```

## 🔗 Python

```

import json
#### falcon 框架
class SugarAPI(object):
    def on_post(self, req, resp):
        data = json.load(req.bounded_stream)

#### flask 框架
from flask import request
@app.route('/api', methods=['POST'])
def sugar_api():
    data = json.loads(request.data)

```

## 🔗 Ruby

```

require 'sinatra'
require 'json'

post '/api/' do
    request.body.rewind
    request_payload = JSON.parse request.body.read
end

```

## 本地开发调试

爱速搭中的 api 地址默认会经过代理，如果希望连本机地址，需要这两步操作：

1. 填入接口地址时需要在前面加上 raw:，比如 raw:http://localhost:8080/api
2. 这个接口需要返回跨域 header，包括如下这些
  - Access-Control-Allow-Origin: https://suda.bce.baidu.com，或者爱速搭部署的域名
  - Access-Control-Allow-Credentials: true
  - Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept
  - Access-Control-Allow-Methods: POST, GET, OPTIONS

如果本地接口不方便返回前面提到的跨域 header，可以使用第三方的工具代理来自动添加，下面介绍一个工具的使用方法：

1. 首先通过 `npm install -g local-cors-proxy` 命令安装，如果没有 `npm` 命令需要本机先安装 [Node](#)。
2. 使用 `lcp --proxyUrl http://localhost:8080 --proxyPartial '' --origin 'https://aisuda.bce.baidu.com' --credentials` 命令启动代理。其中 8080 是本地服务的端口，如果是私有部署，请将其中的 `https://aisuda.bce.baidu.com` 改成实际部署地址。
3. lcp 程序默认监听 8010 端口，因此在爱速搭中填入的是这个端口的地址，比如 `raw:http://localhost:8010/api`，后面的路径就是本地 8080 服务之前所使用的路径。

## 应用搭建

### 页面设计

#### 页面设计器介绍

#### 🔗 组件模板管理

##### 组件模板

爱速搭平台支持统一资源管理，其中就包含组件模板管理。目前在平台级别、组织级别、以及应用级别都可以独立维护组件模板，并且组件模板资源会通过继承机制，实现公共资源的下发和使用。

组件模板是爱速搭提供的基于预置系统组件自定义模板的能力，主要是为了实现降低相同场景重复配置的时间成本，同时也降低了一些复杂场景的配置门槛，一键拖拽即用。

一个组件模板可以包含一个组件或多个组件，组件的配置项也是基于系统组件已提供的能力，组件模板设计暂时不支持自定义扩展组件的类型和功能。

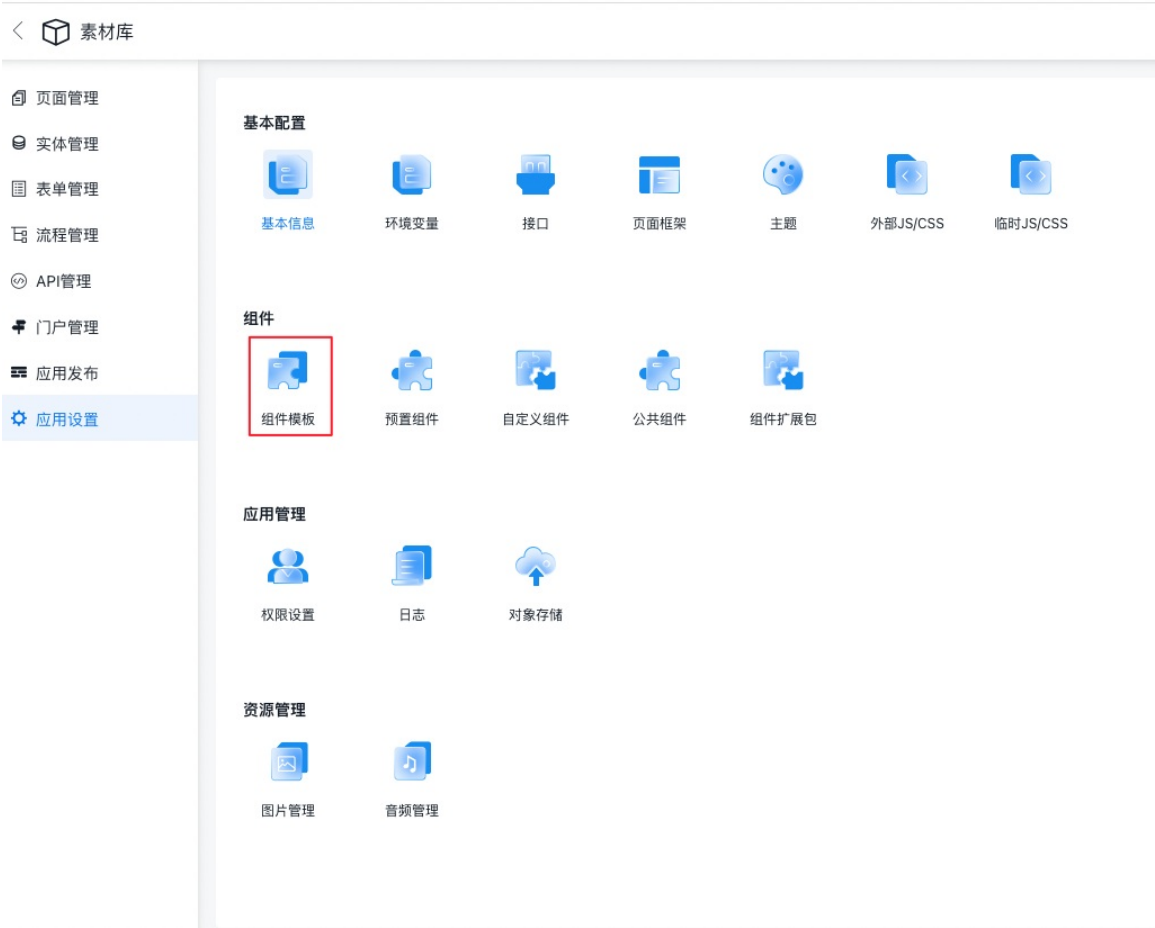
资源的管理最终是为资源的使用而服务，资源的使用始终是在开发环境下，因此组织级和应用级组件模板管理只在平台开发环境下才支持。

##### 层级管理

爱速搭目前在平台级别、组织级别、以及应用级别分别有不同的入口，来进行组件模板管理。不同层级不仅会展示当前层级创建的分类和上传的文件，也会展示继承而来的资源。需要注意的是，这里的资源继承是引用式继承，因为不支持对继承资源的新增、编辑和删除。

##### 1. 应用级

对于与应用或业务强相关，更重要的是不需要或不希望共享给其他项目使用的组件模板资源，建议在应用级进行维护，应用级组件模板管理的入口为：应用开发>应用设置>组件-组件模板。



不同类型的应用支持的应用设置的配置项不完全相同，示例为web应用的应用设置支持的配置项。对于其他类型的应用，入口的路径是一样的。

2. 组织级

对于需要在一个组织内共享、组织标准资源或者统一规范类的音频文件，建议在组织级进行维护。组织级音频管理功能的入口为：平台管理>组件模板

3. 平台级

平台级别的音频管理更多的是公开资源，适用面会比组织级更广，平台的所有组织都可以使用。适用于集团类的公司，或其他涉及多个组织的场景。平台级资源管理的入口为：超管后台-模板管理。

对于爱速搭私有化的客户来说，只有购买了多租户授权才能在平台上创建多个组织。

操作说明

不同层级的组件模板管理能力都是一样的，主要包括如下几点：

- 分类管理：分类的创建、编辑、删除
- 模板管理：模板的创建、设计、设置、删除等
- 批量处理：批量删除
- 筛选及查询：对模板列表进行筛选和检索

下面一应用级组件模板管理为例，详细介绍组件模板管理的具体功能和操作说明。

分类管理

1. 新增分类

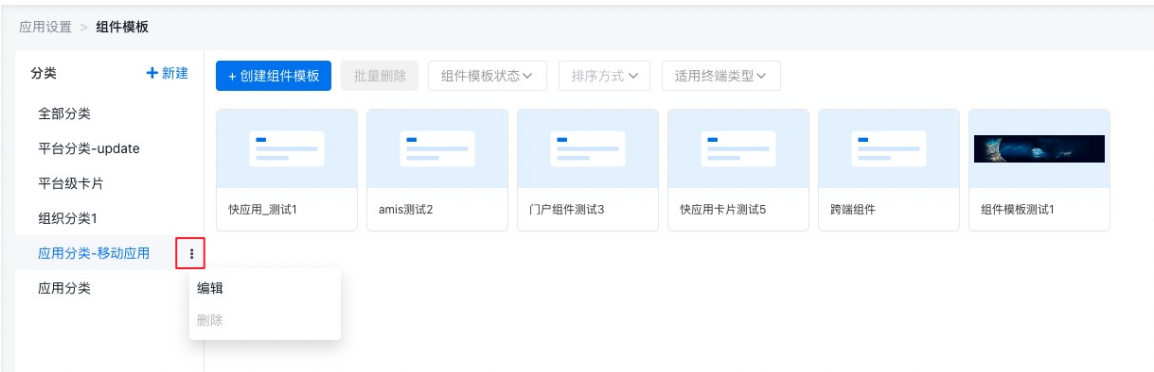
目前爱速搭的组件模板分类管理只支持一级分类，其中系统自带的「全部」为所有资源的全部集合，是个虚拟分类，无实际含义，在编辑器中选择模板时也无此分类。



目前分类名称的长度限制为8个字符，为提高分类名的可读性也不支持纯数字、不支持特殊字符，分类说明的最大长度为100个字符。

2. 编辑分类

如果需要对分类信息进行修改，可以通过分类的更多进入信息编辑页面。



3. 删除分类

如果需要对分类进行删除，默认的机制是只能删除分类下没有模板资源的分类。主要是因为组件模板的删除不可逆，且组件模板资源不可导出不利于备份。为了避免删除分类时将分类下的模板资源误删，所以不支持删除有模板资源的分类。

对于继承而来的分类，不支持编辑和删除，因此没有操作入口。

创建组件模板

爱速搭支持的组件模板类型及其说明如下表：

| 模板类型   | 终端         | 可使用的编辑器 |
|--------|------------|---------|
| amis组件 | PC/H5      | 普通页面设计器 |
| 门户组件   | PC/H5      | 小程序编辑器  |
| 跨端组件   | H5/小程序/APP | 小程序编辑器  |
| 快应用    | 快应用        | 快应用编辑器  |
| 快应用卡片  | 快应用卡片      | 快应用编辑器  |

如果对可使用的编辑器部分难以理解的话，建议先对爱速搭的页面设计器进行基本的了解。可以查看[页面设计简介](#)。

上传表单中，需要填写模板的名称并选择分类和适配终端。封面图为卡片展示组件模板时的封面图，非必填。组件状态决定了组件模板在编辑器中是否可见，启动状态的组件才会在编辑器的组件模板列表中显示，才可以被拖拽使用。

创建组件模板

\*名称

0/12

\*分类

请选择

\*适配终端

请选择

封面图

+

图片上传

推荐 140 \* 82尺寸（支持 1M 以内的图片）

状态

启用

说明

0/100

取消

确认

如果没有上传封面图，将使用系统默认的封面图进行展示。

组件模板操作

组件模板支持在线设计和设置信息、删除等操作，鼠标悬浮在组件模板上即显示。

应用设置 > 组件模板

分类

+ 新建

全部分类

平台分类-update

平台级卡片

组织分类1

应用布局模板

+ 创建组件模板

批量删除

组件模板状态

排序方式

适用终端类型

一栏两列

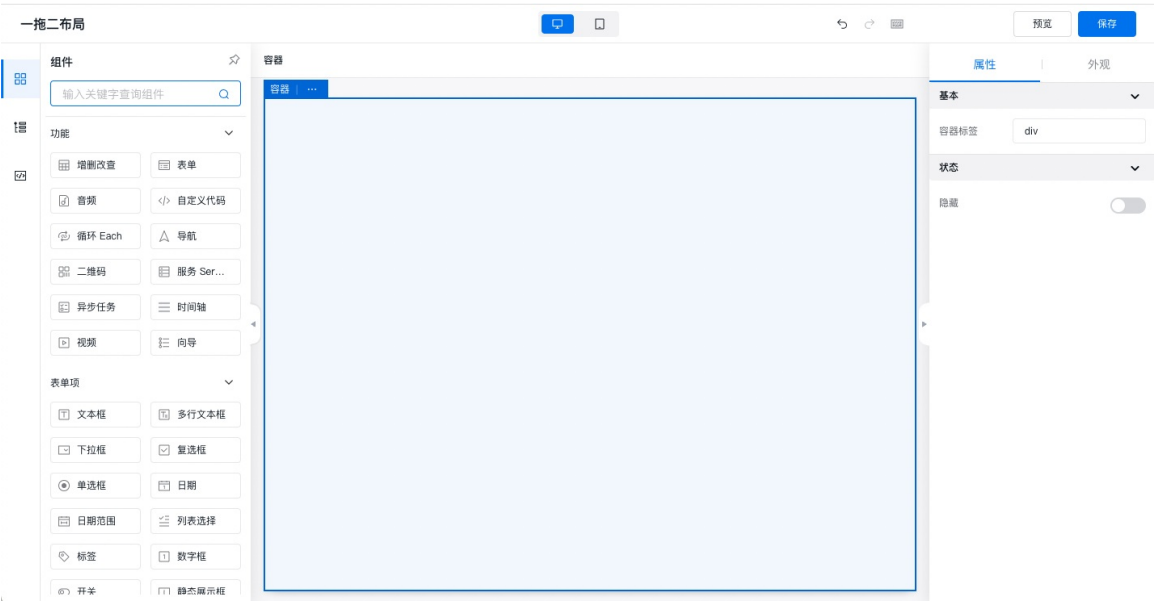
webp

一拖二布局

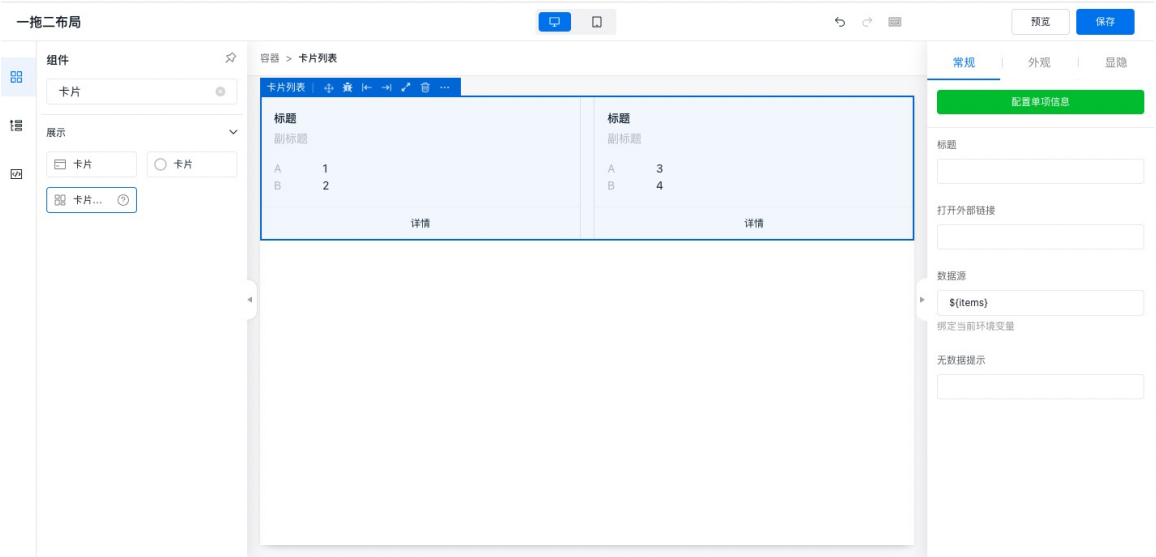
1、编辑

组件模板都是基于平台在线设计完成的，只能基于平台已有的组件，暂时不支持自定义扩展。模板默认外层为容器，不支持修改，可从左侧将所需的组件拖入，并完成配置。

201



如图是拖入一个卡片列表的效果，可以在画布中拖入任意组件，且同样支持编辑器自带的基础布局设计、层级设计等能力。



2. 设置

可以修改名称和分类信息，但不支持修改适配终端类型，并可以修改封面图、启用状态和说明信息。



设置

\*名称

一拖二布局

5/12

\*分类

应用布局模板

▼

适配终端

amis组件 (pc/h5)

适配终端类型不支持更改

封面图

推荐 140 \* 82尺寸 (支持 1M 以内的图片)

状态

启用

说明

0/100

取消

确认

3. 删除

点击删除后会进行二次确认，如需删除请点击确认，点击取消则取消该操作。

系统消息

确定要删除组件模板「一拖二布局」吗？

取消

确认

注意：删除操作不可逆，且平台无回收站功能，删除后将无法找回。在组件模板列表删除模板后，不影响已使用该模板的页面。

批量处理

默认批量操作的按钮是置灰的，当选中某个文件后，激活批量操作的按钮。单个卡片的选中热区为卡片操作入口icon之外的区域。



筛选及排序

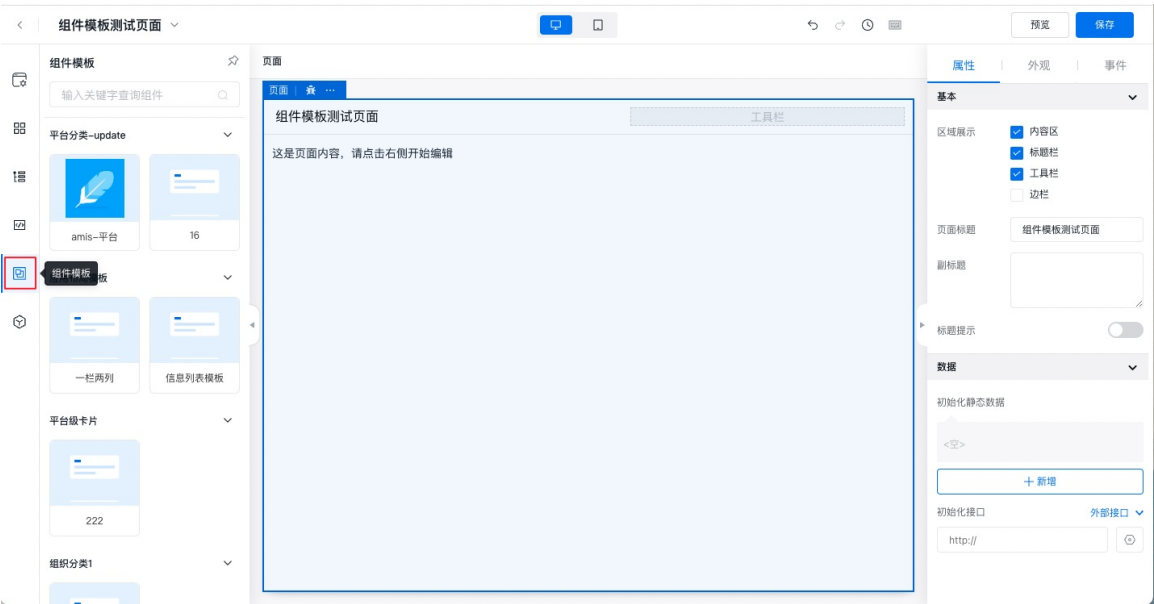
支持按启动状态筛选，按文件名称、创建时间、更新时间进行排序，按适用终端类型进行筛选，以及通过组件名称模糊搜索。



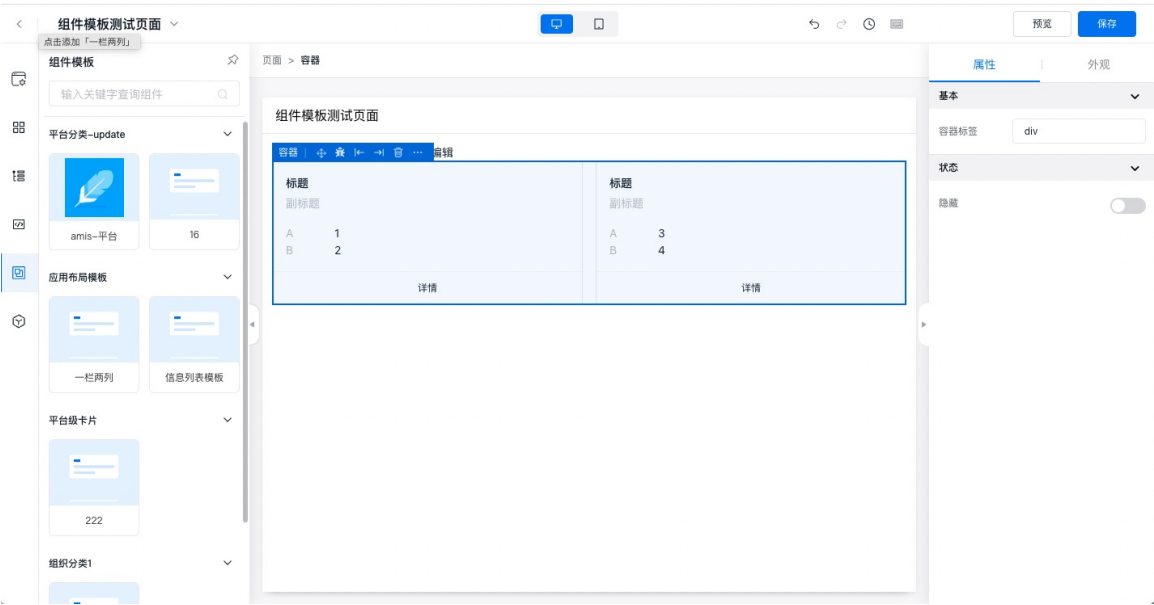
组件模板使用

组件模板在设计器中的使用是复制的关系，使用组件模板后，会将组件模板的JSON写入到页面中，因此更新组件模板或者删除组件模板也不会对已使用的页面产生影响。如果需要实现关联修改的效果，可使用编辑器panel中的「公共组件」。

1. 进入页面设计器中，点击左侧panel的组件模板，可以看到可使用的组件模板。



2. 点击任意组件模板或将组件模板拖入画布，即完成了组件模板的添加。组件模板默认用容器组件包裹，可对内部元素在进行个性化配置。



- 注意：
- 1. 页面设计器中只会将平台上适用于该类设计器、为启用状态且模板内容不为空的组件模板都按照分类名展示出来。
  - 2. 分类会随着组件模板更新时间调整顺序，优先展示最新更新过的。
  - 3. 不同设计器左侧panel的功能入口数量不一样，组件模板的入口所处的顺序可能有所差异。

## 页面设计简介

### 页面设计简介

#### 功能定位

在爱速搭上，应用即一个APP，是包含完整的业务逻辑和前端交互界面的可运行应用程序。类似于常见的前后端分离的开发模式，在爱速搭上也提供了前后端低耦合的开发模式，前端模块为一系列用户界面，后端模块为各种业务逻辑，用户界面设计的过程中完成和后端逻辑的对接，从而完成整个应用的开发。

页面设计是爱速搭的核心能力之一，解决的就是应用开发过程中用户界面的可视化开发。但爱速搭提供的页面设计能力不局限于对单个页面进行样式设计，同时还能可视化实现与后端能力的快速对接，以及配置应用内不同页面之间的前端交互逻辑。

#### 页面场景

目前爱速搭已经支持多种页面类型场景，包括常见的管理类页面、表单类页面、展示类页面、Markdown页面、营销类页面，在端的覆盖度上也支持了PC、H5、小程序、APP等多种终端类型。对于不同终端、不同场景的页面类型，爱速搭也对页面设计器做了差异化处理，以下是爱速搭三种页面设计器支持的终端类型：

| 页面设计器         | PC | H5 | 小程序 |
|---------------|----|----|-----|
| 普通页面设计器       | √  | √  |     |
| 小程序页面设计器      |    | √  | √   |
| Markdown页面设计器 | √  | √  |     |

- 1. APP不是平台直接支持的终端类型，但是可以基于小程序页面设计器开发的H5导出后进行打包生成APP
- 2. 平台即将上线快应用设计器，用于快应用及快应用卡片设计，敬请期待

如下是不同的页面设计器在不同类型页面开发时的推荐选择：

| 页面设计器         | 管理类页面 | 表单类页面 | 展示类页面 | Markdown页面 |
|---------------|-------|-------|-------|------------|
| 普通页面设计器       | √     | √     |       |            |
| 小程序页面设计器      |       | √     | √     |            |
| Markdown页面设计器 |       |       |       | √          |

对于PC端的展示页面，也可以通过小程序设计器，在web应用中，可以通过创建展示页面，使用小程序编辑器，来实现展示类页面的可视化开发。

页面设计流程

在爱速搭上进行页面设计，首先要完成应用类型的选择，目前应用有web应用和小程序应用两大类，在 [页面场景](#) 中有详细介绍不同终端、不同场景的情况下如何选择设计器。完成应用类型的选择以及应用创建后，页面设计大致的流程包括几步：

- 1. 按需创建一个页面，完成基础信息的填写。

创建页面

\* 页面终端

PC端

移动端

默认响应式页面。

\* 页面类型

普通页面

使用外部 API，实现各种表单及表格页面。

\* 页面名称

请输入页面名称

父级页面

默认为首页

页面 Key

留空将使用页面名称

应用在地址栏的目录，后续后修改

菜单名称

导航中显示的名称，留空将采用标题作为菜单名称

描述

请输入描述

菜单标识

☒ 图片

☐ 图标



