

腾讯云微搭低代码

操作指南

产品文档



腾讯云

【版权声明】

© 2013-2023 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

操作指南

应用

- 创建应用
- 发布应用
- 应用的导入与导出
- 应用自定义域名

编辑器

- 编辑器介绍
- 页面管理
 - 页面及应用生命周期
 - 页面跳转传参配置
- 表达式
- 行为事件
 - 事件介绍
 - 执行动作
 - 触发条件及 event 对象
 - 页面事件流
- 自定义变量
- 自定义 JavaScript 方法
- Query 数据查询介绍
- 代码编辑器
- 外部资源（js/css）加载

组件

- 官方组件列表
- 组件数据
 - 获取组件属性值
 - 组件绑定数据
 - 数据筛选条件
- 组件动态显隐
- 组件样式配置
- 区块使用说明
 - 协议声明区块
- 自定义组件
 - 自定义组件库管理
 - 组件接入控制台

数据源

- 数据源概述
- 数据模型
 - 数据模型概述
 - 标准数据模型
 - 概述
 - 新建标准数据模型
 - 系统数据模型
 - 数据模型字段
 - 使用和管理数据模型
 - 数据模型方法
 - 数据模型方法V2
 - 简单与复杂查询
 - 管理数据
 - 服务端表达式

APIs

开放服务

API 概述

新建 API

使用和管理 API

腾讯文档

腾讯会议

腾讯地图

微信支付

腾讯云文字识别

腾讯云短信

腾讯云邮件

企业微信

自定义 API

自定义 API 概述

新建自定义 API

使用和管理自定义 API

自定义代码

图表统计方法参数规范

通用选项集

概述

查询通用选项集选项值

连接器

MySQL 连接器

工作流

工作流概述

流程的组成

操作步骤

消息订阅功能介绍

流程触发方式

数据模型触发

定时触发

页面事件触发

自定义 API 触发

流程元素

基础元素

对象活动

人工任务

高级活动

API

业务场景实践

请假审批流程配置示例

审批流

流程审批页面定制化

用户权限

概述

用户管理

企业内部用户

组织架构

企业外部用户

用户登录

自定义应用登录

小程序端登录

PC 或 H5 端登录

权限设置
安全设置
企业工作台登录
获取当前登录用户信息
角色与权限
 预制角色
 新建角色
 编辑角色
 删除角色
 查看并定义角色
 应用访问权限
 页面访问权限
 数据模型权限
 APIS 接口权限
 流程访问权限
 企业工作台访问权限
发布角色
运行态使用角色权限
权限组
身份源管理
 概述
 Oauth 协议
 Saml 协议
开发者权限

操作指南

应用

创建应用

最近更新时间：2023-08-03 18:46:54

应用类型

应用分为自定义应用和数据模型应用，两类应用的适用场景如下：

- **自定义应用：**支持公开访问或外部用户注册登录，适用于官网建站、问卷调查、电商团购等 C 端业务场景。
- **数据模型应用：**支持内部用户登录访问，应用支持展示在 [企业工作台](#)，适用于协同办公、流程审批等业务场景。

目前，我们正在迭代合并两种应用类型，未来微搭将只提供一种能力统一、体验优雅的通用型应用，您无需再纠结应该创建什么类型的应用，敬请期待！

新建空白应用

在 [微搭低代码控制台](#) 中访问创建应用 > 从空白新建模块，可创建空白应用，从0开始搭建应用。



从模板新建应用

在控制台中访问创建应用 > 从模板新建模块，可以选择各类模块，创建可开箱即用的应用，同时应用中的已有内容也可以进行灵活的定制调整，匹配自身业务需求。

The screenshot shows the 'Create Application from Template' section. It features a grid of 12 mobile application templates categorized into four rows:

- Row 1:**
 - 快速开始模板 (Quick Start Template)
 - 学校预约咨询系统 (School Appointment System)
 - 企业小程序 (Enterprise Mini Program)
 - 酒水营销活动管理 (Alcohol Marketing Activity Management)
- Row 2:**
 - 电商展示模板 (E-commerce Display Template)
 - 宠物预约 (Pet Appointment)
 - 银行节日营销模板 (Bank Holiday Marketing Template)
 - 机械行业营销模板 (Mechanical Industry Marketing Template)
- Row 3:**
 - 名片展示模板 (Business Card Display Template)
 - 企业礼品管理模板 (Enterprise Gift Management Template)
 - H5 (HTML5) Template
 - PC + H5 Template
- Row 4:**
 - 小程序 + PC + H5 Template
 - 数据报表 (Data Report)
 - OCR识别 (OCR Recognition)

通过 Excel 新建应用

1. 在控制台访问创建应用 > 从场景新建模块，选择新建表单应用或新建数据管理应用。

The screenshot shows the 'Create New Application' page under the 'From Scenario' section. It highlights the 'New Form Application' (新建表单应用) and 'New Data Management Application' (新建数据管理应用) options, which are enclosed in a red box.

From Scenario

- New Portal Application**: Used for enterprise portals and official websites.
- New Form Application**: Used for collecting form data.
- New Data Management Application**: Used for data management后台.

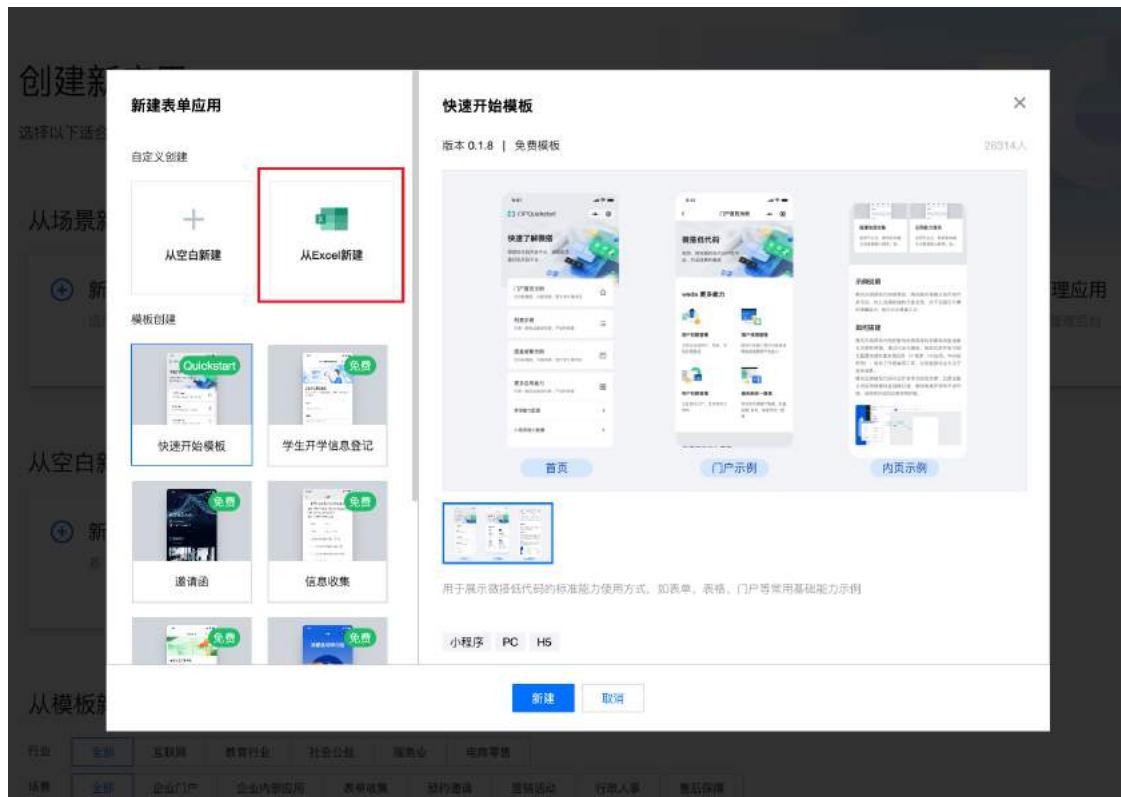
From Blank

- New Data Model Application**: Based on data models to generate improved pages.
- New Custom Application**: Starts from a blank page.

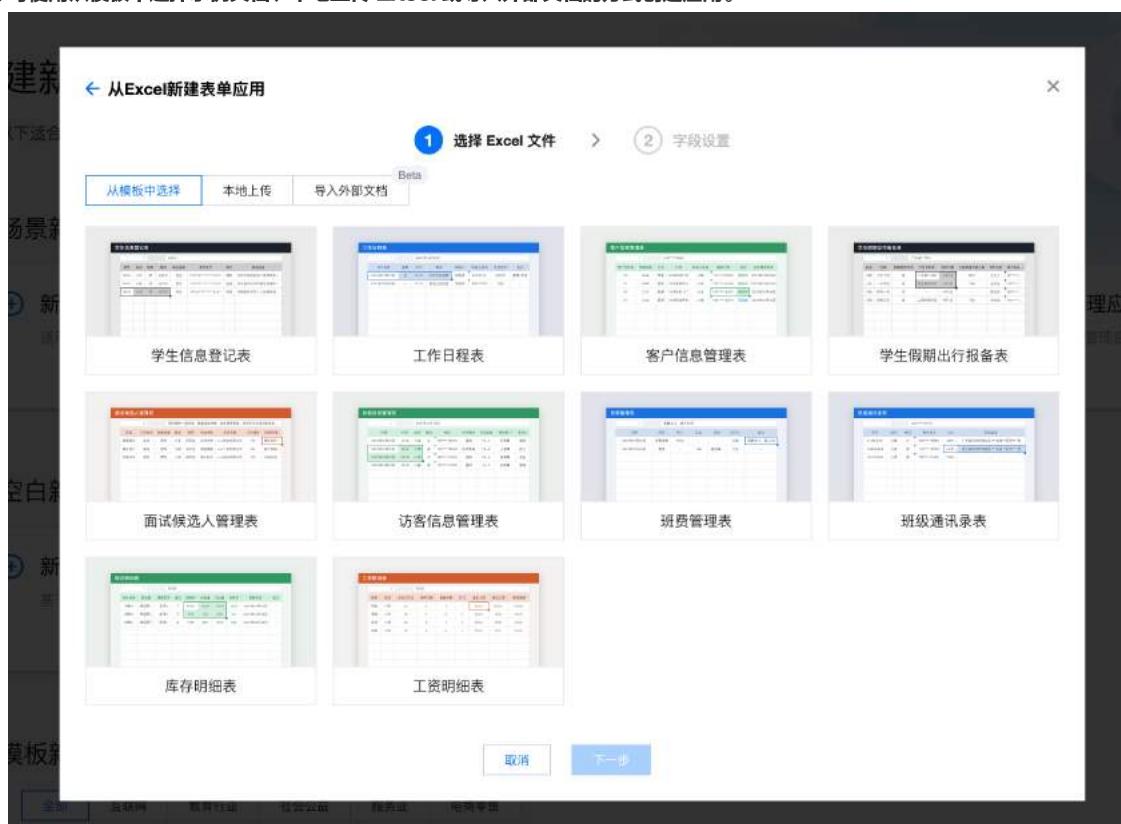
From Template

Filter options at the top include: 行业 (Industry), 场景 (Scenario), and 功能 (Function).

2. 在新建弹窗中，选择从 Excel 新建。



3. 可使用从模板中选择示例文档、本地上传 Excel 或导入外部文档的方式创建应用。



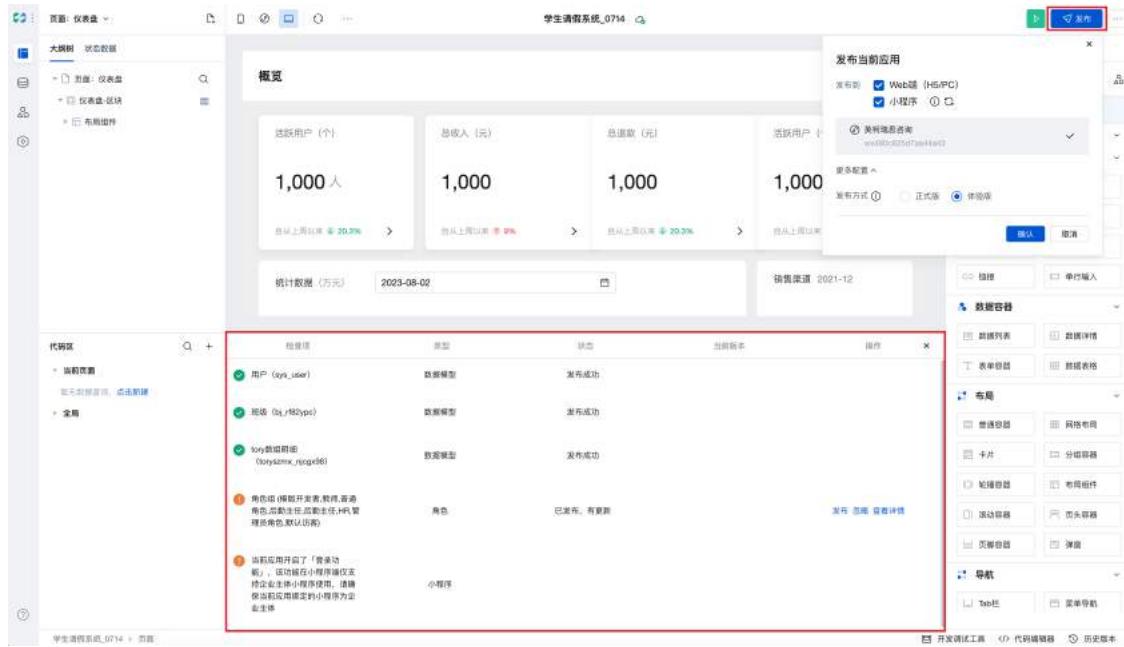
发布应用

最近更新时间：2023-08-04 09:11:05

操作步骤

应用开发完成后，单击编辑器右上角发布，即可进入应用发布流程。

1. 编辑器的检查机制会对应用发布依赖的组件、角色、数据源等进行检查，并展示出检查提示列表，告知开发者具体的警告或错误信息和相关操作项，开发者可根据实际情况处理问题，或进行忽略操作。



2. 检查通过后，用户可选择发布的端类型，支持发布到：Web 端 (H5/PC) 和小程序端。



3. 应用的发布方式可选择发布为正式版或体验版，两种发布方式的使用场景如下：

发布方式	差异点
体验版应用	<ul style="list-style-type: none"> 1. 适用于应用正式上线前，开发者对应用进行体验测试。 2. 会展示体验环境下的数据模型数据，应用中对数据进行的增、删、改操作，也会更新到体验环境数据中。 3. 数据模型、角色、工作流等资源无需发布，应用中也能正常使用。
正式版应用	<ul style="list-style-type: none"> 1. 适用于应用开发测试完成后，应用正式上线并提供给用户使用。 2. 会展示正式环境下的数据模型数据（如正式版应用中数据展示为空，请有限检查正式环境中是否已录入所需数据），应用中对数据进行的增删改操作，也会更新到正式环境数据中。 3. 数据模型、角色、工作流等资源均需发布，才可在应用中正常使用。



4. 应用发布成功后，可查看到应用的二维码和 URL 链接，可进行扫码访问，或单击链接跳转访问。



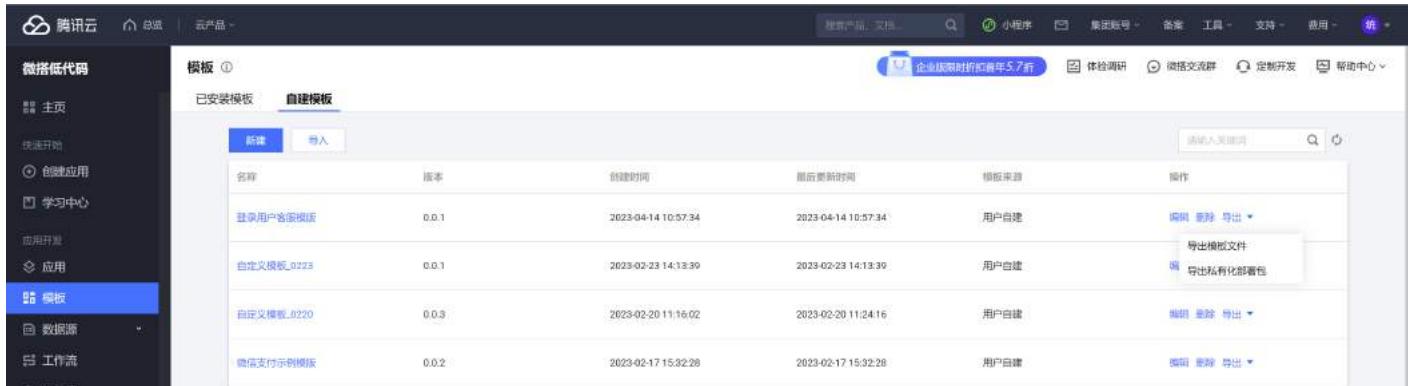
应用的导入与导出

最近更新时间：2023-05-06 10:44:56

本文将为您介绍导出应用至本地的方法以及通过上传模板文件导入应用的方法。您可以通过自定义模板的导入与导出来实现跨账号的应用流转。自定义模板相关功能可参见 [自定义模板](#)。

导出应用

1. 打开腾讯云微搭低代码控制台，进入 [模板](#) > [自建模板](#) 页面，选择需要导出的应用所关联的模板，单击导出操作。



名称	版本	创建时间	最后更新时间	模板来源	操作
登录用户鉴权模板	0.0.1	2023-04-14 10:57:34	2023-04-14 10:57:34	用户自建	编辑 删除 导出
自定义模板_0223	0.0.1	2023-02-23 14:13:39	2023-02-23 14:13:39	用户自建	编辑 删除 导出模板文件 导出私有化部署包
自定义模板_0220	0.0.3	2023-02-20 11:16:02	2023-02-20 11:24:16	用户自建	编辑 删除 导出
微信支付示例模板	0.0.2	2023-02-17 15:32:28	2023-02-17 15:32:28	用户自建	编辑 删除 导出

2. 在弹窗中输入导出版本号，单击导出按钮即可将模板以及模板关联的所有应用以文件压缩包的形式下载至本地。



① 导出前需注意

- 导出前模板内容需发布为“正式版本”，否则导出后可能不生效
- 导出文件为zip格式，导出后将自动命名为“模板名称_版本号.zip”

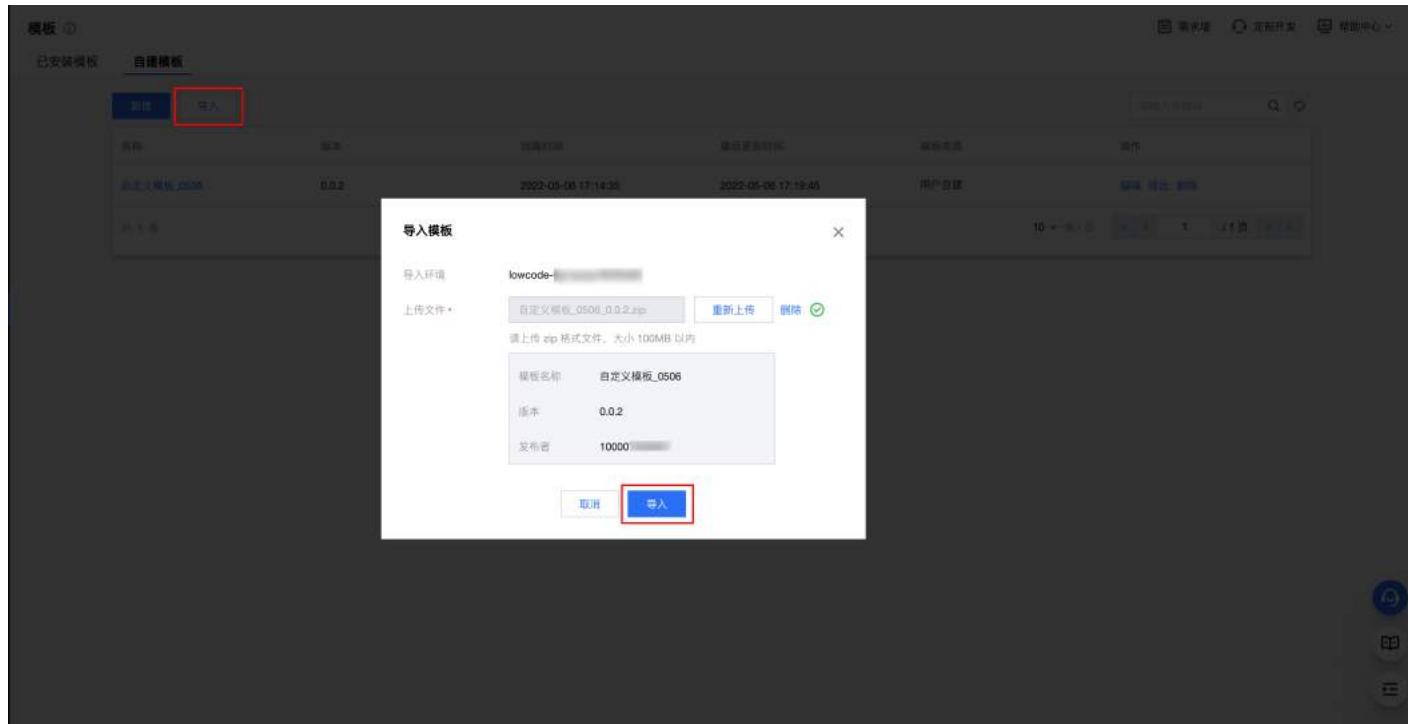
模板名称：自定义模板_0506
发布者：default
当前版本：0.0.1
导出版本 ① • 0.0.2

[导出](#) [取消](#)

导入应用

1. 打开腾讯云微搭低代码控制台，进入 [模板](#) > [自建模板](#) 页面，单击导入。

2. 在弹窗中上传模板文件压缩包，上传完成后，单击导入，即可将模板及其包含的应用一起导入微搭平台中。



应用自定义域名

最近更新时间：2023-06-29 16:50:21

腾讯云微搭低代码为 Web 应用提供了自定义域名和应用路径的能力，统称为访问管理。

默认域名

腾讯云微搭低代码提供了可供开发、测试使用的默认域名，在开发期间，您可以通过默认域名访问网站应用。一个微搭账号/环境对应一个默认域名，若未进行自定义域名和应用地址设置，则应用地址默认由默认域名、应用 ID 和页面 ID 组成。

默认域名有一定的访问频率限制，超过访问频率限制，默认域名将不可访问。

② 说明：

对于生产环境的应用，我们推荐您使用自定义域名来避免限频，以加速用户的访问。

自定义域名操作指引

您可以在微搭绑定已备案的自定义域名，操作指引如下。

⚠ 注意：

我们推荐使用腾讯云进行域名备案，若非腾讯云的备案域名，需要在腾讯云上完成接入备案后，才能在微搭绑定和使用自定义域名，更多详情可参见 [腾讯云网站备案](#) 相关帮助文档。

添加自定义域名

1. 登录 [腾讯云微搭控制台](#)，进入 [自定义域名](#) 页面。

自定义域名

 微搭交流群  官方公众号  业务咨询



2. 单击立即新建，进行域名添加，如下图所示：

The screenshot shows a 'Add Custom Domain' dialog box. At the top left is a note section with four items: 1. In binding custom domains, you need to handle website filing, refer to domain filing. 2. Binding HTTPS custom domains, default domain HTTPS cannot be visited. 3. Binding custom domains needs to be resolved to CNAME domain (custom domain.cdn.dnsv1.com), refer to CNAME resolution configuration. 4. Custom domain configuration is effective for all HTTP access services within the current environment. Below the note is a 'Default Domain' field containing 'lov [REDACTED].com'. Under 'Domain *' is a text input field with placeholder '请输入正确的域名'. Under 'Certificate *' is a dropdown menu with placeholder '请选择SSL证书'. Below these fields is a note: 'If no suitable certificate, upload it to the SSL certificate control panel, click here'. At the bottom are two buttons: 'Add' (highlighted in blue) and 'Cancel'.

3. 添加后，系统会为您自动分配一个以 .cdn.dnsv1.com 为后缀的 CNAME 域名，CNAME 域名不能直接访问，您需要在域名服务提供商处完成 CNAME 配置，配置生效后，您的托管服务方可对自定义域名生效，如何配置 CNAME 请参见 [下文](#)。

CNAME 配置

腾讯云设置方法

若您的 DNS 服务商为腾讯云，您可通过如下步骤添加 CNAME 记录。

1. 登录 [域名服务](#) 控制台，在列表中，找到需要添加 CNAME 记录的域名所在行，单击操作栏的解析。

<input type="checkbox"/>	域名	服务状态	注册时间	到期时间	操作
<input type="checkbox"/>	[REDACTED]	实名认证失败	2017-11-30	2019-11-30 已开启自动续费	解析 管理 更多
<input type="checkbox"/>	[REDACTED]	正常	2017-08-23	2022-08-23 已开启自动续费	解析 管理 更多

2. 在跳转到的页面中，单击添加记录。

The screenshot shows the 'Add Record' page. At the top is a toolbar with buttons: 'Add Record' (highlighted with a red box), 'Quick Add Website/Email Resolution', 'Pause', 'Resume', 'Delete', and 'Assign to Project'. Below the toolbar is a table with columns: 'Host Record', 'Record Type', 'Path Type', 'Record Value', and 'MX Priority'. A note at the bottom right says 'Current list is empty'.

3. 在 **主机记录** 处填写域名前缀（例如：www），将 **记录类型** 设置为 CNAME，在 **记录值** 处填写 CNAME 域名，单击保存，即可添加 CNAME 记录。

主机记录	记录类型	线路类型	记录值	MX优先级	TTL (秒)	最后操作时间	操作
<input type="text" value="按如下提示选填"/>	<input type="text" value="A"/>	<input type="text" value="默认"/>	<input type="text" value="按如下提示选填"/>	-	<input type="text" value="600"/>	-	<button>保存</button> <button>取消</button>
提示 要解析 www.qclouddocument.xyz，请填写 www。主机记录就是域名前缀，常见用法有： www 解析后的域名为 www.qclouddocument.xyz @ 直接解析主域名 qclouddocument.xyz * 泛解析，匹配其他所有域名 *.qclouddocument.xyz mail 将域名解析为 mail.qclouddocument.xyz，通常用于解析邮箱服务器 二级域名 如：abc.qclouddocument.xyz，填写abc 手机网站 如：m.qclouddocument.xyz，填写m							

② 说明：

域名解析各种记录类型之间是有优先级差异的，在主机记录相同的情况下，同一条线路有几种不同的记录类型不能共存，否则将会提示冲突。
CNAME 记录与除 CNAME 记录以外的任何记录类型都冲突，需要先删除掉其他记录，再进行配置。详情请参见 [常见问题](#)。

DNSPod 设置方法

若您的 DNS 服务商为 DNSPod，登录 [DNSPod 域名服务控制台](#)，在列表中，找到需要添加 CNAME 记录的域名所在行，单击对应域名名称，跳转至“添加记录”界面，通过如下步骤添加 CNAME 记录。



1: 主机记录处填子域名（例如需要添加 www.123.com 的解析，只需要在主机记录处填写 www 即可。如果只是想添加 123.com 的解析，主机记录直接留空，系统会自动填一个“@”到输入框内，@的 CNAME 会影响到 MX 记录的正常解析，添加时慎重考虑）。

2: 记录类型为 CNAME。

3: 线路类型（默认为必填项，否则会导致部分用户无法解析。在上图中，默认的作用为：除了联通用户之外的所有用户，都会指向 1.com）。

4: 记录值为 CNAME 指向的域名，只可以填写域名，记录生成后会自动在域名后面补一个“.”，这是正常现象。

5: MX 优先级不需要填写。

6: TTL 不需要填写，添加时系统会自动生成，默认为 600 秒（TTL 为缓存时间，数值越小，修改记录各地生效时间越快）。

万网设置方法

若您的 DNS 服务商为万网，您可通过如下步骤添加 CNAME 记录。

1. 登录万网会员中心。
2. 单击会员中心左侧导航栏中的产品管理 > 我的云解析进入万维网云解析列表页。
3. 单击要解析的域名，进入解析记录页。
4. 进入解析记录页后，单击新增解析按钮，开始设置解析记录。



5. 若要设置 CNAME 解析记录，将记录类型选择为 CNAME。主机记录即域名前缀，可任意填写（如：www）。记录值填写为当前域名指向的另一个域名。解析线路，TTL 默认即可。

hichina.com

The screenshot shows the 'New Record' section of the DNS management interface. A red box highlights the 'Record Type' dropdown set to 'CNAME'. Another red box highlights the 'Value' field containing 'hichina.com'. The TTL is set to '10分钟' (10 minutes). Below the form are buttons for 'Save' (保存) and 'Cancel' (取消).

6. 填写完成后，单击保存，完成解析设置。

新网设置方法

若您的 DNS 服务商为新网，您可通过如下步骤添加 CNAME 记录。

设置别名（CNAME 记录）

即：别名记录。这种记录允许您将多个名字映射到同一台计算机。通常用于同时提供 WWW 和 MAIL 服务的计算机。例如，有一台计算机名为 host.mydomain.com（A 记录）。它同时提供 WWW 和 MAIL 服务，为了便于用户访问服务。可以为该计算机设置两个别名（CNAME）：WWW 和 MAIL。如下图：



验证 CNAME 是否生效

不同的 DNS 服务商，CNAME 生效的时间略有不同，一般会在半个小时之内生效。您也可以通过 dig 的方式来查询 CNAME 是否生效，如果 dig 到后缀为 .cdn.dnsv1.com 的域名，表示域名 CNAME 已生效。

应用地址设置指引

对于 Web 应用，访问地址由“域名”和“路径”两部分组成。您可以在应用详情中查看应用的正式访问地址并进行设置。

② 说明：

目前支持自定义应用的地址设置，数据模型应用的地址设置将在后续支持。

1. 选择含 Web 端的应用，进入应用详情，可查看应用访问地址。访问地址需发布应用后才生效。

应用设置

三 应用详情

实验室

开发设置

基础设置

应用名称：
应用图标：
应用描述：
应用ID：
环境ID：
小程序ID：

删除应用

访问地址设置

访问地址：
https://lowcod
发布结果：
查看二维码

2. 单击地址后的设置，打开地址设置弹窗。您可以选择已绑定的自定义域名，并填写路径。相关设置规则和校验见下图。



3. 设置完成后，需进入应用编辑器进行应用发布，新的访问地址才会生效。新地址生效的同时，旧地址会立即失效。

相关说明

强制 HTTPS

所有对于自定义域名的 HTTP 请求，默认开启强制 HTTPS，会被转换为 HTTPS 请求。

② 说明：

强制 HTTPS 可以提高您的网站安全性，如有特殊需求，可联系我们调整。

数量上限

一个微搭环境中的自定义域名上限为5个，路径上限为200个。

删除环境时域名解绑

在删除环境时，自定义域名会与微搭账号/环境进行绑定，并删除相关路径配置信息。自定义域名解绑后，您可以在其他微搭账号/环境绑定该域名。

编辑器

编辑器介绍

最近更新时间：2023-06-27 16:20:15

本文将为您介绍微搭低代码应用开发的IDE，即编辑器，您可通过以下图片和表格快速了解应用编辑器界面工作台的基本信息、各个功能的作用及使用方式。

编辑器界面概览

进入编辑器的默认界面模块



各模块功能介绍

位置	功能
页面管理	新建和管理页面，以及页面设置等操作。可在此管理应用下的所有页面，对页面进行新增、删除、克隆和属性设置等。
顶部导航栏	包含了多端切换、应用标题修改、预览、发布等基本功能。
大纲树和状态数据	展示完整的组件大纲结构。对于较难在编辑区选中组件时，大纲树可轻松区分组件的层级，同时组件树会展示组件的特殊状态，例如：绑定数据、条件展示 IF。切换到状态数据后，可查看当前应用所有自定义变量、query 以及组件 API 的运行时数据结构或运行值。
组件列表区	展示平台官方组件及自定义组件，以及区块列表。
编辑区	对页面中的组件进行编辑、实时效果预览。
变量/代码区	展示和管理当前应用中所有自定义的变量、自定义 JS 方法以及 query 数据查询等，以便于统一管理和操作。

事件面板	为组件的事件交互行为绑定相应的执行动作或平台方法。
底部开发工具区	包含前端开发常用的开发调试工具（Console）、代码编辑器以及应用历史记录。 进入开发调试工具，可以通过调试区进行信息调试和打印。 进入代码编辑器界面，开发者能在此根据自己的业务逻辑，更加灵活地完成定制化的业务需求。

顶部导航栏详细介绍

顶部导航栏为您提供使用应用编辑器的基本功能。

功能	功能说明
左上角微搭 Logo	单击可返回控制台，或切换当前环境下的其他应用进行开发。
端切换	可在 H5、小程序和 PC 中三端切换。（支持切换的端依赖于创建应用时的配置）
预览	切换到应用的预览态，可生成实时预览快照、二维码或 URL 链接。
发布	开发完成的应用可发布体验版或正式版，体验版和正式版应用的内容相互隔离，数据管理内容和访问地址都不同。可生成二维码或链接。
保存	将编辑器当前内容进行保存，可保存多个版本并重新恢复历史。
更多	<ul style="list-style-type: none"> 历史版本：通过线上历史列表可以选择自己想要恢复的版本到本地。 修复大纲树插槽：当组件插槽出现 BUG 或更新的情况时，可进行一键修复。 清除组件：一键删掉页面上的所有组件。 调起微信 IDE 预览：在编辑小程序应用时，能调出“微信开发者工具”来预览小程序。

编辑器进行组件配置时的界面模块

当您进入编辑器，并且拖入第一个组件或选中任意组件时，会激活组件配置面板，如上图右侧区域所示。

组件区功能	说明
属性	对组件、页面或应用的数据属性进行静态设置，或与变量进行动态绑定。
样式	对组件或页面进行可视化的样式设置。
事件	对组件进行事件的绑定及监听，可实现单击弹窗、单击跳转等功能。

页面管理

页面及应用生命周期

最近更新时间：2023-08-11 16:53:37

微搭支持页面、应用级别的生命周期函数，当对应的生命周期函数触发时，便会自动执行函数下已配置的事件动作或自定义方法。

页面生命周期

页面生命周期说明：

函数名	说明
onPageLoad	页面加载时触发。一个页面只会调用一次，可以在 onLoad 的参数中获取打开当前页面路径中的参数。
onPageShow	页面显示/切入前台时触发。
onPageReady	页面初次渲染完成时触发。一个页面只会调用一次，代表页面已经准备妥当，可以和视图层进行交互。
onPageHide	页面隐藏/切入后台时触发。例如 app.navigateTo 或底部 tab 切换到其他页面，小程序切入后台等。
onPageUnload	页面卸载时触发。例如 app.redirectTo 或 app.navigateBack 到其他页面时。

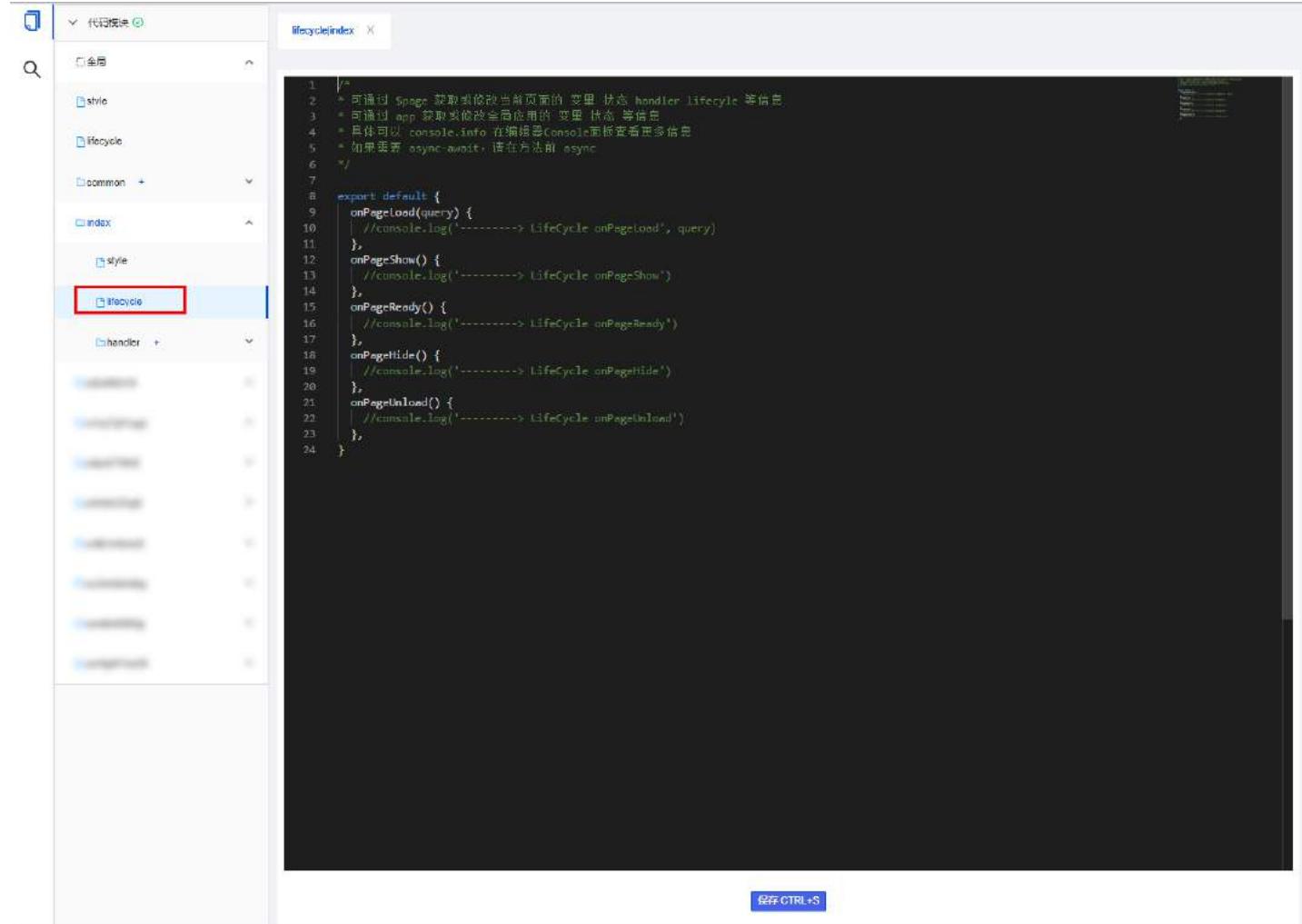
页面生命周期支持可视化配置来调用事件动作或自定义方法，也可以支持在代码编辑器的页面生命周期函数中来执行相应的自定义代码化，两种配置方式分别如下：

页面生命周期的可视化配置

选中左侧大纲树的页面根节点，即可在右侧属性面板下方的事件面板中看到对应配置入口，如下所示：

页面生命周期的代码化配置

可在编辑器右下角打开 [代码编辑器](#)，选择对应页面文件夹下的 lifecycle 文件，即可对该页面的生命周期进行管理。



The screenshot shows the Tencent Cloud Low-Code Platform's code editor interface. On the left, there is a tree view of files and folders under the '全局' (Global) category. A red box highlights the 'lifecycle' folder. The main editor area displays the 'lifecycleindex.js' file with the following code:

```
1 // 可通过 $page 轻取或修改当前页面的 变量 状态 handler lifecycle 通知
2 // 可通过 $app 轻取或修改全局应用的 变量 状态 等信息
3 // 具体可以 console.info 在编辑器Console面板查看更多信息
4 // 如果要署 async await, 请在方法前加 sync
5 //
6 //
7 export default {
8   onPageLoad(query) {
9     //console.log('-----> lifeCycle onPageLoad', query)
10 },
11   onPageShow() {
12   //console.log('-----> lifeCycle onPageShow')
13 },
14   onPageReady() {
15   //console.log('-----> lifeCycle onPageReady')
16 },
17   onPageHide() {
18   //console.log('-----> lifeCycle onPageHide')
19 },
20   onPageUnload() {
21   //console.log('-----> lifeCycle onPageUnload')
22 },
23 }
24 }
```

At the bottom right of the editor, there is a blue '保存 CTRL+S' button.

应用生命周期

打开 [代码编辑器](#)，选择全局文件夹下的 lifecycle 文件，即可对应用的生命周期进行管理和代码编写。

The screenshot shows the Tencent Cloud Low-Code Platform's code editor interface. On the left, there's a sidebar with a search bar and a tree view showing a folder structure with 'style' and 'lifecycle' selected. The main area displays the 'lifecycleglobal.js' file content:

```
1  /**
2   * 可通过 app 调取或修改全局应用的某些状态 等信息
3   * 具体可以 console.info 在端页器Console面板查看更多信息
4   * 如果需要 esnext-await 请在方法前 async
5   */
6  export default {
7    onAppLaunch(launchOpts) {
8      //console.log('-----> LifeCycle onAppLaunch', launchOpts)
9    },
10   onAppShow(appShowOpts) {
11     //console.log('-----> LifeCycle onAppShow', appShowOpts)
12   },
13   onAppHide() {
14     //console.log('-----> LifeCycle onAppHide')
15   },
16   onAppError(options) {
17     //console.log('-----> LifeCycle onAppError', options)
18   },
19   onAppPageNotFound(options) {
20     //console.log('-----> LifeCycle onAppPageNotFound', options)
21   },
22   onAppUnhandledRejection(options) {
23     //console.log('-----> LifeCycle onAppUnhandledRejection', options)
24   }
25 }
```

At the bottom right of the editor, there is a blue '保存 CTRL+S' button.

应用生命周期说明：

函数名	说明
onAppLaunch	当应用初始化完成时触发（全局只触发一次）。
onAppShow	监听小程序切前台事件。该事件与 App.onShow 的回调参数一致。
onAppHide	监听小程序切后台事件。该事件与 App.onHide 的回调时机一致。
onAppError	小程序发生脚本错误或 API 调用报错时触发。
onAppPageNotFound	小程序要打开的页面不存在时触发。
onAppUnhandledRejection	小程序有未处理的 Promise 拒绝时触发。

更多生命周期相关介绍

- 了解更多微搭运行时的生命周期相关说明，请参见 [应用/页面/组件生命周期](#)。
- 了解如何在生命周期中调用自定义 JS 方法，请参见 [代码编辑器](#)。

页面跳转传参配置

最近更新时间：2023-06-27 16:20:25

使用场景

页面 URL 参数通常用于，两个页面间跳转数据传递。

基础功能

- 支持定义多个 URL 参数，用于接收数据。
- 若开启必填，则跳转到页面时，必须携带该参数。



应用示例

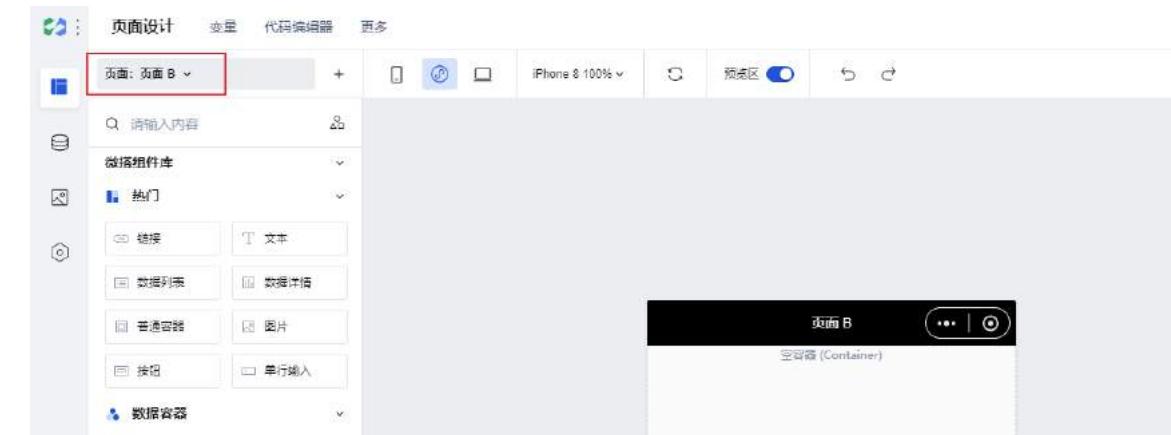
页面 A 单击图标，跳转到页面 B，并且带数据 ID。

- 在编辑器中创建两个页面，分别命名为页面 A 与页面 B。



- 目标页面 B，定义从其他页面跳转需要传入的 URL 参数，用于接收数据。

- 切换到目标页面 B。



b. 在大纲树中选择页面根节点，在右侧属性面板中新建页面 URL 参数。

The screenshot shows the Tencent Cloud Micro搭 Low-Code Platform interface. On the left, there is a sidebar with a tree view labeled '大纲树' (Outline Tree) containing '页面: 页面B'. Below it is a '代码区' (Code Area) with sections for '全局' (Global) and 'query2'. In the center, there is a preview window showing a mobile screen with the text '页面B'. On the right, there is a '组件' (Components) panel with tabs for '属性' (Properties) and '配置' (Configuration). Under '属性' (Properties), there is a section for '页面URL参数' (Page URL Parameters) which has a red box around it. A button labeled '新建页面URL参数' (Create New Page URL Parameter) is also highlighted with a red box. Other sections in the configuration panel include '屏幕配置' (Screen Configuration), '页面分享' (Page Share), and '小程序转发' (Mini Program Forwarding).

新建 URL 参数名称，填入名称，是否开启必填开关后确认即可。

新建页面URL参数

* 参数名称
id

必填
 是
 否

若参数必填，则从其他页面跳转至本页面时必须提供该参数

c. 完成 URL 参数新建后，给页面 B 拖入一个文本组件，并绑定对应 URL 参数。



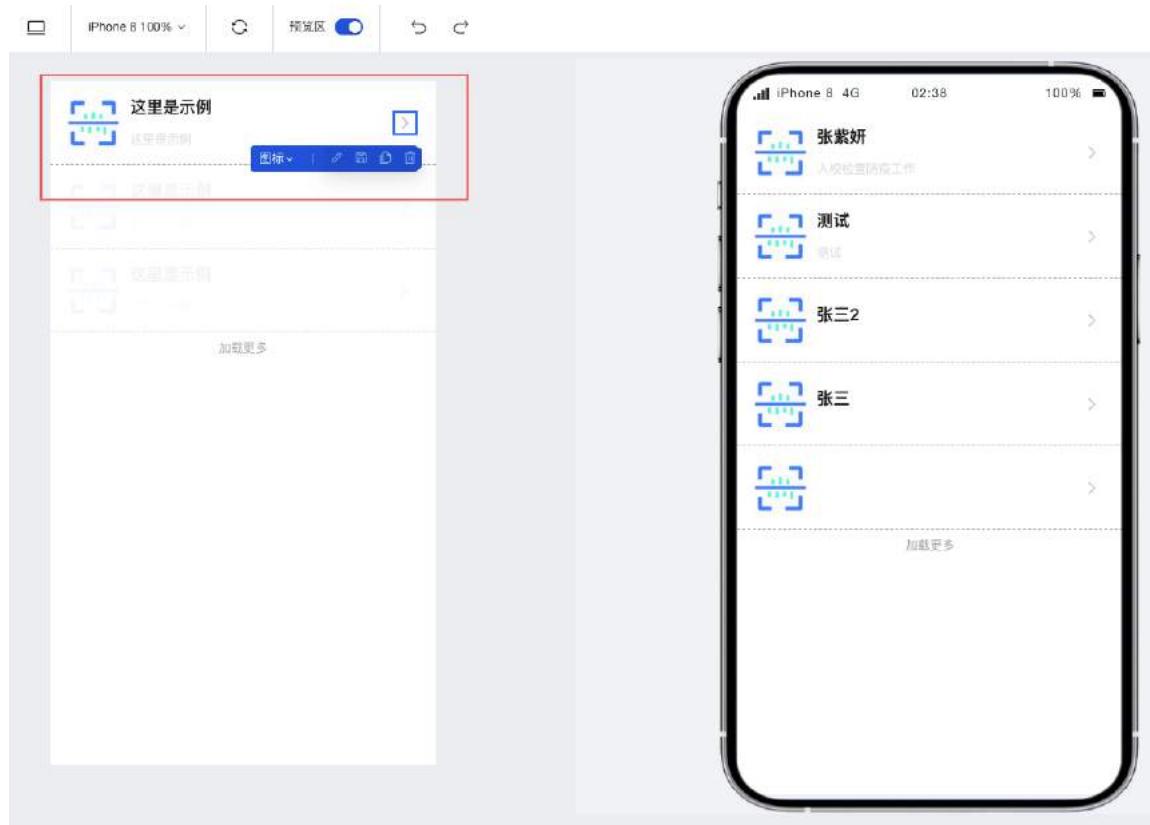
此时，目标页面 B 需要的参数就定义好了。下一步需要从其他页面跳转时，设置传入参数。

3. 页面 A 设置跳转动作，并设置所需的传递参数。

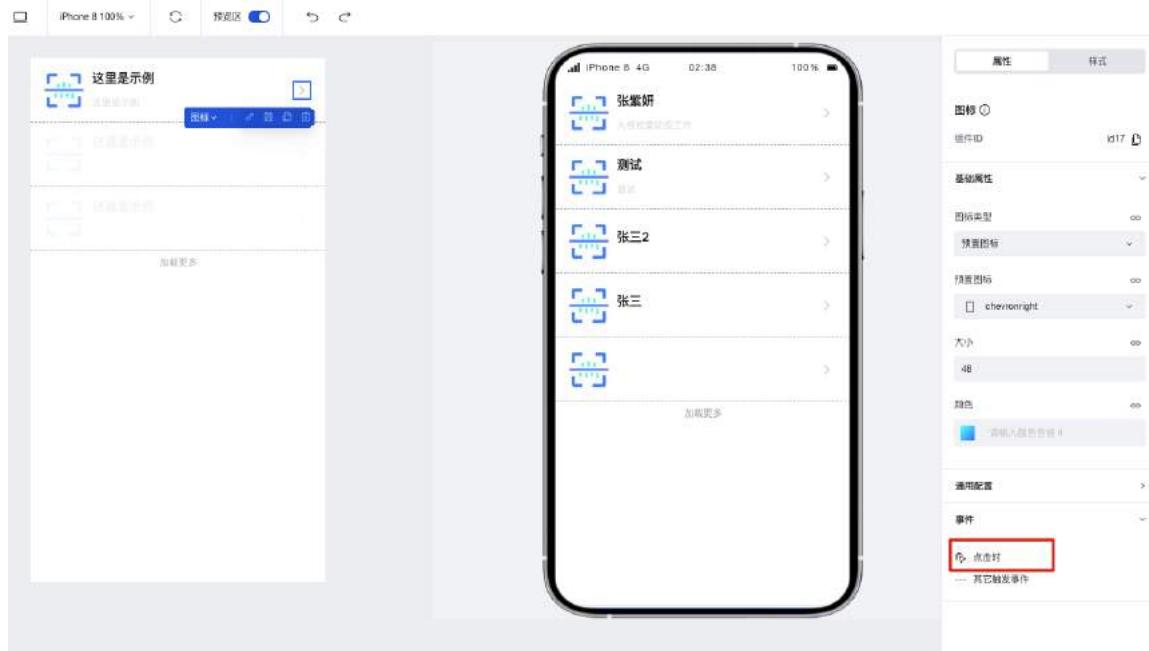
i. 切换到另一个页面 A。



ii. 选中要跳转页面的组件。



iii. 打开事件设置跳转。



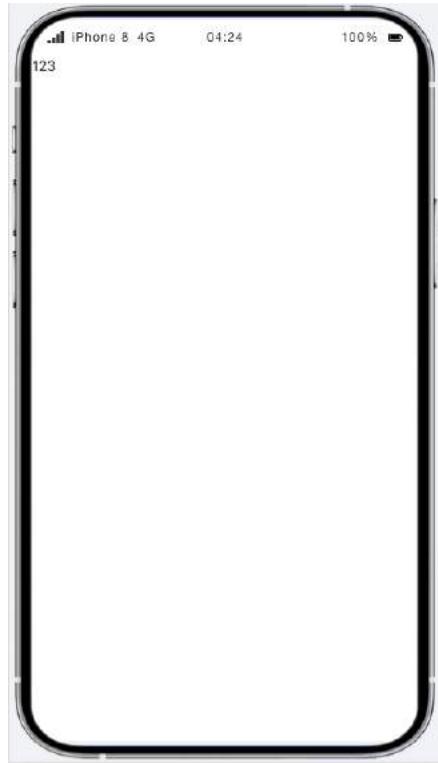
iv. 添加事件打开页面，选择要跳转的页面，输入需要传递的参数，单击保存。



参数传入可以用静态数据，或使用表达式计算。

4. 开启实时预览，体验查看跳转结果。

开启实时预览，在页面 A 中单击按钮跳转至页面 B，可以看到文本组件的内容会根据传参变更为123。



表达式

最近更新时间：2023-07-21 15:51:05

微搭的表达式采用的 JavaScript 语法，您可以在表达式中直接执行原生的 JavaScript 代码，也可以直接在表达式中引用微搭内置的变量、函数以及组件属性 API 等。表达式是微搭可视化组件连接动态数据的重要桥梁，也是低代码平台区分无代码灵活度的体现。

当进入微搭编辑器时，您可以在组件面板的属性配置以及事件面板的属性配置处，切换到 fx 表达式输入框或面板中自定义的 JavaScript 表达式来读取、转换和操作页面上所有页面变量和数据，以及全局变量。

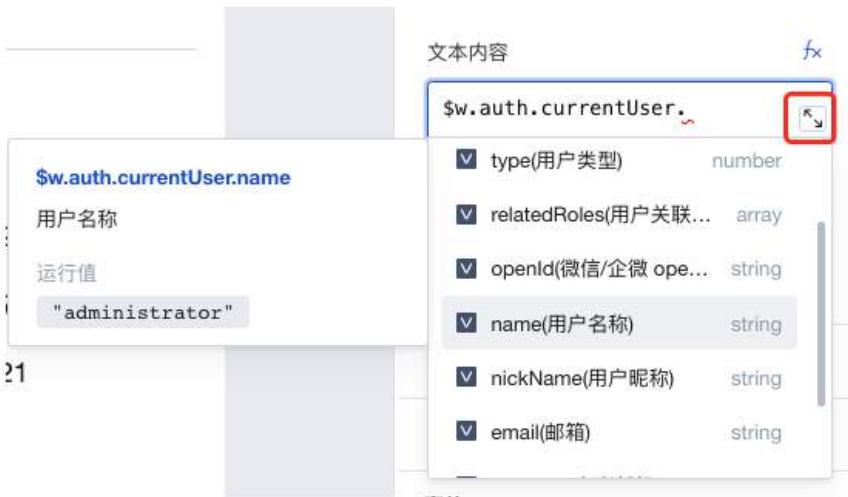
表达式的使用

可通过组件配置面板，单击任意输入框右上角的 fx 图标来激活表达式模式（默认为静态文本），如下图所示：



访问数据

当表达式的输入框聚焦时，会列出变量以及组件属性等内置表达式的自动提示，Hover 时可展示对应表达式运行态的值，选中任意变量后效果，如下图所示：



其中单击上图中的红框放大 icon，可以打开放表达式面板，放大模式下，可便于输入更长文本的表达式代码。放大版表达式界面如下所示：



访问对象数据

在微搭编辑器中，Query、组件 API 以及部分系统变量等都是 JS 对象。针对这些 JS 对象，它们都有相应的内置方法或属性。当输入表达式时，在对象名后添加一个 .，就会弹出一个下拉的自动提示列表。您可以在提示列表中单击选择相应的属性或方法，也可以使用上下键进行选择。

输入框就地表达式

访问数组数据

如果要访问存储在数组中的值，可通过数组的下标索引值 index 来访问。数组下标索引从 0 开始，例如访问数组变量 list 的第一个元素，则可使用 \$w.page.dataset.state.list[0]（输入 list 后根据自动提示选择对应变量即可），如下所示：



操作数据

在表达式中支持使用 JavaScript 语法以及调用第三方 JS 库，同时也支持对组件 API、Query 对外暴露的数据进行转换操作。

例如：

- 对日期的格式进行转换：\$w.DateText(\$w.formDate1.value, 'YYYY-MM-DD HH:mm:ss')。
- 对某个数字类型变量值执行加9527：\$page.dataset.state.count + 9527。
- 对一个用户列表的对象数组进行遍历并返回用户名：queryUsers.data.map((user) => user.name)。



⚠ 注意：

表达式中的 JS 需满足一定的语法规规范，支持类型包括：内置系统方法、自定义 JS 方法以及标准 JS 方法如：map()/reduce()/filter() 结合箭头函数、三目运算符等单行代码。如需编写多行代码，建议使用 [自定义 JS 方法](#) 实现业务逻辑后，在表达式中调用即可，例如自定义 JS 方法 function1，则可通过 \$page.handler.function1({...}) 在表达式中进行调用。

查看数据/运行值

对于表达式在运行态的值，可以通过以下几种方式进行查看。

编辑器左侧状态数面板：

表达式可引用的资源类型

在表达式中，除了可以写常规的 JavaScript 表达式代码和自定义变量，也可以引用微搭内置的系统变量、系统函数公式，以及开放的组件属性（WidgetAPI），以上几种引用资源类型的具体介绍参考如下：

内置系统变量

微搭内置的系统变量详细介绍请参见 [微搭内置系统变量说明文档](#)。

组件属性（WidgetAPI）

详细的组件属性 API 介绍请参见 [组件属性 API 文档](#)。

内置函数公式

微搭内置的函数公式详细介绍请参见 [微搭内置函数公式说明文档](#)。

其他平台方法

微搭内置的其他平台方法，例如调用数据源、调用云端能力等，请参见 [微搭内置平台方法文档](#)。

表达式应用示例

- 从输入框的输入值动态显示在文本组件中。



- 从数据表或 API 中获取数据绑定到数据列表中。

考虑以下场景，您有一个通过 webhook 主体接收数据的 webhook 触发器。您想要提取其中一些数据以用于数据列表展示。

- 写一段 JavaScript 代码，例如 filter/transformer。

例如有这样一个场景：从数据源中拿到一个用户列表的数据（例如 queryUsers.data），它的数据结构是：

```
[  
 {  
   "id": 1,  
   "name": "rocky luo"  
 },  
 {  
   "id": 2,  
   "name": "judy li"  
 },  
 {  
   "id": 3,  
   "name": "elon ma"  
 }  
 ]
```

如果您希望将用户列表中的 name 字段拆分成姓氏 surname 和名字 givename，则可通过如下方式在表达式中完成数据结构的转换，将下述内容复制到表达式面板中即可。

```
queryUsers.data.map((row) => {  
  return {  
    ...row,  
    givename: row.name.split(' ')[0],  
    surname: row.name.split(' ')[1],  
  };  
})
```

行为事件

事件介绍

最近更新时间：2023-06-27 16:19:36

什么是事件

事件（Event）是页面的交互组成，例如按钮单击打开新页面，表单按钮单击提交数据。

适用场景

适用于所有页面交互的场景，例如提示信息、打开页面、查询数据等。

基础能力

事件由下述触发条件、[执行动作](#) 两部分构成。例如按钮单击打开页面，触发条件是按钮单击，执行动作是打开新页面。

② 说明：

如果有多个事件方法连续调用，则不同事件方法之间通过 `event` 对象 进行传参，例如调用数据源查询方法后，将返回的数据通过弹窗显示出来。若多个事件方法的调用需要在页面不同组件中进行复用，也可使用 [页面事件流](#) 进行定义和调用。

触发条件

事件的执行需要由某个条件来触发，不同的组件存在不同类型的触发条件，例如按钮的点击 `click`、输入框的聚焦 `focus` 等。

由组件行为触发

通用触发条件

触发条件	功能
<code>onClick</code> (单击时)	鼠标单击时触发

表单类组件的触发条件

触发条件	功能
<code>change</code> (输入改变)	输入值发生改变时触发
<code>focus</code> (聚焦)	组件获取焦点时触发
<code>Blur</code> (失焦)	组件失去焦点时触发
<code>confirm</code> (确认)	组件回车确认时触发
<code>clear</code> (清除内容)	组件清空输入时触发

其他组件触发条件

每个组件支持的事件，您可以在单击编辑器组件属性面板下方 > 事件查看，或请直接参见 [组件列表文档](#) 对应每个组件介绍的事件模块进行了解。

由页面生命周期触发

除组件触发外，页面打开和关闭也可作为触发条件。例如加载时，查询数据并储存，以下是页面打开和关闭时支持的触发条件。

事件名	功能
<code>页面加载时 (loading)</code>	页面加载时触发动作
<code>页面显示时 (show)</code>	页面显示时触发动作
<code>页面初次渲染完成 (ready)</code>	页面渲染完成触发动作

事件名	功能
页面隐藏 (hide)	页面隐藏时触发动作
页面卸载 (unload)	页面卸载前触发动作

使用路径为：在大纲树中选中页面，打开事件面板，选择对应触发条件和执行动作方法。

The screenshot shows the Tencent Cloud Low-Code Platform's event configuration interface. On the left, there is a sidebar with a tree view of components and a search bar. The main area displays a preview of a page with user profiles and sections for '权限管理' (Permission Management) and '数据连接' (Data Connection). Below the preview, a modal window titled '添加事件: 页面加载 (load)' is open, prompting the user to choose actions for the event. The '应用动作' (Action Application) section contains several options like '调用数据查询 (callQuery)', '调用数据源方法 (callDataSource)', and '触发流程 (callProcess)'. The '事件' (Event) section on the right lists three options: '页面加载 (load)', '页面显示 (show)', and '其他事件 (Other Events)'. A red box highlights the search bar in the sidebar and the '事件' section in the modal.

执行动作

触发事件后，可配置页面、数据查询、组件内置方法和自定义方法等平台内置的动作方法。更多执行动作相关介绍，请参见 [执行动作说明](#)。

执行动作

最近更新时间：2023-08-09 11:00:06

本文主要介绍微搭编辑器内置的平台方法及使用方式。

② 说明：

当前微搭内置的平台包括但不限于以下常用方法，除了通过事件面板的可视化方式调用，下述方法也可以在自定义 JavaScript 方法中进行调用，可视化更多详细的平台方法使用说明和参数介绍请参见 [微搭内置方法 API](#)。

调用数据源方法 callDataSource

调用数据源中已配置完成的方法，对数据源进行增删改查的操作。

方法参数

参数名称	参数说明
数据源名称	从当前环境中所有已创建的数据源进行选择
方法名	对数据源方法进行选择
传入参数	若数据源方法需要入参，则在此处可以传入对应参数（非必填）
显示加载	在数据源请求过程中显示加载提示
显示提示	数据源请求完成后显示成功或失败的消息提示

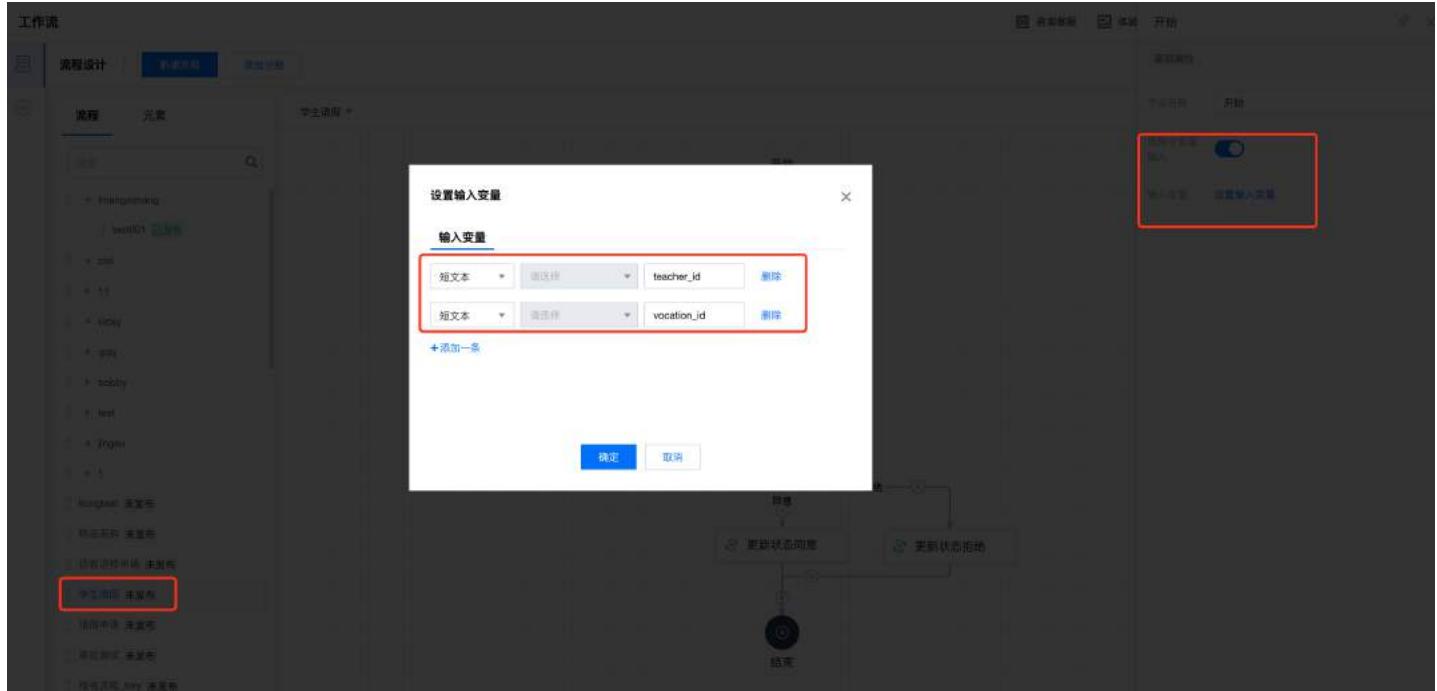
触发流程 callProcess

触发当前环境下工作流中创建的流程

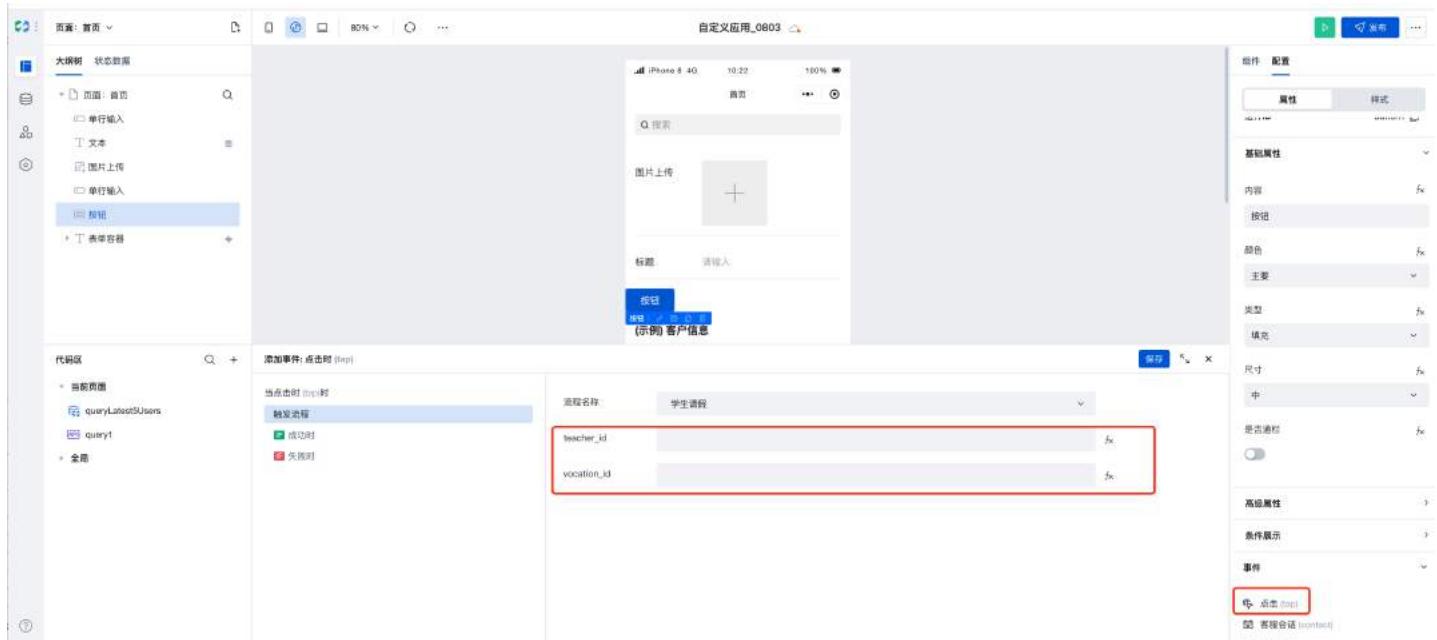
方法参数

参数名称	参数说明
流程名称	从当前环境中所有已创建的流程进行选择

如选中流程开始节点配置了输入变量。



需在触发流程项中同步配置触发流程入参。



变量赋值 setState

为当前页面或全局变量进行赋值操作。

方法参数

参数名称	参数说明
变量名	支持下拉选择当前页面或者是全局中已创建的全部变量
变量值	输入需要赋予所选变量的值

打开页面 navigateTo

控制应用的页面跳转，页面跳转后支持使用 `navigateBack` 方法进行页面返回操作。

方法参数

参数名称	参数说明
模式	微搭：跳转至当前应用编辑器的其他页面 Web：跳转至外部页面（当构建平台为小程序时不可用）
链接	模式选择微搭时：选择应用编辑器的内部页面 模式选择 Web 时：输入外部页面链接（当构建平台为小程序时不可用）

扫码 scanCode

触发后调出手机扫码功能，支持对条形码、二维码进行识别。

方法参数

参数名称	参数说明
是否只能从手机扫码，不允许从相册选择图片	扫码时是否支持选择手机相册图片
是否开启默认管理	在 Web 环境下若扫码内容为链接则自动弹窗打开，否则展示扫码内容。在小程序环境下自动弹窗显示内容
扫码类型	选择可识别的扫码类型，目前支持条形码、二维码

使用示例

如何把扫码后获取的值填写到表单中？

1. 新建一个普通变量（`codeResult`），页面中添加一个按钮和一个单行输入组件，单行输入组件的输入值绑定新建的普通变量。

The screenshot shows the Tencent Cloud Micro-Assembly low-code editor interface. On the left, the '大纲树' (Outline Tree) shows a '单行输入' (Single-line Input) component under the '测试文档用' (For Test Document) page. In the center, a mobile device preview shows a blue button labeled '点击扫码' (Scan) and a text input field labeled '扫码结果' (Scan Result) with placeholder text '请输入'. On the right, the '组件' (Component) panel is open, showing the configuration for the '单行输入' component. The '属性' (Properties) tab is selected, and the '输入值' (Input Value) field contains the expression '\$w.page.dataset.state.codeResult'. A red box highlights this input value field.

2. 选择按钮，为按钮配置事件 > 点击时进行扫码，扫码成功时给变量赋值（event.detail.result 为获取二维码对应的链接值）。

The screenshot shows the Tencent Cloud WeDai low-code platform interface. On the left, there's a sidebar with a tree view of components: 大部件 - 状态数据, 布局 - 页面: 测试文档用, and 代码区. Under 代码区, there's a section for '当前页面' containing a variable 'codeResult'. The main workspace shows a mobile phone simulation screen with a blue button labeled '点击扫码' (Scan). Below the phone screen, a modal window titled '添加事件: 点击时 (tap)' is open, showing event triggers: '当点击时 (tap) 时' (Scan) and '成功时' (Success). In the '成功时' section, a variable assignment is configured: 变量名 'codeResult [文本]' and 变量值 'event.detail.result'. On the right side, there's a '组件 配置' panel with tabs for 属性 and 样式. The '属性' tab is active, showing settings for the component: 内容 '点击扫码', 颜色 '主要', 类型 '填充', 尺寸 '中', 是否透明 '否', and 事件. The '事件' section is expanded, showing the selected event '当点击时 (tap) 时' with the trigger '扫码'.

3. 应用发布为小程序后，即可用手机打开小程序来查看效果，单击按钮扫描任意二维码可查看结果。

其他平台方法说明

更多平台方法的使用说明，请参见 [微搭内置工具/交互方法说明](#)。

触发条件及 event 对象

最近更新时间：2023-07-04 09:57:12

一个组件的事件由触发条件以及执行动作两部分构成，执行动作又分为 平台方法 和自定义方法，本文章主要对触发条件及对应 event 对象进行介绍说明。

什么是 event 对象

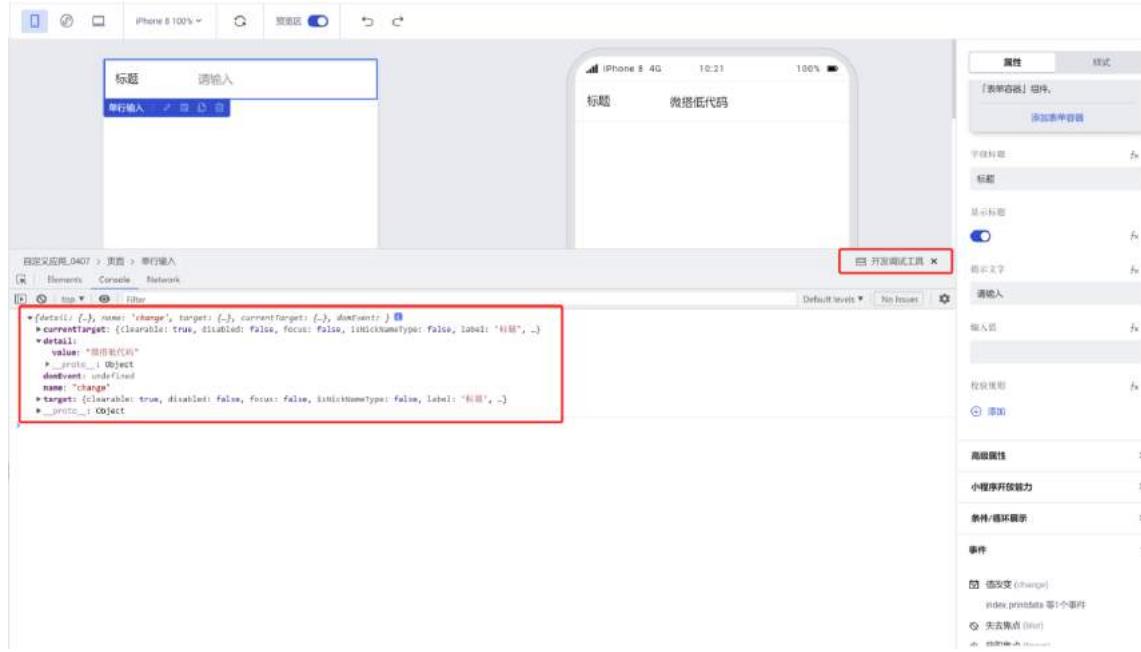
在微搭中，当一个组件的触发条件被触发时则会返回该触发条件的 event 对象，触发条件不同，返回的 event 对象中存储的值也不同，我们可以通过获取 event 对象中的值来实现一些常见功能。

我们可以通过自定义 JS 方法来打印 event，查看其结构，下文为打印 event 的示例：

1. 通过左下角的代码区添加一个自定义 Javascript 方法，代码内容如下，保存后命名为 eventLog，示例如下：

```
export default function({event, data}) {
  console.log(event);
}
```

2. 在编辑区拖入任意一个组件，如单行输入组件，然后在按钮绑定一个点击事件，事件动作选择 **Javascript 代码方法** 节点，然后下拉方法列表中选择刚刚新建的 eventLog 方法。
3. 在输入框中进行输入改变的操作，查看下方开发调试工具（面板 > 开发调试工具）控制台中的查看打印信息。



4. 根据上图所示，我们便可以在编辑器中通过使用 event.detail.value 来调用当前触发条件所返回的值。

不同组件的事件触发时返回的 event 对象

单行输入

事件	event.detail
change (输入改变)	{ value: string }, value 为输入的新值
focus (聚焦)	{ value: string }, value 为当前输入框的值
blur (失焦)	{ value: string }, value 为当前输入框的值
confirm (确认)	{ value: string }, value 为当前输入框的值
clear (清除内容)	{ originValue: string }, originValue 为清除前输入框的值

多选

事件	event.detail
change (选中状态改变)	{ value: string[] }, value 为当前选中状态的值的集合

单选

事件	event.detail
change (选中状态改变)	{ value: boolean }, value 为当前选中状态

下拉选择

事件	event.detail
change (值改变)	{ value: string }, value 为当前选中的值

更多组件

其余更多组件对应的触发条件以及返回的 event 对象结构, 请参见 [组件列表](#) 中的事件部分, 例如查看 [开关组件](#) 的事件介绍。

页面事件流

最近更新时间：2023-06-14 08:59:21

事件流（eventflow）是一系列前端行为动作（action）的集合，可按照开发者指定的顺序和逻辑执行。事件流主要应用于页面中同一类行为动作的抽象和复用（例如提交数据后跳转页面），以及自定义代码和数据查询回调等场景。

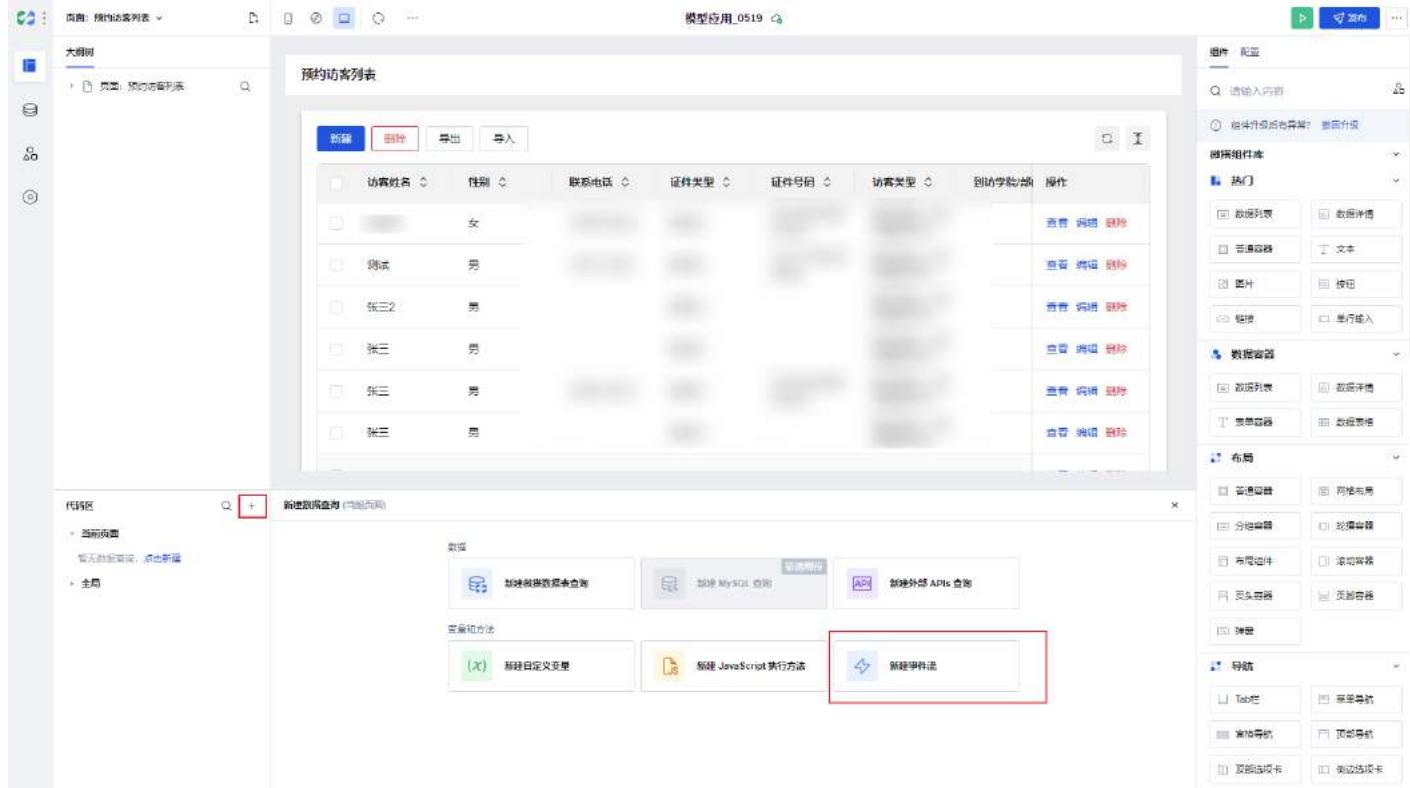
⚠ 注意：

目前事件流仅支持页面内作用域，暂不支持应用全局作用域。

事件流的使用

• 新建事件流

在编辑器左下角的代码区可通过单击 + 号新建。



• 配置事件流

i. 进入事件面板，选择所需要的事件动作节点。



The screenshot shows the 'eventflow1' configuration screen. On the left, there's a sidebar with '当前页面' expanded, showing 'eventflow1'. The main area has tabs for '应用动作', '页面交互', '微信能力', and '组件方法'. Under '应用动作', there are four groups of nodes: '调用数据查询 (callQuery)', '调用数据源方法 (API) (cloud...', '触发流程 (callProcess)', '设置状态 (setState)', '逻辑分支 (if)', '调用事件流 (eventflow)', and 'JavaScript (code)'. Other tabs show nodes for navigating pages, displaying toast messages, hiding loading indicators, opening modals, making phone calls, and scanning codes.

ii. 进入任意事件动作节点后，可以继续配置节点执行成功或失败时的下一个动作，其余节点以此类推，就可以完成一个完整事件流的配置和复用了。



This screenshot shows the configuration of an 'eventflow1' node. It's set to trigger on '调用数据查询' (callQuery). The configuration includes fields for '选择Query' (Select Query), '方法' (Method), and '出参' (Out Parameters). A tip at the bottom says: 'Tip: 鼠标Hover上方条件可新增执行动作' (Tip: Hover over the top condition to add execution actions).

事件流的调用/触发方式

- 通过事件面板的调用事件流节点触发，参考如下：

i. 为按钮添加点击事件，选择调用事件流。

The screenshot shows the platform's interface for creating a mobile application. On the left, there's a sidebar with sections like '大屏网' (Big Screen Network) and '代购区' (Purchase Area). In the center, there's a preview of a mobile screen with various cards and sections. On the right, there's a detailed configuration panel for a 'button' component. Under the '事件' (Events) tab, there's a section for '添加事件: 点击时 (click)' (Add Event: Clicked). A tooltip says '请选择下方的执行动作' (Please select the execution action below). Below it, several actions are listed, with '调用事件流 (callEventFlow)' highlighted with a red border. Other actions include '读取数据 (readData)', '读取数据 (读取) (readData)', '触点流程 (callProcess)', '发送短信 (sendSms)', '发送文件 (sendFile)', and '发送邮件 (sendEmail)'. The configuration panel also includes tabs for '属性' (Properties), '样式' (Style), and '逻辑' (Logic).

ii. 选择事件流并保存。

This screenshot shows the configuration of an event flow named 'eventflow1'. On the left, there's a '当时' (When) section with two options: '调用事件流' (Call Event Flow) and '成功时' (Success). Below that is a note: 'Tips: 鼠标Hover上方条件可新增执行动作' (Tip: You can add new execution actions by hovering over the condition above). On the right, there's a '选择eventFlow' (Select eventFlow) dropdown set to 'eventflow1', and an '入参' (Input Parameters) input field containing 'eventflow1'. There's also a '保存' (Save) button.

其中，如果要对事件流传参，则可以在入参输入框中切换到 **fx 表达式模式**，填入对应的 JS 对象和变量即可，例如：{foo: 1, bar: 2}

- 在 **数据查询 (query)** 的回调中触发。

The screenshot shows the 'Edit Data Query' interface for a query named 'query2'. Key settings include:

- 选择数据表 ***: ai用户表 (ai_user_2c647ar)
- 触发方式**: 入参变化时自动执行 (Selected)
- 方法**: 查询单条(wedaGetItem)
- 入参**: Data Identifier (Text | Short Text)
- 备注信息**: None
- 事件回调**: Success (Selected)
- 配置事件流**: eventflow1
- 成功时**: Select and Configure Event Flow

如果需要在数据查询 query 的回调中对事件流传参，可在入参输入框切换到 **fx 表达式模式**，填入对应的 JS 对象和变量即可，例如：{foo: 1, bar: 2}，或在 query 的回调中，可直接填入 query 执行的返回值，例如：\$w.query1.data。

- 在 **自定义 JS 代码** 中触发。

在 JavaScript 代码中触发方式：\$w.eventflow1.trigger({...})，其中 trigger() 括号中传入 JS 对象，例如：\$w.eventflow1.trigger({foo: 1, bar: 2})。

获取调用事件流时的入参

事件流的入参方式，默认通过表达式传入对象（JS Object）的方式进行传参，在事件流中可以通过对 event.detail 对象来读取入参对象的属性值，如下图所示：

示例：如 query_one 执行成功时，调用事件流 eventflow1，对应入参结构如下所示：

```
{  
  "createdAt": 1676xxxxxxxx30,  
  "appname": "程序员",  
  "intro": "精通各种编程语言的程序员，帮你一键生成SQL/算法等多种编程语言代码"  
  ...  
}
```

代码区 Q + 编辑数据查询 query_one     

当前页面 全局 事件流

query_one text6 query2 query666 eventflow1 eventflow2

选择数据表 * ai产品表 (aiapp_0j27pvv)   

触发方式 入参变化时自动执行 手动触发执行

方法 查询单条(wedaGetItem)

入参

数据标识 (文本 | 单行文本) 

查询条件 (文本 | 单行文本) 0个条件 

展开更多字段 

备注信息 备注信息

事件回调 成功时    

配置事件流

选择事件流 * eventflow1   

事件入参

`$w.query_one.data`  

```

"appname": "程序员",
"updateBy": "1502",
"intro": "精通各种编程语言的程序员"

```

在事件流 eventflow1 中获取入参值方式示例如下：

代码区 Q + 编辑事件流 eventflow1    

当前页面 全局 事件流

query_one text6 query2 query666 eventflow1  eventflow2

打开弹窗

当时

IF 成功时  

内容 标题 是否显示取消按钮

JSON.stringify(event.detail)  

取消按钮的文字 确认按钮的文字

event.detail.appname  

最多 4 个字符  

取消按钮的文字颜色 确认按钮的文字颜色

#000000  #576B95 

当 query_one 执行成功后，则会回调事件流 eventflow1，效果如下图所示：



编辑事件流

eventflow1

保存 fx

当时

打开弹窗

IF 成功时

IF 失败时

Tips: 鼠标Hover上方条件可新增执行动作

标题: Query回调事件流示例

内容: JSON.stringify(event.detail)

是否显示取消按钮: fx

取消按钮的文字: event.detail.appname

最多 4 个字符

以上是页面事件流的介绍和使用示例，更多详情请参见 [事件执行动作](#)。

自定义变量

最近更新时间：2023-06-27 16:20:05

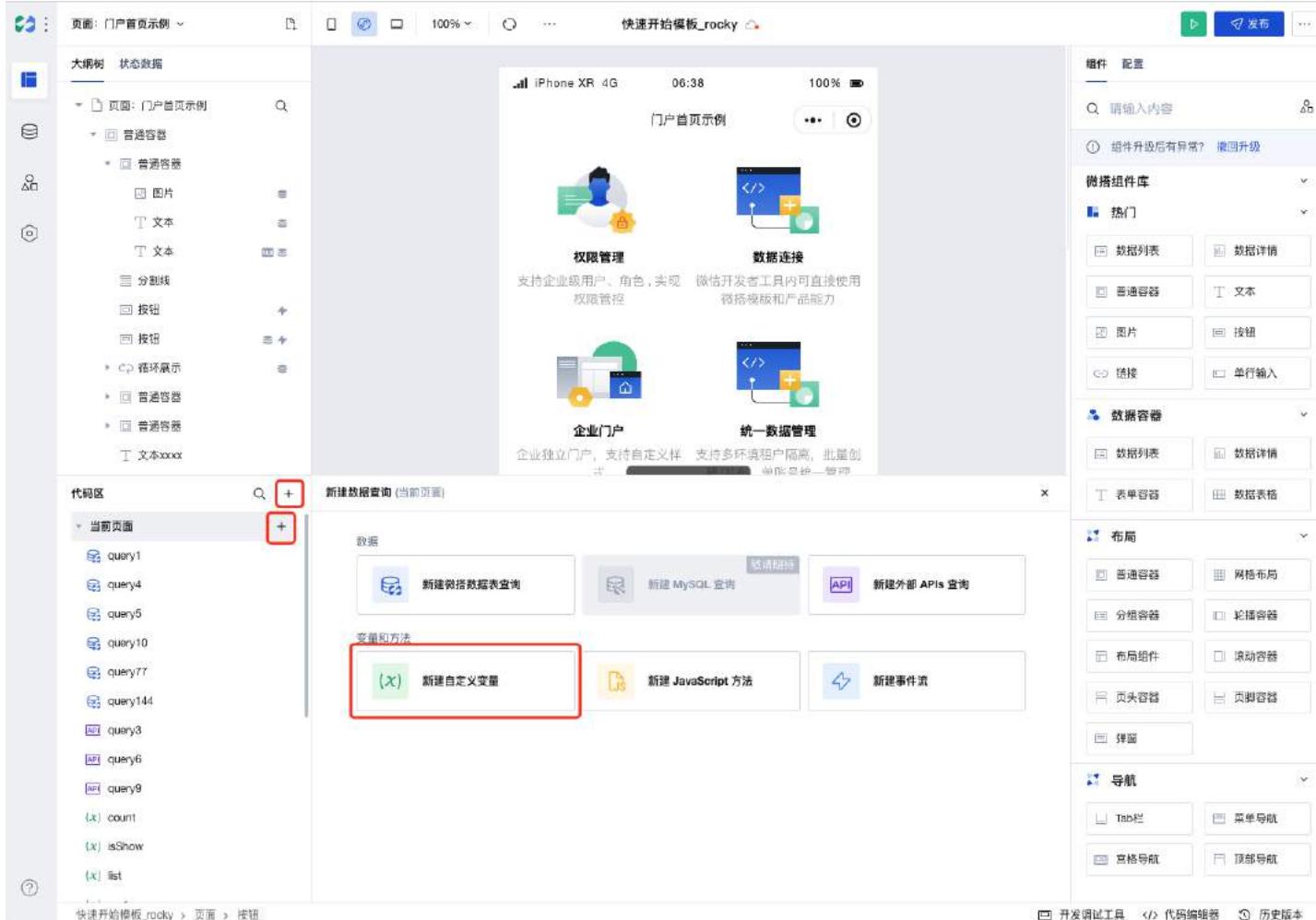
介绍

用于前端页面中的 JavaScript 临时状态变量，非持久化数据，当应用页面刷新时，临时的自定义变量会被重置。

参数	参数说明
变量名	变量唯一的名称，可用于变量引用。
变量类型	支持选择文本（String）、数字（Number）、布尔值（Boolean）、数组（Array）、对象（Object）。
初始值	变量的初始值，不同数据类型存在不同的初始值写法，示例如下： • String: test • Number: 123456 • Boolean: false • Array: [1, 2, 3] • Object: {key:"test",val:2}
作用域	变量生效的作用域，分为页面级作用域和应用全局作用域，可以按需声明。
备注	变量的使用说明，根据使用场景自定义填写。

变量的新建/定义

可在 code 代码区新建自定义变量。



The screenshot shows the Tencent Cloud Micro-Assembly low-code platform interface. On the left, there's a sidebar with '大纲树' (Outline Tree) showing a tree structure of components like '普通容器', '图片', '文本', etc. Below it is the '代码区' (Code Area) which contains a list of variables: 'query1', 'query4', 'query5', 'query10', 'query77', 'query144', 'query3', 'query6', 'query9', '(x) count', '(x) isShow', and '(x) list'. A red box highlights the '+' button next to '当前页面'. In the center, there's a preview of a mobile application with sections for '权限管理', '数据连接', '企业门户', and '统一数据管理'. To the right, there's a '组件' (Components) panel with various component categories like '热门', '数据容器', '布局', and '导航'. At the bottom, there's a navigation bar with links for '开发调试工具', '代码编辑器', and '历史版本'.

单击新建自定义变量，即可进行变量的新建。

The screenshot shows the 'Edit Custom Variable' dialog. The variable name is 'var3'. The type is set to 'Text(String)'. The default value is 'HelloWeda'. A note says '一个测试用的自定义JS变量'. The variable path is '\$page.dataset.state.var3'. There are '+' buttons next to 'Current Page' and 'Global' in the left sidebar.

快速开始模板_rocky > 页面 > 按钮

开

⚠ 注意：

在左侧代码区中，hover 到不同分组时（当前页面或全局），单击对应分组的 + 号，可分别建立不同作用域的自定义变量或自定义 JS 方法。

变量的引用和赋值示例

- 在自定义 JavaScript 代码进行引用：

例如：定义了一个全局自定义变量 balance，则实际引用路径为 \$app.dataset.state.balance。

若定义了一个页面作用域的自定义变量 chatList，则引用路径为 \$page.dataset.state.chatList。

- 在 fx 表达式中进行引用：

在表达式中调用 JS 方法的传参方式，与上述自定义 JavaScript 方法传参相同。

The screenshot shows the configuration panel for a text component. The 'Text' tab is selected. The component ID is 'text50'. In the 'Text Content' field, the expression '\$app.dataset.state.balance' is entered, which is highlighted with a red box. The 'Text Format' dropdown is set to '正文(默认)'.

- 对自定义变量的赋值

除了在自定义 JS 代码通过 = 对变量进行直接赋值，还可以通过调用事件的变量赋值节点来完成变量赋值操作。

添加事件: 点击时 (tap)

取消

保存



请选择下方的执行动作

选择后将对该动作进行配置，当【点击时 (tap)】应用将执行此动作

应用动作

调用数据源方法 (callDataSource)

触发流程 (callProcess)

变量赋值 (setState)

逻辑分支 (if)

Javascript代码 (code)

页面交互

进入变量赋值节点，选择所需要被赋值的变量（没有则新建即可），然后在后面输入框中输入对应值，可以是静态值，例如123，或者单击 **fx**，通过表达式使用变量或上一事件节点的返回值 **event 对象**（例如event.detail）进行赋值。

The screenshot shows the configuration of a 'Variable Assignment' node within a workflow. On the left, there's a tree view of components and nodes. In the center, the 'Variable Assignment' node is selected. It has two input fields: 'Variable Name' (set to '组件A') and 'Variable Value' (containing the static value '123'). A red box highlights the 'Variable Value' input field. On the right, there are buttons for saving and canceling changes.

自定义 JavaScript 方法

最近更新时间：2023-08-02 14:17:26

介绍

运行在前端的 JavaScript 代码块，遵循标准的 JS 语言规范。可通过自定义 JavaScript 方法实现自定义的业务逻辑，或者调用微搭官方的 前端 API 以及外部 JSSDK 等。

如何新建自定义 JavaScript 方法

- 建议通过编辑器左下角代码区，单击 + 直接新建自定义 JavaScript 方法。



说明：

自定义 JS 方法的作用域分为页面作用域和应用全局作用域，可以按需声明。默认情况下，使用页面作用域即可。

- 也可在代码编辑器中直接新建自定义 JavaScript 方法，具体请参见 [代码编辑器](#)。

如何调用自定义 JavaScript 方法

- 在内部自定义 JS 代码之间进行调用。

例如：定义了一个全局 JS 方法 `getBalance()`，则在代码中的调用路径为 `$app.common.getBalance(...)`。若定义了一个页面作用域的 JS 方法 `getChatList()`，则调用路径为 `$page.handler.getChatList(...)`。

说明：

自定义 JavaScript 方法以及自定义变量和 Query 对象等的引用路径都可以在编辑器左下角代码区 > 单击 ··· > 复制调用路径快捷获取，如下所示：

The screenshot shows the code editor interface. On the left is a sidebar with a tree view of code snippets: 'query1', 'query_one', 'query3', 'var1', 'var2', 'function1' (selected and highlighted with a blue background), 'eventflow1', 'eventflow2', and 'eventflow3'. The main area is titled '编辑 JavaScript 执行方法' (Edit JavaScript Execution Method) and contains a code editor with the following content:

```

1  /**
2  *
3  * 可以
4  * 可以
5  * 具备
6  * 注释
7  * 如何
8  * 帮助
9  */
10 /**
11 * @param
12 * @param
13 * @param
14 * @param
15 * @param
16 */

```

A context menu is open over the 'function1' entry in the sidebar, with the '复制成功' (Copied successfully) option highlighted. Other options visible in the menu include '复制调用路径' (Copy call path), '克隆' (Clone), and '删除' (Delete).

- 在表达式中进行调用。

在表达式中调用 JS 方法的传参方式，与上述自定义 JavaScript 方法传参相同。

The screenshot shows the expression editor with the following content:

基础属性

文本内容: \$app.common.getBalance()

文本格式: 正文(默认)

- 在事件面板的 JavaScript 代码方法节点中进行调用，例如：

The screenshot shows the event panel configuration for a '点击 (tap)' event. The '高级属性' (Advanced Properties) section on the right shows the event type as '点击 (tap)'. In the '应用动作' (Action) section, the 'Javscript代码 (code)' input field is highlighted with a blue border, containing the code '\$app.common.getBalance()'. The '事件' (Event) section also lists the '点击 (tap)' event.

