阿里云
DataWorks

使用指南

文档版本：20190818

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **警告:** 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 **说明:** 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
*## *	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

1 管理控制台

1.1 管理控制台概览

您可以通过管理控制台中的概览页面，查找最近使用的工作空间，进入相应工作空间的数据开发、数据集成、数据服务页面，或对其进行工作空间配置。您也可以在此页面创建工作空间和一键导入CDN。

以组织管理员（主账号）身份登录DataWorks控制台。

说明:

概览页面根据您的使用情况和创建时间更新显示内容，仅显示您最近使用或最近创建的三个工作空间。

- 如果子账号登录时，没有创建相应的工作空间，会提示您联系管理员，开通工作空间权限。如果需要给子账号授予创建工作空间的权限，需要首先#unique_5/unique_5_Connect_42_section_alf_oov_lne，然后#unique_6。
- 子账号最多显示两个工作空间，您可以进入工作空间列表页面查看所有的工作空间。
- 如果子账号权限为部署，则不能进入数据开发页面。
- DataWorks中的工作空间即MaxCompute中的项目，详情请参见项目。

概览页面说明如下：

- 工作空间

  显示您最近打开的三个工作空间，您可以单击对应工作空间后的工作空间配置或进入数据开发，对工作空间进行具体操作。您也可以进入工作空间列表页面进行相关操作，详情请参见工作空间列表。

- 常用功能

  - 您可以在此创建工作空间，详情请参见#unique_6。
  - 您也可以在此一键导入CDN。

1.2 工作空间列表

您可以通过工作空间列表页面，查看该账号下所有的工作空间，对工作空间进行配置、删除、激活和重试等操作，也可以在该页面创建工作空间和刷新列表。

操作步骤

1. 以组织管理员（主账号）身份登录DataWorks（数据工场，原大数据开发套件）产品详情页。
2. 单击管理控制台，进入概览页面。
3. 单击工作空间列表，进入工作空间列表页面，查看该账号下所有的工作空间。

- 状态：工作空间包括正常、初始化中、初始化失败、删除中和删除等5种状态。创建工作空间开始会进入初始化中，通常会显示初始化失败或正常2种结果。

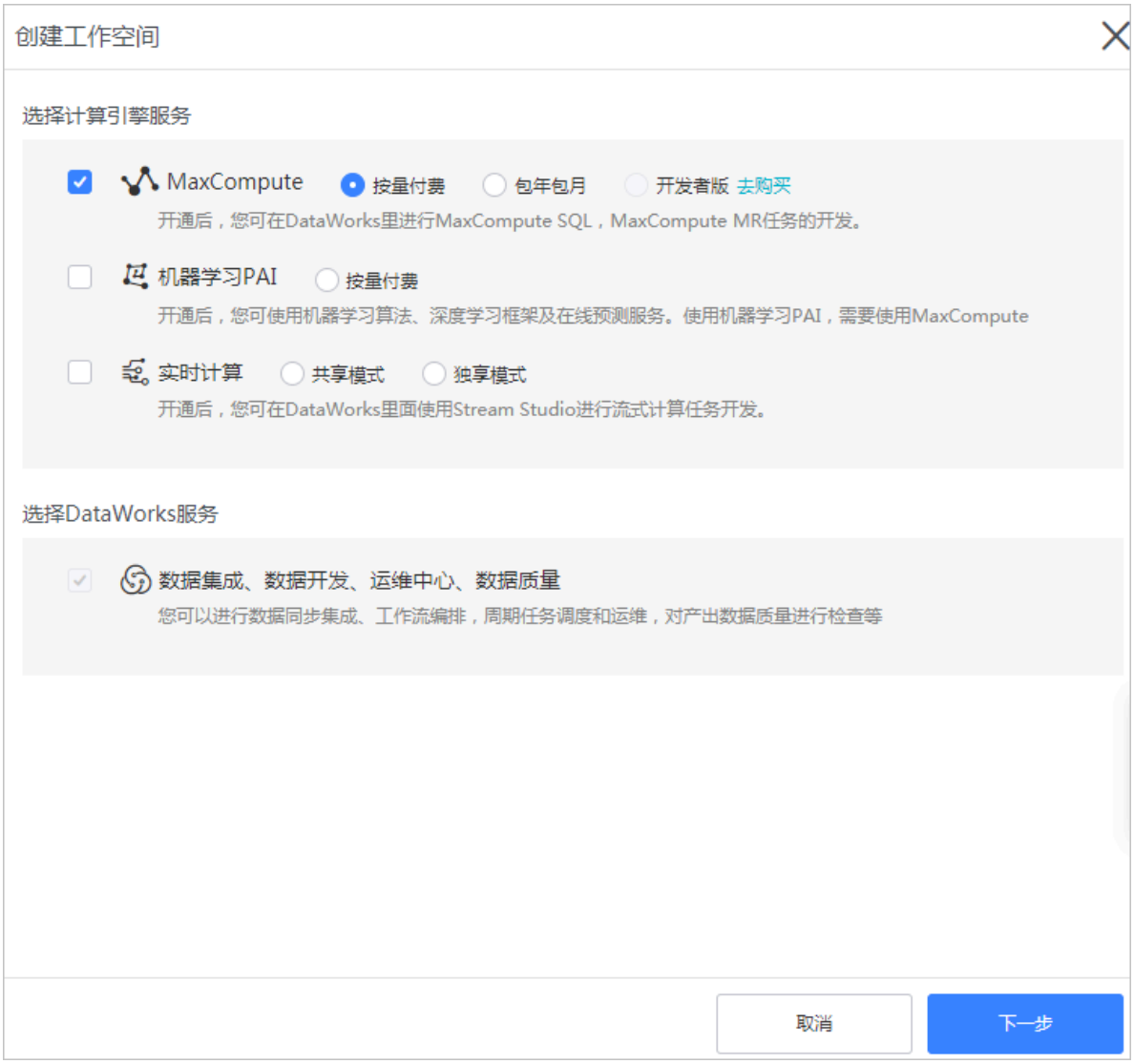
  禁用后，您也可以激活和删除工作空间，激活后工作空间正常。

- 开通服务：您的鼠标移至服务上，会展示您开通的所有服务。通常正常服务的图标为蓝色，欠费服务图标为红色并有相应的欠费标志，欠费已删除的服务的图标为灰色。通常服务欠费7天后仍未续费，会自动删除。

创建工作空间

1. 单击创建工作空间，选择计算引擎服务和DataWorks服务。

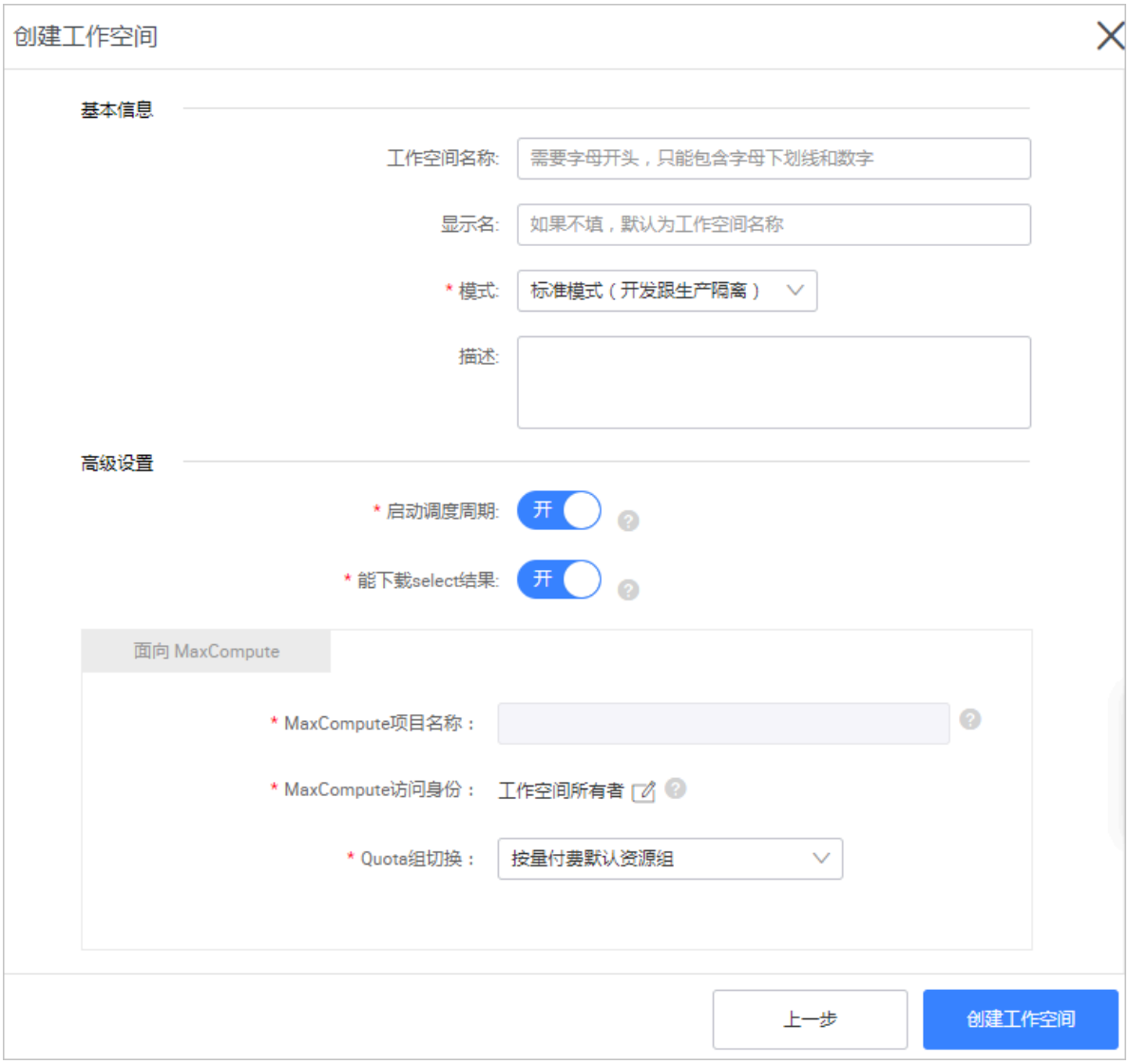
   DataWorks已启动商业化，如果该区域没有开通，需要首先开通商业化服务。默认选中数据集成、数据开发、运维中心和数据质量。



选项	配置	说明
选择计算引擎服务	MaxCompute	MaxCompute是一种快速、完全托管的TB/PB级数据仓库解决方案，能够更快速为您解决海量数据计算问题，有效降低企业成本，并保障数据安全。 说明: 完成创建Dataworks工作空间后，需要关联MaxCompute项目，否则现执行命令会报project not found的错误。

选项	配置	说明
	机器学习PAI	机器学习是指机器通过统计学算法，对大量的历史数据进行学习从而生成经验模型，利用经验模型指导业务。
	实时计算	开通后，您可以在DataWorks使用Stream Studio，进行流式计算任务开发。
选择DataWorks服务	数据集成	数据集成是稳定高效、弹性伸缩的数据同步平台。致力于提供复杂网络环境下、丰富的异构数据源之间数据高速稳定的数据移动及同步能力。详情请参见数据集成模块的文档。
	数据开发	该页面是您根据业务需求，设计数据计算流程，并实现为多个相互依赖的任务，供调度系统自动执行的主要操作页面。详情请参见数据开发模块的文档。
	运维中心	该页面可对任务和实例进行展示和操作，您可以在此查看所有任务的实例。详情请参见运维中心模块的文档。
	数据质量	DataWorks数据质量依托DataWorks平台，为您提供全链路的数据质量方案，包括数据探查、数据对比、数据质量监控、SQLScan和智能报警等功能。详情请参见数据质量模块的文档。

2. 单击下一步，配置新建工作空间的基本信息和高级设置。



分类	配置	说明
基本信息	工作空间名称	工作空间名称的长度需要在3到27个字符，以字母开头，且只能包含字母下划线和数字。
	显示名	显示名不能超过27个字符，只能字母、中文开头，仅包含中文、字母、下划线和数字。

分类	配置	说明
	模式	工作空间模式是DataWorks新版推出的新功能，分为简单模式和标准模式，双项目开发模式的区别请参见#unique_14。 · 简单模式：指一个Dataworks工作空间对应一个MaxCompute项目，无法设置开发和生产环境，只能进行简单的数据开发，无法对数据开发流程以及表权限进行强控制。 · 标准模式：指一个Dataworks工作空间对应两个MaxCompute项目，可以设置开发和生产双环境，提升代码开发规范，并能够对表权限进行严格控制，禁止随意操作生产环境的表，保证生产表的数据安全。
	描述	对创建的工作空间进行简单描述。
高级设置	启用调度周期	控制当前工作空间是否启用调度系统，如果关闭则无法周期性调度任务。
	能下载select结果	控制数据开发中查询的数据结果是否能够下载，如果关闭无法下载select的数据查询结果。
	MaxCompute项目名称	默认与DataWorks工作空间名称一致。
	MaxCompute访问身份	推荐使用工作空间所有者。
	Quota组切换	Quota用来实现计算资源和磁盘配额。

3. 配置完成后，单击创建工作空间。

工作空间创建成功后，即可在工作空间列表页面查看相应内容。

说明:

· 如果您成为工作空间所有者，代表该工作空间内的所有东西都属于您。在给别人赋权之前，任何人无权限访问您的空间。如果您使用的是子账号创建的工作空间，则该工作空间会同时属于这个子账号和对应的主账号。
· 子账号可以不用创建工作空间，只需被加入到某个工作空间，即可使用MaxCompute。

工作空间配置

您可以通过配置工作空间的操作，对当前工作空间的基本属性和高级属性进行设置，主要对空间、调度等进行管理和配置。

单击对应工作空间后的工作空间配置。

进入数据开发或数据集成

单击对应工作空间后的进入数据开发或进入数据集成，即可进入相应工作空间的数据开发或数据集成页面进行相关操作。

修改服务

通常修改服务是对计算引擎服务和DataWorks服务的操作，需要先购买才可以对相应的服务进行选择。根据您购买的服务自动选择付费形式。您可以对MaxCompute进行充值、升级、降配和续费等操作。

- **充值：当您的MaxCompute服务出现欠费预警，可以对您的服务进行充值操作。**
- **升级、降配：如果您购买的MaxCompute预付费资源无法满足您业务计算量需求，需要购买更多资源，可以进行资源升级操作。详情请参见#unique_15。**
- **续费：如果包年包月的实例过期，将会冻结使用对应实例的预付费项目，您可以对包年包月实例进行续费操作。详情请参见续费管理。**

说明：

- **后付费：仅显示充值按钮。**
- **预付费：显示全部按钮。**

删除工作空间和禁用工作空间

单击对应工作空间后的更多，即可对工作空间进行删除和禁用操作。

- 删除工作空间

  单击删除工作空间，填写删除工作空间对话框中的验证码，单击确定。

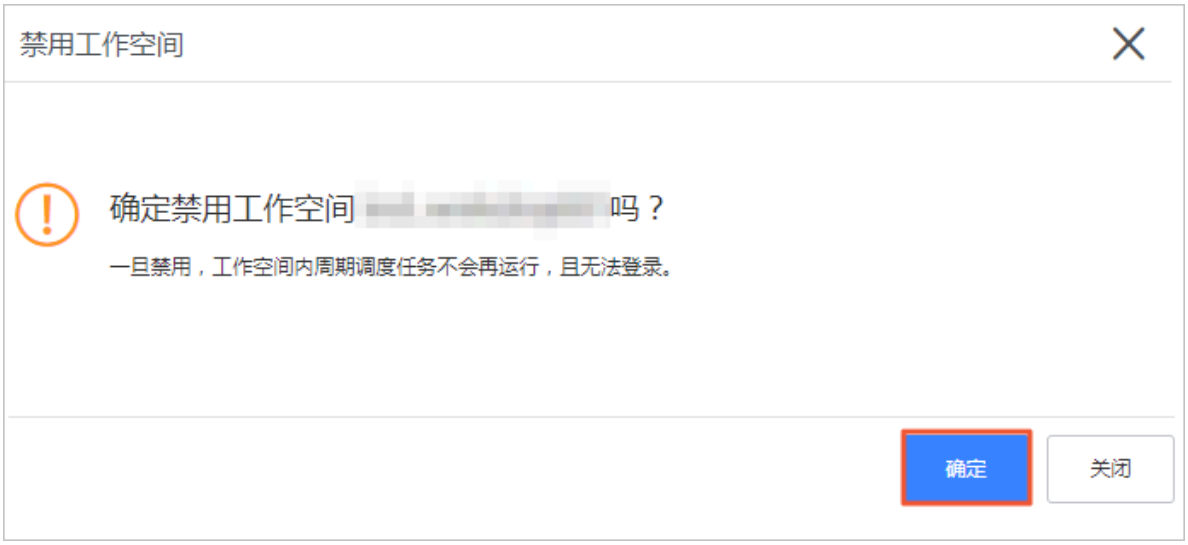
说明:

- 删除工作空间对话框中的验证码YES是固定的。
- 删除工作空间的操作为不可逆操作，请慎重使用。

- 禁用工作空间

  单击禁用工作空间，弹出禁用工作空间对话框，单击确定。

  一旦禁用工作空间，工作空间内周期调度任务不会再生成实例。禁用前生成的实例到运行时间会自动运行，只是无法登录工作空间查看相应的情况。



1.3 资源列表

您可以通过资源列表页面，查看该账号下所有的独享资源和公共资源，并对其进行管理。

操作步骤

1. 以组织管理员（主账号）身份登录DataWorks（数据工场，原大数据开发套件）产品详情页。
2. 单击管理控制台，进入概览页面。
3. 单击上方的资源列表，进入资源列表页面。您可以查看该账号下所有的资源，并对其进行操作。

资源列表页面包括独享资源和公共资源。

独享资源

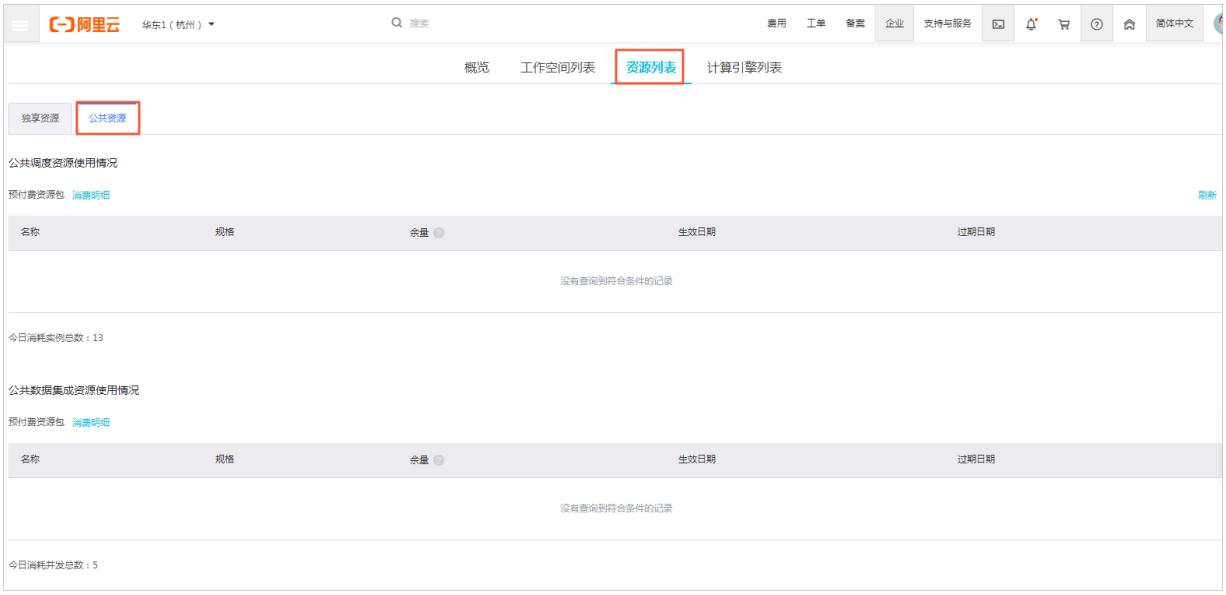
单击资源列表页面下的独享资源，即可查看该账号下的独享资源，并对其进行操作。

列表项	说明
新增独享资源	DataWorks为您提供独享资源模式，支持购买独享的机器资源，来分配给工作空间运行任务。您可以单击右上角的新增独享资源进行添加，详情请参见#unique_17/unique_17_Connect_42_section_5dk_xbo_yo2。
刷新	单击右上角的刷新，即可同步最新的独享资源列表。
资源名称	独享资源名称，由英文字母、下划线、数字组成，不超过60个字符，创建后不支持修改。
备注	创建资源时，进行的简单描述。
类型	资源的使用类型。独享资源包括独享调度资源和独享数据集成资源两种类型，分别适用于通用任务调度和数据同步任务专用。
状态	资源的状态，包括正常、冻结、删除、创建中、创建失败、更新中、更新失败、删除中和删除失败等状态。
到期时间	到期时间与独享资源购买订单中选定的时间相关联，即独享资源的有效期。详情请参见#unique_18。
资源数	表示购买的资源个数。
资源使用率	资源的使用情况（负载），用百分比表示。

列表项	说明
操作	您可以对响应的相应的独享资源进行以下操作： · 查看信息：查看独享资源的资源类型、安全组、VPC、VSwitch、EIP地址和网段等信息。 · 扩容：如果独享资源使用率过高，不能满足实际需求，可以单击扩容进行资源变更配置，调大资源数量来进行扩容。 · 缩容：当资源出现闲置不使用的情况，可以单击缩容进行资源变更配置，调小资源数量来进行缩容。 · 续费：单击相应资源后的续费，可以延长该独享资源的到期时间。 · 专有网络绑定：独享资源部署在DataWorks托管的专有网络（VPC）中，如果需要与您自己的专有网络连通，需要进行专有网络绑定操作。 · 修改归属工作空间：独享资源需要绑定归属的工作空间，方可被任务真正使用。一个独享资源可分配给多个工作空间使用。 具体操作请参见#unique_17/unique_17_Connect_42_section_maj_9t9_pfl。

公共资源

单击资源列表页面下的公共资源，即可查看该账号下公共资源的使用情况。更多详情请参见#unique_19。



列表项	说明
购买资源包	单击后即可跳转至购买页面。DataWorks为您提供公共调度资源组、公共数据集成资源组和AppStudio开发环境运行空间3种资源包，您可以根据自身需求进行购买。

列表项	说明
消费明细	单击后即可跳转至用户中心页面，查看当前账户的详细消费信息。
名称资源名称	公共资源名称，由英文字母、下划线、数字组成，不超过60个字符，创建后不支持修改。
规格	购买时选择的资源规格。 · 公共调度资源组-资源包的规格为1,500,000个实例/月。 · 公共数据集成资源组-资源包的规格为1,500,000个并发/月。 · AppStudio开发环境运行空间-资源包的额度为100计算时。
余量	自购买日起，所有运行成功的实例将会优先消耗该资源包内的额度，余量为剩余的额度。
生效日期	购买成功后即时生效。
过期日期	过期日期与公共资源购买订单中选定的时间相关联，即公共资源的有效期。

1.4 独享资源模式

DataWorks为您提供独享资源模式，支持购买独享的机器资源，来分配给工作空间运行任务。

独享资源模式下，机器的物理资源（网络/磁盘/CPU/内存等）完全独享。不仅可以隔离用户间的资源使用，也可以隔离不同工作空间任务的资源使用。此外，独享资源也支持灵活的扩容、缩容功能，可以满足资源独享、灵活配置等需求。

独享资源组只能访问同一地域的VPC数据源，也可以访问跨地域的公网RDS地址，但速度较慢。

购买独享资源

DataWorks独享资源采用预付费包年包月的方式购买，您可以通过产品详情页或新增独享资源两个入口进行购买。

- 产品详情页入口

  进入DataWorks产品页面，单击独享资源组，即可跳转至购买页面。

- 新增独享资源入口

  1. 登录DataWorks控制台，进入资源列表 > 独享资源页面。
  2. 如果您在该Region未购买过独享资源，单击右上角的新增独享资源。

  3. 单击新增独享资源对话框中订单号后的购买，即可跳转至购买页面。

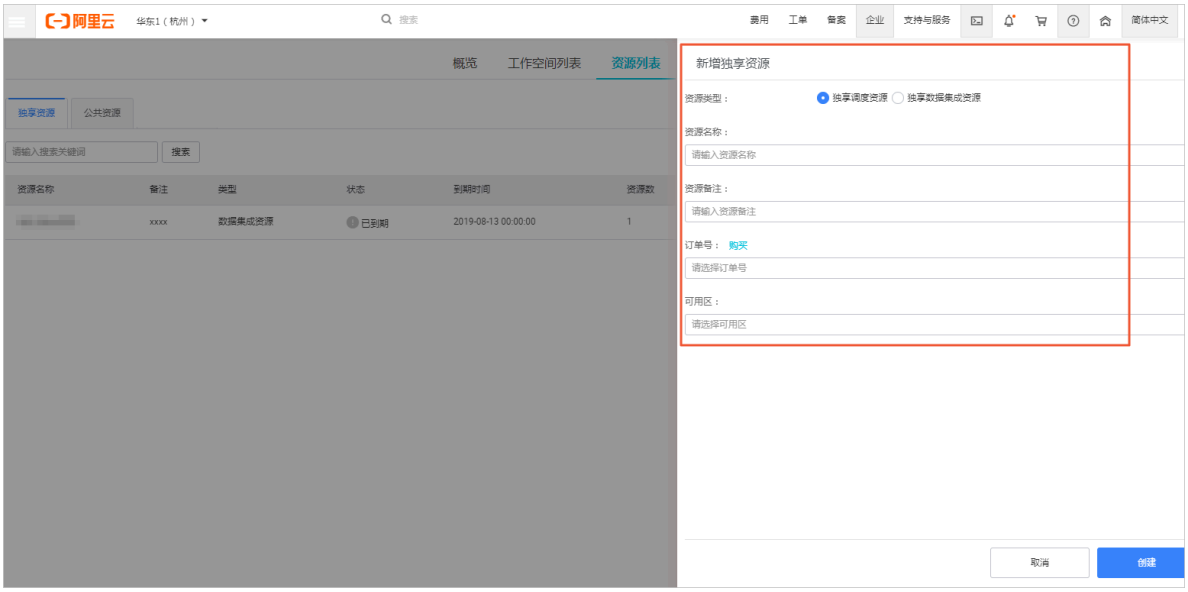
进入购买页面后，请根据实际需要，选择相应的地域、独享资源类型、独享调度资源、资源数量和计费周期，单击立即购买。

说明：

独享资源不支持跨地域使用，即华东2（上海）地域的独享资源，只能给华东2（上海）地域的工作空间使用。

新增独享资源

1. **进入资源列表 > 独享资源页面，单击右上角的新增独享资源。**
2. **填写新增独享资源对话框中的配置。**



配置	说明
资源类型	资源的使用类型。独享资源包括独享调度资源和独享数据集成资源两种类型，分别适用于通用任务调度和数据同步任务专用。

配置	说明
资源名称	资源的名称，租户内唯一，请避免重复。 说明: 租户即主账号，一个租户（主账号）下可以有多个用户（子账号）。
资源备注	对资源进行简单描述。
订单号	此处选择购买的独享资源订单。如果没有购买，可以单击购买，跳转至售卖页进行购买。
可用区	单个地域提供了不同机器的可用区，请根据自身情况进行选择。

3. 配置完成后，单击创建，即可新增独享资源。

说明:
独享资源在20分钟内完成环境初始化，请耐心等待其状态更新为运行中。

查看独享资源

完成独享资源的创建后，您可以在独享资源列表中查看到期时间、资源数和资源使用率等基本信息。

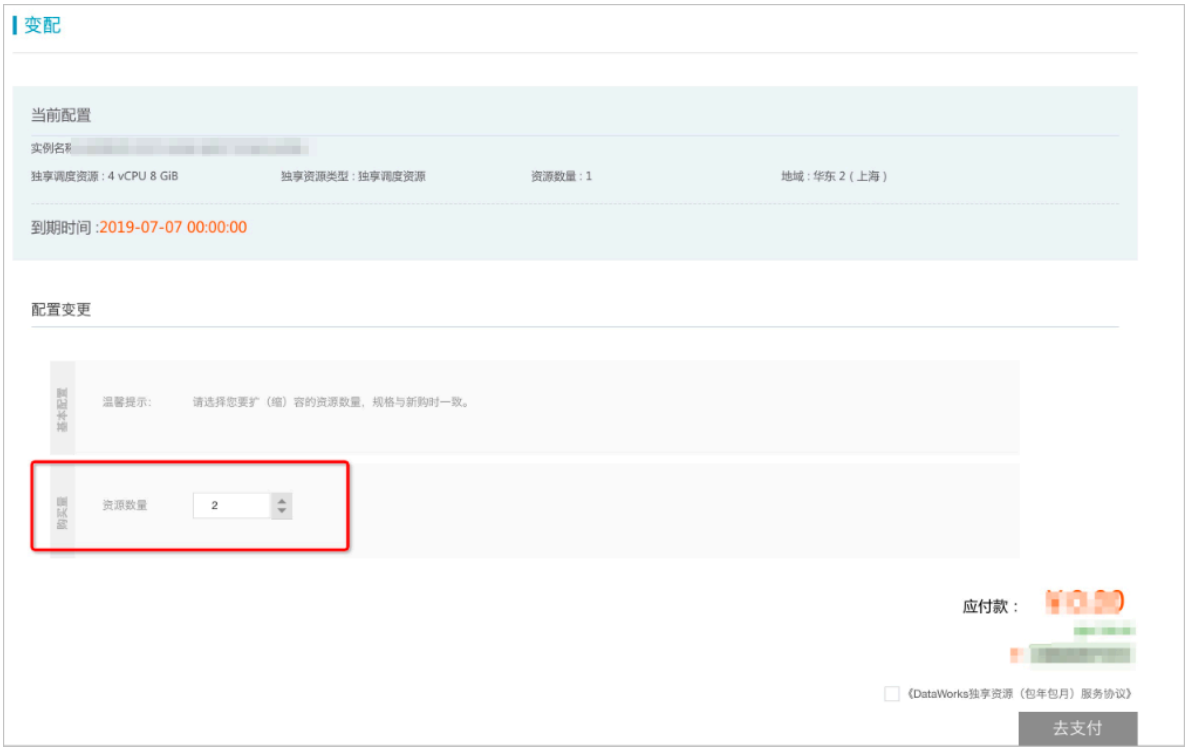
- 到期时间：到期时间与独享资源购买订单中选定的时间相关联，即独享资源的有效期。
  - 您可以在到期前进行续费。如果到期后未续费，状态会变为过期，独享资源不能被新任务使用。
  - 您也可以在七天内对过期后的独享资源进行激活。如果超过7天仍未激活，独享资源会被释放。
- 资源数：表示购买的资源个数。
- 资源使用率：资源的使用情况（负载），用百分比表示。

操作独享资源

完成独享资源的创建后，您可以在独享资源列表中进行扩容、缩容、续费、专有网络绑定和修改归属工作空间等操作。

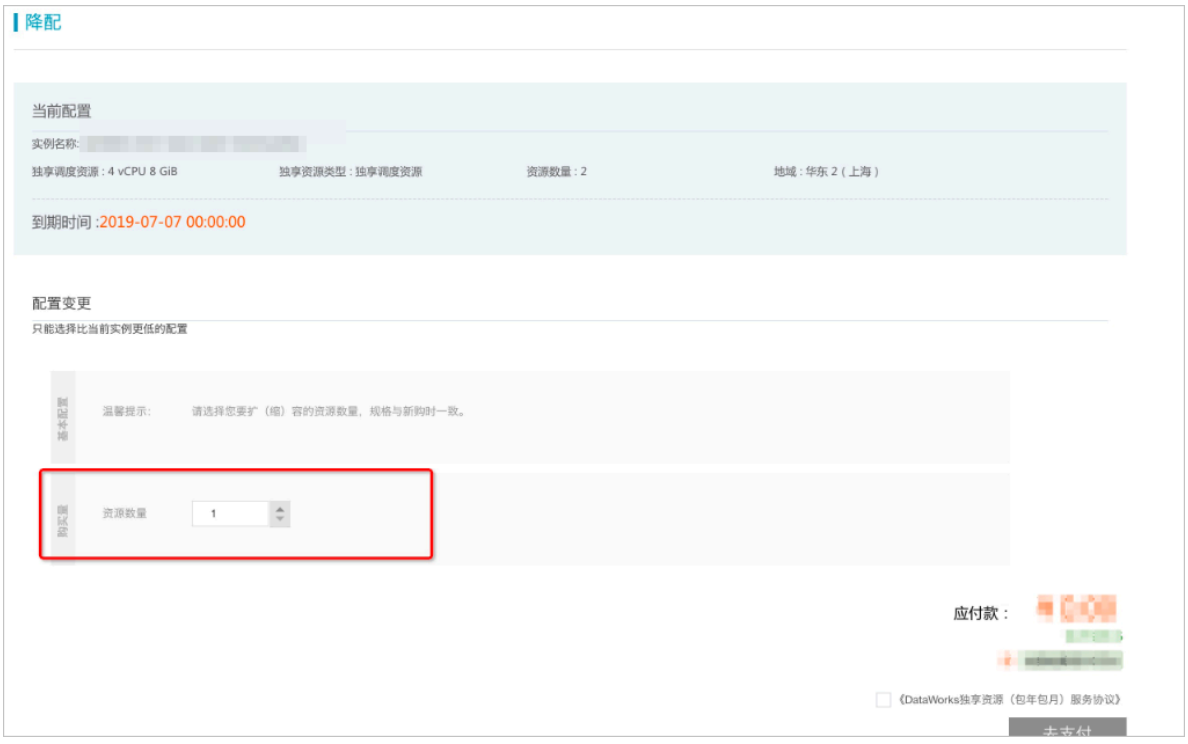
- 扩容

  如果独享资源使用率过高，不能满足实际需求，可以单击扩容进行资源变更配置，调大资源数量来进行扩容。



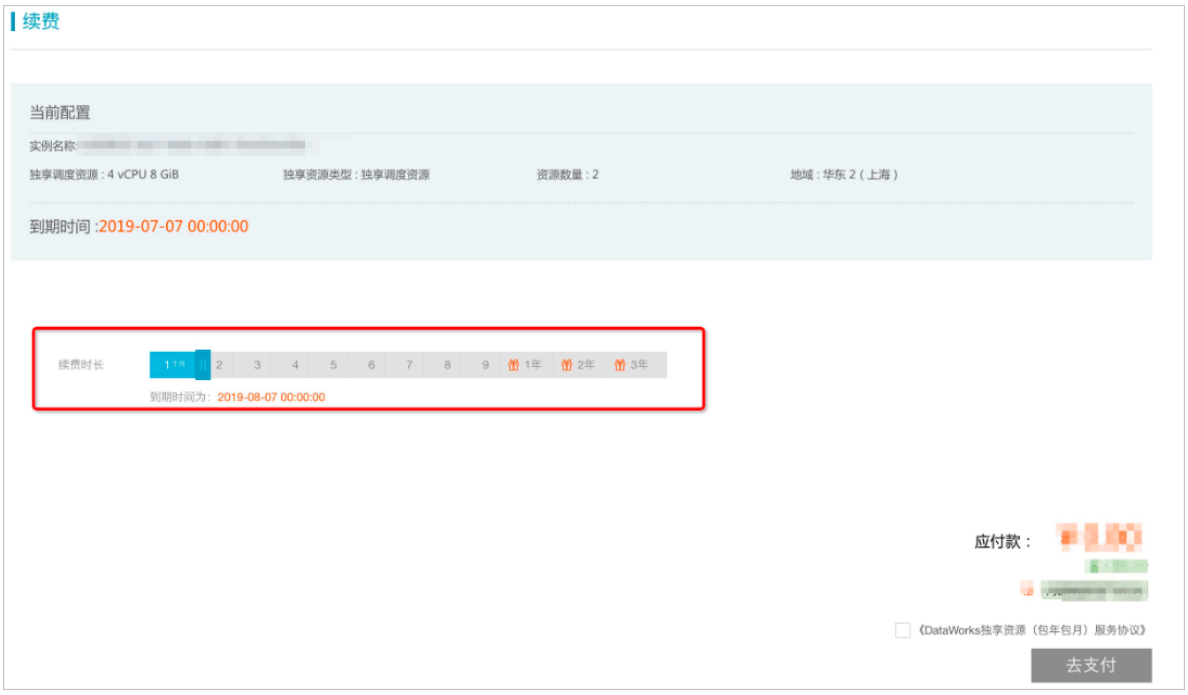
- 缩容

  当资源出现闲置不使用的情况，可以单击缩容进行资源变更配置，调小资源数量来进行缩容。



- 续费

  单击相应资源后的续费，可以延长该独享资源的到期时间。



- 专有网络绑定

  独享资源部署在DataWorks托管的专有网络（VPC）中，如果需要与您自己的专有网络连通，需要进行专有网络绑定操作。

  单击相应资源后的专有网络绑定，即可进入绑定页面。

  说明:

  绑定前，需要进行RAM授权，让DataWorks拥有访问您的云资源的权限。

授权完成后，单击新增绑定。填写对话框中的配置，即可添加一个新的专有网络绑定。

说明：

独享资源组的可用区必须选择要访问数据源的可用区，绑定专有网络时，选择访问数据源所绑定的交换机。

绑定成功后，如果信息发生变化，可以重新进行绑定，目前暂不支持解绑。

- 修改归属工作空间

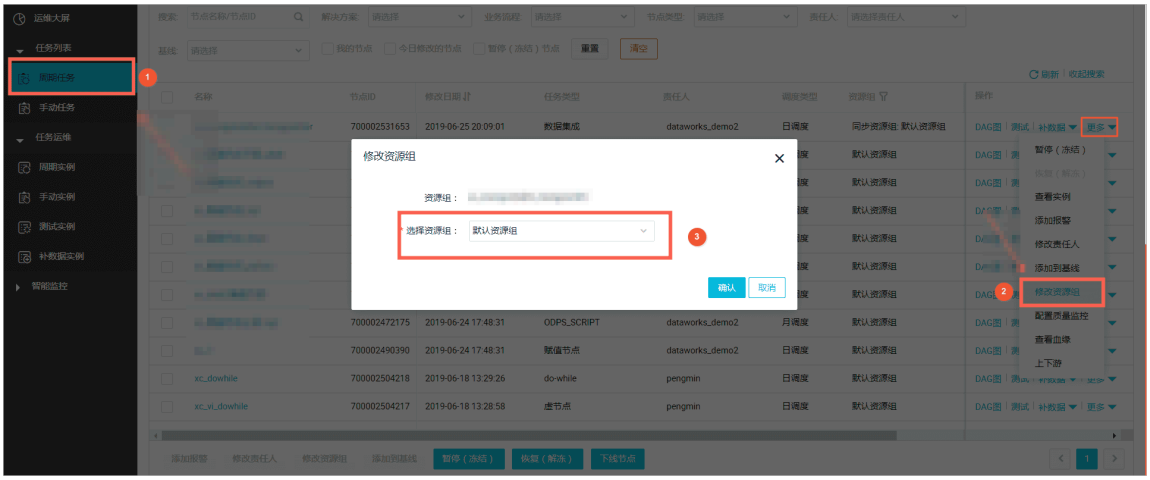
  独享资源需绑定归属的工作空间，方可被任务真正使用。一个独享资源可分配给多个工作空间使用。

使用独享资源

独享资源绑定到工作空间后，可以在工作空间中，将任务分配到特定的独享资源上使用。

- 独享调度资源

  1. 单击左上角的图标，选择全部产品 > 运维中心（工作流），进入运维中心页面。
  2. 单击左侧导航栏中的周期任务，选择相应节点后的更多 > 修改资源组。



  3. 在修改资源组对话框中，选择相应的独享调度资源，单击确认。

- 独享数据集成资源

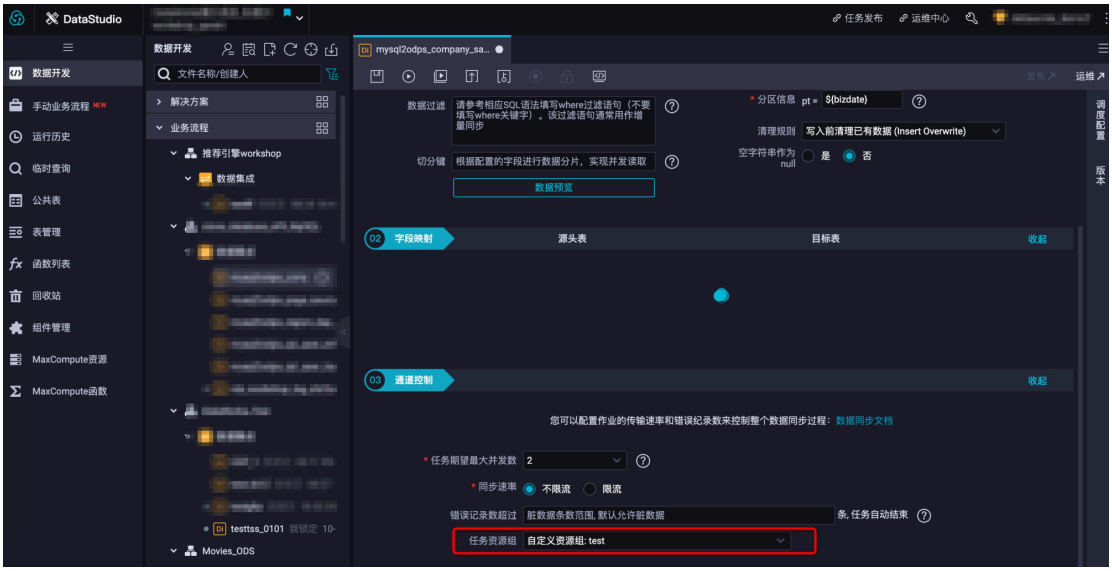
  独享数据集成资源创建成功后，需要在配置数据集成任务时修改任务运行资源。

  说明：

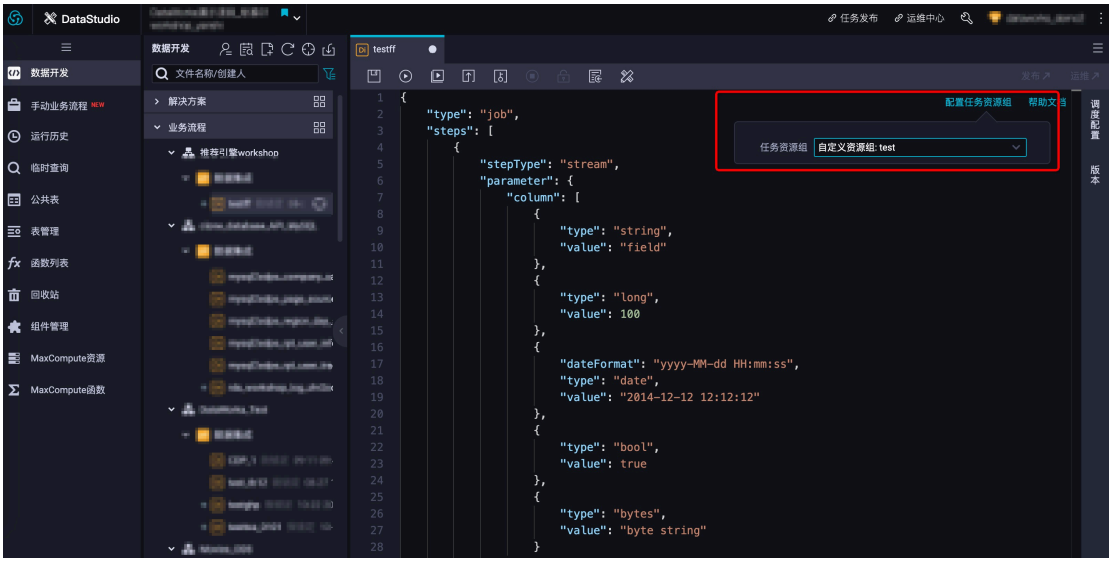
  - 独享数据集成资源组不支持本地机房的数据库，请使用自定义资源组。

- 独享数据集成资源组不支持跨区域的VPC数据库同步。

1. 单击相应资源后的修改归属工作空间，绑定独享数据集成资源归属的工作空间。
2. 配置数据集成任务时，将默认资源组配置为需要的独享数据集成资源。
   - 向导模式开发时，在通道控制 > 任务资源组下拉框中，选择相应的独享数据集成资源。

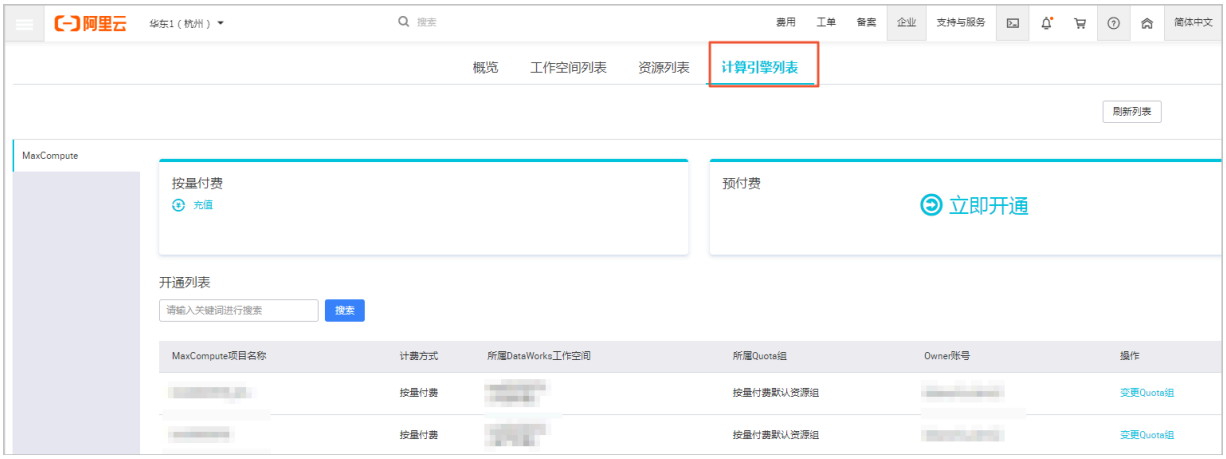


   - 脚本模式开发时，单击右上角的配置任务资源组，在任务资源组下拉框中，选择相应的独享数据集成资源。



1.5 计算引擎列表

您可通过管理控制台中的计算引擎列表页面，查看MaxCompute项目空间的付费方式和开通列表。



- MaxCompute目前支持按量付费和预付费两种付费方式，已开通的付费方式下会显示续费管理，未开通的付费方式下则会显示立即开通。
- 开通列表：您可以根据您的项目空间名称进行搜索，项目空间列表会显示项目空间的基本信息。

  您可以变更Quota组，但只有预付费的项目空间单击变更Quota组后，会跳转至CU管家界面，如果您没有开通预付费，则会提示账号没有购买预付费资源。

2 数据集成

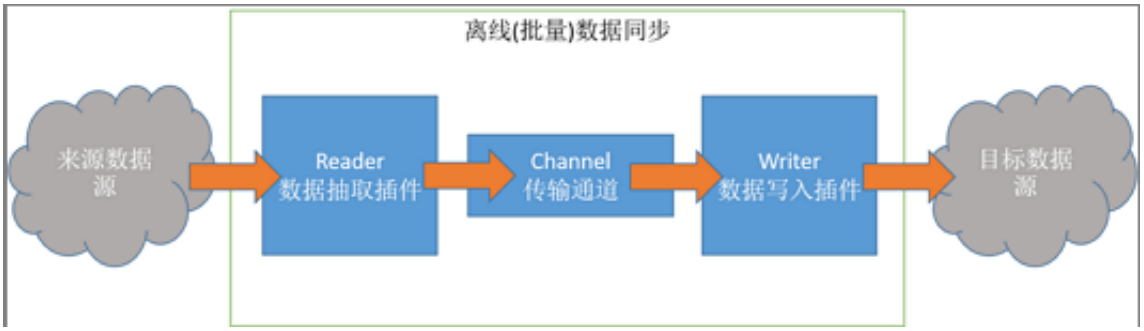
2.1 数据集成简介

2.1.1 数据集成概述

数据集成是阿里集团对外提供的稳定高效、弹性伸缩的数据同步平台。致力于提供复杂网络环境下、丰富的异构数据源之间数据高速稳定的数据移动及同步能力。

离线（批量）数据同步简介

离线（批量）的数据通道主要通过定义数据来源和去向的数据源和数据集，提供一套抽象化的数据抽取插件（称之为Reader）、数据写入插件（称之为Writer），并基于此框架设计一套简化版的中间数据传输格式，从而达到任意结构化、半结构化数据源之间数据传输的目的。



支持的数据源类型

数据集成提供丰富的数据源支持，如下所示。

- 文本存储（FTP/SFTP/OSS/多媒体文件等）。
- 数据库（RDS/DRDS/MySQL/PostgreSQL等）。
- NoSQL（Memcache/Redis/MongoDB/HBase等）。
- 大数据（MaxCompute/AnalyticDB/HDFS等）。
- MPP数据库（HybridDB for MySQL等）。

更多详情请参见#unique_25。

说明:
由于每个数据源的配置信息差距较大，需要根据使用情况详细查询参数配置信息。所以在数据源配置、作业配置页面提供了详细描述，请您根据自身情况进行查询使用。

同步开发说明

同步开发提供向导模式和脚本模式两种开发模式。

- 向导模式：提供向导式的开发引导，通过可视化的填写和下一步的引导，帮助快速完成数据同步任务的配置工作。向导模式的学习成本低，但无法享受到一些高级功能。
- 脚本模式：您可以通过直接编写数据同步的JSON脚本来完成数据同步开发，适合高级用户，学习成本较高。脚本模式可以提供更丰富灵活的能力，做精细化的配置管理。

说明：

- 向导模式生成的代码可以转换为脚本模式，此转换为单向操作，转换完成后无法恢复到向导模式，因为脚本模式能力是向导模式的超集。
- 代码编写前需要完成数据源的配置和目标表的创建。

网络类型说明

网络类型分为经典网络、专有网络（VPC）和本地IDC网络（规划中）。

- 经典网络：统一部署在阿里云的公共基础网络内，网络的规划和管理由阿里云负责，更适合对网络易用性要求比较高的客户。
- 专有网络：基于阿里云构建出一个隔离的网络环境。您可以完全掌控自己的虚拟网络，包括选择自有的IP地址范围，划分网段，以及配置路由表和网关。
- 本地IDC网络：您自身构建机房的网络环境，与阿里云网络隔离。

经典网络和专有网络相关问题请参见经典网络和VPC常见问题。

补充说明：

- 网络连接可以支持公网连接，网络类型选择经典网络即可。需要注意公网带宽的速度和相关网络费用消耗。无特殊情况不建议使用。
- 规划中的网络连接，进行数据同步，可以使用本地新增运行资源+脚本模式的方案进行数据同步传输。您也可以使用Shell+DataX方案。
- 专有网络VPC是构建一个隔离的网络环境，可以自定义IP地址范围、网段、网关等。随着专有网络安全性提高，专有网络运用越来越广，所以数据集成提供了RDS-MySQL、RDS-SQL Server、RDS-PostgreSQL，在专有网络下不需要购买一台和VPC同网络的ECS，系统通过反向代理会自动检测从而网络能够互通。对于阿里云其他的数据库PPAS、OceanBase、Redis、MongoDB、Memcache、Tabl eStore、HBase等，后续也会提供支持。所以非RDS的数据源在专有网络下配置数据集成的同步任务需要购买同网络的ECS，这样可以通过ECS连通网络。

约束与限制

- 支持且仅支持结构化（例如RDS、DRDS等）、半结构化、无结构化（OSS、TXT等，要求具体同步数据必须抽象为结构化数据）的数据同步。也就是说，数据集成支持传输能够抽象为逻辑二维表的数据同步，其他完全非结构化数据，例如OSS中存放的一段MP3，数据集成暂不支持将其同步到MaxCompute，这个功能会在后期实现。
- 支持单个和部分跨Region地域内数据存储相互同步、交换的数据同步需求。

  部分地域通过经典网络是可以传输的，但不能保证。如果必须使用且测试经典网络不通，可以考虑使用公网方式连接。
- 仅完成数据同步（传输），本身不提供数据流的消费方式。

参考文档

- 数据同步任务配置的详细介绍请参见创建数据同步任务。
- 如果处理像OSS等非结构化数据的详细介绍请参见MaxCompute访问OSS数据。

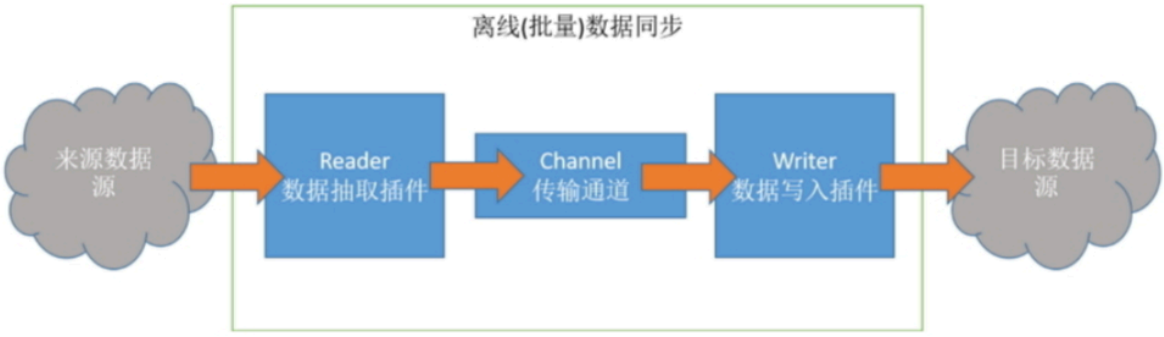
2.1.2 创建数据集成任务

本文为您介绍创建数据集成任务的流程和操作步骤。

数据集成是阿里巴巴集团对外提供的可跨异构数据存储系统、可靠、安全、低成本、可弹性扩展的数据同步平台，为数据源提供不同网络环境下的全量/增量数据进出通道。

Reader插件通过远程连接数据库，并执行相应的SQL语句，将数据从数据库中Select出来，从底层实现了从数据库读取数据。

Writer插件通过远程连接数据库，并执行相应的SQL语句，将数据写入数据库，从底层实现了向数据库写入数据。



数据集成任务准备工作

创建阿里云账号

1. 开通阿里云主账号，并创建账号的访问秘钥，即AccessKeys。

2. 开通MaxCompute，自动产生一个默认的MaxCompute数据源，并使用主账号登录DataWorks。
3. 创建工作空间。您可在工作空间中协作完成工作流，共同维护数据和任务等，因此使用DataWorks前需要先创建工作空间。

> 说明:
> 如果您想通过子账号创建数据集成任务，可以赋予其相应的权限。详情请参见准备RAM子账号。

创建源端和目标端数据库和表

1. 您可以使用建表语句或直接通过客户端建表，不同的数据源库创建数据库和表，请参见相应数据库的官方文档进行创建。
2. 给相关数据库和表赋予读写的权限。

> 说明:
> 一般至少需要赋予Reader端读的权限，赋予Writer端增、删、改的权限，建议提前赋予数据库中的表足够的权限。

数据集成任务操作步骤

创建数据源

1. 从数据库获取相关的数据源信息。
2. 在界面配置相关的数据源。

> 说明:
> - 界面配置数据源只支持一部分，如果在界面找不到相关的配置数据源界面可以直接脚本模式配置，将相关的数据源信息填写在JSON脚本中。
> - 支持数据源的情况，请参见支持的数据源类型。
> - 如何配置数据源和注意细节请参见数据源配置。

创建自定义资源组（可选）

1. 创建资源组。
2. 添加服务器。
3. 安装Agent。
4. 检查连通性。

> 说明:

- 网络环境不通或者DataWorks提供的资源满足不了您任务运行条件的情况下，你可以选择添加自定义资源组。
- 建议无论是专有网络还是经典网络都选择专有网络的添加形式。
- 配置自定义资源组的方式，请参见新增调度资源。
- 最佳实践：
  - （仅一端不通）数据源网络不通的情况下的数据同步
  - （两端都不通）数据源网络不通的情况下的数据同步

配置数据集成任务

1. 配置同步任务的读取端，每个Reader插件的配置细节请参见配置Reader插件。
2. 配置同步任务的写入端，每个Writer插件的配置细节请参见配置Writer插件。
3. 配置同步任务读写端的映射关系。
4. 配置同通道控制，您可以在这个步骤切换相关的资源组。

说明：

- 配置任务有向导模式 、脚本模式两种模式 。
- 配置任务时，您可以对您的任务进行速度调优，详情请参见优化配置。
- 向导模式可以转换成脚本模式，脚本模式不能转换成向导模式，我们已为您提供全部插件的模板。

运行数据集成任务

1. 您可以直接在界面运行数据集成任务，日志不会保存。
2. 提交之前需要进行调度配置，提交后一般第二天产生实例。详情请参见调度配置模块的文档。

说明：

- 您配置任务时可以设置相关调度参数。
- 测试同步任务时，不能直接调用调度配置中的参数，您需要提交后，才可以自动调用调度配置中配置的参数。

查看运行日志

您可以到运维中心查看运行结果。

说明：

- 您可以进入运维中心找到DAG图，右键查看运行日志。

- 在同步任务是幂等可自动重跑的前提下，如果您的任务运行失败，可以配置调度重跑，这样失败的任务可以自动重跑，增加系统稳定性。

2.1.3 基本概念

本文将为您介绍并发数、限速、脏数据和数据源等基本概念。

并发数

并发数是数据同步任务内，可从源并行读取或并行写入数据存储端的最大线程数。

限速

限速是数据集成同步任务最高能达到的传输速度。

脏数据

脏数据对于业务没有意义或者格式非法的数据。例如源端是Varchar类型的数据，写到Int类型的目标列中，导致因为转换不合理而无法写入的数据。

数据源

DataWorks所处理的数据的来源，可能是一个数据库或数据仓库。DataWorks支持各种类型的数据源，并且支持数据源之间的转换。

2.2 数据源配置

2.2.1 支持的数据源

数据集成是稳定高效、弹性伸缩的数据同步平台，为阿里云大数据计算引擎（MaxCompute、AnalyticDB和OSS等）提供离线、批量数据的进出通道。

数据同步支持的数据源类型如下表所示。

数据源分类	数据源类型	抽取（Reader）	导入（Writer）	支持方式	支持类型
关系型数据库	MySQL	支持，详情请参见#unique_44	支持，详情请参见#unique_45	向导/脚本	阿里云/自建
	SQL Server	支持，详情请参见#unique_47	支持，详情请参见#unique_48	向导/脚本	阿里云/自建
	PostgreSQL	支持，详情请参见#unique_50	支持，详情请参见#unique_51	向导/脚本	阿里云/自建
	Oracle	支持，详情请参见#unique_53	支持，详情请参见#unique_54	向导/脚本	自建
	DM（达梦）	支持	支持	脚本	自建

数据源分类	数据源类型	抽取（Reader）	导入（Writer）	支持方式	支持类型
	DRDS	支持，详情请参见#unique_57	支持，详情请参见#unique_58	向导/脚本	阿里云
	POLARDB	支持，详情请参见配置POLARDB Reader	支持，详情请参见#unique_61	向导/脚本	阿里云
	HybridDB for MySQL	支持，详情请参见#unique_63	支持，详情请参见#unique_64	向导/脚本	阿里云
	AnalyticDB for PostgreSQL	支持，详情请参见配置AnalyticDB for PostgreSQL Reader	支持，详情请参见配置AnalyticDB for PostgreSQL Writer	向导/脚本	阿里云
	DB2	支持，详情请参见#unique_68	支持，详情请参见#unique_69	脚本	自建
	RDS for PPAS	支持	支持	脚本	阿里云
大数据存储	MaxCompute（ODPS）	支持，详情请参见#unique_71	支持，详情请参见#unique_72	向导/脚本	阿里云
	DataHub	不支持	支持，详情请参见#unique_74	脚本	阿里云
	AnalyticDB（ADS）	支持，详情请参见配置AnalyticDB Reader	支持，详情请参见#unique_77	向导/脚本	阿里云
	Elasticsearch	支持，详情请参见#unique_78	支持，详情请参见配置Elasticsearch Writer	脚本	阿里云
非结构化存储	OSS	支持，详情请参见#unique_81	支持，详情请参见#unique_82	向导/脚本	阿里云
	HDFS	支持，详情请参见#unique_84	支持，详情请参见#unique_85	脚本	自建
	FTP	支持，详情请参见#unique_87	支持，详情请参见#unique_88	向导/脚本	自建
NoSQL	MongoDB	支持，详情请参见#unique_90	支持，详情请参见#unique_91	脚本	阿里云/自建

数据源分类	数据源类型	抽取（Reader）	导入（Writer）	支持方式	支持类型
	Memcache（OCS）	不支持	支持，详情请参见#unique_93	脚本	阿里云/自建Memcached
	Redis	不支持	支持，详情请参见#unique_95	脚本	阿里云/自建
	Table Store（OTS）	支持，详情请参见#unique_97	支持，详情请参见#unique_98	脚本	阿里云
	OpenSearch	不支持	支持，详情请参见#unique_99	脚本	阿里云
	HBase	支持，详情请参见#unique_100	支持，详情请参见#unique_101	脚本	阿里云/自建
消息队列	LogHub	支持，详情请参见#unique_103	支持，详情请参见#unique_104	向导/脚本	阿里云
性能测试	Stream	支持，详情请参见#unique_105	支持，详情请参见#unique_106	脚本	-

2.2.2 数据源测试连通性

本文将为您介绍支持连通性测试的数据源类型，以及数据源连通性测试常见问题示例。

数据源	数据源类型	网络类型	是否支持测试连通性	是否添加自定义资源组
MySQL	云数据库	经典网络	支持	-
		专有网络	支持	-
	连接串模式（数据集成网络可直接连通）		支持	-
	连接串模式（数据集成网络不可直接连通）		不支持	是
	ECS自建	经典网络	支持	-
		专有网络	不支持	是
SQL Server	云数据库	经典网络	支持	-
		专有网络	支持	-
	连接串模式（数据集成网络可直接连通）		支持	-
	连接串模式（数据集成网络不可直接连通）		不支持	是
	ECS自建	经典网络	支持	-

数据源	数据源类型	网络类型	是否支持测试连通性	是否添加自定义资源组
		专有网络	不支持	是
PostgreSQL	云数据库	经典网络	支持	-
		专有网络	支持	-
	连接串模式（数据集成网络可直接连通）		支持	-
	连接串模式（数据集成网络不可直接连通）		不支持	是
	ECS自建	经典网络	支持	-
		专有网络	不支持	是
Oracle	连接串模式（数据集成网络可直接连通）		支持	-
	连接串模式（数据集成网络不可直接连通）		不支持	是
	ECS自建	经典网络	支持	-
		专有网络	不支持	是
DRDS	云数据库	经典网络	支持	-
		专有网络	排期中	是
HybridDB for MySQL	云数据库	经典网络	支持	-
		专有网络	排期中	是
HybridDB for PostgreSQL	云数据库	经典网络	支持	-
		专有网络	排期中	是
MaxCompute（对应odps数据源）	云数据库	经典网络	支持	-
AnalyticDB（对应ADS数据源）	云数据库	经典网络	支持	-
		专有网络	排期中	是
OSS	云数据库	经典网络	支持	-
		专有网络	支持	-
Hdfs	连接串模式（数据集成网络可直接连通）		支持	-
	ECS自建	经典网络	支持	-
		专有网络	不支持	是

数据源	数据源类型	网络类型	是否支持测试连通性	是否添加自定义资源组
FTP	连接串模式（数据集成网络可直接连通）		支持	-
	连接串模式（数据集成网络不可直接连通）		不支持	是
	ECS自建	经典网络	支持	-
		专有网络	不支持	是
MongoDB	云数据库	经典网络	支持	-
		专有网络	排期中	是
	连接串模式（数据集成网络可直接连通）		支持	-
	ECS自建	经典网络	支持	-
		专有网络	不支持	是
Memcache	云数据库	经典网络	支持	-
		专有网络	排期中	是
Redis	云数据库	经典网络	支持	-
		专有网络	排期中	是
	连接串模式（数据集成网络可直接连通）		支持	-
	ECS自建	经典网络	支持	-
		专有网络	不支持	是
Table Store（对应OTS数据源）	云数据库	经典网络	支持	-
		专有网络	排期中	是
DataHub	云数据库	经典网络	支持	-
		专有网络	不支持	-

 说明:

是否添加自定义资源组，请参见#unique_33。

上述表格中的-表示没有此种说法，不支持并不代表不能配置同步任务，只是单击测试连通性无效，需要添加自定义资源组。

- VPC环境数据源
  - VPC环境的RDS数据源支持测试连通性。
  - 其他数据源VPC网络正在排期中。
  - 金融云网络暂时不支持测试连通性。
- ECS自建数据源
  - 经典网络支持JDBC的格式测试连通性，通常走公网。
  - VPC环境暂时不支持测试连通性。
  - 跨区域暂时不支持测试连通性。
  - 金融云网络暂时不支持测试连通性。

  关于ECS自建的数据源，需要特别注意安全组的添加。在ECS安全组中入/出方向添加调度集群的IP（公网和经典网络都要在对应的入/出方向添加），如果没有添加相应的安全组，数据同步时，会出现连接不上的问题。详情请参见#unique_108。

  大的端口范围无法在ECS安全组界面添加，请使用ECS的安全组API进行添加，详情请参见AuthorizeSecurityGroup。
- 没连接串模式（数据集成网络可直接连通）本地IDC机房或ECS搭建的数据源
  - 不支持测试连通性。
  - 配置同步任务要添加自定义资源组。

    更多详情请参见连接串模式（数据集成网络不可直接连通）数据库的数据上云。
- 连接串模式（数据集成网络可直接连通）本地IDC机房或ECS搭建的数据源

  一律走公网JDBC格式，如果测试连通性失败，则检查您本地网络的限制或者数据库本身的限制。

调度集群

- 目前调度集群在华东2、华南1、中国（香港）、新加坡均有部署，以调度集群在华东2为准和用户数据源进行对比。假设用户的MongoDB数据源在华北经典网络，以调度集群在华东2经典网络为准，跨区域连接不通。
- OXS集群和ECS集群，内网不通。

  RDS的调度集群是OXS，OXS集群和内网大陆所有区域的RDS互通。其他数据源的调度集群是另外一套ECS经典网络的调度集群。

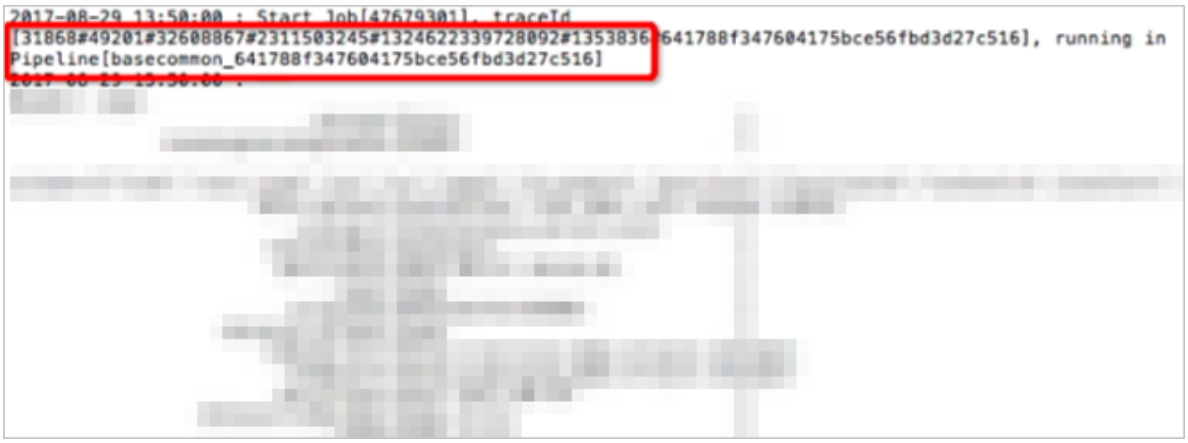
  比如RDS同步到自建数据库测试时，RDS和自建数据库数据源测试连通性都能成功。但实际调度时，RDS会下发到OXS调度集群，自建的会跑到ECS集群，RDS和ECS集群不通，所以会失败。通常建议您将RDS改为MySQL>JDBC方式，这样都会跑到ECS集群上连接是可以通的。

如何查看任务下发执行集群

- 出现RDS作为数据源时，任务会到OXS集群同步。

- 当数据源为其他数据源在ECS调度集群时。

- 当调度集群为自定义调度资源时，日志如下图所示（非常重要，用于判断用户是否是自定义资源组）。

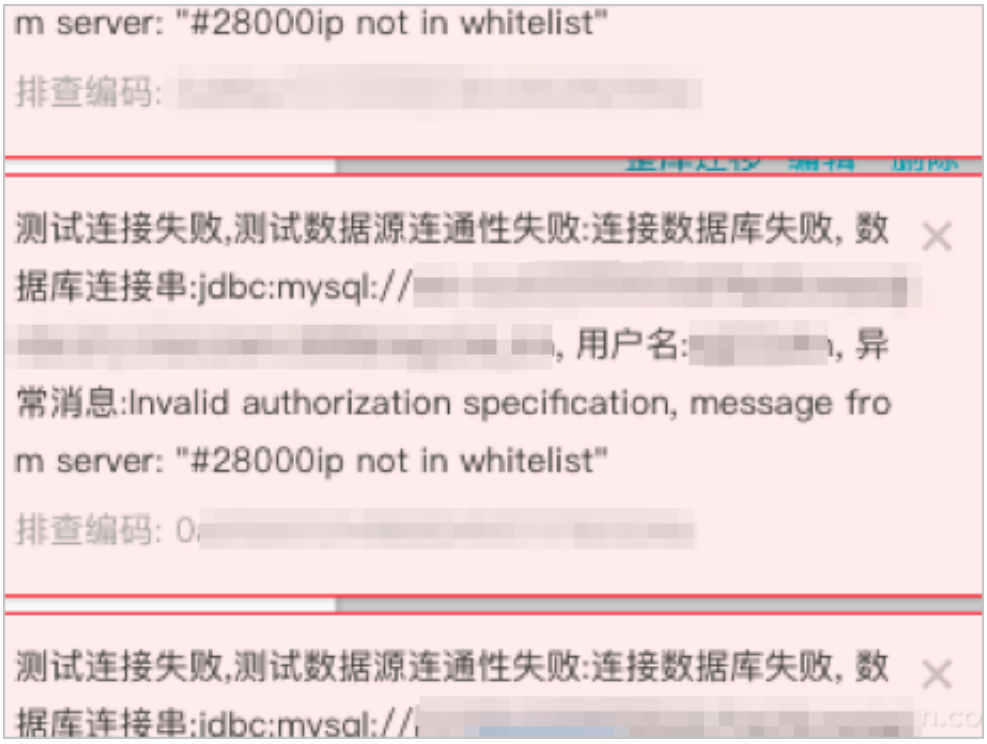


- 进入数据集成测试页面，直接单击运行，统一走的是ECS调度集群，所以会有用户反馈，RDS相关任务手动运行成功，调度失败。因为RDS作为数据源跨区域时，需要在OXS调度集群执行。所以需要您选择调度运维 > 测试运行。

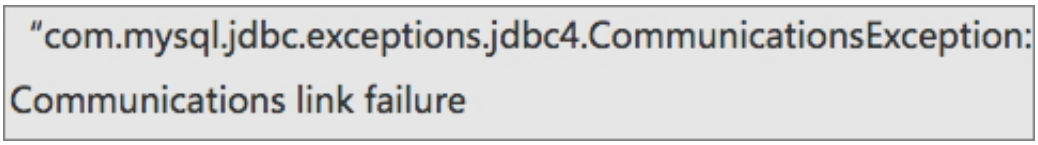
测试连通性失败的常见场景

当测试连通性失败时，需核实数据源区域、网络类型、RDS白名单是否添加完整实例ID、数据库名称和用户名是否正确。如果您的测试连通性失败，可以首先参见#unique_109及#unique_110进行排查。常见错误示例如下所示：

- 数据库密码错误，如下所示。



- 网络不通错误，如下所示。



- 同步过程中出现网络断开等情况。

  首先要查看完整日志，确定是哪个调度资源，是否是自定义资源。

  如果是自定义资源，核实自定义资源组的IP是否添加到数据源比如RDS白名单（MongoDB也有白名单限制，需要添加）。

  核实两端数据源连通性是否通过，核实RDS，MongoDB白名单是否会添加完整（如果不完整，有时会成功有时会失败，如果任务下发到已添加的调度服务器上会成功，没添加的会失败）。

- 任务显示成功，但是日志出现8000断开报错。

  出现上述报错，是因为您使用的自定义调度资源组，没有对10.116.134.123，访问8000端口在安全组内网入方向放行，添加后重新运行即可。

测试连通性失败的示例

示例一

- 问题现象

  测试连接失败，测试数据源连通性失败。连接数据库失败，数据库连接串：`jdbc:mysql://xx.xx.xx.x:xxxx/t_uoer_bradef`，用户名：xxxx_test，异常消息：`Access denied for user ‘xxxx_test’@’%’ to database ‘yyyy_demo’`。

- 排查思路

  1. 确认其添加的信息是否有问题。
  2. 密码、白名单或者用户的账号是否具有对应数据库的权限，RDS管控台可以添加授权。

示例二

- 问题现象

  测试连接失败，测试数据源连通性失败。报错如下：

```
error message: Timed out after 5000 ms while waiting for a server
that matches ReadPreferenceServerSelector{readPreference=primary}.
Client view of cluster state is {type=UNKNOWN, servers=[(xxxxxxxxxx
), type=UNKNOWN, state=CONNECTING, exception={com.mongodb.MongoSocke
tReadException: Prematurely reached end of stream}}]
```

- 排查思路

  非VPC的MongoDB，添加MongoDB数据源测试连通性要添加相应的白名单，详情请参见#unique_111。

2.2.3 数据源隔离

数据源隔离模式可以满足标准模式下，开发环境和生产环境的数据隔离需求。

同一个名称的数据源存在开发环境和生产环境两套配置，可以通过数据源隔离使其在不同环境隔离使用。

> 说明:
> 目前只有标准模式的工作空间支持数据源隔离。

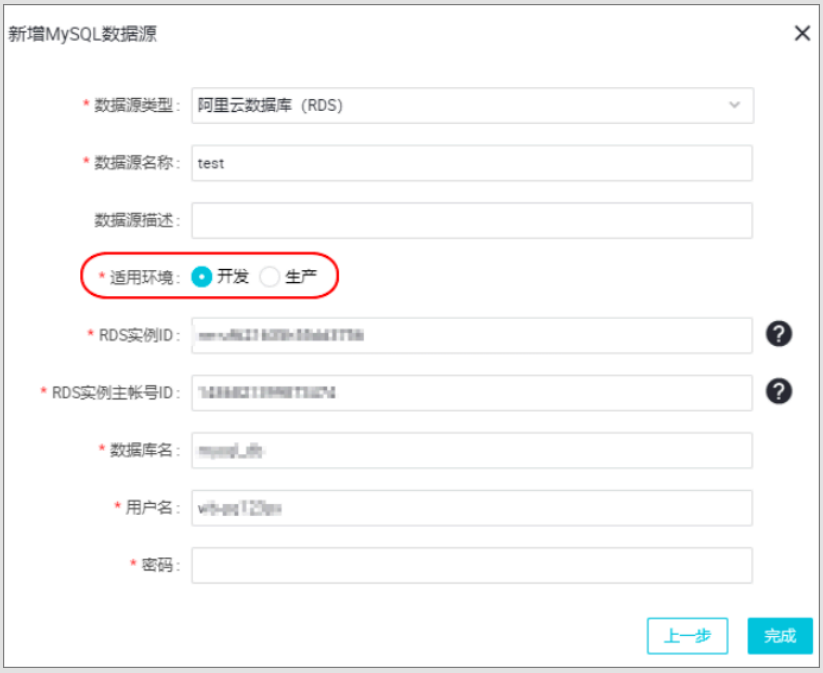
配置数据同步任务时会使用开发环境的数据源，提交生产运行时会使用生产环境的数据源。如果您要将任务提交到生产环境调度，同一个数据源名需要同时添加生产环境和开发环境的数据源配置。

新增数据源隔离模式后，对工作空间有以下影响。

- 简单模式：数据源功能和界面与之前保持一致，详情请参见数据源配置。
- 标准模式：数据源界面按照数据源隔离模式进行相应调整，增加了适用环境的参数。

- 简单模式升级成标准模式：进行模式升级时，会提示对数据源进行升级，将数据源拆分成生产环境和开发环境隔离的模式。

页面功能	说明
多库多表搬迁	单击多库多表搬迁，可直接跳转至批量上云页面。 **说明：** 必须确保生产环境和开发环境都存在数据源，且数据源测试连通性成功，方可在批量上云页面选择相应的数据源。
批量新增数据源	目前仅支持MySQL、SQLServer和Oracle数据源。模板内容：显示数据源类型、数据源名称、数据源描述、环境类别（0开发、1生产）、链接地址，您可根据模板中的格式填写内容，选择上传文件进行新建操作，文本框中会显示添加详情。

页面功能	说明
新增数据源	· 开发环境可用的数据源：可在新建数据同步节点时选择并在开发环境运行，但无法提交到生产环境或在生产环境运行。 · 生产环境可用的数据源：只允许在生产环境运行时使用，不可在新建数据同步节点时选择。 **说明:** 同一个开发环境和生产环境的数据源名称必须相同。 
适用环境	简单模式下的工作空间不显示此列，标准模式下的工作空间，用以展现数据源适用的环境。

页面功能	说明
操作	· 整库迁移批量配置：该按钮仅对开发环境的数据源显示。 · 新建：若不存在适用环境下的数据源，显示新建按钮。 · 编辑/删除：若存在适用环境下的数据源，则显示编辑和删除按钮。 - 删除开发环境和生产环境的数据源：需确认是否存在生产环境关联的同步任务，操作不可逆，删除后，在开发环境配置同步任务时此数据源不可见。 如果生产环境在使用此数据源配置的同步任务，删除后，生产环境任务不可正常运行。请删除同步任务后再删除此数据源。 - 删除开发环境的数据源：需确认是否存在生产环境关联的同步任务，操作不可逆，删除后，在开发环境配置同步任务时此数据源不可见。 若生产环境在使用此数据源配置的同步任务，删除后，任务编辑时将不能获取到元数据信息，但生产环境任务可以正常运行。 - 删除生产环境的数据源：需确认是否存在生产环境关联的同步任务，删除后，在开发环境使用此数据源配置的同步任务将不能提交生产发布。 若生产环境在使用此数据源配置的同步任务，删除后，生产环境任务不可正常运行。
选择	勾选后，可以进行批量测试连通性和批量删除操作。

2.2.4 配置AnalyticDB数据源

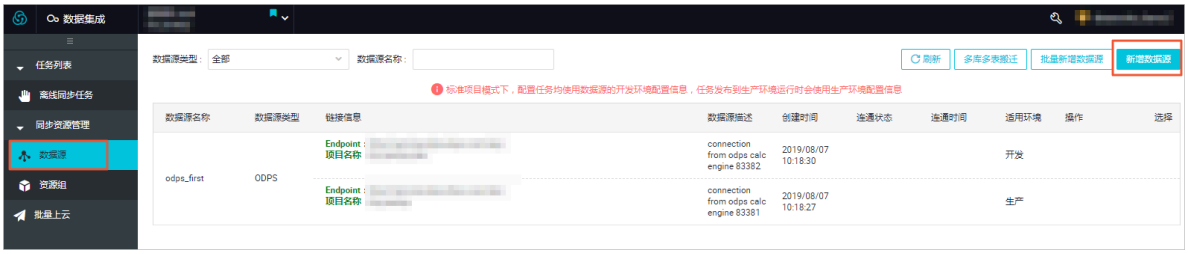
AnalyticDB为您提供其他数据源向AnalyticDB写入的功能，但不能读取数据，支持数据集成中的向导模式和脚本模式。

> 说明:
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击相应工作空间操作栏中的进入数据集成。

2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新建数据源弹出框中，选择数据源类型为AnalyticDB（ADS）。

4. 填写AnalyticDB数据源的各配置项。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 **说明:** 仅标准模式工作空间会显示此配置。
连接Url	AnalyticDB连接信息，格式为Address:Port。

配置	说明
数据库	AnalyticDB的数据库名称。
AccessKey ID/ AceessKey Secret	访问秘匙（AccessKey ID和AccessKey Secret），相当于登录密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

   提供测试连通性能力，可以判断输入的信息是否正确 。

后续步骤

现在，您已经学习了如何配置AnalyticDB数据源，您可以继续学习下一个教程。在该教程中您将学习如何配置AnalyticDB Writer插件，详情请参见配置AnalyticDB Writer。

2.2.5 配置SQL Server数据源

SQL Server数据源为您提供读取和写入SQL Server双向通道的功能，您可以通过向导模式和脚本模式配置同步任务。

说明:

标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源，并进行隔离，以保护您的数据安全。

目前仅支持SQL Server 2005及以上版本。如果是VPC环境下的SQL Server，需要注意以下问题：

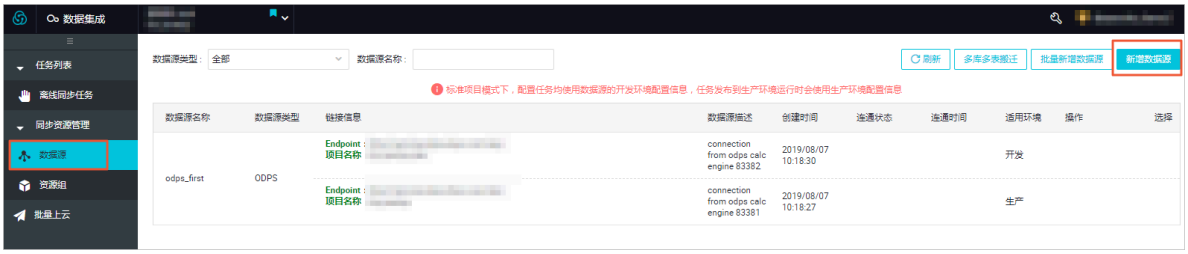
- 自建的SQL Server数据源。
  - 不支持测试连通性，但仍支持配置同步任务，创建数据源时，直接单击完成即可。
  - 必须使用自定义调度资源组运行对应的同步任务，请确保自定义资源组可以连通您的自建数据库，详情请参见#unique_34和#unique_35。
- 通过RDS创建的SQL Server数据源。

  您无需选择网络环境，系统会根据您填写的RDS实例信息，自动进行判断。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击对应工作空间操作栏中的进入数据集成。

2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为SQL Server。

4. 填写SQL Server数据源的各配置项。

   SQL Server数据源类型包括阿里云数据库（RDS）、连接串模式（数据集成网络可直接连通）和连接串模式（数据集成网络不可直接连通），您可以根据自身需求进行选择。

   · 以新增SQL Server > 阿里云数据库（RDS）类型的数据源为例。

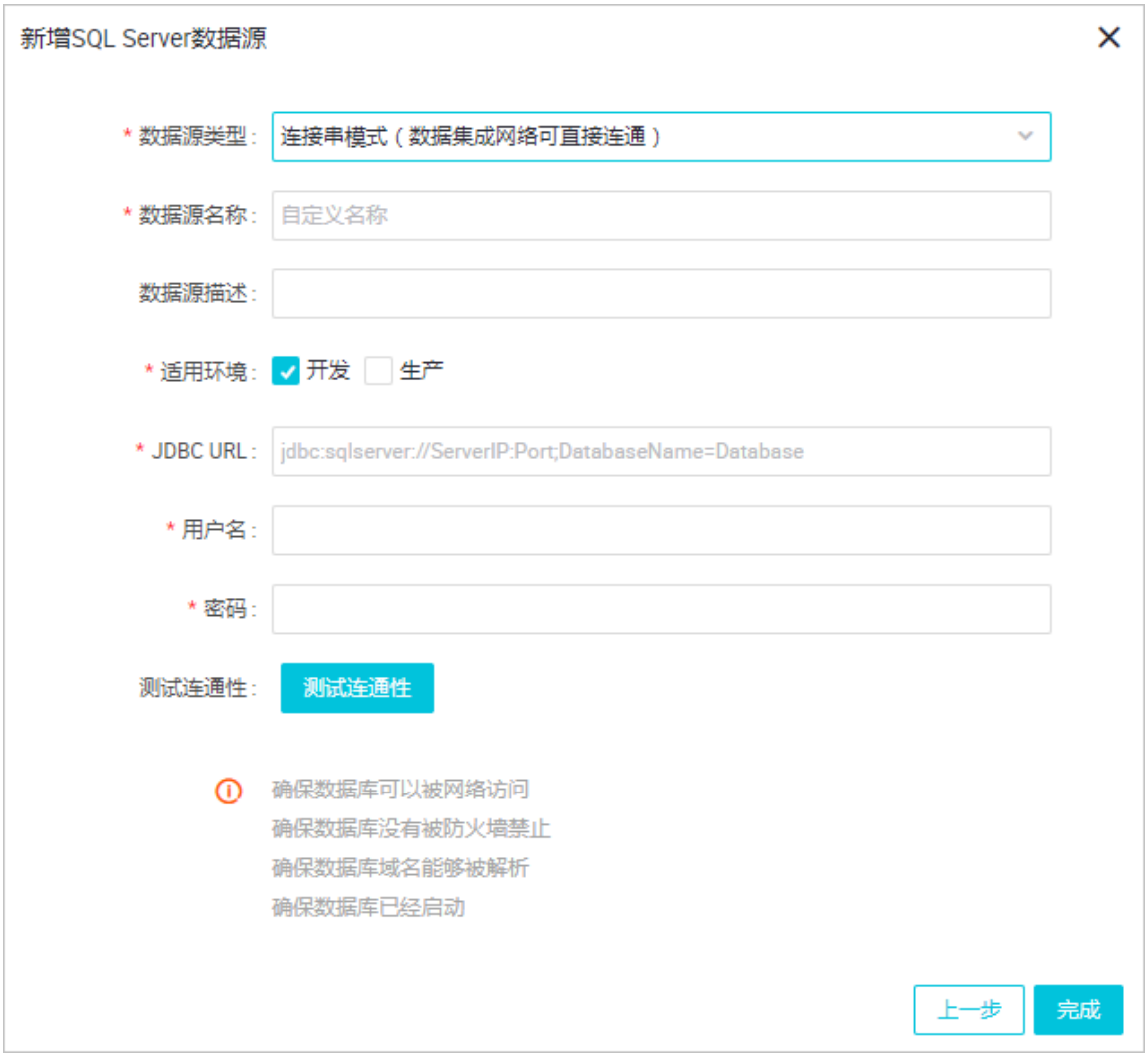
配置	说明
数据源类型	当前选择的数据源类型为SQL Server > 阿里云数据库（RDS）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 **说明:** 仅标准模式工作空间会显示此配置。
地区	选择相应的Region。
RDS实例ID	您可以进入RDS的控制台，查看RDS的实例ID。 
RDS实例主账号ID	您可以进入RDS控制台的安全设置页面，查看相应的信息。 
数据库名	填写对应的数据库名称。
用户名/密码	数据库对应的用户名和密码。

说明:

您需要先添加RDS白名单才能连接成功，详情请参见添加白名单。

- 以新增SQL Server > 连接串模式（数据集成网络可直接连通）类型的数据源为例。



配置	说明
数据源类型	当前选择的数据源类型为SQL Server > 连接串模式（数据集成网络可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
JDBC URL	JDBC连接信息，格式为`jdbc:sqlserver://ServerIP:Port;DatabaseName=Database`。

配置	说明
用户名/密码	数据库对应的用户名和密码。

- 以新增SQL Server > 连接串模式（数据集成网络不可直接连通）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为SQL Server > 连接串模式（数据集成网络不可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
资源组	可以用于执行同步任务，通常添加资源组时可以绑定多台机器。详情请参见#unique_33。
JDBC URL	JDBC连接信息，格式为`jdbc:sqlserver://ServerIP:Port;DatabaseName=Database`。
用户名/密码	数据库对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络下，如果您使用实例模式配置数据源，可以判断输入的信息是否正确。
- 专有网络下，如果您将VPC内部地址作为JDBC URL添加数据源，测试连通性会报告失败。
- 经典网络/专有网络下，如果您将数据源的公网地址作为JDBC URL添加数据源，可以判断输入的信息是否正确。

后续步骤

现在，您已经学习了如何配置SQL Server数据源，您可以继续学习下一个教程。在该教程中您将学习如何配置SQL Server插件，详情请参见 #unique_48和#unique_47。

2.2.6 配置MongoDB数据源

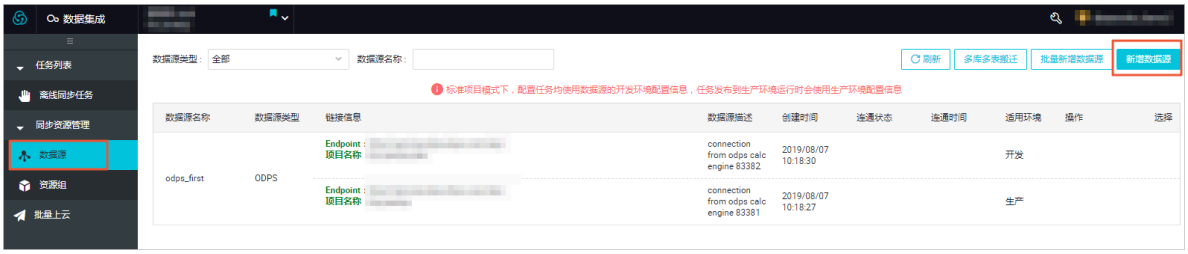
MongoDB是目前仅次于Oracle、MySQL的文档型数据库，为您提供读取和写入MongoDB双向通道的功能，您可以通过脚本模式配置同步任务。

说明:
标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击相应工作空间操作栏中的进入数据集成。

2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为MongoDB。

4. 填写MongoDB数据源的各配置项。

   MongoDB数据源类型包括实例模式（阿里云数据源）和连接串模式（数据集成网络可直接连通）。

   - 实例模式（阿里云数据源）：通常使用经典网络类型，同地区的经典网络可以连通，跨地区的经典网络不保证可以连通。
   - 连接串模式（数据集成网络可直接连通）：通常使用公网类型，可能产生一定的费用。

   以新增MongDB > 实例模式（阿里云数据源）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为MongDB > 实例模式（阿里云数据源）。  说明: 如果您尚未授权数据集成系统默认角色，需要主账号前往RAM进行角色授权，然后刷新此页面。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。  说明: 仅标准模式工作空间会显示此配置。
地区	是指在购买MongoDB时所选择的区域。
实例ID	您可以在MongoDB控制台查看MongoDB实例ID。
数据库名	您可以在MongoDB控制台新建数据库，设置相应的数据名、用户名和密码。

配置	说明
用户名/密码	数据库对应的用户名和密码。

以新增MongDB > 连接串模式（数据集成网络可直接连通）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为MongDB > 连接串模式（数据集成网络可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
访问地址	格式为host:port。如果此处您需要同时添加多个地址，可以单击添加访问地址进行添加。 说明: 添加的访问地址必须全部为公网地址或全部为私网地址，不可以公网、私网地址混杂。
数据库名	该数据源对应的数据库名称。
用户名/密码	数据库对应的用户名和密码。

说明:

连接串模式（数据集成网络不可直接连通）的数据库，可以通过下述操作添加MongoDB数据源。

a. 选择数据源类型为连接串模式（数据集成网络可直接连通）。

b. 填写新增MongoDB数据源对话框中的配置项，其中访问地址填写您的内网地址。

c. 添加完成后，不需要进行连通性测试，单击完成。

d. 添加自定义资源组，将任务运行在自定义资源组上，详情请参见#unique_33。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

说明:

- VPC环境的MongoDB云数据库，添加连接串模式（数据集成网络可直接连通）数据源类型并保存。
- VPC环境不支持测试连通性。

后续步骤

现在，您已经学习了如何配置MongoDB数据源，您可以继续学习下一个教程。在该教程中您将学习如何配置MongoDB插件。详情请参见#unique_90和#unique_91。

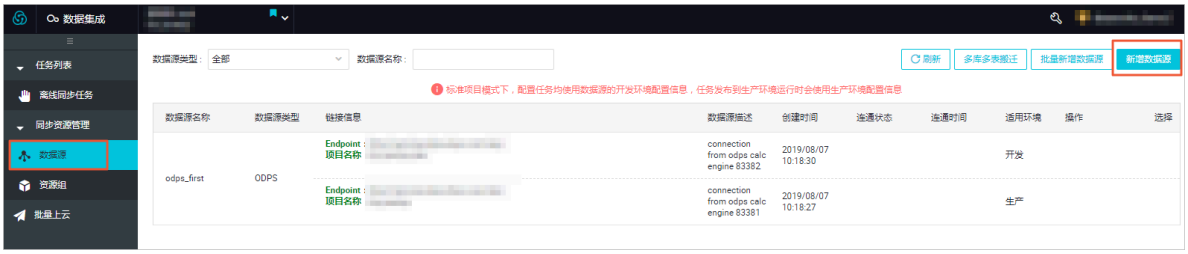
2.2.7 配置DataHub数据源

DataHub为您提供完善的数据导入方案，能够快速解决海量数据的计算问题。DataHub数据源作为数据中枢，为您提供其它数据源将数据写入DataHub的功能，支持DataHub Writer插件。

> **说明:**
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

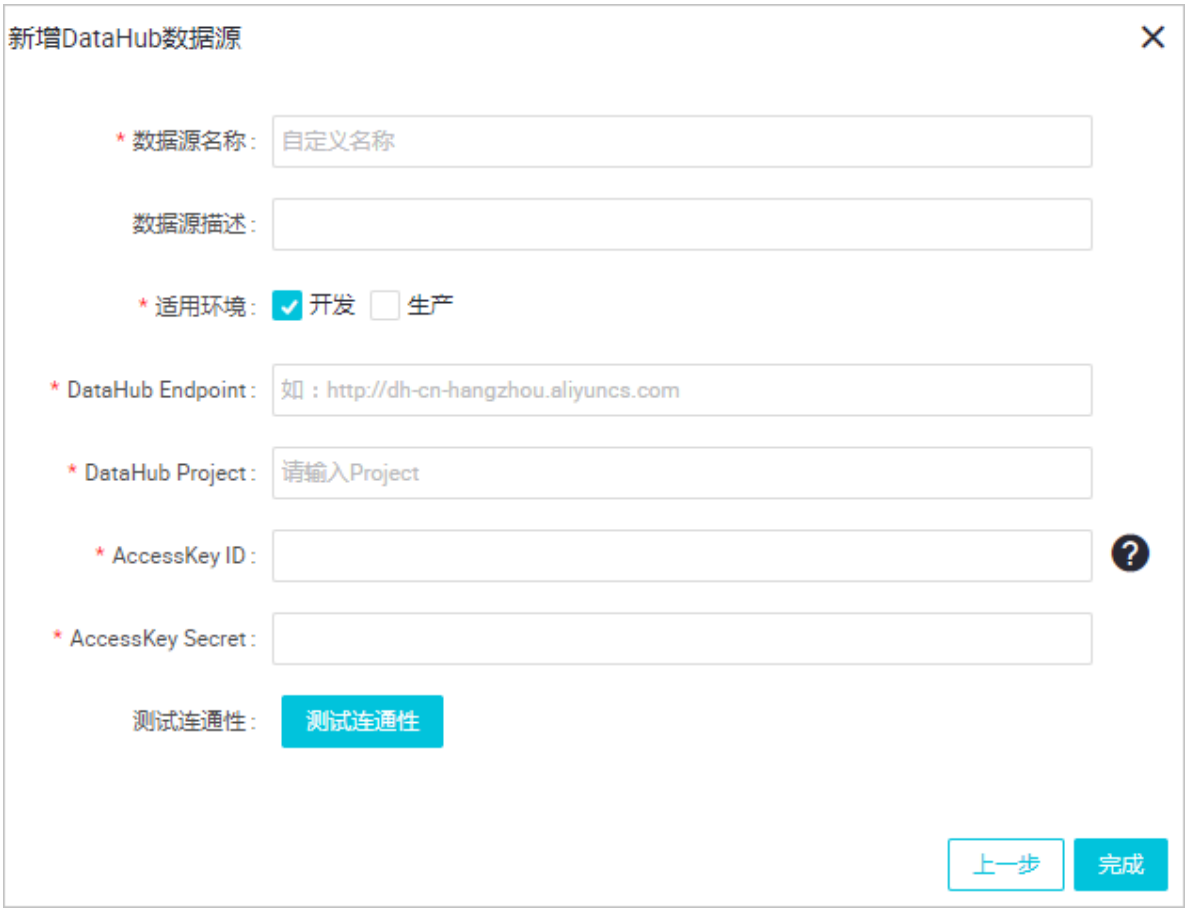
操作步骤

1. 以项目管理员身份进入DataWorks控制台，单击相应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为DataHub。

4. 填写DataHub数据源的各配置项。



配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源的简单描述，不超过80个字。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
DataHub Endpoint	默认只读，从系统配置中自动读取。
DataHub Project	对应的DataHub Project标识。
AccessKey ID/ AceessKey Secret	访问秘匙（AccessKeyID和AccessKeySecret），相当于登录密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

   提供测试连通性能力，可以判断输入的信息是否正确 。

后续步骤

现在，您已经学习了如何配置DataHub数据源，您可以继续学习下一个教程。在该教程中，您将学习如何配置DataHub Writer插件，详情请参见#unique_74。

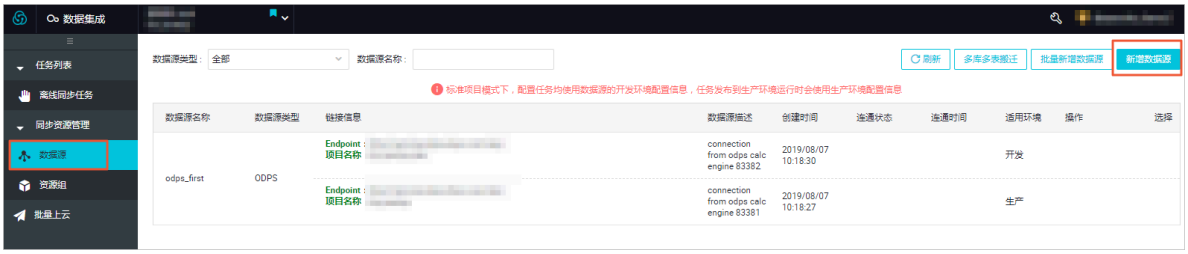
2.2.8 配置DM数据源

DM数据源为您提供读取和写入DM双向通道的功能，您可以通过向导模式和脚本模式配置同步任务。

> **说明:**
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击相应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。

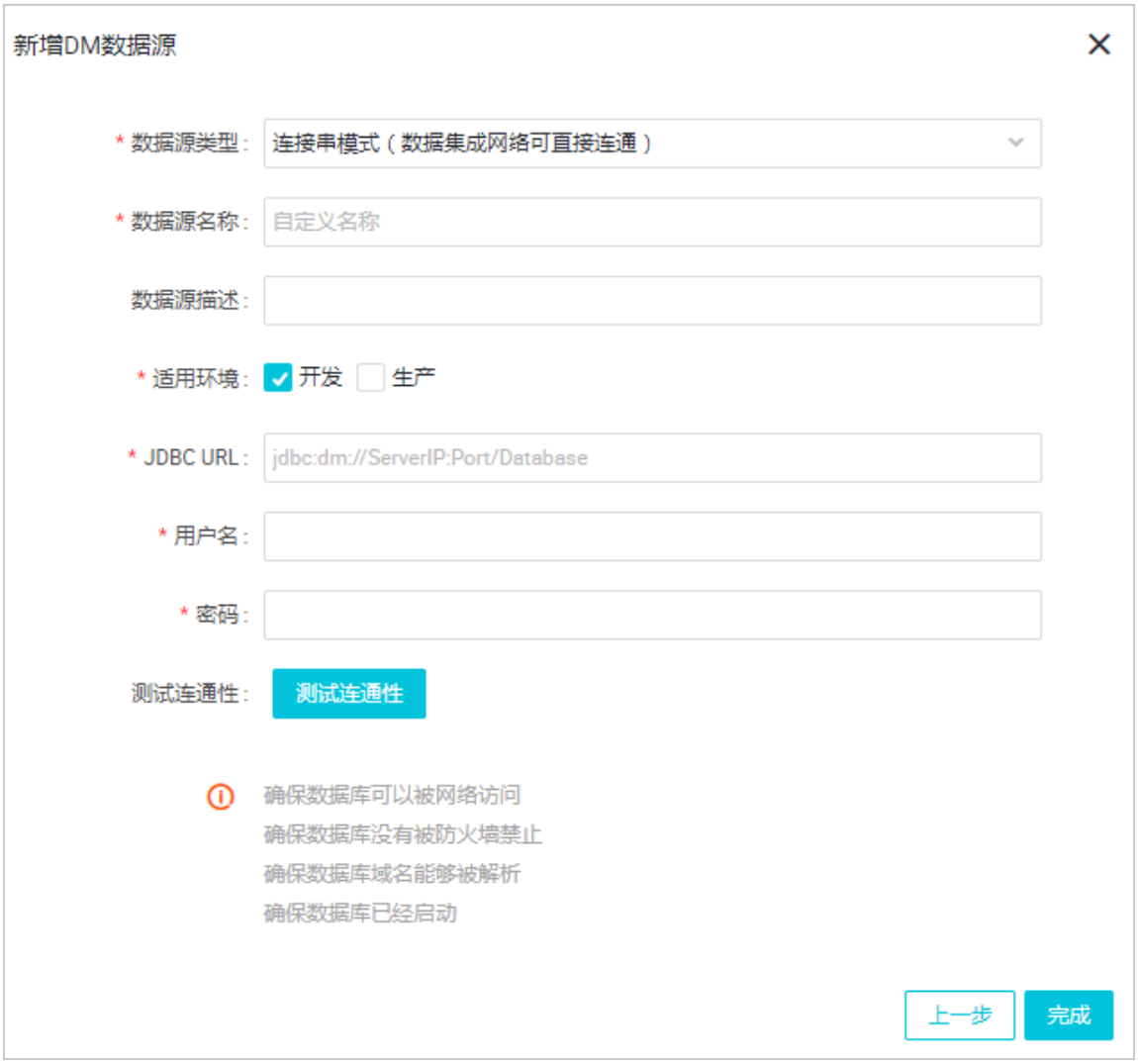


3. 在新增数据源弹出框中，选择数据源类型为DM。

4. 填写DM数据源的各配置项。

   DM数据源类型包括连接串模式（数据集成网络可直接连通）和连接串模式（数据集成网络不可直接连通），您可以根据自身需求进行选择。

   · 以新增DM > 连接串模式（数据集成网络可直接连通）类型的数据源为例。



配置	说明
数据源类型	当前选择的数据源类型为DM > 连接串模式（数据集成网络可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。

配置	说明
JDBC URL	JDBC连接信息，格式为`jdbc:mysql://ServerIP:Port/Database`。
用户名/密码	数据库对应的用户名和密码。

- 以新增DM > 连接串模式（数据集成网络不可直接连通）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为DM > 连接串模式（数据集成网络不可直接连通）。选择此类型的数据源，需要使用自定义调度资源才能进行同步，您可以单击帮助手册查看详情。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
资源组	可以用于执行同步任务，通常添加资源组时可以绑定多台机器。详情请参见#unique_33。
JDBC URL	JDBC连接信息，格式为`jdbc:mysql://ServerIP:Port/Database`。
用户名/密码	数据库对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

   提供测试连通性能力，可以判断输入的信息是否正确 。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络和连接串模式（数据集成网络不可直接连通）下，目前不支持数据源连通性测试，直接单击完成。

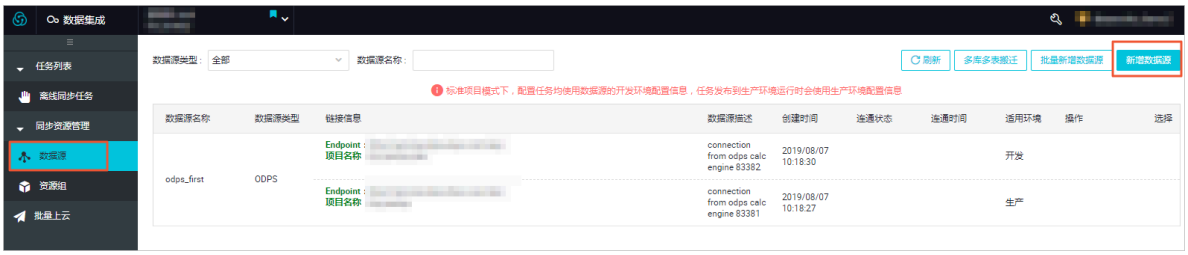
2.2.9 配置DRDS数据源

DRDS（分布式RDS）数据源为您提供读取和写入DRDS双向通道的功能，您可以通过向导模式和脚本模式配置同步任务。

说明:
标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击相应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为DRDS。
4. 填写DRDS数据源的各配置项。

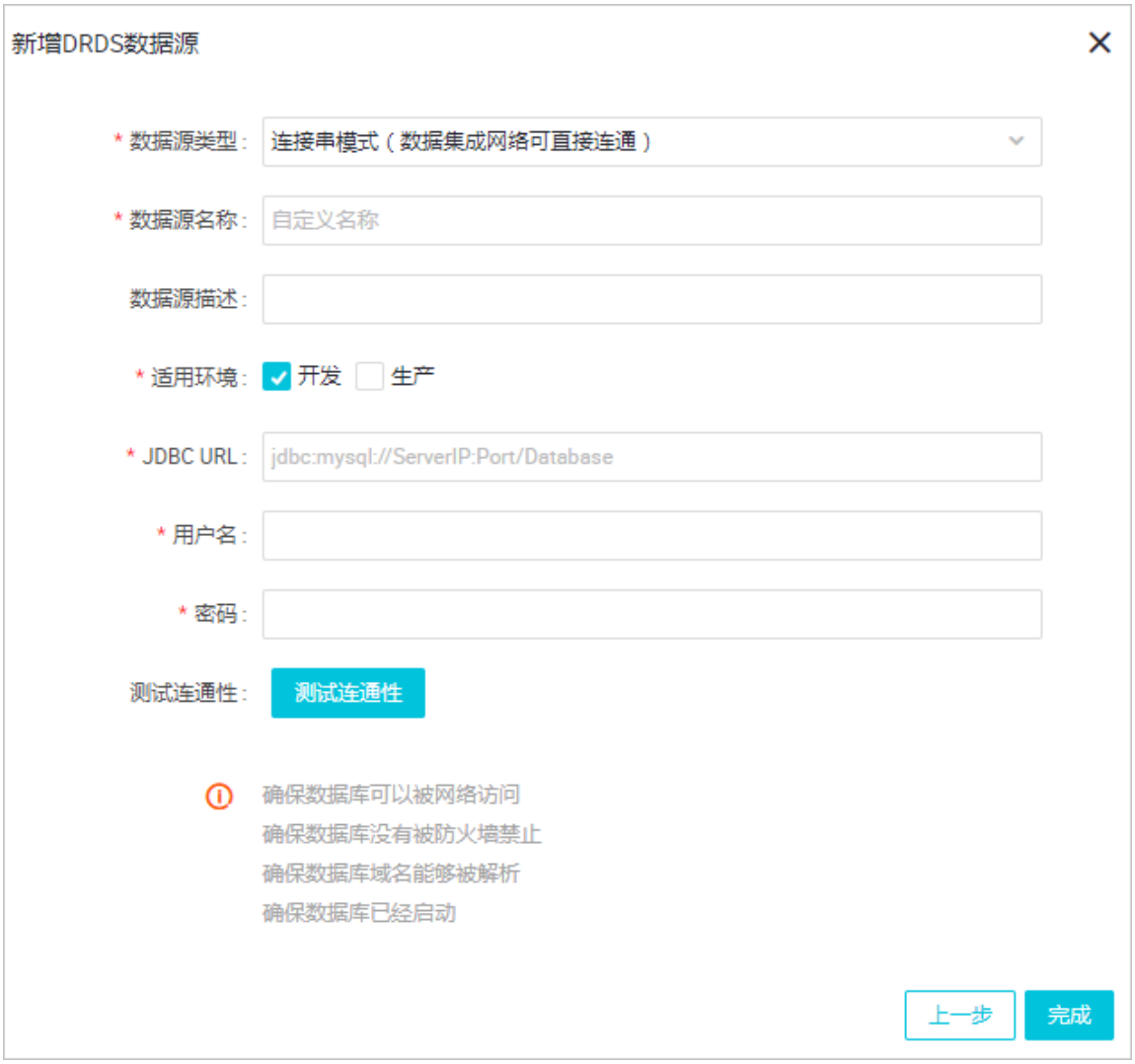
   DRDS数据源类型包括阿里云数据库（DRDS）和连接串模式（数据集成网络可直接连通），您可以根据自身需求进行选择。

   · 以新增DRDS > 阿里云数据库（DRDS）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为DRDS > 阿里云数据库（DRDS）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
实例ID	您可以进入DRDS控制台查看相关实例ID。
主账号ID	您可以进入DRDS控制台的安全设置页面，查看相应的信息。
数据库名	填写数据库对应的名称。
用户名/密码	数据库对应的用户名和密码。

- 以新增DRDS > 连接串模式（数据集成网络可直接连通）类型的数据源为例。



配置	说明
数据源类型	当前选择的数据源类型为DRDS > 连接串模式（数据集成网络可直接连通）。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
JDBC URL	JDBC连接信息，格式为`jdbc:sqlserver://ServerIP:Port;DatabaseName=Database`。
用户名/密码	数据库对应的用户名和密码。

说明:
对于连接串模式（数据集成网络可直接连通）的数据源，您需要添加白名单才能连接成功。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

   提供测试连通性能力，可以判断输入的信息是否正确 。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络下，如果您使用实例模式配置数据源，可以判断输入的信息是否正确。
- 专有网络下，如果您将VPC内部地址作为JDBC URL添加数据源，测试连通性会报告失败。
- 经典网络/专有网络下，如果您将数据源的公网地址作为JDBC URL添加数据源，可以判断输入的信息是否正确。

后续步骤

现在，您已经学习了如何配置DRDS数据源，您可以继续学习下一个教程。在该教程中您将学习如何配置DRDS插件。详情请参见#unique_58和#unique_57。

2.2.10 配置FTP数据源

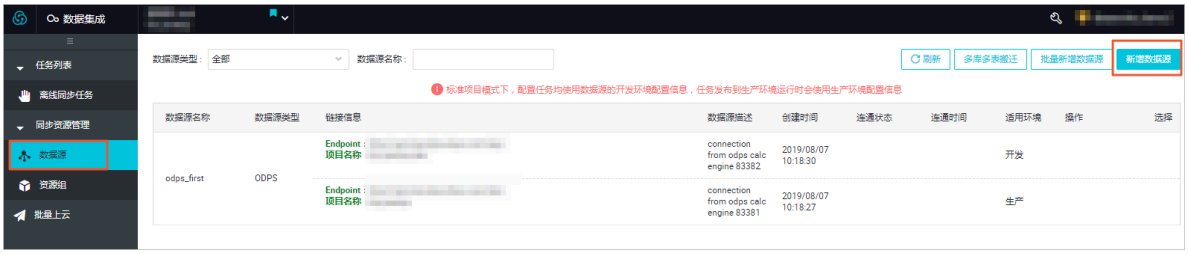
FTP数据源为您提供读取和写入FTP双向通道的功能，您可以通过向导模式和脚本模式配置同步任务。

说明:

> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源，并进行隔离，以保护您的数据安全。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。

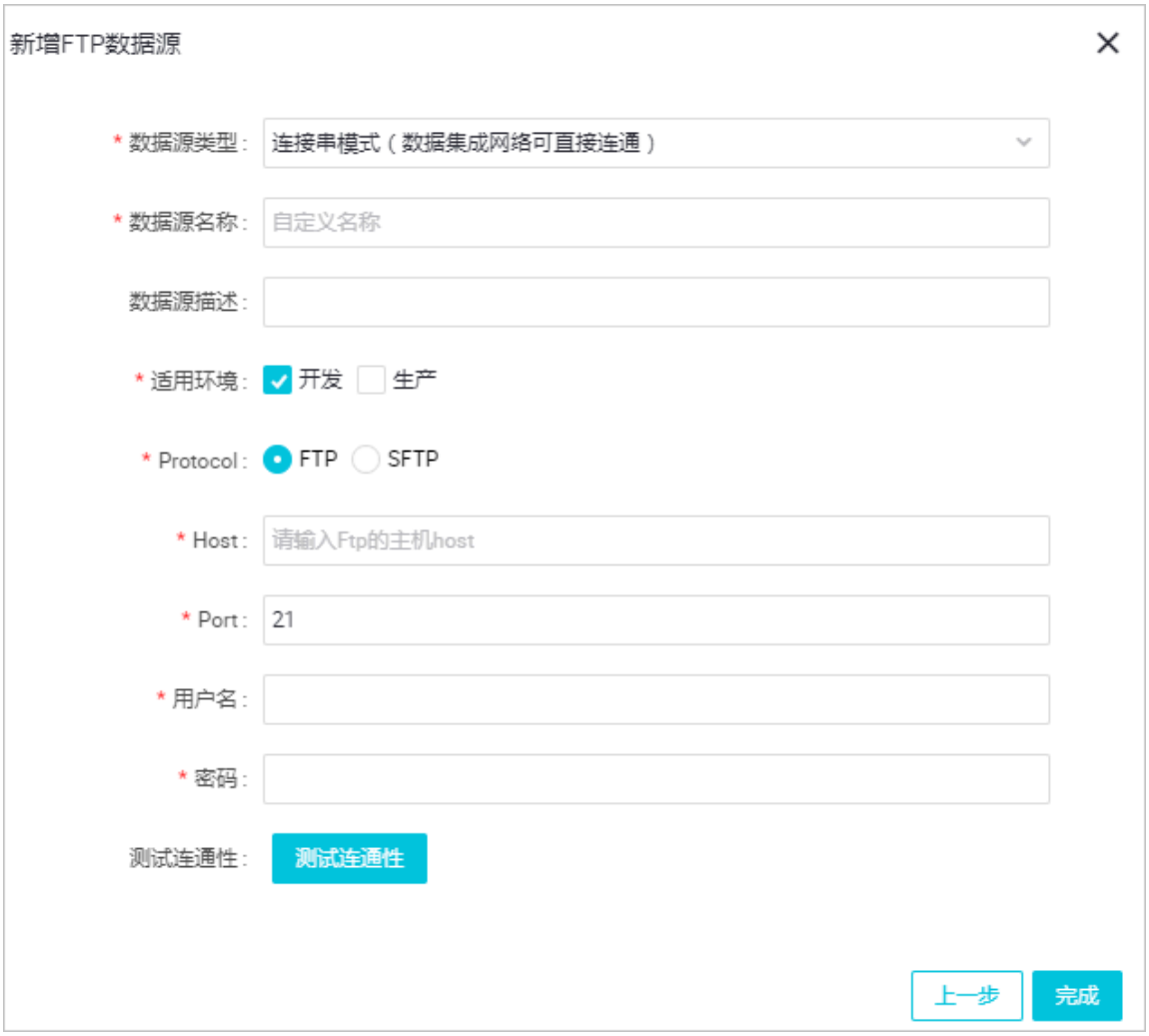


3. 在新增数据源弹出框中，选择数据源类型为FTP。

4. 填写FTP数据源的各配置项。

   FTP数据源类型包括连接串模式（数据集成网络可直接连通）和连接串模式（数据集成网络不可直接连通），您可以根据自身需求进行选择。

   · 以新增FTP > 连接串模式（数据集成网络可直接连通）类型的数据源为例。



配置	说明
数据源类型	当前选择的数据源类型为FTP > 连接串模式（数据集成网络可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。

配置	说明
Portocol	目前仅支持FTP和SFTP协议。
Host	对应FTP主机的IP地址。
Port	如果选择的是FTP协议，则端口默认为21。如果选择的是SFTP协议，则端口默认为22。

配置	说明
用户名/密码	访问该FTP服务的账号密码。

- 以新增FTP > 连接串模式（数据集成网络不可直接连通）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为FTP > 连接串模式（数据集成网络不可直接连通）。 选择此类型的数据源需要使用自定义调度资源才能进行同步，您可以单击帮助手册查看详情。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
资源组	可以用于执行同步任务，通常添加资源组时可以绑定多台机器。详情请参见#unique_33。
Portocol	目前仅支持FTP和SFTP协议。
Host	对应FTP主机的IP地址。
Port	如果选择的是FTP协议，则端口默认为21。如果选择的是SFTP协议，则端口默认为22。
用户名/密码	访问该FTP服务的账号密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

   提供测试连通性能力，可以判断输入的信息是否正确 。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络目前不支持数据源连通性测试，直接单击完成。

后续步骤

现在，您已经学习了如何配置FTP数据源，您可以继续学习下一个教程。在该教程中您将学习如何配置FTP插件。详情请参见#unique_87和#unique_88。

2.2.11 配置HDFS数据源

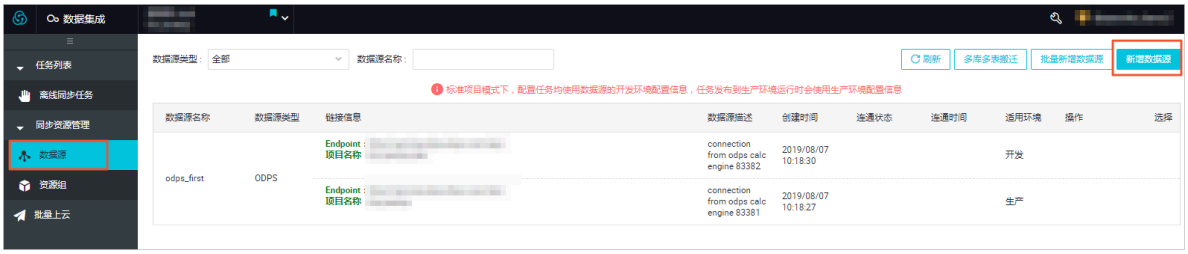
HDFS是一个分布式文件系统，它为您提供读取和写入HDFS双向通道的功能，您可以通过脚本模式配置同步任务。

说明:
标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

操作步骤

1. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。

2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为HDFS。

4. 填写HDFS数据源的各配置项。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
DefaultFS	nameNode节点地址，格式为`hdfs://ServerIP:Port`。

5. 单击测试连通性。

6. 测试连通性通过后，单击完成。

   提供测试连通性能力，可以判断输入的信息是否正确 。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络目前不支持数据源连通性测试，直接单击完成。

后续步骤

现在，您已经学习了如何配置HDFS数据源，您可以继续学习下一个教程。在该教程中，您将学习如何配置HDFS插件。详情请参见#unique_84和#unique_85。

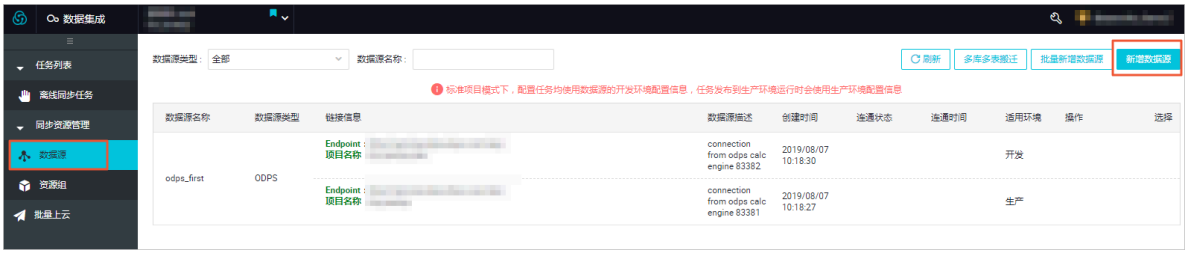
2.2.12 配置LogHub数据源

LogHub数据源作为数据中枢，为您提供读取和写入LogHub双向通道的功能，支持Reader和Writer插件。

> 说明:
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

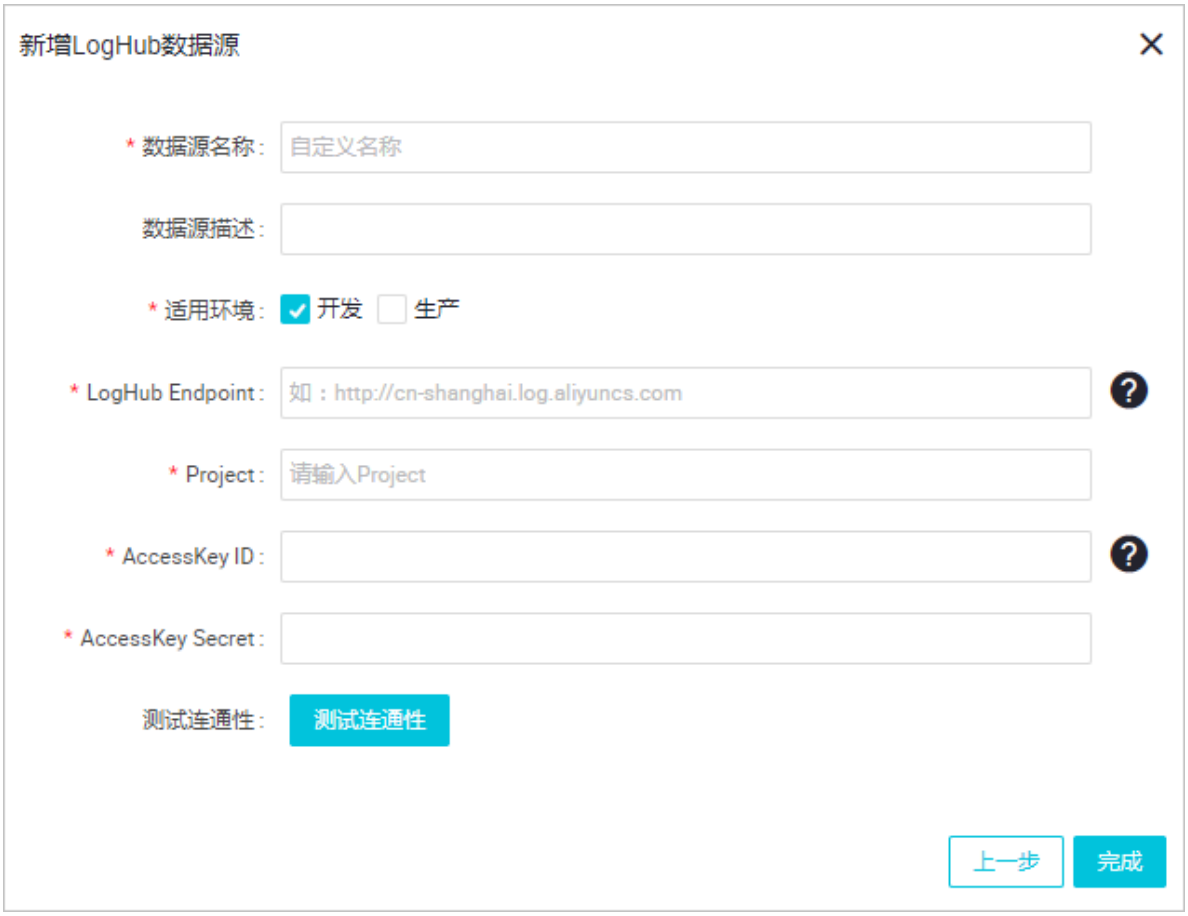
操作步骤

1. 以项目管理员身份登录DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为LogHub。

4. 填写LogHub数据源的各配置项。



配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明： 仅标准模式工作空间会显示此配置。
LogHub Endpoint	通常格式为`http://cn-shanghai.log.aliyun.com`。详情请参见服务入口。
Project	输入对应的Project。
AccessID/AceessKey	访问密钥（AccessKeyID和AccessKeySecret），相当于登录密码。

5. 单击测试连通性。

6. 测试连通性通过后，单击确定。

   提供测试连通性功能，可以判断输入的信息是否正确 。

后续步骤

现在，您已经学习了如何配置LogHub数据源，您可以继续学习下一个教程。在该教程中您将学习如何配置LogHub插件，详情请参见#unique_103和#unique_104。

2.2.13 配置MaxCompute数据源

大数据计算服务（MaxCompute，原名ODPS）为您提供了完善的数据导入方案，能够更快速地解决海量数据计算问题。MaxCompute数据源作为数据中枢，为您提供读取和写入MaxCompute双向通道的功能，支持Reader和Writer插件。

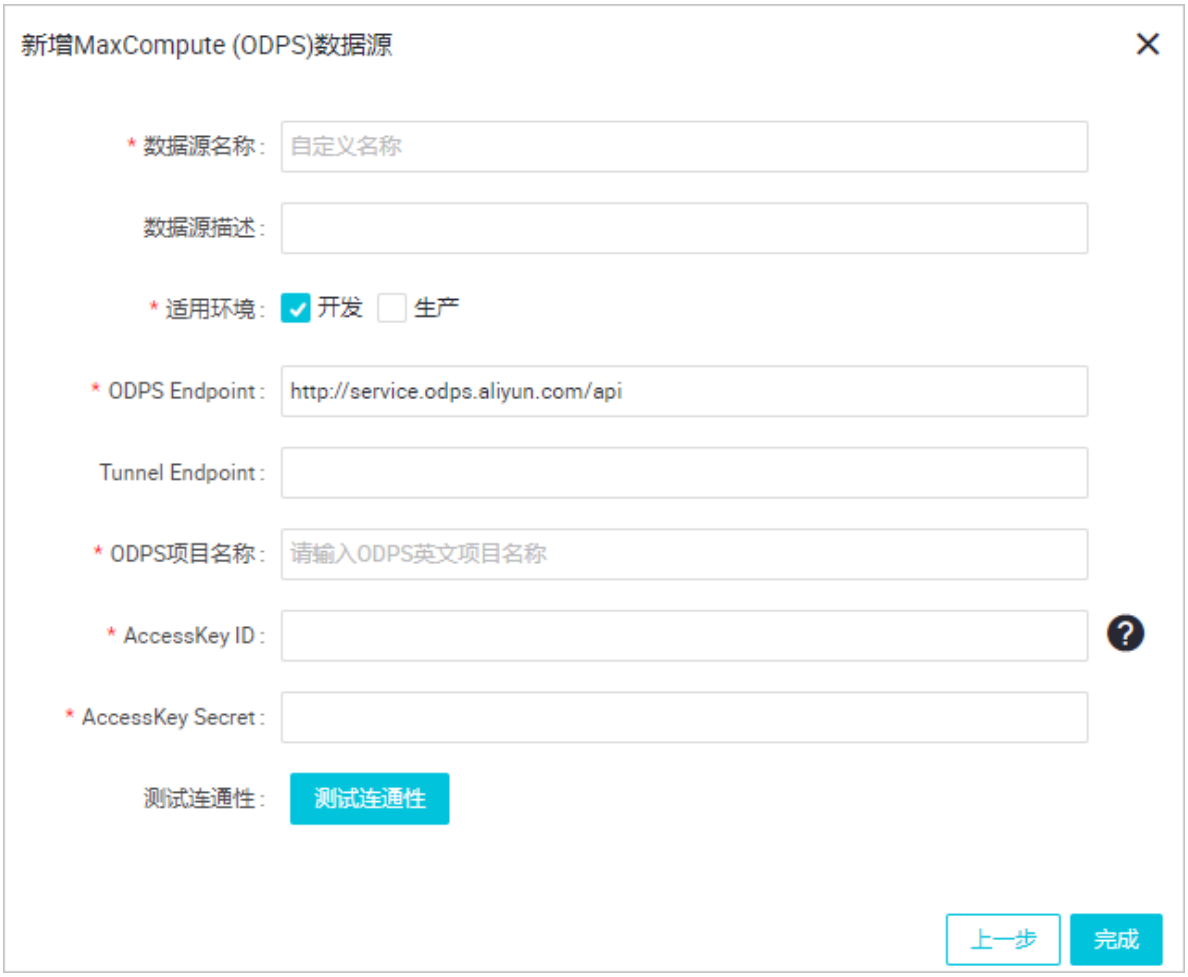
> 说明:
>
> - 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。
> - 每个项目空间系统都将生成一个默认的数据源（odps_first），对应的MaxCompute项目名称为当前项目空间对应的计算引擎MaxCompute项目名称。您可以单击右上方的用户信息，在修改AccessKey信息页面切换默认数据源的AK，但需注意以下问题：
>   - 只能从主账号AK切换到主账号AK。
>   - 切换时当前必须没有任务在运行中（数据集成或数据开发等一切和DataWorks相关的任务），您自行添加的MaxCompute数据源可以使用子账号AK。

操作步骤

1. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。

3. 在新增数据源弹出框中，选择数据源类型为MaxCompute（ODPS）。

4. 填写MaxCompute数据源的各配置项。



配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
ODPS Endpoint	默认只读，从系统配置中自动读取。
Tunnel Endpoint	MaxCompute Tunnel服务的连接地址，详情请参见#unique_125。
ODPS项目名称	MaxCompute（ODPS）项目名称。
AccessID/AceessKey	访问秘匙（AccessKeyID和AccessKeySecret），相当于登录密码。

5. 单击测试连通性。

6. 测试连通性通过后，单击完成。

   提供测试连通性能力，可以判断输入的信息是否正确 。

后续步骤

现在，您已经学习了如何配置MaxCompute数据源，您可以继续学习下一个教程。在该教程中，您将学习如何配置MaxCompute插件。详情请参见#unique_71和#unique_72。

2.2.14 配置Memcached数据源

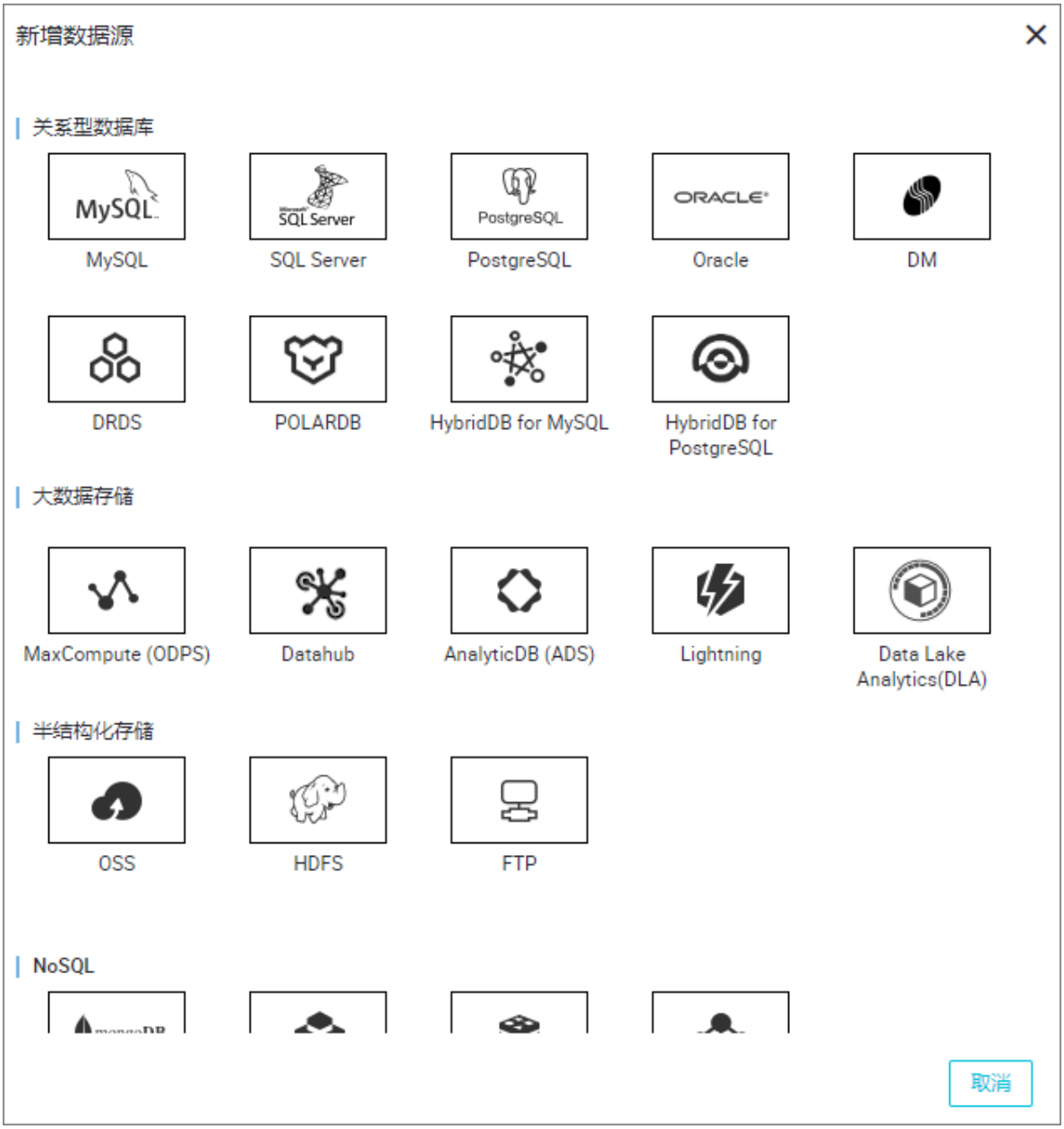
Memcache（原名OCS）数据源提供了其它数据源将数据写入Memcache的能力，目前只能通过脚本模式配置同步任务。

> 说明:
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击对应项目操作栏中的进入数据集成。

2. 单击数据源 > 新增数据源，弹出支持的数据源。



3. 在新建数据源弹出框中，选择数据源类型为Memcached。

4. 填写Memcached数据源的各配置项。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
数据源类型	当前选择的数据源类型Memcache。
Proxy Host	相应的Memcache Proxy。
Port	相应的Memcache端口，默认为11211。
用户名/密码	对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击确定。

   提供测试连通性能力，可以判断输入的各项信息是否正确 。

后续步骤

现在，您已经学习了如何配置Memcache数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置Memcache Writer插件。详情请参见#unique_93。

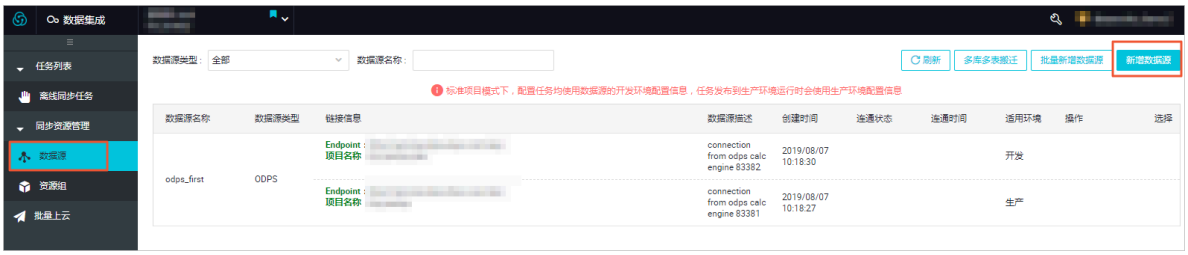
2.2.15 配置MySQL数据源

MySQL数据源为您提供读取和写入MySQL双向通道的功能，可以通过向导模式和脚本模式配置同步任务。

> **说明:**
>
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。
>
> 如果是在VPC环境下的MySQL，需要注意以下问题：
>
> - 自建的MySQL数据源
> - 不支持测试连通性，但仍支持配置同步任务，创建数据源时单击完成即可。
> - 必须使用自定义调度资源组运行对应的同步任务，请确保自定义资源组可以连通您的自建数据库，详情请参见#unique_34和#unique_35。
> - 通过RDS创建的MySQL数据源
>
> 您无需选择网络环境，系统会自动根据您填写的RDS实例信息进行判断。
>
> 目前DataWorks数据集成驱动无法直接支持MySQL 8.0版本。如果您使用MySQL 8.0，请#unique_33，详情请参见#unique_128，配合#unique_129及#unique_130，完成与MySQL数据库的连接和读写。

操作步骤

1. 以项目管理员身份登录DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为MySQL。

4. 填写MySQL数据源的各配置项。

   MySQL数据源类型包括阿里云数据库（RDS）、连接串模式（数据集成网络可直接连通）和连接串模式（数据集成网络不可直接连通）。

   以新增MySQL > 阿里云数据库（RDS）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为MySQL > 阿里云数据库（RDS）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 **说明:** 仅标准模式工作空间会显示此配置。
地区	选择相应的区域。
RDS实例ID	您可以进入RDS管控台，查看RDS的实例ID。 
RDS实例主账号ID	您可以进入RDS控制台的安全设置页面，查看相应的信息。 
数据库名	填写对应的数据库名称。
用户名/密码	数据库对应的用户名和密码。

说明:

您需要先添加RDS白名单才能连接成功，详情请参见#unique_111。

以新增MySQL > 连接串模式（数据集成网络可直接连通）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为MySQL > 连接串模式（数据集成网络可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
JDBC URL	JDBC连接信息，格式为`jdbc:mysql://ServerIP:Port/Database`。

配置	说明
用户名/密码	数据库对应的用户名和密码。

以新增MySQL > 连接串模式（数据集成网络不可直接连通）类型的数据源为例。

说明:

连接串模式（数据集成网络不可直接连通）的数据源不支持测试连通性。

配置	说明
数据源类型	当前选择的数据源类型为MySQL > 连接串模式（数据集成网络不可直接连通）。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
资源组	可以用于执行同步任务，通常添加资源组时可以绑定多台机器。详情请参见#unique_33。
JDBC URL	JDBC连接信息，格式为`jdbc:mysql://ServerIP:Port/Database`。
用户名/密码	数据库对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络下，如果您使用实例模式配置数据源，可以判断输入的信息是否正确。
- 专有网络下，如果您将VPC内部地址作为JDBC URL添加数据源，测试连通性会报告失败。
- 经典网络/专有网络下，如果您将数据源的公网地址作为JDBC URL添加数据源，可以判断输入的信息是否正确。

后续步骤

现在，您已经学习了如何配置MySQL数据源，您可以继续学习下一个教程。在该教程中您将学习如何配置MySQL插件。详情请参见#unique_44和#unique_45。

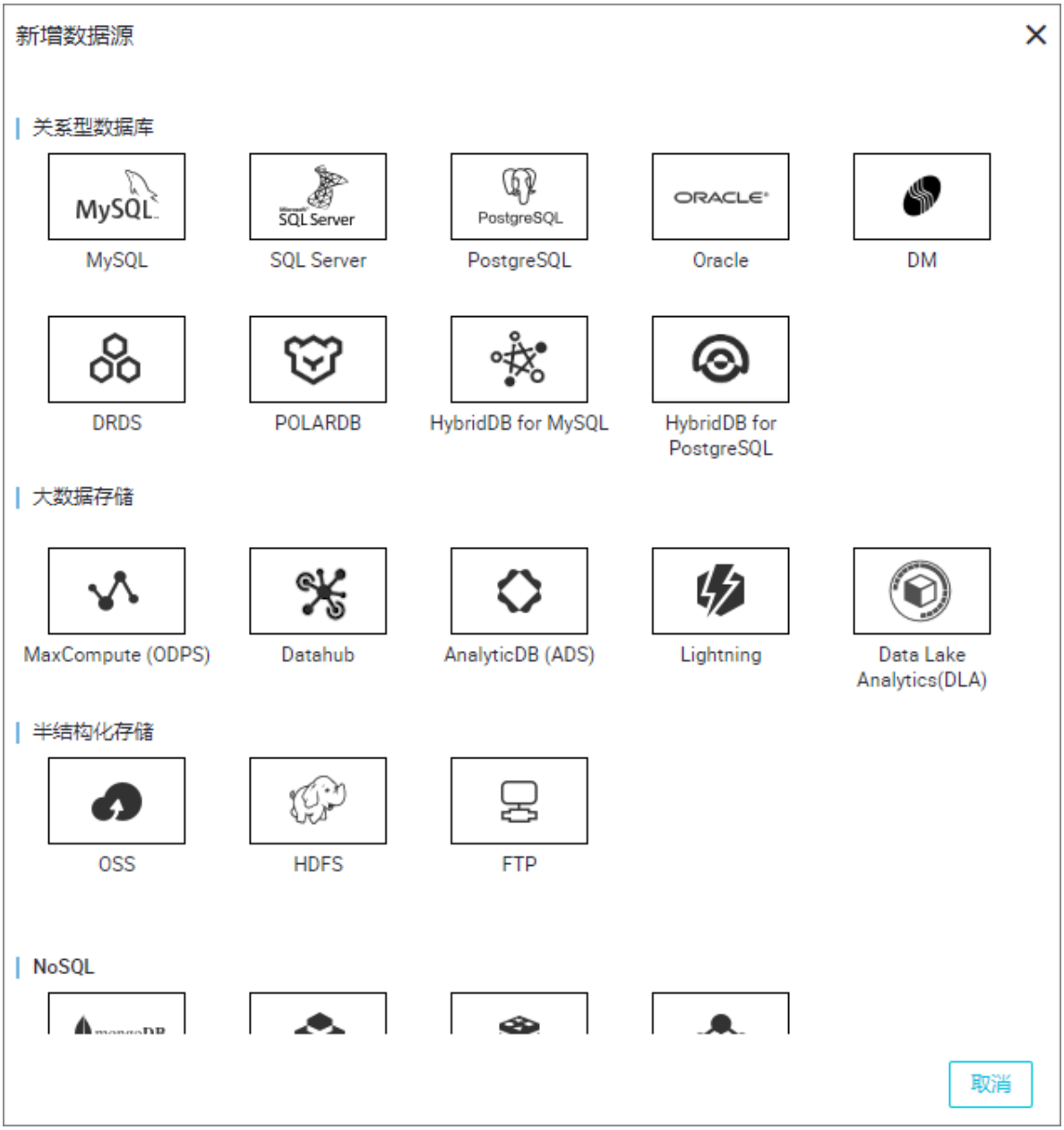
2.2.16 配置Oracle数据源

Oracle数据源提供了读取和写入Oracle双向通道的能力，您可以通过向导模式和脚本模式配置同步任务。

说明:
标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击对应项目操作栏中的进入数据集成。
2. 选择数据源 > 新增数据源，弹出支持的数据源。

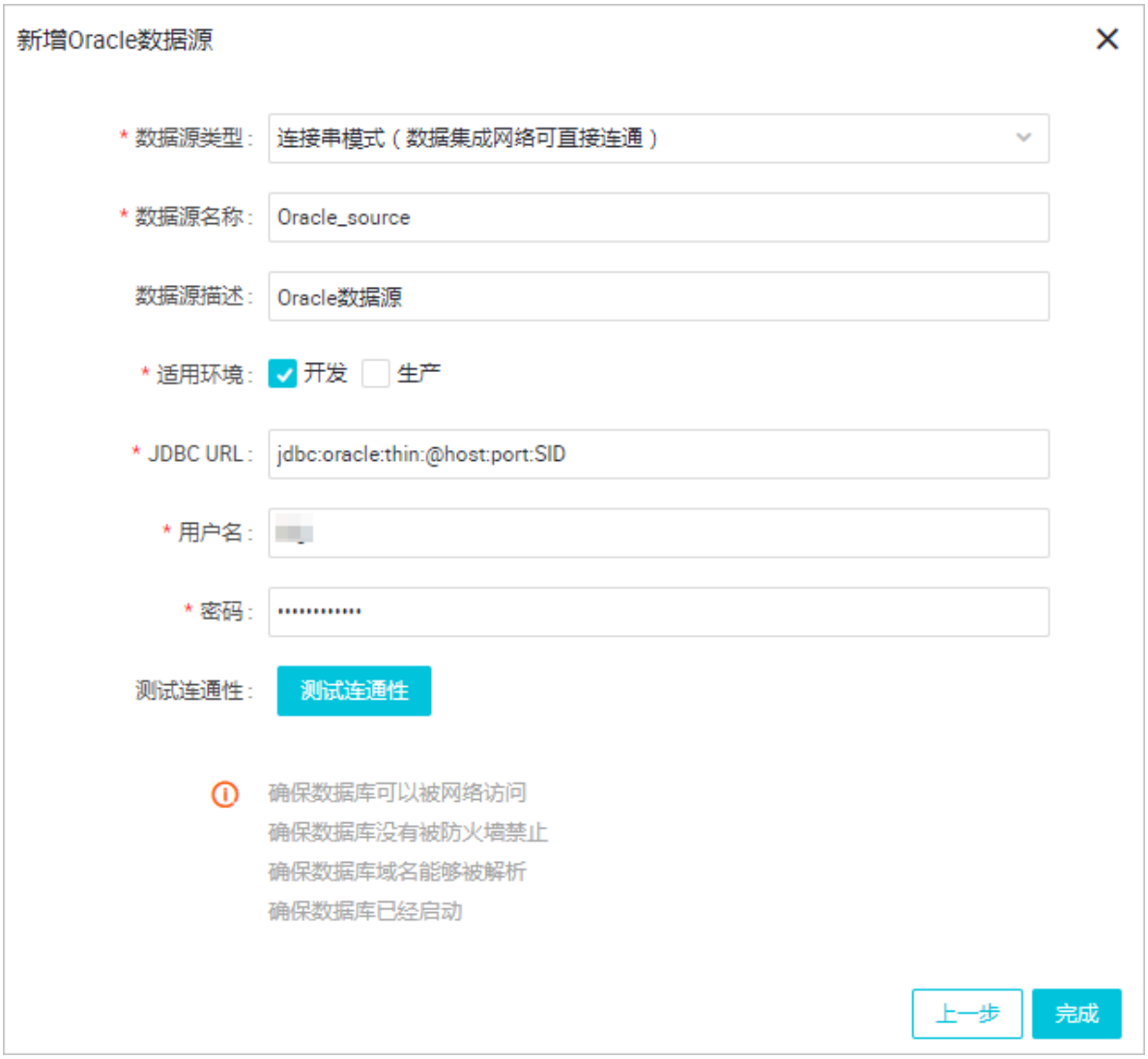


3. 在新建数据源弹出框中，选择数据源类型为Oracle。

4. 填写新增Oracle数据源对话框中的配置信息。

   Oracle数据源类型分为连接串模式（数据集成网络可直接连通）和连接串模式（数据集成网络不可直接连通），您可根据自身情况进行选择。

   以新增Oracle > 连接串模式（数据集成网络可直接连通）类型的数据源为例。



配置	说明
数据源类型	连接串模式（数据集成网络可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	根据自身需求勾选开发或生产。
JDBC URL	JDBC连接信息，格式为`jdbc:oracle:thin:@host:port:SID`或`jdbc:oracle:thin:@//host:port/service_name`。

配置	说明
用户名/密码	数据库对应的用户名和密码。

以新增Oracle > 连接串模式（数据集成网络不可直接连通）类型的数据源为例。

配置	说明
数据源类型	连接串模式（数据集成网络不可直接连通），此种类型的数据源需要使用自定义调度资源才能进行同步，可单击帮助手册进行查看。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	根据自身需求勾选开发或生产。
资源组	选择相应的资源组，您也可新增自定义资源组。
JDBC URL	JDBC连接信息，格式为`jdbc:oracle:thin:@host:port:SID`或`jdbc:oracle:thin:@//host:port/service_name`。
用户名/密码	数据库对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击确定。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的JDBC URL、用户名/密码是否正确。
- 专有网络无公网和其他本地网络无公网，数据集成不可直接连通（不支持测试连通性），需要通过JDBC连接形式配置数据源，并且通过自定义资源组进行任务同步。

后续步骤

现在，您已经学习了如何配置Oracle数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置Oracle Writer插件。详情请参见#unique_54和#unique_53。

2.2.17 配置OSS数据源

对象存储（Object Storage Service，简称OSS），是阿里云对外提供的海量、安全和高可靠的云存储服务。

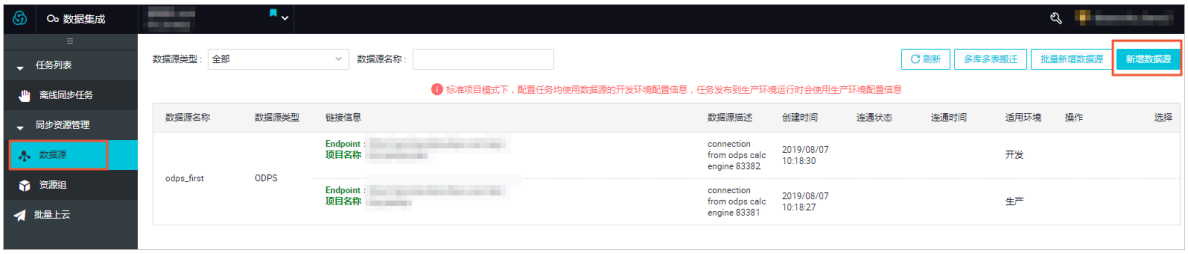
说明:

- 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源，并进行隔离，以保护您的数据安全。
- 如果您想对OSS产品有更深了解，请参见OSS产品概述。
- OSS Java SDK请参见阿里云OSS Java SDK。

操作步骤

1. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。

2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为OSS。

4. 填写OSS数据源的各配置项。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明： 仅标准模式工作空间会显示此配置。

配置	说明
Endpoint	OSS Endpoint信息，格式为http://oss.aliyuncs.com，OSS服务的Endpoint和区域有关。访问不同的区域时，需要填写不同的域名。 说明: Endpoint的正确的填写格式为http://oss.aliyuncs.com，但http://oss.aliyuncs.com在OSS前加上Bucket值，以点号的形式连接。例如http://xxx.oss.aliyuncs.com，测试连通性可以通过，但同步会报错。
Bucket	相应的OSS Bucket信息，指存储空间，是用于存储对象的容器。 您可以创建一个或多个存储空间，每个存储空间可添加一个或多个文件。 您可在数据同步任务中查找此处填写的存储空间中相应的文件，没有添加的存储空间，则不能查找其中的文件。
AccessKey ID/ AceessKey Secret	访问秘匙（AccessKeyID和AccessKeySecret），相当于登录密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

说明:
准备OSS数据时，如果数据为CSV文件，则必须为标准格式的CSV文件。例如：列内容如果在半角引号（"）内时，需要替换成两个半角引号（""），否则会造成文件被错误分割。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络目前不支持数据源连通性测试，直接单击完成。

后续步骤

现在，您已经学习了如何配置OSS数据源，您可以继续学习下一个教程。在该教程中，您将学习如何配置OSS插件。详情请参见#unique_81和#unique_82。

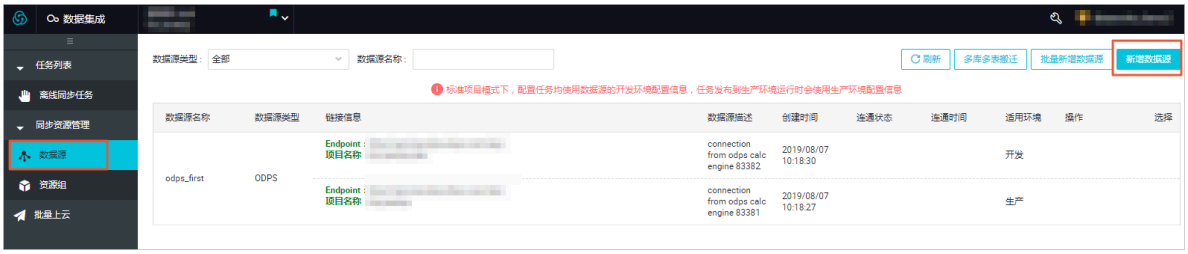
2.2.18 配置Table Store（OTS）数据源

表格存储（Table Store）是构建在阿里云飞天分布式系统之上的NoSQL数据存储服务，提供海量结构化数据的存储和实时访问。

> **说明:**
>
> - 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。
> - 如果您想对表格存储有更深入的了解，请参见#unique_135。

操作步骤

1. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为Table Store（OTS）。

4. 填写Table Store（OTS）数据源的各配置项。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
Endpoint	Table Store服务对应的Endpoint。
Table Store实例ID	Table Store服务对应的实例ID。
AccessID/AceessKey	访问秘匙（AccessKeyID和AccessKeySecret），相当于登录密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络目前不支持数据源连通性测试，直接单击完成。

后续步骤

现在，您已经学习了如何配置OTS数据源，您可以继续学习下一个教程。在该教程中，您将学习如何配置Table Store（OTS） Reader插件。详情请参见#unique_97。

2.2.19 配置PostgreSQL数据源

PostgreSQL数据源为您提供读取和写入PostgreSQL双向通道的功能，您可以通过向导模式和脚本模式配置同步任务。

> 说明:
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源，并进行隔离，以保护您的数据安全。
>
> 如果是在VPC环境下的PostgreSQL，需要注意以下问题：
>
> - 自建的PostgreSQL数据源：
> - 不支持测试连通性，但仍支持配置同步任务，创建数据源时单击完成即可。
> - 必须使用自定义调度资源组运行对应的同步任务，请确保自定义资源组可以连通您的自建数据库，详情请参见#unique_34和#unique_35。
> - 通过RDS创建的PostgreSQL数据源。
>
> 您无需选择网络环境，系统自动根据您填写的RDS实例信息进行判断。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。

3. 在新增数据源弹出框中，选择数据源类型为PostgreSQL。

4. 填写PostgreSQL数据源的各配置项。

   PostgreSQL数据源类型分为阿里云数据库（RDS）、连接串模式（数据集成网络可直接连通）和连接串模式（数据集成网络不可直接连通），您可以根据自身情况进行选择。

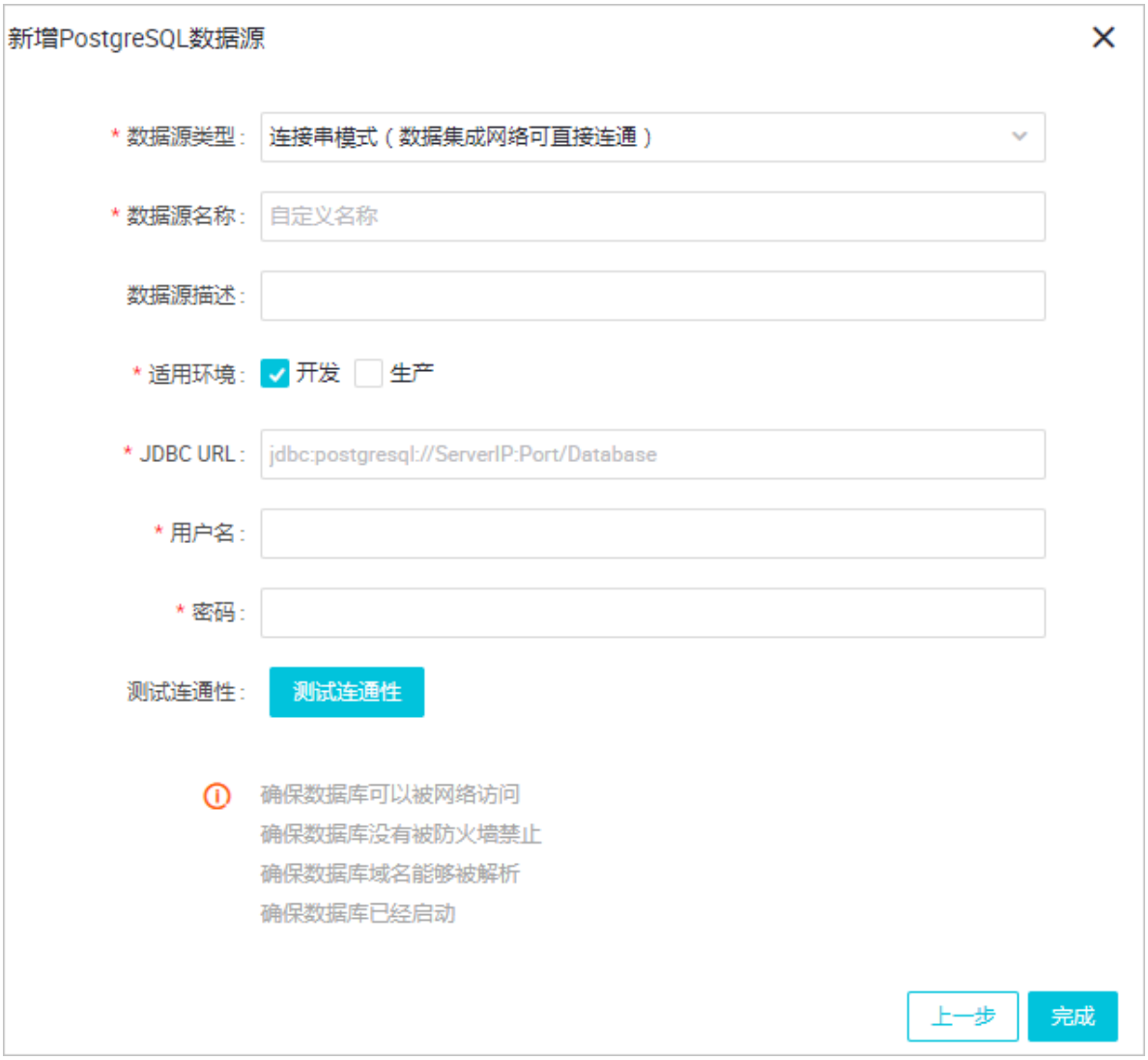
   以新增PostgreSQL > 阿里云数据库（RDS）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为PostgreSQL > 阿里云数据库（RDS）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
地区	选择相应的Region。
RDS实例ID	您可以进入RDS管控台，查看RDS的实例ID。 
RDS实例主账号ID	购买RDS实例的主账号的ID。
数据库名	填写对应的数据库名称。

配置	说明
用户名/密码	数据库对应的用户名和密码。

以新增PostgreSQL > 连接串模式（数据集成网络可直接连通）类型的数据源为例。



配置	说明
数据源类型	当前选择的数据源类型为PostgreSQL > 连接串模式（数据集成网络可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。

配置	说明
JDBC URL	JDBC连接信息，格式为`jdbc:postgresql://ServerIP:Port/Database`。
用户名/密码	数据库对应的用户名和密码。

以新增PostgreSQL > 连接串模式（数据集成网络不可直接连通）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为PostgreSQL > 连接串模式（数据集成网络不可直接连通）。 选择此类型的数据源需要使用自定义调度资源才能进行同步，您可以单击帮助手册查看详情。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
资源组	可以用于执行同步任务，通常添加资源组时可以绑定多台机器。详情请参见#unique_33。
JDBC URL	JDBC连接信息，格式为`jdbc:postgresql://ServerIP:Port/Database`。
用户名/密码	数据库对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络下，如果您使用实例模式配置数据源，可以判断输入的信息是否正确。
- 专有网络下，如果您将VPC内部地址作为JDBC URL添加数据源，测试连通性会报告失败。
- 经典网络/专有网络下，如果您将数据源的公网地址作为JDBC URL添加数据源，可以判断输入的信息是否正确。

后续步骤

现在，您已经学习了如何配置PostgreSQL数据源，您可以继续学习下一个教程。在该教程中您将学习如何配置PostgreSQL插件。详情请参见#unique_50和#unique_51。

2.2.20 配置Redis数据源

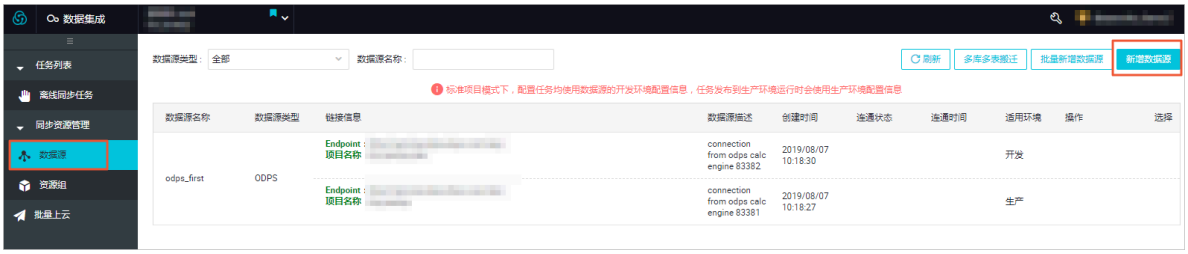
Redis数据源为您提供读取和写入Redis双向通道的功能，您可以通过脚本模式配置同步任务。

Redis是文档型的NoSQL数据库，提供持久化的内存数据库服务，基于高可靠双机热备架构及可无缝扩展的集群架构，满足高读写性能场景，以及容量需要弹性变化的业务需求。

说明:
标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源，并进行隔离，以保护您的数据安全。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为Redis。
4. 填写Redis数据源的各配置项。

   Redis数据源类型包括实例模式（阿里云数据源）和连接串模式（数据集成网络可直接连通），您可以根据自身需求进行选择。

   - 以新增Redis > 实例模式（阿里云数据源）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为Redis > 实例模式（阿里云数据源）。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
地区	填写购买Redis时所选择的区域。
Redis实例ID	您可以进入Redis管控台，查看Redis实例ID。
Redis访问密码	Redis Server的访问密码，如果没有则不填。

- 以新增Redis > 连接串模式（数据集成网络可直接连通）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为FTP > 连接串模式（数据集成网络不可直接连通）。 选择此类型的数据源需要使用自定义调度资源才能进行同步，您可以单击帮助手册查看详情。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
服务器地址	格式为`host:port`。
添加访问地址	添加访问地址，格式为`host:port`。
Redis访问密码	Redis的服务访问密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

后续步骤

现在，您已经学习了如何配置Redis数据源，您可以继续学习下一个教程。在该教程中您将学习如何配置Redis Writer插件。详情请参见#unique_95。

2.2.21 配置HybridDB for MySQL数据源

HybridDB for MySQL数据源为您提供读取和写入HybridDB for MySQL的双向能力，本文将为您介绍如何配置HybridDB for MySQL数据源。

说明:
标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

您可以通过向#unique_27和#unique_36配置同步任务。

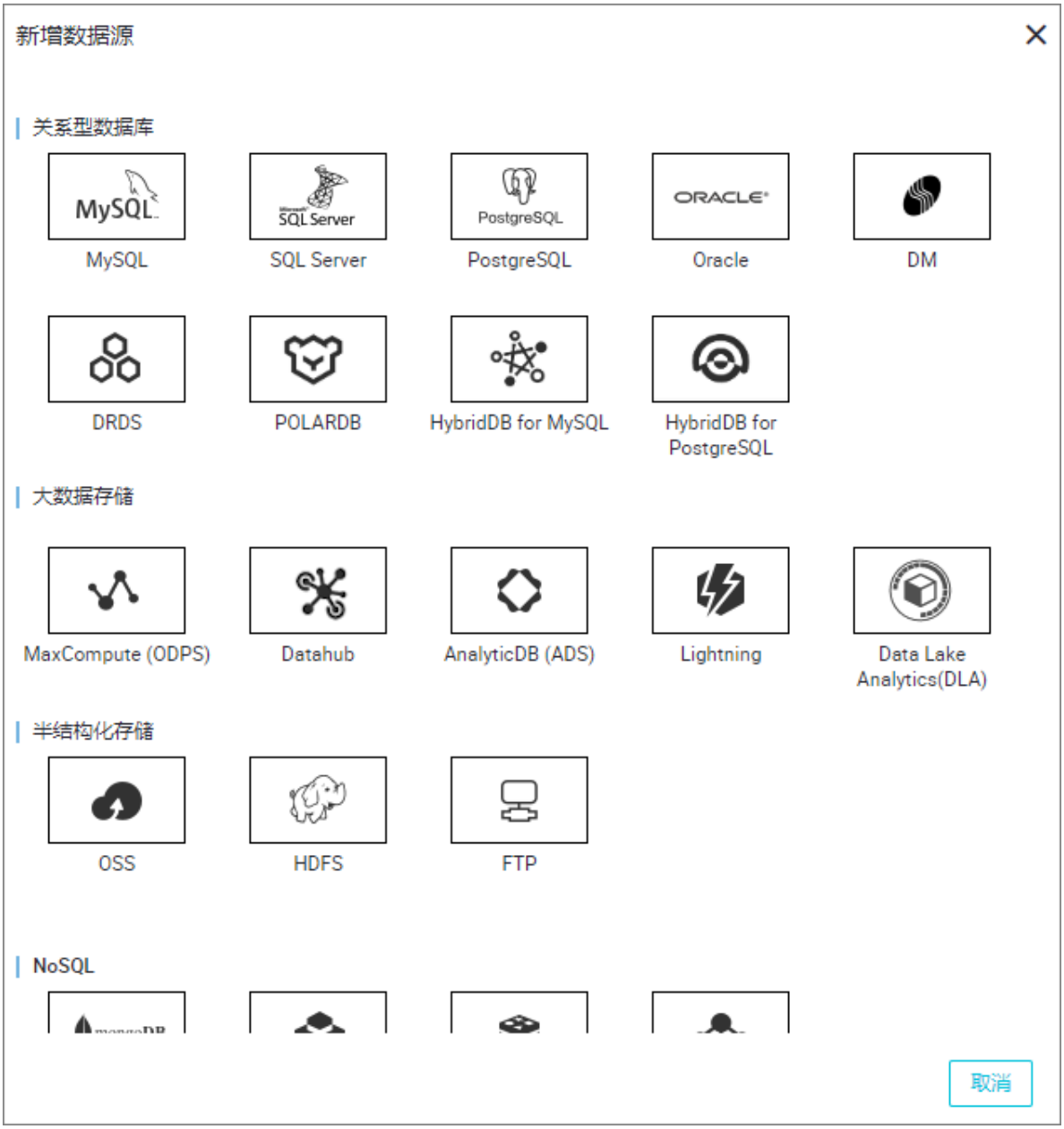
说明:
如果是在VPC环境下的HybridDB for MySQL，需要注意以下问题。

- 自建的MySQL数据源
  - 不支持测试连通性，但仍支持配置同步任务，创建数据源时单击确认即可。
  - 必须使用自定义调度资源组运行对应的同步任务，请确保自定义资源组可以连通您的自建数据库，详情请参见#unique_34和#unique_35。

- 对于通过实例ID创建的HybridDB for MySQL数据源，您无需选择网络环境，系统自动根据您填写的HybridDB for MySQL实例信息进行判断。

操作步骤

1. 以项目管理员身份进入DataWorks管理控制台，单击对应项目操作栏中的进入数据集成。
2. 单击数据源 > 新增数据源，弹出支持的数据源。



3. 在新增数据源弹出框中，选择数据源类型为HybridDB for MySQL。

4. 填写HybridDB for MySQL数据源的各配置项。

配置	说明
数据源类型	当前选择的数据源类型为阿里云数据源（HybridDB）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对新建的数据源进行简单描述。
适用环境	分为开发环境和生产环境。
实例ID	您可进入HybridDB for MySQL管控台，查看相关的实例ID。

配置	说明
主账号ID	您可在HybridDB for MySQL管控台安全设置中查看相应的信息。 
用户名/密码	数据库对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击确定。

> 说明:
> 您需要先添加白名单才能连接成功，详情请参看#unique_111文档。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的用户名/密码、实例ID/JDBC URL是否正确。
- 专有网络下，如果您使用实例模式配置数据源，可以判断输入的实例ID、主账号ID、用户名/密码是否正确。
- 专有网络下，如果您将VPC内部地址作为JDBC URL添加数据源，测试连通性会报告失败。
- 经典网络/专有网络下，如果您将数据源的公网地址作为JDBC URL添加数据源，可以判断输入的JDBC URL、用户名/密码是否正确。

后续步骤

现在，您已经学习了如何配置HybridDB for MySQL数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置HybridDB for MySQL插件，详情请参见#unique_63和#unique_139。

2.2.22 配置AnalyticDB for PostgreSQL数据源

AnalyticDB for PostgreSQL数据源为您提供读取和写入AnalyticDB for PostgreSQL的双向功能，本文将为您介绍如何配置AnalyticDB for PostgreSQL数据源。

> 说明:
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

您可以通过#unique_27和#unique_36配置同步任务。

> 说明:
> 如果是在VPC环境下的AnalyticDB for PostgreSQL，需要注意以下问题。
>
> · 自建的PostgreSQL数据源
>
> - 不支持测试连通性，但仍支持配置同步任务，创建数据源时单击完成即可。
> - 必须使用自定义调度资源组运行对应的同步任务，请确保自定义资源组可以连通您的自建数据库，详情请参见#unique_34和#unique_35。
>
> · 通过实例ID创建的AnalyticDB for PostgreSQL数据源
>
> 您无需选择网络环境，系统自动根据您填写的RDS实例信息进行判断。

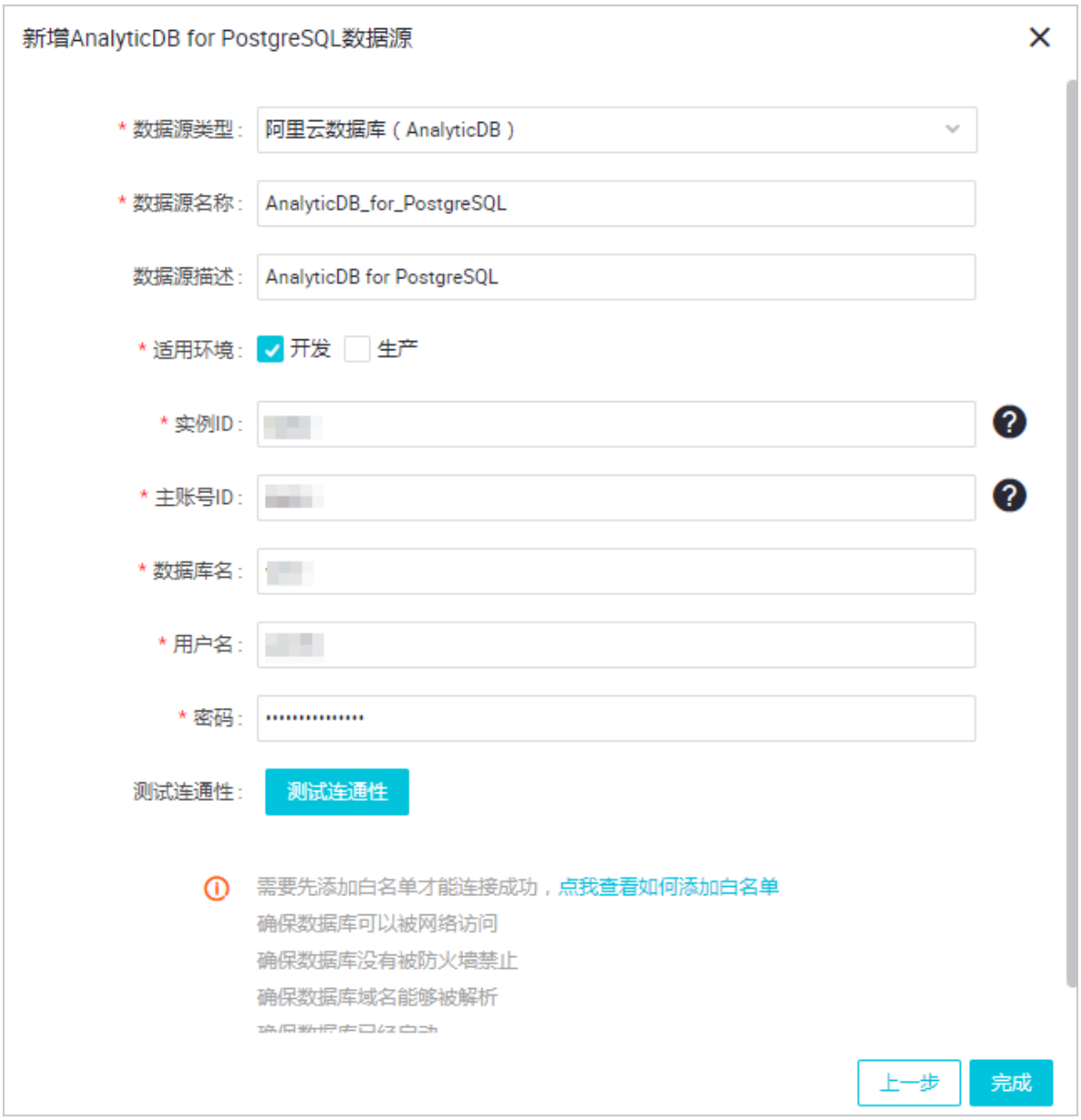
操作步骤

1. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。

3. 在新增数据源弹出框中，选择数据源类型为AnalyticDB for PostgreSQL。

4. 填写AnalyticDB for PostgreSQL数据源的各配置项。

   以新增AnalyticDB for PostgreSQL > 阿里云数据库（AnalyticDB）类型的数据源为例。



配置	说明
数据源类型	当前选择的数据源类型为AnalyticDB for PostgreSQL > 阿里云数据库（AnalyticDB）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
RDS实例ID	您可以进入AnalyticDB for PostgreSQL的控制台，查看相应的实例ID。
主账号ID	您可以进入AnalyticDB for PostgreSQL控制台的安全设置页面，查看相应的信息。 

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

说明:
您需要先添加白名单才能连接成功，详情请参见#unique_111。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络下，如果您使用实例模式配置数据源，可以判断输入的信息是否正确。
- 专有网络下，如果您将VPC内部地址作为JDBC URL添加数据源，测试连通性会报告失败。
- 经典网络/专有网络下，如果您将数据源的公网地址作为JDBC URL添加数据源，可以判断输入的信息是否正确。

后续步骤

现在，您已经学习了如何配置AnalyticDB for PostgreSQL数据源，您可以继续学习下一个教程。在该教程中，您将学习如何配置AnalyticDB for PostgreSQL插件。详情请参见#unique_66和#unique_141。

2.2.23 配置POLARDB数据源

POLARDB关系型数据库数据源提供了读取和写入POLARDB双向通道的能力，本文将为您介绍如何配置POLARDB数据源。

> 说明:
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

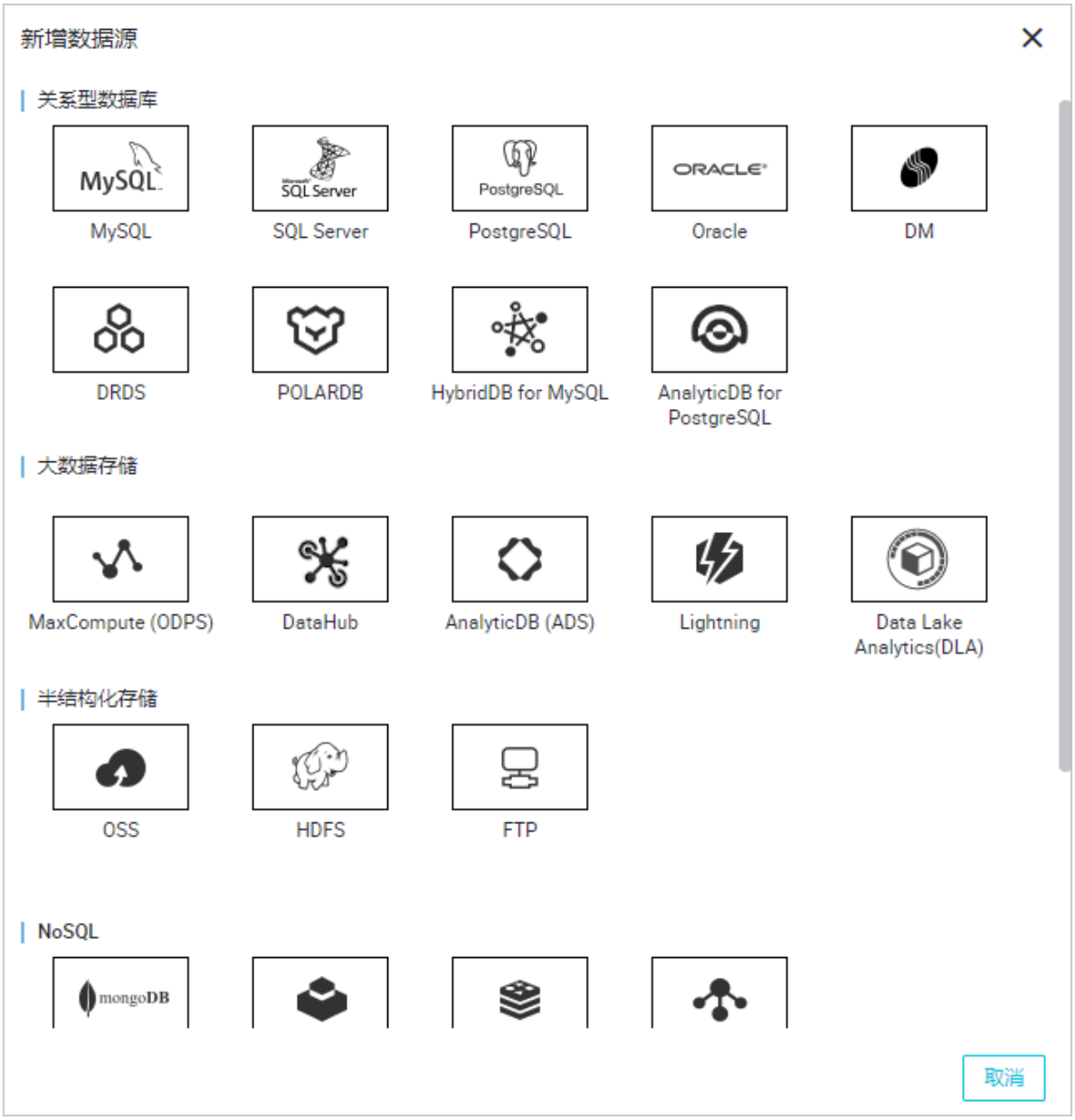
您可以通过向#unique_27和#unique_36配置同步任务。

> 说明:
> 当前POLARDB数据源不支持自定义资源组，请使用默认资源组。如果您需要使用自定义资源组，请选择添加无公网IP的MySQL数据源。如果您的数据源是在VPC环境下的POLARDB，需要注意以下问题。
>
> · 自建的POLARDB数据源
>
> - 不支持测试连通性，但仍支持配置同步任务，创建数据源时单击确认即可。
> - 必须使用自定义调度资源组运行对应的同步任务，请确保自定义资源组可以连通您的自建数据库，详情请参见#unique_34和#unique_35。
>
> · 通过实例ID创建的POLARDB数据源
>
> 您无需选择网络环境，系统自动根据您填写的POLARDB实例信息进行判断。

操作步骤

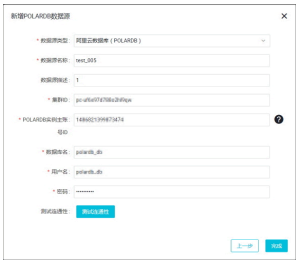
1. 以项目管理员身份进入DataWorks管理控制台，单击对应项目操作栏中的进入数据集成。

2. 单击数据源 > 新增数据源，弹出支持的数据源。



3. 在新建数据源弹出框中，选择数据源类型为阿里云数据库（POLARDB）。

4. 填写POLARDBL数据源的各配置项。



配置	说明
数据源类型	当前选择的数据源类型为阿里云数据源（POLARDB）。

配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
集群ID	可以在POLARDB管控台找到相应的集群ID内容。
POLARDB实例主账号ID	您可在POLARDB管控台安全设置中查看相应的信息。 
数据库名	POLARDB中创建的数据库名。
用户名/密码	数据库对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击确定。

> 说明:
> 您需要先添加白名单才能连接成功，详情请参见#unique_111。

测试连通性说明

- 经典网络下，能够提供测试连通性能力。
- 专有网络以添加实例ID形式能够添加成功，提供相关反向代理功能。

后续步骤

现在，您已经学习了如何配置POLARDB数据源，您可以继续学习下一个教程。在该教程中您将学习如何通过配置POLARDB插件，详情请参见#unique_142和#unique_143。

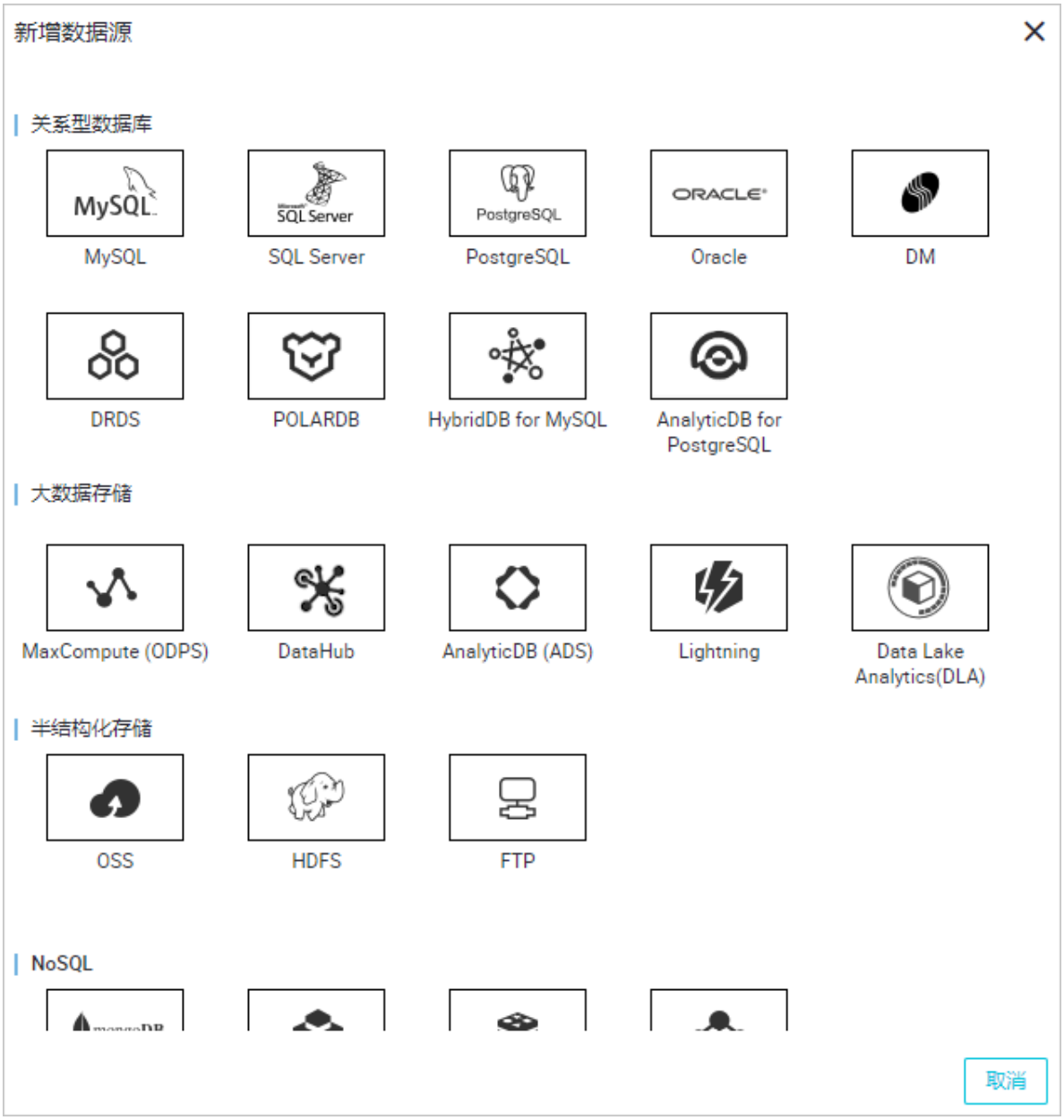
2.2.24 配置Lightning数据源

MaxCompute Lightning是MaxCompute产品的交互式查询服务，支持以PostgreSQL协议及语法连接访问Maxcompute项目，让您使用熟悉的工具以标准SQL查询分析MaxCompute项目中的数据，快速获取查询结果。

操作步骤

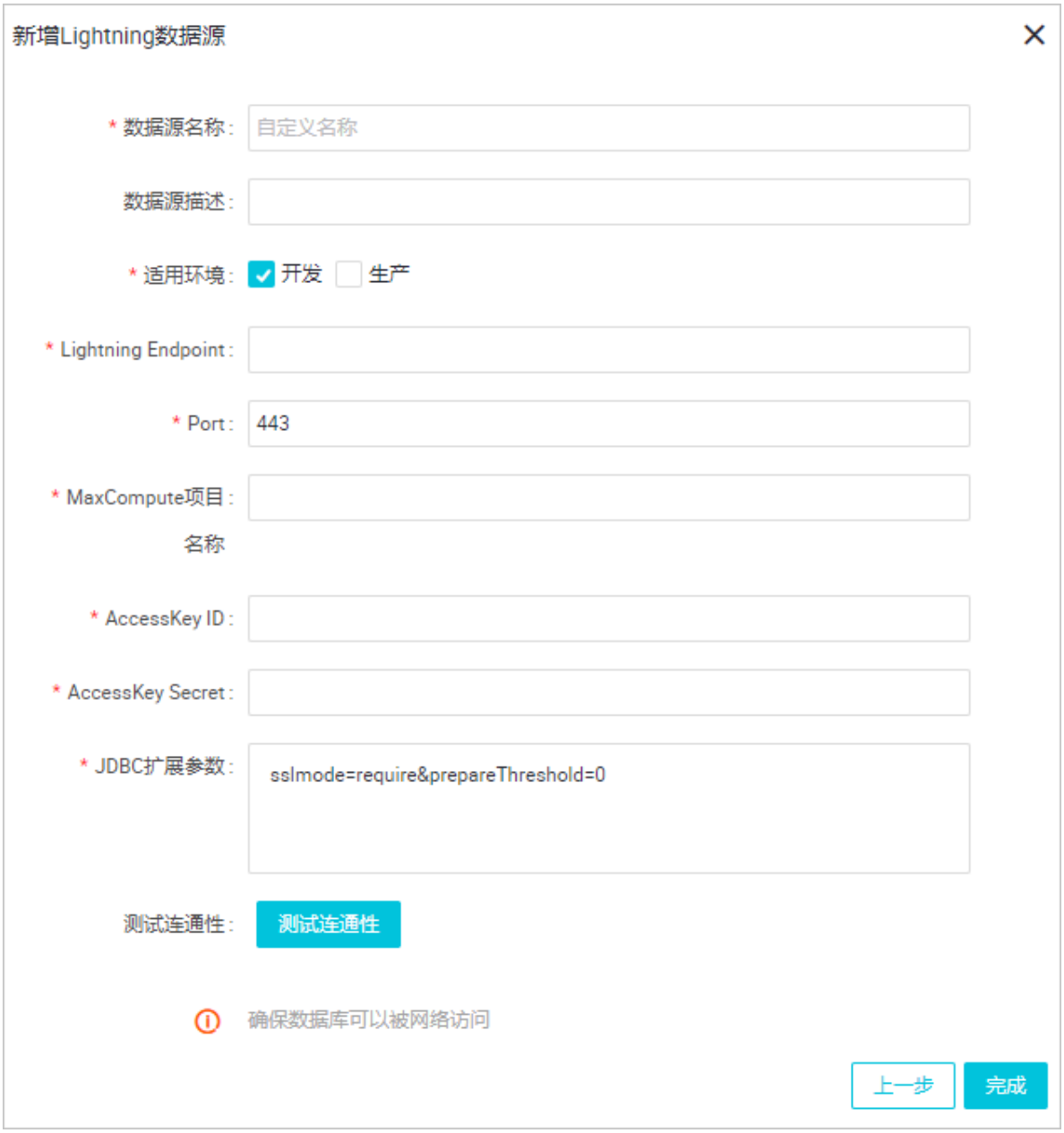
1. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。

2. 单击数据源 > 新增数据源，弹出支持的数据源。



3. 在新增数据源弹出框中，选择数据源类型为Lightning。

4. 填写Lightning数据源的各配置项。



配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
Lightning Endpoint	Lightning的连接信息，详情请参见#unique_145。

配置	说明
Port	默认值为443。
MaxCompute项目名称	填写MaxCompute的项目名称。
AccessKey ID/AccessKey Secret	访问秘匙（AccessKey ID和AccessKey Secret），相当于登录密码。
JDBC扩展参数	JDBC扩展参数中的`sslmode=require&prepareThreshold=0`是默认且不可删除的，否则会无法连接。详情请参见#unique_146。

5. 单击测试连通性。
6. 测试连通性通过后，单击确定。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络目前不支持数据源连通性测试，直接单击完成。

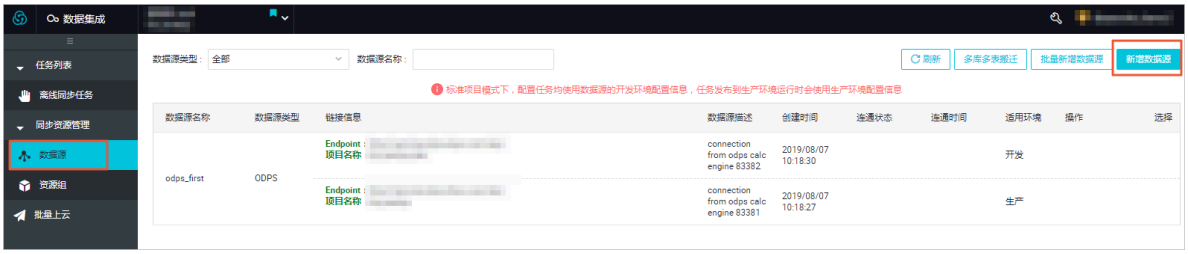
2.2.25 配置AnalyticDB for MySQL数据源

本文将为您介绍如何配置AnalyticDB for MySQL数据源。

> 说明:
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

操作步骤

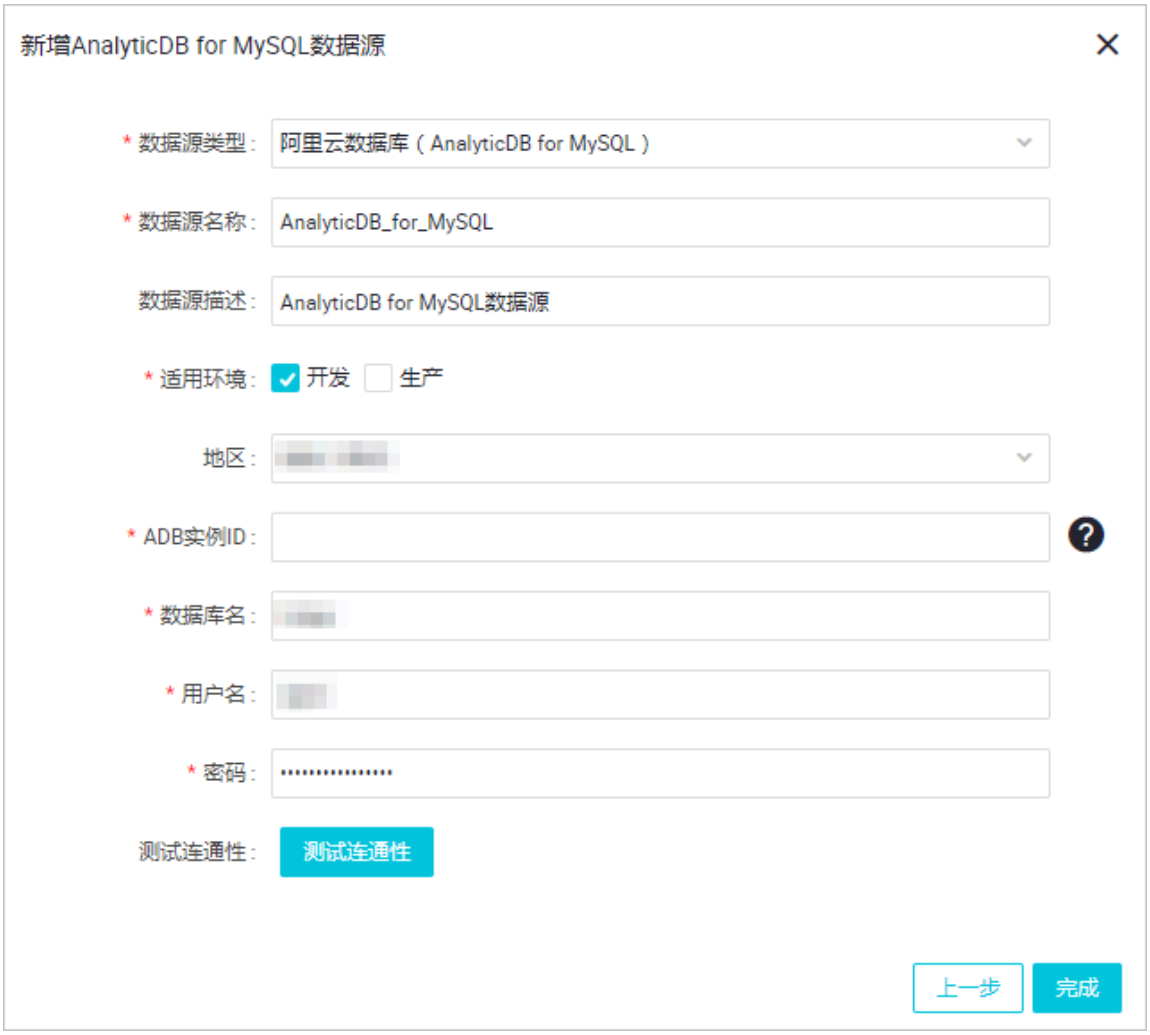
1. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为AnalyticDB for MySQL。

4. 填写AnalyticDB for MySQL数据源的各配置项。

   · 以新增AnalyticDB for MySQL > 阿里云数据库（AnalyticDB for MySQL）类型的数据源为例。



配置	说明
数据源类型	当前选择的数据源类型为AnalyticDB for MySQL > 阿里云数据库（AnalyticDB for MySQL）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
地区	选择数据源所在地区。
ADS实例ID	您可以进入RDS控制台，查看RDS实例ID。

配置	说明
数据库名	您可以新建数据库，设置相应的数据名、用户名和密码。
用户名/密码	数据库对应的用户名和密码。

- 以新增AnalyticDB for MySQL > 连接串模式（数据集成网络可直接连通）类型的数据源为例。

配置	说明
数据源类型	当前选择的数据源类型为AnalyticDB for MySQL > 连接串模式（数据集成网络可直接连通）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 **说明:** 仅标准模式工作空间会显示此配置。
JDBC URL	JDBC连接信息，格式为`jdbc:mysql://ServerIP:Port/Database`。

配置	说明
用户名/密码	数据库对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。
- 专有网络下，如果您使用实例模式配置数据源，可以判断输入的信息是否正确。

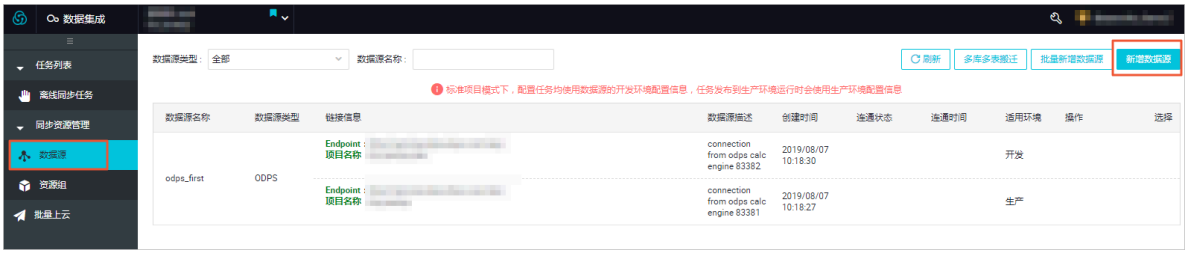
2.2.26 配置Data Lake Analytics（DLA）数据源

本文将为您介绍如何配置Data Lake Analytics（DLA）数据源。

> 说明:
> 标准模式的工作空间支持数据源隔离功能，您可以分别添加开发环境和生产环境的数据源并进行隔离，以保护您的数据安全。

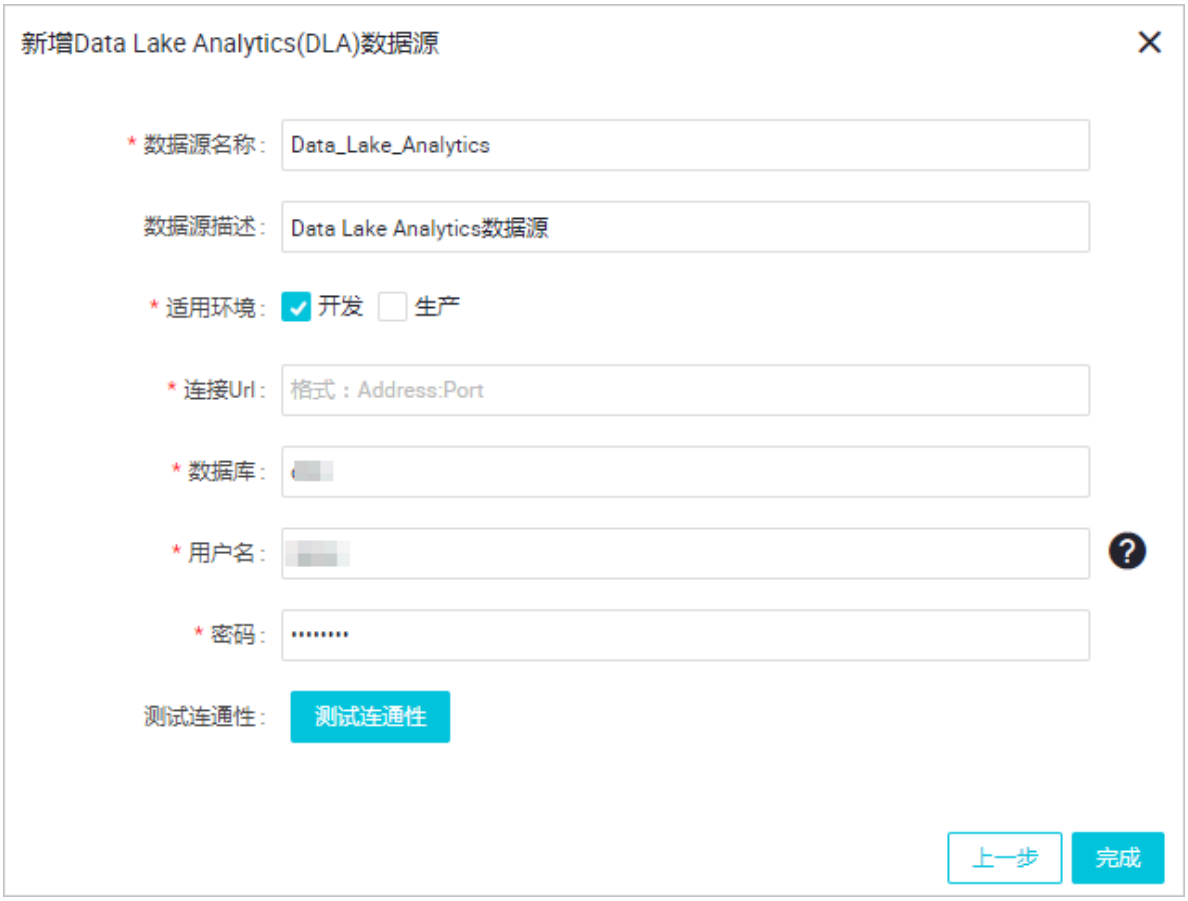
操作步骤

1. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。
2. 选择同步资源管理 > 数据源，单击新增数据源。



3. 在新增数据源弹出框中，选择数据源类型为Data Lake Analytics（DLA）。

4. 填写Data Lake Analytics（DLA）数据源的各配置项。



配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
连接Url	格式为`Address:Port`。
数据库	填写对应的数据库名称。
用户名/密码	数据库对应的用户名和密码。

5. 单击测试连通性。
6. 测试连通性通过后，单击完成。

测试连通性说明

- 经典网络下，能够提供测试连通性能力，可以判断输入的信息是否正确。

- 专有网络VPC目前不支持数据源连通性测试，直接单击完成。

  如果是专有网络VPC，需要使用独享资源，详情请参见独享资源模式和Data Lake Analytics节点。

2.3 作业配置

2.3.1 配置Reader插件

2.3.1.1 脚本模式配置

本文将为您介绍如何通过数据集成的脚本模式进行任务配置。

任务配置的操作步骤如下所示：

1. 新建数据源。
2. 新建数据同步节点。
3. 导入模板。
4. 配置同步任务的读取端。
5. 配置同步任务的写入端。
6. 配置字段的映射关系。
7. 配置作业速率上限、脏数据检查规则等信息。
8. 配置调度属性。

说明：

下文将为您介绍操作步骤的具体实现，以下每个步骤都会跳转到对应的指导文档中，请在完成当前步骤后，单击链接回到本文，继续下一步操作。

新建数据源

同步任务支持多种同构、异构数据源间的数据传输。首先，将需要同步的数据源在数据集成中完成注册。注册完成后，在数据集成配置同步任务时，可以直接选择数据源。数据集成支持同步的数据源类型请参见#unique_153。

确认需要同步的数据源已经被数据集成支持后，可以开始在数据集成中注册数据源。详细的数据源注册步骤请参见配置数据源信息。

说明：

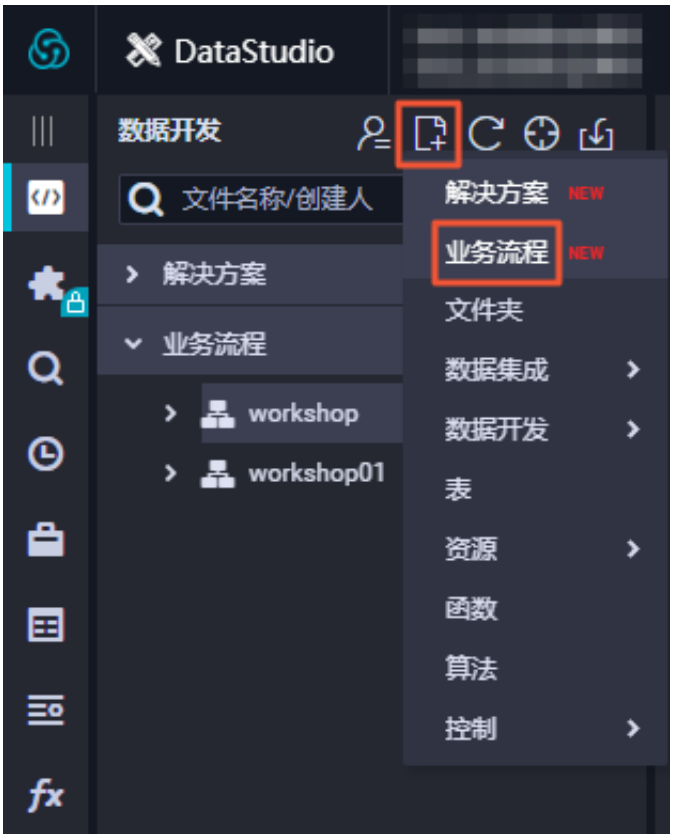
- 有部分数据源数据集成不支持测试连通性，数据源测试连通性的支持详情请参见#unique_154。

- 很多时候，数据源都是创建在本地，没有公网IP或网络无法直达。在这种情况下，配置数据源的时候测试连通性会直接失败，数据集成支持#unique_33来解决这种网络不可达的情况。但在新建同步任务的时候只能选择脚本模式（因为网络不可直达，在向导模式中就无法获取表结构等信息）。

新建数据同步节点

说明:
本文主要为您介绍向导模式下的同步任务配置，在数据集成中新建同步任务时请选择脚本模式。

1. 以开发者身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据开发。
2. 进入DataStudio（数据开发）页面，选择新建 > 业务流程。



3. 在新建业务流程对话框中，填写业务流程名称和描述，单击新建。

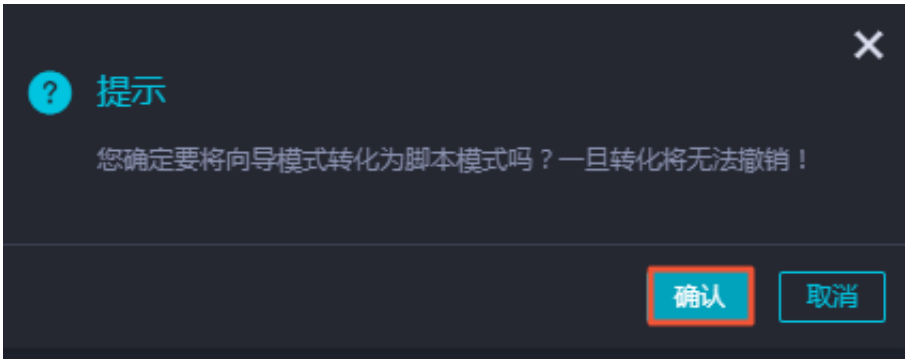
4. 展开业务流程，右键单击数据集成，选择新建数据集成节点 > 数据同步，输入节点名称。

