

# 腾讯轻联 控制台指南



腾讯云

**【 版权声明 】**

©2013–2025 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內容。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或95716。

# 文档目录

## 控制台指南

工作台

集成开发

流程

流程管理

流程配置

逻辑组件使用指南

删除变量 (Remove Variable)

子流引用 (Flow Reference)

定时任务 (Scheduler)

并行分支 (Scatter Gather)

并行遍历 (Parallel Foreach)

延时 (Sleep)

抛出错误 (Raise Error)

数据转换 (Transform)

数据映射 (Mapping)

条件重试 (Until Successful)

终止循环 (Break)

跳过循环 (Continue)

遍历 (For Each)

配置 Payload (Set Payload)

配置变量 (Set Variable)

错误捕获 (Try-Catch)

条件判断 (Choice)

API 管理

运维中心

监控统计

运行日志

告警管理

告警策略

通知模板

通知模板

接口回调

告警接收渠道

告警历史

管理工具

授权配置

本地代理

数据存储

全局变量

连接

连接器

开放平台

模板中心

系统管理

成员管理

项目管理

环境管理

操作审计

Dataway 表达式

简介

快速入门

开发指南

Dataway 语言基础概念

文本模式

表达式模式

表达式模式附录

代码模式 Python

Python 附录

代码模式 Java

场景案例

常见问题

# 控制台指南

## 工作台

最近更新時間：2024-11-07 11:41:52

腾讯轻联 **工作台** 集中展示了新建流程、模板推荐、帮助中心及产品动态等内容，本文将展开详细介绍。

### 新建流程

主页提供了快速创建流程的入口，分为两种方式：快速创建和通过模板创建。



### 选择快速创建

在快速创建中，进行下拉选择，通过点选就可以快速创建流程。

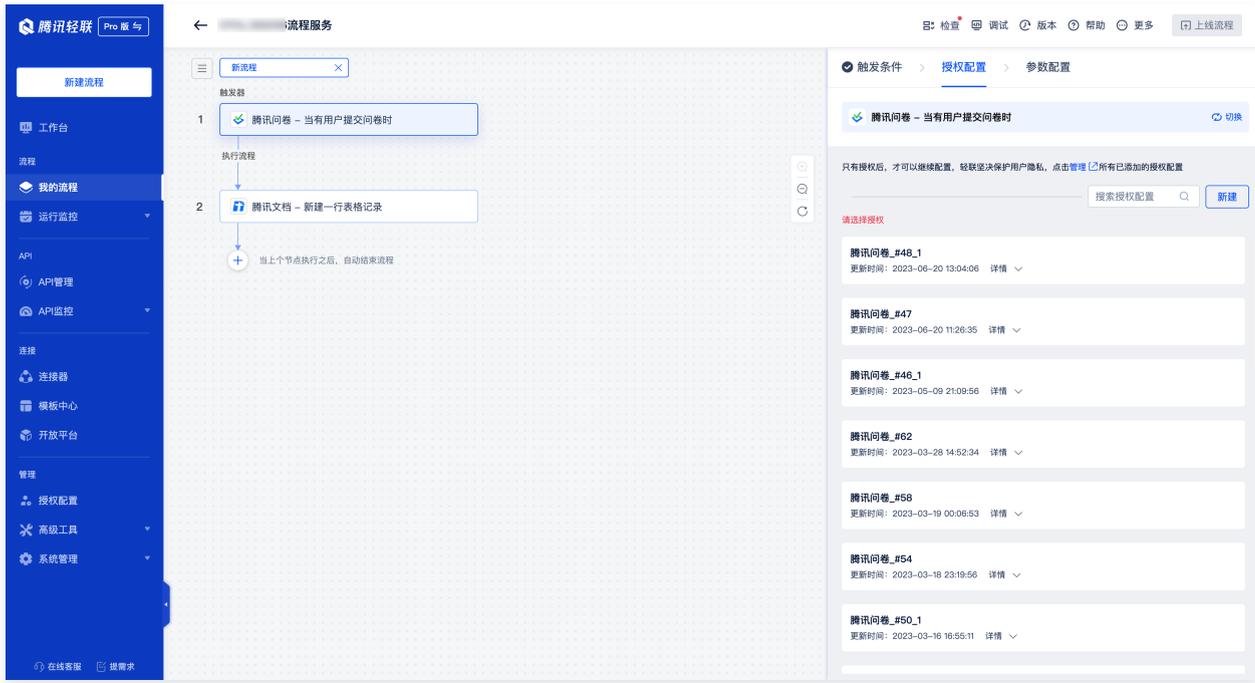
1. 选择一个触发器，并选择一个触发器的触发事件。



2. 选择执行操作，并选择一个执行操作的事件。



3. 单击**新建流程**后，进入到流程画布中，开始继续配置流程，可参见 [流程管理](#)。



### 通过模板创建

在创建空白应用中，您可以从零开始，自由配置。

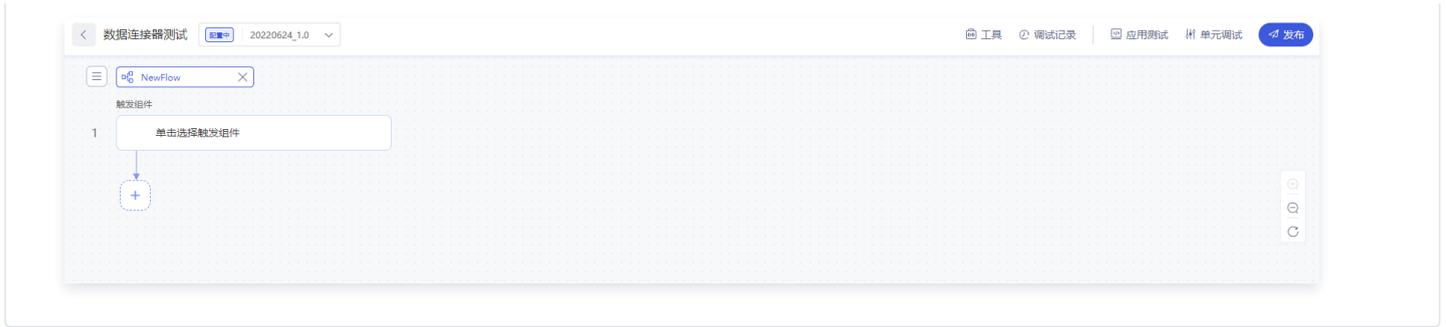
1. 在创建集成应用的创建空白应用模块，单击开始创建。



2. 填写应用名称，单击完成。



3. 进入集成流开发界面配置集成流，详情请参见 [应用配置](#)。



## 流程模板推荐

推荐常用或者热门的流程模板，支持开箱即用，快速创建自动化流程，也可以查看 [全部模板](#)。



## 流程列表

主页中的流程列表展示了对应账户下部分应用的基本信息，按照更新时间展示最近更新五个流程的信息，若想查询账号下所有应用信息，单击[更多](#)即可。



## 产品动态

主页中的产品动态展示了产品最近新增或更新的文档或功能等相关信息，方便用户更直观的了解产品的新增特性。若需了解更多历史更新内容，可单击[更多](#)进入 [产品动态](#) 进行查看。

**流程模板推荐** 查看更多 >

每天定时发送疫情信息到企业微信内部群  
★★★★

疫情信息服务 + 企业微信机器人 →

维格表神奇表单上传健康码或行程码自动识别...  
★★★★

维格表 + AI 通信行程卡识别 + ... →

每天定时发送疫情信息到钉钉内部群  
★★★★

疫情信息服务 + 钉钉机器人 →

客户到访时间提交归档到维格表并发送企微...  
★★★★

腾讯问卷 + 维格表 + 企业微信... →

**流程列表** 查看更多 >

概览	名称	状态	上线
	0714_152236流程服务	配置中 版本1	<input type="checkbox"/>
	0605_164943流程服务	配置中 版本1	<input type="checkbox"/>
	企微demo	配置中 版本17	<input type="checkbox"/>
	定时	配置中 版本2	<input type="checkbox"/>
	webhook	配置中 版本4	<input type="checkbox"/>

**产品动态** 查看更多 >

- 2023年04月
  - 优化告警信息
  - 优化API导出
  - 新增API配置状态
- 2023年03月
  - 优化API监控
  - 连接配置优化
  - 简化凭证新建
- 2023年02月
  - 优化服务创建方式
  - 新增参数映射
  - 新增错误码配置

开发者能力认证 去认证

**帮助中心** 查看更多 >

## 帮助中心

主页中的帮助中心，为用户提供了进一步了解腾讯轻联的学习视频，使用文档。

**帮助中心** 查看更多 >

新手必看轻联是什么?

腾讯轻联介绍 Hot

新手入门如何使用轻联

新人入门必看 Hot

玩转企业微信

企微机器人场景教学

腾讯轻联介绍 Hot

如何新建流程 Hot

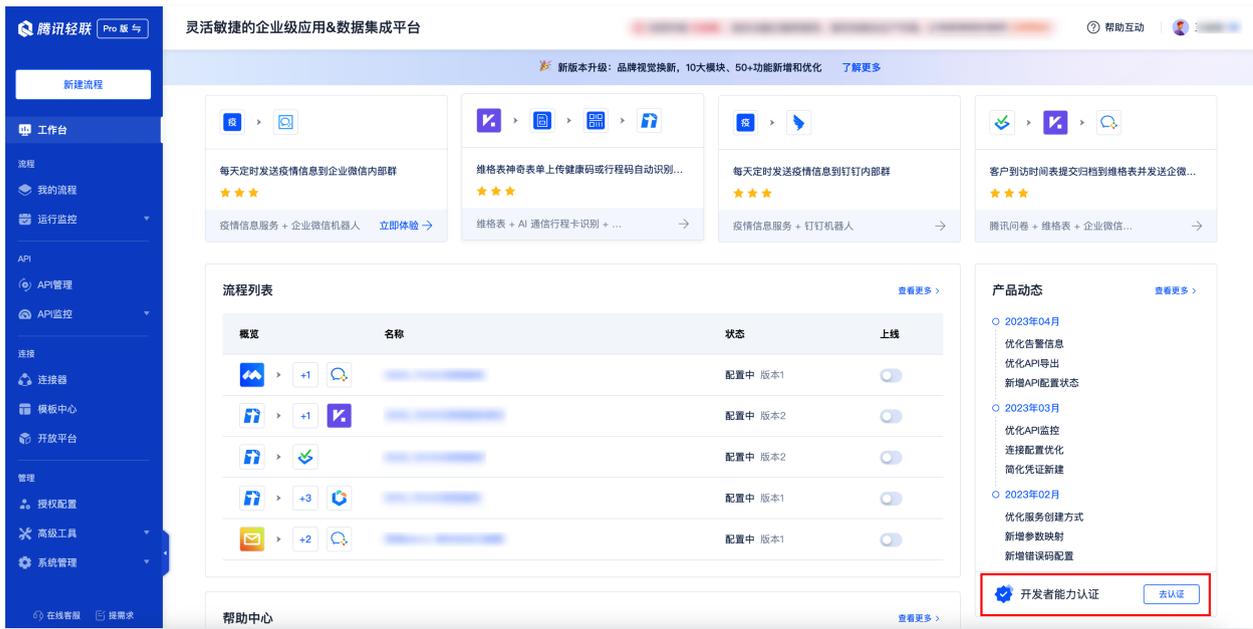
如何配置腾讯文档

如何配置腾讯问卷

如何配置企业微信群机器人

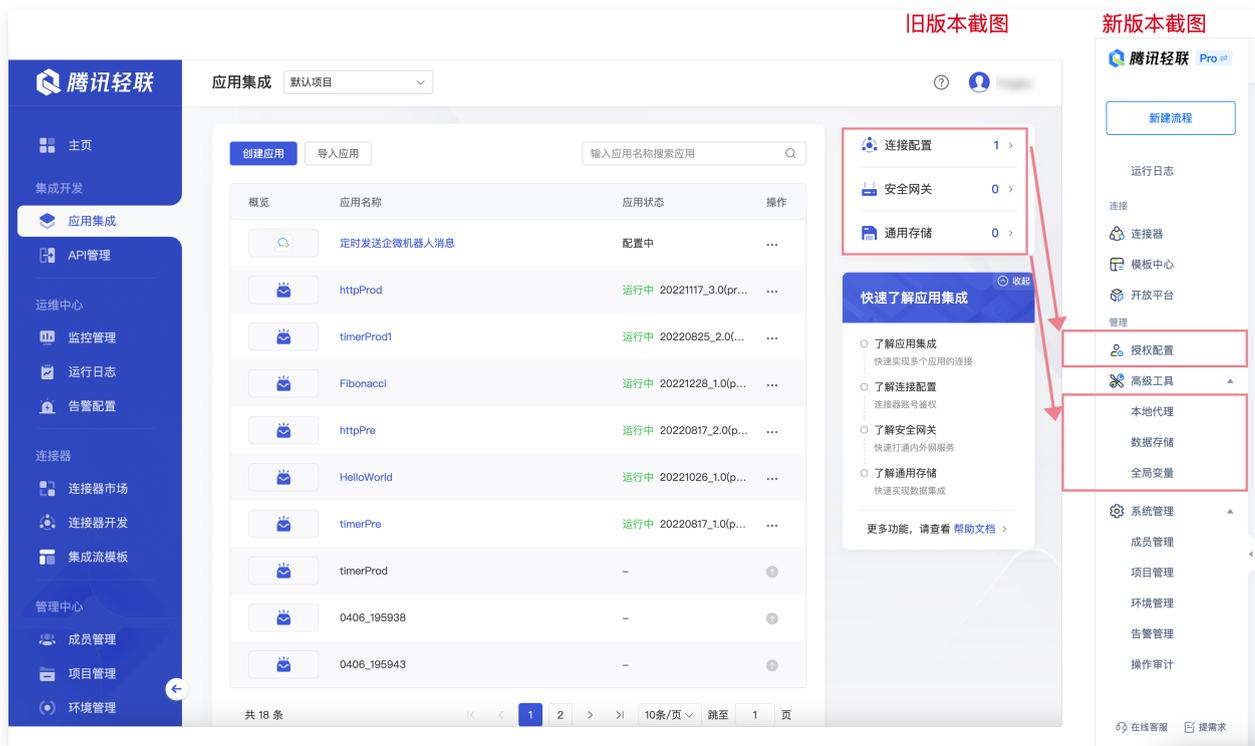
## 开发者能力认证

腾讯轻联开发者认证是腾讯轻联联合腾讯产业互联网学堂发起的一款针对腾讯轻联学习考试的认证。拥有该证书表示您对腾讯轻联产品具备基本的了解及应用能力。



## 菜单栏

- 原应用集成中的连接配置、安全网关、通用存储、项目变量，升级为到菜单栏中。



- 新增功能：不同版本切换入口、快速创建流程入口、在线客服、提需求。
- 功能模块重组和整合：体现在修改名称和移动位置。

### 旧版本截图

### 新版本截图

- 1、新增版本切换入口
- 2、新增快速创建流程入口
- 3、应用集成：改为「我的流程」
- 4、流程日志：单独拆分为「监控统计」和「运行日志」
- 5、API：单独拆分为「监控统计」和「运行日志」
- 6、连接器开发：升级为「开放平台」
- 7、集成流模板：改为「模板中心」
- 8、二级菜单升级为到菜单栏中，并修改名称  
连接配置：改为「授权配置」  
安全网关：改为「本地代理」  
通用存储：改为「数据存储」  
项目变量：改为「全局变量」
- 9、成员管理、项目管理、环境管理：移动位置
- 10、告警配置：改为「告警管理」
- 11、审计日志：改为「操作审计」
- 12、新增「在线客服」&「提需求」

# 集成开发 流程 流程管理

最近更新时间：2024-06-26 11:01:31

## 操作场景

集成需求可以通过创建流程进行实施，复杂的场景可以创建多个流程，每个流程对应一个集成场景。进入画布，您可以操作新建流程、配置流程、修改流程名称、导出导入应用以及删除应用等操作。

不同的连接器状态，对应操作下展示的功能是有区别的，具体如下：

### ⚠ 注意：

独立控制台的调试，操作的是当前应用的一个特定版本，当前版本在缩小调试框后仍然可以进行配置。

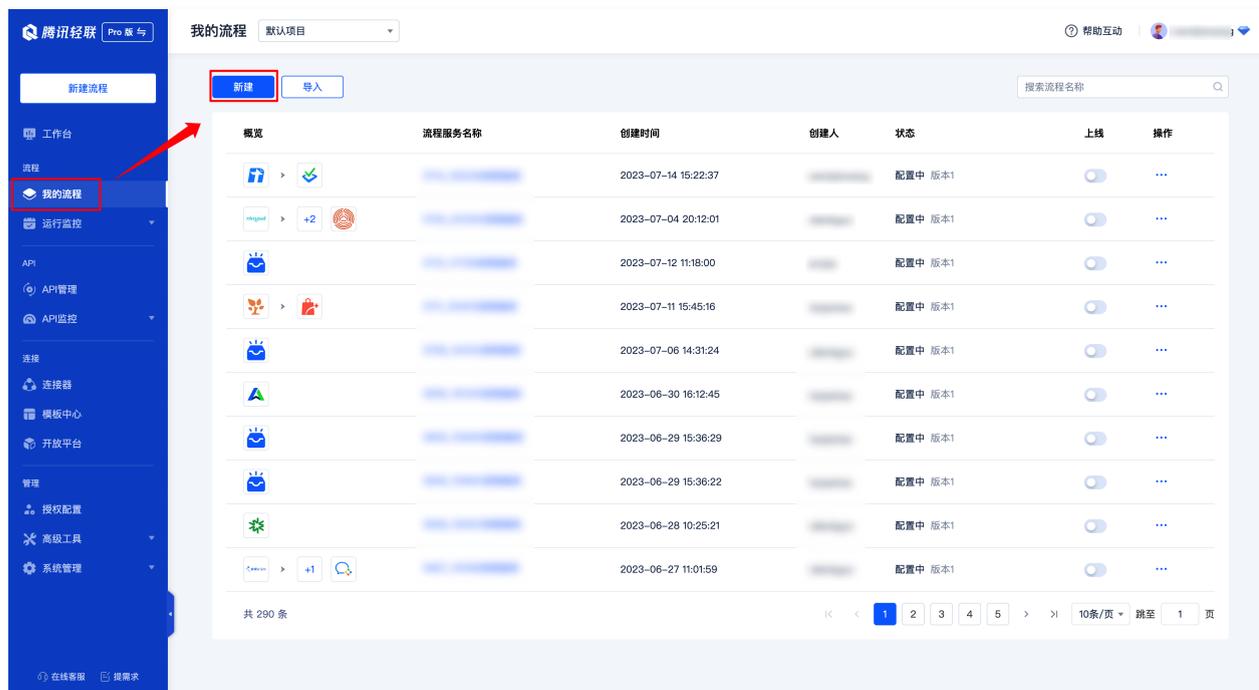
状态	功能
运行中	允许用户进行查看、修改名称、复制、配置、导出和停止。
配置中	允许用户进行查看、修改名称、导出、配置、上线和删除。
已停止	允许用户查看、配置、修改名称、发布和删除。
调试中	允许用户查看、配置、修改名称、导出、流程测试、节点测试和退出调试模式。

## 操作步骤

### 创建流程

当需要新建业务流时，您可按以下步骤操作：

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**我的流程**。
2. 在我的流程页面，选择对应项目名称后单击**新建流程**。



## 配置流程

以腾讯问卷的数据，自动同步到腾讯文档为例，配置一个完整流程，请参见 [新建流程教学指引](#)。

## 流程管理

流程列表页可以进行修改流程名称、添加备注、上/下线、导入/导出、删除等操作。



## 导入和导出流程

当您想和其他账号共享应用或者修改应用所属项目时（目前暂不支持直接修改应用所属项目及跨账号共享等操作），您可以使用导出导入功能。

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**我的流程**。

2. 在应用集成页面，找到目标应用，单击左上角**导入**或流程列表页的**导出**即可完成操作。

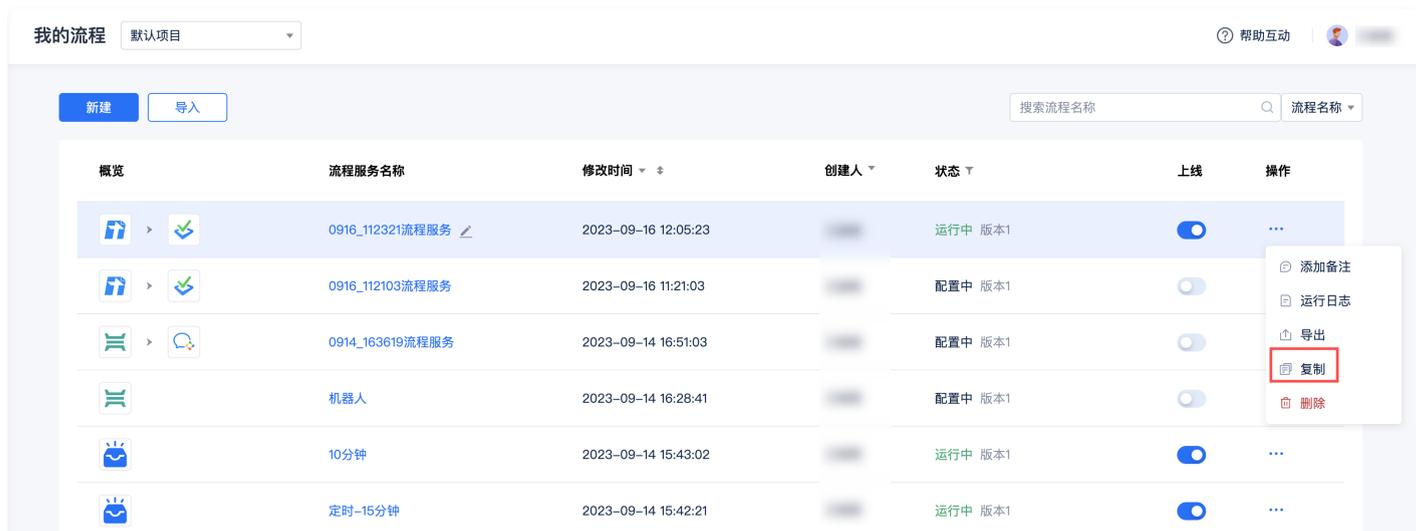
- 导出时可以选择不同版本的全部导出或选择需要的流程和授权配置导出，导出后以 `.ipaas` 为后缀的文件将自动下载。
- 导入时只需要上传以 `.ipaas` 为后缀的文件即可。



## 复制流程

针对某些类似的场景，为了避免重复创建流程，可以使用流程复制功能。

- 列表中展示的是版本，即为所要复制的版本。
- 复制版本不会复制其他历史版本。
- 复制后，流程默认是下线状态，需要手动单击上线。



## 删除流程

处于非运行中状态下的流程可以进行删除操作。

### 注意：

删除后数据无保留，请删除前导出做好数据备份或确认数据必要性再进行删除操作。

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**我的流程**。
2. 在我的流程页面，找到目标流程，单击**删除流程**。



## 流程筛选和排序

流程列表较多的时候，可以使用流程筛选和排序功能。

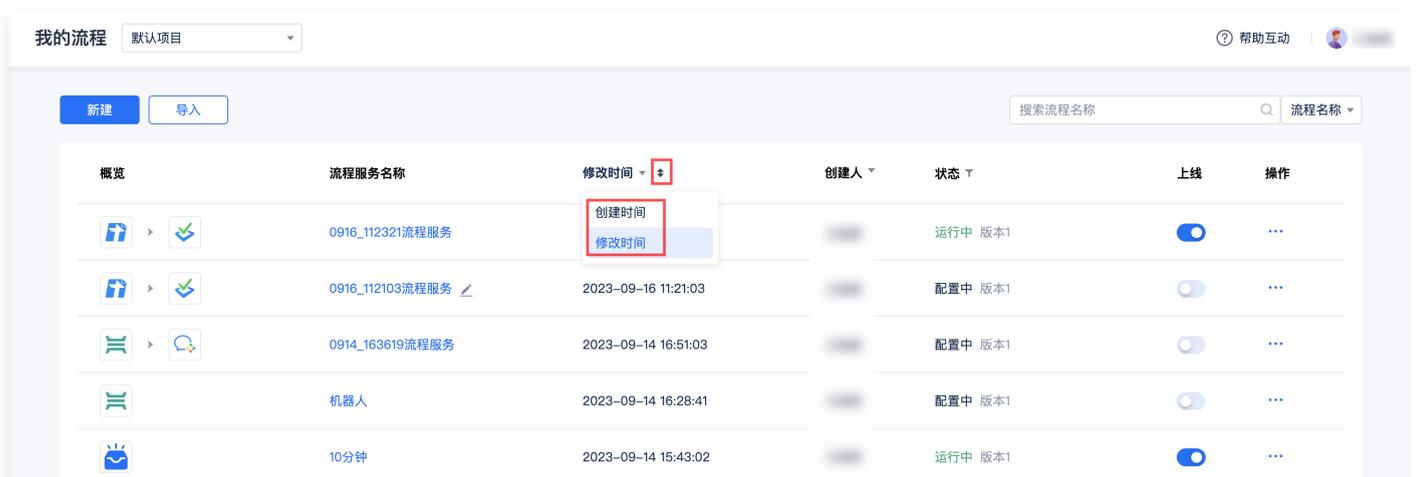
- 运行状态筛选：可以筛选运行中、调试中、配置中和已下线的状态。



- 流程搜索：支持流程名称搜索、支持创建人/修改人搜索。



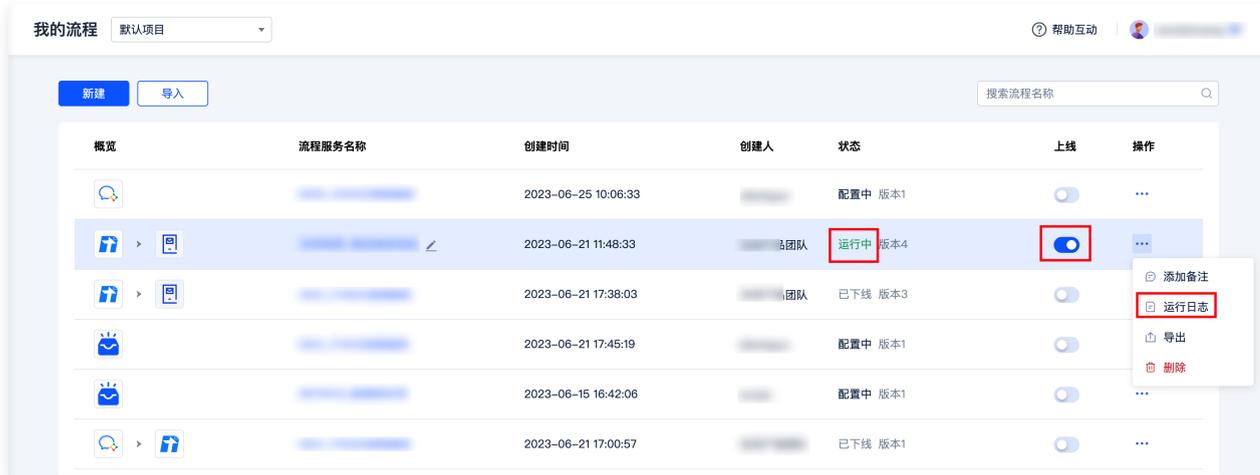
- 流程排序：支持安装创建时间排序、修改时间排序。



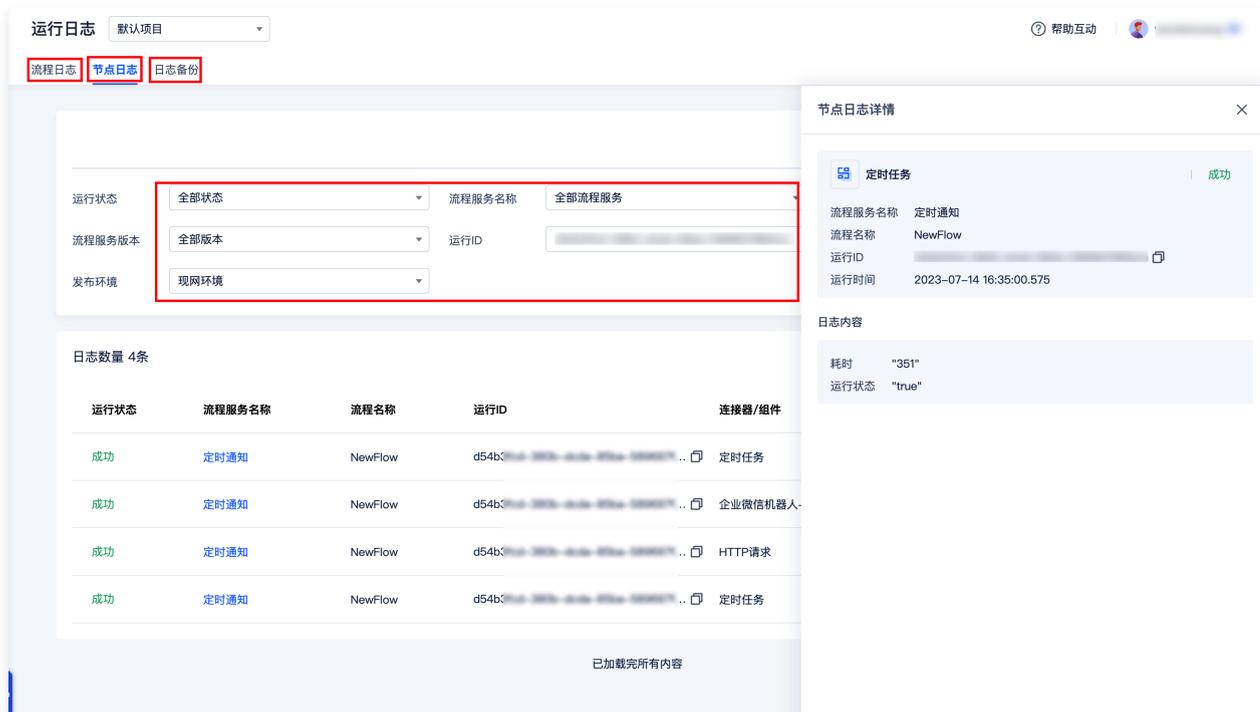
## 运行日志

处于运行中状态下的应用，可以快速查看应用的运行日志：

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**我的流程**。
2. 在我的流程页面，找到目标流程，单击操作列的 **运行日志**。



3. 在**运行日志**页面，您可查看当前运行应用、当前运行版本的运行日志（默认返回近30分钟时间段内的日志）。



# 流程配置

最近更新时间：2024-06-26 14:37:51

## 操作场景

新建或进入一个流程后，可以通过新建多条流来实现业务逻辑。流程画布中预留触发类组件位，可以放置拥有触发能力的逻辑组件或连接器，例如：HTTP Listener、Kafka、定时器等，拥有触发能力的流为主流，不具备触发能力的流则只能被其他流通过逻辑组件 [Flow Reference](#) 引用为子流。画布顶部导航有多个工具包括：检查、调试、版本管理、帮助中心、更多等。

## 流程

流程由触发器和执行操作组成，流程上线之后，当触发器符合条件时，流程自动启动，并自动执行操作，例如：

触发器为 **腾讯问卷**，触发条件为 **当有新的问卷提交时**。

执行操作为 **腾讯文档**，新建一行表格记录。

那么流程可以实现：当腾讯问卷有新的问卷提交时，自动将问卷信息，写入到腾讯文档中。



## 流程功能模块

流程的各功能模块如下：

功能名称	说明
流程标题	描述流程的场景或效果等，例如生日祝福提醒。
触发器	控制流程何时启动，例如每天上午9点定时启动、当腾讯问卷有新的问卷提交时启动等。
执行操作	流程启动后，自动执行的操作，例如在腾讯文档中写入一行数据、发送一个企微信群消息等。
配置面板	配置流程的重要模块，例如选择某个触发器，以及触发器的具体配置等。
流程变量	执行动作配置中，可以引用触发器中的数据，例如在腾讯文档中，可以引用问卷中数据。
工具栏	可以对流程进行上下线、试运行、查看或恢复历史版本等操作。
节点	触发器、执行动作，也称为节点，下图流程中有两个节点。



## 流程创建入口

创建流程有以下3个入口：

- 入口一：快速创建，可以根据下拉选择触发器和执行操作（更适合新用户）。
- 入口二：创建新的流程，创建空白流程（更适合有经验的用户）。
- 入口三：根据模板创建，在配置有某些参数的模板中创建（更适合全部用户）。

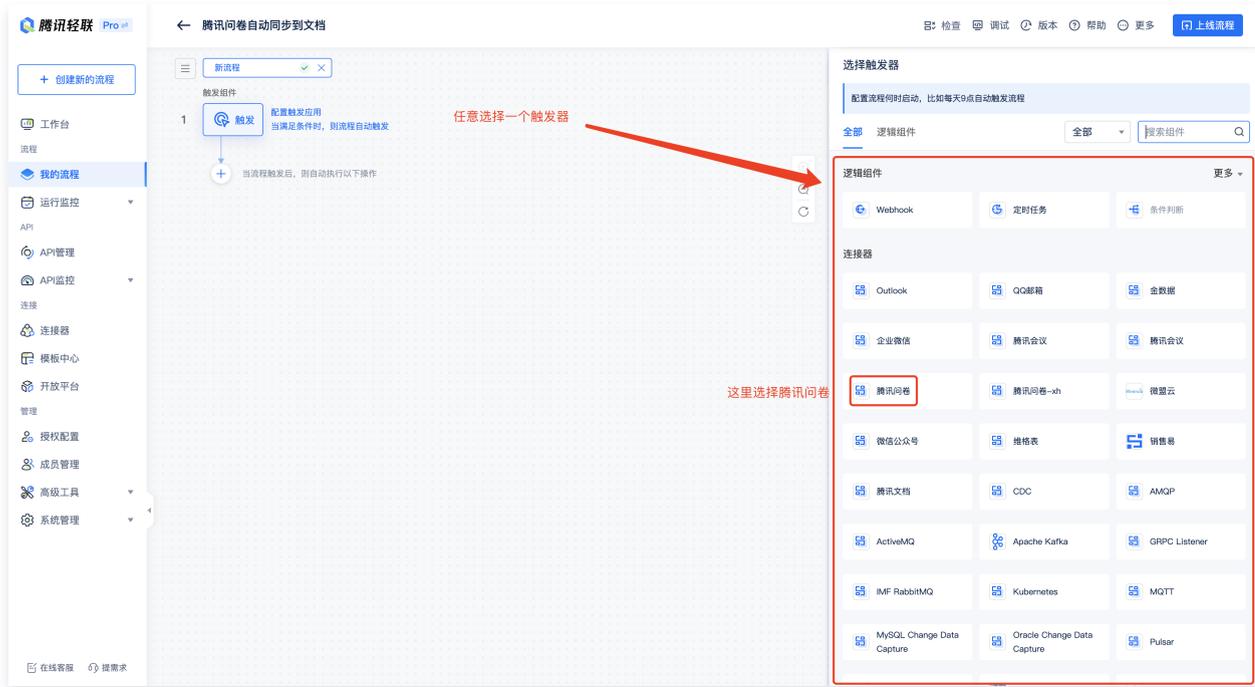


## 新建流程实例教学

以腾讯问卷的数据，自动同步到腾讯文档为例，配置一个完整流程。

### 步骤1：配置触发器

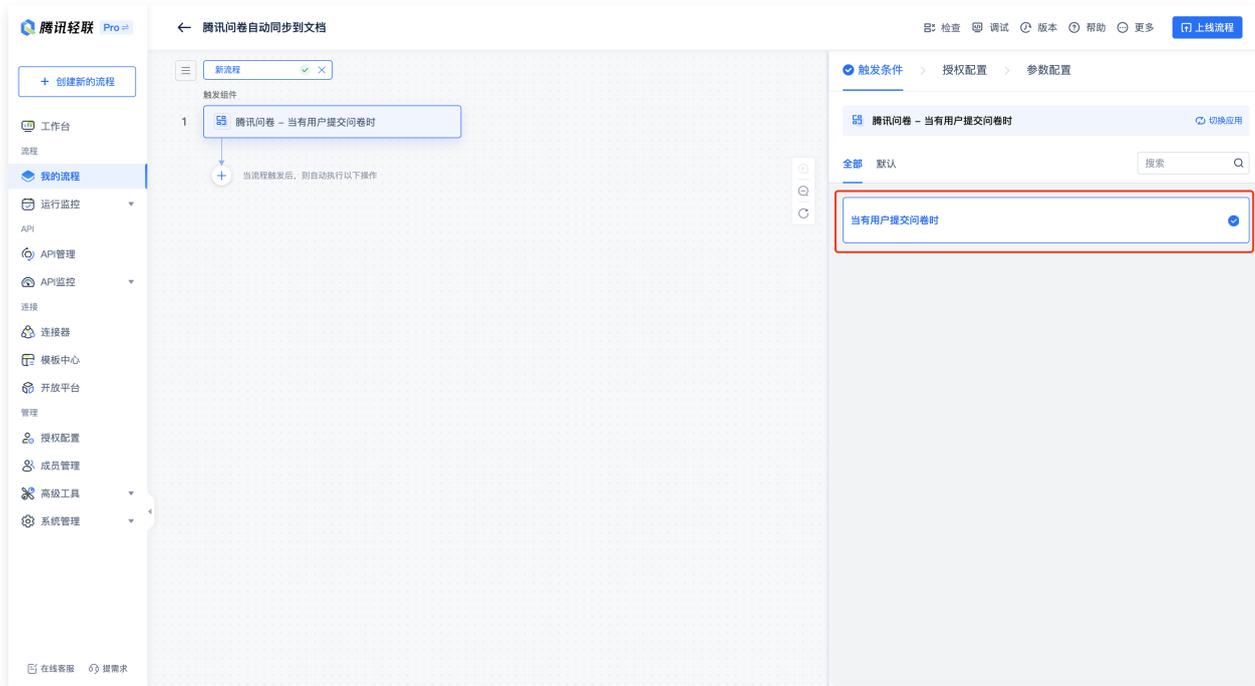
1. 选择触发器，这里选择腾讯问卷。



### 2. 触发条件。

这里选择当有新用户提交问卷时，触发条件一般有以下类型：

- 实时触发：例如当有新用户提交问卷时，流程自动触发。
- 定时触发：例如每天9点，流程自动触发。

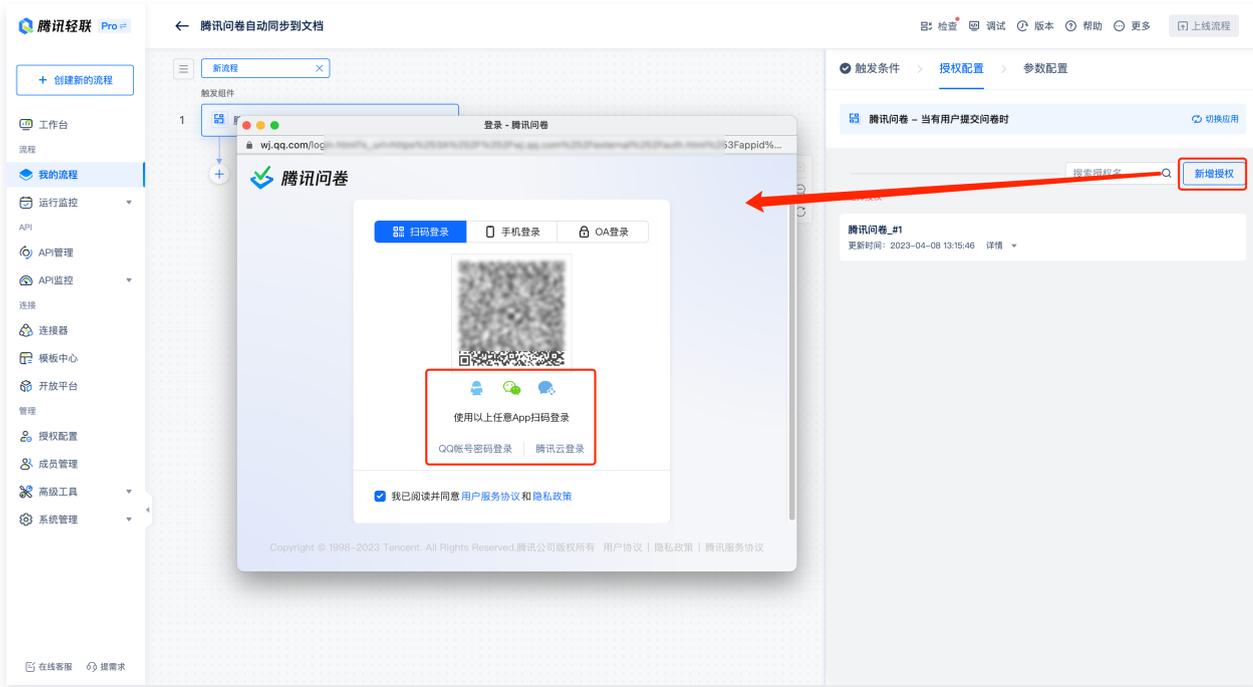


### 3. 授权配置。

只有授权后，流程才可以继续配置，轻联坚决保护用户隐私，请放心授权。根据自己问卷的登录方式进行扫码登录，如果有多账号时要注意，不同账号数据不通。

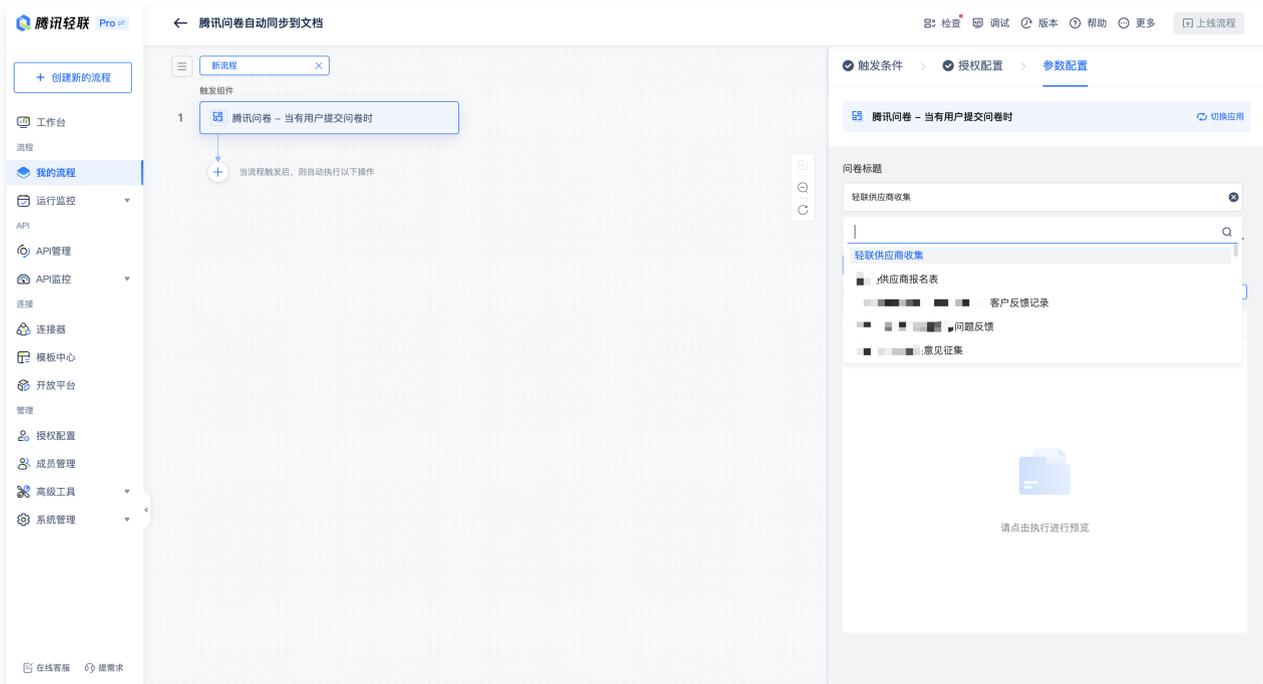
此处常见问题如下：

- 扫码异常：可能不是管理员，需要管理员才有权限。
- 扫码后，没有问卷的数据：只有付费的团队，才可以获取到数据。
- 扫码后，有些问卷有，有些没有：只能看到自己创建的问卷，看不到别人创建的问卷。



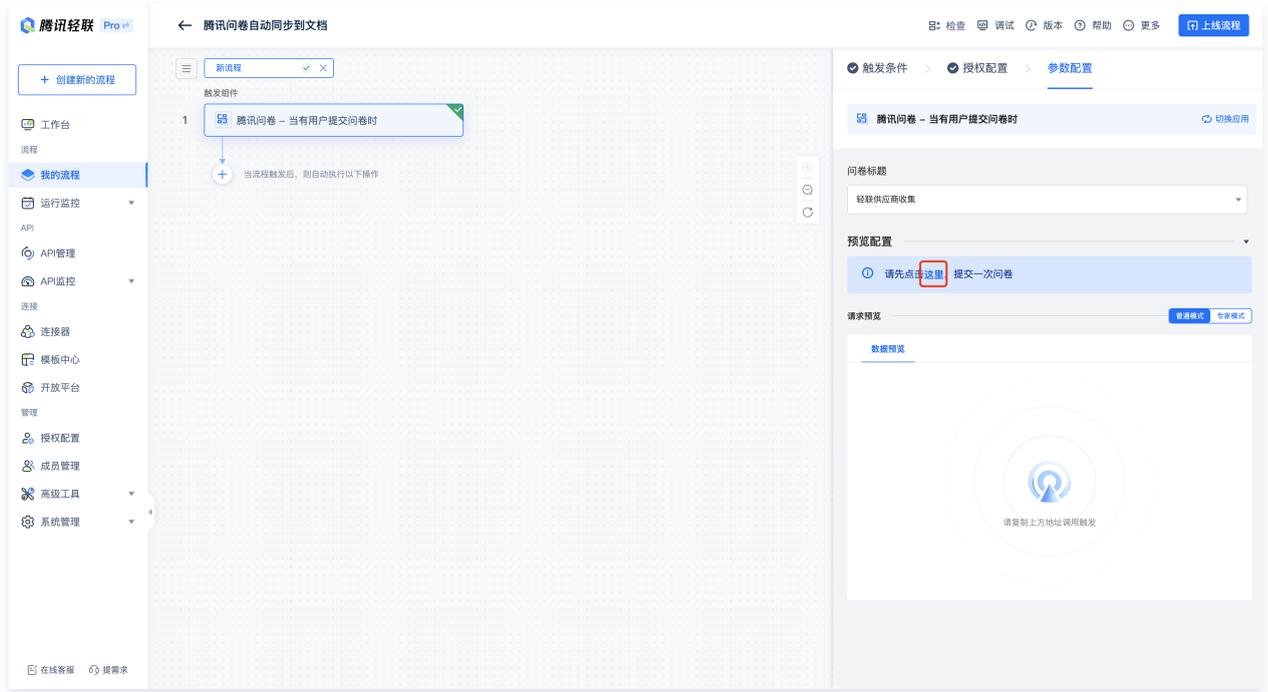
#### 4. 参数配置。

- **基础配置：**选择想要监控的表单，一个流程只能监控一个表单。



- **预览配置：**非必须操作，一般是建议预览。例如后续节点需要引用此节点的数据时，建议进行预览，以确保有数据输出。每个触发器的预览方式不同。例如定时任务，可以单击执行预览。但对于腾讯问卷，需要填写一个真实的问卷才能成功预览。具体操作步骤如下：

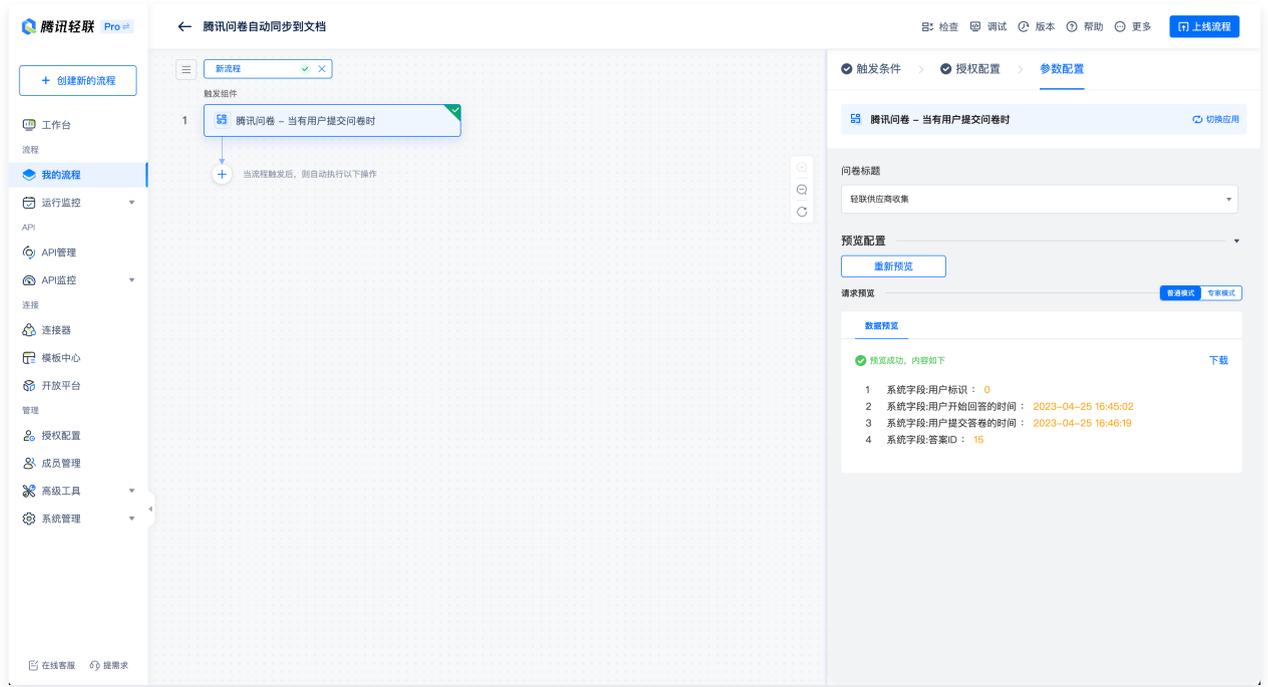
##### 4.1.1 单击这里去填写一个问卷。



4.1.2 填写真实问卷，并提交。

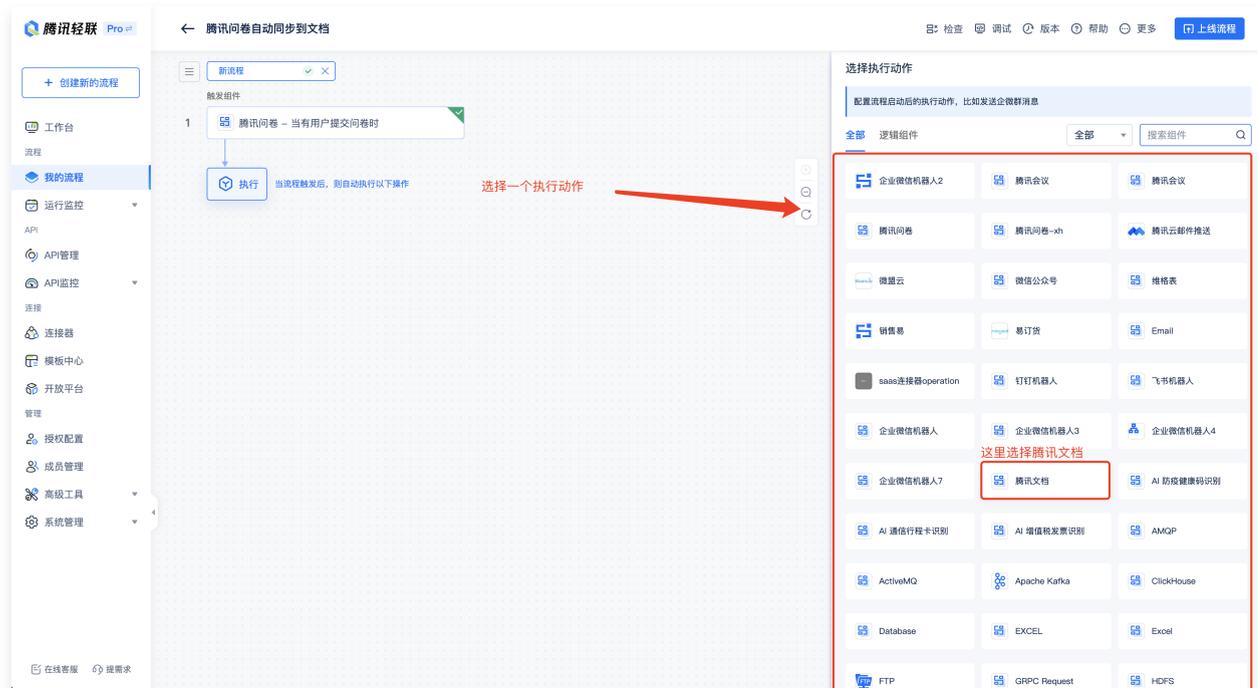


4.1.3 此时等待几秒钟，刚才提交的数据，会自动展现在下方。



## 步骤2：配置执行操作

1. 选择执行操作这里选择腾讯文档。



2. 执行操作。

这里选择新建一行表格记录：自动在指定的表格中，新增一行数据内容。



### 3. 授权配置。

只有授权后，流程才可以运行，轻联坚决保护用户隐私，请放心授权。根据自己文档的登录方式进行扫码登录，如果有多账号时要注意，不同账号数据不通。

此处常见问题如下：

- 扫码后，看不到想要的表格：只能看到自己创建的表格，看不到别人创建的表格。
- 为什么没有企业微信扫码：目前腾讯文档没有提供企业微信登录的扫码方式。等对方支持后，我们第一时间加上。

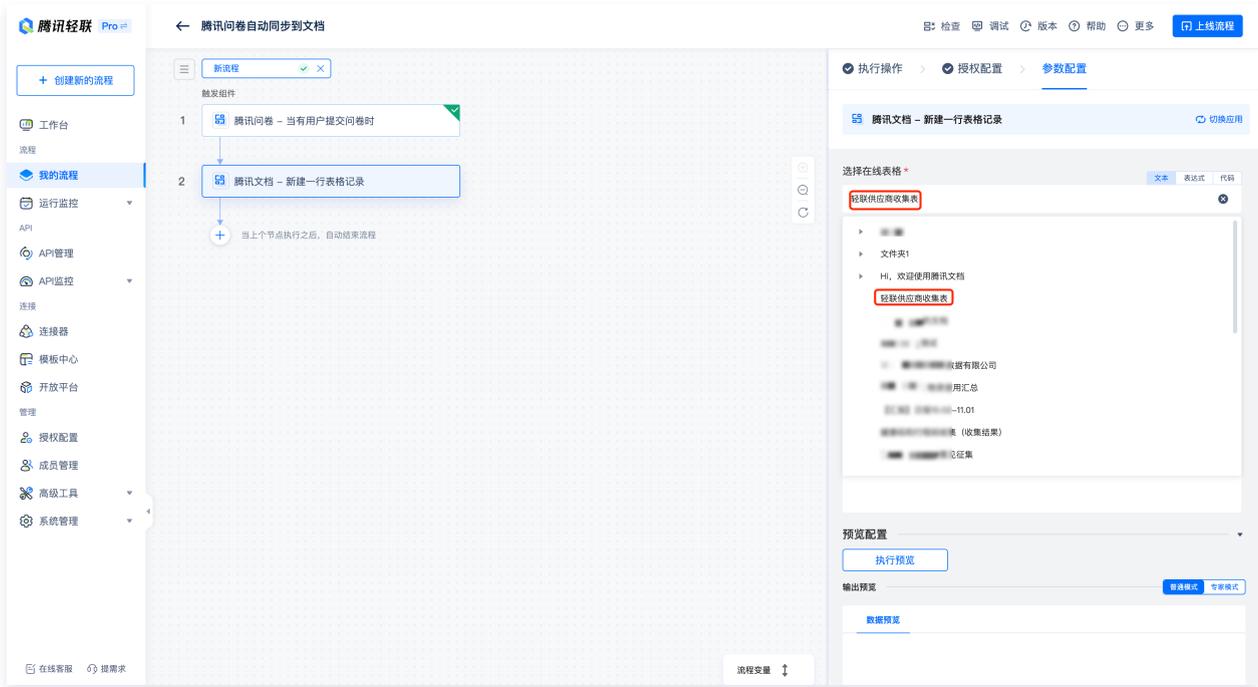


### 4. 参数配置。

4.1 选择想要写入的表格，再选择工作表，注意一定要确保工作表第一行有内容。

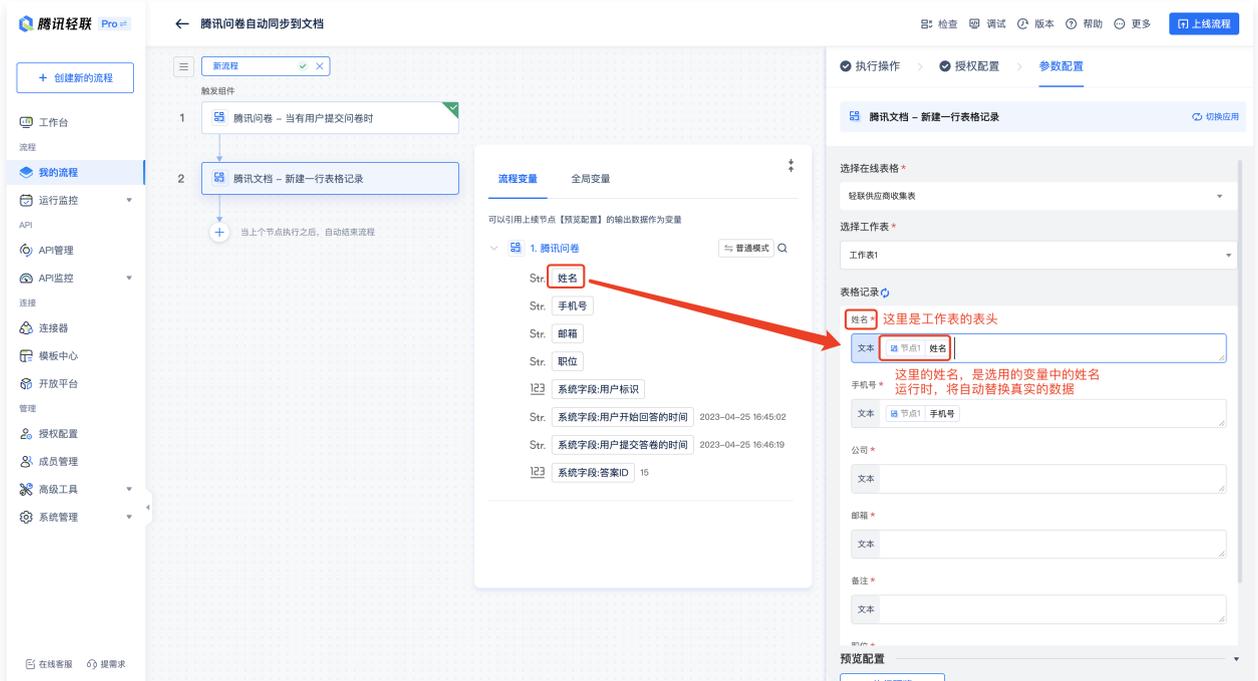
此处常见问题如下：

- 该工作表无表头：请先在表格第一行添加数据后，再重试。
- 注意甄别工作表和智能表：工作表是常见的在线表格，智能表是一种新型在线表格，类型不同，执行操作事件也不同。



4.2 逐项匹配表格需要写入的内容。

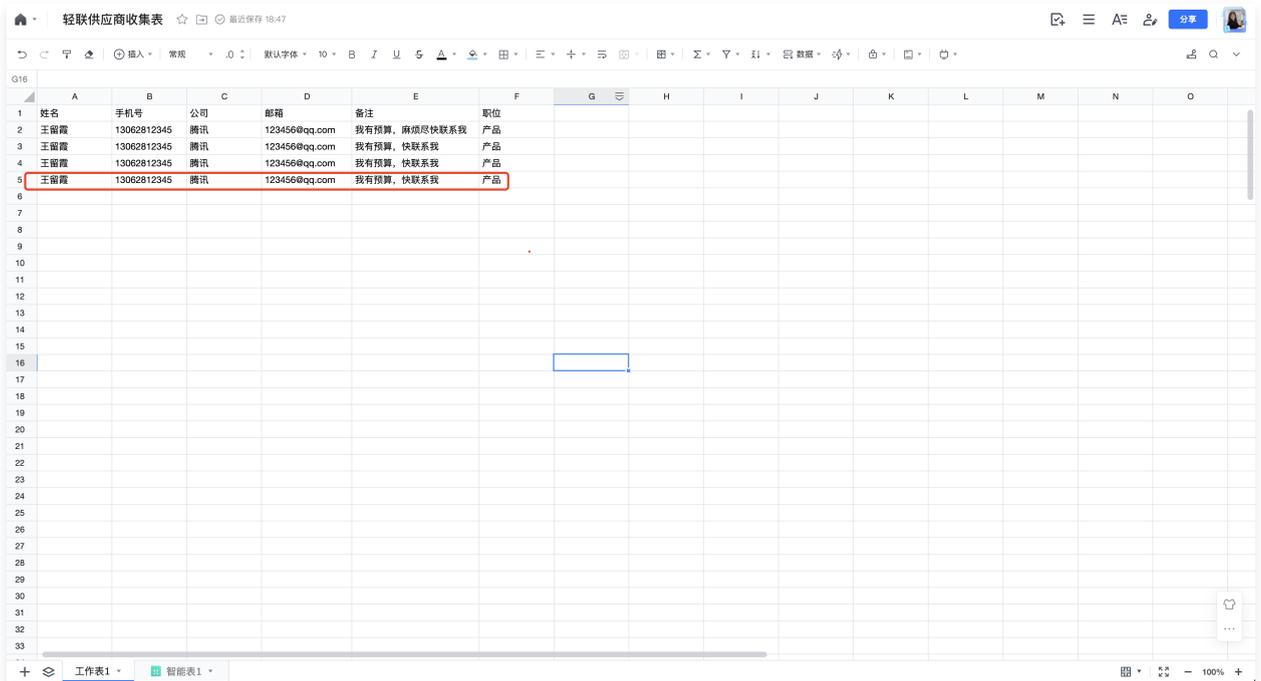
例如：想要将问卷中的姓名，填写到问卷中的姓名。则在姓名中，填入问卷中输出的变量-姓名，实际运行时，将替换真实的数据。



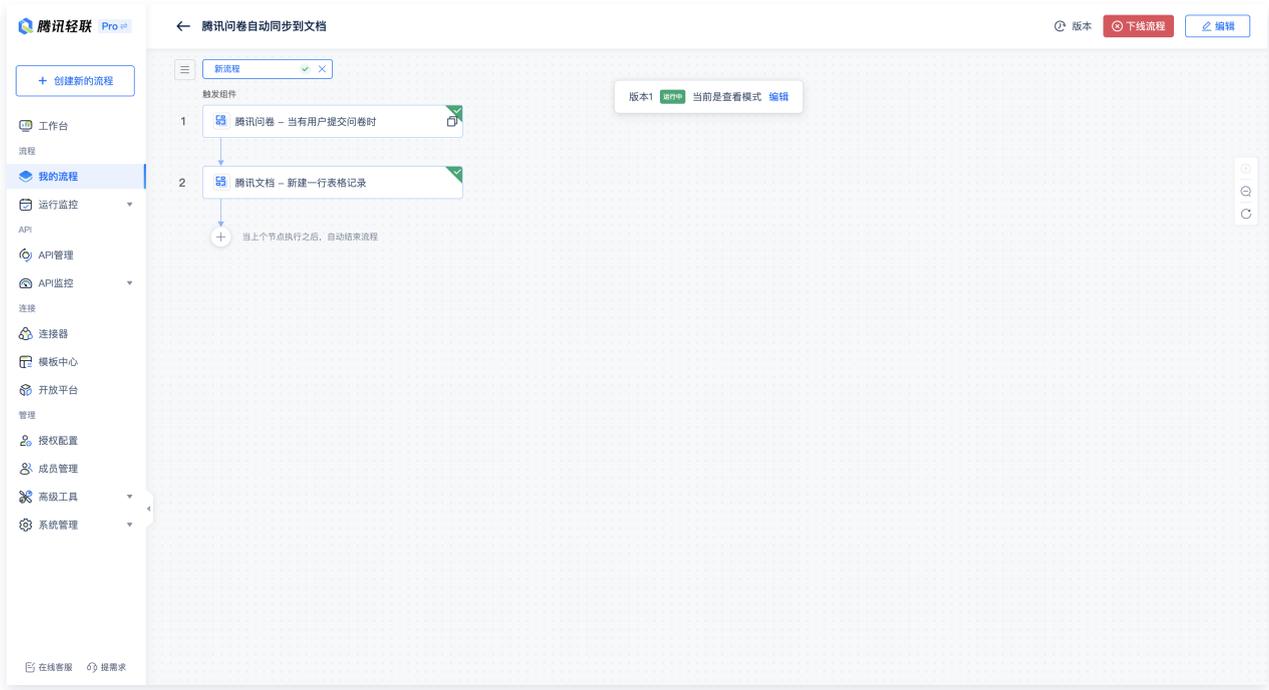
预览配置：非必须操作，一般是建议预览。例如后续节点需要引用此节点的数据时，建议进行预览，以确保有数据输出。以下是预览效果图：



4.3 腾讯文档已经自动填写进来了，在第5行。



4.4 恭喜您，到这里流程已经搭建完成啦，赶快上线流程吧。

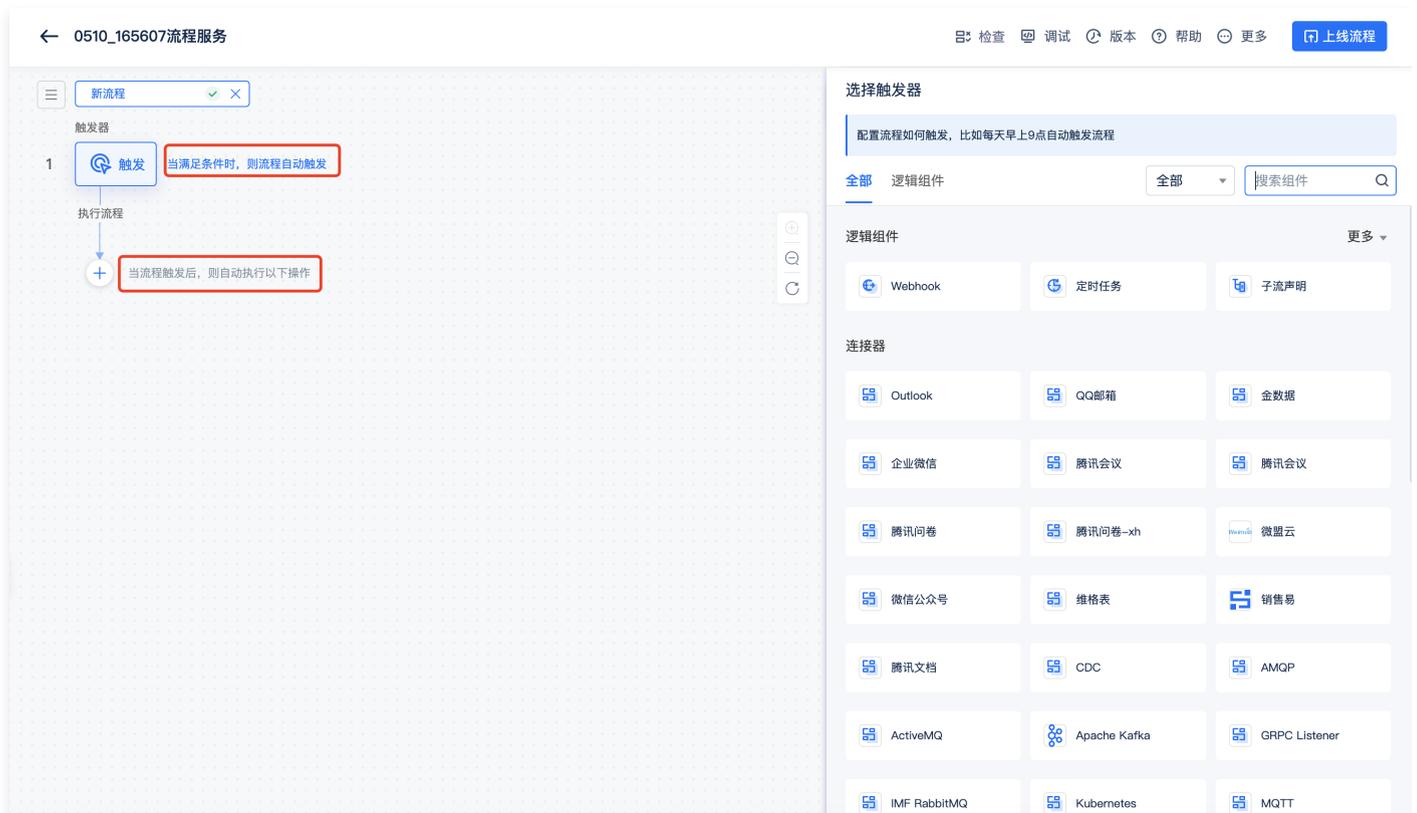


5. 此时可以实现：当腾讯问卷有新的问卷提交时，可以自动写入到腾讯问卷。

## 流程编辑

### 节点增删改

1. 新建或进入一个流程后，可以在画布内通过选择触发器和执行操作来配置流程触发和执行。



2. 触发器支持的操作：添加备注和删除。

← 0510\_165607流程服务

吕: 检查 调试 版本 帮助 更多 上线流程

新流程

触发器

1 腾讯会议 - 会议回调事件接收器

添加备注  
删除

执行流程

2 163邮箱

当上个节点执行之后, 自动结束流程

执行操作 > 授权配置 > 参数配置

163邮箱 切换

全部 SMTP POP3 IMAP 搜索

发送邮件

收取邮件

收取邮件(IMAP)

添加备注后, 备注即可展示在连接器下方, 便于查看。

← 0510\_165607流程服务

吕: 检查 调试 版本 帮助 更多 上线流程

新流程

触发器

1 腾讯会议 - 会议回调事件接收器

腾讯会议

执行流程

2 QQ邮箱

当上个节点执行之后, 自动结束流程

若添加的连接器暂时不需要, 可通过删除按钮或者右侧面板的切换来删除或更改。

← 0510\_165607流程服务

吕: 检查 调试 版本 帮助 更多 上线流程

新流程

触发器

1 腾讯会议 - 会议回调事件接收器

腾讯会议

修改备注  
删除

执行流程

2 QQ邮箱

当上个节点执行之后, 自动结束流程

触发条件 > 授权配置 > 参数配置

腾讯会议 - 会议回调事件接收器 切换

只有授权后, 才可以继续配置, 轻联坚决保护用户隐私, 点击管理 所有已添加的授权配置

三方开发应用的AppID \*

文本

需拥有企业管理员权限, 点击用户中心->高级->restApi, 即可看到

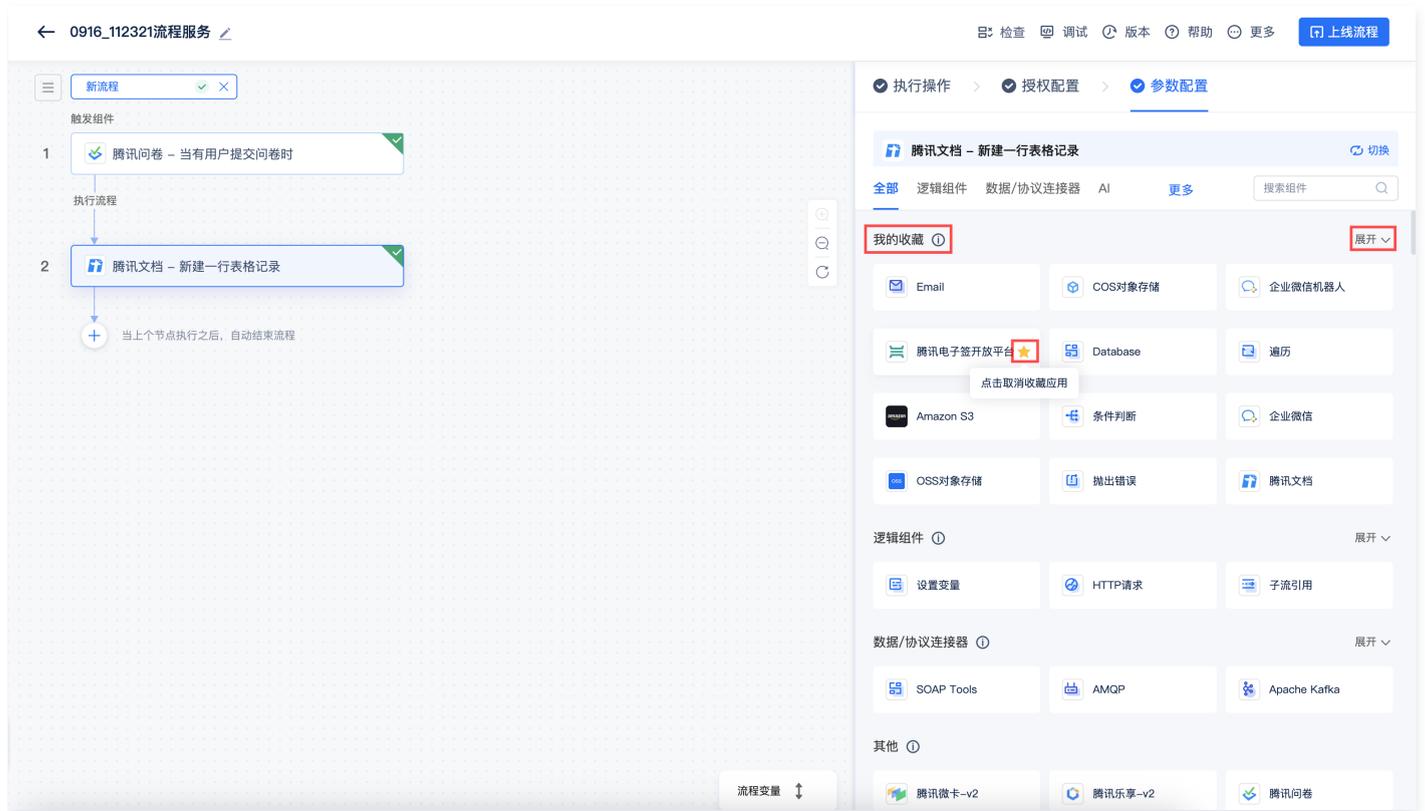
SecretKey \*

文本

需拥有企业管理员权限, 点击用户中心->高级->restApi->添加应用, 即可看到

## 我的收藏

收藏之后，将置顶展示，最新收藏的展示在最前面。



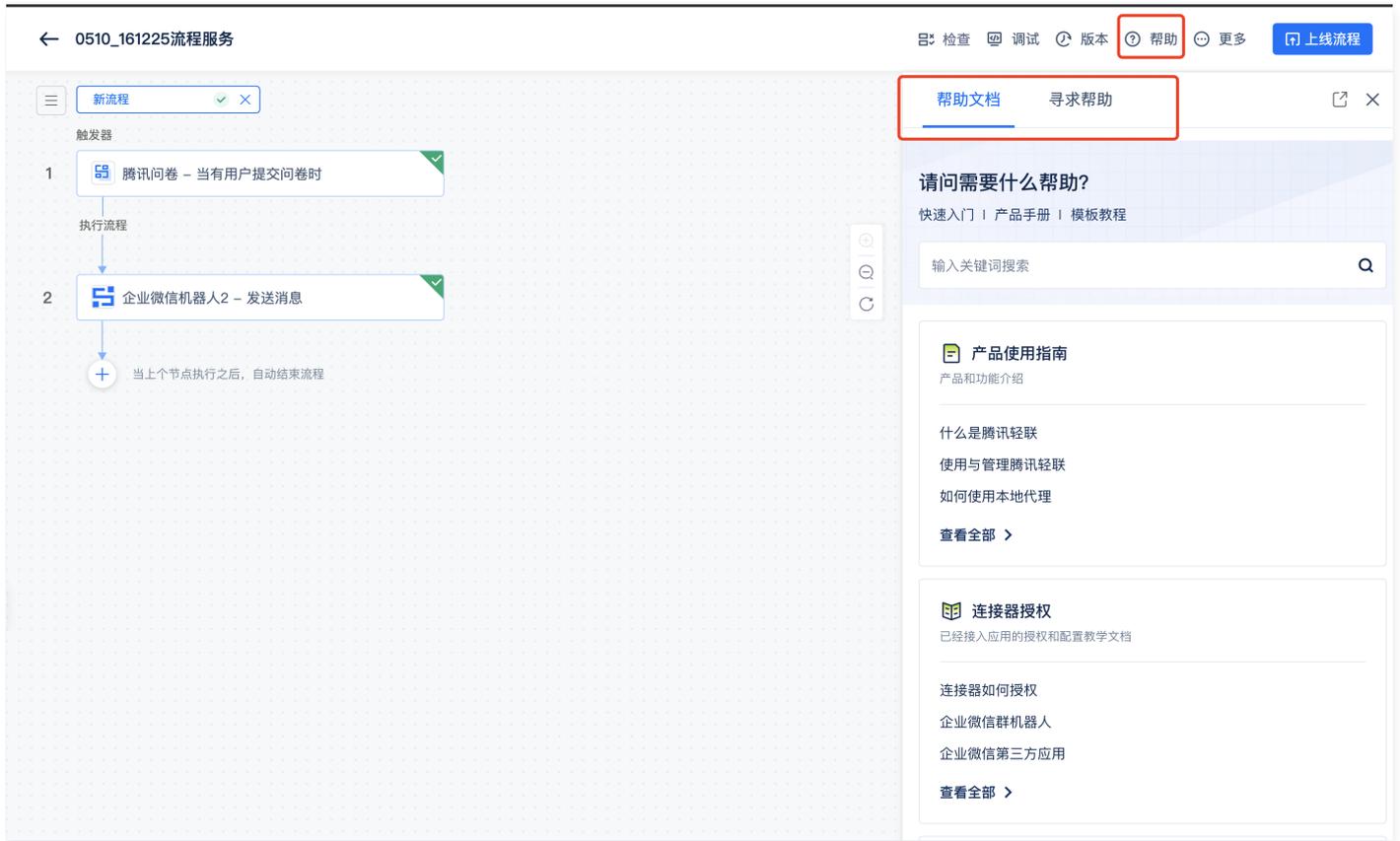
## 版本

每一个流程都有对应的版本号，当一个流程发布后处于运行中状态是不支持修改的，您可以通过恢复此版本复制一个流程对应一个新的版本号，您可以在新的版本中进行修改和调试，且不影响之前已经运行中的版本，便于迭代管理。



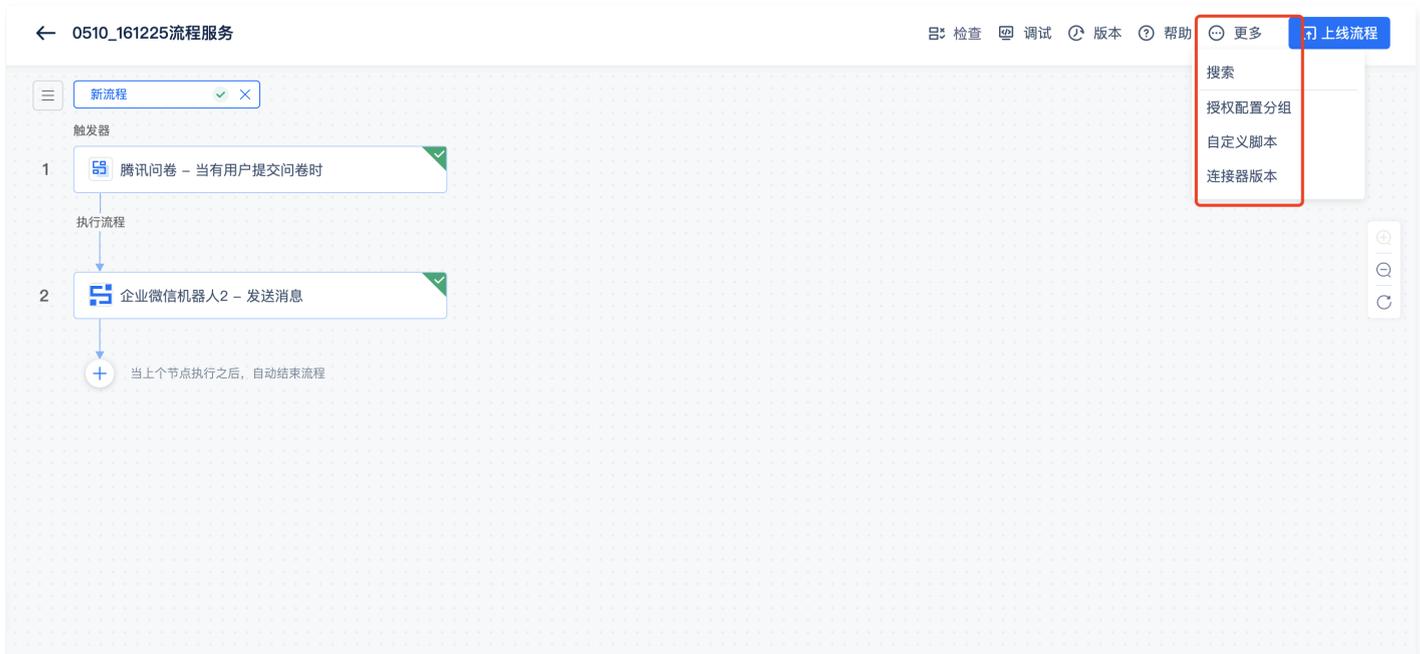
## 帮助

帮助中心提供产品培训视频、产品手册、模板教程，和联系人工客服入口。



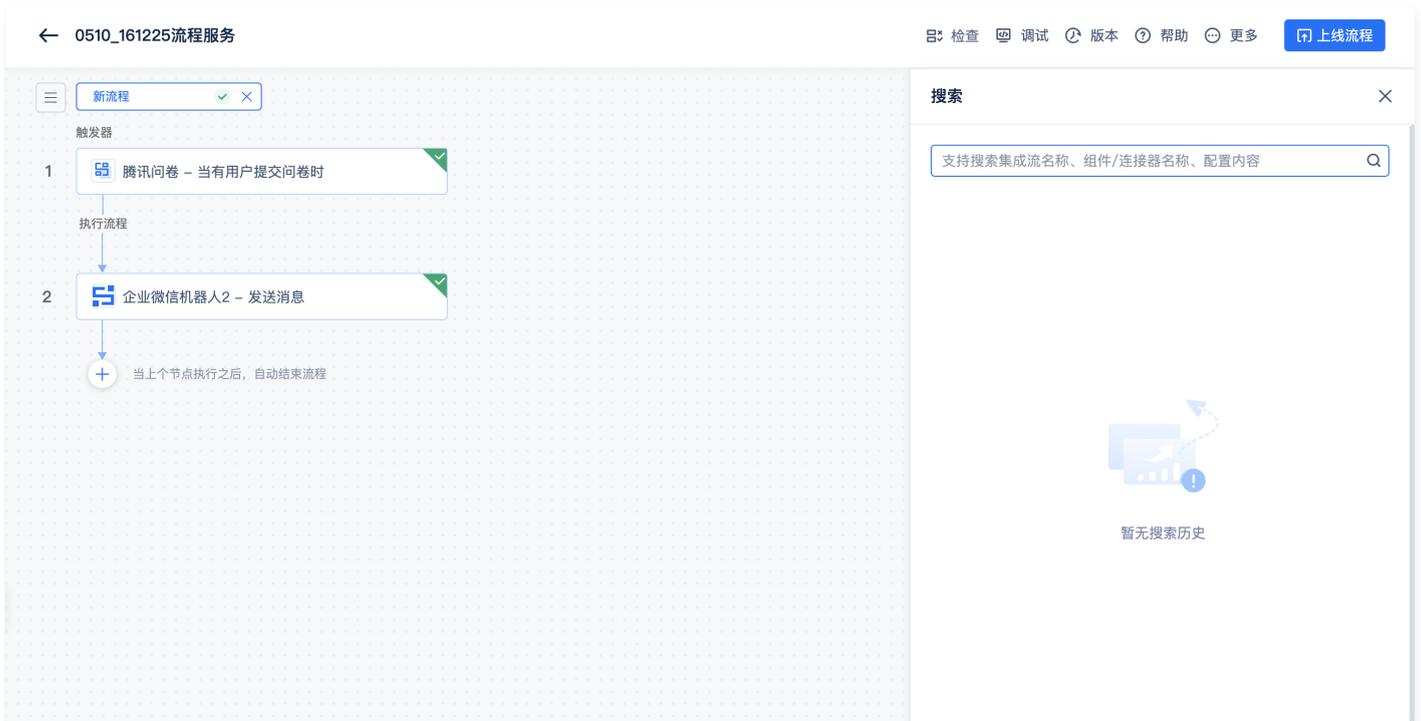
## 更多

更多入口内提供了 搜索、授权配置分组、自定义脚本、连接器版本4个功能。



## 搜索

支持搜索集成流名称、组件/连接器名称、配置内容。方便用户进一步快速确定应用内的有效内容及内置字段。



单击搜索，输入对应关键字。此处以“企业”字段为例。



列表会展示“企业”相关的搜索结果，单击结果可以快速定位到该组件/连接器在流程中的位置。

### 授权配置分组

授权配置分组支持切换配置分组，便于客户在预览和调试下采用设置的分组来进行调试。

### 连接器版本

对于选择的连接器，若连接器存在多版本，可在右上方的更多 > 连接器版本中自行切换所需版本。

**说明：**  
对应版本后若存在红色感叹号符号，代表目前所选版本非最新版本。

### 常见问题

#### 流程何时开始运行？

上线后，流程就开始运行了，当有新的问卷提交时，流程才会触发。

#### 问卷内容修改了，该怎么办？

如果修改了题目描述，流程是不需要修改的；如果新增了题目，此时也不需要修改流程，但如果文档中也希望写入新增的题目，那么需要修改流程。

## 流程运行的数据可以看到吗？

可以在运行日志中查看流程运行详情。

# 逻辑组件使用指南

## 删除变量（Remove Variable）

最近更新时间：2024-08-01 15:33:01

### 简介

删除变量（Remove Variable）组件与配置变量（Set Variable）组件的作用相反，Remove Variable 组件的作用是删除 message 中的一个指定变量。

### 操作配置

#### 参数配置

参数	数据类型	描述	是否必填	默认值
变量名称	string	要移除的变量名称	是	无

#### 配置界面

#### 配置

删除变量 切换组件 ▶

配置内容

变量名 \*

 请输入变量名

变量的名称

### 输出

输出的 message 中，不再包括删除的变量。组件输出的 message 信息如下：

message 属性	值
payload	继承上个组件的 payload。
error	执行成功后，error 为空；执行失败后，error 为 dict 类型，包含“Code”和“Description”字段：“Code”字段表示错误类型，“Description”字段表示错误具体信息。
attribute	继承上个组件的 attribute 信息。
variable	上个组件的 variable 信息中去除删掉的变量。

### 数据预览

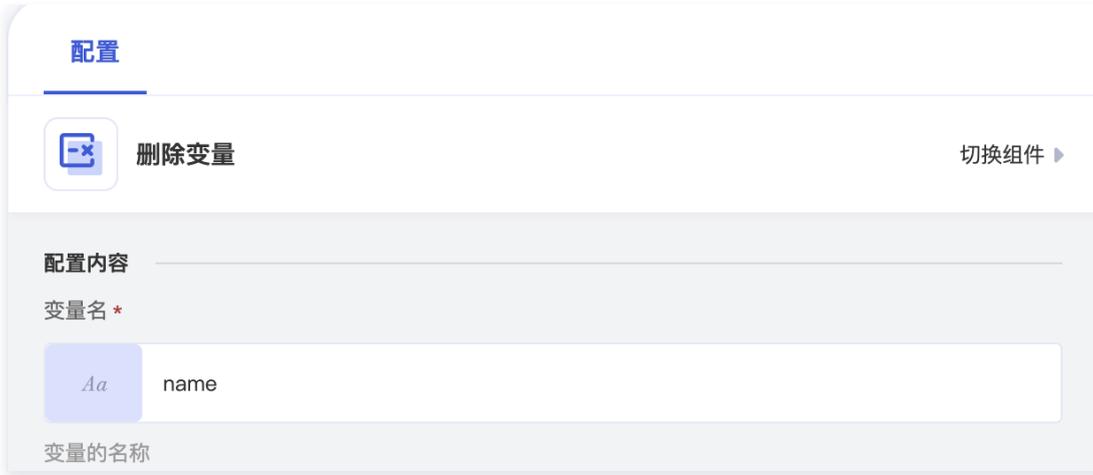
无

### 案例

1. 添加 Remove Variable 组件。



2. 填入要移除的变量名称。



# 子流引用 (Flow Reference)

最近更新时间: 2024-08-01 15:33:01

## 简介

子流引用 (Flow Reference) 组件用来引用当前应用中的其他集成流。与异步处理 (Async) 不同, 子流引用是一个同步过程, 当引用的集成流执行完成后, 才会继续执行下一个动作, 并且子流引用中的子流执行完成后, message 会传递到主流中, 下一个节点基于该子流的 message 继续执行。

当子流引用中的子流包含触发器 (Trigger) 节点时, 如果该子流的执行由子流引用组件触发, 那么该子流的触发器节点不会执行, 即该子流从第二个节点开始执行。

## 操作说明

### 参数配置

参数	数据类型	描述	是否必填	默认值
集成流	string	集成流名称。可以选择同项目内其他应用共享的流	是	无

### 配置界面



### 数据预览

无。

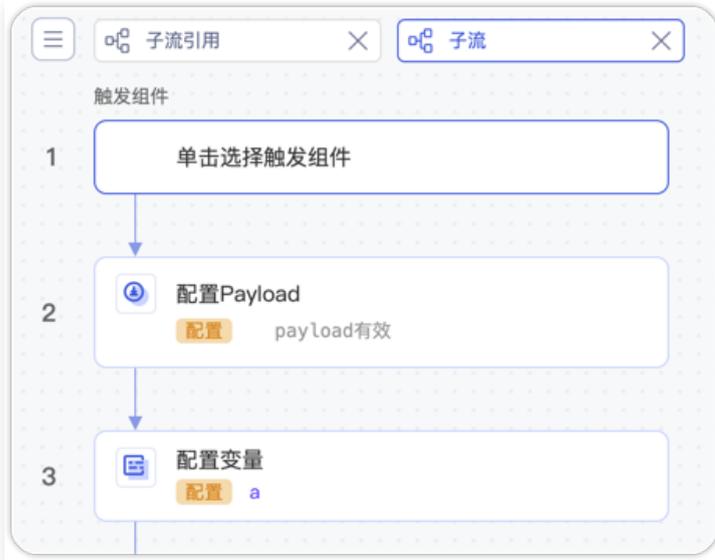
### 输出

message 属性	值
payload	子流最后一个节点输出的 payload
error	执行成功后, error 为空
attribute	子流输出的 attribute
variable	子流所有设置过的 variable