添加事件: 页面加载 (load)

保存

当页面加载 (load)时

app:getUserInfo

IF 成功时

IF 失败时

Tips: 鼠标Hover上方条件可新增执行动作

方法类型

 已有方法 添加新方法

执行方法

getUserInfo

入参

fx

出参

保存为 请输入变量名

管理自定义方法

应用场景示例

示例1：通过自定义 JS 代码打印当前登录用户信息

该示例以点击某个按钮时，触发一个自定义 JS 方法，来打印当前登录用户的详细信息。

1. 首先，在编辑器左下角代码区单击 + 新建一个 JavaScript 方法，例如命名为 logger 并保存，函数体示例代码如下。

```
export default function({event, data}) {
  console.log('data', data) // 打印入参data
  console.log('event', event) // 打印event对象
  if(!data) {
    throw Error; // 如需要在事件中触发动作执行失败的分支，则在判断分支中throw Error
  }
  return 'success'
}
```

② 说明：

在多个事件动作节点的执行过程中，某个节点执行完成后默认进入执行成功的分支，如果需要在节点执行之后触发动作执行失败的分支，则可在自定义 JS 方法中添加抛出异常代码，如上述的 throw Error 即可。

代码区
+
编辑 JavaScript 执行方法
logger
... 保存

+ 当前页面
var14
var22
func22
function1
function2
getUserInfo
logger
query1
testfunc
eventflow1
+ 全局

JS 代码

```

1  /**
2   * @param {Object} event - 事件对象
3   * @param {string} event.type - 事件名
4   * @param {any} event.detail - 事件携带自定义数据
5   *
6   * @param {Object} data
7   * @param {any} data.target - 获取事件传参的数据
8   */
9
10 export default function({event, data}) {
11   console.log('data', data) // 打印入参data
12   console.log('event', event) // 打印系统参数event
13   return 'success'
14 }
```

备注
自定义打印日志方法

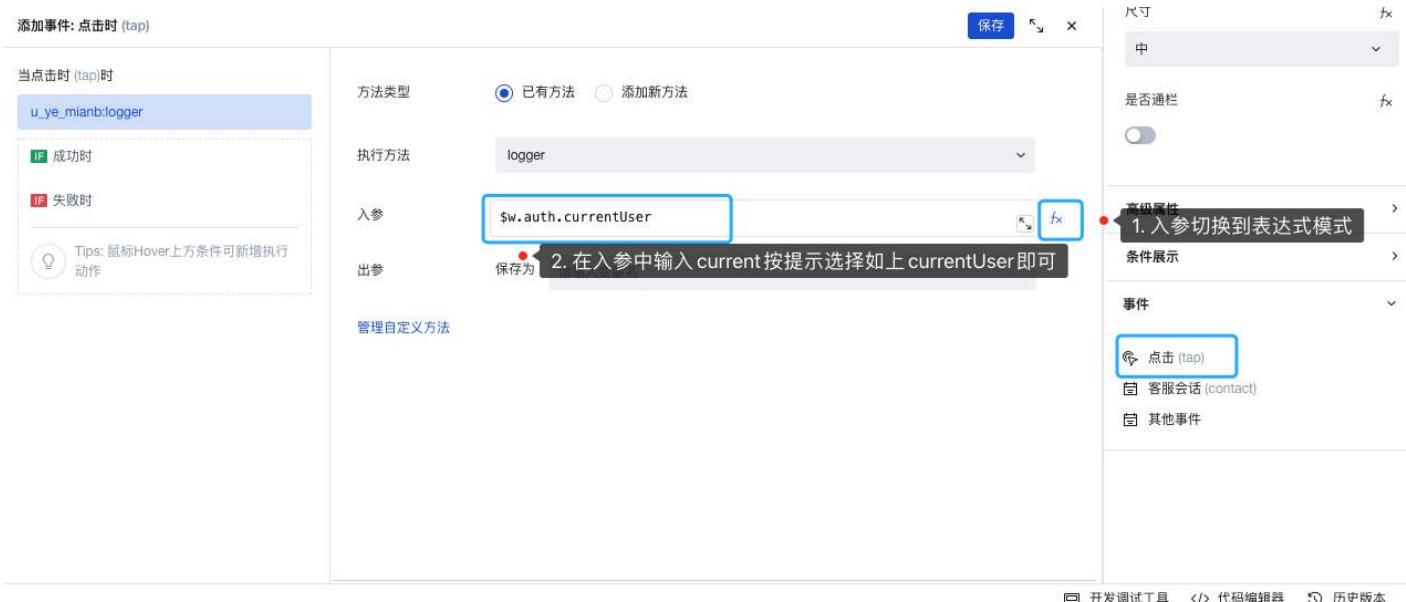
版权所有：腾讯云计算（北京）有限责任公司

第53共379页

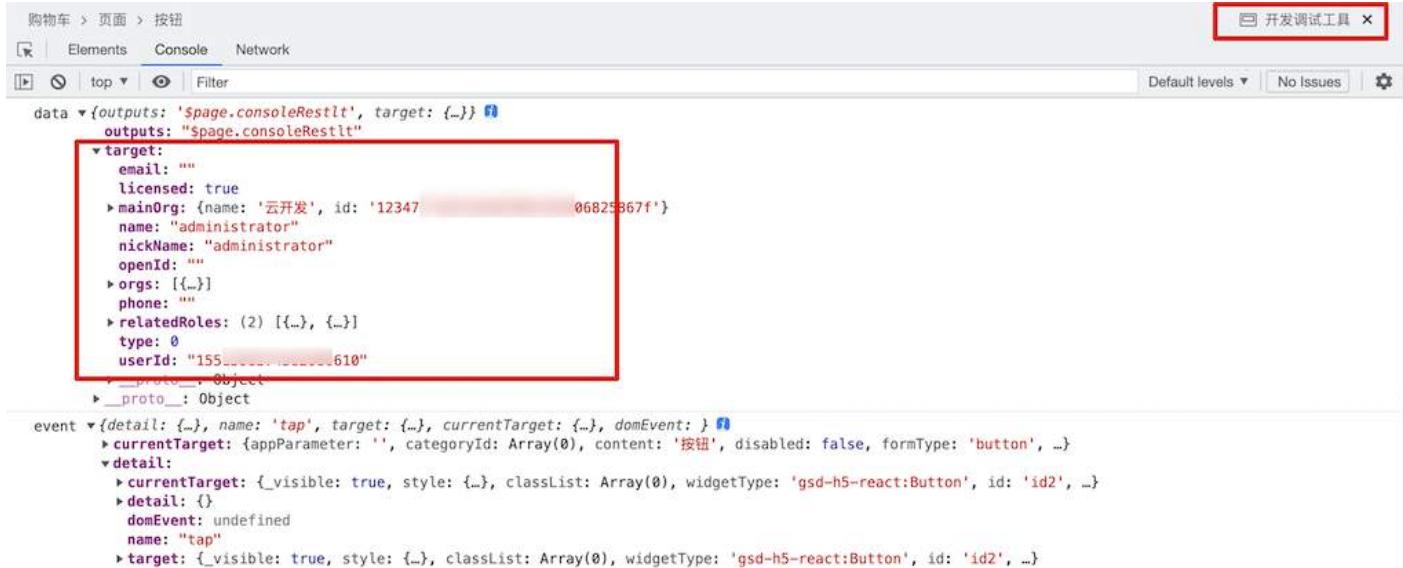
2. 然后，可在编辑区拖入一个按钮组件，选中按钮组件后，在右侧组件属性面板的右下角为其配置点击事件，打开事件面板 > 选择 JavaScript 代码方法节点 > 选择已有方法中刚新建的 logger 自定义 JS 方法，如下所示：



3. 在事件面板中，为选择的自定义 JS 方法 logger 配置方法入参，例如根据表达式提示选择入参为 \$w.auth.currentUser (currentUser 为微搭 内置系统变量，表示当前登录用户），完成配置后单击保存。



4. 确认上述配置后，回到编辑区单击“按钮”即可触发事件对应的自定义 JS 方法，打开编辑器右下角开发调试工具，即可查看单击时的打印日志，如下所示：



```

购物车 > 页面 > 按钮
Elements Console Network
top Filter
Default levels ▾ No Issues
开发调试工具

data ▶ {outputs: '$page.consoleRestlt', target: {}}
  outputs: "$page.consoleRestlt"
    ▶ target:
      email: ""
      licensed: true
      mainOrg: {name: '云开发', id: '1234706825367f'}
        name: "administrator"
        nickName: "administrator"
        openId: ""
      orgs: [{}]
      phone: ""
    relatedRoles: (2) [{...}, ...]
    type: 0
    userId: "155.....610"
    __proto__: Object
  __proto__: Object

event ▶ {detail: {...}, name: 'tap', target: {}, currentTarget: {}, domEvent: {}}
  ▶ currentTarget: {appParameter: '', categoryId: Array(0), content: '按钮', disabled: false, formType: 'button', ...}
  ▶ detail:
    ▶ currentTarget: {_visible: true, style: {}}, classList: Array(0), widgetType: 'gsd-h5-react:Button', id: 'id2', ...
    ▶ detail: {}
    domEvent: undefined
    name: "tap"
  ▶ target: {_visible: true, style: {}}, classList: Array(0), widgetType: 'gsd-h5-react:Button', id: 'id2', ...

```

示例2：实现通过用户 openid 来查询用户表中的详细信息

该示例以在页面加载时，通过一个自定义 JS 方法，实现通过已知的小程序 openid 来从数据表中查询当前用户的更多详细信息为例，具体操作过程如下。

1. 首先，在编辑器左下角代码区单击 + 新建一个 JavaScript 方法，例如命名为 getUserInfo，函数体示例代码如下。

```

export default async function({event, data}) {

  var wxa = await app.utils.getWXContext();
  const openid = wxa.OPENID || wxa.FROM_OPENID || app.auth.currentUser.openId;
  console.log("get userinfo openid: ", openid);

  // 提前定义一个全局变量`userInfo`，用于存储获取到的用户信息，方便其他地方引用
  $app.dataset.state.userInfo = await app.cloud.callModel({
    name: 'diy_user_xxxx', // 数据源标识，请自行新建一个用户表`diy_user`，之后可在数据源列表中看到该标识
    methodName: 'wedaGetItem',
    params: {
      where: [
        {
          key: "openid",
          rel: "eq",
          val: openid,
        },
      ],
    },
  });

  console.log("getuserinfo: ", $app.dataset.state.userInfo);

}

```

2. 然后，在编辑器左侧大纲树选择页面根节点之后，在右侧属性面板右下角可选择对应的事件触发条件，例如选择页面加载，在之后打开的事件面板中选择 JavaScript 代码节点，即可完成在页面加载时，例如上自定义 JS 方法的调用，参考步骤如下：

The screenshot shows the configuration interface for a mobile page template named '门户首页示例'. The left sidebar contains a tree view of page components like '普通容器' (General Container), '图片' (Image), '文本' (Text), etc. The main area displays a preview of the page on an iPhone XR with the title '门户首页示例'. The right panel includes sections for '组件' (Components), '页面URL参数' (Page URL Parameters), '屏幕配置' (Screen Configuration), '页面分享' (Page Share), and '事件' (Events). A central configuration panel shows the addition of an event '页面加载 (load)' with the action 'Javascipt代码 (code)' selected.

3. 完成上述配置之后，可刷新页面，打开编辑器右下角的开发调试工具，即看到打印日志。

This screenshot shows the developer tools interface for the same mobile page template. It highlights the '事件' (Events) panel where the '当页面加载 (load) 时' (When page loads) section is selected. In the '方法类型' (Method Type) dropdown, '已有方法' (Existing method) is chosen. The '执行方法' (Execute method) dropdown is set to '选择要执行的JavaScript方法' (Select the JavaScript method to execute). A search bar shows the method name 'query1'. Step 3 indicates '3. 选择事件面板的JavaScript节点' (Select the JavaScript node in the event panel). Step 4 indicates '4. 选择刚刚新建的JS方法，保存即可' (Select the newly created JS method, save it). The right side of the interface shows other panels for '属性' (Properties), '样式' (Style), '页面信息' (Page Information), and '屏幕配置' (Screen Configuration).

Query 数据查询介绍

最近更新时间：2023-08-14 14:25:44

Query 介绍

Query 是一个静态 JS 对象，主要作用于后端相关的数据获取和更新等操作，支持关联主流数据库 MySQL 以及任何第三方的 HTTP API。其可以通过与变量类似的方式在组件配置的表达式中进行引用，例如 `query1.data`，可以很方便的让组件从各种数据源读取或写入数据，具备与变量一致的生命周期以及作用域。

说明：

Query 的作用域：分为页面级别作用域的 Query 和应用全局作用域的 Query，开发者可按需声明。当选择定义全局作用域的 Query 时，应注意避免引用页面作用域的局部变量作为入参，以避免入参值为空时导致 Query 执行异常。

Query 触发方式介绍



手动触发更新

手动触发需要用户主动通过单击某按钮或者类型前端行为，查询 Query 才会触发执行。手动触发适用于由事件触发时的按需更新场景（例如单击某个按钮触发查询）。

入参变化时自动执行

对于读取类请求的 Query，可在入参更新时自动刷新。适合根据查询条件的输入变化，实时响应查询结果（例如列表的筛选条件变化等）的场景。

而对于写请求（包括增/删/改）的 Query 查询类型，并不建议使用入参变化时自动执行的触发方式，一般是用户执行某个行为或者完成输入后才触发写操作，这种写请求的场景一般选择手动触发方式。

注意：

针对写入数据（增/删/改）的 Query 数据查询，不建议选择入参变化自动更新方式，以避免反复操作导致的脏数据。

Query 的引用

可以直接在表达式、自定义代码以及事件节点中使用：

- 例如在表达式中引入 Query 的返回结果，输入：`$w.query1.data`。
- 例如在自定义 JS 代码手动触发查询某个 Query，输入：`$w.query1.trigger()`，或引用 Query 的返回数据：`userList = $w.query1.data`。
- 例如在用户的某个点击行为时，手动触发调用 Query。



Query 执行成功/失败的回调

在完成 Query 运行后，如果还希望触发其他的行为动作（例如显示加载成功或失败的 toast 提示），则可通过事件回调来完成对应事件流的调用（可复用已定义的事件流或者新建事件流）。

编辑数据查询 query3

触发方式 入参变化时自动执行 手动触发执行

方法 提问通用(ask)

入参 提问内容 (文本 | 短文本) \$app.label

备注信息

事件回调

- 成功时 eventflow1.未绑定入参
- 失败时 eventflow2.未绑定入参
- [新建事件回调](#)

如上图，在定义 Query 查询时，在下方选择执行成功或失败时的回调，然后关联对应的事件流（没有可新建），则在查询成功或失败时做出相应的交互行为。

Query 的内置属性和方法

可访问属性有

参数名称	参数类型	参数描述
\$w.query1.data	Any	数据请求成功时，返回的数据结果，默认值：null。
\$w.query1.error	Error	数据请求失败时，返回的 Error 错误对象，默认值：null。
\$w.query1.isFetching	Boolean	数据请求状态，是否在请求加载中，默认值：false。

可访问方法有

参数名称	参数描述
\$w.query1.reset()	重置 Query 对象的 data 和 error 值为 null。
\$w.query1.trigger()	在代码中手动触发 Query 的执行。如果需要触发 Query 时进行传参，可以通过类似 \$w.query1.trigger({aaa:10}) 的方式传入额外数据，然后在 Query 的配置中通过绑定表达式 params.aaa 来获取到传入的数据10（进行手动触发时传参）。

示例：

手动调用 Query 的传参示例：例如在点击事件中调用 Query 时的入参配置如下：

添加事件: 点击时 (tap)

当点击时 (tap) 时

调用数据查询

成功时

失败时

Tips: 显示Hover上方条件可新增执行动作

选择Query: query_test3

方法: 触发Trigger

入参: {{name: \$w.Input1.value}}

出参: 保存为 请输入变量名

然后可在 Query 配置中引用该参数。

MySQL配置名称: mysqldemo_qmsrx6i
触发方式: 手动触发执行
SQL语句: SELECT * FROM users WHERE users.name = "{params.name}"
当前MySQL表信息:
users
id int
name text
time timestamp

想了解更多 Query 对象的定义封装，请参见 [Query 实例介绍](#)。

Query 的数据来源

微搭数据查询 Query 的数据来源目前主要有以下几种（Query 的数据来源未来会持续增加），可通过编辑器左下角代码区单击 + 打开新建面板，来进行不同数据来源 Query 对象的新建。

内置微搭数据表查询

可用于查询操作微搭平台内置的数据表，适用于简单的数据表 CURD 场景，数据库由微搭维护。

选择数据表: 用户表 (@_user_20647ar)
触发方式: 入参变化时自动执行
方法
入参
备注信息
事件回调: 成功时 -> eventflow1
+ 新增事件回调

了解更多数据表/数据模型相关介绍，请参见 [数据表概述](#)。

外部 HTTP APIs 查询

可用于查询操作 HTTP 协议的外部 API 数据或服务，适用于依赖 HTTP 标准接口做数据对接的场景。

选择APIs: 请选择
HTTP请求
腾讯文档
腾讯会议
腾讯会议
腾讯网盘数据
+ 新建APIs
方法
入参
备注信息
事件回调
+ 新建事件回调

了解更多 APIs 相关介绍，请参见 [自定义 API 概述](#)。

外部 MySQL 数据查询

可用于查询操作有读写权限的外部 MySQL 数据库，适用于开发者有自主维护 MySQL 数据库的场景，以及相对复杂业务多表关联查询等场景，数据库需自行保障稳定性且确保有外网访问权限。

了解更多 MySQL 连接相关介绍，请参见 [MySQL 连接器](#)。

SQL语句数据查询的使用说明

- SQL 语句中可支持插入前端 JS 表达式，JS 表达式需使用 {{ }} 进行包裹，如：{{ \$w.input1.value }}，例如需要根据单行输入组件 input1 的输入值进行用户信息的查询，SQL 语句可以写作：SELECT * FROM users WHERE users.name={{ \$w.input1.value }}，再例如使用 SQL 进行模糊查询：SELECT * FROM users WHERE users.name like {{ '%' + \$w.input1.value + '%' }}。

△ 注意：

在 SQL 中使用 JS 表达式时，如需要字符串拼装，则必须在 JS 表达式中进行，即 {{ "string" + js_expr }}，SQL 语句中不支持使用 string{{js_expr}} 进行拼接。

- SQL 的批量操作示例：例如批量查询或批量删除等场景，可通过传入前端数组变量来实现。假设新建了自定义数组变量 \$w.page.dataset.state.array，默认值为 [50,51,52]，那么 SQL 语句可写作：SELECT * FROM users WHERE users.id IN ({{ \$w.page.dataset.state.array }})。
- SQL 语句中如需条件判断的查询示例：假设场景为查询某个用户列表时，当某个输入框组件 (input1) 的值不为空时则进行条件查询过滤，否则不过滤查询全量，参考 SQL 可以写作：SELECT * FROM users WHERE {{ \$w.input1.value ? ("name=''' + \$w.input1.value + ''') : 'true' }}。
- 在使用 SQL 语句进行数据查询时，如果涉及针对当前登录用户的数据查询，为保障安全性，建议开发者使用服务端保留的占位符 SERVER.xxx 作为查询条件代入查询，使用占位符时服务端会校验当前登录用户的合法性。例如，在微信小程序中根据当前用户的 OPENID 进行数据查询，则查询 SQL 条件应该写作：SELECT * FROM users WHERE users.openid={{ SERVER.OPENID }}。
- 目前保留字 SERVER 对象下有如下字段可使用：
 - SERVER.USERID：表示当前用户 USERID，与前端系统变量 \$w.auth.currentUser.userId 取值一致。
 - SERVER.OPENID：表示小程序下当前登录用户的 OPENID，与前端系统变量 \$w.auth.currentUser.openId 取值一致。

使用场景示例

示例1：获取微搭内置数据表中的列表数据和单条详情数据

列表页的配置

- 首先，在代码区单击 + 可新建一个查询列表的 Query，选择新建微搭数据表查询，并且命名为 query_list（也可以使用默认名，例如 Query1）。



相关的配置参考如下：



② 说明：

其中，截图中的 AI 产品表为在微搭自行新建内置数据表，关于如何新建微搭数据表，请参见 [数据源概述](#)。

完成 query_list 的配置后，也可以通过单击运行，来实时查看 Query 执行的返回结果，如下所示。

name	id	time
weda111	1	2023-03-01T13:58:19Z
weda222	2	2023-03-14T13:58:28Z
rocky	3	
rocky222	4	

2. 然后，完成 Query 新建之后，就可以在组件中通过表达式直接用 query_list 绑定和渲染列表数据了。如下图，我们可以往编辑区拖入多个所需要组件，并且分别对组件属性进行数据绑定。

② 说明：

这里我们使用循环展示组件作为列表展示的示例（这里也可以选择数据列表或数据表格组件进行绑定，尤其列表数据量大需要分页时），参见配置如下：

The screenshot shows the Tencent Cloud Low-Code Platform's visual editor. On the left, the '大纲树' (Outline Tree) shows a tree structure with '页面: 列表页面测试' expanded, revealing '循环展示' (Repeater), '普通容器' (General Container), '文本' (Text), and other components. Below it is the '代码区' (Code Area) with code snippets for 'query1' and 'query_list'. The main workspace displays a mobile preview for an iPhone XR 4G device at 08:44 with 100% battery. The preview shows a placeholder '拖拽组件到此处' (Drag component here). To the right is the '配置' (Configuration) panel for the selected '循环展示' (Repeater) component. The '属性' (Properties) tab is active. The '基础属性' (Basic Properties) section includes a '循环数据' (Loop Data) field set to '\$w.query_list.data'. A red box highlights this field and the expanded JSON data view below it, which shows a total of 32 records with various fields like 'owner', 'count', 'avatar', and 'appcode'. The '条件展示' (Conditional Display) section is also visible.

3. 再对循环组件内的其他文本组件进行字段绑定，参见配置如下：

This screenshot continues from the previous one, showing the configuration of a text component within the repeater loop. The '大纲树' (Outline Tree) now has '文本' (Text) selected under '循环展示'. The '配置' (Configuration) panel shows the '文本' component with '组件ID' (Component ID) set to 'text1'. The '基础属性' (Basic Properties) section has '文本内容' (Text Content) set to '\$w.item_repeater3.appname'. A red box highlights this field and the expanded JSON data view below it, specifically pointing to the 'appname' field under the 'item_repeater3' object. A tooltip provides instructions: '3. 循环组件下可直接选择循环对象下的属性字段进行绑定, 如 appname' (In a loop component, you can directly select the attribute fields under the loop object for binding, such as appname). The '代码区' (Code Area) shows the generated code: `# text1/文本内容 1 \$w.item_repeater3.appname` and the resulting output 'AI翻译官'.

4. 分别完成数据绑定后，即可看到如下效果，一个简单的列表渲染就完成了。

详情页的配置

接下来，我们再新建一个详情页，通过刚搭建的这个列表跳转到这个详情页，展示对应的详情内容，完成一个列表展示到详情展示的闭环，这样的话我们就需要用一个查询单条的 Query。

1. 首先，在新建 Query 之前，先新建一个内容详情页页面，并且给这个页面配置一个 URL 参数变量，参考配置如下：

2. 完成页面新建之后，我们同样的在代码区单击 + 可新建一个查询单条的 Query，选择新建微搭数据表查询，并且命名为 query_one，相关的配置参见如下：

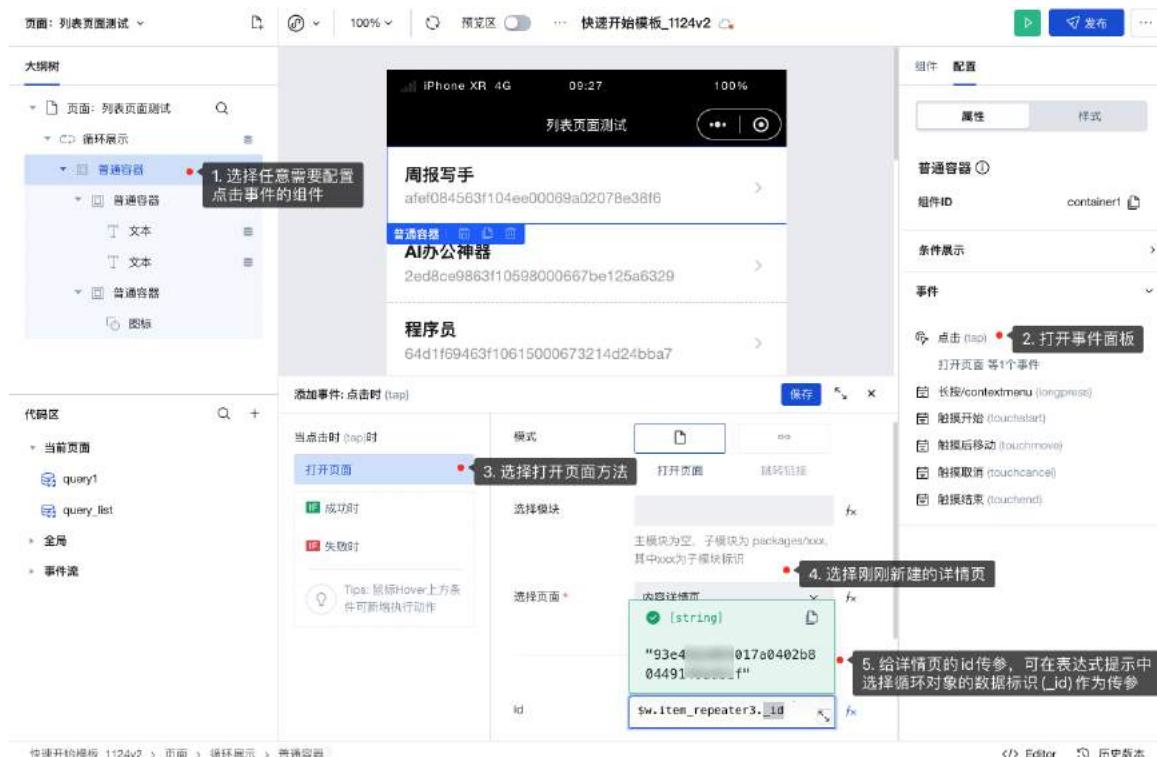
3. 然后，我们就可以用 query_one 给详情页内的组件进行数据绑定了。例如给文本组件绑定数据的参见配置如下：



以上，一个详情页就搭建完了，有兴趣也可以继续在此基础上按需扩展。

列表页跳转详情页的配置

最后，切换到之前的列表页，我们给列表页面做一个页面跳转配置，并完成详情页的 URL 参数（id）进行传值，相关配置很简单，参见如下：



示例2：从第三方 HTTP API 中获取用户列表数据

从第三方 HTTP API 中获取数据的配置与上一个示例从微搭数据表获取数据流程大致相同。

1. 首先，在代码区单击 + 可新建一个查询列表的 Query，选择新建外部 APIs 查询，可重命名为 query_api 或使用默认命名。



相关的配置参考如下：



② 说明：

其中，截图中的 测试 diy 连接器为自行新建的数据源 APIs 连接器，关于如何新建数据源 APIs，请参见 [自定义 API 概述](#)。

2. 然后，完成 Query 的新建之后，就可以在组件中通过表达式直接用 query_api 绑定和渲染列表数据了。如下图，我们可以往编辑区拖入多个所需要组件，并且分别对组件属性进行数据绑定。

② 说明：

这里我们使用循环展示组件作为列表展示的示例（这里也可以选择数据列表或数据表格组件进行绑定，尤其列表数据量大需要分页时），参见配置如下：

页面：列表页面测试

大纲树

- 页面：列表页面测试
- 循环展示APIs (highlighted with a red box)
- 文本
- 循环展示数据表

属性

样式

循环展示

组件ID: repeated4

基础属性

表达式中输入 query_api，按自动提示选择对应的列表数组字段

```
sw.query_api.data.data
  (array)
  [
    {
      "last_name": "Bluth",
      "id": 1,
      "avatar": "https://reqres.in/images/l-image.jpg"
    }
  ]
```

代码区

当前页面

query_list

query_api (highlighted with a red box)

编辑数据查询

query_api

选择APIs: 测试DIY连接器 (or_pdbwmc7)

3. 最后，再对循环组件内的其他文本组件进行字段绑定，一个简单的通过外部 APIs 请求的数据列表就完成了，此处后续步骤与上一个示例获取微搭内置数据表中的列表数据和单条详情数据一致，在此不再赘述。

示例3：获取外部 MySQL 的用户列表以及删除某条数据

从外部 MySQL 获取数据之前，我们需要先新建一个 MySQL 连接器，参考步骤如下。

1. 在代码区单击 + 可新建一个查询列表的 Query，选择新建 MySQL 查询，可重命名为 query_list 或使用默认命名。

代码区

当前页面

query1

query_insert

query_list

query_one

var1

全局

新建数据查询 (当前页面)

数据 Query

新建 MySQL 查询 (highlighted with a red box)

新建外部 APIs 查询

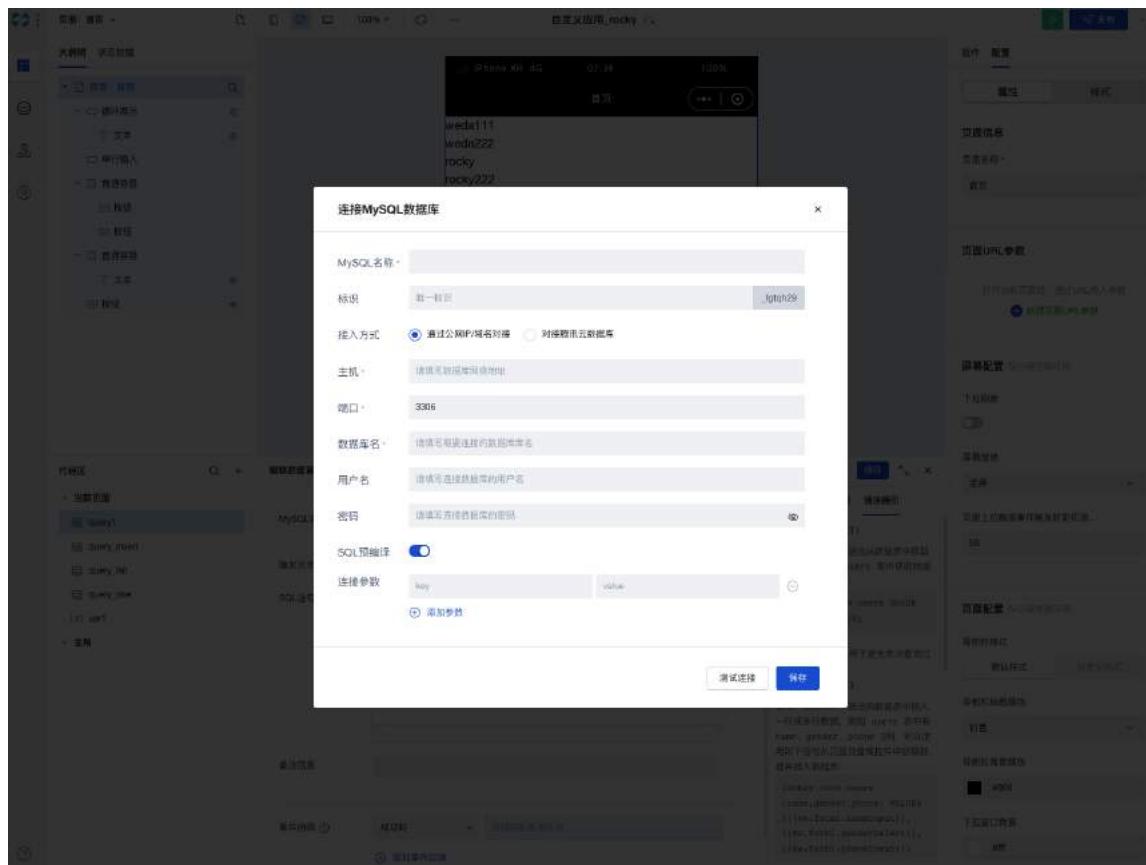
变量和方法

新建自定义变量

新建 JavaScript 方法

新建事件流

填写好需要创建的内容后单击保存。



如果是首次新建 MySQL 连接器，则会直接打开 MySQL 连接的配置弹层，如果不是则可在 MySQL 的下拉列表中找到新建入口，如下所示。

This screenshot shows the MySQL configuration interface. On the left, there's a sidebar with 'Current Page' containing items like 'query1', 'query_insert', 'query_list', 'query_one', '(x) var1', and '全局'. The main area has tabs for 'Current MySQL Table Information' and 'Syntax Guide'. A red box highlights the 'New MySQL Connection' (新建MySQL连接) option in the dropdown menu under 'MySQL Name'.

2. 完成 MySQL 配置并且测试连接正常之后，即可进行 Query 的配置，编写 SQL 语句，例如输入：SELECT * FROM users ORDER BY id LIMIT 10，单击运行，即可在结果弹层中看到查询到的数据结果，如下图所示。

The screenshot shows the configuration of a MySQL query component within the platform's interface. On the left, there's a sidebar with a tree view of components like '大组件' (Large Components) and '编辑区' (Editor). In the center, a preview window shows a mobile phone screen with a search form and a table of user data. To the right, the configuration panel is open, showing tabs for '属性' (Properties), '样式' (Style), and '事件' (Events). Under '属性', there are sections for '页面信息' (Page Information) and '页面URL参数' (Page URL Parameters). The 'SQL语句' (SQL Statement) tab is selected, containing the query: `SELECT * FROM users ORDER BY id LIMIT 10;`. Below it, there are sections for '事件回调' (Event Callback) and '备注信息' (Remarks). The bottom right of the configuration panel has tabs for '开发调试工具' (Development Debug Tools), '代码检测器' (Code Detector), and '历史版本' (History Versions).

3. 完成 Query 的配置并且运行返回正常之后，即完成了外部 MySQL 数据库的用户列表数据的获取，接下来可以通过在列表中绑定 query_list.data 来对获取到的用户列表数据进行渲染展示。后续的数据绑定步骤配置与上一个示例的流程大致相同。如下图，我们可以往编辑区拖入多个所需要组件，并且分别对组件属性进行数据绑定。

② 说明：

这里我们使用循环展示组件作为列表展示的示例（这里也可以选择数据列表或数据表格组件进行绑定，尤其列表数据量大需要分页时），循环展示组件参见配置如下：

The screenshot shows the configuration of a repeater component in the Tencent Cloud Low-Code Platform. The main interface includes:

- 左侧工具栏:** 大模型、状态数据、组件库、全局变量、普通容器、按钮。
- 上方操作栏:** 页面: 首页, 100%, ...
- 中间设计区:**
 - 组件树:** 右侧显示了当前选中的“循环展示”组件。
 - 预览窗口:** 显示了一个手机设备的预览，显示了用户名列表：weda111, weda222, rocky, rocky222。
 - 属性面板:** 显示了“循环展示”的属性，包括组件ID (repeater1)。
 - 右侧配置区:**
 - 数据:** 显示了从 MySQL 表 query_list 中查询的数据结果，包含 name, id, time 三列。
 - SQL 语句:** SELECT * FROM users ORDER BY id LIMIT 10
 - 事件回调:** 成功时: 插件并配置事件流。
- 下方状态栏:** 开发调试工具, 代码编辑器, 历史版本。

4. 再对循环组件内的其他文本组件进行字段绑定，参见配置如下：

The screenshot shows the configuration of a text component within a repeater component in the Tencent Cloud Low-Code Platform. The main interface includes:

- 左侧工具栏:** 大模型、状态数据、组件库、全局变量、普通容器、按钮。
- 上方操作栏:** 页面: 首页, 100%, ...
- 中间设计区:**
 - 组件树:** 右侧显示了当前选中的“文本”组件。
 - 预览窗口:** 显示了一个手机设备的预览，显示了用户名列表：weda111, weda222, rocky, rocky222。
 - 属性面板:** 显示了“文本”的属性，包括组件ID (text1)。
 - 右侧配置区:**
 - 文本:** 文本内容: \${sv.item_repeater1.name}，右侧展示了文本内容的表达式树，高亮了“\${sv.item_repeater1.name}”部分。
 - 基础属性:** 文本内容: \${sv.item_repeater1.name}
 - 事件回调:** 成功时: 插件并配置事件流。
- 下方状态栏:** 开发调试工具, 代码编辑器, 历史版本。

5. 分别完成上图所述的数据绑定后，一个简单的从外部 MySQL 获取的用户列表数据渲染就完成了。

6. 接下来我们来实现从列表中删除某条数据。先新建一个自定义变量和一个用于执行删除的 Query。自定义变量用于存储点击某一行时对应的数据 ID，例如变量命名为 current_row_id，如下所示：



用于执行删除的 Query 则命名为 query_del，SQL语句为：DELETE FROM users WHERE id={{\$w.page.dataset.state.current_row_id}}，如下所示：



7. 完成自定义变量和 Query 新建之后，开始进行组件的配置。这里我们为了达到更好的列表展示效果，可以把前面的循环展示组件，换成数据列表组件，数据列表与循环展示组件的数据绑定流程类似，在数据列表的数据源中选择表达式，然后绑定第1步中新建的 query_list 即可。



8. 完成数据绑定后，即可看到渲染后的列表效果。然后往数据列表对应的节点拖入一个按钮，按钮标题可修改为删除，具体结构如下图所示：

The screenshot shows the low-code development interface for creating a mobile application. On the left, there's a sidebar with a tree view of components and a code editor area. In the center, a preview window shows a list of items with a 'Delete' button next to each. A red box highlights the 'Delete' button in the first item. To the right, a configuration panel for a 'button' component is open, with its 'content' field set to '删除'. Below the preview, an event configuration panel shows a 'click' event being triggered by a 'trigger' node, which is connected to a 'query_del' query and a 'trigger' method. The 'current_row_id' variable is assigned the value of the current row's ID.

9. 配置删除按钮对应事件：选中按钮组件，在组件属性面板右下角配置对应的点击动作。

i. 首先配置第1个动作节点变量赋值，即将单击对应行的数据 ID 复制给前面新建的自定义变量 current_row_id，如上图所示。

This screenshot shows the event configuration for the 'Delete' button. It's a 'click' event with the following configuration:

- 选择Query:** query_del (highlighted with a red box)
- 方法:** 触发Trigger (highlighted with a red box)
- 出参:** 保存为: 请输入变量名 (the variable name is currently empty)

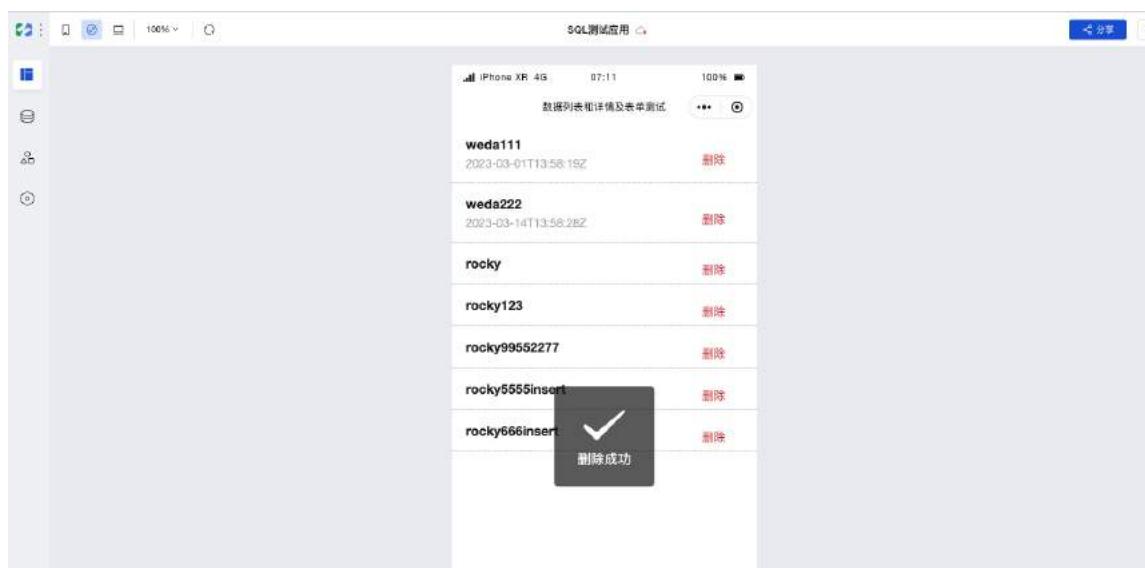
ii. 第2个动作节点选择调用数据查询，配置项如上图所示。



iii. 上述的第3个动作刷新数据列表，和第4个动作显示消息提示均为可选配置，主要用作交互提示，具体配置如图所示，可按需参考。



0. 完成上述配置后，可通过单击编辑器右上角预览按钮，打开预览界面来进行操作和查看效果。如下所示，单击页面上的删除，即可删除对应的行数据，同时完成删除后的数据列表也会同步刷新。



示例4：向外部 MySQL 数据库新增一条用户数据

1. 在编辑区拖入一个单行输入组件，以及一个按钮组件，大纲树结构参考下图所示，将单行输入组件标题修改为“用户名”，按钮标题修改为“提交数据”。同时我们可以看到单行输入的组件 id 为 input1，在接下来的 SQL 语句中会用到。

The screenshot shows the Tencent Cloud Low-Code Platform interface. On the left, the '大纲树' (Outline Tree) shows components like '单行输入' (Single-line Input) and '按钮' (Button). In the center, a preview window shows a mobile phone screen with a '用户名' (Username) input field and a '提交数据' (Submit Data) button. The right side shows the '组件' (Component) configuration panel for the '单行输入' component, with the '组件ID' set to 'input1'. Below it is a '编辑数据查询' (Edit Data Query) panel for a 'query_insert' entry. The 'MySQL名称' (MySQL Name) is set to 'sql_rockydb_lnwulqf'. The 'SQL语句' (SQL Statement) field contains the following code:

```
INSERT INTO users (name)
VALUES ({$w.input1.value
+ Math.floor(Math.random()
* 1000)})
```

The '备注信息' (Remarks) field is set to 'insert'.

2. 然后，我们开始新建新增数据的 Query，在代码区单击 + 可新建一个查询列表的 Query，选择新建 MySQL 查询，可重命名为 query_insert 或使用默认命名，在 MySQL 名称的下拉列表中，选择之前新建好的 MySQL 连接器。

△ 注意：

前置步骤，提前完成 MySQL 连接器的配置新建，可参考上一个示例。

3. 接着在 query_insert 配置中编写 SQL 语句，例如输入：INSERT INTO users (name) VALUES ({\$w.input1.value})，表示将输入框的值作为一条新数据，插入到 MySQL 数据库的 users 表中的 name 字段，单击运行，即可在弹层查看 Query 的执行结果（执行成功则表示数据完成了插入，如果是测试效

果请注意脏数据的处理），如下图所示。

The screenshot shows the configuration of a 'query_insert' component. In the '代码区' (Code Area), the 'SQL语句' (SQL Statement) field contains the following code:

```
INSERT INTO users (name) VALUES ({$w.input1.value})
```

The '事件设置' (Event Settings) section shows the event type as '成功时' (Success). A note in the interface states: '针对上述新增数据的 Query 数据查询，会默认选择手动触发执行，以避免入参变化自动执行导致的反复插入数据。' (For the above newly added data, the Query data query will default to manual triggering execution, to avoid repeated insertions caused by parameter changes automatically triggering execution.)

⚠ 注意：

针对上述新增数据的 Query 数据查询，会默认选择手动触发执行，以避免入参变化自动执行导致的反复插入数据。

4. 最后，再选中之前拖入的按钮组件，在组件属性区配置点击事件，即通过点击按钮时来触发前面 Query 的执行。具体配置参考下图所示。

The screenshot shows the configuration of a 'button2' component. In the '事件' (Events) section, the '点击时' (On Click) event is selected. Under '选择Query' (Select Query), 'query.insert' is chosen. Under '方法' (Method), '触发Trigger' (Trigger Trigger) is selected. The '事件' (Events) section also includes '点我' (Click Me) and '看谁会选 (context)'.

5. 完成上述配置后，即可通过单击提交数据来进行输入框用户名的提交，每次单击都会向外部数据库的 users 表插入一条数据。

以上仅为使用 SQL Query 的数据插入示例，实际生产环境请注意提交数据的合法性校验以及脏数据的处理。

代码编辑器

最近更新时间：2023-06-27 16:20:35

代码编辑器是用于自定义代码的 Code 编辑器，可进行 JavaScript 和 CSS 的原生代码开发。结合可视化开发和传统代码开发，增强了扩展性。如下图单击代码编辑器入口：



适用场景

使用代码编辑要求开发者有一定的 JavaScript 基础，它能帮助开发者在使用微搭低代码进行拖拽式开发时，根据业务需求完善应用的逻辑丰富度和复杂度。

基本功能

模块介绍

代码编辑器可以分为全局配置区、页面配置区和编译区，在页面配置区中包含了对应用中每个单独的页面的文件夹。

• 全局配置区

The screenshot shows the 'Global Configuration Area' interface. On the left is a tree view of configuration modules:

- 全局 (selected)
- theme
- lifecycle
- style
- common
- 首页(index)
- 页面 A(u_ye_mian_a) (selected)
- lifecycle
- style
- handler
- 页面 B(u_ye_mian_b)
- lifecycle
- style
- handler

The right side is a code editor window titled 'style/u_ye_mian_a'. It contains the following CSS-like code:

```
1 /**
2  * 可以使用theme中定义的 mixin 和 定义变量
3  */
4
5 .u_ye_mian_a-class-name {
6
7     // 使用theme中的变量
8     // color: @example-primary-color;
9
10    // 默认 px 会转换成 rem, 含大写字母 PX 和 Px 则不会转换
11    // margin-top: 20px;
12
13    // 使用theme中的混合 (mixins)
14    // .mixins();
15
16    // 可写嵌套样式
17    .u_ye_mian_a-nest-class {
18        // color: #fff;
19
20        // 使用theme中的混合 (mixins)
21        // .mixins(green);
22    }
23}
```

• 页面配置区

The screenshot shows the 'Page Configuration Area' interface. The tree view on the left is identical to the global configuration, with '全局' and '页面 A(u_ye_mian_a)' being the selected nodes.

The right side is a code editor window titled 'style/u_ye_mian_a'. The code is identical to the one shown in the global configuration editor:

```
1 /**
2  * 可以使用theme中定义的 mixin 和 定义变量
3  */
4
5 .u_ye_mian_a-class-name {
6
7     // 使用theme中的变量
8     // color: @example-primary-color;
9
10    // 默认 px 会转换成 rem, 含大写字母 PX 和 Px 则不会转换
11    // margin-top: 20px;
12
13    // 使用theme中的混合 (mixins)
14    // .mixins();
15
16    // 可写嵌套样式
17    .u_ye_mian_a-nest-class {
18        // color: #fff;
19
20        // 使用theme中的混合 (mixins)
21        // .mixins(green);
22    }
23}
```

• 编译区

The screenshot shows the code editor interface. On the left, there is a tree view of project structure under '编译区'. Under 'style' for '页面 A(u_ye_mian_a)', there is a file named 'style.u_ye_mian_a'. The code in this file is:

```
1 /**
2  * 可以使用theme中定义的 mixin 和 定义变量
3  */
4
5
6 .u_ye_mian_a-class-name {
7   // 使用theme中的变量
8   // color: example-primary-color;
9
10  // 默认 px 变换为 rem，舍弃字母 PX 和 Px 则不会转换
11  // margin-top: 20px;
12  // 使用theme中的混合 (mixins)
13  // .mixins();
14
15  // 司马懿宫样式
16  .u_ye_mian_a-nest-class {
17    // color: #fff;
18    // 使用theme中的混合 (mixins)
19    // .mixins(green);
20  }
21
22 }
23
```

模块	功能
handler	Handler 是页面的方法集合，若您需要在某页面下调用，则在该页面 Handler 下新建方法，即能够在页面中调用组件、变量和平台方法等，请参见 Handler 方法使用介绍 。
common	Common 是公共的方法集合，用于多个页面间的方法复用，详情请参见 Common 全局方法复用说明 。
style	全局配置区和每个页面都有 style 模块，在全局配置区的设置会作用到整个应用，单个页面的 style 只对页面内有效，详情请参见 Style 用法示例 。
lifecycle	应用和每个页面都有自己的 lifecycle，全局配置区的 lifecycle 在应用使用的时候会生效，而只有当某个页面在页面堆栈最上层时，才会触发该页面的 lifecycle，详情请参见 应用生命周期介绍 。

Style 用法示例

进入 style 页面，定义样式参数。

```
.button-font-size {  
    font-size: 30px;  
}  
.button-line-height {  
    line-height: 300px;  
}
```

添加一个组件，选中该组件后进行 classname 参数选择。

iPhone 8 100% 02:18 100%

按钮

样式编辑

classname

button-line-height
button-font-size

LifeCycle 生命周期介绍

微搭支持应用、页面级别的生命周期函数，当对应的生命周期函数触发时，便会自动执行函数下已配置的自定义方法。

应用生命周期

在代码编辑器中，选择全局文件夹下的 lifecycle 文件，即可对应用的生命周期进行管理。

The screenshot shows the Tencent Cloud Low-Code Platform's code editor interface. On the left, there's a sidebar with a tree view of project files. A red box highlights the 'lifecycle' file under the '全局' (Global) section. The main editor area displays the following code:

```

1  /**
2  * 可通过 app.生命周期函数获取全局应用的: 环境, 状态, 等信息
3  * 具体可以参考 console.info 在微信小程序控制台的输出了解更多信息
4  * 例如: 要使用 async-await, 需要先将 await 异步
5  */
6
7  export default {
8    onAppLaunch(launchOpts) {
9      //console.log('-----> LifeCycle onAppLaunch', launchOpts)
10 },
11 onAppShow(appShowOpts) {
12   //console.log('-----> LifeCycle onAppShow', appShowOpts)
13 },
14 onAppHide() {
15   //console.log('-----> LifeCycle onAppHide')
16 },
17 onAppError(options) {
18   //console.log('-----> LifeCycle onAppError', options)
19 },
20 onAppPageNotFound(options) {
21   //console.log('-----> LifeCycle onAppPageNotFound', options)
22 },
23 onAppUnhandledRejection(options) {
24   //console.log('-----> LifeCycle onAppUnhandledRejection', options)
25 }
}

```

A blue button at the bottom right of the editor says '保存 CTRL+S'.

函数名	说明
onAppLaunch	当应用初始化完成时触发（全局只触发一次）
onAppShow	监听小程序切前台事件。该事件与 App.onShow 的回调参数一致
onAppHide	监听小程序切后台事件。该事件与 App onHide 的回调时机一致
onAppError	小程序发生脚本错误或 API 调用报错时触发
onAppPageNotFound	小程序要打开的页面不存在时触发
onAppUnhandledRejection	小程序有未处理的 Promise 拒绝时触发

页面生命周期

在代码编辑器中，选择对应页面文件夹下的 lifecycle 文件，即可对该页面的生命周期进行管理。

函数名	说明
onPageLoad	页面加载时触发。一个页面只会调用一次，可以在 onLoad 的参数中获取打开当前页面路径中的参数
onPageShow	页面显示/切入前台时触发
onPageReady	页面初次渲染完成时触发。一个页面只会调用一次，代表页面已经准备妥当，可以和视图层进行交互
onPageHide	页面隐藏/切入后台时触发。例如 app.navigateTo 或底部 tab 切换到其他页面，小程序切入后台等
onPageUnload	页面卸载时触发。例如 app.redirectTo 或 app.navigateBack 到其他页面时触发

Common 全局方法使用介绍

在应用编辑器中打开代码编辑器，在全局 common 方法集合下我们可以添加一个可为全局调用的名称为 getTime 的方法。

The screenshot shows the application editor interface. On the left, there's a sidebar with project components like theme, recycle, style, common, and lifecycle. The 'common' component is selected and expanded, showing a file named 'getTime'. This file contains the following code:

```
1 //
2 // 通过全局注册，通过 app.common.name.ool 项目从全局引用方法或值
3 // 通过外部引用，通过 import 在外部的 handler 引用的例子：import { xxx } from './common/name'
4 //
5
6 export function getLocaleTime(){
7     //直接使用 new Date()
8     const date = new Date()
9     return date.toLocaleString()
10 }
11
12
```

At the bottom right of the editor, there is a 'Save' button labeled '保存 CTRL+S'.

全局方法的调用介绍

在代码中调用前面定义的全局函数，打开代码编辑器，在首页 lifecycle 中我们调用 getTime 方法来作为示例，把获取到的当前日期给变量（调用方法放到 lifecycle 的 onPageLoad 中，可以在页面加载时触发执行方法调用和变量赋值操作）。

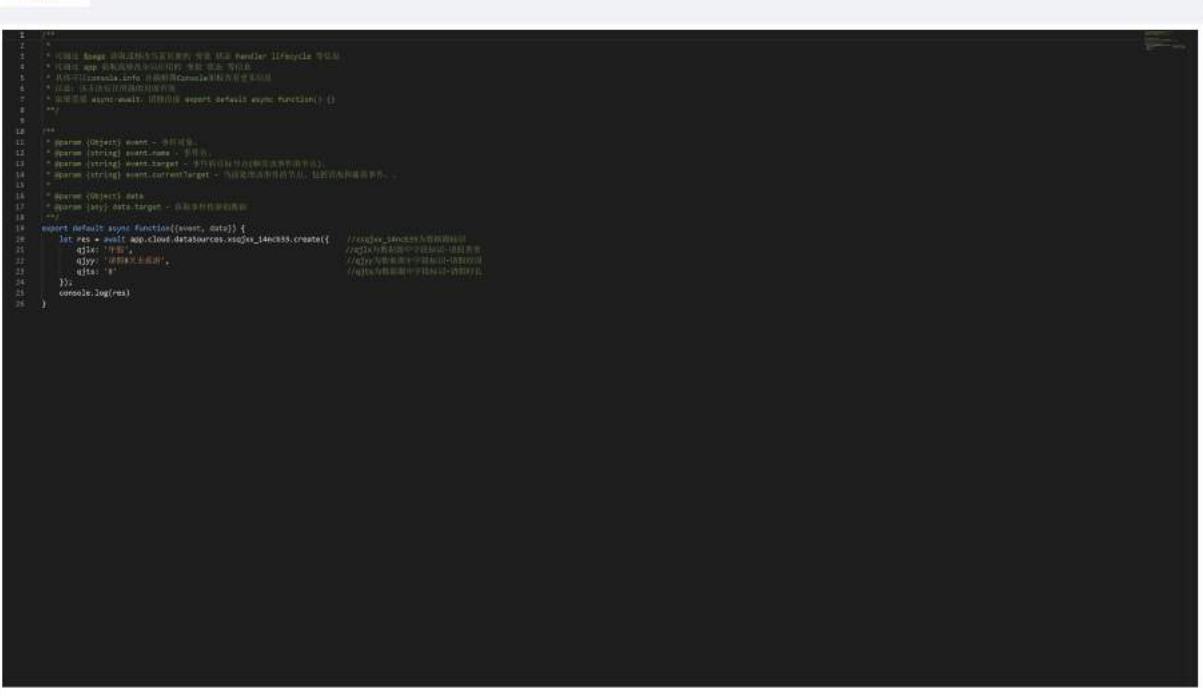
The screenshot shows the application editor interface. The sidebar now includes 'lifecycle' under the 'common' component. The 'lifecycle/index' file is open, containing the following code:

```
1 // 可通过 $page 调取及修改当前页面的 全局 handler:lifecycle 单例
2 // 可通过 app 作用域调取全局可用的 变量、方法 等信息
3 // 具体使用 common.time 在解析的	common 的时候更多取
4 // 例如直接 await $page.common.time
5
6 export default {
7     async onPageLoad(query) {
8         let res = await app.common.getTime.getLocaleTime()
9         $page.dataset.state.time = res
10        console.log(res)
11    },
12
13    onPageShow() {
14        //console.log('-----> Lifecycle onPageShow')
15    },
16    onPageReady() {
17        //console.log('-----> Lifecycle onPageReady')
18    },
19    onPageHide() {
20        //console.log('-----> Lifecycle onPageHide')
21    },
22    onPageUnload() {
23        //console.log('-----> Lifecycle onPageUnload')
24    },
25}
```

The code block from line 7 to 13 is highlighted with a red rectangle. At the bottom right of the editor, there is a 'Save' button labeled '保存 CTRL+S'.

Handler 方法使用介绍

在代码编辑器页面的 handler 中可以新增方法 method1，我们可以调用 [数据源相关接口](#) cloud.dataSources，通过 create 新增数据源记录。



```

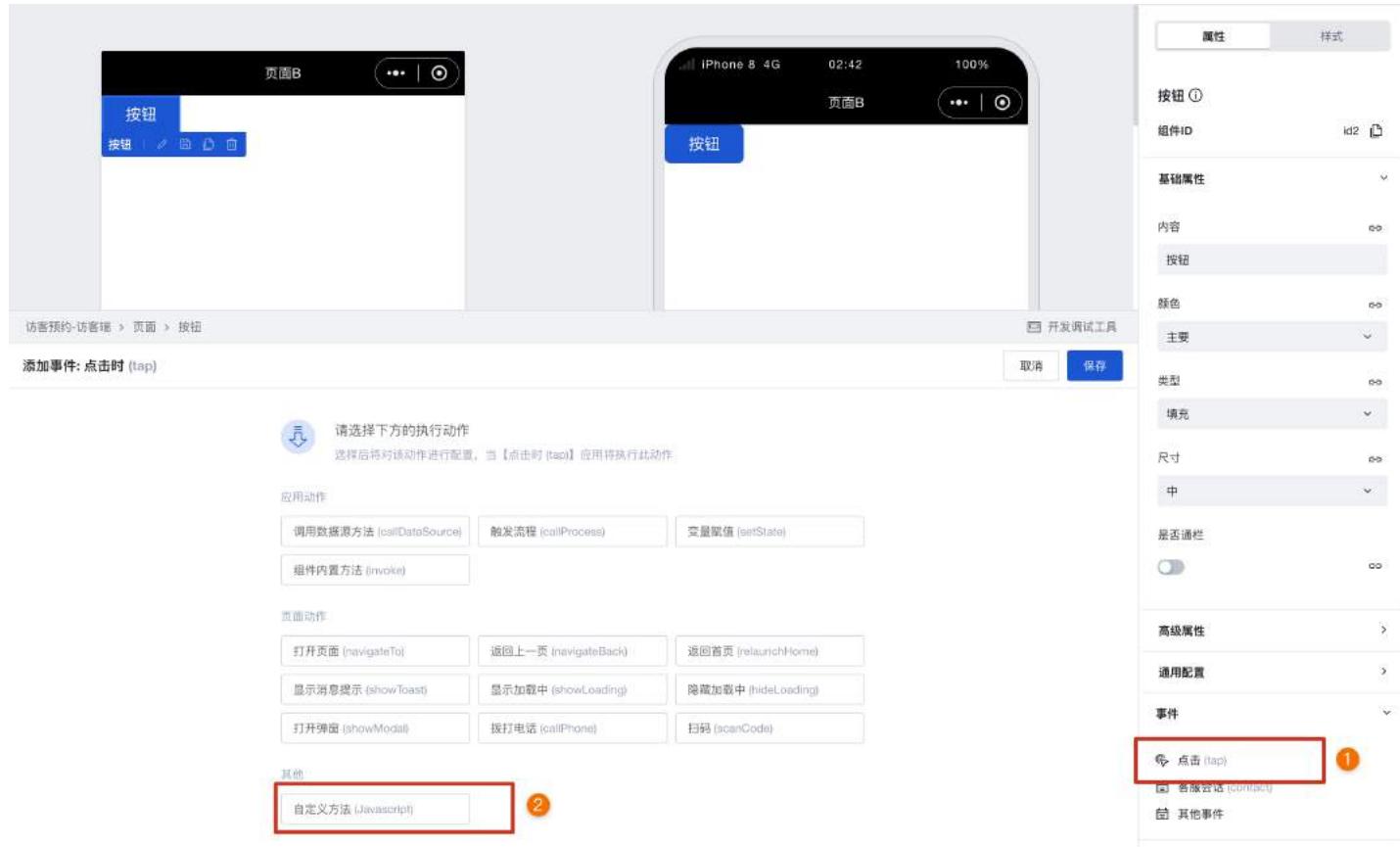
1 // ...
2 // 可以通过 logger 读取模块内部日志的变量, 这是 Handler 与 myHandler 区别
3 // 代码注释 app 宏将被忽略不会应用到 代码, 帮助 球拍点
4 // 通过云函数 console.info 打印到腾讯云控制台, 有更多帮助
5 // 代码注释会忽略的注释语句
6 // 代码注释会忽略的注释语句
7 // 代码注释会忽略的注释语句
8 // 代码注释会忽略的注释语句
9 // ...
10 // ...
11 // 变量 (Object) event - 事件对象
12 // 变量 (String) event.name - 事件名
13 // 变量 (String) event.target - 事件的目标对象(触发该事件的节点)
14 // 变量 (String) event.currentTarget - 当前处理该事件的节点, 包括冒泡和捕获事件
15 // ...
16 // 变量 (Object) data
17 // 变量 (Object) data.target - 该事件操作的节点
18 // ...
19 // 导入方法或事件函数(event, data) {
20 //   let res = await cloud.dataSources().create() // cloud.dataSources 为数据源对象
21 //   res.id // 生成的 id
22 //   res.name // 生成的名字
23 //   res.type // 生成的类型
24 // }
25 // console.log(res)
26 }

```

⚠ 注意

数据源新建及相关介绍请参见 [数据源与数据管理](#)。

返回应用编辑器页面，页面中新增一个按钮，按钮行为可配置点击时调用自定义方法。



属性

按钮 ①

组件ID: id2

基础属性

内容: 按钮

颜色: 主要

类型: 填充

尺寸: 中

是否通栏:

高级属性

通用配置

事件

- 点击 (tap) **②**
- 客服电话 (Contact)
- 其他事件

自定义方法可选择已建好页面 handler 方法 method1。

添加事件: 点击时 (tap)

取消

保存

当点击时 (tap) 时

u_ye_mianb:method1

成功时

失败时

Tips: 鼠标Hover上方条件可新增执行动作

方法类型

已有方法

添加新方法

执行动作

method1

入参

保存为

请插入变量名

管理自定义方法

外部资源（js/css）加载

最近更新时间：2023-06-29 15:52:35

微搭外部资源加载功能指在帮助开发者在应用中加载外部的 JSSDK 或 CSS 样式等资源，方便开发者快速引用第三方 SDK 并在自己的应用中使用，这样可以方便扩展前端应用的能力和样式风格。

⚠ 注意：

当前加载外部资源仅支持在 Web 端使用，暂不支持小程序。

操作步骤

步骤1：添加外部资源（JS/CSS）

- 首先，在编辑器的应用设置 > 开发设置页面中，单击添加外部资源。
- 在弹出的对话框中，输入要加载的资源的 URL。
- 单击添加，即可完成资源的添加。

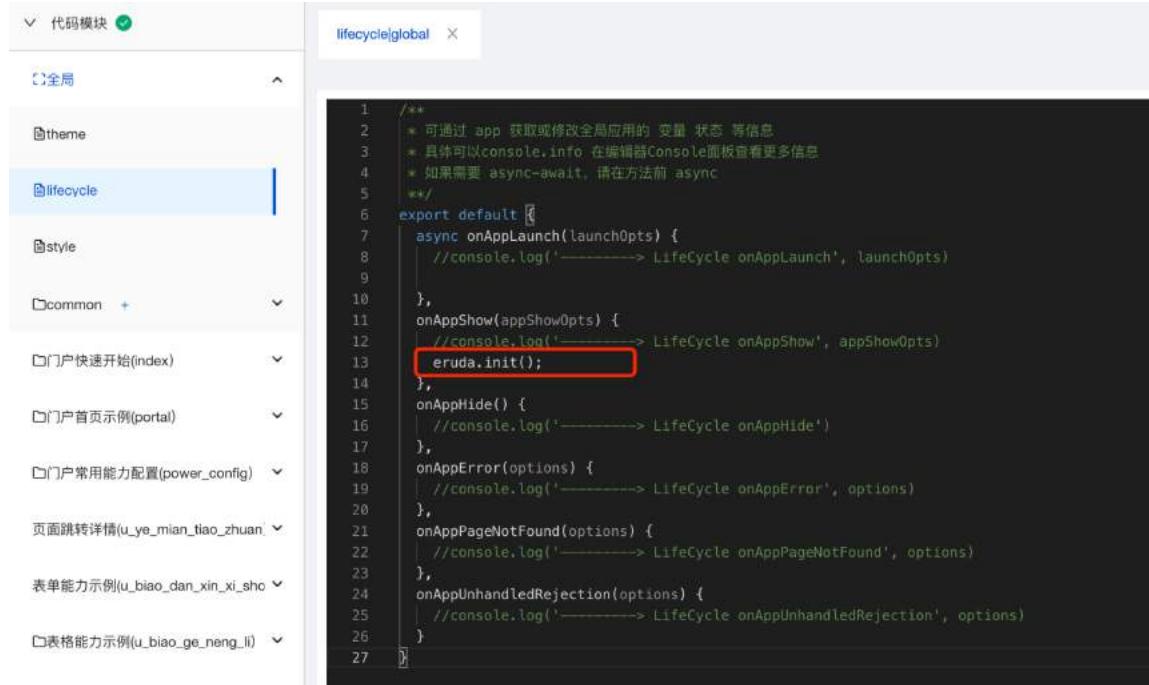


步骤2：引用外部资源并调用方法

外部资源添加成功后，会在应用初始化时默认加载该资源，故只需打开 [代码编辑器](#) 在应用生命周期或页面生命周期直接调用即可，如图所示：



进入 [代码编辑器](#)，在需要引用外部资源的位置输入以下代码（建议在应用周期的 `onAppShow() {...}`、页面生命周期的 `onPageShow() {...}` 应用方法中或之后的方法中进行调用即可）：



保存代码，运行预览应用，即可在应用中使用外部资源中的方法。

调用示例

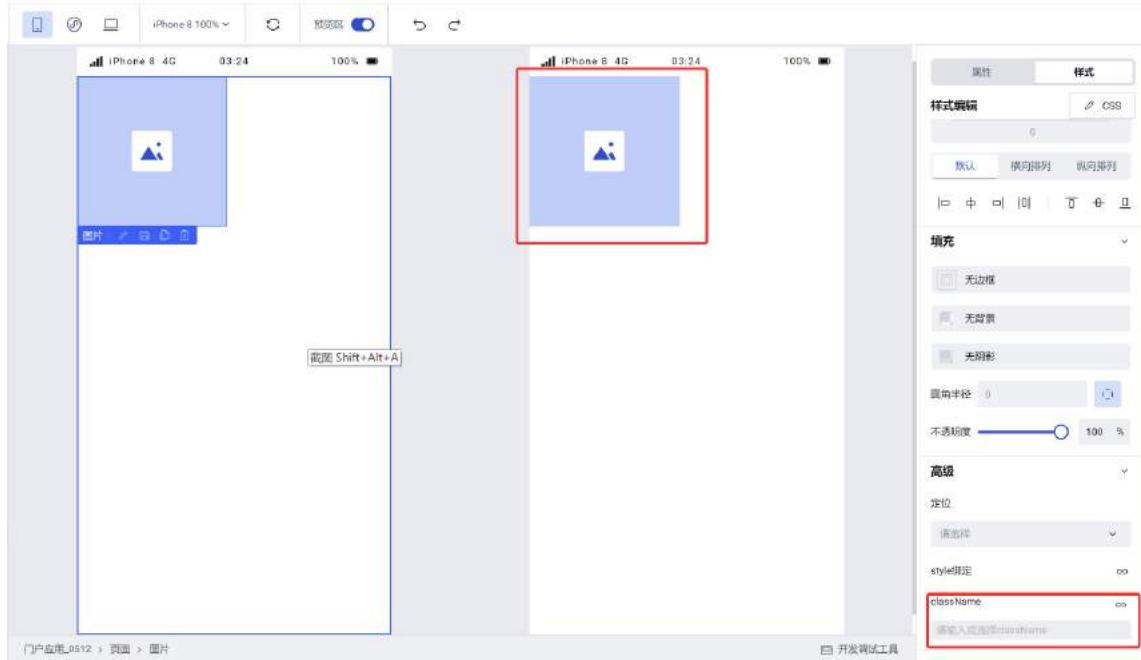
例如我们引用了第三方的 [Tailwind CSS](#)。

利用 Tailwind 提供的标准 Class 进行样式的配置：

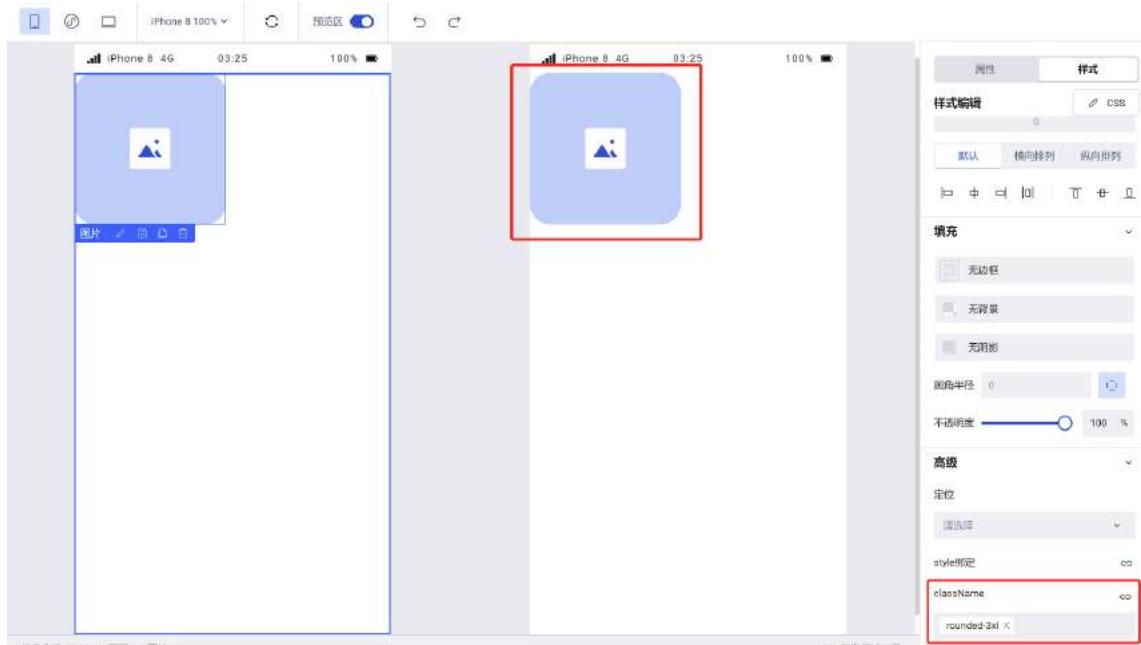
- 通过 `rounded` 控制边框圆角
- 通过 `ml` 控制元素左侧外间距
- 通过 `space-x` 控制子元素间距
- 通过 `font-bold` 控制字体的粗细

示例1

在页面中添加一个图片。

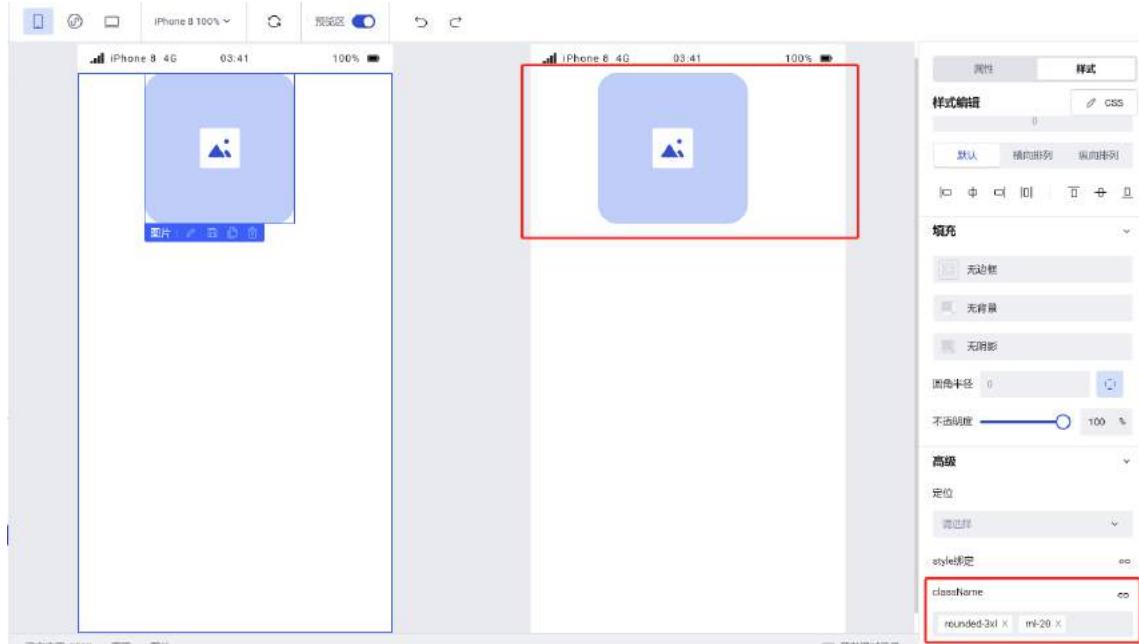


然后在图片的样式 > className 绑定 rounded-3xl，可以查看绑定之后图片圆角的变化。



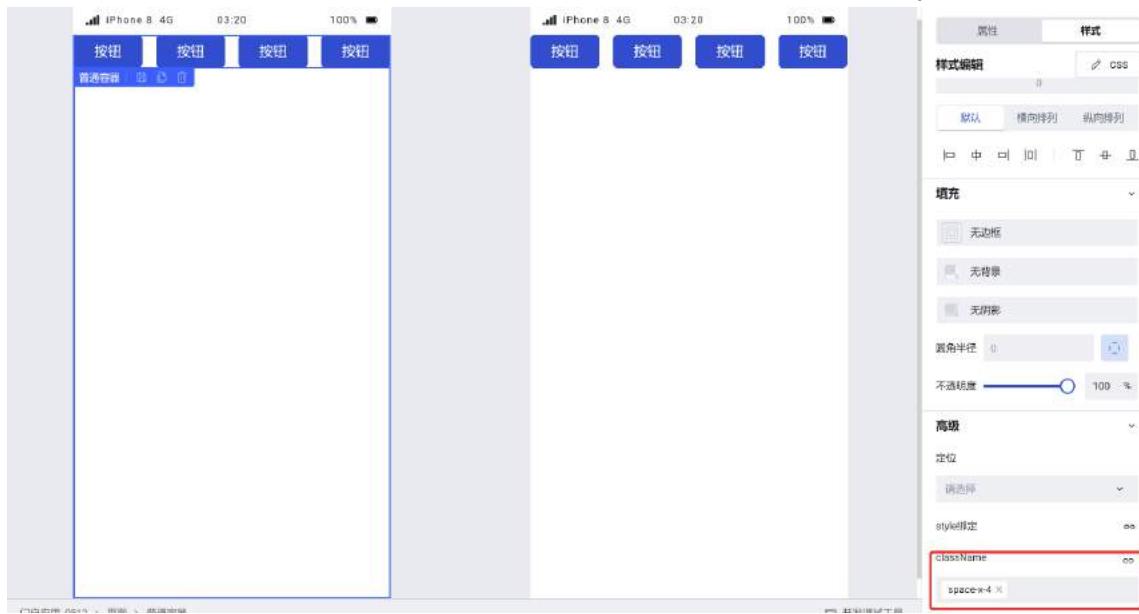
示例2

在示例1的图片组件的 classname 绑定 ml-20，可以查看绑定之后图片组件的左侧外边距的变化。



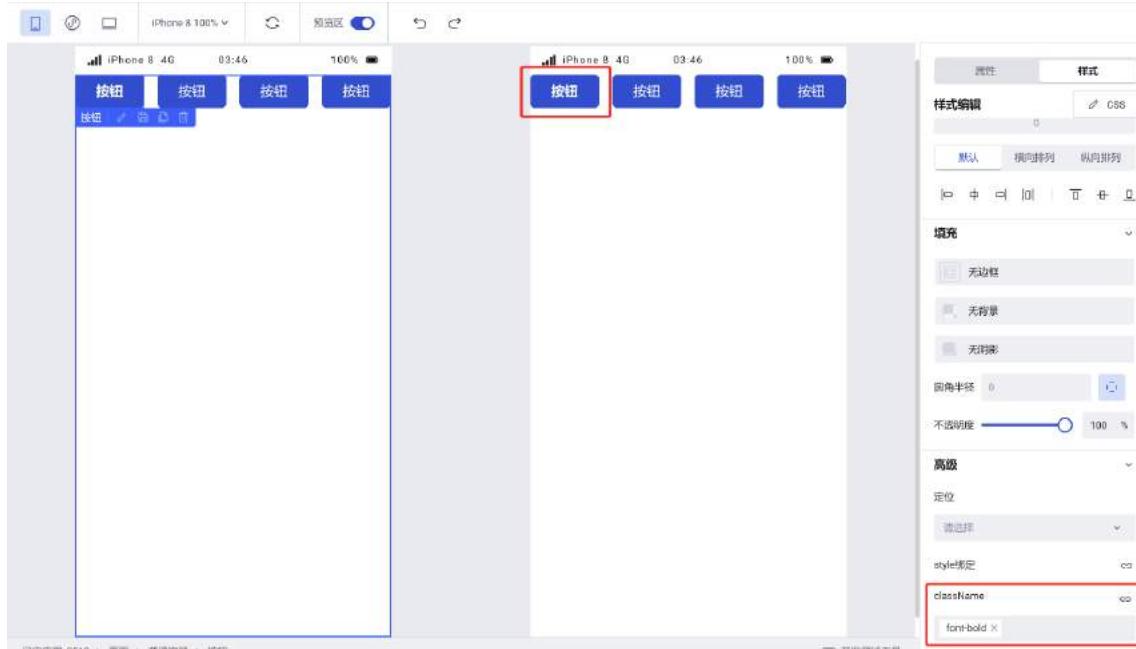
示例3

在页面中添加一个普通容器，普通容器中添加几个按钮。然后在普通容器的样式 > classname 绑定 space-x-4，可以查看绑定之后子元素间距的变化。



示例4

示例3中添加的第一个按钮组件，在其样式 > className 绑定 font-bold，可以查看绑定之后按钮组件的字体粗细变化。



注意事项

- 当前外部资源仅支持在 Web 端使用，暂不支持小程序。
- 请确保加载的资源 URL 是正确的，并且资源可以访问（某些资源是否支持跨域访问）。
- 如果使用的第三方资源，请遵循第三方资源提供商的使用规定，并确保不会侵犯任何第三方权益。
- 请确保您的应用与添加的外部资源兼容，以确保应用的正常运行。
- 请在开发过程中注意保护您的应用代码及数据安全，避免被第三方恶意访问。

组件

官方组件列表

最近更新时间：2023-07-18 14:23:45

本文会为您简单介绍微搭组件的使用方法，您可以通过单击组件的[查看文档](#)，查看该组件的具体使用方式。

若需了解获取组件属性值相关内容，请参见[获取组件属性值](#)。

数据容器

组件名称	组件功能	详细说明
数据列表	适用于从数据源中查询多条数据，并将返回列表循环展示的场景，如查询用户列表。	查看文档
数据详情	适用于从数据源中查询一条数据，并将返回数据展示的场景，如查询访客预约详情。	查看文档
表单容器	快捷实现表单数据的提交与展示场景。绑定数据源后，可根据字段类型自动生成单行输入、下拉选择、提交按钮等组件。	查看文档
数据表格	以表格形式展示多行数据，支持对数据进行排序、搜索、分页、自定义显示内容和操作。	查看文档

布局

组件名称	组件功能	详细说明
普通容器	可放入多个组件，常用于对组件布局进行管理控制。	查看文档
网格布局	网格布局（Grid）是最强大的页面布局方案，可以指定容器内部多个组件的位置。它将页面划分成一个个网格，可以任意组合不同的网格，做出各种各样的布局。	查看文档
分组容器	用于实现常见的内容分组展示效果。	查看文档
轮播容器	用于快速在页面搭建轮播图展示效果，并支持轮播图的跳转、播放配置。	查看文档
布局组件	用于构建带有顶部导航的页面布局。	查看文档
滚动容器	当组件元素内容长度大于滚动容器时，滚动容器会出滚动条用来对页面内容进行滑动展示。	查看文档
页头容器	用于页面顶部固定显示所需的标题或导航内容。	查看文档
页脚容器	用于页面底部固定显示所需的标题或导航内容。	查看文档
弹窗	用于实现基础的弹窗（对话框）展示效果。	查看文档

导航

组件名称	组件功能	详细说明
Tab 栏	用于实现页面的 Tab 栏导航效果。	查看文档
菜单导航	用于构建企业和门户官网应用下的「横向或纵向」菜单导航样式。	查看文档
宫格导航	用于宫格布局样式下的导航项目展示。	查看文档
顶部导航	用于页面顶部固定显示所需的标题或导航内容。	查看文档
顶部选项卡	用于提供顶部的选项卡布局，支持切换多个选项标签。	查看文档
侧边选项卡	用于提供侧边的选项卡布局，支持切换多个选项标签。	查看文档

— —

展示

组件名称	组件功能	详细说明
文本	用于展示页面中的文本或标题的内容。	查看文档
图片	用于在页面进行图片的展示。	查看文档
按钮	用于需要用户单击触发特定事件的场景，例如信息提交、页面跳转等。	查看文档
分割线	为页面添加一条分割线，常用于组件之间的布局区分。	查看文档
轮播图	用于快速搭建轮播图展示效果，并支持配置图片的单击跳转、播放间隔等功能。	查看文档
图文卡片	以卡片形式展示包含图片、标题、描述信息在内的列表信息。	查看文档
图文展示项	用于构建图片+标题文字的内容展示效果。	查看文档
链接	用于实现超链接单击后跳转。	查看文档
富文本展示	以卡片形式展示包含图片、标题、描述信息在内的列表信息。	查看文档
图标	可以通过在图标组件的配置区中选择预置图标实现图标的展示效果，同时图标组件也支持用户对图标进行自定义的上传。	查看文档
视频播放	用于展示播放视频资源。	查看文档
九宫格抽奖	用于实现基础的九宫格抽奖活动。	查看文档
日历	用于日历数据的平铺展示，例如考勤数据展示、日程展示。	查看文档
气泡提示	用于显示气泡的提示。	查看文档
网页嵌套	网页嵌套组件用于在应用中嵌套展示其他的网页。	查看文档
二维码	根据您配置的链接或文本等，生成二维码，并展示出来。	查看文档
卡片	卡片容器，可承载文字、按钮、图片等。	查看文档
树	以树形结构展示多层级的数据内容，例如组织架构、物料信息、客户分类等。	查看文档
循环展示	用于循环展示组件的场景，可将一个数组循环展示在页面中，例如循环一个新闻列表。	查看文档

表单

组件名称	组件功能	详细说明
单行输入	用于输入单行的文字内容。	查看文档
多行输入	用于输入多行的文字内容。	查看文档
数字输入	用于数字内容的输入。	查看文档
下拉选择	以列表形式展示选项，用户可进行搜索和选择操作。	查看文档
下拉多选	以列表形式展示选项，用户可进行多选操作。	查看文档
单选	页面中平铺选项，用户可进行单选操作。	查看文档
多选	页面中平铺选项，用户可进行多选操作。	查看文档
日期时间选择	用于选择日期时间，支持年月日时分秒、年月日等选择模式。	查看文档
时间选择	用于选择时间，支持时分选择模式。	查看文档
图片上传	用于上传保存本地图片。	查看文档

组件名称	组件功能	详细说明
文件上传	用于上传保存本地文件。	查看文档
开关	用于控制布尔值的是/否。	查看文档
成员选择	用于选择组织架构中的用户成员。	查看文档
部门选择	用于选择组织架构中的各级部门。	查看文档
地图定位	用于在表单中进行地图选点定位。	查看文档
地区选择	用于选择各级行政区划（省市区）。	查看文档
富文本编辑	可调整文本样式，支持插入图片、链接等内容的富文本表单组件。	查看文档
电话输入	用于输入电话号码。	查看文档
URL 输入	用于输入网址链接 URL。	查看文档
邮箱输入	用于输入邮箱地址。	查看文档
明细	用于多行明细数据的录入和展示，例如订单明细、报销明细等。	查看文档

图表

组件名称	组件功能	详细说明
统计卡片	提供报表场景下的指标数据统计并展示功能。	查看文档
折线图	用于报表场景下展示数据随时间变化产生的趋势。	查看文档
柱状图	用于报表场景下展示数据之间大小的对比。	查看文档
饼图	用于报表场景下表示数据在总体中的占比。	查看文档
通用图表	通用图表是一个支持多类型切换的图表，可配置出多种图表如折线图、饼状图、面积图、地图、漏斗图、仪表盘等。使用该通用图表组件，可实现快速引入 Echarts 图表。	查看文档

小程序

组件名称	组件功能	详细说明
获取用户手机号	在小程序运行态对用户的手机号进行获取。	查看文档
小程序分享	支持快捷实现小程序分享功能。	查看文档
公众号关注	您可在小程序内配置公众号关注组件，方便用户快捷关注公众号。	查看文档
客服会话	小程序端，单击客服会话组件即可直接进入小程序客服会话界面。客服会话功能是与小程序终端用户沟通建联的最便捷入口之一。只用配置一个组件，即可在 3 分钟内完成客服系统的功能闭环，实现用户-客服人员的双向沟通。	查看文档

插槽说明

组件的插槽是附属在单个组件内的节点（区别于容器组件，容器组件本身即为一个独立节点）。插槽在不同的组件中有不同的使用方式：

- 在选项卡组件中，插槽用来存放每一个 Tab 项中的内容。
- 在列表项组件中，插槽可用来存放列表图标和标题文本等。

组件数据

获取组件属性值

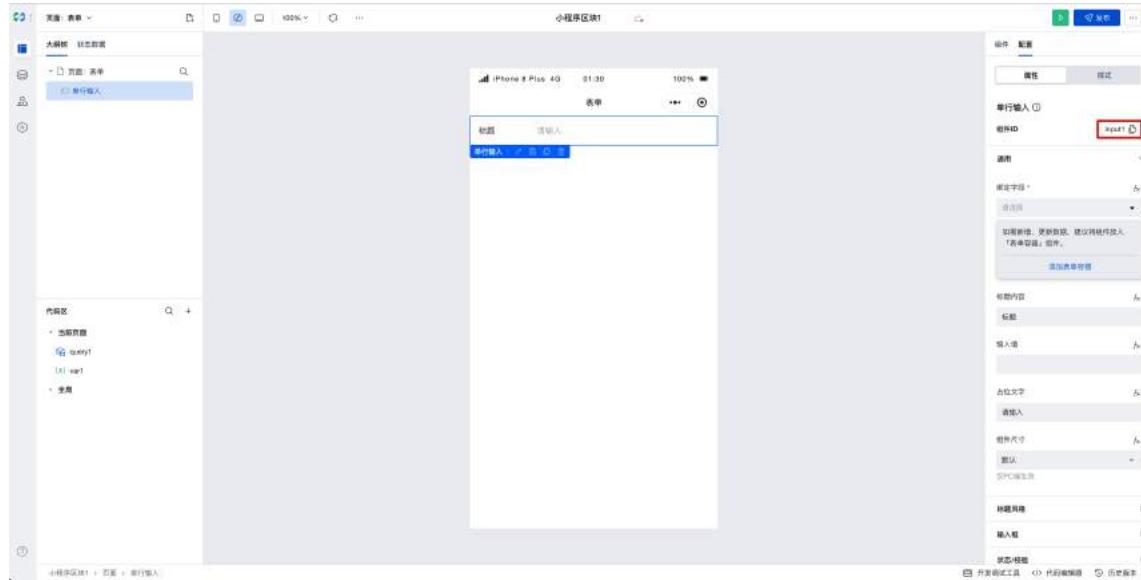
最近更新时间：2023-08-09 10:50:35

在应用构建过程中，需要快速获取组件的某些属性值直接使用，例如，获取单行输入组件的输入值，下拉选择的选项值，表格组件指定行的某个字段值，图片组件的src地址，按钮组件的文本内容等。下面提供获取组件属性值的方法供参考。

对于微搭组件，可以通过\$w.id.xx 获取组件的属性值，下面以单行输入、下拉选择、图片组件为例做介绍：

单行输入组件

想要把单行输入组件的输入值给文本进行内容展示，添加一个单行输入组件，复制单行输入组件的ID，添加一个文本组件。

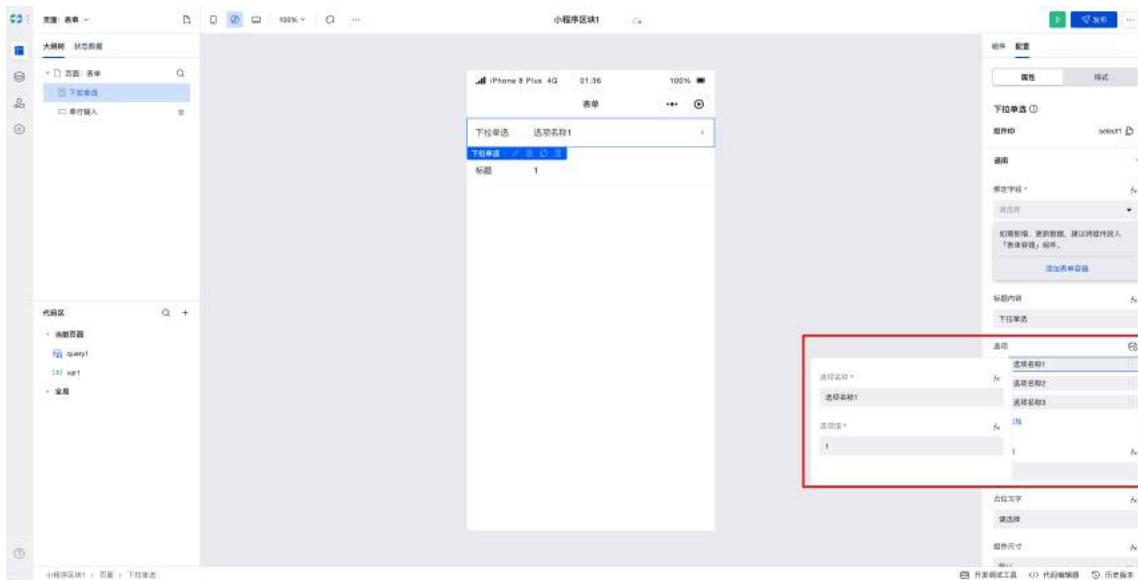


文本内容绑定表达式 \$w.input1.value（其中 input1 为单行输入组件的组件 ID，value 为要获取的 input 值属性名称）。在单行输入中输入内容，单行输入的输入值可以显示在文本中。



下拉选择组件

想要获取下拉选择组件的选项值，赋值给单行输入组件，可以在编辑器中添加下拉选择和单行输入组件，赋值下拉选择组件的 ID。首先可以配置选项名称和对应的选项值。

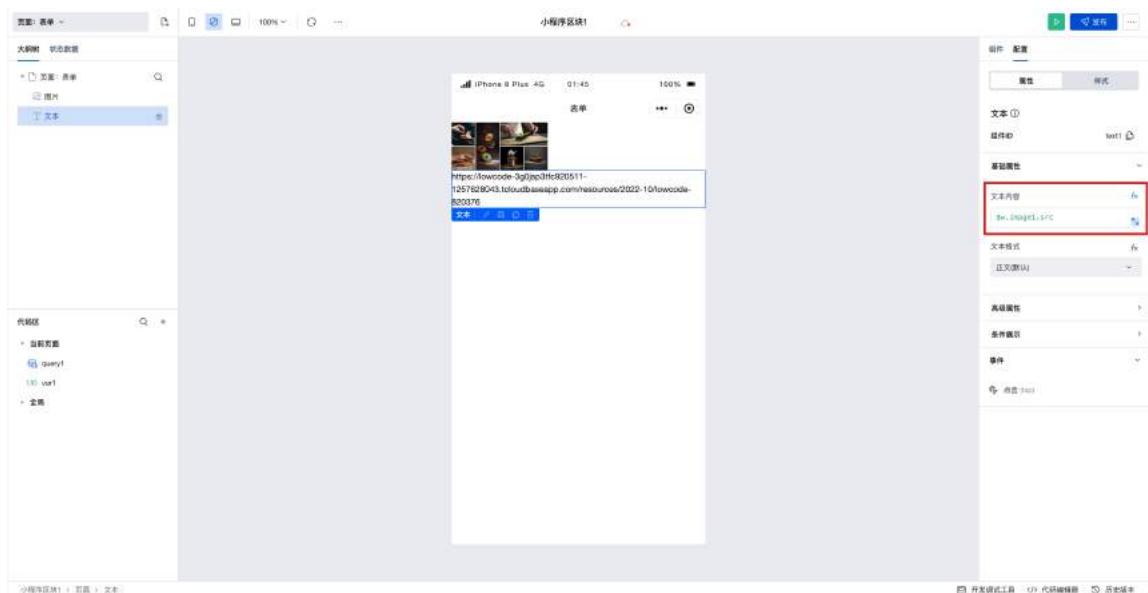


单行输入组件的输入值绑定 \$w.select1.value（其中 select1 为下拉选择组件的组件 ID，value 为要获取的选项值属性名称）。下拉选择中选择一个选项，可以看到单行输入中显示了选项对应的选项值的内容。



图片组件

想要把图片组件的图片地址取得后显示在文本组件中，可以添加一个图片和一个文本组件，文本组件的文本内容绑定 \$w.image1.src（其中 image1 为图片组件的组件 ID，src 为要获取的图片地址属性名称）。



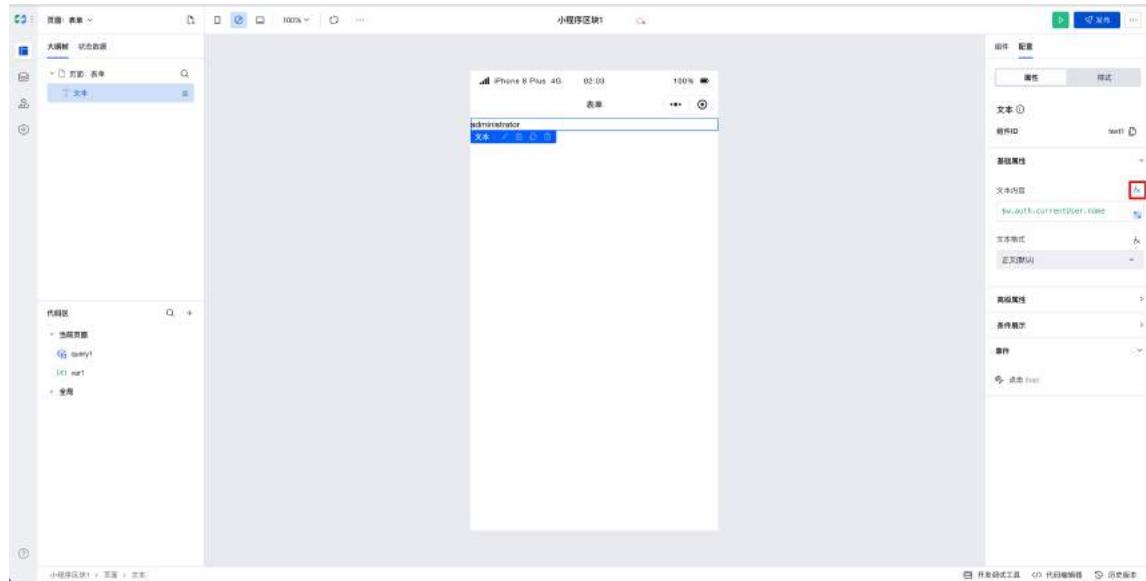
组件绑定数据

最近更新时间：2023-08-09 10:50:42

本文介绍组件如何绑定数据，实现组件属性的动态控制。

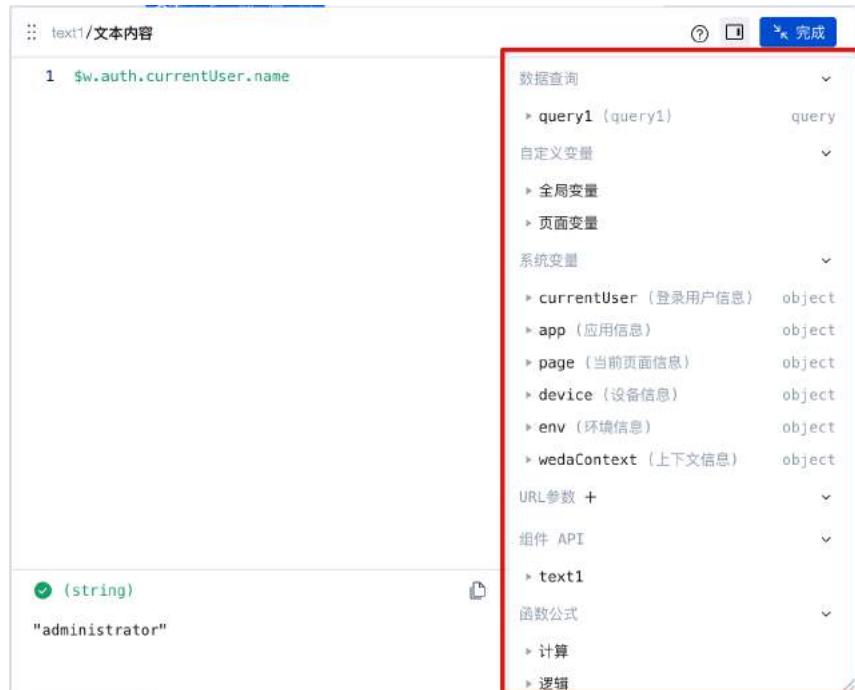
功能介绍

组件的属性上，若有 fx 表达式图标，则该属性可绑定变量等动态数据。单击 fx 可输入或选择数据。例如输入 currentUser，可获取当前登录用户的信息。詳请参见 [表达式使用说明](#)。



组件可使用的数据来源和类型

组件绑定的数据来源有以下几种：组件属性值、其它页面传入的 URL 参数、Query 数据查询、自定义变量、系统变量。通常可以在表达式的右侧面板中可见，如下图：



详细的数据来源有下列几种，可见具体的使用文档：

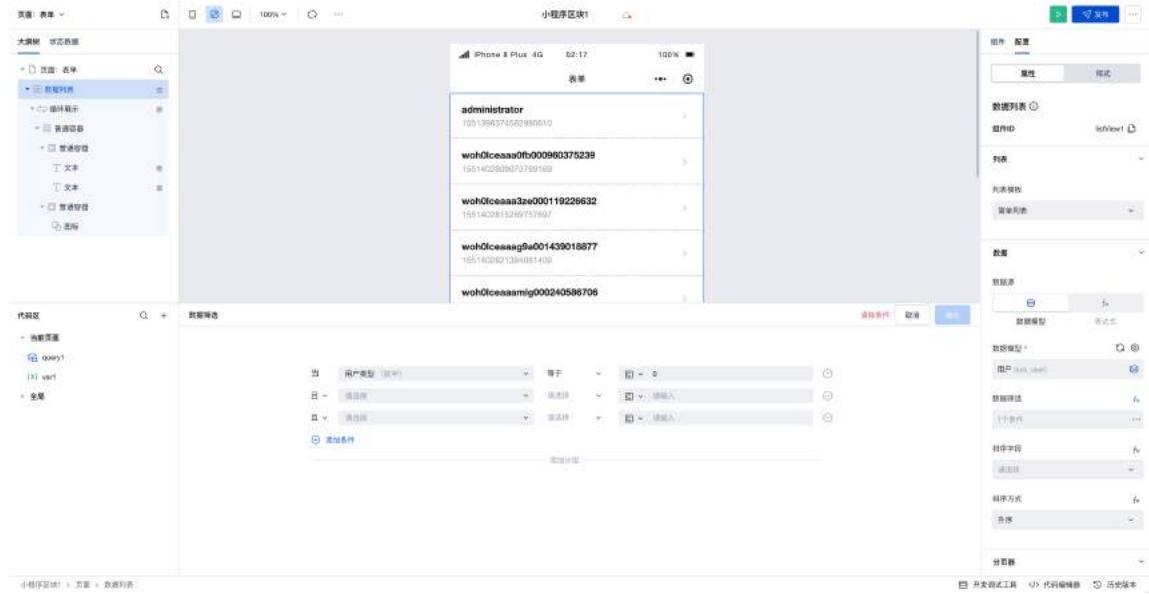
- 获取其他组件的属性，请参见 [获取组件属性值](#)。

- 通过绑定页面 URL 参数实现动态控制，请参见 [页面跳转传参配置](#)。
- 使用变量，请参见 [自定义变量使用说明](#)。系统变量包含当前登录用户、应用、页面、设备、环境等信息。
- 使用数据库查询的数据，请参见 [Query 数据查询](#)

数据筛选条件

最近更新时间：2023-08-09 10:50:52

微搭平台在组件内多处使用了数据筛选条件，主要提供微搭平台下的数据按照固定条件进行筛选功能。



筛选条件说明

数据容器组件内支持设置筛选条件，实现对于数据容器数据源的过滤。筛选条件主要由 KEY 值、计算方式、value 类型、value 值和条件间关系组成。

筛选条件	筛选说明
条件 KEY 值	支持选择本数据源的字段类型。允许进行筛选的字段类型包括：文本、数字、布尔值、日期时间、邮箱、电话、网址、枚举。
计算方式	支持等于、不等于、大于、小于、大于等于、小于等于、包含、不包含、不同类型的字段所支持的计算方式不同，详情可在编辑器内体验。
value 类型	支持常量和变量两种类型： <ul style="list-style-type: none">当选择常量时，允许输入传递参数，具体参数规范可参见 数据模型字段。当选择变量时，允许通过变量传递参数，具体可以参见 变量使用说明。

条件之间的关系

- 支持且关系。
- 多个条件可以使用条件关系进行逻辑运算。
- 条件关系支持分组，同一个分组内，仅允许存在一种条件关系。

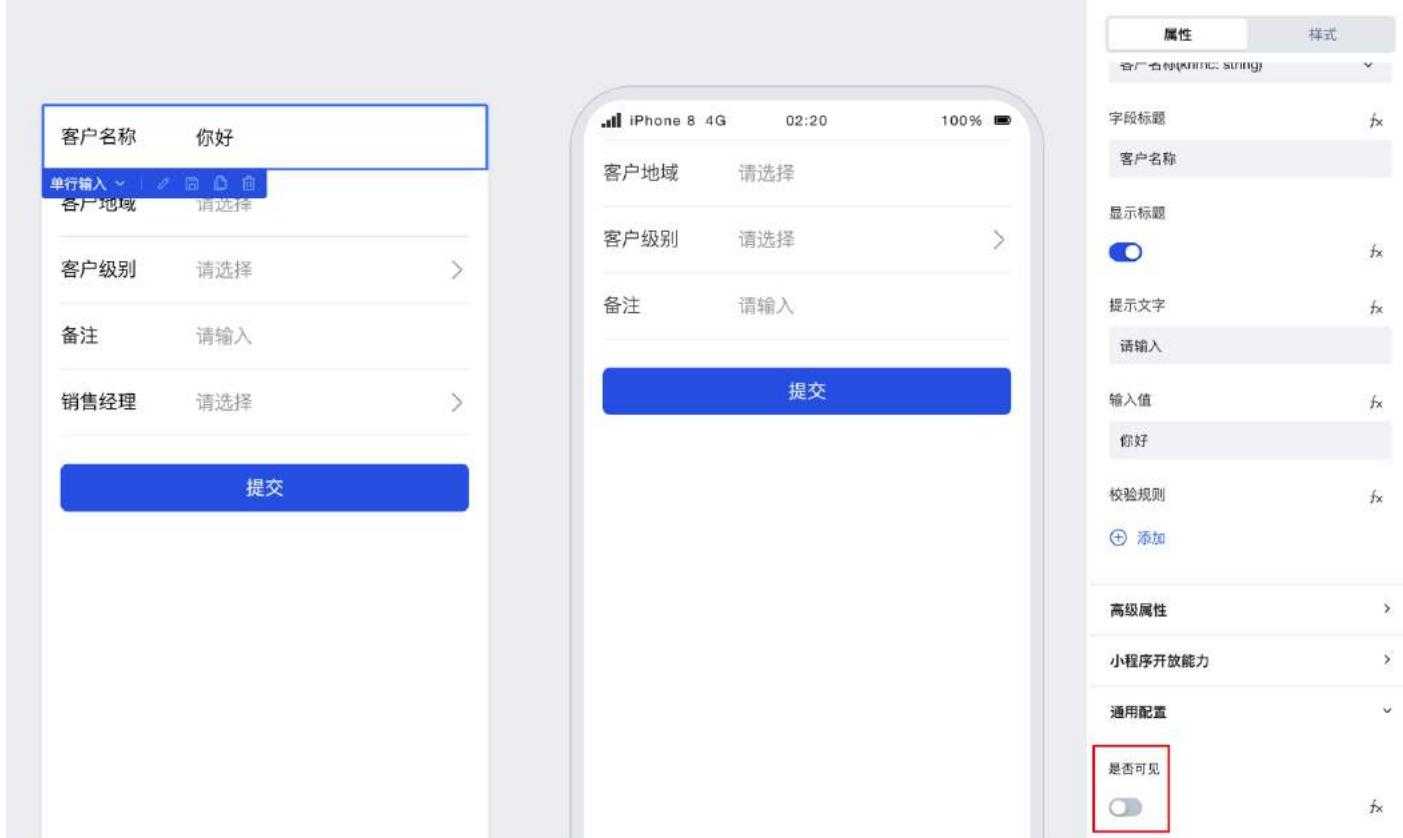
詳情请参见 [数据模型的简单与复杂查询](#)。

组件动态显隐

最近更新时间：2023-08-14 15:25:39

可见属性控制组件显隐

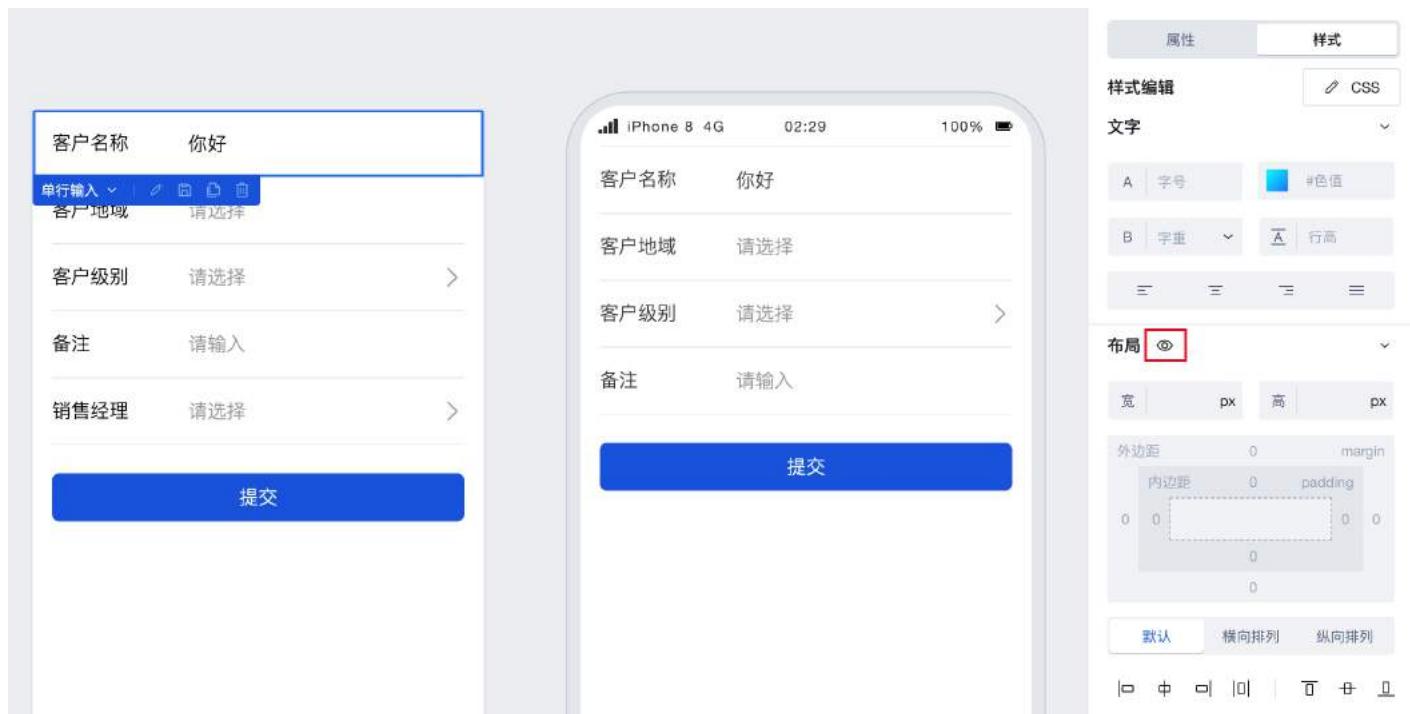
1. 所有组件均提供是否可见属性，用于控制组件的显示或隐藏状态，该属性可绑定表达式，实现动态的显隐控制，具体实践案例，请参见 [单选组件控制其他组件的显隐](#)。



2. 组件的是否可见属性关闭时。

- 组件将会隐藏不可见，同时在页面中不再占位。
- 组件节点被实际加载，组件的属性 API 和方法 API 均可正常生效。
- 表单组件的必填、正则等校验，均为触发（如期望不触发校验，则请使用 [是否渲染属性](#)），同时表单值可正常提交入库，具体实践案例，请参见 [表单组件隐藏时，提交对应表单数据](#)。

3. 在组件的样式配置面板中，也提供了隐藏设置。单击该设置项，也可隐藏组件。该功能通过直接为组件添加 display:none 的 style 样式来隐藏组件，但是此处无法绑定表达式，实现动态显隐效果，故优先推荐使用；是否可见属性来动态显隐组件。



4. 组件的隐藏效果会优先生效。

- 即满足以下任意条件时，组件就会被隐藏。
 - 是否可见属性处于关闭状态。
 - 布局 > 隐藏设置被开启。
- 如您发现开启是否可见属性后，组件却并不可见，请优先检查布局 > 隐藏设置是否开启。

是否渲染属性说明

1. 当您通过是否可见属性隐藏较多组件后，发现页面加载较慢，可尝试使用是否渲染属性来禁止渲染组件节点来提高加载性能，具体操作方式如下：

- i. 在应用设置 > 实验室菜单下，增加组件渲染控制配置项。开启后，组件属性中会展示是否渲染属性（即原有的条件展示属性），并默认开启。

ii. 在组件属性面板中，找到并关闭是否渲染属性后，将不会在页面中渲染并展示该组件节点。

The screenshot shows the Tencent Cloud Low-Code Platform's design interface. On the left is the component library with categories like 'Micro搭组件库' (Micro搭 Components Library), '热门' (Hot), '数据列表' (Data List), '普通容器' (General Container), '图片' (Image), '链接' (Link), '数据容器' (Data Container), '表单容器' (Form Container), and '布局' (Layout). A search bar at the top says '请输入内容'. The main area displays a form with fields: '客户地域' (Customer Region) with placeholder '请选择', '客户级别' (Customer Level) with placeholder '请选择', '备注' (Remarks) with placeholder '请输入', and '销售经理' (Sales Manager) with placeholder '请选择'. A large blue '提交' (Submit) button is at the bottom. To the right is the 'Properties' panel. Under '是否渲染' (Is Rendered), the switch is off, and a note says '原条件展示属性，关闭时，组件节点将不会渲染加载' (Original condition display property, when turned off, the component node will not be rendered or loaded). Other properties shown include '绑定字段' (Bound Field), '字段标题' (Field Title), '显示标题' (Show Title), '提示文字' (Prompt Text), '输入值' (Input Value), and '校验规则' (Validation Rules).