普通页面编辑器支持的页面类型较多，且对PC端页面支持H5自适应,但如果选择了创建的是H5页面则不支持自动适配PC端。因此建议在创建页面前先确定需要创建什么类型的页面。

2. 进入页面编辑状态，进行页面设计

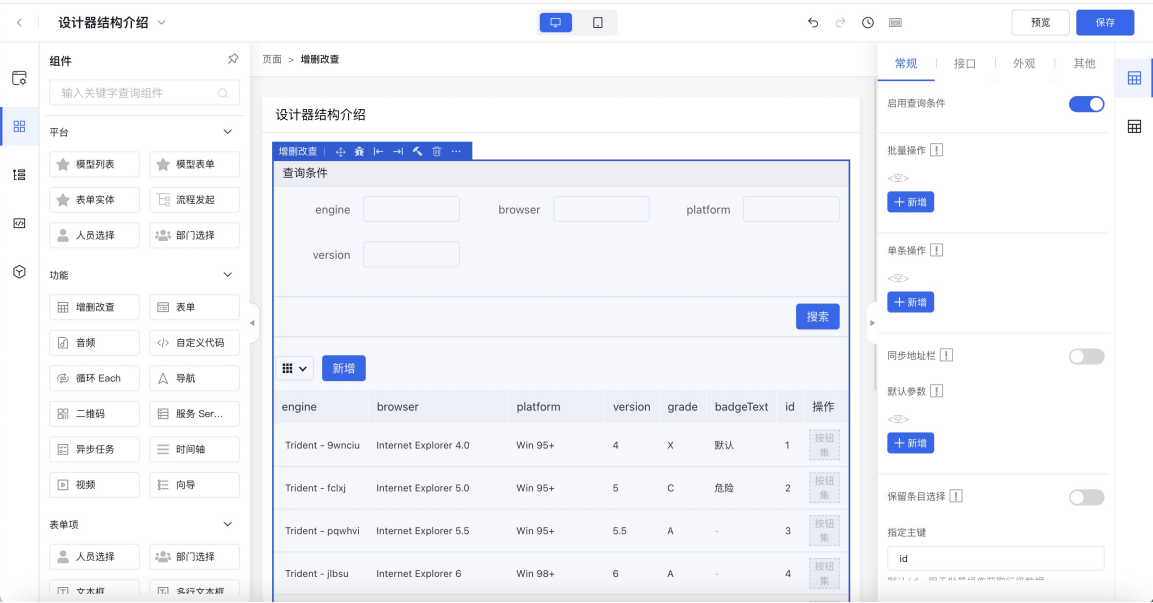
所有设计器的布局都分为功能面板、顶部工具栏、配置面板、内容区（画布区）四大部分，下面以普通页面设计器为例进行介绍。



- 功能面板：页面设计器的主要功能区，不同的设计器功能上会有差异，主要提供页面参数配置、组件列表、页面大纲、页面配置代码、组件模板和公共组件等功能，小程序设计器还支持数据管理能力。
- 顶部工具栏：编辑器的常见操作如编辑的撤销、历史，页面的预览和保存，页面的退出等。
- 配置面板：画布区选中元素的配置项面板，可对选中的组件/元素进行详细的配置
- 内容区（画布区）：页面设计的效果区域，可拖入组件或调整结构等，可以实现每一步操作的所见即所得

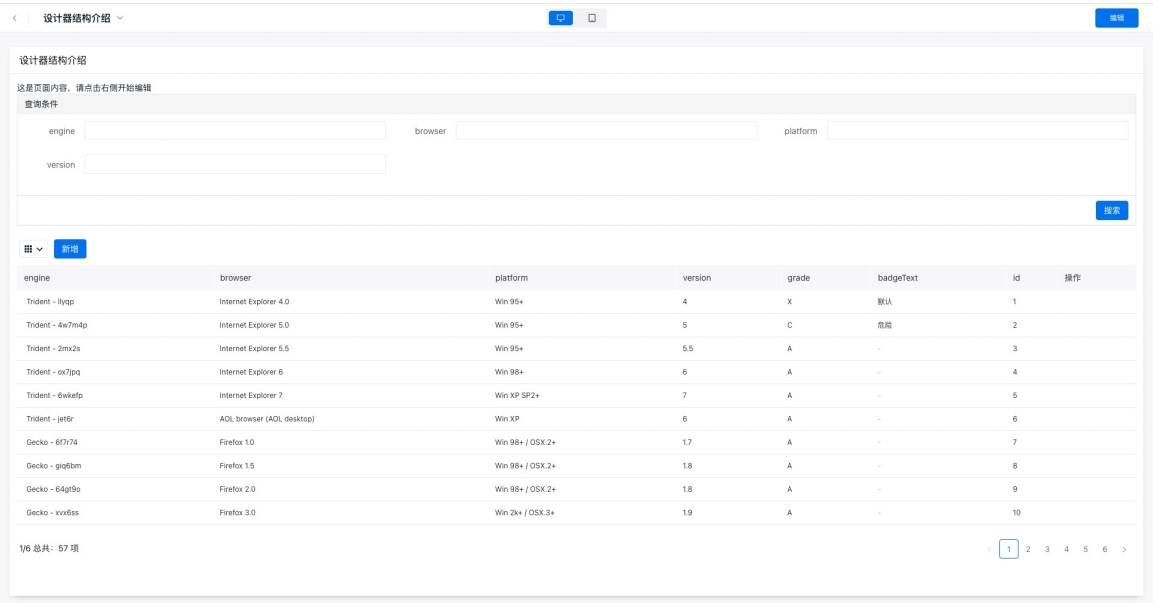
3. 拖入所需的组件并完成组件配置

如图是拖入一个增删改查组件并进行简单配置后的效果，在右侧的属性面板还可以进行更详细的配置。修改配置后会在画布中实时生效。



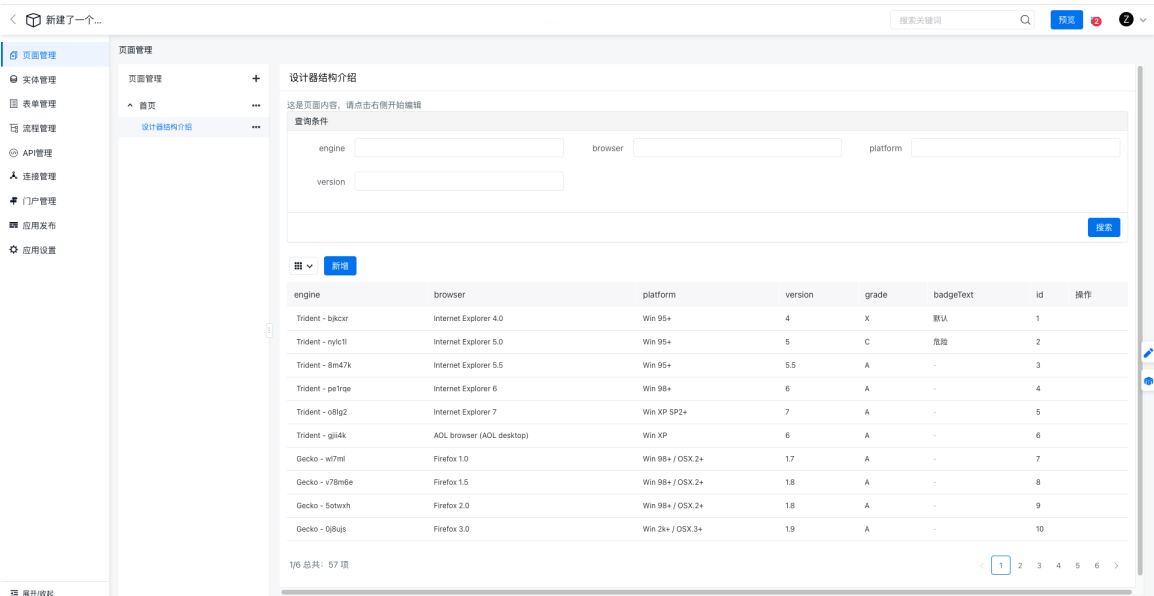
4. 完成画布区组件的配置后可点击顶部工作栏的「预览」进行效果预览。

如图是对上一步操作的页面进行预览的效果，如果预览效果不满足需求可以点击「编辑」继续编辑，如果效果符合需求，点击「编辑」后返回设计态，点击「保存」将页面配置保存下来，并生成一个页面历史版本。



5. 编辑好的页面可通过顶部工具栏左侧进行返回，或点击页面名称右侧的下拉icon，打开页面目录，切换至其他页面或新增下一个要编辑的页面。

退出后返回页面管理列表，效果如下图所示，如需重新编辑，可点击页面右侧中部的编辑icon。



操作完尽量退出设计器，如不退出设计器，则默认占有当前页面编辑权限，其他协作用户无法对当前页面进行编辑。

## 应用发布

### 版本管理

## 1. 简介

### 1.1 功能简介

应用发布后，可以根据发布的版本号，对已发布版本进行管理。实现历史版本查看，版本回退，版本对比等功能。

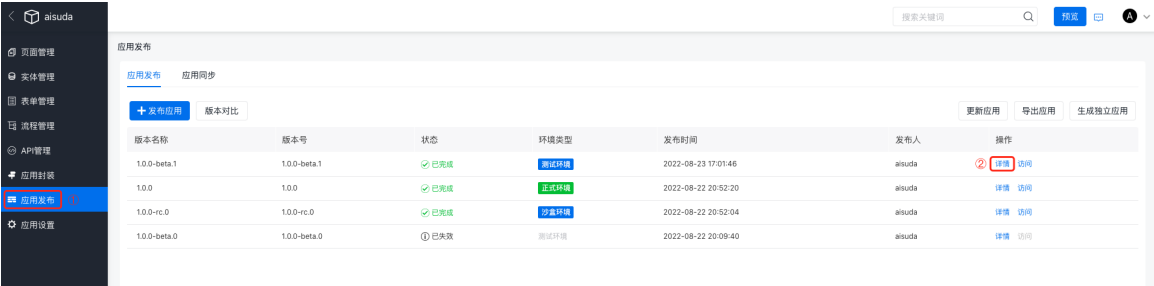
### 1.2 应用场景

当发布版本存在问题时，通过查看详情来查看发布详情。通过『版本对比』功能比较不同版本间的差异，通过指定历史版本号来实现版本回滚。

## 2. 操作步骤

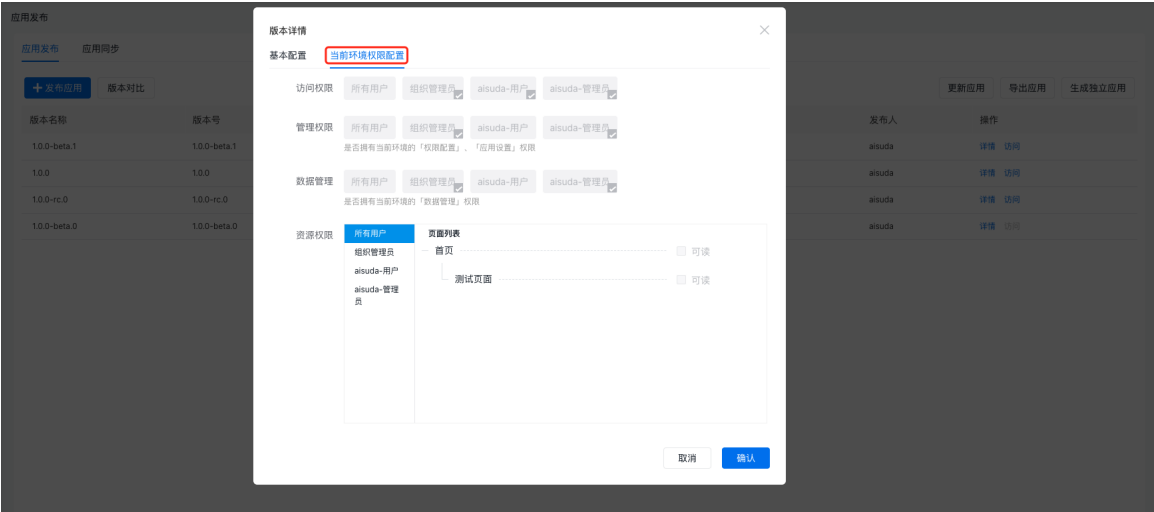
### 2.1 查看详情

点击『应用发布』，选择需要查看的版本，点击『详情』按钮。



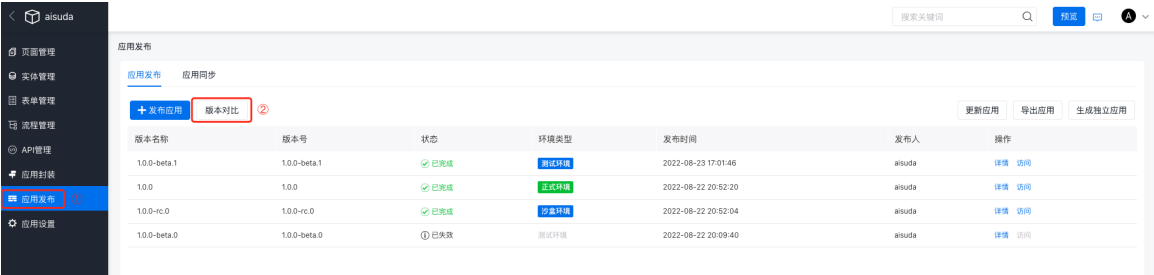


点击『当前环境权限配置』，查看该版本的权限配置详情



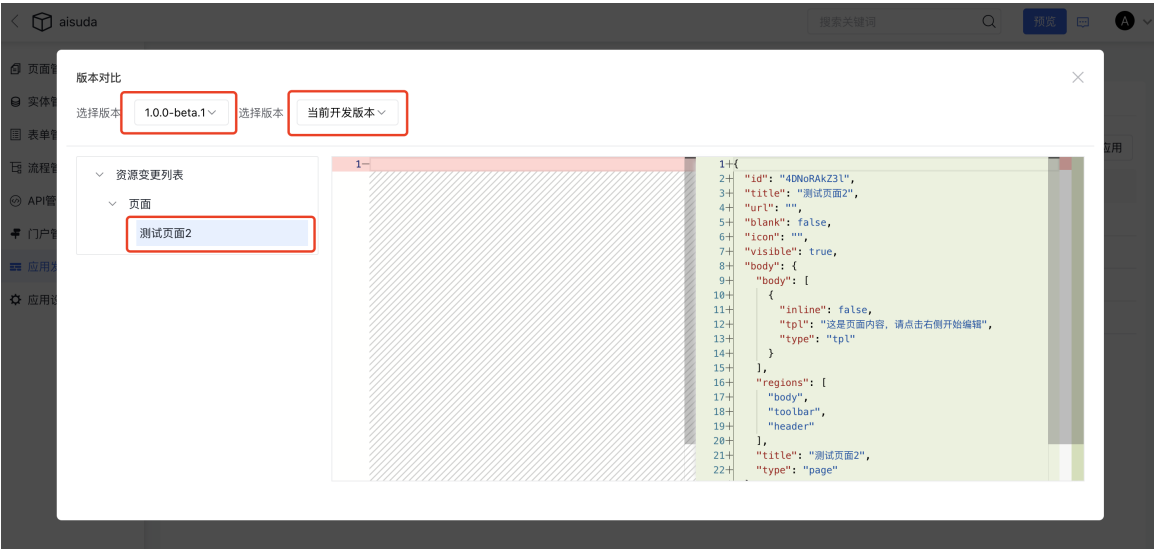
2.2 版本对比

点击『应用发布』，点击『版本对比』按钮



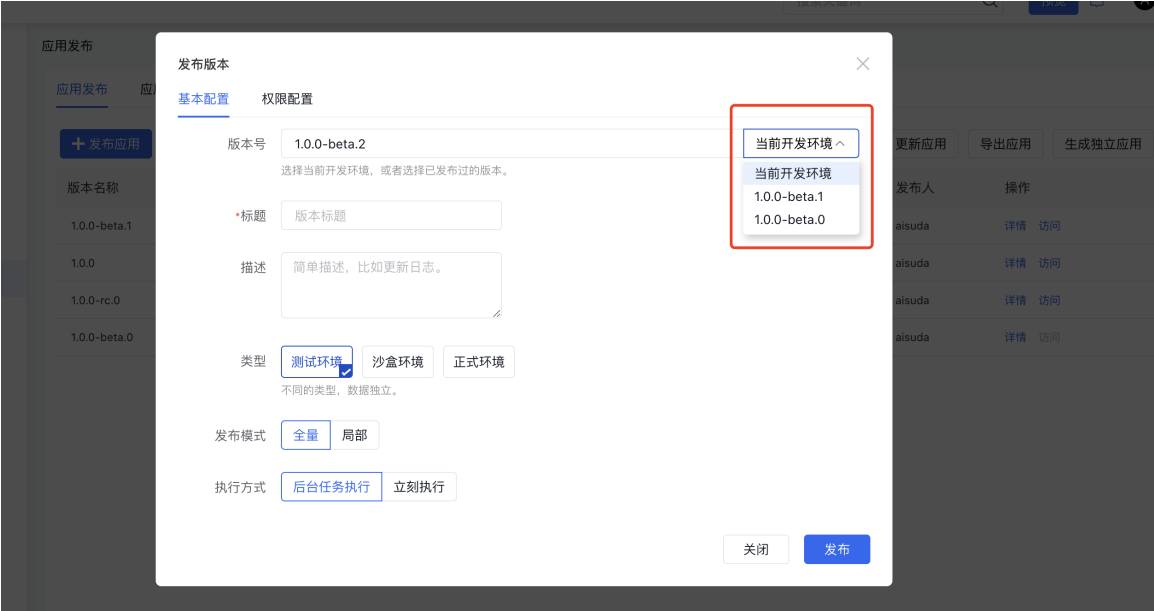
选择需要对比的两个版本，选择有变动的页面，可以看到两个版本页面的差异。





## 2.3 版本回滚

点击『应用发布』，点击『发布应用』，打开下列弹窗



选择需要回滚到的版本，点击『发布』



发布完成，回滚到指定版本

应用发布应用同步

+ 发布应用

版本对比

更新应用

导出应用

生成独立应用

| 版本名称         | 版本号          | 状态    | 环境类型 | 发布时间                | 发布人    | 操作   |
|--------------|--------------|-------|------|---------------------|--------|------|
| 1.0.0-beta.2 | 1.0.0-beta.2 | ✅ 已完成 | 测试环境 | 2022-09-07 20:54:35 | aisuda | 详情访问 |
| 1.0.0-beta.1 | 1.0.0-beta.1 | ❶ 已失效 | 测试环境 | 2022-08-23 17:01:46 | aisuda | 详情访问 |
| 1.0.0        | 1.0.0        | ✅ 已完成 | 正式环境 | 2022-08-22 20:52:20 | aisuda | 详情访问 |
| 1.0.0-rc.0   | 1.0.0-rc.0   | ✅ 已完成 | 沙盒环境 | 2022-08-22 20:52:04 | aisuda | 详情访问 |
| 1.0.0-beta.0 | 1.0.0-beta.0 | ❶ 已失效 | 测试环境 | 2022-08-22 20:09:40 | aisuda | 详情访问 |

多环境

1. 功能简介

应用发布时可以配置环境，支持线上环境、沙盒环境以及测试环境的发布。

应用发布应用同步

+ 发布应用

版本对比

更新应用

导出应用

生成独立应用

| 版本名称         | 版本号          | 状态    | 环境类型 | 发布时间                | 发布人    | 操作   |
|--------------|--------------|-------|------|---------------------|--------|------|
| 1.0.0-beta.2 | 1.0.0-beta.2 | ✅ 已完成 | 测试环境 | 2022-09-07 20:54:35 | aisuda | 详情访问 |
| 1.0.0-beta.1 | 1.0.0-beta.1 | ❶ 已失效 | 测试环境 | 2022-08-23 17:01:46 | aisuda | 详情访问 |
| 1.0.0        | 1.0.0        | ✅ 已完成 | 正式环境 | 2022-08-22 20:52:20 | aisuda | 详情访问 |
| 1.0.0-rc.0   | 1.0.0-rc.0   | ✅ 已完成 | 沙盒环境 | 2022-08-22 20:52:04 | aisuda | 详情访问 |
| 1.0.0-beta.0 | 1.0.0-beta.0 | ❶ 已失效 | 测试环境 | 2022-08-22 20:09:40 | aisuda | 详情访问 |

发布版本

基本配置

权限配置

版本号

1.0.0-beta.3

当前开发环境

标题

版本标题

描述

简单描述，比如更新日志。

类型

测试环境

沙盒环境

正式环境

发布模式

全量

局部

执行方式

后台任务执行

立刻执行

关闭

发布

2. 应用场景

开发环境配置完应用，可以先将应用发布在测试环境验证，再不同环境间数据、版本、环境变量不互通。

3. 预期效果

同一应用的不同环境独立发布，独立运行，互不干扰。

应用发布应用同步

+ 发布应用

版本对比

更新应用

导出应用

生成独立应用

| 版本名称         | 版本号          | 状态    | 环境类型 | 发布时间                | 发布人    | 操作   |
|--------------|--------------|-------|------|---------------------|--------|------|
| 1.0.0-beta.2 | 1.0.0-beta.2 | ✅ 已完成 | 测试环境 | 2022-09-07 20:54:35 | aisuda | 详情访问 |
| 1.0.0-beta.1 | 1.0.0-beta.1 | ❶ 已失效 | 测试环境 | 2022-08-23 17:01:46 | aisuda | 详情访问 |
| 1.0.0        | 1.0.0        | ✅ 已完成 | 正式环境 | 2022-08-22 20:52:20 | aisuda | 详情访问 |
| 1.0.0-rc.0   | 1.0.0-rc.0   | ✅ 已完成 | 沙盒环境 | 2022-08-22 20:52:04 | aisuda | 详情访问 |
| 1.0.0-beta.0 | 1.0.0-beta.0 | ❶ 已失效 | 测试环境 | 2022-08-22 20:09:40 | aisuda | 详情访问 |

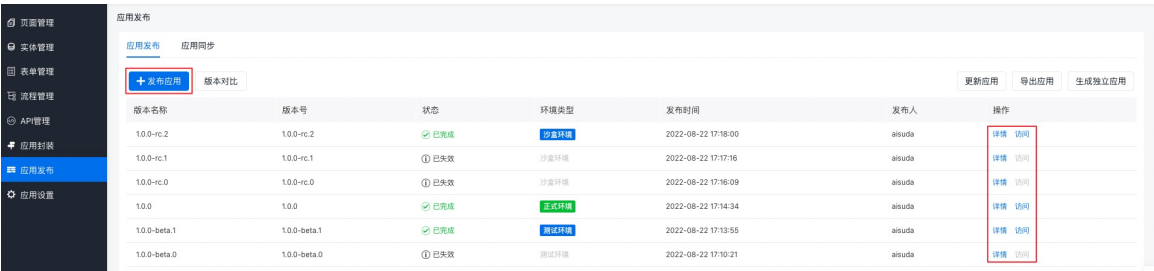
应用发布

1. 概述

爱速搭支持强大的版本管理能力，可以轻松实现版本迭代和发布控制。自带版本发布详情，可快速溯源，可快速回滚。应用通过发布完成上线流程。

编辑完成的应用，可以通过『应用发布』进行发布，完成上线流程。已发布版本可以快速访问和查看详情，版本发布后不支持版本信息修改。

应用发布时可以配置环境、权限、环境变量等，支持线上版本、沙盒版本以及测试版本，并可以轻松实现版本控制。



## 2. 操作步骤

1. 点击「应用发布 >> 发布应用」：



2. 填写版本标题，版本描述，发布环境。点击「发布」发布版本。



3. 发布成功后，可以通过详情按钮查看本次版本的基本信息和变更日志





# 对象存储

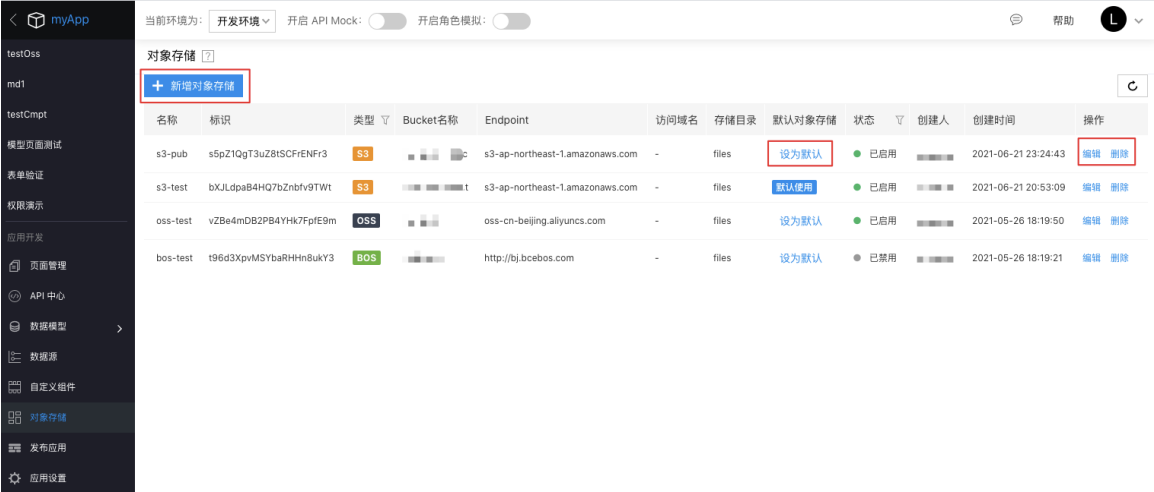
## 对象存储管理

### 对象存储管理

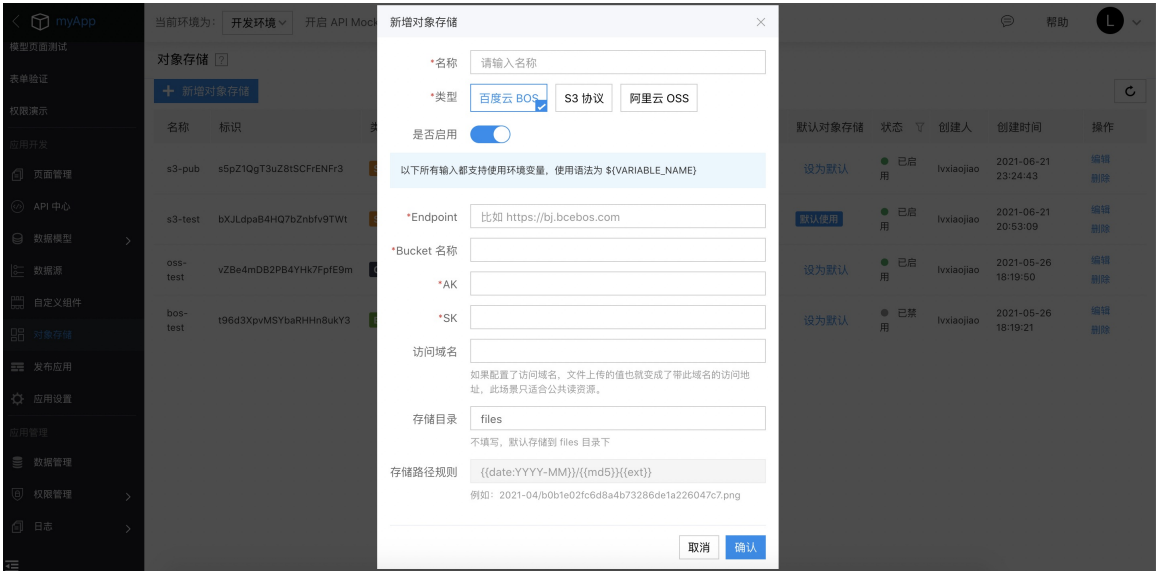
想要实现应用之间上传文件存储位置的隔离？可以尝试在应用下添加对象存储资源，并在应用下相应的上传配置中选择想要的对象存储资源即可。

### 编辑对象存储

可以在应用开发的【对象存储】页面进行对象存储资源的新增、修改、删除、设置默认对象存储、启用/禁用操作。



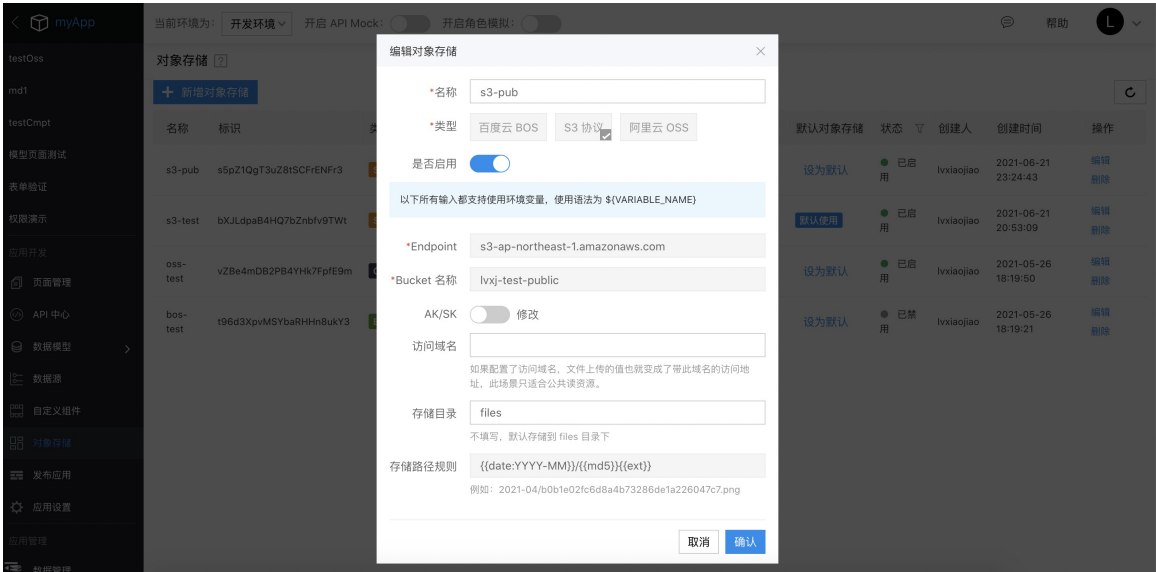
### 新增对象存储



新增对象存储时可以配置如下信息：

- 名称：取一个好记的对象存储资源的名称吧
- 类型：支持百度云 BOS、S3 协议、阿里云 OSS
- 是否启用：可以设置该对象存储资源是否生效
- Endpoint：对象存储资源的Endpoint，例如：智能云 BOS <https://bj.bcebos.com>
- Bucket 名称：对象存储资源的桶名称
- AK/SK：对象存储资源的AK、SK，AK/SK 会加密存储，用户不用担心安全性问题
- 访问域名：如果配置了访问域名，文件上传的值也就变成了带此域名的访问地址，此场景只适合公共读资源
- 存储目录：用来指定存储的位置，不填写，默认存储到 files 目录下
- 存储路径规则：默认为{{date:YYYY-MM}}/{{md5}}{{ext}}，例如：2021-04/b0b1e02fc6d8a4b73286de1a226047c7.png，当前不支持修改

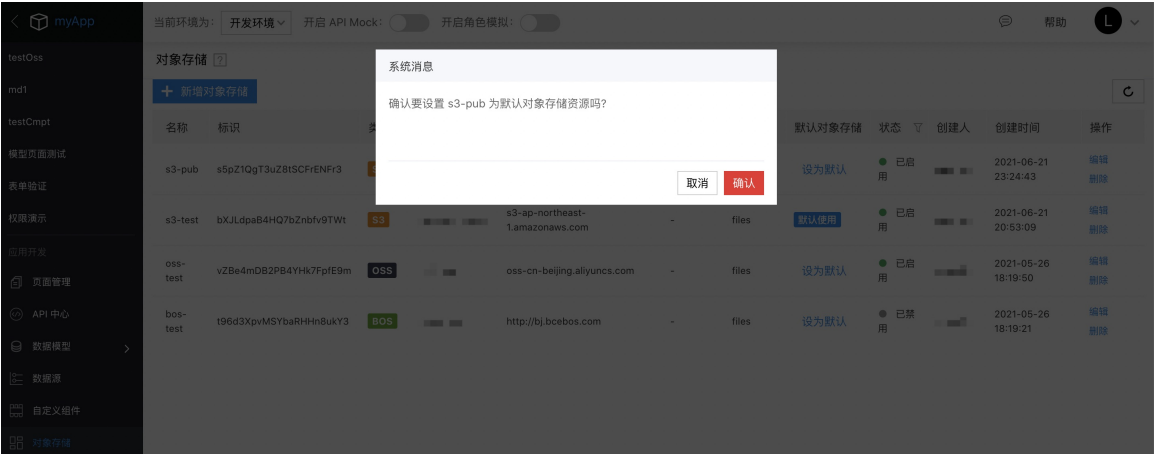
编辑对象存储



如果想修改 AK/SK，可以打开AK/SK右侧的开关进行修改。

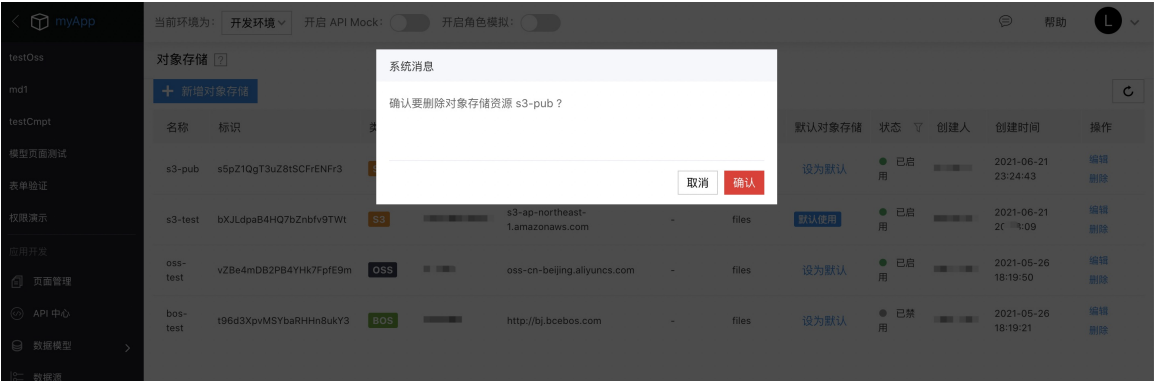
设置默认对象存储

一般新增的首个对象存储会自动设置为默认对象存储，默认对象存储会作为文件上传、图片上传等静态资源上传存储的默认存储位置。



删除对象存储

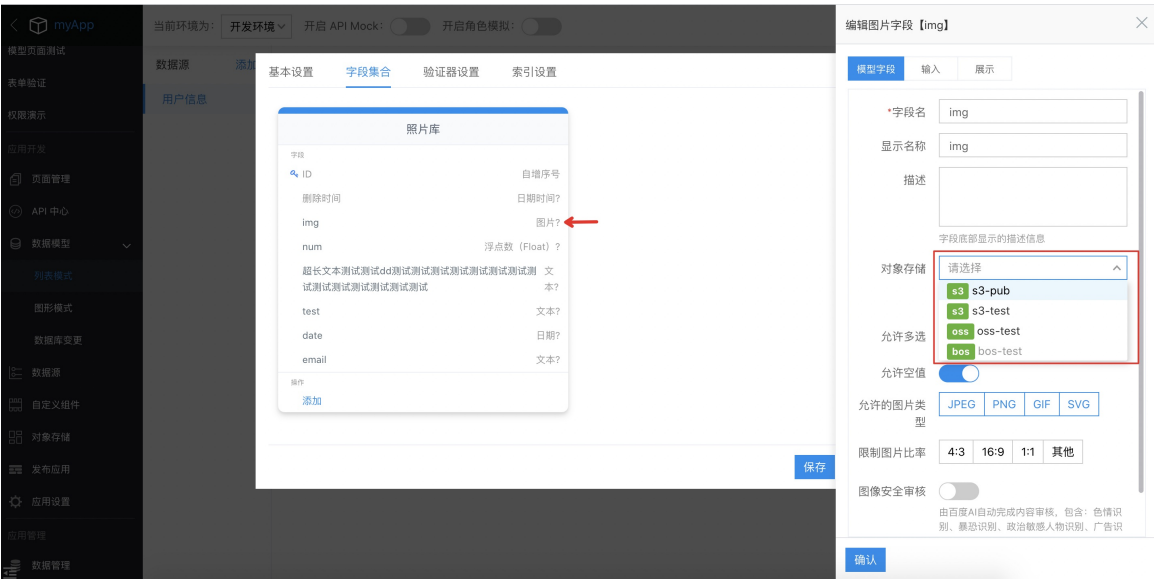
删除对象存储的时候，将检测该对象存储是否存在上传记录，如果存在上传记录将不允许被删除。



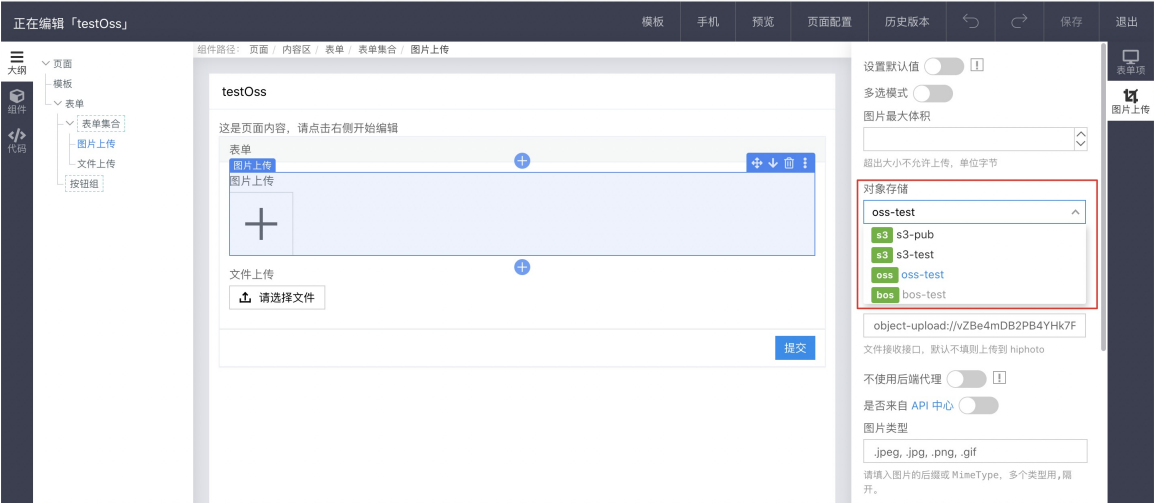
配置资源上传

爱速搭平台中，可以为具备资源上传的页面组件和数据模型字段配置对象存储。

数据模型



页面组件



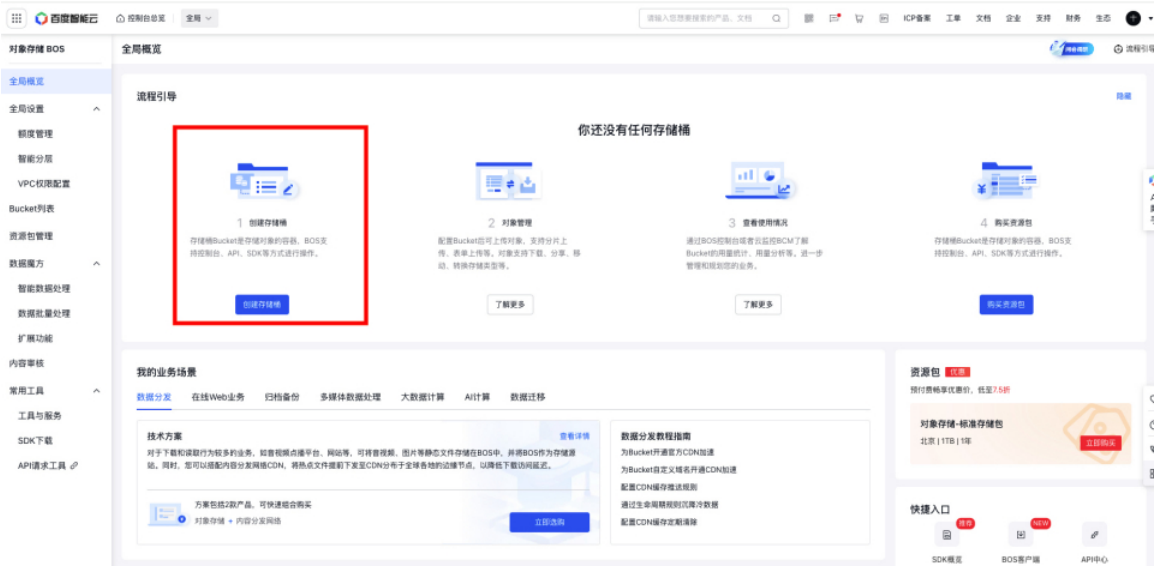
配置对象存储

如何配置相应aksk

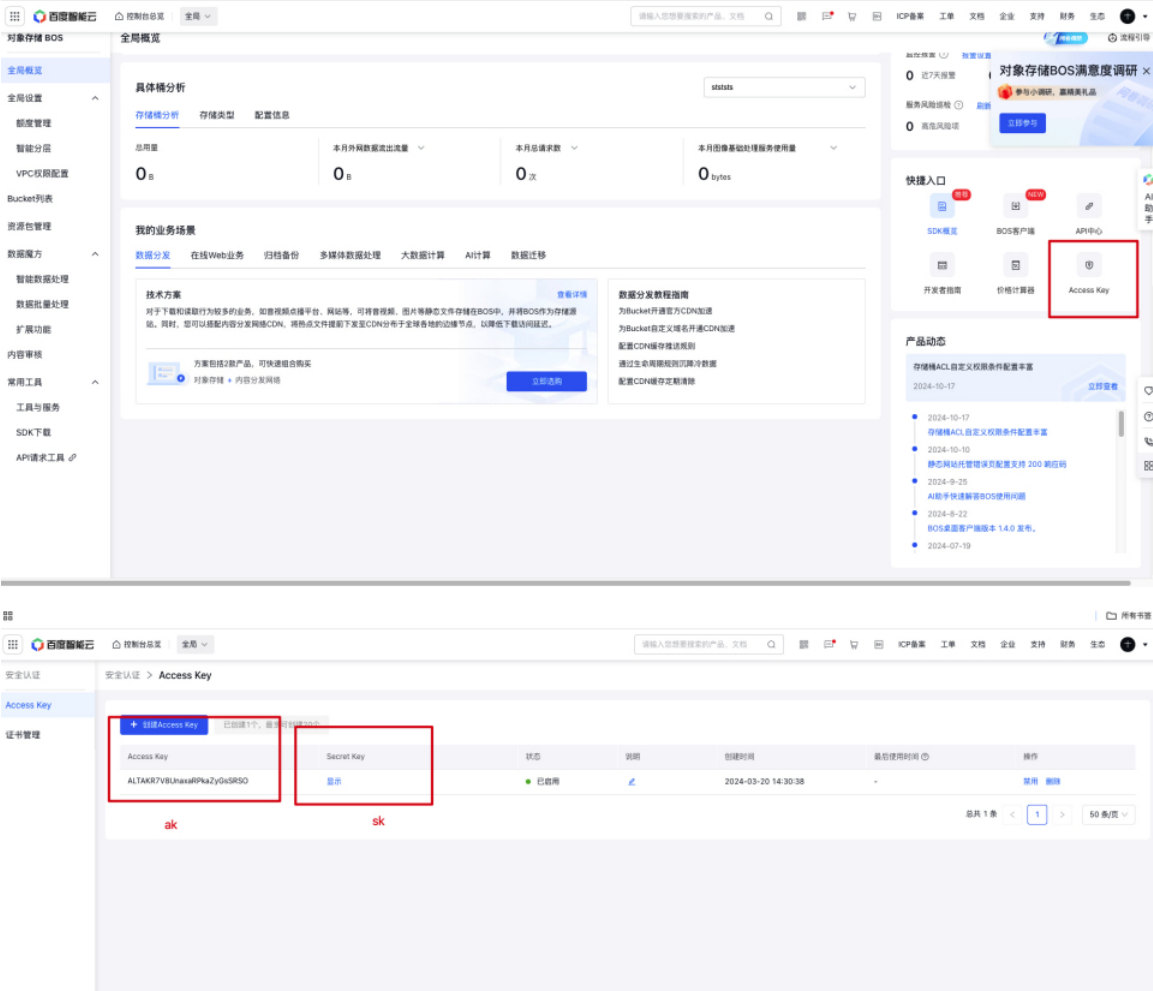
系统中用到文件上传、图片上传功能时，需要为系统配置相应的对象存储。  
类型：支持百度云BOS、S3协议、阿里云OSS

获取百度云BOS的aksk

- 登录[管理控制台](#)，点击“用户账号->安全认证”进入Access Key管理界面，创建存储桶。



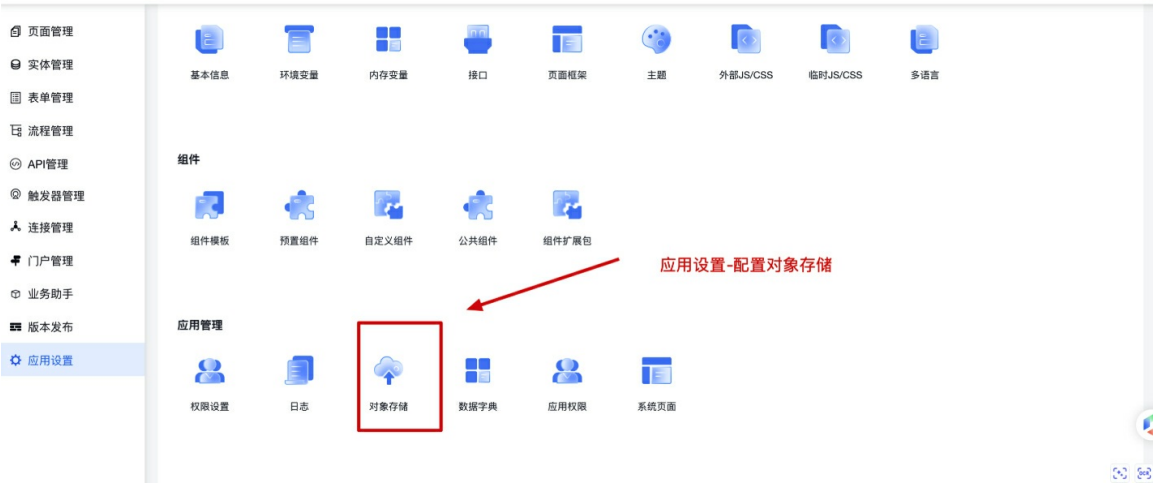
- 创建后找到Access key，点击进入，查看相应的aksk。



在应用中配置对象存储

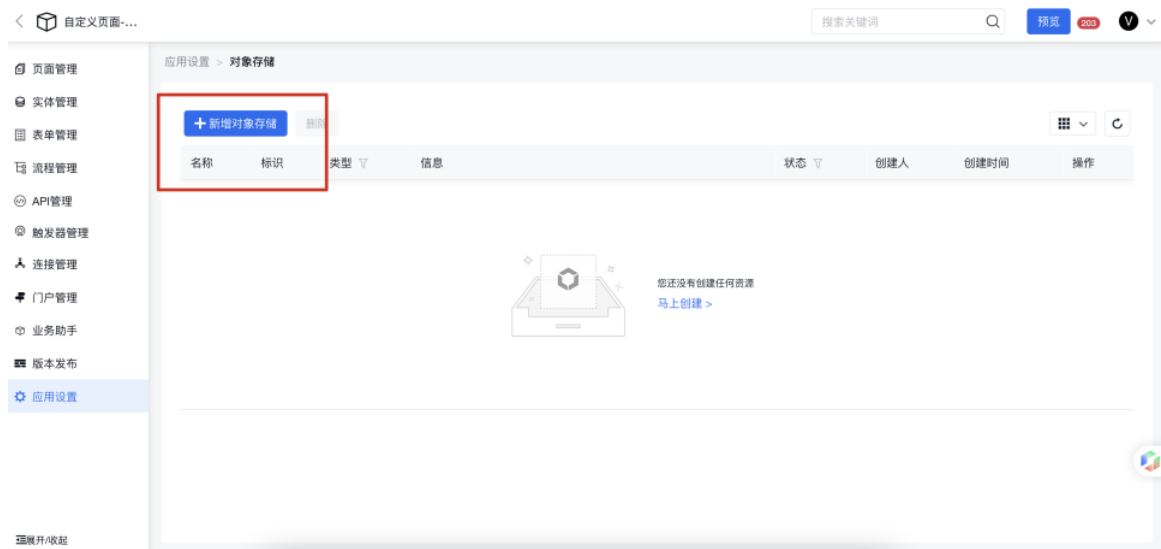
1.新增对象存储

- 在应用设置，点击进入对象存储进行配置

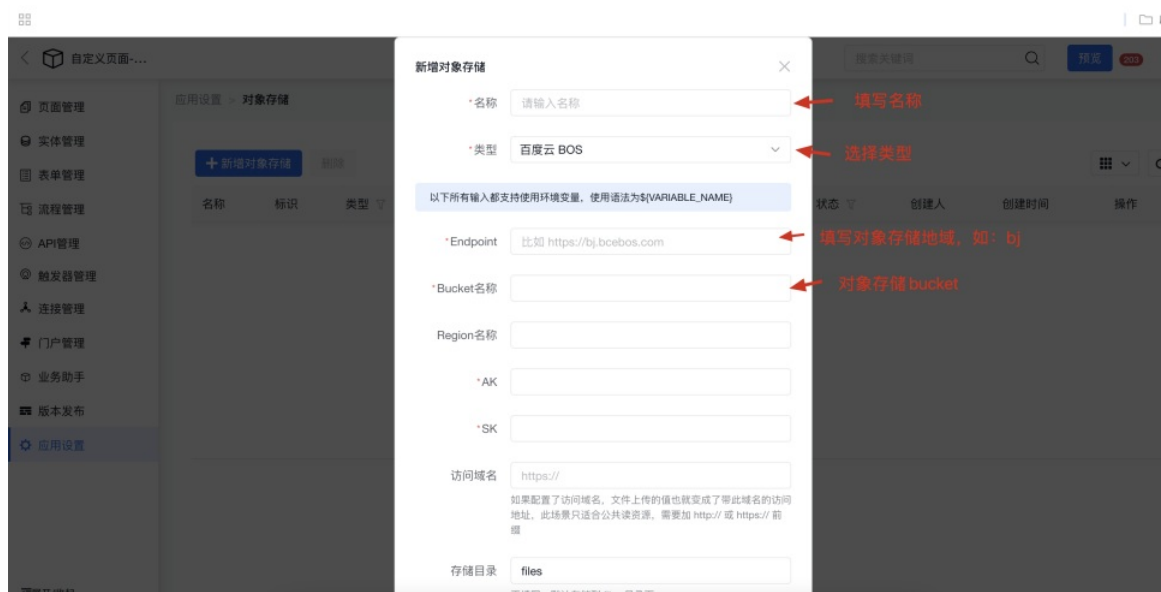


- 新增对象存储



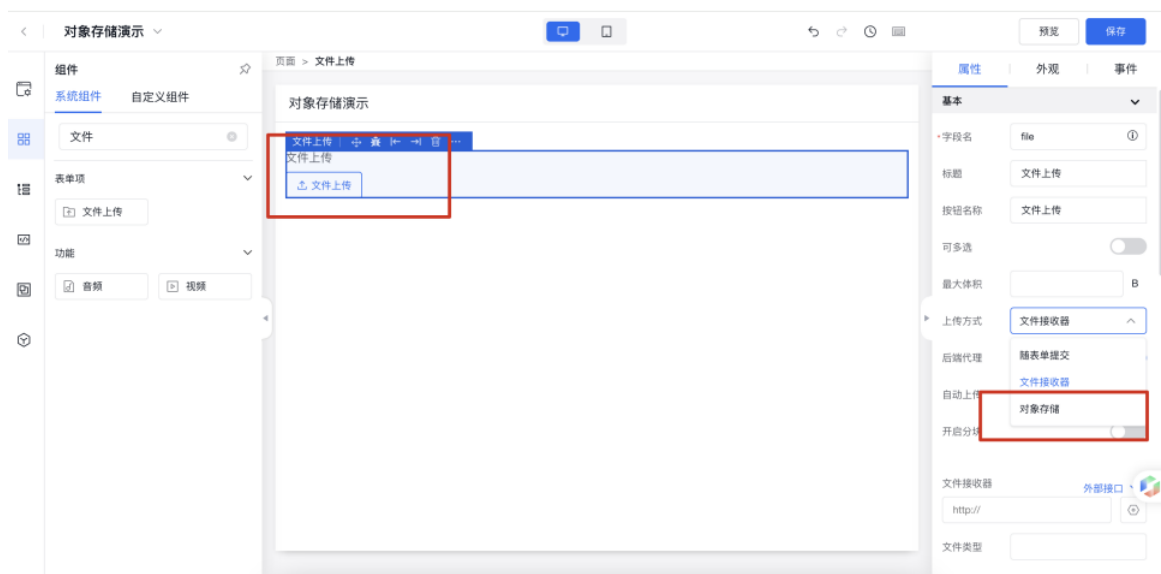


- 填写相应信息，进行创建

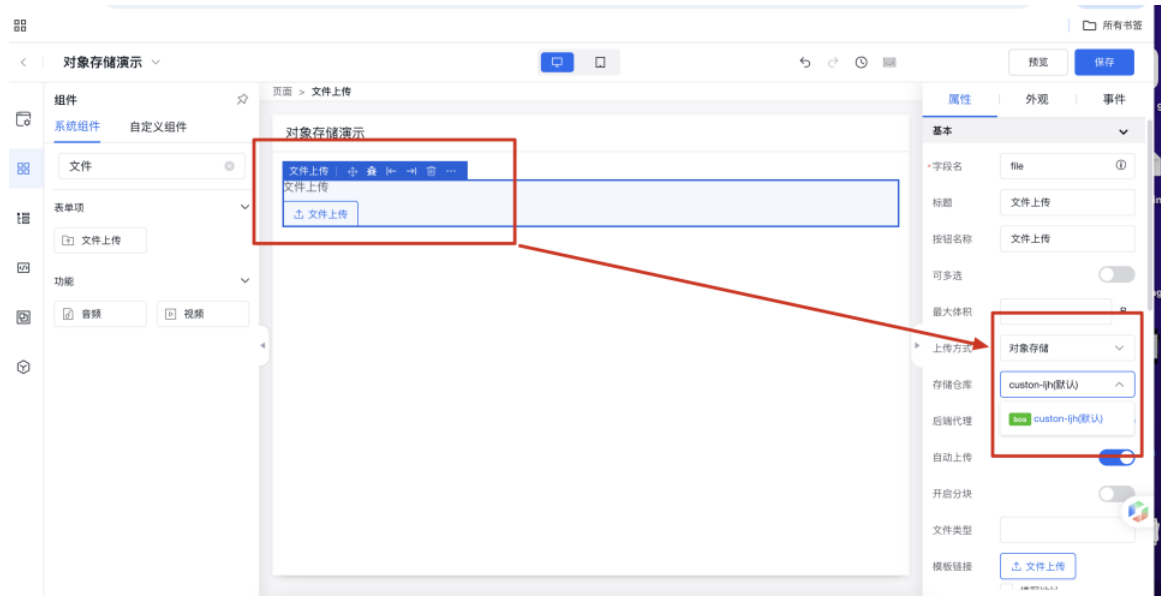


2.相关组件使用方式

- 在对应组件的属性配置中，上传方式选择对象存储



- 选择已经创建的对象存储



# BPM引擎

## BPM概述

### 概述

BPM引擎是爱速搭基于BPM2.0规范，自研的业务流程引擎，场景覆盖审批流程场景、业务流程的场景。

### BPM2.0简介

#### BPMN2.0发展历史

BPMN最初由业务流程管理计划(BPMI)于2004年开发。2005年BPMI与对象管理组(OMG)合并。一年后，BPMN被OMG正式采纳为标准。BPMN2.0于2010年开发，但直到2013年才发布。该标准于同年由国际标准化组织(ISO)正式发布——ISO/IEC19510。