5. 单击提交。

导入模板

1. 成功创建数据同步节点后，单击工具栏中的转换脚本。

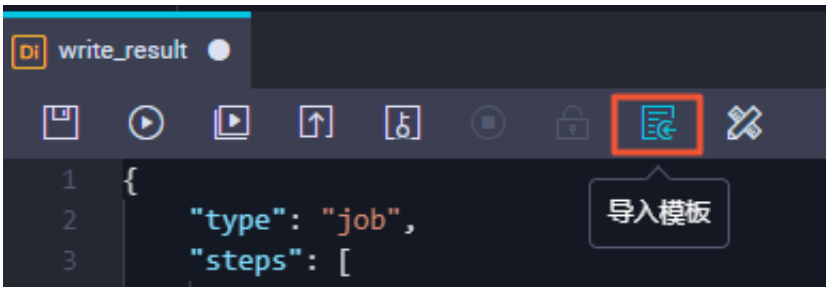
2. 单击提示对话框中的确认，即可进入脚本模式进行开发。



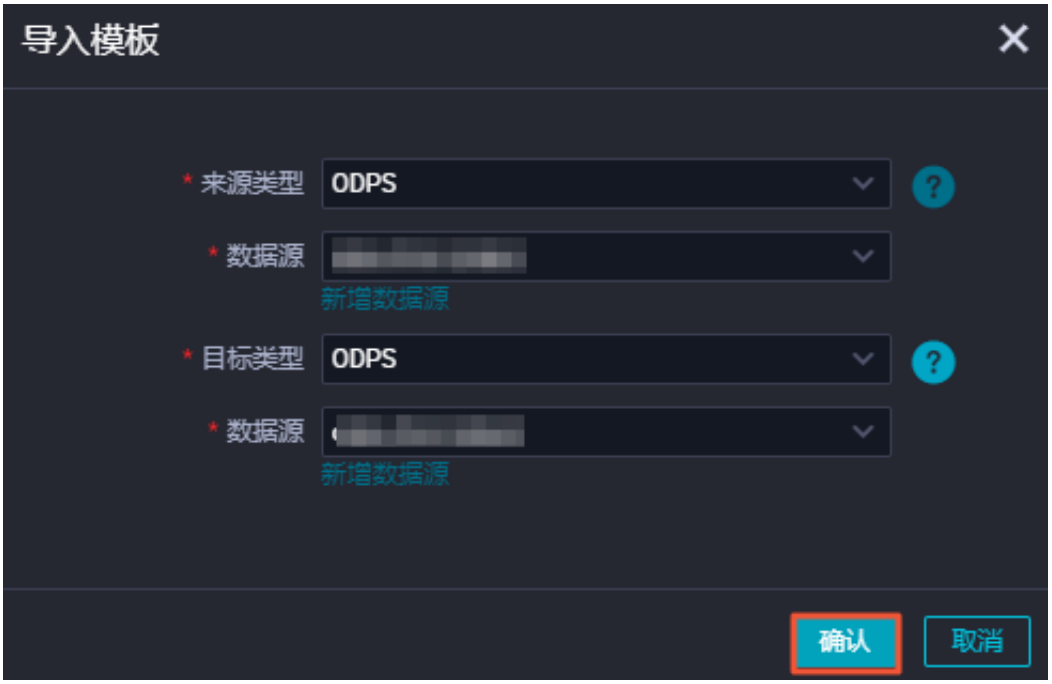
说明:

脚本模式支持更多功能，例如网络不可达情况下的同步任务编辑。

3. 单击工具栏中的导入模板。



4. 在导入模板对话框中，选择来源类型、数据源、目标类型及数据源。



5. 单击确认。

配置同步任务的读取端

新建同步任务完成后，通过导入模板已生成了基本的读取端配置。此时您可以继续手动配置数据同步任务的读取端数据源，以及需要同步的表信息等。

```
{"type": "job",
    "version": "2.0",
    "steps": [   //上述配置为整个同步任务头端代码，可以不进行修改。
        {
            "stepType": "mysql",
            "parameter": {
                "datasource": "MySQL",
                "column": [
                    "id",
                    "value",
                    "table"
                ],
                "socketTimeout": 3600000,
                "connection": [
                    {
                        "datasource": "MySQL",
                        "table": [
                            "`case`"
                        ]
                    }
                ],
                "where": "",
                "splitPk": "",
                "encoding": "UTF-8"
            },
            "name": "Reader",
            "category": "reader"     //说明分类为reader读取端。
        },    //以上配置为读取端配置。
```

配置项说明如下：

- type：指定本次提交的同步任务，仅支持Job参数，所以您只能填写为Job。
- version：目前所有Job支持的版本号为1.0或2.0。

说明：

- 选择读取端的数据源时，请参见配置Reader中的脚本开发介绍。
- 很多任务在配置读取端数据源时，需要进行数据增量同步。此时可以结合DataWorks提供的#unique_39来获取相对日期，以完成获取增量数据的需求。

配置同步任务的写入端

配置完成读取端数据源信息后，可以继续手动配置数据同步任务的写入端数据源，以及需要同步的表信息等。

```
{
  "stepType": "odps",
  "parameter": {
      "partition": "",
```

```
        "truncate": true,
        "compress": false,
        "datasource": "odps_first",
        "column": [
            "*"
         ],
         "emptyAsNull": false,
         "table": ""
       },
       "name": "Writer",
       "category": "writer"    //说明分类为writer写入端。
    }
 ],     //以上配置为读取端配置。
```

说明:

- 选择写入端的数据源时，请参见配置Writer。
- 很多任务在写入时，需要选择写入模式。例如覆盖写入还是追加写入，针对不同的数据源，有不同的写入模式。

配置字段的映射关系

脚本模式仅支持同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。

说明:

请注意列与列之间映射的字段类型是否数据兼容。

配置通道控制

当上述步骤都配置完成后，则需进行效率配置。setting域描述的是Job配置参数中除源端、目的端外，有关Job全局信息的配置参数。您可以在setting域中进行效率配置，主要包括同步并发数设置、同步速率设置、同步脏数据设置和同步资源组设置等信息。

```
"setting": {
        "errorLimit": {
            "record": "1024"    //脏数据条目设置。
        },
        "speed": {
            "throttle": false,    //是否进行限速。
            "concurrent": 1,  //同步并发数设置。
        }
    },
```

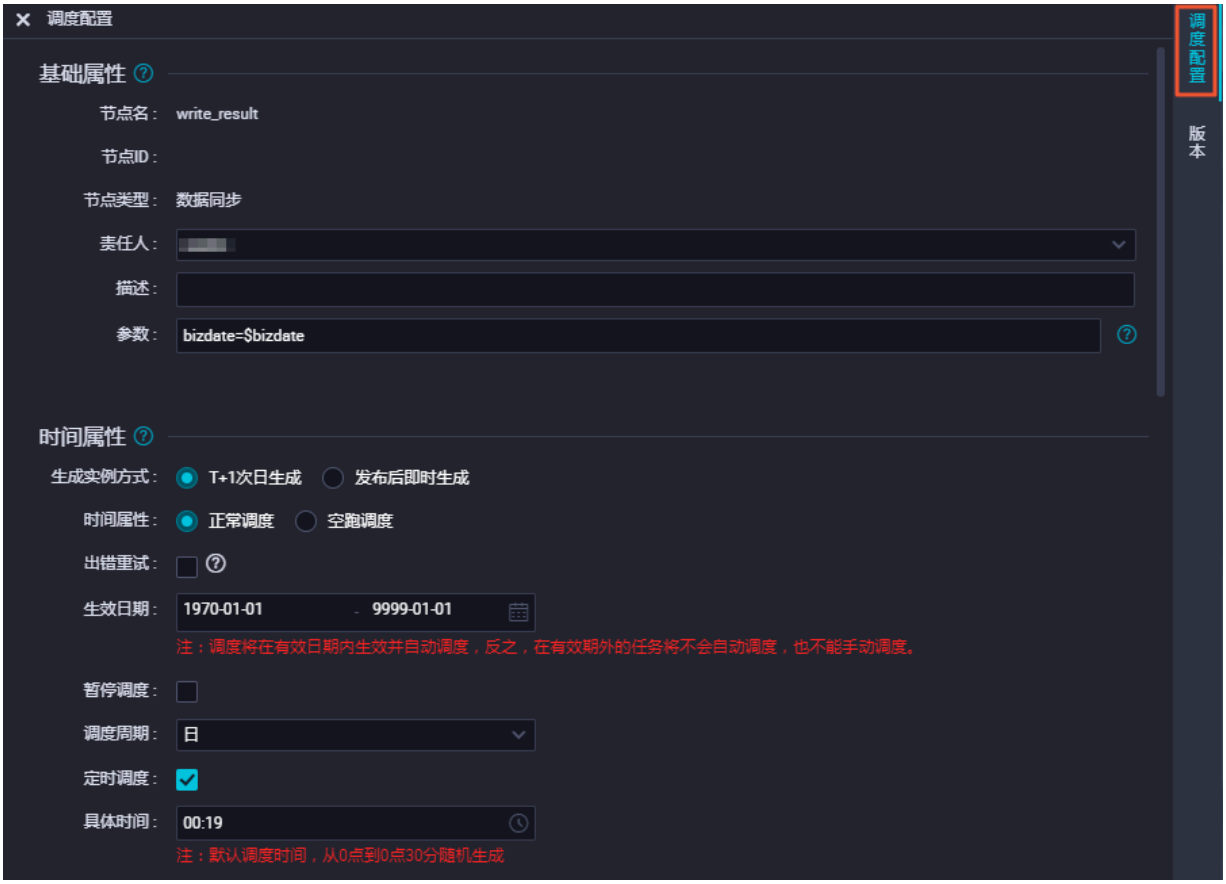
配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。

配置	说明
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	单击当前页面右上角的配置任务资源组，即可指定资源组配置。 任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

配置调度属性

数据同步节点中，经常需要使用调度参数进行数据过滤，下文将为您介绍如何在同步任务中配置调度参数。

进入数据同步节点编辑界面，单击右侧的调度配置。



您可以设置数据同步节点的运行周期、运行时间和调度依赖等属性。由于数据同步节点是ETL工作的开始，所以没有上游节点，此时建议使用工作空间根节点作为上游。

完成数据同步节点的配置后，请保存并提交节点。

2.3.1.2 向导模式配置

本文将为您介绍如何通过数据集成向导模式进行任务配置。

任务配置的操作步骤如下所示：

1. 新建数据源。
2. 新建数据同步节点。
3. 选择数据来源。
4. 选择数据去向。
5. 配置字段的映射关系。
6. 配置作业速率上限、脏数据检查规则等信息。
7. 配置调度属性。

说明：

下文将为您介绍操作步骤的具体实现，以下每个步骤都会跳转到对应的指导文档中。请在完成当前步骤后，单击链接回到本文，继续下一步操作。

新建数据源

同步任务支持多种同构、异构数据源间的数据传输。首先，将需要同步的数据源在数据集成中完成注册。注册完成后，在数据集成配置同步任务时，可以直接选择数据源。数据集成支持同步的数据源类型请参见#unique_153。

确认需要同步的数据源已经被数据集成支持后，可以开始在数据集成中注册数据源。详细的数据源注册步骤请参见配置数据源信息。

说明：

- 有部分数据源数据集成不支持测试连通性，数据源测试连通性的支持详情请参见#unique_154。
- 很多时候，数据源都是创建在本地，没有公网IP或网络无法直达。在这种情况下，配置数据源的时候测试连通性会直接失败，数据集成支持#unique_33来解决这种网络不可达的情况。但在新建同步任务的时候只能选择脚本模式（因为网络不可直达，在向导模式中就无法获取表结构等信息）。

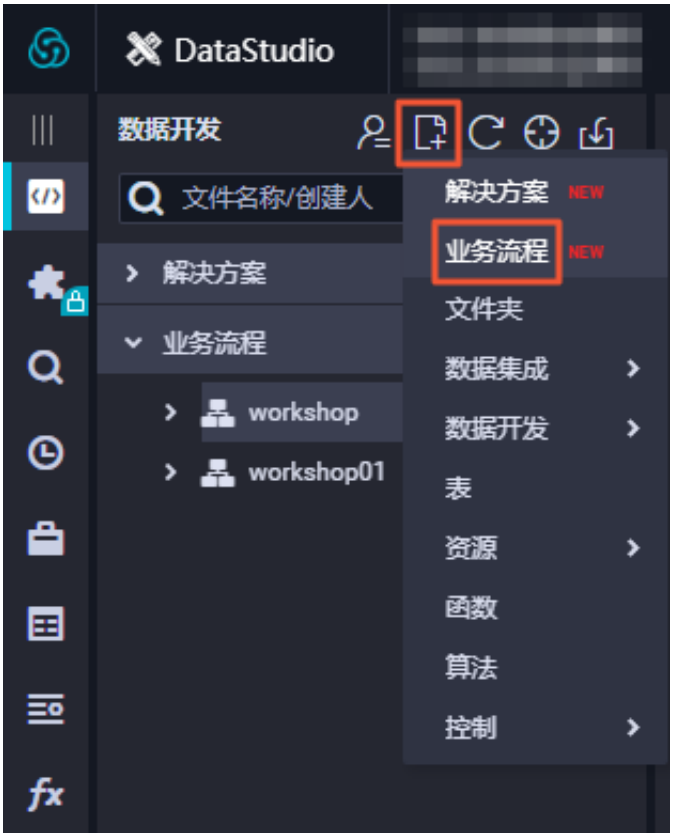
新建数据同步节点

说明：

本文主要为您介绍向导模式下的同步任务配置，在数据集成中新建同步任务时请选择向导模式。

1. 以开发者身份进入DataWorks管理控制台，单击对应工作空间操作栏中的进入数据开发。

2. 进入DataStudio（数据开发）页面，选择新建 > 业务流程。



3. 在新建业务流程对话框中，填写业务流程名称和描述，单击新建。

4. 展开业务流程，右键单击数据集成，选择新建数据集成节点 > 数据同步，输入节点名称，单击提交。

选择数据来源

新建数据同步节点后，首先需要配置数据同步节点的读取端数据源，以及需要同步的表等信息。

说明:

- 选择读取端的数据源时，请参见配置Reader。
- 很多任务在配置读取端数据源时，需要进行数据增量同步。此时可以结合DataWorks提供的#unique_39来获取相对日期，以完成获取增量数据的需求。

选择数据去向

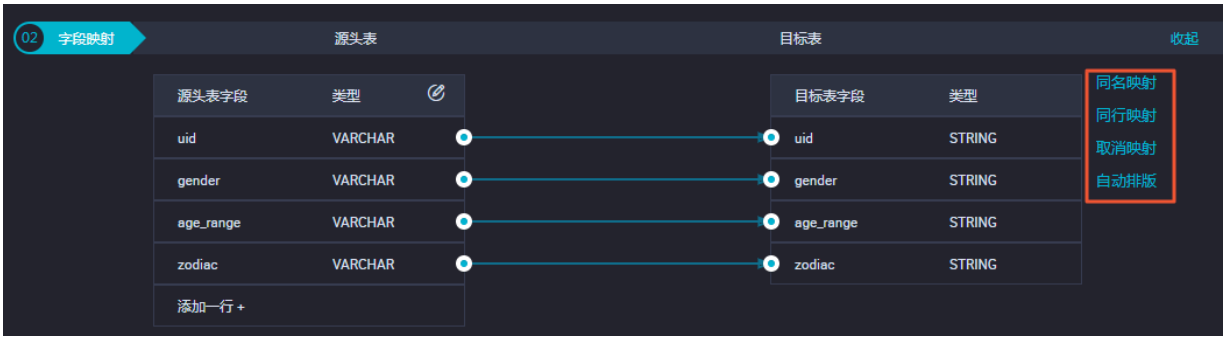
配置完成读取端数据源信息后，可以配置右侧的写入端数据源，以及需要写入的表信息等。

说明:

- 选择写入端的数据源时，请参见配置Writer。
- 很多任务在写入时，需要选择写入模式。比如覆盖写入还是追加写入，针对不同的数据源，有不同的写入模式。

配置字段的映射关系

选择好数据来源和数据去向后，需要指定读取端和写入端列的映射关系，可以选择同名映射、同行映射、取消映射或自动排版。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

说明:

请注意列与列之间映射的字段类型是否数据兼容。

配置通道控制

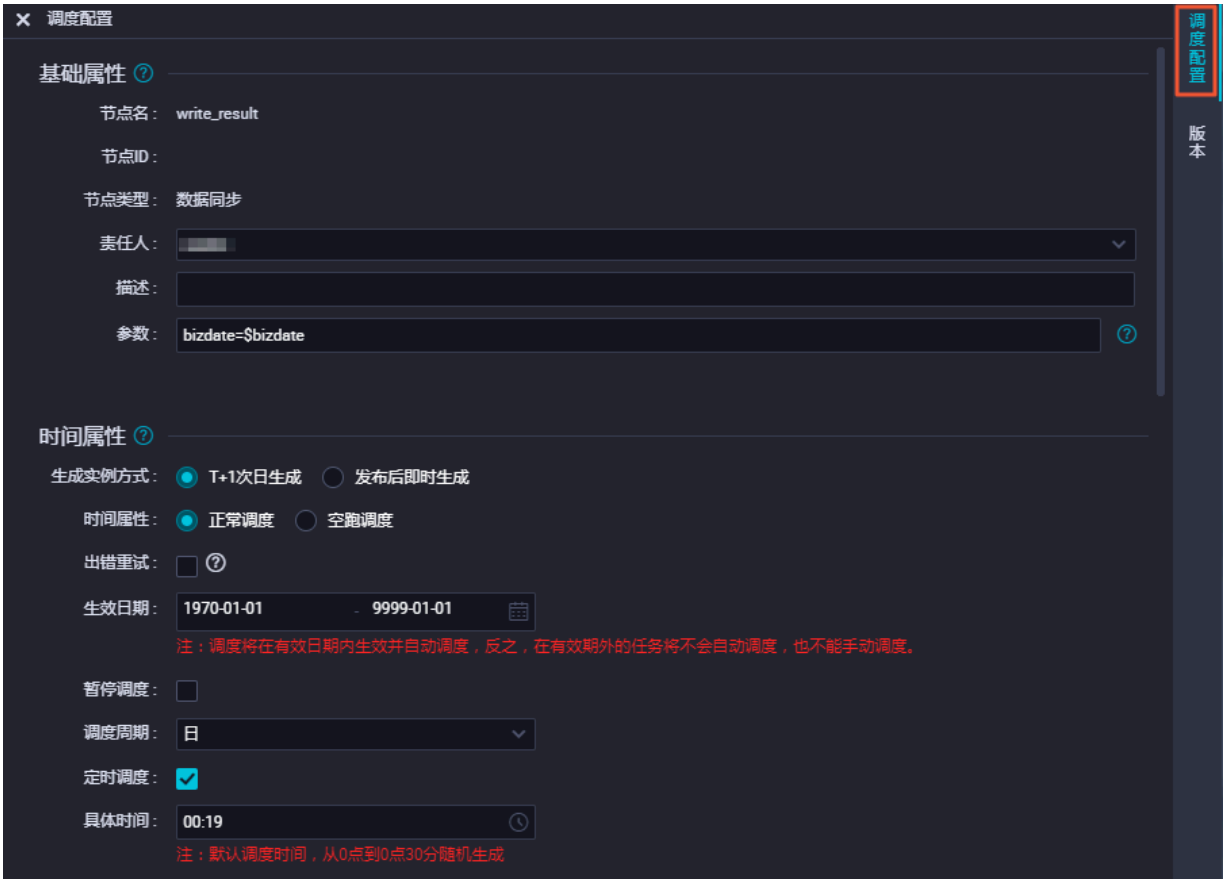
配置完成上述操作后，需要进行通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_157和#unique_33。

配置调度属性

数据同步节点中，经常需要使用调度参数进行数据过滤，下文将为您介绍如何在同步任务中配置调度参数。

进入数据同步节点编辑界面，单击右侧的调度配置。



您可以通过${变量名}的方式声明调度参数变量。当变量声明完成后，在调度的参数属性中写上变量的初始化值，此处变量初始化的值以$[]为标识，其中的内容可以填时间表达式或者一个常量。

例如在代码中写了${today}，在调度参数中赋值today=$[yyyymmdd]，则可获取到当天的日期。如果需要对日期进行加减操作，请参见#unique_39。

您可以设置数据同步节点的运行周期、运行时间和调度依赖等属性。由于数据同步节点是ETL工作的开始，所以没有上游节点，此时建议使用工作空间根节点作为上游。

在同步任务中使用自定义调度参数

在同步任务中只需要在代码中声明如下参数即可。

- bizdate：获取到业务日期，运行日期-1。
- cyctime：获取到当前运行时间，格式为yyyymmddhhmiss。
- Dataworks提供了两个系统默认调度参数bizdate和cyctime 。

完成数据同步节点的配置后，请保存并提交节点。

2.3.1.3 配置DRDS Reader

DRDS Reader插件实现了从DRDS（分布式RDS）读取数据。在底层实现上，DRDS Reader通过JDBC连接远程DRDS数据库，并执行相应的SQL语句，从DRDS库中选取数据。

DRDS的插件目前仅适配了MySQL引擎的场景，DRDS是一套分布式MySQL数据库，并且大部分通信协议遵守MySQL使用场景。

简而言之，DRDS Reader通过JDBC连接器连接至远程的DRDS数据库，根据您配置的信息生成查询SQL语句，发送至远程DRDS数据库，执行该SQL语句并返回结果。然后使用数据同步自定义的数据类型拼装为抽象的数据集，传递给下游Writer处理。

对于您配置的table、column、where等信息，DRDS Reader将其拼接为SQL语句发送至DRDS数据库。不同于普通的MySQL数据库，DRDS作为分布式数据库系统，无法适配所有MySQL的协议，包括复杂的Join等语句，DRDS暂时无法支持。

DRDS Reader支持大部分DRDS类型，但也存在个别类型没有支持的情况，请注意检查您的类型。

DRDS Reader针对DRDS类型的转换列表，如下所示。

类型分类	DRDS数据类型
整数类	INT、TINYINT、SMALLINT、MEDIUMINT和BIGINT
浮点类	FLOAT、DOUBLE和DECIMAL
字符串类	VARCHAR、CHAR、TINYTEXT、TEXT、MEDIUMTEXT和LONGTEXT
日期时间类	DATE、DATETIME、TIMESTAMP、TIME和YEAR

类型分类	DRDS数据类型
布尔类	BIT和BOOL
二进制类	TINYBLOB、MEDIUMBLOB、BLOB、LONGBLOB和VARBINARY

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	所选取的需要同步的表。	是	无
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息，默认使用所有列配置，例如[*]。 · 支持列裁剪，即列可以挑选部分列进行导出。 · 支持列换序，即列可以不按照表组织结构信息的顺序进行导出。 · 支持常量配置，您需要按照MySQL的语法格式，例如[“id”, “`table`“, “1”, “‘bazhen.csy’”, “null”, “to_char(a + 1)”, “2.3” , “true”]。 - id为普通列名。 - table包含保留的列名。 - 1为整型数字常量。 - bazhen.csy为字符串常量。 - null为空指针。 - to_char(a + 1)为计算字符串长度函数表达式。 - 2.3为浮点数。 - true为布尔值。 · column必须显示您指定同步的列集合，不允许为空 。	是	无
where	筛选条件，DRDS Reader根据指定的column、table、where条件拼接SQL，并根据这个SQL进行数据抽取。例如在测试时，可以将where条件指定实际业务场景，往往会选择当天的数据进行同步，可以将where条件指定为STRTODATE(‘${bdp.system.bizdate}’, ‘%Y%m%d’) <= taday AND taday < DATEADD(STRTODATE(‘${bdp.system.bizdate}’, ‘%Y%m%d’), interval 1 day)。 · where条件可以有效地进行业务增量同步。 · where条件不配置或者为空时，视作全表同步数据 。	否	无

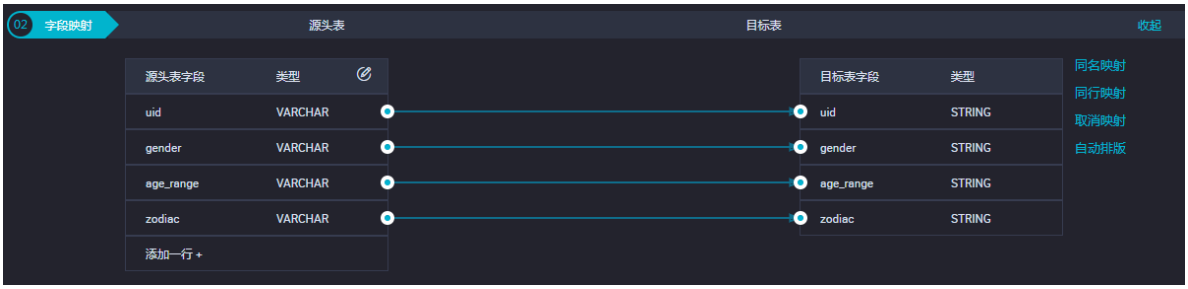
向导开发介绍

1. 配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。
数据过滤	您将要同步数据的筛选条件，暂时不支持limit关键字过滤。SQL语法与选择的数据源一致。
切分键	您可以将源数据表中某一列作为切分键，建议使用主键或有索引的列作为切分键，仅支持类型为整型的字段。 读取数据时，根据配置的字段进行数据分片，实现并发读取，可以提升数据同步效率。 **说明:** 切分键与数据同步中的选择来源有关，配置数据来源时才显示切分键配置项。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段。鼠标放至需要删除的字段上，即可单击删除图标进行删除 。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	添加一行的功能如下所示： · 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。

配置	说明
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

配置一个从DRDS数据库同步抽取数据作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        {
            "stepType":"drds",//插件名。
            "parameter":{
                "datasource":"",//数据源名。
                "column":[//列名。
                    "id",
                    "name"
                ],
                "where":"",//过滤条件。
                "table":"",//表名。
                "splitPk": ""//切分键。
            },
            "name":"Reader",
            "category":"reader"
        },
        {//此处以stream为例，如果您需要其他插件，可以查找相应的文档。
            "stepType":"stream",//插件名
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//是否限流。
            "concurrent":1,//并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}:"Writer"
```

```
                }
            ]
        }
}
```

补充说明

- **一致性视图问题**

  DRDS本身属于分布式数据库，对外无法提供一致性的多库多表视图。不同于MySQL等单库单表同步，DRDS Reader无法抽取同一个时间切片的分库分表快照信息，即DRDS Reader抽取底层不同的分表将获取不同的分表快照，无法保证强一致性。

- **数据库编码问题**

  DRDS本身的编码设置非常灵活，包括指定编码到库、表、字段级别，甚至可以设置不同编码。优先级从高到低为字段、表、库、实例。建议您在库级别将编码统一设置为UTF-8。

  DRDS Reader底层使用JDBC进行数据抽取，JDBC天然适配各类编码，并在底层进行了编码转换。因此DRDS Reader不需要您指定编码，可以自动获取编码并转码。

  对于DRDS底层写入编码和其设定的编码不一致的混乱情况，DRDS Reader对此无法识别，也无法提供解决方案，这类情况的导出结果有可能为乱码。

- **增量数据同步**

  DRDS Reader使用JDBC SELECT语句完成数据抽取工作，因此可以使用`SELECT...WHERE...`进行增量数据抽取，有以下几种方式：

  - **数据库在线应用写入数据库时，填充modify字段为更改时间戳，包括新增、更新、删除（逻辑删除）。对于这类应用，DRDS Reader只需要where条件后跟上一同步阶段时间戳即可。**
  - **对于新增流水型数据，DRDS Reader在where条件后跟上一阶段最大自增ID即可。**

  对于业务上无字段区分新增、修改数据的情况，DRDS Reader无法进行增量数据同步，只能同步全量数据。

- **SQL安全性**

  DRDS Reader提供querySql语句交给您自己实现SELECT抽取语句，DRDS Reader本身对querySql不进行任何安全性校验。

2.3.1.4 配置HBase Reader

HBase Reader插件实现了从HBase中读取数据。在底层实现上，HBase Reader通过HBase的Java客户端连接远程HBase服务，并通过Scan方式读取您指定的rowkey范围内的数据，将读取的数据使用数据集成自定义的数据类型拼装为抽象的数据集，并传递给下游Writer处理。

支持的功能

- 支持HBase0.94.x和HBase1.1.x版本
  - 如果您的HBase版本为HBase0.94.x，Reader端的插件请选择094x。

```
"reader": {
        "plugin": "094x"
    }
```

  - 如果您的HBase版本为HBase1.1.x，Reader端的插件请选择11x。

```
"reader": {
        "plugin": "11x"
    }
```

说明:

HBase1.1.x插件当前可以兼容HBase 2.0，如果您在使用上遇到问题请提交工单。

- 支持normal和multiVersionFixedColumn模式
  - normal模式：把HBase中的表当成普通二维表（横表）进行读取，获取最新版本数据。

```
hbase(main):017:0> scan 'users'
ROW                                   COLUMN+CELL
lisi                                  column=address:city,
timestamp=1457101972764, value=beijing
lisi                                  column=address:contry,
timestamp=1457102773908, value=china
lisi                                  column=address:province,
timestamp=1457101972736, value=beijing
lisi                                  column=info:age, timestamp=
1457101972548, value=27
lisi                                  column=info:birthday,
timestamp=1457101972604, value=1987-06-17
lisi                                  column=info:company,
timestamp=1457101972653, value=baidu
xiaoming                              column=address:city,
timestamp=1457082196082, value=hangzhou
xiaoming                              column=address:contry,
timestamp=1457082195729, value=china
xiaoming                              column=address:province,
timestamp=1457082195773, value=zhejiang
xiaoming                              column=info:age, timestamp=
1457082218735, value=29
xiaoming                              column=info:birthday,
timestamp=1457082186830, value=1987-06-17
xiaoming                              column=info:company,
timestamp=1457082189826, value=alibaba
```

```
2 row(s) in 0.0580 seconds }
```

读取后的数据如下所示。

rowKey	address:city	address:contry	address:province	info:age	info:birthday	info:company
lisi	beijing	china	beijing	27	1987-06-17	baidu
xiaoming	hangzhou	china	zhejiang	29	1987-06-17	alibaba

- multiVersionFixedColumn模式：把HBase中的表当成竖表进行读取。读出的每条记录是四列形式，依次为rowKey、family:qualifier、timestamp和value。读取时需要明确指定要读取的列，把每一个cell中的值，作为一条记录（record），若有多个版本则存在多条记录。

```
hbase(main):018:0> scan 'users',{VERSIONS=>5}
ROW                                   COLUMN+CELL
lisi                                  column=address:city,
timestamp=1457101972764, value=beijing
lisi                                  column=address:contry,
timestamp=1457102773908, value=china
lisi                                  column=address:province,
timestamp=1457101972736, value=beijing
lisi                                  column=info:age, timestamp=
1457101972548, value=27
lisi                                  column=info:birthday,
timestamp=1457101972604, value=1987-06-17
lisi                                  column=info:company,
timestamp=1457101972653, value=baidu
xiaoming                              column=address:city,
timestamp=1457082196082, value=hangzhou
xiaoming                              column=address:contry,
timestamp=1457082195729, value=china
xiaoming                              column=address:province,
timestamp=1457082195773, value=zhejiang
xiaoming                              column=info:age, timestamp=
1457082218735, value=29
xiaoming                              column=info:age, timestamp=
1457082178630, value=24
xiaoming                              column=info:birthday,
timestamp=1457082186830, value=1987-06-17
xiaoming                              column=info:company,
timestamp=1457082189826, value=alibaba
2 row(s) in 0.0260 seconds }
```

读取后的数据（4列） 如下所示。

rowKey	column:qualifier	timestamp	value
lisi	address:city	1457101972764	beijing
lisi	address:contry	1457102773908	china
lisi	address:province	1457101972736	beijing
lisi	info:age	1457101972548	27

rowKey	column:qualifier	timestamp	value
lisi	info:birthday	1457101972604	1987-06-17
lisi	info:company	1457101972653	beijing
xiaoming	address:city	1457082196082	hangzhou
xiaoming	address:contry	1457082195729	china
xiaoming	address:province	1457082195773	zhejiang
xiaoming	info:age	1457082218735	29
xiaoming	info:age	1457082178630	24
xiaoming	info:birthday	1457082186830	1987-06-17
xiaoming	info:company	1457082189826	alibaba

支持的数据类型

支持读取HBase数据类型及HBase Reader针对HBase类型的转换列表如下表所示。

类型分类	数据集成column配置类型	数据库数据类型
整数类	long	short、int和long
浮点类	double	float和double
字符串类	string	binary_string和string
日期时间类	date	date
字节类	bytes	bytes
布尔类	boolean	boolean

参数说明

参数	描述	是否必选	默认值
haveKerberos	haveKerberos值为true时，表示HBase集群需要kerberos认证。  说明: · 如果该值配置为true，必须要配置下面五个kerberos认证相关参数： - kerberosKeytabFilePath - kerberosPrincipal - hbaseMasterKerberosPrincipal - hbaseRegionserverKerberosPrincipal - hbaseRpcProtection · 如果HBase集群没有kerberos认证，则不需要配置以上参数。	否	false
hbaseConfig	连接HBase集群需要的配置信息，JSON格式。必填的配置为hbase.zookeeper.quorum，表示HBase的ZK链接地址。同时可以补充更多HBase client的配置，例如设置scan的cache、batch来优化与服务器的交互。	是	无
mode	读取HBase的模式，支持normal模式、multiVersionFixedColumn模式，即normal/multiVersionFixedColumn。	是	无
table	读取的HBase表名（大小写敏感）。	是	无
encoding	编码方式，UTF-8或GBK，用于对二进制存储的HBase byte[]转为String时的编码。	否	utf-8

参数	描述	是否必选	默认值

column	要读取的HBase字段，normal模式与multiVersionFixedColumn模式下必填。
	· normal模式下
	name指定读取的HBase列，除rowkey外，必须为列族:列名的格式。type指定源数据的类型，format指定日期类型的格式。value指定当前类型为常量，不从HBase读取数据，而是根据value值自动生成对应的列。配置格式如下所示：
	`"column": [ { "name": "rowkey", "type": "string" }, { "value": "test", "type": "string" } ]`
	normal模式下，对于您指定的Column信息，type必须填写，name/value必须选择其一。
	· multiVersionFixedColumn模式
	name指定读取的HBase列，除rowkey外，必须为列族:列名的格式，type指定源数据的类型，format指定日期类型的格式。multiVersionFixedColumn模式下不支持常量列。配置格式如下所示：
	`"column": [ { "name": "rowkey", "type": "string" }, { "name": "info:age", "type": "string" } ]`
		是	无

maxVersion	指定在多版本模式下的HBase Reader读取的版本数，取值只能为-1或大于1的数字，-1表示读取所有版本。	multiVersionFixedColumn模式下必填项	无

参数	描述	是否必选	默认值
range	指定HBase Reader读取的rowkey范围。 · startRowkey：指定开始rowkey。 · endRowkey：指定结束rowkey。 · isBinaryRowkey：指定配置的startRowkey和endRowkey转换为 byte[]时的方式，默认值为false。如果为true，则调用Bytes.toBytesBinary(rowkey)方法进行转换。若为false，则调用 Bytes.toBytes(rowkey)。配置格式如下所示： `"range": { "startRowkey": "aaa", "endRowkey": "ccc", "isBinaryRowkey":false }`	否	无
scanCacheSize	HBase client每次rpc从服务器端读取的行数。	否	256
scanBatchSize	HBase client每次rpc从服务器端读取的列数。	否	100

向导开发介绍

暂不支持向导开发模式开发。

脚本开发介绍

配置一个从HBase抽取数据到本地的作业（normal模式）。

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[
        {
            "stepType":"hbase",//插件名。
            "parameter":{
                "mode":"normal",//读取HBase的模式，支持normal模式、
multiVersionFixedColumn模式。
                "scanCacheSize":"256",//HBase client每次rpc从服务器端读取
的行数。
                "scanBatchSize":"100",//HBase client每次rpc从服务器端读取
的列数。
                "hbaseVersion":"094x/11x",//HBase版本。
                "column":[//字段。
                    {
                        "name":"rowkey",//字段名。
                        "type":"string"//数据类型。
                    },
                    {
                        "name":"columnFamilyName1:columnName1",
```

```
                            "type":"string"
                        },
                        {
                            "name":"columnFamilyName2:columnName2",
                            "format":"yyyy-MM-dd",
                            "type":"date"
                        },
                        {
                            "name":"columnFamilyName3:columnName3",
                            "type":"long"
                        }
                    ],
                    "range":{//指定HBase Reader读取的rowkey范围。
                        "endRowkey":"",//指定结束rowkey。
                        "isBinaryRowkey":true,//指定配置的startRowkey和
endRowkey转换为byte[]时的方式，默认值为false。
                        "startRowkey":""//指定开始rowkey。
                    },
                    "maxVersion":"",//指定在多版本模式下的HBase Reader读取的版
本数。
                    "encoding":"UTF-8",//编码格式。
                    "table":"",//表名。
                    "hbaseConfig":{//连接HBase集群需要的配置信息，JSON格式。
                        "hbase.zookeeper.quorum":"hostname",
                        "hbase.rootdir":"hdfs://ip:port/database",
                        "hbase.cluster.distributed":"true"
                    }
                },
                "name":"Reader",
                "category":"reader"
            },
            {//下面是关于Reader的模板，您可以查看相应的读插件文档。
                "stepType":"stream",
                "parameter":{},
                "name":"Writer",
                "category":"writer"
            }
        ],
        "setting":{
            "errorLimit":{
                "record":"0"//错误记录数。
            },
            "speed":{
                "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
                "concurrent":1,//作业并发数。
            }
        },
        "order":{
            "hops":[
                {
                    "from":"Reader",
                    "to":"Writer"
                }
            ]
        }
```

```
}
```

2.3.1.5 配置HDFS Reader

HDFS Reader提供了读取分布式文件系统数据存储的能力。在底层实现上，HDFS Reader获取分布式文件系统上文件的数据，并转换为数据集成传输协议传递给Writer。

HDFS Reader实现了从Hadoop分布式文件系统HDFS中，读取文件数据并转为数据集成协议的功能 。

示例如下：

TextFile是Hive建表时默认使用的存储格式，数据不进行压缩。本质上TextFile是以文本的形式将数据存放在HDFS中，对于数据集成而言，HDFS Reader在实现上与OSS Reader有很多相似之处。

ORCFile的全名是Optimized Row Columnar File，是对RCFile的优化，这种文件格式可以提供一种高效的方法来存储Hive数据。HDFS Reader利用Hive提供的OrcSerde类，读取解析ORCFile文件的数据。

说明：

- 由于打通默认资源组到HDFS的网络链路比较复杂，建议您使用自定义资源组完成数据同步任务。您需要确保您的自定义资源组具备HDFS的namenode和datanode的网络访问能力。
- HDFS默认情况下，使用网络白名单进行数据安全。基于此种情况，建议您使用自定义资源组完成针对HDFS的数据同步任务。
- 您通过脚本模式配置HDFS同步作业，并不依赖HDFS数据源网络连通性测试通过，针对此类错误您可以临时忽略。
- 数据集成同步进程以admin账号启动，您需要确保操作系统的admin账号具备访问相应HDFS文件的读写权限。

支持的功能

目前HDFS Reader支持的功能如下所示：

- 支持TextFile、ORCFile、rcfile、sequence file、csv和parquet格式的文件，且要求文件内容存放的是一张逻辑意义上的二维表。
- 支持多种类型数据读取（使用String表示），支持列裁剪，支持列常量。
- 支持递归读取、支持正则表达式*和?。
- 支持ORCFile数据压缩，目前支持SNAPPY和ZLIB两种压缩方式。
- 支持sequence file数据压缩，目前支持lzo压缩方式。
- 多个File可以支持并发读取。

- csv类型支持压缩格式有gzip、bz2、zip、lzo、lzo_deflate和snappy。
- 目前插件中Hive版本为1.1.1，Hadoop版本为2.7.1（Apache适配JDK1.6），在Hadoop 2.5.0、Hadoop 2.6.0和Hive 1.2.0测试环境中写入正常。

说明:

HDFS Reader暂不支持单个File多线程并发读取，此处涉及到单个File内部切分算法。

支持的数据类型

由于这些文件表的元数据信息由Hive维护，并存放在Hive自己维护的元数据库（如MySQL）中。目前HDFS Reader不支持对Hive元数据的数据库进行访问查询，因此您在进行类型转换时，必须指定数据类型。

RCFile、ParquetFile、ORCFile、TextFile和SequenceFile中的类型，会默认转为数据集成支持的内部类型，如下表所示。

类型分类	数据集成column配置类型	Hive数据类型
整数类	long	tinyint、smallint、int和bigint
浮点类	double	float和double
字符串类	string	string、char、varchar、struct、map、array、union和binary
日期时间类	date	date和timestamp
布尔类	boolean	boolean

说明如下：

- long：HDFS文件中的整型类型数据，例如123456789。
- double：HDFS文件中的浮点类型数据，例如3.1415。
- bool：HDFS文件中的布尔类型数据，例如true、false，不区分大小写。
- date：HDFS文件中的时间类型数据，例如2014-12-31 00:00:00。

说明:

Hive支持的数据类型TIMESTAMP可以精确到纳秒级别，所以TextFile、ORCFile中TIMESTAMP存放的数据类似于`2015-08-21 22:40:47.397898389`。如果转换的类型配置为数据集成的DATE，转换之后会导致纳秒部分丢失。所以如果需要保留纳秒部分的数据，请配置转换类型为数据集成的字符串类型。

参数说明

参数	描述	必选	默认值
path	要读取的文件路径，如果要读取多个文件，可以使用简单正则表达式匹配，例如/hadoop/data_201704*。 · 当指定单个HDFS文件时，HDFS Reader暂时只能使用单线程进行数据抽取。 · 当指定多个HDFS文件时，HDFS Reader支持使用多线程进行数据抽取，线程并发数通过作业速度mbps指定。 说明: 实际启动的并发数是您的HDFS待读取文件数量和您配置作业速度两者中的小者。 · 当指定通配符，HDFS Reader尝试遍历出多个文件信息。例如指定/代表读取/目录下所有的文件，指定/bazhen/代表读取bazhen目录下游所有的文件。HDFS Reader目前只支持*和?作为文件通配符，语法类似于通常的Linux命令行文件通配符。 说明: · 数据集成会将一个同步作业所有待读取文件视作同一张数据表。您必须自己保证所有的File能够适配同一套schema信息，并且提供给数据集成权限可读。 · 注意分区读取： Hive在建表时，可以指定分区（partition），例如创建分区partition(day="20150820", hour="09")，对应的HDFS文件系统中，相应的表的目录下则会多出/20150820和/09两个目录且/20150820是/09的父目录。 分区会列成相应的目录结构，在按照某个分区读取某个表所有数据时，则只需配置好JSON中path的值即可。例如需要读取表名叫mytable01下分区day为20150820这一天的所有数据，则配置如下： `"path": "/user/hive/warehouse/mytable01/20150820/*"`	是	无
defaultFS	Hadoop HDFS文件系统namenode节点地址。默认资源组不支持Hadoop高级参数HA的配置，请新增自定义资源，详情请参见#unique_33。	是	无

参数	描述	必选	默认值
fileType	文件的类型，目前只支持您配置为text、orc、rc、seq、csv和parquet。HDFS Reader能够自动识别文件的类型，并使用对应文件类型的读取策略。HDFS Reader在做数据同步前，会检查您配置的路径下所有需要同步的文件格式是否和fileType一致，如果不一致任务会失败。 fileType可以配置的参数值列表如下所示。 · text：表示TextFile文件格式。 · orc：表示ORCFile文件格式。 · rc：表示rcfile文件格式。 · seq：表示sequence file文件格式。 · csv：表示普通HDFS文件格式（逻辑二维表）。 · parquet：表示普通parquet file文件格式。 说明: 由于TextFile和ORCFile是两种不同的文件格式，所以HDFS Reader对这两种文件的解析方式也存在差异，这种差异导致Hive支持的复杂复合类型（例如map、array、struct和union）在转换为数据集成支持的String类型时，转换的结果格式略有差异，以map类型为例。 · ORCFile map类型经HDFS Reader解析，转换成数据集成支持的STRING类型后，结果为`{job=80, team=60, person=70}`。 · TextFile map类型经HDFS Reader解析，转换成数据集成支持的STRING类型后，结果为`{job:80, team:60, person:70}`。 如上述转换结果所示，数据本身没有变化，但是表示的格式略有差异。所以如果您配置的文件路径中要同步的字段在Hive中是复合类型的话，建议配置统一的文件格式 。 最佳实践建议： · 如果需要统一复合类型解析出来的格式，建议您在Hive客户端将TextFile格式的表导成ORCFile格式的表。 · 如果是Parquet文件格式，后面的parquetSchema则必填，此属性用来说明要读取的Parquet格式文件的格式。	是	无

参数	描述	必选	默认值
column	读取字段列表，type指定源数据的类型，index指定当前列来自于文本第几列（以0开始），value指定当前类型为常量。不从源头文件读取数据，而是根据value值自动生成对应的列。默认情况下，您可以全部按照STRING类型读取数据，配置为`"column": ["*"]`。 您也可以指定column字段信息（文件数据列和常量列配置二选一），配置如下： `{ "type": "long", "index": 0 //从本地文件文本第一列（下标索引从0开始计数）获取INT字段，index表示从数据文件中获取列数据。 }, { "type": "string", "value": "alibaba" //HDFS Reader内部生成alibaba的字符串字段作为当前字段，value表示常量列。 }` 说明: 建议您指定待读取的每一列数据的下标和类型，避免配置column *通配符。	是	无
fieldDelimiter	读取的字段分隔符，HDFS Reader在读取TextFile数据时，需要指定字段分割符，如果不指定默认为','，HDFS Reader在读取ORCFile时，您无需指定字段分割符，Hive本身的默认分隔符为\u0001。 · 如果您想将每一行作为目的端的一列，分隔符请使用行内容不存在的字符，例如不可见字符\u0001。 · 分隔符不能使用\n。	否	,
encoding	读取文件的编码配置。	否	utf-8
nullFormat	文本文件中无法使用标准字符串定义null（空指针），数据集成提供nullFormat定义哪些字符串可以表示为null。 例如您配置`nullFormat:"null"`，如果源头数据是null，数据集成会将其视作null字段。 说明: 字符串的null（n、u、l、l四个字符）和实际的null不同。	否	无

参数	描述	必选	默认值
compress	当fileType（文件类型）为csv下的文件压缩方式，目前仅支持gzip、bz2、zip、lzo、lzo_deflate、hadoop-snappy和framing-snappy压缩。 说明： · lzo存在lzo和lzo_deflate两种压缩格式，您在配置时需要留心，不要配错。 · 由于snappy目前没有统一的stream format，数据集成目前仅支持最主流的hadoop-snappy（hadoop上的snappy stream format）和framing-snappy（google建议的snappy stream format）。 · rc表示rcfile文件格式。 · orc文件类型下无需填写。	否	无

参数	描述	必选	默认值

parquetSchema	如果您的文件格式类型为Parquet，在配置column配置项的基础上，您还需配置parquetSchema，具体表示parquet存储的类型说明。您需要确保填写parquetSchem后，整体配置符合JSON语法。parquetScema的配置格式说明如下：
	`message MessageType名 {`
	`是否必填，数据类型，列名;`
	`.....;`
	`}`
	· MessageType名：填写名称。
	· 是否必填：required表示非空，optional表示可为空。推荐全填optional。
	· 数据类型：Parquet文件支持BOOLEAN、Int32、Int64、Int96、FLOAT、DOUBLE、BINARY（如果是字符串类型，请填BINARY）和fixed_len_byte_array等类型。
	· 每行列设置必须以分号结尾，最后一行也要写上分号。
	配置示例如下所示。
	`"parquetSchema": "message m { optional int32 minute_id; optional int32 dsp_id; optional int32 adx_pid; optional int64 req; optional int64 res; optional int64 suc; optional int64 imp; optional double revenue; }"`
		否	无

csvReaderConfig	读取CSV类型文件参数配置，Map类型。读取CSV类型文件使用的CsvReader进行读取，会有很多配置，不配置则使用默认值。
	常见配置如下所示。
	`"csvReaderConfig":{ "safetySwitch": false, "skipEmptyRecords": false, "useTextQualifier": false }`
	所有配置项及默认值，配置时csvReaderConfig的map中请严格按照以下字段名字进行配置。
	`boolean caseSensitive = true;`
	`char textQualifier = 34;`
	`boolean trimWhitespace = true;`
	`boolean useTextQualifier = true;//是否使用csv转义字符。`
	`char delimiter = 44;//分隔符`
	`char recordDelimiter = 0;`
		否	无

参数	描述	必选	默认值
kerberosKeytabFilePath	Kerberos认证keytab文件的绝对路径。如果haveKerberos为true，则必选。	否	无
kerberosPrincipal	Kerberos认证Principal名，如****/hadoopclient@**.*** 。如果haveKerberos为true，则必选。 说明: 由于Kerberos需要配置keytab认证文件的绝对路径，您需要在自定义资源组上使用此功能。配置示例如下: `"haveKerberos":true, "kerberosKeytabFilePath":"/opt/datax/**.keytab", "kerberosPrincipal":"**/hadoopclient@**.**"`	否	无

向导开发介绍

暂不支持向导开发模式开发。

脚本开发介绍

配置一个从HDFS抽取数据到本地的作业，详情请参见上述参数说明。

```
{
    "type": "job",
    "version": "2.0",
    "steps": [
        {
            "stepType": "hdfs",//插件名
            "parameter": {
                "path": "",//要读取的文件路径
                "datasource": "",//数据源
                "column": [
                    {
                        "index": 0,//序列号
                        "type": "string"//字段类型
                    },
                    {
                        "index": 1,
                        "type": "long"
                    },
                    {
                        "index": 2,
                        "type": "double"
                    },
                    {
                        "index": 3,
                        "type": "boolean"
                    },
                    {
                        "format": "yyyy-MM-dd HH:mm:ss", //日期格式
```

```
                        "index": 4,
                        "type": "date"
                    }
                ],
                "fieldDelimiter": ","//列分隔符
                "encoding": "UTF-8",//编码格式
                "fileType": ""//文本类型
            },
            "name": "Reader",
            "category": "reader"
        },
        { //下面是关于Writer的模板，您可以查找相应的写插件文档。
            "stepType": "stream",
            "parameter": {},
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "errorLimit": {
            "record": ""//错误记录数
        },
        "speed": {
            "concurrent": 3,//作业并发数
            "throttle": false,//false代表不限流，下面的限流的速度不生效，
true代表限流。
            "dmu": 1//DMU值
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

parquetSchema的HDFS Reader配置样例如下。

说明:

- fileType配置项必须设置为parquet。
- 如果您要读取parquet文件中的部分列，需在parquetSchema配置项中，指定完整schema结构信息，并在column中根据下标，筛选需要的同步列进行列映射。

```
"reader":  {
    "name": "hdfsreader",
    "parameter": {
        "path": "/user/hive/warehouse/addata.db/dw_ads_rtb_monitor_m
inute/thedate=20170103/hour_id=22/*",
        "defaultFS": "h10s010.07100.149:8020",
        "column": [
            {
                "index": 0,
                "type": "string"
            },
            {
```

```
                "index": 1,
                "type": "long"
            },
            {
                "index": 2,
                "type": "double"
            }
        ],
        "fileType": "parquet",
        "encoding": "UTF-8",
        "parquetSchema": "message m { optional int32 minute_id;
optional int32 dsp_id; optional int32 adx_pid; optional int64 req;
optional int64 res; optional int64 suc; optional int64 imp; optional
double revenue; }"
    }
}
```

2.3.1.6 配置MaxCompute Reader

本文将为您介绍MaxCompute Reader支持的数据类型、字段映射和数据源等参数及配置示例。

MaxCompute Reader插件实现了从MaxCompute读取数据的功能，有关MaxCompute的详细介绍请参见MaxCompute简介。

根据您配置的源头项目/表/分区/表字段等信息，在底层实现上可通过Tunnel从MaxCompute系统中读取数据。常用的Tunnel命令请参见Tunnel命令操作。

MaxCompute Reader支持读取分区表、非分区表，不支持读取虚拟视图。当读取分区表时，需要指定出具体的分区配置，例如读取t0表，其分区为pt=1，ds=hangzhou，那么您需要在配置中配置该值。当读取非分区表时，您不能提供分区配置。表字段既可以依序指定全部列、部分列，也可以调整列顺序、指定常量字段和指定分区列（分区列不是表字段）。

支持的数据类型

MaxCompute Reader针对MaxCompute的类型转换列表，如下所示。

类型分类	数据集成column配置类型	数据库数据类型
整数类	long	bigint、int、tinyint和smallint
布尔类	boolean	boolean
日期时间类	date	datetime和timestamp
浮点类	double	float、double和decimal
二进制类	bytes	binary
复杂类	string	array、map和struct

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	读取数据表的表名称（大小写不敏感）。	是	无
partition	读取数据所在的分区信息，支持linux shell通配符，包括表示0个或多个字符，?代表一个字符是否存在。例如现在有分区表test，其存在pt=1/ds=hangzhou、pt=1/ds=shanghai、pt=2/ds=hangzhou和pt=2/ds=beijing四个分区。 · 如果您想读取pt=1/ds=shanghai分区的数据，则应该配置为"partition":"pt=1/ds=shanghai"。 · 如果您想读取pt=1下的所有分区，则应该配置为"partition":"pt=1/ds=*"。 · 如果您想读取整个test表的所有分区的数据，则应该配置为"partition":"pt=*/ds=*"。	如果表为分区表，则必填。如果表为非分区表，则不能填写。	无

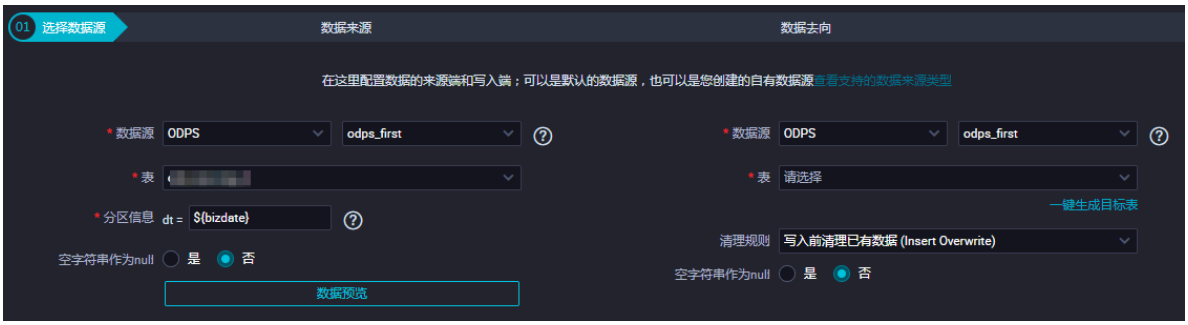
参数	描述	必选	默认值
column	读取MaxCompute源头表的列信息。例如现在有表test，其字段为id、name和age。 · 如果您想依次读取id、name和age，则应该配置为"column":["id","name","age"]或者配置为"column":["*"]。 说明: 不推荐您配置抽取字段为（*），因为它表示依次读取表的每个字段。如果您的表字段顺序调整、类型变更或者个数增减，您的任务会存在源头表列和目的表列不能对齐的风险，则直接导致您的任务运行结果不正确甚至运行失败。 · 如果您想依次读取name和id，则应该配置为"coulumn":["name","id"]。 · 如果您想在源头抽取的字段中添加常量字段（以适配目标表的字段顺序）。例如您想抽取的每一行数据值为age列对应的值，name列对应的值，常量日期值1988-08-08 08:08:08，id列对应的值，那么您应该配置为"column":["age","name","'1988-08-08 08:08:08'","id"]，即常量列首尾用符号' 包住即可。 内部实现上识别常量是通过检查您配置的每一个字段，如果发现有字段首尾都有'，则认为其是常量字段，其实际值为去除'之后的值。 说明: - MaxCompute Reader抽取数据表不是通过MaxCompute的Select SQL语句，所以不能在字段上指定函数。 - column必须显示指定同步的列集合，不允许为空。	是	无

向导开发介绍

打开新建的数据同步节点，即可进行同步任务的配置，详情请参见#unique_164。

1. 选择数据源。

   配置同步任务的数据来源和数据去向。



配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table，选择需要同步的表。
分区信息	填写相应的分区信息。
压缩	可以选择压缩或不压缩。
空字符串是否作为null	可以选择空字符串是否作为null处理。

说明:

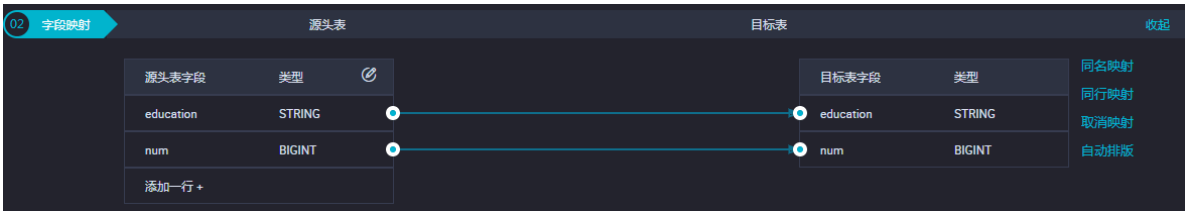
如果是指定所有的列，可以在column配置，例如"column": [""]。partition支持配置多个分区和通配符的配置方法。

- `"partition":"pt=20140501/ds=*"`代表ds中的所有的分区。
- `"partition":"pt=top?"`中的?代表前面的字符是否存在，指pt=top和pt=to两个分区。

可以输入您要同步的分区列，如分区列pt等。例如MaxCompute的分区为pt=${bdp.system.bizdate}，您可以直接将您的分区的名称pt添加到源头表字段中，可能会有未识别的标志直接忽视下一步。如果要同步所有的分区将前面显示的分区值配置成为pt=${*}，如果是同步某个分区可以直接选择您要同步的时间值。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如’abc’、’123’等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。

配置	说明
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

配置一个从MaxCompute抽取数据到本地的作业，详情请参见上述参数说明。

```
{
    "type":"job",
    "version":"2.0",
    "steps":[
        {
            "stepType":"odps",//插件名。
            "parameter":{
                "partition":[],//读取数据所在的分区。
                "isCompress":false,//是否压缩。
                "datasource":"",//数据源。
                "column":[//源头表的列信息。
                    "id"
                ],
                "emptyAsNull":true,
                "table":""//表名。
            },
            "name":"Reader",
            "category":"reader"
        },
        { //下面是关于Writer的模板，您可以查看相应的写插件文档。
            "stepType":"stream",
            "parameter":{
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
```

```
    }
}
```

如果您需要指定MaxCompute的Tunnel Endpoint，可以通过脚本模式手动配置数据源。将上述示例中的"datasource":"",替换为数据源的具体参数，示例如下：

```
"accessId":"********************",
"accessKey":"********************",
"endpoint":"http://service.eu-central-1.maxcompute.aliyun-inc.com/api
",
"odpsServer":"http://service.eu-central-1.maxcompute.aliyun-inc.com/
api",
"tunnelServer":"http://dt.eu-central-1.maxcompute.aliyun.com",
"project":"*****",
```

2.3.1.7 配置MongoDB Reader

本文将为您介绍MongoDB Reader支持的数据类型、字段映射和数据源等参数及配置示例。

MongoDB Reader插件通过MongoDB的Java客户端MongoClient，进行MongoDB的读操作。最新版本的Mongo已经将DB锁的粒度，从DB级别降低到document级别，配合MongoDB强大的索引功能，即可达到高性能读取MongoDB的需求。

说明：

- 如果您使用的是云数据库MongoDB版，MongoDB默认会有root账号。出于安全策略的考虑，数据集成仅支持使用MongoDB数据库对应账号进行连接。您添加使用MongoDB数据源时，也请避免使用root作为访问账号。
- query不支持JS语法。

MongoDB Reader通过数据集成框架从MongoDB并行地读取数据，通过主控的Job程序，按照指定规则对MongoDB中的数据进行分片并行读取，然后将MongoDB支持的类型通过逐一判断转换为数据集成支持的类型。

类型转换列表

MongoDB Reader支持大部分MongoDB类型，但也存在部分没有支持的情况，请注意检查您的数据类型。

MongoDB Reader针对MongoDB类型的转换列表，如下所示。

类型分类	MongoDB数据类型
Long	int、long、document.int和document.long
Double	double和document.double
String	string、array、document.string、document.array和combine

类型分类	MongoDB数据类型
Date	date和document.date
Boolean	bool和document.bool
Bytes	bytes和document.bytes

说明:

- document类型为嵌入文档类型，即object类型。
- combine类型的使用如下:

  使用MongoDB Reader插件读出数据时，支持将MongoDB document中的多个字段合并成一个JSON串。

  例如将MongoDB中的字段导入至MaxCompute，有字段如下（下文均省略了value使用key来代替整个字段）的三个document，其中a、b是所有document均有的公共字段，x_n是不固定字段。

  ```
  doc1:  a b x_1 x_2

  doc2: a b x_2 x_3 x_4

  doc3: a b x_5
  ```

  配置文件中要明确指出需要一一对应的字段，需要合并的字段则需另取名称（不可与document中已存在字段同名），并指定类型为combine，如下所示:

  ```
  "column": [
  {
  "name": "a",
  "type": "string",
  },
  {
  "name": "b",
  "type": "string",
  },
  {
  "name": "doc",
  "type": "combine",
  }
  ]
  ```

  最终导出的MaxCompute结果如下所示:

odps_column1	odps_column2
a	b
a	b

odps_column1	odps_column2
a	b

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
collectionName	MonogoDB的集合名。	是	无
column	MongoDB的文档列名，配置为数组形式表示MongoDB的多个列。 · name：column的名字。 · type：column的类型。 · splitter：因为MongoDB支持数组类型，但数据集成框架本身不支持数组类型，所以MongoDB读出来的数组类型，需要通过该分隔符合并成字符串。	是	无
query	您可以通过该配置型来限制返回MongoDB数据范围，仅支持时间类型。例如您可以配置"query":"{'operationTime':{'$gte':ISODate('${last_day}T00:00:00.424+0800')}}"，限制返回operationTime大于等于${last_day}零点的数据。此处${last_day}为 DataWorks调度参数，格式为$[yyyy-mm-dd]。您可以根据需要具体使用其他MongoDB支持的条件操作符号（$gt、$lt、$gte和$lte等），逻辑操作符（and和or等），函数（max、min、sum、avg和ISODate等），详情请参见MongoDB查询语法。	否	无

向导开发介绍

暂不支持向导开发模式。

脚本开发介绍

配置一个从MongoDB抽取数据到本地的作业，详情请参见上述参数说明。

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[
            "reader": {
            "plugin": "mongodb", //插件名称。
```

```
            "parameter": {
                "datasource": "datasourceName", //数据源名称。
                "collectionName": "tag_data", //集合名称。
                "query":"",
                "column": [
                        {
                            "name": "unique_id", //字段名称。
                            "type": "string"  //字段类型。
                        },
                        {
                            "name": "sid",
                            "type": "string"
                        },
                        {
                            "name": "user_id",
                            "type": "string"
                        },
                        {
                            "name": "auction_id",
                            "type": "string"
                        },
                        {
                            "name": "content_type",
                            "type": "string"
                        },
                        {
                            "name": "pool_type",
                            "type": "string"
                        },
                        {
                            "name": "frontcat_id",
                            "type": "array",
                            "splitter": ""
                        },
                        {
                            "name": "categoryid",
                            "type": "array",
                            "splitter": ""
                        },
                        {
                            "name": "gmt_create",
                            "type": "string"
                        },
                        {
                            "name": "taglist",
                            "type": "array",
                            "splitter": " "
                        },
                        {
                            "name": "property",
                            "type": "string"
                        },
                        {
                            "name": "scorea",
                            "type": "int"
                        },
                        {
                            "name": "scoreb",
                            "type": "int"
                        },
                        {
                            "name": "scorec",
                            "type": "int"
                        },
```

```
                {
                            "name": "a.b",
                            "type": "document.int"
                        },
                        {
                            "name": "a.b.c",
                            "type": "document.array",
                            "splitter": " "
                        }
                ]
            }
        },
        { //下面是关于Writer的模板，您可以查找相应的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

说明:

暂时不支持取出array中的指定元素。

2.3.1.8 配置DB2 Reader

本文将为您介绍DB2 Reader支持的数据类型、字段映射和数据源等参数及配置示例。

DB2 Reader插件实现了从DB2读取数据。在底层实现上，DB2 Reader通过JDBC连接远程DB2数据库，并执行相应的SQL语句，从DB2库选取数据。

DB2 Reader通过JDBC连接器连接至远程的DB2数据库，根据您配置的信息生成查询SQL语句，发送至远程DB2数据库，执行该SQL语句并返回结果。然后使用数据同步自定义的数据类型拼装为抽象的数据集，传递给下游Writer处理。

- 对于您配置的table、column、where等信息，DB2 Reader将其拼接为SQL语句发送至DB2数据库。

· 对于您配置的querySql信息，DB2 Reader直接将其发送到DB2数据库。

DB2 Reader支持大部分DB2类型，但也存在个别类型没有支持的情况，请注意检查您的数据类型。

DB2 Reader针对DB2类型的转换列表，如下所示。

类型分类	DB2数据类型
整数类	SMALLINT
浮点类	DECIMAL、REAL和DOUBLE
字符串类	CHAR、CHARACTER、VARCHAR、GRAPHIC、VARGRAPHIC、LONG VARCHAR、CLOB、LONG VARGRAPHIC和DBCLOB
日期时间类	DATE、TIME和TIMESTAMP
布尔类	—
二进制类	BLOB

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
jdbcUrl	描述的是到DB2数据库的JDBC连接信息，jdbcUrl按照DB2官方规范，DB2格式为`jdbc:db2://ip:port/database`，并可以填写连接附件控制信息。	是	无
username	数据源的用户名。	是	无
password	数据源指定用户名的密码。	是	无
table	所选取的需要同步的表，一个作业只能支持一个表同步。	是	无

参数	描述	必选	默认值
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息，默认使用所有列配置，例如[*]。 · 支持列裁剪，即列可以挑选部分列进行导出。 · 支持列换序，即列可以不按照表schema信息顺序进行导出。 · 支持常量配置，您需要按照DB2 SQL语法格式，例如["id", "1", "'const name'", "null", "upper('abc_lower')", "2.3" , "true"]。 - id为普通列名。 - 1为整型数字常量。 - 'const name'为字符串常量（需要加上一对单引号）。 - null为空指针。 - upper(‘abc_lower’)为函数表达式。 - 2.3为浮点数。 - true为布尔值。 · column必须显示您指定同步的列集合，不允许为空 。	是	无
splitPk	DB2 Reader进行数据抽取时，如果指定splitPk，表示您希望使用splitPk代表的字段进行数据分片，数据同步系统因此会启动并发任务进行数据同步，这样可以大大提供数据同步的效能。 · 推荐splitPk用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 · 目前splitPk仅支持整形数据切分，不支持浮点、字符串和日期等其他类型。如果您指定其他非支持类型，DB2 Reader将报错。	否	""
where	筛选条件，DB2 Reader根据指定的column、table、where条件拼接SQL，并根据这个SQL进行数据抽取。在实际业务场景中，往往会选择当天的数据进行同步，可以将where条件指定为gmt_create>$bizdate。where条件可以有效地进行业务增量同步。如果该值为空，代表同步全表所有的信息。	否	无
querySql	在部分业务场景中，where配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选SQL。当您配置了这项后，数据同步系统就会忽略table、column等配置，直接使用这个配置项的内容对数据进行筛选。 例如需要进行多表join后同步数据，使用select a,b from table_a join table_b on table_a.id = table_b.id 。当您配置querySql时，DB2 Reader直接忽略table、column、where条件的配置。	否	无

参数	描述	必选	默认值
fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据同步系统和服务器端的网络交互次数，能够较大的提升数据抽取性能。 说明: fetchSize值过大（>2048）可能造成数据同步进程OOM。	否	1024

向导开发介绍

暂不支持向导开发模式。

脚本开发介绍

配置一个从DB2数据库同步抽取数据作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        {
            "stepType":"db2",//插件名。
            "parameter":{
                "password":"",//密码。
                "jdbcUrl":"",//DB2数据库的JDBC连接信息。
                "column":[
                    "id"
                ],
                "where":"",//筛选条件。
                "splitPk":"",//splitPk代表的字段进行数据分片。
                "table":"",//表名。
                "username":""//用户名。
            },
            "name":"Reader",
            "category":"reader"
        },
        { //下面是关于Writer的模板，可以查找相应的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
```

```
                "to":"Writer"
            }
        ]
    }
}
```

补充说明

- **主备同步数据恢复问题**

  主备同步问题指DB2使用主从灾备，备库从主库不间断通过binlog恢复数据。由于主备数据同步存在一定的时间差，特别在于某些特定情况，例如网络延迟等问题，导致备库同步恢复的数据与主库有较大差别，从备库同步的数据不是一份当前时间的完整镜像。

- **一致性约束**

  DB2在数据存储划分中属于RDBMS系统，对外可以提供强一致性数据查询接口。例如一次同步任务启动运行过程中，当该库存在其他数据写入方写入数据时，由于数据库本身的快照特性，DB2 Reader完全不会获取到写入更新数据。

  上述是在DB2 Reader单线程模型下数据同步一致性的特性，DB2 Reader可以根据您配置信息使用并发数据抽取，因此不能严格保证数据一致性。

  当DB2 Reader根据splitPk进行数据切分后，会先后启动多个并发任务完成数据同步。多个并发任务相互之间不属于同一个读事务，同时多个并发任务存在时间间隔，因此这份数据并不是完整的、一致的数据快照信息。

  针对多线程的一致性快照需求，目前在技术上无法实现，只能从工程角度解决。工程化的方式存在取舍，在此提供以下解决思路，您可根据自身情况进行选择。

  - **使用单线程同步，即不再进行数据切片。缺点是速度比较慢，但是能够很好保证一致性。**
  - **关闭其他数据写入方，保证当前数据为静态数据，例如锁表、关闭备库同步等。缺点是可能影响在线业务。**

- **数据库编码问题**

  DB2 Reader底层使用JDBC进行数据抽取，JDBC天然适配各类编码，并在底层进行了编码转换。因此DB2 Reader不需您指定编码，可以自动识别编码并转码。

- 增量数据同步

  DB2 Reader使用JDBC SELECT语句完成数据抽取工作，因此可以使用SELECT...WHERE...进行增量数据抽取，有以下几种方式：

  - 数据库在线应用写入数据库时，填充modify字段为更改时间戳，包括新增、更新、删除（逻辑删除）。对于该类应用，DB2 Reader只需要where条件后跟上一同步阶段时间戳即可。
  - 对于新增流水型数据，DB2 Reader在where条件后跟上一阶段最大自增ID即可。

  对于业务上无字段区分新增、修改数据的情况，DB2 Reader无法进行增量数据同步，只能同步全量数据。

- SQL安全性

  DB2 Reader提供querySql语句交给您自己实现SELECT抽取语句，DB2 Reader本身对querySql不进行任何安全性校验。

2.3.1.9 配置MySQL Reader

本文将为您介绍MySQL Reader支持的数据类型、字段映射和数据源等参数及配置示例。

MySQL Reader插件通过JDBC连接器连接至远程的MySQL数据库，根据您配置的信息生成查询SQL语句，发送至远程MySQL数据库，执行该SQL语句并返回结果。然后使用数据同步自定义的数据类型拼装为抽象的数据集，传递给下游Writer处理。

在底层实现上，MySQL Reader插件通过JDBC连接远程MySQL数据库，并执行相应的SQL语句，从MySQL库中选取数据。

MySQL Reader插件支持读取表和视图。表字段可以依序指定全部列、指定部分列、调整列顺序、指定常量字段和配置MySQL的函数，例如now()等。

类型转换列表

MySQL Reader针对MySQL类型的转换列表，如下所示。

类型分类	MySQL数据类型
整数类	INT、TINYINT、SMALLINT、MEDIUMINT和BIGINT
浮点类	FLOAT、DOUBLE和DECIMAL
字符串类	VARCHAR、CHAR、TINYTEXT、TEXT、MEDIUMTEXT和LONGTEXT
日期时间类	DATE、DATETIME、TIMESTAMP、TIME和YEAR
布尔型	BIT和BOOL
二进制类	TINYBLOB、MEDIUMBLOB、BLOB、LONGBLOB和VARBINARY

说明:

- 除上述罗列字段类型外，其他类型均不支持。
- MySQL Reader插件将tinyint(1)视作整型。
- 目前MySQL Reader暂不支持MySQL 8.0及以上版本。

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称，一个数据集成Job只能同步一张表。	是	无

column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息 。默认使用所有列配置，例如[*]。
	- 支持列裁剪：列可以挑选部分列进行导出。
	- 支持列换序：列可以不按照表schema信息顺序进行导出。
	- 支持常量配置：您需要按照MySQL SQL语法格式，例如`["id","table","1","'mingya.wmy'","'null'","to_char(a+1)","2.3","true"]` 。
	  - id为普通列名。
	  - table为包含保留字的列名。
	  - 1为整型数字常量。
	  - 'mingya.wmy'为字符串常量（注意需要加上一对单引号）。
	  - 关于null：
	  ■ ""表示空。
	  ■ null表示null。
	  ■ 'null'表示null这个字符串。
	  - to_char(a+1)为计算字符串长度函数。
	  - 2.3为浮点数。
	  - true为布尔值。
	- column必须显示指定同步的列集合，不允许为空。
		是	无

参数	描述	必选	默认值
splitPk	MySQL Reader进行数据抽取时，如果指定splitPk，表示您希望使用splitPk代表的字段进行数据分片，数据同步因此会启动并发任务进行数据同步，提高数据同步的效能。 · 推荐splitPk用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 · 目前splitPk仅支持整型数据切分，不支持字符串、浮点和日期等其他类型 。如果您指定其他非支持类型，忽略splitPk功能，使用单通道进行同步。 · 如果不填写splitPk，包括不提供splitPk或者splitPk值为空，数据同步视作使用单通道同步该表数据 。	否	无
where	筛选条件，在实际业务场景中，往往会选择当天的数据进行同步，将where条件指定为gmt_create>$bizdate。 · where条件可以有效地进行业务增量同步。如果不填写where语句，包括不提供where的key或value，数据同步均视作同步全量数据。 · 不可以将where条件指定为limit 10，这不符合MySQL SQL WHERE子句约束。	否	无
querySql（高级模式，向导模式不提供）	在部分业务场景中，where配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选SQL 。当配置此项后，数据同步系统就会忽略tables、columns和splitPk配置项，直接使用这项配置的内容对数据进行筛选，例如需要进行多表 join 后同步数据，使用select a,b from table_a join table_b on table_a.id = table_b.id。当您配置querySql时，MySQL Reader直接忽略table、column、where和splitPk条件的配置，querySql优先级大于table、column、where和splitPk选项。datasource通过它解析出用户名和密码等信息。	否	无

参数	描述	必选	默认值
singleOrMulti（仅适用于分库分表）	表示分库分表，向导模式转换成脚本模式主动生成此配置"singleOrMulti":"multi"，但配置脚本任务模板不会直接生成此配置必须手动添加，否则只会识别第一个数据源。singleOrMulti仅前端使用，后端没有用此进行分库分表判断。	是	multi

向导开发介绍

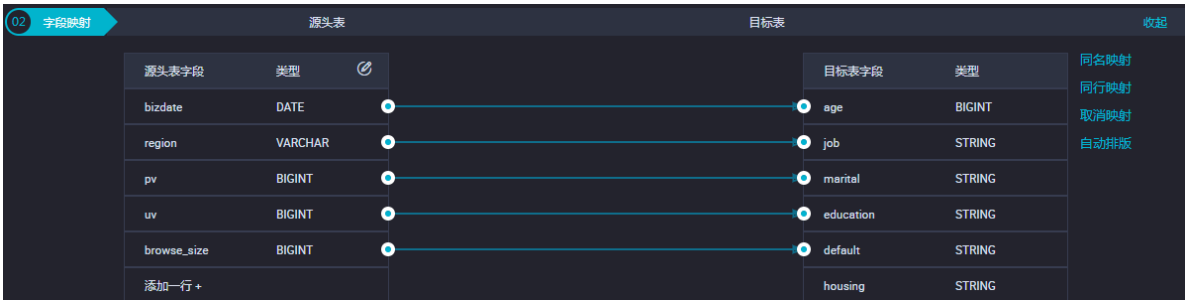
1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。
数据过滤	您将要同步数据的筛选条件，暂时不支持limit关键字过滤。SQL语法与选择的数据源一致。
切分键	您可以将源数据表中某一列作为切分键，建议使用主键或有索引的列作为切分键，仅支持类型为整型的字段。 读取数据时，根据配置的字段进行数据分片，实现并发读取，可以提升数据同步效率。 **说明:** 切分键与数据同步中的选择来源有关，配置数据来源时才显示切分键配置项。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除 。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如’abc’、’123’等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。

配置	说明
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

单库单表的脚本样例如下，详情请参见上述参数说明。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        {
            "stepType":"mysql",//插件名。
            "parameter":{
                "column":[//列名。
                    "id"
                ],
                "connection":[
                    {   "querysql":["select a,b from join1 c join
join2 d on c.id = d.id;"], //使用字符串的形式，将querySql写在connection中。
                        "datasource":"",//数据源。
                        "table":[//表名。
                            "xxx"
                        ]
                    }
                ],
                "where":"",//过滤条件。
                "splitPk":"",//切分键。
                "encoding":"UTF-8"//编码格式。
            },
            "name":"Reader",
            "category":"reader"
        },
        {//下面是关于writer的模板，您可以查找相应的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
```

```
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

分库分表的脚本样例如下，详情请参见上述参数说明。

说明:

分库分表是指在MySQL Reader端可以选择多个MySQL数据表，且表结构要一致。

```
{
    "type": "job",
    "version": "1.0",
    "configuration": {
        "reader": {
            "plugin": "mysql",
            "parameter": {
                "connection": [
                    {
                        "table": [
                            "tbl1",
                            "tbl2",
                            "tbl3"
                        ],
                        "datasource": "datasourceName1"
                    },
                    {
                        "table": [
                            "tbl4",
                            "tbl5",
                            "tbl6"
                        ],
                        "datasource": "datasourceName2"
                    }
                ],
                "singleOrMulti": "multi",
                "splitPk": "db_id",
                "column": [
                    "id", "name", "age"
                ],
                "where": "1 < id and id < 100"
            }
        },
        "writer": {
        }
    }
}
```

2.3.1.10 配置Oracle Reader

本文为您介绍Oracle Reader支持的数据类型、字段映射和数据源等参数及配置举例。

Oracle Reader插件实现了从Oracle读取数据。在底层实现上，Oracle Reader通过JDBC连接远程Oracle数据库，并执行相应的SQL语句，从Oracle数据库中选取数据。

公共云上RDS/DRDS不提供Oracle存储引擎，Oracle Reader目前更多用于专有云数据迁移、数据集成项目。

简单来说，Oracle Reader通过JDBC连接器连接到远程的Oracle数据库，根据您配置的信息生成查询语句，并发送至远程Oracle数据库。然后使用CDP自定义的数据类型，将该SQL执行返回结果拼装为抽象的数据集，并传递给下游Writer处理。

- 对于您配置的table、column和where信息，Oracle Reader将其拼接为SQL语句，发送至Oracle数据库。
- 对于您配置的querySql信息，Oracle直接将其发送至Oracle数据库。

类型转换列表

Oracle Reader支持大部分Oracle类型，但也存在部分类型没有支持的情况，请注意检查您的数据类型 。

Oracle Reader针对Oracle类型的转换列表，如下所示。

类型分类	Oracle数据类型
整数类	NUMBER、RAWID、INTEGER、INT和SMALLINT
浮点类	NUMERIC、DECIMAL、FLOAT、DOUBLE PRECISIOON和REAL
字符串类	LONG、CHAR、NCHAR、VARCHAR、VARCHAR2、NVARCHAR2、CLOB、NCLOB、CHARACTER、CHARACTER VARYING、CHAR VARYING、NATIONAL CHARACTER、NATIONAL CHAR、NATIONAL CHARACTER VARYING、NATIONAL CHAR VARYING和NCHAR VARYING
日期时间类	TIMESTAMP和DATE
布尔型	BIT和BOOL
二进制类	BLOB、BFILE、RAW和LONG RAW

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无

参数	描述	是否必选	默认值
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息。默认使用所有列配置，例如["*"]。 · 支持列裁剪，即列可以挑选部分列进行导出。 · 支持列换序，即列可以不按照表schema信息顺序进行导出。 · 支持常量配置，您需要按照JSON格式进行配置。 `["id", "1", "'mingya.wmy'", "null", "to_char(a + 1)", "2.3" , "true"]` - id为普通列名。 - 1为整型数字常量。 - 'mingya.wmy'为字符串常量（注意需要加上一对单引号）。 - null为空指针。 - to_char(a + 1)为表达式。 - 2.3为浮点数。 - true为布尔值。 · column必须显示填写，不允许为空。	是	无
splitPk	Oracle Reader进行数据抽取时，如果指定splitPk，表示您希望使用splitPk代表的字段进行数据分片，数据同步因此会启动并发任务进行数据同步，这样可以大大提高数据同步的效能。 · 推荐splitPk用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 · splitPk支持数字类型、字符串类型，浮点和日期等其他类型。 · 如果不填写splitPk，将视作您不对单表进行切分，Oracle Reader使用单通道同步全量数据。	否	无
where	筛选条件，Oracle Reader根据指定的column、table和where条件拼接SQL，并根据这个SQL进行数据抽取。例如在做测试时，可以将where条件指定为row_number()。在实际业务场景中，往往会选择当天的数据进行同步，可以将where条件指定为`id>2 and sex=1`。 · where条件可以有效地进行业务增量同步 。 · where条件不配置或为空时，将视作全表同步数据 。	否	无

参数	描述	是否必选	默认值
querySql（高级模式，向导模式不支持）	在部分业务场景中，where配置项不足以描述所筛选的条件，您可以通过该配置来自定义筛选SQL。当您配置这项后，数据同步系统就会忽略table和column等配置，直接使用这个配置项的内容对数据进行筛选，例如需要进行多表join后同步数据，使用`select a,b from table_a join table_b on table_a.id = table_b.id`。当您配置querySql时，Oracle Reader直接忽略table、column和where条件的配置。	否	无
fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据同步系统和服务器端的网络交互次数，能够较大的提升数据抽取性能。 说明: fetchSize值过大（>2048）可能造成数据同步进程OOM。	否	1024

向导开发介绍

1. 选择数据源。

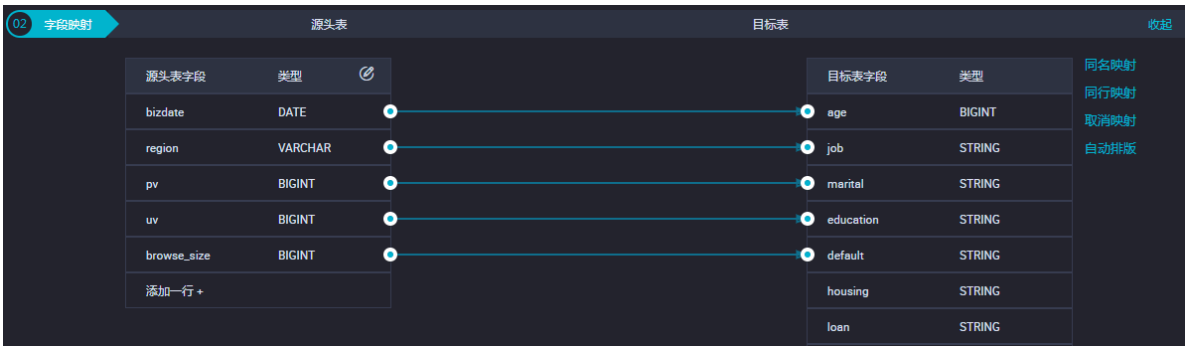
   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。
数据过滤	您将要同步数据的筛选条件，暂时不支持limit关键字过滤。SQL语法与选择的数据源一致。

配置	说明
切分键	您可以将源数据表中某一列作为切分键，建议使用主键或有索引的列作为切分键，仅支持类型为整型的字段。 读取数据时，根据配置的字段进行数据分片，实现并发读取，可以提升数据同步效率。 说明: 切分键与数据同步中的选择来源有关，配置数据来源时才显示切分键配置项。

2. 字段映射，即上述参数说明中的column。

左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除 。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如’abc’、’123’等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

配置一个从Oracle数据库同步抽取数据的作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        {
            "stepType":"oracle",
            "parameter":{
                "fetchSize":1024,//该配置项定义了插件和数据库服务器端每次批量数据获取条数。
                "datasource":"",//填写添加的数据源名。
                "column":[//列名。
                    "id",
                    "name"
                ],
                "where":"",//筛选条件。
                "splitPk":"",//切分键。
                "table":""//表名。
            },
            "name":"Reader",
            "category":"reader"
        },
        {//此处以stream为例，如果您需要使用其他插件，可以查找对应的插件填写相应的内容。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
```

```
                "category":"writer"
            }
        ],
        "setting":{
            "errorLimit":{
                "record":"0"//错误记录数。
            },
            "speed":{
                "throttle":false,////false代表不限流，下面的限流的速度不生效，
true代表限流。
                "concurrent":1,//作业并发数。
            }
        },
        "order":{
            "hops":[
                {
                    "from":"Reader",
                    "to":"Writer"
                }
            ]
        }
} "to":"Writer"
                }
            ]
        }
}
```

补充说明

- **主备同步数据恢复问题**

  主备同步问题指Oracle使用主从灾备，备库从主库不间断通过binlog恢复数据。由于主备数据同步存在一定的时间差，特别在于某些特定情况，例如网络延迟等问题，导致备库同步恢复的数据与主库有较大差别，从备库同步的数据不是一份当前时间的完整镜像。

- 一致性约束

  Oracle在数据存储划分中属于RDBMS系统，对外可以提供强一致性数据查询接口。例如一次同步任务启动运行过程中，当该库存在其他数据写入方写入数据时，由于数据库本身的快照特性，Oracle Reader完全不会获取到写入更新数据。

  上述是在Oracle Reader单线程模型下数据同步一致性的特性，Oracle Reader可以根据您配置的信息使用并发数据抽取，因此不能严格保证数据一致性。

  当Oracle Reader根据splitPk进行数据切分后，会先后启动多个并发任务完成数据同步。多个并发任务相互之间不属于同一个读事务，同时多个并发任务存在时间间隔。因此这份数据并不是完整的、一致的数据快照信息。

  针对多线程的一致性快照需求，目前在技术上无法实现，只能从工程角度解决。工程化的方式存在取舍，在此提供以下解决思路，您可以根据自身情况进行选择。

  - 使用单线程同步，即不再进行数据切片。缺点是速度比较慢，但是能够很好保证一致性。
  - 关闭其他数据写入方，保证当前数据为静态数据，例如锁表、关闭备库同步等。缺点是可能影响在线业务。
- 数据库编码问题

  Oracle Reader底层使用JDBC进行数据抽取，JDBC天然适配各类编码，并在底层进行了编码转换。因此Oracle Reader不需您指定编码，可以自动获取编码并转码。
- 增量数据同步

  Oracle Reader使用JDBC SELECT语句完成数据抽取工作，因此可以使用`SELECT…WHERE…`进行增量数据抽取，有以下几种方式：

  - 数据库在线应用写入数据库时，填充modify字段为更改时间戳，包括新增、更新、删除（逻辑删除）。对于该类应用，Oracle Reader只需要where条件后跟上一同步阶段时间戳即可。
  - 对于新增流水型数据，Oracle Reader在where条件后跟上一阶段最大自增ID即可。

  对于业务上无字段区分新增、修改数据的情况，Oracle Reader无法进行增量数据同步，只能同步全量数据。
- SQL安全性

  Oracle Reader提供querySql语句交给您自己实现SELECT抽取语句，Oracle Reader本身对querySql不进行任何安全性校验。

2.3.1.11 配置OSS Reader

本文将为您介绍OSS Reader支持的数据类型、字段映射和数据源等参数及配置示例。

OSS Reader插件提供了读取OSS数据存储的能力。在底层实现上，OSS Reader使用OSS官方Java SDK获取OSS数据，并转换为数据同步传输协议传递给Writer。

- 如果您想对OSS产品有更深了解，请参见OSS产品概述。
- OSS Java SDK的详细介绍，请参见阿里云OSS Java SDK。
- 处理OSS等非结构化数据的详细介绍，请参见处理非结构化数据。

OSS Reader实现了从OSS读取数据并转为数据集成/DataX协议的功能，OSS本身是无结构化数据存储。对于数据集成/DataX而言，目前OSS Reader支持的功能如下所示。

- 支持且仅支持读取TXT格式的文件，且要求TXT中schema为一张二维表。
- 支持类CSV格式文件，自定义分隔符。
- 支持多种类型数据读取（使用String表示），支持列裁剪、列常量。
- 支持递归读取、支持文件名过滤。
- 支持文本压缩，现有压缩格式为gzip、bzip2和zip。

  说明:

  一个压缩包不允许多文件打包压缩。
- 多个Object可以支持并发读取。

OSS Reader暂时不能实现以下功能。

- 单个Object（File）支持多线程并发读取。
- 单个Object在压缩情况下，从技术上无法支持多线程并发读取。

OSS Reader支持OSS中的BIGINT、DOUBLE、STRING、DATATIME和BOOLEAN数据类型。

支持的数据类型

类型分类	数据集成column配置类型	数据库数据类型
整数类	long	long
字符串类	string	string
浮点类	double	double
布尔类	boolean	bool
日期时间类	date	date

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
Object	OSS的Object信息，此处可以支持填写多个Object。例如xxx的bucket中有yunshi文件夹，文件夹中有ll.txt文件，则Object直接填yunshi/ll.txt。 · 当指定单个OSS Object时，OSS Reader暂时只能使用单线程进行数据抽取。后期将考虑在非压缩文件情况下针对单个Object可以进行多线程并发读取。 · 当指定多个OSS Object时，OSS Reader支持使用多线程进行数据抽取。线程并发数通过通道数指定。 · 当指定通配符时，OSS Reader尝试遍历出多个Object信息。例如配置 abc[0-9]表示abc0、abc1、abc2、abc3等。配置通配符会导致内存溢出，通常不建议您进行配置。详情请参见OSS产品概述。 说明: · 数据同步系统会将一个作业下同步的所有Object视作同一张数据表。您必须保证所有的Object能够适配同一套schema信息。 · 请注意控制单个目录下的文件个数，否则可能会触发系统OutOfMemoryError报错。若遇到此情况，请将文件拆分到不同目录后再尝试进行同步。	是	无

参数	描述	必选	默认值

column	读取字段列表，type指定源数据的类型，index指定当前列来自于文本第几列（以0开始），value指定当前类型为常量，不是从源头文件读取数据，而是根据value值自动生成对应的列。
	默认情况下，您可以全部按照String类型读取数据，配置如下：
	`json`
	`"column": ["*"]`
	您可以指定column字段信息，配置如下：
	`json`
	`"column":`
	`{`
	`"type": "long",`
	`"index": 0 //从OSS文本第一列获取int字段。`
	`},`
	`{`
	`"type": "string",`
	`"value": "alibaba" //从OSSReader内部生成alibaba的字符串字段作为当前字段。`
	`}`
	说明：
	对于您指定的column信息，type必须填写，index/value必须选择其一。
		是	全部按照STRING类型读取。

fieldDelimiter	读取的字段分隔符。 说明： OSS Reader在读取数据时，需要指定字段分割符，如果不指定默认为（,），界面配置中也会默认填写为（,）。	是	,
compress	文本压缩类型，默认不填写（即不压缩）。支持压缩类型为gzip、bzip2和zip。	否	不压缩
encoding	读取文件的编码配置。	否	utf-8
nullFormat	文本文件中无法使用标准字符串定义null（空指针），数据同步系统提供nullFormat定义哪些字符串可以表示为null。例如您配置`nullFormat="null"`，那么如果源头数据是"null"，数据同步系统会视作null字段。针对空字符串，需要加一层转义：\N=\\N。	否	无
skipHeader	类CSV格式文件可能存在表头为标题情况，需要跳过。默认不跳过，压缩文件模式下不支持skipHeader。	否	false

参数	描述	必选	默认值
csvReaderConfig	读取CSV类型文件参数配置，Map类型。读取CSV类型文件使用的CsvReader进行读取，会有很多配置，不配置则使用默认值。	否	无

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
Object前缀	即上述参数说明中的Object。 说明: 假如您的OSS文件名有根据每天的时间命名的部分，例如aaa/20171024abc.txt，关于Object系统参数就可以设置aaa/${bdp.system.bizdate}abc.txt。
列分隔符	即上述参数说明中的fieldDelimiter，默认值为（,）。
编码格式	即上述参数说明中的encoding，默认值为utf-8。
null值	即上述参数说明中的nullFormat，将要表示为空的字段填入文本框，如果源端存在则将对应的部分转换为空。
压缩格式	即上述参数说明中的compress，默认值为不压缩。
是否包含表头	即上述参数说明中的skipHeader，默认值为No。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。

配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

脚本配置样例如下所示，具体参数填写请参见参数说明。

```
{
```

```
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        {
            "stepType":"oss",//插件名。
            "parameter":{
                "nullFormat":"",//定义可以表示为null的字符串。
                "compress":"",//文本压缩类型。
                "datasource":"",//数据源。
                "column":[//字段。
                    {
                        "index":0,//列序号。
                        "type":"string"//数据类型。
                    },
                    {
                        "index":1,
                        "type":"long"
                    },
                    {
                        "index":2,
                        "type":"double"
                    },
                    {
                        "index":3,
                        "type":"boolean"
                    },
                    {
                        "format":"yyyy-MM-dd HH:mm:ss", //时间格式。
                        "index":4,
                        "type":"date"
                    }
                ],
                "skipHeader":"",//类CSV格式文件可能存在表头为标题情况，需要跳过。
                "encoding":"",//编码格式。
                "fieldDelimiter":",",//字段分隔符。
                "fileFormat": "",//文本类型。
                "object":[]//object前缀。
            },
            "name":"Reader",
            "category":"reader"
        },
        {//下面是关于Writer的模板，可以查看相应的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":""//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
```

```
            }
        ]
    }
}
```

ORC/Parquet文件读取OSS

目前通过复用HDFS Reader的方式完成OSS读取ORC/Parquet格式的文件，在OSS Reader已有参数的基础上，增加了Path、FileFormat等扩展配置参数，参数含义请参见配置HDFS Reader。

- 以ORC文件格式读取OSS，示例如下：

```
{
      "stepType": "oss",
      "parameter": {
        "datasource": "",
        "fileFormat": "orc",
        "path": "/tests/case61/orc__691b6815_9260_4037_9899_a
a8e61dc7e4b",
        "column": [
          {
            "index": 0,
            "type": "long"
          },
          {
            "index": "1",
            "type": "string"
          },
          {
            "index": "2",
            "type": "string"
          }
        ]
      }
    }
```

- 以Parquet文件格式读取OSS，示例如下：

```
{
      "stepType": "oss",
      "parameter": {
        "datasource": "",
        "fileFormat": "parquet",
        "path": "/tests/case61/parquet",
        "parquetSchema": "message test { required int64 int64_col;
\n required binary str_col (UTF8);\nrequired group params (MAP) {\
nrepeated group key_value {\nrequired binary key (UTF8);\nrequired
 binary value (UTF8);\n}\n}\nrequired group params_arr (LIST) {\n
  repeated group list {\n    required binary element (UTF8);\n  }\n
}\nrequired group params_struct {\n  required int64 id;\n required
binary name (UTF8);\n }\nrequired group params_arr_complex (LIST) {\
n  repeated group list {\n    required group element {\n required
int64 id;\n required binary name (UTF8);\n}\n  }\n}\nrequired group
params_complex (MAP) {\nrepeated group key_value {\nrequired binary
key (UTF8);\nrequired group value {\n  required int64 id;\n required
 binary name (UTF8);\n  }\n}\n}\nrequired group params_struct_comple
```

```
x {\n  required int64 id;\n required group detail {\n  required
int64 id;\n required binary name (UTF8);\n  }\n  }\n}",
        "column": [
          {
            "index": 0,
            "type": "long"
          },
          {
            "index": "1",
            "type": "string"
          },
          {
            "index": "2",
            "type": "string"
          },
          {
            "index": "3",
            "type": "string"
          },
          {
            "index": "4",
            "type": "string"
          },
          {
            "index": "5",
            "type": "string"
          },
          {
            "index": "6",
            "type": "string"
          },
          {
            "index": "7",
            "type": "string"
          }
        ]
      }
    }
```

2.3.1.12 配置FTP Reader

本文将为您介绍FTP Reader支持的数据类型、字段映射和数据源等参数及配置示例。

FTP Reader为您提供读取远程FTP文件系统数据存储的功能。在底层实现上，FTP Reader获取远程FTP文件数据，并转换为数据同步传输协议传递给Writer。

本地文件内容存放的是一张逻辑意义上的二维表，例如CSV格式的文本信息。

FTP Reader实现了从远程FTP文件读取数据并转为数据同步协议的功能，远程FTP文件本身是无结构化数据存储，对于数据同步而言，目前FTP Reader支持的功能如下所示。

- **支持且仅支持读取TXT的文件，并要求TXT中的schema为一张二维表。**
- **支持类CSV格式文件，自定义分隔符。**
- **支持多种类型数据读取（使用STRING表示）、支持列裁剪和列常量。**
- **支持递归读取、支持文件名过滤。**
- **支持文本压缩，现有压缩格式为gzip、bzip2、zip、lzo和lzo_deflate。**

- 多个File可以支持并发读取。

暂时不支持以下两种功能。

- 单个File支持多线程并发读取，此处涉及到单个File内部切分算法。
- 单个File在压缩情况下，从技术上无法支持多线程并发读取。

远程FTP文件本身不提供数据类型，该类型是DataX FtpReader定义。

DataX内部类型	远程FTP文件数据类型
LONG	LONG
DOUBLE	DOUBLE
STRING	STRING
BOOLEAN	BOOLEAN
DATE	DATE

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无

参数	描述	必选	默认值
path	远程FTP文件系统的路径信息，这里可以支持填写多个路径。 · 当指定单个远程FTP文件，FTP Reader暂时只能使用单线程进行数据抽取。后期会在非压缩文件情况下针对单个File进行多线程并发读取。 · 当指定多个远程FTP文件，FTP Reader支持使用多线程进行数据抽取。线程并发数通过通道数指定。 · 当指定通配符，FTP Reader尝试遍历出多个文件信息。例如，指定/代表读取/目录下所有的文件，指定/bazhen/代表读取bazhen目录下游所有的文件。FTP Reader目前只支持*作为文件通配符。 说明: · 通常不建议您使用*，易导致任务运行报JVM内存溢出的错误。 · 数据同步会将一个作业下同步的所有Text File视作同一张数据表。您必须自己保证所有的File能够适配同一套schema信息。 · 您必须保证读取文件为类CSV格式，并且提供给数据同步系统权限可读。 · 如果Path指定的路径下没有符合匹配的文件抽取，同步任务将报错。	是	无

参数	描述	必选	默认值
column	读取字段列表，type指定源数据的类型，index指定当前列来自于文本第几列（以0开始），value指定当前类型为常量，不从源头文件读取数据，而是根据value值自动生成对应的列。 默认情况下，您可以全部按照STRING类型读取数据，配置为`"column":["*"]`。您可以指定column字段信息，配置如下： `{ "type": "long", "index": 0 //从远程FTP文件文本第一列获取INT字段。 }, { "type": "string", "value": "alibaba" //从FTP Reader内部生成alibaba的字符串字段作为当前字段。 }` 对于您指定的column信息，type必须填写，index/value必须选择其一。	是	全部按照STRING类型读取
fieldDelimiter	读取的字段分隔符。 说明： FTP Reader在读取数据时，需要指定字段分割符，如果不指定会默认为（,），界面配置也会默认填写（,）。	是	,
skipHeader	类CSV格式文件可能存在表头为标题情况，需要跳过。默认不跳过，压缩文件模式下不支持skipHeader。	否	false
encoding	读取文件的编码配置。	否	utf-8
nullFormat	文本文件中无法使用标准字符串定义null（空指针），数据同步提供nullFormat定义哪些字符串可以表示为null。 例如，您配置`nullFormat:"null"`，如果源头数据是null，则数据同步视作null字段。	否	无
markDoneFileName	标档文件名，数据同步前检查标档文件。如果标档文件不存在，等待一段时间重新检查标档文件，如果检查到标档文件开始执行同步任务。	否	无
maxRetryTime	表示检查标档文件重试次数，默认重试60次，每一次重试间隔为1分钟，共60分钟。	否	60

参数	描述	必选	默认值
csvReaderConfig	读取CSV类型文件参数配置，Map类型。读取CSV类型文件使用的CsvReader进行读取，会有很多配置，不配置则使用默认值。	否	无
fileFormat	读取的文件类型，默认情况下文件作为csv格式文件进行读取，内容被解析为逻辑上的二维表结构处理。如果您配置为binary，则表示按照纯粹二进制格式进行复制传输。 通常在FTP、OSS等存储之间进行目录结构对等复制时使用，通常不需配置此项。	否	无

向导开发介绍

1. 选择数据源。

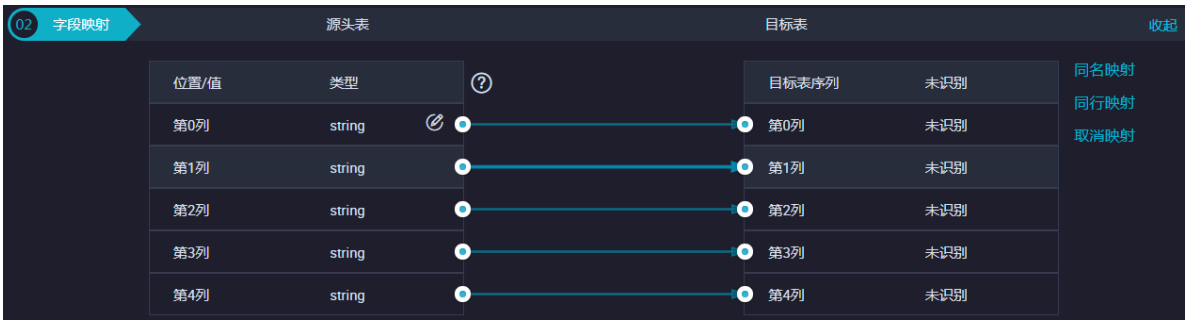
   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
文件路径	即上述参数说明中的path。
文本类型	读取的文件类型，默认情况下文件作为csv格式文件进行读取。
列分隔符	即上述参数说明中的fieldDelimiter，默认值为（,）。
编码格式	即上述参数说明中的encoding，默认值为utf-8。
null值	即上述参数说明中的nullFormat，定义表示null值的字符串。

配置	说明
压缩格式	即上述参数说明中的compress，默认值为不压缩。
是否包含表头	即上述参数说明中的skipHeader，默认值为No。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，将鼠标放至需要删除的字段上，即可单击删除按钮进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。

配置	说明
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

配置一个从FTP数据库同步抽取数据作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        {
            "stepType":"ftp",//插件名。
            "parameter":{
                "path":[],//文件路径。
                "nullFormat":"",//null值。
                "compress":"",//压缩格式。
                "datasource":"",//数据源。
                "column":[//字段。
                    {
                        "index":0,//序列号。
                        "type":""//字段类型。
                    }
                ],
                "skipHeader":"",//是否包含表头。
                "fieldDelimiter":",",//列分隔符。
                "encoding":"UTF-8",//编码格式。
                "fileFormat":"csv"//文本类型。
            },
            "name":"Reader",
            "category":"reader"
        },
        {//Reader的模板如下所示，您可以查看相应数据源的读插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
```

```
}
```

2.3.1.13 配置Table Store（OTS） Reader

本文为您介绍OTS Reader支持的数据类型、读取方式、字段映射和数据源等参数及配置举例。

OTS Reader插件实现了从Table Store（OTS）读取数据，通过您指定抽取数据范围可方便地实现数据增量抽取的需求。目前支持以下三种抽取方式。

- 全表抽取
- 范围抽取
- 指定分片抽取

Table Store是构建在阿里云飞天分布式系统之上的NoSQL数据库服务，提供海量结构化数据的存储和实时访问。Table Store以实例和表的形式组织数据，通过数据分片和负载均衡技术，实现规模上的无缝扩展。

简而言之，OTS Reader通过Table Store官方Java SDK连接到Table Store服务端，获取并按照数据同步官方协议标准转为数据同步字段信息传递给下游Writer端。

OTS Reader会根据Table Store的表范围，按照数据同步并发的数目N，将范围等分为N份Task。每个Task都会有一个OTS Reader线程来执行。

目前OTS Reader支持所有Table Store类型，OTS Reader针对Table Store的类型转换表，如下所示。

类型分类	MySQL数据类型
整数类	Integer
浮点类	Double
字符串类	String
布尔型	Boolean
二进制类	Binary

 说明:

Table Store本身不支持日期型类型。应用层一般使用Long报错时间的Unix TimeStamp。

参数说明

参数	描述	必选	默认值
endpoint	OTS Server的EndPoint（服务地址），详情请参见服务地址。	是	无
accessId	Table Store的accessId。	是	无

参数	描述	必选	默认值
accessKey	Table Store的accessKey。	是	无
instanceName	Table Store的实例名称，实例是您使用和管理Table Store服务的实体。 您在开通Table Store服务后，需要通过管理控制台来创建实例，然后在实例内进行表的创建和管理。 实例是Table Store资源管理的基础单元，Table Store对应用程序的访问控制和资源计量都在实例级别完成。	是	无
table	所选取的需要抽取的表名称，这里有且只能填写一张表。在Table Store不存在多表同步的需求。	是	无
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息。由于Table Store本身是NoSQL系统，在OTS Reader抽取数据过程中，必须指定相应的字段名称。 · 支持普通的列读取，例如{"name":"col1"} · 支持部分列读取，如果您不配置该列，则OTS Reader不予读取。 · 支持常量列读取，例如{"type":"STRING", "value":"DataX"}。使用type描述常量类型，目前支持String、Int、Double、Bool、Binary（使用Base64编码填写）、INF_MIN（Table Store的系统限定最小值，如果使用该值，您不能填写value属性，否则报错）、INF_MAX（Table Store的系统限定最大值，如果使用该值，您不能填写value属性，否则报错）。 · 不支持函数或者自定义表达式，由于Table Store本身不提供类似SQL的函数或者表达式功能，OTS Reader也不能提供函数或表达式列功能。	是	无

参数	描述	必选	默认值

begin/end	该配置项必须配对使用，用于支持Table Store表范围抽取。begin/end中描述的是OTS PrimaryKey的区间分布状态，而且必须保证区间覆盖到所有的 PrimaryKey，需要指定该表下所有的PrimaryKey范围，对于无限大小的区间，可以使用{"type":"INF_MIN"}, {"type":"INF_MAX"}指代。例如对一张主键为[DeviceID, SellerID]的Table Store进行抽取任务，begin/end的配置如下所示。
	`"range": { "begin": [ {"type":"INF_MIN"}, //指定 deviceID 最小值 {"type":"INT", "value":"0"} //指定 SellerID 最小值 ], "end": [ {"type":"INF_MAX"}, //指定 deviceID 抽取最大值 {"type":"INT", "value":"9999"} //指定 SellerID 抽取最大值 ] }`
	如果要对上述表抽取全表，可以使用如下配置。
	`"range": { "begin": [ {"type":"INF_MIN"}, //指定 deviceID 最小值 {"type":"INF_MIN"} //指定 SellerID 最小值 ], "end": [ {"type":"INF_MAX"}, //指定 deviceID 抽取最大值 {"type":"INF_MAX"} //指定 SellerID 抽取最大值 ] }`
		是	空

split	该配置项属于高级配置项，是您自己定义切分配置信息，普通情况下不建议使用。 适用场景：通常在Table Store数据存储发生热点，使用OTS Reader自动切分的策略不能生效的情况下，使用您自定义的切分规则。 split指定在Begin、End区间内的切分点，且只能是partitionKey的切分点信息，即在split仅配置partitionKey，而不需要指定全部的PrimaryKey。 如果对一张主键为[DeviceID, SellerID]的Table Store进行抽取任务，配置如下： `"range": {`	否	无

脚本开发介绍

配置一个从Table Store同步抽取数据到本地的作业。

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[
        {
            "stepType":"ots",//插件名
            "parameter":{
                "datasource":"",//数据源
                "column":[//字段
                    {
                        "name":"column1"//字段名
                    },
                    {
                        "name":"column2"
                    },
                    {
                        "name":"column3"
                    },
                    {
                        "name":"column4"
                    },
                    {
                        "name":"column5"
                    }
                ],
                "range":{
                    "split":[
                        {
                            "type":"INF_MIN"
                        },
                        {
                            "type":"STRING",
                            "value":"splitPoint1"
                        },
                        {
                            "type":"STRING",
                            "value":"splitPoint2"
                        },
                        {
                            "type":"STRING",
                            "value":"splitPoint3"
                        },
                        {
                            "type":"INF_MAX"
                        }
                    ],
                    "end":[
                        {
                            "type":"INF_MAX"
                        },
                        {
                            "type":"INF_MAX"
                        },
                        {
                            "type":"STRING",
                            "value":"end1"
                        },
                        {
                            "type":"INT",
```

```
                            "value":"100"
                        }
                    ],
                    "begin":[
                        {
                            "type":"INF_MIN"
                        },
                        {
                            "type":"INF_MIN"
                        },
                        {
                            "type":"STRING",
                            "value":"begin1"
                        },
                        {
                            "type":"INT",
                            "value":"0"
                        }
                    ]
                },
                "table":""//表名
            },
            "name":"Reader",
            "category":"reader"
        },
        { //下面是关于Writer的模板，可以找相应的写插件文档
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流
            "concurrent":1,//作业并发数
            "dmu":1//DMU值
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
```

```
}
```

2.3.1.14 配置PostgreSQL Reader

本文将为您介绍PostgreSQL Reader支持的数据类型、读取方式、字段映射和数据源等参数及配置示例。

PostgreSQL Reader插件从PostgreSQL读取数据。在底层实现上，PostgreSQL Reader通过JDBC连接远程PostgreSQL数据库，并执行相应的SQL语句，从PostgreSQL库中选取数据。RDS提供PostgreSQL存储引擎。

PostgreSQL Reader通过JDBC连接器连接至远程的PostgreSQL数据库，根据您配置的信息生成查询SQL语句，发送至远程PostgreSQL数据库，执行该SQL并返回结果。然后使用数据同步自定义的数据类型拼装为抽象的数据集，传递给下游Writer处理。

- 对于您配置的`table`、`column`和`where`等信息，PostgreSQL Reader将其拼接为SQL语句发送至PostgreSQL数据库。
- 对于您配置的`querySql`信息，PostgreSQL直接将其发送至PostgreSQL数据库。

类型转换列表

PostgreSQL Reader支持大部分PostgreSQL类型，但也存在部分类型没有支持的情况，请注意检查您的数据类型。

PostgreSQL Reader针对PostgreSQL的类型转换列表，如下所示。

类型分类	PostgreSQL数据类型
整数类	BIGINT、BIGSERIAL、INTEGER、SMALLINT和SERIAL
浮点类	DOUBLE、PRECISION、MONEY、NUMERIC和REAL
字符串类	VARCHAR、CHAR、TEXT、BIT和INET
日期时间类	DATE、TIME和TIMESTAMP
布尔型	BOOL
二进制类	BYTEA

 说明:

- 除上述罗列字段类型外，其他类型均不支持。
- MONEY、INET和BIT需要您使用`a_inet::varchar`类似的语法进行转换。

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息 。默认使用所有列配置，例如[*]。 · 支持列裁剪，即列可以挑选部分列进行导出。 · 支持列换序，即列可以不按照表schema信息顺序进行导出。 · 支持常量配置，您需要按照MySQL SQL语法格式，例如["id", "table","1", "'mingya.wmy'", "'null'", "to_char(a+1)", "2.3" , "true"] 。 - id为普通列名。 - table为包含保留字的列名。 - 1为整形数字常量。 - 'mingya.wmy'为字符串常量（注意需要加上一对单引号）。 - 'null'为字符串。 - to_char(a+1)为计算字符串长度函数。 - 2.3为浮点数。 - true为布尔值。 · column必须显示指定同步的列集合，不允许为空。	是	无
splitPk	PostgreSQL Reader进行数据抽取时，如果指定splitPk，表示您希望使用splitPk代表的字段进行数据分片，数据同步因此会启动并发任务进行数据同步，这样可以提高数据同步的效能。 · 推荐splitPk用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 · 目前splitPk仅支持整型数据切分，不支持字符串、浮点、日期等其他类型 。如果您指定其他非支持类型，忽略plitPk功能，使用单通道进行同步。 · 如果splitPk不填写，包括不提供splitPk或者splitPk值为空，数据同步视作使用单通道同步该表数据 。	否	无

参数	描述	是否必选	默认值
where	筛选条件，PostgreSQL Reader根据指定的column、table和where条件拼接SQL，并根据该SQL进行数据抽取。例如在测试时，可以将where条件指定实际业务场景，往往会选择当天的数据进行同步，将where条件指定为 `id>2 and sex=1`。 · where条件可以有效地进行业务增量同步。 · where条件不配置或者为空，视作全表同步数据。	否	无
querySql（高级模式，向导模式不提供）	在部分业务场景中，where配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选SQL。当配置此项后，数据同步系统就会忽略tables、columns和splitPk配置项，直接使用这项配置的内容对数据进行筛选，例如需要进行多表join后同步数据，使用 `select a,b from table_a join table_b on table_a.id = table_b.id`。当您配置querySql时，PostgreSQL Reader直接忽略table、column和where条件的配置。	否	无
fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据集成和服务器端的网络交互次数，能够较大的提升数据抽取性能。 说明: fetchSize值过大（>2048）可能造成数据同步进程OOM。	否	512

向导开发介绍

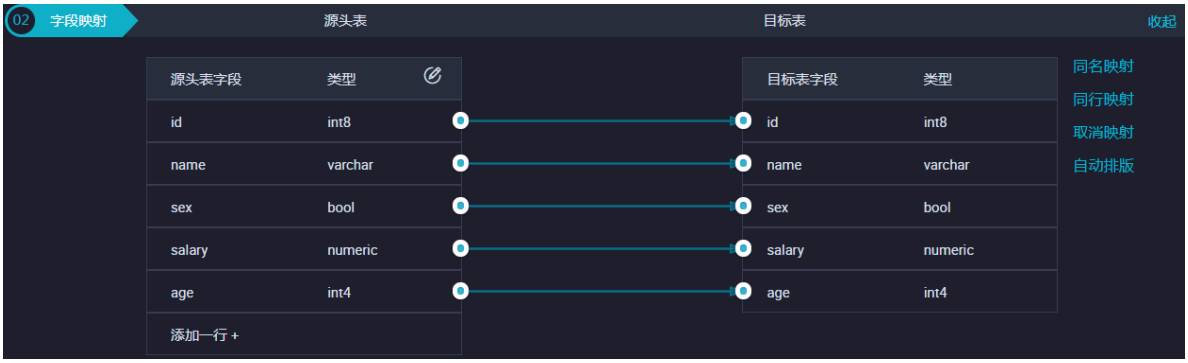
1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的`datasource`，通常填写您配置的数据源名称。
表	即上述参数说明中的`table`，选择需要同步的表。
数据过滤	您将要同步数据的筛选条件，暂时不支持limit关键字过滤。SQL语法与选择的数据源一致。
切分键	您可以将源数据表中某一列作为切分键，建议使用主键或有索引的列作为切分键，仅支持类型为整型的字段。 读取数据时，根据配置的字段进行数据分片，实现并发读取，可以提升数据同步效率。 **说明:** 切分键与数据同步中的选择来源有关，配置数据来源时才显示切分键配置项。

2. 字段映射，即上述参数说明中的column。

左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如’abc’、’123’等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

配置一个从PostgreSQL数据库同步抽取数据作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        {
            "stepType":"postgresql",//插件名。
            "parameter":{
                "datasource":"",//数据源。
                "column":[//字段。
                    "col1",
                    "col2"
                ],
                "where":"",//筛选条件
                "splitPk":"",//用splitPk代表的字段进行数据分片，数据同步会启动并发任务进行数据同步。
                "table":""//表名。
            },
            "name":"Reader",
            "category":"reader"
        },
        { //下面是关于writer的模板，您可以查找相应的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
```

```
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

补充说明

- **主备同步数据恢复问题**

  主备同步问题指PostgreSQL使用主从灾备，备库从主库不间断通过binlog恢复数据。由于主备数据同步存在一定的时间差，特别在于某些特定情况，例如网络延迟等问题，导致备库同步恢复的数据与主库有较大差别，从备库同步的数据不是一份当前时间的完整镜像。

- **一致性约束**

  PostgreSQL在数据存储划分中属于RDBMS系统，对外可以提供强一致性数据查询接口。例如一次同步任务启动运行过程中，当该库存在其他数据写入方写入数据时，由于数据库本身的快照特性，PostgreSQL Reader完全不会获取到写入的更新数据。

  上述是在PostgreSQL Reader单线程模型下数据同步一致性的特性，PostgreSQL Reader可以根据您配置的信息使用并发数据抽取，因此不能严格保证数据一致性。

  当PostgreSQL Reader根据splitPk进行数据切分后，会先后启动多个并发任务完成数据同步。多个并发任务相互之间不属于同一个读事务，同时多个并发任务存在时间间隔，因此这份数据并不是完整的、一致的数据快照信息。

  针对多线程的一致性快照需求，目前在技术上无法实现，只能从工程角度解决。工程化的方式存在取舍，在此提供以下解决思路，您可以根据自身情况进行选择。

  - **使用单线程同步，即不再进行数据切片。缺点是速度比较慢，但是能够很好保证一致性。**
  - **关闭其他数据写入方，保证当前数据为静态数据，例如锁表、关闭备库同步等。缺点是可能影响在线业务。**

- 数据库编码问题

  PostgreSQL在服务器端仅支持EUC_CN和UTF-8两种简体中文编码，PostgreSQL Reader底层使用JDBC进行数据抽取，JDBC天然适配各类编码，并在底层进行了编码转换。因此PostgreSQL Reader不需您指定编码，可以自动获取编码并转码。

  对于PostgreSQL底层写入编码和其设定的编码不一致的混乱情况，PostgreSQL Reader对此无法识别，也无法提供解决方案，导出结果有可能为乱码。

- 增量数据同步

  PostgreSQL Reader使用JDBC SELECT语句完成数据抽取工作，因此可以使用`SELECT…WHERE…`进行增量数据抽取，有以下几种方式：

  - 数据库在线应用写入数据库时，填充modify字段为更改时间戳，包括新增、更新、删除（逻辑删除）。对于该类应用，PostgreSQL Reader只需要where条件后跟上一同步阶段时间戳即可。
  - 对于新增流水型数据，PostgreSQL Reader在where条件后跟上一阶段最大自增ID即可。

  对于业务上无字段区分新增、修改数据的情况，PostgreSQL Reader无法进行增量数据同步，只能同步全量数据。

- SQL安全性

  PostgreSQL Reader提供querySql语句交给您自己实现SELECT抽取语句，PostgreSQL Reader本身对querySql不进行任何安全性校验。

2.3.1.15 配置SQL Server Reader

本文将为您介绍SQL Server Reader支持的数据类型、字段映射和数据源等参数及配置示例。

SQL Server Reader插件从SQL Server读取数据。在底层实现上，SQL Server Reader通过JDBC连接远程SQL Server数据库，并执行相应的SQL语句，从SQL Server库中读取数据。

SQL Server Reader通过JDBC连接器连接至远程的SQL Server数据库，根据您配置的信息生成查询SQL语句，发送至远程SQL Server数据库，执行该SQL并返回结果。然后使用数据同步自定义的数据类型拼装为抽象的数据集，传递给下游Writer处理。

- 对于您配置的`table`、`column`和`where`等信息，SQL Server Reader将其拼接为SQL语句发送至SQL Server数据库。
- 对于您配置的`querySql`信息，SQL Server直接将其发送至SQL Server数据库。

SQL Server Reader支持大部分SQL Server类型，但也存在部分类型没有支持的情况，请注意检查您的数据类型。

SQL Server Reader针对SQL Server的类型转换列表，如下所示。

类型分类	SQL Server数据类型
整数类	BIGINT、INT、SMALLINT和TINYINT
浮点类	FLOAT、DECIMAL、REAL和NUMERIC
字符串类	CHAR、NCHAR、NTEXT、NVARCHAR、TEXT、VARCHAR、NVARCHAR（MAX）和VARCHAR（MAX）
日期时间类	DATE、DATETIME和TIME
布尔型	BIT
二进制类	BINARY、VARBINARY、VARBINARY（MAX）和TIMESTAMP

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称，一个作业只能支持一个表同步。	是	无
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息 。默认使用所有列配置，例如[*]。 · 支持列裁剪，即列可以挑选部分列进行导出。 · 支持列换序，即列可以不按照表schema信息顺序进行导出。 · 支持常量配置，您需要按照MySQL SQL语法格式，例如`["id", "table","1", "'mingya.wmy'", "'null'", "to_char(a+1)", "2.3" , "true"]` 。 - id为普通列名。 - table为包含保留字的列名。 - 1为整型数字常量。 - 'mingya.wmy'为字符串常量（注意需要加上一对单引号）。 - 'null'为字符串。 - to_char(a + 1)为函数表达式。 - 2.3为浮点数。 - true为布尔值。 · column必须显示指定同步的列集合，不允许为空。	是	无

参数	描述	必选	默认值
splitPk	SQL Server Reader进行数据抽取时，如果指定splitPk，表示您希望使用splitPk代表的字段进行数据分片。数据同步系统因此会启动并发任务进行数据同步，这样可以提高数据同步的效能。 · 推荐splitPk用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 · 目前splitPk仅支持整型数据切分，不支持字符串、浮点、日期等其他类型 。如果您指定其他非支持类型，SQL Server Reader将报错。	否	无
where	筛选条件，SQL Server Reader根据指定的column、table和where条件拼接SQL，并根据该SQL进行数据抽取。例如在测试时，可以将where条件指定为limit 10。在实际业务场景中，往往会选择当天的数据进行同步，将where条件指定为gmt_create > $bizdate。 · where条件可以有效地进行业务增量同步。 · where条件为空，视作同步全表所有的信息。	否	无
querySql	使用格式："querysql" : "查询statement",在部分业务场景中，where配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选SQL。当配置此项后，数据同步系统就会忽略tables、columns配置项，直接使用这项配置的内容对数据进行筛选，例如需要进行多表join后同步数据，使用select a,b from table_a join table_b on table_a.id = table_b.id。当您配置querySql时，SQL Server Reader直接忽略column、table和where条件的配置。	否	无
fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据集成和服务器端的网络交互次数，能够提升数据抽取性能。 说明: fetchSize值过大（>2048）可能造成数据同步进程OOM。	否	1024

向导开发介绍

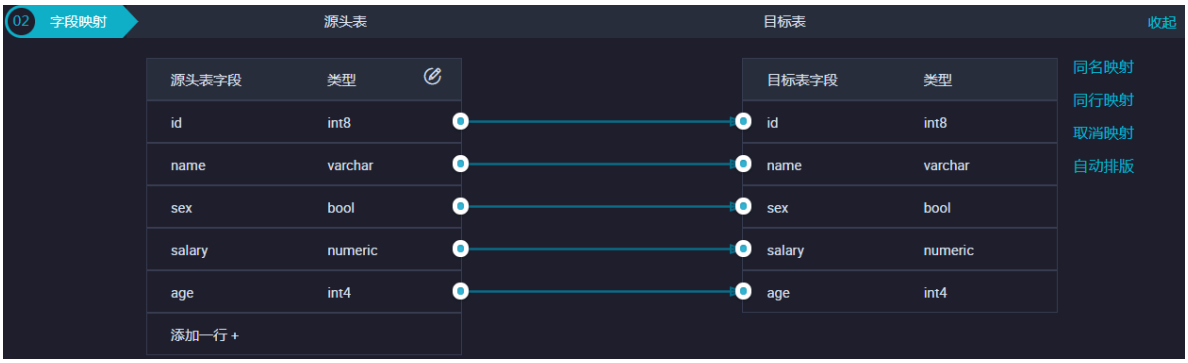
1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table，选择需要同步的表。
数据过滤	您将要同步数据的筛选条件，暂时不支持limit关键字过滤。SQL语法与选择的数据源一致。
切分键	您可以将源数据表中某一列作为切分键，建议使用主键或有索引的列作为切分键。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。

配置	说明
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如’abc’、’123’等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

配置一个从SQL Server数据库同步抽取数据的作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
```

```
    "steps":[
        {
            "stepType":"sqlserver",//插件名。
            "parameter":{
                "datasource":"",//数据源。
                "column":[//字段。
                    "id",
                    "name"
                ],
                "where":"",//筛选条件。
                "splitPk":"",//如果指定splitPk，表示您希望使用splitPk代表的
字段进行数据分片。
                "table":""//数据表。
            },
            "name":"Reader",
            "category":"reader"
        },
        {//下面是关于Writer的模板，您可以查找相应的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

如果您想使用querySql查询，Reader部分脚本代码示例如下（SQL Server数据源是sql_server_source，待查询的表是dbo.test_table，待查询的列是name）。

```
{
    "stepType": "sqlserver",
    "parameter": {
        "querySql": "select name from dbo.test_table",
        "datasource": "sql_server_source",
        "column": [
            "name"
        ],
        "where": "",
        "splitPk": "id"
    },
    "name": "Reader",
    "category": "reader"
```

```
},
```

补充说明

- **主备同步数据恢复问题**

  主备同步问题指SQL Server使用主从灾备，备库从主库不间断通过binlog恢复数据。由于主备数据同步存在一定的时间差，特别在于某些特定情况，例如网络延迟等问题，导致备库同步恢复的数据与主库有较大差别，从备库同步的数据不是一份当前时间的完整镜像。

- **一致性约束**

  SQL Server在数据存储划分中属于RDBMS系统，对外可以提供强一致性数据查询接口。例如一次同步任务启动运行过程中，当该库存在其他数据写入方写入数据时，由于数据库本身的快照特性，SQL Server Reader完全不会获取到写入的更新数据。

  上述是在SQL Server Reader单线程模型下数据同步一致性的特性，SQL Server Reader可以根据您配置的信息使用并发数据抽取，因此不能严格保证数据一致性。

  当SQL Server Reader根据splitPk进行数据切分后，会先后启动多个并发任务完成数据同步。多个并发任务相互之间不属于同一个读事务，同时多个并发任务存在时间间隔，因此这份数据并不是完整的、一致的数据快照信息。

  针对多线程的一致性快照需求，目前在技术上无法实现，只能从工程角度解决。工程化的方式存在取舍，在此提供以下解决思路，您可以根据自身情况进行选择。

  - **使用单线程同步，即不再进行数据切片。缺点是速度比较慢，但是能够很好保证一致性。**
  - **关闭其他数据写入方，保证当前数据为静态数据，例如锁表、关闭备库同步等。缺点是可能影响在线业务。**

- **数据库编码问题**

  SQL Server Reader底层使用JDBC进行数据抽取，JDBC天然适配各类编码，并在底层进行了编码转换。因此SQL Server Reader不需您指定编码，可以自动获取编码并转码。

- 增量数据同步

  SQL Server Reader使用JDBC SELECT语句完成数据抽取工作，因此可以使用`SELECT…WHERE…`进行增量数据抽取，有以下几种方式：

  - 数据库在线应用写入数据库时，填充modify字段为更改时间戳，包括新增、更新、删除（逻辑删除）。对于该类应用，SQL Server Reader只需要where条件后跟上一同步阶段时间戳即可。
  - 对于新增流水型数据，SQL Server Reader在where条件后跟上一阶段最大自增ID即可。

  对于业务上无字段区分新增、修改数据的情况，SQL Server Reader无法进行增量数据同步，只能同步全量数据。

- SQL安全性

  SQL Server Reader提供querySql语句交给您自己实现SELECT抽取语句，SQL Server Reader本身对querySql不进行任何安全性校验。

2.3.1.16 配置LogHub Reader

本文将为您介绍LogHub Reader支持的数据类型、字段映射和数据源等参数及配置示例。

日志服务（Log Service）是针对实时数据的一站式服务，为您提供日志类数据采集、消费、投递及查询分析功能，全面提升海量日志处理/分析能力。LogHub Reader是使用日志服务的Java SDK消费LogHub中的实时日志数据，并将日志数据转换为数据集成传输协议传递给Writer。

实现原理

LogHub Reader通过日志服务Java SDK消费LogHub中的实时日志数据，具体使用的日志服务Java SDK版本，如下所示。

```
<dependency>
    <groupId>com.aliyun.openservices</groupId>
    <artifactId>aliyun-log</artifactId>
    <version>0.6.7</version>
</dependency>
```

日志库（Logstore）是日志服务中日志数据的采集、存储和查询单元，Logstore读写日志必定保存在某一个分区（Shard）上。每个日志库分若干个分区，每个分区由MD5左闭右开区间组成，每个区间范围不会相互覆盖，并且所有的区间的范围是MD5整个取值范围，每个分区可提供一定的服务能力。

- 写入：5MB/s，2000次/s。
- 读取：10MB/s，100次/s。

LogHub Reader消费Shard中的日志，具体消费过程（GetCursor、BatchGetLog相关API）如下所示。

- 根据时间区间范围获得游标。
- 通过游标、步长参数读取日志，同时返回下一个位置游标。
- 不断移动游标进行日志消费。
- 根据Shard进行任务的切分并发执行。

LogHub Reader针对LogHub类型的转换列表，如下所示。

DataX内部类型	LogHub数据类型
STRING	STRING

参数说明

参数	描述	是否必选	默认值
endpoint	日志服务入口endpoint是访问一个项目（Project）及其内部日志数据的URL。它和Project所在的阿里云区域（Region）及Project名称相关。各Region的服务入口请参见#unique_177。	是	无
accessId	访问日志服务的访问秘匙，用于标识用户。	是	无
accessKey	访问日志服务的访问秘匙，用来验证用户的密钥。	是	无
project	目标日志服务的项目名字，是日志服务中的资源管理单元，用于资源隔离和控制。	是	无
logstore	目标日志库的名字，logstore是日志服务中日志数据的采集、存储和查询单元。	是	无
batchSize	一次从日志服务查询的数据条数。	否	128
column	每条数据中column的名字，这里可以配置日志服务中的元数据作为同步列，支持的元数据有 日志主题、采集机器唯一标识、主机名、路径和日志时间等。 说明: 列名区分大小写。元数据写法请参见日志服务机器组#unique_178。	是	无

参数	描述	是否必选	默认值
beginDateTime	数据消费的开始时间位点，即日志数据到达Loghub的时间。该参数为时间范围（左闭右开）的左边界，yyyyMMddHHmmss格式的时间字符串（例如20180111013000），可以和DataWorks的调度时间参数配合使用。  说明: beginDateTime和endDateTime需互相组合配套使用。	和beginTimestampMillis选择一种	无
endDateTime	数据消费的结束时间位点，为时间范围（左闭右开）的右边界，yyyyMMddHHmmss格式的时间字符串（例如20180111013010），可以和DataWorks的调度时间参数配合使用。  说明: 请尽量保证周期之间重合：即上周期的endDateTime时间和下周期的beginDateTime时间一致，或比下周期的beginDateTime时间晚。否则，可能造成部分区域数据无法拉取。	和endTimestampMillis选择一种	无
beginTimestampMillis	数据消费的开始时间位点。该参数为时间范围（左闭右开）的左边界，单位毫秒。  说明: beginTimestampMillis和endTimestampMillis组合配套使用。 -1表示日志服务游标的最开始CursorMode.BEGIN。推荐使用beginDateTime模式。	和beginDateTime选择一种	无
endTimestampMillis	数据消费的结束时间位点，为时间范围（左闭右开）的右边界，单位毫秒。  说明: endTimestampMillis和beginTimestampMillis组合配套使用。 -1表示日志服务游标的最后位置CursorMode.END。推荐使用endDateTime模式。	和endDateTime选择一种	无

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
Logstore	目标日志库的名称。
日志开始时间	数据消费的开始时间位点，即日志数据到达Loghub的时间。时间范围（左闭右开）的左边界，yyyyMMddHHmmss格式的时间字符串（例如20180111013000），可以和DataWorks的调度时间参数配合使用。
日志结束时间	数据消费的结束时间位点，时间范围（左闭右开）的右边界，yyyyMMddHHmmss格式的时间字符串（例如20180111013010），可以和DataWorks的调度时间参数配合使用。
批量条数	一次从日志服务查询的数据条数。

2. 字段映射，即上述参数说明中的column。

左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。

配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如’abc’、’123’等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

脚本配置样例如下所示，具体参数填写请参见参数说明。

```
{
 "type":"job",
 "version":"2.0",//版本号。
 "steps":[
     {
         "stepType":"loghub",//插件名。
         "parameter":{
             "datasource":"",//数据源。
             "column":[//字段。
                 "col0",
                 "col1",
                 "col2",
                 "col3",
                 "col4",
                 "=Topic",//日志主题。
                 "HostName",//主机名。
                 "Path",//路径。
                 "LogTime"//日志时间。
             ],
             "beginDateTime":"",//数据消费的开始时间位点。
             "batchSize":"",//一次从日志服务查询的数据条数。
             "endDateTime":"",//数据消费的结束时间位点。
             "fieldDelimiter":",",//列分隔符。
             "encoding":"UTF-8",//编码格式。
             "logstore":""//: 目标日志库的名字。
         },
```

```
            "name":"Reader",
            "category":"reader"
        },
        { //下面是关于Writer的模板，可以查找相应的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true代表
限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

2.3.1.17 配置OTSReader-Internal

本文将为您介绍OTSReader-Internal支持的数据类型、字段映射和数据源等参数及配置示例。

表格存储（Table Store，简称OTS）是构建在阿里云飞天分布式系统之上的NoSQL数据库服务，提供海量结构化数据的存储和实时访问。Table Store以实例和表的形式组成数据，通过数据分片和负载均衡技术，实现规模上的无缝扩展。

OTSReader-Internal主要用于OTS Internal模型的表数据导出，而另外一个插件OTS Reader则用于OTS Public模型的数据导出。

OTS Internal模型支持多版本，所以该插件提供两种模式数据的导出：

- 多版本模式：因为Table Store本身支持多版本，特此提供一个多版本模式，将多版本的数据导出。

  导出方案：Reader插件将Table Store的一个Cell展开为一个一维表的4元组，分别是主键（PrimaryKey，包含1-4列）、ColumnName、Timestamp和Value（原理和HBase Reader的多版本模式类似），将这个4元组作为Datax record中的4个Column传输给消费端（Writer）。
- 普通模式：和HBase Reader普通模式一致，只需导出每行数据中每列的最新版本的值，详情请参见#unique_180中HBase Reader支持的normal模式内容。

OTS Reader通过Table Store官方Java SDK连接至OTS服务端，并通过SDK读取数据。OTS Reader本身对读取过程做了很多优化，包括读取超时重试、异常读取重试等。

目前OTS Reader支持所有Table Store类型，OTSReader-Internal针对Table Store类型的转换列表，如下所示。

数据集成内部类型	Table Store数据类型
LONG	INTEGER
DOUBLE	DOUBLE
STRING	STRING
BOOLEAN	BOOLEAN
BYTES	BINARY

参数说明

参数	描述	必选	默认值
mode	插件的运行方式，支持normal和multiVersion，分别表示普通模式和多版本模式。	是	无
endpoint	OTS Server的endpoint（服务地址）。	是	无
accessId	Table Store的accessId。	是	无
accessKey	Table Store的accessKey。	是	无
instanceName	Table Store的实例名称，实例是您使用和管理Table Store服务的实体。 您在开通Table Store服务后，需要通过管理控制台来创建实例，然后在实例内进行表的创建和管理。实例是Table Store资源管理的基础单元，Table Store对应用程序的访问控制和资源计量都在实例级别完成。	是	无
table	选取的需要抽取的表名称，这里有且只能填写一张表。在Table Store中不存在多表同步的需求。	是	无

参数	描述	必选	默认值
range	导出的范围，读取的范围是[begin,end)，左闭右开的区间。 · begin小于end，表示正序读取数据。 · begin大于end，表示反序读取数据。 · begin和end不能相等。 · type支持的类型有string、int和binary，binary输入的方式采用二进制的Base64字符串形式传入，INF_MIN表示无限小，INF_MAX表示无限大。	否	从表的开始位置读取到表的结束位置
range:{"begin"}	导出的起始范围，这个值的输入可以填写空数组或PK前缀，也可以填写完整的PK。正序读取数据时，默认填充PK后缀为INF_MIN，反序为INF_MAX。 该配置是OTS主键的值范围，用于进行数据过滤。 如果没有配置开始的值，则默认最小值。 binary类型的PrimaryKey列比较特殊，因为JSON不支持直接输入二进制数，所以系统定义：如果您要传入二进制，必须使用（Java）Base64.encodeBase64String方法，将二进制转换为一个可视化的字符串，然后将这个字符串填入value中，Java示例如下。 · `byte[] bytes = “hello”.getBytes();`：构造一个二进制数据，这里使用字符串hello的byte值。 · `String inputValue = Base64.encodeBase64String(bytes)`：调用Base64方法，将二进制转换为可视化的字符串。 上面的代码执行之后，可以获得inputValue为"aGVsbG8="。 最终写入配置`{“type”:”binary”,”value” : “aGVsbG8=”}`。	否	从表的开始位置读取数据

参数	描述	必选	默认值
range:{“end”}	导出的结束范围，这个值的输入可以填写空数组或PK前缀，也可以填写完整的 PK。正序读取数据时，默认填充PK后缀为INF_MAX，反序为INF_MIN。 binary类型的PrimaryKey列比较特殊，因为JSON不支持直接输入二进制数，所以系统定义：如果您要传入二进制，必须使用（Java）Base64.encodeBase64String方法，将二进制转换为一个可视化的字符串，然后将这个字符串填入value中，Java示例如下。 · byte[] bytes = “hello”.getBytes();：构造一个二进制数据，这里使用字符串hello的byte值。 · String inputValue = Base64.encodeBase64String(bytes)：调用Base64方法，将二进制转换为可视化的字符串。 上面的代码执行之后，可以获得inputValue为"aGVsbG8="。 最终写入配置{"type":"binary", "value":"aGVsbG8="}。	否	读取到表的结束位置
range:{"split"}	当前用户数据较多时，需要开启并发导出，Split可以将当前范围的的数据按照切分点切分为多个并发任务。 说明: · split中的输入值只能PK的第一列（分片建），且值的类型必须和PartitionKey一致。 · 值的范围必须在begin和end之间。 · split内部的值必须根据begin和end的正反序关系而递增或者递减。	否	空切分点
column	指定要导出的列，支持普通列和常量列。 格式（支持多版本模式） 普通列格式：{“name”:”{your column name}”}	是	无
timeRange（仅支持多版本模式）	请求数据的Time Range，读取的范围为[begin,end)，左闭右开的区间。 说明: begin必须小于end。	否	默认读取全部版本

参数	描述	必选	默认值
timeRange:{"begin"}（仅支持多版本模式）	请求数据的Time Range开始时间，取值范围是0~LONG_MAX。	否	默认为0
timeRange:{"end"}（仅支持多版本模式）	请求数据的Time Range结束时间，取值范围是0~LONG_MAX。	否	默认为Long Max(9223372036854775806L)
maxVersion（仅支持多版本模式）	请求的指定Version，取值范围是1~INT32_MAX。	否	默认读取所有版本

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

- 多版本模式

```
{
    "type": "job",
    "version": "1.0",
    "configuration": {
        "reader": {
            "plugin": "otsreader-internalreader",
            "parameter": {
                "mode": "multiVersion",
                "endpoint": "",
                "accessId": "",
                "accessKey": "",
                "instanceName": "",
                "table": "",
                "range": {
                    "begin": [
                        {
                            "type": "string",
                            "value": "a"
                        },
                        {
                            "type": "INF_MIN"
                        }
                    ],
                    "end": [
                        {
                            "type": "string",
                            "value": "g"
                        },
```

```
                        {
                            "type": "INF_MAX"
                        }
                    ],
                    "split": [
                        {
                            "type": "string",
                            "value": "b"
                        },
                        {
                            "type": "string",
                            "value": "c"
                        }
                    ]
                },
                "column": [
                    {
                        "name": "attr1"
                    }
                ],
                "timeRange": {
                    "begin": 1400000000,
                    "end": 1600000000
                },
                "maxVersion": 10
            }
        }
    },
    "writer": {}
}
```

- **普通模式**

```
{
    "type": "job",
    "version": "1.0",
    "configuration": {
        "reader": {
            "plugin": "otsreader-internalreader",
            "parameter": {
                "mode": "normal",
                "endpoint": "",
                "accessId": "",
                "accessKey": "",
                "instanceName": "",
                "table": "",
                "range": {
                    "begin": [
                        {
                            "type": "string",
                            "value": "a"
                        },
                        {
                            "type": "INF_MIN"
                        }
                    ],
                    "end": [
                        {
                            "type": "string",
                            "value": "g"
                        },
                        {
                            "type": "INF_MAX"
```

```
                        }
                    ],
                    "split": [
                        {
                            "type": "string",
                            "value": "b"
                        },
                        {
                            "type": "string",
                            "value": "c"
                        }
                    ]
                },
                "column": [
                    {
                        "name": "pk1"
                    },
                    {
                        "name": "pk2"
                    },
                    {
                        "name": "attr1"
                    },
                    {
                        "type": "string",
                        "value": ""
                    },
                    {
                        "type": "int",
                        "value": ""
                    },
                    {
                        "type": "double",
                        "value": ""
                    },
                    {
                        "type": "binary",
                        "value": "aGVsbG8="
                    }
                ]
            }
        }
    },
    "writer": {}
}
```

2.3.1.18 配置OTSStream Reader

本文为您介绍OTSStream Reader支持的数据类型、读取方式、字段映射和数据源等参数及配置举例。

OTSStream Reader插件主要用于Table Store增量数据的导出，增量数据可以看作操作日志，除数据本身外还附有操作信息。

与全量导出插件不同，增量导出插件只有多版本模式，且不支持指定列，这与增量导出的原理有关，导出格式的详细介绍请参见下文。

使用插件前必须确保表上已经开启Stream功能，您可以在建表的时候指定开启，也可以使用SDK的UpdateTable接口开启。

开启Stream的方法，如下所示。

```
SyncClient client = new SyncClient("", "", "", "");
建表的时候开启：
CreateTableRequest createTableRequest = new CreateTableRequest(
tableMeta);
createTableRequest.setStreamSpecification(new StreamSpecification(true
, 24)); // 24代表增量数据保留24小时
client.createTable(createTableRequest);
如果建表时未开启，可以通过UpdateTable开启：
UpdateTableRequest updateTableRequest = new UpdateTableRequest("
tableName");
updateTableRequest.setStreamSpecification(new StreamSpecification(true
, 24));
client.updateTable(updateTableRequest);
```

实现原理

您使用SDK的UpdateTable功能，指定开启Stream并设置过期时间，即开启了增量功能。开启后，Table Store服务端就会将您的操作日志额外保存起来，每个分区有一个有序的操作日志队列，每条操作日志会在一定时间后被垃圾回收，这个时间即为您指定的过期时间。

Table Store的SDK提供了几个Stream相关的API用于将这部分操作日志读取出来，增量插件也是通过Table Store SDK的接口获取到增量数据的，并将增量数据转化为多个6元组的形式（pk、colName、version、colValue、opType和sequenceInfo）导入到MaxCompute中。

导出的数据格式

在Table Store多版本模型下，表中的数据组织为行>列>版本三级的模式， 一行可以有任意列，列名也并非固定的，每一列可以含有多个版本，每个版本都有一个特定的时间戳（版本号）。

您可以通过Table Store的API进行一系列读写操作，Table Store通过记录您最近对表的一系列写操作（或数据更改操作）来实现记录增量数据的目的，所以您也可以把增量数据看作一批操作记录。

Table Store有PutRow、UpdateRow和DeleteRow三类数据更改操作。

- PutRow：写入一行，若该行已存在即覆盖该行。
- UpdateRow：更新一行，对原行其他数据不做更改， 更新可能包括新增或覆盖（若对应列的对应版本已存在）一些列值、删除某一列的全部版本、删除某一列的某个版本。
- DeleteRow：删除一行。

Table Store会根据每种操作生成对应的增量数据记录，Reader插件会读出这些记录，并导出为Datax的数据格式。

同时，由于Table Store具有动态列、多版本的特性，所以Reader插件导出的一行不对应Table Store中的一行，而是对应Table Store中的一列的一个版本。即Table Store中的一行可能会导出

很多行，每行包含主键值、该列的列名、该列下该版本的时间戳（版本号）、该版本的值、操作类型。如果设置isExportSequenceInfo为true，还会包括时序信息。

转换为Datax的数据格式后，定义了四种操作类型，如下所示。

- U（UPDATE）：写入一列的一个版本。
- DO（DELETE_ONE_VERSION）：删除某一列的某个版本。
- DA（DELETE_ALL_VERSION）：删除某一列的全部版本，此时需要根据主键和列名，将对应列的全部版本删除。
- DR（DELETE_ROW）：删除某一行，此时需要根据主键，将该行数据全部删除。

假设该表有两个主键列，主键列名分别为pkName1， pkName2，示例如下。

pkName1	pkName2	columnName	timestamp	columnValue	opType
pk1_V1	pk2_V1	col_a	1441803688001	col_val1	U
pk1_V1	pk2_V1	col_a	1441803688002	col_val2	U
pk1_V1	pk2_V1	col_b	1441803688003	col_val3	U
pk1_V2	pk2_V2	col_a	1441803688000	—	DO
pk1_V2	pk2_V2	col_b	—	—	DA
pk1_V3	pk2_V3	—	—	—	DR
pk1_V3	pk2_V3	col_a	1441803688005	col_val1	U

假设导出的数据如上，共7行，对应Table Store表内的3行，主键分别是（pk1_V1，pk2_V1），（pk1_V2， pk2_V2），（pk1_V3， pk2_V3）。

- 对于主键为（pk1_V1，pk2_V1）的一行，包含三个操作，分别是写入col_a列的两个版本和col_b列的一个版本。
- 对于主键为（pk1_V2，pk2_V2）的一行，包含两个操作，分别是删除col_a列的一个版本、删除col_b列的全部版本。
- 对于主键为（pk1_V3， pk2_V3）的一行，包含两个操作，分别是删除整行、写入col_a列的一个版本。

目前OTSStream Reader支持所有的Table Store类型，其针对Table Store类型的转换列表，如下所示。

类型分类	OTSStream数据类型
整数类	Integer
浮点类	Double
字符串类	String
布尔类	Boolean
二进制类	Binary

参数说明

参数	描述	必选	默认值
dataSource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
dataTable	导出增量数据的表的名称。该表需要开启Stream，可以在建表时开启，或者使用UpdateTable接口开启。	是	无
statusTable	Reader插件用于记录状态的表的名称，这些状态可用于减少对非目标范围内的数据的扫描，从而加快导出速度。statusTable是Reader用于保存状态的表，如果该表不存在，Reader会自动创建该表，一次离线导出任务完成后，您不需删除该表，该表中记录的状态可用于下次导出任务中。 · 您不需要创建该表，只需要给出一个表名。Reader插件会尝试在您的instance下创建该表，如果该表不存在即创建新表，如果该表已存在，会判断该表的Meta是否与期望一致，如果不一致会抛出异常。 · 在一次导出完成之后，您不需删除该表，该表的状态可用于下次导出任务。 · 该表会开启TTL，数据自动过期，因此可认为其数据量很小。 · 针对同一个instance下的多个不同的dataTable的Reader配置，可以使用同一个statusTable，记录的状态信息互不影响。 综上所述，您配置一个类似TableStoreStreamReaderStatusTable的名称即可，请注意不要与业务相关的表重名。	是	无

参数	描述	必选	默认值
startTimestampMillis	增量数据的时间范围（左闭右开）的左边界，单位毫秒。 · Reader插件会从statusTable中找对应startTimestampMillis的位点，从该点开始读取开始导出数据。 · 如果statusTable中找不到对应的位点，则从系统保留的增量数据的第一条开始读取，并跳过写入时间小于startTimestampMillis的数据。	否	无
endTimestampMillis	增量数据的时间范围（左闭右开）的右边界，单位毫秒。 · Reader插件从startTimestampMillis位置开始导出数据后，当遇到第一条时间戳大于等于endTimestampMillis的数据时，结束导出数据，导出完成。 · 当读取完当前全部的增量数据时，结束读取，即使未达到endTimestampMillis。	否	无
date	日期格式为yyyyMMdd，如20151111，表示导出该日的数据。如果没有指定date，则必须指定startTimestampMillis和endTimestampMillis，反之也成立。 例如采云间调度只支持天级别，所以提供该配置，作用与startTimestampMillis和endTimestampMillis类似。	否	无
isExportSequenceInfo	是否导出时序信息，时序信息包含了数据的写入时间等。默认该值为false，即不导出。	否	无
maxRetries	从TableStore中读增量数据时，每次请求的最大重试次数，默认为30，重试之间有间隔，重试30次的总时间约为5分钟，一般无需更改。	否	无
startTimeString	增量数据的时间范围（左闭右开）的左边界，格式为yyyymmddhh24miss，单位毫秒。	否	无
endTimeString	增量数据的时间范围（左闭右开）的右边界，格式为yyyymmddhh24miss，单位毫秒。	否	无

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

脚本配置样例如下所示，具体参数填写请参见参数说明。

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[
        {
            "stepType":"otsstream",//插件名
            "parameter":{
```

```
                    "statusTable":"TableStoreStreamReaderStatusTable",//用
于记录状态的表的名称
                    "maxRetries":30,//从 TableStore 中读增量数据时，每次请求的
最大重试次数，默认为 30
                    "isExportSequenceInfo":false,//是否导出时序信息
                    "datasource":"$srcDatasource",//数据源
                    "startTimeString":"${startTime}",//增量数据的时间范围（左
闭右开）的左边界
                    "table":"",//表名
                    "endTimeString":"${endTime}"//增量数据的时间范围（左闭右
开）的右边界
                },
                "name":"Reader",
                "category":"reader"
            },
            { //下面是关于Writer的模板，可以找相应的写插件文档
                "stepType":"stream",
                "parameter":{},
                "name":"Writer",
                "category":"writer"
            }
        ],
        "setting":{
            "errorLimit":{
                "record":"0"//错误记录数
            },
            "speed":{
                "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流
                "concurrent":1,//作业并发数
                "dmu":1//DMU值
            }
        },
        "order":{
            "hops":[
                {
                    "from":"Reader",
                    "to":"Writer"
                }
            ]
        }
}
```

2.3.1.19 配置RDBMS Reader

本文为您介绍RDBMS Reader支持的数据类型、字段映射和数据源等参数及配置示例。

SQL Server Reader插件从SQL Server读取数据。在底层实现上，SQL Server Reader通过JDBC连接远程SQL Server数据库，并执行相应的SQL语句，从SQL Server库中读取数据。

RDBMS Reader插件从RDBMS读取数据。在底层实现上，RDBMS Reader通过JDBC连接远程RDBMS数据库，并执行相应的SQL语句，从RDBMS库中读取数据。目前RDBMS Reader支持读取DM、DB2、PPAS和Sybase等数据库的数据。RDBMS Reader是一个通用的关系数据库读插件，您可以通过注册数据库驱动等方式，增加任意多样的关系数据库读支持。

RDBMS Reader通过JDBC连接器连接至远程的RDBMS数据库，并根据您配置的信息生成查询SQL语句，发送至远程RDBMS数据库，执行该SQL并返回结果。然后使用数据同步自定义的数据类型拼装为抽象的数据集，传递给下游Writer处理。

- 对于您配置的`table`、`column`和`where`等信息，RDBMS Reader将其拼接为SQL语句发送至RDBMS数据库。
- 对于您配置的`querySql`信息，RDBMS直接将其发送至RDBMS数据库。

RDBMS Reader支持大部分通用的关系数据库数据类型，例如数字、字符等。但也存在部分类型没有支持的情况，请注意检查您的数据类型，根据具体的数据库进行选择。

参数说明

参数	描述	必选	默认值
jdbcUrl	描述的是到对端数据库的JDBC连接信息，jdbcUrl按照RDBMS官方规范，并可以填写连接附件控制信息。请注意不同的数据库JDBC的格式是不同的，DataX会根据具体JDBC的格式选择合适的数据库驱动完成数据读取。 · DM格式：jdbc:dm://ip:port/database · DB2格式：jdbc:db2://ip:port/database · PPAS格式：jdbc:edb://ip:port/database RDBMS Writer可以通过以下方式增加新的数据库支持。 · 进入RDBMS Reader对应目录，${DATAX_HOME}为DataX主目录，即*${DATAX_HOME}/plugin/reader/rdbmswriter*。 · 在RDBMS Reader插件目录下有plugin.json配置文件，在此文件中注册您具体的数据库驱动，放在drivers数组中。RDBMS Reader插件在任务执行时会动态选择合适的数据库驱动连接数据库。 { "name": "rdbmsreader", "class": "com.alibaba.datax.plugin.reader.rdbmsreader.RdbmsReader", "description": "useScene: prod. mechanism: Jdbc connection using the database, execute select sql, retrieve data from the ResultSet. warn: The more you know about the database, the less problems you encounter.", "developer": "alibaba", "drivers": ["dm.jdbc.driver.DmDriver", "com.ibm.db2.jcc.DB2Driver", "com.sybase.jdbc3.jdbc.SybDriver", "com.edb.Driver"] } `` - 在rdbmsreader插件目录下有libs子目录，您需要将您具体的数据库驱动放到libs目录下。 `` $tree . |-- libs | |-- Dm7JdbcDriver16.jar | |-- commons-collections-3.0.jar | |-- commons-io-2.4.jar | |-- commons-lang3-3.3.2.jar | |-- commons-math3-3.1.1.jar | |-- datax-common-0.0.1-SNAPSHOT.jar | |-- datax-service-face-1.0.23-20160120.024328-1.jar | |-- db2jcc4.jar | |-- druid-1.0.15.jar | |-- edb-jdbc16.jar | |-- fastjson-1.1.46.sec01.jar	是	无

参数	描述	必选	默认值
password	数据源指定用户名的密码。	是	无
table	所选取的需要同步的表。	是	无
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息，默认使用所有列配置，例如[*]。 · 支持列裁剪，即列可以挑选部分列进行导出。 · 支持列换序，即列可以不按照表schema信息顺序进行导出。 · 支持常量配置，您需要按照JSON格式`["id","1", "'bazhen.csy'", "null", "to_char(a + 1)", "2.3" , "true"]`。 - id为普通列名。 - 1为整型数字常量。 - 'bazhen.csy'为字符串常量。 - null为空指针。 - to_char(a + 1)为函数表达式。 - 2.3为浮点数。 - true为布尔值。 · column必须显示您指定同步的列集合，不允许为空。	是	无
splitPk	RDBMS Reader进行数据抽取时，如果指定splitPk，表示您希望使用splitPk代表的字段进行数据分片。数据同步系统因此会启动并发任务进行数据同步，从而提高数据同步的效能。 · 推荐splitPk用户使用表主键，因为表主键通常情况下比较均匀，切分出来的分片也不容易出现数据热点。 · 目前splitPk仅支持整型数据切分，不支持浮点、字符串和日期等其他类型。如果您指定其他非支持类型，RDBMS Reader将报错。 · 如果不填写splitPk，将视作您不对单表进行切分，RDBMS Reader使用单通道同步全量数据。	否	空
where	筛选条件，RDBMS Reader根据指定的column、table和where条件拼接SQL，并根据该SQL进行数据抽取。例如在做测试时，可以将where条件指定为limit 10。在实际业务场景中，往往会选择当天的数据进行同步，可以将where条件指定为gmt_create>$bizdate。 · where条件可以有效地进行业务增量同步。 · where条件不配置或为空时，则视作全表同步数据。	否	无

参数	描述	必选	默认值
querySql	在部分业务场景中，where配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选SQL。当您配置该项后，数据同步系统会忽略column、table等配置，直接使用该配置项的内容对数据进行筛选。 例如需要进行多表join后同步数据，使用`select a,b from table_a join table_b on table_a.id = table_b.id`。当您配置querySql时，RDBMS Reader直接忽略column、table和where条件的配置。	否	无
fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据同步系统和服务器端的网络交互次数，能够提升数据抽取性能。 **说明:** fetchSize值过大（>2048）可能造成数据同步进程OOM。	否	1,024

向导开发介绍

暂不支持向导开发模式。

脚本开发介绍

配置一个从RDBMS数据库同步抽取数据作业。

```
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "concurrent": 1,
            "throttle": false
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
                "column": [
                    {
                        "type": "string",
```

```
                        "value": "field"
                    },
                    {
                        "type": "long",
                        "value": 100
                    },
                    {
                        "dateFormat": "yyyy-MM-dd HH:mm:ss",
                        "type": "date",
                        "value": "2014-12-12 12:12:12"
                    },
                    {
                        "type": "bool",
                        "value": true
                    },
                    {
                        "type": "bytes",
                        "value": "byte string"
                    }
                ],
                "sliceRecordCount": "10"
            },
            "stepType": "stream"
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {
                "connection": [
                    {
                        "jdbcUrl": "jdbc:dm://ip:port/database",
                        "table": [
                            "table"
                        ]
                    }
                ],
                "username": "username",
                "password": "password",
                "table": "table",
                "column": [
                    "*"
                ],
                "preSql": [
                    "delete from XXX;"
                ]
            },
            "stepType": "rdbms"
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

2.3.1.20 配置Stream Reader

本文将为您介绍Stream Reader支持的数据类型、字段映射和数据源等参数及配置示例。

Stream Reader插件实现了从内存中自动产生数据的功能，主要用于数据同步的性能测试和基本的功能测试。

Stream Reader支持的数据类型，如下所示。

数据类型	类型描述
string	字符型
long	长整型
date	日期类型
bool	布尔型
bytes	字节型

参数说明

参数	描述	必选	默认值
column	产生的源数据的列数据和类型，可以配置多列。可以配置产生随机字符串，并制定范围，示例如下。 `"column" : [ { "random": "8,15" }, { "random": "10,10" } ]` 配置项说明如下： · "random": "8, 15": 表示随机产生8~15位长度的字符串。 · "random": "10, 10": 表示随机产生10位长度的字符串。	是	无
sliceRecordCount	表示循环产生column的份数。	是	无

向导开发介绍

暂不支持向导开发模式。

脚本开发介绍

配置一个从内存中读数据的同步作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        {
            "stepType":"stream",//插件名。
            "parameter":{
```

```
                "column":[//字段。
                    {
                        "type":"string",//数据类型。
                        "value":"field"//值。
                    },
                    {
                        "type":"long",
                        "value":100
                    },
                    {
                        "dateFormat":"yyyy-MM-dd HH:mm:ss",//时间格式。
                        "type":"date",
                        "value":"2014-12-12 12:12:12"
                    },
                    {
                        "type":"bool",
                        "value":true
                    },
                    {
                        "type":"bytes",
                        "value":"byte string"
                    }
                ],
                "sliceRecordCount":"100000"//表示循环产生column的份数。
            },
            "name":"Reader",
            "category":"reader"
        },
        { //下面是关于Writer的模板，您可以查找相应的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
```

```
}
```

2.3.1.21 配置HybridDB for MySQL Reader

本文将为您介绍HybridDB for MySQL Reader支持的数据类型、字段映射和数据源等参数及配置示例。

HybridDB for MySQL Reader插件支持读取表和视图。表字段可以依序指定全部列、部分列、调整列顺序、指定常量字段和配置HybridDB for MySQL的函数，如now()等。

HybridDB for MySQL Reader插件从HybridDB for MySQL读取数据。在底层实现上，HybridDB for MySQL Reader通过JDBC连接远程HybridDB for MySQL数据库，并执行相应的SQL语句，从HybridDB for MySQL库中选取数据。

HybridDB for MySQL Reader插件通过JDBC连接器连接至远程的HybridDB for MySQL数据库，根据您配置的信息生成查询SQL语句，发送至远程HybridDB for MySQL数据库，执行该SQL语句并返回结果。然后使用数据同步自定义的数据类型将其拼装为抽象的数据集，传递给下游Writer处理。

类型转换列表

HybridDB for MySQL Reader针对HybridDB for MySQL类型的转换列表，如下所示。

类型分类	HybridDB for MySQL数据类型
整数类	INT、TINYINT、SMALLINT、MEDIUMINT和BIGINT
浮点类	FLOAT、DOUBLE和DECIMAL
字符串类	VARCHAR、CHAR、TINYTEXT、TEXT、MEDIUMTEXT和LONGTEXT
日期时间类	DATE、DATETIME、TIMESTAMP、TIME和YEAR
布尔型	BIT和BOOL
二进制类	TINYBLOB、MEDIUMBLOB、BLOB、LONGBLOB和VARBINARY

 说明:

- 除上述罗列字段类型外，其他类型均不支持。
- HybridDB for MySQL Reader插件将tinyint(1)视作整型。

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称，一个数据集成Job只能同步一张表。	是	无
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息 。默认使用所有列配置，例如[*]。 · 支持列裁剪，即列可以挑选部分列进行导出。 · 支持列换序，即列可以不按照表Schema信息顺序进行导出。 · 支持常量配置，您需要按照SQL语法格式，例如`["id", "table","1","'mingya.wmy'","'null'", "to_char(a+1)","2.3","true"]`。 - id为普通列名。 - table为包含保留字的列名。 - 1为整型数字常量。 - ‘mingya.wmy’为字符串常量（注意需要加上一对单引号）。 - 'null'为字符串常量。 - to_char(a+1)为计算字符串长度函数。 - 2.3为浮点数。 - true为布尔值。 · column必须显示指定同步的列集合，不允许为空。	是	无
splitPk	HybridDB for MySQL Reader进行数据抽取时，如果指定splitPk，表示您希望使用splitPk代表的字段进行数据分片，数据同步因此会启动并发任务进行数据同步，从而提高数据同步的效能。 · 推荐splitPk用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片也不容易出现数据热点。 · 目前splitPk仅支持整型数据切分，不支持字符串、浮点、日期等其他类型 。如果您指定其他非支持类型，忽略plitPk功能，使用单通道进行同步。 · 如果splitPk不填写，包括不提供splitPk或者splitPk值为空，数据同步视作使用单通道同步该表数据 。	否	无

参数	描述	必选	默认值
where	筛选条件，在实际业务场景中，往往会选择当天的数据进行同步，将where条件指定为gmt_create>$bizdate。 · where条件可以有效地进行业务增量同步。如果不填写where语句，包括不提供where的key或value，数据同步均视作同步全量数据。 · 不可以将where条件指定为limit 10，不符合SQL WHERE子句约束。	否	无
querySql（高级模式，向导模式不提供）	在部分业务场景中，where配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选SQL 。当配置此项后，数据同步系统就会忽略column、table和where配置项，直接使用该项配置的内容对数据进行筛选。例如需要进行多表join后同步数据，使用[“id”,“table”,“1”,“‘mingya.wmy’”,“‘null’”,“to_char(a+1)”,“2.3”,“true”]。当您配置querySql时，HybridDB for MySQL Reader直接忽略column、table和where和splitPk条件的配置，querySql优先级大于table、column、where、splitPk选项。datasource会使用它解析出用户名和密码等信息。	否	无
singleOrMulti（只适合分库分表）	表示分库分表，向导模式转换成脚本模式主动生成此配置“singleOrMulti”:“multi”，但配置脚本任务模板不会直接生成此配置，您需要手动添加，否则只会识别第一个数据源。singleOrMulti只是前端在用，后端没有用这个进行分库分表判断。	是	multi

向导模式

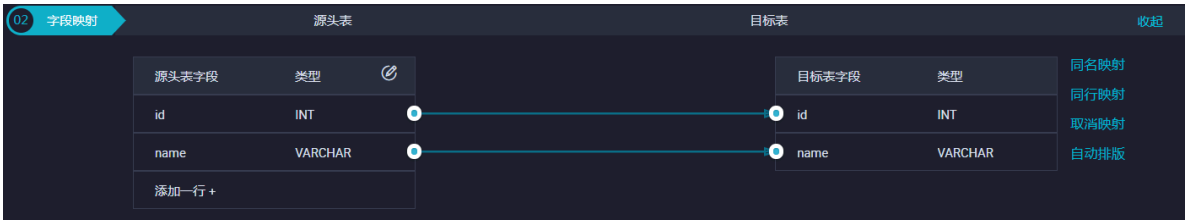
1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。
数据过滤	您将要同步数据的筛选条件，暂时不支持limit关键字过滤。SQL语法与选择的数据源一致。
切分键	您可以将源数据表中某一列作为切分键，建议使用主键或有索引的列作为切分键，仅支持类型为整型的字段。读取数据时，根据配置的字段进行数据分片，实现并发读取，可以提升数据同步效率。 **说明:** 切分键和数据同步中的选择来源有关，配置数据来源时才显示切分键配置项。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，将鼠标放至需要删除的字段上，即可单击删除按钮进行删除。



配置	说明
同名映射	单击同名映射，即可根据名称建立相应的同行映射关系，请注意匹配数据类型。
同行映射	单击同行映射，即可在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，即可取消建立的映射关系。
自动排版	单击自动排版，即可根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	添加一行的功能如下所示： · 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。

配置	说明
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

单库单表的脚本样例如下，详情请参见上述参数说明。

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {
                "datasource": "px_aliyun_hymysql",//数据源名。
                "column": [//源端列名。
                    "id",
                    "name",
                    "sex",
                    "salary",
                    "age",
                    "pt"
                ],
                "where": "id=10001",//过滤条件。
                "splitPk": "id",//切分键。
                "table": "person"//源端表名。
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {}
    ],
    "version": "2.0",//版本号。
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {//错误记录数。
            "record": ""
        },
        "speed": {
            "concurrent": 7,//并发数。
            "throttle": true,//同步速度限流。
            "mbps": 1,//限流值。
        }
    }
```

```
}
```

2.3.1.22 配置AnalyticDB for PostgreSQL Reader

本文将为您介绍AnalyticDB for PostgreSQL Reader支持的数据类型、读取方式、字段映射和数据源等参数及配置示例。

AnalyticDB for PostgreSQL Reader插件从AnalyticDB for PostgreSQL读取数据。在底层实现上，AnalyticDB for PostgreSQL Reader通过JDBC连接远程AnalyticDB for PostgreSQL数据库，并执行相应的SQL语句，从AnalyticDB for PostgreSQL库中选取数据。RDS提供AnalyticDB for PostgreSQL存储引擎。

AnalyticDB for PostgreSQL Reader通过JDBC连接器连接到远程的AnalyticDB for PostgreSQL数据库，根据您配置的信息生成查询SQL语句，发送至远程AnalyticDB for PostgreSQL数据库，执行该SQL并返回结果。然后使用数据同步自定义的数据类型将其拼装为抽象的数据集，传递给下游Writer处理。

- **对于您配置的`table`、`column`和`where`等信息，AnalyticDB for PostgreSQL Reader将其拼接为SQL语句，发送至AnalyticDB for PostgreSQL数据库。**
- **对于您配置的`querySql`信息，AnalyticDB for PostgreSQL直接将其发送至AnalyticDB for PostgreSQL数据库。**

类型转换列表

AnalyticDB for PostgreSQL Reader支持大部分AnalyticDB for PostgreSQL类型，但也存在部分类型没有支持的情况，请注意检查您的数据类型。

AnalyticDB for PostgreSQL Reader针对AnalyticDB for PostgreSQL的类型转换列表，如下所示。

类型分类	AnalyticDB for PostgreSQL数据类型
整数类	BIGINT、BIGSERIAL、INTEGER、SMALLINT和SERIAL
浮点类	DOUBLE、PRECISION、MONEY、NUMERIC和REAL
字符串类	VARCHAR、CHAR、TEXT、BIT和INET
日期时间类	DATE、TIME和TIMESTAMP
布尔型	BOOL
二进制类	BYTEA

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息 。默认使用所有列配置，例如[*]。 · 支持列裁剪，即列可以挑选部分列进行导出。 · 支持列换序，即列可以不按照表Schema信息顺序进行导出。 · 支持常量配置，您需要按照SQL语法格式，例如`["id", "table","1","'mingya.wmy'","'null'", "to_char(a+1)","2.3","true"]`。 - id为普通列名。 - table为包含保留字的列名。 - 1为整型数字常量。 - ‘mingya.wmy’为字符串常量（注意需要加上一对单引号）。 - 'null'为字符串常量。 - to_char(a+1)为计算字符串长度函数。 - 2.3为浮点数。 - true为布尔值。 · column必须显示指定同步的列集合，不允许为空。	是	无

参数	描述	必选	默认值
splitPk	AnalyticDB for PostgreSQL Reader进行数据抽取时，如果指定splitPk，表示您希望使用splitPk代表的字段进行数据分片。数据同步因此会启动并发任务进行数据同步，从而提高数据同步的效能。 · 因为通常表主键较为均匀，切分出的分片不易出现数据热点，所以推荐splitPk用户使用表主键。 · 目前splitPk仅支持整型数据切分，不支持字符串、浮点、日期等其他类型 。如果您指定其他非支持类型，忽略splitPk功能，使用单通道进行同步。 · 如果不填写splitPk，包括不提供splitPk或者splitPk值为空，数据同步视作使用单通道同步该表数据 。	否	无
where	筛选条件，AnalyticDB for PostgreSQLReader根据指定的column、table和where条件拼接SQL，并根据该SQL进行数据抽取。例如测试时，可以将where条件指定实际业务场景，往往会选择当天的数据进行同步，将where条件指定为`id>2 and sex=1`。 · where条件可以有效地进行业务增量同步。 · where条件不配置或者为空，视作全表同步数据。	否	无
querySql（高级模式，向导模式不提供）	在部分业务场景中，where配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选SQL。当配置此项后，数据同步系统就会忽略column、table等配置项，直接使用该项配置的内容对数据进行筛选。例如需要进行多表join后同步数据，使用`select a,b from table_a join table_b on table_a.id = table_b.id`。 当您配置querySql时，AnalyticDB for PostgreSQL Reader直接忽略column、table和where条件的配置。	否	无

参数	描述	必选	默认值
fetchSize	该配置项定义了插件和数据库服务器端每次批量数据获取条数，该值决定了数据集成和服务器端的网络交互次数，能够提升数据抽取性能。 **说明：** fetchSize值过大（>2048）可能造成数据同步进程OOM。	否	512

向导开发介绍

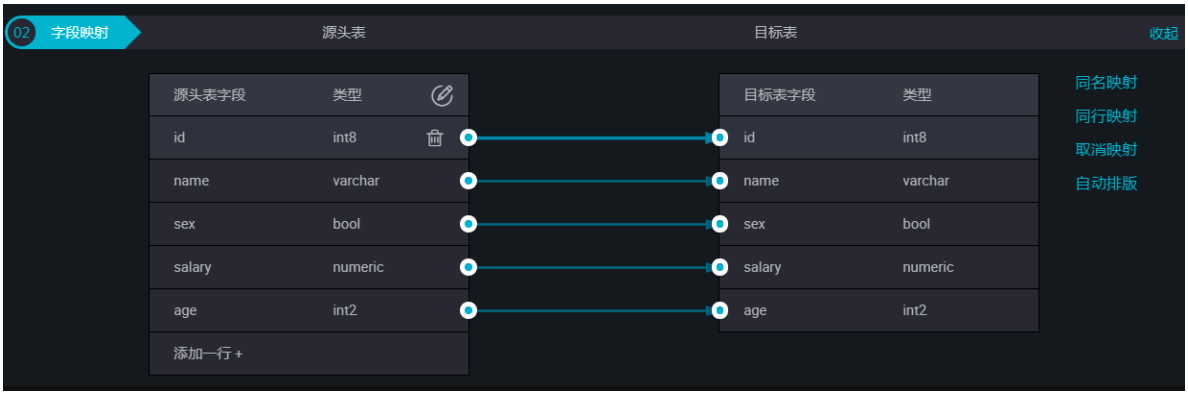
1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table，选择需要同步的表。
数据过滤	您将要同步数据的筛选条件，暂时不支持limit关键字过滤，SQL语法与选择的数据源一致。
切分键	您可以将源数据表中某一列作为切分键，建议使用主键或有索引的列作为切分键，仅支持类型为整型的字段。读取数据时，根据配置的字段进行数据分片，实现并发读取，可以提升数据同步效率。 **说明：** 切分键的设置跟数据同步里的选择来源有关，在配置数据来源时才显示切分键配置项。

2. 字段映射，即上述参数说明中的column。

左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，将鼠标放至需要删除的字段上，即可单击删除按钮进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {
                "datasource": "test_004",//数据源名称。
                "column": [//源端表的列名。
                    "id",
                    "name",
                    "sex",
                    "salary",
                    "age"
                ],
                "where": "id=1001",//过滤条件。
                "splitPk": "id",//切分键。
                "table": "public.person"//源端表名。
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {},
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",//版本号
    "order": {
```

```
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {//错误记录数。
            "record": ""
        },
        "speed": {
            "concurrent": 6,//并发数。
            "throttle": false,//同步速率是否限流。
        }
    }
}
```

2.3.1.23 配置POLARDB Reader

本文将为您介绍POLARDB Reader支持的数据类型、字段映射和数据源等参数及配置示例。

POLARDB Reader插件通过JDBC连接器连接至远程的POLARDB数据库，根据您配置的信息生成查询SQL语句，发送至远程POLARDB数据库，执行该SQL语句并返回结果。然后使用数据同步自定义的数据类型将其拼装为抽象的数据集，传递给下游Writer处理。

在底层实现上，POLARDB Reader插件通过JDBC连接远程POLARDB数据库，并执行相应的SQL语句，从POLARDB库中读取数据。

POLARDB Reader插件支持读取表和视图。表字段可以依序指定全部列、指定部分列、调整列顺序、指定常量字段和配置POLARDB的函数，例如now()等。

类型转换列表

POLARDB Reader针对POLARDB类型的转换列表，如下所示。

类型分类	POLARDB数据类型
整数类	INT、TINYINT、SMALLINT、MEDIUMINT和BIGINT
浮点类	FLOAT、DOUBLE和DECIMAL
字符串类	VARCHAR、CHAR、TINYTEXT、TEXT、MEDIUMTEXT和LONGTEXT
日期时间类	DATE、DATETIME、TIMESTAMP、TIME和YEAR
布尔型	BIT和BOOL
二进制类	TINYBLOB、MEDIUMBLOB、BLOB、LONGBLOB和VARBINARY

说明:

- **除上述罗列字段类型外，其他类型均不支持。**

- POLARDB Reader插件将tinyint(1)视作整型。

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称，一个数据集成Job只能同步一张表。	是	无
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息 。默认使用所有列配置，例如[*]。 · 支持列裁剪，即列可以挑选部分列进行导出。 · 支持列换序，即列可以不按照表Schema信息顺序进行导出。 · 支持常量配置，您需要按照SQL语法格式，例如["id", "table","1","'mingya.wmy'","'null'", "to_char(a+1)","2.3","true"]。 - id为普通列名。 - table为包含保留字的列名。 - 1为整型数字常量。 - ‘mingya.wmy’为字符串常量（注意需要加上一对单引号）。 - 'null'为字符串常量。 - to_char(a+1)为计算字符串长度函数。 - 2.3为浮点数。 - true为布尔值。 · column必须显示指定同步的列集合，不允许为空。	是	无

参数	描述	必选	默认值
splitPk	POLARDB Reader进行数据抽取时，如果指定splitPk，表示您希望使用splitPk代表的字段进行数据分片，数据同步因此会启动并发任务进行数据同步，从而提高数据同步的效能。 · 推荐splitPk用户使用表主键，因为表主键通常情况下比较均匀，因此切分出来的分片不容易出现数据热点。 · 目前splitPk仅支持整型数据切分，不支持字符串、浮点、日期等其他类型 。如果您指定其他非支持类型，忽略plitPk功能，使用单通道进行同步。 · 如果splitPk不填写，包括不提供splitPk或者splitPk值为空，数据同步视作使用单通道同步该表数据 。	否	无
where	筛选条件，在实际业务场景中，往往会选择当天的数据进行同步，将where条件指定为gmt_create>$bizdate。 · where条件可以有效地进行业务增量同步。如果不填写where语句，包括不提供where的key或value，数据同步均视作同步全量数据。 · 不可以将where条件指定为limit 10，这不符合MySQL SQL WHERE子句约束。	否	无
querySql（高级模式，向导模式不提供）	在部分业务场景中，where配置项不足以描述所筛选的条件，您可以通过该配置型来自定义筛选SQL。当配置该项后，数据同步系统就会忽略column、table和where配置项，直接使用该项配置的内容对数据进行筛选。例如需要进行多表 join 后同步数据，使用select a,b from table_a join table_b on table_a.id = table_b.id。当您配置querySql时，POLARDB Reader直接忽略column、table和where条件的配置，querySql优先级大于table、column、where、splitPk选项。datasource会使用它解析出用户名和密码等信息。	否	无

参数	描述	必选	默认值
singleOrMulti（只适合分库分表）	表示分库分表，向导模式转换成脚本模式主动生成此配置“singleOrMulti”: “multi”，但是配置脚本任务模板不会直接生成此配置必须手动添加，否则只会识别第一个数据源。singleOrMulti只是前端在用，后端没有用这个进行分库分表判断。	是	multi

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。
数据过滤	您将要同步数据的筛选条件，暂时不支持limit关键字过滤。SQL语法与选择的数据源一致。

配置	说明
切分键	您可以将源数据表中某一列作为切分键，建议使用主键或有索引的列作为切分键，仅支持类型为整型的字段。读取数据时，根据配置的字段进行数据分片，实现并发读取，可以提升数据同步效率。 说明: 切分键和数据同步中的选择来源有关，配置数据来源时才显示切分键配置项。

2. 字段映射，即上述参数说明中的`column`。

左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，将鼠标放至需要删除的字段上，即可单击删除按钮进行删除。

配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

单库单表的脚本样例如下，详情请参见上述参数说明。

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {
                "datasource": "test_005",//数据源名。
                "column": [//源端列名。
                    "id",
                    "name",
                    "age",
                    "sex",
                    "salary",
                    "interest"
                ],
                "where": "id=1001",//过滤条件。
                "splitPk": "id",//切分键。
                "table": "polardb_person"//源端表名。
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {}
    ],
    "version": "2.0",//版本号。
    "order": {
```

```
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {//错误记录数。
            "record": ""
        },
        "speed": {
            "concurrent": 6,//并发数。
            "throttle": false,//同步速率限流。
        }
    }
}
```

2.3.1.24 配置Elasticsearch Reader

本文将为您介绍Elasticsearch Reader的工作原理、功能和参数。

工作原理

- 通过Elasticsearch的_search+scroll+slice（即游标+分片）方式实现，slice结合DataX job的task多线程分片机制使用。
- 根据Elasticsearch中的mapping配置，进行数据类型转换。

更多详情请参见Elasticsearch官方文档。

基本配置

```
{
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    },
    "setting":{
        "errorLimit":{
            "record":"0" //错误记录数。
        },
        "jvmOption":"",
        "speed":{
            "concurrent":3,
            "throttle":false
        }
    },
    "steps":[
        {
            "category":"reader",
            "name":"Reader",
            "parameter":{
                "column":[ //读取列。
                    "id",
                    "name"
                ],
```

```
                "endpoint":"", //服务地址。
                "index":"",  //索引。
                "password":"",  //密码。
                "scroll":"",  //scroll标志。
                "search":"",  //查询query参数，与Elasticsearch的query内容
相同，使用_search api，重命名为search。
                "type":"default",
                "username":""  //用户名。
            },
            "stepType":"elasticsearch"
        },
        {
            "category":"writer",
            "name":"Writer",
            "parameter":{ },
            "stepType":"stream"
        }
    ],
    "type":"job",
    "version":"2.0" //版本号。
}
```

高级功能

- 支持全量拉取

  支持将Elasticsearch中一个文档的所有内容拉取为一个字段。

- 支持半结构化到结构化数据的提取

分类	说明
产生背景	Elasticsearch中的数据特征为字段不固定，且有中文名、数据使用深层嵌套的形式。为更好地方便下游业务对数据的计算和存储需求，特推出从半结构化到结构化的转换解决方案。
实现原理	将Elasticsearch获取到的JSON数据，利用JSON工具的路径获取特性，将嵌套数据扁平化为一维结构的数据。然后将数据映射至结构化数据表中，拆分Elasticsearch复合结构数据至多个结构化数据表。

分类	说明
解决方案	- JSON有嵌套的情况，通过path路径来解决。 ■ 属性 ■ 属性.子属性 ■ 属性[0].子属性 - 附属信息有一对多的情况，需要进行拆表拆行处理，进行遍历。 属性[*].子属性 - 数组归并，一个字符串数组内容，归并为一个属性，并进行去重。 属性[] 去重 - 多属性合一，将多个属性合并为一个属性。 属性1,属性2 - 多属性选择处理 属性1|属性2

参数说明

参数	描述	是否必选	默认值
endpoint	Elasticsearch的连接地址。	是	无
username	http auth中的username。	否	空
password	http auth中的password。	否	空
index	Elasticsearch中的index名。	是	无
type	Elasticsearch中index的type名。	否	index名
pageSize	每次读取数据的条数。	否	100
search	Elasticsearch的query参数。	是	无
scroll	Elasticsearch的分页参数，设置游标存放时间。	是	无
sort	返回结果的排序字段。	否	无
retryCount	失败后重试的次数。	否	300

参数	描述	是否必选	默认值
connTimeOut	客户端连接超时时间。	否	600000
readTimeOut	客户端读取超时时间。	否	600000
multiThread	http请求，是否有多线程。	否	true
column	Elasticsearch所支持的字段类型，样例中包含了全部。	是	无
full	是否支持将Elasticsearch的数据拉取为一个字段。	否	false
multi	是否支持将数组进行列拆多行的处理，需要辅助设置子属性。	否	false

补充配置：

```
"full":false,
        "multi": {
          "multi": true,
          "key":"crn_list[*]"
        }
```

2.3.1.25 配置AnalyticDB Reader

AnalyticDB Reader插件实现了从AnalyticDB读取数据。在底层实现上，AnalyticDB Reader通过JDBC连接远程AnalyticDB数据库，并根据AnalyticDB的推荐分页大小，执行相应的SQL语句，将数据从AnalyticDB库中分批Select出来。

数据类型转换

AnalyticDB类型	DataX类型	MaxCompute类型
bigint	long	bigint
tinyint	long	int
timestamp	date	datetime
varchar	string	string
smallint	long	int
int	long	int
float	string	double
double	string	double

AnalyticDB类型	DataX类型	MaxCompute类型
date	date	datetime
time	date	datetime

 说明:
不支持multivalue，会直接异常退出。

使用限制

当前版本，在大批量数据导出并且配置较低的机器上，会出现超时的情况。

- 当前mode=Select时，上限为30万行。
- 当前mode=ODPS时，上限为1亿行。
- 50列以上为AnalyticDB本身的限制，需要联系AnalyticDB的管理员进行手动调整。
- Java版本需要1.8及以上，编译转码`native2ascii LocalStrings.properties > LocalStrings_zh_CN.properties`。

参数说明

参数	描述	是否必选	默认值
table	需要导出的表的名称。	是	无
column	列名，如果没有，则为全部。	否	*
limit	限制导出的记录数。	否	无
where	where条件，方便添加筛选条件，此处的String会被直接作为SQL条件添加到查询语句中，例如`where id < 100`。	否	无
mode	导入类型，目前支持两种类型。 · Select：使用limit分页。 · ODPS：使用ODPS DUMP来导出数据，需要有ODPS的访问权限。	否	Select

参数	描述	是否必选	默认值
odps.accessKey	当mode=ODPS时必填，AnalyticDB访问ODPS使用的云账号AccessKey，需要有Describe、Create、Select、Alter、Update和Drop权限。	否	无
odps.accessId	当mode=ODPS时必填，AnalyticDB访问ODPS使用的云账号AccessID，需要有Describe、Create、Select、Alter、Update和Drop权限。	否	无
odps.odpsServer	当mode=ODPS时必填，ODPS API地址。	否	无
odps.tunnelServer	当mode=ODPS时必填，ODPS Tunnel地址。	否	无
odps.project	当mode=ODPS时必填，ODPS Project名称。	否	无
odps.accountType	当mode=ODPS时生效，ODPS访问账号类型。	否	aliyun

配置文件示例

```
{
    "type": "job",
    "steps": [
        {
            "stepType": "ads",
            "parameter": {
                "datasource": "ads_demo",
                "table": "th_test",
                "column": [
                    "id",
                    "testtinyint",
                    "testbigint",
                    "testdate",
                    "testtime",
                    "testtimestamp",
                    "testvarchar",
                    "testdouble",
                    "testfloat"
```

```
                ],
                "odps": {
                    "accessId": "*********",
                    "accessKey": "*********",
                    "account": "*********@aliyun.com",
                    "odpsServer": " http://service.cn.maxcompute.
aliyun-inc.com/api",
                    "tunnelServer": "http://dt.cn-shanghai.maxcompute.
aliyun-inc.com",
                    "accountType": "aliyun",
                    "project": "odps_test"
                },
                "mode": "ODPS"
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": ""
        },
        "speed": {
            "concurrent": 2,
            "throttle": false,
            "dmu": 1
        }
    }
}
```

2.3.1.26 配置Kafka Reader

Kafka Reader通过Kafka服务的Java SDK从Kafka读取数据。

Apache Kafka是一个快速、可扩展、高吞吐和可容错的分布式发布订阅消息系统。Kafka具有高吞吐量、内置分区、支持数据副本和容错的特性，适合在大规模消息处理的场景中使用。

消费消息的详情参见订阅者最佳实践。

实现原理

Kafka Reader通过`Kafka Java SDK`读取Kafka中的数据，使用的日志服务Java SDK版本如下所示。

```
<dependency>
```

```
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka-clients</artifactId>
    <version>2.0.0</version>
</dependency>
```

主要涉及的Kafka SDK调用方法如下，您可以参见Kafka官方了解接口的功能和限制。

- 使用KafkaConsumer作为消息消费的客户端。

```
org.apache.kafka.clients.consumer.KafkaConsumer<K,V>
```

- 根据unix时间戳查询Kafka点位offSet。

```
Map<TopicPartition,OffsetAndTimestamp> offsetsForTimes(Map<
TopicPartition,Long> timestampsToSearch)
```

- 定位到开始点位offSet。

```
public void seekToBeginning(Collection<TopicPartition> partitions)
```

- 定位到结束点位offSet。

```
public void seekToEnd(Collection<TopicPartition> partitions)
```

- 定位到指定点位offSet。

```
public void seek(TopicPartition partition,long offset)
```

- 客户端从服务端拉取poll数据。

```
public ConsumerRecords<K,V> poll(final Duration timeout)
```

说明:

Kafka Reader消费数据使用了自动点位提交机制。

参数说明

参数	说明	是否必填
server	Kafka的broker server地址，格式为`ip:port`。	是
topic	Kafka的topic，是Kafka处理资源的消息源（feeds of messages）的聚合。	是

参数	说明	是否必填
column	需要读取的Kafka数据，支持常量列、数据列和属性列。 · 常量列：使用单引号包裹的列为常量列，例如["'abc'", "'123'"]。 · 数据列 - 如果您的数据是一个JSON，支持获取JSON的属性，例如["event_id"]。 - 如果您的数据是一个JSON，支持获取JSON的嵌套子属性，例如["tag.desc"]。 · 属性列 - __key__ 表示消息的key。 - __value__ 表示消息的完整内容 。 - __partition__ 表示当前消息所在分区。 - __headers__ 表示当前消息headers信息。 - __offset__ 表示当前消息的偏移量。 - __timestamp__ 表示当前消息的时间戳。 完整示例如下： `"column": [ "__key__", "__value__", "__partition__", "__offset__", "__timestamp__", "'123'", "event_id", "tag.desc" ]`	是
keyType	Kafka的key的类型，包括BYTEARRAY、DOUBLE、FLOAT、INTEGER、LONG和SHORT。	是
valueType	Kafka的value的类型，包括BYTEARRAY、DOUBLE、FLOAT、INTEGER、LONG和SHORT。	是
beginDateTime	数据消费的开始时间位点，为时间范围（左闭右开）的左边界。yyyymmddhhmmss格式的时间字符串，可以和#unique_188配合使用。Kafka 0.10.2以上的版本支持此功能。	需要和beginOffset二选一。 说明: beginDateTime和endDateTime配合使用。

参数	说明	是否必填
endDateTime	数据消费的结束时间位点，为时间范围（左闭右开）的右边界。yyyymmddhhmmss格式的时间字符串，可以和#unique_188配合使用。Kafka 0.10.2以上的版本支持此功能。	需要和endOffset二选一。 说明: endDateTime和beginDateTime配合使用。
beginOffset	数据消费的开始时间位点，您可以配置以下形式。 · 例如15553274的数字形式，表示开始消费的点位。 · seekToBeginning：表示从开始点位消费数据。 · seekToLast：表示从上次的偏移位置读取数据。 · seekToEnd：表示从最后点位消费数据，会读取到空数据。	需要和beginDateTime二选一。
endOffset	数据消费的结束位点，用来控制什么时候应该结束数据消费任务退出。	需要和endDateTime二选一。
skipExceedRecord	Kafka使用public ConsumerRecords<K, V> poll(final Duration timeout)消费数据，一次poll调用获取的数据可能在endOffset或者endDateTime之外。skipExceedRecord用来控制这些多余的数据是否写出到目的端。由于消费数据使用了自动点位提交，建议： · Kafka 0.10.2之前版本：建议skipExceedRecord配置为false。 · Kafka 0.10.2及以上版本：建议skipExceedRecord配置为true。	否，默认值为false。
partition	Kafka的一个topic有多个分区（partition），正常情况下数据同步任务是读取topic（多个分区）一个点位区间的数据。您也可以指定partition，仅读取一个分区点位区间的数据。	否，无默认值。
kafkaConfig	创建Kafka数据消费客户端KafkaConsumer可以指定扩展参数，例如bootstrap.servers、auto.commit.interval.ms、session.timeout.ms等，您可以基于kafkaConfig控制KafkaConsumer消费数据的行为。	否

kafkaConfig参数说明如下：

- fetch.min.bytes：指定消费者从broker获取消息的最小字节数，即等到有足够的数据时才把它返回给消费者。
- fetch.max.wait.ms：等待broker返回数据的最大时间，默认500ms。fetch.min.bytes和fetch.max.wait.ms哪个条件先得到满足，便按照哪种方式返回数据。
- max.partition.fetch.bytes：指定broker从每个partition中返回给消费者的最大字节数，默认1MB。
- session.timeout.ms：指定消费者不再接收服务之前，可以与服务器断开连接的时间，默认是30s。
- auto.offset.reset：消费者在读取没有偏移量或者偏移量无效的情况下（因为消费者长时间失效，包含偏移量的记录已经过时并被删除）的处理方式。默认为latest（消费者从最新的记录开始读取数据），可更改为earliest（消费者从起始位置读取partition的记录）。
- max.poll.records：单次调用poll方法能够返回的消息数量。
- key.deserializer：消息key的反序列化方法，例如`org.apache.kafka.common.serialization.StringDeserializer`。
- value.deserializer：数据value的反序列化方法，例如`org.apache.kafka.common.serialization.StringDeserializer`。
- ssl.truststore.location：SSL根证书的路径。
- ssl.truststore.password：根证书store的密码，如果是Aliyun Kafka，则配置为KafkaOnsClient。
- security.protocol：接入协议，目前支持使用SASL_SSL协议接入。
- sasl.mechanism：SASL鉴权方式，如果是Aliyun Kafka，使用PLAIN。

配置示例如下：

```
{
    "group.id": "demo_test",
    "java.security.auth.login.config": "/home/admin/kafka_client_jaas.
conf",
    "ssl.truststore.location": "/home/admin/kafka.client.truststore.
jks",
    "ssl.truststore.password": "KafkaOnsClient",
    "security.protocol": "SASL_SSL",
    "sasl.mechanism": "PLAIN",
    "ssl.endpoint.identification.algorithm": ""
}
```

脚本开发示例

从Kafka读取数据的JSON配置，如下所示。

```
{
    "type": "job",
    "steps": [
```

```
        {
            "stepType": "kafka",
            "parameter": {
                "server": "host:9093",
                "column": [
                    "__key__",
                    "__value__",
                    "__partition__",
                    "__offset__",
                    "__timestamp__",
                    "'123'",
                    "event_id",
                    "tag.desc"
                ],
                "kafkaConfig": {
                    "group.id": "demo_test"
                },
                "topic": "topicName",
                "keyType": "ByteArray",
                "valueType": "ByteArray",
                "beginDateTime": "20190416000000",
                "endDateTime": "20190416000006",
                "skipExceedRecord": "false"
            },
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "stream",
            "parameter": {
                "print": false,
                "fieldDelimiter": ","
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "throttle": false,
            "concurrent": 1,
            "dmu": 1
        }
    }
```

```
}
```

2.3.1.27 配置InfluxDB Reader

InfluxDB是由InfluxData开发的开源时序型数据库，它由Go写成，致力于高性能地查询与存储时序型数据。InfluxDB Reader插件实现了从InfluxDB读取数据。

目前InfluxDB Reader仅支持脚本模式配置，更多详情请参见InfluxDB。

实现原理

在底层实现上，InfluxDB Reader通过Java Client，将SQL查询请求发送到InfluxDB实例，扫描出指定的数据点。整个同步的过程通过Database、Metric和时间段进行切分，组合为一个迁移Task。

约束限制

- 指定起止时间会被自动转为整点时刻，例如2019-4-18的`[3:35, 4:55)`，会被转为`[3:00, 4:00)`。
- 目前仅支持兼容InfluxDB 0.9及以上版本。

支持的数据类型

类型分类	数据集成column配置类型	TSDB数据类型
字符串	string	TSDB数据点序列化字符串，包括timestamp、metric、tags和value。

参数说明

参数	描述	是否必选	默认值
endpoint	InfluxDB的HTTP连接地址。	是，格式为`http://IP:Port`。	无
database	指定InfluxDB的数据库。	是	无
username	用于连接InfluxDB的账号。	是	无
password	用于连接InfluxDB的密码。	是	无
column	数据迁移任务需要迁移的Metric列表。	是	无

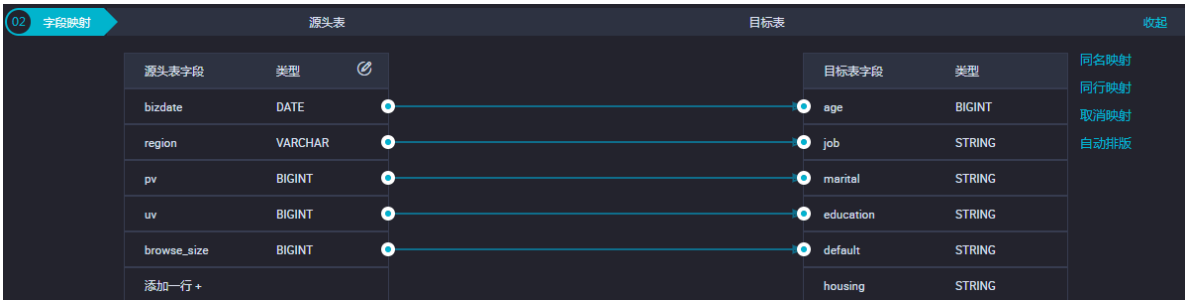
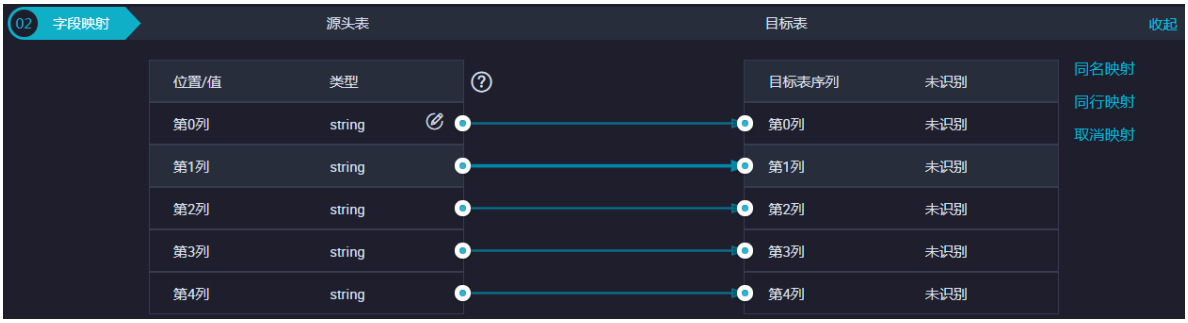
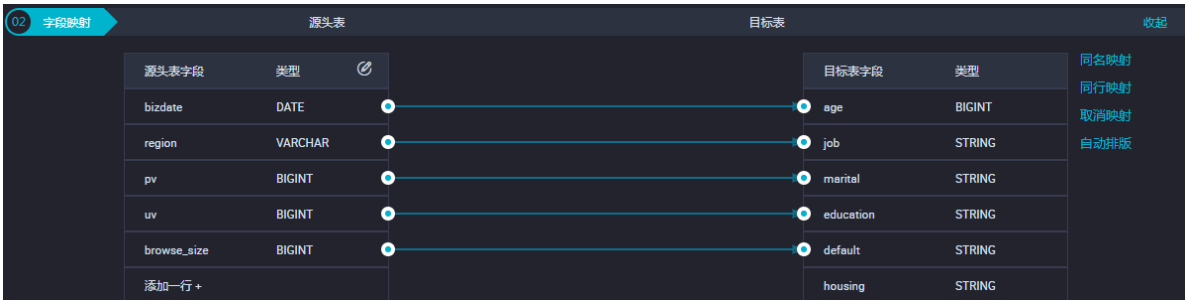
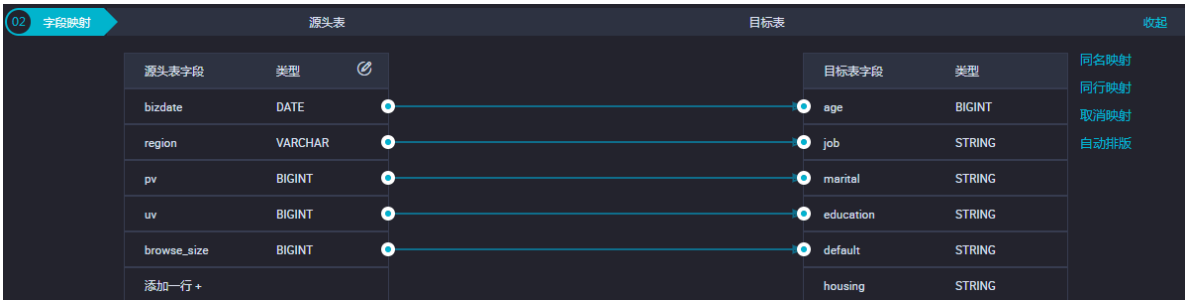
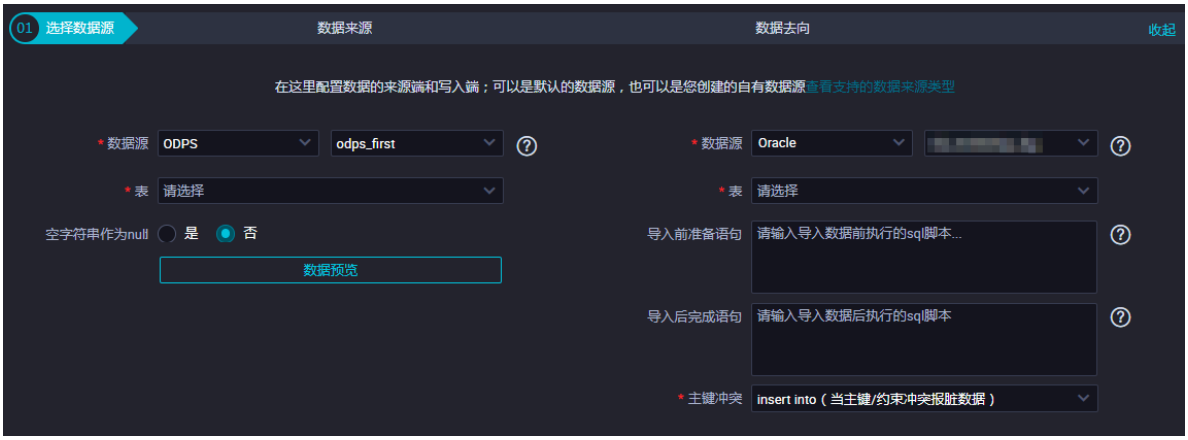
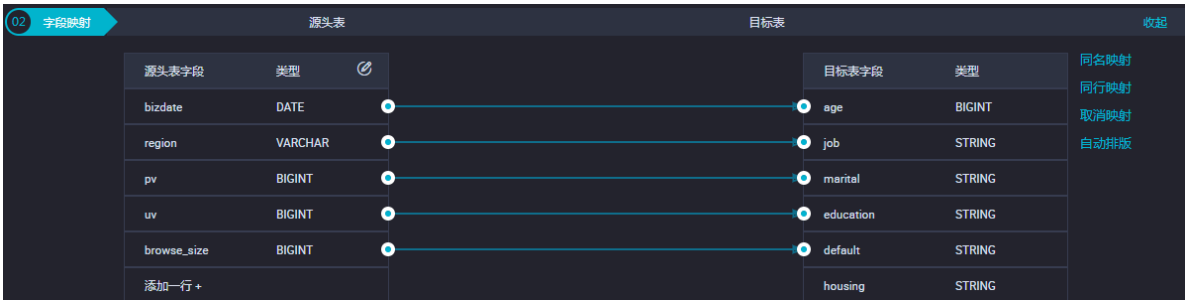
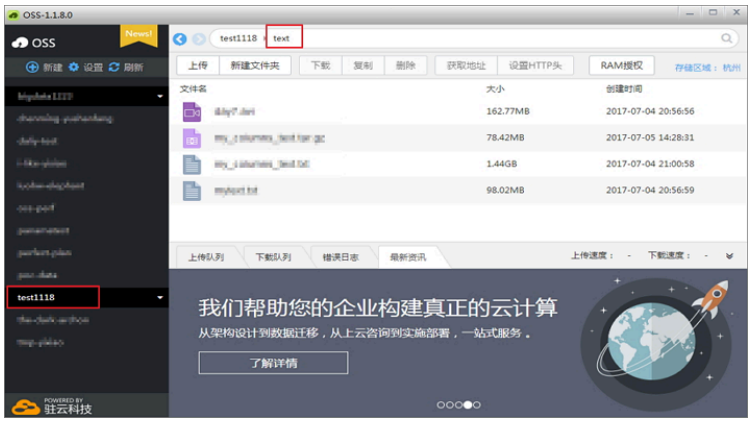
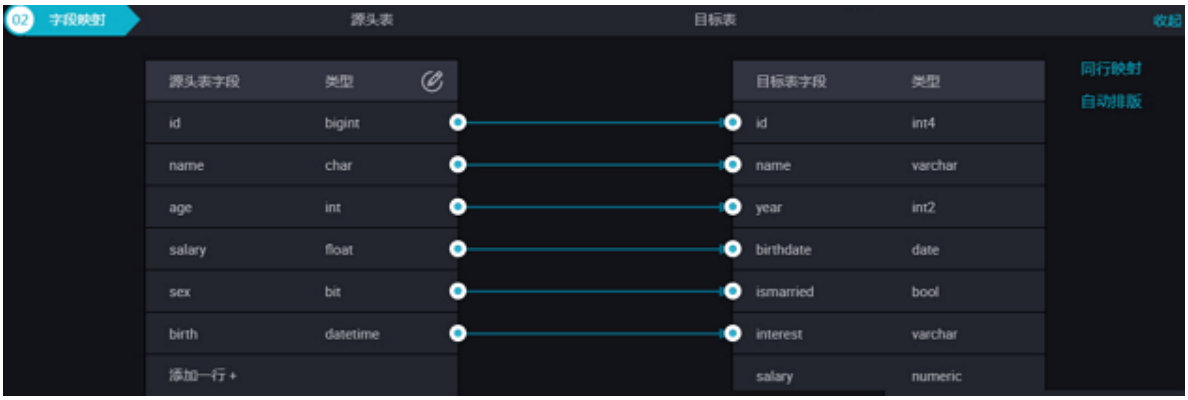
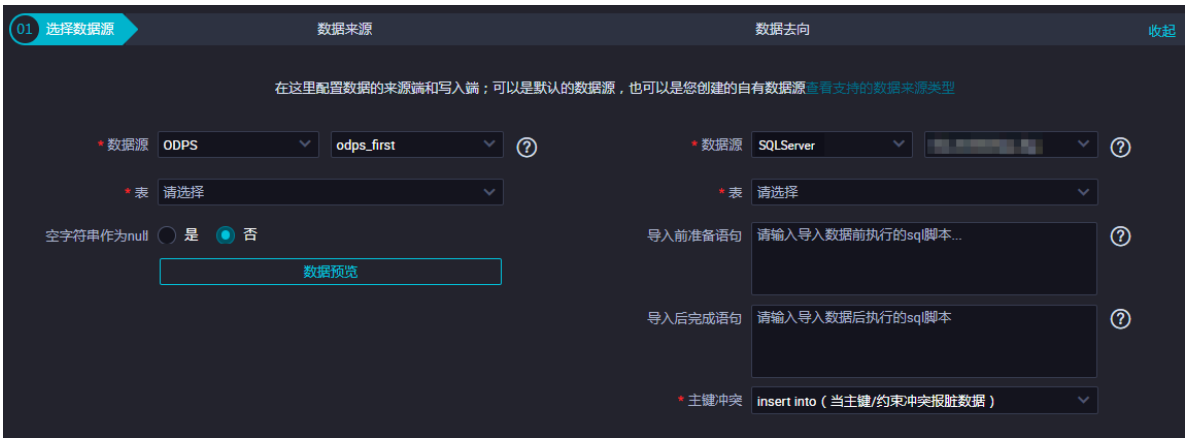
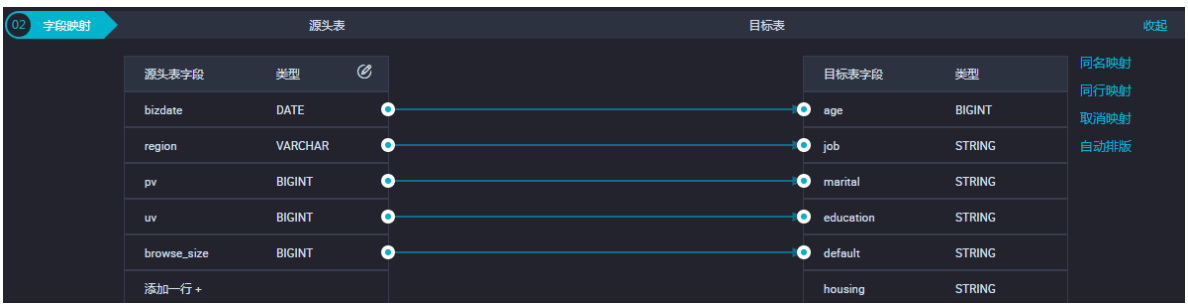
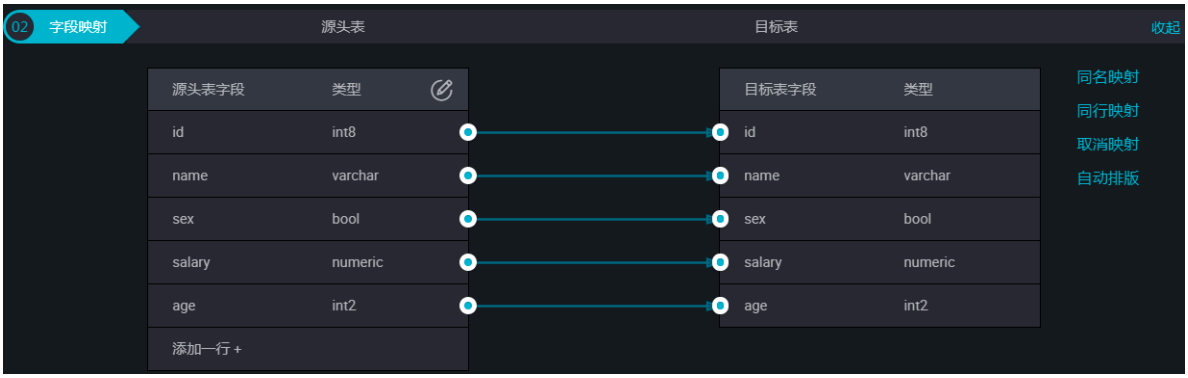
参数	描述	是否必选	默认值
beginDateTime	和endDateTime配合使用，用于指定哪个时间段内的数据点需要被迁移。	是，格式为yyyyMMddHHmmss。	无 说明: 指定起止时间会自动忽略分钟和秒，转为整点时刻。例如2019-4-18的[3:35，4:55)会被转为[3:00，4:00)。
endDateTime	和beginDateTime配合使用，用于指定哪个时间段内的数据点需要被迁移。	是，格式为yyyyMMddHHmmss。	无 说明: 指定起止时间会自动忽略分钟和秒，转为整点时刻。例如2019-4-18的[3:35，4:55)会被转为[3:00，4:00)。