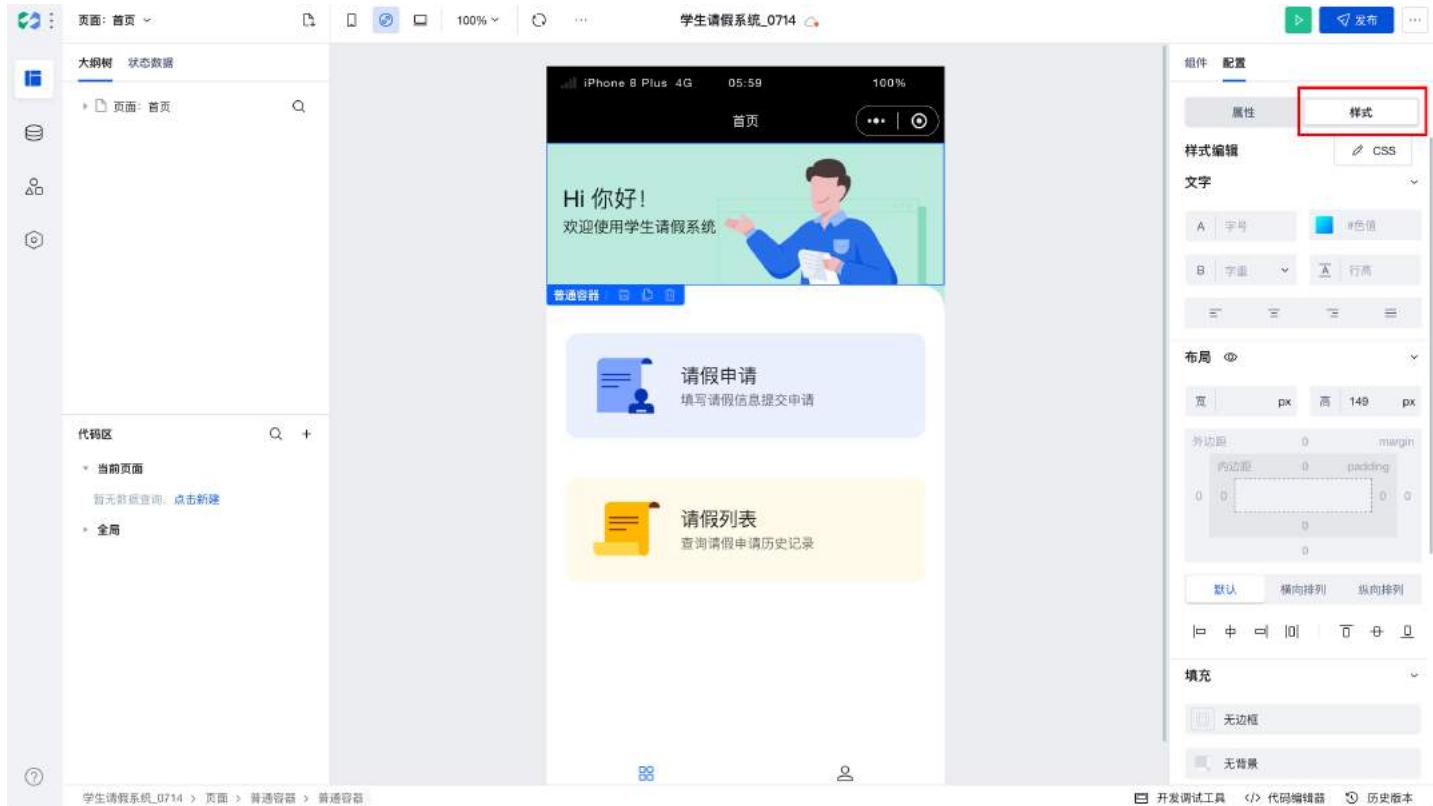
2. 对比组件的是否可见属性，通过是否渲染属性隐藏组件，页面加载性能更快，同时表单组件的必填、正则等校验，均不会触发，且表单值不会提交入库；但因为组件实际并没有加载渲染，所以无法通过属性 API 和方法 API 进行组件值的获取或赋值。

3. 是否可见属性和是否渲染属性有任一属性被关闭时，组件就会处于隐藏状态。

组件样式配置

最近更新时间：2023-07-18 14:54:49

微搭提供组件样式快捷编辑功能，在 [微搭应用编辑器](#) 中选中组件之后，在编辑器的右侧可以看到属性配置区域和样式配置区域，单击样式页签，可进行组件样式的个性化配置，基础的样式配置包含文字、布局、填充、高级，支持通过样式代码编辑器和自定义两种方式灵活的对样式进行配置。以下对样式配置功能做详细介绍说明：



文字

可调整文字的字号、字色、字重、行高，对齐方式。



- **字号：**可以手动输入字号调整文字大小。
- **字色：**可以单击选择文字颜色。
- **字重：**字重分为粗、正常、细三种。
- **行高：**可以手动输入数字调整文字所在行高度。
- **对齐：**文字的对齐方式（左对齐，居中，右对齐，两端对齐）。

② 说明：

当文字只有一行时，样式代码编辑器中 `text-align` 修改为 `text-align-last` 两端对齐生效。

布局

通过布局可以调整组件的宽高尺寸和内外边距，可以配置容器中多个组件的排列和对齐方式，可以控制组件的隐藏布局。

布局

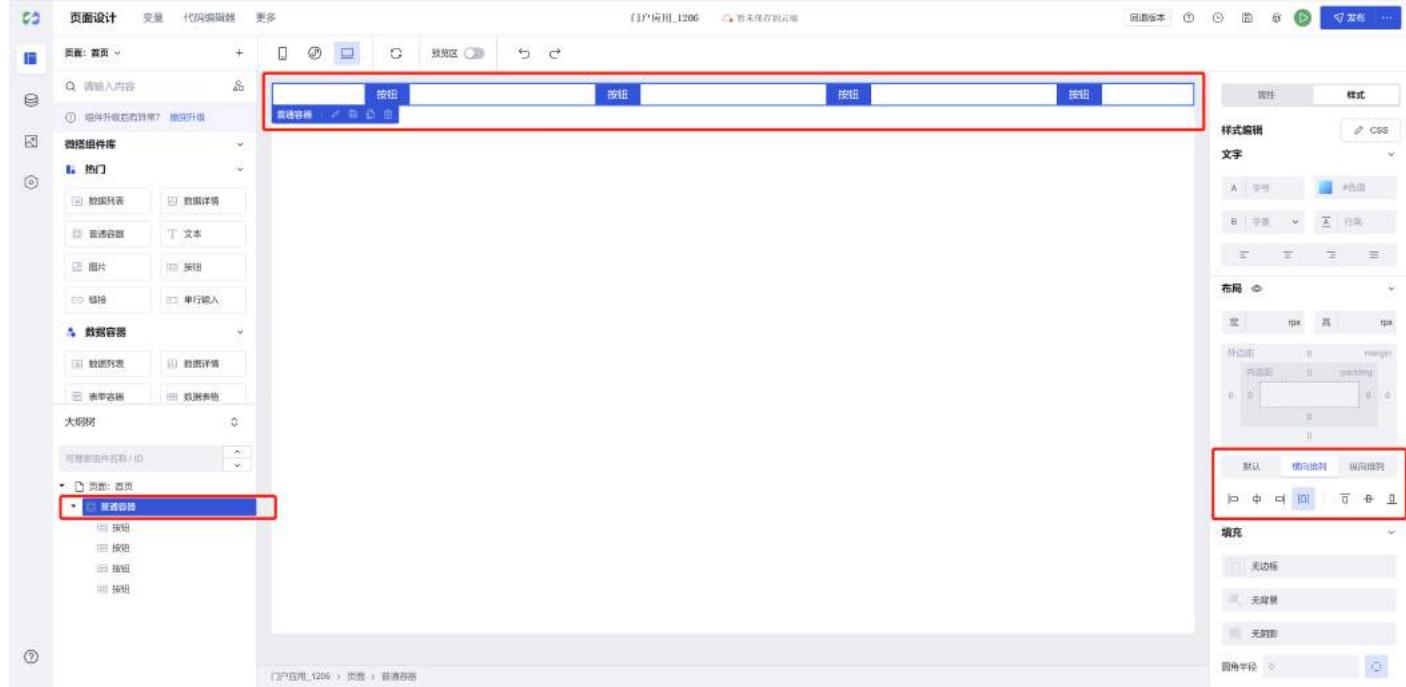


- **宽高尺寸：**可设置组件宽度和组件高度，目前提供 px、rpx、% 和 rem 下拉可选单位。
- **内外边距：**支持输入数值设置组件上下左右的外边距（margin）和内边距（padding）。
- **隐藏布局：**单击下图的图标可设置组件在编辑器中隐藏布局（display: none），再次单击该图标可取消隐藏布局。



- **布局模式：**布局模式分为排列和对齐。排列可选择默认、横向排列、纵向排列；对齐可选择：左对齐、水平居中、右对齐、平分、上对齐、垂直居中、下对齐。布局模式一般不针对单个组件设定，多用于一个容器中有多个组件的场景，例如一个普通容器中我们放了多个按钮组件，想要实现按钮在容器中水平平

分，我们可以选中普通容器，设置横向排列、平分，效果如下图：



② 说明：

所有组件初始化拖进编辑器中我们都会选中默认布局，当选择了某个布局模式之后，想要恢复默认布局，只需要再次点击您的选中项目，即可取消选中。

填充

填充包含边框配置、背景配置、阴影配置、圆角配置、透明度配置。

填充



- 边框：**默认无边框，支持配置边框颜色、边框类型（实线或虚线）、边框宽度，单击下图中边框关联配置图标可取消边框宽度关联，可以单独配置任何一个边框的宽度（系统会默认关联，即配置一条边框宽度后适用于所有边框宽度）。

边框



- 背景:** 默认无背景, 可选择颜色背景和图片背景。可以选择指定某一种颜色作为背景, 也可以选择特定的图片作为背景。
- 阴影:** 默认无阴影, 可配置组件的外阴影, X 表示水平阴影位置 (数值可以为负值), Y 表示垂直阴影的位置 (数值可以为负值), 扩展表示阴影的尺寸值, 距离表示阴影的模糊距离, 可以灵活调整各项数值配置合适的组件阴影。

- 圆角:** 支持配置圆角的半径, 输入的数值越大, 圆角效果越明显, 和边框类似, 圆角支持统一配置四个圆角, 支持配置某一个独立圆角。
- 不透明度:** 配置组件显示的透明度, 100%为完全不透明, 0%为完全透明。

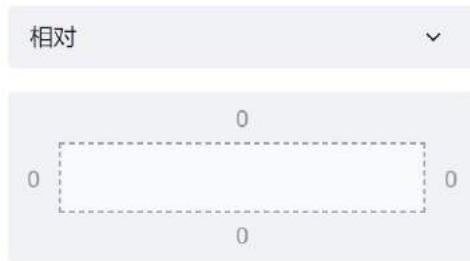
高级

高级-定位

定位: 配置多个组件在页面中的相对和绝对位置 (层叠显示或不层叠显示), 把某个组件固定在页面中、页面滚动组件固定不动、或者有多个重叠组件想让某些组件优先展示, 可通过定位来控制。配置了相对、绝对、固定定位之后, 支持配置组件基于当前定位的四边边距。微搭低代码应用编辑器提供以下四种定位方式:

高级

定位

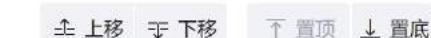


层级



- 默认:** static，静态定位，组件按照 Normal Flow（标准流、文档流、常规流、正常流）进行排布，left、right、top、bottom 设置无效。
- 相对:** relative，组件按照 Normal Flow（标准流、文档流、常规流、正常流）进行排布，可以通过 left、right、top、bottom 设置，定位参照对象是自己原来的位置。使用相对定位的组件不脱档，即组件原本占用的空间仍保留（后面组件不会贴上来）。
- 绝对:** absolute，组件脱离 Normal Flow（标准流、文档流、常规流、正常流），可以通过 left、right、top、bottom 设置，定位参考对象是相邻的父元素，如果找不到父级元素，参考对象就是整个页面；在绝大多数情况下，子元素的绝对定位都是相对于父元素进行定位，如果希望子元素相对于父元素进行定位，又不希望父元素脱标，那就设置父组件为相对定位，子组件为绝对定位，即一般可使用子绝父相的配置。
- 固定:** 固定定位可以理解为一种特殊的绝对定位，组件脱离 Normal Flow（标准流、文档流、常规流、正常流），可以通过 left、right、top、bottom 设置，定位参考对象是整个页面；当页面滚动时，组件悬浮固定不动。
- 层级:** 提供上移下移置顶置底的层级选择，对于开启定位的组件，可以通过上移下移置顶置底按钮控制组件的 z-index 属性来指定组件的层级，组件层级越高的越优先显示。组件新拖入编辑器中，层级都一样的情况下可以先操作上移 (z-index+1) 下移 (z-index-1)，然后可以单击置顶（判断当前页面中组件的最大 z-index，进行 +1 处理）置底（判断当前页面中组件的最小 z-index，进行 -1 处理）如果组件层级一样，则优先显示后面添加的组件，父组件层级高于子组件时，也不会盖住子组件。

层级



高级-style 绑定

style 绑定: 可以通过 style 直接绑定表达式、绑定变量的方式灵活进行样式配置，提供示例如下：

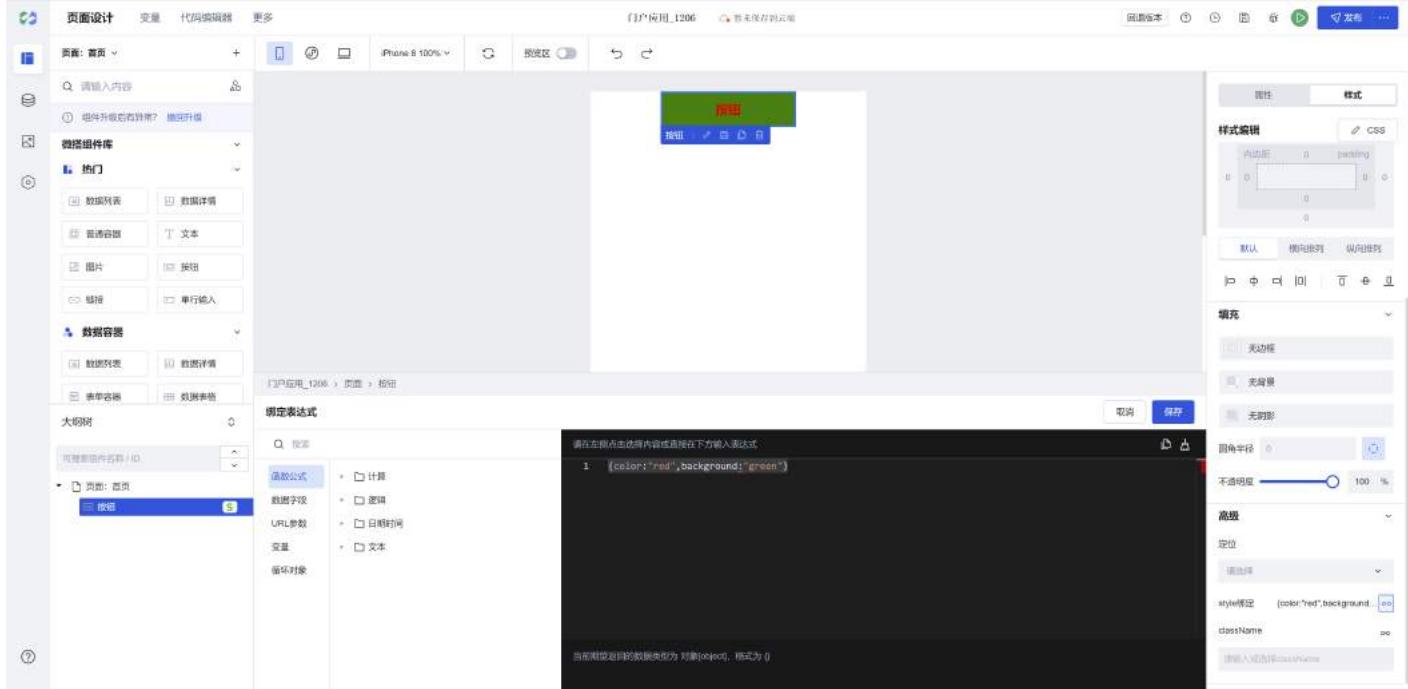
- 先添加一个按钮组件，选中该组件后单击 style 绑定。表达式中输入颜色和背景的对象代码。

△ 注意:

style 仅支持绑定对象。

```
{color:"red",background:"green"}
```

2. 单击保存，可以看到按钮组件文本变为红色，背景变为绿色。



当组件样式需要通过事件进行修改时，可使用 style 绑定变量的方式实现，组件样式通过事件修改实例可参见 [组件样式如何通过事件修改](#)。

高级-classname

classname 选择或 classname 绑定，需要定义整个应用级别或者页面级别的某些通用样式，整个应用或者页面的所有组件都可以快捷选择该样式。可以直接关联定义好的 classname 参数进行样式配置，提供示例如下：

- 首先需要在代码编辑器中定义样式 classname 参数（eg：字号50px，行高200px）。

```
.button-font-size {  
    font-size: 50px;  
}  
.button-line-height {  
    line-height: 200px;  
}
```

```

1 .button {
2   font-size: 50px;
3 }
4 .button-line-height {
5   line-height: 200px;
6 }
7 /* 可以使用 theme 中定义的 mixin 解释变量
8   ...
9 */
10 .index-class-name {
11   ...
12 }
13 // 使用 theme 中的变量
14 // color: example-primary-color;
15 .
16 // 想从 px 转换为 rem，将大括号外的 px 和 px 都去掉并用
17 // length-unit: rem;
18 // length-unit: rem (mixin);
19 // mixins();
20 // 可以直接写成
21 // <style>nest</style>
22 // color: #fff;
23 // 使用 theme 中的混合 (mixins)
24 // .mixins(green);
25 .
26 }
27
28

```

保存 CTRL+S

- 然后添加一个文本组件，文本组件的 `classname` 选项选择代码编辑器中定义的内容（支持多选），选择后组件效果如下，字号变为了50px，行高变成了200px：

门户应用_1206

页面设计 变量 代码编辑器 更多

页面：首页 + iPhone 6 100% 稀疏区

组件升级若有异常？[撤回升级](#)

微搭组件库

热门

- 数据列表
- 普通容器
- 图片
- 链接
- 文本
- 按钮
- 单行输入

数据容器

- 数据列表
- 表单容器

大纲树

可搜索组件名称：门户应用_1206

首页 > 文本

门户应用_1206 > 首页 > 文本

属性 样式

样式编辑

内边距 padding: 0 0 0 0;

填充

圆角半径 0px

不透明度 100%

高级

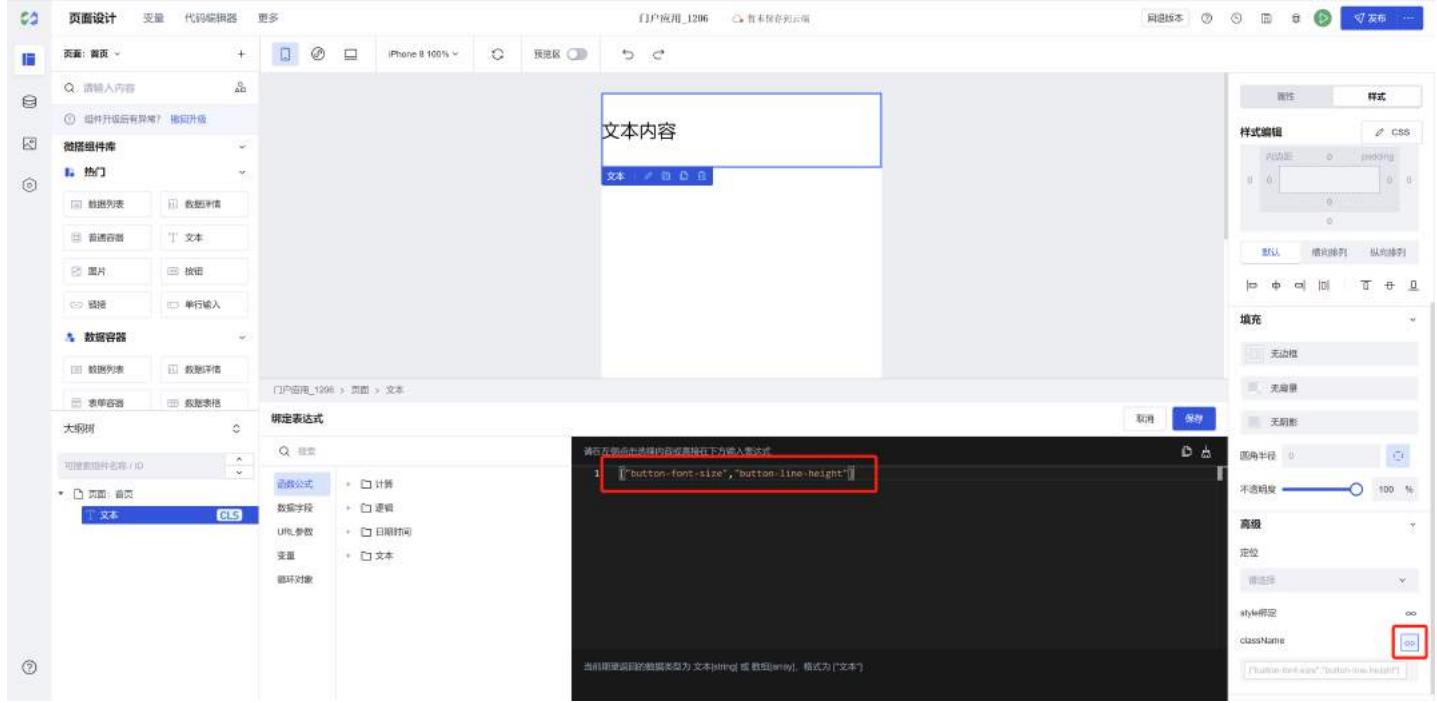
定位

button-font-size
button-line-height
className
button-line-height

② 说明：

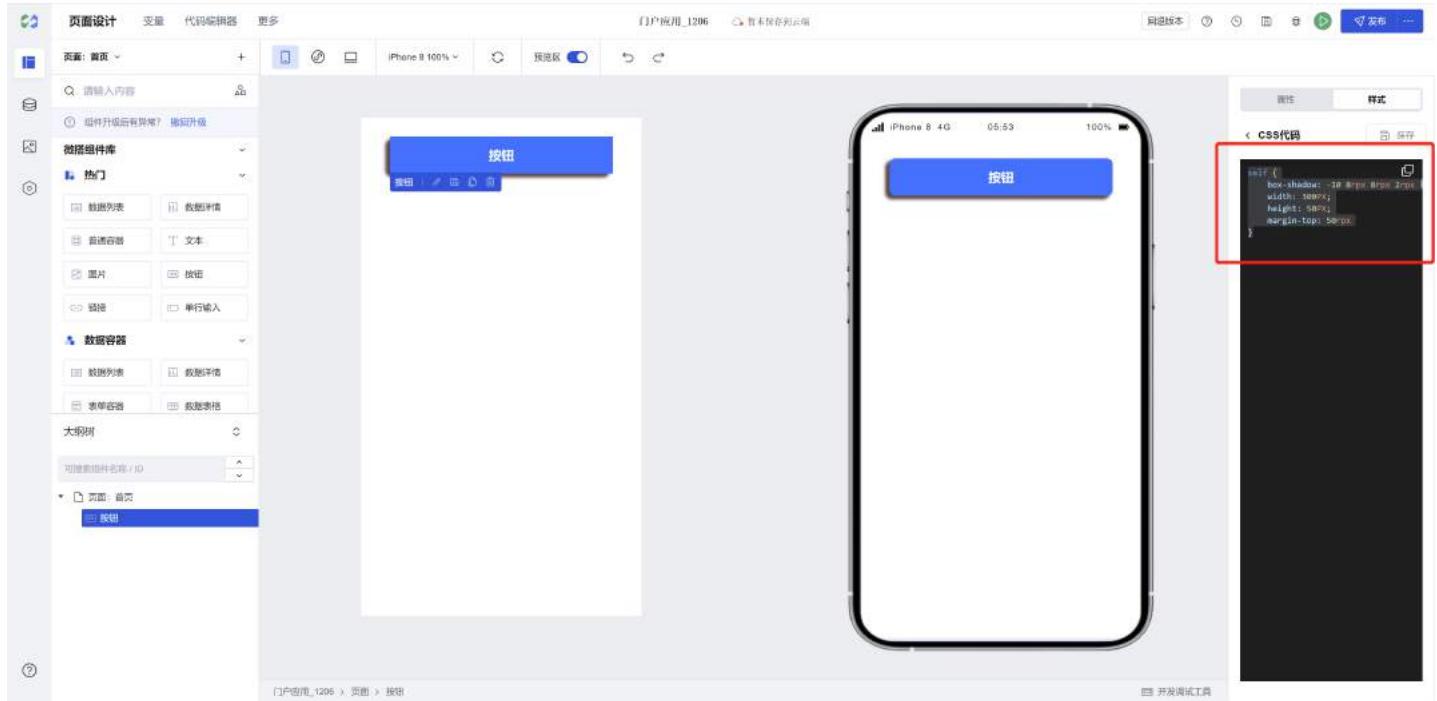
- 单个页面 style 中定义 classname 参数：classname 参数适用于当前页面，只有该页面中组件设定 classname 时可以选择。
- 全局 style 中定义 classname 参数：classname 参数适用于所有页面，所有页面组件设定 classname 时都可以选择。

同样的我们可以通过后面的绑定按钮进行参数的绑定:



CSS

样式面板的配置和 CSS CODE 为内容双向同步，例如在样式面板中配置了宽100px，那么 CSS CODE 中会显示 width: 100px，同时，可在 CSS CODE 中直接编写、修改样式代码来定义组件样式。



区块使用说明

协议声明区块

最近更新时间：2023-04-17 14:58:34

区块主要使用场景

- 用户信息的表单提交
- 用户信息的协议声明

图片展示及主要模块构成

图片展示

我已阅读并同意[《用户服务协议》](#)及[《隐私政策》](#)

主要模块构成

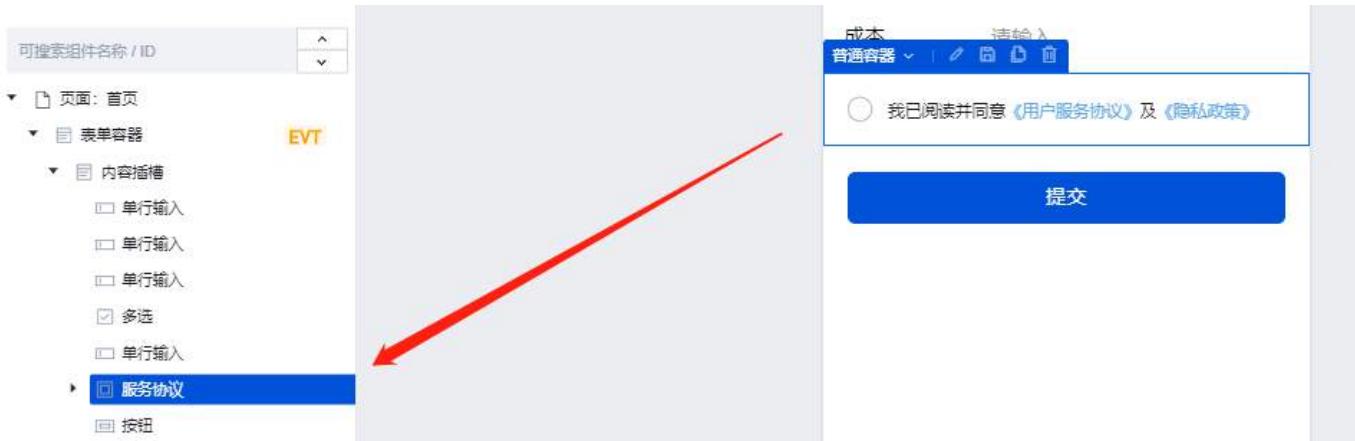


- 整体模块分为弹窗组件、普通容器、勾选组件。
- 在弹窗组件中，用文本组件包含了协议信息和隐私政策，滚动容器来进行滑动展示。
- 在普通容器中，用文本组件包含了协议声明勾选提示。
- 勾选组件是勾选协议的入口。

使用说明

表单场景使用

1. 当在表单场景提交信息时，可直接拖动服务协议区块至提交按钮上方，即可使用。



2. 当未勾选时，必填提示会在勾选框文字下方进行展示，勾选后即可提交。

我已阅读并同意《用户服务协议》及《隐私政策》

请确认协议再提交

3. 未勾选文字提示设置入口：单击勾选组件 > 在右侧属性栏 > 找到必填校验提示。



非表单场景使用

1. 当在非表单场景使用时，例如用户登录场景，先拖动服务协议区块至登录按钮上方。

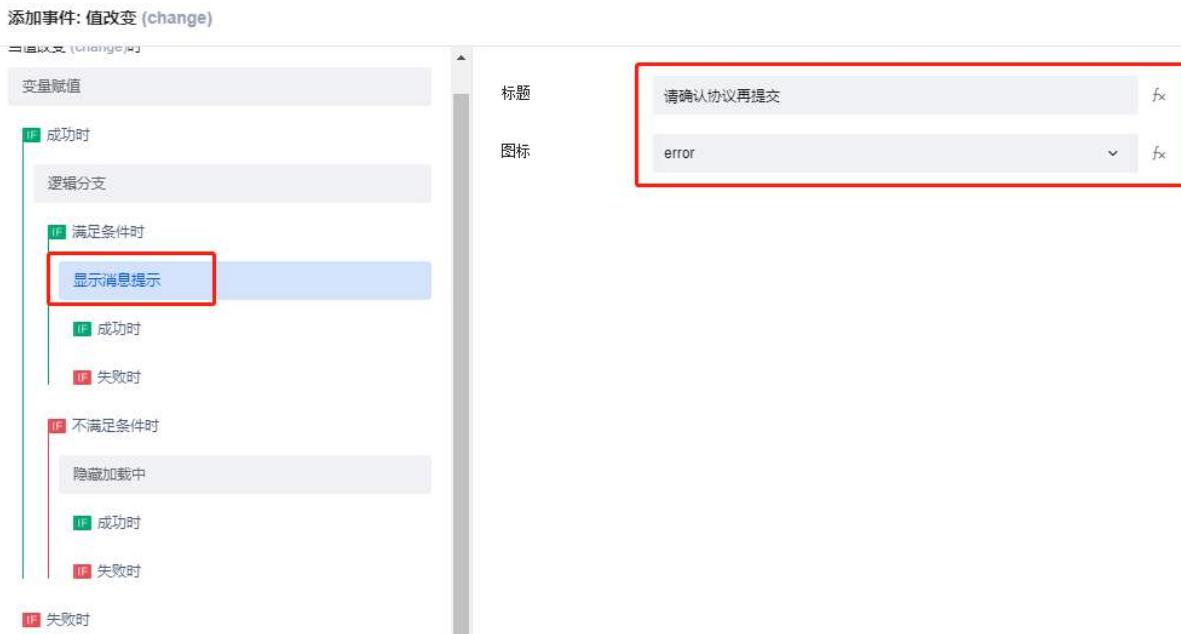
2. 在按钮的禁用属性上绑定已有变量 !\$page.dataset.state.customControl，如下图所示：



3. 当未勾选时，必填提示会以弹窗形式进行展示，勾选后即可提交。



4. 未勾选文字提示设置入口：单击勾选组件 > 在右侧属性栏 > 单击事件变量赋值等4个事件 > 单击显示消息提示 > 右侧即可设置标题和图标。



自定义组件

自定义组件库管理

最近更新时间：2021-12-31 10:28:01

基于微搭的低代码平台自定义组件，用户可以从源码级重新构建定制自己的组件逻辑和风格样式，并在应用内按需自由拖拽使用，更好地实现丰富的业务定制化需求。自定义组件库按账号维度存储，同一账号下的自定义组件库，可以支持跨应用跨环境使用。

② 说明：

由于自定义组件所依赖的资源（如图片等）存储在用户环境中，如果用户对应的环境被删除，则可能会导致该组件所依赖资源无法正常访问。

安装

使用低码自定义组件，请安装最新的版本的 cloudbase/cli，可以通过 npm 安装：

```
npm install -g @cloudbase/cli
```

自定义组件使用文档 (API)

- [创建第一个组件库](#)
- [实现自定义button组件](#)
- [实现自定义 showToast 动作](#)
- [组件库配置](#)
- [组件配置](#)
- [动作配置](#)

组件接入控制台

最近更新时间：2023-03-09 15:36:23

组件接入准备

注册腾讯云账号并完成实名认证

在开通腾讯云微搭低代码服务前，您需要先[注册腾讯云账号](#)，并前往[腾讯云控制台](#)完成[账号实名认证](#)。（如果您已完成，请跳过该步骤）

登录控制台

[腾讯云微搭低代码控制台](#)

创建组件库

1. 在微搭控制台，进入组件库页面，单击新建组件库来新建组件库。

组件库管理



The screenshot shows a table with columns: Name, Identifier, Description, Submission Time, Last Update Time, and Operation. A red box highlights the 'New Component Library' button at the top left of the table area.

名称	标识	描述	提交时间	最后更新时间	操作
暂无数据					

共 0 条

10 条 / 页 1 / 1 页

2. 在组件库管理中，包含了发布组件库、添加组件和导入组件三种选项：

- 发布组件库：将组件库从编辑状态变为发布状态。
- 添加组件：添加由线上编辑器拖拽生成的组件。

- 导入组件：导入通过本地代码开发生成的组件。

[组件详情页 / test](#)

自定义组件

组件发布历史

发布组件库

添加组件

导入组件

低码组件

100x100

item1



组件标识 Item

组件类型 图片

共 1 条

安装cli

使用低码自定义组件，请安装最新的版本的cloudbase/cli，可以通过 npm 安装。如果 `npm install -g @cloudbase/cli` 失败，您可能需要修改 npm 权限 ([opens new window](#))，或者以系统管理员身份运行。

```
npm install -g @cloudbase/cli
```

以源码自定义组件

在完成组件库配置后，用户可以从源码级重新构建定制自己的组件逻辑和风格样式，并在应用内按需自由拖拽使用，更好地实现丰富的业务定制化需求。

1. 编辑组件配置，详情请参见 [组件配置](#)。部分重要参数意义如下所示：

参数	作用
data	选填，组件可以接收的数据，用作组件入参，需要符合 Json Schema 规范。低码编辑器也会根据该配置，绘制组件的数据面板。
events	选填，组件可以被捕获的事件信息。低码编辑器也会根据该配置，绘制组件的事件面板的触发条件。
meta	选填，仅用于在低码平台的信息展示。

2. 编辑源码，详情请参见 [小程序组件源码](#)。部分重要参数意义如下所示：

参数	作用
properties	和组件配置的 data 字段对应，用户在低代码编辑器数据栏里配置的参数，会被透传到该字段内。
properties.style	平台保留字段，用户在低代码编辑器样式栏里配置的参数，会被透传到该字段内。

methods

和组件配置的 events 字段对应，通过 triggerEvent 方法，在合适的时机抛出自定义事件。

3. 编辑源码，详情请参见 [Web 组件源码](#)：

参数	作用
props	和组件配置的 data 字段对应，用户在低代码编辑器数据栏里配置的参数，会被透传到该字段内。
props.style	平台保留字段，用户在低代码编辑器样式栏里配置的参数，会被透传到该字段内。
props.events	平台保留字段，和组件配置的 events 字段对应，通过 props.events.\$customevent 方法，在合适的时机触发自定义事件。

在组件编辑器中自定义组件

除了使用源码实现自定义组件，也能通过组件编辑器将现有组件编辑组合成新组件。组件编辑器的使用方式与应用编辑器相似。

1. 单击添加组件或单击组件对应的 ，进入组件编辑器。

2. 组件编辑器介绍：

序号	模块	功能

1	组件选择区	展示平台官方组件及自定义组件。
2	组件编辑区	通过组件选择区可快速选中组件，以拖拽的方式调整组件的位置，并能实时预览。
3	组件配置区	该区域包括了元素编辑和组件编辑两种： <ul style="list-style-type: none">◦ 元素编辑：对使用到的单个组件元素进行数据、样式或事件的修改。◦ 组件编辑：针对该次编辑用到的所有组件，且只能以源码的方式进行编辑。

组件库发布

在完成组件编辑后，需要将组件库发布后才能正常使用。

1. 发布本地组件库（仅源码组件库需要）。

```
tcb lowcode publish
```

发布本地组件库，相当于更新了云端组件库的草稿，还需要到控制台里把草稿发布成正式版本。

2. 发布组件库。

完成组件库发布后，就能正常地使用自定义组件。

← 组件详情页 / test

自定义组件 组件发布历史

发布组件库 添加组件 导入组件

低码组件
100x100

item1

组件标识: Item
组件类型: 图片

数据源

数据源概述

最近更新时间：2023-07-03 15:38:57

腾讯云微搭低代码的数据源提供了数据集合本身的存储以及操纵数据的各种方法，本质上是一系列操作数据的方法集合。低码平台提供了数据源管理功能，可以创建、管理多个数据源。

数据源类型及功能概览

功能	说明
数据模型	标准数据模型 标准数据模型是一种最简单、方便的业务数据模型。开发者只需要关注业务模型即可在应用中展示或操作数据，无需关心数据的读写方式、存储过程以及权限控制。标准数据模型的模型字段由系统字段和自定义字段构成，开发者无法修改、删除系统字段。
	系统数据模型 系统数据模型是环境内默认创建且包含一定约束的数据模型，微搭低代码平台功能会使用系统数据模型数据，例如用户、部门等。
APIs	自定义 APIs 自定义 API 支持调用第三方服务接口或使用代码来实现业务逻辑。开发者可以在应用、工作流或其他自定义 API 中使用 API。
	API（开放服务） 可以打通外部应用数据或调用外部应用开放的服务，例如：腾讯地图、微信支付、企业微信等。微搭开发者只需要创建 API 时完成授权或少量填写一些配置信息，即可使用对应产品服务接口，而无需操心这些接口的实现细节。
数据管理后台	标准数据模型产生的数据均存储在微搭平台中，微搭提供了专门的数据管理后台来管理查看这些数据。

数据模型

数据模型概述

最近更新时间：2022-04-20 16:53:23

腾讯云微搭低代码支持开发者根据业务建立数据模型以及管理业务数据。开发者可以通过开发应用并引用相关数据模型来显示或操作这些业务数据。

开发应用时，可以使用标准、系统（只读）数据模型或混合使用。腾讯云微搭低代码已预置了一些系统数据模型和示例标准数据模型。

⚠ 注意

对数据模型的改动（创建、更新、删除）只有在使用该数据模型的应用发布时才会在正式环境生效。一个数据模型若被多个应用使用，则需要每一个使用该数据模型的应用重新发布，才可以使用最新的数据模型。特别地，应用区分预览和正式两种状态，两种状态使用的数据模型亦互不影响。

数据模型分类

腾讯云微搭低代码数据模型包含两种类型：

类型	说明
标准数据模型	标准数据模型是一种最简单、方便的业务数据模型。开发者只需要关注业务模型即可在应用中展示或操作数据，无需关心数据的读写方式、存储过程以及权限控制。标准数据模型的模型字段由系统字段和自定义字段构成，开发者无法修改、删除系统字段。 标准数据模型包含以下模型方法：新增、新增多条、删除、删除多条、更新、更新多条、查询单条、查询列表，是微搭默认的数据读写方式，数据存储于微搭云数据库内。
系统数据模型	系统数据模型是环境内默认创建且包含一定约束的数据模型，微搭低代码平台功能会使用系统数据模型数据，例如用户、部门等。系统数据模型对比标准数据模型具有更多的系统字段，用户无法删除系统数据模型。在模型方法中只支持查询单条和查询列表，用户无法通过模型方法进行写操作。

标准数据模型

概述

最近更新时间：2023-02-23 14:24:17

标准数据模型是一种最简单、方便的业务数据模型。开发者只需要关注业务模型即可在应用中展示或操作数据，无需关心数据的读写方式、存储过程以及权限控制。

创建方式

标准数据模型目前支持两种创建方式：

- 空白创建：新建一个只包含系统字段的数据模型，开发者可以后续新增模型字段。
- 从 Excel 导入：通过上传 Excel 文件，平台将对文件进行解析，完成模型字段和数据值的初始化。

更多详情请参见 [新建标准数据模型](#)。

模型字段

数据模型的模型字段描述了一个模型的数据结构，包括但不限于数据的类型、内容以及与其他模型的关系等。标准数据模型的模型字段由系统字段和自定义字段构成。开发者无法编辑、删除系统字段。

标准数据模型支持的字段类型，可以参见 [数据模型字段](#)。

模型方法

标准数据模型支持以下模型方法：新增 wedaCreate、新增多条 wedaBatchCreate、删除 wedaDelete、批量删除 wedaBatchDelete、更新 wedaUpdate、批量更新 wedaBatchUpdate、查询 wedaGetItem、查询列表 wedaGetRecords。

各数据模型的参数说明及示例用法，可以参见 [数据模型方法](#)。

⚠ 注意

原自建数据源已升级为标准数据模型，原自定义方法保留至模型方法中，开发者可以继续编辑、删除自定义方法，但不再支持新增。若开发者想要自定义业务逻辑，请参见 [自定义 API](#)。

数据存储与管理

目前标准数据模型的数据存储于腾讯云云开发数据库中。开发者可以通过以下方式对数据进行管理：

- 数据管理后台：微搭提供了数据管理后台来查看管理存储在微搭的数据，具体请参见 [管理数据](#)。
- 模型应用：具体请参见 [创建模型应用](#)。

新建标准数据模型

最近更新时间：2023-02-17 16:02:04

标准数据模型是一种最简单、方便的业务数据模型。本文介绍如何创建标准数据模型。

操作步骤

步骤1：填写基础信息

进入微搭控制台 > [数据模型](#) 页面，单击新建数据模型。

⚠ 注意：

标识为数据模型的唯一标识，在微搭应用编辑器、自定义代码中均需要借助这个标识来使用数据模型。

步骤2：选择创建方式

创建方式分为两种：

- **空白创建**：使用默认行为创建数据模型，只包含预置的系统字段，开发者可以稍后自行添加新字段。
- **Excel 上传**：通过上传 Excel 文件，平台将对文件进行解析，完成模型字段和数据值的初始化。

步骤3：编辑模型字段

标准数据模型会预置以下系统字段：数据标识 `_id`、创建时间 `createdAt`、更新时间 `updatedAt`、所有人 `owner`、创建人 `createBy`、修改人 `updateBy`、所属部门 `_departmentList`。

② 说明：

- 系统字段不可编辑不可删除，可单击 [添加字段](#) 来添加自定义字段。
- 数据模型只包含固定的模型方法，若想实现自己的业务逻辑，可以使用 [自定义 API](#)，具体方法参见 [新建自定义 API](#)。

系统数据模型

最近更新时间：2022-01-24 09:56:56

系统数据模型是微搭默认创建且包含一定约束的数据模型，微搭平台系统中会使用到系统数据模型的数据，例如用户、部门等。

创建方式

系统数据模型由微搭系统自动生成。开发者无法创建、删除系统数据模型，也无法修改系统数据模型名称。

模型字段

开发者可以根据业务需求新增、编辑、删除模型字段，能力与 [标准数据模型](#) 基本相同，但不可使用[关联关系字段](#)类型。

模型方法

系统数据模型暂时只支持以下模型方法：

- [查询](#)：查询一条指定数据标识的记录。
- [查询列表](#)：根据条件以分页列表格式返回多条记录。

数据存储

系统数据模型的存储方式同标准数据模型。

数据模型字段

最近更新时间：2023-08-11 15:48:02

概述

数据模型 字段为数据源的表结构基础，可以理解为 Excel 表格的表头（列），而数据可以理解为 Excel 表格的行。

字段类型

开发者可自定义数据模型字段，自定义字段支持类型如下：

新版字段	历史字段	类型	校验规则	传参示例
文本-短文本	中文姓名、字符串	string	无	"this is a string"
文本-长文本	-	string	无	"this is a long-text"
富文本	富文本	string	"this is a rtf"	-
邮箱	邮箱	string	含有 xx@yy.zz	"email@qq.com"
电话-电话号码	电话号码	string	有0开头的2, 3位区号或者没有区号的7-8位字符串	"0271-1234567"或"027-1234567"或"1234567"
电话-手机号码	手机号码	string	11位字符串	"12345678900"
图片	图片	string	默认从前端组件获得图片的 cloudId	cloud://xxx.xxx.xxx.png
文件	文件	string	默认从前端组件获得文件的 cloudId	cloud://xxx.xxx.xxx.png
网址	网址	string	符合网址规范的字符	http://xxx.xxx.xx
关联关系	关联关系	string	默认从前端组件获得	关联数据的_id值
主子明细	主子明细	string	默认从前端组件获得	关联数据的_id值
数字	数字、年龄	number	数字	123456.78
日期时间-日期时间	新增	number	默认从前端组件获取的时间戳 ms 为单位	1645977600000
日期时间-日期	日期	number	默认从前端组件获取的时间戳 ms 为单位	1645977600000
日期时间-时间	时间	number	默认从前端组件获取的时间戳 ms 为单位	1645977600000
布尔值	布尔值	boolean	true or false	true
数组	数组	array	根据数组的元素类型进行校验	["abc", "321"]
对象	对象	object	-	-
枚举-单选	是否枚举属性	string	所填值必须为用户设置的枚举值中的某一个	"牛奶"
枚举-多选	-	array	数组	["牛奶", "面包"]
地理位置	-	object	对象的固定格式，详见传参示例	{ "geopoint": { "type": "Point", "coordinates": [40.56, 5.89] }, "address": "深圳市南山区深南大道**号" }
地区	-	string	无	以逗号分割的地理位置 eg：陕西省,西安市,雁塔区

新版字段	历史字段	类型	校验规则	传参示例
计算公式	-	string	无	客户自定义的表达式；例如 (#lastname +#firstname)
自动编号	-	string	无	不填（后端自动补齐）；如果客户传参，使用客户定义的值

字段通用配置

- 字段名称：支持中英文。
- 字段标识：仅支持英文字符、下划线。
- 数据类型：即字段类型选择，可参考字段类型设置所需字段设置。
- 是否必填：该字段是否必填。
- 是否唯一：该字段的取值是否允许重复，设置为唯一则不允许填重复值。
- 是否为主列：数据类型为文本时可选择。主列是记录显示的用户可理解的文本标识(通常是名称或数字)。当用户需要从父数据模型业务数据列表中选中此条业务数据后，系统将向用户显示此列中的值。

字段使用说明

枚举

字段的“是否枚举”属性已升级为枚举字段，枚举字段属性设置参考：

- 选择设置：支持单选、多选。
 - 单选：仅允许选择单个枚举值。
 - 多选：允许选择多个枚举值。
- 关联选项集：枚举字段的枚举值依赖于通用选项集，当没有合适选项集时，可单击立即创建。

② 说明：

历史已经设置了“是否枚举”属性的字段仍可正常使用，官方建议升级为枚举类型字段。

计算公式

- 新增计算公式类型字段，选择结果类型。
- 结果类型支持文本、日期时间、数字。
- 输入计算表达式，表达式规则请参见 [表达式](#)。

自动编号

自动编号性质与字符串类型相同，只是在创建一条记录时，若用户没有输入值，则自动生成的指定格式的字符串值。开发者可以按其偏好自定义这些字段的格式，然后依靠系统在运行时生成自动填充它们的匹配值。

自动编号类型字段支持三种编号类型：字符串前缀、日期前缀、自定义。

关联关系

可以通过该字段关联其它数据模型，该字段会记录关联数据模型的数据 ID。如学生表中可设置课程 ID 字段为关联关系，查询时可以通过课程 ID 获取到关联课程的信息。

- 关联类型：可选一条数据对应一条关联数据，或多条数据对一条关联数据。
- 删除行为：可选删除数据同时删除关联表数据，或删除数据时不删除关联表数据。

字符串前缀

在该类型中，自动编号将包含自动递增的数字，具有可选的字符串常量前缀。

- 规则：{前缀}-{SEQNUM:最小位数+起始值}
- 最小位数：自动生成的编号序列要包含的最小位数。随着编号序列继续增加，它可能比此最小值长。
- 起始值：用于自动编号列的序列号部分的起始值。

- 示例：前缀：CAR，最小位数：4，起始值：1000
- 结果：CAR-1000, CAR-1001, CAR-1002, CAR-1003

日期前缀

在该类型中，自动编号将包含自动递增的数字，具有确定格式的日期前缀。行的日期部分将反映以 UTC 时间创建行的当前日期和时间。内置供选择的不同日期格式，包含：

- yyyy-MM-dd
- MM-dd-yyyy
- MM-dd-yy
- dd-MM-yy

规则：{DATETIMEUTC:日期格式}-{SEQNUM:最小位数+起始值}

举例：日期格式：yyyy-MM-dd，最小位数：4，起始值：1000

结果：2010-01-01-0001, 2010-01-01-0002, 2010-01-01-0003

自定义

在该类型中，可能包括字符串常量、自动递增数字、确定格式的日期格或随机的字母数字序列。

单击添加格式可以向格式字段输入框文本后面插入默认格式。

支持的格式语法如下：

格式类型	语法示例	预览	默认添加格式
连续数字 SEQNUM	-CAR-{SEQNUM:3}	CAR-123	{SEQNUM:4}
时间日期 DATETIMEUTC	CAR- {DATETIMEUTC:yyyyMMddhhmmss}	CAR- 20211223164721	{DATETIMEUTC:yyyy-MM- dd}
随机字符串 RANDSTRING	CAR-{SEQNUM:3}-{RANDSTRING:6}	CAR-123-AB7LSF	{RANDSTRING:4}

② 说明：

自动编号类型的字段和其他类型字段功能一致。一样可以用于表单提交或用户自己输入值。唯一不同的是该字段设置成非必填的情况下，用户创建记录时又没有输入该字段的值，则按照指定格式填充数据。

用户自己填入值的时候，不占用自增序列值。即：自动创建了 1000, 1001后，用户新增一条 aaa 的值，再一次自动新增时为1002，不会产生序列中断。

② 说明：

自动编号类型字段在模型的方法中同数据标识（_id），新增、更新方法的入参不包含自动编号类型字段，即不可新增或修改。

使用和管理数据模型

最近更新时间：2023-04-13 10:36:57

本文将介绍如何使用和管理数据模型。

使用数据模型

目前在应用编辑器、工作流、自定义数据模型和自定义 API 中，都可以使用数据模型，支持可视化和自定义代码两种方式。

通过变量调用

在微搭应用编辑器中，通过变量管理可以添加数据源变量来使用数据模型，详情可参见 [变量使用文档](#)。

通过组件调用

- 在数据容器等组件中，可以绑定数据模型，并借助数据模型的描述信息，自动对组件进行配置，详情可参见 [组件绑定数据](#)。
- 在组件行为事件中也可以操作数据模型，详情可参见 [行为事件-执行动作](#)。

通过低代码编辑器调用

进入应用编辑器，可以在应用的 [代码编辑器](#) 及微搭组件的组件代码中使用。

调用示例：

```
export default async function({event, data}) {
  const result = await app.cloud.callModel({
    name: 'bzsjmxs1_ho50gz8', // 数据模型标识
    methodName: 'wedaCreate', // 数据模型方法
    params: { // 数据模型方法入参
      name: "张三",
      age: 20,
    },
  });
}
```

在工作流中调用

可以在工作流中通过数据触发流，具体请参见 [数据模型触发](#)。

自定义 API 调用

在自定义 API 中通过[自定义代码](#)方式，也可以调用数据模型。

调用示例：

```
module.exports = async function (params, context) {
  const result = await context.callModel({
    name: 'bzsjmxs1_ho50gz8',
    methodName: 'wedaCreate',
    params: { // 数据模型方法入参
      name: "张三",
      age: 20,
    },
  });

  return {
    _id: '123456',
  };
};
```

编辑数据模型

有两种方式可以对数据模型进行编辑：

- 在 [数据模型](#) 列表页面，找到编辑的数据模型，在操作列单击编辑。
- 在数据模型详情页面，单击右上角编辑。

发布数据模型

数据模型需要发布才会在正式环境生效，有两种方式可以发布数据模型：

- 在 [数据模型](#) 页面，找到编辑的数据模型，在操作列单击更多 > 发布。



- 在数据模型详情页面，进入基本设置选项页，在基础信息卡片中状态 > 立即发布。



删除数据模型

在数据模型列表页面，找到编辑的数据模型，在操作列单击更多 > 删除。

⚠ 注意：

数据模型删除前，必须解绑在所有应用的关联使用。

数据模型方法

最近更新时间：2023-03-30 09:04:13

腾讯云微搭低代码的数据源提供了数据模型，开发者可以使用 `context.callModel` 方法来操作数据模型，具体模型方法包括：新增 `wedaCreate`、新增多条 `wedaBatchCreate`、删除 `wedaDelete`、批量删除 `wedaBatchDelete`、更新 `wedaUpdate`、批量更新 `wedaBatchUpdate`、查询 `wedaGetItem`、查询列表 `wedaGetRecords`。

本文以如下数据模型为例，说明数据模型方法的使用。

[数据模型 / 文档示例](#) 导入 导出

模型配置 视图配置 事件处理 关联应用 基本设置

模型字段

字段名称	字段标识	数据类型	是否必填	是否唯一 ①
姓氏	lastname	文本 短文本	否	否
名字	firstname	文本 短文本	否	否
单价	price	数字	否	否
数量	amount	数字	否	否
姓名	name	计算公式	否	否
总价	total	计算公式	否	否

[展开系统字段 ▾](#)

模型方法

① 数据模型不支持新增自定义方法，可以使用[自定义APIs](#)实现自定义业务逻辑

新增

新增多条

删除

删除多条

更新

更新多条

查询

查询列表

名称 新增
标识 wedaCreate
入参

字段名称	字段标识	数据类型	是否必填
姓氏	lastname	文本 短文本	否
名字	firstname	文本 短文本	否
单价	price	数字	否
数量	amount	数字	否

出参

字段名称	字段标识	数据类型
新记录的ID	_id	数字

新增 wedaCreate

新增一条记录。

入参说明

新增方法的入参为数据模型全部的非系统字段，例如：`name`、`age` 等。

```
{  
  "name": "张三",  
  "age": 15  
}
```

出参说明

字段名称	字段标识	数据类型	说明
新记录的 ID	_id	文本	创建成功的数据标识

```
{
  "_id": "16db756f62d7c0870b8caa2a39d33dcb"
}
```

使用示例

```
module.exports = async function (params, context) {
  const result = await context.callModel({
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看
    methodName: 'wedaCreate', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
    params: {
      lastname: "张",
      firstname: "三",
      price: 15,
      amount: 20,
    }, // 数据模型方法的入参
  });

  // 在这里返回这个方法的结果，需要与出参定义的结构映射
  return result;
};
```

新增多条 wedaBatchCreate

新增多条记录。

入参说明

字段名称	字段标识	数据类型	是否必填	说明
数据内容	records	对象数组	是	待新增的多条数据

```
{
  "records": [
    {
      "name": "张三",
      "age": 15
    }
  ]
}
```

出参说明

字段名称	字段标识	数据类型	说明
新记录的 ID 数组	ids	文本数组	创建成功的数据标识

```
{  
  "ids": ["16db756f62d7c0870b8caa2a39d33dcb"]  
}
```

使用示例

```
module.exports = async function (params, context) {  
  const result = await context.callModel({  
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看  
    methodName: 'wedaBatchCreate', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法  
    params:  
      records:[ {  
        lastname: "张",  
        firstname: "三",  
        price: 15,  
        amount: 20,  
      }]  
    },// 数据模型方法的入参  
  );  
  
  // 在这里返回这个方法的结果，需要与出参定义的结构映射  
  return result;  
};
```

删除 wedaDelete

删除一条指定数据标识的记录。

入参说明

字段名称	字段标识	数据类型	是否必填	说明
数据标识	_id	文本	是	待删除的记录数据标识

```
{  
  "_id": "16db756f62d7c0870b8caa2a39d33dcb"  
}
```

出参说明

字段名称	字段标识	数据类型	说明
删除的记录条数	deleted	数字	成功删除的记录条数

```
{  
  "deleted": 1  
}
```

使用示例

```

module.exports = async function (params, context) {
  const result = await context.callModel({
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看
    methodName: 'wedaDelete', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
    params: {
      _id: "84f7894263706d0900950bc52687737f"
    }, // 数据模型方法的入参
  });

  // 在这里返回这个方法的结果，需要与出参定义的结构映射
  return result;
}

```

批量删除 wedaBatchDelete

根据条件删除多条记录。

入参说明

字段名称	字段标识	数据类型	是否必填	说明
查询条件	where	对象数组	否	匹配的查询条件
字段名称	where[].key	文本	是	模型字段标识
关系	where[].rel	文本	是	判断关系。可选值： "eq" 表示相等， "neq" 表示不相等， "lt" 表示小于， "lte" 表示小于或等于， "gt" 表示大于， "gte" 表示大于或等于， "search" 表示包含文字内容
值	where[].val	-	是	判断值。根据字段类型写入不同的值

```
{
  "where": [
    {
      "key": "name",
      "rel": "eq",
      "val": "张三"
    }
  ]
}
```

出参说明

字段名称	字段标识	数据类型	说明
删除的记录条数	deleted	数字	成功删除的记录条数

使用示例

```

module.exports = async function (params, context) {
  const result = await context.callModel({
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看
    methodName: 'wedaBatchDelete', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
    params: {
      where: [{ ...
    }
  });

```

```
key: "firstname",
rel: "eq", // 可选值有: eq(相等) neq(不相等) lt(小于) lte(小于或等于) gt(大于) gte(大于或等于) search(包含文字内容)
val: "张",
}],
},
}, // 数据模型方法的入参
});

// 在这里返回这个方法的结果，需要与出参定义的结构映射
return result;
};
```

更新 wedaUpdate

更新一条指定数据标识的记录。

入参说明

字段名称	字段标识	数据类型	是否必填	说明
数据标识	_id	文本	是	待更新记录的数据标识

```
{
  "_id": "16db756f62d7c0870b8caa2a39d33dcb",
  "name": "张三",
  "age": 15
}
```

出参说明

字段名称	字段标识	数据类型	说明
新记录的 ID 数组	ids	文本数组	创建成功的数据标识

```
{
  "ids": ["16db756f62d7c0870b8caa2a39d33dcb"]
}
```

使用示例

```
module.exports = async function (params, context) {
  const result = await context.callModel({
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看
    methodName: 'wedaUpdate', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
    params: {
      _id: "84f7894263706d0900950bc52687737f", // 需要更新的记录标识
      lastname: "黄", // 更新后的值
    },
  });

  // 在这里返回这个方法的结果，需要与出参定义的结构映射
  return result;
};
```

批量更新 wedaBatchUpdate

根据条件更新多条记录。

入参说明

字段名称	字段标识	数据类型	是否必填	说明
查询条件	where	对象数组	是	匹配的查询条件
字段名称	where[].key	文本	是	模型字段标识
关系	where[].rel	文本	是	判断关系。可选值： "eq" 表示相等， "neq" 表示不相等， "lt" 表示小于， "lte" 表示小于或等于， "gt" 表示大于， "gte" 表示大于或等于， "search" 表示包含文字内容
值	where[].val	-	是	判断值。根据字段类型写入不同的值
更新数据	record	对象	是	使用该数据，更新全部匹配条件的记录

```
{
  "where": [
    {
      "key": "name",
      "rel": "eq",
      "val": "张三"
    }
  ],
  "record": {
    "name": "张三",
    "age": 12
  }
}
```

出参说明

字段名称	字段标识	数据类型	说明
更新的记录条数	updated	数字	更新成功的数据标识

```
{
  "updated": 1
}
```

使用示例

匹配 where 条件的记录都更新为 record 的内容。

```
module.exports = async function (params, context) {
  const result = await context.callModel({
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看
    methodName: 'wedaBatchUpdate', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
    params: {
      record: {
        lastname: "黄", // 更新后的值
      },
    }
  })
}
```

```
where: [{  
    key: "firstname",  
    rel: "eq", // 可选值有: eq(相等) neq(不相等) lt(小于) lte(小于或等于) gt(大于) gte(大于或等于) search(包含文字内容)  
    val: "张",  
},  
},  
});  
  
// 在这里返回这个方法的结果, 需要与出参定义的结构映射  
return result;  
};
```

查询 wedaGetItem

查询一条指定数据标识的记录。

入参说明

字段名称	字段标识	数据类型	是否必填	说明
查询条件	where	对象数组	否	匹配的查询条件
字段名称	where[] .key	文本	是	模型字段标识
关系	where[] .rel	文本	是	判断关系。可选值: "eq"表示相等, "neq"表示不相等, "lt"表示小于, "lte"表示小于或等于, "gt"表示大于, "gte"表示大于或等于, "search"表示包含文字内容
值	where[] .val	-	是	判断值。根据字段类型写入不同的值

```
{  
"where": [  
{  
"key": "name",  
"rel": "eq",  
"val": "张三"  
}]  
}
```

出参说明

出参为数据模型字段, 例如: name、age 等。

```
{  
"_id": "16db756f62d7c0870b8caa2a39d33dcb",  
"name": "张三",  
"age": 15  
}
```

使用示例

根据查询条件返回一条匹配的记录。

```
module.exports = async function (params, context) {  
const result = await context.callModel({
```

```

name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看
methodName: 'wedaGetItem', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
params: {
  _id: "84f7894263706d0900950bc52687737f", // 指定某个具体的数据标识
  where: [{ // 或根据指定的条件
    key: "firstname",
    rel: "eq", // 可选值有: eq(相等) neq(不相等) lt(小于) lte(小于或等于) gt(大于) gte(大于或等于) search(包含文字内容)
    val: "张",
  }],
},
});
};

// 在这里返回这个方法的结果，需要与出参定义的结构映射
return result;
};

```

查询列表 wedaGetRecords

根据条件以分页列表格式返回多条记录。

入参说明

字段名称	字段标识	数据类型	是否必填	说明
查询条件	where	对象数组	否	匹配的查询条件
字段名称	where[].key	文本	是	模型字段标识
关系	where[].rel	文本	是	判断关系。可选值: "eq"表示相等, "neq"表示不相等, "lt"表示小于, "lte"表示小于或等于, "gt"表示大于, "gte"表示大于或等于, "search"表示包含文字内容
值	where[].val	-	是	判断值。根据字段类型写入不同的值
页码	pageNo	数字	否	从1开始
每页记录数	pageSize	数字	否	每次查询的最大记录数
排序字段名称	orderBy	文本	否	按某个字段进行排序
排序方式	orderType	文本	否	"desc" 表示降序, "asc"表示升序

```

{
  "where": [
    {
      "key": "name",
      "rel": "eq",
      "val": "张三"
    }
  ],
  "pageNo": 1,
  "pageSize": 20,
  "orderBy": "age",
  "orderType": "desc",
}

```

出参说明

字段名称	字段标识	数据类型	说明
记录总条数	total	数字	所有匹配条件的总条数
页码	pageNo	数字	当前页码
每页记录数	pageSize	数字	每次查询的最大记录数
记录列表	records	对象数组	查询的结果列表

```
{  
  "records": [  
    {  
      "name": "张三",  
      "age": 2  
    }  
  ],  
  "total": 100,  
  "pageNo": 1,  
  "pageSize": 20  
}
```

使用示例

根据查询条件分页返回匹配的记录。

```
module.exports = async function (params, context) {  
  const result = await context.callModel({  
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看  
    methodName: 'wedaGetRecords', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法  
    params: {  
      pageNo: 1,  
      pageSize: 10,  
      orderBy: "firstname",  
      orderType: "desc",  
      where: [{ // 或根据指定的条件  
        key: "firstname",  
        rel: "eq", // 可选值有: eq(相等) neq(不相等) lt(小于) lte(小于或等于) gt(大于) gte(大于或等于) search(包含文字内容)  
        val: "张",  
      }],  
    },  
  });  
  
  // 在这里返回这个方法的结果，需要与出参定义的结构映射  
  return result;  
};
```

数据模型方法查询条件 Where

数据模型方法中 **where** 字段可以支持简单与复杂两种类型的查询条件，请参见 [简单与复杂查询](#)。

数据模型方法V2

最近更新时间: 2023-08-14 15:54:27

开发者可以使用 `$w.cloud.callDataSource` 方法来调用数据源，包括数据模型及 APIs。

入参

params: ICallDataSourceParams

属性	类型	必填	说明
dataSourceName	string	是	数据源标识
methodName	string	是	数据源方法名
params	object	是	方法参数，根据方法实际入参填写

如果数据源是数据模型，因为其方法皆由平台提供，其可用的数据源方法（methodName）是：

- 新增: wedaCreateV2
- 新增多条: wedaBatchCreateV2
- 删除: wedaDeleteV2
- 删除多条: wedaBatchDeleteV2
- 更新: wedaUpdateV2
- 更新多条: wedaBatchUpdateV2
- 查询: wedaGetItemV2
- 查询多条: wedaGetRecordsV2

方法入参结构及示例

② 说明:

示例数据源 sjmx_ftf4loj 包含名为 name，类型为 string 的字段，包含名为 email，类型为 string 的字段。

字段名称	字段标识	数据类型	是否必填	是否唯一 ①
关联关系	relation	关联关系	否	否
email	email	邮箱	否	否
name	name	文本 单行文本	否	否

新增 (wedaCreateV2)

入参结构

属性	类型	默认值	示例	必填	说明
data	{ [key: string]: any; }	无	{name: "juli"}	是	对应数据源的字段结构

出参结构

属性	类型	默认值	示例	必填	说明
id	string	无	"7L5G32U9PE"	是	id 为对应创建数据源数据对应的标识

属性	类型	默认值	示例	必填	说明
id	string	无	"7L5G32U9PE"	是	Id 为对应创建数据源数据对应的标识 (后续废弃)

编辑器 Javascript 代码示例

```
export default async function ({event, data}) {
  try {
    const data = await $w.cloud.callDataSource({
      dataSourceName: "sjmx_ftf41oj",
      methodName: "wedaCreateV2",
      params: {
        data: {
          email: "bar@weda.io",
          name: "foo",
        },
      },
    });
    console.log("请求结果", data); // {"id": "f8f6930864c11bde006ff6c4662f9bf6"}
  } catch (e) {
    console.log("错误代码", e.code, "错误信息", e.message);
  }
}
```

添加关联关系

现有数据模型 sjmx_ftf41oj (数据模型) , glmx_b2z3oh7 (关联模型) 。

如果试图在 sjmx_ftf41oj 模型中添加一条数据关联 glmx_b2z3oh7 中的一条数据，数据标识为 62fd7a3664d1e8ba00cc579835a1ded2。

文本	数据标识	创建时间	更新时间	所有人	创建人	修改人	操作
test	62fd7a3664d1e8ba...	2023-08-08 15:03:22	2023-08-08 15:03:22	administrator	administrator	administrator	[View] [Edit] [Delete]

```
export default async function ({event, data}) {
  try {
    const data = await $w.cloud.callDataSource({
      dataSourceName: "sjmx_ftf41oj",
      methodName: "wedaCreateV2",
      params: {
        data: {
          email: "bar@weda.io",
          name: "foo",
        },
        relation: "62fd7a3664d1e8ba00cc579835a1ded2"
      },
    },
  });
}
```

```

console.log("请求结果", data); // {"id": "f8f6930864c11bde006ff6c4662f9bf6"}
} catch (e) {
console.log("错误代码", e.code, "错误信息", e.message);
}
}

```

新增多条 (wedaBatchCreateV2)

入参结构

属性	类型	默认值	示例	必填	说明
data	{ [key: string]: any }[]	无	[{name: "juli"}]	是	不能为空数组，传入空对象会被忽略

出参结构

属性	类型	默认值	示例	说明
idList	idList: string[]	无	["7L5G32U9PE"]	idList 为对应创建数据源数据对应的标识列表
IdList	IdList: string[]	无	["7L5G32U9PE"]	IdList 为对应创建数据源数据对应的标识列表（后续废弃）

编辑器 Javascript 代码示例

```

export default async function ({event, data}) {
try {
const data = await $w.cloud.callDataSource({
dataSourceName: "sjmx_ftf41oj",
methodName: "wedaBatchCreateV2",
params: {
data: [
{
email: "bar@weda.io",
name: "foo",
},
{
email: "juli@weda.io",
name: "juli",
},
],
},
});
console.log("请求结果", data); // {"idList": ["f8f6930864c11fee007010104a2589c4","f8f6930864c11fee0070101171d96063"]}
} catch (e) {
console.log("错误代码", e.code, "错误信息", e.message);
}
}

```

添加关联关系

现有数据模型 sjmx_ftf41oj (数据模型) , glmx_b2z3oh7 (关联模型) 。

如果试图在 sjmx_ftf41oj 模型中添加多条数据关联 glmx_b2z3oh7 中的多条数据，数据标识为6243df、76589xf。

例如： foo 关联6243df, juli 关联76589xf。

```

export default async function ({event, data}) {
  try {
    const data = await $w.cloud.callDataSource({
      dataSourceName: "sjmx_ftf41oj",
      methodName: "wedaBatchCreateV2",
      params: {
        data: [
          {
            email: "bar@weda.io",
            name: "foo",
            relation: "6243df",
          },
          {
            email: "juli@weda.io",
            name: "juli",
            relation: "76589xf",
          },
        ],
      },
    });
    console.log("请求结果", data); // {"idList": ["f8f6930864c11fee007010104a2589c4", "f8f6930864c11fee0070101171d96063"]}
  } catch (e) {
    console.log("错误代码", e.code, "错误信息", e.message);
  }
}

```

更新 (wedaUpdateV2)

入参结构

属性	类型	默认值	示例	必填	说明
data	{ [key: string]: any }[]	无	[{name: "juli"}]	是	不能为空数组，传入空对象会被忽略
filter	{ where: FilterObject }[]	无	{filter: {where: {_id:{\$eq:"foo"}}}}	是	不能为空数组，传入空对象会被忽略

② 说明:

- FilterObject 为复杂查询结构，请参考查询参数说明。
- 数据源筛选条件满足条数为 1 以上进行时进行单条更新会出错

出参结构

属性	类型	默认值	示例	说明
count	count: 0 or 1	无	1	变更的条数，返回非 0 值代表更新成功。
Count	Count: 0 or 1	无	1	变更的条数，返回非 0 值代表更新成功。（后续废弃）

编辑器 Javascript 代码示例

```

export default async function ({event, data}) {
  try {
    const data = await $w.cloud.callDataSource({

```

```
dataSourceName: "sjmx_ftf41oj",
methodName: "wedaUpdateV2",
params: {
// 更新数据
data: {
email: "zhangSan@weda.io",
},
// 筛选内容，筛选内容推荐使用编辑器数据筛选器生成
filter: {
where: {
$and: [
{
_id: {
$eq: "4ebb756064c11bc9006e5d2e4f9b73a8", // 更新单条时，推荐传入_id数据标识进行操作
},
},
],
},
},
},
},
});
console.log("请求结果", data); // { count: 1 }
} catch (e) {
console.log("错误代码", e.code, "错误信息", e.message);
}
}
```

添加关联关系

现有数据模型 sjmx_ftf41oj (数据模型) , glmx_b2z3oh7 (关联模型) 。

如果试图在 sjmx_ftf41oj 模型中添加一条数据关联 glmx_b2z3oh7 中的一条数据，数据标识为62fd7a3664d1e8ba00cc579835a1ded2。

文本	数据标识	创建时间	更新时间	所有人	创建人	修改人	操作
test	62fd7a3664d1e8ba00cc579835a1ded2	2023-08-08 15:03:22	2023-08-08 15:03:22	administrator	administrator	administrator	查看 编辑 删除

```
export default async function ({event, data}) {
try {
const data = await $w.cloud.callDataSource({
dataSourceName: "sjmx_ftf41oj",
methodName: "wedaUpdateV2",
params: {
// 更新数据
data: {
email: "zhangSan@weda.io",
relation: "62fd7a3664d1e8ba00cc579835a1ded2",

```

```
},
// 筛选内容，筛选内容推荐使用编辑器数据筛选器生成
filter: {
where: {
$and: [
{
{
_id: {
$eq: "4ebb756064c11bc9006e5d2e4f9b73a8", // 更新单条时，推荐传入_id数据标识进行操作
},
},
],
},
},
},
},
}),
console.log("请求结果", data); // { count: 1 }
} catch (e) {
console.log("错误代码", e.code, "错误信息", e.message);
}
}
```

更新多条 (wedaBatchUpdateV2)

入参结构

属性	类型	默认值	示例	必填	说明
data	{ [key: string]: any }[]	无	[{name: "juli"}]	是	不能为空数组，传入空对象会被忽略
filter	{ where: FilterObject }[]	无	{filter: {where: {}}}	是	复杂查询结构

② 说明：

FilterObject 为复杂查询结构，请参考查询参数说明。

出参结构

属性	类型	默认值	示例	说明
count	count: 0 or 1	无	1	变更的条数，返回非 0 值代表更新成功
Count	Count: 0 or 1	无	1	变更的条数，返回非 0 值代表更新成功（后续废弃）

编辑器 Javascript 代码示例

```
export default async function ({event, data}) {
try {
const data = await $w.cloud.callDataSource({
dataSourceName: "sjmx_ftf4loj",
methodName: "wedaBatchUpdateV2",
params: {
// 更新数据
data: {
email: "zhangSan@weda.io",
},
}
```

```
// 筛选内容，筛选内容推荐使用编辑器数据筛选器生成
filter: {
where: {
$and: [
{
name: {
$eq: "juli",
},
},
],
},
},
},
});
console.log("请求结果", data); // { count: 3 }
} catch (e) {
console.log("错误代码", e.code, "错误信息", e.message);
}
}
```

添加关联关系

现有数据模型 sjmx_ftf41oj (数据模型) , glmx_b2z3oh7 (关联模型) 。

如果试图在 sjmx_ftf41oj 模型中添加一条数据关联 glmx_b2z3oh7 中的一条数据，数据标识为62fd7a3664d1e8ba00cc579835a1ded2。



文本	数据标识	创建时间	更新时间	所有人	创建人	修改人	操作
test	62fd7a3664d1e8ba...	2023-08-08 15:03:22	2023-08-08 15:03:22	administrator	administrator	administrator	查看 编辑 删除

```
export default async function ({event, data}) {
try {
const data = await $w.cloud.callDataSource({
dataSourceName: "sjmx_ftf41oj",
methodName: "wedaBatchUpdateV2",
params: {
// 更新数据
data: {
email: "zhangSan@weda.io",
relation: "62fd7a3664d1e8ba00cc579835a1ded2",
},
// 筛选内容，筛选内容推荐使用编辑器数据筛选器生成
filter: {
where: {
$and: [
{
_id: {

```

```
$eq: "4ebb756064c11bc9006e5d2e4f9b73a8", // 更新单条时，推荐传入_id数据标识进行操作
},
},
],
},
},
},
},
}),
console.log("请求结果", data); // { count: 1 }
} catch (e) {
console.log("错误代码", e.code, "错误信息", e.message);
}
}
```

删除 (wedaDeleteV2)

入参结构

属性	类型	默认值	示例	必填	说明
filter	{ where: FilterObject }]	无	{filter: {where: {}}}	是	复杂查询结构

② 说明:

- FilterObject 为复杂查询结构，请参考查询参数说明。
- 数据源筛选条件满足条数为 1 以上进行时进行单条更新会出错。

出参结构

属性	类型	默认值	示例	说明
count	count: 0 or 1	无	1	变更的条数，返回非 0 值代表更新成功
Count	Count: 0 or 1	无	1	变更的条数，返回非 0 值代表更新成功（后续废弃）

编辑器 Javascript 代码示例

```
export default async function ({event, data}) {
try {
const data = await $w.cloud.callDataSource({
dataSourceName: "sjmx_ftf41oj",
methodName: "wedaDeleteV2",
params: {
// 筛选内容，筛选内容推荐使用编辑器数据筛选器生成
filter: {
where: {
$and: [
{
_id: {
$eq: "4ebb756064c11bc9006e5d2e4f9b73a8", // 更新单条时，推荐传入_id数据标识进行操作
},
},
],
}
},
}
});
```

```
},
},
});

console.log("请求结果", data); // { count: 1 }
} catch (e) {
console.log("错误代码", e.code, "错误信息", e.message);
}
}
```

删除多条 (wedaBatchDeleteV2)

入参结构

属性	类型	默认值	示例	必填	说明
filter	{ where: FilterObject}]	无	{filter: {where: {}}}	是	复杂查询结构

② 说明:

FilterObject 为复杂查询结构, 请参考查询参数说明。

出参结构

属性	类型	默认值	示例	说明
count	count: 0 or 1	无	1	变更的条数, 返回非 0 值代表更新成功
Count	Count: 0 or 1	无	1	变更的条数, 返回非 0 值代表更新成功 (后续废弃)

编辑器 Javascript 代码示例

```
export default async function ({event, data}) {
try {
const data = await $w.cloud.callDataSource({
dataSourceName: "sjmx_ftf41oj",
methodName: "wedaBatchDeleteV2",
params: {
// 筛选内容, 筛选内容推荐使用编辑器数据筛选器生成
filter: {
where: {
$and: [
{
name: {
$eq: "juli", // 更新单条时, 推荐传入_id数据标识进行操作
},
},
],
},
},
},
},
});
console.log("请求结果", data); // { count: 3 }
} catch (e) {
console.log("错误代码", e.code, "错误信息", e.message);
}
```

```

    }
}

```

查询 (wedaGetItemV2)

入参结构

属性	类型	默认值	示例	必填	说明
filter	{ where: FilterObject}]	无	{filter: {where: {}}}	是	复杂查询结构
select	{ [key: string]: boolean }	无	{ \$master: true }	是	可以指定返回本表或者关联表的字段，如果想查询本表所有字段，请使用 { \$master: true }

② 说明：

FilterObject 为复杂查询结构，请参考查询参数说明。

出参结构

属性	类型	默认值	示例	说明
对应数据源字段结构	{ [key: string]: any }	无	{ name: "juli" }	返回满足筛选条件的数据源详情

编辑器 Javascript 代码示例

```

export default async function ({event, data}) {
  try {
    const data = await $w.cloud.callDataSource({
      dataSourceName: "sjmx_ftf41oj",
      methodName: "weda.GetItemV2",
      params: {
        // 筛选内容，筛选内容推荐使用编辑器数据筛选器生成
        filter: {
          where: {
            $and: [
              {
                _id: {
                  $eq: "f8f6930864c11fee007010104a2589c4", // 获取单条时，推荐传入_id数据标识进行操作
                },
                ...
              ],
              ...
            },
            ...
          },
          select: {
            $master: true, // 常见的配置，返回主表
          },
          ...
        });
        console.log("请求结果", data); // "[{"owner": "1559148626461061122", "createdAt": 1690378222467, "createBy": "1559148626461061122", "updateBy": "1559148626461061122", "name": "foo", "_id": "f8f6930864c11fee007010104a2589c4", "email": "bar@weda.io", "updatedAt": 1690378222467}]"
      } catch (e) {
    }
  }
}

```

```

    console.log("错误代码", e.code, "错误信息", e.message);
}
}

```

查询多条 (wedaGetRecordsV2)

入参结构

属性	类型	默认值	必填	说明
filter	{ where: FilterObject }]	无	是	复杂查询结构
select	{ [key: string]: boolean }	无	{ \$master: true }	是
getCount	{ \$master: true }	boolean	否	获取 filter 命中条件的查询条数
pageSize	false	number	否	分页大小, 建议指定
pageNumber	undefined	1	否	分页数目
orderBy	number	无	否	排序, 当前仅支持最多 3 字段排序

出参结构

属性	类型	默认值	示例	说明
{[key: string]: "acs" "desc" }[]	records	无	{ [key: string]: any }[]	records 内的数组为对应的数据源数据对象
{ name: "juli" }	total	无	number	入参配置了 3 为 getCount 时会返回满足筛选查询条件的大小, 注意, 此字段不代表返回 records 的长度, 可以用作页面大小的计算。当 true 为 getCount 的时候, 不应该对此值做任何期待

编辑器 Javascript 代码示例

```

export default async function ({event, data}) {
try {
const data = await $w.cloud.callDataSource({
dataSourceName: "sjmx_ftf4loj",
methodName: "wedaGetRecordsV2",
params: {
// 筛选内容, 当前筛选的含义为: 名字为 "juli" 或者 "foo"
filter: {
where: {
$or: [
{
name: {
$eq: "juli",
},
},
{
name: {
$eq: "foo",
},
}
]
}
}
}

```

```
],
},
},
// 排序
orderBy: [
{
createdAt: "asc", // 创建时间，正序
},
{
updateBy: "desc", // 更新时间，倒序
},
],
// 返回字段选择
select: {
$master: true, // 常见的配置，返回主表
},
// 返回total字段
getCount: true,
// 页面大小
pageSize: 10,
// 当前页面
pageNumber: 1,
},
});
console.log("请求结果", data); // "{records:[{"owner":"1559148626461061122","createdAt":1690378222467,"createBy":"1559148626461061122","updateBy":"1559148626461061122","name":"foo","_id":"f8f6930864c11fee007010104a2589c4","email":"bar@weda.io","updatedAt":1690378222467},{"owner":"1559148626461061122","createdAt":1690382002594,"createBy":"1559148626461061122","updateBy":"1559148626461061122","name":"juli","_id":"f95d024c64c12eb2006fd51d38654e28","email":"juli@weda.io","updatedAt":1690382002594}],"total":2}"
} catch (e) {
console.log("错误代码", e.code, "错误信息", e.message);
}
}
```

查询参数说明

逻辑运算符

名称	描述
\$and	使用逻辑 and 连接字段，返回与这两个字段条件匹配的数据
\$or	使用逻辑 or 连接字段，返回与任一个字段条件匹配的数据

示例1

```
{
"$and": [
{
"key": {
"$eq": "val"
}
}
]
```

```
]  
}
```

示例2

```
{  
  "$or": [  
    {  
      "$and": [  
        {  
          "key": {  
            "$eq": "val"  
          }  
        },  
        {  
          "key2": {  
            "$neq": 3  
          }  
        }  
      ]  
    },  
    {  
      "key3": {  
        "$eq": 0  
      }  
    }  
  ]  
}
```

比较运算符

名称	描述	适用类型
\$eq	匹配等于指定值的值	字符串, 布尔值, 数字
\$neq	匹配所有不等于指定值的值	字符串, 布尔值, 数字
\$gt	匹配大于指定值的值	数字
\$gte	匹配大于或等于指定值的值	数字
\$lt	匹配小于指定值的值	数字
\$lte	匹配小于或等于指定值的值	数字
\$in	匹配数组中指定的任何值	数组
\$nin	不匹配数组中指定的任何值	数组

示例1

```
{  
  "$and": [  
    {  
      "key": {  
        "$eq": "val"  
      }  
    }  
  ]  
}
```

```
}
```

```
}
```

```
]
```

```
}
```

示例2

```
{
  "$and": [
    {
      "key1": {
        "$in": [
          "foo",
          "bar"
        ]
      }
    },
    {
      "key2": {
        "$in": [
          1,
          2
        ]
      }
    }
  ]
}
```

特殊运算符

名称	描述	适用类型
\$search	模糊查询	字符串
\$nsearch	不包含，会把 null 值查找出来	字符串
\$eq-current-user	等于当前用户	字符串
\$ne-current-user	不等于当前用户	字符串
\$empty	数据为 null	任意类型
\$nempty	数据不为 null	任意类型

② 说明：

false、\$empty 在 mongo 中，会查询字段不存在和字段值为 null 的两种情况。

常见错误说明

标识[xxx]必填参数未填

xxx 字段被设置未必填属性时，如果参数的值没取到，就会出现这个错误。一般在返回的错误里有 errRecord 字段。例如 errRecord:{"foo":"bar"}。可以看到 errRecord 中不存在xxx字段。所以必填校验失败。