#### BPMN2.0的用途和好处

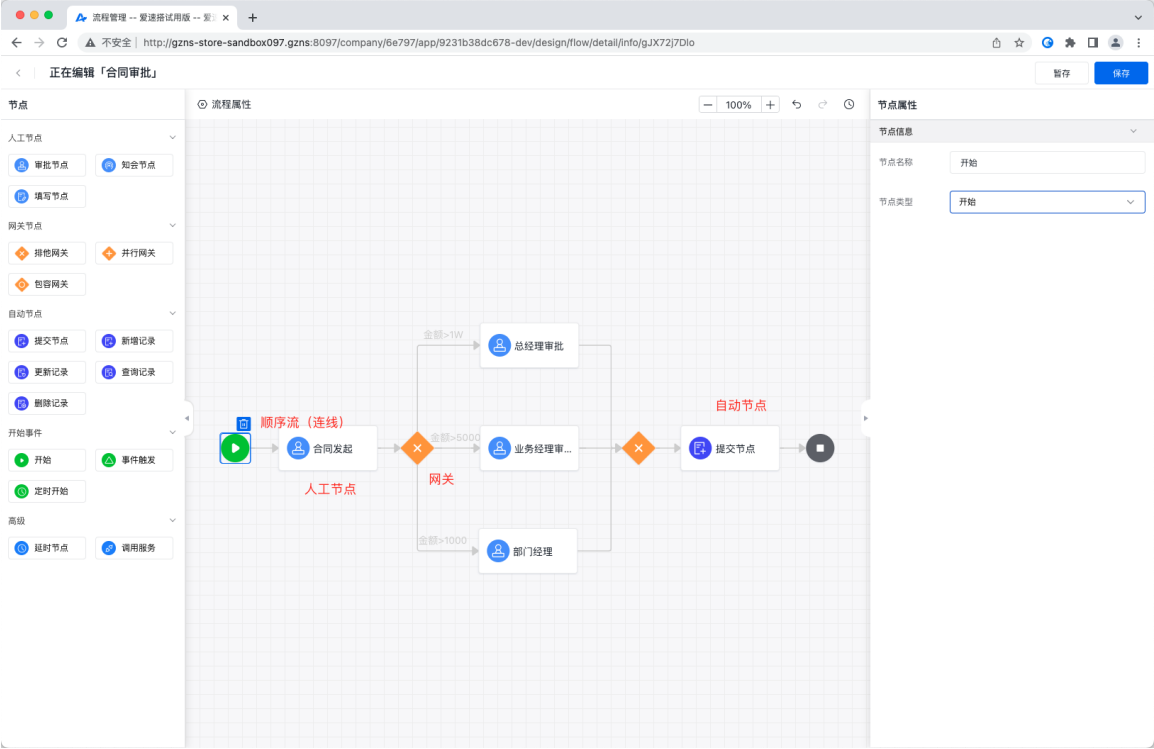
BPMN2.0用于构建易于遵循的业务流程模型流程图。与其他业务流程建模工具一样，BPMN2.0通过一种普遍理解的语言帮助利益相关者更好地理解流程。业务流程中涉及的步骤的可视化表示使业务用户可以轻松了解流程的工作方式。在技术层面，BPMN2.0提供了足够的细节来实现流程。

BPMN2.0还有助于缩小业务流程管理各个阶段之间的差距，从而更轻松地从业务流程设计转向实施。这主要是由于BPMN2.0提供的直观理解水平，使服务不同功能的利益相关者能够协同工作。

BPMN2.0相对于其他业务流程建模工具的另一个主要优势是，利用BPMN2.0的业务流程图可以通过基于XML的BPMN格式无缝转换为流程模型。

### 基本概念

业务流程主要由事件、活动、网关、顺序流组成。其中事件、活动、网关在BPMN2.0中也被称为流对象。



| 要素  | 说明                                        |
|-----|-------------------------------------------|
| 事件  | 事件扮演着流程触发器的角色，通过事件可以完成触发流程的启动、暂停、终止等      |
| 活动  | 即相关的业务活动，包含人工活动（如人工审批活动）和服务活动（级后端自动执行的活动） |
| 网关  | 即决策节点，通过网关可以判断不同后续分支的执行策略和分支等待汇聚策略。       |
| 顺序流 | 显示将执行活动的顺序，可以理解为连线                        |

事件

事件用来表明流程的生命周期中发生了什么事。事件总是画成一个圆圈， 在BPMN 2.0中事件有两大分类：

- 捕获（Catching）事件。当流程执行到该事件, 它会中断执行，等待被触发，如图中延时节点，抛出一个定时器，等待计时到期后，触发流程继续执行。
- 抛出（Throwing）事件。当流程执行到该事件, 抛出一个结果



| 分类    | 节点类型     | 简要说明                                                           |
|-------|----------|----------------------------------------------------------------|
| 开始类事件 | 开始事件     | 常规的开始事件，不指定事件的起因，支持人工、API触发。                                   |
|       | 定时开始     | 定时开始事件用来在指定的时间启动一个流程，也可以在指定周期内循环启动多次流程，例如每月1号凌晨2点开始启动账务结算处理流程。 |
|       | 实体事件开始   | 当接收到特定的实体事件时该流程被触发，例如指定实体的增删改事件后，触发当前流程。                       |
|       | 外部http触发 | 由外部http节点触发。                                                   |
| 中间类事件 | 延时节点     | 当执行到达【延时节点】时中断在这里，引擎会创建一个定时器，当定时器触发后事件结束，流程沿后继路线继续执行。          |
| 结束类事件 | 结束事件     | 表示流程或分支的自然结束，什么都不做。当流程有多个分支路线被激活时，最后一个分支自然结束后，流程实例结束。          |
|       | 终止事件     | 表示流程被强制终止，什么都不做。当流程有多个分支路线被激活时，这些分支上的活动任务也被终止。                 |

活动

即流程中执行的任务，包含人工活动、自动活动。

- 人工活动：需要人员参与的任务，一般包含任务处理人、任务详情页等。
- 自动活动：无需人员参与的任务，由后端自动执行，也称之为自动任务。

| 分类   | 活动名称 | 说明                                                 |
|------|------|----------------------------------------------------|
| 人工活动 | 填写节点 | 即人工填写指定表单，流程流转到该节点，会创建一个流程待办，由任务执行者进行填写任务。         |
|      | 审批节点 | 即人工审批节点，流程流转到该节点，会创建一个人工审批待办，可在待办中心访问，由任务执行者执行审批任务 |
|      | 知会节点 | 即知会指定人员，流程流转到该节点，被知会人员会收到一条知会任务，可在待办中心访问查看         |
| 自动活动 | 提交记录 | 针对当前流程中的临时数据，提交到对应的实体中，也就是入库                       |
|      | 新增记录 | 在实体表单中新增一条、批量新增多条记录                                |
|      | 删除记录 | 在实体表单中删除一条、批量删除多条记录                                |
|      | 更新记录 | 更新实体中的指定记录，包含更新实体记录字段、批量更新实体记录                     |
|      | 查询记录 | 据条件查询平台对象的实例记录，类似于数据库中的SELECT命令                    |
|      | 调用服务 | 调用API及服务编                                          |

网关

网关即流程中的决策点，决定了流程分支执行、汇聚等待的规则。

| 网关名称 | 图标 | 说明                                                             |
|------|----|----------------------------------------------------------------|
| 排他网关 |    | 排他网关定义了一组分支的唯一决策，所有流出的分支被顺序评估，第一个条件被评估为true的分支被执行，并不再继续评估下面的分支 |
| 并行网关 |    | 并行网关根据前置连线或后继连线，无条件创建分支或回收分支                                   |
| 包容网关 |    | 包容网关是排他网关和并行网关的综合体。当决策时，与排他网关所不同的是，所有条件为true的后继分支都会被执行         |

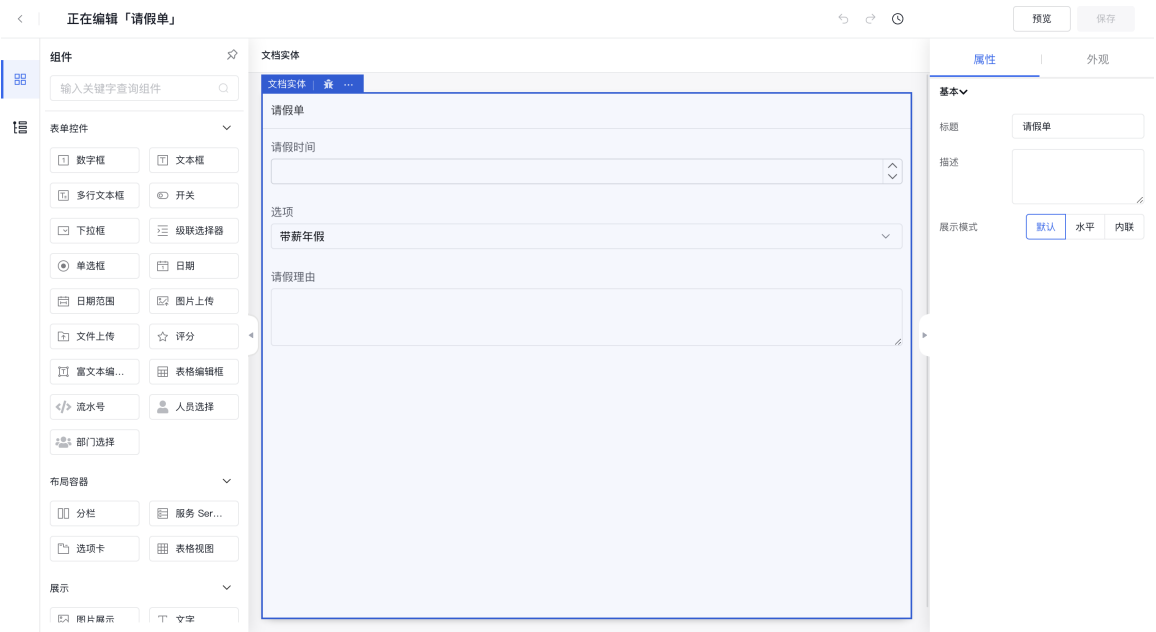
## 快速入门

本节会以请假为示例。展示一个完整的流程配置以及运行示例。

### 🔗 请假单审批流程

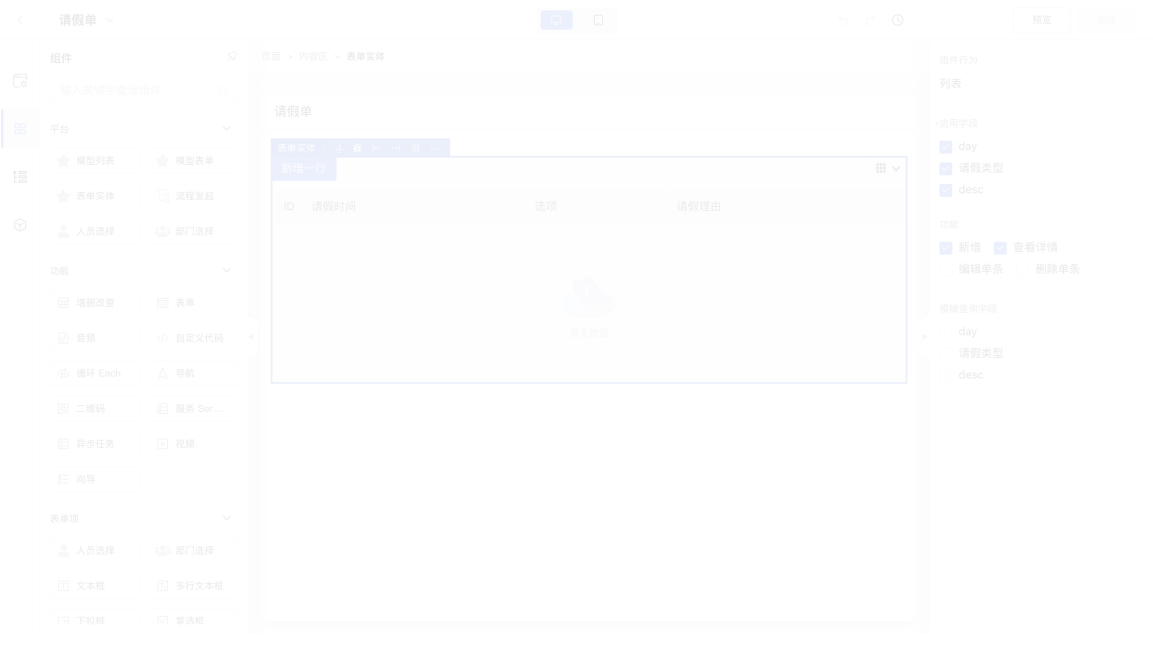
#### step 1: 设计表单

定义了业务模型后，在表单管理中新建一个表单视图。



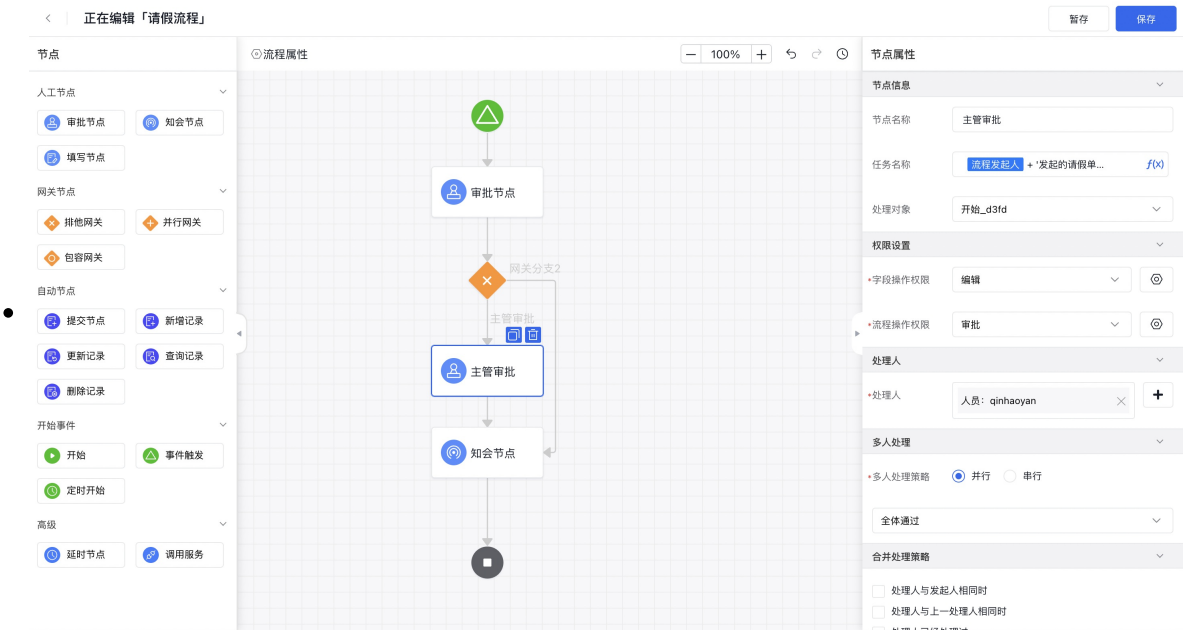
#### step 2: 配置页面

在页面管理中新建页面，并拖入表单组件。



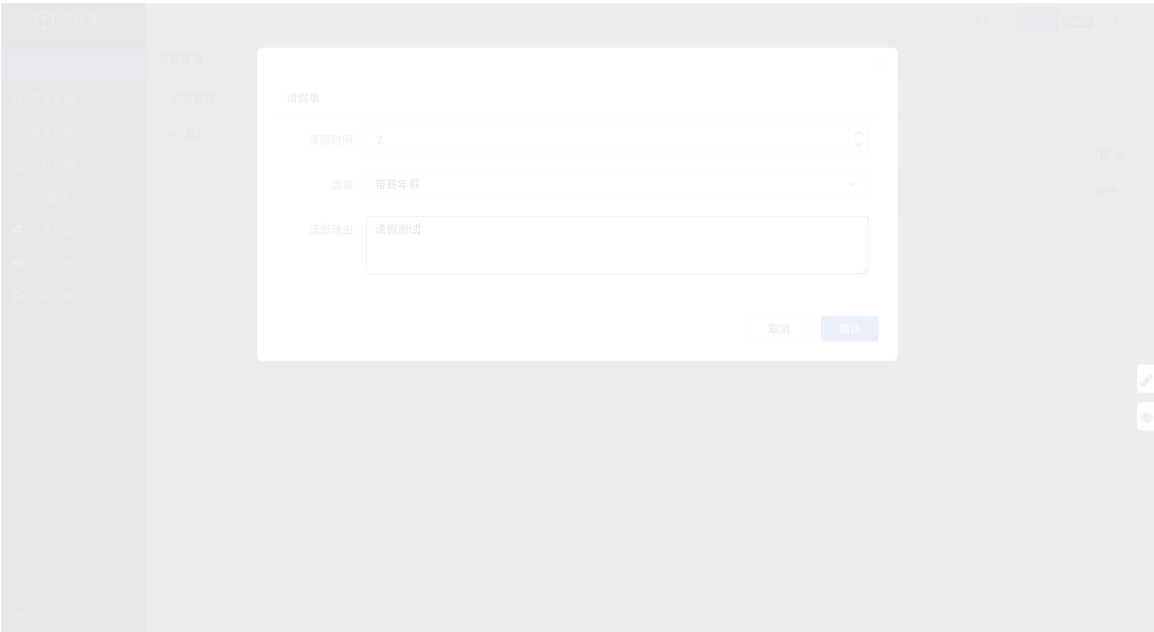
step 3: 设计流程

- 配置开始节点类型为表单事件触发，并选择刚刚新建的表单，事件类型为新建记录前；
- 配置审批节点任务名称为流程发起人 + '发起的请假单：' + 请假时间 + '天'，选择处理对象为开始节点输出参数；配置字段操作权限和流程操作权限；添加处理人；
- 从左侧拖入一个排他网关，再拖入一个审批节点改为主管审批，并为审批节点配置属性；
- 从左侧拖入一个知会节点，并为知会节点配置属性；
- 从排他网关拉出一条线直接连接到知会节点；
- 在排他网关和主管审批的连线配置名称为『主管审批』，并添加条件 请假时间 大于或等于 5；
- 切换到排他网关，配置不满足条件进入的分支。
- 保存配置的流程

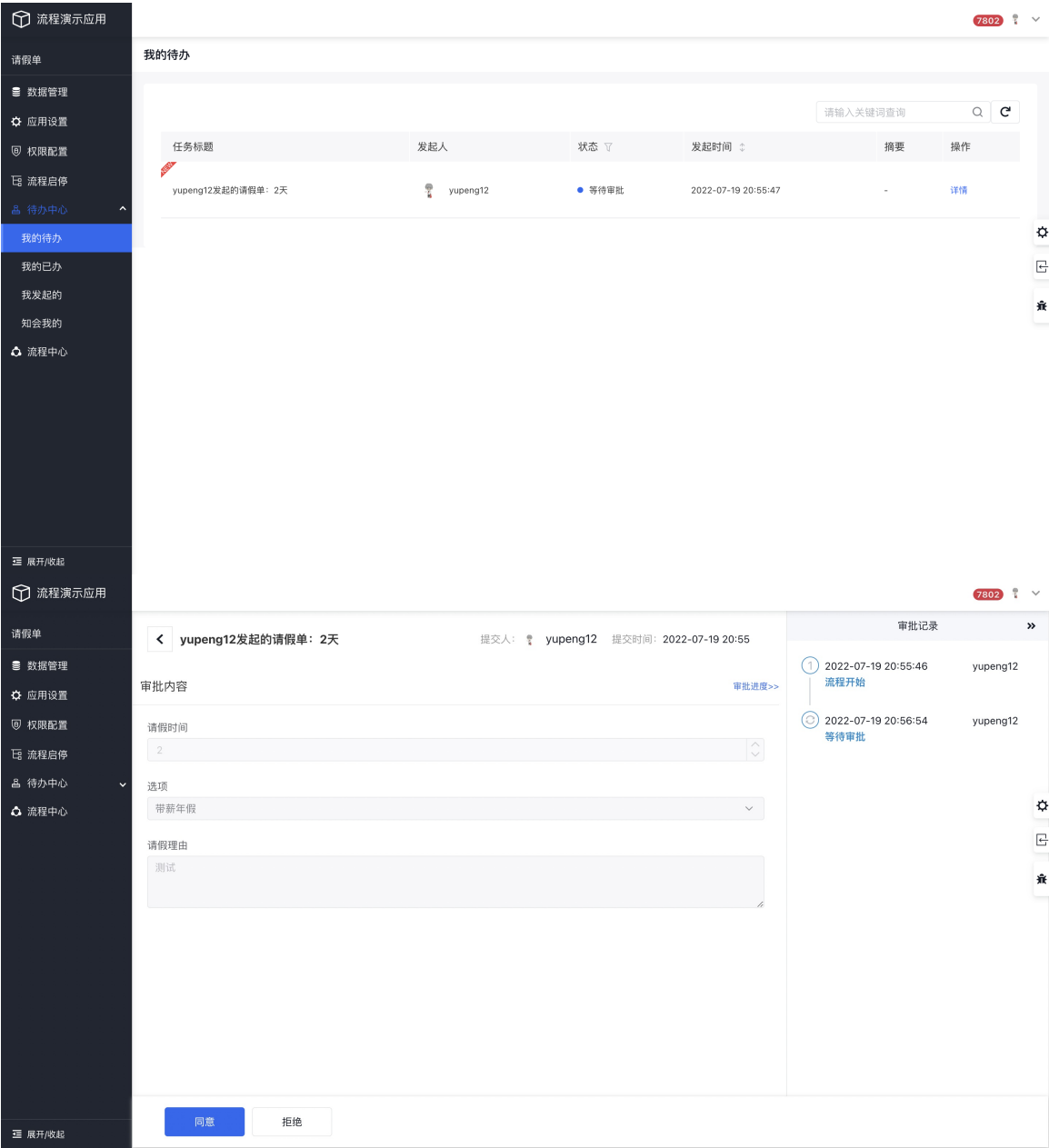


step 4: 发起流程

回到页面管理，新增请假实体数据。



step 5: 审批流程



step 6: 完成

完成流程审批，数据自动写入实体

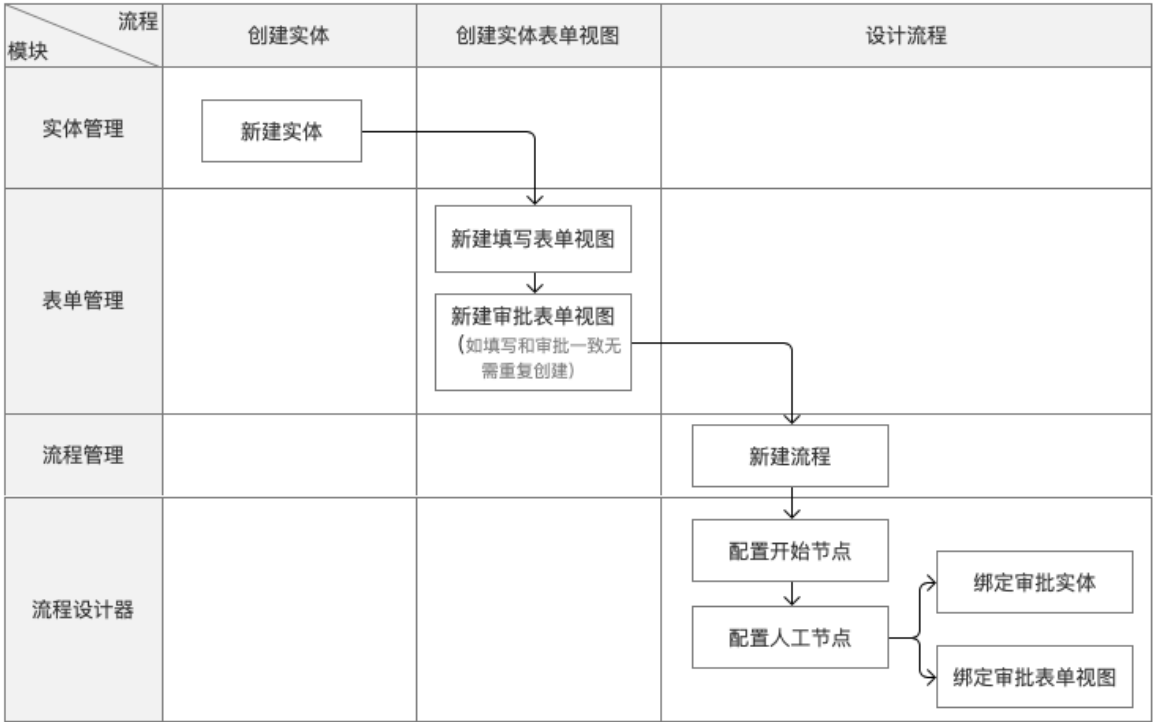


## 流程设计

### 如何设计一个流程

#### 如何创建一个流程

流程设计时一个长线的过程，涉及很多模块，这里以审批场景为例，拆解常用的流程设计过程。



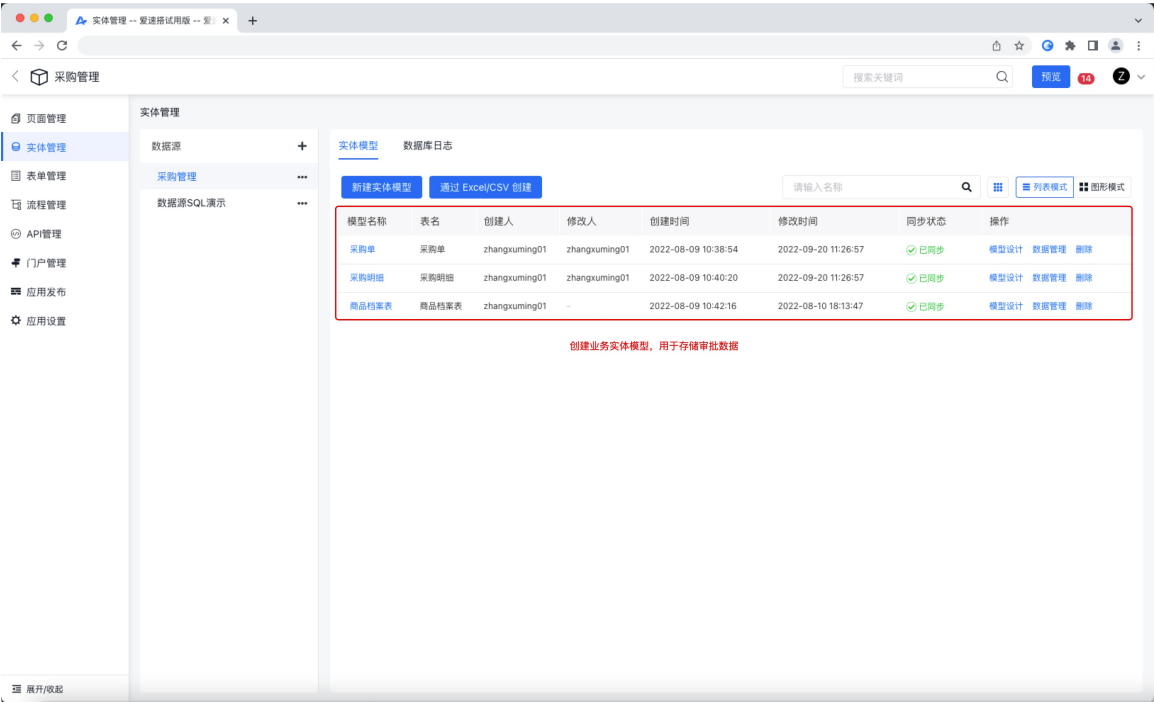
流程设计流程如下图所示

- 创建实体
- 创建表单视图：即创建需要人工任务中需要表单视图，图中包含填写场景、审批场景的表单视图
- 新建流程：创建业务流程
- 设计流程：在流程设计器中设计流程

#### Step1 创建业务实体

创建用于存储流程数据的实体。





Step2 创建表单视图

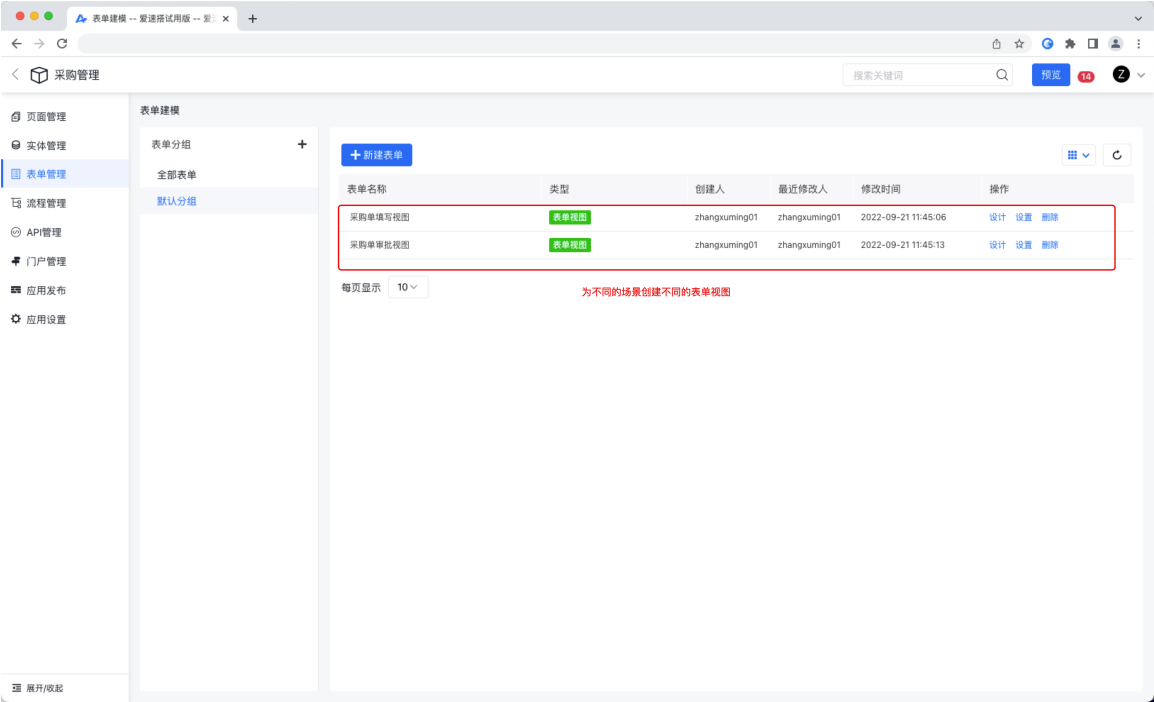
针对任务场景设计表单视图

在填写节点、审批节点中需要使用表单视图作为任务详情页的一部分，因此针对不同的场景创建审批视图，大部分情况分成两类场景：表单填写场景、表单审批场景；这两类场景表单视图，部分场景下填写和审批使用的是同一个表单视图的场景。

具体的人工节点任务详情页的组成，详细参考[人工任务概述.md](#)

通过字段权限

如果统一使用一个表单的视图，也可以通过节点【字段操作权限】，设置部分字段的显示、隐藏、只读、编辑状态微调实际展示的审批视图。



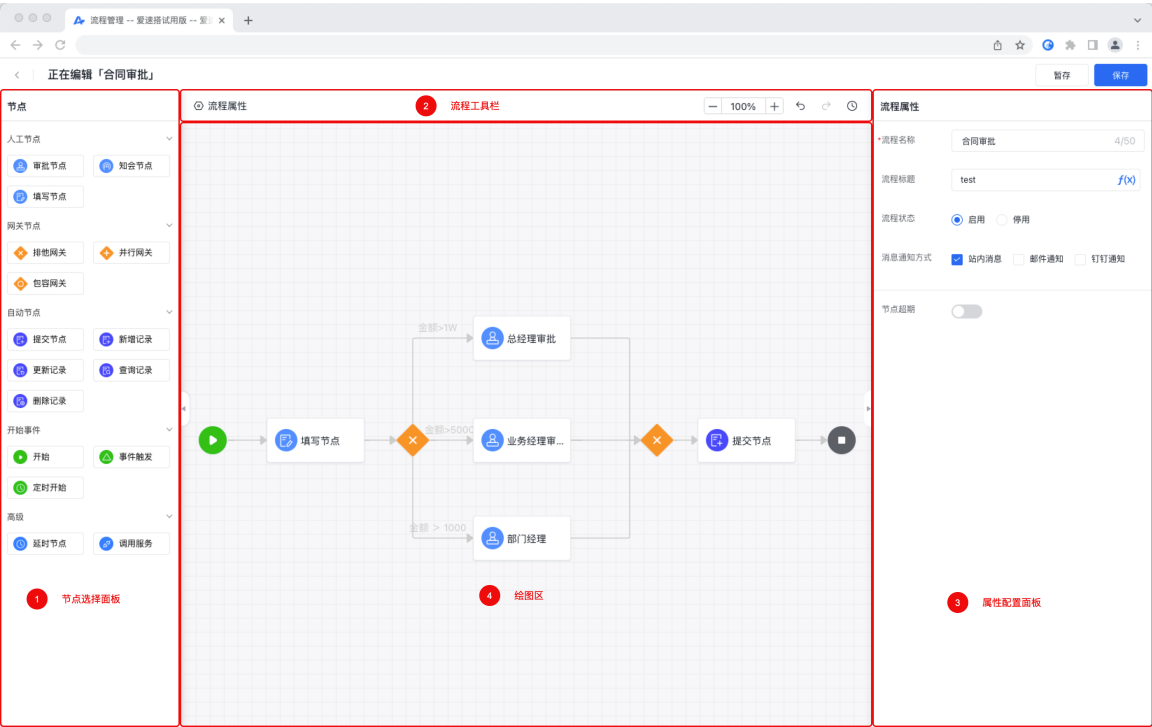
Step3 新建与设计流程

新建流程，并进入流程设计器设计流程。

设计器介绍

爱速搭提供可视化、拖拉拽进行业务流程的设计，整体流程设计器可以分成以下几个部分。

- 节点选择区
- 流程工具栏区：设置流程属性，绘图区缩放、历史版本等设置
- 属性面板区：流程属性、节点属性配置区，根据选择的目标
- 绘图区：流程绘图区，采用自由画布的模式，自由绘制业务流程



流程组成

整个流程由以下要素组成：

- 事件：负责流程的启动与触发，如开始事件、延时事件等；
- 活动：表达流程需要做什么任务，包含人工任务节点（如人工填写、人工审批）、自动任务节点（如调用后端服务等）；
- 网关：扮演着流程决策者的概念，起着两大作用，拆分分支（按规则执行指定分支）、汇聚等待（即等待前面分支的执行完成）。

详细可参加[BPM概述](#)

节点出参

设定节点出参的目的是为了便于后续的节点使用当前节点的出参，扮演着节点之间的协作媒介的作用。

典型场景：使用填写节点的数据进行审批

- 填写节点填写设置了出参【填写的采购单数据】；
- 后续的审批节点可以选择填写节点的出参【填写的采购单数据】作为审批对象，执行审批任务。

节点描述

🔗 事件节点

🔗 开始事件

开始事件

常规的开始事件，不指定事件的起因，支持人工、API触发。

PS：此类的流程不依赖外部事件的参数，因此可以直接发起和调用，包含：

- 1. 在【流程中心】中人工发起流程。
- 2. 在【服务编排】直接调用

🔗 定时开始

功能简介

定时开始事件用来在指定的时间启动一个流程，也可以在指定周期内循环启动多次流程，例如每月1号凌晨2点开始启动账务结算处理流程，整体分成两大类：指定周期重复、指定间隔重复。

指定周期重复

典型场景

按周期时间重复触发，如每日/周/月/季/年重复触发，典型场景如下：

- 每周一上午9.00推送业务报表，开始时间为年初，结束时间为年末
- 每周五中午12.00自动发起周报提醒
- 每周1、3、5 早上10点发起周报提醒 | 界面与属性

节点属性

节点信息

节点名称

开始

节点类型

定时开始

触发规则

触发模式

☐ 间隔触发

☒ 周期触发

触发周期

每周

重复日期

请选择

重复时间

请选择时间

开始时间

请选择日期以及时间

结束时间

请选择日期以及时间

触发次数

自定义

触发次数

1

| 分类   | 说明                                |
|------|-----------------------------------|
| 重复周期 | 可设定按每分、每时、每天、每周、每月内的指定时间进行重复，如每周一 |
| 开始时间 | 即周期重复的开始时间，在开始时间之后，平台会执行重复周期逻辑    |
| 结束时间 | 即周期重复的结束时间，在结束时间之后，平台停止重复逻辑       |
| 触发次数 | 可根据业务自定义触发次数                      |

指定间隔重复

典型场景

按业务指定时间+间隔时间重复触发任务，退货后，每隔3小时通知商家“买家退货，已过去3小时/6小时…”。

界面与属性

节点属性

节点信息

节点名称

开始

节点类型

定时开始

触发规则

触发模式

☒ 间隔触发

☐ 周期触发

开始时间

请选择日期以及时间

结束时间

请选择日期以及时间

间隔设置

5分钟

触发次数

自定义

触发次数

1

| 分类   | 说明                          |
|------|-----------------------------|
| 开始时间 | 即第一次重复开始时间                  |
| 结束时间 | 即间隔重复的结束时间，在结束时间之后，平台停止重复逻辑 |