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个从InfluxDB数据库同步的作业。

```
```json
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "concurrent": 1,
            "throttle": true
        }
    },
    "steps": [
        {
            "category": "reader",
```
```

```
            "name": "Reader",
            "parameter": {
                "endpoint": "http://host:8086",
                "database": "",
                "username": "",
                "password": "",
                "column": [
                    "xc"
                ],
                "endDateTime": "20190515180000",
                "beginDateTime": "20190515170000"
            },
            "stepType": "influxdb"
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {},
            "stepType": ""
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

2.3.1.28 配置OpenTSDB Reader

OpenTSDB是主要由Yahoo维护、可扩展、分布式的时序数据库，OpenTSDB Reader插件实现了从OpenTSDB读取数据。

OpenTSDB与阿里巴巴自研TSDB的关系与区别，请参见相比OpenTSDB优势。

目前OpenTSDB Reader仅支持脚本模式配置方式。

实现原理

在底层实现上，OpenTSDB Reader通过HTTP请求连接到OpenTSDB实例，用`/api/config`接口获取其底层存储HBase的连接信息。然后通过AsyncHBase框架连接HBase，以Scan的方式将数据点扫描出来。整个同步的过程通过Database、Metric和时间段进行切分，即某个Metric在某一个小时内的数据迁移，组合成一个迁移Task。

约束限制

- 指定起止时间会被自动转为整点时刻，例如2019-4-18的`[3:35, 4:55)`，会被转为`[3:00, 4:00)`。
- 目前仅支持兼容OpenTSDB 2.3.x版本。
- 不可直接使用`/api/query`查询获取数据点，需要连接OpenTSDB的底层存储。

  因为通过OpenTSDB的HTTP接口（`/api/query`）读取数据，在数据量较大的情况下，会导致OpenTSDB的异步框架报CallBack过多的异常。所以通过连接底层HBase存储，以Scan的

方式扫描数据点，可避免此问题。且通过指定Metric和时间范围，可顺序扫描HBase表，提高查询效率。

支持的数据类型

类型分类	数据集成column配置类型	TSDB数据类型
字符串	string	TSDB数据点序列化字符串，包括timestamp、metric、tags和value。

参数说明

参数	描述	是否必选	默认值
endpoint	OpenTSDB的HTTP连接地址。	是，格式为`http://IP:Port`。	无
column	数据迁移任务需要迁移的Metric列表。	是	无
beginDateTime	和endDateTime配合使用，用于指定哪个时间段内的数据点需要被迁移。	是，格式为`yyyyMMddHHmmss`。	无 说明: 指定起止时间会自动忽略分钟和秒，转为整点时刻。例如2019-4-18的`[3:35, 4:55)`会被转为`[3:00, 4:00)`。
endDateTime	和beginDateTime配合使用，用于指定哪个时间段内的数据点需要被迁移。	是，格式为`yyyyMMddHHmmss`。	无 说明: 指定起止时间会自动忽略分钟和秒，转为整点时刻。例如2019-4-18的`[3:35, 4:55)`会被转为`[3:00, 4:00)`。

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个从OpenTSDB数据库同步抽取数据到本地的作业。

```
```json
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "concurrent": 1,
            "throttle": true
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
                "endpoint": "http://host:4242",
                "column": [
                    "xc"
                ],
                "beginDateTime": "20190101000000",
                "endDateTime": "20190101030000"
            },
            "stepType": "opentsdb"
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {},
            "stepType": ""
        }
    ],
    "type": "job",
    "version": "2.0"
}
```
```

```
```

性能报告

- 性能数据特征

  从Metric、时间线、Value和采集周期四个方面进行描述。

  - Metric：指定一个Metric为m。
  - tagkv：前四个tagkv全排列，形成`10*20*100*100=2,000,000`条时间线，最后IP对应2,000,000条时间线，从1开始自增。

tag_k	tag_v
zone	z1~z10
cluster	c1~c20
group	g1~100
app	a1~a100
ip	ip1~ip2,000,000

  - value：度量值为[1, 100]区间内的随机值。
  - interval：采集周期为10秒，持续摄入3小时，总数据量为`3*60*60/10*2,000,000=2,160,000,000`个数据点。

- 性能测试结果

通道数	数据集成速度（Rec/s）	数据集成流量（MB/s）
1	215,428	25.65
2	424,994	50.60
3	603,132	71.81

2.3.1.29 配置Prometheus Reader

Prometheus是时间序列数据库，由SoundCloud开发并维护，是Google BorgMon监控系统的开源版本。Prometheus Reader插件实现了从Prometheus读取数据。

目前Prometheus Reader仅支持脚本模式配置方式。

实现原理

在底层实现上，Prometheus Reader通过HTTP请求连接到Prometheus实例，用`/api/v1/query_range`接口获取原始数据点。整个同步的过程通过Metric和时间段进行切分，组合为一个迁移Task。

约束限制

- 指定起止时间会被自动转为整点时刻，例如2019-4-18的`[3:35, 4:55)`，会被转为`[3:00, 4:00)`。
- 目前仅支持兼容Prometheus 2.9.x版本。
- 时间上切分的粒度，默认只有10s。

  `/api/v1/query_range`接口对查询的数据点数量有所限制。如果查询的时间范围过大，会报`exceeded maximum resolution of 11,000 points per timeseries`的异常。因此插件中默认选择10s作为查询的切分粒度。即使原始数据点的存储粒度为毫秒级，也只会查询出10,000个数据点，可满足`/api/v1/query_range`接口的限制。

支持的数据类型

类型分类	数据集成column配置类型	TSDB数据类型
字符串	string	TSDB数据点序列化字符串，包括timestamp、metric、tags和value。

参数说明

参数	描述	是否必选	默认值
endpoint	Prometheus的HTTP连接地址。	是，格式为`http://IP:Port`。	无
column	数据迁移任务需要迁移的Metric列表。	是	无
beginDateTime	和endDateTime配合使用，用于指定哪个时间段内的数据点需要被迁移。	是，格式为`yyyyMMddHHmmss`。	无 说明： 指定起止时间会自动忽略分钟和秒，转为整点时刻。例如2019-4-18的`[3:35, 4:55)`会被转为`[3:00, 4:00)`。

参数	描述	是否必选	默认值
endDateTime	和beginDateTime配合使用，用于指定哪个时间段内的数据点需要被迁移。	是，格式为yyyyMMddHHmmss。	无 说明：指定起止时间会自动忽略分钟和秒，转为整点时刻。例如2019-4-18的[3:35，4:55)会被转为[3:00，4:00)。

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个从Prometheus数据库同步的作业。

```
```json
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "concurrent": 1,
            "throttle": true
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
                "endpoint": "http://localhost:9090",
                "column": [
                    "up"
                ],
                "beginDateTime": "20190520150000",
                "endDateTime": "20190520160000"
            },
            "stepType": "prometheus"
        },
        {
            "category": "writer",
            "name": "Writer",
```
```

```
            "parameter": {},
            "stepType": ""
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

性能测试报告

通道数	数据集成速度（Rec/s）	数据集成流量（MB/s）
1	45,000	5.36
2	55,384	6.60
3	60,000	7.15

2.3.2 配置Writer插件

2.3.2.1 配置AnalyticDB Writer

本文将为您介绍AnalyticDB Writer支持的数据源、数据类型、字段映射、参数配置以及一个完整的导入数据操作示例。

数据集成通过实时导入的方式将数据导入AnalyticDB中，要求您必须提前在AnalyticDB中创建好实时表（普通表）。实时导入方式效率高，且流程简单。

如果数据源来源是RDS for SQLServer，详细的导入操作，请参见使用数据集成迁移。

开始配置AnalyticDB Writer插件前，请首先配置好数据源，详情请参见#unique_75。

AnalyticDB Writer针对AnalyticDB类型的转换列表，如下所示。

类型	AnalyticDB数据类型
整数类	INT、TINYINT、SMALLINT、BIGINT
浮点类	FLOAT和DOUBLE
字符串类	VARCHAR
日期时间类	DATE和TIMESTAMP
布尔类	BOOLEAN

参数说明

参数	描述	必选	默认值
连接url	AnalyticDB连接信息，格式为Address:Port。	是	无
数据库	AnalyticDB的数据库名称。	是	无

参数	描述	必选	默认值
Access Id	AnalyticDB对应的AccessKey Id。	是	无
Access Key	AnalyticDB对应的AccessKey Secret。	是	无
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须与添加的数据源名称保持一致。	是	无
table	目标表的表名称。	是	无
partition	目标表的分区名称，当目标表为普通表，需要指定该字段。	否	无
writeMode	Insert模式，在主键冲突情况下新的记录会覆盖旧的记录。	是	无
column	目的表字段列表，可以为["*"]，或者具体的字段列表，例如["a", "b", "c"]。	是	无
suffix	AnalyticDB url配置项的格式为ip:port，此部分为您定制的连接串，是可选参数（请参见MySQL支持的JDBC控制参数）。实际在AnalyticDB数据库访问时，会变成JDBC数据库连接串。例如配置suffix为autoReconnect=true&failOverReadOnly=false&maxReconnects=10。	否	无
batchSize	AnalyticDB提交数据写的批量条数，当writeMode为insert时，该值才会生效。	writeMode为insert时，为必选。	无
bufferSize	DataX数据收集缓冲区大小，缓冲区的目的是积累一个较大的Buffer，源头的数据首先进入到此Buffer中进行排序，排序完成后再提交到AnalyticDB。排序是根据AnalyticDB的分区列模式进行的，排序的目的是数据顺序对AnalyticDB服务端更友好（出于性能考虑）。 BufferSize缓冲区中的数据会经过batchSize批量提交到ADB中，通常需要设置bufferSize为batchSize数量的多倍。当writeMode为insert时，该值才会生效。	writeMode为insert时，为必选。	默认不配置不开启此功能。

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。

   表 2-1: 数据去向（Writer）配置信息

配置项	说明
数据源	选择ADS，系统将自动关联配置AnalyticDB数据源时设置的数据源名称。
表	选择AnalyticDB中的一张表，将Reader数据库中的数据同步至该表中。
导入模式	根据AnalyticDB中表的更新方式设置导入模式，包括批量导入和实时导入。 说明: 批量导入不支持从非MaxCompute数据源批量导入数据至AnalyticDB。请配置两个同步任务，先将数据导入MaxCompute，再批量导入AnalyticDB。
导入规则	· 写入前清理已有数据：导数据之前，清空表或者分区的所有数据，相当于`insert overwrite`。 · 写入前保留已有数据：导数据之前，不清理任何数据，每次运行数据都是追加进去的，相当于`insert into`。
一级分区	默认，不可修改。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。

配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。

配置	说明
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

```
{
    "type":"job",
    "version":"2.0",
    "steps":[ //下面是关于Writer的模板，您可以查找相应数据源的写插件文档。
        {
            "stepType":"stream",
            "parameter":{
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"ads",//插件名。
            "parameter":{
                "partition":"",//目标表的分区名称。
                "datasource":"",//数据源。
                "column":[//字段。
                     "id"
                ],
                "writeMode":"insert",//写入模式。
                "batchSize":"256",//一次性批量提交的记录数大小。
                "table":"",//表名
                "overWrite":"true"//AnalyticDB写入是否覆盖当前写入的表,
true为覆盖写入，false为不覆盖。（追加）写入。当 writeMode 为 Load 时，该值才会
生效。
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
```

```
    }
}
```

2.3.2.2 配置DataHub Writer

本文将为您介绍DataHub Writer支持的数据类型、字段映射和数据源等参数及配置示例。

DataHub是实时数据分发平台、流式数据（Streaming Data）的处理平台，提供对流式数据的发布（Publish）、订阅（Subscribe）和分发功能，让您可以轻松构建基于流式数据的分析和应用。

DataHub服务基于阿里云自研的飞天平台，具有高可用、低延迟、高可扩展和高吞吐的特点。它与阿里云流计算引擎StreamCompute无缝连接，您可以轻松使用SQL进行流数据分析。DataHub同时提供分发流式数据至MaxCompute（原ODPS）、OSS等云产品的功能。

 说明:

STRING字符串仅支持UTF-8编码，单个STRING列最长允许1MB。

参数配置

通过Channel将Source与Sink连接起来，所以在Writer端的Channel要对应Reader端的Channel类型。通常Channel包括Memory-Channel和File-channel两种类型，如下配置即File通道。

```
"agent.sinks.dataXSinkWrapper.channel": "file"
```

参数说明

参数	描述	是否必选	默认值
accessId	DataHub的accessId。	是	无
accessKey	DataHub的accessKey。	是	无
endpoint	对DataHub资源的访问请求，需要根据资源所属服务，选择正确的域名。 详情请参见DataHub访问域名。	是	无
maxRetryCount	任务失败的最多重试次数。	否	无
mode	value是STRING类型时，写入的模式。	是	无
parseContent	解析内容。	是	无

参数	描述	是否必选	默认值
project	项目（Project）是DataHub数据的基本组织单元，一个Project下包含多个Topic。 说明: DataHub的项目空间与MaxCompute的工作空间相互独立，您在MaxCompute中创建的项目不能复用于DataHub，需要独立创建。	是	无
topic	Topic是DataHub订阅和发布的最小单位，您可以用Topic来表示一类或者一种流数据。 详情请参见Project及Topic的数量限制。	是	无
maxCommitSize	为提高写出效率，DataX-On-Flume会积累Buffer数据，待积累的数据大小达到maxCommitSize大小（单位MB）时，批量提交到目的端。默认是1,048,576，即1MB数据。	否	1MB
batchSize	为提高写出效率，DataX-On-Flume会积累Buffer数据，待积累的数据条数达到batchSize大小（单位条数）时，批量提交到目的端。默认1,024，即1,024条数据。	否	1,024
maxCommitInterval	为提高写出效率，DataX-On-Flume会积累Buffer数据，待积累的数据条数达到maxCommitSize、batchSize大小限制时，批量提交到目的端。 如果数据采集源头长时间没有产出数据，为了保证数据的及时投递，增加了maxCommitInterval参数（单位毫秒），即Buffer数据的最长时间，超过此时间会强制投递。默认30,000，即30秒。	否	30,000
parseMode	日志解析模式，目前有不解析default模式和csv模式。不解析即采集到的一行日志，直接作为DataX的Record单列Column写出。csv模式支持配置一个列分隔符，一行日志通过分隔符分隔成DataX的Record的多列。	否	default

向导开发介绍

暂不支持向导开发模式。

脚本开发介绍

配置一个从内存中读数据的同步作业。

```
{
    "type": "job",
    "version": "2.0",//版本号。
    "steps": [
        { //下面是关于Writer的模板，您可以查找相应数据源的写插件文档。
            "stepType": "stream",
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "datahub",//插件名。
            "parameter": {
                "datasource": "",//数据源。
                "topic": "",//Topic是DataHub订阅和发布的最小单位，您可以用
Topic来表示一类或者一种流数据。
                "maxRetryCount": 500,//任务失败的重试的最多次数。
                "maxCommitSize": 1048576//待积累的数据Buffer大小达到
maxCommitSize大小（单位MB）时，批量提交到目的端。
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "errorLimit": {
            "record": ""//错误记录数。
        },
        "speed": {
            "concurrent": 20,//并发线程数。
            "throttle": false,//false代表不限流，下面的限流的速度不生效，
true代表限流。
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
}
```

2.3.2.3 配置DB2 Writer

本文为您介绍DB2 Writer支持的数据类型、字段映射和数据源等参数及配置示例。

DB2 Writer插件为您提供写入数据至DB2数据库的目标表的功能。在底层实现上，DB2 Writer通过JDBC连接远程DB2数据库，执行相应的`insert into`语句，将数据写入DB2，内部会分批次提交入库。

DB2 Writer面向ETL开发工程师，使用DB2 Writer从数仓导入数据至DB2。同时DB2 Writer可以作为数据迁移工具，为数据库管理员等用户提供服务。

DB2 Writer通过数据同步框架获取Reader生成的协议数据，通过`insert into`（当主键/唯一性索引冲突时，冲突的行会写不进去）语句，写入数据至DB2。另外出于性能考虑采用了`PreparedStatement + Batch`，并且设置了`rewriteBatchedStatements=true`，将数据缓冲到线程上下文Buffer中，当Buffer累计到预定阈值时，才发起写入请求。

说明:
整个任务至少需要具备`insert into`的权限，是否需要其他权限，取决于您配置任务时在preSql和postSql中指定的语句。

DB2 Writer支持大部分DB2类型，但也存在个别没有支持的情况，请注意检查您的数据类型。

DB2 Writer针对DB2类型的转换列表，如下所示。

类型分类	DB2数据类型
整数类	SMALLINT
浮点类	DECIMAL、REAL和DOUBLE
字符串类	CHAR、CHARACTER、VARCHAR、GRAPHIC、VARGRAPHIC、LONG VARCHAR、CLOB、LONG VARGRAPHIC和DBCLOB
日期时间类	DATE、TIME和TIMESTAMP
布尔类	—
二进制类	BLOB

参数说明

参数	描述	必选	默认值
jdbcUrl	描述的是到DB2数据库的JDBC连接信息，jdbcUrl按照DB2官方规范，DB2格式为`jdbc:db2://ip:port/database`，并可以填写连接附件控制信息。	是	无
username	数据源的用户名 。	是	无
password	数据源指定用户名的密码 。	是	无
table	所选取的需要同步的表 。	是	无
column	目标表需要写入数据的字段，字段之间用英文逗号分隔。例如：`"column": ["id", "name", "age"]`。如果要依次写入全部列，使用（*）表示。例如`"column": ["*"]`。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句，目前仅允许执行一条SQL语句，例如清除旧数据。	否	无

参数	描述	必选	默认值
postSql	执行数据同步任务之后执行的SQL语句，目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如加上某一个时间戳。	否	无
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据同步系统与MySQL的网络交互次数，并提升整体吞吐量。如果该值设置过大，会导致数据同步运行进程OOM异常。	否	1,024

向导开发介绍

暂不支持向导开发模式。

脚本开发介绍

配置一个写入DB2的数据同步作业。

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[
        { //下面是关于Writer的模板，您可以查找相应数据源的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"db2",//插件名。
            "parameter":{
                "postSql":[],//执行数据同步任务之前率先执行的SQL语句。
                "password":"",//密
                "jdbcUrl":"jdbc:db2://ip:port/database",//DB2数据库的
JDBC连接信息。
                "column":[
                    "id"
                ],
                "batchSize":1024,//一次性批量提交的记录数大小。
                "table":"",//表名。
                "username":"",//用户名。
                "preSql":[]//执行数据同步任务之后执行的SQL语句。
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
```

```
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

2.3.2.4 配置DRDS Writer

本文为您介绍DRDS Writer支持的数据类型、字段映射和数据源等参数及配置示例。

DRDS Writer插件为您提供将数据写入DRDS表的功能。在底层实现上，DRDS Writer通过JDBC连接远程DRDS数据库的Proxy，执行相应的`replace into`语句，写入数据至DRDS。

说明:

- 执行的SQL语句是`replace into`，为避免数据重复写入，需要您的表具备主键（Primary Key）或唯一性索引（Unique index）。
- 开始配置DRDS Writer插件前，请首先配置好数据源，详情请参见#unique_197。

DRDS Writer面向ETL开发工程师，使用DRDS Writer从数仓导入数据至DRDS。同时DRDS Writer可以作为数据迁移工具，为数据库管理员等用户提供服务。

DRDS Writer通过数据同步框架获取Reader生成的协议数据，通过`replace into`（没有遇到主键/唯一性索引冲突时，与`insert into`行为一致，冲突时会用新行替换原有行所有字段）的语句写入数据至DRDS。DRDS Writer累积一定数据，提交给DRDS的Proxy，该Proxy内部决定数据是写入一张还是多张表，以及多张表写入时如何路由数据。

说明:

整个任务至少需要具备`replace into`的权限，是否需要其他权限，取决于您配置任务时在preSql和postSql中指定的语句。

类似于MySQL Writer，目前DRDS Writer支持大部分MySQL类型，但也存在个别类型没有支持的情况，请注意检查您的数据类型。

DRDS Writer针对DRDS类型的转换列表，如下所示。

类型分类	DRDS数据类型
整数类	INT、TINYINT、SMALLINT、MEDIUMINT、BIGINT和YEAR
浮点类	FLOAT、DOUBLE和DECIMAL
字符串类	VARCHAR、CHAR、TINYTEXT、TEXT、MEDIUMTEXT和LONGTEXT

类型分类	DRDS数据类型
日期时间类	DATE、DATETIME、TIMESTAMP和TIME
布尔类	BIT和BOOL
二进制类	TINYBLOB、MEDIUMBLOB、BLOB、LONGBLOB和VARBINARY

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	所选取的需要同步的表。	是	无
writeMode	选择导入模式，可以支持insert into、on duplicate key update和replace into三种方式。 · insert into：当主键/唯一性索引冲突时会写不进去冲突的行，以脏数据的形式体现。 · on duplicate key update：没有遇到主键/唯一性索引冲突时，与insert into行为一致。冲突时会用新行替换已经指定的字段的语句，写入数据至MySQL。 · replace into：没有遇到主键/唯一性索引冲突时，与insert into行为一致。冲突时会先删除原有行，再插入新行。即新行会替换原有行的所有字段。	否	insert
column	目标表需要写入数据的字段，字段之间用英文逗号分隔，例如"column": ["id", "name", "age"]。如果要依次写入全部列，使用（*）表示，例如"column": ["*"]。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句，目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如清除旧数据。	否	无
postSql	执行数据同步任务之后执行的SQL语句，目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如加上某一个时间戳。	否	无
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据同步系统与MySQL的网络交互次数，并提升整体吞吐量。如果该值设置过大，会导致数据同步运行进程OOM异常。	否	1,024

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。
导入前准备语句	即上述参数说明中的preSql，输入执行数据同步任务之前率先执行的SQL语句。
导入后完成语句	即上述参数说明中的postSql，输入执行数据同步任务之后执行的SQL语句。
主键冲突	即上述参数说明中的writeMode，可以选择需要的导入模式。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。

配置	说明
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

配置一个写入DRDS的数据同步作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
```

```
    "steps":[
        {//下面是关于Writer的模板，您可以查找相应数据源的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
                },
        {
            "stepType":"drds",//插件名。
            "parameter":{
                "postSql":[],//执行数据同步任务之后执行的SQL语句。
                "datasource":"",//数据源。
                "column":[//列名。
                "id"
                ],
                "writeMode":"insert ignore",
                "batchSize":"1024",//一次性批量提交的记录数大小。
                "table":"test",//表名。
                "preSql":[]//执行数据同步任务之前执行的SQL语句。
                },
            "name":"Writer",
            "category":"writer"
                }
                ],
    "setting":{
        "errorLimit":{
        "record":"0"//错误记录数。
            },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//并发数。
                }
            },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
                }
            ]
        }
    }
```

2.3.2.5 配置FTP Writer

本文为您介绍FTP Writer支持的数据类型、字段映射和数据源等参数及配置示例。

FTP Writer实现了向远程FTP文件写入CSV格式的一个或多个文件。在底层实现上，FTP Writer将数据集成传输协议下的数据转换为CSV格式，并使用FTP相关的网络协议写出至远程FTP服务器。

说明:

开始配置FTP Writer插件前，请首先配置好数据源，详情请参见#unique_199。

写入FTP文件内容存放的是一张逻辑意义上的二维表，例如CSV格式的文本信息。

FTP Writer实现了从数据集成协议转为FTP文件功能，FTP文件本身是无结构化数据存储。目前FTP Writer支持的功能如下：

- 支持且仅支持写入文本类型（不支持BLOB，如视频数据）的文件，且要求文本中schema为一张二维表。
- 支持类CSV和TEXT格式的文件，自定义分隔符。
- 写出时不支持文本压缩。
- 支持多线程写入，每个线程写入不同子文件。

暂时不支持以下两种功能：

- 单个文件不能支持并发写入。
- FTP本身不提供数据类型，FTP Writer均将数据以STRING类型写入FTP文件。

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
timeout	连接FTP服务器连接超时时间，单位毫秒。	否	60,000（1分钟）
path	FTP文件系统的路径信息，FTP Writer会写入Path目录下多个文件。	是	无
fileName	FTP Writer写入的文件名，该文件名会添加随机的后缀作为每个线程写入实际文件名。	是	无
writeMode	FTP Writer写入前数据清理处理模式。 - truncate：写入前清理目录下，fileName前缀的所有文件。 - append：写入前不做任何处理，数据集成FTP Writer直接使用filename写入，并保证文件名不冲突。 - nonConflict：如果目录下有fileName前缀的文件，直接报错。	是	无
fieldDelimiter	写入的字段分隔符。	是，单字符	无
skipHeader	类CSV格式文件可能存在表头为标题情况，需要跳过。默认不跳过，压缩文件模式下不支持skipHeader。	否	false
compress	支持gzip和bzip2两种压缩形式。	否	无压缩
encoding	读取文件的编码配置。	否	utf-8

参数	描述	必选	默认值
nullFormat	文本文件中无法使用标准字符串定义null（空指针），数据集成提供nullFormat定义哪些字符串可以表示为null。 例如您配置nullFormat="null"，如果源头数据是null，数据集成视作null字段。	否	无
dateFormat	日期类型的数据序列化到文件中时的格式，例如"dateFormat":"yyyy-MM-dd"。	否	无
fileFormat	文件写出的格式，包括CSV和TEXT两种，CSV是严格的CSV格式，如果待写数据包括列分隔符，则会按照CSV的转义语法转义，转义符号为双引号。TEXT格式是用列分隔符简单分割待写数据，对于待写数据包括列分隔符情况下不做转义。	否	TEXT
header	txt写出时的表头，例如['id', 'name', 'age']。	否	无
markDoneFileName	标档文件名，同步任务结束后生成标档文件，根据此标档文件可以判断同步任务是否成功。此处应配置为绝对路径。	否	无

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
文件路径	即上述参数说明中的path。

配置	说明
文本类型	读取的文件类型，默认情况下文件作为csv格式文件进行读取。
列分隔符	即上述参数说明中的fieldDelimiter，默认值为（,）。
编码格式	即上述参数说明中的encoding，默认值为utf-8。
null值	即上述参数说明中的nullFormat，定义表示null值的字符串。
时间格式	即上述参数说明中的dateFormat。
前缀冲突	即上述参数说明中的writeMode，定义表示null值的字符串。

2. 字段映射，即上述参数说明中的column。

左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，将鼠标放至需要删除的字段上，即可单击删除按钮进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

配置一个写入FTP数据库的同步作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        { //下面是关于Writer的模板，您可以查找相应数据源的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"ftp",//插件名。
            "parameter":{
                "path":"",//文件路径。
                "fileName":"",//文件名称。
                "nullFormat":"null",//null值。
                "dateFormat":"yyyy-MM-dd HH:mm:ss",//时间格式。
                "datasource":"",//数据源。
                "writeMode":"",//写入模式。
                "fieldDelimiter":",",//列分隔符。
                "encoding":"",//编码格式。
                "fileFormat":""//文本类型。
            },
            "name":"Writer",
            "category":"writer"
        }
```

```
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

2.3.2.6 配置HBase Writer

本文为您介绍HBase Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置示例。

HBase Writer插件实现了向HBase中写入数据。在底层实现上，HBase Writer通过HBase的Java客户端连接远程HBase服务，并通过put方式写入HBase。

支持的功能

- 支持HBase0.94.x和HBase1.1.x版本
  - 如果您的HBase版本为HBase0.94.x，Writer端的插件请选择hbase094x。

    ```
    "writer": {
            "hbaseVersion": "hbase094x"
        }
    ```

  - 如果您的HBase版本为HBase1.1.x，Writer端的插件请选择hbase11x。

    ```
    "writer": {
            "hbaseVersion": "hbase11x"
        }
    ```

    

    说明:

    HBase1.1.x插件当前可兼容HBase 2.0，如果您在使用上遇到问题请提交工单。
- 支持源端多个字段拼接作为rowkey

  目前HBase Writer支持源端多个字段拼接作为HBase表的rowkey。

- 写入HBase的版本支持

  写入HBase的时间戳（版本）支持：

  - 当前时间作为版本。
  - 指定源端列作为版本。
  - 指定一个时间作为版本。

支持的数据类型

支持读取HBase数据类型，HBase Writer针对HBase类型的转换列表，如下表所示。

> 说明：
> - column的配置需要和HBase表对应的列类型保持一致。
> - 除下表中罗列的字段类型外，其他类型均不支持。

类型分类	数据库数据类型
整数类	int、long和short
浮点类	float和double
布尔类	boolean
字符串类	string

参数说明

参数	描述	必选	默认值

haveKerberos	haveKerberos值为true时，表示HBase集群需要kerberos认证。
	说明：
	- 如果该值配置为true，必须要配置下面五个kerberos认证相关参数：
	  - kerberosKeytabFilePath
	  - kerberosPrincipal
	  - hbaseMasterKerberosPrincipal
	  - hbaseRegionserverKerberosPrincipal
	  - hbaseRpcProtection
	- 如果HBase集群没有kerberos认证，则不需要配置以上参数。
		否	false

参数	描述	必选	默认值
hbaseConfig	连接HBase集群需要的配置信息，JSON格式。必填的配置为hbase.zookeeper.quorum，表示HBase的ZK链接地址。同时可以补充更多HBase client的配置，例如设置scan的cache、batch来优化与服务器的交互。	是	无
mode	写入HBase的模式，目前仅支持normal模式，后续考虑动态列模式。	是	无
table	要写入的HBase表名（大小写敏感）。	是	无
encoding	编码方式，UTF-8或GBK，用于String转HBase byte[]时的编码。	否	utf-8
column	要写入的HBase字段： · index：指定该列对应Reader端column的索引，从0开始。 · name：指定HBase表中的列，格式必须为列族:列名。 · type：指定写入的数据类型，用于转换HBase byte[]。	是	无
maxVersion	指定在多版本模式下的HBase Reader读取的版本数，取值只能为-1或大于1的数字，-1表示读取所有版本。	multiVersionFixedColumn模式下必填项	无

参数	描述	必选	默认值

range	指定HBase Reader读取的rowkey范围：
	· startRowkey：指定开始rowkey。
	· endRowkey：指定结束rowkey。
	· isBinaryRowkey：指定配置的startRowkey和endRowkey转换为byte[]时的方式，默认值为false。如果为true，则调用Bytes.toBytesBinary(rowkey)方法进行转换。若为false，则调用 Bytes.toBytes(rowkey)。配置格式如下所示：
	`"range": { "startRowkey": "aaa", "endRowkey": "ccc", "isBinaryRowkey":false }`
	配置格式如下所示。
	`"column": [ { "index":1, "name": "cf1:q1", "type": "string" }, { "index":2, "name": "cf1:q2", "type": "string" } ]`
		否	无

rowkeyColumn	要写入的HBase的rowkey列： · index：指定该列对应Reader端column的索引，从0开始。如果是常量，index为-1。 · type：指定写入的数据类型，用于转换HBase byte[]。 · value：配置常量，常作为多个字段的拼接符。HBase Writer会将rowkeyColumn中所有列按照配置顺序进行拼接作为写入HBase的rowkey，不能全为常量。 配置格式如下所示。 `"rowkeyColumn": [ { "index":0, "type":"string" }, { "index":-1, "type":"string", "value":"_" } ]`	是	无
versionCol	指定写入HBase的时间戳。支持当前时间、指定时间列或指	否	无

参数	描述	必选	默认值
walFlag	HBae Client向集群中的RegionServer提交数据时（Put/Delete操作），首先会先写WAL（Write Ahead Log）日志（即HLog，一个RegionServer上的所有Region共享一个HLog），只有当WAL日志写成功后，才会接着写MemStore，最后客户端被通知提交数据成功。如果写WAL日志失败，客户端则被通知提交失败。关闭（false）放弃写WAL日志，从而提高数据写入的性能。	否	false
writeBufferSize	设置HBae Client的写Buffer大小，单位字节，配合autoflush使用。 autoflush： · 开启（true）：表示HBase Client在写的时候有一条put就执行一次更新。 · 关闭（false）：表示HBase Client在写的时候只有当put填满客户端写缓存时，才实际向HBase服务端发起写请求。	否	8M

向导开发介绍

暂不支持向导开发模式开发。

脚本开发介绍

配置一个从本地写入hbase1.1.x的作业。

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[
        { //下面是关于Reader的模板，可以找相应的读插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"hbase",//插件名
            "parameter":{
                "mode":"normal",//写入HBase的模式。
                "walFlag":"false",//关闭（false）放弃写WAL日志
                "hbaseVersion":"094x",//Hbase版本
                "rowkeyColumn":[//要写入的HBase的rowkey列。
                    {
                        "index":"0",//序列号
                        "type":"string"//数据类型
                    },
                    {
                        "index":"-1",
                        "type":"string",
```

```
                        "value":"_"
                    }
                ],
                "nullMode":"skip",//读取的为null值时，如何处理。
                "column":[//要写入的HBase字段。
                    {
                        "name":"columnFamilyName1:columnName1",//字段名
                        "index":"0",//索引号
                        "type":"string"//数据类型
                    },
                    {
                        "name":"columnFamilyName2:columnName2",
                        "index":"1",
                        "type":"string"
                    },
                    {
                        "name":"columnFamilyName3:columnName3",
                        "index":"2",
                        "type":"string"
                    }
                ],
                "writeMode":"api",//写入模是
                "encoding":"utf-8",//编码格式
                "table":"",//表名
                "hbaseConfig":{//连接HBase集群需要的配置信息，JSON格式。
                    "hbase.zookeeper.quorum":"hostname",
                    "hbase.rootdir":"hdfs: //ip:port/database",
                    "hbase.cluster.distributed":"true"
                }
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
```

```
}
```

2.3.2.7 配置HBase11xsql Writer

本文为您介绍HBase11xsql Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置举例。

HBase11xsql Writer实现了向Hbase中的SQL表（phoenix）批量导入数据的功能。Phoenix因为对rowkey做了数据编码，所以直接使用HBaseAPI进行写入会面临手工数据转换的问题，麻烦且易错。HBase11xsql Writer插件为您提供了单间的SQL表的数据导入方式。

在底层实现上，通过Phoenix的JDBC驱动，执行UPSERT语句向Hbase写入数据。

支持的功能

支持带索引的表的数据导入，可以同步更新所有的索引表。

限制

HBase11xsql Writer插件的限制如下所示。

- 仅支持1.x系列的Hbase。
- 仅支持通过phoenix创建的表，不支持原生HBase表。
- 不支持带时间戳的数据导入。

实现原理

通过Phoenix的JDBC驱动，执行UPSERT语句向表中批量写入数据。因为使用上层接口，所以可以同步更新索引表。

参数说明

参数	描述	是否必选	默认值
plugin	插件名字，必须是hbase11xsql。	是	无
table	要导入的表名，大小写敏感，通常phoenix表都是大写表名。	是	无
column	列名，大小写敏感。通常phoenix的列名都是大写。 说明: · 列的顺序必须与Reader输出的列的顺序一一对应。 · 不需要填写数据类型，会自动从phoenix获取列的元数据。	是	无

参数	描述	是否必选	默认值
hbaseConfig	hbase集群地址，zk为必填项，格式为ip1, ip2, ip3。 说明: · 多个IP之间使用英文的逗号分隔。 · znode是可选的，默认值是/hbase。	是	无
batchSize	批量写入的最大行数。	否	256
nullMode	读取到的列值为null时，您可以通过以下两种方式进行处理。 · skip：跳过这一列，即不插入这一列（如果该行的这一列之前已经存在，则会被删除）。 · empty：插入空值，值类型的空值是0，varchar的空值是空字符串。	否	skip

脚本开发介绍

脚本配置示例如下。

```
{
  "type": "job",
  "version": "1.0",
  "configuration": {
    "setting": {
      "errorLimit": {
        "record": "0"
      },
      "speed": {
        "mbps": "1",
        "concurrent": "1"
      }
    },
    "reader": {
      "plugin": "odps",
      "parameter": {
        "datasource": "",
        "table": "",
        "column": [],
        "partition": ""
      }
    },
    "plugin": "hbase11xsql",
    "parameter": {
      "table": "目标hbase表名，大小写有关",
      "hbaseConfig": {
        "hbase.zookeeper.quorum": "目标hbase集群的ZK服务器地址，向PE咨询",
        "zookeeper.znode.parent": "目标hbase集群的znode，向PE咨询"
      },
      "column": [
        "columnName"
      ],
      "batchSize": 256,
      "nullMode": "skip"
```

```
    }
  }
}
```

约束限制

Writer中的列的定义顺序必须与Reader的列顺序匹配，Reader中的列顺序定义了输出的每一行中，列的组织顺序。而Writer的列顺序，定义的是在收到的数据中，Writer期待的列的顺序。示例如下：

Reader的列顺序为c1，c2，c3，c4。

Writer的列顺序为x1，x2，x3，x4。

则Reader输出的列c1就会赋值给Writer的列x1。如果Writer的列顺序是x1，x2，x4，x3，则c3会赋值给x4，c4会赋值给x3。

常见问题

Q：并发设置多少比较合适？速度慢时增加并发有用吗？

A：数据导入进程默认JVM的堆大小是2GB，并发（channel数）是通过多线程实现的，开过多的线程有时并不能提高导入速度，反而可能因为过于频繁的GC导致性能下降。一般建议并发数（channel）为5-10。

Q：batchSize设置多少比较合适？

A：默认是256，但应根据每行的大小来计算最合适的batchSize。通常一次操作的数据量在2MB-4MB左右，用这个值除以行大小，即可得到batchSize。

2.3.2.8 配置HDFS Writer

本文为您介绍HDFS Writer支持的数据类型、字段映射和数据源等参数及配置示例。

HDFS Writer提供向HDFS文件系统指定路径中写入TextFile文件、 ORCFile文件以及ParquetFile格式文件，文件内容可以与Hive中的表关联。开始配置HDFS Writer插件前，请首先配置好数据源，详情请参见#unique_203。

说明：

HBase1.1.x插件目前可以兼容HBase 2.0，如果您在使用上遇到问题请提交工单。

实现过程

HDFS Writer的实现过程如下所示：

1. 根据您指定的path，创建一个HDFS文件系统上不存在的临时目录。

   创建规则：`path_随机`。

2. 将读取的文件写入这个临时目录。
3. 全部写入后，将临时目录下的文件移动到您指定的目录（在创建文件时保证文件名不重复）。
4. 删除临时目录。如果在此过程中，发生网络中断等情况造成无法与HDFS建立连接，需要您手动删除已经写入的文件和临时目录。

说明:

数据同步需要使用Admin账号，并且有访问相应文件的读写权限。

功能限制

- 目前HDFS Writer仅支持TextFile、ORCFile和ParquetFile三种格式的文件，且文件内容存放的必须是一张逻辑意义上的二维表。
- 由于HDFS是文件系统，不存在schema的概念，因此不支持对部分列写入。
- 目前不支持DECIMAL、BINARY、ARRAYS、MAPS、STRUCTS和UNION等Hive数据类型。
- 对于Hive分区表目前仅支持一次写入单个分区。
- 对于TextFile，需要保证写入HDFS文件的分隔符与在Hive上创建表时的分隔符一致，从而实现写入HDFS数据与Hive表字段关联。
- 目前插件中的Hive版本为1.1.1，Hadoop版本为2.7.1（Apache为适配JDK1.7）。在Hadoop2.5.0、Hadoop2.6.0和Hive1.2.0测试环境中写入正常。

数据类型转换

目前HDFS Writer支持大部分Hive类型，请注意检查您的数据类型。

HDFS Writer针对Hive数据类型的转换列表，如下所示。

说明:

column的配置需要和Hive表对应的列类型保持一致。

类型分类	数据库数据类型
整数类	TINYINT、SMALLINT、INT和BIGINT
浮点类	FLOAT和DOUBLE
字符串类	CHAR、VARCHAR和STRING
布尔类	BOOLEAN
日期时间类	DATE和TIMESTAMP

参数说明

参数	描述	必选	默认值
defaultFS	Hadoop HDFS文件系统namenode节点地址，例如hdfs://127.0.0.1:9000。默认资源组不支持Hadoop高级参数HA的配置，请#unique_33。	是	无
fileType	文件的类型，目前仅支持您配置为text、orc和parquet。 · text：表示TextFile文件格式。 · orc：表示ORCFile文件格式。 · parquet：表示普通parquet file文件格式。	是	无
path	存储到Hadoop HDFS文件系统的路径信息，HDFS Writer会根据并发配置在path目录下写入多个文件。 为了与Hive表关联，请填写Hive表在HDFS上的存储路径。例如Hive上设置的数据仓库的存储路径为/user/hive/warehouse/，已建立数据库test表hello，则对应的存储路径为/user/hive/warehouse/test.db/hello。	是	无
fileName	HDFS Writer写入时的文件名，实际执行时会在该文件名后添加随机的后缀作为每个线程写入实际文件名。	是	无

参数	描述	必选	默认值
column	写入数据的字段，不支持对部分列写入。 为了与Hive中的表关联，需要指定表中所有字段名和字段类型，其中name指定字段名，type指定字段类型。 您可以指定column字段信息，配置如下： `"column": [ { "name": "userName", "type": "string" }, { "name": "age", "type": "long" } ]`	是（如果filetype为parquet，此项无需填写）	无
writeMode	HDFS Writer写入前数据清理处理模式。 · append：写入前不做任何处理，数据集成HDFS Writer直接使用filename写入，并保证文件名不冲突。 · nonConflict：如果目录下有fileName前缀的文件，直接报错。 说明： Parquet格式文件不支持Append，所以只能是noConflict。	是	无
fieldDelimiter	HDFS Writer写入时的字段分隔符，需要您保证与创建的Hive表的字段分隔符一致，否则无法在Hive表中查到数据。	是（如果filetype为parquet，此项无需填写）	无
compress	HDFS文件压缩类型，默认不填写，则表示没有压缩。 其中text类型文件支持gzip和bzip2压缩类型，orc类型文件支持SNAPPY压缩类型（需要您安装SnappyCodec）。	否	无
encoding	写文件的编码配置。	否	无压缩

参数	描述	必选	默认值

parquetSchema	写Parquet格式文件时的必填项，用来描述目标文件的结构，所以此项当且仅当fileType为parquet时生效。格式如下：
	`message MessageType名 {`
	`是否必填，数据类型，列名;`
	`.....;`
	`}`
	配置项说明如下：
	· MessageType名：填写名称。
	· 是否必填：required表示非空，optional表示可为空。推荐全填optional。
	· 数据类型：Parquet文件支持BOOLEAN、INT32、INT64、INT96、FLOAT、DOUBLE、BINARY（如果是字符串类型，请填BINARY）和FIXED_LEN_BYTE_ARRAY等类型。
	说明：
	每行列设置必须以分号结尾，最后一行也要写上分号。
	示例如下。
	`message m {`
	`optional int64 id;`
	`optional int64 date_id;`
	`optional binary datetimestring;`
	`optional int32 dspId;`
	`optional int32 advertiserId;`
	`optional int32 status;`
	`optional int64 bidding_req_num;`
	`optional int64 imp;`
	`optional int64 click_num;`
	`}`
		否	无

hadoopConfig	hadoopConfig中可以配置与Hadoop相关的一些高级参数，例如HA的配置。默认资源组不支持Hadoop高级参数HA的配置，请#unique_33。
	`"hadoopConfig":{`
	`"dfs.nameservices": "testDfs",`
	`"dfs.ha.namenodes.testDfs": "namenode1,namenode2",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode1": "",`
	`"dfs.namenode.rpc-address.youkuDfs.namenode2": "",`
	`"dfs.client.failover.proxy.provider.testDfs": "org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider`
		否	无

参数	描述	必选	默认值
kerberosKeytabFilePath	Kerberos认证keytab文件的绝对路径。	如果haveKerberos为true，则必选。	无
kerberosPrincipal	Kerberos认证Principal名，如****/hadoopclient@**.**** 。如果haveKerberos为true，则必选。 说明: 由于Kerberos需要配置keytab认证文件的绝对路径，您需要在自定义资源组上使用此功能。配置示例如下: `"haveKerberos":true, "kerberosKeytabFilePath":"/opt/datax/**.keytab", "kerberosPrincipal":"**/hadoopclient@**.**"`	否	无

向导开发介绍

暂不支持向导开发模式开发。

脚本开发介绍

脚本配置示例如下，详情请参见上述参数说明。

```
{
    "type": "job",
    "version": "2.0",//版本号。
    "steps": [
        { //以下为Writer模板，您可以查找相应数据源的写插件文档。
            "stepType": "stream",
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "hdfs",//插件名
            "parameter": {
                "path": "",//存储到Hadoop HDFS文件系统的路径信息。
                "fileName": "",//HDFS Writer写入时的文件名。
                "compress": "",//HDFS文件压缩类型。
                "datasource": "",//数据源。
                "column": [
                    {
                        "name": "col1",//字段名。
                        "type": "string"//字段类型。
                    },
                    {
                        "name": "col2",
                        "type": "int"
                    },
```

```
                        {
                            "name": "col3",
                            "type": "double"
                        },
                        {
                            "name": "col4",
                            "type": "boolean"
                        },
                        {
                            "name": "col5",
                            "type": "date"
                        }
                    ],
                    "writeMode": "",//写入模式。
                    "fieldDelimiter": ",",//列分隔符。
                    "encoding": "",//编码格式。
                    "fileType": "text"//文本类型。
                },
                "name": "Writer",
                "category": "writer"
            }
        ],
        "setting": {
            "errorLimit": {
                "record": ""//错误记录数。
            },
            "speed": {
                "concurrent": 3,//作业并发数。
                "throttle": false,//false代表不限流，下面的限流的速度不生效，
true代表限流。
            }
        },
        "order": {
            "hops": [
                {
                    "from": "Reader",
                    "to": "Writer"
                }
            ]
        }
}
```

2.3.2.9 配置MaxCompute Writer

本文将为您介绍MaxCompute Writer支持的数据类型、字段映射和数据源等参数及配置示例。

MaxCompute Writer插件用于实现向MaxCompute中插入或更新数据，主要适用于开发者，可以将业务数据导入MaxCompute，适合于TB、GB等数量级的数据传输。

说明:

开始配置MaxCompute Writer插件前，请首先配置好数据源，详情请参见#unique_70。MaxCompute的详情请参见#unique_162。

根据您配置的源头项目/表/分区/表字段等信息，在底层实现上，可以通过Tunnel将数据写入MaxCompute。常用的Tunnel命令请参见#unique_163。

对于MySQL、MaxCompute等强Schema类型的存储，数据集成会将源数据逐步读取到内存中，并根据目的端数据源的类型，将源头数据转换为目的端对应的格式，写入目的端存储。

如果数据转换失败，或数据写出至目的端数据源失败，则将数据作为脏数据，您可以配合脏数据限制阈值使用。

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须与添加的数据源名称保持一致。	是	无
table	写入的数据表的表名称（大小写不敏感），不支持填写多张表。	是	无
partition	需要写入数据表的分区信息，必须指定到最后一级分区。例如把数据写入一个三级分区表，必须配置到最后一级分区，例如pt=20150101, type=1, biz=2。 · 对于非分区表，该值务必不要填写，表示直接导入至目标表。 · MaxCompute Writer不支持数据路由写入，对于分区表请务必保证写入数据到最后一级分区。	如果表为分区表，则必填。如果表为非分区表，则不能填写。	无
column	需要导入的字段列表。当导入全部字段时，可以配置为"column": ["*"]。当需要插入部分MaxCompute列，则填写部分列，例如"column": ["id","name"]。 · MaxCompute Writer支持列筛选、列换序。例如一张表中有a、b和c三个字段，您只同步c和b两个字段，则可以配置为"column": ["c","b"]，在导入过程中，字段a自动补空，设置为null。 · column必须显示指定同步的列集合，不允许为空。	是	无

参数	描述	是否必选	默认值
truncate	通过配置"truncate": "true"保证写入的幂等性。即当出现写入失败再次运行时，MaxCompute Writer将清理前述数据，并导入新数据，可以保证每次重跑之后的数据都保持一致 。 因为利用MaxCompute SQL进行数据清理工作，SQL无法做到原子性，所以truncate选项不是原子操作。因此当多个任务同时向一个Table/Partition清理分区时，可能出现并发时序问题，请务必注意。 针对这类问题，建议您尽量不要多个作业DDL同时操作同一个分区，或者在多个并发作业启动前，提前创建分区。	是	无

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。

配置	说明
分区信息	如果是指定所有的列，可以在column配置，例如"column":[""]。partition支持配置多个分区和通配符的配置方法。 · "partition":"pt=20140501/ds=*"代表ds中所有的分区。 · "partition":"pt=top?"中的?代表前面的字符是否存在，指pt=top和pt=to两个分区。 可以输入您要同步的分区列，如分区列有pt等。例如MaxCompute的分区为pt=${bdp.system.bizdate}，您可以直接将您的分区的名称pt添加到源头表字段中。可能会有未识别的标志，可以直接忽略进行下一步。 如果要同步所有的分区，将前面显示的分区值配置成为pt=${*}。如果同步某个分区，可以直接选择您要同步的时间值。
清理规则	· 写入前清理已有数据：导数据之前，清空表或者分区的所有数据，相当于insert overwrite。 · 写入前保留已有数据：导数据之前，不清理任何数据，每次运行数据都是追加进去的，相当于insert into。 说明: · MaxCompute通过Tunnel服务读取数据，同步任务本身不支持数据过滤，需要读取某一个表或分区内的数据。 · MaxCompute通过Tunnel服务写出数据，没有使用MaxCompute的insert SQL语句进行数据写出。数据同步任务执行成功后，方可对表可见完整数据。请注意建立好任务依赖关系。
空字符串是否作null	默认值为是。

2. 字段映射，即上述参数说明中的column。

左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

脚本配置样例如下，详情请参见上述参数说明。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        {//下面是关于Writer的模板，您可以查找相应数据源的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"odps",//插件名。
            "parameter":{
                "partition":"",//分区信息。
                "truncate":true,//清理规则。
                "compress":false,//是否压缩。
                "datasource":"odps_first",//数据源名。
            "column": [//源端列名。
                "id",
                "name",
                "age",
                "sex",
                "salary",
                "interest"
                ],
                "emptyAsNull":false,//空字符串是否作为null。
```

```
                "table":""//表名。
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数，表示脏数据的最大容忍条数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

如果您需要指定MaxCompute的Tunnel Endpoint，可以通过脚本模式手动配置数据源：将上述示例中的`"datasource":""`,替换为数据源的具体参数，示例如下：

```
"accessId":"***********",
 "accessKey":"*********",
 "endpoint":"http://service.eu-central-1.maxcompute.aliyun-inc.com/api
",
 "odpsServer":"http://service.eu-central-1.maxcompute.aliyun-inc.com/
api",
"tunnelServer":"http://dt.eu-central-1.maxcompute.aliyun.com",
"project":"**********",
```

补充说明

- 关于列筛选的问题

  通过配置MaxCompute Writer，可以实现MaxCompute本身不支持的列筛选、重排序和补空等操作。例如需要导入的字段列表，当导入全部字段时，可以配置为`"column": ["*"]`。

  MaxCompute表有a、b和c三个字段，您只同步c和b两个字段，可以将列配置为`"column": ["c","b"]`，表示会把Reader的第一列和第二列导入MaxCompute的c字段和b字段，而MaxCompute表中新插入的a字段会被置为null。

- 列配置错误的处理

  为保证写入数据的可靠性，避免多余列数据丢失造成数据质量故障。对于写入多余的列，MaxCompute Writer将报错。例如MaxCompute表字段为a、b和c，如果MaxCompute Writer写入的字段多于3列，MaxCompute Writer将报错。

- 分区配置注意事项

  MaxCompute Writer仅提供写入到最后一级分区的功能，不支持写入按照某个字段进行分区路由等功能。假设表一共有3级分区，那么在分区配置中就必须指明写入到某个三级分区，例如把数据写入一个表的第三级分区，可以配置为`pt=20150101, type=1, biz=2`，但不能配置为`pt=20150101, type=1`或者`pt=20150101`。

- 任务重跑和failover

  MaxCompute Writer通过配置`"truncate": true`，保证写入的幂等性。即当出现写入失败再次运行时，MaxCompute Writer将清理前述数据，并导入新数据，这样可以保证每次重跑之后的数据都保持一致。如果在运行过程中，因为其他的异常导致了任务中断，便不能保证数据的原子性，数据不会回滚也不会自动重跑，需要您利用幂等性这一特点重跑，以确保数据的完整性。

   说明:
  truncate为true的情况下，会将指定分区或表的数据全部清理，请谨慎使用。

2.3.2.10 配置Memcache（OCS） Writer

本文将为您介绍Memcache（OCS） Writer支持的数据类型、字段映射和数据源等参数及配置示例。

云数据库Memcache版（ApsaraDB for Memcache，原简称OCS）是一种高性能、高可靠、可平滑扩容的分布式内存数据库服务。基于飞天分布式系统及高性能存储，并提供了双机热备、故障恢复、业务监控和数据迁移等方面的全套数据库解决方案。

云数据库Memcache版支持即开即用的方式快速部署，对于动态Web、APP应用，可以通过缓存服务减轻对数据库的压力，从而提高网站整体的响应速度。

云数据库Memcache版与本地MemCache的异同点如下：

- 相同点：云数据库Memcache版兼容Memcached协议，与您的环境兼容，可以直接用于云数据库Memcache版服务。
- 不同点：云数据库Memcache版的硬件和数据部署在云端，有完善的基础设施、网络安全保障和系统维护等服务。所有服务只需要按量付费即可。

Memcache Writer基于Memcached协议的数据写入Memcache通道。

Memcache Writer目前支持一种格式的写入方式，不同写入方式的类型转换方式不一致。

- text：Memcache Writer将来源数据序列化为STRING类型格式，并使用您的`fieldDelimiter`作为间隔符。
- binary：目前暂不支持。

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
writeMode	Memcache Writer写入方式，具体如下： · set：存储这个数据。 · add：存储这个数据，当且仅当这个key不存在时（目前不支持）。 · replace：存储这个数据，当且仅当这个key存在时（目前不支持）。 · append：将数据存放在已存在的key对应的内容后面，忽略exptime（目前不支持）。 · prepend：将数据存放在已存在的key对应的内容的前面，忽略 exptime（目前不支持）。	是	无

参数	描述	是否必选	默认值

writeFormat	Memcache Writer写出数据的格式，目前仅支持TEXT数据写入方式。
	TEXT：将源端数据序列化为文本格式，其中第一个字段作为Memcache写入的key，后续所有字段序列化为String类型，使用您指定的fieldDelimiter作为间隔符，将文本拼接为完整的字符串再写入Memcache。
	例如源头数据如下所示。
	`| ID | NAME | COUNT|`
	`| ---- |:------|:-----|`
	`| 23 | "CDP" | 100 |`
	如果您指定fieldDelimiter为\^，则写入Memcache的格式如下。
	`| KEY (OCS) | VALUE(OCS) |`
	`| --------- |:---------- |`
	`| 23 | CDP\^100 |`
		否	无

expireTime	Memcache值缓存失效时间，目前MemCache支持两类过期时间。 · Unix时间（自1970.1.1开始到现在的秒数），该时间指定了到未来某个时刻的数据失效。 · 相对当前时间的秒数，该时间指定了从现在开始多长时间后数据失效。 说明: 如果过期时间的秒数大于60*60*24*30（即30天），则服务端认为是Unix时间。	否	0，0永久有效
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据同步系统与MySQL的网络交互次数，并提升整体吞吐量。如果该值设置过大，会导致数据同步运行进程OOM异常。	否	1,024

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个写入Memcache的数据同步作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        { //下面是关于Writer的模板，您可以查找相应数据源的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"ocs",//插件名
            "parameter":{
                "writeFormat":"text",//Memcache Writer写出数据格式。
                "expireTime":1000,//Memcache值缓存失效时间。
                "indexes":0,
                "datasource":"",//数据源。
                "writeMode":"set",//写入模式。
                "batchSize":"256"//一次性批量提交的记录数大小。
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

2.3.2.11 配置MongoDB Writer

本文为您介绍MongoDB Writer支持的数据类型、写入方式、字段映射和数据源等参数和配置示例。

MongoDB Writer插件利用MongoDB的Java客户端MongoClient进行MongoDB的写操作。最新版本的Mongo已经将DB锁的粒度从DB级别降低到Document级别，配合MongoDB强大的索引功能，基本可以满足数据源向MongoDB写入数据的需求。针对数据更新的需求，通过配置业务主键的方式也可以实现。

说明:

- 在开始配置MongoDB Writer插件前，请首先配置好数据源，详情请参见#unique_207。
- 如果您使用的是云数据库MongoDB版，MongoDB默认会有root账号。
- 出于安全策略的考虑，数据集成仅支持使用 MongoDB数据库对应账号进行连接，您添加使用MongoDB数据源时，也请避免使用root作为访问账号。

MongoDB Writer通过数据集成框架获取Reader生成的协议数据，然后将支持的类型通过逐一判断转换成MongoDB支持的类型。数据集成本身不支持数组类型，但MongoDB支持数组类型，并且数组类型的索引很强大。

为了使用MongoDB的数组类型，您可以通过参数的特殊配置，将字符串可以转换成MongoDB中的数组，类型转换之后，便可并行写入MongoDB。

类型转换列表

MongoDB Writer支持大部分MongoDB类型，但也存在部分没有支持的情况，请注意检查您的类型。

MongoDB Writer针对MongoDB类型的转换列表，如下所示。

类型分类	MongoDB数据类型
整数类	INT和LONG
浮点类	DOUBLE
字符串类	STRING和ARRAY
日期时间类	DATE
布尔型	BOOL
二进制类	BYTES

说明:
此处DATE类型，写入到MongoDB后即为DATETIME类型。

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无

参数	描述	是否必选	默认值
collectionName	MonogoDB的集合名。	是	无
column	MongoDB的文档列名，配置为数组形式表示MongoDB的多个列。 · name：Column的名字。 · type：Column的类型。 · splitter：特殊分隔符，当且仅当要处理的字符串要用分隔符分隔为字符数组Array时，才使用此参数。通过此参数指定的分隔符，将字符串分隔存储到MongoDB的数组中。	是	无
writeMode	指定了传输数据时是否覆盖的信息。 · isReplace：当设置为true时，表示针对相同的replaceKey做覆盖操作。当设置为false时，表示不覆盖。 · replaceKey：replaceKey指定了每行记录的业务主键，用来做覆盖时使用（不支持replaceKey为多个键，一般是指Monogo中的主键）。	否	无

参数	描述	是否必选	默认值

preSql	表示数据同步写出MongoDB前的前置操作，例如清理历史数据等。如果preSql为空，表示没有配置前置操作。配置preSql时，需要确保preSql符合JSON语法要求。preSql的格式要求如下：
	· 需要配置type字段，表示前置操作类别，支持drop和remove，例如`"preSql":{"type":"remove"}`。
	- drop：表示删除集合和集合内的数据，collectionName参数配置的集合即是待删除的集合。
	- remove：表示根据条件删除数据。
	- json：您可以通过JSON控制待删除的数据条件，例如`"preSql":{"type":"remove", "json":"{'operationTime':{'$gte':ISODate('${last_day}T00:00:00.424+0800')}}"}`。此处的`${last_day}`为DataWorks调度参数，格式为`$[yyyy-mm-dd]`。您可以根据需要具体使用其他MongoDB支持的条件操作符号（$gt、$lt、$gte和$lte等）、逻辑操作符（and和or等）或函数（max、min、sum、avg和ISODate等），详情请参见MongoDB查询语法。
	数据集成通过如下MongoDB标准API执行您的数据，删除query：
	`query=(BasicDBObject) com.mongodb.util.JSON.parse(json);`
	`col.deleteMany(query);`
	说明：
	如果您需要条件删除数据，建议您优先使用JSON配置形式。
	- item：您可以在item中配置数据过滤的列名（name）、条件（condition）和列值（value）。例如`"preSql":{"type":"remove","item":[{"name":"pv","value":"100","condition":"$gt"},{"name":"pid","value":"10"}]}`。
	数据集成会基于您配置的item条件项，构造查询query条件，进而通过MongoDB标准API执行删除。例如`col.deleteMany(query);`。
	· 不识别的preSql，不需进行任何前置删除操作。
		否	无

向导开发介绍

暂不支持向导开发模式。

脚本开发介绍

配置写入MongoDB的数据同步作业，详情请参见上述参数说明。

```
{
    "type": "job",
    "version": "2.0",//版本号
    "steps": [//下面是关于Reader的模板，可以查看相应的读插件文档。
        {
            "stepType": "stream",
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "mongodb",//插件名
            "parameter": {
                "datasource": "",//数据源名
                "column": [
                    {
                        "name": "name",//列名
                        "type": "string"//数据类型
                    },
                    {
                        "name": "age",
                        "type": "int"
                    },
                    {
                        "name": "id",
                        "type": "long"
                    },
                    {
                        "name": "wealth",
                        "type": "double"
                    },
                    {
                        "name": "hobby",
                        "type": "array",
                        "splitter": " "
                    },
                    {
                        "name": "valid",
                        "type": "boolean"
                    },
                    {
                        "name": "date_of_join",
                        "format": "yyyy-MM-dd HH:mm:ss",
                        "type": "date"
                    }
                ],
                "writeMode": {//写入模式
                    "isReplace": "true",
                    "replaceKey": "id"
                },
                "collectionName": "datax_test"//连接名称
            },
            "name": "Writer",
            "category": "writer"
```

```
            }
        ],
        "setting": {
            "errorLimit": {//错误记录数
                "record": "0"
            },
            "speed": {
                "jvmOption": "-Xms1024m -Xmx1024m",
                "throttle": true,//false代表不限流，下面的限流的速度不生效，true
代表限流。
                "concurrent": 1,//作业并发数
                "mbps": "1"//限流的速度
            }
        },
        "order": {
            "hops": [//从reader同步writer
                {
                    "from": "Reader",
                    "to": "Writer"
                }
            ]
        }
}
```

2.3.2.12 配置MySQL Writer

本文将为您介绍MySQL Writer支持的数据类型、字段映射和数据源等参数及配置示例。

MySQL Writer插件实现了写入数据至MySQL数据库目标表的功能。在底层实现上， MySQL Writer通过JDBC连接远程MySQL数据库，并执行相应的`insert into`或`replace into`语句，将数据写入MySQL。数据库本身采用InnoDB引擎，以将数据分批次提交入库。

 说明:

- 开始配置MySQL Writer插件前，请首先配置好数据源，详情请参见#unique_209。
- 目前MySQL Writer暂不支持MySQL 8.0及以上版本。

MySQL Writer作为数据迁移工具，为数据库管理员等用户提供服务。根据您配置的`writeMode`，通过数据同步框架获取Reader生成的协议数据。

 说明:

整个任务必须具备`insert/replace into`的权限。您可以根据配置任务时，在`preSql`和`postSql`中指定的语句，判断是否需要其他权限。

类型转换列表

目前MySQL Writer支持大部分MySQL类型，但也存在个别类型没有支持的情况，请注意检查您的数据类型。

MySQL Writer针对MySQL类型的转换列表，如下所示。

类型分类	MySQL数据类型
整数类	INT、TINYINT、SMALLINT、MEDIUMINT、BIGINT和YEAR
浮点类	FLOAT、DOUBLE和DECIMAL
字符串类	VARCHAR、CHAR、TINYTEXT、TEXT、MEDIUMTEXT和LONGTEXT
日期时间类	DATE、DATETIME、TIMESTAMP和TIME
布尔型	BOOL
二进制类	TINYBLOB、MEDIUMBLOB、BLOB、LONGBLOB和VARBINARY

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无
writeMode	选择导入模式，可以支持insert into、on duplicate key update和replace into三种方式。 · insert into：当主键/唯一性索引冲突时会写不进去冲突的行，以脏数据的形式体现。 · on duplicate key update：没有遇到主键/唯一性索引冲突时，与insert into行为一致。冲突时会用新行替换已经指定的字段的语句，写入数据至MySQL。 · replace into：没有遇到主键/唯一性索引冲突时，与insert into行为一致。冲突时会先删除原有行，再插入新行。即新行会替换原有行的所有字段。	否	insert
column	目标表需要写入数据的字段，字段之间用英文所逗号分隔，例如"column": ["id", "name", "age"]。 如果要依次写入全部列，使用*表示， 例如"column": ["*"]。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如清除旧数据。 说明: 当有多条SQL语句时，不支持事务。	否	无

参数	描述	必选	默认值
postSql	执行数据同步任务之后执行的SQL语句，目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如加上某一个时间戳。 说明: 当有多条SQL语句时，不支持事务。	否	无
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据同步系统与MySQL的网络交互次数，并提升整体吞吐量。如果该值设置过大，会导致数据同步运行进程OOM异常。	否	1,024

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。
导入前准备语句	即上述参数说明中的preSql，输入执行数据同步任务之前率先执行的SQL语句。
导入后完成语句	即上述参数说明中的postSql，输入执行数据同步任务之后执行的SQL语句。
主键冲突	即上述参数说明中的writeMode，可以选择需要的导入模式。

2. 字段映射，即上述参数说明中的column。

左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，将鼠标放至需要删除的字段上，即可单击删除按钮进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

脚本配置样例如下，详情请参见上述参数说明。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[ //下面是关于Writer的模板，您可以查找相应数据源的写插件文档。
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"mysql",//插件名。
            "parameter":{
                "postSql":[],//导入后的准备语句。
                "datasource":"",//数据源。
                "column":[//列名。
                    "id",
                    "value"
                ],
                "writeMode":"insert",//写入模式。
                "batchSize":1024,//一次性批量提交的记录数大小。
                "table":"",//表名。
                "preSql":[]//导入前的准备语句。
            },
            "name":"Writer",
            "category":"writer"
```

```
            }
        ],
        "setting":{
            "errorLimit":{//错误记录数。
                "record":"0"
            },
            "speed":{
                "throttle":false,//是否限流。
                "concurrent":1,//并发数。
            }
        },
        "order":{
            "hops":[
                {
                    "from":"Reader",
                    "to":"Writer"
                }
            ]
        }
}
```

2.3.2.13 配置Oracle Writer

本文将为您介绍Oracle Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置示例。

Oracle Writer插件实现了写入数据到Oracle主库的目标表的功能。在底层实现上，Oracle Writer通过JDBC连接远程Oracle数据库，并执行相应的`insert into...`SQL语句，将数据写入Oracle。

说明:
开始配置Oracle Writer插件前，请首先配置好数据源，详情请参见#unique_52。

Oracle Writer面向ETL开发工程师，使用Oracle Writer从数仓导入数据至Oracle。同时Oracle Writer也可以作为数据迁移工具，为数据库管理员等用户提供服务。

Oracle Writer通过数据同步框架获取Reader生成的协议数据，然后通过JDBC连接远程Oracle数据库，并执行相应的SQL语句，将数据写入Oracle。

类型转换列表

Oracle Writer支持大部分Oracle类型，但也存在个别类型没有支持的情况，请注意检查您的数据类型。

Oracle Writer针对Oracle类型的转换列表，如下所示。

类型分类	Oracle数据类型
整数类	NUMBER、RAWID、INTEGER、INT和SMALLINT
浮点类	NUMERIC、DECIMAL、FLOAT、DOUBLE PRECISIOON和REAL

类型分类	Oracle数据类型
字符串类	LONG、CHAR、NCHAR、VARCHAR、VARCHAR2、NVARCHAR2、CLOB、NCLOB、CHARACTER、CHARACTER VARYING、CHAR VARYING、NATIONAL CHARACTER、NATIONAL CHAR、NATIONAL CHARACTER VARYING、NATIONAL CHAR VARYING和NCHAR VARYING
日期时间类	TIMESTAMP和DATE
布尔型	BIT和BOOL
二进制类	BLOB、BFILE、RAW和LONG RAW

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	目标表名称，如果表的schema信息和上述配置username不一致，请使用schema.table的格式填写table信息。	是	无
writeMode	选择导入模式，可以支持insert into、on duplicate key update和replace into三种方式。 · insert into：当主键/唯一性索引冲突时会写不进去冲突的行，以脏数据的形式体现。 · on duplicate key update：没有遇到主键/唯一性索引冲突时，与insert into行为一致。冲突时会用新行替换已经指定的字段的语句，写入数据至MySQL。 · replace into：没有遇到主键/唯一性索引冲突时，与insert into行为一致。冲突时会先删除原有行，再插入新行。即新行会替换原有行的所有字段。	否	insert
column	目标表需要写入数据的字段，字段之间用英文逗号分隔。例如"column": ["id","name","age"]。如果要依次写入全部列，使用*表示。例如"column":["*"]。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如清除旧数据。	否	无
postSql	执行数据同步任务之后执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如加上某一个时间戳。	否	无

参数	描述	是否必选	默认值
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据同步系统与MySQL的网络交互次数，并提升整体吞吐量。如果该值设置过大，会导致数据同步运行进程OOM异常。	否	1,024

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。



配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。
导入前准备语句	即上述参数说明中的preSql，输入执行数据同步任务之前率先执行的SQL语句。
导入后完成语句	即上述参数说明中的postSql，输入执行数据同步任务之后执行的SQL语句。
主键冲突	即上述参数说明中的writeMode，可以选择需要的导入模式。

2. 字段映射，即上述参数说明中的column。

左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

配置一个写入Oracle的作业。

```
{
    "type":"job",
    "version":"2.0",//版本号。
    "steps":[
        { //下面是关于Writer的模板，您可以查找相应数据源的写插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"oracle",//插件名。
            "parameter":{
                "postSql":[],//执行数据同步任务之后执行的SQL语句。
                "datasource":"",
                "session":[],//数据库连接会话参数。
                "column":[//字段。
                    "id",
                    "name"
                ],
                "encoding":"UTF-8",//编码格式。
                "batchSize":1024,//一次性批量提交的记录数大小。
                "table":"",//表名。
                "preSql":[]//执行数据同步任务之前执行的SQL语句。
            },
            "name":"Writer",
```

```
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数。
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//并发数。
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

2.3.2.14 配置OSS Writer

本文为您介绍OSS Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置示例。

OSS Writer插件提供了向OSS写入类CSV格式的一个或者多个表文件的功能，写入的文件个数和您的任务并发及同步的文件数有关。

> 说明:
> 开始配置OSS Writer插件前，请首先配置好数据源，详情请参见#unique_80。

写入OSS内容存放的是一张逻辑意义上的二维表，例如CSV格式的文本信息。如果您想对OSS产品有更深入的了解，请参见OSS产品概述。

OSS Java SDK的详细介绍，请参见阿里云OSS Java SDK。

OSS Writer实现了从数据同步协议转为OSS中的文本文件功能，OSS本身是无结构化数据存储，目前OSS Writer支持的功能如下所示：

- 支持且仅支持写入文本文件，并要求文本文件中的shema为一张二维表。
- 支持类CSV格式文件，自定义分隔符。
- 支持多线程写入，每个线程写入不同子文件。
- 文件支持滚动，当文件大于某个size值时，支持文件切换。当文件大于某个行数值时，支持文件切换。

OSS Writer暂时不能实现以下功能：

- 单个文件不能支持并发写入。
- OSS本身不提供数据类型，OSS Writer均以STRING类型写入OSS对象。

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
object	OSS Writer写入的文件名，OSS使用文件名模拟目录的实现。例如数据同步到OSS数据源中的bucket为test118的test文件夹。  则Object只需填写test，不要带上bucket名称，同步到OSS端的文件名与源端填写的文件名相同。 使用"object": "test/DI"，写入的Object以test/DI开头。test为文件夹，DI为文件名称的开头，后缀随机添加字符串，/是OSS模拟目录的分隔符。 说明: OSS object前缀支持配置正则。	是	无
writeMode	OSS Writer写入前数据清理处理。 · truncate：写入前清理Object名称前缀匹配的所有Object。例如"object":"abc"，将清理所有abc开头的Object。 · append：写入前不进行任何处理，数据集成OSS Writer直接使用Object名称写入，并使用随机UUID的后缀名来保证文件名不冲突。例如您指定的Object名为数据集成，实际写入为DI_****_****_****。 · nonConflict：如果指定路径出现前缀匹配的Object，直接报错。例如"object":"abc"，如果存在abc123的Object，将直接报错。	是	无

参数	描述	是否必选	默认值
fileFormat	文件写出的格式，包括csv和text两种。 · csv是严格的csv格式，如果待写数据包括列分隔符，则会按照csv的转义语法转义，转义符号为双引号（"）。 · text格式是用列分隔符简单分割待写数据，对于待写数据包括列分隔符情况下不做转义。	否	text
fieldDelimiter	读取的字段分隔符。	否	,
encoding	写出文件的编码配置。	否	utf-8
nullFormat	文本文件中无法使用标准字符串定义null（空指针），数据同步系统提供nullFormat定义哪些字符串可以表示为null。例如您配置nullFormat="null"，那么如果源头数据是"null"，数据同步系统会视作null字段。	否	无
header（高级配置，向导模式不支持）	OSS写出时的表头，例如['id', 'name', 'age']。	否	无
maxFileSize（高级配置，向导模式不支持）	OSS写出时单个Object文件的最大值，默认为10000*10MB，类似于log4j日志打印时根据日志文件大小轮转。OSS分块上传时，每个分块大小为10MB（也是日志轮转文件最小粒度，即小于10MB的maxFileSize会被作为10MB），每个OSS InitiateMultipartUploadRequest支持的分块最大数量为10000。 轮转发生时，Object名字规则是在原有Object前缀加UUID随机数的基础上，拼接_1,_2,_3等后缀。	否	100,000MB
suffix（高级配置，向导模式不支持）	数据同步写出时，生成的文件名后缀，例如配置suffix为.csv，则最终写出的文件名为fileName****.csv。	否	无

向导开发介绍

1. 选择数据源。

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
Object前缀	即上述参数说明中的Object，填写OSS文件夹的路径，其中不要填写bucket的名称。
文本类型	包括csv和text2种文本类型。
列分隔符	即上述参数说明中的fieldDelimiter，默认值为（,）。
编码格式	即上述参数说明中的encoding，默认值为utf-8。
null值	即上述参数说明中的nullFormat，将要表示为空的字段填入文本框，如果源端存在则将对应的部分转换为空。

配置	说明
时间格式	日期类型的数据序列化到Object时的格式，例如 " dateFormat": "yyyy-MM-dd"。
前缀冲突	有同样的文件时，可以选择替换、保留或报错。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。

配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。

3. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。

配置	说明
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

脚本配置示例如下所示，具体参数填写请参见参数说明。

```
{
    "type":"job",
    "version":"2.0",
    "steps":[
        { //下面是关于Reader的模板，可以找相应的读插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"oss",//插件名
            "parameter":{
                "nullFormat":"",//数据同步系统提供nullFormat，定义哪些字符
串可以表示为null。
                "dateFormat":"",//日期格式
                "datasource":"",//数据源
                "writeMode":"",//写入模式
                "encoding":"",//编码格式
                "fieldDelimiter":",//字段分隔符
                "fileFormat":"",//文本类型
                "object":""//Object前缀
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
```

```
}
```

ORC/Parquet文件写入OSS

目前通过复用HDFS Writer的方式完成OSS写ORC/Parquet格式的文件，在OSS Writer已有参数的基础上，增加了Path、FileFormat等扩展配置参数，参数含义请参见配置HDFS Writer。

- 以ORC文件格式写入OSS，示例如下：

```
{
      "stepType": "oss",
      "parameter": {
        "datasource": "",
        "fileFormat": "orc",
        "path": "/tests/case61",
        "fileName": "orc",
        "writeMode": "append",
        "column": [
          {
            "name": "col1",
            "type": "BIGINT"
          },
          {
            "name": "col2",
            "type": "DOUBLE"
          },
          {
            "name": "col3",
            "type": "STRING"
          }
        ],
        "writeMode": "append",
        "fieldDelimiter": "\t",
        "compress": "NONE",
        "encoding": "UTF-8"
      }
    }
```

- 以Parquet文件格式写入OSS，示例如下：

```
{
      "stepType": "oss",
      "parameter": {
        "datasource": "",
        "fileFormat": "parquet",
        "path": "/tests/case61",
        "fileName": "test",
        "writeMode": "append",
        "fieldDelimiter": "\t",
        "compress": "SNAPPY",
        "encoding": "UTF-8",
        "parquetSchema": "message test { required int64 int64_col;
\n required binary str_col (UTF8);\nrequired group params (MAP) {\
nrepeated group key_value {\nrequired binary key (UTF8);\nrequired
 binary value (UTF8);\n}\n}\nrequired group params_arr (LIST) {\n
  repeated group list {\n    required binary element (UTF8);\n  }\n
}\nrequired group params_struct {\n  required int64 id;\n required
binary name (UTF8);\n }\nrequired group params_arr_complex (LIST) {\
n  repeated group list {\n    required group element {\n required
int64 id;\n required binary name (UTF8);\n}\n  }\n}\nrequired group
params_complex (MAP) {\nrepeated group key_value {\nrequired binary
```

```
key (UTF8);\nrequired group value {\n  required int64 id;\n required
 binary name (UTF8);\n  }\n}\n}\nrequired group params_struct_comple
x {\n  required int64 id;\n required group detail {\n  required
int64 id;\n required binary name (UTF8);\n  }\n  }\n}",
        "dataxParquetMode": "fields"
      }
    }
```

2.3.2.15 配置PostgreSQL Writer

本文为您介绍PostgreSQL Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置举例。

PostgreSQL Writer插件实现了向PostgreSQL写入数据。在底层实现上，PostgreSQL Writer通过JDBC连接远程PostgreSQL数据库，并执行相应的SQL语句将数据从PostgreSQL库中SELECT出来，公共云上RDS提供PostgreSQL存储引擎。

说明:
在开始配置PostgreSql Writer插件前，请首先配置好数据源，详情请参见#unique_49。

简而言之，PostgreSQL Writer通过JDBC连接器连接到远程的PostgreSQL数据库，根据您配置的信息生成查询SELECT SQL语句并发送到远程PostgreSQL数据库，并将该SQL执行返回结果使用CDP自定义的数据类型拼装为抽象的数据集，并传递给下游Writer处理。

- 对于您配置的table、column和where等信息，PostgreSQL Writer将其拼接为SQL语句发送到PostgreSQL数据库。
- 对于您配置的querySql信息，PostgreSQL直接将其发送到PostgreSQL数据库。

类型转换列表

PostgreSQL Writer支持大部分PostgreSQL类型，但也存在部分类型没有支持的情况，请注意检查您的类型。

PostgreSQL Writer针对PostgreSQL的类型转换列表，如下所示。

数据集成内部类型	PostgreSQL数据类型
LONG	BIGINT、BIGSERIAL、INTEGER、SMALLINT和SERIAL。
DOUBLE	DOUBLE、PRECISION、MONEY、NUMERIC和REAL。
STRING	VARCHAR、CHAR、TEXT、BIT和INET
DATE	DATE、TIME和TIMESTAMP
BOOLEAN	BOOL
BYTES	BYTEA

 说明:

- 除上述罗列字段类型外，其他类型均不支持。
- MONEY、INET和BIT需要您使用`a_inet::varchar`类似的语法进行转换。

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无
writeMode	选择导入模式，可以支持insert和copy方式。 - insert：执行PostgreSQL的`insert into...values ...`语句，将数据写出到PostgreSQL中。当数据出现主键/唯一性索引冲突时，待同步的数据行写入PostgreSQL失败，当前记录行成为脏数据。建议您优先选择insert模式。 - copy：PostgreSQL提供copy命令，用于表与文件（标准输出，标准输入）之间的相互复制。数据集成支持使用`copy from`，将数据加载到表中。建议您在遇到性能问题时再尝试使用该模式。	否	insert
column	目标表需要写入数据的字段，字段之间用英文逗号分隔。例如`"column":["id","name","age"]`。如果要依次写入全部列，使用（*）表示，例如`"column":["*"]`。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如清除旧数据。	否	无
postSql	执行数据同步任务之后执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如加上某一个时间戳。	否	无
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据集成与PostgreSQL的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据集成运行进程OOM情况。	否	1024

参数	描述	是否必选	默认值
pgType	PostgreSQL特有类型的转化配置，支持bigint[]、double[]、text[]、jsonb和json类型。配置示例如下： `{ "job": { "content": [{ "reader": {...}, "writer": { "parameter": { "column": [// 目标表字段列表 "bigint_arr", "double_arr", "text_arr", "jsonb_obj", "json_obj"], "pgType": { // 特殊的类型设置，key为目标表的字段名，value为字段类型。 "bigint_arr": "bigint[]", "double_arr": "double[]", "text_arr": "text[]", "jsonb_obj": "jsonb", "json_obj": "json" }] } } }] } }`	否	无

向导开发介绍

1. 选择数据源

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，一般填写您配置的数据源名称。
表	即上述参数说明中的table。
导入前准备语句	即上述参数说明中的preSql，输入执行数据同步任务之前率先执行的SQL语句。
导入后完成语句	即上述参数说明中的postSql，输入执行数据同步任务之后执行的SQL语句。
导入模式	即上述参数说明中的writeMode，包括insert和copy两种模式。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应的关系，单击添加一行可增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。



   - 同行映射：单击同行映射可以在同行建立相应的映射关系，请注意匹配数据类型。
   - 自动排版：可以根据相应的规律自动排版。

3. 通道控制

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

脚本配置样例如下，具体参数填写请参见参数说明。

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[ //Reader的模板如下所示，您可以查看相应数据源的Reader文档。
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"postgresql",//插件名
            "parameter":{
                "postSql":[],//执行数据同步任务之后率先执行的SQL语句。
                "datasource":"//数据源
                    "col1",
                    "col2"
                ],
                "table":"",//表名
                "preSql":[]//执行数据同步任务之前率先执行的SQL语句。
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
```

```
        "errorLimit":{
            "record":"0"//错误记录数
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

2.3.2.16 配置Redis Writer

Redis Writer是基于数据集成框架实现的Redis写入插件，可以借助Redis Writer从数仓或者其它数据源导入数据到Redis。Redis Writer与Redis Server之间的交互是基于Jedis实现的，Jedis是Redis官方首选的Java客户端开发包，几乎实现了Redis的所有功能。

Redis（REmote DIctionary Server）是一个支持网络、可基于内存也可持久化的日志型、高性能的key-value存储系统，可用作数据库、高速缓存和消息队列代理。Redis支持较丰富的存储value类型，包括String （字符串）、List（链表） 、Set（集合） 、ZSet（sorted set 有序集合）和 Hash（哈希类型）。Redis详情请参见redis.io。

 说明:

- 开始配置Redis Writer插件前，请首先配置好数据源，详情请参见#unique_94。
- 使用Redis Writer向Redis写入数据时，如果value类型是list，重跑同步任务同步结果不是幂等的。因此如果value类型是list，重跑同步任务时，需要您手动清空Redis上相应的数据。

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无

参数	描述	是否必选	默认值
keyIndexes	keyIndexes表示源端哪几列需要作为key（第一列是从0开始）。如果是第一列和第二列需要组合作为key，那么keyIndexes的值则为[0,1]。  说明: 配置keyIndexes后，Redis Writer会将其余的列作为value。如果您只想同步源表的某几列作为key，某几列作为value，不需要同步所有字段，那么在Reader插件端指定好column进行列筛选即可。	是	无
keyFieldDelimiter	写入Redis的key分隔符。例如key=key1\u0001id，如果key有多个需要拼接时，该值为必填项，如果key只有一个则可以忽略该配置项。	否	\u0001
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据集成与PostgreSQL的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据集成运行进程OOM情况。	否	1000
expireTime	Redis value值缓存失效时间（如果需要永久有效则可以不填该配置项），单位为秒。 · seconds：相对当前时间的秒数，该时间指定了从现在开始多长时间后数据失效。 · unixtime：Unix时间（自1970.1.1开始到现在的秒数），该时间指定了到未来某个时刻数据失效。  说明: 如果过期时间的秒数大于60*60*24*30（即30天），则服务端认为是Unix时间。	否	0（0表示永久有效）
timeout	写入Redis的超时时间，单位为毫秒。	否	30000（即可以cover住30秒的网络断连时间）
dateFormat	写入Redis时，Date的时间格式为"yyyy-MM-dd HH:mm:ss"。	否	无

参数	描述	是否必选	默认值

writeMode	Redis的一大亮点是支持丰富的value类型，包括字符串（string）、字符串列表（list）、字符串集合（set）、有序字符串集合（zset）和哈希（hash）。Redis Writer也支持该五种类型的写入，根据不同的value类型，writeMode配置会略有差异，writeMode详细配置如下。
	
	说明:
	您在配置Redis Writer时，只能配置以下五种类型中的一种。
	
	· 字符串（string）
	
	`"writeMode":{ "type": "string", "mode": "set", "valueFieldDelimiter": "\u0001" }`
	
	配置项说明:
	
	- type
	■ 描述：value为string类型
	■ 必选：是
	- mode
	■ 描述：value为string类型时，写入的模式
	■ 必选：是，可选值为set（存储这个数据，如果已经存在则覆盖）
	- valueFieldDelimiter
	■ 描述：该配置项是考虑了当源数据每行超过两列的情况（如果您的源数据只有两列即key和value时，则可忽略该配置项，不用填写），value类型为string时，value之间的分隔符，比如 value1\u0001value2\u0001value3。
	■ 必选：否
	■ 默认值：\u0001
	· 字符串列表（list）
	
	`"writeMode":{ "type": "list", "mode": "lpush|rpush", "valueFieldDelimiter": "\u0001" }`
	
	配置项说明:
	
	- type
	■ 描述：value为list类型
		否	string

向导开发介绍

暂不支持向导开发模式。

脚本开发介绍

配置写入Redis的数据同步作业，具体参数填写请参见参数说明。

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[
        { //下面是关于Reader的模板，可以找相应的读插件文档
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"redis",//插件名
            "parameter":{
                "expireTime":{//Redis value 值缓存失效时间
                    "seconds":"1000"
                },
                "keyFieldDelimiter":"u0001",//写入 Redis 的 key 分隔符
                "dateFormat":"yyyy-MM-dd HH:mm:ss",//写入 Redis 时,
Date 的时间格式
                "datasource":"",//数据源
                "writeMode":{//写入模式
                    "mode":"",//alue 是某类型时的写入的模式
                    "valueFieldDelimiter":"",//value 之间的分隔符
                    "type":""//value 类型
                },
                "keyIndexes":[//主键索引
                    0,
                    1
                ],
                "batchSize":"1000"//一次性批量提交的记录数大小
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数
        },
        "speed":{
            "throttle":false,////false代表不限流，下面的限流的速度不生效,
true代表限流
            "concurrent":1,//作业并发数
            "dmu":1//DMU值
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
```

```
}Writer"
            }
        ]
    }
}
```

2.3.2.17 配置SQL Server Writer

本文为您介绍SQL Server Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置举例。

SQL Server Writer插件实现了写入数据到SQL Server主库的目标表的功能。在底层实现上，SQL Server Writer通过JDBC连接远程SQL Server数据库，并执行相应的`insert into ...`的SQL语句将数据写入SQL Server，内部会分批次提交入库。

> **说明:**
> 在开始配置SQL Server Writer插件前，请首先配置好数据源，详情请参见配置SQL Server数据源。

SQL Server Writer面向ETL开发工程师，他们使用SQL Server Writer从数仓导入数据到SQL Server。同时SQL Server Writer可以作为数据迁移工具为DBA等用户提供服务。

SQL Server Writer通过数据集成框架获取Reader生成的协议数据，生成`insert into ...`当主键/唯一性索引冲突时，会写不进去冲突的行，出于性能考虑采用了`PreparedStatement+Batch`并且设置了`rewriteBatchedStatements=true`，将数据缓冲到线程上下文Buffer中，当Buffer累计到预定阈值时，才发起写入请求。

> **说明:**
> - 目标表所在数据库必须是主库才能写入数据。
> - 整个任务至少需要具备insert into…的权限，是否需要其他权限，取决于您配置任务时在preSql和postSql中指定的语句。

类型转换列表

SQL Server Writer支持大部分SQL Server类型，但也存在部分类型没有支持的情况，请注意检查您的类型。

SQL Server Writer针对SQL Server的类型转换列表，如下所示。

类型分类	SQL Server数据类型
整数类	BIGINT、INT、SMALLINT和TINYINT
浮点类	FLOAT、DECIMAL、REAL和NUMERIC

类型分类	SQL Server数据类型
字符串类	CHAR、NCHAR、NTEXT、NVARCHAR、TEXT、VARCHAR、NVARCHAR（MAX）和VARCHAR（MAX）
日期时间类	DATE、TIME和DATETIME
布尔类	BIT
二进制类	BINARY、VARBINARY、VARBINARY（MAX）和TIMESTAMP

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无
column	目标表需要写入数据的字段，字段之间用英文逗号分隔。例如"column":["id","name","age"]。如果要依次写入全部列，使用*表示，例如"column":["*"]。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如清除旧数据。	否	无
postSql	执行数据同步任务之后执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如加上某一个时间戳。	否	无
writeMode	选择导入模式，可以支持insert方式。 当主键/唯一性索引冲突时，数据集成视为脏数据但保留原有的数据。	否	insert
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据集成与PostgreSQL的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据集成运行进程OOM情况。	否	1,024

向导开发介绍

1. 选择数据源

   配置同步任务的数据来源和数据去向。



配置	说明
数据源	即上述参数说明中的datasource，通常填写您配置的数据源名称。
表	即上述参数说明中的table。
导入前准备语句	即上述参数说明中的preSql，输入执行数据同步任务之前率先执行的SQL语句。
导入后完成语句	即上述参数说明中的postSql，输入执行数据同步任务之后执行的SQL语句。
主键冲突	即上述参数说明中的writeMode，可以选择需要的导入模式。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应关系。单击添加一行可以增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。

配置	说明
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。
手动编辑源表字段	请手动编辑字段，一行表示一个字段，首尾空行会被采用，其他空行会被忽略。
添加一行	· 可以输入常量，输入的值需要使用英文单引号，如'abc'、'123'等。 · 可以配合调度参数使用，如${bizdate}等。 · 可以输入关系数据库支持的函数，如now()、count(1)等。 · 如果您输入的值无法解析，则类型显示为未识别。

3. 通道控制

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

向导模式介绍

配置写入SQL Server的作业，具体参数填写请参见参数说明。

```
{
    "type":"job",
    "version":"2.0",//版本号
```

```
    "steps":[//下面是关于Reader的模板，您可以查找相应的读插件文档。
        {
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"sqlserver",//插件名
            "parameter":{
                "postSql":[],//执行数据同步任务之后率先执行的SQL语句。
                "datasource":"",//数据源
                "column":[//字段
                    "id",
                    "name"
                ],
                "table":"",//表名
                "preSql":[]//执行数据同步任务之前率先执行的SQL语句。
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数
        },
        "speed":{
            "throttle":false,////false代表不限流，下面的限流的速度不生效,
true代表限流。
            "concurrent":1,//作业并发数
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

2.3.2.18 配置Elasticsearch Writer

本文将为您介绍Elasticsearch Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置示例。

Elasticsearch是遵从Apache开源条款的一款开源产品，是当前主流的企业级搜索引擎。Elasticsearch是一个基于Lucene的搜索和数据分析工具，它提供分布式服务。Elasticsearch核心概念同数据库核心概念的对应关系如下所示。

```
Relational DB（实例）-> Databases（数据库）-> Tables（表）-> Rows（一行数据）-
> Columns（一行数据的一列）
```

```
Elasticsearch        -> Index              -> Types        -> Documents
       -> Fields
```

Elasticsearch中可以有多个索引/数据库，每个索引可以包括多个类型/表，每个类型可以包括多个文档/行，每个文档可以包括多个字段/列。Elasticsearch Writer插件使用Elasticsearch的Rest API接口，批量把从Reader读入的数据写入Elasticsearch中。

参数说明

参数	描述	是否必选	默认值
endpoint	Elasticsearch的连接地址，通常格式为http://xxxx.com:9999。	否	无
accessId	Elasticsearch的username，用于与Elasticsearch建立连接时的鉴权。 说明: AccessID和AccessKey为必填项，如果不填写会产生报错。如果您使用的是自建Elasticsearch，不设置basic验证，不需要账号密码，此处AccessId和AccessKey填写随机值即可。	否	无
accessKey	Elasticsearch的password。	否	无
index	Elasticsearch中的index名。	否	无
indexType	Elasticsearch中index的type名。	否	Elasticsearch
cleanup	是否删除所配索引中已有数据，清理数据的方法为删除并重建对应索引，默认值为false，表示保留已有索引中的数据。	否	false
batchSize	每次批量数据的条数。	否	1,000
trySize	失败后重试的次数。	否	30
timeout	客户端超时时间。	否	600,000
discovery	启用节点发现将轮询并定期更新客户机中的服务器列表。	否	false
compression	HTTP请求，开启压缩。	否	true
multiThread	HTTP请求，是否有多线程。	否	true
ignoreWriteError	忽略写入错误，不重试，继续写入。	否	false

参数	描述	是否必选	默认值
ignoreParseError	忽略解析数据格式错误，继续写入。	否	true
alias	Elasticsearch的别名类似于数据库的视图机制，为索引my_index创建一个别名my_index_alias，对my_index_alias的操作与my_index的操作一致。 配置alias表示在数据导入完成后，为指定的索引创建别名。	否	无
aliasMode	数据导入完成后增加别名的模式，包括append（增加模式）和exclusive（只留这一个）。	否	append
settings	如果待插入目标端数据列类型是array数组类型，则使用指定分隔符（-,-），将源头数据进行拆分写出。示例如下： 源头列是字符串类型数据a-,-b-,-c-,-d，使用分隔符（-,-）拆分后是数组["a", "b", "c", "d"]，最终写出至Elasticsearch对应Filed列中。	否	-,-

参数	描述	是否必选	默认值

column	column用来配置文档的多个字段Filed信息，具体每个字段项可以配置name（名称）、type（类型）等基础配置，以及Analyzer、Format和Array等扩展配置。
	
	Elasticsearch所支持的字段类型如下所示。
	
	- id //type id对应Elasticsearch中的_id，可以理解为唯一主键。写入时，相同id的数据会被覆盖，且不会被索引。
	- string
	- text
	- keyword
	- long
	- integer
	- short
	- byte
	- double
	- float
	- date
	- boolean
	- binary
	- integer_range
	- float_range
	- long_range
	- double_range
	- date_range
	- geo_point
	- geo_shape
	- ip
	- token_count
	- array
	- object
	- nested
	
	· 列类型为text类型时，可以配置analyzer（分词器）、norms和index_options等参数，示例如下。
	
	{
	"name": "col_text",
	"type": "text",
	"analyzer": "ik_max_word"
	}
	
	· 列类型为日期Date类型时，可以配置Format和Timezone参数，分别表示日期序列化格式和时区，示例如下。
	
	{
	"name": "col_date",
	"type": "date",
	"format": "yyyy-MM-dd HH:mm:ss",
	"timezone": "UTC"
	}
	
	· 列类型为地理形状geo_shape时，可以配置tree（geohash或quadtree）、precision
		是	无

参数	描述	是否必选	默认值
actionType	表示Elasticsearch在数据写出时的action类型，目前数据集成支持index和update两种actionType，默认值为index。 · index：底层使用了Elasticsearch SDK的Index.Builder构造批量请求。Elasticsearch index插入的逻辑为：首先判断插入的文档数据中是否指定ID。 - 如果没有指定ID，Elasticsearch会默认生成一个唯一ID。该情况下会直接添加文档至Elasticsearch中。 - 如果已指定ID，会进行更新（替换整个文档），且不支持针对特定Field进行修改。 说明: 此处的更新并非Elasticsearch中的更新（替换部分指定列替换）。 · update：底层使用了Elasticsearch SDK的Update.Builder构造批量请求。Elasticsearch update更新的逻辑为：每次update都会调用InternalEngine中的get方法，来获取整个文档信息，从而实现针对特定字段进行修改。该逻辑导致每次更新都需获取一遍原始文档，对性能有较大影响，但可以更新用户指定的列。如果匹配的文档不存在，则执行文档插入操作。	否	index

脚本开发介绍

脚本配置示例如下，具体参数请参见上文的参数说明。

```
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "concurrent": 1,
```

```
            "throttle": false
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {
                 //下面是关于Reader的模板，您可以查找相应的读插件文档。
            },
            "stepType": "stream"
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {
                "endpoint": "http://xxxx.com:9999",
                "accessId": "xxxx",
                "accessKey": "yyyy",
                "index": "test-1",
                "type": "default",
                "cleanup": true,
                "settings": {
                    "index": {
                        "number_of_shards": 1,
                        "number_of_replicas": 0
                    }
                },
                "discovery": false,
                "batchSize": 1000,
                "splitter": ",",
                "column": [
                    {
                        "name": "pk",
                        "type": "id"
                    },
                    {
                        "name": "col_ip",
                        "type": "ip"
                    },
                    {
                        "name": "col_double",
                        "type": "double"
                    },
                    {
                        "name": "col_long",
                        "type": "long"
                    },
                    {
                        "name": "col_integer",
                        "type": "integer"
                    },
                    {
                        "name": "col_keyword",
                        "type": "keyword"
                    },
                    {
                        "name": "col_text",
                        "type": "text",
                        "analyzer": "ik_max_word"
                    },
                    {
                        "name": "col_geo_point",
                        "type": "geo_point"
```

```
                    },
                    {
                        "name": "col_date",
                        "type": "date",
                        "format": "yyyy-MM-dd HH:mm:ss"
                    },
                    {
                        "name": "col_nested1",
                        "type": "nested"
                    },
                    {
                        "name": "col_nested2",
                        "type": "nested"
                    },
                    {
                        "name": "col_object1",
                        "type": "object"
                    },
                    {
                        "name": "col_object2",
                        "type": "object"
                    },
                    {
                        "name": "col_integer_array",
                        "type": "integer",
                        "array": true
                    },
                    {
                        "name": "col_geo_shape",
                        "type": "geo_shape",
                        "tree": "quadtree",
                        "precision": "10m"
                    }
                ]
            },
            "stepType": "elasticsearch"
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

说明:

目前VPC环境的Elasticsearch仅能使用自定义调度资源，运行在默认资源组会存在网络不通的情况。添加自定义资源组的具体操作请参见#unique_33。

2.3.2.19 配置LogHub Writer

本文将为您介绍LogHub Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置示例。

LogHub Writer使用SLS的Java SDK，可以将DataX Reader中的数据推送到指定的SLS LogHub上，供其他程序消费。

说明:

由于LogHub无法实现幂等，FailOver重跑任务时会引起数据重复。

LogHub Writer通过Datax框架获取Reader生成的数据，然后将Datax支持的类型通过逐一判断转换成STRING类型。当达到您指定的batchSize时，会使用SLS Java SDK一次性推送至LogHub。默认情况下，一次推送1,024条数据，batchSize值最大为4,096。

类型转换列表

LogHub Writer针对LogHub类型的转换，如下表所示。

DataX内部类型	LogHub数据类型
LONG	STRING
DOUBLE	STRING
STRING	STRING
DATE	STRING
BOOLEAN	STRING
BYTES	STRING

参数说明

参数	描述	是否必选	默认值
endpoint	SLS地址。	是	无
accessKeyId	访问SLS的AccessKeyId。	是	无
accessKeySecret	访问SLS的AccessKeySecret。	是	无
project	目标SLS的项目名称。	是	无
logstore	目标SLS LogStore的名称。	是	无
topic	选取topic。	否	空字符串
batchSize	每次批量数据的条数。	否	1,024
column	每条数据中的column名称。	是	无

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

脚本配置示例如下，具体参数的填写请参见上述的参数说明。

```
{
    "type": "job",
    "version": "2.0",//版本号
    "steps": [
        { //下面是关于Reader的模板，您可以查找相应的读插件文档。
            "stepType": "stream",
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "stepType": "loghub",//插件名
            "parameter": {
                "datasource": "",//数据源
                "column": [//字段
                    "col0",
                    "col1",
                    "col2",
                    "col3",
                    "col4",
                    "col5"
                ],
                "topic": "",//选取topic
                "batchSize": "1024",//一次性批量提交的记录数大小。
                "logstore": ""//目标SLS LogStore的名字
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "setting": {
        "errorLimit": {
            "record": ""//错误记录数
        },
        "speed": {
            "concurrent": 3,//作业并发数
            "throttle": false,//false代表不限流，下面的限流的速度不生效，
true代表限流。
            "dmu": 1//DMU值
        }
    },
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    }
```

```
}
```

2.3.2.20 配置OpenSearch Writer

本文为您介绍OpenSearch Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置举例。

OpenSearch Writer插件用于向OpenSearch中插入或者更新数据，主要提供给数据开发者，将处理好的数据导入到OpenSearch，以搜索的方式输出。数据传输的速率取决于OpenSearch表对应的帐号的qps。

实现原理

在底层实现上，OpenSearch Writer通过OpenSearch对外提供的开放搜索接口，关于接口的更多详情请参见开发搜索。

> 说明:
>
> - V2版本请参见请求结构。
> - V3版本使用二方包，依赖pom为：com.aliyun.opensearch aliyun-sdk-opensearch 2.1.3。
> - 如果您需要使用OpenSearchWriter插件，请务必使用JDK 1.6-32及以上版本，使用java -version查看Java版本号。
> - 目前默认资源组还不支持连接VPC环境，如果是VPC环境可能会存在网络问题。

插件特点

关于列顺序的问题

OpenSearch的列是无序的，因此OpenSearch Writer写入的时候，要严格按照指定的列的顺序写入，不能多也不能少。若指定的列比OpenSearch的列少，则其余列使用默认值或null。

例如需要导入的字段列表有b，c两个字段，但OpenSearch表中的字段有a，b，c三列，在列配置中可以写成”column”:[“c”,”b”]，表示会把Reader的第一列和第二列导入OpenSearch的c字段和b字段，而OpenSearch表中新插入纪的录的a字段会被置为默认值或null。

- 列配置错误的处理

  为保证写入数据的可靠性，避免多余列数据丢失造成数据质量故障。对于写入多余的列，OpenSearch Writer将报错。例如OpenSearch表字段为a，b，c，但是OpenSearch Writer写入的字段为多于3列的话，OpenSearch Writer将报错。
- 表配置注意事项

  OpenSearch Writer一次只能写入一个表。

- 任务重跑和failover

  重跑后会自动根据ID覆盖。所以插入OpenSearch的列中，必须要有一个ID，这个ID是OpenSearch的一行记录的唯一标识。唯一标识一样的数据，会被覆盖掉。

- 任务重跑和failover

  重跑后会自动根据ID覆盖。

OpenSearch Writer支持大部分OpenSearch类型，但也存在部分没有支持的情况，请注意检查您的类型，OpenSearch Writer针对OpenSearch类型的转换，如下表所示。

类型分类	OpenSearch数据类型
整数类	Int
浮点类	Double和Float
字符串类	TEXT、Literal和SHORT_TEXT
日期时间类	Int
布尔类	Literal

参数说明

参数	描述	是否必选	默认值
accessId	aliyun系统登录ID。	是	无
accessKey	aliyun系统登录Key。	是	无
host	OpenSearch连接的服务地址。该地址信息可以在应用详情页面进行查看。一般情况下，生产的服务地址为：http://opensearch-cn-internal.aliyuncs.com，测试的服务地址为：http://opensearch-cn-corp.aliyuncs.com 。	是	无
indexName	OpenSearch项目的名称。	是	无
table	写入数据的表名，不能填写多张表，因为DataX不支持同时导入多张表。	是	无
column	需要导入的字段列表，当导入全部字段时，可以配置为”column”:[“*”]，当需要插入部分OpenSearch列填写部分列，例如：”column”:[“id”，“name”]。OpenSearch支持列筛选、列换序，例如：表有a，b，c三个字段，您只同步c，b两个字段。可以配置成[“c”,”b”]，在导入过程中，字段a自动补空，设置为null。	是	无

参数	描述	是否必选	默认值
batchSize	单次写入的数据条数。OpenSearch写入为批量写入，一般来说OpenSearch的优势在查询，写入的tps不高，请根据自己的帐号申请的资源设置。一般情况OpenSearch的单条数据小于1MB，单次写入小于2MB。	如果是分区表，该选项必填，如果非分区表，该选项不可填写。	300
writeMode	OpenSearch Writer通过配置”writeMode”：“add/update”，保证写入的幂等性。 · “add”：当出现写入失败再次运行时，OpenSearch Writer 将清理该条数据，并导入新数据（原子操作）。 · “update”：表示该条插入数据是以修改的方式插入的（原子操作）。 说明: OpenSearch的批量插入并非原子操作，有可能会部分成功，部分失败。所以writeMode是个非常关键的选项，对于version=v3，暂不支持update操作。	是	无
ignoreWriteError	忽略写错误。 配置示例：“ignoreWriteError”: true。OpenSearch的写是批量写入的，当前批次的写失败是否忽略。若忽略，则继续执行其它的写操作。若不忽略，则直接结束当前任务，并返回错误。建议使用默认值。	否	false
version	opensearch的版本信息。配置示例“version”：“v3”，由于v2版本对于push操作限制比较多，建议首选v3版本。	否	v2

脚本开发介绍

配置写入OpenSearch的数据同步作业。

```
{
    "type": "job",
    "version": "1.0",
    "configuration": {
        "reader": {},
```

```
        "writer": {
            "plugin": "opensearch",
            "parameter": {
                "accessId": "*********",
                "accessKey": "********",
                "host": "http://yyyy.aliyuncs.com",
                "indexName": "datax_xxx",
                "table": "datax_yyy",
                "column": [
                "appkey",
                "id",
                "title",
                "gmt_create",
                "pic_default"
                ],
                "batchSize": 500,
                "writeMode": add,
                "version":"v2",
                "ignoreWriteError": false
            }
        }
    }
}
```

2.3.2.21 配置Table Store（OTS） Writer

本文为您介绍Table Store（OTS） Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置示例。

表格存储（Table Store）是构建在阿里云飞天分布式系统之上的NoSQL数据库服务，提供海量结构化数据的存储和实时访问。Table Store以实例和表的形式组织数据，通过数据分片和负载均衡技术，实现规模上的无缝扩展。

简而言之，Table Store Writer通过Table Store官方Java SDK连接到Table Store服务端，并通过SDK写入Table Store服务端 。Table Store Writer本身对于写入过程进行诸多优化，包括写入超时重试、异常写入重试、批量提交等功能。

目前Table Store Writer支持所有Table Store类型，其针对Table Store类型的转换，如下表所示。

类型分类	Table Store数据类型
整数类	INTEGER
浮点类	DOUBLE
字符串类	STRING
布尔类	BOOLEAN
二进制类	BINARY

说明:

您需要将INTEGER类型的数据，在脚本模式中配置为INT类型，DataWorks会将其转换为INTEGER类型。如果您直接配置为INTEGER类型，日志将会报错，导致任务无法顺利完成。

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
endPoint	Table Store Server的EndPoint（服务地址）。	是	无
accessId	Table Store的AccessID。	是	无
accessKey	Table Store的AccessKey。	是	无
instanceName	Table Store的实例名称。 实例是您使用和管理Table Store服务的实体。开通Table Store服务后，需要通过管理控制台来创建实例，然后在实例内进行表的创建和管理。实例是Table Store资源管理的基础单元，Table Store对应用程序的访问控制和资源计量都在实例级别完成。	是	无
table	所选取的需要抽取的表名称，此处能且只能填写一张表。在Table Store中不存在多表同步的需求。	是	无

参数	描述	是否必选	默认值
primaryKey	Table Store的主键信息，使用JSON的数组描述字段信息。Table Store本身是NoSQL系统，在Table Store Writer导入数据过程中，必须指定相应的字段名称。 说明: Table Store的PrimaryKey仅支持STRING和INT两种类型，因此Table Store Writer本身也限定填写上述两种类型。 数据同步系统本身支持类型转换的，因此对于源头数据非STRING/INT，Table Store Writer会进行数据类型转换。配置示例如下: `"primaryKey" : [ {"name":"pk1", "type":"string"}, {"name":"pk2", "type":"int"} ],`	是	无
column	所配置的表中需要同步的列名集合，使用JSON的数组描述字段信息。 使用格式为: `{"name":"col2", "type":"INT"},` 其中的name指定写入的Table Store列名，type指定写入的类型。Table Store类型支持STRING、INT、DOUBLE、BOOL和BINARY类型。	是	无
writeMode	writeMode表示数据写入表格存储的格式，目前支持以下两种模式: · PutRow：对应于Table Store PutRow API，插入数据到指定的行。如果该行不存在，则新增一行。如果该行存在，则覆盖原有行。 · UpdateRow：对应于Table Store UpdateRow API，更新指定行的数据。如果该行不存在，则新增一行。如果该行存在，则根据请求的内容在这一行中新增、修改或者删除指定列的值。	是	无

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个写入Table Store作业。

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[
        { //下面是关于Reader的模板，您可以查看相应的读插件文档。
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"ots",//插件名
            "parameter":{
                "datasource":"",//数据源
                "column":[//字段
                    {
                        "name":"columnName1",//字段名
                        "type":"INT"//数据类型
                    },
                    {
                        "name":"columnName2",
                        "type":"STRING"
                    },
                    {
                        "name":"columnName3",
                        "type":"DOUBLE"
                    },
                    {
                        "name":"columnName4",
                        "type":"BOOLEAN"
                    },
                    {
                        "name":"columnName5",
                        "type":"BINARY"
                    }
                ],
                "writeMode":"",//写入模式
                "table":"",//表名
                "primaryKey":[//Table Store的主键信息
                    {
                        "name":"pk1",
                        "type":"STRING"
                    },
                    {
                        "name":"pk2",
                        "type":"INT"
                    }
                ]
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数
        },
        "speed":{
```

```
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流。
            "concurrent":1,//作业并发数
            "dmu":1//DMU值
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
}
```

2.3.2.22 配置RDBMS Writer

本文为您介绍RDBMS Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置示例。

RDBMS Writer插件实现了写入数据至RDBMS主库的目的表的功能。在底层实现上，RDBMS Writer通过DataX框架获取Reader生成的协议数据，通过JDBC连接远程RDBMS数据库，并执行相应的`insert into...`的SQL语句，将数据写入RDBMS。RDBMS Writer是一个通用的关系数据库写插件，您可以通过注册数据库驱动等方式，增加任意多样的关系数据库写支持。

RDBMS Writer面向ETL开发工程师，通过RDBMS Writer从数仓导入数据至RDBMS。同时RDBMS Writer也可以作为数据迁移工具，为数据库管理员等用户提供服务。

类型转换

目前RDBMS Writer支持数字、字符等大部分通用的关系数据库类型，但也存在部分类型没有支持的情况，请注意检查您的数据类型。

参数说明

参数	描述	是否必选	默认值
jdbcUrl	描述的是到对端数据库的JDBC连接信息，JDBCUrl按照RDBMS官方规范，并可填写连接附件控制信息。请注意不同的数据库JDBC的格式是不同的，DataX会根据具体jdbc的格式选择合适的数据库驱动完成数据读取。 · DM格式：`jdbc:dm://ip:port/database` · DB2格式：`jdbc:db2://ip:port/database` · PPAS格式：`jdbc:edb://ip:port/database`	是	无

参数	描述	是否必选	默认值
username	数据源的用户名。	是	无
password	数据源指定用户名的密码。	是	无
table	目标表名称，如果表的schema信息和上述配置username不一致，请使用schema.table的格式填写table信息。	是	无
column	所配置的表中需要同步的列名集合。以英文逗号（,）进行分隔。  说明: 建议您不要使用默认列情况。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句，目前只允许执行一条SQL语句，例如清除旧数据。  说明: 当有多条SQL语句时，不支持事务。	否	无
postSql	执行数据同步任务之后执行的SQL语句，目前只允许执行一条SQL语句，例如加上某一个时间戳。  说明: 当有多条SQL语句时，不支持事务。	否	无
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据集成与PostgreSQL的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据集成运行进程OOM情况。	否	1024

功能说明

配置一个写入RDBMS的作业。

```
{
    "job": {
        "setting": {
            "speed": {
                "channel": 1
            }
        },
        "content": [
            {
                "reader": {
                    "name": "streamreader",
                    "parameter": {
```

```
                        "column": [
                            {
                                "value": "DataX",
                                "type": "string"
                            },
                            {
                                "value": 19880808,
                                "type": "long"
                            },
                            {
                                "value": "1988-08-08 08:08:08",
                                "type": "date"
                            },
                            {
                                "value": true,
                                "type": "bool"
                            },
                            {
                                "value": "test",
                                "type": "bytes"
                            }
                        ],
                        "sliceRecordCount": 1000
                    }
                },
                "writer": {
                    "name": "RDBMS Writer",
                    "parameter": {
                        "connection": [
                            {
                                "jdbcUrl": "jdbc:dm://ip:port/database
",
                                "table": [
                                    "table"
                                ]
                            }
                        ],
                        "username": "username",
                        "password": "password",
                        "table": "table",
                        "column": [
                            "*"
                        ],
                        "preSql": [
                            "delete from XXX;"
                        ]
                    }
                }
            }
        ]
    }
}
```

RDBMS Writer增加新的数据库支持的操作如下。

1. **进入RDBMS Writer对应目录，这里${DATAX_HOME}为DataX主目录，即***${DATAX_HOME}/plugin/writer/RDBMS Writer*。

2. 在RDBMS Writer插件目录下有*plugin.json*配置文件，在此文件中注册您具体的数据库驱动，具体放在drivers数组中。RDBMS Writer插件在任务执行时，会动态选择合适的数据库驱动连接数据库。

```
{
    "name": "RDBMS Writer",
    "class": "com.alibaba.datax.plugin.reader.RDBMS Writer.RDBMS
Writer",
    "description": "useScene: prod. mechanism: Jdbc connection using
 the database, execute select sql, retrieve data from the ResultSet
. warn: The more you know about the database, the less problems you
encounter.",
    "developer": "alibaba",
    "drivers": [
        "dm.jdbc.driver.DmDriver",
        "com.ibm.db2.jcc.DB2Driver",
        "com.sybase.jdbc3.jdbc.SybDriver",
        "com.edb.Driver"
    ]
}
```

3. 在RDBMS Writer插件目录下有libs子目录，您需要将您具体的数据库驱动放到libs目录下。

```
$tree
.
|-- libs
|   |-- Dm7JdbcDriver16.jar
|   |-- commons-collections-3.0.jar
|   |-- commons-io-2.4.jar
|   |-- commons-lang3-3.3.2.jar
|   |-- commons-math3-3.1.1.jar
|   |-- datax-common-0.0.1-SNAPSHOT.jar
|   |-- datax-service-face-1.0.23-20160120.024328-1.jar
|   |-- db2jcc4.jar
|   |-- druid-1.0.15.jar
|   |-- edb-jdbc16.jar
|   |-- fastjson-1.1.46.sec01.jar
|   |-- guava-r05.jar
|   |-- hamcrest-core-1.3.jar
|   |-- jconn3-1.0.0-SNAPSHOT.jar
|   |-- logback-classic-1.0.13.jar
|   |-- logback-core-1.0.13.jar
|   |-- plugin-rdbms-util-0.0.1-SNAPSHOT.jar
|   `-- slf4j-api-1.7.10.jar
|-- plugin.json
|-- plugin_job_template.json
`-- RDBMS Writer-0.0.1-SNAPSHOT.jar
```

2.3.2.23 配置Stream Writer

本文为您介绍Stream Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置举例。

Stream Writer 插件实现了从 Reader 端读取数据并向屏幕上打印数据或者直接丢弃数据，主要用于数据同步的性能测试和基本的功能测试。

参数说明

- print
  - 描述：是否向屏幕打印输出。
  - 必选：否。
  - 默认值：true。

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个从 Reader 端读取数据并向屏幕打印的作业：

```
{
    "type":"job",
    "version":"2.0",//版本号
    "steps":[
        { //下面是关于Reader的模板，可以找相应的读插件文档
            "stepType":"stream",
            "parameter":{},
            "name":"Reader",
            "category":"reader"
        },
        {
            "stepType":"stream",//插件名
            "parameter":{
                "print":false,//是否向屏幕打印输出
                "fieldDelimiter":","//列分隔符
            },
            "name":"Writer",
            "category":"writer"
        }
    ],
    "setting":{
        "errorLimit":{
            "record":"0"//错误记录数
        },
        "speed":{
            "throttle":false,//false代表不限流，下面的限流的速度不生效，true
代表限流
            "concurrent":1,//作业并发数
            "dmu":1//DMU值
        }
    },
    "order":{
        "hops":[
            {
                "from":"Reader",
                "to":"Writer"
            }
        ]
    }
```

```
}
```

2.3.2.24 配置HybridDB for MySQL Writer

本文将为您介绍HybridDB for MySQL Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置举例。

HybridDB for MySQL Writer插件实现了写入数据到MySQL数据库目标表的功能。在底层实现上，HybridDB for MySQL Writer通过JDBC连接远程HybridDB for MySQL数据库，并执行相应的`insert into`或`replace into`的SQL语句将数据写入HybridDB for MySQL，分批次提交入库，需数据库本身采用InnoDB引擎。

 说明:

在开始配置HybridDB for MySQL Writer插件前，请首先配置好数据源，详情请参见配置HybridDB for MySQL数据源。

HybridDB for MySQL Writer面向数据开发工程师，使用HybridDB for MySQL Writer从数仓导入数据到HybridDB for MySQL。同时，HybridDB for MySQL Writer也可以作为数据迁移工具为DBA等用户提供服务。HybridDB for MySQL Writer通过数据同步框架根据您配置的writeMode获取Reader生成的协议数据。

 说明:

整个任务至少需要具备`insert/replace into`的权限。是否需要其他权限，取决于您配置任务时在preSql和postSql中指定的语句。

类型转换列表

目前HybridDB for MySQL Writer存在小部分HybridDB for MySQL类型未支持的情况，请注意检查您的类型。

HybridDB for MySQL Writer针对HybridDB for MySQL类型的转换列表如下所示。

类型分类	HybridDB for MySQL数据类型
整数类	Int、Tinyint、Smallint、Mediumint、Bigint和Year
浮点类	Float、Double和Decimal
字符串类	Varchar、Char、Tinytext、Text、Mediumtext和LongText
日期时间类	Date、Datetime、Timestamp和Time
布尔类	Bool
二进制类	Tinyblob、Mediumblob、Blob、LongBlob和Varbinary

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无
writeMode	选择导入模式，可以支持insert/replace方式。 · replace into…：没有遇到主键/唯一性索引冲突时，与insert into行为一致，冲突时会用新行替换原有行所有字段。 · insert into…：当主键/唯一性索引冲突时会写不进去冲突的行，以脏数据的形式体现。	否	insert
column	目标表需要写入数据的字段，字段之间用英文所逗号分隔。例如`"column":["id","name","age"]`。如果要依次写入全部列，使用*表示，例如`"column":["*"]`。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如清除旧数据。	否	无
postSql	执行数据同步任务之后执行的SQL语句，目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如加上某一个时间戳。	否	无
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据同步系统与MySQL的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据同步运行进程OOM情况。	否	1024

向导开发介绍

1. 选择数据源

配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，一般选择您配置的数据源名称。
表	即上述参数说明中的table。
导入前准备语句	即上述参数说明中的preSql，输入执行数据同步任务之前率先执行的SQL语句。
导入后完成语句	即上述参数说明中的postSql，输入执行数据同步任务之后执行的SQL语句。
主键冲突	即上述参数说明中的writeMode，可选择需要的导入模式。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应的关系，单击添加一行可增加单个字段，单击删除即可删除当前字段 。

配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。

3. 通道控制

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。

配置	说明
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

96811

脚本开发介绍

脚本配置样例如下，详情请参见上述参数说明。

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {},
        {
            "parameter": {
                "postSql": [],//导入后完整语句
                "datasource": "px_aliyun_hymysql",//数据源名
                "column": [//目标端列名
                    "id",
                    "name",
                    "sex",
                    "salary",
                    "age",
                    "pt"
                ],
                "writeMode": "insert",//写入模式
                "batchSize": 256,//一次性批量提交的记录数大小
                "encoding": "UTF-8",//编码格式
                "table": "person_copy",//目标表名
                "preSql": []//导入前准备语句
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",//版本号
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {//错误记录数
            "record": ""
        },
        "speed": {
            "concurrent": 7,//并发数
            "throttle": true,//同步速度限流
            "mbps": 1,//限流值
        }
    }
```

```
}
```

2.3.2.25 配置AnalyticDB for PostgreSQL Writer

本文将为您介绍AnalyticDB for PostgreSQL Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置示例。

AnalyticDB for PostgreSQL Writer插件实现了向AnalyticDB for PostgreSQL写入数据。在底层实现上，AnalyticDB for PostgreSQL Writer通过JDBC连接远程AnalyticDB for PostgreSQL数据库，并执行相应的SQL语句，从AnalyticDB for PostgreSQL库中选取数据。RDS在公共云提供AnalyticDB for PostgreSQL存储引擎。

说明:

开始配置AnalyticDB for PostgreSQL Writer插件前，请首先配置好数据源，详情请参见配置AnalyticDB for PostgreSQL数据源。

简而言之，AnalyticDB for PostgreSQL Writer通过JDBC连接器连接至远程的AnalyticDB for PostgreSQL数据库，根据您配置的信息生成查询SELECT SQL语句，发送至远程AnalyticDB for PostgreSQL数据库。然后使用CDP自定义的数据类型，将该SQL执行返回结果拼装为抽象的数据集，并传递给下游Writer处理。

- 对于您配置的table、column和where等信息，AnalyticDB for PostgreSQL Writer将其拼接为SQL语句，发送至AnalyticDB for PostgreSQL数据库。
- 对于您配置的querySql信息，AnalyticDB for PostgreSQL直接将其发送至AnalyticDB for PostgreSQL数据库。

类型转换列表

AnalyticDB for PostgreSQL Writer支持大部分AnalyticDB for PostgreSQL类型，但也存在部分类型没有支持的情况，请注意检查您的类型。

PAnalyticDB for PostgreSQL Writer针对AnalyticDB for PostgreSQL的类型转换列表，如下所示。

类型分类	AnalyticDB for PostgreSQL数据类型
LONG	BIGINT、BIGSERIAL、INTEGER、SMALLINT和SERIAL
DOUBLE	DOUBLE、PRECISION、MONEY、NUMERIC和REAL
STRING	VARCHAR、CHAR、TEXT、BIT和INET
DATE	DATE、TIME和TIMESTAMP
BOOLEAN	BOOL
BYTES	BYTEA

 说明:

- 除上述罗列字段类型外，其他类型均不支持。
- MONEY、INET和BIT需要您使用`a_inet::varchar`类似的语法进行转换。

参数说明

参数	描述	是否必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无
writeMode	选择导入模式，可以支持insert和copy方式。 - insert：执行PostgreSQL的`insert into...values...`语句，将数据写出到PostgreSQL中。当数据出现主键/唯一性索引冲突时，待同步的数据行写入PostgreSQL失败，当前记录行成为脏数据。建议您优先选择insert模式。 - copy：PostgreSQL提供copy命令，用于表与文件（标准输出，标准输入）之间的相互复制。数据集成支持使用`copy from`，将数据加载到表中。建议您在遇到性能问题时再尝试使用该模式。	否	insert
column	目标表需要写入数据的字段，字段之间用英文逗号分隔。例如`"column":["id","name","age"]`。如果要依次写入全部列，使用*表示，例如"column":["*"]。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如清除旧数据。	否	无
postSql	执行数据同步任务之后执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如加上某一个时间戳。	否	无

参数	描述	是否必选	默认值
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据集成与AnalyticDB for PostgreSQL的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据集成运行进程OOM情况。	否	1024

向导开发介绍

1. 选择数据源

   配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，通常选择您配置的数据源名称。
表	即上述参数说明中的table，选择需要同步的表。
导入前准备语句	即上述参数说明中的preSql，输入执行数据同步任务之前率先执行的SQL语句。
导入后完成语句	即上述参数说明中的postSql，输入执行数据同步任务之后执行的SQL语句。
导入模式	即上述参数说明中的writeMode，包括insert和copy两种模式。

2. 字段映射，即上述参数说明中的column。

左侧的源头表字段和右侧的目标表字段为一一对应的关系。单击添加一行可以增加单个字段，将鼠标放至需要删除的字段，即可选择删除按钮进行删除。



配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。

3. 通道控制

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。

配置	说明
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {
                "postSql": [],//导入后的完成语句。
                "datasource": "test_004",//数据源名。
                "column": [//目标表的列名。
                    "id",
                    "name",
                    "sex",
                    "salary",
                    "age"
                ],
                "table": "public.person",//目标表的表名。
                "preSql": []//导入前的准备语句。
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",//版本号。
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {//错误记录数。
            "record": ""
        },
        "speed": {
            "concurrent": 6,//并发数。
            "throttle": false,//同步速率是否限流。
        }
    }
```

```
}
```

2.3.2.26 配置POLARDB Writer

本文将为您介绍POLARDB Writer支持的数据类型、写入方式、字段映射和数据源等参数及配置举例。

POLARDB Writer插件实现了写入数据到POLARDB数据库目标表的功能。在底层实现上，POLARDB Writer通过JDBC连接远程POLARDB 数据库，并执行相应的`insert into...`或`replace into...`的SQL语句将数据写入POLARDB，内部会分批次提交入库，需要数据库本身采用innodb引擎。

 说明:

在开始配置POLARDB Writer插件前，请首先配置好数据源，详情请参见配置POLARDB数据源。

POLARDB Writer面向ETL开发工程师，他们使用POLARDB Writer从数仓导入数据到POLARDB 。同时POLARDB Writer也可以作为数据迁移工具为DBA等用户提供服务。

POLARDB Writer通过数据同步框架获取Reader生成的协议数据，根据您配置的writeMode生成。

 说明:

整个任务至少需要具备`insert/replace into...`的权限，是否需要其他权限，取决于您配置任务时在preSql和postSql中指定的语句。

类型转换列表

类似于POLARDB Reader ，目前POLARDB Writer支持大部分POLARDB类型，但也存在部分类型没有支持的情况，请注意检查您的类型。

POLARDB Writer针对POLARDB类型的转换列表，如下所示。

类型分类	POLARDB数据类型
整数类	Int、Tinyint、Smallint、Mediumint、Bigint和Year
浮点类	Float、Double和Decimal
字符串类	Varchar、Char、Tinytext、T ext、Mediumtext和LongText
日期时间类	Date、Datetime、Timestamp和Time
布尔型	Bool
二进制类	Tinyblob、Mediumblob、Blob、LongBlob和Varbinary

参数说明

参数	描述	必选	默认值
datasource	数据源名称，脚本模式支持添加数据源，此配置项填写的内容必须要与添加的数据源名称保持一致。	是	无
table	选取的需要同步的表名称。	是	无
writeMode	选择导入模式，可以支持insert/replace方式。 · replace into…：没有遇到主键/唯一性索引冲突时，与insert into行为一致，冲突时会用新行替换原有行所有字段。 · insert into…：当主键/唯一性索引冲突时会写不进去冲突的行，以脏数据的形式体现。 · INSERT INTO table (a,b,c) VALUES (1,2,3) ON DUPLICATE KEY UPDATE…：没有遇到主键/唯一性索引冲突时，与insert into行为一致，冲突时会用新行替换已经指定的字段的语句写入数据到POLARDB。	否	insert
column	目标表需要写入数据的字段，字段之间用英文所逗号分隔。例如“column”: [“id”, “name”, “age”]。如果要依次写入全部列，使用表示。 例如“column”: [“”]。	是	无
preSql	执行数据同步任务之前率先执行的SQL语句。目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如清除旧数据。	否	无
postSql	执行数据同步任务之后执行的SQL语句，目前向导模式仅允许执行一条SQL语句，脚本模式可以支持多条SQL语句，例如加上某一个时间戳。	否	无
batchSize	一次性批量提交的记录数大小，该值可以极大减少数据同步系统与POLARDB的网络交互次数，并提升整体吞吐量。但是该值设置过大可能会造成数据同步运行进程OOM情况。	否	1024

向导开发介绍

1. 选择数据源

配置同步任务的数据来源和数据去向。

配置	说明
数据源	即上述参数说明中的datasource，一般填写您配置的数据源名称。
表	即上述参数说明中的table，选择需要同步的表。
导入前准备语句	即上述参数说明中的preSql，输入执行数据同步任务之前率先执行的SQL语句。
导入后完成语句	即上述参数说明中的postSql，输入执行数据同步任务之后执行的SQL语句。
主键冲突	即上述参数说明中的writeMode，可选择需要的导入模式。

2. 字段映射，即上述参数说明中的column。

   左侧的源头表字段和右侧的目标表字段为一一对应的关系，单击添加一行可增加单个字段，鼠标放至需要删除的字段上，即可单击删除图标进行删除。

配置	说明
同名映射	单击同名映射，可以根据名称建立相应的映射关系，请注意匹配数据类型。
同行映射	单击同行映射，可以在同行建立相应的映射关系，请注意匹配数据类型。
取消映射	单击取消映射，可以取消建立的映射关系。
自动排版	可以根据相应的规律自动排版。

3. 通道控制

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。

配置	说明
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

脚本开发介绍

脚本配置样例如下，详情请参见上述参数说明。

```
{
    "type": "job",
    "steps": [
        {
            "parameter": {},
            "name": "Reader",
            "category": "reader"
        },
        {
            "parameter": {
                "postSql": [],//导入后完成语句
                "datasource": "test_005",//数据源名
                "column": [//目标列名
                    "id",
                    "name",
                    "age",
                    "sex",
                    "salary",
                    "interest"
                ],
                "writeMode": "insert",//写入模式
                "batchSize": 256,//一次性批量提交的记录数大小
                "encoding": "UTF-8",//编码格式
                "table": "POLARDB_person_copy",//目标表名
                "preSql": []//导入前准备语句
            },
            "name": "Writer",
            "category": "writer"
        }
    ],
    "version": "2.0",//版本号
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {//错误记录数
            "record": ""
        },
        "speed": {
            "concurrent": 6,//并发数
            "throttle": false,//同步速率限流
        }
    }
```

```
}
```

2.3.2.27 配置TSDB Writer

TSDB Writer插件实现了将数据点写入阿里巴巴自研TSDB数据库。

时间序列数据库（Time Series Database , 简称TSDB）是一种高性能、低成本、稳定可靠的在线时序数据库服务。提供高效读写、高压缩比存储、时序数据插值及聚合计算，广泛应用于物联网（IoT）设备监控系统 、企业能源管理系统（EMS）、生产安全监控系统和电力检测系统等行业场景。

TSDB提供百万级时序数据秒级写入，高压缩比低成本存储、预降采样、插值和多维聚合计算，查询结果可视化功能。TSDB可解决由于设备采集点数量大、数据采集频率高，造成的存储成本高、写入和查询分析效率低等问题。

目前仅支持脚本模式配置方式，更多详情请参见时序时空数据库文档。

实现原理

TSDB Writer通过HTTP连接TSDB实例，并通过`/api/put`接口将数据点写入。

约束限制

目前仅支持兼容TSDB 2.4.x及以上版本。

支持的数据类型

类型分类	数据集成column配置类型	TSDB数据类型
字符串	string	TSDB数据点序列化字符串，包括timestamp、metric、tags和value。

参数说明

数据源	参数	描述	是否必选	默认值
公共参数	sourceDbType	数据源的类型。	否	TSDB 说明: 目前支持TSDB和RDB两个取值。其中，TSDB包括OpenTSDB、InfluxDB、Prome。RDB包括MySQL、Oracle、PostgreSQL、

数据源	参数	描述	是否必选	默认值
数据源为TSDB	endpoint	TSDB的HTTP连接地址。	是，格式为`http://IP:Port`。	无
	batchSize	每次批量写入数据的条数。	否，数据类型为int，需要确保大于0。	100
	maxRetryTime	失败后重试的次数。	否，数据类型为int，需要确保大于1。	3
	ignoreWriteError	如果设置为true，则忽略写入错误，继续写入。如果多次重试后仍写入失败，则终止写入任务。	否，数据类型为bool。	false
数据源为RDB	endpoint	TSDB的HTTP连接地址。	是，格式为`http://IP:Port`。	无
	column	关系型数据库中表的字段名。	是	无 说明: 此处的字段顺序，需要和 Reader插件中配置的column字段的顺序保持一致。
	columnType	关系型数据库中表字段，映射到TSDB中的类型。支持的类型如下所示： · timestamp：该字段为时间戳。 · tag：该字段为tag。 · metric_num：该Metric的value是数值类型。 · metric_string：该Metric的value是字符串类型。	是	无 说明: 此处的字段顺序，需要和 Reader插件中配置的column字段的顺序保持一致。

数据源	参数	描述	是否必选	默认值
	batchSize	每次批量写入数据的条数。	否，数据类型为int，需要确保大于0。	100

向导开发介绍

暂不支持向导模式开发。

脚本开发介绍

配置一个同步数据至TSDB的作业。

```
```json
{
    "order": {
        "hops": [
            {
                "from": "Reader",
                "to": "Writer"
            }
        ]
    },
    "setting": {
        "errorLimit": {
            "record": "0"
        },
        "speed": {
            "concurrent": 1,
            "throttle": true
        }
    },
    "steps": [
        {
            "category": "reader",
            "name": "Reader",
            "parameter": {},
            "stepType": ""
        },
        {
            "category": "writer",
            "name": "Writer",
            "parameter": {
                "endpoint": "http://localhost:8242",
                "sourceDbType": "RDB",
                "batchSize": 256,
                "column": [
                    "name",
                    "type",
                    "create_time",
                    "price"
                ],
                "columnType": [
                    "tag",
                    "tag",
                    "timestamp",
                    "metric_num"
                ]
            },
```
```

```
            "stepType": "tsdb"
        }
    ],
    "type": "job",
    "version": "2.0"
}
```

性能报告

- 性能数据特征

  从Metric、时间线、Value和采集周期四个方面进行描述。

  - Metric：指定一个Metric为m。
  - tagkv：前四个tagkv全排列，形成10*20*100*100=2,000,000条时间线，最后IP对应2,000,000条时间线，从1开始自增。

tag_k	tag_v
zone	z1~z10
cluster	c1~c20
group	g1~100
app	a1~a100
ip	ip1~ip2,000,000

  - value：度量值为[1, 100]区间内的随机值。
  - interval：采集周期为10秒，持续摄入3小时，总数据量为3*60*60/10*2,000,000=2,160,000,000个数据点。

- 性能测试结果

通道数	数据集成速度（Rec/s）	数据集成流量（MB/s）
1	129,753	15.45
2	284,953	33.70
3	385,868	45.71

2.3.3 优化配置

本文将为您介绍数据同步速度的影响因素，如何通过调整同步作业的并发配置来达到最大化同步速度，作业限速和不限速的区别，以及自定义资源组的注意事项。

DataWorks数据集成支持任意位置任意网络环境下的任意数据源之间的实时、离线数据互通，是一站式数据同步的全栈平台，并允许您在各种云和本地数据存储中每天复制数十TB的数据。

速度超快的数据传输性能以及400+对异构数据源之间的数据互通是确保您专注于核心大数据问题的关键，您可构建高级分析解决方案并从所有数据获得深入洞察。

数据同步速度的影响因素

影响数据同步速度的因素如下所示。

- 来源端数据源
  - 数据库的性能：CPU、内存、SSD硬盘、网络和硬盘等。
  - 并发数：如果数据源并发数高，数据库负载便高。
  - 网络：网络的带宽（吞吐量）、网速。一般来说，数据库的性能越好，它可以承载的并发数越高，可为数据同步作业设置更高的并发进行数据的抽取。
- 数据集成的同步任务配置
  - 传输速度：是否设置任务同步速度上限值。
  - 并发：从源并行读取或并行写入数据存储端的最大线程数。
  - WAIT资源。
  - Bytes的设置：单个线程的Bytes=1048576，在网速比较敏感时，会出现超时现象，建议设置小一些。
  - 查询语句是否建索引。
- 目的端数据源
  - 性能：CPU、内存、SSD 硬盘、网络和硬盘。
  - 负载：目的数据库负载过高会影响同步任务数据写入效率。
  - 网络：网络的带宽（吞吐量），网速。

数据源端和目的端数据库的性能、负载和网络情况主要由您自己关注和调优，下文将为您重点介绍在数据集成产品中配置同步任务的核心配置。

并发

向导模式通过界面化配置并发数，指定任务所使用的并行度。通过脚本模式配置并发数的示例如下。

```
"setting": {
      "speed": {
        "concurrent": 10
      }
```

```
    }   }
```

限速

商业化之后，数据集成同步任务默认不限速，任务将在所配置的并发数的限制上以最高能达到的速度进行同步。另一方面，考虑到速度过高可能对数据库造成过大的压力从而影响生产，数据集成同时提供了限速选项，您可以按照实际情况调优配置（建议选择限速之后，最高速度上限不应超过30MB/s）。脚本模式通过如下示例代码配置限速，代表1MB/s的传输带宽。

```
"setting": {
      "speed": {
         "throttle": true // 限流
        "mbps": 1,  // 具体速率值
      }
    }
```

说明:

- 当throttle设置为false时，表示不限速，则mbps的配置无意义。
- 流量度量值是数据集成本身的度量值，不代表实际网卡流量。通常情况下，网卡流量往往是通道流量膨胀到1至2倍，实际流量膨胀看具体的数据存储系统传输序列化情况。
- 半结构化的单个文件没有切分键的概念，多个文件可以设置作业速率上限来提高同步的速度，但作业速率上限和文件的个数有关。例如有n个文件，作业速率上限最多设置为nMB/s，如果设置n+1MB/s还是以nMB/s速度同步，如果设置为n-1MB/s，则以n-1MB/s速度同步。
- 关系型数据库设置作业速率上限和切分键才能根据作业速率上限将表进行切分，关系型数据库只支持数值型作为切分键，但Oracle数据库支持数值型和字符串类型作为切分键。

数据同步过慢的场景

- 场景一：同步任务使用公共调度（WAIT）资源时，一直在等待状态。
  - 场景示例

    在DataWorks中对任务进行测试时，出现任务一直等待的状态，或好多测试任务都处于等待状态，而且还提示了系统内部错误。

    例如一个数据同步任务执行完成，共等待了约800s，但是日志显示任务只运行了 18s，使用的是默认资源组，现在运行其他同步任务，也是RDS到MaxCompute，一共几百条数据，一直处于等待中 。

    显示的等待日志如下所示：

    ```
    2017-01-03 07:16:54 : State: 2(WAIT) | Total: 0R 0B | Speed: 0R/s
    0B/s | Error: 0R 0B | Stage: 0.0%
    ```

  - 解决方法

    因为您使用的是公共调度资源，公共资源能力是受限的有很多项目都在使用，不只是单个用户的2-3个任务，任务实际运行10秒，但是延长到800秒，是因为您的任务下发执行时，发现资源不足，需等待获取资源。

    如果对于同步速度和等待时间比较敏感，建议在低峰期配置同步任务，一般晚上零点到3点同步任务比较多，这样可以避开零点到3点的时间段，便可相对减少等待资源的情况。
- 场景二：

  提高多个任务导入数据到同一张表的同步速度。

  - 场景示例

    想要将多个数据源的表同步到一张表里，所以将同步任务设置成串行任务，但是最后发现同步时间很长。
  - 解决方法

    可以同时启动多个任务，同时往一个数据库进行写入，需注意以下问题：

    ■ 确保目标数据库负载能力是能够承受，避免不能正常工作。

    ■ 在配置工作流任务时，可以选择单个任务节点，配置分库分表任务或者在一个工作流中设置多个节点同时执行。

    ■ 如果任务执行时，出现等待资源（WAIT）情况，可以低峰期配置同步任务，这样任务有较高的执行优先级。

- 场景三：

  数据同步任务where条件没有索引，导致全表扫描同步变慢。

  - 场景示例

    执行的SQL如下所示：

    ```
    select bid,inviter,uid,createTime from `relatives` where
    createTime>='2016-10-2300:00:00'and reateTime<'2016-10-24 00:00:00
    ';
    ```

    从2016-10-25 11:01:24.875开始执行，到2016-10-25 11:11:05.489开始返回结果。同步程序在等待数据库返回SQL查询结果，MaxCompute需等待很久才能执行。

  - 分析原因

    where条件查询时，createTime列没有索引，导致查询全表扫描。

  - 解决方法

    建议where条件使用有索引相关的列，提高性能，索引也可以补充添加。

2.4 常见配置

2.4.1 添加安全组

本文将为您介绍选择不同区域的DataWorks时，如何添加需要的安全组。通常情况下，如果您使用的是ECS自建数据库，则必须添加安全组才能保证数据源连通性正常。

为保证数据库的安全稳定，在开始使用某些数据库的实例前，您需要将访问数据库的IP地址或IP段加到目标实例的#unique_111或安全组中。

添加安全组

- 如果您的ECS上的自建数据源同步任务运行在自定资源组上，要给自定资源组机器授权，将自定义机器内/外网IP和端口添加到ECS安全组上。
- 如果您的ECS上的自建数据源运行默认的资源组上，要给默认的机器授权，根据您的ECS的机器Region来选择添加您的安全组内容，例如您的ECS是华北2，安全组便添加华北2（北京）：sg-2ze3236e8pcbxw61o9y0和1156529087455811内容，并且只能在华北2添加数据源，如下表所示。

Region	授权对象	账号ID
华东1（杭州）	sg-bp13y8iuj33uqpqvgqw2	1156529087455811
华东2（上海）	sg-uf6ir5g3rlu7thymywza	1156529087455811
华南1（深圳）	sg-wz9ar9o9jgok5tajj7ll	1156529087455811

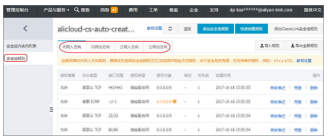
Region	授权对象	账号ID
亚太东南1（新加坡）	sg-t4n222njci99ik5y6dag	1156529087455811
中国（香港）	sg-j6c28uqpqb27yc3tjmb6	1156529087455811
美国西部 1（硅谷）	sg-rj9bowpmdvhyl53lza2j	1156529087455811
美国东部1	sg-0xienf2ak8gs0puz68i9	1156529087455811
华北2（北京）	sg-2ze3236e8pcbxw61o9y0	1156529087455811

说明:

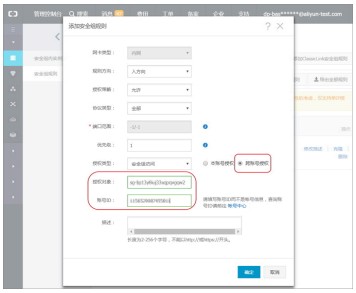
VPC环境的ECS不支持添加上面的安全组，因为上面的都是经典网络类型的IP，会存在网络类型不同的问题。

ECS添加安全组

1. 登录云服务器ECS的管理控制台。
2. 选择左侧导航栏中的网络和安全 > 安全组。
3. 选择目标地域。
4. 找到要配置授权规则的安全组，单击操作列下的配置规则。
5. 进入安全组规则页面，单击添加安全组规则。



6. 填写添加安全组规则对话框中的配置。



7. 单击确定。

2.4.2 添加白名单

本文将为您介绍选择不同区域的DataWorks时，如何添加需要的不同白名单的内容。通常情况下，如果您使用的是RDS数据源，则必须添加白名单才能保证数据连通性正常。

为保证数据库的安全稳定，在开始使用某些数据库时实例前，您需要将访问数据库的IP地址或者IP段加到目标实例的白名单或#unique_111中。

说明:

添加白名单功能仅对数据集成生效，其他类型任务不支持白名单功能。

添加白名单

1. 以开发者身份进入DataWorks控制台，导航至工作空间列表页面。
2. 选择工作空间区域。

   目前DataWorks支持多个区域，此处要选择的工作空间区域和您购买不同区域的MaxCompute有关，例如华东2（上海）、华南1（深圳）、中国（香港）等，都代表您开通这些区域的MaxCompute，您可以手动切换不同的区域入口。

3. 根据工作空间所在的区域选择相应的白名单。

   目前一部分数据源有白名单的限制，需要对数据集成的访问IP进行放行，比较常见的数据源如RDS、MongoDB和Redis等，需要在相应的控制台对这边的IP进行开放。一般添加白名单有以下两种情况：

   - 同步任务运行在自定资源组上，要给自定资源组机器授权，将自定义机器内/外网IP添加数据源的白名单列表。
   - 同步任务运行在默认资源组上，需要给底层运行机器授予访问权限，根据您选择DataWorks的Region来填写您需要添加的白名单，内容如下表所示。

region	白名单
华东1（杭州）	100.64.0.0/8,11.193.102.0/24,11.193.215.0/24,11.194.110.0/24,11.194.73.0/24,118.31.157.0/24,47.97.53.0/24,11.196.23.0/24,47.99.12.0/24,47.99.13.0/24,114.55.197.0/24,11.197.246.0/24,11.197.247.0/24

region	白名单
华东2（上海）	11.193.109.0/24,11.193.252.0/24,47.101.107.0/24,47.100.129.0/24,106.15.14.0/24,10.117.28.203,10.117.39.238,10.143.32.0/24,10.152.69.0/24,10.153.136.0/24,10.27.63.15,10.27.63.38,10.27.63.41,10.27.63.60,10.46.64.81,10.46.67.156,11.192.97.0/24,11.192.98.0/24,11.193.102.0/24,11.218.89.0/24,11.218.96.0/24,11.219.217.0/24,11.219.218.0/24,11.219.219.0/24,11.219.233.0/24,11.219.234.0/24,118.178.142.154,118.178.56.228,118.178.59.233,118.178.84.74,120.27.160.26,120.27.160.81,121.43.110.160,121.43.112.137,100.64.0.0/8
华南1（深圳）	100.106.46.0/24,100.106.49.0/24,10.152.27.0/24,10.152.28.0/24,11.192.91.0/24,11.192.96.0/24,11.193.103.0/24,100.64.0.0/8,120.76.104.0/24,120.76.91.0/24,120.78.45.0/24
中国（香港）	10.152.162.0/24,11.192.196.0/24,11.193.11.0/24,100.64.0.0/8,11.192.196.0/24,47.89.61.0/24,47.91.171.0/24,11.193.118.0/24,47.75.228.0/24
亚太东南1（新加坡）	100.106.10.0/24,100.106.35.0/24,10.151.234.0/24,10.151.238.0/24,10.152.248.0/24,11.192.153.0/24,11.192.40.0/24,11.193.8.0/24,100.64.0.0/8,100.106.10.0/24,100.106.35.0/24,10.151.234.0/24,10.151.238.0/24,10.152.248.0/24,11.192.40.0/24,47.88.147.0/24,47.88.235.0/24,11.193.162.0/24,11.193.163.0/24,11.193.220.0/24,11.193.158.0/24,47.74.162.0/24,47.74.203.0/24,47.74.161.0/24,11.197.188.0/24
亚太东南2（澳洲、悉尼）	11.192.100.0/24,11.192.134.0/24,11.192.135.0/24,11.192.184.0/24,11.192.99.0/24,100.64.0.0/8,47.91.49.0/24,47.91.50.0/24,11.193.165.0/24,47.91.60.0/24
华北2（北京）	100.106.48.0/24,10.152.167.0/24,10.152.168.0/24,11.193.50.0/24,11.193.75.0/24,11.193.82.0/24,11.193.99.0/24,100.64.0.0/8,47.93.110.0/24,47.94.185.0/24,47.95.63.0/24,11.197.231.0/24,11.195.172.0/24,47.94.49.0/24,182.92.144.0/24
美国西部1	10.152.160.0/24,100.64.0.0/8,47.89.224.0/24,11.193.216.0/24,47.88.108.0/24
美国东部1	11.193.203.0/24,11.194.68.0/24,11.194.69.0/24,100.64.0.0/8,47.252.55.0/24,47.252.88.0/24

region	白名单
亚太东南3（马来西亚、吉隆坡）	11.193.188.0/24,11.221.205.0/24,11.221.206.0/24,11.221.207.0/24,100.64.0.0/8,11.214.81.0/24,47.254.212.0/24,11.193.189.0/24
欧洲中部1（德国、法兰克福）	11.192.116.0/24,11.192.168.0/24,11.192.169.0/24,11.192.170.0/24,11.193.106.0/24,100.64.0.0/8,11.192.116.14,11.192.116.142,11.192.116.160,11.192.116.75,11.192.170.27,47.91.82.22,47.91.83.74,47.91.83.93,47.91.84.11,47.91.84.110,47.91.84.82,11.193.167.0/24,47.254.138.0/24
亚太东北1（日本）	100.105.55.0/24,11.192.147.0/24,11.192.148.0/24,11.192.149.0/24,100.64.0.0/8,47.91.12.0/24,47.91.13.0/24,47.91.9.0/24,11.199.250.0/24,47.91.27.0/24
中东东部1（阿联酋、迪拜）	11.192.107.0/24,11.192.127.0/24,11.192.88.0/24,11.193.246.0/24,47.91.116.0/24,100.64.0.0/8
亚太东南1（印度、孟买）	11.194.10.0/24,11.246.70.0/24,11.246.71.0/24,11.246.73.0/24,11.246.74.0/24,100.64.0.0/8,149.129.164.0/24,11.194.11.0/24
英国	11.199.93.0/24,100.64.0.0/8
亚太东南5（印度尼西亚、雅加达）	11.194.49.0/24,11.200.93.0/24,11.200.95.0/24,11.200.97.0/24,100.64.0.0/8,149.129.228.0/24,10.143.32.0/24,11.194.50.0/24
华北2（政务云）	11.194.116.0/24,100.64.0.0/8 如果IP地址段添加不成功，请添加IP地址：11.194.116.160,11.194.116.161,11.194.116.162,11.194.116.163,11.194.116.164,11.194.116.165,11.194.116.167,11.194.116.169,11.194.116.170,11.194.116.171,11.194.116.172,11.194.116.173,11.194.116.174,11.194.116.175

RDS添加白名单

RDS数据源可以通过以下2种方式进行配置：

- RDS实例形式

  通过RDS实例创建数据源，目前是支持测试连通性（其中包括VPC环境的RDS）。如果RDS实例形式测试连通性失败，可以尝试用JDBCUrl形式添加数据源。

- JDBCUrl形式

  JDBCUrl中的IP请优先填写内网地址，若没有内网地址请填写外网地址。其中，内网地址是走阿里云机房内网同步时同步速度会更快，外网地址在同步时同步速度受限于您开通外网带宽。

配置RDS白名单

数据集成连接RDS同步数据需要使数据库标准协议连接数据库。RDS默认允许所有IP连接，但如果您在RDS配置指定了IP白名单，您需要添加数据集成执行节点IP白名单。如果您没有指定RDS白名单，不需要给数据集成提供白名单。

如果您设置了RDS的IP白名单，请进入RDS管理控制台，并导航至安全控制，根据上面的白名单列表进行白名单设置。

说明:

如果使用自定义资源组调度RDS的数据同步任务，必须把自定义资源组的机器IP也加到RDS的白名单中。

2.4.3 新增任务资源

DataWorks可通过免费传输能力（默认任务资源组），进行海量数据上云，但默认资源组无法实现传输速度存在较高要求或复杂环境中的数据源同步上云的需求。您可以新增自定义的任务资源运行数据同步任务，解决DataWorks默认资源组与您的数据源不通的问题，或实现更高速度的传输能力。

项目管理员可以在数据集成 > 同步资源管理 > 资源组页面新增或修改任务资源。

当默认任务资源无法与您的复杂的网络环境连通时，可通过数据集成自定义资源的部署，打通任意网络环境之间的数据传输同步，详情请参见#unique_229和#unique_230。

说明:

- 您在数据集成 > 同步资源管理 > 资源组页面增加的任务资源，只能给当前工作空间作为数据同步资源组使用，不会显示在调度资源列表。目前该页面添加的任务资源不支持手动业务流程数据同步节点。
- 添加自定义资源时一台机器只能添加一个自定义资源组，每个自定义资源组只能选择一种网络类型。
- 注册服务器时，只有华东2可以选择经典网络的方式注册（输入主机名），建议您优先使用专有网络VPC。其他Region只能选择专有网络方式注册（输入UUID）。
- 自定义资源组上运行的部分文件需要Admin权限。例如，在您自己写的Shell脚本任务中调用自定义ECS上的Shell文件、SQL文件等。

- 因为调度资源组主要用于调度任务，资源有限，并不适合用来完成计算任务，所以不推荐在调度资源组上安装数据处理模块。MaxCompute具有海量数据处理能力，推荐您通过MaxCompute进行大数据计算。

购买云服务器ECS

购买ECS云服务器的具体操作请参见购买ECS云服务器。

说明:

- 使用CentOS 6、CentOS 7或Aliyun OS。
- 如果您添加的ECS需要执行MaxCompute任务或同步任务，需要检查当前ECS的Python版本是否是Python2.6或2.7（CentOS 5的版本为Python 2.4，其它OS自带Python 2.6以上版本）。
- 请确保ECS有访问公网能力，您可将是否ping通`www.aliyun.com`作为衡量标准。
- 建议ECS的配置为8核16G。

查看ECS主机名和内网IP地址

您可进入云服务器ECS > 实例页面，查看购买的ECS主机名和IP。

开通8000端口，以便读取日志

说明:

如果您的ECS是VPC专有网络类型，则不需开通8000端口。下列步骤仅适用于经典网络。

1. 添加安全组规则

   进入云服务器ECS > 网络和安全 > 安全组页面，单击配置规则，进入配置规则页面 。

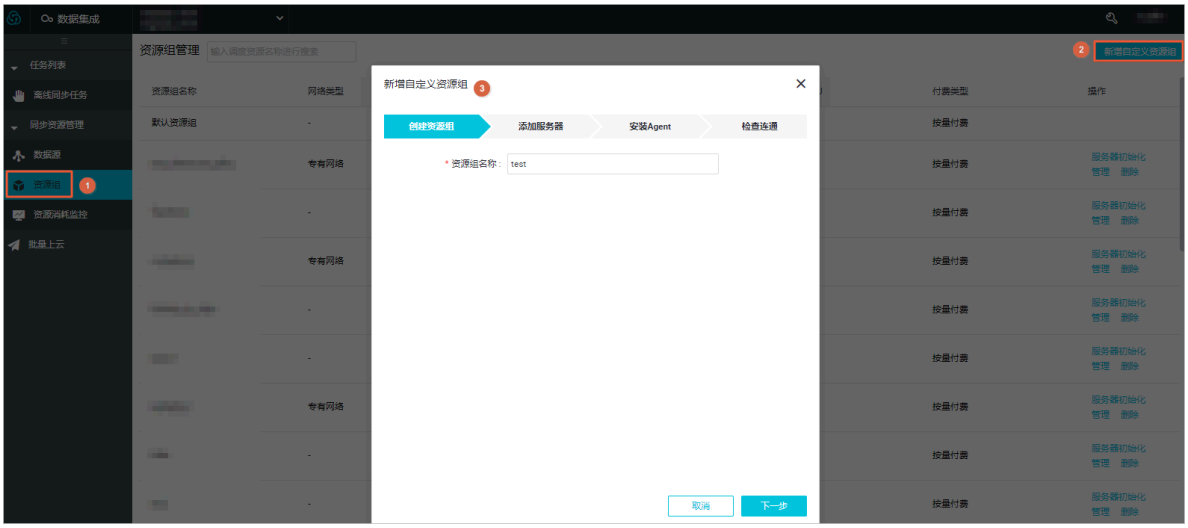
2. 进入安全组规则 > 内网入方向页面，单击右上角的添加安全组规则。

3. 填写添加安全组规则对话框中的配置信息，配置IP为数据集成的固定IP ，访问端口为8000。

新增任务资源

1. 以开发者身份进入DataWorks管理控制台，单击对应工作空间操作栏中的进入数据集成。

2. 选择同步资源管理 > 资源组，单击新增自定义资源。



3. 单击下一步，在添加服务器对话框中，填写购买的ECS云服务器的主机IP等信息。

配置	说明
网络类型	目前除上海Region支持经典网络外，其他Region均只支持专有网络。
ECS UUID	登录ECS，执行`dmidecode | grep UUID`，取返回值。
机器IP	请输入内网机器IP。
机器CPU（核）	推荐的自定义资源组机器CPU配置至少为4核。

配置	说明
机器内存（GB）	推荐的自定义资源组机器内存配置至少为8GB RAM和80GB磁盘。

说明:

- 填写专有网络下的ECS作为服务器时，需要填写ECS的UUID作为服务器名称。登录到ECS机器执行`dmidecode | grep UUID`即可获取。
- 例如执行`dmidecode | grep UUID`，返回结果是UUID: 713F4718-8446-4433-A8EC-6B5B62D7****，则对应的UUID为713F4718-8446-4433-A8EC-6B5B62D7****。

4. 安装Agent并初始化。

如果是新添加的服务器，请按照如下步骤进行操作。

a. SSH登录ECS服务器，保持在root用户下。

b. 执行下述命令：

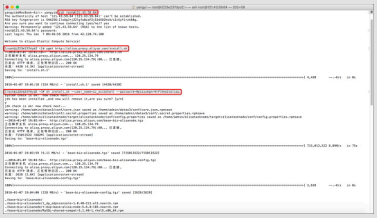
```
chown admin:admin /opt/taobao  //用于给admin用户授予/opt/taobao目录权限。
wget https://alisaproxy.shuju.aliyun.com/install.sh --no-check-certificate
```

```
sh install.sh --user_name=*****19d --password=****h1bm --
enable_uuid=false
```

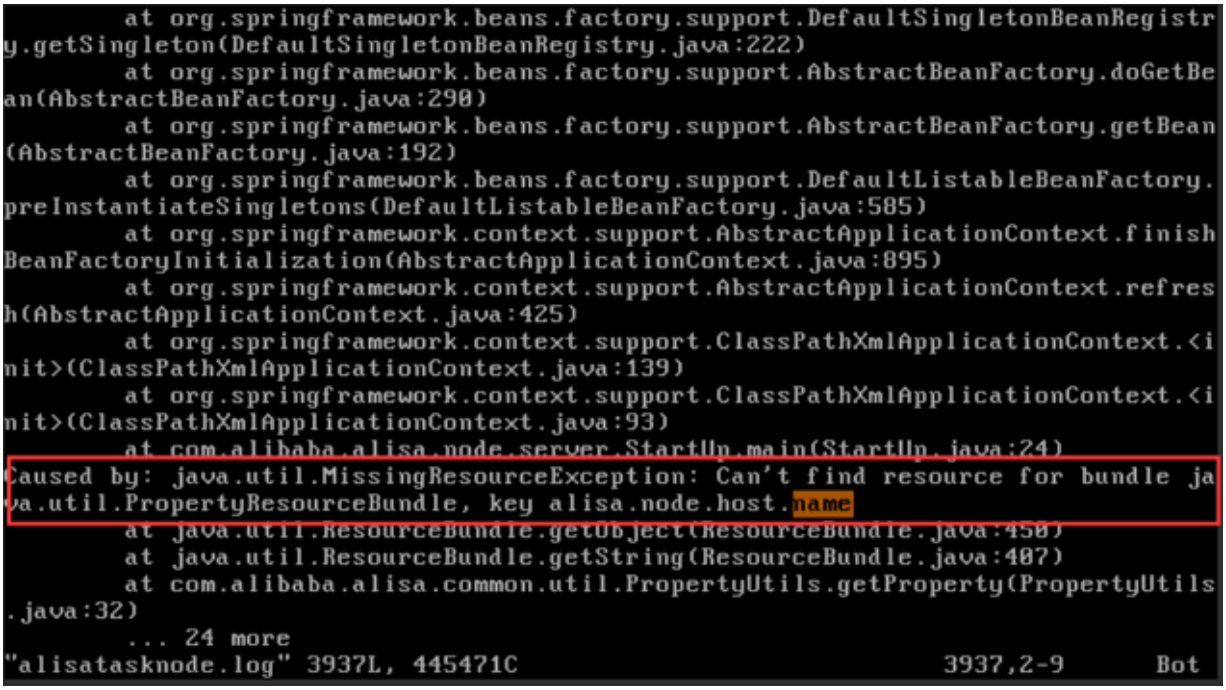
c. 稍后在添加服务器页面，单击刷新，查看服务状态是否转为可用。

d. 开通服务器的8000端口。

> 说明:
> 如果执行install.sh过程中出错或需要重新执行，请在install.sh的同一个目录下执行`rm -rf install.sh`，删除已经生成的文件。然后执行`install.sh`。上面的初始化界面对于每个用户的命令都不一样，请根据自己的初始化界面执行相关命令。



执行完上述操作后，如果服务状态一直是停止，您可能碰到以下问题。



上图的错误原因是没有绑定host，请参见以下步骤进行修改。

1. 切换到admin账号。
2. 执行`hostname -i`，查看host的绑定情况。
3. 执行`vim/etc/hosts`，添加IP地址和主机名。
4. 刷新页面服务状态，查看ECS服务器注册是否成功。

> 说明:

- 如果刷新后还是停止状态，您可以重启alisa。

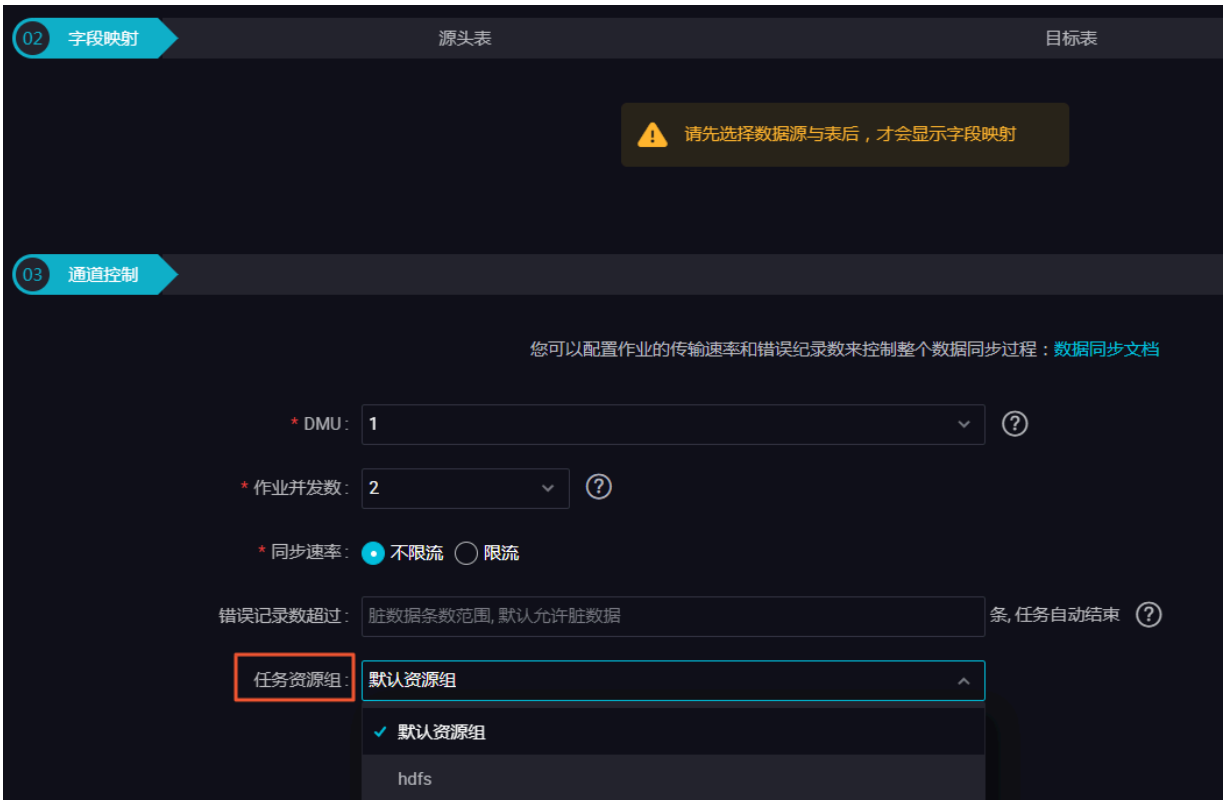
  切换到admin账号，执行下述命令。

  ```
  /home/admin/alisatasknode/target/alisatasknode/bin/serverctl
  restart
  ```

- 命令中涉及到您的AK信息，请不要轻易暴露给他人。

数据同步选择任务资源组

在数据同步任务中的通道控制选择任务资源组。



使用限制

- 自定义任务资源所在的ECS服务器的时间与当前互联网时间差必须在2分钟之内，否则会导致部署的自定义任务资源服务请求接口超时服务异常，无法执行任务。
- 如果您发现alisatasknode日志中有超时报错信息`response code is not 200`，通常是因为某个时段访问服务接口不稳定的异常导致。只要不是持续10分钟异常，自定义资源组服务器就依然可以正常服务。您可以查阅日志`/home/admin/alisatasknode/logs/heartbeat.log`进行确认。

2.5 整库迁移

2.5.1 整库迁移概述

本文将为您介绍整库迁移的任务生成规则和约束限制。

整库迁移是帮助提升用户效率、降低用户使用成本的一种快捷工具，它可以快速把一个MySQL数据库内所有表一并上传到MaxCompute的工作，节省大量初始化数据上云的批量任务创建时间。

假设数据库内有100张表，您原本可能需要配置100次数据同步任务，但有了整库迁移便可以一次性完成。同时，由于数据库的表设计规范性的问题，此工具并无法保证一定可以一次性完成所有表按照业务需求进行同步的工作，即它有一定的约束性。

任务生成规则

完成配置后，根据选择的需要同步的表，依次创建MaxCompute表，生成数据同步任务。

MaxCompute表的表名、字段名和字段类型根据高级配置生成，如果没有填写高级配置，则与MySQL表的结构完全相同。表的分区为pt，格式为yyyymmdd。

生成的数据同步任务是按天调度的周期任务，会在第二天凌晨自动运行，传输速率为1M/s，它在细节上会因为同步的方式、并发配置等有所不同，您可以在同步任务目录树的clone_database > 数据源名称 > mysql2odps_表名中找到生成的任务，然后对其进行更加个性化的编辑操作。

说明:

建议您当天对数据同步任务进行冒烟测试，相关任务节点可以在运维中心 > 任务管理中的project_etl_start > 整库迁移 > 数据源名称 下找到所有此数据源生成的同步任务，然后右键单击，测试相应的节点即可。

约束限制

由于数据库的表设计规范性的问题，整库迁移具有一定的约束性。

- 目前仅提供MySQL和Oracle数据源的整库迁移至MaxCompute，后续Hadoop、Hive数据源功能会逐渐开放。

- 仅提供每日增量、每日全量的上传方式。

  如果您需要一次性同步历史数据，则此功能无法满足您的需求，建议如下：

  - 建议您配置为每日任务，而非一次性同步历史数据。您可以通过调度提供的补数据，来对历史数据进行追溯，这样可避免全量同步历史数据后，还需要做临时的SQL任务来拆分数据。
  - 如果您需要一次性同步历史数据，可以在任务开发页面进行任务的配置，然后单击运行。完成后通过SQL语句进行数据的转换，因为这两个操作均为一次性行为。

  如果您每日增量上传有特殊业务逻辑，而非一个单纯的日期字段可以标识增量，则此功能无法满足您的需求，建议如下：

  - 数据库数据的增量上传有两种方式：通过binlog（DTS产品可提供）和数据库提供数据变更的日期字段来实现。

    目前数据集成支持的为后者，所以要求您的数据库有数据变更的日期字段，通过日期字段，系统会识别您的数据是否为业务日期当天变更，即可同步所有的变更数据。
  - 为了更方便地增量上传，建议您在创建所有数据库表的时候都有gmt_create和gmt_modify字段，同时为了效率更高，建议增加id为主键。
- 整库迁移提供分批和整批迁移的方式

  分批上传为时间间隔，目前不提供数据源的连接池保护功能，此功能正在规划中。

  - 为了保障对数据库的压力负载，整库迁移提供了分批迁移的方式，您可以按照时间间隔把表拆分为几批运行，避免对数据库的负载过大，影响正常的业务能力。建议如下：
    - 如果您有主、备库，建议同步任务全部同步备库数据。
    - 批量任务中每张表都会有1个数据库连接，上限速度为1M/s。如果您同时运行100张表的同步任务，就会有100个数据库进行连接，建议您根据自己的业务情况谨慎选择并发数。
  - 如果您对任务传输效率有自己特定的要求，此功能无法实现您的需求。所有生成任务的上限速度均为1M/s。
- 仅提供整体的表名、字段名和字段类型映射

  整库迁移会自动创建MaxCompute表，分区字段为pt，类型为字符串String，格式为yyyymmdd。

  说明：

  选择表时必须同步所有字段，它不能对字段进行编辑。

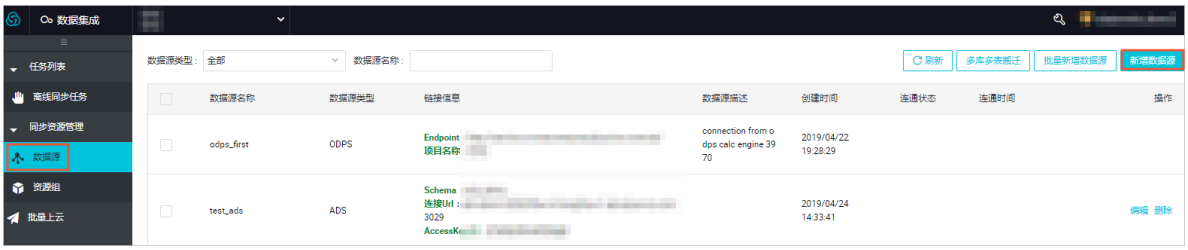
2.5.2 配置MySQL整库迁移

本文将为您介绍如何通过整库迁移功能，将MySQL数据整库迁移至MaxCompute。

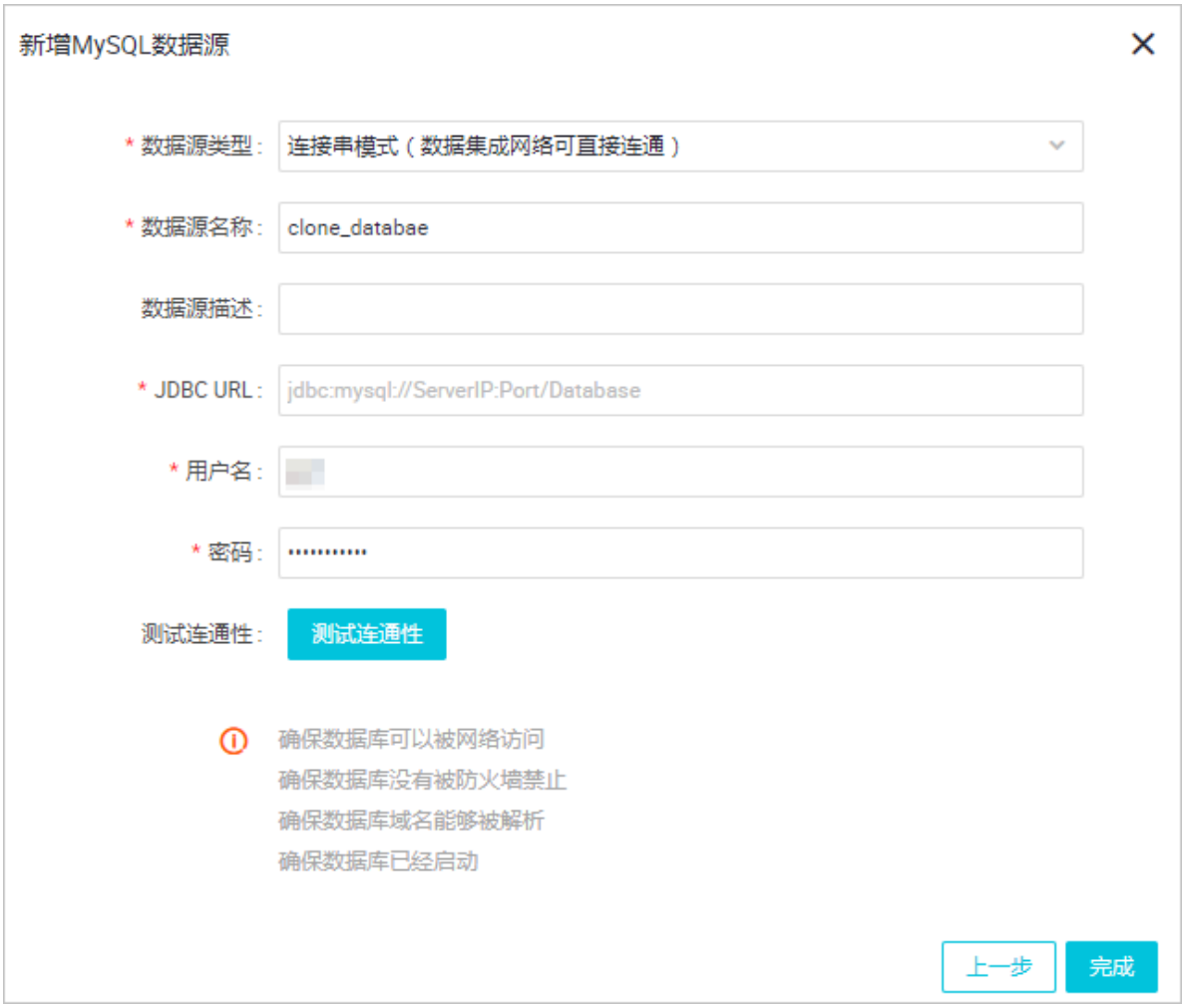
整库迁移是为了提升用户效率、降低用户使用成本的一种快捷工具，它可以快速把MySQL数据库内所有表一并上传至MaxCompute。整库迁移的详细介绍请参见#unique_235。

操作步骤

1. 登录DataWorks控制台，单击相应工作空间后的进入数据集成。
2. 选择左侧导航栏中的同步资源管理 > 数据源，进入数据源管理页面。

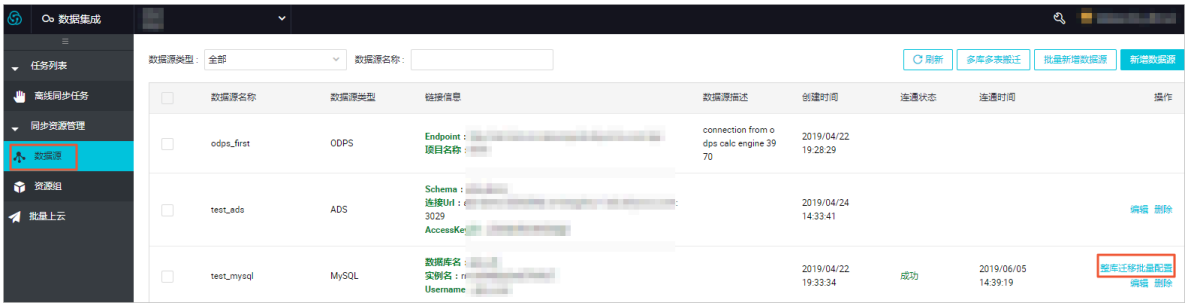


3. 单击右上角的新增数据源，添加一个面向整库迁移的MySQL数据源clone_databae。



4. 单击测试连通性，验证数据源访问正确无误后，确认并保存该数据源。

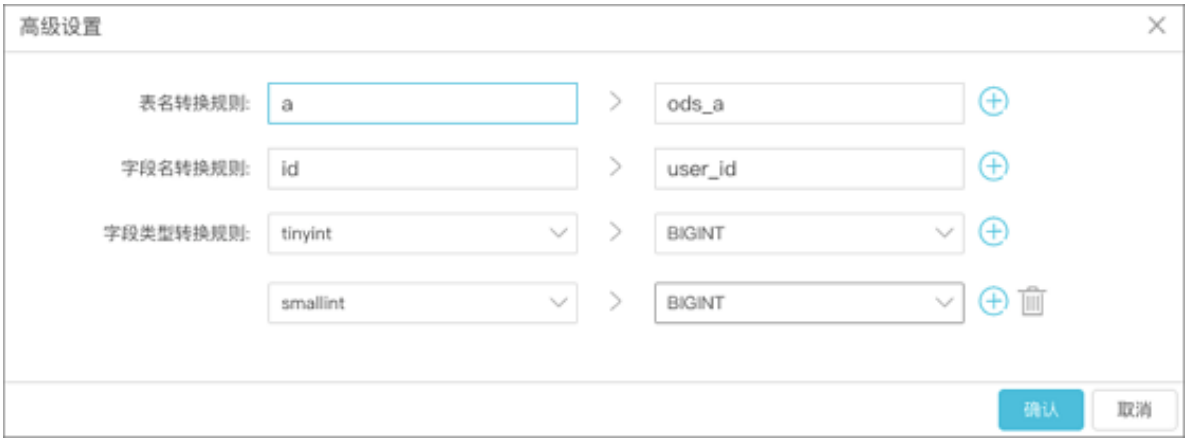
5. 新增数据源成功后，即可在数据源列表中看到新增的MySQL数据源clone_databae。单击对应MySQL数据源后的整库迁移批量配置，即可进入对应数据源的整库迁移功能页面。



整库迁移页面主要分3块功能区域。

序号	功能区域	说明
1	待迁移表筛选区	此处将MySQL数据源clone_databae下的所有数据库表以表格的形式展现出来，您可以根据实际需要批量选择待迁移的数据库表。
2	高级设置	此处提供了MySQL数据表和MaxCompute数据表的表名称、列名称、列类型的映射转换规则。
3	迁移模式、并发控制区	此处可以控制整库迁移的模式（全量、增量）、并发度配置（分批上次、整批上传）、提交迁移任务进度状态信息等。

6. 单击高级设置，您可以根据具体的需求选择转换规则。例如MaxCompute端建表时统一增加了ods_这一前缀。



7. 在迁移模式、并发控制区中，选择同步方式为每日增量，并配置增量字段为gmt_modified，数据集成默认会根据您选择的增量字段生成具体每个任务的增量抽取where条件，并配合DataWorks调度参数比如${bdp.system.bizdate}形成针对每天的数据抽取条件。

数据集成抽取MySQL库表的数据是通过JDBC连接远程MySQL数据库，并执行相应的SQL语句，将数据从MySQL库中Select出来。由于是标准的SQL抽取语句，可以配置Where子句控制数据范围。此处您可以查看到增量抽取的Where条件如下所示：

```
STR_TO_DATE('${bdp.system.bizdate}', '%Y%m%d') <= gmt_modified AND
gmt_modified < DATE_ADD(STR_TO_DATE('${bdp.system.bizdate}', '%Y%m%d
'), interval 1 day)
```

为了对源头MySQL数据源进行保护，避免同一时间点启动大量数据同步作业带来数据库压力过大，此处选择分批上传模式，并配置从每日0点开始，每1小时启动3个数据库表同步。

最后单击提交任务，查看迁移进度信息，以及每一个表的迁移任务状态。

8. 单击a1表对应的迁移任务，跳转至数据开发页面，查看迁移结果。

此时便完成了将一个MySQL数据源clone_databae整库迁移到MaxCompute的工作。这些任务会根据配置的调度周期（默认天调度）被调度执行，您也可以使用DataWorks调度补数据功能完成历史数据的传输。通过数据集成 > 整库迁移功能可以极大减少您初始化上云的配置、迁移成本。

查看整库迁移a1表任务执行成功的日志。

```
PHASE                 |    AVERAGE RECORDS |      AVERAGE BYTES |        MAX RECORDS | MAX RECORD`S BYTES |         MAX TASK ID |
MAX TASK INFO
READ_TASK_DATA        |              56345 |           128.12K |              56345 |           128.12K |               0-0-0 |
al,jdbcUrl:[jdbc:mysql://dataxtest.mysql.rds.aliyuncs.com:3306/base_cdp]
2017-05-11 20:43:47.907 [job-31340023] INFO  LocalJobContainerCommunicator - Total 56345 records, 128121 bytes | Speed 62.56KB/
s, 28172 records/s | Error 0 records, 0 bytes |  All Task WaitWriterTime 0.486s |  All Task WaitReaderTime 0.082s | Percentage
 100.00%
2017-05-11 20:43:47.908 [job-31340023] INFO  LogReportUtil - report datax log is turn off
2017-05-11 20:43:47.908 [job-31340023] INFO  JobContainer -
任务启动时刻                    : 2017-05-11 20:43:42
任务结束时刻                    : 2017-05-11 20:43:47
任务总计耗时                    :                  5s
任务平均流量                    :           62.56KB/s
记录写入速度                    :          28172rec/s
读出记录总数                    :              56345
读写失败总数                    :                  0
2017-05-11 20:43:47 INFO ==================================================================
2017-05-11 20:43:47 INFO Exit code of the Shell command 0
2017-05-11 20:43:47 INFO --- Invocation of Shell command completed ---
2017-05-11 20:43:47 INFO Shell run successfully!
```

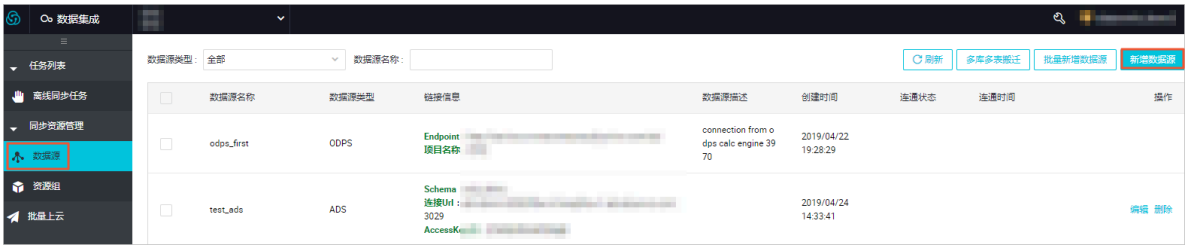
2.5.3 配置Oracle整库迁移

本文将为您介绍如何通过整库迁移功能，将Oracle数据整库迁移至MaxCompute。

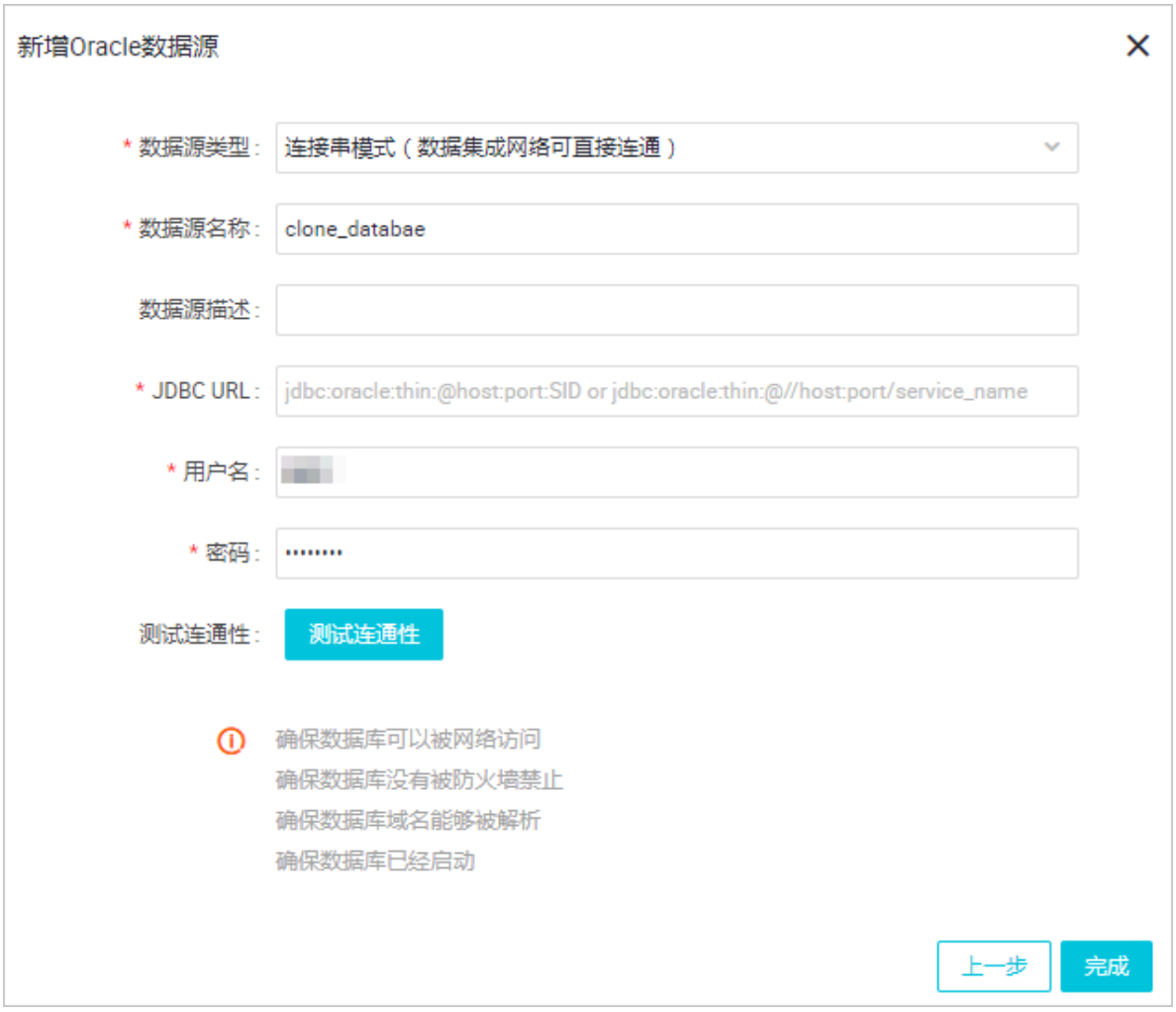
整库迁移是为了提升用户效率、降低用户使用成本的一种快捷工具，它可以快速把Oracle数据库内所有表一并上传至MaxCompute。整库迁移的详细介绍请参见#unique_237。

操作步骤

1. 登录DataWorks控制台，单击相应工作空间后的进入数据集成。
2. 选择左侧导航栏中的同步资源管理 > 数据源，进入数据源管理页面。



3. 单击右上角的新增数据源，添加一个面向整库迁移的Oracle数据源clone_databae。



4. 单击测试连通性，验证数据源访问正确无误后，确认并保存该数据源。

5. 新增数据源成功后，即可在数据源列表中看到新增的Oracle数据源clone_databae。单击对应Oracle数据源后的整库迁移批量配置，即可进入对应数据源的整库迁移功能页面。

   整库迁移页面主要分3块功能区域。

序号	功能区域	说明
1	待迁移表筛选区	此处将Oracle数据源clone_databae下的所有数据库表以表格的形式展现出来，您可根据实际需要批量选择待迁移的数据库表。
2	高级设置	此处提供了Oracle数据表和MaxCompute数据表的表名称、列名称、列类型的映射转换规则。
3	迁移模式、并发控制区	并发控制区：此处可以控制整库迁移的模式（全量、增量）、并发度配置（分批上次、整批上传）、提交迁移任务进度状态信息等。

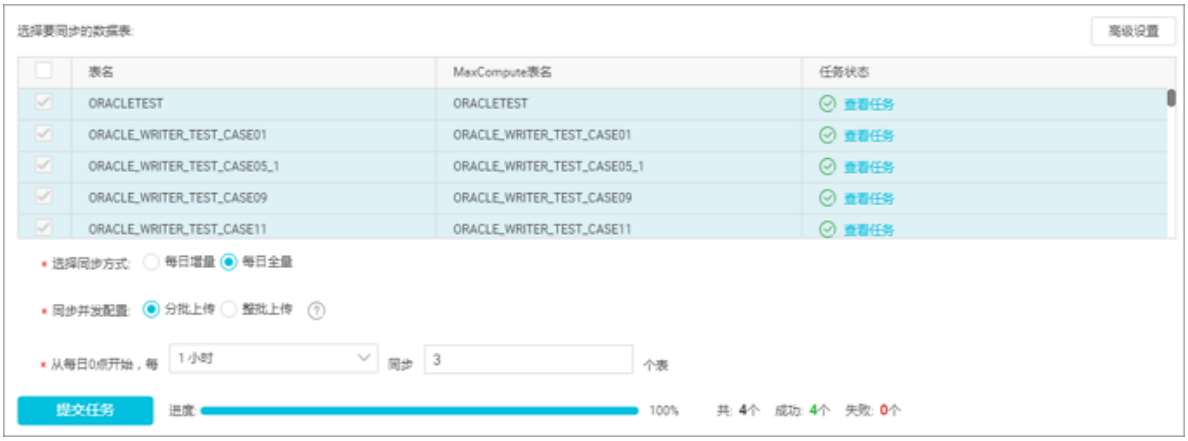
6. 单击高级设置，您可以根据具体的需求选择转换规则。
7. 在迁移模式、并发控制区中，选择同步方式为每日全量。

> 说明:

如果您的表中有日期字段，可以选择同步方式为每日增量，并配置增量字段为日期字段，数据集成默认会根据您选择的增量字段生成具体每个任务的增量抽取where条件，并配合DataWorks调度参数，例如${bdp.system.bizdate}形成针对每天的数据抽取条件。

为了对源头Oracle数据源进行保护，避免同一时间点启动大量数据同步作业导致数据库压力过大，此处选择分批上传模式，并配置从每日0点开始，每1小时启动3个数据库表同步。

最后单击提交任务，这里可以看到迁移进度信息，以及每一个表的迁移任务状态。



8. 单击表对应的查看任务，跳转至数据集成的任务开发页面，您可查看任务的运行详情。

此时便完成了将一个Oracle 数据源clone_databae整库迁移到MaxCompute的工作。这些任务会根据配置的调度周期（默认天调度）被调度执行，您也可以使用DataWorks调度补数据功能完成历史数据的传输。通过数据集成 > 整库迁移功能可以极大减少您初始化上云的配置、迁移成本。

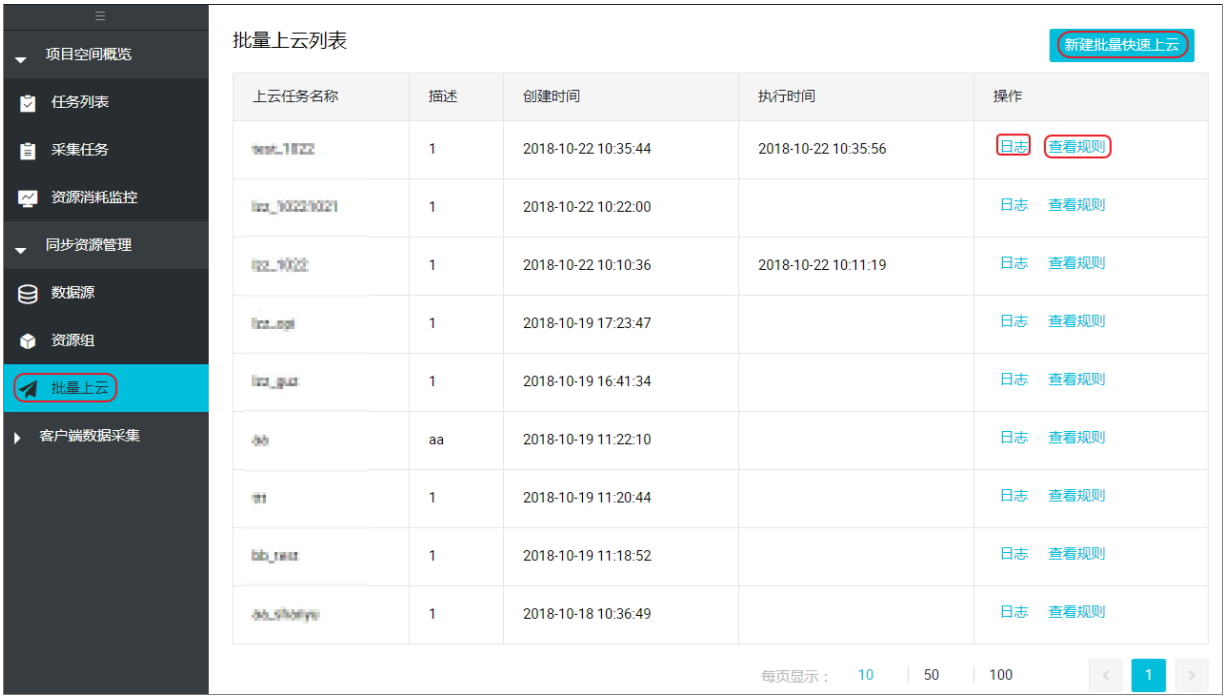
2.6 批量上云

2.6.1 批量上云

批量上云是帮您提升效率、降低使用成本的一种快捷工具，它可以快速把MySQL、Oracle、SQL Server数据库内的所有表一并上传到MaxCompute中，节省大量初始化数据上云的批量任务创建时间。

您可以灵活地配置表名转换、字段名转换、字段类型转换、目标表新增字段、目标表字段赋值、数据过滤、目标表名前缀等规则，来满足您的业务需求。

您可以进入数据集成 > 同步资源管理 > 批量上云页面，即可查看您配置的上云任务。



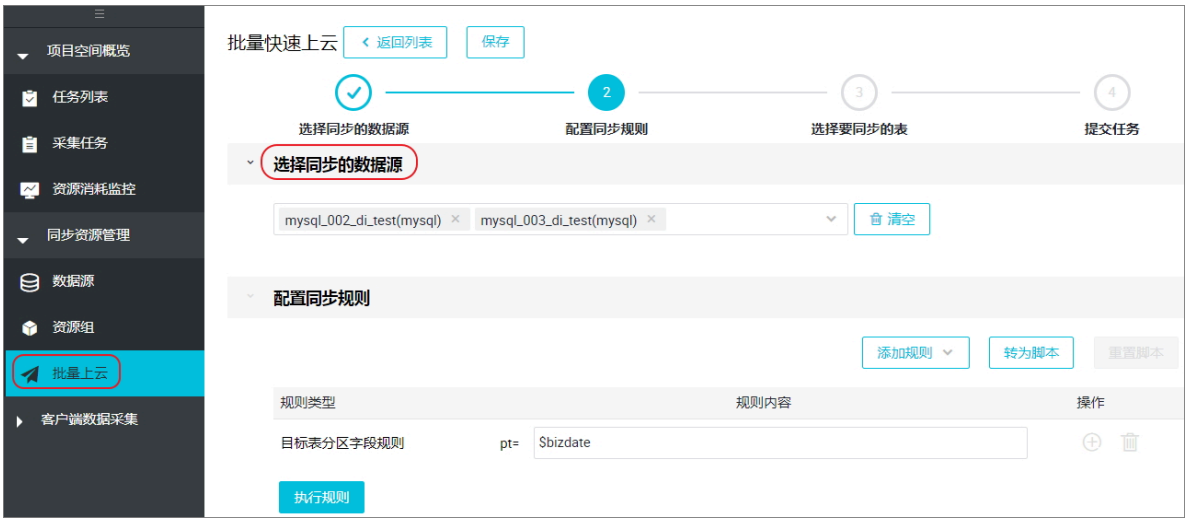
说明：

- 批量上云列表中，操作栏下的日志和规则只能查看不能修改。
- 如果您提交规则后，没有提交任务，则没有运行时间，并且此配置规则无效。

操作步骤

1. 选择同步的数据源。

   选择添加成功的同步数据源，此处可以选择多个数据源并且数据源类型相同，如都是MySQL、Oracle或SQL Server，详情请参见批量添加数据源。



2. 配置同步规则。

目前支持9个配置规则，您可以根据自身需求选择相应的规则配置，然后执行规则，并检查DDL和同步脚本确认规则效果。

> 说明:
>
> - 如果界面中的规则无法满足您的需求，可以尝试脚本模式。
> - 配置完规则后，您必须执行规则并提交任务，否则您配置的规则在刷新或关闭浏览器后没有相关的记录。
> - 如果您需要在批量上云时对表前缀进行设置，请参考#unique_241

操作	配置	说明
添加规则	目标表分区字段规则	展现分区的内容，符合调度参数配置，详情请参见#unique_39。
	表名转换规则	选择您的数据库表名的任何词，转换成您需要的内容。
	字段名转换规则	选择您的表中字段名的任何词，转换成您需要的内容。
	字段类型转换规则	选择您的数据源表中具有的数据类型，转换成您需要的数据类型。
	目标表新增字段规则	可以在MaxCompute表中增加一列，名称根据您的需求设定。
	目标表字段赋值规则	给您增加的字段赋值。
	数据过滤规则	针对您选择的数据源，对表中的数据进行过滤。

操作	配置	说明
	目标表名前缀规则	给表名添加一个前缀。
转为脚本	配置规则时可以转为脚本模式配置，与UI模式相比，单个规则可以指定作用范围。但UI模式转为脚本模式后，无法反向转换回UI配置模式。	
重置脚本	转换脚本后才能重置脚本，单击后提供统一的脚本模板。	
执行规则	单击执行规则，可以看到规则对DDL脚本和同步脚本的影响，此按钮不会创建任务，仅提供DDL和同步脚本的预览。 您可以选择一部分表检查对应的DDL和同步脚本，看是否符合规则。	

3. 选择要同步的表并提交。

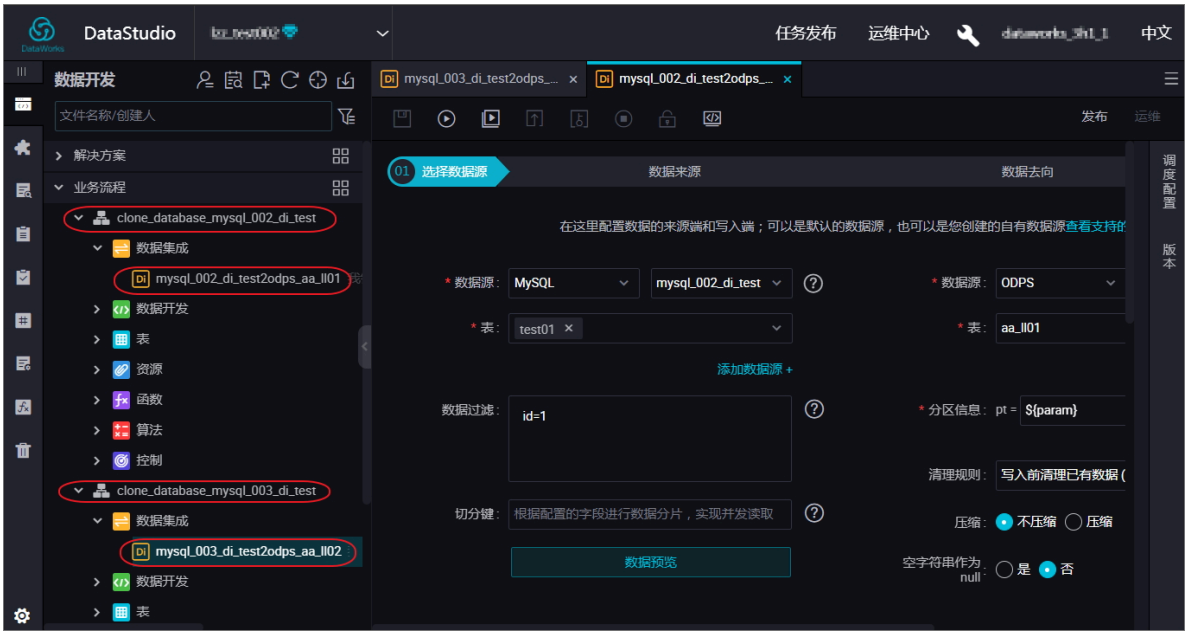
   您可以选择多个表进行批量提交，MaxCompute表会根据上面配置规则生成。如果执行失败您将鼠标放到执行结果上，会提示相关的原因。