简单与复杂查询

最近更新时间：2023-08-07 08:14:48

数据模型方法中的 `where` 参数支持 **简单条件** 和 **复杂条件** 两种。本文将分别介绍两种查询的用法。

简单条件

`where` 简单条件模式需要传入一个对象数组，以数据模型查询 `wedaGetItem` 方法为例。

参数说明

参数名	说明	示例
key	数据模型字段标识	"firstname"
rel	判断条件	"eq": 相等, "neq": 不相等, "lt": 小于, "lte": 小于或等于, "gt": 大于, "gte": 大于或等于, "search": 包含文字内容
val	判断值，由字段数据类型决定	"张" 或 123
_id	数据模型数据标识	"95f2f2a963885c8900b0a415189efab7"

用法示例

示例一

```
module.exports = async function (params, context) {
  const result = await context.callModel({
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看
    methodName: 'wedaGetItem', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
    params: {
      where: [
        {
          key: "firstname",
          rel: "eq",
          val: "张",
        },
      ],
    },
  });

  // 在这里返回这个方法的结果，需要与出参定义的结构映射
  return result;
};
```

示例二

```
module.exports = async function (params, context) {
  const result = await context.callModel({
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看
    methodName: 'wedaGetItem', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
    params: {
      _id:"95f2f2a963885c8900b0a415189efab7"
    },
  });

  // 在这里返回这个方法的结果，需要与出参定义的结构映射
};
```

```
return result;
};
```

复杂条件

⚠ 注意：

当前仅 wedaGetRecords 支持复杂条件。

where 复杂条件模式同样需要传入一个对象数组。

参数说明

逻辑运算符

名称	描述
\$and	使用逻辑 and 连接字段，返回与这两个字段条件匹配的数据
\$or	使用逻辑 or 连接字段，返回与任一个字段条件匹配的数据

比较运算符

名称	描述
\$eq	匹配等于指定值的值
\$gt	匹配大于指定值的值
\$gte	匹配大于或等于指定值的值
\$in	匹配数组中指定的任何值
\$nin	不匹配数组中指定的任何值
\$lt	匹配小于指定值的值
\$lte	匹配小于或等于指定值的值
\$ne	匹配所有不等于指定值的值

特殊运算符

用法等同于比较运算符。

名称	描述
\$search	模糊查询
\$nsearch	不包含，会把 null 值查找出来
\$regex	正则匹配的值
\$eq-current-user	等于当前用户
\$ne-current-user	不等于当前用户
\$empty	数据为空
\$nempty	数据不为空

特殊符号

支持关联模型查询

名称	描述
\$related	关联模型查询条件
\$name	数据源标识
\$relatedType	关联类型： 1: 多对一 2: 一对多
\$relatedKey	关联字段
\$empty	关联字段是否为空。 1: 不为空 2: 为空
\$where	查询条件

- 多对一：A 的字段关联 B 表。多行 A 数据关联一个 B 数据。\$related 最多 join 2层。子表 join 父表。
- 一对多：A 的字段关联 B 表。对于 B 来说，一个 B 数据关联多个 A 数据。父表 join 子表。

用法示例

关联查询

```
module.exports = async function (params, context) {
  const result = await context.callModel({
    name: 'example', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看
    methodName: 'wedaGetRecords', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
    params: {
      where: [
        {
          "$and": [
            {
              "address": {
                "$eq": "上海"
              }
            },
            {
              "$related": {
                "$relatedType": 1,
                "$name": "data-123mnk23",
                "$relatedKey": "age",
                "$empty": "1",
                "$where": {
                  "$and": [
                    {
                      "name": {
                        "$eq": "张三"
                      }
                    }
                  ]
                }
              }
            }
          ],
        }
      ]
    }
  });
};
```

```
});

// 在这里返回这个方法的结果，需要与出参定义的结构映射
return result;
};
```

or 查询条件

```
{
  "$or": [
    {
      "address": {
        "$eq": "西安"
      }
    },
    {
      "address": {
        "$eq": "上海"
      }
    }
  ]
}
```

or 和 and 混合查询条件- and 的子条件

```
{
  "$or": [
    {
      "address": "武汉"
    },
    {
      "$and": [
        {
          "address": {
            "$eq": "上海"
          }
        },
        {
          "depName": {
            "$eq": "人事部"
          }
        }
      ]
    }
  ]
}
```

or 和 and 混合查询条件- or 的子条件， or 和 and 混合查询条件- 并列

```
{
  "$and": [
    {
      "address": "上海"
    },
    {
      "$or": [
        {
          "name": "张三"
        },
        {
          "name": "李四"
        }
      ]
    }
  ]
}
```

```
"depName": {"$eq": "上海研发部"}  
}, {  
"depName": {"$eq": "人事部"}  
}, {  
"depName": {"$eq": "武汉研发部"}  
}  
]  
]  
}
```

管理数据

最近更新时间：2023-08-07 18:01:40

微搭数据模型的数据管理后台，其定位是给企业、小程序开发者、服务商快速查看和管理数据的一套通用解决方案。

数据管理后台入口

腾讯云控制台进入微搭低代码数据源 > 数据模型页面，单击数据管理或每条数据模型操作列里的管理数据，会直接进入微搭数据管理后台页面。

数据管理

进入数据管理后台可以快速查看体验数据和正式数据，正式数据和体验数据不会互相干扰，体验环境仅显示体验数据，正式环境仅显示正式数据。

正式数据

1. 如果单击数据管理后台默认进入到数据模型列表，可以管理正式数据。

2. 默认从未发布过的数据模型，正式环境不存在对应数据，不能进行管理查看，会提示“该数据模型从未发布过，请先发布数据模型，再查看正式数据”，一旦正式环境发布过数据模型，就可以查看正式环境的数据。

3. 查看正式数据时，应用列表默认仅显示已经发布过正式环境的应用，发布到体验环境或者未发布的应用不在此展示。

- 单击对应应用，会默认显示当前应用所关联的全部数据模型。

名称	标识	类型	描述	所属应用	发布状态	创建时间	更新时间	操作
面试记录		自建数据源		招聘进度管理、面试 测试	已发布	2021-12-06 12:11:20	2021-12-06 12:14:39	管理数据
简历信息		自建数据源		招聘进度管理、面试 投递	已发布	2021-11-02 17:59:02	2021-11-02 18:08:25	管理数据
岗位信息		自建数据源		招聘进度管理、面试 投递	已发布	2021-11-02 13:20:47	2021-11-02 14:46:54	管理数据
招聘进度管理		自建数据源		招聘进度管理、面试 投递、选区测试	已发布	2021-11-02 13:01:42	2021-12-09 17:48:08	管理数据

- 单击管理数据，可以进入对应数据模型列表页，进行正式数据的增删查改。默认数据模型新增了字段，只有数据模型发布后，对应的页面才会自动显现出来新增的字段。

<input type="checkbox"/> 姓名	手机号	身份证号	核酸检测截图	截图姓名	采样结果	行程卡	行程卡颜色	操作
<input type="checkbox"/> 测试	18888888888	21020120101010...		测试	阴性		绿色	查看 编辑 删除

体验数据

1. 在页面的左上角可将正式数据切换到体验数据，可以查看对应数据模型的体验数据。

名称	标识	类型	描述	所属应用	发布状态	创建时间	更新时间	操作
面试记录		自建数据源	招聘进度管理、选区 测试	已发布	2021-12-06 12:11:20	2021-12-06 12:14:39		管理数据
简历信息		自建数据源	招聘进度管理、面试 投递	已发布	2021-11-02 17:59:02	2021-11-02 18:08:25		管理数据
岗位信息		自建数据源	招聘进度管理、面试 投递	已发布	2021-11-02 13:20:47	2021-11-02 14:46:54		管理数据
招聘进度管理		自建数据源	招聘进度管理、面试 投递、选区测试	已发布	2021-11-02 13:01:42	2021-12-09 17:48:08		管理数据

2. 查看体验数据时，应用列表默认会显示所有应用，包括发布到体验环境、正式环境的应用，以及未发布的应用均会在此展示。

我的应用 (共7条)

cds	招聘进度管理	面试投递	CRM销售管理系统
aaa	选区测试	简历	

- 单击对应应用，会默认显示当前应用所关联的所有数据模型。

招聘进度管理

名称	标识	类型	描述	所属应用	发布状态	创建时间	更新时间	操作
面试记录		自建数据源	招聘进度管理、选区 测试	已发布	2021-12-06 12:11:20	2021-12-06 12:14:39		管理数据
简历信息		自建数据源	招聘进度管理、面试 投递	已发布	2021-11-02 17:59:02	2021-11-02 18:08:25		管理数据
岗位信息		自建数据源	招聘进度管理、面试 投递	已发布	2021-11-02 13:20:47	2021-11-02 14:46:54		管理数据
招聘进度管理		自建数据源	招聘进度管理、面试 投递、选区测试	已发布	2021-11-02 13:01:42	2021-12-09 17:48:08		管理数据

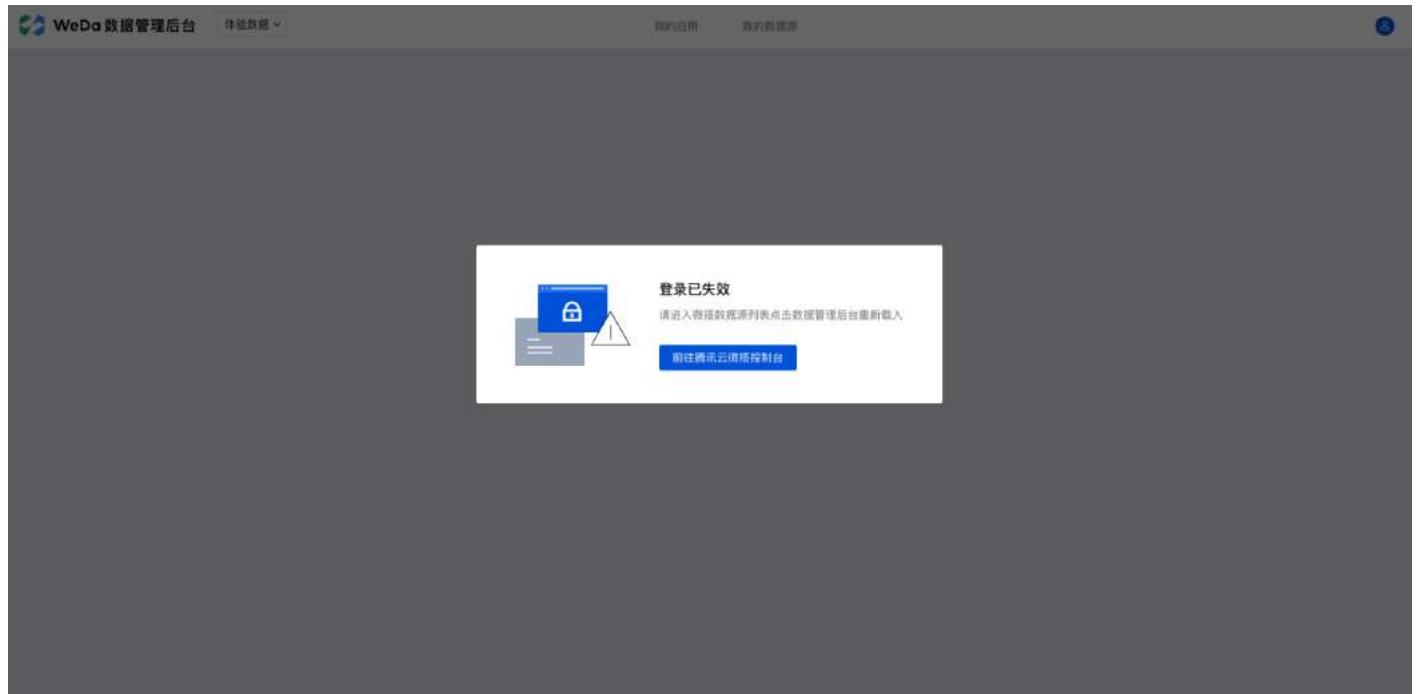
- 单击管理数据，可以进入对应数据模型列表页，进行体验数据的增删查改。默认数据模型只要新增字段，无需发布数据模型，对应的页面就会立即显现出来。

小程序openId	数据类型	访问时间	访问人数	数据标识	创建时间	更新时间	操作
<input type="checkbox"/> [REDACTED]	-	-	-	[REDACTED]	2022-04-28 21:20:39	2022-04-28 21:20:39	查看 编辑 删除
<input type="checkbox"/> [REDACTED]	-	-	-	[REDACTED]	2022-04-29 12:03:35	2022-04-29 12:03:35	查看 编辑 删除

数据管理后台功能失效解决方法

数据管理后台登录态失效后，可以单击前往腾讯云微搭控制台。

- 若腾讯云登录态失效，可以重新登录腾讯云账号，登录成功后直接进入腾讯云微搭控制台的数据模型列表页，可以再重新进入数据管理后台。
- 若腾讯云登录态存在，会直接进入腾讯云微搭控制台的数据模型列表页，可以再重新进入数据管理后台。



服务端表达式

最近更新时间：2023-07-27 15:11:05

功能介绍

表达式可以通过一些简单的语法（例如：单行语法）来实现动态的数据计算以及各类规则的判断，并且表达式引擎内部预先支持了各类的常见函数，例如：日期与时间函数、数学函数、文本函数等。

使用场景

数据模型

计算公式

假设已有如下数据模型：

模型字段

字段名称	字段标识	数据类型	是否必填	是否唯一 ①
姓氏	lastname	文本 短文本	否	否
名字	firstname	文本 短文本	否	否
单价	price	数字	否	否
数量	amount	数字	否	否

[展开系统字段](#) ▾

现在想在表格中显示姓名以及总价两个字段，分别是用姓氏+名字和单价*数量计算而来：

文档示例列表

姓氏	名字	姓名	单价	数量	总价	操作
张	三	张三	150	10	1500	查看 编辑 删除

共 1 条

10 条 / 页 1 / 1 页

我们可以利用计算公式类型字段来实现

- 姓名字段设置如下：

字段设置 ① ×

字段名称 *	姓名
字段标识 *	name
字段描述	请输入字段描述
数据类型 *	计算公式
结果类型 *	文本
计算表达式 *	#lastname + #firstname ***

确定 取消

确定后在表达式编辑器编辑：

```
#lastname + #firstname
```

- 总价字段设置如下：

字段设置 ① ×

字段名称 *	总价
字段标识 *	total
字段描述	请输入字段描述
数据类型 *	计算公式
结果类型 *	数字
计算表达式 *	#amount * #price ***

确定 取消

确定后在表达式编辑器编辑：

```
#amount * #price
```

② 说明：

表达式需要返回字段设置中结果类型相同的数据类型，例如：文本/数字/日期时间。

事件处理

数据模型 > 事件处理能设置数据模型事件（新增、删除和更新等）触发工作流，同时也支持设置执行前条件来判断在何种情况下才需要触发工作流。例如下图表示当单价 * 数量 > 10000 才触发工作流：

新建事件处理

触发时机 * 新建后

执行前置条件 `(#price * #amount) > 10000`

流程 * 表达式工作流

设置入参 param1 新增后值

提交 取消

编辑器内设置：

```
(#amount * #pric) > 10000
```

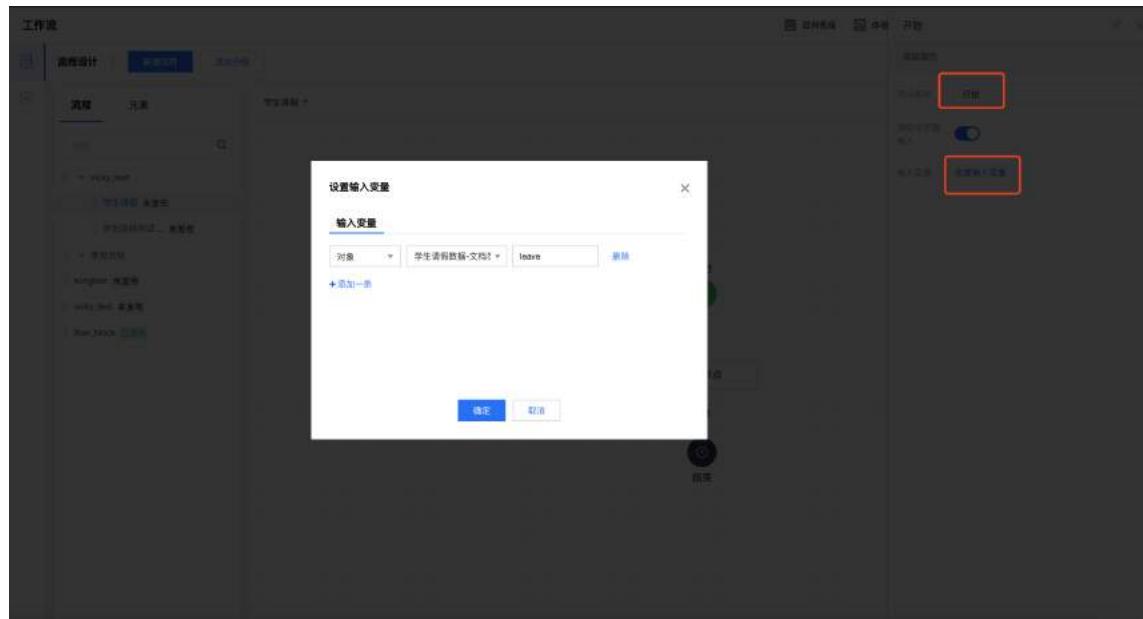
② 说明：

执行前条件需要表达式返回布尔值 Boolean 类型数据。

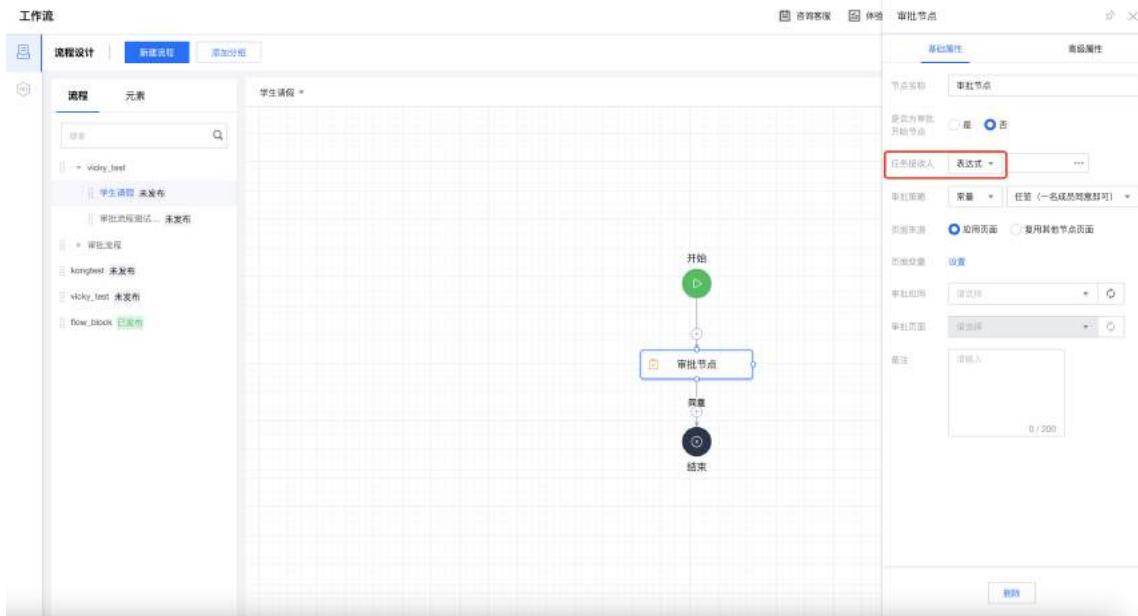
工作流

属性配置

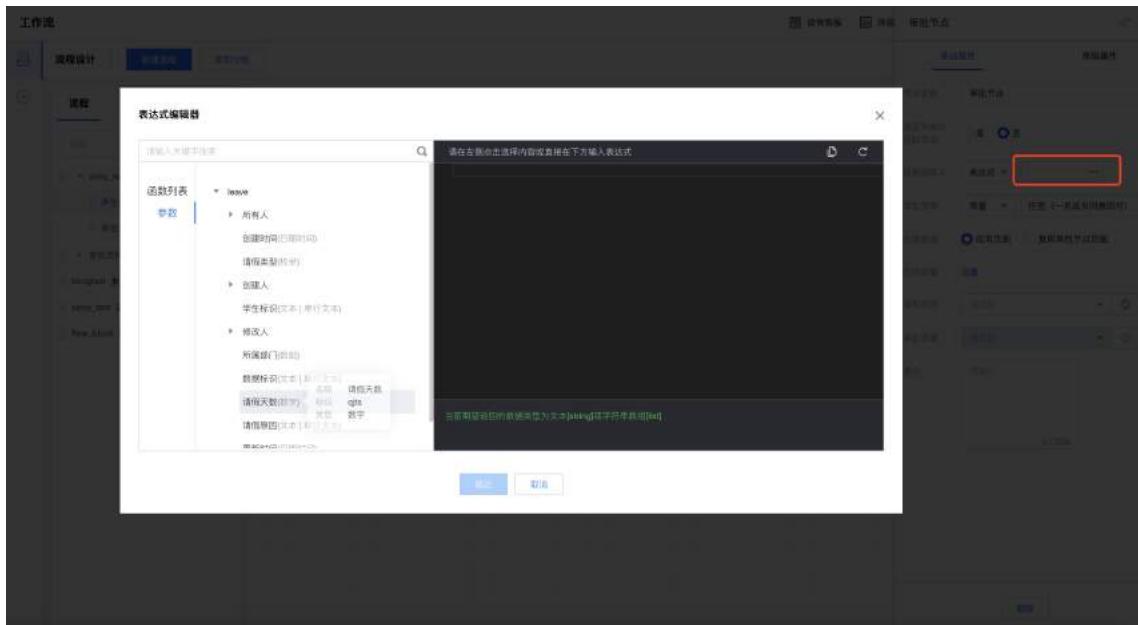
假设已有开始节点配置了输入变量的工作流。



如果想在审批节点中设定审批人为该输入变量中的某字段，将审批节点任务接收人类型选为表达式。



在表达式编辑器页面，从右侧参数列表可以选择流程中的变量，从右侧函数列表对变量进行函数公式处理。



表达式元素

具体可参见 [高级活动-表达式](#)。

基础语法

目前表达式仅支持单行表达式，可以进行数值、逻辑、变量提取、函数调用及混合调用。

② 说明:

表达式在语法校验及执行时会判断操作对象的类型，例如：大于号（>）会判断左右操作类型是否为布尔类型，如果不为布尔类型会提示类型错误。

支持的语法

大类	小类	运算符	示例
优先级	变更优先级	()	(1 + 2) * 3
数值计算	取负	-	-1

大类	小类	运算符	示例
乘法	*	$2 * 3$	-
除法	/	$1 / 2$	-
取模	%	$23 \% 2$	-
加法	+	$1 + 2$	-
减法	-	$1 - 2$	-
逻辑运算	大于	>	$1 > 2$
大于等于	\geq	$1 \geq 2$	-
小于	<	$3 < 99$	-
小于等于	\leq	$1 \leq -1$	-
等于	\equiv	$1 \equiv 1$	-
不等于	\neq	$1 \neq 2$	-
非	!	$!(2 > 1)$	-
与	$\&\&$	$(2 > 1) \&\& (3 < 8)$	-
或	$\ $	$(2 > 1) \ (3 < 8)$	-
文本运算	文本拼接	+	'hello' + 'world'
变量取值	基本变量取值	#	#date
变量属性取值	#	#Person.age	-
函数调用	内置函数调用	FuncName(params)	Length('abc')
混合调用	-	-	If(7 < Length('abc'), 'abc', '321')

支持的变量类型

类型	说明
数字类型	数值运算, 函数调用会使用到的类型
文本类型	文本运算, 函数调用会使用到的类型
布尔类型	逻辑运算, 函数调用会使用到的类型
日期时间类型	函数调用会使用到的类型
对象类型	变量取值, 函数调用会使用到的类型
数组类型	函数调用会使用到的类型

函数列表

时间日期

Date 使用年月日创建日期时间

函数签名: Date (数字, 数字, 数字)

函数说明: 根据输入的年月日数值返回一个日期时间类型的数据

返回值类型: 日期时间

使用示例: 创建 2021-5-17 号的日期类型

Date(2021, 5, 17)

DateText 日期时间格式化

函数签名: DateText (日期时间, 文本)

函数说明: 格式化日期时间类型为指定格式的字符串

返回值类型: 文本

使用示例: 指定创建日期格式为 YYYY-MM-DD HH:mm:ss。

```
DateText(#createdTime, 'YYYY-MM-DD HH:mm:ss')
```

Timestamp 获取时间戳

函数签名: Timestamp (日期时间)

函数说明: 根据输入的日期返回该日期的时间戳，精确到秒

返回值类型: 数字

使用示例: 获取当前时间的时间戳。

```
Timestamp(Now())
```

DateDiff 天数差

函数签名: DateDiff (日期时间, 日期时间)

函数说明: 返回两个日期中相差的天数

返回值类型: 数字

使用示例: 计算开始时间 (startDay) , 结束时间 (endDay) 相差的天数。

```
DateDiff(#startDay, #endDay)
```

SecondDiff 秒数差

函数签名: SecondDiff (日期时间, 日期时间)

函数说明: 返回两个日期中相差的秒数

返回值类型: 数字

使用示例: 计算开始时间 (startDay) , 结束时间 (endDay) 相差的秒数。

```
SecondDiff(#startDay, #endDay)
```

MinuteDiff 分钟差

函数签名: MinuteDiff (日期时间, 日期时间)

函数说明: 计算两个日期中间的相差的分钟部分

返回值类型: 数字

使用示例: 计算开始时间 (startDay) , 结束时间 (endDay) 相差的分钟数。

```
MinuteDiff(#startDay, #endDay)
```

HourDiff 小时差

函数签名: HourDiff (日期时间, 日期时间)

函数说明: 计算两个日期中间的相差的小时数

返回值类型: 数字

使用示例: 计算开始时间 (startDay) , 结束时间 (endDay) 相差的小时数。

```
HourDiff(#startDay, #endDay)
```

DateAdd 增加 X 天

函数签名: DateAdd (日期时间, 数字)

函数说明: 在传入的日期时间上增加 X 天, 支持负数

返回值类型: 日期时间

使用示例: 开始时间 (startDay) 减少一天。

```
DateAdd(#startDay,-1)
```

MonthAdd 增加 X 月

函数签名: MonthAdd(date1,number)

函数说明: 在传入的日期和时间上增加 X 月, 支持负数

返回值类型: 日期时间

使用示例: 在开始时间 (startDay) 基础上减少一个月。

```
MonthAdd(#startDay,-1)
```

YearAdd 增加 X 年

函数签名: YearAdd(date1,number)

函数说明: 在传入的日期和时间上增加 X 年, 支持负数

返回值类型: 日期时间

使用示例: 在开始时间 (startDay) 基础上减少一年。

```
YearAdd(#startDay,-1)
```

DateTimeValue 使用时间日期文本创建日期时间

函数签名: DateTimeValue(文本, 文本)

函数说明: 将日期时间文本根据指定格式转化为日期时间

返回值类型: 日期时间

使用示例: 将字符串"2021-12-1 1:9:12"转换为日期时间类型。

```
DateTimeValue('2021-12-1 1:9:12', 'yyyy-MM-dd HH:mm:ss')
```

DateValue 使用日期文本创建日期时间

函数签名: DateValue(文本, 文本)

函数说明: 将日期文本根据指定格式转化为日期时间

返回值类型: 日期时间

使用示例: 将字符串"2021-12-1"转换为日期时间类型。

```
DateValue('2021-12-1', 'yyyy-MM-dd')
```

IsToday 是否为今天

函数签名: IsToday(日期时间)

函数说明: 判断某个日期时间是否为今天

返回值类型: 布尔值

使用示例: 判断今天日期是否为 2020-1-1。

```
IsToday(Date(2020,1,1))
```

AgeOfNow 计算当前年龄

函数签名: AgeOfNow(日期时间)

函数说明: 计算用户当前的年龄

返回值类型: 数字

使用示例: 计算用户当前的年龄, 用户出生日期为 1980-3-24。

```
AgeOfNow(Date(1980,3,24))
```

Age 计算年龄

函数签名: Age(日期时间,日期时间)

函数说明: 计算用户在某一天的年龄

返回值类型: 数字

使用示例: 计算用户在 2021-3-24 日的年龄, 用户出生日期为 1980-3-24, 返回值为 31。

```
Age(Date(1980,3,24), Date(2021,3,24))
```

Now 获取现在时间

函数签名: Now()

函数说明: 返回当前时刻的时间

返回值类型: 日期时间

使用示例: 获取当前时间。

```
Now()
```

Second 获取秒数

函数签名: Second(日期时间)

函数说明: 根据输入的日期时间返回该时间的秒部分

返回值类型: 数字

使用示例: 返回当前时间的秒数。

```
Second(Now())
```

DateTime 使用时间戳创建日期时间

函数签名: DateTime(数字)

函数说明: 将数值类型的时间戳转化为日期时间类型

返回值类型: 日期时间

使用示例: 计算时间戳 12345 所在日期, 返回值为 1970-01-01 08:00:12。

```
DateTime(12345)
```

Year 获取年份

函数签名: Year(日期时间)

函数说明: 获取输入日期的年份, 范围为 1900~9999。

返回值类型: 数字

使用示例: 获取当前的年份。

Year(Now())

Minute 获取分钟数

函数签名: Minute(日期时间)

函数说明: 根据输入的日期时间返回该时间的分钟部分

返回值类型: 数字

使用示例: 返回当前时间的分钟数。

Minute(Now())

Hour 获取小时数

函数签名: Hour(日期时间)

函数说明: 根据输入的日期时间返回该时间的小时部分，24 小时制

返回值类型: 数字

使用示例: 返回当前时间的小时数。

Hour(Now())

Day 获取天数

函数签名: Day(Date)

函数说明: 获取输入日期的日，范围为 1–31

返回值类型: 数字

使用示例: 获取当天的日。

Day(Now())

Month 获取月数

函数签名: Month(日期时间)

函数说明: 获取输入日期的月份，范围为 1–12

返回值类型: 数字

使用示例: 获取当天的月份。

Month(Now())

::: YearDiff 年数差

函数签名: YearDiff(日期时间,日期时间)

函数说明: 计算两个日期中间的相差的年数

返回值类型: 数字

使用示例: 计算开始时间 (startDay)，结束时间 (endDay) 相差的年数。

YearDiff(#startDay, #endDay)

:::

::: MonthDiff 月数差

函数签名: MonthDiff(日期时间, 日期时间)

函数说明: 返回两个日期中相差的月数

返回值类型: 数字

使用示例: 计算开始时间 (startDay)，结束时间 (endDay) 相差的月数。

```
MonthDiff(#startDay, #endDay)
```

...

... DayOfWeek 获取星期数

函数签名: DayOfWeek(日期时间)

函数说明: 根据输入的日期时间返回该时间的星期数

返回值类型: 数字

使用示例: 返回当前时间的星期数。

```
DayOfWeek(Now())
```

计算

Rand 随机数

函数签名: Rand(数字)

函数说明: 获取一个伪随机数

返回值类型: 数字

使用示例: 获取一个 10 以内的随机数。

```
Rand(10)
```

Average 平均值

函数签名: AgeOfNow(数字,[数字...])

函数说明: 返回一组数据中的平均值

返回值类型: 数字

使用示例: 获取 1, 15, 100, 100.6 的平均值, 返回值为 54.15。

```
Average(1,15,100,100.6)
```

ABS 绝对值

函数签名: ABS(数字)

函数说明: 计算一个数值的绝对值

返回值类型: 数字

使用示例: 计算数值-9.7 的绝对值。

```
ABS(-9.7)
```

Min 最小值

函数签名: Min(数字, [数字...])

函数说明: 返回一组数据中的最小值

返回值类型: 数字

使用示例: 获取 1, 15, 100, 100.6 中的最小值, 返回值为 1。

```
Min(1,15,100,100.6)
```

Max 最大值

函数签名: Max(数字, [数字...])

函数说明: 返回一组数据中的最大值

返回值类型: 数字

使用示例: 获取 1, 15, 100, 100.6 中的最大值, 返回值为 100.6。

Max(1,15,100,100.6)

Sum 求和

函数签名: Sum(数字, [数字...])

函数说明: 获取一组数据中的求和

返回值类型: 数字

使用示例: 获取 1, 15, 100, 100.6 的和, 返回值为 216.6。

Sum(1,15,100,100.6)

Ceiling 向上取整

函数签名: Ceiling(数字)

函数说明: 获取一个数值的上取整, 及获取大于等于自身的最小整数

返回值类型: 数字

使用示例: 计算数值 7.1 的上取整, 返回值为 8。

Ceiling(7.1)

Floor 向下取整

函数签名: Floor(数字)

函数说明: 获取一个数值的下取整, 及获取小于等于自身的最大整数

返回值类型: 数字

使用示例: 计算数值 7.7 的下取整, 返回值为 7。

Floor(7.7)

Round 四舍五入

函数签名: Round(数字, 数字)

函数说明: 计算一个值的四舍五入结果, 也可使用第二个参数指定小数点后位数

返回值类型: 数字

使用示例: 计算 216.6 四舍五入的值, 返回值为 217。

Round(216.6)

文本

Regular 匹配正则

函数签名: Regular(文本, 正则文本)

函数说明: 判断文本类型是否符合指定的正则规则, 如果符合则返回 True, 不符合则返回 False。

返回值类型: 布尔值

使用示例:

Regular(#input.value, '^[0-9]*\$')

Ends 匹配结尾

函数签名: Ends(文本, 文本)

函数说明: 检查文本是否以特定字符或者字符串结尾, 如果包含则返回 True, 不包含则不返回 False

返回值类型: 布尔值

使用示例: 例如检查对象“订单”中的字段“订单号”是否以“SO”结尾。

```
Ends(#Order.order_id, 'SO')
```

Begins 匹配开头

函数签名: Begins(文本, 文本)

函数说明: 检查文本是否以指定文本开头, 如果包含则返回 True, 不包含则不返回 False。

返回值类型: 布尔值

使用示例: 校验上下文中的订单对象 (order) 的订单号属性 (no) 是否以 “SO” 开头。

```
Begins(#order.no, 'SO')
```

TrimStarts 删除开头空格/制表符

函数签名: TrimStarts(文本)

函数说明: 删除文本字符串开头的所有空格及制表符

返回值类型: 文本

使用示例: 删除文本" 腾讯科技 s"开头的空格。

```
TrimStarts(' 腾讯科技s')
```

TrimEnds 删除结尾空格/制表符

函数签名: TrimEnds(文本)

函数说明: 删除文本字符串结尾的所有空格及制表符

返回值类型: 文本

使用示例: 删除文本"腾讯科技 s "结尾的空格。

```
TrimEnds('腾讯科技s ')
```

Upper 转换为大写

函数签名: Upper(文本)

函数说明: 将字符串中的字符转化为全大写

返回值类型: 文本

使用示例: 将字符串"abc"转化为"ABC"。

```
Upper('abc')
```

SubString 文本截取

函数签名: SubString(文本, 数字, 数字)

函数说明: 根据输入位置截取部分文本

返回值类型: 文本

使用示例: 获取字符串"您好, 我的世界"中的"您好"。

```
SubString('您好, 我的世界', 0, 2)
```

Text 转换为文本

函数签名: Text(数字/日期时间)

函数说明: 将数字、日期时间等其他类型的数据转换为文本类型

返回值类型: 文本

使用示例: 上下文中订单对象 (order) 中有两个字段, 分别为数量 (quantity) 和 (unit), 需要将这两个字段进行拼接。

```
Text(order.quantity) + order.unit
```

Lower 转换为小写

函数签名: Lower(文本)

函数说明: 将字符串中的字符转化为全小写

返回值类型: 文本

使用示例: 将字符串"ABC"转化为"abc"。

```
Lower('ABC')
```

UUID 生成唯一标识

函数签名: UUID()

函数说明: 生成一个 UUID 文本

返回值类型: 文本

使用示例: 生成一个 UUID 文本。

```
UUID()
```

Len 获取文本长度

函数签名: Len(文本)

函数说明: 获取传入文本的字符数

返回值类型: 文本

使用示例: 计算文本“腾讯科技”所包含的字符数。

```
Len('腾讯科技')
```

Contains 是否包含指定文本

函数签名: Contains(文本, 文本)

函数说明: 检查文本是否包含指定文本, 如果包含则返回 True, 不包含则不返回 False

返回值类型: 布尔值

使用示例: 校验上下文中的商机报备对象 (opportunityRegister) 的行业属性 (profession) 是否包含“教育”。

```
Contains(#opportunityRegister.profession, '教育')
```

Split 拆分文本

函数签名: Split(文本, 文本)

函数说明: 根据指定字符串将原字符串拆分成字符串数组

返回值类型: 文本

使用示例: 用", "将字符串"您好,我的世界,我来了"拆分成字符串数组, 返回结果为["您好", "我的世界", "我来了"]。

```
Split('您好,我的世界,我来了', ',')
```

Trim 删除开头/结尾空格和制表符

函数签名: Trim(文本)

函数说明: 删除文本开头和结尾的所有空格和制表符, 文本中间的空格和制表符不会删除

返回值类型: 文本

使用示例: 删除文本“腾讯科技 s”前后的空格。

```
Trim('腾讯科技 s')
```

Value 转换为数字

函数签名: Value(文本)

函数说明: 将文本类型转化为数字类型

返回值类型: 数字

使用示例: 将字符串 “12.34567” 转化为数值 12.34567

```
Value('12.34567')
```

逻辑

If 条件分支

函数签名: If(判断条件, 满足时返回的值, [不满足时返回的值])

函数说明: 按判断条件进行逻辑比较, 满足时返回一个值, 不满足时返回另一个值

返回值类型: 不定类型

使用示例: 判断 1 是否等于 1, 正确返回"您好", 错误返回"hello", 结果返回"您好"。

```
If(1==1, '您好', 'hello')
```

Equals 是否相等

函数签名: Equals(值1, 值2)

函数说明: 判断两个值是否相等

返回值类型: 布尔值

使用示例: 判断上下文中的对象 1 (object1) 和对象 2 (object2) 是否相等。

```
Equals(object1, object2)
```

IsNumber 是否为数字

函数签名: IsNumber(文本)

函数说明: 判断输入文本是否为数值, 如果为数字则返回 True, 不为数字则返回 False。

返回值类型: 布尔值

使用示例: 判断文本 “123” 是否为数值文本。

```
IsNumer('123')
```

IsNull 是否为空 (Null)

函数签名: IsNull(值)

函数说明: 判断输入对象是否为空

返回值类型: 布尔值

使用示例: 判断上下文中订单对象 (order) 是否为空。

```
IsNull(#order)
```

集合函数

Size 获取个数

函数签名: Size(数组)

函数说明: 计算数组中元素个数

返回值类型: 数字

使用示例: 上下文中存在数组 persionList，计算 persionList 的元素数量。

```
Size(#persionList)
```

Merge 合并对象/数组

函数签名: Merge(对象/数组, 对象/数组)

函数说明: 合并两个对象或两个数组为一个数组

返回值类型: 数组

使用示例: 合并对象 1 (#object1)、对象 2 (#object2) 成一个列表。

```
Merge(#object1, #object)
```

APIs

开放服务

API 概述

最近更新时间：2023-02-24 16:00:20

腾讯云微搭低代码最新上线 API（开放服务）能力，可以打通外部应用数据或调用外部应用开放的服务，例如：腾讯文档、腾讯会议、腾讯地图、微信支付、腾讯云文字识别、腾讯云短信等。微搭开发者只需要创建 API 时完成授权或少量填写一些配置信息，即可使用对应产品的服务接口，而无需操心这些接口的实现细节。

新建 API（开放服务）

新建开放服务仅需进行部分配置，即可实现API连接，详情请参见 [新建 API](#)。

使用API（开放服务）

在应用编辑器、自定义数据模型和自定义 API 中，都可以使用 API，支持可视化和自定义代码两种方式。详情请参见 [使用和管理 API](#)。

新建 API

最近更新时间：2023-05-08 11:12:59

腾讯云微搭低代码最新上线 API（开放服务）能力，可以打通外部应用数据或调用外部应用开放的服务。本文介绍如何创建 API（开放服务）。

下面将以腾讯文档作为示例：

操作步骤

步骤1：选择要创建的 API

从腾讯云微搭低代码控制台，进入 [数据源 > APIs 列表页](#)，选择新建 APIs，选择开放服务 > 腾讯文档。

步骤2：单击确定开始授权



步骤3：登录并授权

1. 跳转到授权窗口，进行安全身份登录。单击同意。
2. 登录成功后，需要同意授权腾讯云微搭应用使用腾讯云文档权限。

步骤4：填写 API 配置

自定义APIs

名称 * weda 名称不能为空

标识 * 48nfzz7

描述

② 说明：

标识为 API 唯一标识，在微搭应用编辑器、自定义代码中均需要借助这个标识来使用。

使用和管理 API

最近更新时间：2023-04-13 10:36:36

本文将介绍如何管理和使用 API（开放服务）。

编辑 API

有两种方式可以对 API 进行编辑：

- 在 [API](#) 页面，找到编辑的 API，在操作列单击编辑。
- 在 [API 详情](#) 页面，单击右上角编辑。

重新授权 API

在 [API](#) 页面，若在状态列出现鉴权失败，则可以单击重新授权重新发起授权。

```
[](id:delete)
```

删除 API

在 [API](#) 列表页面，找到编辑的 API，在操作列单击更多 > 删除。



注意：

API 删除前，必须解绑在所有应用的关联使用。

使用 API

目前在应用编辑器、自定义数据模型和自定义 API 中，都可以使用 API，支持可视化和自定义代码两种方式。

通过组件行为调用

进入应用编辑器，在组件行为可以调用 API 方法，具体可参见 [添加行为事件](#)。

通过低代码编辑器调用

进入应用编辑器，可以在应用的 [代码编辑器](#) 及微搭组件的组件代码中使用。具体可参见 [自定义方法](#)。

示例代码：

```
export default async function({event, data}) {
  const result = await app.cloud.callConnector({
    name: 'API 标识',
    methodName: '方法标识',
    params: {}, // 方法入参
  });
}
```

在自定义数据模型或自定义 API 中调用

在自定义数据模型和自定义 API 中通过 [自定义代码](#) 方式，可以调用其他 API，具体可参见 [自定义代码（云函数）](#)。

示例代码：

```
module.exports = async function (params, context) {
  const result = await context.callConnector({
    name: 'API 标识',
    methodName: '方法标识',
  });
}
```

```
params: {}, // 方法入参
});

return {
_id: '123456',
};

};
```

腾讯文档

最近更新时间：2022-09-28 10:22:22

[腾讯文档开放平台](#) 面向第三方应用提供了一套基于 HTTP 协议的 Open API，开发者可以通过 Open API 对用户的文档资源进行访问和操作。在腾讯云微搭，开发者只需要完成登录授权即可创建[腾讯文档 API](#) 来使用腾讯文档提供的 API，无需关心接口鉴权。

API 方法

腾讯文档 API 支持以下方法：

方法	方法标识	方法描述	接口文档
添加子表	addSheet	在表格的某个位置添加一个工作表	查看
删除多行(列)	deleteDimension	删除工作表的多行或者多列	查看
清空区域内容	clearArea	清空区域内的所有单元格的内容	查看
更新区域内容	updateArea	更新区域内的所有单元格的内容	查看

腾讯会议

最近更新时间：2022-09-28 10:22:07

腾讯会议提供的 Rest API 是为参与腾讯会议生态系统建设的合作方开发者接入并访问腾讯会议资源提供的一组工具，是访问腾讯会议 SaaS 服务的入口。开发者可以通过腾讯会议 API 进行二次开发，例如创建一个会议，修改会议，查询会议信息等。腾讯云微搭提供了腾讯会议 API，帮助开发者在应用中方便的使用腾讯会议开放能力。

API方法

腾讯会议 API 支持以下方法：

方法	方法标识	方法描述	接口文档
创建会议	createMeeting	快速创建/预定一个会议	查看
取消会议	cancelMeeting	取消用户创建的会议。用户只能取消自己创建的会议，且该会议是一个有效的会议。如果不是会议创建者或者无效会议号将会返回错误	查看
查询会议	getMeetingDetail	通过会议 ID 查询会议详情	查看
修改会议	modifyMeeting	修改某指定会议的会议信息	查看
获取参会成员列表	getMeetingMemberList	会议拥有者获取某指定会议的参与人员列表。只有会议的拥有者即创建者可以查询参会成员的列表，其他用户的调用会被拒绝。如果会议还未开始，调用此接口查询会返回空列表	查看
查询用户会议列表	getMeetingList	获取某指定用户的进行中或待开始的会议列表	查看
查询用户已结束会议列表	getEndMeetingList	获取某指定用户的已结束的会议列表，可返回用户创建与参加过的已结束会议列表	查看

腾讯地图

最近更新时间：2023-07-12 17:15:24

腾讯地图 API 是基于 [腾讯地图 WebService API](#) 提供的一系列服务接口，配合腾讯地图组件可以实现更高级的地图能力。

⚠ 注意

若要创建腾讯地图 API，需要注册腾讯位置服务账号且创建一个 API KEY，[前往获取](#)。

API 方法

腾讯地图 API 支持以下方法：

方法	方法标识	方法描述	接口文档
获取腾讯地图 API Key	getApiKey	获取腾讯地图 API Key	-
坐标位置描述	coord2label	本接口提供由经纬度到文字地址及相关位置信息的转换能力，广泛应用于物流、出行、O2O、社交等场景	查看
批量距离矩阵计算	distanceMatrix	距离矩阵 (DistanceMatrix)，用于批量计算一组起终点的路面距离（或称导航距离），可应用于网约车派单、多目的地最优路径智能计算等场景中，支持驾车、步行、骑行多种交通方式，满足不同应用需要	查看
获取下级行政区划	getDistrictChildren	获取指定行政区划的子级行政区划	查看
获取省市区列表	getDistrictList	本接口用于获取全部省市区三级行政区划	查看
行政区划搜索	searchDistrict	本接口用于获取全部省市区三级行政区划	查看
IP 定位	ipLocation	实现从其它地图供应商坐标系或标准 GPS 坐标系，批量转换到腾讯地图坐标系	查看
地址转坐标	label2coord	本接口提供由文字地址到经纬度的转换能力，并同时提供结构化的省市区地址信息	查看
骑行路线规划	routeBicycling	腾讯地图 Direction API，提供多种交通方式的路线计算能力。此接口提供基于自行车的骑行路线	查看
驾车路线规划	routeDriving	腾讯地图 Direction API，提供多种交通方式的路线计算能力。支持结合实时路况、少收费、不走高速等多种偏好，精准预估到达时间 (ETA)	查看
公交路线规划	routeTransit	腾讯地图 Direction API，提供多种交通方式的路线计算能力。支持公共汽车、地铁等多种公共交通工具的换乘方案计算	查看
步行路线规划	routeWalking	腾讯地图 Direction API，提供多种交通方式的路线计算能力，基于步行路线规划。	查看
货车路线规划	routeTrucking	货车 (trucking) 路线规划，基于道路客观条件产生的宽高重限制和行政部门规定的限制，结合实时路况、少收费、不走高速、躲避拥堵等多种偏好，对货车路线进行规划。	查看
货车距离矩阵计算	trucklingMatrix	支持针对货车的大批量和多种计算方式	查看
地点搜索	searchPlace	提供多种搜索功能(城市/区域/周边/矩形区域/周边推荐/ID查POI信息)	查看
智能地址解析	smartCoder	地址解析 (又称地理编码)，提供由文字地址到经纬度的转换能力，并同时提供结构化的省市区地址信息	查看

方法	方法标识	方法描述	接口文档
关键词输入提示	suggestion	用于获取输入关键字的补充与提示，帮助用户快速输入，本接口为纯数据接口	查看
坐标转换	translateCoords	实现从其它地图供应商坐标系或标准 GPS 坐标系，批量转换到腾讯地图坐标系	查看

微信支付

最近更新时间：2023-04-12 11:02:33

微信支付 API 提供了 [微信支付](#) 相关接口和回调，包括统一下单、查询订单、关闭订单、下载对账单等。

⚠ 注意：

所有接口均需要在小程序内使用。

新建微信支付 API

准备工作

在创建微信支付 API 之前，需要准备好一个非个人主体微信小程序及微信支付普通商户号（小商店注册的商户号不可用），并且两者之前需要提前做好绑定。相关帮助文档：[通过微搭快速注册小程序 / 获取商户号 / 商户号绑定 AppID](#)。

⚠ 注意

- 微信支付商户号的超级管理员需要关注[微信支付商家助手公众号](#)。
- 微搭 APIs 支付使用微信支付服务商模式实现，使用前请查看[交易类小程序运营规范](#) 和 [小程序交易结算管理确认说明](#)，确保小程序符合使用条件。

步骤1：通过微搭小程序开发认证

前往微搭 [小程序开发认证](#) 并根据流程指引完成小程序认证。

步骤2：新建 API

⚠ 注意：

选择商户号并进行绑定时，微信商户号的超级管理员会收到确认授权消息，需要超管确认授权。

前往 [API 列表页](#) 单击新建 API，选择微信支付：



步骤3：授权 API 权限

1. 在发起 JSAPI 和退款 API 授权后，单击去微信支付授权链接。



△ 注意

发起授权后，微信支付超级管理员会收到微信支付商家助手公众号推送的模板消息，根据提示进行授权。

2. 前往 [微信支付商户平台](#) 完成授权。

The screenshot shows the 'WeChat Pay Merchant Platform' interface under the 'Product Center'. In the 'My Authorized Products' section, it lists various payment methods. For each method, there is a 'Status' column (e.g., 'Not Authorized') and an 'Authorization' button. Two specific buttons are highlighted with red boxes: one for 'JSAPI Payment' and another for 'Refund API Refund'. These buttons correspond to the ones in the previous screenshot.

产品名称	授权状态	操作
服务商现金红包	未授权	授权 查看协议
服务商分账	未授权	授权 查看协议
服务商JSAPI支付	未授权	授权
服务商H5支付	未授权	授权
服务商APP支付	未授权	授权
服务商付款码支付	未授权	授权
服务商平台退款	未授权	授权 查看协议
服务商API退款	未授权	授权 查看协议
服务商免充值代金券	未授权	授权 查看协议
服务商免充值立减折扣	未授权	授权 查看协议
服务商刷脸支付	未授权	授权

② 说明：

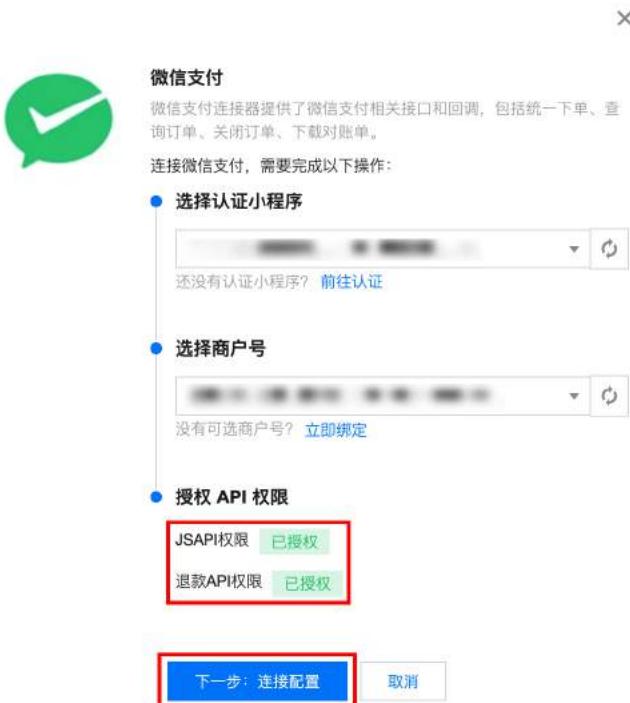
若在等待授权期间关闭了创建窗口，可再次通过新建 API 并选择相同的小程序及商户号即可随时查看当前授权状态。

⚠ 注意

如果没有授权的产品目录，请检查微信支付商户号的超级管理员是否已经关注微信支付商家助手公众号。

步骤4：API 配置

完成 步骤3 的授权后，单击下一步：连接配置。



进入 API 配置信息：

根据微信支付的业务逻辑，统一下单接口需要有一个回调处理方法，当用户完成支付后微信支付侧会通过此方法通知支付结果以便进行下一步业务逻辑。可参见[微信支付逻辑](#)。微搭提供自动创建示例或选择已有云函数两种方式方便开发者选用。

步骤5：完成新建

完成新建后，可以在 API 详情页查看具体的 API 的基本信息、API 方法及描述：

连接器方法

参数	字段名称	字段标识	数据类型	是否必填
设备号 ①	deviceInfo	文本 短文本	否	
商品描述 ①	body	文本 短文本	是	
商品详情 ①	detail	文本 短文本	否	
附加数据 ①	attach	文本 短文本	否	
商户订单号 ①	outTradeNo	文本 短文本	是	
货币类型 ①	feeType	文本 短文本	否	
总金额 ①	totalFee	数字	是	
终端IP ①	spbillCreateIp	文本 短文本	否	
交易起始时间 ①	timeStart	文本 短文本	否	
交易结束时间 ①	timeExpire	文本 短文本	否	

使用微信支付 API

微信支付 API 和其他 API 功能相同，可在应用编辑器中使用，以下是一个示例：

创建支付按钮和行为

1. 设置按钮的点击行为：

添加事件: 点击时 (tap)

当点击时 (tap) 时

调用触控源方法

成功时

失败时

Tips: 保持Hover上方条件可继续执行动作

名称: 微信支付 | init_order

方法名: 统一下单(unifiedOrder)

入参: { body: "微信充值", outTradeNo: ... }

出参: 保存为: 请填写变量名

显示加载:

显示提示:

2. 传入参数可设置统一下单方法中需要的参数：

The screenshot shows a code editor window with the following content:

```
请在左侧点击选择内容或直接在下方输入表达式
1  [
2   body:"微搭充值",
3   outTradeNo:"xxxxxxxx_xxxxxx",
4   totalFee:200,
5 ]]
```

Below the code editor, a message indicates: "当前期望返回的数据类型为 文本[string]" (The expected return data type is Text [string]).

发起微信支付

1. 设置统一下单接口成功返回后的自定义方法：

The screenshot shows the configuration of a success callback method for WeChat Pay. The configuration details are as follows:

- 事件：点击时 (tap)
- 方法类型：已有方法 (Existing Method)
- 执行动作：payment
- 入参：event.detail
- 出参：保存为 (Save as)：请输入变量名

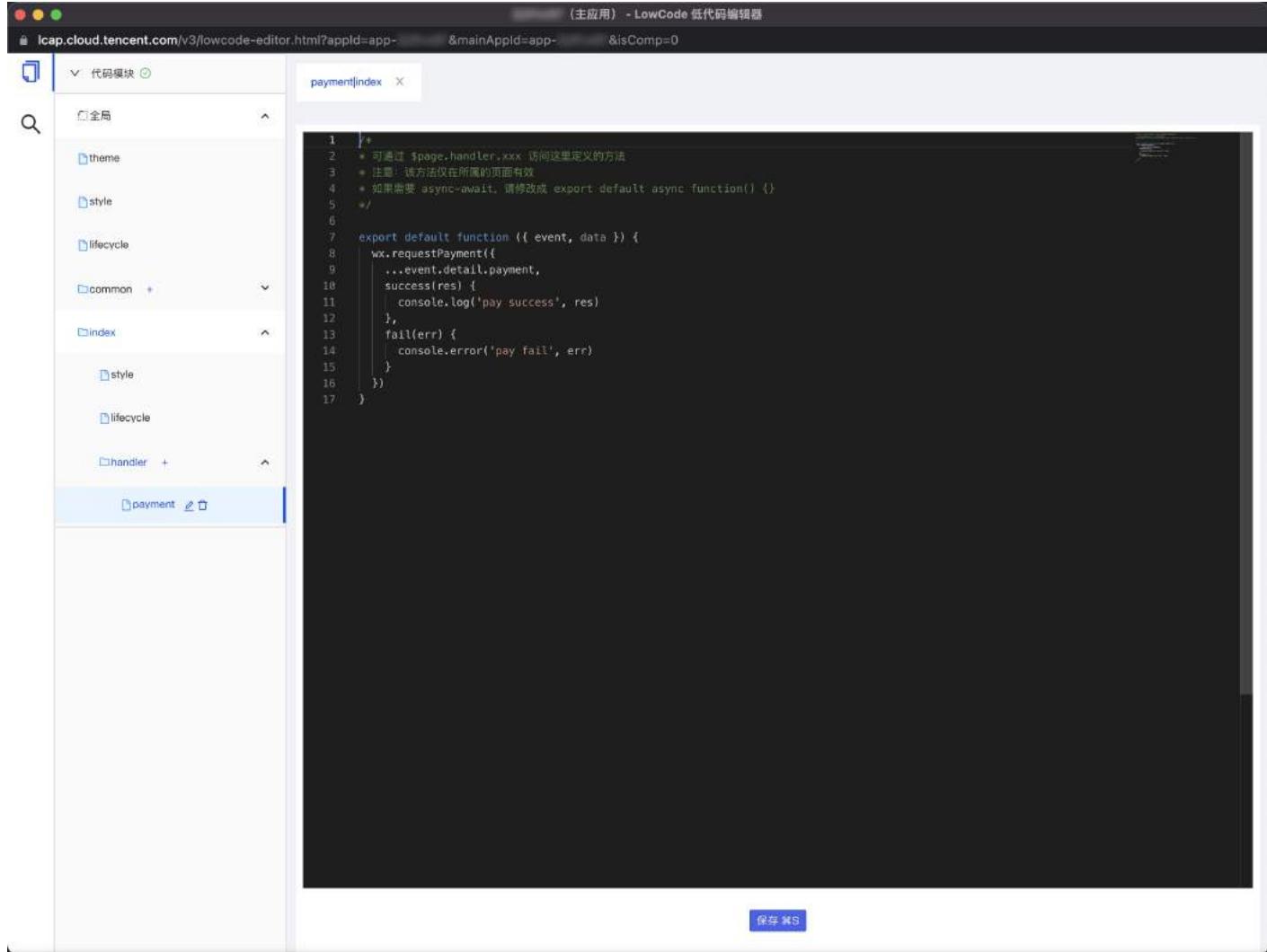
传入参数绑定表达式：

The screenshot shows a code editor window with the following content:

```
请在左侧点击选择内容或直接在下方输入表达式
1 event.detail
```

Below the code editor, a message indicates: "当前期望返回的数据类型为 文本[string]" (The expected return data type is Text [string]).

2. 在该方法中使用微信支付 API:



```
1  /*
2   * 可通过 $page.handler.xxx 访问这里定义的方法
3   * 注意：该方法仅在所属的页面有效
4   * 如果需要 async-await, 请修改成 export default async function() {}
5   */
6
7  export default function ({ event, data }) {
8    wx.requestPayment({
9      ...event.detail.payment,
10      success(res) {
11        console.log('pay success', res)
12      },
13      fail(err) {
14        console.error('pay fail', err)
15      }
16    })
17 }
```

```
export default function ({ event, data }) {
wx.requestPayment({
...event.detail.payment,
success(res) { // 在这里处理用户支付成功的逻辑
console.log('pay success', res);
},
fail(err) { // 在这里处理用户支付失败的逻辑
console.error('pay fail', err);
},
});
};
```

⚠ 注意：

由于微信支付是异步行为，因此用户完成支付后，微信支付侧会调用统一下单方法指定的回调云函数，详情可参见 [微信支付结果回调云函数协议](#)。

至此，微信支付 APIs 服务配置已完成。更多微信支付相关使用方法，请参见 [微信支付功能模板](#)。

API 方法

微信支付 API 支持以下方法：

方法	方法标识	方法描述	接口文档
统一 下单	unifiedOrder	商户在小程序中先调用该接口在微信支付服务后台生成预支付交易单，返回正确的预支付交易后调起支付。此接口与微信支付原接口（文档）的不同点在于： * 私有安全链路，免证书管理，免签名计算 * 商户号填入 subMchId 字段，小程序/公众号 appid 填入 sub_appid 字段 * 免填写以下字段：mch_id、appid、sign、sign_type * 接口入参和返回值都为 JSON 而不是 XML。	查看
查询 订单	queryOrder	该接口提供所有微信支付订单的查询，商户可以通过该接口主动查询订单状态，完成下一步的业务逻辑。	查看
关闭 订单	closeOrder	以下情况需要调用关单接口：商户订单支付失败需要生成新单号重新发起支付，要对原订单号调用关单，避免重复支付；系统下单后，用户支付超时，系统退出不再受理，避免用户继续，请调用关单接口。 注意： 订单生成后不能马上调用关单接口，最短调用时间间隔为5分钟。	查看
下载 对账 单	downloadBill	商户可以通过该接口下载历史交易清单。例如掉单、系统错误等导致商户侧和微信侧数据不一致，通过对账单核对后可校正支付状态。	查看

腾讯云文字识别

最近更新时间：2023-04-13 10:37:09

腾讯云文字识别 API 提供了 [腾讯云文字识别](#) 相关接口和回调，包括健康码识别、通信行程卡识别、通用印刷体识别、智能结构化识别等。

新建腾讯云文字识别 API

步骤1：准备工作

在创建腾讯云文字识别 API 之前，需要创建 [腾讯云 API 访问密钥](#)，并开通 [腾讯云文字识别](#) 产品服务。

 注意

腾讯云文字识别产品需要额外购买和开通。

步骤2：新建腾讯云文字识别 API

前往 [APIs](#) 单击新建 APIs，选择 [腾讯云文字识别](#)，并填入上一步创建的腾讯云密钥。

步骤3：完成新建

完成新建后，可以在 API 详情页查看具体的 API 的基本信息、API 方法及描述。

使用腾讯云文字识别 API

腾讯云文字识别 API 和其他 API 功能相同，可在应用编辑器中使用，以下是一个示例：

添加图片上传组件和行为

1. 设置组件的事件 `success`（[上传成功](#)），设置指定动作为 `Javascript 代码`。

2. 选择一个已有的方法或添加一个新方法并保存，方法的代码为：

```
export default async function({event, data}) {
  // getTempFileURL 用来获取上传文件的临时访问地址
  return await app.cloud.getTempFileURL(event.detail.value);
}
```

3. 继续设置 `IF 成功事件`，选择 [调用数据源方法](#)并且选中前面创建的腾讯云文字识别 API 及需要使用的方法。选择对应的参数设置表达式 `event.detail`。

4. 最终文字识别接口返回的数据在 `cloud.callDataSource` 成功的事件中返回，其返回值在 `event.detail.Response` 中，您可以将其赋值给一个变量。

API 方法

腾讯云文字识别 API 支持以下方法：

方法	方法标识	方法描述	接口文档
健康码识别	RecognizeHealthCodeOCR	本接口支持北京、上海、广东、江苏、吉林、黑龙江、天津、辽宁、浙江、河南、四川、贵州、山东、安徽、福建、江西、湖北、湖南等省份健康码的识别，包括持码人姓名、持码人身份证号、健康码更新时间、健康码颜色、核酸检测结果、核酸检测间隔时长、核酸检测时间，疫苗接种信息，八个字段的识别结果输出。不同省市健康码显示的字段信息有所不同，上述字段的识别结果可能为空，以图片上具体展示的信息为准。	健康码识别
通信行程卡识别	RecognizeTravelCardOCR	本接口支持通信大数据行程卡识别，包括行程卡颜色、更新时间、途经地、存在中高风险地区的城市、电话号码，五个字段的识别结果输出。	通信行程卡识别

方法	方法标识	方法描述	接口文档
通用印刷体识别 (高精度版)	GeneralAccurateOCR	本接口支持图像整体文字的检测和识别。支持中文、英文、中英文、数字和特殊字符的识别，并返回文字框位置和文字内容。适用于文字较多、版式复杂、对识别准确率要求较高的场景，如试卷试题、网络图片、街景店招牌、法律卷宗等场景。与通用印刷体识别相比，提供更高精度的文字识别服务，在文字较多、长串数字、小字、模糊字、倾斜文本等困难场景下，高精度版的准确率和召回率更高。	通用印刷体识别 (高精度版)
通用印刷体识别	GeneralBasicOCR	本接口支持图像整体文字的检测和识别。可以识别中文、英文、中英文、日语、韩语、西班牙语、法语、德语、葡萄牙语、越南语、马来语、俄语、意大利语、荷兰语、瑞典语、芬兰语、丹麦语、挪威语、匈牙利语、泰语，阿拉伯语20种语言，且各种语言均支持与英文混合的文字识别。	通用印刷体识别
智能结构化识别	SmartStructuralOCR	本接口支持识别并提取各类证照、票据、表单、合同等结构化场景的字段信息。无需任何配置，灵活高效。适用于各类结构化信息录入场景。	智能结构化识别

腾讯云短信

最近更新时间：2023-04-13 10:37:14

腾讯云短信 API 提供了 [腾讯云短信](#) 相关接口，目前支持发送短信接口。

新建腾讯云短信 API

步骤1：准备工作

在创建腾讯云短信 API 之前，需要创建 [腾讯云 API 访问密钥](#)，并开通 [腾讯云短信](#) 产品服务。因为相关法律法规的要求，您还需要根据指引完成 [短信签名](#) 的创建与审核后，方可发送短信。

 注意

腾讯云短信产品需要额外购买和开通。

步骤2：新建腾讯云短信 API

前往 [APIs](#) 单击新建 APIs，选择腾讯云短信，并填入上一步创建的腾讯云密钥。

步骤3：完成新建

完成新建后，可以在 API 详情页查看具体的 API 的基本信息、API 方法及描述。

使用腾讯云短信 API

腾讯云短信 API 和其他 API 功能相同，可在应用编辑器中使用，以下是一个示例：

例如在编辑器中设置组件的事件 [点击 \(tap\)](#)，设置指定动作选择调用数据源方法并且选中前面创建的腾讯云短信 API 及需要使用的方法。最后选择对应的参数设置即可，具体参数请参考下文 API 方法中的接口文档。

API 方法

腾讯云短信 API 支持以下方法：

方法	方法标识	方法描述	接口文档
发送短信	SendSms	短信发送接口，用于给用户发短信验证码、通知类短信或营销短信。	发送短信

腾讯云邮件

最近更新时间：2023-04-13 10:37:20

腾讯云邮件 API 提供了 [腾讯云邮件推送](#) 相关接口，目前支持发送邮件接口。

新建腾讯云邮件 API

步骤1：准备工作

在创建腾讯云邮件 API 之前，需要创建 [腾讯云 API 访问密钥](#)，并开通 [腾讯云邮件推送](#) 产品服务。因为相关法律法规的要求，您还需要根据指引完成邮件服务的创建与审核后，才可发送邮件，具体请参见 [腾讯云邮件推送概览页](#) 中的快速入门指引。

 注意

腾讯云邮件产品需要额外购买和开通。

步骤2：新建腾讯云邮件 API

前往 [APIs](#) 单击新建 APIs，选择腾讯云邮件，并填入上一步创建的腾讯云密钥。

步骤3：完成新建

完成新建后，可以在 API 详情页查看具体的 API 的基本信息、API 方法及描述。

使用腾讯云邮件 API

腾讯云邮件 API 和其他 API 功能相同，可在应用编辑器中使用，以下是一个示例：

例如在编辑器中设置组件的事件点击 (tap)，设置指定动作选择调用数据源方法并且选中前面创建的腾讯云邮件 API 及需要使用的方法。最后选择对应的参数设置即可，具体参数请参见下文 API 方法中的接口文档。

API 方法

腾讯云短信 API 支持以下方法：

方法	方法标识	方法描述	接口文档
发送邮件	SendEmail	发送 HTML 或者 TEXT 邮件，适用于触发类邮件（验证码、交易类）。默认仅支持使用模板发送邮件。	发送邮件

企业微信

最近更新时间：2023-07-12 17:34:15

企业微信服务端 API 开放了丰富的能力接口，开发者可以借助接口能力，实现企业服务及企业微信的集成。

⚠ 注意

若要使用企业微信 API 需要先设置 [企业微信集成](#)。

API 方法

企业微信 API 支持以下方法：

方法	方法标识	方法描述	接口文档
读取成员	user_get	应用只能获取可见范围内的成员信息，且每种应用获取的字段有所不同，在返回结果说明中会逐个说明	查看
创建成员	user_create	仅通讯录同步助手或第三方通讯录应用可调用。注意，每个部门下的部门、成员总数不能超过3万个	查看
更新成员	user_update	仅通讯录同步助手或第三方通讯录应用可调用。注意，每个部门下的部门、成员总数不能超过3万个	查看
删除成员	user_delete	仅通讯录同步助手或第三方通讯录应用可调用。若是绑定了腾讯企业邮箱，则会同时删除邮箱帐号	查看
批量删除成员	user_batchdelete	仅通讯录同步助手或第三方通讯录应用可调用	查看
获取部门成员	user_simplelist	应用须拥有指定部门的查看权限	查看
获取部门成员详情	user_list	应用须拥有指定部门的查看权限	查看
手机号获取 userid	user_getuserid	通过手机号获取其所对应的 userid	查看
邮箱获取 userid	get_userid_by_email	通过邮箱获取其所对应的 userid	查看
创建部门	department_create	部门的最大层级为15层；部门总数不能超过3万个；每个部门下的节点不能超过3万个	查看
更新部门	department_update	如果非必须的字段未指定，则不更新该字段。应用须拥有指定部门的管理权限。如若要移动部门，需要有新父部门的管理权限	查看
删除部门	department_delete	应用须拥有指定部门的管理权限。第三方仅通讯录应用可以调用	查看
获取部门列表	department_list	由于该接口性能较低，建议换用获取子部门 ID 列表与获取单个部门详情	查看
获取子部门 ID 列表	department_simplelist	获取子部门 ID 列表	查看
获取单个部门详情	department_get	获取单个部门详情	查看
创建标签	tag_create	创建的标签属于该应用，只有该应用的 secret 才可以增删成员	查看
更新标签名字	tag_update	调用的应用必须是指定标签的创建者	查看
删除标签	tag_delete	调用的应用必须是指定标签的创建者	查看
获取标签成员	tag_get	无限制，但返回列表仅包含应用可见范围的成员	查看

方法	方法标识	方法描述	接口文档
增加标签成员	tag_addtagusers	调用的应用必须是指定标签的创建者；成员属于应用的可见范围。注意，每个标签下部门数和人员数总和不能超过3万个	查看
删除标签成员	tag_deltagusers	调用的应用必须是指定标签的创建者；成员属于应用的可见范围	查看
获取标签列表	tag_list	自建应用或通讯同步助手可以获取所有标签列表；第三方应用仅可获取自己创建的标签	查看
发送应用消息	message_send	支持推送文本、图片、文本卡片消息	查看
撤回应用消息	message_recall	本接口可以撤回24小时内通过发送应用消息接口推送的消息，仅可撤回企业微信端的数据，微信插件端的数据不支持撤回	查看
创建群聊会话	appchat_create	只允许企业自建应用调用，且应用的可见范围必须是根部门	查看
修改群聊会话	appchat_update	只允许企业自建应用调用，且应用的可见范围必须是根部门	查看
获取群聊会话	appchat_get	只允许企业自建应用调用，且应用的可见范围必须是根部门	查看
应用推送消息到群聊会话	appchat_send	只允许企业自建应用调用，且应用的可见范围必须是根部门	查看
创建日历	calendar_add	该接口用于通过应用在企业内创建一个日历	查看
更新日历	calendar_update	该接口用于通过应用在企业内更新一个日历	查看
获取日历详情	calendar_get	获取日历详情	查看
删除日历	calendar_del	删除日历	查看
创建日程	schedule_add	该接口用于在日历中创建一个日程	查看
更新日程	schedule_update	该接口用于在日历中更新指定的日程	查看
新增日程参与者	schedule_add_attendees	该接口用于在日历中更新指定的日程参与者列表	查看
删除日程参与者	schedule_del_attendees	该接口用于在日历中更新指定的日程参与者列表	查看
获取日程详情	schedule_get	该接口用于获取指定的日程详情	查看
取消日程	schedule_del	该接口用于取消指定的日程	查看
获取日历下的日程列表	schedule_get_by_calendar	该接口用于获取指定的日历下的日程列表。仅可获取应用自己创建的日历下的日程	查看
发送学校通知	externalcontact_message_send	学校可以通过此接口来给家长发送不同类型的学校通知，来满足多种场景下的学校通知需求。目前支持的消息类型为文本、图片、语音、视频、文件、图文	查看
获取学校通知二维码	get_subscribe_qr_code	学校可通过此接口获取学校通知二维码，家长可通过扫描此二维码关注学校通知并接收学校推送的消息	查看
设置关注学校通知的模式	set_subscribe_mode	可通过此接口修改家长关注学校通知的模式：“可扫码填写资料加入”或“禁止扫码填写资料加入”	查看
外部联系人openid转换	convert_to_openid	企业和服务商可通过此接口，将微信外部联系人的userid转为微信openid，用于调用支付相关接口	查看
获取可使用的家长范围	get_allow_scope	获取可在微信学校通知 > 学校应用使用该应用的家长范围，以学生或部门列表的形式返回	查看
读取学生或家长	school_user_get	读取学生或家长	查看

方法	方法标识	方法描述	接口文档
获取部门学生详情	school_user_list	获取部门学生详情	查看
获取部门家长详情	school_user_list_parent	获取部门家长详情	查看
获取部门列表(家校)	get_userid_by_email	获取部门列表(家校)	查看
获取企业的jsapi_ticket	get_corp_jsapi_ticket	JS-SDK 使用签名方法，该方法会计算 JS-SDK 所需签名并且通过 corpId 和 agentId 作为 ticket 缓存依据，到期自动更新	查看
获取应用的jsapi_ticket	get_agent_jsapi_ticket	JS-SDK 使用签名方法，该方法会计算 JS-SDK 所需签名并且通过 corpId 和 agentId 作为 ticket 缓存依据，到期自动更新	查看
获取客户群列表	externalcontact_groupchat_list	该接口用于获取配置过客户群管理的客户群列表	查看
获取客户群详情	externalcontact_groupchat_get	通过客户群 ID，获取详情。包括群名、群成员列表、群成员入群时间、入群方式。（客户群是由具有客户群使用权限的成员创建的外部群）	查看
客户群opengid转换	opengid_to_chatid	用户在微信里的客户群里打开小程序时，某些场景下可以获取到群的 opengid，如果该群是企业微信的客户群，则企业或第三方可以调用此接口将一个 opengid 转换为客户群 chat_id	查看
配置客户联系我方式	add_contact_way	企业可通过此接口为具有客户联系功能的成员生成专属的联系我二维码或者联系我按钮	查看
获取企业已配置的联系我方式	get_contact_way	获取企业配置的联系我二维码和联系我小程序按钮	查看
更新企业已配置的联系我方式	update_contact_way	更新企业配置的联系我二维码和联系我小程序按钮中的信息，如使用人员和备注等	查看
删除企业已配置的联系我方式	del_contact_way	删除一个已配置的联系我二维码或者联系我小程序按钮	查看
结束临时会话	close_temp_chat	将指定的企业成员和客户之前的临时会话断开，断开前会自动下发已配置的结束语	查看
配置客户群进群方式	add_join_way	企业可调用此接口来生成并配置加入群聊的二维码或者小程序按钮，客户通过扫描二维码或点击小程序上的按钮，即可加入特定的客户群	查看
获取客户群进群方式配置	get_join_way	获取企业配置的群二维码或小程序按钮	查看
更新客户群进群方式配置	update_join_way	更新进群方式配置信息	查看
删除客户群进群方式配置	del_join_way	删除一个进群方式配置	查看

自定义 API

自定义 API 概述

最近更新时间：2023-02-24 15:47:21

腾讯云微搭低代码自定义 API 支持调用第三方服务接口或使用代码来实现业务逻辑。开发者可以在应用、工作流或其他自定义 API 中使用 API。

自定义 API 方法实现方式

目前自定义 API 支持三种方式来实现自定义业务逻辑：

- **HTTP 请求**：使用 HTTP 方式调用外部服务，通过简单的配置 HTTP 请求地址、方法、参数等即可完成方法的配置。
- **自定义代码**：集成了常用 NPM 包、数据模型、API 等 [方法](#)，只支持 JS 开发语言，可以用来实现自定义业务逻辑。
- **云开发云函数**：用户可以创建和使用云开发的云函数来实现完整的服务端功能，并通过自定义 API 的方法绑定并调用。

具体如何实现自定义 API 可以参见 [新建自定义 API](#)。

自定义 API 的使用

开发者可以在应用编辑器或其他自定义 API 中使用：

使用场景	操作指引	
应用编辑器	添加行为事件	在组件行为可以调用自定义 API 方法。
	自定义方法	可以在应用的低代码编辑器及微搭组件的组件代码中使用。
自定义 API 方法		在其他自定义 API 中通过 自定义代码 方式，可以调用其他自定义 API。

新建自定义 API

最近更新时间：2023-08-14 16:37:17

腾讯云微搭低代码自定义 API 支持调用第三方服务接口或使用代码来实现自定义业务逻辑。本文将介绍如何创建自定义 API。

操作步骤

步骤1：填写基础信息

进入腾讯云微搭控制台 > APIs 页面，单击新建 APIs。

⚠ 注意：

标识为自定义 API 的唯一标识，在微搭应用编辑器、自定义代码中均需要借助这个标识来使用。

步骤2：选择要创建的 APIs 类型

当前支持 Postman、OpenAPI 3.0、HTTP 请求、自定义代码、云开发云函数等多种类型的 APIs。

步骤3：实现 API 方法

开发者可以根据业务需求使用 HTTP 请求、自定义代码或云开发云函数方式实现自定义 API 方法。

HTTP 请求

可简单对接第三方 HTTP 接口，通过简单的配置 HTTP 请求地址、方法、参数等即可完成方法的配置。详情请参见 [HTTP 请求](#)。

自定义代码

集成了常用 NPM 包、数据模型、API 等 API，只支持 JS 开发语言，可以用来实现自定义业务逻辑，详情请参见 [自定义代码](#)。

云开发云函数

云函数是一种特殊的 Javascript 函数，底层基于云开发的云函数能力封装，最终会运行在服务器端的 Nodejs 10.15 环境中，可以通过函数的 context 参数来访问云开发 node sdk 的所有能力。详情请参见 [云函数](#)。

Postman

通过 Postman 导出的 Collection v2.1 文件，生成包含对应 HTTP 方法的自定义 API。

步骤4：启用方法

最后请勾选方法以启用已经实现的方法。

使用和管理自定义 API

最近更新时间：2023-04-13 10:36:44

本文将介绍如何使用和管理自定义 API。

使用自定义 API

目前在应用编辑器和自定义代码中，都可以使用自定义 API，支持可视化和自定义代码两种方式。

通过组件行为调用

进入应用编辑器，在组件行为事件中可以调用 API 方法，详情可参见 [行为事件-执行动作](#)。

通过低代码编辑器调用

进入应用编辑器，可以在应用的低代码编辑器及微搭组件的组件代码中使用，详情可参见 [代码编辑器](#)。

示例代码：

```
export default async function({event, data}) {
  const result = await app.cloud.callConnector({
    name: '自定义 API 标识',
    methodName: '方法标识',
    params: {}, // 方法入参
  });
}
```

在自定义代码中调用

在自定义 APIs 中通过[自定义代码](#)方式，可以调用其他自定义 API。详情请参见 [自定义代码](#)。

示例代码：

```
module.exports = async function (params, context) {
  const result = await context.callConnector({
    name: '自定义 API 标识',
    methodName: '方法标识',
    params: {}, // 方法入参
  });

  return {
    _id: '123456',
  };
};
```

编辑 API

有两种方式可以对 API 进行编辑：

- 在 [控制台 APIs 列表](#) 页面，找到编辑的 API，在操作列单击编辑。
- 在 [控制台 APIs 详情](#) 页面，单击右上角编辑。

发布数据模型

数据模型需要发布才会在正式环境生效，有两种方式可以发布数据模型：

- 在 [数据模型列表](#) 页面，找到编辑的数据模型，在操作列单击更多 > 发布。
- 在 [自定义 API 详情](#) 页面，在基础信息卡片中状态 > 立即发布。

删除 API

在自定义 API 列表页面，找到编辑的自定义 API，在操作列单击更多 > 删除。

⚠ 注意：

自定义 API 删除前，必须解绑在所有应用的关联使用。

自定义代码

最近更新时间：2023-02-27 09:40:52

在自定义 API 中使用自定义代码能力，可以方便开发者实现不同的业务场景。开发者可以通过自定义代码中内置的 API 或 npm 包调用外部 HTTP 服务、微搭数据模型、API、自定义 API 或云开发云函数。

自定义代码

自定义代码是一种运行在服务端的 NodeJS 代码，底层是云开发云函数。以下是一个自定义代码的主要构成：

```
/**  
 * 可以在这里引用内置 NPM 包，例如：  
 **/  
const fetch = require('node-fetch');  
  
/**  
 * 在这个函数体内可以实现业务逻辑  
 **/  
module.exports = async function (params, context) {  
    // params 即为入参定义的结构  
    // context 为平台内置对象，包含了内置的 API  
  
    // 在这里返回这个方法的结果，需要与出参定义的结构映射  
    return {  
        _id: 123456  
    };  
};
```

使用场景示例

发起 HTTP 请求

开发者可以使用 node-fetch 库来发起 HTTP 请求。详情请参见 [node-fetch 使用文档](#)。

```
const fetch = require('node-fetch');  
  
module.exports = async function (params, context) {  
  
    const response = await fetch('https://reqres.in/api/users');  
    const result = await response.json();  
  
    // 在这里返回这个方法的结果，需要与出参定义的结构映射  
    return {  
        _id: '123456'  
    };  
};
```

数据模型

开发者可以使用 context.callModel 方法来操作数据模型，完整的方法请参见 [数据模型方法](#)。

```
module.exports = async function (params, context) {  
    const result = await context.callModel({  
        name: 'example_xxxxxxx', // 数据模型标识，可以前往「数据源 - 数据模型」列表页查看  
    });  
};
```

```
methodName: 'wedaCreate', // 数据模型方法标识，支持的方法可以前往「数据源 - 数据模型」的任一数据模型详情页查看当前模型支持的方法
params: {}, // 数据模型方法的入参
});

// 在这里返回这个方法的结果，需要与出参定义的结构映射
return {
_id: '123456',
};
};
```

API

开发者可以使用 `context.callConnector` 来调用 API 方法：

```
module.exports = async function (params, context) {
const result = await context.callConnector({
name: 'API 标识',
methodName: '方法标识',
params: {}, // 方法入参
});

// 在这里返回这个方法的结果，需要与出参定义的结构映射
return {
_id: '123456'
};
};
```

云开发云函数

开发者可以使用 `context.app.callFunction` 来调用同环境下的云开发云函数：

```
module.exports = async function (params, context) {
const result = await context.app.callFunction({
name: '云开发云函数名称',
data: {}, // 方法入参
});

// 在这里返回这个方法的结果，需要与出参定义的结构映射
return {
_id: 123456
};
};
```

⚠ 注意：

此方法只能调用与微搭相同云开发环境中的云函数。

若开发者有自己的数据库（腾讯云或自有数据库），可以使用云开发云函数来实现数据库连接和读写，在与微搭相同云开发环境中新建以下云函数，示例：

```
const mysql = require("mysql2/promise");
exports.main = async (event, context) => {
try {
const connection = await mysql.createConnection({
host: process.env.HOST,
```

```
user: process.env.USERNAME,
password: process.env.PASSWORD,
port: process.env.PORT,
database: process.env.DB,
});

console.log('已连接')
const [rows, fields] = await connection.execute('SELECT * FROM `weda_model_example`;');
// 这里可以对返回数据做加工
return rows;
} catch(err) {
console.log('错误连接', err);
return err;
}
};
```

然后在自定义代码调用上述云函数，参考上述 `context.app.callFunction` 的使用示例。

云开发数据库

开发者可以使用 `context.database` 来操作 [云开发数据库](#)，查看 [云开发数据库 API](#)。

```
module.exports = async function (params, context) {
const result = await context.database.collection('数据库集合名称').get();

// 在这里返回这个方法的结果，需要与出参定义的结构映射
return {
_id: 123456
};
};
```

 **注意：**

此方法只能操作微搭云开发环境下的数据库，无法跨环境操作数据库。

图表统计方法参数规范

最近更新时间：2023-02-21 14:11:13

当图表组件使用自定义 API 时，需在自定义 API 内，按照以下规范定义出入参，用来实现数据的统计。图表组件使用自定义 API 实现数据统计，示例请参见 [使用自定义 API 实现统计图表](#)。

柱状图、折线图、饼图

出参数据格式

字段名	类型	是否必须	描述
result[]	array	是	<p>返回图表渲染的数据数组</p> <pre>[{ XLabel: {Value: "张三"}, YLabels: [{Name: "娃的数量", Value: 3}, {Name: "销售金额", Value: 123123}] }, { XLabel: {Value: "李四"}, YLabels: [{Name: "娃的数量", Value: 3}, {Name: "销售金额", Value: 123123}] }, { XLabel: {Value: "麻子"}, YLabels: [{Name: "娃的数量", Value: 3}, {Name: "销售金额", Value: 123123}] }]</pre>

统计卡片

出参数据格式

字段名	类型	是否必须	描述
result[]	array	是	返回字段统计列表 [{Name:a,Value:1}]
result[].Value	number	是	字段统计值
result[].Name	string	否	字段标识

通用选项集

概述

最近更新时间：2023-02-17 16:02:58

概述

建立一个通用的选项维护入口，可在所有枚举字段中直接使用，同样的选项值在同一套环境中无需反复添加，方便用户操作，提高工作效率。

通用选项集管理

预制通用选项

系统中预制了部门和用户通用选项值，供环境中所有数据模型重复使用，在枚举字段中可以快速应用。

新建通用选项

进入 [数据源](#) > [通用选项集](#) 页面，单击[新建选项集](#)即可新建。

② 说明：

一个通用选项集最多支持1000个选项。

- 选项集名称：必填，默认不超过60个字符，允许汉字、字母、数字或特殊字符。
- 选项集标识：必填，默认不超过20个字符，只允许字母、数字或“_”。
- 描述：非必填。
- 选项集：必填。
 - 选项标识：
 - 每个选项标识默认按照数字1、2、3、4、5依次递增，新建选项时会有默认值，按照当前选项的个数，例如第4个选项，默认选项标识为4。
 - 每个选项标识至多不超过20个字符，允许字母、数字或“_”，可以字母或数字开头。
 - 选项值：每个选项值至多不超过30个字符，允许汉字、字母、数字或特殊字符，可以汉字、字母、数字开头。

② 说明：

选项标识会自动生成，可以自定义。每个选项集中选项标识不能重复。选项值只能自定义。

编辑通用选项

默认选项集标识不能被修改，选项值可以根据需要修改。

查看通用选项

单击选项名称，可以快速查看通用选项的值。

删除通用选项

如果当前选项集未被字段引用，则单击[更多](#) > [删除](#)的时候会有弹窗提醒，单击[确定](#)以后删除。

如果当前选项集已经被引用，则删除的时候，会提示“当前选项集已经被字段引用，请先删除引用的字段！”。

复制通用选项

当有选项集可以被复用时，可以先复制一份再进行修改，提高配置选项效率。

使用通用选项

进入 [数据源](#) > [数据模型](#)页面，可在数据类型为枚举的字段中，单击[关联选项集](#)选择环境中已经存在的通用选项集，并设置单选或者多选，以及默认值。如果没有合适的选项值集，也可以单击[立即创建](#)新建一个选项集。

查看引用字段

可单击[查看引用](#)查看当前通用选项集被字段使用的情况。

 注意：

有字段引用时选项集不能被删除。

查询通用选项集选项值

最近更新时间：2023-08-02 15:43:44

本文将介绍如何通过通用选项集选项标识，查询选项集的值。

操作步骤

步骤1：创建通用选项集服务

- 选择一个应用，进入数据源 > APIs > 新建 APIs，微搭内置开放服务选择通用选项集。

The screenshot shows the 'APIs' section of the Tencent Cloud Low-Code Platform. At the top, there are four tool icons: 'HTTP Request', 'Postman 2.1', 'OpenAPI 3', and 'Cloud Function'. Below them are two more icons: 'Custom Code' and 'Cloud Function'. On the left, there's a sidebar with application filters: '当前应用' (Current Application), 'Q', '已发布' (Published), '待审核' (Pending Review), '已发布' (Published), '已发布' (Published), '已发布' (Published), '直连测试-勿删' (Direct Connection Test - Do Not Delete), '已发布' (Published), '已发布' (Published), and '已发布' (Published). The main area is titled '开放服务' (Open Services) and lists several services: '腾讯会议' (Tencent Meeting), '腾讯地图' (Tencent Map), '腾讯云邮件' (Tencent Cloud Mail), '企业微信' (Enterprise WeChat), '腾讯云短信' (Tencent Cloud SMS), '腾讯云文字识别' (Tencent Cloud Text Recognition), and '小程序开放能力' (Microservice Open Ability). At the bottom, under '微搭内置开放服务' (Built-in Open Services), there are two options: '系统工作流' (System Workflow) and '通用选项集' (General Options Set), with the latter being highlighted by a red box.

- 创建通用选项集服务，录入名称、SecretId 和 SecretKey 信息。

新建APIs

① 选择APIs类型 > ② 配置APIs信息

自定义APIs 更换

通用选项集

自定义APIs

名称：通用选项集 ✓
标识：optionSet ✓
请输入APIs英文标识，作为APIs唯一标识，不可重复
描述：请输入APIs描述，最多支持 255 个字符

参数配置

SecretId：AKIDCLRXmx ✓
SecretKey：..... ✓
[如何获取访问密钥](#)

创建 取消

SecretId 和 SecretKey 获取如下图指引：

访问管理 API密钥管理

安全提示

- 所有API密钥代表您的账号身份和所拥有的权限。使用腾讯云API可以操作您名下的所有腾讯云资源。
- 为了您的账户和服务器安全，请妥善存储和定期更换密钥。请勿将访问方式（如curl命令）上传或者分享您的密钥信息。建议您参考[安全设置策略](#)。
- 使用强密码TLS（安全传输协议）调用云API有安全风险。建议使用TLS1.2及以上版本。
- 可使用密钥管理系统（KMS）生成密钥进一步保护API密钥，提升安全性。详细请参考[KMS使用说明](#)。

使用提示

- 云API密钥是构建腾讯云API请求的重要凭证。用于连接腾讯云API时生成签名，查看生成签名方法。
- 最近访问时间指最近一次使用该密钥访问云API V3.0 端口的时间。此时间仅供参考密钥是否过期，以此决定是否启用或删除密钥。

APPID	密钥	创建时间	最近访问时间	状态	操作
12	SecretId: AKIDCLRXmx, SecretKey: 显示	2023-07-27 14:56:44	2023-07-27 15:10	已启用	禁用 更多操作

3. 创建成功后，选项集服务如下图：

通用选项集 (optionSet)

APIs方法 权限设置 鉴权设置 关联应用 基本信息

APIs方法 管理公共变量 (1)

APIs方法列表

名称：查询多个选项集内容
标识：describeOptionSet
 [查询多个选项集内容](#)

描述：
URL：{{vars.j}}
Method：POST

入参

字段名称	字段标识	数据类型	是否必填
通用选项集标识数组	optionSetNames	数组	否

HTTP配置

Query Headers Body

KEY VALUE 操作
智元数据

出参

字段名称	字段标识	数据类型	字段映射
response	response	对象	

步骤2：使用通用选项集服务查询选项值

创建需要使用的通用选项集。

[查看通用选项集](#) ①

选项集名称 * 性别

选项集标识 * sex

描述 -

选项集 * 1 男 2 女

编辑器代码区创建查询 Query，使用通用选项集服务进行查询。

代码区

当前页面 新建数据查询 (当前页面) x

新建数据查询 (当前页面)

新建 SQL 表查询 新建 MySQL 定义 新建外部 APIs 查询 (当前)
新建自定义变量 新建 JavaScript 方法 新建事件流

选项集标识可以选择页面或全局变量承载，为方便演示此处固定为性别，选项集的标识 sex。

The screenshot shows the 'Edit Data Request' configuration screen. Under 'APIs', 'query1' is selected. The 'Trigger Mode' is set to 'On parameter change'. The 'Method' dropdown shows '多个方法内嵌 (describeOptionSet)'. The 'Input' section contains a note: '通用选项集标识数组... 1. 使用选项集标识数组'.

场景1：文本、单行输入等展示类组件

通用选项集函数具体使用请参见 [通用选项集获取枚举值](#)。

在输入值中绑定表达式，即可获取到对应 key 的 value 值。

The screenshot shows the configuration of a 'text1' component. In the 'Properties' tab, the 'Component ID' is 'text1'. The 'Text Content' field contains the expression: `Sw.utils.getEnumValue({ enumOptions: Sw.query1.data, optionSetName: 'sex', key: '1' })`. The 'Advanced Properties' tab shows the 'Event' section with a click event.

场景2：下拉选择、下拉多选等表单类组件

在选项中绑定变量表达式，获取到对应的选项集列表。

iPhone 8 4G 03:56 100% 空白页

男

下拉选择 请选择 >

下拉选择 清空 搜索

男 女

必填

选项

变量 \$w.utils.getEnumValue... (highlighted with a red box)

选项名称 请选择字段

选项值 请选择字段

```

:: select1/选项
1 $w.utils.getEnumValue({
2   enumOptions:$w.query1.data,
3   optionSetName:'sex'
4 })

```

数据查询
query1 (query)
系统变量
currentUser (登录用户信息) object
app (应用信息) object
page (当前页面信息) object
device (设备信息) object
env (环境信息) object
wedaContext (上下文信息) object
URL参数 +
组件 API
text1
select1
函数公式
计算
逻辑
文本
日期时间

场景3：代码编辑器中使用

可以在自定义代码中使用，以获取选项集内容，执行所需逻辑。

```

8 export default {
9   onPageLoad(query) {
10     //console.log('-----> LifeCycle onPageLoad', query)
11   },
12   onPageShow() {
13     //console.log('-----> LifeCycle onPageShow')
14   },
15   onPageReady() {
16     //console.log('-----> LifeCycle onPageReady')
17     const sexLabel = $w.utils.getEnumValue({
18       enumOptions: $w.query1.data, optionSetName: 'sex', key: '1'
19     });
20     console.log(sexLabel);
21   },
22   onPageHide() {
23     //console.log('-----> LifeCycle onPageHide')
24   },
25   onPageUnload() {
26     //console.log('-----> LifeCycle onPageUnload')
27   },
28 }

```

连接器

MySQL 连接器

最近更新时间：2023-07-13 15:39:57

MySQL 连接器介绍

MySQL 连接器提供了连接您自有的 MySQL 数据库，对接微搭开发的应用，以获取或操作数据库中的数据。



在使用 MySQL 连接器时，有两项配置需要进行操作：连接器配置及表达式编写。

- **连接配置：**通过配置连接器，确定连接 MySQL 数据库的地址、方法、连接的数据库、认证等相关信息。
- **Query 表达式编写：**通过编写表达式，可以使用 SQL 语句进行数据库操作。同时可以通过表达式与页面组件、代码实现联动。

连接配置说明

通过微搭编辑器的代码区，可以通过**新建 MySQL 查询**能力来新建表达式。在首次创建 MySQL 表达式时，如果没有 MySQL 连接配置，将会弹窗要求新建 MySQL 连接配置；在完成连接配置后，后续如果希望创建新的连接配置，或者修改当前连接配置，也可以通过连接配置选择的下拉菜单选择执行。

连接MySQL数据库

MySQL配置名称 *

此名称将在编辑器创建查询时使用

标识

唯一标识

_0umvk8r

接入方式

通过公网IP/域名对接 对接腾讯云数据库

主机 *

请填写数据库网络地址

端口 *

3306

数据库名 *

请填写需要连接的数据库库名

用户名

请填写连接数据库的用户名

密码

请填写连接数据库的密码

SQL预编译

连接参数

key value

+ 添加参数

测试连接 保存

1. 连接配置用于告知微搭平台连接 MySQL 数据库的相关配置信息，在创建或修改连接配置时，有如下信息需要填写：
 - 连接配置名称、标识：用于标明数据库连接配置的名称、唯一标识。

- 连接类型选择：MySQL 连接器支持通过公网对接第三方 MySQL，通过内网对接腾讯云 TDSQL-C MySQL 版及 TencentDB MySQL。
 - 主机及端口：当选择通过 IP/域名连接数据库时，需要填写 MySQL 数据的主机地址及连接端口，主机地址可以填写外网 IP 或域名。
 - 数据库 ID：当选择连接腾讯云数据库时，可以通过填写对应的数据库实例 ID 确定所需连接的数据库。
 - 数据库名：填写需要连接及使用的数据库。
 - 用户名及密码：填写连接数据库所使用的用户名及密码认证信息。
 - SQL 预编译：默认开启状态，开启后，表达式中的 SQL 语句将预编译后执行。预编译可以极大的提高 SQL 执行时的安全性，避免 SQL 注入等风险。预编译详细说明可见 [MySQL Prepared Statements](#)。开启预编译后，SQL 语句中不可以在数据库名、表名等位置使用参数传递。如需动态表名、数据库名，可以通过关闭预编译的方式提供，同时也可以通过编写多个表达式并选择执行的方式达到同样目的。
 - 连接参数：用于设置与数据库建立连接时的可选参数配置。连接参数详细请参见 [MySQL connection options](#)。
2. 在完成连接配置填写后，可以通过连接测试能力，验证连接配置的正确性。如测试失败，可以根据测试返回的 MySQL 报错，调整修正连接配置。
3. 完成测试后，可以单击 **保存**，保存连接配置。如果是修改连接配置，在保存后，新的配置信息将生效，基于此连接配置的表达式，均将使用新的配置信息进行后续 SQL 的执行。
4. 如果后续需要修改连接配置，也可以通过下拉菜单选中配置，并通过单击 **修改**，在打开的修改窗口中进行配置的修改调整。

② 说明：

通过域名或 IP 的方式，仅支持通过公网访问数据库；通过数据库 ID 对接腾讯云数据库，仅可以支持上海地区的 TDSQL-C MySQL 版及 TencentDB MySQL 数据库

Query 表达式编写说明

通过微搭编辑器的代码区，可以通过新建 MySQL 查询能力来新建表达式。在完成 MySQL 连接配置后，就可以通过编写 SQL 语句实现表达式获取数据或操作数据的需求。

The screenshot shows the MySQL Configuration interface. It includes fields for MySQL Configuration Name (test_jaurpx0), Trigger Mode (set to 'On change automatically execute'), and SQL Statement (SELECT * FROM userinfo;). On the right, there's a preview pane for SELECT queries, showing examples like 'SELECT * FROM users ORDER BY id LIMIT 10;' and 'INSERT INTO users (name,gender,phone) VALUES ({\$w.form.nameInput}, {\$w.form.genderSelect}, {\$w.form.phoneInput})'. There's also a note about using LIMIT to avoid large result sets.