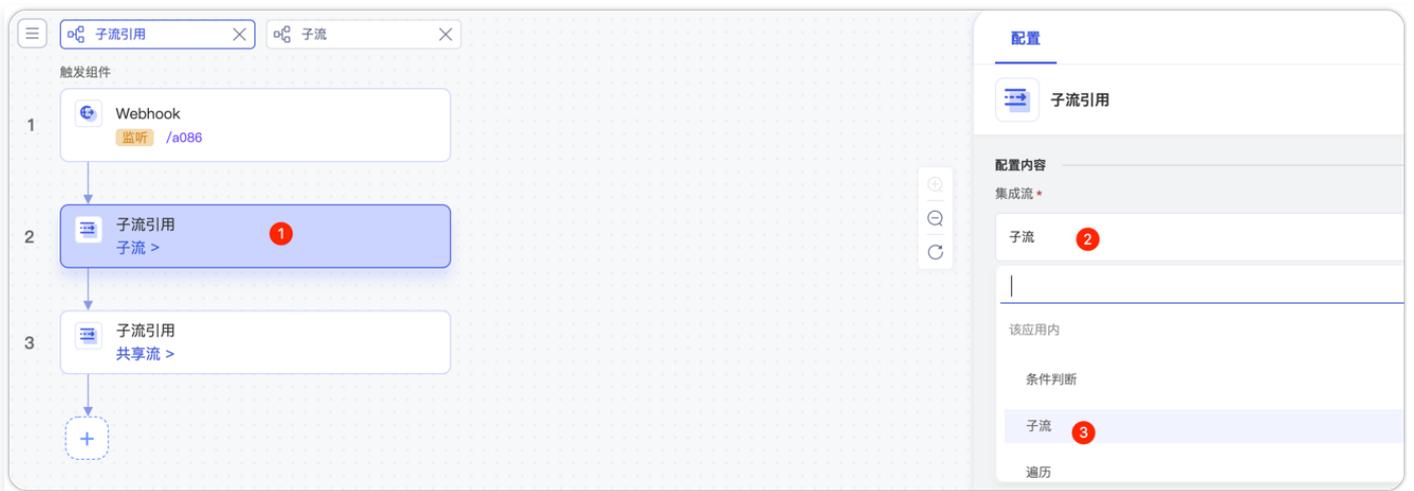
### 案例

## 同一应用内的集成流引用

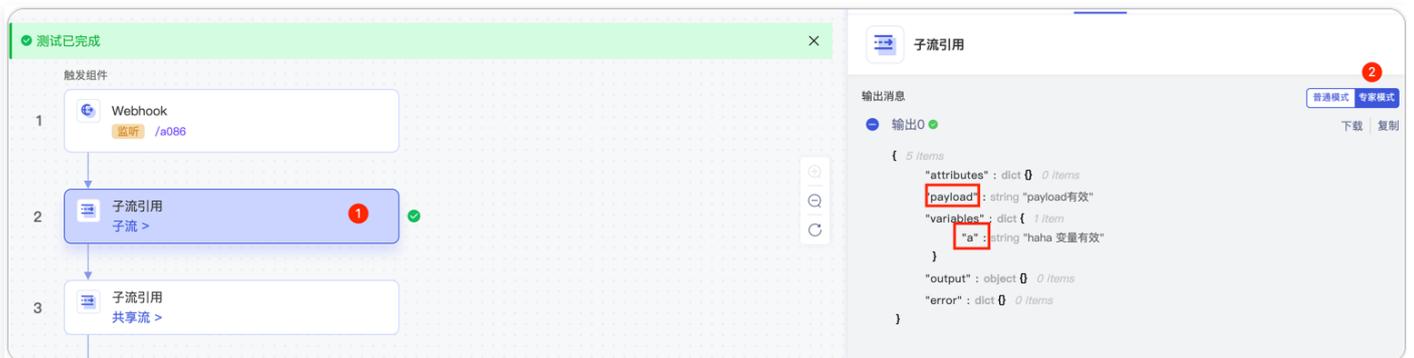
1. 新建一条集成流，名称为“子流”。子流中增加配置 Payload 组件，输出“payload”；增加配置变量组件，设置变量“a”。



2. 添加 Flow Reference 组件，在下拉框中选择“子流”。



3. 单元测试完成后，单击子流引用组件，选择专家模式查看输出。可以看到子流中设置的 payload 和变量“a”被传递到了当前流。



## 项目内不同应用之间的集成流引用

## 集成流引用方式

同一个项目内多个应用可能同时用到某一个通用功能的集成流，例如“用户登录鉴权”。多个应用都要使用“用户登录鉴权”的集成流功能时，通常有三种方式：

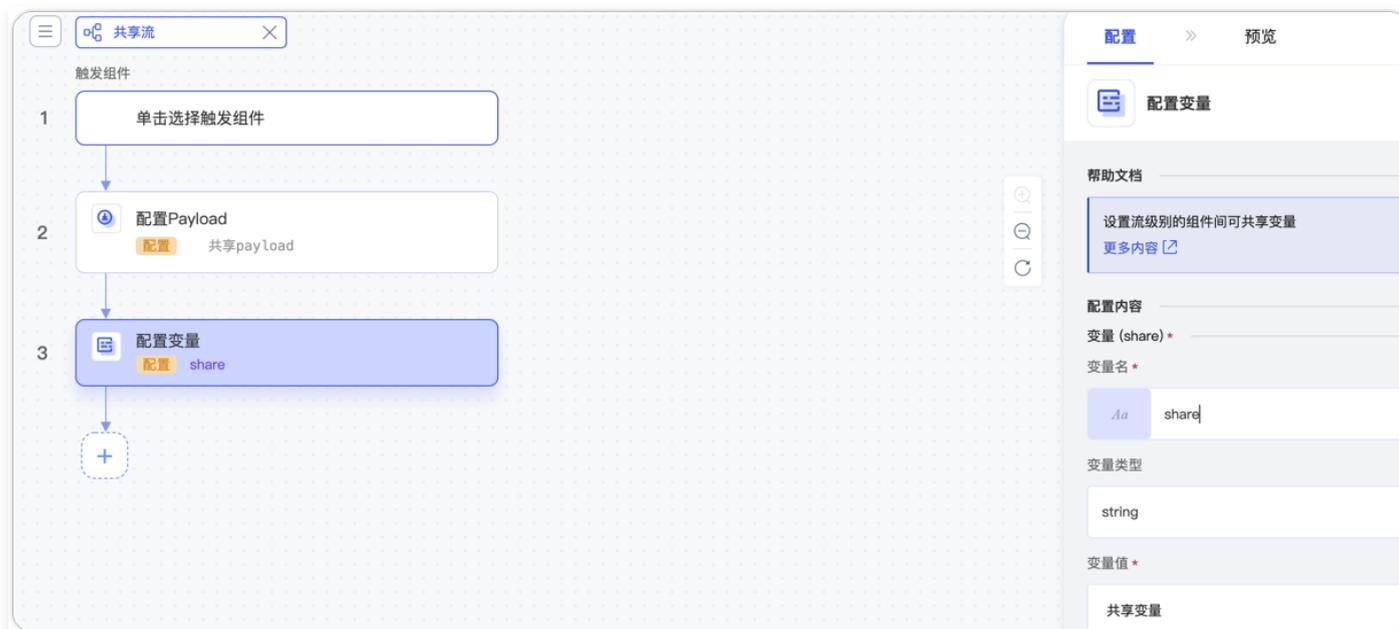
使用方式	优缺点
每个应用都编辑或复制一个“用户登录鉴权”的集成流	存在大量重复集成流，维护不便
其中某个应用以 HTTP 接口的方式提供公共接口供其他应用调用	有 HTTP 调用的消耗，且调试维护不便
使用子流引用的跨应用流引用	跟应用内的流应用体验类似，只需要两个应用同时运行即可

## 使用子流引用的跨应用流引用

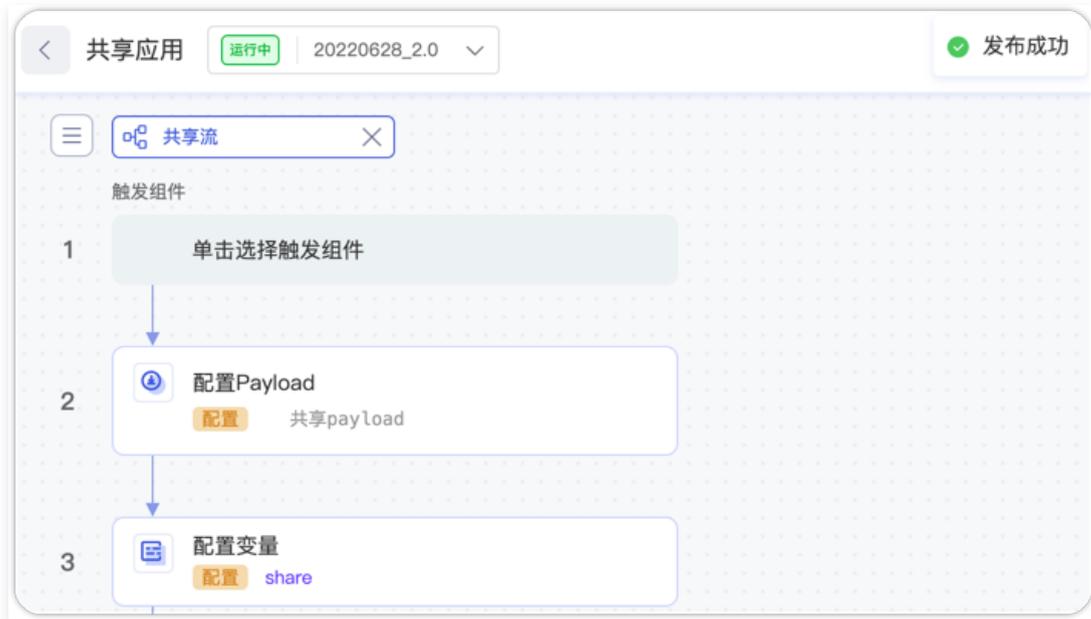
1. 建立共享应用，该应用内提供通用功能的集成流（名称“共享流”），并设置为共享。



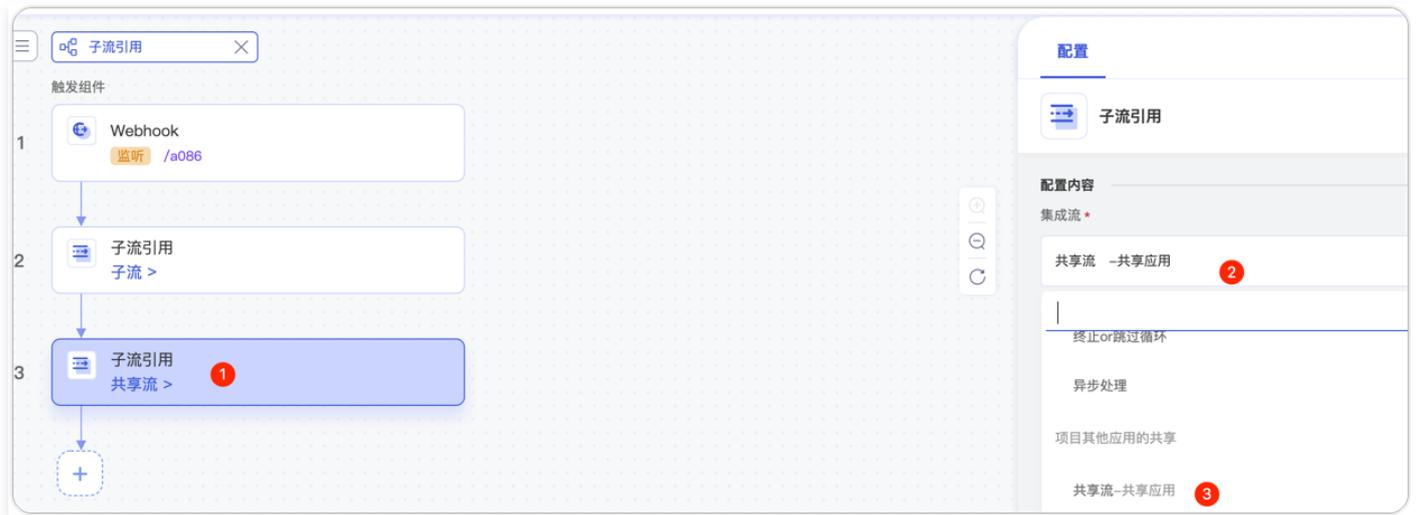
2. 在共享流中设置 payload 和变量 share。



3. 发布共享应用，此时“共享流”的集成流就可以在项目内其他应用的子流引用列表中选择并使用了。同时启动“应用测试”，这样其他应用内引用“共享流”之后也可以进行测试。



4. 在项目内另一个应用编辑集成流，添加子流引用组件，选择引用项目其他应用（共享应用）的“共享流”。



5. 编辑完成执行单元测试，查看子流引用（共享流）的节点输出，切换专家模式，可以看到共享流中设置的 payload 和 变量 share 被传递到了当前流。

The screenshot displays the Tencent Cloud workflow editor interface. On the left, a workflow diagram shows three steps: 1. Webhook (监听 /a086), 2. 子流引用 (子流 >), and 3. 子流引用 (共享流 >). A green banner at the top indicates '测试已完成' (Test completed). On the right, the '输出信息' (Output Message) panel shows the following JSON output:

```
{
  "attributes": dict 0 / 0 Items
  "payload": string "共享payload"
  "variables": dict 1 / 1 Item
    "share": string "共享变量"
}
"output": object 0 / 0 Items
"error": dict 0 / 0 Items
```

# 定时任务 (Scheduler)

最近更新时间: 2024-08-01 15:33:01

## 简介

定时任务 (Scheduler) 作为触发器, 用于按设定的规则定时触发一条流。图形化 Scheduler 支持以下三种触发模式:

- 单次触发: 支持选择多个指定的时间点触发。
- 周期触发: 按照固定的周期触发。
- Cron 表达式: 配置中包含一条或多条 Cron 规则, 当任意一条 Cron 规则与当前时间匹配时, 该定时器所在的集成流将被触发。

## 操作配置

### 参数配置

参数	数据类型	描述	是否必填	默认值
触发模式	Int	设置触发模式: 单次触发, 周期触发, Cron 表达式	是	0 (Cron 表达式模式)
Cron 表达式	string	设置触发规则, 例如每分钟触发1次等	是	无
时区	string	指定时区	是	亚洲/北京 UTC+08:00
仅在上次任务执行完后触发	bool	若勾选则仅在上次任务执行完后触发	否	false

Scheduler 包含一条或多条 Cron 规则, 当存在多条规则时, 使用 `\r` 分隔, Cron 表达式的配置规则如下:

参数	含义	取值范围
seconds	秒	0 - 59
minutes	分钟	0 - 59
hours	小时	0 - 23
days	日期 (选填, 默认每天)	1 - 31
months	月份 (选填, 默认每月)	1 - 12
weekdays	星期几 (选填, 默认不指定)	1 - 7
years	年份 (选填, 默认每年)	1970 - 2099

在配置 Cron 表达式时, 可使用以下运算符:

- \* 表示任意匹配, 例如 `hours="*"` 表示每个小时。
- - 表示范围, 例如 `weekdays="1-5"` 表示周一至周五。
- , 表示并列, 例如 `months="1,3,5,7,8,10,12"` 表示大月。
- / 表示增量, 例如 `hours="8/2"`, 表示8点开始, 每两个小时。
- L 表示最后一个, 例如 `weekdays="6L"`, 表示当月最后一个周六。
- ? 表示不指定。这里有一个约束, 即年月日和星期几中至少要有个为不指定。这是为了避免冲突, 例如同时指定2020年2月20日 (原本周四) 和周三。默认星期几是不指定的。

## 配置界面

配置 >> 预览

定时任务
切换组件 ▶

帮助文档

用于按设定的规则定时触发一条流  
[更多内容](#)

配置内容

配置模式 \*

周期触发

单次触发

Cron表达式

触发周期 \*

每


分钟

触发

高级配置

时区 \*

亚洲/北京 UTC+08:00
▼

仅在上次任务执行完后触发

## 输出

Scheduler 作为 trigger 组件，是集成流的第一个组件，Scheduler 会生成一个空的 message 消息，触发集成流的运行。组件输出的 message 信息如下：

message 属性	值
payload	空，没有值存在
error	<ul style="list-style-type: none"> <li>• 执行成功后，error 为空</li> <li>• 执行失败后，error 为 dict 类型，包含“Code”和“Description”字段：“Code”字段表示错误类型，“Description”字段表示错误具体信息</li> </ul>
attribute	空，没有值存在
variable	空，没有值存

## 数据预览

输出预览

[数据预览](#) Schema预览

✔ 预览成功, 内容如下 [下载](#)

```

{ 1 item
  " payload " : dict { 2 items
    " scheduledTime " : string "2022-06-28 10:15:00"
    " scheduledTimeStamp " : int 1656382500
  }
}
    
```

## 案例

### 周期触发模式

每30秒触发一次。

配置 >> 预览

定时任务 [切换组件](#)

帮助文档

用于按设定的规则定时触发一条流  
[更多内容](#)

配置内容

配置模式 \*

**周期触发** 单次触发 Cron表达式

触发周期 \*

每 30 秒 触发

高级配置

时区 \*

亚洲/北京 UTC+08:00

仅在上次任务执行完后触发

### 单次触发模式

2023年1月1日0时0分0秒触发一次。

配置 >> 预览

定时任务
切换组件 ▶

帮助文档

用于按设定的规则定时触发一条流

[更多内容](#)

配置内容

配置模式 \*

周期触发
单次触发
Cron表达式

触发时间 \*

2023-01-01 00:00:00

+ 添加

高级配置

时区 \*

亚洲/北京 UTC+08:00
▼

仅在上次任务执行完后触发

## Cron 表达式模式

每5分钟触发一次。

配置 >> 预览

定时任务
切换组件 ▶

帮助文档

用于按设定的规则定时触发一条流

[更多内容](#)

配置内容

配置模式 \*

周期触发
单次触发
Cron表达式

Cron表达式 \*

0 0/5 \* \* \* ?

[访问更多Cron信息](#)

指定cron表达式,由空格分割,主要结构为: 秒 分 小时 日期 月份 星期 (年, 可选)

高级配置

时区 \*

亚洲/北京 UTC+08:00
▼

仅在上次任务执行完后触发

# 并行分支 ( Scatter Gather )

最近更新时间：2024-08-01 15:33:01

## 简介

并行分支 ( Scatter Gather ) 支持并行执行多个任务。该组件中，可以添加多个分支，每个分支中都可以配置子流以独立执行任务。

## 操作配置

### 参数配置

参数	数据类型	描述	是否必填	默认值
最大并行数	int	并行执行的任务数，取值范围为2~8。实际并行数取分支数和最大并行数之间较小值。	是	4
根信息	string	根信息是一个变量，这里填入变量名称，根信息中保存了主流的 Message 信息。msg.vars.get('#根信息名称#').payload 即可访问主流的 payload 数据。当使用默认值 rootMessage 时，使用 msg.vars.get('rootMessage').payload 即可在子流中访问主流的 payload 数据。	是	rootMessage

### 配置界面

The screenshot displays the configuration interface for the Scatter Gather component. On the left, a flowchart shows the component at step 4, which branches into two paths: '分支1' (Branch 1) at step 5 and '分支2' (Branch 2) at step 7. Each branch contains a '配置Payload' (Configure Payload) component. The right pane shows the configuration panel for the '并行分支' component, including a '帮助文档' (Help Document) section, a '配置内容' (Configuration Content) section, and a '基本配置' (Basic Configuration) section where '最大并行数' (Maximum Parallelism) is set to 2. The '高级配置' (Advanced Configuration) section shows the '根信息' (Root Message) field set to 'rootMessage'.

### 数据预览

无。

### 输入到子流中的 Message

Message 属性	值
payload	继承主流 Message 中的 payload 信息。
error	空。

attribute	继承主流 Message 中的 attribute 信息。
variable	集成主流的变量。

## 输出

并行分支的输出结果中，不会包含处理逻辑中使用的 variable 变量，最终输出的只有 payload 里的数据。

输出的 payload 是 dict 类型变量，里边汇总了各分支处理的结果。

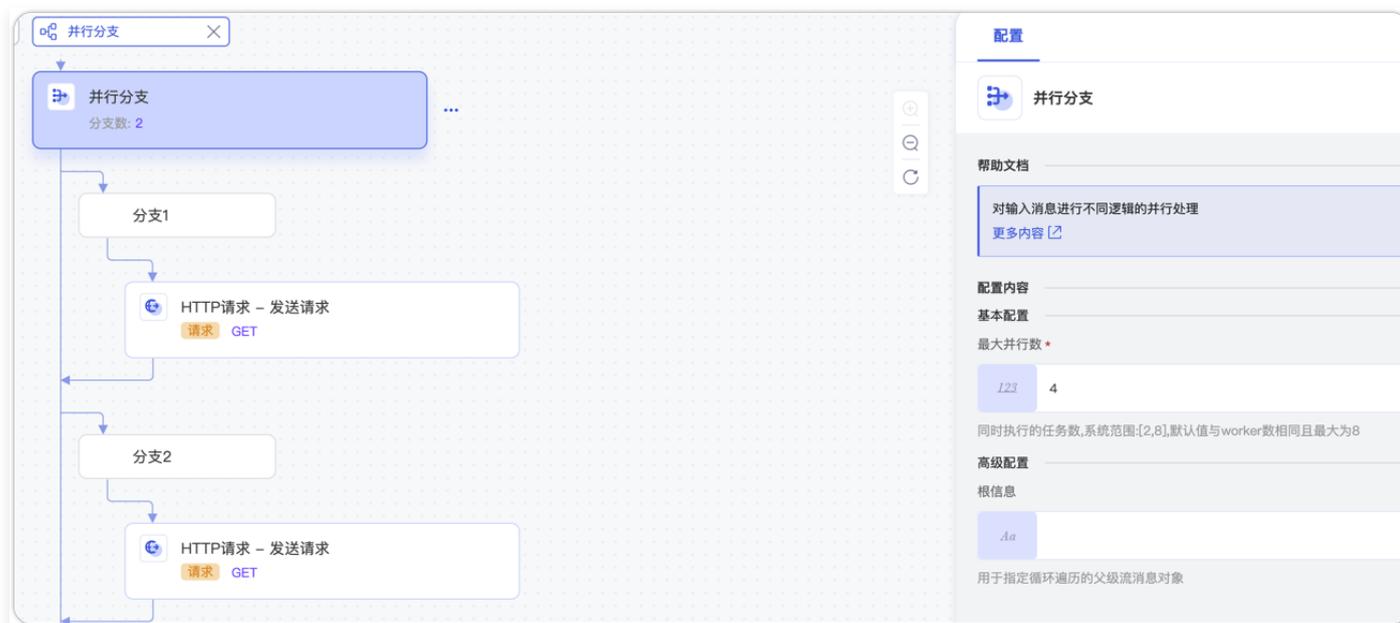
组件输出的 Message 信息如下：

Message 属性	值
payload	dict 类型，key 是分支的编号，从"1"开始，value 是分支执行完成后的结果（最后一个组件输出的 payload）。
error	<ul style="list-style-type: none"> <li>执行成功后，error 为空。</li> <li>执行失败后，error 为 dict 类型，包含“Code”和“Description”字段：“Code”字段表示错误类型，“Description”字段表示错误具体信息。</li> </ul>
attribute	不改变输入 attribute。
variable	不改变输入变量。

## 案例

当需要并行执行不同的任务时，使用并行分支比较合适。例如：要根据用户订单数据，去查询客户信息及产品信息，可以配置两个分支，一个执行客户信息查询，一个执行产品信息查询。

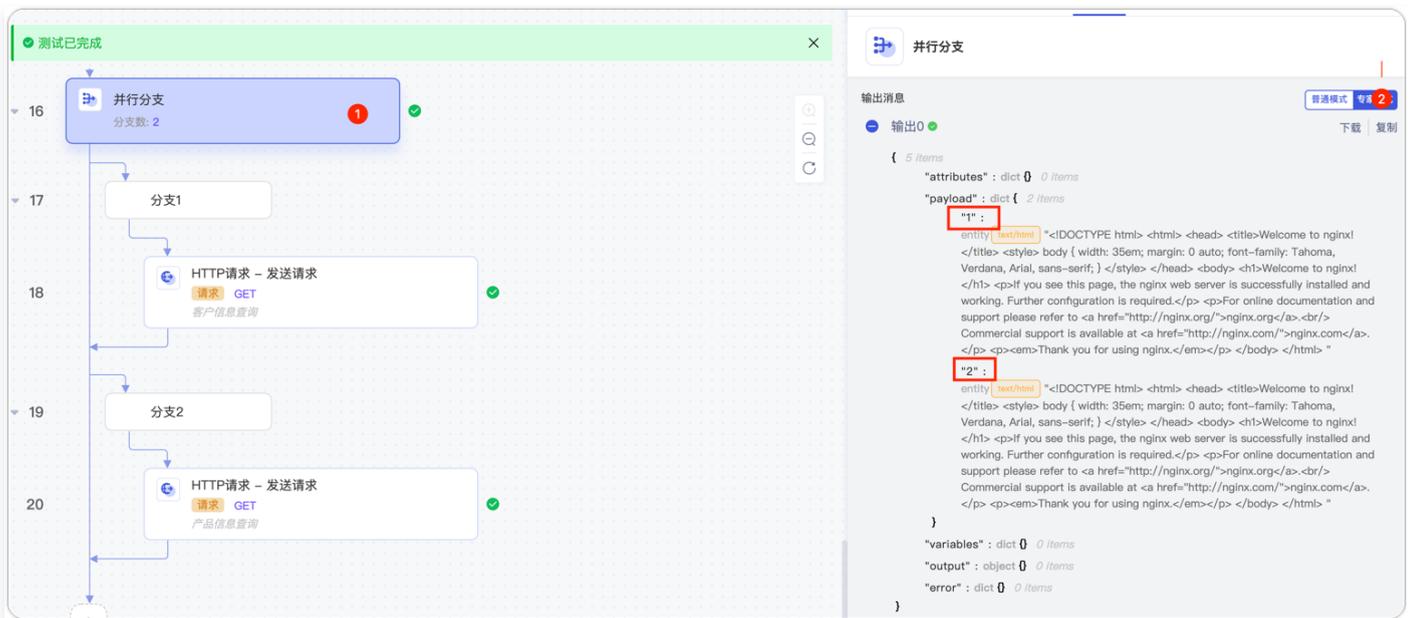
1. 添加并行分支组件，添加2个分支，配置默认即可。



2. 第一个分支配置客户信息查询，第二个分支配置产品信息查询，这里用两个简单的 HTTP 请求模拟。



3. 执行完成后查看并行分支的输出，切换到专家模式，可以看到 Payload 为包含2个 key 的字典。  
key "1"代表分支1的结果，即查询到的客户信息；key "2"代表分支2的结果，即查询到的产品信息。



# 并行遍历 (Parallel Foreach)

最近更新时间: 2024-11-27 09:40:32

## 简介

并行遍历 (Parallel Foreach) 用于并行执行任务，对数据集上的元素并行执行相同的处理逻辑。实际并行数取当前剩余元素数与设置最大并行度之间较小值。

并行遍历的子流可以只读访问主流的变量和其他组件输出，但子流所做的修改均是不影响主流。

处理完成后，每个元素的处理结果按照原始顺序输出到 message 的 payload 中。并行遍历一般用于批量数据处理的场景，例如批量查询、批量导入数据等。

## 操作配置

### 参数配置

参数	数据类型	描述	是否必填	默认值
数据集	string、list、dict、int	待遍历的数据。 <ul style="list-style-type: none"><li>当类型为 string 时，遍历字符串的每个字符。</li><li>当类型为 list 时，遍历 list 的每个元素。</li><li>当类型为 dict 时，遍历 dict 中的 value。</li><li>当数据集类型为int时，例如3，实际遍历的数据集为[0,1,2]。</li></ul>	是	无
最大并行数	int	并行执行的任务数，取值范围为2 - 8。	是	4
计数器	string	计数器是一个变量，该变量存储了当前的迭代次数，从0开始，这里填入变量名称，msg.vars.get('#计数器变量#')即可使用。 例如：当计数器变量使用默认值 counter 时，第1次循环，msg.vars.get('counter')值为0，第2次循环，msg.vars.get('counter')值为1。	否	counter
根信息	string	根信息同样是一个变量，这里填入变量名称，根信息中保存了主流的消息信息。msg.vars.get('#根信息名称#').payload 即可访问主流的 payload 数据。当使用默认值 rootMessage 时，使用 msg.vars.get('rootMessage').payload 即可在 Parallel Foreach 的子流中访问主流的 payload 数据。	否	rootMessage
错误时终止	bool	若子任务发生错误，则会在已发起的子任务执行完成后终止遍历。	否	False

### 配置界面

配置 >> 预览

并行遍历

[切换组件 ▶](#)

帮助文档

对数据集的每个元素进行相同逻辑的并行处理

[更多内容](#)

配置内容

基本配置

数据集 \*

6

循环遍历的数据集对象

最大并行数 \*

123 4

+  
-

同时执行的任务数,系统范围:[2,8],默认为4

高级配置

循环变量名

Aa

循环遍历时用于统计次数（从0开始）的变量名，默认是""

根变量名

Aa rootMsg

循环遍历时指向父级流消息的变量名，默认是""

错误时终止

bool False

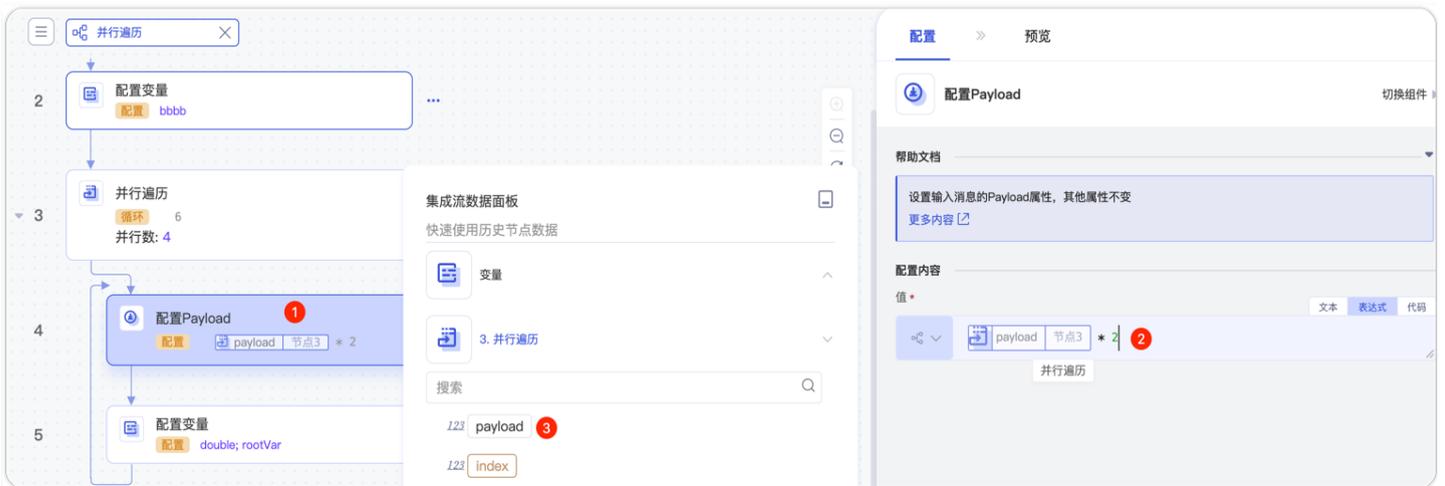
若子任务发生错误，则会在已发起的子任务执行完成后终止遍历

### 数据预览

预览字段	数据类型	描述
payload	any	每次遍历的输入值，属于数据集的其中一个元素。
index	int	每次遍历的位置，代表当前输入值在数据集中的下标位置，从0开始计数。



数据预览的内容仅为子流可见，子流中的组件可以直接使用遍历组件的 Payload 和 index，如图所示：



### 输入到子流中的 message

message 属性	值
payload	数据集中的元素。例如：待迭代的数据集为[1,2,3]，第一次循环，子流中 payload 的数据为1，第二次循环为2。当迭代的数据集为 dict 类型 {"key":"value", "key1":"value1"}，第一次循环，子流中的 payload 为 value1，第二次为 value2。

error	空。
attribute	空。
variable	继承主流中的 variable 数据，同时新增两个变量，一个是计数器，一个是根信息，若用户使用默认值，可使用表达式 msg.vars.get('counter')和 msg.vars.get('rootMessage')访问。 若 For Each 中使用了 Set Variable，则在子流执行过程中，新增的变量也会添加到 variable 中。

## 输出

并行遍历的输出结果中，不会包含处理逻辑中使用的 variable 变量，最终输出的只有 payload 里的数据，输出的 payload 是个 list 类型变量，里边包含了原始数据集中每个元素迭代处理的结果，顺序与原始数据集一致。attribute 的值继承自上一个组件。

组件输出的 message 信息如下：

message 属性	值
payload	list 类型，包含了每个数据的处理结果，输出顺序与原始的输入顺序一致。
error	<ul style="list-style-type: none"> <li>执行成功后，error 为空。</li> <li>执行失败后，error 为 dict 类型，包含 “Code” 和 “Description” 字段：“Code” 字段表示错误类型，“Description” 字段表示错误具体信息。</li> </ul>
attribute	继承 Parallel ForEach 上一个组件的 attribute 信息。
variable	继承 Parallel ForEach 上一个组件的 variable 信息。

输出样例如下：

The screenshot displays a workflow configuration in the Tencent Cloud console. The workflow consists of five steps:

- 配置变量 (Configure Variable): Variable name 'bbbb'.
- 并行遍历 (Parallel Loop): Loop count '6', parallelism '4'.
- 配置Payload (Configure Payload): Expression 'payload | 节点3 | \* 2'.
- 配置变量 (Configure Variable): Variable names 'double; rootVar'.

The right-hand panel shows the '输出信息' (Output Message) for the '并行遍历' component. The output is a JSON object with the following structure:

```

{
  "attributes": dict 0 / 0 items
  "payload": list 6 items
    0: int 0
    1: int 2
    2: int 4
    3: int 6
    4: int 8
    5: int 10
  }
  "variables": dict { 1 item
    "bbbb": string "adfasfds"
  }
  "output": object 0 / 0 items
  "error": dict 0 / 0 items
}
    
```

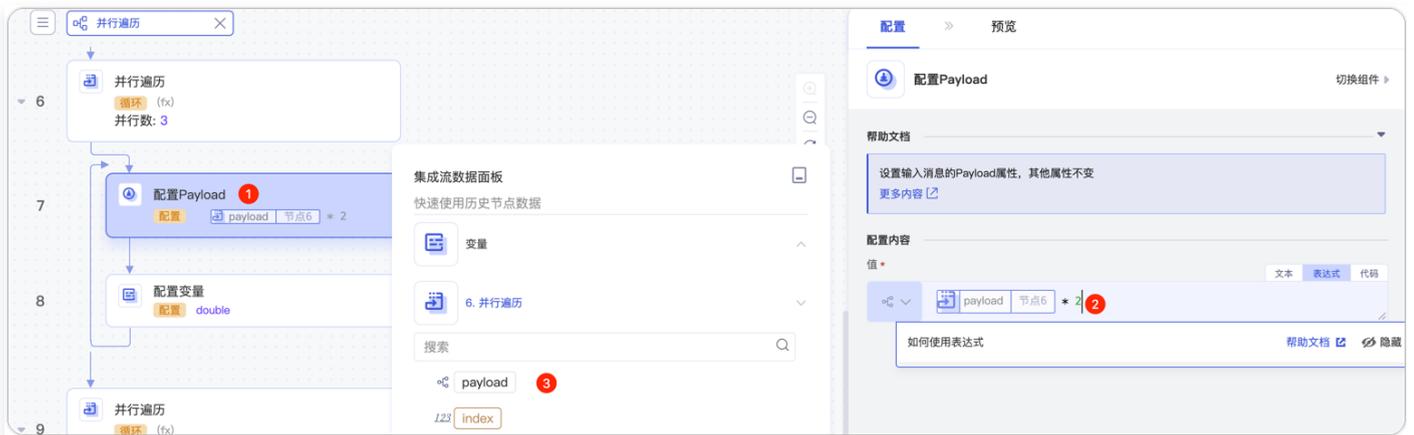
## 案例

本案例中，我们使用并行遍历对列表中的数据统一进行乘2操作，原始的数据集为 [1,2,3,4]。

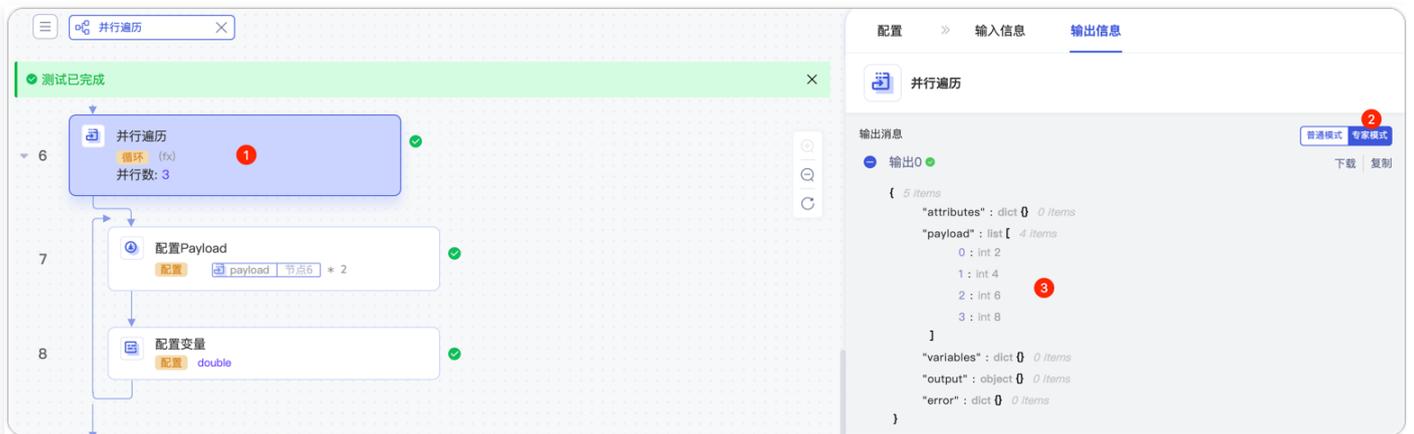
1. 添加并行遍历组件，配置数据集[1,2,3,4]，并设置并行度为3。



2. 在并行遍历中加入配置 Payload 组件，对子流 Payload 执行乘2操作。



3. 输出结果。



# 延时 (Sleep)

最近更新时间: 2024-08-01 15:33:01

## 简介

延时 (Sleep) 组件可以延迟指定时间后, 继续执行集成流。

## 操作配置

### 参数配置

参数	数据类型	描述	是否必填	默认值
延迟时间	int	指定的延迟时间 (单位: 毫秒)	是	1000

## 输出

组件输出的 message 信息如下:

message 属性	值
payload	继承上个组件的 attribute 信息。
error	<ul style="list-style-type: none"><li>执行成功后, error 为空。</li><li>执行失败后, error 为 dict 类型, 包含 “Code” 和 “Description” 字段: “Code” 字段表示错误类型, “Description” 字段表示错误具体信息。</li></ul>
attribute	继承上个组件的 attribute 信息。
variable	继承上个组件的 variable 信息。

## 数据预览

无。

## 案例

1. 添加 “延时组件”。



2. 填入要设置的延迟时间。

配置



延时

切换组件 ▶

配置内容

延迟时间（毫秒） \*

123

1000

有效范围：[1,300000]

# 抛出错误 (Raise Error)

最近更新时间: 2024-08-01 15:33:01

## 简介

抛出错误 (Raise Error) 组件用来抛出异常，中断流的执行。该组件可以单独使用，也可以搭配“错误捕获”组件使用。单独使用时，命中“抛出错误”，集成流会中断执行，返回错误信息。搭配“错误捕获”组件使用时，可以在“捕获错误”中捕获本组件定义的异常，然后执行“捕获错误”中配置的子流。

## 操作指引

### 连接说明

无。

### 参数配置

参数	数据类型	描述	是否必填	默认值
错误类型	string	用户自定义错误类型	是	通用错误
错误描述	string	错误的描述信息	是	无

### 配置

 抛出错误 切换组件 ▶

帮助文档

用于产生自定义异常  
[更多内容](#)

配置内容

错误类型 \*

通用错误

点击 [添加](#) 可创建自定义错误类型

错误描述 \*

Aa ...

错误的描述支持表达式输入

错误类型通过下拉列表选择，可以通过新增按钮添加新的错误类型，项目内所有应用可见。

**配置**

 抛出错误
切换组件 ▶

---

帮助文档 ▼

用于产生自定义异常

[更多内容](#)

配置内容

错误类型 \*

请求失败 ^

请求失败

通用错误

+ 添加

错误的描述,支持表达式输入

## 数据预览

无

## 输出

组件输出的 message 信息如下:

message 属性	值
payload	继承上个组件的 payload。
error	error 为 dict 类型, 包含 “Code” 和 “Description” 字段: <ul style="list-style-type: none"> <li>“Code” 字段表示错误类型, 使用内部编码表示。</li> <li>“Description” 字段表示错误描述。</li> </ul>
attribute	继承上个组件的 attribute 信息。
variable	继承上个组件的 variable 信息。

## 案例

1. 添加 “抛出错误” 组件。
2. 新增错误类型 “请求失败”。

添加错误类型
✕

错误名称

保存
取消

3. 选择错误类型，配置错误描述。

配置

抛出错误

切换组件 ▶

帮助文档 ▼

用于产生自定义异常  
[更多内容](#) ↗

配置内容

错误类型 \*

请求失败
▼

点击 [添加](#) 可创建自定义错误类型

错误描述 \*

Aa

连接超时

错误的描述.支持表达式输入

4. 输出信息。

配置 >> 输入信息 **输出信息**

 抛出错误

输出消息

普通模式 专家模式

— 输出0 

下载 | 复制

```
{ 5 items
  "attributes" : dict {} 0 items
  "payload" : string ""
  "variables" : dict {} 0 items
  "output" : object {} 0 items
  "error" : dict { 2 items
    "code" : string "CUSTOM:Oy4dJpJl2X"
    "desc" : string "连接超时"
  }
}
```

# 数据转换（Transform）

最近更新时间：2024-06-26 11:01:31

## 简介

数据转换（Transform）组件可以对 message 消息进行数据编排和格式转换，支持 payload、attribute 和 variable 的修改。目前产品已经支持在各输入框内，使用 code 或者表达式对变量进行简单转换或复杂处理。

## 操作配置

### 参数配置

参数	数据类型	描述	是否必填	默认值
payload	any	配置的 Payload，同配置 Payload 组件	否	无
attribute	dict	配置的 attribute（用处较少，可忽略）	否	无
variable	dict	配置的变量，同配置变量组件	是	无

### 名词解释：

- **payload**：在轻联产品中，payload 指上一个节点的输出内容的有效部分，例如 HTTP 响应中的实际数据部分。
- **attribute**：响应的附加信息，一般无用。例如"Content-Type"表示响应的内容类型，"Content-Length"表示响应的内容长度等。
- **variable**：在轻联产品中，创建变量和替换变量是最核心逻辑，此处可以创建并设置变量值，供下游节点使用。

## 配置界面

根据需要设置对应的 payload、attributes、variable 的信息。

### 参数配置

数据转换
切换

通过Transform对Payload、Attribute、Variables进行数据的转换  
[更多帮助](#)

Payload

```
1 def dw_process(msg):
2   return
```

Attributes

Variables

预览配置

模拟流程运行时的效果，也可以在后续节点中引用为变量数据

执行预览

数据预览
Schema预览



请点击执行进行预览

## 输出

组件输出的 message 信息如下：

message 属性	值
payload	如果“输出信息”中添加了 payload，输出为 payload 中的执行结果，否则继承上一个组件的payload
error	<ul style="list-style-type: none"> <li>执行成功后，error 为空</li> <li>执行失败后，error 为 dict 类型，包含“Code”和“Description”字段：“Code”字段表示错误类型，“Description”字段表示错误具体信息</li> </ul>
attribute	类型为 dict，如果“输出信息”中添加了 attributes，输出为 attributes 的执行结果，否则继承上一个组件的 attribute
variable	如果“输出信息”中添加了 variables，输出为上一个组件的 variable 加上 transform 中新增的 variable，否则继承上一个组件的 variable

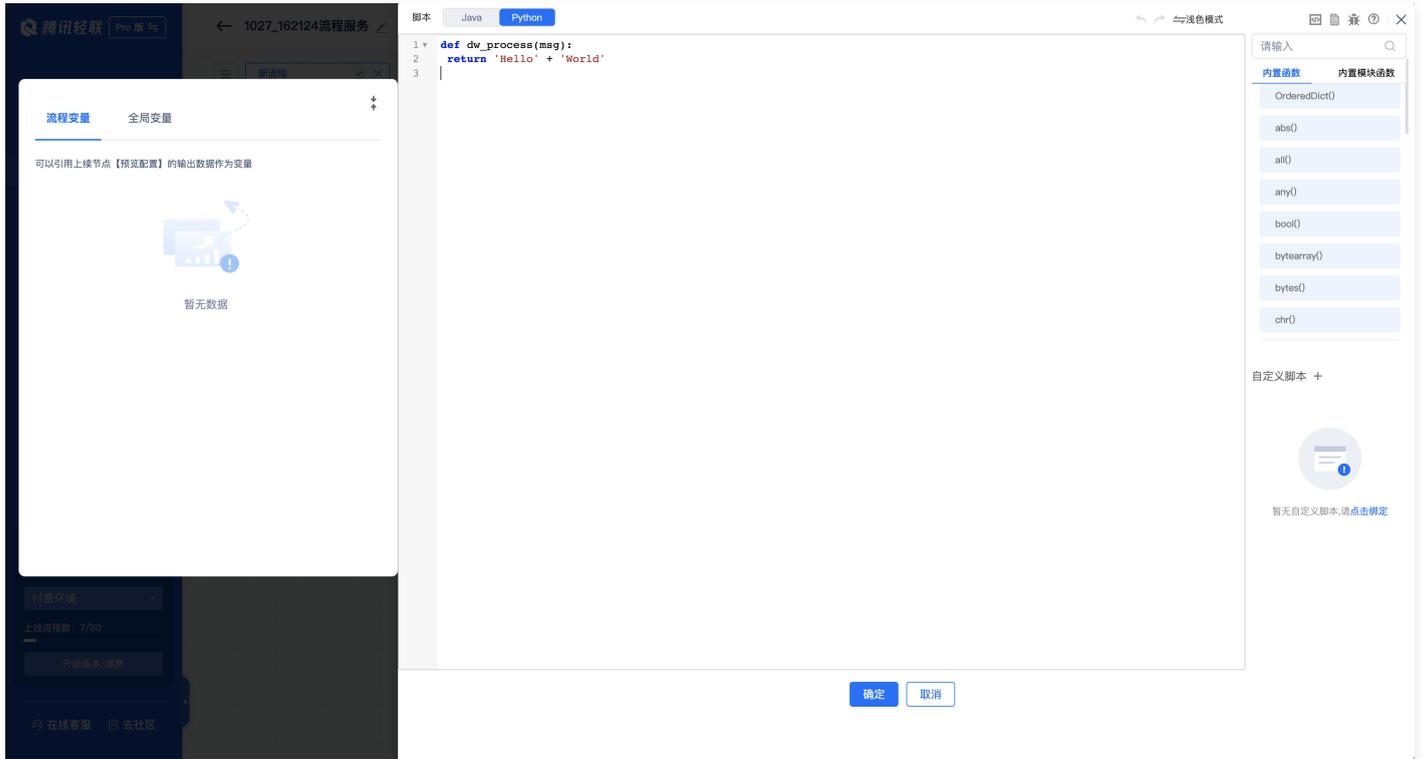
## 案例

### 设置 payload

添加 payload。

**注意：**

只有轻联的 Pro 版本，才支持使用 Python 和 Java。



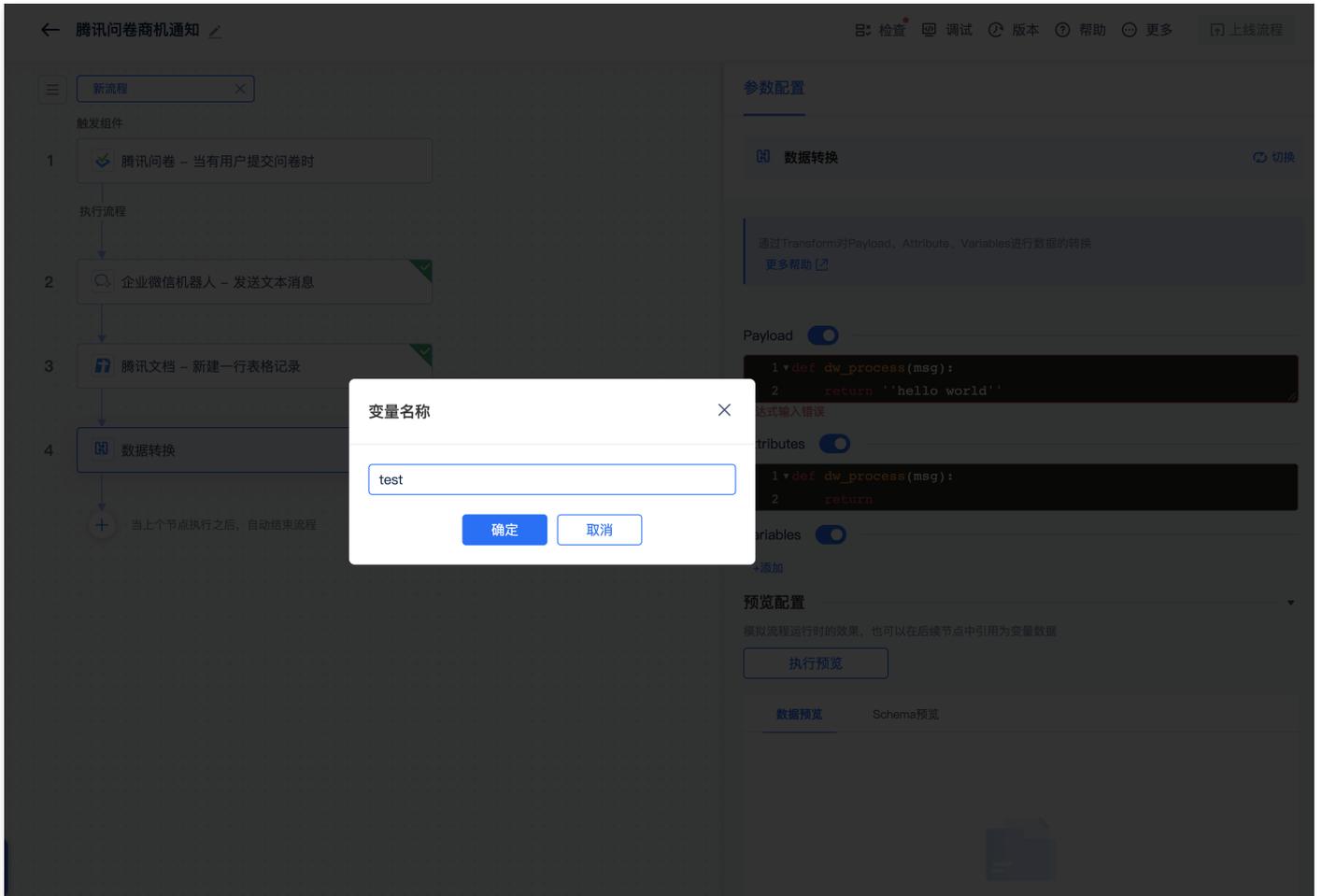
### 设置 attribute

添加 attributes，并编辑 attributes，attributes 的类型为 dict，因此表达式的输出需要保证类型为 dict。



## 设置 variable

1. 添加 variables, “变量名称” 处填入要声明的变量名字。



2. 添加表达式，编辑变量数据。

### 参数配置

数据转换 切换

通过Transform对Payload、Attribute、Variables进行数据的转换  
[更多帮助](#)

**Payload**

```
1 def dw_process(msg):
2     return
```

**Attributes**

```
1 def dw_process(msg):
2     return {"company": "tencent"}
```

**Variables**

test

```
1 def dw_process(msg):
2     return "test"
```

[+添加](#)

# 数据映射 ( Mapping )

最近更新时间：2024-08-01 15:33:01

## ⚠ 注意：

目前产品已经升级，支持将各连接器需要配置的参数渲染到页面中，并且支持用户在输入框内使用 code 或者表达式对变量进行简单转换或复杂处理。

因此数据映射组件已经弃用。

以下为原有组件逻辑：

## 简介

数据映射组件可以对输入和输出信息进行映射。

## 操作配置

### 参数配置

首先配置输入信息、输出信息，然后配置映射逻辑，最后校验是否配置无误。

### 配置界面



## 案例

设置输入和输出信息映射。

参数配置

数据映射
切换

输入信息

- 1 string

逻辑映射

输出信息

- a string

映射逻辑配置

通过表达式配置

查询key信息

参数名	参数...	默认值
key	string	ipaas

输出value信息

参数名	参数...	默认值
新版本	string	qingli...

# 条件重试 (Until Successful)

最近更新时间: 2025-07-03 17:43:01

## 简介

条件重试 (Until Successful) 组件的作用是对其子流执行重试操作。

## 操作指引

根据您创建应用的时间不同，条件重试 (Until Successful) 组件的配置也不相同，具体如下：

配置方法	适用场景	配置入口
<a href="#">方法一</a>	在2022年9月1日前创建的应用	可选择按条件重试
<a href="#">方法二</a>	在2022年9月2日及之后创建的应用	仅在满足条件时重试

## 方法一 (2022年9月1日前创建的应用)

### 参数配置

参数	数据类型	描述	是否必填	默认值
重试条件类型	枚举：无；按成功条件重试；按失败条件重	重试条件的类型	是	无
重试条件	bool	当重试条件类型为按成功条件重试或按失败条件重试时，定义判断的方法	否	-
重试次数	int	重试次数，取值范围1 - 100	是	3
重试时间间隔	int	重试间隔，取值范围1 - 300，单位：秒	是	60

**配置**

条件重试 切换组件 ▶

---

帮助文档 ▼

执行失败后重试，直到最大执行次数或成功

[更多内容](#)

配置内容

重试条件类型 \*

按成功条件重试 ▼

重试条件 \*

`bool` `msg.payload > 10`

设置重试条件

重试次数 \*

`123` 3 +  
-

最大重试次数,取值范围[1,100]

重试时间间隔 \*

`123` 60 +  
-

两次重试的最小时间间隔,取值范围[1,300],单位秒

## 数据预览

无

## 输入到子流中的 message

message 属性	值
payload	继承条件重试上一个组件的 payload
error	空
attribute	继承条件重试上一个组件的 attribute 信息
variable	继承条件重试上一个组件的 variable 信息

## 输出

message 属性	值
payload	继承子流输出的 payload
error	<ul style="list-style-type: none"> <li>执行成功后，error 为空</li> <li>执行失败后，error 为 dict 类型，包含“Code”和“Description”字段：“Code”字段表示错误类型，“Description”字段表示错误描述</li> </ul>
attribute	继承子流的 attribute 信息
variable	继承子流的 variable 信息

## 案例

当需要对某类操作进行重试时，可以使用该组件，例如发送 request 请求，当请求失败时，使用 Until Successful 进行重试。

1. 添加 Until Successful 组件，当服务器返回的 statusCode 不等于200时，重新发送请求，设置重试次数为3，重试间隔为5秒。



2. 在 Until Successful 组件中添加 Request 组件，用于发送 HTTP 请求。



## 方法二（2022年9月2日及之后创建的应用）

### 参数配置

参数	数据类型	描述	是否必填	默认值
重试条件	bool	定义重试的条件，当满足条件时触发重试	否	-
重试次数	int	重试次数，取值范围1 - 100	是	3
重试时间间隔	int	重试间隔，取值范围1 - 300，单位秒	是	60

**配置**

**条件重试** 切换组件 ▾

---

帮助文档 ▾

执行失败后重试，直到最大执行次数或成功  
[更多内容](#)

---

配置内容

重试条件类型 \*

按成功条件重试 ▾

---

重试条件 \*

`bool` `msg.payload > 10`

设置重试条件

重试次数 \*

`123` 3 +  
-

最大重试次数,取值范围[1,100]

重试时间间隔 \*

`123` 60 +  
-

两次重试的最小时间间隔,取值范围[1,300],单位秒

## 数据预览

无

## 输入到子流中的 message

message 属性	值
payload	继承条件重试上一个组件的 payload
error	空
attribute	继承条件重试上一个组件的 attribute 信息
variable	继承条件重试上一个组件的 variable 信息

## 输出

message 属性	值
payload	继承子流输出的 payload
error	<ul style="list-style-type: none"> <li>执行成功后，error 为空</li> <li>执行失败后，error 为 dict 类型，包含“Code”和“Description”字段：“Code”字段表示错误类型，“Description”字段表示错误描述</li> </ul>
attribute	继承子流的 attribute 信息
variable	继承子流的 variable 信息

## 案例

当需要对某类操作进行重试时，可以使用该组件，例如发送 request 请求，当请求失败时，使用 Until Successful 进行重试。

1. 添加 Until Successful 组件，当服务器返回的 statusCode 不等于200时，重新发送请求，设置重试次数为3，重试间隔为5秒。

配置

条件重试
切换组件 ▶

条件重试 (Until Successful) 组件的作用是对其子流执行重试操作。 [更多内容](#)

**重试条件\***

`msg.payload.get('statusCode') != 200`

满足条件时触发重试

**重试次数\***

123

3

+  
-

最大重试次数,取值范围[1,100]

**重试时间间隔\***

123

60

+  
-

两次重试的最小时间间隔,取值范围[1,300],单位秒

2. 在 Until Successful 组件中添加 Request 组件，用于发送 HTTP 请求。



# 终止循环（Break）

最近更新时间：2024-08-01 15:33:01

## 简介

终止循环（Break）组件需要和遍历（For Each）或者条件循环（While）组件搭配使用，用来中断循环，即使序列还未递归完成或者循环条件没有 False，也会停止执行循环语句。当存在嵌套循环时，终止循环组件将跳出自己所在层的循环，并开始执行下一个组件。

## 操作指引

### 连接说明

无。

### 参数配置

无。

### 数据预览

无。

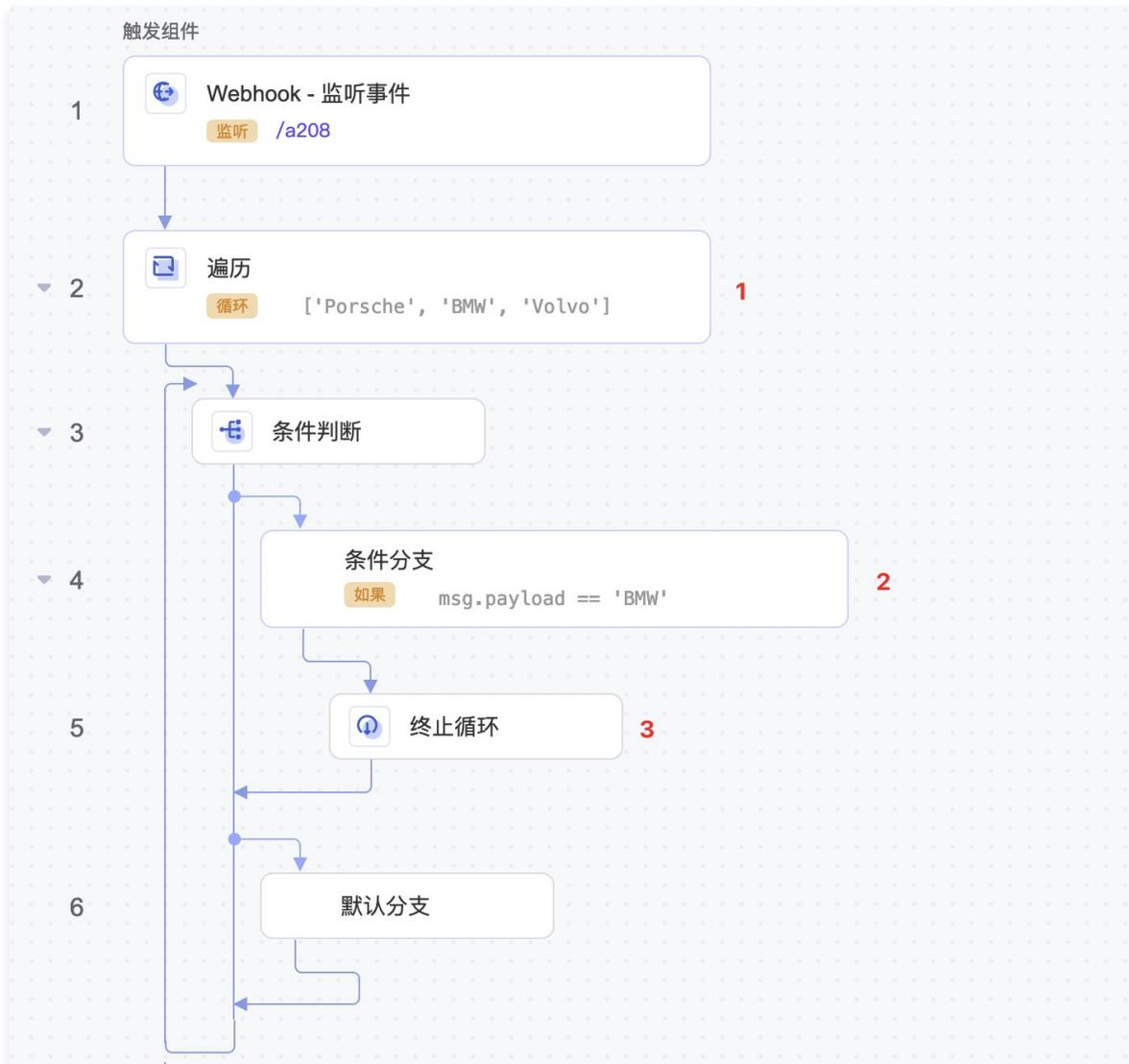
### 输出

组件输出的 message 信息如下：

message 属性	值
payload	<ul style="list-style-type: none"><li>当在“遍历”组件中使用时，终止循环后，payload 中的数据继承自“遍历”的上一个组件的 payload。</li><li>当在“条件循环”组件中使用时，终止循环后，payload 的数据继承自终止循环的上一个组件输出的 payload。</li></ul>
error	空。
attribute	<ul style="list-style-type: none"><li>在“遍历”组件中使用时，终止循环后，attribute 继承自“遍历”的上一个组件输出的 attribute。</li><li>在“条件循环”组件中使用时，终止循环后，attribute 继承自“条件循环”的上一个组件输出的 attribute。</li></ul>
variable	<ul style="list-style-type: none"><li>在“遍历”中使用时，终止循环后，variable 中的变量是“遍历”的上一个组件输出的 variable 变量加上“遍历”子流中声明的 variable 变量。</li><li>在“条件循环”组件中使用时，variable 中的变量是“条件循环”的上一个组件输出的 variable 变量加上“条件循环”子流中声明的 variable 变量。</li></ul>

## 案例

- 添加“遍历”组件，设置要遍历的列表。
- 使用“条件判断”组件，并在“条件分支”中添加判断条件。
- 在“条件分支”子节点中添加“终止循环”组件，当遍历到“BMW”时，跳出遍历流程。



# 跳过循环（Continue）

最近更新时间：2024-08-01 15:33:01

## 简介

跳过循环（Continue）组件类似于终止循环组件，搭配遍历和条件循环组件使用。跳过循环的作用是跳出当前循环，执行下一次循环。

## 操作指引

### 连接说明

无

### 参数配置

无

### 数据预览

无

### 输出

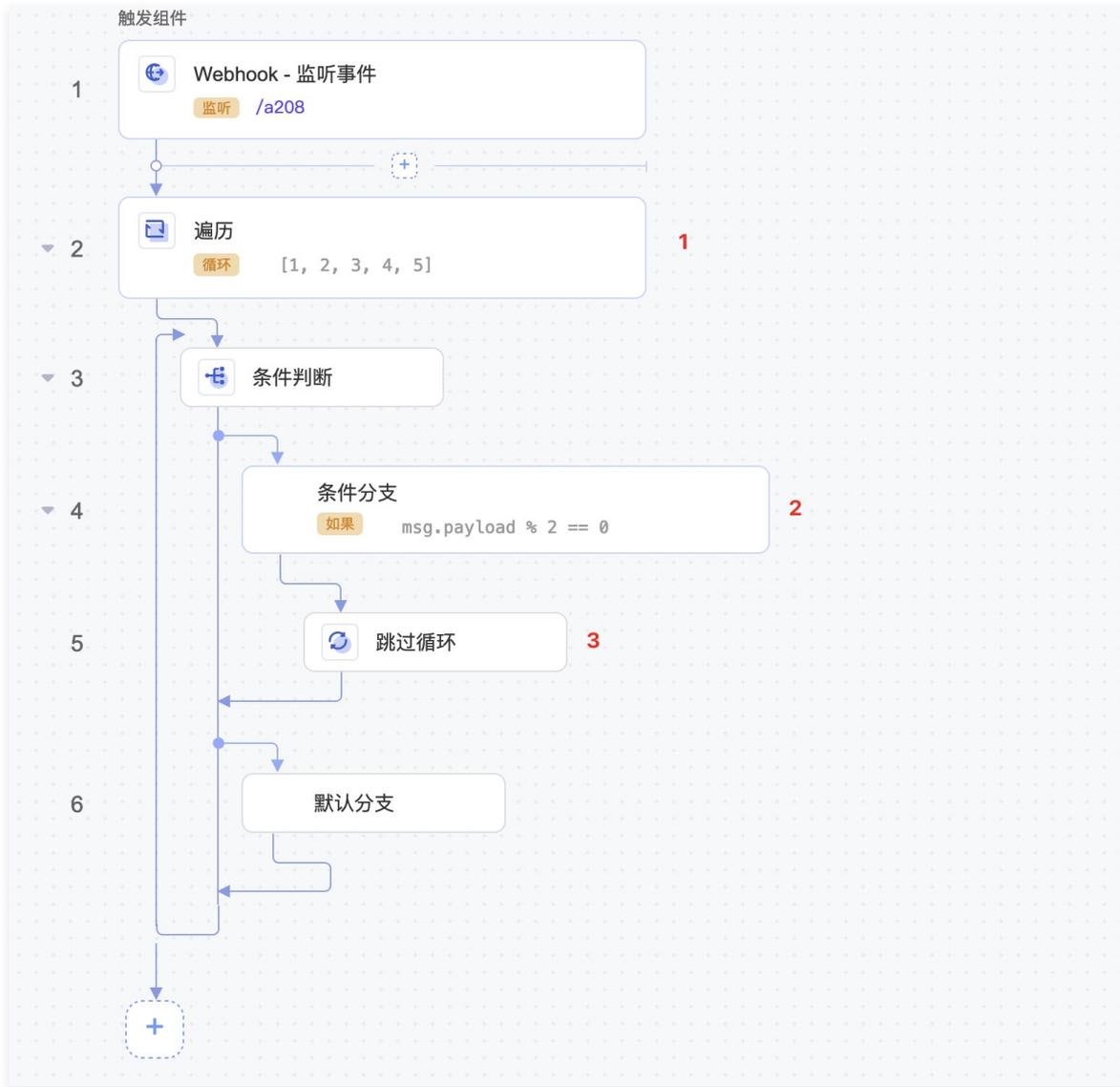
组件输出的 message 信息如下：

message 属性	值
payload	<ul style="list-style-type: none"><li>在“遍历”组件中，“跳过循环”后，payload 的值为下一次遍历的数据。</li><li>在“条件循环”组件中，“跳过循环”后，payload 继承自“跳过循环”的上一个组件输出的 payload。</li></ul>
error	无。
attribute	<ul style="list-style-type: none"><li>在“遍历”组件中使用时，“跳过循环”后，attribute 继承自“遍历”的上一个组件输出的 attribute。</li><li>在“条件循环”组件中使用时，“跳过循环”后，attribute 继承自“条件循环”的上一个组件输出的 attribute。</li></ul>
variable	<ul style="list-style-type: none"><li>在“遍历”中使用时，“跳过循环”后，variable 中的变量继承自“跳过循环”的上一个组件输出的 variable 变量。</li><li>在“条件循环”组件中使用时，variable 中的变量继承自“跳过循环”的上一个组件输出的 variable 变量。</li></ul>

## 案例

在本案例中，我们使用 For Each 对列表中的奇数求和，遇到偶数，使用 Continue 跳出当次循环。

- 添加 For Each 组件，填入要遍历的集合 [1,2,3,4,5]。
- 添加 Choice 组件，在 When 节点中对数据进行筛选。
- 在 When 节点中加入 Continue 节点。



# 遍历（For Each）

最近更新时间：2024-08-01 15:33:01

## 简介

遍历（For Each）是循环控制组件，类似于编程语言中的 for/foreach，在遍历组件中可以配置子流，对指定的数据集中每个元素执行子流处理逻辑。

## 操作说明

### 参数配置

参数	数据类型	描述	是否必填	默认值
数据集	string、list、dict、int	待遍历的数据集。 <ul style="list-style-type: none"><li>当类型为 string 时，遍历字符串的每个字符。</li><li>当类型为 list 时，遍历 list 的每个元素。</li><li>当类型为 dict 时，遍历 dict 中的 value。</li><li>当数据集类型为 int 时，例如3；实际遍历的数据集为[0,1,2]。</li></ul>	是	无
计数器	string	计数器是一个变量，该变量存储了当前的迭代次数，从0开始。这里填入变量名称，msg.vars.get('#计数器变量#')即可使用。例如：当计数器变量使用默认值 counter 时： 第1次循环，msg.vars.get('counter')值为0。 第2次循环，msg.vars.get('counter')值为1。	是	counter
根信息	string	根信息同样是一个变量，这里填入变量名称，根信息中保存了主流的消息信息。msg.vars.get('#根信息名称#').payload 即可访问主流的消息 payload 数据。当使用默认值 rootMessage 时，使用 msg.vars.get('rootMessage').payload 即可在遍历的子流中访问主流的消息 payload 数据。	是	rootMessage

#### 说明：

通常情况下，配置数据集即可。

## 配置界面

配置 >> 预览

遍历
切换组件 ▶

帮助文档

对指定数据集进行循环处理

[更多内容](#)

配置内容

基本配置

数据集 \*

```

1 def dw_process(msg):
2     return 6
                    
```

组件遍历循环的数据集对象

高级配置

### 数据预览

预览字段	数据类型	描述
payload	any	每次遍历的输入值，属于数据集的其中一个元素。
index	int	每次遍历的位置，代表当前输入值在数据集中的下标位置，从0开始计数。

配置 >> 预览

遍历
切换组件 ▶

帮助文档

对指定数据集进行循环处理

[更多内容](#)

配置并执行

点击配置并执行,将返回数据和Schema

输出预览

[数据预览](#)   Schema预览

✔ 预览成功, 内容如下
下载

```

{ 2 items
  "exts": dict { 1 item
    "index": int 0
  }
  "payload": int 0
}
                    
```

数据预览的内容仅为子流可见，子流中的组件可以直接使用遍历组件的 Payload 和 index，如图所示：



## 输入到子流中的 message

message 属性	值
payload	数据集中的元素，例如待迭代的数据集为[1,2,3]: 第一次循环，子流中 payload 的数据为1。 第二次循环为2。 当迭代的数据集为 dict 类型{"key":"key1", "value":"value1"}: 第一次循环，子流中的 payload 为 value1。 第二次为 value2。
error	空。
attribute	空。
variable	继承主流中的 variable 数据，同时新增两个变量，一个是计数器，一个是根信息，若用户使用默认值，可使用表达式 msg.vars.get('counter') 和 msg.vars.get('rootMessage') 访问。 若 For Each 中使用了 Set Variable，则在子流执行过程中，新增的变量也会添加到 variable 中。

## 输出

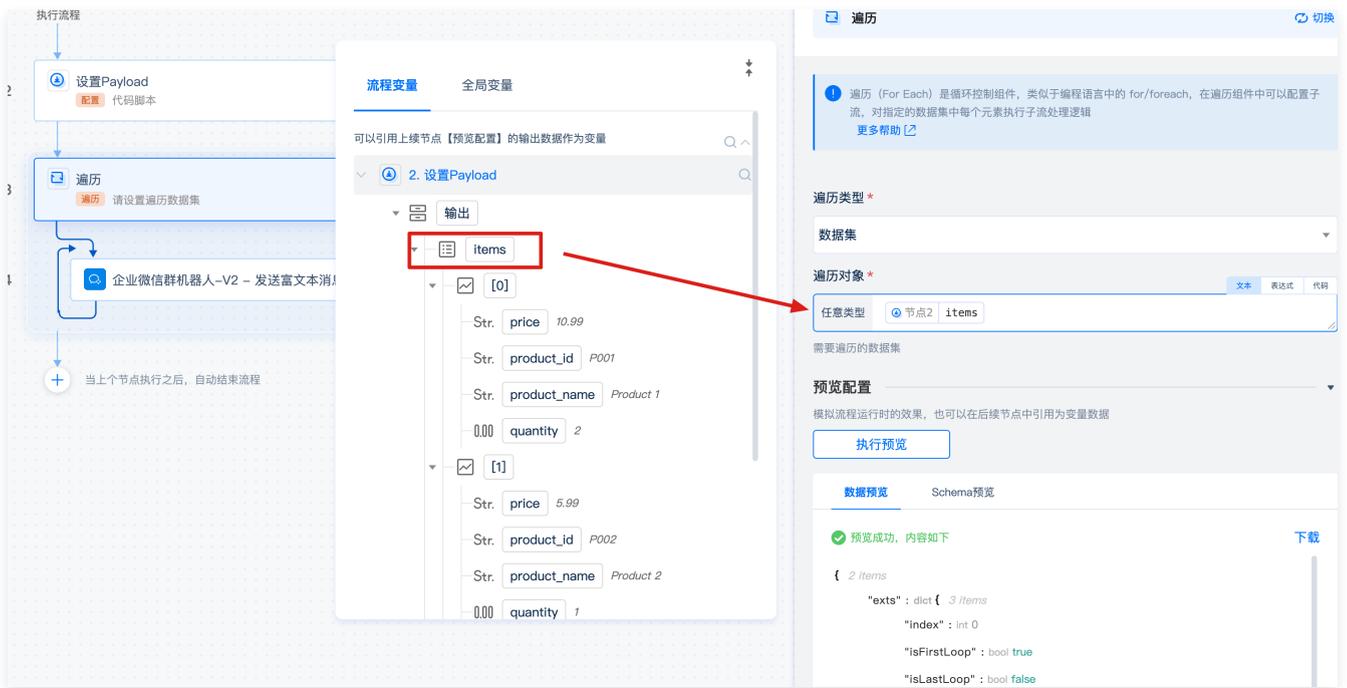
遍历组件不会改变 Message 内容，后续节点只能感知到变量的变化。

## 案例

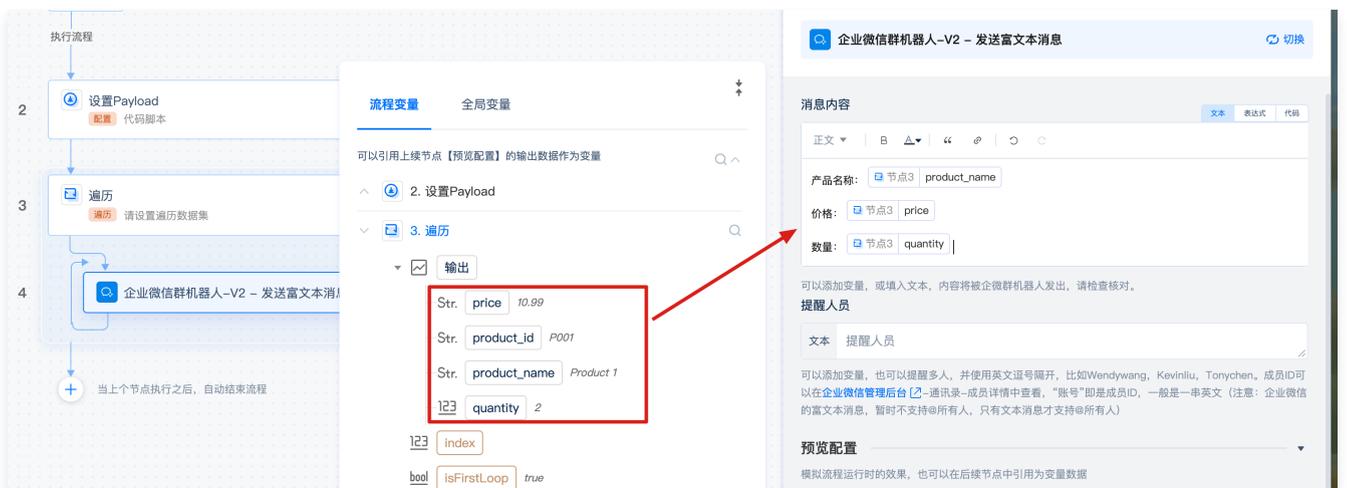
### 1. 循环一个数组。

将查询到的订单信息，逐条发送到企微中进行提醒：

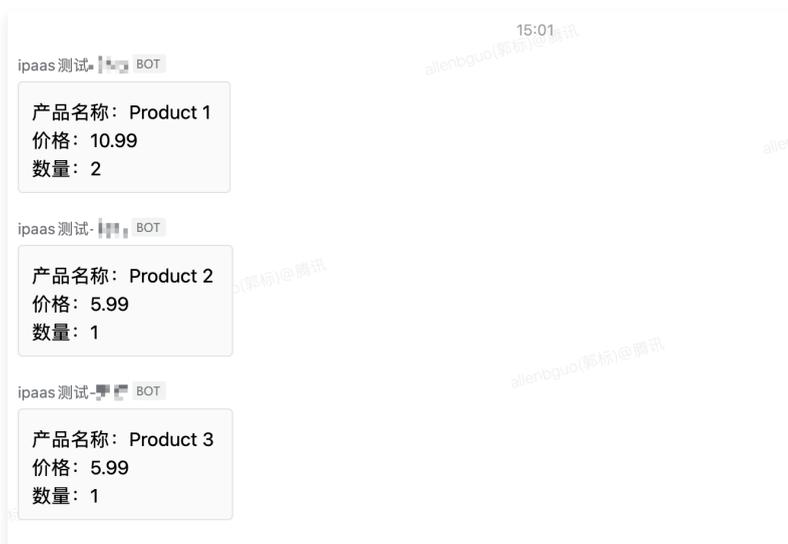
1.1 首先选择遍历类型为：数据集，遍历对象为：查询到的数组。



1.2 在遍历内插入发送消息节点，发送内容选择遍历后的数据。



1.3 得到的内容：



2. 输出等差数列。

如：1、3、5、7、9...