配置	说明
DDL	单击后可以查看相关建表语句，只能查看不能修改。
同步配置	单击可以查看您配置的任务，以脚本模式展现。
查看表	跳转到相应的数据管理控制台页面，您可以查看MaxCompute建表的具体情况。

4. 查看任务。

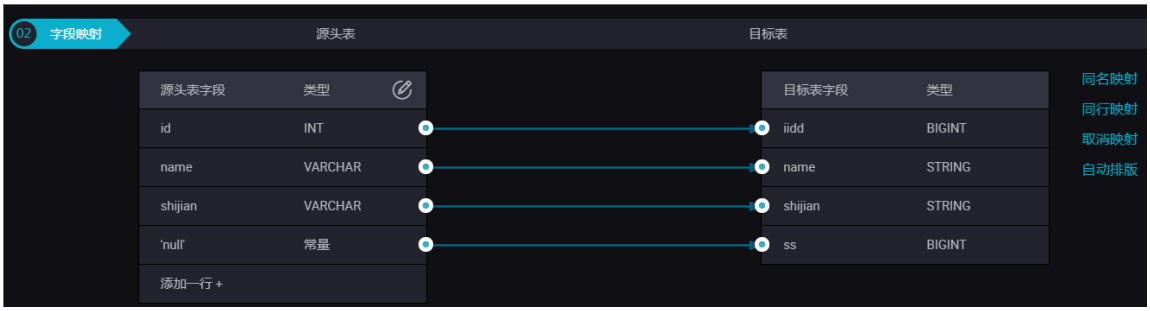
提交成功后，您可以进入数据开发 > 业务流程页面，查看您的批量上云任务。

您选择几个数据源，便会产生几个业务流程。一般命名规则是clone_database_数据源名。每张表会产生一个同步任务，命名规则是数据源名2odps_表名。



a. 任务配置：根据批量上云生成的MySQL同步到odps的同步任务，数据过滤条件是配置数据过滤规则后产生的。

b. 字段映射：目标端是根据您配置相关字段规则而产生，可以根据您配置的规则进行查看。



c. 通道配置。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

说明：

任务的具体配置请参见配置Reader插件和配置Writer插件。

5. 运行任务。

直接单击运行，同步任务会立刻运行。您也可单击提交，将同步任务提交到调度系统中，调度系统会按照配置属性在从第二天开始自动定时执行，详情请参见调度配置。

说明：

- 简单模式：提交之后直接到生产环境。
- 标准模式：提交后到开发环境，然后发布到生产环境。

2.6.2 批量添加数据源

本文将为您介绍如何批量添加数据源。

说明：

- 快速上云目前仅支持MySQL、Oracle和SQL Server三种类型的数据源。

- 批量添加数据源目前只能选择有公网IP。
- 添加MySQL和Oracle，SQL Server数据源后，需测试连通性，当连通状态为成功时，批量上云选择同步数据源列表才能选择此数据源。

1. 以项目管理员身份登录DataWorks控制台。
2. 单击对应项目后的进入数据集成。
3. 在数据集成 > 同步资源管理 > 数据源页面，单击新增数据源。
4. 在新增数据源对话框中分别选择MySQL、Oracle，SQL Server三个数据源，下图以新增MySQL数据源为例。

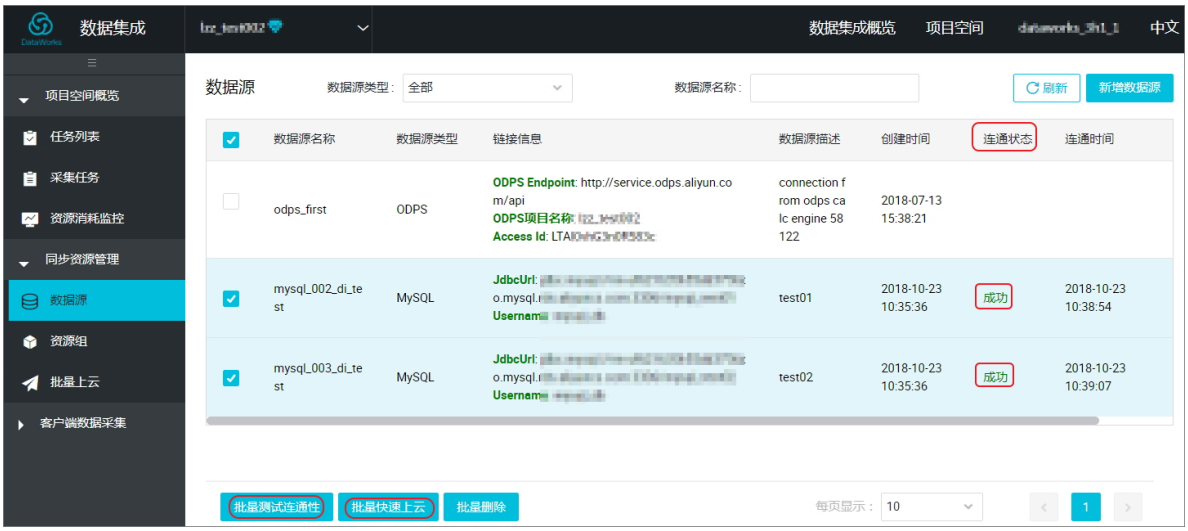
配置	说明
数据源类型	均选择有公网IP。
配置方式	均选择批量模式。
脚本上传	首先单击模板下载，在模板中添加您的数据源名，数据源描述，链接地址，用户名和密码。 说明: 一般会有一个默认的数据源mysql_001_di_test，你可以直接删除添加您自己的数据源。
选择文件	单击选择文件，选择修改好的模板。
开始创建	文件上传成功后，单击开始创建，您上传的结果会在文本框中展现，如成功个数、失败的个数、原因等。

5. 上传成功后，单击完成。
6. 在数据源列表页面，勾选相应数据源，单击批量测试连通性。

说明:

必须保证数据源的连通状态为成功，方可进行批量上云操作。

7. 勾选您要批量上传的数据源，单击批量上云。



2.7 最佳实践

2.7.1 （仅一端不通）数据源网络不通的情况下的数据同步

本文将通过实践操作，为您介绍如何使用整库迁移功能，将MySQL数据整库迁移到MaxCompute。

场景说明

复杂网络环境主要包含以下两种情况：

- 数据的来源端和目的端有一端为私网环境。
  - VPC环境（除如DS）<->公网环境
  - 金融云环境<->公网环境
  - 本地自建无公网环境<->公网环境
- 数据的来源端和目的端均为私网环境。
  - VPC环境（除RDS）<->VPC环境（除RDS）
  - 金融云环境<->金融云环境
  - 本地自建无公网环境<->本地自建无公网环境
  - 本地自建无公网环境<->VPC环境（除RDS）
  - 本地自建无公网环境<->金融云环境

您可以通过部署数据集成Agent，打通任意网络环境，在复杂的网络环境下完成数据传输和同步。本文主要为您介绍仅一端数据源网络不通情况下的操作方案，具体实现逻辑和操作步骤见下文所述，两端数据源均无法连通的情况请参见（两端都不通）数据源网络不通情况下的数据同步。

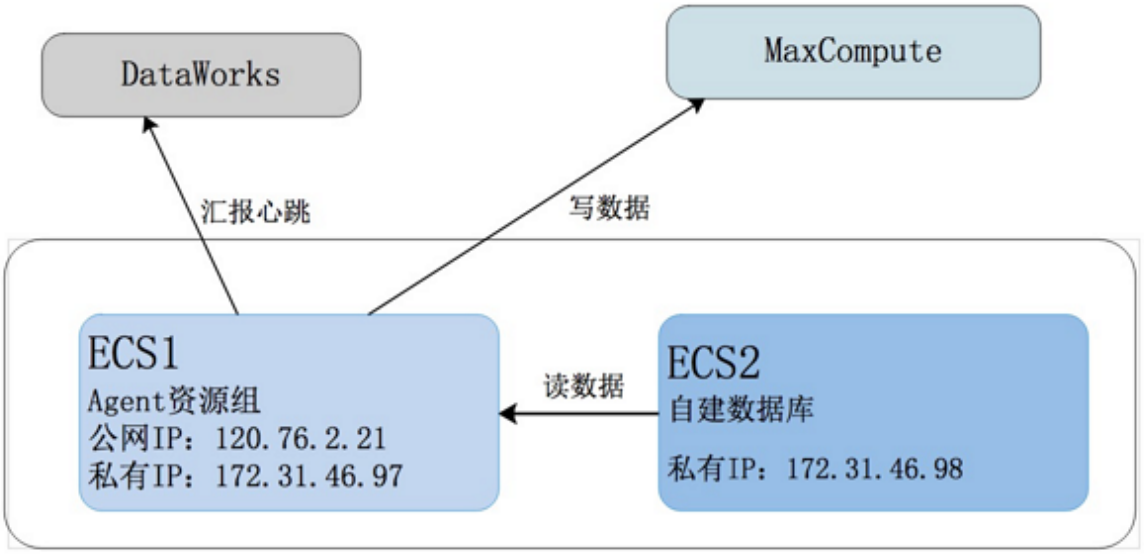
实现逻辑

针对第一种复杂网络环境，采用在私网环境的那一端相同网络环境下的机器上部署数据集成Agent，通过Agent与外部公网联通。私网环境通常有以下两种情况：

- 购买云服务ECS上搭建的数据库，没有分配公网IP或弹性公网IP。
- 本地IDC机房无公网IP。

云服务ECS

此场景下的数据同步方式，如下图所示。



- 由于ECS2服务器没有访问公网的能力，所以需要准备一台和ECS2在同一网段并且有访问公网能力的ECS1机器行部署Agent。
- 将ECS1作为资源组并且同步任务运行在此机器上。

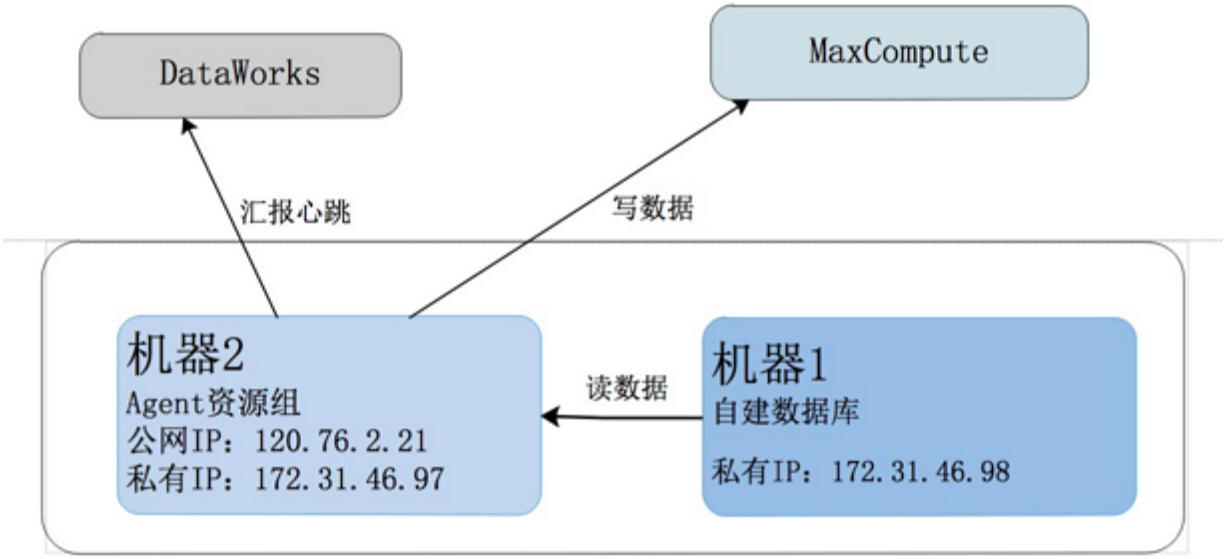
> **说明:**
> 您需要给数据库赋权限，让ECS2服务器能访问到相应的数据库，才能将此数据库的数据读取到ECS1中。授权命令如下所示。

```
grant all privileges on *.* to 'demo_test'@'%' identified by '密码';
 --> %号代表给所有 IP 授权<br>
```

ECS2上的自建数据源同步任务运行在自定资源组上，要给自定资源组机器授权，将ECS2机器内/外网IP和端口添加到ECS1安全组上，详情请参见#unique_108。

无公网IP本地IDC机房

此场景下的数据同步方式，如下图所示。



- 由于机器1没有访问公网的能力，所以需要准备一台和机器1在同一网段并且具备访问公网的能力机器2部署Agent。
- 将机器2作为任务资源组并且同步任务运行在此机器上。

配置数据源

1. 以开发者身份进入DataWorks管理控制台，单击对应项目操作栏中的进入数据集成。

2. 进入数据源页面，单击新增数据源，弹出支持的数据源类型。

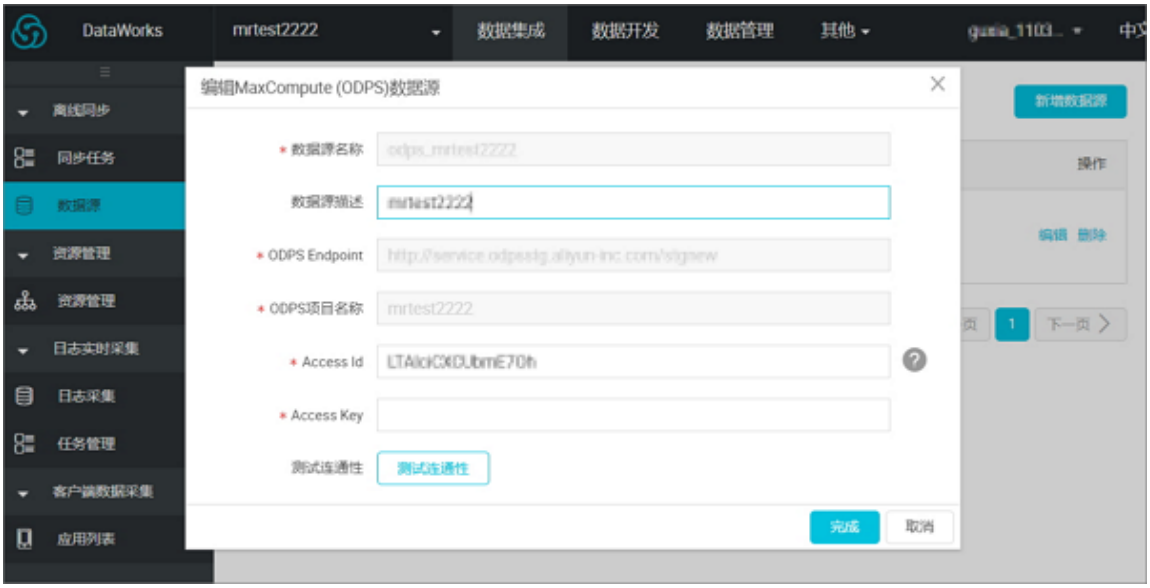
3. 选择关系数据库MySQL的数据源里的无公网IP的数据源类型添加数据源。

- 源端数据源（无公网IP）

配置	说明
数据源类型	无公网IP。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
资源组	是选择部署Agent的机器，通过Agent与外部公网联通，特殊网络环境的数据源可以将同步任务运行在资源组上。添加资源组的详情请参见#unique_33。
JDBC URL	JDBC连接信息，格式是`jdbc:mysql://ServerIP:Port/Database`。
用户名和密码	数据库对应的用户名和密码。

配置	说明
测试连通性	无公网IP的数据源不支持测试连通性，直接单击完成即可。

- 目标数据源（有公网环境）



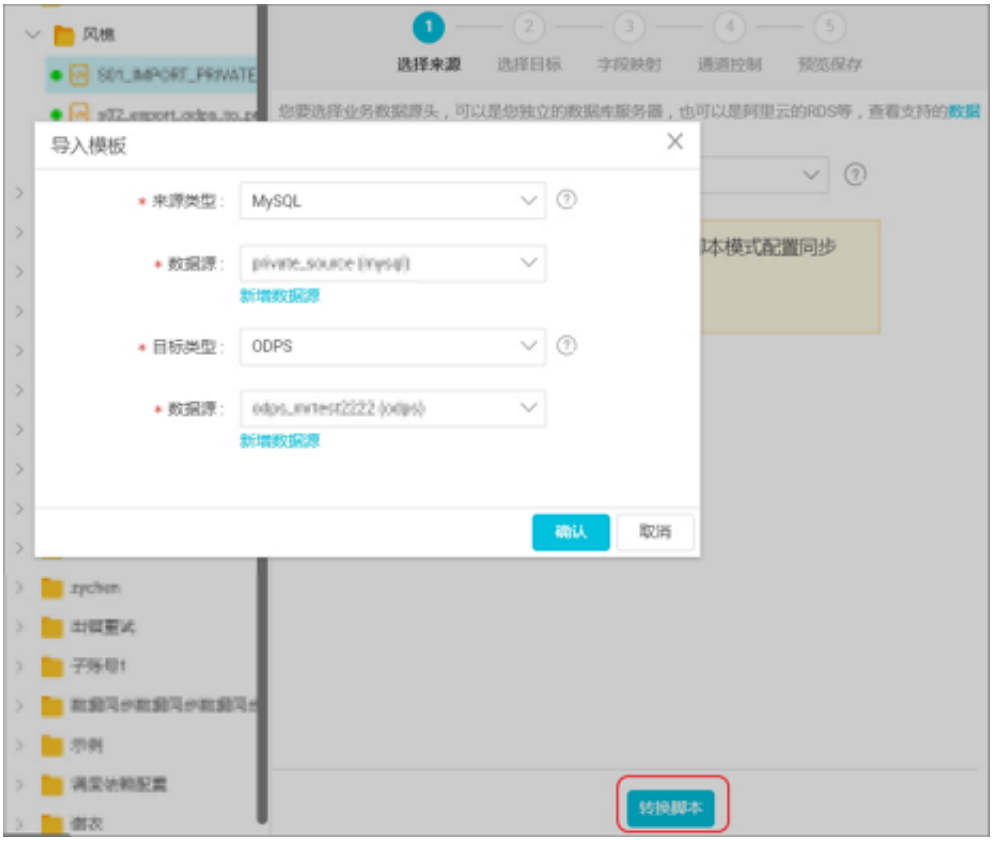
配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
ODPS Endpoint	默认只读，从系统配置中自动读取。
ODPS项目名称	对应的MaxCompute Project标识。
Access Id	与MaxCompute Project Owner云账号对应的AccessID。
Access Key	与MaxCompute Project Owner云账号对应的AccessKey，与AccessID成对使用。访问密钥相当于登录密码。
测试连通性	支持测试连通性。

配置同步任务

1. 选择来源

由于数据来源是无公网IP的数据源，所以此数据源网络无法联通，需要使用脚本模式配置同步任务，直接单击转换脚本按钮。

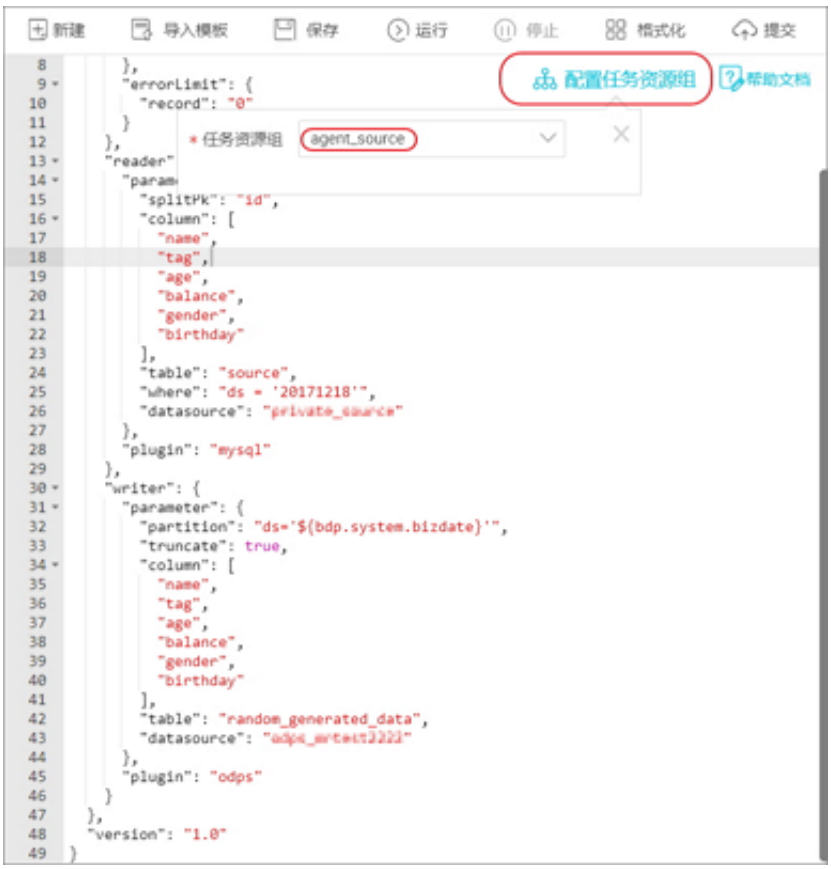
2. 导入模板



配置	说明
来源类型	根据您的向导模式选择的数据源直接填写数据源名称。

配置	说明
目标类型	您可以在下拉框中选择要写入的目标数据源。 说明: · 如果数据库支持界面添加数据源，可以在模板中选择。 · 如果数据库不支持界面添加数据源，则需在模板的JSON代码中编写相关的数据源信息，然后单击新增数据源。

3. 转换成脚本模式样例



配置任务资源组：可以修改和查看同步任务运行的资源组的情况，默认是收起状态。

```
{
"type": "job",
"configuration": {
 "setting": {
   "speed": {
     "concurrent": "1",//作业并发数
     "mbps": "1"//作业速率上限
   },
   "errorLimit": {
     "record": "0"//错误记录数超过值
   }
 },
 "reader": {
   "parameter": {
     "splitPk": "id",//切分键
     "column": [//目标端的列名
```

```
        "name",
        "tag",
        "age",
        "balance",
        "gender",
        "birthday"
      ],
      "table": "source",//源端的表名
      "where": "ds = '20171218'",//过滤条件
      "datasource": "private_source"//数据源名称，要跟添加的数据源名保持一致
    },
    "plugin": "mysql"
  },
  "writer": {
    "parameter": {
      "partition": "ds='${bdp.system.bizdate}'",//分区信息
      "truncate": true,
      "column": [//目标端的列
        "name",
        "tag",
        "age",
        "balance",
        "gender",
        "birthday"
      ],
      "table": "random_generated_data",//目标端的表名
      "datasource": "odps_mrtest2222"//数据源名称，要跟添加的数据源名保持一致
    },
    "plugin": "odps"
  }
},
"version": "1.0"
}
```

运行同步任务

您可通过以下两种方式运行任务：

- 数据集成的界面直接单击运行。
- 调度运行。提交调度的步骤请参见调度配置。

2.7.2 （两端都不通）数据源网络不通的情况下的数据同步

场景说明

复杂网络环境主要包含以下两种情况：

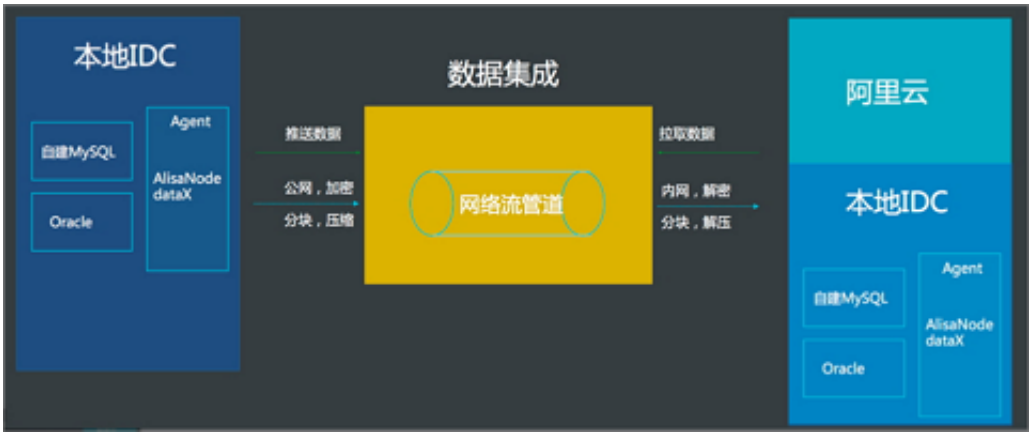
- 数据的来源端和目的端有一端为私网环境。
  - VPC环境（除RDS）<->公网环境
  - 金融云环境<->公网环境
  - 本地自建无公网环境<->公网环境
- 数据的来源端和目的端均为私网环境。
  - VPC环境（除RDS）<->VPC环境（除RDS）
  - 金融云环境<->金融云环境
  - 本地自建无公网环境<->本地自建无公网环境
  - 本地自建无公网环境<->VPC环境（除RDS）
  - 本地自建无公网环境<->金融云环境

数据集成提供复杂网络环境下的穿墙过壁能力，主要是通过数据集成Agent的部署，打通任意网络环境之间的数据传输同步。具体实现逻辑和操作步骤见下文所述，本文主要为您介绍两端数据源均无法连通的情况下的操作方案，仅一端数据源无法连通的情况请参见#unique_34。

实现逻辑

针对第二种复杂网络环境，采用在两端数据源的相同网络环境下均部署数据集成Agent，来源端Agent负责推送数据至数据集成服务端，目的端Agent负责拉取数据至本地，且数据在传输过程中会进行数据的分块、压缩、加密，保障数据传输的及时性和安全性。

此场景的数据同步方式，如下图所示：



操作步骤

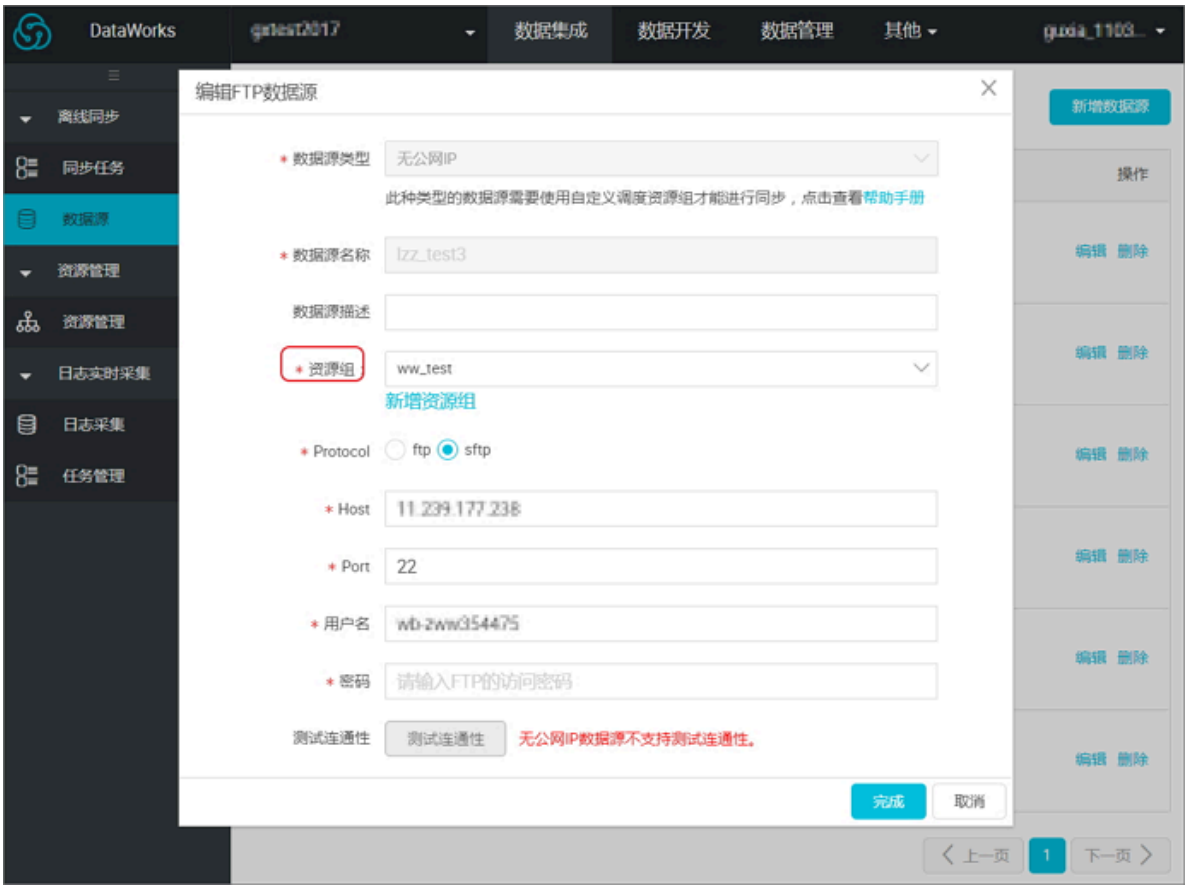
配置数据源

1. 以开发者身份进入DataWorks管理控制台，单击对应项目操作栏中的进入数据集成。

2. 进入数据源页面，单击新增数据源，弹出支持的数据源类型。

3. 选择半结构化存储的FTP数据源里的无公网IP的数据源类型添加数据源。

添加源端的数据源



配置	说明
数据源类型	无公网IP。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
资源组	是选择部署Agent的机器，来源端Agent负责推送数据至数据集成服务端。添加资源组的详情请参见#unique_33。
Protocol	ftp或sftp。
Host	ftp默认的端口号为21，sftp的默认端口号为22。
用户名和密码	数据库对应的用户名和密码。

配置	说明
测试连通性	公网IP的数据源不支持测试连通性，直接单击完成即可。

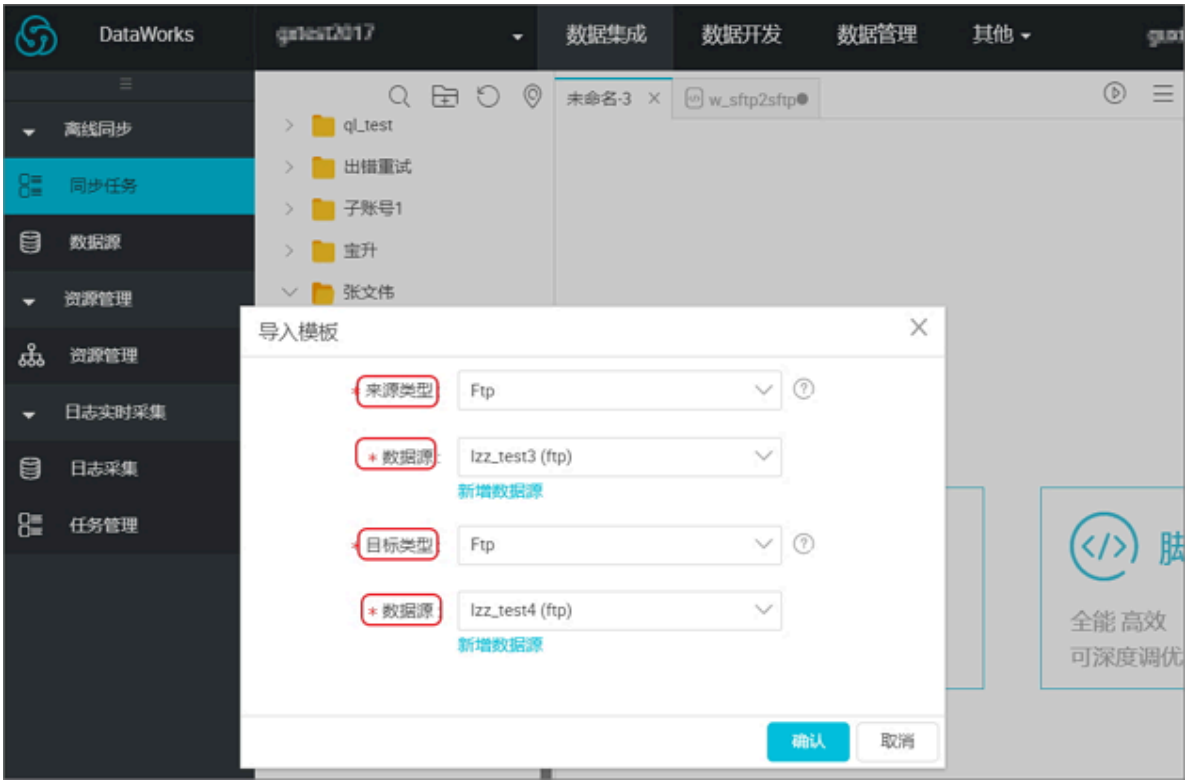
添加目标端的数据源

资源组：是选择部署Agent的机器，目的端Agent负责拉取数据至本地。

选择脚本模式

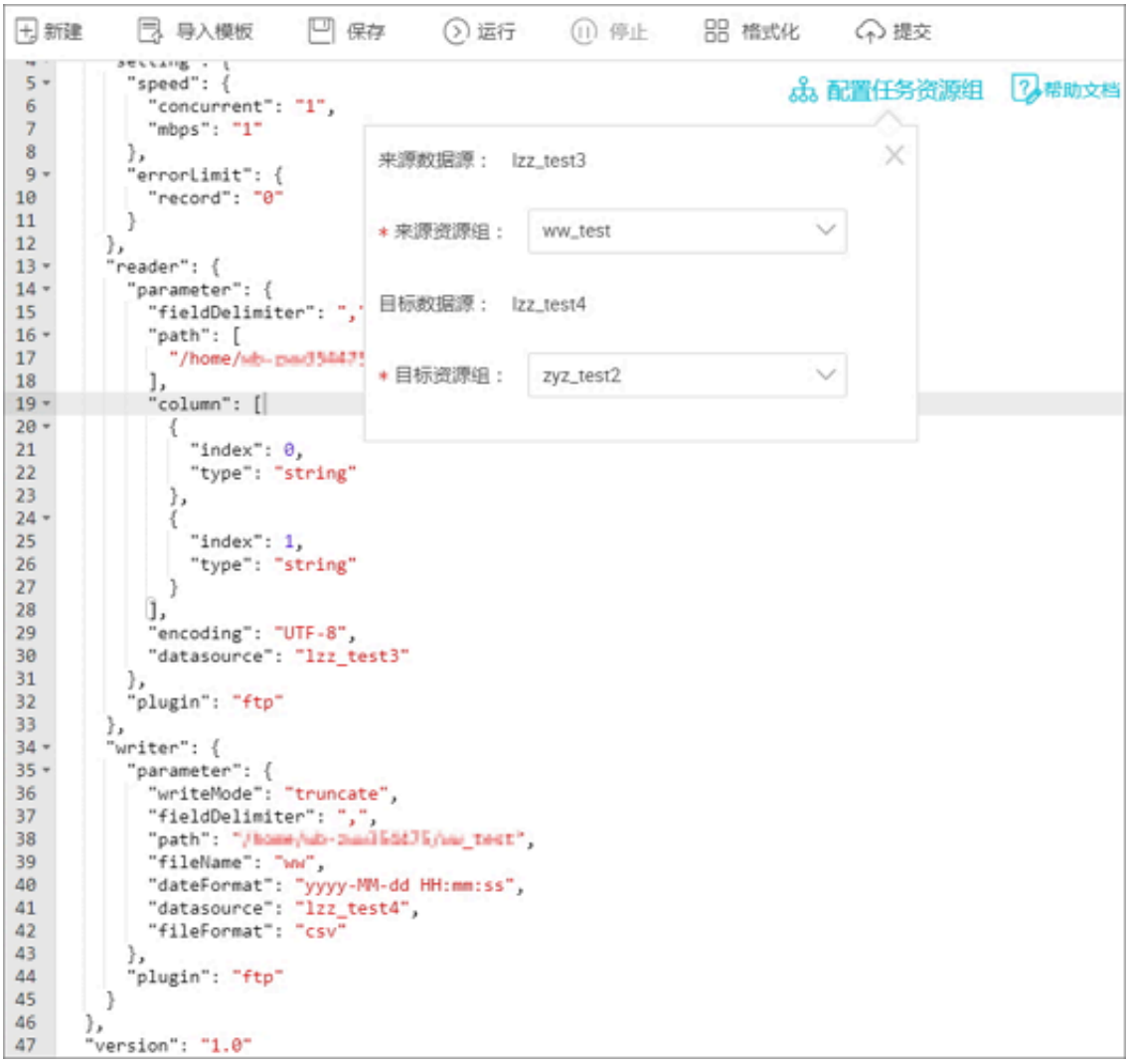
1. 单击顶部菜单栏中的数据集成，进入同步任务页面。
2. 选择新建数据集成节点 > 同步节点，输入同步任务名称。
3. 成功创建同步节点后，单击新建同步节点右上角的转换脚本，选择确认即可进入脚本模式。

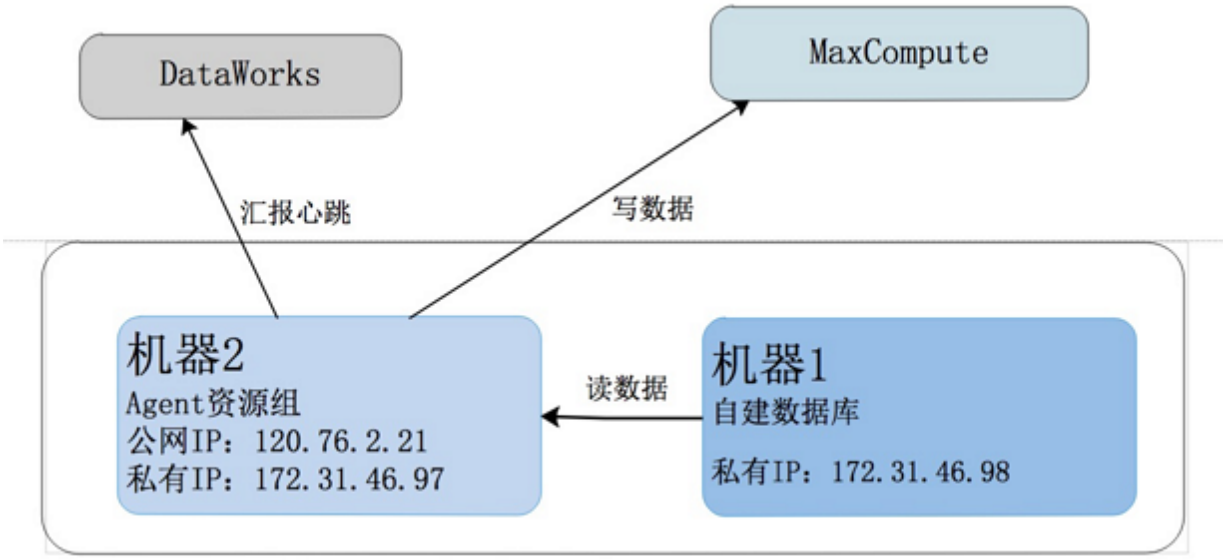
4. 选择从ftp同步到ftp的导入模板。



配置	说明
来源类型	根据您的向导模式选择的数据源直接填写数据源名称。
目标类型	可以根据下拉框选择您要写入的目标数据源。 **说明:** 数据库支持界面添加数据源，您可以在模板中选择，但是不支持的要在模板中的json代码编写相关的数据源信息，直接单击新增数据源。

5. 配置同步任务，如下图所示：





- 由于机器1没有访问公网的能力，所以需要准备一台和机器1在同一网段并且具备访问公网的能力机器2部署Agent。
- 将机器2作为任务资源组并且同步任务运行在此机器上。

操作步骤

配置数据源

1. 以开发者身份登录DataWorks，单击对应项目操作栏中的进入数据集成。
2. 进入数据源页面，单击新增数据源，弹出支持的数据源类型。

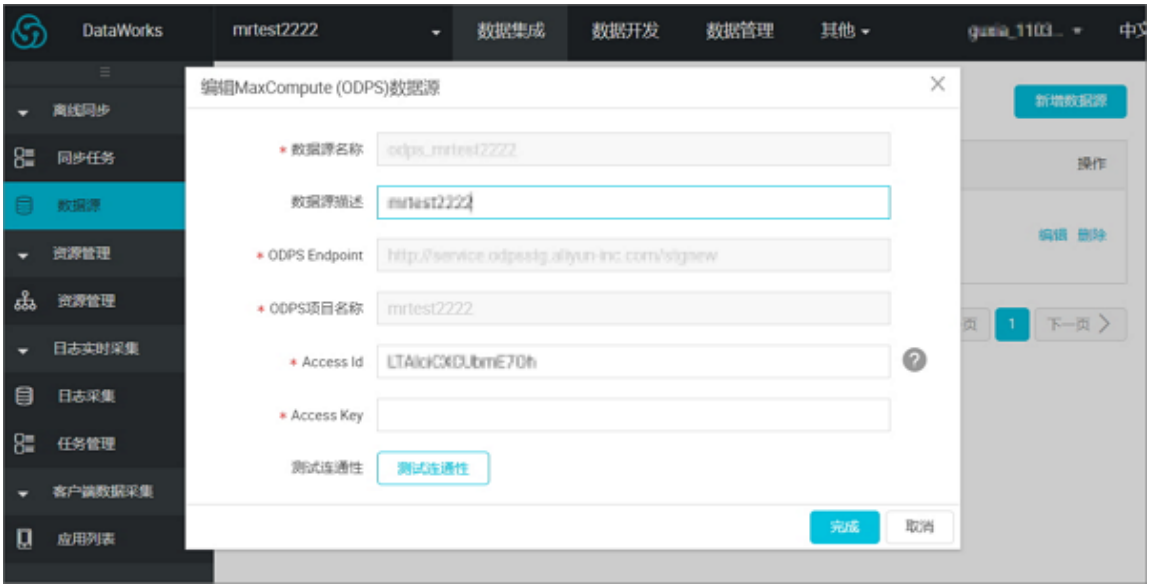
3. 选择关系数据库MySQL的数据源里的无公网IP的数据源类型添加数据源。

- 源端数据源（无公网IP）：

配置	说明
数据源类型	无公网IP。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
资源组	选择部署Agent的机器，通过Agent与外部公网联通，特殊网络环境的数据源可以将同步任务运行在资源组上。添加资源组的详情请参见#unique_33
JDBC URL	JDBC连接信息，格式是jdbc:mysql://ServerIP:Port/Database。
用户名和密码	数据库对应的用户名和密码。

配置	说明
测试连通性	公网IP的数据源不支持测试连通性，直接单击完成即可。

- 目标数据源（有公网环境）：



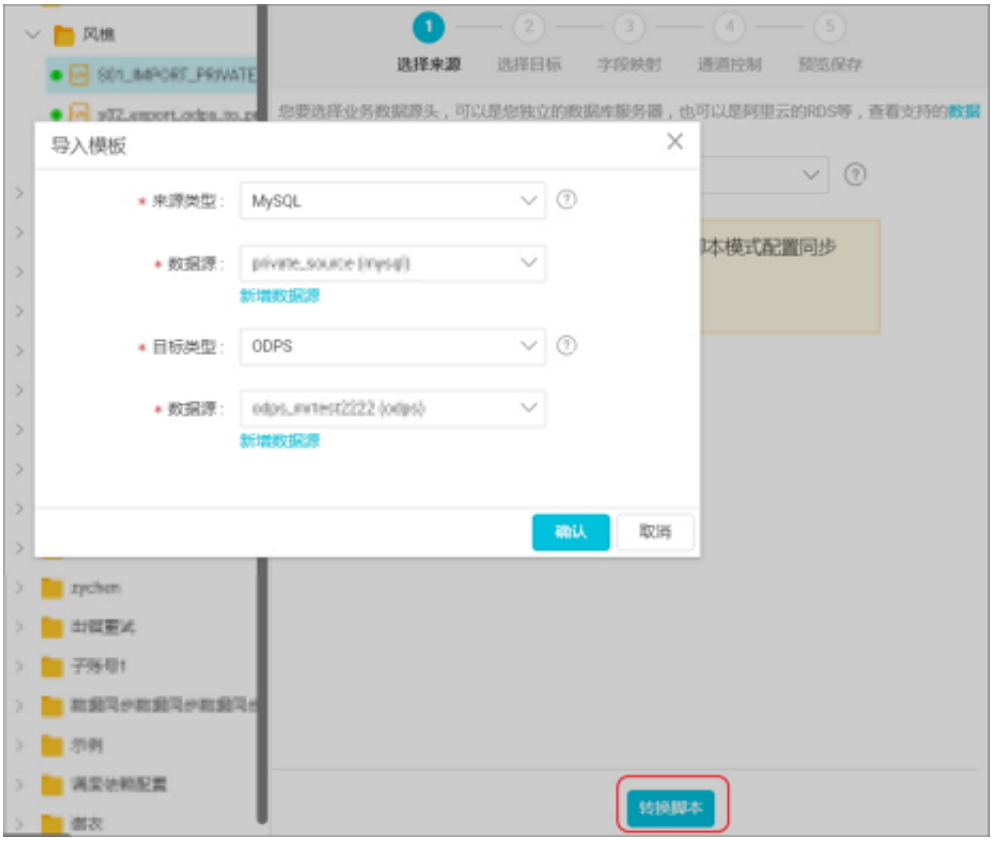
配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
ODPS Endpoint	默认只读，从系统配置中自动读取。
ODPS项目名称	对应的MaxCompute Project标识。
Access Id	与MaxCompute Project Owner云账号对应的AccessID。
Access Key	与MaxCompute Project Owner云账号对应的AccessKey，与AccessID成对使用。访问密钥相当于登录密码。
测试连通性	支持测试连通性。

配置同步任务

1. 选择来源

由于数据来源是无公网IP的数据源，所以此数据源网络无法联通，需要使用脚本模式配置同步任务，直接单击转换脚本按钮。

2. 导入模板



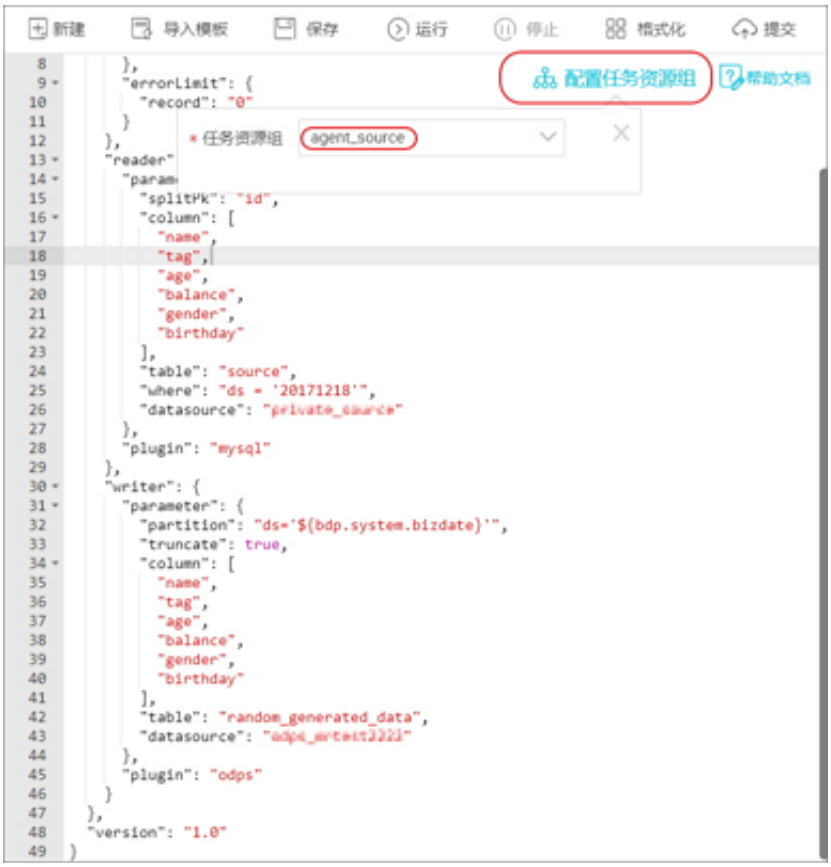
参数项说明如下：

- 来源类型：会根据您的向导模式选择的数据源直接填写数据源名称。
- 目标类型：可以根据下拉框选择你要写入的目标的数据源。

 说明：

数据库支持界面添加数据源，可以在模板中选择，但是不支持的要在模板里的json代码编写相关的数据源信息，直接单击新增数据源。

3. 转换成脚本模式样例。



配置任务资源组：可以修改和查看同步任务运行的资源组的情况，来源类型和目标类型默认会显示您添加数据源选择的资源组。

```
{
"configuration": {
 "setting": {
   "speed": {
     "concurrent": "1",//作业并发数
     "mbps": "1"//作业速率上限
   },
   "errorLimit": {
     "record": "0"//错误记录数超过值
   }
 },
 "reader": {
   "parameter": {
     "fieldDelimiter": ",",//切分键
     "encoding": "UTF-8",//编码格式
     "column": [//数据来源的列
       {
         "index": 0,
         "type": "string"
       },
       {
         "index": 1,
         "type": "string"
```

```
        }
      ],
      "path": [//文件路径
        "/home/wb-zww354475/ww.txt"
      ],
      "datasource": "lzz_test3"//数据源名称，要跟添加的数据源名保持一致
    },
    "plugin": "ftp"
  },
  "writer": {
    "parameter": {
      "writeMode": "truncate",//写入模式
      "fieldDelimiter": ",",//分隔符
      "fileName": "ww",//文件的名称
      "path": "/home/wb-zww354475/ww_test",//文件路径
      "dateFormat": "yyyy-MM-dd HH:mm:ss",
      "datasource": "lzz_test4",//数据源名称，要跟添加的数据源名保持一致
      "fileFormat": "csv"//文件的类型
    },
    "plugin": "ftp"
  }
},
"type": "job",
"version": "1.0"
}
```

运行同步任务

您可通过以下两种方式运行任务：

- 数据集成的界面直接单击运行。
- 调度运行。提交调度的步骤请参见调度配置。

2.7.3 数据增量同步

需要同步的两种数据

需要同步的数据根据数据写入后是否会发生变化，分为不会发生变化的数据（一般是日志数据）和会变化的数据（人员表，例如人员的状态会发生变化）。

示例说明

针对以上两种数据场景，需要设计不同的同步策略，本文以把业务RDS数据库的数据同步到MaxCompute为例进行说明，其他数据源的原理一样。

根据等幂性原则（一个任务多次运行的结果一样，则该任务支持重跑调度，因此该任务若出现错误，清理脏数据会比较容易），每次导入数据都是导入到一张单独的表/分区里，或者覆盖里面的历史记录。

本文定义任务测试时间是2016-11-14，全量同步是在14号进行的，同步历史数据到ds=20161113这个分区中。增量同步的场景配置了自动调度，把增量数据在15号凌晨同步到ds=20161114的分区中。数据里有一个时间字段optime，用来表示这条数据的修改时间，从而判断这条数据是否是增量数据。

不变的数据进行增量同步

由于数据生成后不会发生变化，因此可以很方便地根据数据的生成规律进行分区，较常见的是根据日期进行分区，例如每天一个分区。

数据准备

```
drop table if exists oplog;
create table if not exists oplog(
optime DATETIME,
uname varchar(50),
action varchar(50),
status varchar(10)
 );
Insert into oplog values(str_to_date('2016-11-11','%Y-%m-%d'),'LiLei
','SELECT','SUCCESS');
Insert into oplog values(str_to_date('2016-11-12','%Y-%m-%d'),'HanMM
','DESC','SUCCESS');
```

这里有两条数据作为历史数据，需先做一次全量数据同步，将历史数据同步到昨天的分区。

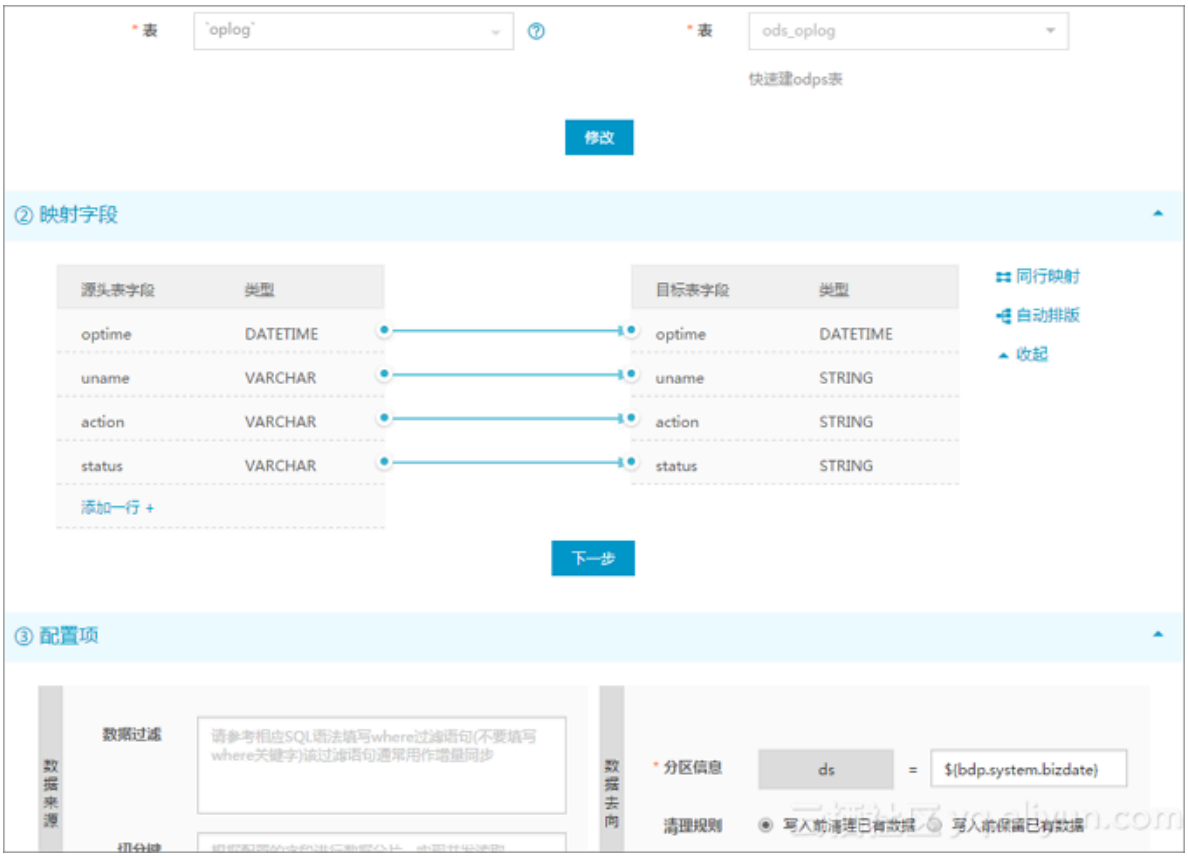
操作步骤

1. 创建MaxCompute表。

```
--创建好MaxCompute表，按天进行分区
create table if not exists ods_oplog(
 optime datetime,
 uname string,
 action string,
 status string
```

```
) partitioned by (ds string);
```

2. 配置同步历史数据的任务。



因为只需执行一次，所以只需操作一次测试即可。测试成功后，进入数据开发模块把任务的状态改成暂停（最右边的调度配置）并重新提交/发布，避免任务自动调度执行。

查看MaxCompute表的结果

```
8  select * from ods_oplog;
```

日志 结果[3]

序号	optime	uname	action	status	ds
1	2016-10-31 00:00:00	LiLei	SELECT	SUCCESS	20161113
2	2016-11-30 00:00:00	HanMM	DESC	SUCCESS	20161113

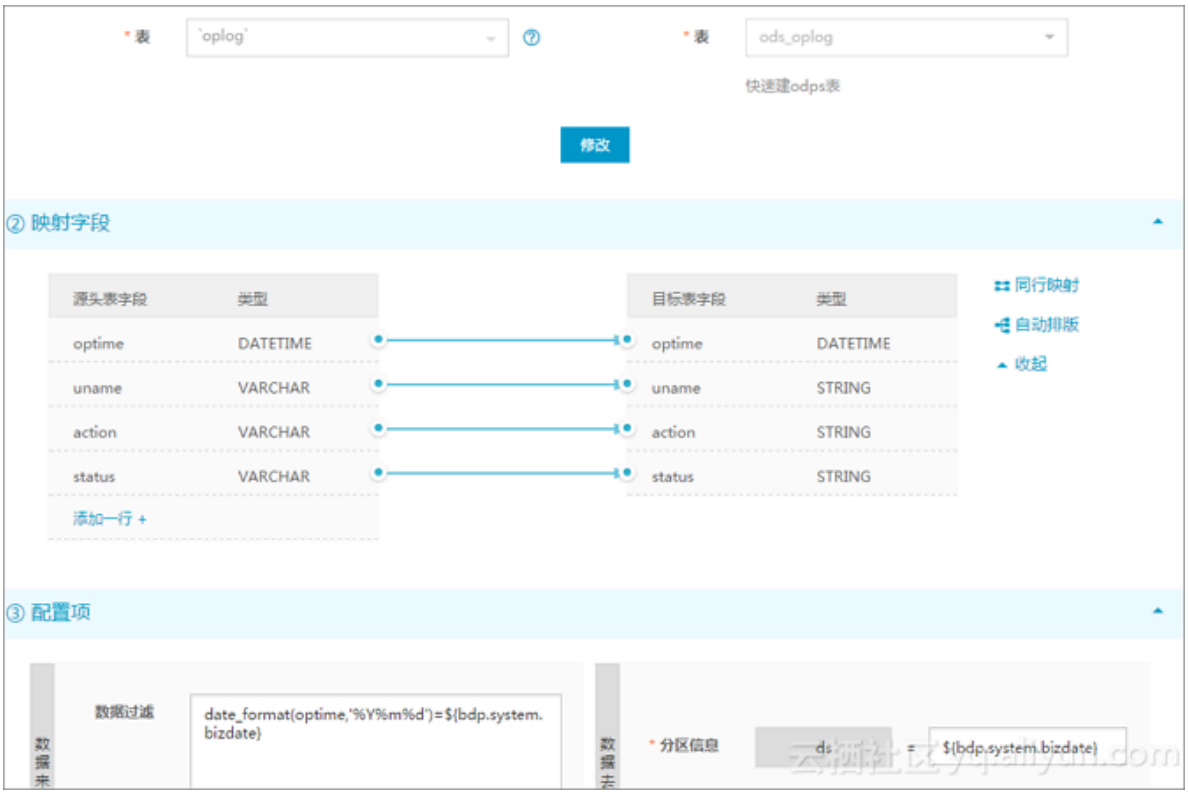
3. 往RDS源头表中多写一些数据作为增量数据。

```
insert into oplog values(CURRENT_DATE,'Jim','Update','SUCCESS');
insert into oplog values(CURRENT_DATE,'Kate','Delete','Failed');
insert into oplog values(CURRENT_DATE,'Lily','Drop','Failed');
```

4. 配置同步增量数据的任务。

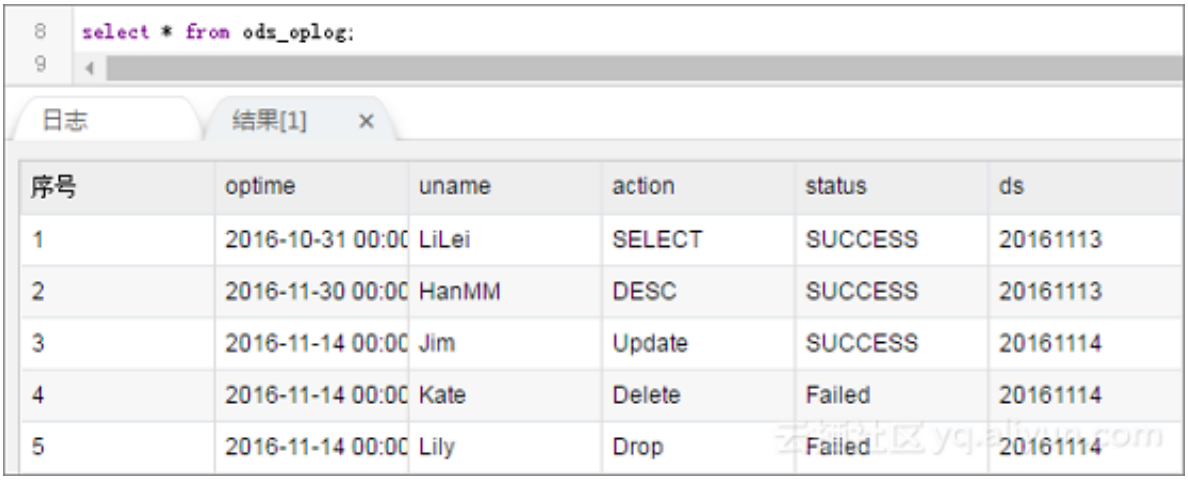
说明:

通过配置数据过滤，在15号凌晨进行同步时，可以把14号源头表全天新增的数据查询出来，并同步到目标表增量分区里。



5. 查看同步结果。

任务设置调度周期为每天调度，提交/发布后，第二天任务将自动调度执行，执行成功后，可以查看到MaxCompute目标表的数据。



序号	optime	uname	action	status	ds
1	2016-10-31 00:00	LiLei	SELECT	SUCCESS	20161113
2	2016-11-30 00:00	HanMM	DESC	SUCCESS	20161113
3	2016-11-14 00:00	Jim	Update	SUCCESS	20161114
4	2016-11-14 00:00	Kate	Delete	Failed	20161114
5	2016-11-14 00:00	Lily	Drop	Failed	20161114

会变的数据进行增量同步

根据数据仓库反映历史变化的特点，建议每天对人员表、订单表等会发生变化的数据进行全量同步，也就是说每天保存的都是全量数据，方便您获取历史数据和当前数据。

真实场景中因为某些特殊情况，需要每天只做增量同步，又因为MaxCompute不支持Update语句进行修改数据，只能用其他方法来实现。下文将为您介绍两种同步策略（全量同步、增量同步）的具体操作。

数据准备

```
drop table if exists user ;
create table if not exists user(
    uid int,
    uname varchar(50),
    deptno int,
    gender VARCHAR(1),
    optime DATETIME
    );
--历史数据
insert into user values (1,'LiLei',100,'M',str_to_date('2016-11-13','%
Y-%m-%d'));
insert into user values (2,'HanMM',null,'F',str_to_date('2016-11-13
','%Y-%m-%d'));
insert into user values (3,'Jim',102,'M',str_to_date('2016-11-12','%Y-
%m-%d'));
insert into user values (4,'Kate',103,'F',str_to_date('2016-11-12','%Y
-%m-%d'));
insert into user values (5,'Lily',104,'F',str_to_date('2016-11-11','%Y
-%m-%d'));
--增量数据
update user set deptno=101,optime=CURRENT_TIME  where uid = 2; --null
改成非null
update user set deptno=104,optime=CURRENT_TIME  where uid = 3; --非
null改成非null
update user set deptno=null,optime=CURRENT_TIME  where uid = 4; --非
null改成null
delete from user where uid = 5;
insert into user(uid,uname,deptno,gender,optime) values (6,'Lucy',105
,'F',CURRENT_TIME);
```

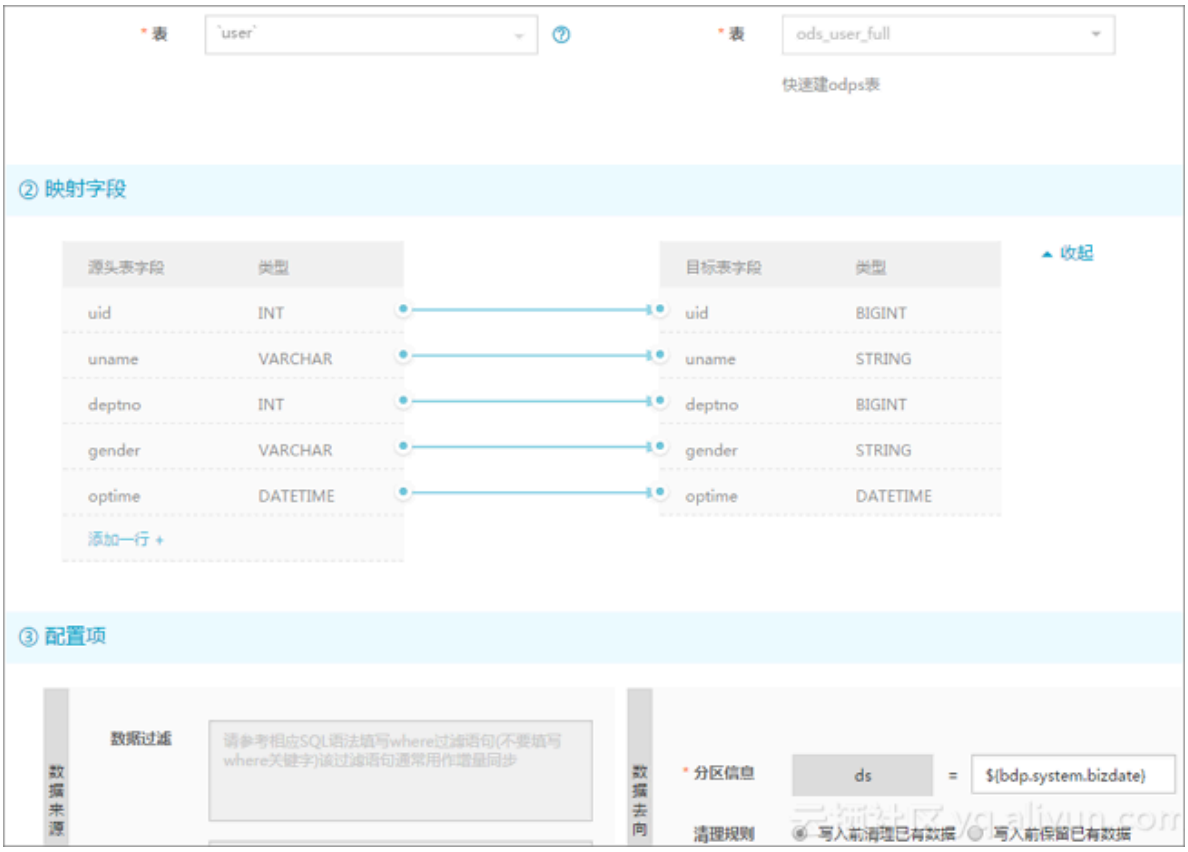
每天全量同步

1. 创建MaxCompute表。

```
--全量同步
create table ods_user_full(
    uid bigint,
    uname string,
    deptno bigint,
    gender string,
    optime DATETIME
```

```
) partitioned by (ds string);ring);
```

2. **配置全量同步任务。**



说明:

需要每天都全量同步，因此任务的调度周期需要配置为天调度。

3. 测试任务，并查看同步后MaxCompute目标表的结果。

```
11  select * from ods_user_full;
```

日志 结果[1]

序号	uid	uname	deptno	gender	optime	ds
1	1	LiLei	100	M	2016-11-13 00:00	20161113
2	2	HanMM	\N	F	2016-11-13 00:00	20161113
3	3	Jim	102	M	2016-11-12 00:00	20161113
4	4	Kate	103	F	2016-11-12 00:00	20161113
5	5	Lily	104	F	2016-11-11 00:00	20161113

因为每天都是全量同步，没有全量和增量的区别，所以第二天任务自动调度执行成功后，即可看到数据结果。

```
11  select * from ods_user_full;
```

日志 结果[1]

序号	uid	uname	deptno	gender	optime	ds
1	1	LiLei	100	M	2016-11-13 00:00	20161113
2	2	HanMM	\N	F	2016-11-13 00:00	20161113
3	3	Jim	102	M	2016-11-12 00:00	20161113
4	4	Kate	103	F	2016-11-12 00:00	20161113
5	5	Lily	104	F	2016-11-11 00:00	20161113
6	1	LiLei	100	M	2016-11-13 00:00	20161114
7	2	HanMM	101	F	2016-11-14 15:07	20161114
8	3	Jim	104	M	2016-11-14 15:07	20161114
9	4	Kate	\N	F	2016-11-14 15:07	20161114
10	6	Lucy	105	F	2016-11-14 15:07	20161114

您可执行`where ds ='20161114'`获取全量数据。

每天增量同步

不推荐使用此方式，只有在极特殊的场景下才考虑。首先这种场景不支持Delete语句，因为被删除的数据无法通过SQL语句的过滤条件查到。当然实际上公司的代码很少直接删除数据，都是使用逻辑删除，将Delete转化为Update进行处理。但是这里毕竟限制了一些特殊的业务场景不能做了，当出现特殊情况可能导致数据不一致。另外还有一个缺点就是同步后要对新增的数据和历史数据做合并。

数据准备

需要创建两张表，一张写当前的最新数据，一张写增量数据。

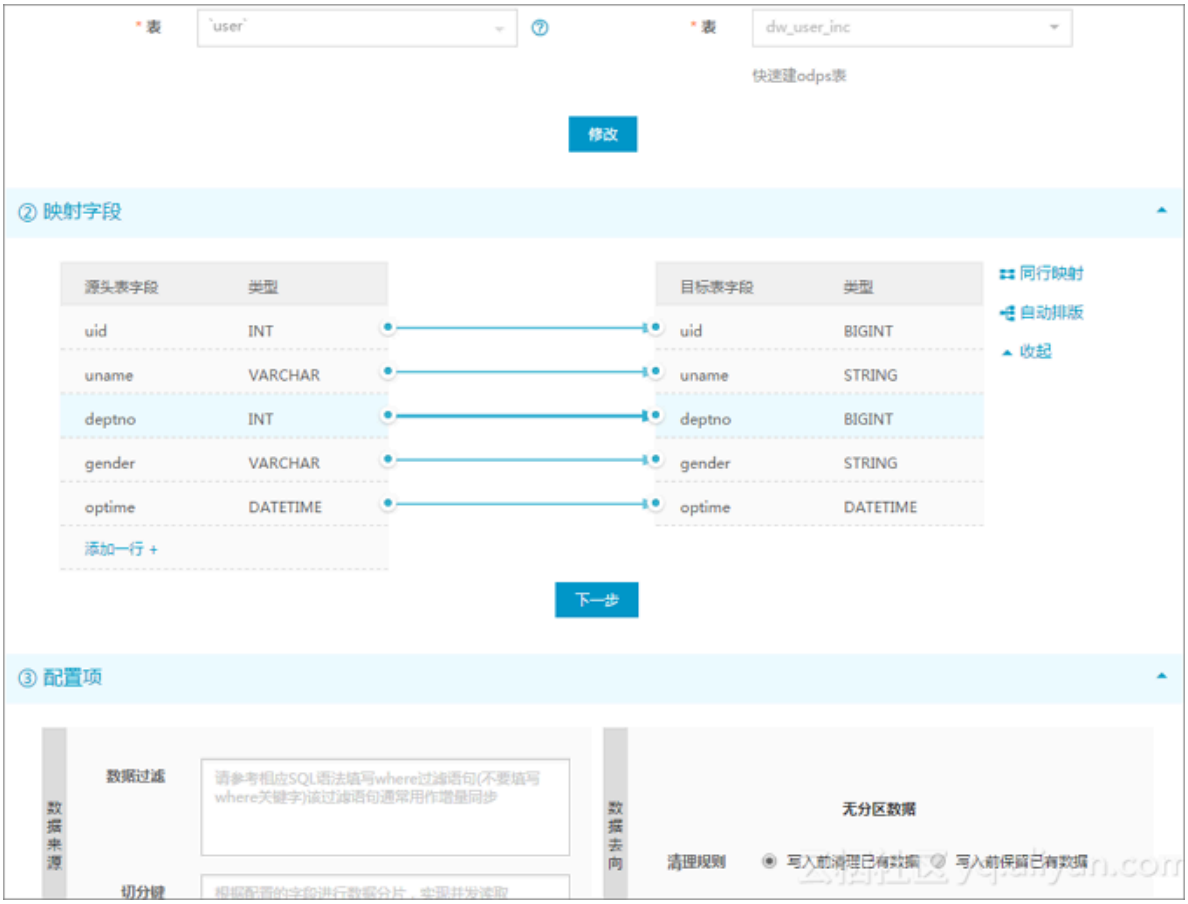
```
--结果表
create table dw_user_inc(
    uid bigint,
```

```
    uname string,
    deptno bigint,
    gender string,
    optime DATETIME
);
--增量记录表
create table ods_user_inc(
    uid bigint,
    uname string,
    deptno bigint,
    gender string,
    optime DATETIME
)
```

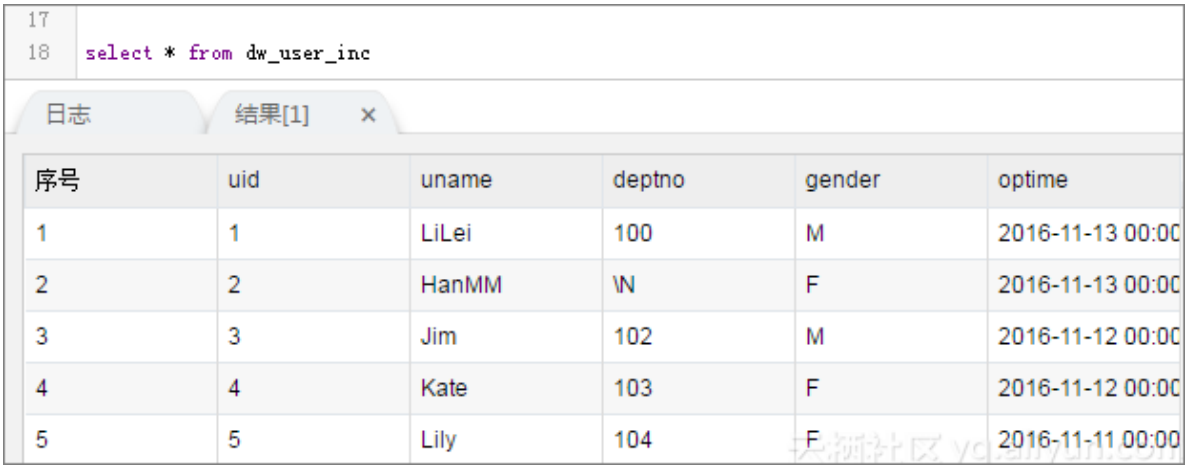
1. **配置任务将全量数据直接写入结果表。**

说明:

只需执行一次，执行成功后需进入数据开发页面将任务设置暂停。



结果如下：

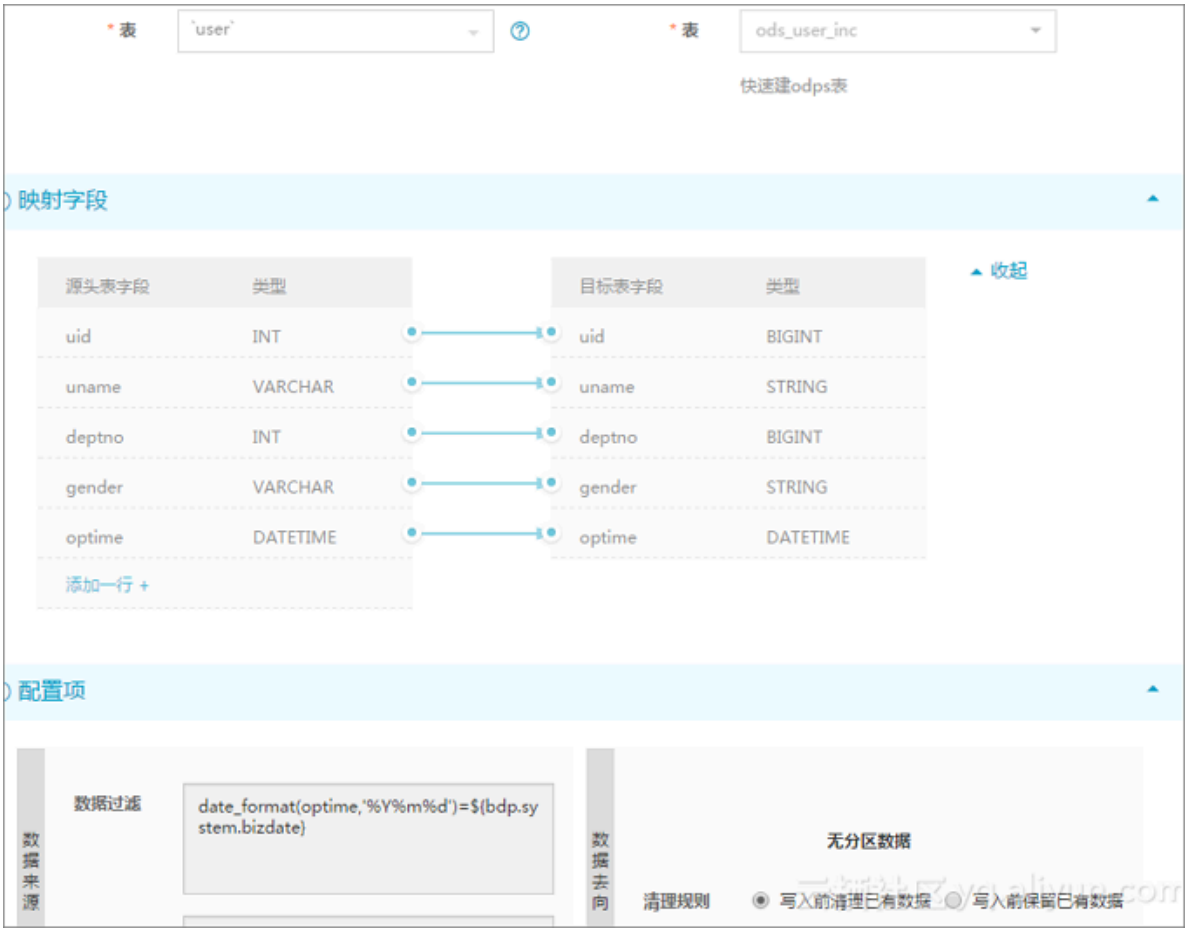


```
17
18  select * from dw_user_inc
```

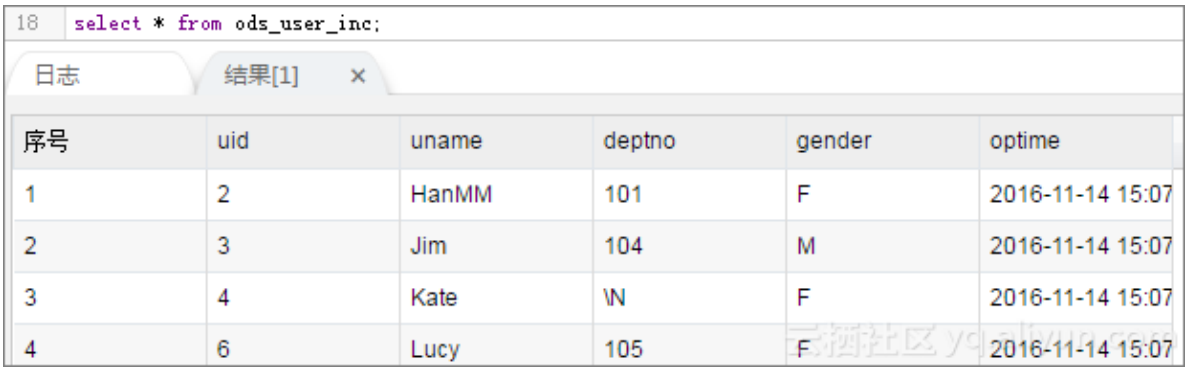
日志 结果[1]

序号	uid	uname	deptno	gender	optime
1	1	LiLei	100	M	2016-11-13 00:00
2	2	HanMM	\N	F	2016-11-13 00:00
3	3	Jim	102	M	2016-11-12 00:00
4	4	Kate	103	F	2016-11-12 00:00
5	5	Lily	104	F	2016-11-11 00:00

2. 配置任务将增量数据写入到增量表。



结果如下：



序号	uid	uname	deptno	gender	optime
1	2	HanMM	101	F	2016-11-14 15:07
2	3	Jim	104	M	2016-11-14 15:07
3	4	Kate	\N	F	2016-11-14 15:07
4	6	Lucy	105	F	2016-11-14 15:07

3. 合并数据。

```
insert overwrite table dw_user_inc
select
--所有select操作，如果ODS表有数据，说明发生了变动，以ODS表为准
case when b.uid is not null then b.uid else a.uid end as uid,
case when b.uid is not null then b.uname else a.uname end as uname,
case when b.uid is not null then b.deptno else a.deptno end as
deptno,
case when b.uid is not null then b.gender else a.gender end as
gender,
case when b.uid is not null then b.optime else a.optime end as
optime
```

```
from
dw_user_inc a
full outer join ods_user_inc b
on a.uid  = b.uid ;
```

结果如下：

```
18  select * from dw_user_inc;
```

日志 结果[1]

序号	uid	uname	deptno	gender	optime
1	3	Jim	104	M	2016-11-14 15:07
2	6	Lucy	105	F	2016-11-14 15:07
3	1	LiLei	100	M	2016-11-13 00:00
4	4	Kate	\N	F	2016-11-14 15:07
5	2	HanMM	101	F	2016-11-14 15:07
6	5	Lily	104	F	2016-11-11 00:00

由结果可见，Delete的那条记录没有同步成功。

对比上述两种同步方式，可以很清楚看到两种同步方法的区别和优劣。每天增量同步的优点是同步的增量数据量比较小，但是带来的缺点有可能有数据不一致的风险，而且还需要用额外的计算进行数据合并。

如果不是必要情况，会变化的数据进行每天全量同步即可。如果对历史数据希望只保留一定的时间，超出时间的做自动删除，可以设置Lifecycle。

2.7.4 数据同步任务调优

数据同步任务调度运行时，您可能会遇到实例的执行时间超过预期的情况。本文为您介绍如何在数据同步任务实例执行慢、时间差异大等不满足预期的情况下进行调优的方法。

场景分类

通常数据同步任务执行慢的场景分为以下三种：

- 任务开始运行的时间和调度时间差异比较大。
- 任务长时间处于WAIT状态。
- 任务同步的速率慢。

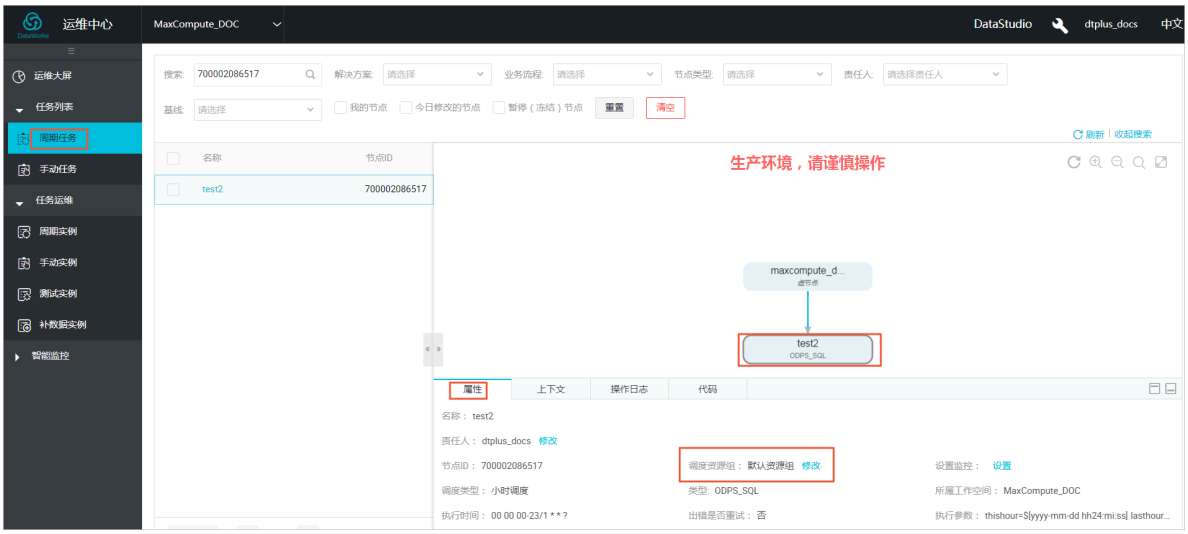
前提条件

正式开始数据同步任务调优前，请首先收集下列信息：

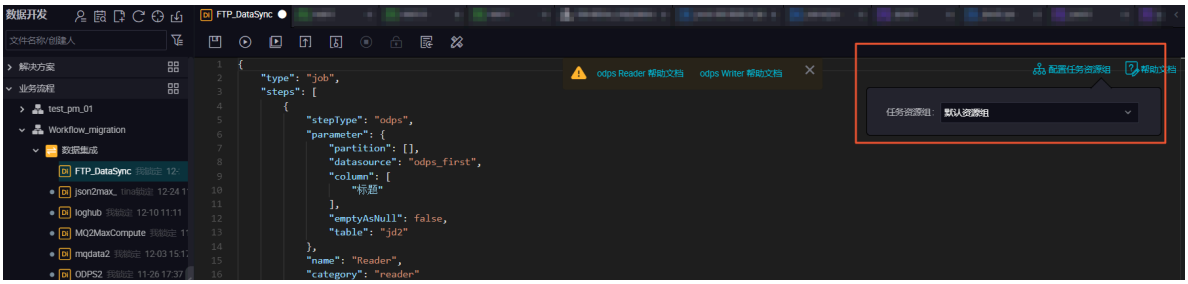
- 任务运行日志（从日志开始打印到结束）
- 任务的属性标签页信息

针对数据同步任务，DataWorks的调度资源分为一级调度资源和二级运行资源。

- 一级调度资源： 可以在运维中心 > 周期任务 > 属性的调度资源组中查看或配置。



- 二级运行资源：可以在数据开发 > 任务资源组中配置 。任务资源组下的可选自定义资源组 需要在数据集成的资源组中完成配置。

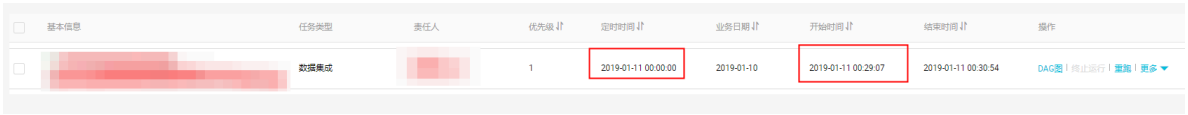


场景一：任务开始运行时间和调度时间差异比较大

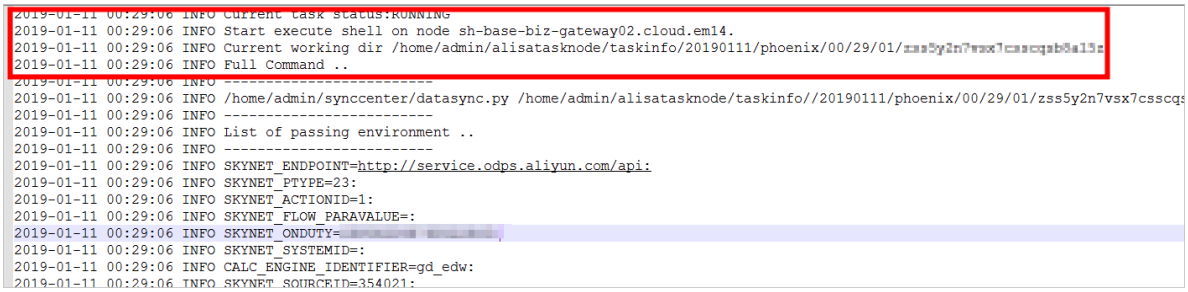
在该场景下，您首先需要任务运行日志和任务属性标签页信息 。对比分析发现， 任务运行日志中开始running的时间和属性节点的调度时间是有差异的，时间主要耗费在等待调度上。

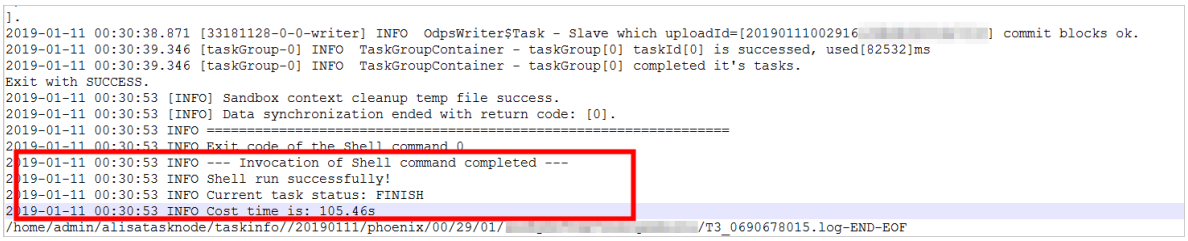
问题示例

1. 在运维中心中的周期任务页面查看用户任务的属性标签页查，发现调度时间在00:00， 但是开始运行时间在00:29，怀疑时间主要消耗在等待调度上。



2. 在实例页面右键查看用户任务运行日志，任务从00:29分开始运行，00:30执行结束，整个任务执行仅仅花费了1分钟。说明本次任务本身执行无问题。





问题解法

1. 首先建议您观察您的项目下是否有较多的任务同时调度。默认资源组下的一级调度资源有限，同时调度的任务较多会有其他任务排队等待。
2. 通常每天0点-2点是 业务调度的高峰期， 建议您的业务运行时间尽量避开高峰期 。

场景二：任务同步速率慢

在该场景下，通过任务运行日志分析，通常有两种情况：

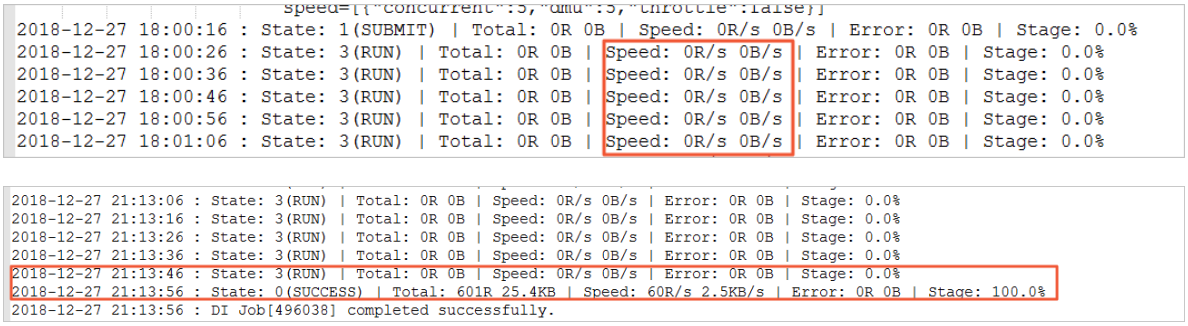
1. 任务一直在运行，但速率是0。
2. 任务速率较低。

任务速率为0

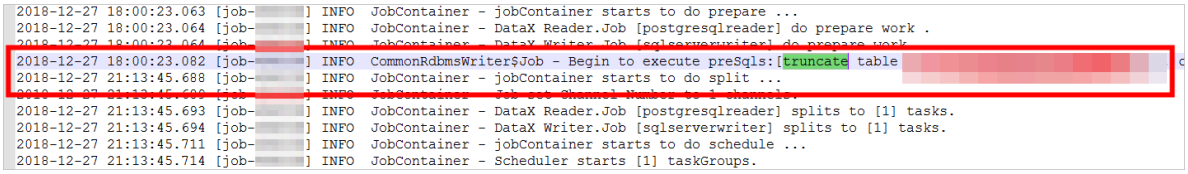
查看运行日志，看到任务长时间处于run的状态，速率为0。通常是由于拉取的SQL执行比较慢（源数据库CPU负载高或网络流量占用高），或在拉取SQL前进行truncate等操作，导致处理时间较长。

问题示例

1. 查看任务运行日志，任务长时间在run，速率为0。 从18:00开始到21:13结束。



2. 查看运行日志信息有truncate操作记录，从18:00开始到21:13 truncate操作结束。



问题解法

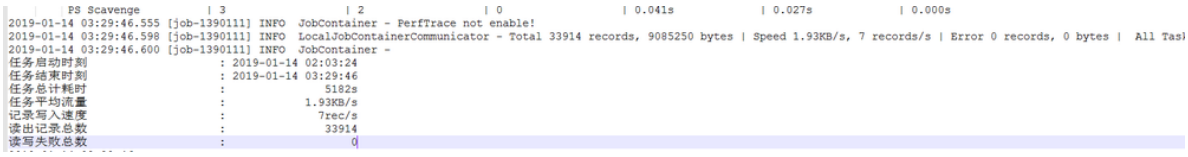
综上，可以推断是truncate操作导致的同步任务慢，您可能需要检查源数据库truncate慢的原因。

任务速率慢

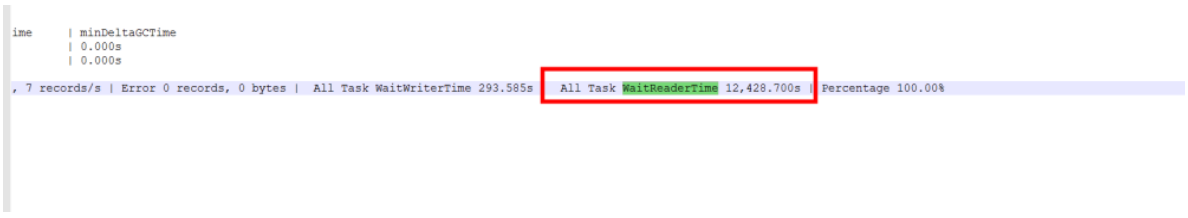
查看运行日志，看到任务同步速率不为0，但是速率慢。

问题示例

1. 获取运行日志后，查看日志中信息同步速率确实比较慢， 约为1.93kb/s。



2. 查看运行日志中同步时间消耗字段 WaitWriterTime、WaitReaderTime的信息， 发现WaitReaderTime时长较长，主要在等待读数据。



问题解法

针对速率比较慢的情况， 可以看下主要在等Writer还是Reader，如果是读或写慢，需要查看对应的源数据库或目的数据库的负载情况。

2.7.5 通过数据集成导入数据到Elasticsearch

本文将为您介绍如何通过数据集成对离线Elasticsearch进行数据导入的操作。

数据集成是阿里巴巴集团提供的数据同步平台。该平台具备可跨异构数据存储系统、可靠、安全、低成本、可弹性扩展等特点，可为20多种数据源提供不同网络环境下的离线（全量/增量）数据进出通道。数据源类型的详情请参见#unique_25。

前提条件

通过数据集成导入数据前，您需要满足以下条件。

- #unique_30，并建好账号的访问秘钥，即AccessKeys。
- 开通MaxCompute，自动产生一个默认的MaxCompute数据源。
- 已使用主账号创建项目。

  您可以在项目中协作完成工作流，共同维护数据和任务等，因此使用DataWorks之前需要先创建一个项目。

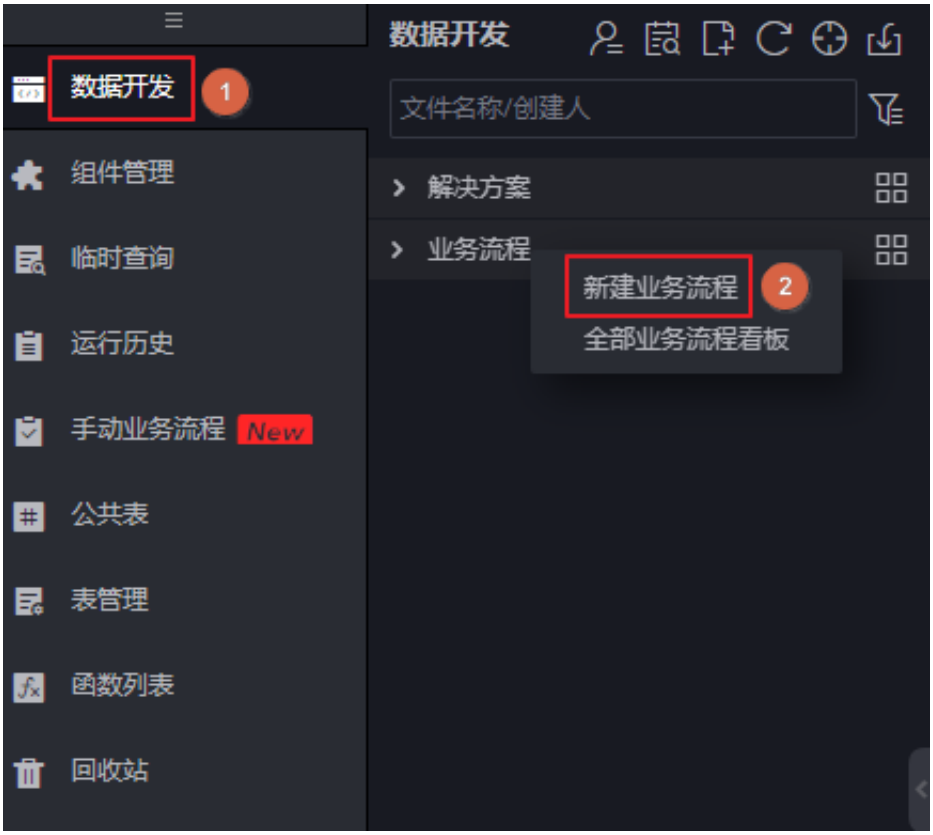
  > 说明:
  > 如果您想通过子账号创建数据集成任务，可以赋予其相应的权限。详情请参见#unique_32和#unique_250。

- 配置好相关的数据源，详情请参见数据源配置。

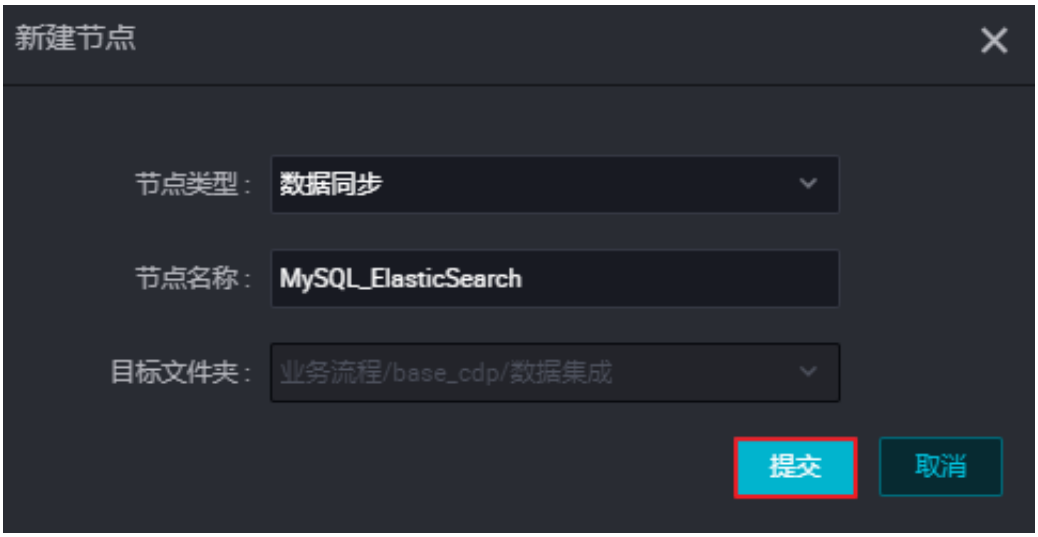
操作步骤

1. 以开发者身份登录DataWorks控制台，单击对应项目下的进入数据开发。

2. 右键单击业务流程，选择新建业务流程。

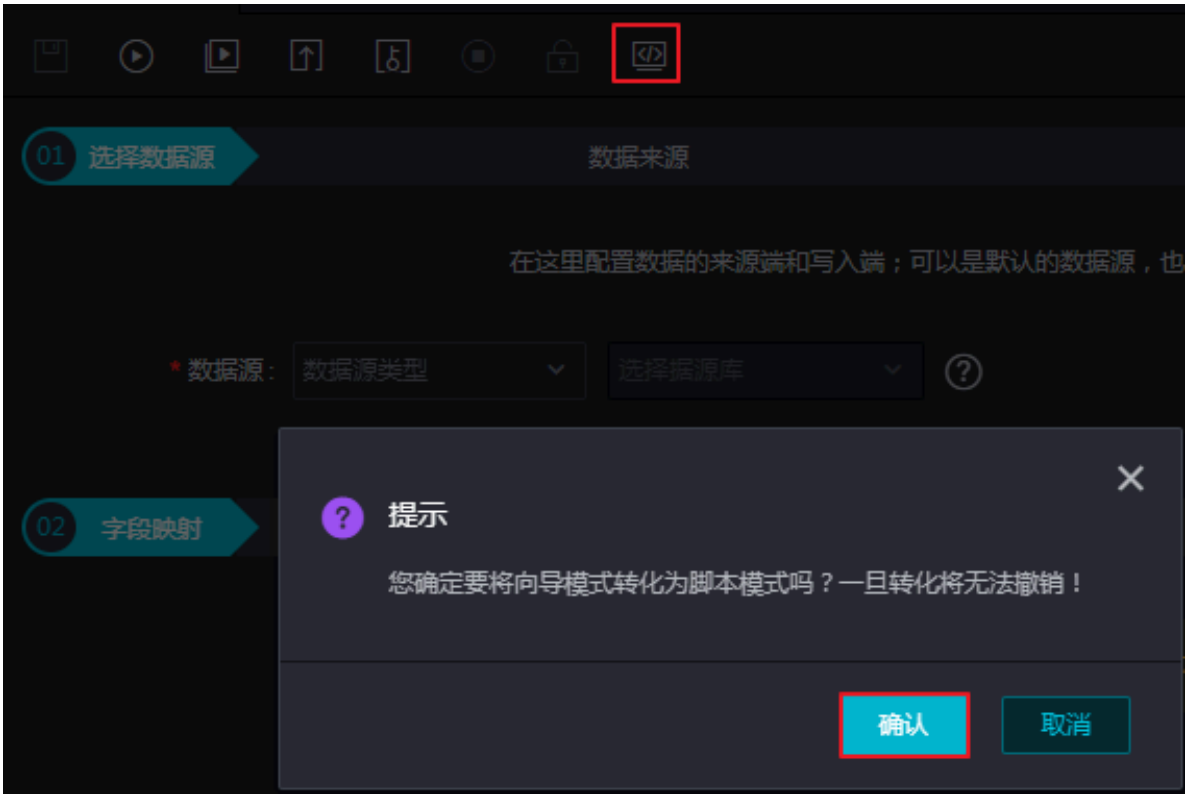


3. 右键单击业务流程下的数据集成，选择新建数据集成节点 > 同步节点。
4. 填写新建节点对话框中的配置，单击提交。



配置	说明
节点类型	默认数据同步类型。
节点名称	填写该节点名称。
目标文件夹	默认放在相应的业务流程下。

5. 单击新建同步节点右上角的转换脚本，选择确认即可进入脚本模式。



6. 单击工具栏中的导入模板，填写导入模板对话框中的配置。

配置	说明
来源类型	此处选择MySQL类型。
数据源	选择配置好的数据源。
目标类型	此处选择Elasticsearch类型。

7. 单击确认生成初始脚本，可根据自身情况进行配置。

```
{
"configuration": {
"setting": {
  "speed": {
    "concurrent": "1", //作业并发数
    "mbps": "1" //作业速率上限
  }
},
"reader": {
  "parameter": {
    "connection": [
      {
        "table": [
          "`es_table`" //源端表名
        ],
        "datasource": "px_mysql_OK" //数据源名，建议跟添加的数据源名保持一
致
      }
    ],
```

```
    "column": [ //源端表的列名
      "col_ip",
      "col_double",
      "col_long",
      "col_integer",
      "col_keyword",
      "col_text",
      "col_geo_point",
      "col_date"
    ],
    "where": "", //过滤条件
  },
  "plugin": "mysql"
},
"writer": {
  "parameter": {
    "cleanup": true, //是否在每次导入数据到Elasticsearch的时候清空原有数据，全量导入/重建索引的时候需要设置为true，同步增量的时候必须为false，这里因为是同步，则需要设置为false。
    "accessKey": "nimda", //如果使用了X-PACK插件，则这里需要填写password，如果没使用，则这里填空字符串即可。阿里云Elasticsearch使用了X-PACK插件，这里需要填写password。
    "index": "datax_test", // Elasticsearch的索引名称，如果之前没有，插件会自动创建。
    "alias": "test-1-alias", //数据导入完成后写入别名
    "settings": {
      "index": {
        "number_of_replicas": 0,
        "number_of_shards": 1
      }
    },
    "batchSize": 1000, //每次批量数据的条数
    "accessId": "default", //如果使用了X-PACK插件，则这里需要填写username，如果没使用，则这里填空字符串即可。阿里云Elasticsearch使用了X-PACK插件，这里需要填写username。
    "endpoint": "http://xxx.xxxx.xxx:xxxx", //Elasticsearch的连接地址，控制台上有
    "splitter": ",", //如果插入数据是array，就使用指定分隔符。
    "indexType": "default", //Elasticsearch中相应索引下的类型名称。
    "aliasMode": "append", //数据导入完成后增加别名的模式，append(增加模式)，exclusive（只留这一个）。
    "column": [ //Elasticsearch中的列名，顺序和Reader中的Column顺序一致。
      {
        "name": "col_ip",//对应于TableStore中的属性列：name
        "type": "ip"//文本类型，采用默认分词
      },
      {
        "name": "col_double",
        "type": "string"
      },
      {
        "name": "col_long",
        "type": "long"
      },
      {
        "name": "col_integer",
        "type": "integer"
      },
      {
        "name": "col_keyword",
        "type": "keyword"
      },
      {
        "name": "col_text",
```

```
          "type": "text"
        },
        {
          "name": "col_geo_point",
          "type": "geo_point"
        },
        {
          "name": "col_date",
          "type": "date"
        }
      ],
      "discovery": false//是否自动发现，设置为true
    },
    "plugin": "elasticsearch"//Writer插件的名称：ElasticsearchWriter，不
需要修改
}
},
"type": "job",
"version": "1.0"
}
```

8. 单击保存并运行。

> 说明：
>
> - Elasticsearch仅支持以脚本模式导入数据。
> - 如果想选择新模板，可单击工具栏中的导入模板，一旦导入新模板，原有内容将会被全部覆盖。
> - 同步任务保存后，直接单击运行，任务会立刻运行。您也可单击提交，将同步任务提交到调度系统中，调度系统会按照配置属性在从第二天开始自动定时执行。

参考文档

其他的配置同步任务详细信息请参见下述文档。

- 配置Reader插件。
- 配置Writer插件。

2.7.6 日志服务（Loghub）通过数据集成投递数据

本文将以LogHub数据同步至MaxCompute为例，为您介绍如何通过数据集成功能同步LogHub数据至数据集成已支持的目的端数据源（如MaxCompute、OSS、OTS、RDBMS和DataHub等）。

> 说明：
>
> 此功能已在华北2、华东2、华南1、中国（香港）、美西1、亚太东南1、欧洲中部1、亚太东南2、亚太东南3、亚太东北1、亚太南部1等多个地域发布。

支持场景

- 支持跨地域的LogHub与MaxCompute等数据源的数据同步。
- 支持不同阿里云账号下的LogHub与MaxCompute等数据源间的数据同步。
- 支持同一阿里云账号下的LogHub与MaxCompute等数据源间的数据同步。
- 支持公共云与金融云账号下的LogHub与MaxCompute等数据源间的数据同步。

跨阿里云账号的特别说明

以B账号进入数据集成配置同步任务，将A账号的LogHub数据同步至B账号的MaxCompute为例。

1. 用A账号的AccessId和Accesskey创建LogHub数据源。

   此时B账号可以拖A账号下所有SLS Project的数据。

2. 用A账号下子账号A1的AccessId和Accesskey创建LogHub数据源。

   - A给A1赋权日志服务的通用权限，即`AliyunLogFullAccess`和`AliyunLogReadOnlyAccess`，详情请参见授权RAM子用户访问日志服务资源。
   - A给A1赋权日志服务的自定义权限。

     主账号A进入RAM控制台 > 策略管理页面，选择自定义授权策略 > 新建授权 > 空白模板。

     相关的授权请参见访问控制RAM和RAM子用户访问。

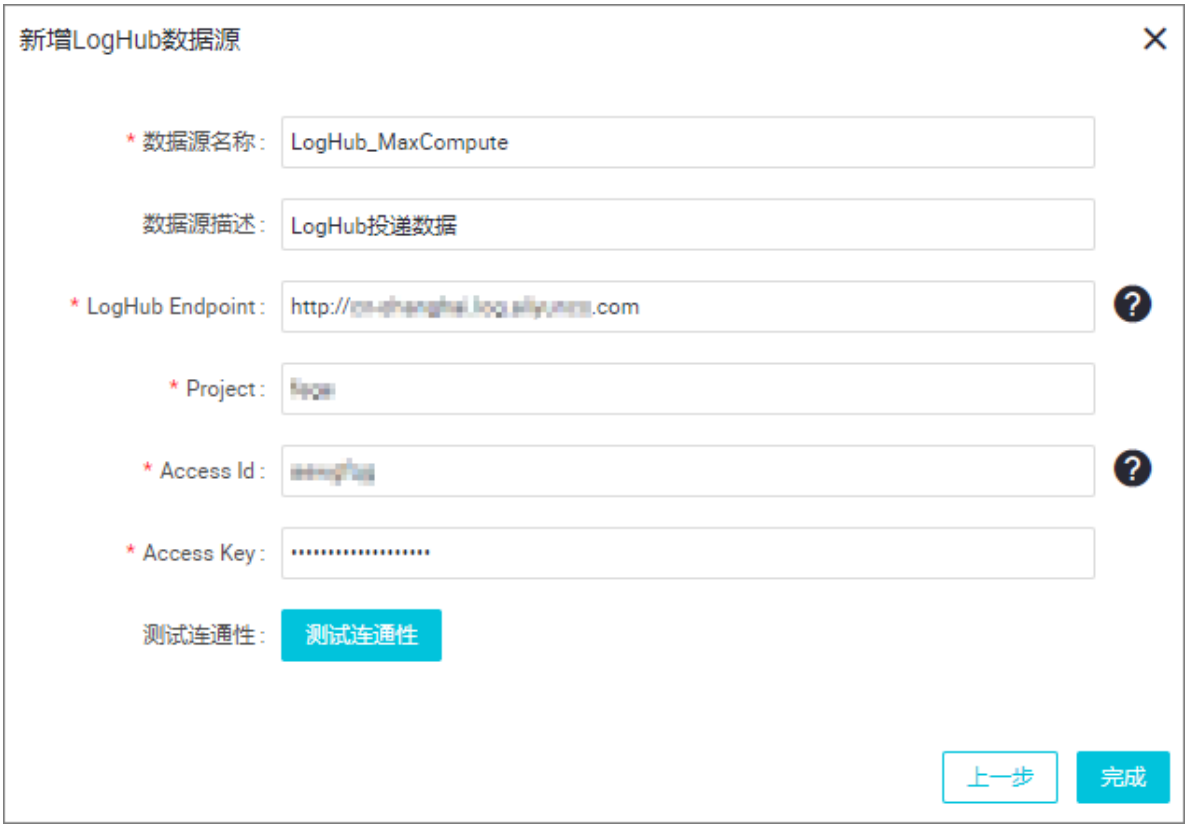
     根据下述策略进行授权后，B账号通过子账号A1只能同步日志服务project_name1以及project_name2的数据。

```
{
"Version": "1",
"Statement": [
{
"Action": [
"log:Get*",
"log:List*",
"log:CreateConsumerGroup",
"log:UpdateConsumerGroup",
"log:DeleteConsumerGroup",
"log:ListConsumerGroup",
"log:ConsumerGroupUpdateCheckPoint",
"log:ConsumerGroupHeartBeat",
"log:GetConsumerGroupCheckPoint"
],
"Resource": [
"acs:log:*:*:project/project_name1",
"acs:log:*:*:project/project_name1/*",
"acs:log:*:*:project/project_name2",
"acs:log:*:*:project/project_name2/*"
],
"Effect": "Allow"
}
]
```

```
}
```

新增数据源

1. B账号或B的子账号以开发者身份登录DataWorks控制台，单击对应项目下的进入数据集成。
2. 进入同步资源管理 > 数据源页面，单击右上角的新增数据源。
3. 选择数据源类型为LogHub，填写新增LogHub数据源对话框中的配置。



配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。
LogHub Endpoint	LogHub的Endpoint，格式为`http://yyy.com`。
Project	详情请参见服务入口。
Access Id/Access Key	即访问密钥，相当于登录密码。您可以填写主账号或子账号的Access Id和Access Key。

4. 单击测试连通性。
5. 测试连通性通过后，单击确定。

通过向导模式配置同步任务

1. 进入数据开发 > 业务流程页面，单击左上角的新建数据同步节点。

2. 填写新建数据同步节点对话框中的配置，单击提交，进入数据同步任务配置页面。
3. 选择数据来源。

配置	说明
数据源	填写LogHub数据源的名称。
Logstore	导出增量数据的表的名称。该表需要开启Stream，可以在建表时开启，或者使用UpdateTable接口开启。
日志开始时间	数据消费的开始时间位点，为时间范围（左闭右开）的左边界，为yyyyMMddHHmmss格式的时间字符串（比如20180111013000），可以和DataWorks的调度时间参数配合使用。
日志结束时间	数据消费的结束时间位点，为时间范围（左闭右开）的右边界，为yyyyMMddHHmmss格式的时间字符串（比如20180111013010），可以和DataWorks的调度时间参数配合使用。
批量条数	一次读取的数据条数，默认为256。

数据预览默认收起，您可以单击进行预览。

说明:

数据预览是选择LogHub中的几条数据展现在预览框，可能您同步的数据会跟您的预览的结果不一样，因为您同步的数据会指定开始时间和结束时间。

4. 选择数据去向。

选择MaxCompute数据源及目标表。

配置	说明
数据源	填写配置的数据源名称。
表	选择需要同步的表。
分区信息	此处需同步的表是非分区表，所以无分区信息。
清理规则	· 写入前清理已有数据：导数据之前，清空表或者分区的所有数据，相当于insert overwrite。 · 写入前保留已有数据：导数据之前不清理任何数据，每次运行数据都是追加进去的，相当于insert into。
压缩	默认选择不压缩。
空字符串作为null	默认选择否。

5. 字段映射。

选择字段的映射关系。需对字段映射关系进行配置，左侧源头表字段和右侧目标表字段为一一对应的关系。

6. 通道控制。

配置作业速率上限和脏数据检查规则。

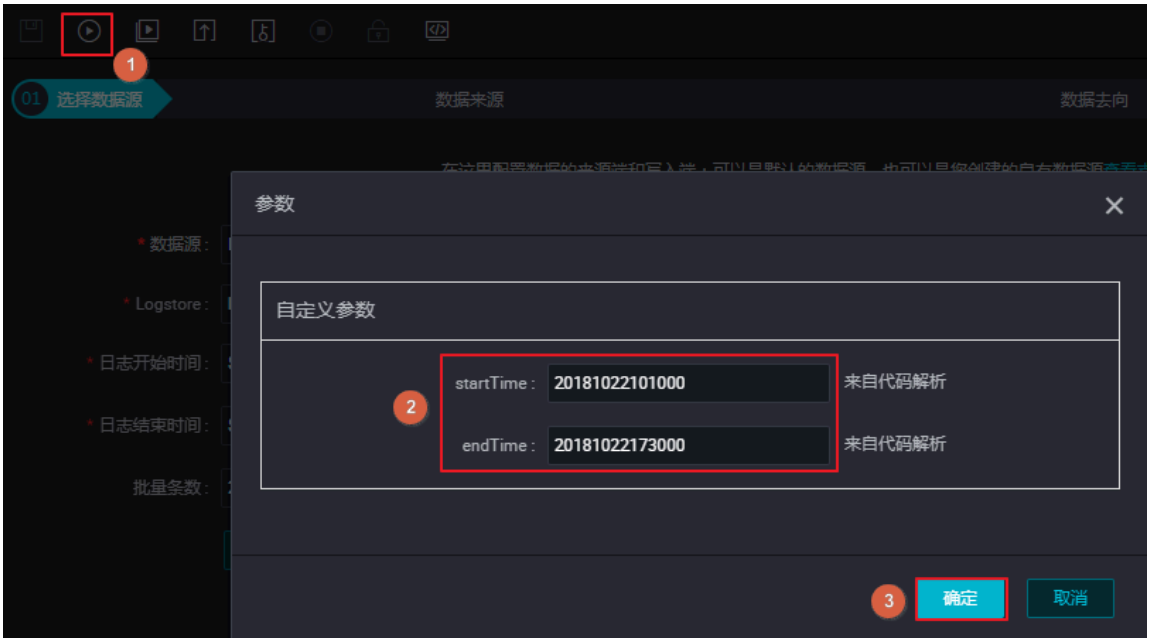
配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

7. 运行任务。

您可以通过以下两种方式运行任务。

- 直接运行（一次性运行）

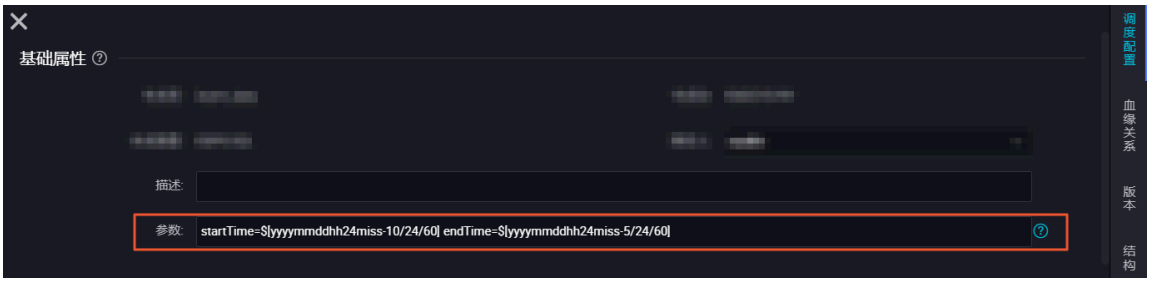
单击任务上方的运行按钮，将直接在数据集成页面运行任务，运行之前需要配置自定义参数的具体数值。



如上图所示，代表同步10:10到17:30这段时间的LogHub记录到MaxCompute。

- 调度运行

单击提交按钮，将同步任务提交到调度系统中，调度系统会按照配置属性在从第二天开始自动定时执行。

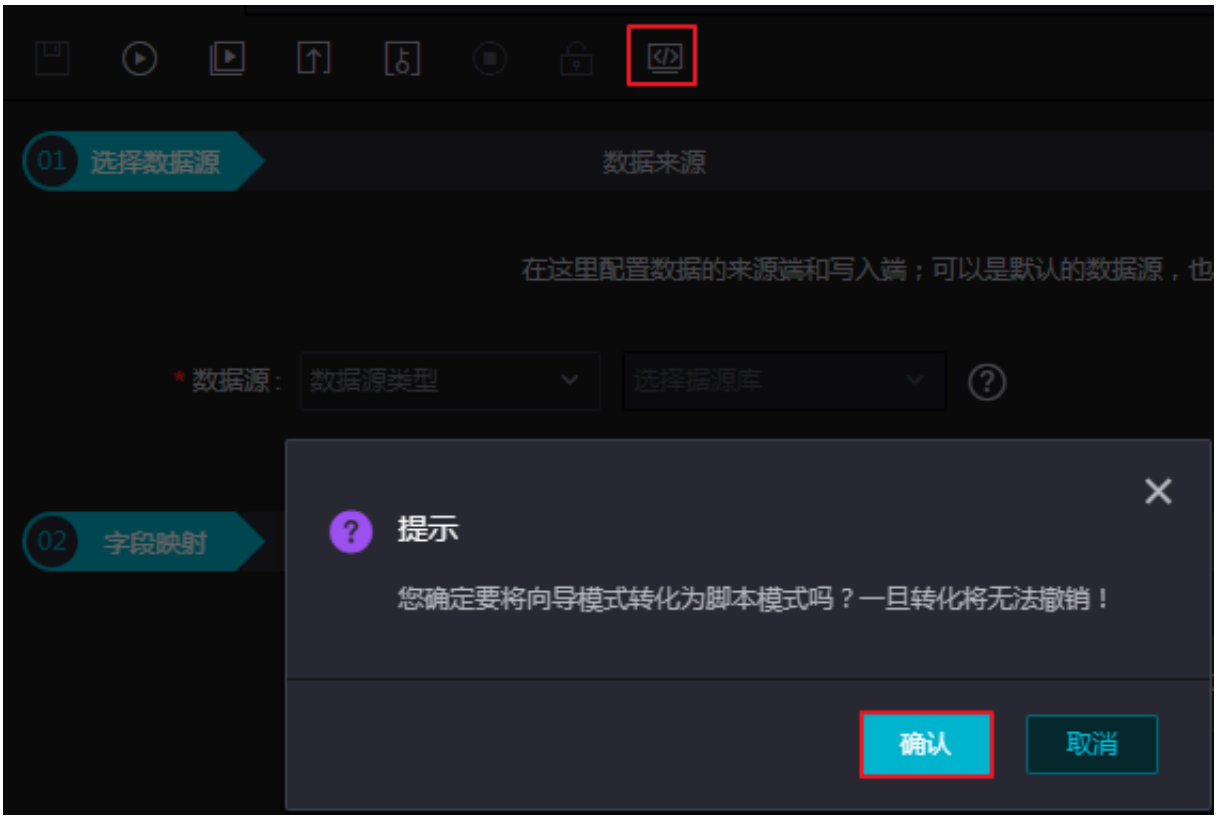


如上图所示，设置开始时间和结束时间：startTime=$[yyyymmddhh24miss-10/24/60]系统前10分钟到 endTime=$[yyyymmddhh24miss-5/24/60]系统前5分钟时间。

如上图所示，设置按分钟调度，从00:00~23:59每5分钟调度一次。

通过脚本模式配置同步任务

如果您需要通过脚本模式配置此任务，单击工具栏中的转换脚本，选择确认即可进入脚本模式。



您可以根据自身进行配置，示例脚本如下。

```
{
"type": "job",
"version": "1.0",
"configuration": {
"reader": {
"plugin": "loghub",
"parameter": {
"datasource": "loghub_lzz",//数据源名，需要和您添加的数据源名一致。
"logstore": "logstore-ut2",//目标日志库的名字，LogStore是日志服务中日志数据的采集、存储和查询单元。
"beginDateTime": "${startTime}",//数据消费的开始时间位点，为时间范围（左闭右开）的左边界。
"endDateTime": "${endTime}",//数据消费的开始时间位点，为时间范围（左闭右开）的右边界。
"batchSize": 256,//一次读取的数据条数，默认为256。
"splitPk": "",
"column": [
"key1",
"key2",
"key3"
]
}
},
"writer": {
"plugin": "odps",
"parameter": {
"datasource": "odps_first",//数据源名，需要和您添加的数据源名一致。
"table": "ok",//目标表名。
"truncate": true,
"partition": "",//分区信息。
"column": [//目标列名。
"key1",
"key2",
"key3"
]
}
},
"setting": {
"speed": {
"mbps": 8,//作业速率上限。
"concurrent": 7//并发数。
}
}
}
}
```

2.7.7 DataHub通过数据集成批量导入数据

本文将为您介绍如何通过数据集成对离线DataHub进行数据的导入操作。

数据集成是阿里巴巴集团提供的数据同步平台。该平台具备可跨异构数据存储系统、可靠、安全、低成本、可弹性扩展等特点，可为20多种数据源提供不同网络环境下的离线（全量/增量）数据进出通道。数据源类型的详情请参见#unique_25。

准备工作

1. #unique_30，并创建账号的访问秘钥，即AccessID和AccessKey。

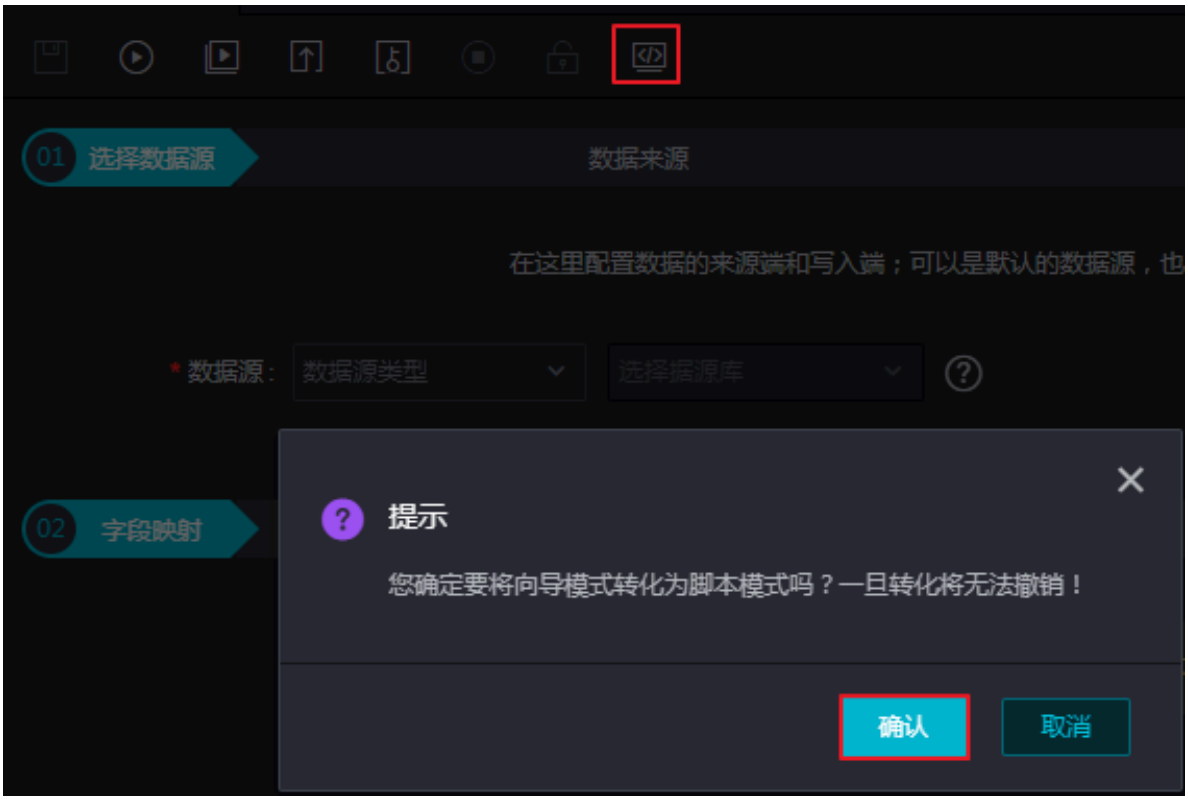
2. 开通MaxCompute，这样会自动产生一个默认的MaxCompute数据源，并使用主账号登录DataWorks。
3. #unique_31。您可以在项目中协作完成工作流，共同维护数据和任务等，因此使用DataWorks之前需要先创建一个项目。

> 说明:
> 如果您想通过子账号创建数据集成任务，可以赋予其相应的权限。详情请参见#unique_32和#unique_250。

操作步骤

以脚本模式将Stream数据同步到DataHub为例，操作如下。

1. 以开发者身份登录DataWorks控制台，单击对应项目下的进入数据集成。
2. 进入项目空间概览 > 任务列表页面，单击右上角的新建任务。
3. 填写新建节点对话框中的配置，单击提交，进入数据同步任务配置页面。
4. 单击新建同步节点右上角的转换脚本，选择确认即可进入脚本模式。



5. 单击工具栏中的导入模板，填写导入模板对话框中的配置。

配置	说明
来源类型	此处选择Stream类型。
目标类型	此处选择Elasticsearch类型。
数据源	选择配置好的数据源。 说明: 如果没有提前配置数据源，可单击新增数据源进行新增操作。

6. 单击确认生成初始脚本，可根据自身情况进行配置。

```
{
"type": "job",
"version": "1.0",
"configuration": {
 "setting": {
   "errorLimit": {
     "record": "0"
   },
   "speed": {
     "mbps": "1",
     "concurrent": 1,//作业并发数
     "dmu": 1,//数据移动单位（DMU）是数据集成消耗资源（包含 CPU、内存、网络等资源分配）的度量单位
     "throttle": false
   }
 },
 "reader": {
   "plugin": "stream",
   "parameter": {
     "column": [//源端列名
       {
         "value": "field",//列属性
```

```
          "type": "string"
        },
        {
          "value": true,
          "type": "bool"
        },
        {
          "value": "byte string",
          "type": "bytes"
        }
      ],
      "sliceRecordCount": "100000"
    }
  },
  "writer": {
    "plugin": "datahub",
    "parameter": {
      "datasource": "datahub",//数据源名
      "topic": "xxxx",//Topic是DataHub订阅和发布的最小单位，您可以用Topic来
表示一类或者一种流数据。
      "mode": "random",//随机写入。
      "shardId": "0",//Shard 表示对一个Topic进行数据传输的并发通道，每个
Shard会有对应的ID。
      "maxCommitSize": 524288,//为了提高写出效率，待攒数据大小达到
maxCommitSize大小（单位MB）时，批量提交到目的端。默认是1048576 ，即1MB数据。
      "maxRetryCount": 500
    }
  }
}
}
```

7. 单击保存并运行。

说明:

- DataHub仅支持以脚本模式导入数据。
- 如果想选择新模板，可单击工具栏中的导入模板，一旦导入新模板，原有内容将会被全部覆盖。
- 同步任务保存后，直接单击运行，任务会立刻运行。

  您也可单击提交，将同步任务提交到调度系统中，调度系统会按照配置属性在从第二天开始自动定时执行。

参考文档

其他的配置同步任务详细信息请参见下述文档。

- 配置Reader插件。
- 配置Writer插件。

2.7.8 OTSStream配置同步任务

OTSStream插件主要用于Table Store增量数据的导出，增量数据可以看作操作日志，除数据本身外还附有操作信息。

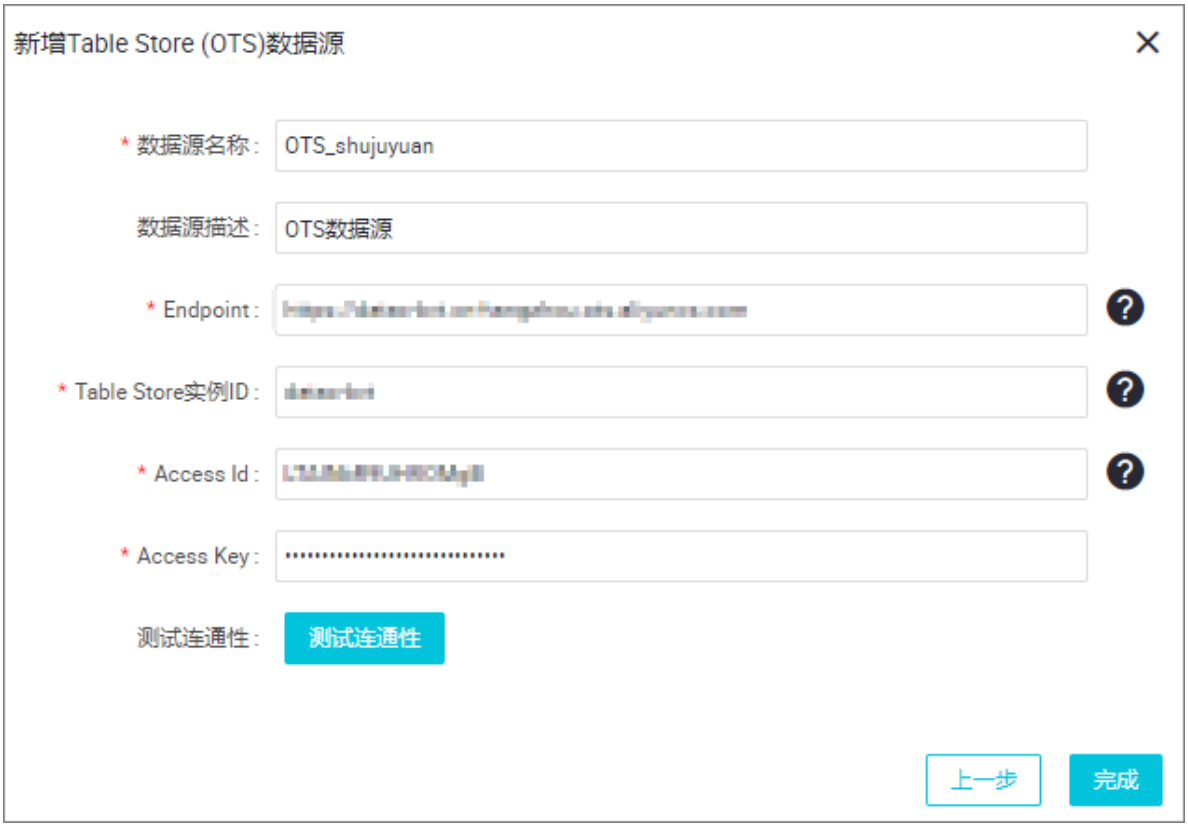
与全量导出插件不同，增量导出插件只有多版本模式，且不支持指定列，这与增量导出的原理有关，详情请参见#unique_254。

> 说明:
> OTSStream配置同步任务时，请注意以下几点：
>
> - 当前时间的前5分钟之前和24小时之内是可读数据。
> - 设置的结束时间不能超过系统显示的时间（即您设置的结束时间要比运行时间早5分钟）。
> - 配置日调度会有数据丢失。
> - 不可配置周期调度和月调度。
>
> 示例如下：
>
> 开始时间和结束时间，要包含操作Table Store表的时间，例如20171019162000您有向Table Store插入2条数据，那您的时间可以设置为开始时间：20171019161000，结束时间：20171019162600。

新增数据源

1. 以项目管理员身份登录DataWorks控制台，单击对应项目下的进入数据集成。
2. 进入同步资源管理 > 数据源页面，单击右上角的新增数据源。

3. 选择数据源类型为Table Store（OTS），填写对话框中的配置项。



配置	说明
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述。
Endpoint	Table Store Server的Endpoint，格式为`http://yyy.com`。
Table Store实例ID	Table Store服务对应的实例ID。
AccessID/AceessKey	访问密匙（AccessKeyID和AccessKeySecret），相当于登录密码。

4. 单击测试连通性。
5. 测试连通性通过后，单击确定。

通过向导模式配置同步任务

1. 进入项目空间概览 > 任务列表页面，单击右上角的新建任务。
2. 填写新建节点对话框中的配置，单击提交，进入数据同步任务配置页面。

3. 选择数据来源。

配置	说明
数据源	填写LogHub数据源的名称。
表	导出增量数据的表的名称。该表需要开启Stream，可以在建表时开启，或者使用UpdateTable接口开启。
开始时间	增量数据的时间范围（左闭右开）的左边界，格式yyyymmddhh24miss，单位毫秒。
结束时间	增量数据的时间范围（左闭右开）的右边界，格式yyyymmddhh24miss，单位毫秒。
状态表	用于记录状态的表的名称。
最大重试次数	从TableStore中读增量数据时，每次请求的最大重试次数，默认是30。
导出时序信息	是否导出时序信息，时序信息包含了数据的写入时间等。

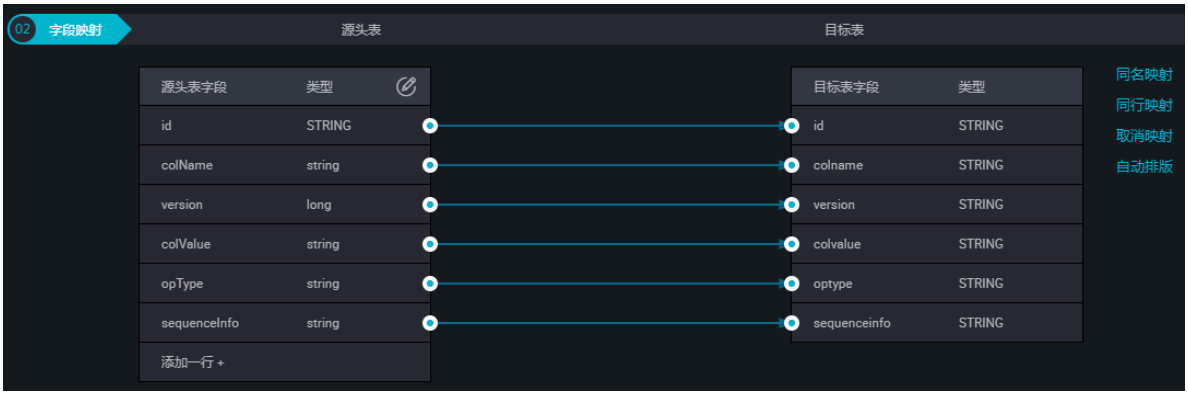
4. 选择数据去向。

   选择MaxCompute数据源及目标表。

配置	说明
数据源	填写配置的数据源名称。
表	选择需要同步的表。
分区信息	此处需同步的表是非分区表，所以无分区信息。
清理规则	· 写入前清理已有数据：导数据之前，清空表或者分区的所有数据，相当于insert overwrite。 · 写入前保留已有数据：导数据前不清理任何数据，每次运行数据都追加进去，相当于insert into。
压缩	默认选择不压缩。
空字符串作为null	默认选择否。

5. 字段映射。

选择字段的映射关系。需对字段映射关系进行配置，左侧源头表字段和右侧目标表字段为一一对应的关系。



6. 通道控制。

配置作业速率上限和脏数据检查规则。

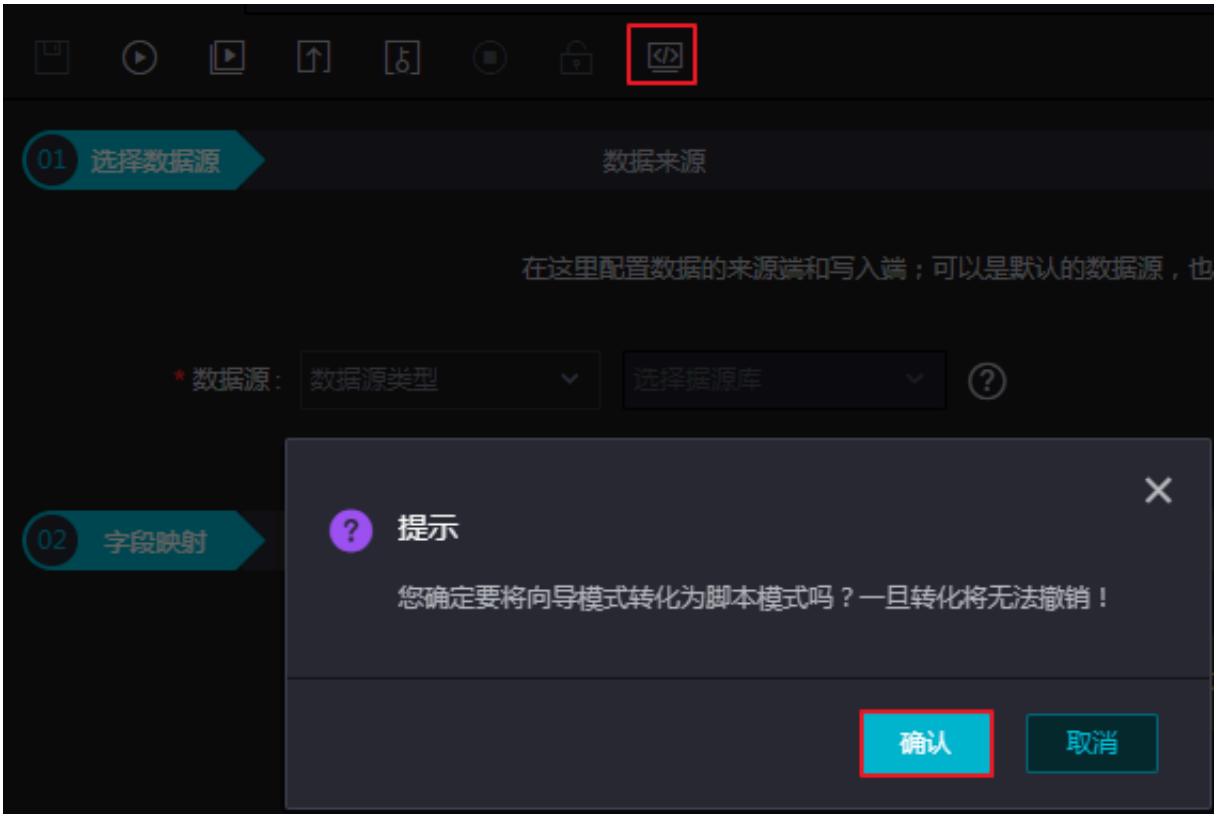
配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

7. 保存并运行任务。

   单击任务上方的运行按钮，将直接在数据集成页面运行任务，运行之前需要配置自定义参数的具体数值。

通过脚本模式配置同步任务

如果您需要通过脚本模式配置此任务，单击工具栏中的转换脚本，选择确认即可进入脚本模式。



您可根据自身进行配置，示例脚本如下。

```
{
  "type": "job",
  "version": "1.0",
  "configuration": {
    "reader": {
      "plugin": "otsstream",
      "parameter": {
        "datasource": "otsstream",//数据源名，保持跟您添加的数据源名一致
        "dataTable": "person",//导出增量数据的表的名称。该表需要开启 Stream
，可以在建表时开启，或者使用 UpdateTable 接口开启
        "startTimeString": "${startTime}",//增量数据的时间范围（左闭右
开）的左边界，格式yyyymmddhh24miss，单位毫秒
        "endTimeString": "${endTime}",//运行时间
        "statusTable": "TableStoreStreamReaderStatusTable",//用于记录状
态的表的名称
        "maxRetries": 30,//请求的最大重试次数
        "isExportSequenceInfo": false,
      }
    },
    "writer": {
      "plugin": "odps",
      "parameter": {
        "datasource": "odps_first",//数据源名
        "table": "person",//目标表名
        "truncate": true,
        "partition": "pt=${bdp.system.bizdate}",//分区信息
        "column": [//目标列名
          "id",
          "colname",
          "version",
          "colvalue",
          "optype",
```

```
            "sequenceinfo"
          ]
        }
      },
      "setting": {
        "speed": {
          "mbps": 7,//作业速率上限
          "concurrent": 7//并发数
        }
      }
    }
}
```

 说明:

- 关于运行时间参数和结束时间参数，有两种表现形式（配置任务选择其中一种）。
  - `"startTimeString": "${startTime}"`

    增量数据的时间范围（左闭右开）的左边界，格式yyyymmddhh24miss，单位毫秒。

    `"endTimeString": "${endTime}"`

    增量数据的时间范围（左闭右开）的右边界，格式yyyymmddhh24miss，单位毫秒。
  - `"startTimestampMillis":""`

    增量数据的时间范围（左闭右开）的左边界，单位毫秒。

    Reader插件会从statusTable中找对应startTimestampMillis的位点，从该点开始读取开始导出数据。

    若statusTable中找不到对应的位点，则从系统保留的增量数据的第一条开始读取，并跳过写入时间小于startTimestampMillis的数据。

    `"endTimestampMillis":" "`

    增量数据的时间范围（左闭右开）的右边界，单位毫秒。

    Reade插件startTimestampMilli位置开始导出数据后，当遇到第一条时间戳大于等于endTimestampMilli的数据时，结束导出数据，导出完成。

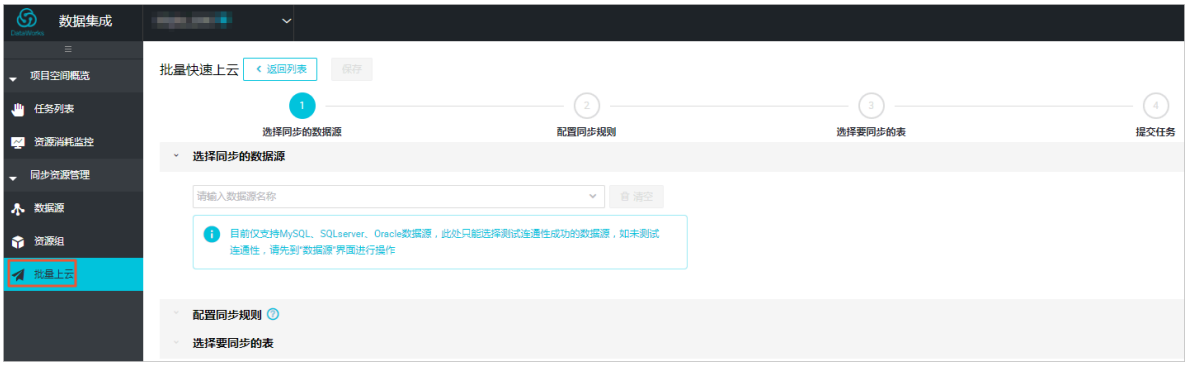
    当读取完当前全部的增量数据时，结束读取，即使未达endTimestampMillis。

    这个格式是时间式戳形式，单位毫秒。
- 如果配置isExportSequenceInfo项为true，如“isExportSequenceInfo”:true则会导出时序信息，目标会多出一行，目标column列则要多一列。时序信息包含了数据的写入时间等，默认该值为false，即不导出。

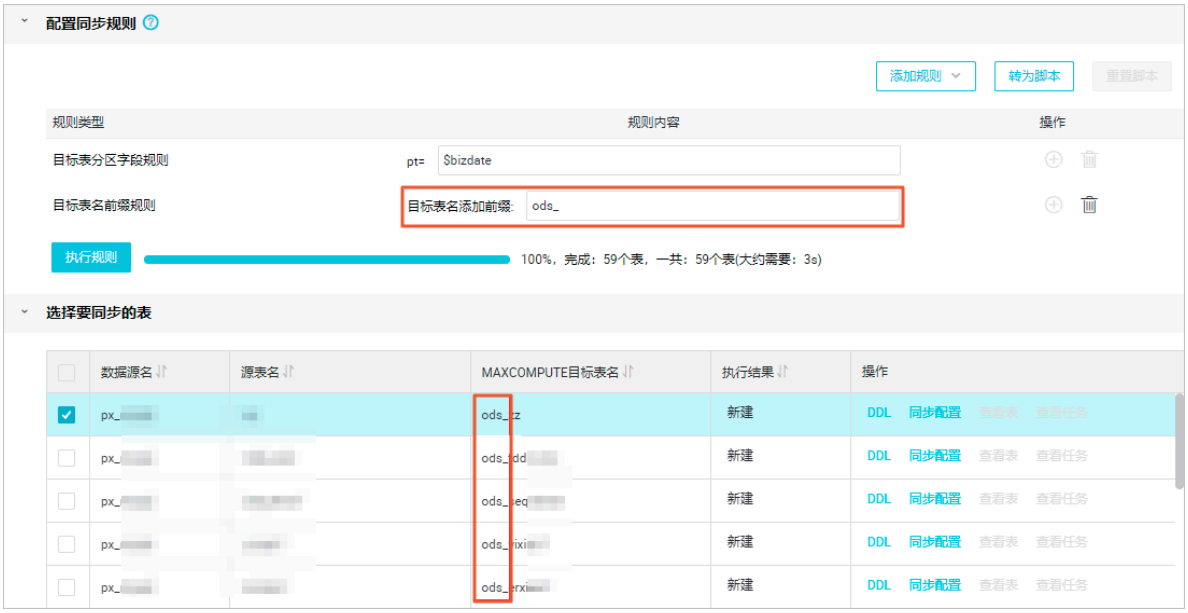
2.7.9 批量上云时给目标表名加上前缀

本文将为您介绍如何在批量上云时，给目标表名加上前缀。

1. 请参见#unique_256添加数据源。
2. 新建批量快速上云任务，并选择您创建的数据源。

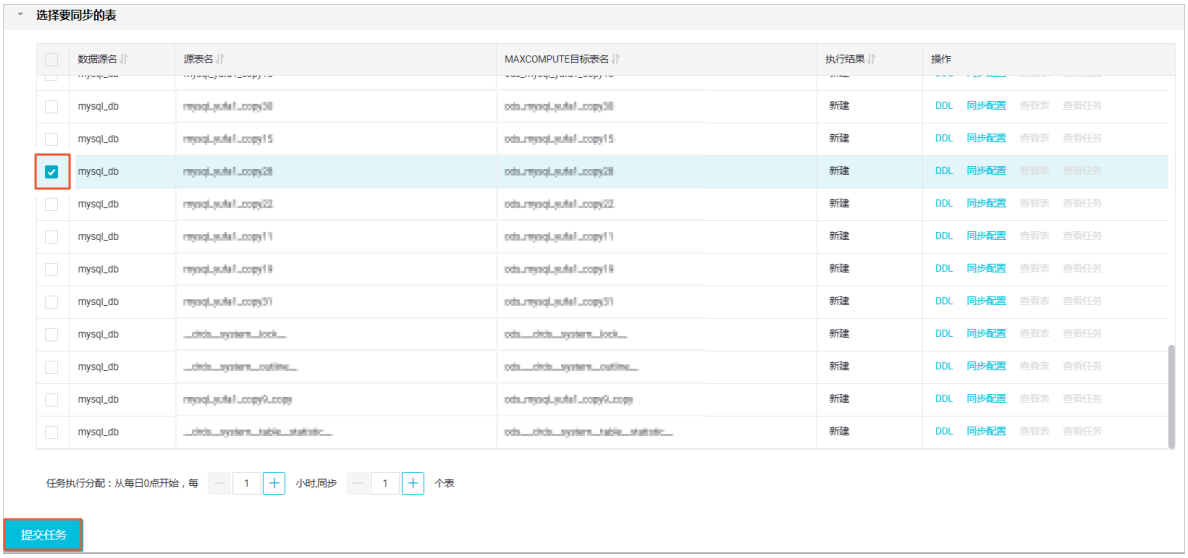


3. 单击添加规则，选择表名转换规则，输入您的表名转换正则表达式。本示例中使用`(.+)`匹配所有表头，使用`ods_$1`表示给表头加上前缀`ods_`。



4. 完成设置后，单击执行规则，您即可下方选择要同步的表处看到，表名已经进行了转换。

5. 勾选您需要同步的表，单击执行任务，即可在批量上云时完成对表前缀的设置。



2.7.10 RDBMS添加关系型数据库驱动最佳实践

RDBMS Reader插件通过JDBC连接远程RDBMS数据库，并执行相应的SQL语句将数据从RDBMS库中SELECT出来。目前支持达梦、DB2、PPAS、Sybase数据库的读取。RDBMS Reader是一个通用的关系数据库读插件，您可以通过添加、注册数据库驱动等方式增加各种关系型数据库的读支持。

背景信息

RDBMS Reader通过JDBC连接器连接到远程的RDBMS数据库，根据您配置的信息生成查询SQL语句并发送到远程RDBMS数据库，将该SQL执行返回的结果使用DataX自定义的数据类型拼装为抽象的数据集，并传递给下游Writer处理。

对于您配置的Table、Column、Where等信息，RDBMS Reader将其拼接为SQL语句发送到RDBMS数据库。对于您配置的querySql信息，RDBMS直接将其发送到RDBMS数据库。

目前RDBMS Reader支持大部分通用的关系数据库类型如数字、字符等，但也存在部分类型没有支持的情况，请注意检查您的数据库类型。

准备工作

在添加关系型数据库驱动前，您需要已购买ECS服务器作为您的自定义资源组资源，建议购买规格如下：

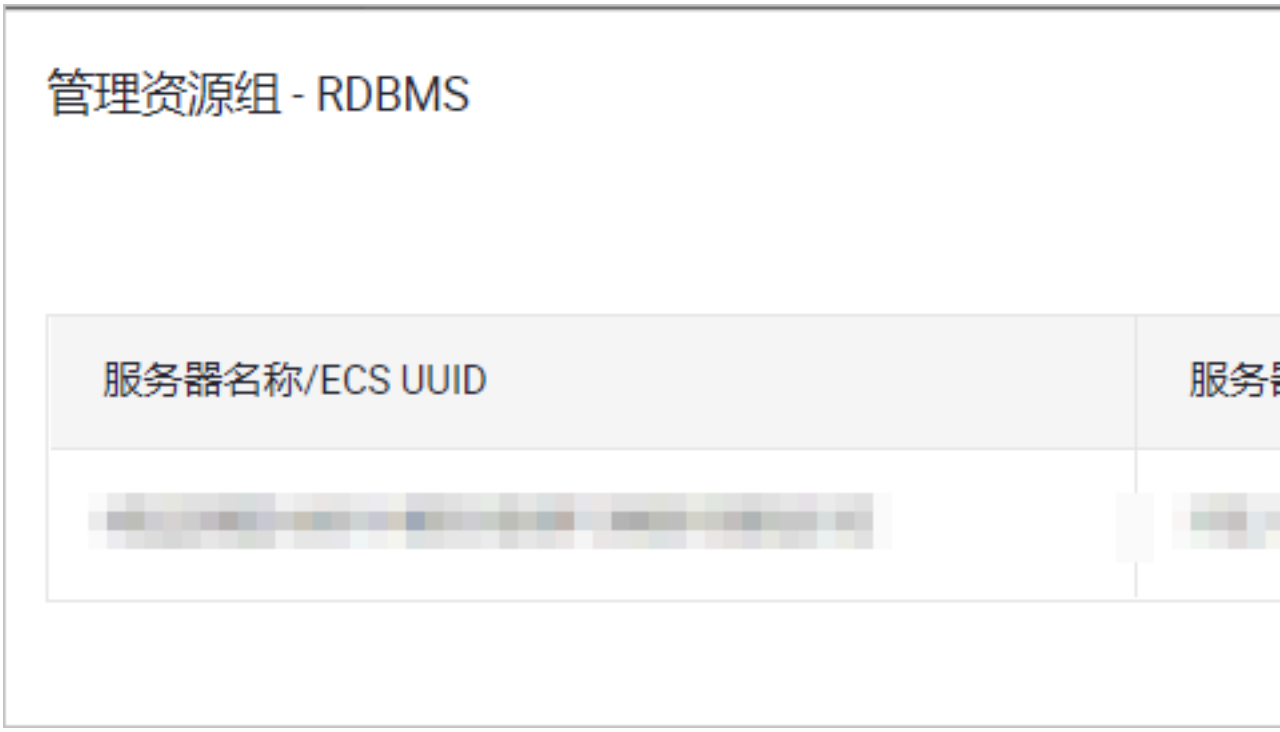
- 使用CentOS 6、CentOS 7或AliyunOS。
- 如果您添加的ECS需要执行MaxCompute任务或同步任务，需要检查当前ECS的python版本是否是Python2.6或2.7的版本（CentOS 5的Python版本为2.4，其它OS自带2.6以上版本）。

- 请确保ECS有访问公网能力，可以是否能ping通 www.aliyun.com 作为衡量标准。
- 建议ECS的配置为8核16G。

添加自定义资源组

首先您可以参考#unique_33添加自定义资源组：

1. 创建项目后，单击对应项目的进入数据集成。
2. 选择数据集成页面里的资源组 > 新增资源组。
3. 遵照步骤提示完成Agent安装与初始化，待服务器为可用状态时，则说明自定义资源组完成。



说明:

如果刷新后还是停止状态，您可以重启alisa命令。切换到admin账号，执行下述命令：

```
/home/admin/alisatasknode/target/alisatasknode/bin/serverct1
restart
```

添加MySQL驱动

我们以添加MySQL驱动为例，说明添加关系型数据库驱动操作步骤。

1. 进入RDBMS Reader对应目录，${DATAX_HOME}为DataX主目录，即/home/admin/datax3/plugin/reader/rdbmsreader目录。

```
[root@izbp1czjkv9fpzmsbv0qcdz rdbmsreader]# pwd
/home/admin/datax3/plugin/reader/rdbmsreader
[root@izbp1czjkv9fpzmsbv0qcdz rdbmsreader]# ls
```

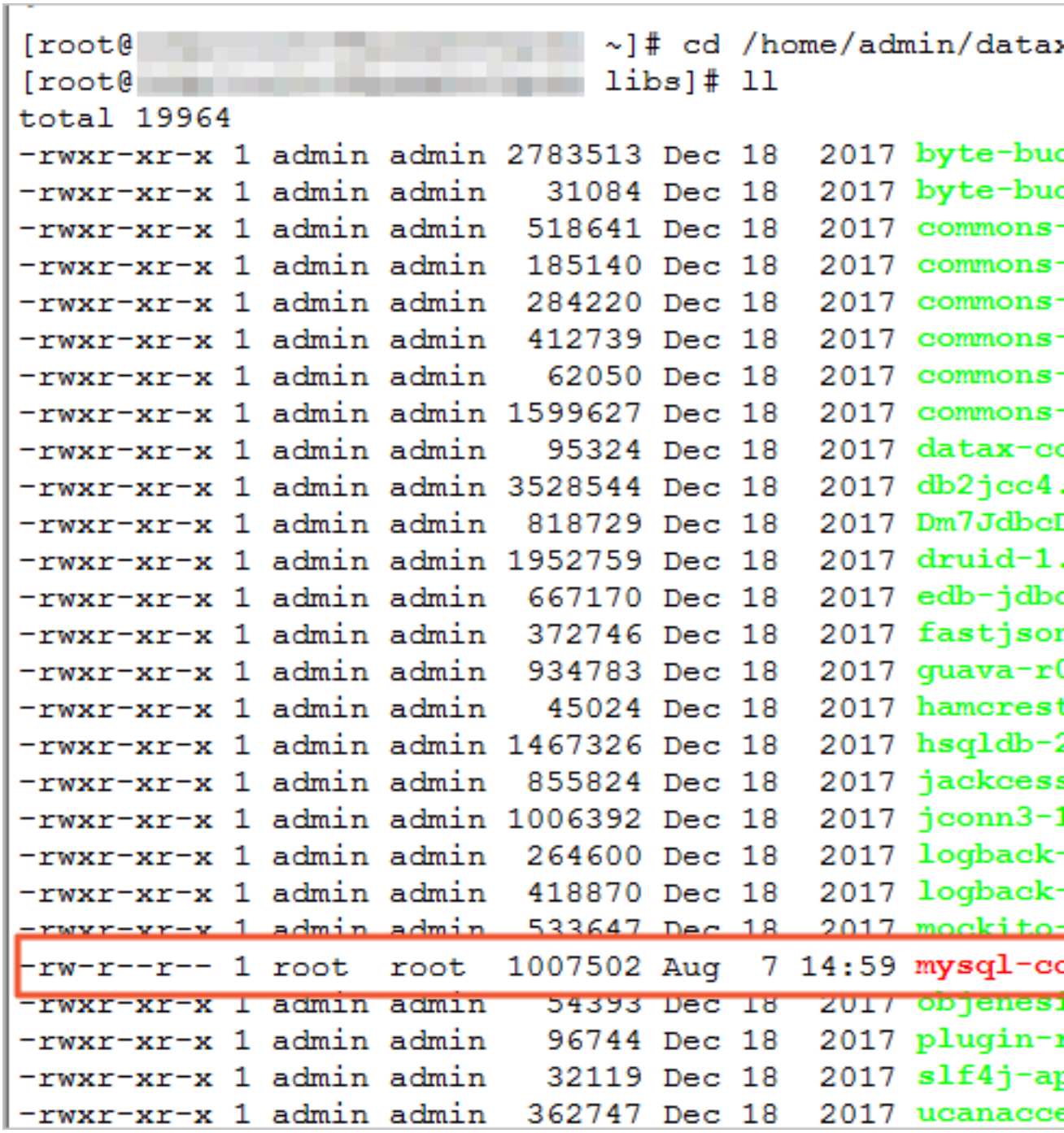
```
libs plugin.json rdbmsreader-0.0.1-SNAPSHOT.jar
```

2. **在RDBMS Reader插件目录下找到plugin.json配置文件，在此文件中注册您具体的数据库驱动，如下面的"com.mysql.jdbc.Driver"，放在drivers数组中，如下所示。RDBMS Reader插件在任务执行时会动态选择合适的数据库驱动连接数据库。**

```
[root@izbp1czjkv9fpzmsbv0qcdz rdbmsreader]# vim plugin.json
{
    "name": "rdbmsreader",
    "class": "com.alibaba.datax.plugin.reader.rdbmsreader.RdbmsReade
r",
    "description": "useScene: prod. mechanism: Jdbc connection using
 the database, execute select sql, retrieve data from the ResultSet
. warn: The more you know about the database, the less problems you
encounter.",
    "developer": "alibaba",
    "drivers":["dm.jdbc.driver.DmDriver", "com.sybase.jdbc3.jdbc.
SybDriver", "com.edb.Driver","com.mysql.jdbc.Driver"]
```

```
}
```

3. 在rdbmsreader插件目录下找到libs子目录，将您下载的mysql的jar包上传上去，如下图的mysql-connector-java-5.1.47.jar。



配置RDBMS数据同步任务

目前通过RDBMS Reader插件只能在脚本模式中配置同步任务，配置示例如下所示：

```
{
"job": {
        "setting": {
            "speed": {
                "byte": 1048576
```

```
            },
            "errorLimit": {
                "record": 0,
                "percentage": 0.02
            }
        },
        "content": [
            {
                "reader": {
                    "name": "rdbmsreader",
                    "parameter": {
                        "username": "xxxxx",
                        "password": "yyyyyy",
                        "column": [
                            "*",
                        ],
                        "splitPk": "id",
                        "connection": [
                            {
                                "table": [
                                    "a2"
                                ],
                                "jdbcUrl": [
                                    "jdbc:mysql://xxx.mysql.yy.
aliyuncs.com:3306/xxx"  //直接
配置您的SQL地址                                    ]
                            }
                        ],

                        "where": ""
                    }
                },
                "writer": {  //writer部分根据您的
需要配置即可                    "name": "streamwriter",
                    "parameter": {
                        "print": true
                    }
                }
            }
        ]
    }
}
```

2.7.11 独享数据集成资源组最佳实践

在数据集成任务高并发执行且无法错峰运行的情况下，企业需要独享的计算资源来保障数据快速、稳定地传输，此时您可以选择独享数据集成资源。

说明:

购买的独享数据集成资源和新增的数据源必须在同地域同购买购可用区，暂不支持跨可用区：

- 目前购买的独享数据集成资源需要和您的VPC可用区进行绑定，即独享数据集成资源需要和数据源在同一个可用区。
- 独享数据集成资源和独享调度资源绑定VPC时，选择需要访问的数据源所绑定的交换机。
- 独享数据集成资源绑定VPC后，独享数据集成资源能够访问您的VPC对应可用区的数据源，暂时不能直接访问您的VPC其他可用区内的数据源。

- 建议您在购买创建独享数据集成资源时，确认好可用区。
- 独享数据集成资源无法访问阿里云经典网络。如果您的数据源是经典网络，建议使用默认资源组进行同步任务运行。
- 目前正在开发独享数据集成资源支持一个VPC多个可用区的网络打通功能。

购买独享数据集成资源

1. 登录DataWorks控制台，进入资源列表 > 独享资源页面。
2. 如果您在该地域未购买过独享资源，单击右上角的新增独享资源。

3. 单击新增独享资源对话框中订单号后的购买，即可跳转至购买页面。

4. 进入购买页面后，请根据实际需要，选择相应的地域、独享资源类型、独享调度资源、资源数量和计费周期，单击立即购买。

> 说明:
> 此处的独享资源类型选择独享数据集成资源。

5. 确认订单信息无误后，勾选《DataWorks独享资源（包年包月）服务协议》，单击去支付。

> 说明:
> 独享资源不支持跨地域使用，即华东2（上海）地域的独享资源，只能给华东2（上海）地域的工作空间使用。

新增独享数据集成资源

1. 进入资源列表 > 独享资源页面，单击右上角的新增独享资源。
2. 填写新增独享资源对话框中的配置。

配置	说明
资源类型	资源的使用类型。独享资源包括独享调度资源和独享数据集成资源两种类型，分别适用于通用任务调度和数据同步任务专用。
资源名称	资源的名称，租户内唯一，请避免重复。 说明: 租户即主账号，一个租户（主账号）下可以有多个用户（子账号）。

配置	说明
资源备注	对资源进行简单描述。
订单号	此处选择购买的独享资源订单。如果没有购买，可以单击购买，跳转至售卖页进行购买。
可用区	单个地域提供了不同机器的可用区，请根据自身情况进行选择。

3. 配置完成后，单击创建，即可新增独享资源。

> 说明:
> 独享资源在20分钟内完成环境初始化，请耐心等待其状态更新为运行中。

专有网络绑定

独享资源部署在DataWorks托管的专有网络（VPC）中，如果需要与您自己的专有网络连通，需要进行专有网络绑定操作。

1. 单击相应资源后的专有网络绑定。

> 说明:
> 绑定前，需要进行RAM授权，让DataWorks拥有访问您的云资源的权限。

2. 授权完成后，单击右上角的新增绑定，填写新增专有网络绑定对话框中的配置。

- 如果没有可用的专有网络，您可以单击创建专有网络，跳转至专有网络控制台中的专有网络页面进行新建。

  单击创建专有网络，填写创建专有网络对话框中的配置，单击确定。

  创建完成后，即可跳转至专有网络列表页面进行查看。

- 如果没有可用的交换机，您可以单击创建交换机，跳转至专有网络控制台中的交换机页面进行新建。

  单击创建交换机，填写创建交换机对话框中的配置，单击确定。

  创建完成后，即可跳转至交换机列表页面进行查看。

- 如果没有可用的安全组，您可以单击创建安全组，跳转至ECS控制台中的安全组列表页面进行新建。

  单击创建安全组，填写创建安全组对话框中的配置，单击确定。

  创建完成后，即可跳转至安全组列表页面进行查看。

3. 配置完成后，单击创建。

购买RDS实例

1. 鼠标悬浮至左上角的图标，单击云数据库RDS版，进入实例列表页面。

2. 单击右上角的创建实例。

3. 选择购买页面的各配置项。

> 说明:
> 配置过程中，需特别注意版本、可用区和网络类型的选择，必须与上文的配置保持一致。

4. 配置完成后，单击立即购买。
5. 确认订单无误后，勾选《关系型数据库RDS服务条款》，单击去支付。
6. 购买完成后，即可返回实例列表页面进行查看。

设置白名单

1. 在实例列表页面，单击新建的RDS实例ID。
2. 单击左侧导航栏中的数据安全性。
3. 在白名单设置页签中，单击default白名单分组中的修改。
4. 在修改白名单分组对话框中，输入相应地域的白名单，并添加上文创建的专有网络的IP。
5. 进入账号管理和数据库管理页面，分别创建账号和数据库。

新增数据源

1. 以项目管理员身份进入DataWorks控制台，单击对应工作空间操作栏中的进入数据集成。
2. 单击数据源 > 新增数据源，弹出支持的数据源。
3. 在新增数据源弹出框中，选择数据源类型为MySQL。
4. 填写MySQL数据源的各配置项。

   MySQL数据源类型分为阿里云数据库（RDS）、连接串模式（数据集成网络可直接连通）和连接串模式（数据集成网络不可直接连通）。

   本文选择MySQL > 阿里云数据库（RDS）类型的数据源。

配置	说明
数据源类型	当前选择的数据源类型为MySQL > 阿里云数据库（RDS）。
数据源名称	数据源名称必须以字母、数字、下划线组合，且不能以数字和下划线开头。
数据源描述	对数据源进行简单描述，不得超过80个字符。

配置	说明
适用环境	可以选择开发或生产环境。 说明: 仅标准模式工作空间会显示此配置。
地区	选择相应的地区。
RDS实例ID	即上文创建的RDS的实例ID，您可以进入RDS控制台进行查看。
RDS实例主账号ID	您可以进入在RDS控制台安全设置页面进行查看。
用户名/密码	数据库对应的用户名和密码。

说明:
您需要先添加RDS白名单才能连接成功，详情请参见添加白名单。

5. 单击测试连通性。
6. 测试连通性通过后，单击确定。

修改归属工作空间

独享资源需绑定归属的工作空间，方可被任务真正使用。一个独享资源可分配给多个工作空间使用。

1. 进入DataWorks控制台的资源列表页面。
2. 单击相应资源后的修改归属工作空间。

3. 在修改归属对话框中勾选需要的工作空间，单击确定。

绑定工作空间后，即可在数据同步任务中使用独享数据集成资源。

2.8 常见问题

2.8.1 如何排查数据集成问题

当通过数据集成实现某操作出现问题时，首先要定位问题的相关信息。例如查看运行资源、数据源信息，确认配置任务的区域等。

查看运行资源

- 运行在默认的资源组上：

```
running in Pipeline[basecommon_ group_xxxxxxxxx]
```

- 运行在数据集成自定义资源组上：

```
running in Pipeline[basecommon_xxxxxxxxx]
```

- 运行在独享数据集成资源上：

```
running in Pipeline[basecommon_S_res_group_xxx]
```

查看数据源信息

当出现数据集成问题时，您可以参见添加数据源典型问题场景进行排查。

需要查看的数据源的相关信息如下：

1. 确认是什么数据源之间的同步。
2. 确认是什么环境的数据源。

   阿里云数据库、连接串模式（数据集成网络可直接连通）、连接串模式（数据集成网络不可直接连通）、VPC网络环境的数据源（RDS或其他数据源）、金融云环境（VPC环境，经典网络）。
3. 确认数据源测试连通性是否成功。

   请参见配置数据源文档，确认数据源的相关信息是否正确。通常填错的情况如下：

   - 多个数据源库填错。
   - 填写的信息中加了空格或特殊字符。
   - 不支持测试连通性的问题，例如连接串模式（数据集成网络不可直接连通）的数据源、除RDS的VPC环境的数据源。

确认配置任务的区域

进入DataWorks控制台，可以查看相关的区域，例如华东2、华南1、中国（香港）、亚太东南亚1、欧洲中部1、亚太东南2等，通常默认是华东2。

说明:

购买MaxCompute后，才能查查看相应的区域。

界面模式报错，复制排查码

如果报错，请复制排查码，并提供给处理人员。

解读日志报错

- 日志中报SQL语句执行失败（列包含关键字）。

```
2017-05-31 14:15:20.282 [33881049-0-0-reader] ERROR ReaderRunner -
Reader runner Received Exceptions:com.alibaba.datax.common.exception
.DataXException: Code:[DBUtilErrorCode-07]
```

错误解读：

读取数据库数据失败，请检查您配置的column/table/where/querySql或者向数据库管理员寻求帮助。

执行的SQL如下所示：

```
select **index**,plaid,plarm,fget,fot,havm,coer,ines,oumes from xxx
```

错误信息如下所示：

```
You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use
 near **index**,plaid,plarm,fget,fot,havm,coer,ines,oumes from xxx
```

排查思路：

1. 本地运行SQL语句`select **index**,plaid,plarm,fget,fot,havm,coer,ines, oumes from xxx`，查看其结果，通常也会有相应的报错。
2. 字段中有关键字index，解决方法加单引号或修改字段。

- 日志中报SQL语句执行失败（表名带有双引号包单引号）。

```
com.alibaba.datax.common.exception.DataXException: Code:[DBUtilErro
rCode-07]
```

错误解读：

读取数据库数据失败，请检查您配置的column/table/where/querySql或者向数据库管理员寻求帮助。

执行的SQL如下所示：

```
select /_+read_consistency(weak) query_timeout(100000000)_/ _ from**
 'ql_ddddd_[0-31]' **where 1=2
```

错误信息如下所示：

```
You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to
use near ‘‘ql_live_speaks[0-31]’ where 1=2’ at line 1 - com.mysql.
jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: You have an error
in your SQL syntax; check the manual that corresponds to your MySQL
```

```
 server version for the right syntax to use near **'‘ql_ddddd_[0-31
]' where 1=2' **
```

排查思路：

配置表名时，需要双引号包单引号。通常配置常量是双引号包单引号，例如“table”:[“‘qlddddd[0-31]’”]，直接去掉里面的单引号。

- **测试数据源连通性失败（Access denied for）。**

  连接数据库失败，数据库连接串：jdbc:mysql://xx.xx.xx.x:3306/t_demo，用户名：fn_test， 异常消息：Access denied for user ‘fn_test’@’%’ to database ‘t_demo’，请确定RDS中已经添加白名单。

  排查思路：

  - **通常出现Access denied for异常，是因为填写的信息有问题，请确认您填写的信息。**
  - **白名单或者用户的账号有没有对应数据库的权限，RDS管控台可以添加相应的白名单和授权。**

- **路由策略有问题，运行的池子oxs和ECS集群。**

```
2017-08-08 15:58:55 : Start Job[xxxxxxx], traceId **running
in Pipeline[basecommon_group_xxx_cdp_oxs]**ErrorMessage:Code:[
DBUtilErrorCode-10]
```

错误解读：

连接数据库失败，请检查您的账号、密码、数据库名称、IP、Port或者向数据库管理员寻求帮助（注意网络环境）。数据库连接失败，因为根据您配置的连接信息，无法从jdbc:oracle:thin:@xxx.xxxxx.x.xx:xxxx:prod 中找到可连接的JDBCUrl，请检查您的配置并进行修改。

- ava.lang.Exception：DataX无法连接对应的数据库。

  错误解读：

  出现该错误可能的原因如下：

  - 配置的ip/port/database/jdbc错误，无法连接。
  - 配置的username/password错误，鉴权失败。请和数据库管理员确认该数据库的连接信息是否正确。

  排查思路：

  情况一：

  - Oracle同步的RDS-PostgreSQL直接单击运行，不能在调度中运行，因为运行的池子不同。
  - 可以在添加RDS的数据源时，改成添加普通JDBC形式的数据源，这样Oracle同步的RDS-PostgreSQL是可以的。

  情况二：

  - VPC环境的RDS-PostgreSQL不能运行在自定义资源组上，因为VPC环境的RDS有反向代理功能，这样与用户自定义资源组存在网络问题。因此，通常VPC环境的RDS直接运行在DataWorks默认的资源即可。如果默认资源不能满足您的需要，要运行在自己的资源上，可以将VPC环境的RDS作为VPC环境JDBC形式的数据源，购买一个同网段的ECS。

    详情请参见 VPC环境数据同步配置
  - 通常VPC环境的RDS映射出的URL为`jdbc:mysql://100.100.70.1:4309/xxx`，100开头的IP是后台映射出来的，如果是一个域名的表现形式则为非VPC环境。
- HBase Writer不支持date类型。

  HBase同步到hbase:2017-08-15 11:19:29 : State: 4(FAIL) | Total: 0R 0B | Speed: 0R/s 0B/s | Error: 0R 0B | Stage: 0.0% ErrorMessage:Code:[Hbasewriter-01]

  错误解读：

  您填写的参数值不合法。

  Hbasewriter不支持date类型，目前支持的类型包括string、boolean、short、int、long、float和double。

  排查思路：

  - HBase的writer不支持date类型，所以在writer中不能配置date类型。
  - 直接配置string类型，因为HBase没有数据类型的概念，底层通常是byte数组。

- JSON格式配置错误。

  column配置错误。

  经DataX智能分析，该任务最可能的错误原因如下：

  ```
  com.alibaba.datax.common.exception.DataXException: Code:[Framework-
  02]
  ```

  错误解读：

  DataX引擎运行过程出错，详情请参见DataX运行结束时的错误诊断信息。

  java.lang.ClassCastException：com.alibaba.fastjson。JSONObject cannot be cast to java.lang.String

  排查思路：

  发现其JSON配置有问题。

  ```
  writer端：
  “column”:[
  {
  “name”:“busino”,
  “type”:“string”
  }
  ]
  正确的写法：
  “column”:[
  {
  “busino”
  }
  ]
  ```

- JSON List编写缺少[]。

  经DataX智能分析，该任务最可能的错误原因如下所示：

  ```
  com.alibaba.datax.common.exception.DataXException: Code:[Framework-
  02]
  ```

  错误解读：

  DataX引擎运行过程出错，详情请参见DataX运行结束时的错误诊断信息。

  ```
  java.lang.String cannot be cast to java.util.List - java.lang.String
   cannot be cast to java.util.List
  at com.alibaba.datax.common.exception.DataXException.asDataXExc
  eption(DataXException.java:41)
  ```

  排查思路：

  少了[]，list型变成其他的形式，找到对应的地方填上[]即可解决问题。

- 权限问题
  - 缺少delete权限。

    MaxCompute同步到RDS-MySQL，报错如下：

    ```
    ErrorMessage:Code:[DBUtilErrorCode-07]
    ```

    错误解读：

    读取数据库数据失败，请检查您配置的column/table/where/querySql或者向数据库管理员寻求帮助。

    执行的SQL如下所示：

    ```
    delete from fact_xxx_d where sy_date=20170903
    ```

    具体错误信息如下所示：

    ```
    **DELETE command denied** to user ‘xxx_odps’@‘[xx.xxx.xxx.xxx](
    http://xx.xxx.xxx.xxx)’ for table ‘fact_xxx_d’ - com.mysql.jdbc.
    exceptions.jdbc4.MySQLSyntaxErrorException: DELETE command denied
     to user ‘xxx_odps’@‘[xx.xxx.xxx.xxx](http://xx.xxx.xxx.xxx)’ for
    table 'fact_xxx_d’
    ```

    排查思路：

    DELETE command denied to没有删除此表的权限，到相应的数据库设置相关表的删除权限。
  - 缺少drop权限。

    ```
    Code:[DBUtilErrorCode-07]
    ```

    错误解读：

    读取数据库数据失败，请检查您配置的column/table/where/querySql或者向数据库管理员寻求帮助。

    执行的SQL为`truncate table be_xx_ch`

    具体错误信息如下所示：

    ```
    **DROP command denied to user** ‘xxx’@‘[xxx.xx.xxx.xxx](http://
    xxx.xx.xxx.xxx)’ for table ‘be_xx_ch’ - com.mysql.jdbc.exceptions
    ```

```
.jdbc4.MySQLSyntaxErrorException: DROP command denied to user ‘xxx
’@‘[xxx.xx.xxx.xxx](http://xxx.xx.xxx.xxx)’ for table 'be_xx_ch’
```

排查思路：

MySQL Writer配置执行前准备语句truncate删除表中的数据报上面错误，是因为没有drop的权限。

- **ADS权限问题。**

```
2016-11-04 19:49:11.504 [job-12485292] INFO  OriginalConfPretreat
mentUtil - Available jdbcUrl:jdbc:mysql://100.98.249.103:3306/
ads_rdb?yearIsDateType=false&zeroDateTimeBehavior=convertToNull&
tinyInt1isBit=false&rewriteBatchedStatements=true.
```

```
2016-11-04 19:49:11.505 [job-12485292] WARN  OriginalConfPretreat
mentUtil
```

您的配置文件中的列配置存在一定的风险. 因为您未配置读取数据库表的列，当您的表字段个数、类型有变动时，可能影响任务正确性甚至会运行出错。请检查您的配置并进行修改。

```
2016-11-04 19:49:11.528 [job-12485292] INFO Writer$Job
```

如果是MaxCompute>ADS的数据同步，需要完成以下两方面的授权。

- ADS官方账号至少需要有需要同步的表的describe和select权限，因为ADS系统需要获取MaxCompute需要同步表的结构和数据信息。
- 您配置的ADS数据源访问账号AK，需要拥有向指定的ADS数据库发起load data的权限，您可以在ADS系统中添加授权。

```
2016-11-04 19:49:11.528 [job-12485292] INFO Writer$Job
```

如果是RDS（或其他非MaxCompute数据源）>ADS的数据同步，实现逻辑为先将数据装载到MaxCompute临时表，再从MaxCompute临时表同步至ADS，中转MaxCompute项目为cdp_ads_project，中转项目账号为cloud-data-pipeline@aliyun-inner.com。

权限方面：

- ADS官方账号至少需要有需要同步的表（这里指MaxCompute临时表）的describe和select权限，因为ADS系统需要获取MaxCompute需要同步的表的结构和数据信息，此部分部署时已经完成授权。
- 中转MaxCompute对应的账号cloud-data-pipeline@aliyun-inner.com，要拥有向指定的ADS数据库发起load data的权限，您可以在ADS系统中添加授权。

排查思路：

出现此问题是因为没有设置load data权限。

中转项目账号为cloud-data-pipeline@aliyun-inner.com，权限方面：ADS官方账号至少需要拥有需要同步的表（这里指 MaxCompute 临时表）的 describe 和 select 权限，因为ADS系统需要获取MaxCompute需要同步的表的结构和数据信息，此部分部署时已经完成授权，登录ADS管控台给ADS授予load data的权限。

· 白名单问题

- 没有添加白名单导致测试连通性失败。

  测试连接失败，测试数据源连通性失败：

  ```
  error message: **Timed out after 5000** ms while waiting for a
  server that matches ReadPreferenceServerSelector{readPreference=
  primary}. Client view of cluster state is {type=UNKNOWN, servers
  ```

```
=[{[address:3717=dds-bp1afbf47fc7e8e41.mongodb.rds.aliyuncs.com
](http://address:3717=dds-bp1afbf47fc7e8e41.mongodb.rds.aliyuncs
.com), type=UNKNOWN, state=CONNECTING, exception={com.mongodb.
MongoSocketReadException: Prematurely reached end of stream}},
 {[address:3717=dds-bp1afbf47fc7e8e42.mongodb.rds.aliyuncs.com](
http://address:3717=dds-bp1afbf47fc7e8e42.mongodb.rds.aliyuncs.
com), type=UNKNOWN, state=CONNECTING,** exception={com.mongodb.
MongoSocketReadException: Prematurely reached end of stream**}}]
```

排查思路

非VPC环境的MongoDB，添加数据源时报Timed out after 5000，白名单添加有问题。

> **说明:**
> **如果您使用的是云数据库MongoDB版，MongoDB默认会有root账号。出于安全策略的考虑，数据集成仅支持使用MongoDB数据库对应账号进行连接，您添加使用MongoDB数据源时，也请避免使用root作为访问账号。**

- **白名单不全。**

```
for Code:[DBUtilErrorCode-10]
```

错误解读:

连接数据库失败，请检查您的账号、密码、数据库名称、IP、Port或者向数据库管理员寻求帮助（注意网络环境）。

错误信息如下所示:

```
java.sql.SQLException: Invalid authorization specification,
message from server: "#**28000ip not in whitelist, client ip is xx
.xx.xx.xx".**
2017-10-17 11:03:00.673 [job-xxxx] ERROR RetryUtil - Exception
when calling callable
```

排查思路:

白名单没有添全，没有将用户自己的资源添加白名单内。

· **数据源信息填写错误。**

- **脚本模式配置缺少相应数据源信息（could not be blank）。**

```
2017-09-06 12:47:05 [INFO] Success to fetch meta data for table
with **projectId [43501]** **项目ID **and instanceId **[mongodb]数
据源名.**
2017-09-06 12:47:05 [INFO] Data transport tunnel is CDP.
2017-09-06 12:47:05 [INFO] Begin to fetch alisa account info
for 3DES encrypt with parameter account: [zz_683cdbcefba143b7b
709067b362d4385].
2017-09-06 12:47:05 [INFO] Begin to fetch alisa account info
for 3DES encrypt with parameter account: [zz_683cdbcefba143b7b
709067b362d4385].
```

```
[Error] Exception when running task, message:** Configuration
property [accessId]通常是odps数据源要填写的信息 could not be blank!**
```

排查思路：

报错显示没有相应的 accessId 信息，通常出现这种现象是脚本模式，查看用户配置的 json 代码，是否忘记写相应的的 数据源名。

- 未填写数据源配置。

```
2017-10-10 10:30:08 INFO
 ==================================================
File “/home/admin/synccenter/src/Validate.py”, line 16, in notNone

raise Exception(“Configuration property [%s] could not be blank!”
 % (context))
**Exception: Configuration property [username] could not be blank!
**
```

排查思路：

■ 根据正常的打出的日志对比：

```
[56810] and instanceId(instanceName) [spfee_test_mysql]…
2017-10-09 21:09:44 [INFO] Success to fetch meta data for table
 with projectId [56810] and instanceId [spfee_test_mysql].
```

■ rds-mysql反应出这样的信息说明调数据源失败成功，而且报用户为空，说明没有配置数据源或数据源的位置配置错误，这个用户就将数据源的位置配置错误。

- DRDS连接数据超时。

MaxCompute同步数据到DRDS，经常出现下面的错误：

```
[2017-09-11 16:17:01.729 [49892464-0-0-writer] WARN CommonRdbm
sWriter$Task
```

回滚此次写入，采用每次写入一行方式提交，原因如下：

```
com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: **
Communications link failure  **
```

```
The last packet successfully received from the server was 529
milliseconds ago.  The last packet sent successfully to the server
 was** 528 milliseconds ago**.
```

排查思路：

datax客户端的超时，在添加DRDS数据源的时候可以加上`?useUnicode=true&characterEncoding=utf-8&socketTimeout=3600000`超时参数

示例如下：

```
jdbc:mysql://10.183.80.46:3307/ae_coupon?useUnicode=true&
characterEncoding=utf-8&socketTimeout=3600000
```

- **系统内部问题。**

排查思路：

通常是在开发环境将JSON格式改错了然后直接保存，保存后通常会报系统内部问题。界面显示为空白，碰到这样的问题，直接将您的项目名和节点名称提供给技术支持在后台进行处理。

- **脏数据**
  - **脏数据（String[“”]不能转为Long）。**

```
2017-09-21 16:25:46.125 [51659198-0-26-writer] ERROR WriterRunner
 - Writer Runner Received Exceptions:
```

```
com.alibaba.datax.common.exception.DataXException: Code:[Common-01
]
```

错误解读：

同步数据出现业务脏数据情况，数据类型转换错误。String[“”]不能转为Long。

排查思路：

String[“”]不能转为Long：两张表格中的建表语句一样，报上述错误是因为字段类型中的空字段不能转换成Long类型，直接配置成String类型。

- **脏数据（Out of range value）。**

```
2017-11-07 13:58:33.897 [503-0-0-writer] ERROR StdoutPlug
inCollector
脏数据:
{“exception”:“Data truncation: Out of range value for column ‘
id’ at row 1”,“record”:[{“byteSize”:2,“index”:0,“rawData”:-3,“
type”:“LONG”},{“byteSize”:2,“index”:1,“rawData”:-2,“type”:“LONG
”},{“byteSize”:2,“index”:2,“rawData”: “其他” ,“type”:“STRING”},{“
```

```
byteSize”:2,“index”:3,“rawData”: “其他” ,“type”:“STRING”}],“type”:“
writer”}
```

排查思路：

mysql2mysql，源端设置的是smallint(5)，目标端是int(11) unsigned，因为smallint(5)范围有负数，unsigned不允许有负数，所以产生脏数据。

- **脏数据（存储emoji）。**

  数据表配置成了可以存储emoji的，同步时报脏数据。

  排查思路：

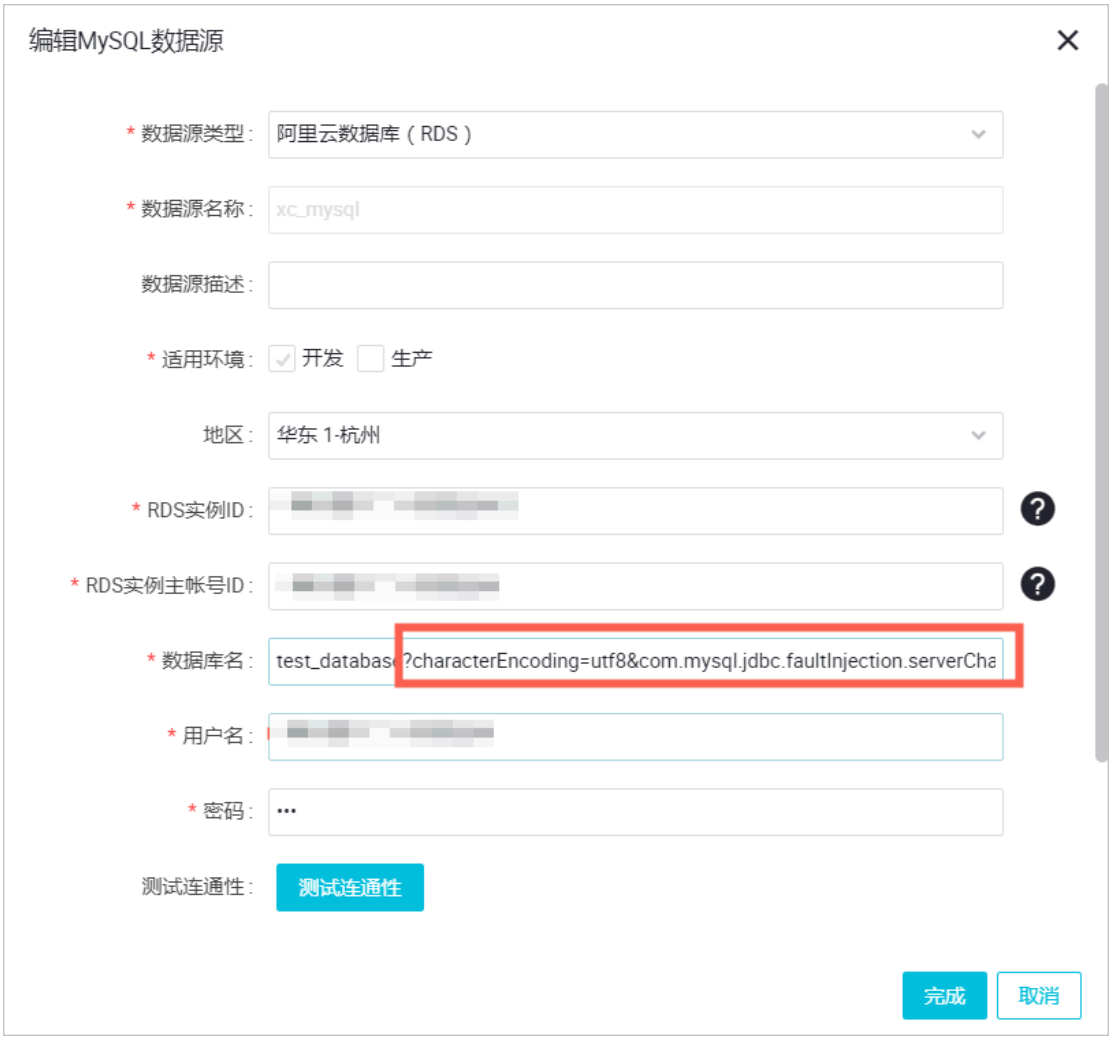
  同步emoji时报错脏数据，需要修改编码格式：

  ■ JDBC形式添加数据源

  ```
  jdbc:mysql://xxx.x.x.x:3306/database?characterEncoding=utf8&com
  .mysql.jdbc.faultInjection.serverCharsetIndex=45
  ```

  ■ 实例ID形式添加数据源

  在数据库名后拼接`?characterEncoding=utf8&com.mysql.jdbc.faultInjection.serverCharsetIndex=45`**。**



- **空字段引起的脏数据。**

```
{“exception”:“Column ‘xxx_id’ cannot be null”,“record”:[{“byteSize
”:0,“index”:0,“type”:“LONG”},{“byteSize”:8,“index”:1,“rawData”:-1
```

```
,“type”:“LONG”},{“byteSize”:8,“index”:2,“rawData”:641,“type”:“LONG
”}
```

经DataX智能分析，该任务最可能的错误原因如下所示：

```
com.alibaba.datax.common.exception.DataXException: Code:[Framework
-14]
```

错误解读：

DataX传输脏数据超过用户预期，该错误通常是由于源端数据存在较多业务脏数据导致。请仔细检查DataX汇报的脏数据日志信息，或者您可以适当调大脏数据阈值。

脏数据条数检查不通过，限制是1条，但实际上捕获了7条。

排查思路：

设置Column ‘xxx_id’ cannot be null字段不能为空，但数据中用空数据导致脏数据，修改其数据或对字段进行修改。

- **Data too long for column ‘flash’字段设置太小引起的脏数据。**

```
2017-01-02 17:01:19.308 [16963484-0-0-writer] ERROR StdoutPlug
inCollector
脏数据：
{“exception”:“Data truncation: Data too long for column ‘flash’ at
 row 1”,“record”:[{“byteSize”:8,“index”:0,“rawData”:1,“type”:“LONG
”},{“byteSize”:8,“index”:3,“rawData”:2,“type”:“LONG”},{“byteSize
”:8,“index”:4,“rawData”:1,“type”:“LONG”},{“byteSize”:8,“index”:5
,“rawData”:1,“type”:“LONG”},{“byteSize”:8,“index”:6,“rawData”:1,“
type”:“LONG”}
```

排查思路：

设置Data too long for column ‘flash’字段设置太小，但数据中数据太大导致脏数据，修改其数据或对字段进行修改。

- **read-only数据库权限设置问题，设置只读权限。**

```
2016-11-02 17:27:38.288 [12354052-0-8-writer] ERROR StdoutPlug
inCollector
脏数据：
{“exception”:“The MySQL server is running with the --read-only
option so it cannot execute this statement”,“record”:[{“byteSize
”:3,“index”:0,“rawData”:201,“type”:“LONG”},{“byteSize”:8,“index”:
1,“rawData”:1474603200000,“type”:“DATE”},{“byteSize”:8,“index”:2,“
```

```
rawData”:“9月23号12点” ,“type”:“STRING”},{“byteSize”:5,“index”:3,“
rawData”:“12:00”,“type”:“STRING”}
```

排查思路：

设置read-only模式，同步数据全为脏数据，修改其数据库模式，运行可以写入。

- **分区错误。**

  参数配置为$[yyyymm] 报错，日志如下所示：

```
[2016-09-13 17:00:43]2016-09-13 16:21:35.689 [job-10055875] ERROR
Engine
```

  经DataX智能分析，该任务最可能的错误原因如下：

```
com.alibaba.datax.common.exception.DataXException: Code:[
OdpsWriter-13]
```

  错误解读：

  执行MaxCompute SQL时抛出异常，可重试。MaxCompute目的表在运行MaxCompute SQL时抛出异常，请联系MaxCompute管理员处理。SQL内容如下所示：

```
alter table db_rich_gift_record add IF NOT EXISTS
      partition(pt='${thismonth}');
```

  排查思路：

  由于加了单引号，调度参数替换无效，解决方法：’${thismonth}’去掉引号调度参数。

- **column没有配成数组形式。**

```
Run  command failed.
com.alibaba.cdp.sdk.exception.CDPException: com.alibaba.fastjson.
JSONException: syntax error, **expect {,** actual error, pos 0
at com.alibaba.cdp.sdk.exception.CDPException.asCDPException(
CDPException.java:23)
```

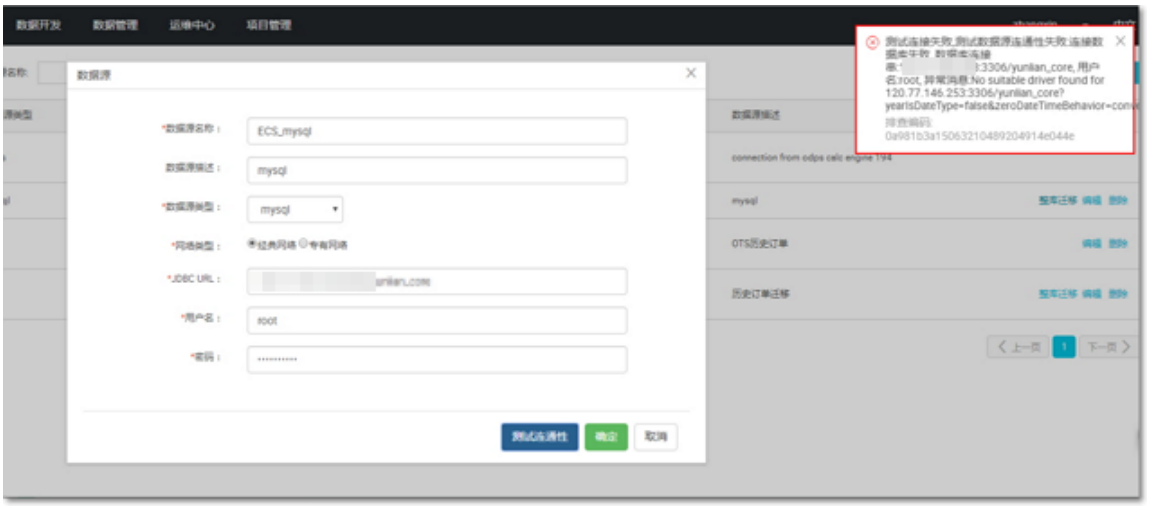
  排查思路：

  JSON有问题，如下所示：

```
"plugin": “mysql”,**
“parameter”: {
“datasource”: “xxxxx”,
** “column”: “uid”,**
“where”: “”,
“splitPk”: “”,
“table”: “xxx”
}
```

```
**“column”: “uid”,-----没有配成数组形式**
```

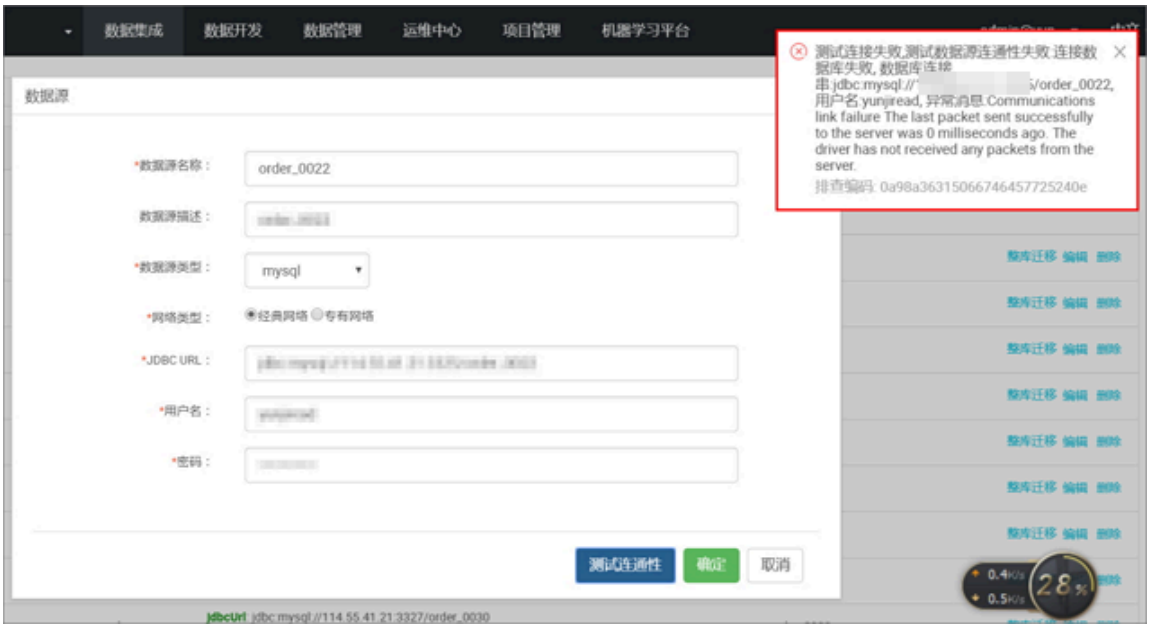
- JDBC的格式填写错误。



排查思路：

JDBC格式填错，正确的格式是：`jdbc:mysql://ServerIP:Port/Database`。

- 测试连通性失败。



排查思路：

- **防火墙对IP和端口账号有没有相关的限制。**
- **安全组的端口开发情况。**

- 日志中报uid[xxxxxxxx]问题。

```
Run  command failed.
com.alibaba.cdp.sdk.exception.CDPException: RequestId[F9FD049B-
xxxx-xxxx-xxx-xxxx] Error: CDP server encounter problems, please
contact us, reason: 获取实例的网络信息发生异常,请检查RDS购买者id和RDS实例
```

```
名,uid[xxxxxxxx],instance[rm-bp1cwz5886rmzio92]ServiceUnavailable
 : The request has failed due to a temporary failure of the server
.
RequestId : F9FD049B-xxxx-xxxx-xxx-xxxx
```

排查思路：

通常RDS同步至MaxCompute时，如果报上述错误，您可以直接将RequestId：F9FD049B-xxxx-xxxx-xxx-xxxx复制给RDS人员。

- **MongoDB中的query参数错误。**

  MongoDB同步到MySQL报下面的问题，排查出JSON没有写好，是JSON中的query参数没有配置好。

```
Exception in thread “taskGroup-0” com.alibaba.datax.common.
exception.DataXException: Code:[Framework-13]
```

错误解读：

DataX插件运行时出错，具体原因请参见DataX运行结束时的错误诊断信息。

```
org.bson.json.JsonParseException: Invalid JSON input. Position: 34
. Character: ‘.’.
```

排查思路：

- **错误示例：**`“query”:”{‘update_date’:{’$gte’:new Date().valueOf()/1000}}”`**，不支持如**`new Date()`**的参数。**
- **正确示例：**`“query”:”{‘operationTime’{’$gte’:ISODate(’${last_day}T00:00:00.424+0800’)}}”`**。**

- **Cannot allocate memory**

```
2017-10-11 20:45:46.544 [taskGroup-0] INFO  TaskGroupContainer -
taskGroup[0] taskId[358] attemptCount[1] is started
Java HotSpot™ 64-Bit Server VM warning: INFO: os::commit_memory
(0x00007f15ceaeb000, 12288, 0) failed; error=‘**Cannot allocate
memory’** (errno=12)
```

排查思路：

内存不够。如果运行在自己的资源上，需要自行添加内存。如果运行在阿里的资源上，请提交工单进行咨询。

- **max_allowed_packet参数错误。**

  错误信息如下所示：

```
Packet for query is too large (70 > -1). You can change this
value on the server by setting the max_allowed_packet’ variable
. - **com.mysql.jdbc.PacketTooBigException: Packet for query is
```

```
too large (70 > -1). You can change this value on the server by
setting the max_allowed_packet' variable.**
```

排查思路：

- **max_allowed_packet参数用来控制其通信缓冲区的最大长度。MySQL根据配置文件会限制server接受的数据包大小。有时候大的插入和更新会被max_allowed_packet参数限制掉，导致失败。**
- **max_allowed_packet参数的设置太大就改小的一点，通常10m=10_1024_1024。**

- **HTTP Status 500读取日志失败。**

```
Unexpected Error:
Response is com.alibaba.cdp.sdk.util.http.Response@382db087[proxy
=HTTP/1.1 500 Internal Server Error [Server: Tengine, Date: Fri,
27 Oct 2017 16:43:34 GMT, Content-Type: text/html;charset=utf-8,
Transfer-Encoding: chunked, Connection: close,
**HTTP Status 500** - Read timed out**type** Exception report**
message****Read timed out****description****The server encountered
 an internal error that prevented it from fulfilling this request.
++++exception**
java.net.SocketTimeoutException: Read timed out
```

排查思路：

调度运行报500的问题，若是运行在默认资源上，读取日志失败，请直接联系技术支持帮您解决。如果运行在您自己的资源上，请重启Alisa即可。

说明：

如果刷新后还是停止状态，您可以重启Alisa命令：切换到admin账号执行`/home/admin/alisatatasknode/target/alisatatasknode/bin/serverct1 restart`**。**

- **hbasewriter参数：hbase.zookeeper.quorum配置错误。**

```
2017-11-08 09:29:28.173 [61401062-0-0-writer] INFO  ZooKeeper -
Initiating client connection, connectString=xxx-2:2181,xxx-4:2181
,xxx-5:2181,xxxx-3:2181,xxx-6:2181 sessionTimeout=90000 watcher=
hconnection-0x528825f50x0, quorum=node-2:2181,node-4:2181,node-5:
2181,node-3:2181,node-6:2181, baseZNode=/hbase
Nov 08, 2017 9:29:28 AM org.apache.hadoop.hbase.zookeeper.
RecoverableZooKeeper checkZk
```

```
WARNING: **Unable to create ZooKeeper Connection**
```

排查思路：

■ 错误示例：“hbase.zookeeper.quorum”：“xxx-2,xxx-4,xxx-5,xxxx-3,xxx-6”

■ 正确示例：“hbase.zookeeper.quorum”：“您的zookeeperIP地址”

- **没有找到相应的文件。**

  经DataX智能分析，该任务最可能的错误原因如下所示：

  ```
  com.alibaba.datax.common.exception.DataXException: Code:[
  HdfsReader-08]
  ```

  错误解读：

  您尝试读取的文件目录为空。未能找到需要读取的文件，请确认您的配置项。

  ```
  path:  /user/hive/warehouse/tmp_test_map/*
  at com.alibaba.datax.common.exception.DataXException.asDataXExc
  eption(DataXException.java:26)
  ```

  排查思路：

  按照path找到相应的地方，检查对应的文件。如果没有找到文件，则对文件进行处理。

- **表不存在。**

  经DataX智能分析，该任务最可能的错误原因如下所示：

  ```
  com.alibaba.datax.common.exception.DataXException: Code:[
  MYSQLErrCode-04]
  ```

  错误解读：

  表不存在，请检查表名或者联系数据库管理员确认该表是否存在。

  表名为：xxxx。

  执行的SQL为`select * from xxxx where 1=2;`

  错误信息如下所示：

  ```
  Table ‘darkseer-test.xxxx’ doesn’t exist - com.mysql.jdbc.
  exceptions.jdbc4.MySQLSyntaxErrorException: Table ‘darkseer-test.
  xxxx’ doesn’t exist
  ```

  排查思路：

  `select * from xxxx where 1=2`判断表xxxx是否存在问题，如果有问题则要对表进行处理。

2.8.2 添加数据源典型问题场景

DataWorks添加数据源的典型问题可以分为连通性问题、参数问题和权限问题三类。

连通性问题

连通性问题主要体现为测试连通性失败。

- 如果您使用的是RDS数据源，建议您首先为#unique_111/unique_111_Connect_42_section_nwx_mm5_q2b。
- 如果您使用的是ECS上自建数据库，建议您首先为#unique_108/unique_108_Connect_42_section_vzl_bk5_q2b。
- 问题现象

  添加MySQL数据源时，网络类型选择为经典网络，单击测试连通性时失败报错：测试连接失败，测试数据源联通性失败，连接数据库失败，数据库连接串…异常消息：`Communications link failure. The last packet sent successfully to the server was 0 milliseconds ago.The dirver has not received any packets from the server`。

  解决方案

  出现上述报错通常都是网络连通性问题导致。建议检查您的网络是否可达、防火墙是否对该IP/端口有相关限制，以及安全组是否已配置对IP/端口放通。
- 问题现象：

  添加阿里云MongoDB数据源，测试数据源连通性失败，报错如下：

```
error message: Timed out after 5000 ms while waiting for a server
that matches ReadPreferenceServerSelector{readPreference=primary}.
```

```
Client view of cluster state is {type=UNKNOWN, servers=[..] error
with code: PROJECT_DATASOURCE_CONN_ERROR
```

问题解法

处理此类问题时，首先需要确定您的DataWorks工作空间所处地域。使用阿里云MongoDB，需要确定网络类型是否为VPC。VPC环境下MongoDB不支持数据连通性测试（使用方案一可以避免该问题）。

VPC环境下阿里云MongoDB数据同步有两种方案：

- 方案一：通过公网进行数据同步
  1. 数据源配置时，数据源类型选择连接串模式（数据集成网络可直接连通）。
  2. VPC环境下，您的MongoDB需要开通公网访问。
  3. 在MongoDB上放行相关白名单IP，详情请参见#unique_111。
  4. 进行数据连通性测试。
- 方案二：配置自定义资源组，从内网进行数据同步
  1. 准备一台和MongoDB同区域、同网络的ECS作为调度资源，详情请参见#unique_33。
  2. 将这台ECS的IP加入MongoDB的白名单或者安全组。
  3. 数据源测试连通时直接确定保存（不支持测试连通性）。
  4. 修改资源组为自定义调度资源，测试运行。

说明：

请务必添加相应的白名单。

- 问题现象

添加自建MongoDB数据源，测试数据源连通性失败。

问题解法

1. 数据源配置时，数据源类型选择连接串模式（数据集成网络可直接连通）。
2. 如果是VPC环境下ECS上自建的MongoDB，需要开通公网访问。
3. 确保网络和端口之间是否能连通，检查 ECS 的防火墙以及安全组设置
4. 确保自建的数据库涉及的安全访问限制，权限的限制和能否远程登录的情况。
5. 确认访问地址`host:port`填写正确，数据库名和用户名填写正确。

说明：

添加MongoDB数据源时，使用的用户名必须是用户需要同步的这张表所在的数据库创建的用户名，不能用root。

例如需要导入name表，name表在test库，则此处数据库名称填写为test。

用户名为指定数据库中创建的用户名，不要使用root。例如之前指定的是test库，则用户名需使用test数据库中创建的账户。

- 问题现象

  VPC环境下添加Redis数据源，测试数据源连通性失败，报错如下。

  问题解法

  Redis添加数据源时如果没有公网IP，需要保证数据源和DataWorks工作空间地域一致，通过新增调度资源完成数据源的打通。

- 问题现象

  添加MongoDB数据源，已经配置白名单，测试数据源连通性仍然失败，报错如下：

  ```
  error message: Timed out after 5000 ms while waiting for a server
  that matches ReadPreferenceServerSelector{readPreference=primary}
  ```

  问题解法

  VPC网络的MongoDB数据源和Dataworks的默认资源组在内网上是不通的，所以无法直接进行同步任务，需要通过公网或者自定义资源组的方式进行连通。

- 问题现象

  Docker中安装的MySQL如何添加到数据源?