SQL 语句编写

SQL 语句根据实际业务需求进行编写即可。可以使用的包括 SELECT、INSERT、UPDATE、DELETE 语句。当前在 MySQL 连接器的表达式中暂时不提供库表相关操作的支持，包括 CREATE、DROP、ALTER、TRUNCATE。

SQL 语句示例如下：

```

SELECT * FROM users;
# 从指定表中查询数据；可通过修改'users'指定到所需操作的表，可通过修改 '*' 为指定获取字段；

INSERT INTO users (name, email, phone) VALUES ('{$w.form.inputname.value}', '{$w.form.inputemail.value}', '{$w.form.inputphone.value}');
# 向指定表中插入数据；可通过修改'users'指定到所需操作的表，可通过 {{参数}} 引用组件、变量数据；

UPDATE users SET email='{$w.form.modifyemail.value}', phone='{$w.form.modifyphone.value}' WHERE id='{$w.selectuser.dat'

```

```
a.id};

# 更新指定表中的数据; 可通过修改'users'指定到所需操作的表, 可通过 {{参数}} 引用组件、变量数据;

DELETE FROM user WHERE id={{$w.selectuser.data.id}};

# 删除指定表中的数据; 可通过修改'users'指定到所需操作的表, 可通过 {{参数}} 引用组件、变量数据;
```

在 SQL 语句的输入框中支持同时写入多条语句, 使用 ; 进行语句间分隔即可。表达式的输出值, 以最后一条执行的 SQL 语句为准。

参数获取

在 SQL 语句中, 可以通过 {{ }} 的方式编写表达式, 获取应用或页面的变量、组件值, 用于 SQL 语句执行时的参数传递。

表达式的详细说明请参见 [表达式介绍](#), 如果需要获取组件、控件中的数据、属性, 同样可以参见 [获取组件属性值](#)。

例如, 在表达式中可以通过 {{'%' + \$w.input.value + '%'}} 的方式实现拼接, 达到使用 SQL 语句, 通过用户输入, 进行模糊查询的效果: SELECT * from userinfo where username = {{'%' + \$w.input.value + '%'}}。

特定系统参数获取: 在表达式中, 可以通过 {{SERVER.userId}} 获取当前应用的登录用户 ID, 此表达式在运行后端解析, 不依赖前端控件参数传递。

Table	Type	Fields
orderinfo	Bigint	id, orderid, userid, productid, totalfee
userinfo	Text	id, username, password, email, phone, created_at, updated_at

表达式返回值

SQL 语句执行完成以后, 结果将以 JSON 的形式返回。返回值可以在表达式参数中获取。

在表达式执行后, 可以通过表达式的 records 字段获取到 SQL 的执行返回内容。当查询语句时, records 字段内包含查询到的具体数据, 当插入、更新、删除语句时, records 字段内包含 affectedRows 字段, 用于标识执行操作的行数。

在前端应用、代码、控件中, 可以通过 \$w.queryname.records 的方式, 获取到 Query 的执行内容, 其中 queryname 根据具体创建 Query 时的名称来确定。

表达式的详细引用和使用方式请参见 [Query 数据查询介绍](#)。

使用示例

通过本示例, 将提供数据库数据表准备、连接配置、表达式编写和控件对接的完整示例。

数据库数据表准备

1. 准备自有的 MySQL 数据库, 准备好数据库的连接信息, 认证信息, 例如:

- Host: gz-cdb-xxxxxxx.sql.tencentcdb.com
- Port: 63999
- User: wedatest
- Password: xxxxxxx

2. 在数据库中创建 test 库, 并使用如下接口创建表 userinfo。

```
CREATE TABLE `userinfo` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `username` text NOT NULL,
  `password` text,
```

```
`email` text,  
 `phone` text,  
 `created_at` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,  
 `updated_at` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
 PRIMARY KEY (`id`)  
 ) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8
```

3. 表创建完成后，通过执行如下 SQL 语句，在表中预先准备一定的测试数据。

```
INSERT INTO `userinfo` ('username', `email`, `phone`) VALUES ('user1', 'user1@qq.com', '18612345678');  
INSERT INTO `userinfo` ('username', `email`, `phone`) VALUES ('user2', 'user2@qq.com', '18612345679');  
INSERT INTO `userinfo` ('username', `email`, `phone`) VALUES ('user3', 'user3@qq.com', '18612345680');
```

应用创建

通过微搭控制台，创建自定义应用。

配置数据库连接，创建查询表达式

1. 在应用代码区，新建数据查询，选择新建 MySQL 查询，创建表达式名字为 query1。

如果弹出连接配置信息，根据数据库的连接信息配置连接，填写连接配置名，填写数据库的 HOST 地址、PORT 端口、数据库名 test、认证 User 用户名、Password 登录密码。

连接MySQL数据库

MySQL配置名称 *

此名称将在编辑器创建查询时使用

标识 唯一标识 _0umvk8r

接入方式 通过公网IP/域名对接 对接腾讯云数据库

主机 * 请填写数据库网络地址

端口 * 3306

数据库名 * 请填写需要连接的数据库库名

用户名 请填写连接数据库的用户名

密码 请填写连接数据库的密码

SQL预编译

连接参数 key value

2. 配置完成后测试连接，确保测试通过后保存配置。

3. 编写如下 SQL 语句：

```
select * from userinfo
```

4. 保存表达式并测试运行，检查是否获取到数据库中已插入的数据，确保 SQL 运行正常。

MySQL配置名称: mysqltest_h59gqh2
触发方式: 入参变化时自动执行
SQL语句: select * from userinfo;
事件设置: 成功时
备注信息:
当前MySQL表信息
当前MySQL表信息
获取数据 (SELECT)
使用 'SELECT' 语法从数据表中获取数据。例如, 从 users 表中获取数据可以使用如下语句:
`SELECT * FROM users ORDER BY id LIMIT 10;`
建议使用 LIMIT 用于避免单次查询过大数据。
插入数据 (INSERT)
使用 'INSERT' 语法向数据表中插入一行或多行数据。假设 users 表中有 name, gender, phone 3列, 可以使用如下语句从页面变量或组件中读取数据并插入数据库:
`INSERT INTO users (name,gender,phone) VALUES ($w.form1.nameInput), ($w.form1.genderSelect), ($w.form1.phoneInput);`

对接应用控件，提供数据展示

- 在应用中添加数据表格组件，编辑表格组件的表达式，填写为 \$w.query1.data.records，其中 query1 为前面步骤创建的 MySQL 表达式名称。
- 调整表格组件的列管理，确保包含了 id、username、email、phone、created_at、updated_at 列。
- 刷新编辑区页面，可以查看到当前表格中已有了数据库中的内容。

组件: 首页
数据表格
全局按钮
按钮
按钮
操作列
按钮
数据表格
属性
组件ID: table1
表达式: \$w.query1.data.records
表达式
数据主键: id
表结构
列管理
id
username
phone
email
created_at
updated_at
新增列

增加添加用控件，提供数据插入

- 在页面上新增表单容器 form1，并在表单容器中新增3个单行输入，组件 ID 分别为 input1、input2、input3，分别用于用户名、手机、邮箱输入，根据情况配置好标题及数据校验规则。
- 在表单容器中增加按钮，标题为新增，组件 ID 为 button1。

创建新增表达式

- 在代码区再次新建 MySQL 查询，创建表达式名字为 query2，连接配置使用之前已经创建好的连接。
- 编写如下 SQL 语句：

```
INSERT INTO userinfo (`username`, `email`, `phone`) VALUES ({$w.input1.value}, {$w.input3.value}, {$w.input2.value});
```

3. 触发方式选择为手动触发执行。

The screenshot shows the low-code platform's configuration interface. On the left, there's a sidebar with '大屏树' (Dashboard Tree) and '代碼區' (Code Area). The main area has a table of user information (id, username, email, phone, updated_at, created_at) and a search bar. A modal window is open for creating a new user, showing fields for '用户名' (Username), '手机' (Mobile), and '邮箱' (Email). Below the table is a configuration panel for a trigger named 'query2'. It shows the MySQL configuration 'mydbtest; charset: utf8mb4', the trigger type '手动触发的自动执行' (Automatic execution of manual trigger), and the SQL query: `insert into userinfo (username, email, phone) VALUES ({{isn.input.value}}, {{isn.input.value}}, {{isn.input.value}});`. The configuration panel also includes sections for '事件' (Events) and '操作按钮' (Operation Buttons).

关联表达式与按钮控件

选择表单控件中的按钮，添加点击事件，选择调用数据查询，选择 Query 为新增表达式创建的 query2。

This screenshot shows the configuration of a button's click event. In the '事件' (Events) section of the configuration panel, there is a red box around the '点击时 (click)' (On Click) event. The '操作' (Action) dropdown is set to '提交 (submit)'. The '事件' (Event) dropdown is set to '调用数据查询 (Call Data Query)'. The '选择Query (Select Query)' dropdown is set to 'query2'. The configuration panel also includes sections for '组件' (Components), '属性' (Properties), and '样式' (Styles).

验证表达式执行

在页面的控件中填写具体插入用户信息，例如：“user4”，“1861234578”，“user4@qq.com”，并单击按钮，执行新增过程。执行完成后刷新页面，查看数据表格中是否已有新增的用户信息。

工作流

工作流概述

流程的组成

最近更新时间：2023-01-12 09:13:29

概述

企业业务高效率的执行，离不开规范的业务流程管理。工作流——是为达到特定的价值目标而由不同的人分别共同完成的一系列活动。活动之间不仅有严格的先后顺序限定，而且活动的内容、方式、责任等也都必须有明确的安排和界定，以使不同活动在不同岗位角色之间进行转手交接成为可能。

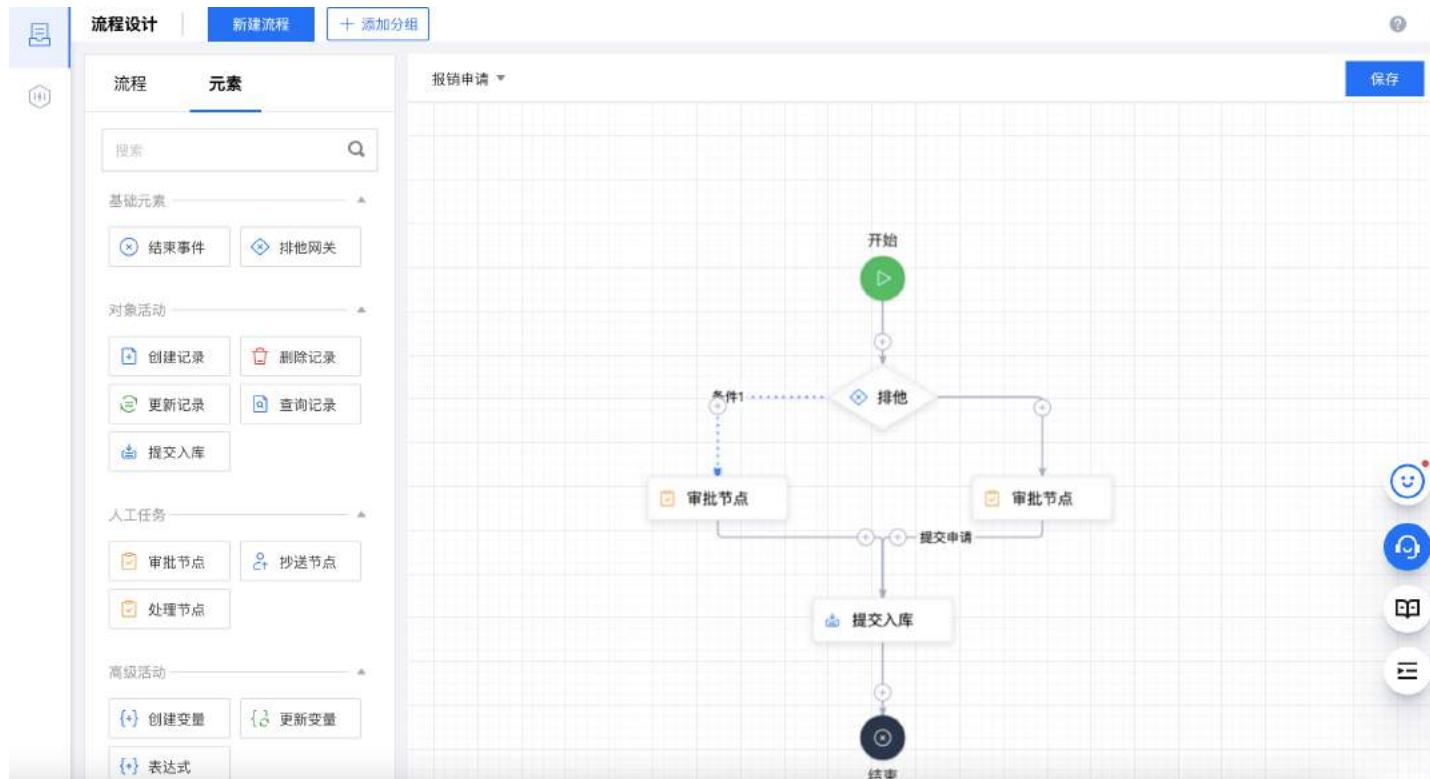
微搭“工作流”将一系列业务活动抽象为不同类型的“流程节点”，通过流程编辑器进行连接，用户可根据业务需求，通过不同类型节点的组合绘制流程，最终实现标准重复性的任务由自动化代替人工操作，提高业务执行效率。

流程的组成

一个工作流由触发器和若干个动作节点组成。

- 触发器：流程启动的条件和开关，可参见 [流程触发方式](#)，按照业务需求配置流程的触发方式和条件。
- 动作节点：将一系列业务活动抽象为不同类型的“流程节点”，通过多种类型的节点来完成各种任务，以实现流程自动化。

微搭工作流可通过可视化拖拽的方式设计应用业务逻辑。包括：以人工任务执行为核心的审批流设计。自动化执行的业务逻辑流转，例如：创建或更新对象记录、连接第三方系统、自定义表达式等。



操作步骤

最近更新时间：2023-05-06 10:27:35

步骤1：新建流程

1. 登录 微搭控制台，进入工作流模块，并单击左上角新建流程。

The screenshot shows the Tencent Cloud Weidai Low-Code Platform interface. On the left, there is a sidebar with various options: 首页 (Home), 快速开始 (Quick Start), 创建应用 (Create Application), 学习中心 (Learning Center), 应用开发 (Application Development), 应用 (Application), 模板 (Template), 数据源 (Data Source), 工作流 (Workflow) [highlighted with a red box], 组件库 (Component Library), 管理中心 (Management Center), 资源管理 (Resource Management), 自定义域名 (Custom Domain), 用户权限 (User Permissions), 企业工作台 (Enterprise Workstation), 企业微信集成 (WeChat Enterprise Integration), and 小程序开发认证 (Mini Program Development Certification). The main area is titled '工作流' (Workflow) and contains tabs for '流程设计' (Process Design) and '新建流程' (Create Process), with '新建流程' highlighted by a red box. Below these tabs is a search bar and a list of processes, with one item named '学生请假' (Student Leave) listed.

2. 弹窗中输入流程名称、流程唯一标识、描述，并单击确定。

创建新流程

流程名称 *

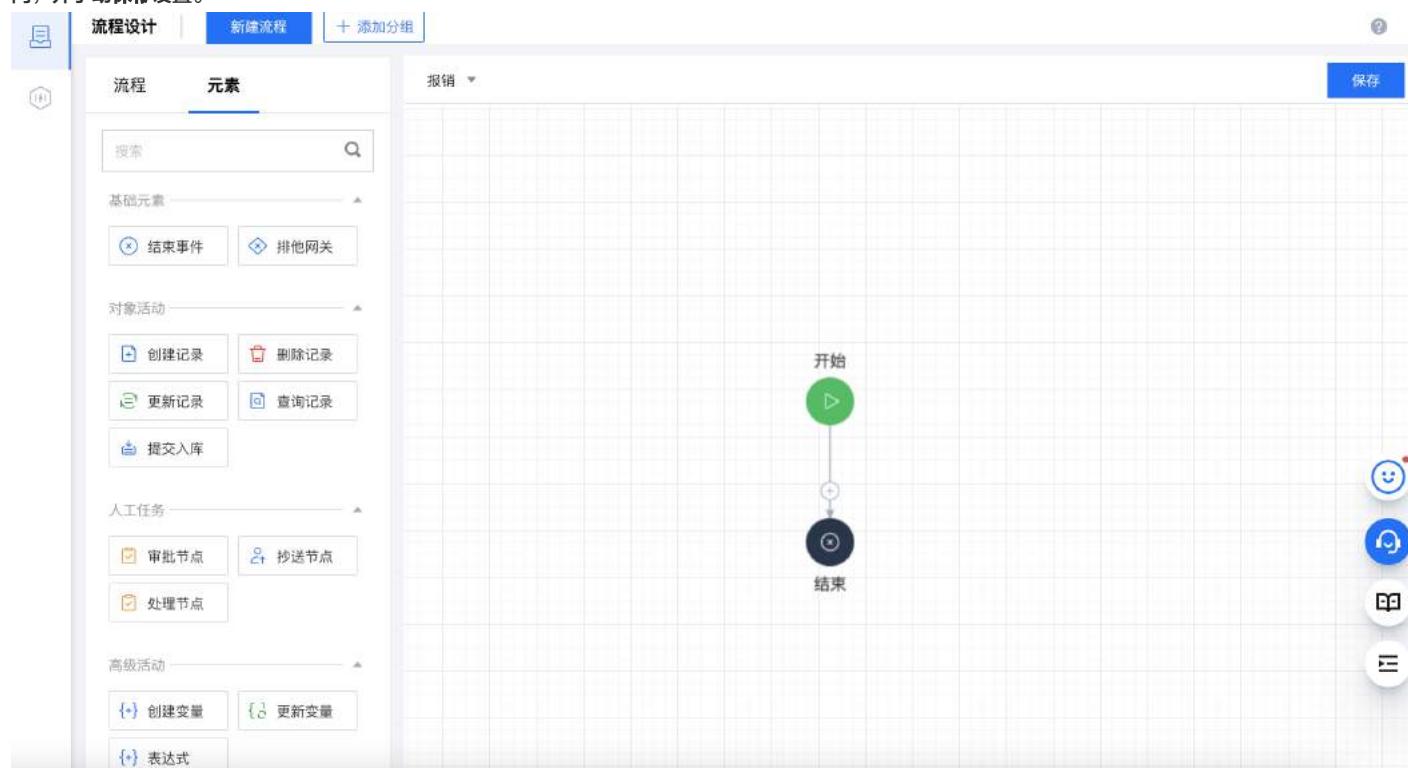
流程唯一标识 * flow_xbqin2q

流程所属分组 请选择所属分组

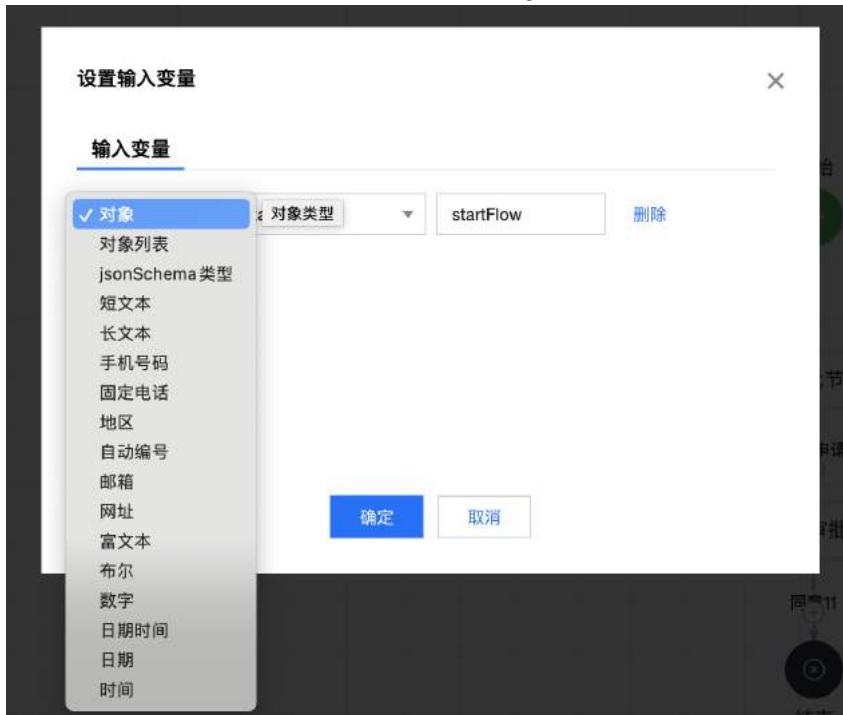
描述

确定 **取消**

3. 创建流程完成后，流程自动在画布区生成一个开始与结束节点。用户可在画布区操作，从左侧元素区列表选中所需元素，然后拖入画布区开始与结束节点之间，并手动保存设置。

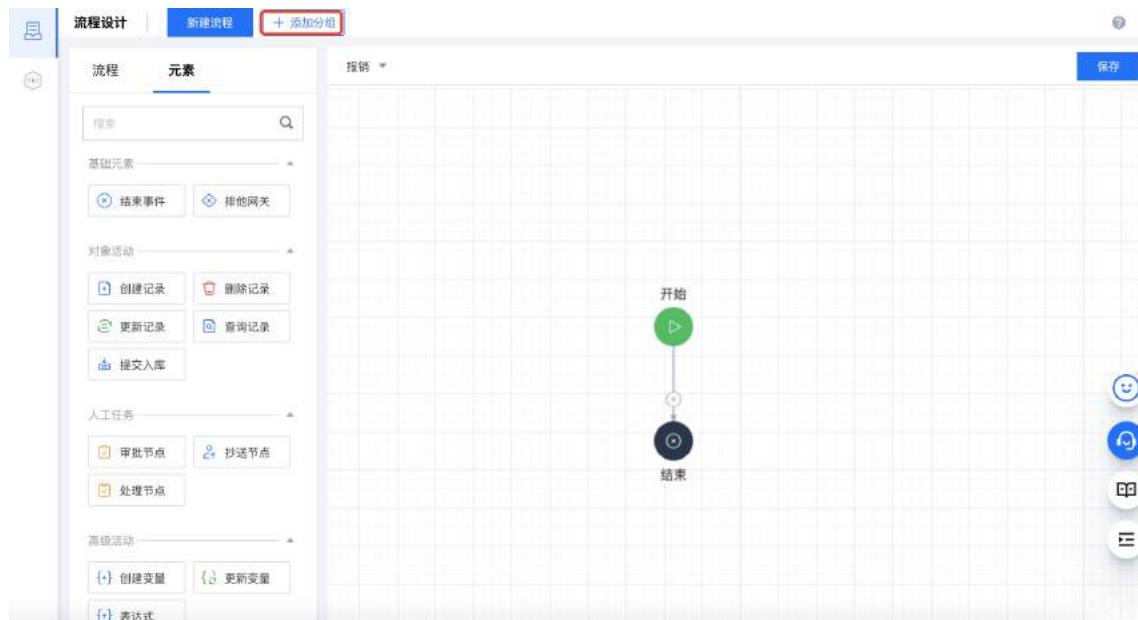


4. 在开始节点中，可对输入变量设置为对象类型、对象列表、jsonSchema 类型等多种类型。



步骤2：流程分组

单击添加分组，可以创建新的流程分组。



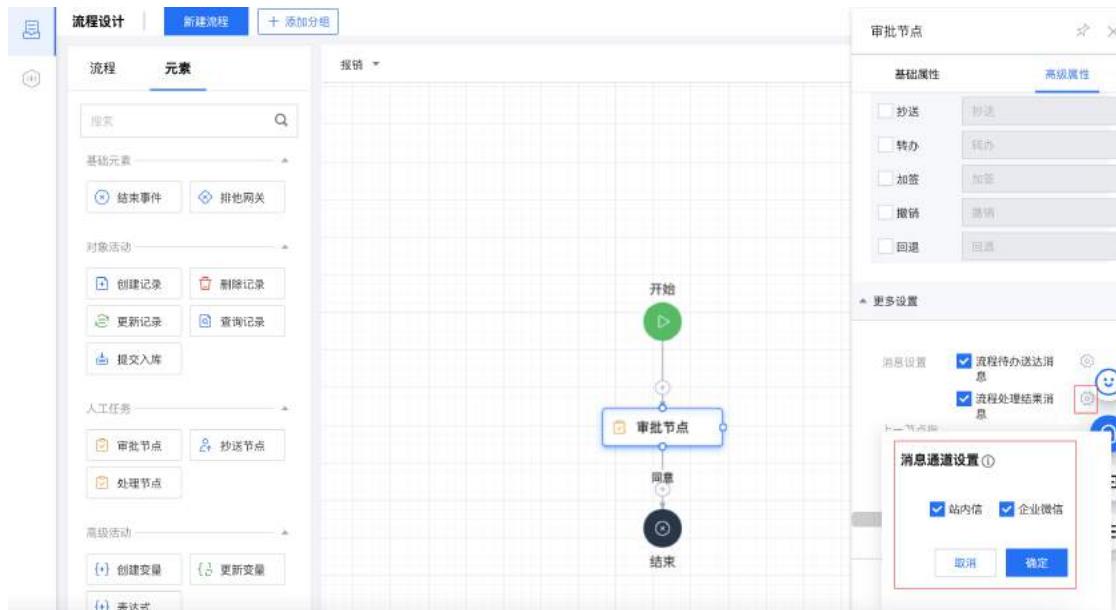
流程分组创建完毕后，单击新建流程，即可在新建的流程中选择创建的分组，更好的管理相关流程的分组情况。

创建新流程

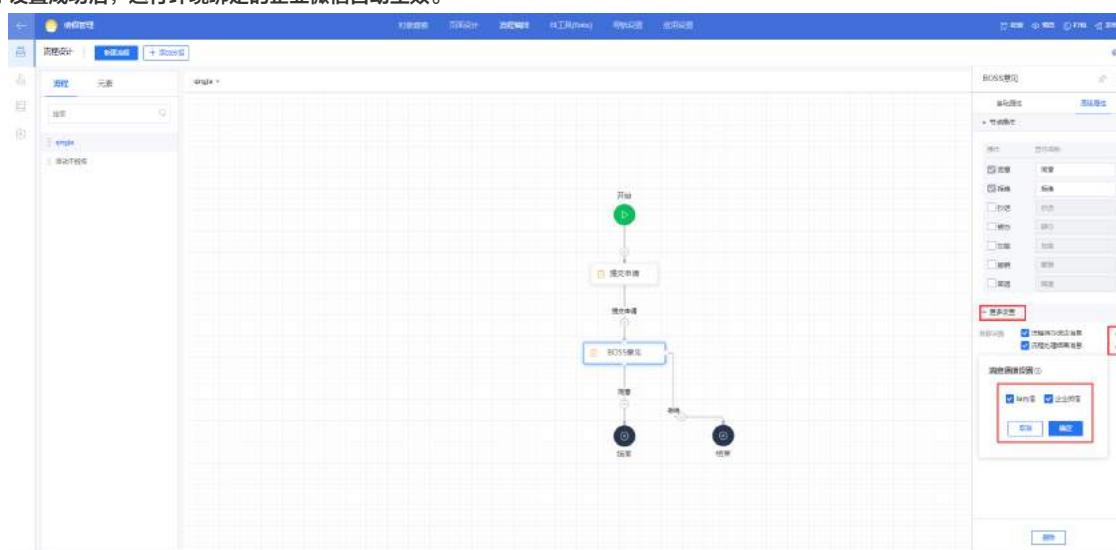
流程名称 *	liucheng03	
流程唯一标识 *	flow_016	
流程所属分组	请选择所属分组	
描述	组01	
<input type="button" value="确定"/> <input type="button" value="取消"/>		

步骤3：审批待办通知发送企微消息

1. 用户通过高级属性设置可以设置将此流程的消息发送到企业微信中进行审批，前置依赖在用户管理模块完成了企业微信授权。



2. 设置成功后，运行环境绑定的企业微信自动生效。



3. 在流程中心发起相关审批流程，流程会通过企业微信应用文本卡片消息推送该流程的审批消息到企业微信侧“微搭低代码”应用通知消息中。

流程其他操作

用户在左侧流程列表选中流程，可单击下拉按钮选择执行删除、复制、版本管理等操作。

消息订阅功能介绍

最近更新时间：2023-07-12 15:34:33

使用场景

当人工任务中消息推送无法满足需求时，希望能够个性化定制消息内容或推送至其他消息平台。

功能实现

流程消息订阅配置

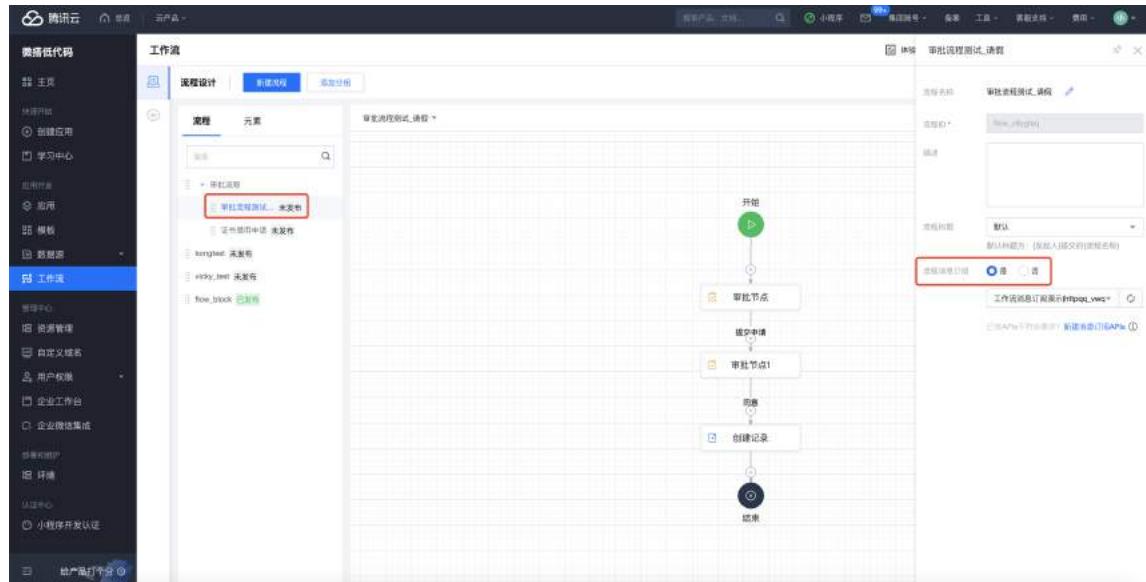
开启流程消息订阅功能

前往流程属性配置区，开启流程消息订阅配置。开启并选择 APIs 后，当流程&任务状态变更时会主动推送消息体至该 APIs 下 messageDeliver 方法内。

注意

选择的 APIs 需存在且满足如下命名规则的方法才可接收到推送消息：

- 方法标识：messageDeliver
- 入参：msgContext: json 字符串（消息内容）
- 出参：code: int (0 表示成功，非 0 表示失败)



新建消息订阅 APIs 模板

如环境下 APIs 不满足命名规则要求，可以新建消息订阅 APIs 模板。

如图，单击创建消息订阅 APIs：

流程名称 审批流程测试_请假 

流程ID * flow_c8yfqeq

描述

流程标题 默认
默认标题为：{发起人}提交的{流程名称}

流程消息订阅 是 否
工作流消息订阅演示: http://http://qq_vwq 
已有APIs不符合需求 [新建消息订阅APIs](#) 

默认生成 HTTP 请求类型的 APIs，需提供请求 URL、方法 Method。创建后可前往该 APIs，根据业务调整请求类型。

新建消息订阅APIs

APIs名称 * 工作流消息订阅

APIs标识 * http://qq_vwq  

方法名称 * 消息订阅 

方法标识 messageDeliver

方法类型 HTTP请求

方法URL *

方法Method * 请选择Method

推送内容

推送时机

流程实例和任务状态变更时，会触发消息推送。

推送消息体

流程实例变更时（例如：流程发起、撤回、终止、完成、作废等）

字段	类型	描述
msgId	string	消息 ID
msgType	string	消息类型： Instance: 流程实例 Task: 任务状态
action	int	操作
actionName	string	操作名称
processName	string	流程名称
processKey	string	流程标识
flowTitle	string	流程标题
flowCode	string	流程编码

startTime	date	开始时间
endTime	date	结束时间
instanceId	string	实例 ID
envType	string	环境类型
sourceType	string	请求来源
requestId	string	请求 ID
approverMap	list<map<string, string>>	当前任务审批人
startUserMap	list<map<string, string>>	流程发起人
elementMap	list<map<string, string>>	当前节点列表
instanceStatus	int	流程实例状态
instanceStatusName	string	流程实例状态名称

任务状态变更时（例如：任务提交、同意、拒绝、转办、加签、回退、撤回等）

字段	类型	描述
msgId	string	消息 ID
msgType	string	消息类型： Instance: 流程实例 Task: 任务状态
action	int	操作
actionName	string	操作名称
processName	string	流程名称
processKey	string	流程标识
flowTitle	string	流程标题
flowCode	string	流程编码
startTime	date	开始时间
endTime	date	结束时间
instanceId	string	实例 ID
envType	string	环境类型
sourceType	string	请求来源
requestId	string	请求 ID
approverMap	list<map<string, string>>	当前任务审批人
startUserMap	list<map<string, string>>	流程发起人
taskStatus	int	任务状态
taskStatusName	string	任务状态名称
elementId	string	当前节点 ID
elementName	string	当前节点名称

elementType	string	节点类型
elementTypeName	string	节点类型名称
comment	string	审批意见
turnToUserMap	list<map<string, string>>	任务转签人
addAssigneeMap	list<map<string, string>>	任务加签人
ccUserMap	list<map<string, string>>	任务抄送人
taskMap	list<map<string, string>>	当前审批人任务
pageUrlMap	list<map<string, string>>	任务跳转链接
taskStartTime	date	任务开始时间
taskEndTime	date	任务结束时间
rollbackElementId	string	回退目标节点 ID
rollbackElementName	string	回退目标节点名称
revokeElementId	string	撤销目标节点 ID
revokeElementName	string	撤销目标节点名称
isCcTask	boolean	是否为抄送节点任务

流程触发方式

数据模型触发

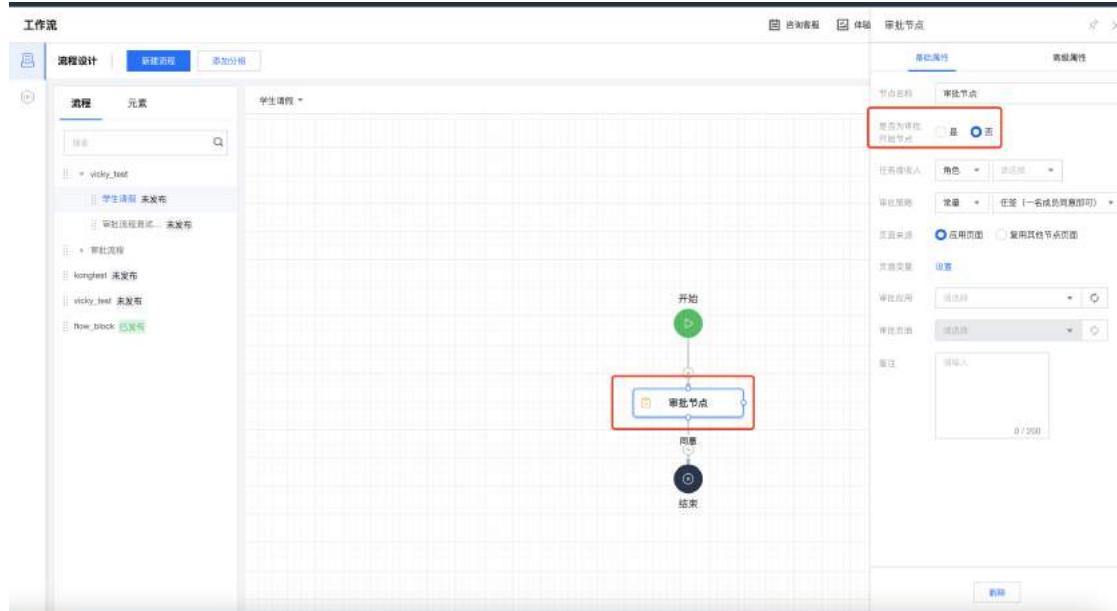
最近更新时间：2023-07-27 15:10:56

应用场景

当根据业务需求，数据模型执行一个操作后，触发一个流程。如提交请假后，触发请假流程。

操作步骤

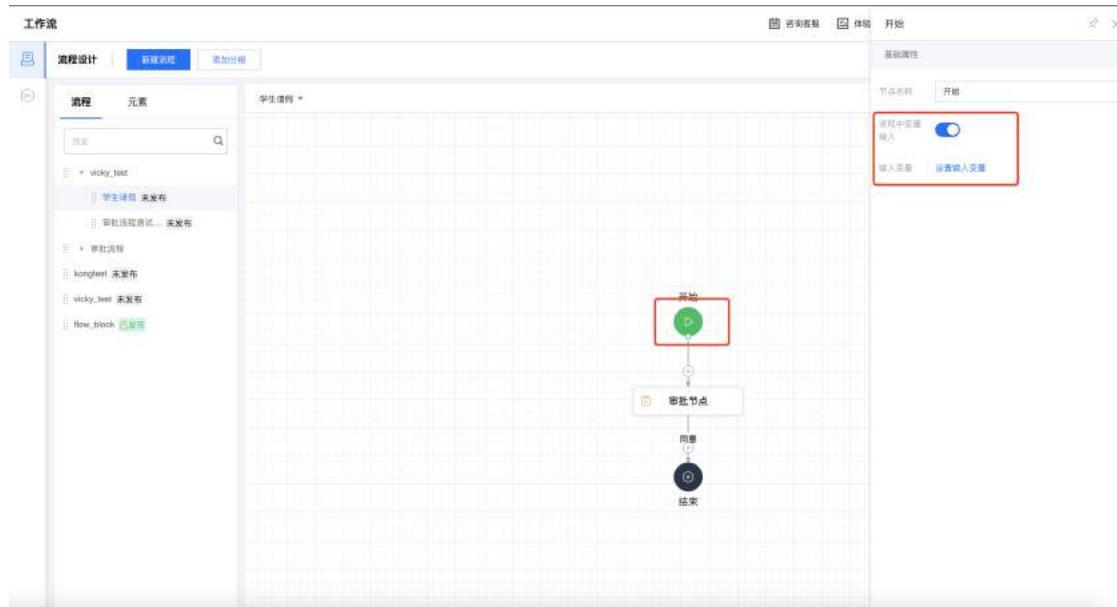
1. 配置流程。添加审批节点，将是否为审批开始节点设为否。



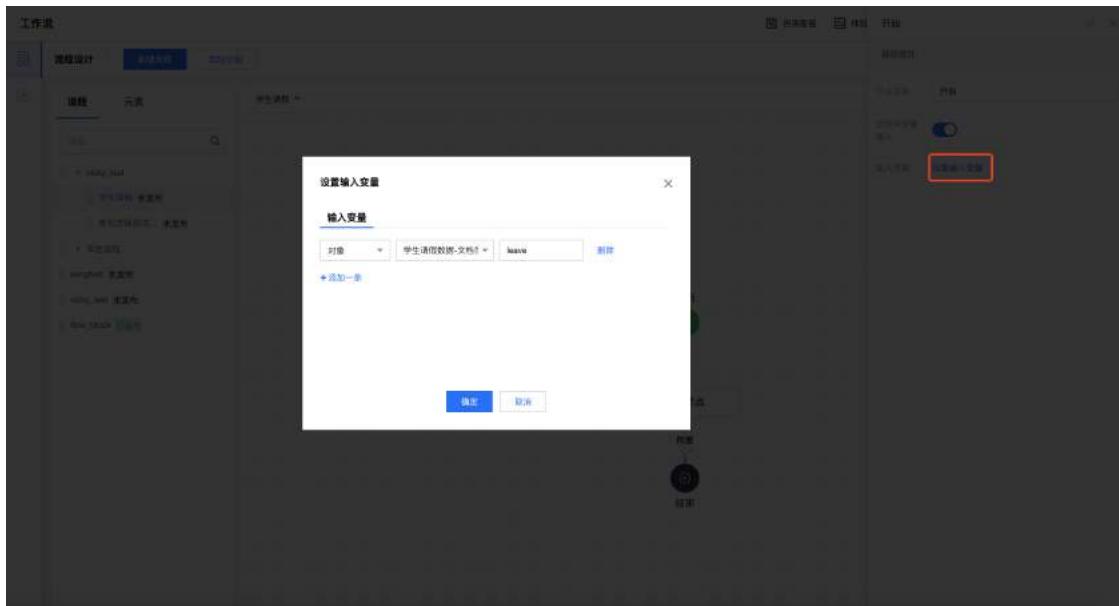
△ 注意：

数据模型触发流程情况下，流程中的第一个节点不能设置为审批开始节点。

2. 设置流程入参。



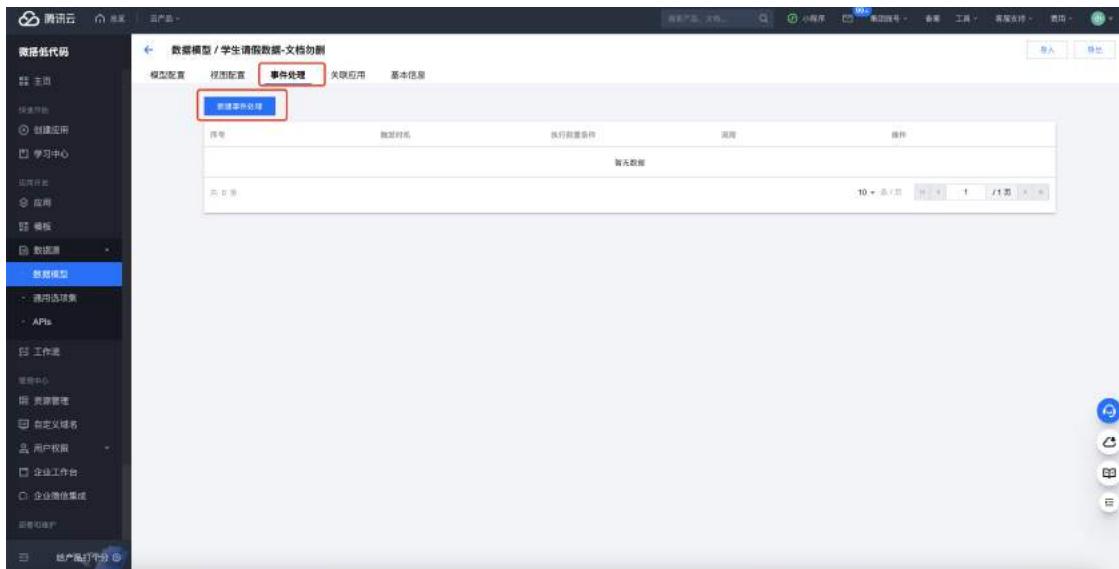
设置输入变量后保存流程。



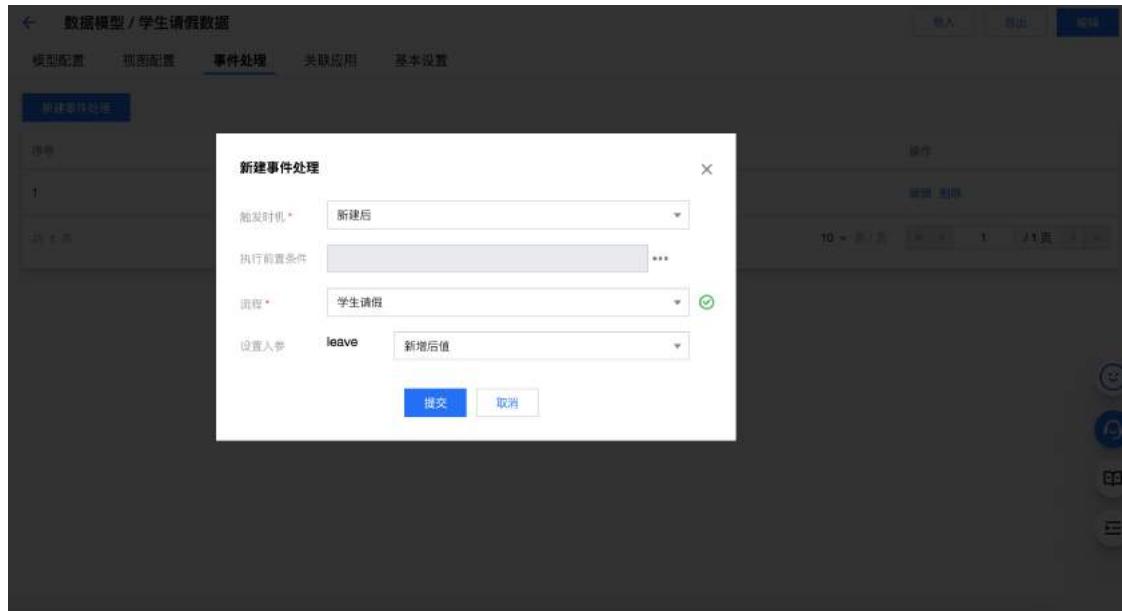
△ 注意：

为保证数据模型可触发，开始节点只能配置对象、对象列表类型字段。

3. 在数据模型下单击新建事件处理创建事件触发流程：



填写相关信息后单击提交。



② 说明：

- **执行前置条件：**当前条目的执行条件。当该表达式计算结果为真时，会触发执行对应的流程。单击执行前置条件的输入框，在弹出的表达式编辑器中输入条件表达式。

表达式编辑器

函数列表	表达式	表达式使用帮助	参数列表
#field3.field2 > 0			UpdatedTime field1 field2 field3.entity001 所有人 所属部门 创建人 修改人 field3.createdTime field3.updatedTime field3.field1 field3.field2 field3.field3 field4
IsNull()	描述：判断公式将返回空值 (null)，如果为空则返回 TRUE，不为空则返回 FALSE 使用示例：IsNull(#a)	语法动态	日期时间 文本 文本 文本 所有人 所属部门 创建人 修改人 日期时间 日期时间 文本 数值 日期时间 日期时间

- **触发时机：**当前条目关注的实体事件，可选项包括创建后、编辑后、删除后等。单击触发时机的输入框，选择对应的处理流程。
- **选择流程：**选择事件对应的处理流程。
- **设置入参：**给流程传递新增后的数据。

定时触发

最近更新时间：2023-03-09 15:58:00

功能简介

通过定时任务设置，定时触发执行某一流程。执行流程可选择一次性执行或某一段时间内按时间间隔，周期性重复执行该流程。

定时任务管理

登录 微搭控制台，单击工作流，再单击左侧功能栏中的定时任务进入定时任务管理页面。

新建定时任务

进入定时任务管理页面，点击新建，创建定时任务。

属性说明：

属性	说明
名称	定时任务名称
状态	启用/未启用，创建时默认启用，可手动停用

执行类型	支持一次性、周期性两种执行类型
首次执行时间	定位任务首次执行时间，注意：应用重新发布后，定时任务才能生效
执行周期	<ul style="list-style-type: none">• 执行类型为"周期性"时，必填执行周期，• 定时任务按执行周期间隔，重复触发当前定时任务• 执行周期单位可选择：年、月、天、小时、分钟、秒。
有效期	<ul style="list-style-type: none">• 执行类型为"周期性"时，必填有效期，定时任务有效期，• 在有效期内按执行周期重复触发执行当前定时任务的执行流程。• 有效期最大可选择到2037年。
执行流程	当前定时任务执行的流程。未有输入参数的流程方可选为执行流程。
描述	定时任务描述。

一次性定时任务

根据所填首次执行时间，执行当前定时任务所选流程，流程只触发执行一次，流程首次执行完成，不会再此执行流程。

新建定时任务

名称

状态

执行类型 一次性 周期性

首次执行时间

执行流程

□描述

确定 取消

周期性定时任务

根据首次执行时间与有效期时间范围，按照执行周期重复循环执行所选流程。

- 执行周期：执行周期最大单位到年，最小可到秒。

- 有效期：最大有效期到2037年。

编辑定时任务

名称

状态

执行类型 一次性 周期性

首次执行时间

执行周期 每 天 执行一次

有效期
开始时间需要小于等于首次执行时间，结束时间需要大于首次执行时间

执行流程

描述

确定 取消

停用定时任务

定时任务进入定时任务管理页面，单击停用，停用所选定时任务。

定时任务

① 定时任务新增、编辑、删除、启用停用等设置在应用发布后方可生效。

新建

名称	状态	执行类型	首次执行时间	执行周期	流程	描述	最近修改时间	操作
sta	已启用	一次性	2021-12-06 02:16:17	-	请假流程		2021-12-06 02:15:34	编辑 停用 删除

共 1 条 10 条 / 页 1 / 1 页

删除定时任务

定时任务进入定时任务管理页面，单击删除，删除所选定任务。



The screenshot shows the 'Scheduled Tasks' management interface. At the top, there is a note: '定时任务新增、编辑、删除、启/停用等设置在应用发布后方可生效。' (Scheduled task creation, editing, deletion, enable/disable settings take effect after application publishing). Below this is a search bar with placeholder text '请输入名称' and a magnifying glass icon. A blue button labeled '新建' (Create New) is visible. The main area contains a table with the following data:

名称	状态	执行类型	首次执行时间	执行周期	流程	描述	最近修改时间	操作
sta	已启用	一次性	2021-12-06 02:18:17	-	流程流程		2021-12-06 02:16:34	编辑 停用 删除

At the bottom of the table, it says '共 1 条' (1 item) and has a page navigation section with '10 条 / 页' (10 items per page), page number '1', and a total of '1 页' (1 page).

定时任务应用

定时任务新增、编辑、删除、启/停用等设置在应用发布后方可生效。

页面事件触发

最近更新时间：2023-08-11 11:23:27

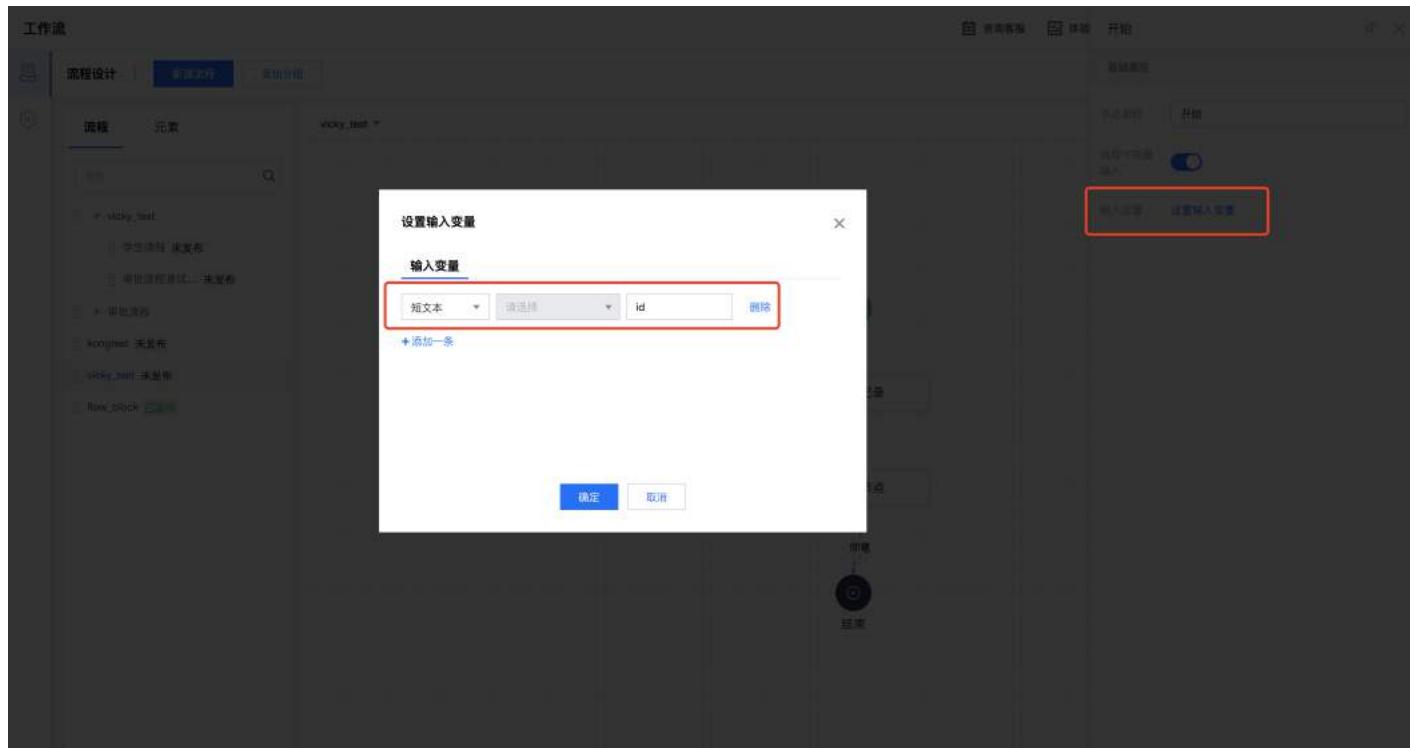
功能简介

微搭编辑器提供了触发流程的内置平台方法请参见[执行动作-触发流程 callProcess](#)，可以通过页面组件事件触发一个流程的执行。

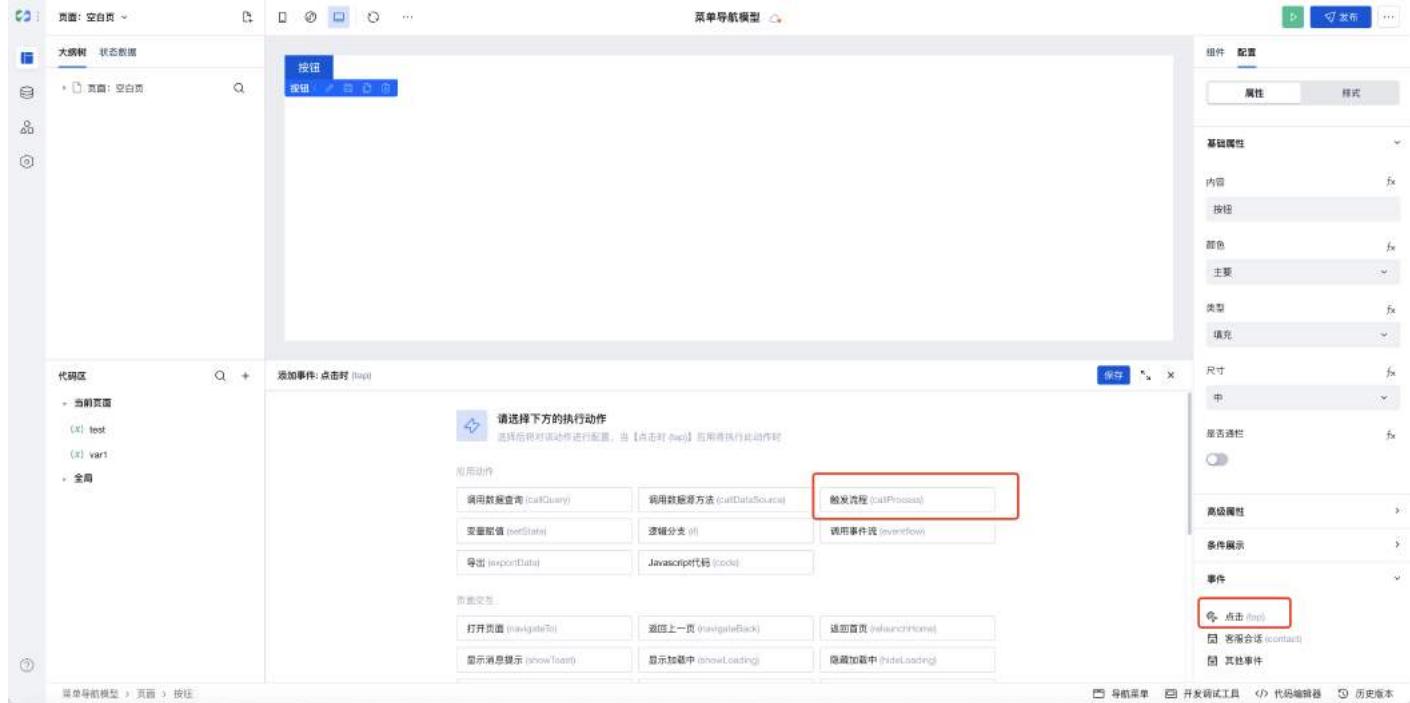
场景示例

操作场景：通过按钮单击事件去触发一个开始节点带输入变量的流程。

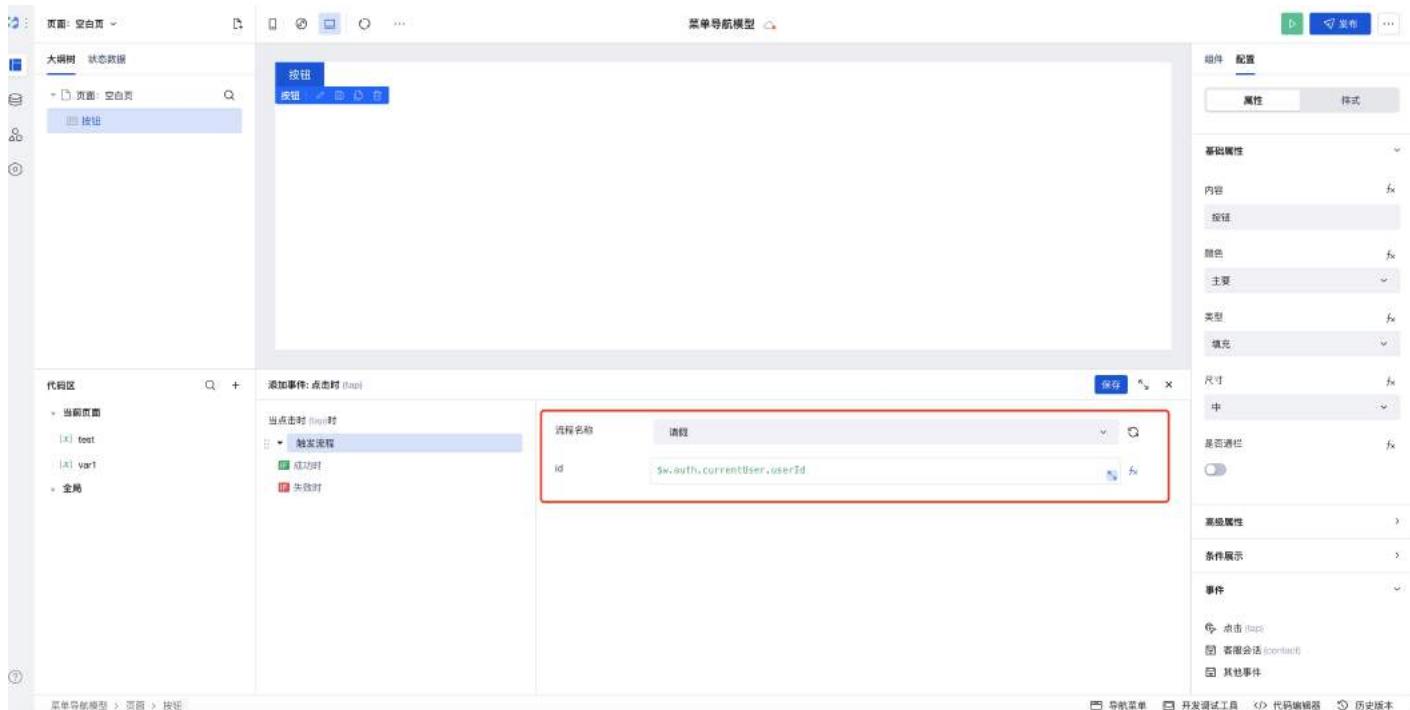
1. 创建名称为请假的流程，给流程开始节点设置输入变量 id，例如。



2. 编辑器中使用按钮组件，通过按钮事件进行赋值，选择触发流程后进入配置页面。



3. 触发流程中，流程选择请假，配置入参 id。



4. 保存后单击按钮即可触发流程，其中编辑器配置的 id 值将会作为流程的输入变量。

自定义 API 触发

最近更新时间：2023-07-12 17:19:47

功能简介

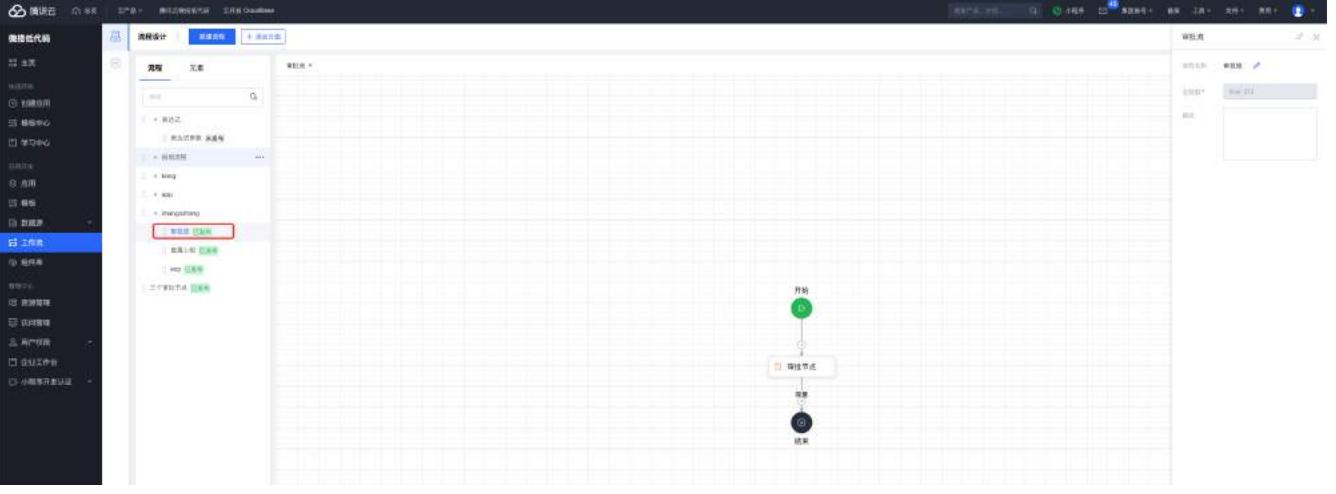
自定义 API 的自定义代码可以触发流程实例。

场景示例

1. 配置流程：流程开始节点支持配置变量：

⚠ 注意：

如果仅在方法测试、应用编辑器的实时预览或者应用的预览环境触发流程，保存流程即可，不需要发布；如果要在应用的正式环境触发流程，需要发布流程。



设置输入变量

输入变量

短文本	请选择	start_reason	删除
-----	-----	--------------	----

+ 添加一条

确定 **取消**

2. 配置自定义 API。

The screenshot shows the 'APIs / startflow' configuration page. The 'APIS方法' tab is selected. The '名称' field contains 'startFlow', and the '描述' field contains 'method_65d91583fc619'. The code editor displays the following JavaScript code:

```

18  /*
19   * const response = await context.callWorkflow({action: 'StartProcessWithParams', data: {
20     TriggerType: 6,
21     ConnectorParam: {
22       ConnectorName: "startflow_n17f6cw",
23       FunctionName: "method_65d91583fc619"
24     },
25     ProcessKey: "flow_ttmzhe5",
26     StartParams: [
27       {
28         "name": "form_test3",
29         "paramCode": "168049347798818754",
30         "fieldType": "object",
31         "fieldValueMap": {
32           ...
33         }
34       }
35     ]
36   })
37   */
38 
```

Below the code editor, there is a table for input parameters:

字段名称	字段标识	数据类型	是否必填	操作
testName	testname	文本 单行文本	否	<input checked="" type="checkbox"/>

At the bottom right of the interface, there are several icons for export, print, and copy.

参考下面的代码配置 APIs 和工作流相关信息即可触发流程，可以通过方法测试验证，验证后通过“出参映射”可以快速配置 API 的出参。

```

/**
* 使用 npm 包 node-fetch 发送http请求, 详细使用文档可以参考
* https://github.com/node-fetch/node-fetch
*/
const fetch = require('node-fetch');

module.exports = async function (params, context) {
// 这里是方法入参
console.log(params);

/***
* 可以在这里编写业务逻辑, 例如:
* 1. 使用 node-fetch 通过 HTTP 方式请求外部数据, 并对获取的数据进行加工;
* 2. 使用 context.database API 来直接操作云开发云数据库;
* 3. 使用 context.callModel 来操作其他数据模型数据;
* 4. 使用 context.callConnector 来使用 API ;
* 5. 使用 context.app.callFunction 来调用同环境的云开发云函数;
*/
const response = await context.callWorkflow({action: 'StartProcessWithParams', data: {
TriggerType: 6,
ConnectorParam: {
ConnectorName: "startflow_n17f6cw",
FunctionName: "method_65d91583fc619"
},
ProcessKey: "flow_ttmzhe5",
StartParams: [
{
"name": "form_test3",
"paramCode": "168049347798818754",
"fieldType": "object",
"fieldValueMap": {
...
}
}
]
}}

```

```
"_id":'50b8a5676425824e001b06661838b821',
"xh":'1311311236743'
},
],
ExtraUserInfo: {
UserId: 10000000000000000000,
Source: 4
}
});
return response;
};
```

其中：

- ConnectorName：触发的 APIs 标识。

← APIs / startflow

The screenshot shows the 'APIs / startflow' page. At the top, there are tabs: 'APIs方法', '鉴权设置', '关联应用', and '基本信息'. The '基本信息' tab is highlighted with a red border. Below the tabs, the page title is '基础信息' with an '编辑' button. The details are listed in a table:

名称	startflow
标识	startflow_n17f6cw
状态	未发布 (立即发布)
上次发布时间	-
描述	-
创建时间	2023-04-21 16:06:09

- FunctionName: 触发的方法标识。

The screenshot shows the 'APIs / startflow' interface. Under 'APIs方法', a new method 'startFlow' is being created. The 'Method Identifier' field is highlighted with a red box and contains the value 'method_65d91583fc619'. The code editor displays a sample code snippet:

```

28     "name": "form_test3",
29     "paramCode": "168049347798818754",
30     "fieldType": "object",
31     "fieldValueMap": {
32       "_id": "5b0ba5676425824e01b06661b38b821",
33       "xh": "1311311236743"
34     }
35   },
36   ],
37 };
38   return response;

```

Below the code editor, there's a table for '入参' (Input Parameters) with one entry: 'testName' (字段名称), 'testname' (字段标识), '文本 | 单行文本' (数据类型), and '否' (必填). There are also tabs for '测试' (Test) and '方法测试' (Method Test).

- ProcessKey: 流程的唯一标识。

The screenshot shows the '流程设计' (Workflow Design) interface. A workflow named 'test3' is displayed. The process starts with a green '开始' (Start) node, followed by a task node labeled 'test3串...'. From the task node, a transition labeled '提交申请' (Submit Application) leads to a black '结束' (End) node.

- StartParams中：
- name: 开始节点变量名称。
- paramCode: 开始节点变量标识（获取方式参考以下截图示例，对照接口出参中 rightId 字段内容）。
- fieldType: 开始节点变量类型。

- fieldValueMap: {_id: 数据模型记录 ID, 数据模型字段 key-value}。

The screenshot shows a browser window with two main panels. The left panel is a 'Process Design' interface with a sidebar containing user names like 'mangouzhang', 'test001', 'old', '11', 'vicky', '999', 'bobby', 'test.', 'jingsu', '1', 'kongtest', '学生请假', '审批流程_请假', and '物品采购'. A specific process named 'test0' is selected and highlighted with a red box. The right panel shows the 'Network' tab of the developer tools, displaying a list of API requests. One request, with the URL 'http://flow_tianzhe5/api/test?processId=168...&rightId=168...', is highlighted with a red box and has its details expanded. The response body contains JSON data related to the process start.

```
code: "NORMAL", result: (RequestId: "b068bd7a-3d33-4692-bc41-2b625e88befd", code: "NORMAL", reqId: "d922fa89-4bcc-482a-a925-ebe18f71e255", result: (RequestId: "b068bd7a-3d33-4692-bc41-2b625e88befd", ..., * Data: (Status: 2, FlowCode: "1645267720369475587", ... description: "", EditJson: "\\"nodes\\":{\\\"size\\":156,\\\"labelCfg\\\":{},\\\"style\\\":{},\\\"FlowCode\\\":1645267720369475587", FlowProcessSetStr: "\\\"flowTitle\\\":\\\"#PROCESS_START_USER + '提交的测试3\\\"", Name: "test3", ProcessKey: "flow_tianzhe5", ProcessVersion: "", Status: 2, RequestId: "b068bd7a-3d33-4692-bc41-2b625e88befd"}
```

3. 应用中使用自定义 API。

自定义 API 发布后，在应用编辑器中可以配置表单容器。

什么是APIs?

APIs可以打通外部应用数据或调用外部应用开放的服务，开发者只需要对API完成授权或少量填写一些配置信息，即可使用对应产品的服务接口。

功能介绍 | 操作指南

名称	释义	类型	状态	发布状态	所属应用	创建时间	发布时间	操作
startflow	自定义	正常	未发布	vicky_test	2023-04-21 16:06:09	-	-	<button>编辑</button> <button>更多</button>
lovelist	自定义	正常	未发布	-	2023-04-20 11:21:27	-	-	<button>编辑</button> <button>挂起</button>
APIs实训示例-lylan	自定义	正常	未发布	vicky_test	2023-04-17 19:33:07	-	-	<button>编辑</button> <button>更多</button>
自定义代码	自定义	正常	未发布	-	2023-04-10 15:04:43	-	-	<button>编辑</button> <button>更多</button>
云开发云函数	自定义	正常	未发布	-	2023-04-04 16:44:28	-	-	<button>编辑</button> <button>更多</button>
计算当前位置到目标位置的距离	自定义	正常	未发布	小程序批量计算距离...	2023-04-03 18:17:00	-	-	<button>编辑</button> <button>更多</button>
HTTP请求	自定义	正常	未发布	-	2023-04-03 11:11:05	-	-	<button>编辑</button> <button>更多</button>
APIs实训示例-lylan	自定义	正常	未发布	APIs实训示例-lylan	2023-03-29 14:00:00	-	-	<button>编辑</button> <button>更多</button>

选择配置好的 API 方法，单击提交即可触发流程。

模型应用_0423 暂未保存到云端

属性 样式

表单容器

组件ID: form1

基础属性

表单场景

新增

数据源

功能模块: APIs

自定义API:

startflow (startflow_01790ew)

新方法:

startFlow[method_id_05d91583fc8119]

布局方式: 水平

字段:

全选

testName

免检/循环提示

如果需要在自定义应用中使用，需要先配置登录组件，确认应用登录授权信息。

The screenshot shows two main sections of the Tencent Cloud Low-Code Platform:

- Identity Management:** A table listing identity providers:

认证组件	认证类型	状态	描述
匿名管理	匿名用户认证	已启用	启用对匿名登录用户的认证，用户无需提供任何身份信息
用户基础设置	基础用户认证	已禁用	基础认证方式禁用，可以通过账户名密码登录
短信验证码登录	短信验证码登录	已启用	启用短信验证码登录后，可以通过短信验证码登录
- Application Settings:** The "Login Settings" tab is selected. It includes:
 - A toggle switch for enabling login functionality.
 - Three tabs: 登录页设置 (selected), 访问授权, and 安全设置.
 - Two radio button options:
 - 无需登录访问应用 (Unauthenticated access)
 - 需登录后访问应用 (Authenticated access) (selected)
 - A dropdown for external user default permissions set to "默认外部用户".
 - A note about roles: 无合法角色? 立即新建.
 - A "Save" button at the bottom right.

可通过单击应用编辑器右上角发布成功 > 访问链接进行访问。

⚠ 注意:

如果不配登录组件，只能在自定义代码中指定用户触发，即传入 ExtraUserInfo 参数的 UserId 和 Source (如上文示例所示)。如果已经配置了登录组件则不需要配置该入参。

流程元素

基础元素

最近更新时间：2023-05-06 10:24:41

开始节点

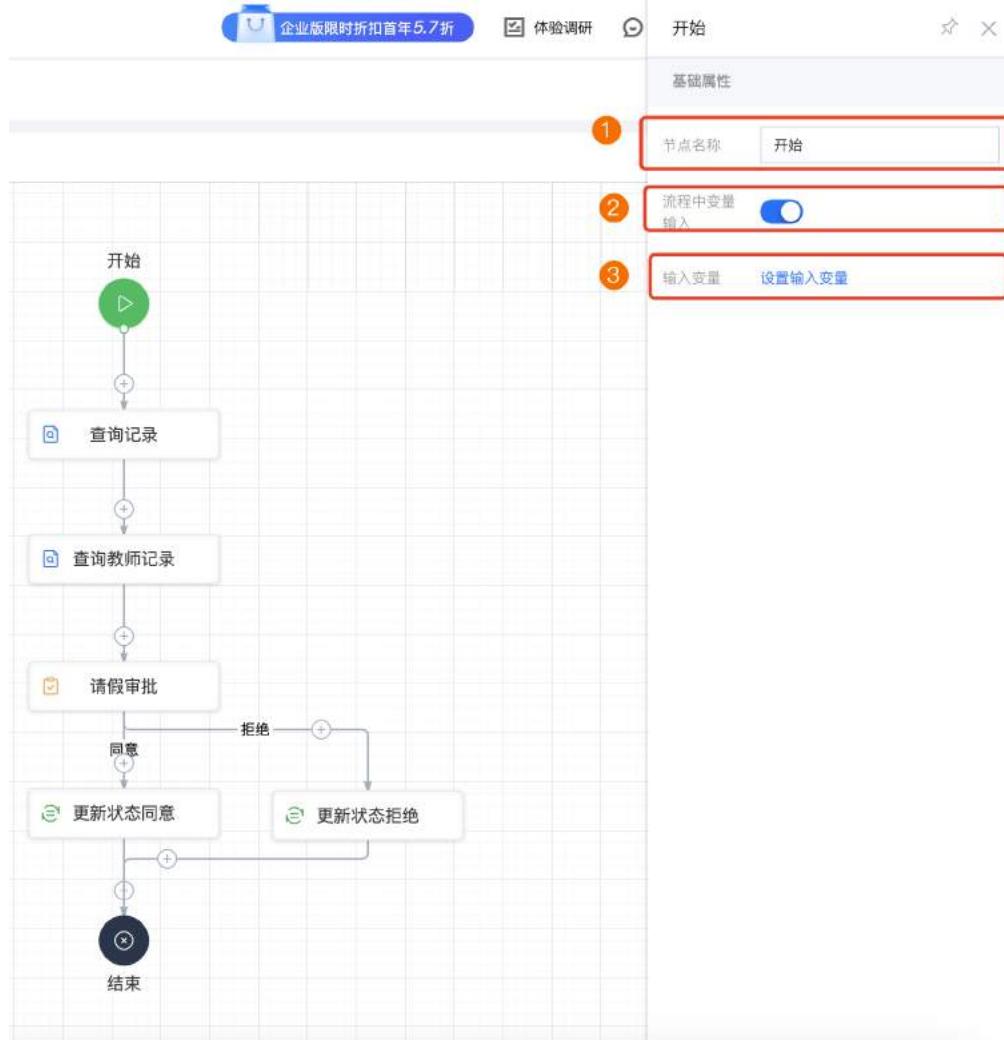
功能介绍

多参数接收流程启动入参，供后续节点使用。

参数类型	参数说明
对象类型	可以是 N 个不同或者 N 个相同的对象。
对象列表	可以是 N 个不同或者 N 个相同对象的列表。
文本类型	可以是 N 个不同或者 N 个相同的文本。
实数类型	可以是 N 个不同或者 N 个相同的实数。
日期类型	可以是 N 个不同或者 N 个相同的日期。
整数类型	可以是 N 个不同或者 N 个相同的整数。
图片类型	可以是 N 个不同或者 N 个相同的图片。
文件类型	可以是 N 个不同或者 N 个相同的文件。
关联类型	可以是 N 个不同或者 N 个相同的关联对象。
jsonSchema 类型	可以是 N 个不同或者 N 个相同的 jsonSchema。

属性面板

基础属性可选择不设置变量输入。



序号	描述
1	节点名称
2	是否设置变量输出
3	设置变量具体信息

变量设置

可以是 N 个不同或者 N 个相同的对象、对象列表或文本、日期等。



序号	描述
1	选择变量类型
2	选择变量
3	设置变量名称

结束节点

功能介绍

多参数输出流程变量。

参数类型	参数说明
对象类型	可以是 N 个不同或者 N 个相同的对象。
对象列表	可以是 N 个不同或者 N 个相同对象的列表。
jsonSchema 类型	可以是 N 个不同或者 N 个相同的 jsonSchema。

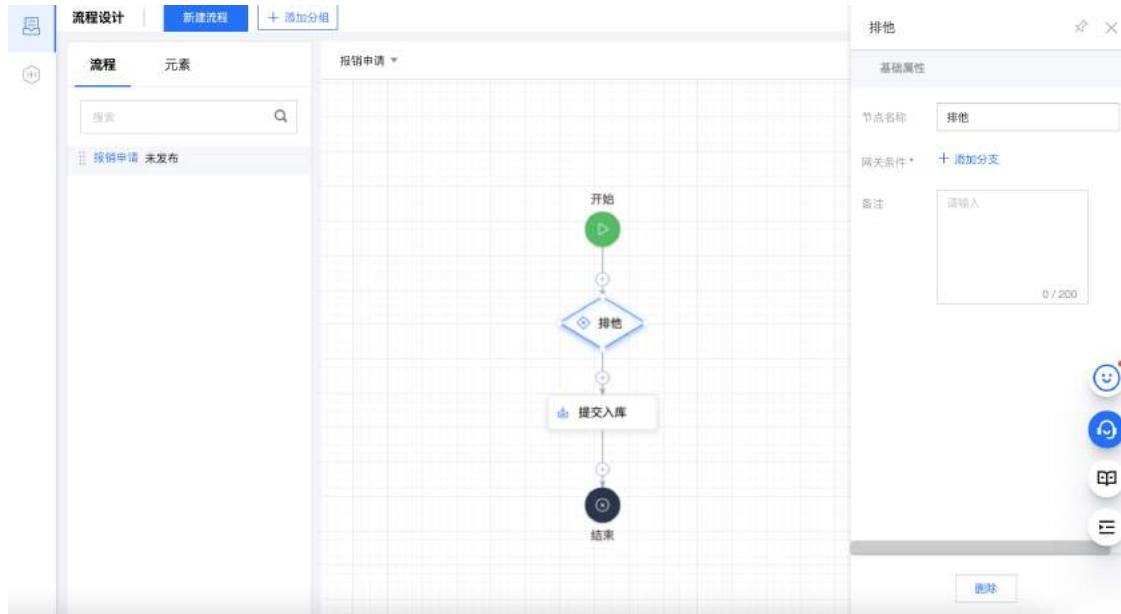
排他网关

操作场景

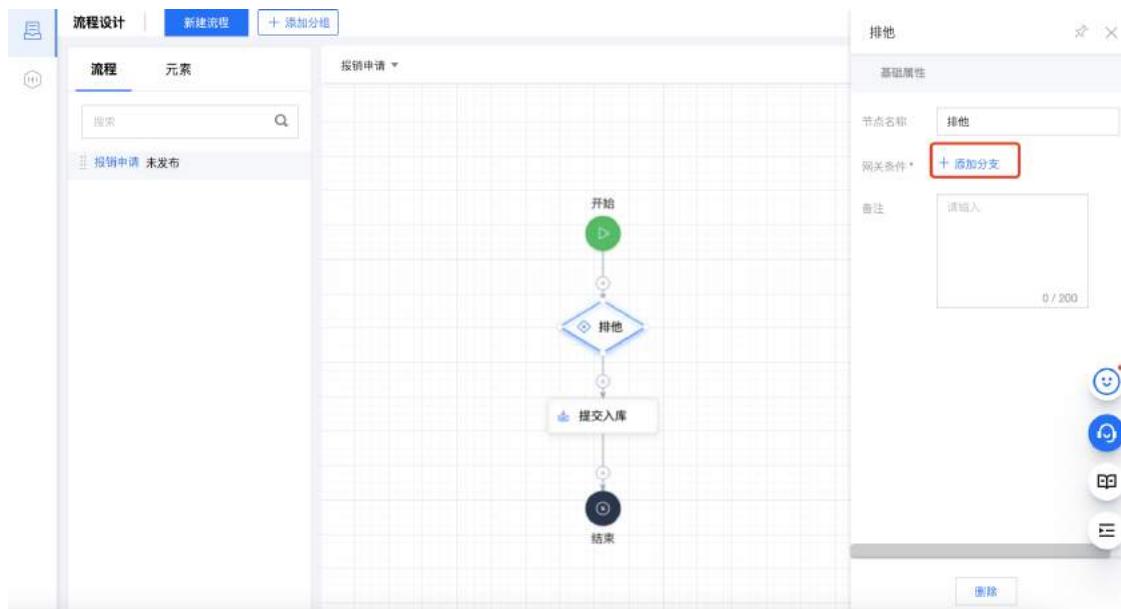
当根据业务需求，流程逻辑执行过程中，根据不同的条件需要走不同的后续分支流程时，可增加“排他网关”节点实现。设置步骤如下：

操作步骤

1. 将排他网关节点拖入画布区，选中排他网关，可在右侧属性区设置节点名称。



2. 单击添加分支，可弹窗添加分支条件，如步骤3。添加完成，即以列表形式展示于右侧。



3. 设置分支条件，包括名称与条件。可通过普通条件/特殊公式两种方式设置，常用简单的条件可直接采用普通条件设置即可，复杂条件可通过表达式实现。普通条件：可视化选择前面变量，并根据变量类型设置变量值的条件，例如：等于、为空等。多个条件之间可采用 and/or 组合为分支规则，如“1 or 2”。

注意：

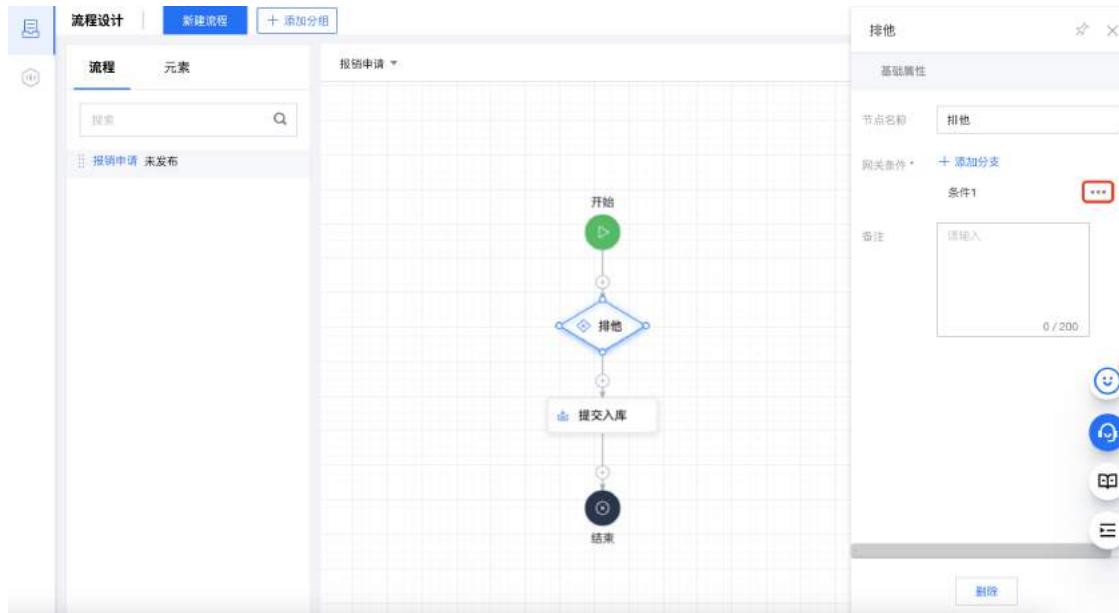
此处的1为“条件1”中的数字。



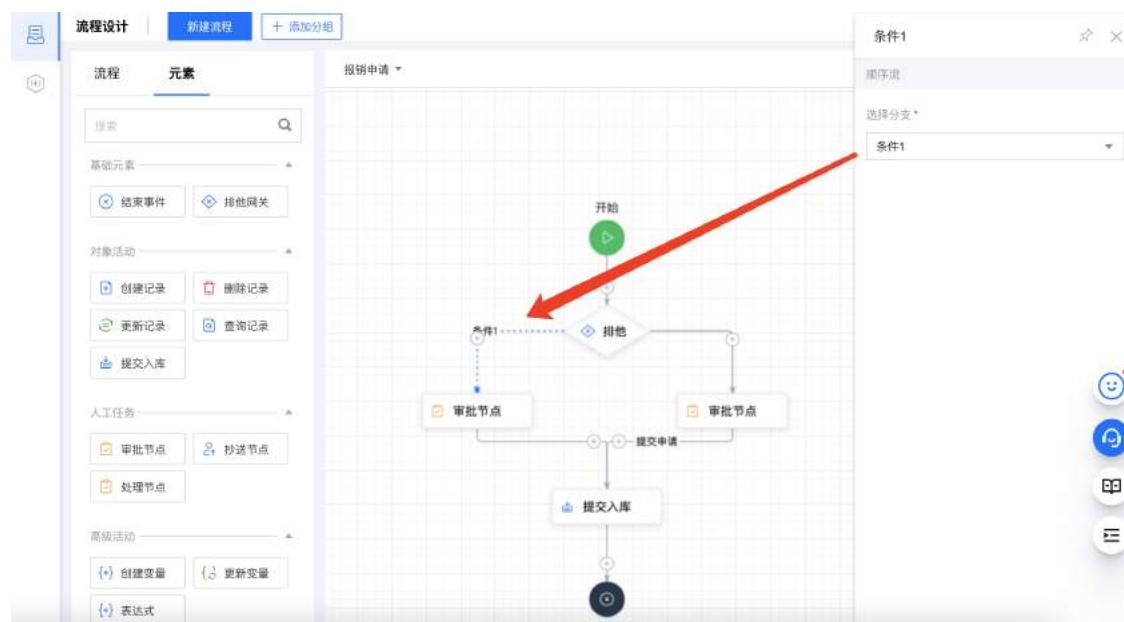
特殊公式：有开发能力的用户可撰写表达式，输出 true/false 等布尔值即可。



4. 用户可单击右侧分支列表中的，弹窗编辑分支条件信息。



5. 在排他网关之后拖入对应的节点活动，连接分支线，然后选中连接线，右侧属性区可选择具体的分支条件。选择后，连接线上出现所选分支名称，即满足该分支条件时即执行该连接线的后续节点活动。



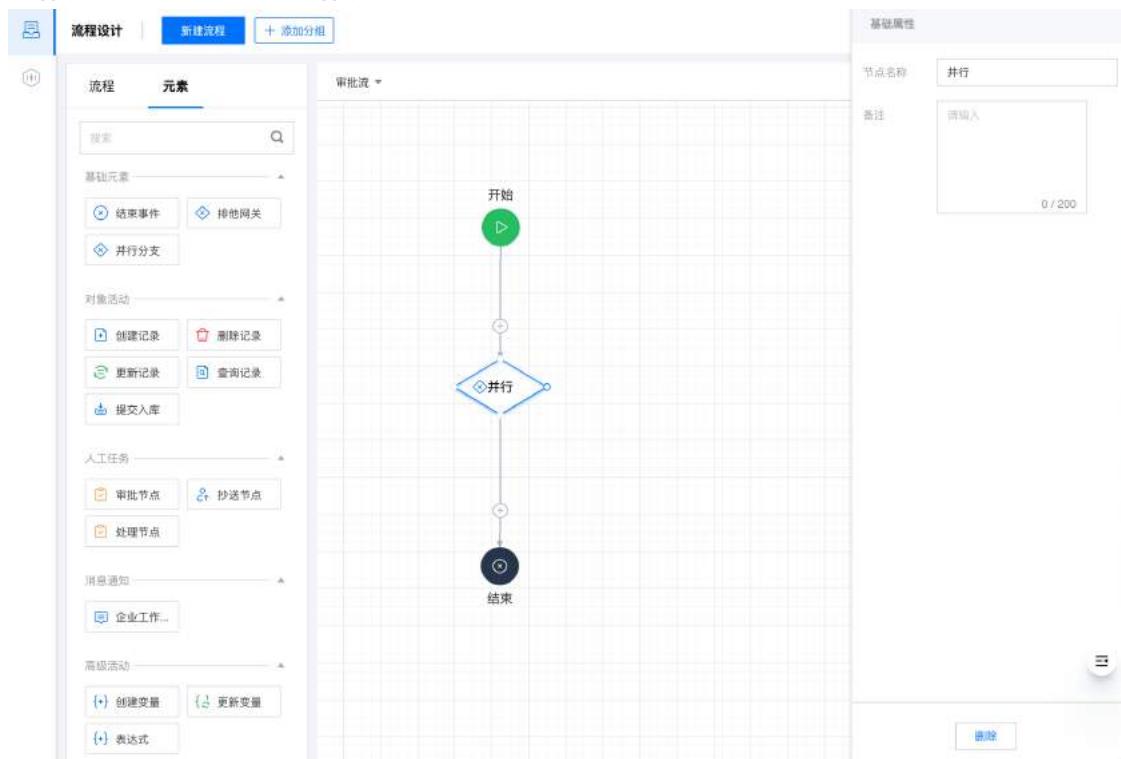
并行分支

操作场景

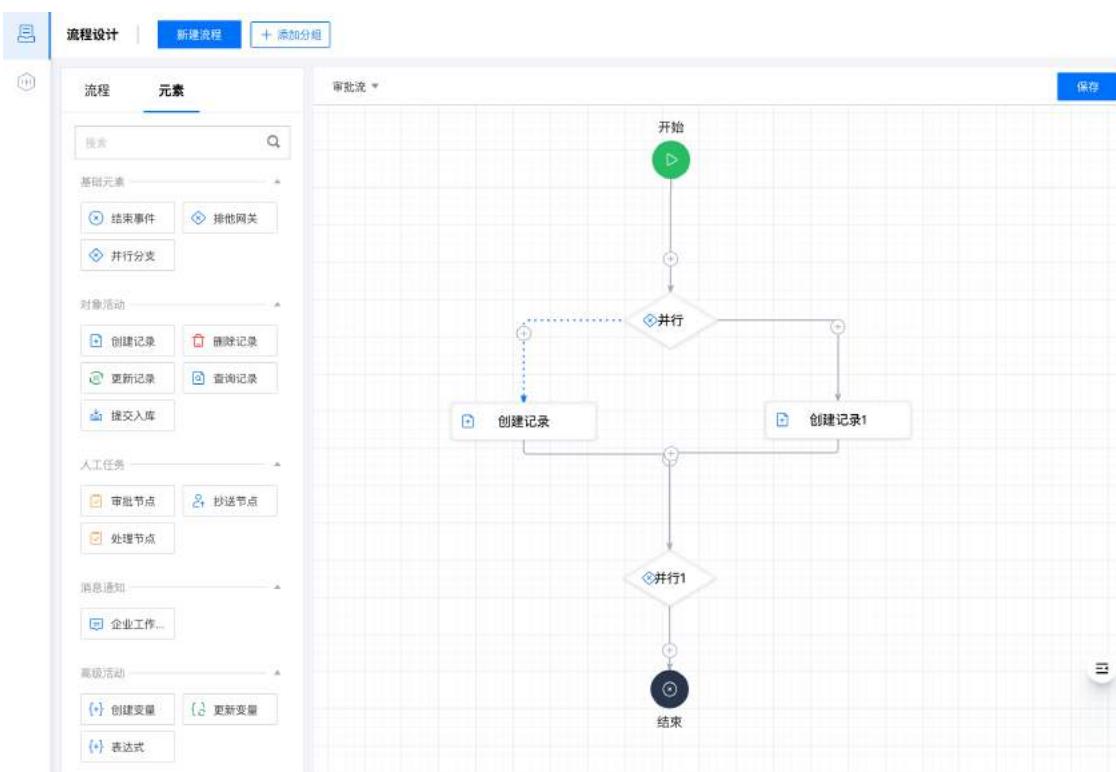
当根据业务需求，流程逻辑执行过程中，需要并行走向不同的分支流程时，可增加“并行分支”节点实现。设置步骤如下：

操作步骤

- 将并行分支节点拖入画布区，选中并行分支，可在右侧属性区设置节点名称。



- 在并行网关节点之后拖入对应的节点活动，并连接分支线。最后拖入一个新的“并行分支”节点，作为流程汇合点。该节点会等待全部分支都执行完成后才继续执行后续节点。

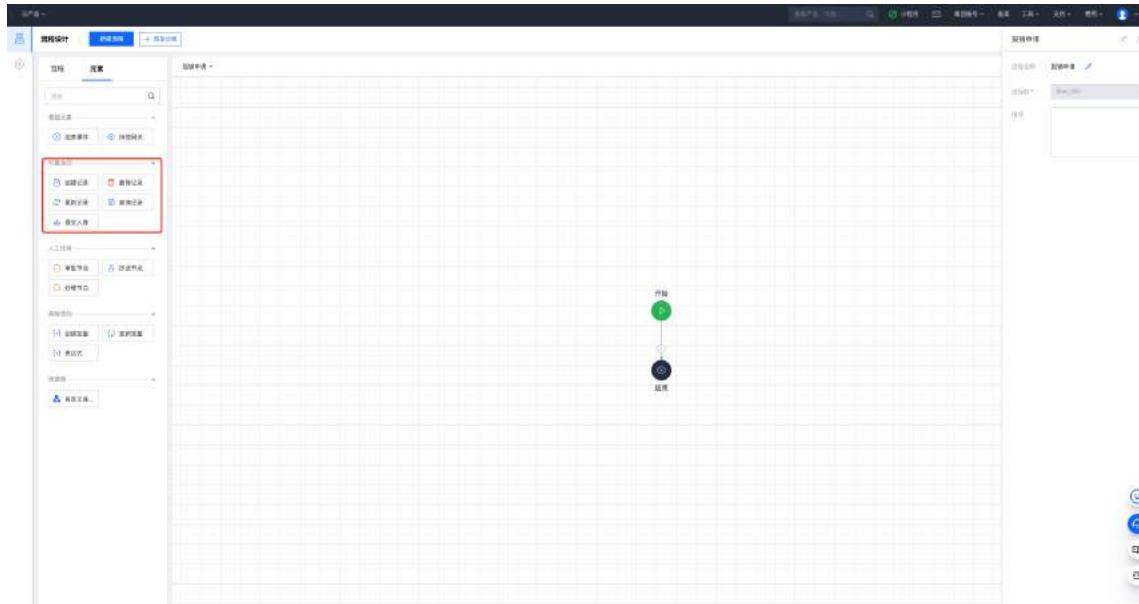


对象活动

最近更新时间：2023-07-12 17:20:49

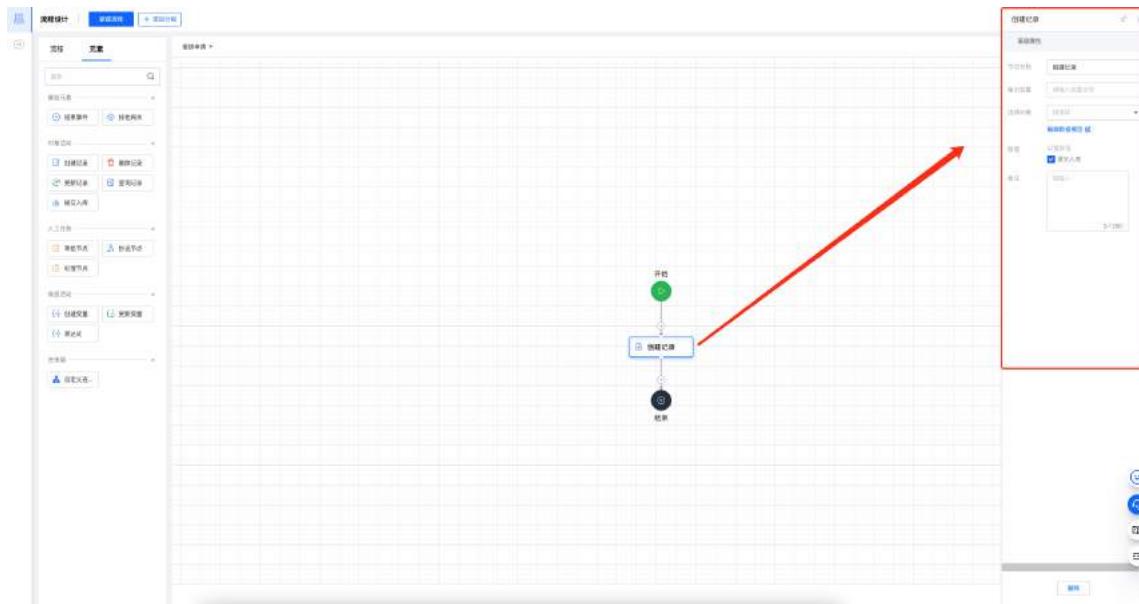
功能介绍

对象活动节点是在流程编排中操作数据模型中的业务数据的节点。目前包括：创建记录、删除记录、更新记录、查询记录、提交入库共 5 个节点。这些对象记录的操作会持久化到数据模型的数据存储中。5 种对象节点配合其他流程节点灵活编排，能实现复杂的对象数据读写操作。