| 间隔设置 | 重复的间隔时间，如每5分钟、每10分钟         |
| 触发次数 | 可根据业务自定义触发次数                |

🔗 实体事件开始

实体事件开始

典型场景

实体事件开始主要包含监听实体的业务活动，触发对应的流程，比如实体的增、删、改事件触发对应的业务流程，典型场景如：

- 监听实体的新增事件，如当新增采购审批单时，自动触发对应的业务流程；

- 监听实体的更新事件，如当更新合同状态时，自动触发合同状态变更的业务流程；
- 监听实体的删除事件，如当删除合同档案时，自动触发档案变更的业务流程。

界面与属性

如图所示

节点属性

节点信息

节点名称

开始

节点类型

实体事件触发

触发规则

\*请选择实体

采购管理 / 采购单

\*事件类型

新增记录前

输出参数名称

新增采购单

输出当前记录的ID，便于流程中的其他节点引用它；可输入中文、数字、字母或者下划线\_

过滤器

类型选择

☒ 普通模式

☐ 高级模式

采购类型

等于

内采

f(x)

添加条件

| 分类   | 说明                                                                                                                                                                                                                                 |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 监听目标 | 即指定需要监听的实体。                                                                                                                                                                                                                        |
| 事件类型 | 即监听实体的事件类型，包含： <ul style="list-style-type: none"><li>- 新增记录后：</li><li>- 删除记录后</li><li>- 更新记录后</li><li>- 新建记录前：拦截新增记录的写入到实体的行为，触发当前业务流程</li><li>- 删除记录前：拦截删除记录的写入到实体的行为，触发当前业务流程</li><li>- 更新记录前：拦截更新记录的写入到实体的行为，触发当前业务流程</li></ul> |
| 过滤器  | 基于增删改的记录，按条件进行过滤，仅满足条件的记录才能触发当前业务流程                                                                                                                                                                                                |

🔗 延时事件

功能概述

当执行到达【延时节点】时中断在这里，引擎会创建一个定时器，当定时器触发后事件结束，流程沿后继路线继续执行。延期的方式主要包含两种：

- 延时到指定日期

- 延时一段时间

延期到指定时间

流程执行到该节点，会触发定时器，到指定时间后，再执行后续节点，如：在《订单管理》中订单填写了预计发货日期，等待预计发货日期时当天，就通知配货人员，然后执行后续流程

节点名称

延时节点

类型

☐ 延期一段时间

☒ 延期到指定时间

指定日期

请选择日期以及时间

📅

f(x)

延时一段时间

即按基准时间延期一定时间后，在执行后续流程，如客户未“未联系上”，触发流程，延期一定时间后后，执行通知负责的销售进行跟进。

基准时间

即以哪个时间为基准进行延期，整体包含两种模式

| 分类        | 说明                                       |
|-----------|------------------------------------------|
| 上节点任务完成时间 | 即上个节点完成后，延时几天，执行后续动作，如合同状态变更后，1个小时后通知XXX |
| 自定义基准时间   | 由用户自定义的业务基准时间，支持流程中的表单变量、流程变量、以及时间表达式    |

延时时间

根据业务规则定义延期的时间，如2小时、1天、2周、1个月等

产品配置

产品操作界面如图所示

节点属性

节点信息

节点名称

延时节点

类型

☒ 延期一段时间

☐ 延期到指定时间

基准时间

☐ 上节点完成后

☒ 自定义

延期时间

请选择日期以及时间

f(x)

天

时

分

天

f(x)

时

f(x)

分

f(x)

活动节点

人工节点

人工任务概述

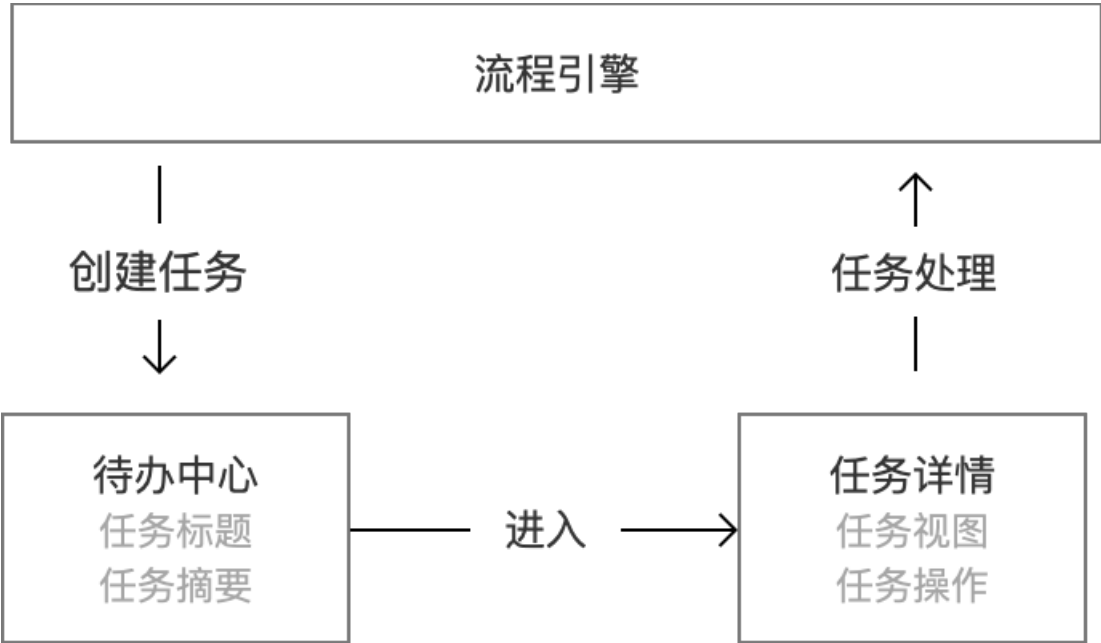
功能概述

人工任务（User Task）用来表示业务流程中由人参与完成的工作。当引擎处理到该节点时，给指定的用户（参与者）或者一组用户（如部门、角色、小组）创建待处理的任务项，等待用户的处理。

实现原理

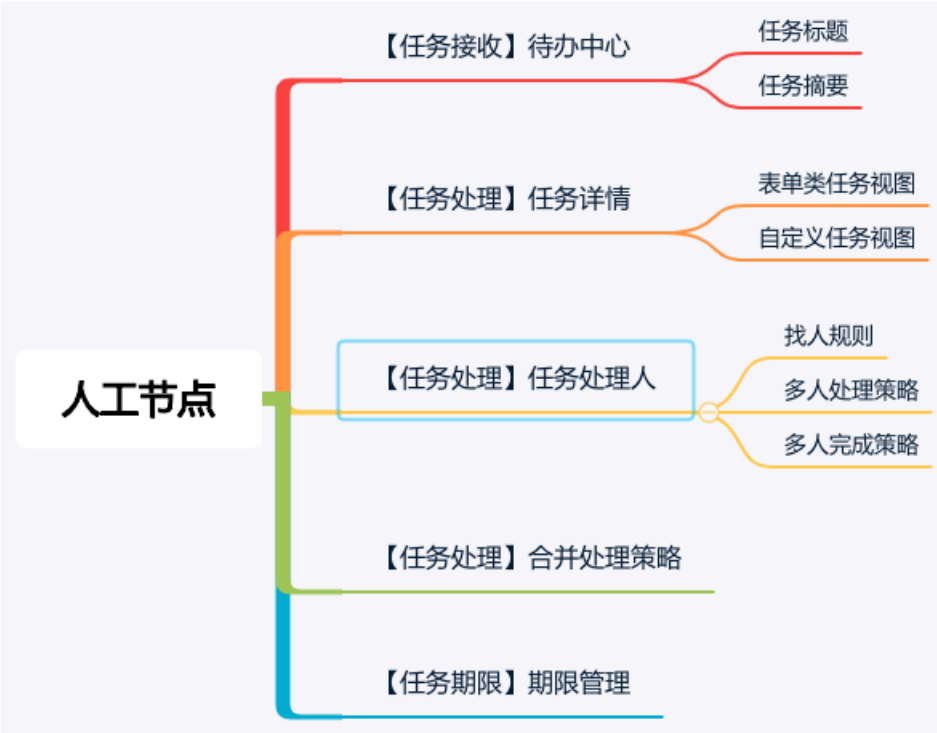
整体由流程引擎进行流程的调度，当运行到人工节点时，执行以下步骤：

- 人工节点创建任务并推送至待办中心；
- 处理人从待办中心进入任务详情并处理任务；
- 任务处理完成后返回任务处理结果，由流程进行处理和调度。



人工节点通用机制概要

如下图所示，人工节点主要包含以下范围的内容

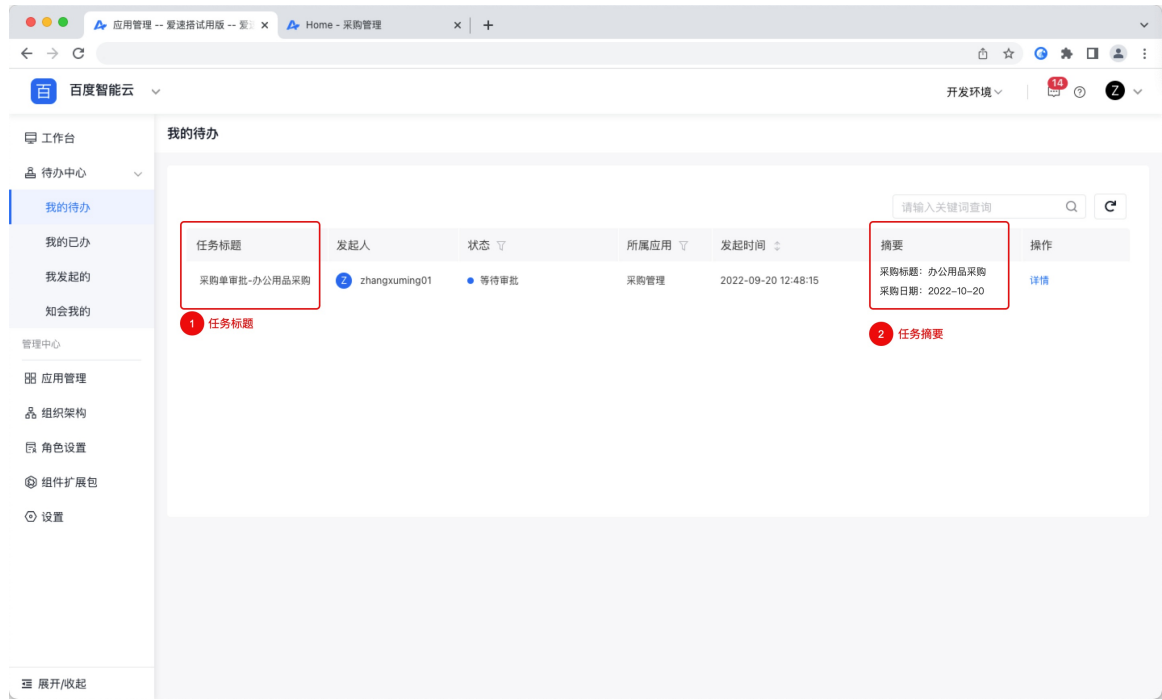


任务接收

爱速搭提供组织级、应用级待办中心，以便于组织成员、应用成员查看当前组织、当前应用的个人待办。

任务列表的呈现如图所示





详细参考 [流程运行](#)

任务标题

使用场景

每个人工任务都会有一个任务标题，任务标题在待办列表中展示，以便于任务处理人通过标题能够快速了解任务的概要。

配置示例

任务名称

"采购单审批-" + 采购名称f(x)

配置任务名称

任务名称

"采购单审批-" + 采购名称

表单变量

流程变量

请输入

填写采购单

采购名称

文本

采购人

人员信息

采购日期

日期

采购类型

枚举

【采购明细】ID

数字

函数

请输入

逻辑函数

IF

AND

OR

XOR

IFS

数学函数

IFS

IFS(condition1, result1, condition2, result2,...condition N, resultN)

判断函数集合，相当于多个 else if 合并成一个。示例：IFS(语文成绩 > 80, "优秀", 语文成绩 > 60, "良", "继续努力") 如果语文成绩大于 80，则返回优秀，否则判断大于

取消

确认

支持通过以下元素定义任务标题。

| 元素   | 说明              |
|------|-----------------|
| 文本   | 直接使用文本定义任务的标题   |
| 表单变量 | 使用表单中的变量来定义标题   |
| 流程变量 | 使用流程中的变量定义任务的标题 |

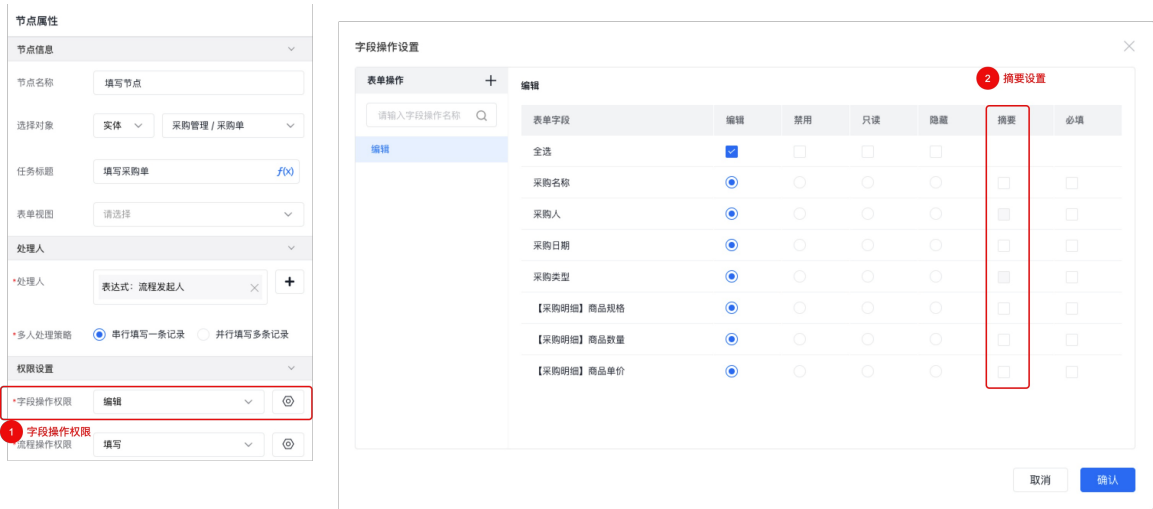
任务摘要

使用场景

在待办列表中显示任务的摘要字段，以便任务处理人能够从任务列表中了解审批任务的关键信息，如合同金额、合同金额、发起人员等。

配置方式

配置路径：人工节点/字段操作权限/字段操作明细/摘要设置，如下图所示



注意事项

为了防止摘要字段过多，影响待办列表的阅读，仅可最多选择5个摘要字段。

任务详情

处理人在处理任务时，需要进入任务详情，对于大部分任务的详情都是由前端页面承接，也就是说定义任务详情基本等同于任务详情页面的定义。

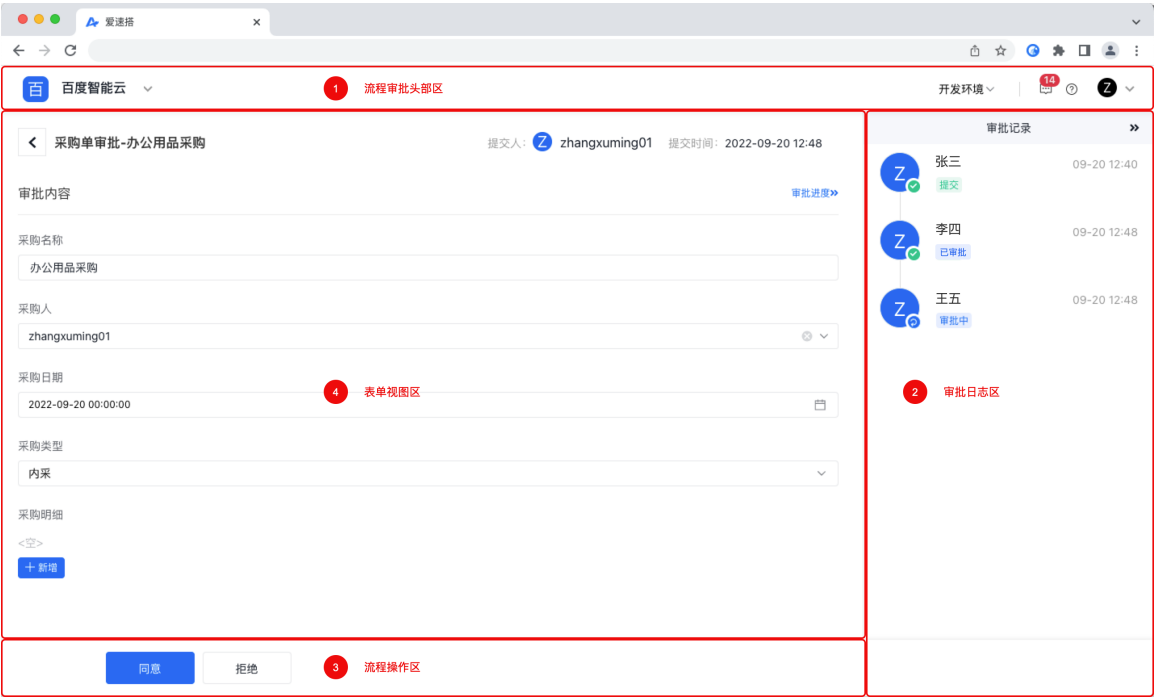
表单审批类任务详情页定义

对于大部分表单审批场景，审批任务界面没有太多个性化诉求，可以使用任务详情页框架+表单视图的方式，快速定义表单类填写、审批的任务详情页面。

表单类任务详情页要素

表单类（填写、审批）的任务详情页面，主要包含几个要素

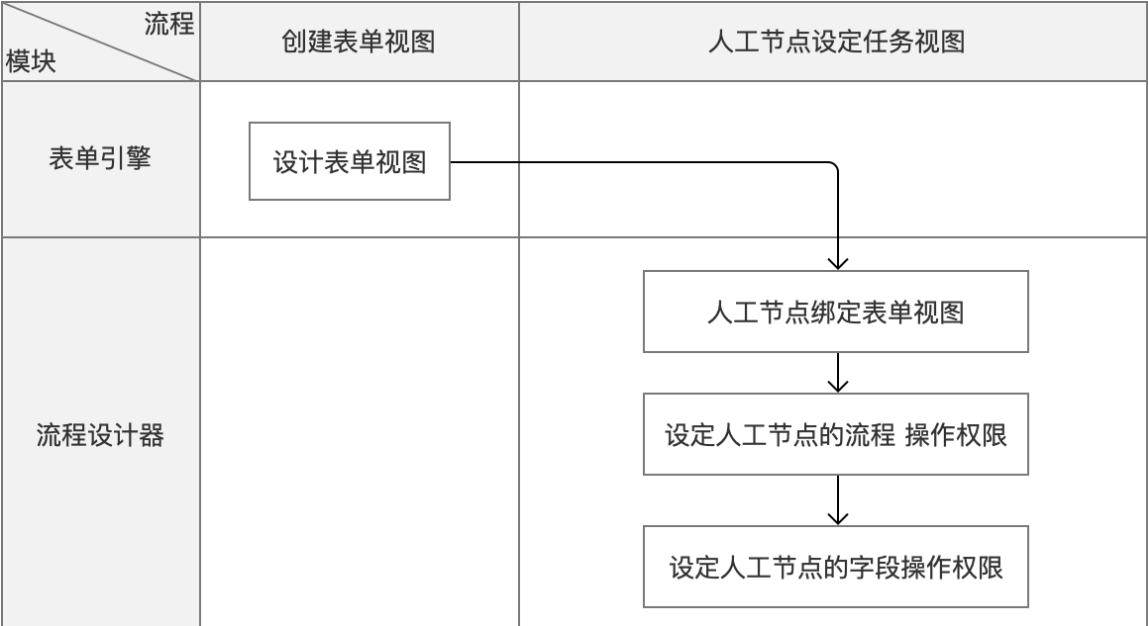
- ① 流程审批头部区
- ② 审批日志区
- ③ 流程操作区
- ④ 表单视图区



| 区域      | 说明                                     |
|---------|----------------------------------------|
| 任务详情页框架 | ①+②+③构成了任务详情页的框架，由爱速搭平台预置，解决详情页快速设计的问题 |
| 表单视图区   | 区域④表单视图区，展示当前人工节点对应的表单视图               |

配置过程

整体设计过程如图所示。



Step1 设计表单视图

详情见 [表单视图](#)

Step2 绑定表单视图

人工节点绑定表单视图，如图所示，绑定处理对象后，即可选择关联的表单视图，或者单独新建表单视图。

节点属性

节点信息

节点名称

审批节点

任务名称

"采购单审批-"+

采购名称

f(x)

处理对象

填写采购单

表单视图

请选择

权限设置

采购审批视图

+ 新建表单视图

字段操作权限

流程操作权限

审批

Step3 设定字段操作权限

因为审批节点不同、审批角色不同，常常需要根据设定表单的中字段的操作权限，如针对不同审批节点

- 设定字段的编辑、禁用、只读
- 设定字段的显示、隐藏

1、选择字段操作权限组

在同一个复杂的流程中，多个审批节点字段操作权限相同的问题，可以定义一个字段操作权限组，流程中的每个节点可以复用统一一个字段操作权限组。

字段操作权限

编辑

2、配置字段操作明细

设计每个字段操作权限组的字段操作明细，定义表单中的每个字段操作权限，如图所示

字段操作设置

表单操作

+ 请输入字段操作名称

编辑

查看

字段操作权限组

编辑

| 表单字段       | 编辑                                  | 禁用                       | 只读                       | 隐藏                       | 摘要                                  | 必填                       |
|------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|
| 全选         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                                     |                          |
| 采购名称       | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 采购人        | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="checkbox"/>            | <input type="checkbox"/> |
| 采购日期       | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 采购类型       | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="checkbox"/>            | <input type="checkbox"/> |
| 【采购明细】商品规格 | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 【采购明细】商品数量 | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="checkbox"/>            | <input type="checkbox"/> |
| 【采购明细】商品单价 | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="checkbox"/>            | <input type="checkbox"/> |

字段操作权限明细

取消

确认

Step4 设置流程操作权限

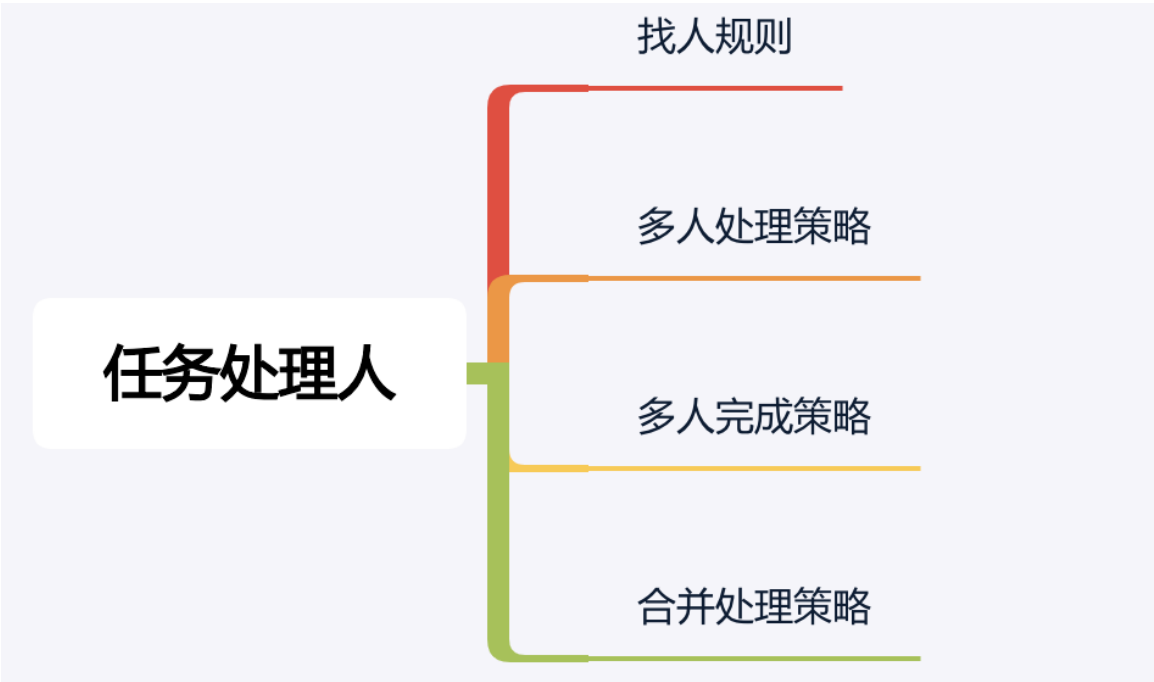
即设置当前节点，流程处理人可以进行哪些流程操作。爱速搭针对不同的场景，预置不同的流程动作。

自定义任务详情页面

研发中，尽情期待

任务处理人

任务处理人即定义当前人工节点的任务由谁处理，主要包含以下内容



处理人定义

任务处理人的定义包含：

- 找人规则：爱速搭提供多种找人规则
- 人员排序：当任务处理人为多人时，提供人员排序功能，以便于明确串行执行的任务顺序。

处理人

\*处理人

人员： zhaoyongzhi

人员： v\_jinruolin

人员： zhangxuming01

表达式： 流程发起人