  问题解法

  Docker中安装的MySQL直接用服务器的公网IP组成的JDBC地址是无法连接的，连通性测试无法通过。您需要将MySQl的端口映射到宿主机上，使用映射出的端口链接。

- 问题现象

  配置Redis数据源失败，测试数据源连通性失败报错如下：

  ```
  error message: java.net.SocketTimeoutException: connect timed out
  ```

  问题解法

  目前DataWorks不支持Redis通过内网添加数据源。建议您为Redis数据源开通公网访问能力。数据源配置时，选择连接串模式（数据集成网络可直接连通），通过公网连接。

- 问题现象

  新增阿里云RDS数据源时，测试连通性不通。

  问题解法

  1. 当RDS数据源测试连通性不通时，需要到自己的RDS上添加数据同步机器IP白名单，详情请参见#unique_111。

     
     说明:
     注意：若使用自定义资源组调度 RDS 的数据同步任务，必须把自定义资源组的机器 IP 也加到 RDS 的白名单中 。

  2. 确保添加的信息正确：RDS实例ID和RDS实例主帐号ID、用户名、密码数据库名必须确保正确。

- 问题现象

  新增自建ECS中的MySQL数据源时，数据源测试连通性不通。

  问题解法

  1. 确保网络和端口之间是否能连通，检查ECS的防火墙以及安全组设置。
  2. 确保自建的数据库涉及的安全访问限制，权限的限制和能否远程登录的情况。
  3. 确保添加的信息正确：用户名、密码、JDBC URL中的 IP 地址和端口必须确保正确。
  4. 在VPC的环境下购买的ECS，只能用脚本模式运行任务，在添加数据源时测试连通性不能成功。购买ECS后，您可以添加自定义资源，将同步任务下发到相应的资源组运行。

参数问题

- 问题现象:

  添加MySQL类型数据源时，单击测试连通性，报错如下:

  ```
  测试连接失败，测试数据源连通性失败，连接数据库失败…异常消息：No suitable
  direver found for...
  ```

  问题解法

  出现上述情况可能是JDBC URL格式填写错误导致，JDBC URL在填写时，请不要在URL中添加空格或任何特殊字符。正确格式为：`jdbc:mysql://ServerIP:Port/Database`。

- 问题现象

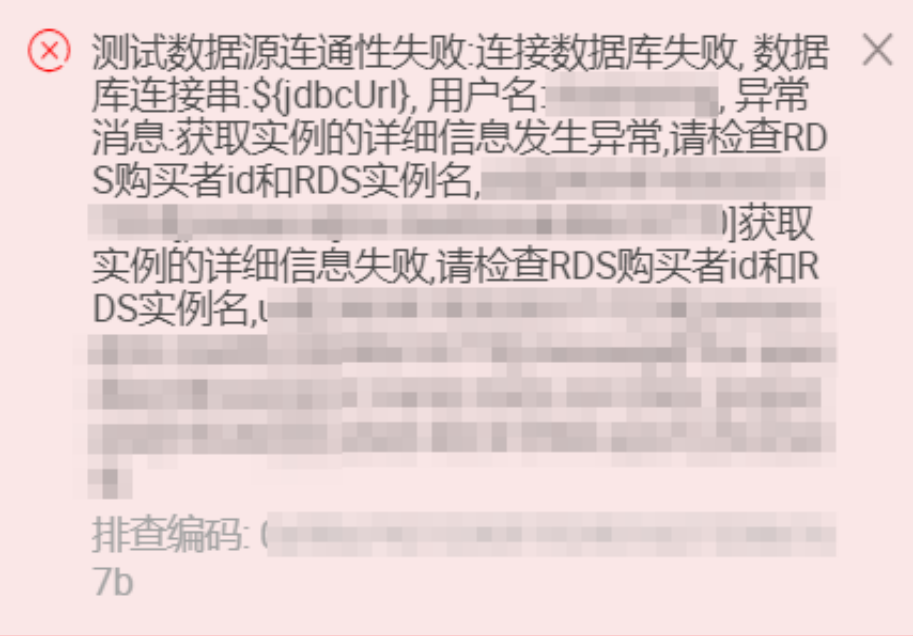
  使用用户名root添加MongoDB数据源时报错。

  问题解法

  添加MongoDB数据源时，使用的用户名必须是用户需要同步的这张表所在的数据库创建的用户名，不能用root。例如需要导入name表，name表在test库，则此处数据库名称填写为test。用户名为指定数据库中创建的用户名，不要使用root。例如之前指定的是test库，则用户名需使用test数据库中创建的账户。

- 问题现象

  添加RDS数据源失败，数据库连接不上，报错如下。

  问题解法

  需要检查填写的UID是否为是子账号的UID，此处要填写RDS所属主账号的UID才可以成功添加数据源。

- 问题现象

  加ODPS默认数据源时报测试连通性失败。

  问题解法

  ODPS默认数据源无需添加，默认为odps_fisrt。

- 问题现象

  DataWorks的数据源支持HybridDB for PostgreSQL吗?

  问题解法

  支持，添加时选择关系型数据库PostgreSQL即可。

- 问题现象

  没有外网地址的DRDS实例，配置数据源的时候,能否支持将实例的内网地址,映射为自定义的域名?

  问题解法

  需要严格按照格式来，目前不支持域名映射的方式。

- 问题现象

  添加RDS数据源时为什么白名单已添加，依然报错提示`user not exist ip white list reference`。

  问题解法

  出现这种情况通常是由于用户名输入错误。您可以参见创建账号和数据库检查自己输入的用户名是否正确。

权限问题

- 问题现象

  添加ADS数据源时，测试数据连通性报错如下:

  ```
  连接数据库失败，数据库连接串:${jdbcUrl}，用户名:XXXXXX，异常消息:You don
  't have privilege for connecting database 'dw', userId=RAM$XXX,
  schemaId=XX
  ```

  问题解法

  首先，您需要检查在数据源中填写的子账号是否有ADS的访问权限。分析型数据库用户基于阿里云帐号进行认证，用户建立的数据库属于该用户，用户也可以授权给其他用户访问其数据库下的表，所以连接的用户是需要在ADS上进行授权的，具体的说明参见用户账号类型与用户管理。

- 问题现象

  子帐户无权限查看数据源，无法创建数据源，提示您没有权限进行此操作。

  问题解法

  只有项目管理员权限的RAM子账户才可以增删改数据源。

2.8.3 同步任务等待槽位

问题描述

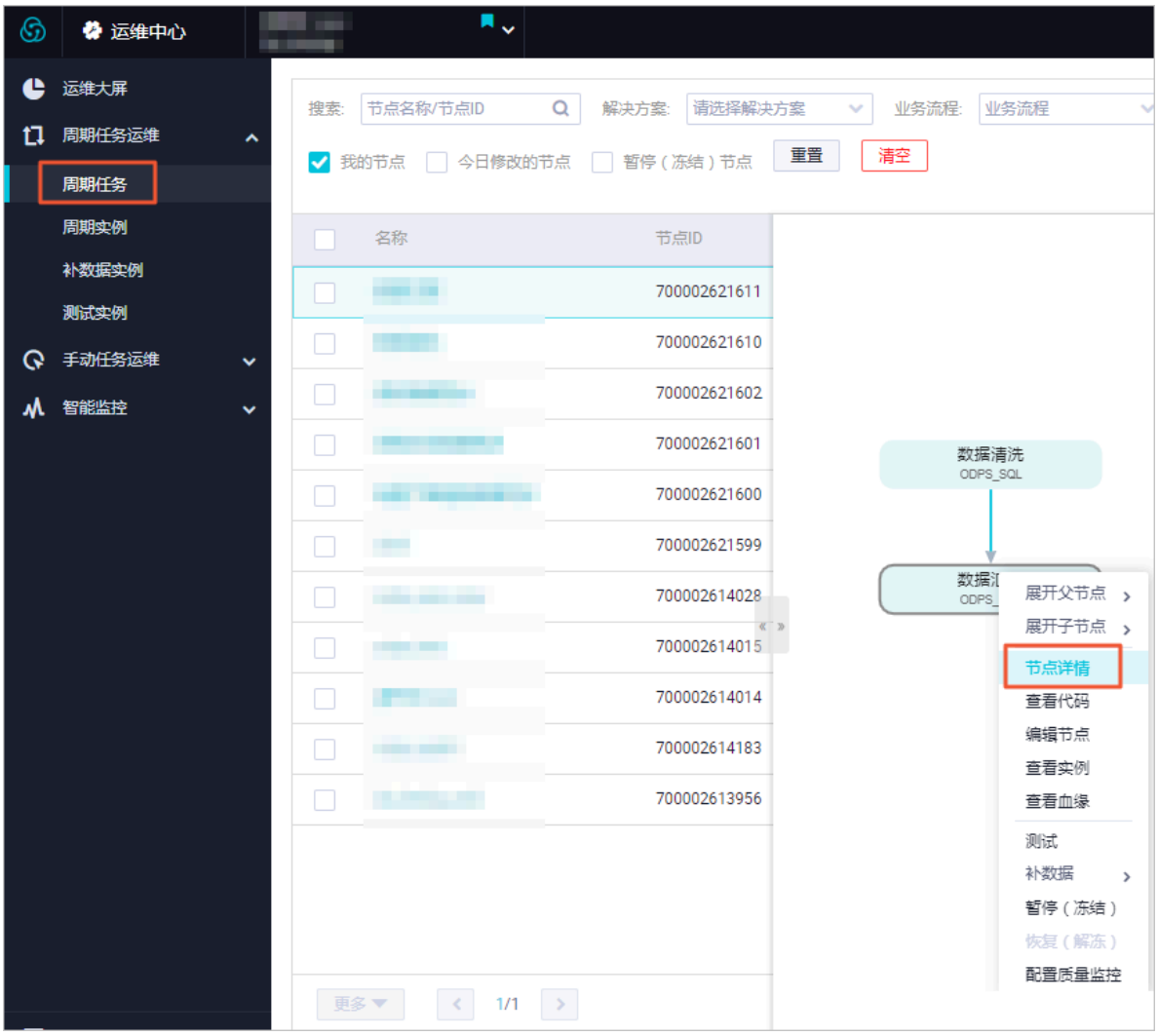
任务未正常运行，日志提示目前实例还没有产生日志信息，在等待槽位。

问题原因

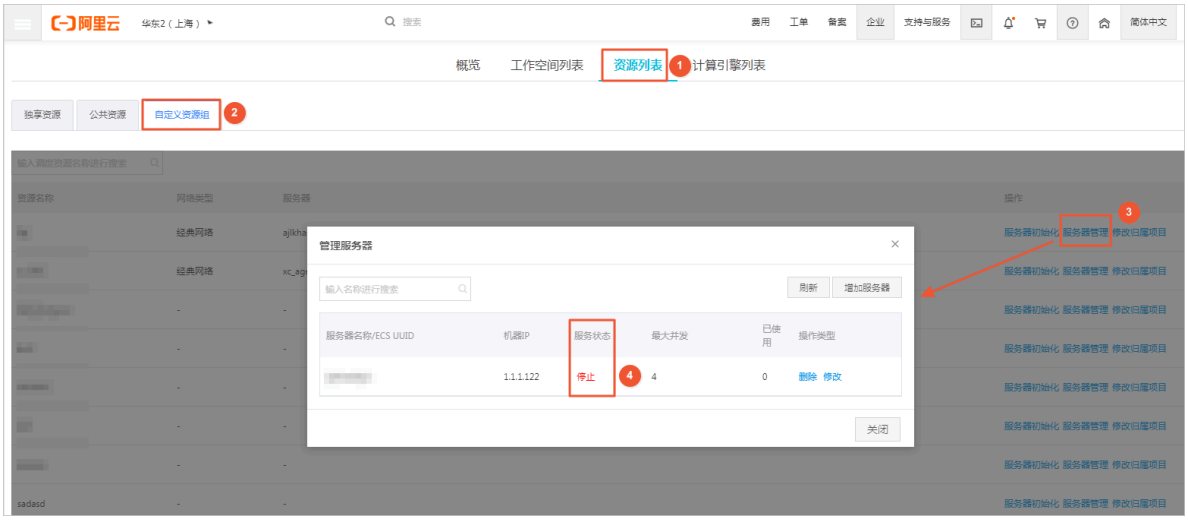
出现上述提示的原因是任务的配置调度使用的是自定义资源，但目前没有可用的自定义资源。

解决方法

1. 您可以进入DataWorks > 运维中心 > 周期任务运维 > 周期任务页面，右键单击DAG图中没有按照预期进行调度的任务，选择节点详情，查看任务使用的资源组。



2. 进入资源列表 > 自定义资源组页面，找到任务使用的调度资源，单击服务器管理，查看服务器的状态是否停止，或是否被其他任务占用。



3. 如果以上排查无法解决问题，可以执行下述命令重启服务。

```
su - admin
/home/admin/alisatasknode/target/alisatasknode/bin/serverctl restart
```

2.8.4 编码格式设置问题

数据集成的同步任务设置编码格式后，如果数据含有表情符，在运行任务时可能出现同步失败且产生脏数据或同步成功但数据乱码的问题。

同步失败且产生脏数据

问题描述

数据集成任务失败，且因编码问题产生脏数据，报错日志如下所示。

```
016-11-18 14:50:50.766 [13350975-0-0-writer] ERROR StdoutPlug
inCollector - 脏数据:<br>
{"exception":"Incorrect string value: '\\xF0\\x9F\\x98\\x82\\xE8\\
xA2...' for column 'introduction' at row 1","record":[{"byteSize":8,"
index":0,"rawData":9642,"type":"LONG"},
{"byteSize":33,"index":1,"rawData":"A公司出来的女汉子，扛得了箱子，招待好顾
客![1](http://docs-aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/pic/
56134/cn_zh/149872864****/%E5%9B%BE%E7%89%877.png)
被自己感动cry","type":"STRING"},
{"byteSize":8,"index":4,"rawData":0,"type":"LONG"}],"type":"writer"}
2016-11-18 14:50:51.265 [13350975-0-0-writer] WARN  CommonRdbmsWriter$
Task - 回滚此次写入，采用每次写入一行方式提交:java.sql.BatchUpdateException
: Incorrect string value: '\xF0\x9F\x88\xB6\xEF\xB8...' for column '
introduction' at row 1
```

问题原因

在对数据库做相应的编码格式设置或添加数据源时，未将编码设置为utf8mb4。只有utf8mb4编码支持同步表情符。

解决方法

- 添加JDBC格式的数据源时，需要修改utf8mb4的设置，例如jdbc:mysql://xxx.x.x.x:3306/database?com.mysql.jdbc.faultInjection.serverCharsetIndex=45。这样，在数据源设置表情符可以同步成功。
- 将数据源编码格式改成utf8mb4。例如在RDS控制台修改RDS的数据库编码格式。

> 说明:
> 如果需要设置RDS数据源编码格式set names utf8mb4，在添加数据源时必须使用无公网IP+连接串方式。

同步成功但数据乱码

问题描述

数据同步任务虽然成功，但数据乱码。

问题原因

发生乱码的原因有以下三种：

- 源端的数据本身就是乱码。
- 数据库和客户端的编码不一样。
- 浏览器编码不一样，导致预览失败或乱码。

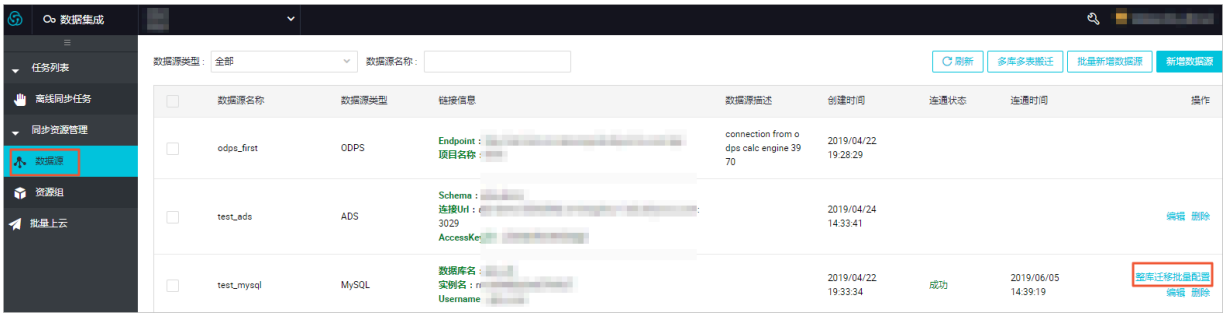
解决方法

您可以针对产生乱码的不同原因，选择相应的解决方法：

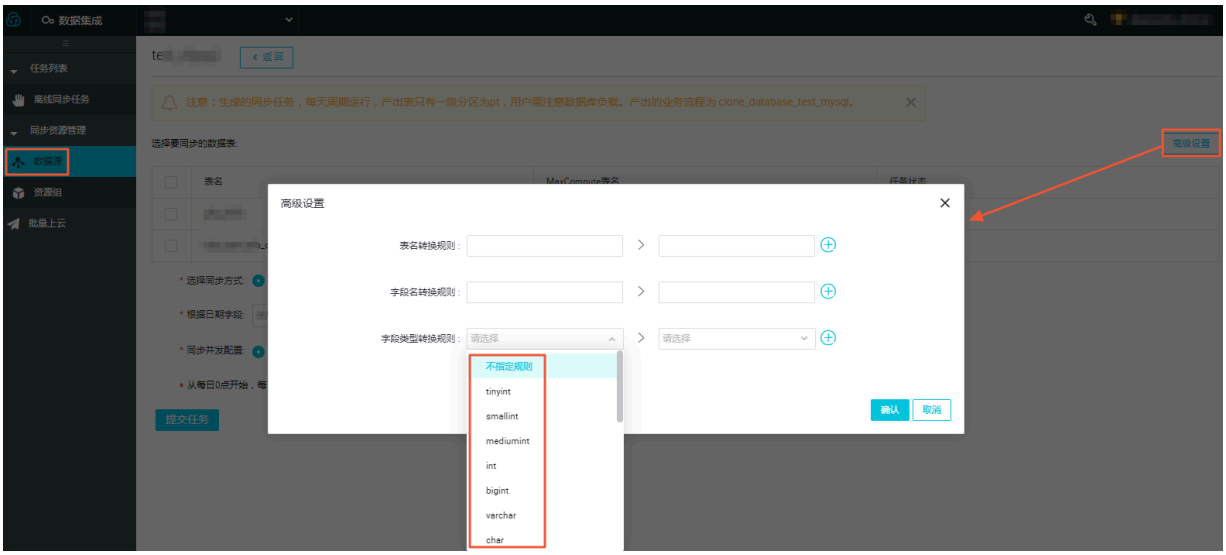
- 如果您的原始数据乱码，需首先处理好原始数据，再进行同步任务。
- 数据库和客户端编码格式不一致，需先修改编码格式。
- 浏览器编码和数据库或客户端编码格式不一致，需先统一编码格式，然后进行数据预览。

2.8.5 整库迁移数据类型

整库迁移目前仅支持MySQL（包括RDS中的MySQL）、Oracle数据源同步至MaxCompute，可以从已经添加好的MySQL/Oracle数据源中进入整库迁移页面。



下面仅针对整库迁移中高级设置的数据类型进行介绍。



整库迁移源端MySQL支持的数据源类型包括TINYINT、SMALLINT、MEDIUMINT、INT、BIGINT、VARCHAR、CHAR、TINYTEXT、TEXT、MEDIUMTEXT、LONGTEXT、YEAR、FLOAT、DOUBLE、DECIMAL、DATE、DATETIME、TIMESTAMP、TIME和BOOL。

目标端MaxCompute支持的数据源类型包括BIGINT、STRING、DOUBLE、DATETIME和BOOLEAN。

上述MySQL支持的数据类型均支持与MaxCompute数据源类型之间的转换。

> 说明：
> MySQL中的BIT，如果是bit(2)以上，则目前不支持与BIGINT、STRING、DOUBLE、DATETIME和BOOLEAN等类型转换。如果是bit(1)，则会被转换成BOOLEAN。

2.8.6 RDS同步失败转换成JDBC格式

问题描述

从RDS（MySQL/SQL Server/PostgreSQL）同步到自建MySQL/SQL Server/PostgreSQL时，报错为：DataX无法连接对应的数据库。

解决方法

以 RDS（MySQL）的数据同步到自建SQL Server为例，操作如下：

1. 新建一个数据源，将数据源配置为MySQL>JDBC 格式。
2. 使用新数据源配置同步任务，重新执行即可。

> 说明:
> 如果是在RDS（MySQL）>RDS（SQL Server）等云产品之间同步时，建议选择RDS（MySQL）>RDS（SQL Server）数据源来配置同步任务。

2.8.7 同步表列名是关键字任务失败

问题描述

用户做同步任务时，同步的表的列名是关键字，导致任务失败。

解决方法

以MySQL数据源为例。

1. 新建一张表aliyun，建表语句如下：

```
create table aliyun (`table` int ,msg varchar(10));
```

2. 创建视图，给table列取别名。

```
create view v_aliyun as  select `table` as col1,msg as col2 from
aliyun;
```

> 说明:
> - table是MySQL的关键字，在数据同步时，拼接出来的代码会报错。因此通过创建视图，给table列起别名，以绕过此限制。
> - 不建议使用关键字作为表的列名。

3. 上述语句给有关键字的列取了别名，那么在配置数据同步任务时，可以选择v_aliyun视图来代替aliyun这张表 。

> 说明:
> - MySQL的转义符是`关键字`。
> - Oracle和PostgreSQl的转义符是"关键字"。
> - SQlServer的转义符是[关键字]。

2.8.8 数据同步任务如何自定义表名

数据背景

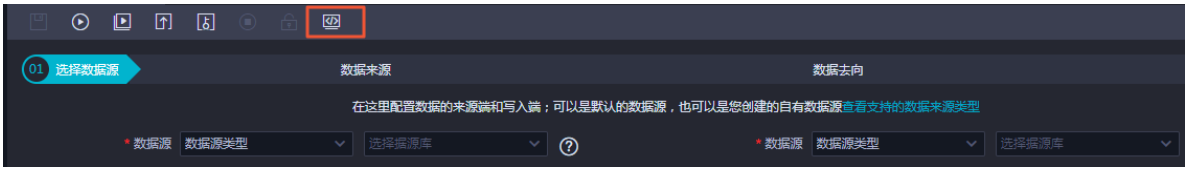
表是按天分的（如orders_20170310、orders_20170311和orders_20170312），每天一个表，表结构一致。

实现需求

创建数据同步任务，将表数据导入至MaxCompute中。希望只需要创建一个同步任务，自定义表名，实现每天凌晨自动从源数据库读取昨天的表数据（例如今天是2017年3月15日，自动从源数据库中读取orders_20170314的表的数据导入，以此类推）。

实现方式

1. 登录DataWorks控制台，单击对应工作空间操作栏中的进入数据开发。
2. 通过向导模式创建数据同步任务，配置时数据来源表先选一个表名如orders_20170310。详情请参见#unique_268。
3. 单击转换脚本按钮，将向导模式转换为脚本模式。



4. 在脚本模式中，改用来源表的表名为变量，如orders_${tablename}。

   在任务的参数配置中，给变量tablename赋值。由数据背景得知，表名是按天区分，而需求是每天读取昨天的表，所以赋值为$[yyyymmdd-1]。

   > **说明:**
   > 您也可以改用来源表的表名为变量时，直接写为orders_${bdp.system.bizdate}。

完成上述配置后，保存并提交，然后再进行后续操作。

2.8.9 使用用户名root添加MongoDB数据源报错

问题描述

使用用户名root添加MongoDB数据源时报错。

问题原因

添加MongoDB数据源时，使用的用户名必须是用户需要同步的这张表所在的数据库创建的用户名，不能用root。

解决方法

例如需要导入name表，name表在test库，则此处数据库名称填写为test。

用户名为指定数据库中创建的用户名，不要使用root。例如之前指定的是test库，则用户名需使用test数据库中创建的账户。

2.8.10 自定义资源组常见问题

本文将为您介绍自定义资源组在使用、配置文件、命令等方面的常见问题和解决方案。

应用场景

- 保证运行资源：由于集群共享默认资源组，会存在水位变高导致任务长时间等待的情况。如果您对任务有较高的资源使用需求，可以使用自定义资源组来自建任务运行集群。
- 连通网络：由于默认资源组无法连通VPC环境下的数据库，您可以使用自定义资源组进行网络连通。
- 用于调度资源组：调度槽位资源紧张的情况下，您可以使用自定义资源组。
- 提升并发能力： 默认资源组的运行槽位有限，您可以通过自定义资源组扩大槽位资源，允许更多的并发任务同时调度运行。

使用限制

- 一台ECS只能注册到一个自定义资源组下，一个自定义资源组可以添加多个ECS。
- 经典网络和专有网络注册的区别为：经典网络是主机名称，专有网络是UUID。
- 一个自定义资源中只允许存在一种网络类型。
- 不支持运行手动任务实例。
- ECS需要具备公网访问能力，ECS可以配置公网IP、EIP、NAT网关SNAT。

配置文件

通过DataWorks界面引导，完成自定义资源组的安装后，您可以登录ECS查看agent插件的下述信息。

- 默认安装路径：*/home/admin/*， 默认路径下通常会有以下目录信息。
  - alisatasknode：agent有关配置和命令所在目录。
  - datax和datax-on-flume：数据同步插件库和配置所在目录。

```
[root@iZwz9ef7rof3l2xye5tuwvZ ~]# cd /home/admin/
[root@iZwz9ef7rof3l2xye5tuwvZ admin]# ls
alisatasknode  datax3  datax-on-flume
```

- agent有关命令

  当前支持对agent进程进行stop/start/restart等命令操作，具体操作命令为`/home/admin/alisatasknode/target/alisatasknode/bin/serverctl start/stop/restart`

- 运行日志

  agent运行日志有以下两个存放路径：

  - */home/admin/alisatasknode/taskinfo/*：存放Shell脚本运行的日志信息，和DataWorks节点运行日志页面中查看的结果一致。
  - */home/admin/alisatasknode/logs*：alisatasknode.log日志文件中存放的是agent插件的运行信息，如接收到的任务运行/kill操作、agent心跳状态等。
  - */home/admin/datax3/log*：存放DataX任务的详细运行日志，遇到任务执行失败，可以查看该部分日志查找原因。

监控手段

您可以通过下述方法监控agent进程的运行状态，在监控到agent进程退出后，可以及时进行恢复。

1. root用户登录到ECS机器。
2. 执行命令`wget https://alisaproxy.shuju.aliyun.com/install_monitor.sh --no-check-certificate`。
3. 执行命令`sh install_monitor.sh`，监控日志默认存放在*/home/admin/alisatasknode/monitor/monitor.log*中。

DataWorks调度分类

自定义资源组在DataWorks调度体系中使用， 当前DataWorks调度体系分为一级调度资源和二级运行资源。

- 一级调度资源：进入运维中心 > 周期实例 > 属性页面查看一级调度资源，用来调度实例。
- 二级运行资源：进入数据开发 > 数据同步任务 > 任务资源组页面查看二级任务运行资源。

自定义资源组的使用

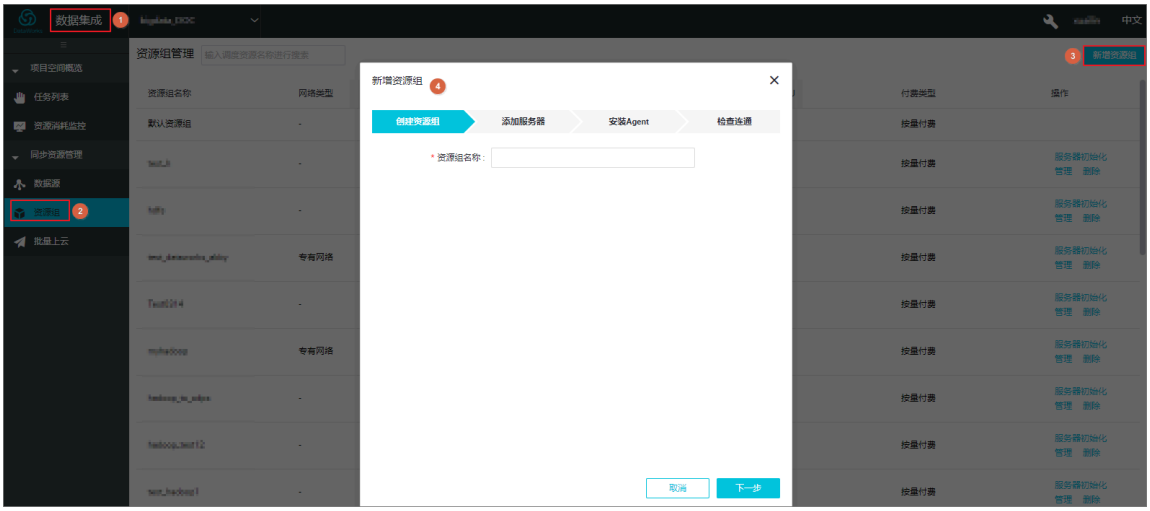
- 配置一级调度资源

  登录DataWorks控制台，进入调度资源列表页面，创建自定义资源组。

> **说明:**
> 在该页面创建的调度资源适用于Shell任务，配置的是一级调度资源。

- 配置二级运行资源

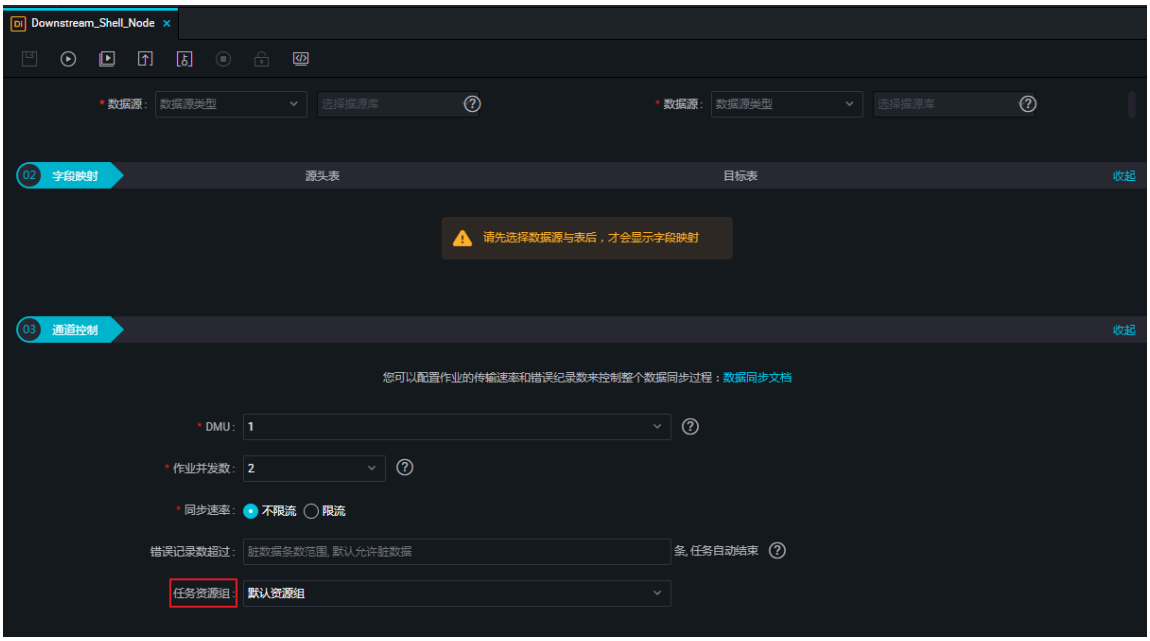
  1. 登录DataWorks控制台，进入数据集成 > 资源组页面新增自定义资源组。



> **说明:**

此页面添加的资源组仅用于数据同步任务，配置的是二级运行资源。

2. 新建完成后，能且只能在数据开发页面配置任务时，在通道控制 > 任务资源组配置中选择。



自定义资源组的常见问题

- 常见问题一：添加ECS资源组时，报错gateway already exists。

1. 报错提示对应ECS已经在gateway中注册过，因为一个ECS只能添加到一个自定义资源组下，所以需要您在调度资源列表和数据集成 > 资源组页面中查看是否存在同名或同UUID的自定义资源组。
2. 如果工作空间中没有发现对应的自定义资源组，则收集request ID信息联系阿里云技术支持进行排查解决。

- 常见问题二：添加自定义资源组后状态不可用。
  - 查看alisatasknode.log日志，确认是否有心跳上报302的情况。如果上报心跳302，则可以排查下述问题。
    - 查看UUID是否一致。对比自定义资源组页面的UUID信息和ECS上执行命令`dmidecode |grep UUID`的结果是否一致。

      说明:

此处的UUID区分大小写。

- 如果UUID不一致，需要填写正确的UUID并重新安装agent。

  说明:
  dmidecode命令在3.0.5版本及之前版本会是大写的方式显示UUID， 如果升级到3.1.2或以上版本， 则会小写显示UUID，此时会导致心跳异常，需要重新安装agent进行恢复。

- 确认config.properties中配置的用户名及密码是否和自定义资源组添加界面一致，如果不一致，请参见agent插件安装界面给出的命令重新安装agent。
- 如果UUID和密码信息正确，请查看config.properties中的字段node.uuid.enable，针对VPC类型的ECS，这个字段的值需要为true，如果VPC类型下node.uuid.enable的值为false，则修改为true，重启agent进程即可。

· 查看alisatasknode.log日志，确认是否存在connection timeout的信息，如果有，则可以进行如下处理。

1. 查看ECS是否有公网能力，如公网IP、EIP、NAT网关SNAT IP，可以执行`ping www.taobao.com`确认是否可以连通。
2. 如果ECS有公网能力，请查看ECS的安全组配置中内网出方向或公网出方向是否进行访问限制，如果有访问限制，需要对gateway的IP和端口放行。

```
        ... 34 more
2019-01-19 14:39:04,028 WARN  [main] [AlisaNodeRretrieval.java:160] [] - 一次执行获取初始化任务失败，Connect to alisa-cn-shenzhen.data.aliyun.com:80 [alisa-cn-shenzhen.data.aliyun.com/119.23.169.51, alisa-cn-shenzhen.data.aliyun.com/119.23.169.55, alisa-cn-shenzhen.data.aliyun.com/119.23.169.52, alisa-cn-shenzhen.data.aliyun.com/119.23.169.59, alisa-cn-shenzhen.data.aliyun.com/119.23.169.56, alisa-cn-shenzhen.data.aliyun.com/119.23.169.54, alisa-cn-shenzhen.data.aliyun.com/120.25.116.21, alisa-cn-shenzhen.data.aliyun.com/119.23.169.57, alisa-cn-shenzhen.data.aliyun.com/119.23.169.53, alisa-cn-shenzhen.data.aliyun.com/119.23.169.58] failed: connect timed out
org.apache.http.conn.ConnectTimeoutException: Connect to alisa-cn-shenzhen.data.aliyun.com:80 [alisa-cn-shenzhen.data.aliyun.com/119.23.169.51, alisa-cn-shenzhen.data.aliyun.com/119.23.169.55, alisa-cn-shenzhen.data.aliyun.com/119.23.169.52, alisa-cn-shenzhen.data.aliyun.com/119.23.169.59, alisa-cn-shenzhen.data.aliyun.com/119.23.169.56, alisa-cn-shenzhen.data.aliyun.com/119.23.169.54, alisa-cn-shenzhen.data.aliyun.com/120.25.116.21, alisa-cn-shenzhen.data.aliyun.com/119.23.169.57, alisa-cn-shenzhen.data.aliyun.com/119.23.169.53, alisa-cn-shenzhen.data.aliyun.com/119.23.169.58] failed: connect timed out
```

- 场景三：ECS状态正常，但Shell任务执行失败。
  - 任务执行异常，运行日志显示如下：

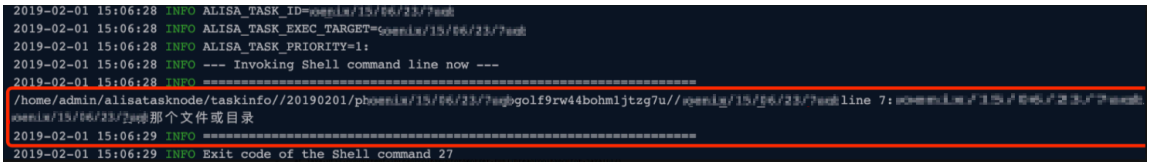
```
2019-01-17 19:32:36 INFO ALISA_TASK_EXEC_TARGET ALISA_TASK_EXEC_TARGET
2019-01-17 19:32:36 INFO ALISA_TASK_PRIORITY=1:
2019-01-17 19:32:36 INFO --- Invoking Shell command line now ---
2019-01-17 19:32:36 INFO
================================================================
The task process was abnormal exit, system set task rerun!!!
/home/admin/alisatasknode/taskinfo//20190117/phoenix/19/32/35/usa5vzt2fso5fcyu3q95fpn9/T3_0699121848.log-END-EOF
```

    您可以进行如下操作：

    - 查看alisatasknode.log日志中具体的报错信息，可根据T3_0699121848关键字进行搜索。
    - 登录ECS，切换到admin用户下，执行`python -V`命令查看python版本是否为2.7或2.6版本。

      说明：
      agent当前支持的是python2.7或2.6版本，通常python版本不对会导致replace user hive conf error的错误。

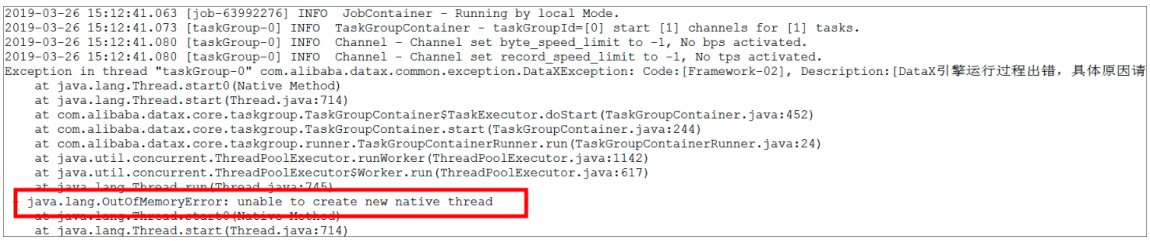
  - 查看DataWorks运行日志，找不到对应的文件。



    您可以登录ECS并切换到admin用户，执行命令sh -x脚本名，确认是否可以正常执行，根据报错信息进行调试。

- 常见错误四：自定义资源组下任务执行OOM。
  - 报错问题

    获取用户运行报错日志如下图所示，提示无法分配内存给作业线程。



- 排查思路

  自定义资源组创建时设置的内存数决定了资源组可提供的槽位能力。资源组内的系统进程和agent进程也会占用一部分内存，不能将ECS实例所有内存都用于槽位资源，会导致高并发下某些作业OOM。

- 解决方法

  建议您调小自定义资源组下的内存数设置，可以考虑预留2G的空间给系统和agent进程使用，如果还有其他进程 则需预留更多的内存。

3 数据开发

3.1 解决方案

数据开发模式全面升级，包括工作空间>解决方案>业务流程三级结构，抛弃陈旧的目录组织方式。

工作空间>解决方案>业务流程

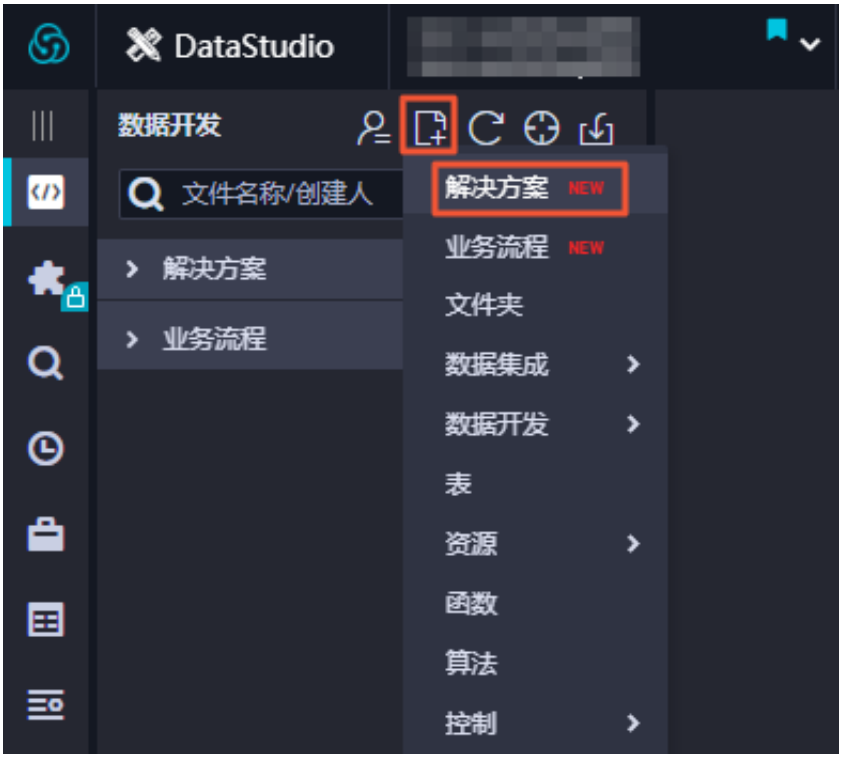
在DataWorks V2.0中，对数据开发模式进行全面升级，按照业务种类将相关的不同类型的节点任务组织在一起，这样的结构能够更好地以业务为单元进行代码的开发。在开发过程中，可以站在更高视角横跨多个业务流程进行开发。通过工作空间>解决方案>业务流程三级结构，全新定义用户的开发流程，提升开发体验。

- 工作空间：权限组织的基本单位，用来控制用户的开发、运维等权限。工作空间内成员的所有代码均可以协同开发管理。
- 解决方案：您可以自定义组合一些业务流程为一个解决方案。优势如下所示：
  - 包含多个业务流程。
  - 解决方案之间可复用相同的业务流程。
  - 自定义组合的解决方案，可以进行沉浸式开发。
- 业务流程：对业务的抽象实体，以赋能用户能够以业务的视角来组织数据代码开发。业务流程可以被多个解决方案复用。优势如下所示：
  - 帮助您从业务视角组织代码，更清晰。提供基于任务类型的代码组织方式。支持多级子目录（建议不超过四级）。
  - 可以从业务视角查看整体的工作流，并进行优化。
  - 提供业务流程看板，开发更高效。
  - 可以按照业务流程组织进行发布和运维。

沉浸式的开发体验

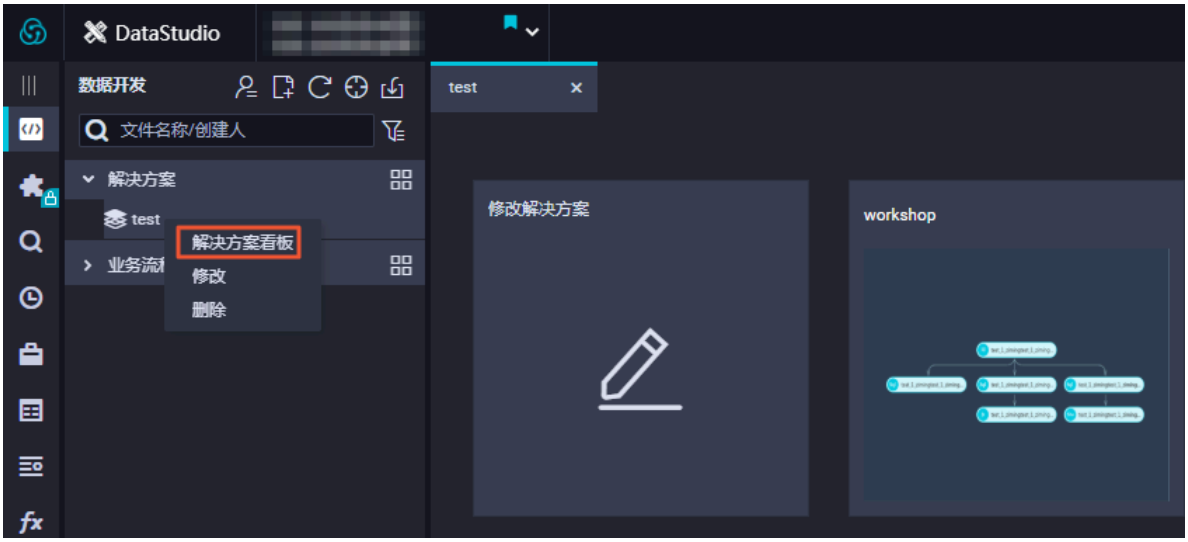
双击您的解决方案，开发区域切换进入到解决方案。目录仅显示当前解决方案的内容，为您提供更加清爽的环境，避免受到工作空间内其他不相关代码的影响。

1. 进入DataStudio（数据开发）页面，选择新建 > 解决方案。

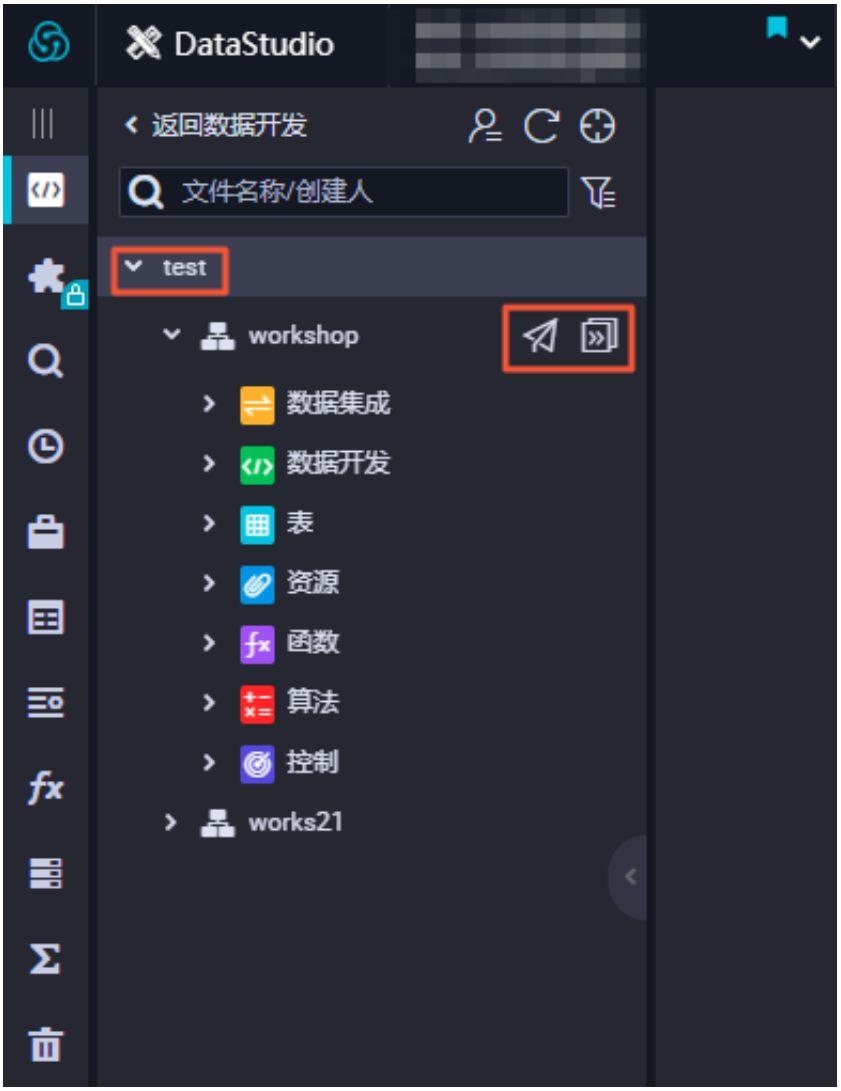


2. 在新建解决方案对话框中，填写解决方案名称、描述，并选择业务流程，单击新建。

3. 右键单击新建的解决方案名称，选择解决方案看板，即可看到选择的业务流程的节点，并可以修改解决解决方案。



4. 双击解决方案的名称，即可展示该解决方案下所有的业务流程，您可以进行编辑。详情请参见#unique_273。



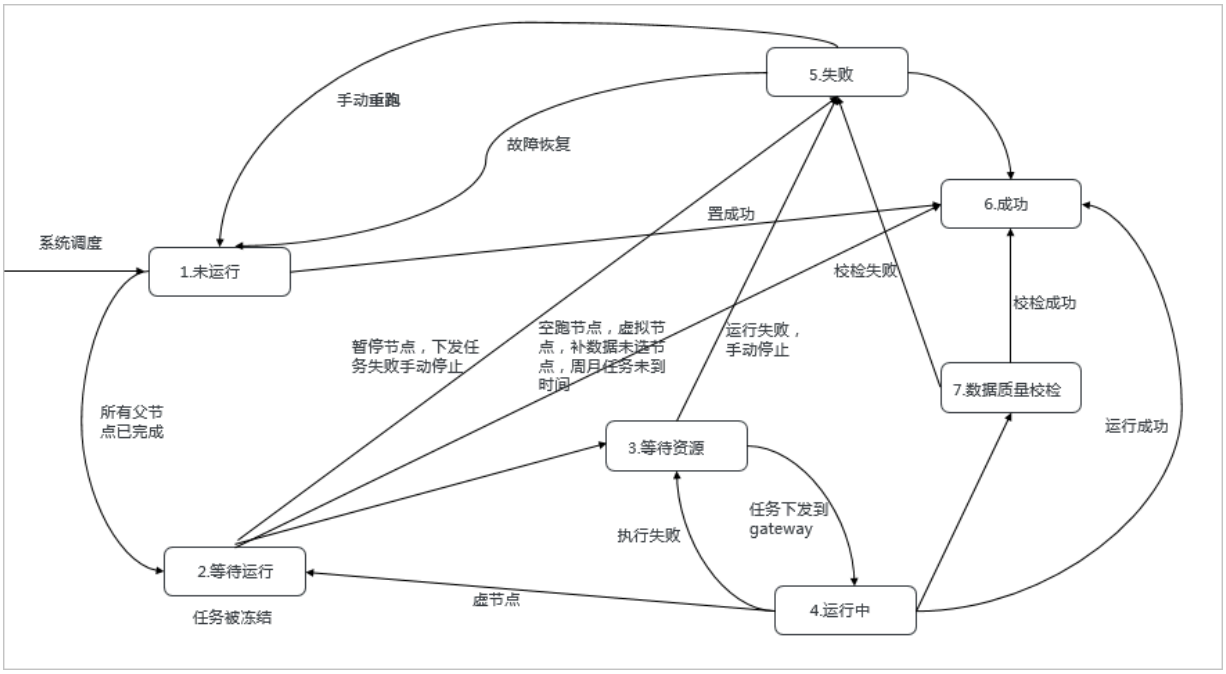
鼠标悬浮至解决方案名称上，会显示和两个按钮。

- 单击按钮，即可跳转至任务发布页面，并展示当前解决方案下待发布状态的节点。
- 单击按钮，即可跳转至运维中心 > 周期实例页面，默认展示当前解决方案下所有的节点的周期实例。

业务流程可以被多个解决方案复用，您只需要开发自己的解决方案。其他人可以在其他解决方案或者业务流程中直接编辑您引用的业务流程，构成协同开发。

任务状态机模型

任务状态机模型是针对数据任务节点在整个运行生命周期的状态定义，共有6种状态，状态之间的转换逻辑如下图所示。



3.2 SQL代码编码原则与规范

本文将为您介绍SQL编码的基本原则和详细的编码规范。

编码原则

SQL代码的编码原则如下：

- 代码功能完善。
- 代码行清晰、整齐，代码行的整体层次分明、结构化强，具有一定的可观赏性。
- 代码编写要充分考虑执行速度最优的原则。
- 代码中应有必要的注释以增强代码的可读性。
- 规范要求非强制性约束代码开发人员的代码编写行为。实际应用中，在不违反常规要求的前提下，允许存在可以理解的偏差。本规范不仅对日常的代码开发工作起到指导作用，而且不断进行完善和补充。
- SQL代码中应用到的所有关键字、保留字都使用小写，如select、from、where、and、or、union、insert、delete、group、having、count等。
- SQL代码中应用到的除关键字、保留字之外的代码，也都使用小写，如字段名、表别名等。
- 四个空格为一个缩进量，所有的缩进皆为一个缩进量的整数倍，按代码层次对齐。
- 禁止使用select * 操作，所有操作必须明确指定列名。
- 对应的括号要求在同一列的位置上。

SQL编码规范

SQL代码的编码规范如下：

- 代码头部

  代码头部添加主题、功能描述、作者和日期等信息，并预留修改日志及标题栏，以便后续添加修改记录。注意每一行不超过80个字符。模板如下：


```
-- MaxCompute(ODPS) SQL
--
********************************************************************************
-- ** 所属主题: 交易
-- ** 功能描述: 交易退款分析
-- ** 创建者 : 有码
-- ** 创建日期: 20170616
-- ** 修改日志:
-- ** 修改日期 修改人 修改内容
-- yyyymmdd name comment
-- 20170831 无码 增加对biz_type=1234交易的判断
--
********************************************************************************
```

- 字段排列要求

  - SELECT语句选择的字段按每行一个字段方式编排。
  - SELECT单字后面一个缩进量后直接跟首个选择的字段，即字段离首起二个缩进量。
  - 其它字段前导二个缩进量，再在逗号后放置字段名。
  - 两个字段之间的逗号分割符紧跟在第二个字段的前面。
  - AS语句应与相应的字段在同一行。多个字段的AS建议尽量对齐在同一列上。

```
select  channel_id              as channel_id
       ,trade_channel_desc      as trade_channel_desc
       ,trade_channel_edesc     as trade_channel_edesc
       ,inst_date               as inst_date
       ,trade_iswap             as trade_iswap
       ,channel_type            as channel_type
       ,channel_second_desc     as channel_second_desc
from   (
```

- INSERT子句排列要求

  INSERT子句写在同一行，请勿换行。

- SELECT子句排列要求

  SELECT语句中所用到的from、where、group by、having、order by、join和union等子句，需要遵循如下要求：

  - 换行编写。
  - 与相应的SELECT语句左对齐编排。
  - 子句后续的代码离子句首字母二个缩进量起编写。
  - where子句下的逻辑判断符and、or等与where左对齐编排。
  - 超过两个缩进量长度的子句加一空格后编写后续代码，如order by和group by等。

```
select      trim(channel) channel
           ,min(id)       id
from       ods_trd_trade_base_dd
where      channel is not null
and        dt = ${tmp_uuuummdd}
and        trim(channel) <> ''
group by   trim(channel)
order by   trim(channel)
```

- 运算符前后间隔要求

  算术运算符、逻辑运算符前后要保留一个空格，除非超过每行80个字符长度的限制，否则都写在同一行。

```
select      trim(channel) channel
           ,min(id)       id
from       ods_trd_trade_base_dd
where      channel is not null
and        dt = ${tmp_uuuummdd}
and        trim(channel) <> ''
group by   trim(channel)
order by   trim(channel)
```

- CASE语句的编写

  SELECT语句中对字段值进行判断取值的操作将用到的CASE语句，正确的编排CASE语句的写法对加强代码行的可阅读性也是很关键的一部分。

  对CASE语句编排作如下约定：

  - when子语在CASE语句的同一行并缩进一个缩进量后开始编写。
  - 每个WHEN子句一行编写，当然如果语句较长可换行编排。
  - CASE语句必须包含ELSE子语，ELSE子句与WHEN子句对齐。

```
,case   when p1.trade_from = '3008' and p1.trade_email is null then 2
        when p1.trade_from = '4000' and p1.trade_email is null then 1
        when p9.trade_from_id is not null then p9.trade_from_id
 end                                as trade_from_id
,p1.trade_email                     as partner_id
```

- 查询嵌套编写规范

  子查询嵌套在数据仓库系统ETL开发中是经常要用到，因此代码的分层编排就非常重要。示例如下。

```
select     p.channel
           ,rownumber()  order_id
from       (
               select   s1.channel
                        ,s1.id
               from     (
                            select  trim(channel)       as channel
                                    ,min(id)            as id
                            from    ods_trd_trade_base_dd
                            where   channel is not null
                            and     dt = ${tmp_yyyymmdd}
                            and     trim(channel) <> ''
                            group by trim(channel)
                        ) s1
               left outer join
                        dim_trade_channel s2
               on       s1.channel = s2.trade_channel_edesc
               where    s2.trade_channel_edesc is null
               order by id
           ) p
;
```

- 表别名定义约定
  - 所有的表都加上别名。因为一旦在SELECT语句中给操作表定义了别名，在整个语句中对此表的引用都必须惯以别名替代。考虑到编写代码的方便性，约定别名尽量简单、简洁，同时避免使用关键字。
  - 表别名采用简单字符命名，建议按a、b、c、d……的顺序进行命名。
  - 多层次的嵌套子查询别名之前要体现层次关系，SQL语句别名的命名，分层命名，从第一层次至第四层次，分别用P 、S、 U 、D表示，取意为Part、Segment、Unit和Detail。也可以用a、b、c、d来表示第一层次到第四层次。对于同一层次的多个子句，在字母后加1、2、3、4……区分。有需要的情况下对表别名加注释。

```
select     p.channel
           ,rownumber()  order_id
from       (
               select   s1.channel
                        ,s1.id
               from     (
                            select  trim(channel)       as channel
                                    ,min(id)            as id
                            from    ods_trd_trade_base_dd
                            where   channel is not null
                            and     dt = ${tmp_yyyymmdd}
                            and     trim(channel) <> ''
                            group by trim(channel)
                        ) s1
               left outer join
                        dim_trade_channel s2
               on       s1.channel = s2.trade_channel_edesc
               where    s2.trade_channel_edesc is null
               order by id
           ) p
;
```

- SQL注释

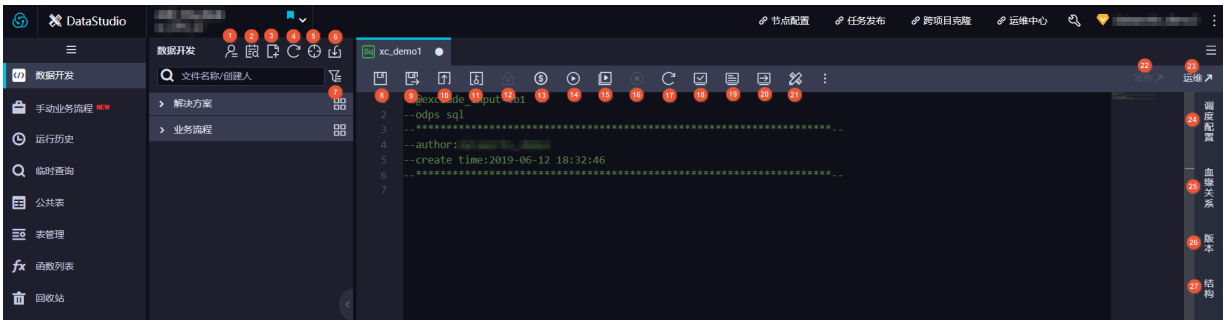
```
-- ********************************************************************************
-- STEP1:清理当天的数据分区 tmp_dws_tbd_alijr_user_relation_dd_5
-- ********************************************************************************
```

  - 每条SQL语句均应添加注释说明。
  - 每条SQL语句的注释单独成行、放在语句前面。
  - 字段注释紧跟在字段后面。
  - 对不易理解的分支条件表达式加注释。
  - 对重要的计算应说明其功能。
  - 过长的函数实现，应将其语句按实现的功能分段加以概括性说明。
  - 常量及变量注释时，应注释被保存值的含义（必须），合法取值的范围（可选）。

3.3 界面功能

3.3.1 界面功能点介绍

本文将为您介绍DataWorks数据开发（DataStudio）界面各按钮的功能。



界面功能点说明如下：

序号	功能	说明
1	我的文件	查看当前工作空间自己名下的节点。
2	代码搜索	搜索某个或者某段代码。
3	新建【+】	新建解决方案、业务流程、文件夹、节点、表、资源、函数的入口。
4	刷新	刷新当前目录树。
5	定位	定位选中的文件位置。
6	导入	导入本地数据到线上某张表中，注意选择编码格式。 说明： 标准模式下，此处导入的是开发环境下的表。

序号	功能	说明
7	筛选	按照条件筛选需要查询的节点。
8	保存	保存当前代码。
9	另存为临时查询文件	将当前代码另存为一个临时文件，可以在临时查询界面进行查看。
10	提交	提交当前节点。
11	提交并解锁	提交当前节点并解锁节点，对代码进行编辑。
12	偷锁编辑	非节点责任人编辑节点。
13	成本估计	单击该按钮后，选择业务日期，可以对运行的任务进行成本估计。  说明: 按量付费用户每次运行都会产生相应费用，请谨慎进行。小于1分钱按1分钱估算，实际以账单为准。
14	运行	运行当前节点代码，您只需赋值一次，即使节点的代码发生变更，也会保留初始的赋值。
15	高级运行（带参数运行）	使用配置的参数运行当前节点代码。每次都需要手动给SQL中的变量进行赋值，运行的初始赋值会传递给高级运行，高级运行的自定义参数赋值后，会刷新运行的自定义参数。 例如运行设置的时间是4月2日，则之后每次运行都会是4月2日。然后单击高级运行，设置运行时间为4月3日，运行一次之后，下次再单击运行，时间便会变为4月3日。
16	停止运行	停止正在运行的代码。
17	重新加载	刷新页面，返回至上次保存的页面。
18	在开发环境执行冒烟测试	在开发环境测试当前节点的代码。开发环境冒烟测试可以模拟右侧的调度参数，选择业务日期后，根据您填写的调度参数替换该业务日期下的值。您可以通过该功能测试调度参数的替换情况。  说明: 开发环境冒烟测试每次变更调度属性后，其中的参数配置需要重新保存并提交，然后选择开发环境冒烟测试，否则替换的调度属性还是原来的值。
19	查看开发环境的冒烟测试日志	查看运行在开发环境的节点运行日志。

序号	功能	说明
20	前往开发环境的调度系统	前往开发环境的运维中心。
21	格式化	对当前节点代码排序，常用于单行代码过长的情况。
22	发布	提交后的代码可执行发布，发布后处于生产环境。
23	运维	前往生产环境的运维中心。
24	调度配置	配置节点的调度属性、参数、资源组等相关信息。
25	血缘关系	代码对其他表的血缘使用关系。
26	版本	当前节点提交发布记录。
27	结构	当前节点代码的结构。当代码过长时，可以通过结构中的关键信息快速定位代码段。

3.3.2 版本

版本是指当前节点的提交、发布记录。每提交一次节点，都会生成一个新的版本。

您可以根据自身需求，查看相关状态、变更类型、发布备注等信息，以便后续对节点进行操作。

说明:

只有提交过的节点才会存在版本信息。

信息	说明
文件ID	当前节点的节点ID。
版本	每次发布都会生成一个新的版本，第一次新增为V1，第二次修改为V2，以此类推。
提交人	提交发布节点的操作人。
提交时间	版本发布的时间。如果此版本先提交，之后再发布，发布时间会刷新提交时间，默认记录最后一次操作的发布时间。

信息	说明
变更类型	当前节点的操作历史。首次的提交记录为新增，之后对节点进行修改的提交记录为修改。
状态	当前节点的操作状态记录。
备注	在提交时，对该节点的变更描述。以便其他人操作该节点时，找到相应的版本。
操作	操作分为代码和回滚两部分。 · 代码：查看此版本的代码，精确查找想要回滚的记录版本。 · 回滚：将当前节点回滚到之前某个需要的版本，回滚后需要重新提交发布。
比较	将两个版本的代码和参数进行比较。  单击查看详情，可以进入详情页，对比代码、调度属性的变更情况。 **说明：** 比较只可以在两个版本中进行，无法选中一个或三个以上（包括三个）节点进行比较。

3.3.3 结构

结构是根据当前代码，解析出SQL下发运行的流程结构图，帮助用户快速梳理编辑的SQL情况，方便修改查看。

结构

如下SQL所示：

```
INSERT OVERWRITE TABLE dw_user_info_all_d PARTITION (dt='${bdp.system.
bizdate}')
SELECT COALESCE(a.uid, b.uid) AS uid
   , b.gender
   , b.age_range
   , b.zodiac
   , a.region
```

```
  , a.device
  , a.identity
  , a.method
  , a.url
  , a.referer
  , a.time
FROM (
  SELECT *
  FROM ods_log_info_d
  WHERE dt = ${bdp.system.bizdate}
) a
LEFT OUTER JOIN (
  SELECT *
  FROM ods_user_info_d
  WHERE dt = ${bdp.system.bizdate}
) b
ON a.uid = b.uid;
```

根据这段代码，解析出结构：

当鼠标放在圆圈中，会出现对应的解释：

1. **源表：select查询的目标表。**
2. **筛选：筛选表中要查询的具体分区。**

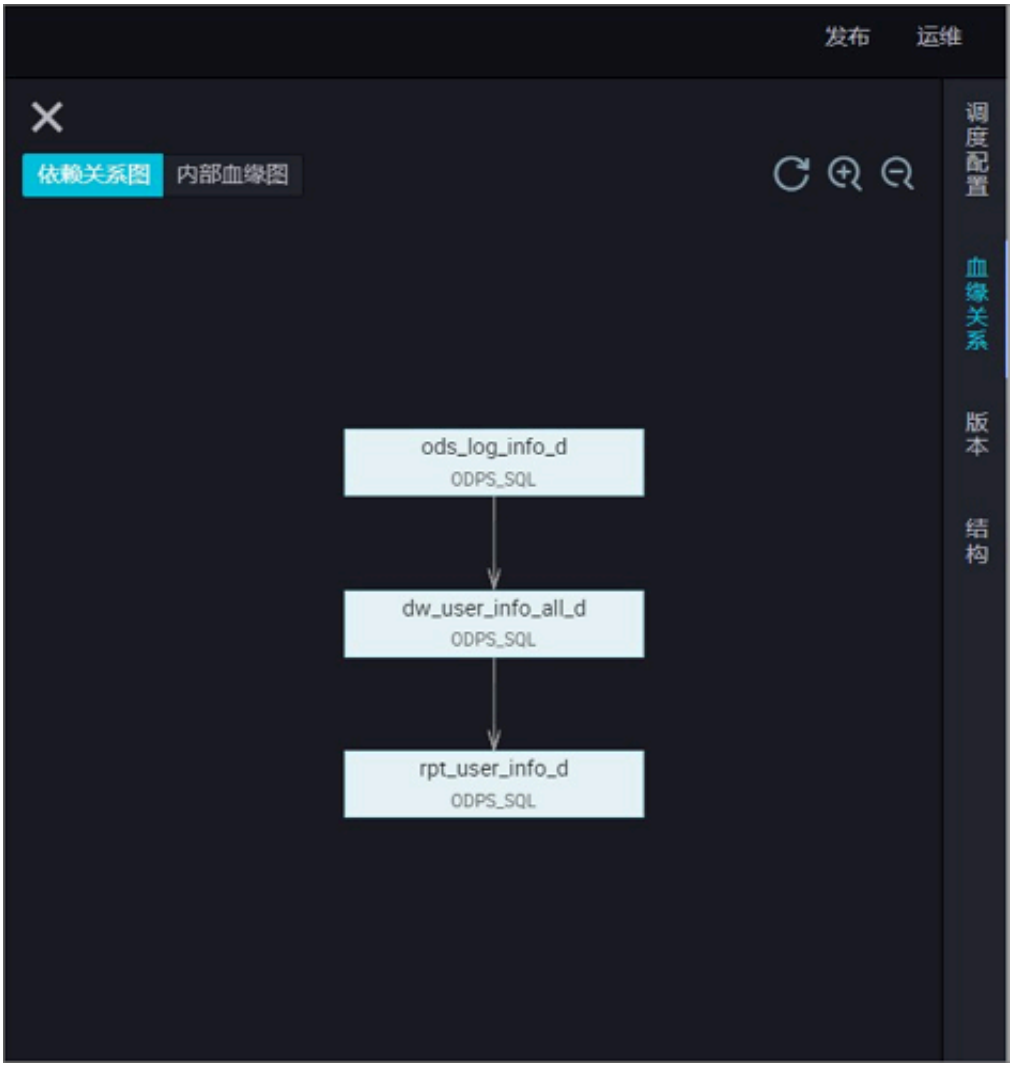
3. 第一部分中间表（查询视图）：将查询数据的结果放入一张临时表。
4. 关联（join）：将两部分查询结果通过join拼接。
5. 第二部分中间表（查询视图）：将join的结果汇总到一张临时表中，这张临时表存在三天，三天后自动清除。
6. 目标表（插入）：将第二部分得到的数据插入到insert overwrite的表中。

3.3.4 血缘关系

血缘关系展示当前节点和其他节点的关系。此关系会展示依赖关系图和内部血缘图两部分。

依赖关系图

依赖关系图根据节点的依赖关系，展示当前节点的依赖是否为自己预期的情况。如果不是，可以返回调度配置界面重新设置。

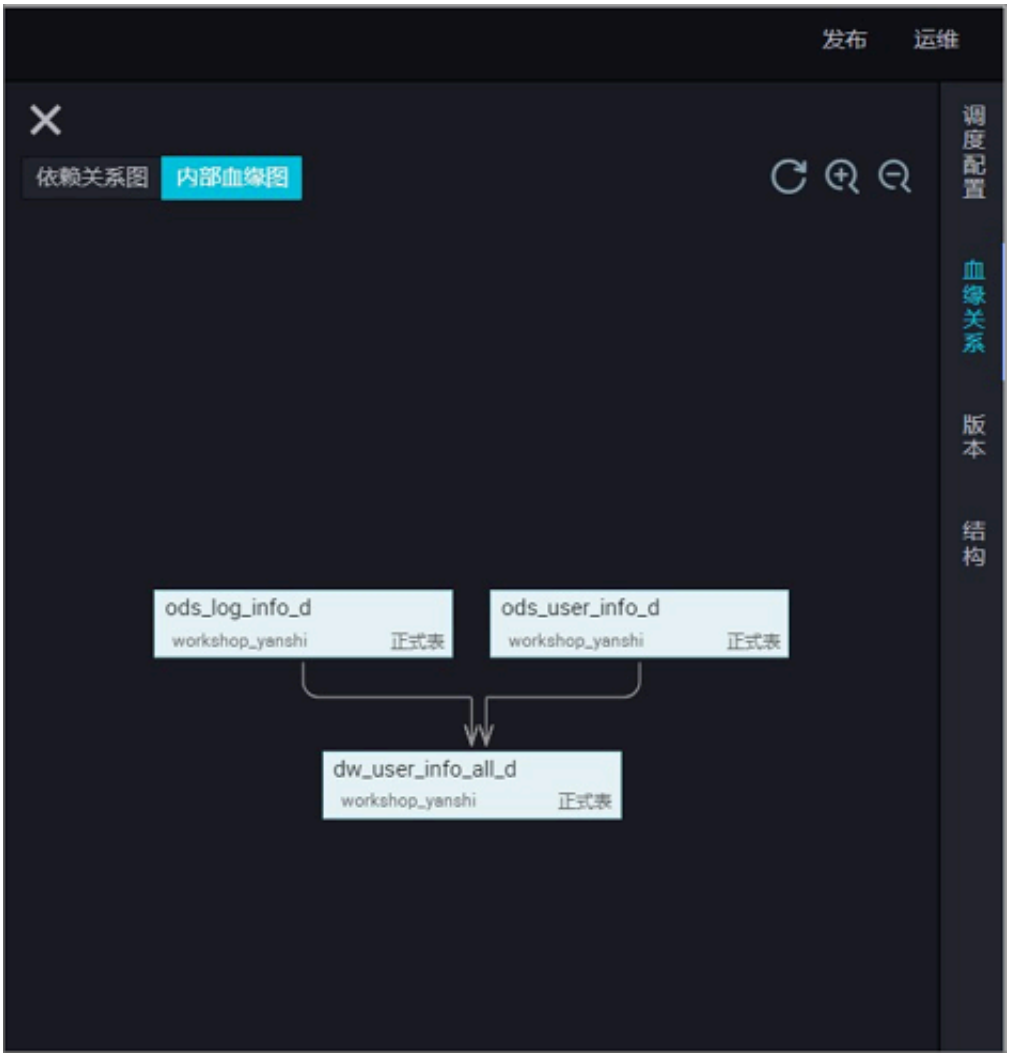


内部血缘图

内部血缘图根据节点的代码进行解析，如下所示：

```
INSERT OVERWRITE TABLE dw_user_info_all_d PARTITION (dt='${bdp.system.
bizdate}')
SELECT COALESCE(a.uid, b.uid) AS uid
  , b.gender
  , b.age_range
  , b.zodiac
  , a.region
  , a.device
  , a.identity
  , a.method
  , a.url
  , a.referer
  , a.time
FROM (
  SELECT *
  FROM ods_log_info_d
  WHERE dt = ${bdp.system.bizdate}
) a
LEFT OUTER JOIN (
  SELECT *
  FROM ods_user_info_d
  WHERE dt = ${bdp.system.bizdate}
) b
ON a.uid = b.uid;
```

根据上述SQL语句，解析出如下内部血缘图。将dw_user_info_all_d将作为join拼接ods_log_info_d的输出表解析，展示表之间的血缘关系。



3.4 业务流程

3.4.1 业务流程介绍

按照业务种类将相关的不同类型的节点任务组织在一起，即构成业务流程，能够更好地以业务为单元进行代码的开发。业务流程对应DataWorks V1.0中工作流的概念。

以业务流程为中心组织数据开发，通过各种类型开发节点的容器看板，将相关的工具和优化/管理操作围绕数据看板中的对象来组织，使得开发的管理更加方便和智能化。

DataWorks的代码结构

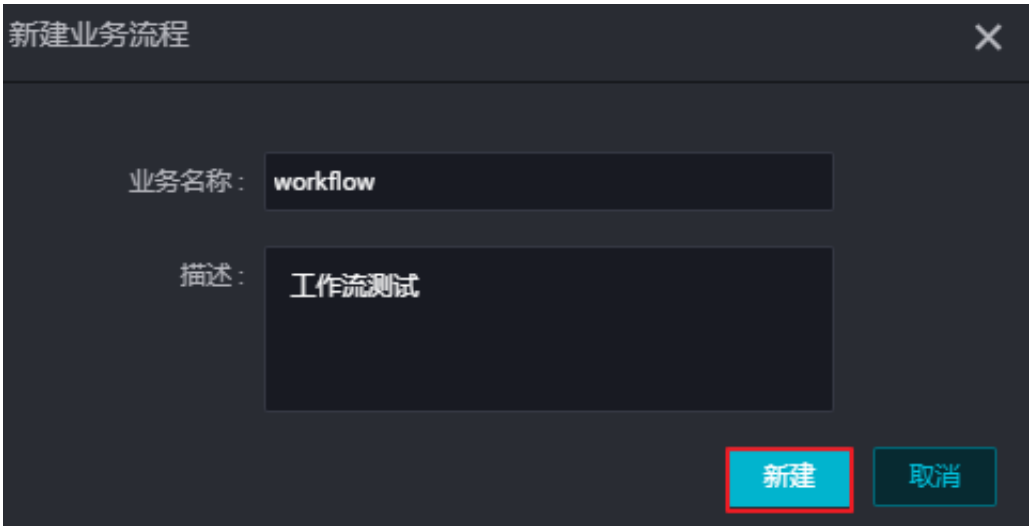
一个工作项目空间可以支持多种类型的计算引擎。一个工作项目空间中可以包含多个业务流程。一个业务流程是一套有机关联的各种类型的对象的集合，系统支持以自动生成的流程图的直观视角来查看该业务流程。流程中的对象类型有数据集成任务、数据开发任务、表、资源、函数、算法和操作流等多种类型。

每种对象类型对应一个独立的文件夹，在每个对象类型文件夹下， 支持继续创建子文件夹，为了便于管理，建议子文件夹的层数不要超过4层。 如果超过4层，可能规划的业务流程结构过于复杂，建议将该业务流程拆分成两个或多个业务流程，并将这几个相关的业务流程收纳到一个解决方案中进行管理，这样的代码组织方式会更加高效。

新建业务流程

1. 单击左上角的图标，选择全部产品 > DataStudio（数据开发）。
2. 右键单击业务流程，选择新建业务流程。

3. 在新建业务流程对话框中，填写业务流程名称和描述。



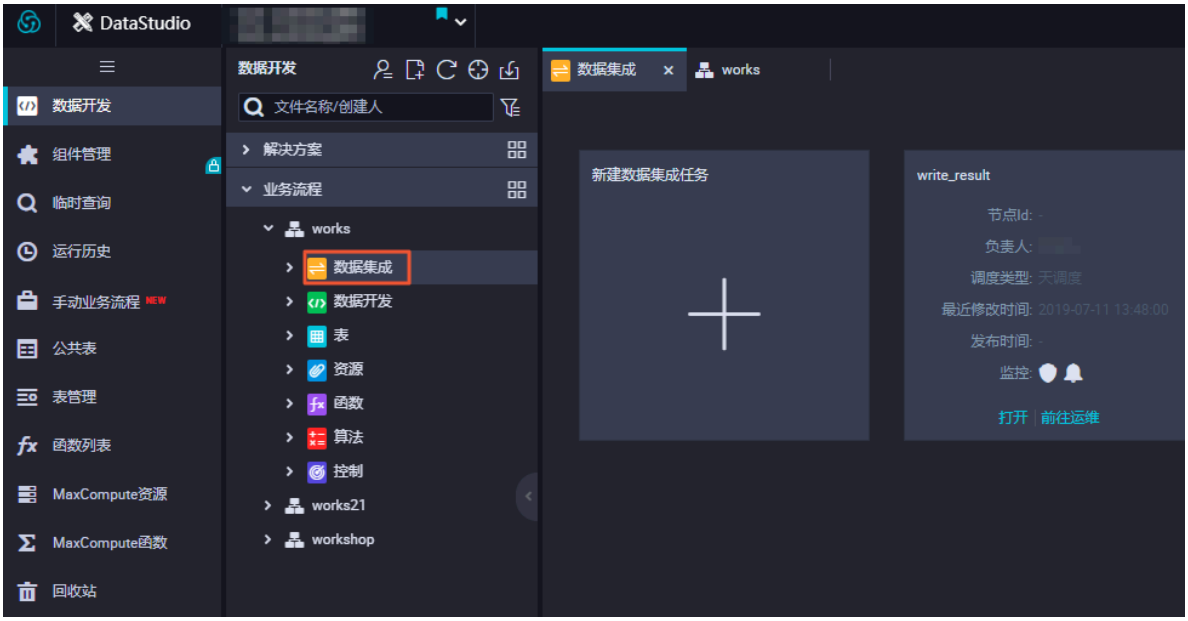
4. 单击新建，即可完成业务流程的创建。

业务流程组成

业务流程由以下各模块的节点组成。

- 数据集成

  双击相应业务流程下的数据集成，即可查看所有的数据集成任务，详情请参见#unique_282。

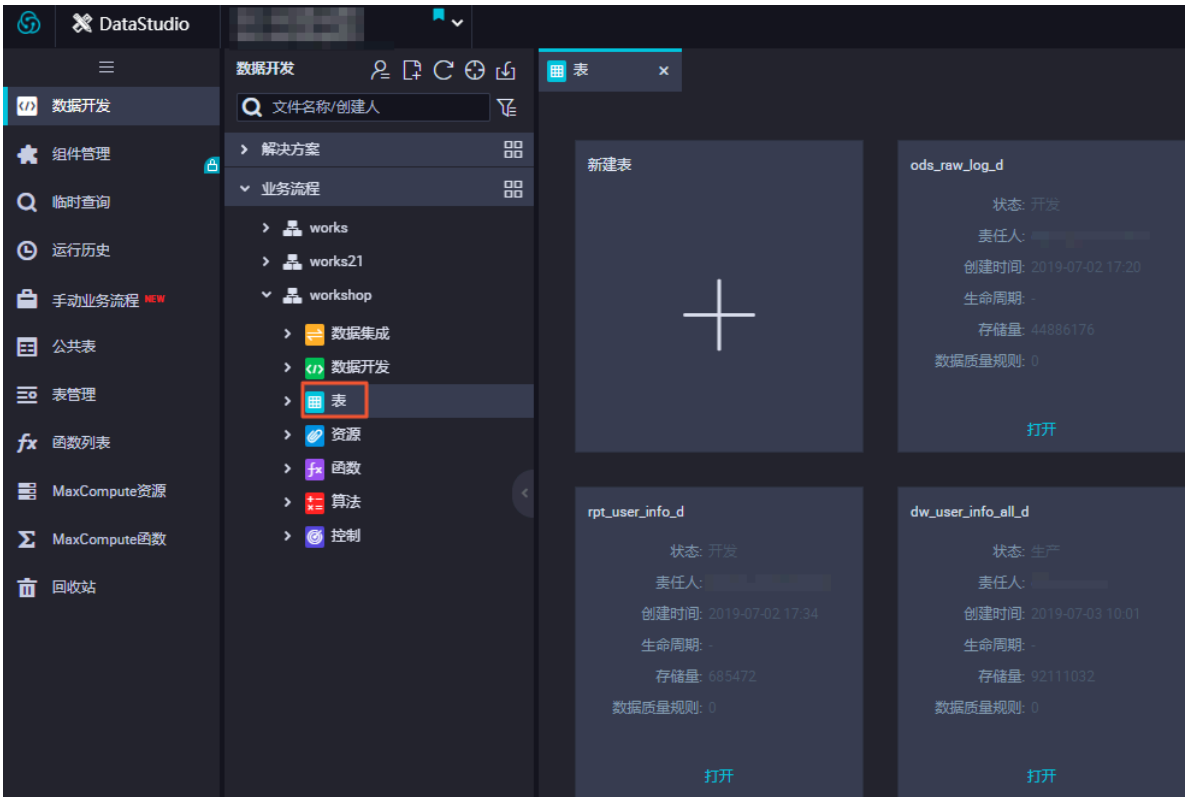


- 数据开发

  双击相应业务流程下的数据开发，即可查看所有的数据开发任务，详情请参见#unique_283。

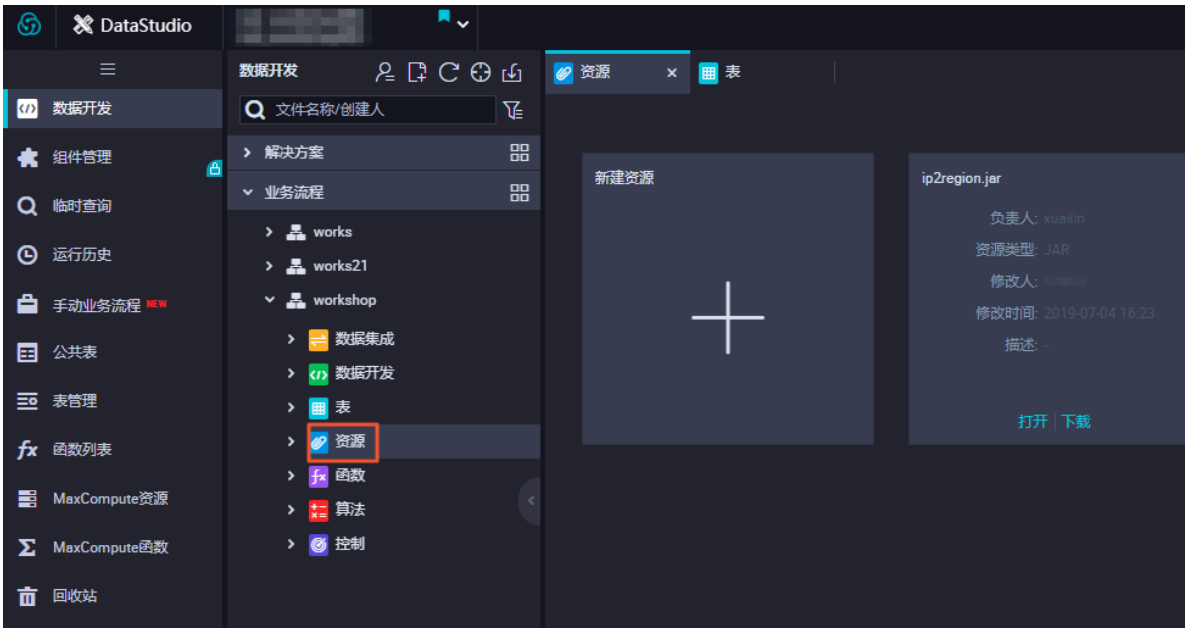
- 表

  双击相应业务流程下的表，即可查看所有创建的表，详情请参见#unique_284。



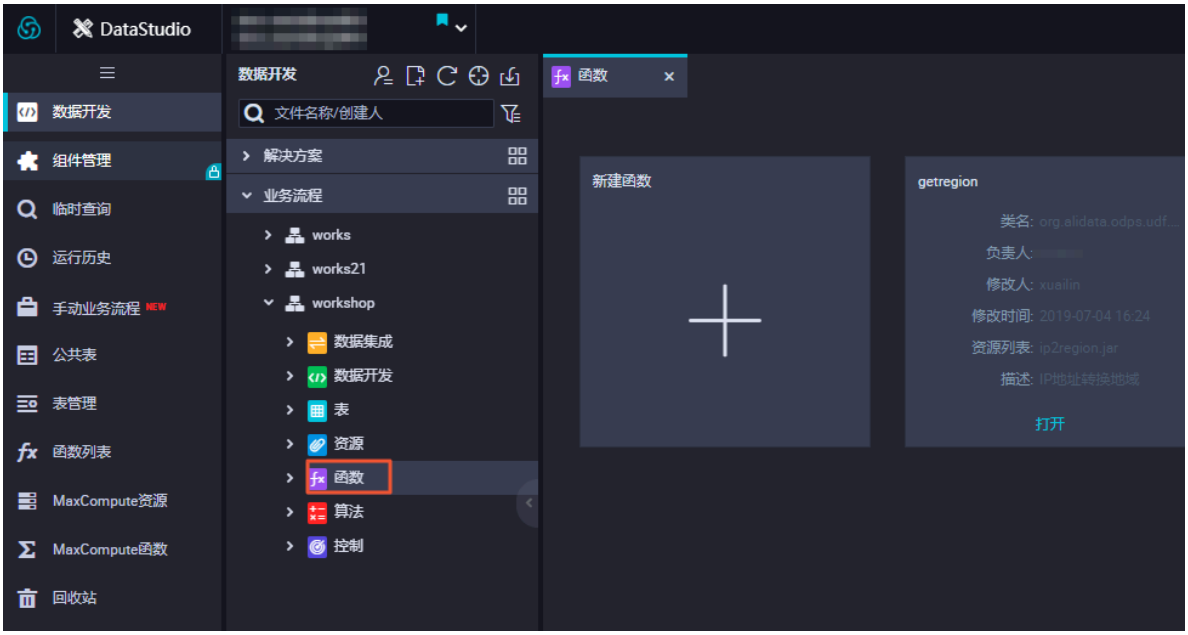
- 资源

  双击相应业务流程下的资源，即可查看所有创建的资源，详情请参见资源介绍。



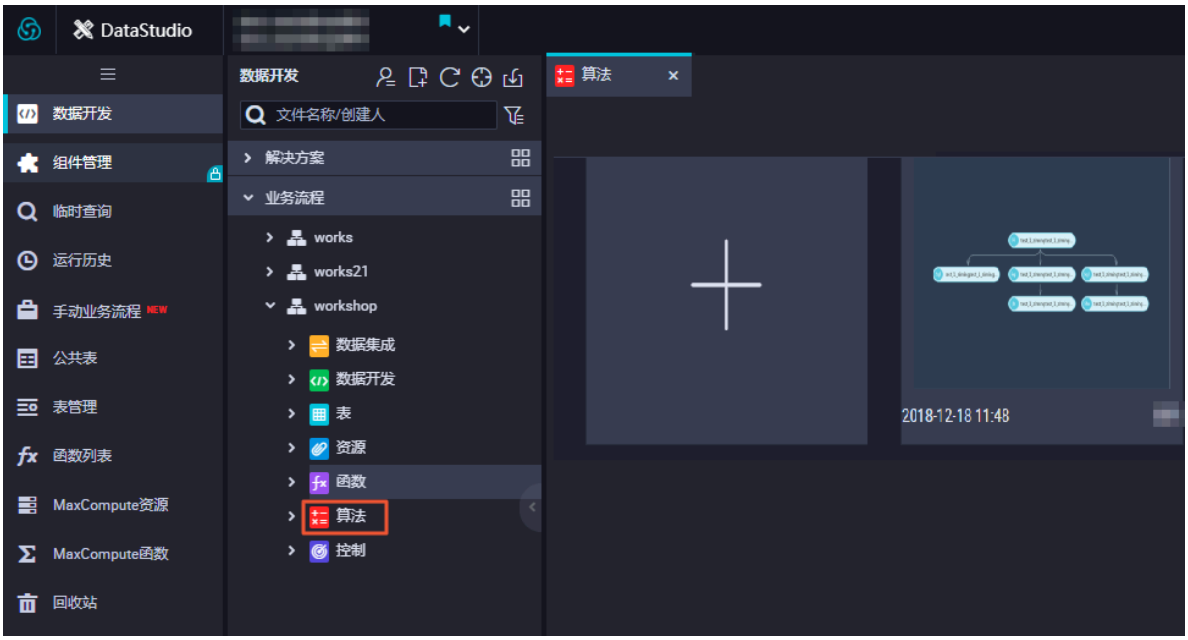
- 函数

  双击相应业务流程下的函数，即可查看所有创建的函数，详情请参见函数介绍。



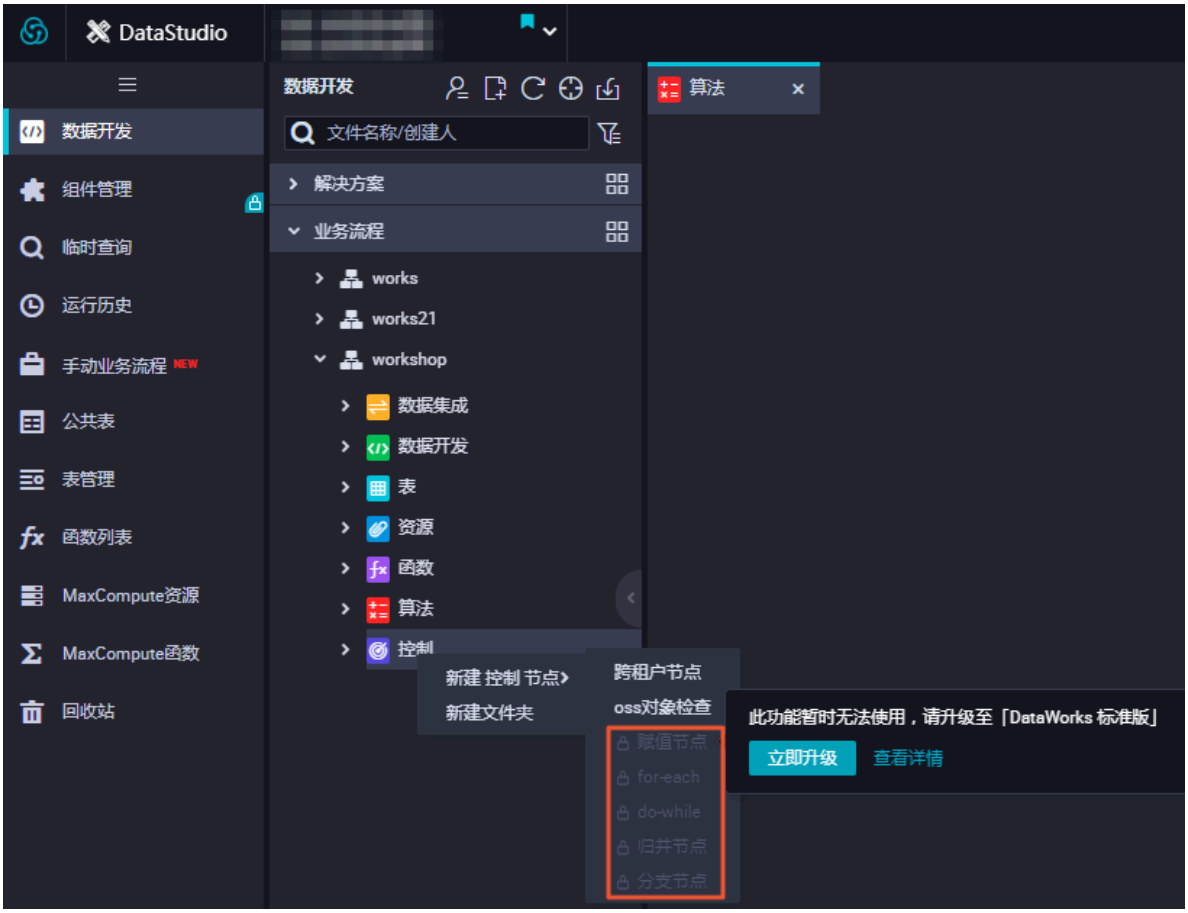
- 算法

  您可以新建机器学习（PAI）节点，双击相应业务流程下的算法，即可查看所有创建的算法。



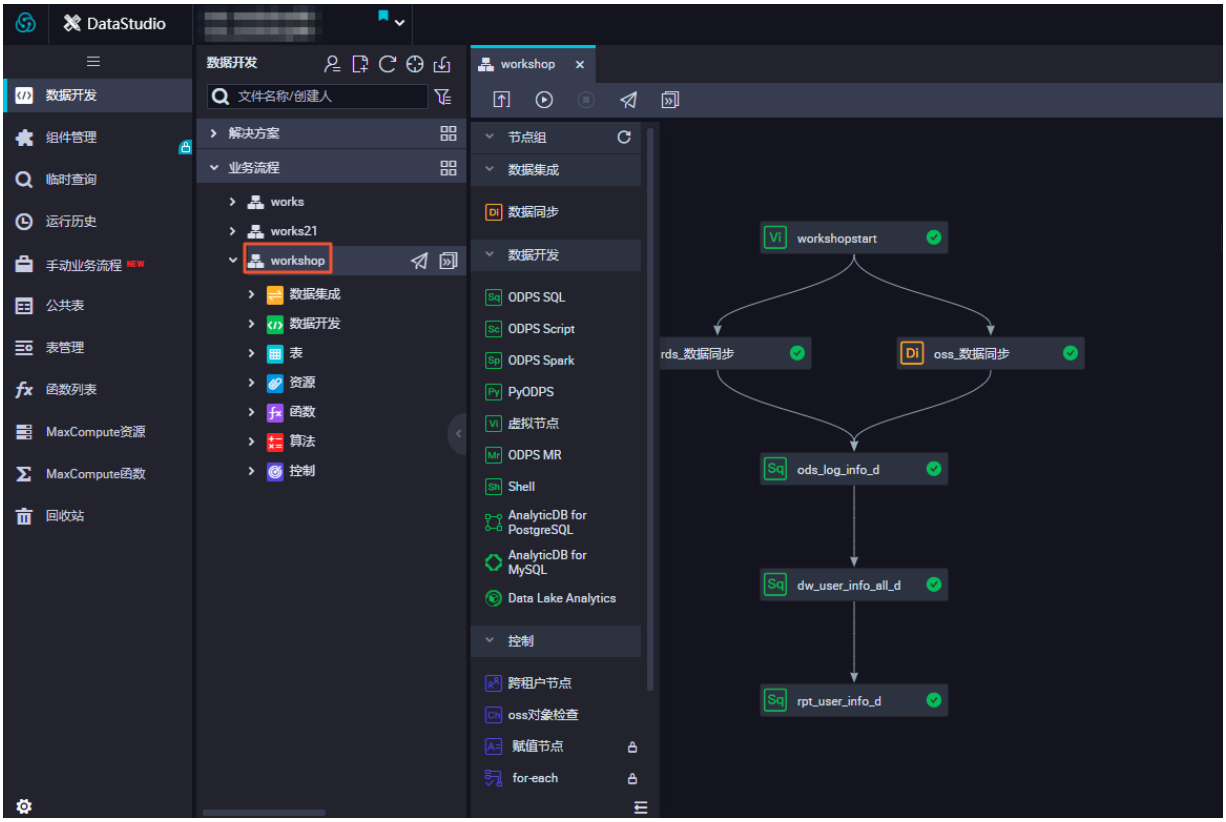
- 控制

  控制节点包括控制节点包括跨租户节点、oss对象检查、赋值节点、for-each、do-while、归并节点和分支节点，详情请参见#unique_287。



> **说明:**
> 除跨租户节点和OSS对象检查所有版本均支持外，其他功能均DataWorks标准版及以上版本方可支持。如果您需要使用相应功能，可以单击立即升级，进行版本升级操作。

双击业务流程的名称，即可在控制面板以工作流图的方式查看各节点之间的关系。

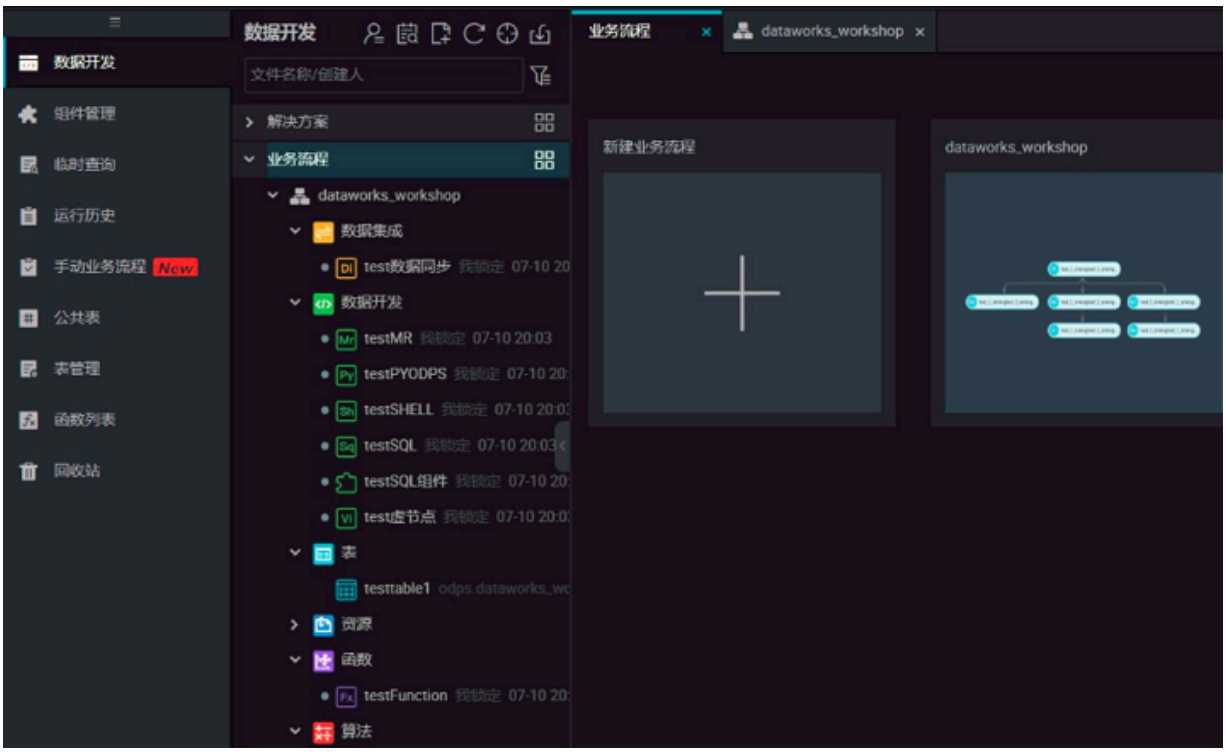


说明:

建议单个业务流程下节点总数不要超过100个。

查看所有的业务流程

在数据开发页面，双击业务流程，即可查看该工作空间下所有的业务流程。



业务流程对象看板

在业务流程中，为每种类型对象都增加了相应的对象集合看板。每个对象在看板上都有对应的一张对象卡片，可以将相关的操作和优化建议附着到相应对象后面，使得相关对象的管理更加智能化更加方便。

例如在数据开发任务对象卡片中，增加了该任务是否有基线强保障或自定义提醒的状态图标提示，方便您了解任务的当前保障状态。

双击业务流程文件夹下每个对象的名称，即可打开该对象类型的对象看板。

提交业务流程

如果您之前使用过DataWorks V1.0版本，在工作流切换至DataWorks V2.0的业务流程后，提交业务流程时请您保证已添加好备注，否则无法提交。

说明:

如果您的节点已经提交过，在没有修改节点内容，只是修改了业务流程或节点属性的情况下，可以不选择节点（如果节点已经被提交过，在不改变节点内容的情况下节点无法被再次选择），填写备注后提交业务流程。相关改动会正常被提交。

3.4.2 资源

本文将为您介绍如何新建、上传、引用和下载资源。

如果您的代码或函数中需要使用.jar等资源文件，可以先将资源上传至该工作空间，然后进行引用。

如果现有的系统内置函数无法满足您的需求，DataWorks支持创建自定义函数，实现个性化处理逻辑。将实现逻辑的JAR包上传至工作空间下，便可在创建自定义函数时进行引用。

您可以将文本文件、MaxCompute表、Python代码以及.zip、.tgz、.tar.gz、.tar、.jar等压缩包作为不同类型的资源上传到MaxCompute，在UDF及MapReduce的运行过程中读取、使用这些资源。

MaxCompute为您提供读取、使用资源的接口。目前资源包括以下类型：

- File类型
- Archive类型：通过资源名称中的后缀识别压缩类型，支持的压缩文件类型包括.zip、.tgz、.tar.gz、.tar和.jar。
- JAR类型：编译好的Java JAR包。
- Python类型：您编写的Python代码，用于注册Python UDF函数。

DataWorks新建资源就是add resource的过程，当前DataWorks仅支持可视化添加JAR和File类型的资源。新建入口都一样，区别如下：

- JAR资源是用户在线下Java环境编辑Java代码，打JAR包上传到JAR资源类型文件。
- File类型的小文件资源可以直接在DataWorks上编辑。
- File类型资源新建时勾选大文件后，也可以上传本地资源文件。

说明：

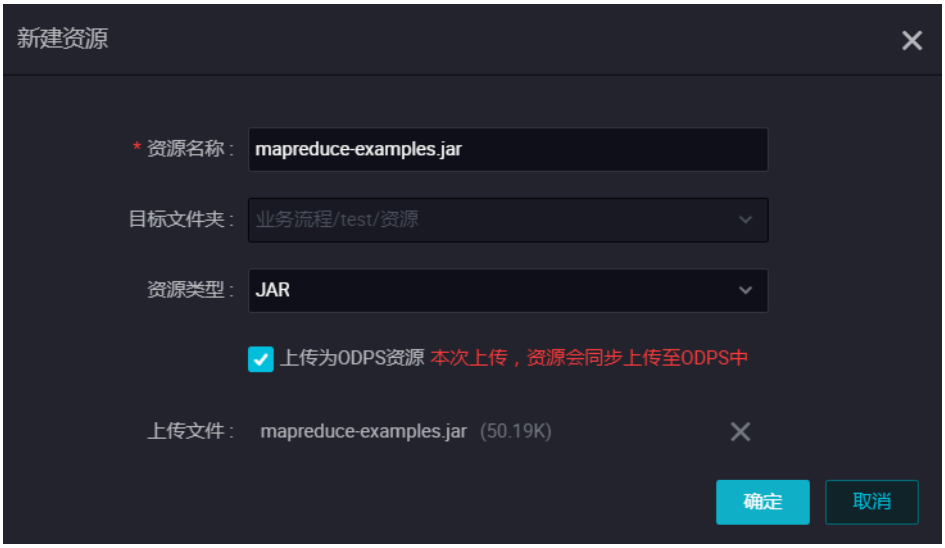
当前支持最大可上传30MB资源。

新增JAR资源实例

1. 右键单击数据开发下的业务流程，选择新建业务流程。

2. 打开新建的业务流程，右键单击资源，选择新建资源 > JAR。

3. 按照命名规则在新建资源对话框输入资源名称，并选择资源类型为JAR，同时选择需要上传本机的JAR包。



说明:

- 如果此JAR包已经在ODPS客户端上传过，则需要取消勾选上传为ODPS资源本次上传，资源会同步上传至ODPS中，否则上传会报错。
- 资源名称不一定与上传的文件名一致。
- 资源名命名规范：1到128个字符，字母、数字、下划线、小数点，大小写不敏感，JAR资源时后缀为.jar。

4. 单击确定，将资源提交到调度开发服务器端。

5. 发布节点任务。

具体操作请参见#unique_289。

新增Python资源并注册函数实例

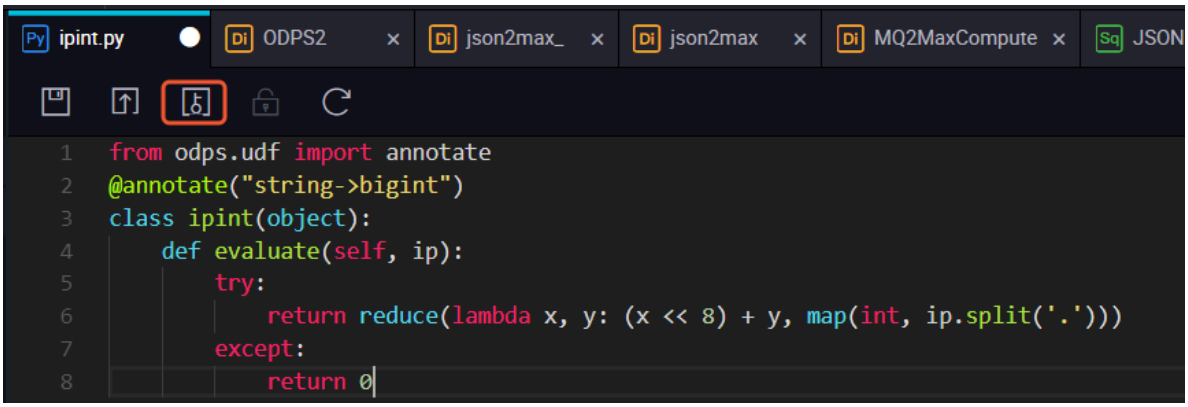
1. 打开新建的业务流程，右键单击资源，选择新建资源 > Python。

2. 按照命名规则在新建资源对话框输入资源名称，并选择资源类型为Python，并勾选上传为ODPS资源，单击确定。

3. 在您新建的Python资源内编写Python资源代码，示例如下。

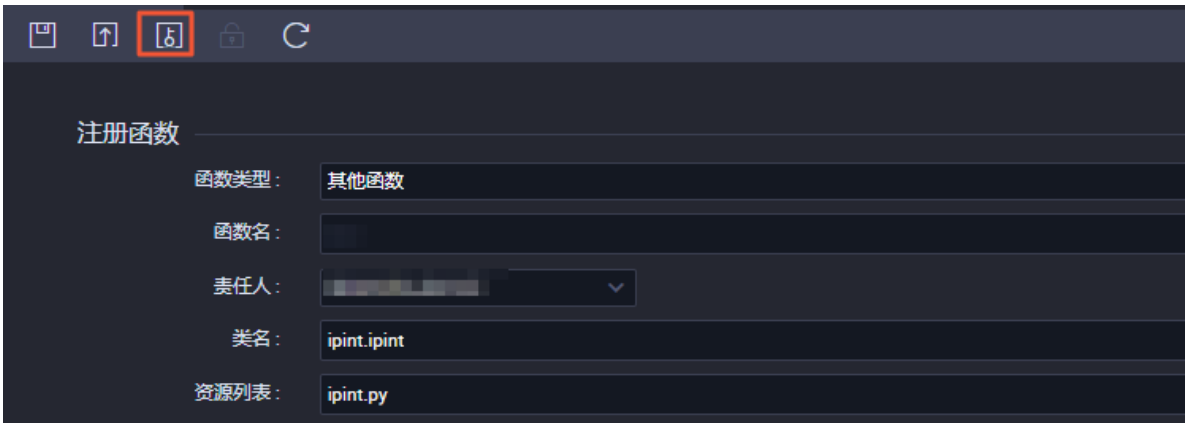
```
from odps.udf import annotate
@annotate("string->bigint")
class ipint(object):
    def evaluate(self, ip):
        try:
            return reduce(lambda x, y: (x << 8) + y, map(int, ip.
split('.')))
        except:
            return 0
```

单击提交并解锁。

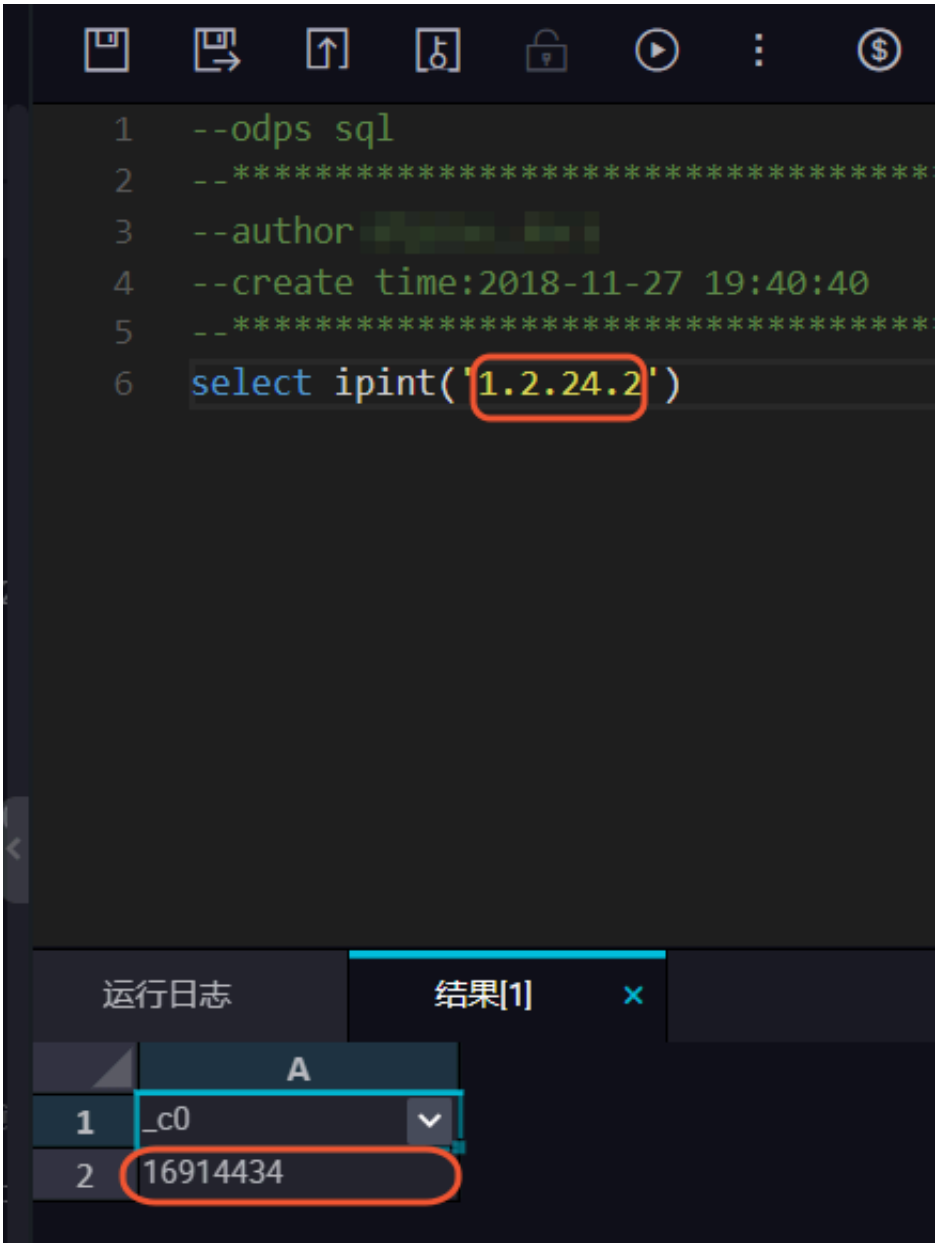


4. 右键单击业务流程下的函数，选择新建函数。在新建函数对话框中填写函数名称，单击提交。

5. 填写函数的类名，本例中为`ipint.ipint`，资源列表填写提交的资源名称，单击提交并解锁。



6. 验证ipint函数是否生效并满足预期值，您可以在DataWorks上新建一个ODPS SQL类型节点运行SQL语句查询，示例如下。



您也可以在本地创建ipint.py文件，使用MaxCompute客户端上传资源，详情请参见MaxCompute客户端。

```
odps@ MaxCompute_DOC>add py D:/ipint.py;
OK: Resource 'ipint.py' have been created.
```

完成上传后，使用客户端直接注册函数，详情请参见注册函数。

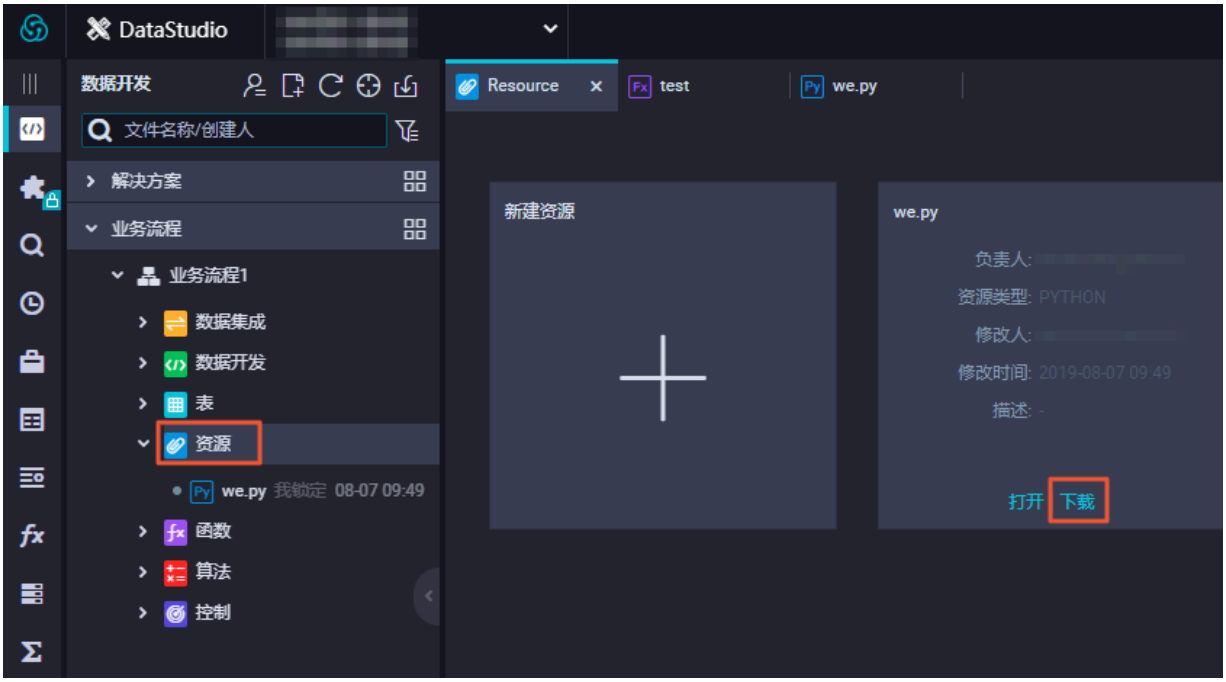
```
odps@ MaxCompute_DOC>create function ipint as ipint.ipint using ipint.
py;
Success: Function 'ipint' have been created.
```

完成注册后，即可正常使用该函数。

引用和下载资源

- 在函数中引用资源请参见注册函数。
- 在节点中引用资源请参见ODPS_MR节点。

如果您需要下载资源，可以双击资源选择您需要的资源，单击下载。



3.4.3 注册函数

DataWorks当前支持Python和Java两种语言接口。本文为您介绍如何注册您的UDF代码。

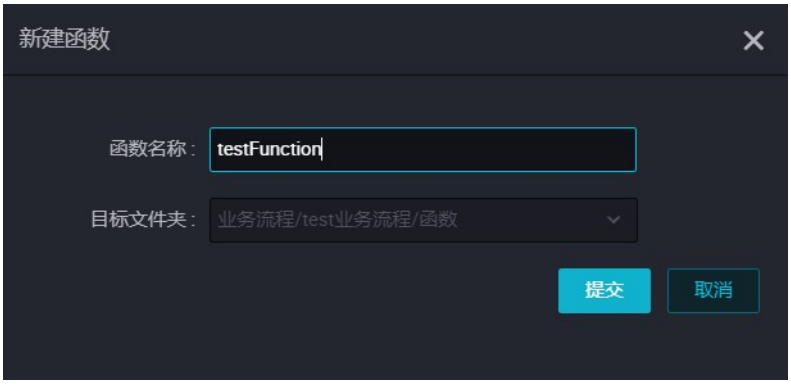
首先通过添加资源 的方式将UDF代码上传，然后再注册函数。

注册函数的操作步骤，如下所示：

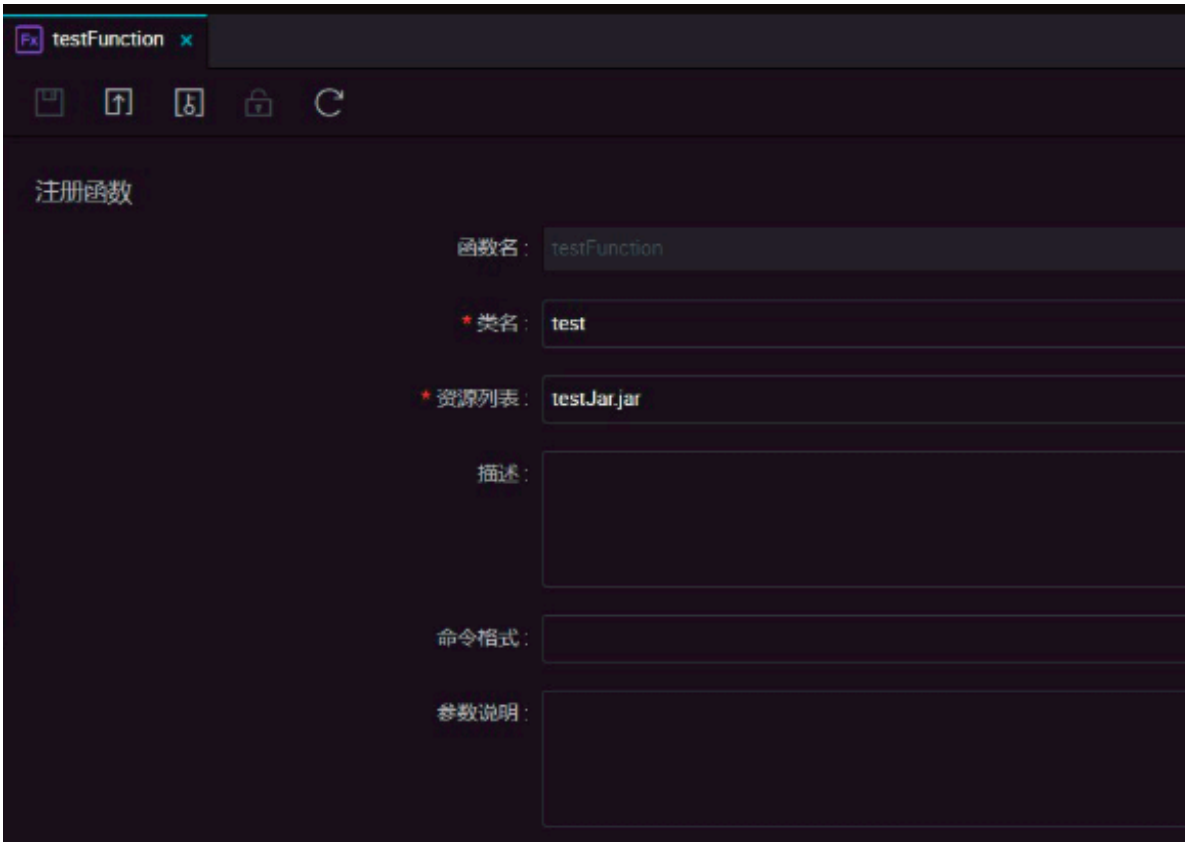
1. 右键单击数据开发下的业务流程，选择新建业务流程。

2. 本地Java环境编辑程序打Jar包，新建Jar资源，提交发布。详情请参见添加资源 。
3. 新建函数。

   右键单击函数，选择新建函数，输入新建函数的名称。



4. 编辑函数配置。



- 类名：实现UDF的主类名。
- 资源列表：第二步中的资源名称，多个资源用逗号分隔。
- 描述：UDF描述，非必填项。

5. 提交任务。

   完成配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

6. 发布任务。

   具体操作请参见#unique_289。

3.4.4 节点组

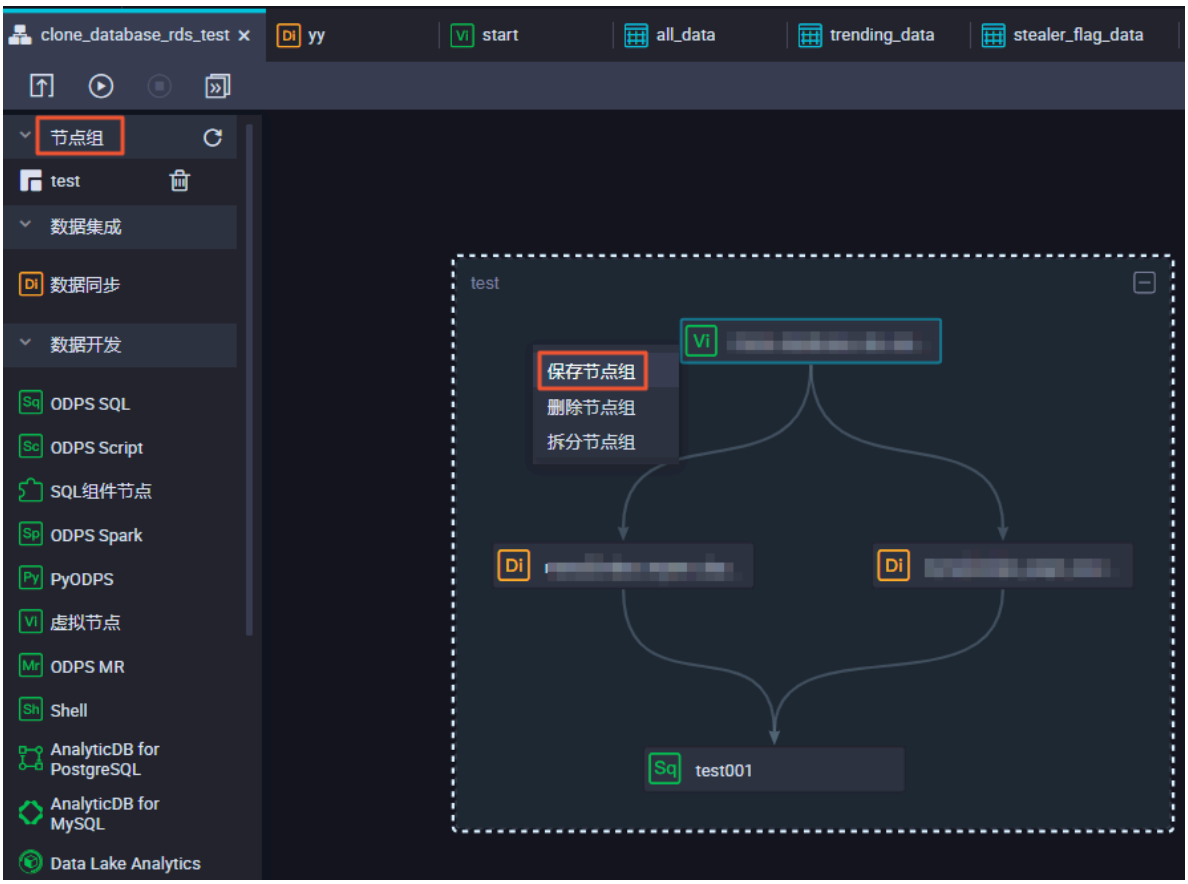
本文将为您介绍如何新建和引用节点组。

新建节点组

1. 进入数据开发页面，新建业务流程。

2. 单击工作流看板右上角的框选按钮，右键单击相应节点，选择新增节点组。

3. 创建好节点组后，右键单击节点组，选择保存节点组，才能在节点组下拉框看到相应的内容。

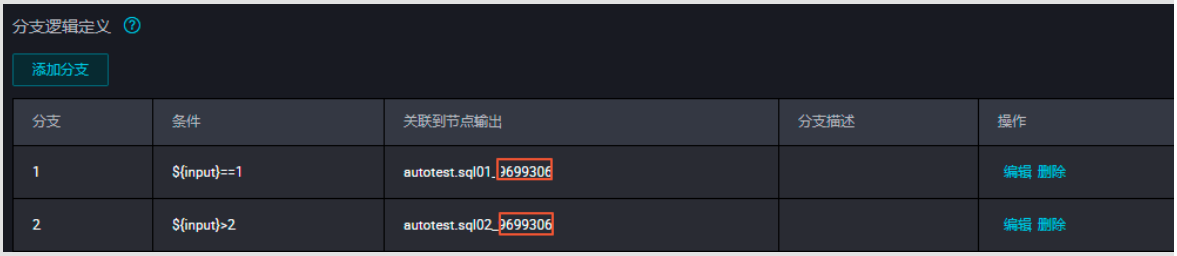


操作	说明
保存节点组	单击保存节点组，才能在节点组列表中展现。如果不保存，则不能在其他工作流中引用。

操作	说明	
删除节点组	您可以单击删除节点组，将选择的节点直接删除。	
拆分节点组	拆分节点组只能影响您目前选择的节点，能使状态回到没有选择节点的时候，这样您可以重新选择节点。已经保存在节点组列表的不受影响。	

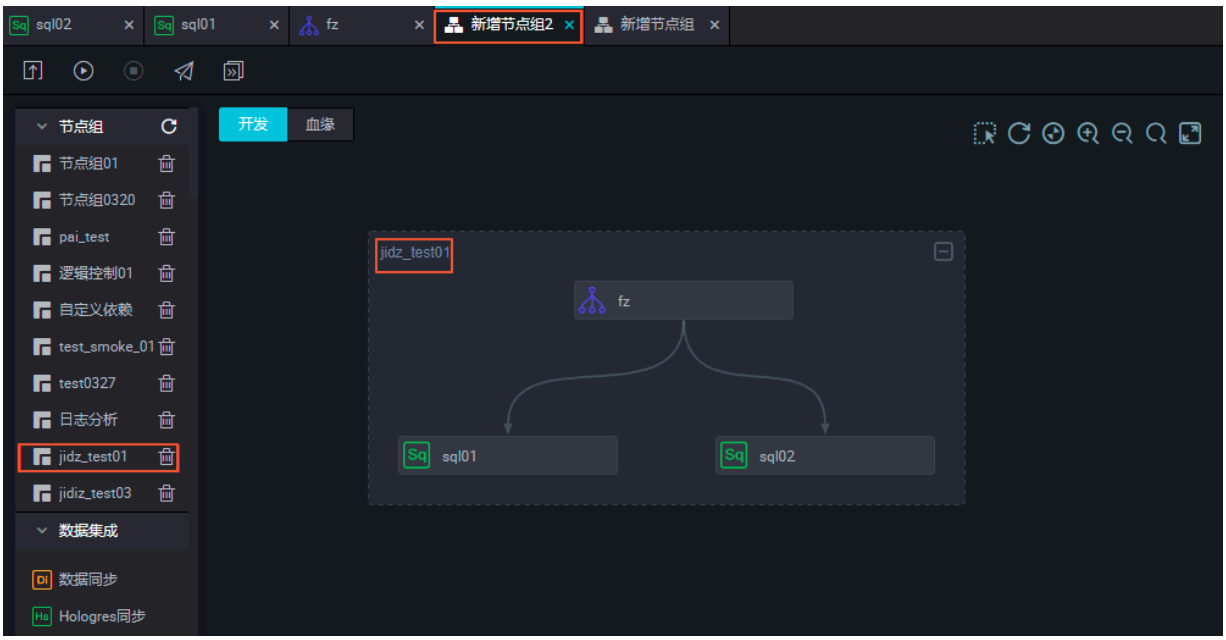
说明:

如果创建节点组中包含PAI节点，在其他业务流程中要重新创建实验。如果是分支节点，在关联到节点输出加上数字。



引用节点组

直接将节点组拖入业务流程中，即可在其他业务流程引用该节点组，节点组中的依赖关系保持不变。



您可以直接运行或提交后发布工作流，进入运维中心页面查看相关的运行结果。

3.5 节点类型

3.5.1 节点类型介绍

DataWorks（数据工场，原大数据开发套件）支持多种类型的节点，分别适用于不同的使用场景。

数据同步节点

数据同步节点是阿里云数加平台对外提供的稳定高效、弹性伸缩的数据同步云服务。您可以通过数据同步节点，轻松地将业务系统数据同步至MaxCompute。详情请参见#unique_282。

ODPS Script节点

DataWorks提供ODPS Script节点类型，新建和配置操作请参见ODPS Script节点。

ODPS SQL节点

ODPS SQL任务支持您直接在Web端编辑和维护SQL代码，并可以方便地调试运行和协作开发。DataWorks还支持代码内容的版本管理和上下游依赖自动解析等功能，使用示例请参见#unique_298。

DataWorks默认使用MaxCompute的项目作为开发生产空间，因此ODPS SQL节点的代码内容遵循MaxCompute SQL的语法。MaxCompute SQL采用的是类似于Hive的语法，可以看作是标准SQL的子集，但不能因此简单地把MaxCompute SQL等价成一个数据库，它在很多方面并不具备数据库的特征，如事务、主键约束和索引等。

具体的MaxCompute SQL语法请参见#unique_299。

SQL组件节点

SQL组件是一种带有多个输入参数和输出参数的SQL代码过程模板，SQL代码的处理过程一般是引入一到多个源数据表，通过过滤、连接和聚合等操作，加工出新的业务需要的目标表，详情请参见#unique_300。

ODPS Spark节点

DataWorks提供ODPS Spark节点类型，新建和配置操作请参见#unique_301。

虚拟节点

虚拟节点属于控制类型节点，它不产生任何数据的空跑节点，常用于工作流统筹节点的根节点，虚节点任务详情请参见#unique_302。

说明:

工作流的最终输出表有多个分支输入表，且这些输入表没有依赖关系时便经常用到虚拟节点。

ODPS MR节点

MaxCompute提供MapReduce编程接口，您可以使用MapReduce提供的接口（Java API）编写MapReduce程序处理MaxCompute中的数据，您可以通过创建ODPS_MR类型节点的方式在任务调度中使用，使用示例请参见#unique_292。

Shell节点

Shell节点支持标准Shell语法，不支持交互性语法。Shell节点可以在默认资源组上运行，如果需要访问IP/域名，请在项目管理下的项目配置页面将IP/域名添加到白名单中。详情请参见Shell节点。

PyODPS节点

Maxcompute提供了#unique_304，您可以使用Python的SDK来操作Maxcompute。

DataWorks也提供PyODPS节点类型，集成了Maxcompute的Python SDK，可以在DataWorks的PyODPS节点上直接编辑Python代码操作Maxcompute。详情请参见#unique_305。

for-each节点

您可以通过for-each节点实现循环N次，每次循环中把当前的循环次数打印出来的需求。详情请参见遍历（for-each）节点。

> 说明:
> 您需要购买DataWorks标准版及以上版本，方可使用此功能。

do-while节点

您可以在do-while节点中定义相互依赖的任务，任务中包含一个名为end的循环判断节点。Dataworks会不断重复执行这一批任务，直到循环判断节点end把判断结果置为false，Dataworks才会退出整个循环。详情请参见#unique_307。

> 说明:
> 您需要购买DataWorks标准版及以上版本，方可使用此功能。

跨租户节点

跨租户节点主要用于不同租户的节点之间的联动，分为发送节点和接收节点。详情请参见#unique_308。

归并节点

归并节点可以对上游节点的运行状态进行归并，用来解决分支节点下游节点的依赖挂载和运行触发问题。详情请参见#unique_309。


说明:
您需要购买DataWorks标准版及以上版本，方可使用此功能。

分支节点

分支节点是DataStudio中提供的逻辑控制系列节点中的一类。分支节点可以定义分支逻辑和不同逻辑条件时下游分支走向。详情请参见#unique_310。


说明:
您需要购买DataWorks标准版及以上版本，方可使用此功能。

赋值节点

赋值节点是一种特殊的节点类型，支持在节点中通过编写代码的方式对输出参数赋值，结合节点上下文传递，供下游节点引用和使用其取值。详情请参见#unique_311。


说明:
您需要购买DataWorks标准版及以上版本，方可使用此功能。

OSS对象检查节点

当下游任务需要依赖该OSS对象何时传入OSS时，可以使用OSS对象检查节点功能。例如同步OSS数据到DataWorks，需要先检测OSS数据文件已经产生，方可进行OSS同步任务。详情请参见#unique_312。

3.5.2 数据同步节点

您只需要输入原表名称和目标表名称即可完成一个简单的任务配置。

目前数据同步任务支持的数据源类型包括ODPS、MySQL、DRDS、SQL Server、PostgreSQL、Oracle、MongoDB、DB2、OTS、OTS Stream、OSS、FTP、Hbase、LogHub、HDFS和Stream，更多支持的数据源请参见#unique_25。

当您输入表名时，页面会自动弹所有匹配表名的对象列表（当前只支持精确匹配，请输入完整的正确的表名）。部分对象当前同步中心不支持，会被打上“不支持”标签。您可以将鼠标移动到列表对象上，页面会自动展示对象的详细信息，例如表所在库、IP、Owner等，这些信息可以协助

您选择正确的表对象。选中后单击对象，列信息会自动填充。当然您也可以对非ODPS表进行列编辑，包括移动、删除、添加等操作。

创建同步任务

详情请参见#unique_27/unique_27_Connect_42_section_tfn_1kc_p2b。

节点调度配置

单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

提交

完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

发布节点任务

具体操作请参见发布管理。

在生产环境测试

具体操作请参见#unique_314。

3.5.3 ODPS Script节点

ODPS Script节点的SQL开发模式是MaxCompute基于2.0的SQL引擎提供的脚本开发模式。

编译脚本时，将一个多语句的SQL脚本文件作为一个整体进行编译，不需逐条语句进行编译。将其作为一个整体提交运行，生成一个执行计划，保证一次排队、一次执行，充分利用MaxCompute的资源。

新建ODPS Script节点

1. 进入DataStudio（数据开发）页面，选择新建 > 数据开发 > ODPS Script。

说明:

您也可以找到相应的业务流程，右键单击数据开发，选择新建数据开发节点 > ODPS Script。

2. 填写新建节点对话框中的配置，单击提交。

3. 编辑ODPS SCRIPT节点。

   您可以在此节点中进行脚本模式的脚本编辑，详情请参见#unique_316。

4. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

   具体操作请参见#unique_314。

Script Mode语法结构

Script Mode的SQL编译较为简单，只需按照业务逻辑，用类似于普通编程语言的方式进行编译，不需考虑如何组织语句。

```
--SET语句
set odps.sql.type.system.odps2=true;
[set odps.stage.reducer.num=***;]
[...]
--DDL语句
create table table1 xxx;
[create table table2 xxx;]
[...]
--DML语句
@var1 := SELECT [ALL | DISTINCT] select_expr, select_expr, ...
    FROM table3
    [WHERE where_condition];
@var2 := SELECT [ALL | DISTINCT] select_expr, select_expr, ...
    FROM table4
    [WHERE where_condition];
@var3 := SELECT [ALL | DISTINCT] var1.select_expr, var2.select_expr
, ...
    FROM @var1 join @var2 on ...;
INSERT OVERWRITE|INTO TABLE [PARTITION (partcol1=val1, partcol2=val2
 ...)]
    SELECT [ALL | DISTINCT] select_expr, select_expr, ...
    FROM @var3;
[@var4 := SELECT [ALL | DISTINCT] var1.select_expr, var.select_expr
, ... FROM @var1
    UNION ALL | UNION
    SELECT [ALL | DISTINCT] var1.select_expr, var.select_expr, ...
FROM @var2;
CREATE [EXTERNAL] TABLE [IF NOT EXISTS] table_name
    AS
    SELECT [ALL | DISTINCT] select_expr, select_expr, ...
    FROM var4;]
```

- 脚本模式支持SET语句、部分DDL语句（结果是屏显类型的语句除外，例如desc、show）和DML语句。
- 一个脚本的完整形式是SET语句>DDL语句>DML语句。每种类型语句都可以有0到多个语句，但不同类型的语句不能交错。
- 多个语句以@开始，表示变量连接。

- 一个脚本，目前最多支持一个屏幕显示结果的语句（例如单独的Select语句），否则会报错。不建议您在脚本中执行屏幕显示的Select语句。
- 一个脚本，目前最多支持一个`Create table as`语句，并且必须是最后一句。建议将建表语句和Insert语句分开写。
- 脚本模式下，如果有一个语句失败，整个脚本的语句都不会执行成功。
- 脚本模式下，只有所有输入的数据都准备完成，才会生成一个作业进行数据处理。
- 脚本模式下，如果一个表被写入后，又被读取，会报错。如下所示：

```
insert overwrite table src2 select * from src where key > 0;
@a := select * from src2;
select * from @a;
```

为避免先写后读，可以进行如下修改。

```
@a := select * from src where key > 0;
insert overwrite table src2 select * from @a;
select * from @a;
```

示例如下：

```
create table if not exists dest(key string , value bigint) partitioned
 by (d string);
create table if not exists dest2(key string,value bigint ) partitioned
 by (d string);
@a := select * from src where value >0;
@b := select * from src2 where key is not null;
@c := select * from src3 where value is not null;
@d := select a.key,b.value from @a left outer join @b on a.key=b.key
and b.value>0;
@e := select a.key,c.value from @a inner join @c on a.key=c.key;
@f := select * from @d union select * from @e union select * from @a;
insert overwrite table dest partition (d='20171111') select * from @f;
@g := select e.key,c.value  from @e join @c on e.key=c.key;
insert overwrite table dest2 partition (d='20171111') SELECT * from @g
;
```

Script Mode适用场景

- 脚本模式更适合用来改写需要层层嵌套子查询的单个语句，或因为脚本复杂性而不得不拆成多个语句的脚本。
- 多个输入的数据源数据准备完成的时间相差很大（例如一个凌晨1点可以准备好，另一个上午7点可以准备好），不适合通过table variable衔接，可以拼接为一个大的脚本模式SQL。

3.5.4 ODPS SQL节点

ODPS SQL采用类似SQL的语法，适用于海量数据（TB级）但实时性要求不高的分布式处理场景。

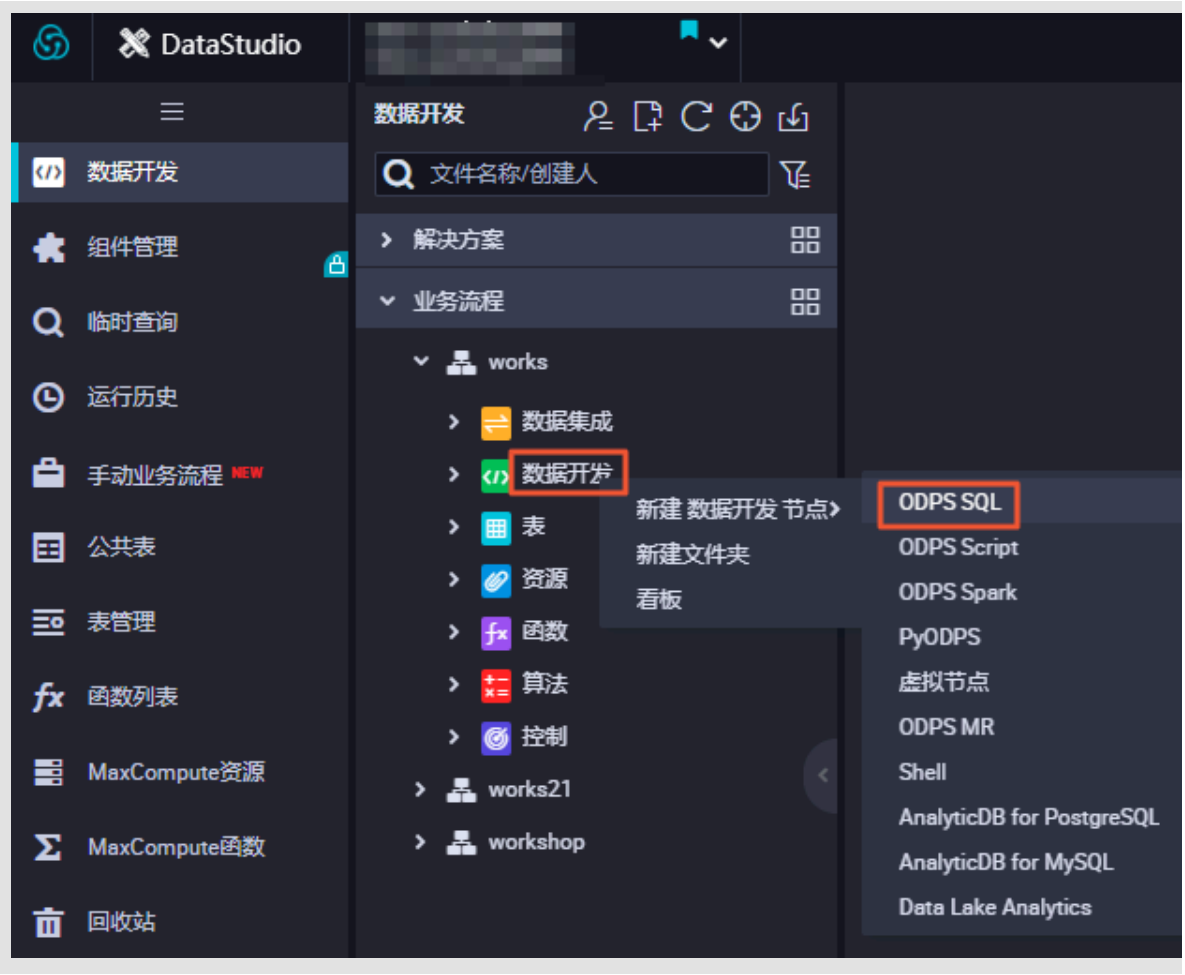
因为每个作业从前期准备到提交等阶段都需要花费较长时间，因此如果要求处理几千至数万笔事务的业务，可以使用ODPS SQL顺利完成。它是主要面向吞吐量的OLAP应用。

新建ODPS SQL节点

1. 进入DataStudio（数据开发）页面，选择新建 > 数据开发 > ODPS SQL。

说明:

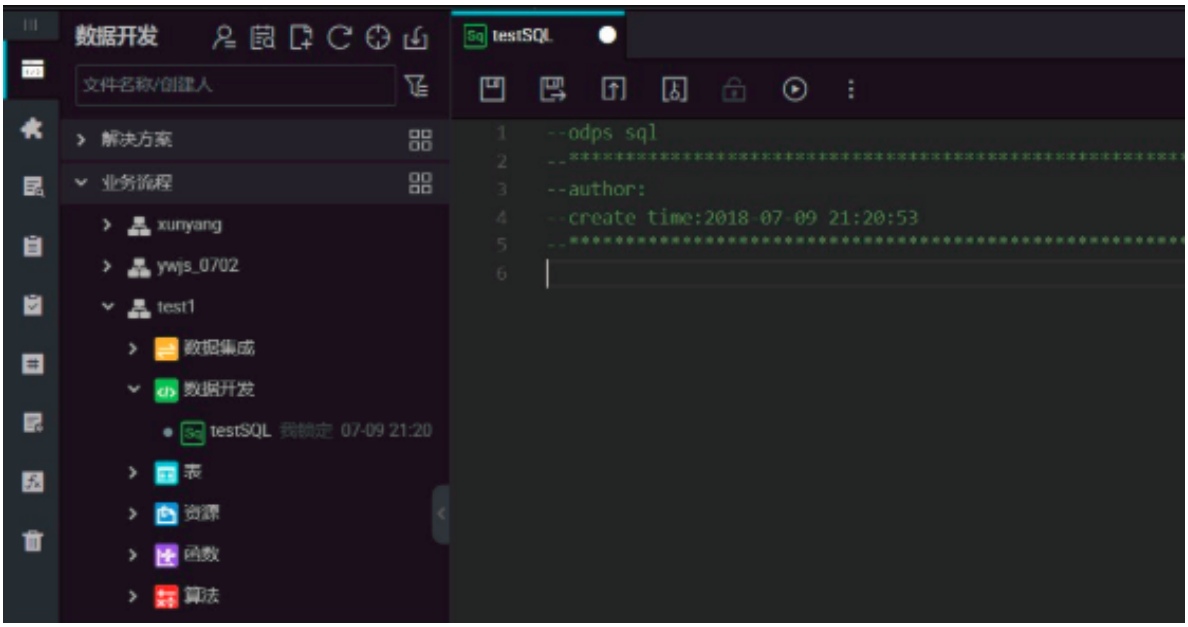
您也可以找到相应的业务流程，右键单击数据开发，选择新建数据开发节点 > ODPS SQL。



2. 填写新建节点对话框中的配置，单击提交。

3. 编辑节点代码。

编写符合语法的ODPS SQL代码，SQL语法请参见#unique_318。



说明:

由于国际标准化组织发布的中国时区信息调整，通过DataWorks执行相关SQL时，日期显示某些时间段会存在时间差异：1900-1928年的日期时间差异5分52秒，1900年之前的日期时间差异9秒。

目前DataWorks不允许节点代码中只包含set语句。如果您需要运行set语句，可以和其他SQL语句一起执行，如下所示：

```
setproject odps.sql.allow.fullscan=true;
select 1;
```

示例：创建一张表并向表中插入数据，查询结果。

a. 创建一张表test1。

```
CREATE TABLE IF NOT EXISTS test1
( id BIGINT COMMENT '' ,
  name STRING COMMENT '' ,
  age BIGINT COMMENT '' ,
  sex STRING COMMENT '');
```

b. 插入准备好的数据。

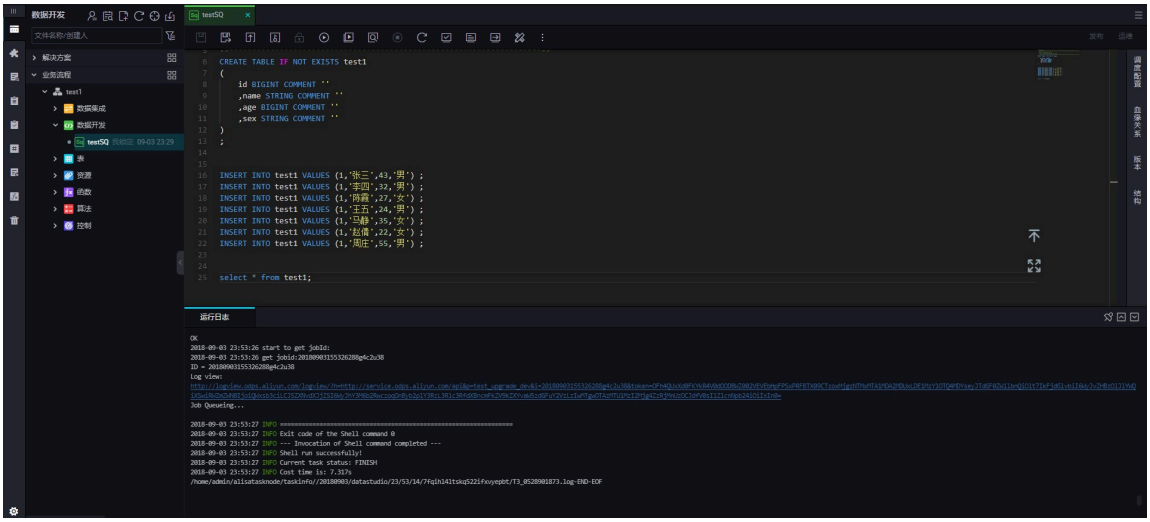
```
INSERT INTO test1 VALUES (1,'张三',43,'男') ;
INSERT INTO test1 VALUES (1,'李四',32,'男') ;
INSERT INTO test1 VALUES (1,'陈霞',27,'女') ;
INSERT INTO test1 VALUES (1,'王五',24,'男') ;
INSERT INTO test1 VALUES (1,'马静',35,'女') ;
INSERT INTO test1 VALUES (1,'赵倩',22,'女') ;
```

```
INSERT INTO test1 VALUES (1,'周庄',55,'男') ;
```

c. 查询表数据。

```
select * from test1;
```

d. 写好SQL，单击顶部的运行或单击F8，此时系统会将我们的SQL按照从上往下的顺序执行，并打印日志。



说明:

当使用insert into语句时，日志中会提示：!!!警告!!! 。

在SQL中使用insert into语句有可能造成不可预料的数据重复。尽管对于insert into语句已经取消SQL级别的重试，但仍然存在进行任务级别重试的可能性，请尽量避免对insert into语句的使用。如果继续使用insert into语句，表明您已经明确insert into语句存在的风险，且愿意承担由于使用insert into语句造成的潜在的数据重复后果。

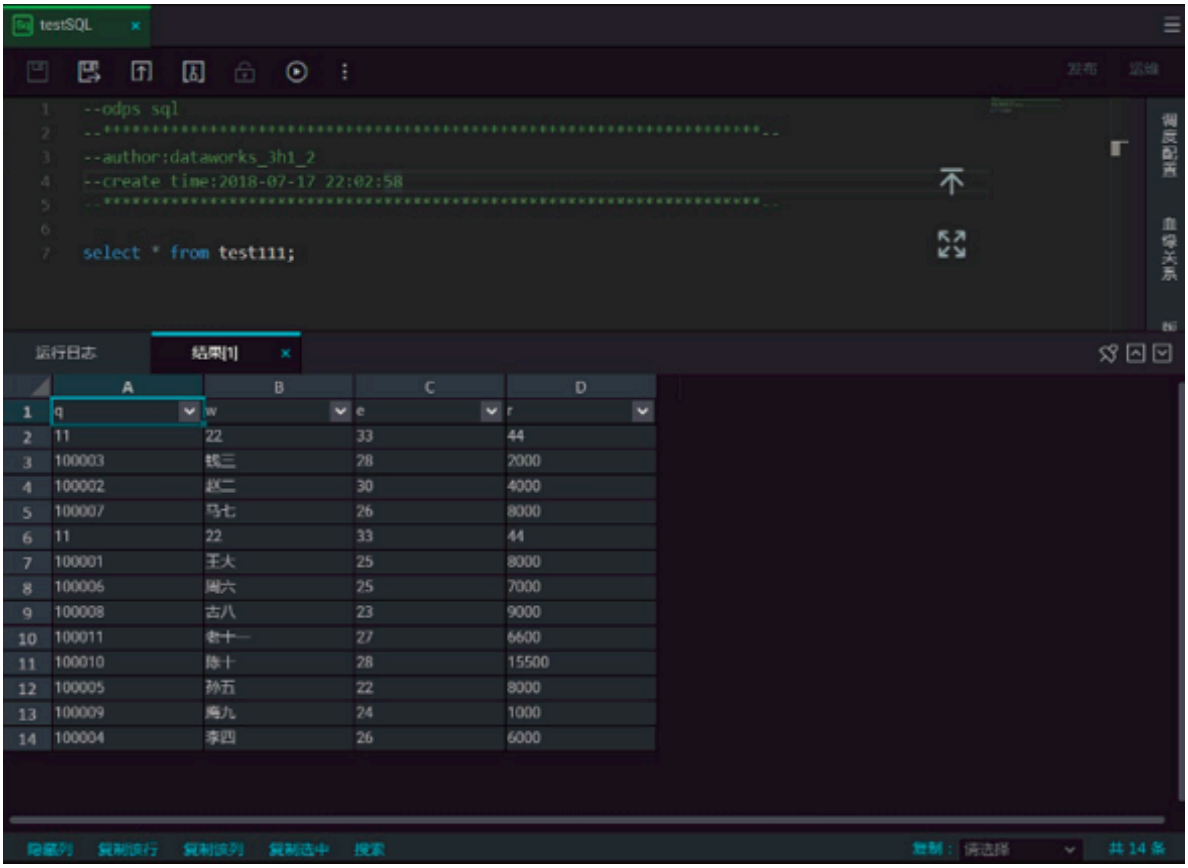
此提示可以根据实际使用需求选择。无论您如何选择，执行SQL的数据已经成功插入到表中，不需要重复执行来确定数据的插入情况，避免出现数据重复。

创建好后，单击左上角的保存，即可保存当前SQL代码。

4. 查询结果展示。

DataWorks的查询结果接入了电子表格功能，方便您对数据结果进行操作。

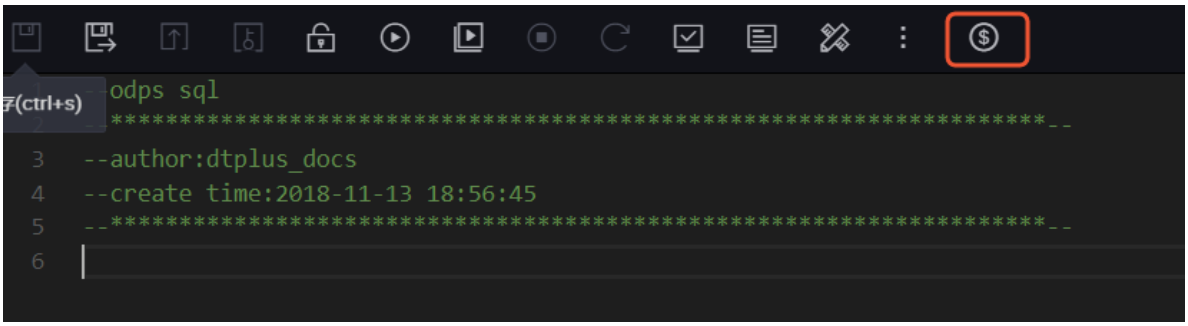
查询出的结果，会直接以电子表格的风格展示出来，您可以在DataWorks中执行操作，或者在电子表格中打开，也可自由复制内容粘贴在本地Excel中。



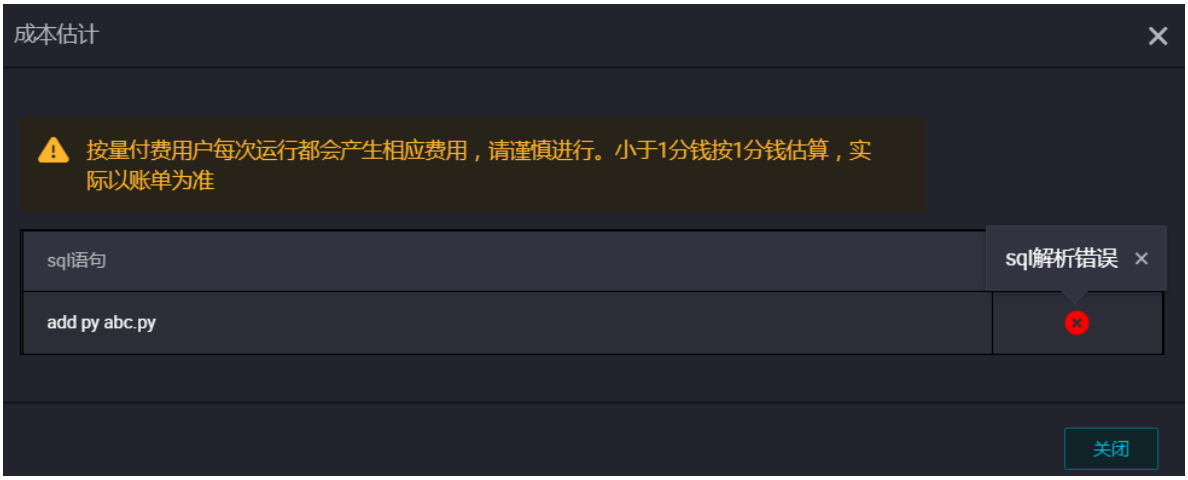
- 隐藏列：选择隐藏其中的一列或多列，可以隐藏该列。
- 复制该行：左侧选中需要复制的一行或多行后，单击复制该行。
- 复制该列：顶部选中需要复制的一列或多列后，单击复制该列。
- 复制：可以自由复制选中的内容。
- 搜索：在查询结果的右上角会出现搜索框，方便对表中数据进行搜素。

5. 成本估计

   您可以点击成本估计按钮估算本次SQL任务可能产生的费用。



   点击后会显示估算出的任务执行费用，如果预估费用一栏报错，您可以将鼠标悬停在X按钮上查看报错原因。



6. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

7. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

8. 发布节点任务。

   具体操作请参见发布管理。

9. 在生产环境测试。

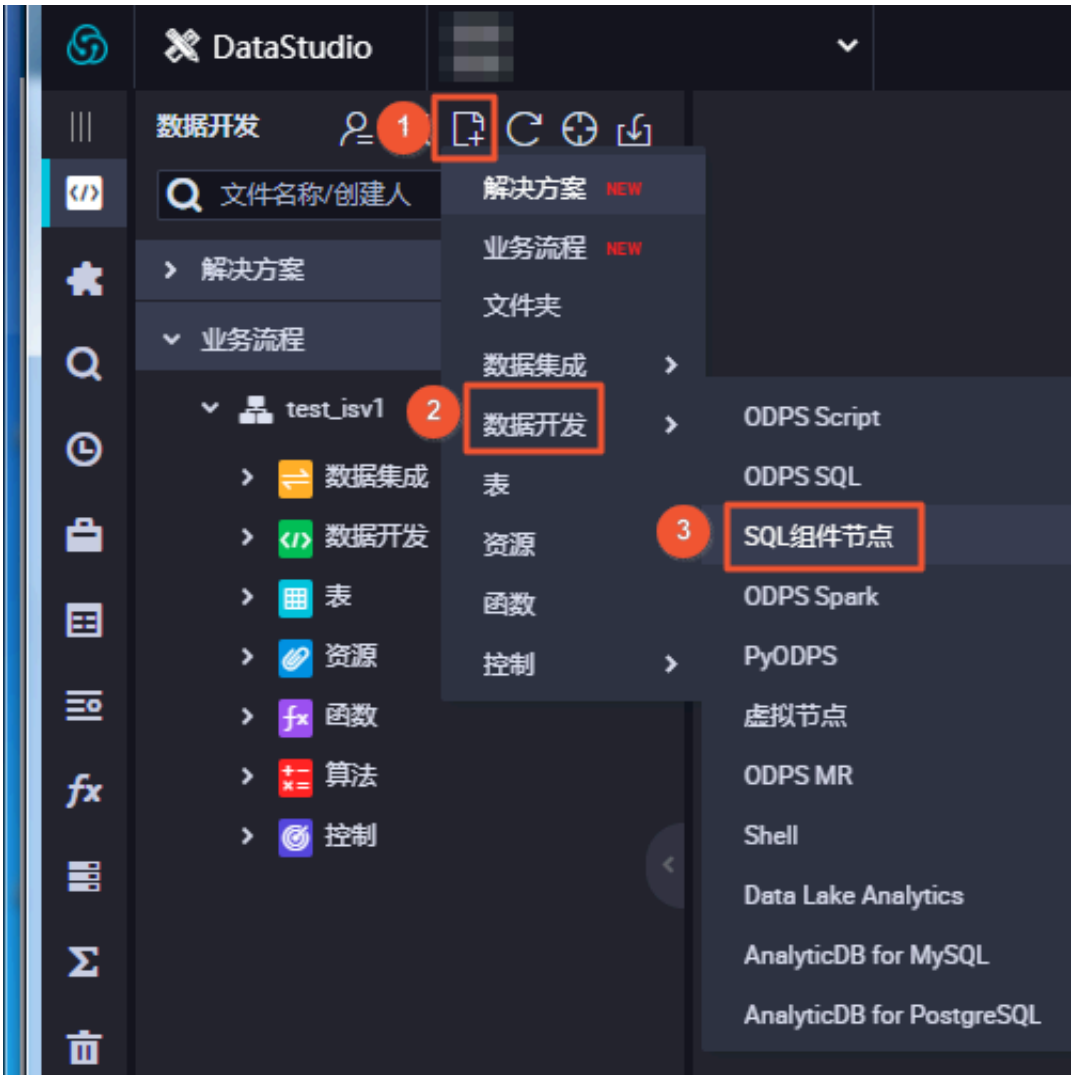
   具体操作请参见#unique_314。

3.5.5 SQL组件节点

SQL组件是一种带有多个输入参数和输出参数的SQL代码过程模板，SQL代码的处理过程通常是引入一到多个源数据表，通过过滤、连接和聚合等操作，加工出新的业务需要的目标表。

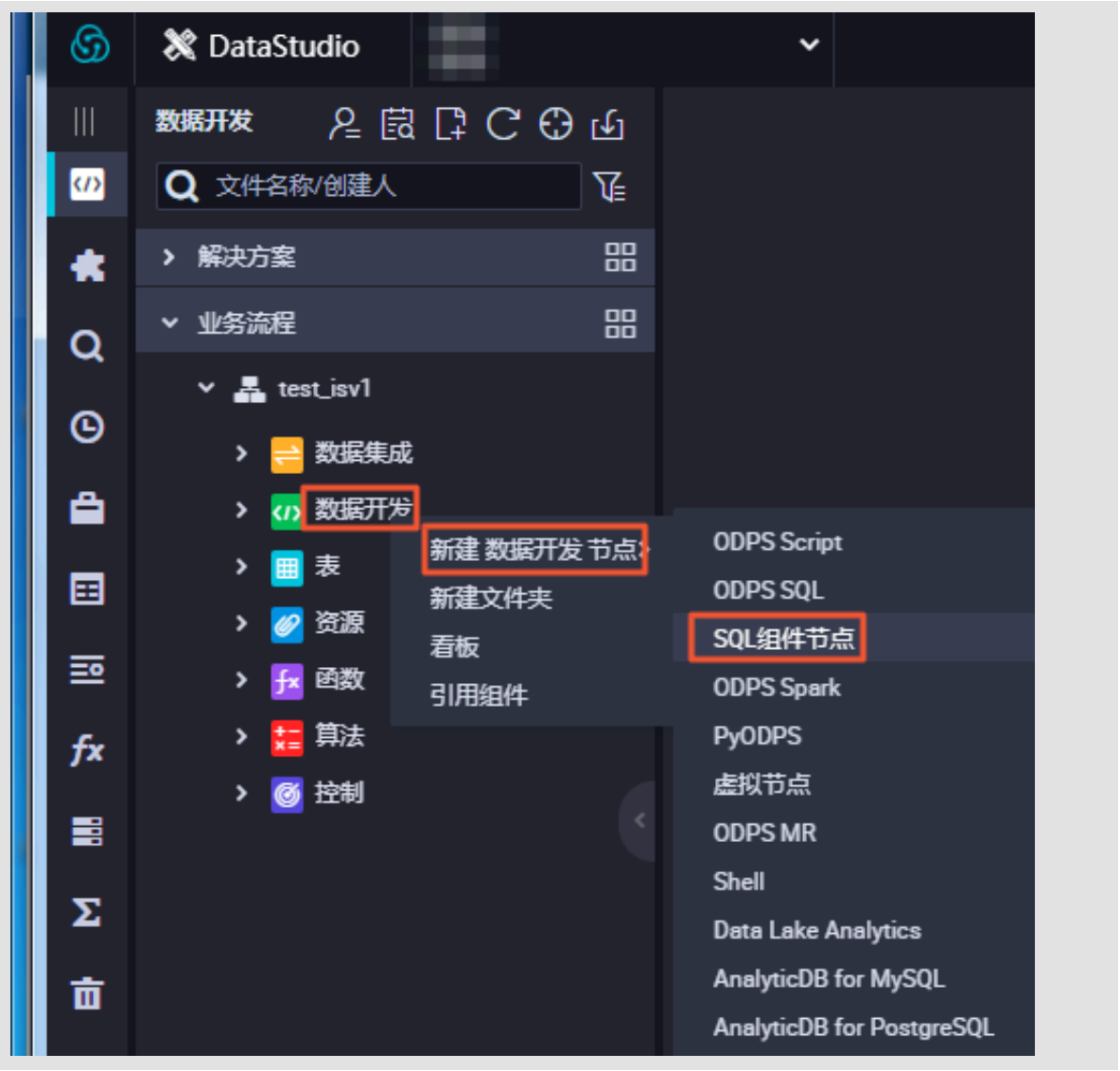
操作步骤

1. 进入DataStudio（数据开发）页面，选择新建 > 数据开发 > SQL组件节点。



说明:

您也可以找到相应的业务流程，右键单击数据开发，选择新建数据开发节点 > SQL组件节点。

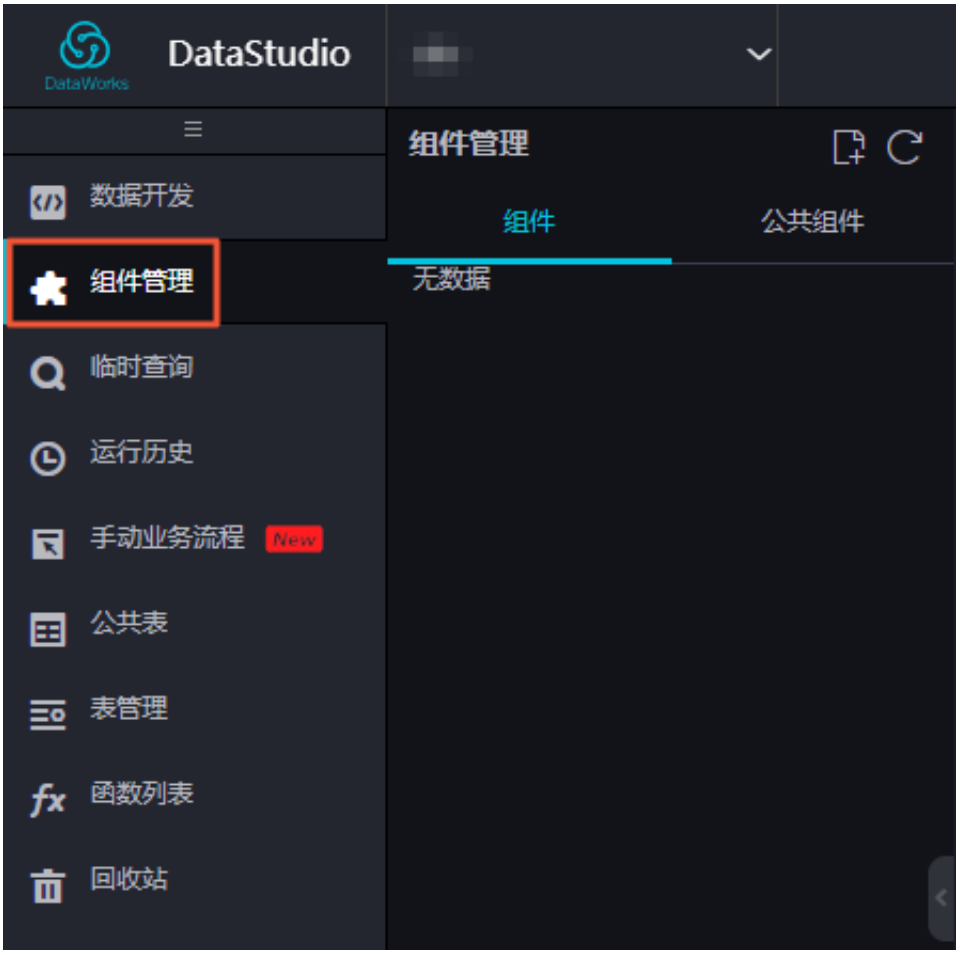

DataStudio
数据开发
文件名称/创建人
解决方案
业务流程
test_isv1
数据集成
数据开发
表
资源
函数
算法
控制
新建 数据开发 节点
新建文件夹
看板
引用组件
ODPS Script
ODPS SQL
SQL组件节点
ODPS Spark
PyODPS
虚拟节点
ODPS MR
Shell
Data Lake Analytics
AnalyticDB for MySQL
AnalyticDB for PostgreSQL

2. 填写新建节点对话框中的配置，单击提交。

为提高开发效率，数据任务的开发者可以使用工作空间成员和租户成员贡献的组件来新建数据处理节点。

- 本工作空间成员创建的组件在项目组件下。
- 租户成员创建的组件在公共组件下。

新建节点时，选择节点类型为SQL组件节点类型，并为该节点指定名字。



为选定的组件指定参数。

输入参数名称后，选择参数类型为Table或String。

get_top_n有三个参数，依次指定。

为table类型参数指定一个输入表test_project.test_table。

3. 节点调度配置。

   单击节点任务编辑区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

4. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

5. 发布节点任务。

   具体操作请参见发布管理。

6. 在生产环境测试。

   具体操作请参见#unique_314。

升级SQL组件节点的版本

在组件的开发者发布新版本后，组件的使用者可以选择是否将现有组件的使用实例升级到使用组件的最新版本。

组件的版本机制支持开发者对组件不断升级，组件的使用者可以不断享受到组件升级后带来的流程执行效率的提升或业务效果优化的种种好处，示例如下。

当A用户使用了B的组件V1.0 版本，这时组件所有人B将组件升级到V2.0，A用户依然可以使用V1.0版本，但是可以看到更新提醒，和新旧代码对比后，A用户发现新的版本业务效果更加优化，就可以自己决定是否升级到最新的组件版本。

升级根据组件模板开发的SQL组件节点的过程也很简单，只要选择升级，然后确认新版本SQL组件节点的参数配置是否继续有效，并根据组件新版本的说明进行些许调整后，就可以像普通SQL节点开发一样，保存提交并进入发布流程。

界面功能

界面功能说明如下：

序号	功能	说明
1	保存	保存当前组件的设置。
2	提交	将当前组件提交到开发环境。
3	提交并解锁	提交当前节点并解锁节点，对代码进行编辑。
4	偷锁编辑	非组件责任人可以偷锁编辑此节点。
5	运行	在本地（开发环境）运行组件。
6	高级运行（带参数运行）	如果代码中有参数，带参数运行代码。
7	停止运行	停止正在运行的代码。
8	重新加载	刷新页面，返回至上次保存的页面。
9	执行冒烟测试	测试当前节点的代码。
10	查看冒烟测试日志	查看节点的运行日志。

3.5.6 ODPS Spark节点

DataWorks提供ODPS Spark节点类型，本文将为您介绍如何新建和配置ODPS Spark节点。

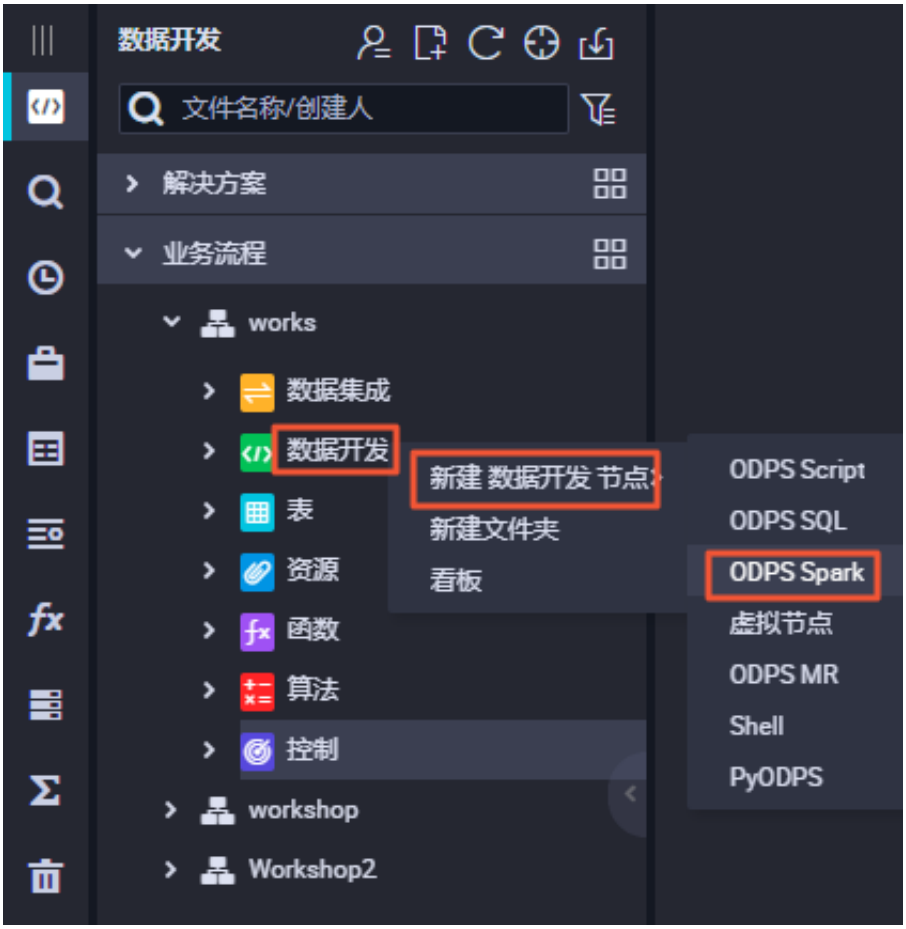
WordCount

1. 右键单击数据开发下的业务流程，选择新建业务流程。

2. 右键单击资源，选择新建资源 > JAR，上传编译的Jar包。

WordCount的示例代码请参见#unique_322。

3. 右键单击业务流程下的数据开发，选择新建数据开发节点 > ODPS Spark，新建ODPS Spark节点。

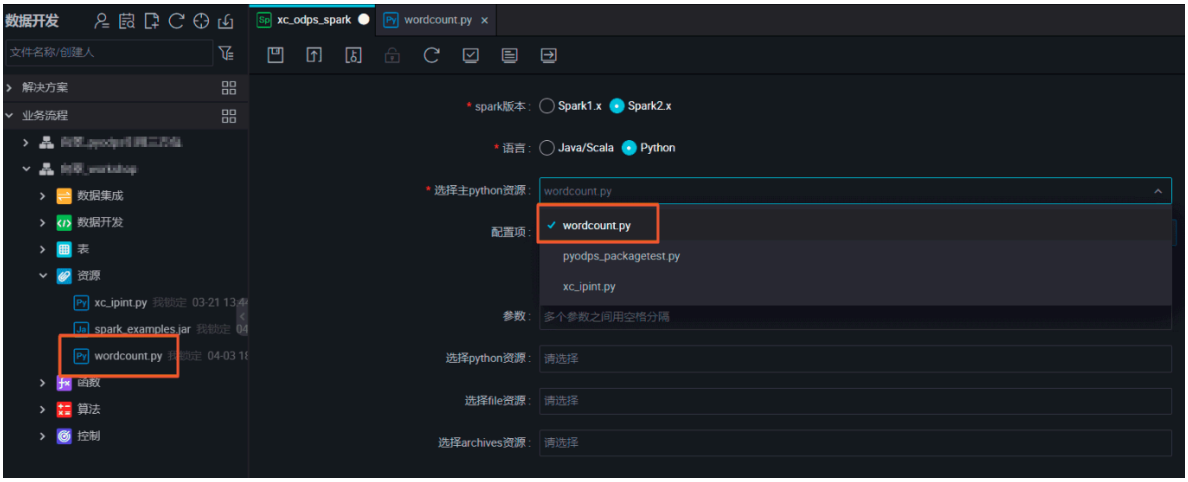


4. 填写ODPS Spark对话框中的配置。

5. 配置完成后，发布和执行该节点。

Python读写表

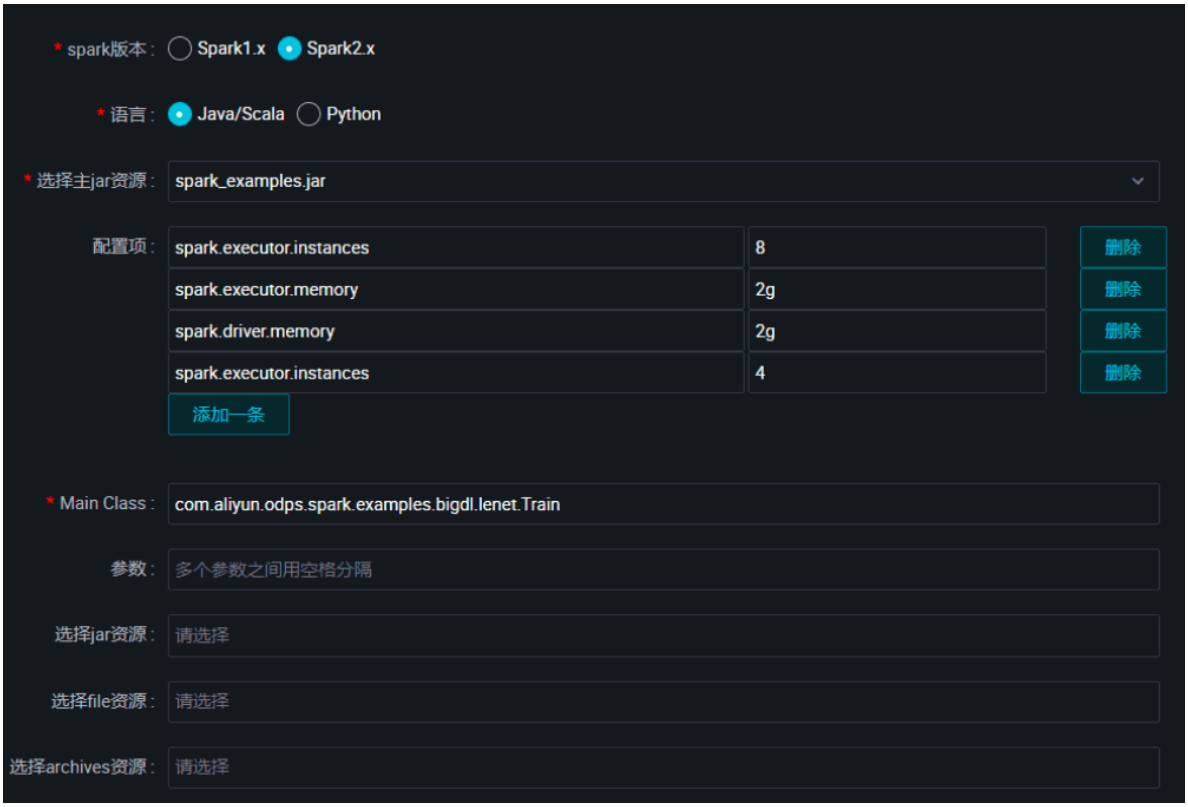
1. 准备好Python代码，并上传Python资源。
2. 新建并配置ODPS Spark节点。



3. 配置完成后，发布和运行该节点。

Lenet（BigDL）

1. 上传Jar包和数据（以archive资源类型上传mnist.zip）。
2. 新建并配置ODPS Spark节点。



3. 配置完成后，发布和运行该节点。

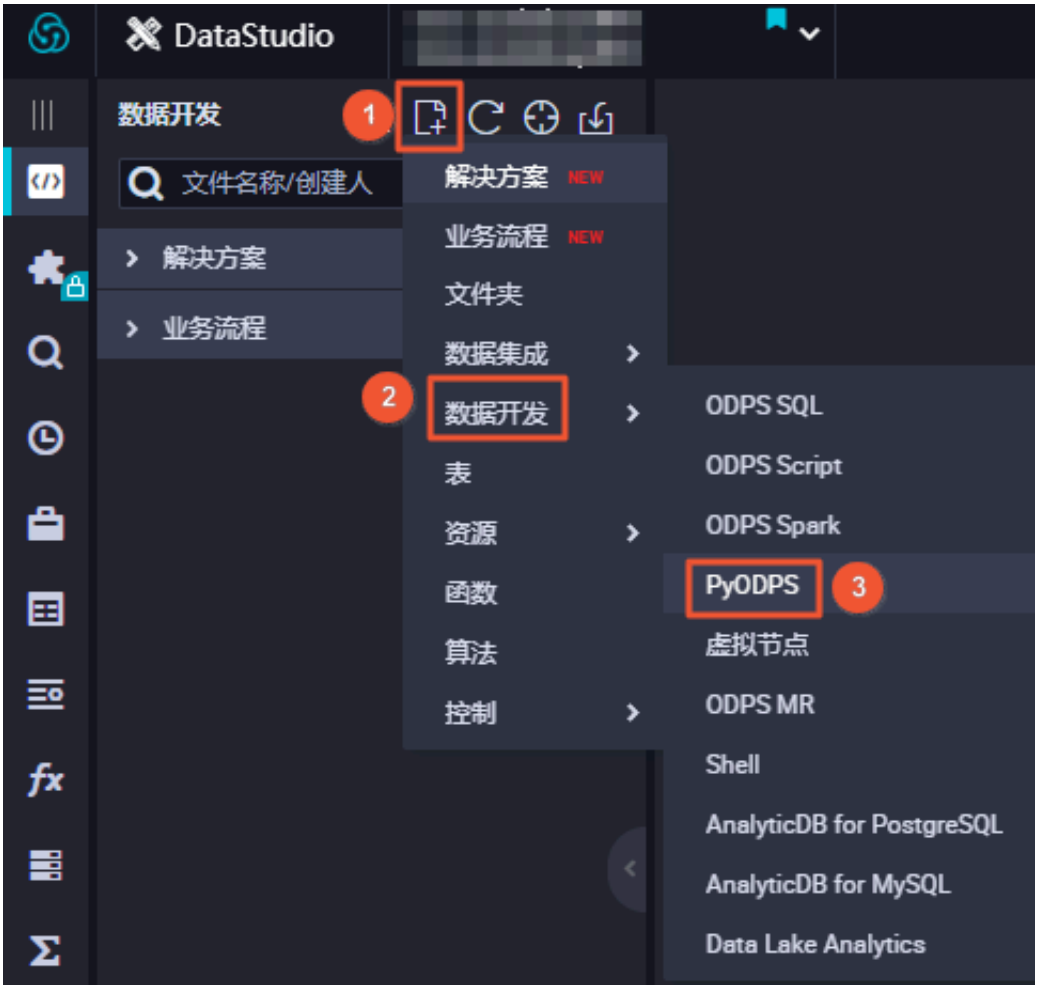
3.5.7 虚拟节点

虚拟节点属于控制类型节点，它是不产生任何数据的空跑节点，常用于工作流统筹节点的根节点。

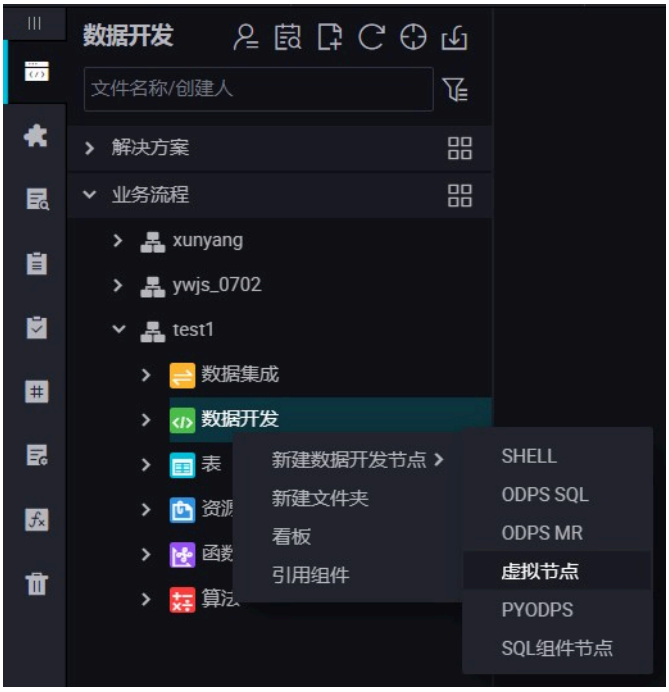
> 说明:
> 工作流中最终输出表有多个分支输入表，且这些输入表没有依赖关系时便经常用到虚拟节点。

新建虚节点任务

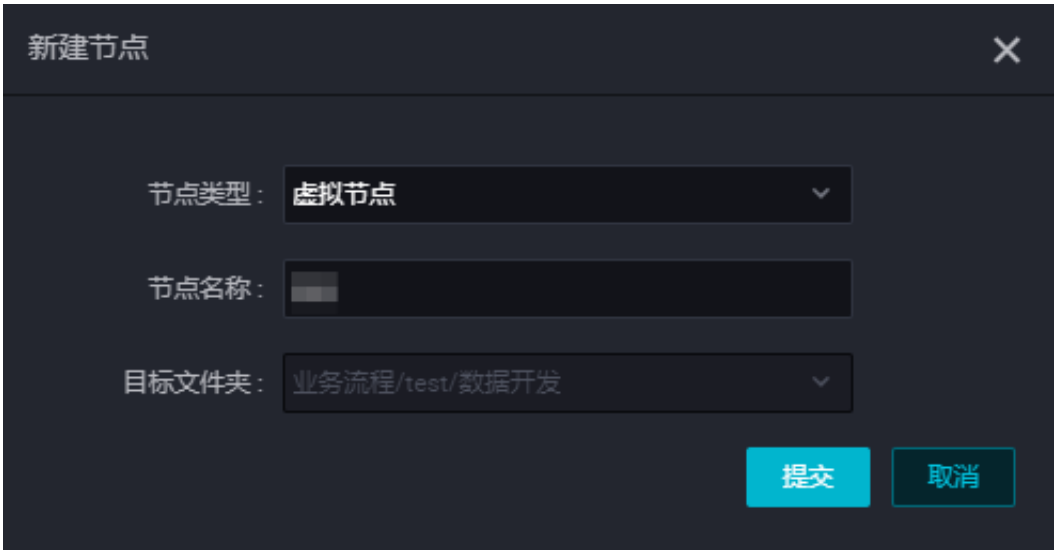
1. 右键单击数据开发下的业务流程，选择新建业务流程。



2. 右键单击数据开发，选择新建数据开发节点 > 虚拟节点。



3. 选择节点类型为虚拟节点，命名节点名称并选择目标文件夹，单击提交。



4. 编辑节点代码：虚拟节点的代码可以不用编辑。
5. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。
6. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。
7. 发布节点任务。

   具体操作请参见发布管理。

8. 在生产环境测试。

   具体操作请参见#unique_314。

3.5.8 ODPS MR节点

MaxCompute提供MapReduce编程接口。您可以通过创建ODPS MR类型节点并提交任务调度，使用MapReduce Java API编写MapReduce程序来处理MaxCompute中的数据。

ODPS MR类型节点的编辑和使用方法，请参见MaxCompute文档示例WordCount示例。

请将需要用到的资源上传并提交发布后，再建立ODPS MR节点。

新建资源实例

1. 右键单击数据开发下的业务流程，选择新建业务流程。

2. 右键单击资源，选择新建资源 > jar。

3. 按照命名规则在新建资源对话框输入资源名称，并选择资源类型为jar，同时选择需要上传本机的Jar包（可以通过 Eclipse 的 Export 功能打包，也可以通过 ant 或其他工具生成）。本例中使用的示例mapreduce_example.jar。

说明：

- 如果此Jar包已经在odps客户端上传过，则需要取消勾选上传为ODPS资源本次上传，资源会同步上传至ODPS中，否则上传会报错。
- 资源名称不一定与上传的文件名一致。
- 资源名命名规范：1到128个字符，字母、数字、下划线、小数点，大小写不敏感，Jar资源时后缀是.jar，Python资源时后缀为.py。

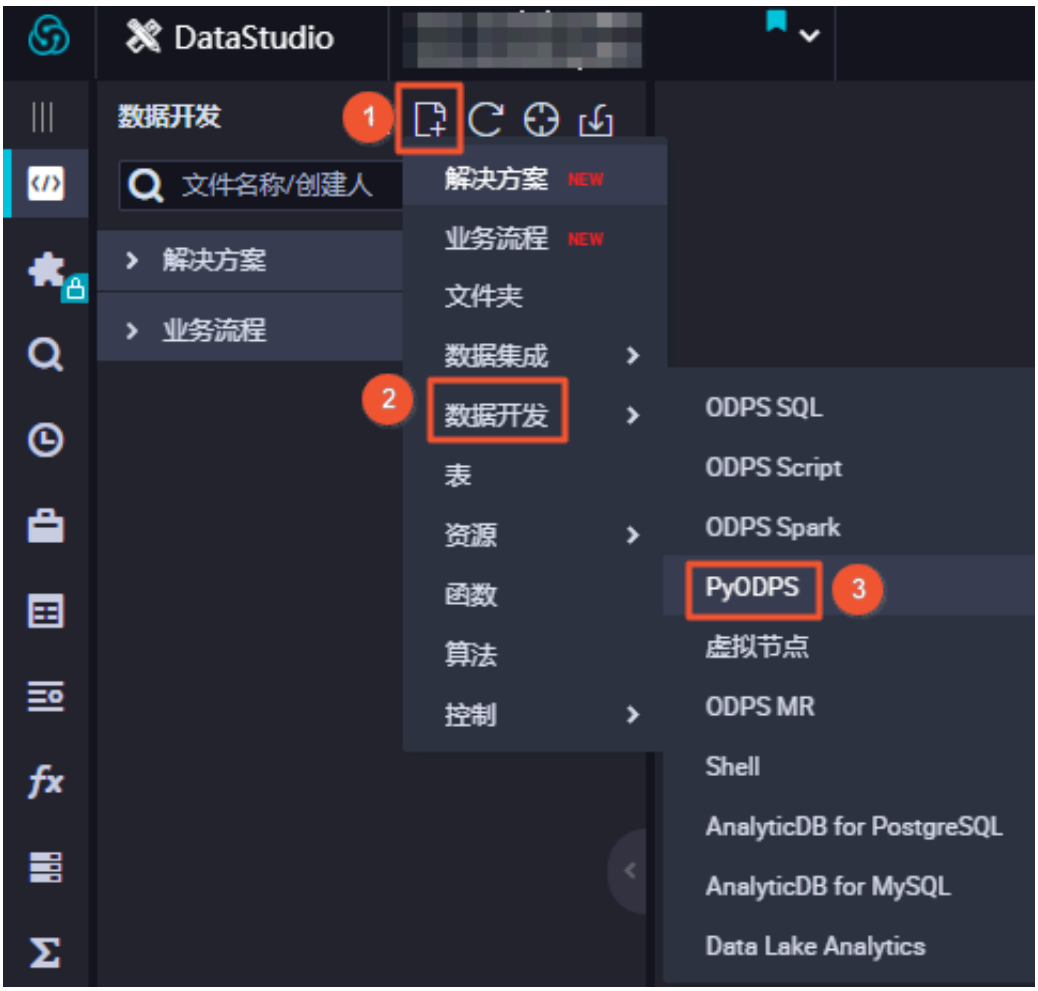
4. 单击提交，将资源提交到调度开发服务器端。

5. 发布节点任务。

   具体操作请参见发布管理。

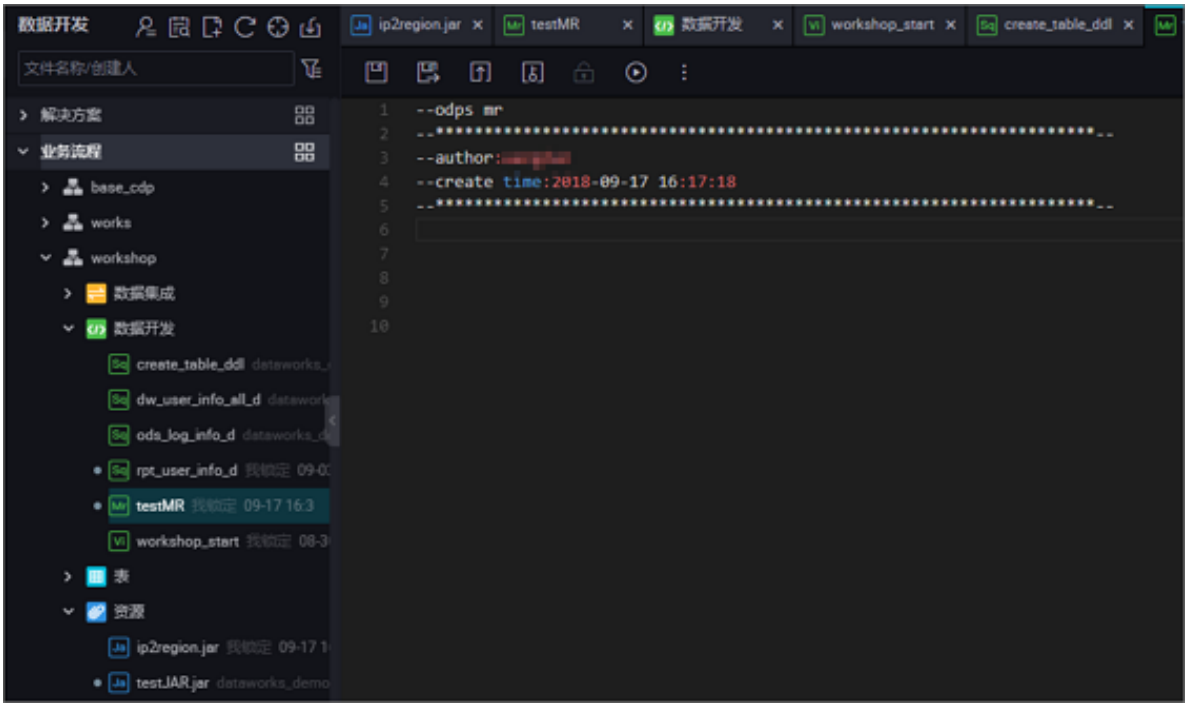
新建ODPS MR节点

1. 右键单击数据开发下的业务流程，选择新建业务流程。



2. 右键单击数据开发，选择新建数据开发节点 > ODPS MR。

3. 编辑节点代码。双击新建的ODPS MR节点，进入如下界面：



编辑节点代码示例：

```
--创建输入表
CREATE TABLE if not exists jingyan_wc_in (key STRING, value STRING);
--创建输出表
CREATE TABLE if not exists jingyan_wc_out (key STRING, cnt BIGINT);
    ---创建系统dual
    drop table if exists dual;
    create table dual(id bigint); --如project中不存在此伪表，则需创建并初始化数据
    ---向系统伪表初始化数据
```

```
    insert overwrite table dual select count(*)from dual;
    ---向输入表 wc_in 插入示例数据
    insert overwrite table jingyan_wc_in select * from (
    select 'project','val_pro' from dual
    union all
    select 'problem','val_pro' from dual
    union all
    select 'package','val_a' from dual
    union all
    select 'pad','val_a' from dual
      ) b;
-- 引用刚刚上传的Jar包资源，可在资源管理栏中找到该资源，右键引用资源。
--@resource_reference{"mapreduce-examples.jar"}
jar -resources mapreduce-examples.jar -classpath ./mapreduce
-examples.jar com.aliyun.odps.mapred.open.example.WordCount
jingyan_wc_in jingyan_wc_out
```

代码说明如下：

- `--@resource_reference{"mapreduce-examples.jar"}` 通过右键资源，选择引用自动产生。
- `-resources`：引用到的Jar资源文件名。
- `-classpath`：Jar包路径，由于已经引用了资源，此处路径统一为./下的Jar包即可。
- `com.aliyun.odps.mapred.open.example.WordCount`：执行过程调用Jar中的主类，需与Jar中的主类名称保持一致。
- `jingyan_wc_in`：MR的输入表名称，已在上述代码中提前建立好。
- `jingyan_wc_out`：MR的输出表名称，已在上述代码中提前建立好。
- 一个MR调用多个Jar资源时，classpath写法为`-classpath ./xxxx1.jar,./xxxx2.jar`，即两个路径之间用英文逗号分隔。