配置如下：

遍历类型 \*

次数

循环总次数 \*

整数 100

循环初始值 \*

整数 1

数字递增值

整数 2

# 配置 Payload ( Set Payload )

最近更新时间：2024-10-08 11:47:01

## 简介

配置 Payload ( Set Payload ) 可以设置 message 中的 payload 属性。支持文本、表达式和代码三种输入形式，如果选择文本，需要先选择数据类型，再在输入框中填入字面量。如果需要输入表达式，选择 any 类型，写入表达式即可，如果是代码模式，无需选择数据类型，直接单击输入代码即可。

## 操作配置

### 参数配置

参数	数据类型	描述	是否必填	默认值
值	any	payload 中要保存的数据	是	无

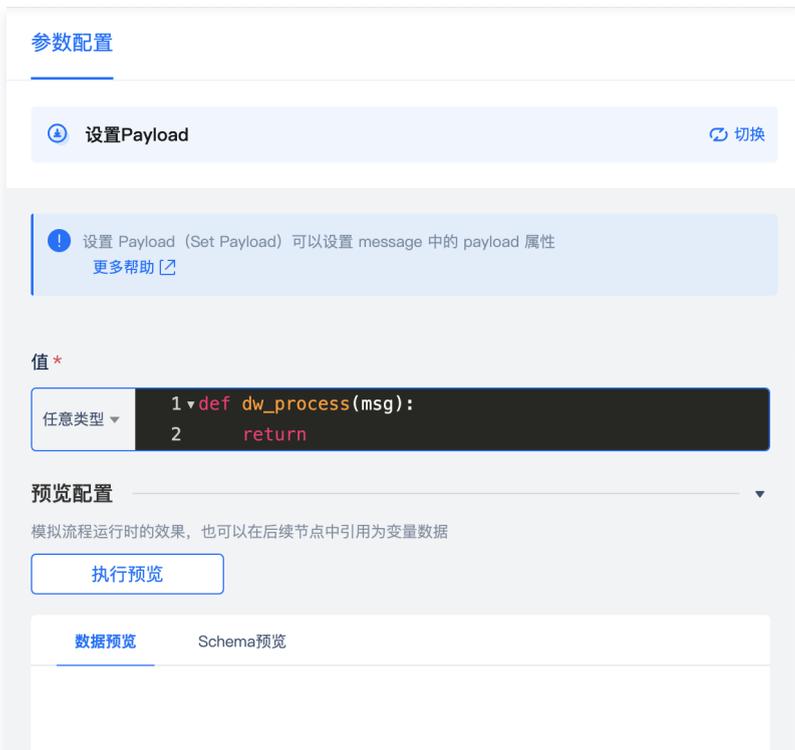
### 配置界面

- 文本模式：选择文本的数据类型，再在输入框中输入字面量即可。

- 表达式模式：直接在输入框中输入表达式即可。



- 代码模式：单击输入框进入代码模式编辑界面，输入代码即可。



## 输出

组件输出的 message 信息如下：

message 属性	值
payload	用户填入的数据
error	<ul style="list-style-type: none"> <li>执行成功后, error 为空</li> <li>执行失败后, error 为 dict 类型, 包含 “Code” 和 “Description” 字段: “Code” 字段表示错误类型, “Description” 字段表示错误具体信息</li> </ul>
attribute	继承上个组件的 attribute 信息
variable	继承上个组件的 variable 信息

## 数据预览

**参数配置**

设置Payload
切换

! 设置 Payload (Set Payload) 可以设置 message 中的 payload 属性  
[更多帮助](#)

值 \*

任意类型 ▾

## 案例

1. 添加 Set Payload 组件。

设置Payload
配置

2. 填入要设置的数据。

**参数配置**

设置Payload
切换

! 设置 Payload (Set Payload) 可以设置 message 中的 payload 属性  
[更多帮助](#)

值 \*

任意类型 ▾

3. 查看预览的数据。

数据预览    Schema预览

✔ 预览成功, 内容如下 [下载](#)

```
{ 1 item
  "输出": string "hello world"
}
```

# 配置变量（Set Variable）

最近更新时间：2025-07-03 17:43:01

## 简介

配置变量（Set Variable）的作用是声明一个变量，并保存在 message 的 variables 中，后续节点可通过 `msg.vars.get('name')` 形式引用该变量。

## 操作配置

根据您的创建应用的时间不同，配置变量（Set Variable）组件的配置也不相同，具体如下：

配置方法	适用场景
<a href="#">方法一</a>	在2022年9月1日前创建的应用
<a href="#">方法二</a>	在2022年9月2日及之后创建的应用

## 方法一（2022年9月1日前创建的应用）

### 参数配置

参数	数据类型	描述	是否必填	默认值
变量名	string	变量名称	是	无
变量值	any	变量的具体值	是	无

### 配置界面



## 输出

对 variables 变量的引用，需要使用表达式：`msg.vars.get('company')`。

组件输出的 message 信息如下：

message 属性	值
payload	继承上个组件的 payload
error	<ul style="list-style-type: none"> <li>执行成功后，error 为空</li> <li>执行失败后，error 为 dict 类型，包含“Code”和“Description”字段：“Code”字段表示错误类型，“Description”字段表示错误具体信息</li> </ul>
attribute	继承上个组件的 attribute 信息
variable	上个组件的 variable 信息加上当前组件添加的变量

## 数据预览

输出预览

数据预览

Schema预览

✔ 预览成功, 内容如下

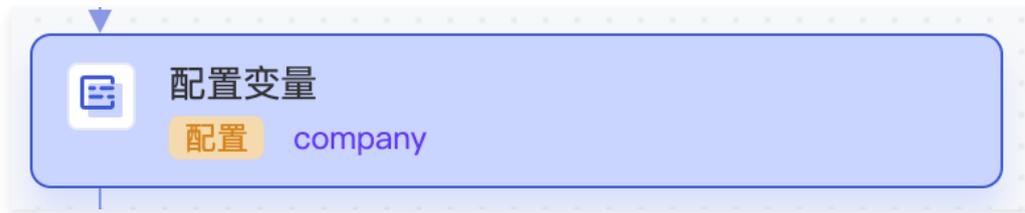
下载

```

{ 1 item
  " vars " : dict { 1 item
    " company " : string "tencent"
  }
}
    
```

案例

1. 添加 Set Variable 组件。



2. 在“变量名”处填入变量名称，“变量值”处填入要保存的值。



## 方法二（2022年9月2日及之后创建的应用）

### 参数配置

参数	数据类型	描述	是否必填	默认值	备注
变量名	string	变量名称	是	无	无
变量值	any	变量的具体值	是	无	无
变量类型	string	变量的类型	是	string	无
操作	string	变量的操作	否	无	仅当变量类型为 list 或 dict，并且已存在该变量时支持

### 配置界面

配置 >> 预览

配置变量
切换组件 ▾

帮助文档 ▾

设置流级别的组件间可共享变量

[更多内容](#)

配置内容 ▾

变量 (company) \* 🗑️

变量名 \*

Aa

变量类型

▾

变量值 \* 📄 表达式 代码

//

[+添加变量](#)

## 输出

对 variables 变量的引用，需要使用表达式：`msg.vars.get('company')`。  
 组件输出的 message 信息如下：

message 属性	值
payload	继承上个组件的 payload
error	<ul style="list-style-type: none"> <li>• 执行成功后，error 为空</li> <li>• 执行失败后，error 为 dict 类型，包含“Code”和“Description”字段：“Code”字段表示错误类型，“Description”字段表示错误具体信息</li> </ul>
attribute	继承上个组件的 attribute 信息
variable	上个组件的 variable 信息加上当前组件添加的变量

## 数据预览

## 输出预览

数据预览

Schema预览

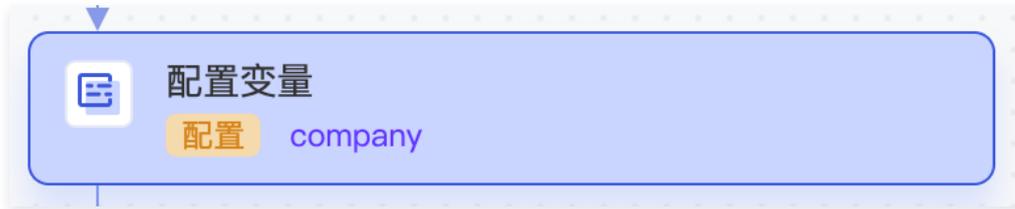
✔ 预览成功, 内容如下

下载

```
{ 1 item
  " vars " : dict { 1 item
    " company " : string "tencent"
  }
}
```

## 案例

1. 添加 Set Variable 组件。



2. 在“变量名”处填入变量名称,“变量值”处填入要保存的值。



3. 选择已存在的 dict 类型的变量 “dictVar”，选择追加操作，将键值对追加到 “dictVar” 中。

配置



预览



设置变量

切换组件 ▶

设置变量 (Set Variable) 的作用是声明一个变量，并保存在 message 的 variables 中，后续节点可通过 msg.vars.get('name') 形式引用该变量 [更多内容](#)

变量 (dictVar)\*



变量名 \*

dictVar

变量类型

dict



操作 \*

追加(若键名相同则自动重置)



追加内容

键名	键值
k	Str. v
+添加	

# 错误捕获 (Try-Catch)

最近更新时间: 2024-08-01 15:33:01

## 简介

错误捕获 (Try-Catch) 组件由一个执行子流和一个或多个捕获错误子流组成。通过配置, 可以捕获执行子流运行时抛出的错误和系统错误, 也可与抛出错误组件搭配使用, 捕获用户自定义错误。如果执行子流发生错误, 将运行第一个匹配的捕获错误子流, 如果没有匹配, 则抛出错误到外层。

## 操作指引

### 连接说明

无

### 参数配置

参数	数据类型	描述	是否必填	默认值
事务	枚举 (无; 本地事务)	和 Database 组件搭配使用, 执行失败时回滚数据库操作	是	无
错误类型	string	子流执行过程中抛出的错误类型, 通过下拉列表选择, 支持任意多选, 支持选择全部错误或者某一类错误	是	无
重试次数	int	发生此类错误时重试执行子流的次数 (0-5次), 每个捕获错误单独统计	是	不重试
重试间隔	int	重试时两次执行的间隔 (1-300秒)	否	60

事务在开启错误捕获节点配置。

### 配置

开启错误捕获 切换组件 ▶

#### 帮助文档

捕获并处理异常  
[更多内容](#)

#### 配置内容

事务

本地事务

错误类型和重试在捕获错误节点配置。

配置



捕获错误

帮助文档

捕获和处理异常的分支

[更多内容](#)

配置内容

错误类型 \*

全部错误

点击 [添加](#) 可创建自定义错误类型, 或 [点击此处](#) 管理自定义错误类型

重试次数

1

配置当捕获指定错误时的自动重试次数

重试间隔 (秒)

123

60



两次重试的最小时间间隔,取值范围[1,300],单位秒

数据预览

无

输入到子流中的 message

message 属性	值
payload	继承错误捕获上一个组件的 payload
error	继承错误捕获上一个组件的 error
attribute	继承错误捕获上一个组件的 attribute
variable	继承错误捕获上一个组件的 variable

输出

组件输出的 message 信息如下:

message 属性	值
------------	---

payload	<ul style="list-style-type: none"><li>若<b>执行子流</b>正常运行，payload 的结果为<b>执行子流</b>输出的 payload。</li><li>若抛出错误，且错误被捕获，则 payload 的结果为<b>捕获错误子流</b>输出的 payload。</li><li>若抛出的错误未被捕获，则子流终止运行。</li></ul>
error	<ul style="list-style-type: none"><li>若<b>执行子流</b>抛出错误，且未被捕获，或者<b>捕获错误子流</b>中抛出错误，则 error 保存了错误信息，为 dict 类型，包含“Code”和“Description”字段：“Code”字段表示错误类型，“Description”字段表示错误描述。</li><li>否则 error 为空。</li></ul>
attribute	<ul style="list-style-type: none"><li>若<b>执行子流</b>正常运行，attribute 的结果为<b>执行子流</b>输出的 attribute。</li><li>若抛出错误，且错误被捕获，则 attribute 的结果为<b>捕获错误子流</b>输出的 attribute。</li><li>若抛出的错误未被捕获，则子流终止运行。</li></ul>
variable	<ul style="list-style-type: none"><li>若<b>执行子流</b>正常运行，variable 的结果为子流输出的 variable。</li><li>若抛出错误，且错误被捕获，则 variable 的结果为<b>捕获错误子流</b>输出的 variable。</li><li>若抛出的错误未被捕获，则子流终止运行。</li></ul>

## 案例

1. 新增自定义错误类型请求失败。
2. 当请求失败时，抛出该错误。
3. 捕获该错误，执行重试策略，会重新执行**执行子流**中的 HTTP 请求，条件判断。重试后仍失败，则记录日志。



# 条件判断 (Choice)

最近更新时间: 2024-08-01 15:33:01

## 简介

条件判断 (Choice) 是分支选择语句，基于不同的条件执行不同的动作，类似于 if-else。条件判断包含两类子节点，“条件分支”和“默认分支”，在条件判断中可以添加多个条件分支节点，每个条件分支节点中都包含一个真值表达式，条件判断组件会对条件分支节点依次逐个判断真值，直到第一个真值表达式满足条件，则执行该条件分支节点中配置的子流。当所有的 When 条件都无法匹配时，会执行默认分支的动作。

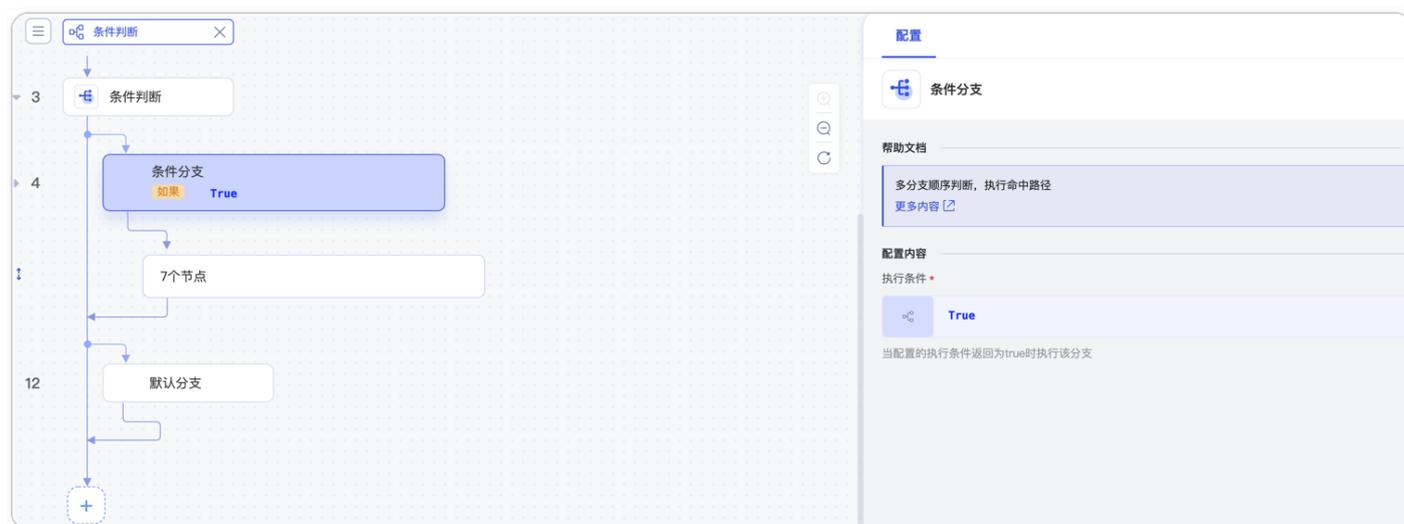
## 操作说明

### 表达式模式参数配置

在 When 节点中，可以配置条件语句，用来控制分支选择。

参数	数据类型	描述	是否必填	默认值
执行条件	bool	条件判断，当条件满足时，执行对应的子流	是	无

### 表达式模式配置界面



### 列表模式参数配置

图形化地配置多组比较条件，多组条件之间可以通过逻辑运算符“OR”或者“AND”连接。

参数	数据类型	描述	是否必填	默认值
值	any	值	是	无
条件	枚举	条件，即比较运算符	是	无

### 列表模式配置界面

### 配置

条件分支

多分支顺序判断，执行命中路径 [更多内容](#)

执行条件\* 列表配置 表达式

值\*

条件\*

值\*

+ AND

OR

+ or

## 数据预览

无

## 输入到子流中的 Message

完全继承于主流当前的 Message。

## 输出

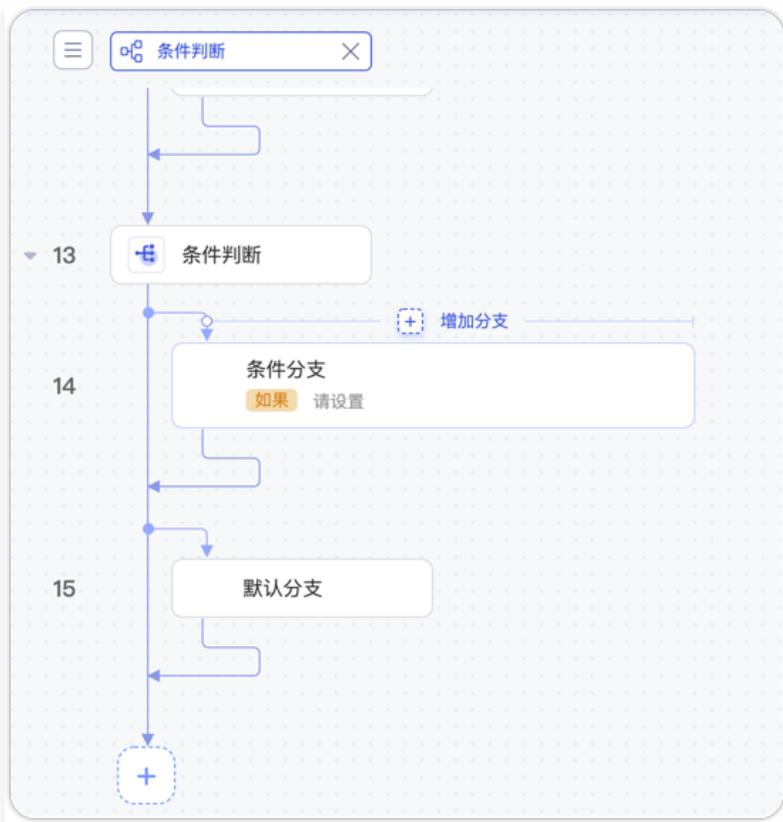
完全输出子流最终输出的 Message，包括错误。

## 案例

在该案例中，我们将 score 映射为不同的 level：

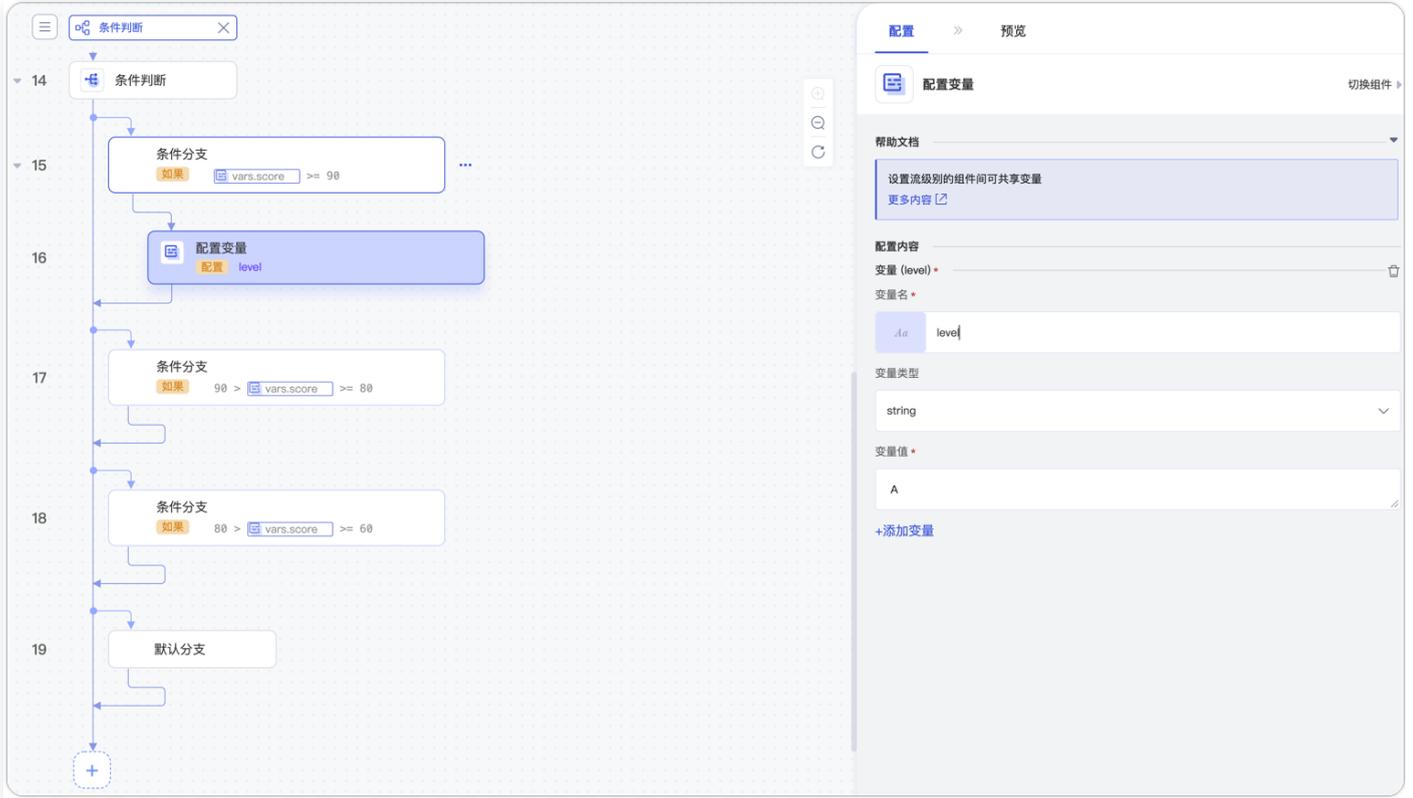
- 当 score 大于等于90时，level 为“A”。
- 当 score 大于等于80、小于90时，level 为“B”。
- 当 score 大于等于60、小于80时，level 为“C”。
- 当 score 小于60时，level 为“D”。

1. 添加条件判断组件。

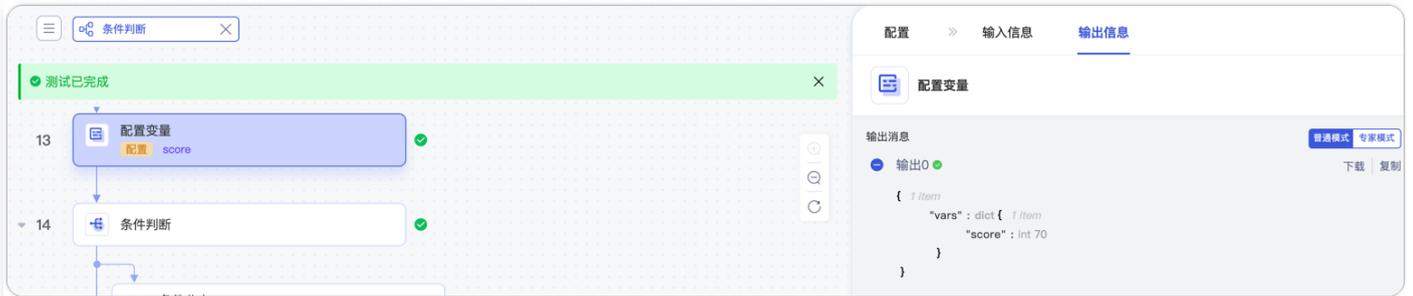


2. 在现有条件分支的上面，可以添加新的条件分支节点，每个条件分支对应一个 level。

3. 在对应的条件分支中设置 level。例如：score >= 90的分支，设置 level 为“A”。



4. 配置完成执行单元测试。当 score 值为70时，条件判断执行第三个分支，level 设置为“C”。



显示如下：

The screenshot displays a workflow editor interface. On the left, a flowchart is shown with steps numbered 14 to 21. Step 14 is a '条件判断' (Condition Judgment) node. Step 15 is a '条件分支' (Condition Branch) node with the condition 'vars.score >= 90'. Step 16 is a '配置变量' (Configure Variable) node for 'level'. Step 17 is another '条件分支' node with the condition '90 > vars.score >= 80'. Step 18 is a '配置变量' node for 'level'. Step 19 is a third '条件分支' node with the condition '80 > vars.score >= 60'. Step 20 is a '配置变量' node for 'level'. Step 21 is a '默认分支' (Default Branch) node. A green bar at the top indicates '测试已完成' (Test Completed). On the right, the '输出信息' (Output Information) panel shows the output of the workflow, which is a JSON object: 

```
{ 1 item\n  "vars": dict { 1 item\n    "level": string "C"\n  }\n}
```

# API 管理

最近更新时间：2024-04-25 09:58:11

## 注意：

- 试用公有云版本 API 管理需扫码申请（登录 [腾讯轻联控制台](#)，单击 **API 管理** 菜单即可申请）。
- API 管理的独立环境与应用集成环境不互通。

## 操作场景

各大企业每天都有大量的 API 增长，同时越来越多公司开始公开 Web API，API 的使用场景正在累积。现在，每日 API 调用量在不断飙升，如何能够安全有效将这些 API 管理起来对于企业而言并不容易。

腾讯轻联提供 API 发布功能，可以一键将已发布的应用打包生成 API，方便用户进行管理和调用；同时提供了 API 管理能力，可以针对 API 进行访问权限管控和流量调度。

## 操作步骤

### API 管理主页

登录 [腾讯轻联控制台](#)，在左侧导航栏单击 **API > API 管理**，即可进入 API 管理的主页。

在 API 管理主页，您可以创建或查看 API 服务、可以查看 API 目录、管理 API 订阅凭证，同时可对 API 服务进行审核管理。

### API 服务列表

此页面可创建或查看 API 服务。创建完成的 API 服务展示在此列表，项目内所有成员都可查看。

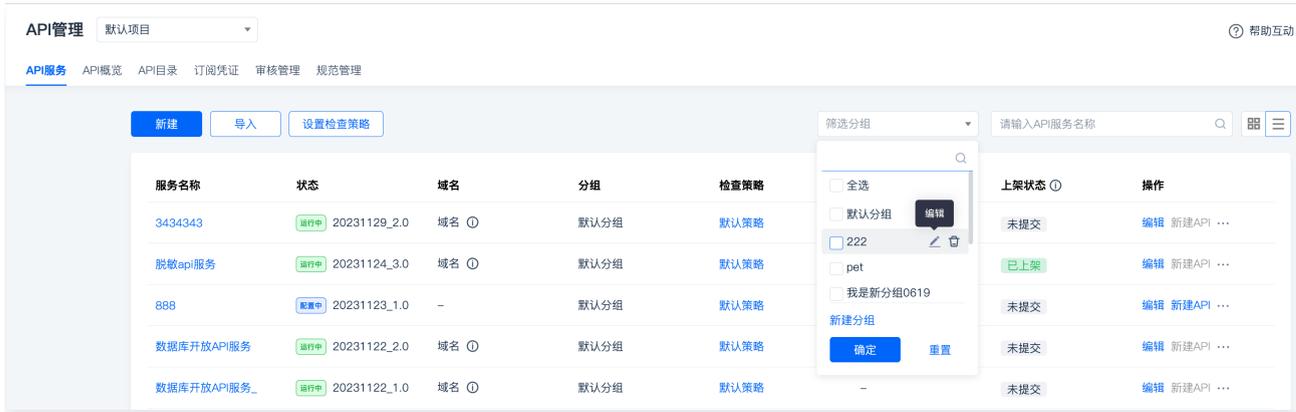
服务名称	状态	域名	分组	检查策略	描述	上架状态	操作
3434343	运行中	20231129_2.0 域名	默认分组	默认策略	-	未提交	编辑 新建API ...
脱敏api服务	运行中	20231124_3.0 域名	默认分组	默认策略	-	已上架	编辑 新建API ...
888	配置中	20231123_1.0 -	默认分组	默认策略	-	未提交	编辑 新建API ...
数据库开放API服务	运行中	20231122_2.0 域名	默认分组	默认策略	-	未提交	编辑 新建API ...
数据库开放API服务_	运行中	20231122_1.0 域名	默认分组	默认策略	-	未提交	编辑 新建API ...
sap_rfc_服务_时间戳	运行中	20231109_1.0 域名	默认分组	默认策略	-	未提交	编辑 新建API ...
数据库开放API服务2	配置中	20231122_1.0 -	默认分组	默认策略	-	未提交	编辑 新建API ...
3434343-2	运行中	20231120_1.0 域名	默认分组	默认策略	-	未提交	编辑 新建API ...
66	运行中	20231114_1.0 域名	默认分组	默认策略	-	未提交	编辑 新建API ...
AI家-测试环境2222	配置中	20231114_1.0 -	默认分组	默认策略	AI家-测试环境	未提交	编辑 新建API ...

- API 服务状态有三种：配置中、运行中、已停止。鼠标可 hover 到**服务域名**处查看该 API 服务的发布环境和域名。
- API 服务支持的操作有：查看、新建 API、上架、下架、删除、查看描述文件、查看发布历史。

## 注意：

- 只有服务状态为**配置中**的 API 服务才有新建 API 和删除的操作。
- 只有服务状态为**运行中**的 API 服务才可进行上架和下架的操作。
- 单击 API 服务的名称可跳转到 API 列表页。

支持管理服务分组，包括新建、删除、编辑服务分组。每个项目下的服务分组相互独立，互不影响。

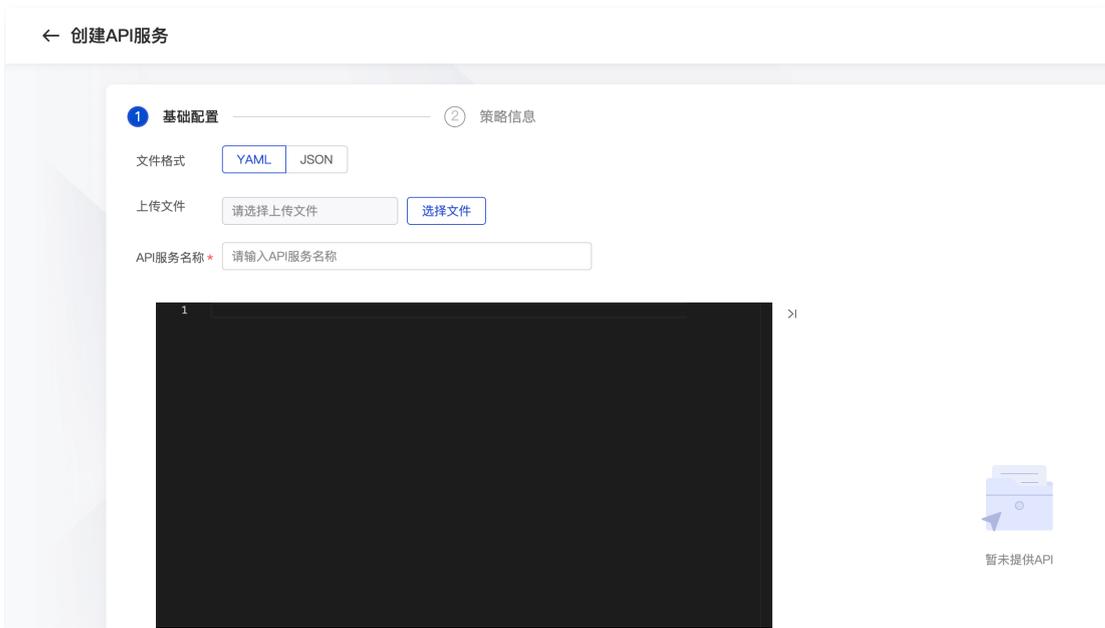


## 新建 API 服务

API 管理功能支持 3.0.0 版本的 OpenAPI 规范。OpenAPI 3.0.0 规范的对象定义请参见 [OpenAPI Specification](#)。用户可以通过单击 **新建** 进入 API 服务创建界面。创建 API 服务一共有2个步骤。

### 方式一：导入服务

单击**导入**，即可支持通过上传 YAML 或 JSON 格式的文件，来导入服务。



### 方式二：新建服务

#### 1. 配置服务名称，协议与描述。

服务名称和描述自定义即可，协议支持3种：HTTP、HTTPS、HTTP+HTTPS。分组可选择默认分组、新建分组。如下图所示。

← 创建API服务

1 基础配置 2 策略信息

服务名称\*

协议  HTTP&HTTPS  HTTP  HTTPS

分组

描述  0 / 150

## 2. 配置策略信息。

- 配置黑白名单：您可以按需开启，开启后可以输入多条 IP 进行黑白名单的访问限制。
- 基础流量控制：您可以按需开启，开启后可配置服务请求频率，每单位时间内允许的最大请求次数，填写范围为1 - 1000。

当上述配置全部完成后，单击**完成**，则会返回 API 服务列表，同时已经创建好的 API 服务信息将会展示在此处。

← 创建API服务

1 基础配置 2 策略信息

IP访问限制

类型  黑名单  白名单

请求IP黑名单

基础流量控制

服务请求频率  次/秒

## 新建 API

当我们创建好一个 API 服务之后，可以开始编辑其具体 API 行为，包括 API 的请求路径（API Path）、请求方法、分组、鉴权策略、请求参数、策略配置、API 绑定的后端服务类型等操作。

新建 API 一共有3个步骤。

### 步骤一：基础配置

← 创建API
?
👤

① 基础配置      ② 后端配置      ③ 响应示例

**基础配置**

API名称 \*

分组

描述

请求路径 \*

所有Path位置参数的入参必须在-之后，放置在请求路径中，如:/param/

请求方法 GET POST PATCH PUT DELETE HEAD

鉴权策略 NoAuth BasicAuth OAuth2.0 HMAC-1 HMAC-256

后端服务类型 集成流  
通过API调用集成流 第三方服务  
将第三方服务开放 数据库  
基于表单或SQL创建 Mock  
模拟响应测试

参数名	位置	参数类型	默认值	设为必填	描述	操作
+ 添加一条(0/30)						

策略配置

IP访问限制

API 名称、描述支持自定义。分组可选择默认分组、新建分组。

- 请求方法支持：GET、POST、PATCH、PUT、DELETE、HEAD。
- 鉴权方式支持：NoAuth、BasicAuth、OAuth2.0、HMAC。
- 后端服务类型支持：集成流、第三方服务、数据库、Mock 类型。
- 请求参数支持：可自行添加请求参数，参数类型包括 integer、string、number、boolean、array、object，最多添加30条。

请求参数	参数名	位置	参数类型	默认值	设为必填	描述	操作
	<input type="text"/>	Header	string	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="text"/>	🗑
	<input type="text"/>	Header	string	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="text"/>	🗑
+ 添加一条(2/30)							

- 数据库类型支持新建或选择 MySQL、Oracle、Postgre SQL、Sql Server 类型的数据源。

后端服务类型 流程  
通过API调用流程 第三方服务  
将第三方服务开放 数据库  
基于表单或SQL创建 Mock  
模拟响应测试

数据源类型 DataBase MongoDB

数据源配置 \*  + 新建数据源

**策略配置：**

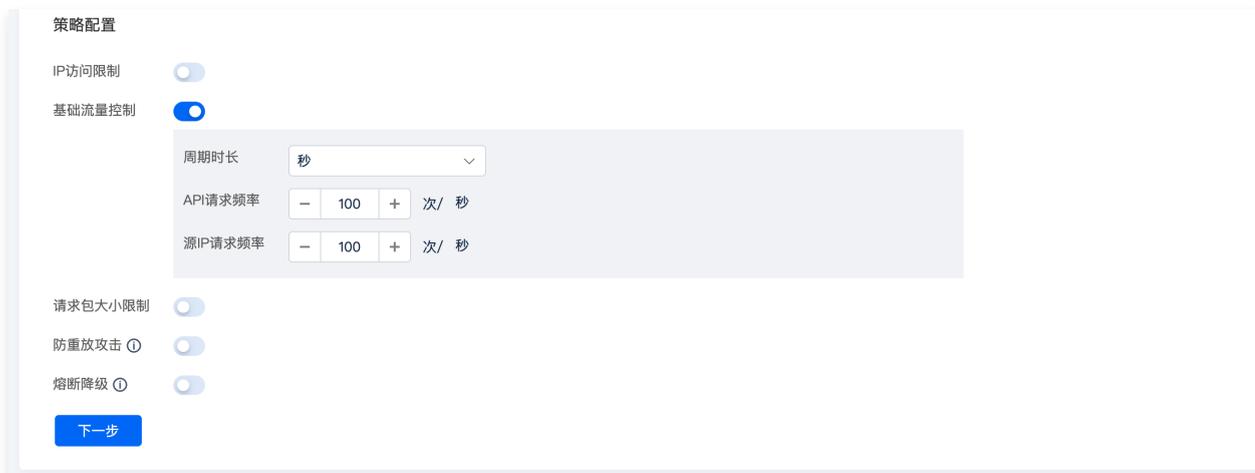
轻联 API 管理支持多样的策略配置，如下：

- **IP 访问限制：**  
支持针对 API 接口粒度设置 IP 访问黑名单/白名单。



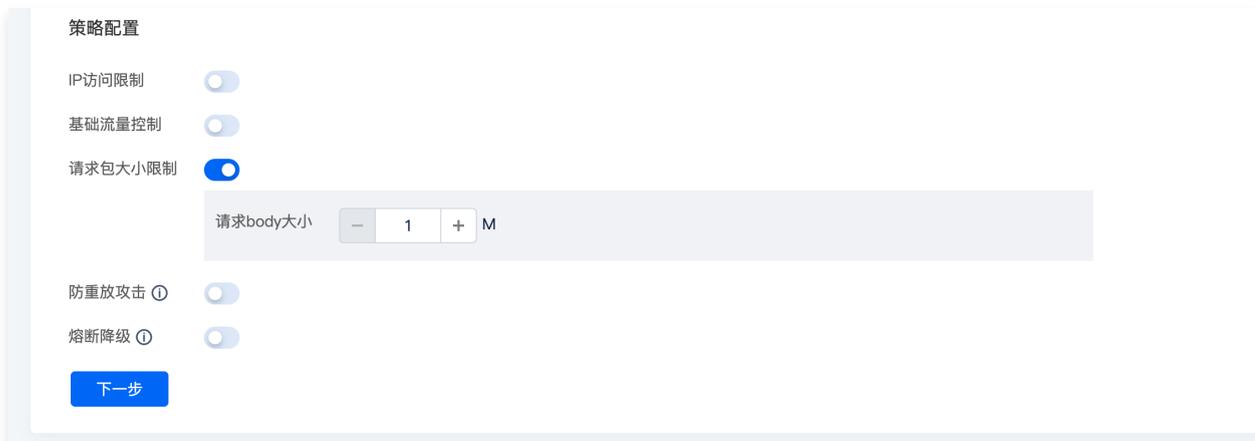
● **基础流量控制：**

支持针对 API 接口粒度设置基础流量控制，包括 API 请求频率（该 API 被请求的最大频率）和源 IP 请求频率（该 API 被同一源 IP 请求的最大频率），周期时长包括：秒、时、日、7日、30日、365日。



● **请求包大小限制：**

支持针对 API 接口粒度设置请求包大小限制，指定请求 body 大小。



● **防重放攻击：**

支持针对 API 接口粒度设置防重放攻击，开启后需配置 header 请求参数 X-Ca-Nonce 和 X-Ca-Timestamp，并在每次请求时传入这两个参数：X-Ca-Timestamp 代表请求时刻时间戳（秒级），X-Ca-Nonce 代表请求的唯一标识（每个有效时长内不允许重复）。基于这两个参数来保障有效时长内请求的唯一性，保障 API 安全。

nonce 有效时长代表底层存储每个 API 请求唯一标识的时长，如果太大可能影响校验性能。并且会将 nonce 有效时长视作正常 API 请求从发出到到达服务端的最大时间差，如果太小可能误判正常请求。

**策略配置**

IP访问限制

基础流量控制

请求包大小限制

防重放攻击

nonce有效时长  秒  
需配置和传入header请求参数X-Ca-Nonce和X-Ca-Timestamp

熔断降级

**下一步**

**熔断降级：**

支持针对 API 接口粒度设置熔断降级策略，当每个统计周期内该 API 发生的错误（按错误定义）达到错误次数后，即会触发熔断并维持断路持续时长，此期间所有请求均转发到降级后端配置，从而适应 API 请求高并发场景。

**熔断降级**

错误定义 当以下  条件满足时，被视作一次错误

包含  其中至少一个

+ 新增条件 (2/2)

统计周期  秒

错误次数  次

断路持续时长  秒

降级后端类型  无  流程  第三方服务  自定义

返回错误码

**步骤二：后端配置**

← 编辑API

① 基础配置 | ② 后端配置 | ③ 响应配置

集成流 \*

仅支持选择运行中的且由Webhook触发的集成流

后端超时时间  秒

请求方法 \*  GET  POST  PATCH  PUT  DELETE  HEAD

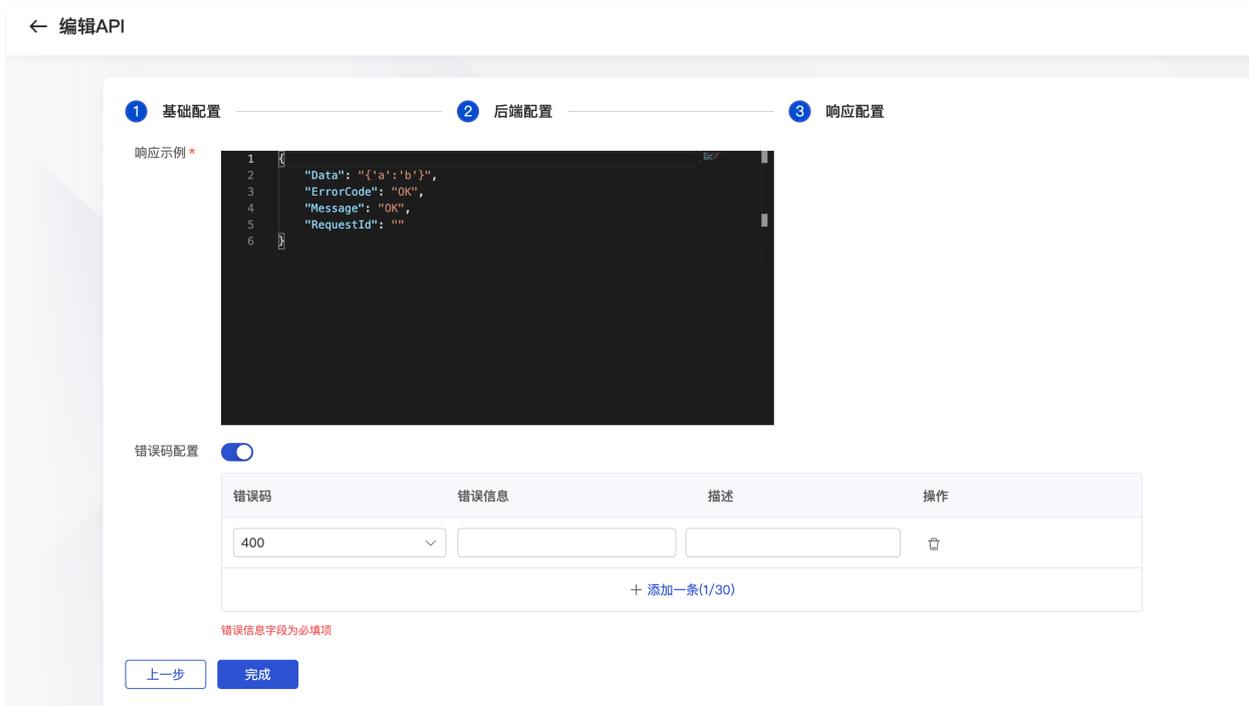
后端参数名	后端参数位置	前端参数名	前端参数位置	前端参数类型
<input type="text" value="name"/>	<input type="text" value="Header"/>	name	header	string

**上一步** **下一步** **完成**

- 选择集成流时：需选择 webhook 触发的且在运行中的集成流。
- 后端超时时间：可默认系统预设的，也可自定义。
- 请求方法：根据用户需求选择即可。支持：GET、POST、PATCH、PUT、DELETE、HEAD。
- 参数定义：支持配置前后端参数映射。

### 步骤三：响应配置

支持配置响应示例，以及错误码配置。



当上述配置全部完成后，单击**完成**，则会返回 API 列表，同时已经创建好的 API 信息将会展示在此处。

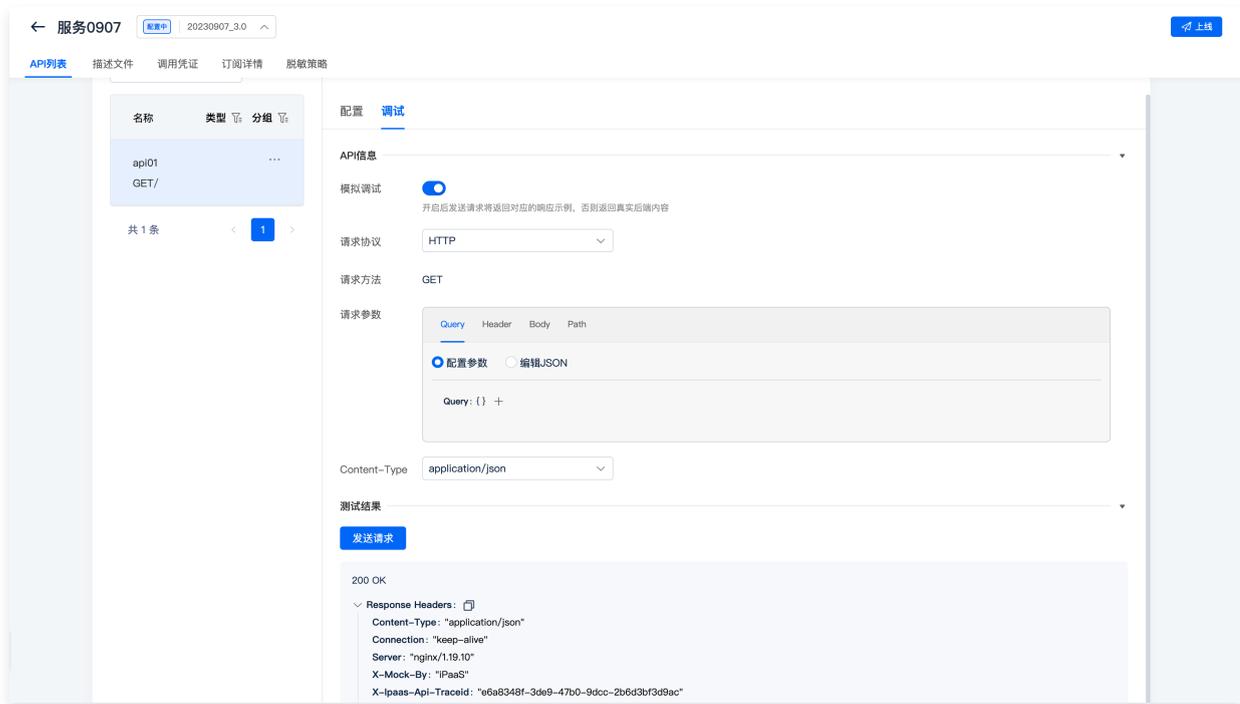
### API 列表页

创建好的 API 会展示在 API 列表中，此页面可新建、查看和编辑 API。可发布 API、可设置、查看 API 描述文件和调用凭证，可查看 API 的订阅详情。



左侧是 API 列表，右侧默认 tab 页是 API 的详细配置信息：API 的访问路径、请求方法、参数、后端服务类型等都可在此处查看。

1. 单击**调试** tab 页，进入 API 调试页面。在 API 调试页面，您可以配置此 API Endpoint 的请求 Header 和 Body 内容，并单击**发送请求**。



2. 随后即可获取到测试的结果。我们会将后端服务返回的 Response 状态码和结果返回给用户，方便进行进一步的调试工作。

3. 右上角单击**发布**，可发布此 API 到多个环境。

4. 发布后，发布后的 API 状态为**运行中**。

5. 右上角的**复制**，可将当前版本覆盖到配置中的版本。复制后，API 服务可以再次发布。一个 API 服务可以发布到多个环境。

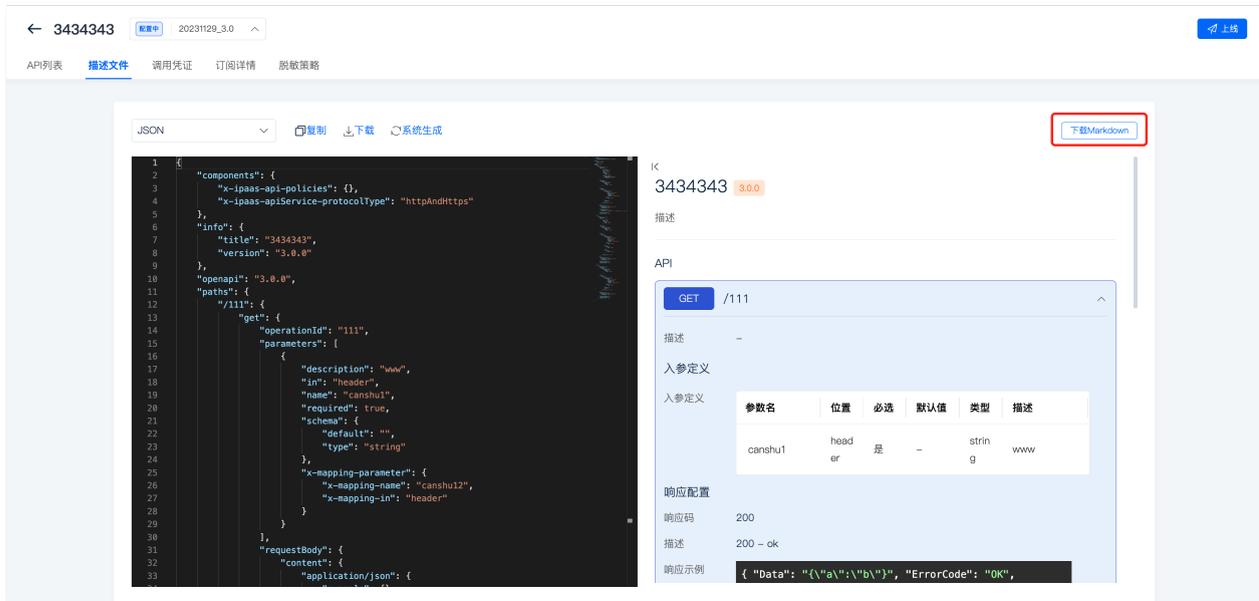
6. 可在 API 详情页可查看其日志和监控。日志与监控详细信息参见 [运维中心-监控管理](#) 和 [运维中心-运行日志](#)。

7. 发布后的 API 服务可停止服务。

## 描述文件

描述文件是针对当前 API 服务的说明，左侧展示 YAML/JSON 格式文件，右侧对应展示 Swagger 可视化内容。

支持将 API 文档导出为 Markdown 格式文件，方便调用方查看。

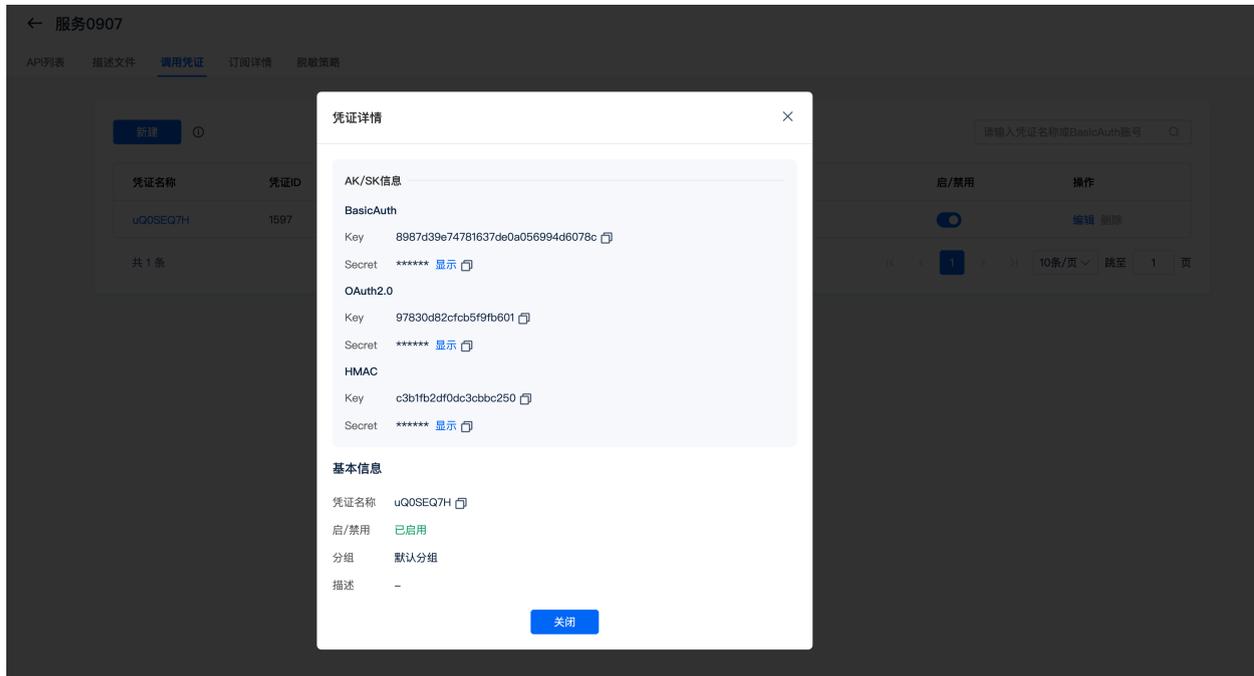


## 调用凭证

创建 API 服务时，若选择的鉴权策略为 NoAuth，则可忽略此选项，反之，若 API 服务需要鉴权，则需在此页面配置调用凭证。通过当前服务下的任意调用凭证，即可调用服务下任意 API。



上图为凭证列表页，创建好的凭证会展示在此处，选择新建凭证，即可创建新的凭证。自定义凭证的信息，保存即可。



## 订阅详情

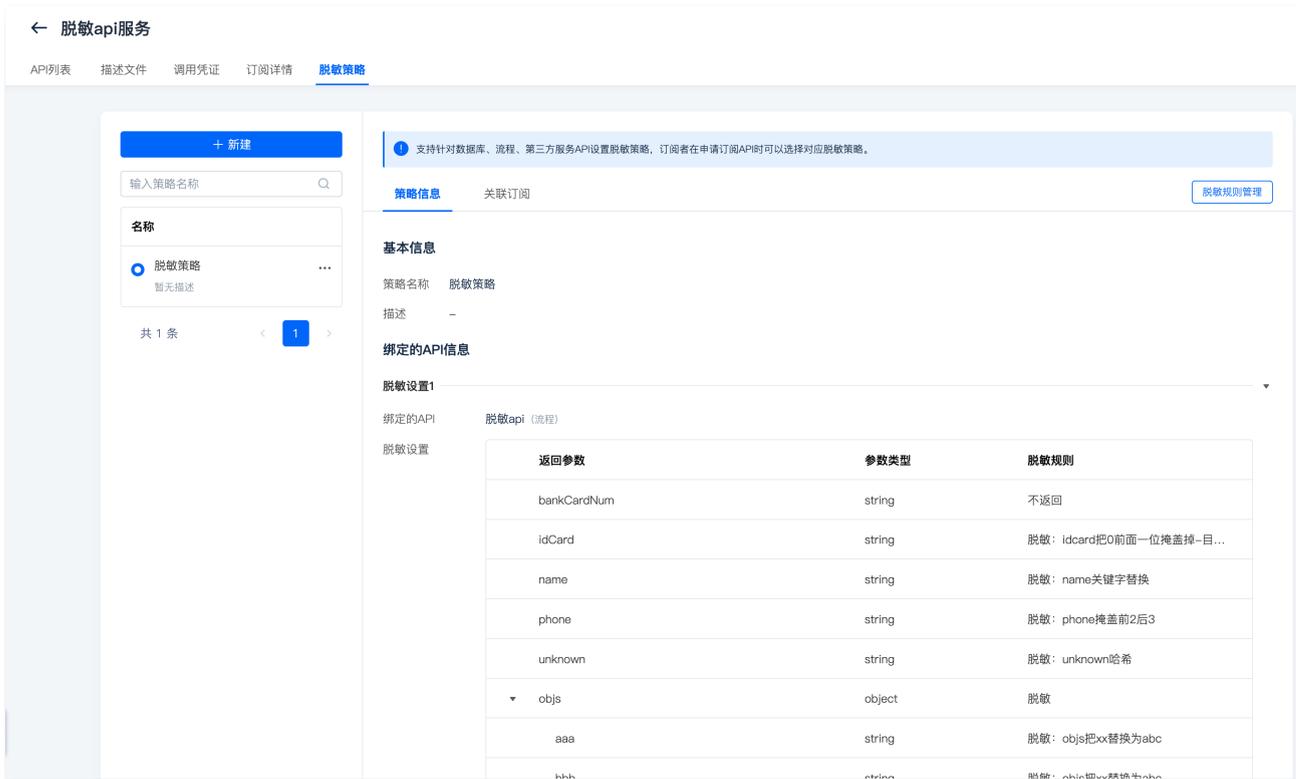
上架后的 API 服务，可以被该企业主 UIN 下的所有子 UIN 订阅并调用。该菜单可以查看当前 API 服务被订阅的情况。



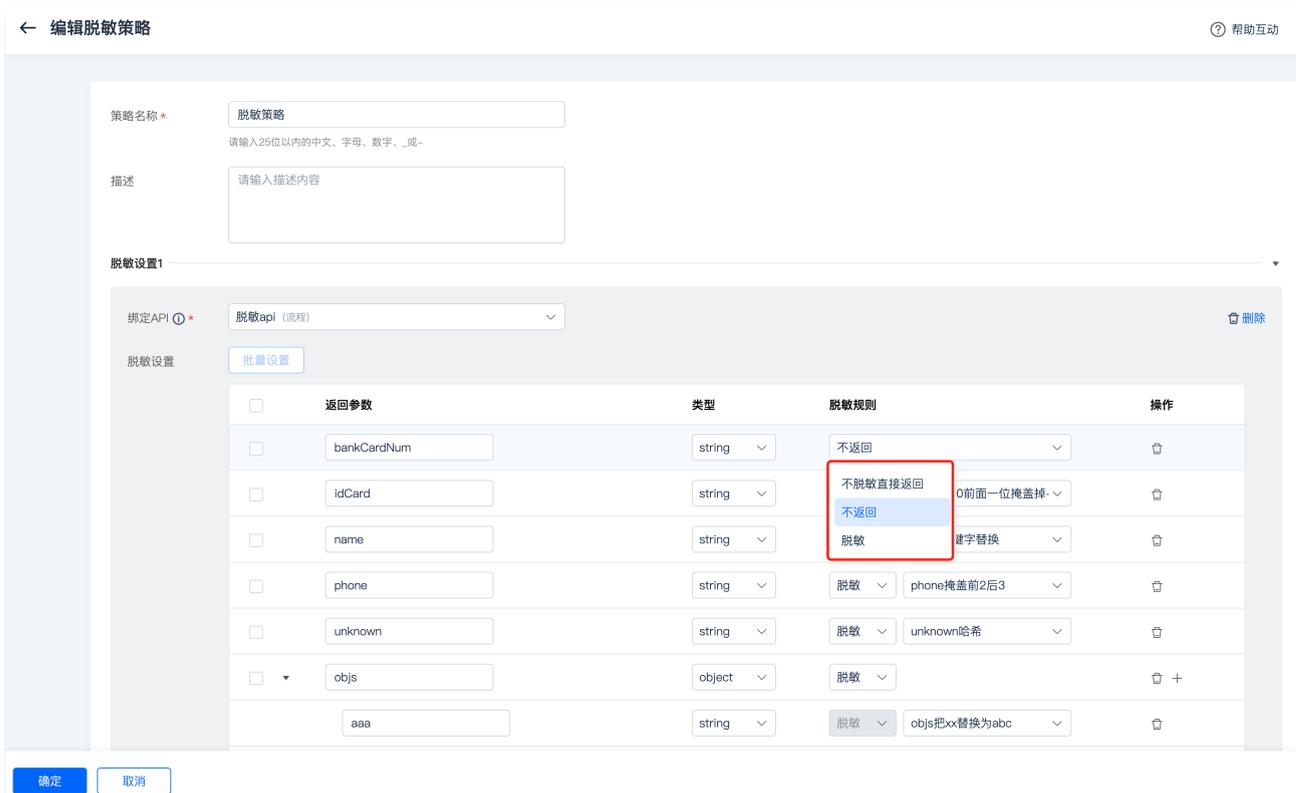
此处可以看到订阅该 API 的所有用户名单，同时，可以移除某用户的订阅。

## 脱敏策略

支持针对 API 服务设置脱敏策略，上架后被订阅时可以选择绑定脱敏策略，从而实现订阅者返回按脱敏策略处理过的响应报文。



新建脱敏策略时，支持选择运行中的 API 服务，目前支持的脱敏规则如下，支持在一个脱敏策略中新增多个脱敏设置（每个脱敏设置对应一个 API 接口）。可以设置脱敏规则为：不脱敏直接返回、不返回、脱敏。



当选择脱敏时，支持选择预置脱敏规则和自定义脱敏规则。

← 编辑脱敏策略 帮助互动

策略名称 \* 脱敏策略  
请输入25位以内的中文、字母、数字、\_或-

描述 请输入描述内容

脱敏设置1

绑定API \* 脱敏api (流程) 删除

脱敏设置 [批量设置](#)

返回参数	类型	脱敏规则	操作
<input type="checkbox"/> bankCardNum	string	不返回	删除
<input type="checkbox"/> idCard	string	脱敏	删除
<input type="checkbox"/> name	string	脱敏	删除
<input type="checkbox"/> phone	string	脱敏	删除
<input type="checkbox"/> unknown	string	脱敏	删除
<input type="checkbox"/> objjs	object	脱敏	删除
<input type="checkbox"/> aaa	string	脱敏	删除

自定义脱敏规则如下，支持设置掩码、哈希、关键字替换的脱敏算法。

脱敏规则管理 ×

[新建](#)

规则名称	脱敏算法	算法实现	描述	操作
中文姓名脱敏 (系统预设)	-	中文名称: 展示最后一字	-	效果测试
身份证号脱敏 (系统预设)	-	18位身份证: 展示首三末二与生日...	-	效果测试
手机号脱敏 (系统预设)	-	11位手机号: 展示首三末四	-	效果测试
银行卡号脱敏 (系统预设)	-	银行卡号: 展示首六末四	-	效果测试
idcard把0前面一位掩盖	掩码	字符0前遮盖1位, 掩码字符*	-	效果测试 删除
phone掩盖前2后3	掩码	遮盖首2后3, 掩码字符*	-	效果测试 删除
把2换成999	关键字替换	关键字2替换成999	-	效果测试 删除
objjs把xx替换为abc	关键字替换	关键字xx替换成abc	-	效果测试 删除
unknown哈希	哈希	SHA1	-	效果测试 删除
name关键字替换	关键字替换	关键字明波替换成abc	-	效果测试 删除

共 10 条 < > 1 > > 10条/页 跳至 1 页

## 服务操作

### 删除 API 服务

单击删除，可删除当前 API 服务。删除后，该 API 服务下的所有配置将被清空，且无法恢复。运行中的服务不可直接删除，需先停止再删除。

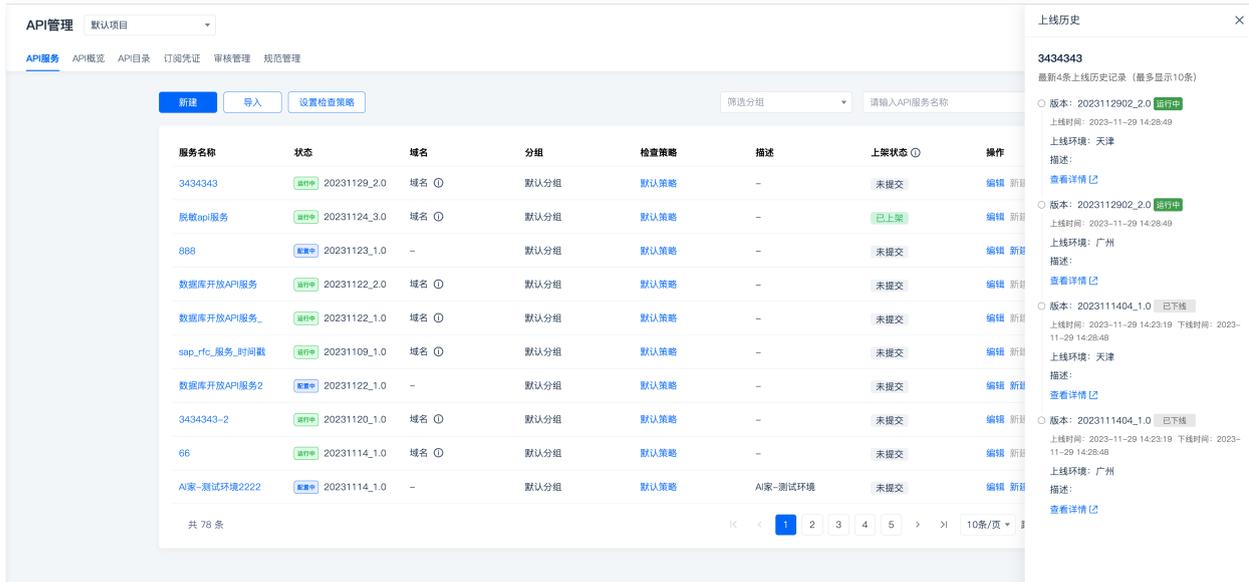
## API 上架与下架

- **上架**：运行中的 API 服务，可通过**上架**功能共享给企业的其他员工使用。在 **API 服务 > 操作 > 更多 > 上架**路径提交上架申请，提交后的 API 服务会被企业管理员审核，审核通过后可展示在 API 目录中，支持被当前主账号下的所有子账号订阅。
- **下架**：上架后的 API 服务，若不想继续被其他员工订阅，则可通过**下架**来完成。在 **API 服务 > 操作 > 更多 > 下架**路径提交下架申请，提交申请后需联系系统管理员或该项目的项目管理员进行审核，审核通过后即可下架，API 下架后不能被订阅。下架后的 API 服务会移出 API 目录。



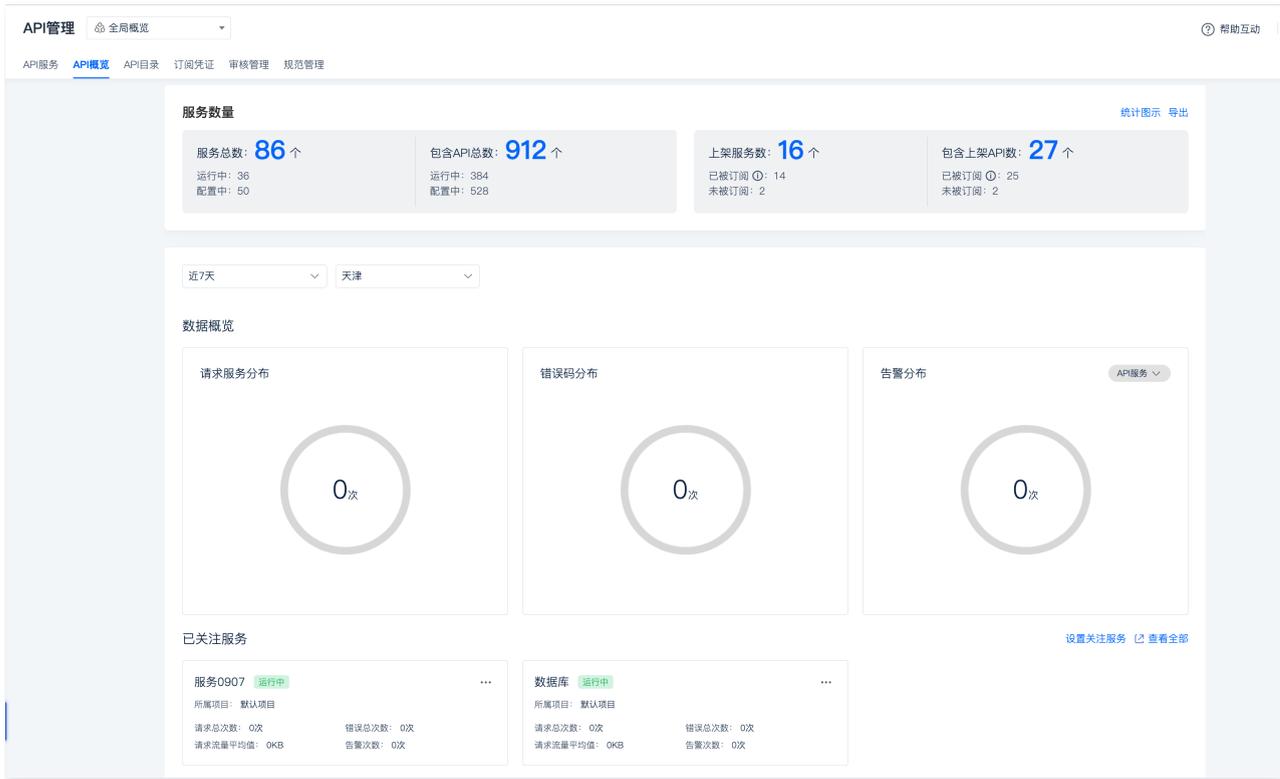
## 查看上线历史

发布后的 API 服务可以更改状态、环境等。在 **API 服务 > 操作 > 更多 > 查看发布历史**路径。此功能可查看到 API 服务发布后的历史情况（最多显示10条）。



## API 概览

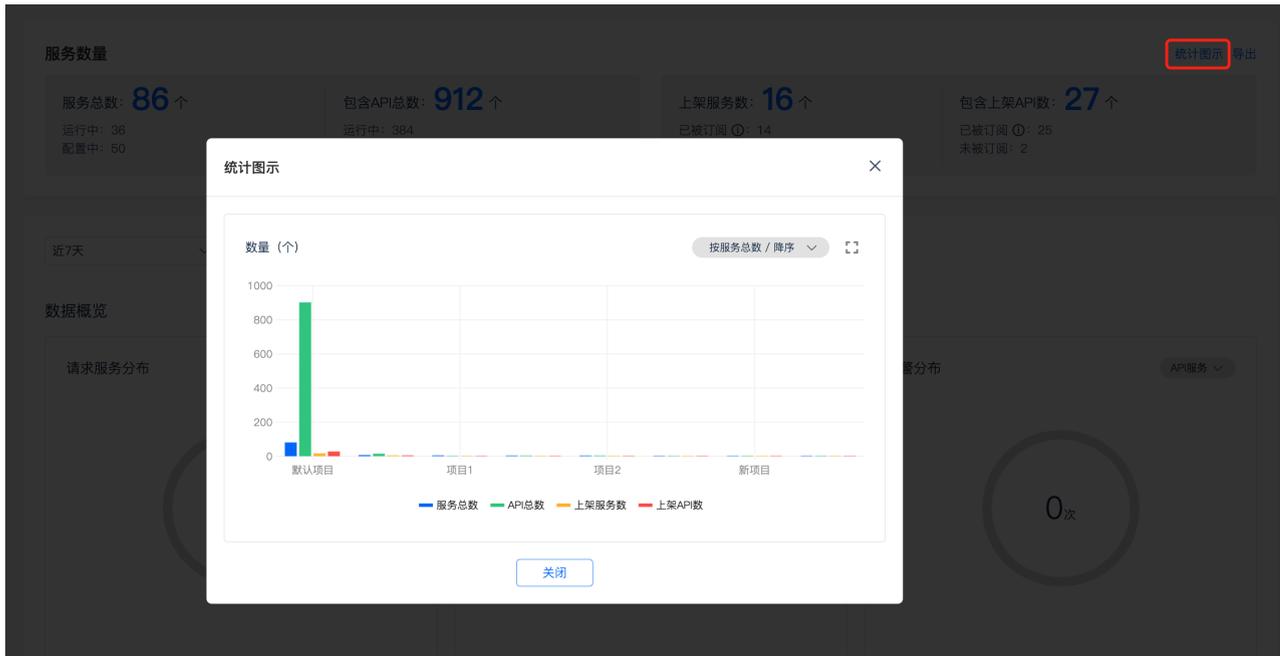
支持查看 API 概览，包括服务数量统计，指标数据概览和查看已关注服务。可以在左上角选择**全局概览**（仅系统管理员可以查看）或**项目概览**（仅系统管理员和对应项目管理员可以查看）。



如图为查看 API 服务的总数统计，可以看到每个项目下的服务总数、每个服务下的 API 明细。



也支持查看项目统计图示，以及单击导入导出项目、服务、API 明细列表。



支持设置关注服务，从而在 API 概览中直接查看其关键指标。

API管理 默认项目

API服务 API概览 API目录 订阅凭证 审核管理 规范管理

新建 导入 设置检查策略

筛选分组 请输入API服务名称

服务名称	状态	域名	分组	检查策略	描述	上架状态	操作
3434343	运行中	20231129_2.0 域名	默认分组	默认策略	-	未提交	编辑 新建API ...
脱敏api服务	运行中	20231124_3.0 域名	默认分组	默认策略	-	已上架	上架 下架
898	配置中	20231123_1.0 -	默认分组	默认策略	-	未提交	全局关注 项目关注 导出 设置检查策略 查看描述文件 查看上线历史 删除
数据库开放API服务	运行中	20231122_2.0 域名	默认分组	默认策略	-	未提交	
数据库开放API服务_	运行中	20231122_1.0 域名	默认分组	默认策略	-	未提交	
sap_rfc_服务_时间戳	运行中	20231109_1.0 域名	默认分组	默认策略	-	未提交	
数据库开放API服务2	配置中	20231122_1.0 -	默认分组	默认策略	-	未提交	
3434343-2	运行中	20231120_1.0 域名	默认分组	默认策略	-	未提交	编辑 新建API ...
66	运行中	20231114_1.0 域名	默认分组	默认策略	-	未提交	编辑 新建API ...
AI家-测试环境2222	配置中	20231114_1.0 -	默认分组	默认策略	AI家-测试环境	未提交	编辑 新建API ...

共 78 条

10条/页 跳至 1 页

## API 目录

API 目录展示已上架的 API 服务。类似一个 API 服务市场，上架后的服务，不局限项目维度，可以被当前主账号下的所有子账号查看、订阅并调用。此页面可通过 API 服务的属性快速搜索服务。同时，可申请订阅或取消订阅 API 服务。

API管理 帮助互动

API服务 API概览 **API目录** 订阅凭证 审核管理 规范管理

运行环境 全部环境 所属项目 全部项目 分组名称 请选择分组

上架时间 全部 服务名称 请输入服务名称 搜索 重置

API服务名称	运行环境	订阅状态	订阅API数	所属项目	分组	上架更新时间	操作
脱敏api服务	广州	已订阅	1/1	默认项目	默认分组	2023-11-24 16:38:08	取消订阅
服务1030	广州	已订阅	2/2	默认项目	默认分组	2023-10-30 15:17:22	取消订阅
服务1027	广州	已订阅	2/2	默认项目	默认分组	2023-10-27 14:48:29	取消订阅
99990	广州	已订阅	1/2	默认项目	默认分组	2023-08-29 11:02:05	申请订阅 取消订阅
数据库服务	广州	待审核	0/1	project0301	默认分组	2023-07-31 11:14:27	申请订阅
22223	广州	已订阅	1/2	默认项目	默认分组	2023-07-03 14:18:09	申请订阅 取消订阅
新api服务032343	广州	未订阅	0/1	project0301	默认分组	2023-06-02 11:24:25	申请订阅
222	天津	已订阅	1/1	默认项目	默认分组	2023-05-29 15:05:04	取消订阅
服务0315	天津	待审核	0/1	默认项目	默认分组	2023-03-15 19:21:05	申请订阅
222	广州	已订阅	3/3	默认项目	默认分组	2023-03-14 17:02:52	取消订阅

共 18 条 1 2 > 10条/页 跳至 1 页

### 申请订阅

申请订阅 API 服务时，需选择或新建订阅凭证。将凭证与 API 服务关联上。待系统管理员审核后，即可成功订阅。

API订阅申请 ×

API服务名称 22223

运行环境 广州

选择API 共2条, 已选0条 请输入API名称

<input type="checkbox"/>	API名称	状态	分组	请求路径	请求方法	描述
<input checked="" type="checkbox"/>	15	已订阅	默认分组	/	GET	-
<input type="checkbox"/>	2	未订阅	默认分组	/3	GET	-

订阅凭证 \* crm系统

描述 请输入描述内容

流量控制

脱敏策略 请选择脱敏策略

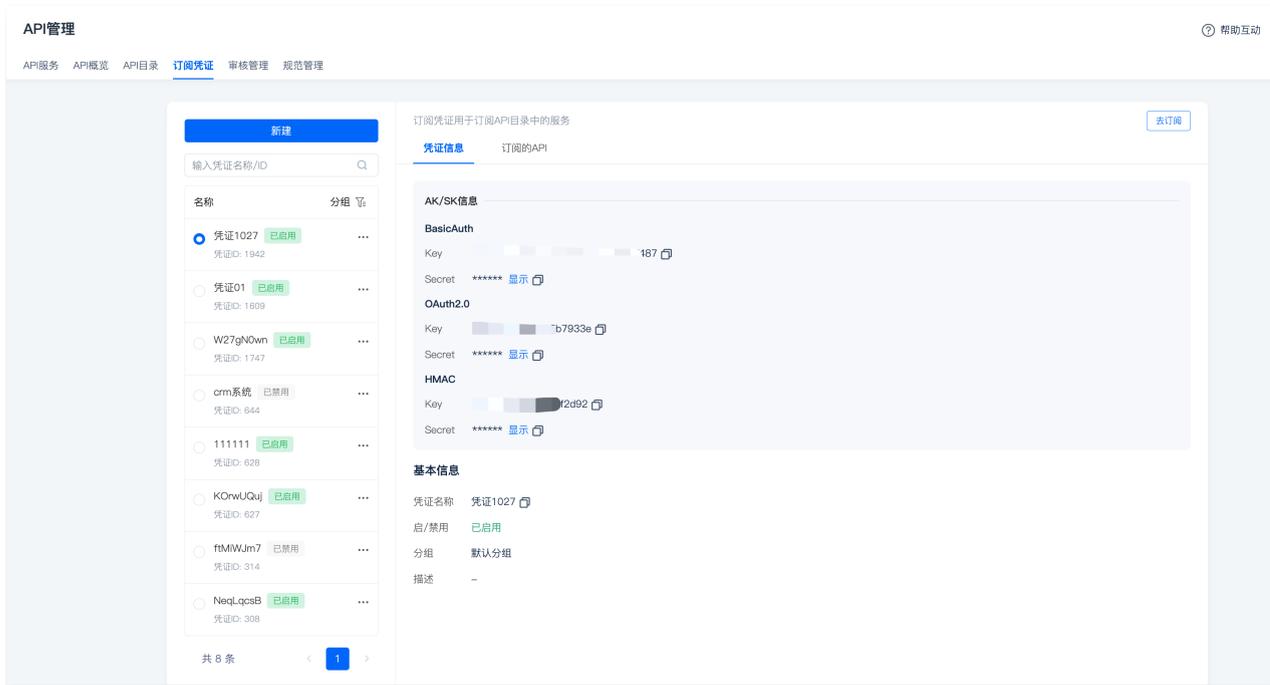
确认 取消

### 取消订阅

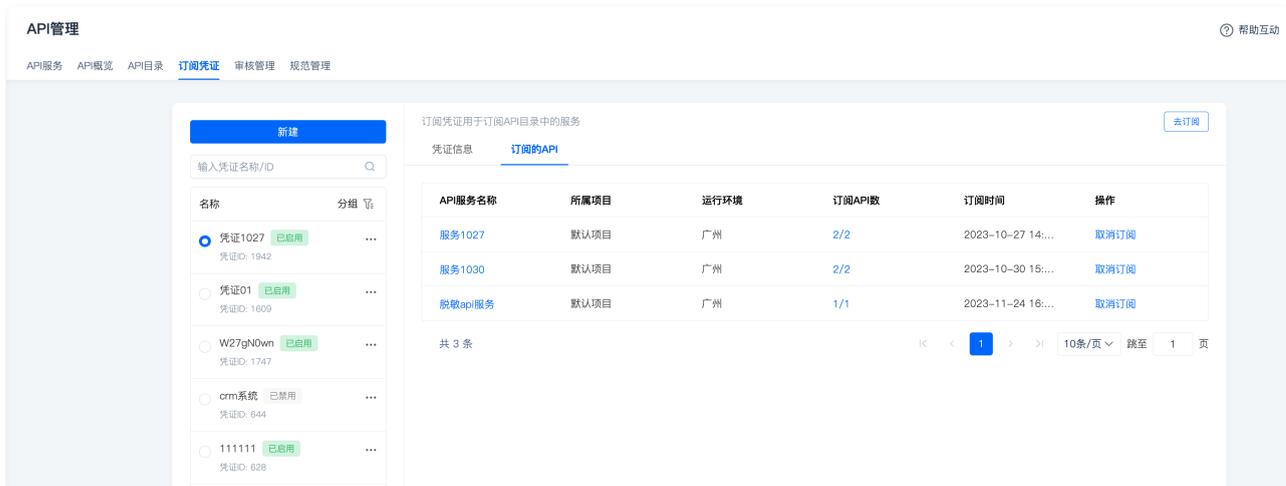
取消订阅后将不能调用该 API 服务，此操作无需系统管理员审核。

### 订阅凭证

此列表可展示或搜索所有的订阅凭证，同时，可新建凭证。订阅凭证用于订阅 API 目录中的服务。凭证即为某 API 服务的钥匙。在申请订阅 API 服务时将凭证与 API 服务关联上，调用的时候填写该凭证，即可成功调用该服务。同时，能看到该凭证各种鉴权类型的 Key 和 Secret，调用时直接复制即可使用。



一个凭证支持与多个 API 服务关联。可在订阅的 API tab 页查看此凭证关联的所有 API 服务。

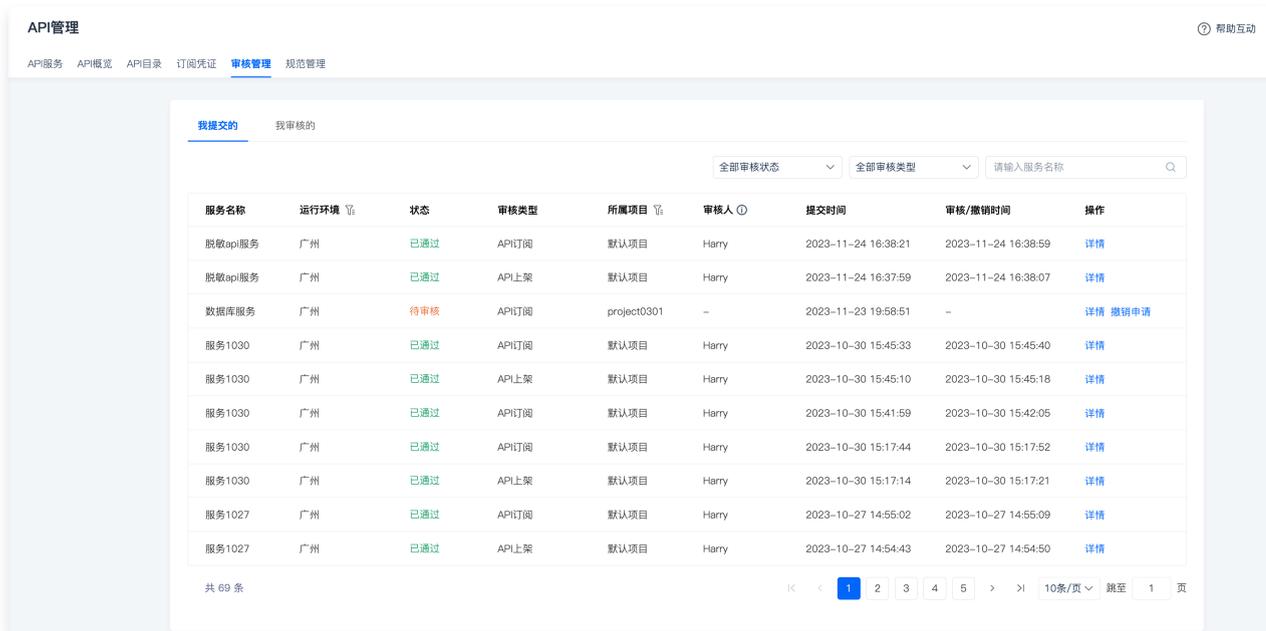


新建凭证时，自定义相关属性即可。

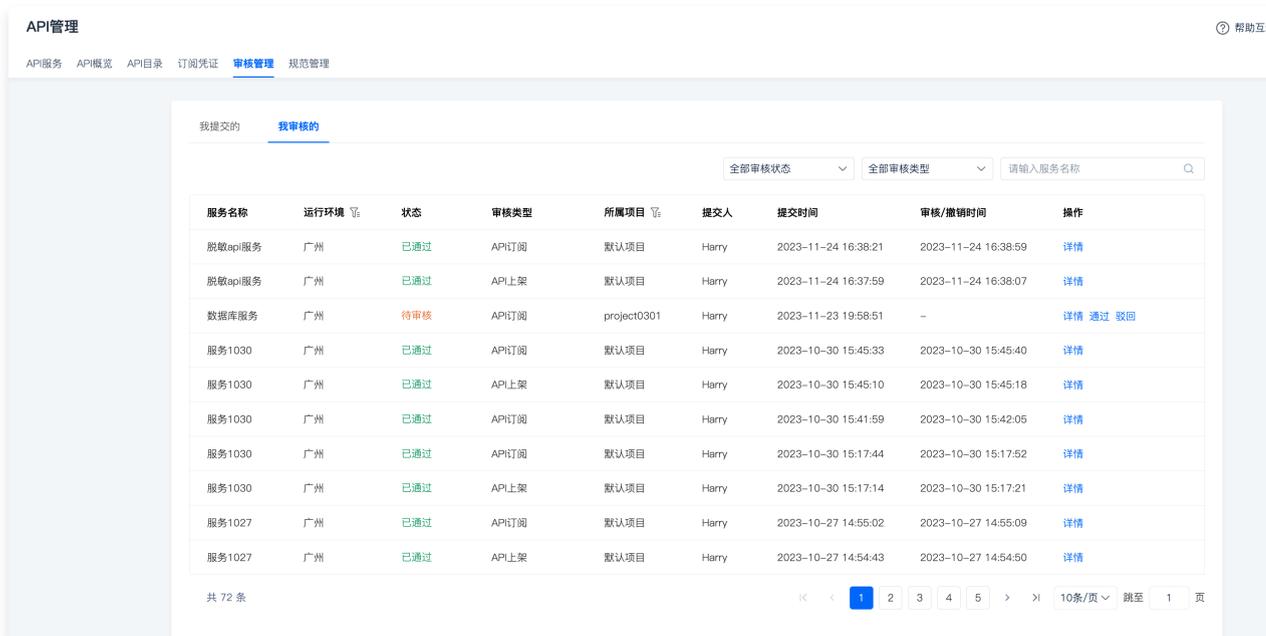
## 审核管理

审核管理分两个功能：我提交的和我审核的。涉及到审核的事项都在此功能页处理。

- 我提交的：展示个人提交的所有审核信息。全部角色都可见。



- 我审核的：此页面展示需审核的信息。只有系统管理员和项目管理员角色能看到。其他角色访问页面数据为空。

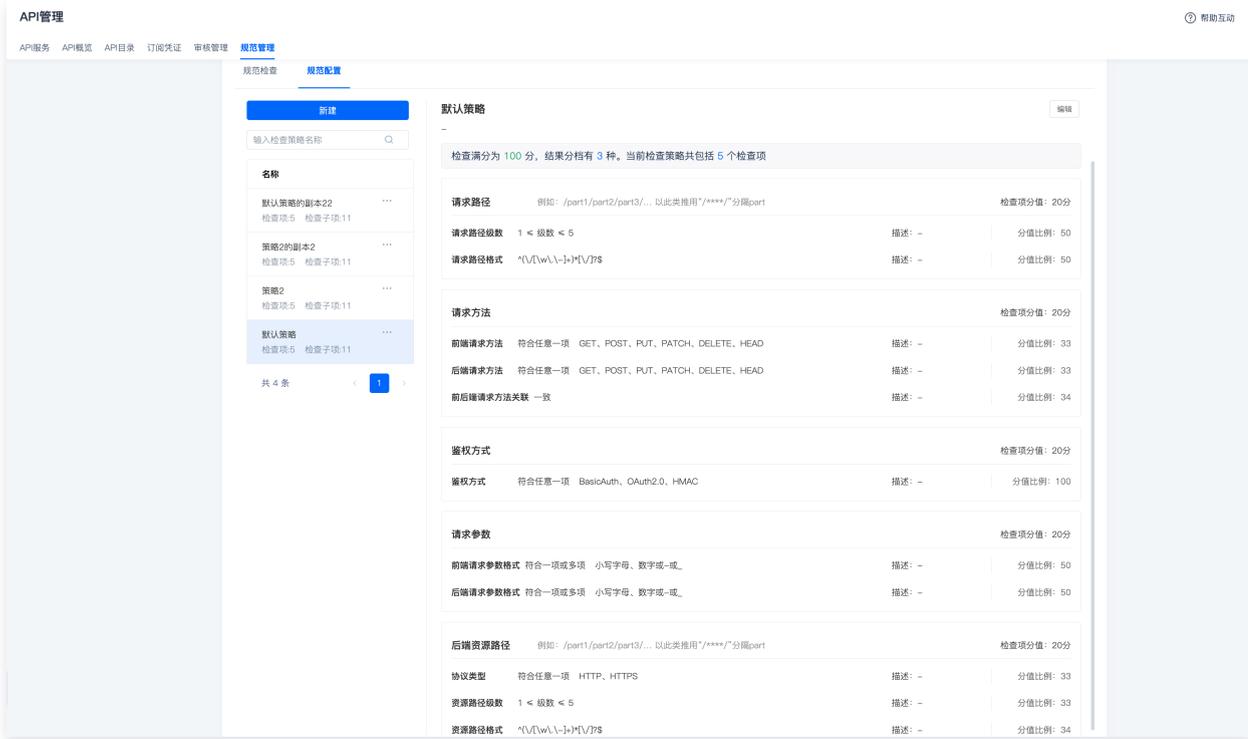


- 系统管理员审批全部项目 API 服务上下架或 API 服务订阅的请求。
- 项目管理员审批所在项目内的 API 服务上下架或 API 服务订阅的请求。

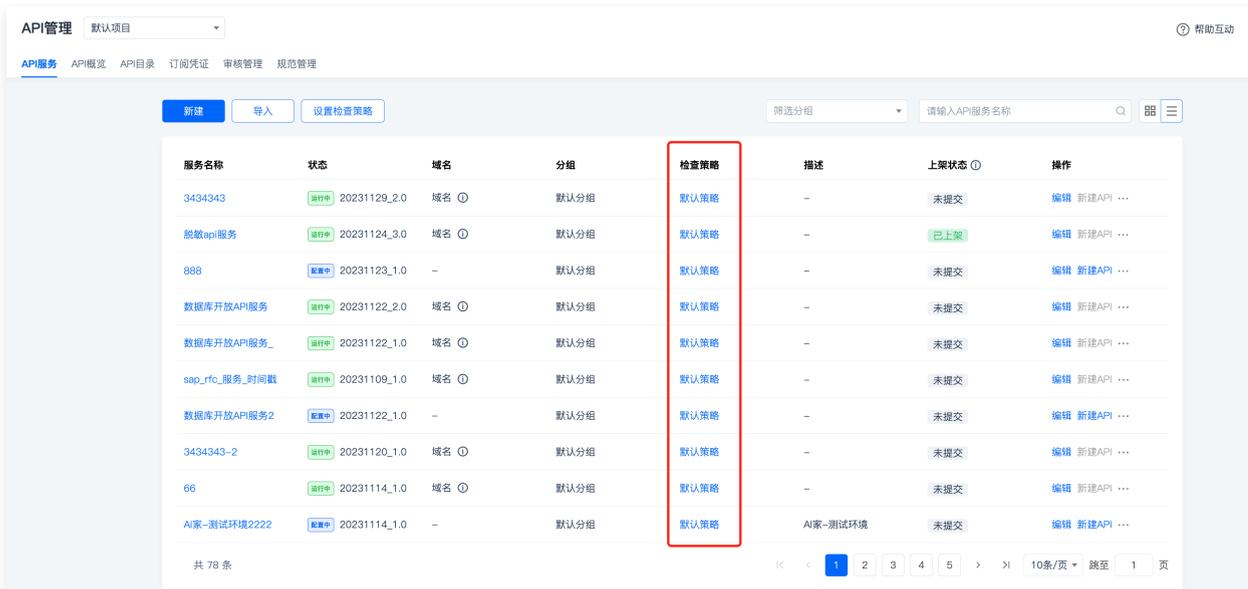
## 规范管理

支持系统管理员对全部 API 接口按所选检查策略进行规范检查，得出每一个 API 接口的检查明细、得分，并支持导出检查报告 Excel 文件，以作进一步分析。

- 规范配置：系统默认提供一个“默认策略”，其中包括针对请求路径、请求方法、鉴权方式、请求参数、后端资源路径的规范检查项配置。用户也可以自行新建检查策略。



- 系统默认会给所有API服务，都关联默认检查策略。



- 规范检查：

进入规范检查菜单，可以按指引启动规范检查，检查结束后可以查看检查列表，呈现每个 API 接口对应的待修正检查项，检查结果，结果分值。

API管理 帮助互动

API服务 API概览 API目录 订阅凭证 审核管理 **规范管理**

---

**规范检查** 规范配置

检查项目: 全部项目 重新检查 下载报告

检查时间: 2023-11-21 15:26:50

共检查916个API

完全规范率: 2%

● 完全规范: 22个
● 部分规范: 873个
● 不规范: 21个

[分档设置](#)

API名称	后端服务类型	所属项目	所属API服务	关联策略	待修正检查项	检查结果	结果分值
090	第三方服务	project0301	服务2	默认策略	6	不规范	46
API2-数据库服务	未配置	project0301	服务2	默认策略	4	不规范	56
API2-数据库服务	第三方服务	project0301	服务233	默认策略	7	不规范	36
API1-第三方服务	第三方服务	project0301	服务233	默认策略	6	不规范	46
23232	第三方服务	默认项目	1111111	默认策略	4	不规范	53
5566	第三方服务	默认项目	123456	默认策略	6	不规范	46
uploadFile	未配置	默认项目	sudouu	默认策略	5	不规范	46
deletePet	未配置	默认项目	sudouu	默认策略	5	不规范	46

支持单击待修正检查项，查看详细的不规范项内容。

API管理 帮助互动 Harry

API服务 API概览 API目录 订阅凭证 审核管理 **规范管理**

---

**规范检查** 规范配置

检查项目: 全部项目 重新检查 下载报告

检查时间: 2023-11-21 15:26:50

共检查916个API

完全规范率: 2%

● 完全规范: 22个
● 部分规范: 873个
● 不规范: 21个

API名称	后端服务类型	所属项目	所属API服务	关联策略	待修正检查项	检查结果
090	第三方服务	project0301	服务2	默认策略	6	不规范
API2-数据库服务	未配置	project0301	服务2	默认策略	4	不规范
API2-数据库服务	第三方服务	project0301	服务233	默认策略	7	不规范
API1-第三方服务	第三方服务	project0301	服务233	默认策略	6	不规范
23232	第三方服务	默认项目	1111111	默认策略	4	不规范
5566	第三方服务	默认项目	123456	默认策略	6	不规范

**待修正检查项 (090:6项)**

**请求方法** 前端方法: GET、后端方法: 未配置

未满足以下 2 项规范标准:

- 后端请求方法 符合任意一项 GET、POST、PUT、PATCH、DELETE、HEAD
- 前后端请求方法关联 一致

**鉴权方式** NoAuth

未满足以下 1 项规范标准:

- 鉴权方式 符合任意一项 BasicAuth、OAuth2.0、HMAC

**后端资源路径** 未配置

未满足以下 3 项规范标准:

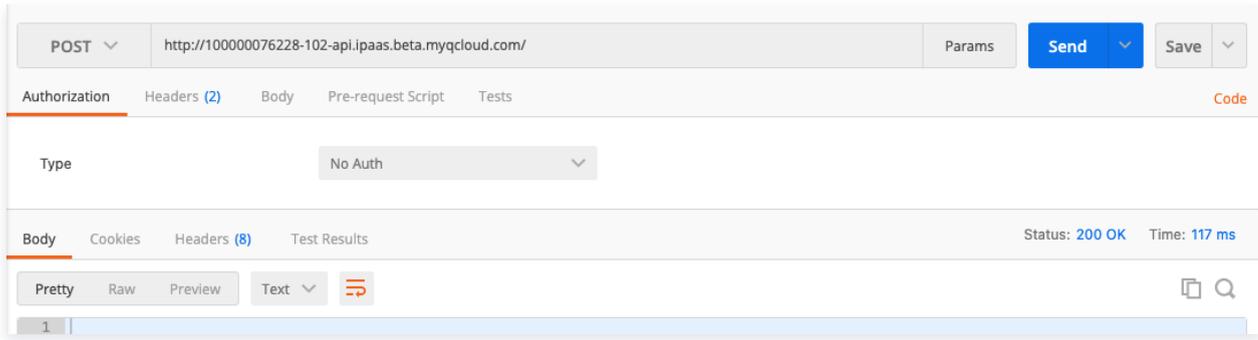
- 资源路径级数 1 < 级数 <= 5
- 资源路径格式 ^[\w\.\-]+\$
- 协议类型 符合任意一项 HTTP、HTTPS

还支持单击右上角导出检查报告，以及设置检查结果分档。

## API 调用步骤

### 从用户侧用不同鉴权方式调用 API (以 postman 为例)

- API 服务无需验证的情况:



• API 服务需要 Basic Auth 的情况:

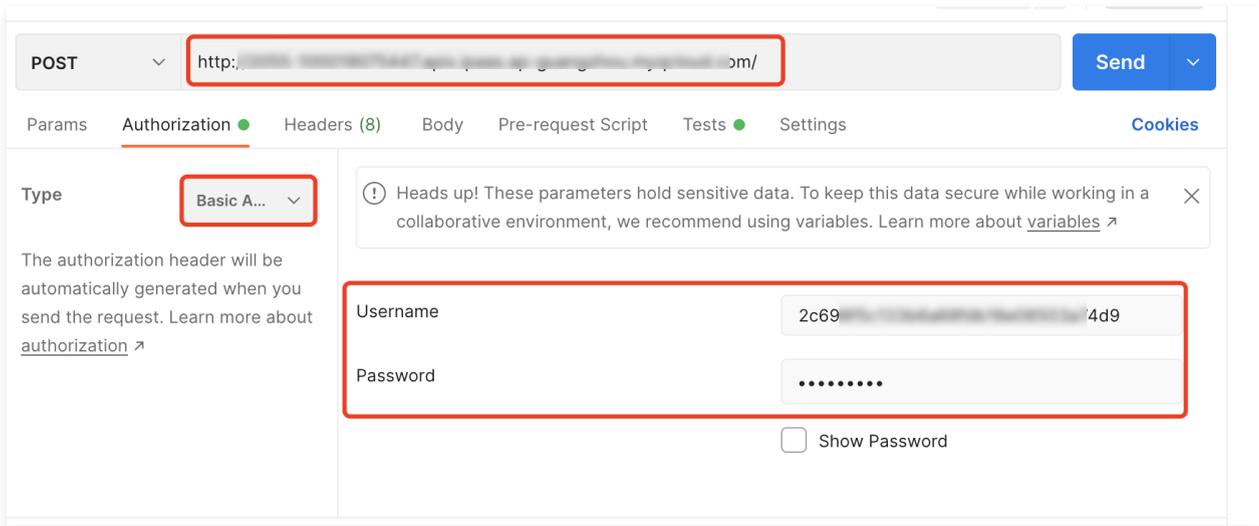
1.1 复制 API 的调用地址 (需先成功发布 API 服务):



1.2 进入 API 服务详情页面, 新建或打开已有调用凭证, 即可查看用于 Basic Auth 的 AK/SK 信息:



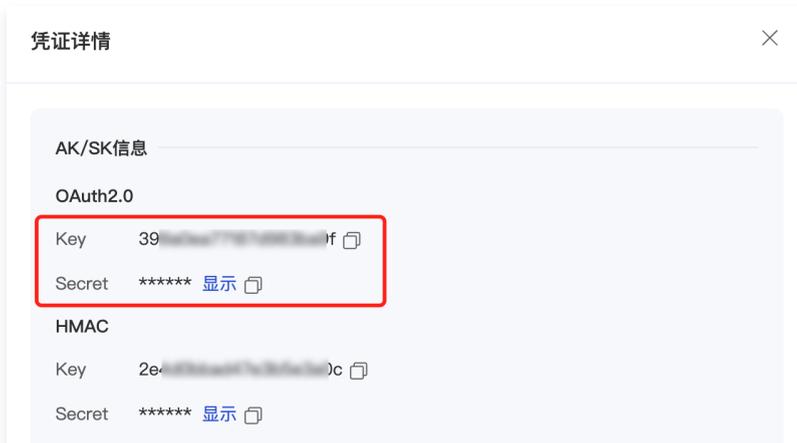
1.3 打开 postman, 将上述获取的 API 调用地址和用于 Basic Auth 的 AK/SK 信息分别填入:



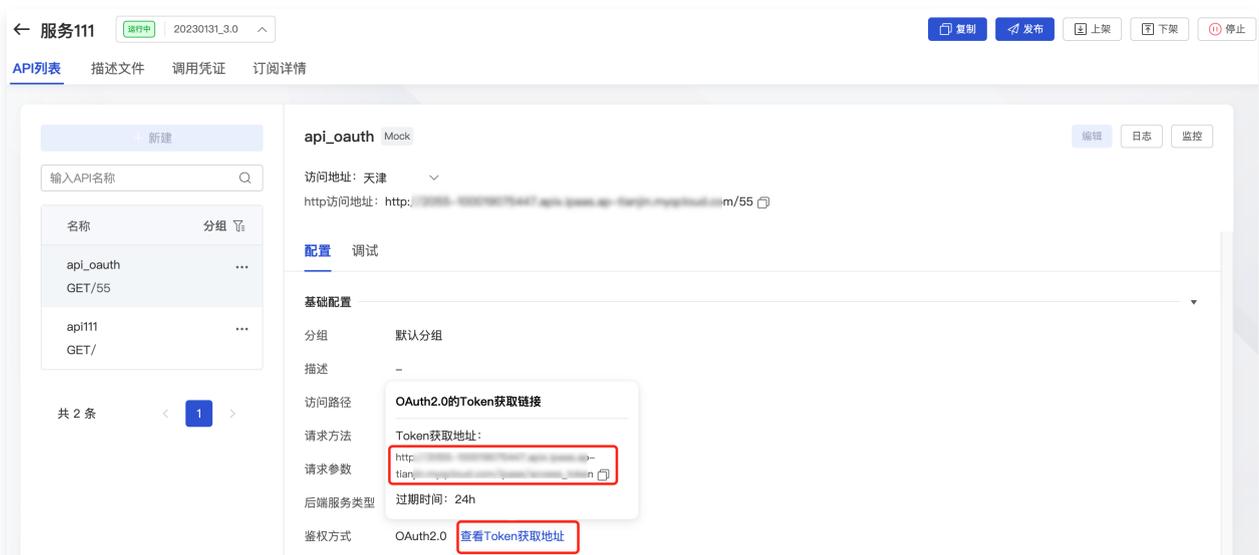
• API 服务需要 OAuth2.0 的情况:

1.1 复制 API 的调用地址 (需先成功发布 API 服务), 方法同上, 不再附图。

1.2 进入 API 服务详情页面, 新建或打开已有调用凭证, 即可查看用于 OAuth2.0 的 AK/SK 信息:



1.3 进入 API 服务详情页面, 单击并复制对应 API 的 Token 获取链接。

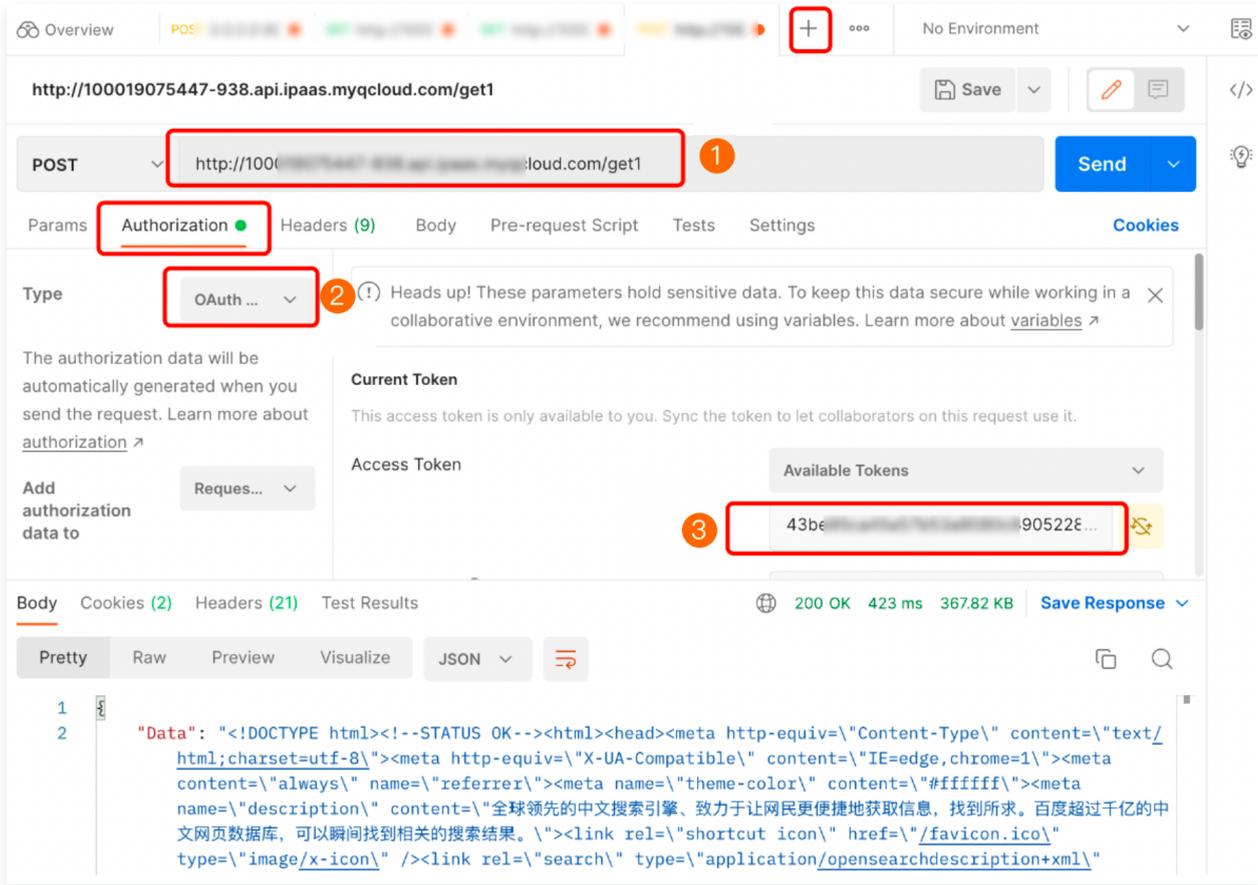


1.4 在 postman 中创建一个新的请求, 来获取 accsee\_token。

1.4.1 首先, 在输入栏中输入第3步获取的 Token 获取链接, 并选择 GET 请求方法。

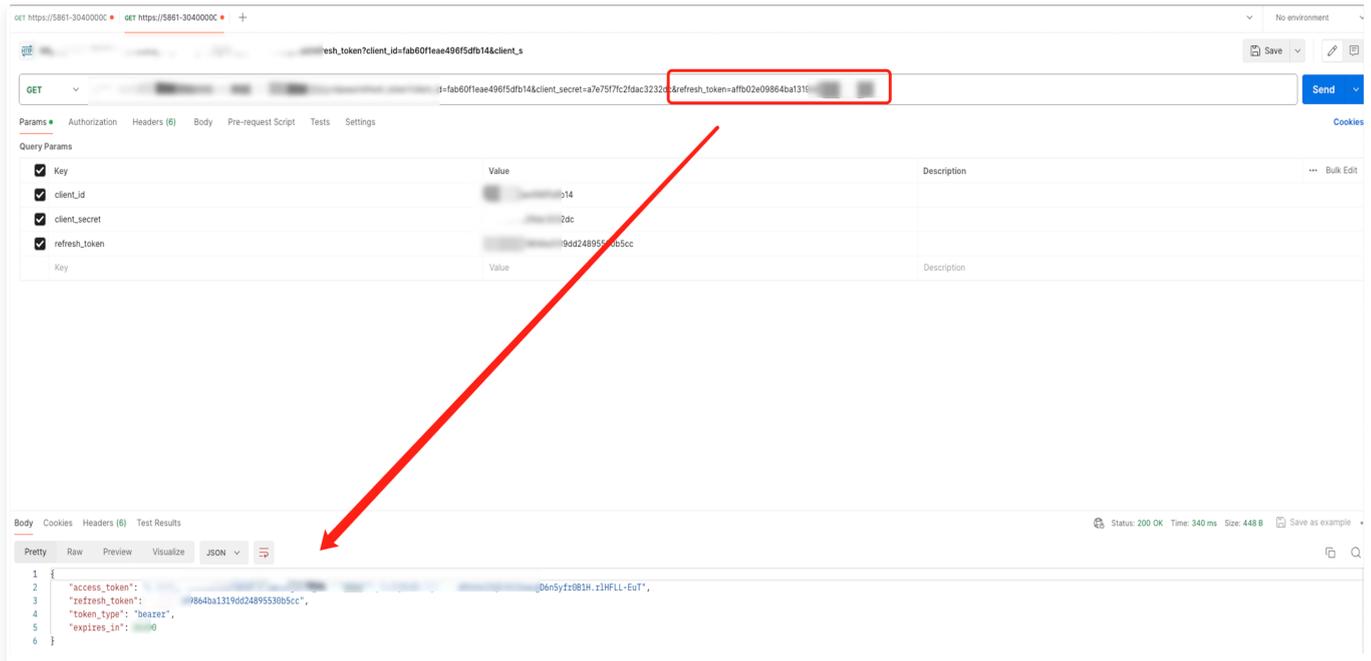
1.4.2 其次, 选择 Params 标签页, 分别输入第2步获得的用于 OAuth2.0 的 AK/SK 信息 (格式见下图)。





1.6 当 access\_token 过期 (24h) 之后, 需要借助 refresh\_token 获取新的 access\_token:

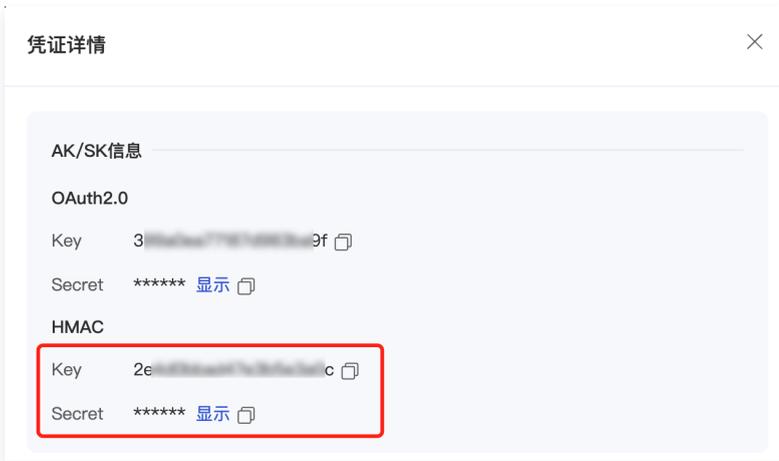
在1.4步骤中 access\_token 接口的返回值中会包含 refresh\_token, 利用 refresh\_token 请求 refresh\_token 接口, 可以拿到最新的 access\_token.