+

组织人员

表达式

找上下级

找人规则

基于组织找人

支持基于组织通讯录找人，包含

| 找人方式    | 说明                                      |
|---------|-----------------------------------------|
| 通讯录找人   | 基于通讯录指定人员处理                             |
| 按部门找人   | 基于组织架构，指定部门成员处理，如XX部门全员、XX部门成员（包含子部门成员） |
| 按部门角色找人 | 基于组织架构，指定部门角色处理，如部门主管、部门分管领导            |
| 按应用角色找人 | 基于应用的角色，指定业务角色成员处理，如合同管理员               |
| 按平台角色找人 | 基于组织的管理角色，指定组织角色处理                      |

组织人员

最近

部门

人员

岗位

角色

☐

Z

zhaoyongzhi

qatest

☐

☐

V

v\_jinruolin

qatest

☐

☐

Z

zhangxuming01

fff

☐

☐

R

renpin

qatest

☐

☐

L

lvxiaojiao

fff

☐

☐

D

dengqianchuan

qatest

☐

☐

W

wanghaotong

qatest

☐

已选项

清空

请选择左侧数据

取消

确认

基于表达式找人

支持基于表达式找人。

| 找人方式   | 说明                                       |
|--------|------------------------------------------|
| 表单变量找人 | 基于表单中的人员字段进行找人，如设定表单中的【下节点处理人】字段为当前节点处理人 |
| 流程变量找人 | 基于流程变量找人，如流程发起人找人                        |
| 其他表达式  | 基于自定义的表达式找人                              |

找上下级

| 找人方式    | 说明                            |
|---------|-------------------------------|
| 按人员汇报关系 | 按人员的直接主管的汇报关系找人，如处理人为发起人的上级主管 |
| 按部门层级找人 | 按部门层级找指定部门角色，如任务处理人为上级部门主管    |

找上下级

×

人员变量

流程发起人

f(x)

路线方向

☒ 找上级

☐ 找下级

部门层级

当前部门

▼

岗位

部门经理

▼

取消

确认

多人处理与完成策略

当一个人工节点的处理人为多个成员时，往往存在并签、串签等场景。

串行（串签）

即多个处理人串行处理当前节点任务，任务处理人的串行按以下优先级别进行排序

| 排序因素       | 优先级别 | 说明           |
|------------|------|--------------|
| 处理人列表排序    | 1    | 即处理人列表的人员排序  |
| 部门、角色内成员排序 | 2    | 按部门、角色内的成员排序 |

并行（会签）

即当前节点的多个任务处理人并行处理当前节点任务。

并行完成策略

当任务处理人并行处理时，如何判定当前节点任务完成并向后续节点执行，流程设计师可以根据业务规则设定并行完成策略。

| 分类      | 说明                                                                     |
|---------|------------------------------------------------------------------------|
| 全体通过    | 全部人员通过，才算通过，任意一人否决，该节点就被否决，执行否决动作，或者进入否决流程                             |
| 一人通过即通过 | 有时也叫或签，任意一人通过即可，当节点被通过后，其他人员的任务自动完成并消失，消息通知完成                          |
| 按比例通过   | 指定比例、数量的人员通过，怎该节点通过，进入下一个节点，当超过（100%-指定比例）时，则默认为该节点被否决，执行否决动作，或者进入否决流程 |

多人处理

\*多人处理策略

☒ 并行

☐ 串行

一人拒绝则拒绝

全体通过

一人通过即通过

按比例通过

自动处理策略

业务场景

在同一个流程中，如果处理人已经处理过前面节点的任务，为了简便流程处理，可以执行自动处理。继承上次节点处理态度，支持以下场景的合并处理。

配置说明

| 分类           | 说明                         |
|--------------|----------------------------|
| 处理人与发起人相同时   | 即当发起人即为处理人时，执行自动处理，默认为同意   |
| 处理人与上一个处理相同时 | 即当处理人刚处理完上一节点时，下一个节点执行自动处理 |
| 处理人已经处理过时    | 当处理已经处理过该流程时，执行自动处理        |

产品界面

合并处理策略

☐ 处理人与发起人相同时

☐ 处理人与上一处理人相同时

☐ 处理人已经处理过

任务期限

业务场景

对于人工任务节点，每个任务都有期限，可以针对超期场景，定制相关的业务规则，典型场景如



|     |                                |
|-----|--------------------------------|
| 分类  | 场景实例                           |
| 超期前 | 通知提醒任务处理人：“XX任务即将超期，请尽快处理”     |
| 超期后 | 升级转交指定上级主管人员办理审批               |
|     | 通知提醒任务处理人：“XX任务已经超期，已升级转交主管审批” |
|     | 执行流程绩效管理扣分逻辑                   |

功能概要



超期定义

指定业务超期时间

当表单中包含了任务的超期时间，可以使用表单中的变量或者流程中的变量定义超期时间，作为任务的超期时间

计时时间节点+超期期限

流程管理员可以在设计流程时设定流程级【合理处理期限】，这样所有人工任务节点默认按超期开始计时规则，超期期限则判断为超期。相关计时规则如下：

|          |               |
|----------|---------------|
| 规则       | 说明            |
| 接收任务开始计时 | 即接收到待办任务后开始计时 |
| 查看任务开始计时 | 即查看了待办任务后开始计时 |

超期动作

| 分类    | 动作        | 说明                          |
|-------|-----------|-----------------------------|
| 消息通知类 | 发送站内信     | 向指定人员发送站内信                  |
|       | 钉钉、企微消息   | 向企微、钉钉等终端消息                 |
| 流程处理类 | 转交        | 转办升级，移交指定人处理：升级领导审批、转交秘书审批等 |
|       | 自动同意      | 超期自动同意                      |
|       | 自动拒绝      | 超期自动拒绝                      |
| 业务规则类 | 调用服务（API） | 调用接口，执行相关的业务规则              |
|       | 调用http    | 调用http，执行相关业务逻辑             |

配置过程

如下图所示，此图界面进行配置

超期设置

超期设置

自定义节点超期规则

计时规则

接收任务后开始计时

期限设置

1

小时

超期前规则

超期前

10

分

重复间隔

5分钟

动作设置

消息提醒处理人

消息模板

暂无数据

f(x)

超期后规则

填写节点

填写节点

即人工填写节点，如果流程中需要某个成员提供一些信息来推进流程执行，可以使用该节点。流程流转到该节点，会创建一条填写单据的待办任务，处理人可以去待办中心执行任务（填写对应的单据并提交）。、

典型场景

填写单据，发起流程

即提交表单数据，常用适用于填写流程表单场景，比如填写采购单、请假单等单据，其特征收集一条数据。

数据采集，填写问卷

即问卷收集场景，常用于面向一个群体的问卷调研，提交的是多份数据。

分项描述

任务标题

自定义该节点的任务节点的表给，即在待办中心显示的任务标题

任务标题

"填写采购单-" + 流程发起人 f(x)

支持通过表达式设定标题，支持以下常用变量定任务标题

| 支持变量 | 说明                                                |
|------|---------------------------------------------------|
| 表单变量 | 基于表单中的变量定义任务标题，如任务标题为“审批采购单-[采购人]”，其中[采购人]为一个表单变量 |
| 流程变量 | 基于流程变量定义任务标题，如任务标题为“填写采购单-[流程发起人]”，其中[流程发起人]为流程变量 |

配置任务名称

任务名称

采购人

表单变量

流程变量

请输入

Q

新增采购单

ID 数字

采购名称 文本

采购人 人员信息

采购日期 日期

采购类型 枚举

函数

请输入

Q

逻辑函数

IF

AND

OR

XOR

IFS

数学函数

IF

IF(condition, consequent, alternate)

示例：IF(A, B, C) 如果满足条件A，则返回B，否则返回C，支持多层嵌套IF函数。也可以用表达式如：A ? B : C

取消

确认

填写对象

即选择向哪个实体中填写数据

选择对象

实体

采购管理 / 采购单

表单视图

绑定填写对象对应的表单视图，主要用于任务视图的展现，以便于设计前端填写、审批任务中擦使用的表单视图。

处理人

即填写任务的处理人，详细见找人规则

多人处理策略

可根据场景选择不同的多人处理策略

| 分类       | 说明                                                               |
|----------|------------------------------------------------------------------|
| 串行填写一条记录 | 用于表单填写场景，一般仅有一人填写，即提交表单数据，常用适用于填写流程表单场景，比如填写采购单、请假单等单据，其特征收集一条数据 |
| 并行填写多条记录 | 用于问卷收集场景，常用于面向一个群体的问卷调研，提交的是多份数据，最终的节点输出参数为一个记录列表                |

字段操作权限

因为审批节点不同、审批角色不同，常常需要根据设定表单的中字段的操作权限，如针对不同审批节点

- 设定字段的编辑、禁用、只读
- 设定字段的显示、隐藏

详细见[人工任务概述](#)

摘要字段

在流程运行时，为了更好提升待办中心的流程待办的可读性，增加流程任务摘要字段的功能，以便于在待办列表中直观查看流程的摘要字段。

设置路径： 字段操作权限/ 字段操作明细 / 摘要字段

字段操作设置

表单操作

+

请输入字段操作名称

编辑

查看

编辑

| 表单字段       | 编辑                                  | 禁用                       | 只读                       | 隐藏                       | 摘要                                  | 必填                       |
|------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|
| 全选         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                                     |                          |
| 采购名称       | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 采购人        | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |
| 采购日期       | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 采购类型       | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |
| 【采购明细】商品规格 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 【采购明细】商品数量 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |
| 【采购明细】商品单价 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |

取消

确认

如上图所示，支持任意勾选摘要字段，最多支持5个字段作为待办摘要字段

详细见[人工任务概述](#)

流程操作权限

即设定处理人处理当前节点任务时，可以进行哪些流程操作，| 动作 | 说明 | | --- | | 提交 | 提交表单数据，进入后续节点流转 | | 暂存 | 暂存当前填写数据，后续补充完成再提交任务 | | 回退 | 回退流程到指定节点 | | 终止 | 终止当前流程 | | 移交 | 移交当前任务，由他人处理 |

详细见[人工任务概述](#)

提交规则

即填写内容的提交规则，包含

| 规则  | 说明                                                          |
|-----|-------------------------------------------------------------|
| 提交  | 即直接提交实体表中（入库），提交后，后续节点会返回可以对当前数据进行使用，并且会返回填写记录的ID，以便于后续节点定位 |
| 不提交 | 即填写的数据不提交，作为流程变量存储于流程中，后续节点可以使用当前填写的数据进行相关的操作               |

输出参数

即改节点的输出参数，以便于后续节点使用，解决节点之间的协作问题。

审批节点

功能概述

即人工审批节点，主要用于流程审批，流程流转到该节点，会触发审批任务，任务处理人可以从待办中心查看并处理改审批任务

典型场景

主要用于表单审批场景，如请假单、采购单等审批场景

分项说明

任务标题

自定义该节点的任务节点的表给，即在待办中心显示的任务标题

任务标题

"填写采购单-"+流程发起人

f(x)

支持通过表达式设定标题，支持以下常用变量定任务标题

| 支持变量 | 说明                                                |
|------|---------------------------------------------------|
| 表单变量 | 基于表单中的变量定义任务标题，如任务标题为“审批采购单-[采购人]”，其中[采购人]为一个表单变量 |
| 流程变量 | 基于流程变量定义任务标题，如任务标题为“填写采购单-[流程发起人]”，其中[流程发起人]为流程变量 |

配置任务名称

任务名称

采购人

表单变量

流程变量

请输入

Q

新增采购单

ID

数字

采购名称

文本

采购人

人员信息

采购日期

日期

采购类型

枚举

函数

请输入

Q

逻辑函数

IF

AND

OR

XOR

IFS

数学函数

IF

IF(condition, consequent, alternate)

示例：IF(A, B, C) 如果满足条件A，则返回B，否则返回C，支持多层嵌套IF函数。也可以用表达式如：A ? B : C

取消

确认

审批对象

即需要审批的对象，可以选择当前流程节点之前的其他节点出参作为审批对象。

选择对象

实体

▼

采购管理 / 采购单

▼

表单视图

绑定填写对象对应的表单视图，主要用于任务视图的展现，以便于设计前端填写、审批任务中使用的表单视图。

处理人

即填写任务的处理人，详见[人工任务概述](#)

多人处理策略

可根据场景选择不同的多人处理策略，即

- 串行（串签）
- 并签（会签）

详见[人工任务概述](#)

字段操作权限

因为审批节点不同、审批角色不同，常常需要根据设定表单的中字段的操作权限，如针对不同审批节点

- 设定字段的编辑、禁用、只读
- 设定字段的显示、隐藏

详见[人工任务概述](#)

摘要字段

在流程运行时，为了更好提升待办中心的流程待办的可读性，增加流程任务摘要字段的功能，以便于在待办列表中直观查看流程的摘要字段。

设置路径： 字段操作权限/ 字段操作明细 / 摘要字段

字段操作设置

表单操作

+

请输入字段操作名称

Q

编辑

×

查看

×

编辑

| 表单字段       | 编辑                                  | 禁用                       | 只读                       | 隐藏                       | 摘要                                  | 必填                       |
|------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|
| 全选         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |                                     |                          |
| 采购名称       | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 采购人        | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="checkbox"/>            | <input type="checkbox"/> |
| 采购日期       | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 采购类型       | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="checkbox"/>            | <input type="checkbox"/> |
| 【采购明细】商品规格 | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 【采购明细】商品数量 | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="checkbox"/>            | <input type="checkbox"/> |
| 【采购明细】商品单价 | <input checked="" type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="radio"/>    | <input type="checkbox"/>            | <input type="checkbox"/> |

取消

确认

如上图所示，支持任意勾选摘要字段，最多支持5个字段作为待办摘要字段

详细见 [人工任务概述](#)

流程操作权限

即设定处理人处理当前节点任务时，可以进行哪些流程操作，| 动作 | 说明 | | --- | --- | | 提交 | 提交表单数据，进入后续节点流转 | | 暂存 | 暂存当前填写数据，后续补充完成再提交任务 | | 回退 | 回退流程到指定节点 | | 终止 | 终止当前流程 | | 移交 | 移交当前任务，由他人处理 | | 加签 | 支持当前会签、后加签等操作 |

详细见 [人工任务概述](#)

提交规则

即填写内容的提交规则，包含

| 规则  | 说明                                                          |
|-----|-------------------------------------------------------------|
| 提交  | 即直接提交实体表中（入库），提交后，后续节点会返回可以对当前数据进行使用，并且会返回填写记录的ID，以便于后续节点定位 |
| 不提交 | 即填写的数据不提交，作为流程变量存储与流程中，后续节点可以使用当前填写的数据进行相关的操作               |

输出参数

即改节点的输出参数，以便于后续节点使用，解决节点之间协作问题。

配置过程

如图所示，按图所示进行配置即可

节点属性

节点信息

节点名称

审批节点

任务名称

"采购单审批-"+采购名称f(x)

处理对象

填写采购单

表单视图

请选择

权限设置

\*字段操作权限

编辑

\*流程操作权限

审批

处理人

\*处理人

人员: zhaoyongzhi

+

多人处理

\*多人处理策略

并行

串行

合并处理策略

处理人与发起人相同时

处理人与上一处理人相同时

处理人已经处理过

超期设置

超期设置

自定义节点超期规则

计时规则

接收任务后开始计时

知会节点

功能概述

知会是在用户在设置流程时加入「知会」节点，用于在审批人审批后给知会人发送消息提醒，知会人不需要审批和执行。不影响流程的流转。

典型场景

”知会“ 等于 ”抄送“，常用于的抄送部门主管等场景。

分项说明

任务标题

自定义该节点的任务节点的表给，即在待办中心显示的任务标题



任务标题

"填写采购单-"+

流程发起人

f(x)

支持通过表达式设定标题，支持以下常用变量定任务标题

| 支持变量 | 说明                                                |
|------|---------------------------------------------------|
| 表单变量 | 基于表单中的变量定义任务标题，如任务标题为“审批采购单-[采购人]”，其中[采购人]为一个表单变量 |
| 流程变量 | 基于流程变量定义任务标题，如任务标题为“填写采购单-[流程发起人]”，其中[流程发起人]为流程变量 |

配置任务名称

任务名称

采购人

表单变量

流程变量

请输入

Q

新增采购单

ID

数字

采购名称

文本

采购人

人员信息

采购日期

日期

采购类型

枚举

函数

请输入

Q

逻辑函数

IF

AND

OR

XOR

IFS

数学函数

IF

IF(condition, consequent, alternate)

示例：IF(A, B, C) 如果满足条件A，则返回B，否则返回C，支持多层嵌套IF函数。也可以用表达式如：A ? B : C

取消

确认

知会对象

即需要知会的对象，可以选择当前流程节点之前的其他节点出参作为审批对象。

表单视图

绑定知会对象对应的表单视图，主要用于任务视图的展现，以便于设计前端填写、审批、知会任务中使用的表单视图。

知会人

即填写任务的处理人，[人工任务概述](#)

消息模板

设定消息的内容

字段操作权限

即设定表单视图中字段的显示、隐藏的设置 [人工任务概述](#)

🔗 自动节点

🔗 提交节点

功能概述

用于提交流程中的数据到实体中，也就是数据入库。

典型场景

提交实体未入库数据：当开始节点实体事件开始（采用新建、删除记录前、更改记录前）时，流程会拦截数据入库的行为，后续可以由【提交节点】提交数据到目标实体中。

功能配置

如图所示，仅需选择提交对象就可以了，可以选择当前流程节点之前的其他节点出参作为审批对象。

节点属性

节点信息

节点名称

提交节点

处理对象

请选择

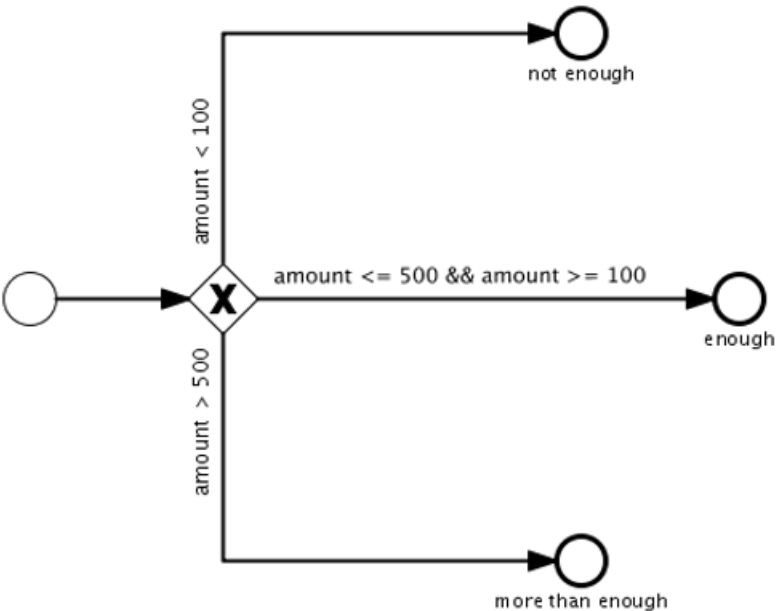
网关节点

排除网关

**排除网关** 排除网关表达了一个流程中的**唯一决策**。会有一个外向顺序流被使用，根据定义在顺序流中的条件。也经常称为**XOR网关**。XOR网关被描绘为一个菱形，内部有一个'X'，一个空的菱形，没有网关也象征着唯一网关。

**使用场景 分支** 当该节点用于分支的作用时，其起着**唯一决策**的作用，作为分支，其执行规则如下：

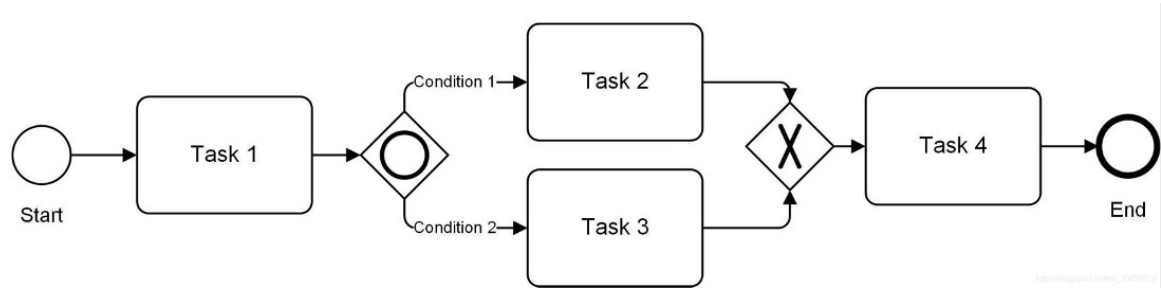
- 后续的所有分支，按分支条件的排序，仅执行满足条件的第一个分支；
- 当所有分支条件都不满足时，执行默认分支。



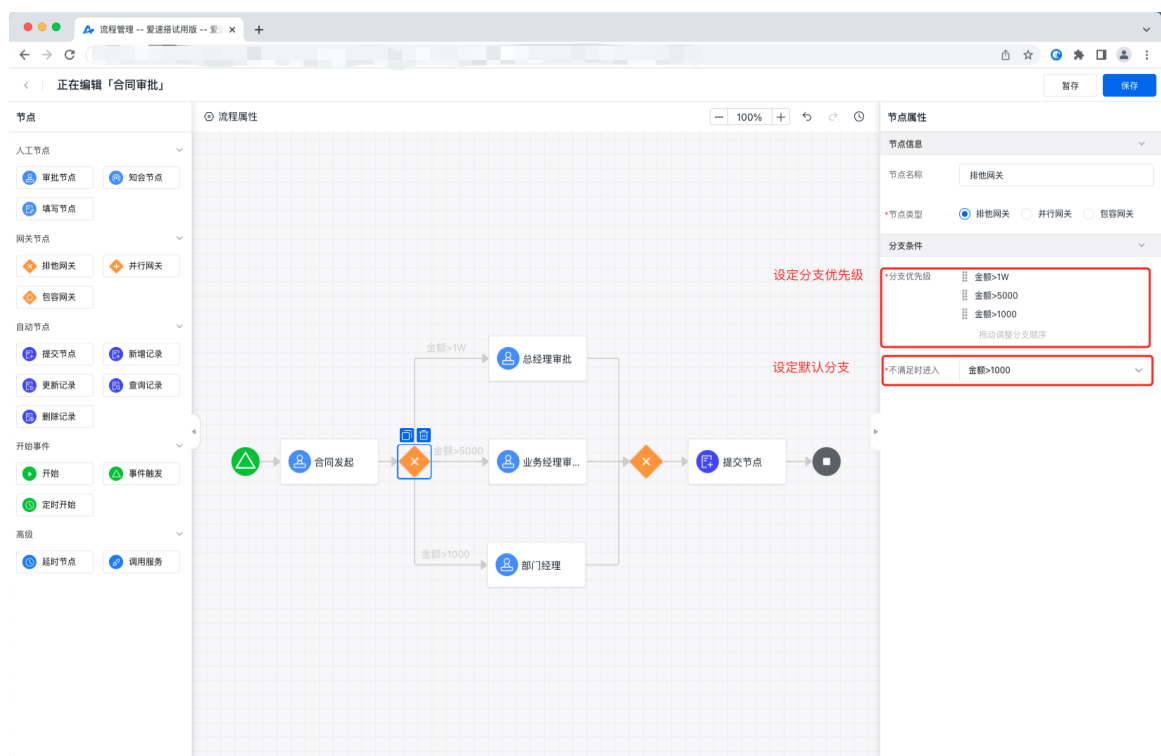
如图所示，作为分支节点时，只会触发满足条件的第一个节点，如果有多个节点满足按分支排序，只取第一个节点

汇聚

作为汇聚节点时，只要有一个活动节点到达该网关那么就触发该网关，对于每个组到达这个网关的分支流程，都会触发一次排除网关，每次触发网关执行排除网关后续分支一次，下图中，当condition1 和 condition2 都满足条件时，排除网关作为汇聚作用，后续的分支会被执行两次，也就是task 4 会被执行两次。



产品配置



设置该节点时，需设置两个关键属性

分支优先级

每次触发排他网关，是按分支优先级，默认执行满足第一个条件的分支，其余分支不在执行，因此需要根据业务规则，设定分支优先级。

默认分支

设定默认分支，以便于当所有分支条件都不满足时，执行默认分支。

包容网关

包含网关 包含网关也叫or-gateway，被用来 进行“条件性”切分或汇聚顺序流，它有两大用途：

- 作为分支时：流转满足条件的的所有分支
- 作为汇聚时：等待满足条件的的所有分支

典型场景

分支

当包含网关作为分支时，它的作用有两层：

- 仅执行满足条件的的所有分支，忽略不满足条件的分支。
- 当所有分支条件都不满足时，执行默认分支



如上图为一个采购单的流程，采购单的内容有两项关键指标：

- 根据采购金额大小，选择主管 or 总监审批
- 当采购类型为“外采”时，执行【外采加批】节点，当采购类型为“内采”时，无需执行【外采加批】。

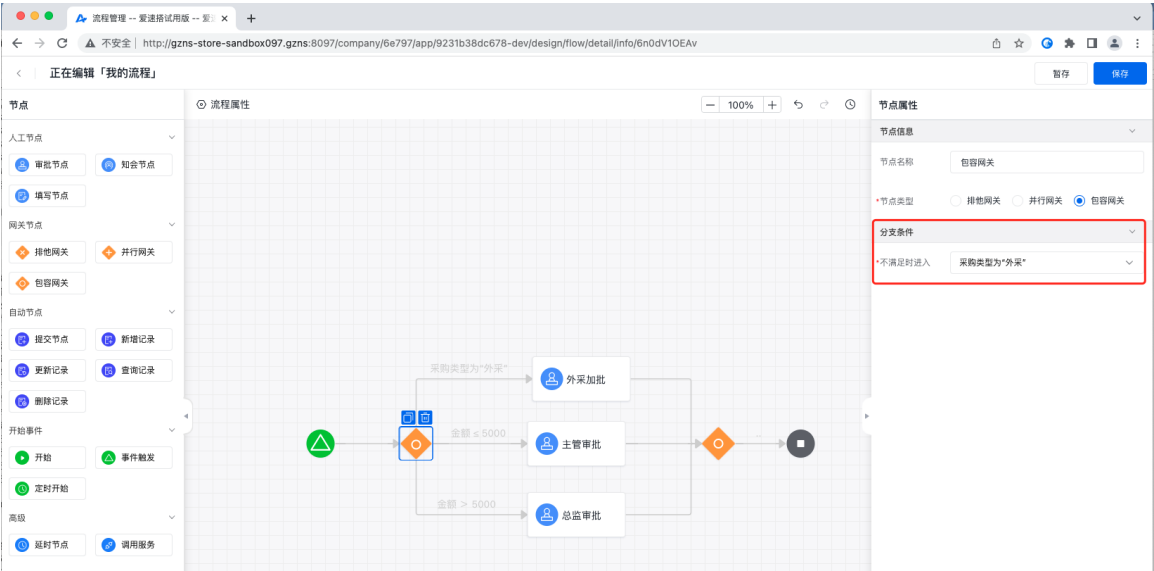