 说明：

在ODPS MR节点中调用参数时，请参考SHELL节点调用方式。

4. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

 说明：

您也可以直接点击运行按钮测试运行。

5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

   具体操作请参见#unique_314。

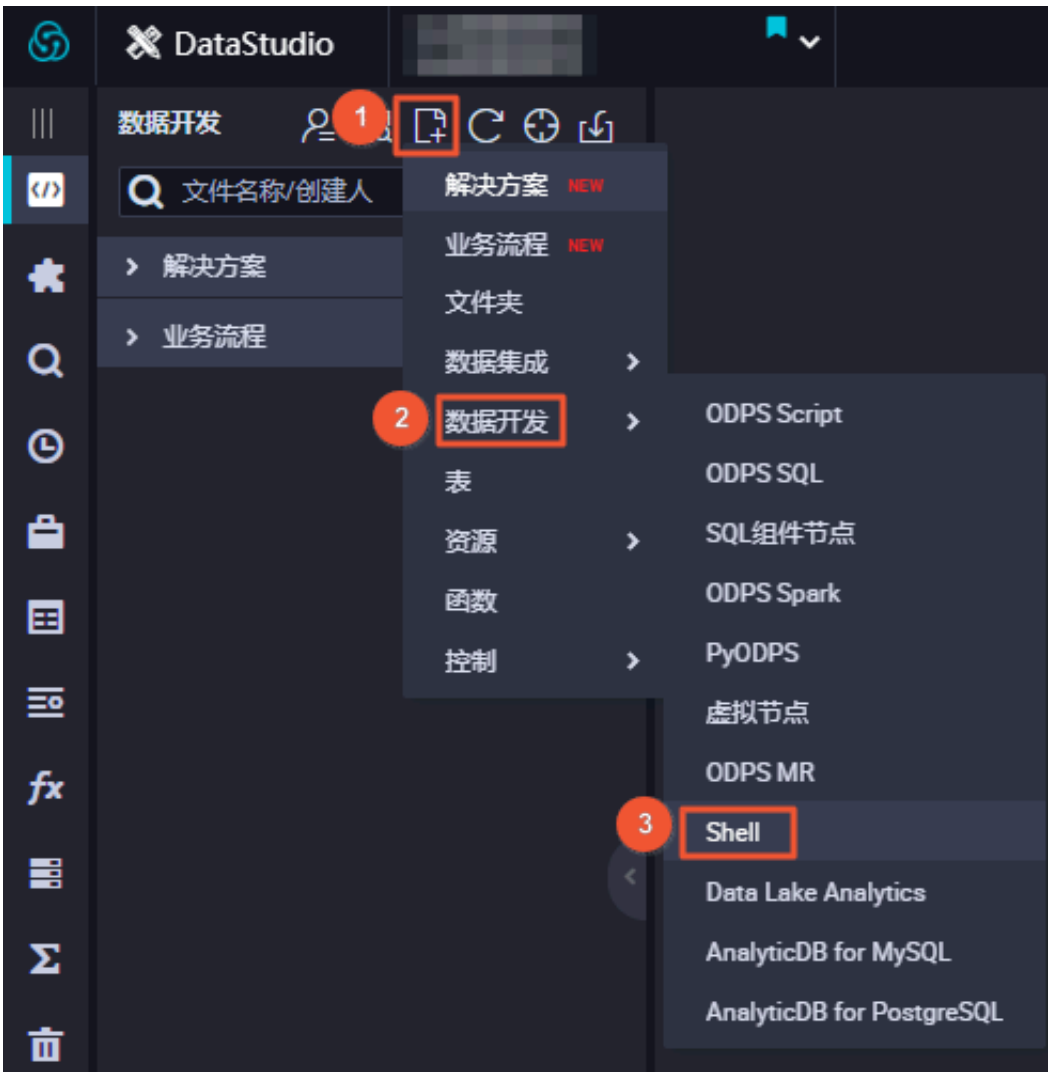
3.5.9 Shell节点

Shell节点支持标准Shell语法，不支持交互性语法。

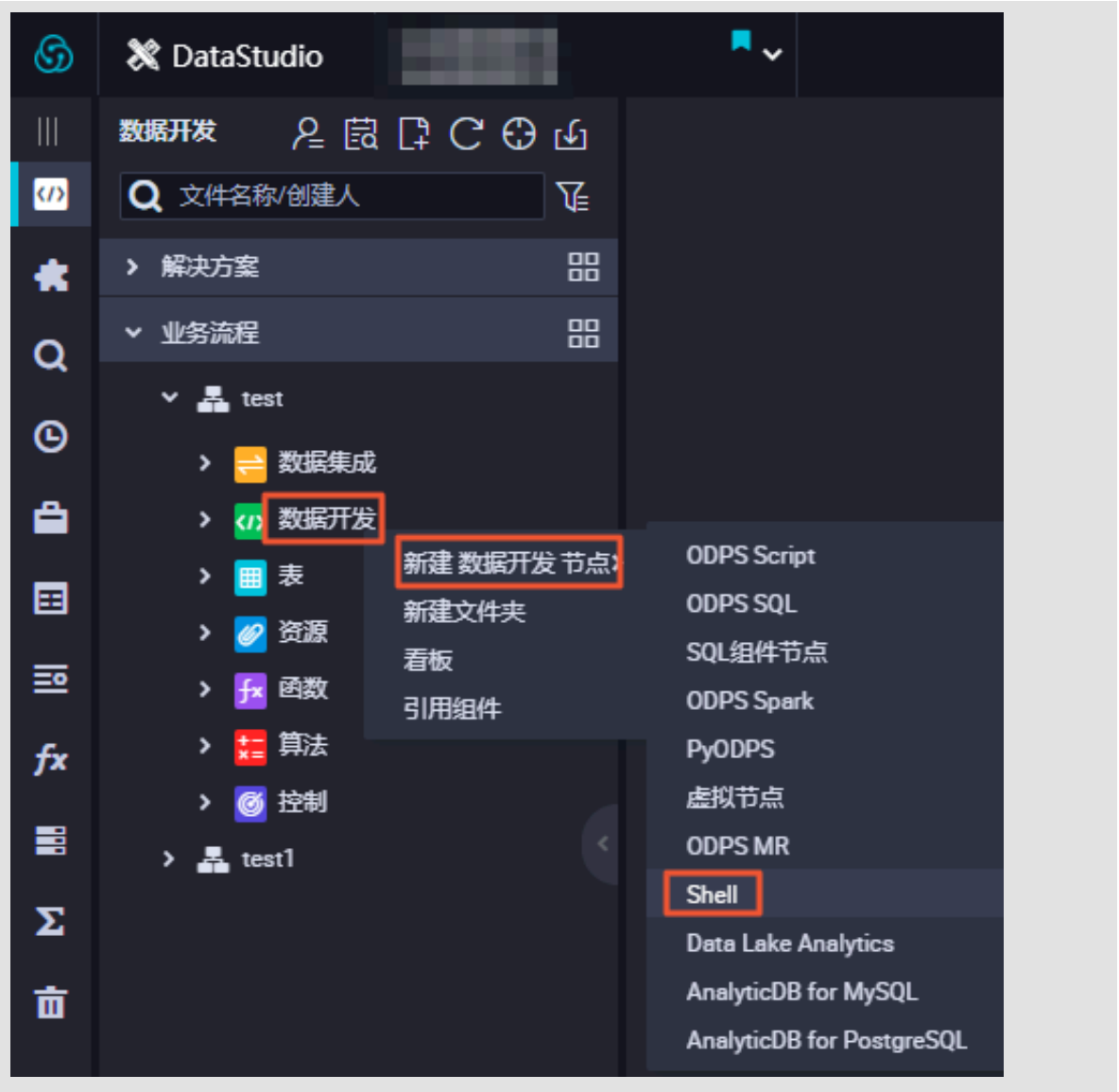
Shell节点任务可以在默认资源组上运行，如果您想要访问IP/域名，需要将IP/域名添加到白名单中。

新建Shell节点

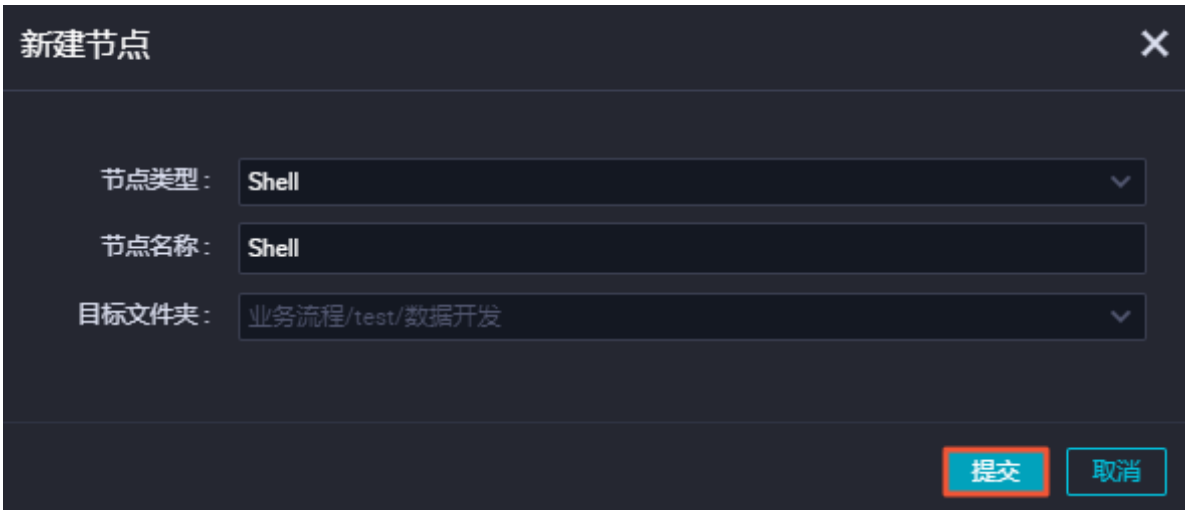
1. 进入DataStudio（数据开发）页面，选择新建 > 数据开发 > Shell。



> **说明:**
> 您也可以打开相应的业务流程，右键单击数据开发，选择新建数据开发节点 > Shell。

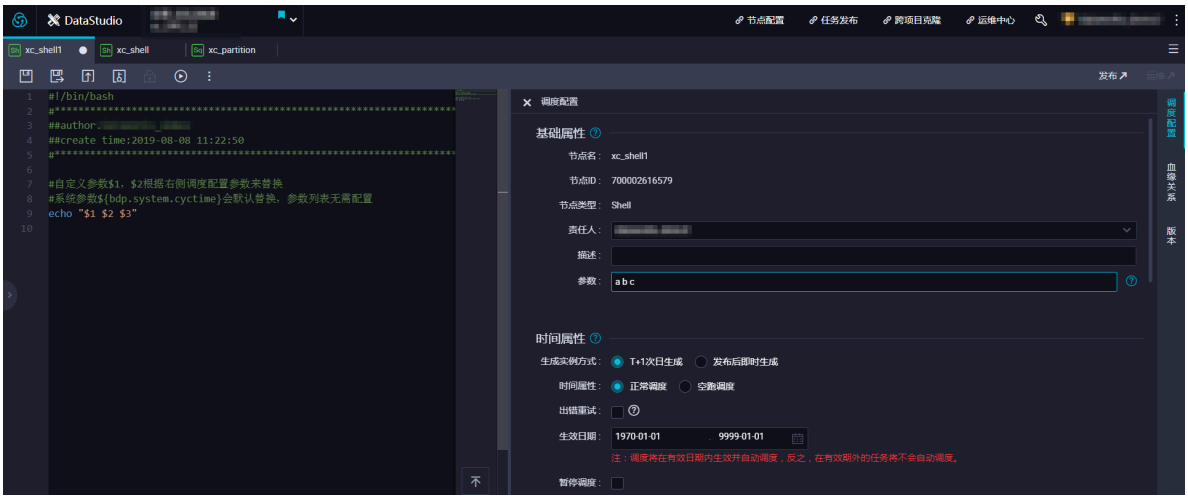


2. 填写新建节点对话框中的配置，单击提交。



3. 编辑节点代码。

   进入Shell节点代码编辑页面编辑代码。



   如果需要在Shell中调用系统调度参数，Shell语句如下所示：

```
echo "$1 $2 $3"
```

> **说明:**
> 参数1 参数2…多个参数之间用空格分隔。更多系统调度参数的使用，请参见#unique_39。

4. 节点调度配置。

   单击节点编辑区域右侧的调度配置，即可进入调度配置页面，详情请参见调度配置模块。

5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

   具体操作请参见#unique_314。

3.5.10 PyODPS节点

DataWorks提供PyODPS节点类型，集成了MaxCompute的Python SDK。您可以在DataWorks的PyODPS节点上，直接编辑Python代码，用于操作MaxCompute。

MaxCompute提供了#unique_304，您可以使用Python的SDK来操作MaxCompute。

> **说明:**
> PyODPS节点底层的Python版本为2.7。

PyODPS节点主要针对MaxCompute的Python SDK应用。对于纯Python代码的执行，您可以使用Shell节点执行上传至DataWorks的py脚本。

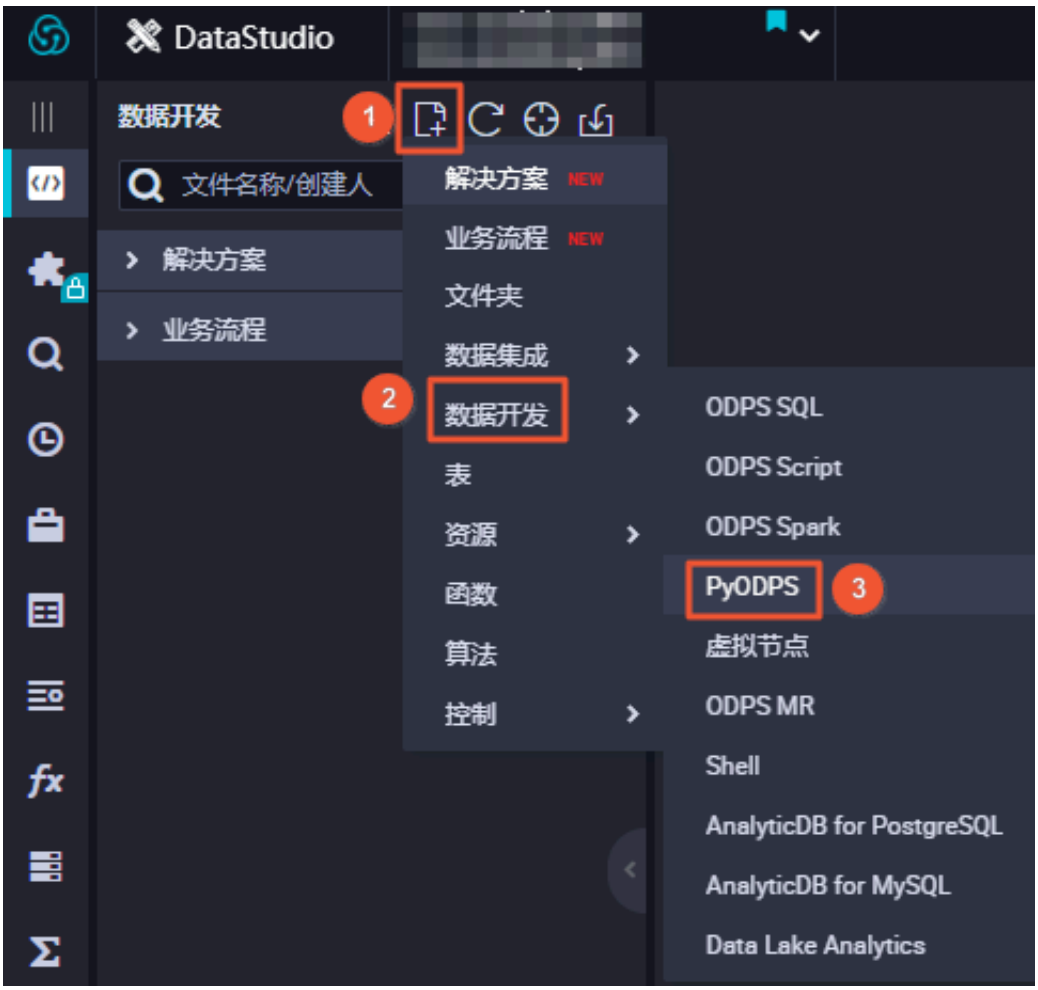
推荐通过SQL或者Dataframe的方式处理数据，详情请参见#unique_328。不建议您直接调用pandas等第三方包来处理数据。

PyODPS节点获取到本地处理的数据不能超过50MB，节点运行时占用的内存不能超过1G，否则节点任务会被系统Kill。请避免在PyODPS节点中写入过多的数据处理代码。

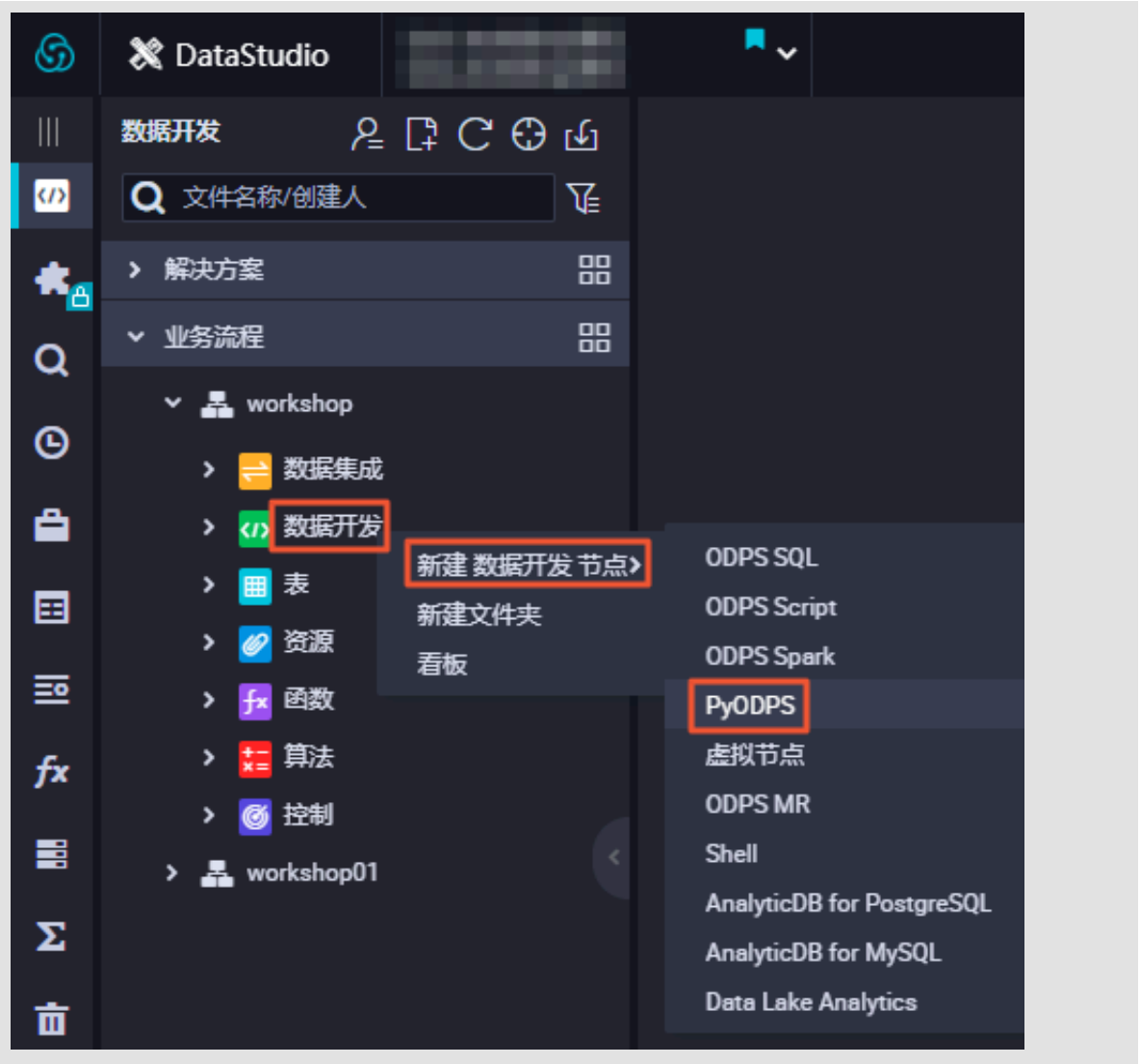
PyODPS操作实践请参见#unique_329和#unique_330，更多信息请参见PyODPS文档。

新建PyODPS节点

1. 进入数据开发，选择新建 > 数据开发 > PyODPS。



说明:
您也可以找到相应的业务流程，右键单击数据开发，选择新建数据开发节点 > PyODPS。



2. 填写新建节点对话框中的配置，单击提交。

3. 编辑PyODPS节点。

   a. ODPS入口。

   DataWorks的PyODPS节点中，将会包含一个全局的变量odps或o，即ODPS入口，您不需要手动定义ODPS入口。

   ```
   print(odps.exist_table('PyODPS_iris'))
   ```

   b. 执行SQL。

   PyODPS支持ODPS SQL的查询，并可以读取执行的结果。execute_sql或run_sql方法的返回值是运行实例。

    说明:

   并非所有在MaxCompute客户端中可以执行的命令，都是PyODPS支持的SQL语句。调用非DDL/DML语句时，请使用其他方法。

   例如，执行GRANT/REVOKE等语句时，请使用run_security_query方法。PAI命令请使用run_xflow或execute_xflow方法。

   ```
   o.execute_sql('select * from dual')  #  同步的方式执行，会阻塞直到SQL执行完成。
   instance = o.run_sql('select * from dual')  # 异步的方式执行。
   print(instance.get_logview_address())  # 获取logview地址。
   instance.wait_for_success()  # 阻塞直到完成。
   ```

   c. 设置运行参数。

   您可以通过设置hints参数，来设置运行时的参数，参数类型是dict。

   ```
   o.execute_sql('select * from PyODPS_iris', hints={'odps.sql.mapper.split.size': 16})
   ```

   对全局配置设置sql.settings后，每次运行时，都需要添加相关的运行时的参数。

   ```
   from odps import options
   options.sql.settings = {'odps.sql.mapper.split.size': 16}
   o.execute_sql('select * from PyODPS_iris')  # 根据全局配置添加hints。
   ```

   d. 读取SQL执行结果。

   运行SQL的instance能够直接执行open_reader的操作，有以下两种情况：

   · SQL返回了结构化的数据。

   ```
   with o.execute_sql('select * from dual').open_reader() as reader:
   ```

```
for record in reader:  # 处理每一个record。
```

- 可能执行的是desc等SQL语句，通过reader.raw属性，获取到原始的SQL执行结果。

```
with o.execute_sql('desc dual').open_reader() as reader:
print(reader.raw)
```

> 说明:
> 如果使用了自定义调度参数，页面上直接触发运行PyODPS节点时，需要写死时间，PyODPS节点无法直接替换。

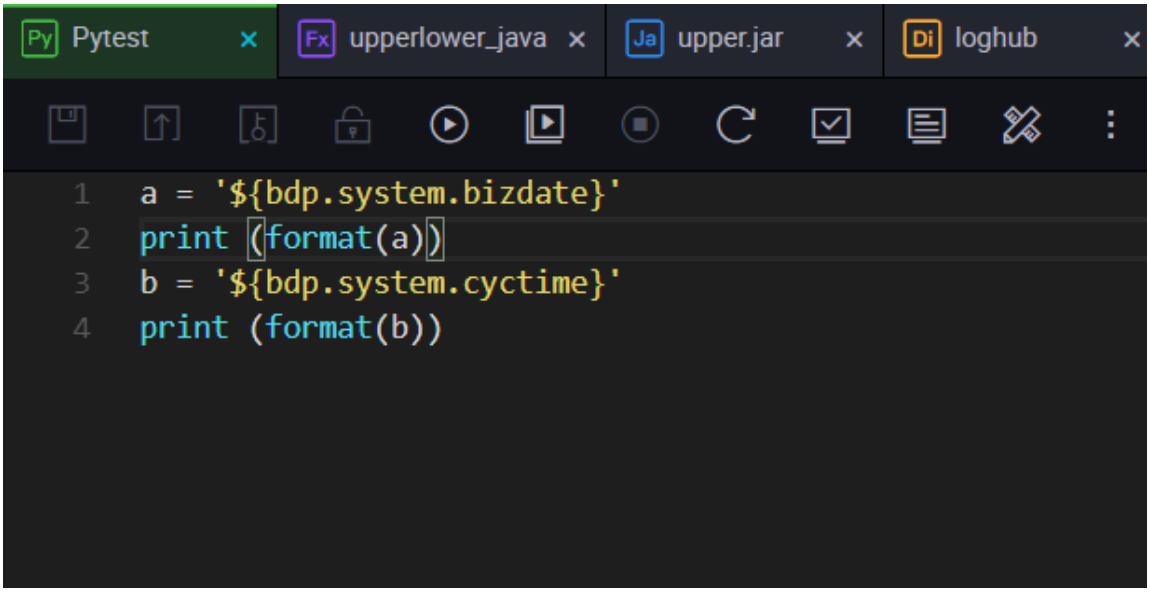
4. PyODPS节点参数与调度配置。

   a. 单击右侧导航栏中的调度配置，即可弹出调度配置对话框。

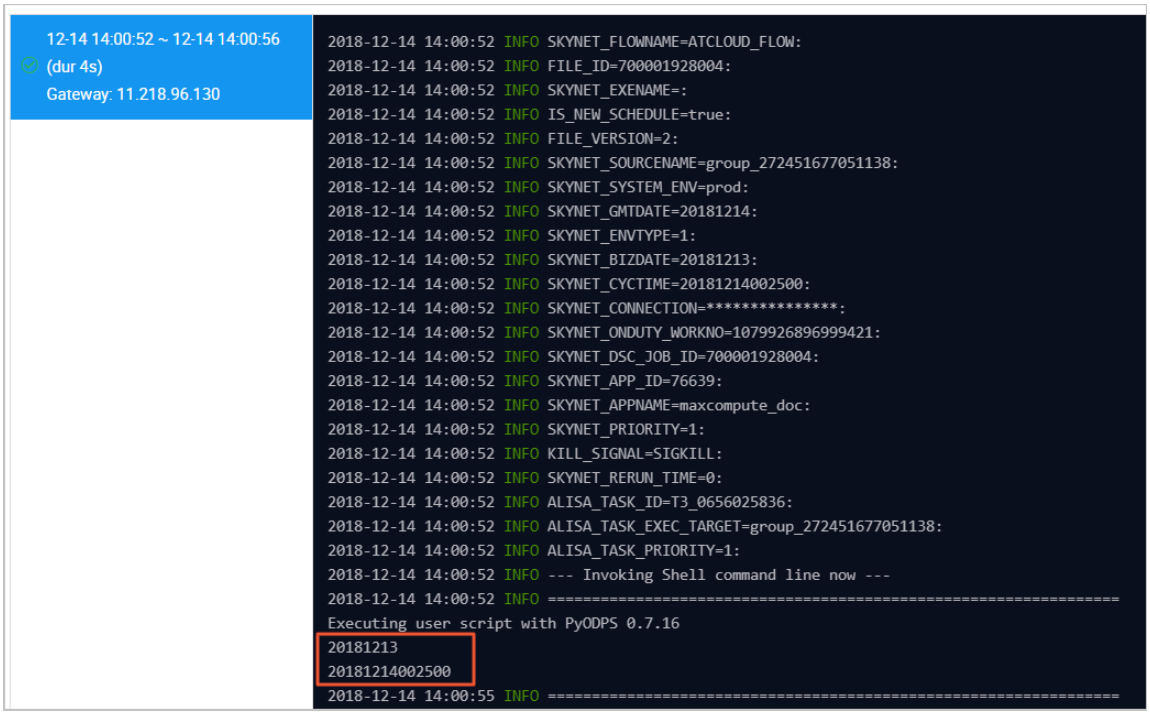
   PyODPS节点使用调度参数时，如果使用系统定义的调度参数，可以直接在页面赋值获取。

   > 说明:

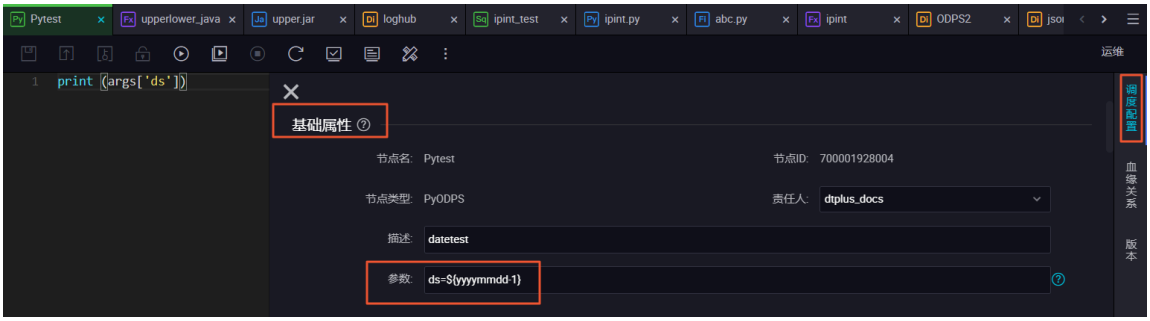
由于默认资源组无法直接访问外网环境，建议您有公网访问需求时，使用自定义资源组或独享调度资源。仅专业版DataWorks提供自定义资源组，任意版本均可购买独享调度资源，详情请参见#unique_18。



b. 赋值完成后，提交节点并进入运维中心页面，进行测试运行，即可查看赋值结果。



c. 您可以在调度配置 > 基础属性中，配置自定义参数。



说明:

自定义参数需要使用args['参数名']的形式调用，例如print (args['ds'])。

d. 完成配置后，提交节点并进入运维中心页面，进行测试运行，即可查看赋值结果。

5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）至开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

   具体操作请参见#unique_314。

PyODPS节点预装模块列表

PyODPS节点包括以下预装模块：

- setuptools
- cython
- psutil
- pytz
- dateutil
- requests
- pyDes
- numpy
- pandas

- scipy
- scikit_learn
- greenlet
- six
- 其他Python 2.7内置已安装的模块，如smtplib等。

3.5.11 遍历（for-each）节点

本文将为您介绍如何通过for-each节点实现循环2次，每次循环中把当前的循环次数打印出来的需求。

说明:

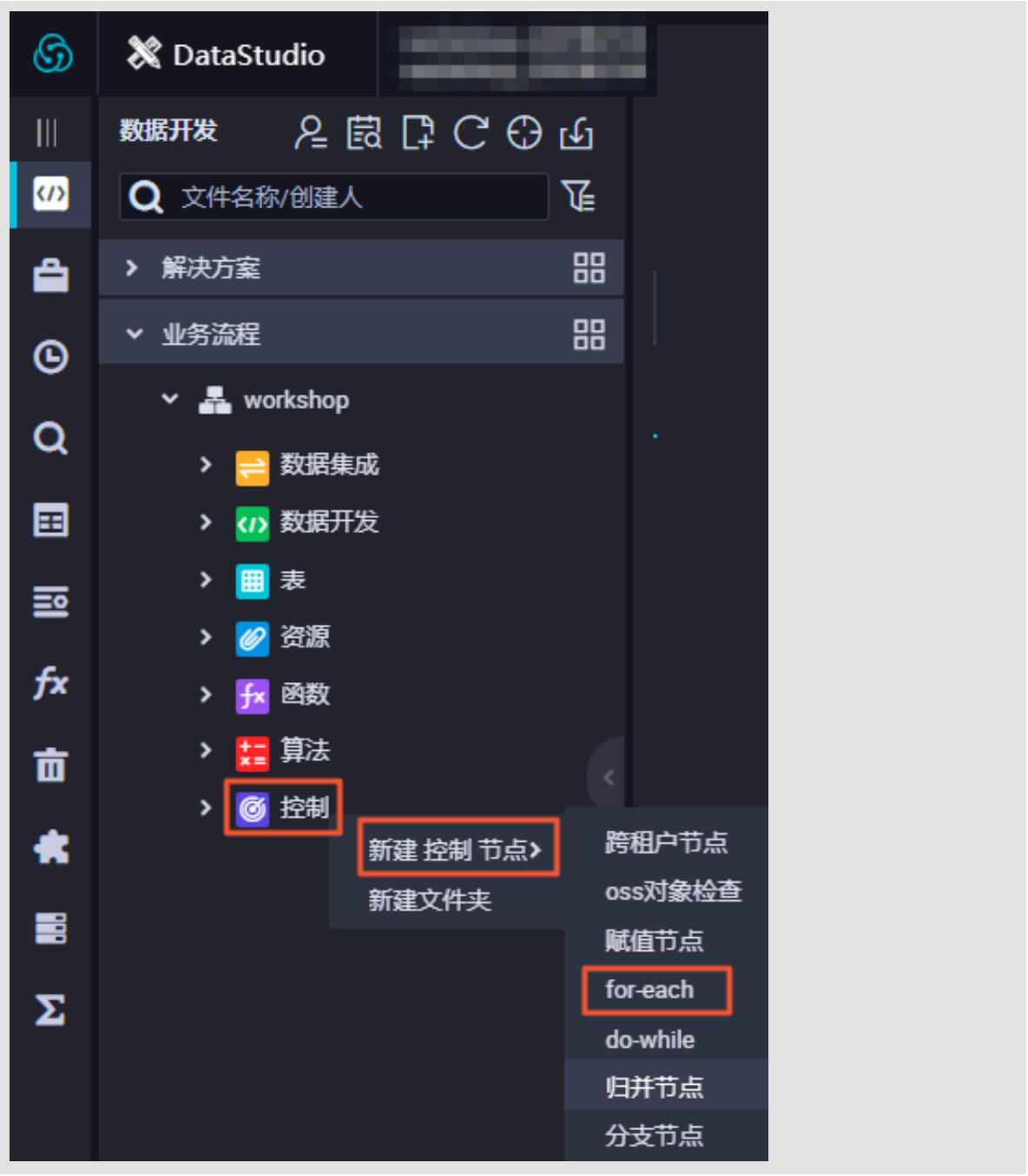
您需要购买DataWorks标准版及以上版本，方可使用for-each节点功能。

创建工作流

1. 进入DataStudio（数据开发）页面，选择新建 > 控制 > for-each。

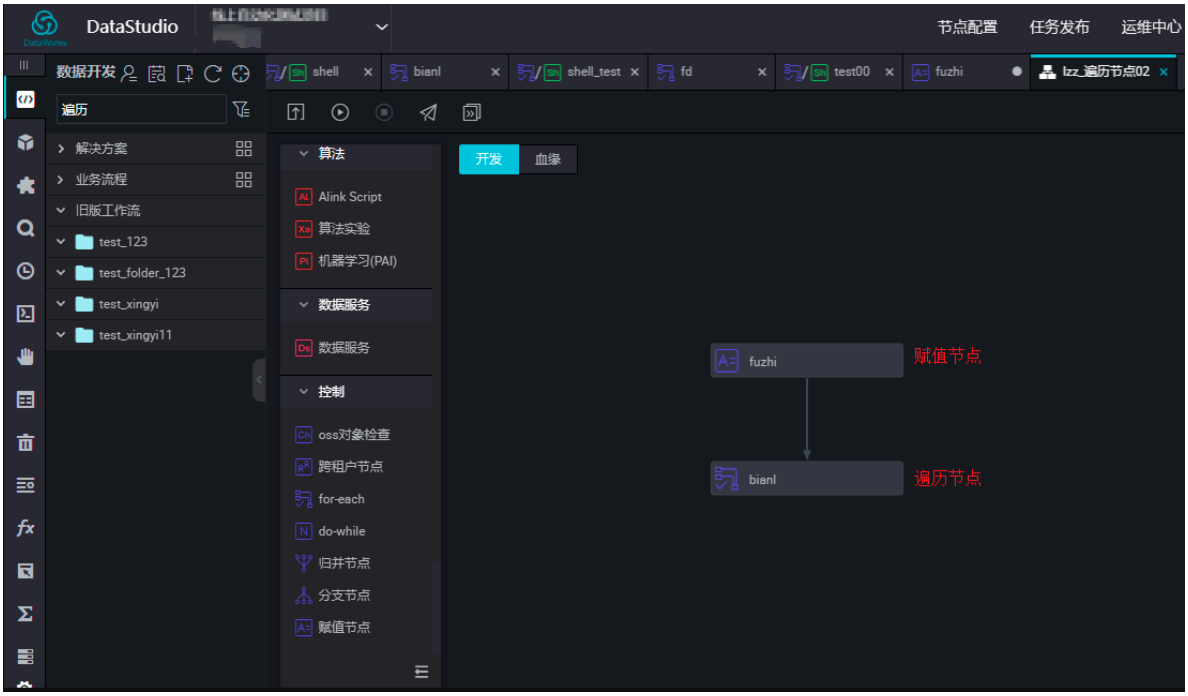
说明:

您也可以找到相应的业务流程，右键单击控制，选择新建控制节点 > for-each。


DataStudio
数据开发
文件名称/创建人
解决方案
业务流程
workshop
数据集成
数据开发
表
资源
函数
算法
控制
新建 控制 节点
新建文件夹
跨租户节点
oss对象检查
赋值节点
for-each
do-while
归并节点
分支节点

2. 填写新建节点对话框中的配置，单击提交。

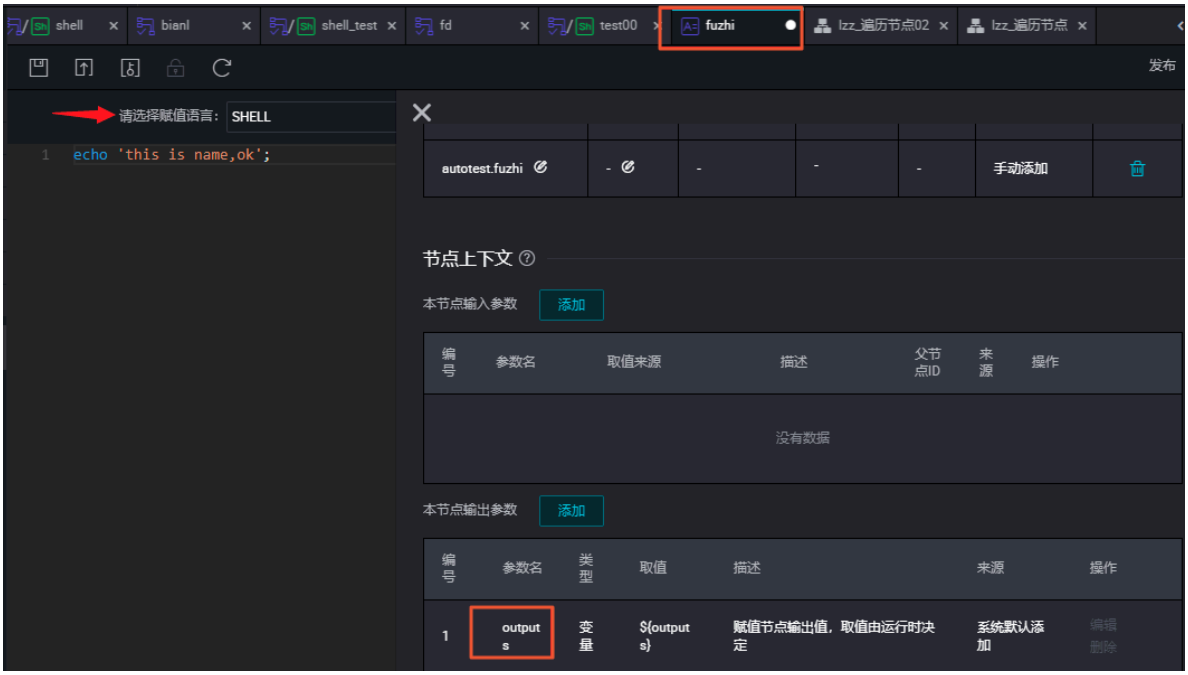
3. 创建的工作流上游为赋值节点，下游为遍历节点。



上游赋值节点选择的是Shell节点，代码如下。

```
echo 'this is name,ok';
```

赋值节点默认一个outputs参数。



编辑遍历节点

> **说明:**

- 遍历节点的start节点和end节点不能编辑，其逻辑是固定的。
- Shell节点中的代码修改后一定要保存，提交时不会提示您进行保存，如果没有保存，最新的代码便不能及时更新。

Shell节点的代码为：

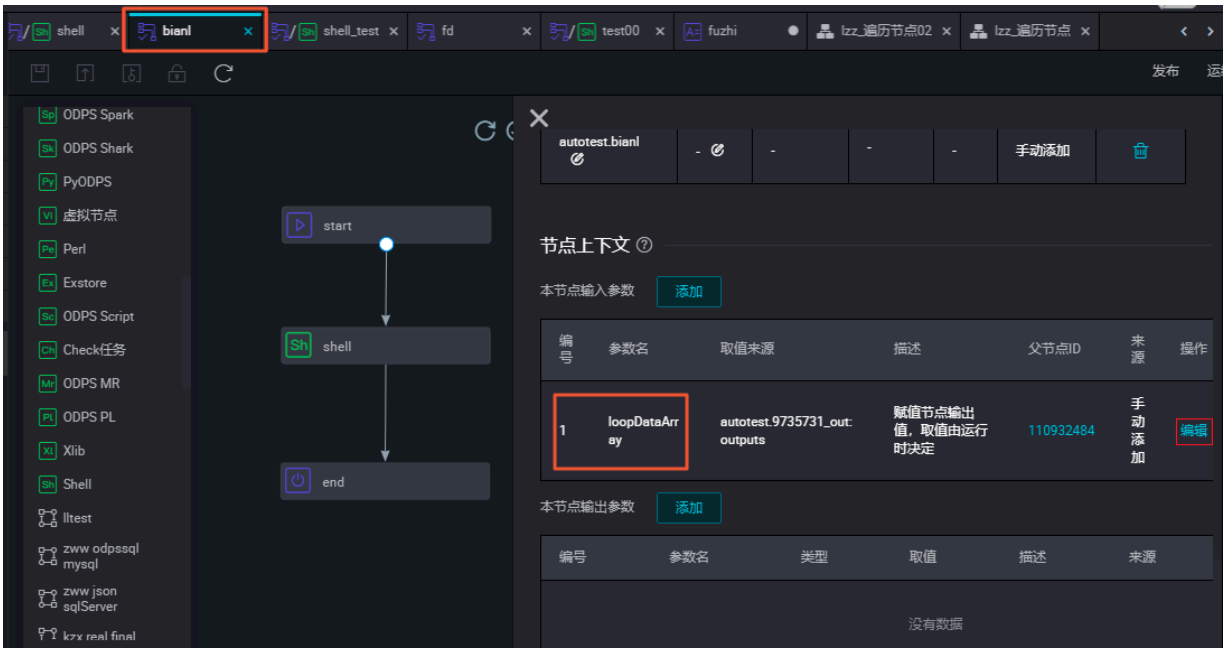
```
echo ${dag.loopTimes}  ----打印循环的次数。
```

遍历节点支持以下四种环境变量。

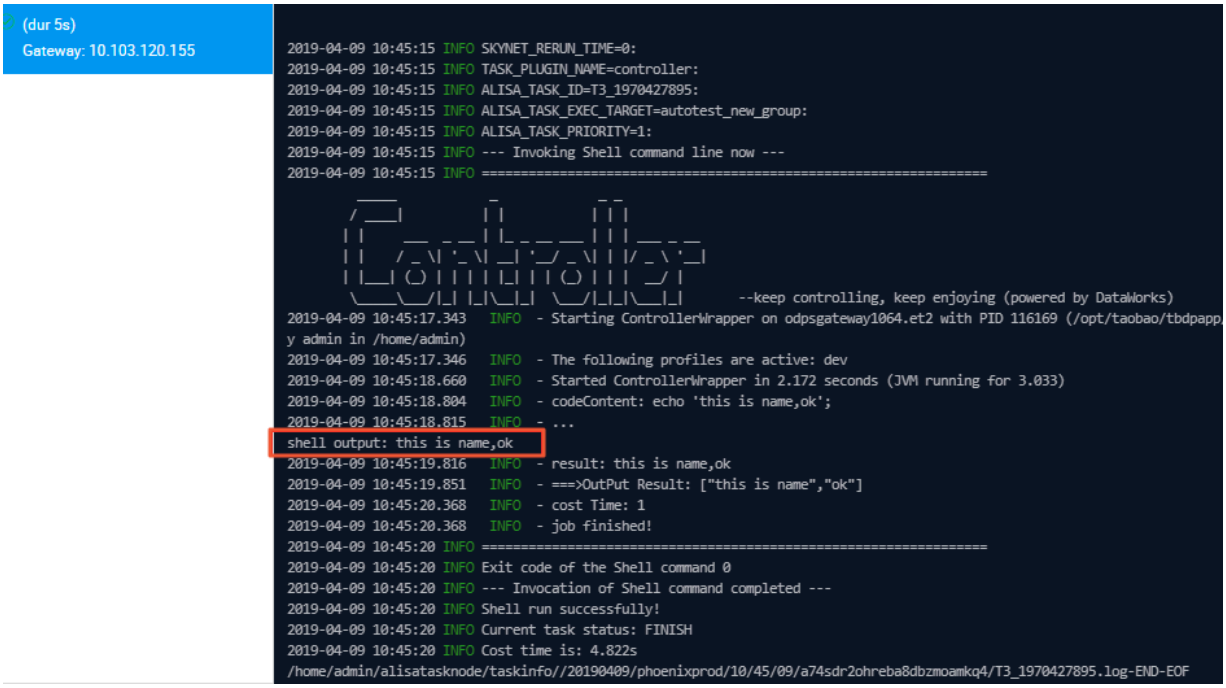
- ${dag.foreach.current}：当前遍历到的数据行。
- ${dag.loopDataArray}：输入的数据集。
- ${dag.offset}：偏移量。
- ${dag.loopTimes}：当前循环次数，值为${dag.offset}+1。

```
// 以常见的for循环代码进行类比
data=[]  // 相当于${dag.loopDataArray}
// i相当于${dag.offset}
for(int i=0;i<data.length;i++) {
  print(data[i]);  // data[i]相当于${dag.foreach.current}
}
```

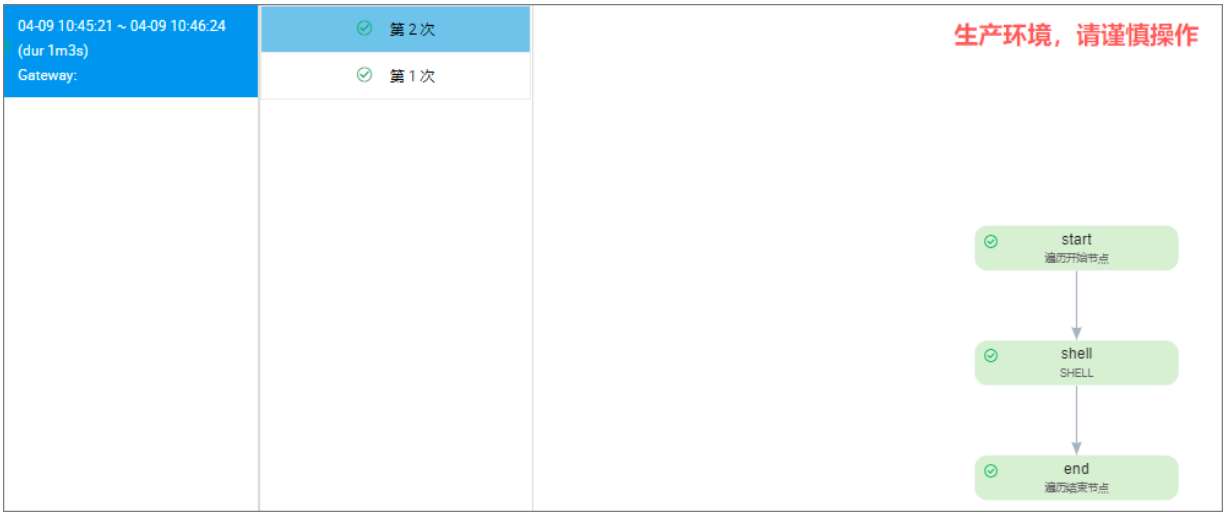
遍历节点默认参数名为loopDataArray，取值来源是上游赋值节点的outputs参数，需要手动添加，如果没有添加，提交时会报错。

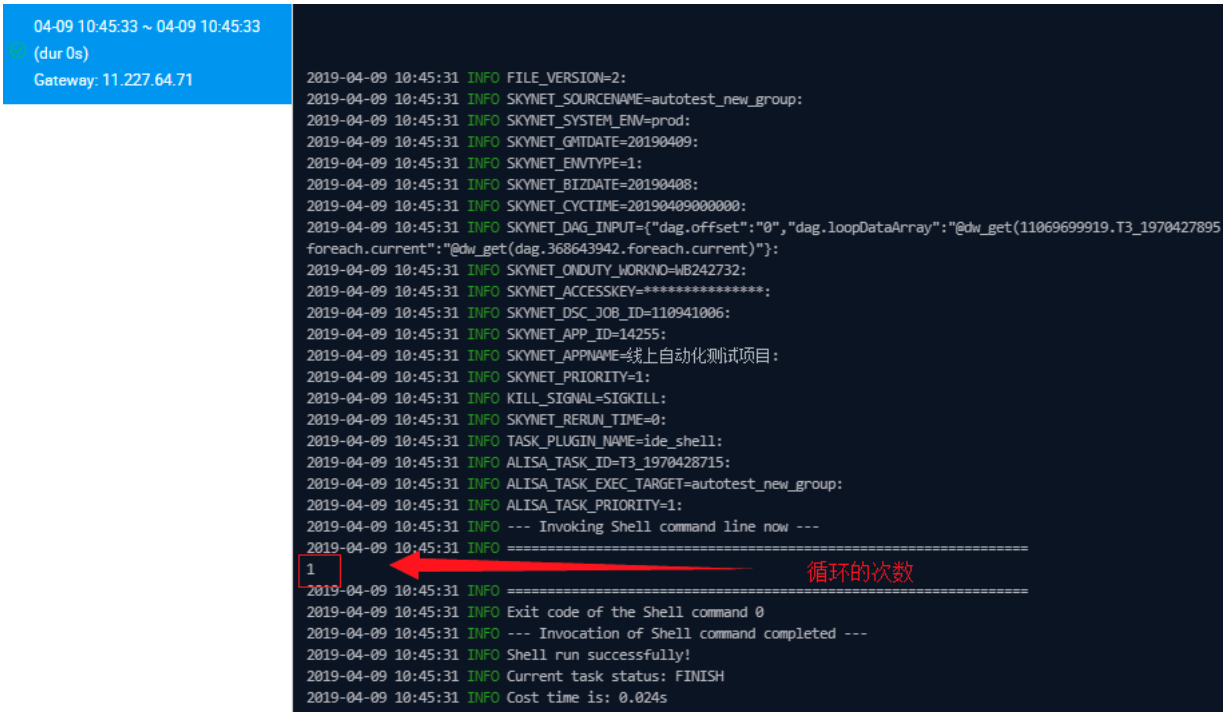


提交后发布，进入运维中心页面查看相应结果。



遍历节点的结果如下图所示。





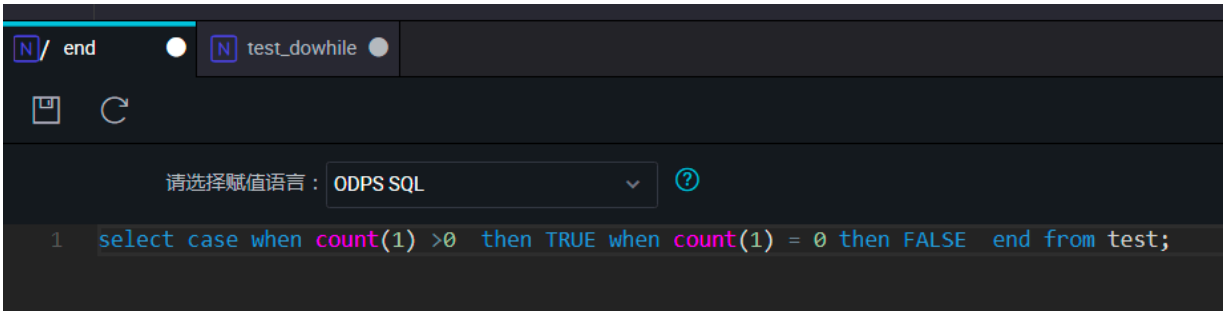
3.5.12 循环（do-while）节点

您可以在do-while节点中定义相互依赖的任务，任务中包含一个名为end的循环判断节点。Dataworks会不断重复执行该批任务，直至循环判断节点end把判断结果置为false，Dataworks才会退出整个循环。

说明:

- 您需要购买DataWorks标准版及以上版本，方可使用do-while节点功能。
- 循环节点最多可以循环128次，一旦超过便会报错。

循环（do- while）节点支持ODPS SQL、SHELL和Python三种赋值语言。选择ODPS SQL赋值语言时，您可以通过`case when`语句进行判断。



循环节点的简单示例

本节将为您介绍使用循环节点循环5次，每次循环中把当前的循环次数打印出来的简单场景。

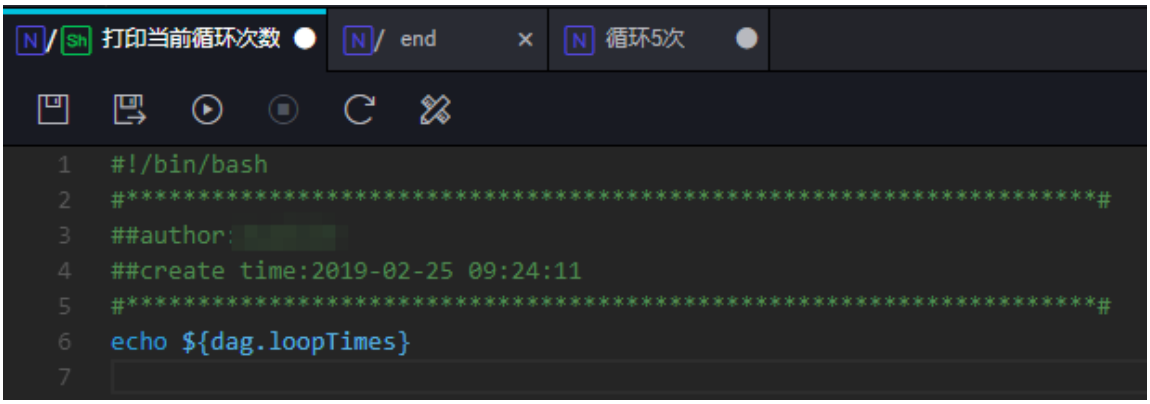
1. 进入数据开发页面，选择新建 > 控制 > do-while。

2. 填写新建节点对话框中的配置，单击提交。

3. 双击新建的do-while节点，对循环体进行定义。

双击进入do-while节点内部时，会自带start-sql-end三个节点。

- start节点是一个循环开始的标记节点，并无业务作用。
- SQL节点是Dataworks提供的一个业务处理节点示例，此处需要将其删除，替换为自己的业务处理Shell节点（打印当前循环次数）。



- end节点有标记循环结束和判断是否开启下一次循环两大功能，此处对do-while节点的结束条件进行定义。

  end节点本质上是一个赋值节点，仅输出true/false两种字符串，分别代表继续下一个循环和不再继续循环。

在打印循环次数和end节点中，都用到了`${dag.loopTimes}`变量。它是系统的保留变量，代表当前的循环次数，从1开始，do-while的内部节点可以直接引用该变量。

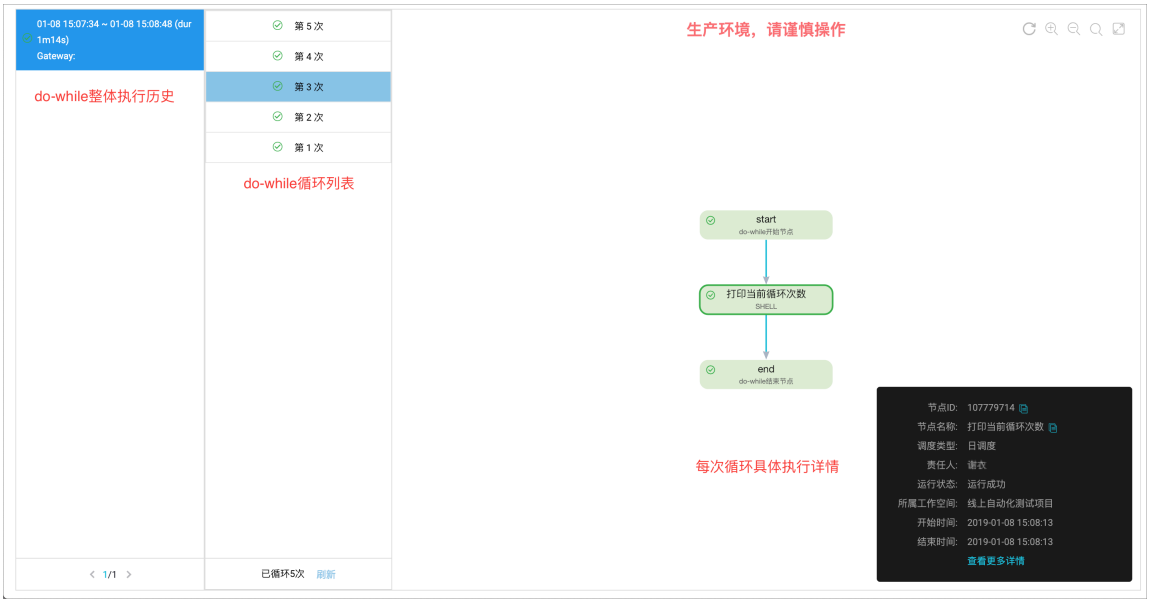
代码中把`dag.loopTimes`和5进行比较，可以限制整体的循环次数。第一次循环`dag.loopTimes`为1、第二次为2，以此类推，第五次为5。至此表达式${dag.loopTimes}<5结果为false，退出循环。

4. 执行do-while节点。

您可以根据自身需求进行调度配置后，将do-while节点提交至运维中心执行。

- 外层do-while节点：do-while节点整体在运维中心中被当作一个整体节点来展示。如果您想要查看do-while节点的循环详情，可以右键单击节点，选择查看内部节点，即可跳转至内部节点视图。

- 内部循环体：该视图分三个部分。

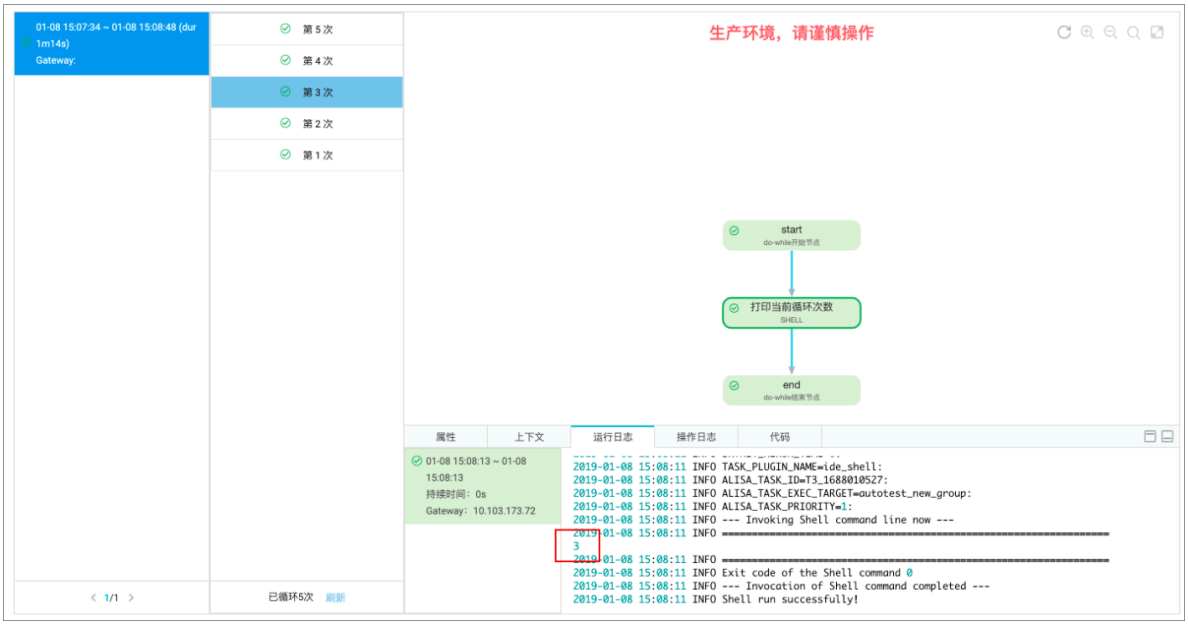


  - 视图左侧为do-while节点的重跑历史列表，只要do-while实例整体执行一次，历史列表便会产生一条相应的记录。

- 视图中部为循环记录列表，会列出当前do-while节点共执行多少次循环，以及每次循环的状态。
- 视图右侧为每次循环的具体信息，单击循环记录列表中的某次循环，即可展示出该循环每个实例的执行情况。

5. 查看运行结果。

进入内部循环体后，单击循环记录列表中的第3次循环，即可看到执行日志中打印出本次的循环次数3。



您也可以查看第3次循环和第5次循环中，end节点的执行日志。

图 3-1: 第3次循环end节点的执行日志

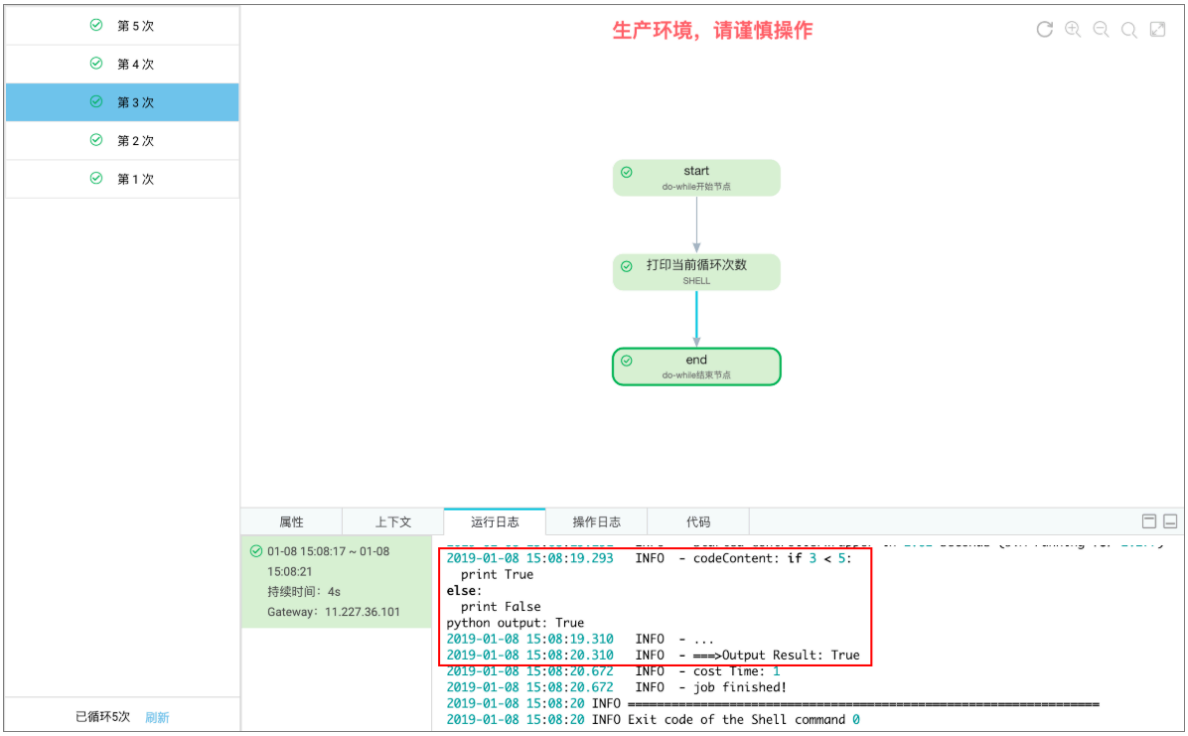
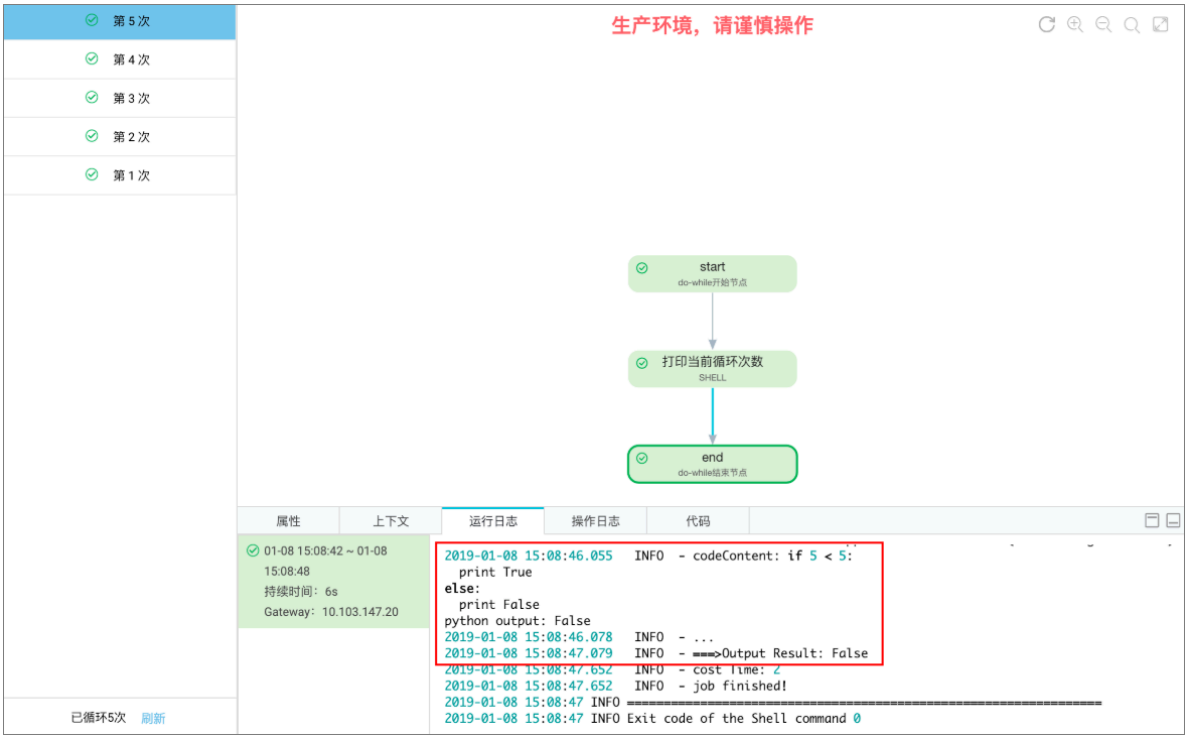


图 3-2: 第5次循环end节点的执行日志



由上面两张end节点日志图可见，第3次循环的结果是3<5 => True，第5次循环的结果是5<5 => False。所以在第5次执行完毕后，退出循环。

由上述简单示例可总结出do-while节点的工作流程如下：

1. 从start节点开始执行。
2. 按照定义的任务依赖关系依次执行每个任务。
3. 在end节点中定义循环的结束条件。
4. 一组任务执行完毕之后，执行end的结束条件语句。
5. 如果end的判断语句在日志中打印true，则从1开始继续下一个循环。
6. 如果end的判断语句在日志中打印false，则退出整个循环，do-while节点整体结束。

循环节点的复杂示例

除了上文中提到的简单场景外，还会遇到用循环的方式依次处理一组数据的每一行的复杂场景。处理此场景前，需要满足以下条件：

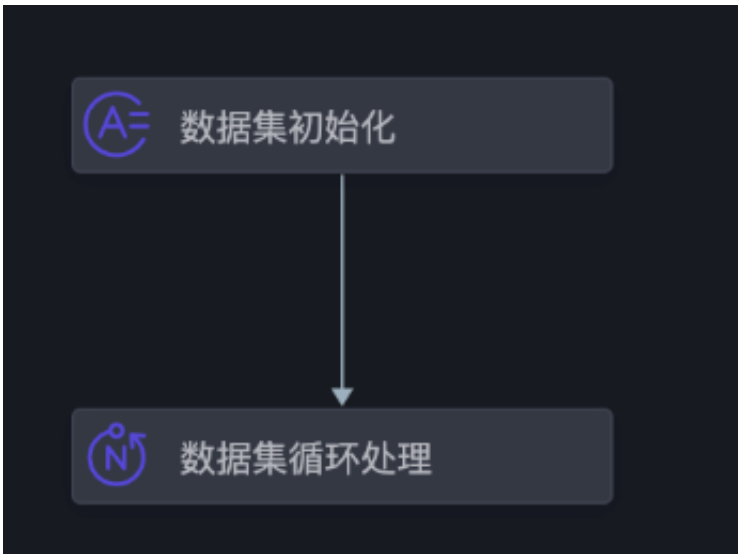
- 需要部署一个上游节点，能够把查询出的数据输出给下游节点使用，您可以使用赋值节点实现该条件。
- 循环节点需要能够拿到上游赋值节点的输出，您可以通过配置上下文依赖来实现该条件。
- 循环节点的内部节点需要能够引用到每一行的数据，增强已有的节点上下文，并额外下发了系统变量`${dag.offset}`，可以帮您快速引用循环节点的上下文。

下文将为您介绍如何实现用循环节点把表tb_dataset的每一行数据分别在每次循环中打印出来，数据为c、0、1。

1. 进入数据开发页面，选择新建 > 控制 > do-while。
2. 填写新建节点对话框中的配置，单击提交。

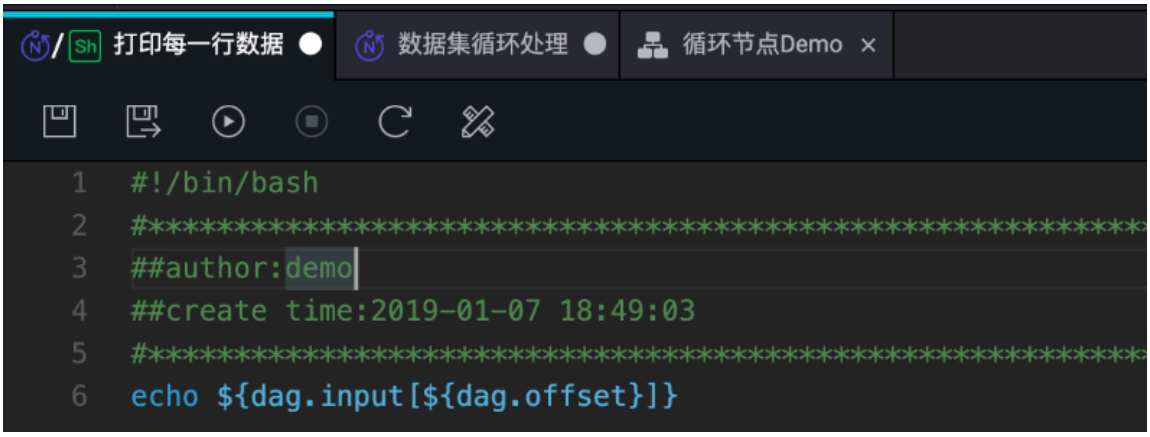
3. 双击新建的do-while节点，对循环体进行定义。

   a. 本示例需要给do-while节点添加一个上游节点（数据集初始化），它会产生测试数据集。



   b. 创建节点后，对do-while节点进行调度配置，为其单独配置一个节点上下文，key为input，值为上游节点数据集初始化的outputs。

c. 输入业务处理节点（打印每一行数据）的代码。



- `dag.offset`：Dataworks系统的保留变量，代表每一次循环次数相对于第一次的偏移量。即第一次循环中offset为0、第二次为1、第三次为2…第n次为n-1。
- `dag.input`：该变量是您配置的do-while上下文依赖，上文中提到do-while节点配置了一个叫input的上下文依赖，值为上游节点（数据集初始化）的outputs。

  do-while内部节点可以直接用`${dag.${ctxKey}}`的方式，引用上下文依赖的值。因为上文中配置的上下文依赖key为input，因此可以使用`${dag.input}`来引用该值。
- `${dag.input[${dag.offset}]}`：节点数据集初始化的输出是一个表格，Dataworks可以用偏移量的方式来获取表格数据的某一行。由于每次循环

中dag.offset的值从0开始递增，则最后打印出来的数据为${dag.input[0]}、${dag.input[1]}，以此类推达到遍历数据集的效果。

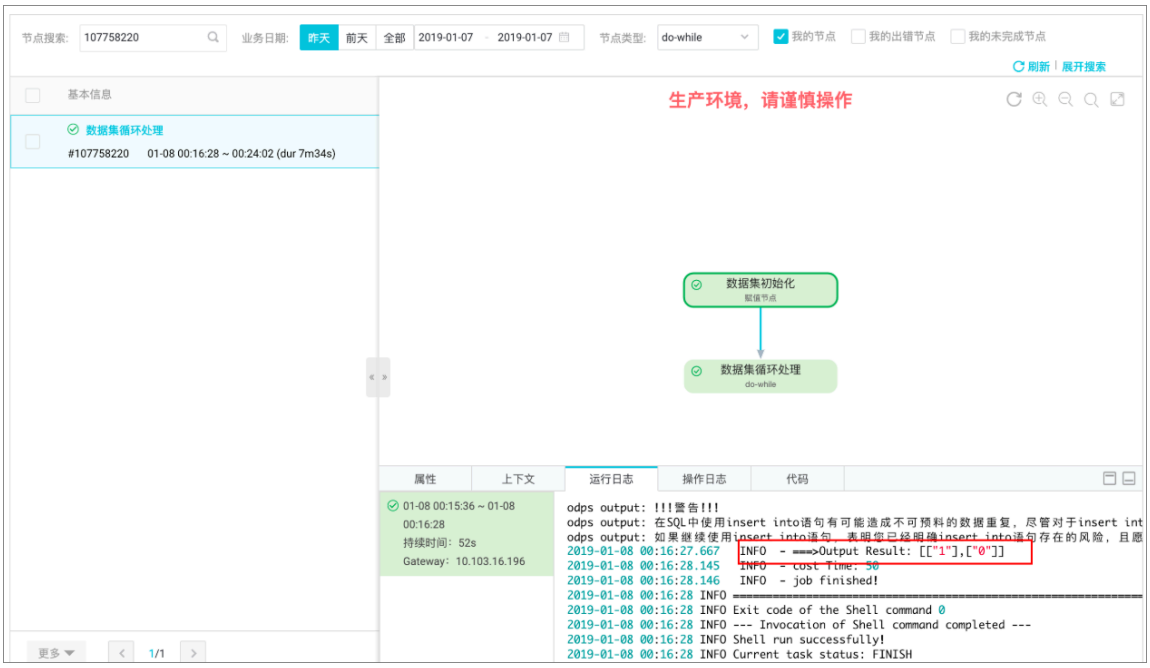
d. 定义end节点的结束条件。如下图所示，把dag.loopTimes与dag.input.length变量进行比较，小于则输出True继续循环，不小于则输出False退出循环。

说明:

dag.input.length变量标识的是上下文参数input数组的行数，是系统自动根据节点配置的上下文下发的变量。

4. 执行节点并查看执行结果。

   · 数据集初始化节点最终会产生0和1两条数据。



- 打印每一行数据节点执行结果如下所示。

图 3-3: 打印第1行数据

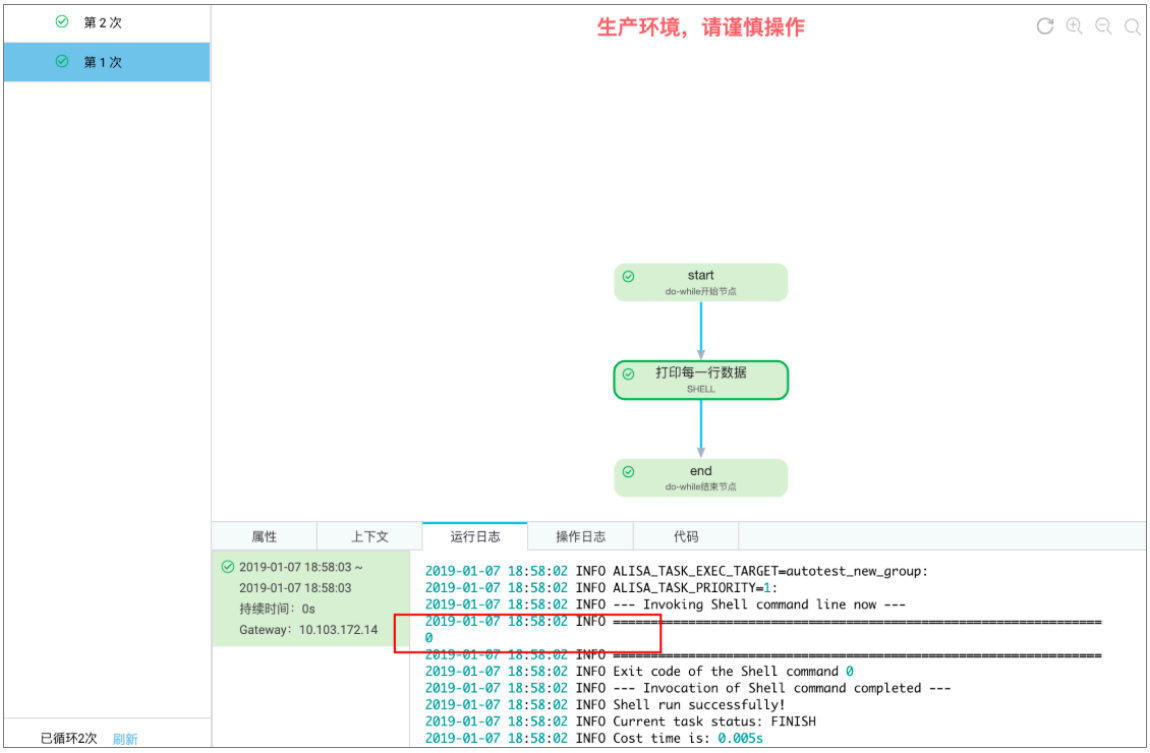
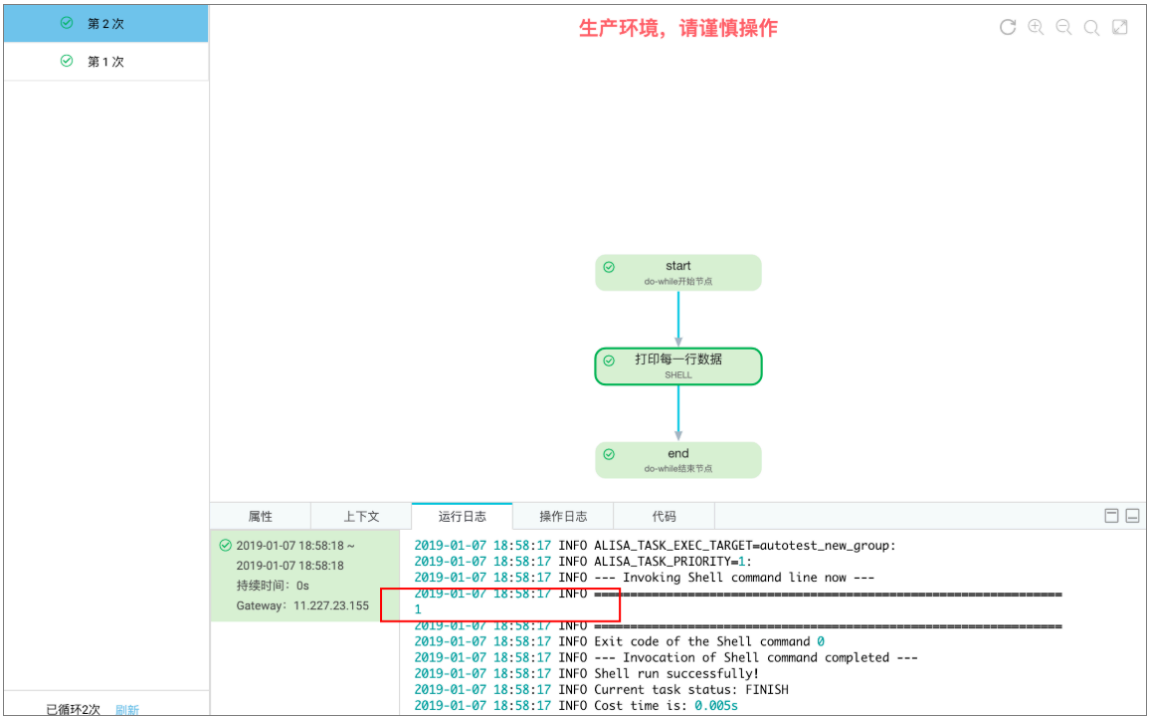


图 3-4: 打印第2行数据



- end节点的结果如下所示。

图 3-5: 第1次end执行结果

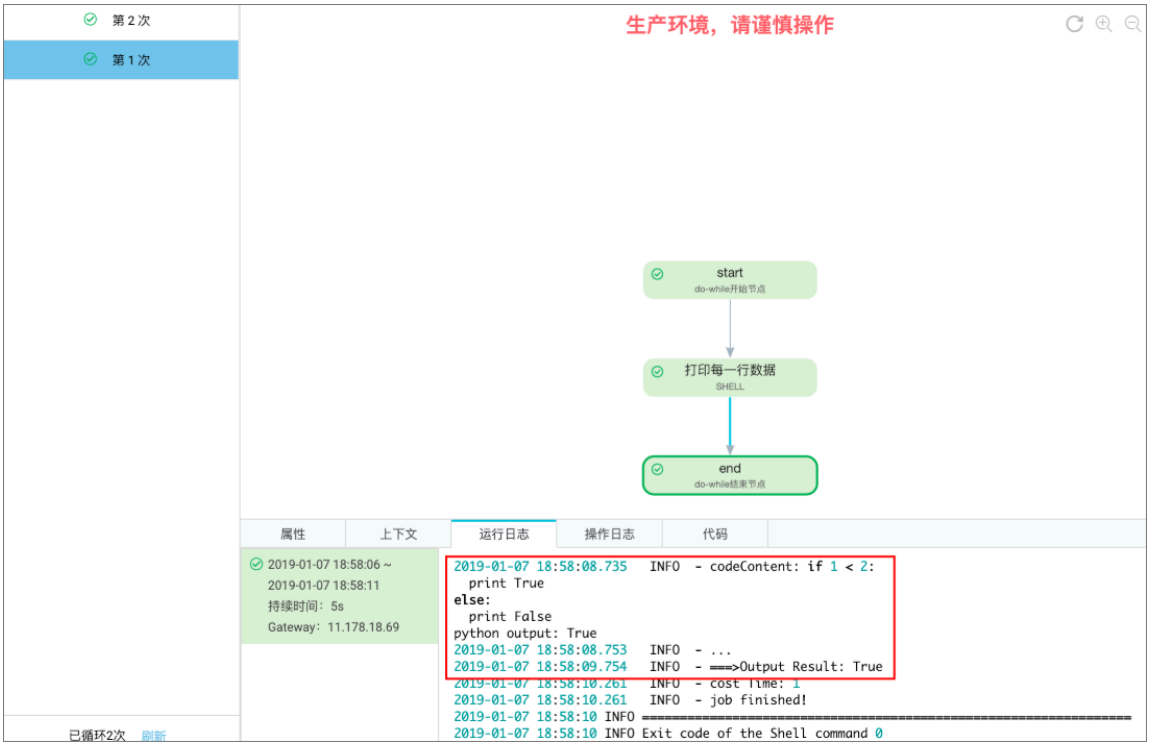
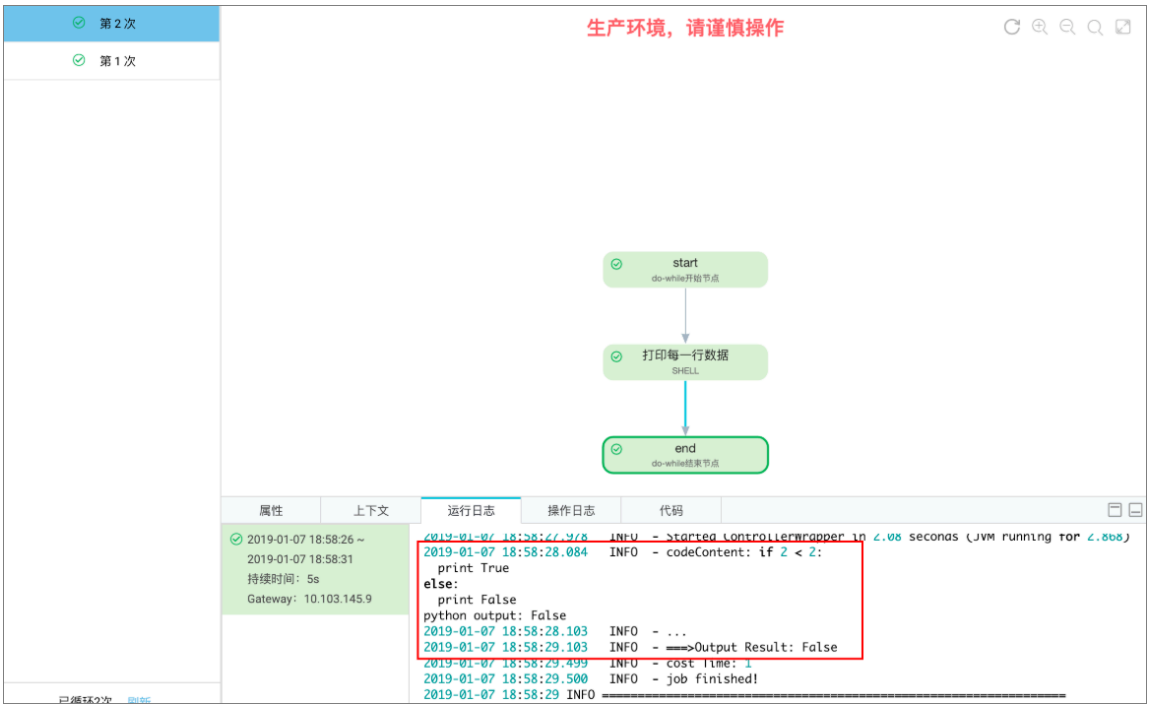


图 3-6: 第2次end执行结果



由上面两张end节点日志图可见，第1次循环的次数小于行数，返回True继续执行，第2次循环的次数等于行数，返回False停止循环。

总结

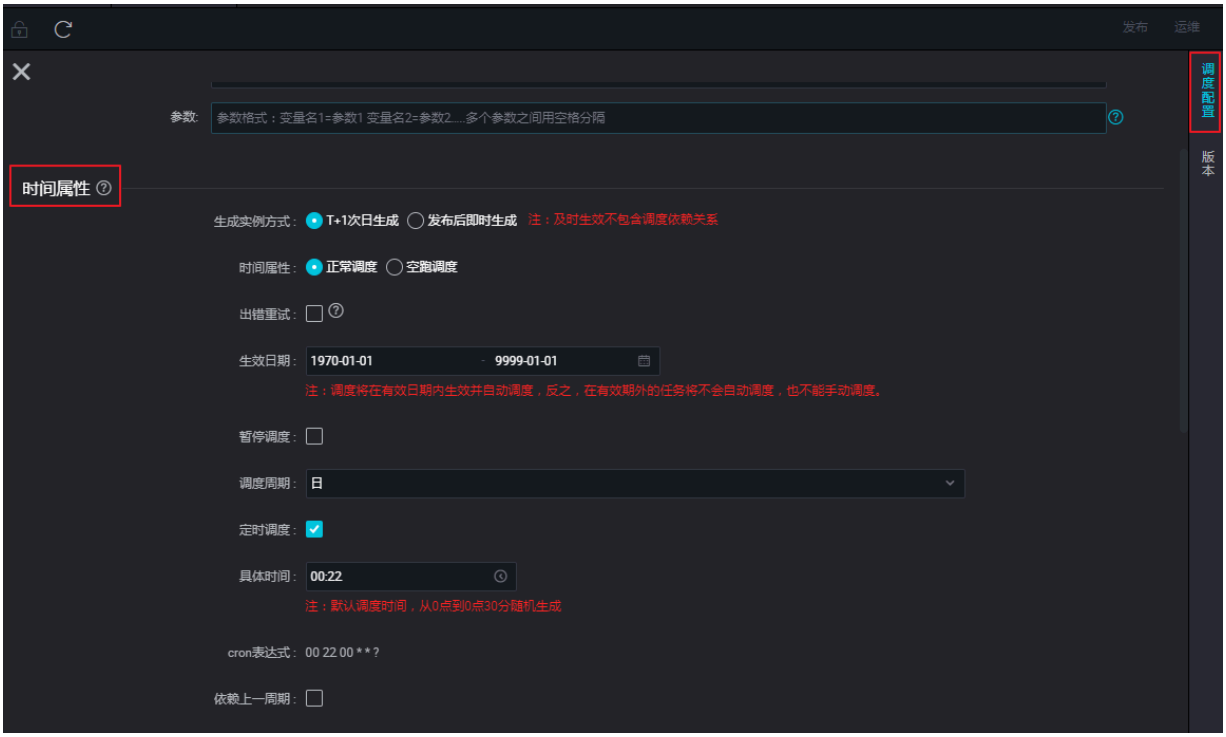
- do-while与while/foreach/do…while三种循环类型对比如下：
  - do-while能够实现先循环再判断的循环体，即do…while语句，能够通过系统的变量dag.offset结合节点上下文间接实现foreach语句。
  - do-while不能实现先判断再循环的方式，即while语句。
- do-while执行流程
  1. 从start开始按任务依赖关系依次执行循环体中的任务。
  2. 执行用户在end节点中定义的代码。
     - 如果end节点输出True，则继续下一个循环。
     - 如果end节点输出False，则终止循环。
- 如何使用上下文依赖：do-while的内部节点可以通过${dag.上下文变量名}的方式引用到do-while节点定义的节点上下文。
- 系统参数：Dataworks会为do-while内部节点自动下发两个系统变量。
  - dag.loopTimes从1开始标识这一次循环的次数。
  - dag.offset从0开始标识这一次循环相对于第一次循环的次数偏移量。

3.5.13 跨租户节点

跨租户节点主要用于不同租户的节点之间的联动，分为发送节点和接收节点。

使用前提

发送和接收节点的时间表达式必须一致。您可以在调度配置 > 时间属性查看表达式。

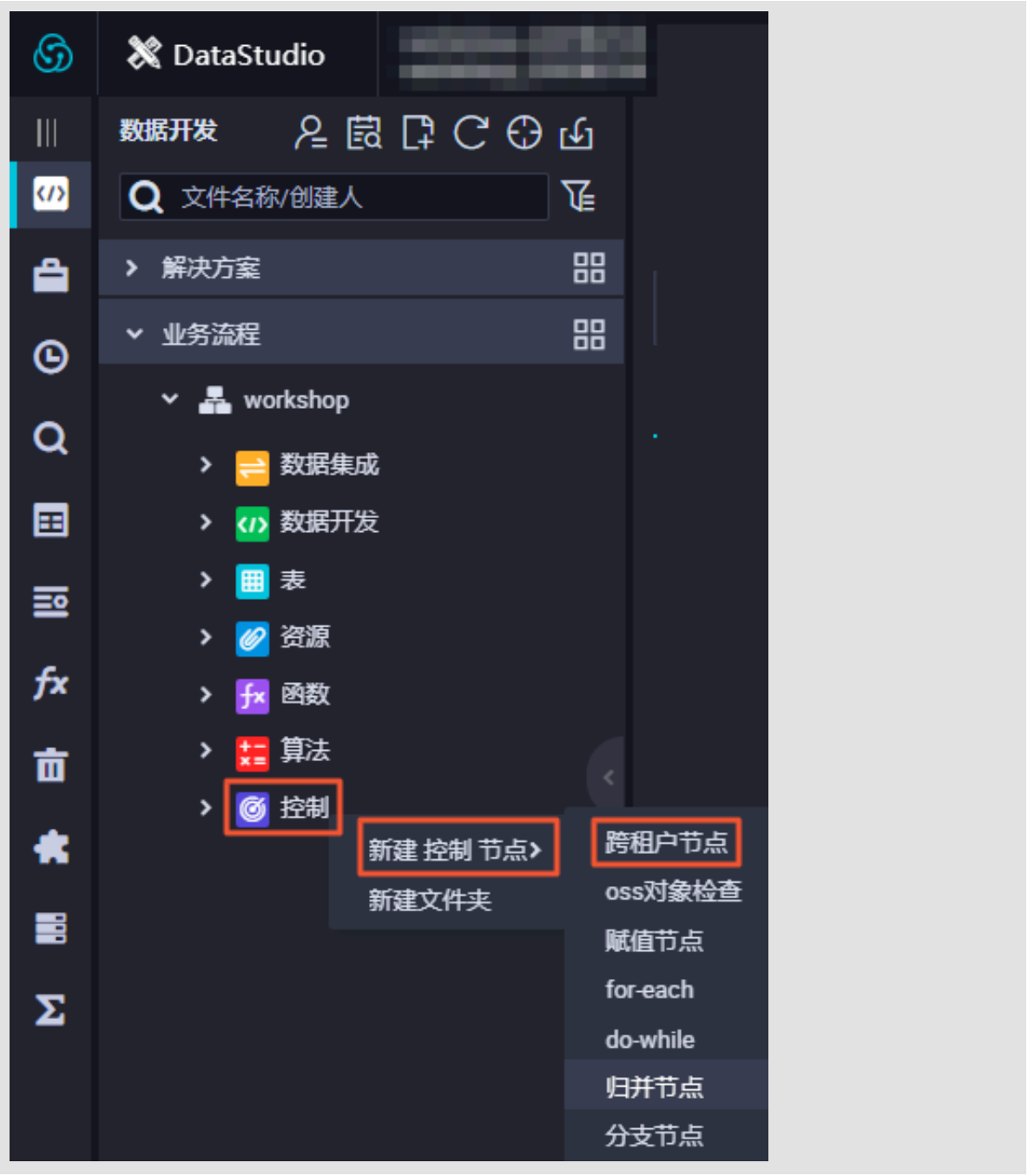

发布
运维
调度配置
版本
参数:
参数格式：变量名1=参数1 变量名2=参数2....多个参数之间用空格分隔
时间属性
生成实例方式：
T+1次日生成
发布后即时生成
注：及时生效不包含调度依赖关系
时间属性：
正常调度
空跑调度
出错重试：
生效日期：
1970-01-01
9999-01-01
注：调度将在有效日期内生效并自动调度，反之，在有效期外的任务将不会自动调度，也不能手动调度。
暂停调度：
调度周期：
日
定时调度：
具体时间：
00:22
注：默认调度时间，从0点到0点30分随机生成
cron表达式：
00 22 00 * * ?
依赖上一周期：

新建节点流程

1. 进入DataStudio（数据开发）页面，选择新建 > 控制 > 跨租户节点。

说明:

您也可以找到相应的业务流程，右键单击控制，选择新建控制节点 > 跨租户节点。


DataStudio
数据开发
文件名称/创建人
解决方案
业务流程
workshop
数据集成
数据开发
表
资源
函数
算法
控制
新建 控制 节点
新建文件夹
跨租户节点
oss对象检查
赋值节点
for-each
do-while
归并节点
分支节点

2. 填写新建节点对话框中的配置。

3. 单击提交。

4. 在跨租户节点配置页面进行节点配置。

配置	说明
类型	类型包括发送和接收。
节点标识	默认为当前节点所在路径，不可修改。
授权项目	授权项目和云账号需要写对端项目和账号，此处您配置的为发送方，则此处写接收方的授权项目和云账号。

5. 完成发送节点的配置后，需要到对应的接收节点账号和项目下，新建同样的控制节点。

类型选择接收，即可查看对应的节点信息。最后，您还需配置超时时间（在接收节点的开始时间以后，往后顺延超时时间）。

发送节点首先给消息中心发送一个消息，当发送成功后，发送节点便运行成功了。接收节点会循环去拉取消息中心的节点信息，当在超时时间范围内拉取到，就会将接收节点置为成功。

如果未在超时时间范围内接收到消息，则任务会被置为失败。发送消息的生命周期是24小时。

例如：20181008这天的周期实例已经运行完成，表示消息中心里面这条消息已存在，我们在接收节点补数据的时候，选择的业务日期为20181007，生成的接受消息实例会直接置为成功。

6. 配置完成后保存并提交节点。

3.5.14 归并节点

本文将为您介绍归并节点的概念，以及如何新建归并节点、定义归并逻辑，并通过实践案例为您展示归并节点的调度配置和运行详情。

> 说明:
> 您需要购买DataWorks标准版及以上版本，方可使用归并节点功能。

概念

- 归并节点是DataStudio中提供的逻辑控制系列节点中的一类。
- 归并节点可以对上游节点的运行状态进行归并，用来解决分支节点下游节点的依赖挂载和运行触发问题。
- 当前归并节点的逻辑定义不支持选择节点运行状态，只支持将分支节点的多个下游节点归并为成功，以便更下游节点能够直接挂载归并节点作为依赖。

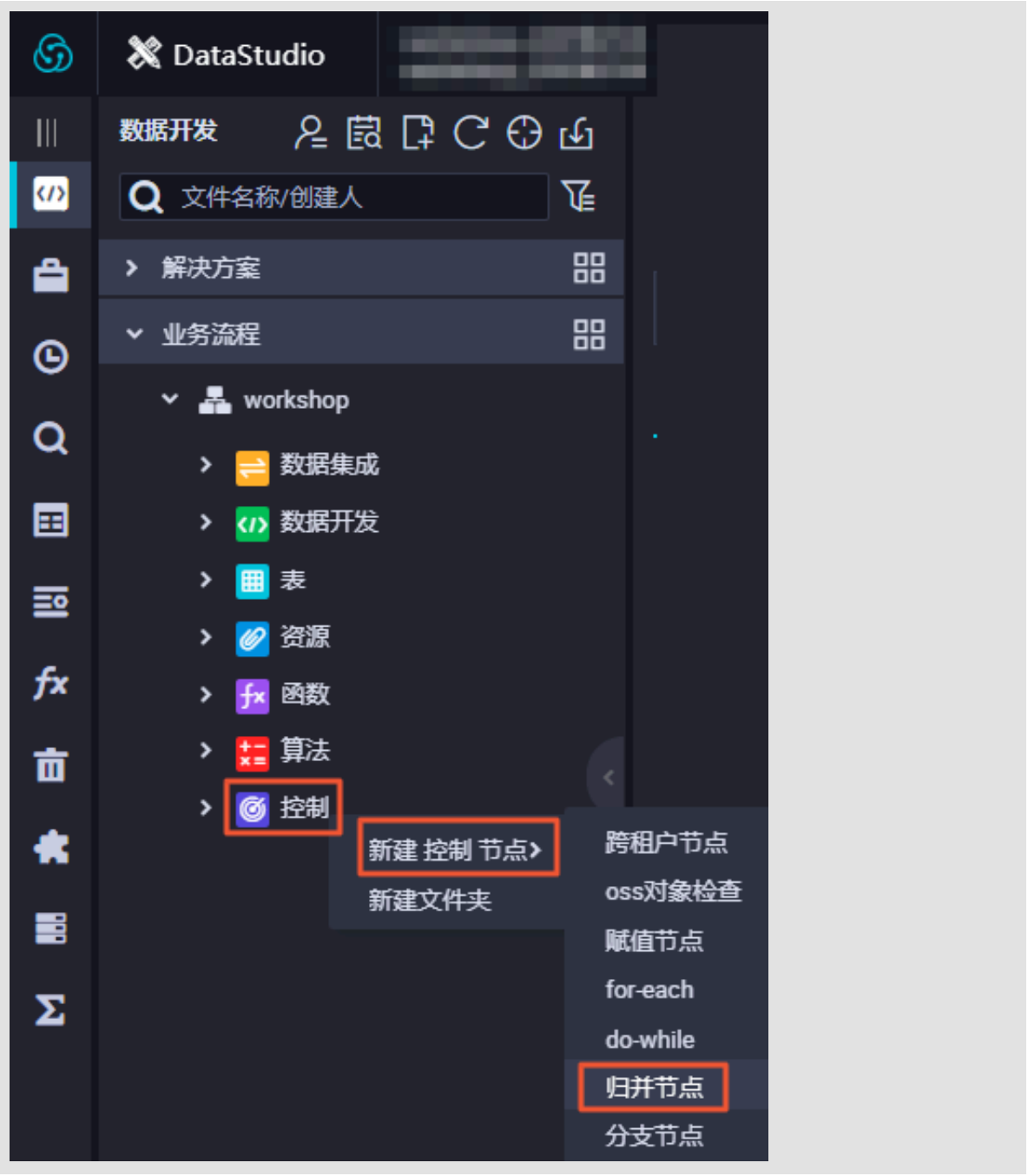
例如，分支节点C定义了两个逻辑互斥的分支走向C1和C2，不同分支使用不同的逻辑写入同一张MaxCompute表，若更下游节点B依赖此MaxCompute表的产出，就必须使用归并节点J先将分

支归并后，再把归并节点J作为B的上游依赖。若直接把B挂载在C1、C2下，任何时刻，C1和C2总有一个会因分支条件不满足而未运行，B则不能被调度触发运行。

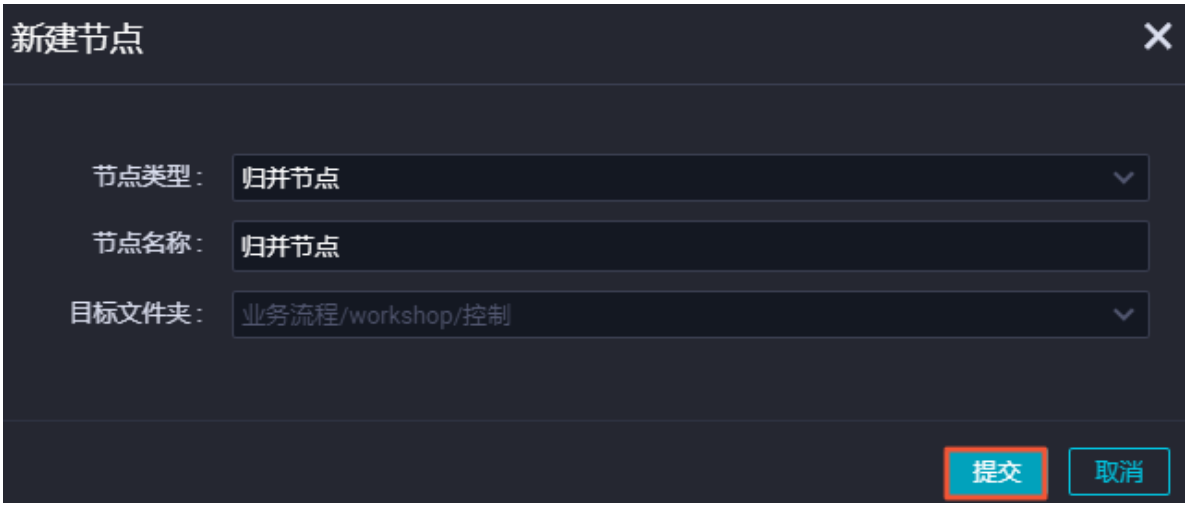
新建归并节点

1. 进入DataStudio（数据开发）页面，选择新建 > 控制 > 归并节点。

说明:
您也可以找到相应的业务流程，右键单击控制，选择新建控制节点 > 归并节点。


DataStudio
数据开发
文件名称/创建人
解决方案
业务流程
workshop
数据集成
数据开发
表
资源
函数
算法
控制
新建 控制 节点
新建文件夹
跨租户节点
oss对象检查
赋值节点
for-each
do-while
归并节点
分支节点

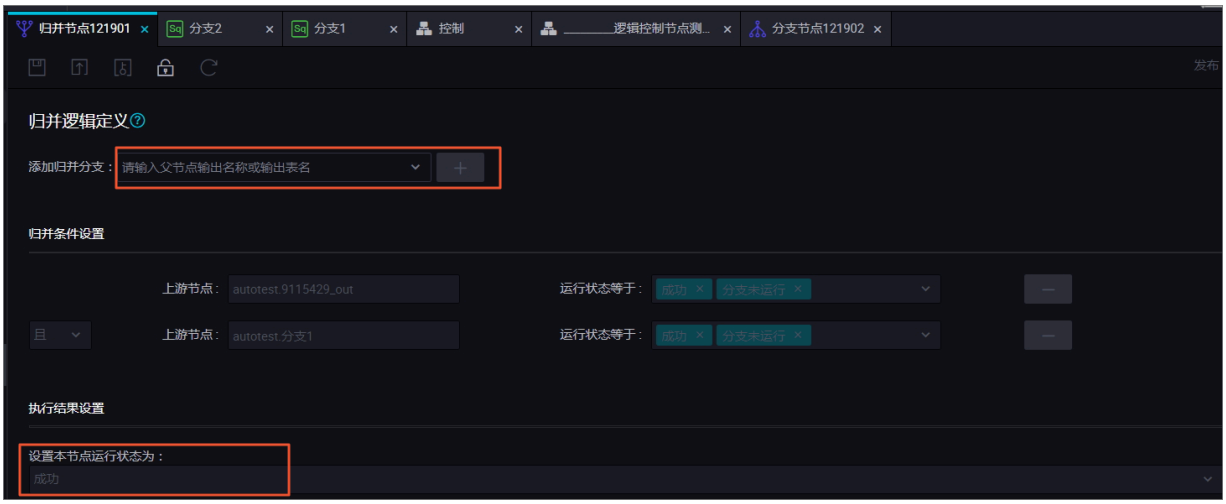
2. 填写新建节点对话框中的配置。



3. 单击提交。

定义归并逻辑

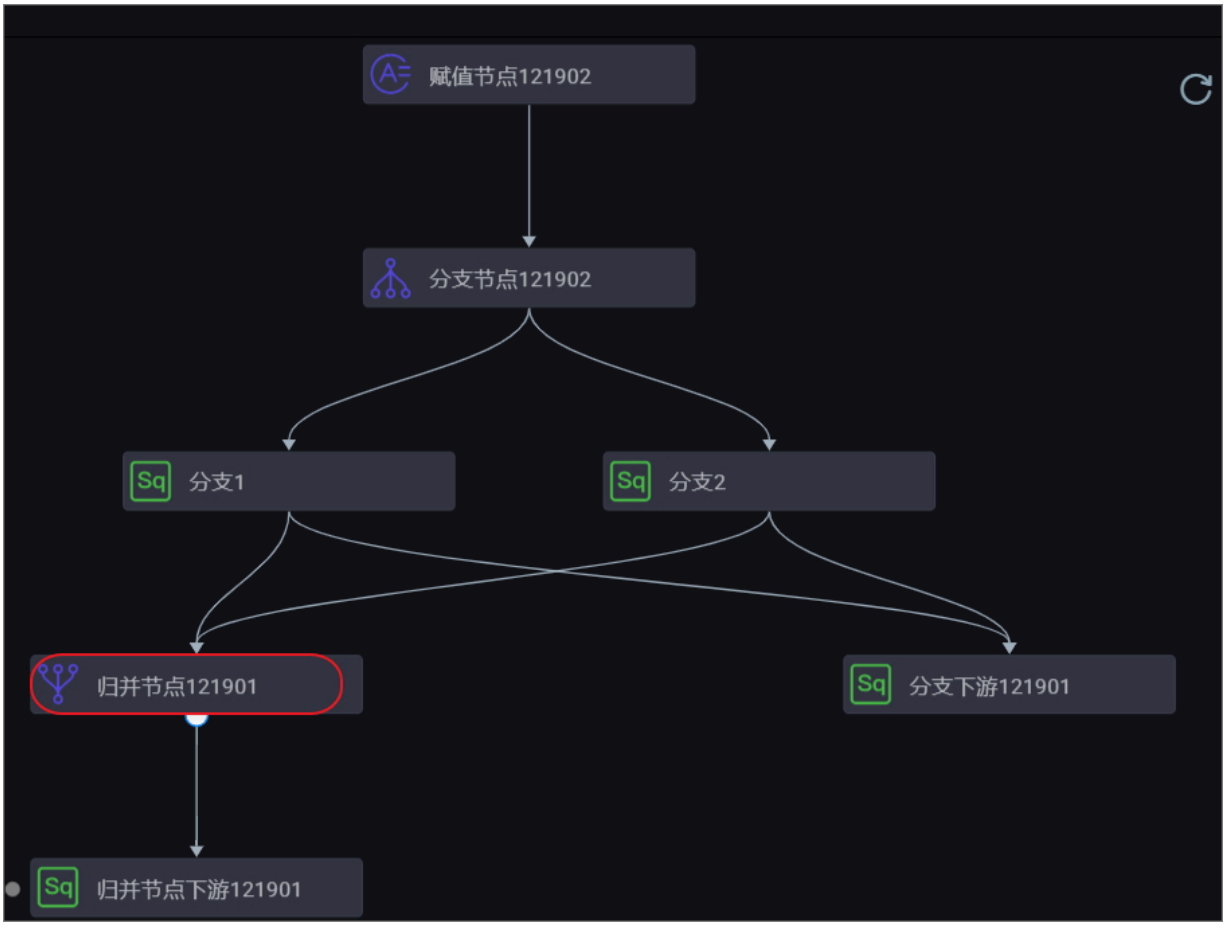
新建归并节点后，进入编辑页面添加归并分支。您可以输入父节点的输出名称或输出表名，单击添加按钮。您可以在执行结果中查看运行状态，目前只有成功和分支未运行两种状态。



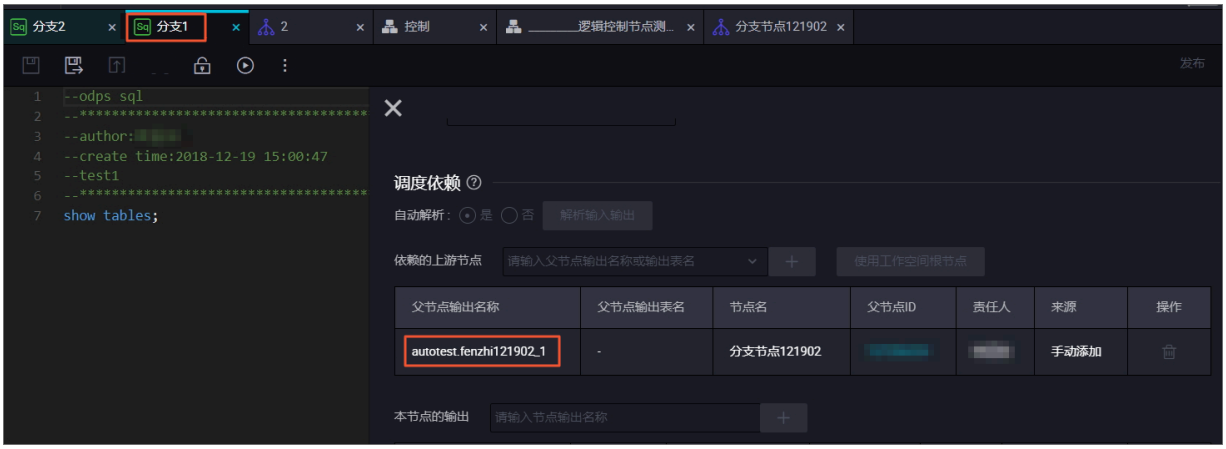
单击右侧的调度属性，即可对归并节点的调度属性进行设置。

归并节点示例

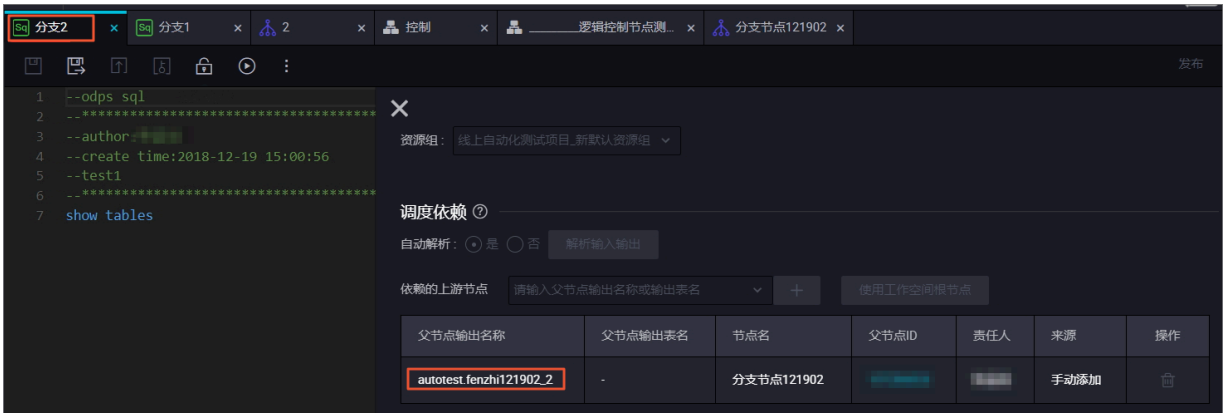
在下游节点中，添加分支节点作为上游节点后，通过选择对应的分支节点输出来定义不同条件下的分支走向。例如在下图所示的业务流程中，分支1和分支2均为分支节点的两个下游节点。



分支1依赖于autotest.fenzhi121902_1输出。

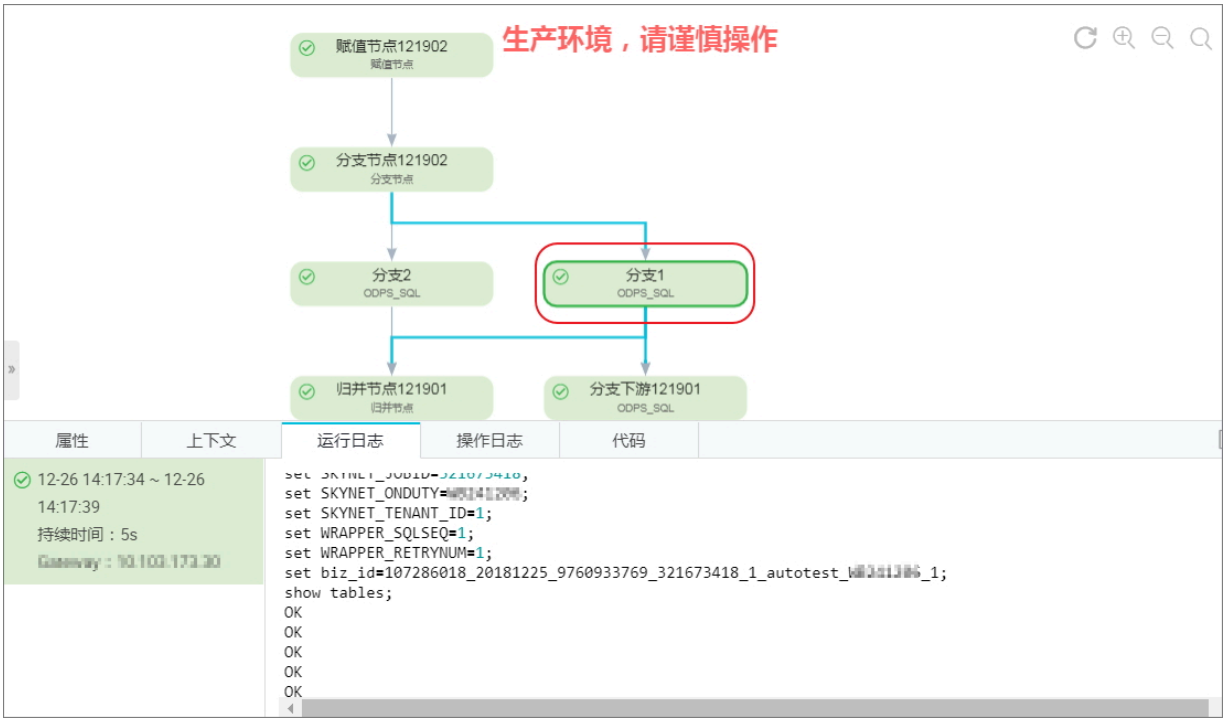


分支2依赖于autotest.fenzhi121902_2输出。

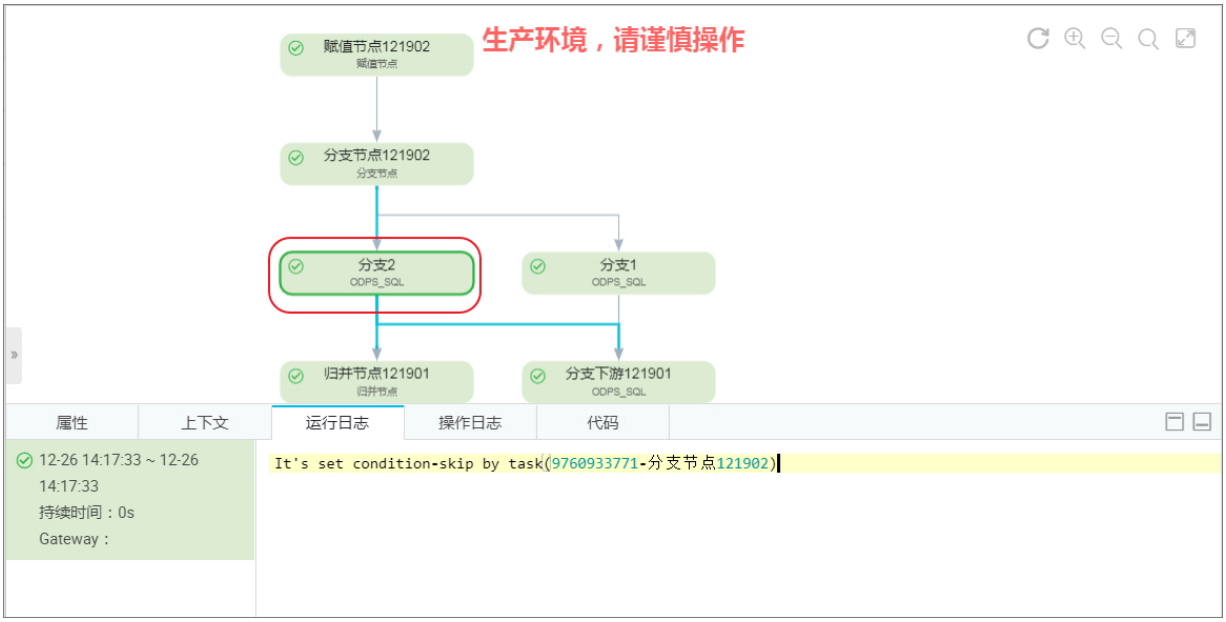


运行任务

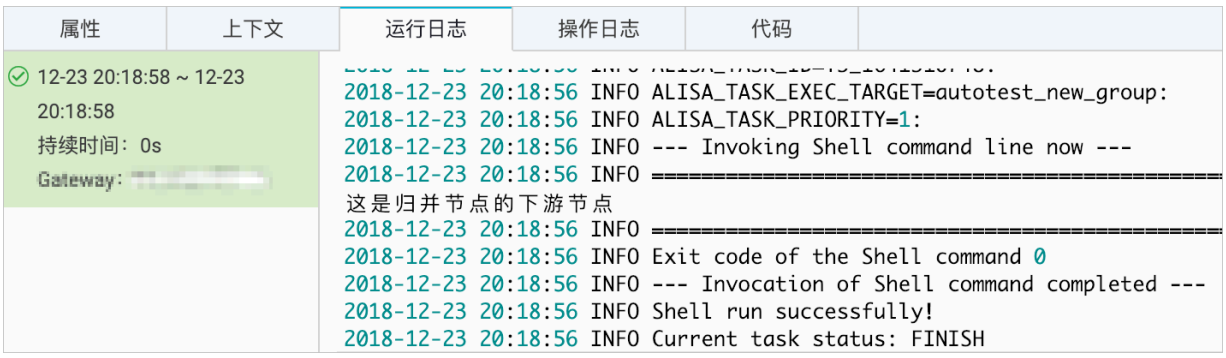
您可以在运行日志中查看满足分支条件、被选中运行的分支下游节点的运行情况。



您可以在运行日志中查看到不满足分支条件、未被选中运行的分支下游节点，被置为跳过。



归并节点的下游节点正常运行。



3.5.15 分支节点

分支节点是DataStudio中提供的逻辑控制系列节点中的一类。分支节点可以定义分支逻辑和不同逻辑条件时下游分支走向。

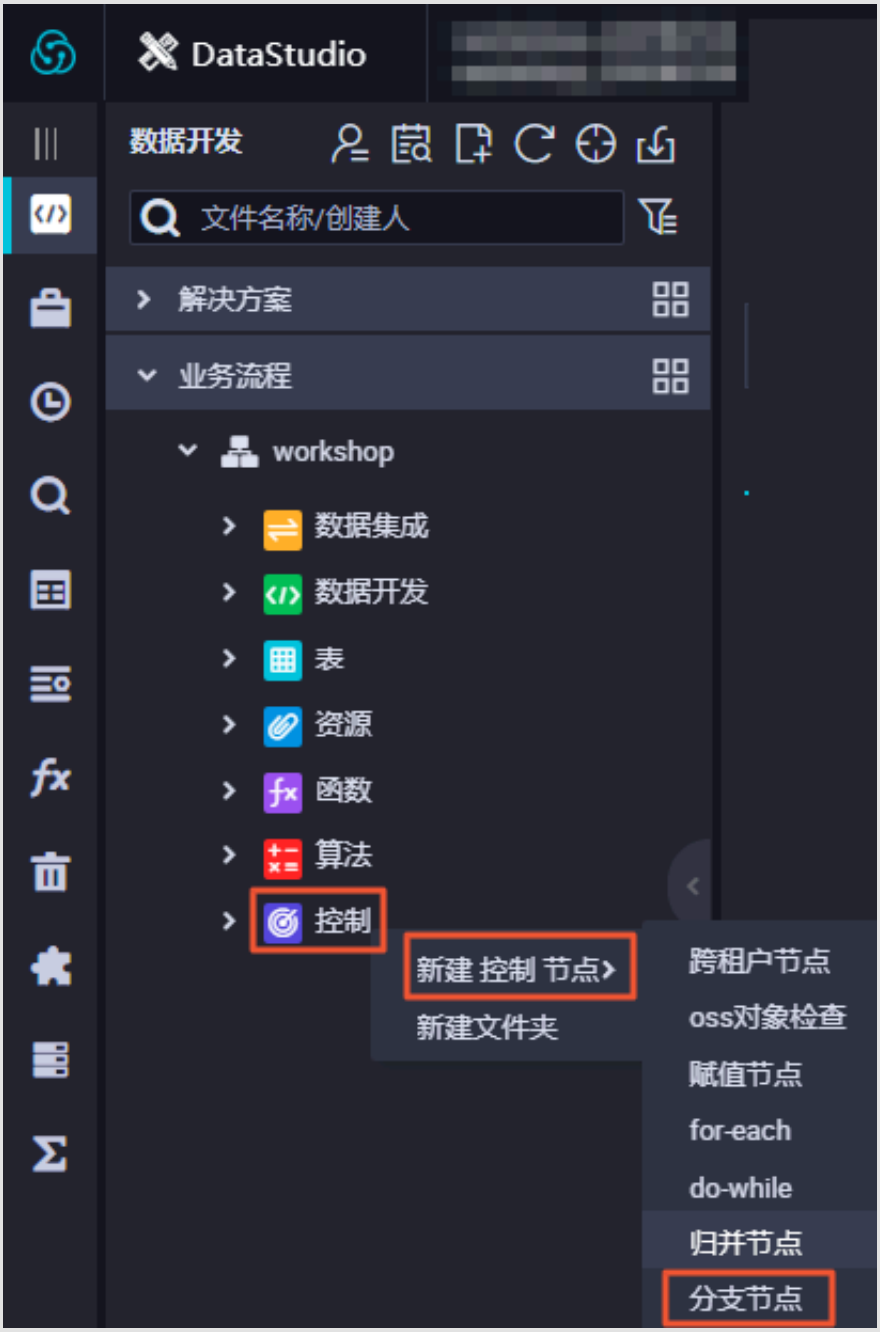
> 说明:
>
> · 您需要购买DataWorks标准版及以上版本，方可使用分支节点功能。
>
> · 分支节点通常需要与#unique_337配合使用。

新建分支节点

1. 进入DataStudio（数据开发）页面，选择新建 > 控制 > 分支节点。

> 说明:

您也可以找到相应的业务流程，右键单击控制，选择新建控制节点 > 分支节点。



2. 填写新建节点对话框中的配置。

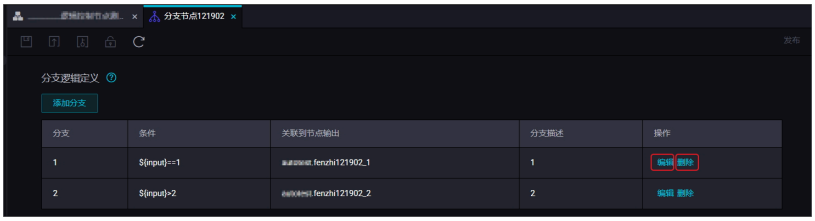
3. 单击提交。

定义分支逻辑

1. 创建分支节点后，跳转至分支逻辑定义页面。

2. 单击添加分支，在配置分支定义对话框中，填写分支条件、关联到节点输出和分支描述。

配置	说明
分支条件	· 分支条件只支持按照Python比较运算符定义逻辑判断条件。 · 如果运行态表达式取值为true，表示对应的分支条件满足，反之为不满足。 · 如果运行态表达式解析报错，会将整个分支节点运行状态置为失败。 · 分支条件中支持使用全局变量和节点上下文定义的参数，例如图中的`${input}`可以是定义在分支节点的节点输入参数。
关联到节点输出	· 节点输出供分支节点下游节点挂载依赖关系使用。 · 分支条件满足时，对应的关联的节点输出上挂载的下游节点被选中运行（同时需要参考该节点依赖的其它上游节点的状态）。 · 分支条件不满足时，对应的关联的节点输出上挂载的下游节点不会被选中执行，该下游节点会被置成“因为分支条件不满足而未运行”的状态。

配置	说明
分支描述	对于分支定义的描述。定义${input}==1和${input}>2两个分支。  · 编辑：单击编辑按钮，可以修改设置的分支并且相关的依赖关系也会改动。 · 删除：单击删除按钮，可以删除设置的分支并且相关的依赖关系也会改动。

3. 配置完成后，单击确认。

调度配置

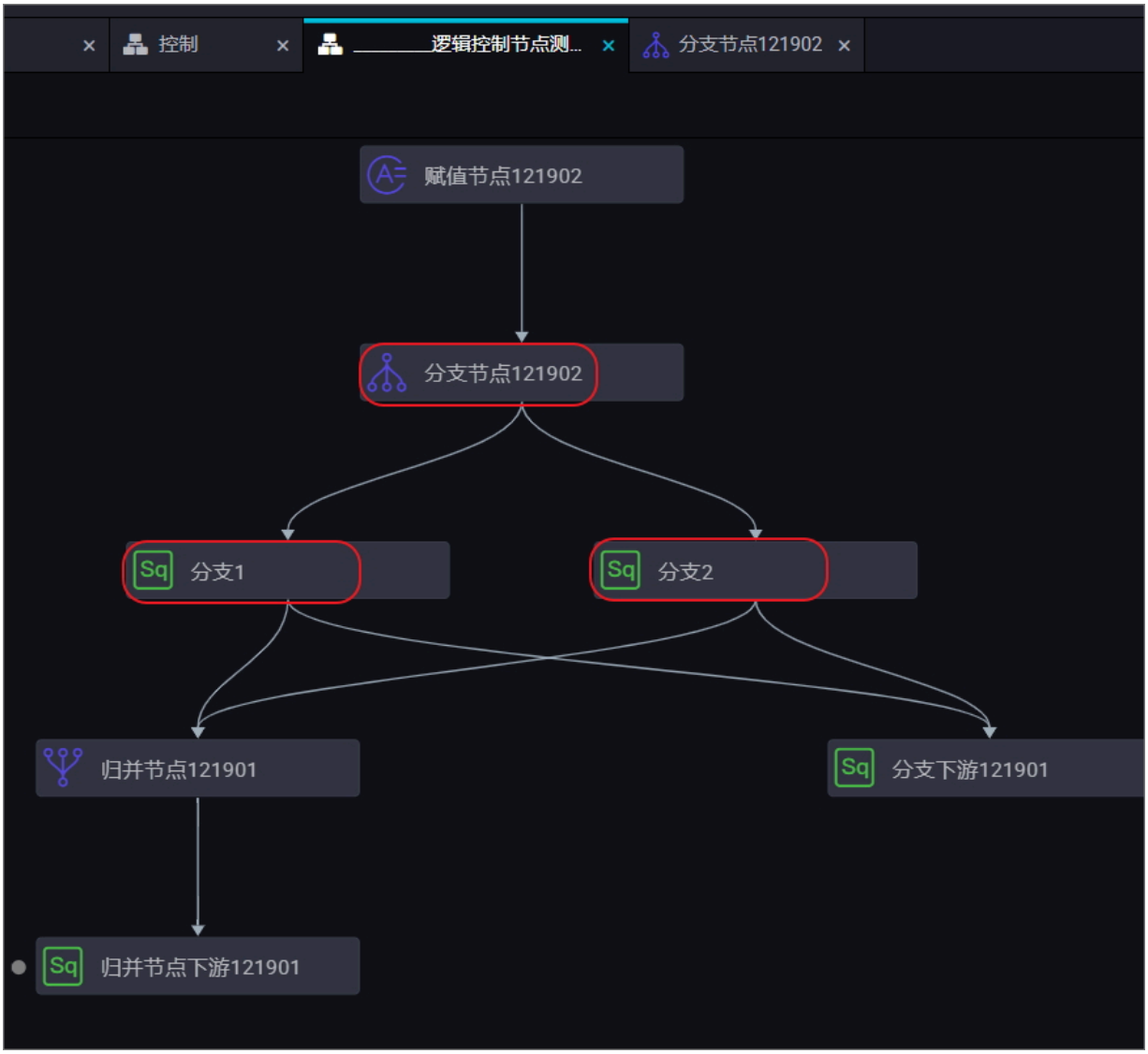
定义好分支条件后，会在调度配置中本节点的输出自动添加输出名称，下游节点可以通过输出名称进行依赖挂载。

说明：

如果连线建立上下文的依赖，在调度配置中没有输出记录，请手动输入。

输出案例：下游节点挂载分支节点

在下游节点中，添加分支节点做为上游节点后，通过选择对应的分支节点输出来定义不同条件下的分支走向。例如在下图所示的业务流程中，分支1和分支2均为分支节点的两个下游节点。



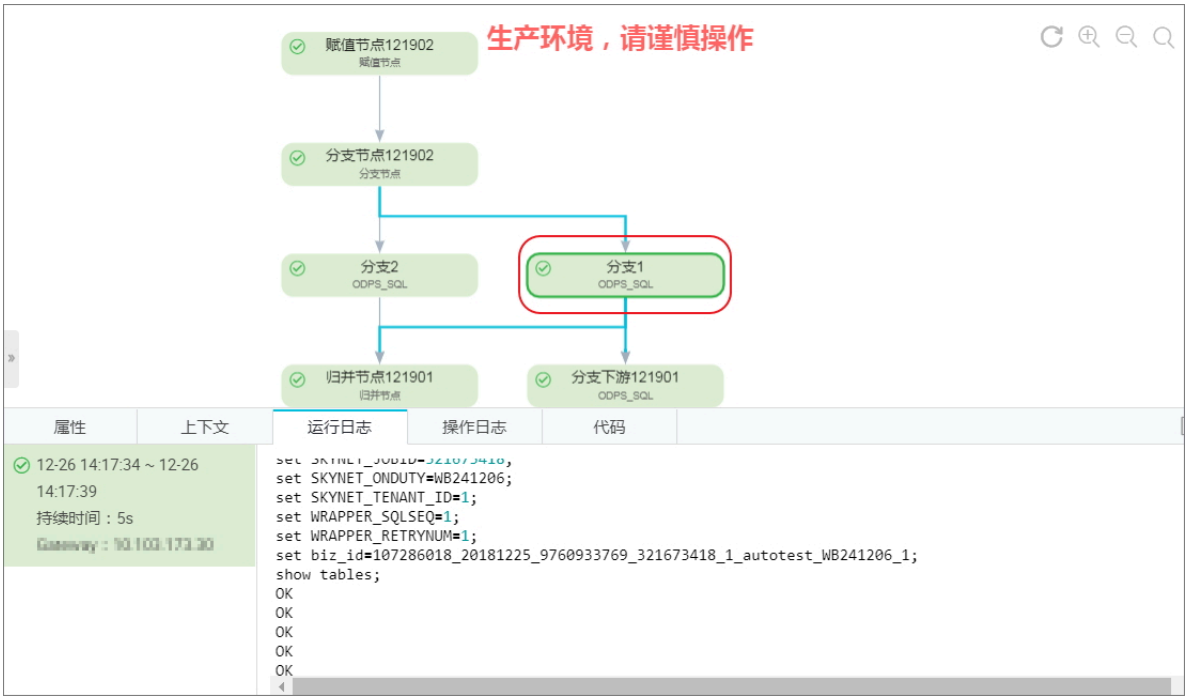
分支1：依赖于autotest.fenzhi121902_1输出。

分支2：依赖于autotest.fenzhi121902_2输出。

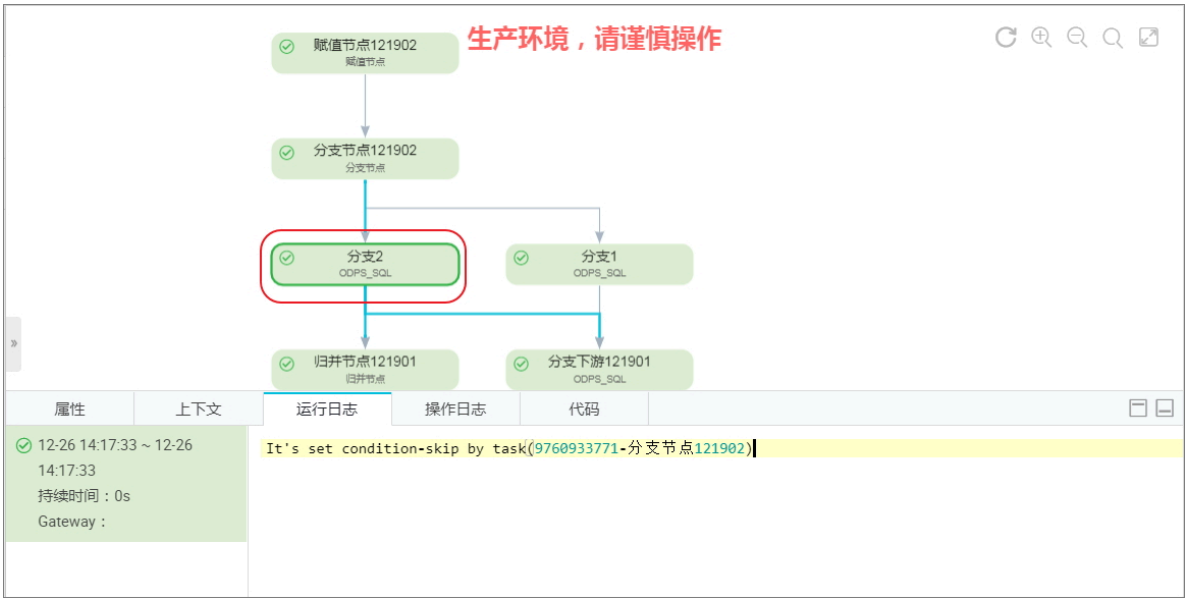
提交调度运行

提交调度到运维中心运行，分支节点满足条件一（依赖于autotest.fenzhi121902_1），所以其日志的打印结果具体如下。

- **您可以在运行日志中查看满足分支条件、被选中运行的分支下游节点的运行情况。**



- 您可以在运行日志中查看到不满足分支条件、未被选中运行的分支下游节点，被置为跳过。



支持的Python比较运算符

以下假设变量a为10，变量b为20。

运算符	描述	实例
==	等于-比较对象是否相等。	（a==b）返回False。
!=	不等于-比较两个对象是否不相等。	（a!=b）返回true。
<>	不等于-比较两个对象是否不相等。	（a<>b）返回true。这个运算符类似!=。
>	大于-返回x是否大于y。	（a>b）返回False。
<	小于-返回x是否小于y。所有比较运算符返回1表示真，返回0表示假。这分别与特殊的变量True和False等价。	（a<b）返回true。
>=	大于等于-返回x是否大于等于y。	（a>=b）返回False。
<=	小于等于-返回x是否小于等于y。	（a<=b）返回true。

3.5.16 赋值节点

赋值节点是一种特殊的节点类型，支持在节点中通过编写代码的方式对输出参数赋值，结合节点上下文传递，供下游节点引用和使用其取值。

> **说明:**
> 您需要购买DataWorks标准版及以上版本，方可使用赋值节点功能。

新建赋值节点

1. 进入DataStudio（数据开发）页面，选择新建 > 控制 > 赋值节点。

说明:

您也可以找到相应的业务流程，右键单击控制，选择新建控制节点 > 赋值节点。

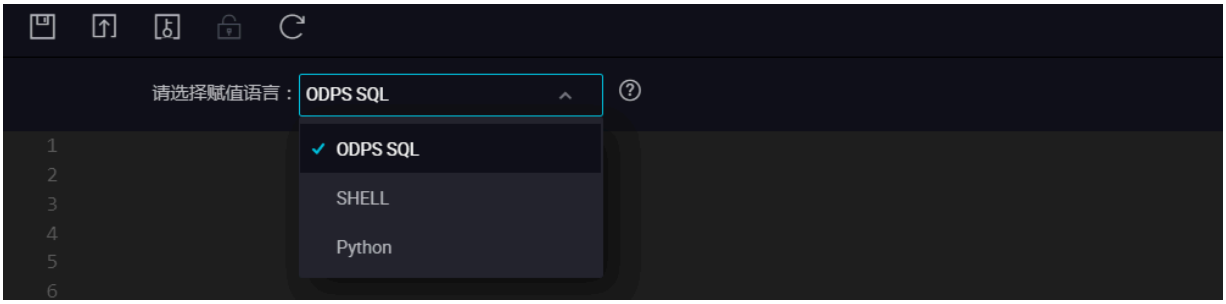

DataStudio
数据开发
文件名称/创建人
解决方案
业务流程
workshop
数据集成
数据开发
表
资源
函数
算法
控制
新建 控制 节点
新建文件夹
跨租户节点
oss对象检查
赋值节点
for-each
do-while
归并节点
分支节点

2. 填写新建节点对话框中的配置。

3. 单击提交。

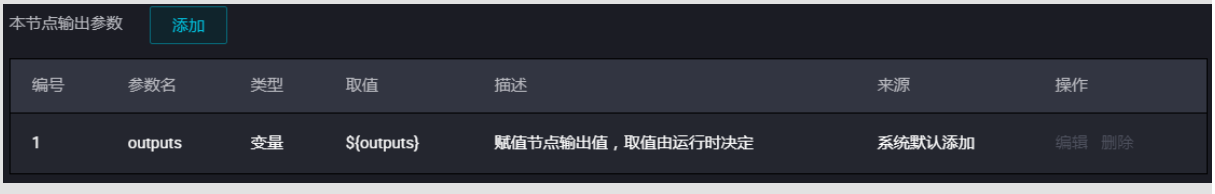
编写赋值节点取值逻辑

赋值节点在节点上下文中有一个固定的名为outputs的输出参数。支持使用ODPS SQL、SHELL和Python三种语言来编写代码对参数进行赋值，其取值是节点代码的运行、计算结果。单个赋值节点只能选择一种语言。



说明:

- outputs参数的取值只取最后一行代码的输出结果。
  - ODPS SQL最后一行SELECT语句的输出。
  - SHELL最后一行ECHO语句的数据。
  - Python最后一行PRINT语句的输出。
- outputs参数的取值有一定限制，其传递值最大为2M。如果赋值语句的输出结果超过此限制，则赋值节点会运行失败。

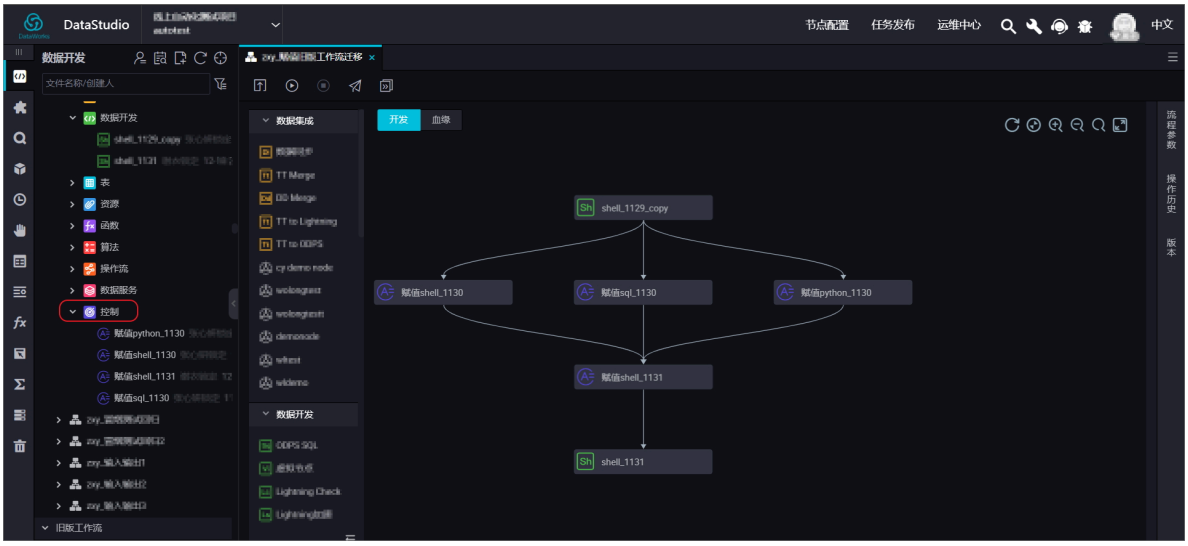


在下游节点使用赋值节点的输出

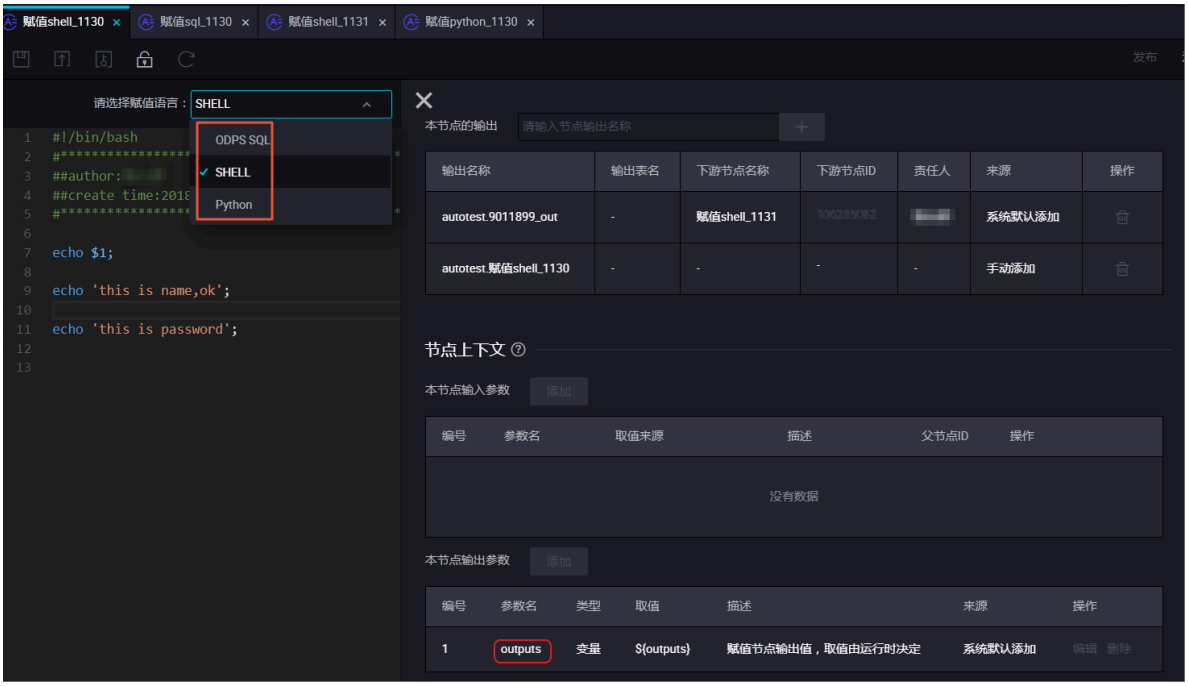
在下游节点中，添加赋值节点作为上游依赖后，通过节点上下文的方式，将赋值节点的输出定义为本节点的输入参数，并在代码中引用，即可取得上游赋值节点输出参数的具体取值。详情请参见节点上下文。

赋值节点示例

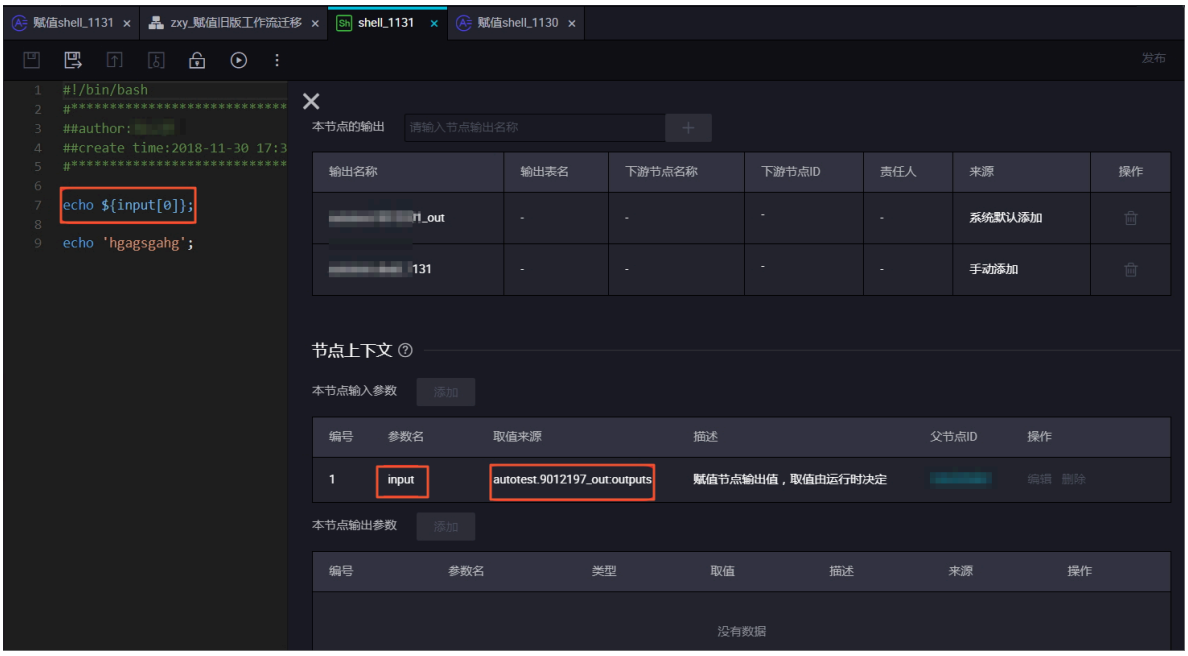
1. 创建业务流程，再分别创建下图中的节点。



2. 配置赋值节点时，系统默认会显示一个outputs参数，运行后您在相关的运维中心 > 基本属性 > 上下文页面可以找到相关的参数结果。



3. 上游的outputs参数在下游作为下游的输入参数。

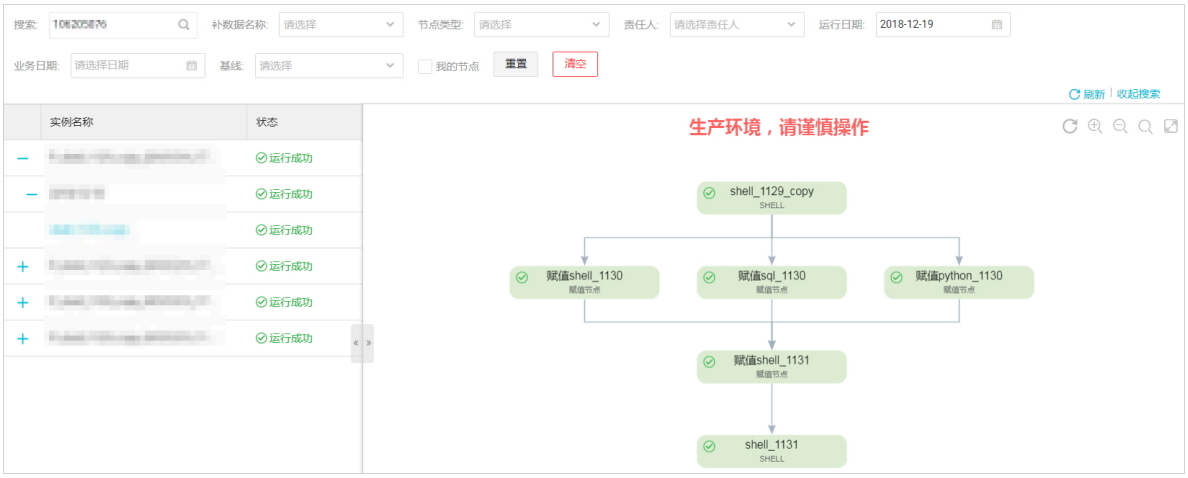


运行赋值节点任务

说明:

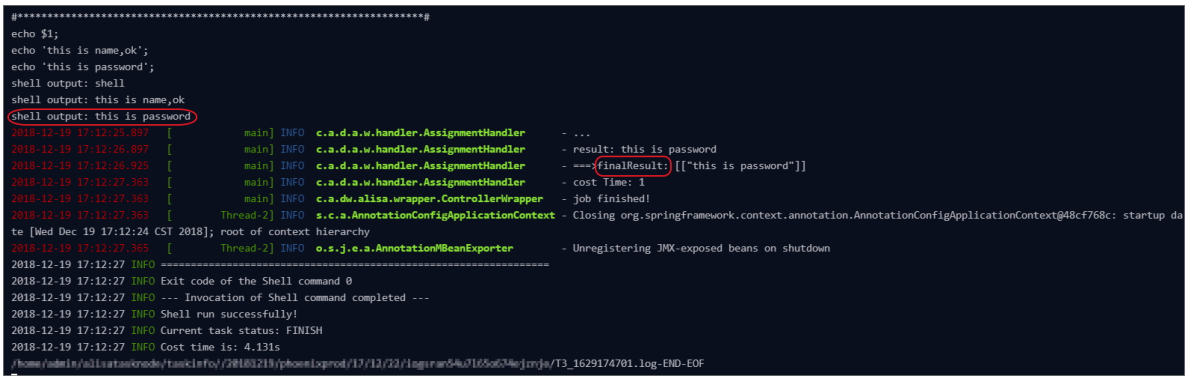
通过运维中心的补数据运行，可以将上面的配置参数生效，但测试运行参数不能生效。

1. 任务配置完提交调度，通常会在第二天产生运行实例。



2. 在运行时，您可以查看上下文的输入输出参数，单击后面的链接可以看到您的输入/输出结果。

3. 在运行日志中，您可以通过finalResult查看代码的最后输出结果。

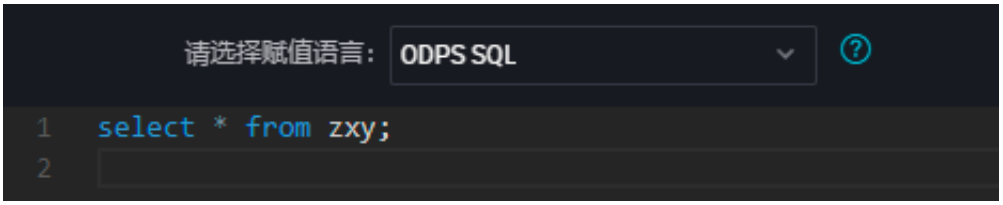


总结

> **说明:**
> outputs输出您的代码中最后一条输出结果。

下文为您介绍ODPS SQL、SHELL和Python数组的常规用法，均以普通SHELL节点中的input参数输出结果。

- ODPS SQL：支持二维数组和一维数组。



查询的结果是一个二维数组，如下所示。

```
2,this is name6
1,this is name5
```

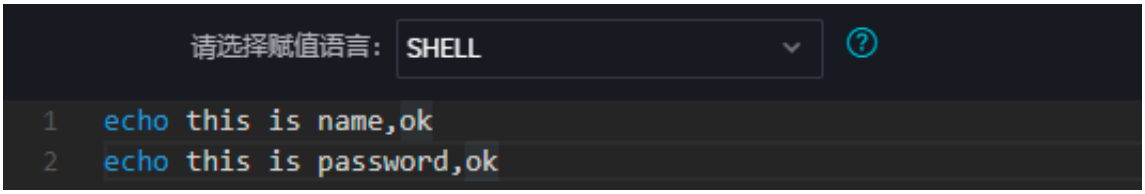
SHELL中的输出代码，如下所示。

```
echo ${input[0][0]};
echo ${input[0][1]};
echo ${input[1][0]};
echo ${input[1][1]};
echo ${input[0]};
```

输出结果如下所示。

```
2019-02-21 11:11:55 INFO --- Invoking Shell command line now ---
2019-02-21 11:11:55 INFO ================================================================
2
this is name6
1
this is name5
2,this is name6
```

- SHELL：输出为一维数组。



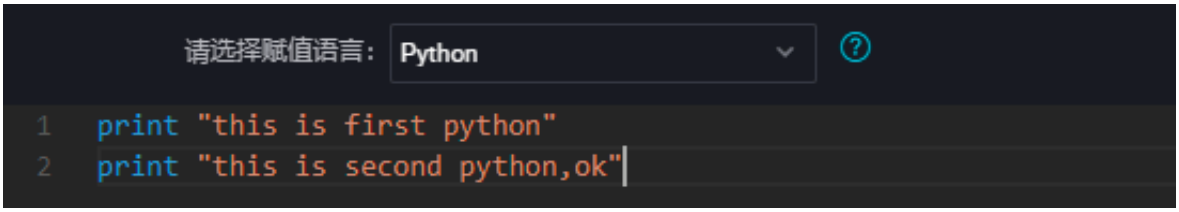
SHELL中的输出代码，如下所示。

```
#**********************************************************************
echo ${input};
echo ${input[0]};
echo ${input[1]}
```

输出结果如下所示。

```
2019-02-21 11:59:13 INFO --- Invoking Shell command line now ---
2019-02-21 11:59:13 INFO ================================================================
this is password,ok
this is password
ok
```

- Python：输出为一维数组。



SHELL中的输出代码，如下所示。

```
#**********************************************************************#
echo ${input};
echo ${input[0]};
echo ${input[1]}
```

输出结果如下所示。

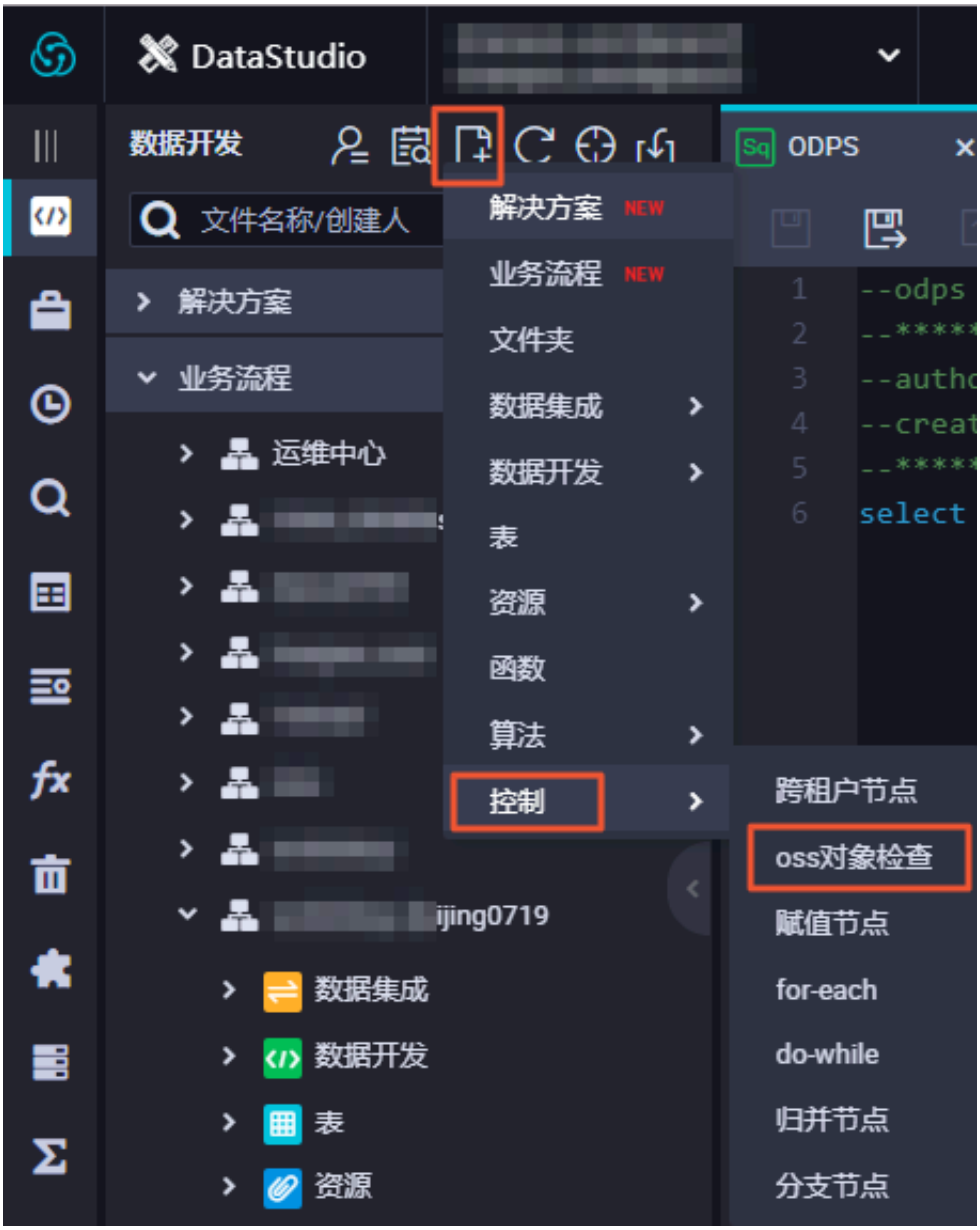
```
2019-02-21 12:07:13 INFO --- Invoking Shell command line now ---
2019-02-21 12:07:13 INFO ================================================================
this is second python,ok
this is second python
ok
2019-02-21 12:07:13 INFO ================================================================
```

3.5.17 OSS对象检查

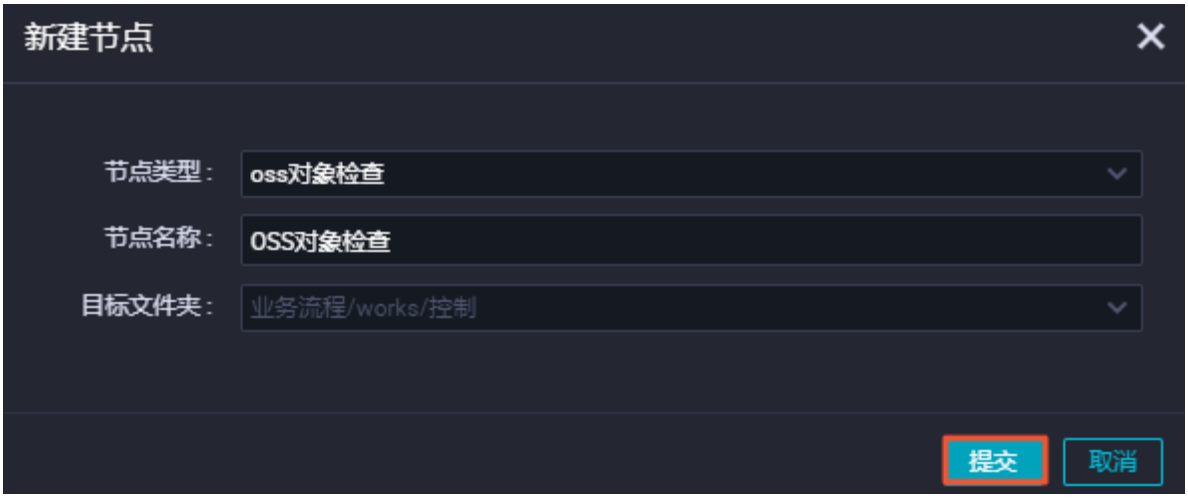
您可以在下游任务需要依赖该OSS对象传入OSS时，使用OSS对象检查功能。例如同步OSS数据至DataWorks，需要检测出已经产生OSS数据文件，方可进行OSS同步任务。

检测对象：可以检测所有租户下的OSS对象。

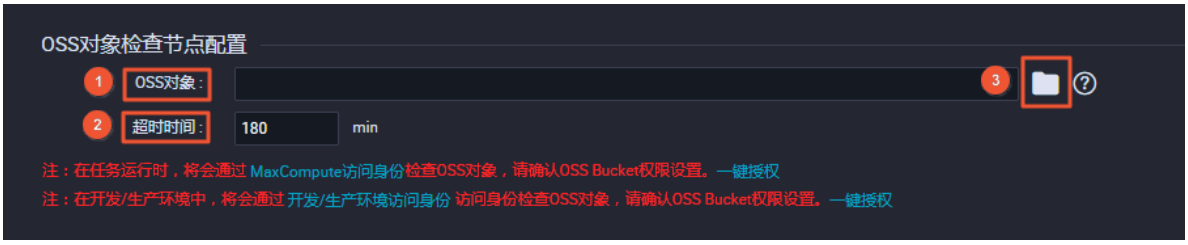
1. 进入数据开发页面，选择新建 > 控制 > OSS对象检查。



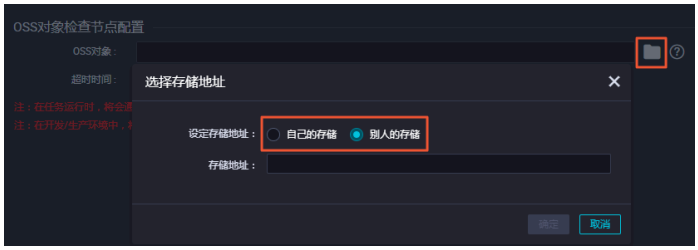
2. 填写新建节点对话框中的配置，单击提交。



3. 新建成功后，进行OSS对象检查节点配置。



序号	配置	说明
1	OSS对象	此处可以手动填写OSS对象的存储路径，路径支持使用调度参数，详情请参见#unique_341。
2	超时时间	在超时时间内，每5秒检测该OSS对象是否存在于OSS中。如果超出超时时间，仍未检测到OSS对象的存在，则OSS对象检查任务会失败。 

序号	配置	说明
3	选择存储地址	您可以选择以下两种存储地址： · 自己的存储：检测当前租户下的OSS对象。 · 别人的存储：检测非当前租户下的OSS对象。 

说明：

- 任务在运行时，会通过MaxCompute访问身份检查OSS对象，请确认OSS Bucket的权限设置，详情请参见#unique_342。
- 在开发/生产环境中，任务会通过开发/生产环境访问身份检查OSS对象，请确认OSS Bucket的权限设置。

4. 在RAM中授权MaxCompute访问OSS的权限。

   MaxCompute结合了阿里云的访问控制服务（RAM）和令牌服务（STS），来解决账号的安全问题。

   - 当MaxCompute和OSS的owner是同一个账号时，可以直接在RAM控制台进行一键授权操作。
   - 当MaxCompute和OSS的owner不是同一账号时，可以通过以下操作进行授权：

     a. 在RAM中授权MaxCompute访问OSS的权限。

     创建如AliyunODPSDefaultRole或AliyunODPSRoleForOtherUser的角色，并设置如下策略内容：

```
--MaxCompute和OSS的Owner不是同一个账号。
{
"Statement": [
{
"Action": "sts:AssumeRole",
"Effect": "Allow",
"Principal": {
"Service": [
"MaxCompute的Owner云账号id@odps.aliyuncs.com"
]
}
}
],
"Version": "1"
```

```
}
```

b. 授予角色访问OSS必要的权限AliyunODPSRolePolicy。

```
{
"Version": "1",
"Statement": [
{
"Action": [
 "oss:ListBuckets",
 "oss:GetObject",
 "oss:ListObjects",
 "oss:PutObject",
 "oss:DeleteObject",
 "oss:AbortMultipartUpload",
 "oss:ListParts"
],
"Resource": "*",
"Effect": "Allow"
}
]
}
--您可以自定义其他权限。
```

c. 将权限AliyunODPSRolePolicy授权给该角色。

5. 进入运维中心页面，查看运行日志。

如果出现如下所示的日志信息，说明未检测到OSS对象产生。

```
<Error>
 <Code>NoSuchKey</Code>
 <Message>The specified key does not exist.</Message>
 <RequestId></RequestId>
 <HostId>oss对象</HostId>
 <Key>xc/111.txt</Key>
</Error>
```

3.5.18 机器学习节点

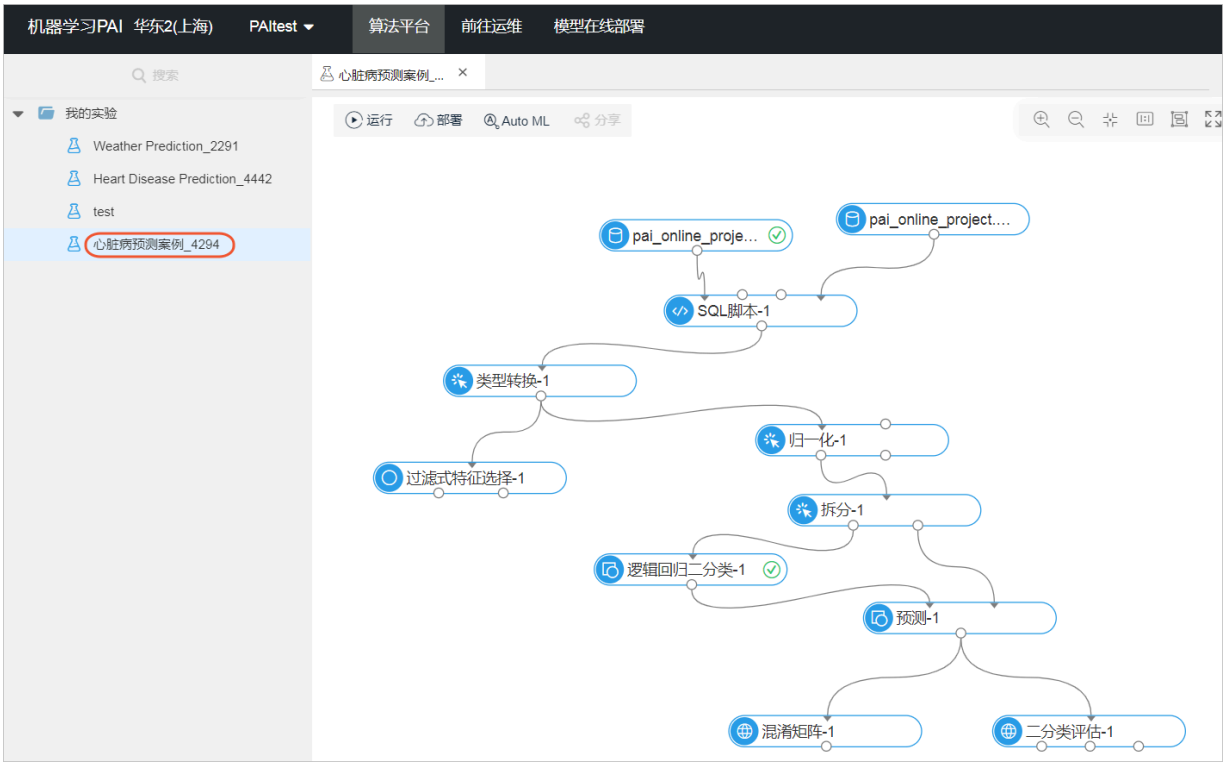
机器学习节点用来调用机器学习平台中构建的任务，并按照节点配置进行调度生产。机器学习任务，只有在机器学习平台创建了机器学习实验以后，才可以在DataWorks中添加。

创建机器学习实验

只有在机器学习平台中可以查找到的实验，才能被加载到机器学习节点中。配置机器学习实验请参见机器学习文档。

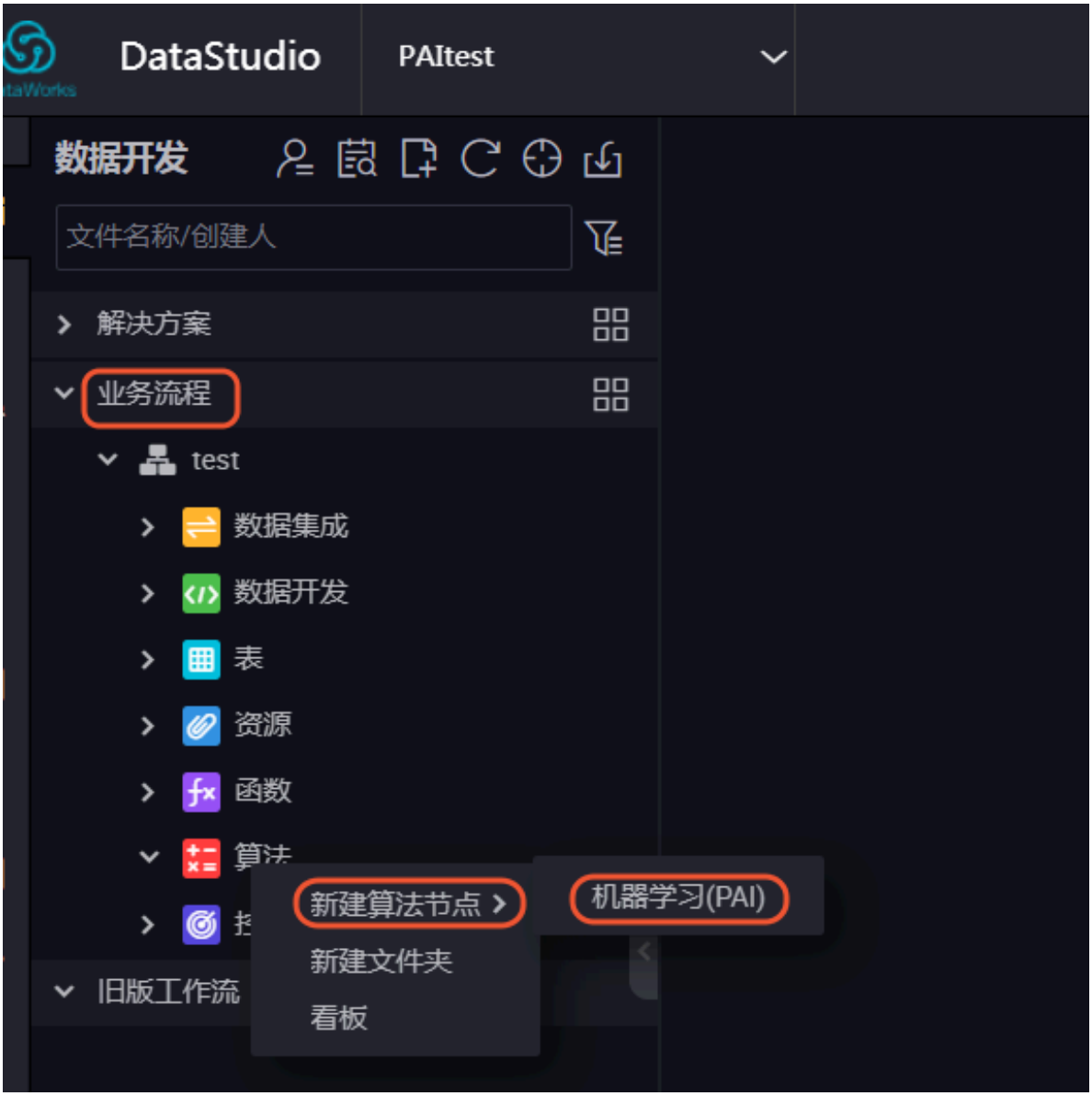
创建机器学习节点

根据上一节的操作，创建机器学习实验。本例中实验名称为心脏病预测案例_4294，然后在DataWorks中创建机器学习节点。


机器学习PAI 华东2(上海)
PAItest
算法平台
前往运维
模型在线部署
搜索
心脏病预测案例_...
我的实验
Weather Prediction_2291
Heart Disease Prediction_4442
test
心脏病预测案例_4294
运行
部署
Auto ML
分享
pai_online_proje...
pai_online_project....
SQL脚本-1
类型转换-1
归一化-1
过滤式特征选择-1
拆分-1
逻辑回归二分类-1
预测-1
混淆矩阵-1
二分类评估-1

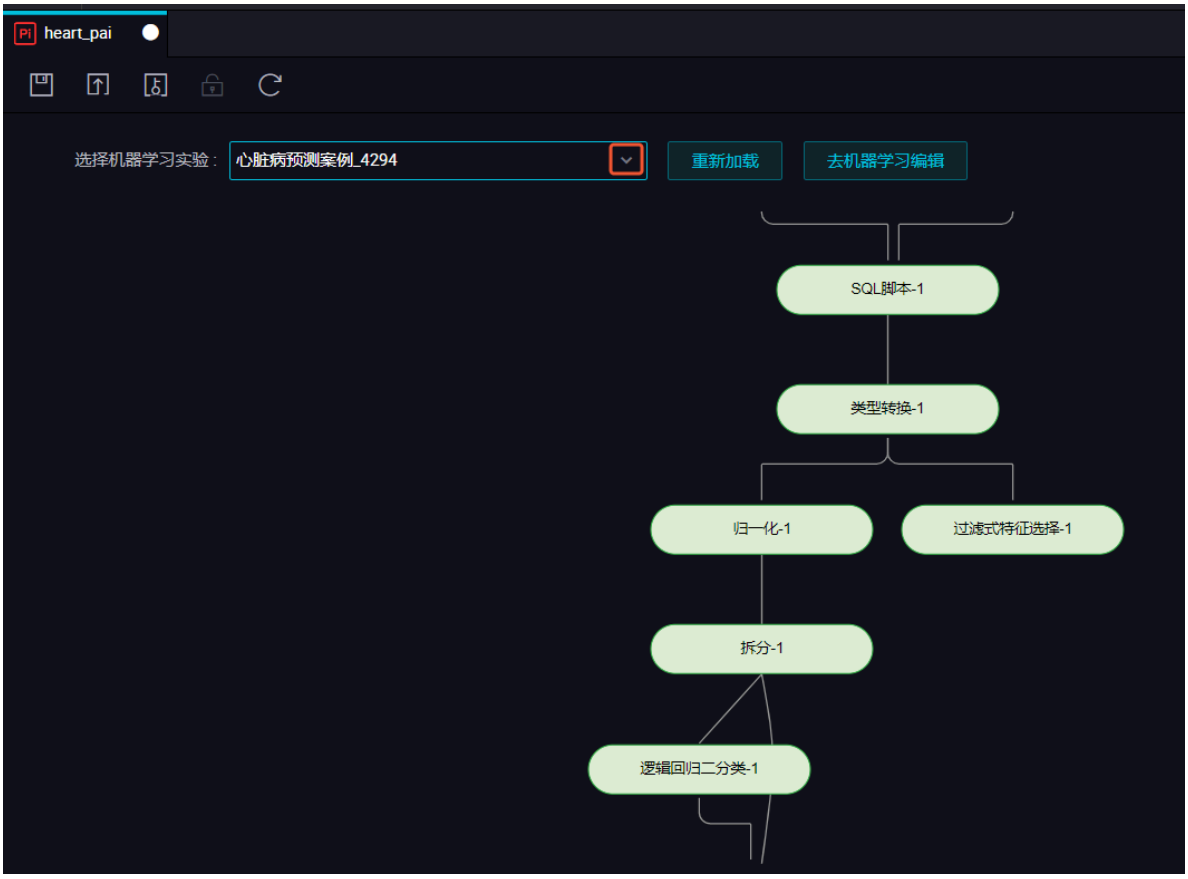
具体操作步骤如下：

1. 在您创建好的业务流程中，右键单击算法，选择新建算法节点 > 机器学习（PAI）。



2. 输入节点名称。

3. 选择您创建好的机器学习实验，如果在机器学习平台中已经创建了机器学习实验，可以直接进行选择，加载实验。



完成加载后，您还可以选择去机器学习编辑或直接提交实验。

3.5.19 自定义节点

3.5.19.1 自定义节点概述

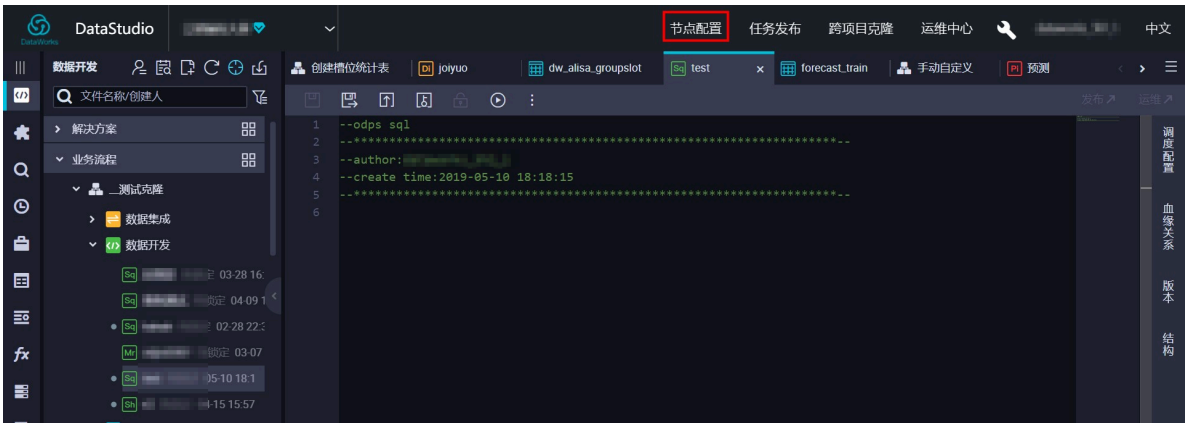
DataStudio不仅支持原生的ODPS SQL、Shell等系统节点，也支持自定义节点来满足需求。

新增自定义节点包括新增自定义插件、使用插件定义新节点类型两个步骤。

配置入口

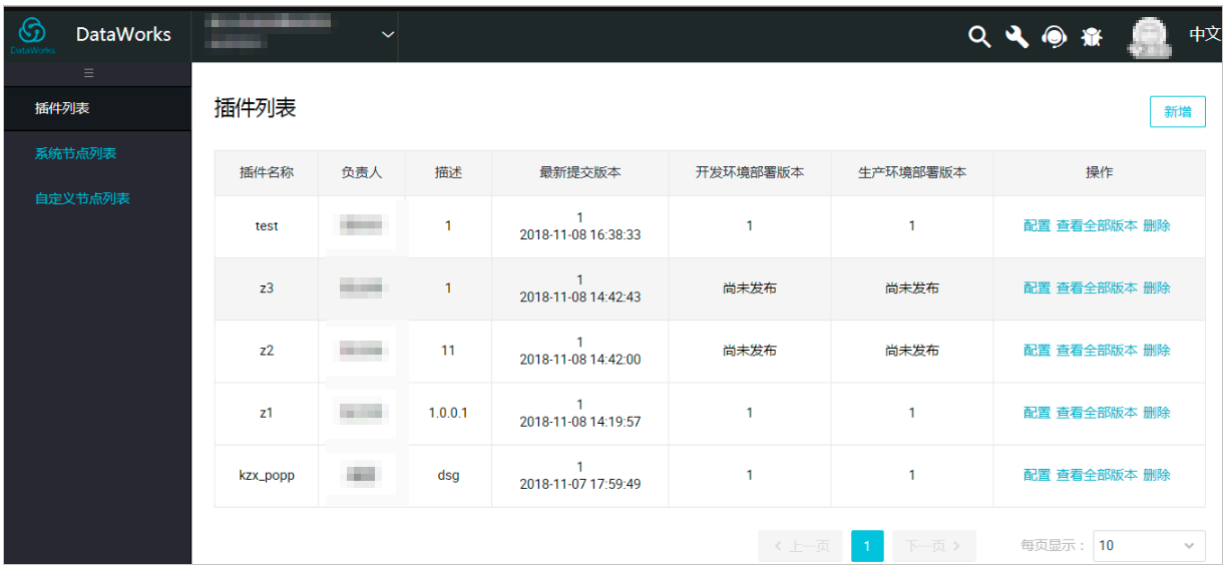
1. 登录DataWorks控制台，单击相应工作空间后的进入数据开发。
2. 单击右上角的节点配置，进入节点配置页面。

> **说明:**
> 仅项目Owner和项目管理员可进行此操作。



插件列表

插件列表将展示您添加的所有插件节点，您可单击右上角的新增，添加自定义插件。



最新提交版本、开发环境部署版本和生产环境部署版本的版本显示逻辑，如下所示：

- 如果节点是新创建的且未进行发布，则显示尚未发布。
- 如果节点已经发布，则会显示具体的版本号和部署时间。
- 如果节点正在发布中，则显示版本号为发布中。

您可以在相应插件后进行配置、查看全部版本和删除等操作。

操作	说明
配置	单击配置后，需要根据插件的状态，进入相关页面，通常展示的是发布到生产环境的页面。
查看全部版本	单击查看全部版本，可对插件所有的历史版本进行查看。  · 查看：单击后跳转至新增自定义节点的基本信息页面。 · 回滚：提示使用老版Jar和配置重新发布插件，但会更新版本号。回滚的版本号为全部版本页面中最大的版本号+1。 · 下载：单击下载，即可下载对应的资源文件。
删除	如果有节点使用此插件且提示报错，则需要删除节点。 说明: 删除插件的前提是没有节点关联此插件。

新增自定义插件

插件是指节点的核心处理逻辑。以ODPS SQL节点为例，您在编辑器中编写的SQL，提交运行后，会用后台对应的插件来进行解析并执行。新增一个自定义节点，首先需要开发自定义插件的处理逻辑，目前仅支持Java语言。

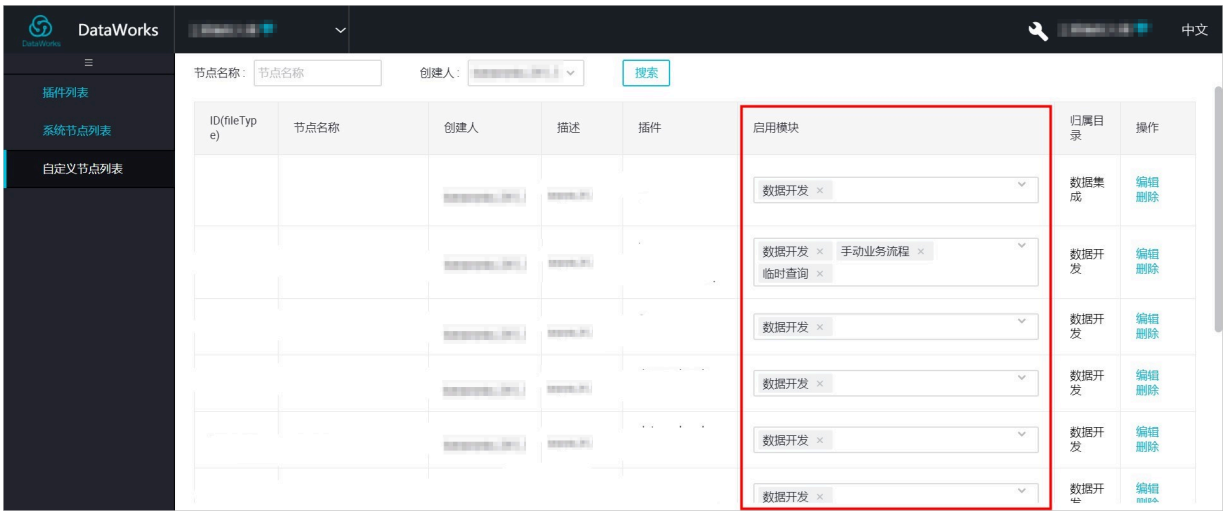
新增自定义插件主要分为基本设置、发布到开发环境、在开发环境进行测试和发布到生产环境四个步骤，详情请参见#unique_346。

系统节点列表

系统节点列表仅为展示界面，目前您不可以进行修改，启动模块默认为数据开发。

自定义节点列表

自定义节点列表中展示项目中所有的自定义节点，单击新增即可添加一个自定义节点，详情请参见#unique_347。



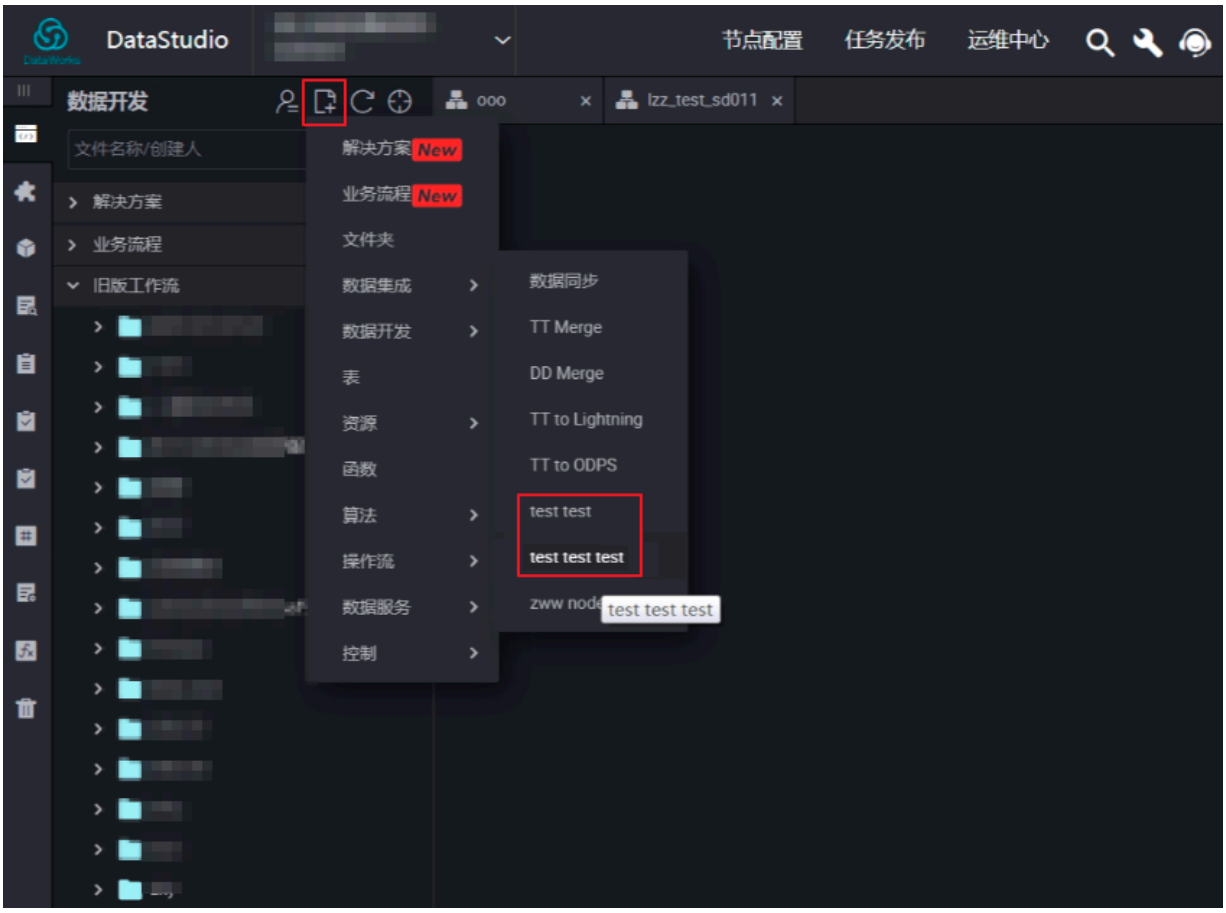
启用模块包括数据开发、手动业务流程和临时查询，您可以根据自身需求进行选择。

目前仅项目Owner或节点创建人可以进行编辑和删除操作。

- 编辑：单击编辑跳转至创建自定义节点页面，您可根据自身需求更改相关的节点。
- 删除：在没有任务使用此节点的前提下，方可直接删除插件。如果有任务使用此节点，会提示报错，您需要先下线任务，方可进行删除操作。

使用自定义节点

创建好自定义节点后，进入数据开发页面。



单击新建按钮，即可找到新添加的自定义节点，创建相应的任务。

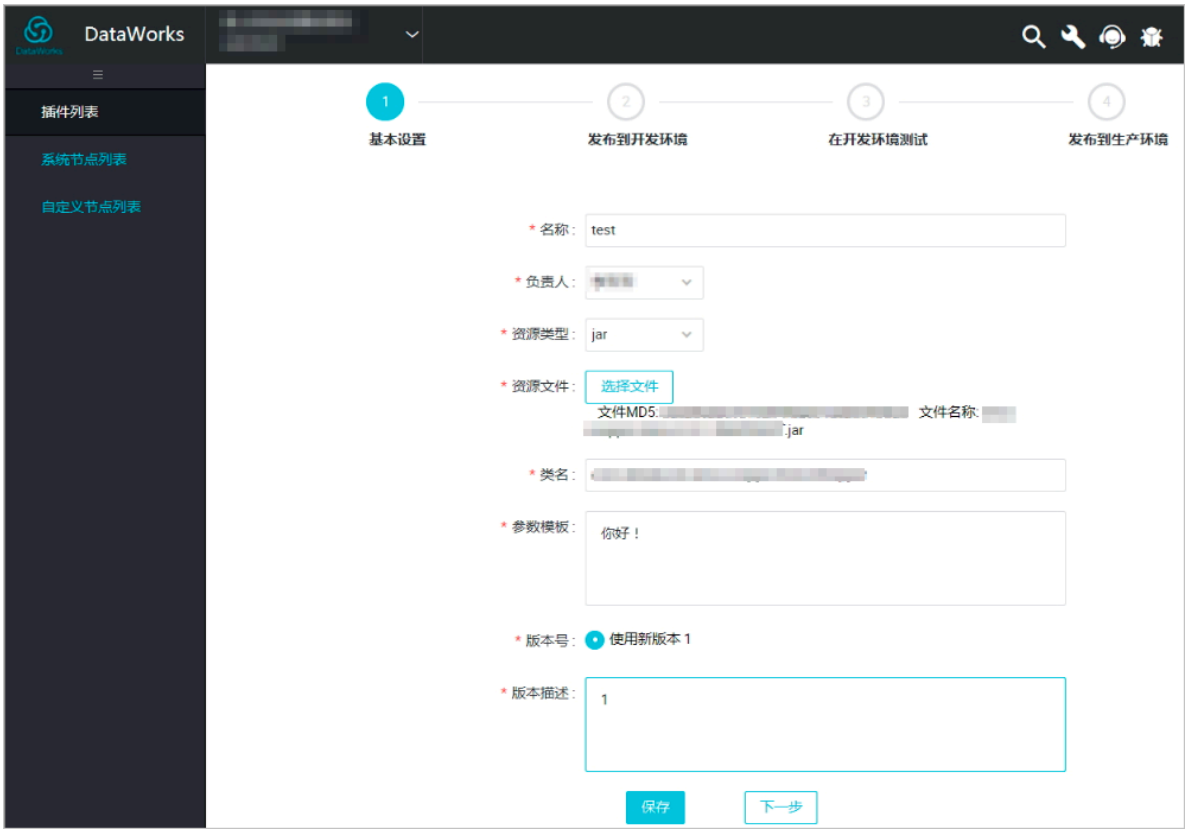
3.5.19.2 新增自定义插件

新增自定义插件包括基本设置、发布到开发环境、在开发环境测试和发布到生产环境四个步骤。

基本设置

1. 进入插件列表页面，单击右上角的新增。

2. 填写基本设置对话框中的配置。



配置	说明
名称	插件名称仅允许字母、下划线和数字，且需以字母开头。
负责人	根据项目的成员进行选择，当选择其他用户时，如果您是管理员，则不能编辑其他用户的自定义插件。如果您是项目owner，则可以编辑其他用户的插件。
资源类型	目前仅支持Java的Jar和压缩包（zip）两种类型，且大小不可以超过50M。
资源文件	提供本地上传和OSS路径两种方式。 说明: 本地文件上传方式最大支持50M，OSS下载方式最大支持200M。
类名	用户插件实现的类全路径名称。
参数模板	根据您上传的Jar包来设计您的参数内容。
版本号	新增时，选择使用新版本。编辑和回滚时，默认选择覆盖当前版本。
版本描述	对插件版本进行简单描述。

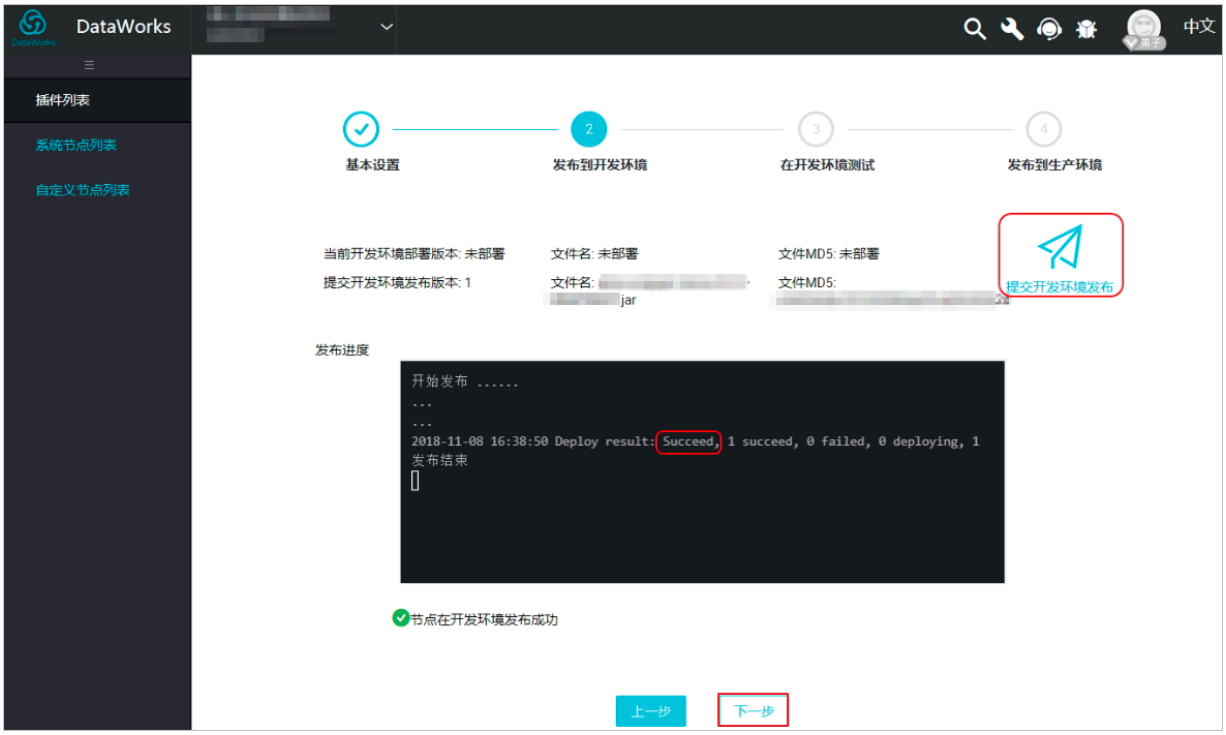
3. 单击保存并进行下一步。

> 说明:
> 单击保存后，可将改动的配置保存至数据库。
>
> · 基本信息修改（非插件包修改）保存后即生效，不需要发布。
> · 如果修改Jar包，必须发布后才会生效。

发布到开发环境

填写好基本设置并单击下一步后，会根据基本设置将相关信息展现出来，您可以通过文件名和文件MD5来判断是否有变化。

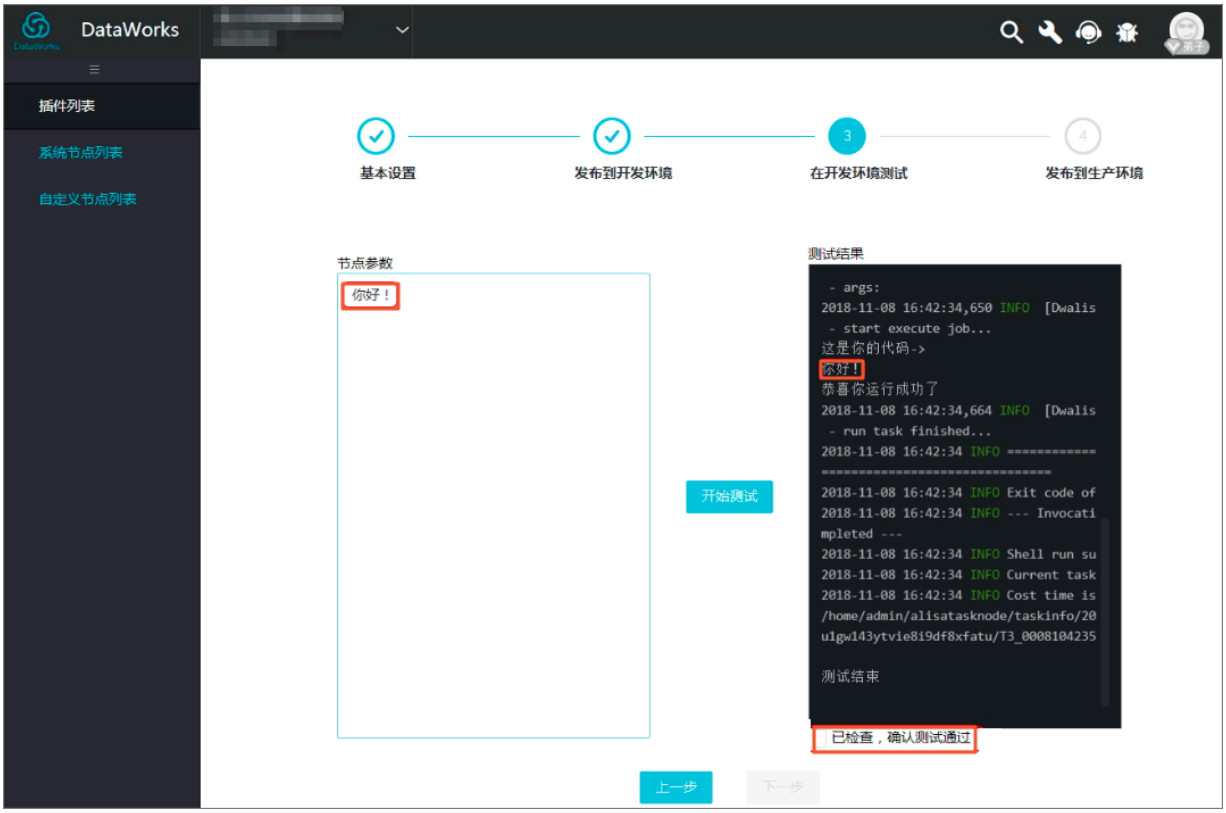
确认后单击提交开发环境发布，您可以实时查看发布进度，发布成功后单击下一步。



在开发环境测试

您可以在节点参数框中给出用于测试的参数，单击开始测试会将参数提交给wrapper进行处理，此处用以验证部署成功与否和插件逻辑的正确性。您也可以在本地进行测试后再提交插件进行发布。

测试完成后，需要自行检查右侧测试结果中输出的日志，来判断测试时是否成功。如果成功，勾选已检查，确认测试通过，然后单击下一步发布到生产环境。

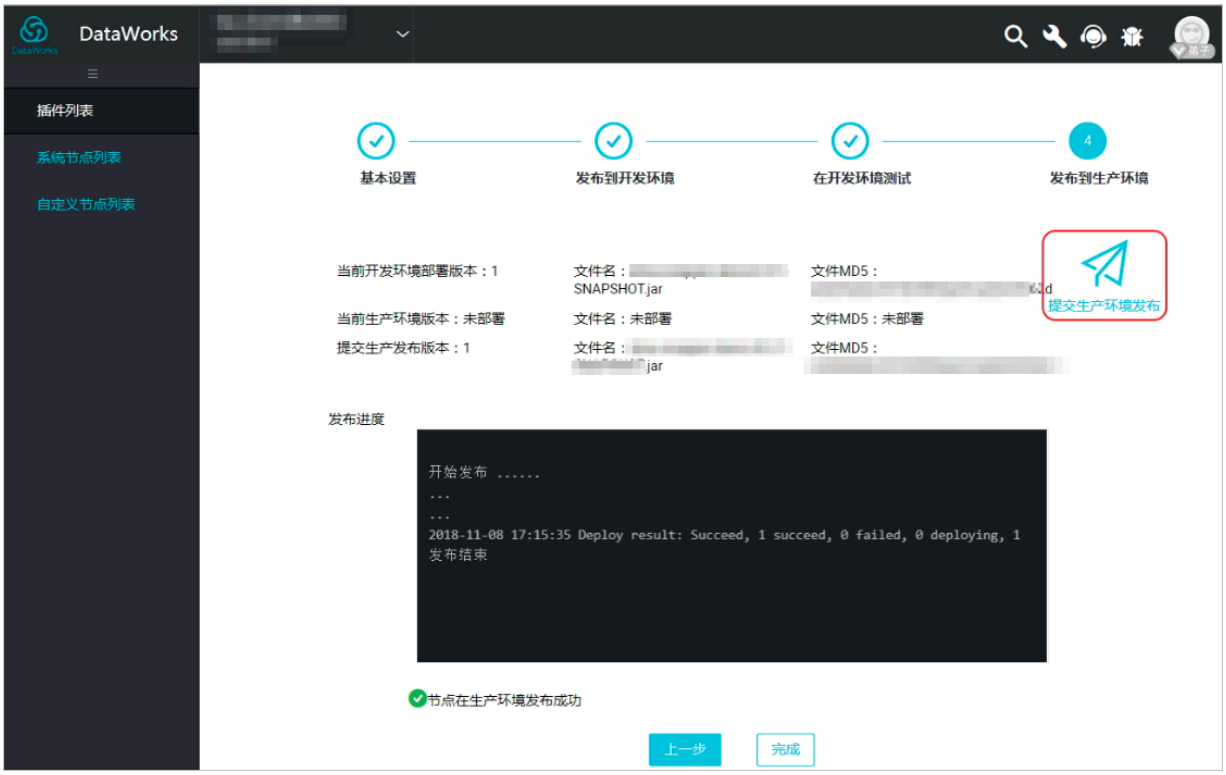


说明:

勾选已检查，确认测试通过，方可进行下一步操作。

发布到生产环境

单击提交生产环境发布后，将会把在开发环境中部署、测试通过的版本提交生产环境发布，您可以实时查看发布进度。



说明:

提交到生产环境的版本必须是开发环境已经部署、测试通过的最新版本，否则生产环境会提示发布失败。

单击完成，即可成功新建自定义插件·，您可以在插件列表查看并编辑您的插件。

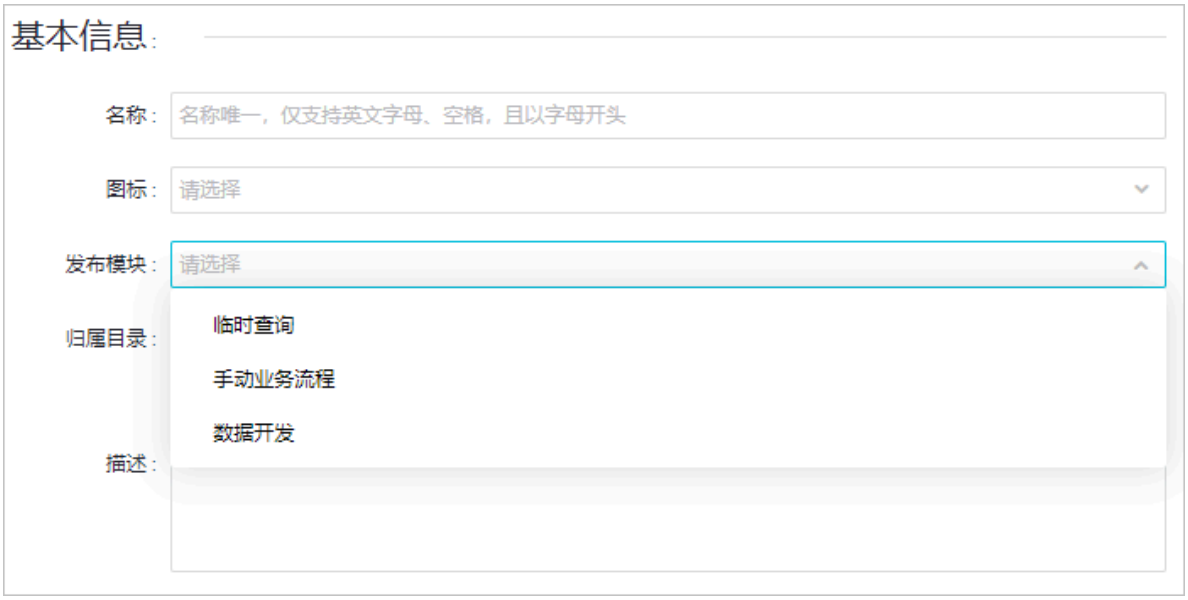
3.5.19.3 新建自定义节点

新增自定义节点包括基本信息、交互配置和插件配置三个步骤。

1. 进入节点配置 > 自定义节点列表页面。

2. 单击右上角的新增。

3. 填写基本信息对话框中的配置。



配置	说明
名称	保存后不允许修改，且在工作空间内唯一。仅支持英文字母、空格和下划线，且长度不得超过20个字符。
图标	选择新增节点图标。
发布模板	目前支持选择临时查询、手动业务流程和数据开发。
归属目录	目前归属目录仅支持业务流程模块和手动业务流程模块。

4. 进行交互配置。

配置	说明
文件右键操作	· 默认选项：重命名、移动、克隆、偷锁、查看历史版本、在运维中心定位、删除和发起Review。 · 可选项：编辑、复制文件名和添加为桌面快捷方式。
顶部操作按钮	· 默认选项：保存、提交、提交并解锁、偷锁编辑、运行、折叠、高级运行、停止、重新加载、在开发环境执行冒烟测试、查看开发环境冒烟测试日志、、执行冒烟测试、前往开发环境的调度系统和格式化。 · 可选项：运维中心、发布和预编译。
编辑器类型	包括编辑器和数据源选择+编辑器两种类型，默认为编辑器。
右侧Tab	· 默认选项：调度配置和版本。 · 可选项：血缘关系和结构。
自动解析	如果开启，调度配置界面会显示相关入口。如果关闭，则调度配置界面没有相关入口。自动解析是根据代码中的血缘关系，解析出本节点的输入及本节点的输出。

5. 进行插件配置。

- 选择编辑器类型。

配置	说明
选择插件	您可以选择发布成功的插件。
编辑器语言类型	目前支持ODPS SQL、JSON、Shell、MySQL、XML等多种类型。
是否使用MaxCompute引擎	如果您的插件需要使用MaxCompute引擎则必须开启，如果不需要则可关闭，默认开启。

- 选择数据源选择+编辑器类型。

配置	说明
选择插件	您可以选择发布成功的插件。
编辑器语言类型	目前支持ODPS SQL、JSON、Shell、MySQL、XML等多种类型。
数据源类型	选择相应的数据源类型。