• API 服务需要 HMAC Auth 的情况:

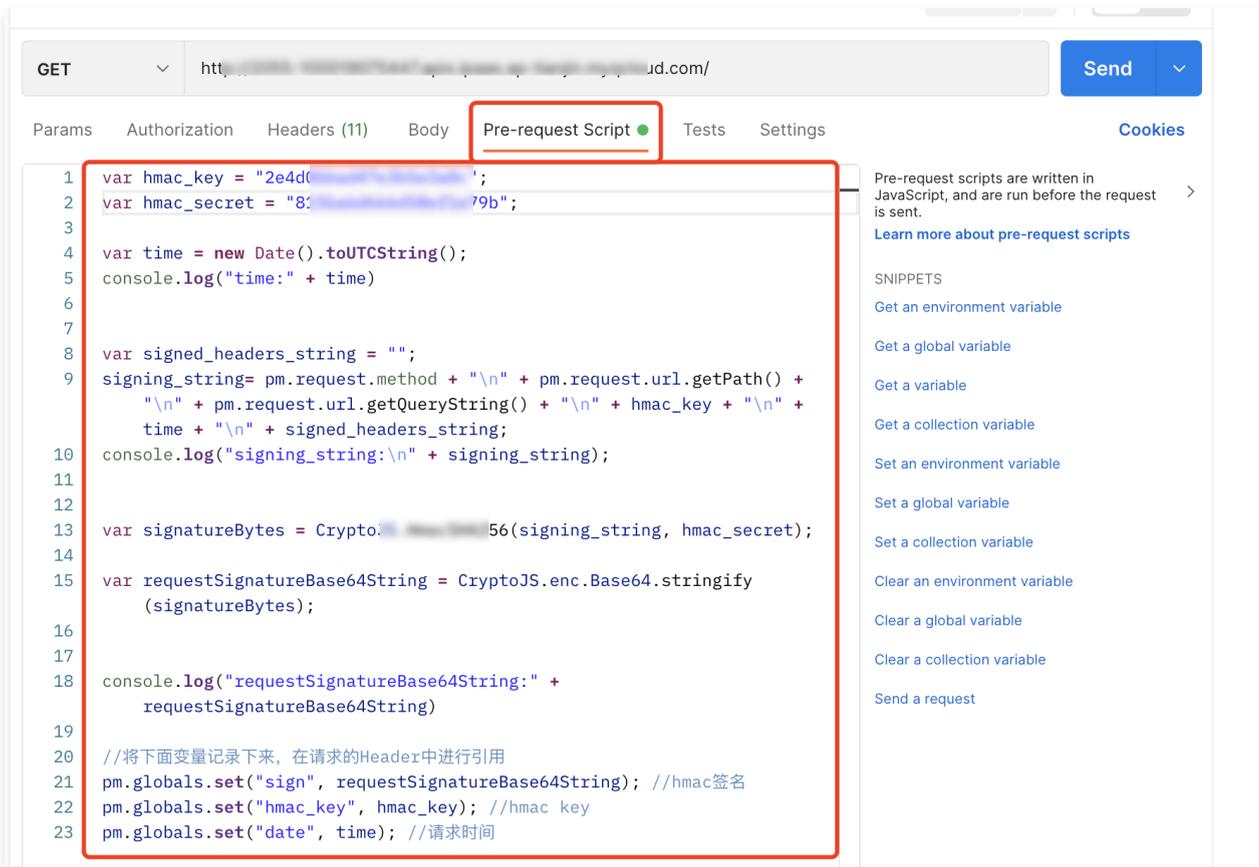
1.1 复制 API 的调用地址 (需先成功发布 API 服务), 方法同上, 不再附图。

1.2 进入 API 服务详情页面, 新建或打开已有调用凭证, 即可查看用于 HMAC 的 AK/SK 信息:



1.3 打开 postman, 在输入框填入第1步获取的 API 调用地址。

1.4 将 postman 切换到 Pre-request Script 标签下, 粘贴下方代码段 (注意用第2步获取的 HMAC 的 Key 和 Secret, 分别来替换代码段中的 hmac\_key 和 hmac\_secret 变量值)。

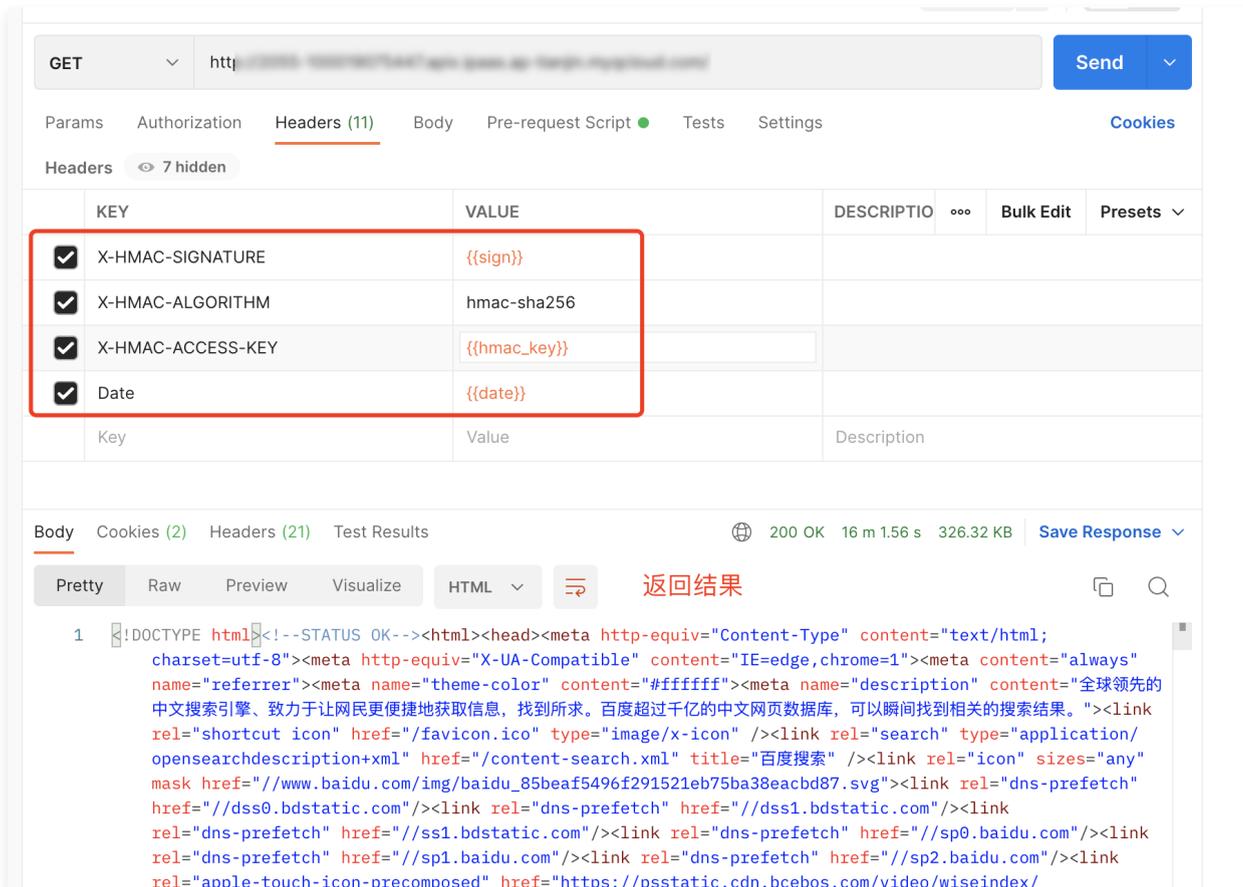


```

var hmac_key = "2e4d0bbad47e3b5e3a0c";
var hmac_secret = "815ba6d666d58ef1e79b";
var time = new Date().toUTCString();
console.log("time:" + time)
var signed_headers_string = "";
signing_string= pm.request.method + "\n" + pm.request.url.getPath() + "\n" +
pm.request.url.getQueryString() + "\n" + hmac_key + "\n" + time + "\n" +
signed_headers_string;
console.log("signing_string:\n" + signing_string);
var signatureBytes = CryptoJS.HmacSHA256(signing_string, hmac_secret);
  
```

```
var requestSignatureBase64String = CryptoJS.enc.Base64.stringify(signatureBytes);
console.log("requestSignatureBase64String:" + requestSignatureBase64String)
//将下面变量记录下来，在请求的Header中进行引用
pm.globals.set("sign", requestSignatureBase64String); //hmac签名
pm.globals.set("hmac_key", hmac_key); //hmac key
pm.globals.set("date", time); //请求时间
```

1.5 将 postman 切换到 Headers 标签下，输入下图中的4个 KEY-VALUE 对。最后，单击 send，即可看到调用 API 的返回结果。



# 运维中心

## 监控统计

最近更新时间：2024-08-01 15:33:01

### 操作场景

运行监控集合了集成应用维度、集成流、连接器维度、组件维度的基本运行情况概览、错误统计的数据上报、API 服务和单个 API 路径维度的监控数据，您可以查看某个集成流在一段时间内的实时及历史运行情况。

关于监控指标中各参数的说明请参见 [监控指标名词解释](#)。

#### 说明：

目前腾讯轻联的监控统计以项目维度划分，查看监控请先选择对应项目。

### 操作步骤

#### 监控管理

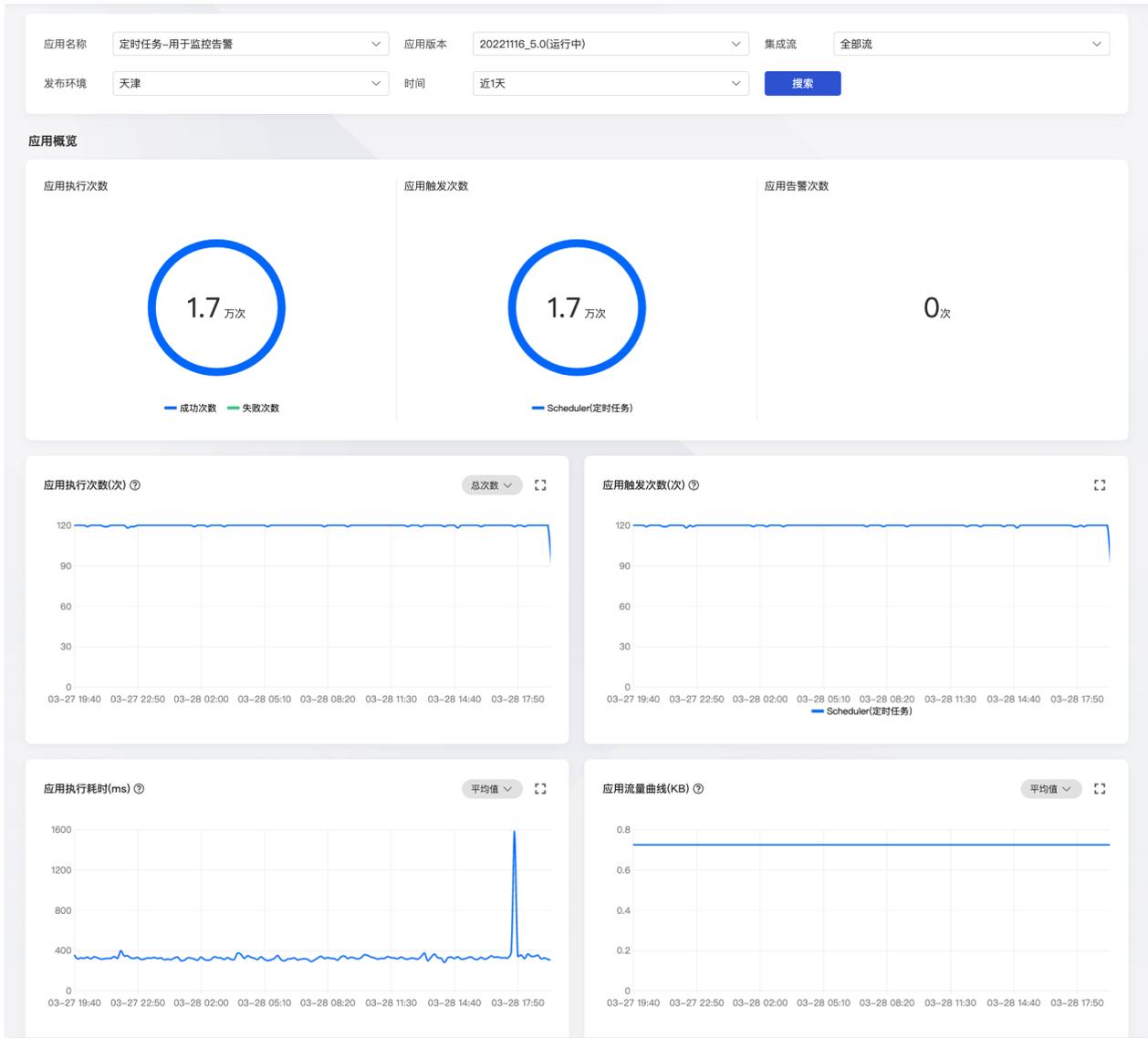
支持查看实时和历史监控数据，可以通过监控数据分析对应流的运行情况。默认展示当天集成应用的监控概览，若需要查看历史数据，可自定义时间维度后选择对应应用或单击搜索进行查看。默认展示最近应用的近一天的概览情况。

#### 注意：

目前控制台支持检索近30天内的监控。若有检索更早期日志的需求，请 [提交工单](#) 联系我们。

#### 应用概览数据查询

1. 登录 [腾讯轻联控制台](#) > [监控管理](#) > [监控管理](#)。
2. 在 [监控管理](#) 页面，默认选中全部流即为应用概览数据，支持查看某个应用的应用触发次数、应用执行次数、应该告警次数、应用执行耗时（ms）、应用流量曲线（kb）等应用维度指标。



3. 搜索应用监控数据说明：支持按应用名称、版本、集成流（必选全部流）、运行环境、时间等维度来进行搜索。

单击时间下拉，将显示设置搜索时间的面板，支持近5分钟、近15分钟、近1小时、近6小时、近1天、近3天、近7天、今天、昨天、前天、本周、自定义时间区间的时间设置。通过预定义的常用的时间段可快速此区间的监控数据。

监控管理 默认项目

[监控概览](#) 错误管理 API监控 投递管理

应用名称 服务0315\_数据库应用 应用版本 20230315\_2.0(运行中) 集成流 全部流

发布环境 天津 时间 近1天 搜索

应用概览

应用执行次数

0次

最近时间	相对时间
近5分钟	今天
近15分钟	昨天
近30分钟	前天
近1小时	本周
近6小时	

自定义时间

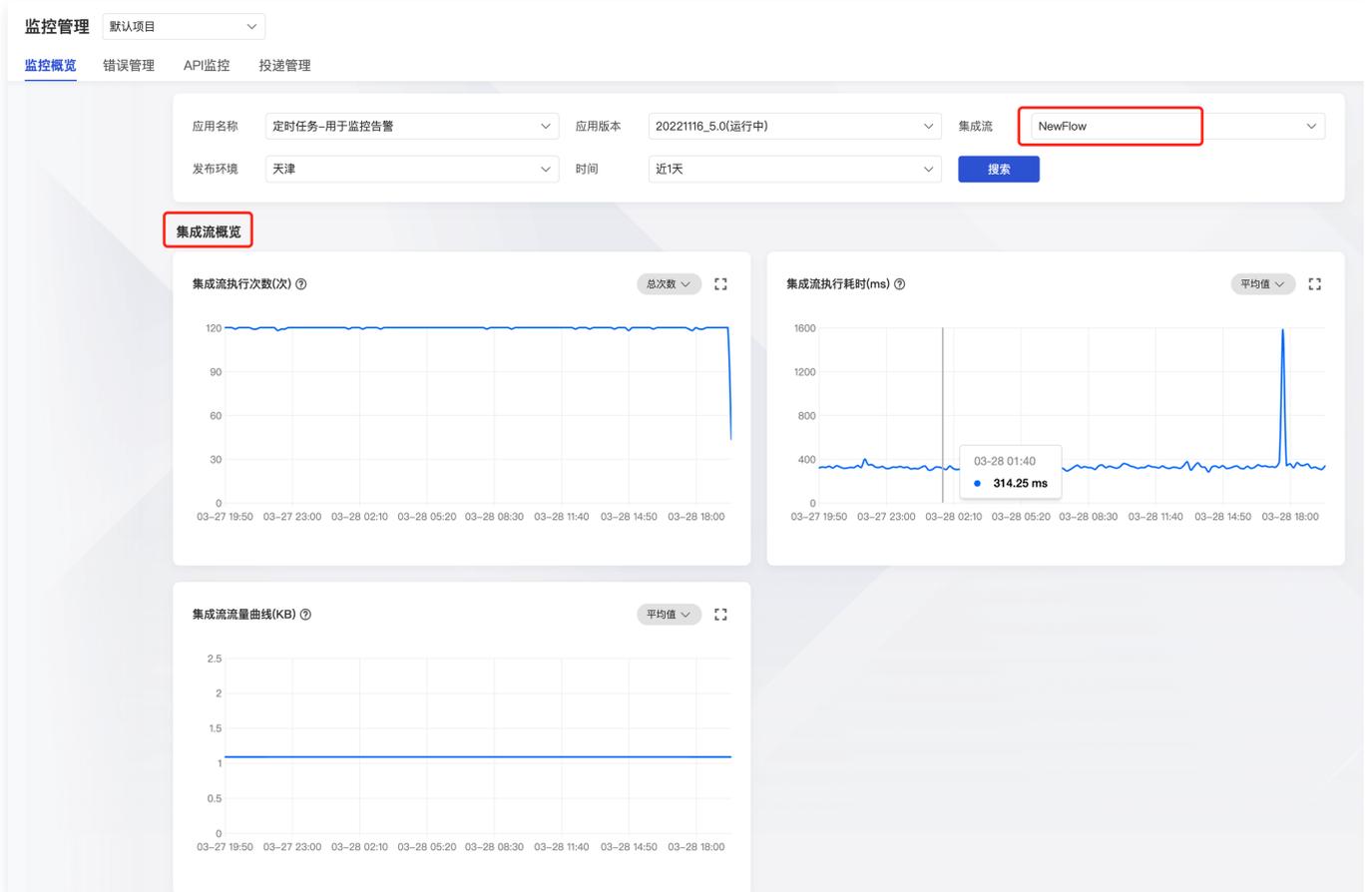
2023-03-27 19 : 43 : 28 - 2023-03-28 19 : 43 : 28

仅支持查看近7天的监控数据

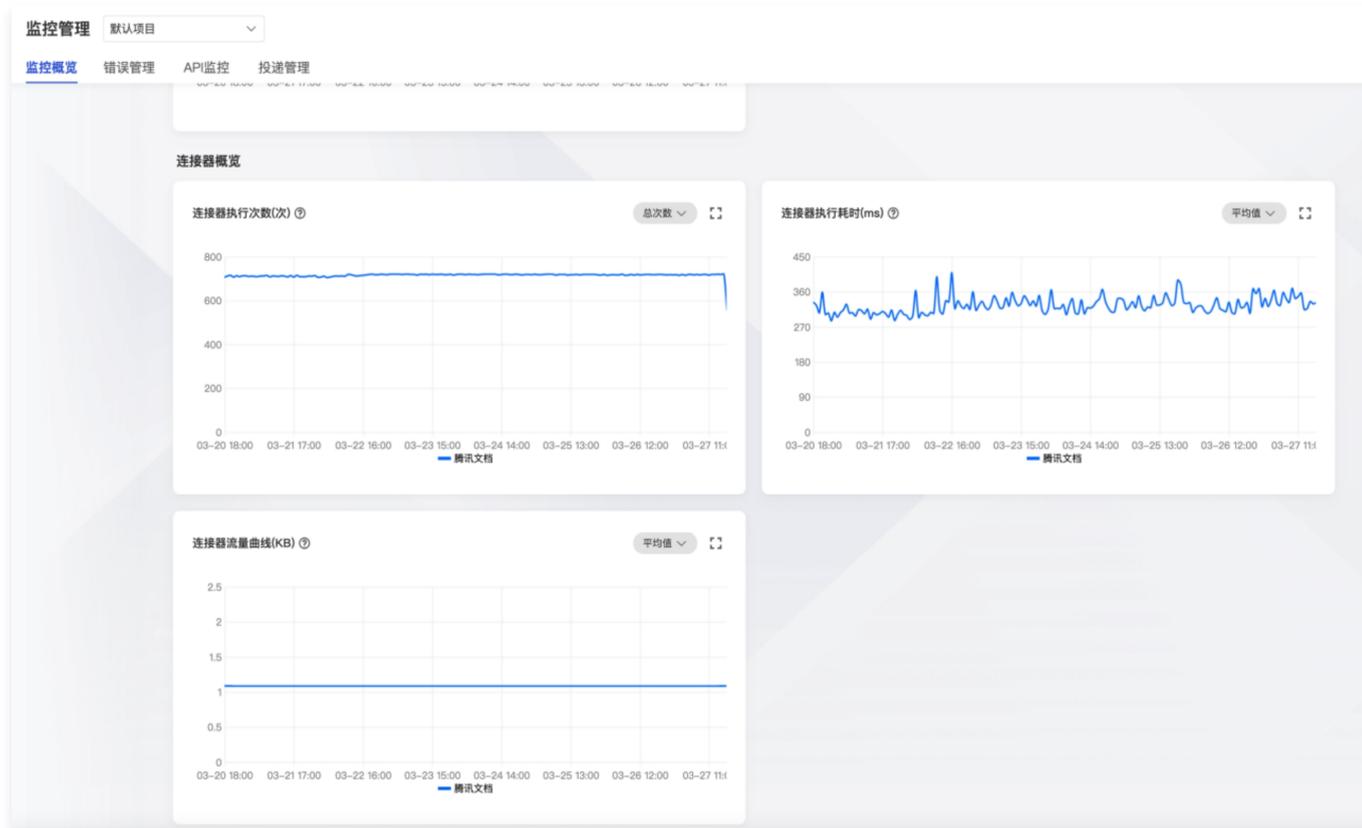
确定 取消

### 集成流数据查询

1. 登录 [腾讯轻联控制台](#) > [监控管理](#) > [监控管理](#)。
2. 在[监控管理](#)页面，选择应用中的某条集成流即可查询集成流概览数据及连接器概览数据。
  - 集成流概览：支持查看某个集成流的集成流执行次数（总次数/成功次数/失败次数）、集成流执行耗时（ms）、集成流流量曲线（kb）等集成流维度指标。



- 连接器概览：支持查看集成流中连接器的连接器执行次数（总次数/成功次数/失败次数）、连接器执行耗时（ms）、连接器流量曲线（kb）等连接器维度指标。



- 搜索集成流监控数据说明：支持按项目、应用、版本、集成流（必选某个集成流）、运行地域、时间等维度来搜索查看指定集成流的集成流概览和连接器概览。

单击时间下拉，将显示设置搜索时间的面板，支持近5分钟、近15分钟、近1小时、近6小时、近1天、近3天、近7天、今天、昨天、前天、本周、自定义时间区间的时间设置。通过预定义的常用的时间段可快速此区间的监控数据。

## 错误管理

### ⚠ 注意：

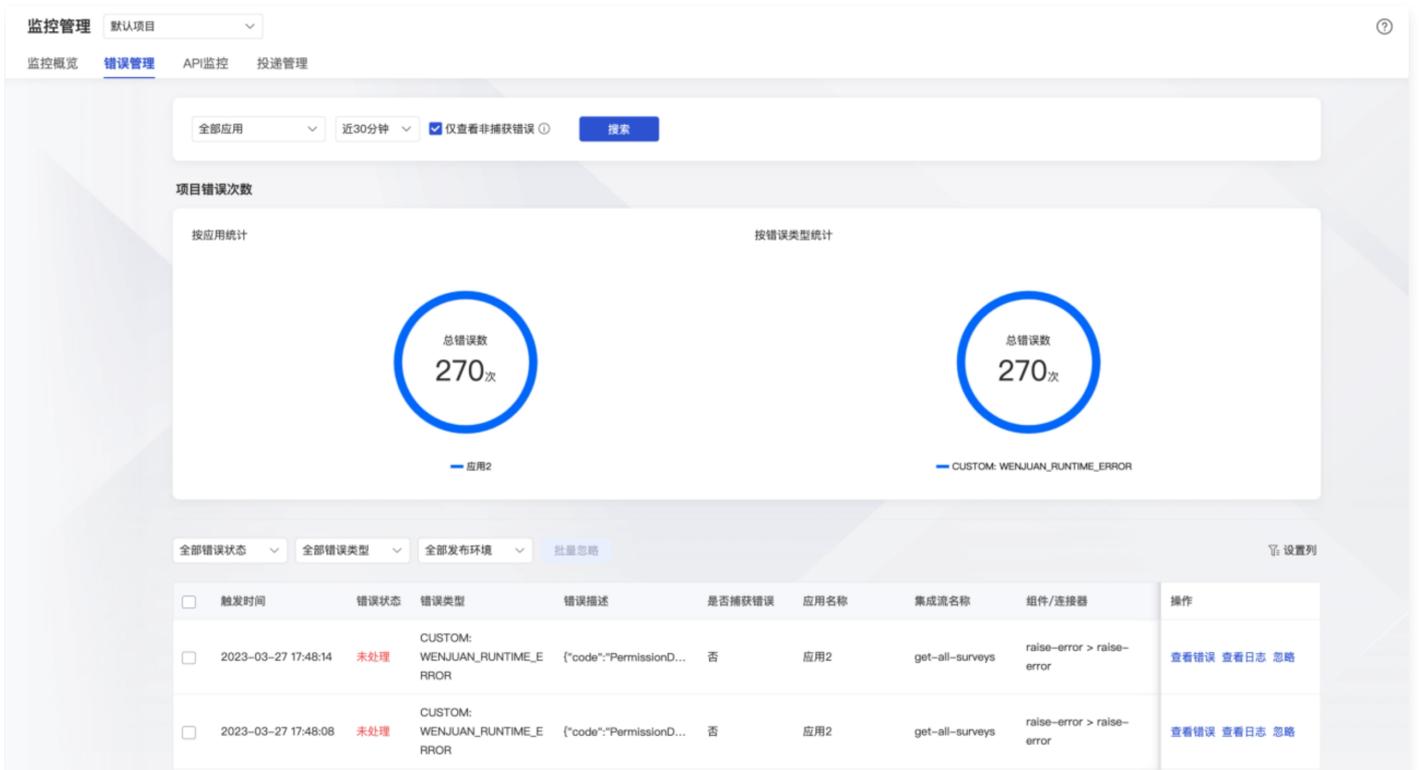
目前控制台支持检索近一个月的错误统计内容。若有检索更早期日志的需求，请 [提交工单](#) 联系我们。

## 查看错误统计列表

支持对运行失败的应用的错误次数和错误类型等数据进行集中展示及快速处理。

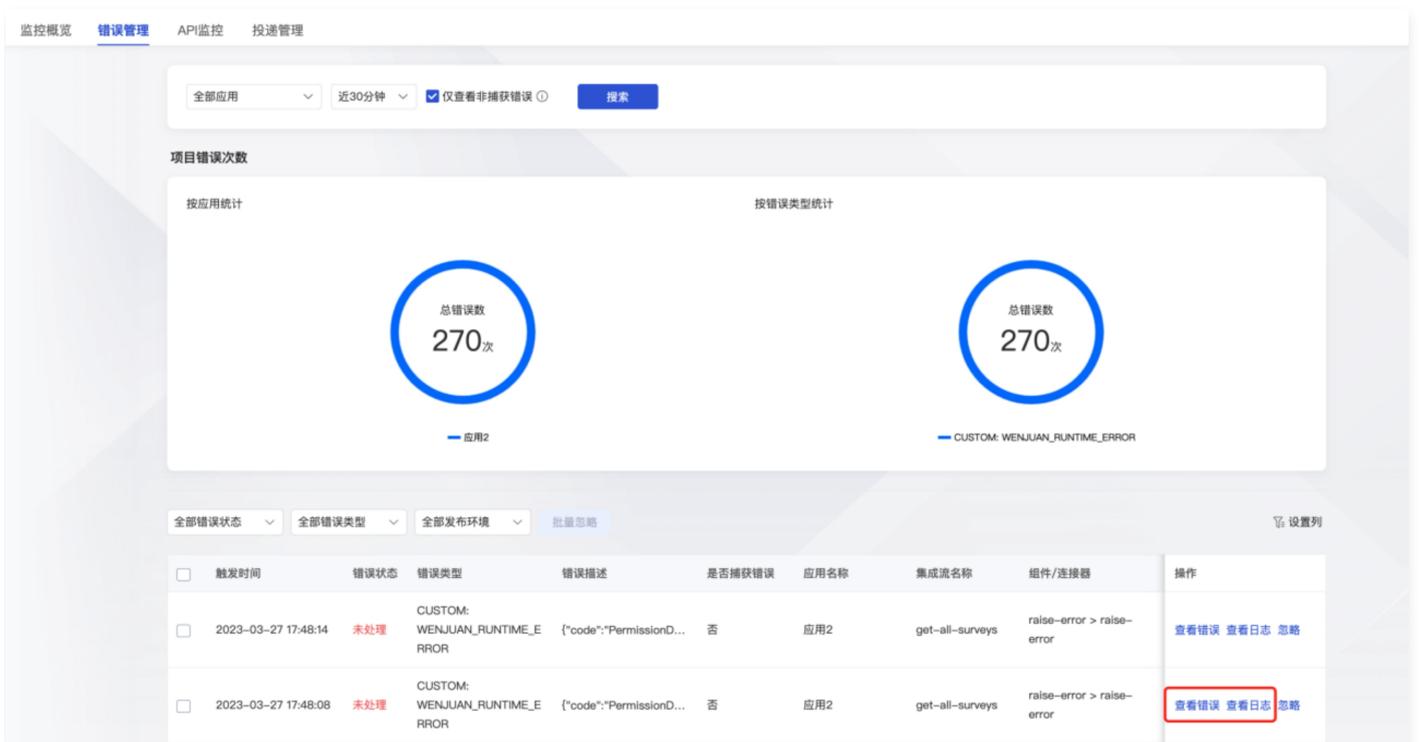
支持通过项目、应用、错误状态、环境、集成流、错误类型、时间等维度来进行搜索。列表如下：

- 触发时间：集成流触发报错时的具体时间。
- 错误状态：已处理的错误不在显示，只展示未处理或已忽略。
- 错误类型：将常见的报错分为大类，方便根据错误类型筛选错误流。
- 错误描述：进一步解释报错的详细原因，方便尽快找到错误根源进行处理。
- 是否捕获错误：是否为“捕获错误”组件捕获的错误。
- 应用名称：出现错误的名称。
- 集成流名称：出现错误的集成流名称。
- 组件/连接器：出现错误的组件/连接器名称。
- 环境：报错的应用所属的具体环境/地域。方便尽快定位出现错误的环境。



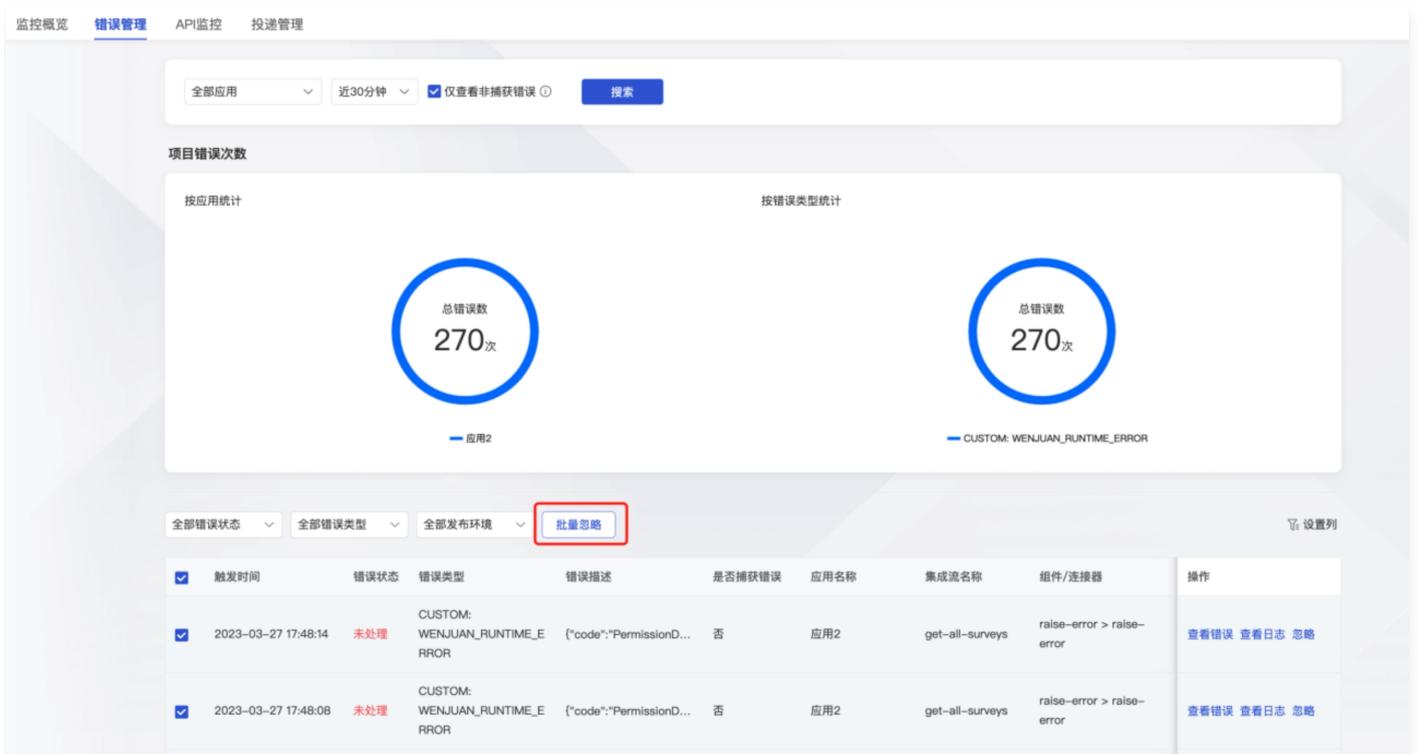
### 查看错误

单击查看错误将快速跳转到对应报错的集成流界面可快速定位到出现错误的集成流。



### 忽略/批量忽略

忽略错误的操作，若某个集成流的错误已处理完成，其余相邻时间段内的相同报错可单击对应错误条目的忽略，即可忽略此错误。或者选中对应错误条目后单击批量忽略。

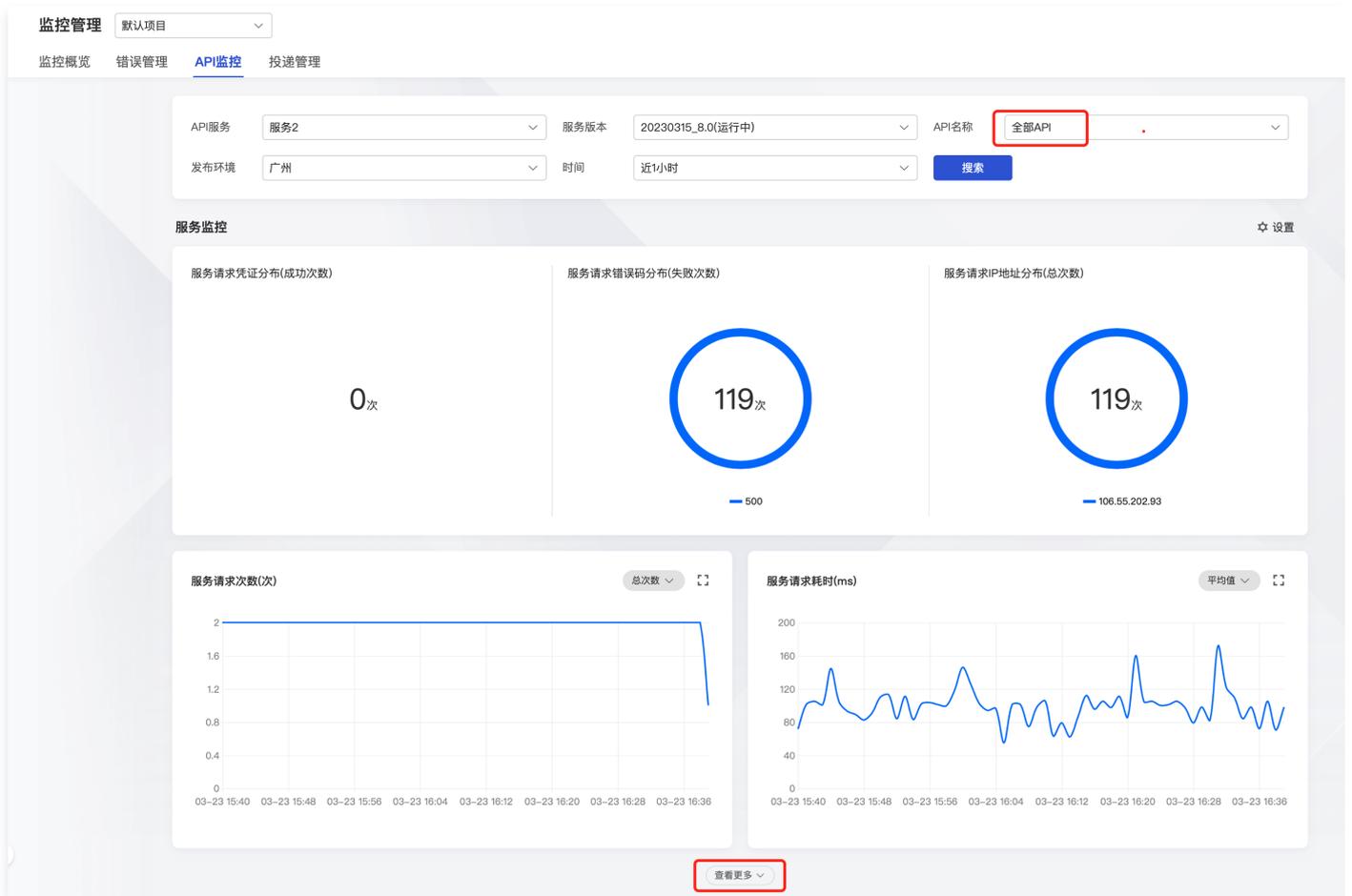


## API 监控

支持查看 API 历史监控数据，可以通过监控数据分析对应 API 的运行情况。默认展示近1天的 API 监控概览，若需要查看历史数据，可自定义时间维度后选择对应应用或单击搜索进行查看。

### 服务维度、API 维度监控数据查询

1. 登录 [腾讯轻联控制台](#) > [监控管理](#) > [API 监控](#)。
2. 在 [API 监控](#) 页面，默认通过扇形图展示所选服务维度的服务请求凭证分布（成功次数）、服务请求错误码分布（失败次数）、服务请求 IP 地址分布（总次数）。除此之外还支持单击设置，展示服务请求终端分布（总次数）扇形图。



同时通过折线图展示所选服务维度的服务请求次数（次）、服务请求耗时（ms）、服务请求凭证分布（次）、服务请求流量曲线（KB）、服务应答流量曲线（KB）和服务错误次数（次）。其中针对请求次数和请求凭证分布指标可以展示总次数、成功次数、失败次数。针对请求耗时、请求流量曲线、应答流量曲线指标可以展示最小值、最大值、平均值、P50、P95、P99、P99.9分位值。针对错误次数指标，支持统计 4xx 错误和 5xx 错误。下同。



最后，还支持通过列表直观展示 API 监控指标，并且支持单击设置来选择展示更多指标。

查看更多

服务/API名称	请求总次数(次)	请求耗时平均值(ms)	请求流量平均值(KB)	应答流量平均值(KB)	错误总次数(次)	4xx错误次数(次)	5xx错误次数(次)
▼ 新api服务0323	25	71.333336	595.3333	477.66666	24	24	0
集成流api	24	0	603	263	24	24	0
第三方服务api	1	214	580	907	0	0	0

共 2 条

10条/页 跳至 1 页

当在筛选栏中选择一个 API 时，可以查看针对所选单个 API 维度的监控数据。指标类别同服务维度，不再赘述。

监控管理 默认项目

监控概览 错误管理 **API监控** 投递管理

API服务: 新api服务0323 服务版本: 20230323\_4.0(运行中) API名称: **第三方服务api GET /3**

发布环境: 广州 时间: 近30分钟 搜索

API监控 设置

API请求凭证分布(成功次数)

1次

— NoAuth

API请求错误码分布(失败次数)

0次

API请求IP地址分布(总次数)

1次

— 119.147.10.193

API请求次数(次) 总次数

API请求耗时(ms) 平均值

查看更多

搜索监控数据说明：支持按服务名称、服务版本、API 路径名称、运行环境、时间等维度来进行搜索。

单击时间下拉，将显示设置搜索时间的面板，支持近5分钟、近15分钟、近1小时、近6小时、近1天、近3天、近7天、今天、昨天、前天、本周、自定义时间区间的时间设置。通过预定义的常用的时间段可快速此区间的监控数据。

## 投递管理

如果用户需要在控制台之外查看或处理集成应用监控数据，我们支持将集成应用监控数据投递到 Prometheus 平台，从而实现对监控数据做后续分析和处理。

用户在创建一个投递任务时，需要选择被投递的集成应用（最多10个）。并且完成目标集群配置，支持进行 BasicAuth 鉴权和免鉴权。

← 投递任务信息
?

**监控投递**

任务名称 \*

投递类型 \* 投递到Prometheus

**源配置**

投递对象 \*

集成应用	发布环境	操作
<input type="text" value="请选择应用"/>	<input type="text" value="请选择环境"/>	自
+ 添加投递对象 (1/10)		

**目标配置**

集群地址 \*

鉴权方式 \* 免认证 BasicAuth

用户名 \*

密码 \*

保存
取消

## 监控指标名词解释

指标类别	指标名称	指标详解
应用	应用监控概览	以应用维度对数据进行聚合（选中全部流默认展示应用维度），可以查看选定应用的符合筛选条件的对应数据和图表信息。
	应用触发次数（次）	统计某应用下全部流被触发的次数之和，涵盖 Scheduler、HTTP Listener、Kafka Consumer 等触发器。触发一次会执行所引用的所有流，因此应用触发次数不超过应用执行次数。
	应用执行次数（次）	统计某应用下全部流被执行次数之和，分为总次数、成功次数、失败次数。以“近1天（粒度10分钟）的总次数”为例，折线上每个值是指近10分钟内该应用被执行的总次数。
	应用执行耗时（ms）	统计某应用被执行所耗时间。以“近1天（粒度10分钟）的 P99.9 分位值”为例，折线上每个值是将近10分钟内该应用全部耗时数据正序排列后第99.9分位的值，即99.9%的耗时数据都不超过该值。其余分位值同理。
	应用消耗流量（kb）	统计某应用被执行所耗流量。以“近1天（粒度10分钟）的 P99.9 分位值”为例，折线上每个值是将近10分钟内该应用全部流量数据正序排列后第99.9分位的值，即99.9%的流量数据都不超过该值。其余分位值同理。
集成流	集成流监控概览	以流维度对数据进行聚合，可以查看选定应用的选定流的符合筛选条件的对应数据和图表信息。
	集成流执行次数（次）	统计某条流被执行的次数，分为总次数、成功次数、失败次数。以“近1天（粒度10分钟）的总次数”为例，折线上每个值是指近10分钟内该流被执行的总次数。
	集成流执行耗时（ms）	统计某条流被执行所耗时间。以“近1天（粒度10分钟）的 P99.9 分位值”为例，折线上每个值是将近10分钟内该流全部耗时数据正序排列后第99.9分位的值，即99.9%的耗时数据都不超过该值。其余分位值同理。
	集成流消耗流量	统计某条流被执行所耗流量。以“近1天（粒度10分钟）的 P99.9 分位值”为例，折线上每个值是将近10分钟内该流全部流量数据正序排列后第99.9分位的值，即99.9%的流量数据都不超过该值。其余分位值同理。

	( kb )	
触发器	连接器监控概览	以连接器维度对数据进行聚合，只会统计被执行的情况，可以查看选定应用的选定流的选定连接器的符合筛选条件的对应数据和图表信息。
	连接器执行次数 ( 次 )	统计某条流下各连接器被执行的次数，分为总次数、成功次数、失败次数。以“近1天（粒度10分钟）的总次数”为例，折线上每个值是指近10分钟内该连接器被执行的总次数。
	连接器执行耗时 ( ms )	统计某条流下各连接器被执行所耗时间。以“近1天（粒度10分钟）的 P99.9 分位值”为例，折线上每个值是将近10分钟内该连接器全部耗时数据正序排列后第99.9分位的值，即99.9%的耗时数据都不超过该值。其余分位值同理。
	连接器消耗流量 ( kb )	统计某条流下各连接器被执行所耗流量。以“近1天（粒度10分钟）的 P99.9 分位值”为例，折线上每个值是将近10分钟内该连接器全部流量数据正序排列后第99.9分位的值，即99.9%的流量数据都不超过该值。其余分位值同理。

# 运行日志

最近更新时间：2024-06-26 11:01:31

## 操作场景

节点日志集合了流程日志、节点日志和日志备份功能。您可以查看某个流程在一段时间内的实时及历史节点日志情况。目前腾讯轻联的节点日志以项目维度划分，查看日志前请先选择对应项目。

运行日志

默认项目

流程日志

节点日志

日志备份

近7天

收起筛选

运行状态

全部状态

流程服务名称

全部流程服务

流程名称

触发类型

全部类型

流程服务版本

全部版本

运行ID

## 操作步骤

### 流程日志

系统支持查看30天内流程运行详情，包括流程服务的运行状态、运行 ID、触发时间、结束时间和详情。

#### 说明：

目前控制台支持检索30天内流程日志。若有检索更早期日志的需求，请 [提交工单](#) 联系我们。

1. 登录 [腾讯轻联控制台](#)，进入 **运行日志 > 流程日志** 页面。
2. 在 **流程日志** 页面，默认展示最近流程服务的近30分钟的日志，日志按就近原则进行排序，日志列表包括：
  - **运行状态**：流程的运行状态。
  - **流程服务名称**：产生日志的流程服务。
  - **流程名称**：产生日志的流程。
  - **运行 ID**：产生日志时运行的标记，通过运行 ID 可将一次运行的日志全部找出来，便于分析。
  - **触发时间**：触发流程服务的时间。
  - **结束时间**：触发完毕的时间。
  - **操作**：查看日志详情。

运行日志 默认项目 帮助互动

[流程日志](#) [节点日志](#) [日志备份](#)

近7天 收起筛选

运行状态 全部状态 流程服务名称 全部流程服务 流程名称 请选择流

触发类型 全部类型 流程服务版本 全部版本 运行ID 请输入运行ID

发布环境 试用环境 搜索 重置

运行状态	流程服务名称	流程名称	运行ID	触发时间	结束时间	操作
------	--------	------	------	------	------	----

- 搜索日志说明：**支持按流程服务名称、版本、流程名称、运行环境、运行 ID、时间等维度来进行搜索。  
 单击时间下拉，将显示设置搜索时间的面板，支持近5分钟、近15分钟、近1小时、近6小时、近1天、近3天、近7天、近30天、今天、昨天、前天、本周、上周、自定义时间区间的时间设置。通过预定义的常用的时间段可快速此区间的日志。

近7天 收起筛选

最近时间	相对时间
近5分钟	近1天
近15分钟	近3天
近30分钟	<b>近7天</b>
近1小时	近30天
近6小时	本周
	上周

自定义时间

2023-05-05 20:59:15 - 2023-05-12 20:59:15

仅支持查看近30天的流程日志

确定 取消

## 节点日志

### 步骤1: 查询节点日志

节点日志支持查看历史日志和实时日志，可以通过日志信息回溯一次作业的运行情况和消息数据。

- 说明：**目前控制台支持检索近一个月的节点日志。若有检索更早期日志的需求，请 [提交工单](#) 联系我们。

1. 登录 [腾讯轻联控制台](#)，进入 **运行日志 > 节点日志** 页面。
2. 在 **节点日志** 页面，默认展示最近流程服务的近30分钟的节点日志，日志按就近原则进行排序，日志列表包括：
  - **运行状态**：不同状态采用不同颜色标记。
  - **流程服务名称**：产生日志的流程服务。
  - **流程名称**：产生日志的流程。
  - **运行 ID**：产生日志时运行的标记，通过运行 ID 可将一次运行的日志全部找出来，便于分析。
  - **组件/连接器**：产生日志的组件或连接器。
  - **运行时间**：产生日志的时间。
  - **日志详情**：日志详细内容。

The screenshot shows the '运行日志' (Run Logs) page with the following elements:

- Header: '运行日志' with a dropdown menu set to '默认项目' and a '帮助互动' (Help Interaction) link.
- Navigation: '流程日志', '节点日志' (selected), and '日志备份'.
- Filters: A '近7天' (Last 7 days) dropdown and a '收起筛选' (Collapse filters) button.
- Search Criteria:
  - 运行状态: 全部状态
  - 流程服务名称: 全部流程服务
  - 流程名称: 请选择流
  - 流程服务版本: 全部版本
  - 运行ID: 请输入运行ID
  - 记录内容: 请输入记录内容
  - 发布环境: 试用环境
- Buttons: '搜索' (Search) and '重置' (Reset).
- Summary: '日志数量 6条' (6 log entries) with '设置列' (Set columns) and '下载日志' (Download logs) options.
- Table: A table with columns: 运行状态, 流程服务名称, 流程名称, 运行ID, 连接器/组件, 运行时间, 操作.

- **搜索日志说明**：支持按流程服务名称、版本、流程名称、运行环境、运行状态、运行 ID、日志内容、时间等维度来进行搜索。  
单击时间下拉，将显示设置搜索时间的面板，支持近5分钟、近15分钟、近1小时、近6小时、近1天、近3天、近7天、近30天、今天、昨天、前天、本周、上周、自定义时间区间的时间设置。通过预定义的常用的时间段可快速此区间的日志。
- **支持自动刷新日志的能力**：单击自动刷新开关将显示刷新间隔时间。设置自动刷新时间后将根据间隔时间不断刷新日志。

The screenshot shows the '是否自动刷新' (Auto Refresh) dialog box with the following elements:

- Header: '运行日志' with a dropdown menu set to '默认项目'.
- Navigation: '流程日志', '节点日志' (selected), and '日志备份'.
- Filters: A '近7天' (Last 7 days) dropdown and a '收起筛选' (Collapse filters) button.
- Auto Refresh: A toggle switch is turned on.
- Refresh Interval: A dropdown menu showing '刷新间隔10秒' (Refresh interval 10 seconds), with a list of options: '刷新间隔10秒', '刷新间隔15秒', and '刷新间隔30秒'.
- Buttons: '确定' (Confirm) and '取消' (Cancel).

- 查看日志详情：单击某日志内容，右侧展示日志的详情。

运行日志 默认项目
帮助互动

流程日志
节点日志
日志备份

近7天
收起筛选

运行状态 全部状态

流程服务版本 全部版本

发布环境 试用环境

流程服务名称 全部流程服务

运行ID 请输入运行ID

日志数量 6条

运行状态	流程服务名称	流程名称	运行ID
成功	512_150823流程服	新流程	e61d9bc8-0d46-f1f2-bd0e-...
成功	512_150823流程服	新流程	e61d9bc8-0d46-f1f2-bd0e-...

Webhook
成功

流程服务名称 0512\_150823流程服务

流程名称 新流程

运行ID e61d9bc8-0d46-f1f2-bd0e-388b3fdf034f

运行时间 2023-05-12 15:12:37.909

日志内容

```
triggerBy:http.listener,messageVolume:460,responseCode:ok,cost:6215,runningStatus:true
```

### 步骤2：重试集成任务

在发布上线后，为了给集成项目保驾护航，腾讯轻联已完善流程监控指标，增加可靠性分析能力。当触发集成任务发布失败时允许自动重试或者用户手动重试，避免数据丢失。

- 自动重试：后台自动执行，若自动重试成功则更新控制台此条日志状态为成功，不记录之前失败的日志。
- 手动重试：若自动重试的最终结果是失败，则会更新控制台此条日志状态为失败，允许用户手动重试。

运行日志 默认项目
帮助互动 | 中扬

流程日志
节点日志
日志备份

全部版本
请输入运行ID
请输入记录内容

发布环境 试用环境

搜索
重置

日志数量 6条 设置列 下载日志

运行状态	流程服务名称	流程名称	运行ID	连接器/组件	运行时间	操作
成功	512_150823流程服	新流程	e61d9bc8-0d46-f1f2-bd0e-...		2023-05-12 15:12:37...	节点详情 <span style="border: 2px solid red; padding: 2px 5px;">重试</span>
成功	512_150823流程服	新流程	e61d9bc8-0d46-f1f2-bd0e-...		2023-05-12 15:12:37...	节点详情

### 步骤3：下载日志

节点日志提供日志下载功能。单击下载日志即可下载日志。

运行日志 默认项目

帮助互动 | 中扬

流程日志 **节点日志** 日志备份

近7天 收起筛选

运行状态 全部状态 流程服务名称 全部流程服务 流程名称 请选择流

流程服务版本 全部版本 运行ID 请输入运行ID 记录内容 请输入记录内容

发布环境 试用环境 搜索 重置

日志数量 6条 设置列 下载日志

运行状态	流程服务名称	流程名称	运行ID	连接器/组织	运行时间	操作
------	--------	------	------	--------	------	----

## 日志备份

为了满足用户对集成流程服务节点日志的后续分析和处理需求，我们支持将集成流程服务节点日志投递到 Kafka、ElasticSearch、CLS 平台上，从而灵活实现后续数据查看和处理。不同投递类型需要完成不同的配置内容，具体如下：

### 投递到 Kafka

← 投递任务信息

日志投递

任务名称 \*

投递类型 \* Kafka ElasticSearch CLS

源配置

投递对象 \*

集成应用	发布环境	操作
订单表有新数据时同步到企业微信群1207_143936	天津, 广州	<span>🗑</span>

+ 添加投递对象 (1/10)

目标配置

集群地址 \*   
多个域名/ip用逗号隔开, 例如 ip:port,ip:port

集群Kafka版本 \* 1.1

SASL安全认证模式 \* PlainText

使能TLS安全传输协议 \* False

KafkaTopic \* 请选择 刷新

压缩模式 不压缩

保存 取消

### 投递到 ElasticSearch

← 投递任务信息

**日志投递**

任务名称 \*

投递类型 \* Kafka ElasticSearch CLS

**源配置**

投递对象 \*

集成应用	发布环境	操作
订单表有新数据时同步到企业微信群1207_143936	天津, 广州	🗑

+ 添加投递对象 (1/10)

**目标配置**

集群地址 \*

TSL证书  选择文件

用户名 \*

密码 \*  🔍

索引名称 \*  刷新

保存 取消

**投递到 CLS**

← 投递任务信息

**日志投递**

任务名称 \*

投递类型 \* Kafka ElasticSearch CLS

**源配置**

投递对象 \*

集成应用	发布环境	操作
订单表有新数据时同步到企业微信群1207_143936	天津, 广州	🗑

+ 添加投递对象 (1/10)

**目标配置**

Cls地域 \*

Secret ID \*  🔍

Secret Key \*  🔍

日志集 \*  刷新

日志集主题 \*

保存 取消

# 告警管理

## 告警策略

最近更新时间：2024-08-01 15:33:02

### 操作场景

本文指导您在轻联告警模块中如何创建、修改、启停、复制和删除告警策略。

告警策略针对支持的已发布的应用设置执行失败次数等指标的阈值告警，在发生异常时及时通知您采取措施。告警策略包括基本信息、告警规则、通知模板三个必要组成部分。

### 基本概念

术语	定义
策略名称	策略名称用于命名告警策略。
告警类型	包括应用告警、集成流告警、API 服务告警、API 告警、环境告警。
描述	告警描述用于简单定义对应告警策略的作用，便于使用时进行区分。
告警级别	包括一般、重要、紧急。
告警对象	根据所选择的告警类型来选择不同的告警对象。
告警条件	是指标、判断逻辑、判断条件、统计周期和持续 N 个监控周期组成的一个有语义的条件。
通知模板	多个告警策略支持一键复用模板，适用于多种场景接收告警通知，详情请参见 <a href="#">新建告警通知模板</a> 。

### 使用限制

功能	限制
告警对象	最多可添加3个。
告警条件	最多可添加4个。

### 操作步骤

#### 新建告警策略

- 登录 [轻联控制台](#)。
- 在左侧导航栏，单击 **告警配置 > 告警策略**，进入告警策略配置页面。
- 单击**新建**，新建告警策略，此处重点对告警规则进行说明：

配置类型	配置项	说明
告警规则	告警对象	<ul style="list-style-type: none"><li>当选择不同告警类型，支持选择的告警对象不同。以集成流告警类型为例，告警对象需要先选择应用名称（单选），再选择集成流名称（可单选、多选、全选）。</li><li>选择发布环境（可多选），只有选中环境下的告警对象满足触发条件才会触发告警。</li></ul>
	告警条件	<ul style="list-style-type: none"><li><b>告警触发条件：</b>指标、判断逻辑、判断条件、统计周期和持续 N 个周期组成的一个有语义的条件。您可以根据图表中指标变化趋势设置告警阈值。例如，指标为应用触发次数/总次数、判断逻辑为&gt;、判断条件为10、统计周期为1分钟、持续周期为3个周期。表示：每1分钟收集一次应用触发次数/总次数的数据，若某个应用的触发次数/总次数连续3次大于10则触发告警。</li></ul>

		<ul style="list-style-type: none"> <li>● <b>告警频率</b>：您可以为您的每一条告警规则设置重复通知策略。即当告警产生时，您可以定义告警以特定的频率重复通知。可选：5分钟、15分钟、30分钟...12小时等告警频率。</li> </ul>
配置告警通知	通知模板	告警通知，用户创建自定义通知模板，每个告警策略可绑定一个通知模板，详情请参见 <a href="#">通知模板</a> 。

4. 配置完以上信息后单击**确定**，即成功创建告警策略。

← 新建告警策略

**基本信息**

策略名称 \*

告警类型 \*

应用告警  
应用层级告警

集成流告警  
集成流层级告警

API服务告警  
API服务层级告警

API告警  
API层级告警

环境告警  
独立环境告警

描述

告警等级 一般

**告警规则**

告警对象 ⊙ \*

项目	应用	集成流	发布环境	操作
<span style="border: 1px solid #ccc; padding: 2px 5px;">默认项目</span>	<span style="border: 1px solid #ccc; padding: 2px 5px;">订单表有新数据时</span>	<span style="border: 1px solid #ccc; padding: 2px 5px;">全选</span>	<span style="border: 1px solid #ccc; padding: 2px 5px;">全选</span>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>

+ 添加告警对象

告警条件 \*

指标	判断逻辑	判断条件	操作
<span style="border: 1px solid #ccc; padding: 2px 5px;">集成流执行次数 / 成功次数</span>	<span style="border: 1px solid #ccc; padding: 2px 5px;">&gt;=</span>	<span style="border: 1px solid #ccc; padding: 2px 5px;">1</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">次</span>	<span style="border: 1px solid #ccc; padding: 2px 5px;">🗑</span>

+ 添加告警条件

当以上 任意 告警条件, 按统计周期 1分钟 满足 持续三个周期 ● 每2小时告警一次

确定
取消

**说明：**  
指标解释，可参见 [指标名词](#) 说明。

### 修改告警策略

1. 进入 [轻联控制台](#) > [告警策略](#) 列表页。
2. 找到需要修改的告警策略，单击对应的策略名称。
3. 进入管理告警策略页，在对应的区域直接修改相关信息，单击**确定**即可保存。

### 告警策略启停

您能够通过告警启停功能对告警策略进行启用/停用操作，方便您在不需要使用某项告警策略时，及时停用告警策略，避免冗余消息对您的打扰，当您需要恢复告警时，也能一键快速启动告警策略。

1. 登录 [轻联控制台](#)。
2. 在左侧导航栏中，单击[告警配置](#)，再单击[告警策略](#)，进入管理页面。
3. 找到您需要操作的策略，在“告警启停”一列下，单击**运行状态**开关，即可启动或停止该策略的告警功能。

告警历史 **告警策略** 通知模板

新建 请输入策略名称关键词 🔍

策略名称	告警类型	告警等级	告警对象个数	告警频率	通知模板	运行状态	操作
测试	集成流告警	一般	1	每2小时告警一次	我是通知模板	<input checked="" type="checkbox"/>	告警历史 复制 删除
集成流告警策略1	集成流告警	一般	2	每2小时告警一次	我是通知模板	<input checked="" type="checkbox"/>	告警历史 复制 删除
我是告警策略1	应用告警	一般	1	每2小时告警一次	我是通知模板	<input type="checkbox"/>	告警历史 复制 删除

共 3 条 10条/页 跳至 1 页

## 复制告警策略

1. 进入 [轻联控制台](#) > [告警策略](#) 列表页。
2. 找到需要复制的告警策略，单击操作列的复制。
3. 在跳转页中修改所复制告警策略的相关信息，修改完后单击完成即可。

告警历史 **告警策略** 通知模板

新建 请输入策略名称关键词 🔍

策略名称	告警类型	告警等级	告警对象个数	告警频率	通知模板	运行状态	操作
测试	集成流告警	一般	1	每2小时告警一次	我是通知模板	<input checked="" type="checkbox"/>	告警历史 <b>复制</b> 删除
集成流告警策略1	集成流告警	一般	2	每2小时告警一次	我是通知模板	<input checked="" type="checkbox"/>	告警历史 复制 删除
我是告警策略1	应用告警	一般	1	每2小时告警一次	我是通知模板	<input type="checkbox"/>	告警历史 复制 删除

共 3 条 10条/页 跳至 1 页

## 删除告警策略

1. 进入 [轻联控制台](#) > [告警策略](#) 列表页。
2. 找到需要删除的告警策略，单击操作列的删除，在弹框中确认删除即可。

**说明：**  
只有已禁用的策略才能被删除。

告警历史 **告警策略** 通知模板

新建 请输入策略名称关键词 🔍

策略名称	告警类型	告警等级	告警对象个数	告警频率	通知模板	运行状态	操作
测试	集成流告警	一般	1	每2小时告警一次	我是通知模板	<input checked="" type="checkbox"/>	告警历史 复制 删除
集成流告警策略1	集成流告警	一般	2	每2小时告警一次	我是通知模板	<input checked="" type="checkbox"/>	告警历史 复制 删除
我是告警策略1	应用告警	一般	1	每2小时告警一次	我是通知模板	<input type="checkbox"/>	告警历史 复制 删除

共 3 条 10条/页 跳至 1 页

# 通知模板

## 通知模板

最近更新时间：2024-08-01 15:33:02

### 操作场景

本文指导您在轻联告警模块中通知模板相关内容。

支持多个策略一键复用模板，减少用户重复配置用户通知。可以自定义配置用户通知方式。例如：可以配置告警接收渠道为站内信、邮件、短信、微信、企业微信。并满足不同用户在不同通知时段生效。例如：A用户在白天接收告警通知；B用户在夜间接收告警通知。

通知模板支持新建、编辑、删除操作。

### 前提条件

- 查看通知模板：子账号需拥有腾讯轻联的项目读权限。
- 创建、编辑、删除通知模板：子账号需拥有腾讯轻联的项目写权限。

#### 说明：

详情请参见 [访问权限](#) 进行子账号授权。

### 使用限制

功能	限制
用户通知	最多可添加5项
接口回调	最多填写5个公网可访问的 URL

### 操作步骤

#### 新建通知模板

- 登录轻联控制台，进入告警配置 > [通知模板](#) 页面。
- 单击新建，在“新建通知模板”中填写基本信息。
  - 模板名称：自定义模板名称。
- 填写通知操作，参数说明如下：
  - 用户通知：

#### 说明：

- [系统管理员](#) 和 [项目管理员](#)：下拉支持选择当前项目下的所有成员账号。
- [项目成员](#)：只能选择该成员账号。
- [普通成员](#)：只能选择该成员账号。

参数	说明
接收对象	选择接收用户（可多选）。
通知时段	定义接收告警的时间段。
通知周期	可按照天维度自定义通知周期，默认是每天通知。
接收渠道	支持站内信、邮箱、短信、微信、企业微信五种告警渠道。您还可以根据不同的用户维度，设置不同的告警接收渠道

和通知时段，详情请参见 [告警接收渠道](#)。

○ 接口回调：

参数	说明
接口 URL	填写公网可访问到的 URL 作为回调接口地址，最多可填写5个告警回调地址。腾讯轻联将及时把告警信息推送到该地址（例如：企业微信等）。
通知时段	定义接收告警的时间段。

📌 说明：

- 回调地址保存成功后，当用户创建的告警策略被触发或被恢复均会通过接口回调推送告警消息。
- 当用户创建的告警策略被触发或恢复时，均会通过接口回调推送告警消息。接口回调也支持重复告警。
- 腾讯轻联告警配置的回调是通过 GET 方法调用回调地址，具体日志可通过您配置的服务地址的日志查看是否触发了回调。

4. 单击**确定**，即可新建一个通知模板。

← 新建通知模板

---

**基本信息**

模板名称\*  ✔

---

**通知操作**

用户通知 请确保所选接收对象已绑定手机，邮箱等通讯方式 [去查看](#)

接收对象 用户 13900000000 test 添加用户 删除

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  🕒 ⓘ

接收渠道 站内信 邮件 短信 微信 ⓘ 企业微信 ⓘ

+ 添加用户通知

---

接口回调 ⓘ

接口URL  删除

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  🕒 ⓘ

+ 添加接口回调

确定
取消

### 修改通知模板

1. 登录轻联控制台，进入告警配置 > [通知模板](#) 页面。

2. 找到需要修改的模板名称，单击名称进入编辑页面。
3. 进入编辑页面后，对需要修改的内容进行编辑，修改完后单击**确定**即可。

← 编辑通知模板

---

**基本信息**

模板名称

---

**通知操作**

用户通知 请确保所选接收对象已绑定手机，邮箱等通讯方式 [去查看](#)

接收对象 用户 test 添加用户 删除

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  🕒 ⓘ

接收渠道 站内信 邮件 短信 微信 ⓘ 企业微信 ⓘ

+ 添加用户通知

---

接口回调 ⓘ

接口URL  删除

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  🕒 ⓘ

+ 添加接口回调

确定
取消

## 删除通知模板

1. 登录轻联控制台，进入告警配置 > **通知模板** 页面。
2. 找到需要修改的模板名称，在其右侧的操作一栏中单击**删除**，在弹框中确认删除即可。

**说明：**  
被告警策略引用的模板无法删除，请先在告警策略中解除引用再进行删除。

告警历史 告警策略 **通知模板**

新建

模板名称	用户通知对象	回调地址	关联告警策略	操作
测试模板	1000011111111111test	-	告警测试	删除

共 1 条

10条/页 跳至 1 页

该通知模板已被引用在告警策略中，不允许删除

# 接口回调

最近更新时间：2024-08-01 15:33:02

## 操作场景

通过接口回调，您的企业微信群或自建系统可以直接收到腾讯轻联监控的告警通知。接口回调具备将告警信息通过 GET 请求推送到可访问公网 URL 的功能，您可基于接口回调推送的告警信息做进一步的处理。

## 注意事项

- 当前告警回调没有认证机制，不支持 HTTP 认证。
- 当用户创建的告警策略被触发或恢复时，均会通过接口回调推送告警消息。接口回调也支持重复告警。
- 腾讯轻联监控回调 API 出方向 IP 为动态随机分配，无法将具体的 IP 信息提供给您，建议您在安全组上配置加全放通策略。

## 操作步骤

1. 登录轻联控制台，进入告警配置 > [通知模板](#) 页面。
2. 单击新建，进入新建通知模板。
3. 在新建通知模板页配置完基础信息后，在接口回调模块中填写公网可访问到的 URL 作为回调接口地址（例如 `域名或 IP[:端口][/path]`），腾讯轻联将及时把告警信息推送到该地址。
4. 进入 [告警策略列表](#)，单击需要绑定告警回调的策略名称，进入管理告警策略页，并在告警策略页单击通知模板。
5. 告警消息将会通过 GET 请求推送到可访问公网 URL 地址。

← 编辑通知模板

基本信息

模板名称 \*

通知操作

用户通知 请确保所选接收对象已绑定手机，邮箱等通讯方式 [去查看](#)

接收对象   [添加用户](#) [删除](#)

通知周期  周一  周二  周三  周四  周五  周六  周日

通知时段  [🕒](#) [①](#)

接收渠道  站内信  邮件  短信  微信 [①](#)  企业微信 [①](#)

[+ 添加用户通知](#)

接口回调 [①](#)

接口URL  [删除](#)

通知周期  周一  周二  周三  周四  周五  周六  周日

通知时段  [🕒](#) [①](#)

[+ 添加接口回调](#)

[确定](#) [取消](#)

# 告警接收渠道

最近更新时间：2024-08-01 15:33:02

腾讯轻联告警配置目前提供五种告警渠道，分别为：**站内信、短信、邮件、微信、企业微信**。

默认开启全部告警渠道，但开启所有告警渠道需在 [访问管理控制台](#) 填写验证接收人联系方式，包括手机、邮箱和绑定微信。

## 使用微信接收告警通知

### 配置微信告警渠道

用户使用微信接收告警信息前需于腾讯云控制台发起绑定接收人微信操作。指引如下：

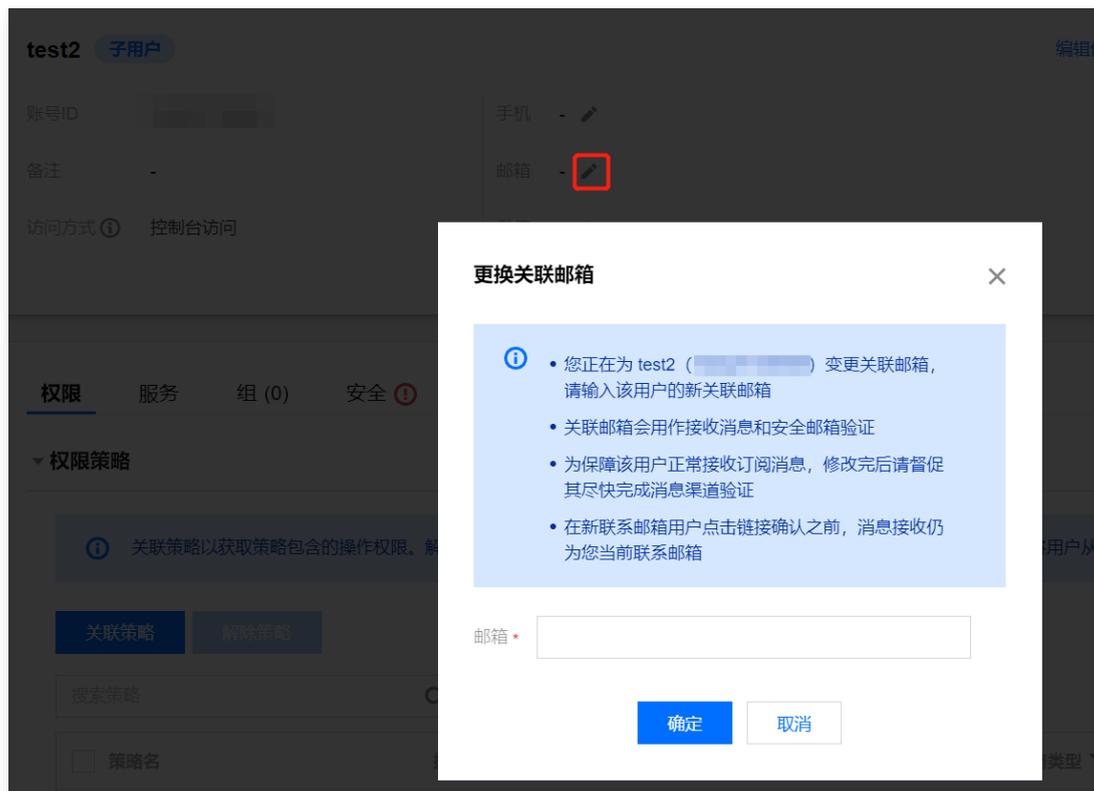
1. 登录 [访问管理控制台](#)。
2. 在左侧导航栏中，单击**用户** > **用户列表**，进入用户列表页面。
3. 找到需要配置接收微信通知的用户，单击用户名称进入用户详情页。
  - 3.1 填写并验证邮箱，用于接收微信验证消息（邮箱已验证的用户可忽略此步骤），请参见 [步骤1](#)。
  - 3.2 开启微信渠道并验证微信，请参见 [步骤2](#)。

### 步骤1：填写并验证邮箱

#### ⚠ 注意：

此步骤用于接收微信验证消息，邮箱已验证的用户可忽略此步骤。

1. 如下图单击邮箱右边的编辑按钮，填写您的邮箱并单击**确定**。

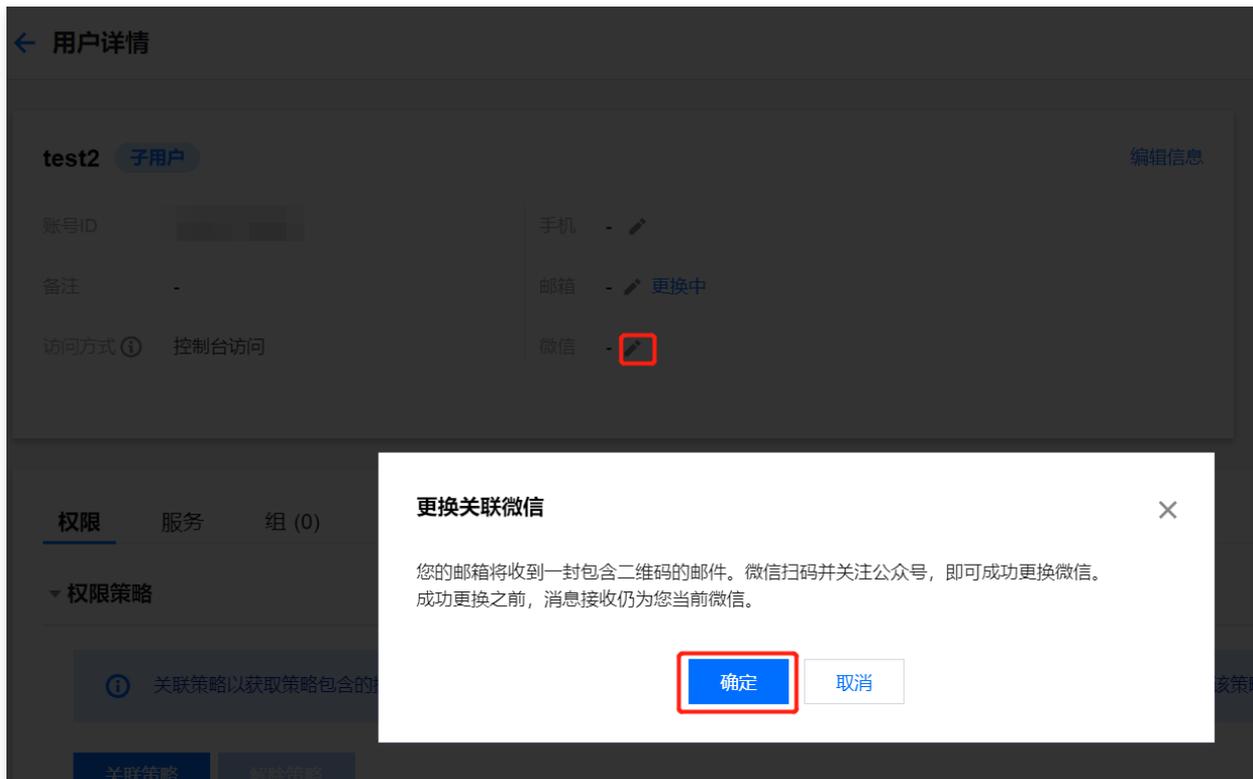


2. 登录您所填写的邮箱，在“**腾讯云邮箱接收消息验证**”邮件中，单击**确认接收**即可。



## 步骤2: 开启微信渠道并验证微信

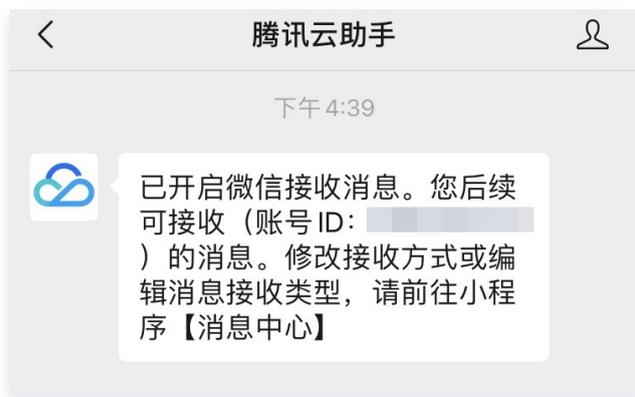
1. 返回用户详情页, 单击微信右侧的编辑按钮, 并在弹框中单击确定。



2. 在用户详情中单击 **更换中**, 使用接收告警信息的微信客户端进行扫码。按照提示关注**腾讯云助手**服务号, 即可完成绑定接收人微信。



3. 绑定成功如下图所示。



### 启用微信告警渠道

1. 进入 [腾讯轻联控制台](#) > [告警配置](#) > [通知模板](#)。
2. 单击**新建**，进入新建通知模板。
3. 在新建通知模板页配置完基础信息后，在告警接收渠道勾选**微信**。

### 基本信息

模板名称

### 通知操作

用户通知 请确保所选接收对象已绑定手机，邮箱等通讯方式 [去查看](#)

接收对象 用户  添加用户 删除

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  需开启微信通知才能接收公众号【腾讯云助手】的告警通知, [马上开启](#)

接收渠道 站内信 邮件 短信 微信 企业微信

[+ 添加用户通知](#)

接口回调 ① 删除

接口URL

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  🕒 ①

[+ 添加接口回调](#)

确定
取消

4. 进入 [告警策略列表](#)，单击需要绑定微信告警的策略名称，进入管理告警策略页，并在告警策略页绑定通知模板。



填写界面如下：

### 更换关联手机

- 您正在为 test\_alarm (XXXXXXXXXX) 变更关联手机，请输入该用户的新关联手机
- 关联手机会用作接收消息和安全手机验证
- 为保障该用户正常接收订阅消息，修改完后请督促其尽快完成消息渠道验证
- 在新联系手机用户点击链接确认之前，消息接收仍为您当前联系手机

手机 +86

## 启用短信告警渠道

1. 进入 [腾讯轻联控制台](#) > [告警配置](#) > [通知模板](#)。
2. 单击**新建**，进入新建通知模板。
3. 在新建通知模板页配置完基础信息后，在告警接收渠道勾选**短信**。

### 基本信息

模板名称 \*

### 通知操作

用户通知 请确保所选接收对象已绑定手机，邮箱等通讯方式 [去查看](#)

接收对象 用户  添加用户 删除

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  🕒 ①

接收渠道 站内信 邮件 短信 微信 ① 企业微信 ①

+ 添加用户通知

接口回调 ① 删除

接口URL

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  🕒 ①

+ 添加接口回调

确定
取消

4. 进入 [告警策略列表](#)，单击需要绑定短信告警的策略名称，进入管理告警策略页，并在告警策略页绑定通知模板。

### 基本信息

策略名称 \*

描述

### 告警规则

告警对象 \*

应用 ①	发布环境	操作
<input type="text" value="请选择应用"/>	<input type="text" value="请选择发布环境"/>	删除

+添加告警对象

告警条件 \*

指标	判断逻辑	判断条件	操作
<input type="text" value="应用执行次数 / 总次数"/>	<input "="" type="text" value="&gt;="/>	<input type="text" value="1"/>	次 删除

+添加告警条件

当以上  告警条件, 按统计周期  满足

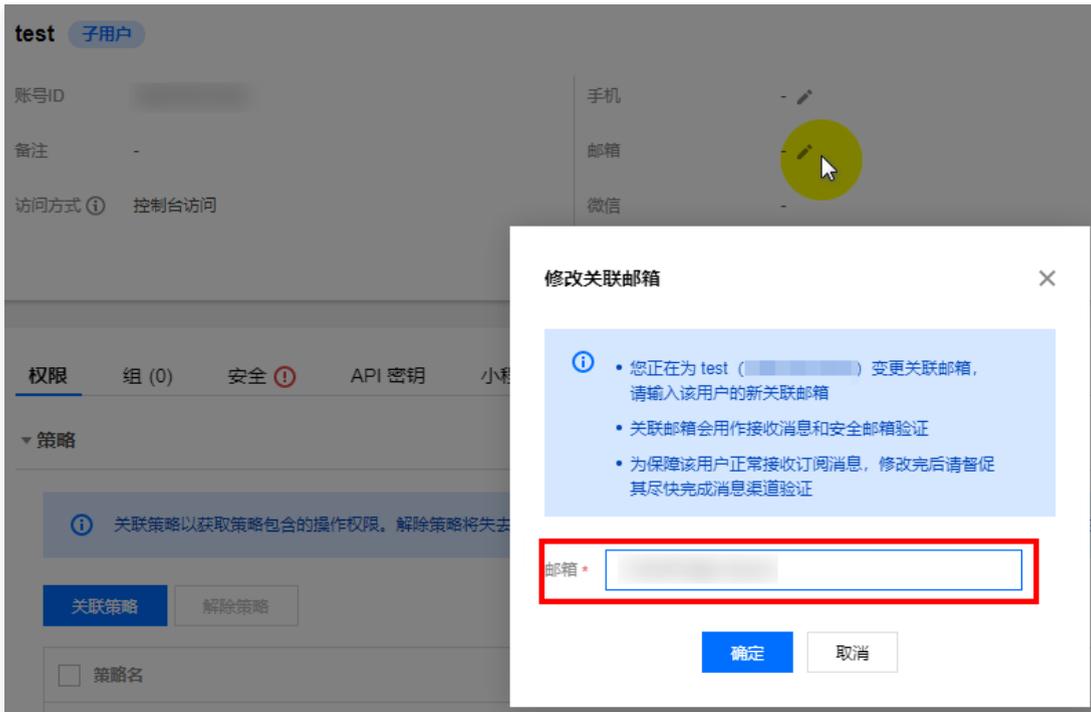
### 告警通知

通知模板 \*  +新建

## 使用邮箱接收告警通知

### 配置邮件告警渠道

1. 前往 [用户列表](#) 页面。
2. 找到需要配置接收邮箱通知的用户，单击用户名进入用户详情页。
3. 如下图单击邮箱右边的编辑按钮，填写您的邮箱并单击确定。



4. 在用户详情页邮箱右侧，单击**发送链接验证**。



5. 登录您所填写的邮箱，在“**腾讯云邮箱接收消息验证**”邮件中，单击**确认接收**即可。



## 启用邮件告警渠道

1. 进入 [腾讯轻联控制台](#) > [告警配置](#) > [通知模板](#)。
2. 单击新建，进入新建通知模板。
3. 在新建通知模板页配置完基本信息后，在告警接收渠道勾选**邮件**。

### 基本信息

模板名称

### 通知操作

用户通知 请确保所选接收对象已绑定手机，邮箱等通讯方式 [去查看](#)

接收对象   [添加用户](#) [删除](#)

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  [🕒](#) [①](#)

接收渠道 站内信 邮件 短信 微信 [①](#) 企业微信 [①](#)

[+ 添加用户通知](#)

### 接口回调 [①](#)

接口URL  [删除](#)

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  [🕒](#) [①](#)

[+ 添加接口回调](#)

4. 进入 [告警策略列表](#)，单击需要绑定短信告警的策略名称，进入管理告警策略页，并在告警策略页绑定通知模板。

### 基本信息

策略名称 \*

描述

### 告警规则

告警对象 \*

应用 ①	发布环境	操作
<input type="text" value="请选择应用"/>	<input type="text" value="请选择发布环境"/>	删除

+添加告警对象

告警条件 \*

指标	判断逻辑	判断条件	操作
<input type="text" value="应用执行次数 / 总次数"/>	<input "="" type="text" value="&gt;="/>	<input type="text" value="1"/>	次 删除

+添加告警条件

当以上  告警条件, 按统计周期  满足

### 告警通知

通知模板 \*  +新建

## 使用企业微信接收告警通知

### 配置企业微信告警渠道

#### 步骤1: 开通企业微信子用户管理

1. 进入 [访问管理](#)，单击左侧菜单栏的联合账号 > [企业微信](#)，单击立即开通。
2. 在企业微信授权页，企业微信管理员使用企业微信或微信扫码登录，选择可见范围用户后单击同意以上授权，并添加。
3. 授权成功，被选的可见用户在企业微信上会接收到成功安装腾讯云助手的消息。

#### ⚠ 注意:

非企业微信子用户或企业微信子用户但不在腾讯云助手可见范围内的用户无法通过企业微信接收消息。

#### 步骤2: 创建企业微信子用户

1. 单击左侧菜单栏用户 > 用户列表，单击新建用户。
2. 在新建用户页面，单击企业微信导入。
3. 选择企业名称，勾选用户，单击下一步，完成页面验证，企业微信子用户添加成功。

### 启用企业微信告警渠道

1. 进入 [腾讯轻联控制台](#) > [告警配置](#) > [通知模板](#)。
2. 单击新建，进入新建通知模板。

3. 在新建通知模板页配置完基本信息后，在告警接收渠道勾选企业微信。

### 基本信息

模板名称

### 通知操作

用户通知 请确保所选接收对象已绑定手机，邮箱等通讯方式 [去查看](#)

接收对象 用户  添加用户 删除

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  ⓘ ⓘ

接收渠道 站内信 邮件 短信 微信 ⓘ 企业微信 ⓘ

+ 添加用户通知

接口回调 ⓘ

接口URL  删除

通知周期 周一 周二 周三 周四 周五 周六 周日

通知时段  ⓘ ⓘ

+ 添加接口回调

确定
取消

4. 进入 [告警策略列表](#)，单击需要绑定短信告警的策略名称，进入管理告警策略页，并在告警策略页绑定通知模板。

基本信息

策略名称 \*

描述

告警规则

告警对象 \*

应用 ①	发布环境	操作
<input type="text" value="请选择应用"/>	<input type="text" value="请选择发布环境"/>	删除

+添加告警对象

告警条件 \*

指标	判断逻辑	判断条件	操作
<input type="text" value="应用执行次数 / 总次数"/>	<input "="" type="text" value="&gt;="/>	<input type="text" value="1"/>	次 删除

+添加告警条件

当以上  告警条件, 按统计周期  满足  !