汇聚

包含网关有着汇聚的左右，即由多条件分支进入包含网关时，包含网关会进行汇聚等待，等待当前包含网关前的所有满足条件的分支都执行完成，流转到当前包含网关，才会激活后续节点。

如下案例中，当包含网关作为汇聚节点时，起着“汇聚等待”的作用。



产品配置



包含网关的关键属性配置时，配置默认分支，当所有条件都不满足时，执行后续的哪个分支。

🔗 并行网关

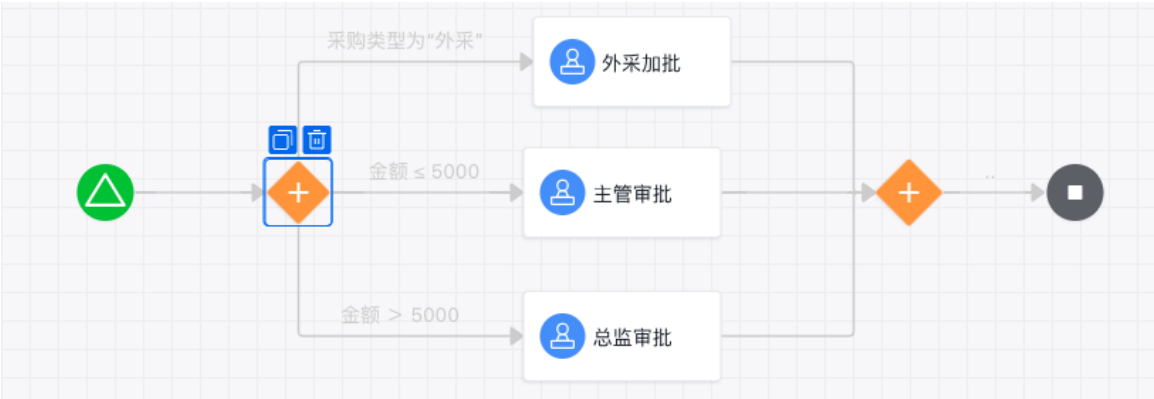
**并行网关** 并行网关根据前置连线或后继连线，无条件创建分支或回收分支，它的用途有两类：

- 作为分支时：忽略分支条件，所有分支并行执行；
- 作为汇聚时：等待所有输入线的任务到达后，汇聚进入后续的节点；

**使用场景**

**分支**

并行网关的每个后继分支路径都被无条件执行。如下图中，左侧的并行网关的所有后续分支都会被执行，忽略其中的条件。



**汇聚**

所有到达并行网关的分支路径都汇聚于此等待，直到每个输入流的分支都执行完毕，然后执行该网关的输出流，如果其中有分支未被执行或中断，那么该并行网关将处于永久等待状态。如上图中的【外采加批】节点若一直处于审批状态，则右侧的并行网关会一直处于等待状态。

流程运行

🔗 流程发起

流程的发起包含两类，自动触发、人工触发类。

**自动触发类流程**

如实体事件触发、定时触发类的流程，无需人工进行发起，会监听对应的业务事件，当业务事件发生后，会自动触发。

**人工发起类流程**

对于开始节点为“开始事件”、“定时触发”的流程，可以通过导航菜单、流程中心人工进行流程的发起。

🔗 流程中心

爱速搭预置了流程中心，显示支持人工发起的业务流程，我们可以直接在流程中心发起流程。

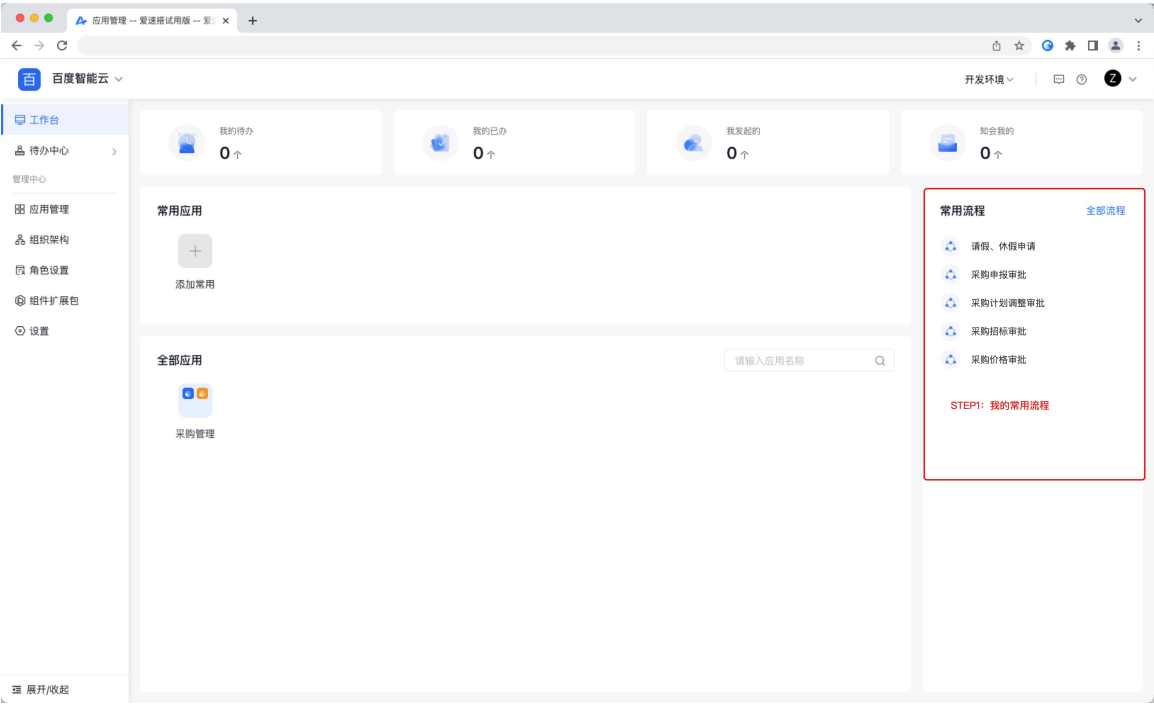
爱速搭提供了组织级的流程中心、应用级的流程中心。

🔗 组织级的流程中心

组织流程中心是当前组织中所有支持人工发起的流程合集，我们可以通过组织级的流程中心，发起当前组织下应用的流程。

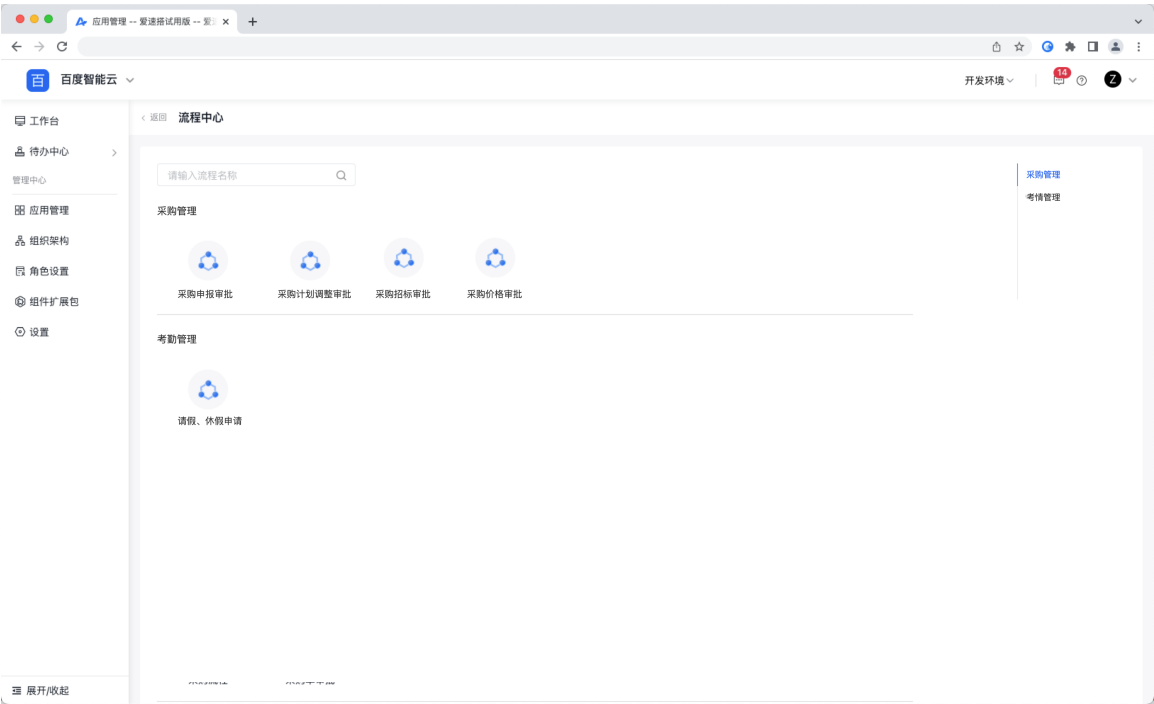
**发起我常用的流程**

发起路径：组织工作台/常用流程



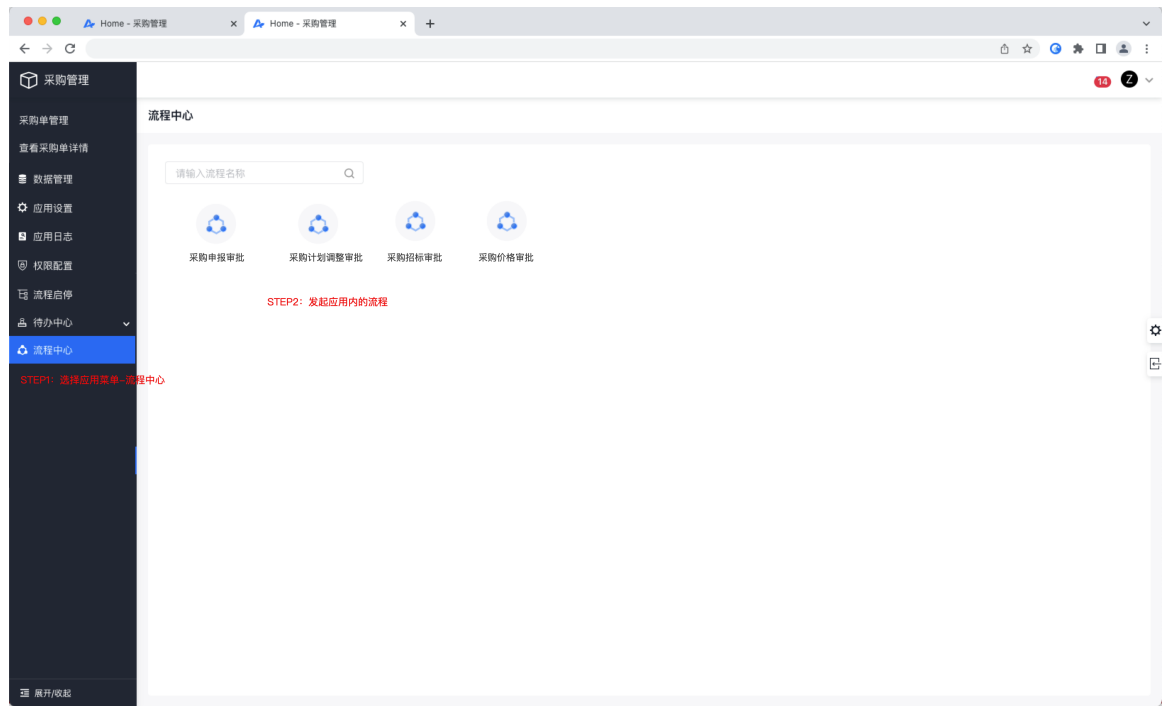
发起组织内的流程

发起路径：组织工作台/常用流程/流程中心



应用级流程中心

应用级流程中心显示了当前应用可以人工发起的流程，我们可以通过应用级的流程中心，发起当前应用内的流程



PS：应用级流程中心的入口是菜单，如果配置流程中心是否在应用导航中出现，可以通过应用门户配置实现，应用门户的配置参考[应用门户](#)

🔗 流程处理

流程运行时，人工节点创建任务后，创建的任务会被推送到个人待办中心，可以通过待办中心进行流程事项的处理。

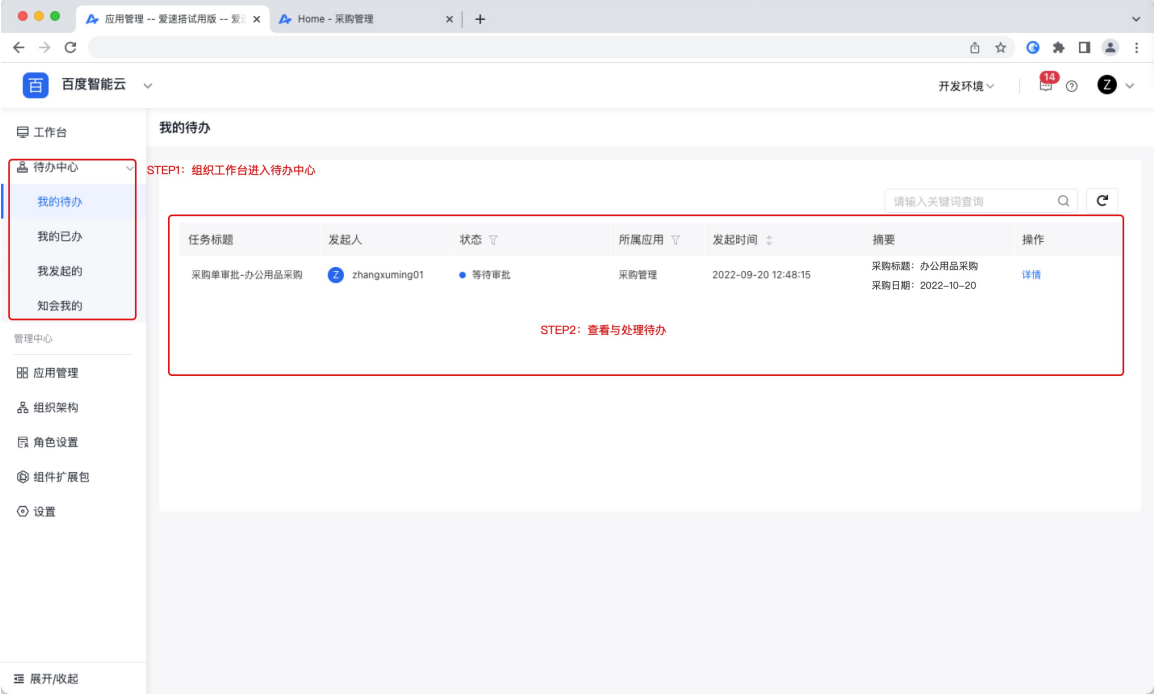
🔗 待办中心

爱速搭预无需应用开发者单独开发待办中心，可直接使用预置的待办中心，提升开发效率。

| 分类   | 说明                              |
|------|---------------------------------|
| 我的待办 | 查看当前登录人员的待办事项，点击详情可以进入任务详情页处理待办 |
| 我的已办 | 查看我处理过的待办事项列表                   |
| 我发起的 | 查看我发起的流                         |
| 知会我的 | 查看我的知会事项，注意该类事项不会影响流程的流转        |

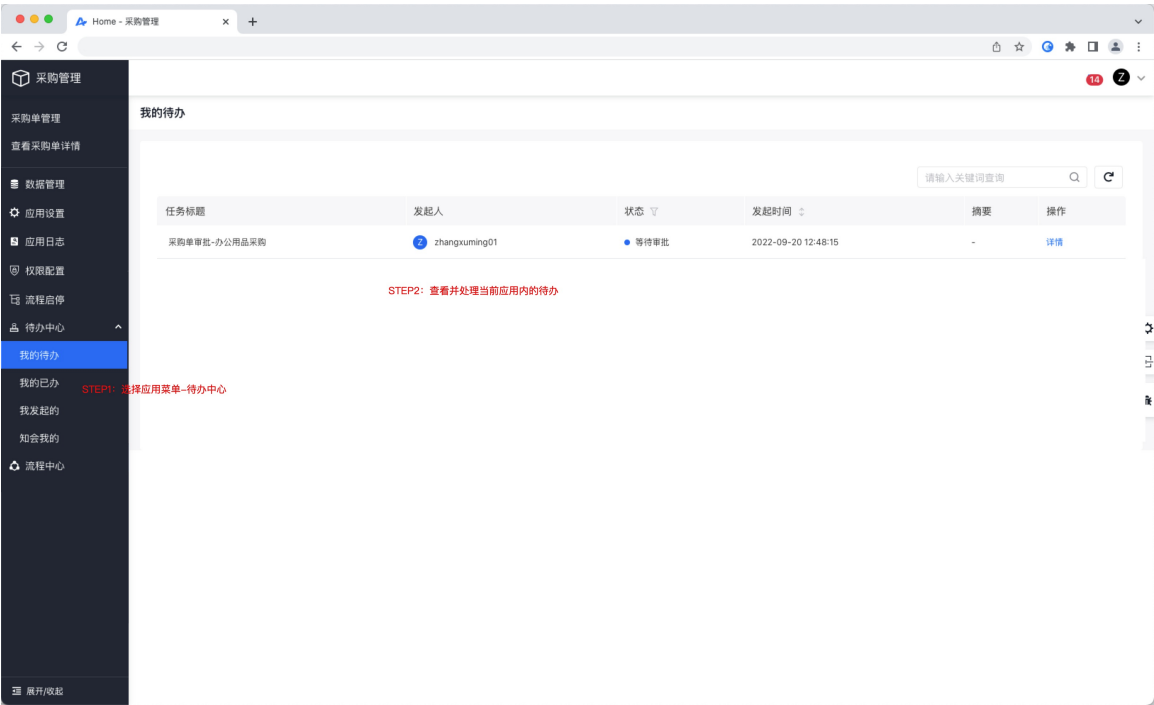
🔗 组织级待办中心

组织级待办中心显示了当前组织内应用创建的所有待办，我们可以通过组织级待办中心，查看和处理当前组织内应用的待办事项。



应用级待办中心

应用级待办中心显示了当前应用内的待办事项，我们可以通过该页面查看和处理当前应用内的待办事项。



# 资源管理

## 图片管理

图片管理

爱速搭平台支持统一资源管理，其中就包含图片资源管理。目前在平台级别、组织级别、以及应用级别都可以独立维护图片，并且图片资源会通过继承机制，实现公共资源的下发和使用。

资源的管理最终是为资源的使用而服务，资源的使用始终是在开发环境下，因此组织级和应用级图片管理只在平台开发环境下才支持。

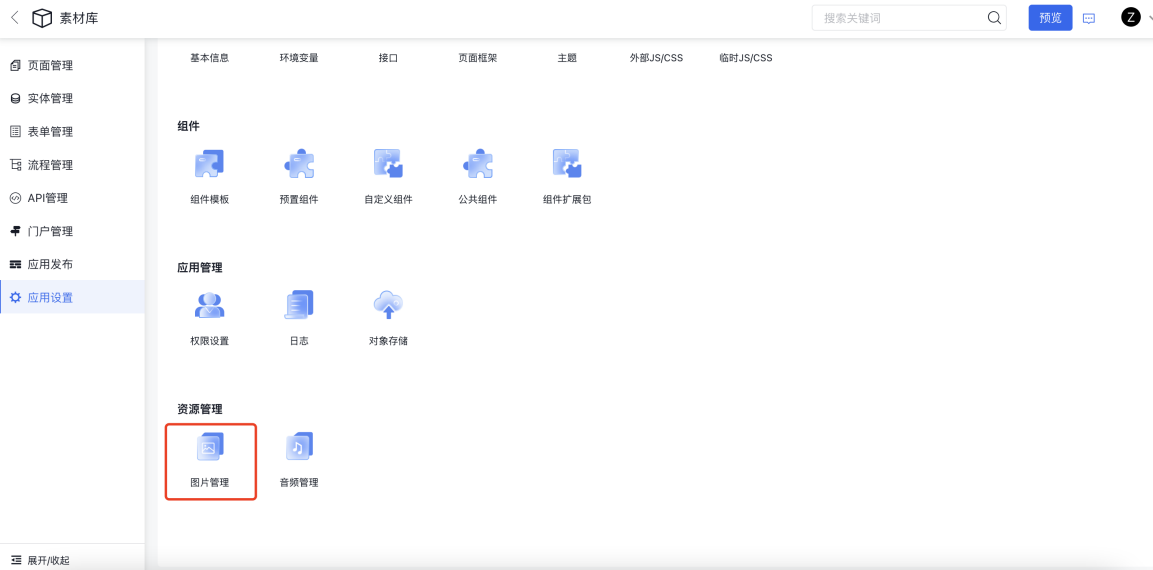
层级管理



爱速搭目前在平台级别、组织级别、以及应用级别分别有不同的入口，来进行图片文件管理。不同层级只会展示当前层级创建的分类和上传的文件。

应用级

对于与应用或业务强相关，且不需要共享给其他项目使用的图片资源，建议在应用级进行维护，应用级图片管理的入口为：应用开发 > 应用设置 > 资源管理 > 图片管理。



不同类型的应用支持的应用设置的配置项不完全相同，示例为web应用的应用设置支持的配置项。对于其他类型的应用，入口的路径是一样的。

组织级

对于需要在一个组织内共享、组织标准资源或者统一规范类的图片文件，建议在组织级进行维护。组织级图片管理功能的入口为：平台管理-资源管理-图片管理。

平台级

平台级别的图片管理更多的是公开资源，适用面会比组织级更广，平台的所有组织都可以使用。适用于集团类的公司，或其他涉及多个组织的场景。平台级资源管理的入口为：超管后台-资源管理-图片管理。

对于爱速搭私有化的客户来说，只有购买了多租户授权才能在平台上创建多个组织。

操作说明

不同层级的图片管理能力都是一样的，主要包括如下几点：

- 分类管理：分类的创建、编辑、删除
- 文件管理：图片的上传、下载、裁剪、编辑、删除、复制地址等
- 批量处理：批量删除、批量下载
- 筛选及查询：对文件列表进行筛选和检索

下面以应用级图片管理为例，详细介绍图片管理的具体功能和操作说明。

分类管理

1. 新增分类

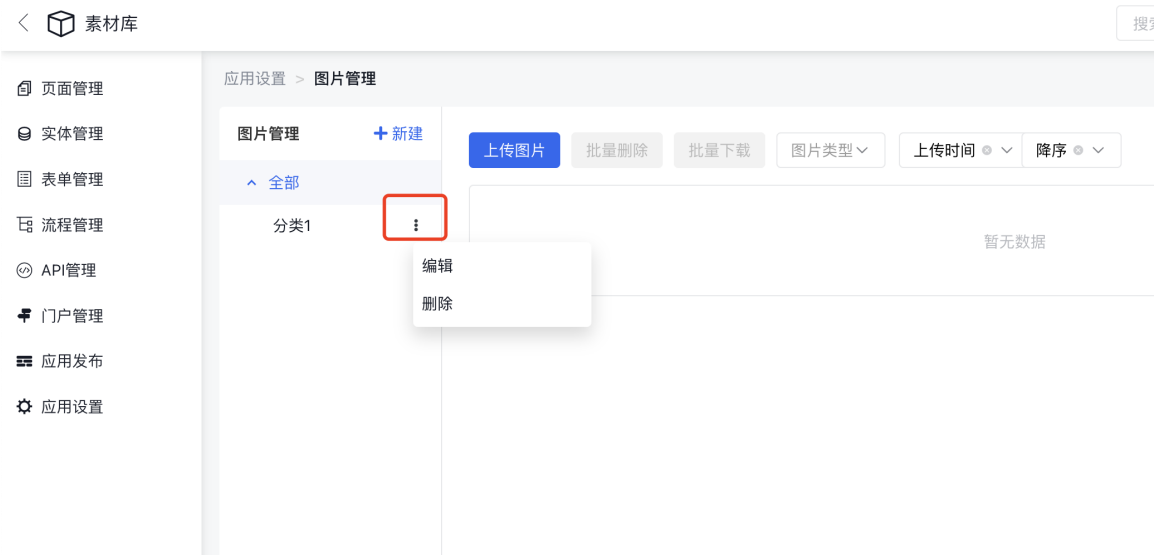
目前爱速搭支持分类的最大层级为3级，其中系统自带的「全部」为第一级，自定义创建的分类均为全部分类下的子集，在创建分类的时候，可以选择父级分类，从而实现在父级分类下创建子分类。如不选择父级分类，则默认父级分类为「全部」。



目前分类名称的长度限制为8个字符，且不支持特殊字符。

2. 编辑分类

如果需要对分类信息进行修改，可以通过分类的更多进入信息编辑页面。



3. 删除分类

如果需要对分类进行删除，默认的机制是会将删除的分类下的图片全部移入全部这个默认分类。操作入口同编辑分类。

图片上传

爱速搭支持JPG、JPEG、PNG、GIF、WEBP等格式的图片文件上传，可批量上传多个文件。

上传表单中，默认会填充当前所处的分类信息，存储位置优先为对象存储。



因为图片可能体积较大，因此推荐将上传的文件存储在对象存储（单个图片大小为10M）。当平台没有配置对象存储时，图片会存入db中（单个图片大小为5M），建议在上传前先完成对象存储配置，详细操作请参照 [对象存储管理](#)

图片操作

图片支持在线预览和编辑、下载等操作，点击更多可以展示更多操作项。

- 复制链接：一键复制图片的地址
- 编辑：可以修改分类信息，以及在不改变图片地址的情况下更新图片内容
- 裁剪：可以裁剪成自己任意的尺寸大小，以及在不改变图片地址的情况下更新图片内容
- 下载：单个文件是直接下载（多个文件批量下载会下载成一个zip包）
- 删除：对单个文件进行删除



批量处理

默认批量操作的按钮是置灰的，当选中某个文件后，激活批量操作的按钮。单个卡片的选中热区为卡片操作入口icon下方。



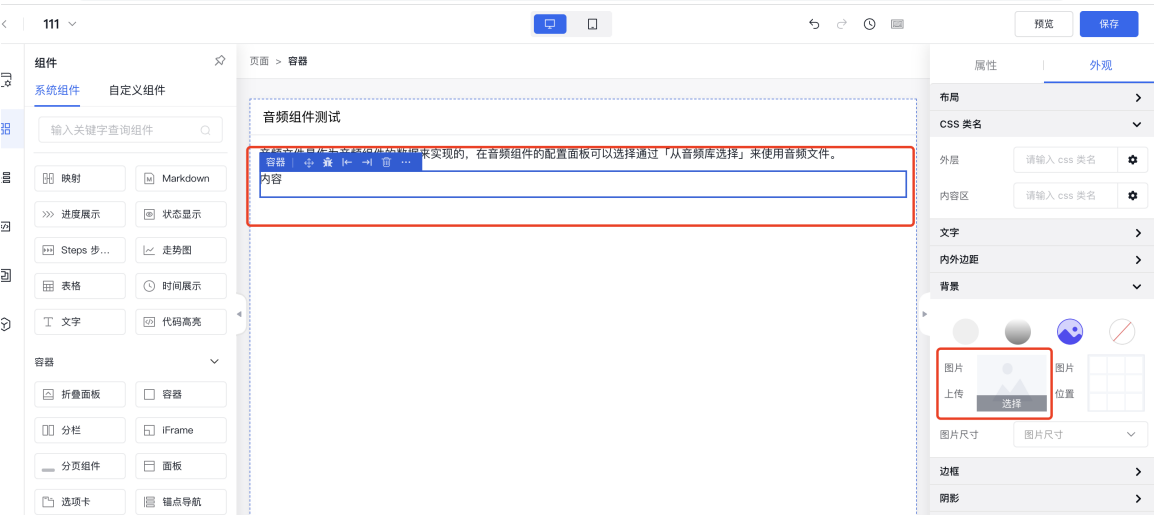
筛选及排序

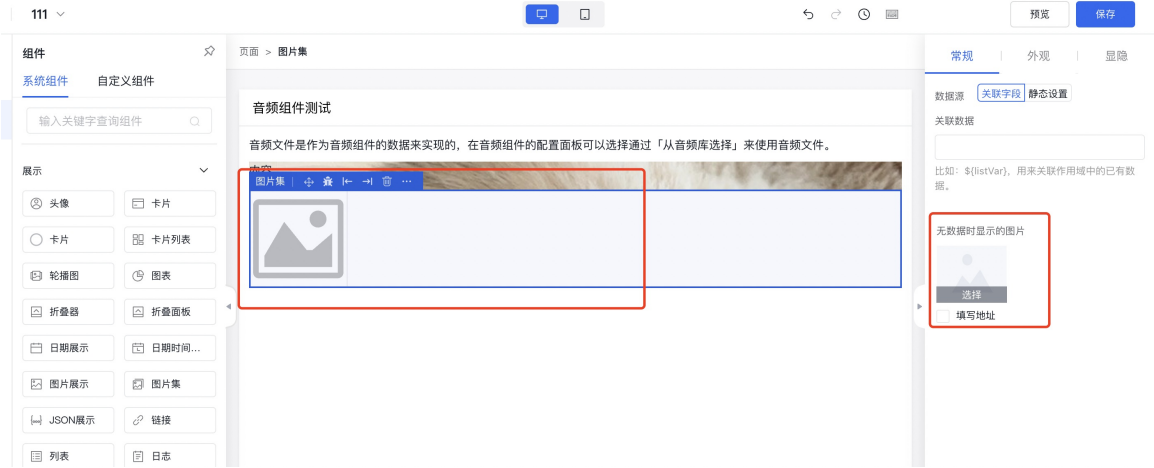
平台支持按文件格式筛选，按文件名称、上传时间、图片大小进行排序，以及支持输入文件名称模糊搜索。



🔗 图片使用

- 1. 在页面编辑器中，往画布中拖入一个带背景设置的组件例如容器组件获取图片类型的组件例如图片集，并打开该组件的配置面板

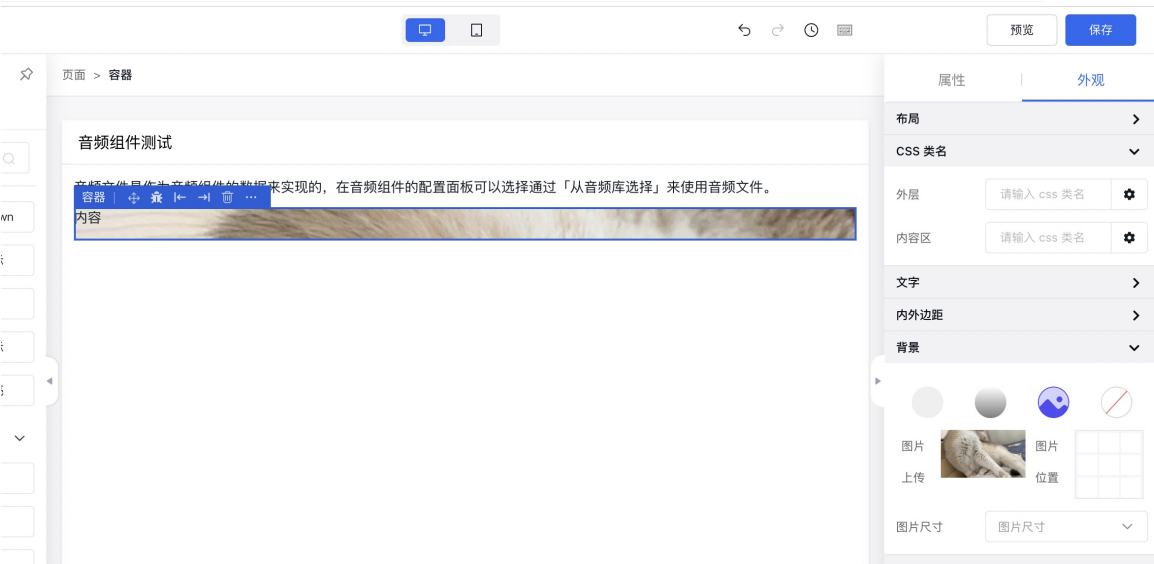




2. 点击图片组件配置面板的「选择」，打开选择面板，选择中需要的图片文件并确认提交。



3. 确认提交后可以看到看到所选图片文件的地址，到这里就完成了图片文件的使用。



在编辑器中选择文件的时候，会将平台上有权限使用的图片文件都展示出来，且分为应用级和平台级。应用级为当前应用维护的图片文件，平台级为组织级和平台超管维护的图片文件的合集，但是分类相互独立，维护各自原有的分类结构。

## 音频管理

### 音频管理

爱速搭平台支持统一资源管理，其中就包含音频资源管理。目前在平台级别、组织级别、以及应用级别都可以独立维护音频，并且音频资源会通过继承机制，实现公共资源的下发和使用。

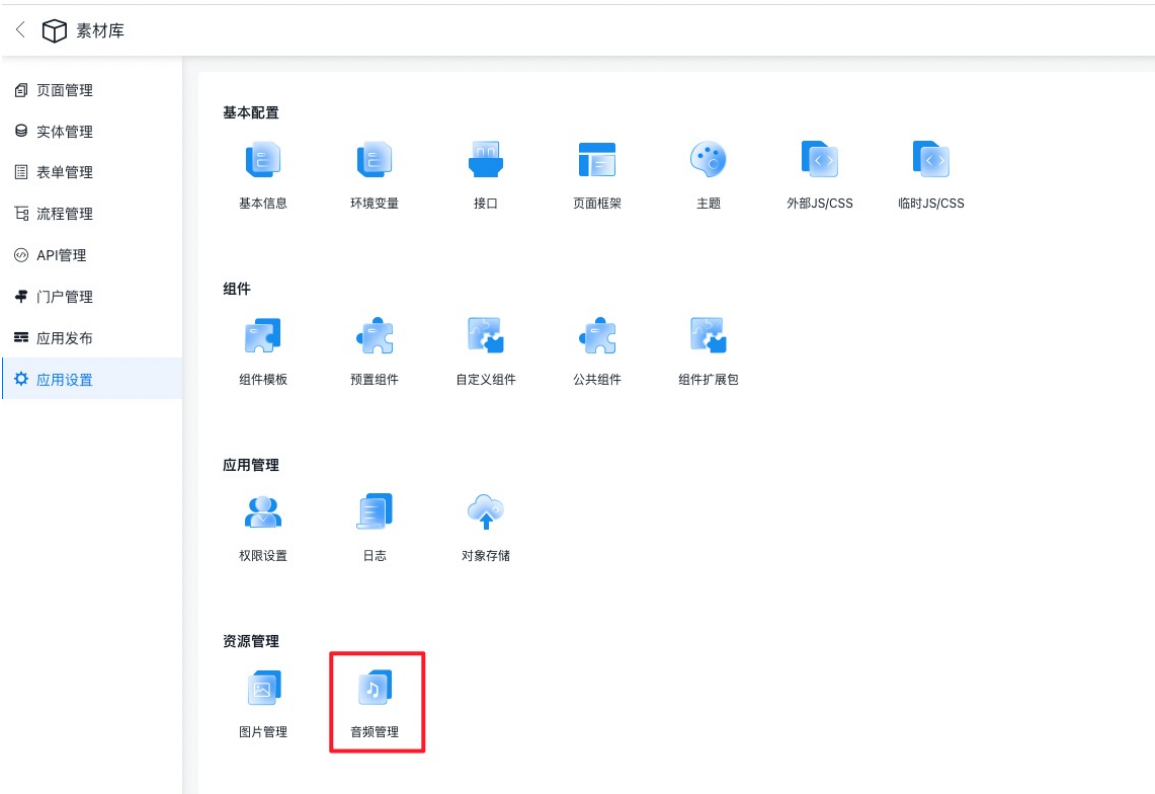
资源的管理最终是为资源的使用而服务，资源的使用始终是在开发环境下，因此组织级和应用级音频管理只在平台开发环境下才支持。

🔗 层级管理

爱速搭目前在平台级别、组织级别、以及应用级别分别有不同的入口，来进行音频文件管理。不同层级只会展示当前层级创建的分类和上传的文件。

1. 应用级

对于与应用或业务强相关，且不需要共享给其他项目使用的音频资源，建议在应用级进行维护，应用级音频管理的入口为：应用开发>应用设置>资源管理>音频管理。



不同类型的应用支持的应用设置的配置项不完全相同，示例为web应用的应用设置支持的配置项。对于其他类型的应用，入口的路径是一样的。

2. 组织级

对于需要在一个组织内共享、组织标准资源或者统一规范类的音频文件，建议在组织级进行维护。组织级音频管理功能的入口为：平台管理>资源管理>音频管理。

3. 平台级

平台级别的音频管理更多的是公开资源，适用面会比组织级更广，平台的所有组织都可以使用。适用于集团类的公司，或其他涉及多个组织的场景。平台级资源管理的入口为：超管后台>资源管理>音频管理。

对于爱速搭私有化的客户来说，只有购买了多租户授权才能在平台上创建多个组织。

🔗 操作说明

不同层级的音频管理能力都是一样的，主要包括如下几点：

- 分类管理：分类的创建、编辑、删除

- 文件管理：音频的上传、下载、播放、编辑、复制地址等
- 批量处理：批量删除、批量下载
- 筛选及查询：对文件列表进行筛选和检索

下面一应用级音频管理为例，详细介绍音频管理的具体功能和操作说明。

分类管理

1. 新增分类

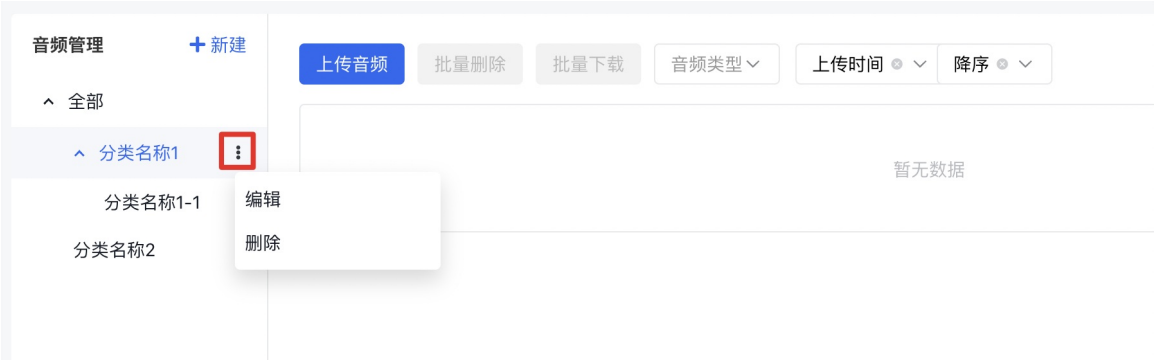
目前爱速搭支持分类的最大层级为3级，其中系统自带的「全部」为第一级，自定义创建的分类均为全部分类下的子集，在创建分类的时候，可以选择父级分类，从而实现在父级分类下创建子分类。如不选择父级分类，则默认父级分类为「全部」。



目前分类名称的长度限制为8个字符，且不支持特殊字符。

2. 编辑分类

如果需要对分类信息进行修改，可以通过分类的更多进入信息编辑页面。



3. 删除分类

如果需要对分类进行删除，默认的机制是会将删除的分类下的音频全部移入全部这个默认分类。操作入口同编辑分类。

音频上传

爱速搭支持WAV，FLAC，APE，MP3，AAC，OGG，Opus等格式的音频文件上传，可批量上传多个文件。

上传表单中，默认会填充当前所处的分类信息，存储位置优先为对象存储。

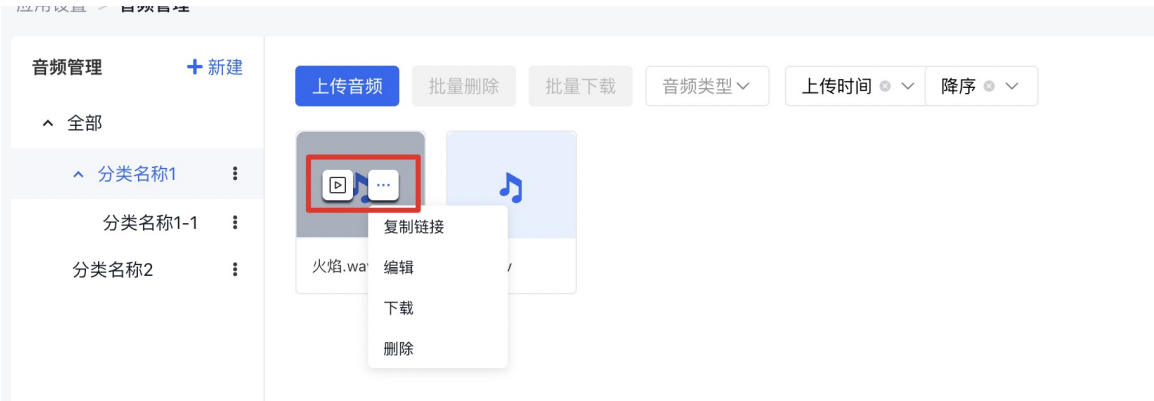


因为音频文件可能体积较大，因此推荐将上传的文件存储在对象存储（单个文件大小为20M）。当平台没有配置对象存储时，音频文件会存入db中（单个文件大小为5M），建议在上传前先完成对象存储配置，详细操作请参照[对象存储管理](#)。

音频操作

音频支持在线播放和编辑、下载等操作，点击更多可以展示更多操作项。

- 复制链接：一键复制文件的地址
- 编辑：可以修改分类信息，以及在不改变文件地址的情况下更新文件内容
- 下载：单个文件是直接下载（多个文件批量下载会下载成一个zip包）
- 删除：对单个文件进行删除



批量处理

默认批量操作的按钮是置灰的，当选中某个文件后，激活批量操作的按钮。单个卡片的选中热区为卡片操作入口icon下方。





筛选及排序

平台支持按文件格式筛选，按文件名称、上传时间、音频大小进行排序，以及支持输入文件名称模糊搜索。

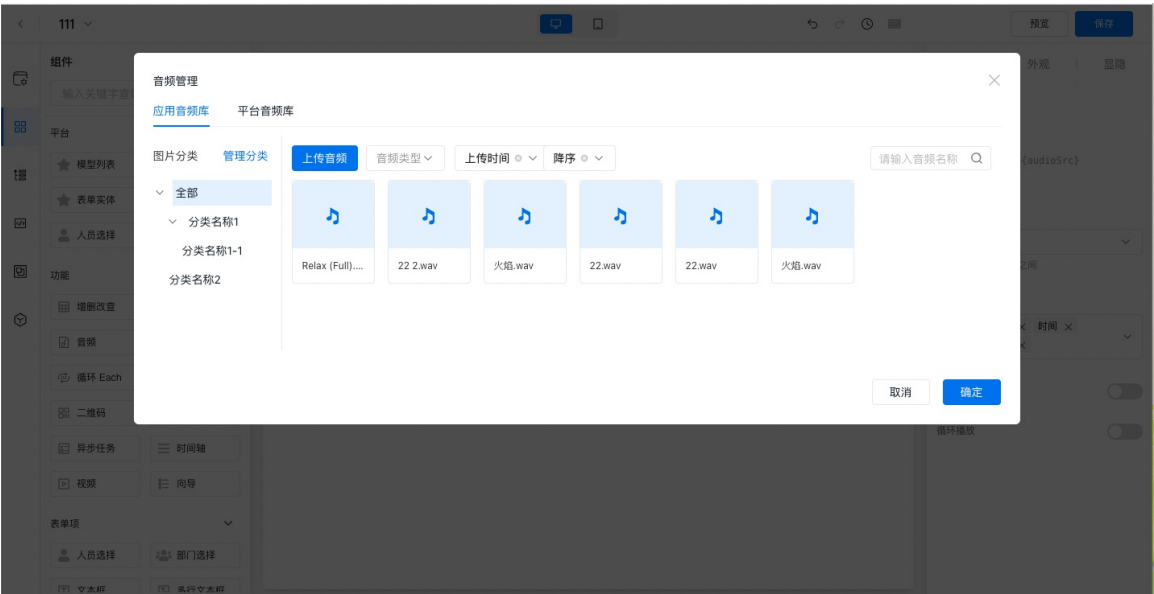


音频使用

1.在页面编辑器中，往画布中拖入一个音频组件，并打开该组件的配置面板



2. 点击音频组件配置面板的「从音频库选择」，打开选择面板，选择中需要的音频文件并确认提交。



3. 确认提交后可以看到看到所选音频文件的地址，到这里就完成了音频文件的使用。



在编辑器中选择文件的时候，会将平台上有权限使用的音频文件都展示出来，且分为应用级和平台级。应用级为当前应用维护的音频文件，平台级为组织级和平台超管维护的音频文件的合集，但是分类相互独立，维护各自原有的分类结构。

## 典型实践

### 配置一个列表

这篇文档介绍如何配置通过接口展示一个表格组件。

添加「增删改查」组件 进入页面的可视化编辑器界面，点击加号，新增组件：

选择「增删改查」组件： 进入快速构建，可以快速编辑要展示的列：

每列所配置属性作用分别是：

设置表头显示字段 该列绑定行数据中字段的 key 值 选择列展示的类型 我们这里创建三列，分别是：「ID」列（绑定行数据 id 字段）、「姓名」列（绑定行数据 name 字段）、「邮箱」列（绑定行数据 email 字段）。

基本的组件结构已经准备好了，现在我们需要准备数据

准备数据接口 准备好一个可用的数据接口，该数据接口返回的数据格式要求如下：