6. 单击保存并退出，则成功创建自定义节点。您可以直接使用已创建好的节点。

3.5.20 AnalyticDB for MySQL节点

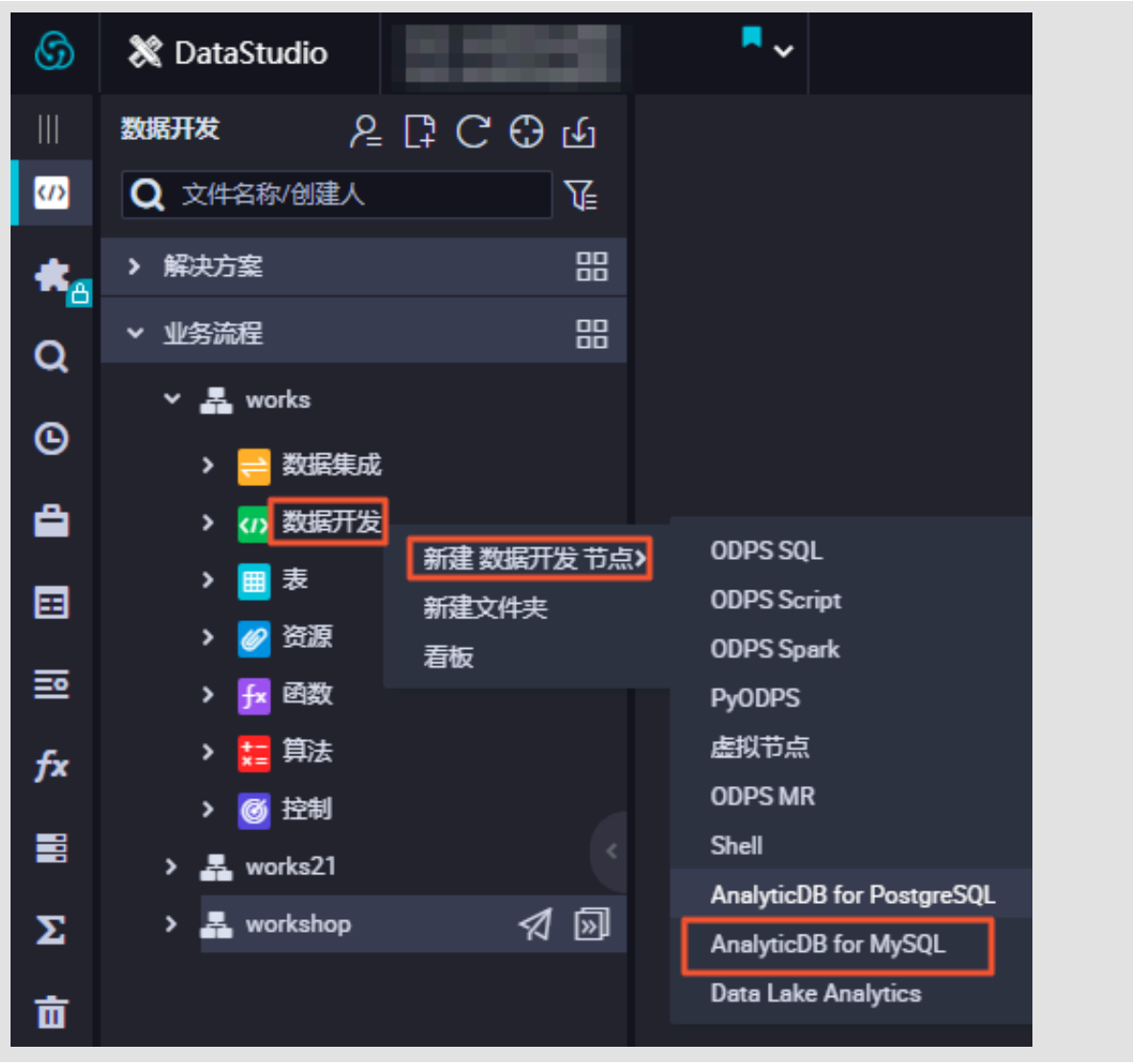
您可以在Dataworks中新建AnalyticDB for MySQL节点，构建在线ETL数据处理流程。

 说明:

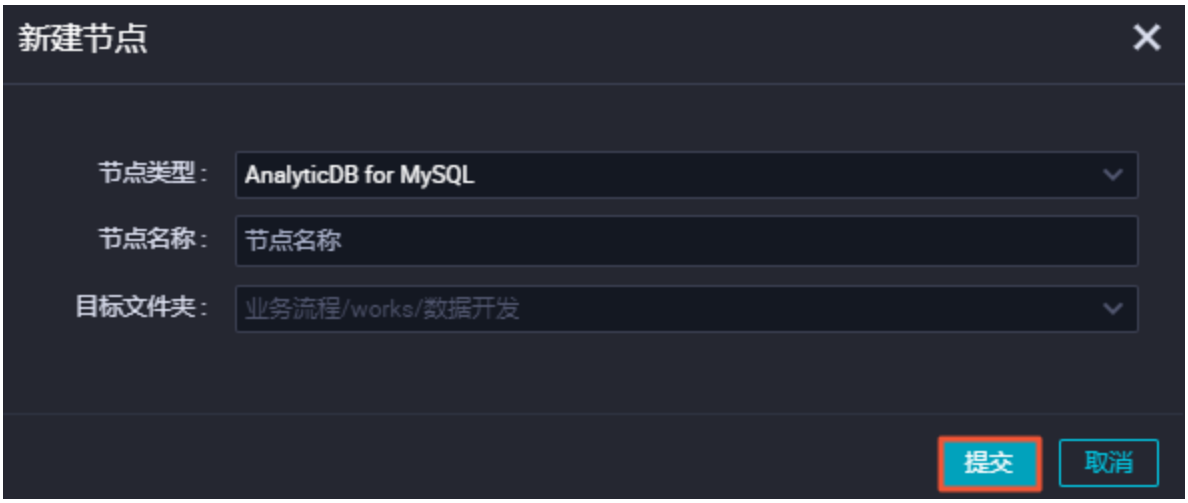
- 建议AnalyticDB for MySQL节点在独享资源组运行，如果在默认资源组运行，会出现网络不通的情况。
- 目前AnalyticDB for MySQL节点仅支持选择生产环境的数据源。

1. 进入DataStudio（数据开发）页面，选择新建 > 数据开发 > AnalyticDB for MySQL。

说明:
您也可以找到相应的业务流程，右键单击数据开发，选择新建数据开发节点 > AnalyticDB for MySQL。



2. 在新建节点对话框中，填写节点名称，选择目标文件夹（用于节点代码分类管理，可以不选），单击提交。

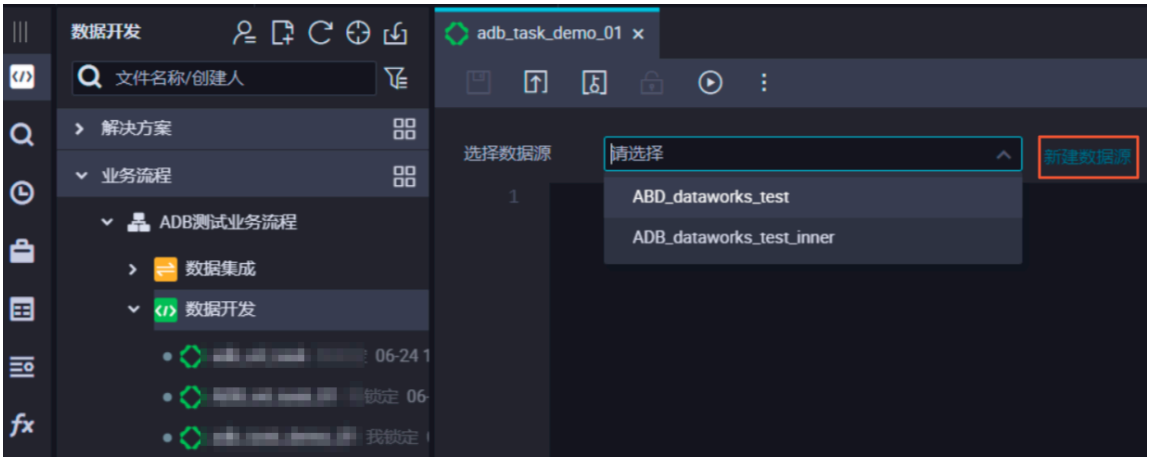


3. 编辑AnalyticDB for MySQL节点。

代码编辑页面分为选择数据源和编辑SQL代码两部分。

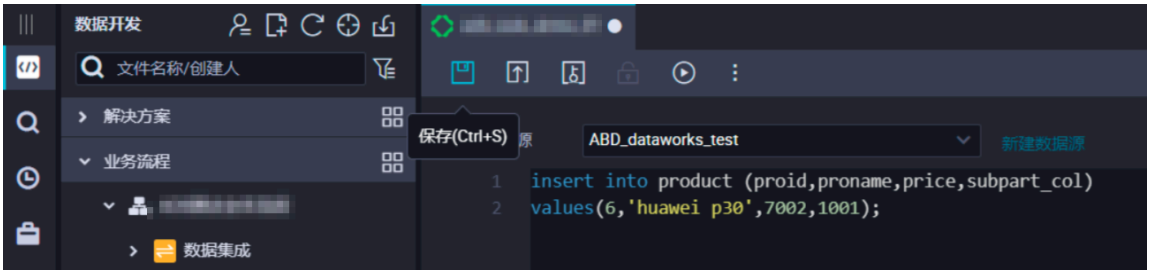
a. 选择数据源。

选择任务要执行的目标数据源。如果下拉选项中没有需要的数据源，单击右侧的新建数据源，前往新建数据源页面进行新建，详情请参见数据源配置。



b. 编辑SQL语句。

选择相应的数据源后，即可根据AnalyticDB for MySQL支持的语法，编写SQL语句。通常支持DML语句，您也可以执行DDL语句。



c. 保存并执行SQL语句。

代码编辑完成后，单击保存按钮，将其保存至服务器。然后单击运行按钮，即可立即执行编辑的SQL语句。

4. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）至开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

   具体操作请参见#unique_314。

您可以通过AnalyticDB for MySQL节点进行最佳实践，详情请参见#unique_352。

3.5.21 Data Lake Analytics节点

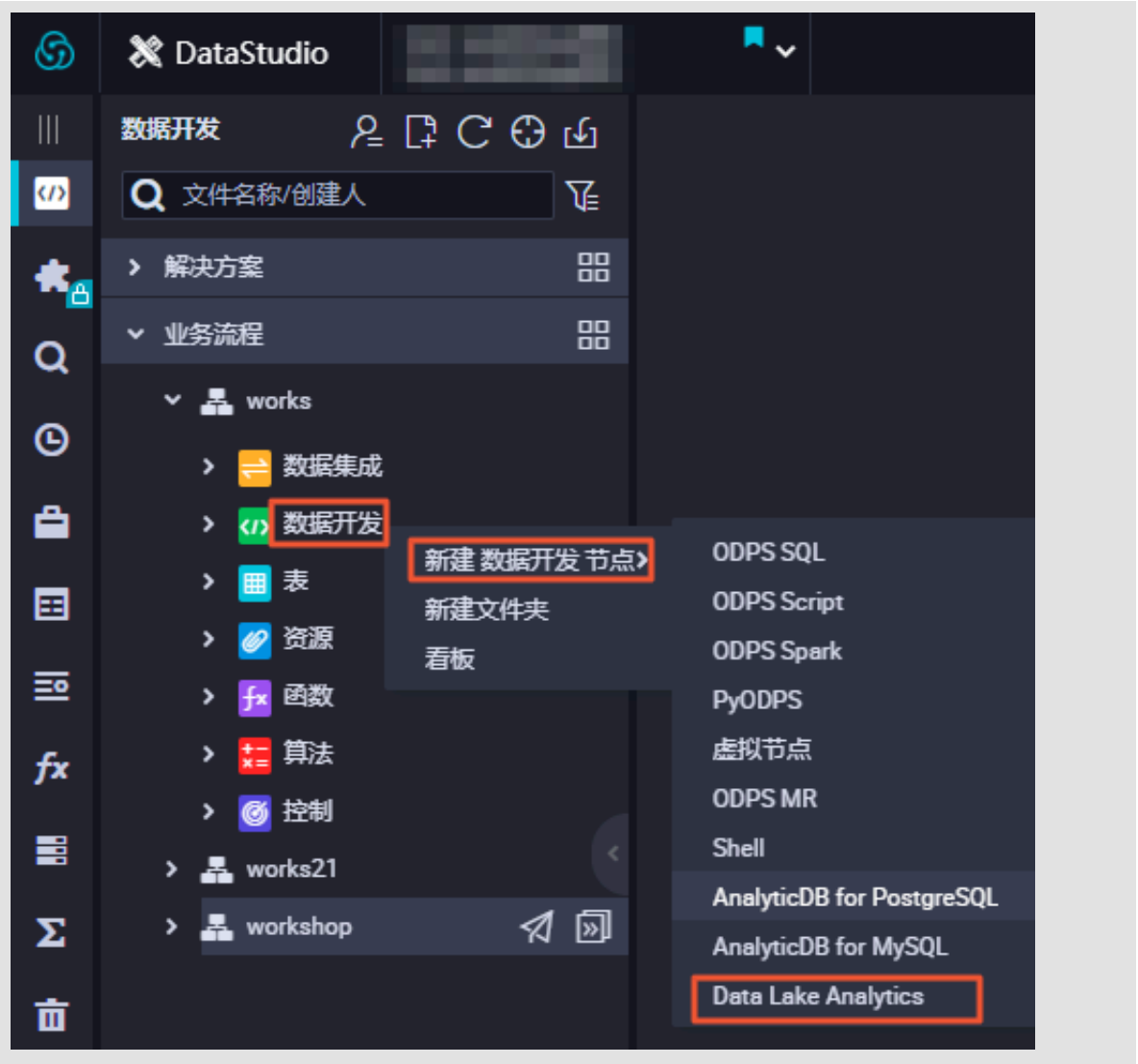
您可以在Dataworks中新建Data Lake Analytics节点，构建在线ETL数据处理流程。

> 说明:
>
> - 建议Data Lake Analytics节点在独享资源组运行，如果在默认资源组运行，会出现网络不通的情况。
> - 目前Data Lake Analytics节点仅支持选择生产环境的数据源。

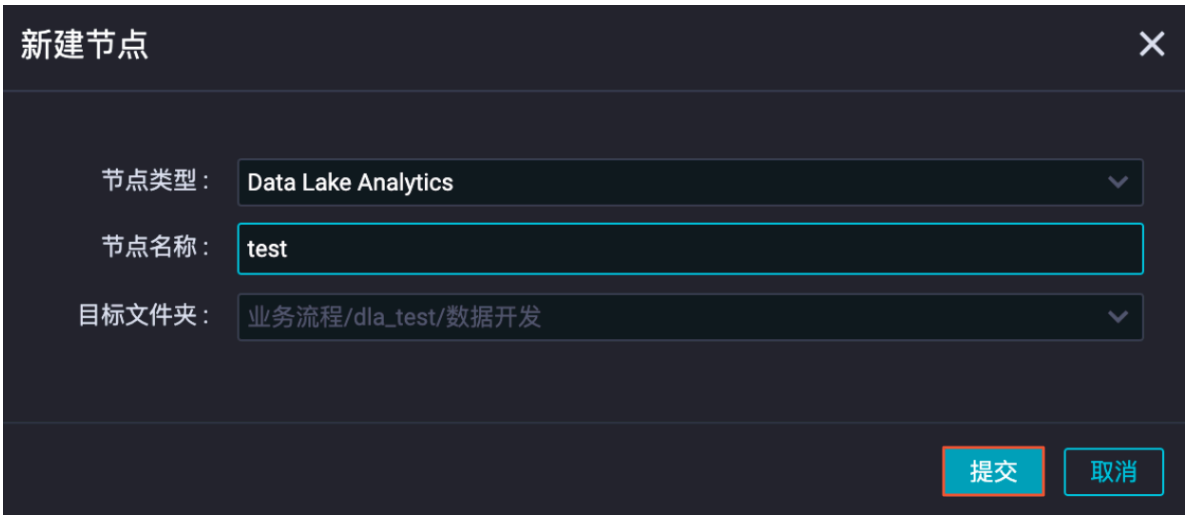
1. 进入DataStudio（数据开发）页面，选择新建 > 数据开发 > Data Lake Analytics。

说明:

您也可以找到相应的业务流程，右键单击数据开发，选择新建数据开发节点 > Data Lake Analytics。



2. 在新建节点对话框中，填写节点名称，选择目标文件夹（用于节点代码分类管理，可以不选），单击提交。

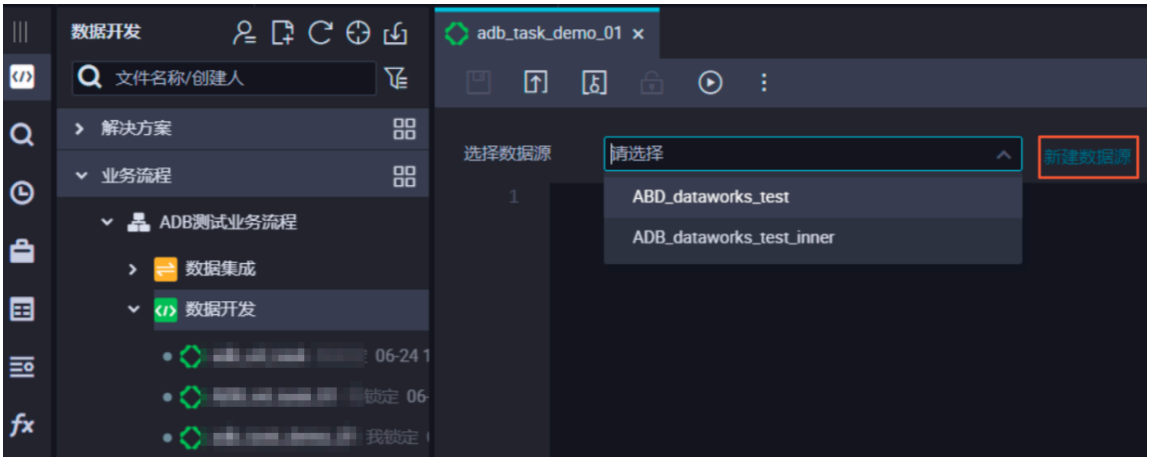


3. 编辑Data Lake Analytics节点。

   代码编辑页面分为选择数据源和编辑SQL代码两部分。

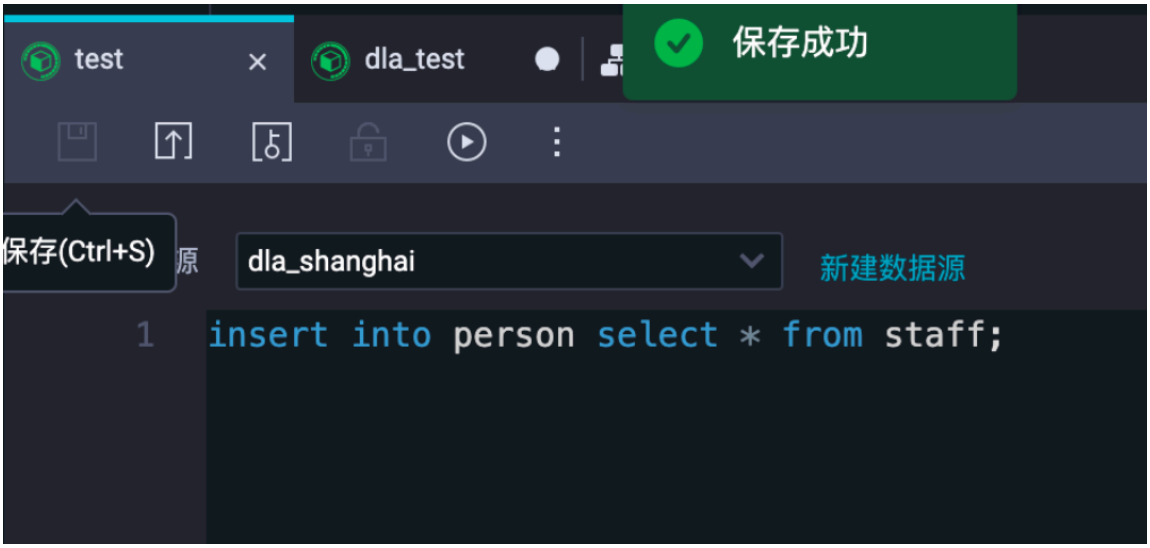
   a. 选择数据源。

      选择任务要执行的目标数据源。如果下拉选项中没有需要的数据源，单击右侧的新建数据源，前往新建数据源页面进行新建，详情请参见数据源配置。



   b. 编辑SQL语句。

      选择相应的数据源后，即可根据Data Lake Analytics支持的语法，编写SQL语句。通常支持DML语句，您也可以执行DDL语句。



   c. 保存并执行SQL语句。

      代码编辑完成后，单击保存按钮，将其保存至服务器。然后单击运行按钮，即可立即执行编辑的SQL语句。

4. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

   具体操作请参见#unique_314。

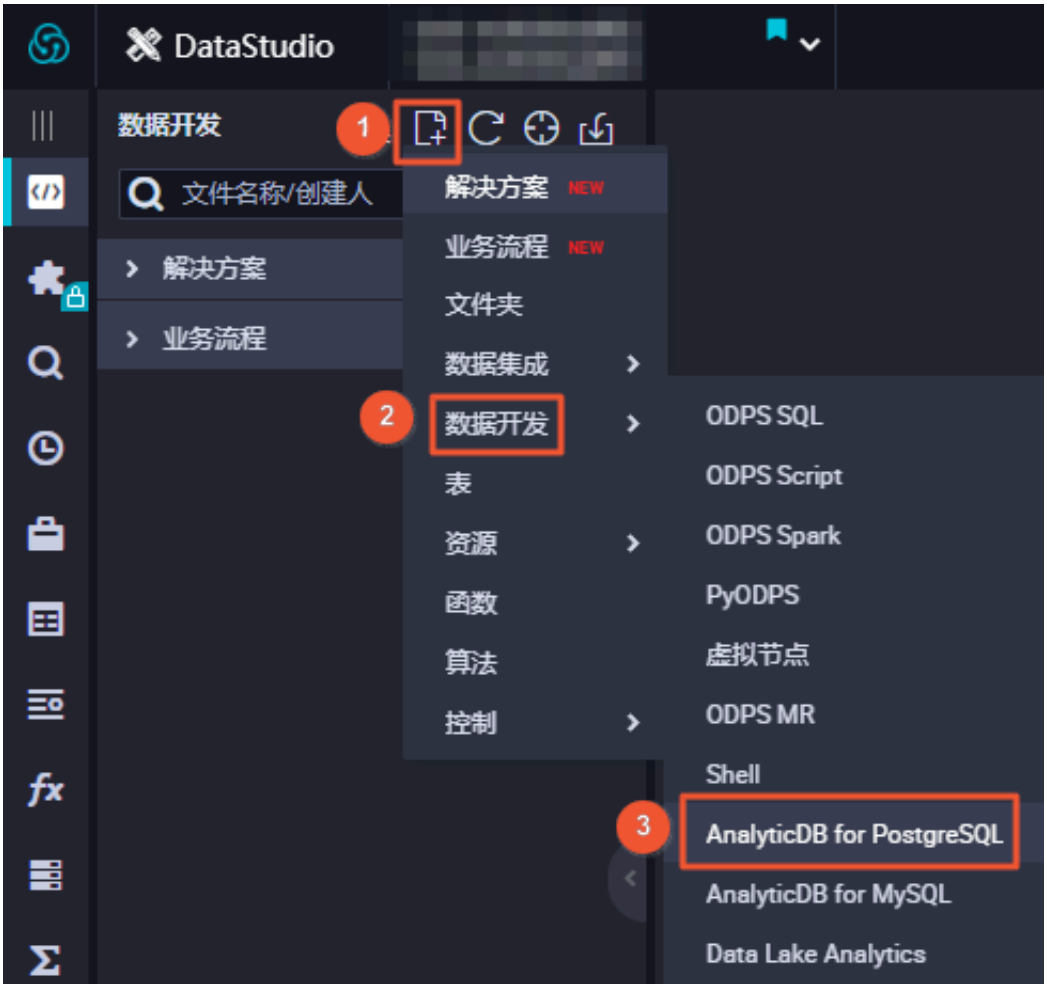
3.5.22 AnalyticDB for PostgreSQL节点

您可以在Dataworks中新建AnalyticDB for PostgreSQL节点，构建在线ETL数据处理流程。

说明:

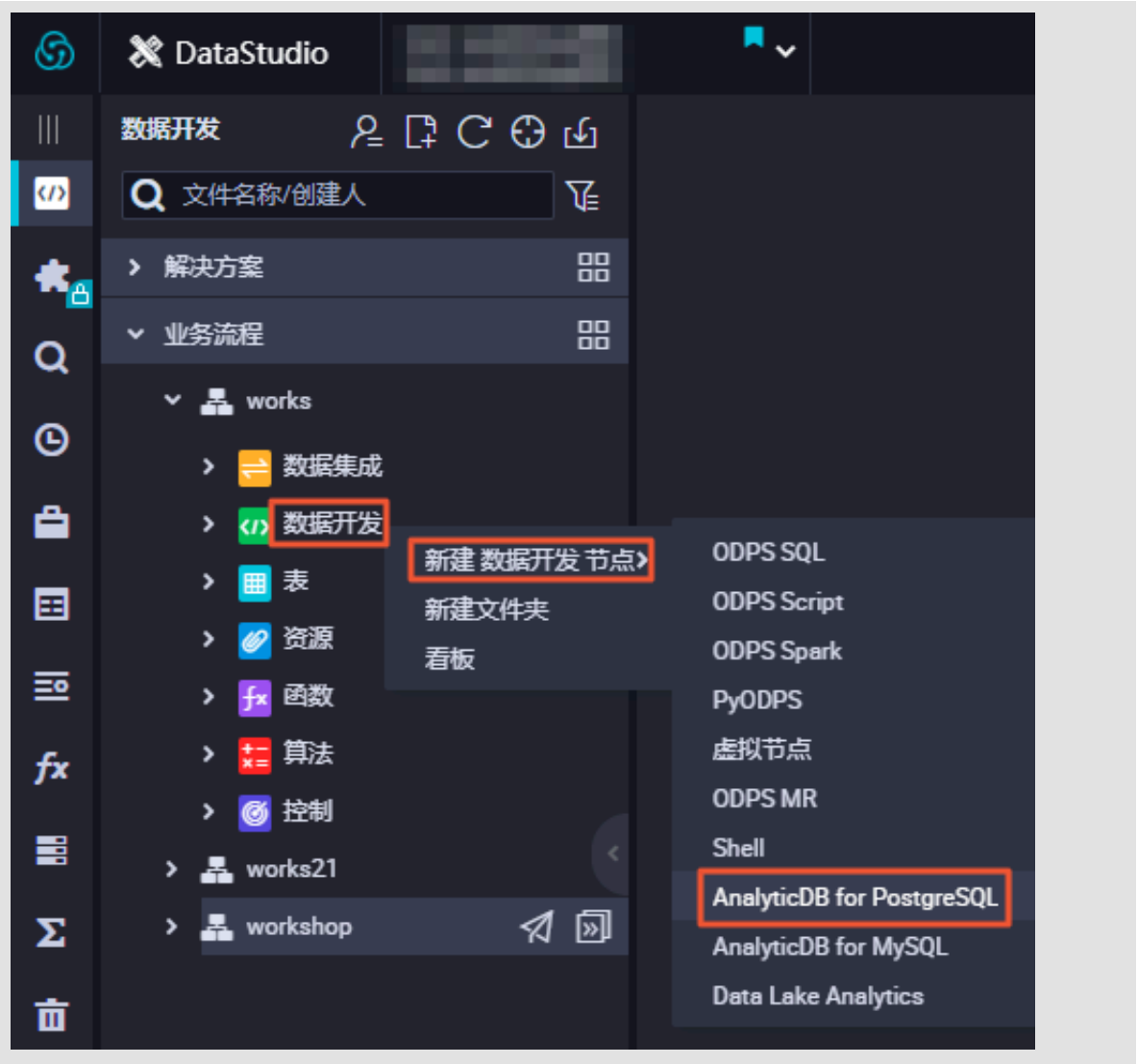
- 建议AnalyticDB for PostgreSQL节点在独享资源组运行，如果在默认资源组运行，会出现网络不通的情况。
- 目前AnalyticDB for PostgreSQL节点仅支持选择生产环境的数据源。

1. 进入DataStudio（数据开发）页面，选择新建 > 数据开发 > AnalyticDB for PostgreSQL。



说明:

您也可以找到相应的业务流程，右键单击数据开发，选择新建数据开发节点 > AnalyticDB for PostgreSQL。



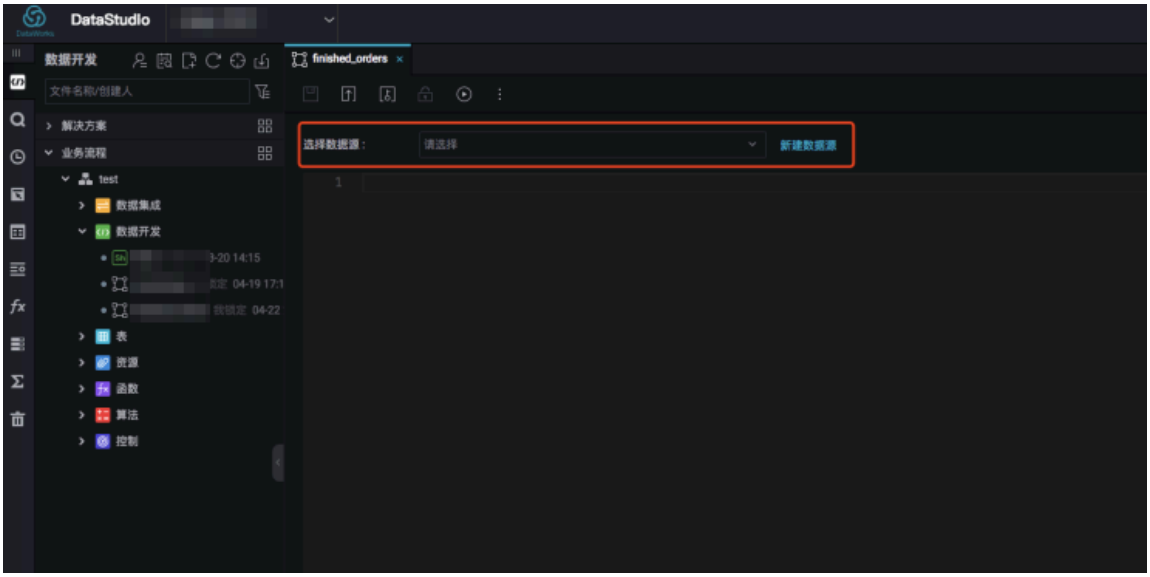
2. 在新建节点对话框中，填写节点名称，选择目标文件夹（用于节点代码分类管理，可以不选），单击提交。

3. 编辑AnalyticDB for PostgreSQL节点。

   代码编辑页面分为选择数据源和编辑SQL代码两部分。

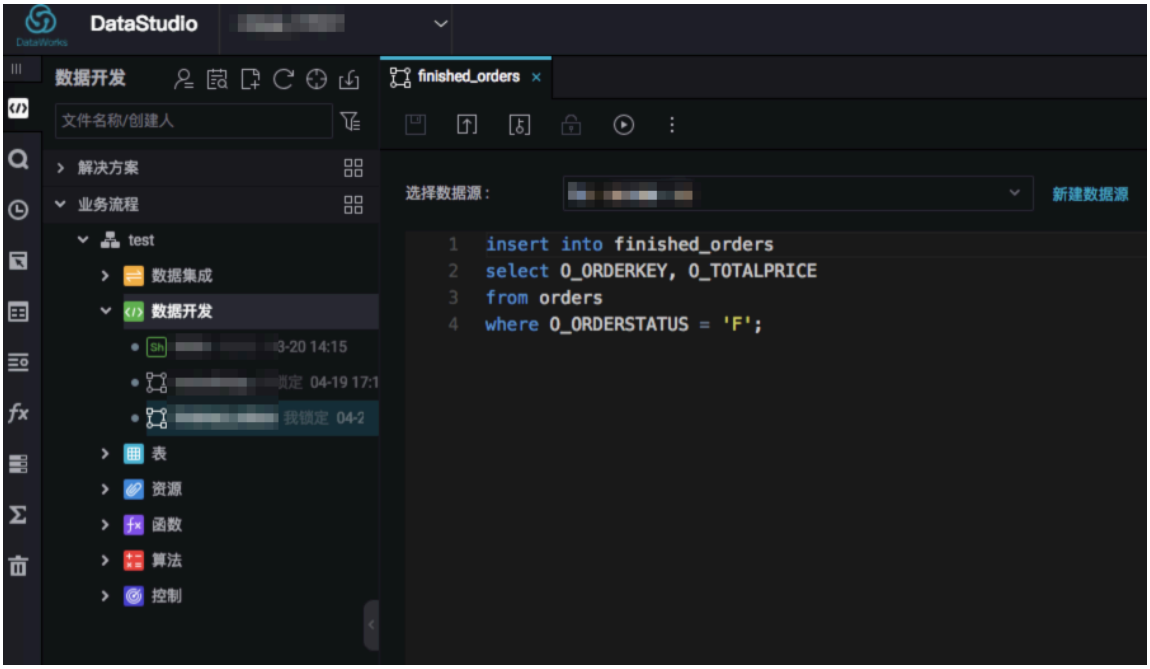
   a. 选择数据源。

      选择任务要执行的目标数据源。如果下拉选项中没有需要的数据源，单击右侧的新建数据源，前往新建数据源页面进行新建，详情请参见数据源配置。



   b. 编辑SQL语句。

      选择相应的数据源后，即可根据PostgreSQL支持的语法，编写SQL语句。通常支持DML语句，您也可以执行DDL语句。

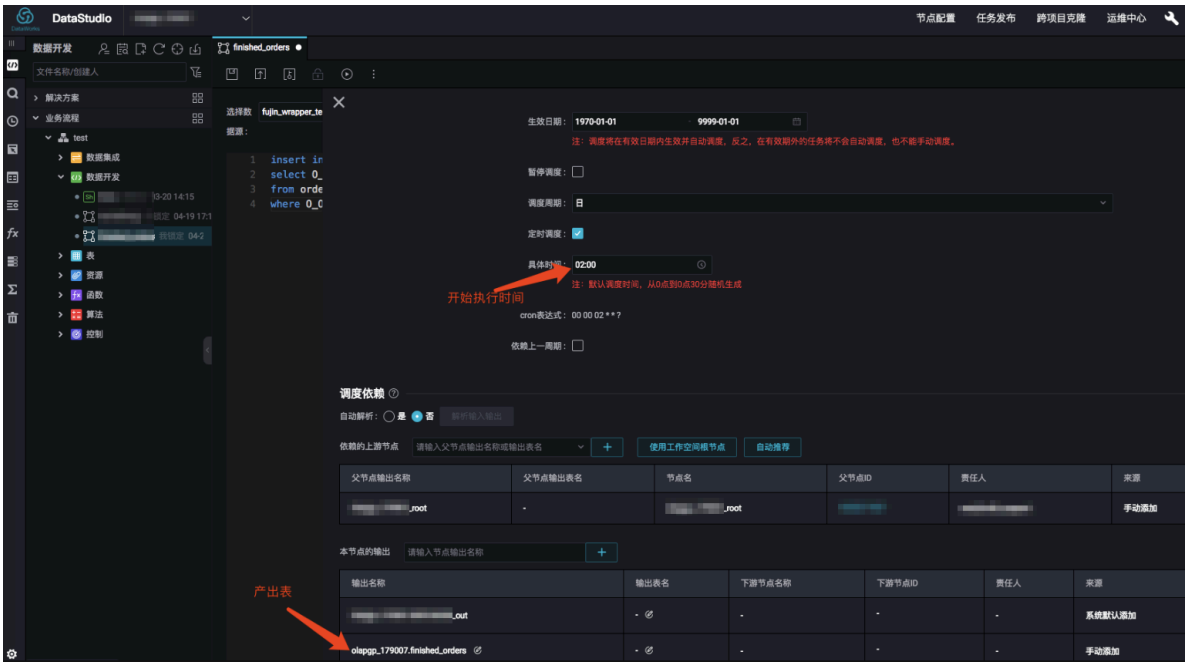


   c. 保存并执行SQL语句。

代码编辑完成后，单击保存按钮，将其保存至服务器。然后单击运行按钮，即可立即执行编辑的SQL语句。

4. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。



5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）至开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

   具体操作请参见#unique_314。

3.6 调度配置

3.6.1 基础属性

您可以在调度配置页面进行基础属性、时间属性、调度依赖和节点上下文的设置。

打开节点任务，单击页面右侧的调度配置，即可进行基础属性的配置。

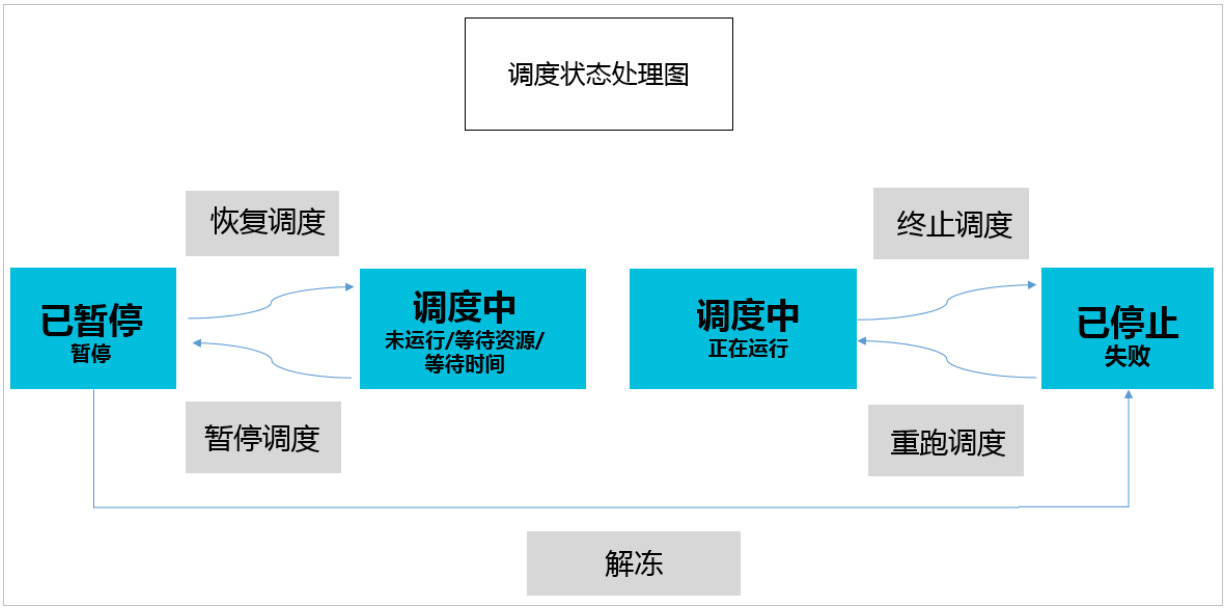
配置	说明
节点名	新建节点时填写的节点名，您可以在目录树右键单击节点，选择重命名进行修改。
节点ID	任务提交后会生成唯一的节点ID，不可修改。
节点类型	新建节点时选择的节点类型，不可修改。
责任人	新建的节点责任人默认是当前登录的用户，您可以修改责任人。 说明: 只能选择当前工作空间下的成员为责任人。
描述	通常用于描述节点业务、用途。
参数	任务调度时，给代码中的变量赋值。

各种节点类型参数赋值格式

- ODPS SQL、ODPS MR类型：参数赋值格式为`变量名1=参数1 变量名2=参数2`，多个参数之间用空格分隔。
- Shell类型：参数赋值格式为`参数1 参数2`，多个参数之间用空格分隔。

调度内置了一些常用的时间参数，具体参数说明请参见#unique_39。

调度状态处理图



3.6.2 参数配置

常见的数据研发场景中，各类型的任务代码不是一次性写入后便持续不变地进行调用。需要根据需求变化、时间变化等，动态传入某些值（例如日期时间），替换变量后再进行计算。

DataWorks的参数配置功能可以满足该业务场景的需求。您在配置参数后，即可赋予周期调度运行的任务自动解析出所需值。目前参数分为系统参数和自定义参数（推荐）两大类。

> 说明:
> 如果您是初次接触参数配置，建议您首先观看DataWorks V2.0数据开发功能与用法解析视频进行学习。

参数类型

参数类型	调用方式	适用类型	参数编辑框示例
系统参数：主要包括bdp.system.bizdate和bdp.system.cyctime	在调度系统中运行时，无需在编辑框设置，可以直接在代码中引用 ${bdp.system.bizdate}和${bdp.system.cyctime}，系统将自动替换这两个参数的取值。	全部节点类型	无

参数类型	调用方式	适用类型	参数编辑框示例
非系统参数：自定义参数（推荐使用）	在代码中引用${key1}和${key2}，然后在参数编辑框中进行设置。例如“key1=value1 key2=value2”。	非Shell节点类型	· 常量参数：param1=”abc” param2=1234。 · 变量参数：param1=$[yyyymmdd]，结果将基于bdp.system.cyctime的取值计算。
	在代码中引用$1、$2和$3，然后在参数编辑框中进行设置。例如“value1 value2 value3”。	Shell节点类型	· 常量参数：”abc” 1234。 · 变量参数：$[yyyymmdd]，结果将基于bdp.system.cyctime的取值计算。

由上表可见，自定义参数中的变量参数类型的取值基于系统参数。您可以灵活地通过自定义变量参数，来定义获取的部分与格式。对于非系统参数，括号使用区别如下：

· 大括号{ }：对应业务时间，例如{yyyymmdd}将基于bdp.system.bizdate取值。
· 中括号[ ]：对应运行时间，例如[yyyymmddhh]将基于bdp.system.cyctime取值。

说明：

任务只有在生产环境才会被调度，所以调度参数只有生产环境调度运行后才会被替换。

如果您需要验证配置的调度参数在调度中替换的值是否符合预期，请使用冒烟测试功能，详情请参见开发环境冒烟测试。

参数需要在调度配置 > 基本属性 > 参数中赋值（下文简称为调度变量赋值），参数配置时请注意以下问题：

· 参数=两边不可以加空格。例如bizdate=$bizdate。
· 如果存在多个参数，需要使用空格分隔每个参数。例如bizdate=$bizdate datetime=${yyyymmdd}。

系统参数

DataWorks提供了两个系统参数，定义如下：

- ${bdp.system.cyctime}：定义为一个实例的定时运行时间，默认格式为yyyymmddhh24miss。仅有运行时间可以取到具体的小时、分钟时间。
- ${bdp.system.bizdate}：定义为一个实例计算时对应的业务日期，业务日期默认为运行日期的前一天，默认以yyyymmdd的格式显示。

运行时间和业务日期的计算公式为运行时间=业务日期+1。

如果使用系统参数，无需在编辑框设置，直接在代码中引用${bizdate}即可，系统将自动替换代码中对这个参数的引用字段。

 说明:
一个周期任务的调度属性，配置的是运行时间的定时规律，因此可以根据实例的定时运行时间反推业务日期，从而得知每个实例中参数的取值。

PyODPS节点的调度参数配置与普通节点稍有不同，详情请参见#unique_305。

系统参数示例

假设您需要设置一个ODPS_SQL任务为小时调度，每天00:00-23:59时间段里每隔1小时执行一次。如果想在代码中使用系统参数，请参见下述步骤进行操作。

1. 在代码中直接引用系统参数，节点代码如下所示：

```
insert overwrite table tb1 partition(ds ='20150304') select
c1,c2,c3
from (
select * from tb2
where ds ='${bdp.system.cyctime}') t
full outer join(
select * from tb3
where ds = '${bdp.system.bizdate}') y
```

```
on t.c1 = y.c1;
```

2. 完成上述步骤后，您的任务已经通过系统参数完成了分区的设置。接下来您可以设置时间属性和调度依赖，详情请参见时间属性调度依赖，本示例设置调度周期为小时。

3. 设置好周期与依赖后，提交任务，您可以在运维中心看到您的任务。从第二天开始，您的任务在运行时会产生周期实例，右键单击查看运行日志，即可看到系统参数被实际解析出来的时间。

```
2019-01-14 11:37:09 sql:
set SKYNET_ID=700002085435;
set SKYNET_BIZDATE=20190113;
set SKYNET_PACKAGEID=-;
set SKYNET_CYCTIME=20190114010000;
set SKYNET_NODENAME=test1;
set SKYNET_TASKID=703561808615;
set SKYNET_SYSTEMID=-;
set SKYNET_PRIORITY=1;
set SKYNET_JOBID=700008453606;
set SKYNET_ONDUTY=1079926896999421;
set SKYNET_TENANT_ID=272451677051138;
set WRAPPER_SQLSEQ=1;
set WRAPPER_RETRYNUM=1;
set biz_id=700002085435_20190113_703561808615_700008453606_1_MaxCompute_DOC_1079926896999421_272451677051138;
insert overwrite table tb1 partition(ds ='20150304') select
c1,c2,c3
from (
select * from tb2
where ds ='20190114010000') t
full outer join(
select * from tb3
where ds = '20190113') y
on t.c1 = y.c1;
```

例如在2019年01月14日这天，调度系统为该任务生成了24个运行实例。业务日期应当全部为2019年01月13日（运行日期-1天），因此${bdp.system.bizdate}全部显示为20190113。运

行时间则为运行日期+定时时间，因此 ${bdp.system.cyctime}显示为20190114000000+每个实例自己的定时时间。

打开每个实例的运行日志后，搜索代码中的替换情况如下：

- 第一个实例定时时间为2019-01-14 00:00:00，则bdp.system.bizdate替换的结果为20190113， bdp.system.cyctime替换的结果为20190114000000。
- 第二个实例定时时间为2019-01-14 01:00:00，则bdp.system.bizdate替换的结果为20190113，bdp.system.cyctime替换的结果为20190114010000，即上图中显示的情况。
- 以此类推，第24个实例定时时间为2019-01-14 23:00:00，则bdp.system.bizdate替换的结果为20190113，bdp.system.cyctime替换的结果为20190114230000。

非Shell节点自定义参数

在非Shell节点中配置调度参数，需要先在代码里${变量名}（表示引用函数），然后在调度参数的赋值中输入具体的值。

> 说明:
> SQL代码中的变量名命名只支持英文的a-z、A-Z、数字和下划线。变量名为date固定会自动赋$bizdate值（详见调度内置参数列表），不需要在调度参数配置处赋值，即便赋值了也不会替换到代码中，代码默认替换的还是$bizdate。

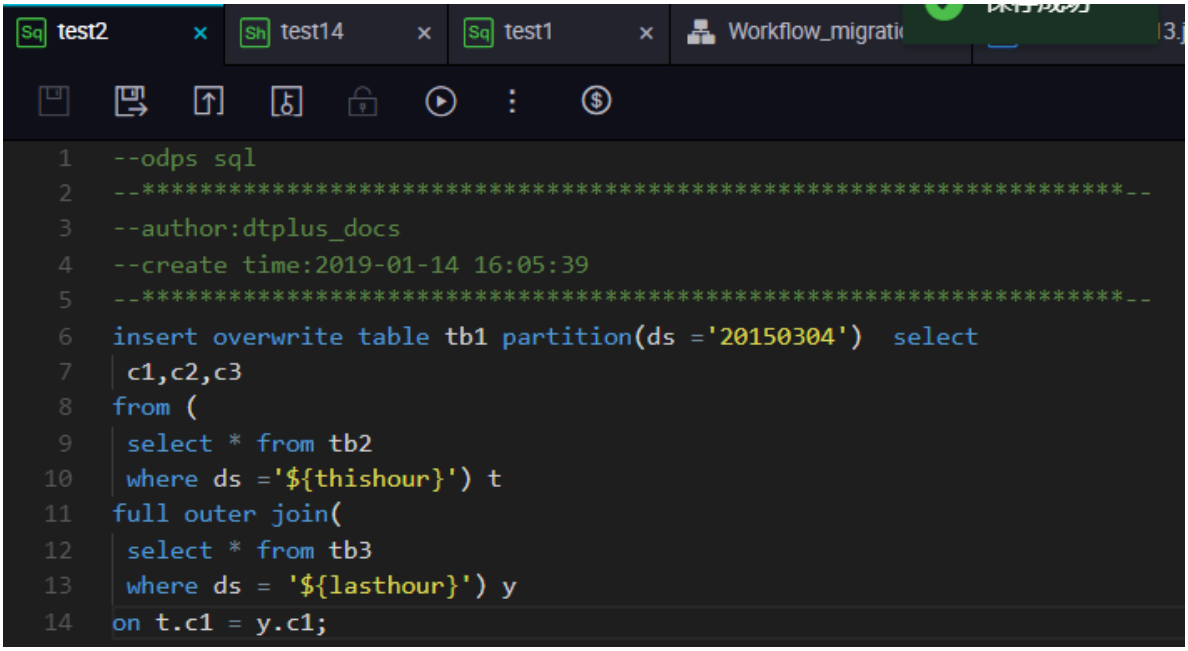
非Shell节点自定义参数示例

设置一个ODPS_SQL任务为按小时调度，每天在00:00-23:59的时间段中，每隔1小时执行一次。如果您想要在代码中使用关于小时的自定义变量参数thishour和lasthour，则应按照如下步骤进行操作。

1. 在代码中引用参数，代码如下所示。

```
insert overwrite table tb1 partition(ds ='20150304')  select
 c1,c2,c3
from (
 select * from tb2
 where ds ='${thishour}') t
full outer join(
 select * from tb3
 where ds = '${lasthour}') y
```

```
on t.c1 = y.c1;
```



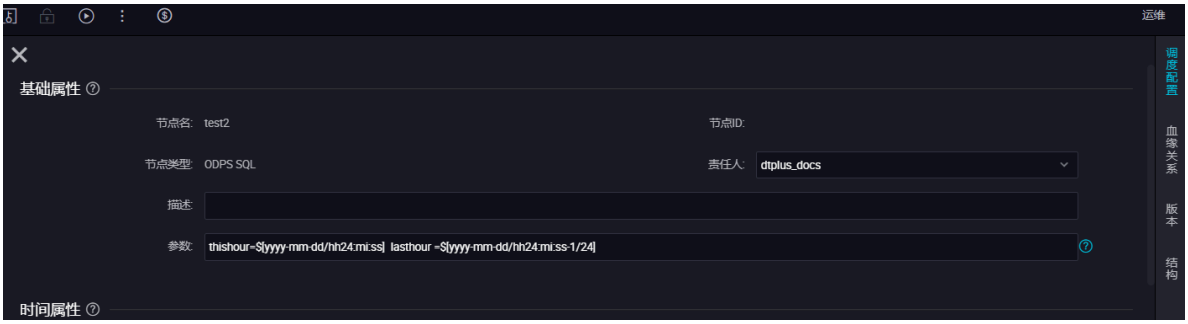
2. 代码中引用的变量，需要调度配置 > 基本属性 > 参数中赋值。

   自定义参数配置如下：

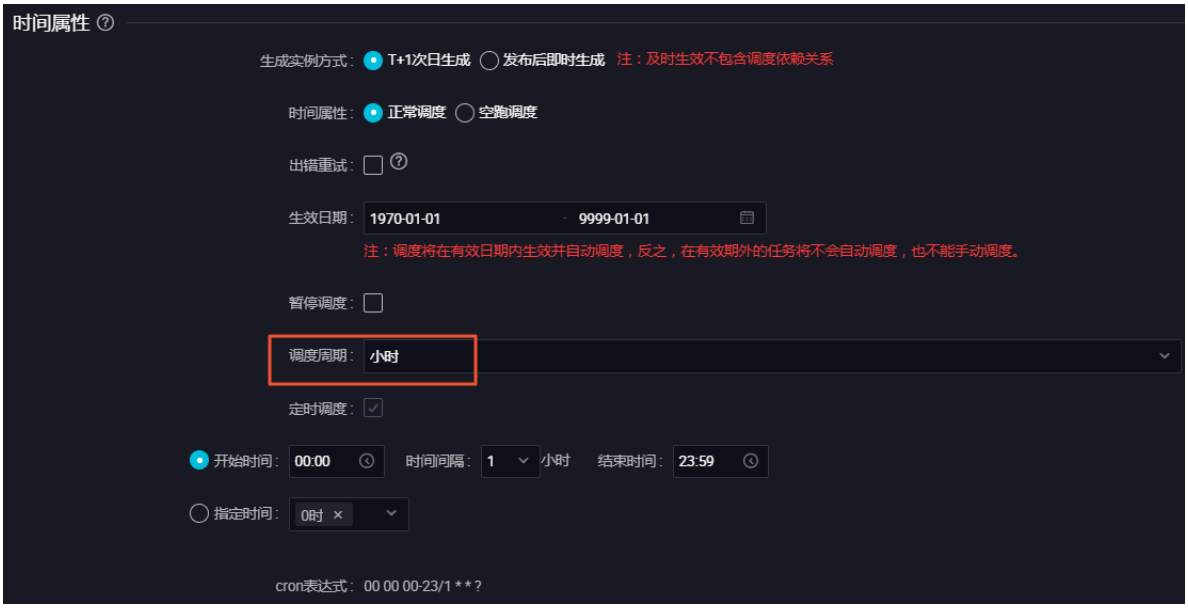
   - thishour=$[yyyy-mm-dd/hh24:mi:ss]
   - lasthour=$[yyyy-mm-dd/hh24:mi:ss-1/24]

   > 说明：
   > yyyy-mm-dd/hh24:mi:ss参数对应cyctime时间，详请请参见自定义参数变量。

   您可以直接在参数一栏中输入`thishour=$[yyyy-mm-dd/hh24:mi:ss] lasthour=$[yyyy-mm-dd/hh24:mi:ss-1/24]`。



3. 设置时间周期为每小时运行一次。



4. 设置好周期与依赖后，提交任务，您可以在运维中心看到您的任务。从第二天开始，您的任务在运行时会产生周期实例，右键实例单击查看运行日志，即可看到自定义参数被实际解析出来的时间：由于CYCTIME为20190114010000，因此解析出来的thishour时间为2019-01-14/01:00:00，而lasthour代表上一个小时，即2019-01-14/00:00:00。

```
2019-01-14 16:14:56 INFO ================================================================
2019-01-14 16:14:58 config file is ok!!!
2019-01-14 16:14:58 sql:
set SKYNET_ID=700002086517;
set SKYNET_BIZDATE=20190113;
set SKYNET_PACKAGEID=-;
set SKYNET_CYCTIME=20190114010000;
set SKYNET_NODENAME=test2;
set SKYNET_TASKID=703562159690;
set SKYNET_SYSTEMID=-;
set SKYNET_PRIORITY=1;
set SKYNET_JOBID=700008460606;
set SKYNET_ONDUTY=1079926896999421;
set SKYNET_TENANT_ID=272451677051138;
set WRAPPER_SQLSEQ=1;
set WRAPPER_RETRYNUM=1;
set biz_id=700002086517_20190113_703562159690_700008460606_1_MaxCompute_DOC_1079926896999421_272451677051138;
insert overwrite table tb1 partition(ds ='20150304')  select
 c1,c2,c3
from (
 select * from tb2
 where ds ='2019-01-14/01:00:00') t
full outer join(
 select * from tb3
 where ds = '2019-01-14/00:00:00') y
on t.c1 = y.c1;
```

Shell节点自定义参数

Shell节点的参数配置和非Shell节点配置的步骤一样，只是规则有所不同：Shell节点中的变量不允许自定义命名，只能以$1,$2,$3…命名。Shell类型节点，代码中Shell语法声明：$1，调度中节点配置参数配置：$xxx（调度内置参数），即$xxx的值替换代码中的$1。

> **说明:**
> 在Shell节点中，参数到达第10个以后，应该使用 ${10} 的方式来声明变量。

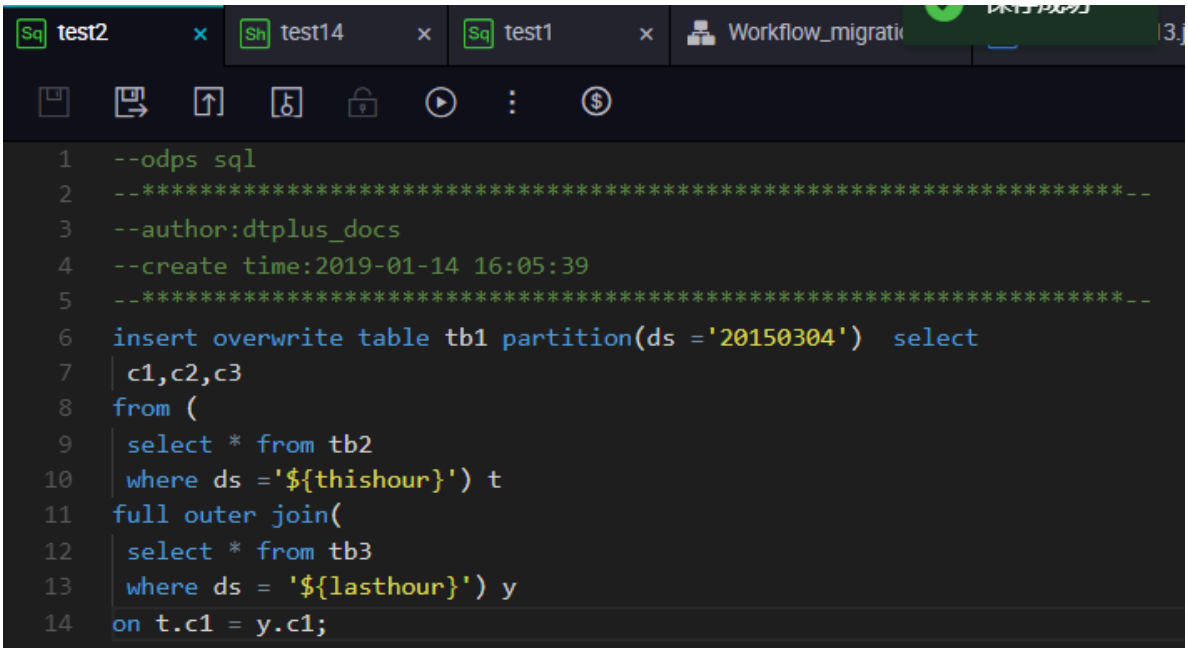
Shell节点自定义参数示例

设置一个Shell类型的任务为按天调度，每天01:00执行一次，若想在代码中使用一个自定义常量参数myname和一个自定义变量参数ct，则可按如下步骤进行操作。

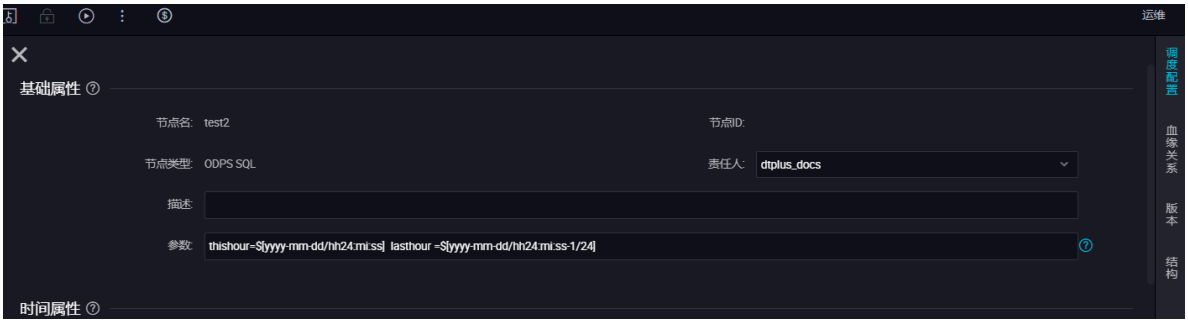
1. 在代码中引用参数，代码如下。

```
echo "hello $1, two days ago is $2, the system param is ${bdp.system
.cyctime}";
```

> **说明:**
> 如果使用系统参数，直接在界面运行可能会报错，建议通过开发环境冒烟测试来测试该任务的运行。



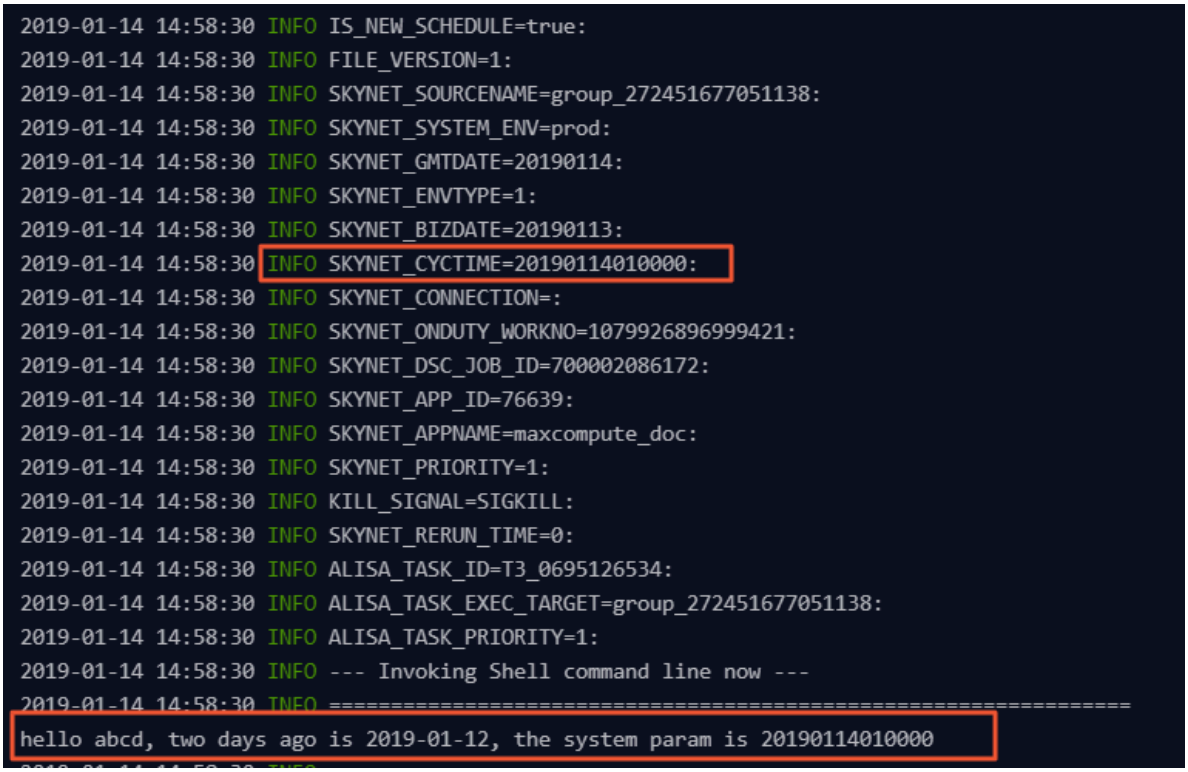
2. 代码中引用的变量，需要调度配置 > 基本属性 > 参数中赋值。赋值规则：参数1 参数2 参数3…（根据参数的位置，解析出来替换的变量，例如$1解析出来的值是参数1）。本示例分别给$1和$2赋值为abcd及$[yyyy-mm-dd-2]。



3. 设置时间周期为每天1点运行。

4. 设置好周期与依赖后，提交任务，您可以在运维中心看到您的任务。从第二天开始，您的任务在运行时会产生周期实例，右键单击查看运行日志，即可看到代码中$1被替换为常

量abcd，$2被替换为2019-01-12，也就是运行日期的前两天， ${bdp.system.cyctime} 被替换为 20190114010000。



自定义参数变量

自定义参数变量分为常量参数和变量（调度内置）参数两种。

- 变量值为常量

  以SQL类型节点为例，代码里${变量名}， 节点配置参数项为变量名=固定值。

  - 代码：`select xxxxxx type='${type}'`
  - 调度变量赋值：`type='aaa'`，调度执行时，代码中会替换为`type='aaa'`。

- 变量值为变量参数

  变量参数即调度内置参数，其取值主要基于系统参数${bdp.system.bizdate}和${bdp.system.cyctime}。

  以SQL类型节点为例，代码中${变量名}，节点配置参数项为变量名=调度参数。

  - 代码：`select xxxxxx dt=${datetime}`
  - 调度变量赋值：`datetime=$bizdate`

    调度执行时，如果今日是2017年07月22日，代码中会替换为`dt=20170721`。

变量参数列表

- $bizdate
  - 参数说明：业务日期（格式为yyyymmdd），日常调度默认为前一天的日期。
  - 示例：ODPS SQL节点代码中pt=${datetime}，节点配置的参数配置为datetime=$bizdate。今日是2017年07月22日，今天节点执行时$bizdate替换的时间即pt=20170721。
- $cyctime
  - 参数说明：任务的定时时间，如果天任务无定时，就是当天0点整（精确到时分秒，一般是小时/分钟级调度任务使用）。

说明:

:

■ $[]和${}配置时间参数区别于$bizdate业务日期，默认为当前时间减一天。

■ $cyctime任务定时调度时间，如果天任务并没有设置定时，就是当天0点整（精确到时分秒，一般是小时/分钟级调度任务使用）。

如果定时在0点30运行，以当天为例，就是yyyy-mm-dd 00:30:00。

■ 如果{}参数，就是以bizdate为基准参与运算，补数据时选择的是什么业务日期，参数替换结果就是什么业务日期。

■ 如果是[]参数，则是以cyctime为基准参与运行，和Oracle的时间运算方式一致。补数据时选中什么业务日期，参数替换结果是业务日期+1天。

例如补数据选择业务日期为20140510，执行时cyctime替换结果是20140511。

- 示例：假设$cyctime=20140515103000。
  - ■ $[yyyy] = 2014, $[yy] = 14, $[mm] = 05, $[dd] = 15, $[yyyy-mm-dd] = 2014-05-15, $[hh24:mi:ss] = 10:30:00, $[yyyy-mm-dd hh24:mi:ss] = 2014-05-1510:30:00
  - ■ $[hh24:mi:ss - 1/24] = 09:30:00
  - ■ $[yyyy-mm-dd hh24:mi:ss -1/24/60] = 2014-05-1510:29:00
  - ■ $[yyyy-mm-dd hh24:mi:ss -1/24] = 2014-05-15 09:30:00
  - ■ $[add_months(yyyymmdd,-1)] = 20140415
  - ■ $[add_months(yyyymmdd,-12*1)] = 20130515
  - ■ $[hh24] =10
  - ■ $[mi] =30
- 测试$cyctime参数的方法如下所示：

  实例运行后，右键单击查看节点属性，定时时间便是该实例的周期定时时间。

  定时时间减1小时的参数执行替换结果。

- $jobid
  - 参数说明：任务所属的工作流ID。
  - 示例：jobid=$jobid。
- $nodeid
  - 参数说明：节点ID。
  - 示例：nodeid=$nodeid。

- $taskid
  - 参数说明：任务ID（节点实例ID）。
  - 示例：taskid=$taskid。
- $bizmonth
  - 参数说明：业务月份（格式为yyyymm），当业务日期的月份等于当前月份时，$bizmonth=业务日期月份-1，否则$bizmonth=业务日期月份。
  - 示例：ODPS SQL节点代码中pt=${datetime}，节点配置的参数配置为datetime=$bizmonth。

    今日是2017年07月22日，今天节点执行时$bizmonth替换的时间即pt=201706。
- ${…}自定义参数
  - 根据$bizdate参数自定义时间格式，其中yyyy表示4位的年份，yy表示2位的年份，mm表示月，dd表示天。可以将参数任意组合，比如${yyyy}、${yyyymm}、${yyyymmdd}、${yyyy-mm-dd}等。
  - $bizdate精确到年月日，因此${……}自定义参数也只能表示到年月日级别。
  - 获取+/-周期的方法，如下表所示：

获取+/-周期	方法
后N年	${yyyy+N}
前N年	${yyyy-N}
后N月	${yyyymm+N}
前N月	${yyyymm-N}
后N周	${yyyymmdd+7*N}
前N周	${yyyymmdd-7*N}
后N天	${yyyymmdd+N}
前N天	${yyyymmdd-N}

- $gmtdate
  - 参数说明：当前日期（格式为yyyymmdd），此参数默认为当天日期，补数据时传入的是业务日期+1。
  - 示例：ODPS SQL节点代码中pt=${datetime}，节点配置的参数配置为datetime=$gmtdate。今日是2017年07月22日，今天节点执行时$gmtdate替换的时间即pt=20170722。

- ${yyyymmdd}
  - 参数说明：业务日期（格式yyyymmdd，值与$bizdate一致），yyyymmdd之间可以支持任意分隔符，例如yyyy-mm-dd。

    日常调度默认为前一天的日期。此参数格式可以自定义格式，如${yyyy-mm-dd}格式为yyyy-mm-dd。
  - 示例：
    - ODPS SQL节点代码中pt=${datetime}，节点配置的参数配置为datetime=${yyyy-mm-dd}。今日是2018年07月22日，今天节点执行时${yyyy-mm-dd}替换的时间即pt=2018-07-21。
    - ODPS SQL节点代码中pt=${datetime}，节点配置的参数配置为datetime=${yyyymmdd-2}。今日是2018年07月22日，今天节点执行时${yyyymmdd-2}替换的时间即pt=20180719。
    - ODPS SQL节点代码中pt=${datetime}，节点配置的参数配置为datetime=${yyyymm-2}。今日是2018年07月22日，今天节点执行时${yyyymm-2}替换的时间即pt=201805。
    - ODPS SQL节点代码中pt=${datetime}，节点配置的参数配置为datetime=${yyyy-2}。今日是2018年07月22日，今天节点执行时${yyyy-2}替换的时间即pt=2016。
    - ODPS SQL节点配置中多个参数赋值如：startdatetime=$bizdate enddatetime=${yyyymmdd+1} starttime=${yyyy-mm-dd} endtime=${yyyy-mm-dd+1}。

参数配置常见问题

- Q：表的分区格式为pt=yyyy-mm-dd hh24:mi:ss，但是调度参数中不允许配置空格，我该如何配置$[yyyy-mm-dd hh24:mi:ss]的格式呢?

  A：使用两个自定义变量参数 datetime=$[yyyy-mm-dd]、 hour=$[hh24:mi:ss]，分别获取日期和时间，然后在代码中拼接为 pt=”${datetime} ${hour}”（注意拼接的两段自定义参数之间需要空格隔开）。
- Q：在代码中表的分区为pt=”${datetime} ${hour}”，希望执行的时侯取上个小时的数据。使用两个自定义变量参数 datetime=$[yyyymmdd]、 hour=$[hh24-1/24]，分别获取日期

和时间可以满足需求。但是0点运行的实例，计算结果会变成当天的23点，实际应当是前一天的23点，怎么办?

A：参数的计算公式稍作修改，datetime修改为$[yyyymmdd-1/24]，hour的计算公式不变仍为$[hh24-1/24]。计算结果如下，即可满足需求。

- 如果一个实例的定时时间是2015-10-27 00:00:00，减1小时就是昨天，所以$[yyyymmdd-1/24]的值是20151026，$[hh24-1/24]的值是23。
- 如果一个实例的定时时间为2015-10-27 01:00:00的实例，减1小时还是今天，所以$[yyyymmdd-1/24]的值是20151027，$[hh24-1/24]的值是00。

DataWorks提供了4种运行方式。

- 数据开发页面运行：需要在参数配置页面临时赋值以保证运行。但该赋值不会保存为任务的属性，因此不会对其他3种运行方式起作用。
- 系统自动周期运行：不需要在参数编辑框做任何配置，调度系统会根据当前实例的定时运行时间自动替换。
- 测试运行/补数据运行：触发时需要指定业务日期，可根据上述计算公式推断定时运行时间，从而得知各实例中这两个系统参数的取值。

3.6.3 时间属性

本文将为您介绍如何配置时间属性，包括调度周期和依赖项。

单击页面右侧的调度属性，进入时间属性模块。

实例生成方式

- T+1次日生成：全量转实例。
  - 23:30之前提交发布的任务，第二天产生实例。
  - 23:30之后提交发布的任务，第三天产生实例。
- 发布后即时生效：详情请参见实时转实例。

节点状态

- 正常调度：会按照调度周期时间配置调度，并正常执行，通常任务默认选中此项。
- 空跑调度：会按照调度周期时间配置调度，但都是空跑执行，即一调度到该任务便直接返回成功，没有真正的执行任务。
- 出错重试：节点出现错误，可以重跑节点。默认出错自动重试3次，时间间隔为2分钟。
- 暂停调度：暂停调度之后，会按照下面的调度周期时间配置调度，但是一调度到该任务会直接返回失败，不会执行。通常用于某个任务暂时不用执行但后面还会继续使用的场景。

调度周期

DataWoks中，当一个任务被成功提交后，底层的调度系统从第二天开始，将会每天按照该任务的时间属性生成实例，并根据上游依赖的实例运行结果和时间点运行。23:30之后提交成功的任务从第三天开始才会生成实例。

 说明:

如果有一个任务需要每周一执行一次，那么只有运行时间是周一的情况下，该任务才会真正执行。运行时间不是周一的情况下，该任务会空跑（直接将任务置为成功），不会实际运行。所以在测试/补数据时，周调度任务需要选择业务日期=运行时间-1。

一个周期运行的任务，其依赖关系的优先级大于时间属性。在时间属性决定的某个时间点到达时，任务实例不会马上运行，而是先检查上游是否全部运行成功。

- 上游依赖的实例没有全部运行成功，并且已到定时运行时间，则实例仍为未运行状态。
- 上游依赖的实例全部运行成功，并且未到定时运行时间，则实例进入等待时间状态。
- 上游依赖的实例全部运行成功，并且已到定时运行时间，则实例进入等待资源状态准备运行。

如果您选择依赖上一周期，配置方法请参见#unique_362。

天调度

天调度任务，即每天自动运行一次。新建周期任务时，默认的时间周期为每天0点运行一次，可根据需要自行指定运行时间点，例如下图指定每天13点运行一次。

- 当勾选定时调度，则每日任务实例定时时间为—当天日期年-月-日 定时时:分:秒。

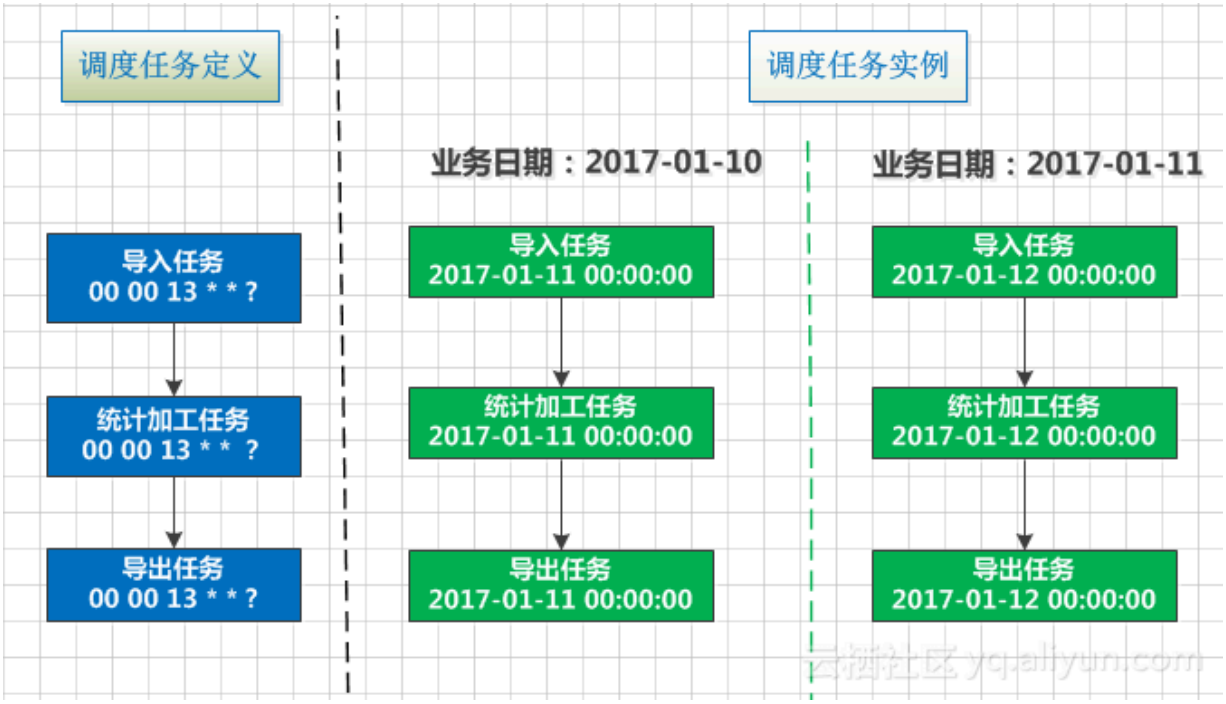
  说明:
  调度任务需满足上游任务执行成功，并且已到定时时间两个条件，任务才能成功执行。任何一个条件没有满足都无法执行，两个条件没有先后顺序。

- 当不勾选定时调度，则每日任务实例定时时间—当天日期年-月-日，默认调度时间，从0点到0点30分随机生成。

应用场景：

导入、统计加工和导出任务，都是天任务，具体时间如上图的13:00 。统计加工任务依赖导入任务，导出任务依赖统计加工任务，依赖配置如下图（统计加工任务的依赖属性配置上游任务为导入任务）所示。

上图的配置，调度系统会自动为任务生成实例并运行。



周调度

周调度任务，即每周的特定几天在特定时间点自动运行一次。当到了没有被指定的日期时，为保证下游实例正常运行，系统也会生成实例但直接设置为运行成功，而不会真正执行任何逻辑，也不会占用资源。

如上图所示，每周一、周五两天生成的实例会正常的调度执行，而周二、三、四、六以及周日5天都是生成实例然后直接设置为运行成功。

上图的配置，调度系统会自动为任务生成实例并运行。

月调度

月调度任务，即每月的特定几天在特定时间点自动运行一次。当到了没有被指定的日期时，为保证下游实例正常运行，系统也会每天生成实例但直接设置为运行成功，而不会真正执行任何逻辑，也不会占用资源。

如上图所示，每月1日生成的实例会正常的调度执行，其他日期每天都是生成实例并直接设为运行成功。

上图的配置，调度系统会自动为任务生成实例并运行。

小时调度

小时调度任务，即每天指定的时间段内，按N*1小时的时间间隔运行一次，例如每天1点到4点的时间段内，每1小时运行一次。

说明：

时间周期按左闭右闭原则计算，例如配置为从0点到3点的时间段内，每隔1个小时运行一次，表明时间区间为[00:00，03:00]，间隔为1小时，调度系统将会每天生成4个实例，分别在0点/1点/2点/3点运行。

如上图的配置，表示每天00点整到23点59分这个时间段内，每隔6小时会自动调度一次，因此调度系统会自动为任务生成实例并运行。

分钟调度

分钟调度任务，即每天指定的时间段内按N*指定分钟的时间间隔运行一次。

如下图所示，每天00:00开始到23:00的时间段内，每隔30分钟调度一次。

目前分钟仅支持最小5分钟的粒度，时间表达式根据上面选择的时间生成，不能手动修改。

说明:

关于实时转实例的详细解释，请参见#unique_361。

常见问题

- **Q：我的上游A是小时任务，下游是日调度，任务每天在A任务全部执行完成之后要汇总执行一次，这样可以相互依赖吗?**

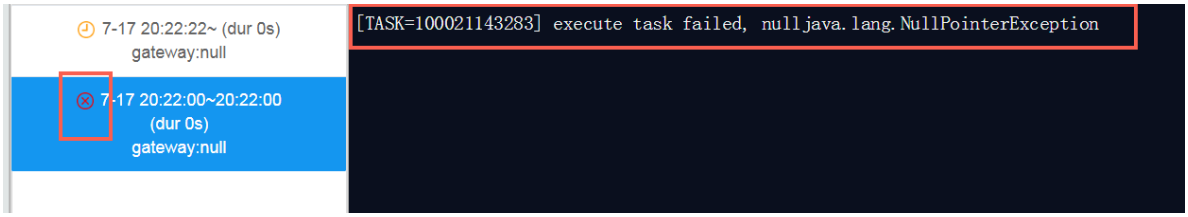
  A：日任务依赖小时任务是可以的，A任务配置成小时调度，任务配置成日调度不定时，配置为上下游依赖（依赖配置请看调度依赖说明），这样每天A任务成功运行24小时的实例后，B任务即可运行。所以每种周期的任务都可以相互依赖，每个任务的调度周期都是任务本身时间属性决定。

- Q：A节点每天每小时整点执行一次，B节点每天跑一次，如何设置A节点每天第一次跑成功后，B节点便开始执行？

  A：配置A节点时，需要勾选依赖上一周期，并选择本节点，B节点的定时时间设为0点，这样每天自动调度实例中，B节点实例便只依赖A节点0点的实例，即A节点的第一个实例。
- Q：A任务每周一跑一次，B任务依赖A任务，也希望跟A任务一样每周1跑一次怎么配置？

  A：B任务的时间属性跟A任务一模一样即可，即调度周期也要选择周调度 > 周一。
- Q：任务被删除，实例是否受影响？

  A：当一个任务运行一段时间后被删除时，由于调度系统每天会按时间属性为该任务生成对应的一个或多个实例，实例不会被删除。因此当这些实例在任务被删除了之后才被触发运行时，会由于找不到需要运行的代码而失败，报错信息如下所示。



- Q：如何在每月的最后一天计算当月数据？

  A：目前系统不支持配置每月最后一天，因此如果时间周期选择每月31日，那么在有31日的月份会有一天调度，其他日期都是生成实例然后直接设为运行成功。

  需要统计每个月的数据时，建议选择每月的1日运行，计算上个月的数据。
- Q：如何让一个依赖小时节点的天节点，到定时时间0点时自动运行？

  A：天节点依赖小时节点，不需要依赖今天的数据，只需要依赖昨天的小时数据（直接依赖今天的小时节点实例，会导致下游天节点实例到第二天才完成）。

  在天节点的调度配置界面，选择依赖上一周期 > 自定义，填入上游小时节点的节点ID，并重新提交发布。
- Q：无法确定上游何时产出数据时，该怎么办？

  A：无法确定上游节点何时产出数据时，本节点可对上游做跨周期依赖。
- Q：修改后的节点任务提交发布到生产环境后，是否会覆盖掉之前生产环境的错误节点？

  A：不会覆盖之前的节点，未运行的实例会用最新代码运行，不会删除已生成的节点实例。如果调度参数有变化，需要重新生成实例去运行。

3.6.4 依赖关系

调度依赖关系是您构建有序业务流程的根本，只有正确构建任务依赖关系，才能保障业务数据有效、适时地产出，形成规范化的数据研发场景。

在DataWorks使用上，通过代码自动解析+设置节点依赖关系配置节点依赖，通过上下游关系正常及节点运行状态来保障业务数据的顺序产出。

设置节点依赖关系的目的：检测SQL所查询的表的数据的产出时间，通过节点的状态成功默认上游数据数据顺利产出。

您可将上游节点的本节点输出作为下游节点的本节点输入，以形成依赖关系。

DataWorks V2.0提供自动推荐、自动解析和自定义配置三种依赖配置模式。依赖关系实际操作示例请参考#unique_367。

无论如何配置依赖关系，调度的总逻辑不变：上游执行成功之后才会触发下游调度。因此，所有工作流节点都必须至少有一个父节点，调度依赖的核心就是设置这个父子依赖关系。下文将为您详细介绍调度依赖的原理及配置方式。

> **说明:**
>
> - 2019年1月10号之前创建的项目，存在数据问题，需要提交工单进行修正申请。1月10号之后创建的项目，则不受影响。
> - 您可以观看DataWorks V2.0常见问题与难点分析学习依赖关系配置与排错。

规范化数据开发场景

- 在进行调度依赖关系配置前，您需要了解以下基本概念：
  - DataWorks任务：定义对数据执行的操作，详情请参见#unique_368。
  - 输出名称：系统将为每个节点默认分配一个以.out结尾的输出名，同时您也可增加自定义输出名，但需注意输出节点名称在租户内不允许重复。详情请参见#unique_368。
  - 产出表：指某节点的SQL语句中，INSERT/CREATE语句之后的表。
  - 输入表：指某节点的SQL语句中，FROM后的表。
  - SQL语句：此处指MaxCompute SQL。

  实际工作中，一个DataWorks任务中可以包含单个SQL语句，也可以包含多个SQL语句。

  每个形成上下游关系的任务均通过输出名进行关联，其中创建的最上游节点的上游节点可配置为本项目的根节点（节点名projectname_root）。

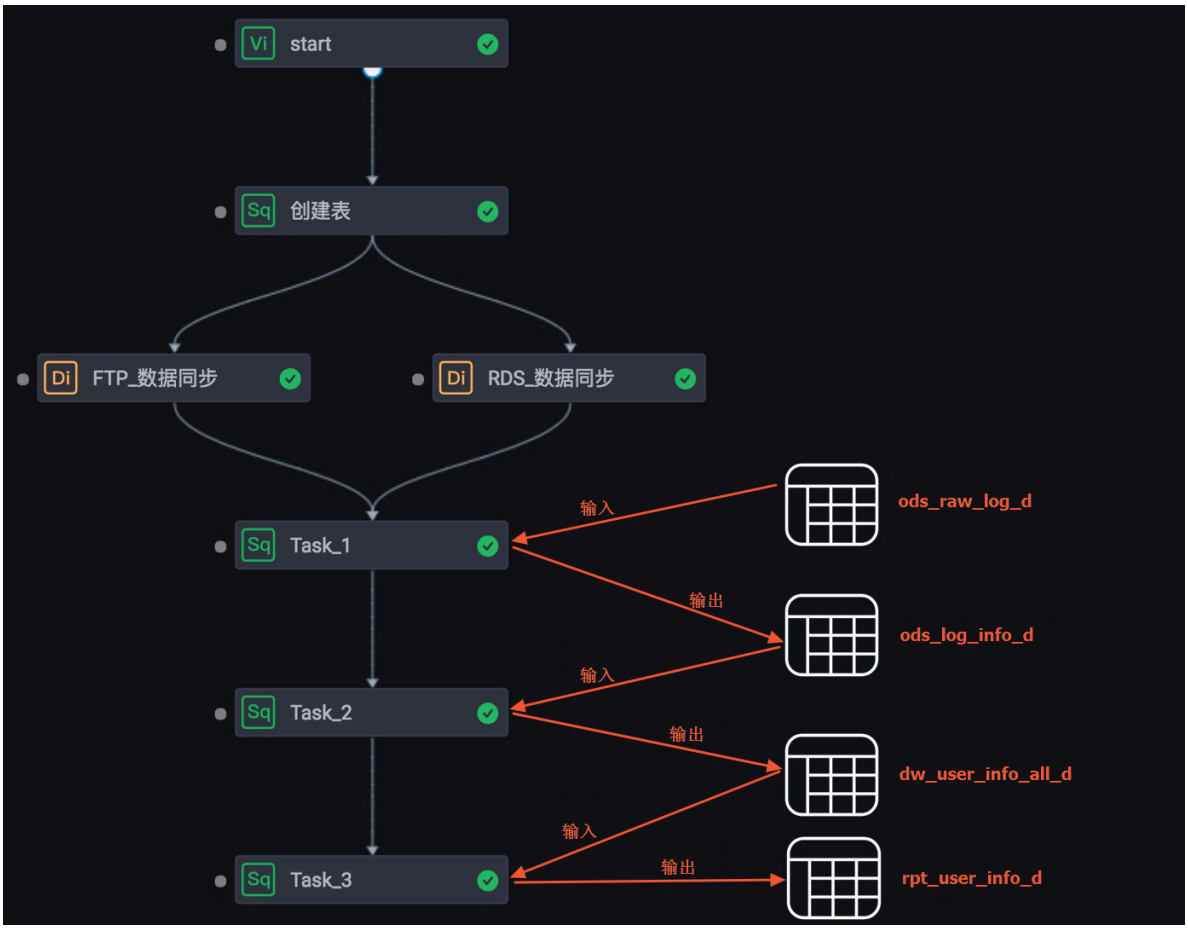
- 规范化数据开发原则

  在规范化的数据开发流程中，会构建多个SQL任务形成具有依赖关系的上下游，同时建议遵循以下原则：

  - 下游任务的输入表必须是上游任务的产出表。
  - 同一张表仅由一个任务产出。
  - 一个任务只产出一张表。

  目的是为了当业务流程无限膨胀时，可快速地通过自动解析方式配置复杂的依赖关系。

- 规范化数据开发流程示例



  上图中，各任务及其代码如下：

  - Task_1任务代码如下，本任务的输入数据来自ods_raw_log_d表，数据输出至ods_log_info_d表。

```
INSERT OVERWRITE TABLE ods_log_info_d PARTITION (dt=${bdp.system.
bizdate})
  SELECT ……  //代表您的select操作
  FROM (
  SELECT ……  //代表您的select操作
  FROM ods_raw_log_d
  WHERE dt = ${bdp.system.bizdate}
```

```
) a;
```

- Task_2任务代码如下，本任务的输入数据来自ods_user_info_d、ods_log_info_d表，数据输出至dw_user_info_all_d表。

```
INSERT OVERWRITE TABLE dw_user_info_all_d PARTITION (dt='${bdp.
system.bizdate}')
SELECT ……  //代表您的select操作
FROM (
  SELECT *
  FROM ods_log_info_d
  WHERE dt = ${bdp.system.bizdate}
) a
LEFT OUTER JOIN (
  SELECT *
  FROM ods_user_info_d
  WHERE dt = ${bdp.system.bizdate}
) b
ON a.uid = b.uid;
```

- Task_3任务代码如下，本任务输入数据来自dw_user_info_all_d表，数据输出至rpt_user_info_d表。

```
INSERT OVERWRITE TABLE rpt_user_info_d PARTITION (dt='${bdp.system
.bizdate}')
SELECT ……  //代表您的select操作
FROM dw_user_info_all_d
WHERE dt = ${bdp.system.bizdate}
GROUP BY uid;
```

依赖的上游节点

依赖的上游节点指当前节点依赖的父节点，此处需填写上游节点的输出名称（一个节点可同时存在多个输出名称，视情况填写您需要的输出即可），而非上游节点名。您可手动搜索上游输出名进行添加，也可通过SQL血缘关系代码解析得到。

说明：

依赖的上游节点，必须使用上游节点的输出名或输出表名进行检索。

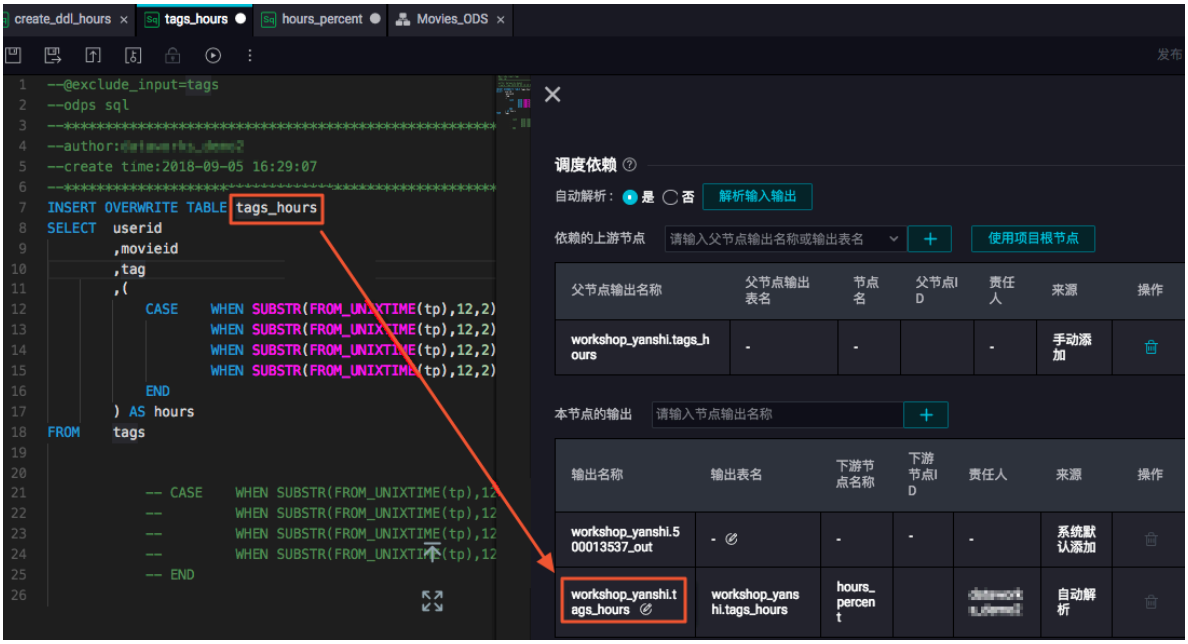
如果您通过手动搜索上游输出名添加，则搜索器会根据已提交至调度系统中的节点的输出名来进行搜索。

- 通过输入父节点输出名搜索

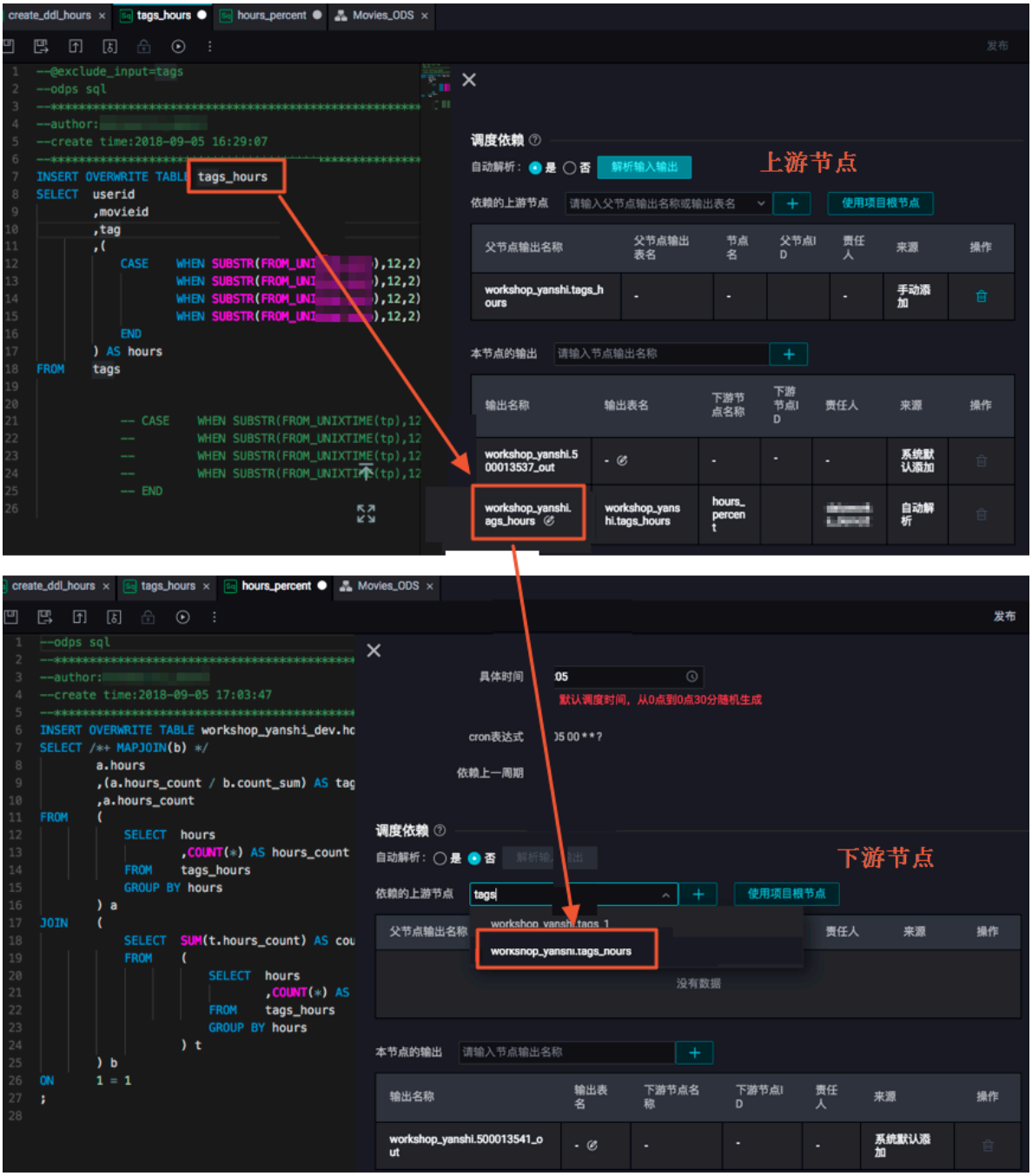
  您可以通过搜索某节点的输出名，将其配置为本节点的上游依赖来形成依赖关系。

- 通过输入父节点输出名的表名称进行搜索

  通过该方法搜索必须保证父节点的某一个输出名，为本节点SQL语句中INSERT或CREATE之后的表名称，形如projectname.表名（此类输出名一般可通过自动解析获得）。



  执行提交后，该输出名即可通过搜索表名的方式被其他节点搜索到。



本节点的输出

本节点的输出指当前节点的输出，您可在右侧的调度配置页面截取本节点的输出信息。

系统将为每个节点默认分配一个以.out结尾的输出名，同时您也可增加自定义输出名或通过自动解析获得输出名。

说明:

输出节点名称是全局唯一的，在整个阿里云账号内不允许重复。

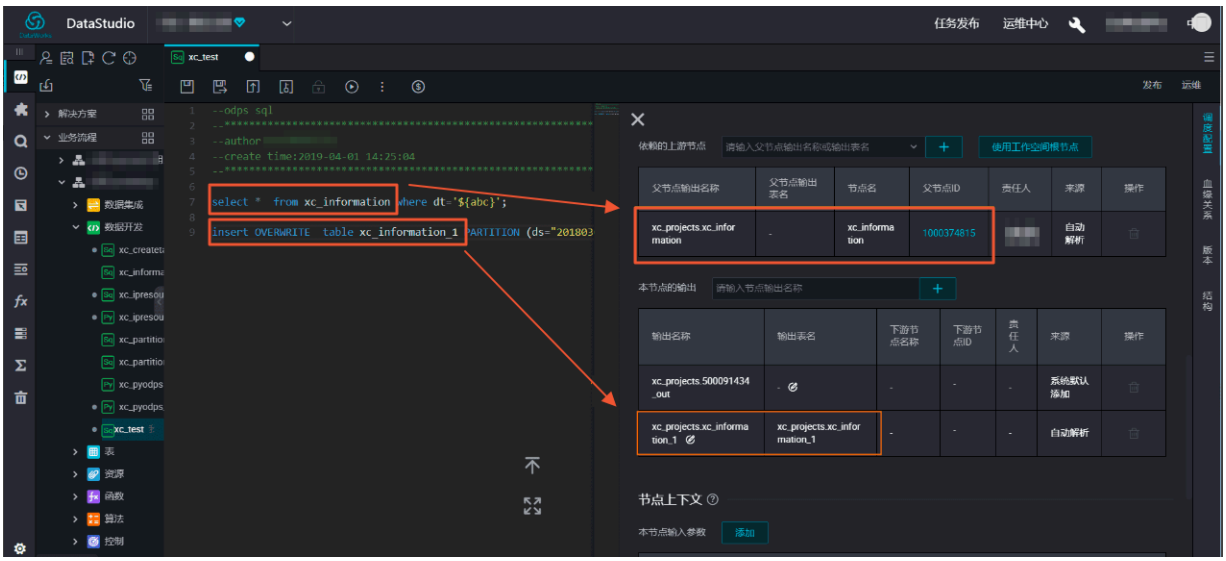
自动解析依赖关系

DataWorks将根据任务节点中实际的SQL内容解析出不同的依赖关系，解析得到的父节点输出名称、本节点输出名称分别为：

- 父节点输出名称：projectname.INSERT后的表名。
- 本节点输出名称：
  - projectname.INSERT后的表名。
  - projectname.CREATE后的表名（一般用于临时表）。

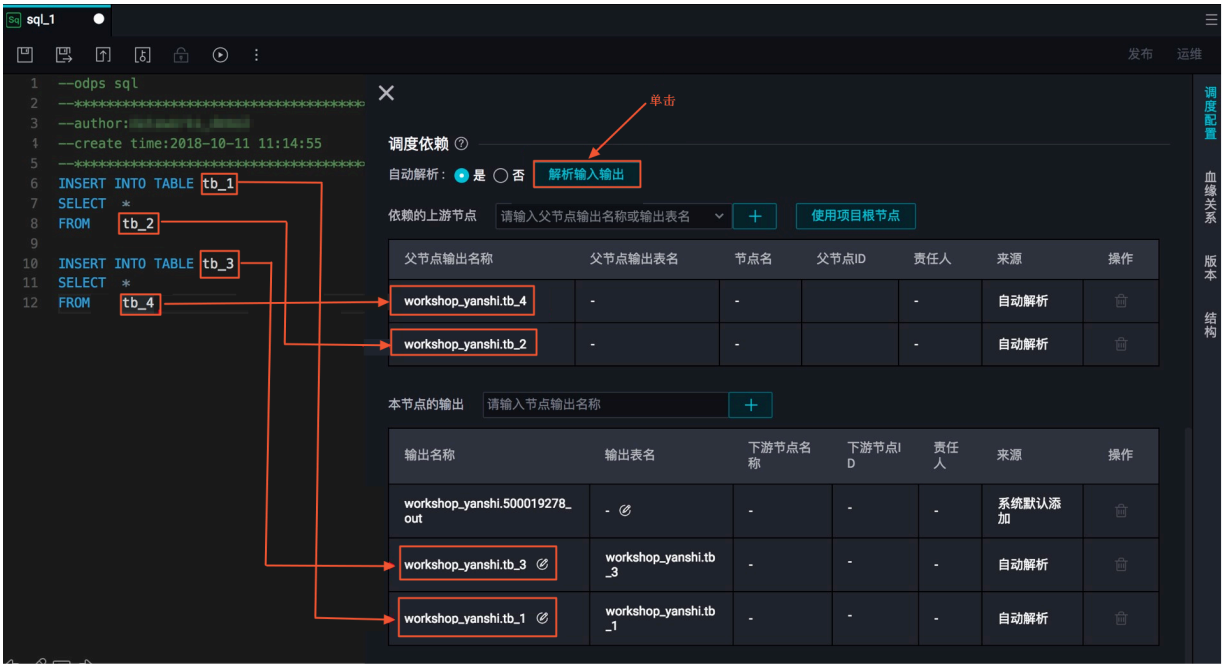
> **说明：**
> 如果您是从DataWorksV1.0升级至DataWorks V2.0的用户，则本节点输出名称为projectname.本节点名。

自定解析依赖关系的原理，如下图所示。

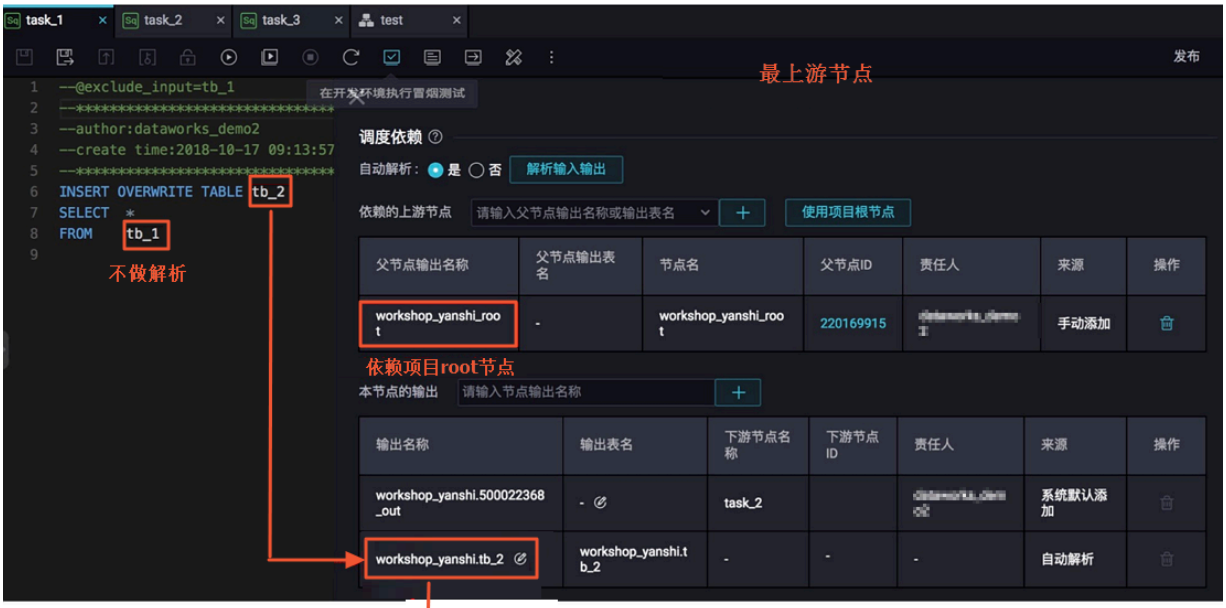


- Select一张表，该表将自动解析为本节点依赖的上游。
- Insert一张表，该表将自动解析为本节点的输出。

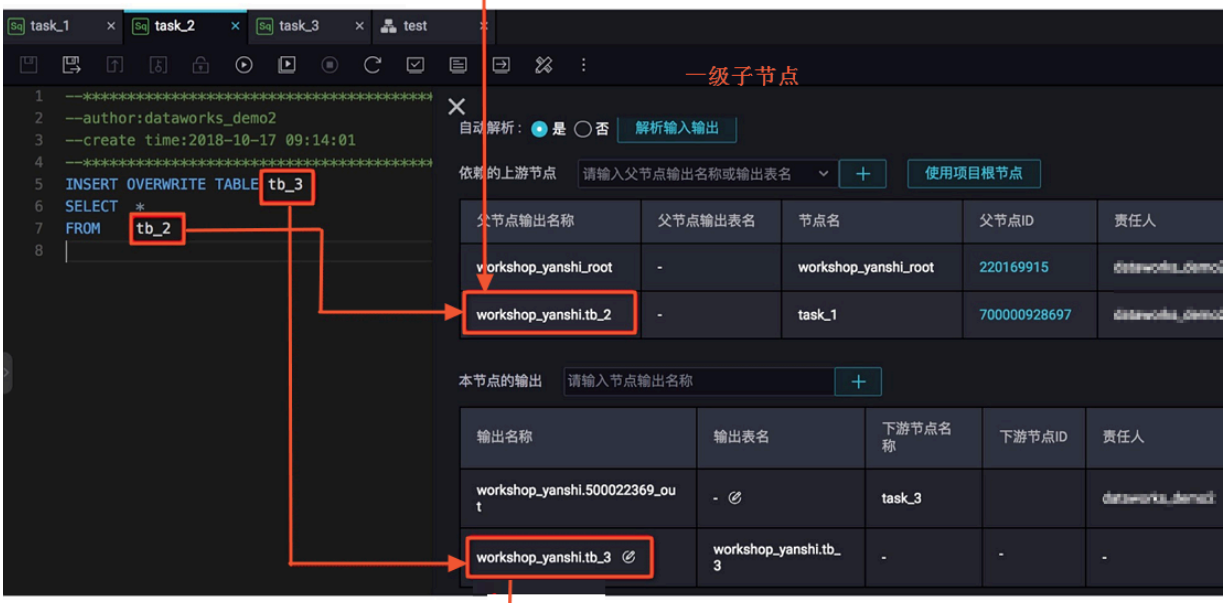
如果出现的多个INSERT、FROM，则会自动解析出多个输出、输入名称。



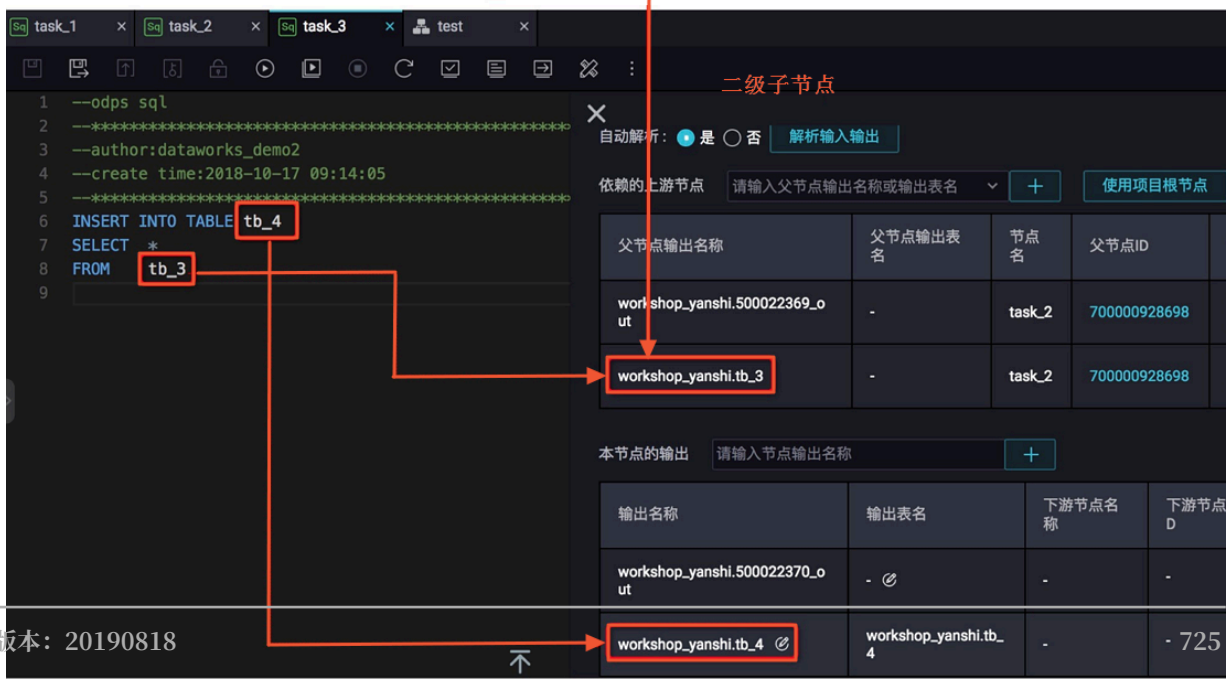
如果您构建了存在依赖关系的多个任务且满足条件：下游任务的输入表均来自上游任务的输出表，则通过自动解析功能即可实现全工作流依赖关系的快速配置。


task_1
task_2
task_3
test
发布
最上游节点
--@exclude_input=tb_1
--author:dataworks_demo2
--create time:2018-10-17 09:13:57
INSERT OVERWRITE TABLE tb_2
SELECT *
FROM tb_1
不做解析
调度依赖
自动解析：是 否
解析输入输出
依赖的上游节点
请输入父节点输出名称或输出表名
使用项目根节点
父节点输出名称
父节点输出表名
节点名
父节点ID
责任人
来源
操作
workshop_yanshi_root
220169915
手动添加
依赖项目root节点
本节点的输出
请输入节点输出名称
输出名称
输出表名
下游节点名称
下游节点ID
workshop_yanshi.500022368_out
task_2
系统默认添加
workshop_yanshi.tb_2
workshop_yanshi.tb_2
自动解析

建立依赖


task_1
task_2
task_3
test
一级子节点
--author:dataworks_demo2
--create time:2018-10-17 09:14:01
INSERT OVERWRITE TABLE tb_3
SELECT *
FROM tb_2
自动解析：是 否
解析输入输出
依赖的上游节点
请输入父节点输出名称或输出表名
使用项目根节点
父节点输出名称
父节点输出表名
节点名
父节点ID
责任人
workshop_yanshi_root
220169915
workshop_yanshi.tb_2
task_1
700000928697
本节点的输出
请输入节点输出名称
输出名称
输出表名
下游节点名称
下游节点ID
责任人
workshop_yanshi.500022369_out
task_3
workshop_yanshi.tb_3
workshop_yanshi.tb_3

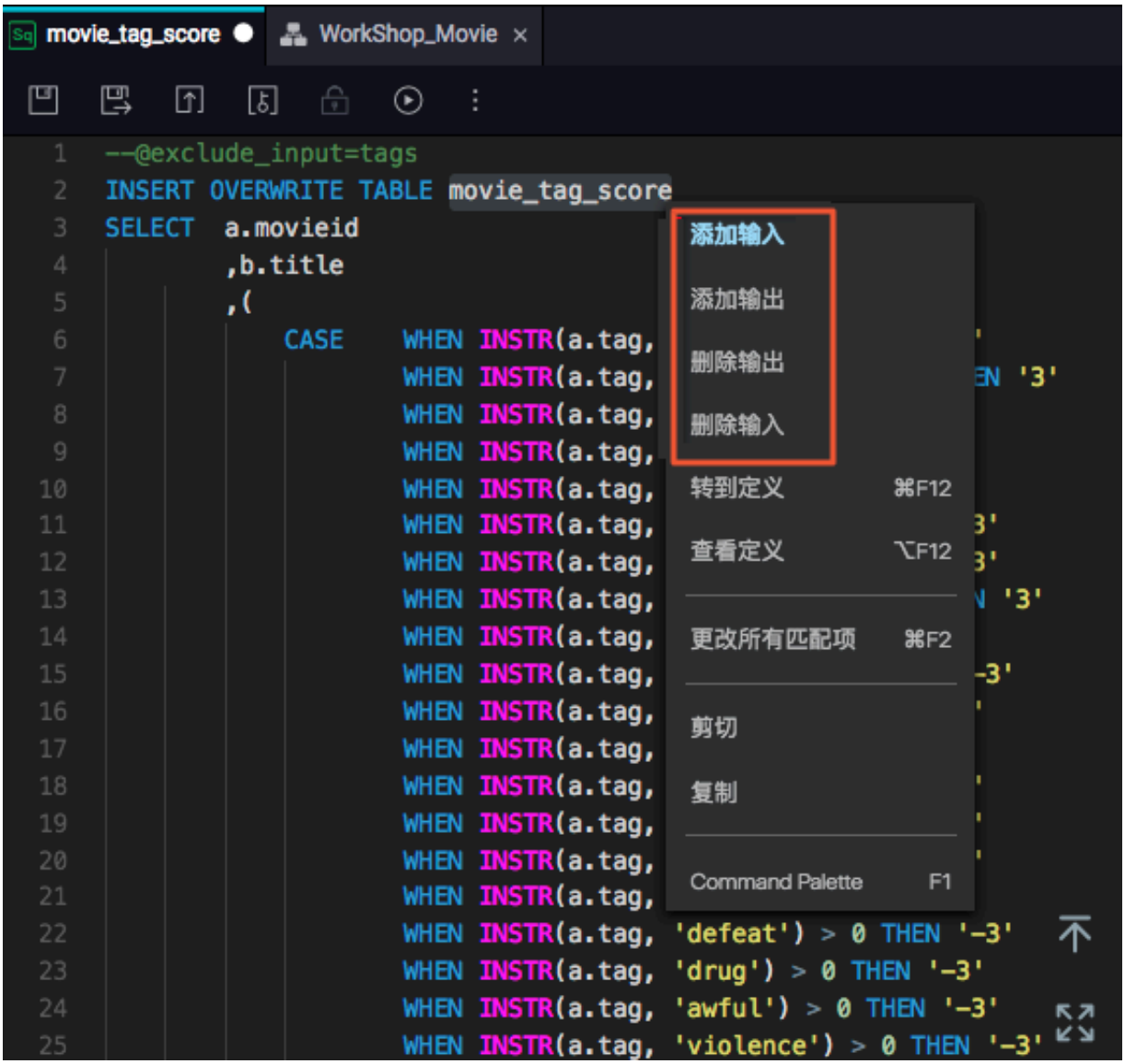
建立依赖


task_1
task_2
task_3
test
二级子节点
--odps sql
--author:dataworks_demo2
--create time:2018-10-17 09:14:05
INSERT INTO TABLE tb_4
SELECT *
FROM tb_3
自动解析：是 否
解析输入输出
依赖的上游节点
请输入父节点输出名称或输出表名
使用项目根节点
父节点输出名称
父节点输出表名
节点名
父节点ID
workshop_yanshi.500022369_out
task_2
700000928698
workshop_yanshi.tb_3
task_2
700000928698
本节点的输出
请输入节点输出名称
输出名称
输出表名
下游节点名称
workshop_yanshi.500022370_out
workshop_yanshi.tb_4
workshop_yanshi.tb_4

说明:

- 为提高任务的灵活性，建议一个任务仅包含一个输出点，以便您可灵活组装SQL业务流程，达到解耦的目的。
- 如果SQL语句中的一个表名既是产出表又是被引用表（被依赖表），则解析时只解析为产出表。
- 如果SQL语句中的一个表名被多次引用或被多次产出，则解析时只解析一个调度依赖关系。
- 如果SQL代码中有临时表（例如在属性配置中指定t_开头的表为临时表），则这个表不会被解析为调度依赖。

在自动解析的前提下，您可通过手工设置添加/删除、输入/输出的方式来决定避免/增加某些SQL语句中的字符被自动解析为输出名/输入名。



选中表名后右键单击，即可对SQL语句中出现的所有表名进行输出、输入的添加或删除。操作后，被添加输入的字符会被解析为父节点输出名称，被添加输出的字符则会被解析为本节点的输出。反之，如果选择删除输入或删除输出则不会被解析。

说明:

除了右键选中SQL语句中的字符，您还可以通过添加注释的方式修改依赖，具体注释代码如下：

```
--@extra_input=表名 --添加输入
--@extra_output=表名 --添加输出
--@exclude_input=表名 --删除输入
--@exclude_output=表名 --删除输出
```

自定义添加依赖关系

当通过SQL血缘关系无法准确自动解析节点之间的依赖关系时，您可选择下图中的否来自行配置依赖关系。

当自动解析选择为否时，您可以单击自动推荐，启用自动推荐上游依赖功能。系统将会基于本项目SQL血缘关系为您推荐产出当前节点输入表的其他所有SQL节点任务，您可根据自身需求，单选或多选推荐列表中的任务，配置为当前节点的上游依赖任务。

说明:

被推荐节点需在前一日提交到调度系统，等到第二日数据产出之后方可被自动推荐功能识别到。

常见场景：

- 本任务输入表≠上游任务产出表。
- 本任务产出表≠下游任务输入表。

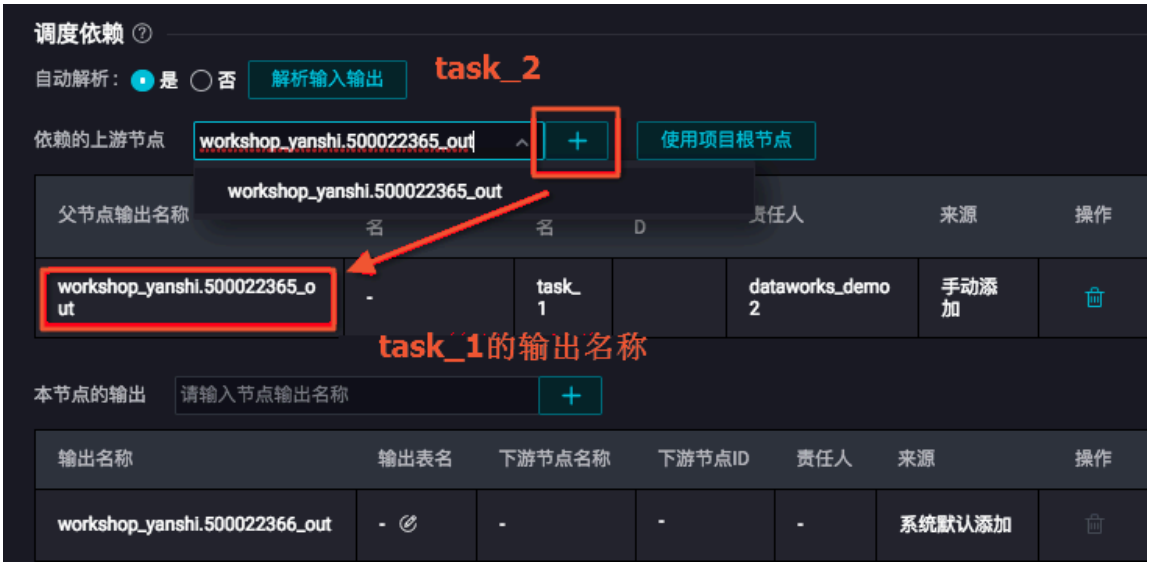
在自定义方式下，您可通过以下两种方式配置依赖。

· 手动添加依赖的上游节点示例

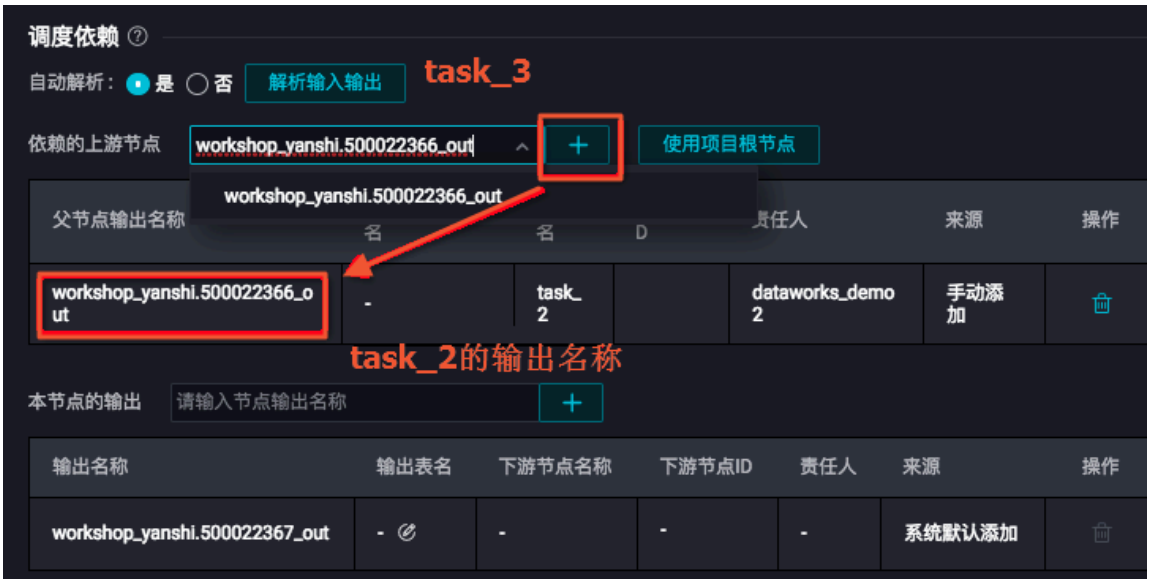
1. 新建三个节点，系统会默认为它们分别配置一个输出名称。

2. 将最上游节点task_1配置依赖本项目root根节点，单击保存。

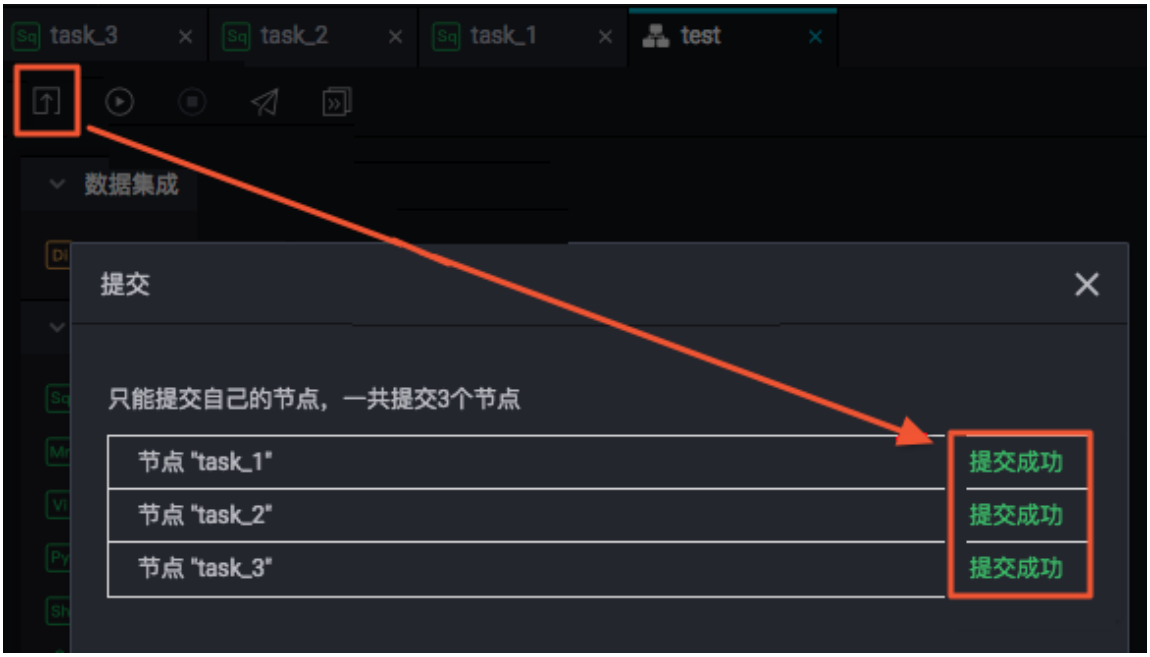
3. 配置task_2依赖task_1的输出名称，单击保存。



4. 配置task_3依赖task_2的输出名称，单击保存。



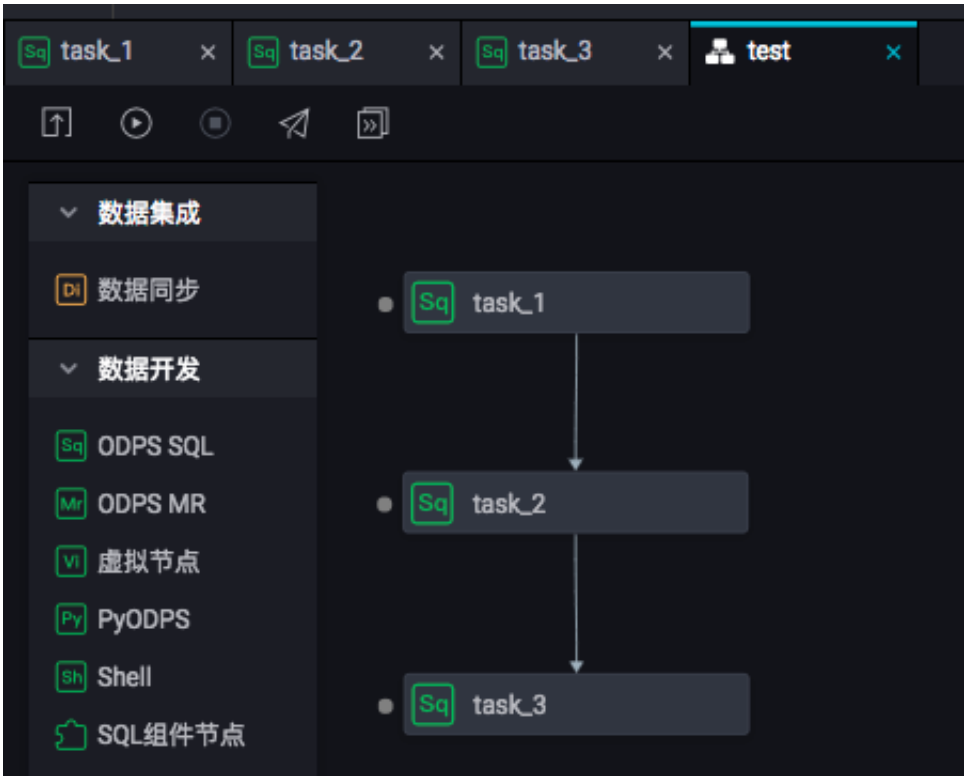
5. 配置完成后，单击提交，判断依赖关系是否正确。如果提交成功则说明依赖配置无误。


task_3
task_2
task_1
test
数据集成
提交
只能提交自己的节点，一共提交3个节点
节点 "task_1"
节点 "task_2"
节点 "task_3"
提交成功
提交成功
提交成功

- 通过拖拽形成依赖关系示例

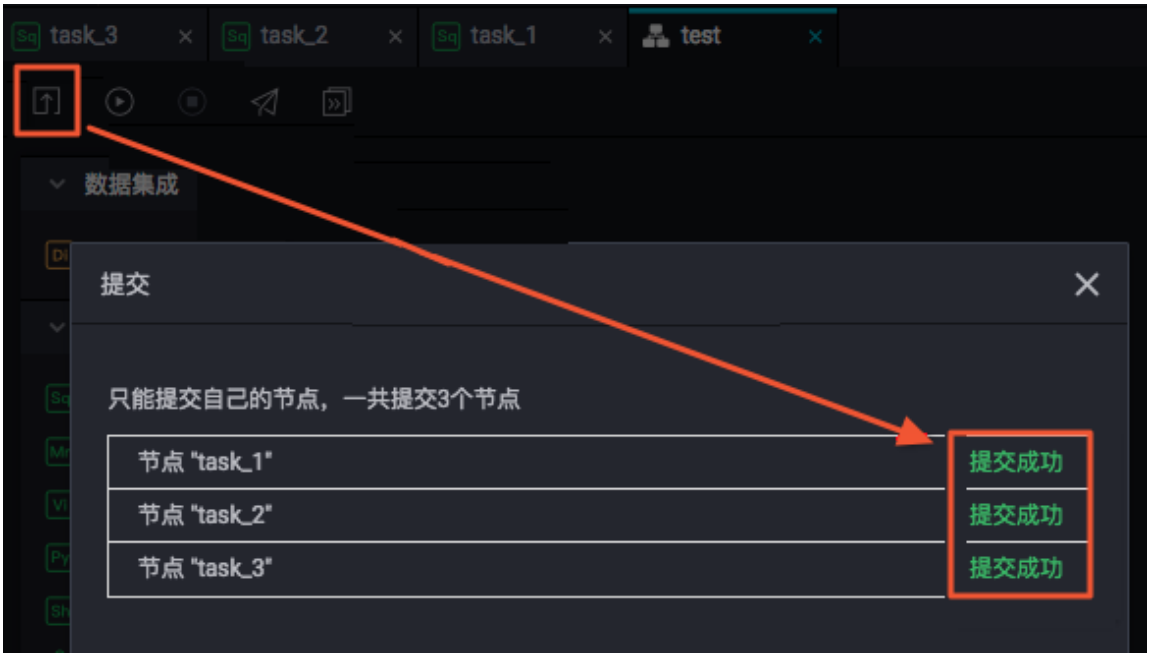
1. 新建三个task节点，将最上游task_1配置依赖上游为根节点，单击保存。

2. 通过拖拽方式将三个task连接起来。



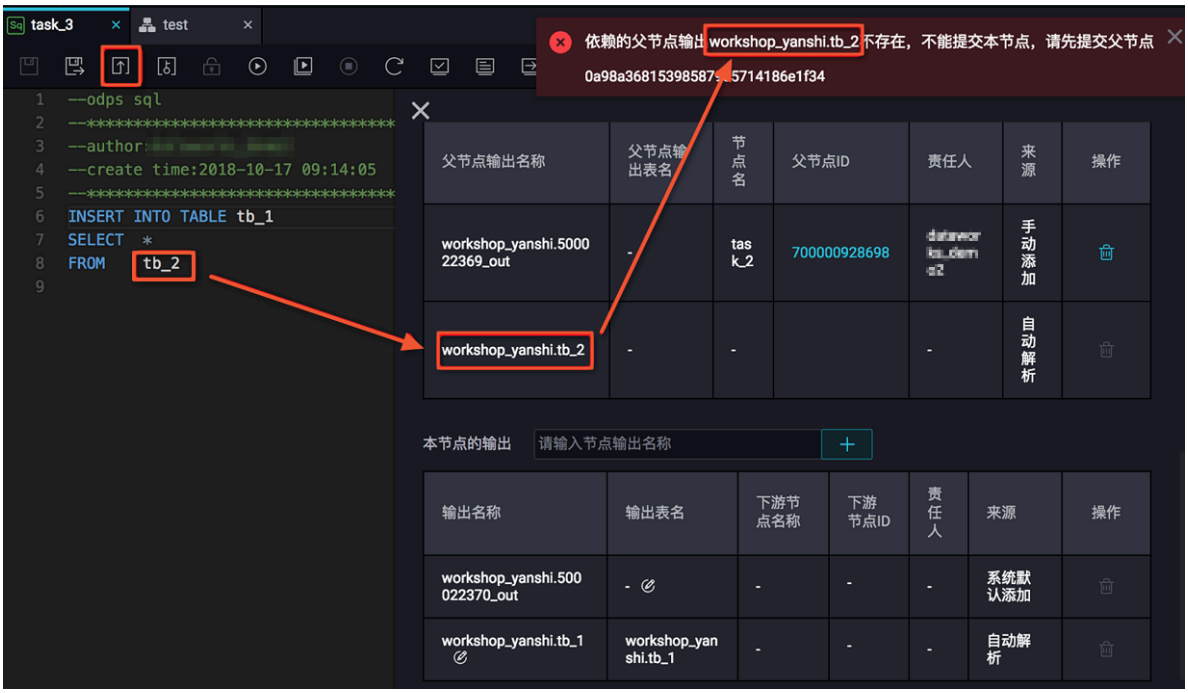
3. 查看task_2、task_3的依赖配置，可看到已自动生成依赖的父节点输出名。

4. 配置完成后，单击提交，判断依赖关系是否正确。如果提交成功则说明依赖配置无误。



常见问题

- Q：自动解析后提交失败，报错依赖的父节点输出projectname.table不存在，不能提交本节点，请先提交父节点。



A：出现上述情况有以下两种原因。

- 上游节点未提交，提交后可再次尝试。
- 上游节点已经提交，但上游节点的输出名不是workshop_yanshi.tb_2。

说明：

通常通过自动解析得到的父节点输出名、本节点输出名会根据INSERT/CREATE/FROM后的表名来得到，请确保配置方式与自动解析依赖关系所介绍的方式一致。

- Q：本节点的输出中，下游节点名称、下游节点ID都是空且不能填写内容？

  A：如果本节点下游无子节点，则无内容。待本节点下游配置子节点后，便会自动解析出内容。

- Q：节点的输出名称用来做什么？

  A：节点的输出名称用于建立节点间的依赖关系。假设A节点的输出名称是ABC，而B节点将ABC作为它的输入，这样节点A与节点B之间便建立了上下游关系。

- Q：一个节点可以有多个输出名称吗？

  A：可以。下游节点引用本节点的任何一个输出名称作为下游节点的父节点输出名称，都将与本节点建立依赖关系。

- Q：多个节点可以有相同的输出名称吗？

  A：不可以。每个节点的输出名称必须在阿里云账号中是唯一的，如果需要多个节点产出数据至同一张MaxCompute表，那么这些节点的输出建议用表名_分区标识。

- Q：使用自动解析依赖关系时，如何不解析到中间表？

  A：在SQL代码中选中中间表名并右键单击删除输入或删除输出，再次执行自动解析输入输出即可。

- Q：最上游任务应如何配置依赖关系？

  A：一般情况下可选择依赖在本项目根节点上。

- Q：为什么在A节点搜索上游节点输出名时，搜索到了B节点不存在的输出名？

  A：因为搜索功能是基于已经提交的节点信息来进行搜索，如果B节点提交成功后，您又删除了B节点的输出名称且未提交至调度系统，则在A节点上仍然能搜到B节点已删除的输出名。

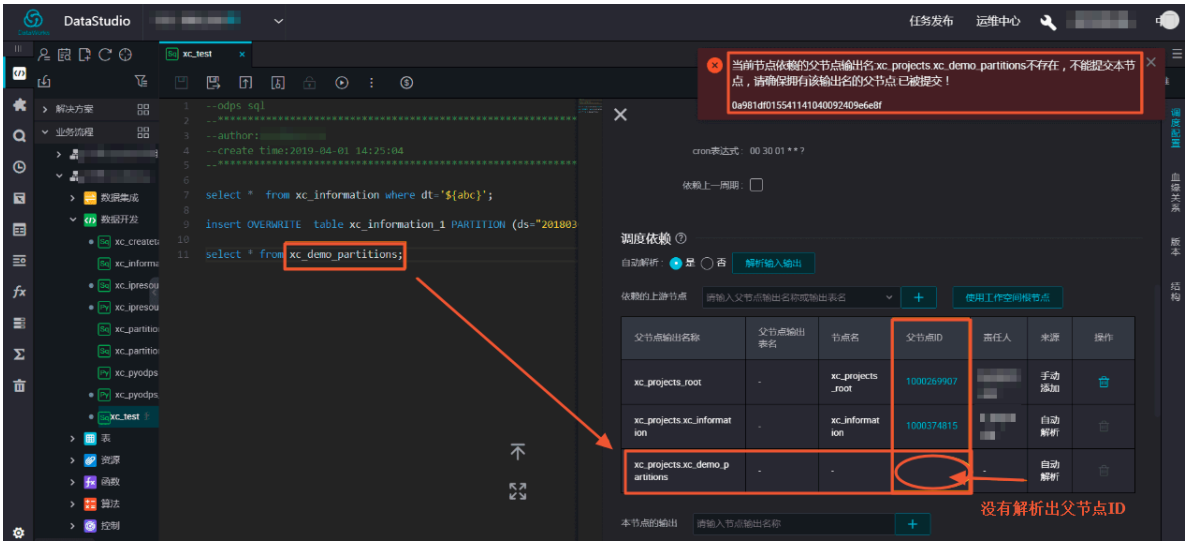
- Q：有A、B、C三个任务，如何实现每个小时执行一次A->B->C（A执行完了B再执行，B执行完了C再执行）的任务流程？

  A：将A、B、C的依赖关系设置为A的输出为B的输入，B的输出为C的输入，同时设置A、B、C的调度周期都为小时即可。

- Q：依赖的上游没有解析到父节点ID，提交报错。

  A：该报错并不是指该表不存在，只是在说明该表不是某个节点的本节点输出，无法通过此表去找到产出这个表数据的节点，从而与这个节点挂上依赖。

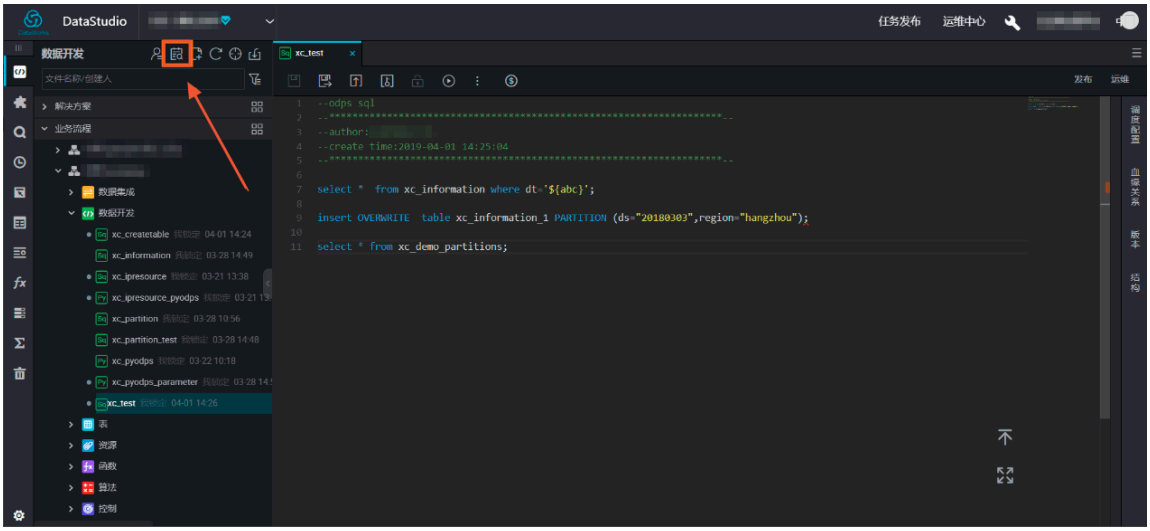
  通过上游节点的本节点输出作为下游节点的本节点输入，根据上文自动解析的原理可知，在SQL中查询xc_demo_partition表，但自动解析时没有通过此表找到上游节点，说明没有一个节点将这个表xc_demo_partition作为本节点输出。



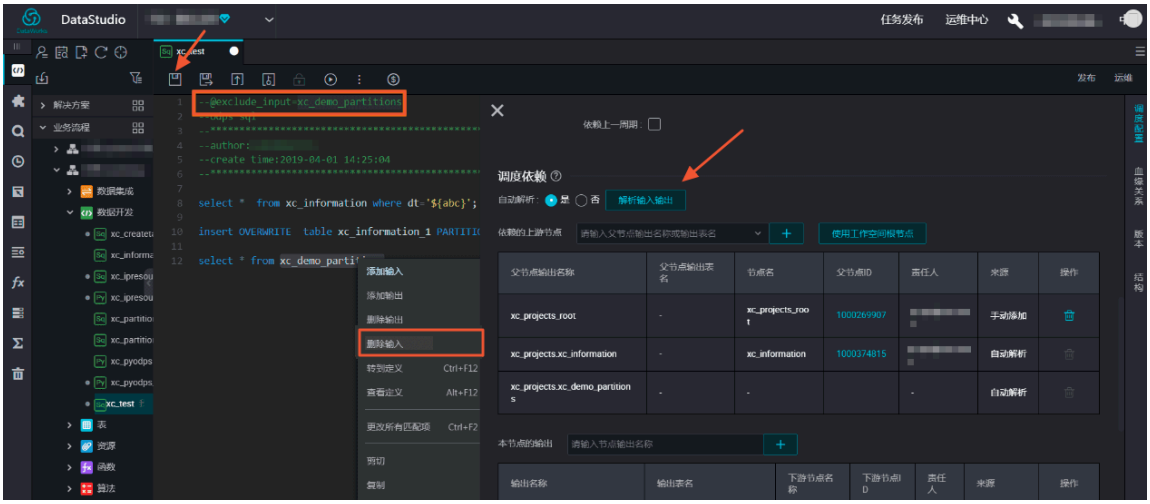
您可以通过下述方法解决此问题。

1. 找到产出该表的节点任务，查看该节点任务的本节点输出。

   如果不知道哪个节点中有操作该表，可以使用代码搜索功能，通过关键字进行模糊查找。



2. 如果是本地上传的表数据，或者不需要依赖该节点，您可以选择在代码区右键，选择删除输入。



说明：

为保证代码血缘的准确性，建议减少使用自定义依赖的次数。

3.6.5 依赖上一周期

依赖上一周期是指依赖某个父节点的上一周期实例，即跨周期依赖。

DataWoks支持以下三种跨周期依赖形式：

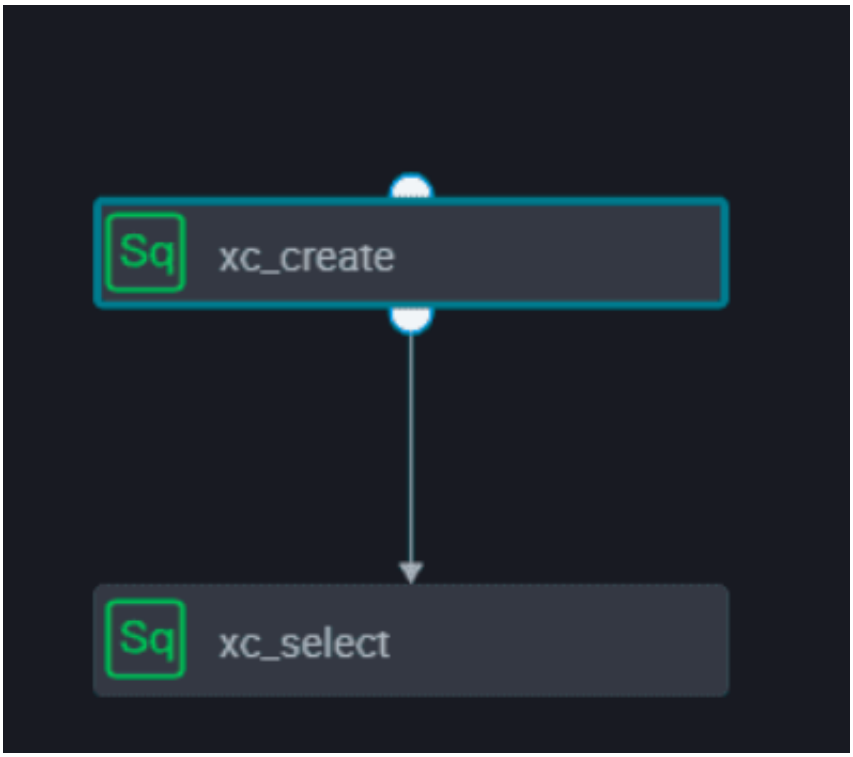
- 一层子节点
  - 节点依赖关系：依赖当前节点的下游。例如节点A存在B、C、D三个下游节点，依赖一层子节点是指节点A依赖B、C、D三个节点的上一周期。
  - 业务场景：本次节点运行依赖上一周期的下游节点，对本节点的结果表（即本节点输出表）进行清洗，查看是否正常产出最终结果。
- 本节点
  - 节点依赖关系：跨周期自依赖（依赖当前节点的上一周期）。
  - 业务场景：本次节点运行依赖上一周期该节点业务数据的产出情况。
- 自定义：手动输入需要依赖的其他节点，此处需要输入节点ID。如果存在多个节点，需用逗号（,）分隔，例如12345,23456。
  - 节点依赖关系：手动输入需要依赖的节点，如果存在多个节点，需用英文逗号（,）进行分隔。
  - 业务场景：业务逻辑上需要依赖其他业务的数据正常产出，但本节点中没有操作该业务数据。

说明：

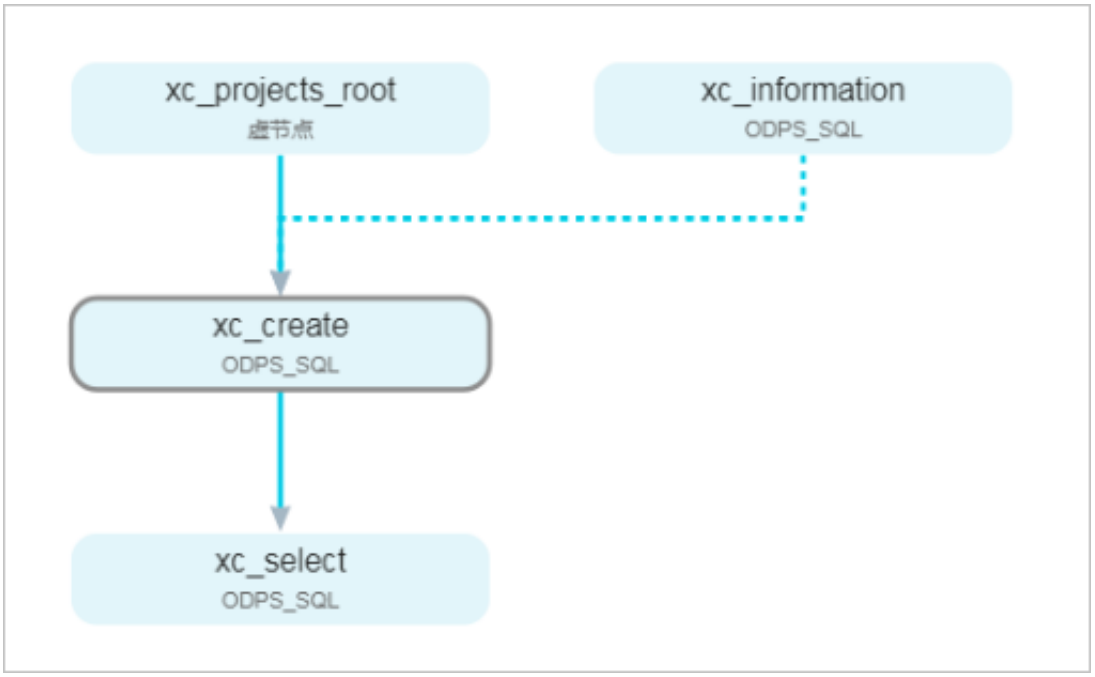
依赖上一周期和依赖本周期的区别：在运维中心中查看节点依赖关系时，所有跨周期依赖的节点都会以虚线的形式展示。

下线节点时，需要删除节点的依赖关系，方可正常运行节点。

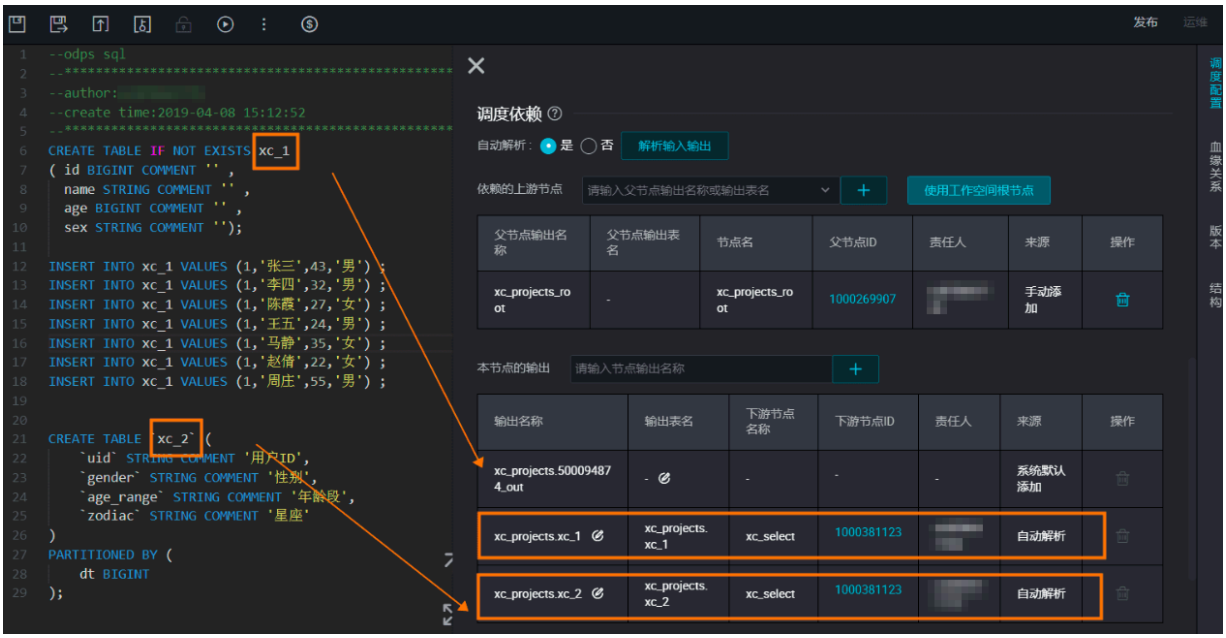
下图为业务流程节点的依赖关系。



运维中心页面为您展示业务流程的依赖关系。

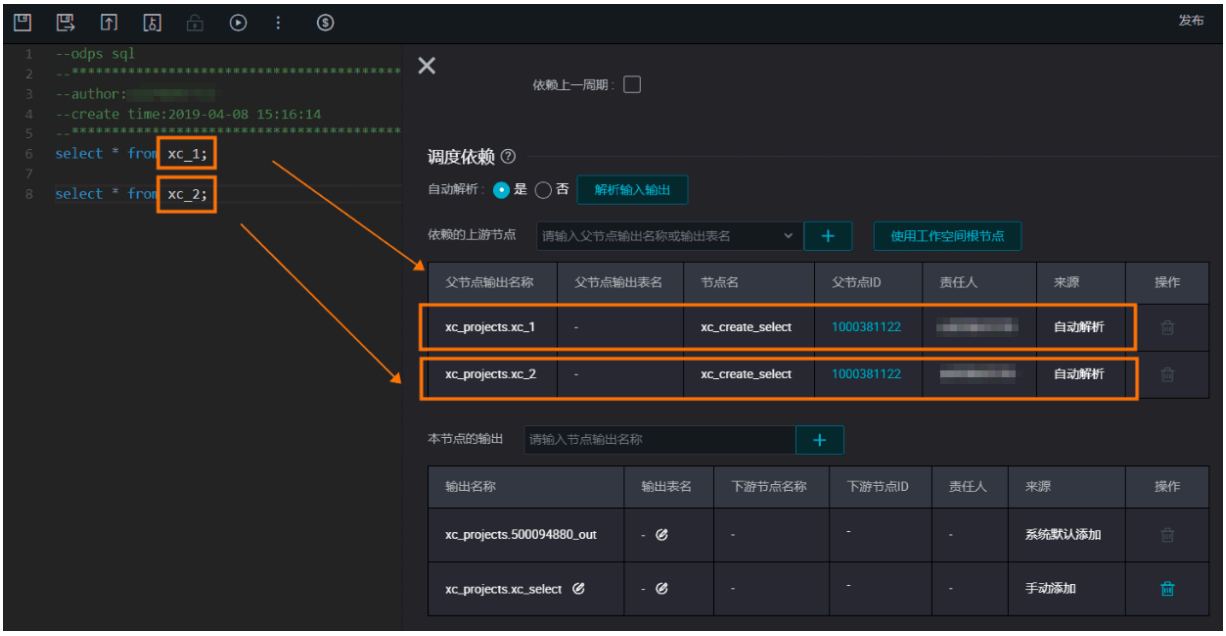


以配置xc_create节点代码为例。



如上图中的SQL节点内容所示，xc_create节点创建xc_1、xc_2两张表（或产出两张表的数据），并将xc_1、xc_2作为本节点的输出。

以配置xc_select节点代码为例。



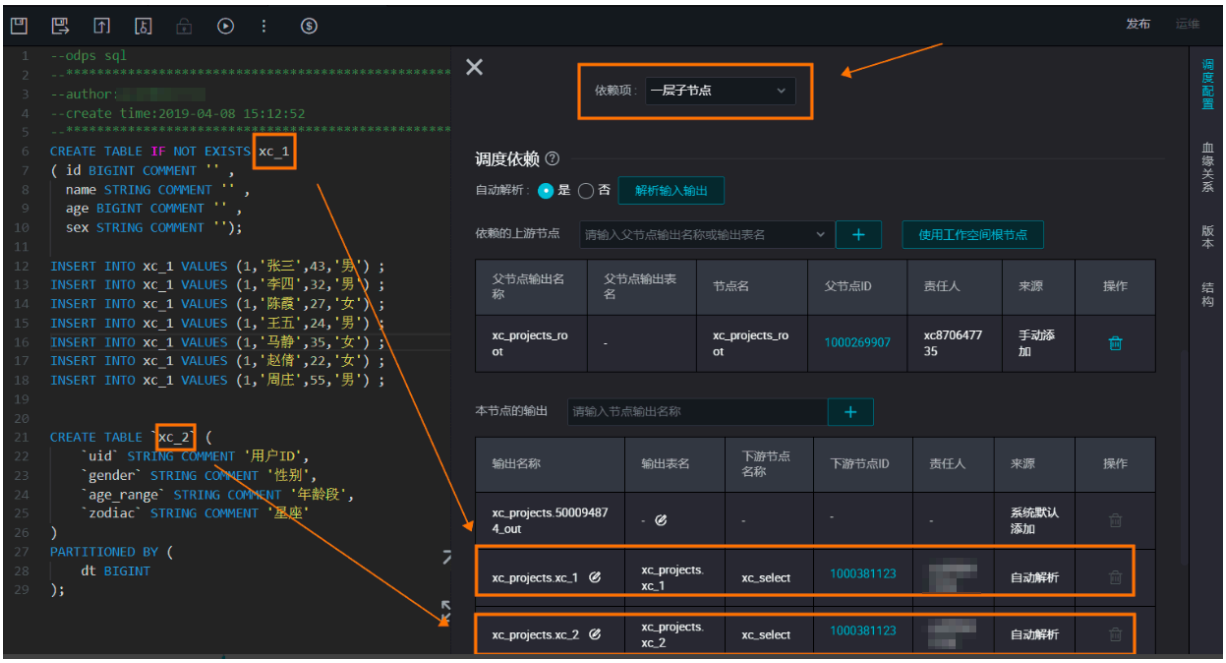
如上图中的SQL节点内容所示，xc_select节点查询xc_create节点中的表数据，通过自动解析功能，自动将xc_create节点解析为本节点依赖的上游。

依赖上一周期：一层子节点

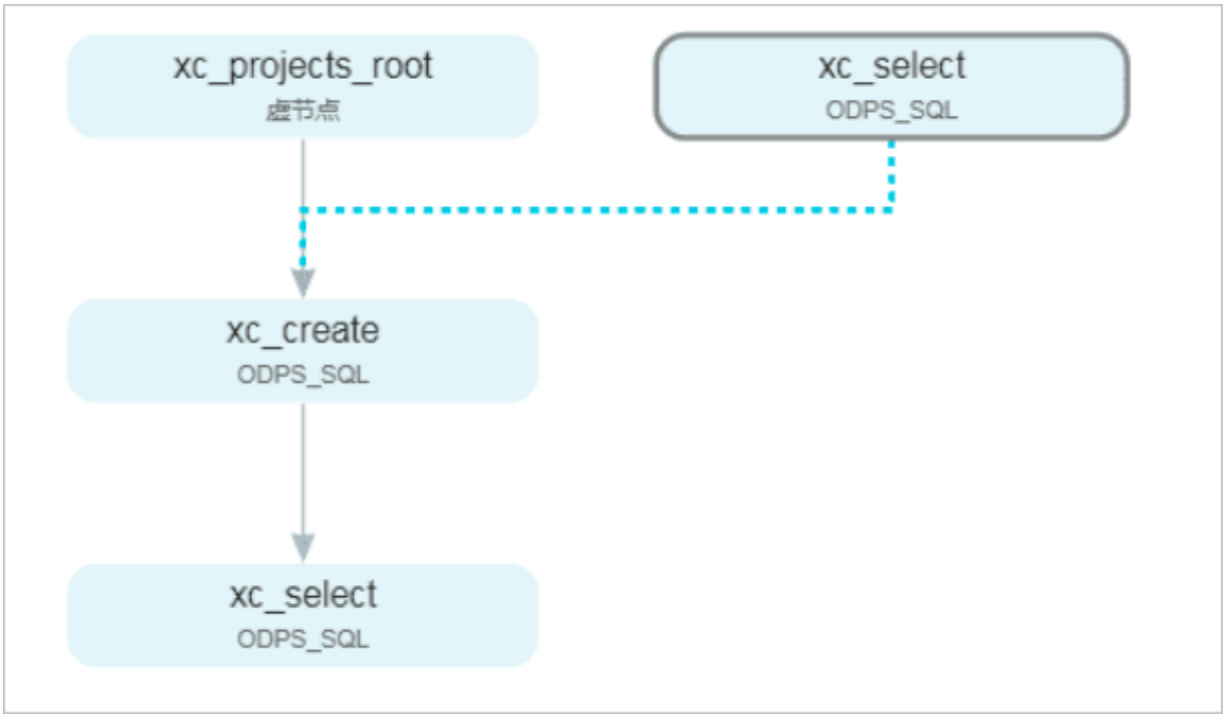
节点依赖：依赖当前节点的下游。例如节点A存在下游节点B、C、D三个节点，依赖一层子节点是节点A依赖B、C、D三个节点的上一周期。

业务场景：该节点运行依赖上一周期的下游节点，对本节点的结果表（即本节点输出表）进行清洗。如果下游节点运行成功，本周期的本节点开始运行，否则将不能运行。

xc_create选择依赖一层子节点。



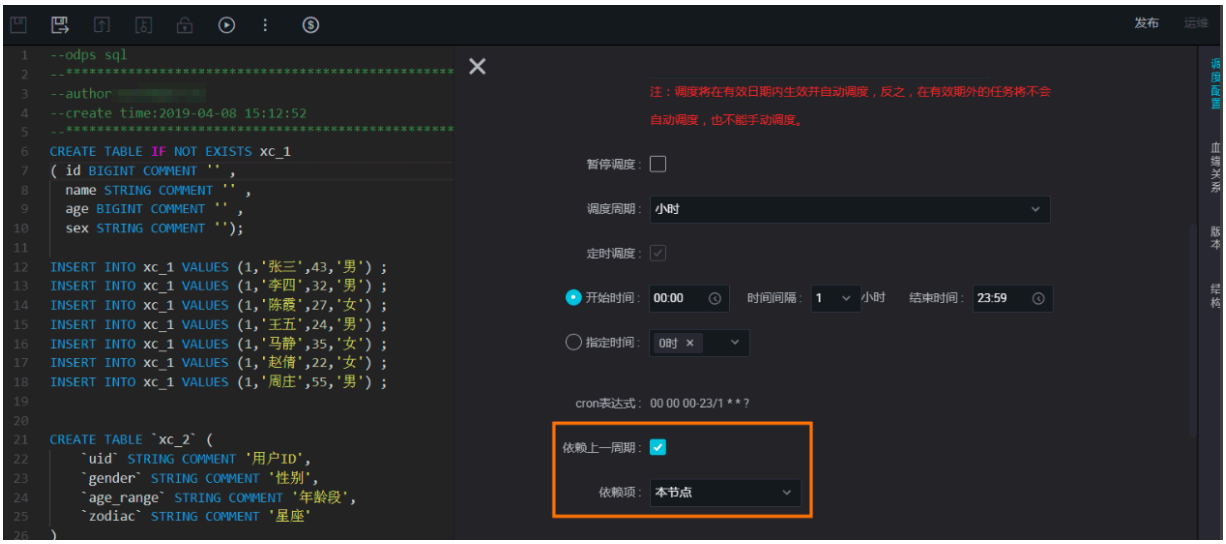
运维中心页面为您展示各节点的依赖关系。



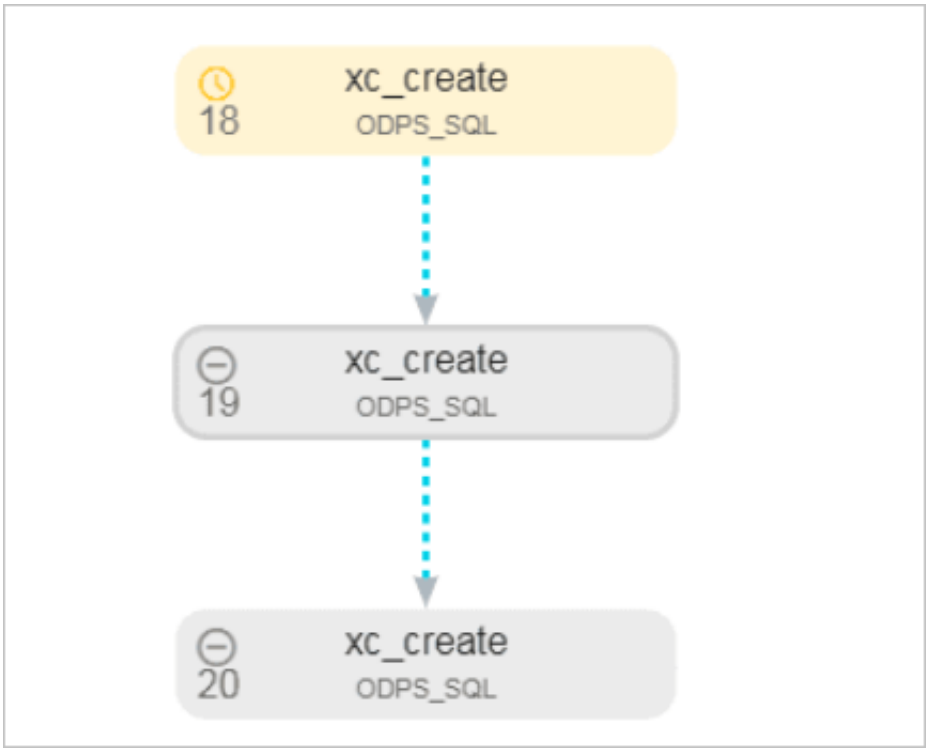
依赖上一周期：本节点

节点依赖：本次节点运行依赖本节点上一周期节点运行情况，如果上一周期节点未完成，将阻碍本周期节点运行。

业务场景：本次节点的数据依赖上次数据的清洗情况，此处将节点设置为小时调度。



您可以进入运维中心 > 周期实例页面，查看节点的依赖情况。

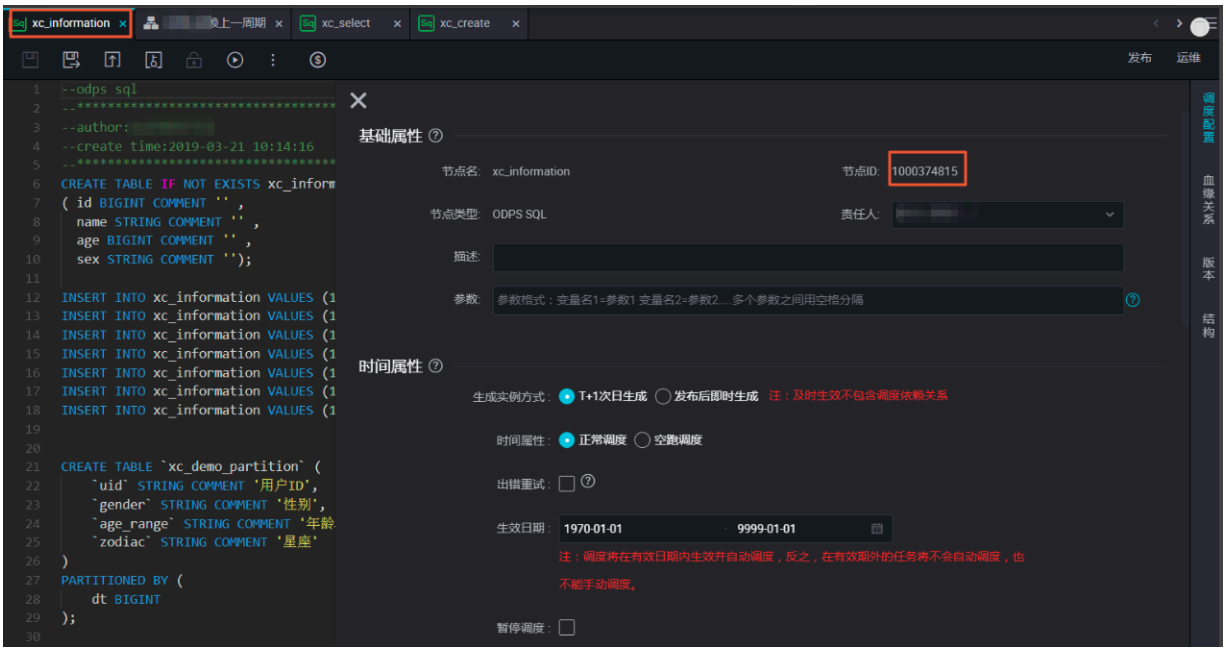


依赖上一周期：自定义节点

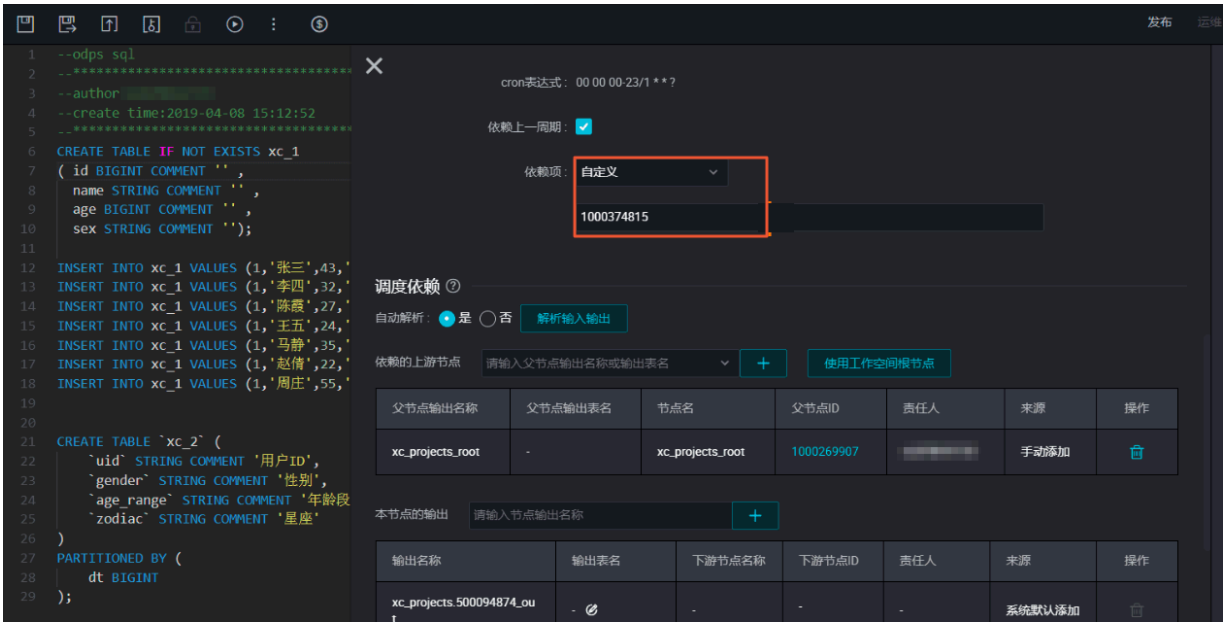
节点依赖：代码中没有用到1000374815节点的产出表，但业务上需要依赖该1000374815节点的上一周期是否正常产出数据。从节点关系来说，xc_create节点需要依赖1000374815节点的上一周期。

业务场景：业务逻辑上需要依赖1000374815节点正常产出的业务数据，但本节点（xc_create）中没有操作该业务数据。

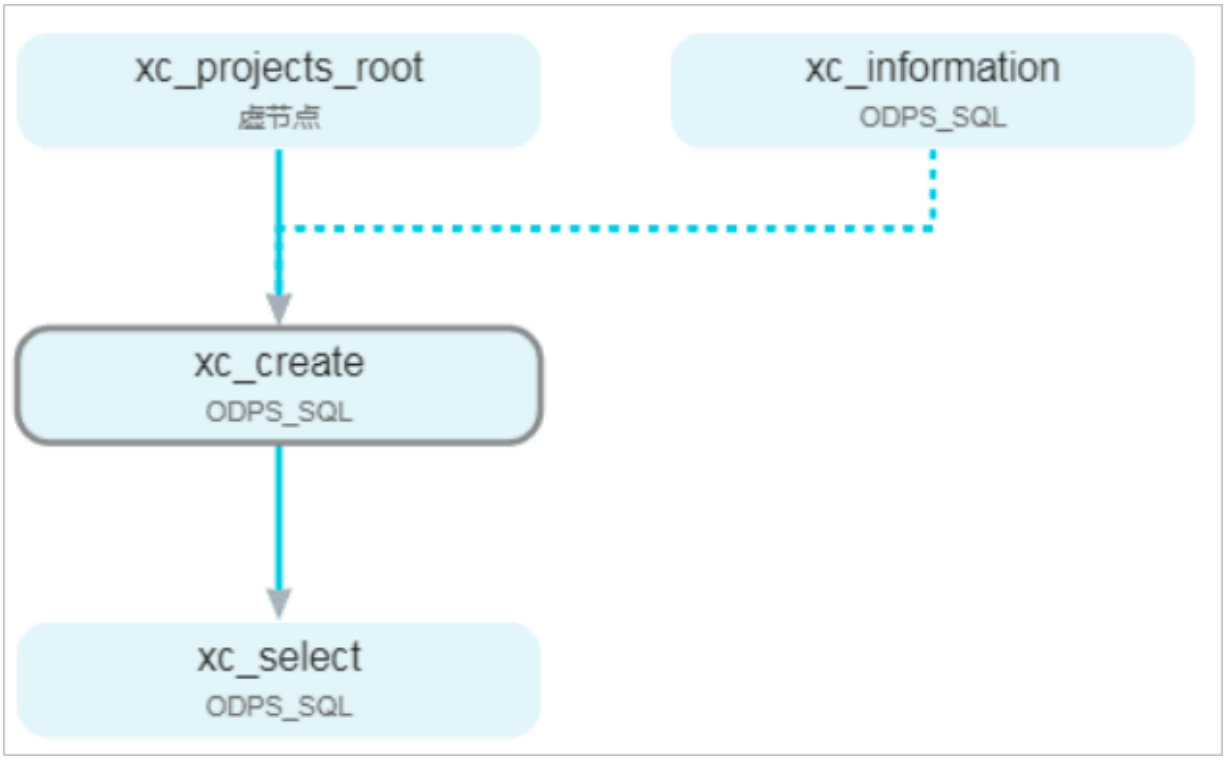
下图中的节点ID为1000374815。



xc_create节点依赖的上游自定义选择1000374815节点。



您可以进入运维中心 > 周期实例页面，查看节点的依赖情况。



3.6.6 资源属性

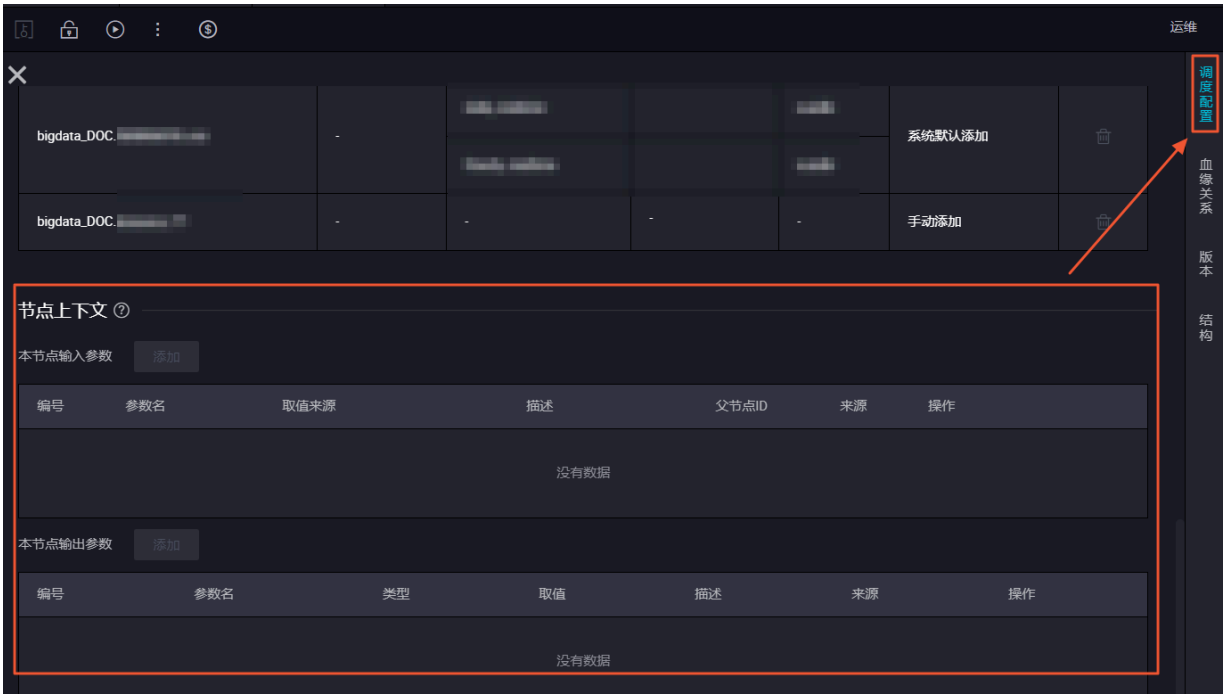
资源属性配置页面如下所示：

资源组：绑定任务调度的机器资源，默认有一个资源组。如果您有特殊需求，需要自己提供机器运行DataWorks任务，可以新增其他资源组。

3.6.7 节点上下文

节点上下文 用于支持参数在上、下游节点之间传递。基本使用方式是先在上游节点定义输出参数及其取值，然后在下游节点定义输入参数（取值引用上游节点的输出参数），即可在下游节点中使用此参数来获取上游节点传递过来的取值。

您可以在特定节点的调度配置页面中的 节点上下文部分进行配置，如下图所示。



输入参数

节点的输入参数用于定义对其依赖的上游节点的输出的引用，并可在节点内部使用，使用方式与其它参数类似。

- 输入参数的定义

1. 在调度依赖中添加依赖的上游节点。

2. 在节点上下文中，添加输入参数定义，取值选择引用上游节点的输出参数。

各字段含义如下。

字段	含义	备注
编号	编号指示，系统控制，自动增加	无
参数名	定义的输入参数名称	无
取值来源	参数的取值来源，引用上游节点取值	取值是上游节点输出参数的具体取值
描述	参数的概要描述	会自动从上游节点解析得到
父节点ID	父节点ID	会自动从上游节点解析得到
操作	提供 编辑 和 删除 两个操作	无

- 输入参数的使用

  在节点中使用定义的输入参数的方法和其它系统变量一样的方式，引用方式为 `${输入参数名}`。例如在Shell节点中引用方式如下图所示。

输出参数

您可以在节点上下文中定义本节点输出参数，输出参数的取值分为常量和变量两种类型：常量为固定字符串，变量指系统支持的全局变量。输出参数定义完成且节点提交后，即可在下游节点中引用，作为下游节点的输入参数的取值。

> **说明:**
> 不支持在当前节点（例如PyODPS节点）内部编写代码的方式来对定义的输出参数进行赋值。

各字段含义如下。

字段	含义	备注
编号	编号值由系统控制，自动增加	无
参数名	定义的输出参数名称	无
类型	参数类型	分常量和变量两种

取值	取值来源	1. 常量可直接输入一个字符串。 2. 变量仅支持系统变量、调度内置参数、${…}自定义参数和$[…]自定义参数。参见系统支持的全局变量及参数配置。
描述	参数的概要描述	无
操作	提供编辑和删除两个操作	当存在下游节点依赖时，不支持编辑和删除。在下游节点添加对上游节点引用之前，请谨慎检查，确保上游输出定义正确。

系统支持的全局变量

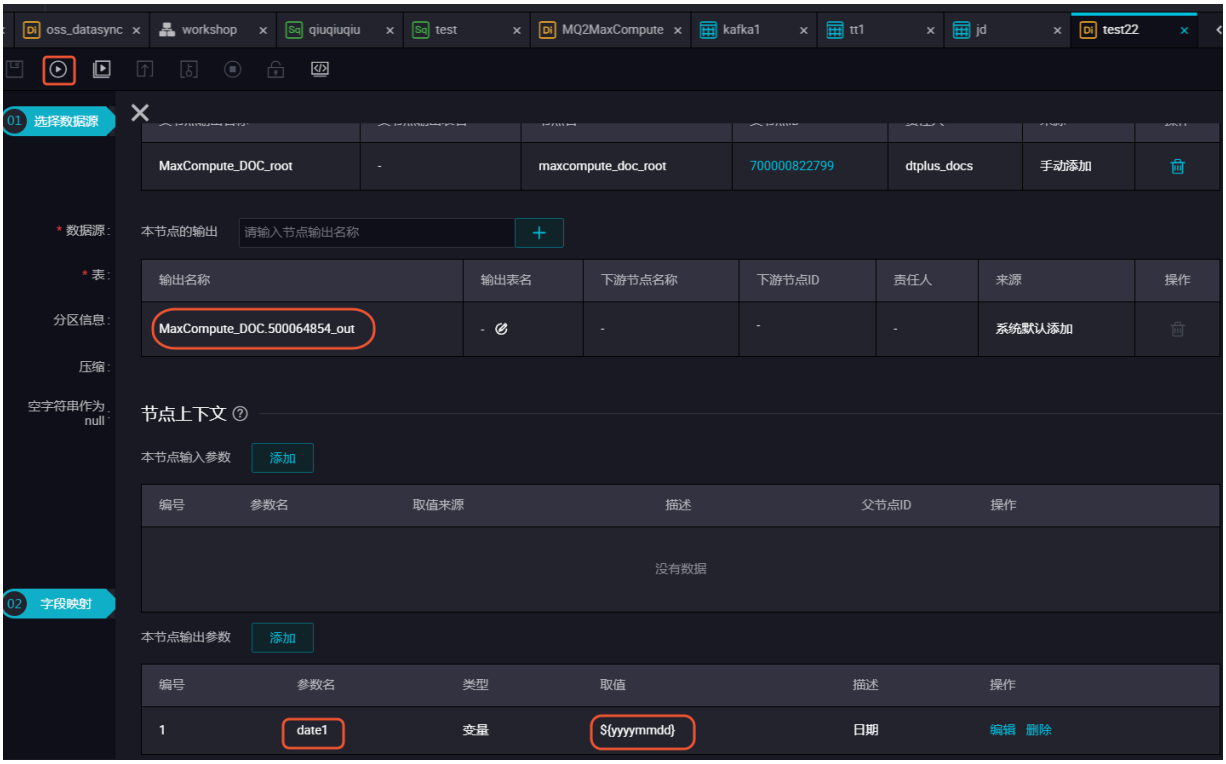
- 系统变量

```
${projectId}: 项目ID
${projectName}: max compute项目名
${nodeId}: 节点ID
${gmtdate}: 实例定时时间所在天的00:00:00，格式为yyyy-MM-dd 00:00:00
${taskId}: 任务实例ID
${seq}: 任务实例序号，代表该实例在当天同节点实例中的序号
${cyctime}: 实例定时时间
${status}: 实例的状态—成功（SUCCESS）、失败（FAILURE）
${bizdate}: 业务日期
${finishTime}: 实例结束时间
${taskType}: 实例运行类型—正常（NORMAL）、手动（MANUAL）、暂停（PAUSE）、空跑（SKIP）、未选择（UNCHOOSE）、周月空跑（SKIP_CYCLE）
${nodeName}: 节点名称
```

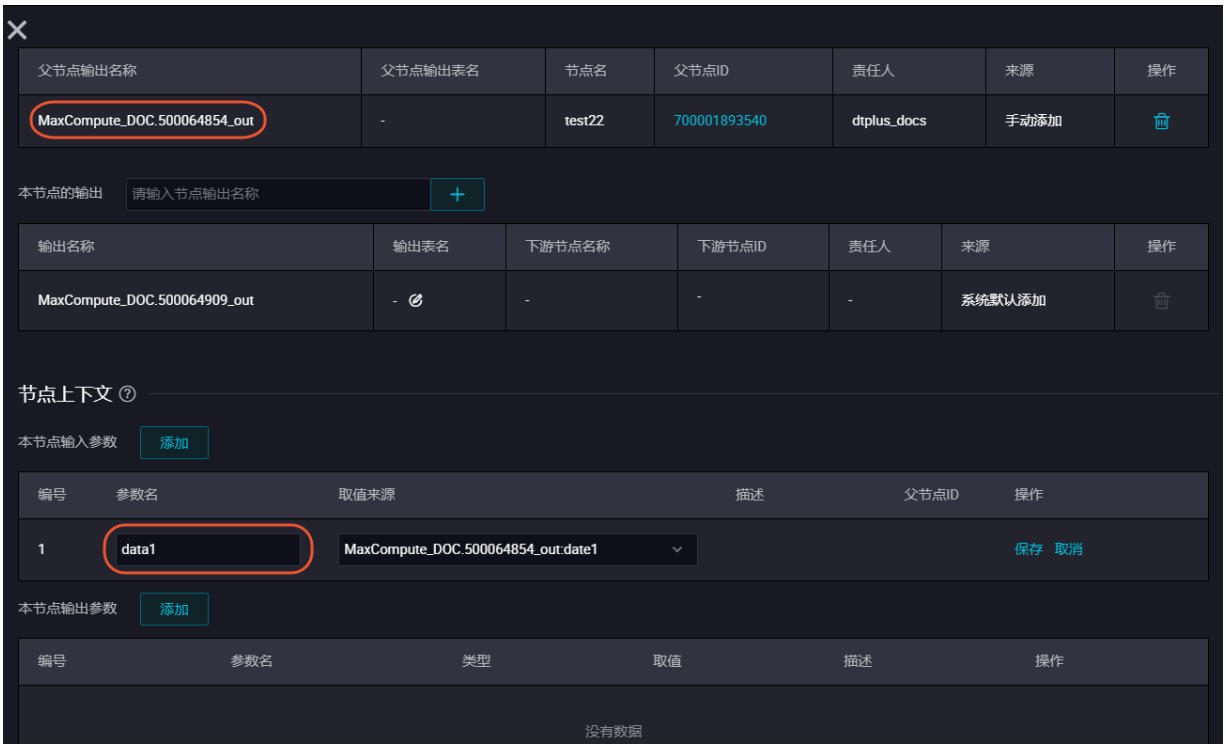
- 其他参数设置请参见#unique_39。

示例

节点test22是节点test223的上游节点,首先配置节点test22节点上下文的本节点输出参数。本例中参数名为date1，取值为${yyyymmdd}，提交节点，如下图所示。



节点test22提交成功后，配置下游节点test223，请注意保证父节点输出名称为test22的本节点的输出。在节点上下文的本节点输入参数准确输入test22的参数名称`date1`，即可进行取值来源的选取。点击保存后即可看到配置效果。



3.6.8 实时转实例

您可以通过实时转实例功能，发布代码后即时生成实例，然后在运维中心实时查看任务的调度依赖情况。

实例生成方式

目前有以下两种实例生成方式：

- T+1次日生成

  全量任务转实例（23:30之前提交发布的任务，第二天产生实例。23:30~00:00提交发布的任务，第三天产生实例）。

- 发布后即时生成

  实时转实例（发布代码后立即生成实例）。

创建实时转实例的节点

1. 新建业务流程。

   右键单击数据开发下的业务流程，选择新建业务流程。

2. 在新建的业务流程下创建节点任务（以ODPS SQL节点为例）。

3. 双击打开该节点，编辑代码后，单击右侧的调度配置，选择节点生成实例方式为发布后即时生成。

说明:

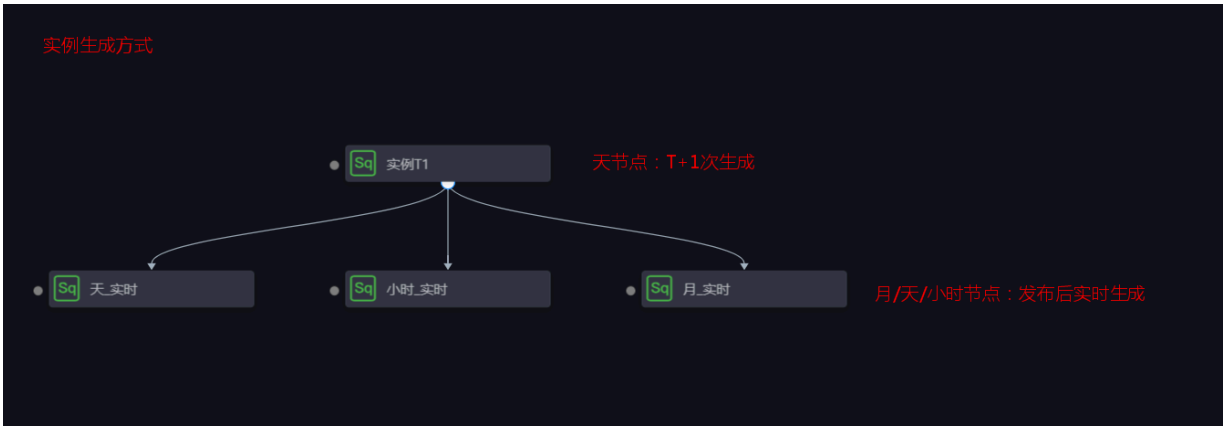
- 组合节点不支持实时转实例功能。
- 全量转实例期间（23:30~24:00点期间）不能进行实时转实例，可以提交发布，但不会转出实例。
- 已有的实例从T+1次日生成变更为发布后即时生成，会影响当天实例的产生情况。已运行的实例会保留，发布时间未运行的实例会被删除并替换为实时转出的实例。
- 上游T+1形式生成实例，下游实时转实例节点会变成孤立节点，任务不会调度。

- 上下游均为实时转实例节点，发布时间点十分钟后的实例会正常转出，任务定时时间在发布时间点十分钟前，会生成已经完成的实例，节点实例属性为实时生成的过期实例，下游正常调度。

  示例场景：假设任务调度时间设置18点执行，修改作业调度时间为17点（立即生成实例）。如果在16:50前发布任务，只会生成一个实例。如果在16:50后发布任务，会生成一个已完成的实例，节点实例属性为实时生成的过期实例。

使用场景

实时转实例使用场景通常为：上游T+1形式生成实例，下游实时转实例，节点间的依赖关系如下图所示。



实际应用时通常出现以下三种情况：

- 上下游均为当天新增节点
  - 下游天节点：下游天节点实时转出来的实例只有该节点的实例，并且没有上游节点的依赖。跨周期自依赖不影响当天的实例，跨周期自定义依赖可依赖到已有的实例上。
  - 下游小时/分钟节点：如果是跨周期自依赖，除第一个实例没有上游节点外，其他的实例都有上游节点。
  - 下游周/月节点：只会存在该节点对应的实例，没有上游节点。

    说明：

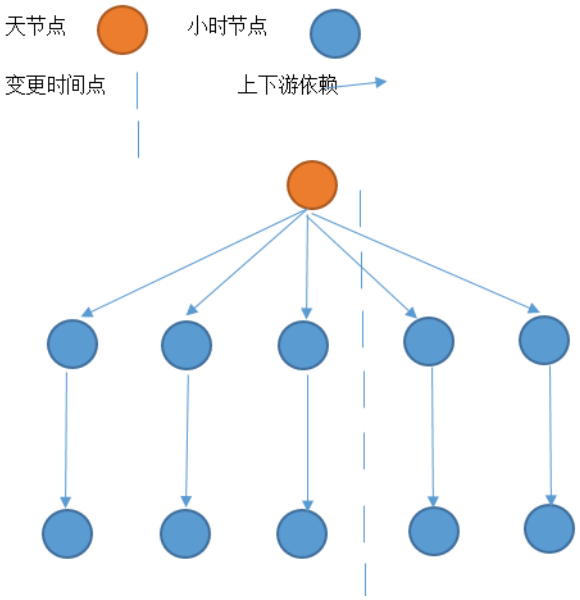
    由于上游天节点第二天才会产生实例，下游的实时转出的实例节点会变成孤立节点。孤立节点没有上游所以不会被调度运行。

    如果下游新增节点为跨周期自依赖，则第二天上游实例产生时，下游会由于跨周期自依赖到前一天的孤立节点上，从而导致整个任务被孤立且不被调度运行。

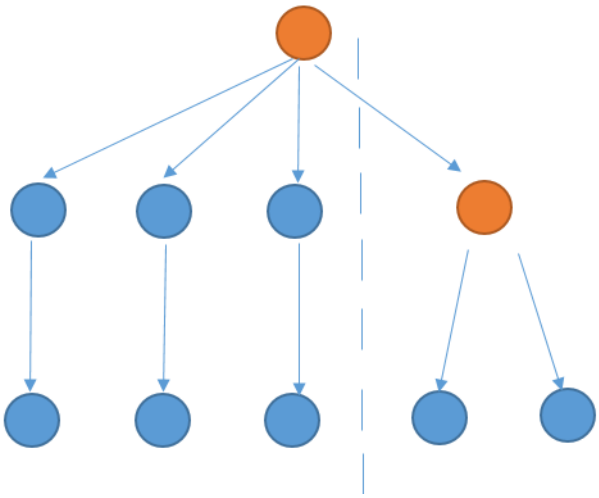
  总结：新增跨周期自依赖的实时转实例节点，除第一个实例没有自依赖的上游外，其他实例依赖关系正常，但整个任务无法正常调度运行。
- 上游实例已经产生，下游新增实时转实例节点
  - 下游天节点：产生的实例会依赖原有的上游天节点实例，跨周期自依赖不影响当天的实例。
  - 下游小时/分钟节点：产生的实例会依赖原有的上游天节点实例，如果跨周期自依赖，第一个实例不会有自依赖的上游，而其他实例则会正常自依赖。
  - 下游周/月节点：产生的实例会依赖原有的上游天节点实例，跨周期依赖不影响当天的实例。自依赖情况下，如果前一天有节点产生，则会相应进行依赖。

  总结：任何跨周期自依赖的调度是否成立，都需要以前一天该节点是否可以正常调度运行作为依据。

- 上游天节点实例已经产生，下游小时节点更新为实时转实例的天节点
  - 变更前任务情况：上下游均为（T+1）的小时节点。



  - 更新操作：将依赖上游天节点的小时节点变更为实时转出实例的天节点。
  - 变更后实例生成与节点依赖情况：假设在上图虚线所示时间点提交发布，则该实例会删除原有该时间点10分钟后此节点的实例，并且产生一个新的天节点实例。该节点下游的小时节点会全部依赖到新产生的天节点，如果是跨周期自依赖的节点变更，产生的天节点实例会依赖上T+1次实例生成的实例。



  - 变更后当天实例情况：发布前的实例为小时节点，发布后为天节点。

总结：变更节点的原有上下游依赖关系不变，当天的实例情况会受到影响。

3.7 配置管理

3.7.1 配置管理概览

配置管理是对DataStudio界面的配置，包括代码、文件夹、主题、增删模块等功能。

您可以通过单击数据开发左下角的小齿轮进入配置管理页面。

配置管理分为五大模块，详情请参见下述文档。

- #unique_377
- #unique_370
- #unique_378
- #unique_379
- #unique_380

3.7.2 配置中心

您可以通过配置中心随心组合您的DataStudio模块和编辑器。

配置中心是对常用功能的设置，包括模块管理、编辑器管理两部分。

模块管理

模块管理是对DataStudio界面左侧栏功能模块的模块增删操作，可以点击筛选需要在左侧栏显示的功能模块，并可以通过拖拽来完成对模块功能的排序。

当鼠标放在需要添加的模块上时，模块会变成蓝色并显示添加字样。

当鼠标放在需要删除的模块上时，模块会变成红色并显示移除字样。

说明:

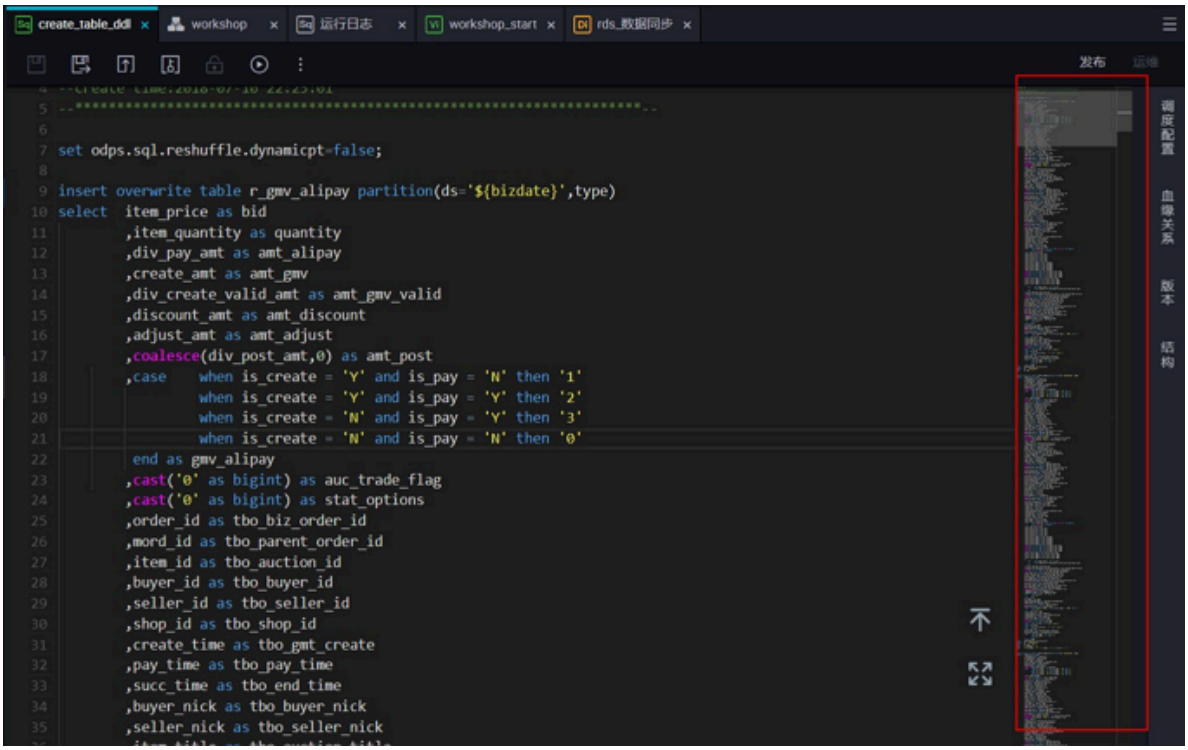
模板管理筛选会立即生效，并对当前项目生效，如需对所有项目生效，请单击下方的以上设置应用到所有项目中。

编辑器管理

编辑器是对代码和关键字的设置，设置实时生效，无需刷新界面。

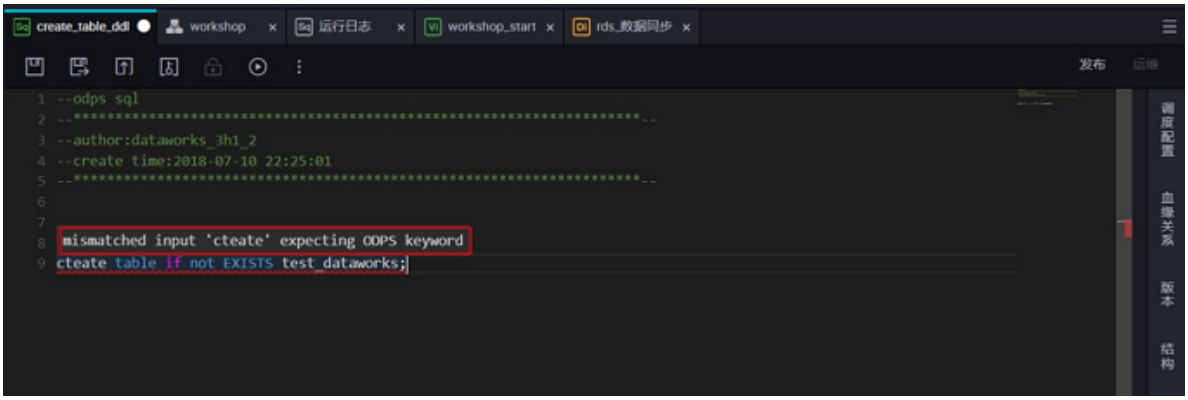
- 缩略图

  在代码的右侧显示当前界面代码的展示图，图中阴影区域为当前正在展示的区域，当代码较长时，可以上下移动鼠标来切换显示的代码区域。



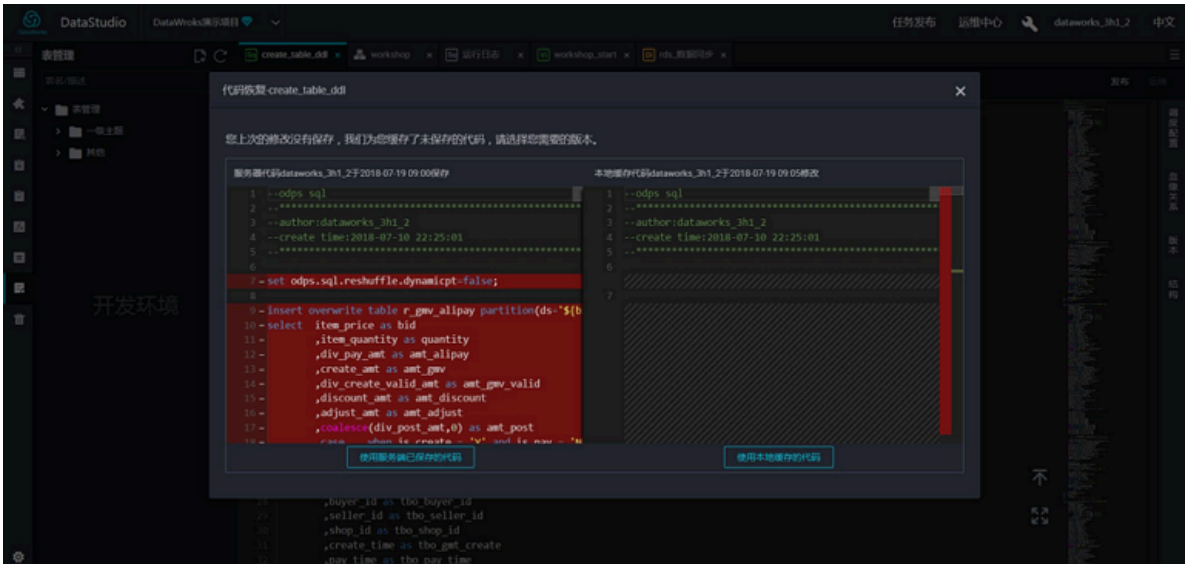
- 错误检查

  检查当前代码中的错误语句，当鼠标放在红色错误代码区域，会显示具体出错的字段情况。



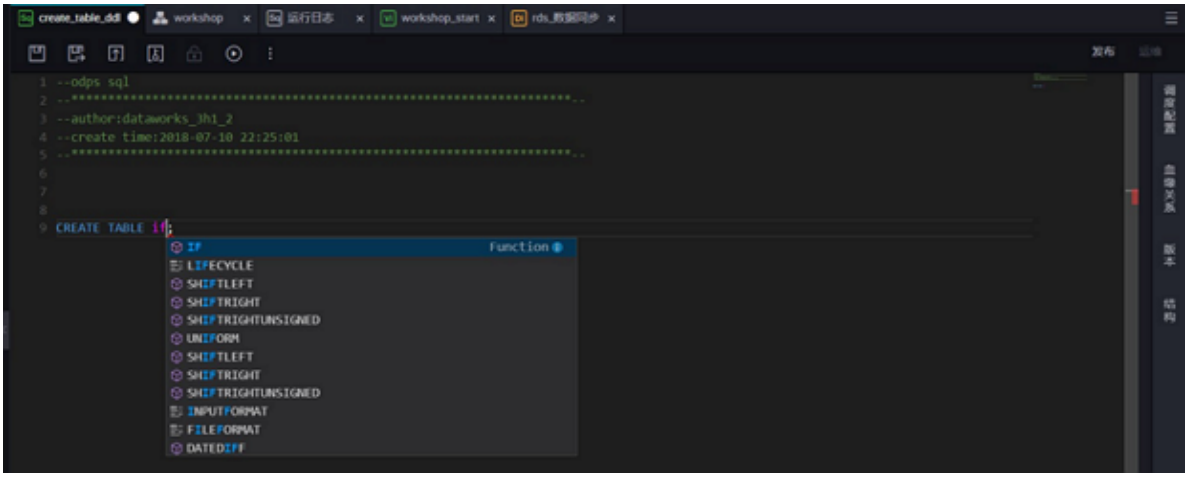
- 自动保存

  对当前编辑的代码自动缓存，避免在编辑过程中，页面崩溃导致代码无法保存。可以选择左侧使用服务端已保存的代码或右侧使用本地缓存的代码。



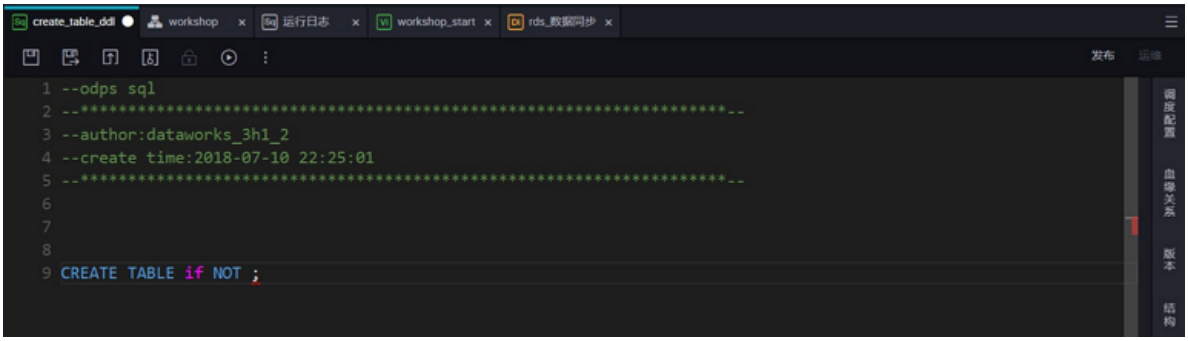
- 代码风格

  代码风格可设置大写或者小写，根据您喜欢的风格选取，输入关键字敲击回车，即可通过联想快捷输入需要的关键字。



- 代码字体大小

  代码字体大小支持最小12号、最大18号字体，根据您的代码书写习惯和数量更改设置。



- 代码提示

  代码提示用于在输入代码过程中，智能提示的显示，分为以下几个部分。

  - 空格智能提示：选取联想的关键字、表、字段后添加一个空格。
  - 关键字：提示代码支持输入的关键字。
  - 语法模板：支持的语法模板。
  - Project：输入联想的项目名称。
  - 表：联想需要输入的表。
  - 字段：智能提示此表中的字段。