告警通知

通知模板 \*  +新建

# 告警历史

最近更新时间：2024-08-01 15:33:02

## 操作场景

腾讯轻联告警配置提供查看告警历史功能，以便您回溯和查看近一个月的告警历史记录。

## 操作步骤

### 查看告警历史

- 登录腾讯轻联控制台，进入告警配置 > 告警历史 页面。
- (可选) 单击左上角的“时间筛选”，筛选需要查看告警历史的时间范围。  
支持近5分钟、近15分钟、近30分钟、近1小时、近6小时、近1天、昨天、近7天、近30天快速筛选，您也可以自定义时间范围。最多可查看近30天的告警历史。
- (可选) 还可以根据 告警状态、告警策略、告警类型、告警级别、告警对象筛选出符合条件的告警历史记录。

告警历史 告警策略 通知模板

近7天

告警状态 全部告警状态 告警策略 全部告警策略 告警类型 API告警 告警级别 全部 告警对象 全部 天津 搜索 重置

记录数量4条

触发时间	告警状态	告警类型	告警等级	告警策略	告警对象	告警内容	持续时长	结束时间	操作
2022-12-06 10:22:00	已失效	API告警	重要	-	现网自动化测试...	API请求次数@总次数 >=1, 最新统计值: 1	0小时1分钟	-	查看详情
2022-12-05 10:16:00	已失效	API告警	重要	-	现网自动化测试...	API请求次数@总次数 >=1, 最新统计值: 1	0小时1分钟	-	查看详情
2022-12-04 10:02:00	已失效	API告警	重要	-	现网自动化测试...	API请求次数@总次数 >=1, 最新统计值: 1	0小时2分钟	-	查看详情
2022-12-01 21:37:00	已恢复	API告警	重要	-	现网自动化测试...	API请求次数@总次数 >=1, 最新统计值: 0	0小时2分钟	2022-12-01 21:39:00	查看详情

共 4 条

10条/页 跳至 1 页

## 重置筛选条件

您成功筛选告警历史信息后，在列表中单击重置即可恢复到默认展示页面。

## 告警状态

告警状态	说明
持续中	被触发且还没有恢复的告警。
已恢复	已经恢复正常状态的告警。
已失效	当告警策略被删除或被禁用时，对应告警历史失效。

# 管理工具

## 授权配置

最近更新时间：2024-11-07 11:41:52

### 操作场景

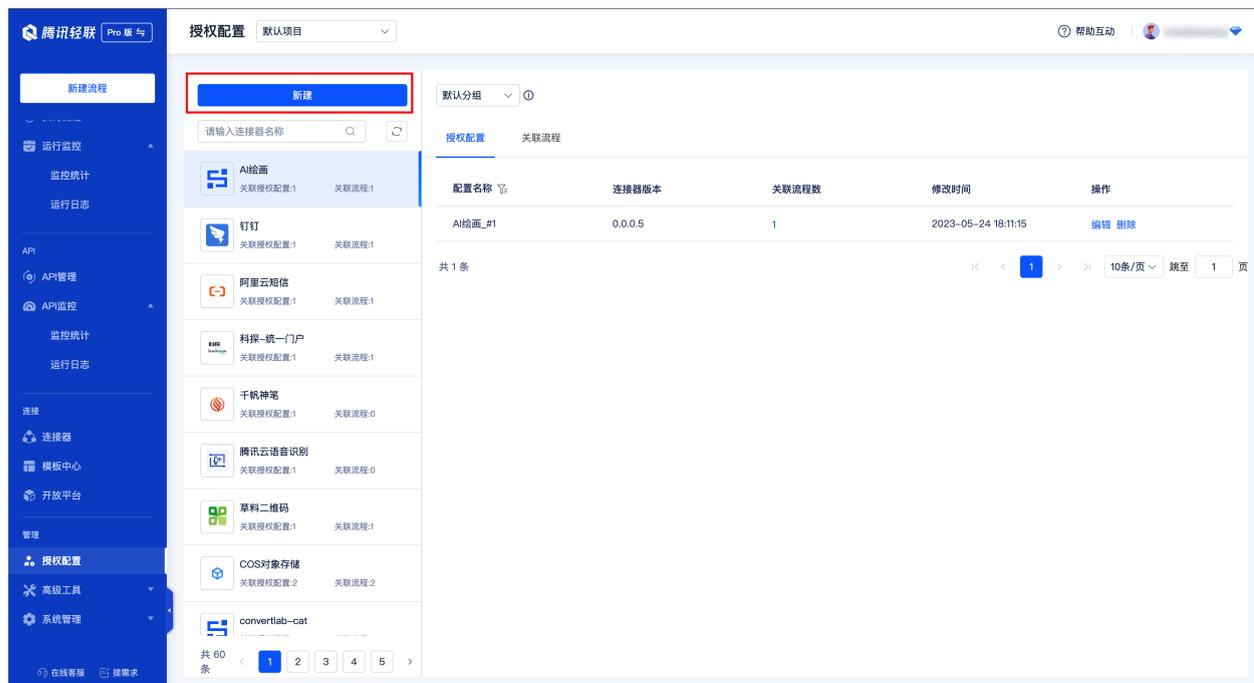
授权配置集中展示当前账号下所有已授权应用账号的配置信息，同一应用可以添加多个授权配置，同一项目下的流程可以选择对应配置使用，也包含快速添加和修改操作，已使用的应用也会同步更新配置内容。

### 操作步骤

#### 添加授权配置

##### 步骤1：基本信息

登录 [轻联控制台](#)，选择**授权配置**，选择对应项目名称后单击**新建**进入授权配置添加界面。



##### 步骤2：授权和配置

按照页面提示填写完成后单击**下一步**，进入授权或配置界面，不同的连接器对应第三方应用授权所需信息不同，请跟进第三方获取密钥或账密文档后再填写。



### 修改配置

针对已存在的连接配置提供修改功能，但需要注意如下两点：

**注意：**

- 修改连接器配置，会影响到关联的集成应用。
- 若要修改连接配置生效，则需要对关联的集成应用任务进行重启，否则会导致运行中的集成应用任务运行异常，且重启会导致运行中集成应用任务的瞬时中断。



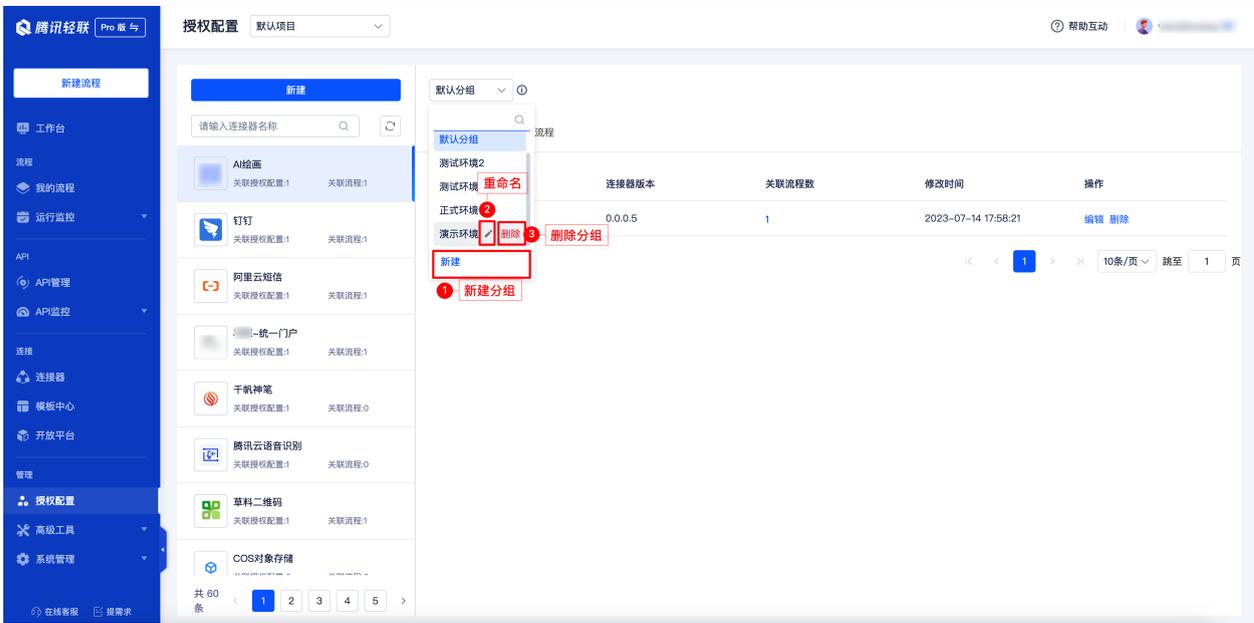
## 授权配置分组

连接管理有分组概念，适用于不同环境不同账号的使用场景，切换分组可以快速切换到不同的配置进行授权。例如：正式环境使用分组一的配置，测试环境使用分组二的配置。

分组：同一个连接配置可在不同的分组中存在，不同分组的连接配置可以不同，便于快速切换，默认分组不可删除。

### 步骤1：添加分组

可新建多个分组，每个分组对同个连接配置，配置不同的授权账号，对应不同的业务环境。



### 步骤2：切换分组

在发布应用时，可选择不同的连接配置分组发布，方便用户一键切换对应环境所需要的连接配置。

← 流程名字修改

☰ 新流程 ×

触发器

1 腾讯问卷 - 当有用户提交问卷时

执行流程

2 腾讯文档 - 新建一行表格记录

+ 当上个节点执行之后，自动结束流程

☰ 检查 调试 版本 帮助 更多 上线流程

授权配置分组

配置分组名称	操作
默认分组 <sup>1</sup>	当前使用的分组
测试环境2	切换 <sup>2</sup>
测试环境	切换
正式环境	切换
演示环境	切换

点击「切换」，使用此分组

# 本地代理

最近更新时间：2025-03-12 16:45:13

## 简介

本地代理是为实现腾讯轻联与用户内网服务集成互通而创新设计的代理系统，主要应用场景为用户内网服务不提供公网访问，而又希望通过腾讯轻联（公有云部署）来完成与内网服务的集成。

本地代理由 Server 和 Agent 构成：

- Server 部署在腾讯轻联系统内网，用户无需关注。
- Agent 部署在用户内网，允许根据地域或服务划分部署多个 Agent，通过本地代理转发来实现腾讯轻联与用户内网服务的数据交互。

## 新建并启动本地代理

### 步骤1：新建本地代理（Agent）并下载

1. 登录 [腾讯轻联控制台](#)，选择本地代理。
2. 单击新建，当公钥配置选择“手动配置”时，请参见下述 [生成公私钥](#) 的步骤生成公钥并上传（如果公钥配置选择“系统生成”则跳过）。
3. 配置 Agent IP 访问控制白名单（可选），并完成内网服务配置，单击下一步并确认。

← 新建本地代理 🗨 购买咨询 📄 帮助文档 👤 用户头像

① 基础配置 ———— ② 下载及启动Agent ———— ③ Agent连接测试

本地代理名称

限制访问IP  启用  不启用

公钥配置  系统生成  手动配置

内网服务配置

服务名称	服务IP地址/服务域名	服务端口
<input type="text" value="请输入"/>	<input type="text" value="请输入单个IPv4地址或域名，如192.168.1.0或www.test.com"/>	<input type="text" value="请输入0-65535"/>

[+ 添加内网服务](#)

4. 在列表页或新建步骤的第二步，单击**下载 Agent**完成 Agent 下载。

Agent 解压后目录结构如下：

- bin 目录中包含 Agent 的可执行程序，按照不同的操作系统放在子目录 Linux、Windows、Mac 下。
- configs 目录中包含 Agent 运行过程中所必须的配置，configs 目录介绍：
  - client 中存放 Agent tls 通信所必须的密钥等配置，与 Server 对应，此目录下文件不可删除，不可修改。
  - secret 目录下存放 Agent 连接 Server 时的私钥，私钥如何生成请参见 [生成公私钥](#)。
  - config.yaml 文件中包含 Agent 运行过程中必须依赖的配置。
  - logger\_config.yaml 文件中包含 Agent 运行过程中的日志配置，可修改日志级别及日志备份策略。

- log 中存放 Agent 运行过程中产生的日志。
- scripts 目录存放 Agent 启动/停止脚本 ( start.sh/stop.sh )。

← 新建本地代理

购买咨询 帮助文档

基础配置 2 下载及启动Agent 3 Agent连接测试

至此您已新建本地代理，该页面为引导页，您也可以从网关自行下载并启动Agent

第一步: 下载Agent

↓ 下载Agent

解压后得到名为ipaas-private-cloud-agent的文件夹，后续步骤我们将该文件夹对应完整路径记为<INSTALL\_HOME>，文件内容释义等更多信息请参见 [帮助文档](#)

第二步: Agent日志配置(可选)

针对<INSTALL\_HOME>/configs目录下的 logger\_config.yaml 文件，可以按需修改网关日志级别和日志备份策略等

第三步: 启动Agent

打开终端 (Windows需用管理员身份打开)，并通过cd命令进入Agent文件夹所在路径，根据不同操作系统执行下方对应命令，当看到“tunnel create successful”等字样则表示已成功启动Agent。

【Mac系统】sudo sh ./ipaas-private-cloud-agent/scripts/mac/start.sh

【Windows系统】./ipaas-private-cloud-agent/scripts/windows/start.bat (若终端使用CMD需将路径中的/替换成\)

【Linux系统】sudo sh ./ipaas-private-cloud-agent/scripts/linux/start.sh

常见问题:

执行启动命令后显示“permission denied”?

没有文件执行权限。需执行chmod 777 <INSTALL\_HOME>/scripts/mac/start。

执行启动命令后显示“authHandShake failed”?

需要用下载的private.pem私钥，去替换<INSTALL\_HOME>/configs/secret中的私钥。

更多问题，详见 [帮助文档](#)

下一步 取消

本地代理 默认项目

购买咨询 帮助文档

新建

请输入本地代理名称

名称	健康状态	内网服务数	创建时间	操作
代理啊	待连接	1	2025-03-12 15:35:08	禁用 下载Agent 删除

## 步骤2: Agent 日志配置 (可选)

修改 Agent 文件夹 `ipaas-private-cloud-agent/configs` 目录下的 `logger_config.yaml` 文件，可以按需修改网关日志级别和日志备份策略等，`logger_config.yaml` 中各参数含义在文件中已做详细说明，本文不再介绍。

## 步骤3: 启动 Agent

打开终端 (Windows 系统需用管理员身份打开)，通过 `cd` 命令进入 Agent 文件夹 `ipaas-private-cloud-agent` 所在路径。并根据不同操作系统执行下方对应命令。

最后终端会提示 `Do you want to run as a background daemon [Y/N]?` 如果填“N”，当在终端和 `agent.zlog` 日志文件中都看到“`tunnel create successful`”字样时，表示 Agent 已成功启用；如果填“Y”，当在 `agent.zlog` 日志文件中看到“`tunnel create successful`”字样时，表示 Agent 已成功启用。( `agent.zlog` 文件在 `./ipaas-private-cloud-agent/log` 目录下)。

### Mac 系统

执行以下命令启用 Agent:

```
sudo sh ./ipaas-private-cloud-agent/scripts/mac/start.sh
```

### Linux 系统

执行以下命令启用 Agent:

```
sudo sh ./ipaas-private-cloud-agent/scripts/linux/start.sh
```

### Windows 系统

用管理员身份打开终端，并执行以下命令启用 Agent:

```
.\ipaas-private-cloud-agent\scripts\windows\start.bat
```

## 禁用 Agent 命令

各操作系统执行禁用 Agent 命令如下:

### Mac系统

执行以下命令可禁用 Agent:

```
sudo sh ./ipaas-private-cloud-agent/scripts/mac/stop.sh
```

### Linux系统

执行以下命令可禁用 Agent:

```
sudo sh ./ipaas-private-cloud-agent/scripts/linux/stop.sh
```

### Windows系统

用管理员身份打开终端，并执行以下命令可禁用 Agent:

```
.\ipaas-private-cloud-agent\scripts\windows\stop.bat
```

## 生成公私钥

### ⚠ 注意:

新建本地代理时，当公钥配置选择手动配置时，才需要手动完成下列配置步骤。如果选择系统生成则忽略下列配置。

### 步骤1: 检查 OpenSSL 版本

执行以下命令，检查系统是否已安装 OpenSSL：

```
openssl version
```

执行命令如果能正常输出 OpenSSL 版本信息，说明系统已安装 OpenSSL，则可跳过下文步骤2，否则请参考下文步骤安装 OpenSSL。

### 步骤2: 安装 OpenSSL

MAC、Linux、Windows 系统安装 OpenSSL 方式不同，详细安装步骤如下：

#### Mac 系统

执行以下命令安装 OpenSSL：

```
brew install openssl
```

#### Linux 系统

执行以下命令安装 OpenSSL：

##### Centos

```
yum install openssl
```

##### Ubuntu

```
sudo apt-get install openssl  
sudo apt-get install libssl-dev
```

#### Windows 系统

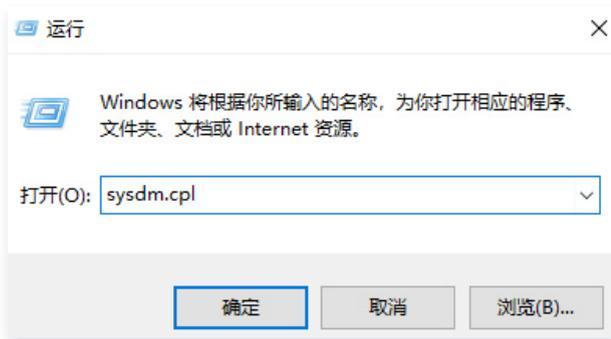
1. 下载 [OpenSSL](#) 安装包，选择不同的安装包下载。例如64位系统，则选择如图所示的安装包下载（选择 Light 的 EXE 版本即可）。

Download Win32/Win64 OpenSSL		
Download Win32/Win64 OpenSSL today using the links below!		
File	Type	Description
Win64 OpenSSL v1.1.1i Light <a href="#">EXE</a>   <a href="#">MSI</a>	3MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v1.1.1i (Recommended for users by the creators of <a href="#">OpenSSL</a> ). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v1.1.1i <a href="#">EXE</a>   <a href="#">MSI</a>	63MB Installer	Installs Win64 OpenSSL v1.1.1i (Recommended for software developers by the creators of <a href="#">OpenSSL</a> ). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.1.1i Light <a href="#">EXE</a>   <a href="#">MSI</a>	3MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v1.1.1i (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.1.1i <a href="#">EXE</a>   <a href="#">MSI</a>	54MB Installer	Installs Win32 OpenSSL v1.1.1i (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.

2. 双击安装包，安装 OpenSSL。注意记录 OpenSSL 的安装目录（例如安装目录为 `C:\Program Files\OpenSSL-Win64`，后续 OpenSSL 配置环境变量时需要用到）。

3. 配置环境变量，以 Win10 系统为例：

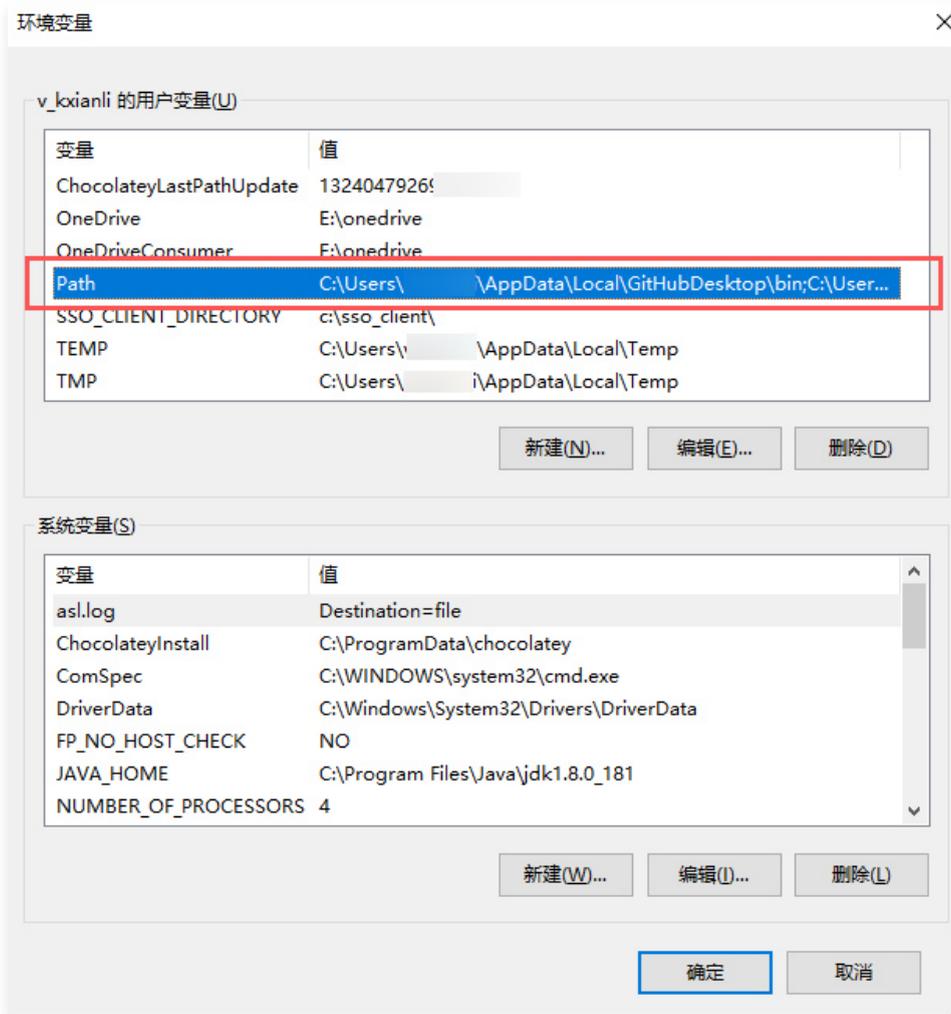
3.1 按 Win+R 键，在弹出的“运行”窗口中输入 `sysdm.cpl`，并按 Enter 键，打开“系统属性”窗口。



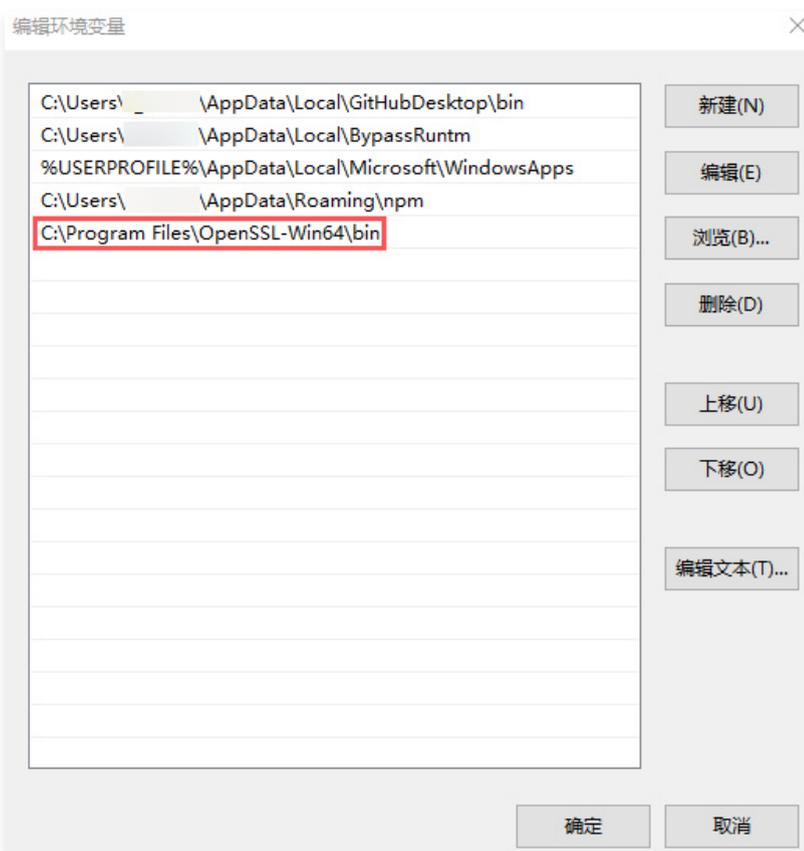
3.2 选择高级 > 环境变量进入环境变量页面。



### 3.3 双击 Path 进入编辑环境变量页面。



### 3.4 单击新建，在左侧变量中输入 OpenSSL 安装目录的 bin 目录（bin 目录 = 安装目录 + bin，例如安装目录为 C:\Program Files\OpenSSL-Win64，则 bin 目录为 C:\Program Files\OpenSSL-Win64\bin）。



3.5 单击**确定**完成环境变量配置。

#### 4. 安装验证

4.1 按 **Win+R** 键，在弹出的“运行”窗口中输入 `cmd`，并按 **Enter** 键，打开“系统属性”窗口。

4.2 执行 `openssl version` 命令，如果出现 OpenSSL 版本信息，则说明 OpenSSL 安装成功，否则仔细核对 OpenSSL 安装步骤。

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.18362.1139]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\          >.openssl version
OpenSSL 1.1.1i  8 Dec 2020

C:\Users\v_kxianli>
```

### 步骤3：生成及更新公私钥

1. 根据不同的 OpenSSL 版本，执行不同命令生成私钥：

如果 OpenSSL 版本为 v1，则执行：

```
openssl genrsa -out private.pem 1024
```

如果 OpenSSL 版本为 v3，则执行：

```
openssl genrsa -traditional -out private.pem 1024
```

#### ⚠ 注意：

请将生成的私钥放置在 `ipaas-private-cloud-agent/configs/secret` 目录下。

2. 执行以下命令，以上述私钥为基础生成公钥。当前目录下生成的 `public.pem` 即为公钥。