创建记录

创建记录节点可以创建数据模型的一条记录。单击右侧“创建记录”将节点拖拽到流程画布中，此时右侧会展示对象节点的属性配置，包括基础属性和赋值配置。

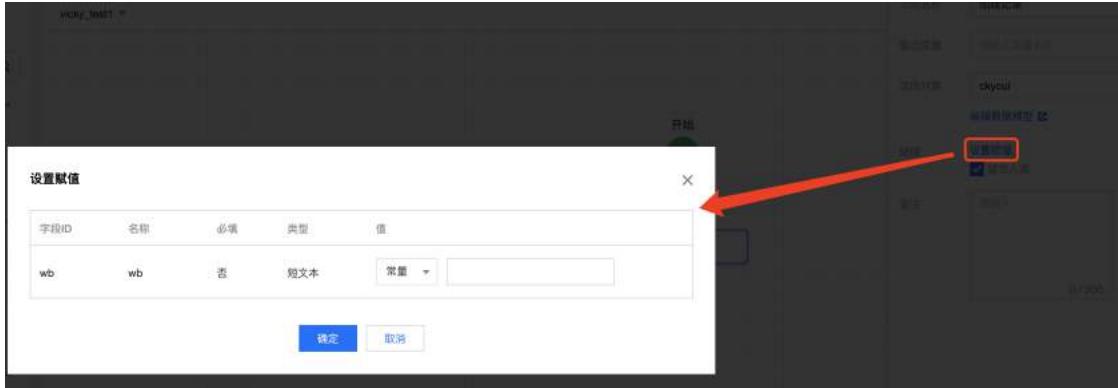


基础配置

- 节点名称：流程节点的名称标记。
- 输出变量名称：创建记录节点除了在数据存储中会创建一条记录，还会生成一个流程变量，该变量的唯一标记在此处定义，可供流程下游节点使用。
- 选择对象：此处选择需要创建哪一种对象模型的业务数据。

字段赋值

1. 选择需要创建的对象类型之后，单击设置赋值，在弹出的设置赋值框中，可对需要创建的对象的字段进行赋值操作。



2. 在设置赋值弹窗中，会自动加载该对象所有的字段信息，包括：字段 ID、字段名称、字段是否必填、字段类型、右值，右值支持三种类型赋值，包括：

- 常量赋值：例如 1、“abc”、0.12 等。
- 变量赋值：流程上游节点输出的流程变量值，包括流程入参变量。
- 系统变量赋值：目前支持“当前用户”、“当前日期”两个系统变量。

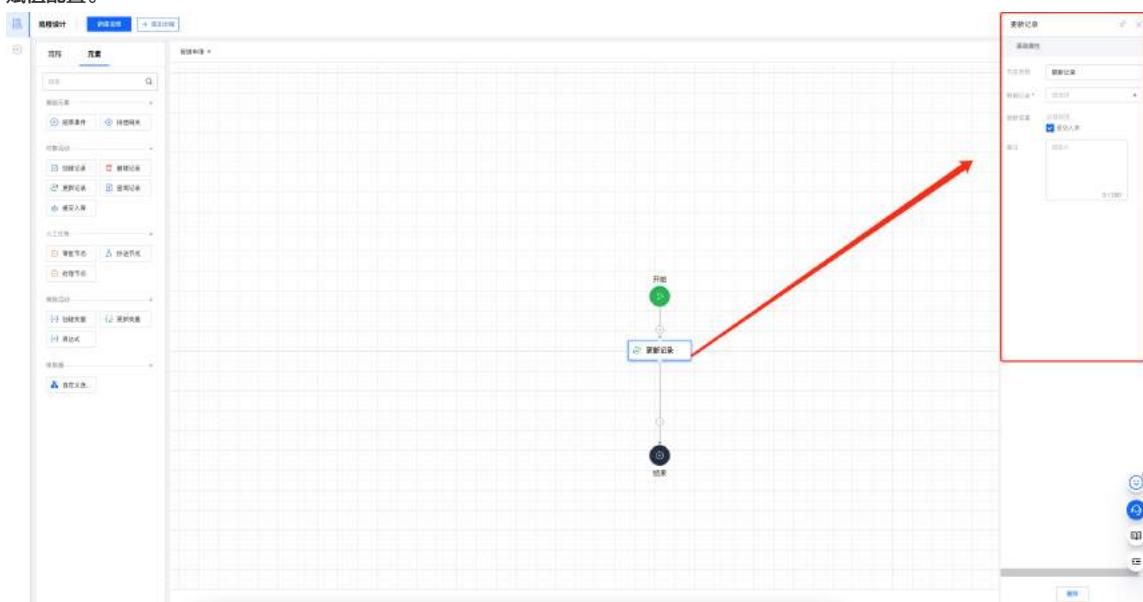
3. 单击确定后，即字段赋值完成。

在对象赋值时：

- 右值的类型需要和左值类型保持一致。
- 必填为“是”的必须进行赋值，必填为“否”的可以不进行赋值。

更新记录

更新记录节点可以更新单条或者批量更新多条对象记录。单击右侧更新记录将节点拖拽到流程画布中，此时右侧会展示更新记录节点的属性配置，包括基础属性和赋值配置。



基础配置

- 节点名称：流程节点的名称标记。
- 对象记录：选择要更新的记录信息。此处可以选择对象类型的流程变量，包括流程对象或者对象列表。

字段赋值

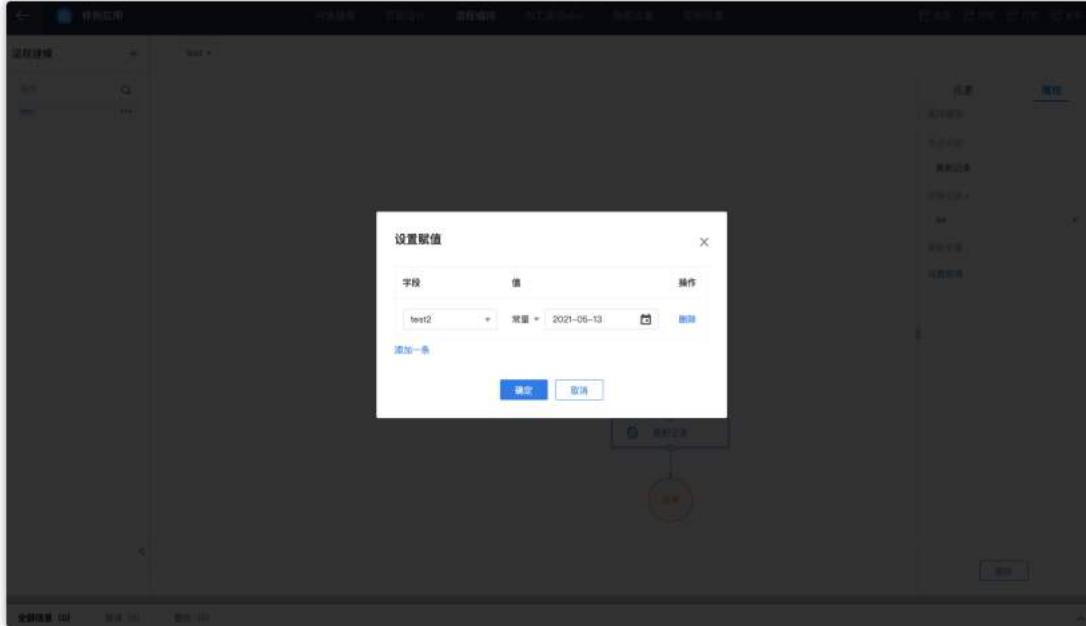
1. 单击设置赋值，会弹出字段赋值弹窗进行字段赋值。和创建对象不同，更新记录默认不展示对象字段。需要手动选择字段，并进行右值赋值。右值支持四种类型赋值，包括：

- 常量赋值：例如 1、“abc”、0.12 等。
- 变量赋值：流程上游节点输出的流程变量值，包括流程入参变量。
- 系统变量赋值：目前支持“当前用户”、“当前日期”两个系统变量。

- 为空赋值。

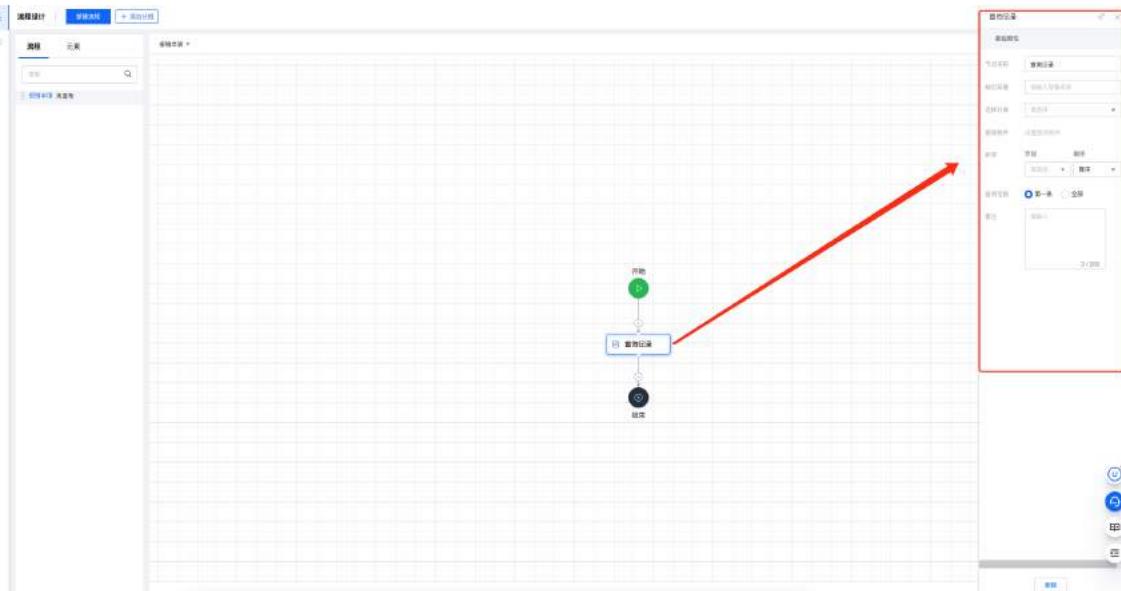
2. 单击确定后，即字段赋值完成。

- 对象赋值时：右值的类型需要和左值类型保持一致。
- 至少设置一条更新字段。



查询记录

查询记录节点可以查询数据存储中的业务数据。单击右侧“查询记录”将节点拖拽到流程画布中，此时右侧会展示查询记录节点的属性配置，包括基础属性和查询条件配置。



基础属性

- 节点名称：流程节点的名称标记。
- 输出变量名称：创建记录节点除了在数据存储中会创建一条记录，还会生成一个流程变量，该变量的唯一标记在此处定义，可供流程下游节点使用。
- 选择对象：此处选择需要创建哪一种对象模型的业务数据。
- 查询范围：可以选择查询其中一条记录或者查询全部记录。

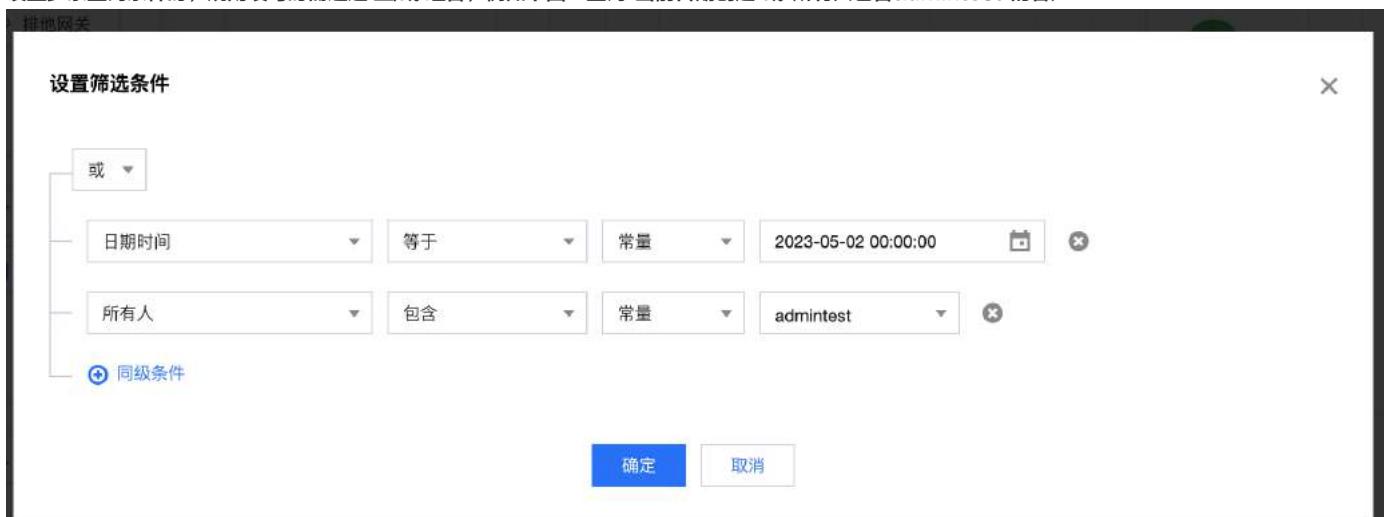
查询条件设置

1. 单击设置查询条件，会弹出查询条件弹窗配置查询字段。查询条件需要设置左值字段、操作符和右值，并根据查询条件填写规则：

设置单个查询规则，规则填写：1，如下图：



◦ 设置多条查询条件时，规则填写时需通过 且/或 组合，例如下图：查询“当前日期创建”或“所有人包含admintest”的客户



左值选择对象字段，查询记录默认不展示对象字段，需要手动选择字段。

- 操作符支持等于/不等于、为空/不为空、包含/不包含、属于/不属于等，根据字段类型不同选择对应支持的操作符。
- 右值的选择和创建/更新记录对象类似，支持三种类型赋值，包括：常量赋值：例如 1、"abc"、0.12 等。
 - 变量赋值：流程上游节点输出的流程变量值，包括流程入参变量。
 - 系统变量赋值：目前支持“当前用户”、“当前日期”两个系统变量。

2. 单击确定后，即字段赋值完成。

- 在对象赋值时请注意：右值的类型需要和左值类型保持一致。
- 至少需要设置一条查询字段。

3. 单击确认后，查询条件的配置会被保存。

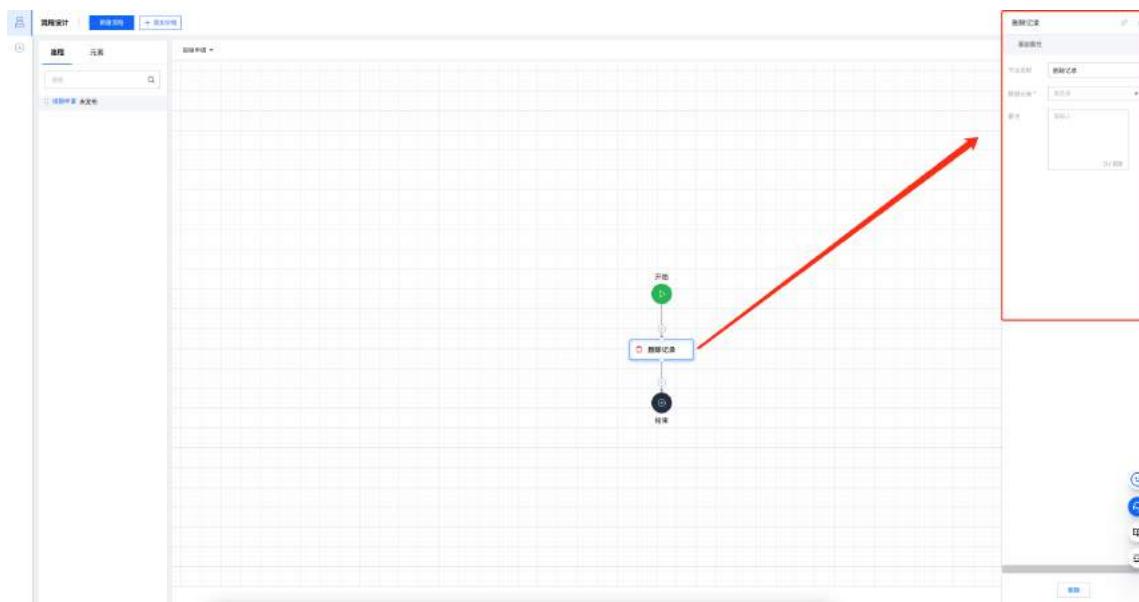
例如：查询年龄为25的所有是自己带饭的用户。



删除记录

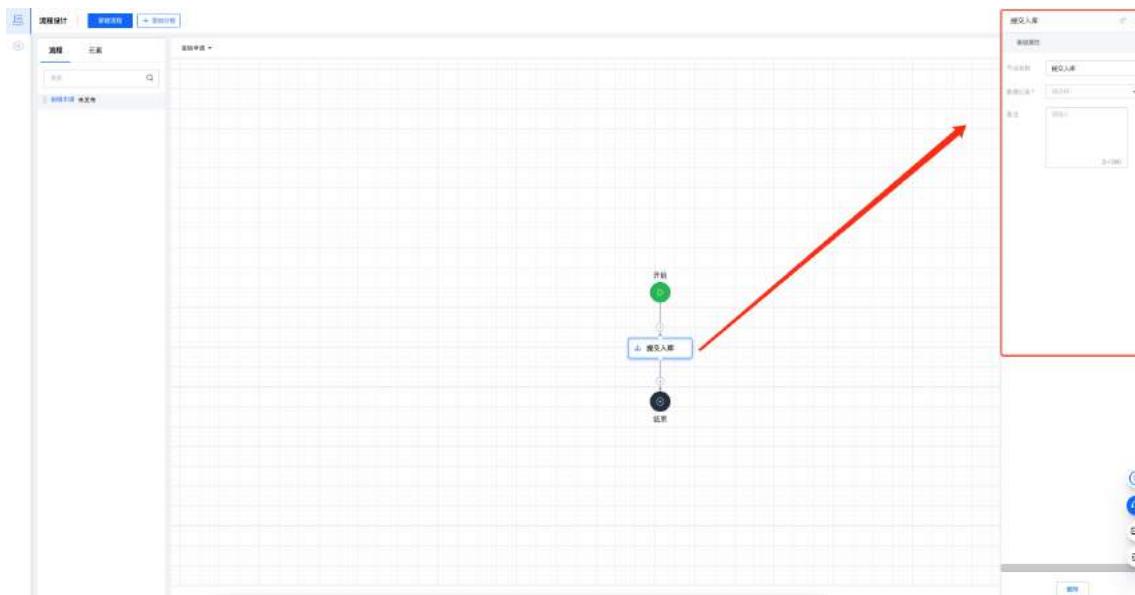
删除记录节点用来删除一条对象记录或者批量删除多条对象记录。单击右侧删除记录将节点拖拽到流程画布中，此时右侧会展示删除节点的属性配置，主要包括：

- 节点名称：流程节点的名称标记。
- 对象记录：选择要更新的记录信息。此处可以选择对象类型的流程变量，包括流程对象或者对象列表。



提交入库

提交入库节点将流程中对象类型的变量提交到数据模型数据中保存。提交入库和创建记录的区别在于，提交入库直接选择一个流程变量入库，不需要像创建记录做字段赋值操作，且支持主子表一起提交，操作起来比较简单，特别适合流程入参数据支持持久化的场景。



单击提交入库将节点拖拽到流程画布中，此时右侧会展示提交入库节点的属性配置，主要包括：

- 节点名称：流程节点的名称标记。
- 数据操作：目前仅支持新增场景。
- 对象记录：选择要新增的记录信息。此处可以选择对象类型的流程变量，包括流程对象或者对象列表。

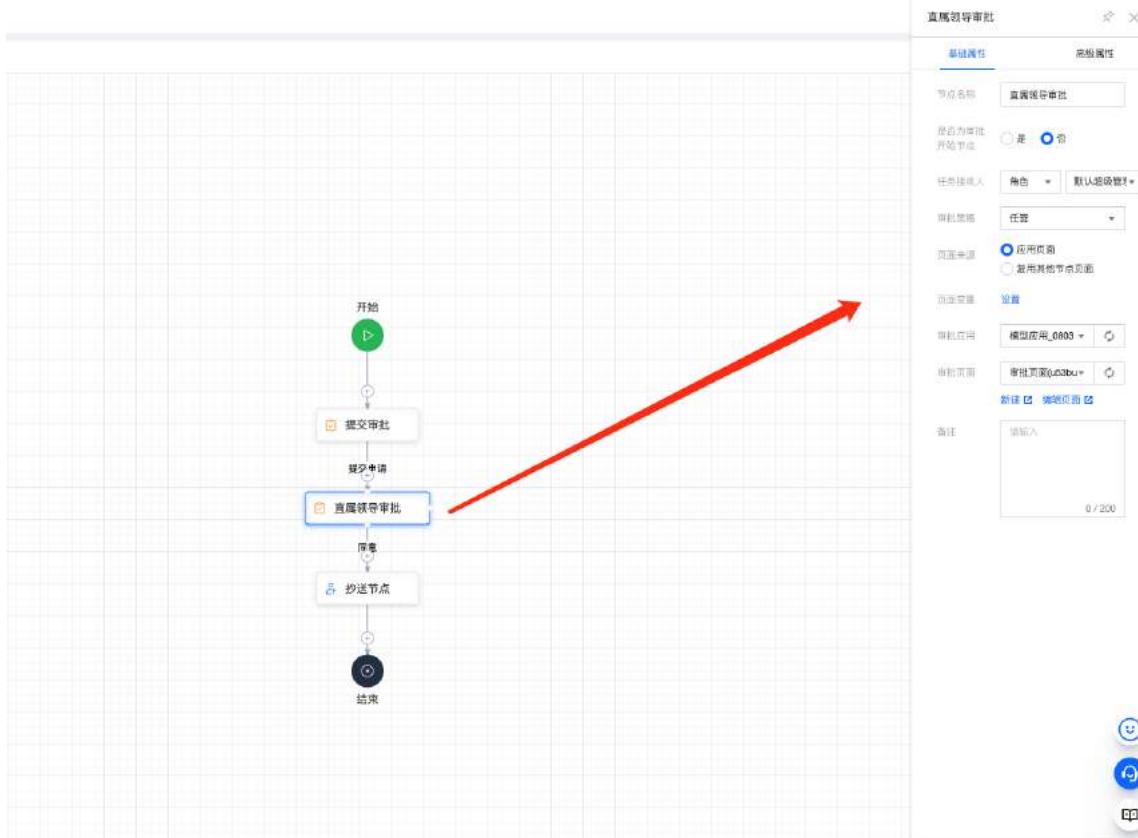
人工任务

最近更新时间：2023-07-13 09:56:39

审批节点

用户审批流配置，主要包括基础属性和高级属性配置。

基础属性



- 节点名称：**用户命名该审批节点，名称唯一。
- 是否为审批开始节点：**通用的审批流程，基本都会存在两种页面：
 - 发起人提交信息的提交页面（审批开始节点），例如：员工填写的请假单里信息，由员工提交。
 - 审核人进行内容审核的页面（非审批开始节点），例如：组长看到员工的请假单信息，组长可以单击同意或者拒绝，填写审批意见。
- 分支：**
 - 审批开始节点只有一个提交分支，可自行定义分支名称。
 - 非开始审批节点后会有两个分支：通过分支和拒绝分支。
- 任务接收人：**该审批环节任务操作人，可以选择以下几种类型作为任务操作人
 - 用户：设计态选择某个具体的用户作为审批人。

注意：

可以通过排序功能对用户进行排序，到达该节点操作人顺序以排序显示顺序为准。

- 角色：**设计态选择某个角色作为审批人，角色为变量，流程运行时将该节点任务分配给该角色下的用户。
- 系统变量：**常规审批场景中审批人，具有通用性，微搭产品已经定义好的变量，封装产品功能里，设计态选择审批人时可以直接选择，减少开发者通过自定义变量实现的复杂性：
 - 上一节点部门负责人：上一节点审批人所在部门负责人作为审批人

- 直属上级：直接上级作为审批人。
 - 表达式：设计态通过表达式定义审批人变量。
 - 用户变量：非常规审批场景，平台不会直接封装产品功能里，但是设计态可以通过定义审批人变量，开发者按照客户场景需求，例如用户表的某个字段值等于“xx”的用户。
- **审批策略：**目前支持以下策略：
- 任签：在多个审批人中只要其中一个审批人完成该节点操作（通过或驳回），该节点即为执行结束。
 - 会签：在多个审批人中所有审批人均需执行通过该节点，该节点即为通过，只要其中有人驳回，该节点即为审批驳回。
 - 依次会签：按照到达节点时任务接收人顺序进行依次审批，目前当任务接收人类型为用户、表达式和用户变量时可用。

⚠ 注意

依次会签下暂不支持加签功能。

- 表达式：设计态通过表达式定义审批策略变量。解析值需要满足以下规则：

- 值类型：数字
- 可选值：1：任签，2：会签，5：依次会签（该策略使用方法请参考以上依次会签相关介绍）

⚠ 注意

表达式能力需要购买 [专业版](#) 套餐。

• **页面来源：**

- 应用页面：在现有应用中创建或者选择一个**审批页面**类型的页面作为该节点的审批页面。
- 复用其他节点页面：在当前流程中已存在的节点已关联了一个审批页面，该节点可复用其中某个节点的页面，复用其他节点的页面变量可以不同，但是必须关联的是同一个数据模型对象。

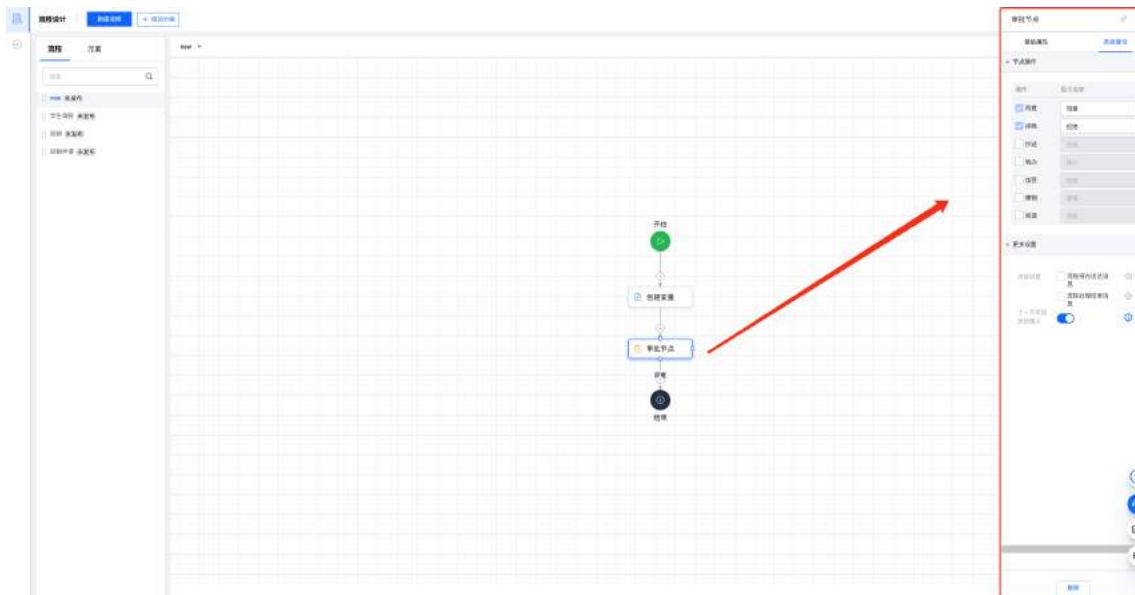
• **页面变量：**可设置页面输入变量和页面输出变量。

- 页面输入变量：从流程到页面的变量，可以将流程中的数据在页面表单里展示。
- 页面输出变量：从页面到流程的变量，可以将页面上的表单数据保存成流程变量。

• **审批应用：**审批页面必须隶属于一个应用，配置该审批节点关联的审批页面所属的应用

• **审批页面：**在编辑器中创建审批页面时，可根据该审批页面的出参变量对应的数据模型的数据结构，自动在审批页面中生成表单、及审批基本信息、审批记录、审批操作、流程图示等组件，用户可根据业务需求进行二次调整。

高级属性

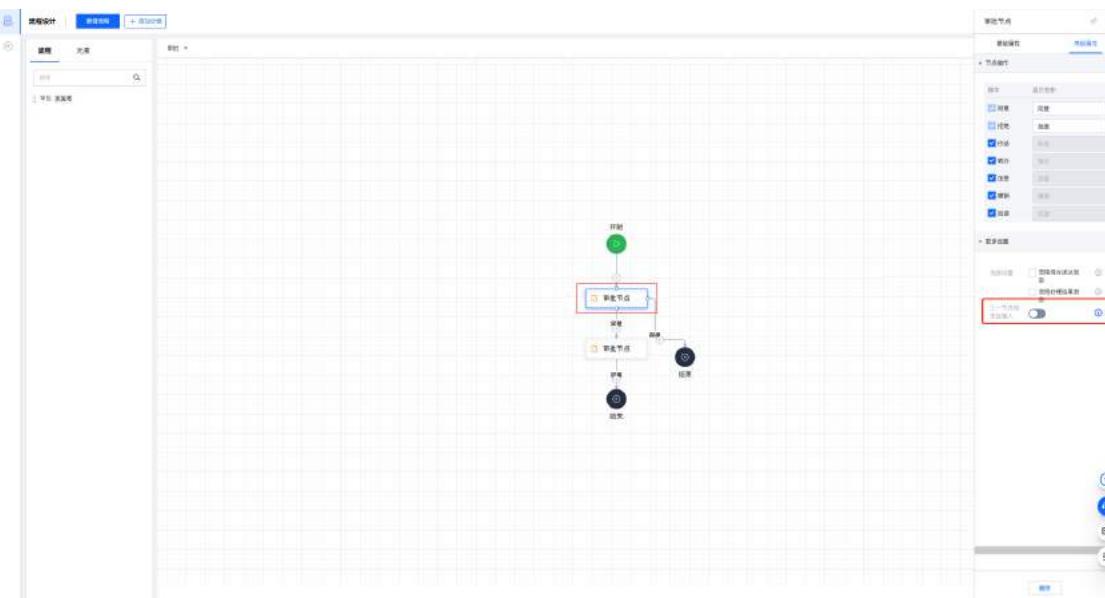


节点操作：通过配置节点操作，定义审批人在该环节可执行的操作，目前支持同意、拒绝、抄送、转办、加签、撤销、回退：

- 同意：运行态该节点审批人同意时，表示该审批人执行“通过”该节点。
- 拒绝：运行态审批人拒绝流程时，流程被驳回，流程结束。
- 抄送：审批人“同意”某一流程节点时，可将该审批节点抄送给其他用户。
- 转办：审批人可将当前审批节点转接给其他用户审批。
- 加签：审批人可添加当前流程节点审批人。
- 撤销：流程发起人可撤回未处理或未结束的流程。
- 回退：运行态当流程运行到该节点，审批人可通过回退操作，回退到流程之前已运行通过的节点。

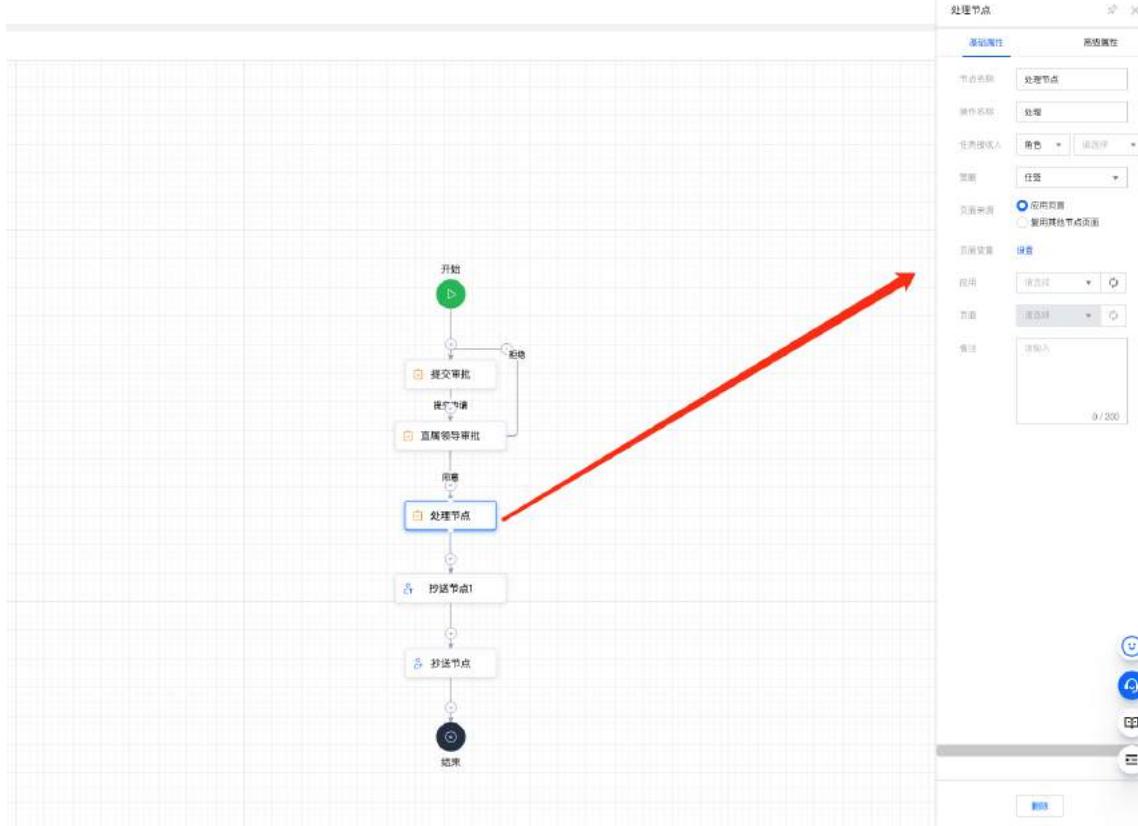
更多设置：

- 配置该节点待办通知发送渠道：
 - 流程待办送达消息：流程提交或流转时，自动发送审批待办消息给流程审批人。
 - 流程处理结果消息：流程审批结束后，自动发送审批处理结果消息给流程发起人。
- 该节点审批人是否由上一节点指定：流程在运行态时，上一节点处理完成，将由上个节点审批人从当前节点设定的处理人中选择具体处理人。流程中第一个审批节点不能配置“上一节点指定处理人”，由于上个节点是发起节点，否则会导致流程无法正常触发。



处理节点

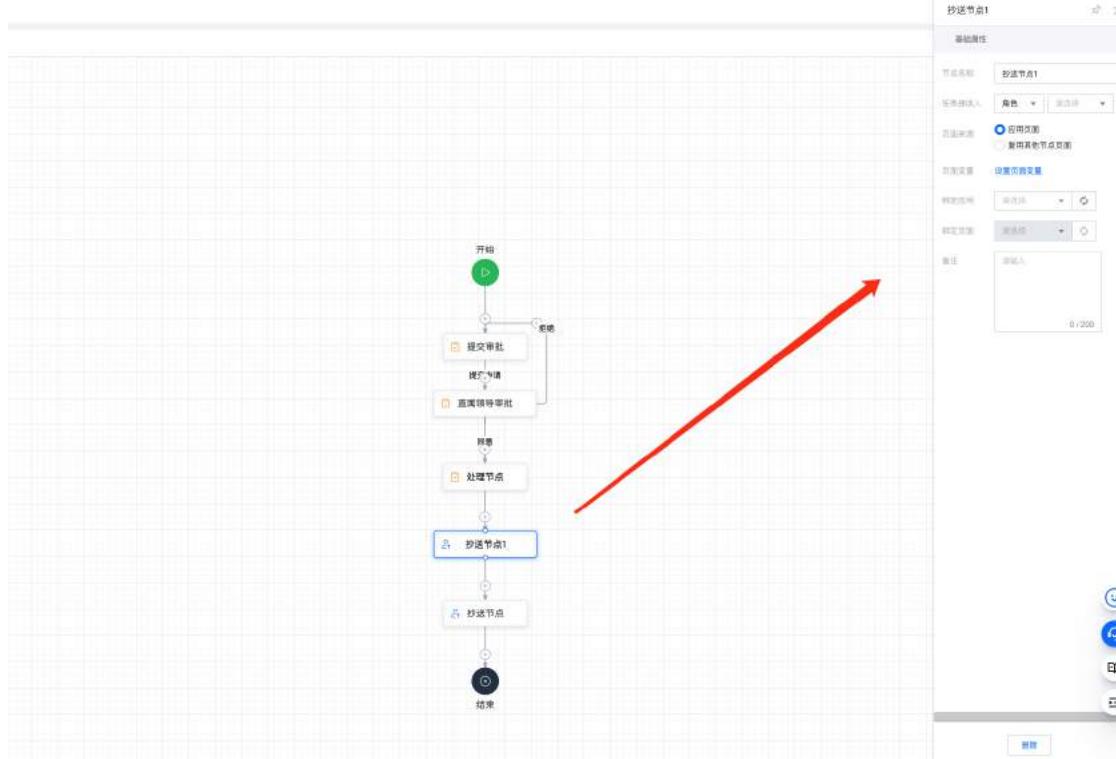
用户按照某流程规则执行某种业务流程，一般用于业务流程执行，可配置处理节点。配置属性和 **审批节点** 基本相同，可以配置操作名称为处理进行文案的展示。



抄送节点

抄送节点和审批节点高级属性中勾选抄送的区别：

- 审批节点高级属性勾选抄送，指在流程运行态时，审批人执行通过可将流程执行结果抄送给其他用户。
- 流程中新增抄送节点，指流程运行到该节点会通知抄送节点配置的相关用户。用户在某种流程中需要在设计态配置抄送通知的用户，可配置抄送节点。



⚠ 注意：

抄送节点前需要拖入审批节点或者处理节点才可正常运行，抄送节点页面是只读状态，因此只能设置输入变量。

高级活动

最近更新时间：2023-05-06 11:03:58

创建变量

- 功能介绍：创建一个变量，用于对象节点的赋值。暂时未支持结束直接输出一个简单变量。

- 变量类型**

- 短文本类型：字符串。
- 布尔类型：true/false。
- 数字类型：可以带小数位。
- 数组类型：长整数，0~ $2^{63}-1$ 。
- 对象类型：对象。
- 对象列表类型：由数据源变量组成的列表对象。

- 变量值类型说明**

- 变量：取前面节点输出的对应当前变量类型的属性值。
- 常量：一个具体的值，如 true/1.11/222/zhangsan。

- 属性面板设置举例布尔类型**

序号	描述
1	节点名称
2	变量名称
3	变量类型
4	变量值类型
5	变量值

The screenshot shows the 'Create Variable' configuration panel in the Tencent Cloud Low-Code Platform. On the left, there is a process flow diagram with nodes: '开始' (Start), a green circle with a play icon; a blue rounded rectangle labeled '创建变量' (Create Variable); and '结束' (End), a dark blue circle with a close/cancel icon. Arrows connect them in sequence. On the right, the 'Create Variable' configuration panel has the following fields:

- 基础属性 (Basic Properties) section:
 - 1 节点名称: 创建变量
 - 2 输出变量: 请输入变量名称 (Please enter variable name)
 - 3 变量类型: 布尔 (Boolean) dropdown menu
- 变量值 (Variable Value) section:
 - 4 变量值: 常量 (Constant) dropdown menu
 - 5 值: true
- 备注 (Remarks) section: 请输入 (Please input) with a character limit of 0 / 200.

更新变量

- 功能介绍：批量更新创建的变量的值类型和值。
- 变量类型：
 - 常量
 - 变量
- 属性面板：
 - 1: 选择创建变量的输出
 - 2: 选择更新值类型
 - 3: 设置更新值



表达式

- 功能介绍：引用流程中变量，通过时间、文本、数字、自定义等函数公式组合运算，输出指定类型变量，供后续节点使用。

基础属性

序号	属性	说明	-
1	节点名称	流程节点的名称标记。	
2	输出变量名称	设置表达式后，可生成一个流程变量，该变量的唯一标记在此处定义，可供流程下游节点使用。	
3	变量类型	选择输出变量类型，可选择字符串、布尔、对象、对象列表、实数、整数等。	-
4	选择对象	如“变量类型”选择，对象类型或对象列表时，需选择具体对象，其他变量类型无需选择。	-



表达式设置

- 单击设置表达式，进入表达式编辑器页面。
- 在左侧函数列表中，选择函数公式，支持日期、文本、数字、逻辑判断、自定义函数等函数公式。具体可参见 [表达式](#)。右侧参数列表选择流程中变量及变量某一字段。
- 表达式设置完成后，点击刷新，语法校验正确后，单击确定。

API

最近更新时间：2023-03-10 09:38:16

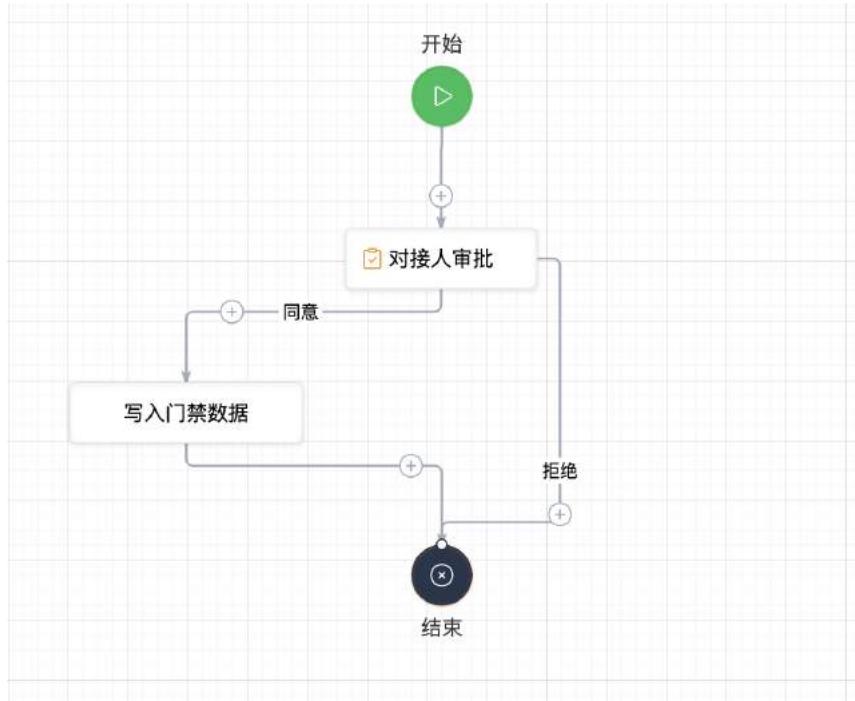
自定义 API 节点

功能简介

按照流程编排的规则运行 API，实现自定义业务逻辑编排。

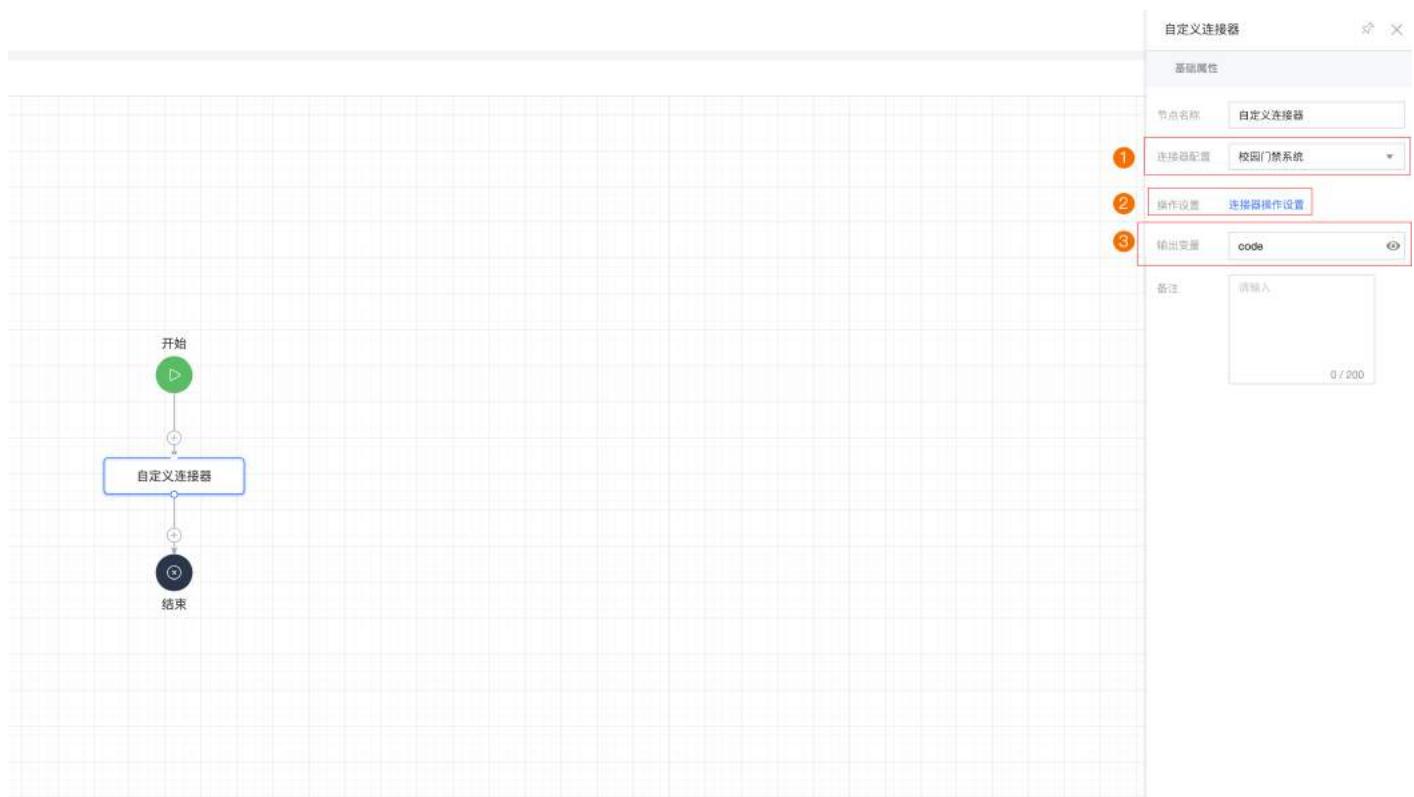
场景示例

假设我们有一个入校审批的场景，当对接人通过审批后，我们向校园的门禁系统写入访客数据：



1. 配置自定义 API 信息：

- 1: 选择连接器
- 2: 选择连接器方法并设置入参
- 3: 出参映射的变量名



2. 选择自定义 API 方法:



3. 设置自定义 API 方法入参:



业务场景实践

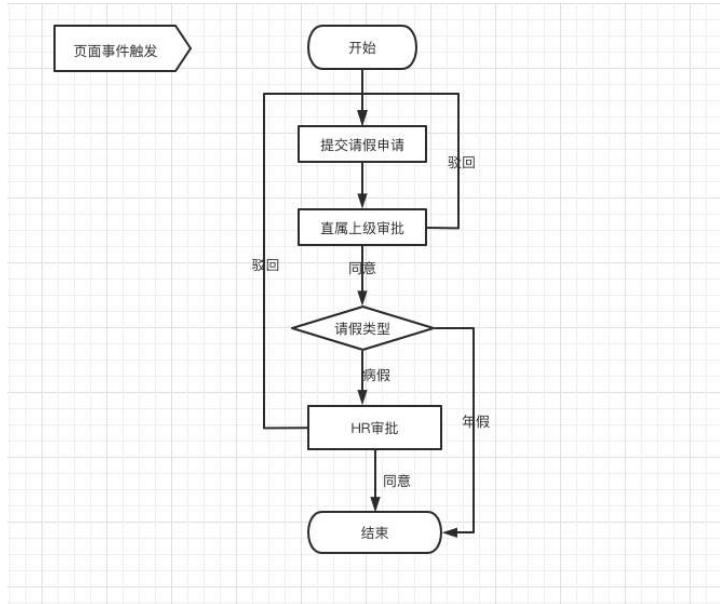
请假审批流程配置示例

最近更新时间：2023-07-06 16:05:09

请假审批

场景说明

员工通过 OA 系统首页，单击请假申请，提交请假申请，直属上级审批通过后，如果请假类型是年假，直属上级审批通过流程结束，如果请假类型是病假，需要 HR 审批，审核病假条是否合规。

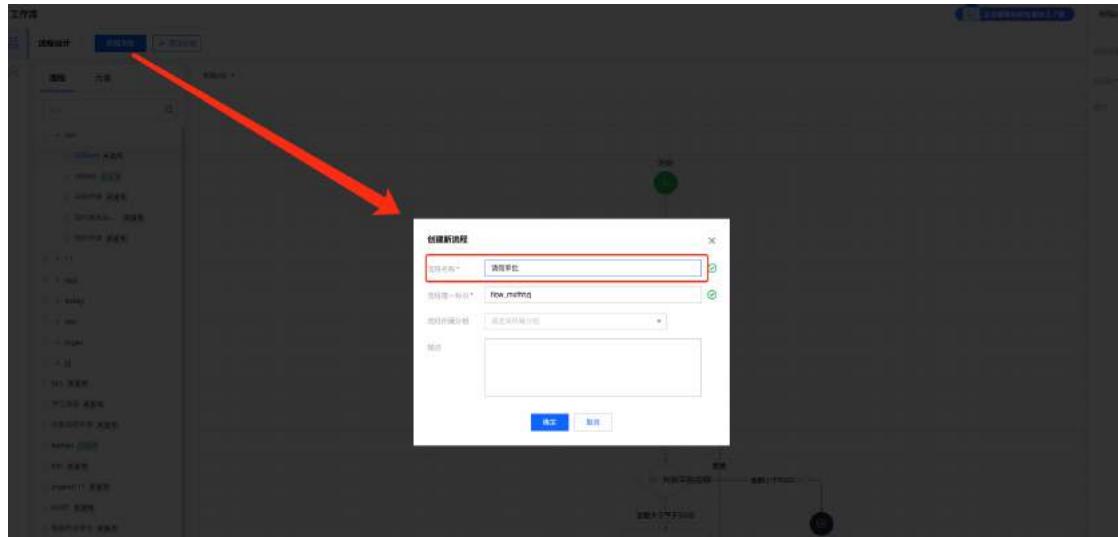


前置准备

- 新建好请假表单的数据模型，请参见 [创建数据模型](#) 相关文档。
- 创建 HR 角色，并分配给相关用户，请参见 [用户权限](#) 相关文档。
- 创建好请假审批所在的模型应用，请参见 [应用创建](#) 相关文档。

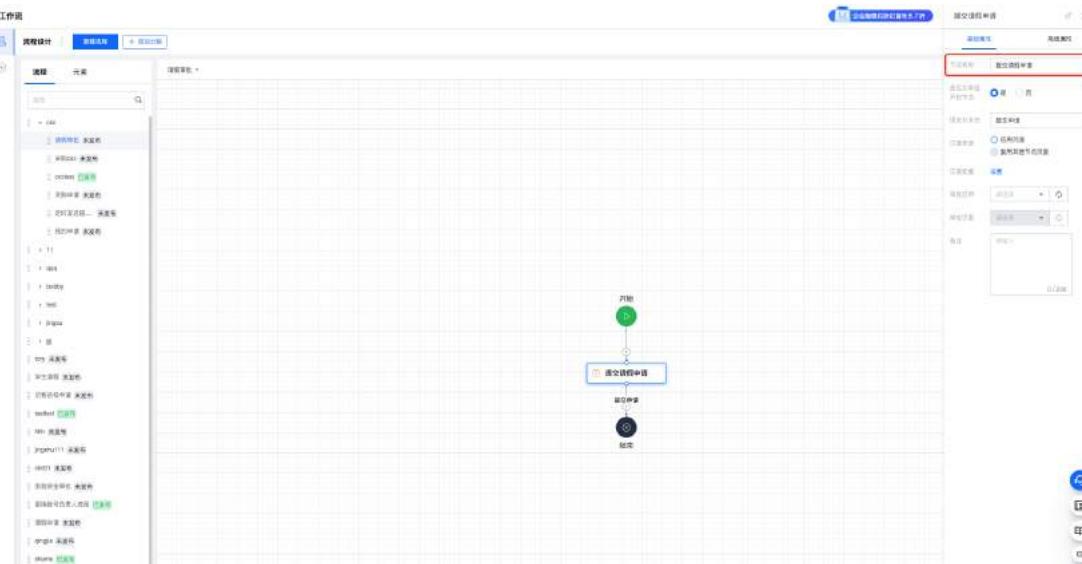
操作步骤

1. 新建名称为请假审批的流程。

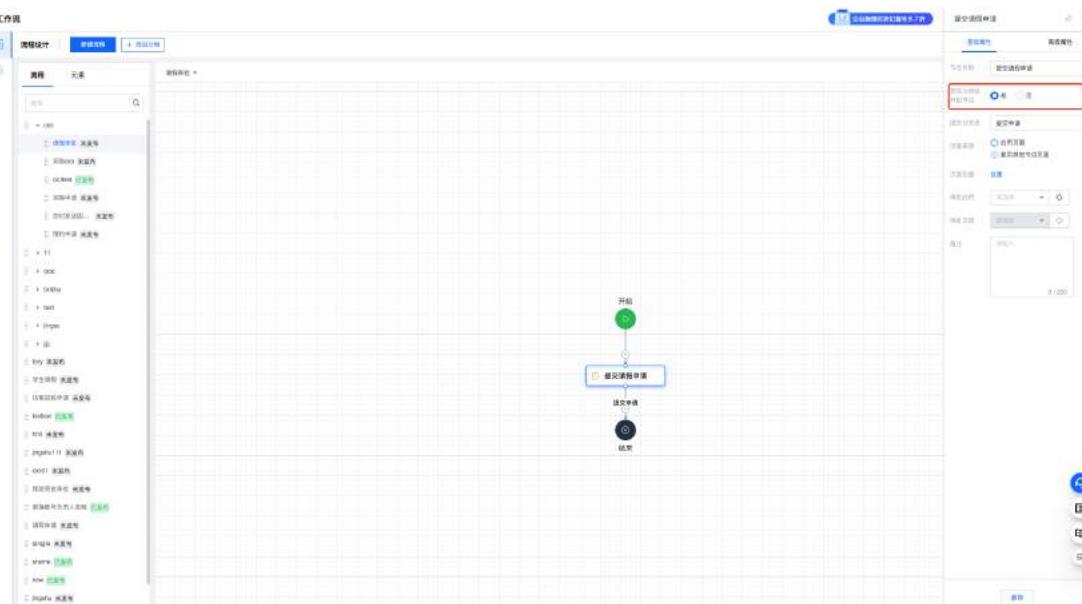


2. 添加审批节点，命名为提交请假申请。

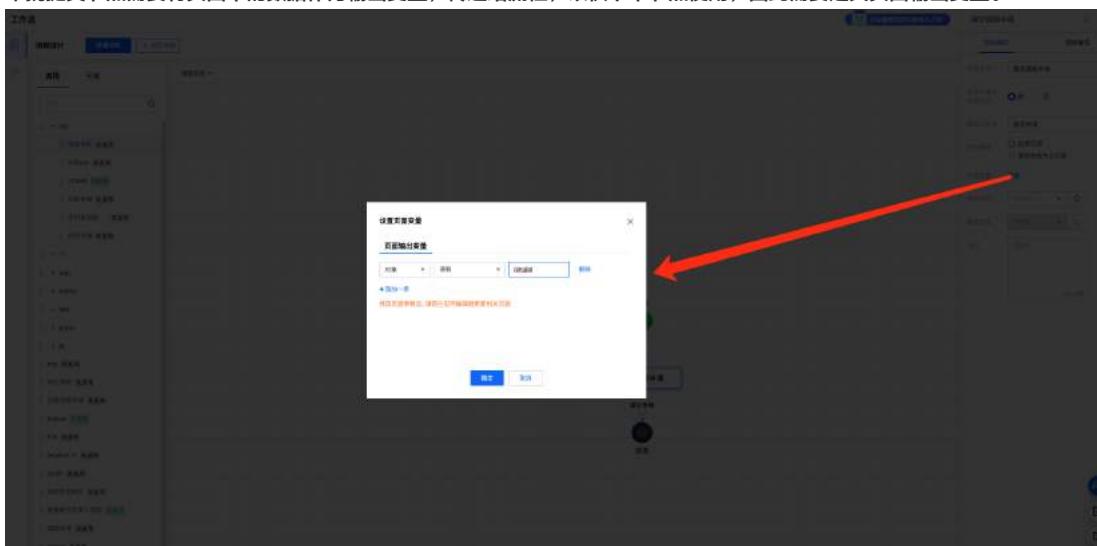
i. 节点名称命名为：提交请假申请。



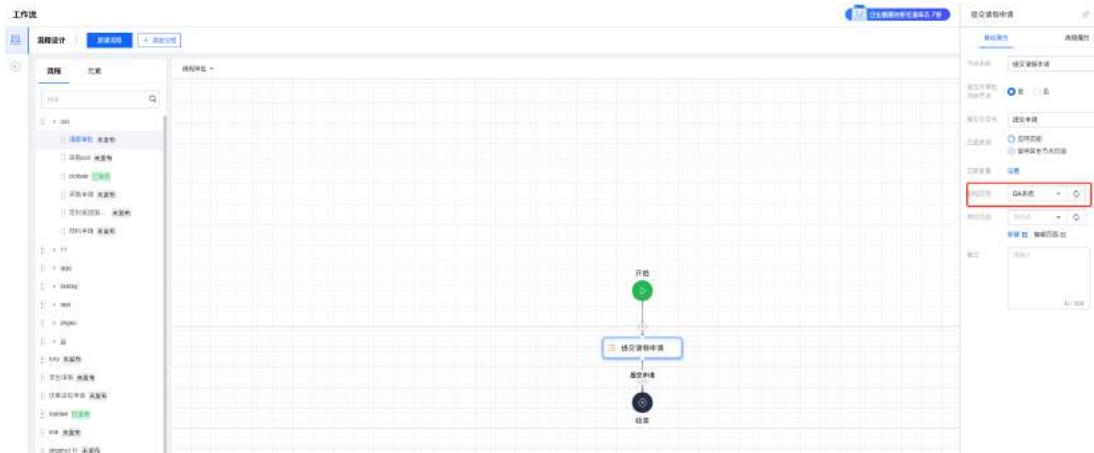
ii. 由于第一个节点是提交请假操作，因此该节点选择是审批开始节点。



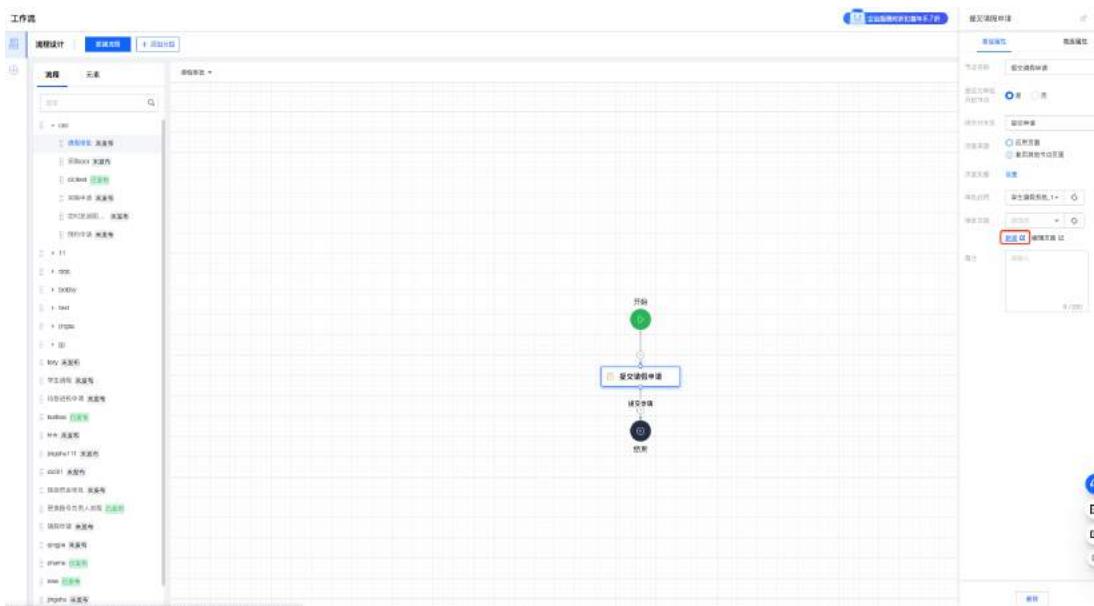
iii. 审批提交节点需要将页面中的数据作为输出变量，传递给流程，以供下个节点使用，因此需要定义页面输出变量。



iv. 选择该审批对应的应用。

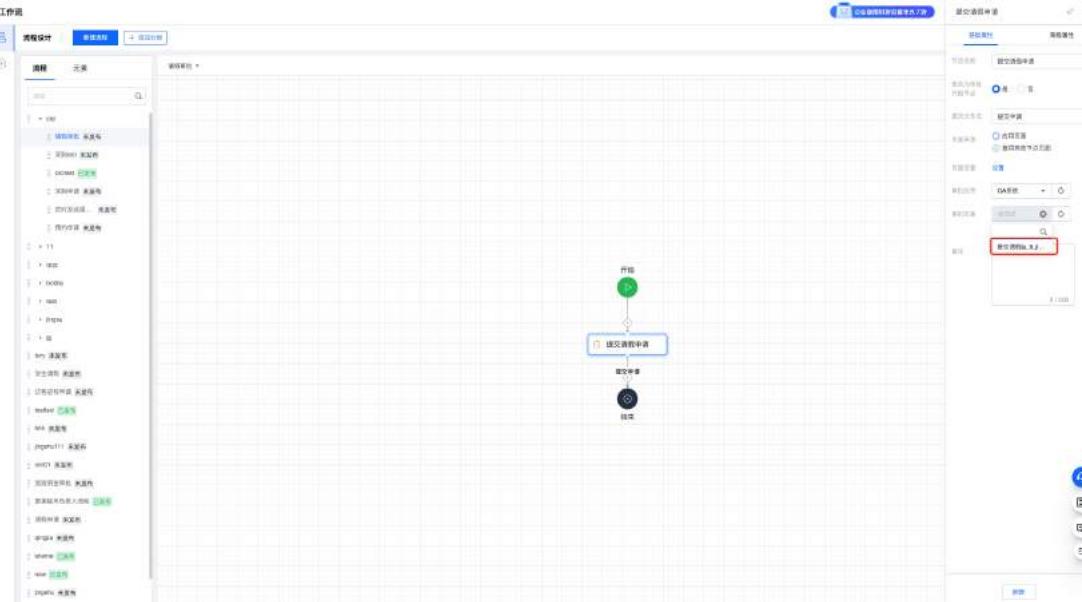


v. 创建提交请假的审批页面，系统会根据页面输出变量的数据对象结构自动帮您生成提交审批的表单页面，您可以根据业务需求在编辑器中进行二次调整。



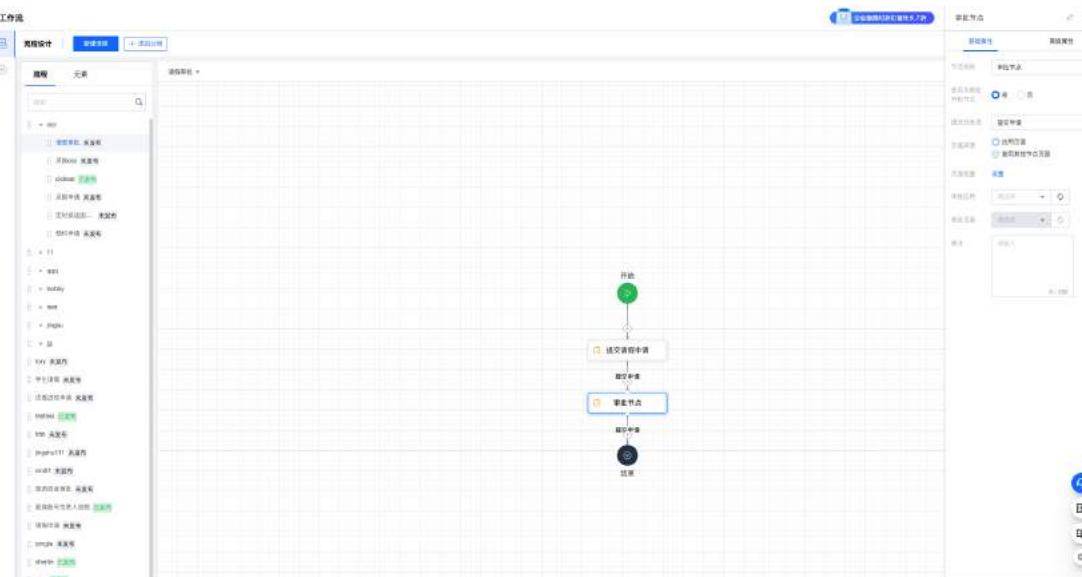
vi. 修改完成审批页面后，保存审批页面，回到流程节点。

审批节点页面选择该页面。

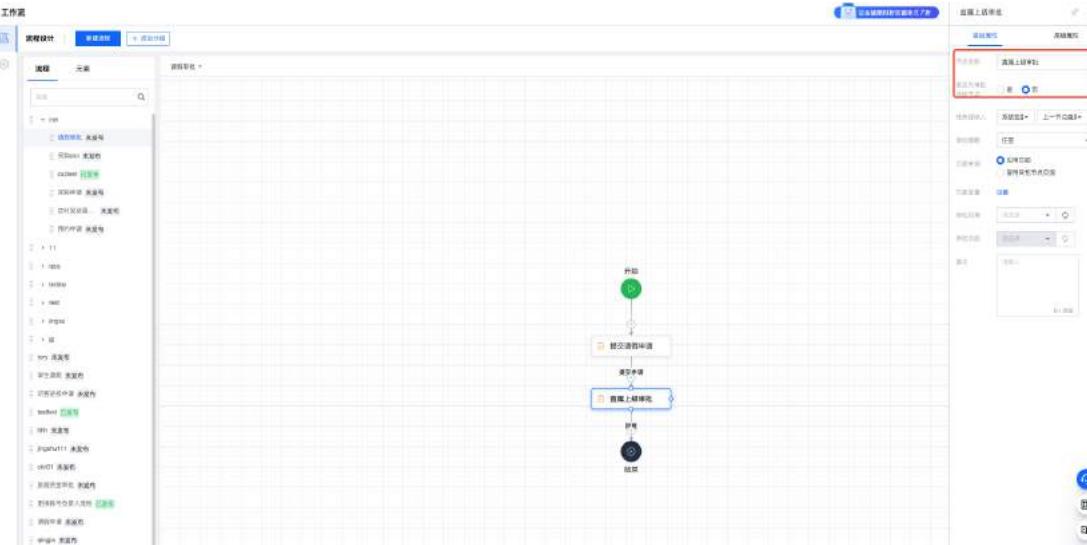


3. 继续添加审批节点，命名为直属上级审批。

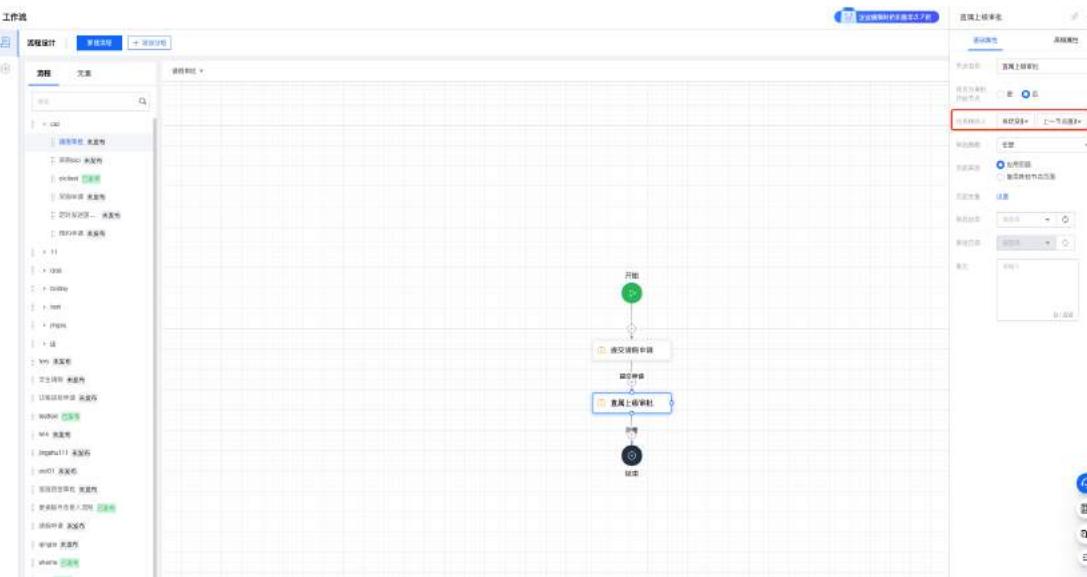
i. 添加审批节点。



ii. 审批节点命名为**直属上级审批**，由于该节点是直属上级审批节点，并非开始提交节点，因此该节点选择否审批开始节点。

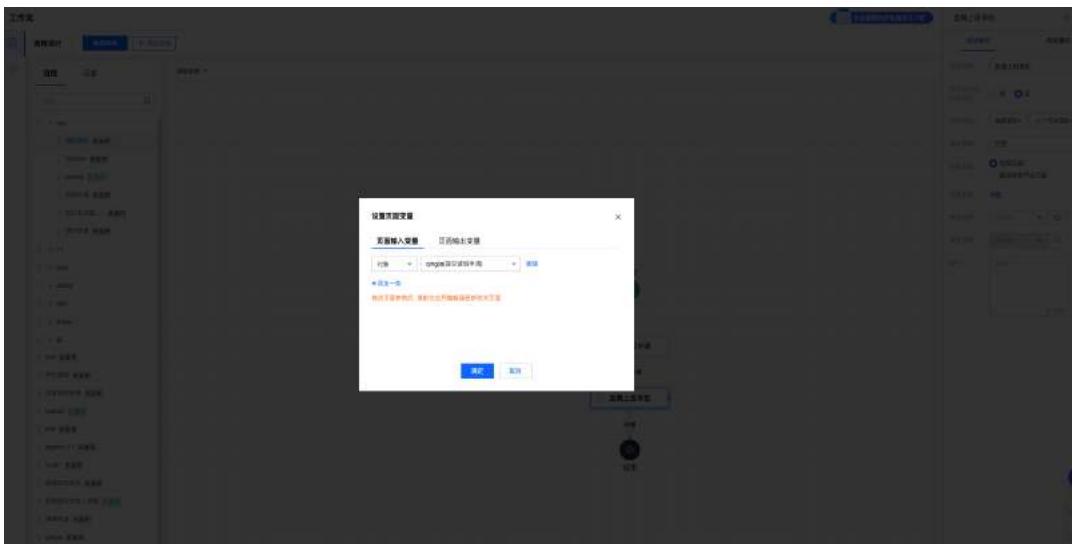


iii. 该节点需要提交人的直属上级审批，因此任务接收人选择为系统变量 > 上个节点直属上级。

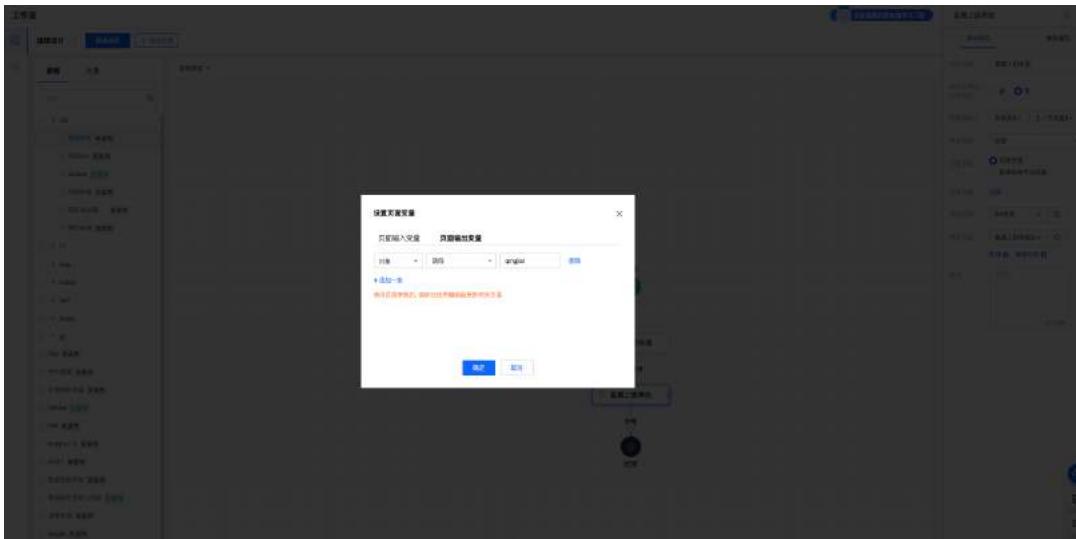


iv. 配置审批节点的变量，上个提交请假的数据存入了上个节点的输出变量中，需要在该页面中展示出来，因此该页面输入变量选择上个节点定义的输出变量；如果该节点直属上级需要编辑数据，定义该节点输出变量。

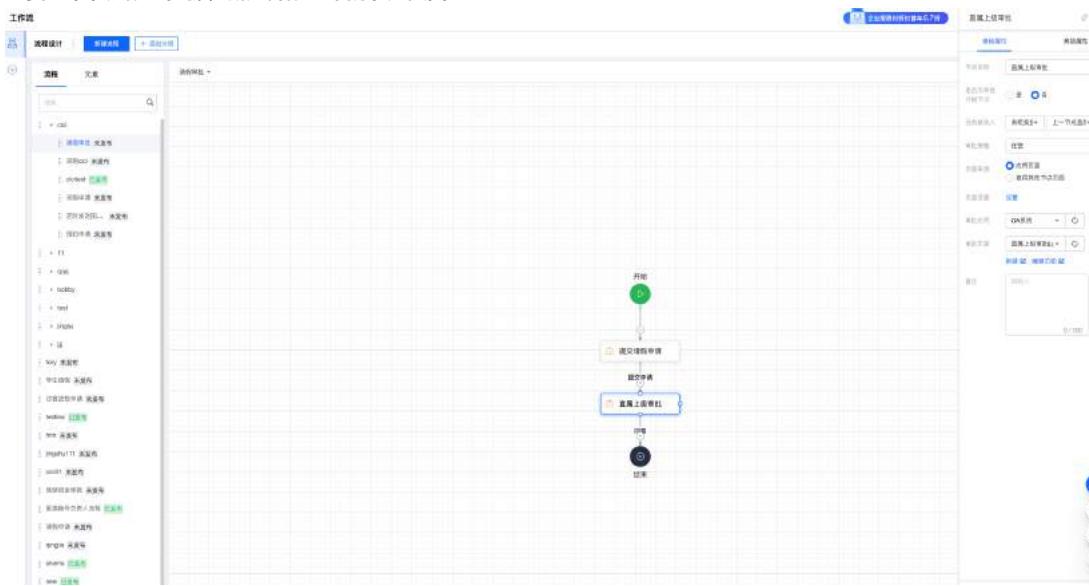
- **输入变量：**



▪ 输出变量：

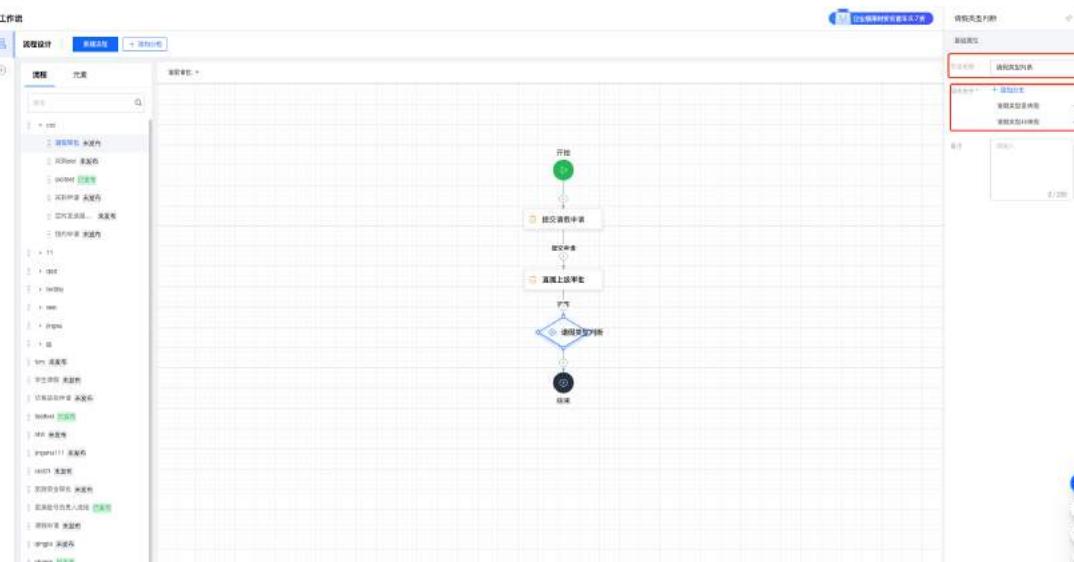


v. 重复上个节点配置步骤，新建并配置审批节点页面。

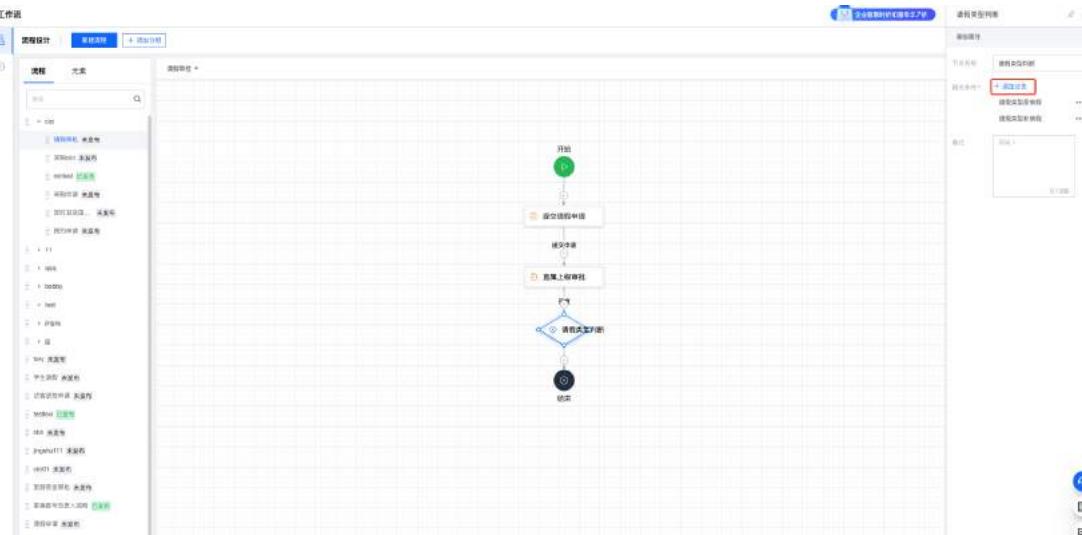


4. 添加排他网关，命名为请假类型。

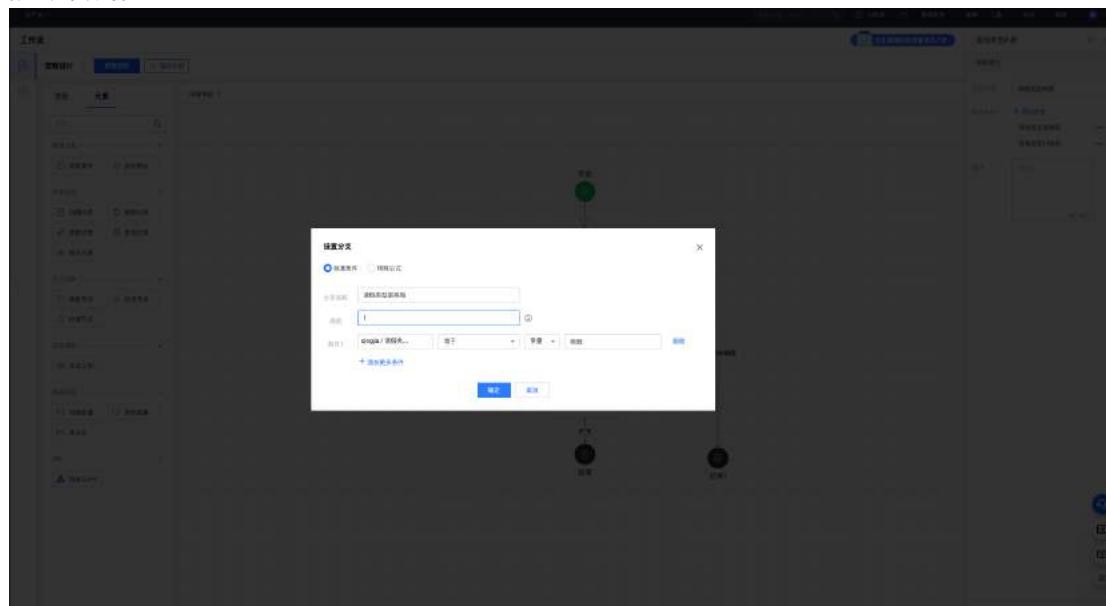
i. 添加排他网关，并命名为请假类型判断。



ii. 添加分支，并设定该分支条件，例如分支1为请假类型是病假；同样的方式设定请假类型为非病假的分支。



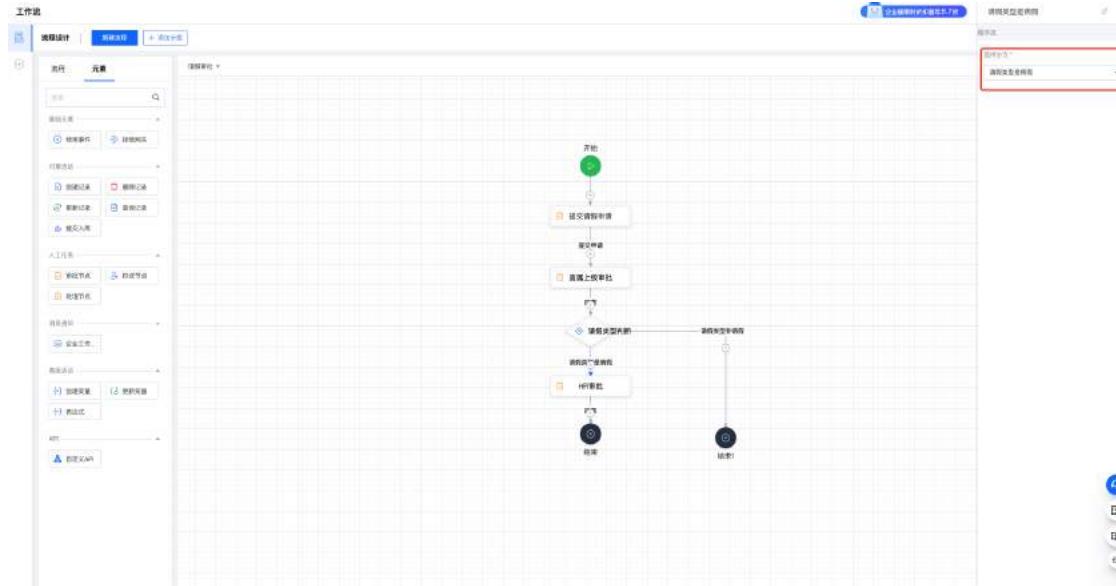
配置分支条件。



5. 继续添加审批节点，命名为 HR 审批。

添加审批节点，命名为**HR 审批**，重复上个审批节点配置步骤，审批任务接收人选择**角色 > HR**，并选中排他网关和审批节点之间的连线，选择分支为**请假类**

型是病假的分支；如果请假类型是非病假不需要 HR 审批，流程走另一个分支，直属上级审批结束后，流程结束。



6. 检查配置无误后，保存或发布流程。

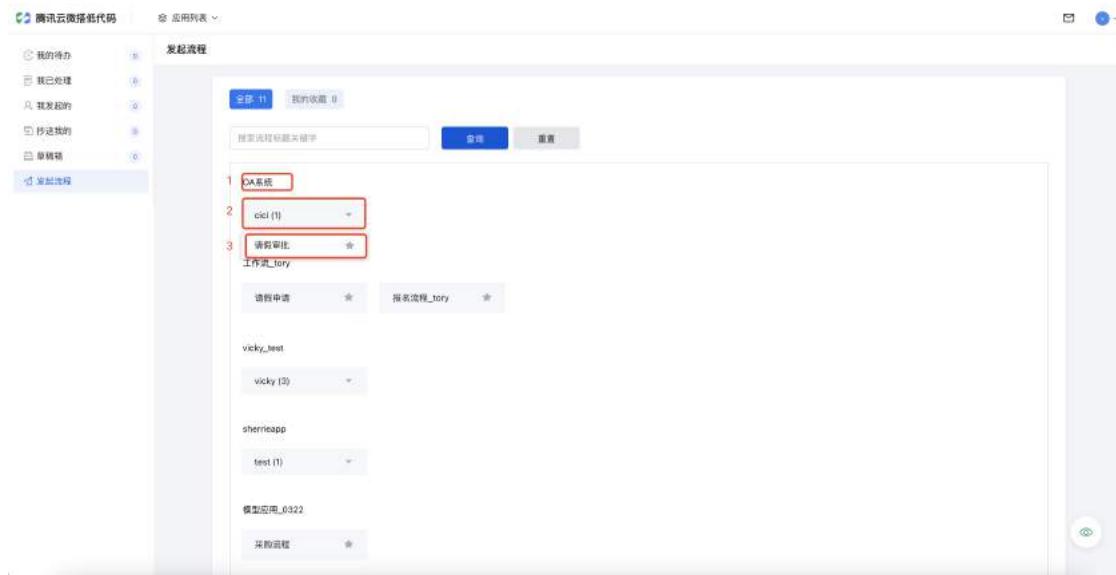
7. 登录企业工作台，在运行态提交审批流程。

② 说明：

保存后的流程可在体验环境下测试，发布后的流程可在正式环境触发。

登录企业工作台，进入流程中心，在流程中心发起流程菜单下，选择已创建的流程并发起。

- 1: 应用名称
- 2: 流程分组
- 3: 流程名称



⚠ 注意：

这里流程列表只展示满足以下条件的流程，并非全部流程：

- 1: 流程开始节点未配置输入变量
- 2: 流程第一个审批节点为审批开始节点
- 3: 当前登录用户有该审批流程的权限

审批流

最近更新时间：2023-07-06 16:05:15

本文以证书申请流程为例进行审批流的业务场景实践说明。

② 说明：

以下是审批正向流程的操作示例，更多业务场景可以通过灵活配置去实现，譬如流程中的转办、回退、会签按钮，设置不需要原件、审批时点击拒绝等等。

操作步骤

步骤1：建立数据模型（证书申请）

进入 [数据模型](#) 页面，单击新建数据模型，填写数据相关信息后单击创建完成数据模型的新建，并在新数据模型添加以下字段：[借用开始时间](#)、[借用结束时间](#)、[证书借用用途](#)、[是否需要原件](#)和[借用备注说明](#)。

模型字段 ①

字段名称	字段标识	数据类型	是否必填	是否唯一 ①	操作
借用开始时间	jyksqj	日期时间	否	否	<input checked="" type="checkbox"/> <input type="radio"/>
借用结束时间	jyjsqj	日期时间	否	否	<input checked="" type="checkbox"/> <input type="radio"/>
证书借用用途	zsjyyt	枚举	否	否	<input checked="" type="checkbox"/> <input type="radio"/>
是否借用原件	sifyyj	布尔值	否	否	<input checked="" type="checkbox"/> <input type="radio"/>
借用备注说明	jybzsm	文本 单行文本	否	否	<input checked="" type="checkbox"/> <input type="radio"/>

[展开系统字段](#) [+ 添加字段](#)

[保存](#) [取消](#)

⚠ 注意：

其中证书借用用途的字段设置 > 关联选项集，须单击立即创建。

字段设置 ①

字段名称 * 证书借用用途

字段标识 * zsiyyt

字段描述 请输入字段描述

数据类型 * 枚举

选择设置 * 单选 多选

关联选项集 * 证书借用用途

没有合适的选项集? [立即创建](#)

默认值 请选择

是否必填 是 否

是否唯一 是 否

该属性在字段新增后不可修改

[确定](#) [取消](#)

如下图创建新的证书借用用途选项集并单击保存。实现与新建选项集进行关联。

修改通用选项集 ①

选项集名称 * 证书借用用途

选项集标识 * zsiyyt

备注 请输入描述文本

选择集 * 选项标识 选项值

1	合同签署
2	材料证明
3	实践申请
4	其他事项

[保存](#) [取消](#)

步骤2：新建角色和部门架构并新增用户

1. 进入 用户权限 > 角色与权限 页面，新建处长、主任、校长、院长和老师五个角色。

名称	标识	描述	发布状态	操作
处长	cz	-	已发布	添加用户 更多
主任	zr	-	已发布	添加用户 更多
校长	xz	-	已发布	添加用户 更多
院长	xyfr	-	已发布	添加用户 更多
老师	lfd	-	已发布	添加用户 更多
默认访客①	anonymousUser	默认访客角色	已发布	添加用户 更多
默认超级管理员①	administrator	默认内核用户角色	已发布	添加用户 更多
默认外部用户①	externalUser	默认外域用户角色	已发布	添加用户 更多

2. 进入 数据模型 页面，找到数据模型中的用户（系统表）并新增两个字段：人员岗位名称和职能部门审核。

字段名称	字段标识	数据类型	是否必填	是否唯一	操作
拓展字段2 [主列字段]	tzsf2	文本 单行文本	否	否	
职能部门审核	znmbsfh	布尔值	否	否	
人员岗位名称	rygwnmc	文本 单行文本	否	否	
筛选字段	sxzd	文本 单行文本	否	否	
排序字段	spxx	数字	否	否	
拓展字段	tzsf	文本 单行文本	否	否	

3. 进入 用户权限 > 用户 页面，单击新建部门，新建XXXX大学为一级部门，校综合办公室、职能部门、教学单位为二级部门，职能部门下挂三级部门（教务处），教学单位下挂三级部门（材料工程学院）。在材料工程学院中单击新建用户新增两个用户 zhangyuanzhang (关联院长角色) 和 lilaoshi (关联老师角色)。

操作	用户名	角色	手机号	邮箱	账号	所属部门	部门负责人	更新时间	操作
<input type="checkbox"/>	zhangyu... 新建	院长	-	-	已授权	材料工程...	是	2022-07-...	
<input type="checkbox"/>	lilaoshi 新建	老师	-	-	已授权	材料工程...	否	2022-07-...	

② 说明：

在新增 zhangyuanzhang 和 lilaoshi 的时候，职能部门审核和人员岗位名称保持初始值不变。

说明：在新增 zhangyuanzhang 和 lilaoshi 的时候，职能部门审核和人员岗位名称保持初始值不变。

职能部门审核

人员岗位名称

4. 通过单击设置部门负责人。

The screenshot shows the 'Material Engineering College' department management page. On the left, there's a sidebar with department categories like 'All Departments', 'Unlinked Departments', and 'XXXX University'. Under 'XXXX University', 'Teaching Affairs Office' is expanded, and 'Teaching Affairs' is selected. A modal window titled 'Set Department Manager' is open, showing a list of users: 'zhangyuanzhang' (selected) and 'liaoshi'. The user 'zhangyuanzhang' has the role 'Dean' and is marked as 'Has Manager'. The 'Set Manager' button at the bottom of the modal is highlighted with a red box.

将 zhangyuanzhang 勾选设置为材料工程学院部门负责人。

This is a close-up view of the 'Set Department Manager' dialog. It shows a search bar and two user entries: 'zhangyuanzhang' (with a checked checkbox) and 'liaoshi' (unchecked). To the right, a list titled 'Selected 1 item' shows 'zhangyuanzhang'. At the bottom are 'Confirm' and 'Cancel' buttons.

5. 教务处新增用户 zhaochuzhang (关联处长角色)。

The screenshot shows the 'Teaching Affairs' department management page. The sidebar shows 'Teaching Affairs' is selected under 'Teaching Affairs Office'. A user named 'zhaochuzhang' is listed in the main table with the role 'Deputy Director' and status 'Has Manager'. The 'Import User' button is visible at the top of the table area.

职能部门审核单击开启。

修改用户

1 填写用户信息 > 2 产品许可证 > 3 设置角色权限 (选填)

用户名 *	zhaochuzhang
密码	不输入表示不修改密码
邮箱	请输入邮箱
手机号	请输入手机号
关联部门	教务处
描述	请输入用户的描述。非必填项，最多支持255个字符
职能部门审核	<input checked="" type="checkbox"/>
人员岗位名称	

取消 下一步

6. 校综合办公室新增用户 kongzhuren (关联主任角色) 。

用户

内部用户/部门管理 外部用户

校综合办公室

用户名	导入用户	关联角色	更多操作
kongzhuren	-	主任	已授权

操作

人员岗位名称设置为主任。

修改用户

1 填写用户信息 > 2 产品许可证 > 3 设置角色权限 (选填)

用户名 *	kongzhuren
密码	不输入表示不修改密码
邮箱	请输入邮箱
手机号	请输入手机号
关联部门	校综合办公室
描述	请输入用户的描述。非必填项，最多支持255个字符
职能部门审核	<input checked="" type="checkbox"/>
人员岗位名称	主任

取消 下一步

7. 系统自带的管理员 administrator 角色关联校长。

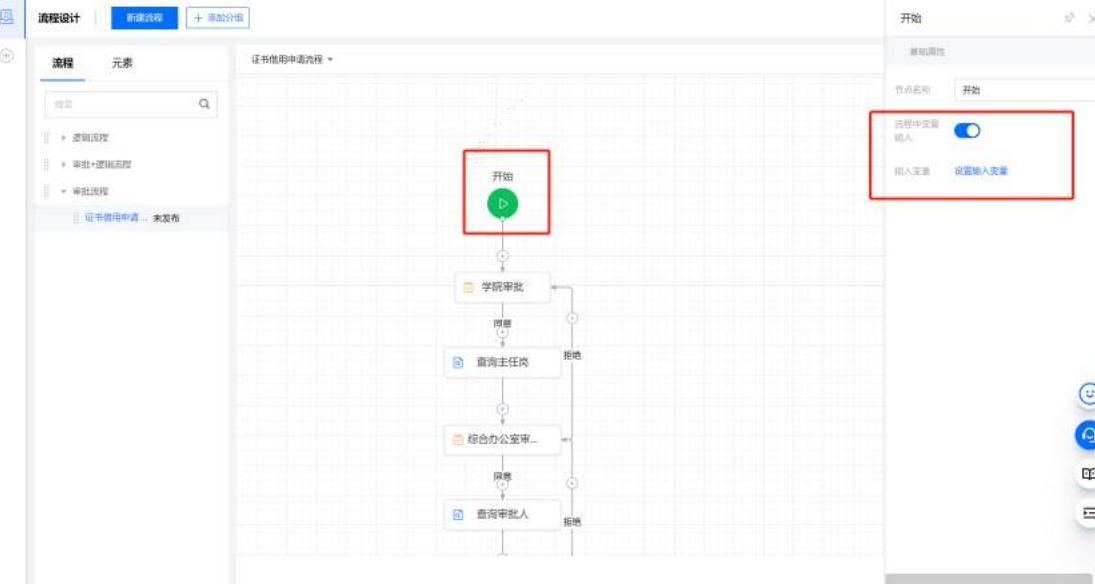
The screenshot shows the internal user management interface. On the left, there's a sidebar for 'Internal User/Department Management' with sections like 'All Internal Users', 'Unlinked Department Users', 'Organization Structure Test', 'Role Association Test', 'XXXX University', 'Administrative Office', 'Teaching Department', 'Student Affairs Office', and 'College Units'. The 'Administrator' user is selected in the list on the right. A red box highlights the 'President' role under the 'Associated Roles' column for the 'Administrator' user.

职能部门审核单击开启。

The screenshot shows the 'Modify User' form. It has three steps: 1. Fill User Information, 2. Product License, and 3. Set Role Permissions (Optional). Step 1 is active. The user information includes: Username: administrator, Password: (empty), Email: (empty), Phone: (empty), Department: (empty), Description: (empty). Below these fields is a switch labeled '职能部门审核' (Departmental Review) which is currently off. Step 2 and Step 3 are shown as optional steps.