```
{ "status": 0, "msg": "", "data": { "items": [ { // 每一行的数据 "id": 1, "xxx": "xxxx" } ],
```

```
    "count": 200 // 注意！！！这里不是当前请求返回的 items 的长度，而是数据库中一共有多少条数据，用于生成分页组件
```

```
}} items或rows：用于返回数据源数据，格式是数组 count: 用于返回数据库中一共有多少条数据，用于生成分页 例如如上，我
```

们配置的表格，我们需要返回的数据结构是：

```
{ "status": 0, "msg": "", "data": { "items": [ { "id": 1, "name": "张三", "email": "zhangsan@baidu.com" }, { "id": 2, "name": "李四", "email": "lisi@baidu.com" } ], "count": 2 } }
```

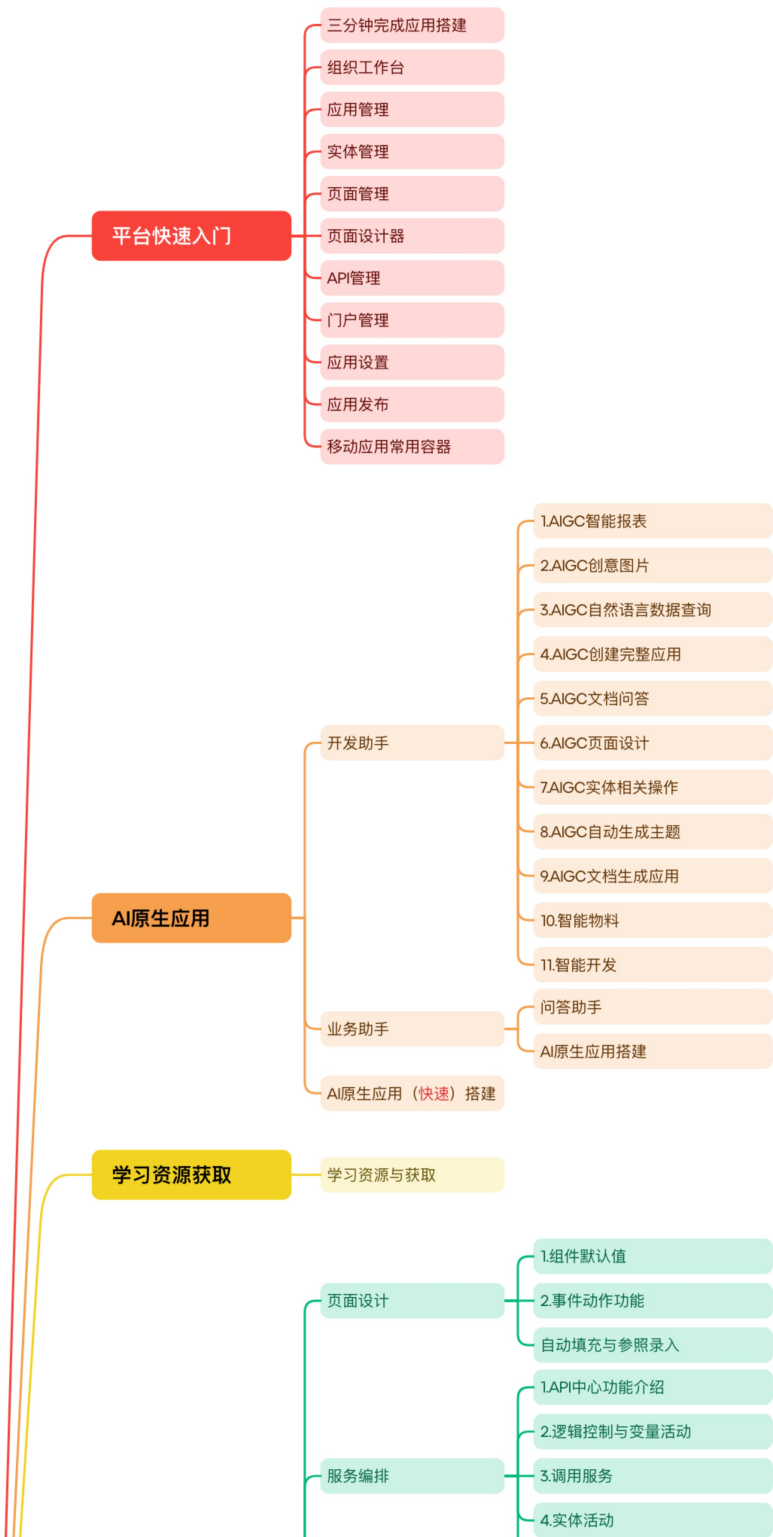
绑定数据接口 在编辑器左侧大纲中，选中「增删改查」组件，切换至「接口」标签，将准备好的数据接口，填入到「数据拉取接口」配置中

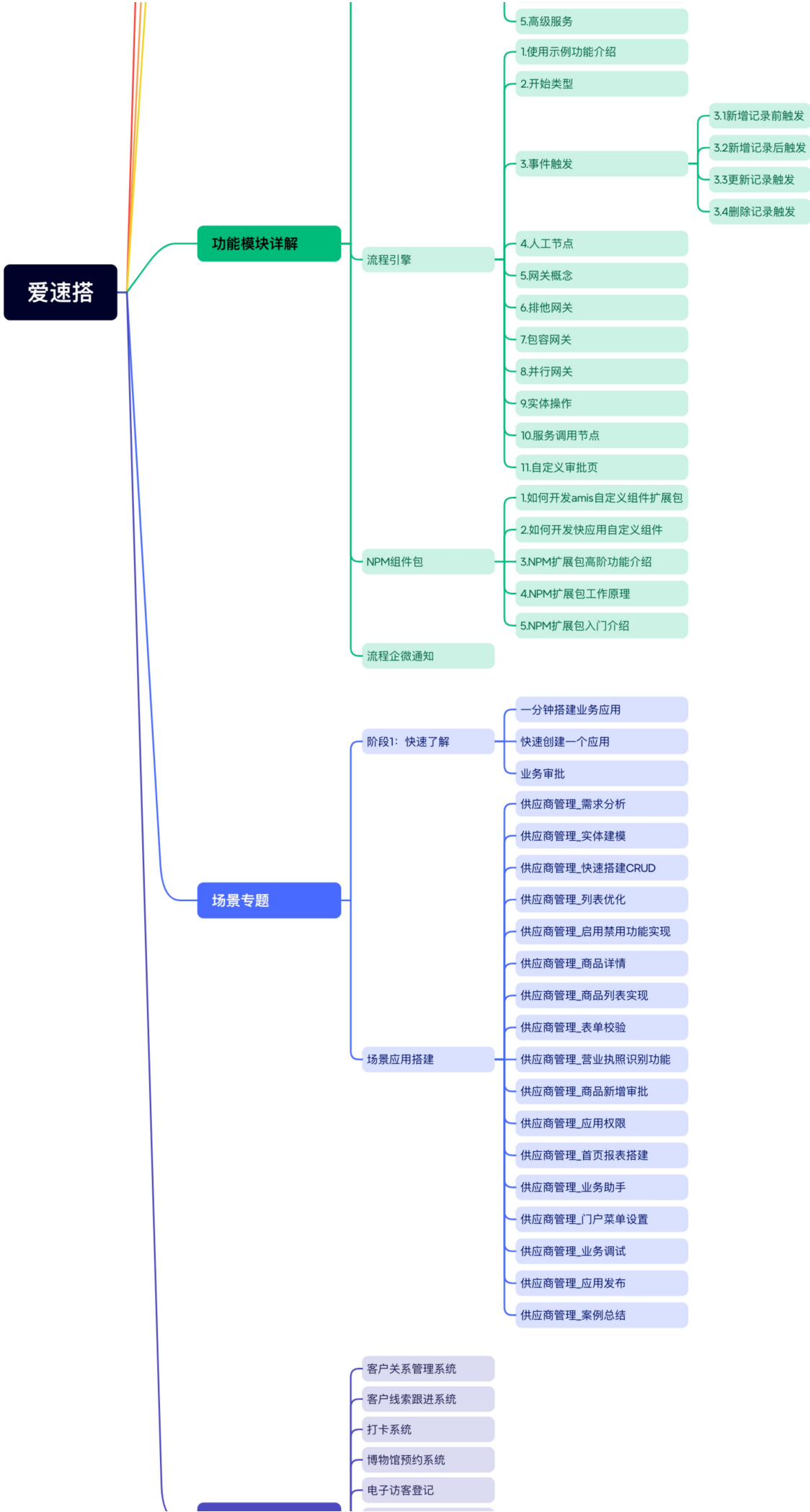
点击预览可以看到已经可以顺利渲染出列表页了。

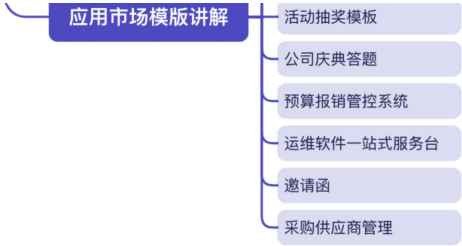
image.png

# 视频教程

## 教学视频大纲







Presented with xmind

# 平台快速入门

## 三分钟完成应用搭建

🔗 三分钟完成应用搭建

演示分钟级完成应用系统开发的操作过程。

### 组织工作台

🔗 组织工作台

本视频主要介绍组织工作台相关功能内容；

### 实体管理

🔗 实体管理

本视频主要介绍实体管理相关功能内容；

### 页面管理

🔗 页面管理

本视频主要介绍页面管理相关功能内容；

### 门户管理

#### 门户管理

本视频讲解，门户管理相关功能。

### 页面设计器

🔗 页面设计器

本视频主要介绍页面设计器相关功能内容；

### 应用管理

🔗 应用管理

本视频主要介绍工作台 -> 应用管理相关功能内容；

### 应用设置

🔗 应用设置

本视频主要介绍应用设置相关功能内容；

应用发布

应用发布

本视频主要介绍应用发布相关功能内容；

API管理

API管理

本视频主要介绍API管理相关功能内容；

移动应用常用容器

移动应用常用容器

本视频介绍移动应用常用容器的使用方式

AI原生应用

开发助手

1.AIGC智能报表

AIGC智能报表

本视频主要介绍AIGC智能报表生成功能。

2.AIGC创意图片

AIGC创意图片

本视频主要介绍AIGC创意图片功能。

3.AIGC自然语言数据查询

AIGC自然语言数据查询

本视频主要介绍如何使用AIGC进行自然语言数据查询。

4.AIGC创建完整应用

AIGC创建完整应用

本视频主要介绍爱速搭AIGC创建完整应用。

5.AIGC文档问答

AIGC文档问答

本视频主要介绍AIGC文档问答功能。

6.AIGC页面设计

AIGC页面设计

本视频主要介绍AIGC页面设计相关功能。

7.AIGC实体相关操作

AIGC实体相关操作

本视频主要介绍AIGC实体相关操作。

## 🔗 8.AIGC自动生成主题

### **AIGC自动生成主题**

本视频主要介绍AIGC自动生成主题功能。

## 🔗 9.AIGC文档生成应用

### **AIGC文档生成应用**

本视频主要介绍如何使用AIGC文档生成应用功能。

## 🔗 10.AIGC智能物料

### **智能物料**

通过对话快速生成个性化的物料需求，高效地产出主题内容并智能生成报表，从一键配置应用所需物料，提高工作效率。

## 🔗 11.AIGC智能开发

### **AIGC智能开发**

AI辅助开发，借助大模型的智能分析能力，更快速地创建传统业务系统，降低应用构建门槛。

## 🔗 12.AI智能引擎

### **AI智能引擎**

系统内置多样ai智能引擎,可自由配置所需能力，驱动应用智能化。

## 🔗 13.ai搭建-图书馆管理系统

### **ai搭建-图书馆管理系统**

本视频主要演示ai搭建-场景应用搭建-图书馆管理系统的搭建过程

## 🔗 14.ai搭建-CRM系统

### **ai搭建-场景应用搭建-CRM系统**

本视频主要介绍ai搭建-场景应用搭建-CRM系统

### **业务助手**

## 🔗 AI原生应用搭建

### **AI原生应用场景搭建**

本视频演示爱速搭AI原生应用的搭建，搭建与资产管理场景配套的业务助理。

## 🔗 问答助手

### **问答助手**

数据查询利器，助力用户查询所需信息，满足各类业务查询需求；提供智能问答助手，为用户解答疑问、排除使用障碍。

## 🔗 业务助手效果演示

### **业务助手效果演示**

本视频主要介绍业务助手生产环境（资产助手演示）效果进行演示

## AI原生应用快速搭建

🔗 AI原生应用快速搭建

本视频主要介绍两分钟快速搭建，AI智能对话应用。

## 学习资源获取

学习资源获取

## 学习资源获取

本视频主要爱速搭平台，学习资源获取方式。

## 功能模块详解

NPM组件包

🔗 1.如何开发amis自定义组件扩展包

**如何开发amis自定义组件扩展包** 本视频对如何开发amis自定义组件扩展包进行了介绍

🔗 2.如何开发快应用自定义组件

**如何开发快应用自定义组件** 本视频介绍如何开发快应用自定义组件

🔗 3.NPM扩展包高阶功能介绍

## NPM扩展包高阶功能介绍

本视频主要对NPM扩展包高阶功能进行介绍；

🔗 4.NPM扩展包工作原理

**NPM扩展包工作原理**

本视频主要介绍NPM扩展包工作原理；

🔗 5.NPM扩展包入门介绍

**NPM扩展包入门介绍** 本视频对npm扩展包进行了入门介绍

## 流程引擎

🔗 1.使用示例功能介绍

**流程引擎1\_使用示例功能介绍**

本视频主要介绍使用示例功能介绍相关功能内容；

🔗 2.开始类型

**流程引擎2\_开始类型功能介绍**

本视频主要介绍开始类型功能介绍相关功能内容；

🔗 3.事件触发

🔗 3.1新增记录前触发

**流程引擎3-1\_新增记录前触发功能介绍**

本视频主要介绍新增记录前触发功能介绍相关功能内容；



### 🔗 3.2新增记录后触发

#### 流程引擎3-2\_新增记录后触发功能介绍

本视频主要介绍新增记录后触发功能介绍相关功能内容；

### 🔗 3.3更新记录触发

#### 流程引擎3-3\_更新记录触发功能介绍

本视频主要介绍更新记录触发功能介绍相关功能内容；

### 🔗 3.4删除记录触发

#### 流程引擎3-4\_删除记录触发功能介绍

本视频主要介绍删除记录触发功能介绍相关功能内容；

### 🔗 4.人工节点

#### 流程引擎4\_人工节点功能介绍

本视频主要介绍人工节点功能介绍相关功能内容；

### 🔗 5.网关概念

#### 流程引擎5\_网关概念功能介绍

本视频主要介绍网关概念相关功能内容；

### 🔗 6.排他网关

#### 流程引擎6\_排他网关功能介绍

本视频主要介绍排他网关功能介绍相关功能内容；

### 🔗 7.包容网关

#### 流程引擎7\_包容网关功能介绍

本视频主要介绍包容网关功能介绍相关功能内容；

### 🔗 8.并行网关

#### 流程引擎8\_并行网关功能介绍

本视频主要介绍并行网关功能介绍相关功能内容；

### 🔗 9.实体操作

#### 流程引擎9\_实体操作功能介绍

本视频主要介绍事件实体操作介绍相关功能内容；

### 🔗 10.服务调用节点

#### 流程引擎10\_服务调用节点功能介绍

本视频主要介绍服务调用节点功能介绍相关功能内容；

### 🔗 11.自定义审批页

#### 流程引擎11\_自定义审批页功能介绍

本视频主要介绍自定义审批页功能介绍相关功能内容；

## 流程企微通知

### 🔗 流程企微通知

本视频介绍了流程设计中，如何配置企微通知功能

## 服务编排

### 🔗 1.API中心功能介绍

#### 服务编排1\_API中心功能介绍

本视频主要介绍API中心功能介绍相关功能内容；

### 🔗 2.逻辑控制与变量活动

#### 服务编排2\_逻辑控制与变量活动

本视频主要介绍逻辑控制与变量活动相关功能内容；

### 🔗 3.调用服务

#### 服务编排3\_调用服务

本视频主要介绍调用服务相关功能内容；

### 🔗 4.实体活动

#### 服务编排4\_实体活动

本视频主要介绍实体活动相关功能内容；

### 🔗 5.高级服务

#### 服务编排5\_高级服务功能介绍

本视频主要介绍高级服务相关功能内容；

## 页面设计

### 🔗 1.组件默认值

#### 页面设计1\_组件默认值功能介绍

本视频主要介绍组件默认值功能介绍相关功能内容；

### 🔗 2.事件动作功能

#### 页面设计3\_事件动作功能介绍

本视频主要介绍爱速搭页面的事件交互，事件动作相关内容讲解；

### 🔗 3.自动填充与参照录入

#### 页面设计3\_自动填充与参照录入功能介绍

本视频主要介绍自动填充与参照录入功能介绍相关功能内容；

## 高级功能

### 🔗 源码导出演示

## 源码导出演示

本视频主要介绍爱速搭源码导出

### 分支管理

#### 分支管理

本视频主要介绍爱速搭分支管理功能

## 场景专题

### 阶段1:快速了解

#### 一分钟搭建业务应用

##### 一分钟搭建业务应用

本视频介绍如何使用开发助手一分钟搭建应用。

#### 业务审批

##### 业务审批

本视频主要介绍爱速搭集成业务审批；

#### 快速创建一个应用

## 快速创建一个应用

本视频主要介绍爱速搭快速搭建应用的过程。创建应用、设计数据模型、搭建页面、发布应用

### 场景应用搭建

#### 供应商管理-1.需求分析

##### 供应商管理-1.需求分析

本视频介绍供应商管理系统搭建后，搭建的功能以及效果

#### 供应商管理-2.实体建模

##### 供应商管理-2.实体建模

本视频介绍供应商管理系统搭建实体建模相关演示

#### 供应商管理-3.快速搭建CRUD

##### 供应商管理-3.快速搭建CRUD

本视频介绍供应商管理系统如何快速搭建CRUD

#### 供应商管理-4.列表优化

##### 供应商管理-4.列表优化

本视频介绍供应商管理系统对搭建列表进行优化

#### 供应商管理-5.启用禁用功能实现

##### 供应商管理-5.启用禁用功能实现

本视频介绍供应商管理系统如何实现供应商状态的启用禁用功能

🔗 供应商管理-6.商品详情

供应商管理-6.商品详情

本视频介绍供应商管理系统如何配置供应商商品详情页面

🔗 供应商管理-7.商品列表实现

供应商管理-7.商品列表实现

本视频介绍供应商管理系统商品列表的实现

🔗 供应商管理-8.表单项校验

供应商管理-8.表单项校验

本视频介绍供应商管理系统表单项校验功能

🔗 供应商管理-9.营业执照识别功能

供应商管理-9.营业执照识别功能

本视频介绍供应商管理系统完成营业执照识别功能搭建

🔗 供应商管理-10.商品新增审批

供应商管理-10.商品新增审批

本视频介绍供应商管理系统设置商品新增需要审批功能

🔗 供应商管理-11.应用权限

供应商管理-11.应用权限

本视频介绍供应商管理系统如何设置应用权限

🔗 供应商管理-12.首页报表搭建

供应商管理-12.首页报表搭建

本视频介绍供应商管理系统如何快速搭建首页报表

🔗 供应商管理-13.业务助手

供应商管理-13.业务助手

本视频介绍供应商管理系统搭建中的业务助手

🔗 供应商管理-14.门户菜单设置

供应商管理-14.门户菜单设置

本视频介绍供应商管理系统如何设置门户

🔗 供应商管理-15.业务调试

供应商管理-15.业务调试

本视频学习供应商管理系统可以进行业务调试的方式

🔗 供应商管理-16.应用发布

供应商管理-16.应用发布

本视频介绍供应商管理系统如何进行应用发布

🔗 供应商管理-17.案例总结

供应商管理案例总结

本视频介绍供应商管理大纲案例总结

应用市场模板操作讲解

采购供应商管理

采购供应商管理

本视频对应用市场模版采购供应商管理的使用进行讲解，您可以到爱速搭官网->模板中心，进行查看及体验模板。

博物馆预约系统

🔗 博物馆预约系统

本视频对应用市场模版博物馆预约系统的使用进行讲解，您可以到爱速搭官网->模板中心，进行查看及体验模板。

电子化访客登记

🔗 电子化访客登记

本视频对应用市场模版电子化访客登记的使用进行讲解，您可以到爱速搭官网->模板中心，进行查看及体验模板。

！

公司庆典答题操作讲解

🔗 公司庆典答题操作讲解

本视频对应用市场的公司庆典答题系统进行了介绍；可以到爱速搭官网->模板中心，进行查看及体验模板具体内容。

运维软件一站式服务台

**运维软件一站式服务台** 本视频对运维软件一站式服务台的使用进行讲解，您可以到爱速搭官网->模板中心，进行查看及体验模板。

客户线索跟进系统

🔗 客户线索跟进系统

本视频对客户线索跟进系统的使用操作进行讲解，您可以到爱速搭官网->模板中心，进行查看及体验模板。

邀请函

🔗 邀请函

本视频对应用市场模版邀请函的使用进行讲解，您可以到爱速搭官网->模板中心，进行查看及体验模板具体内容。

！

预算报销管控系统

🔗 预算报销管控系统

本视频对应用市场模板的预算报销管控系统进行操作讲解，可以到爱速搭官网->模板中心，进行查看及体验模板具体内容。

客户关系管理系统

## 🔗 客户关系管理系统

本视频对客户关系管理系统的使用进行讲解，您可以到[爱速搭官网->模板中心](#)，进行查看及体验模板。

## 员工打卡系统

### 🔗 员工打卡系统

本视频对应用市场模版员工打卡系统的使用进行讲解，您可以到[爱速搭官网->模板中心](#)，进行查看及体验模板。

## 活动抽奖模板

### 🔗 活动抽奖模板

本视频对应用市场模版「活动抽奖模板」系统的使用进行讲解，您可以到[爱速搭官网->模板中心](#)，进行查看及体验模板。

## 爱速搭考核与知识管理系统

### 🔗 爱速搭考核与知识管理系统

本视频对应用市场模版「爱速搭考核与知识管理系统」系统的使用进行讲解，您可以到[爱速搭官网->模板中心](#)，进行查看及体验模板。

## 零代码

### 零代码应用

### 🔗 零代码应用搭建

本视频主要介绍如何零代码搭建应用

# 常见问题

## 私有部署相关问题

### 🔗 爱速搭是否支持私有部署？

爱速搭支持私有部署，您可以部署在自己的内网，私有部署版本具备 SaaS 版本的所有功能

### 🔗 私有部署对环境有什么要求？

需要您的机器安装 Docker，推荐版本为18.09及以上，爱速搭可以运行在单机 Docker 环境上，如果需要集群化、高可用，可以使用 Docker 自带的 swarm 或者 k8s。

Docker 的安装建议参考官方文档，可以安装在Centos、Ubuntu、Windows、Mac等系统上。

生产环境推荐使用 Linux 操作系统，最小机器配置建议是 4 核 4G 内存。

### 🔗 如何部署在某个目录下？

默认情况下爱速搭需要使用独立域名，如果条件不允许，还可以通过如下环境变量让爱速搭运行在某个目录下，比如下面的配置：

```
ISUDA_CONTEXT_PATH: '/aisuda'
```

这样就可以通过 /aisuda 这个目录来访问爱速搭平台了。

### 🔗 如何在无网环境安装？

首先是 Docker，Windows 和 Mac 可以通过上面的安装包离线安装，而 Linux 需要参考[这里](#)下载二进制文件来安装。

其次是 Docker-Compose，参考文档，主要是在有网环境下，在这里下载对应环境的二进制文件。

接下来需要将镜像也保存为文件，方法是找一台能联网且有 Docker 的机器，运行如下命令：

```
docker pull registry.baidubce.com/aisuda/aisuda:1.3.2
```

```
docker save -o suda.tar registry.baidubce.com/aisuda/aisuda:1.3.2
```

```
docker save -o mysql.tar registry.baidubce.com/aisuda/mysql:8
```

```
docker save -o redis.tar registry.baidubce.com/aisuda/redis:5
```

将这三个文件上传到需要安装的服务器上，执行如下命令：

```
docker load -i suda.tar
```

```
docker load -i mysql.tar
```

```
docker load -i redis.tar
```

同样通过类似的 `docker load -i amis.tar` 命令来加载。amis 镜像内部端口是 8888。