```
openssl rsa -in private.pem -RSAPublicKey_out -out public.pem
```

3. 生成新的私钥时，需替换 `ipaas-private-cloud-agent/configs/secret` 目录下的 `private.pem` 文件。

## 常见问题

### 执行 Agent 启动命令后显示 “permission denied” ？

没有文件执行权限。如果是 Mac 系统，需先执行 `chmod 777 INSTALL_HOME/scripts/mac/start.sh`（其中 `INSTALL_HOME` 代表 Agent 文件夹所在路径）。

### 执行 Agent 启动命令后显示 “authHandShake failed” ？

需要用下载的 `private.pem` 私钥，去替换 `INSTALL_HOME/configs/secret` 中的私钥（其中 `INSTALL_HOME` 代表 Agent 文件夹所在路径）。

### Mac 系统执行 Agent 启动命令后，提示无法打开 Agent 文件夹，因为无法验证开发者？

需执行 `sudo spctl --master-disable` 进行验证。

### 执行 Agent 启动命令后显示 “ip:xxx not in whitelist” ？

需要在新建安全网关时，将网络出口 IP 白名单配置进去。

# 数据存储

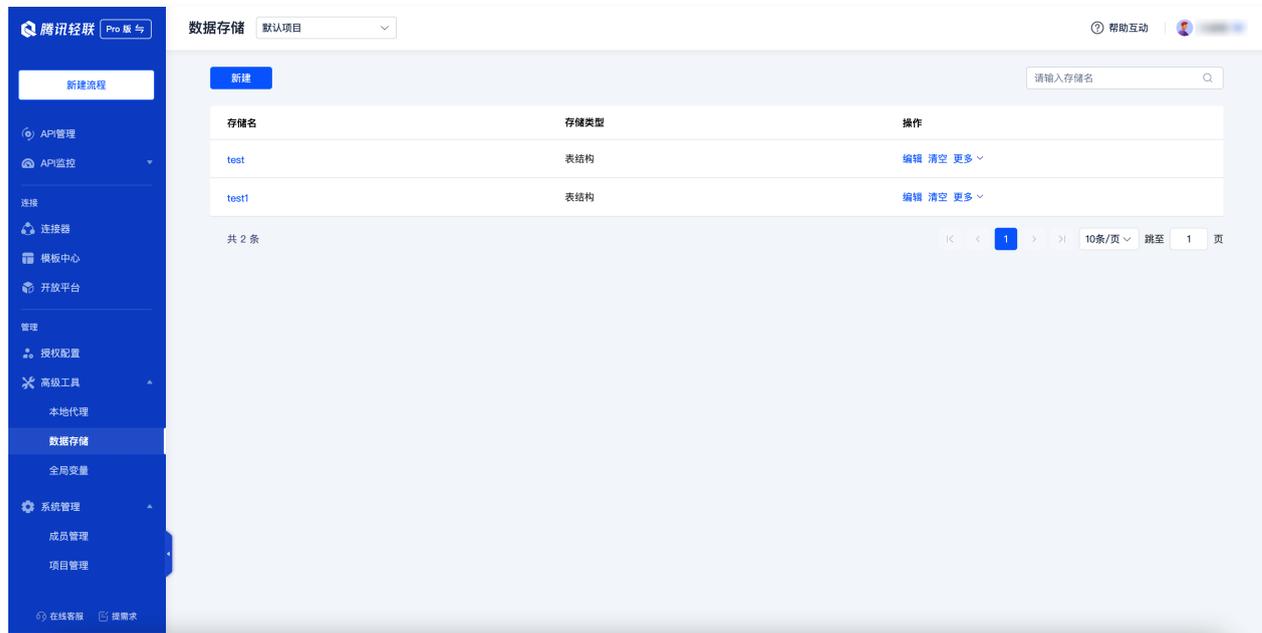
最近更新时间：2024-11-07 11:41:52

## 使用场景

腾讯轻联提供丰富的状态管理方案。登录 [腾讯轻联控制台](#)，单击左侧工具栏高级工具 > 数据存储，进入数据存储功能，即可管理项目中使用到的存储结构和数据。

## 数据存储管理

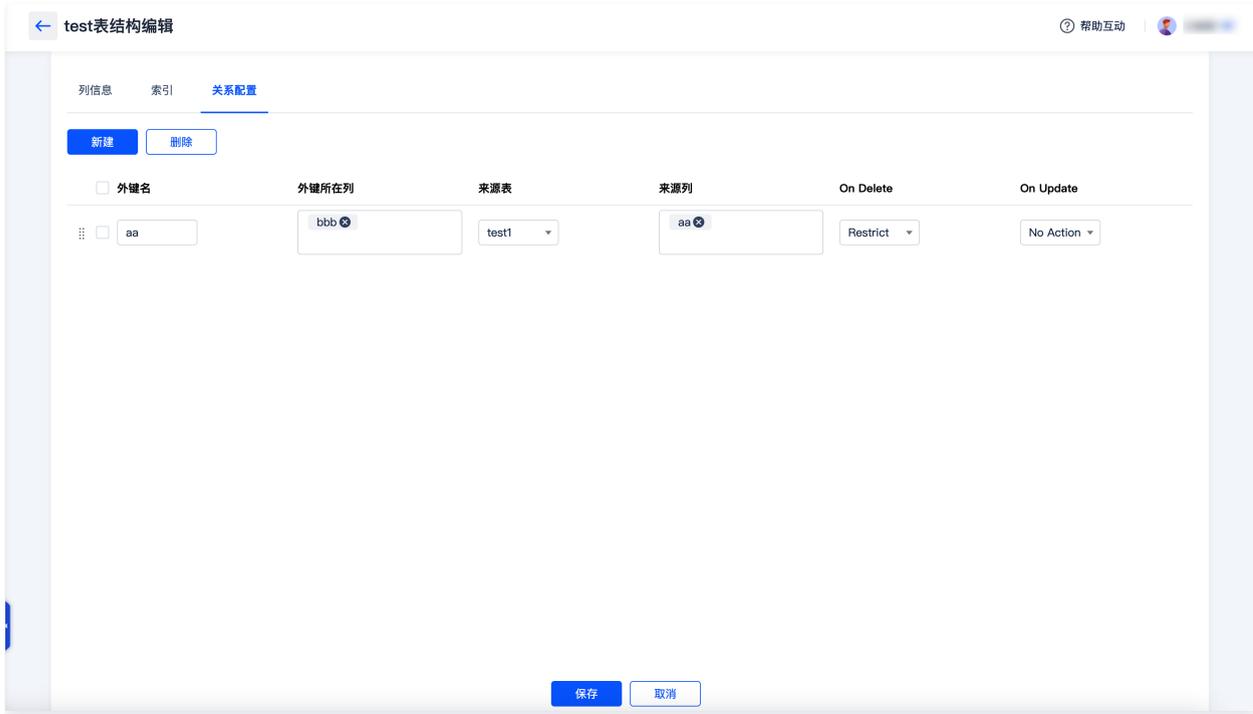
数据存储主页面为存储管理页面。该页面下可以新建存储，同时查看当前已创建的所有存储列表，包括存储名、存储类型以及针对不同类型的存储可以执行的操作。



## 新增存储

新增存储功能可以帮助您快速创建一个全新的存储类型。只需进行简单的配置即可完成创建工作、需要创建的存储名称以及对应的存储类型。

- 存储名称：请输入25位以内的英文字母或\_。
- 存储类型：表结构、哈希结构、列表和字符串四种类型。

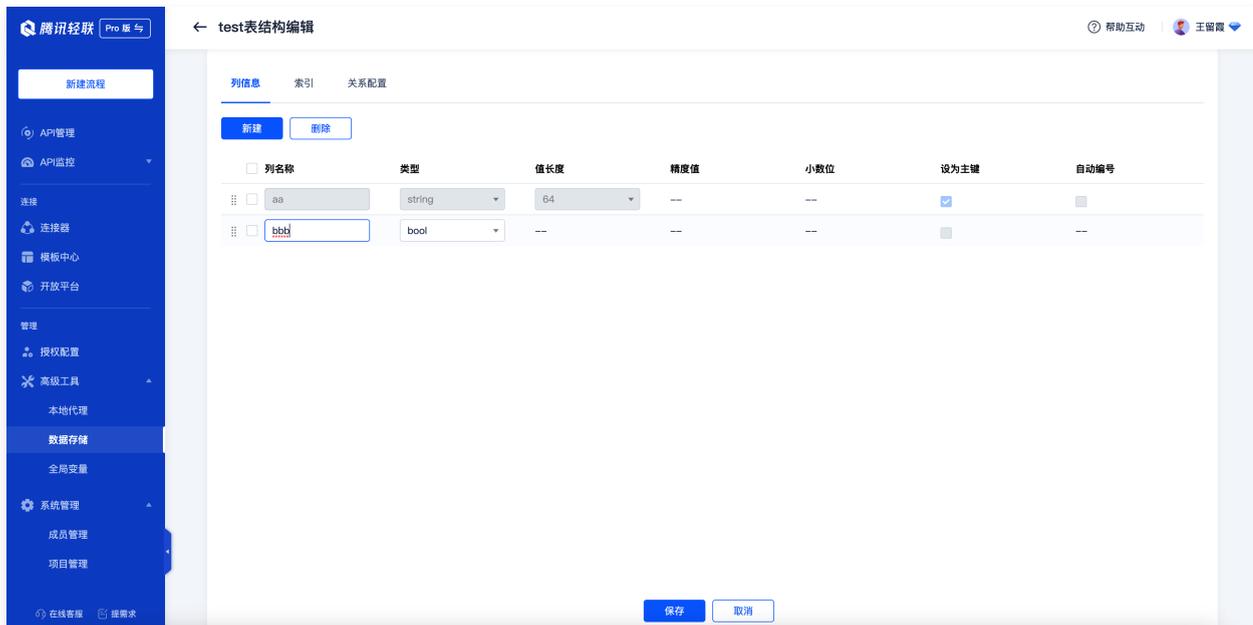


## 表结构

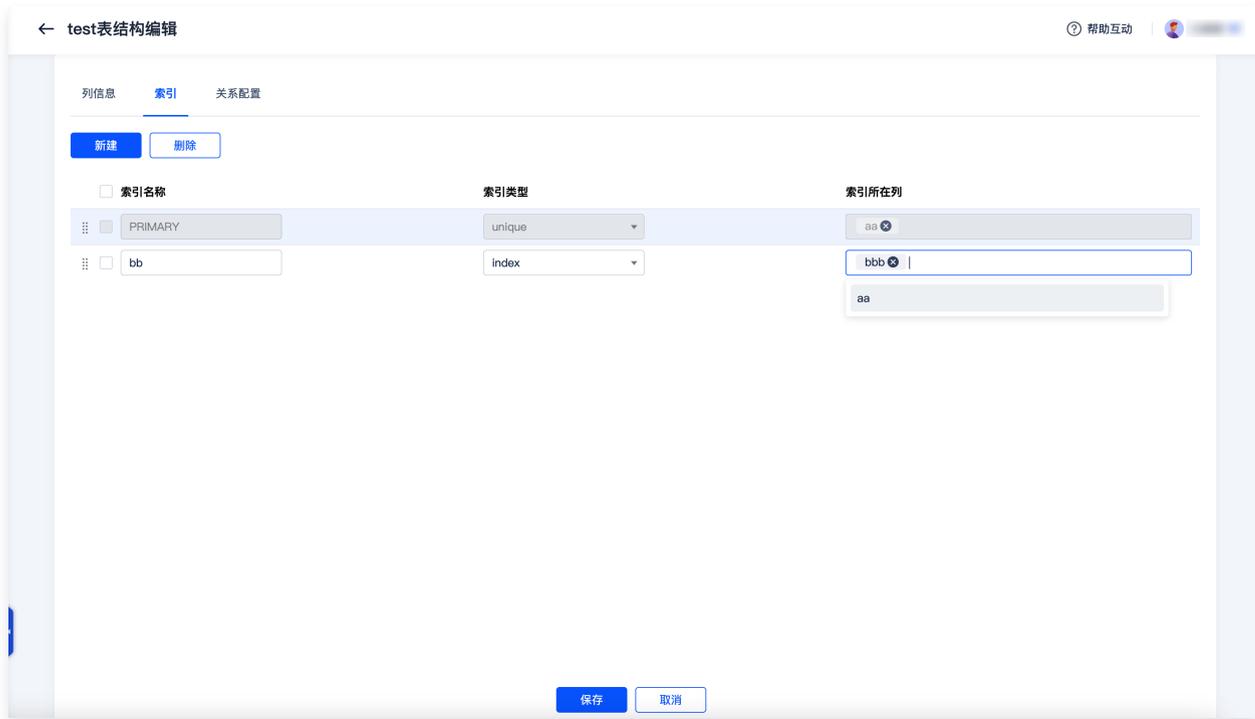
### 编辑结构

用户可以使用该功能进行表结构的维护，包括列信息、索引信息、关系配置。

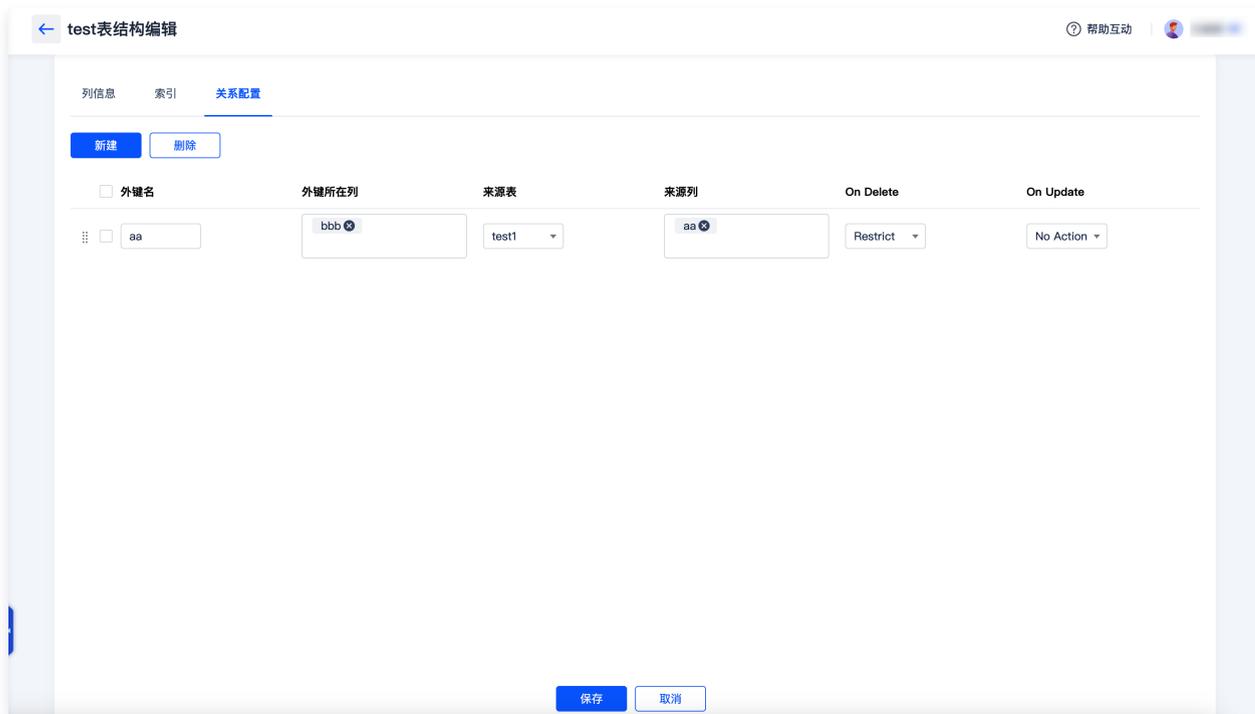
- **列信息：**表结构的列信息配置支持新增、修改和删除数据表中的列配置，可以快速完成结构编排。当前支持用户编辑列名称、列中数据的类型（支持 String、bool、int、float、decimal、datetime、date 和 time）、值长度（当类型为 string 时）、精度值和小数位（当类型为 decimal 时）、是否为主键以及是否支持自动编号（仅当该列为主键时）。



- **索引配置：**索引配置用法同 MySQL 的 index 配置，用户可以通过此标签页建立索引，提高 MySQL 的检索速度。当前支持新建、删除索引、配置索引名称、配置索引类型（当前支持 index 和 unique 两种类型，具体区别请参见 [DOCUMENTATION](#)）、配置索引所在列（在已维护的列信息列表中选择）。



- **关系配置**：腾讯轻联数据存储同时还支持用户配置外键信息。通过给外键命名、配置外键所在列、选择外键所对应的来源表和来源列信息即可完成配置。如果需要进行外键事件触发配置，则可以通过配置 on delete 和 on update 选项，请参见 [DOCUMENTATION](#)。



### 更多操作

- **清空表结构**：允许用户清除当前表结构中的所有数据内容。

**注意：**

清空动作是高危操作，该操作将会清空本存储内的所有数据，且无法恢复，请谨慎使用。

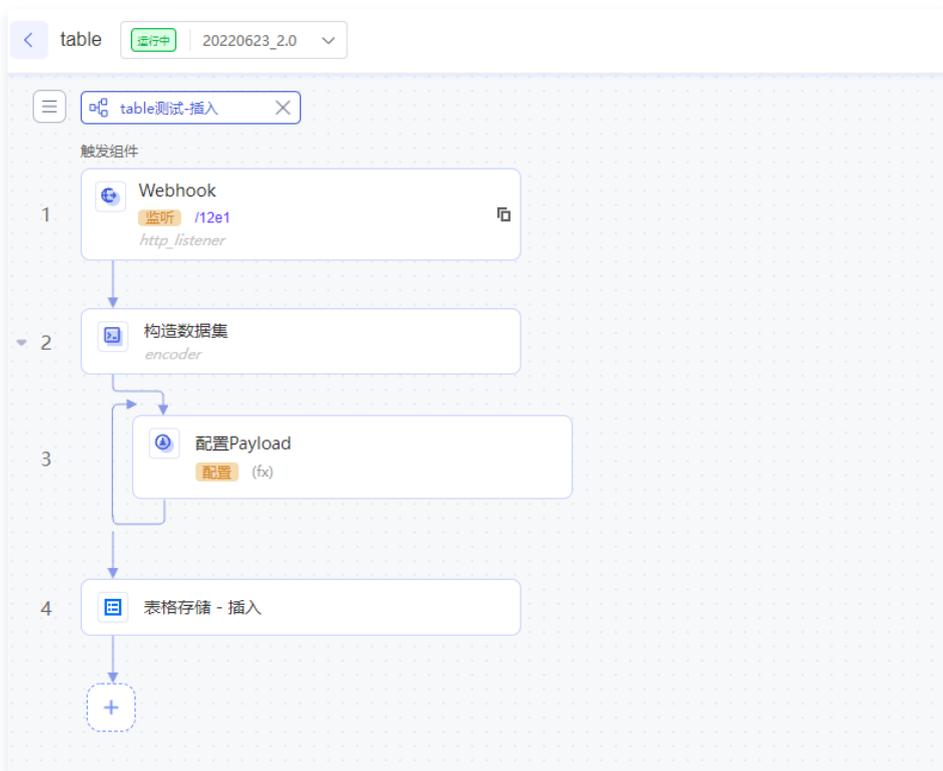


- **复制同结构表：**通过复制同结构表功能可以基于现有的数据表结构快速创建一个新的数据表，方便用户进行数据结构的迁移和备份。

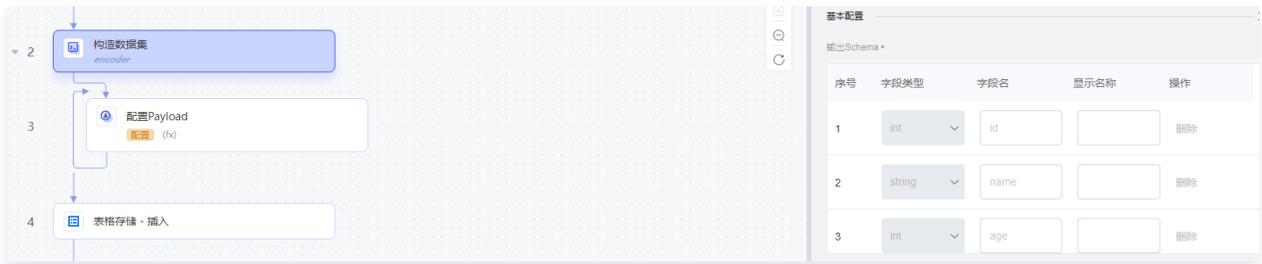


- **删除：**可以删除本条存储记录。同样的，此操作被视为高危操作，会对线上数据造成影响，请确认后再进行。
- **使用示例：**通过集成应用向 table 插入数据。

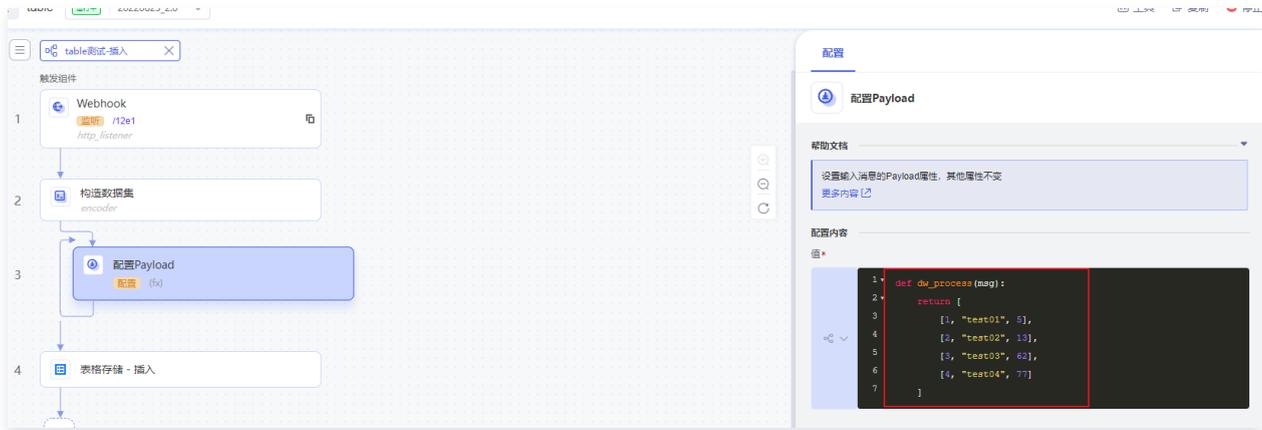
1.1 创建应用，并在流中加入 table 组件节点。



1.2 此处构建数据集如下：包含 name、age、id 三个字段。



模拟数据输入如下：



1.3 配置 table 组件，填写想要添加的表名和表结构信息。



1.4 新建表结构。



1.5 完善集成应用的其他配置，发布并运行此应用以写入数据。

1.6 打开数据存储功能，观察到数据表结构已经成功生成。

列信息    索引    关系配置

新建    删除

<input type="checkbox"/>	列名称	类型	值长度	精度值	小数位	设为主键	自动编号
<input type="checkbox"/>	name	string	64		--	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	age	int			--	<input type="checkbox"/>	--
<input type="checkbox"/>	id	int			--	<input type="checkbox"/>	--

1.7 单击此存储名，可以查看通过集成应用写入的数据内容。

← tables    广州

基本信息

新建    删除    导出    提交    快捷操作

<input type="checkbox"/>	name	age	id
<input type="checkbox"/>	test01	5	1
<input type="checkbox"/>	test02	13	2
<input type="checkbox"/>	test03	62	3
<input type="checkbox"/>	test04	77	4

共 4 条    10条/页    跳至 1 页

### 哈希结构

哈希（hash）是根据键码值（Key value）而直接进行访问的数据结构。它通过把键码值映射到表中一个位置来访问记录，以加快查找的速度。腾讯轻联允许用户创建，哈希表结构：内容为 key-value 格式的数据。

### 列表

也可称为数组，是有序的元素序列，格式为一组拥有 ID 的数据信息。数组是用于储存多个相同类型数据的集合。

### 字符串

字符串（String）是由数字、字母、下划线组成的一串字符。腾讯轻联通过数据存储功能，可以创建和使用字符串，帮助用户完成快速的复用，简化操作。

# 全局变量

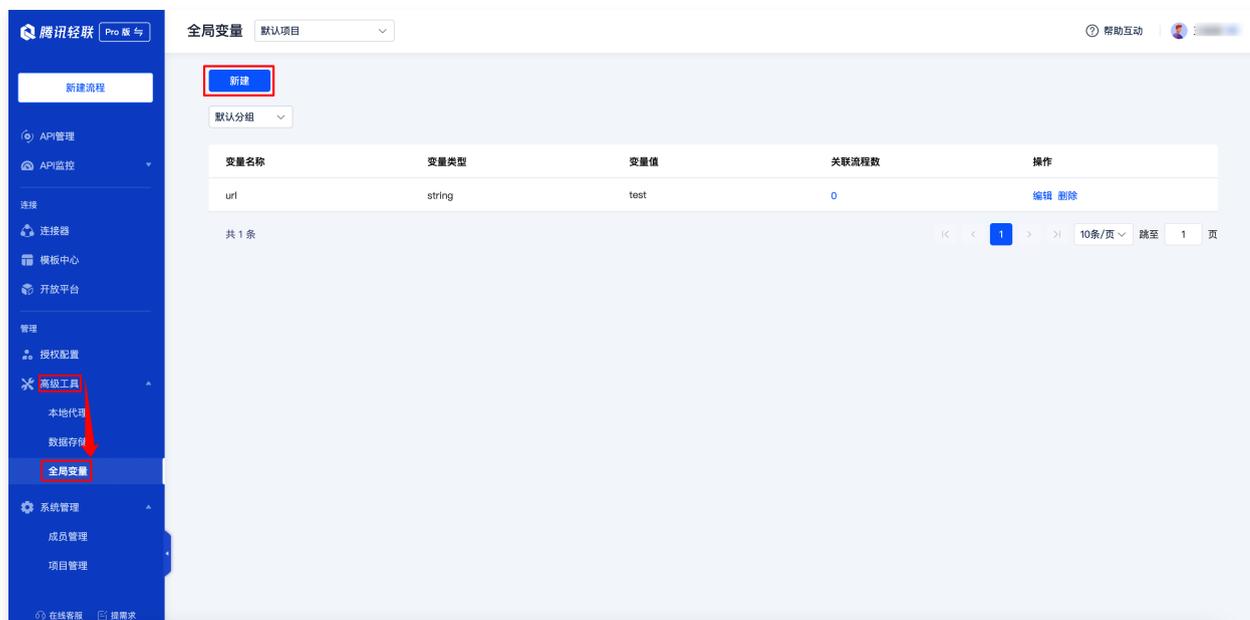
最近更新时间：2024-11-07 11:41:52

## 简介

全局变量是一种项目层级内可复用，且支持修改后热更新到关联应用的变量，在编排集成流时在相关连接器/组件中通过面板点选即可使用。全局变量分为系统变量和自定义变量，其中系统变量包括项目名称、应用名称、环境名称；自定义变量支持用户自定义，上限20个。

## 全局变量配置入口

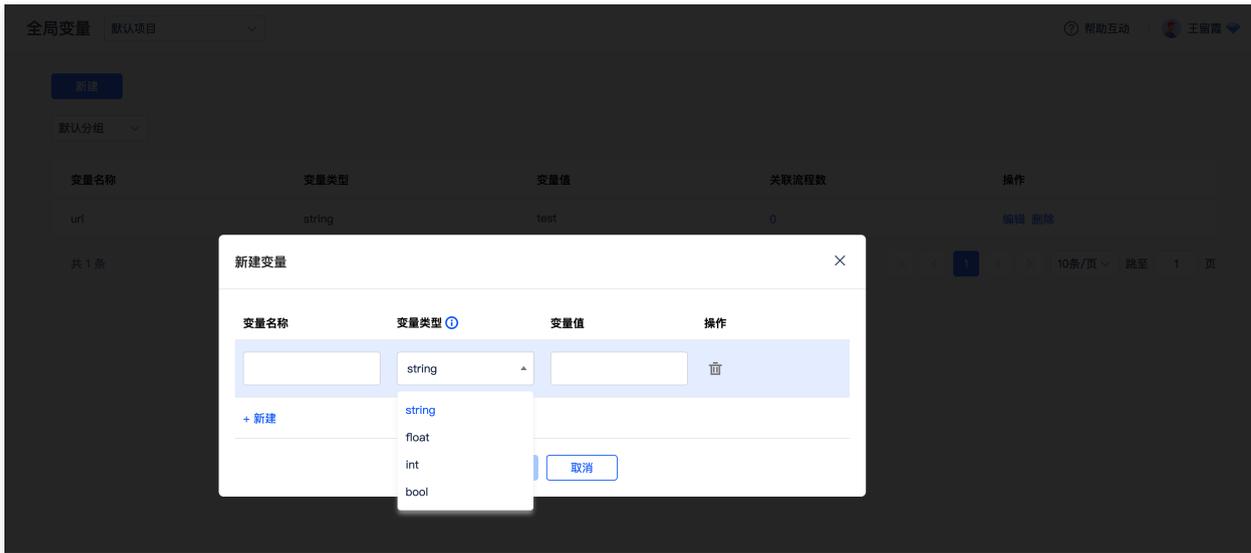
1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击高级工具 > 全局变量。



2. 自定义新建的变量保存在此列表中，上限为20个。



全局变量支持四种类型：string、float、int、bool。



## 全局变量分组

全局变量和连接配置的分组是互通的。不同分组下的全局变量的变量值可以不同，适用于不同生产环境下发布应用时使用不同分组的全局变量。



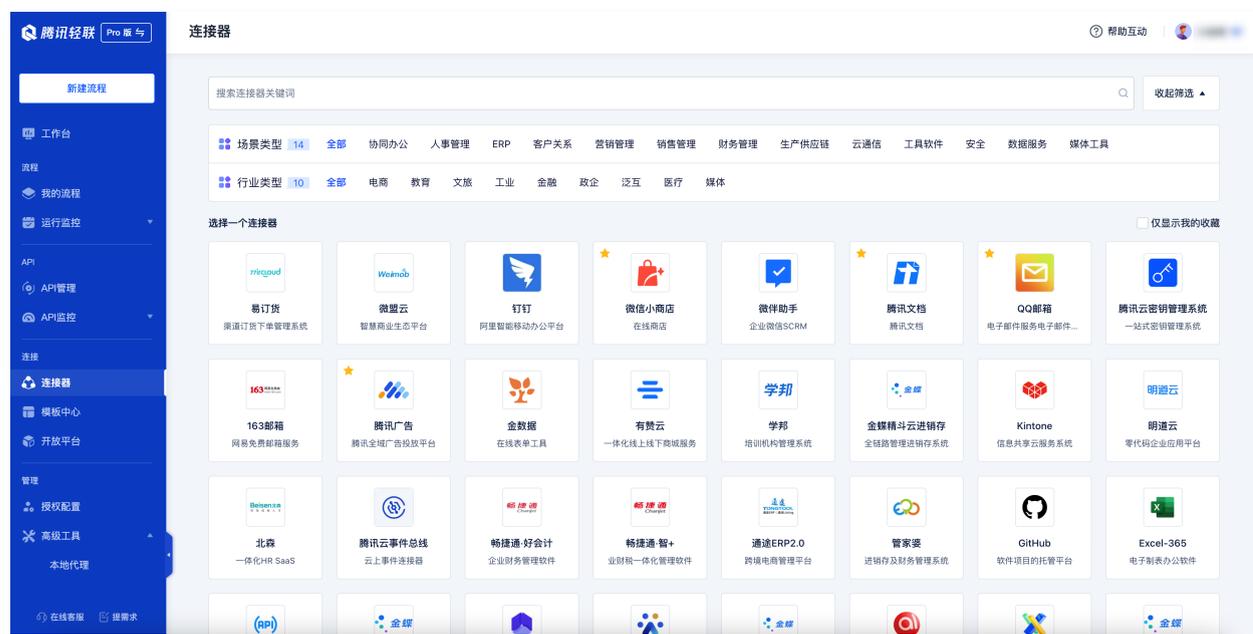
# 连接 连接器

最近更新时间：2025-07-01 12:03:11

## 操作场景

连接器市场页面集中展示腾讯轻联平台对外开箱即用的全部连接器资源，且部分连接器支持自定义编辑，用户可根据业务需要进行对已有字段或者操作进行新增或删除。目前连接器按照场景和行业做了基础划分，方便客户尽快筛选符合业务场景的连接器，且支持按照字段检索。

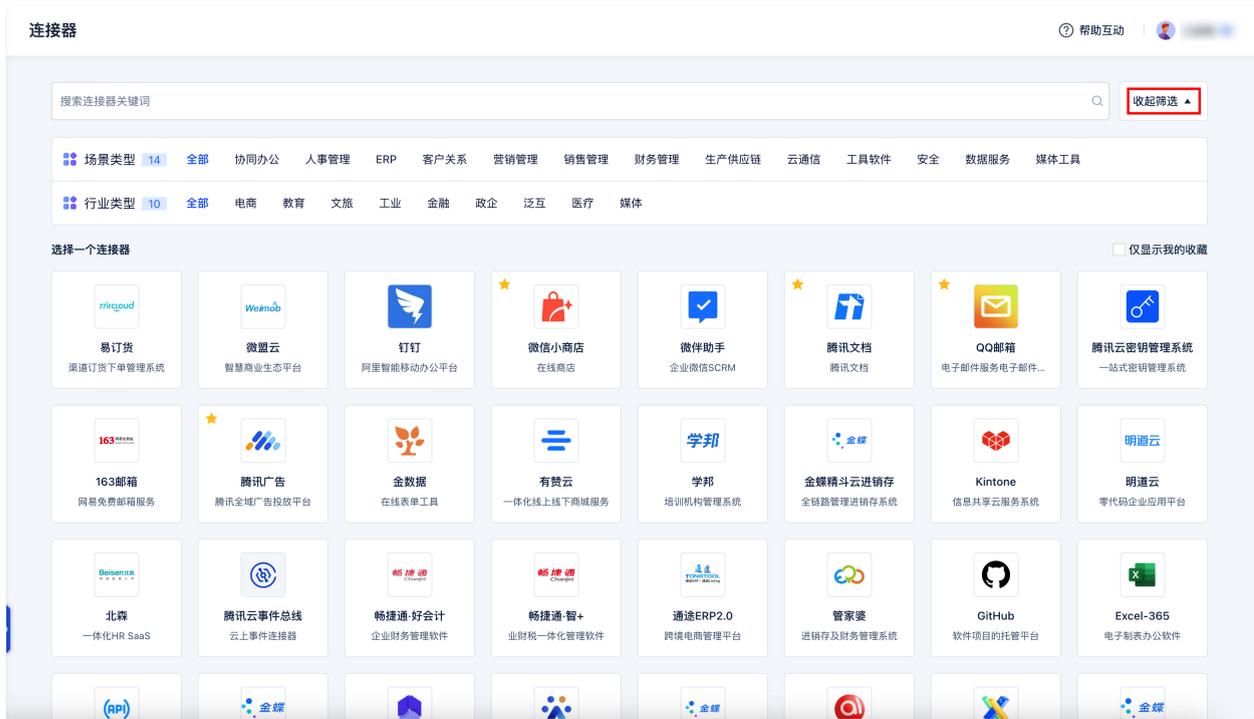
- 场景：协同办公、人事管理、ERP、客户关系、营销管理、销售管理、财务管理、生产供应链、云通信、工具软件、安全数据服务。
- 行业：电商、教育、文旅、工业、金融、政企、泛互、医疗。



## 操作步骤

### 连接器概览页

登录 [腾讯轻联控制台](#)，在左侧导航栏单击[连接](#) > [连接器](#)。



## 收藏连接器

单击连接器前的★即可收藏此连接器，下次可通过勾选仅显示我的收藏，快速找到此连接器。

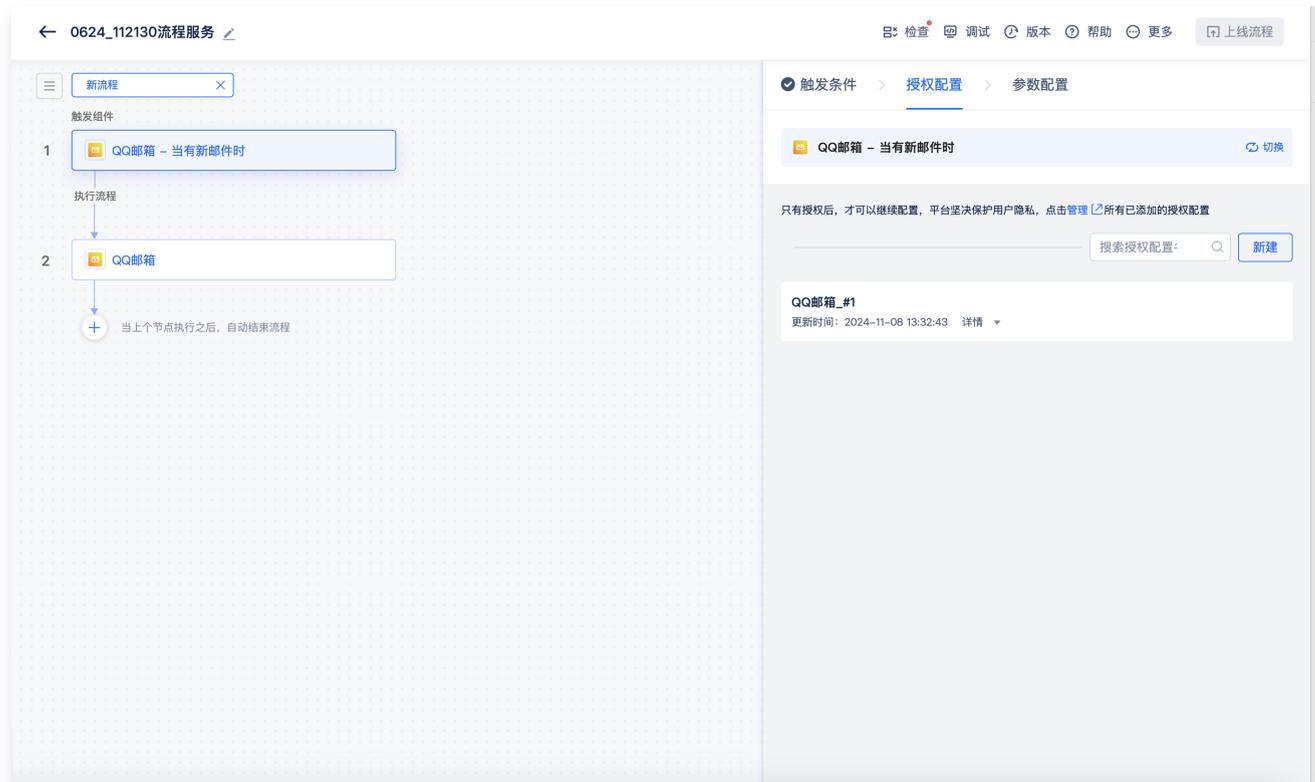


## 查看连接器信息

1. 单击连接器名称，进入连接器详情页，可查看连接器的简介、版本、更新日志、连接器文档等信息。下图以 QQ 邮箱为例：

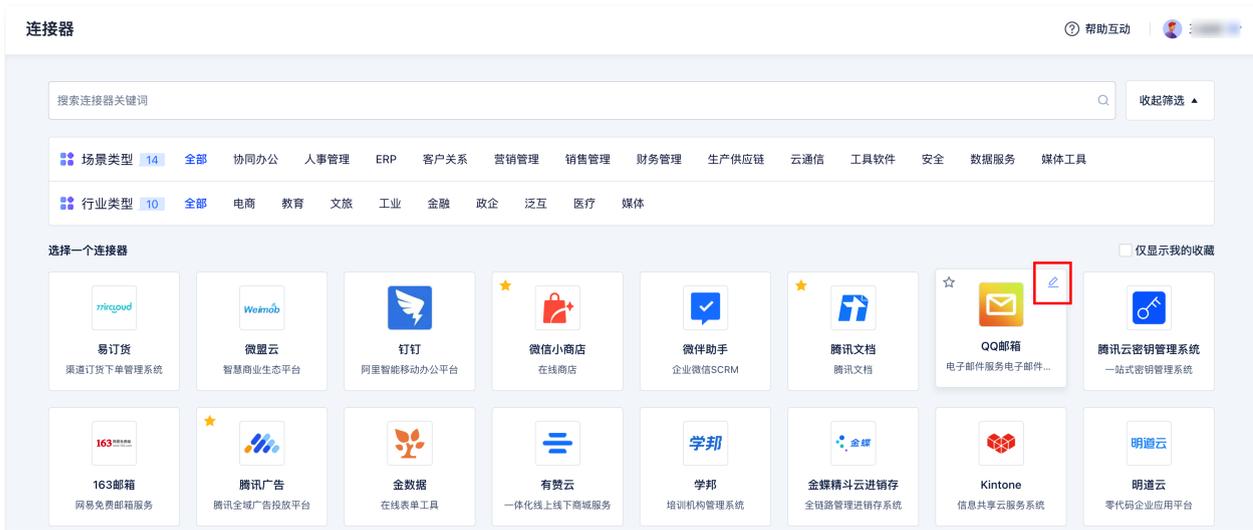


2. 单击 + 新建包含 QQ 邮箱的流程，即可跳转到流程页面进行后续编排。



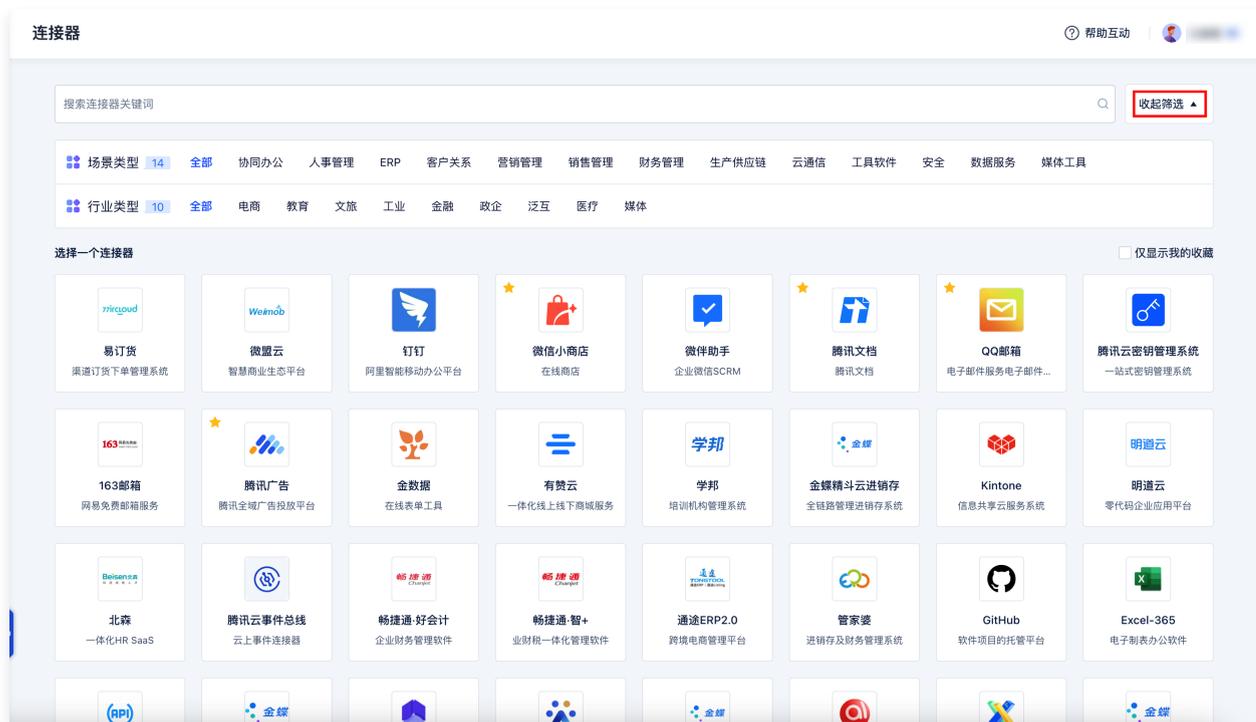
### 编辑连接器

部分连接器不支持编辑仅展示查看操作，对于可编辑的连接器，单击编辑按钮可以编辑此连接器。可进行重命名、修改连接器属性、增加或删除等操作，详情请参见 [连接器开发](#)。



### 筛选连接器

- 单击筛选图标，支持通过场景、行业来筛选连接器。



- 单击仅显示我的收藏可筛选已被收藏的连接器。



- 支持按照名称搜索连接器。

连接器 帮助互动

微信 收起筛选

场景类型 14 全部 协同办公 人事管理 ERP 客户关系 营销管理 销售管理 财务管理 生产供应链 云通信 工具软件 安全 数据服务 媒体工具

行业类型 10 全部 电商 教育 文旅 工业 金融 政企 泛互 医疗 媒体

选择一个连接器  仅显示我的收藏

-  微信小程序商店 在线商店
-  微信公众号 微信服务号、订阅号
-  微信小程序平台 微信小程序平台
-  企业微信机器人 IM自动化程序
-  企业微信 专业办公管理工具

# 开放平台

最近更新时间：2024-11-07 11:41:52

## 操作场景

腾讯轻联设有连接器开发模块，支持用户按需自建连接器，连接器开发页面集中展示了当前账号下所有的自定义连接器，您可以浏览所有由您（主账号及子账号）创建、编辑、发布的连接器及其状态，便于管理和操作。

不同的连接器状态，对应操作下展示的功能是有区别的，具体如下：

状态	功能
已发布	允许用户进行查看、分享、编辑和删除
编辑中	允许用户分享、发布、编辑和删除
审核中	允许用户撤销、查看
已拒绝	允许用户编辑、删除
预发布	允许用户发布、编辑和删除

## 操作步骤

### 步骤1：添加连接器

当平台预提供的连接器不满足您的业务需求时，您可以在开发者中心添加新的连接器或者修改已有的连接器来方便您后续业务的使用。

1. 登录 [腾讯轻联控制台](#)。
2. 单击添加连接器。



3. 根据页面提示，配置信息后，单击确认进入编辑连接器界面进行配置。

### 步骤2：配置连接器

在配置连接器界面，您可以对连接器的基本信息、连接器属性、操作、触发器、自定义数据类型进行配置。

#### 说明：

您可以单击以下页签，查看具体的配置说明。

#### 基本信息

在基本信息界面，您可以添加连接器的描述信息、上传连接器图标（不上传则使用默认图标 ），配置完成后单击保存。

### 连接器属性

在连接器属性界面，您可以通过**新建属性**或**编辑**来新建或者修改属性，属性是连接器操作和触发器的一些共用参数。它们将会作为连接器的配置信息被使用。

配置连接器

- 基本信息
- 连接器属性**
- 操作
- 触发器
- 自定义数据类型

测试连接器

< 返回连接器属性列表

基本配置

属性名

属性名称只能包含字母、数字、中划线和下划线，只能以字母开头，长度不能超过50个字符

属性展示名  ⓘ

长度不超过16个字符

属性类型  ▾

必填  是否为必填

属性描述

0

长度不超过200个字符

显示配置

属性展示分组  ⓘ

长度不超过50个字符

属性条件  ⓘ

属性选项卡  ⓘ

长度不超过20个字符

高级配置

表达式模式  ▾

展示样例

长度不超过200个字符

是否为私有属性  是否为私有属性 ⓘ

密码模式  开启密码模式

内容校验  启用内容校验 ⓘ

保存

取消

单击配置测试连接，可以开启连接器的连接性测试功能，开通后后续您使用相关连接器时，填写完连接器信息后可以预先测试连接器的可用性。

← 编辑连接器

配置连接器 | 基本信息 | **连接器属性** | 操作 | 触发器 | 自定义数据类型 | 预发布 | UI预览

测试连接器

新建属性 | **配置测试连接** |  配置安全开关 | 请输入属性名

连接器属性名	属性类型	属性展示名	①属性展示分组	操作
username	string	账号ID		编辑 删除
password	string	密码		编辑 删除
client_id	string	ClientId		编辑 删除
client_secret	string	ClientSecret		编辑 删除

属性名称支持拖拽 :: 调整顺序，列表中顺序将决定连接器属性在界面的展示顺序。详细展示效果请参考UI预览。  
共 4 条

保存 重置

开启测试连接功能。

测试连接功能

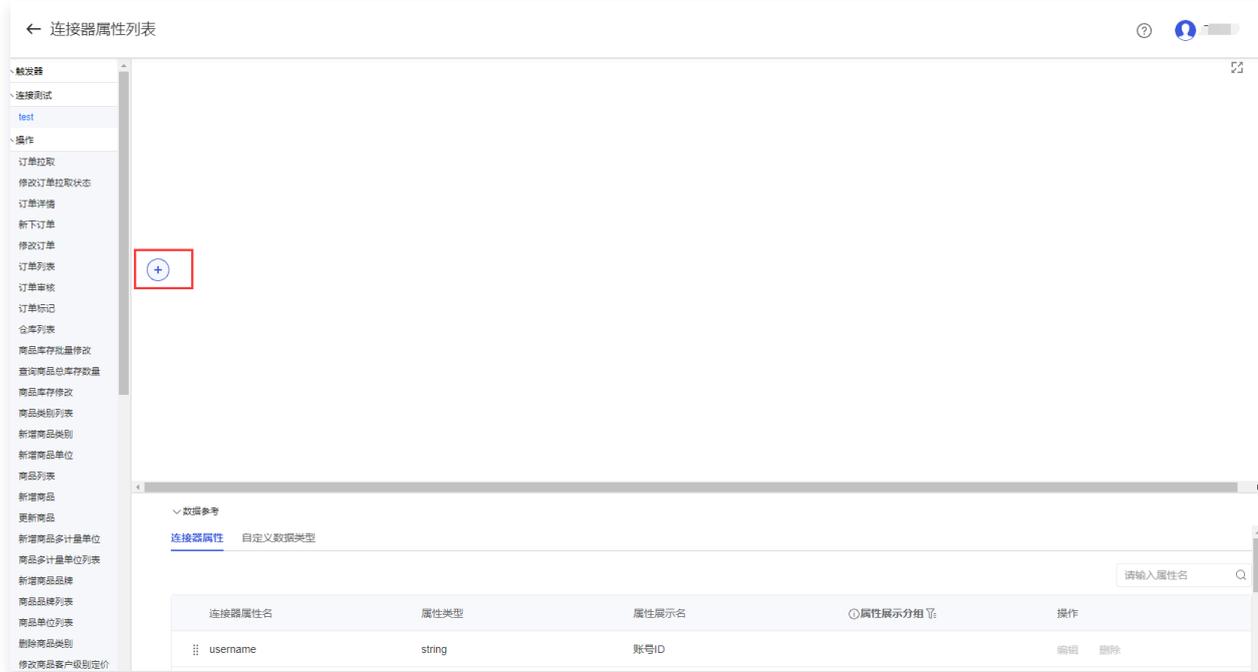
测试连接功能  开启测试连接功能 ①

测试连接展示名

点击进入 **流程预览** 界面编辑测试连接器流程

确定 取消

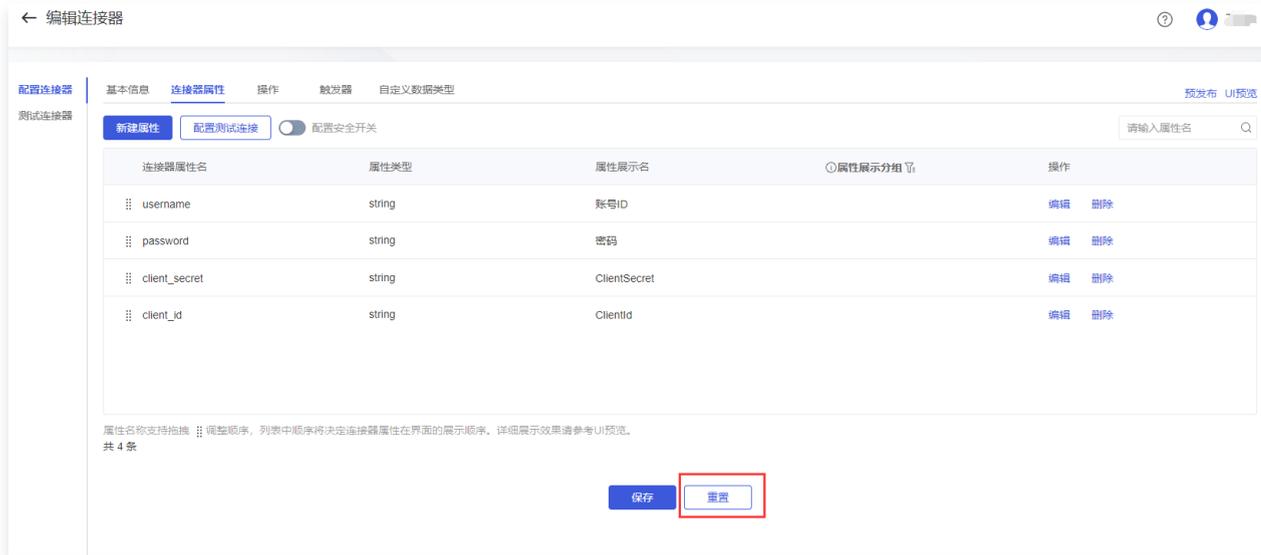
单击**流程预览**，可以编辑测试连接器流程，您需要完成配置测试连接的具体实现。



单击**连接器属性**操作列的**删除**，可以删除此属性。删除后相关数据不可恢复，请您谨慎操作。

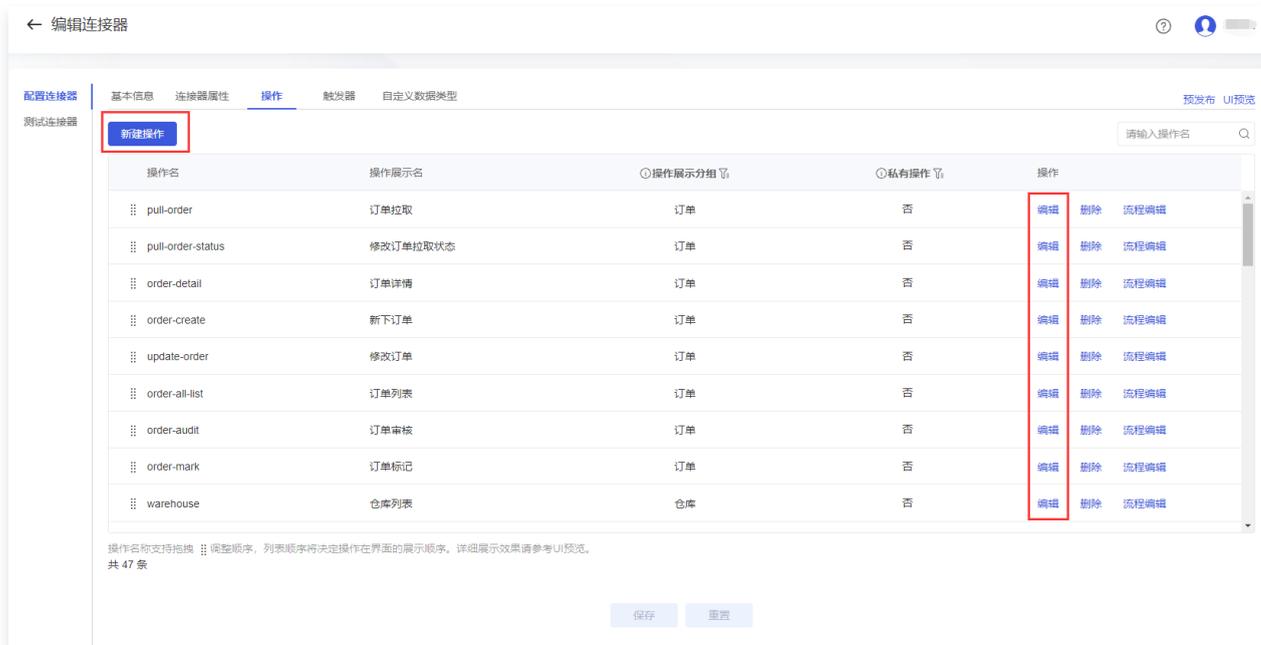


当前页面误删除或拖拽对应属性后，立即单击**重置**可恢复上一步的操作的内容，若跳转页面即默认保存删除操作无法重置。



### 操作

当平台预提供的连接器操作不满足您的业务需求时，您可以修改已有连接器的操作或者是新增连接器并新增操作方便后续使用。在操作页面，您可以通过**新建操作**或者**编辑**对已存在的操作基本信息进行新建或编辑。操作封装了由集成应用主动发起的交互，在流中做流程编排和消息处理，并在处理完成后输出一个新的消息。



所有的操作都需要对其具体的实现进行编辑。单击**流程编辑**，页面将跳转到画布中，并帮助您完成此次操作的具体实现，您可以像绘制一条集成流一样绘制此操作的流程。

配置连接器

基本信息 连接器属性 **操作** 触发器 自定义数据类型

测试连接器

< 返回操作列表

操作基本信息

操作名 \*

名称只能包含小写字母、数字、中划线，并且只能以小写字母开头，长度不超过50个字符

操作展示名  ⓘ

长度不超过16个字符

操作展示分组  ⓘ

操作描述

输出类型 payload schema预览

暂无描述返回数据的结构  
[使用示例快速创建](#) 或 [手工添加](#)

attributes schema预览 (输入json校验必须是dict)

暂无描述返回数据的结构  
[使用示例快速创建](#) 或 [手工添加](#)

输出错误类型  payload  attributes ⓘ

私有操作  是否为私有操作 ⓘ

操作参数配置

+ 新建操作参数配置

参数名称	参数展示名	参数展示分组	操作
params	连接参数		<a href="#">编辑</a> <a href="#">删除</a>

参数名称支持拖拽 ⓘ 调整顺序，列表中顺序将决定其在界面的展示顺序。详细展示效果请参考UI预览。

流程编辑

[开始流程编辑](#)

保存 取消

单击具体属性的删除，可以删除此操作。

← 编辑连接器

配置连接器 基本信息 连接器属性 操作 触发器 自定义数据类型 预发布 UI预览

测试连接器 新建操作 请输入操作名

操作名	操作展示名	操作展示分组	私有操作	操作
pull-order	订单拉取	订单	否	编辑 删除 流程编辑
pull-order-status	修改订单拉取状态	订单	否	编辑 删除 流程编辑
order-detail	订单详情	订单	否	编辑 删除 流程编辑
order-create	新下订单	订单	否	编辑 删除 流程编辑
update-order	修改订单	订单	否	编辑 删除 流程编辑
order-all-list	订单列表	订单	否	编辑 删除 流程编辑
order-audit	订单审核	订单	否	编辑 删除 流程编辑
order-mark	订单标记	订单	否	编辑 删除 流程编辑
warehouse	仓库列表	仓库	否	编辑 删除 流程编辑

操作名称支持拖拽，调整顺序，列表顺序将决定操作在界面的展示顺序。详细展示效果请参考UI预览。共 47 条

保存 重置

当前页面误删除或拖拽对应操作后，立即单击重置可恢复上一步的操作的内容，若跳转页面即默认保存删除操作无法重置。

← 编辑连接器

配置连接器 基本信息 连接器属性 操作 触发器 自定义数据类型 预发布 UI预览

测试连接器 新建操作 请输入操作名

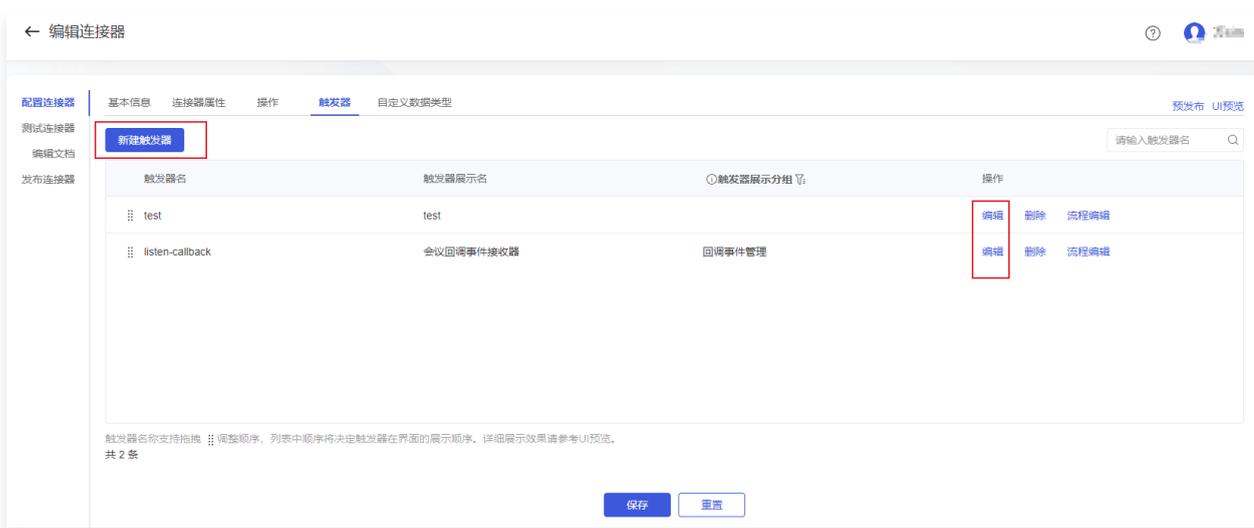
操作名	操作展示名	操作展示分组	私有操作	操作
pull-order	订单拉取	订单	否	编辑 删除 流程编辑
pull-order-status	修改订单拉取状态	订单	否	编辑 删除 流程编辑
order-detail	订单详情	订单	否	编辑 删除 流程编辑
order-create	新下订单	订单	否	编辑 删除 流程编辑
update-order	修改订单	订单	否	编辑 删除 流程编辑
order-all-list	订单列表	订单	否	编辑 删除 流程编辑
order-audit	订单审核	订单	否	编辑 删除 流程编辑
order-mark	订单标记	订单	否	编辑 删除 流程编辑
warehouse	仓库列表	仓库	否	编辑 删除 流程编辑

操作名称支持拖拽，调整顺序，列表顺序将决定操作在界面的展示顺序。详细展示效果请参考UI预览。共 47 条

保存 重置

### 触发器

当平台预提供的连接器的触发器不满足您的业务需求时，您可以修改已有连接器的触发器或者是新增连接器并新增触发器方便后续使用。在触发器页面，您可以通过新建触发器或者编辑对已存在的触发器基本信息及参数进行新建或编辑。触发器封装了由第三方发起、集成应用被动监听的交互，在流中作为源，在收到第三方系统的交互事件时在流中产生消息。触发器的使用流程与操作基本一致。



所有的触发器也需要对其具体的实现进行编辑。单击**流程编辑**，页面将跳转到画布中，并帮助您完成此次操作的具体实现。

配置连接器

- 基本信息
- 连接器属性
- 操作
- 触发器**
- 自定义数据类型

测试连接器

[< 返回触发器列表](#)

触发器基本信息

触发器名 \*

名称只能包含小写字母、数字、中划线，并且只能以小写字母开头，长度不超过50个字符

触发器展示名  ⓘ

长度不超过16字符

触发器展示分组  ⓘ

长度不超过50字符

触发器描述

长度不超过500字符

输出类型 payload schema预览



暂无描述返回数据的结构

[使用示例快速创建](#) 或 [手工添加](#)

attributes schema预览 (输入json校验必须是dict)



暂无描述返回数据的结构

[使用示例快速创建](#) 或 [手工添加](#)

触发器参数配置

[+ 新建触发器参数配置](#)

参数名称	参数展示名	参数展示分组	操作

参数名称支持拖拽 ⓘ 调整顺序，列表中顺序将决定其在界面的展示顺序。详细展示效果请参考UI预览。

流程编辑

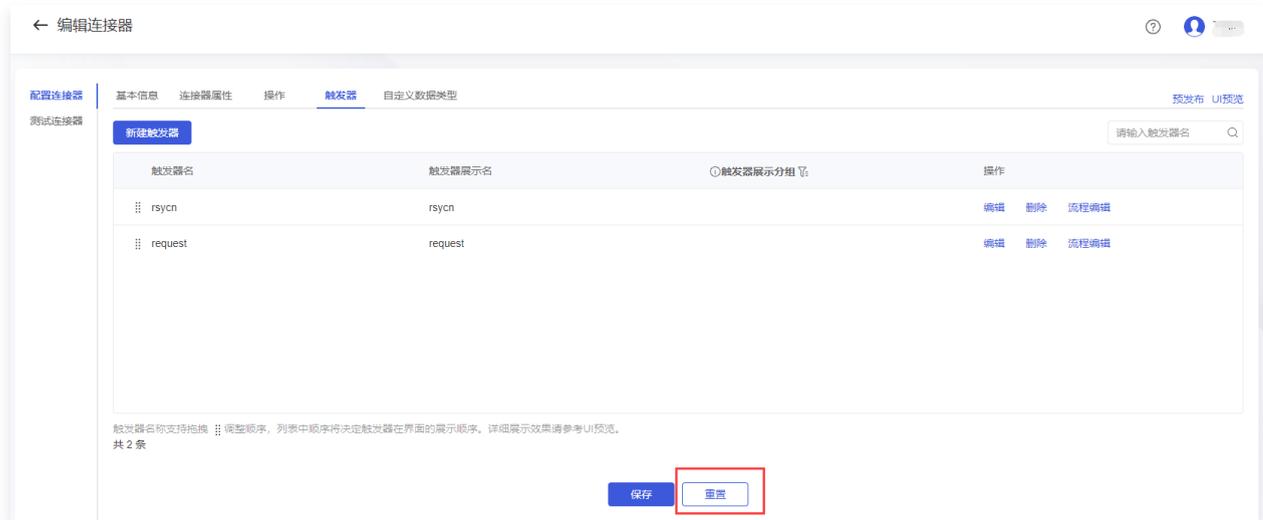
[点击开始编辑](#)

保存 取消

单击具体触发器的删除，可以删除此触发器。

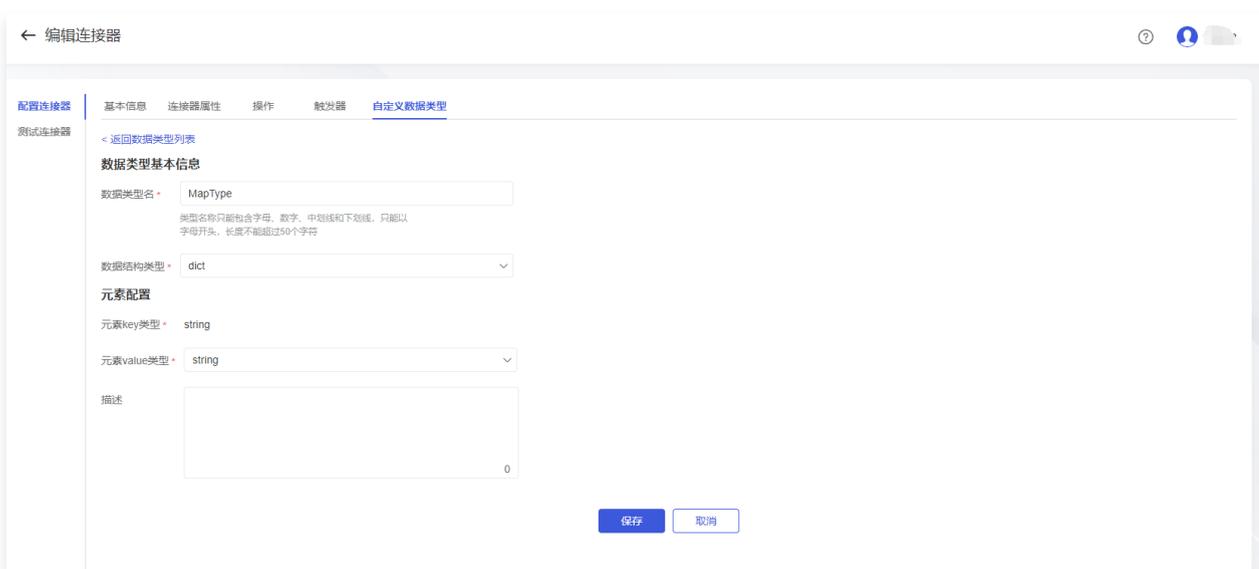


当前页面误删除或拖拽对应操作后，立即单击重置可恢复上一步的操作的内容，若跳转页面即默认保存删除操作无法重置。

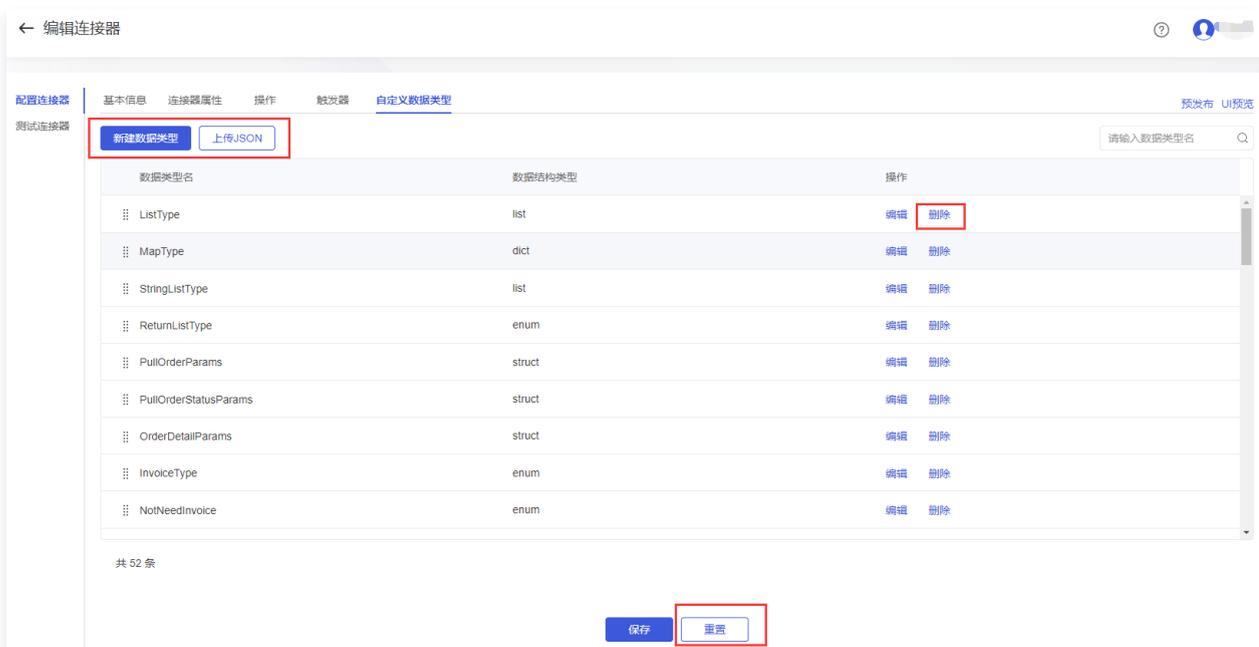


### 自定义数据类型

在自定义数据类型页面，通过新建数据类型和编辑可以创建和编辑连接器对外交互的自定义数据格式。也可以通过上传 JSON 后台会自动识别数据类型并创建。需要注意的是，不同类型的数据结构类型对应的配置信息可能会有不同。

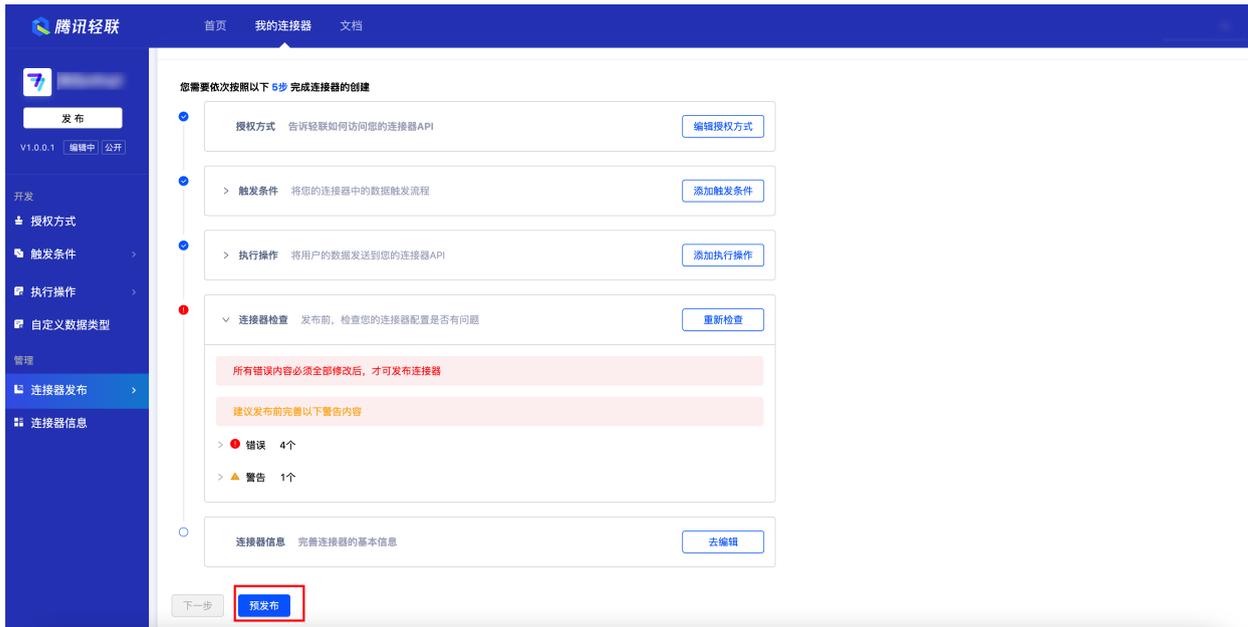


单击具体数据类型的删除，可以删除此数据类型。需要注意的是：已经被引用的数据类型无法删除，需要先解绑定关系再进行删除操作。若在当前页面误操作删除后立即单击重置可恢复上一步的操作的内容，若跳转页面即默认保存删除操作无法重置。



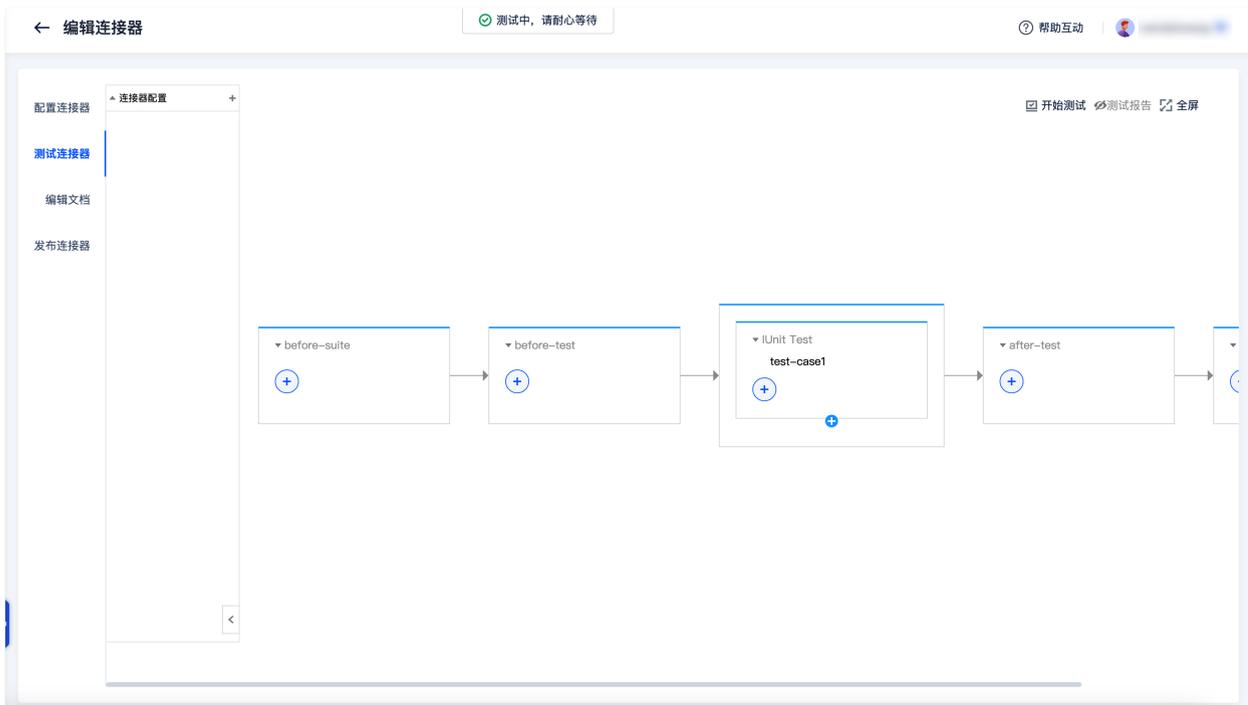
### 步骤3：预发布连接器

在编辑连接器页面，单击编辑连接器右上角的预发布，可以发布连接器到测试环境。需要注意的是，SaaS 类型的连接器将直接进入预发布流程，但是如果是自建类型的连接器，您需要填写版本号之后才可以预发布成功，且版本号也方便您后续连接器功能更新对应不同的版本迭代。预发布之后的连接器可以在所有非正式环境的流中使用，同时允许用户对此连接器进行应用测试和单元测试，直到您觉得此连接器已经满足了所有的发布条件。在进行发布操作。



## 步骤4：测试连接器

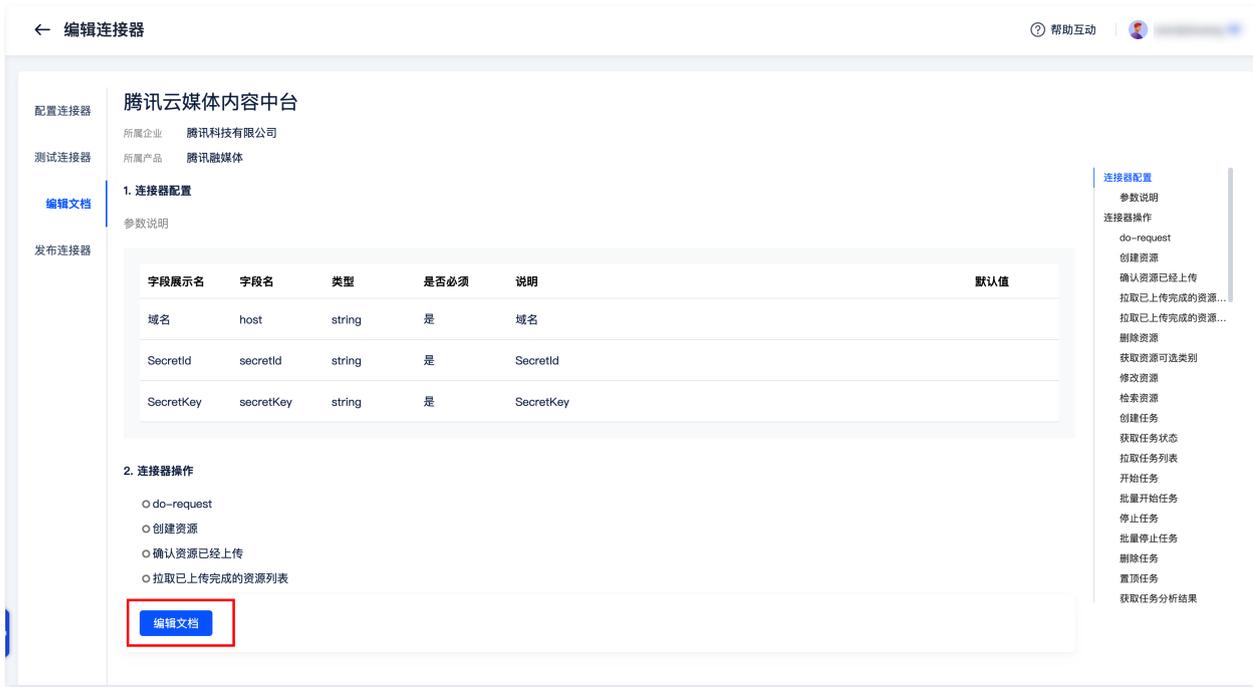
只有预发布成功的连接器才可以进行测试用例的配置，如果该连接器没有通过测试，则不能进行发布。测试连接器详细可参见 [流程配置](#)。



## 步骤5：编辑文档

连接器文档中的大部分内容都将会根据连接器的配置自动生成。您需要手动配置的内容如下，此类信息可以方便发布后其他用户进一步了解如何使用此连接器。

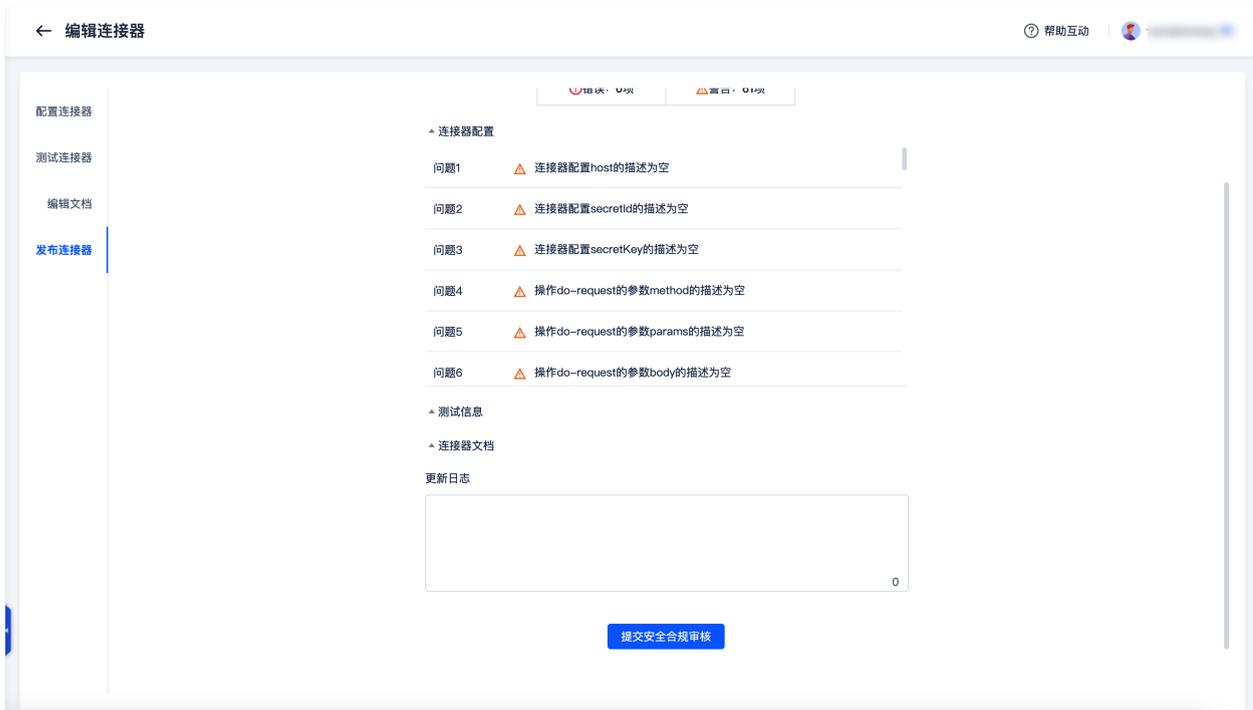
- 连接器操作/触发器是否配置返回值及返回信息。
- 异常信息。
- 连接器补充信息（例如官网链接和 API 文档地址等）。



## 步骤6：发布/撤销

在发布连接器之前，请确保连接器配置信息均已经完备和准确无误，一旦进入审核流程，您将无法直接修改此连接器的配置信息，直到审核流程结束或者您手动撤销审核。

在我的连接器界面单击**发布**或者在编辑连接器界面单击**发布连接器**，系统将进入连接器完成度校验，并且向您提供此次连接器发布审核的校验报告。如果存在错误，请参考提示文字并完善对应的配置内容，否则此连接器无法被发布；同时我们会帮助您标注出一些警告信息，警告信息并不会妨碍您连接器的正常发布，但是可能会对此连接器的使用造成一定的影响，建议您酌情修复。



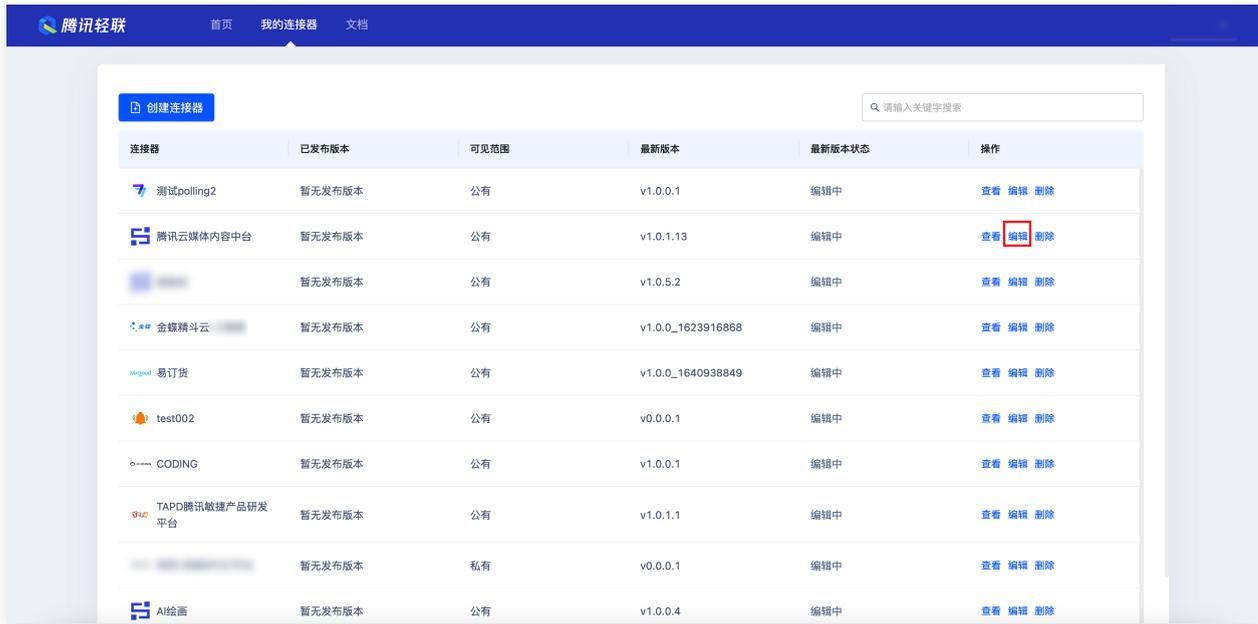
当一切准备妥当并且系统的自动完成度校验没有检测到问题的时候，您便可以提交此连接器进入人工审核的流程中。我们的工程师将会尽快审核您的变更申请，并且第一时间将审核结果反馈给您。您可以随时回到当前页面查看审核进度和结果，如果连接器审核通过，您将可以在连接器开放平台中看到它。若您还需要进一步修改，也可以撤销发布。



## 相关操作

### 编辑连接器

1. 登录 [腾讯轻联控制台](#)。
2. 在[连接器开发](#)页面，单击操作列的**编辑**，前往编辑连接器界面。



3. 具体的编辑内容请参见 [配置连接器](#)。

# 模板中心

最近更新时间：2024-11-07 11:41:52

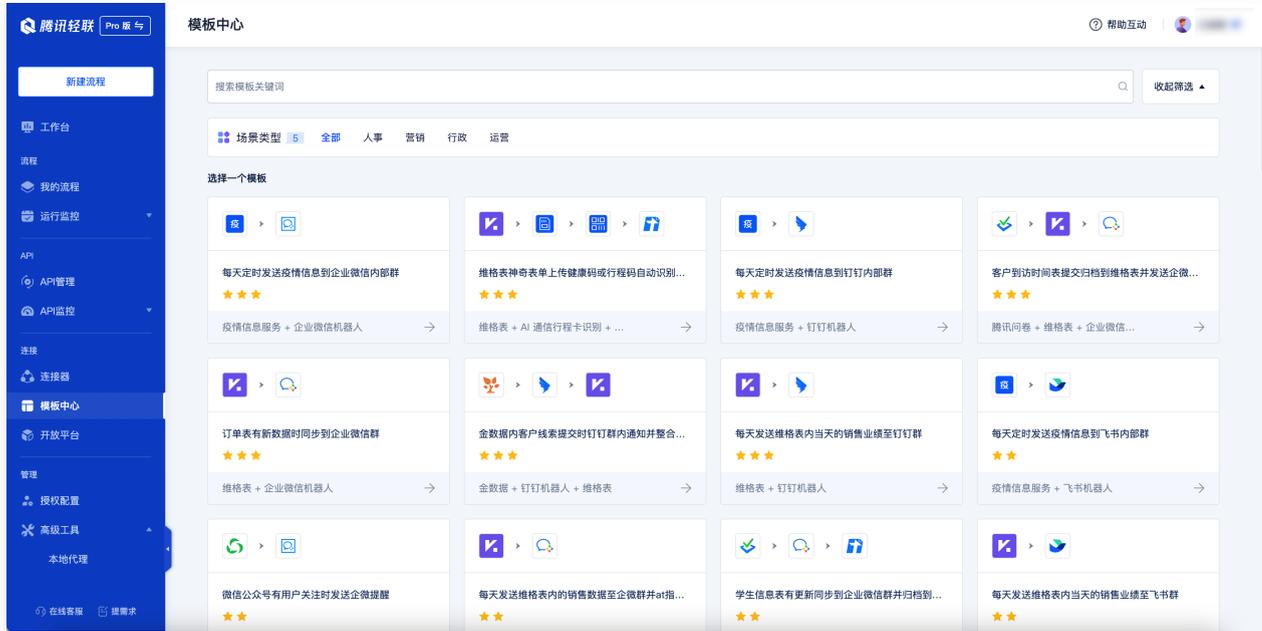
## 操作场景

模板中心集中展示主流连接器常用模板，用户可根据业务所需，找到要连接的应用系统，并搜索相关模板快速搭建自动化流程。

场景：人事、营销、行政、运营（更多场景接入中）。

## 操作步骤

1. 登录 [轻联控制台](#)，选择连接 > 模板中心，进入模板首页。



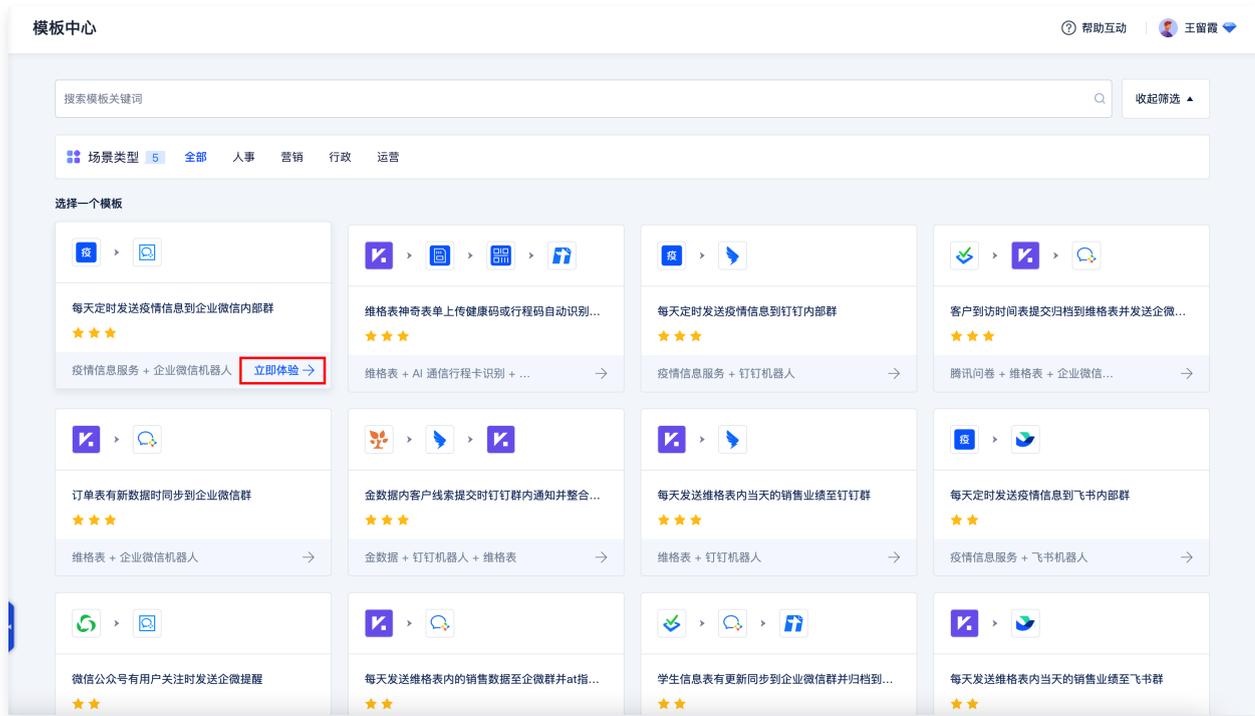
**模板组成：**使用场景 + 所关联的连接器。

2. 搜索所需模板，支持搜索如下：

- 通过搜索连接器名称，快速找到相关模板。
- 根据场景类型筛选模板。

3. 根据所选模板创建流程。

鼠标移动到立即体验上，文字高亮后单击前往。



4. 模板内根据客户自己需求，绑定连接器账号，更改模板内到逻辑即可。

流程相关操作可参见[流程](#) > [我的流程](#) > [流程配置](#)。

# 系统管理

## 成员管理

最近更新时间：2024-08-01 15:33:02

### 操作场景

腾讯轻联是采用 CAM 的账号系统并结合腾讯轻联的单独权限进行权限控制。

腾讯轻联会自动拉取当前主账号下的所有子账号进入成员列表，若要添加/删除成员，需要通过腾讯轻联控制台调整到 CAM 控制台进行操作。主账号成为系统管理员，系统管理员拥有腾讯轻联的最高权限，可设置子账号成为系统管理员或撤销子账号的系统管理员权限。详细请参见 [项目管理 - 成员管理](#)。

### 操作步骤

#### 添加/删除成员

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心 > 成员管理**。
2. 在成员管理页面，单击**添加成员/删除**。



3. 在确认弹框中单击**确认前往**，跳转至 CAM 用户列表界面进行操作。



#### 设置/解除系统管理员

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心 > 成员管理**。
2. 在成员管理页面，单击操作列的**设置为/解除系统管理员**。主账号默认为系统管理员不可解除。

成员管理

添加成员 ①

全部成员 仅管理员

成员名称	账号ID	账号类型	权限	操作
		主账号	系统管理员	--
		子账号	系统管理员	删除 解除系统管理员
		子账号	--	删除 设置为系统管理员
		子账号	--	删除 设置为系统管理员
		子账号	--	删除 设置为系统管理员
		子账号	系统管理员	删除 解除系统管理员

共 6 条

10条/页 跳至 1 页

### 查看成员详情

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心 > 成员管理**。
2. 在成员管理页面，单击目标成员名称，即可查看成员详情。

成员管理

添加成员 ①

全部成员 仅管理员

成员名称	账号ID	账号类型	权限	操作
		主账号	系统管理员	--
		子账号	系统管理员	删除 解除系统管理员
		子账号	--	删除 设置为系统管理员
		子账号	--	删除 设置为系统管理员
		子账号	--	删除 设置为系统管理员
		子账号	系统管理员	删除 解除系统管理员

共 6 条

10条/页 跳至 1 页

成员详情包括基本信息、授权信息，具体如下：

← 成员详情



ee [redacted]

账号ID 1C [redacted]

账号类型 子账号

联系邮箱 -

联系微信 15 [redacted]

联系手机 [redacted]

授权信息

设置为系统管理员

项目名称	应用名称	授权功能
project	HelloWorld	-
project	HelloWorldaaa	-
project	问卷信息采集_0510_150053	-
project	HelloWorld	-
project	测试return	-
project	HelloWorldaaa	-
project	问卷信息采集_0510_150053	-

# 项目管理

最近更新时间：2025-06-20 18:02:42

## 操作场景

集成需求可以通过创建集成应用进行实施，集成应用中可创建多个集成应用，每个集成应用对应一个集成场景。

添加集成应用后，可以查看项目详情、修改项目名称、删除项目。

- 项目名称：通过列表展示当前用户有权限的所有项目的名称。  
默认项目：所有成员默认有默认项目的权限。不支持添加或删除成员，也不支持删除项目，且不支持修改项目名称。
- 项目管理员：最多显示三个项目管理员。
- 描述：用户可以输入文字简单描述项目的内容，方便后续成员进一步区分理解对应项目的内容。
- 项目成员：当前项目包含的成员数量。
- 创建时间：项目首次创建的时间。
- 操作：添加成员、修改项目名称、删除项目。

## 前提条件

已添加 [集成应用](#)。

## 操作步骤

### 添加项目

您可以从项目维度管理集成应用，便于区分业务场景和权限划分。项目支持查看详情、修改名称和删除操作。

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心 > 项目管理**。
2. 在项目管理页面，单击**添加项目**。若提示无权限创建项目，可参见 [访问管理](#) 分配合理权限。
3. 填写项目名称和描述，单击**确定**完成添加。

添加项目 ×

项目名称 \*

请输入25位以内的中文、字母、数字、\_或-

描述

描述

### 删除项目

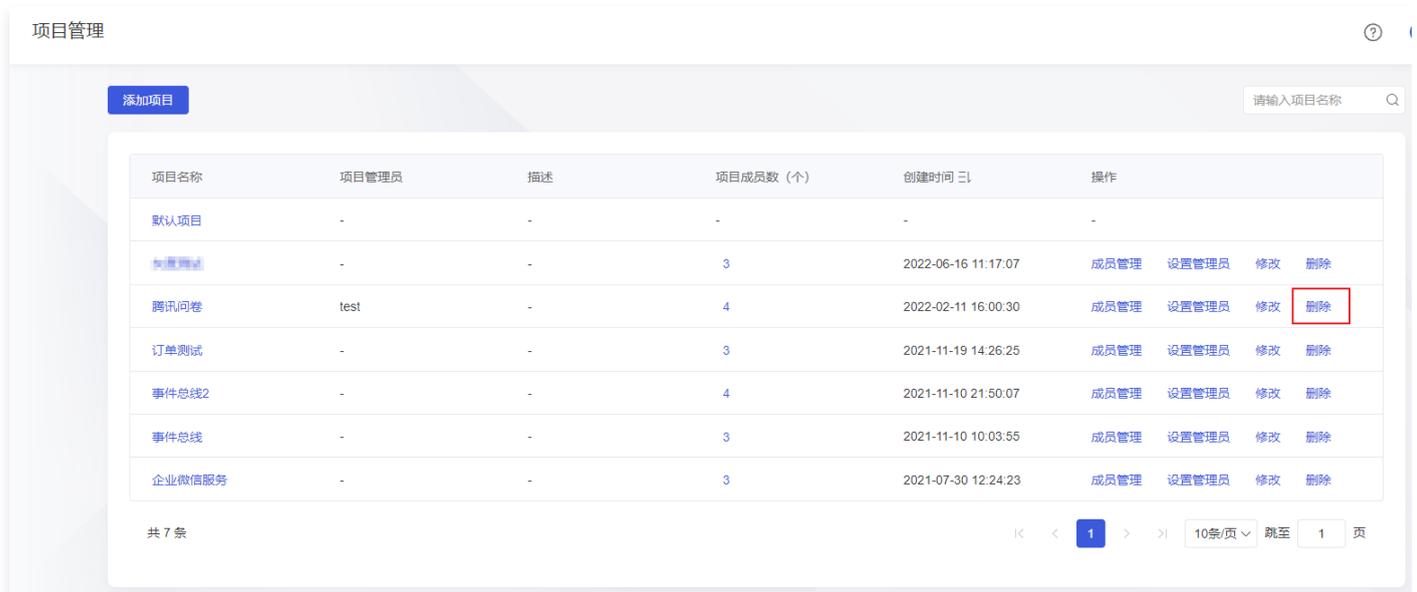
当某项目不再使用时，您可以**停止**项目内应用运行后，再删除项目。

#### ⚠ 注意：

当项目中存在运行中的应用时，无法删除。且项目删除后数据无法恢复，请谨慎操作。

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心 > 项目管理**。

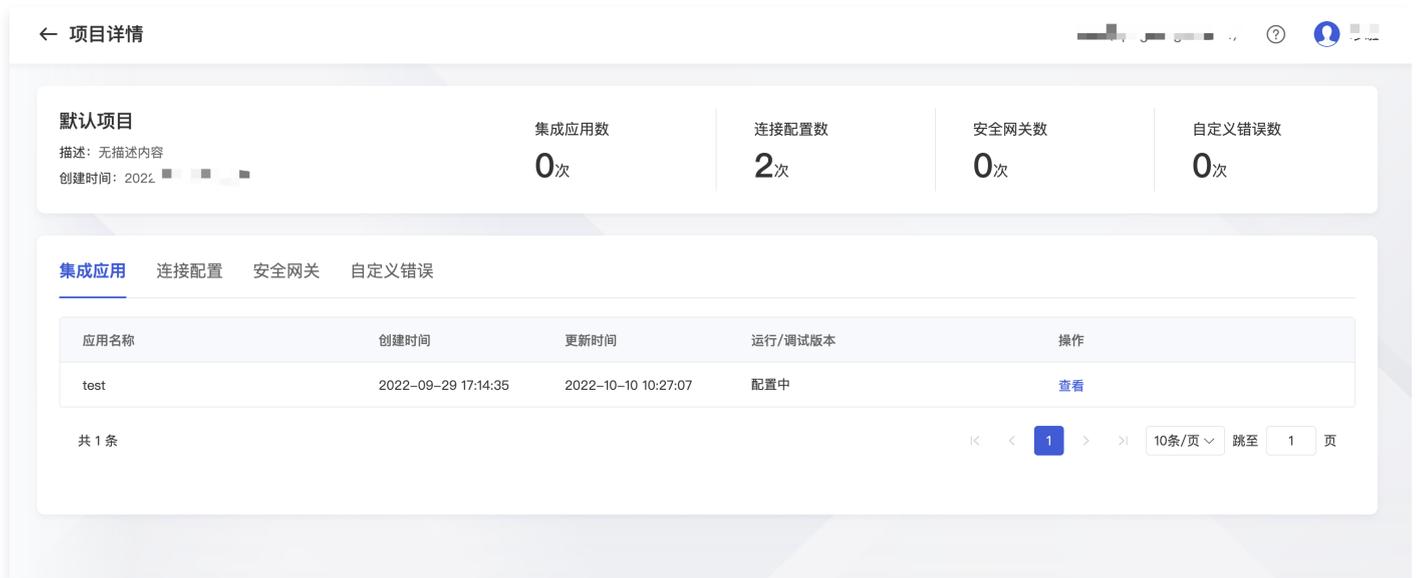
2. 在项目管理页面，单击操作列的删除，并在确认弹框中单击确认删除。



## 查看项目

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心 > 项目管理**。
2. 在项目管理页面，单击目标项目名称，进入项目详情页。

项目详情页包含了对应项目下项目集成应用、连接配置、安全网关和自定义错误的列表信息，您可以单击对应名称快速进入编辑页面。



## 成员管理

支持进行项目成员管理，可以添加或删除协同开发、运维、发布此项目的成员。可添加或删除的成员列表来源于成员管理。

系统管理员、项目管理员可以添加删除项目中的普通成员，也可以赋予或取消普通成员的项目管理员权限，也可以修改普通用户对该项目下集成应用的只读、编辑、编辑并发布的操作权限。

平台默认创建一个默认项目，平台所有成员都具有默认项目的权限。默认项目不支持添加或删除成员，所以默认项目无项目成员管理栏。

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心 > 项目管理 > 成员管理**。

项目管理

添加项目

请输入项目名称

项目名称	项目管理员	描述	项目成员数 (个)	创建时间	操作
默认项目	-	-	-	-	-
设置测试	-	-	3	2022-06-16 11:17:07	成员管理 设置管理员 修改 删除
腾讯问卷	test	-	4	2022-02-11 16:00:30	成员管理 设置管理员 修改 删除
订单测试	-	-	3	2021-11-19 14:26:25	成员管理 设置管理员 修改 删除
事件总线2	-	-	4	2021-11-10 21:50:07	成员管理 设置管理员 修改 删除
事件总线	-	-	3	2021-11-10 10:03:55	成员管理 设置管理员 修改 删除
企业微信服务	-	-	3	2021-07-30 12:24:23	成员管理 设置管理员 修改 删除

共 7 条

10条/页 跳至 1 页

## 2. 添加成员。

只能添加 CAM 中已有子用户的成员。详细请参见 [成员管理](#)。

← 成员管理

请输入成员名称

共3人

10000000000000000000

EIStest

添加成员

**基本信息**

姓名 simeng

联系手机 -

联系邮箱 -

联系微信 -

账号ID 100020126498

创建时间 -

**授权信息**

所有功能权限

请输入应用名称

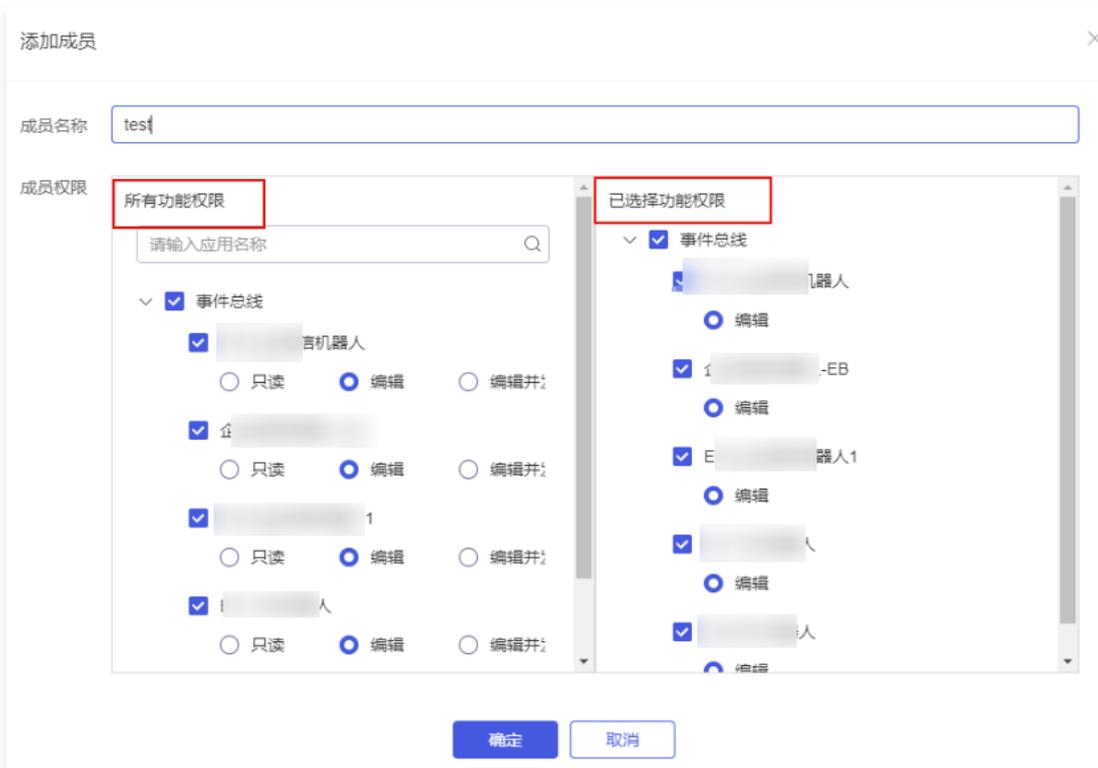
该成员为系统管理员，默认拥有所有权限。

已选择功能权限

该成员为系统管理员，默认拥有所有权限。

## 3. 设置成员权限。

设置成员对应项目内应用，只读、编辑或编辑并发布的权限。

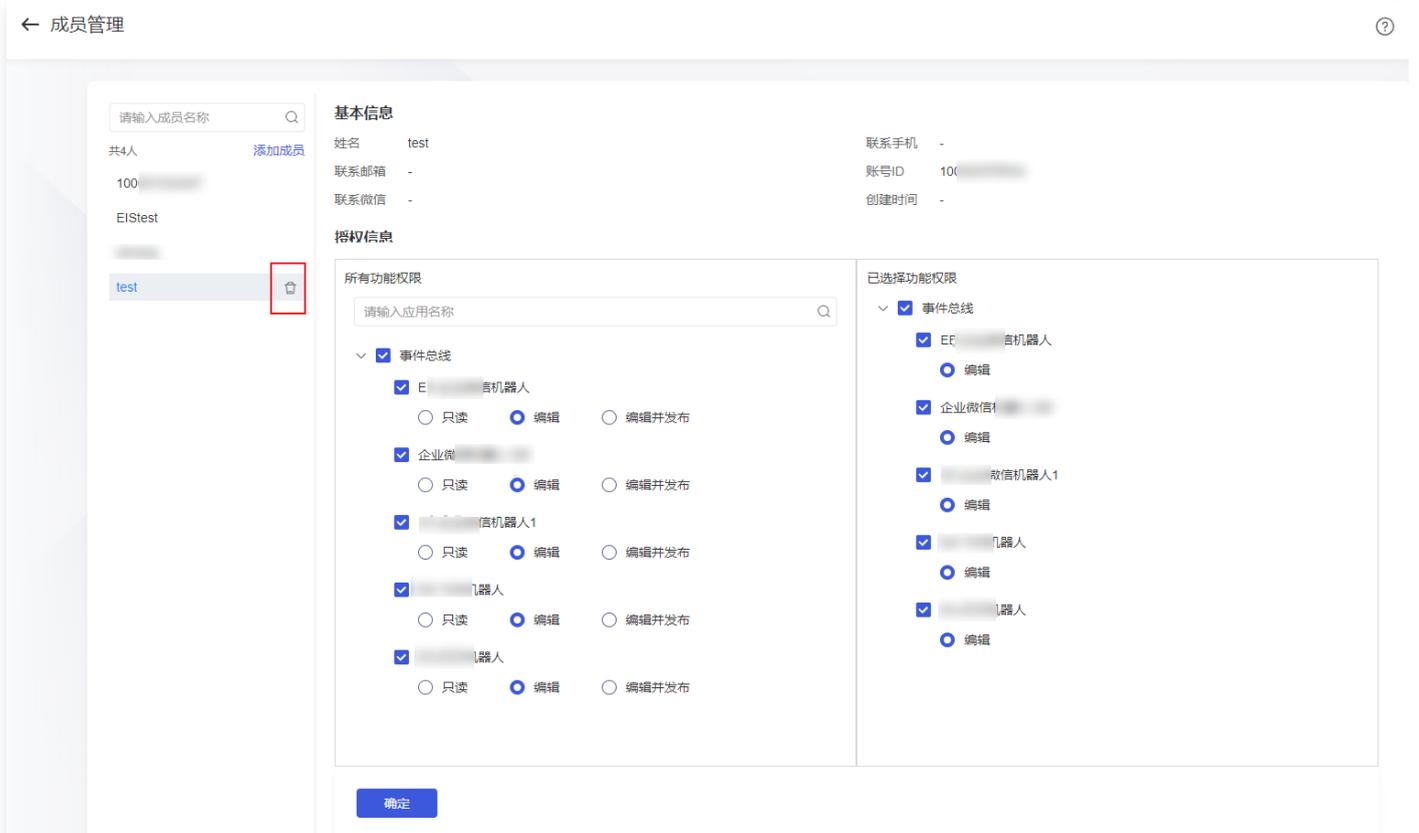


4. 删除成员。

删除成员后此用户不再拥有此项目的相应权限。

**注意：**

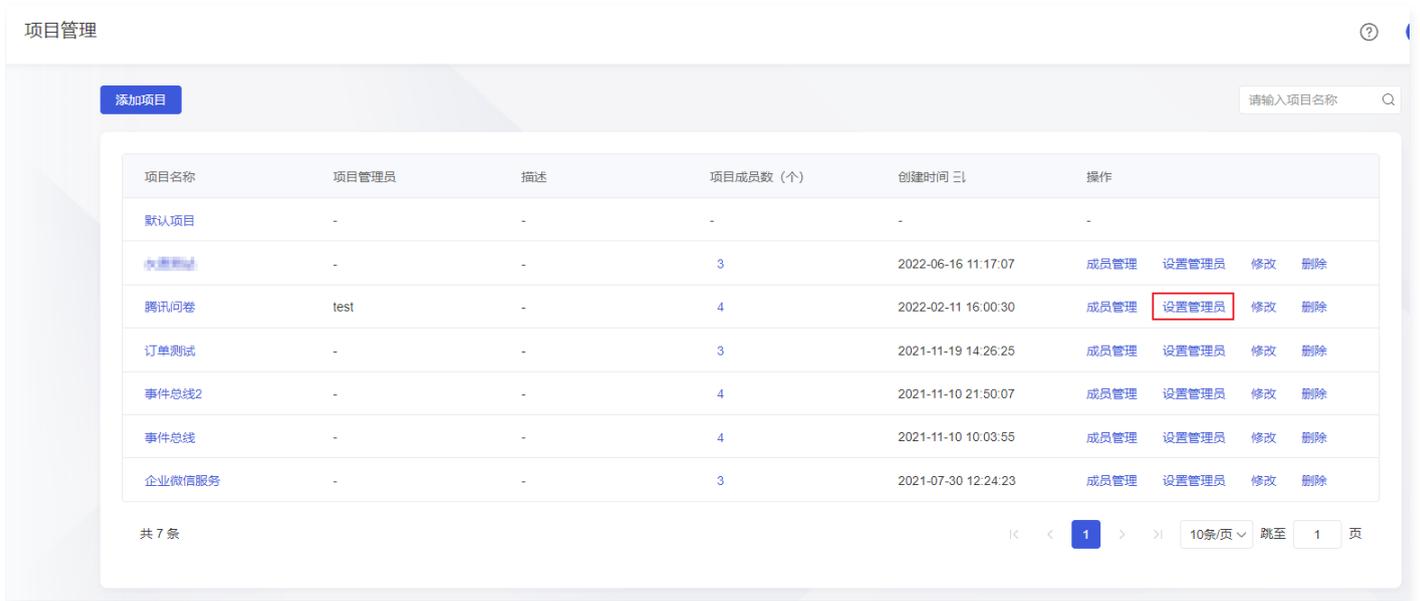
系统管理员无法删除，需要先解除系统管理员角色再进行成员删除操作。



## 设置管理员

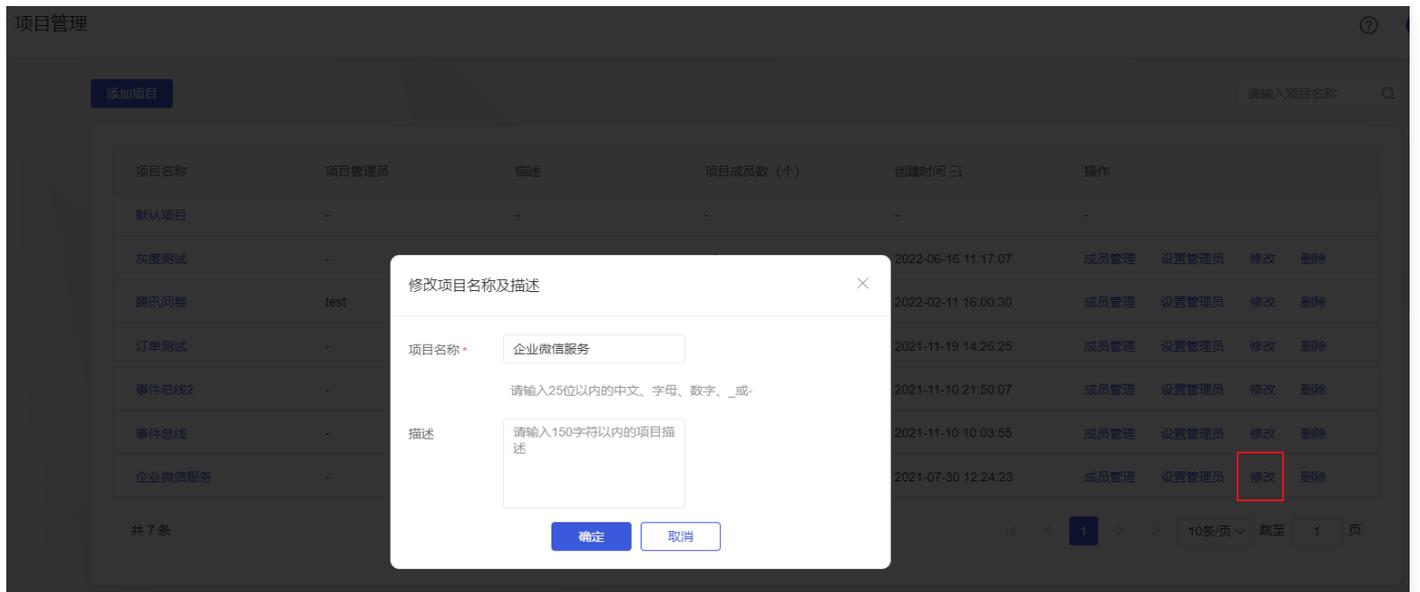
支持为已有项目设置项目管理员，选择项目成员并将其设置为项目管理员后，此成员将拥有对应项目的最高权限，只有项目成员才能设置成为项目管理员。

登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心 > 项目管理 > 设置管理员**。设置成功后，项目管理员处会展示对应成员的名称。



## 修改项目

支持修改项目，可对已存在的项目的项目名称及描述进行修改。



# 环境管理

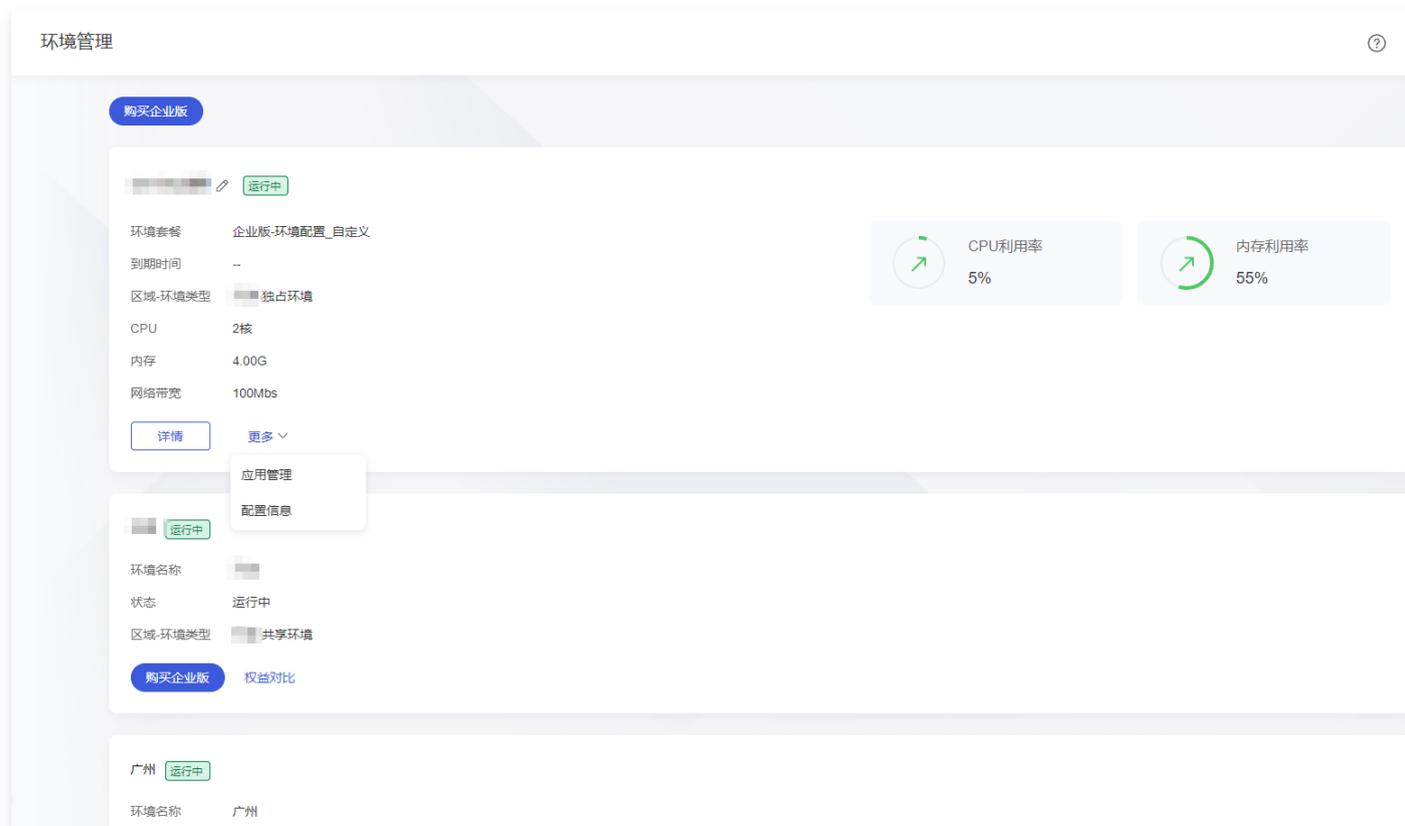
最近更新时间：2024-08-01 15:33:02

## 操作场景

环境管理向用户提供一个便捷的环境和运行时环境查看、管理的路径，您可以一站式的查看部署环境的配置、使用情况以及运维信息。

## 环境管理页面

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心 > 环境管理**。
2. 环境列表页面展示各个环境的基本信息，如下图所示：



- **环境名称**：展示当前用户有权限的所有集群的名称。
- **状态**：展示每个环境的瞬时状态，此处的展示会随着环境状态的变化而变化，目前暂不支持控制台操作创建、删除和扩容等操作，目前平台支持以下几种集群状态（后端执行对应操作，前端展示状态会更新），其中不同状态环境的可用性是有区别的：

环境状态	可用性
创建中	不支持发布
运行中	支持发布
扩展中	支持发布
删除中	不支持发布
不可用	不支持发布
创建失败	不支持发布
已停服	不支持发布

- **区域-环境类型**：展示独立及共享环境的环境列表。当前平台支持两种不同的环境类型，不同的环境类型描述了不同环境的部署特点和资源能力。
  - **共享环境**：所有用户共享的高可用的腾讯轻联的运行环境，可用于快速发布应用。
  - **独立环境**：基于数据和资源的安全性考虑，腾讯轻联也提供资源和数据隔离性更强的部署模式，即独立环境。用户可以独占腾讯轻联的运行环境，达到计算资源和数据的隔离，能够以较低成本，快速提升隔离级别。目前独立环境已开放创建，需要使用该服务可以单击**购买环境**进行购买。
- **环境套餐**：企业版（企业版独立环境才显示此字段）。
- **到期时间**：对应环境使用的到期时间。
- **资源**：CPU、内存、网络带宽的资源只有企业版环境显示购买时或升级后的配置信息，共享环境不展示。
- **资源概览**：展示企业版环境的实际资源使用情况，共享环境不展示。
- **操作**：目前支持应用管理、运行概况及配置信息等操作。

## 环境管理相关操作

### 购买环境

#### 说明：

- 此购买的独立环境仅用于**应用集成**，不可用于**API管理**。
- 购买后会有1-3分钟的创建中状态，请耐心等待。

- **共享环境**适用于新用户测试产品的基本功能，而独立环境（企业版）更适合作为线上业务的部署环境。
- **独立环境**基于数据和资源的安全性考虑，提供了资源和数据隔离性更强的部署模式。用户可以独占腾讯轻联的运行环境，达到计算资源和数据的隔离，能够以较低成本，快速提升隔离级别。适合作为线上业务的部署环境。

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏单击**管理中心** > **环境管理**，单击**购买企业版**。

The screenshot displays the 'Environment Management' (环境管理) interface. At the top, there is a 'Purchase Enterprise Edition' (购买企业版) button. Below it, a card shows details for an 'Enterprise Edition Test' (企业版测试) environment, which is 'Running' (运行中). The configuration includes 'Enterprise Environment Configuration - Custom' (企业版-环境配置\_自定义) as the package, 'Region-Environment Type' (区域-环境类型) as 'Independent Environment' (独占环境), and resources of 2 CPU cores, 4.00G memory, and 100Mbps network bandwidth. To the right, two circular gauges show 'CPU Usage' (CPU利用率) at 5% and 'Memory Usage' (内存利用率) at 61%. A second card below shows a 'Shared Environment' (共享环境) that is also 'Running' (运行中), with details for its name, status, and region.

2. 进入配置选型界面，根据业务的需求选择对应的环境配置即可。若担心过高的配置造成资源浪费，可先选择基础环境配置，后续**变更配置**即可。更多购买相关内容请参见 [购买指南](#)。

**购买企业版环境 权益及资源对比**

购买时长: 1个月 3个月 6个月 1年

环境配置	环境配置A	环境配置B	环境配置C	环境配置D
适用约 2 个应用系统集成	CPU: 4核 安全网关: 4个	内存: 8G 安全网关: 8个	网络带宽: 200Mbps 日志存储: 50GB	可运行流数: 80条
适用约 4 个应用系统集成	CPU: 8核	内存: 16G	网络带宽: 400Mbps	可运行流数: 160条
适用约 8 个应用系统集成	CPU: 16核	内存: 32G	网络带宽: 800Mbps	可运行流数: 320条
适用约 16 个应用系统集成	CPU: 32核	内存: 64G	网络带宽: 1600Mbps	可运行流数: 640条

若不满足需求, 请 [联系我们](#)

续费策略:  自动续费

地域: 广州

协议条款:  我已阅读并同意 [《腾讯云数据连接器服务等级协议》](#)

费用: 元

[立即购买](#) [取消](#)

## 续费和升配

您可以在环境管理中对购买的企业版环境进行续费和升配。

### 升配

**说明:**  
已停服的环境需先续费再升配。

登录到 [环境管理](#)，单击升配。

环境管理

网络带宽: 100Mbps 63%

订单: 运行中

环境名称: 企业版-环境配置 **升配**

到期时间: 2022-08-03 16:41:53 (剩余29天) [续费](#)

区域-环境类型: 广州-独占环境

CPU: 4核

内存: 8.00G

网络带宽: 200Mbps

运行流数: 0/80

CPU利用率: 0%

内存利用率: 0%

单击升配后，跳转至调整配置页面，选择对应要调整的目标配置即可。

**说明:**

升级配置费用：按照使用时长及新配置单价补差价。

购买企业版环境 权益及资源对比

环境配置	环境配置A 适用约 2 个应用系统集成	CPU: 4核 安全网关: 4个	内存: 8G 日志存储: 50GB	网络带宽: 200Mbps	可运行流数: 80条	月
环境配置B 适用约 4 个应用系统集成	CPU: 8核 安全网关: 8个	内存: 16G 日志存储: 50GB	网络带宽: 400Mbps	可运行流数: 160条	月	✓
环境配置C 适用约 8 个应用系统集成	CPU: 16核 安全网关: 16个	内存: 32G 日志存储: 50GB	网络带宽: 800Mbps	可运行流数: 320条	月	
环境配置D 适用约 16 个应用系统集成	CPU: 32核 安全网关: 16个	内存: 64G 日志存储: 50GB	网络带宽: 1600Mbps	可运行流数: 640条	月	

若不满足需求，请联系我们

到期时间 2022-08-03 16:41:53

地域

协议条款

费用

(1) 变配订单金额: [新配置单 时长0.967742\*折扣100% - 旧配置单价 时长0.967742\*折扣100%]

(2) 时长: 0.967742月

元 规则说明

立即购买

取消

### 续费

登录到 [环境管理](#)，单击续费。

环境管理

购买企业版

已停服

环境套餐	企业版-环境配置A 升配	运行流数	0/80	CPU利用率	0%
到期时间	2022-06-13 18:49:51 (已过期) <b>续费</b>	内存利用率	0%		
区域-环境类型	广州-独占环境				
CPU	4核				
内存	8.00G				
网络带宽	200Mbs				

详情 更多

选择要续费的时长即可。

续费
✕

---

当前版本 企业版-环境配置A

描述 广州

原到期时间 2022-06-13 18:49:51

购买时长 1个月 3个月 6个月 1年

协议条款  我已阅读并同意《腾讯云数据连接器服务等级协议》

费用 元

续费
取消

### 查看环境详情

单击对应环境详情，可查看环境详情，包括：概况、应用管理、配置详情（独立环境展示）等信息。

环境管理
?

---

购买企业版

企业版测试 运行中

环境套餐 企业版-环境配置\_自定义

到期时间 --

区域-环境类型 ■ 独占环境

CPU 2核

内存 4.00G

网络带宽 100Mbps

详情 更多 ▼

↑ CPU利用率  
5%

↑ 内存利用率  
61%

---

■ 运行中

环境名称 ■

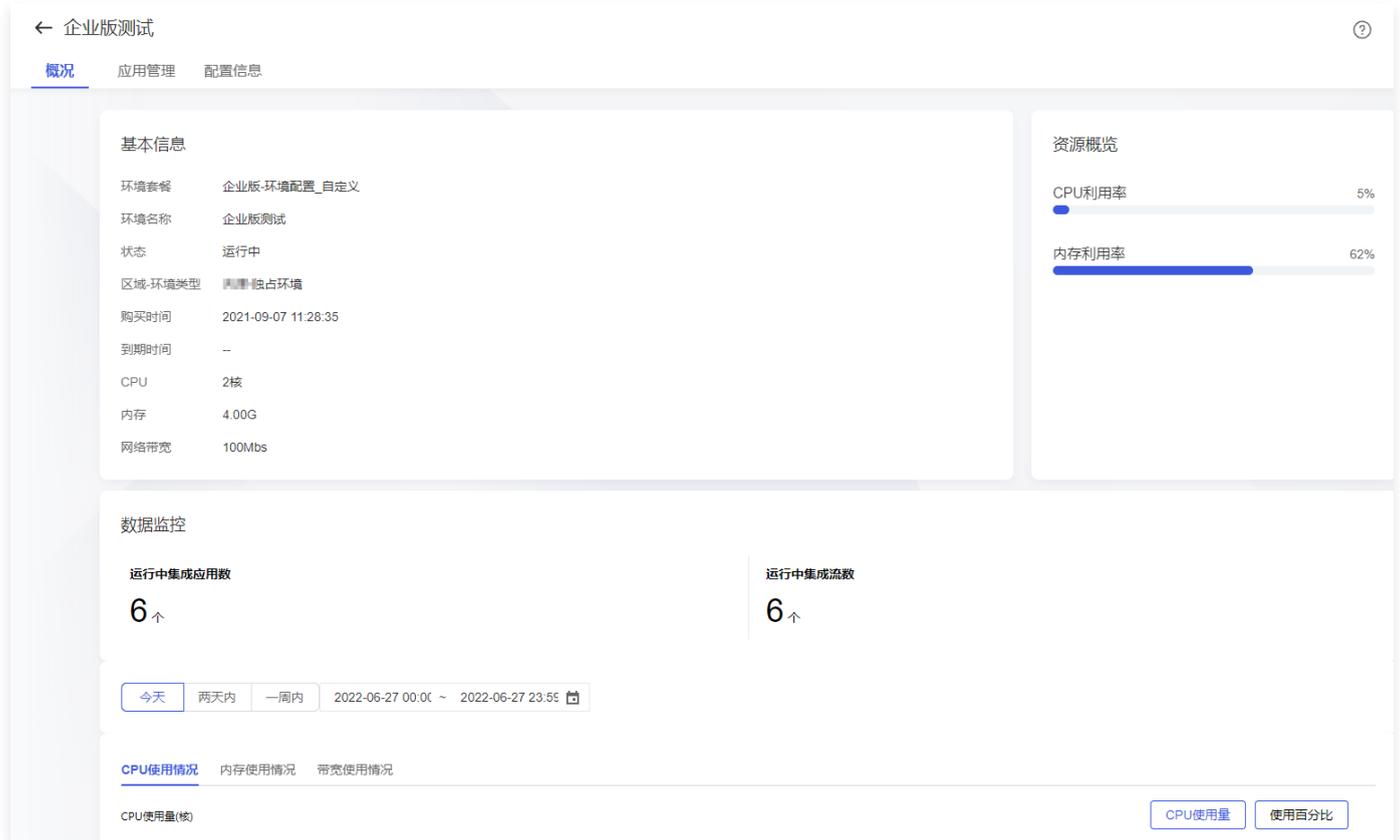
状态 运行中

区域-环境类型 ■ 共享环境

详情 更多 ▼

### 概况

在环境详情页的概况页签，对环境名称、状态、区域-环境类型、运行中集成应用数等相关信息做了集中展示，但共享环境和独立环境所展示的字段略有不同。如下图（红框标注为企业版环境特有展示字段，试用版环境不展示）：



- **共享环境:** 共享环境（试用版）的概况会展示该环境的基本信息、运行区域、运行状态等信息，同时会展示该环境中运行中的集成应用数以及运行中的集成流数。
- **独立环境:** 独立环境（企业版）的概况与测试环境类似，但是除了展示该环境的基本信息、运行区域、运行状态等信息，同时会展示该环境中运行中的集成应用数以及运行中的集成流数之外，还会展示当前环境的使用、运行状况（如CPU内存等的使用情况）。

### 应用管理

在环境详情页的**应用管理**页签，或从**环境管理 > 应用管理**，进入**应用管理**页签。此页面集中展示当前环境下的应用名称、应用状态、项目名称、部署时间、运行/调试版本及操作等内容，方便客户进一步查看和管理指定环境中的相关应用。

← 企业版测试

概况 应用管理 配置信息

应用名称	状态	项目名称	部署时间	运行/测试版本	操作
企微demo	运行中	企微demo	2022-06-24 15:11:04	20220422_16.0	停止
腾讯问卷集成	已停止	默认项目	2022-06-17 11:14:56	20220617_1.0	开始 移除
问卷信息采集_0602_113924	运行中	默认项目	2022-06-14 09:44:05	20220614_3.0	停止
http_demo	已停止	http_demo	2022-05-10 15:24:38	20220510_3.0	开始 移除
http_demo	已停止	http_demo	2022-05-10 15:17:10	20220510_2.0	开始 移除
http_demo	已停止	http_demo	2022-05-10 15:16:14	20220510_1.0	开始 移除
异常集成流测试	运行中	异常demo	2022-04-15 15:44:05	20220310_5.0	停止
企微demo	已停止	企微demo	2022-03-18 10:34:50	20220318_15.0	开始 移除
异常集成流测试	已停止	异常demo	2022-03-10 08:54:31	20220310_4.0	开始 移除
异常集成流测试	已停止	异常demo	2022-03-10 00:22:53	20220310_3.0	开始 移除

共 18 条

10条/页 跳至 1 页

应用管理的相关操作:

- 开始: 对环境内处于已停止状态的应用进行启动操作。
- 停止: 对环境内处于运行中状态的应用进行停止操作。
- 移除: 对环境内处于已停止状态的应用进行移除该环境的操作, 移除后, 将不支持日志查询。

← 企业版测试

概况 应用管理 配置信息

应用名称	状态	项目名称	部署时间	运行/测试版本	操作
企微demo	运行中	企微demo	2022-06-24 15:11:04	20220422_16.0	停止
腾讯问卷集成	已停止	默认项目	2022-06-17 11:14:56	20220617_1.0	开始 移除
问卷信息采集_0602_113924	运行中	默认项目	2022-06-14 09:44:05	20220614_3.0	停止
http_demo	已停止	http_demo	2022-05-10 15:24:38	20220510_3.0	开始 移除
http_demo	已停止	http_demo	2022-05-10 15:17:10	20220510_2.0	开始 移除
http_demo	已停止	http_demo	2022-05-10 15:16:14	20220510_1.0	开始 移除
异常集成流测试	运行中	异常demo	2022-04-15 15:44:05	20220310_5.0	停止
企微demo	已停止	企微demo	2022-03-18 10:34:50	20220318_15.0	开始 移除
异常集成流测试	已停止	异常demo	2022-03-10 08:54:31	20220310_4.0	开始 移除
异常集成流测试	已停止	异常demo	2022-03-10 00:22:53	20220310_3.0	开始 移除

共 18 条

10条/页 跳至 1 页

配置详情

**注意：**

独立环境才展示配置详情页面，共享环境目前是共享环境使用，后台默认配置相关参数不可修改。

在环境详情页的**应用管理**页签，集中展示了环境配置（对应环境下的使用限制）、Dataway 配置、组件配置等参数配置，用户可根据使用场景修改对应上限值，修改后单击**保存**即可，恢复默认值单击**重新配置**。如下图：

企业版测试

概况 应用管理 配置信息

### 环境配置

使用限制

每秒最大请求数 ①	最大并发量 ①	项目每秒最大请求数 ①
300	30	200
项目最大并发量 ①	流每秒最大请求数 ①	流最大并发量 ①
20	200	20

### 执行引擎

最大消息大小 ①	内存最大消息大小 ①	流执行超时时间 ①
10 MB	1 MB	720 分钟

当前配置等量后才能生效

### Dataway配置

超时时间 ①
60000 毫秒

当前配置等量后才能生效

### 组件配置

#### DB

Fetch Size ①
1000

当前配置等量后才能生效

保存 重新配置

**环境配置：****使用限制：**

- 每秒最大请求数：每秒最多可触发请求多少次。
- 项目每秒最大请求数：单个项目每秒最多可触发请求多少次。
- 流每秒最大请求数：单个流每秒最多可触发请求多少次。
- 最大并发量：系统可支持的最大并发量。
- 项目最大并发量：项目最大并发量。
- 流最大并发量：单个流的最大并发量。

**执行引擎：**

- 最大消息大小：Message 在内存中的大小限制。
- 流执行超时时间：流执行超时时间。
- 内存最大消息大小：超过此大小的消息将保存至 COS 进行处理。

- DataWay配置：
  - 超时时间：DataWay 超时时间，配置后需重启对应集成应用才能生效。
- 组件配置：
  - DB：
    - Fetch Size：每次最多从数据库查询多少条数据，配置后需重启对应集成应用才能生效。

# 操作审计

最近更新时间：2024-08-01 15:33:02

## 操作场景

审计日志支持查看一段时间内的历史操作事件，包括基本信息、操作信息和事件详情，通常可用于合规审计、项目变更溯源和安全分析等使用场景。

## 操作步骤

### 步骤1：查询审计日志

控制台默认展示近1天的审计日志，用户可设置筛选项来查看对应审计日志。

#### 说明：

目前控制台支持检索近一个月的审计日志。若有检索更早期日志的需求，请 [提交工单](#) 联系我们。

### 审计日志

近3天



用户名称  项目名称  操作模块

操作类型  操作结果  操作事件

搜索

重置

日志数量 1702条

设置列

操作时间	用户名称(ID)	操作结果	项目名称	操作模块	操作类型	操作事件	操作
2022-12-20 18:10:56	iPaaS测试专用 (7( ))	失败		应用集成	发布	对应用(wenjuan)进行应用调试	<a href="#">查看详情</a>
2022-12-20 18:08:57	iPaaS测试专用 (7( ))	成功		应用集成	编辑	对应用(test)中集成流(NewFlow)进行编辑	<a href="#">查看详情</a>
2022-12-20 18:00:24	iPaaS测试专用 (7( ))	成功		安全网关	编辑	编辑安全网关{内网}	<a href="#">查看详情</a>
2022-12-20 17:58:39	iPaaS测试专用 (7( ))	成功		安全网关	编辑	禁用安全网关{内网}	<a href="#">查看详情</a>

1. 登录 [腾讯轻联控制台](#)，在左侧导航栏中选择**管理中心 > 审计日志**。

2. 设置日志筛选项。筛选项包括：

字段	说明
操作时间	支持按操作时间进行搜索，预定义近5分钟、近15分钟、近1小时、近6小时、近1天、近3天、近7天、近30天、今天、昨天、前天、本周、上周，以及自定义时间区间的设置。
用户名称	支持按用户名称以及账号 ID 进行搜索，可用于查看指定用户的审计日志。

项目名称	支持按项目名称进行搜索，可用于查看属于指定项目的审计日志。
操作模块	支持按操作模块进行搜索，包括集成应用、连接管理、安全网关、运行监控、通用存储、集成资源、项目管理、API管理、API 用户中心等功能模块。
操作类型	支持按操作类型进行搜索，包括新建、编辑、删除、发布、停止、登录、注销。
操作结果	支持按操作结果进行搜索，包括成功、失败。
自定义搜索框	支持按用户输入的操作事件关键词进行搜索，通常用于异常追溯场景。

3. 查看日志。显示对应筛选条件的审计日志，按就近原则进行排序。日志列表内容包括：

字段	说明
操作时间	操作事件发生的时间。
用户名称	操作事件对应的执行者用户名称以及账号ID。
操作结果	操作事件是否执行完成。若失败，则代表该操作未执行完成，支持查看错误信息。
项目名称	操作事件所属的项目。
操作模块	操作事件对应的功能模块。
操作类型	操作事件对应的抽象操作类型。
操作事件	操作事件的内容摘要。
操作	查看详情。支持查看基本信息、操作信息和事件详情（见 <a href="#">步骤2</a> ）。

近3天
🔍
↓

用户名称 全部用户

操作类型 新建, 编辑, 删除, 发布, 停止, ...

项目名称 全部项目

操作结果 全部操作结果

操作模块 应用集成, 连接管理, API管理, 安全网关, 通用存储, ...

操作事件 请输入操作事件关键词

搜索

重置

日志数量 10条 ⚙️ 设置列

操作时间	用户名称(ID)	操作结果	项目名称	操作模块	操作类型	操作事件	操作
2023-01-05 12:05:14	...	成功	默认项目	应用集成	编辑	对应用{HelloW3orld}新增集成流{HelloWorld_Flow}	<a href="#">查看详情</a>
2023-01-05 12:05:13	...	成功	默认项目	应用集成	新建	添加应用{HelloW3orld}	<a href="#">查看详情</a>
2023-01-04 16:40:42	...	成功	默认项目	API管理	编辑	对API服务{服务4}编辑API_Endpint{6565}	<a href="#">查看详情</a>
2023-01-04 11:49:08	...	成功	默认项目	API管理	编辑	对API服务{服务4}添加API_Endpint{6565}	<a href="#">查看详情</a>
2023-01-04 11:46:59	...	成功	默认项目	API管理	编辑	对API服务{服务4}添加API_Endpint{4343}	<a href="#">查看详情</a>

4. 重置日志筛选项。

当前筛选查询完毕后，可通过单击重置快速恢复默认展示。

近3天
🔍 ⬇

用户名称

操作类型

项目名称

操作结果

操作模块

操作事件

搜索 重置

日志数量 10条 🔍 设置列

操作时间	用户名称(ID)	操作结果	项目名称	操作模块	操作类型	操作事件	操作
2023-01-05 12:05:14	👤	成功	默认项目	应用集成	编辑	对应用{HelloW3orld}新增集成流{HelloWorld_Flow}	<a href="#">查看详情</a>
2023-01-05 12:05:13	👤	成功	默认项目	应用集成	新建	添加应用{HelloW3orld}	<a href="#">查看详情</a>
2023-01-04 16:40:42	👤	成功	默认项目	API管理	编辑	对API服务{服务4}编辑API_Endpoint{6565}	<a href="#">查看详情</a>

## 步骤2：查看日志详情

针对关心或存疑的审计日志，用户可单击对应审计日志的**查看详情**以查看详细信息。除列表页面信息外，还额外展示以下信息：

近3天
🔍 ⬇

用户名称

操作类型

项目名称

操作结果

操作模块

操作事件

日志数量 10条

操作时间	用户名称(ID)	操作结果	项目名称	操作模块	操作类型	操作事件
2023-01-05 12:05:14	👤	成功	默认项目	应用集成	编辑	对应用{HelloW3orld}新增集成流{HelloWorld_Flow}
2023-01-05 12:05:13	👤	成功	默认项目	应用集成	新建	添加应用{HelloW3ork}
2023-01-04 16:40:42	👤	成功	默认项目	API管理	编辑	对API服务{服务4}编辑
2023-01-04 11:49:08	👤	成功	默认项目	API管理	编辑	对API服务{服务4}添加
2023-01-04 11:46:59	👤	成功	默认项目	API管理	编辑	对API服务{服务4}添加

日志详情 ✕

**基本信息**

操作时间 2023-01-05 12:05:14

事件ID d849d2827e4e404b244...

用户名称 👤

账号ID 👤

源IP地址 8.229.88.11

**操作信息**

项目名称 默认项目

操作模块 应用集成

操作类型 编辑

操作结果 成功

操作事件 对应用{HelloW3ork}新增集成流{HelloWorld\_Flow}

**事件详情**

事件详情