步骤3：新建流程（证书借用申请流程）

1. 进入 **流程设计** 页面，选中流程开始节点，单击设置输入变量。



选择对象 > 证书申请，输入变量设置为 ZSSQ。



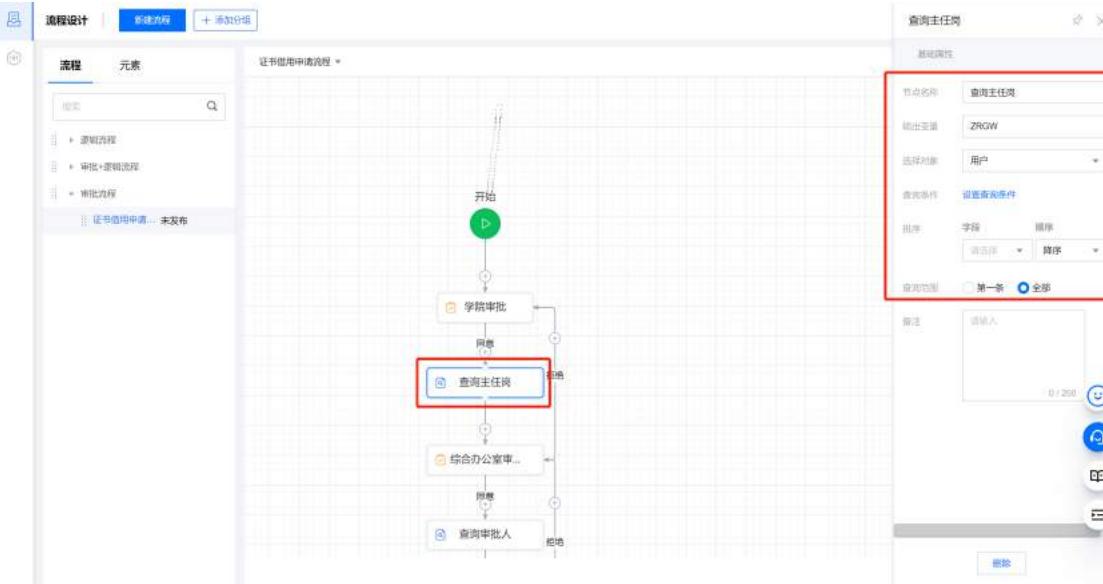
2. 添加审批节点元素，任务接收人设置为系统变量 > 上个节点部门负责人。



② 说明：

此功能实现本部门的人提交的申请给到对应的部门负责人去审批。

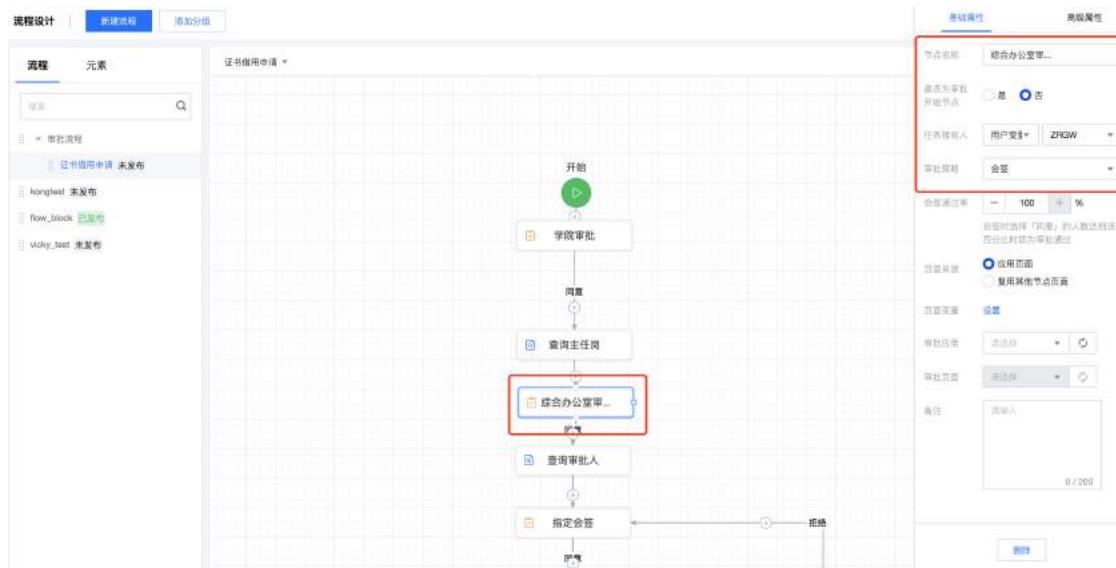
3. 添加查询记录元素，配置内容参考下图，输出变量设置为 ZRGW。



单击设置查询条件（查询系统用户表中所有人员岗位名称等于主任的记录，给到变量 ZRGW）。



4. 继续添加审批节点元素，任务接收人设置为用户变量 > ZRGW。



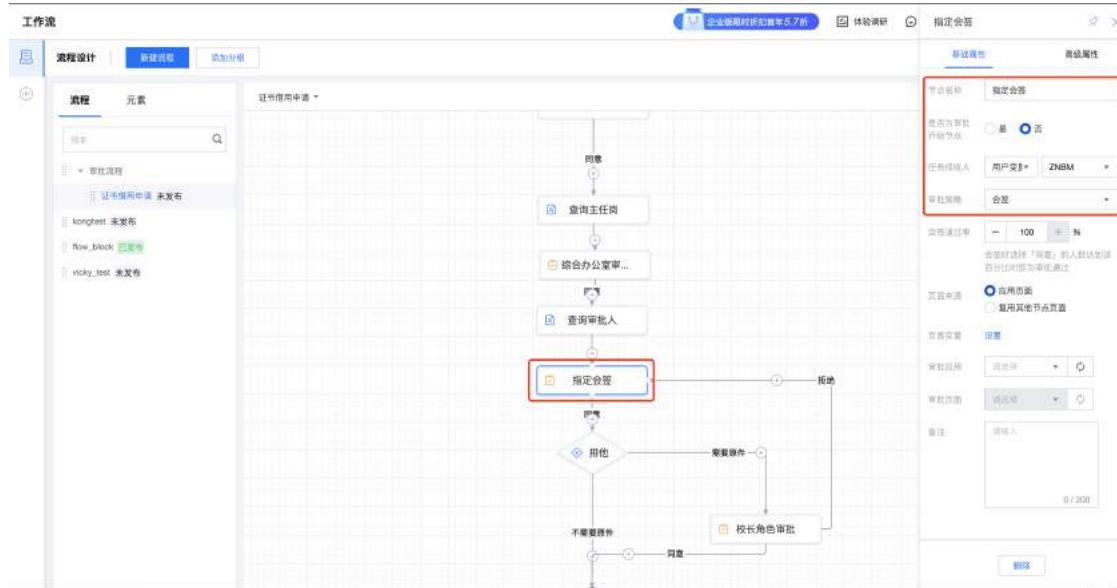
5. 添加查询记录元素，配置内容参考下图，输出变量设置为 ZNBM。



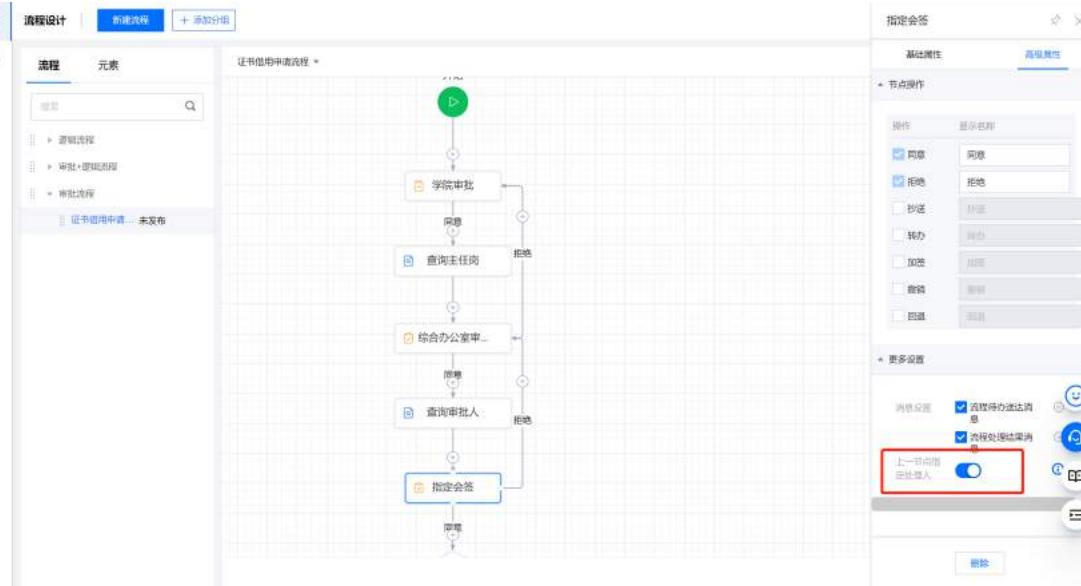
单击设置查询条件（查询系统用户表中所有职能部门审核字段等于 true 的记录，给到变量 ZNBM）。



6. 添加审批节点元素，任务接收人设置为用户变量 > ZNBM。



7. 高级属性中上一节点指定处理人单击开启（通过此功能来设置本节点（指定会签）处理人由上一节点（综合办公室审批）来指定）。



⚠ 注意：

综合办公室指定审批人时会从指定会签节点配置的任务接收人中做选择。

8. 添加一个排他网关元素，配置需要原件和不需原件两个分支。



- 需要原件 (条件为 ZSSQ 变量中的是否需要原件字段等于 true) :

This screenshot shows the 'Set Branch' dialog for the '需要原件' condition of the XOR gateway. It is a modal window with the following fields:

- 标准条件:** Standard condition is selected.
- 分支名称:** Branch name is '需要原件'.
- 规则:** Rule is '1'.
- 条件1:** Condition 1 is 'ZSSQ / 是否需要原件 等于 true'.

 At the bottom are '确定' (Confirm) and '取消' (Cancel) buttons.

- 不需原件（条件为 ZSSQ 变量中的是否需要原件字段等于 false）：



9. 添加审批组件元素，需要原件分支链到审批组件（需要原件时要校长进行最后的复核和审批），任务接收人选择角色-校长。



△ 注意：

- 以上都是同意的正流程配置，拒绝时我们可以从审批节点拉取链接线到对应的节点，标识审批拒绝之后任务回到指定节点
- 通过高级属性我们可以灵活配置审批节点的操作按钮（同意和拒绝为默认按钮，其他按钮按需选择，您可以选择后体验更多功能）；消息设置勾选后流程参与者都会收到对应的消息通知。



0. 流程建立完成之后，单击保存即可保存流程。

步骤4：配置数据模型触发流程

1. 进入 [步骤1 中新建的数据源 证书申请 > 事件处理](#) 页面。

2. 单击 [新建事件处理](#)。

进行如下内容配置：

触发时机 *	新建后
执行前置条件	...
流程 *	证书借用申请流程
设置入参	ZSSQ 新增后值

步骤5：模型应用关联数据模型

1. 进入 [创建新应用](#) 页面，单击从空白新建 > 新建数据模型应用。勾选 [步骤1 新建的证书申请数据模型](#)。

2. 进入 [应用](#) 页面，找到 [步骤5.1 中新建的数据模型](#)，单击 [编辑应用](#) 进入应用编辑器。

3. 可通过单击右上角发布进行应用发布。

The screenshot shows the 'Certificates Application' page in the Low-Code Platform. A modal window titled 'Publish the current application to Web' is open, with a red box highlighting the 'Publish' button. The main area displays a table of certificate applications.

步骤6：配置角色权限

下文通过新建角色处长为例进行操作说明。

1. 返回 [步骤2 中 用户权限 > 角色与权限](#) 页面，对处长、主任、校长、院长和老师五个角色统一进行角色配置。

The screenshot shows the 'Role and Permissions' page. A red box highlights the 'Administrator' row in the table, which has the identifier 'cz'. Other rows include '主任' (zr), '校长' (xz), '院长' (xytz), and '老师' (tdy). Each row shows a status of 'Published' and a 'More' button for adding users.

2. 单击处长进入角色配置页。在角色权限 > 模型应用访问权限中，找到证书申请应用，并勾选是否可以访问。

The screenshot shows the 'Role Configuration' page for the 'Administrator' role. In the 'Role Permissions' section, the 'Model Application Access Permissions' tab is selected. A red box highlights the 'Certificates Application' row, which has the identifier '8yw341rm'. The 'Can Access' checkbox is checked, and a red box highlights the 'Page Permission Settings' button.

单击页面权限设置，对4个页面是否可以访问进行勾选后单击确定。

设定应用“证书申请应用”下的页面权限

X

页面名称	是否可以访问
zssq_td4mulm_list 证书申请列表	<input checked="" type="checkbox"/>
zssq_td4mulm_create 创建证书申请	<input checked="" type="checkbox"/>
zssq_td4mulm_update 更新证书申请	<input checked="" type="checkbox"/>
zssq_td4mulm_detail 证书申请详情	<input checked="" type="checkbox"/>

3. 进入角色权限 > 数据模型权限页面，找到证书申请单击行权限设置和列权限设置进行相关配置。

测试新建	csxj_we4h842	模型应用_0714	2023-04-23 11:08:18	行权限设置	列权限设置 ①
字段测试专用	cszdzy_1tf4t2f	微搭示例官网_0705-管理平台 测试表格组件	2023-04-23 11:08:18	行权限设置	列权限设置 ①
客户-示例	customer	新版编辑器	2023-04-23 11:08:18	行权限设置	列权限设置 ①
DGH_审批流	dgh_spl_7aneqo0	DGH—_审批流_0423	2023-04-23 11:08:18	行权限设置	列权限设置
疫情防控团购商品	GBuyCommodity_hg68ber	事件面板微信能力测试 【勿删】 xgroup[勿删除]	2023-04-23 11:08:18	行权限设置	列权限设置 ①
疫情防控团购订单	GBuyOrder_jgd1rl0	事件面板微信能力测试 【勿删】 新版编辑器	2023-04-23 11:08:18	行权限设置	列权限设置 ①
小商店商品轮播图数据-...	goodsimags		2023-04-23 11:08:18	行权限设置	列权限设置 ①
公司介绍	gsjs_xmdgvlb	自定义应用_0403 冲突优化	2023-04-23 11:08:18	行权限设置	列权限设置 ①

行权限配置：

行权限设置 X

数据模型名称: DGH_审批流
数据模型标识: dgh_spl_7aneqo0

行查看权限 ①

无权限 查看本人 查看本人及下属 查看本部门及下级部门 查看全部 自定义

行修改权限 ①

无权限 修改本人 修改本人及下属 修改本部门及下级部门 修改全部 自定义

◦ 列权限设置：

字段名称	字段标识	数据类型	权限设置
借用开始时间	jyksj	日期时间	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
借用结束时间	jyjsj	日期时间	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
证书借用用途	ztjyf	枚举	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
是否需要条件	sfyj	布尔值	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
借用备注说明	jybzm	文本 单行文本	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
数据标识	_id	文本 单行文本	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
创建时间	createdAt	日期时间	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
更新时间	updatedAt	日期时间	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
所有人	owner	关联关系	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
创建人	createdBy	关联关系	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
修改人	updateBy	关联关系	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改
所属部门	_departmentList	数组 ①	<input type="radio"/> 无权限 <input type="radio"/> 仅查看 <input checked="" type="radio"/> 查看并修改

4. 进入角色权限 > 流程权限页面，找到证书借用申请流程并勾选是否可以访问。

流程名称	流程标识	更新时间	是否可以访问
证书借用申请流程	flow_010	2022-07-12 18:11:50	<input checked="" type="checkbox"/>
记录查询	flow_013	2022-07-12 18:11:50	<input type="checkbox"/>
记录更新	flow_012	2022-07-12 18:11:50	<input type="checkbox"/>

5. 进入角色权限 > 企业工作台权限页面，对是否可登入企业工作台和企业工作台系统配置入口进行勾选授权。

是否可登入企业工作台	企业工作台系统配置入口
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

6. 进入角色权限 > APIs 权限页面，找到标识为 workflow 的 APIs 单击方法权限设置，勾选是否可以访问。

方法名称	方法标识	方法权限	是否可以使用
查询流程实例详情	DescribeProcessInstance	查询流程实例详情	<input checked="" type="checkbox"/>
查询审批页面详情	GetApprovalPageDetail	查询审批页面详情	<input checked="" type="checkbox"/>
执行流程申请	GetInstanceOperation	执行流程申请	<input checked="" type="checkbox"/>
查询流程通知详情	GetProcessNoticeInfo	查询流程通知详情	<input checked="" type="checkbox"/>
发起	AddCounterSignAssignee	发起	<input checked="" type="checkbox"/>
督办	PressTask	督办	<input checked="" type="checkbox"/>
修改可回退的节点列表	GetRefundableList	获取可回退的节点列表	<input checked="" type="checkbox"/>
回退	RollBack	回退	<input checked="" type="checkbox"/>
废止	Revoke	废止	<input checked="" type="checkbox"/>
提交流程单任务列表	SubmitFlowTaskList	提交流程单任务列表	<input checked="" type="checkbox"/>
操作单社任务	CompleteTask	操作单社任务	<input checked="" type="checkbox"/>
删除草稿	DeleteDraftProcessInstance	删除草稿	<input checked="" type="checkbox"/>
查询流程实例审批人列表	DescribeInstanceApproverList	查询流程实例审批人列表	<input checked="" type="checkbox"/>
查询流程实例发起人列表	DescribeInstanceOwnerList	查询流程实例发起人列表	<input checked="" type="checkbox"/>

步骤7：登录企业工作台验证审批流程功能

1. 进入 [应用](#) 页面，找到 [步骤5.1](#) 中新建的数据模型，单击编辑应用进入应用编辑器。

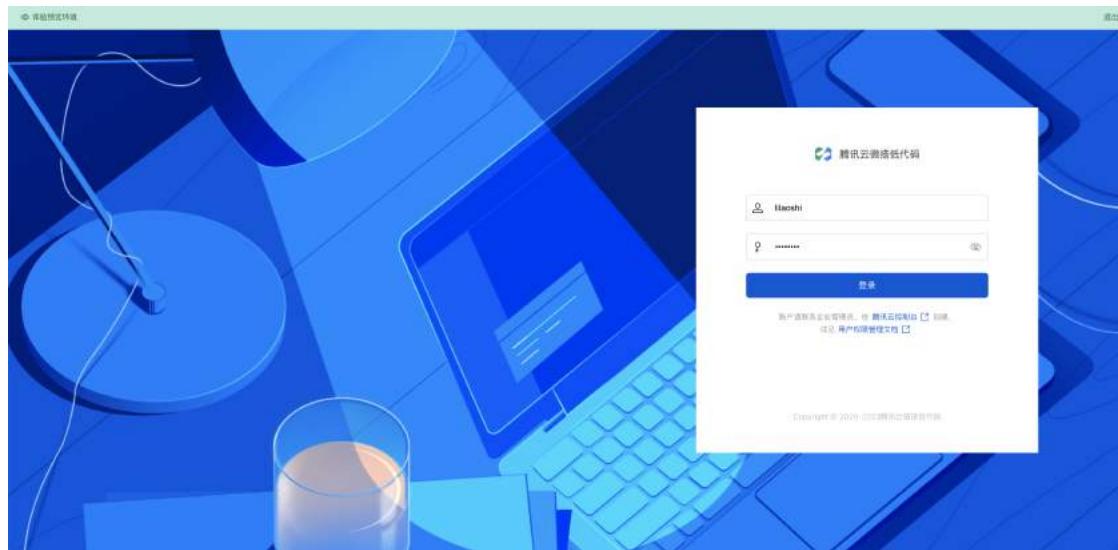
2. 单击右上角发布成功 > 访问企业工作台。

The screenshot shows the application editor interface for a 'Certificate Application' model. The left sidebar contains various data models like '普通字段', '图片', '程序', and '数据容器'. The main area displays a table titled '证书申请列表' with columns: '借用备注说明', '是否借用原件', '证书借用用途', '借用结束时间', '借用开始时间', and '操作'. There are three rows of data. On the right side, there are sections for '最新发布结果' (including a QR code), '屏幕配置', and '事件' (with options like '百度地图', '页面跳转', and '其他事件'). At the bottom, there's a pagination bar showing '共 2 条' and a '1 / 1 页' indicator.

3. 默认会登录 admin 的账号，先单击退出登录。

This screenshot shows the same application editor interface as above, but with a red box highlighting the '退出登录' (Logout) button in the top right corner of the user dropdown menu. The user listed is 'administrator'.

4. 重新以 lilaoshi 的账号进行登录。



5. 登录后进入证书申请列表页，单击新建。

The screenshot shows a table titled 'Certificate Application List'. It includes columns for 'Credit Note Description', 'Is Credit Note Attached', 'Certificate Use Purpose', 'Credit Note End Time', 'Credit Note Start Time', and 'Operations'. There are two rows of data: one for 'As certificate application material' and another for 'Auxiliary certificate material'. The 'Create' button at the top left of the table is highlighted with a red box.

填写相关申请内容后单击提交。

The screenshot shows a form titled 'Create Certificate Application'. It includes fields for 'Credit Note Description' (set to 'Auxiliary certificate material'), 'Is Credit Note Attached' (checkbox checked), 'Certificate Use Purpose' (set to 'Material Proof'), 'Credit Note End Time' (set to 2023-05-01 00:00:00), and 'Credit Note Start Time' (set to 2022-05-31 00:00:00). At the bottom are 'Submit' and 'Cancel' buttons, with 'Submit' being highlighted.

6. 提交后在流程中心可以查看已经发起的申请。

The screenshot shows the 'My Initiated' section of the process center. It lists two applications: 'liaoshi提交的证书借用申请' and 'liaoshi提交的证书借用申请'. Both entries show the applicant as 'liaoshi', the start time as '2023-05-08 16:44:37', and the status as '待办'. The 'Only my initiated' filter is highlighted with a red box.

可单击详情查看流程信息。流程信息通过流程详情和流程图示进行展示，其中流程详情包含流转记录。

流程详情：

基本信息

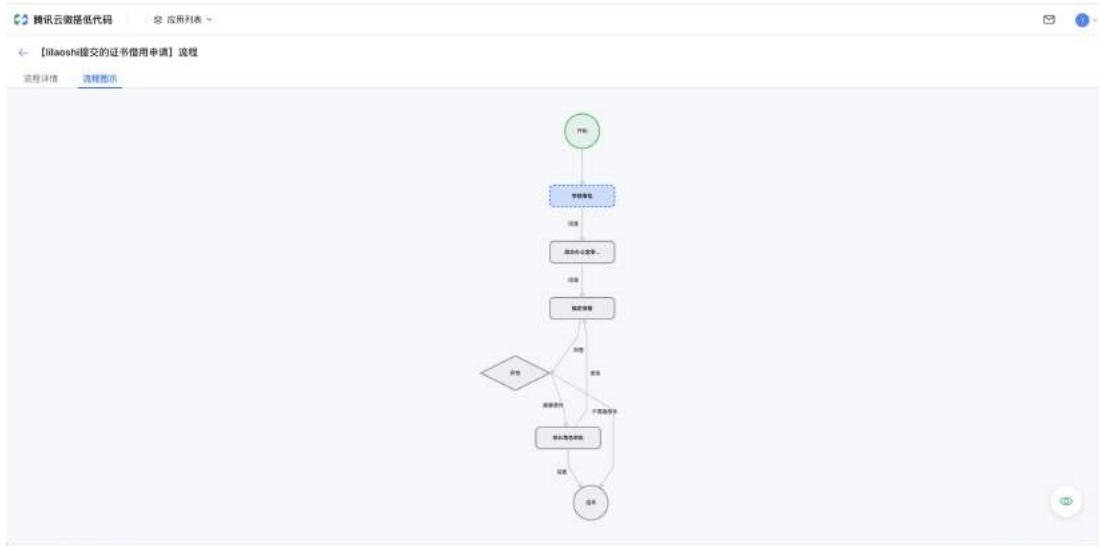
申请人: lilaoshi
发起时间: 2023-05-08 16:44:37
结束时间: --
流程状态: 待审批

申请信息

证书申请详情

借用备注说明: 增加证明材料
是否借用原件: 是
证书借用用途: 样料证明
借用结束时间: 2023-05-31 00:00:00
借用开始时间: 2022-05-31 00:00:00
数据标识: aebde637645886e730019602b1936e92d
创建时间: 2023-05-08 16:44:35
更新时间: 2023-05-08 16:44:35
所有人: lilaoshi
创建人: lilaoshi
修改人: lilaoshi
所属部门: -

流程图示：



7. 切换登录 lilaoshi 的部门负责人 zhangyuanzhang 的账号，zhangyuanzhang 在流程中心可查看 lilaoshi 提交的待审批记录。

我的待办

我的待办

搜索关键词: 搜索关键词
选择类型: 选择类型
开始时间: 选择时间
结束时间: 选择时间
查看
重置

lilaoshi提交的证书借用申请
发起人: lilaoshi
发起时间: 2023-05-08 16:44:37
已审核5分钟
去审批

lilaoshi提交的证书借用申请
发起人: lilaoshi
发起时间: 2023-05-08 16:44:37
已审核11分钟
去审批

8. 单击对应记录查看详情，单击同意。

并填写相关审批意见单击提交后完成审批。

9. 切换登录 kongzhuren 的账号，同样进行审批同意的操作，可看到单击同意 > 提交之后，要求为下一节点指定审批人。

此处我们可以把 admin 和 zhaochuzhang 都勾选，要求他们会签。

0. 分别登录 admin 和 zhaochuzhang 的账号，进行审批通过。

证书信用申请 - 材料证照

信用结束时间: 2023-05-31 00:00:00
信用开始时间: 2022-05-31 00:00:00
数据标识: aebd6376458b6730018602b7936e92d
创建时间: 2023-05-08 16:44:35
更新时间: 2023-05-08 16:44:35
所有人: Haoshi
创建人: Haoshi
修改人: Haoshi
所属部门: -

流转记录

- 学院审批... 审批人: zhangyuancheng [审核通过] 2023-05-08 16:54:44 审批意见: 审批通过
- 综合办公室审批... 审批人: kongzhuren [审核通过] 2023-05-08 16:58:07
- 指定会签 [审批人: 1448239258757623810, zhaochuzhang等2人待签] [待审核] 2023-05-08 16:58:08

搜索 换页

信用结束时间: 2023-05-31 00:00:00
信用开始时间: 2022-05-31 00:00:00
数据标识: aebd6376458b6730018602b7936e92d
创建时间: 2023-05-08 16:44:35
更新时间: 2023-05-08 16:44:35
所有人: Haoshi
创建人: Haoshi
修改人: Haoshi
所属部门: -

流转记录

- 学院审批... 审批人: zhangyuancheng [审核通过] 2023-05-08 16:54:44 审批意见: 审批通过
- 综合办公室审批... 审批人: kongzhuren [审核通过] 2023-05-08 16:58:07
- 指定会签 [审批人: 1448239258757623810] [待审核] 2023-05-08 16:58:08
- 指定会签 [审批人: zhaochuzhang] [待审核] 2023-05-08 16:58:08

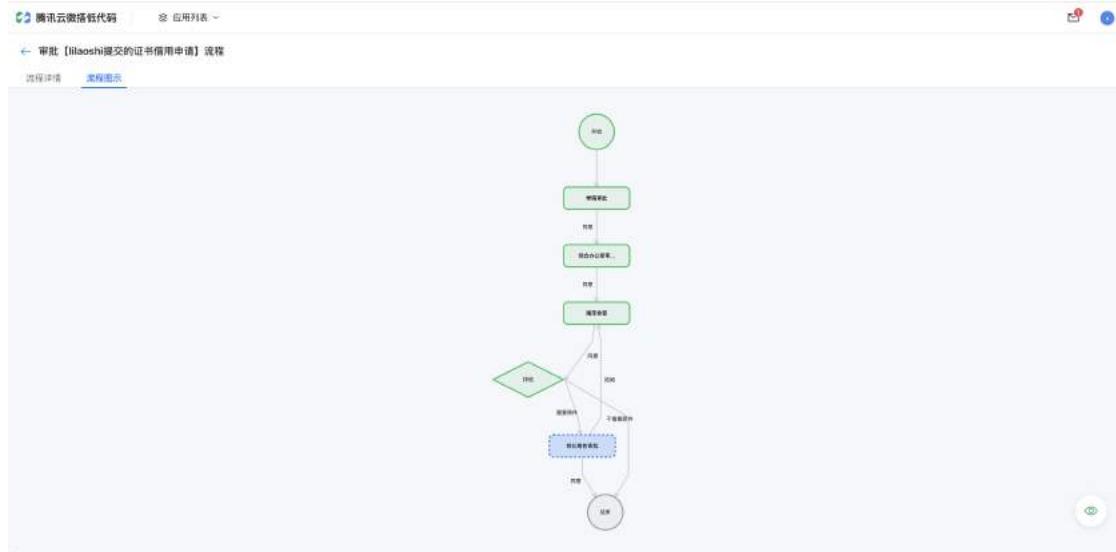
1. 此时流程会流转到校长角色的人审批，因为流程中我们发起时配置的是需要原件，所以需要校长角色进行复核，admin 之前我们设定的是校长，再登录 admin 的账号，可以看到待审核记录。

我的待办

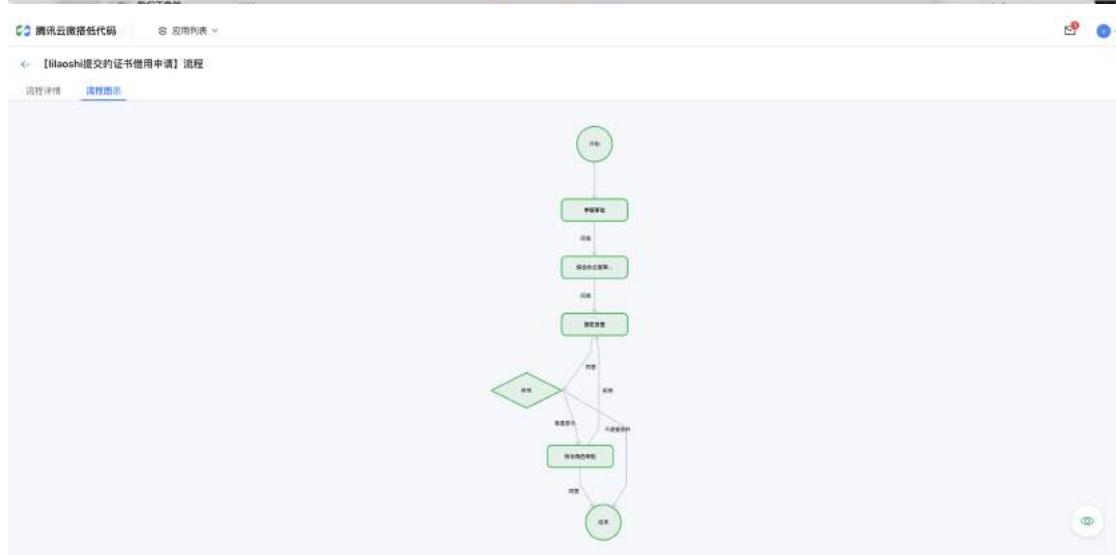
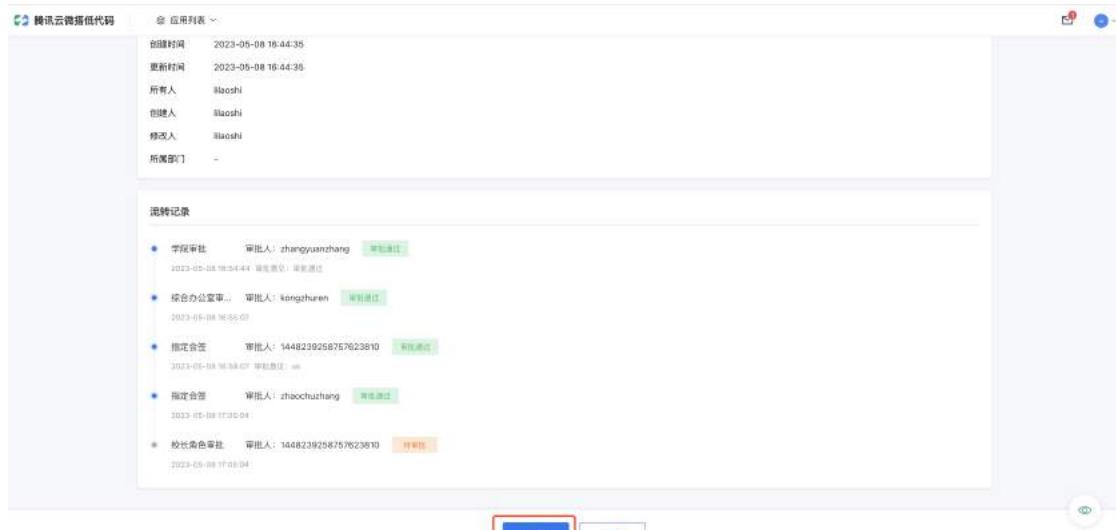
搜索流程标题关键字	选择发起人	发起时段	完成时段	查询	重置
Haoshi提交的证书信用申请	发起人: Haoshi	发起时间: 2023-05-08 16:44:37	已审核分钟	去审批	

10 页 1 / 1 1 / 1 1 / 1

2. 可以点击进入查看流程图示，流程已经走到最后节点。



3. 校长角色 admin 审批之后，整个流程完成。



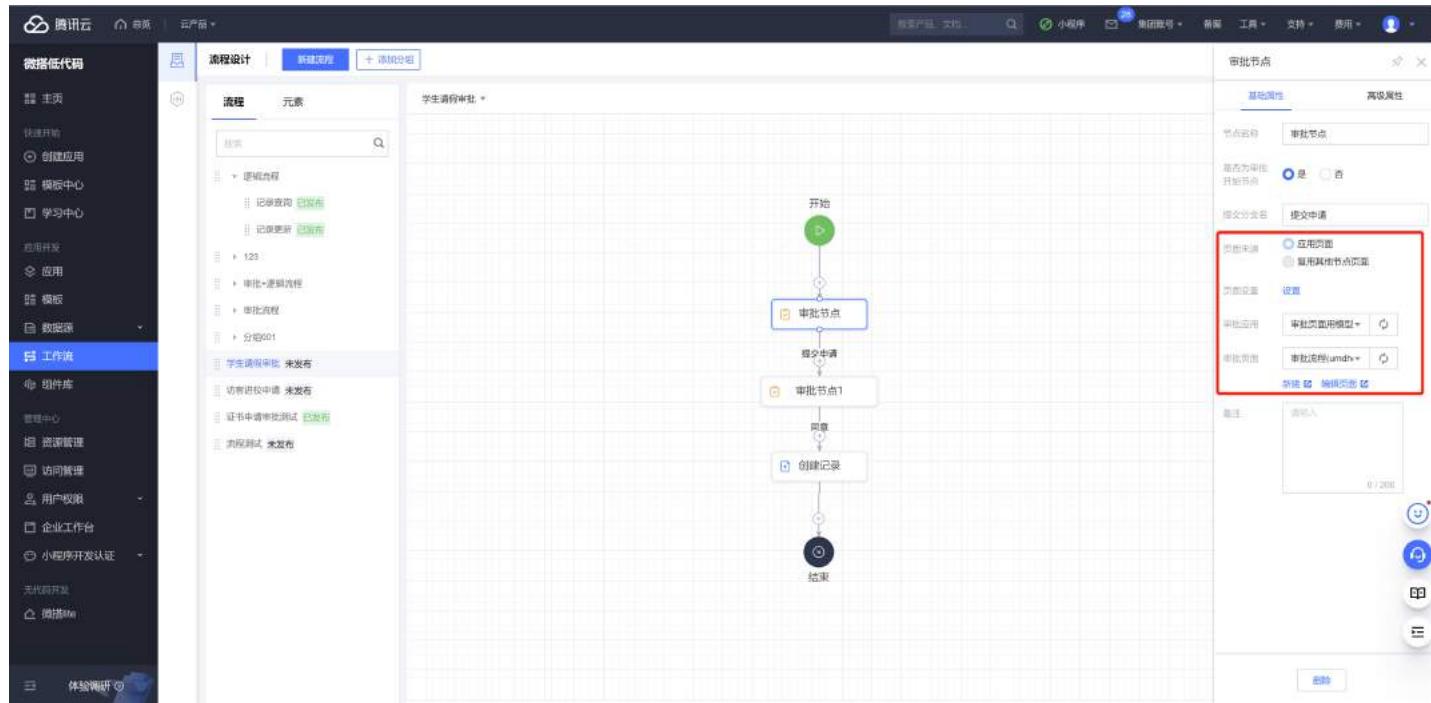
流程审批页面定制化

最近更新时间：2023-07-27 15:10:37

审批页面介绍

审批页面使用场景

审批页面为工作流中的人工任务节点定制页面，目的是实现工作流中的审批页面的定制化，例如工作流中的特定审批页面的内容想要实现定制，可以通过工作流节点关联应用中审批页面的方式实现。

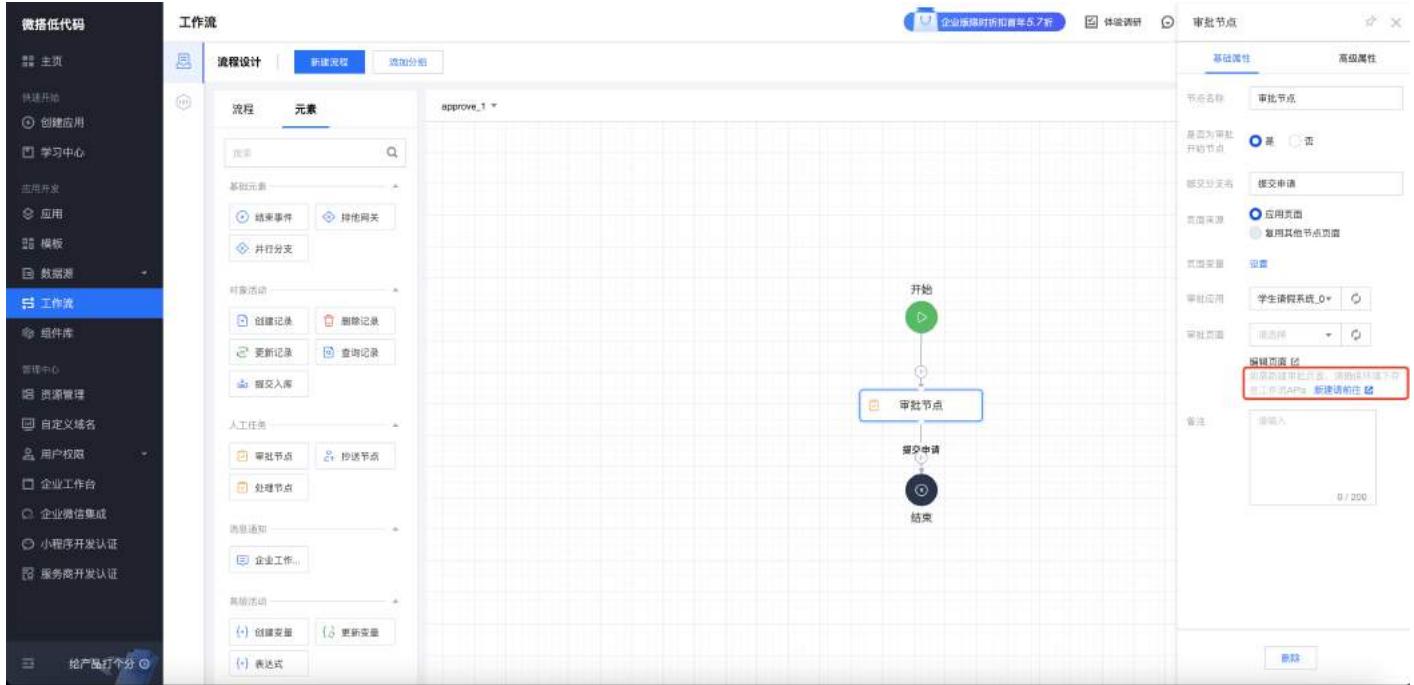


审批页面新建

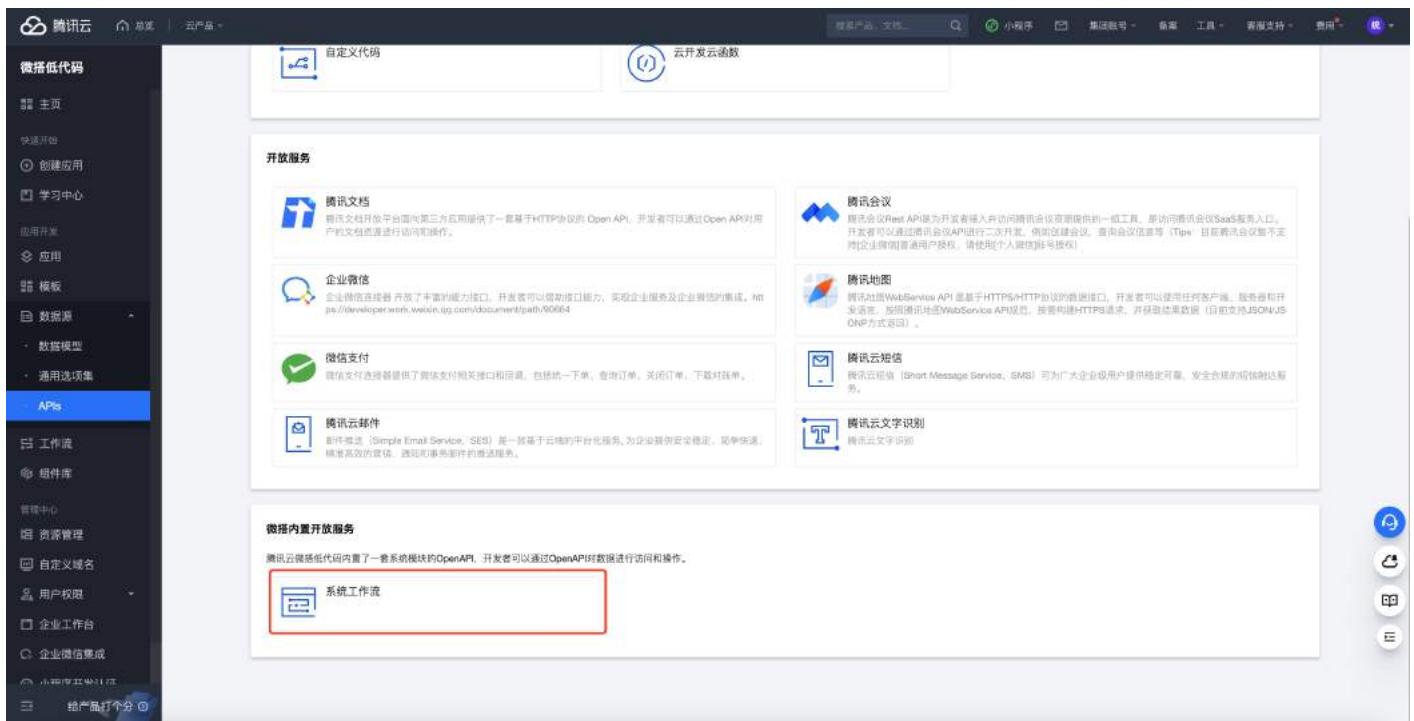
审批页面从工作流发起新建

从工作流发起新建

1. 进入工作流页面，如环境下不存在内置系统工作流 APIs 服务，单击新建请前往。



在微搭内置开放服务单击系统工作流。



填写相关内容后，单击创建。

选择 APIs 类型 > 配置 APIs 信息

自定义 APIs

名称 * 系统工作流

标识 * workflow

描述 请输入 APIs 描述，最多支持 256 个字符

参数配置

SecretId * user5

SecretKey * ****

[如何获取访问密钥](#)

完成 取消

⚠ 注意：

新建系统工作流时，SecretId 和 SecretKey 建议使用主账号密钥。

2. 流程的人工节点选择模型应用后，单击新建，可以直接调起所选模型应用审批页面的新建。

The screenshot shows the Tencent Cloud Workflow Designer interface. On the left sidebar, under the '工作流' (Workflow) section, there is a list of existing workflows: '学生请假审批' (Student Leave Application), '学生请假审批_未发布' (Student Leave Application - Not Published), '访客进校申请' (Visitor Entry Application), and '证书申请审批测试' (Certificate Application Approval Test). The '学生请假审批' workflow is currently selected and displayed in the main canvas.

The workflow diagram consists of the following steps:

- 开始 (Start) → 审批节点 (Approval Node)
- 审批节点 → 短文中请 (Short Message Application)
- 短文中请 → 审批节点 (Approval Node)
- 审批节点 → 同意 (Agree)
- 同意 → 创建记录 (Create Record)
- 创建记录 → 审批节点2 (Approval Node 2)
- 审批节点2 → 颁发申请 (Award Application)
- 颁发申请 → 结束 (End)

On the right side of the interface, there is a configuration panel for the current workflow step. It includes fields for '节点名称' (Node Name) set to '审批节点2' (Approval Node 2), '是否必经审批' (Must Be Approved) checked, '提交申请' (Submit Application) button, '页面来源' (Page Source) set to '应用页面' (Application Page), '页面配置' (Page Configuration) settings, and a '新流转' (New Flow) button highlighted with a red box.

发起的新建审批页面会生成以流程名称和节点名称结合命名的审批页面。

审批页面基础要素

新建审批页面之后页面预置内容包含流程图示/流转记录、审批按钮，除流程图示外，页面其他元素均由官方组件搭建，支持根据业务场景进行页面调整。

发起的新建审批页面，如果流程中配置了页面变量，单击新建后，除了预置的页面内容，还会根据流程中配置的页面变量生成内容。

审批页面编辑

选中审批页面大纲树中的内容，可以删除、添加、修改组件，对组件进行编辑。

△ 注意：

审批页面新建和编辑之后，单击保存才可为流程中的节点所选择。

审批页面发布

组件项	类型	状态	当前版本	操作
vicky_test > 页面 > 布局组件	数据模型	未发布	-	发布 忽略 查看详情
区块-表单提交二 (bd9d2_329fa0)	数据模型	未发布	-	发布 忽略 查看详情
区块-表单提交一 (qk0dtly_t9v1ct)	数据模型	未发布	-	发布 忽略 查看详情
vicky_test (vicky_test_30mf653)	数据模型	未发布	-	发布 忽略 查看详情
OpenAPIs 3.0 (openapi00_ue89eep4)	APIs	未发布	-	发布 忽略 查看详情
用户 (sys_user)	数据模型	发布成功	-	
登录用户名 (dlyh6_pc7zmvv)	数据模型	发布成功	-	
部门 (sys_department)	数据模型	发布成功	-	
test3	流程	已发布, 有更新	4	发布 忽略 查看详情
角色组 (权限开发者,企业工作台,企业工作台,权限,官网运营人员,普通角色,后勤主任,后勤主任)	角色	已发布, 有更新	-	发布 忽略 查看详情
流程页面 审批流程-审批节点	流程页面	已发布, 有更新	-	生成 忽略 查看
流程页面 学生请假-审批节点	流程页面	已发布, 有更新	-	生成 忽略 查看

审批页面从流程中新建时，模型应用发布时会和流程人工节点设定的页面变量进行关联检查，如果流程中页面变量有更新，会在发布时提示用户审批页面是否要同步数据：

- 单击发布会同步流程中最新的页面变量。
- 单击忽略保持当前页面内容进行应用发布。
- 单击查看跳转到工作流。

流程定制审批页面使用示例

新建模型应用

首先新建一个模型应用学生请假申请_勿删。

The screenshot shows the 'Student Leave Application -勿删' page in the 'Leave Application Data - Document Delete' category. The left sidebar includes 'Page Design', 'Variables', 'Navigation Menu', 'Code Editor', and 'More'. The main area displays a table titled 'Student Leave Application - Document Delete List' with columns: 'Leave Type', 'Leave Reason', 'Leave Days', 'Student Identifier', and 'Operation'. The right sidebar contains sections for 'Page Properties', 'Page URL Parameters' (with a note about opening the page via URL), 'Screen Configuration', and 'Events' (including 'Page Load' and 'Page Show').

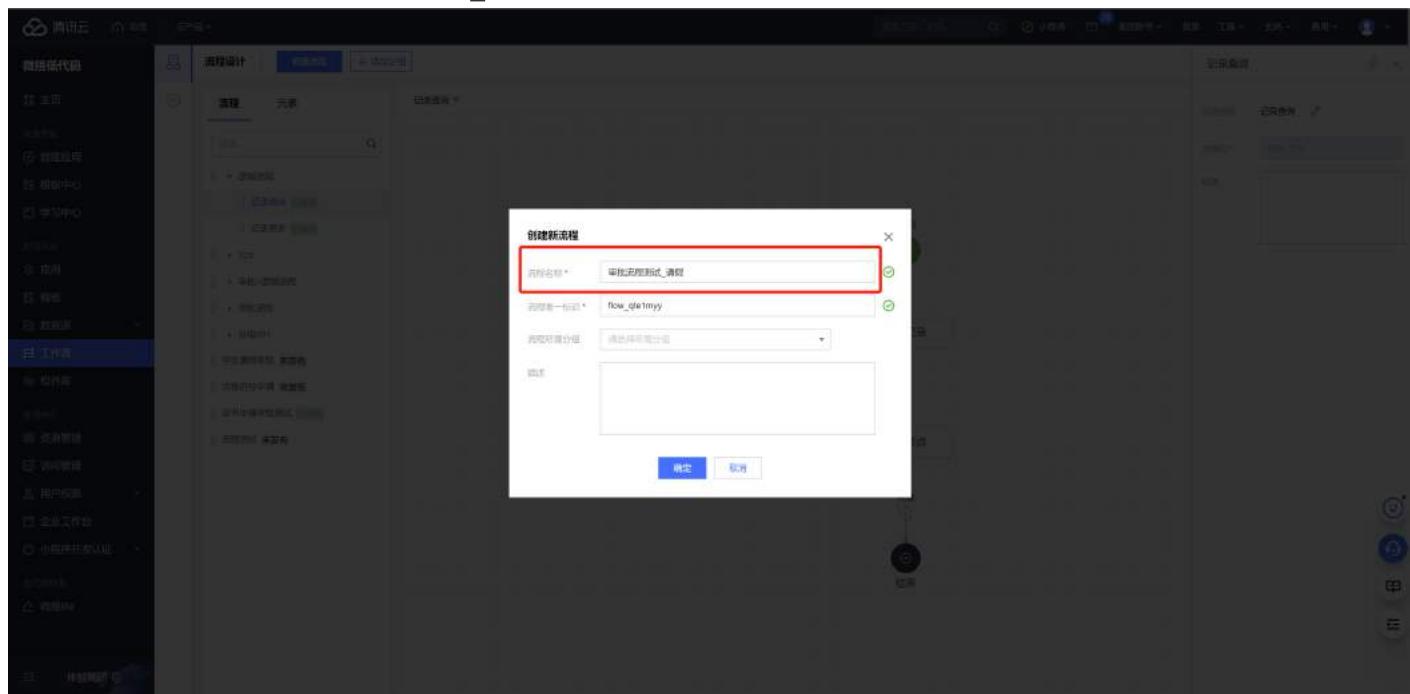
新建数据模型

新建学生请假数据相关的数据模型，包含请假类型、请假原因、请假天数等字段。

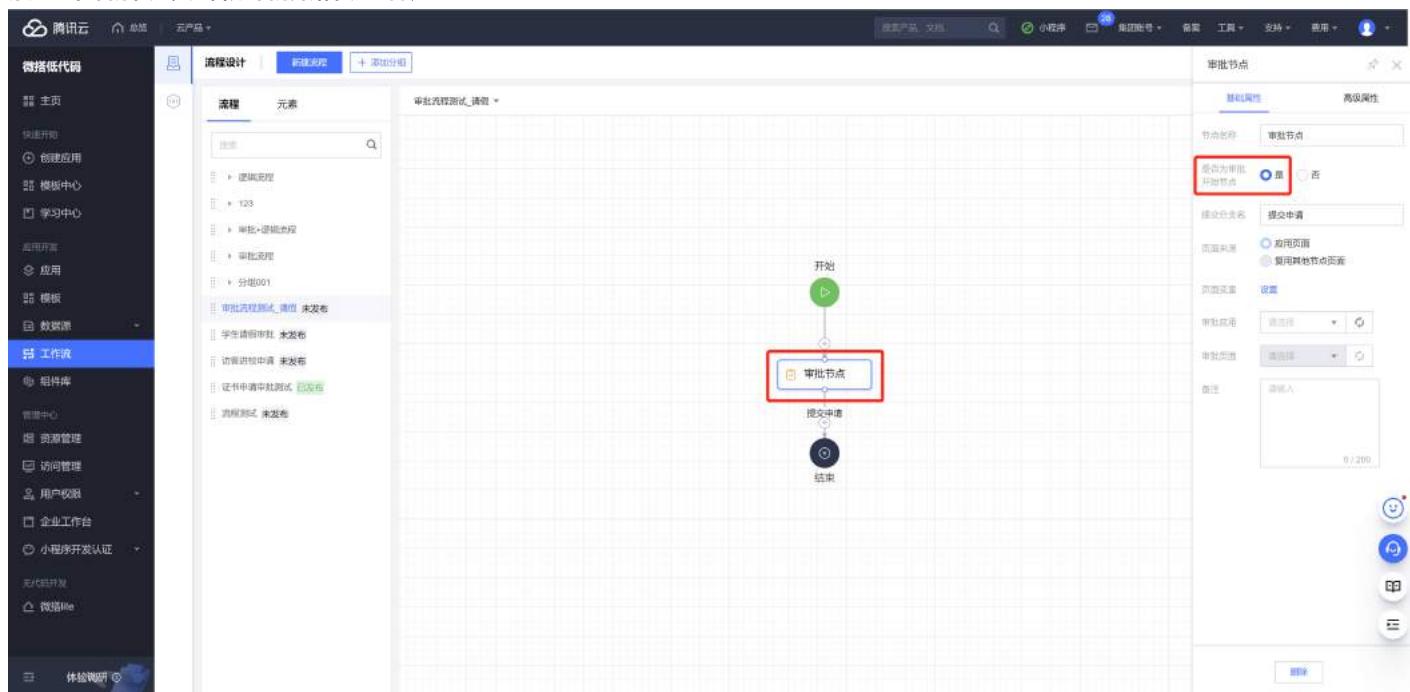
This screenshot is identical to the one above, showing the 'Student Leave Application -勿删' page in the 'Leave Application Data - Document Delete' category. It features the same table structure and sidebar options, demonstrating the process of creating a data model for leave applications.

新建流程

1. 进入工作流页面，单击新建流程，新建审批流程测试_请假流程。



2. 添加一个审批节点，是否为审批开始节点选择是。



3. 单击页面变量 > 设置，页面输出变量配置对象—学生请假数据_勿删—QJXX，单击确定。

字段名称	字段标识	数据类型	是否必填	是否唯一
学生标识	xbs	文本 单行文本	否	否
请假天数	qts	数字	否	否
请假原因	qwy	文本 单行文本	否	否
请假类型	qjx	枚举	否	否

方法名	描述
wodaCreate	新增
wodaDelete	删除
wodaUpdate	更新
wodaBatchUpdate	更新多条
wodaQuery	查询
wodaBatchQuery	查询列表

4. 审批应用选择学生请假申请_勿删。

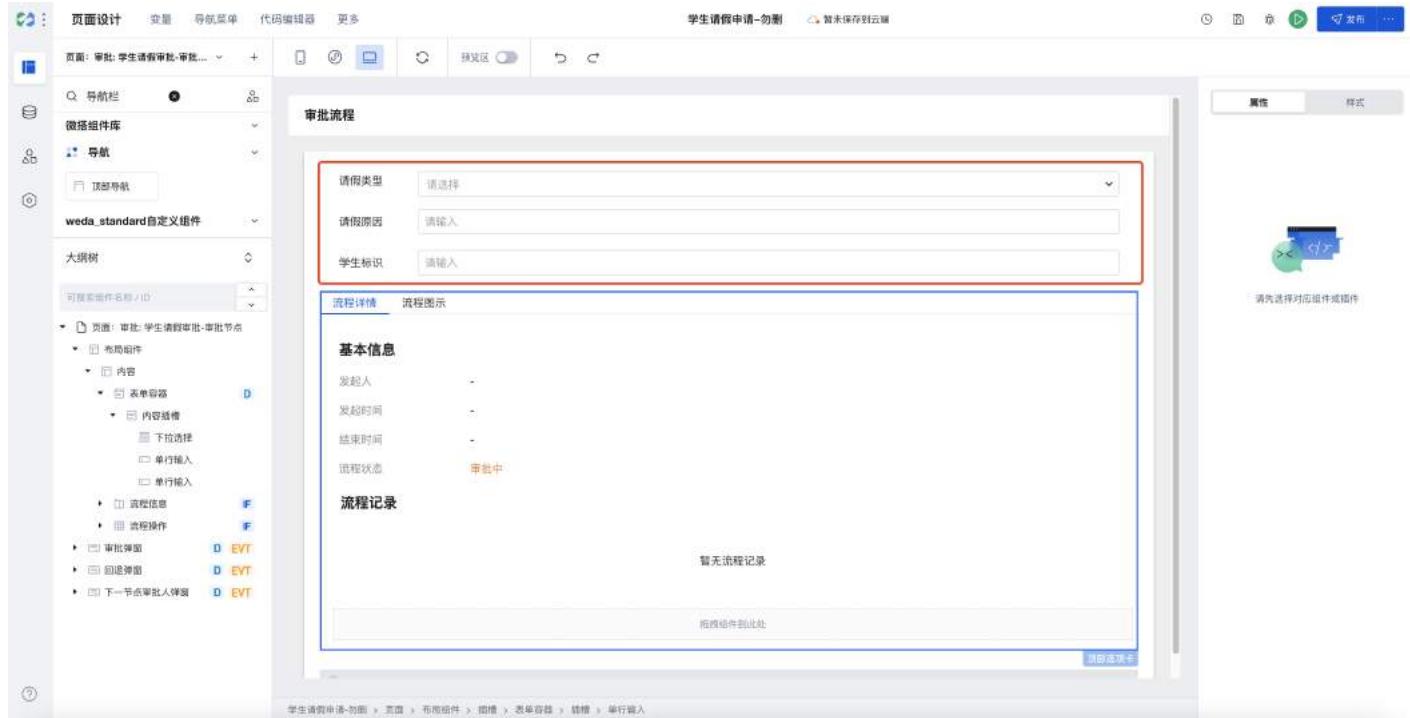
```

graph TD
    Start((开始)) --> Approval{审批节点}
    Approval -- 提交申请 --> End((结束))
  
```

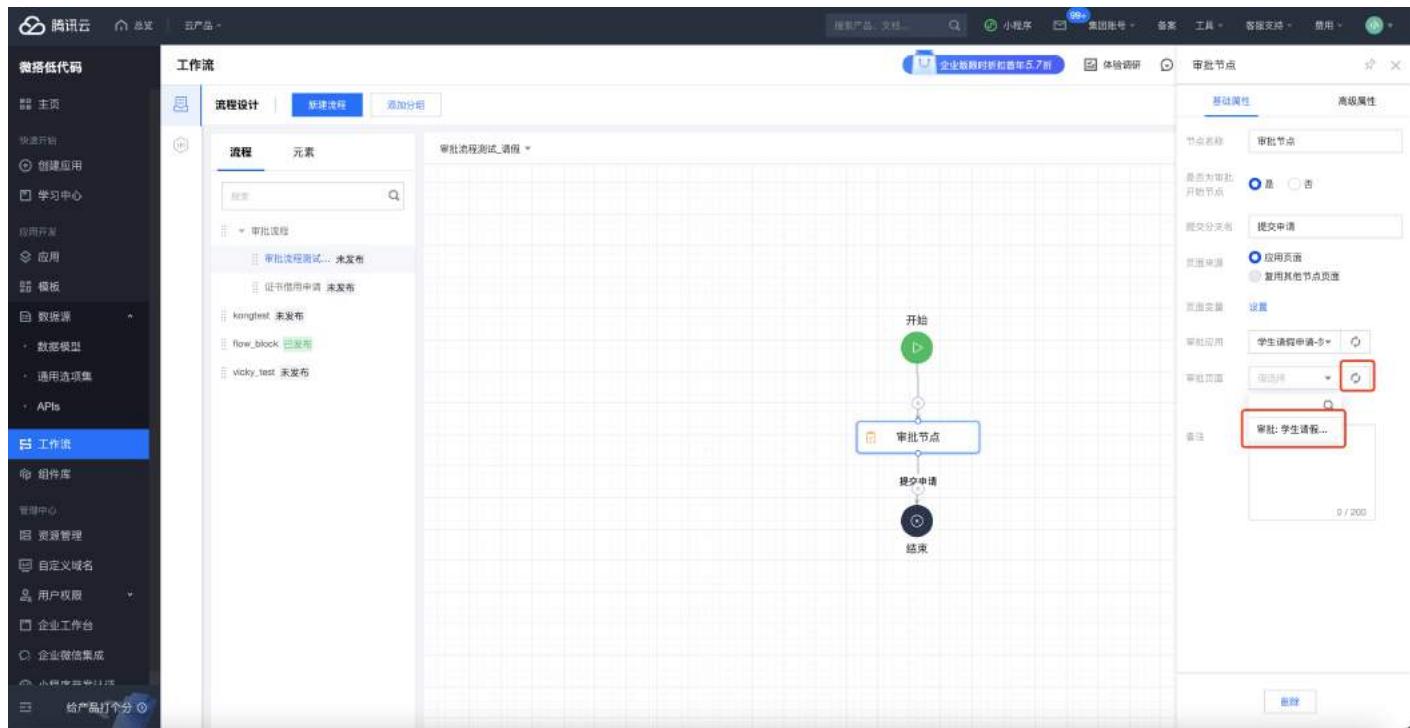
5. 此时审批页面下拉选择时没有审批页面可选，单击新建，跳转新建审批页面。

6. 单击新建跳转到模型应用中新建审批页面，审批页面会根据流程的页面变量自动生成页面内容。

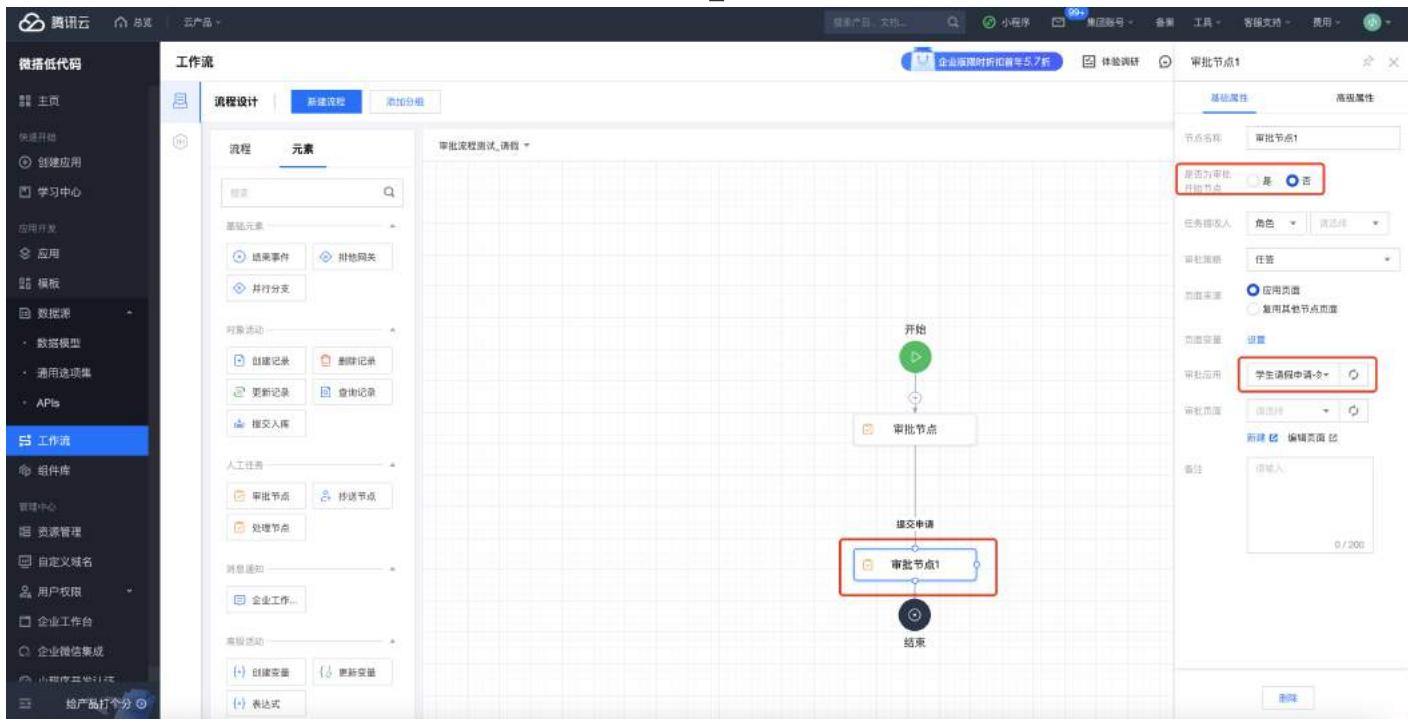
7. 可以修改审批页面中内容，例如我们删除请假天数单行输入框（那么流程发起者在输入内容时就不会看到请假天数输入框）。



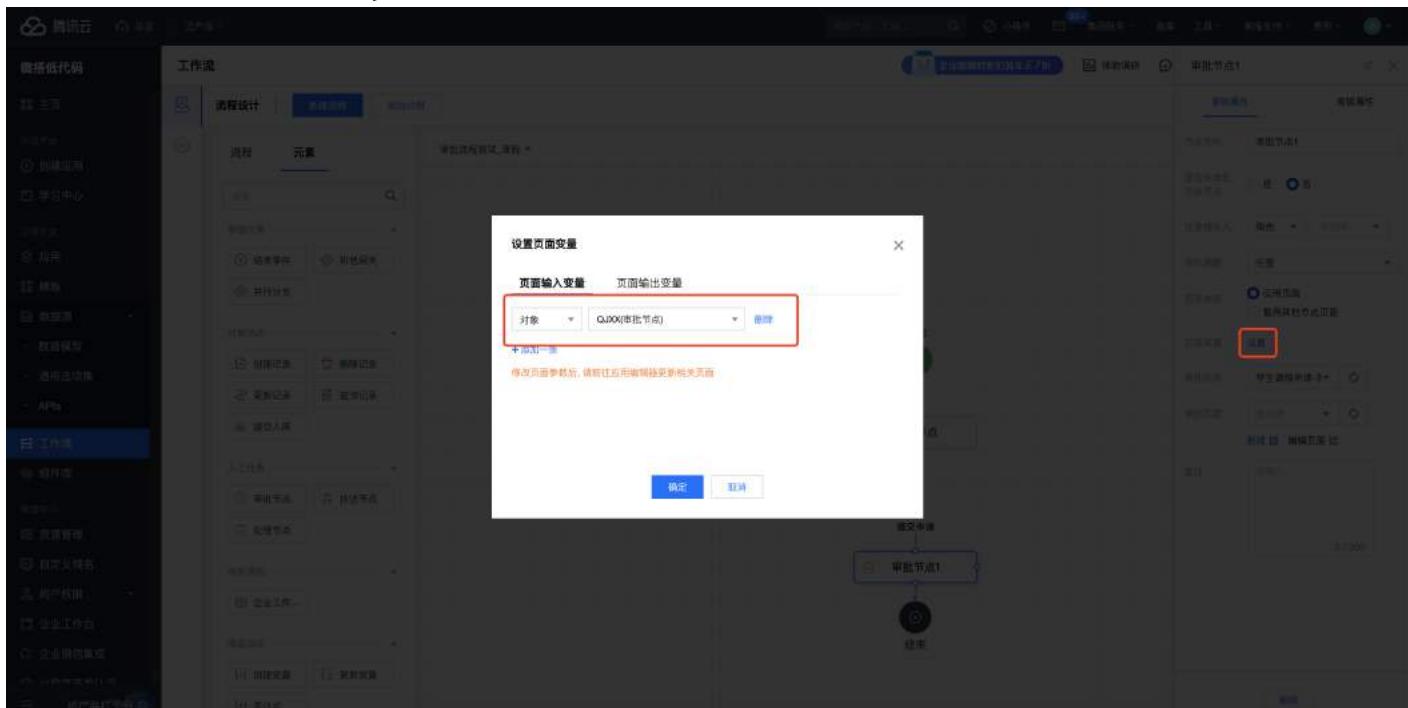
8. 模型应用中可以单击保存，在工作流中单击审批页面刷新按钮，就可以看到可选择的审批页面。



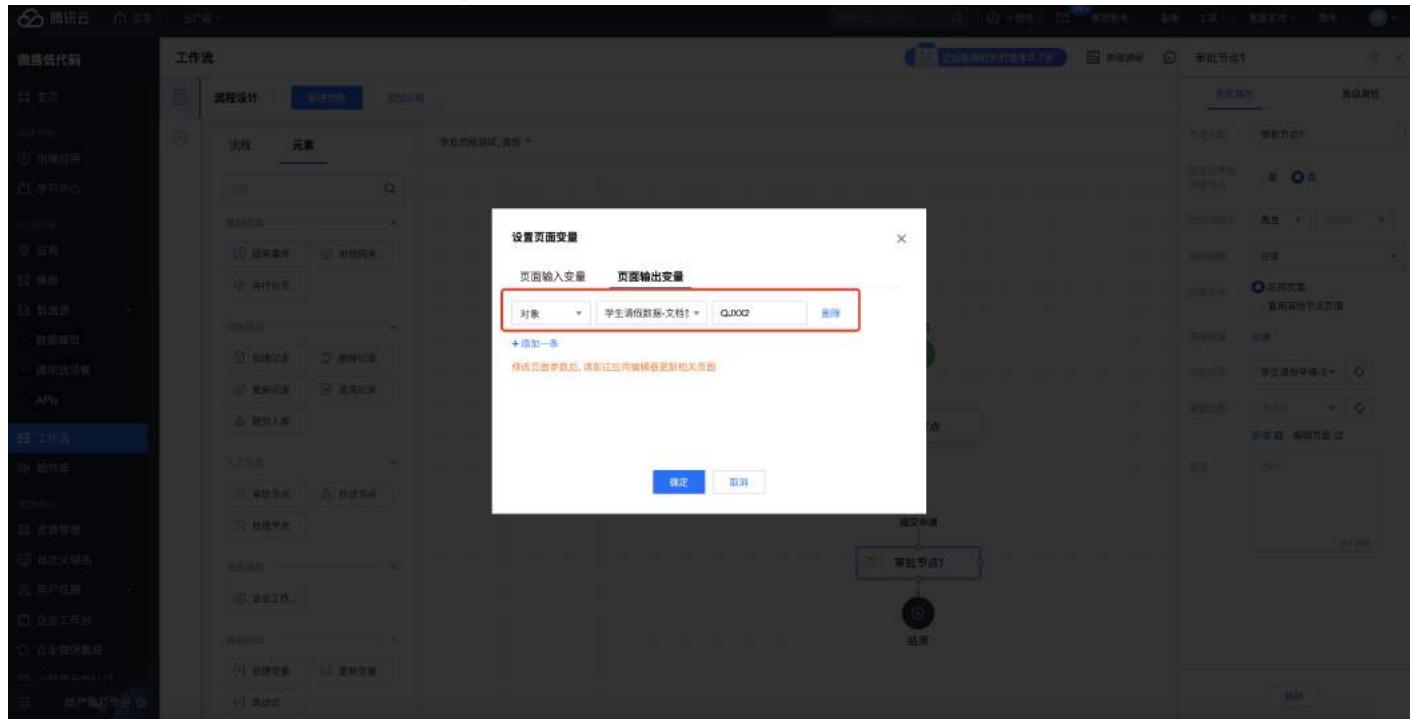
9. 再添加一个审批节点，是否为审批开始节点选择否，审批应用选择学生请假申请_勿删。



页面变量设置，页面输入变量设置对象-QJXX。



页面输出变量设置对象-学生请假申请_文档勿删-QJXX2。



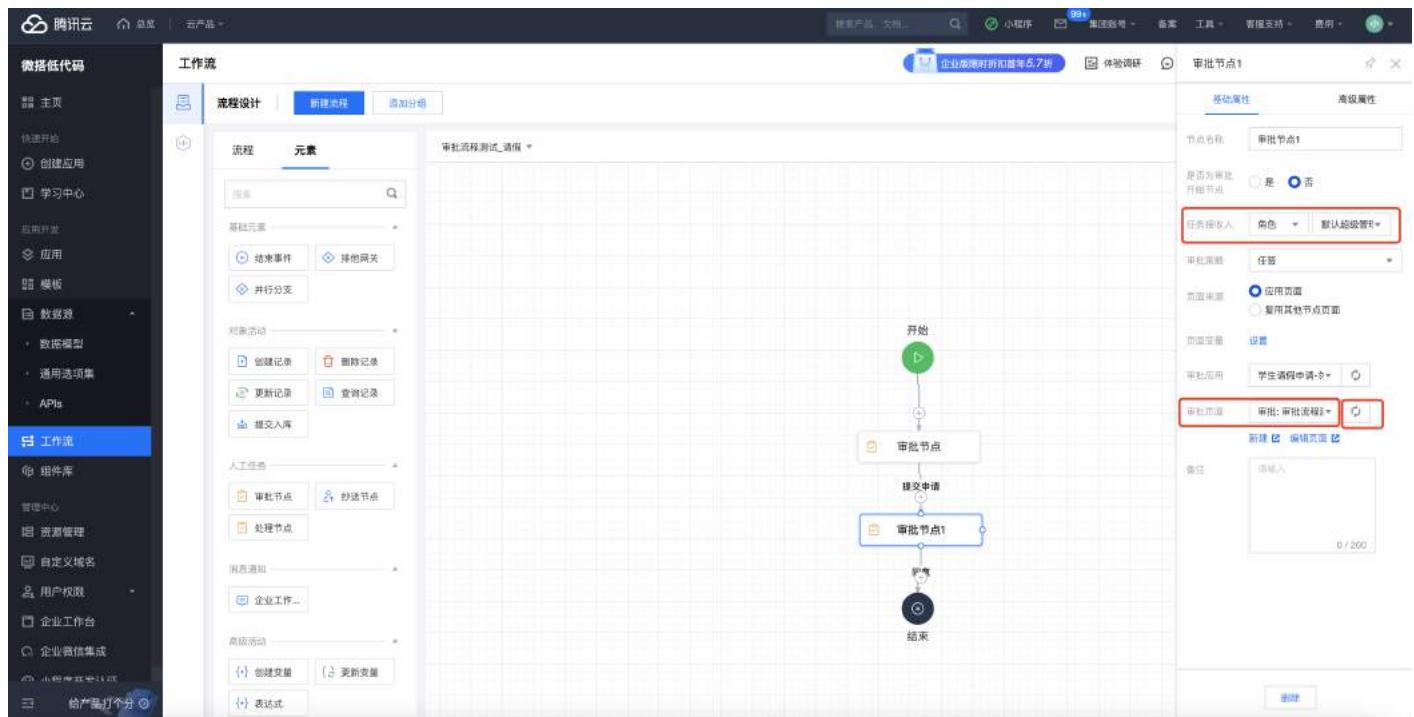
0. 变量配置完成之后，单击新建，再建立一个定制化的审批页面。

新的定制化的审批页面中，可以对请假原因、请假类型设置禁用（审批者这两个要素只可以查看不可以输入和修改），设置完成后单击保存模型应用。

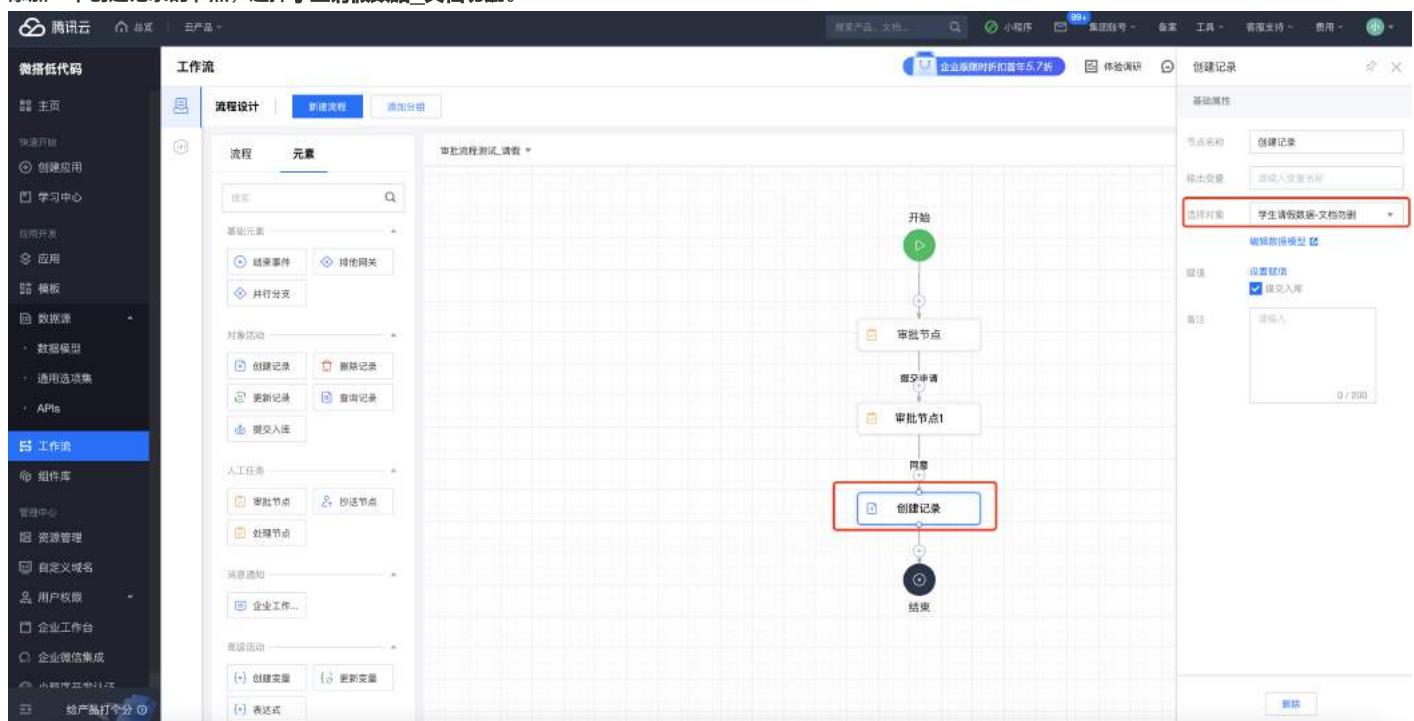
1. 工作流中选择配置好的审批页面，审批任务接收人可以设定角色-默认超级管理员。

⚠ 注意：

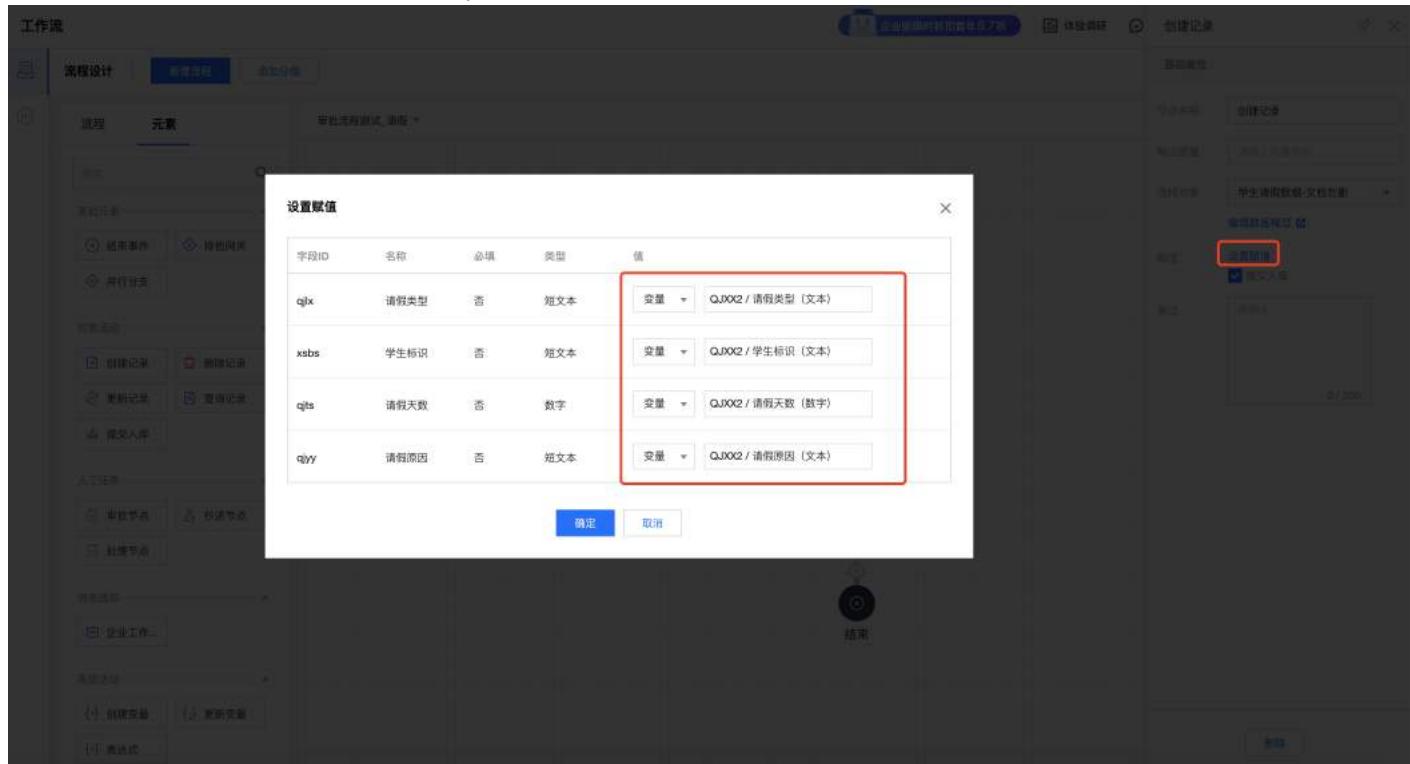
如接收人设定为非超管时，需要对接收人进行权限配置，请参见[角色权限配置](#)。



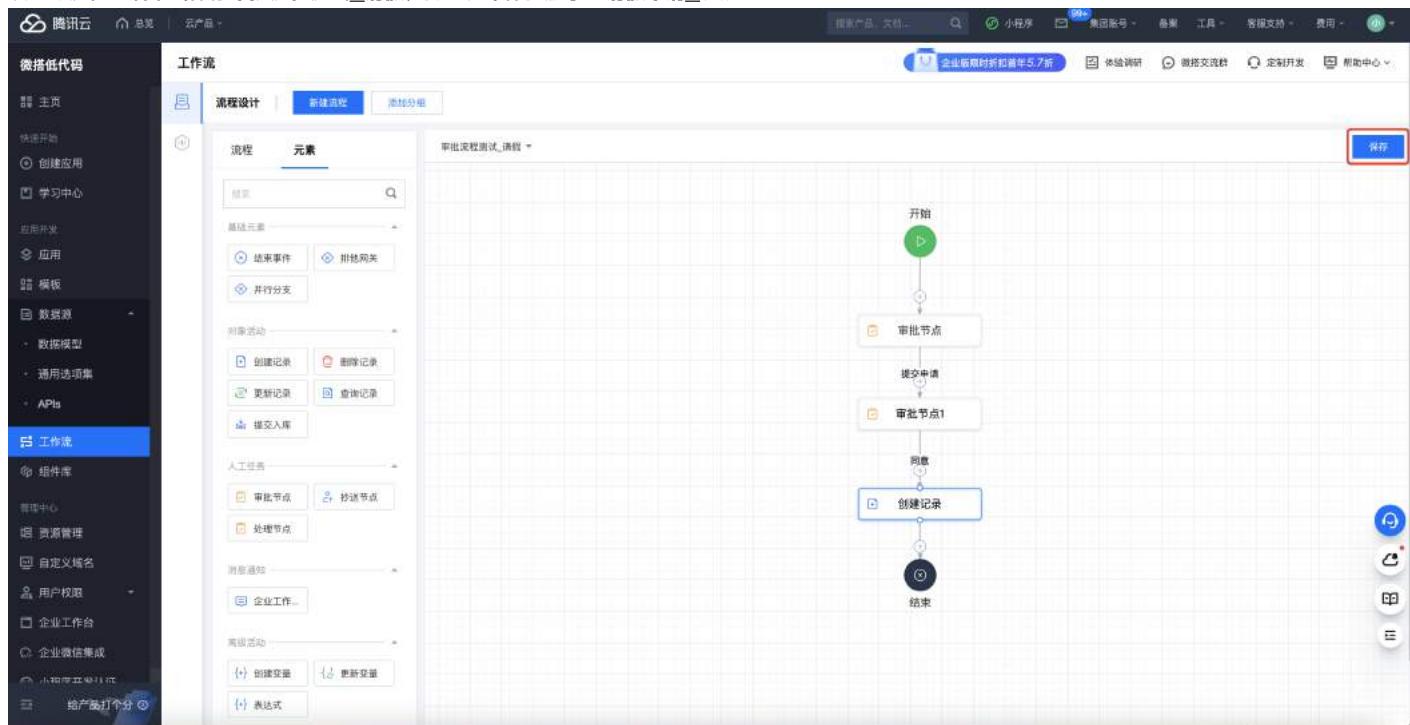
添加一个创建记录的节点，选择学生请假数据_文档勿删。



单击设置赋值，值都选择变量，具体的变量值选择 QJXX2 中对应的值。赋值配置完成的效果如下，配置完成之后单击确定。



2. 设置好流程之后单击保存审批流程测试_请假，同时发布体验版学生请假申请_勿删。



发起流程

1. 模型应用发布后可以单击访问企业工作台。

2. 默认会使用 admin 账号登录企业工作台。

3. 单击切换到流程中心，单击发起流程，可以看到刚才新建的审批流程测试_请假在流程中心中。

The screenshot shows the 'Initiate Process' section of the platform. On the left sidebar, there are tabs for 'My Pending Tasks', 'My Handled Tasks', 'My Initiated Tasks', 'Copied to Me', and 'Draft Box'. The 'Initiate Process' tab is selected and highlighted with a red border. In the main area, there is a search bar with '搜索流程标题关键字' and buttons for 'Search' and 'Reset'. Below the search bar, there is a list of processes. One process, '审批流程 (1)' (Approval Process (1)), has a sub-item '审批流程测试_请假' (Approval Test_Leave Application) which is also highlighted with a red border.

4. 单击审批流程测试_请假，填写内容，单击提交申请按钮发起审批。

The screenshot shows the 'Approval Process' form for the leave application. The left sidebar lists 'Student Leave Application Data List', 'Create Student Leave Application Data', 'Approval - Approval Node', and 'Approval - Approval Node 1'. The 'Approval - Approval Node' item is selected and highlighted with a red border. The main form has fields for 'Leave Type' (年假), 'Leave Reason' (出去看看), 'Leave Days' (5), and 'Student Identifier' (请输入). At the bottom of the form are three buttons: 'Save Draft' (保存草稿), 'Cancel' (取消), and 'Submit Application' (提交申请), with the 'Submit Application' button being blue.

此处我们在流程中审批者（任务接收人）设置的是 admin 角色，提交申请之后可以在我的待办中看到待审批记录。

5. 单击进入该条记录，可以看到请假类型和请假原因不可输入（审批页面中我们设定的请假类型和请假原因不可输入）。

6. 请假时长我们输入6，单击同意通过审批。

The screenshot shows a leave application form. The '请假天数' (Leave Days) field is highlighted with a red border and contains the value '6'. The '流程详情' (Process Details) tab is selected, showing the basic information and history of the application.

发起人	1448239258757623810
发起时间	2023-05-10 16:02:45
结束时间	-
流程状态	审批中

流程记录

审批节点	操作人	审批意见	时间
审批节点1	待审批	-	2023-05-10 15:37:48
审批节点1	1542073835002783489	-	2023-05-10 15:37:48

可以在我已处理中看到审批通过的记录。

The screenshot shows a list of processed leave applications under the '已处理' (Processed) tab. One specific application is highlighted with a red border.

发起人	发起时间	操作	时间
administrator	2023-04-13 14:58:33	我的审批	审批时间 2023-04-13 15:09:49
administrator	2022-10-25 14:38:43	我的审批	审批时间 2022-10-25 15:28:44
administrator	2022-09-14 14:15:12	我的审批	审批时间 2022-09-14 14:23:20
administrator	2022-09-14 14:14:22	我的审批	审批时间 2022-09-14 14:14:46
administrator	2022-09-13 20:02:47	我的审批	审批时间 2022-09-13 20:03:44
administrator	2022-09-13 20:02:47	我的审批	审批时间 2022-09-13 20:03:22
administrator	2022-09-13 20:00:32	我的审批	审批时间 2022-09-13 20:02:06
administrator	2022-09-13 17:10:57	我的审批	审批时间 2022-09-13 17:17:40

同时可以在数据模型-学生请假数据中查看到该条记录。

The screenshot shows a data management interface for a leave application model. At the top, there are tabs for '我的应用' (My Applications) and '我的数据源' (My Data Sources), with '我的数据源' being active. On the right, there is a 'Google Translate' button and a user profile for 'administrator'. The main area displays a table with one row of data, which is highlighted with a red border. The columns in the table are: 请假类型 (Leave Type), 请假原因 (Leave Reason), 请假天数 (Leave Days), 学生标识 (Student Identifier), 数据标识 (Data Identifier), 创建时间 (Create Time), 更新时间 (Update Time), and 操作 (Operations). The data in the table is: 年假 (Annual Leave), 出去看春 (Out to see the spring), 6 (6 days), -, f6a87246645b524..., 2023-05-10 16:13:54, 2023-05-10 16:13:54, and a row of buttons for 查看 (View), 编辑 (Edit), and 删除 (Delete).

用户权限

概述

最近更新时间：2022-03-16 19:56:42

用户泛指可以访问系统应用内资源的人员，在 C 端场景下是终端用户，在 B 端场景下主要就是企业内部员工、外部客户。一般会根据用户的角色权限，管控不同用户可以访问的应用、查看的页面、参与的审批流程以及可以操作数据的范围，这是系统安全中非常重要的一部分。例如，IT 管理员可以访问所有应用、查看所有页面和数据，普通员工则只能查看自己的数据。HR 部门只有特定人员可以查看员工的工资字段，而其他人员只能查看员工的基本信息等等都依赖于这一套用户权限体系。

用户管理

企业内部用户

最近更新时间：2023-07-12 17:25:53

企业内部用户泛指员工，可以登录企业工作台使用模型应用。

例如：为了满足电商场景需求，开发者使用微搭创建并发布了两个应用，一个面向 C 端用户的购物小程序（自定义应用），一个面向内部运营人员的商品进销存管理系统（模型应用）。这种情况下，使用购物小程序购物的用户为外部用户，使用进销存管理系统的用户为内部用户。

前提条件

新建用户

腾讯云控制台进入微搭低代码 **用户权限 > 用户** 页面，单击**新建用户**。

The screenshot shows the 'Internal User / Department Management' interface. On the left, there's a sidebar with 'Internal User' selected. The main area is titled 'All Organization Users' and contains a table with columns: 'User Name', 'User Nickname', 'Enterprise Account Number', 'Associated Roles', 'Mobile Number', and 'Operations'. There are 25 users listed, each with a checkbox and a detailed view button. At the bottom, there are pagination controls for 10 items per page, with the current page being 1 of 3.

User Name	User Nickname	Enterprise Account Number	Associated Roles	Mobile Number	Operations
白丽	teletts11	测试	-	13283445877	关联角色 更多 ▾
白丽	bobby-zhuren	bobbyzhuren	-	主任	关联角色 更多 ▾
白丽	bobby-chuzhang	bobbychuzhang	-	处长	关联角色 更多 ▾
白丽	yanglaoshi	yanglaoshi	-	老师-bob	关联角色 更多 ▾
白丽	bobby-yuanzhang	bobbyyuanzhang	-	院长	关联角色 更多 ▾
白丽	888881	222	-	校外人员、放假、测试权限...	关联角色 更多 ▾
白丽	joming	黎明	-	学生管理员	关联角色 更多 ▾
白丽	simonyyang	系统管理员	-	默认超级管理员	关联角色 更多 ▾
白丽	uin_leader_root	-	-	默认超级管理员	关联角色 更多 ▾
白丽	uin_leader	-	-	默认超级管理员	关联角色 更多 ▾

操作步骤

步骤1：填写用户信息

设置用户信息，此处仅对部分配置项进行必要说明。

- 用户名：**企业工作台进行账号登录时，使用的用户名，定义时需要注意下用途。
- 密码：**结合用户名使用，进行账号登录时的密码，密码信息微搭可以进行初始密码的设定，密码设置以后为了安全起见，无法查看密码。
- 用户昵称：**中文姓名，用于结合用户名显示时使用。

新建用户

X

1 填写用户信息 > 2 产品许可证 > 3 设置角色权限 (选填)

用户名 * test

用户昵称 * 测试

密码 *  

邮箱

手机号

直属上级

主岗部门

兼岗部门

描述

人员岗位名称

测试人员名称

职能部门审核 

 取消  下一步

步骤2：授权产品许可证

对于企业员工，需要创建用户时先授权微搭的用户 license，未授权产品许可证的用户，对微搭应用产物的访问可能受到限制或无法访问。

新建用户

X

1 填写用户信息 > 2 产品许可证 > 3 设置角色权限 (选填)

向用户授权产品许可证

请选择授权产品许可证的套餐包

企业版 {23 个可用/共 48 个} 

创建用户但不授权产品许可证 (不推荐)

未授权产品许可证的用户，对微搭应用产物的访问可能受到限制或无法访问

 上一步  下一步

步骤3：设置角色权限

可以为用户分配其职责范围内的使用权限。

新建用户

1 填写用户信息 > 2 产品许可证 > 3 设置角色权限 (选填)

请为新建的用户设置角色，此步骤可跳过，后续可在用户列表关联角色。

管理员 

无合适角色? [立即新建](#) 

院长 

无合适权限组? [立即新建](#) 

[上一步](#) [保存](#)

设置用户直属上级

新建或修改用户基本信息时，可以设置其直属上级。可用于数据权限管理和审批流相关场景。

新建用户

1 填写用户信息 > 2 产品许可证 > 3 设置角色权限 (选填)

用户名 *

用户昵称 *

密码 * 

邮箱

手机号

直属上级

主岗部门

兼岗部门

描述

人员岗位名称

测试人员名称

职能部门审核

[取消](#) [下一步](#)

用户基本信息扩展

可以在用户表里新增自定义字段，然后在新建用户时填写定义其扩展信息。

The screenshot shows the 'Model Configuration' section under 'Data Model / User'. It displays a table of 'Model Fields' with columns: 字段名称 (Field Name), 字段标识 (Field Identifier), 数据类型 (Data Type), 是否必填 (Required), 是否唯一 (Unique), and 操作 (Operations). Three fields are listed: 测试人员名称 (carymc), 人员岗位名称 (rygwnmc), and 部门ID (zimbmeh). A '展开系统字段' (Expand System Fields) button is visible, along with a '+' icon for adding new fields.

运行动态支持内部用户管理

企业工作台进入系统设置 > 用户权限页面，可以直接在应用产物里直接管理用户。

The screenshot shows the 'User Department Management' page under 'System Settings'. The left sidebar includes sections like Workstation Settings, Portal Configuration, Login Configuration, User Groups, User Department Management (selected), Role and Permission, Permissions, Process Management, Application Management, and Audit Log. The main area shows a list of users in the 'hr部门' (hr Department) group. Each user entry includes fields: 姓名 (Name), 用户名 (Username), 用户昵称 (Nickname), 企业微信号 (Enterprise WeChat ID), 已关联角色 (Associated Roles), 手机号 (Mobile Number), and 操作 (Operations). The first user listed is 'joming'.

组织架构

最近更新时间：2023-05-05 15:15:44

组织架构是企业管理系统常用的功能，包括部门管理，部门上下级管理，以及部门负责人等等，主要用于组织间数据管理、权责划分，以及流程的运转。

The screenshot shows the 'Internal User/Department Management' section. On the left, there's a sidebar with a search bar for 'Search department name' and buttons for 'Create Department' (highlighted in blue), 'Import User', 'Associate Role', and 'More Operations'. Below this are filters for 'All Organization Users' and 'Unlinked Department Users'. The main area displays a table with columns: 机号 (Machine Number), 邮箱 (Email), 账号状态 (Account Status), 主部门 (Primary Department), 隶属部门 (Subordinate Department), 部门负责人 (Department Manager), 更新时间 (Last Update Time), and 操作 (Operations). A single row is shown for 'Test2', which is associated with 'Test2' as the primary department and 'Total Office' as the subordinate department, with '否' (No) as the manager status.

新建部门

腾讯云控制台进入微塔低代码 **用户权限** **用户** > 页面，单击**新建部门**。

可以设置部门名称及对应的上级部门。如果未设置上级部门，则默认为一级部门。

The dialog box is titled 'New Department'. It contains three input fields: 'Department Name' with value 'Test3', 'Department Identifier' with value 'cs3', and 'Parent Department' with value 'Test2'. At the bottom are two buttons: 'Confirm' (highlighted in blue) and 'Cancel'.

设置部门负责人

单击设置部门负责人。

The screenshot shows the 'Internal User/Department Management' section. On the left, there's a sidebar with a search bar for 'Search department name' and buttons for 'Create Department' (highlighted in blue), 'Import User', 'Associate Role', and 'More Operations'. Below this are filters for 'All Organization Users' and 'Unlinked Department Users'. The main area displays a table with columns: 机号 (Machine Number), 邮箱 (Email), 账号状态 (Account Status), 主部门 (Primary Department), 隶属部门 (Subordinate Department), 部门负责人 (Department Manager), 更新时间 (Last Update Time), and 操作 (Operations). Two rows are shown for 'Sales Department', both associated with 'Sales Department' as the primary department and 'Test1' as the subordinate department, with 'Yes' as the manager status.