```
{
  "应用名称": "HelloW3orld",
  "应用版本": "20230105_1.0",
  "集成流名称": "HelloWorld_Flow"
}
```

类型	字段	说明
基本信息	事件 ID	每条审计日志的唯一标识。
	源 IP 地址	操作事件对应用户的源 IP 地址。
操作信息	错误信息	当操作结果为失败时，会显示对应错误信息
时间详情	时间详情	显示当前操作事件对应的请求参数。例如：操作事件为“新建 API 服务{我是 API 服务名称}”，则此处会记录 API 服务名称、协议、描述、标签、版本等信息。

## 步骤3：下载日志

平台提供下载日志的功能。支持将当前列表的审计日志导出为 .xlsx 或 .csv 文件，以便后续进一步分析或整理。详细字段请参见 [日志列表](#)。  
 登录 [腾讯轻联控制台](#)，在左侧导航栏中选择审计日志，单击下载日志按钮后选择导出为 .xlsx 或 .csv 文件即可。

近3天
🔍
↓

用户名称 全部用户

操作类型 全部操作类型

项目名称 全部项目

操作结果 全部操作结果

操作模块 全部操作模块

操作事件

导出为csv文件

导出为xlsx文件

搜索 重置

日志数量 10条 🔍 设置列

操作时间	用户名称(ID)	操作结果	项目名称	操作模块	操作类型	操作事件	操作
2023-01-05 12:05:14	🔍	成功	默认项目	应用集成	编辑	对应用(HelloW3orld)新增集成流 {HelloWorld_Flow}	<a href="#">查看详情</a>
2023-01-05 12:05:13	🔍	成功	默认项目	应用集成	新建	添加应用(HelloW3orld)	<a href="#">查看详情</a>

# Dataway 表达式简介

最近更新时间：2024-08-01 15:33:02

Dataway 是腾讯轻联中用于对流运行数据进行自定义转换与处理的脚本引擎，集成在腾讯轻联服务中，是提供腾讯轻联可扩展性的关键能力。腾讯轻联的许多内置组件和连接器中都提供了基于 Dataway 脚本的自定义能力，可以用于对轻联事件进行动态处理。例如：

- 在配置变量组件中，可以通过 Dataway 脚本来动态地设置变量的值。
- 在数据转换组件中，可以充分利用 DataWay 的灵活语法进行复杂的数据处理与运算，以最终生成期望的产出结果，用于下游组件的处理。

您可以在轻联内置组件配置中直接使用 Dataway 表达式。

- 如果您需要直接开始使用 Dataway 表达式，请参见 [Dataway 快速入门](#)。
- 如果已经有轻联开发经验，要了解更多关于 Dataway 特性，请参见 [Dataway 语言基础概念](#)：
  - [文本](#) 模式流转数据。
  - [表达式](#) 模式处理简单数据。
  - [代码模式 Python](#) / [代码模式 Java](#) 处理复杂数据。
- 如果您想在场景案例中体验 Dataway，请参见 [场景案例](#)。
- 如果您有其他问题，请参见 [Dataway 常见问题](#)。

# 快速入门

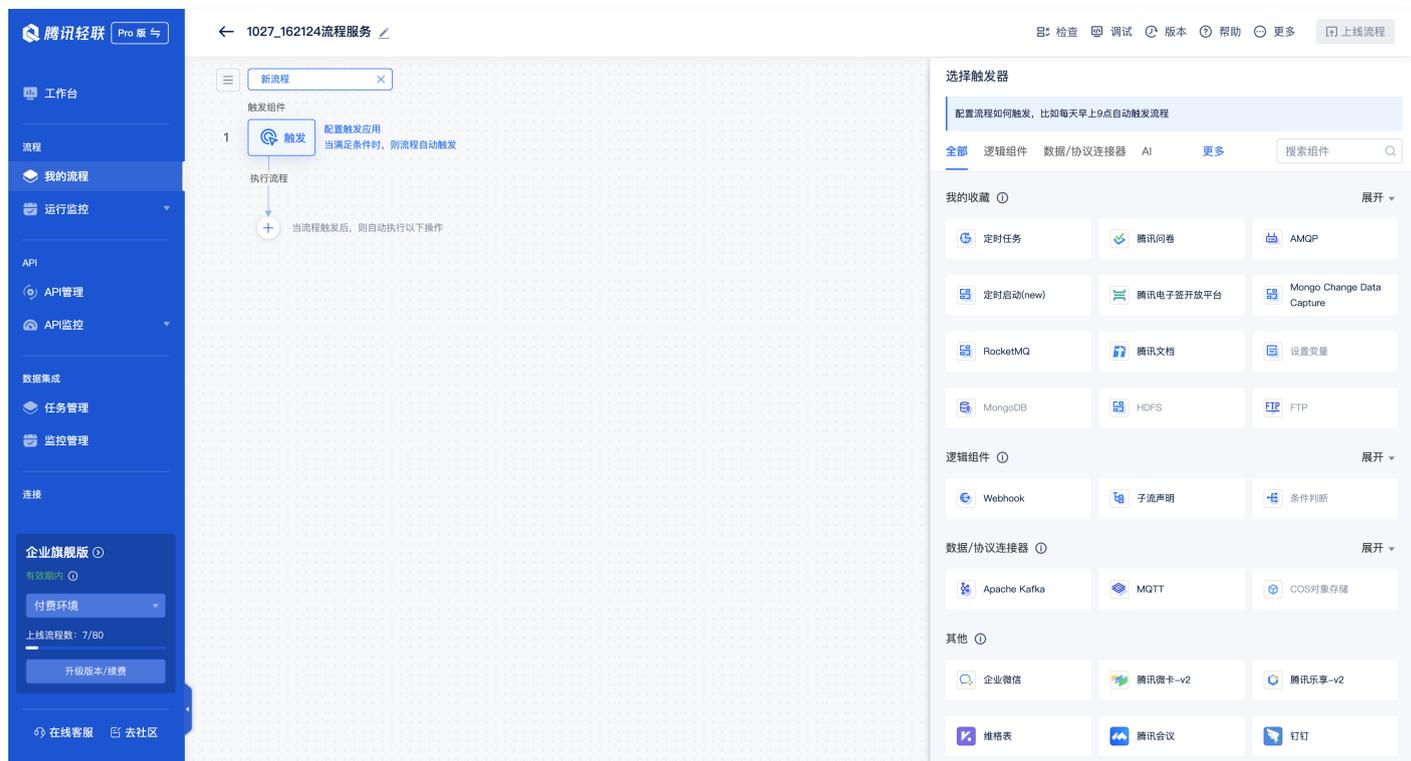
最近更新时间：2024-06-04 15:06:51

## 操作场景

本文介绍如何使用 Dataway 脚本来辅助进行集成流设计。

## 前期准备

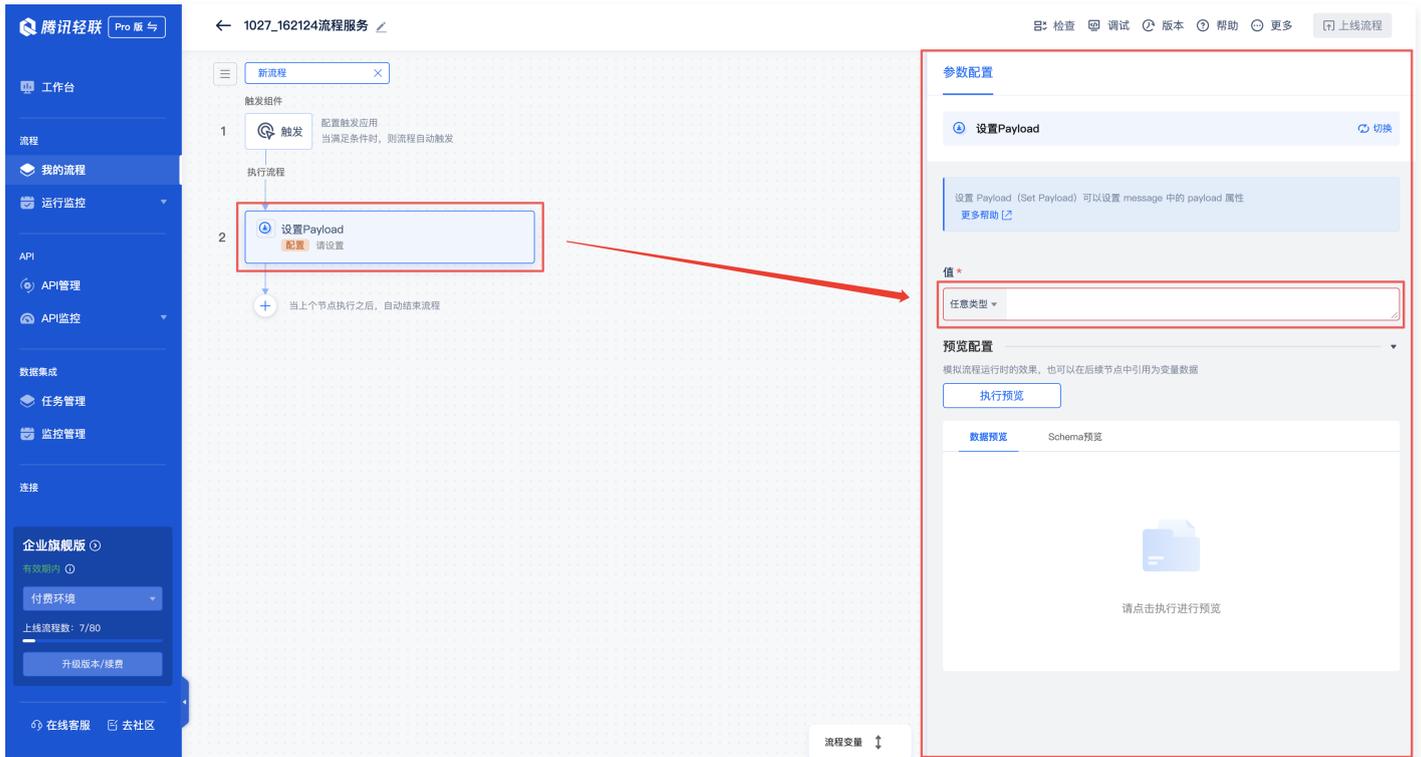
1. 已有腾讯云账号可直接登录 [腾讯轻联控制台](#)，若无账号请先注册账号。
2. 登录成功后，新建一个应用并创建一条集成流。



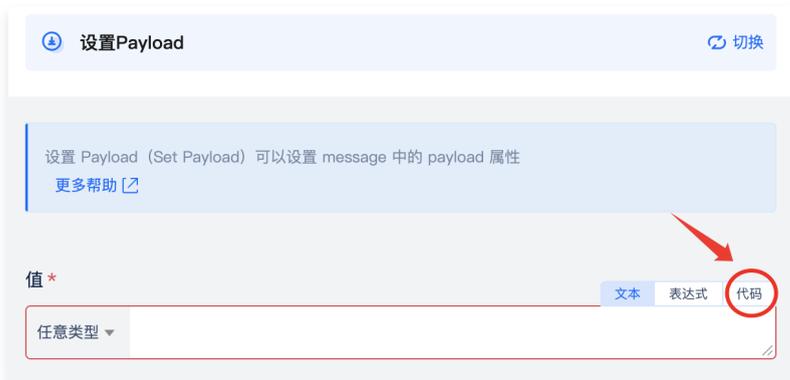
## 使用 Dataway 表达式（以代码模式 Python 脚本为例）

以一个简单的字符串连接为示例，使用步骤如下：

1. 在腾讯轻联的 [应用集成](#) 页面，单击新建应用，新建一个配置 Payload 组件。
2. 在右侧自动弹出组件配置。其中，值配置项需要 Dataway 表达式填写。



3. 鼠标移至值配置项的编辑文本框，弹出模式选择按钮，单击代码，进入代码输入模式。



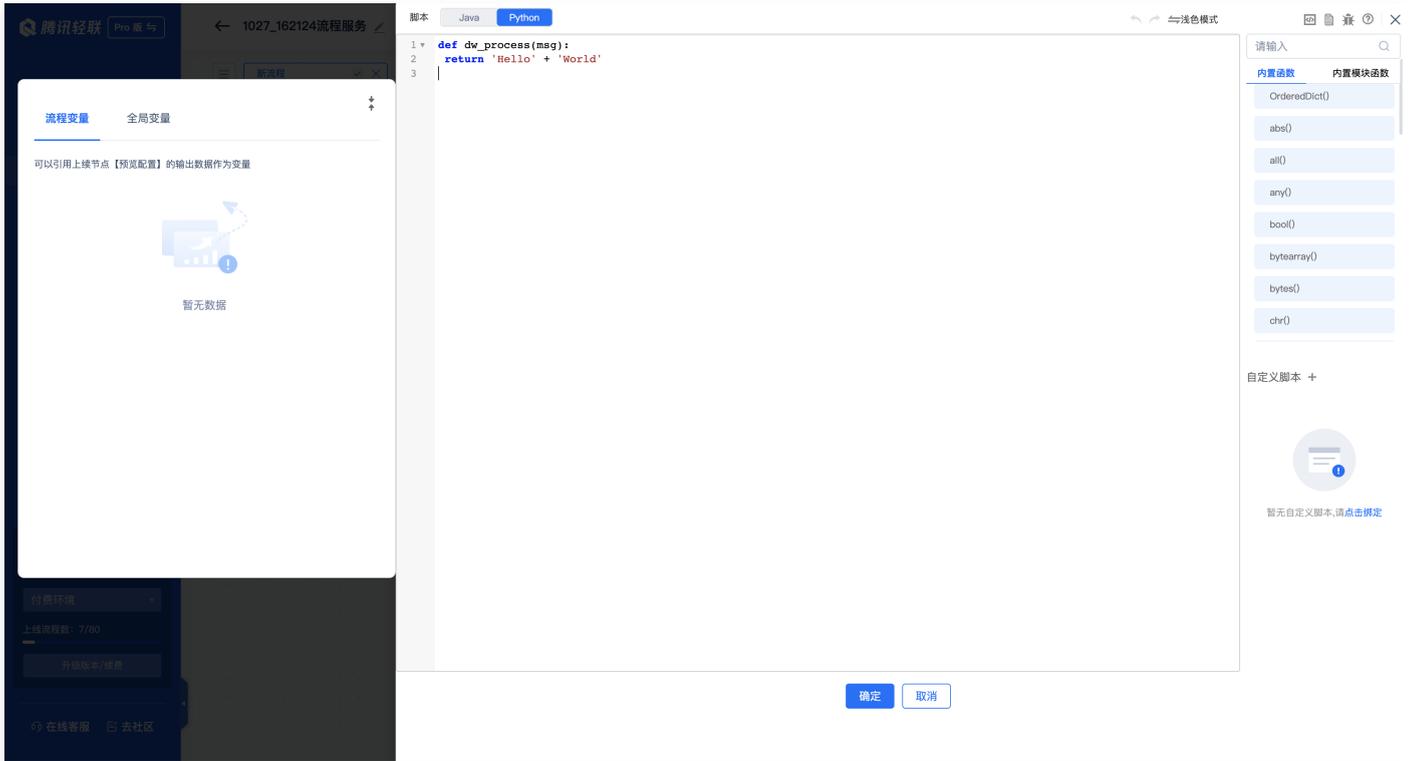
4. 单击编辑文本框，弹出代码编辑器，输入 Dataway 脚本。输入时会实时进行语法检查，若出错则会有对应提示。

```
def dw_process(msg):
    return 'Hello' + 'World'
```

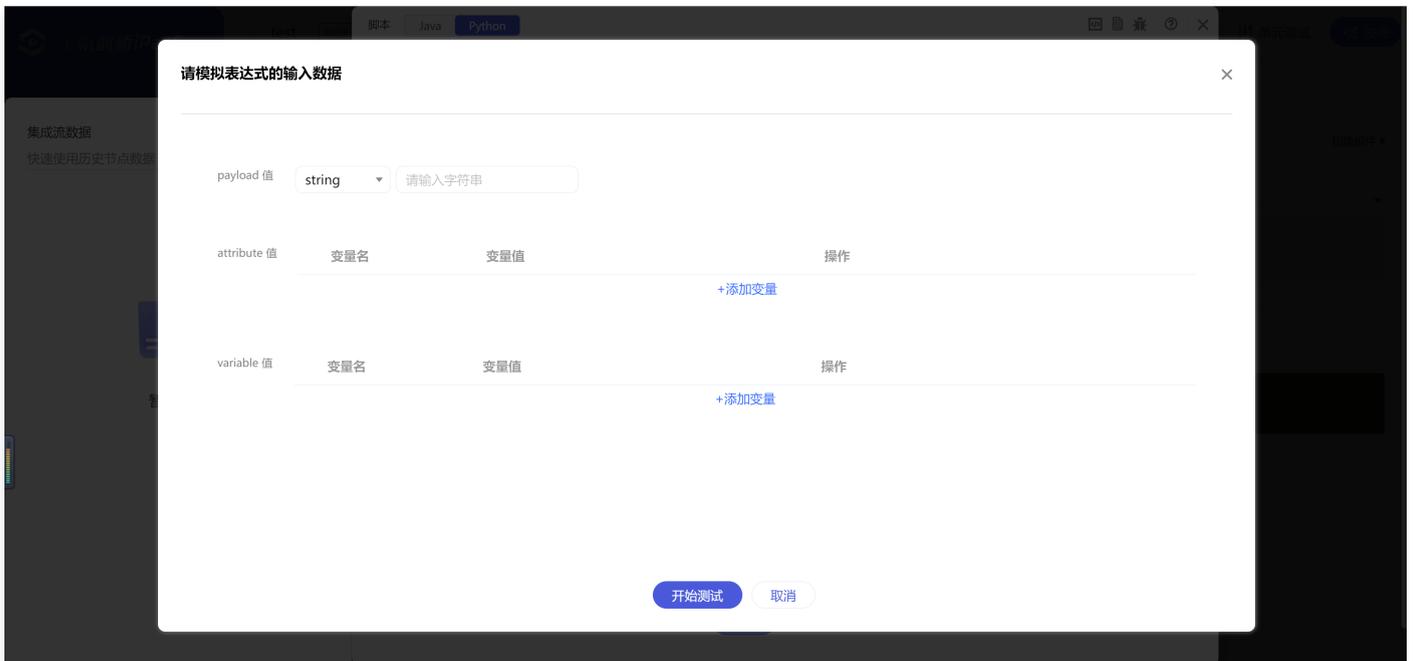
- 完整的 DataWay 代码模式下的 Python 脚本需符合语法定义的 Python3 代码段，其中包含入口函数定义 def dw\_process(msg)。
- DataWay 基于 Python3 语法进行实现，同时内置了多个第三方模块，如 time、json、math等，使用时直接引用模块名即可。

**注意：**

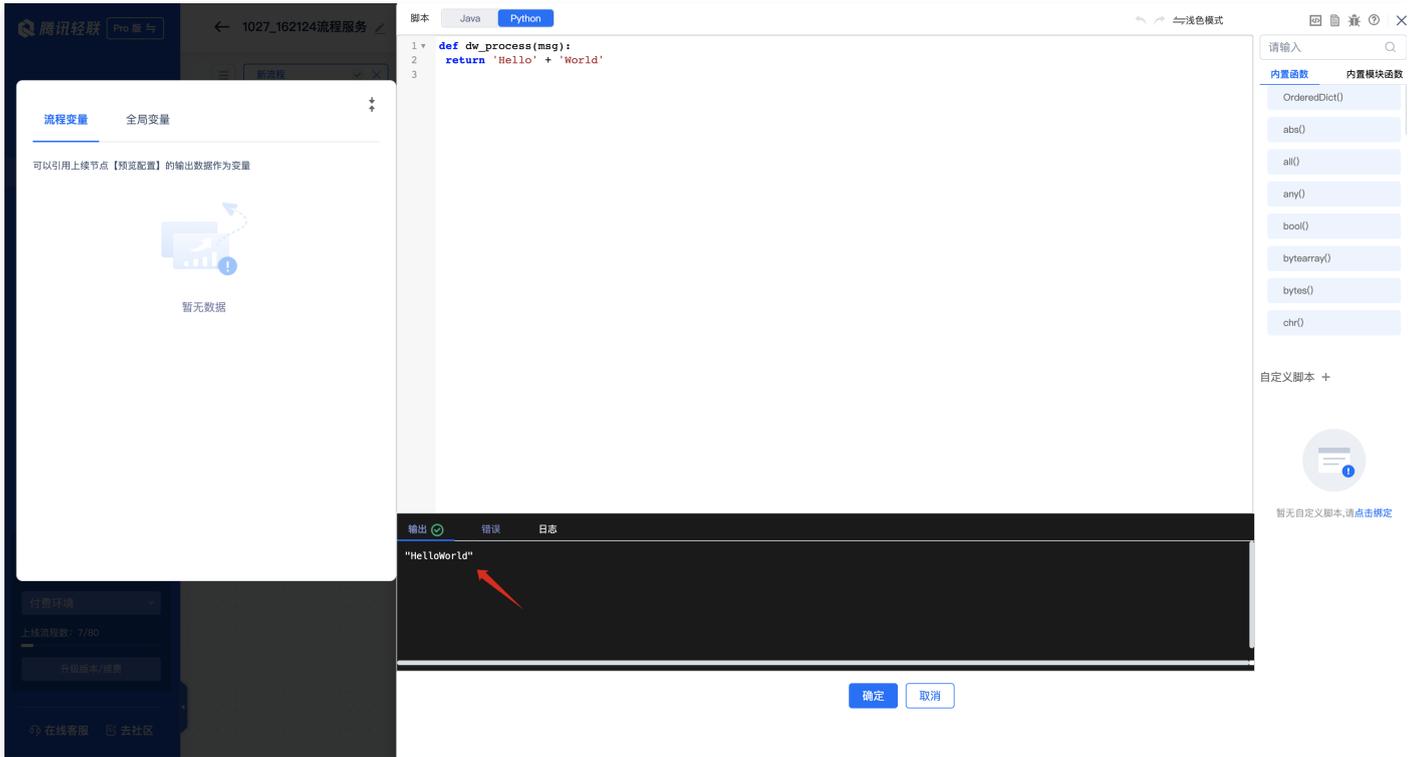
只有轻联的 Pro 版本，才支持使用 Python 和 Java。



5. 验证 Dataway 运行结果：在通过语法检查并单击确定保存表达式之前，可以对 Dataway 脚本的正确性进行验证。在编辑框右上角单击 **Debug**，在弹出的对话框中单击**开始测试**。



测试结束后，Dataway 代码编辑框的下方会有输出结果的展示，可以看到 Dataway 脚本的运行结果为 `HelloWorld`，符合预期。



6. 单击**确定**，完成 Dataway 脚本的保存。

### 表达式模式

对于简单表达式输入，用户可以使用表达式模式。

1. 当鼠标移至**值配置项**的**编辑文本框**，弹出模式选择按钮时，单击**表达式**，进入表达式模式。
2. 单击编辑文本框，即可填写 Dataway 表达式。

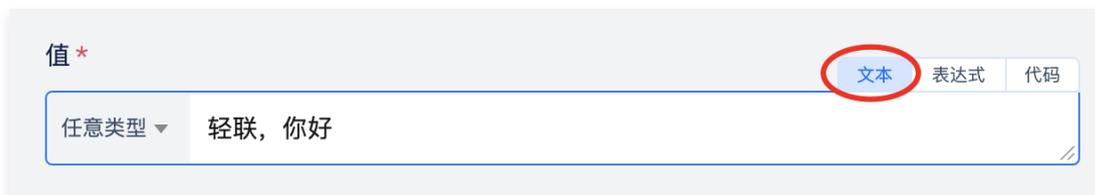


### 文本模式

对于字面量数据的创建或 [集成流数据引用](#) 等简单输入，用户可以使用文本模式。

以生成时间数据为示例，使用步骤如下：

1. 当鼠标移至**值配置项**的**编辑文本框**，弹出模式选择按钮时，单击**文本**，进入文本模式。

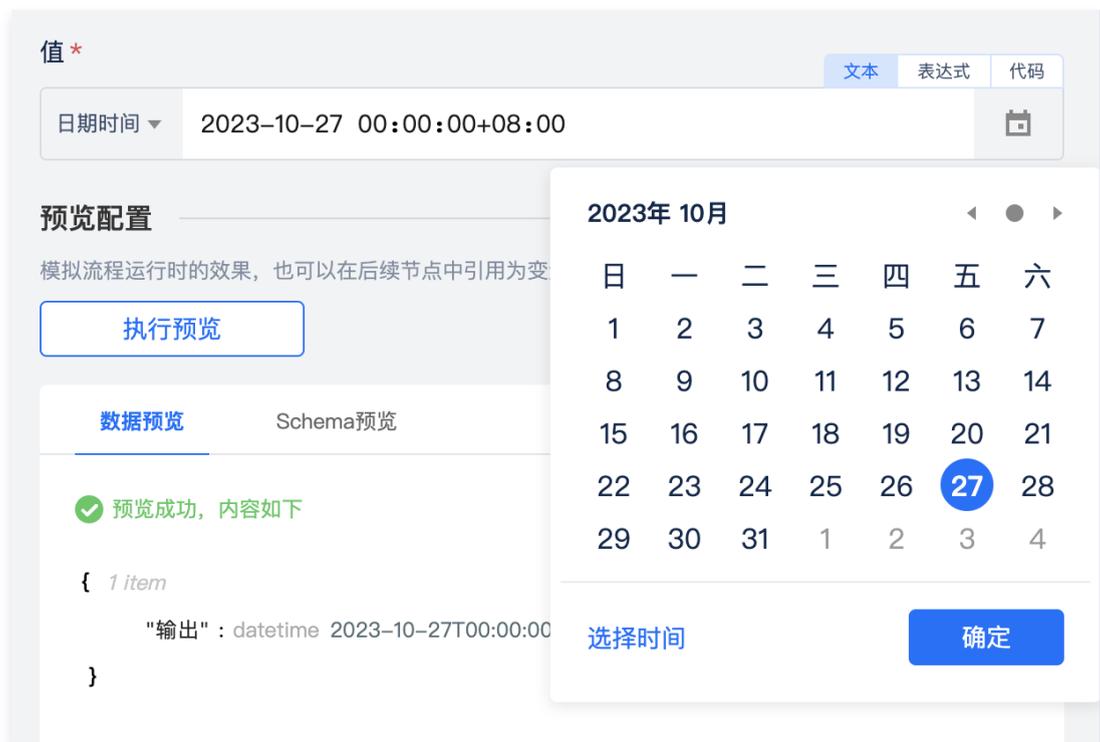


2. 单击左侧类型选择下拉菜单，菜单展开后，找到并单击 **datetime**。



3. 单击编辑文本框，弹出时间设定交互界面，在此界面上设定时间信息。

4. 设定完成后，单击**确定**完成输入。



## 代码模式 Java 脚本

除了 Python 语法，Dataway 提供对 Java 语法的支持，用户可以使用代码模式输入 Java 脚本。

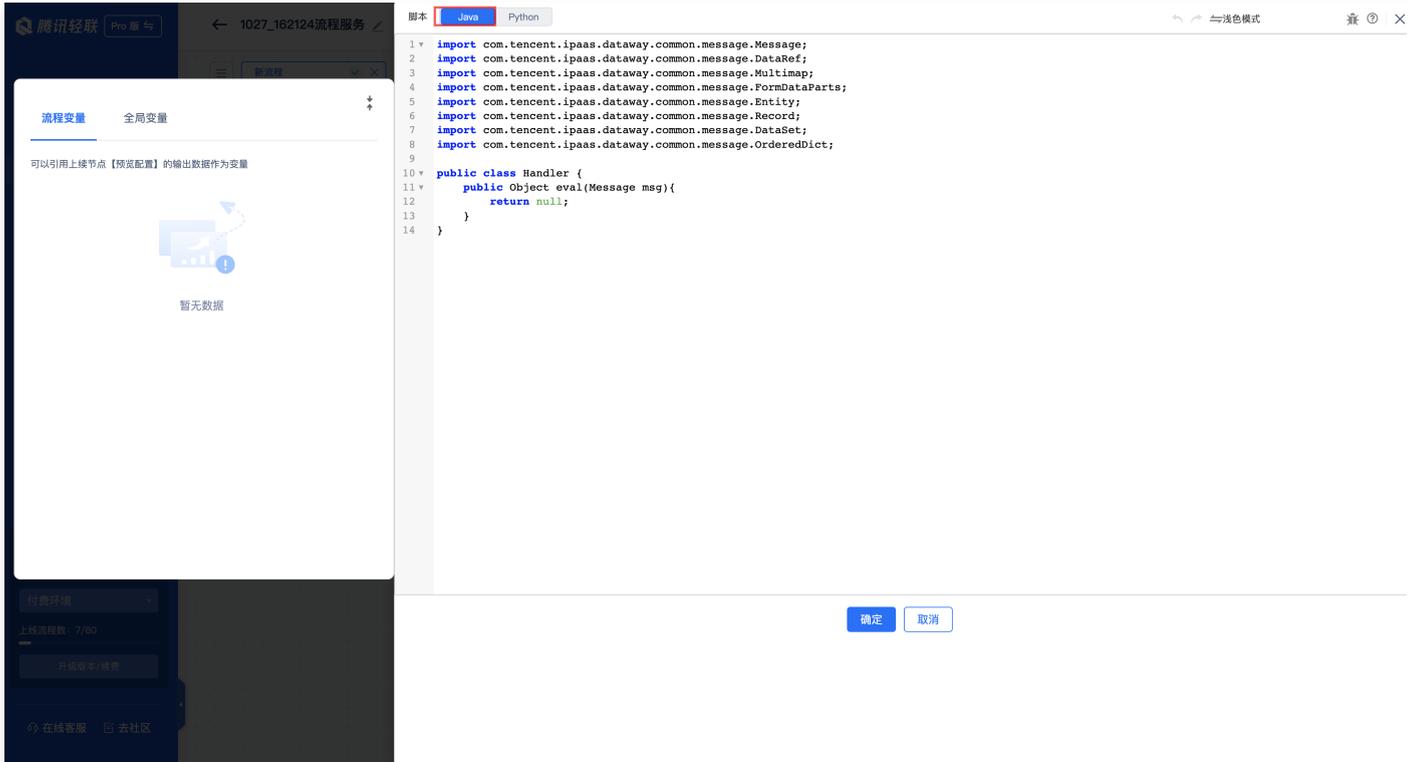
1. 鼠标移至值配置项的编辑文本框，弹出模式选择按钮，单击**代码**，进入代码输入模式。



2. 单击编辑文本框，进入代码编辑交互界面，然后单击 **Java**，开始 Java 脚本编辑。

**注意：**

只有轻联的 Pro 版本，才支持使用 Python 和 Java。



3. 单击确定，完成 Dataway 脚本的保存。

## 集成流数据面板和引用

Dataway 支持可视化引用集成流的上下文数据，打通组件间的数据流转捷径，提升用户体验。所有模式均支持集成流数据面板的数据引用功能，包括 **文本模式**、**表达式模式**、**代码模式 Python** 和 **代码模式 Java**。

编辑 Dataway 输入文本框时自动弹出**集成流数据面板**。单击面板中的数据按钮，即可引用相应数据，并以数据标签的形式显示在文本框中。

新流程

触发组件

1 触发 配置触发应用 当满足条件时, 则流程自动触发

执行流程

2 设置Payload 配置 2023-10-27T00:00:00+08:00

3 企业微信机器人 - 发送文本消息

当上个节点执行之后, 自动结束流程

流程变量 全局变量

可以引用上续节点【预览配置】的输出数据作为变量

2. 设置Payload

输出 2023-10-27T00:00:00+08:00

企业微信机器人 - 发送文本消息

切换

文本内容 文本 表达式 代码

节点2 payload

提醒人员

文本

可以输入@all提醒全部成员, 也可以输入成员ID提醒部分成员, 多个成员ID之间用";"号隔开 成员ID可从企业微信管理后台--通讯录--成员详情中获取

预览配置

模拟流程运行时的效果, 也可以在后续节点中引用为变量数据

执行预览

数据预览 普通模式 专家模式

请点击执行进行预览

## 开发指南

# Dataway 语言基础概念

最近更新时间：2025-07-03 17:43:01

Dataway 是腾讯轻联中用于对数据进行自定义转换与处理的脚本引擎，使用 Dataway 可以编写和执行强大而复杂的数据转换脚本，以下简要介绍 Dataway 的核心概念和功能。

## Dataway 工具箱

Dataway 提供文本模式、表达式模式和代码模式三套脚本工具集，支持各具特色的语义和语法，以应对不同的使用场景和数据需求。Dataway 编辑文本框同时支持三套工具集（部分组件因功能设计禁用了特定工具集），用户可根据应用需求和自身喜好，自由选用其中一套工具集输入脚本。

模式	作用	说明
文本模式	用于数据流转，提供数据创建和传递功能。	用户跟随可视化交互界面的指引生成所需数据或引用集成流上下文数据。
表达式模式	用于简单数据转换和处理，提供轻量脚本执行功能。	用户通过填写表达式获取所需数据。
代码模式	用于复杂数据转换和处理，提供复杂脚本执行、格式化和调试等功能。	用户通过编写完整脚本获取所需数据。目前支持 Python3 脚本和 Java JDK8 脚本。

集成流数据面板同时嵌入三种模式，提供集成流上下文数据引用功能。用户通过可视化点选的方式，快速引用前置组件的数据。

## Dataway 类型系统

Dataway 核心类型如下，可作为 Dataway 脚本的输出结果在组件间传递：

类型名	中文名	描述	是否 Dataway 特有类型	举例
None	空值	空值类型	否	Python: None / Java: null
string	字符串	字符串类型	否	"abc"
bytes	字节数组	字节数组类型	否	Python: b"abc" / Java: "abc".getBytes()
bool	布尔值	布尔类型	否	Python: True/False / Java: true/false
float	浮点数	浮点类型	否	123.456
int/Long	整数	整型	否	Python: 123 / Java: 123L
list	列表	序列类型容器	否	Python: [1,2,3] / Java: new java.util.ArrayList<>()
dict/Map	字典	键值类型容器	否	Python: {1:1, 'key': 'value'} / Java: new java.util.HashMap<>()
decimal	十进制	用于精确十进制数值计算	否	Python: decimal.Decimal(1) / Java: new java.math.BigDecimal("1")

datetime	时刻	时刻, 包括日期和时钟	否	Python: datetime.datetime.now() / Java: java.time.OffsetDateTime.now()
date	日期	日期	否	Python: datetime.date.today() / Java: java.time.LocalDate.now()
time	时间	时间	否	Python: datetime.datetime.now().time() / Java: java.time.OffsetTime.now()
Entity	二进制实体	腾讯轻联中的实体数据, 用于代表一个二进制对象, 包括 blob、mime_type、encoding 等信息	是	http-listener 构造消息中的 payload
MultiMap	多值字典	类似于 xml 而与 dict 不同, 该类型可以支持重复的 key	是	application/www-form-urlencoded 格式的数据解析之后得到的对象
FormDataParts	表单数据	数组+列表的数据结构, 类似于 Python 中的 orderDict 结构	是	multipart/form-data 格式的数据解析后得到的对象
Message	消息	腾讯轻联中的消息, 承载了集成流数据, 包括 payload、variables、attributes 等信息	是	代码模式 Python 脚本 dw_process 入口函数中的 msg 参数
DataSet	数据集	腾讯轻联数据集成中数据集, 通过数据集集成组件操作	是	Builder 组件的输出
Record	单条数据	腾讯轻联数据集成中的单条数据, 附有 Schema	是	可通过 Foreach 组件遍历 DataSet 获取

Dataway 核心类型在三种脚本工具集中是通用的, 均存在对应数据结构。虽然, 在三种脚本工具集中同一类型的不同数据结构操作方法存在差异, 但是, 保证同一类型的不同数据结构核心特性一致, 可以相互间无损转换。

当集成流的上下游使用不同的脚本工具集时, 同一类型的不同数据结构之间自动无感映射。用户可以在各个 Dataway 编辑文本框中使用不同的 Dataway 脚本工具集, 不会影响核心类型的数据处理过程的连贯性和精确性。

**注意:**

如果 Dataway 脚本的输出结果作为集成流的最终返回结果, 则返回值支持类型还会受到相应集成流组件的限制。如在以 http listener 组件作为触发器的集成流中, 其最终返回结果必须为 Entity 类型。

除了核心类型外, 各脚本工具集额外支持部分特有类型, 以贴合脚本工具集使用场景以提高可用性, 但**特有类型的数据无法作为 Dataway 脚本的输出结果**, 导致错误发生, 详情请查阅文本模式、表达式模式、代码模式 Python、代码模式 Java。

## Message 类型

在 Dataway 中, Message 类型承载了腾讯轻联消息, 而腾讯轻联消息伴随集成流的执行过程传递和更新。Message 类型包含**载荷 (payload)、变量 (vars)、属性 (attrs)** 等属性, 称之为**预定义属性 (Predefined Properties)**, 这些属性是由系统根据当前运行信息及处理的数据生成的, 用于在 Dataway 脚本中获取集成流的当前上下文信息。

**说明:**

Dataway 脚本以 msg 变量作为输入, 该变量即 Message 类型, 承载了集成流执行至当前组件节点前夕的上下文信息。

Message 类型中包含的属性及其说明如下:

属性名	获取方法（以代码模式 Python 为例）	描述	属性类型	属性说明
变量集	msg.vars	当前消息上下文中的变量集合	字典类型：键为字符串类型，代表变量名；值为任意 <b>核心类型</b> ，代表变量值	已设置变量在集成流的所有后置组件节点共享，因此可用于在不同的组件节点之间的数据整合
载荷	msg.payload	当前消息的载荷数据	任意 <b>核心类型</b>	载荷是腾讯轻联消息的负载数据，反映了组件节点的执行结果，由组件节点执行后更新，也可以通过特定组件（如“配置 payload”等）进行配置。例如，http listener 组件会根据接收到的网络请求来构造载荷的内容，因此 listener 组件处理之后，载荷为 Entity 类型
属性集	msg.attrs	当前消息的属性数据集合，如消息来源、消息的头部信息等	字典类型：键为字符串类型，代表属性名；值为任意 <b>核心类型</b> ，代表属性值	如果集成流触发器为 http listener，则网络请求的 headers 将会存储在 msg.attrs
唯一标志	msg.id	当前消息的唯一标识 id	字符串类型	经过一个逻辑组件，唯一标志可能会变化
序列号	msg.seq_id	当前消息的序列号	字符串类型	消息在集成流中流转时，序列号保持不变
错误信息	msg.error	当前处理上下文中的错误信息	字典类型：键为字符串类型，代表错误属性名；值为字符串类型，代表属性值	包含的内容有：'code': 错误类型；'desc': 错误描述字符串

## Entity 类型

### Entity 类型简介

在 Dataway 中，Entity 类型承载了腾讯轻联实体数据，是二进制数据的封装对象，其主要组成部分包括原始数据（blob）、MIME 类型（mime\_type）以及编码类型 encoding。

- **原始数据（blob）**：原始的二进制数据。
- **MIME 类型（mime\_type）**：二进制数据的内容格式，例如 application/json、application/www-form-urlencoded、multipart/form-data 等。
- **编码类型（encoding）**：二进制数据的字符编码格式，例如 utf8、gbk 等。

可以通过如下方式访问 Entity 中的内容（以代码模式 Python 为例）：

访问方式	说明
entity['^blob']	获取该二进制对象的负载数据，返回 bytes 类型的对象。
entity['^mime_type']	获取该消息对象的 MIME 类型，返回字符串对象。
entity['^encoding']	获取该消息对象的编码类型，返回字符串对象。

为了方便使用，Dataway 还提供了如 `entity.get(attr, default=None)` 等对象方法以及基于下标的选择器语法（详情请参见 [Entity 选择器](#)）以实现快速访问功能：

- `entity['^value']`: 根据 MIME 类型和编码类型，解析负载数据，并返回解析结果，类型为 [核心类型](#) 之一。
- `entity['xxxx']`: 根据 MIME 类型和编码类型，解析负载数据，并获取其中指定键对应值，相当于 `entity['^value']['xxx']`。
- `entity.get(attr, default=None)`: 根据 MIME 类型和编码类型，解析负载数据，并获取其中指定键对应值，如果获取不到则返回默认值（默认为 `None`），相当于 `entity['^value'].get(attr, default=None)`。

在使用快速访问功能时，系统会尝试解析 Entity 中的二进制负载数据，如果解析失败会导致运行时错误（详情请参见 [MIME 类型支持](#)）。

## Entity 选择器

对于常用 MIME 类型和编码类型，Dataway 支持通过选择器 (selector) 快速访问 Entity 对象内容，支持的操作类型如下：

下标类型	描述	举例
数字	访问当前数组的第 i 个元素	<code>entity[0]</code>
以 '^' 开头的字符串	获取元信息，例如 <code>^mime_type</code> 、 <code>^encoding</code> 、 <code>^blob</code> （原始二进制）、 <code>^value</code> （值）	<code>entity['^mime_type']</code>
普通字符（字母、数字、下划线、横杠、点）	普通字符的 key，按名称的方式获取当前元素的子元素，如果有多个同名的，只返回第一个	<code>entity['list']</code>

下面将对这几种选择器类型举例说明。假设输入消息的载荷 `msg.payload` 是一个 Entity 对象，其原始数据 `blob` 解析后是一个 json 数组，MIME 类型为 `"application/json"`，编码类型为 `"utf-8"`：

```
[{"a1":1}, {"b1":1, "b2":2, "b3":3}, {"c1": [1, 2, 3]}
```

### 示例1：使用数字下标获取数据

使用数字下标可以获取 `msg.payload` 中的元素。如下所示，Dataway 表达式为：

```
def dw_process(msg):
    return msg.payload[1]
```

则表达式的输出为 dict 类型的结果：`{"b1":1, "b2":2, "b3":3}`

### 示例2：使用 '^' 符号获取元数据

要获取 `msg.payload` 的元数据，可以使用 '^' 符号。如下所示，Dataway 表达式为：

```
def dw_process(msg):
    return {
        "mimeType" : msg.payload["^mime_type"],
        "encoding" : msg.payload["^encoding"],
        "blob" : msg.payload["^blob"],
        "value" : msg.payload["^value"],
    }
```

则表达式的输出为 dict 类型的结果：

```
{
  "mimeType" : "application/json",
  "encoding" : "utf-8",
  "blob" : b' [{"a1":1}, {"b1":1, "b2":2, "b3":3}, {"c1": [1, 2, 3]} ]',
  "value" : [{"a1":1}, {"b1":1, "b2":2, "b3":3}, {"c1": [1, 2, 3]}
```

```
}
```

### 示例3：使用普通字符获取元素

假设 `msg.payload` 值仍为 `Entity` 类型，其 MIME 类型仍为 `"application/json"`，编码类型仍为 `"utf-8"`，但负载数据 `blob` 解析后变成了：

```
{"a1":1,"b1":1,"b2":2,"b3":3,"c1":[1,2,3]}
```

如下所示，Dataway 表达式为：

```
def dw_process(msg):
    return {
        "a1" : msg.payload["a1"],
        "b2": msg.payload["b2"],
        "c1": msg.payload['c1'],
    }
```

则表达式的输出为 `dict` 类型的结果：

```
{
  "a1" : 1,
  "b2": 2,
  "c1": [1,2,3]
}
```

## Entity 类型对象构造（以代码模式 Python 为例）

### 1. 值构造方法（`Entity.from_value`）

将数据 `data` 封装为 `Entity` 类型并返回。如下所示：

```
Entity.from_value(data, mime_type=None, encoding="utf-8")
```

值构造方法根据给定的 MIME 类型和编码类型对 `data` 进行序列化得到 `bytes` 类型的原始数据，然后再封装为 `Entity` 类型返回。

其中 `mime_type` 参数必传，目前支持的 MIME 类型有 `text/plain`、`application/json`（别名 `text/json`）、`application/x-www-form-urlencoded`、`application/csv`、`application/xml`（别名 `text/xml`）和 `multipart/form-data` 六种；`encoding` 参数则支持任意合法的编码类型，若为空则默认为 `utf-8` 编码。

### 2. 原始数据构造方法（`Entity.from_bytes`）

将字符串或者 `bytes` 类型数据封装为 `Entity` 类型并返回。如下所示：

```
Entity.from_bytes(data, mime_type=None, encoding="utf-8")
```

原始数据构造方法中的 `mime_type` 和 `encoding` 参数校验规则与值构造方法相似，不同的是不会对 `mime_type` 参数值进行限制，可以为任意的 MIME 类型。

如果传递给原始数据构造方法的 `data` 参数类型为 `bytes` 类型，该方法会直接返回一个原始数据为 `data`，MIME 类型为 `mime_type`，编码类型为 `encoding` 的 `Entity` 对象。

如果传递的 `data` 参数是字符串数据，则会尝试根据 `encoding` 参数将其编码为 `bytes` 类型，并构造 `Entity` 对象。

## MIME 类型支持

Dataway 使用 Entity 类型可支持多种不同的数据类型，例如 json、csv、xml 等。在值构造方法或原始数据构造方法中指定对应的 MIME 类型和编码类型，可以得到一个封装不同数据类型的 Entity 对象。通过使用 Entity 选择器，可以读取解析后的结构数据。

不同的 MIME 类型在 Entity 中有不同的数据格式，对应关系如下：

MIME 类型	数据格式
application/json	JSON 格式
application/x-www-form-urlencoded	HTTP 表单格式
text/plain	文本格式
application/xml	XML 格式
application/csv	CSV 表单格式
multipart/form-data	HTTP FORM DATA 表单格式
其他 MIME 类型	其他格式

不同的数据格式有不同的编码规则、数据结构以及特定的 Entity 选择器语法。本节将对这些不同的数据格式分别进行说明。

### JSON 格式

JSON 格式的数据代表 MIME 类型为 application/json 的 Entity 中数据序列化后得到的类型。

- 使用 [原始数据构造方法](#)，则 Dataway 对输入的 str/bytes 类型最终解析成一个字典类型数据。
- 使用 [值构造方法](#)，支持列表、字典、多值字典等多种输入类型，并最终解析成一个字典类型数据。

以下将通过两个示例来说明如何使用 JSON 格式数据。

#### 示例1：构造 JSON 格式的 Entity

本示例将展示如何使用 Entity 的构造函数构造一个 JSON 格式的 Entity，并使用 Entity 选择器获取 Entity 的属性和数据。

##### • 输入

Dataway 的运行环境依赖于组件的运行，假定在"配置 Payload"组件前已经有一个"Transform"组件，对流的运行消息 msg 的 payload 进行了设置。msg.payload 为一个字典类型对象，内部结构如下。

```
{
  "name": "zhangsan",
  "age": 10,
  "male": True,
  "brothers": ["lisi", "zhaowu"]
}
```

##### • Dataway 脚本

以下 Dataway 脚本使用值构造方法将字典数据类型的 msg.payload 转换成 Entity 对象，然后用选择器获取 Entity 的元数据和元素，并返回。

```
def dw_process(msg):
    entity = Entity.from_value(msg.payload, mime_type='application/json', encoding='utf-8')
    return {
        'blob': entity['^blob'],
        'mimeType': entity['^mime_type'],
        'name': entity['name'],
    }
```

```
'brother': entity['brothers'][0],
'male': entity['^value']['age'],
'other': entity.get('other', 'other_default')
}
```

- 输出

Dataway 的脚本输出为一个字典, 其结果如下。

```
{
  "blob": b'{"name":"zhangsan","age":10,"male":true,"brothers":["lisi","zhaowu"]}',
  "mimeType": "application/json",
  "name": "zhangsan",
  "brother": "lisi",
  "male": 10,
  "other": "other_default"
}
```

## 示例2: 复杂的 JSON 结构使用

本示例将对复杂的 JSON 结构进行解析, 并运行在“配置 Payload”组件中。

- 输入

Dataway 的运行环境依赖于组件的运行, 假定在“配置 Payload”组件前已经有一个“Transform”组件, 对流的运行消息 msg 的 payload 进行了设置。msg.payload 为一个 Entity 类型对象, 内部结构如下:

```
{
  'mime_type': 'application/json',
  'encoding': 'utf-8',
  'blob': b'{"name":"zhangsan","age":10,"male":true,"brothers":["lisi","zhaowu"]}'
}
```

- Dataway 脚本

以下 Dataway 脚本将对 msg 进行处理, 获取对应的元素, 最终返回一个 Entity 对象。

```
def dw_process(msg):
    val = {
        "k0": msg.payload['^blob'].decode(msg.payload['^encoding']),
        "k1": {
            "str": "str",
            "list": [123, "abc", 12.34],
            "dict": {
                "k1": "v1"
            }
        },
        "k2": math.sqrt(10),
        "k3": json.dumps({"a": 1, "b": [1, "d"]}),
        "k4": time.strftime('%Y-%m-%d %H:%M:%S'),
        "k5": "hello" + "world",
        "k6": ["12", 3] + [[1, 2], 4.5],
        "k8": msg.payload['^mime_type'],
    }
    if val['k8'] == 'application/json':
```

```
val['k7'] = msg.payload.get('gbk')
val['k9'] = time.time()
return Entity.from_value(val, mime_type='application/json', encoding='utf-8')
```

#### • 输出

Dataaway 脚本的输出结果为一个 Entity 类型对象，为方便说明，我们将 Entity 中的原始数据进行解析后赋值给 value 属性。

```
{
  "mime_type": "application/json",
  "encoding": "utf-8",
  "value": {
    "k0": "{\\"name\\":\\"zhangsan\\",\\"age\\":10,\\"male\\":true,\\"brothers\\":
[\\"lisi\\",\\"zhaowu\\"]}",
    "k1": {
      "str": "str",
      "list": [
        123,
        "abc",
        12.34
      ],
      "dict": {
        "k1": "v1"
      }
    },
    "k2": 3.1622776601683795,
    "k3": "{\\"a\\": 1, \\"b\\": [1, \\"d\\"]}",
    "k4": "2021-04-27 10:49:54",
    "k5": "helloworld",
    "k6": [
      "12",
      3,
      [
        1,
        2
      ],
      4.5
    ],
    "k8": "application/json",
    "k7": None,
    "k9": 1619491794.670995
  }
}
```

## HTTP 表单格式

HTTP 表单格式的数据代表 MIME 类型为 application/x-www-form-urlencoded 的 Entity 中数据解析后得到的类型。

- 使用 [原始数据构造方法](#)，则 Dataaway 对输入的字符串或 bytes 类型最终解析成一个字典类型数据。
- 使用 [值构造方法](#)，支持字典或多值字典两种输入类型，并最终解析成一个多值字典类型数据。

对 HTTP 表单格式数据，由于其内部实现为多值字典数据类型，因此除了通用的 Entity 选择器语法，还支持特殊的选择器语法。

Selector	含义
[*key']	返回字典下面的 key 对应的所有元素

['key']	返回字典下面的 key 对应的第一个元素
---------	----------------------

### 示例：构造并使用 HTTP 表单格式的 Entity

本示例将展示如何使用 Entity 的构造函数构造一个 HTTP 表单格式的 Entity，并使用 Entity 选择器语法获取 Entity 的属性和数据。

- 输入

msg.payload 存储了 HTTP 表单格式的数据：k1=123&k2=helloworld&k3=2&k3=abc，在 Dataway 中以多值字典类型呈现，可转化为字典类型数据：

```
{
  "k1": 123,
  "k2": "helloworld",
  "k3": [2, "abc"]
}
```

- Dataway 脚本

以下 Dataway 脚本使用值构造方法将字典类型数据转换成 Entity 对象，然后用选择器获取 Entity 的元数据和元素。

```
def dw_process(msg):
    entity = Entity.from_value(msg.payload, mime_type='application/x-www-form-urlencoded',
    encoding='utf-8')
    return {
        'blob': entity['^blob'],
        'mimeType': entity['^mime_type'],
        'k1': entity['k1'],
        'k3': entity['^value']['k3'],
        'k3multi_selector': entity['^value']['*k3'],
        'k5': entity.get('k5', 'default_value')
    }
```

- 输出

Dataway 的脚本输出为一个字典类型数据，结果如下。

```
{
  "blob": b'k1=123&k2=helloworld&k3=2&k3=abc',
  "mimeType": "application/x-www-form-urlencoded",
  "k1": 123,
  "k3": 2,
  "k3multi_selector": [2, "abc"],
  "k5": "default_value"
}
```

### 文本格式

文本格式的数据代表 MIME 类型为 text/plain 的 Entity 中数据解析后得到的类型。无论是 [值构造方法](#) 还是 [原始数据构造方法](#)，data 参数均为字符串或 bytes 类型，entity 内容均为字符串数据类型。

### 示例：构造文本格式的 Entity

本示例将展示如何使用 Entity 的构造函数构造一个文本格式的 Entity，并使用 Entity 选择器语法获取 Entity 的属性和数据。

- 输入

msg.payload 存储了文本格式的数据："This is a text plain message"。

- Dataway 脚本

以下 Dataway 脚本使用值构造方法将字符串数据转换成 Entity 对象，然后用选择器获取 Entity 的元数据和元素。

```
def dw_process(msg):
    entity = Entity.from_value(msg.payload, mime_type='text/plain', encoding='utf-8')
    return entity['^value']
```

#### • 输出

Dataway 的脚本输出为一个字符串，结果为 "This is a text plain message"。

## XML 格式

XML 格式的数据代表 MIME 类型为 application/xml 的 Entity 中数据序列化后得到的类型。

- 使用 [原始数据构造方法](#)，则 Dataway 对输入的字符串或 bytes 类型最终解析成一个字典数据类型。
- 使用 [值构造方法](#)，仅支持字典输入类型，并最终解析成一个字典数据结构。同时，输入的字典仅包含一个默认的键 "root"，则为内置的 MultiMap，在 MultiMap 中可以自由操作 msg 属性。

对 XML 格式数据，除了通用的 Entity 选择器语法，还支持特殊的选择器语法。

Selector	含义
['*key']	返回xml某个节点 key 对应的所有元素
['key']	返回xml某个节点 key 对应的第一个元素
['#text']	获取xml某个节点的文本值
['@attr']	获取xml某个节点的attr属性值

以下将通过两个示例来说明如何使用 XML 格式数据。

### 示例1: 构造 XML 格式的 Entity

本示例将展示如何使用 Entity 的构造函数构造一个 XML 数据格式的 Entity，并使用 Entity 选择器语法获取 Entity 的属性和数据。

#### • 输入

Dataway 输入参数 msg 的 payload 值为常量1，同时 msg.vars 中包含一个键为 "abc"，值为 "123" 的键值对。

```
{
  "payload": 1,
  "vars": {
    "abc": "123"
  }
}
```

#### • Dataway 脚本

以下 Dataway 脚本使用 [值构造方法](#) 将字典类型转换成 Entity 对象，并返回。

```
def dw_process(msg):
    a = math.floor(1.4)
    return Entity.from_value({
        'root': {
            'k1': msg.vars['abc'],
            'k2': json.dumps('哈哈', ensure_ascii=False),
            'k3': a + 1,
            '@id': "hello",
            '#text': "<a>dwad</a>",
        }
    })
```

```
        'k4': ['abc', 'def', None],
    }
}, mime_type = 'application/xml')
```

#### • 输出

Dataway 脚本的输出结果为一个 Entity 类型对象，其中原始数据 (blob) 为符合 XML 语法的二进制数据，如下所示。

```
{
  "mime_type": "application/xml",
  "encoding": "utf-8",
  "blob": b'<?xml version="1.0" encoding="utf-8"?>
<root id="hello"><k1>123</k1><k2>"哈哈"</k2><k3>2</k3><k4>abc</k4><k4>def</k4><k4></k4>
<a>dwad</a></root>'
}
```

### 示例2: 使用 XML 特定选择器

本示例将示范如何在 XML 格式数据中使用特定语法的选择器。

#### • 输入

Dataway 输入参数 msg 的 payload 值为常量1，同时 msg.vars 中包含一个键为 "abc"，值为 "123" 的键值对。

```
{
  "payload": 1,
  "vars": {
    "abc": "123"
  }
}
```

#### • Dataway 脚本

以下 Dataway 脚本使用 [值构造方法](#) 将字典类型转换成 Entity 对象，然后用选择器获取 Entity 的数据。

```
def dw_process(msg):
    a = math.floor(1.4)
    entity = Entity.from_value({
        'root': {
            'k1': msg.vars['abc'],
            'k2': json.dumps('哈哈', ensure_ascii=False),
            'k3': a + 1,
            '@id': "hello",
            '#text': "<a>dwad</a>",
            'k4': ['abc', 'def', None],
        }
    }, mime_type = 'application/xml')
    return {
        'k1': entity['root']['#text'] + entity['root']['@id'],
        'k2': entity['root']['k1'],
        'k3': entity['root']['*k4']['^value'],
        'k4': entity['root']['k4'],
        'k5': entity['^mime_type']
    }
```

## • 输出

Dataway 脚本的输出结果为一个字典类型数据，如下所示。

```
{
  "k1": "<a>dwad</a>hello",
  "k2": "123",
  "k3": ['abc', 'def', None],
  "k4": "abc",
  "k5": "application/xml"
}
```

## ❗ 说明:

在 XML 格式数据中，root 节点为默认节点，其属性使用 @id=123 的方式指定，文本使用 #text 的方式指定。root 节点值为一个 MultiMap 类型，其中键为每一个子节点名称，值为相应子节点的值。

## CSV 格式

CSV 格式的数据代表 MIME 类型为 application/csv 的 Entity 中数据序列化后得到的类型。

- 使用 [原始数据构造方法](#)，则 Dataway 对输入的字符串或 bytes 类型最终解析成一个列表类型数据结构，其中每一项元素均为一个字典。
- 使用 [值构造方法](#)，支持列表类型输入，并最终解析成一个列表类型数据结构，其中每一项元素均为一个字典。

以下将通过示例来说明如何使用 CSV 格式数据。

### 示例：构造 CSV 格式的 Entity

本示例将展示如何使用 Entity 的构造函数构造一个 CSV 格式的 Entity，并使用 Entity 选择器语法获取 Entity 的属性和数据。

#### • 输入

Dataway 输入参数 msg 的 payload 值为常量1，同时 msg.vars 中包含一个键为 "abc"，值为 "123" 的键值对。

```
{
  "payload": 1,
  "vars": {
    "abc": "123"
  }
}
```

#### • Dataway 脚本

以下 Dataway 脚本使用 [值构造方法](#) 将字典类型转换成 Entity 对象，并使用选择器语法获取 Entity 对象的属性值。

```
def dw_process(msg):
    entity = Entity.from_value([
        {'k1': 'abcd', 'k2': 123.0, 'k3': True},
        {'k1': 'def', 'k2': 'dwad,2e', 'k3': 10},
    ], mime_type = 'application/csv')
    return {
        'var1': entity['^blob'],
        'var2': entity['^mime_type'],
        'var3': entity['^encoding'],
        'var4': entity[0]['k2'] + entity[1]['k3'],
        'var5': entity[1]['k2']
    }
```

## • 输出

Dataway 脚本的输出结果为一个字典类型数据，如下所示。

```
{
  "var1": b'k1,k2,k3\r\nabcd,123.0,True\r\ndefs,"dwdw,2e",10\r\n',
  "var2": "application/csv",
  "var3": "utf-8",
  "var4": "133",
  "var5": "dwdw,2e"
}
```

### ⓘ 说明:

对于 CSV 数据格式，接收的列表每一项元素均为字典类型。每一项元素中的键需保持相同，作为 CSV 文本的标题行；每一项元素中的值则代表该行的数据值，用逗号分隔。

## HTTP Form-Data 表单

HTTP Form-Data 表单格式的数据代表 mime-type 为 multipart/form-data 的 Entity 中数据序列化后得到的类型。

### 浏览器中的 multipart/form-data

在浏览器发送 Content-Type 为 multipart/form-data 的请求时，实际传输的 byte 数组转换成字符串如下所示。

每一项参数由一个 boundary 开头，标识着这一项的开始，例如 --34b21。

### ⚠ 注意:

-- 为固定开头，34b21 为浏览器随机的一个不超过70位的字符串。

下面两行分别是 Content-Disposition 和 Content-Type 的固定 headers。其中 Content-Disposition 包含 name 和 filename 两项，Content-Type 即为输入内容的 Content-Type。name 为参数名，filename 为文件名。如果 filename 为 "\*.txt" 或者为空，则 Content-Type 默认为 text/plain；如果 filename 为其他，则 Content-Type 会根据后缀名自动判断。同时，也支持其他的扩展 headers。

再往下是实际的内容，如果 Content-Type 为 text/plain，则为普通字符串，例如 "Book"；如果 Content-Type 为 application/json，则为一个 json 类型的字符串，如下方的 file1 对应的 json 结构。

最后，会用 --34b21-- 来标识这段请求的结束。

```
--34b21
Content-Disposition: form-data; name="text"
Content-Type: text/plain

Book
--34b21
Content-Disposition: form-data; name="file1"; filename="a.json"
Content-Type: application/json

{
  "title": "Java 8 in Action",
  "author": "Mario Fusco",
  "year": 2014
}
--34b21
Content-Disposition: form-data; name="file2"; filename="a.html"
Content-Type: text/html

<!DOCTYPE html>
```

```
<title>
Available for download!
</title>
--34b21--
```

## Form-Data 的构造和使用

- 使用 [原始数据构造方法](#)，则 Dataway 对输入的字符串或 bytes 类型最终解析成一个 [FormDataParts](#) 数据结构。
- 使用 [值构造方法](#)，支持 Entity、Formdata、列表/字典输入类型，并最终解析成一个 [FormDataParts](#) 数据结构。

### ❗ 说明:

- 如果为 Entity 类型，则会首先判断 MIME 类型是否为 multipart/form-data。如果是，直接返回这个 Entity；否则将报错。
- 如果为 FormDataParts 类型（即上文提到的列表/字典结构），则直接赋值给 Entity。
- 如果为列表类型，则需要保证为以下数据结构：每一项的第一个元素为参数名。第二个元素如果为列表，则第二个元素的第一项为文件名，第二项为实际的内容，第三项为 Content-Type，第四项为一个字典，代表 extra\_headers；如果第二个元素类型为字符串，则文件名为空，实际内容即为字符串内容，Content-Type 默认为 text/plain。

对 HTTP Form-Data 格式数据，除了通用的 Entity 选择器语法，还支持特殊的选择器语法。

Selector	含义
['parts']	返回自定义类型 FormDataParts 类型
['parts'][0]	返回 FormData 的第0项
['parts']['a']['content']	返回 FormDataParts 中键为'a'的值中的 content 值
['boundary']	返回 FormDataParts 的分隔符

## 应用示例

以下将通过两个示例来说明如何使用 HTTP FORM-DATA 表单格式数据。

### 示例1: 构造 HTTP Form-Data 格式的 Entity

本示例将展示如何使用 Entity 的构造函数构造一个 HTTP Form-Data 数据格式的 Entity。

#### • 输入

Dataway 输入参数 msg 的变量 (vars) 中包含一个键为“abc”，值为“123”的键值对。

```
{
  "vars": {
    "abc": "123"
  }
}
```

#### • Dataway 脚本

以下 Dataway 脚本使用 [值构造方法](#) 将字典类型转换成 Entity 对象，并返回。

```
def dw_process(msg):
    a = math.floor(1.4)
    c = Entity.from_value({
        'k1': msg.vars['abc'],
        'k2': json.dumps('哈哈', ensure_ascii=False),
    })
```

```

        'k3': a + 1,
    }, mime_type = 'application/json')
    return Entity.from_value(
        [
            ('a', ('test.json', '{"a": a}', 'application/json', {'Test111': 1})),
            ('b', '333'),
            ('c', ('c.json', c, c['^mime_type']))
        ],
        mime_type='multipart/form-data; boundary=123333333'
    )

```

#### • 输出

Dataway 脚本的输出结果为一个 Entity 类型对象，其中原始数据（blob）为符合 multipart/form-data 结构的二进制数据，如下所示。

```

{
  "mime_type": "multipart/form-data; boundary=12345",
  "encoding": "utf-8",
  "blob": b'--123333333
Content-Disposition: form-data; name="a"; filename="test.json"
Content-Type: application/json
Test111: 1

{"a": a}
--123333333
Content-Disposition: form-data; name="b"

333
--123333333
Content-Disposition: form-data; name="c"; filename="c.json"
Content-Type: application/json

{"k1": "123", "k2": "\哈哈", "k3": 2}
--123333333--
' '
}

```

#### 示例2: 使用 FORM-DATA 选择器

本示例将示范如何在 FORM-DATA 格式数据中使用特定语法的选择器。

#### • 输入

Dataway 输入参数 msg 的 payload 为一个 Entity 类型，其 MIME 类型为 multipart/form-data，通过 [值构造方法](#) 创建。

```

def dw_process(msg):
    return Entity.from_value(
        [
            ('a', ('test', '1233', 'application/json', {'test111':1})),
            ('b', (None, '2333', 'text/plain')),
        ],
        mime_type="multipart/form-data; boundary=ce560532019a77d83195f9e9873e16a1"
    )

```

#### • Dataway 脚本

以下 Dataway 脚本将使用 FORM-DATA 选择器对 msg.payload 进行处理，并返回一个字典。

```
def dw_process(msg):
    return {
        'k1': str(type(msg.payload)),
        'k2': msg.payload['^mime_type'],
        'k3': msg.payload['^encoding'],
        'k4': msg.payload['^blob'],
        'k5': msg.payload['boundary'],
        'k6': msg.payload['parts']['a']['headers']['Content-Disposition']['name']
            + '-' + msg.payload['parts'][1]['headers']['Content-Type']
    }
```

#### • 输出

Dataway 脚本的输出结果为一个字典类型数据，结果如下。

```
{
  'k1': 'Entity',
  'k2': 'multipart/form-data;boundary=ce560532019a77d83195f9e9873e16a1',
  'k3': 'utf-8',
  'k4': b'--ce560532019a77d83195f9e9873e16a1
Content-Disposition: form-data; name="a"; filename="test"
Content-Type: application/json
test111: 1

1233
--ce560532019a77d83195f9e9873e16a1
Content-Disposition: form-data; name="b"
Content-Type: text/plain

2333
--ce560532019a77d83195f9e9873e16a1--
'',
  'k5': 'ce560532019a77d83195f9e9873e16a1',
  'k6': 'a-text/plain'
}
```

## 其他类型

对其他类型的 MIME 类型，Dataway 不支持直接用 [值构造方法](#) 构造，但支持从上游读取数据，以及使用 [原始数据构造方法](#) 构造一个封装的 Entity。

下面将通过一个示例来说明，假设输入数据为一个二进制 bytes 流，我们通过"配置 Payload"组件中使用 Dataway 表达式将该二进制 bytes 流封装到 msg.payload 中。

### Dataway 表达式

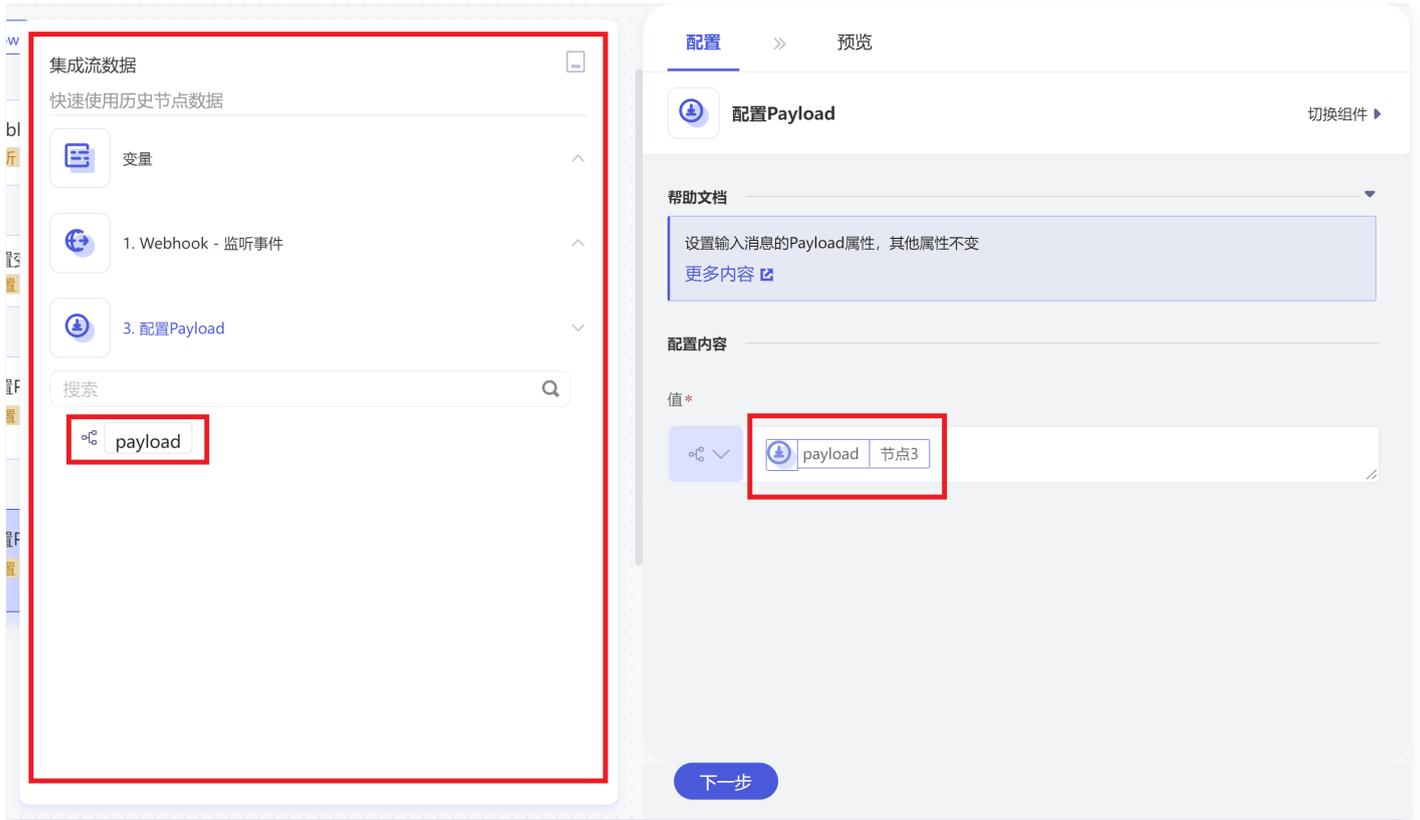
```
def dw_process(msg):
    b = msg.payload
    return Entity.from_bytes(b, mime_type='application/octet-stream')
```

然后在下游可以使用 [Entity 选择器](#) 语法进行操作。

## 集成流数据面板

Dataway 支持可视化数据引用，在**集成流数据面板**单击数据标签即可引用集成流上下文中的相应数据（包括变量、前置组件输出等），打通组件间的数据高速路，提升用户体验。当前 Dataway 所有模式均支持可视化数据引用功能，包括 **文本模式**、**表达式模式**、**代码模式 Python** 和 **代码模式 Java**。

1. 单击 Dataway 输入文本框时自动弹出**集成流数据面板**。
2. 单击选择**集成流数据面板**中的数据标签，Dataway 编辑文本框中会自动引用集成流上下文数据，并在当前光标位置插入对应的数据标签。



# 文本模式

最近更新时间：2024-08-01 15:33:02

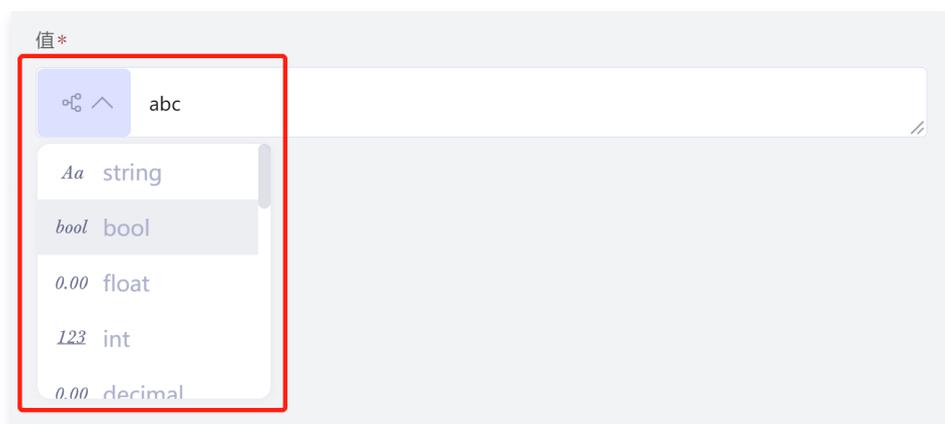
文本模式针对各数据类型进行定制化的交互设计，用户仅需要进行简单操作即可生成所需数据。

## IDE 使用

1. 对于任意 Dataway 编辑文本框，将鼠标移至编辑文本框，会自动弹出模式选择按钮，单击**文本**进入文本模式。



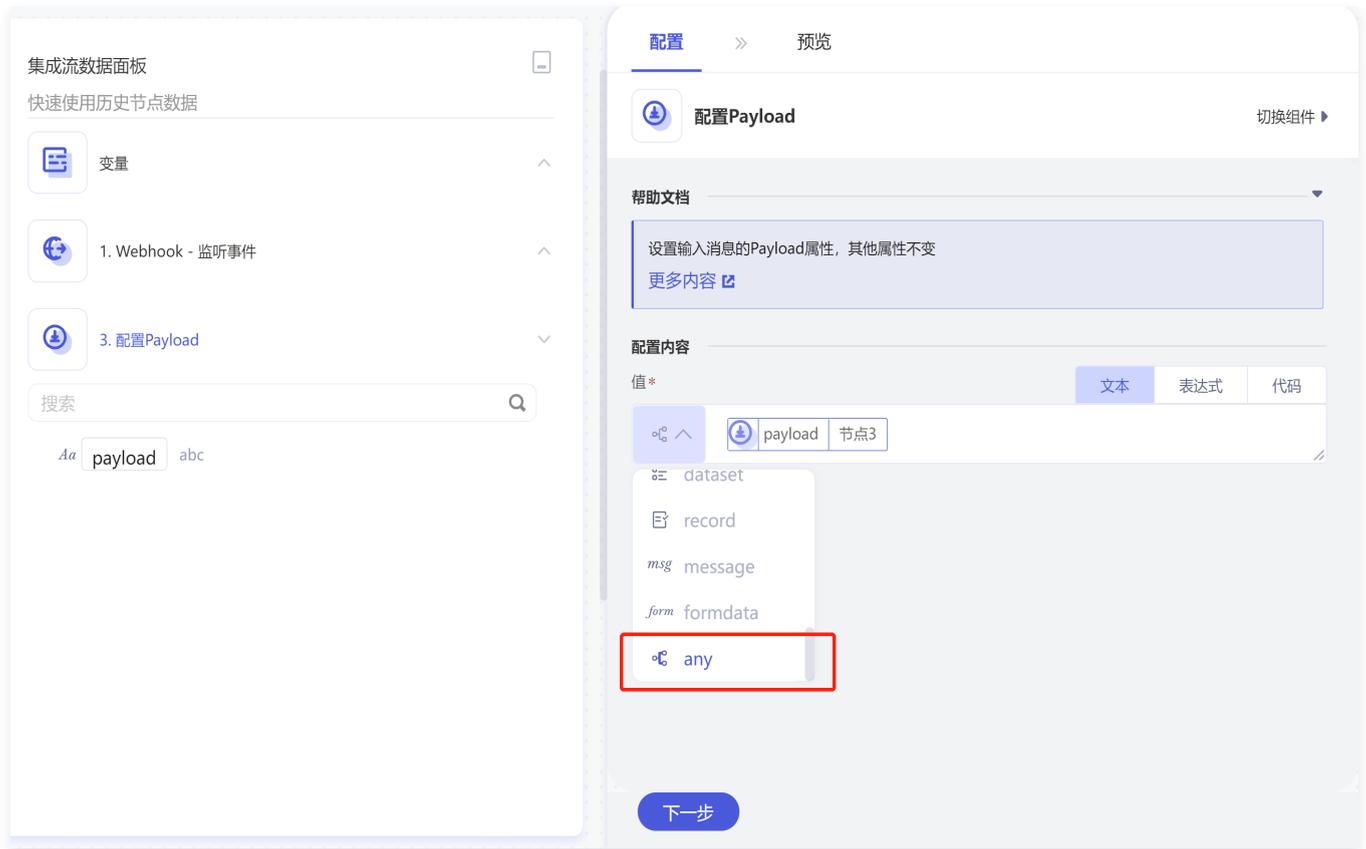
2. 单击选择左侧下拉菜单中的类型选择所需数据类型，Dataway 会提供针对性的输入交互界面。



## 数据类型

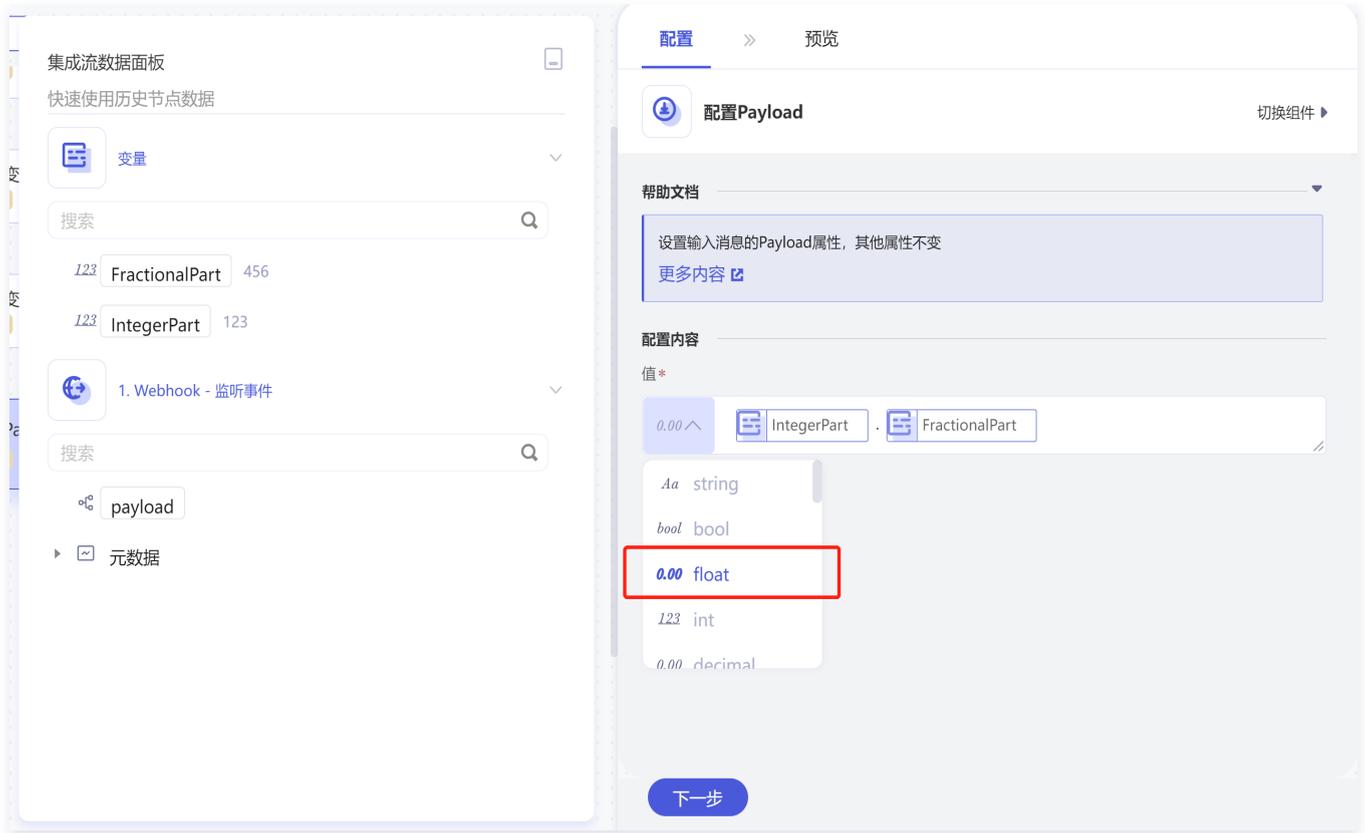
### 任意类型（any，默认类型）

可直接输入文本生成字符串，也支持 [集成流数据面板](#) 引用集成流上下文数据。当文本和集成流数据面板引用存在多项时，将以这些数据项为元素，组建列表。



### 整数 (int)、浮点数 (float)、字符串 (string)、十进制 (decimal)、布尔值 (bool)

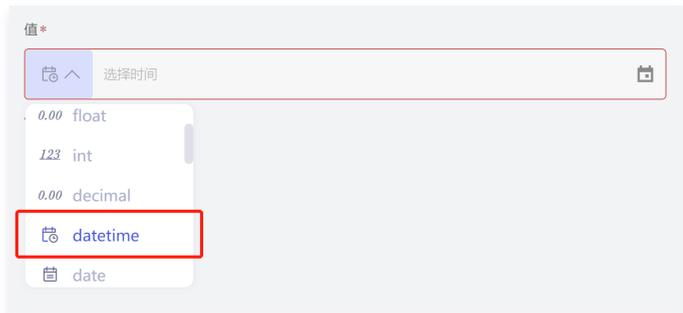
可直接输入字面量生成对应数据, 也支持 [集成流数据面板](#) 引用, 方便用户使用集成流的上下文数据。集成流数据面板引用的数据会转化为字符串, 拼接到字面量中。



**时刻 (datetime)、日期 (date)、时间 (time)**

单击文本框弹出相应可视化交互界面，在可视化交互界面上点选相应时间数据。

时刻：



日期：

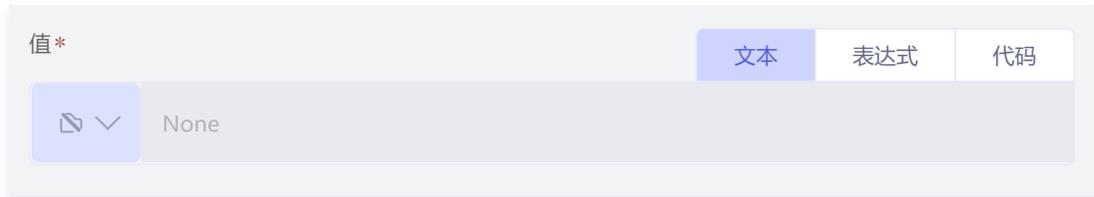


时间：



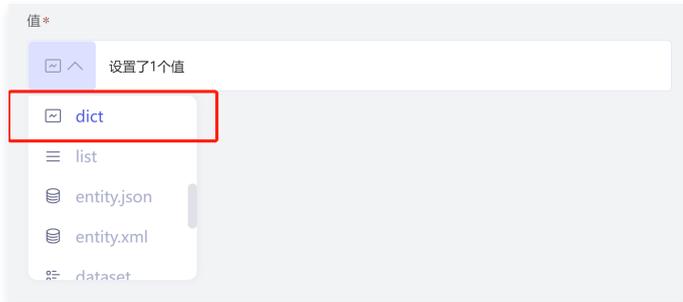
### 空值 (None)

该类型代表输入数据为空值，禁止文本框输入。



### 字典 (dict)、列表 (list)、消息 (Message)、表单 (FormData)

1. 单击文本框，弹出相应交互界面。



2. 在交互界面上逐项添加内容并确认。单击添加，填写好变量名，单击确定添加即可。



### JSON 二进制 (Entity.json)、XML 二进制 (Entity.xml)

1. 单击文本框，弹出相应文本输入界面。



2. 在文本输入界面填写相应文本并单击确定进行保存，自动生成对应 MIME 类型 (json/xml) 的 UTF-8 编码的 Entity 数据。



**数据集 (DataSet)、单条数据 (Record)**

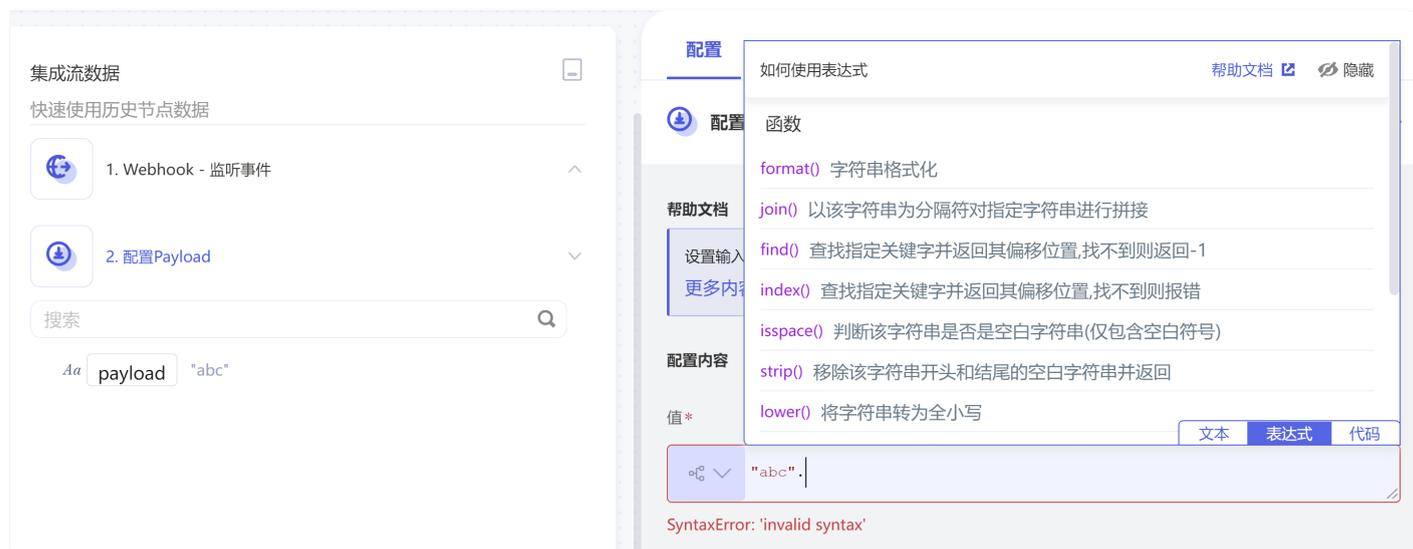
文本模式支持 [集成流数据面板](#) 引用集成流上下文的"数据集成"数据：数据集和单条数据。数据集和单条数据作为数据集成的专有数据类型，需要通过特定组件（构造数据集组件等）生成，Dataway 不提供生成方法。

# 表达式模式

最近更新时间：2024-08-01 15:33:02

## 简介

在表达式模式下，用户可以输入一个合法的 Python 表达式以返回所需要的数据。表达式模式经过针对性优化，提供智能提示功能简化了用户输入的同时，性能相较代码模式进一步获得提升。



## IDE 使用

对于任意 Dataway 编辑文本框，将鼠标移至编辑文本框，会自动弹出模式选择按钮，单击**表达式**模式，再单击文本框即可填写表达式。



### ● 语法检查

在 Dataway 交互界面中，能够实时对表达式进行语法检查，如果出错，将标红文本框，并在文本框下侧提示错误。

用户可以根据错误提示来对表达式进行修改，只有当流中所有表达式均通过语法检查才能进行发布。



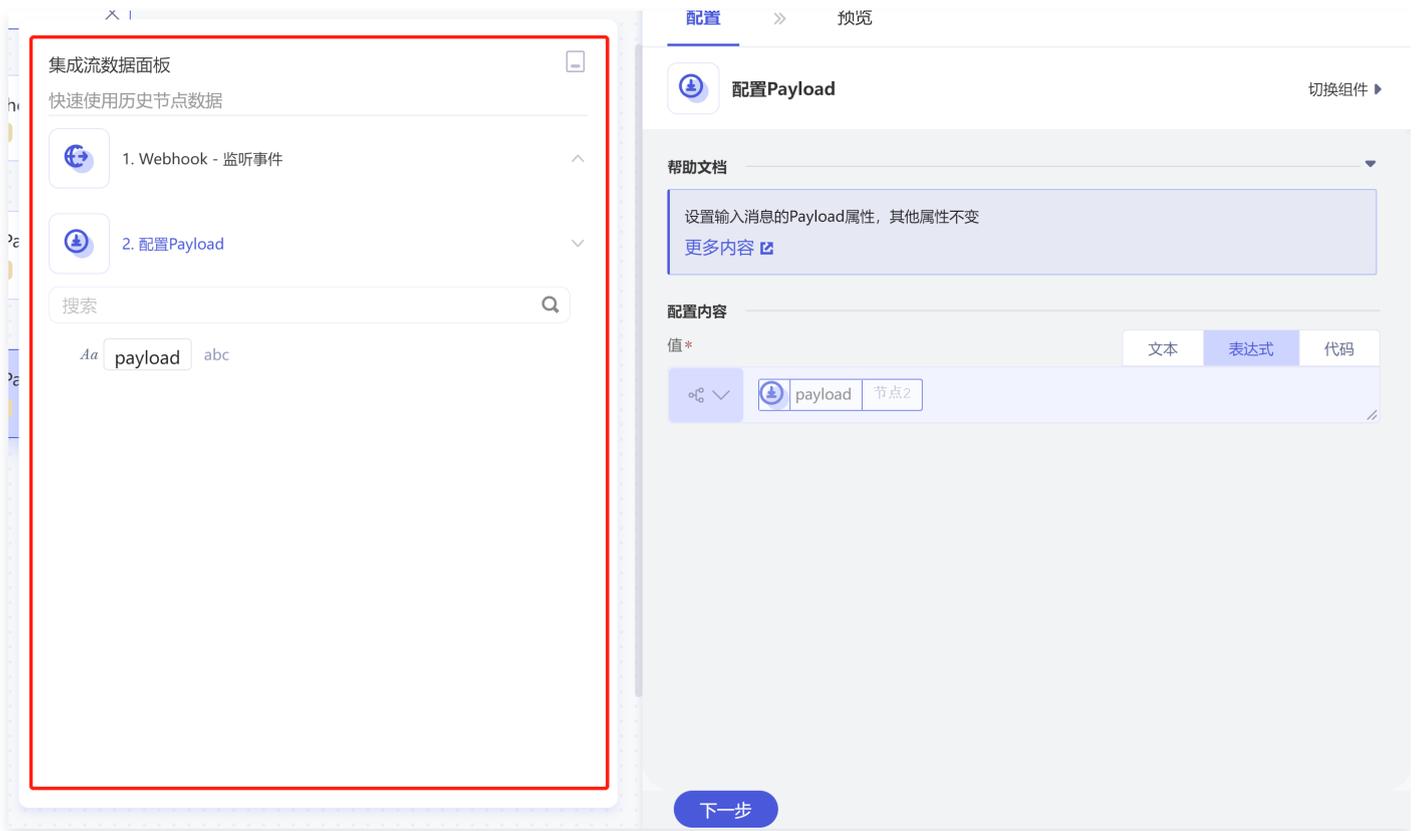
### ● 自动补全

在编辑框中进行输入时，Dataway 交互界面能够根据当前上下文自动给出语法提示和可行补全方案，并展示在文本框的上/下侧。用户可以通过点击对应标签快速补全表达式。语法提示的范围包括属性、方法、内置函数和第三方模块。



● 集成流数据面板引用

表达式模式支持 [集成流数据面板](#) 引用功能, 详情请见 [集成流数据面板](#)。



● 类型转换

除了部分组件对 Dataway 交互界面有特殊要求外, 表达式模式支持便捷类型转换功能。用户可以通过点选文本框左侧的下拉菜单中所需的数据类型, 对表达式输出结果进行显式的类型转换, 强制其符合用户对数据类型的使用需求。类型选择默认为 any, 即不进行类型转换。



## 语法

表达式模式对标 Python3 语法的 eval() 函数以降低用户使用门槛。与代码模式一致，表达式模式可以通过 msg (Message 类型) 引用当前的集成流消息。同时，表达式模式支持 [集成流数据面板](#) 快捷引用前置组件的输出数据。

表达式模式支持的语法结构如下：

表达式分组	表达式类型	释义	举例	说明
原子表达式	literal	字面量	"abc"、123、True、b'abc'	例如 string、int、float、bool、bytes等类型的字面量
	name	标识符	abc	从全局上下文中读取指定名称的变量
	tuple	元组构造	('a', 'b', 'c')、(str(1))	若至少有一个逗号，则返回元组；否则返回单个表达式的值
	list	列表构造	[1,2,3]	枚举元素，构造一个列表
	list-comp	列表推导式	[i for i in 'abc']	通过推导构造一个列表
	set	集合构造	{1,2,'a'}	枚举元素，构造一个集合
	set-comp	集合推导式	{i for i in 'abc'}	通过推导构造一个集合
	dict	字典构造	{1:2, 'a':'b', 3.0:True}	枚举元素，构造一个字典
	dict-comp	字典推导式	{i:i for i in range(10)}	通过推导构造一个字典
原型表达式	generator	生成器构造	(k*k for k in range(10))	返回一个生成器
	attr	属性引用	msg.payload	返回属性
	index	容器下标取值	msg.payload[1]、msg.vars['a']	指定下标
	slice	切片下标	msg.payload[1:3]	指定下标

	call	调用	str('a')	函数调用
数学表达式	binop	二元操作符	3**3、3+3、'a' is not in msg.vars	幂运算(**)、算数运算(+、-、*、/、//、%)、移位运算(>>、<<)、位运算(&、^、 )、比较运算(>、>=、<、<=、==、!=、is、is not、in、not in)
	uniop	一元操作符	not msg.vars、~msg.vars['no']	取自身+、取负-、取反~、逻辑非not
条件表达式	if-expr	条件表达式	'a' if 's' in msg.vars else 'b'	xx if True else xx
	logical	逻辑表达式	a and b、not True	布尔运算(and、not、or)
特殊表达式	dateref	数据引用	通过下拉点选自动生成对应标签	从上下文数据中引用指定路径的数据

## 数据类型

表达式模式在 Python 原生类型的基础上充分支持腾讯云腾讯轻联 [核心类型](#)，对于已绑定腾讯轻联核心类型的数据类型，其数据可以在组件与 Dataway 之间自由流通。

数据类型	绑定腾讯轻联核心类型	特性使用	特性类型	特性功能	输出类型
int	整数	+	操作符	加法	int
		-	操作符	减法	int
		*	操作符	乘法	int
		/	操作符	除法	float
		//	操作符	整除	int
		%	操作符	取余	int
		-x	操作符	取反	int
		&	操作符	按位与	int
			操作符	按位或	int
		^	操作符	按位异或	int
		~	操作符	按位取反	int
		<<	操作符	左移	int
		>>	操作符	右移	int
		<	操作符	小于	bool
		>	操作符	大于	bool
		<=	操作符	小于等于	bool
		>=	操作符	大于等于	bool
==	操作符	等于	bool		

		!=	操作符	不等于	bool
str	字符串	+	操作符	拼接	str
		*	操作符	重复	str
		[index]	下标操作	取指定 index 的值	str
		[index1:index2]	下标操作	取切片	str
		[index1:index2:step]	下标操作	按步长取切片	str
		in	操作符	是否子串	bool
		%	操作符	格式化	str
		==	操作符	等于	bool
		!=	操作符	不等于	bool
bool	布尔值	or	操作符	或	bool
		and	操作符	与	bool
		not	操作符	取反	bool
		==	操作符	等于	bool
		!=	操作符	不等于	bool
float	浮点数	+	操作符	加法	float
		-	操作符	减法	float
		*	操作符	乘法	float
		/	操作符	除法	float
		//	操作符	整除	float
		%	操作符	取余	float
		-x	操作符	取反	float
		<	操作符	小于	bool
		>	操作符	大于	bool
		<=	操作符	小于等于	bool
		>=	操作符	大于等于	bool
		==	操作符	等于	bool
		!=	操作符	不等于	bool
bytes	(非核心类型)	+	操作符	拼接	bytes
		*	操作符	重复	bytes
		[index]	下标操作	取指定 index 的值	int
		[index1:index2]	下标操作	取切片	bytes

		[index1:index2:step]	下标操作	按步长取切片	bytes
		in	操作符	是否子串	bool
		%	操作符	格式化	bytes
		==	操作符	等于	bool
		!=	操作符	不等于	bool
list	列表	+	操作符	拼接	list
		*	操作符	重复	list
		[index]	下标操作	取指定 index 的值	any
		[index1:index2]	下标操作	取切片	list
		[index1:index2:step]	下标操作	按步长取切片	list
		in	操作符	是否元素	bool
		<	操作符	逐项小于	bool
		>	操作符	逐项大于	bool
		<=	操作符	逐项小于等于	bool
		>=	操作符	逐项大于等于	bool
		==	操作符	等于	bool
		!=	操作符	不等于	bool
dict	字典	[key]	下标操作	取指定key的值	any
		==	操作符	等于	bool
		!=	操作符	不等于	bool
set	(非核心类型)	&	操作符	交集	set
			操作符	并集	set
		-	操作符	减法	set
		<	操作符	是否真子集	bool
		>	操作符	是否真超集	bool
		<=	操作符	是否子集	bool
		>=	操作符	是否真超集	bool
		in	操作符	是否元素	bool
		==	操作符	等于	bool
		!=	操作符	不等于	bool
decimal.Decimal	十进制	+	操作符	加法	decimal
		-	操作符	减法	decimal

		*	操作符	乘法	decimal
		/	操作符	除法	decimal
		%	操作符	取余	decimal
		-x	操作符	取反	decimal
		<	操作符	小于	bool
		>	操作符	大于	bool
		<=	操作符	小于等于	bool
		>=	操作符	大于等于	bool
		==	操作符	等于	bool
		!=	操作符	不等于	bool
datetime.datetime	时刻	year	属性	年	int
		month	属性	月	int
		day	属性	日	int
		hour	属性	时	int
		minute	属性	分	int
		second	属性	秒	int
		microsecond	属性	微秒	int
		+	操作符	推进	datetime.datetime
		-	操作符	求时间间隔	datetime.timedelta
		<	操作符	小于	bool
		>	操作符	大于	bool
		<=	操作符	小于等于	bool
		>=	操作符	大于等于	bool
		==	操作符	等于	bool
!=	操作符	不等于	bool		
datetime.date	日期	year	属性	年	int
		month	属性	月	int
		day	属性	日	int
		strftime(format)	方法	格式化	str
		+	操作符	推进	datetime.date
		-	操作符	求时间间隔	datetime.timedelta

					elta
		<	操作符	小于	bool
		>	操作符	大于	bool
		<=	操作符	小于等于	bool
		>=	操作符	大于等于	bool
		==	操作符	等于	bool
		!=	操作符	不等于	bool
datetime.time	时钟	hour	属性	时	int
		minute	属性	分	int
		second	属性	秒	int
		microsecond	属性	微秒	int
		<	操作符	小于	bool
		>	操作符	大于	bool
		<=	操作符	小于等于	bool
		>=	操作符	大于等于	bool
		==	操作符	等于	bool
		!=	操作符	不等于	bool
Entity	二进制实体	from_bytes(bs,mime_type=None,encoding="utf-8")	静态方法	从二进制构建Entity	Entity
		from_value(obj,mime_type=None,encoding="utf-8")	静态方法	从数据构建Entity	Entity
		get(key,dafault=None)	方法	获取数据	any
		[key]	下标操作	取指定key的值	any
		[^value]	下标操作	取解析后的值	any
		[^blob]	下标操作	取二进制原始数据	bytes
RecordSet	数据集	schema()	方法	获取schema	dict
Record	单条数据	[key]	下标操作	取指定key的值	any
Message	消息	payload	属性	返回输出	any
		attrs	属性	返回属性	dict
		vars	属性	返回变量	dict
		id	属性	返回消息唯一标志	str
		seq_id	属性	返回流序列号	str

		error	属性	返回错误	dict
		isthrowing	属性	是否抛错	bool

## 其他支持

表达式模式提供多样的类型方法、内置函数和第三方模块，用户可以按需选用，快速实现既定功能，详情请参见 [表达式模式附录](#)。

# 表达式模式附录

最近更新时间：2024-11-07 11:41:52

## Python 原生类型方法

为了方便用户对 Python 原生类型的操作，Dataway 支持 Python 原生类型的常用方法：

数据类型	方法	方法类型	功能	输出类型
str	endswith(suffix[, start[, end]])	方法	后缀对比	bool
	split(sep=None, maxsplit=-1)	方法	分割	list
	startswith(prefix[, start[, end]])	方法	前缀对比	bool
	count(sub[, start[, end]])	方法	统计子串数量	int
	find(sub[, start[, end]])	方法	查找匹配子串	int
	format(*args, **kwargs)	方法	格式化	str
	index(sub[, start[, end]])	方法	索引匹配子串	int
	isascii()	方法	返回全是 ASCII 字符	bool
	isspace()	方法	返回是否非空且全是空白字符	bool
	encode(encoding="utf-8", errors="strict")	方法	编码	bytes
	join(iterable)	方法	拼接	str
	lower()	方法	小写	str
	replace(old, new[, count])	方法	替换	str
	strip([chars])	方法	移除特定字符组成的前后缀	str
upper()	方法	大写	str	
bytes	count(sub[, start[, end]])	方法	统计子串数量	int
	find(sub[, start[, end]])	方法	查找匹配子串	int
	index(sub[, start[, end]])	方法	索引匹配子串	int
	decode(encoding="utf-8", errors="strict")	方法	解码	str
	replace(old, new[, count])	方法	替换	bytes
	rstrip([chars])	方法	移除特定字符集合组成的后缀	bytes
	strip([chars])	方法	移除特定字符集合组成的前后缀	bytes
	split(sep=None, maxsplit=-1)	方法	分割	list
	startswith(prefix[, start[, end]])	方法	前缀对比	bool
	endswith(suffix[, start[, end]])	方法	后缀对比	bool

float	is_integer()	方法	是否整数	bool
list	count(x)	方法	统计元素数量	int
	index(sub[, start[, end]])	方法	索引元素	int
tuple	count(x)	方法	统计元素数量	int
	index(sub[, start[, end]])	方法	索引元素	int
dict	get(key[, default])	方法	获取 key 对应值	any
	items	方法	获取键值对列表	list
set	union(*others)	方法	返回求并集后的新集合	set
datetime.datetime	today()	类方法	无时区当前时间	datetime.datetime
	fromtimestamp(timestamp, tz=None)	类方法	从时间戳构造时间	datetime.datetime
	now()	类方法	带时区当前时间	datetime.datetime
	strptime(date_string, format)	类方法	从格式化构造时间	datetime.datetime
	time()	方法	转换为时钟	datetime.time
	date()	方法	转换为日期	datetime.date
	strftime(format)	方法	格式化	str
datetime.date	today()	类方法	当前日期	datetime.date
	strftime(format)	方法	格式化	str
datetime.time	strftime(format)	方法	格式化	str
DataSet	id()	方法	获取数据集 ID	int
	partitions()	方法	获取数据集分区数	int
	schema()	方法	获取数据集 schema	Schema
Record	data()	方法	以列表形式返回各项数据	list
	get(name, default=None)	方法	获取字段名对应数据	any
	schema()	方法	获取数据集 schema	Schema

## 内置常量和函数

表达式模式支持常量：None、True 和 False。除此之外，为了方便用户使用，表达式模式内置了许多函数，用户可以调用相关函数，快速实现对应的功能，获取所需数据。

内置函数	函数说明
abs(x)	返回绝对值（整数、浮点数）
all(iterable)	返回是否所有元素为真或为空

<code>any(iterable)</code>	返回是否存在元素为真
<code>ascii(object)</code>	打印 object 但不处理非 ASCII 字母
<code>bool([x])</code>	转换为 bool
<code>bytes([source[,encoding[,errors]]])</code>	转换为 bytes
<code>chr(i)</code>	返回 int 对应的 unicode
<code>dict(kwarg)/dict(mapping,kwarg)/dict(iterable,kwarg)</code>	转换为 dict
<code>float([x])</code>	转换为 float
<code>int(x)/int(x,base)</code>	转换为 int
<code>len(s)</code>	返回长度
<code>list([iterable])</code>	转换为 list
<code>max(iterable,[,key,default])/max(arg1,arg2,args[,key])</code>	返回最大值
<code>min(iterable,[,key,default])/min(arg1,arg2,args[,key])</code>	返回最小值
<code>ord(c)</code>	返回 char 对应编码
<code>pow(x,y[,z])</code>	返回指数
<code>range(stop)/(start,stop[,step])</code>	返回不可改列表
<code>repr(object)</code>	返回可打印对象信息
<code>round(number[,ndigits])</code>	四舍五入偏偶数
<code>set([iterable])</code>	转换为集合
<code>str(object)/(object,encoding,errors)</code>	转换为 str
<code>sum(iterable[,start])</code>	求和
<code>tuple([iterable])</code>	转换为 tuple
<code>type(object)</code>	返回数据类型

## 其他第三方模块

表达式模式支持部分常见 Python 第三方模块。

模块	特性	特性类型	特性描述	特性输出
time	<code>asctime([t])</code>	函数	struct_time 格式化	str
	<code>ctime([secs])</code>	函数	时间戳格式化	str
	<code>gmtime([secs])</code>	函数	生成 UTC 时区的 struct_time	struct_time
	<code>localtime([secs])</code>	函数	生成本地 struct_time	struct_time
	<code>mktime(t)</code>	函数	struct_time 生成时间戳	float

	strptime(format[,t])	函数	struct_time 定制化	str
	strptime(string[,format])	函数	字符串构建 struct_time	struct_time
	time()	函数	生成当前时间戳	float
	time_ns()	函数	生成当前时间戳 ( 纳秒 )	int
math	e	常量	自然对数	float
	pi	常量	圆周率	float
	sqrt(x)	函数	开方	float
	log(x[,base])	函数	对数	float
	ceil(x)	函数	向上取整	int
	floor(x)	函数	向下取整	int
	cos(x)	函数	余弦	float
	fabs(x)	函数	绝对值	float
	log2(x)	函数	以2为底对数	float
	log10(x)	函数	以10为底对数	float
	pow(x,y)	函数	指数	float
	sin(x)	函数	正弦	float
tan(x)	函数	正切	float	
json	dumps(obj, *, skipkeys=False, ensure_ascii=True, check_circular=True, allow_nan=True, cls=None, indent=None, separators=None, default=None, sort_keys=False, **kw)	函数	json 编码	str
	loads(s, *, encoding=None, cls=None, object_hook=None, parse_float=None, parse_int=None, parse_constant=None, object_pairs_hook=None, **kw)	函数	json 解码	any
base64	b64encode(s, altchars=None)	函数	base64 编码	bytes
	b64decode(s, altchars=None, validate=False)	函数	base64 解码	bytes
random	randint(a, b)	函数	[a,b]随机整数	int
	random()	函数	[0,1)随机浮点数	float
urllib	parse.quote(string, safe='/', encoding=None, errors=None)	函数	特殊字符转码	str
	parse.urlencode(query, doseq=False, safe="", encoding=None, errors=None)	函数	url 编码	float

# 代码模式 Python

最近更新时间：2025-07-03 17:43:01

代码模式 Python 对表达式模式进行了扩展，支持更复杂的语法和更强大的功能，另一方面，也提高了用户使用门槛，需要一定的 Python 语言编程基础。

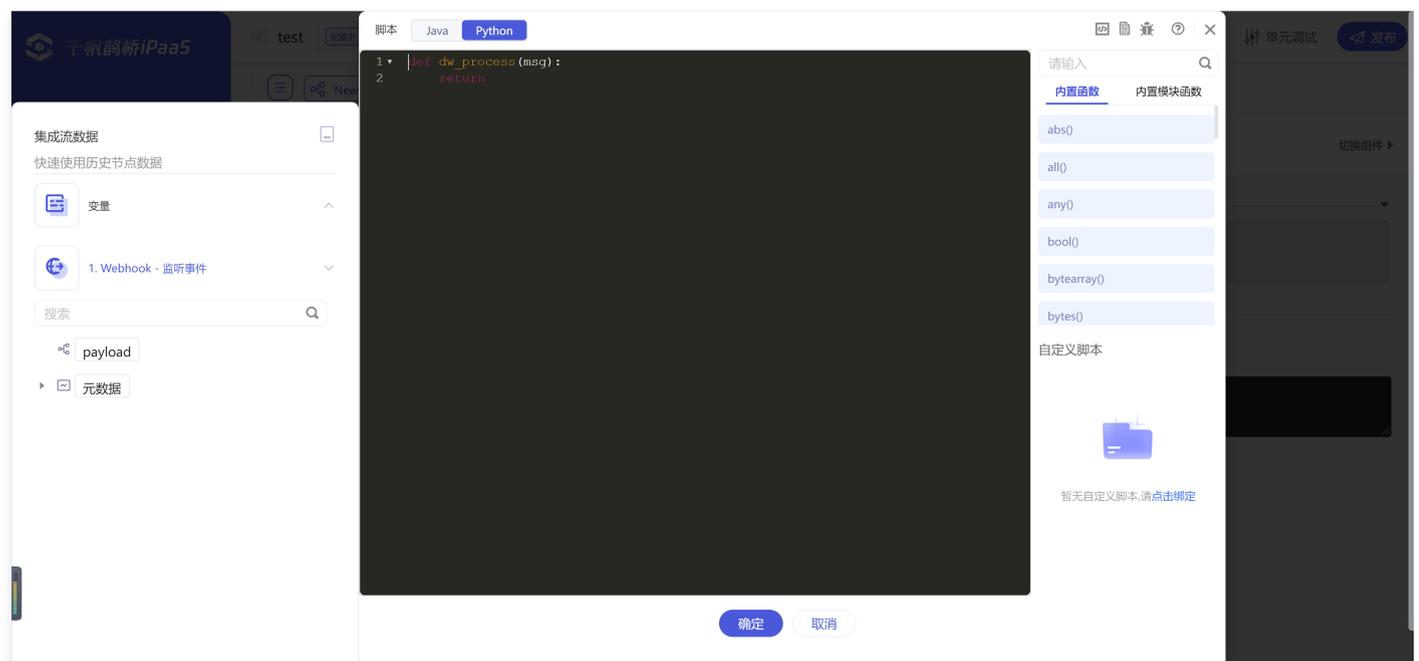
## IDE 使用

### 使用流程

1. 对于任意 Dataway 编辑文本框，将鼠标移至编辑文本框，会自动弹出模式选择按钮，单击**代码**进入代码模式。



2. 单击编辑文本框，弹出代码编辑器，默认即为 Python 脚本编辑器，即可进行 Python 脚本编辑。



3. 编辑完成后，单击**确定**保存。

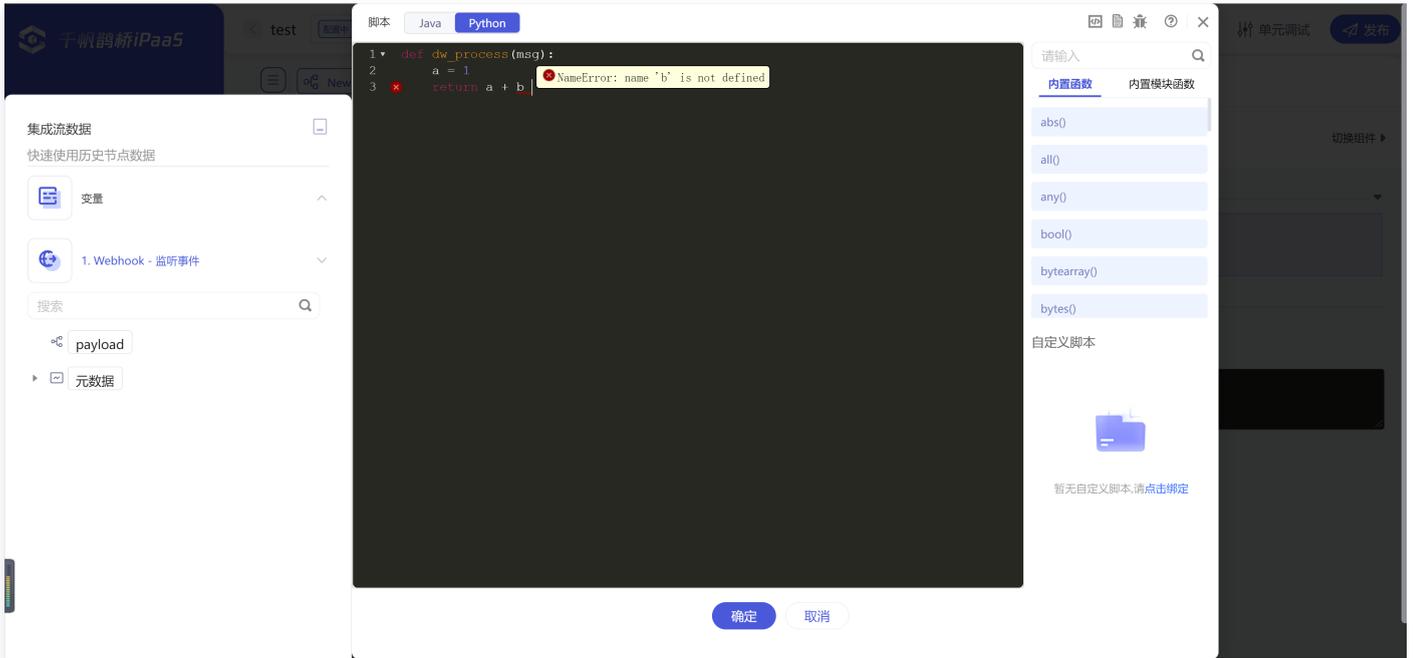
## 功能说明

该编辑器提供了语法检查、格式化、脚本调试、自动联想、代码高亮等类 IDE 功能。

### ● 语法检查

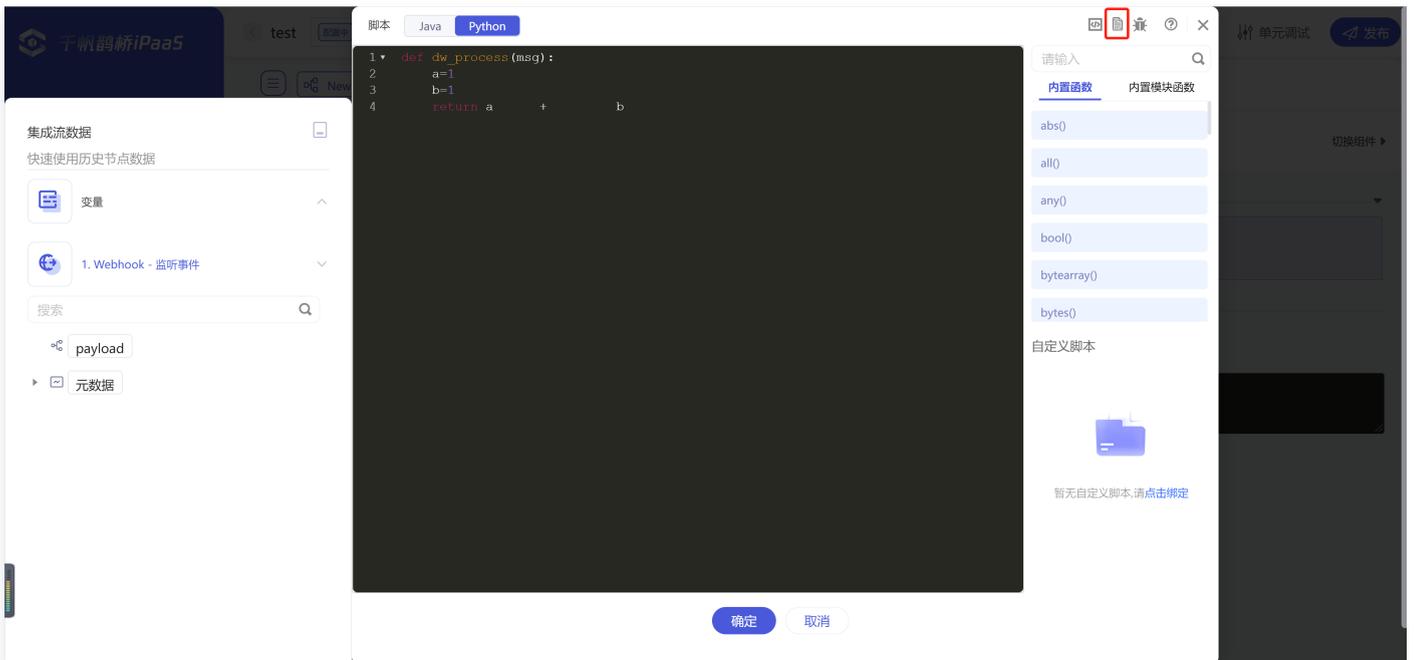
在 Python 脚本编辑器中，能够实时对 Python 脚本进行语法检查，并通过强提示显示在脚本编辑框左侧和出错代码底部。当鼠标移动到错误提醒处，会有详细的错误信息说明。

用户可以根据语法提示来对 Python 脚本进行修改，只有语法检查通过的代码才能够保存成功。

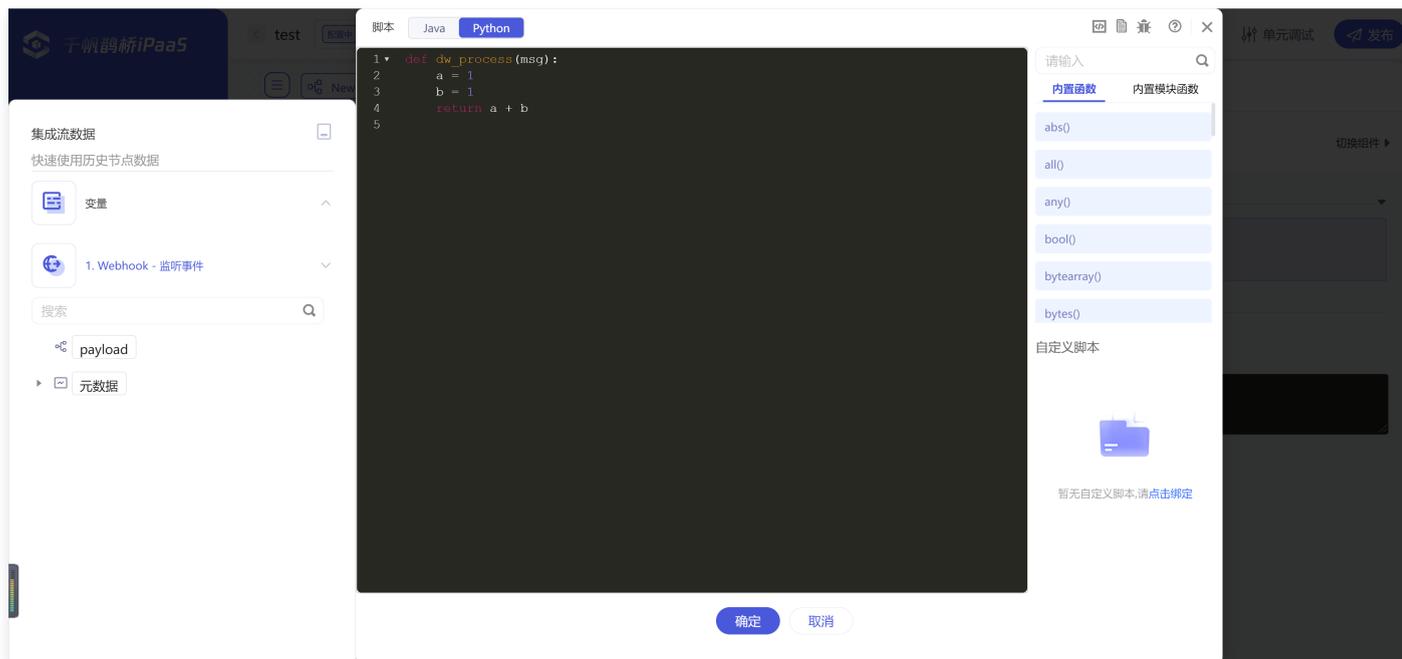


● 格式化

1.1 在 Python 脚本编辑器中，提供有格式化功能按钮。用户可以单击右上角格式化按钮，一键对 Python 脚本格式化，使代码更加简洁规范。

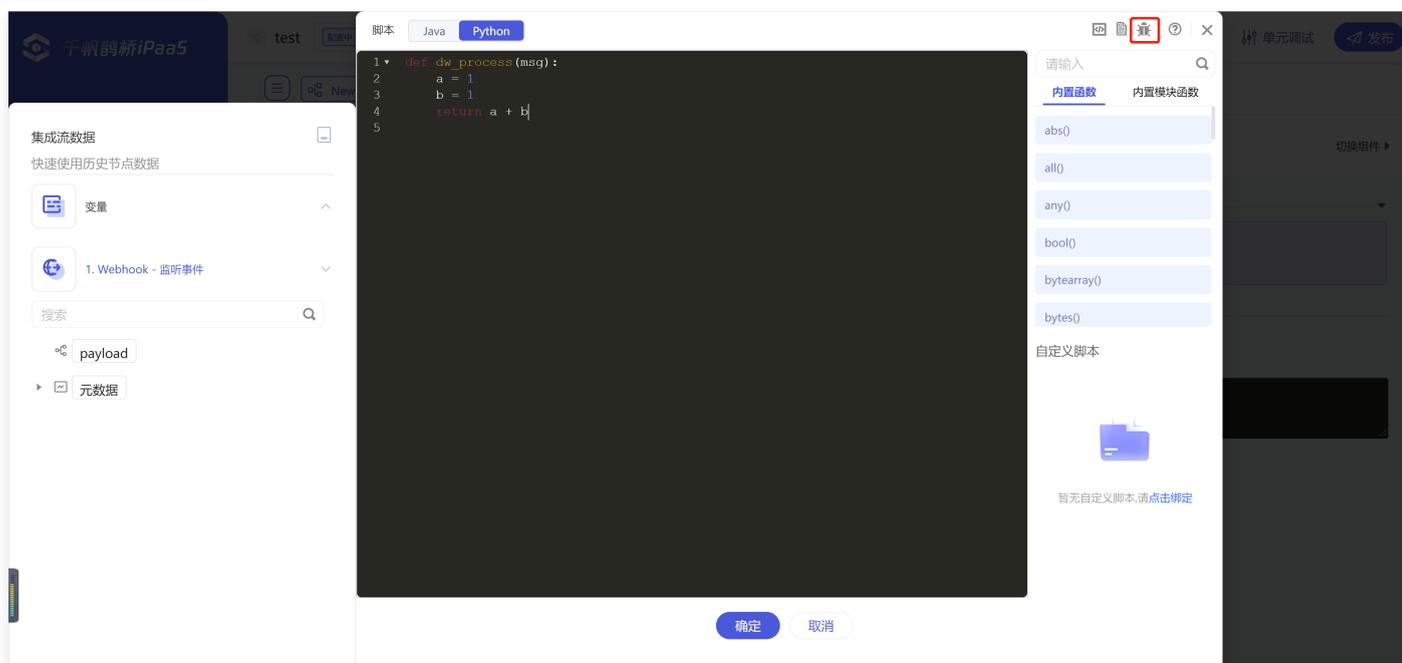


1.2 单击右上角格式化按钮后，格式化后的代码如下图。



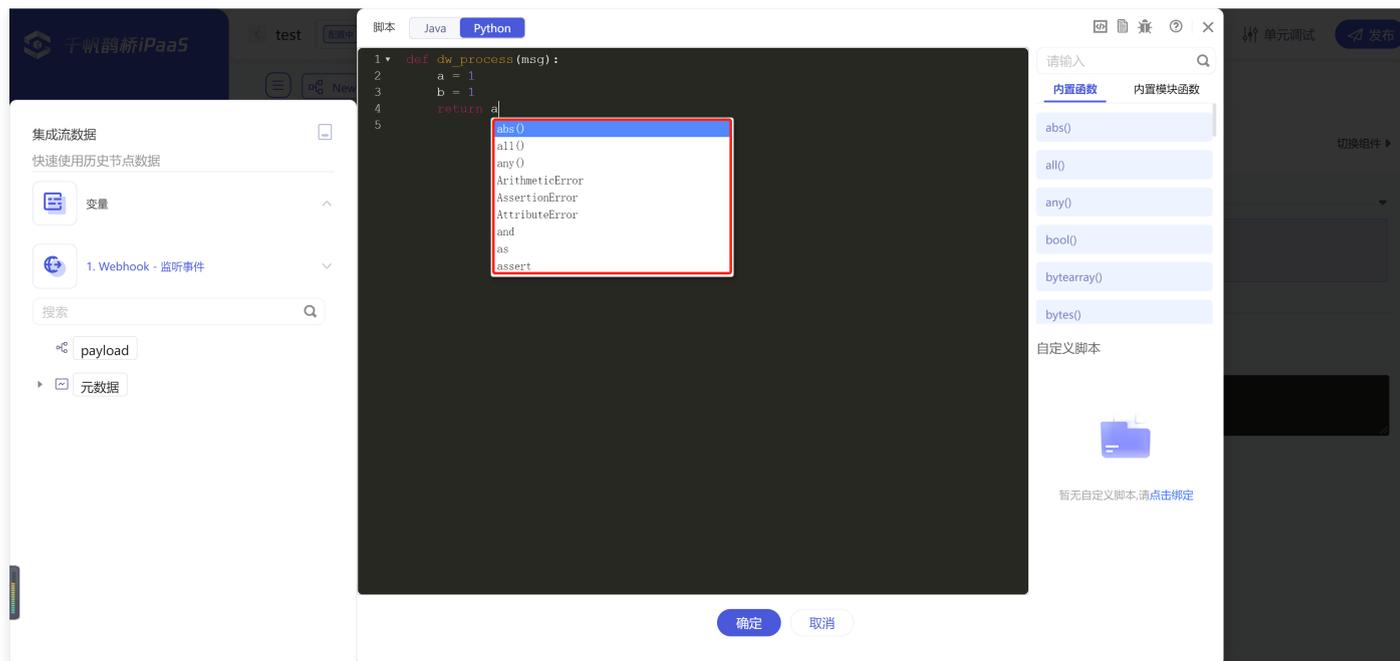
● 脚本调试

在 Python 脚本编辑器中，提供有 Debug 调试功能按钮。用户可以单击右上角 Debug 按钮，在线对 Python 脚本进行调试。详细的使用方法可见 [Python 调试](#)。



● 自动联想

在编辑框中进行输入时，Python 脚本编辑器能够根据当前上下文自动给出语法提示，并展示在当前光标的下方。通常语法提示的范围包括内置函数、关键字和内置第三方模块。



● 集成流数据面板引用

表达式模式支持集成流数据面板引用功能，详情请参见 [集成流数据面板](#)。

● 代码高亮

Python 脚本编辑器默认对 Dataway 代码进行高亮和括号自动匹配。

## 脚本结构

完整的代码模式 Python 脚本需符合 Python3语法，其中包含入口函数定义 def dw\_process(msg)，例如：