在弹窗中选中该部门当前直接负责人

设置部门负责人

X

请选择部门负责人

请输入用户名/手机号/邮箱

用户名
<input type="checkbox"/> joming
<input checked="" type="checkbox"/> yy4_dfwu
<input checked="" type="checkbox"/> sherrie

已选择 2 条

用户名
<input type="button" value=""/>
yy4_dfwu
<input type="button" value=""/>
sherrie
<input type="button" value=""/>

设置部门负责人以后，默认用户列表“部门负责人”列，会显示此人是否是该部门负责人。

销售部

机号	邮箱	账号状态	主部门	兼岗部门	部门负责人	更新时间	操作
-	-	已授权	销售部	-	否	2022-09-20 18:03:33	关联角色 更多
52	41...@qq.com	已授权	-	销售部,测试1	是	2022-09-20 18:04:03	关联角色 更多
-	-	已授权	-	销售部,招聘...	是	2022-09-20 18:03:49	关联角色 更多

主岗、兼岗部门

每个员工默认会有一个主岗部门，默认该员工日常操作数据的记录均归属于主岗部门。主要用于数据生产和数据权限管理。

修改用户

X

1 填写用户信息 > 2 产品许可证 > 3 设置角色权限 (选填)

用户名 *	yy4_dfwu
用户昵称 *	yy4
密码	不输入表示不修改原密码
邮箱	yy4_dfwu@163.com
手机号	13800138000
直属上级	请输入直属上级
主岗部门	销售部
兼岗部门	测试1 销售部
描述	S
人员岗位名称	
测试人员名称	WB
职能部门审核	<input checked="" type="checkbox"/>

企业外部用户

最近更新时间：2023-03-02 14:24:53

企业外部用户泛指终端消费者，可以使用自定义应用。

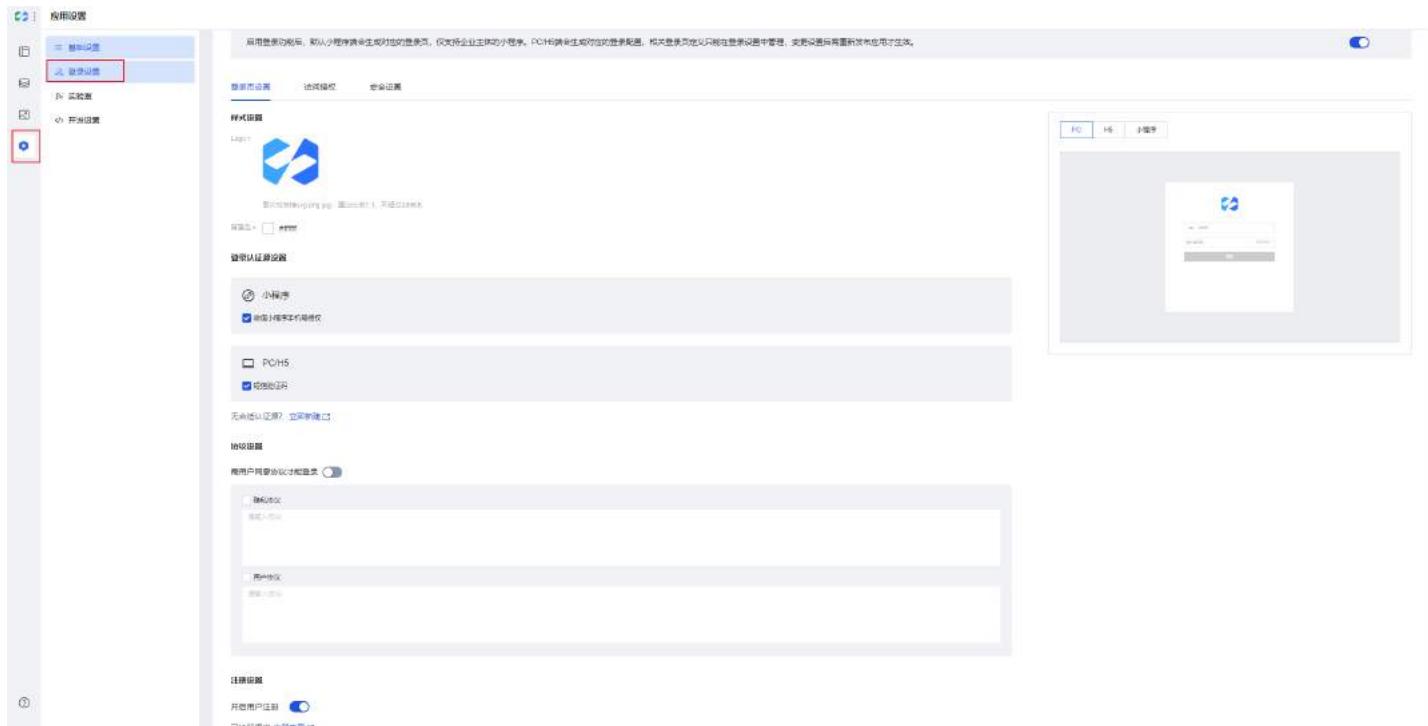
例如：为了满足电商场景需求，开发者使用微搭创建并发布了两个应用，一个面向C端用户的购物小程序（自定义应用），一个面向内部运营人员的商品进销存管理系统（模型应用）。这种情况下，使用购物小程序购物的用户为外部用户，使用进销存管理系统的用户为内部用户。

② 说明：

目前企业外部用户，只支持手机号注册。

操作步骤

应用设置 > 登录设置 > 注册设置，开启用户注册以后，可以通过手机号验证码或微信小程序手机号自动注册。



注册以后可以在企业外部用户列表里进行管理。



User ID	用户名	用户名称	微信号	已关联角色	手机号	更新时间	操作
注册	-	-	13800138000	-	-	2022-10-13 12:10:06	关联角色 删除
注册	-	-	13800138001	-	-	2022-09-28 11:50:51	关联角色 删除

用户登录

自定义应用登录

小程序端登录

最近更新时间：2023-08-02 15:14:16

目前小程序支持微信小程序手机号授权登录。

进入自定义应用编辑器，在左侧导航栏找到应用设置 > 访问控制，选择登录后访问，在登录页设置选项卡中选择对应的登录方式，目前仅支持企业主体的小程序。

The screenshot shows the 'Application Settings' sidebar with '访问控制' (Access Control) selected. In the main area, under '访问控制' (Access Control), the '登录后访问' (Login After Access) option is chosen. Below this, the '登录页设置' (Login Page Settings) tab is active, showing sections for '样式设置' (Style Settings) with a logo preview, '登录认证源设置' (Login Authentication Source Settings) with '小程序' (Mini Program) and '微信小程序手机号授权' (WeChat Mini Program Phone Number Authorization) selected, and 'PC/H5' settings with '账户密码登录' (Account Password Login) and '短信验证码' (SMS Verification Code) checked. At the bottom, there's a '协议设置' (Protocol Settings) section with a toggle for '需用户同意协议才能登录' (User must agree to the protocol to log in) and a note about privacy and user agreements.

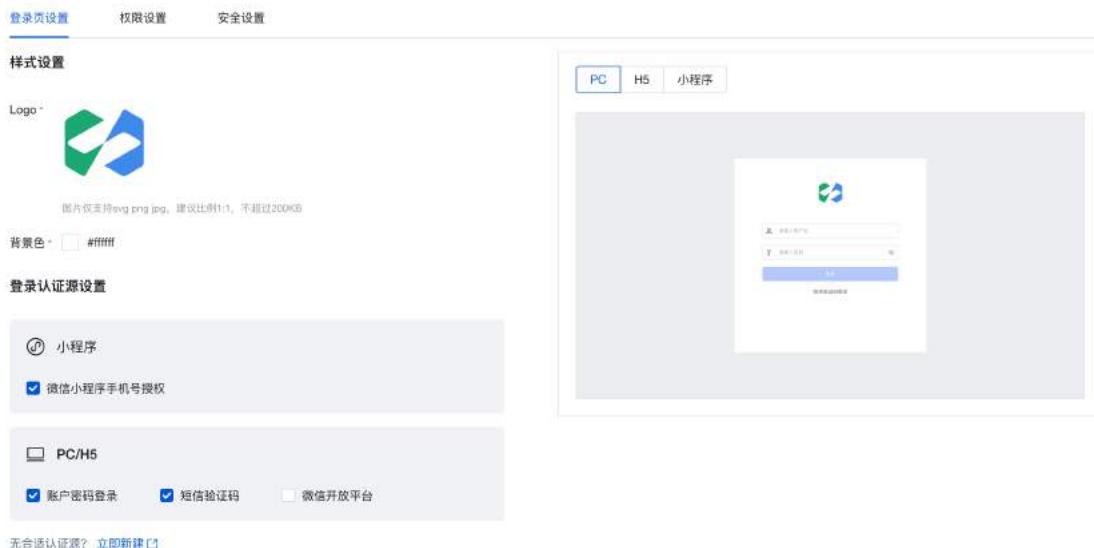
PC 或 H5 端登录

最近更新时间：2023-08-02 15:14:21

目前支持账号密码登录、短信验证码和微信开放平台登录。也可以自定义第三方认证源登录，请参见 [身份源管理](#)。

目前小程序支持微信小程序手机号授权登录。

自定义应用编辑器 > 访问控制 > 登录认证源设置，里面可以选择对应的登录方式，目前仅支持企业主体的小程序。



The screenshot shows the 'Login Source Settings' section of the Tencent Cloud Low-Code Platform. It includes:

- Logo:** A placeholder for a logo, with instructions: "图片仅支持svg png jpg, 建议比例1:1, 不超过200KB".
- Background Color:** A color picker set to #f0f0f0.
- Login Source Settings:**
 - 小程序:** Selected (radio button is checked).
 - 微信小程序手机号授权:** Checked (checkbox is checked).
 - PC/H5:** Not selected (radio button is not checked).
 - 账户密码登录:** Checked (checkbox is checked).
 - 短信验证码:** Checked (checkbox is checked).
 - 微信开放平台:** Not checked (checkbox is not checked).
- Bottom Buttons:** "无合适认证源?" and "立即新建" (Create New).

权限设置

最近更新时间：2023-08-02 15:14:25

开启登录访问规则后，需要用户登录以后才可以访问应用。

可以在权限设置中，设置可访问此应用的角色。同时还能指定该角色可访问的页面。

访问控制

- 应用访问规则 公开访问
适用：企业官网、公开信息门户等
- 登录后访问
适用场景：内部应用、管理后台等

登录页设置 [权限设置](#) 安全设置

角色权限设置

设置各个角色的访问策略

[应用至正式环境](#)

搜索

角色授权

默认超级管理员

默认外部用户

测试角色

体验角色

页面访问权限

可访问全部页面 可访问指定页面

首页 (index)

空白页 (utk6qajn3lo)

空白页-副本 (w7Bhgoob26g)

保存

保存后需要应用到正式环境，才可在正式环境生效

安全设置

最近更新时间：2023-08-02 15:14:30

保持自定义应用登录状态的过期时间（默认为30天），过期后用户需要重新登录，修改登录过期时间设置要10分钟后才会生效。

访问控制

应用访问规则 公开访问
适用场景：企业官网、公开信息门户等

登录后访问
适用场景：内部应用、管理后台等

登录页设置 权限设置 安全设置

自定义过期时间

30 天

保持自定义应用登录状态的过期时间（默认为30天），过期后用户需要重新登录。修改登录过期时间设置要10分钟后才会生效。

保存

企业工作台登录

最近更新时间：2023-08-02 15:14:34

企业工作台登录以后，默认可以访问所有发布后的模型应用。

企业工作台进入系统设置 > 登录配置页面，可以定义企业工作台的登录页样式和登录方式。样式设置、登录认证源设置、安全设置修改保存后要10分钟后才会生效。

登录方式

支持默认登录页和自定义登录页。本文将指导如何配置默认登录页。详情请参见 [自定义企业工作台登录页](#)。



The screenshot shows the 'Login Configuration' page with the 'Default Login Page' tab selected. It includes fields for 'Page URL' (with examples like 'lowcode.cloudbaseapp.com/app-xxxxxx/production'), 'Session Timeout' (set to 30 days), and a 'Save' button.

样式设置

支持定义标题、logo、背景色、背景图片、登录位置、版权信息等值。



The screenshot shows the 'Style Settings' page with various configuration options:

- Title: 腾讯云微搭低代码
- Logo: A green and blue logo icon.
- Background Color: #fff
- Login Position: Centered (center)
- Show Operation Document: Enabled
- Copyright Information: 2020-2023 腾讯云微搭低代码
- Background Image: A blue abstract background image.

登录认证源设置

选中对应的认证源以后，登录页可以使用对应的登录认证方式进行用户身份鉴权。

默认支持账号密码和企业微信登录。也可以自定义第三方认证源登录，请参见 [身份源管理](#)。

登录认证源设置



登录页样式实时显示在右侧，以下是 PC 和 H5 的预览效果：

PC 的预览效果：
H5 的预览效果：
安全设置

保持自定义应用登录状态的过期时间（默认为30天），过期后用户需要重新登录，修改登录过期时间设置要10分钟后才会生效。

安全设置

登录过期时间设置 *

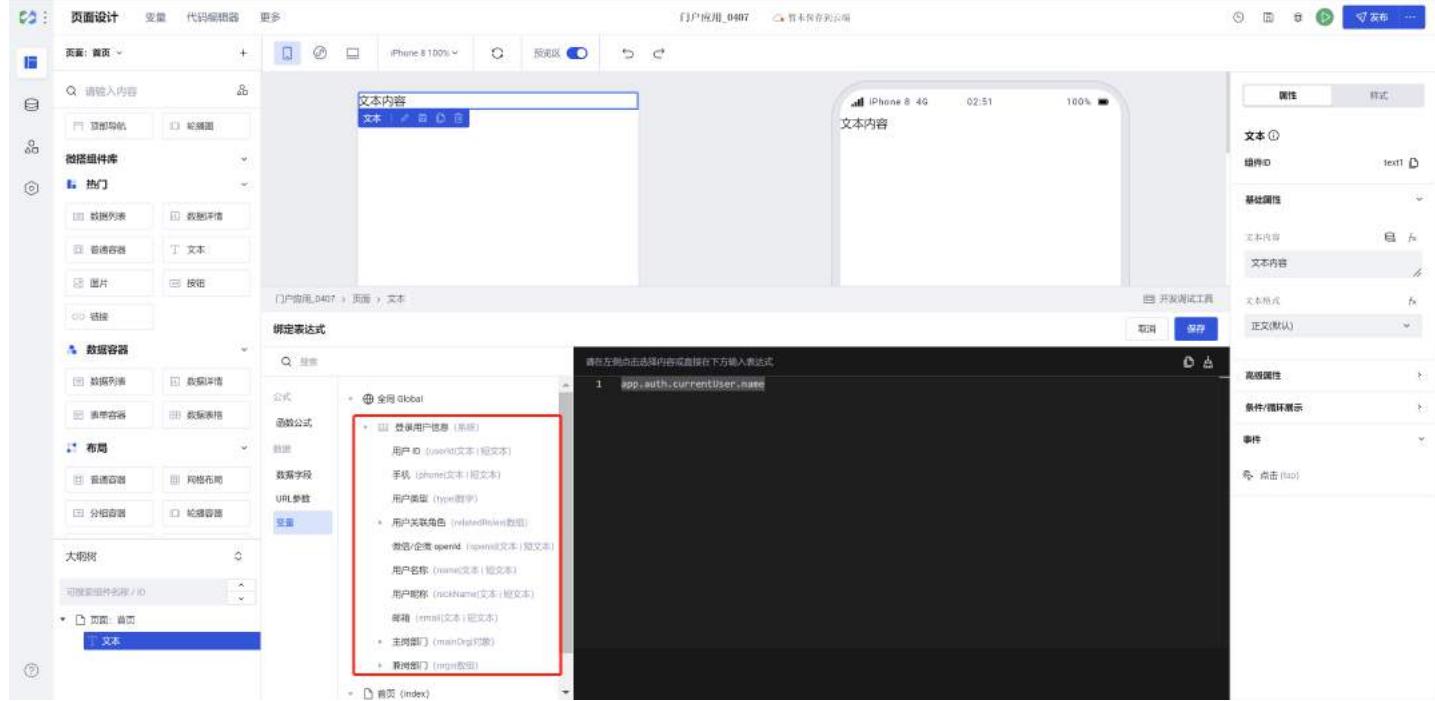
30 天

保持自定义应用登录状态的过期时间(默认为30天),过期后用户需要重新登录

获取当前登录用户信息

最近更新时间：2023-04-12 10:57:51

获取当前登录的用户信息，可以通过全局的系统变量 > 登录用户信息来直接获取。



角色与权限

预制角色

最近更新时间：2022-11-01 14:56:08

预制了“默认访客”、“默认外部用户”和“默认超级管理员”的角色，默认对于初次使用用户权限的开发者或者企业管理员，可以默认关联“默认访客”和“默认外部用户”角色使用自定义应用，关联“默认超级管理员”角色使用模型应用。所有预制角色均不支持修改。

- “默认访客”角色默认给自定义应用匿名访问时使用，如不满足可以重新定义新的角色。默认仅可以访问所有的普通应用和对应的页面，数据源所对应的业务数据可以查看全部、修改全部，可以访问所有数据源方法。
- “默认外部用户”角色默认给自定义应用实名登录时使用，如不满足可以重新定义新的角色。默认仅可以访问所有的普通应用和对应的页面，数据源所对应的业务数据可以查看全部、修改全部，可以访问所有数据源方法。
- “默认超级管理员”默认给模型应用使用，如不满足可以重新定义新的角色。默认仅可以访问所有的模型应用和对应的页面，数据源所对应的业务数据可以查看全部、修改全部，可以访问所有流程，可以访问企业工作台和进行相关的设置

角色与权限

微搭交流群 官方公众号 业务咨询 页面教学

The screenshot shows a table listing eight predefined roles. The columns are: Name, Identifier, Publish Status, Description, and Operations. The roles listed are: 质试官 (Identifier: msg, Published), 质试人员 (Identifier: mary, Published), 部门HR (Identifier: bmhr, Published), 普通员工 (Identifier: user, Published), 默认访客 (Identifier: anonymousUser, Published, Description: Default visitor role), 默认超级管理员 (Identifier: administrator, Published, Description: Default internal user role), and 默认外部用户 (Identifier: externalUser, Published, Description: Default external user role). Each row has an 'Add User' and 'More' button under the operations column.

名称	标识	发布状态	描述	操作
质试官	msg	已发布	-	添加用户 更多
质试人员	mary	已发布	-	添加用户 更多
部门HR	bmhr	已发布	部门HR	添加用户 更多
普通员工	user	已发布	-	添加用户 更多
默认访客①	anonymousUser	已发布	默认访客角色	添加用户 更多
默认超级管理员①	administrator	已发布	默认内部用户角色	添加用户 更多
默认外部用户①	externalUser	已发布	默认外部用户角色	添加用户 更多

新建角色

最近更新时间：2022-11-01 14:56:15

腾讯云控制台进入微搭低代码用户权限 > [角色权限](#) 页面，单击[新建角色](#)。

- 角色名称：用于用户授权时显示，不能重复。
- 角色标识：当前环境下唯一，不能重复。
- 角色描述：为非必填项，用于描述角色的一些信息。

新建角色 ×

角色名称 *

角色标识 *

描述

确定 取消

编辑角色

最近更新时间：2022-11-01 14:56:20

角色权限

帮助文档

角色名称	角色标识	发布状态	角色描述	操作
普通员工	user	未发布		添加用户 更多
匿名访客①	anonymousUser	已发布	匿名访客	添加用户 编辑角色
超级管理员①	administrator	已发布	超级管理员	添加用户 更多
共 1 条 (不包括默认角色)				
10 条 / 页 1 / 1 页				

- 角色名称：可以修改，用于用户授权时显示，不能重复。
- 角色标识：一旦定义以后不能修改。
- 角色描述：可以修改，为非必填项，用于描述角色的一些信息。

修改角色

X

角色名称 *

角色标识 *

描述

请输入角色的描述，非必填项，最多支持255个字符

删除角色

最近更新时间：2022-11-01 14:56:28

删除角色

当角色不需要使用了，可以直接单击删除角色，如果角色已经被用户关联了，则不能进行删除，需要解除关联以后再进行删除。

角色权限

帮助文档

角色名称	角色标识	发布状态	角色描述	操作
普通员工	user	● 未发布		添加用户 更多 编辑角色
匿名访客①	anonymousUser	● 已发布	匿名访客	添加用户 编辑角色
超级管理员①	administrator	● 已发布	超级管理员	添加用户 更多
共 1 条 (不包括默认角色)				
10 / 1 页				

再次确认是否删除角色，若需要删除角色请单击确认。



确定删除角色 普通员工 吗？

删除角色后数据将不可恢复，请谨慎操作

确定

取消

查看并定义角色

应用访问权限

最近更新时间：2023-05-06 11:10:02

对于目前独立发布的自定义应用，只有用户关联的角色勾选了应用访问权限，登录的时候才可以进入到应用的首页，没有勾选的话，即便登录鉴权成功，也无法登入。

角色信息

角色名称 普通员工

角色标识

更新时间 2023-05-05 19:25:14

环境标识

角色权限

自定义应用访问权限	模型应用访问权限	数据模型权限	APIs接口权限	流程权限	企业工作台权限
搜索应用名称和应用标识 <input type="text"/> <input type="button" value="Q"/>					
应用名称	应用标识	更新时间	<input type="checkbox"/>	是否可以访问	应用中的页面权限
自定义应用_0425	0425	2023-05-05 19:25:16	<input type="checkbox"/>		页面权限设置 ①
历史数据_0425	0425	2023-05-05 19:25:16	<input type="checkbox"/>		页面权限设置 ①
自定义应用_0424	0424	2023-05-05 19:25:16	<input type="checkbox"/>		页面权限设置 ①
自定义应用_0412	0412	2023-05-05 19:25:16	<input type="checkbox"/>		页面权限设置 ①
自定义登录	0412	2023-05-05 19:25:16	<input type="checkbox"/>		页面权限设置 ①
企业工作台权限	0412	2023-05-05 19:25:16	<input type="checkbox"/>		页面权限设置 ①
自定义应用_0323	0323	2023-05-05 19:25:16	<input type="checkbox"/>		页面权限设置 ①
共 14 条			10 条 / 页 <input type="button" value="上一页"/> <input type="button" value="下一页"/> 1 / 2 页 <input type="button" value="尾页"/>		

成员管理

添加成员

请输入用户名搜索



对于发布以后的模型应用，登入企业工作台的用户只有具备应用的访问权限，单击应用才可以进入到应用的首页。

角色信息

角色名称 普通员工

角色标识

更新时间 2023-05-05 19:25:14

环境标识

角色权限

自定义应用访问权限	模型应用访问权限	数据模型权限	APIs接口权限	流程权限	企业工作台权限
搜索应用名称和应用标识					
应用名称	应用标识	更新时间	<input type="checkbox"/>	是否可以访问	应用中的页面权限
模型应用_0001	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置
模型应用_0002	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置
模型应用_0003	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置
模型应用_0004	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置
模型应用_0005	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置
模型应用_0006	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置
模型应用_0007	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置
模型应用_0008	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置
模型应用_0009	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置
模型应用_0010	modelApp	2023-05-05 19:29:47	<input type="checkbox"/>		页面权限设置

页面访问权限

最近更新时间：2023-05-06 11:06:21

当对应的角色具备对应的应用访问权限以后，才需要定义页面访问权限。如果对应的角色没有勾选应用的访问权限，则不能设置页面的访问权限。默认勾选应用访问权限以后，会把应用下所有页面勾选。

角色信息

角色名称 普通员工
角色标识 ptyg
更新时间 2023-05-06 19:25:14
环境标识 [查看更多](#)

角色权限

自定义应用访问权限	模型应用访问权限	数据模型权限	APIs接口权限	流程权限	企业工作台权限
<input type="text" value="搜索应用名称和应用标识"/> Q					
应用名称	应用标识	更新时间	<input checked="" type="checkbox"/> 是否可以访问	应用中的页面权限	
模型应用_0425		2023-05-06 19:39:47	<input checked="" type="checkbox"/>	页面权限设置	
模型应用_0423		2023-05-06 19:39:47	<input checked="" type="checkbox"/>	页面权限设置	
...		2023-05-06 19:39:47	<input checked="" type="checkbox"/>	页面权限设置	
餐饮公司官网_0327-管理平台		2023-05-06 19:39:47	<input checked="" type="checkbox"/>	页面权限设置	
模型应用_0323		2023-05-06 19:39:47	<input checked="" type="checkbox"/>	页面权限设置	
模型应用_0320		2023-05-06 19:39:47	<input checked="" type="checkbox"/>	页面权限设置	
...		2023-05-06 19:39:47	<input checked="" type="checkbox"/>	页面权限设置	
模型应用_1216		2023-05-06 19:39:47	<input checked="" type="checkbox"/>	页面权限设置	

共 8 条 10 条 / 共 1 / 1 页

确认好权限后，单击确定。

设定应用“CRM销售管理系统”下的页面权限

页面名称	<input checked="" type="checkbox"/> 是否可以访问
detail_lvxmp6h_list 订单明细列表	<input checked="" type="checkbox"/>
detail_lvxmp6h_create 创建订单明细	<input checked="" type="checkbox"/>
detail_lvxmp6h_update 更新订单明细	<input checked="" type="checkbox"/>
detail_lvxmp6h_detail 订单明细详情	<input checked="" type="checkbox"/>
order_8droj4n_list 订单列表	<input checked="" type="checkbox"/>
order_8droj4n_create	<input type="checkbox"/>

确定 取消

说明：

用户关联的角色没有勾选可以访问的页面，运行态访问未勾选的页面时，展示空白页面，并气泡提示：“当前页面没有访问权限！”。

数据模型权限

最近更新时间：2023-02-24 15:59:49

数据模型行权限

数据源目前仅支持定义业务数据的权限，支持定义查看本人、修改本人，查看本人及下属、修改本人及下属，查看本部门及下级部门，修改本部门及下级部门，查看全部、修改全部，以及自定义的数据访问范围。

行权限设置

数据模型名称 家政订单列表
数据模型标识 order_shjbjy

行查看权限 ①

无权限 查看本人 查看本人及下属 查看本部门及下级部门 查看全部 自定义

行修改权限 ①

无权限 修改本人 修改本人及下属 修改本部门及下级部门 修改全部 自定义

确定 **取消**

可以通过自定义条件设置对应的数据行权限范围。

行权限设置

数据模型名称 家政订单列表
数据模型标识

行查看权限 ①

无权限 查看本人 查看本人及下属 查看本部门及下级部门 查看全部 自定义

条件1 单价 常量 且

行修改权限 ①

无权限 修改本人 修改本人及下属 修改本部门及下级部门 修改全部 自定义

确定 **取消**

数据模型列权限

案例

在学校教职工人力资源管理系统里，对应财务的 HR 可以看到所有教职工薪资、但是不能任意修改薪资。教职工之间无法互相看到薪资，但是可以看到姓名、年龄等基本信息。

场景一：创建一个普通用户及角色

用户为一个普通教职工，普通用户角色看不到工资字段。

步骤1：创建一个用户 joming_user

The screenshot shows the internal user management interface. On the left, there's a sidebar with department navigation. The main area is titled 'Product' and shows a list of users. One user, 'joming_user', is highlighted. The user details are: Name: joming_user, Department: Product, Role: 普通用户 (普通用户角色), Status: 已授权 (Granted), Last Login: 2022-08-25, and More... button.

步骤2：创建一个角色“普通用户角色”

The screenshot shows the role management interface. A new role is being created with the name '普通用户角色' and identifier 'ptyhjs'. The role is assigned to the '学校教职工' group and is marked as '已发布' (Published). There is also a '添加用户' (Add User) button.

步骤3：设置列权限

在角色权限设置中，普通用户角色下员工工资数据源的列权限中，将工资字段设置为无权限。

The screenshot shows the column permission settings dialog for the '普通用户角色' role. It lists various columns and their permissions. For the '工资' column, the permission is set to '无权限' (No Permission). Other columns like '姓名', '年龄', and '学历' have '仅查看' (View Only) permissions.

字段名称	字段标识	数据类型	权限设置
姓名	xm	文本 短文本	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改
工资	gz	数字	<input checked="" type="radio"/> 无权限 <input type="radio"/> 仅查看 <input type="radio"/> 查看并修改
年龄	ní	文本 短文本	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改
学历	xd	枚举	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改
数据标识	_id	文本 短文本	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改
创建时间	createdAt	日期时间	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改
更新时间	updatedAt	日期时间	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改
所有人	owner	关联关系	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改
创建人	createBy	关联关系	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改
修改人	updatedBy	关联关系	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改
所属部门	_departmentList	数组	<input type="radio"/> 无权限 <input checked="" type="radio"/> 仅查看 <input type="radio"/> 查看并修改

场景2：创建一个管理员及角色

用户 joming_admin 为一个 HR 同事，作为一个管理员可以看到所有字段，但是不能修改工资，修改工资由其他专门的 HR 负责。

步骤1：创建一个用户 joming_admin

The screenshot shows the internal user/department management interface. On the left, there's a sidebar with a search bar for department names, a '新建部门' (New Department) button, and a '导入用户' (Import User) button. Below these are filters for '全组织用户' (All Organization Users), '未关联部门用户' (Users Not Associated with a Department), and department categories like '研发' (Research), '产品' (Product), '餐饮部' (Catering Department), and 'HR'. The 'HR' category is currently selected, indicated by a blue highlight. On the right, a list of users is displayed with columns for '用户名' (User Name), '企业账号' (Enterprise Account), '已关联...' (Associated...), '手机号' (Mobile Number), '邮箱' (Email), '组...' (Group), '主部门' (Primary Department), '所属部门' (Department), '部门负责人' (Department Manager), '更新时间' (Last Update Time), and '操作' (Operations). A specific user entry 'joming_admin' is highlighted.

步骤2：创建一个管理员角色

The screenshot shows the role creation interface. It has tabs for '新建角色' (Create New Role) and '已发布' (Published). There are two entries: '管理员角色' (Administrator Role) with identifier 'glrs' and '普通用户角色' (Normal User Role) with identifier 'phyrs'. Both entries have a status of '已发布' (Published) and a '操作' (Operations) column with a '添加用户' (Add User) button.

步骤3：设置列权限

在角色权限设置中，管理员角色下员工工资数据源的列权限中，将工资字段设置为仅查看。

The screenshot shows the column permission settings interface for the 'Employee Salary' data source. It includes sections for '角色信息' (Role Information), '角色权限' (Role Permissions), '自定义应用访问权限' (Custom Application Access Permissions), '员工工资' (Employee Salary), '数据模型权限' (Data Model Permissions), and '成员管理' (Member Management). The main focus is the '列权限设置' (Column Permission Settings) dialog box, which lists columns like '姓名' (Name), '工资' (Salary), '年龄' (Age), etc., with permission options for '无权限' (No Permission), '仅查看' (View Only), and '查看并修改' (View and Modify). The '工资' (Salary) column is set to '仅查看' (View Only). At the bottom of the dialog are '确定' (Confirm) and '取消' (Cancel) buttons.

案例效果

步骤1：超级管理员登录后台录入员工及对应工资

姓名	工资	年龄	学历	创建时间	操作
张三	***	35	本科	2022-08-25 15:59:26	查看 编辑 删除
李四	***	40	硕士	2022-08-25 15:59:43	查看 编辑 删除
王五	***	46	博士	2022-08-25 16:00:06	查看 编辑 删除
小刘	***	20	本科	2022-08-25 16:00:30	查看 编辑 删除
后勤王三	***	38	专科及以下	2022-08-25 16:00:49	查看 编辑 删除

步骤2：用户 joming_user 查看员工工资数据

用户 joming_user 为一个普通教职工， joming_user 普通用户角色看不到工资字段，他访问工资列表时隐藏了员工工资列。

姓名	工资	年龄	学历	操作
张三	***	35	本科	查看 编辑 删除
李四	***	40	硕士	查看 编辑 删除
王五	***	46	博士	查看 编辑 删除
小刘	***	20	本科	查看 编辑 删除
后勤王三	***	38	专科及以下	查看 编辑 删除

步骤3：用户 joming_admin 查看员工工资数据

用户 joming_admin 为一个 HR 同事，可以看到所有字段，但是不能编辑工资字段值。

- 进入到列表页可以查看到工资金额。

姓名	工资	年龄	学历	操作
张三	10000	35	本科	查看 编辑 删除
李四	15000	40	硕士	查看 编辑 删除
王五	20000	46	博士	查看 编辑 删除
小刘	8000	20	本科	查看 编辑 删除
后勤王三	6000	38	专科及以下	查看 编辑 删除

- 进入到详情页可以查看到工资金额。

The screenshot shows a modal dialog titled "员工工资详情" (Employee Wage Details) over a list of employee wages. The modal contains the following data:

数据标识	058dfe1...178
创建时间	2022-08-25 15:59:26
更新时间	2022-08-25 15:59:26
修改人	administrator
创建人	administrator
修改人	administrator
所属部门	luketest_云计算
姓名	张三
工资	[Redacted]
年龄	35
学历	本科

At the top right of the modal are two buttons: "编辑" (Edit) and "取消" (Cancel).

编辑页面工资字段置灰，不能编辑。

The screenshot shows a modal dialog titled "更新员工工资" (Update Employee Wage) over a list of employee wages. The modal has four input fields that are grayed out:

姓名	张三
工资	[Redacted]
年龄	35
学历	本科

At the bottom left of the modal are two buttons: "取消" (Cancel) and "取消" (Cancel).

APIS 接口权限

最近更新时间：2023-01-10 09:13:56

用户对应的角色勾选 API 接口的访问权限以后，用户才能发起或调用 API 接口实现的逻辑，例如数据的增删查改，组件触发对应 API 接口时，如果具备此方法调用权限，则执行此方法；如果不具备此方法调用权限，则拦截此方法执行，并提示“当前用户无权限访问 API 接口，请联系管理员授权访问权限”。如果用户有 API 接口权限不满足的提示，开发者或企业管理员可以在 API 接口权限单击方法权限设置进行修改，入口如下图：

角色权限							
自定义应用访问权限	模型应用访问权限	数据模型权限	APIS接口权限	流程权限	企业工作台权限		
搜索APIs的名称和标识							
名称	标识	所属应用	发布时间	数据权限设置			
工作流API	zdy_i73vz0w	-	2022-11-01 19:11:30	方法权限设置			
疫情信息智能上报	jkbewmoorsb_tjpoasn	疫情信息智能上报_小程序_1018	2022-10-18 13:34:25	方法权限设置			
自定义	zdy_54ufmxt	-	-	方法权限设置			
企业微信	qywx_bj3vvfg	-	2022-10-09 18:02:02	方法权限设置			
空白示例	kbsl_uxfzs2n	-	-	方法权限设置			
腾讯云图像识别	txytbsb_4w6jhpa	健康码上报	2022-09-20 16:46:50	方法权限设置			
共 6 条							
10 条 / 页							

可以选中此角色允许访问的方法。

The screenshot shows the 'Role Permissions' section of the platform. A modal window titled 'Configure Method Permissions for the "WeChat Pay Function Model"' is open, listing methods such as '统一订单' (Unified Order), '查询订单' (Query Order), '关闭订单' (Close Order), and '下载对账单' (Download Settlement Statement). Each method has a checkbox labeled '是否可以使用' (Can be used) which is checked for '统一订单' and '关闭订单', and unchecked for '查询订单' and '下载对账单'. At the bottom of the modal are '确定' (Confirm) and '取消' (Cancel) buttons.

流程访问权限

最近更新时间：2022-11-01 14:57:41

流程访问权限

用户对应的角色勾选流程访问权限以后，用户才能发起或处理流程，未勾选的流程不能发起或处理。

[普通员工](#)

[帮助文档](#)

角色信息

角色名称：普通员工
角色标识：user
更新时间：2021-09-18 11:39:46
环境标识：[REDACTED]

角色权限

普通应用访问权限 模型应用访问权限 数据源权限 流程权限 企业工作台权限

搜索流程名称和流程标识	流程名称	更新时间	是否可以访问
a	process_914	2021-09-18 11:54:23	<input checked="" type="checkbox"/>

共 1 条

10 条 / 页 1 / 1 页

成员管理

添加成员

用户名	用户类型	手机号	邮箱	更新时间	操作
暂无数据					

共 0 条

10 条 / 页 1 / 1 页

更多功能

企业工作台访问权限

最近更新时间：2023-05-06 10:39:42

进入企业工作台需要验证账号密码、以及判断用户关联的角色“是否可登入企业工作台”的权限（避免 C 端自定义应用的用户验证密码成功后，误入工作台）。工作台的系统配置入口，通过开关的策略控制，对应用用户关联的角色勾选以后，才可以进入系统配置页面。

角色信息

角色名称 普通员工

角色标识 ptyg

更新时间 2023-05-05 19:25:14

环境标识 

角色权限

自定义应用访问权限 模型应用访问权限 数据模型权限 APIs接口权限 流程权限 **企业工作台权限**

是否可登入企业工作台 ⓘ

企业工作台系统配置入口

成员管理

添加成员

请输入用户名搜索 

用户名	用户类型	手机号	邮箱	更新时间	操作
-----	------	-----	----	------	----

暂无数据

共 0 条

10 条 / 页

 1 / 1 

发布角色

最近更新时间：2022-11-01 14:57:56

腾讯云控制台进入微搭低代码管理中心 > 用户权限 > 角色与权限页面，单击发布角色。

角色权限

The screenshot shows a table of published roles. The columns are: 角色名称 (Role Name), 角色标识 (Role Identifier), 发布状态 (Published Status), 角色描述 (Role Description), and 操作 (Operations). There are three rows of data:

角色名称	角色标识	发布状态	角色描述	操作
普通员工	user	已发布		添加用户 更多
匿名访客	anonymousUser	已发布	匿名访客	添加用户 更多
超级管理员	administrator	已发布	超级管理员	添加用户 更多

共 1 条 (不包括默认角色)

操作栏包括：帮助文档、发布角色成功、取消、搜索框、筛选框、分页器（1/1页）。

② 说明：

当角色的定义有变更， 默认体验环境直接生效。但是发布的正式环境默认不生效， 只有单击发布角色， 提示“发布成功”以后， 发布的正式环境才生效。对应的角色会有发布状态的显示， 只有已发布的角色， 正式环境才会生效。

运行态使用角色权限

最近更新时间：2023-05-06 10:37:38

运行态可以使用已发布的角色权限，权限的定义和修改只能在设计态操作，定义完成以后随应用发布到运行态使用。

The screenshot shows the 'Role & Permission' list in the 'Application Catalog' section of the platform. The table lists various roles with their names, identifiers, source types, and descriptions. Key entries include:

名称	标识	角色来源	描述
普通员工	ptyg	自建角色	-
局校-多角色	jxds	自建角色	-
局校-专用角色	jczyjs	自建角色	-
企业微信自定义应用管理角色	qywxzdyjs	自建角色	-
nft	nft	自建角色	-
测试1231	cs123	自建角色	-
家长	jz	自建角色	-
设备巡检管理员	sbxjgly	自建角色	-
数据保障组	wzbzz	自建角色	-
教师	teacher	自建角色	-
默认访客①	anonymousUser	自建角色	默认访客角色
默认超级管理员②	administrator	自建角色	默认内部用户角色
默认外部用户③	externalUser	自建角色	默认外部用户角色

仅可以查看角色下权限的定义情况，修改需要在设计态重新发布角色才可以。

The screenshot shows the detailed view of the '普通员工' role in the 'Role & Permission' section. It includes the role's name, identifier, update time, and environment association.

角色信息

- 角色名称: 普通员工
- 角色标识: ptyg
- 更新时间: 2023-05-06 19:44:45
- 环境标识: 企业微信-自定义应用-0424

角色权限

自定义应用访问权限	模型应用访问权限	数据模型权限	APIs接口权限	流程权限	企业工作台权限
企业微信自定义应用权限	应用标识	更新时间	最近可以访问	页面中的页面权限	
danic-局校	d1mf87pv	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	
自定义应用_0425	egkufq79	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	
feredli-历史数据_0425	1mq5cw8p	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	
自定义应用_0424	v945g05n	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	
自定义应用_0412	0k7dtewm	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	
feredli-apis	u79x56pm	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	
自定义登录	90f8e655	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	
餐饮公司官网_0327	71e8gW7x	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	
企业微信自定义应用	qj7v57zk	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	
自定义应用_0323	lk9bccle	2023-05-06 19:47:06	最近可以访问	页面权限设置 ①	

权限组

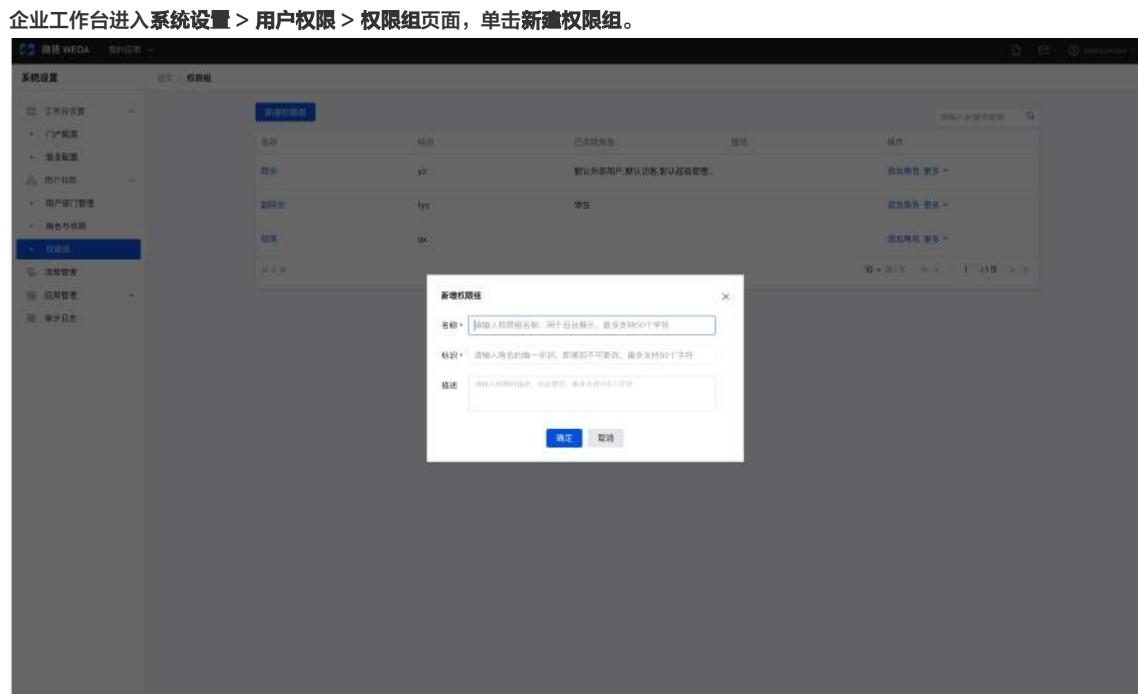
最近更新时间：2022-11-01 14:58:26

权限组可用作用户组或者角色组，也可以作为企业内部的岗位。本质上是角色权限的分组管理。

新建权限组

设计态和运行态目前均支持创建权限组。

腾讯云控制台进入微搭低代码 [用户权限](#) 页面单击新建权限组。



定义权限组

可以设置权限组里的角色和成员。

新建用户时可以选择关联对应的权限组。

修改用户

X

填写用户信息 > 产品许可证 > 3 设置角色权限 (选填)

请为新建的用户设置角色，此步骤可跳过，后续可在用户列表关联角色

请选择角色

无合适角色? [立即新建](#)

院长

无合适权限组? [立即新建](#)

[上一步](#) [保存](#)

身份源管理

概述

最近更新时间：2022-11-01 14:58:41

当企业或组织已有自己的账号体系，同时希望管理组织内成员可以直接使用微搭开发的应用资源。微搭提供适配单点登录 OAuth、SAML 2.0标准协议的集成配置能力，可以快速打通第三方用户身份，集成后可复用已有账号体系来登录使用对应的微搭应用，帮助企业用户通过一套用户账号即可访问所有内外部应用。

Oauth 协议

最近更新时间：2023-05-06 10:35:58

当企业或组织已有自己的账号体系，同时希望管理组织内成员可以直接使用微搭开发的应用资源。微搭提供适配单点登录 OAuth、SAML 2.0标准协议的集成配置能力，可以快速打通第三方用户身份，集成后可复用已有账号体系来登录使用对应的微搭应用，帮助企业用户通过一套用户账号即可访问所有内外部应用。本文通过微搭 SSO 集成操作并以 Github 为例对身份源管理进行讲解。

- 1 在微搭创建一个自定义应用
- 2 在 GitHub 开发平台创建一个 OAuth 应用
- 3 在微搭 > 身份源管理创建一个身份源，填入 GitHub OAuth 应用配置
- 4 在微搭创建的自定义应用的编辑器 > 登录设置里选中 Github 身份源
- 5 应用发布后使用

操作步骤

步骤1：创建自定义应用

1. 进入腾讯云微搭低代码控制台 [创建新应用](#) 页面，选择新建自定义应用。



2. 进入 [腾讯云](#) 页面，进入到应用开发 > 应用列表 > 应用设置 > 应用详情，单击应用详情进入详情页获取访问地址。



步骤2：在 GitHub 开发平台创建 OAuth 应用

1. 登录 GitHub 并进入 [Developer settings](#) 页面。单击 **Register a new application**。

The screenshot shows the GitHub Developer settings interface. On the left, there's a sidebar with 'GitHub Apps', 'OAuth Apps' (which is selected and highlighted in blue), and 'Personal access tokens'. The main area is titled 'No OAuth applications' with a sub-instruction: 'OAuth applications are used to access the GitHub API. Read the docs to find out more.' Below this is a green 'Register a new application' button, which is also highlighted with a red box.

△ 注意：

用户或组织最多可以拥有100个 OAuth 应用程序。

2. 填写相应基础信息后单击 **Register application**。

The screenshot shows the 'Register a new OAuth application' form. It includes fields for 'Application name' (with placeholder 'Something users will recognize and trust.'), 'Homepage URL' (with placeholder 'The full URL to your application homepage.'), 'Application description' (with placeholder 'Application description is optional.' and a note 'This is displayed to all users of your application.'), 'Authorization callback URL' (with placeholder 'Your application's callback URL. Read our OAuth documentation for more information.' and a note 'Allow this OAuth App to authorize users via the Device Flow. Read the Device Flow documentation for more information.'), and a 'Enable Device Flow' checkbox. At the bottom are 'Register application' and 'Cancel' buttons, with 'Register application' highlighted with a red box.

- Application name：可以填写中英文。
- Homepage URL：可在 [步骤1.2](#) 中的应用详情里获取。
- Authorization callback URL：微搭会提供回调地址，在 [步骤3.4](#) 的认证源基础配置里复制回调 URL，粘贴至上方输入框。

△ 注意

GitHub 里创建 OAuth 应用保存，需要完成 [步骤3.4](#) 的认证源基础配置才能完成。

步骤3：创建身份源并配置 GitHub OAuth 应用

1. 进入用户权限 > 身份源管理 页面。

2. 单击新建认证源，选择认证源为 OAuth 2.0 并单击下一步。

3. 填写认证源名称及 logo 等相关信息，单击下一步。

4. 认证源基础配置：

[编辑认证源](#)

选择认证源协议类型 > 定义认证源名称及logo > 3 认证源基础配置 > 4 账号关联

回调URL ① https://lowcode-... .tcloudbaseapp.com/_auth/

Client ID * 89

Client Secret * 79a

授权 URL ① * https://github.com/login/oauth/authorize

Token URL ① * https://github.com/login/oauth/access_token

获取token端点验证方式 ① * CLIENT_SECRET_BASIC

用户信息URL * https://api.github.com/user

Scope 授权范围 ① 请输入, 使用空格分隔不同的授权域

退出URL 请输入退出URL

[上一步](#) [下一步](#)

i. 首先复制回调 URL 到上面 [步骤2.2](#) 的 Authorization callback URL 里, 然后保存, 会生成对应的 Client ID 和 Client secrets。

Authorization callback URL *

Your application's callback URL. Read our [OAuth documentation](#) for more information.

ii. 复制 GitHub 的 Client ID 和 Client secrets 到认证源基础配置页面中, 填写相关信息后单击下一步。

- 授权 URL: <https://github.com/login/oauth/authorize>
- Token URL: https://github.com/login/oauth/access_token
- 用户信息 URL: <https://api.github.com/user>

说明:

更多详情请参见 [GitHub 授权 OAuth 应用程序操作指引](#)。

5. 账号关联目前的策略暂无匹配规则, 无需配置。

[新建认证源](#)

选择认证源协议类型 > 定义认证源名称及logo > 4 认证源基础配置 > 5 账号关联

用户信息映射

默认会按以下标准字典获取用户信息, 无则设置用户信息映射。如果IDP系统返回的用户信息字段和下列标准字段对不上, 则可以自定义对应的映射关系。

sub (用户ID)	遵循: 不限制字符长度且不能为空
name (用户名或昵称)	遵循: 不限制字符长度且不能为空
picture (用户头像)	遵循: 不限制字符长度且不能为空
username (用户名)	遵循: 不限制字符长度且不能为空
email (邮箱)	遵循: 不限制字符长度且不能为空
phone_number (手机号)	遵循: 不限制字符长度且不能为空
groups (用户的组)	遵循: 不限制字符长度且不能为空

登录模式

登录模式 * 仅登录模式
第三方账号只能登录, 不会保存在账户

[上一步](#) [完成](#)

6. 新建完成后可在身份源管理页面中查看。

身份源管理

身份源认证源				
认证源名称	认证源类型	状态	描述	操作
匿名登录	微信自有认证源	已启用	应用引用此登录方式后，用户无需登录即可访问使用	禁用
用户名密码登录	微信自有认证源	已启用	应用引用此方式后，可以通过用户名密码登录	禁用
短信验证码登录	微信自有认证源	已启用	应用引用此方式后，可以通过短信验证码登录	禁用
GitHub	oauth	已启用	...	禁用 更多

步骤4：在微搭自定义应用选择 GitHub 身份源

1. 进入 [微搭低代码](#) 页面，进入到应用开发 > 应用列表 > 应用设置 > 登录设置。

The screenshot shows the 'Login Settings' section under the 'Application Settings' tab. It includes a toggle switch for enabling login functionality, a logo upload area, and a list of authentication sources. The 'GitHub' option is checked under the 'PC/H5' section.

2. 进入登录页设置里勾选 GitHub。

The screenshot shows the 'Login Page Settings' section under the 'Application Settings' tab. It includes a toggle switch for enabling login, a logo upload area, and a list of authentication sources. The 'GitHub' option is checked under the 'PC/H5' section.

3. 访问授权里选择需登录后访问应用。

登录设置

启用登录功能后，默认小程序端会生成对应的登录页。仅支持企业主体的小程序。PC/H5端会生成对应的登录配置。相关登录页定义只能在登录设置中管理，变更设置后需重新发布应用才生效。

无需登录访问应用
访问应用url直接进入首页，允许匿名用户浏览

需登录后访问应用
访问应用url会自动跳转到登录页。登录成功后才可以进入应用首页，不允许匿名浏览

外部用户默认权限

无合适角色?

4. 安全设置为应用自动退出登录的时间，默认无需设置。

登录设置

启用登录功能后，默认小程序端会生成对应的登录页，仅支持企业主体的小程序。PC/H5端会生成对应的登录配置，相关登录页定义只能在登录设置中管理，变更设置后需重新发布应用才生效。

自定义过期时间
30 天
保持自定义应用登录状态的过期时间(默认为30天)。过期后用户需要重新登录，修改登录过期时间设置要10分钟后才会生效。

5. 单击保存后返回应用编辑器页面，单击右上角发布应用。

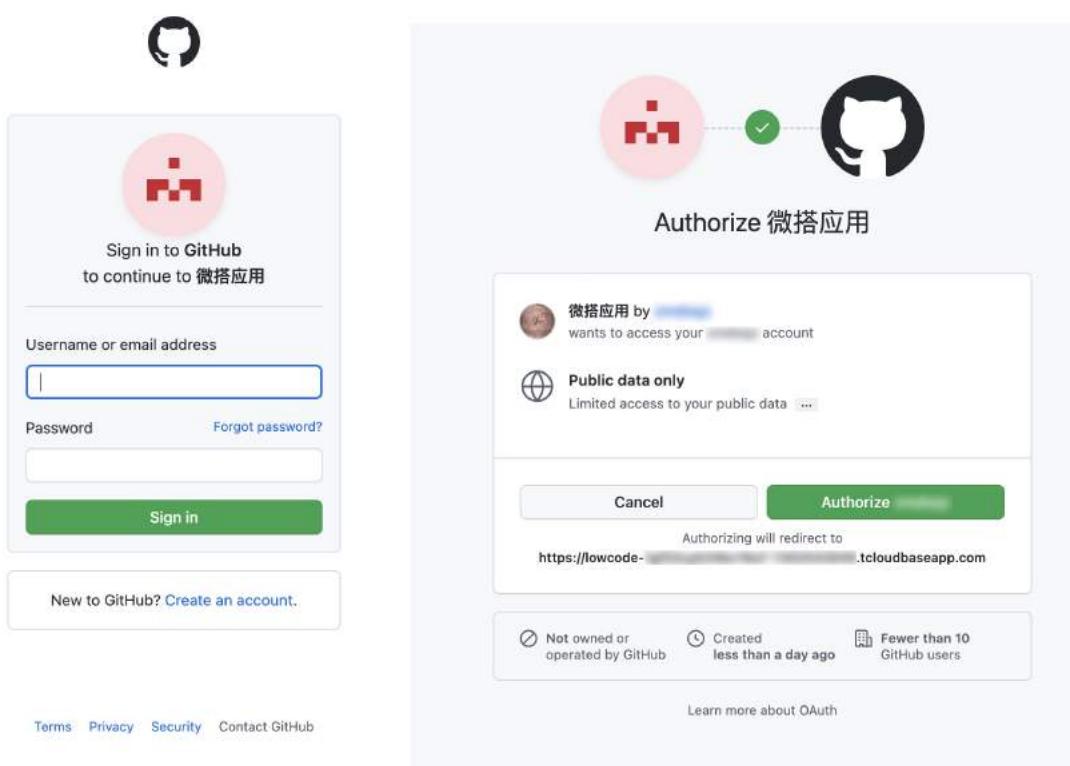
The screenshot shows the application editor interface for a project named '自定义应用_0410'. On the right, there's a '最新发布结果' (Latest Published Result) panel displaying a preview of the application's home page on both iPhone 8 (4.7") and PC (H5) platforms. The mobile preview shows a blurred version of the app's UI. Below the preview, there are tabs for 'iPhone 8 (4.7")' and 'PC (H5)'. At the top of the editor, there are various toolbars and a status bar indicating '已发布 (已更新)' (Published (Updated)).

步骤5：登录应用

1. 通过 [步骤1.2 中的访问链接](#)进入应用登录界面。



2. 填写好 GitHub 账号和密码进行登录认证。



3. 认证成功后即可访问应用。



进校防疫信息统计
为了做好疫情防控工作。
请如实填写各项信息。

· 学生信息

姓名

班级

地区

· 健康码

上传截图

[查看示例](#)

· 核酸检测记录
(最多上传3张图片, [识别]多次核酸检测记录)

上传截图

[查看示例](#)

· 行程卡

上传截图

Saml 协议

最近更新时间：2023-04-13 16:04:20

- 1 创建 saml 协议身份源
- 2 在 samltest 里上传微搭元数据链接
- 3 在编辑器 - 登录设置选中 samltest 身份源
- 4 应用发布后使用第三方认证源登录
- 5 复制第三方用户的角色 ID，创建角色

操作步骤

步骤1：创建 saml 协议身份源

登录 [微搭低代码](#) > 身份源管理 > 新建认证源，创建 saml 协议身份源。

1. 选择认证源协议为 saml。



2. 在微搭 > 身份源管理上传 samltest 元数据。先单击下载 [第三方元数据](#)。

The screenshot shows the 'Download Metadata' page for 'SAMLtest Metadata'. It includes a 'Table of Contents' with sections like '1. SAMLtest Metadata', '2. Shibboleth', '3. Complete Endpoints and Credentials', '3.1. SAMLtest's SP', '3.2. SAMLtest's IdP', and '4. Attributes Sent'. Below the table of contents, there is sample XML code for loading the metadata into a Shibboleth 3.x IdP.

```
<MetadataProvider id="SAMLtest"
    xsi:type="FileBackedHTTPMetadataProvider"
    backingFile="%{idp.home}/metadata/SAMLtest.xml"
    metadataURL="https://samltest.id/saml/sp">
    <!-- You should always check the signature and freshness of remote
        metadata. It's commented out until you get the basics working.
        <MetadataFilter xsi:type="SignatureValidation"
            certificateFile="%{idp.home}/credentials/signet.crt" />
        <MetadataFilter xsi:type="RequiredValidUntil" maxValidityInterval="P30D"/>
    -->
</MetadataProvider>
```

再下载 **SAMLTest IdP only** 文档后上传到微搭。

[新建认证源](#)



选择认证源协议类型 > 定义认证源名称及logo > 3 认证源基础配置 > 4 账号关联

身份提供商元数据文档 * idp 100% 重新上传 删除
查看图示

微搭的元数据URL https://lowcode-1gf52uyb34bc16a7.ap-shanghai.tcb-api.tencentcloudapi.com/auth/v1/saml/sp/metadata/smaltest

退出URL 请输入退出URL

上一步 下一步

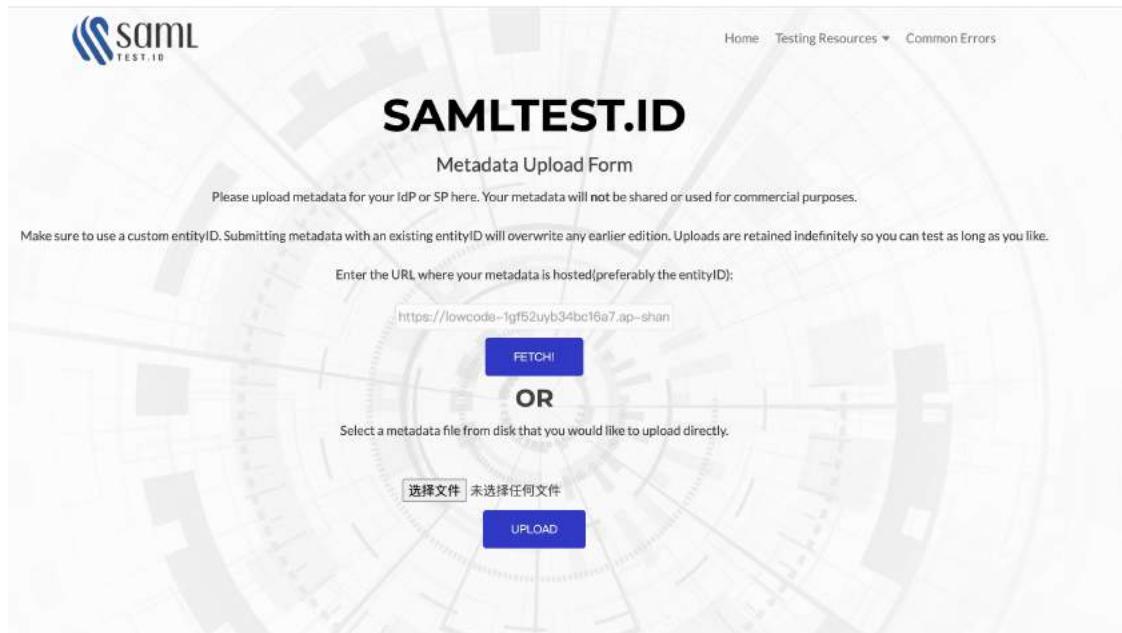
步骤2：在 samltest 里上传微搭元数据链接

在 samltest 里上传微搭元数据链接。

1. 复制微搭的元数据 URL。

https://lowcode-1gf52uyb34bc16a7.ap-shanghai.tcb-api.tencentcloudapi.com/auth/v1/saml/sp/metadata/smaltest

2. 进入到第三方 idp 里上传，URL 粘贴后直接单击 fetch。



Home Testing Resources Common Errors

SAMLTEST.ID

Metadata Upload Form

Please upload metadata for your IdP or SP here. Your metadata will not be shared or used for commercial purposes.

Make sure to use a custom entityID. Submitting metadata with an existing entityID will overwrite any earlier edition. Uploads are retained indefinitely so you can test as long as you like.

Enter the URL where your metadata is hosted(preferably the entityID):

https://lowcode-1gf52uyb34bc16a7.ap-shan

FETCH!

OR

Select a metadata file from disk that you would like to upload directly.

选择文件 未选择任何文件

UPLOAD

3. 上传完成。



The screenshot shows the "Metadata Upload Results" page from SAMLTEST.ID. The page title is "SAMLTEST.ID". Below it, a message says "SAMLtest successfully downloaded and saved your metadata from https://lowcode-1gf52uyb34bc16a7.ap-shanghai tcb-api.tencentcloudapi.com/auth/v1/saml/sp/metadata/smaltest. We now trust you." A note below states: "However, you still need to trust us. Please proceed to load SAMLtest's metadata in your own provider." There is a logo for SAMLTEST.ID and a download link: "Download XML (2008/09.xml)".

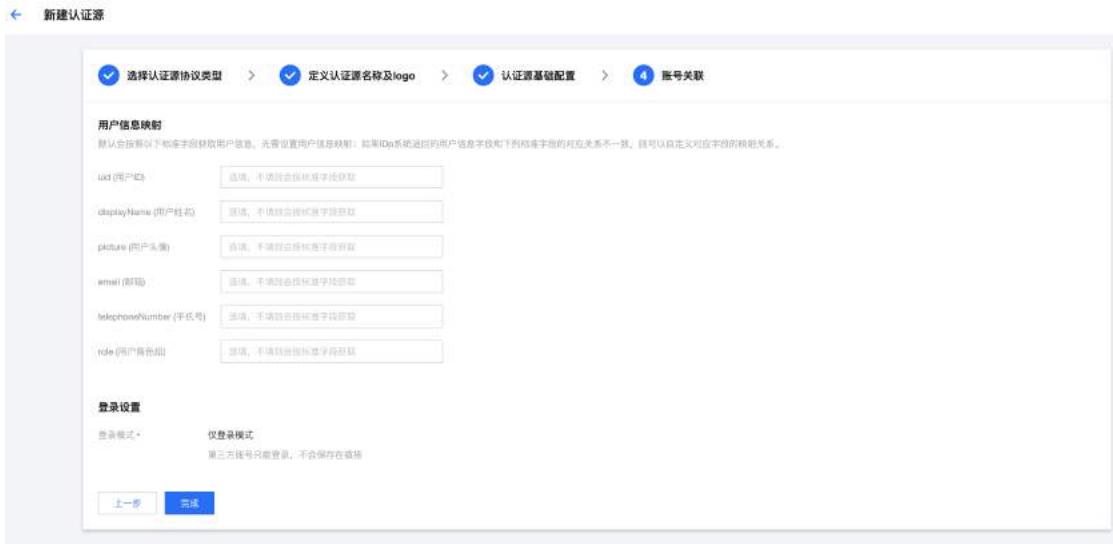
```

<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" validUntil="2022-06-15T12:10:19.977Z" entityID="urn:base:lowcode-1gf52uyb34bc16a7:smaltest">
<sPSSODescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" validUntil="2022-06-15T12:10:19.976888531Z" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  <AuthnRequestSigned>False</AuthnRequestSigned>
  <NonAssertionsSigned>True</NonAssertionsSigned>
  <KeyDescriptor use="encryption">
    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#dsa" />
  </KeyDescriptor>
  <X509Certificate>
    <!-- X509 certificate content -->
  </X509Certificate>
</sPSSODescriptor>
<AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://lowcode-1gf52uyb34bc16a7.ap-shanghai.tcb-api.tencentcloudapi.com/auth/v1/saml/sp/smaltest">
  <AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Index="1" />
  <AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact" Location="https://lowcode-1gf52uyb34bc16a7.ap-shanghai.tcb-api.tencentcloudapi.com/auth/v1/saml/sp/acs/smaltest" Index="2" />
</AssertionConsumerService>
</SPSSODescriptor>
</EntityDescriptor>

```

4. 账号关联。

账号关联目前的策略暂无匹配规则，无需配置。



The screenshot shows the "账号关联" (Account Association) configuration page. It has a breadcrumb navigation: "选择认证源协议类型" → "定义认证源名称及logo" → "认证源基础配置" → "账号关联".

用户信息映射

默认会按照以下标准匹配用户信息。若需变更用户信息映射，如果Idp系统返回的用户信息字段和下方标准字段的对应关系不一致，可以自定义对应字段的映射关系。

uid (用户ID)	必填。手机号会自动转换为MD5
displayName (用户名)	必填。不建议直接填写用户名
picture (用户头像)	必填。手机号转码后直接显示
email (邮箱)	必填。手机号会自动填写至邮箱
telephoneNumber (手机号)	必填。手机号会自动填写至手机号
role (用户角色组)	必填。手机号会自动填写至角色组

登录设置

登录模式：仅登录模式
第三方账号只能登录，不可保存在商场

上一步 完成

5. 创建完成。

身份源管理

新建认证源				
认证源名称	认证源类型	状态	描述	操作
匿名登录	微信自有认证源	已启用	应用引用此登录方式后，用户无需登录即可访问使用	停用
用户名密码登录	微信自有认证源	已启用	应用引用此方式后，可以通过用户名密码登录	停用
短信验证码登录	微信自有认证源	已启用	应用引用此方式后，可以通过短信验证码登录	停用
jomingsaml	saml	已启用	--	停用 更多 ▾
Github	oauth	已启用	--	停用 更多 ▾
ttgithub	oauth	已启用	--	停用 更多 ▾
authingtest	oauth	已启用	这个需要多行文本框	停用 更多 ▾
test	saml	已启用	--	停用 更多 ▾
tgithub	oauth	已启用	--	停用 更多 ▾
Google	oidc	已停用	--	启用 更多 ▾

步骤3：在编辑器 - 登录设置选中 samltest 身份源

在微搭创建一个自定义应用。在编辑器 > 登录设置里选中 samltest 身份源。

登录设置

登录页设置 访问授权 安全设置

登录认证源设置

小程序

微信小程序手机号授权

PC/H5

短信验证码 jomingsaml Github
 ttgithub authingtest test tgithub

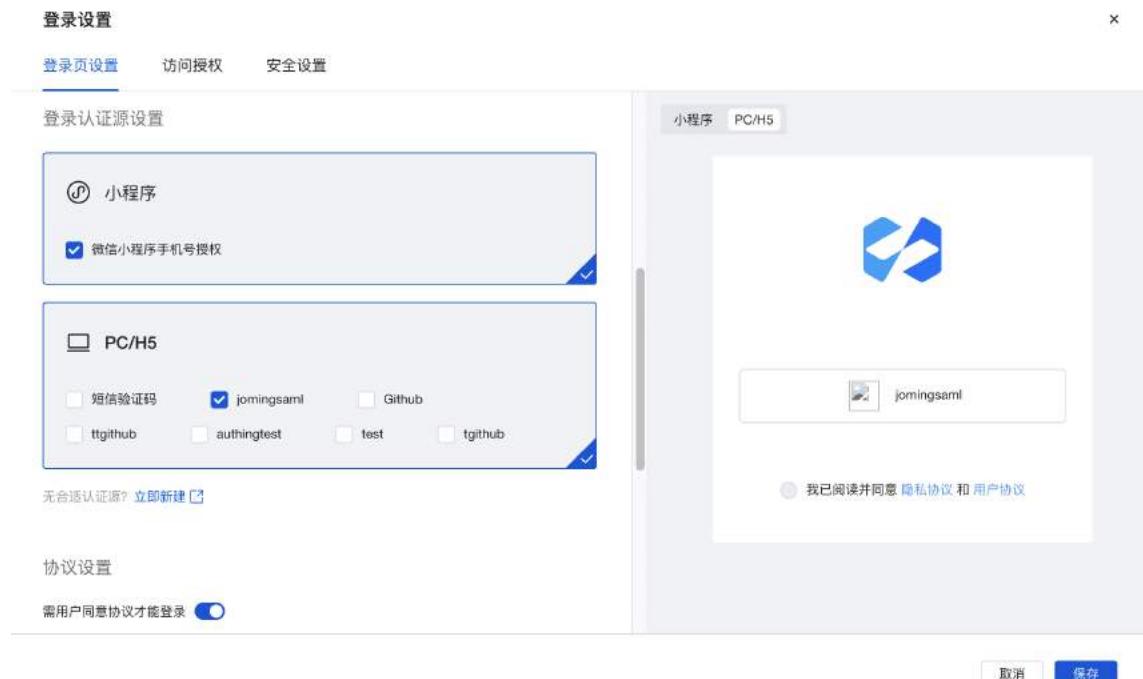
无合适认证源？立即新建

协议设置

需用户同意协议才能登录

我已阅读并同意 [隐私协议](#) 和 [用户协议](#)

取消 保存

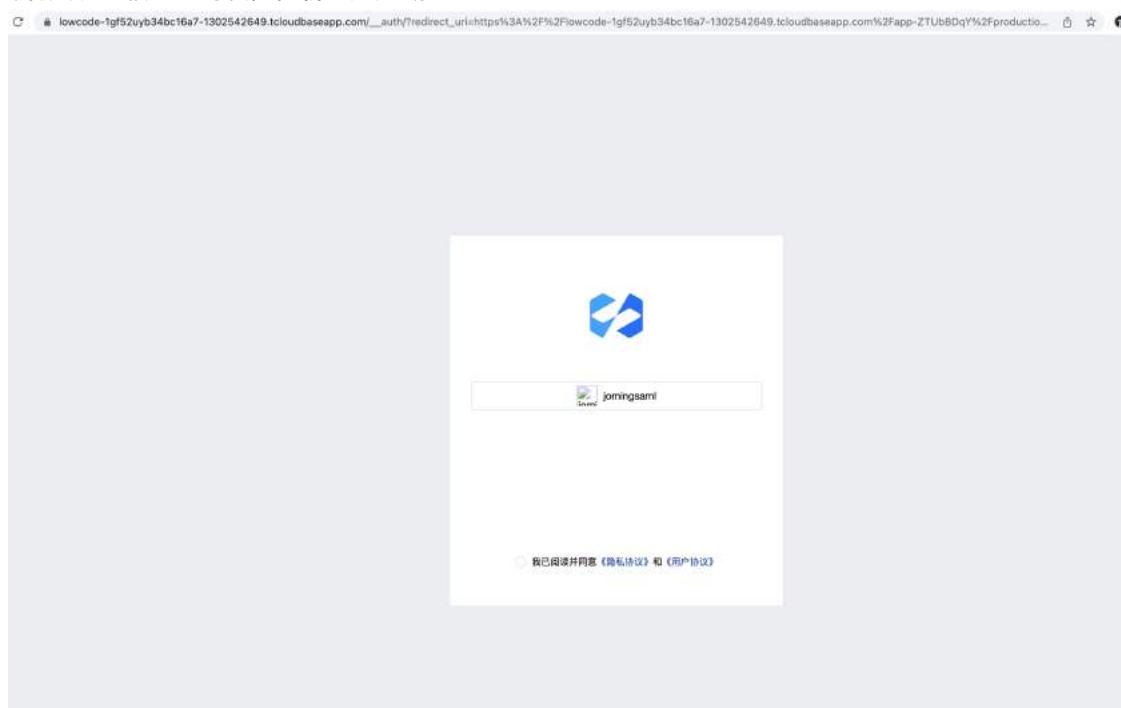


步骤4：应用发布后使用第三方认证源登录

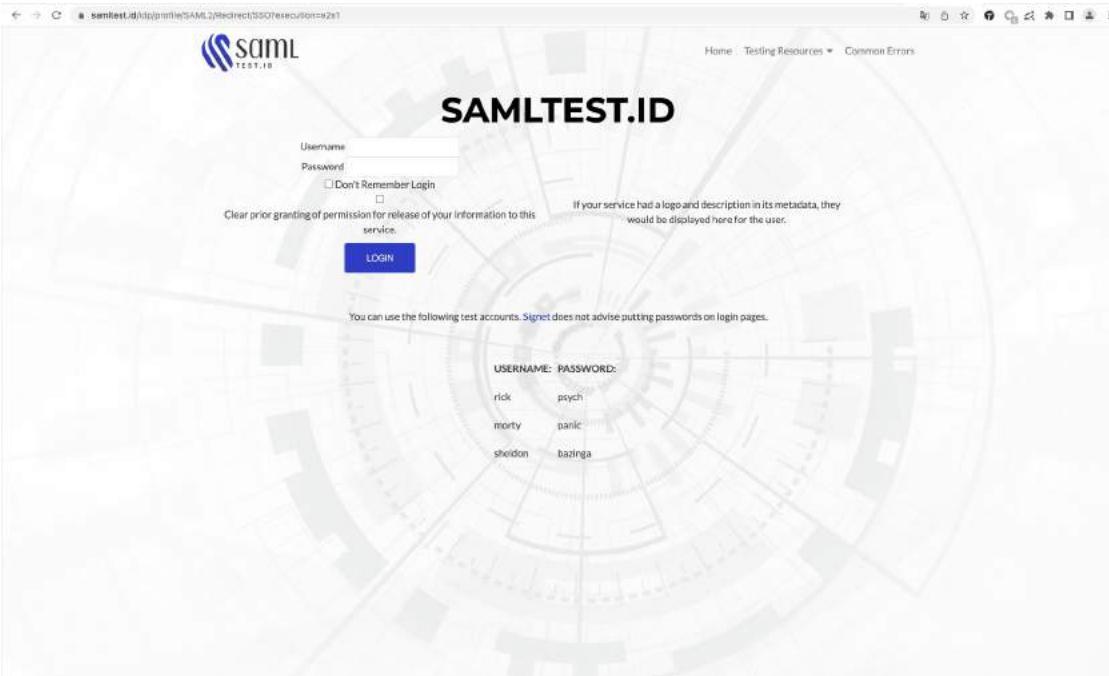
1. 发布应用。



2. 访问应用时直接进入登录页，单击第三方认证源。



3. 跳转到第三方 idp 登录页面。



Home · Testing Resources · Common Errors

SAMLTEST.ID

Username
Password
 Don't Remember Login

Clear prior granting of permission for release of your information to this service.

USERNAME:	PASSWORD:
rick	psych
morty	panic
sheldon	bazinga

LOGIN

4. 用测试账号登录。



Home · Testing Resources · Common Errors

SAMLTEST.ID

Username sheldon
Password *****
 Don't Remember Login

Clear prior granting of permission for release of your information to this service.

USERNAME:	PASSWORD:
rick	psych
morty	panic
sheldon	bazinga

LOGIN

5. 登录成功后，选择 Accept。

samitest.id/idp/profile/SAML2/Redirect/SSO;jsessionid=096344C7E8B7DDED7B9540DA036F66CB?execution=e1s2

You are about to access the service:
urn:base:lowcode-1gf52uyb34bc16a7:samitest

Information to be Provided to Service	
displayName	Sheldon
givenName	Sheldon
identifier	scooper@samltest.id
mail	scooper@samltest.id
role	employee@samltest.id
surname	Cooper
telephoneNumber	+1-555-555-5515
uid	sheldon

The information above would be shared with the service if you proceed. Do you agree to release this information to the service every time you access it?

Select an information release consent duration:

Ask me again at next login
 I agree to send my information this time.
 Ask me again if information to be provided to this service changes
 I agree that the same information will be sent automatically to this service in the future.
 Do not ask me again
 I agree that **all** of my information will be released to **any** service.

This setting can be revoked at any time with the checkbox on the login page.

步骤5（可选）：复制第三方用户的角色 ID，创建角色

复制第三方用户的角色 ID，创建角色。

1. 如果此处不设置，则默认按照登录设置里的统一角色访问应用。

登录设置

启用登录功能后，默认小程序端会生成对应的登录页。仅支持企业主体的小程序。PC/H5端会生成对应的登录配置。相关登录页定义只能在登录设置中管理，变更设置后需重新发布应用才生效。

[访问授权](#)

登录页设置 访问授权 安全设置

无需登录访问应用
访问应用url会直接进入首页，允许匿名用户浏览

需登录后访问应用
访问应用url会自动跳转到登录页。登录成功后才可以进应用首页。不允许匿名浏览

外部用户默认权限 默认外部用户

无合适角色？[立即新建](#)

2. 如果对应用用户想按照明确角色访问微搭应用，则需要定义微搭角色，复制第三方的角色 ID：

mailto:employee@samltest.id

You are about to access the service:
urn:base:lowcode-1gf52uyb34bc16a7:smaltest

Information to be Provided to Service

displayName	Sheldor
givenName	Sheldon
identifier	scooper@samltest.id
mail	scooper@samltest.id
role	employee@samltest.id
surname	Cooper
telephoneNumber	+1-555-555-5515
uid	sheldon

The information above would be shared with the service if you proceed. Do you agree to release this information to the service every time you access it?

Select an information release consent duration:

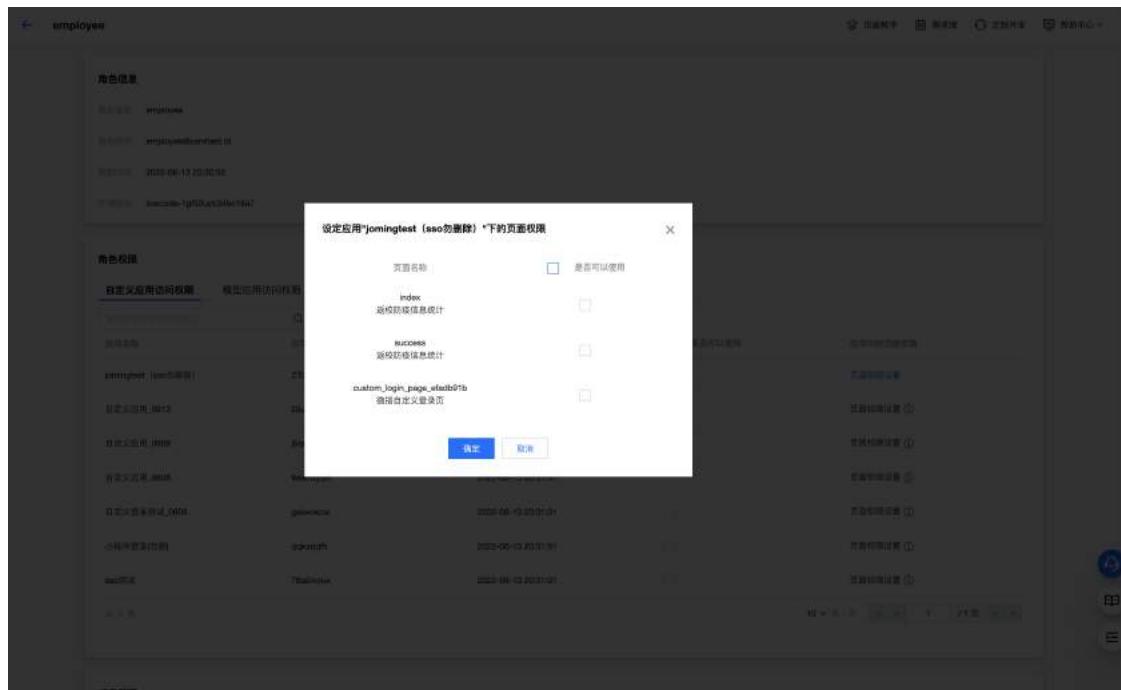
- Ask me again at next login
• I agree to send my information this time.
- Ask me again if information to be provided to this service changes
• I agree that the same information will be sent automatically to this service in the future.
- Do not ask me again
• I agree that **all** of my information will be released to **any** service.

This setting can be revoked at any time with the checkbox on the login page.

粘贴第三方的角色 ID 到微搭角色标识。



默认此角色没有访问权限。



登录成功后直接进入微搭应用首页。

返校防疫信息统计

为了做好新冠肺炎防控工作,
请佩戴好我们发放的健康码。

· 学生信息

姓名:

班级:

校区: 深圳粤康码

· 健康码

上传截图

查看示例

· 核酸检测记录

(最多上传3张图片, 识别多次核酸检测记录)

上传截图

查看示例

· 行程卡

上传截图

查看示例

可以看到登录后的第三方用户信息。

The screenshot shows a browser window with a login form for '返校防疫信息统计'. The form includes fields for '姓名' (Name), '班级' (Class), and '地区' (Region). Below the form is a modal with the message '接口无访问权限' (No access permission for interface). The browser's address bar shows the URL: `lowcode-1gf52uyb34bc18a7-1302642649.tcloudbaseapp.com/app-2TUbBDqYjproduction/index?state=prod-emailset-14cukf8ejc2`. The page title is 'Dimensions: Responsive' with dimensions 1782x410 and 100%. The status bar indicates 'No throttling'.

The main content area displays a cartoon illustration of a person wearing a mask and holding a smartphone.

The right side of the screenshot shows the Chrome DevTools Application tab open, specifically the Local Storage section. It lists items from the URL `https://lowcode-1gf52...`:

- `device_id`: Value: `f400053a7d94a6252053a0057247801a`
- `device_ACInfoABSLJUNGE0mTDY`: Value: `f59402074e6942ca0f0703a26e6fb8010`
- `AE9B_ID`: Value: `a736c6cf0428-4465-91b-3941a4e40003`
- `credentials_ACInfoABSLJUNGE0mTDY`: Value: `[{"token_type": "Bearer", "access_token": "eyJhbGciOiJIaTwkZGJlcyX2UjI202wLThyQ2JHGNhD25NWEw1kU02mf8fWZmOTJ2jO5..."}]`
- `user_info_lowcode-1gf52uyb34bc18a7`: Value: `{"version": "localCache", "content": "sub": "WhV9deDx34zcs2sQNPw", "name": "anonymous", "groups": [{"id": "anonymous"}, {"id": "WhV9de..."}]}`

The Application tab also shows other sections like Manifest, Session Storage, Web SQL, Cookies, and Interest Groups.

开发者权限

最近更新时间：2023-07-12 17:26:55

主账号可以根据实际开发场景，对子账号开发者设定对应的 cam 权限策略。

场景一：子账号拥有云账号下全部资源权限

默认创建的子账号的时候会自动带此策略 **AdministratorAccess**，可以操作云账号下所有资源。

腾讯云控制台进入访问管理 > 用户列表 > 快速新建用户。



默认创建的子账号的时候会自动带此策略 **AdministratorAccess**，可以操作云账号下所有资源。

A screenshot of the 'Quick Create User' dialog in the Tencent Cloud Control Panel. The left sidebar shows 'Access Management', 'User List' (selected), and other options like 'User Settings', 'User Groups', 'Roles', 'Identity Federation', and 'Identity Federation'. The main dialog has a blue header 'Quick Create User'. It contains a note about what 'Quick Create User' means. Below is a table for creating a new user: '用户名' (User Name) is set to 'test2', '访问方式' (Access Method) is '控制台登录' (Console Login), and '用户权限' (User Permissions) is 'AdministratorAccess'. A note says '单次最多创建10个用户' (Up to 10 users can be created at once). At the bottom, there's a checkbox for '用户必须在下次登录时重置密码' (User must reset password upon next login) and a 'Create User' button.

创建子账号完成。

[← 用户详情](#)

test2 子用户

权限策略

① 关联策略以获得策略包含的操作权限。解除策略将失去策略包含的操作权限。特别的，解除策略后该类型的策略是通过将用户从关联该策略的用户组中移出。

策略名	描述	关联类型	策略类型	关联时间	操作
AdministratorAccess	该策略允许您管理账户内所有用户及其权限、财务相关的信息、云服务资产。	直接关联	预设策略	2022-10-13 16:04:58	解除

权限边界 (未设置)

场景二：子账号拥有微搭、云开发所有环境下资源权限

以下设置完成以后，可以使用微搭和云开发所有环境下资源。

步骤1：进入到对应子账号详情里，移除所有策略

前提：创建子账号以后需要先移除默认添加的此策略 AdministratorAccess。

test1 子用户

权限策略

① 开始使用策略

此用户还没有任何权限。在开始时，请将用户添加到组，从其他用户复制权限或者直接附加策略。

[添加策略](#)

权限边界 (未设置)

① 开始使用权限边界

尚未为此用户设置任何权限边界。

[设置边界](#)

步骤2：添加以下cam策略，从策略列表中选取策略关联

然后子账号关联以下预设策略：

- 必须策略
 - QcloudCamSubaccountsAuthorizeRoleFullAccess

- QcloudAccessForTCBRole
- QcloudAccessForTCBRoleInAccessCloudBaseRun
- QcloudLowCodeFullAccess
- 可选策略
 - QcloudTCBFullAccess
 - QcloudTCBRFullAccess

1. 从策略列表中选取策略关联。

[← 添加策略](#)

This screenshot shows the 'Select Policy' step of a wizard. At the top, there are three tabs: 'Add New Policy', 'Reuse Existing Policy', and 'Select from Policy List'. The third tab is selected. Below the tabs is a note about selecting policies based on the current account's permissions. A search bar at the top right contains the text 'QcloudAccessForTCBRole'. The main area displays a table of policies:

Policy Name	Description	Reference Count	Policy Type
QcloudAccessForTCBRole	Cloud Development (TCB) has full access to resources.	4	Predefined Policy
QcloudAccessForTCBRoleInAccessCloudBaseRun	Cloud Development (TCB) has full access to resources in the AccessCloudBaseRun service.	4	Predefined Policy
QcloudLowCodeFullAccess	QCloud Low Code Full Access.	0	Predefined Policy

At the bottom left is a 'Next Step' button.

2. 搜索需要关联的策略名称，然后选中。

[← 添加策略](#)

This screenshot shows the 'Search and Select Policy' step of the wizard. It has the same tabs and note as the previous screen. The search bar now contains 'QcloudAccessForTCBRole'. The policy list table is identical to the one above:

Policy Name	Description	Reference Count	Policy Type
QcloudAccessForTCBRole	Cloud Development (TCB) has full access to resources.	4	Predefined Policy
QcloudAccessForTCBRoleInAccessCloudBaseRun	Cloud Development (TCB) has full access to resources in the AccessCloudBaseRun service.	4	Predefined Policy
QcloudLowCodeFullAccess	QCloud Low Code Full Access.	0	Predefined Policy

At the bottom left is a 'Next Step' button.

3. 审阅用户权限。

[← 添加策略](#)

This screenshot shows the 'Review User Permissions' step. It has the same tabs and note. The policy list table is identical to the previous screens. At the bottom, there are two buttons: 'Previous Step' and 'Finish'.

关联完成：

The screenshot shows the 'User Details' section for a sub-account named 'test1'. It includes fields for User ID (100027764167), Name, Department, and Contact Information. On the right, there's a 'Quick Operation' bar with 'Associate', 'Unassociate', and 'Delete User' buttons. Below it is a 'Quick Search' bar with a URL link. The main content area shows a table of associations with columns: Association Name, Description, Association Type, Resource Type, Associated Time, and Action. One row is highlighted in blue.

场景三：子账号拥有微搭、云开发指定某环境资源权限

根据 cam 策略模板自定义策略管理，以下设置完成以后，可以指定子账号使用微搭和云开发某一环境下资源，做环境维度的权限隔离。

不同应用在不同环境内开发，以此做到隔离。该子账号只能看到当前环境下的应用的资源。

步骤1：进入到云开发获取环境 ID

如果还没有多环境，则需要首先联系微搭工作人员后台发布多个环境，或手动购买并创建多个环境。

如果已经拥有了多个环境，则进入 [云开发控制台](#)，复制要控制访问权限的环境 ID，即 {envId}。

The screenshot shows the CloudBase Control Panel. On the left, there's a sidebar with '管理' (Management) and a '环境' (Environment) tab. Under '环境' tab, it lists an environment named 'lowcode' with environment ID 'lowcode-3gp1xfvc8844809'. On the right, there are sections for '购买推荐' (Purchase Recommendations) showing '个人版' (Personal Edition) and '入门版' (Getting Started Edition) with their respective prices and purchase buttons. There's also a '产品动态' (Product Updates) section and a '快速入口' (Quick Entry) sidebar.

步骤2：创建 cam 策略

进入到 [访问管理](#)，复制以下模板，创建 cam 策略。

```
{
  "version": "2.0",
  "statement": [
    {
      "principal": "lowcode-3gp1xfvc8844809",
      "action": "cloudfunctions:callFunction",
      "resource": "https://cloudfunction-130318-125303448-125303448.cloudfunctions.net/test"
    }
  ]
}
```

```
{  
    "effect": "allow",  
    "action": [  
        "cam>CreateRole",  
        "cam>AttachRolePolicy",  
        "cam>GetRole",  
        "cam>ListAttachedRolePolicies",  
        "cam>UpdatePolicy",  
        "cam>CreateServiceLinkedRole",  
        "cam>DescribeServiceLinkedRole",  
        "tcb>CheckTcbService",  
        "tcb>DescribePackages",  
        "tcb>DescribeEnvLimit",  
        "tcb>DescribeBillingInfo",  
        "tcb>DescribeExtensionsInstalled",  
        "tcb>DescribeCloudBaseRunAdvancedConfiguration",  
        "tcb>DescribeCloudBaseProjectLatestVersionList",  
        "tcb>DescribeExtensions",  
        "tcb>DescribePostPackage",  
        "tcb>DescribeICPResources",  
        "tcb>DescribeExtensionUpgrade"  
    ],  
    "resource": [  
        "*"  
    ]  
},  
{  
    "effect": "allow",  
    "action": [  
        "cos:*",  
        "scf:*",  
        "cls:*",  
        "ssl>DescribeCertificateDetail",  
        "ssl>DescribeCertificates"  
    ],  
    "resource": [  
        "*"  
    ]  
},  
{  
    "effect": "allow",  
    "action": [  
        "tcb:*"  
    ],  
    "resource": [  
        "qcs::tcb:::env/{envID}"  
    ]  
},  
{  
    "effect": "allow",  
    "action": [  
        "lowcode>GetEnvList"  
    ],  
    "resource": [  
        "qcs::lowcode:::env/{envID}"  
    ]  
}
```

```
]
},
{
"effect": "deny",
"action": [
"lowcode>DeleteEnv"
],
"resource": [
"qcs::lowcode::env/{envID}"
]
}
]
```

1. 新建自定义策略。

The screenshot shows the Tencent Cloud Access Management interface. On the left sidebar, '策略' (Policy) is selected. In the main area, there is a button labeled '新建自定义策略' (Create Custom Policy) which is highlighted with a red arrow.

2. 按策略语法创建。

The screenshot shows a modal dialog titled '选择创建策略方式' (Select Policy Creation Method). It lists four options: '按策略生成器创建' (Create via Policy Generator), '按产品功能或项目权限创建' (Create via Product Function or Project Permission), '按策略语法创建' (Create via Policy Syntax), and '按标签授权' (Authorize via Tag). The third option, '按策略语法创建', is highlighted with a red border.

3. 选择空白模板。

步骤3：粘贴模板到 cam 策略内容

可以命名为 weda_test1，用于后面复制子账号权限使用。

1. 模板命名规则，主要是后面 cam 策略关联子账号时好找，weda_ 子账号用户名。

2. 模板备注里可以加上对应的环境 ID，先去微搭资源管理里复制环境 ID。微搭子账号 test1 权限，环境 ID lowcode-3gp1xflvc8844909。

3. 策略内容里将 {envID}，替换成此子账号所属的环境 ID，例如：lowcode-1gf52uyb34bc16a7。

The screenshot shows the Tencent Cloud Policy Management interface. A policy named 'weda_test1' is selected. The JSON code in the editor looks like this:

```

1  {
2     "version": "2.0",
3     "statement": [
4         {
5             "effect": "allow",
6             "action": [
7                 "cam:CreateRole",
8                 "cam:AttachRolePolicy",
9                 "cam:GetRole",
10                "cam:ListAttachedRolePolicies",
11                "cam:PutRole",
12                "cam:CreateServiceLinkedRole",
13                "cam:DescribeServiceLinkedRole",
14                "tcb:CheckTcbService",
15                "tcb:DescribePackages",
16                "tcb:DescribeRoleList",
17                "tcb:DescribeTcbEnvList",
18                "tcb:DescribeExtensionsInstalled",
19                "tcb:DescribeCloudBaseAdvancedConfiguration",
20                "tcb:DescribeCloudBaseProjectLatestVersionList",
21                "tcb:DescribeExtensions",
22                "tcb:DescribeTestPackage"
23            ],
24            "resource": [
25                "qcs::tcb::env/{envID}"
26            ]
27        },
28        {
29            "effect": "allow",
30            "action": [
31                "lowcode:GetEnvList"
32            ],
33            "resource": [
34                "qcs::lowcode::env/{envID}"
35            ]
36        }
37    ]
38 }

```

A tooltip '环境 ID' is shown above the placeholder '{envID}'. At the bottom of the editor, there are buttons for '保存' (Save) and '取消' (Cancel).

4. control+f 可以搜索定位下，然后复制粘贴环境 ID 值，最后单击保存。

The screenshot shows the code editor with the search results for 'envID' highlighted. The result 'env' is highlighted with a yellow background.

5. 定位到替换 envID 处。

The screenshot shows the code editor with the search results for 'envID' highlighted. The result 'env' is highlighted with a yellow background.

6. 替换完成后。

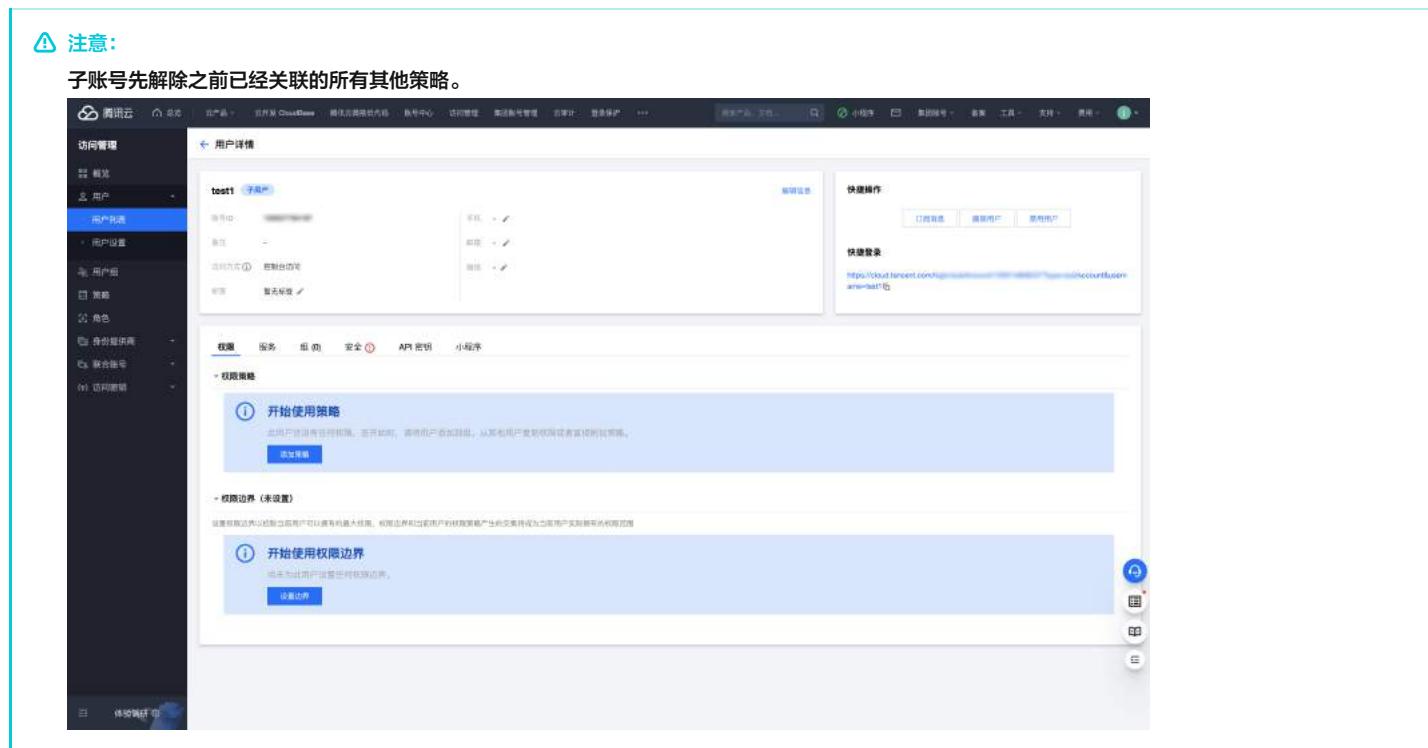


```
44,      "effect": "allow",
45,      "action": [
46,          "tcb:*"
47,      ],
48,      "resource": [
49,          "qcs::tcb::env/lowcode-1gf52uyb34bc16a7"
50,      ]
51,  },
52,  {
53,      "effect": "allow",
54,      "action": [
55,          "lowcode:GetEnvList"
56,      ],
57,      "resource": [
58,          "qcs::lowcode::env/lowcode-1gf52uyb34bc16a7"
59,      ]
60,  },
61,  {
62,      "effect": "deny",
63,      "action": [
64,          "lowcode>DeleteEnv"
65,      ]
```

策略语法说明 [查看projectId对应的项目](#)

步骤4：子账号关联对应的 cam 策略

配置完成后，该子账号 test1 可拉取和访问该环境下的应用。



注意：
子账号先解除之前已经关联的所有其他策略。

在腾讯云控制台的“用户管理”-“用户”-“用户列表”中，找到“test1”子账号，进入其“用户详情”页面。在“权限”-“权限策略”下，可以看到两个策略界面，分别显示了“开始使用策略”和“开始使用权限边界”的说明文字，并且有“设置策略”和“设置边界”的按钮。

1. 可以按照上面命名的策略名称: weda_test1, 搜索下。然后关联此策略。

2. 单击确定后, 完成关联。

3. 子账号拥有特定环境的访问权限配置完成。