```
def dw_process(msg):
    sq = func(3)
    val = {
        'square': sq,
        'data': msg.payload['realData'] + 1
    }
    return Entity.from_value(val, mime_type='application/json')

def func(x):
    return x*x
```

dw\_process 入口函数仅接受一个参数 msg，该参数代表当前 Dataway 脚本需要处理的腾讯轻联消息。dw\_process 函数的返回值即是脚本的返回值。

在“配置 Payload”组件中输入上述表达式，假设该组件的输入消息为 json 结构的数据 {"realData": 123}，经过 Python 脚本的计算，得到的输出结果如下：

```
{
  "square": 9,
  "data": 124
}
```

## Dataway基本语法说明

代码模式 Python 基于 Python3 语法实现，本小节将对代码模式 Python 的基本语法进行说明。

## 关键字

代码模式 Python 支持的关键字如下表所示。关键字作为代码模式 Python 中的保留字，不会被当成任何标识符名称。

关键字名称	说明
True	布尔类型，True 表示真，相对于 False
False	布尔类型，False 表示假，相对于 True
None	空值类型
and	逻辑“与”
or	逻辑“或”
not	逻辑“非”
as	自定义命名
assert	断言，用来测试表达式
break	终止循环语句
continue	跳过本次循环
def	函数定义
if/else/elif	判断语句
for	循环语句
global	全局变量声明
in	判断是否包含在其中
is	判断两个变量指向是否一致
lambda	匿名函数，可以用一行实现一个函数
nonlocal	在嵌套函数中声明，可以修改外部定义的变量
pass	空语句，用于占位
raise	抛出异常
return	函数返回

## 行和缩进

代码模式 Python 使用缩进来标识代码块，不同的缩进行数代表不同的代码层级，同一层级的缩进行数需保持一致。

## 运算符

代码模式 Python 支持常见的运算符：算术运算符、比较运算符、赋值运算符、逻辑运算符、位运算符等。下表列出了常见的运算符。假设变量 a 为 5，b 为 3，对应的示例如下：

运算符名称	说明	示例
=	赋值	c = 3

+	加	$a + b = 3$
-	减	$a - b = 2$
*	乘	$a * b = 15$
/	除	$15 / a = b$
%	取模, 返回除法的余数	$16 \% b = 1$
**	幂	$a ** b = 125$
//	向下取整	$a // b = 1$
+=	加法赋值	$c += a$ 等效于 $c = c + a$
-=	减法赋值	$c -= a$ 等效于 $c = c - a$
*=	乘法赋值	$c *= a$ 等效于 $c = c * a$
/=	除法赋值	$c /= a$ 等效于 $c = c / a$
==	是否相等	$a == b$ 返回 False
!=	是否不等	$a != b$ 返回 True
>	大于比较	$a > b$ 返回 True
<	小于比较	$a < b$ 返回 False
>=	大于等于比较	$a >= b$ 返回 True
<=	小于等于比较	$a <= b$ 返回 False
&	按位与	$a \& b = 1$ (0101 & 0011 = 0001)
	按位或	$a   b = 7$ (0101   0011 = 0111)
^	按位异或	$a \wedge b = 6$ (0101 ^ 0011 = 0110)
~	按位取反	$\sim a = -6$
<<	左移运算符	$a \ll 3 = 20$ (0000 0101 << 3 = 0001 0100)
>>	右移运算符	$a \gg 1 = 2$ (0101 >> 1 = 0010)

## 条件及循环控制语句

- 代码模式 Python 通过 if、elif 或 else 语句来进行条件控制。示例如下，通过判断 a 的值，返回不同的字符串：

```
def dw_process(msg):
    a = 100
    if a < 10:
        return 'a is lower than 10'
    elif a <= 100 and a >= 10:
        return 'a is between 10 and 100'
    else:
        return 'a is bigger than 100'
```

Dataway表达式的运行结果为：`a is between 10 and 100`

- 代码模式 Python 通过 for 循环进行循环控制。示例如下，通过 for 循环，得到 a 中元素的乘积：

```
def dw_process(msg):
    a = [1, 2, 3, 4]
    num = 1
    for i in a:
        num *= i
    return num
```

Dataway 表达式的运行结果为: 24

## 定义函数

在代码模式 Python 中, 可以使用 `def` 关键词定义函数, 后接函数名和参数名列表, 以冒号 `:` 作为定义函数行的结尾, 下一行默认缩进; 最终以 `return` 语句结束函数, 如果不带 `return` 相当于返回 `None`。

定义了一个函数后, 可以在另一个函数中调用执行。在代码模式 Python 中, 默认的入口函数 `dw_process` 函数无需手工声明。如果想自定义函数, 直接在 `dw_process` 入口函数下方定义即可。如下示例, 定义了一个函数 `test()` 用于对列表元素求和, 并在 `dw_process()` 函数中调用, 最终使用 `return` 语句返回结果。

```
def dw_process(msg):
    a = [1, 2, 3, 4]
    return add_list(a)

def add_list(alist):
    sum = 0
    for i in reversed(alist):
        sum += i
    return sum
```

最终的输出结果为: 10

## 模块调用

代码模式 Python 内置了多个第三方模块, 如 `time`、`json`、`math`、`base64`、`hmac`、`random`、`hashlib`、`Crypto`、`socket`、`struct`、`decimal` 和 `datetime` 等, 使用时直接引用模块名即可, 无需使用 `import` 关键字。具体的函数说明请参见 [Dataway 函数参考](#)。具体示例如下, 接收一个 `json` 类型字符串, 转换成一个字典:

```
def dw_process(msg):
    jsonStr = '{"a": 1, "b": 2, "c": 3}'
    jsonDict = json.loads(jsonStr) # 转换成一个dict
    num = 1
    for k, v in jsonDict.items(): # 对dict进行遍历
        num += math.pow(v, 2)
    return num
```

最终的输入结果为: 15.0

## 注释

代码模式 Python 单行注释以 `#` 开头, 多行注释则可以用多个 `#` 号, 或者 `'''` 和 `"""`。举例如下, 执行下面代码:

```
# Dataway 注释

'''
Dataway 注释
Dataway 注释
```

```
'''
'''
Dataaway 注释
Dataaway 注释
'''
def dw_process(msg):
    return 'Dataaway Hello World!'
```

输出结果为:

```
Dataaway Hello World!
```

**说明:**

代码模式 Python 提供了语法检查功能，在编写代码时会进行实时语法检查，并给出错误提示。详细的语法说明可以参见 [Python 官方文档](#)。

## dw\_process 入口函数

dw\_process 是代码模式 Python 的主入口函数，其作用相当于 C/C++语言中的 main 函数。  
dw\_process 仅接受一个类型为 [Message](#) 的参数，而其返回值就是该代码模式 Python 脚本的输出值。  
作为腾讯轻联中数据处理流程的一个环节，dw\_process 函数的返回值目前支持 [核心类型](#)。  
关于代码模式 Python 中数据类型及返回值的详细介绍，可参见 [DataWay 数据类型系统](#)。

## 数据类型系统

类型名	说明	是否 Dataaway 特有类型	举例
str	字符串，即 Python 原生的字符串 str	否	"abc"
None	Python 中的空值 None	否	None
bool	布尔值，即 Python 原生布尔值 bool	否	True/False
float	浮点数，即 Python 原生浮点数 float	否	123.123
int	整数，即 Python 原生整型 int	否	123
bytes	字节数组，即 Python 字节数组类型 bytes	否	b'this_is_a_bytes'
set	集合，即 Python 集合类型 set	否	{1,2,3}
list	列表，序列类型容器，即 Python 原生 list 类型	否	[1,2,3]
dict	字典，kv 类型容器，即 Python 原生 dict 类型	否	{1:1, 'key': 'value'}
Entity	即 EIS 中的实体数据，用于代表一个二进制对象，在 Dataaway 中以 Entity 类型进行访问，包括 blob、mime_type、encoding 等信息	是	http-listener 构造消息中的 payload，例如 msg.payload
MultiMap	多值 map，类似于 xml 而与 dict 不同，该类型可以支持重复的 key。	是	application/www-form-urlencoded 格式的数据解析之后得到的对象
FormDataParts	数组+列表的数据结构，类似于 Python 中的 orderDict 结构	是	multipart/form-data 格式的数据解析后得到的对象

Message	即 eis 中的消息，在 dataway 中以 Message 进行访问	是	dw_process 入口函数中的 msg 参数
---------	--------------------------------------	---	--------------------------

**注意：**

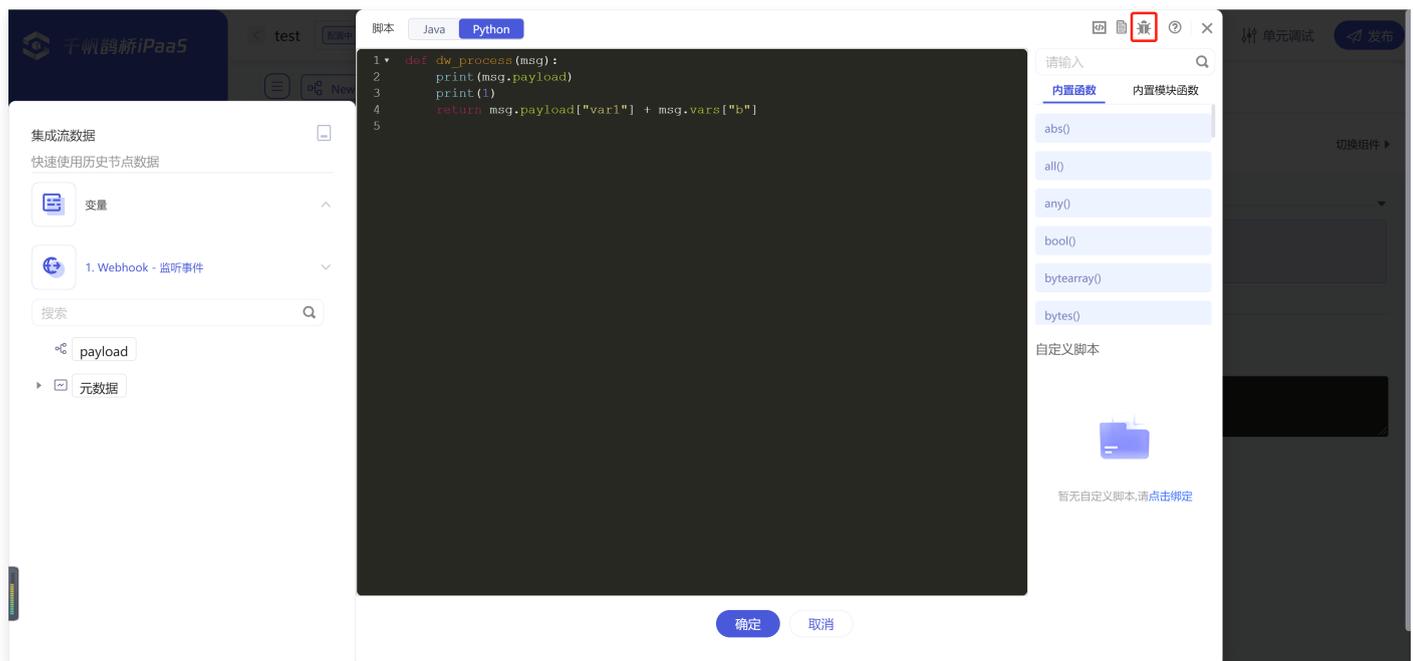
- 上述类型可以在代码模式 Python 中使用，但 dw\_process 函数的返回值的类型为 **核心类型** 之一。
- 如果 Dataway 表达式输出的值会作为集成流的最终返回结果，则支持的返回值类型还会受到相应连接器组件的限制。例如在以 http listener 组件作为第一个组件的流中，其最终的 payload 也需要是一个 Entity 类型。

## 脚本调试

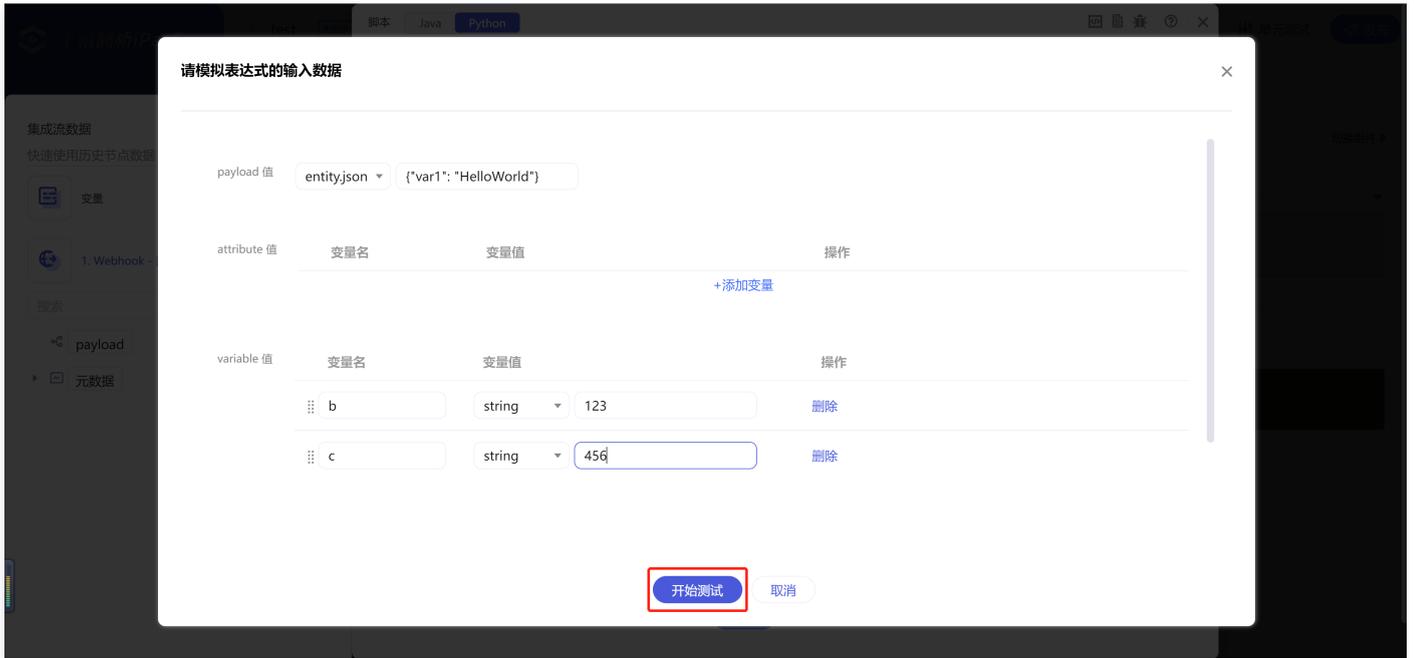
代码模式 Python 支持脚本调试，以方便问题排查和结果验证。该功能可通过手工定义输入参数 msg，单击测试后可以直接查看脚本运行结果、调试日志和错误信息。

1. 在脚本编辑框中输入表达式。

Dataway 调试模式下，支持在表达式中通过 print() 函数打印要观察的信息，运行结束后打印消息会显示在界面上。

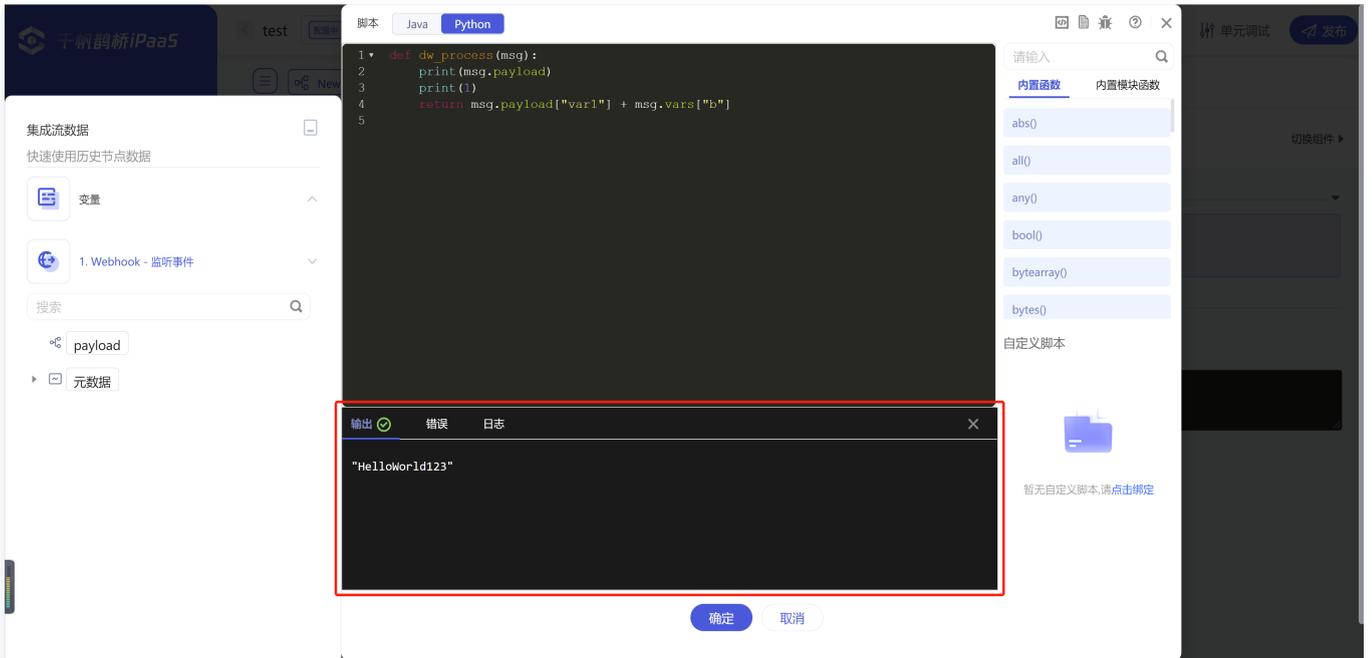


2. 单击脚本编辑框右上角的 Debug 图标，弹出模拟数据填写对话框，在这里可以对 msg 的载荷、属性和变量进行设置。设置完成后单击**开始测试**，系统会自动组装成一个 msg 参数，并作为脚本的输入传递到 dw\_process 函数中。

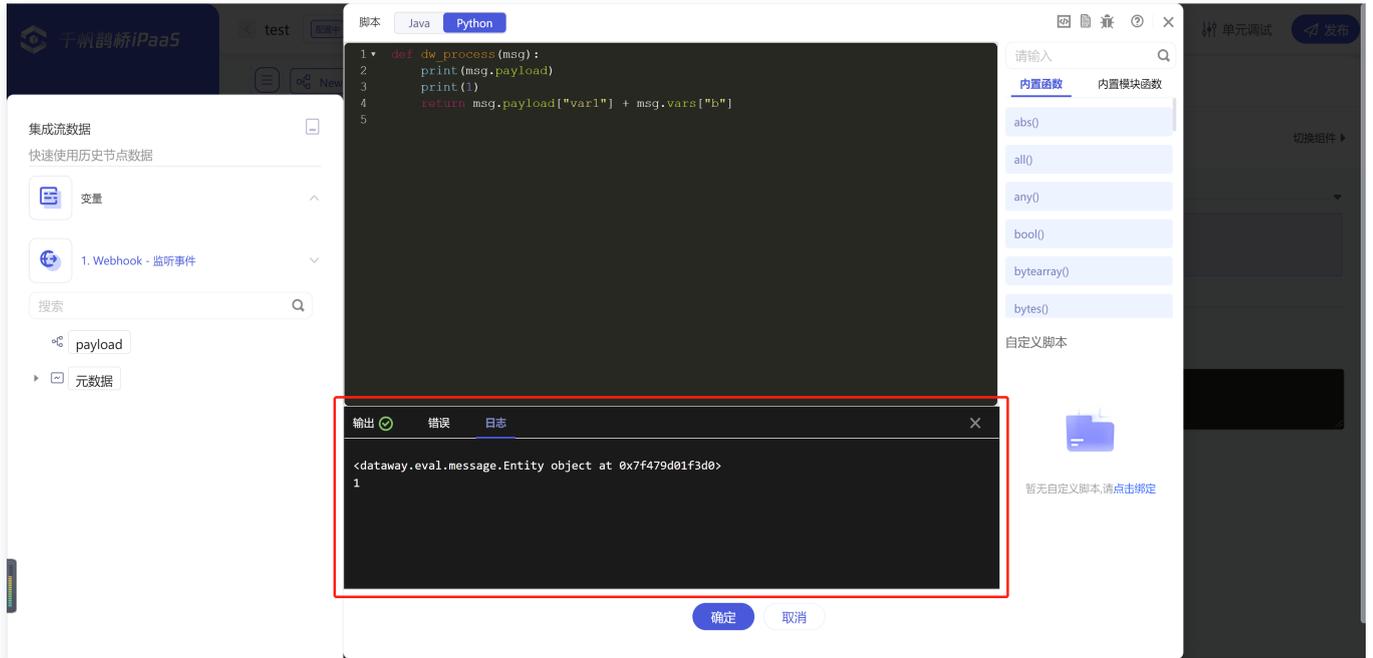


3. 运行完成 dw\_process 函数后，编辑框下方会弹出运行结果和 print 调试日志，如果运行错误会有 error 报错信息。

- 输出：代表 Dataway 表达式的运行结果。



- 日志：代表在脚本中使用 print 函数打印的调试日志。



- **错误:** 代表脚本运行错误，运行正确无错误显示绿色对勾标志。

## 其他支持

代码模式 Python 提供多样的内置函数和第三方模块，用户可以按需选用，快速实现既定功能，详情请参见 [代码模式 Python 附录](#)。

# Python 附录

最近更新时间：2024-11-25 10:11:13

## 内置函数

目前代码模式 Python 支持的内置函数如下：

序号	内置函数	功能说明
1	abs()	求数值绝对值
2	all()	判断序列（集合、列表、元组、dict）中所有元素是否满足给定条件
3	any()	判断集合中是否存在元素满足给定条件
4	bool()	构造布尔值
5	bytearray()	构造字节数组
6	bytes()	构造空字节
7	chr()	0~256的整数对应的ASCII码
8	dict()	创建字典
9	enumerate() ( )	将一个可遍历的数据对象组合列出数据和数据下标，一般用于 for 循环中
10	filter()	集合过滤，例如 list(filter(lambda x:x>=100, [1,3,4,100,102])) -> [100,102]
11	float()	构造浮点数
12	getattr()	求一个对象的属性值
13	hasattr()	判断一个对象是否有某个属性
14	hash()	求哈希值
15	id()	求对象的唯一标识
16	int()	构造整数
17	isinstance() ( )	判断对象是否属于某种类型
18	iter()	生成一个迭代器
19	len()	获取集合元素个数
20	list()	构造列表
21	map()	根据函数对指定序列做映射，例如 list(map(lambda x: x * 2, [1, 2, 3, 4, 5])) -> [2, 4, 6, 8, 10]
22	max()	获取数值最大值
23	min()	数值最小值
24	next()	返回迭代器的下一个项目，和 iter() 一起使用
25	objects()	返回空对象

26	ord()	单个 ASCII 码字符的整数值
27	pow()	求指数
28	print()	代码模式 Python 调试时可以打印相关信息（仅在 Dataway 表达式编辑时使用 Debug 功能生效）
29	range()	创建可迭代对象，例如 list(range(5)) -> [0, 1, 2, 3, 4]
30	reversed()	创建反转的迭代器，例如 list(reversed('abcdefg')) -> ['g', 'f', 'e', 'd', 'c', 'b', 'a']
31	round()	截取数值的整数部分
32	set()	创建一个集合
33	slice()	设置截取元素的切片
34	sorted()	排序
35	str()	构造字符串
36	sum()	数值求和
37	tuple()	构造元组
38	type()	返回对象类型
39	zip()	打包可迭代对象中元素成多个 tuple。例如 list(zip([1,2,3], [4,5,6])) -> [(1, 4), (2, 5), (3, 6)]

## 第三方模块

### time

time 用于时间处理的库，可参见 [Python 官方文档](#)。已内置代码模式 Python，可以直接引用。

目前代码模式 Python 支持的库函数/类型如下：

序号	库函数/类型	功能说明
1	altzone	当前时区相对于 UTC 时区的延迟偏移，单位为秒
2	asctime	将一个 struct_time 转换为时间字符串
3	ctime	将一个时间戳转换为时间字符串
4	mktime()	将一个 struct_time 转换为时间戳
5	strftime()	将一个 struct_time 进行格式化
6	strptime()	按照给定的格式解析事件字符串，并返回一个结构化的 struct_time 对象
7	timezone	当前时区
8	tzname	当前时区名称
9	time()	当前时间
10	localtime	将一个时间戳转换为当前时区的本地时间，返回 struct_time 类型对象

### json

json 用于处理 json 数据的库，可参见 [Python 官方文档](#)。已内置代码模式 Python，可以直接引用。

目前代码模式 Python 支持的 json 模块函数：

序号	json 模块函数	功能说明
1	<code>dumps()</code>	将 json 对象编码为 json 字符串
2	<code>loads()</code>	将一个 json 串解析为 Python 对象

## math

math 用于数学运算的库，可参见 [Python 官方文档](#)。已内置代码模式 Python，可以直接引用。

目前代码模式 Python 支持的 math 模块函数：

序号	json 模块函数	功能说明
1	<code>math.ceil(x)</code>	返回 x 的上限，即大于或者等于 x 的最小整数。如果 x 不是一个浮点数，则委托 <code>x.ceil()</code> ，返回整数值。
2	<code>math.floor(x)</code>	返回 x 的向下取整，小于或等于 x 的最大整数。如果 x 不是浮点数，则委托 <code>x.floor()</code> ，返回整数值。
3	<code>math.fabs(x)</code>	返回 x 的绝对值。
4	<code>math.pow(x,y)</code>	返回 x 的 y 次幂。
5	<code>math.sqrt(x)</code>	返回 x 的平方根。

支持的常量：

序号	常量	功能说明
1	<code>math.pi</code>	数学常数 $\pi = 3.141592\dots$ ，精确到可用精度。
2	<code>math.e</code>	数学常数 $e = 2.718281\dots$ ，精确到可用精度。
3	<code>math.inf</code>	浮点正无穷大。（对于负无穷大，使用 <code>-math.inf</code> ）相当于 <code>float('inf')</code> 的输出。
4	<code>math.nan</code>	浮点“非数字”（NaN）值。相当于 <code>float('nan')</code> 的输出。

## base64

base64 用于 base64 编解码的库，可参见 [Python 官方文档](#)。已内置代码模式 Python，可以直接引用。支持的函数有：

序号	支持的函数	功能说明
1	<code>base64.b64encode(s)</code>	对 bytes 类型参数 base64 编码，返回编码后的 bytes。
2	<code>base64.b64decode(s)</code>	对 bytes/str 类型参数 base64 解码，返回解码后的 bytes。

## hmac

hmac 用于 hmac 加解密的库，可参见 [Python官方文档](#)。已内置代码模式 Python，可以直接引用。支持的函数有：

序号	支持的函数	功能说明
1	<code>hmac.new(key)</code>	返回一个新的 hmac 对象，key 是一个指定密钥的 bytes 或 bytearray 对象。

## random

random 用于随机数生成的库，可参见 [Python 官方文档](#)。已内置代码模式 Python，可以直接引用。支持的函数有：

序号	支持的函数	功能说明
1	<code>random.randint(a,b)</code>	返回随机整数 N 满足 $a \leq N \leq b$ 。

## hashlib

hashlib 用于生成哈希值的库，可参见 [Python 官方文档](#)。已内置代码模式 Python，可以直接引用。支持的函数/属性有：

序号	支持的函数/属性	功能说明
1	<code>hashlib.sha256()</code>	创建一个 SHA-256 hash 对象。
2	<code>hashlib.md5()</code>	创建一个 MD5 的 hash 对象。
3	<code>hashlib.sha1()</code>	创建一个 SHA1 的 hash 对象。

## datetime

datetime 用于时间日期处理的库，可参见 [Python 官方文档](#)。已内置代码模式 Python，可以直接引用。支持的函数/属性有：

序号	支持的函数/属性	功能说明
1	<code>datetime.date</code>	一个理想化的简单型日期，它假设当今的公历在过去和未来永远有效。属性有：year、month 和 day。
2	<code>datetime.time</code>	一个理想化的时间，它独立于任何特定的日期，假设每天一共有 246060 秒。属性有：hour、minute、second、microsecond 和 tzinfo。
3	<code>datetime.datetime</code>	日期和时间的结合。属性有：year、month、day、hour、minute、second、microsecond 和 tzinfo。
4	<code>datetime.timedelta</code>	表示两个 date 对象或者 time 对象，或者 datetime 对象之间的时间间隔，精确到微秒。
5	<code>datetime.timezone</code>	表示相对于世界标准时间（UTC）的偏移量。
6	<code>datetime.tzinfo</code>	描述时区信息对象，用来给 datetime 和 time 类提供自定义的时间调整概念，如处理时区和/或夏令时。

## decimal

decimal 用于定点数处理的库，可参见 [Python 官方文档](#)。已内置代码模式 Python，可以直接引用。支持的函数/属性有：

序号	支持的函数/属性	功能说明
1	<code>decimal.Decimal</code>	表示十进制浮点数对象。

## socket

socket 是 Python 中 tcp 套接字的底层实现，可参见 [Python 官方文档](#)。已内置代码模式 Python，可以直接引用。支持的函数/属性有：

序号	支持的函数/属性	功能说明
1	<code>socket.htonl()</code>	将 32 位正整数从网络序转换成主机字节序。
2	<code>socket.ntohl()</code>	将 32 位正整数从网络字节序转换为主机字节序。

## pycryptodome

pycryptodome 是专用的加密工具三方库，可参见 [pycryptodome 官方文档](#)。已内置代码模式 Python，可以直接引用。支持的函数/属性有：

序号	支持的函数/属性	功能说明
1	Crypto.Util.Padding	该模块为添加和删除数据中的标准填充提供了最小的支持, 提供 pad()、unpad() 方法。
2	Crypto.Cipher.AES	AES 加密模块。它具有16字节的固定数据块大小, 密钥可以是128、192或256位长。提供 new() 方法。

## struct

struct 用于打包二进制文件的库, 可参见 [Python 官方文档](#)。已内置代码模式 Python, 可以直接引用。支持的函数/属性有:

序号	支持的函数/属性	功能说明
1	struct.pack(format, v1, v2, ...)	返回一个 bytes 对象, 其中包含根据格式字符串 format 打包的值 v1, v2, ... 参数个数必须与格式字符串所要求的值完全匹配。
2	struct.unpack(format, buffer)	根据格式字符串 format 从缓冲区 buffer 解包, 返回一个 tuple。

## urllib

urllib 用于 URL 处理的库, 可参见 [Python 官方文档](#)。已内置代码模式 Python, 可以直接引用。支持的函数/属性有:

序号	支持的函数/属性	功能说明
1	urllib.parse.urlparse()	获取 url 参数, 将 url 解析成一个包含6个字符串的 tuple, 分别为: 协议, 位置, 路径, 参数, 查询, 片段识别。
2	urllib.parse.unquote()	将编码的 url 进行解码。

## CSV

csv 用于 CSV 文件读写的库, 可参见 [Python 官方文档](#)。已内置代码模式 Python, 可以直接引用。支持的函数/属性有:

序号	支持的函数/属性	功能说明
1	csv.reader()	创建一个 reader对象, 将逐行遍历 csv 文件对象。

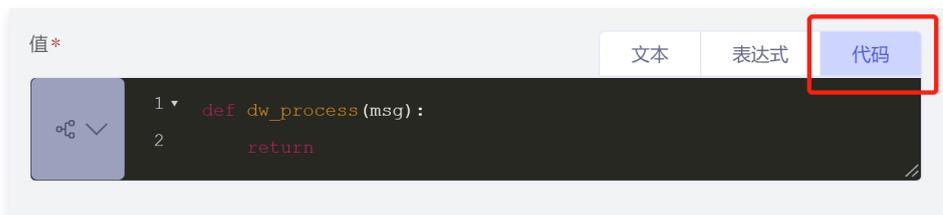
# 代码模式 Java

最近更新时间：2024-08-01 15:33:02

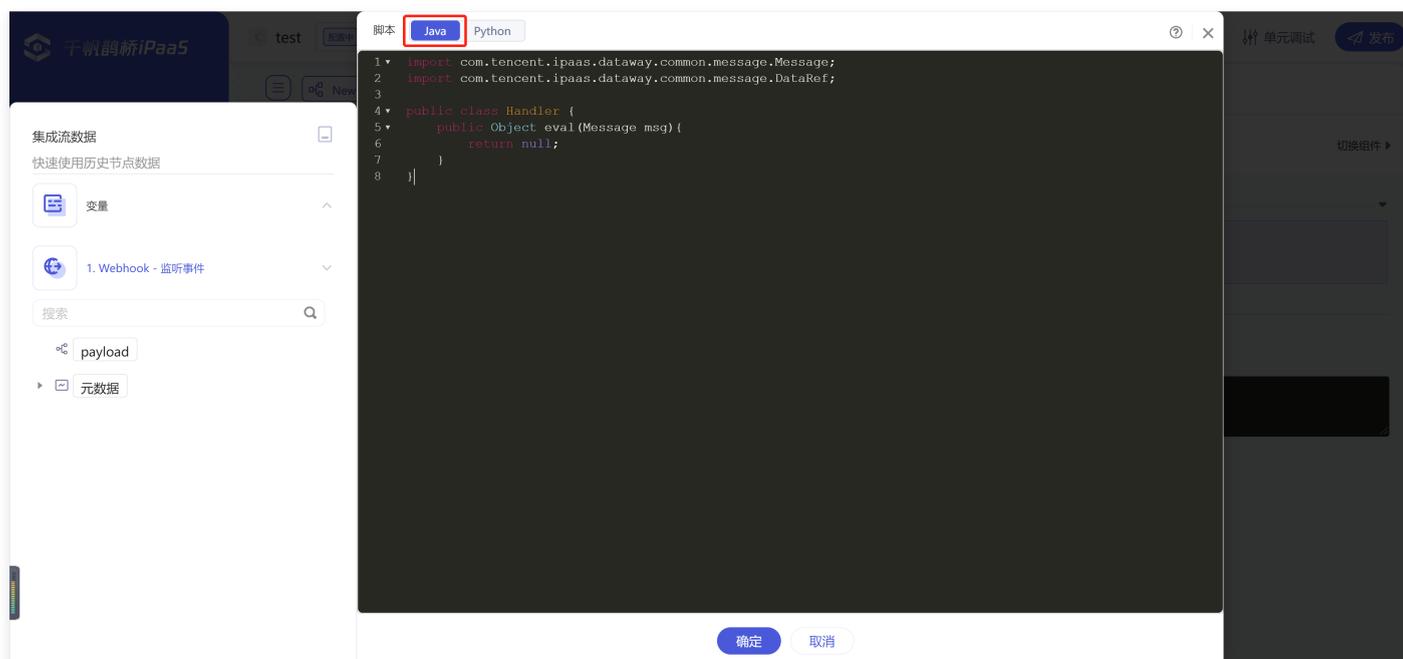
为了降低熟练使用 Java 的用户接入腾讯云腾讯轻联的门槛，Dataway 代码模式提供对 Java 脚本的支持。

## IDE 使用

1. 对于任意 Dataway 编辑文本框，将鼠标移至文本框，会自动弹出模式选择按钮，单击**代码**进入代码模式。



2. 单击编辑文本框，弹出代码编辑器，单击**Java**，进入 Java 脚本编辑器。



3. 编辑完成后，单击**确定**进行保存。

代码模式 Java 支持 [集成流数据面板](#) 引用。

## 脚本结构

Java 脚本需符合 JDK8 语法规定。

类名必须为 Handler，且必须定义一个签名为 Object eval ( Message msg ) 的函数作为入口函数。

```
import com.tencent.ipaas.dataway.common.message.Message;
import com.tencent.ipaas.dataway.common.message.DataRef;
```

不能删除，其它 import 语句可以根据类型使用需求在下面新增。

```
// 这行 import 固定，不能删除
import com.tencent.ipaas.dataway.common.message.Message;
```

```
import com.tencent.ipaas.dataway.common.message.DataRef;

/**
 * dataway-java入口类, 名称必须为 Handler
 */
public class Handler {
    /**
     * 入口函数, 签名必须为 Object eval(Message msg)
     * @param msg 输入 Message 对象
     * @return 任意 dataway 支持的数据对象
     */
    public Object eval(Message msg) {
        return msg.getPayload();
    }
}
```

## 数据类型

代码模式 Java 对各数据类型提供充足支持，方便用户对各数据类型进行操作。

类型名	类型描述	对应 Python 类型	类型举例	类型方法
null	Java 中的空值 null	None	null	略
String	字符串，即 Java 原生的字符串 String	str	“abc”	略
Boolean	布尔值，即 Java 原生布尔值 bool	bool	true/false	略
float/Float	浮点数，即 Java 原生浮点数 float	float	123.456	略
int/Integer	整数，即 Java 原生整型 int	int	123	略
long/Long	长整数，即 Java 原生长整型 Long		123L	略
short/Short	短整数，即 Java 原生短整型 Short		123	略
byte[]	字节数组，即 Java 字节数组类型 byte[]	bytes	byte[]{1,2,3}	略
java.util.List	列表，序列类型容器，即 Java 原生 List 类型	list	new java.util.ArrayList<>()	详情请参见 <a href="#">官方文档</a>
java.util.Map	字典，kv 类型容器，即 Java 原生 Map 类型	dict	new java.util.HashMap<>()	详情请参见 <a href="#">官方文档</a>
java.time.OffsetDateTime	时间，即 Java 原生的 OffsetDateTime	datetime.datetime	java.time.OffsetDateTime.now()	详情请参见 <a href="#">官方文档</a>
java.time.LocalDate	日期，即 Java 原生	datetime.date	java.time.LocalDate	详情请参见 <a href="#">官方文档</a>

	的 LocalDate	e	alDate.now()	
java.time.OffsetTime	时钟，即 Java 原生的 OffsetTime	datetime.time	java.time.OffsetTime.now()	详情请参见 <a href="#">官方文档</a>
java.math.BigDecimal	十进制数，即 Java 原生的 BigDecimal	decimal.Decimal	new java.math.BigDecimal("1")	详情请参见 <a href="#">官方文档</a>
com.tencent.ipaas.dataway.common.message.Entity (腾讯轻联特有类型)	即腾讯轻联中的实体数据，用于代表一个二进制对象，以 Entity 类型进行访问，包括blob、mimeType、encoding 等信息	Entity	HTTP-listener 构造消息中的 payload	详见“使用 Entity 对象”
com.tencent.ipaas.dataway.common.message.Multimap (腾讯轻联特有类型)	多值 map，类似于 xml 而与 dict 不同，该类型可以支持重复的 key，继承自 HashMap<String, List>	MultiMap	application/www-form-urlencoded 格式的数据解析之后得到的对象	构建方法: Multimap(Map dict)
				构建静态方法: Multimap fromSetEntry(Set<Map.Entry> set)
				获取特定键对应的第一个值: Object getFirst(Object key)
				获取特定键对应的所有值组成的列表: List getAll(Object key)
				获取键值对集合: Set<Map.Entry> toSetEntry()
				继承自 HashMap 的方法
com.tencent.ipaas.dataway.common.message.FormDataParts (腾讯轻联特有类型)	数组+列表的数据结构，类似于 Python 中的 orderDict 结构，继承自 LinkedHashMap<String, Object>	FormDataParts	multipart/form-data 格式的数据解析后得到的对象	构建方法: FormDataParts(String boundary)
				获取特定关键词对应值 (当关键词为 int /long时，关键词作为键的序号；其余情况，关键词即为键): Object get(Object key)
				继承自 LinkedHashMap 的方法
com.tencent.ipaas.dataway.common.message.Message (腾讯轻联特有类型，不可通过 dataway 构建)	即腾讯轻联中的集成流消息，以 Message 类型进行访问	Message	public Object eval(Message msg) 入口函数中的 msg 参数	获取 payload: Object getPayload()
				获取 Attributes: Map getAttrs()
				获取 Variables: Map<Object, Object>

				getVars() 或者 Object getVar(String name)
				获取 Error: DataWayError getError()
com.tencent.ipaas.dataway.common.message.DataSet (腾讯轻联特有类型, 不可通过 dataway 构建)	数据集中的数据集, 通过数据集成组件生成	RecordSet	Builder 组件的输出	获取 id: Long getId()  获取 schema: Schema getSchema()  获取 partitions: Long getPartitions()
com.tencent.ipaas.dataway.common.message.Record (腾讯轻联特有类型, 不可通过 dataway 构建)	数据集中的单条数据, 附有 schema	Record	可通过 Foreach 组件遍历 DataSet 获取	获取所有数据: List getData()  获取 schema: Schema getSchema()  获取特定关键词对应数据 (当关键词为 int / long 时, 关键词为列序号; 其余情况, 关键词为字段名): Object get(Object key)  是否包含字段名: boolean contains(String name)  获取字段名迭代器: Iterator<String> getIterator()
com.tencent.ipaas.dataway.common.message.Schema (腾讯轻联特有类型, 不可通过 dataway 构建)	数据集中的数据字典, 描述数据的元信息	Schema	可通过 Record 的 getSchema() 方法获取, 对应于各项数据	获取所有字段信息: RecordField[] getFields()  获取特定字段名对应字段信息: RecordField getField(String name)  转化为类似 {"Fields": [{"Name": "name", "Type": "string"}]} 的字典形式: Map<?, ?> toMap()
com.tencent.ipaas.dataway.common.message.RecordField (腾讯轻联特有类型, 不可通过 dataway 构建)	数据集中的字段信息, 描述单个字段的元信息	Schema	可通过 Schema 的 getField(name) 方法获取 RecordField	获取字段名: String getName()  获取字段类型: String getType()

## 使用 Entity 对象

### 基础方法

在代码模式 Java 中, 用 Entity 类型来表示腾讯轻联集成流中的实体数据, 表示二进制数据的封装对象, 其主要组成部分包括 blob、mimeType 以及 encoding。

组成部分	说明
blob	原始的二进制数据。

contentType	表示二进制数据的内容格式，例如：application/json、application/www-form-urlencoded
encoding	表示二进制数据的字符编码格式，例如：utf-8、gbk 等。

可以通过如下方式访问 Entity 中的内容：

访问方式	说明
byte[] getBlob()	获取该消息对象的负载数据，返回 byte[] 类型的对象。
String getMimeType()	获取该消息对象的 MIME 类型，返回 String 对象。
String getEncoding()	获取该消息对象的编码类型，返回 String 对象。
Object getValue()	根据 MIME 类型和编码类型，对负载数据 blob 进行反序列化，并返回其结果，该类型为代码模式 Java 类型系统中定义的类型之一。
Object get(Object key)	根据 MIME 类型和编码类型对 message 的内容进行反序列化，并获取其中指定 key 的值。

当前反序列化支持的 MIME 类型及其反序列化后的值类型如下：

- text/plain → String
- application/json → Object，与 GSON 一致
- application/x-www-form-urlencoded → Multimap
- application/xml → Map
- application/csv → List<Map<String,String>>，即字段名与取值映射的列表
- multipart/form-data → FormDataParts

## 构造方法

### Entity.fromValue 静态方法

用于将值类型 data 封装为 Entity 类型，并返回。如下所示：

```
Entity.fromValue(Object value, String mimeType, String encoding)
```

在 fromValue 函数内部，会先根据给定的 MIME 类型和编码类型尝试对 value 进行序列化得到 byte[] 类型的数据，然后再封装为 Entity 类型进行返回。

- mimeType 参数必传，且目前支持的 MIME 类型有 text/plain、application/json、application/x-www-form-urlencoded、application/csv、application/xml 和 multipart/form-data 六种。
- encoding 参数必传，且支持任意合法的编码类型。

### Entity.fromBytes 静态方法

用于将 String 或者 byte[] 类型数据封装为 Entity 类型，并返回。如下所示：

```
Entity.fromBytes(Object data, String mimeType, String encoding)
```

fromBytes 函数中的 MIME 类型和编码类型参数校验规则与 fromValue 函数相似，不同的是不会对 MIME 类型的参数值进行限制，可以为任意的 MIME 类型。

- 如果传递给 fromBytes 函数的 data 参数类型为 byte[] 类型，该函数会直接返回一个以 data、mimeType、encoding 作为参数构造的 Entity 对象。
- 如果传递的 data 是 String 数据，则会尝试根据 encoding 参数将其编码为 byte[] 类型，并构造 Entity 对象。

## 使用限制

Entity 对象在本质上是一个对二进制数据的封装对象，为方便使用，提供 `entity.get()` 等访问对象内容的方法。在使用这些功能时，需要注意的是，有些特殊操作下系统会尝试先对 Entity 中的二进制数据进行反序列化解析，如果解析失败则会导致运行时错误。这些特殊操作包括：

- 用 `entity.getValue()` 来获取解析后的结构化结果。
- 用 `entity.get(key)` 来获取结构化结果中的某项内容。

在不符合条件的 Entity 对象上执行上述特殊操作，将会导致运行时错误。

# 场景案例

最近更新时间：2024-08-01 15:33:02

本文通过几个典型的场景案例说明 Dataway 在腾讯云腾讯轻联系统中应用。

## 使用 Dataway 获取 token

腾讯云腾讯轻联对 [企业微信 API](#) 进行了封装，即**企业微信连接器**，用户可以通过可视化配置轻松完成成员管理、消息推送等多种操作。

连接器隐藏了许多复杂的数据处理和 API 调用，例如：业务 API 都依赖 `access_token` 登录凭证来鉴权调用者身份，因此，在调用 API 前，都需要先获取 `access_token`。

本示例将模拟 `access_token` 获取过程，并说明 Dataway 脚本在其中起到的作用。

### 配置流程

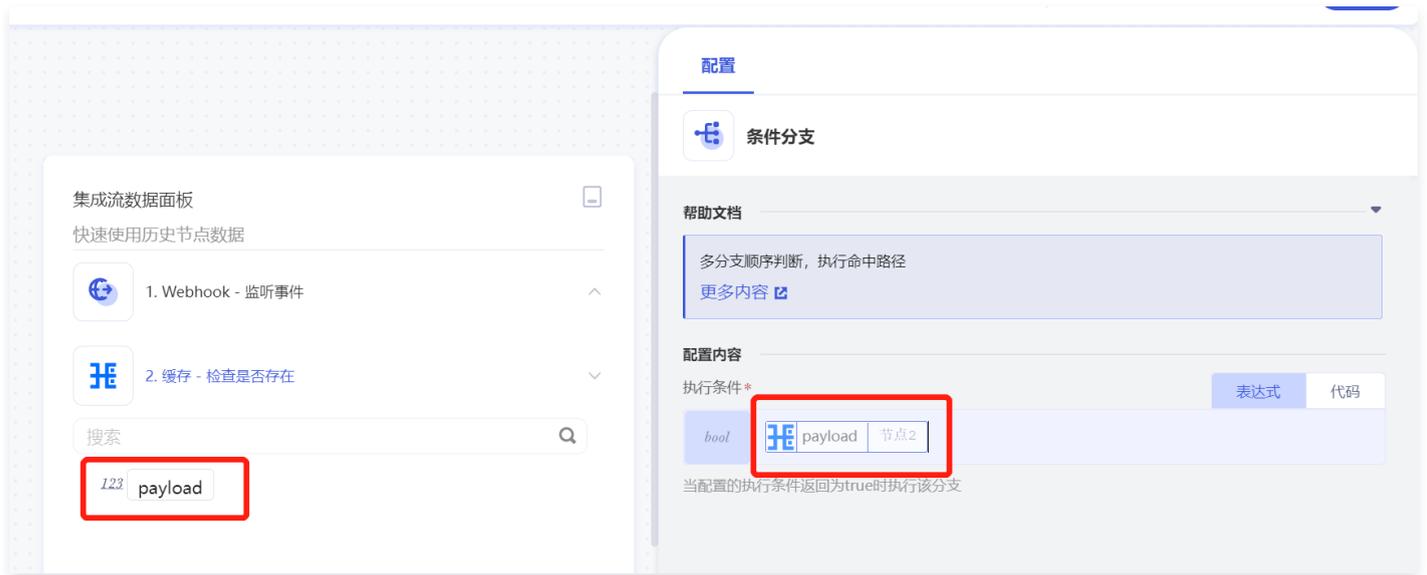
鉴于业务 API 都依赖 `access_token`，`access_token` 必然会频繁使用，因此，在调用 API 获取 `access_token` 后，可以将其缓存起来，再次获取时直接从缓存读取，减少 API 调用。整条流的配置如下，根据是否缓存了 `access-token` 将流执行分为两种情况，若已缓存，从缓存读取即可，否则，需要调用 API 获取 `access_token`：



## 组件说明

### 条件判断（组件3、4、7）

三个条件判断组件组合在一起，将流执行分为了两种情况，若符合组件4的执行条件，则流向组件5和6，其余情况流向组件8、9、10和11。其中，组件4的执行条件期望是一个条件表达式，因而使用 **表达式** 模式引用了“缓存-检查是否存在”组件2的输出，即是否缓存了 access-token。

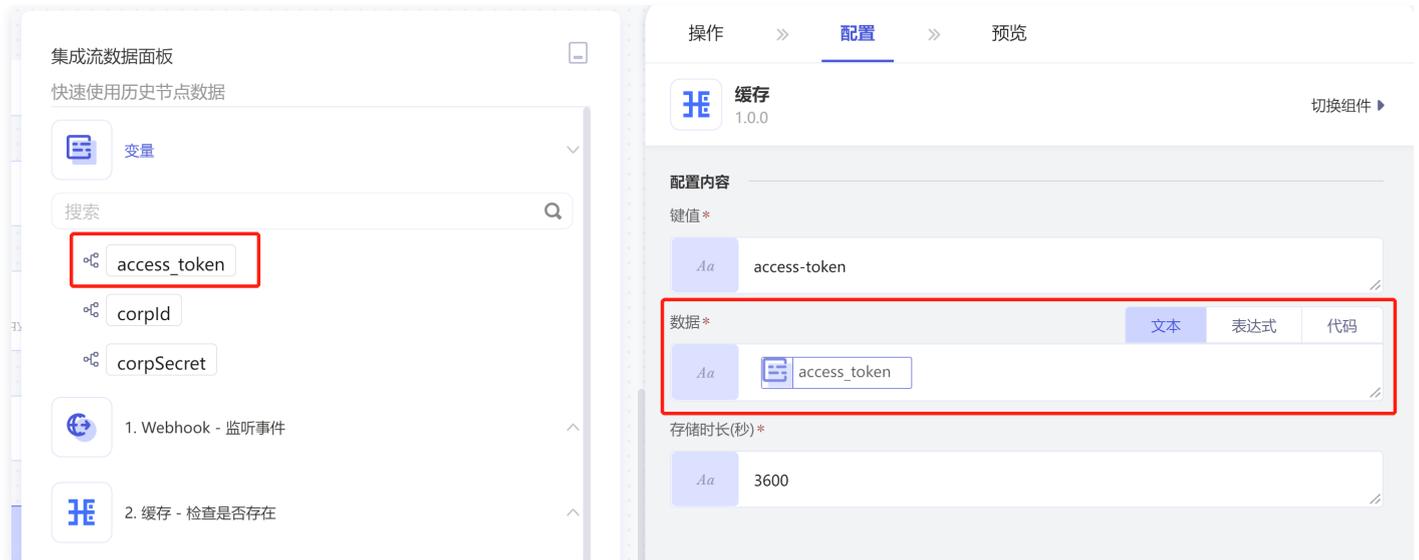


### 缓存（组件2、5、11）

- 在缓存中，以键值"access-token"绑定 access\_token 数据，由于键值是固定的，可以使用 文本 模式以字面量的方式输入"access-token"。

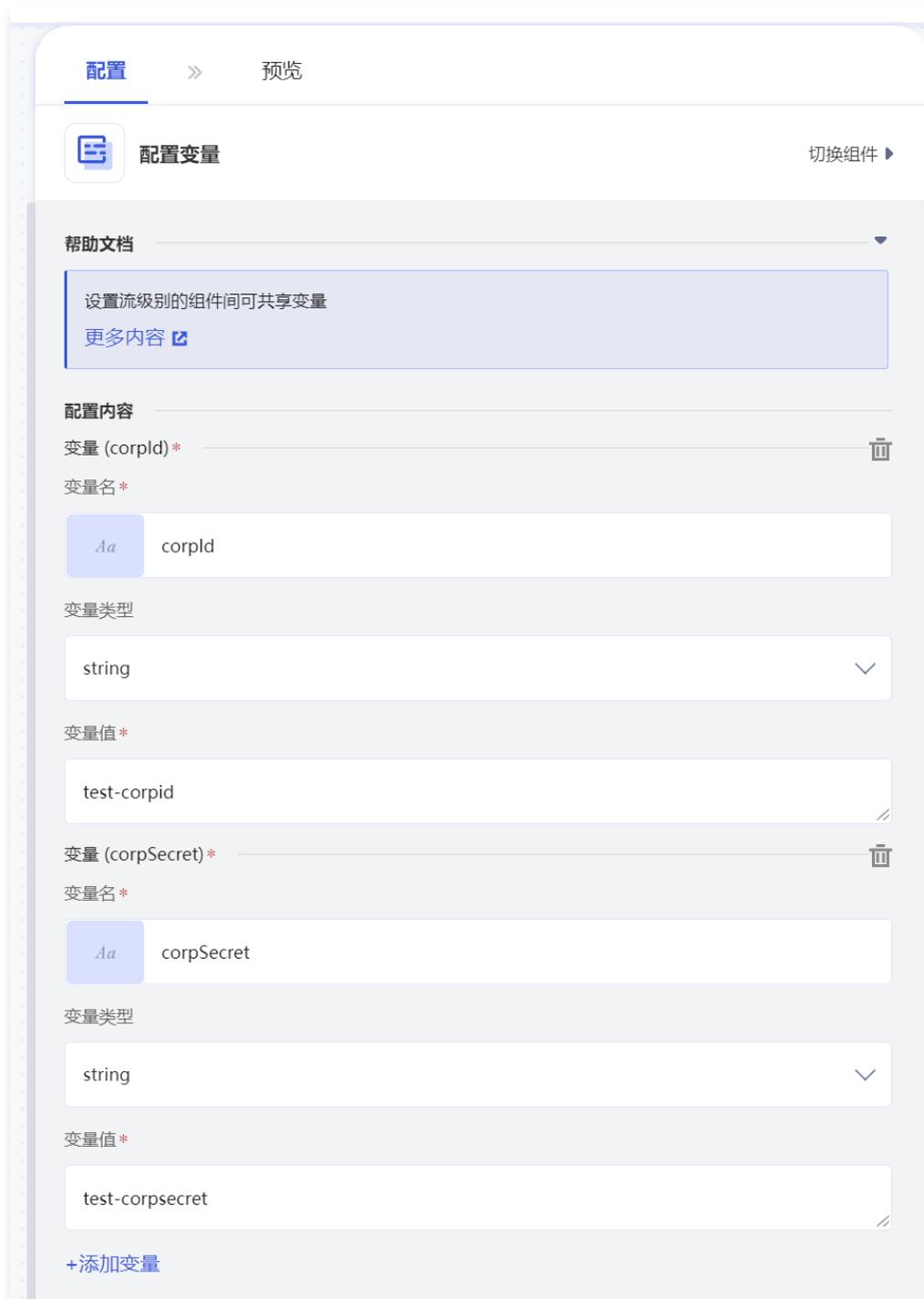


- 在缓存存储（组件11）中，还需要设置需要缓存的数据，即变量 access\_token，可以使用 文本 模式通过集成流数据面板引用变量 access\_token。



### 配置变量（组件6、8、11）

- 获取 `access_token` 的API需要 `corpid` 和 `corpsecret` 两个参数，在“企业微信”连接器中，这两个参数在连接器配置中直接输入即可。在示例中，通过“配置变量”组件8设置变量 `corpid` 和 `corpSecret` 来表示这两个参数的获取过程。`corpid` 和 `corpSecret` 作为固定值，可以使用 **文本** 模式以字面量的方式输入。示例图片中以 `test-corpid` 和 `test-corpsecret` 代替真实 `corpid` 和 `secret`，用户可跟随 [企业微信 API](#) 文档的引导，查询自己真实的 `corpid` 和 `secret`。



- "配置变量"组件6和11将获取的 `access_token` 设置为变量，方便调用业务API等后续过程使用。组件11需要从 API 调用的响应数据中找到对应的 `access-token`，这是一个数据处理过程，使用 [表达式](#) 模式对引用的API调用数据进行解析。
  - 组件6从缓存中取出 `access_token`，直接使用 [文本](#) 模式引用"缓存-读取"组件的输出即可。



- 组件11需要从 API 调用的响应数据中找到 access-token，这是一个数据处理过程，使用 **表达式** 模式对引用的 API 调用数据进行解析。



## HTTP 请求 (组件9)

"HTTP 请求"组件向企业微信 API 地址发起 get-token 请求调用，请求参数为变量中的 corpId 和 corpSecret，使用代码模式动态设置配置中的"URL 参数"。无论是 Python 异或 Java 脚本，均可使用"集成流数据面板"引用变量 corpId 和 corpSecret。

- [代码模式 Python](#)

操作 >> **配置** >> 预览

 HTTP请求 切换组件 ▾

**帮助文档**

通过HTTP请求向其它服务发起HTTP请求  
[更多内容](#)

**配置内容**

请求方法\*

POST ▾

请求地址\*

https://qyapi.weixin.qq.com/cgi-bin/gettoken

URL参数 ⓘ

文本 表达式 **代码**

```

1 def dw_process(msg):
2     return {
3         'corpid': corpid,
4         'corpsecret': corpSecret,
5     }
    
```

- 代码模式 Java

操作 >> **配置** >> 预览

HTTP请求 切换组件 ▾

配置内容

请求方法\*

POST ▾

请求地址\*

https://qyapi.weixin.qq.com/cgi-bin/gettoken

URL参数 ⓘ

文本 表达式 **代码**

```
1 import com.tencent.ipaas.dataway.common.message.Message;
2 import com.tencent.ipaas.dataway.common.message.DataRef;
3
4 import java.util.Map;
5 import java.util.HashMap;
6
7 public class Handler {
8     public Object eval(Message msg) {
9         Map<String, Object> map = new HashMap<>();
10        map.put("corpId", corpId);
11        map.put("corpsecret", corpSecret);
12        return map;
13    }
14 }
```

下一步

**说明:**

实际使用时，企业微信连接器使用 XML 语言完成 get-token，本示例为了展示 Dataway 使用方式在流中复现了这一过程。Dataway 在其中主要的作用是流转和预处理消息 Message，实现动态赋值，并将数据交给上层调用组件使用。

## 将 JSON 类型输入转换成 XML 类型输入

使用 Dataway 可以完成数据类型的转换，例如 JSON 转换成 XML、XML 转换成 CSV 等，在此以 JSON 类型的数据转换成 XML 为例。

### 输入数据

我们在流中预先使用"配置 Payload"组件将当前消息载荷 (msg.payload) 设置为 JSON 格式的 Entity 数据。

```
def dw_process(msg):
    return Entity.from_value({
```

```
        'name': 'zhangsan',
        'age': 12,
        'male': True,
        'brothers': ['lisi', 'wangwu'],
    }, mime_type="application/json")
```

## 代码模式 Python 脚本

新增"配置 Payload"组件，模拟需要输入 XML 数据的组件，此时，可以使用 [代码模式 Python](#) 脚本将 JSON 数据转化为 XML 数据。

```
def dw_process(msg):
    input_param = msg.payload['^value']
    return Entity.from_value({
        'root': {
            'name': input_param['name'],
            'age': input_param['age'],
            'male': input_param['male'],
            'brothers': input_param['brothers'],
            "@id": "haha"
        }
    }, mime_type="application/xml")
```

## 输出数据

使用 Entity['^blob'] 获取二进制数据并解码，可以查看输出的 XML 结构字符串，如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<root id="haha">
  <name>zhangsan</name>
  <age>12</age>
  <male>>true</male>
  <brothers>lisi</brothers>
  <brothers>wangwu</brothers>
</root>
```

### ❗ 说明：

Dataway 表达式通过 Entity 类型提供强大的数据转换功能，通过设置不同的 MIME 类型，可以实现不同类型之间的数据转换。详情请参见 [基本概念-MIME 类型支持](#)。

## 常见问题

最近更新时间：2024-08-01 15:33:02

### DataWay Python 表达式与 Python 脚本是什么关系？

Dataway Python 表达式基于 Python3 语言进行封装，在 Python 的基础上进行了功能的裁剪，并要求必须定义合法的入口函数 `dw_process(msg)`。因此，所有的 DataWay 表达式正常执行结束之后都会有最终的返回值，但“Python脚本”却可以没有明确返回值。

### DataWay 表达式在什么场景下使用？

目前几乎所有的腾讯轻联核心组件都具有表达式求值的能力，Dataway 是腾讯轻联具有动态求值能力的关键所在。

### Dataway 脚本运行错误如何解决？

1. 排除 Dataway 脚本的语法错误，确保在 Dataway 编辑时通过语法检查。
2. 通过查看运行错误日志，定位 Dataway 脚本的错误原因及脚本错误行数，进行问题分析和定位。
3. 您也可以通过 [Dataway 脚本调试](#) 进行问题排查。

### 如何使用其他第三方模块？

Dataway 默认支持的内置模块仅包含 `time`、`json`、`math`、`base64`、`hmac`、`random`、`hashlib`、`Crypto`、`socket`、`struct`、`decimal`、`urllib`、`csv` 和 `datetime` 模块。

一般而言，这些第三方模块对 Dataway 脚本而言已足够。如果确实有需求，需新增第三方模块，请 [提交工单](#) 联系我们，经安全评估后，会为您开放该模块。