- 主题风格

  主题风格是对DataStudio界面风格的设置，目前支持黑色和白色两种。

- 设置应用

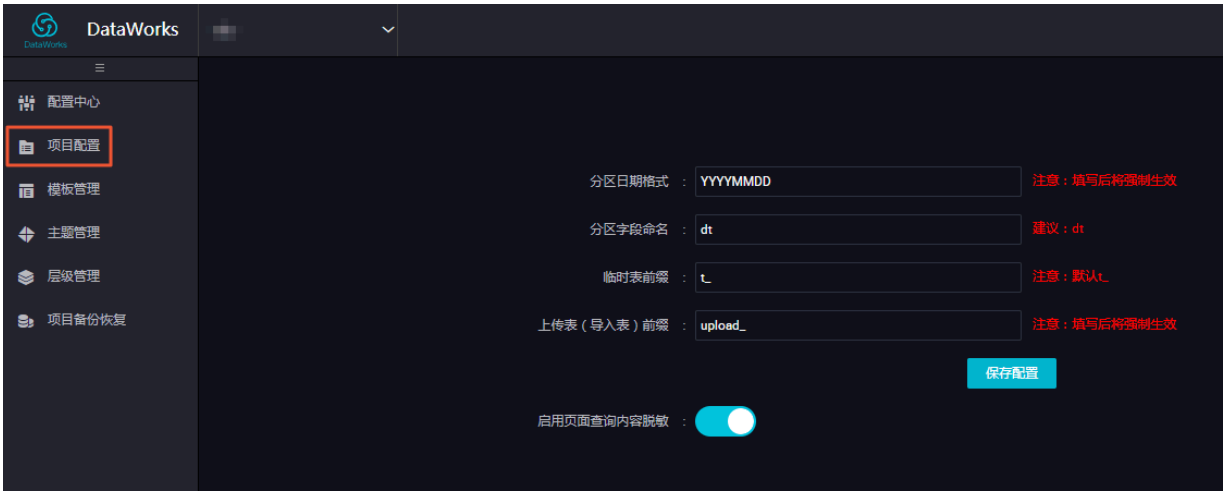
  将以上模板管理和编辑器管理的设置应用到当前已有的所有项目下。

3.7.3 项目配置

项目配置页面包括分区日期格式、分区字段命名、临时表前缀、上传表（导入表）前缀和启用页面查询内容脱敏5个配置项。

单击数据开发左下角的小齿轮进入配置管理页面。

单击左侧菜单栏中的项目配置。



配置	说明
分区日期格式	默认参数、代码中参数的显示格式，您也可以根据自己的需求修改参数的格式。
分区字段命名	分区中默认的字段名称。
临时表前缀	以t_开头的字段，默认识别为临时表。
上传表（导入表）前缀	在DataStudio页面上传表时，表的名称前缀。
启用页面查询内容脱敏	启用后，当前工作空间下的临时查询任务返回的结果将会被脱敏。

开启DataWorks工作空间的查询脱敏

DataWorks的脱敏需要在每个工作空间进行逐一开启。开启脱敏后，会脱敏当前工作空间下的临时查询任务返回的结果。由于仅仅是动态脱敏，不会影响底层存储的数据。

> 说明:
> 例如工作空间A设置了展示脱敏，但工作空间B没有设置，如果您可以从工作空间B访问A的表，将会看到明文结果。

进入项目配置页面，将启用页面查询内容脱敏选项启动，单击保存配置，即可开启DataWorks工作空间的查询脱敏。

> 说明:
> DataWorks默认不允许下载和默认数据脱敏。

DataWorks查询脱敏配置成功后，默认对以下数据进行脱敏。

类型	蚂蚁脱敏规范	原始数据	脱敏数据
身份证	前1后1，适用于15位和18位的身份证。	512345678943215678	5**********8
手机号	前3后2，适用于大陆手机号。	18112345678	181******78
邮箱	@前仅展示前3位，如果不够3位则显示全部，后面跟3个*。	· eftry.abc@gmail.com · af@abc.com	· eft***@gmail.com · af***@abc.com
银行卡	只显示最后4位，适用于信用卡和储蓄卡。	· 1234576834509782 · 643257829145430986	· **********9782 · **********0986
ip/mac地址	只保留第1段。	· 192.000.0.0 · ab:cd:11:a3:a0:50	· 192.*.*.* · ab:**:**:**:**:**
车牌号	地区信息+车牌后3位显示明文，其他都用*展示。	· 浙AP555B · 浙ADP555T	· 浙A**55B · 浙A***55T

说明:

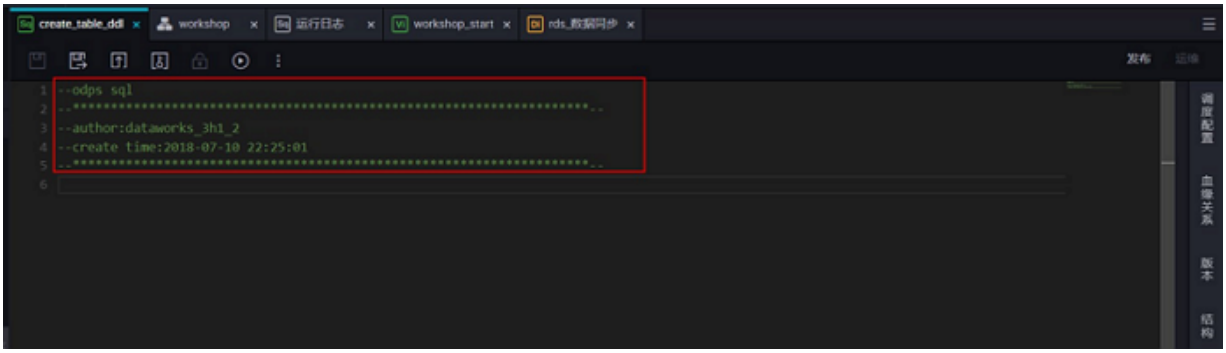
如果需要对更多类型的数据进行脱敏，或者对脱敏格式有自定义要求，请使用数据保护伞的脱敏配置，且为工作空间开启脱敏功能必须配合数据保护伞使用，详情请参见数据保护伞模块。

3.7.4 模板管理

模板管理是在节点创建后，默认展示在代码最前端的内容，项目管理员可以根据需求修改模板的显示样式。

目前支持对ODPS SQL模板、ODPS MR模板和SHELL模板设置title。

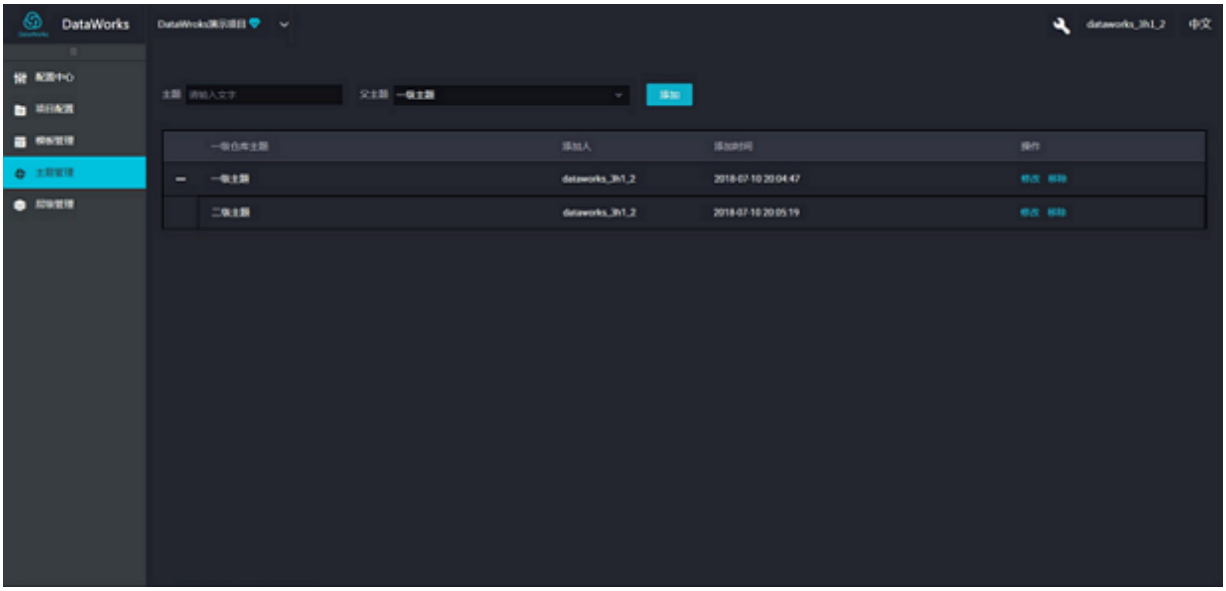
以SQL节点为例，模板展示样式。



3.7.5 主题管理

在表管理中存在非常多的表，您可以按照选取的主题将表存放在二级子文件夹下。这些用于表归纳整理的文件夹，即为主题。

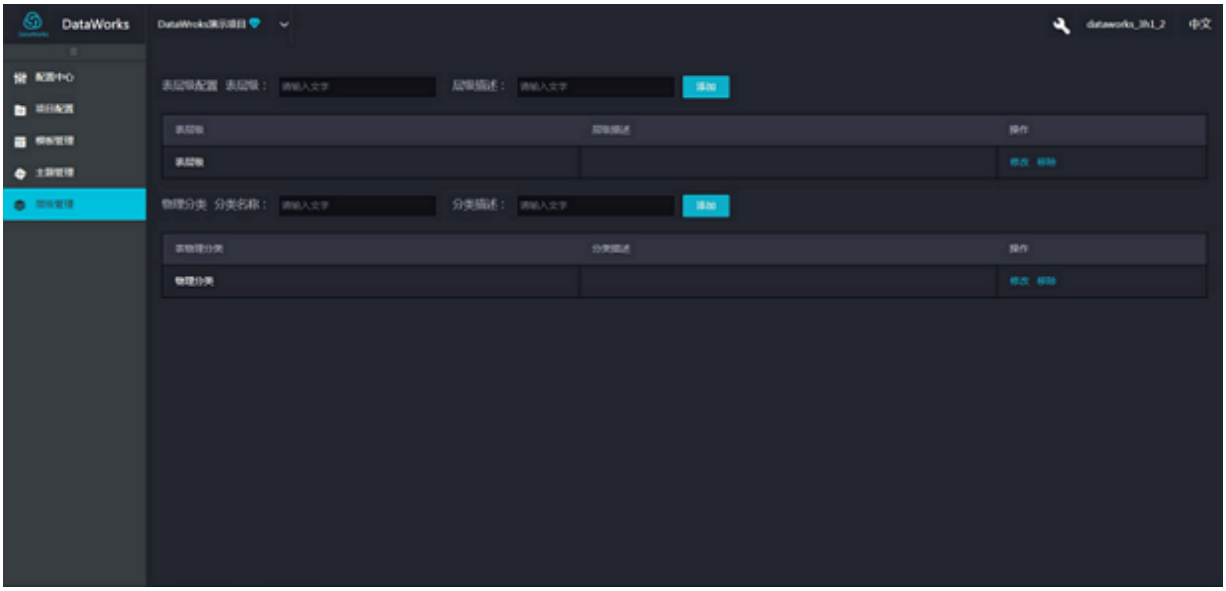
管理员可根据项目需求，添加多个主题，将表按照用途、名称进行分类归纳整理。



3.7.6 层级管理

层级管理用于对表的物理层级进行设计。

根据表对项目的重要程度，划分整理您的表。避免当一个表出现问题时，无法精确定位到此表，影响线上作业的正常运行。



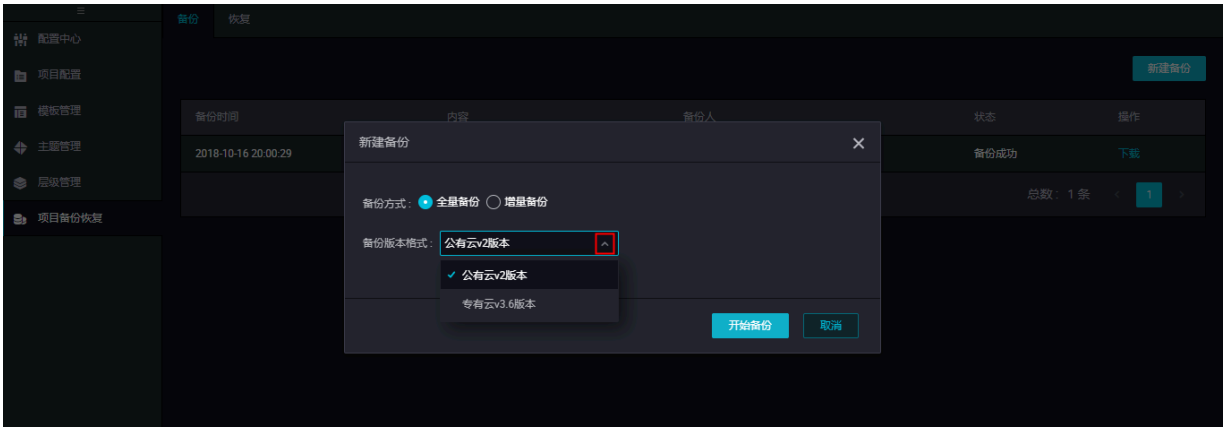
项目不存在默认的层级，需要项目owner或者管理员根据项目的用途和需求手动添加。

3.7.7 项目备份恢复

项目备份恢复主要用于备份代码。在您备份的同时，您的资源也会同时备份。

> 说明:
>
> · 仅项目管理员可以导出配置及恢复配置，进入配置管理页面的方式请参见配置管理概览。
> · 备份时无法对旧版工作流进行备份，建议您使用业务流程进行开发。

新建项目备份时，您可选择全量备份或增量备份。备份包括公有云和专有云两种版本格式。



> 说明:
>
> · 您可直接下载备份的文件，格式为XML。

- 备份后的任务可以进行恢复，但恢复时可能会出现错误，因此建议尽量选择全量备份。

3.8 发布管理

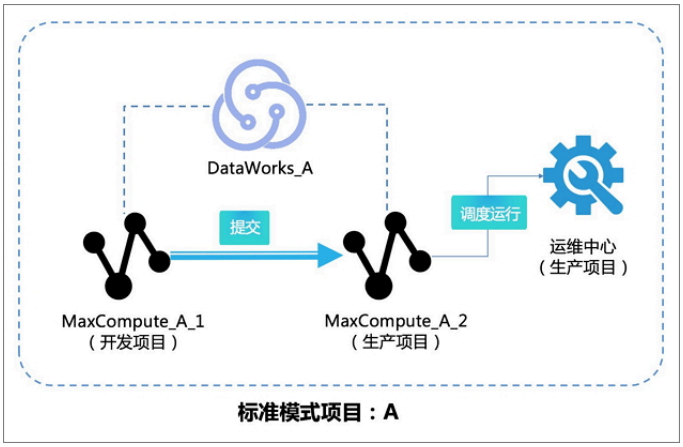
3.8.1 任务发布

在严谨的数据研发流程下，开发者通常会在用于开发的项目内，完成代码研发、流程调试、依赖属性配置和周期调度属性配置后，再将任务提交至用于生产环境进行调度运行。

DataWorks的标准模式为您提供在一个项目内，完成从开发到生产的全链路能力和无缝的体验，建议您通过该模式来完成数据研发与生产发布。

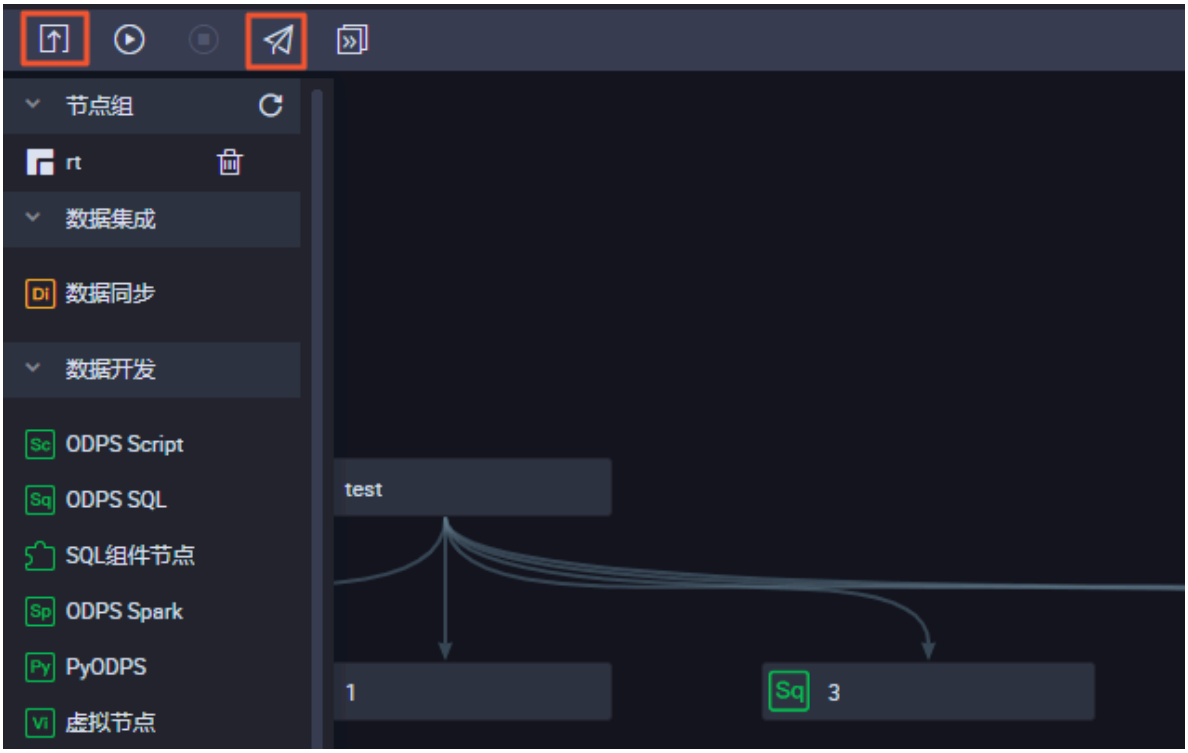
标准模式任务发布

当您的DataWorks工作空间为标准模式时，系统默认一个DataWorks工作空间对应两个相互绑定的MaxCompute项目（开发环境与生产环境），您可以直接将任务从开发环境提交并发布至生产环境。



操作步骤如下：

1. 将代码、任务调试并配置完成后，单击提交，检查代码之间的依赖关系是否正确。



> **说明:**
> 如果您的节点已经提交过。在没有修改节点内容，只是修改了业务流程或节点属性的情况下，可以不选择节点（如果节点已经被提交过，在不改变节点内容的情况下节点无法被再次选择），填写备注后提交业务流程。相关改动会正常被提交。

2. 提交通过后，单击发布。
3. 在创建发布包页面批量勾选所需发布的任务，单击添加到待发布，则任务会进入待发布列表页面。

   您可以根据提交人、节点类型、变更类型、提交时间和任务名称或ID等条件过滤和搜索任务。如果您单击发布选中项，则会立即发布至生产环境调度运行。

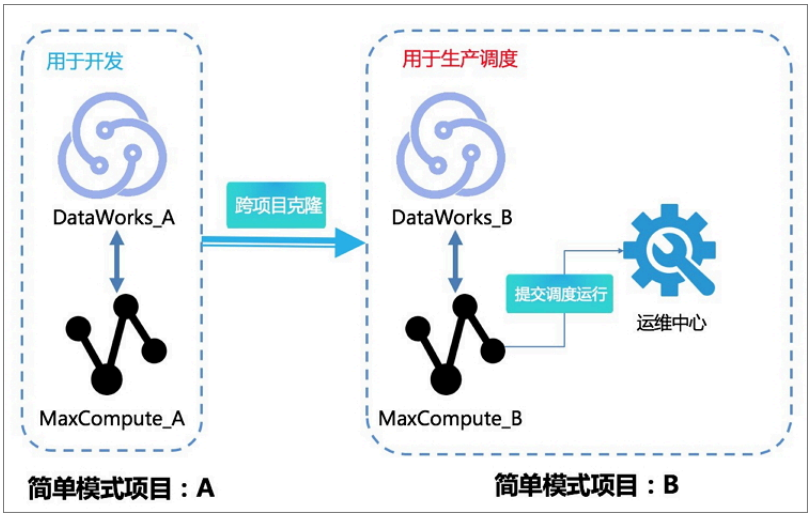
4. 单击打开待发布，确认待发布列表中的信息无误后，单击全部打包发布，即可将列表中的任务发布至生产环境。

> 说明：
> 标准模式严格禁止直接对生产环境内的表数据进行操作，您可以通过标准模式工作空间，获得始终稳定、安全、可靠的生产环境，因此建议您使用标准模式工作空间进行任务的发布与调度。

简单模式跨项目克隆

简单模式项目没有任务发布的概念，如果您想要实现简单项目内的开发、生产环境隔离，仅能通过把任务克隆至用于生产的项目并执行提交来实现，即：简单模式项目（用于开发）+简单模式项目（用于生产）。

如下图所示，用户创建的两个简单模式项目分别用于开发、生产，可以先使用跨项目克隆将A项目中的任务克隆至B项目，再将克隆过来的任务在B项目中提交至调度引擎进行调度。

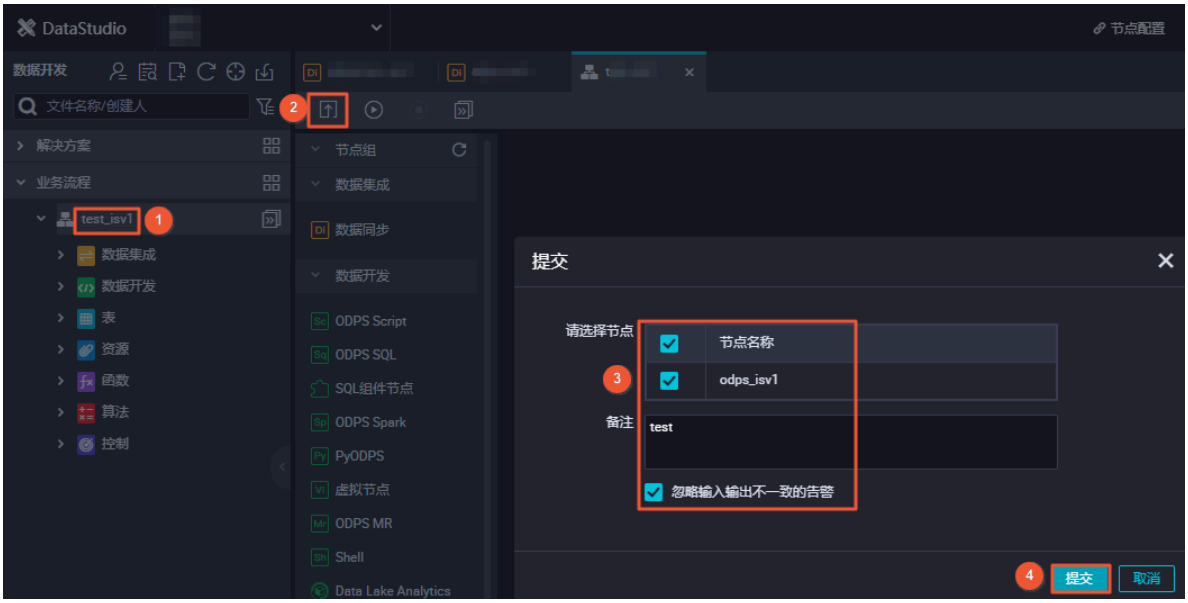


> 说明：
> - 权限要求：除项目管理员之外，执行操作的子账号需具有“运维”角色的权限（创建克隆包、发布克隆任务）才能独立完成该流程。

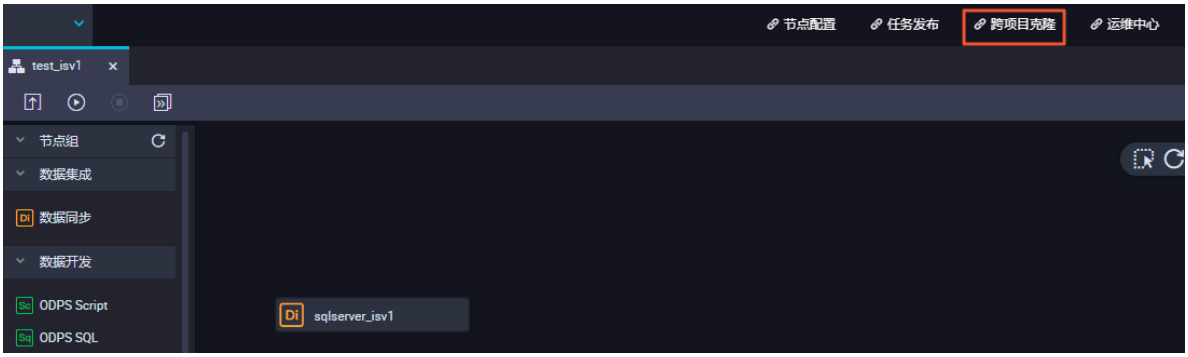
- 项目类型支持：仅简单模式项目支持克隆任务至其他项目，标准模式项目不支持克隆任务至其他项目。
- 准备工作：源项目A（简单模式项目）、目标项目B（标准模式项目）。

1. 提交任务。

   任务编辑完成后，选择需要克隆的任务执行提交。



2. 单击右上角的跨项目克隆。



3. 在已提交过的任务列表中，选择需要克隆的任务名称与需要克隆至的目标工作空间名称，单击添加到待克隆。

4. 执行克隆。

   单击打开待克隆，检查所需克隆的任务信息无误后，单击全部打包克隆。

   在确认克隆对话框中，单击克隆，即可完成克隆流程。

5. 查看克隆成功的任务。

   您可以在源工作空间的克隆包列表页面，查看克隆成功的任务集合。

   进入目标工作空间，可以查看到克隆的任务。

> 说明：
> 跨项目克隆时，处理任务间的依赖关系的详情请参见#unique_392。

3.8.2 任务下线

任务下线是指在某些情况下，需要将任务永久删除，包括开发环境的任务下线和生产环境的任务下线两种场景。

开发环境的任务下线

1. 登录DataWorks控制台，进入数据开发页面。

2. 通过任务节点类型、关键字来搜索需要删除的任务。

3. 右键单击要删除的任务，选择删除，则开发环境任务下线完成。

生产环境的任务下线

当已发布到生产环境的任务因某些原因需要删除时，需要根据删除任务>发布下线任务>执行发布的流程进行下线。

说明:

子节点依赖关系的处理：由于生产环境的任务下线涉及到子节点依赖的问题，因此在删除任务前，请先一层层往下处理好子节点的依赖关系后，再进行删除。否则会提示存在子节点，下线失败。

处理步骤如下：

1. 查找此节点的下游节点，可在工作流管理查看生产调度依赖关系。
2. 在数据开发页面重新编辑此子节点的父节点，或者直接删除此子节点。

如果提示子节点下还有子节点，请参见上述步骤逐层往下处理。

1. 删除任务。

   可以参见前文开发环境的任务下线操作，删除需要下线的任务。

2. 发布下线任务。

   说明：
   仅管理员/运维角色具有发布权限。如果是其他角色，需要通知运维人员进行发布。

   a. 删除需要下线的任务后，单击右上角的任务发布。

   b. 在创建发布包页面，勾选需要下线的任务。

   c. 单击发布选中项。

   您也可以单击添加到待发布，进入待发布列表进行发布。

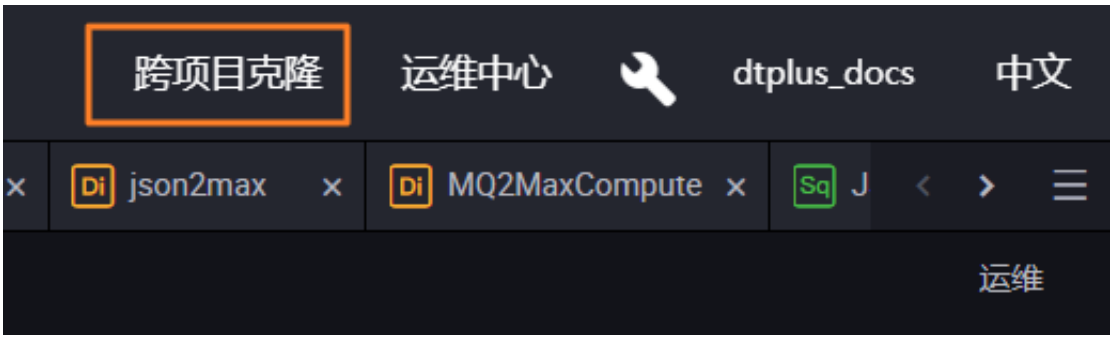
3. 执行发布。

单击确认执行对话框中的发布，完成下线任务的发布。

3.8.3 跨项目克隆说明

跨项目克隆主要用于同租户（云账号）简单模式下开发和生产环境的隔离，您也可以利用跨项目克隆功能实现计算、同步等类型的任务在项目之间的克隆迁移。本文将为您介绍如何处理跨项目克隆时任务间的依赖关系。

通过跨项目克隆功能进行克隆任务后，系统为区分同租户（阿里云帐号）下不同项目（project）之间任务的输出名称，会自动对每个任务输出名称作出一系列命名更改，目的是为了平滑复制依赖关系或保持原有依赖关系不变。



说明:

- 跨项目克隆不支持跨地域发布。

- 克隆责任人分为默认和克隆包创建者。
  - 当克隆责任人为默认的项目管理员时，克隆到目标工作空间后，您可以选择克隆后任务责任人为默认或克隆包创建者。

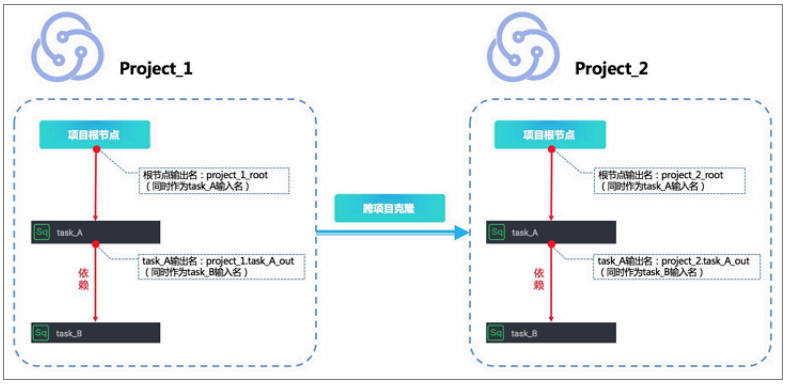
克隆成功后，责任人将第一优先级被置为原责任人。如果原责任人不在目标工作空间，则置为克隆包创建者。

  - 当克隆责任人为克隆包创建者时，克隆到目标工作空间后，您可以选择克隆后任务责任人为默认或克隆包创建者。

克隆成功后，责任人将第一优先级被置为原责任人。如果原责任人不在目标工作空间，会询问是否变更责任人。如果确认变更，则任务克隆成功且责任人变更为克隆包创建者。如果不变更责任人，则克隆任务取消。

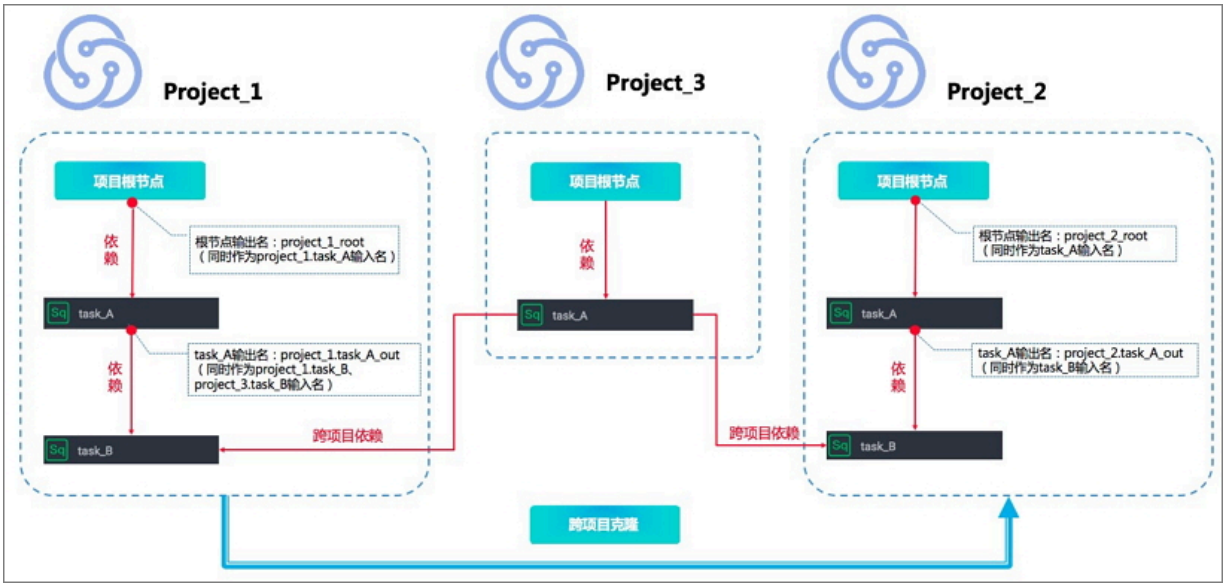
完整的业务流程克隆

用户使用task_A任务的输出点在project_1中为project_1.task_1_out，克隆至project_2之后输出点名为project_2.task_A_out。



跨项目依赖任务克隆

project_1中的任务task_B依赖了project_3中的任务task_A，在将project_1. task_B克隆为project_2.task_B之后，依赖关系将一同克隆，即project_2.task_B仍然依赖project_3.task_A。



3.8.4 跨项目克隆实践

本文将为您介绍跨项目克隆的操作实践。

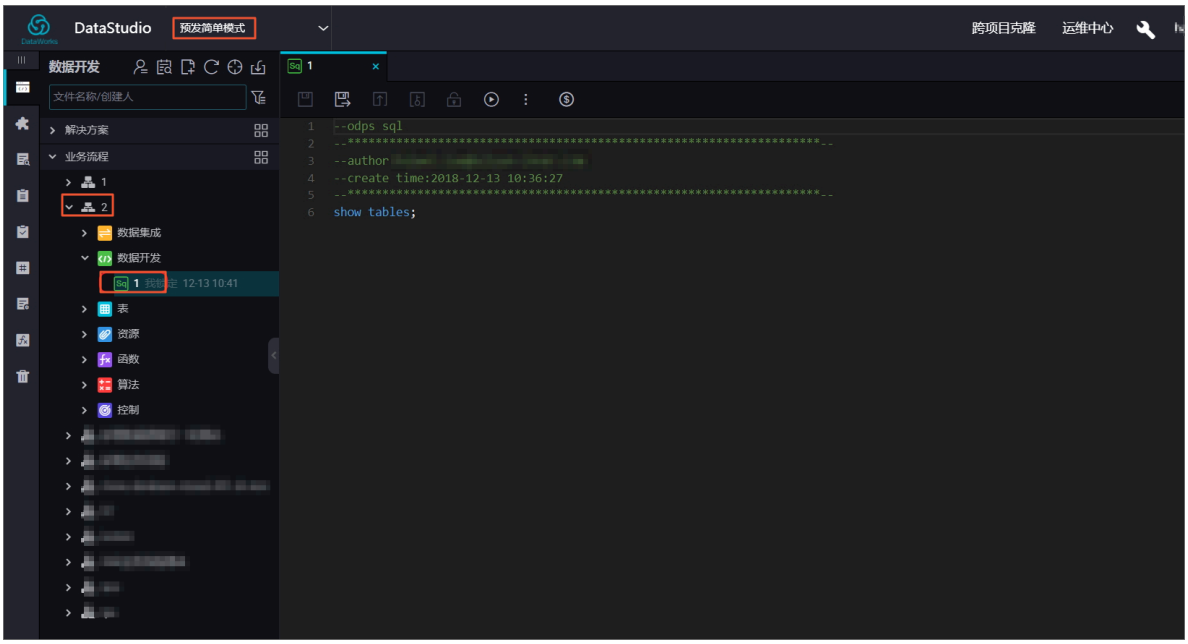
支持的场景

跨项目克隆支持以下两种场景。

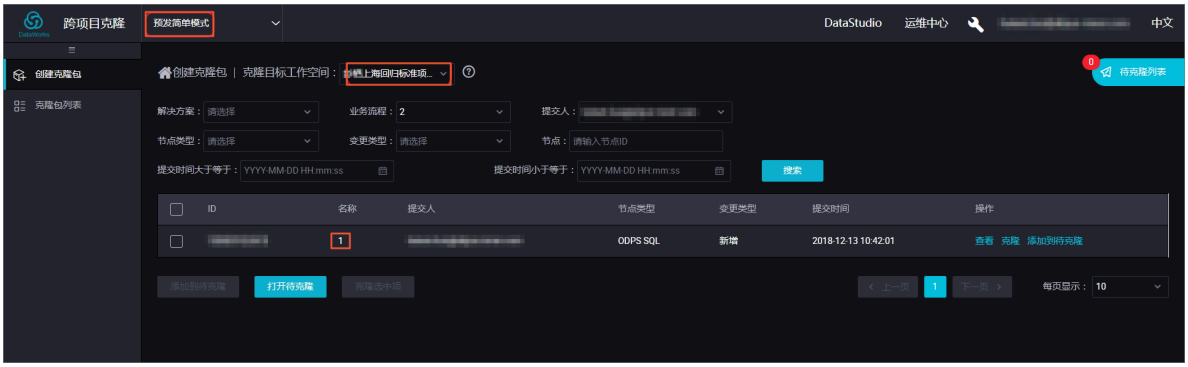
- 从一个简单模式的工作空间克隆到另一个简单模式的工作空间。
- 从一个简单模式的工作空间克隆到另一个标准模式的工作空间。

操作步骤

1. 进入数据开发页面，新建业务流程。

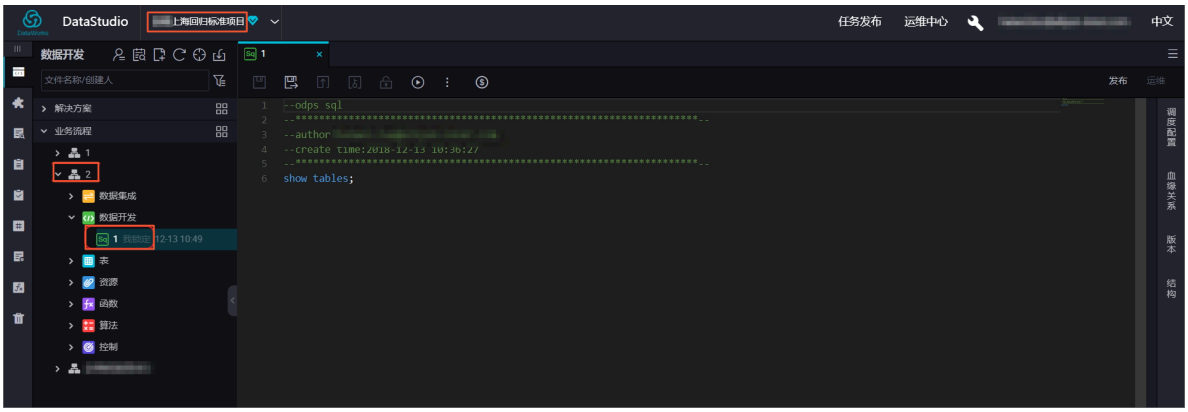


2. 单击右上角的跨项目克隆，跳转至相应的克隆页面，过滤出相应的节点任务，并将任务克隆到目标工作空间。



3. 添加到克隆列表，克隆相关的节点任务。

4. 在目标端工作空间查看相关的克隆结果，通常会克隆业务流程的整体目录结构。



3.9 手动业务流程

3.9.1 手动业务流程介绍

手动业务流程中创建的所有节点都需要手动触发，无法通过调度执行。因此，手动业务流程中的节点不需要配置父节点依赖与本节点的输出。

手动业务流程界面的功能说明如下表所示。

序号	功能	说明
1	提交	提交当前手动业务流程中的所有节点。
2	运行	运行当前手动业务流程下的所有节点，因为手动任务不存在依赖，所以会同时运行。
3	停止运行	停止正在运行的节点。
4	发布	跳转至任务发布页面，可以将当前所有只提交未发布的节点，选择部分或全部发布至生产环境。
5	前往运维	前往运维中心。
6	框选	您可以框选需要的节点组成节点组。
7	刷新	刷新当前手动业务流程界面。
8	自动布局	自动将当前手动业务流程下的节点进行排序。
9	放大	放大界面。
10	缩小	缩小界面。
11	查询	查询当前手动业务流程下的某个节点。
12	全屏	全屏展示当前手动业务流程的节点。

序号	功能	说明
13	流程参数	设置参数，流程参数优先级高于节点参数的优先级。如果参数key与参数对应，会优先执行业务流程设置的参数。
14	操作历史	对此业务流程下所有节点的操作历史。
15	版本	当前手动业务流程下所有节点的提交发布记录。

3.9.2 资源

资源（Resource）是MaxCompute的特有概念，手动业务流程也支持资源的上传和提交。

如果您想使用MaxCompute的自定义函数（UDF）或MaxCompute MR功能，需要依赖资源来完成。

- ODPS SQL UDF：您在编写UDF后，需要将编译好的Jar包上传到ODPS。运行这个UDF时，ODPS会自动下载这个Jar包，获取用户代码，运行UDF。上传Jar包的过程就是在ODPS上创建资源的过程，Jar是ODPS资源的一种。
- ODPS MapReduce：您编写MapReduce程序后，将编译好的Jar包作为一种资源上传到ODPS。运行MapReduce作业时，MapReduce框架会自动下载这个Jar资源，获取用户代码。

您也可以将文本文件、ODPS表以及.zip / .tgz / .tar.gz / .tar / jar等压缩包作为不同类型的资源上传到ODPS，在UDF及MapReduce的运行过程中读取、使用这些资源。

ODPS提供了读取、使用资源的接口。ODPS资源的类型包括：

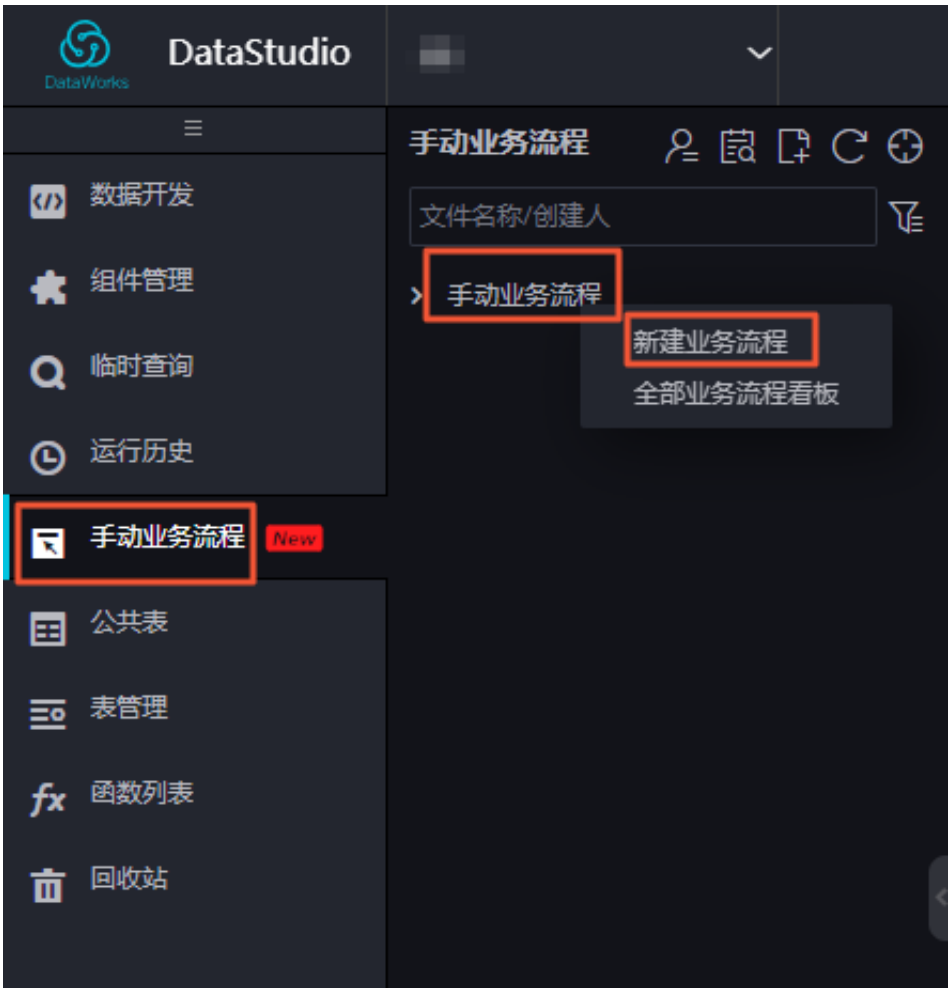
- File类型
- Archive类型：通过资源名称中的后缀识别压缩类型，支持的压缩文件类型包括.zip/.tgz/.tar.gz/.tar/jar。
- Jar类型：编译好的Java Jar包。

DataWorks新建资源就是add resource的过程，当前DataWorks仅支持可视化添加jar、python和file类型的资源。新建入口都一样，区别如下：

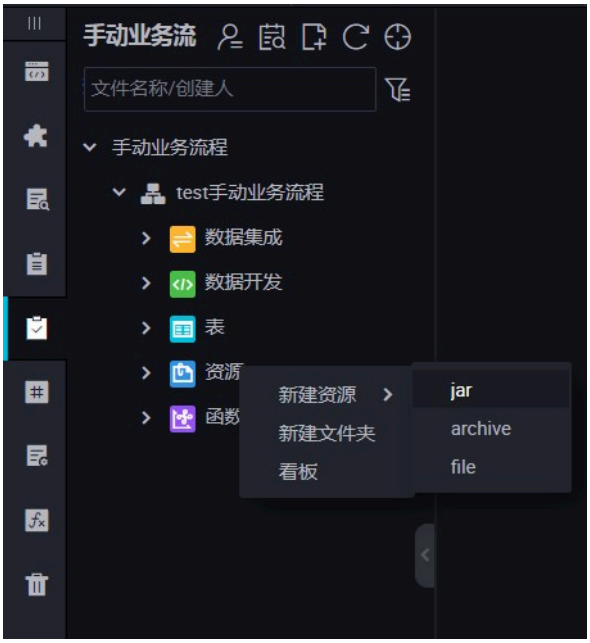
- Jar资源是用户在线下Java环境编辑Java代码，打Jar包上传到Jar资源类型文件。
- 小文件File类型资源是直接在DataWorks上编辑。
- File类型资源新建时勾选大文件后，也可以上传本地资源文件。

新建资源实例

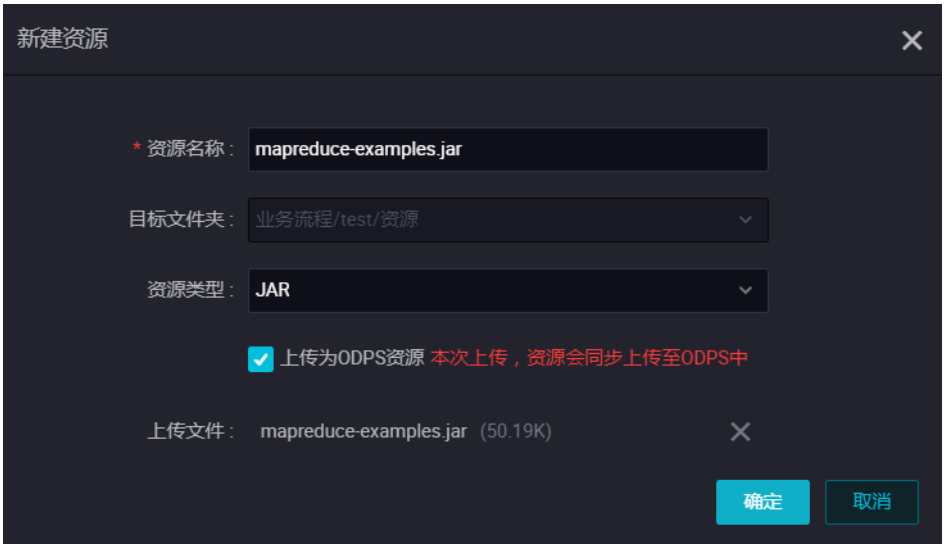
1. 单击左侧导航栏中的手动业务流程，选择新建业务流程。



2. 右键单击资源，选择新建资源 > jar。



3. 按照命名规则在新建资源对话框输入资源名称，并选择资源类型为jar，同时选择需要上传本机的Jar包。



说明:

- 如果此Jar包已经在odps客户端上传过，则需要取消勾选上传为ODPS资源本次上传，资源会同步上传至ODPS中，否则上传会报错。
- 资源名称不一定与上传的文件名一致。
- 资源名命名规范：1到128个字符，字母、数字、下划线、小数点，大小写不敏感，Jar资源时后缀是.jar。

4. 单击提交，将资源提交到调度开发服务器端。

5. 发布节点任务。

具体操作请参见#unique_289。

3.9.3 函数

手动业务流程下，您可以注册您的UDF函数。

注册UDF函数

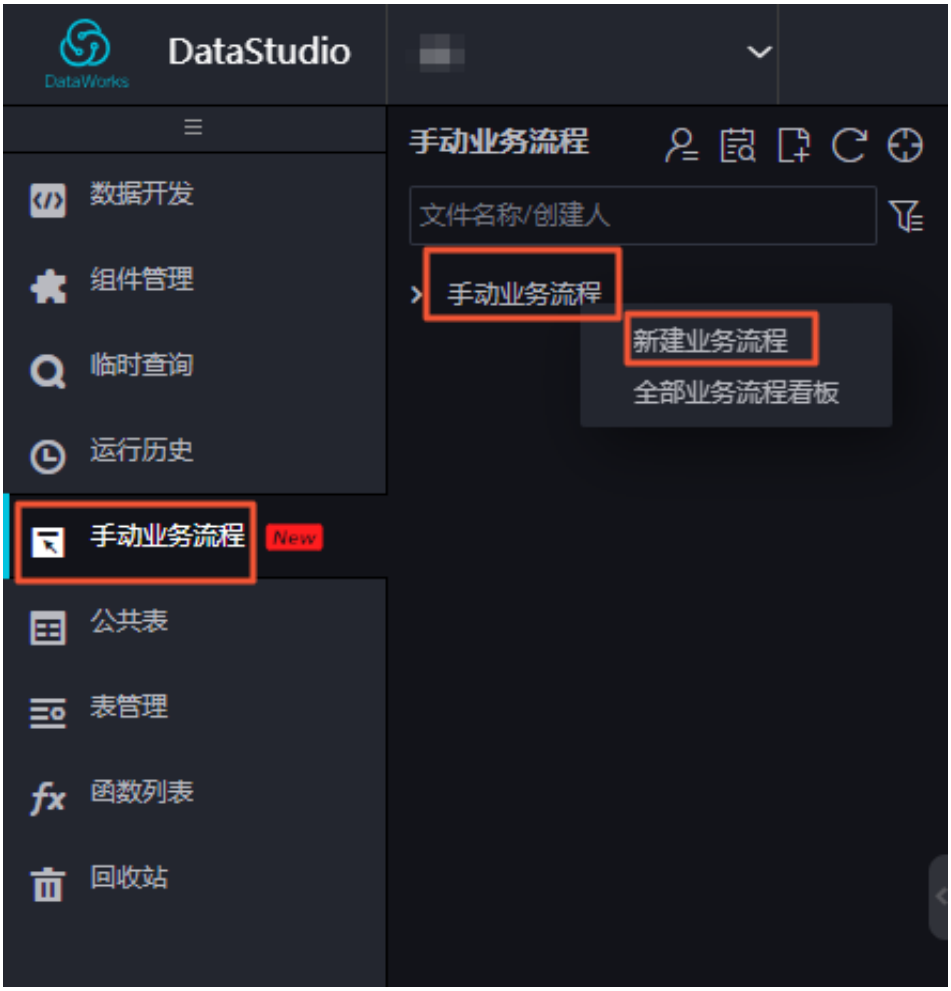
MaxCompute支持自定义UDF，详情请参见UDF概述。

DataWorks上也有对应的可视化界面注册函数替代MaxCompute的`add function`命令。

目前支持Python和Java两种语言接口实现UDF，如果您想编写UDF程序，可以通过添加资源的方式将UDF代码上传，然后再注册函数。

注册步骤

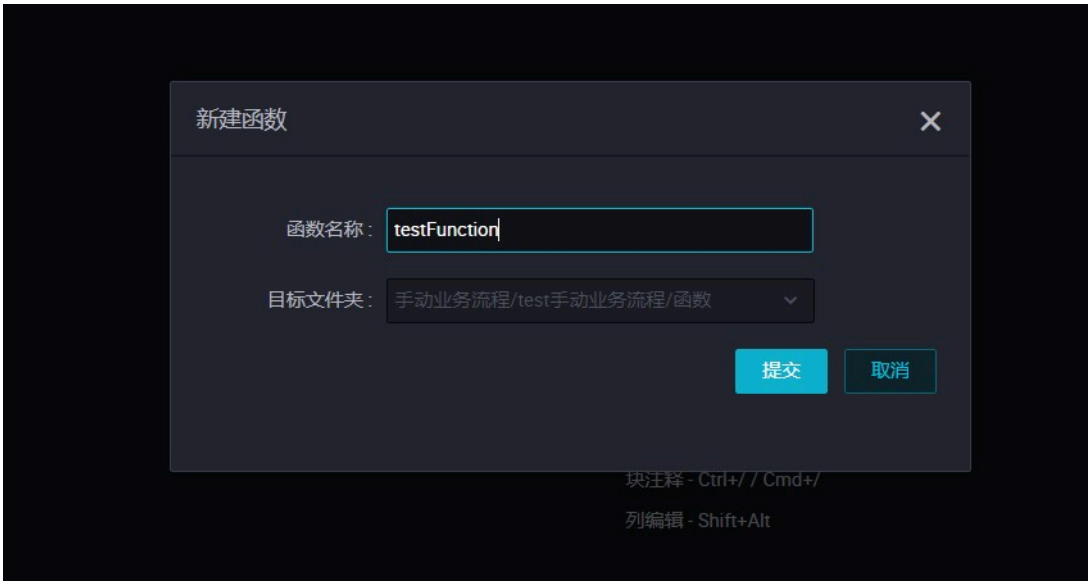
1. 单击左侧导航栏中的手动业务流程，选择新建业务流程。



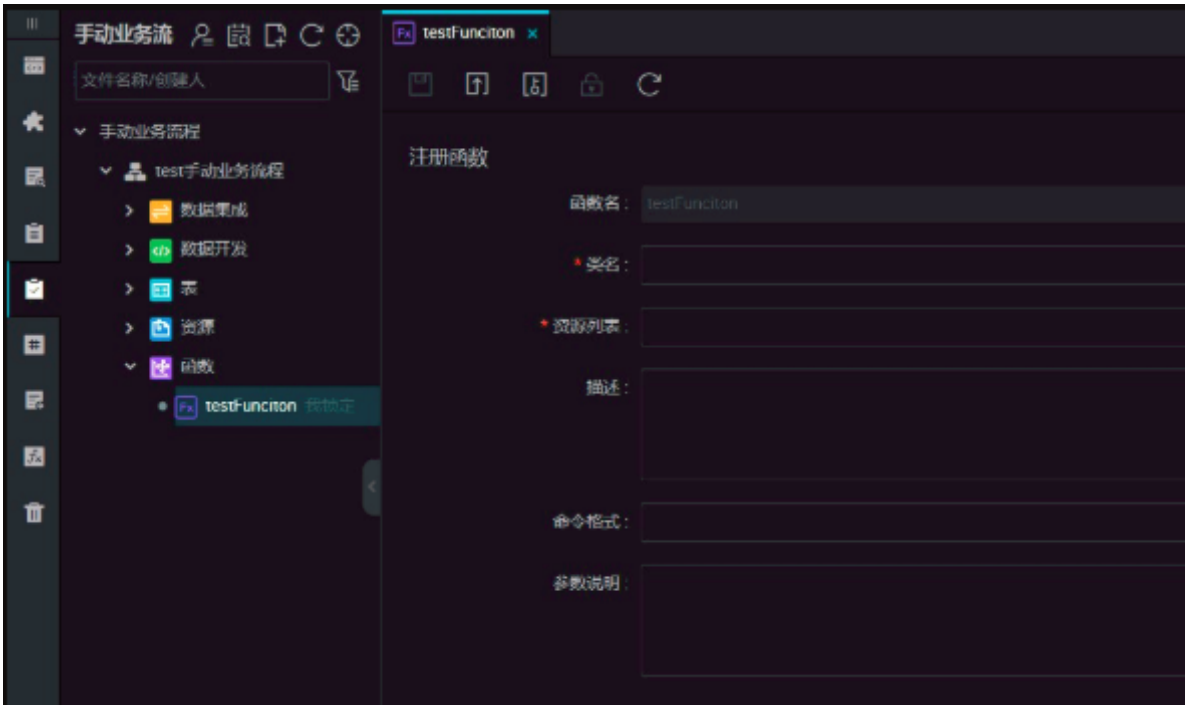
2. 本地Java环境编辑程序打Jar包，新建Jar资源，提交发布。

   您也可以新建Python资源，编写Python代码保存并提交发布。详情请参见新建资源。

3. 选择函数 > 新建函数，输入新建函数的名称，单击提交。



4. 编辑函数配置。



- 类名：实现UDF的主类名，注意当资源类型是python时，类型的写法是python资源名称.类名（资源名中的.py不用写）。
- 资源列表：第二步中的资源名称，多个资源用逗号分隔。
- 描述：UDF描述，非必填项。

5. 提交任务。

完成配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

6. 发布任务。

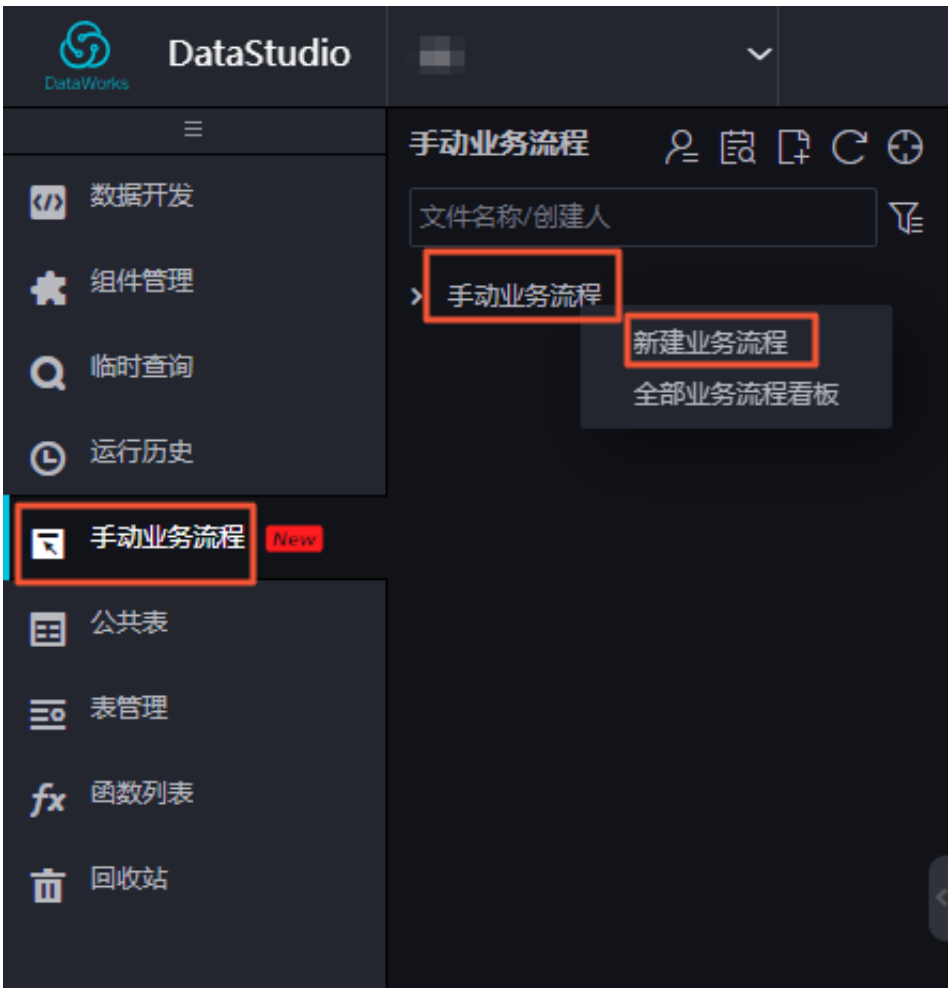
   具体操作请参见#unique_289。

3.9.4 表

手动业务流程支持表的新建、编辑和删除等操作。

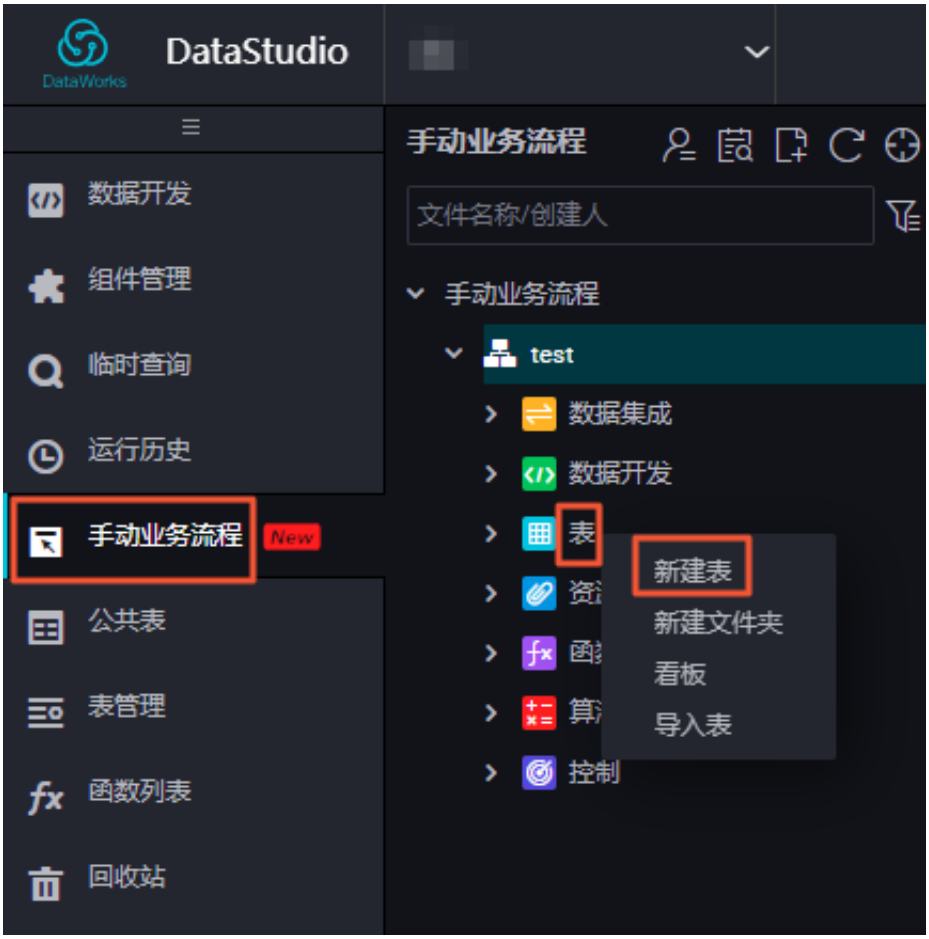
新建表

1. 单击左侧导航栏中的手动业务流程，进入手动业务流程面板。
2. 右键单击手动业务流程，选择新建业务流程。



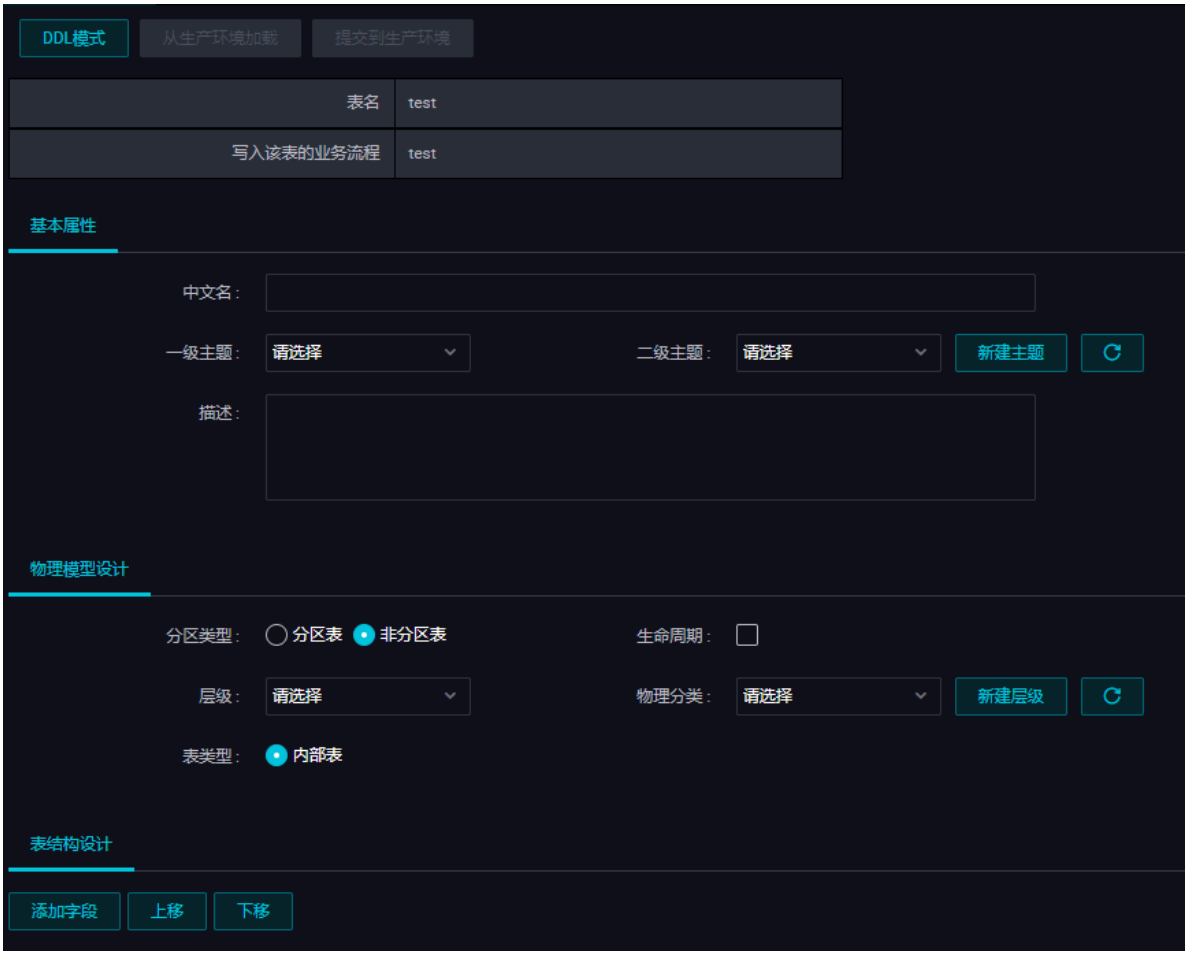
3. 填写业务名称和描述，单击新建，即可完成业务流程的新建。

4. 右键单击相应业务流程下的表，选择新建表。



5. 填写新建表对话框中的表名，单击提交。

6. 设置表的基本属性。



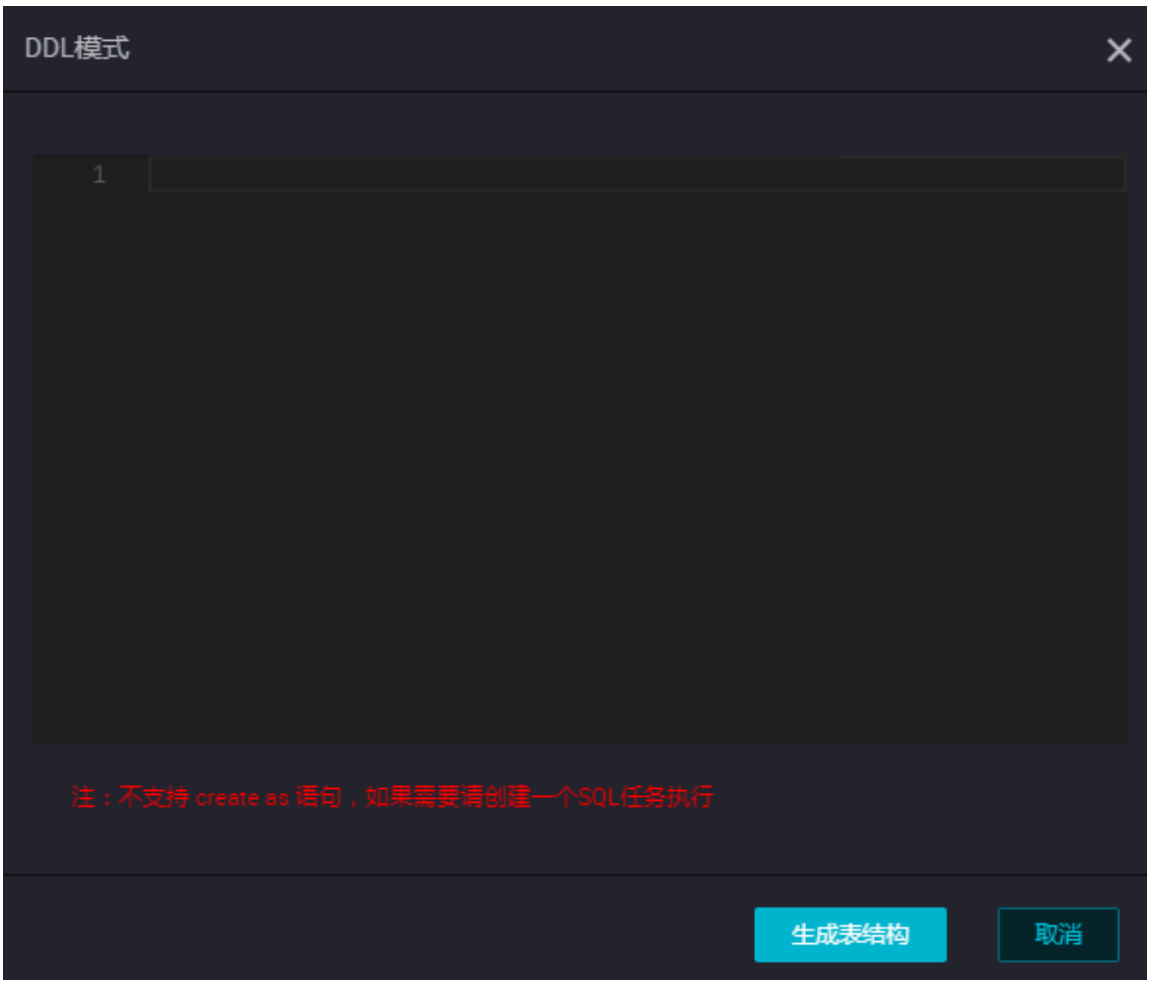
配置	说明
中文名	表的中文名称。
一级主题	新建表所处的一级目标文件夹名称。 **说明：** 一级、二级主题仅仅是DataWorks上文件夹的摆放形式，目的是为了您能更好地管理您的表。
二级主题	新建表所处的二级目标文件夹名称。
新建主题	单击新建主题，跳转至主题管理页面，您可以在该页面创建一级主题、二级主题。 
描述	针对新建表的描述。

7. 创建表。

您可以通过以下两种方式创建表：

- 使用DDL模式创建表。

  单击DDL模式，在对话框中输入标准的建表语句。



  编辑好建表语句后，单击生成表结构，即可自动填充基本属性、物理模型设计、表结构设计中的相关内容。

- 使用图形界面创建表。

  如果不适用于DDL模式建表，您也可以使用图形界面直接建表，相关设置说明如下。

分类	配置	说明
物理模型设计	表类型	包括分区表和非分区表两种类型。
	保存周期	即MaxCompute的生命周期功能。填写一个数字表示天数，该表（或分区）超过一定天数，未更新的数据会被清除。

分类	配置	说明
	层级	通常可以分为DW、ODS和RPT三个层级。 关于物理层级的涉及相关信息请参见#unique_402。
	物理分类	包括基础业务层、高级业务层和其他。 单击新建层级，跳转至层级管理页面，即可在此新增层级。 说明: 物理分类仅为方便您的管理，不涉及底层实现。
表结构设计	字段英文名	字段英文名，由字母、数字和下划线组成。
	中文名	字段的中文名称。
	字段类型	MaxCompute数据类型，仅支持STRING、BIGINT、DOUBLE、DATETIME和BOOLEAN类型，详情请参见数据类型。
	描述	字段的详细描述。
	主键	勾选表示该字段是主键，或者是联合主键的其中一个字段。
	添加字段	新增一列字段。
	删除字段	删除已经创建的字段。 说明: 已经创建的表，删除字段重新提交时，会要求删除当前表，再去建一张同名表，在生产环境中禁止该操作。
	上移	调整未创建的表的字段顺序。如果为已经创建的表调整字段顺序，会要求删除当前已经创建的表，再去建一张同名表，在生产环境中禁止该操作。
	下移	同上移操作。
	添加分区	可以给当前的表新建一个分区。如果为已经创建的表添加分区，会要求删除当前已经创建的表，再去建一张同名表，该操作在生产环境中禁止。
	删除分区	可以删除一个分区。如果删除已创建的表的 分区，会要求删除当前已经创建的表，再去建一张同名表，在生产环境中禁止该操作。

分类	配置	说明
	操作	包括针对新增字段的确认提交和删除，以及更多属性编辑。 更多属性主要是数据质量相关的信息，提供给系统用于生成校验逻辑。 - 允许为零：勾选表示该字段的值允许为零，仅针对BIGINT和DOUBLE类型的字段。 - 允许为负数：勾选表示该字段的值允许为负数，仅针对BIGINT和DOUBLE类型的字段。 - 安全等级：安全等级为0~4，数字越大代表安全要求越高。如果您的安全等级未达到数字要求，则无法访问表格对应字段。 - 单位：指金额单位，元或者分。非金额含义的字段不必填此项。 - Lookup表名/键值：适用于枚举值型的字段（例如会员类型、状态等）。您可以填该字段对应的字典表（即维表）的表名称，例如会员状态对应的字典表名是dim_user_status。 如果您采用的是全局唯一的字典表，此处应填本字段在字典表中对应的key_type键值类型，例如会员状态对应的键值是TAOBAO_USER_STATUS。 - 值域范围：本字段适用的最大值、最小值，仅针对BIGINT和DOUBLE类型的字段。 - 正则校验：本字段使用的正则表达式。例如是手机号码字段，则可以通过正则表达式来约束它的值为11位数字，或其他更严格的约束。 - 最大长度：字段值的最大字符个数，仅针对STRING类型的字段。 - 日期精度：日期值的实际精度，时、日、月等。例如月汇总表中的month_id的精度是月，尽管它存的值是例如2014-08-01（看起来精度是日）。适用于DATETIME类型或以STRING类型存放的日期值。
		- 日期格式：仅适用于以STRING类型存放的日期值。用类似于yyyy-mm-dd hh:mi:ss的方式来描述该字段实际存放的日期值的格式。

分类	配置	说明
分区字段设计 说明：当物理模型设计选择分区表后才显示分区字段设计。	字段类型	建议统一采用STRING类型。
	日期分区格式	如果该分区字段是日期含义（尽管数据类型可能是STRING），则一个或自填一个日期格式，常用格式为yyyymmmdd、yyyy-mm-dd。
	日期分区粒度	支持的分区粒度有秒/分/时/日/月/季度/年。创建分区粒度根据需要可自行填写，如果需要填写多个分区粒度，则默认粒度越大，分区等级越高。例如同时存在日、时、月三个分区，多级分区关系是一级分区（月），二级分区（日），三级分区（时）。

提交表

编辑完表结构信息后，即可将新建表提交到开发环境和生产环境。

配置	说明
从开发环境加载	如果该表已经提交到开发环境之后，该按钮会高亮。单击后，会用开发环境已经创建的表信息覆盖当前的页面信息。
提交到开发环境	首先会检查当前编辑页面的必填项是否已经填写完整，如果有遗漏会告警，且禁止提交。
从生产环境加载	已经提交到生产环境的表的详细信息，会覆盖当前页面。
提交到生产环境	会在生产环境的项目中创建这张表。

3.10 手动任务节点类型

3.10.1 ODPS SQL节点

ODPS SQL采用类似SQL的语法，适用于海量数据（TB级）但实时性要求不高的分布式处理场景。它是OLAP应用，主要面向吞吐量。因为每个作业从前期准备到提交等阶段都需要花费较长时间，因此若要求处理几千至数万笔事务的业务，可以使用ODPS SQL顺利完成。

1. 新建业务流程。

   单击左侧导航栏中的手动业务流程，选择新建业务流程。

2. 新建ODPS SQL节点。

   右键单击数据开发，选择新建数据开发节点 > ODPS SQL。

3. 编辑节点代码。

   编写符合语法的ODPS SQL代码，SQL语法请参见MaxCompute SQL模块。

4. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

5. 发布节点任务。

   具体操作请参见发布管理。

6. 在生产环境测试。

   具体操作请参见#unique_406。

3.10.2 PyODPS节点

DataWorks也推出了PyODPS任务类型，集成了MaxCompute的Python SDK，可在DataWorks的PyODPS节点上直接编辑Python代码操作MaxCompute。

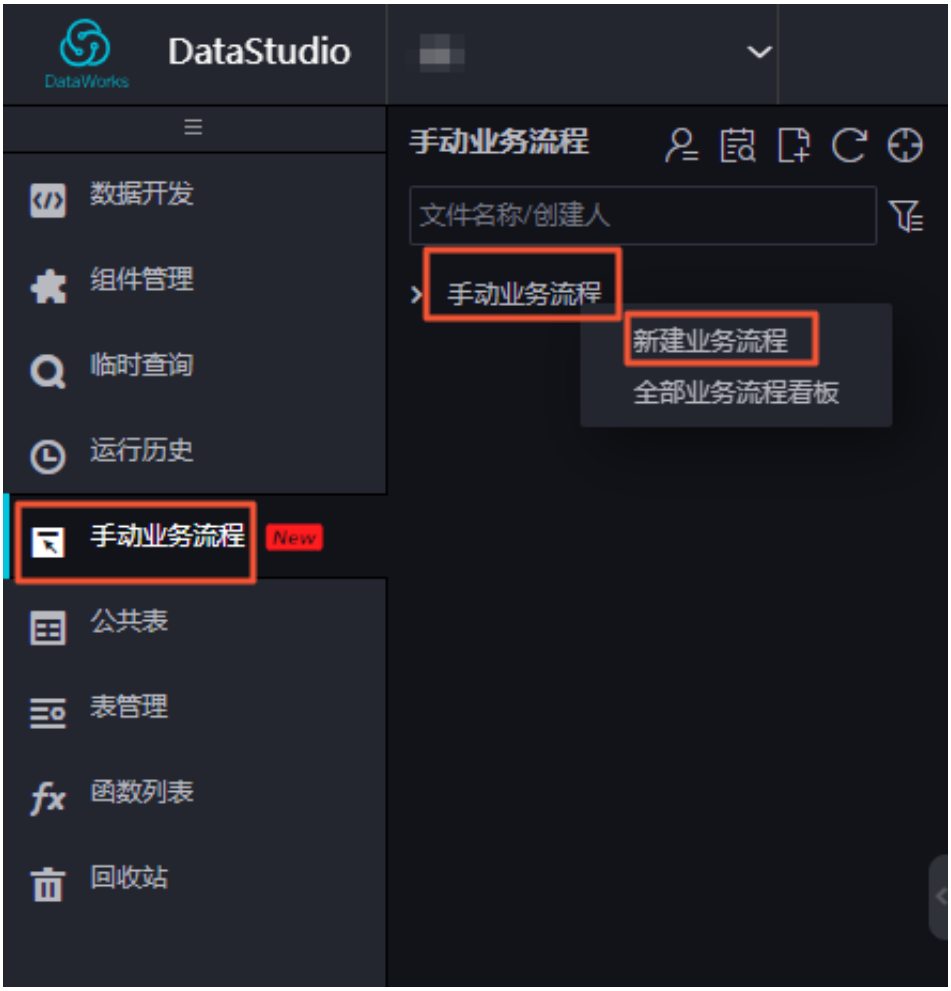
新建PyODPS节点

Maxcompute提供了Python SDK，您可以使用Python的SDK来操作Maxcompute。

新建PyODPS节点的具体操作如下。

1. 新建业务流程。

   单击左侧导航栏中的手动业务流程，选择新建业务流程。



2. 新建PyODPS节点。

   右键单击数据开发，选择新建数据开发节点 > PyODPS。

3. 编辑PyODPS节点。

a. ODPS入口

DataWorks 的PyODPS 节点中，将会包含一个全局的变量'odps'或'o'，即ODPS入口，您不需要手动定义ODPS入口。

```
print(odps.exist_table('PyODPS_iris'))
```

b. 执行SQL

PyODPS支持ODPS SQL的查询，并可以读取执行的结果。execute_sql或run_sql方法的返回值是运行实例。

说明:

并非所有在ODPS Console中可以执行的命令都是ODPS可以接受的SQL语句。在调用非DDL/DML语句时，请使用其他方法，例如GRANT/REVOKE等语句，请使用run_security_query方法，PAI命令请使用run_xflow或execute_xflow方法。

```
o.execute_sql('select * from dual')  #  同步的方式执行，会阻塞直到SQL执行完成
instance = o.run_sql('select * from dual')  # 异步的方式执行
print(instance.get_logview_address())  # 获取logview地址
instance.wait_for_success()  # 阻塞直到完成
```

c. 设置运行参数

您可通过设置hints参数来设置运行时的参数，参数类型是dict。

```
o.execute_sql('select * from PyODPS_iris', hints={'odps.sql.mapper.split.size': 16})
```

对全局配置设置sql.settings后，每次运行时都需要添加相关的运行时参数。

```
from odps import options
options.sql.settings = {'odps.sql.mapper.split.size': 16}
o.execute_sql('select * from PyODPS_iris')  # 会根据全局配置添加hints
```

d. 读取SQL执行结果

运行SQL的instance能够直接执行open_reader的操作，一种情况是SQL返回了结构化的数据。

```
with o.execute_sql('select * from dual').open_reader() as reader:
for record in reader:  # 处理每一个record
```

另一种情况是SQL可能执行的desc等，通过reader.raw属性取到原始的SQL执行结果。

```
with o.execute_sql('desc dual').open_reader() as reader:
```

```
print(reader.raw)
```

> 说明:
> 在数据开发使用了自定义调度参数，页面上直接触发运行PyODPS节点时，需要写死时间，PyODPS节点无法像SQL一样直接替换。

4. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

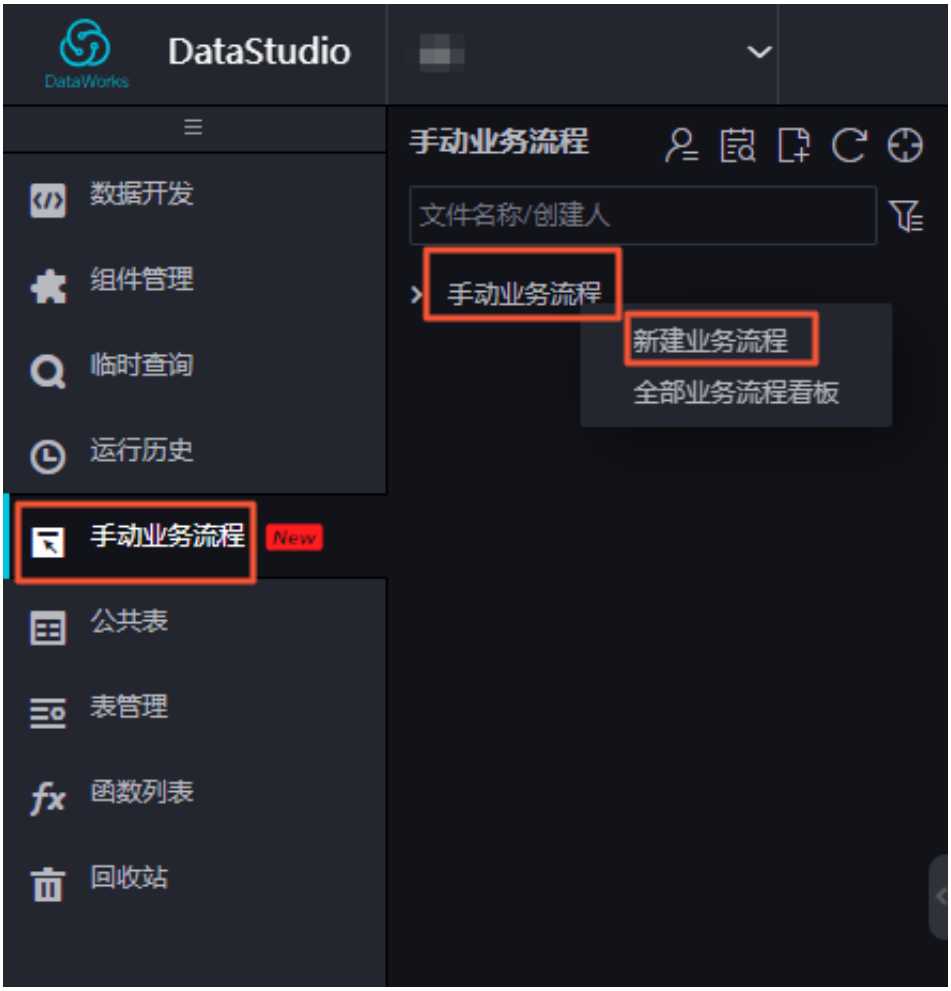
   具体操作请参见#unique_406。

3.10.3 手动数据同步节点

目前数据同步任务支持的数据源类型包括MaxCompute、MySQL、DRDS、SQL Server、PostgreSQL、Oracle、MongoDB、DB2、OTS、OTS Stream、OSS、FTP、Hbase、LogHub、HDFS和Stream，更多支持的数据源请参见#unique_409。

1. **新建业务流程。**

   单击左侧导航栏中的手动业务流程，选择新建业务流程。



2. 新建数据同步节点。

   右键单击数据集成，选择新建数据集成节点 > 数据同步。

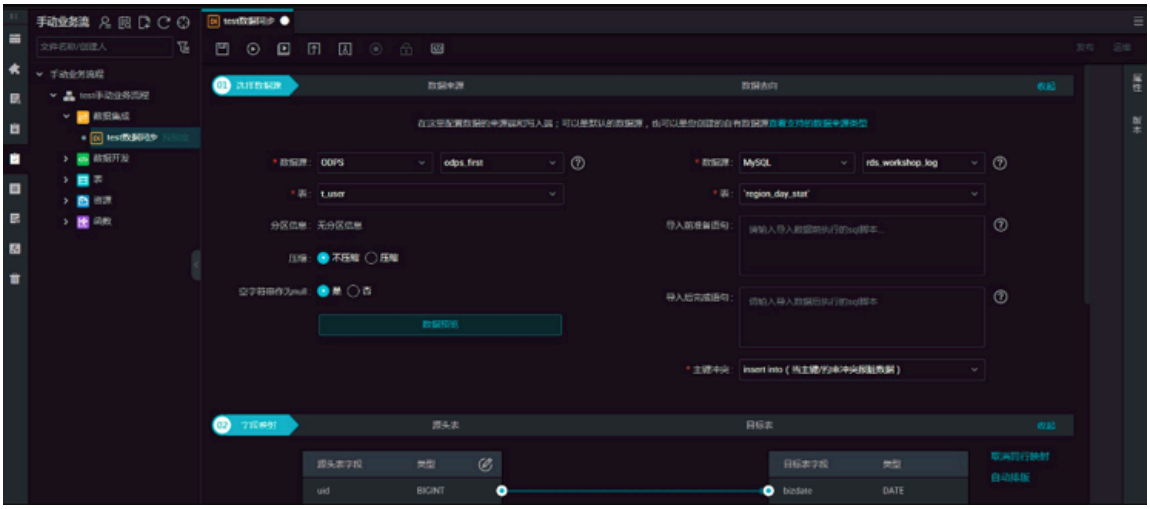
3. 配置同步任务。

   同步中心任务配置非常简单，只需要输入原表名称和目标表名称即可完成一个简单的任务配置。

   当您输入表名时，页面会自动弹所有匹配表名的对象列表（当前只支持精确匹配，所以请输入完整的正确的表名），有些对象是当前同步中心不支持的，会被打上不支持标签。您可以将鼠标移动到列表对象上，页面会自动展示对象的详细信息，例如表所在库、IP、Owner等，这些信息

可以协助你选择正确的表对象。选中后鼠标点击对象，列信息会自动填充。您也可以编辑列，包括移动、删除、添加等操作。

a. 配置同步表。



b. 编辑数据来源。

一般情况下不需要对来源表内容进行编辑，除非您有需要。

- 单击列右侧的插入可以插入新的列。
- 单击列右侧的删除，可以删除列。

c. 编辑数据去向。

一般情况下不需要对去向表的字段信息进行编辑，除非您有需要，例如只需要导入部分列的数据。

> 说明:
> 目的端是ODPS表时，不支持删除列同步中心的配置当中，源头表和目的表的字段配置是按照配置页面的顺序一一匹配的，而不是按照字段名称。

d. 增量同步与全量同步。

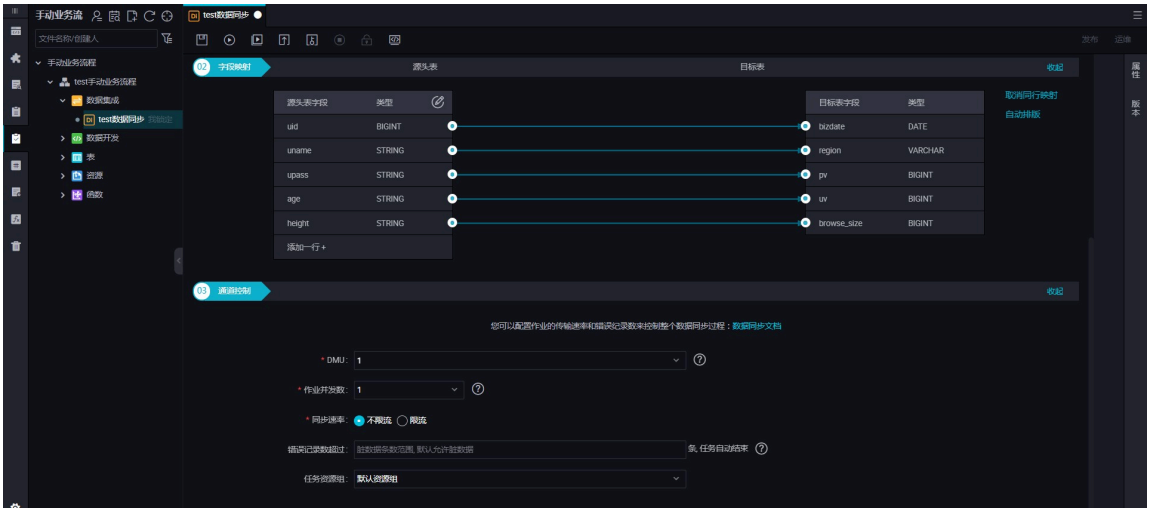
- 增量同步分区格式：ds=${bizdate}
- 全量同步分区格式：ds=*

> 说明:
> 如果需要同步多个分区，同步中心支持简单的正则表达式。
>
> - 例如需要同步多个分区，但是正则又不好写，可以选择这种方式：`ds=20180312 | ds=20180313 | ds=20180314;`

- 需要同步一个区间内的分区，同步中心扩展了一种语法，类似/*query*/ds>=20180313 and ds<20180315;这种方式，一定要加上/query/。
- 变量bizdate必须在下面的参数中做定义-p"-Dbizdate=$bizdate -Denv_path=$env_path -Dhour=$hour"。如果您需要自定义变量，如pt=${selfVar}，则对应也需要在参数中定义，如-p"-Dbizdate=$bizdate -Denv_path=$env_path -Dhour=$hour -DselfVar=xxxx。

e. 字段映射。

根据源表和宿表字段位置对应，与字段名称、字段类型无关。



说明:

如果源表为ODPS表时，无法在数据同步时新增字段，非ODPS表可以在据同步时添加字段。

f. 通道控制。

配置	说明
任务期望最大并发数	数据同步任务内，可以从源并行读取或并行写入数据存储端的最大线程数。向导模式通过界面化配置并发数，指定任务所使用的并行度。

配置	说明
同步速率	设置同步速率可以保护读取端数据库，以避免抽取速度过大，给源库造成太大的压力。同步速率建议限流，结合源库的配置，请合理配置抽取速率。
错误记录数	错误记录数，表示脏数据的最大容忍条数。
任务资源组	任务运行的机器，如果任务数比较多，使用默认资源组出现等待资源的情况，建议购买独享数据集成资源或添加自定义资源组，详情请参见#unique_155和#unique_33。

4. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

   具体操作请参见#unique_314。

3.10.4 ODPS MR节点

MaxCompute提供MapReduce编程接口，您可以使用MapReduce提供的接口（Java API）编写MapReduce程序处理MaxCompute中的数据，您可以通过创建ODPS_MR类型节点的方式在任务调度中使用。

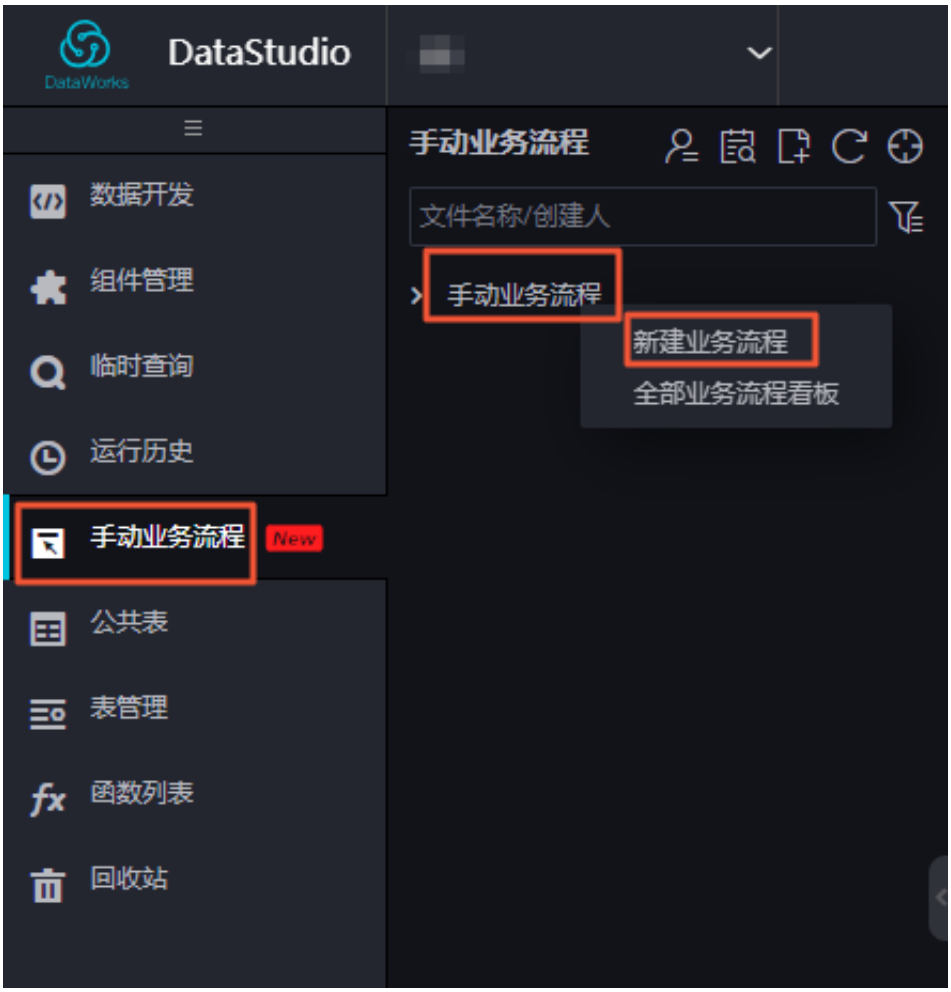
ODPS_MR类型节点的编辑和使用方法，请参见MaxCompute文档示例WordCount示例。

请将需要用到的资源上传并提交发布后，再建立ODPS MR节点。

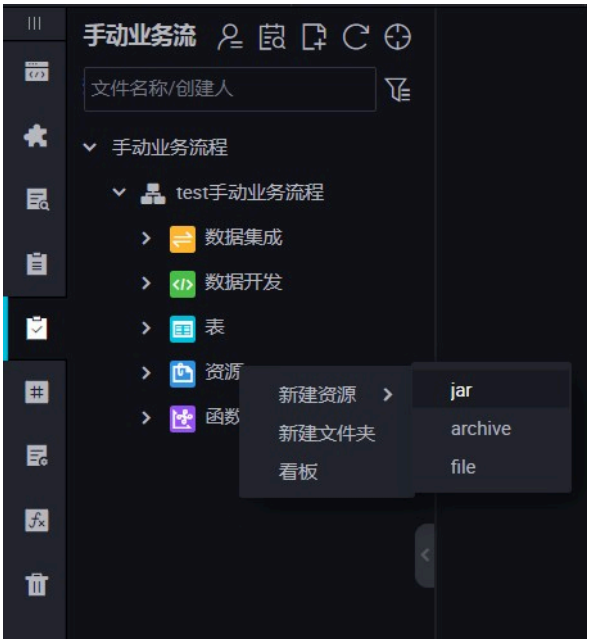
新建资源实例

1. 新建业务流程。

   单击左侧导航栏中的手动业务流程，选择新建业务流程。



2. 右键单击资源，选择新建资源 > jar。



3. 按照命名规则在新建资源对话框输入资源名称，并选择资源类型为jar，同时选择需要上传本机的Jar包。

说明：

- 如果此Jar包已经在odps客户端上传过，则需要取消勾选上传为ODPS资源本次上传，资源会同步上传至ODPS中，否则上传会报错。
- 资源名称不一定与上传的文件名一致。
- 资源名命名规范：1到128个字符，字母、数字、下划线、小数点，大小写不敏感，Jar资源时后缀是.jar，Python资源时后缀为.py。

4. 单击提交，将资源提交到调度开发服务器端。

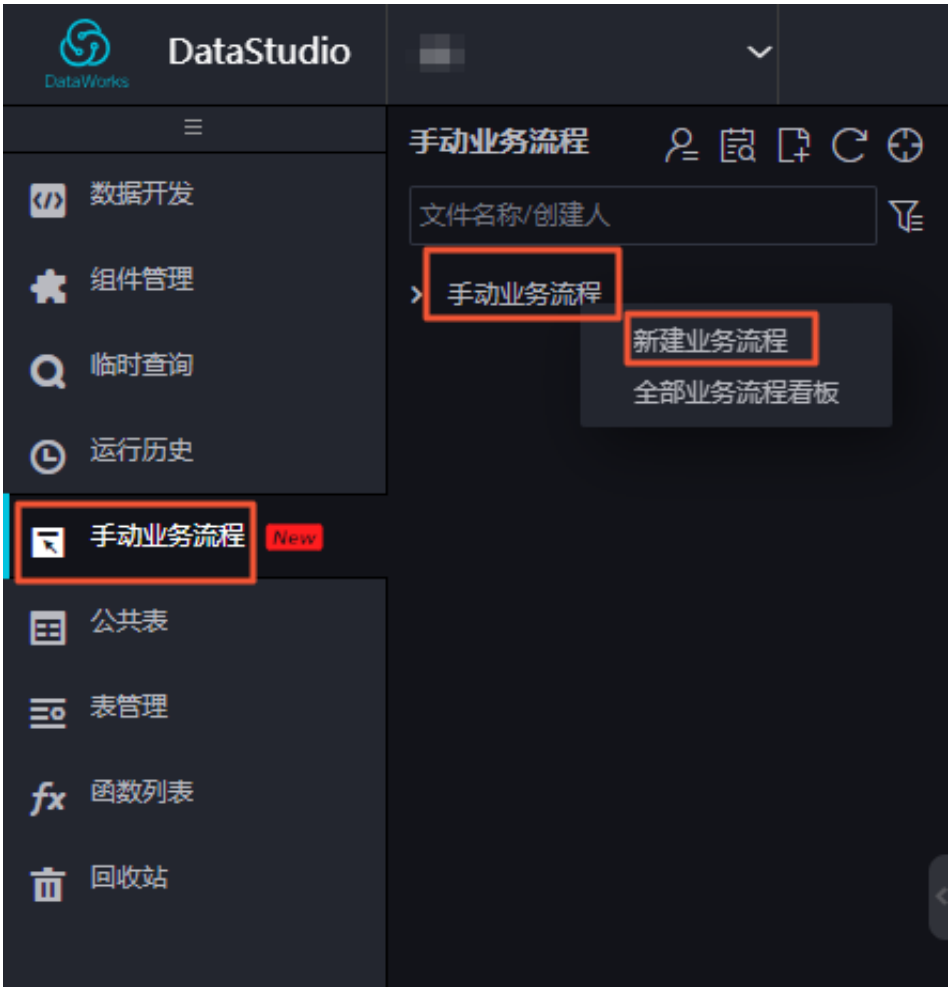
5. 发布节点任务。

   具体操作请参见发布管理。

新建ODPS_MR节点

1. 新建业务流程。

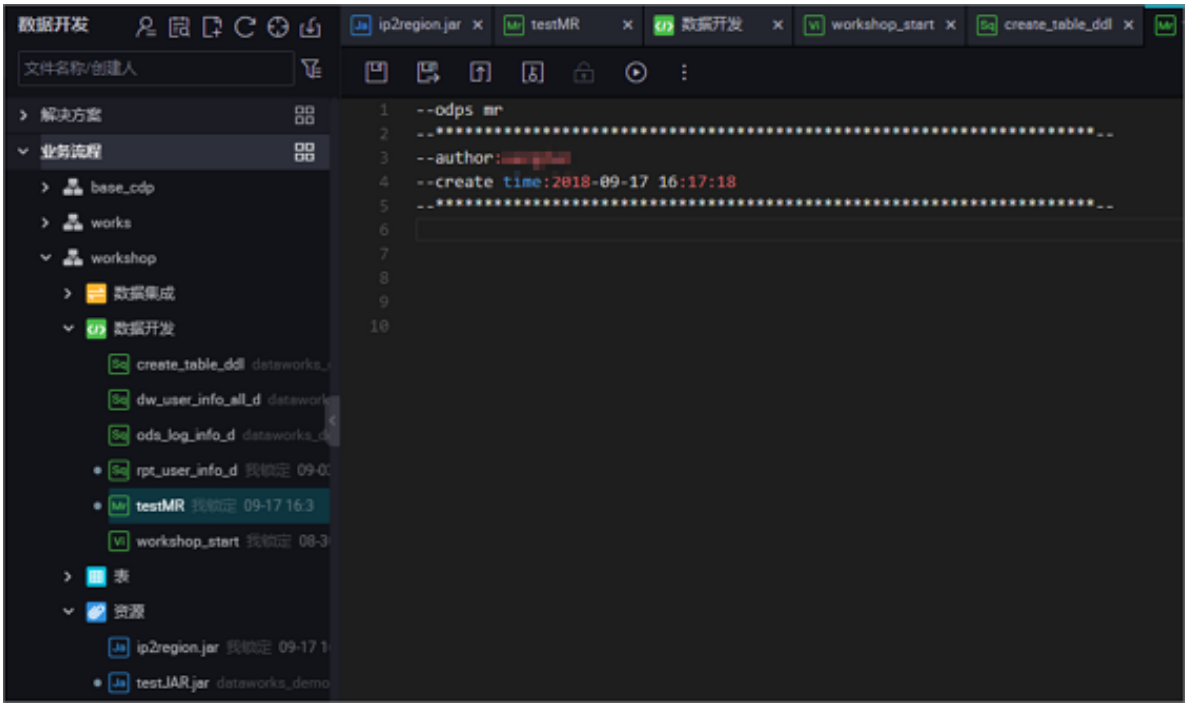
   单击左侧导航栏中的手动业务流程，选择新建业务流程。



2. 新建ODPS MR节点。

右键单击数据开发，选择新建数据开发节点 > ODPS MR。

3. 编辑节点代码。双击新建的ODPS MR节点，进入如下界面：



编辑节点代码示例：

```
jar -resources base_test.jar -classpath ./base_test.jar com.taobao.
edp.odps.brandnormalize.Word.NormalizeWordAll
```

代码说明如下：

- `-resources base_test.jar`：引用到的Jar资源文件名。
- `-classpath`：Jar包路径，可通过对资源文件右键引用资源获得该地址。

  > **说明：**
  > 双击新建的ODPS MR节点，进入ODPS MR节点界面之后引用jar资源。

- `com.taobao.edp.odps.brandnormalize.Word.NormalizeWordAll`：执行过程调用Jar中的主类，需与Jar中的主类名称保持一致。

一个MR调用多个Jar资源时，classpath写法为`-classpath ./xxxx1.jar,./xxxx2.jar`，即两个路径之间用英文逗号分隔。

4. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

5. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

6. 发布节点任务。

   具体操作请参见发布管理。

7. 在生产环境测试。

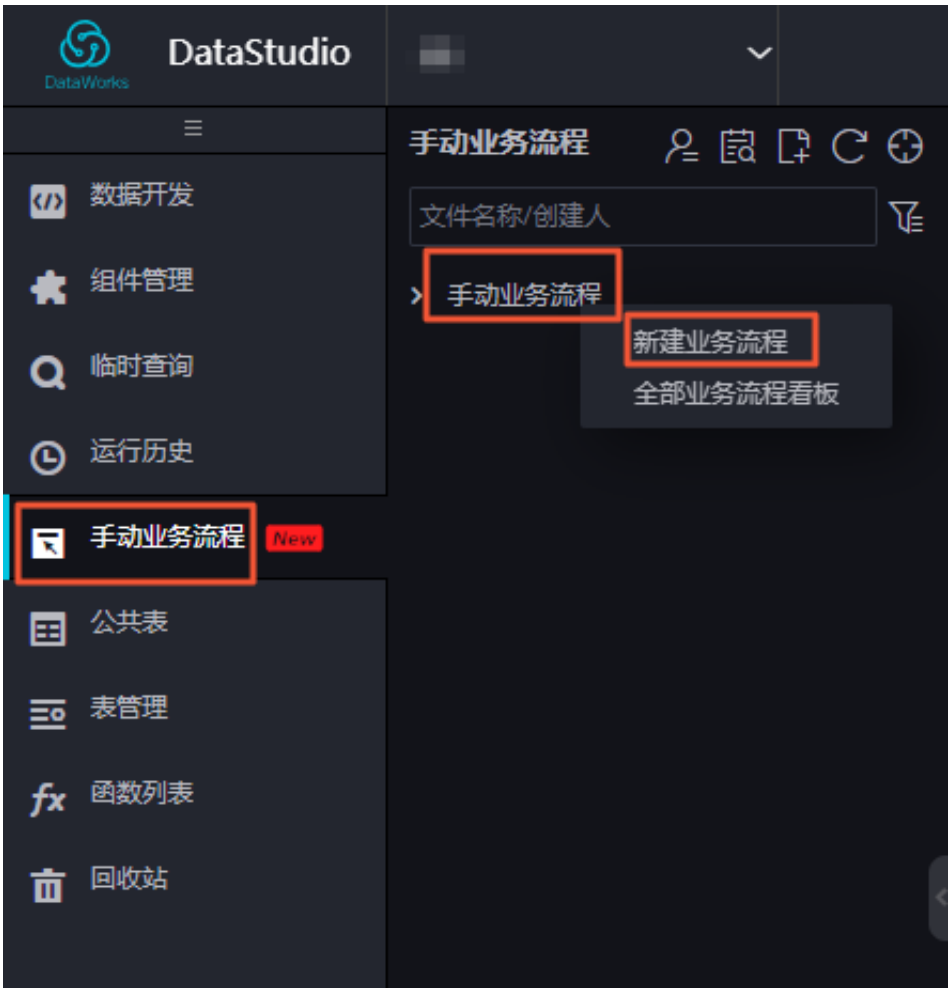
   具体操作请参见#unique_406。

3.10.5 SQL组件节点

本文将为您介绍如何在手动业务流程下，新建和配置SQL组件节点。

操作步骤

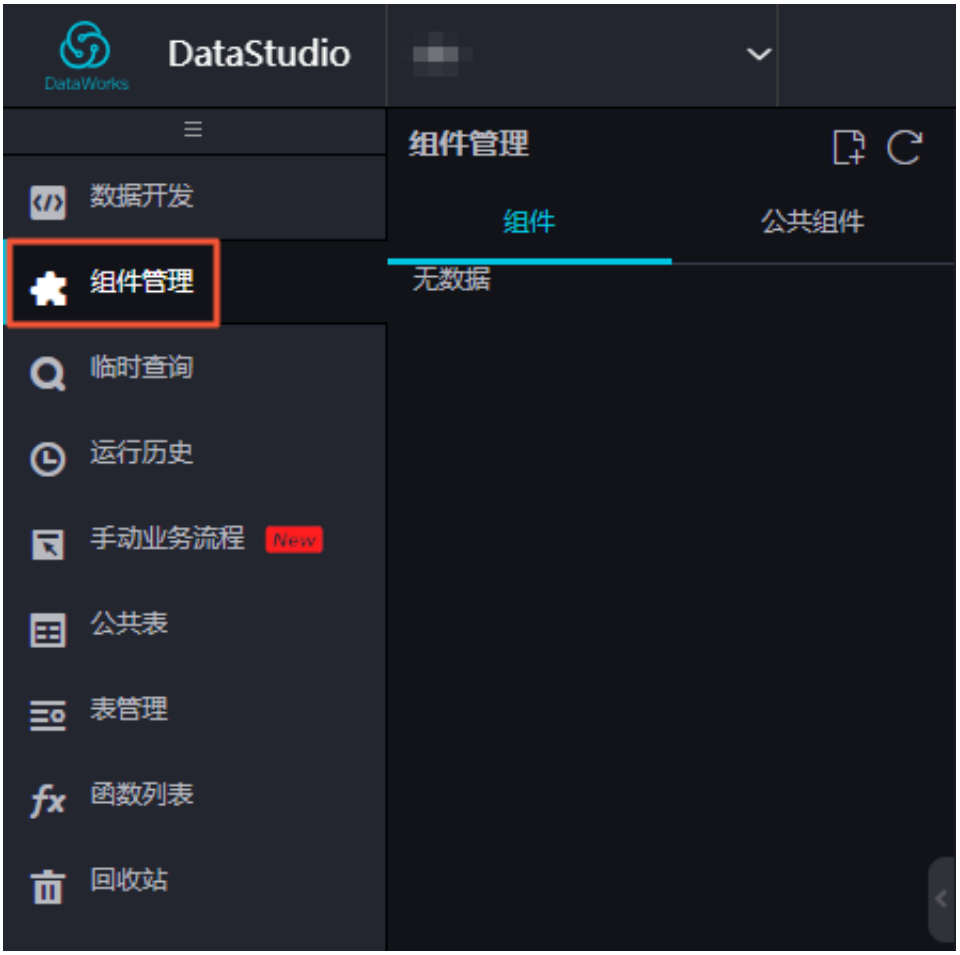
1. 单击左侧导航栏中的手动业务流程，进入手动业务流程面板。
2. 右键单击手动业务流程，选择新建业务流程。



3. 填写业务名称和描述，单击新建，即可完成业务流程的新建。
4. 打开新建的手动业务流程，右键单击数据开发，选择新建数据开发节点 > SQL组件节点。

5. 填写新建节点对话框中的配置，单击提交。

为提高开发效率，数据任务的开发者可以使用项目成员和租户成员贡献的组件，来新建数据处理节点。



- 本项目成员创建的组件在组件下。
- 租户成员创建的组件在公共组件下。

为选定的组件指定参数。

6. 节点调度配置。

   单击节点编辑区右侧的调度配置，即可进入调度配置页面，详情请参见调度配置模块。

7. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

8. 发布节点任务。

   具体操作请参见发布管理。

9. 在生产环境测试。

   具体操作请参见#unique_406。

升级SQL组件节点的版本

在组件的开发者发布新版本后，组件的使用者可以选择是否升级现有组件的使用实例至最新版本。

组件的版本机制支持开发者对组件不断进行升级，提高流程的执行效率并优化业务效果，示例如下。

当A用户使用了B用户的V1.0版本的组件，这时组件所有人B将组件升级到V2.0版本，A用户依旧可以使用V1.0版本，但会看到版本更新的提醒。A用户对比新旧代码后，可以自行决定是否升级到最新的组件版本。

SQL组件节点根据组件模板进行开发，升级过程较为简单。您只需选择升级，确认新版本SQL组件节点的参数配置是否有效，然后根据组件新版本的说明进行调整后，便可保存并提交，进入发布流程。

界面功能点说明

序号	界面功能	说明
1	保存	保存当前组件的设置。

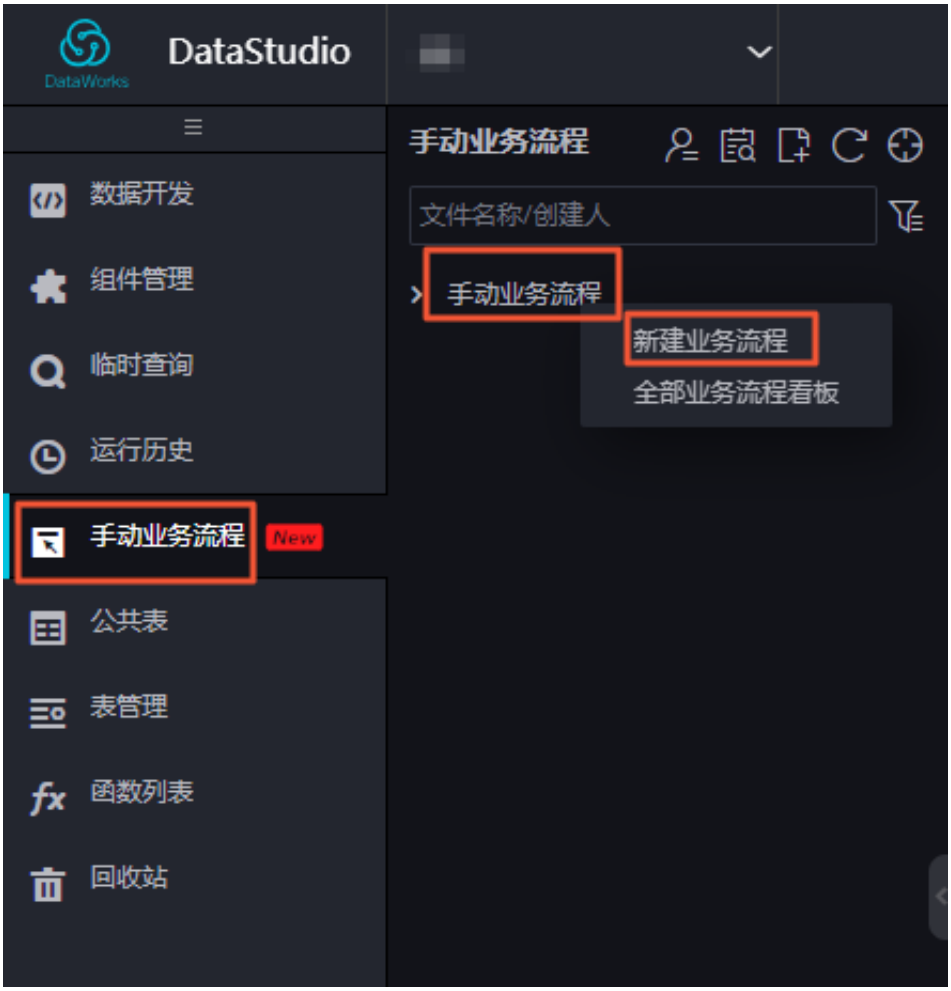
序号	界面功能	说明
2	提交	将当前组件提交到开发环境。
3	提交并解锁	提交当前节点，并解锁进入编辑模式。
4	偷锁编辑	非组件责任人可以偷锁编辑此节点。
5	运行	在本地（开发环境）运行组件。
6	高级运行（带参数运行）	如果代码中有参数，带参数运行代码。 说明: Shell节点不存在高级运行。
7	停止运行	停止运行的组件。
8	重新加载	刷新页面，重新加载后恢复到上一次保存的状态，未保存的内容将丢失。 说明: 如果在配置中心已打开缓存，刷新后会提示缓存了未保存的代码，请选择您需要的版本。
9	在开发环境执行冒烟测试	在开发环境测试当前节点的代码。开发环境冒烟测试可以模拟右侧的调度参数，选择业务日期后，根据您填写的调度参数替换该业务日期下的值。您可以通过该功能测试调度参数的替换情况。 说明: 开发环境冒烟测试每次变更调度属性后，其中的参数配置需要重新保存并提交，然后选择开发环境冒烟测试，否则替换的调度属性还是原来的值。
10	查看开发环境的冒烟测试日志	查看运行在开发环境的节点运行日志。
11	前往开发环境的调度系统	前往开发环境的运维中心。
12	参数配置	组件信息、输入参数、输出参数配置。
13	属性	设置节点的责任人、节点描述、节点参数及设置资源组。
14	血缘关系	查看SQL组件节点依赖血缘关系图以及内部血缘图。
15	版本	当前组件提交发布的记录。

3.10.6 虚拟节点

虚拟节点属于控制类型节点，它是不产生任何数据的空跑节点，通常用于工作流统筹节点的根节点。

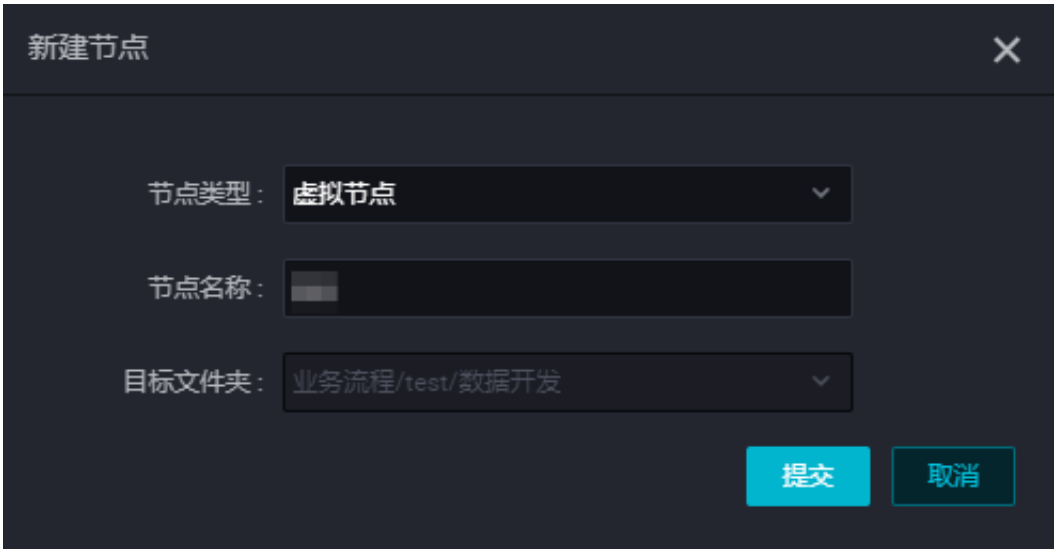
新建虚节点任务

1. 单击左侧导航栏中的手动业务流程，进入手动业务流程面板。
2. 右键单击手动业务流程，选择新建业务流程。



3. 填写业务名称和描述，单击新建，即可完成业务流程的新建。
4. 打开新建的手动业务流程，右键单击数据开发，选择新建数据开发节点 > 虚拟节点。

5. 填写新建节点对话框中的配置，单击提交。



6. 节点调度配置。

   单击节点编辑区右侧的调度配置，即可进入调度配置页面，详情请参见调度配置模块。

7. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

8. 发布节点任务。

   具体操作请参见发布管理。

9. 在生产环境测试。

   具体操作请参见#unique_406。

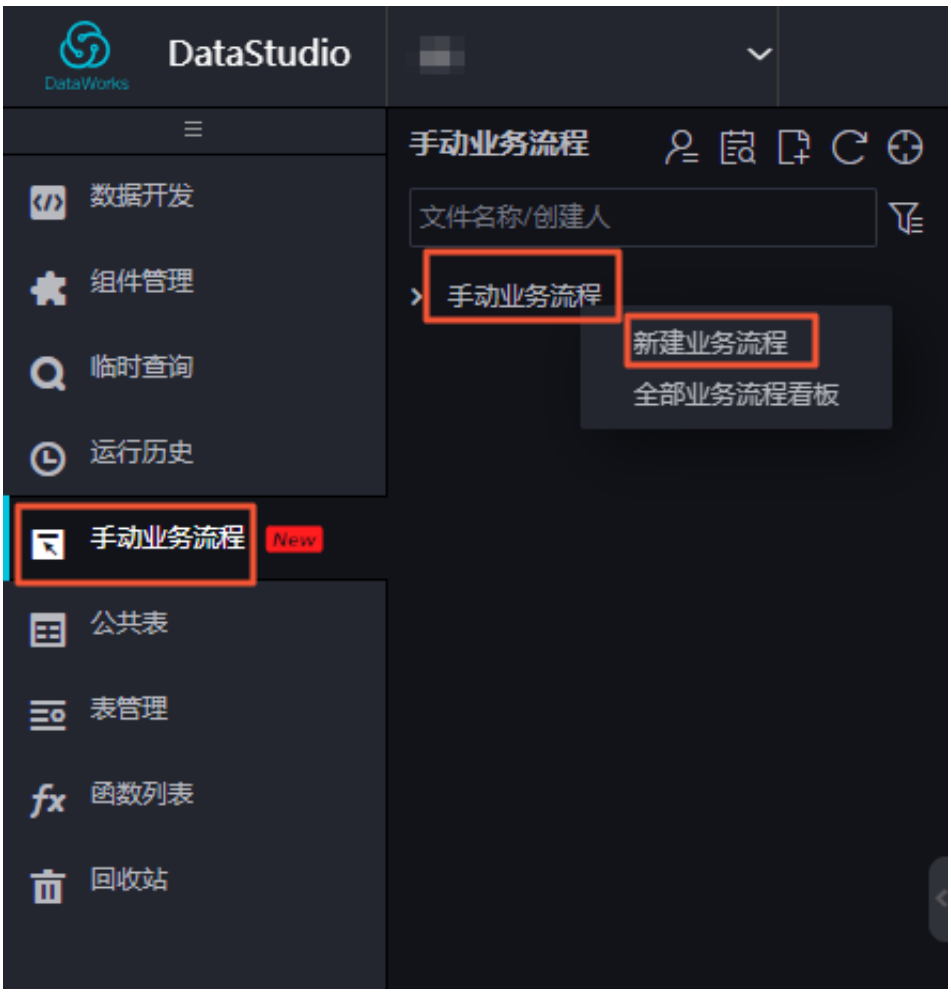
3.10.7 SHELL任务

SHELL任务支持标准SHELL语法，不支持交互性语法。SHELL任务可以在默认资源组上运行，若需要访问IP/域名，请在项目配置页面下将IP/域名添加到白名单中。

操作步骤

1. 新建业务流程。

   单击左侧导航栏中的手动业务流程，选择新建业务流程。

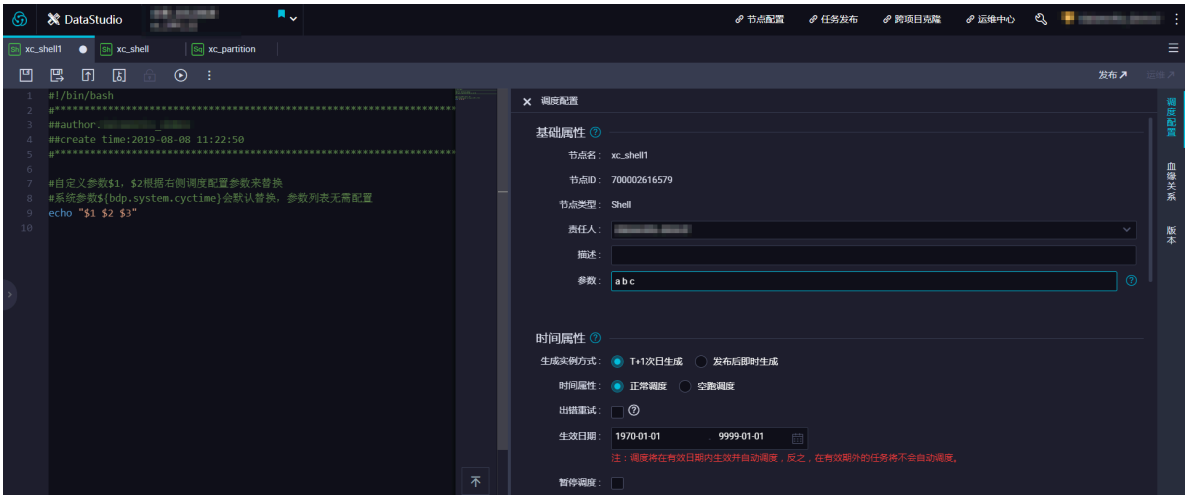


2. 新建SHELL节点。

   右键单击数据开发，选择新建数据开发节点 > SHELL。

3. 选择节点类型为SHELL，命名节点名称并选择目标文件夹，单击提交。
4. 编辑节点代码。

   进入SHELL节点代码编辑页面编辑代码。



   如果想在SHELL中调用系统调度参数，SHELL语句如下所示：

```
echo "$1 $2 $3"
```

   说明：

   参数1 参数2…多个参数之间用空格分隔。更多系统调度参数的使用，请参见#unique_39。

5. 节点调度配置。

   单击节点任务编辑在区域右侧的调度配置，即可进入节点调度配置页面，详情请参见调度配置模块。

6. 提交节点任务。

   完成调度配置后，单击左上角的保存，提交（提交并解锁）到开发环境。

7. 发布节点任务。

   具体操作请参见发布管理。

8. 在生产环境测试。

   具体操作请参见#unique_406。

应用场景

通过SHELL连接数据库

- 如果数据库是在阿里云上搭建的，且区域是华东2，需要将数据库对如下白名单开放，即可连接数据库。

```
10.152.69.0/24,10.153.136.0/24,10.143.32.0/24,120.27.160.26,10.46.67
.156,120.27.160.81,10.46.64.81,121.43.110.160,10.117.39.238,121.43.
112.137,10.117.28.203,118.178.84.74,10.27.63.41,118.178.56.228,10.27
.63.60,118.178.59.233,10.27.63.38,118.178.142.154,10.27.63.15,100.64
.0.0/8
```

  说明:
  如果是在阿里云上搭建的数据库，但区域不是华东2，则建议使用外网或购买与数据库同区域的ECS作为调度资源，将该SHELL任务运行在自定义资源组上。

- 如果数据库是自己在本地搭建的，建议使用外网连接，且将数据库对上述白名单IP开放。

  说明:
  如果使用自定义资源组运行SHELL任务，必须把自定义资源组的机器IP也加到上述白名单中。

3.11 手动任务参数设置

3.11.1 基础属性

基础属性用于配置任务运行的基础参数。

打开节点任务，单击页面右侧的属性，即可进行基础属性的配置。

配置	说明
节点名	新建节点时填写的节点名，您可以在目录树右键单击节点，选择重命名进行修改。
节点ID	任务提交后会生成唯一的节点ID，不可修改。
节点类型	新建节点时选择的节点类型，不可修改。
责任人	新建的节点责任人默认是当前登录的用户，您可以修改责任人。 说明: 只能选择当前工作空间下的成员为责任人。
描述	通常用于描述节点业务、用途。
参数	任务调度时，给代码中的变量赋值。

各种节点类型参数赋值格式

- ODPS SQL、ODPS MR类型：参数赋值格式为`变量名1=参数1 变量名2=参数2`，多个参数之间用空格分隔。
- Shell类型：参数赋值格式为`参数1 参数2`，多个参数之间用空格分隔。

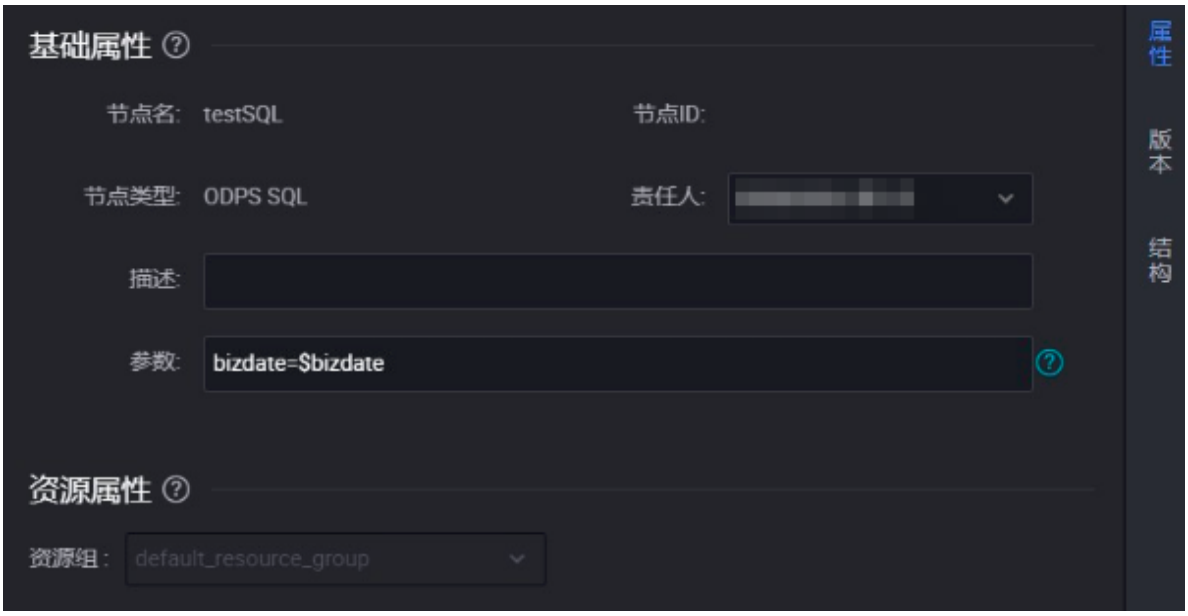
调度内置了一些常用的时间参数，具体参数说明请参见#unique_39。

3.11.2 配置手动节点参数

为使任务自动周期运行时能动态适配环境变化，DataWorks提供手动节点参数配置的功能。

配置参数前，您需要注意以下问题：

- 参数=两边不可以加空格。正确写法示例：`bizdate=$bizdate`。



- 如果存在多个参数，每个参数用空格分开。

系统参数

DataWorks提供了两个系统参数，定义如下：

- `${bdp.system.cyctime}`：定义为一个实例的定时运行时间，默认格式为`yyyymmddhh24miss`。
- `${bdp.system.bizdate}`：定义为一个实例计算时对应的业务日期，业务日期默认为运行日期的前一天，默认以`yyyymmdd`的格式显示。

从定义可知，运行时间和业务日期的计算公式为运行时间=业务日期+1。

如果使用系统参数，无需在编辑框设置，直接在代码中引用${bizdate}即可，系统将自动替换代码中对这个参数的引用字段。

> 说明:
> 一个周期任务的调度属性，配置的是运行时间的定时规律，因此可以根据实例的定时运行时间反推业务日期，从而得知每个实例中参数的取值。

入门示例

设置一个ODPS_SQL任务为小时调度，每天00:00-23:59时间段里每隔1小时执行一次。如果想在代码中使用系统参数，可以执行下述语句。

```
insert overwrite table tb1 partition(ds ='20180606') select
c1,c2,c3
from (
select * from tb2
where ds ='${bizdate}');
```

非Shell节点配置调度参数

> 说明:
> SQL代码中的变量名命名只支持英文的a-z、A-Z、数字和下划线。变量名为date固定会自动赋$bizdate值，不需要在调度参数配置那里赋值，即便赋值了也不会替换到代码中，代码默认替换的还是$bizdate。

在非Shell节点中配置调度参数，需要先在代码里${变量名}（表示引用函数），然后在调度参数的赋值中输入具体的值。

例如odps sql类型节点，代码里${变量名}，节点配置参数项变量名=调度内置参数。

在代码中引用的参数，需要在调度中添加上解析的值。

Shell节点配置调度参数

Shell节点的参数配置和非Shell节点配置的步骤一样，只是规则有所不同，Shell节点中的变量不允许自定义命名，只能以$1,$2,$3…命名。

例如Shell类型节点，代码里Shell语法声明： $1调度中节点配置参数配置： $xxx（调度内置参数），即$xxx的值替换代码中的$1。

在代码中引用的参数，需要在调度中添加上解析的值。

说明:
在Shell节点中，参数到达第10个以后，应该使用 ${10} 的方式来声明变量。

变量值为固定值

以SQL类型节点为例，代码中${变量名}，节点配置参数项：变量名=固定值，例如`select xxxxxx type='${type}'`，如果调度变量赋值为`type='aaa'`，则在调度执行时，代码中会替换为type='aaa'。

变量值为调度内置参数

以SQL类型节点为例，代码里${变量名}， 节点配置参数项 变量名=调度参数。

代码：select xxxxxx dt=${datetime}

调度变量赋值：datetime=$bizdate

调度执行时，如果今日是2017年07月22日，代码里会替换成dt=20170721

调度内置参数列表

$bizdate：业务日期（格式yyyymmdd）说明：这个参数是用非常广泛，日常调度默认为前一天的日期。

示例：odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=$bizdate。今日是2017年07月22日，今天节点执行时$bizdate替换的时间即pt=20170721。

示例：odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=$gmtdate。今日是2017年07月22日，今天节点执行时$gmtdate替换的时间即pt=20170722。

示例：odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=$bizdate。今日是2017年07月01日，今天节点执行时$bizdate替换的时间即pt=20130630。

示例：odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=$gmtdate。今日是2017年07月01日，今天节点执行时$gmtdate替换的时间即pt=20170701。

$cyctime参数解释：任务的定时时间,如果天任务无定时，就是当天0点整（精确到时分秒，一般是小时/分钟级调度任务使用） 使用样例：cyctime=$cyctime。

说明:

$[] 和${} 配置时间参数区别$bizdate 业务日期，默认为当前时间减一天;$cyctime 任务定时调度时间，如果天任务并没有设置定时，就是当天0点整（精确到时分秒，一般是小时/分钟级调度任务使用），如果定时在0点30运行，以当天为例，就是yyyy-mm-dd 00:30:00如果{}参数，就是以bizdate为基准参与运算，补数据的时候选则什么业务日期，参数替换结果就是什么业务日期。如果是[]参数，是以cyctime为基准参与运行，具体使用方法见以下说明文档，和oracle的时间运算方式一致。补数据的时候，选中什么业务日期，参数替换结果是业务日期+1天，如果补数据选20140510这个业务日期，执行时cyctime替换结果是20140511。

$jobid参数解释：任务所属的工作流ID。示例：jobid=$jobid。

$nodeid参数解释：节点ID。示例：nodeid=$nodeid。

$taskid参数解释：任务ID（节点实例ID）。示例：taskid=$taskid。

$bizmonth：业务月份（格式yyyymm）。

- 说明：当业务日期的月份等于当前月份时，$bizmonth=业务日期月份-1，否则$bizmonth=业务日期月份。
- 示例：odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=$bizmonth。今日是2017年07月22日，今天节点执行时$bizmonth替换的时间即pt=201706。

$gmtdate：当前日期（格式yyyymmdd） 说明：此参数默认为当天日期，补数据时传入的是业务日期+1。

${…}自定义参数解释

- 根据$bizdate参数自定义时间格式，其中yyyy表示4位的年份，yy表示2位的年份，mm表示月，dd表示天。可以将参数任意组合，比如：${yyyy}、${yyyymm}、${yyyymmdd}、${yyyy-mm-dd}等。
- $bizdate精确到年月日，因此${……}自定义参数也只能表示到年月日级别。
- 获取+/-周期的方法：

  后N年：${yyyy+N}

  前N年：${yyyy-N}

  后N月：${yyyymm+N}

  前N月：${yyyymm-N}

  后N周：${yyyymmdd+7*N}

  前N周：${yyyymmdd-7*N}

  后N天：${yyyymmdd+N}

  前N天：${yyyymmdd-N}

${yyyymmdd}：业务日期（格式yyyymmdd，值与$bizdate一致）

- 说明：这个参数是用非常广泛，日常调度默认为前一天的日期。此参数格式可以自定义格式，如${yyyy-mm-dd}格式为yyyy-mm-dd。
- 示例：odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=${yyyymmdd}。今日是2013年07月22日，今天节点执行时${yyyymmdd}替换的时间即pt=20130721。

${yyyymmdd-/+N} 年月日加减N天

${yyyymm-/+N} 年月加减N月

${yyyy-/+N} 年（yyyy）加减N年

${yy-/+N} 年（yy）加减N年

说明：yyyymmdd表示业务日期，之间可以支持任意分隔符，例如yyyy-mm-dd。上面的几个参数都是取自业务日期的年、月、日。

示例：

- odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=${yyyy-mm-dd}。今日是2018年07月22日，今天节点执行时${yyyy-mm-dd}替换的时间即pt=2018-07-21。
- odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=${yyyymmdd-2}。今日是2018年07月22日，今天节点执行时${yyyymmdd-2}替换的时间即pt=20180719。

- odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=${yyyymm-2}。今日是2018年07月22日，今天节点执行时${yyyymm-2}替换的时间即pt=201805。
- odps sql节点代码中pt=${datetime}，节点配置的参数配置为datetime=${yyyy-2}。今日是2018年07月22日，今天节点执行时${yyyy-2}替换的时间即pt=2016。

odps sql节点配置中多个参数赋值如：startdatetime=$bizdate enddatetime=${yyyymmdd+1} starttime=${yyyy-mm-dd} endtime=${yyyy-mm-dd+1}。

使用示例（假设$cyctime=20140515103000）

- $[yyyy] = 2014, $[yy] = 14, $[mm] = 05, $[dd] = 15, $[yyyy-mm-dd] = 2014-05-15, $[hh24:mi:ss] = 10:30:00, $[yyyy-mm-dd hh24:mi:ss] = 2014-05-1510:30:00
- $[hh24:mi:ss - 1/24] = 09:30:00
- $[yyyy-mm-dd hh24:mi:ss -1/24/60] = 2014-05-1510:29:00
- $[yyyy-mm-dd hh24:mi:ss -1/24] = 2014-05-1509:30:00
- $[add_months(yyyymmdd,-1)] = 2014-04-15
- $[add_months(yyyymmdd,-12*1)] = 2013-05-15
- $[hh24] =10
- $[mi] =30

测试$cyctime参数的方法：

实例运行后，右键单击查看节点属性，定时时间就是该实例的周期定时时间。

定时时间减1小时的参数执行替换结果。

3.12 组件管理

3.12.1 创建组件

本文为您介绍组件的定义、构成以及如何创建组件。

组件的定义

组件是一种带有多个输入参数和输出参数的SQL代码过程模板，SQL代码过程的处理过程一般是引入一到多个源数据表，通过过滤，连接和聚合等操作，加工出新的业务需要的目标表。

组件的价值

在实际业务实践中，有大量的SQL代码过程很类似，过程中输入的表和输出的表的结构是一样的或者是类型兼容的，仅仅是名字不同而已。此时组件的开发者就可以将这样的一个SQL过程抽象成为一个SQL组件节点，将里面可变的输入表抽象成输入参数，把里面可变的输出表抽象成输出参数，即可实现SQL代码的复用。

在使用SQL组件节点时，您只需要从组件列表中选择和自己业务处理过程类似的组件，为这些组件配置上自己业务中特定的输入表和输出表，不用再重复复制代码，就可以直接生成新的组件SQL节点。从而极大提高了开发效率，避免了重复开发。SQL组件节点生成后的发布与调度的操作方法都和普通的SQL节点一样。

组件的构成

一个组件就像一个函数的定义一样，由输入参数，输出参数和组件代码过程构成。

组件的输入参数

组件的输入参数具有参数名、参数类型、参数描述和参数定义等属性，参数类型有表类型（table）和字符串类型（string）。

- 表类型的参数：指定组件过程中要引用到的表，在使用组件时，组件的使用者可以为该参数填入其特定业务需要的表。

- 字符串类型的参数：指定组件过程中需要变化的控制参数，比如指定过程的结果表只输出每个区域的头N个城市的销售额，这个n是1还是3就可以通过字符串类型的参数进行控制。

  例如要指定过程的结果表输出那个省份的销售总额，可以设置一个省份字符串参数，指定不同的省份，即可获得指定省份的销售数据。
- 参数描述：描述该参数在组件过程中发挥的作用。
- 参数定义：只有表类型的参数才需要，是表结构的一个文本定义。用于指定组件的使用者需要为该参数提供一个和该表参数定义名字和类型兼容个输入表，组件过程才会正确运行。否则，组件过程运行时会因为找不到输入表中指定的字段名而报错。该输入表必须具有该表参数定义中指定的字段名和类型，顺序不限，有多余的其他字段也不禁止。参数定义仅为参考，为使用者提供帮助指示。
- 表参数的定义格式建议为：

  ```
  字段1名 字段1类型 字段1注释
  字段2名 字段2类型 字段2注释
  字段n名 字段n类型 字段n注释
  ```

  示例如下：

  ```
  area_id string ‘区域id’
  city_id string ‘城市id’
  order_amt double ‘订单金额’
  ```

组件的输出参数

- 组件的输出参数具有参数名、参数类型、参数描述和参数定义等属性，参数类型只有表类型（table），字符串类型的输出参数没有逻辑意义。
- 表类型的参数：指定组件过程中最终产出到的表，在使用组件的时候，组件的使用者可以为该参数填入 其特定业务下通过该组件过程要产出的结果表。
- 参数描述：描述该参数在组件过程中发挥的作用。
- 参数定义：是表结构的一个文本定义，用于指定组件的使用者应该为该参数 提供 一个 和 该表参数定义的数目一致，类型兼容的一个输出表，组件过程才会正确运行，否则运行的时候会因为字段个数不匹配或类型不兼容而报错。对于输出表的字段名，不要求和表参数定义的字段名必须一致。该定义仅为参考，为使用者提供帮助指示。
- 表参数的定义格式建议为：

  ```
  字段1名 字段1类型 字段1注释
  字段2名 字段2类型 字段2注释
  字段n名 字段n类型 字段n注释
  ```

  示例如下：

  ```
  area_id string ‘区域id’
  city_id string ‘城市id’
  ```

```
order_amt double  '订单金额'
rank bigint  '排名'
```

组件的过程体

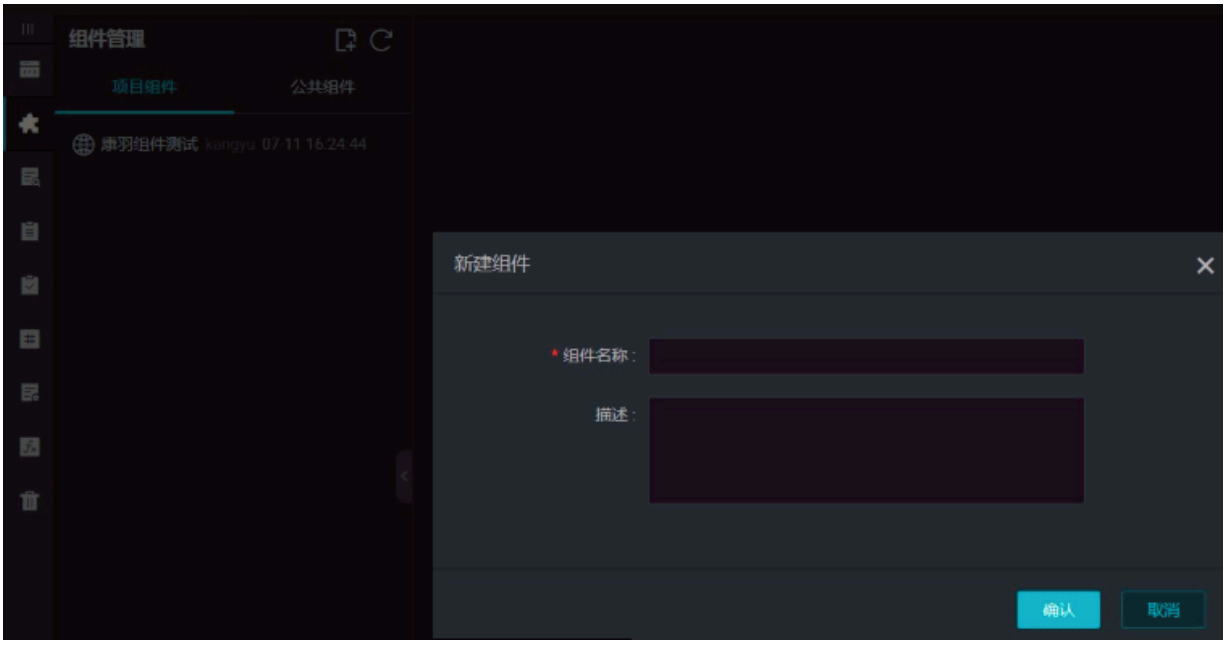
在过程体中参数的引用格式为：@@{参数名}。

过程体通过编写抽象的SQL加工过程，将指定的输入表按照输入参数进行控制加工出有业务价值的输出表。

组件过程的开发具有一定的技巧，组件过程的代码需要巧妙的利用输入参数和输出参数，使得组件过程能够在使用的时刻填入不同的输入参数和输出参数也能生成正确的可运行的SQL代码。

创建组件举例

通过点击下图框可新建组件。



原始表结构定义

销售数据的原始MySQL结构定义如下：

字段名称	字段类型	字段描述
order_id	varchar	订单编号
report_date	datetime	订单日期
customer_name	varchar	客户名称
order_level	varchar	订单等级
order_number	double	订单数量
order_amt	double	订单金额

字段名称	字段类型	字段描述
back_point	double	折扣点
shipping_type	varchar	运输方式
profit_amt	double	利润金额
price	double	单价
shipping_cost	double	运输成本
area	varchar	区域
province	varchar	省份
city	varchar	城市
product_type	varchar	产品类型
product_sub_type	varchar	产品小类
product_name	varchar	产品名称
product_box	varchar	产品包箱
shipping_date	datetime	运输日期

组件的业务含义

组件的名字：get_top_n

组件的描述：

通过指定的销售明细数据表作为输入参数（表类型）和取前多少名作为输入参数（字符串），按照城市销售总额的大小作为排名依据，通过该组件过程，组件的使用者可以轻松获取到各个区域下指定的前多少名的城市排行。

组件的参数定义

输入参数1：

参数名：myinputtable 类型：table

输入参数2：

参数名：topn 类型：string

输出参数3:

参数名：myoutput 类型：table

参数定义：

area_id string

city_id string

order_amt double

rank bigint

建表语句：

```
CREATE TABLE IF NOT EXISTS company_sales_top_n
(
area STRING COMMENT '区域',
city STRING COMMENT '城市',
sales_amount DOUBLE COMMENT '销售额',
rank BIGINT COMMENT '排名'
)
COMMENT '公司销售排行榜'
PARTITIONED BY (pt STRING COMMENT '')
LIFECYCLE 365;
```

组件过程体定义

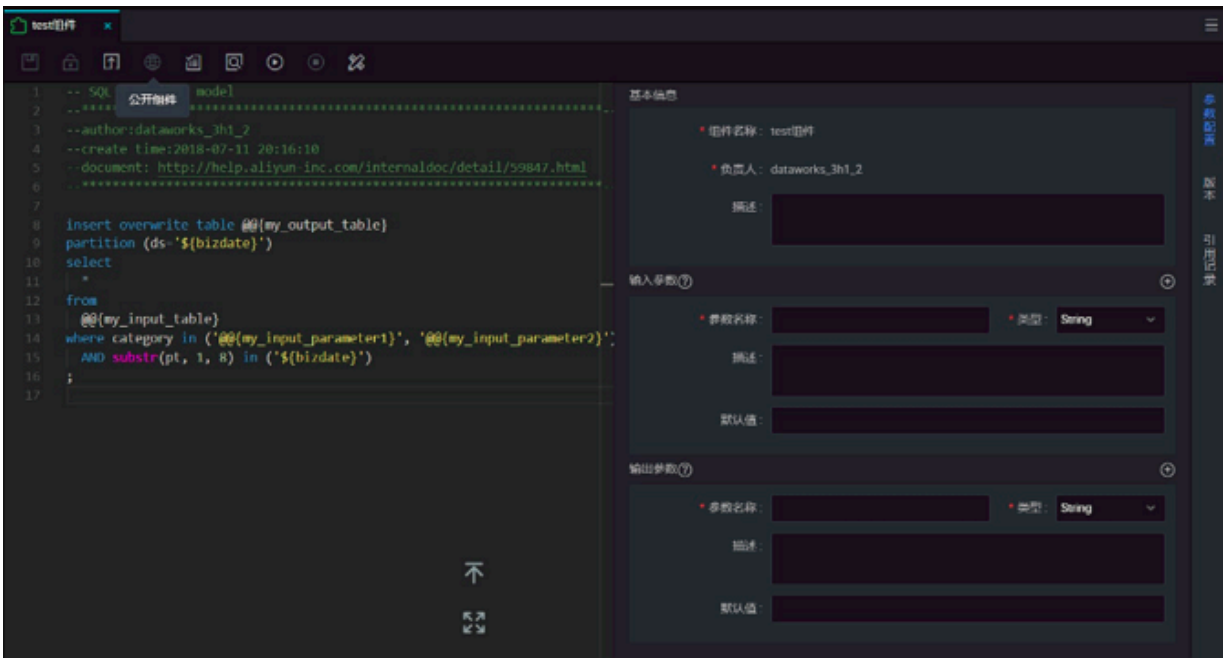
```
INSERT OVERWRITE TABLE @@{myoutput} PARTITION (pt='${bizdate}')
    SELECT r3.area_id,
    r3.city_id,
    r3.order_amt,
    r3.rank
from (
SELECT
    area_id,
    city_id,
    rank,
    order_amt_1505468133993_sum as order_amt ,
    order_number_1505468133991_sum,
    profit_amt_1505468134000_sum
FROM
    (SELECT
    area_id,
    city_id,
    ROW_NUMBER() OVER (PARTITION BY r1.area_id ORDER BY r1.order_amt_
1505468133993_sum DESC)
AS rank,
    order_amt_1505468133993_sum,
    order_number_1505468133991_sum,
    profit_amt_1505468134000_sum
FROM
    (SELECT area AS area_id,
     city AS city_id,
     SUM(order_amt) AS order_amt_1505468133993_sum,
     SUM(order_number) AS order_number_1505468133991_sum,
     SUM(profit_amt) AS profit_amt_1505468134000_sum
FROM
    @@{myinputtable}
WHERE
    SUBSTR(pt, 1, 8) IN ( '${bizdate}' )
GROUP BY
    area,
    city )
    r1 ) r2
WHERE
    r2.rank >= 1 AND r2.rank <= @@{topn}
ORDER BY
```

```
    area_id,
    rank limit 10000) r3;
```

组件的分享范围

组件的分享范围有两种：项目组件和公共组件。

组件发布后默认在本项目内其他用户可见可用。组件的开发者通过点击“公开组件”按钮可以将具有全局通用性的组件发布到 整个租户内，所有租户内的用户都能看到该公共组件并可使用。组件是否是公开组件取决于下列框内图标是否可见：



组件的使用

组件开发完成后的使用请参见#unique_419。

组件的引用记录

单击右侧的引用记录，组件的开发者可以查看该组件的引用记录。

3.12.2 使用组件

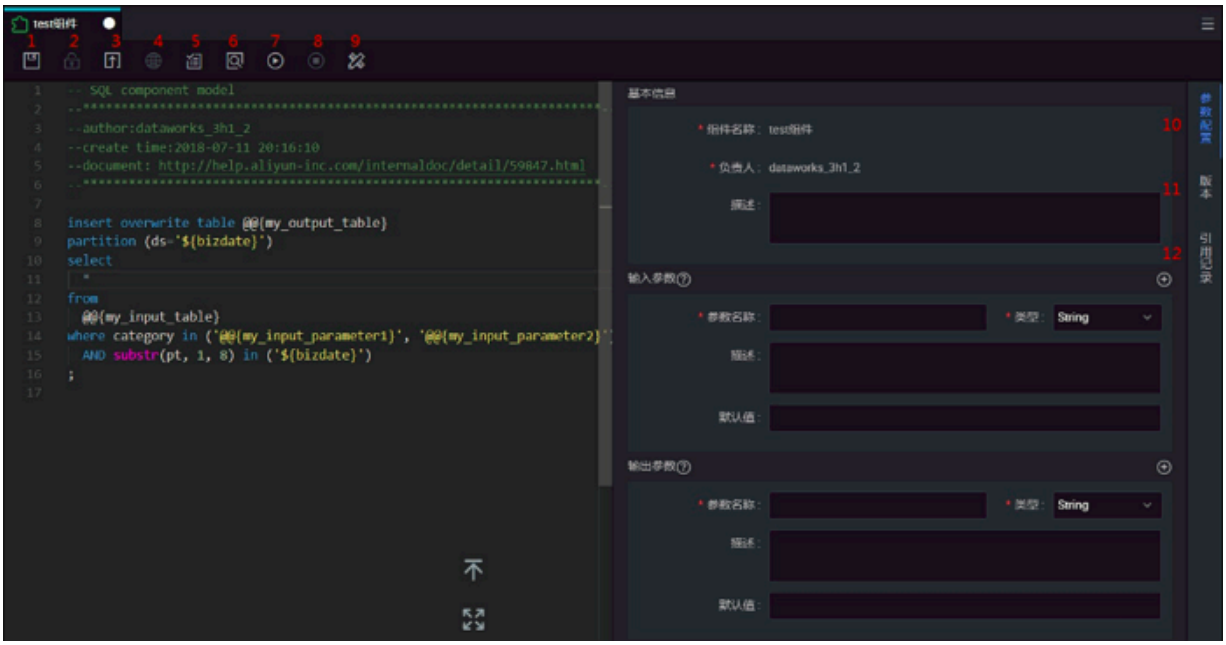
为提高开发效率，数据任务的开发者可以使用项目成员和租户成员贡献的组件来新建数据处理节点。

使用组件的注意事项：

- 本项目成员创建的组件在项目组件下。
- 租户成员创建的组件在公共组件下。

组件的具体使用方法请参见#unique_300。

界面功能介绍



界面功能说明如下：

序号	功能	说明
1	保存	保存当前组件的设置。
2	偷锁编辑	非组件责任人可以偷锁编辑此节点。
3	提交	将当前组件提交到开发环境。
4	公开组件	将具有全局通用性的组件发布到整个租户内，所有租户内的用户都能看到该公共组件并可使用。
5	输入输出参数解析	解析当前代码的输入输出参数。 说明： 此处填写的参数通常为表名称，而非调度参数。
6	预编译	对当前组件的自定义参数、组件参数进行编辑。

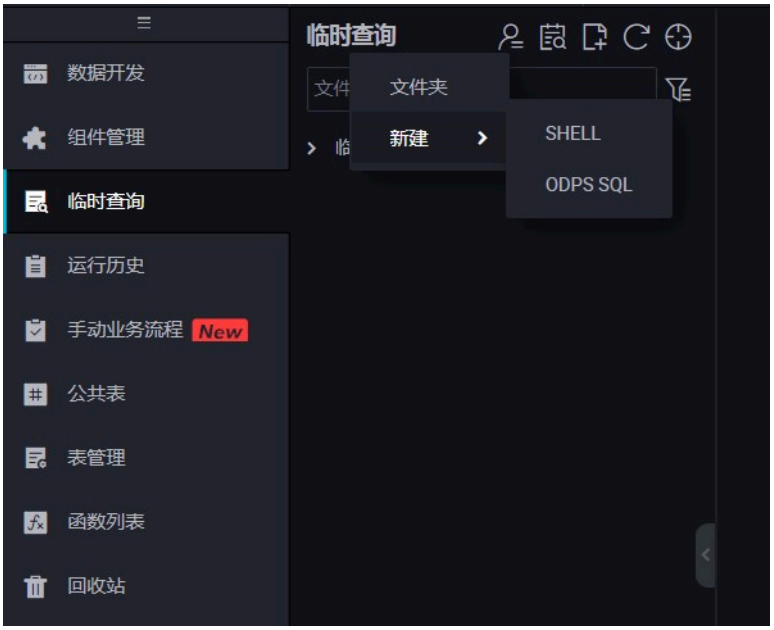
序号	功能	说明
7	运行	在本地（开发环境）运行组件。
8	停止运行	停止运行的组件。
9	格式化	对当前组件代码根据关键字格式排列。
10	参数配置	组件信息、输入参数、输出参数配置。
11	版本	组件提交发布记录。
12	引用记录	组件被使用的记录汇总。

3.13 临时查询

临时查询用来方便您在本地测试代码的实际情况与期望值是否相符、排查代码错误等。因此，临时查询不需提交、发布和设置调度参数。如果您需要使用调度参数，请在数据开发或业务流程中创建节点。

创建文件夹

1. 单击左侧导航栏中的临时查询，选择文件夹。



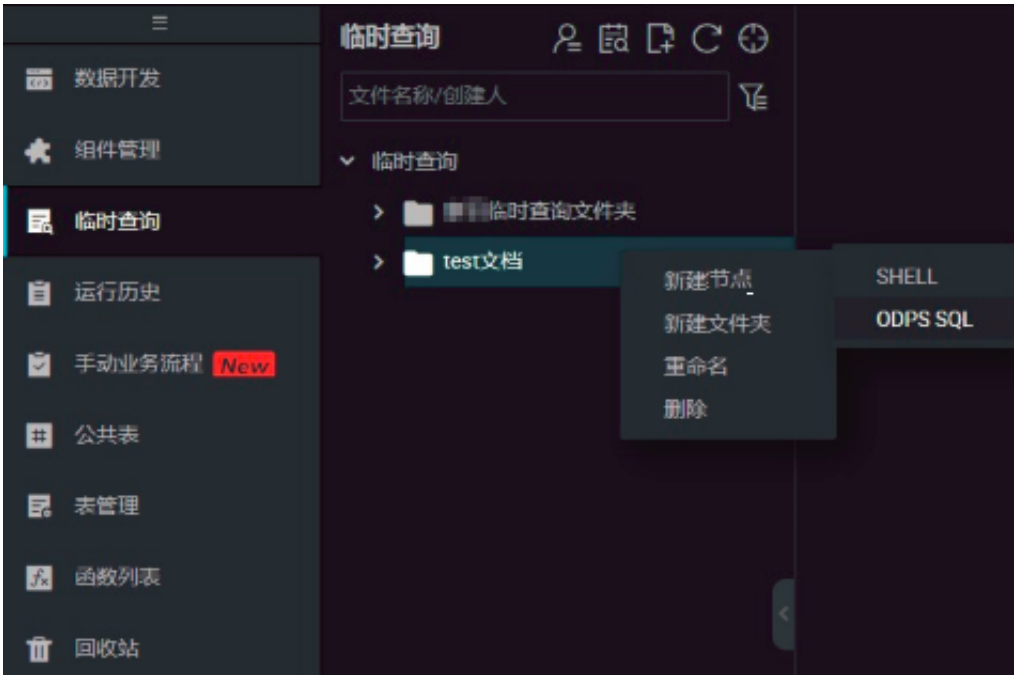
2. 输入文件夹名称，选择文件夹的位置，单击提交。

说明:

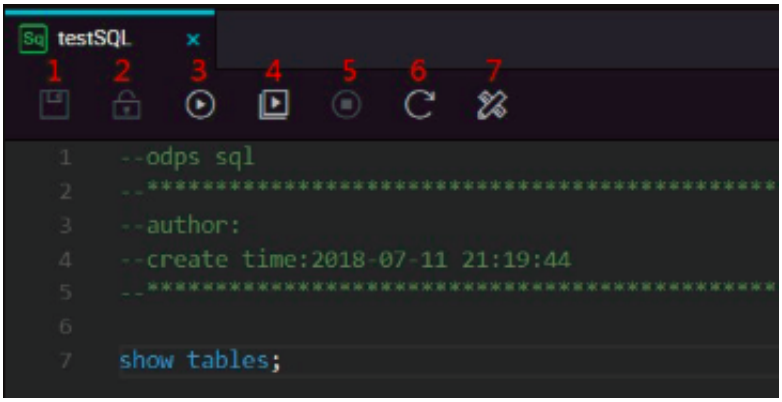
可以存在多级文件夹目录，因此可以将此文件夹放在已创建好的其他文件夹下。

创建节点

临时查询下仅有SHELL和SQL两个节点。



以新建ODPS SQL节点为例，右键单击文件夹名称，选择新建节点 > ODPS SQL。



序号	功能	说明
1	保存	保存当前输入的代码。
2	偷锁编辑	非节点责任人可以通过偷锁编辑对此节点进行操作。
3	运行	在本地（开发环境）运行代码。
4	高级运行（带参数运行）	如果代码中有参数，带参数运行代码。 说明: SHELL节点不存在高级运行。
5	停止运行	停止正在运行的节点。
6	重新加载	刷新页面，重新加载后恢复到上一次保存的状态，未保存的内容将丢失。 说明: 如果在配置中心已打开缓存，刷新后会提示缓存了未保存的代码，请选择您需要的版本。
7	格式化	对当前节点代码根据关键字格式排列，常用于单行代码过长的情况。

3.14 运行历史

运行历史栏会展示最近三天内在本地运行过的所有任务记录，单击打开后可查看任务历史，支持按任务状态去过滤运行记录。

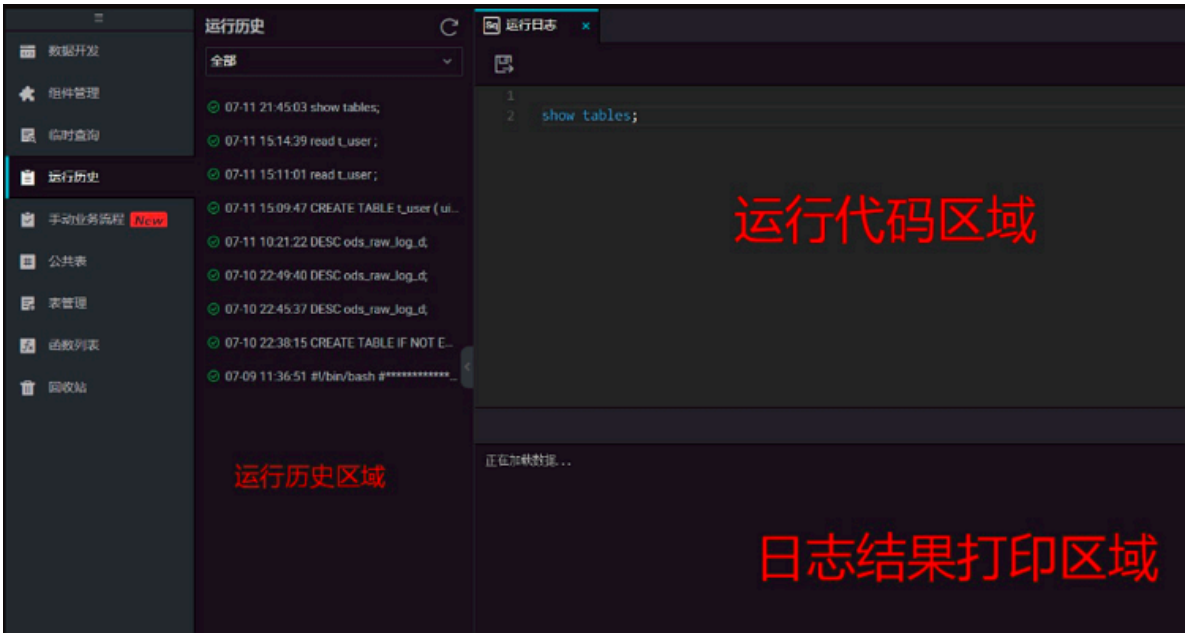
说明:
运行历史只保留三天。

查看运行历史

1. 单击左侧导航栏中的运行历史，切换到运行历史模块（默认展示全部状态）。

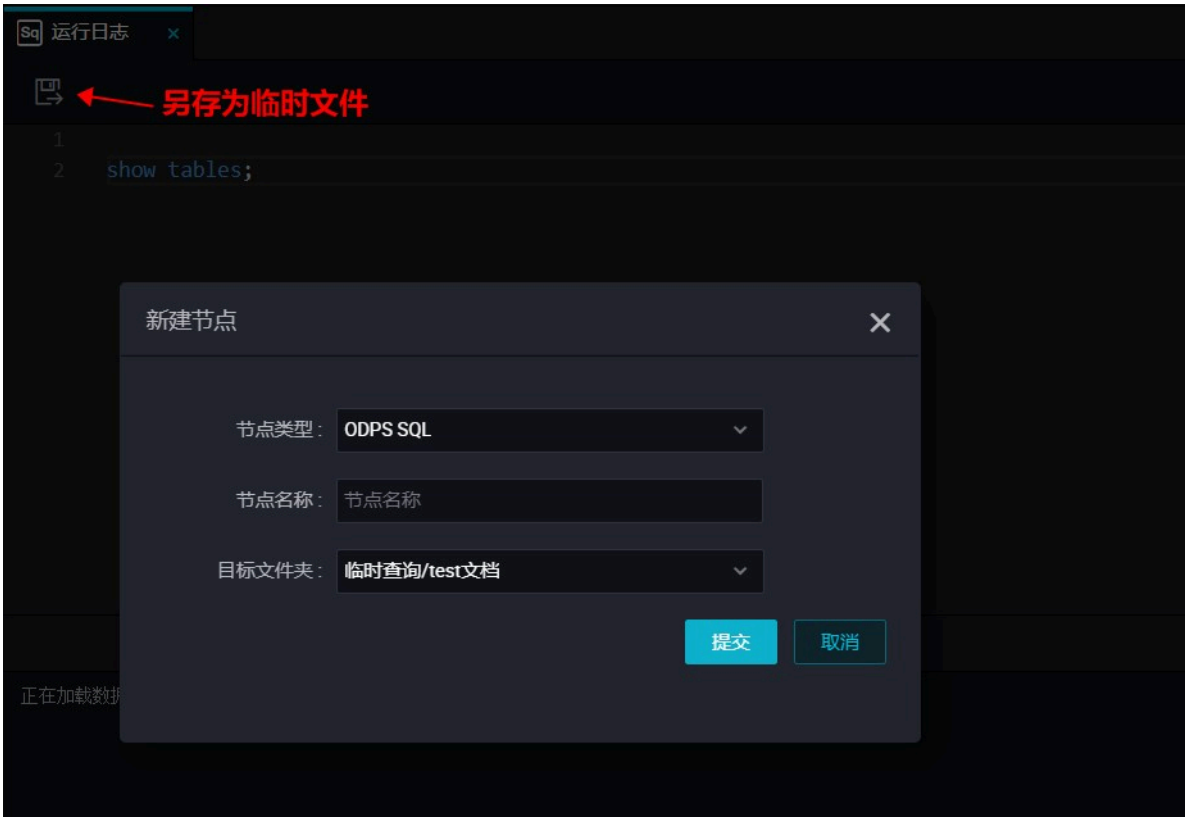
2. 单击下拉框，选择要过滤的任务状态。

3. 单击需要查看的运行记录，打开运行历史页，展示此运行记录的运行日志。



另存为临时文件

如果需要保存运行记录中的SQL语句，您可单击保存，将运行过的SQL记录另存为临时文件。



输入文件名和目录后，单击提交即可。

3.15 公共表

公共表用于查看当前主账号下所有项目创建的表。

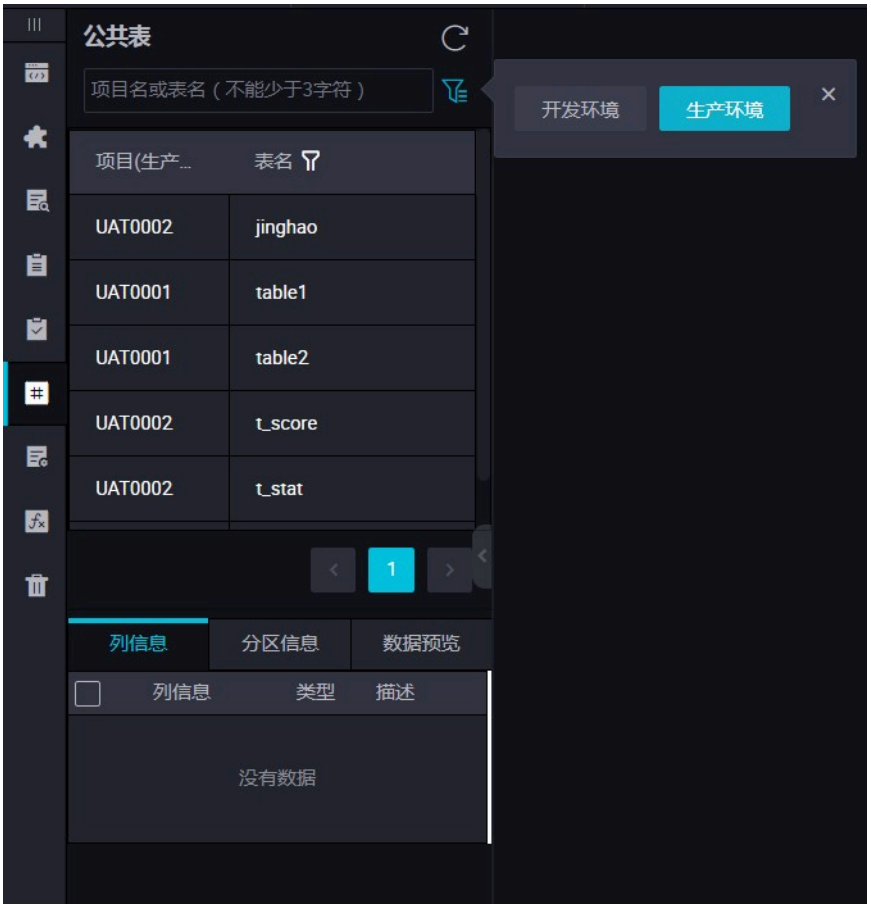
- 项目：指项目名，会在项目名称前加前缀：odps.（例如项目名为test，则显示：odps.test）
- 表名：项目下这张表的名称

点击表名会在下方展示此表的列信息、分区信息、数据预览。

- 列信息：查看这张表的字段数量、字段类型及描述
- 分区信息：查看这张表的分区信息、分区数量，分区数最大为6万个（如果设置过生命周期，实际分区数以生命周期为主）。
- 数据预览：当前表数据预览。

环境切换

与表管理相同，公共表也存在开发与生产两个环境，当前环境会以蓝色底展示，点击需要查询的环境，可切换至相应的环境中。



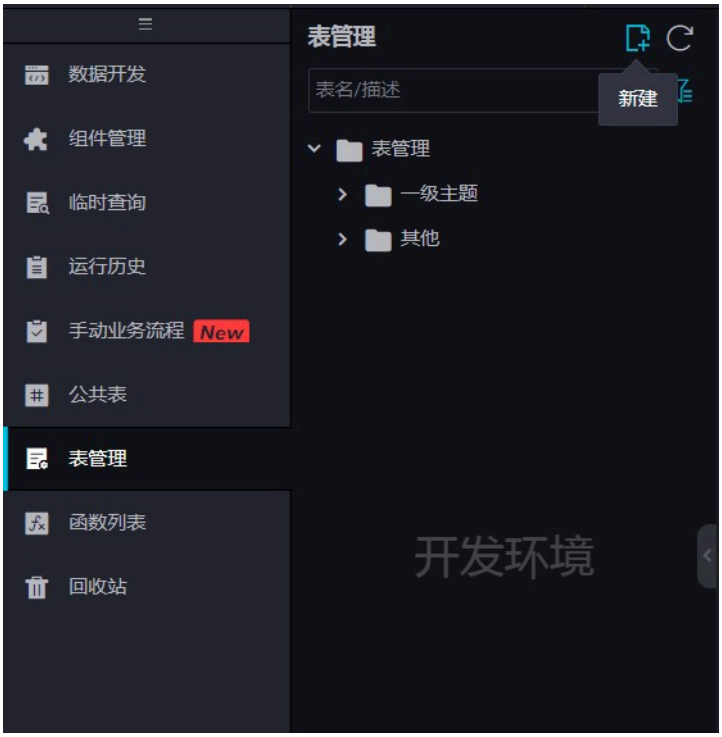
3.16 表管理

本文将为您介绍如何对数据表进行创建、提交和查询等操作，以及大数据数据分层的基础知识。

新建表

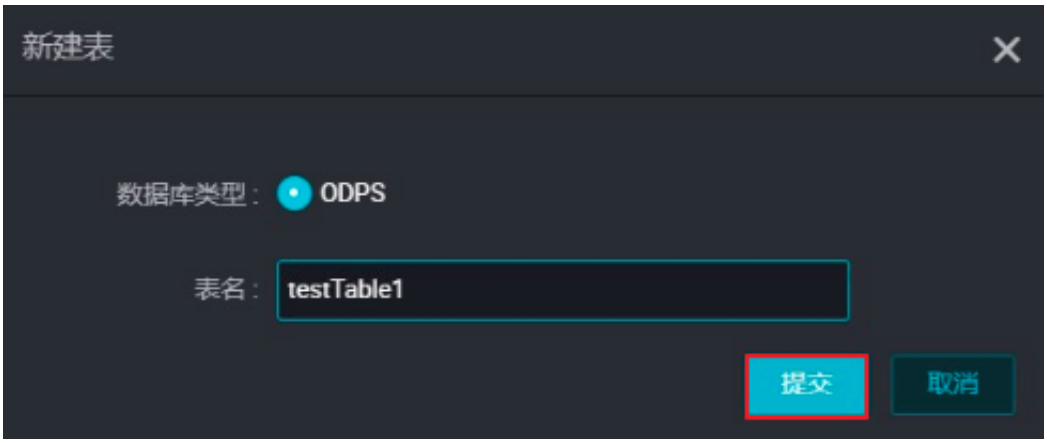
1. 单击左侧导航栏中的表管理。

2. 选择+新建表。

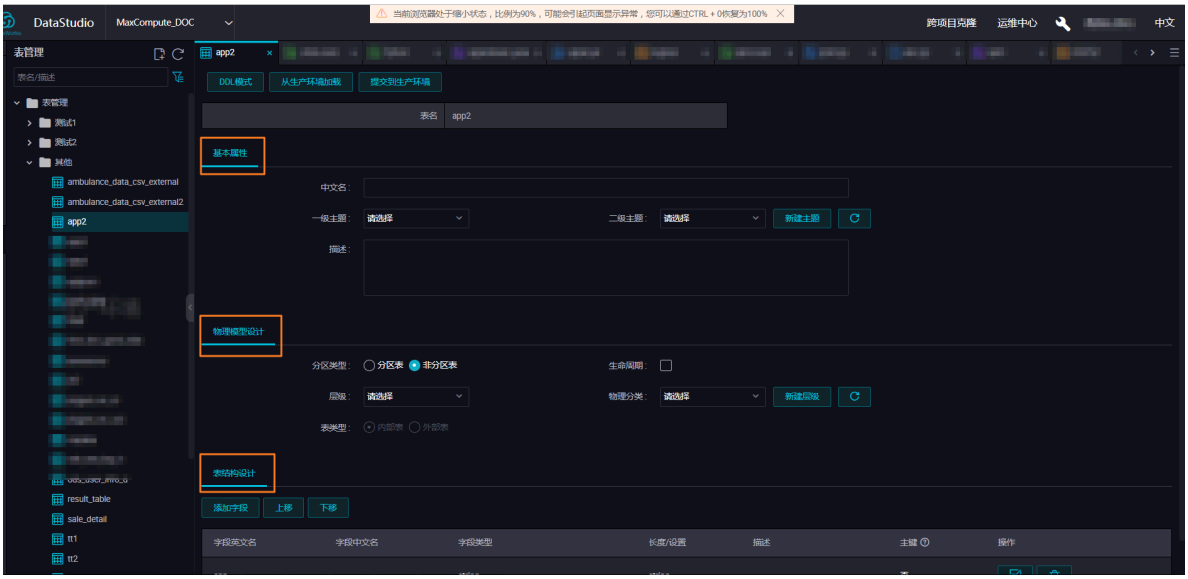


3. 输入表名，单击提交。

> **说明:**
> 目前仅支持新建ODPS表。



4. 设置表的基本属性。



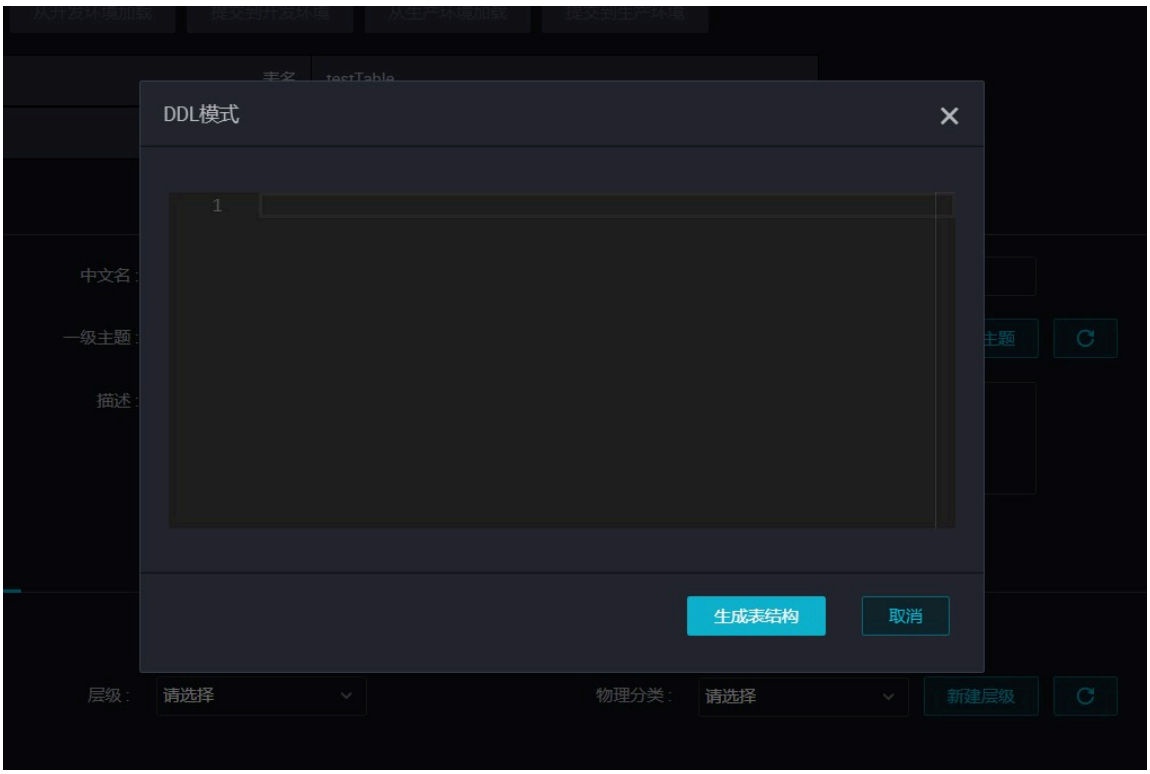
配置	说明
中文名	表的中文名称。
一级主题	新建表所处的一级目标文件夹名称。 **说明:** 一级、二级主题仅仅是DataWorks上文件夹的摆放形式，目的是为了您能更好地管理您的表。
二级主题	新建表所处的二级目标文件夹名称。
新建主题	单击新建主题，跳转至主题管理页面，您可以在该页面创建一级主题、二级主题。 
描述	针对新建表的描述。

5. 创建表。

您可以通过以下两种方式创建表：

- 使用DDL模式创建表。

  单击DDL模式，在对话框中输入标准的建表语句。



  编辑好建表语句后，单击生成表结构，即可自动填充基本属性、物理模型设计、表结构设计中的相关内容。

- 使用图形界面创建表。

  如果不适用于DDL模式建表，您也可以使用图形界面直接建表，相关设置说明如下。

分类	配置	说明
物理模型设计	表类型	包括分区表和非分区表两种类型。
	保存周期	即MaxCompute的生命周期功能。填写一个数字表示天数，该表（或分区）超过一定天数，未更新的数据会被清除。
	层级	通常可以分为DW、ODS和RPT三个层级。 关于物理层级的相关信息请参见#unique_402。

分类	配置	说明
	物理分类	包括基础业务层、高级业务层和其他。 单击新建层级，跳转至层级管理页面，即可在此新增层级。 说明: 物理分类仅为方便您的管理，不涉及底层实现。
表结构设计	字段英文名	字段英文名，由字母、数字和下划线组成。
	中文名	字段的中文名称。
	字段类型	MaxCompute数据类型，仅支持STRING、BIGINT、DOUBLE、DATETIME和BOOLEAN类型，详情请参见数据类型。
	描述	字段的详细描述。
	主键	勾选表示该字段是主键，或者是联合主键的其中一个字段。
	添加字段	新增一列字段。
	删除字段	删除已经创建的字段。 说明: 已经创建的表，删除字段重新提交时，会要求删除当前表，再去建一张同名表，在生产环境中禁止该操作。
	上移	调整未创建的表的字段顺序。如果为已经创建的表调整字段顺序，会要求删除当前已经创建的表，再去建一张同名表，在生产环境中禁止该操作。
	下移	同上移操作。
	添加分区	可以给当前的表新建一个分区。如果为已经创建的表添加分区，会要求删除当前已经创建的表，再去建一张同名表，该操作在生产环境中禁止。
	删除分区	可以删除一个分区。如果删除已创建的表的 分区，会要求删除当前已经创建的表，再去建一张同名表，在生产环境中禁止该操作。

分类	配置	说明
	操作	包括针对新增字段的确认提交和删除，以及更多属性编辑。 更多属性主要是数据质量相关的信息，提供给系统用于生成校验逻辑。将会用于Data Profiling、SQLScan、测试规则生成等场景。 - 允许为零：勾选表示该字段的值允许为零，仅针对BIGINT和DOUBLE类型的字段。 - 允许为负数：勾选表示该字段的值允许为负数，仅针对BIGINT和DOUBLE类型的字段。 - 安全等级：安全等级为0~4，数字越大代表安全要求越高。如果您的安全等级未达到数字要求，则无法访问表格对应字段。 - 单位：指金额单位，元或者分。非金额含义的字段不必填此项。 - Lookup表名/键值：适用于枚举值型的字段（例如会员类型、状态等）。您可以填该字段对应的字典表（即维表）的表名称，例如会员状态对应的字典表名是dim_user_status。 如果您采用的是全局唯一的字典表，此处应填本字段在字典表中对应的key_type键值类型，例如会员状态对应的键值是TAOBAO_USER_STATUS。 - 值域范围：本字段适用的最大值、最小值，仅针对BIGINT和DOUBLE类型的字段。 - 正则校验：本字段使用的正则表达式。例如是手机号码字段，则可以通过正则表达式来约束它的值为11位数字（乃至更严格的约束）。 - 最大长度：字段值的最大字符个数，仅针对STRING类型的字段。 - 日期精度：日期值的实际精度，时、日、月等。例如月汇总表中的month_id的精度是月，尽管它存的值是例如2014-08-01（看起来精度是日）。适用于DATETIME类型或以STRING类型存放的日期值。 - 日期格式：仅适用于以STRING类型存放的日期值。用类似于yyyy-mm-dd hh:mi:ss的方式来

分类	配置	说明
分区字段设计 说明：当物理模型设计选择分区表后才显示分区字段设计。	字段类型	建议统一采用STRING类型。
	日期分区格式	如果该分区字段是日期含义（尽管数据类型可能是STRING），则一个或自填一个日期格式，常用格式为yyyymmmdd、yyyy-mm-dd。
	日期分区粒度	支持的分区粒度有秒/分/时/日/月/季度/年。创建分区粒度根据需要可自行填写，如果需要填写多个分区粒度，则默认粒度越大，分区等级越高。例如同时存在日、时、月三个分区，多级分区关系是一级分区（月），二级分区（日），三级分区（时）。

提交表

编辑完表结构信息后，即可将新建表提交到开发环境和生产环境。

配置	说明
从开发环境加载	如果该表已经提交到开发环境之后，该按钮会高亮。单击后，会用开发环境已经创建的表信息覆盖当前的页面信息。
提交到开发环境	首先会检查当前编辑页面的必填项是否已经填写完整，如果有遗漏会告警，禁止提交。
从生产环境加载	已经提交到生产环境的表的详细信息会覆盖当前页面。
提交到生产环境	会在生产环境的project中创建这张表。

表分类查询

表管理查询支持开发环境、生产环境的筛选条件，查询结果以文件夹为主题展示。

- 开发环境：仅查询开发环境的表。
- 生产环境：仅查询生产环境的表，生产环境表请谨慎操作。

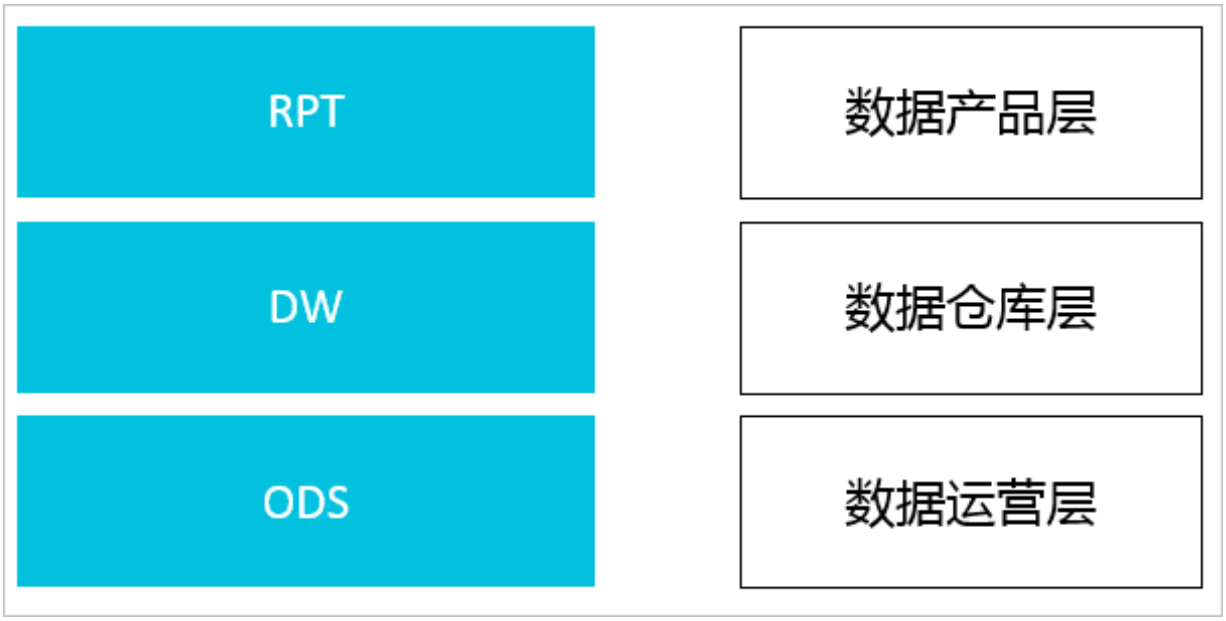
> 说明：
> 以tmp_pyodps开头的表为#unique_425任务在执行过程中产生的临时表，不会被自动删除。您可以通过使用脚本或SQL语句定期清除PyODPS临时表。

修改表名

您的表在创建之后，如果还未提交，可以通过在图形界面删除重建的方式修改表名。如果已经提交表到开发或生产环境，您可以通过odpscmd客户端，使用ALTER语句修改表名。

数仓分层

表管理中的物理模型设计用于为您构建您的数仓分层，让您在管理数据时能对数据有更加清晰的规划和掌控。DW、ODS和RPT三个层级是常见的数仓分层方法。



- ODS数据运营层

  ODS数据运营层用于操作数据存储，是最接近数据源中的数据的一层。数据源中的数据，经过抽取、洗净、传输（ETL）之后导入本层。ODS的数据通常可按照源头业务系统的分类方式而分类。

  > 说明:
  > ODS层的数据不等同于原始数据。在源数据装入这一层时，要进行诸如去噪、去重、去除脏数据、业务提取和单位统一等多项工作。

- DW数据仓库层

  DW数据仓库层是数据仓库的主体。在DW层，从ODS层中获得的数据，根据主题建立各种数据模型。

- RPT数据产品层

  RPT数据产品层提供数据产品、数据挖掘和数据分析使用的数据结果，供线上系统使用。例如报表数据或宽表，通常存放在RPT层。

3.17 外部表

本文为您详细介绍如何使用DataWorks创建、配置外部表及支持的字段。

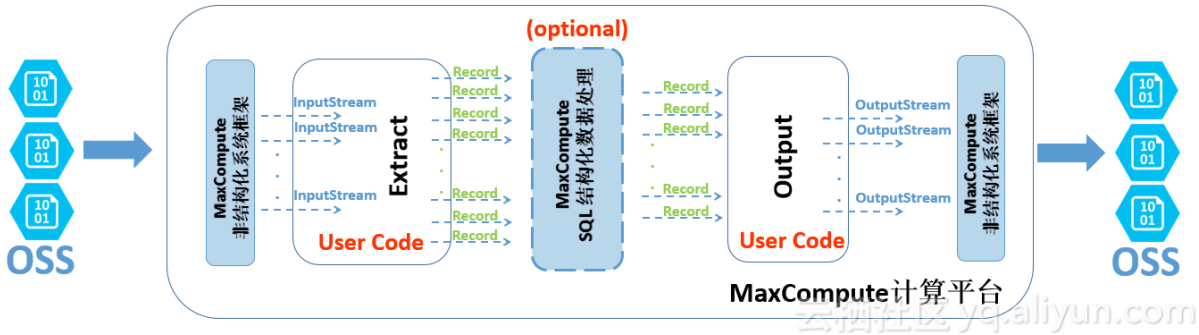
外部表概述

在使用外部表前，您需要了解以下定义。

名称	说明
对象存储OSS	提供标准、低频、归档存储类型，能够覆盖从热到冷的不同存储场景。同时，OSS能够与Hadoop开源社区及EMR、批量计算、MaxCompute、机器学习PAI、DatalakeAnalytics、函数计算等阿里云计算产品进行深度结合。
MaxCompute	一项大数据计算服务，能够提供快速且完全托管的数据仓库解决方案，并可以与OSS结合，高效并经济地分析处理海量数据。MaxCompute的处理性能达到了全球领先水平，被Forrester评为全球云端数据仓库领导者。
MaxCompute外部表	该功能基于MaxCompute新一代的2.0计算框架，可以帮助您直接对OSS中的海量文件进行查询，而不必将数据加载到MaxCompute表中，既节约了数据搬迁的时间和人力，也节省了多地存储的成本。MaxCompute外部表对OTS的处理方法与OSS类似。

外部表整体处理架构如下图所示。

【OSS -> MaxCompute -> OSS】数据计算链路



当前MaxCompute主要支持的外部表为非结构化存储：OSS及OTS。从数据的流动和处理逻辑的角度，您可以简单地把非结构化处理框架理解成在MaxCompute计算平台两端有机耦合的的数据导入以及导出。此处以OSS外部表的处理逻辑举例进行说明。

1. 外部的OSS数据经过非结构化框架转换，使用JAVA InputStream类提供给用户自定义代码接口。 用户自己实现Extract逻辑，只需要负责对输入的InputStream做读取/解析/转化/计算，最终返回MaxCompute计算平台通用的Record格式。
2. 上述Record可以自由参与MaxCompute的SQL逻辑运算，这一部分计算基于MaxCompute内置的结构化SQL运算引擎，并可能产生新的Record。
3. 经过运算的Record传递给用户自定义的Output逻辑，您在这里可以进行进一步的计算转换，并最终将Record里面需要输出的信息通过系统提供的OutputStream输出，由系统负责写入到OSS。

您可以通过DataWorks配合MaxCompute对外部表进行可视化的创建、搜索、查询、配置、加工和分析。

网络与权限认证

由于MaxCompute与OSS是两个独立的云计算与云存储服务，所以在不同的部署集群上的网络连通性有可能影响MaxCompute访问OSS的数据的可达性。在MaxCompute公共云服务上访问OSS存储时，建议您使用OSS私网地址（即以-internal.aliyuncs.com结尾的host地址）。

MaxCompute计算服务要访问OSS 数据需要有一个安全的授权通道。MaxCompute结合了阿里云的访问控制服务（RAM）和令牌服务（STS）来实现对数据的安全访问：MaxCompute在获取权限时，以表的创建者的身份去STS申请权限，OTS的权限设置同理。

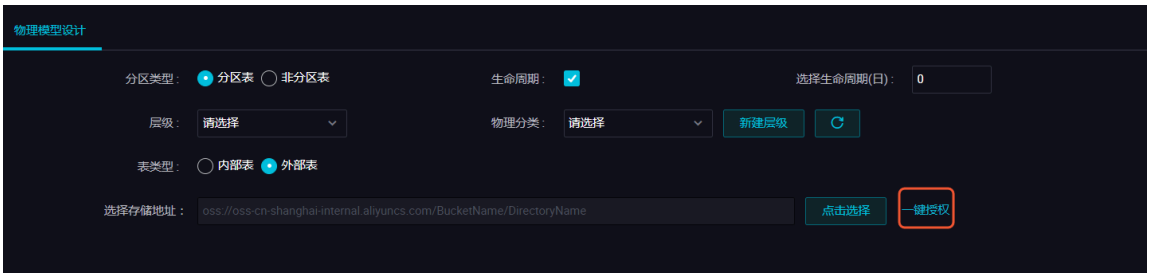
1. STS模式授权

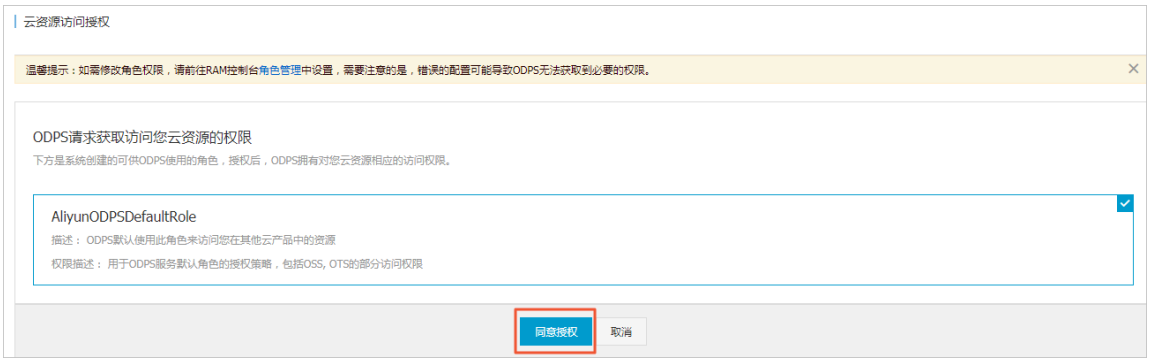
MaxCompute需要直接访问OSS的数据，因此需要将OSS数据相关权限赋给MaxCompute的访问账号。STS是阿里云为客户提供的一种安全令牌管理服务，它是资源访问管理（RAM）产品中的一员。通过STS服务，获得许可的云服务或RAM用户可以自主颁发自定义时效和子权限的访问令牌。获得访问令牌的应用程序可以使用令牌直接调用阿里云服务API操作资源。

详情请参见OSS的STS模式授权。

您可通过以下两种方式授予权限：

- 当MaxCompute和OSS的项目所有者是同一个账号时，可以直接登录阿里云账号后一键授权。一键授权页面可通过数据开发/新建表时点击选取，如下图所示。





- 自定义授权。 首先需要在RAM中授予MaxCompute访问OSS的权限。
登录 RAM控制台（若MaxCompute和OSS不是同一个账号，此处需

由OSS账号登录并授权），通过控制台中的角色管理创建用户角色，角色名为AliyunODPSDefaultRole或AliyunODPSRoleForOtherUser。如下图所示：

配置角色详情：

```
--当MaxCompute和OSS的Owner是同一个账号
{
"Statement": [
{
"Action": "sts:AssumeRole",
"Effect": "Allow",
"Principal": {
"Service": [
"odps.aliyuncs.com"
      ]
    }
  }
],
"Version": "1"
}
--当MaxCompute和OSS的Owner不是同一个账号
{
"Statement": [
{
"Action": "sts:AssumeRole",
"Effect": "Allow",
"Principal": {
"Service": [
"MaxCompute的Owner云账号id@odps.aliyuncs.com"
      ]
    }
  }
],
"Version": "1"
}
```

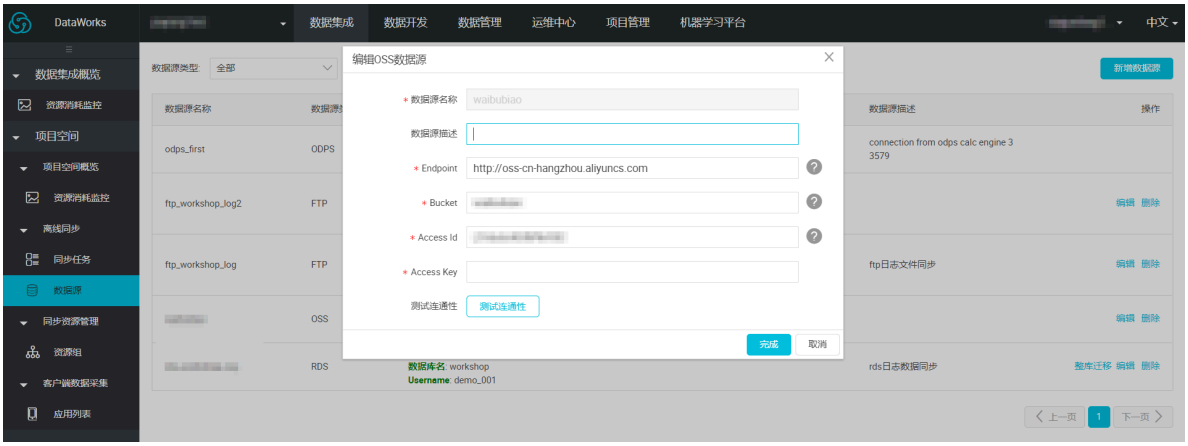
配置角色授权策略。并找到授予角色访问OSS必要的权限AliyunODPSRolePolicy，并将权限AliyunODPSRolePolicy授权给该角色。如果您无法通过搜索授权找到，可通过精确授权直接添加。该权限详细信息如下，供您参考。

```
{
  "Version": "1",
  "Statement": [
    {
```

```
        "Action": [
          "oss:ListBuckets",
          "oss:GetObject",
          "oss:ListObjects",
          "oss:PutObject",
          "oss:DeleteObject",
          "oss:AbortMultipartUpload",
          "oss:ListParts"
          ],
          "Resource": "*",
          "Effect": "Allow"
    },
    {
        "Action": [
          "ots:ListTable",
          "ots:DescribeTable",
          "ots:GetRow",
          "ots:PutRow",
          "ots:UpdateRow",
          "ots:DeleteRow",
          "ots:GetRange",
          "ots:BatchGetRow",
          "ots:BatchWriteRow",
          "ots:ComputeSplitPointsBySize"
        ],
        "Resource": "*",
        "Effect": "Allow"
      }
    ]
}
```

2. **使用数据集成OSS 数据源**

   数据集成OSS数据源保存了已经事先创建好的OSS数据源，可以直接使用。



创建外部表

1. **DDL模式建表**

   进入数据开发页面，参考#unique_284进行DDL模式建表，您需只需遵守正常的MaxCompute语法即可。如果您的STS服务已成功授权，可以不写`odps.properties.rolearn`**属性。DDL建表语句举例如下所示，其中**`EXTERNAL`**参数说明此表为外部表。**

```
CREATE EXTERNAL TABLE IF NOT EXISTS ambulance_data_csv_external(
vehicleId int,
recordId int,
patientId int,
calls int,
locationLatitute double,
locationLongtitue double,
recordTime string,
direction string
)

STORED BY 'com.aliyun.odps.udf.example.text.TextStorageHandler' --
STORED BY用于指定自定义格式StorageHandler的类名或其他外部表文件格式，必选。
with SERDEPROPERTIES (
'delimiter'='\\|', --SERDEPROPERTIES 序列化属性参数，可通过 DataAttrib
utes 传递到 Extractor 代码中，可选。
'odps.properties.rolearn'='acs:ram::xxxxxxxxxxxxx:role/aliyunodps
defaultrole'
)
LOCATION 'oss://oss-cn-shanghai-internal.aliyuncs.com/oss-odps-test
/Demo/SampleData/CustomTxt/AmbulanceData/'                --外部表存放地
址，必选。
USING 'odps-udf-example.jar'; --指定自定义格式时类定义所在的jar包，如果未使
用自定义格式无需指定。
```

关于STORED BY后接参数，其中CSV或TSV文件对应 默认内置的StorageHandler，具体参数如下：

- CSV为com.aliyun.odps.CsvStorageHandler ，定义如何读写CSV格式数据，数据格式约定：英文逗号,为列分隔符，换行符为\n。实际参数输入举例：STORED BY'com.aliyun.odps.CsvStorageHandler'。
- TSV为 com.aliyun.odps.TsvStorageHandler，定义如何读写TSV格式数据，数据格式约定：\t为列分隔符，换行符为\n。

STORED BY后接参数还支

持ORC、PARQUET、SEQUENCEFILE、RCFILE、AVRO和TEXTFILE 开源格式外部表，如下所示。对于 textFile 可以指定序列化类，例如org.apache.hive.hcatalog.data.JsonSerDe。

- org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe -> stored as textfile
- org.apache.hadoop.hive.ql.io.orc.OrcSerde -> stored as orc
- org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe -> stored as parquet
- org.apache.hadoop.hive.serde2.avro.AvroSerDe -> stored as avro
- org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe -> stored as sequencefile

对于开源格式外部表，建表语句如下。

```
CREATE EXTERNAL TABLE [IF NOT EXISTS] (<column schemas>)
[PARTITIONED BY (partition column schemas)]
[ROW FORMAT SERDE '']
STORED AS
[WITH SERDEPROPERTIES ( 'odps.properties.rolearn'='${roleran}'
[,'name2'='value2',...]
) ]
LOCATION 'oss://${endpoint}/${bucket}/${userfilePath}/';
```

SERDEPROPERTIES序列化属性列表如下。MaxCompute目前只支持通过内置extractor读取OSS上gzip压缩的CSV/TSV数据，用户可以选择 文件是否是gzip压缩的。不同的文件格式对应不同的属性设置。

属性名	属性值	默认值	说明
odps.text.option.gzip.input.enabled	true/false	false	打开/关闭读压缩
odps.text.option.gzip.output.enabled	true/false	false	打开/关闭写压缩
odps.text.option.header.lines.count	非负整数	0	跳过文本文件头N行
odps.text.option.null.indicator	字符串	空字符串	在解析或者写出NULL值时代表NULL的字符串
odps.text.option.ignore.empty.lines	true/false	true	是否忽略空行
odps.text.option.encoding	UTF-8/UTF-16/US-ASCII	UTF-8	指定文本的字符编码

LOCATION参数，格式为：oss://oss-cn-shanghai-internal.aliyuncs.com/Bucket名称/目录名称，用户可以通过图形对话框选择获得OSS目录地址，目录后不要加文件名称。

DDL模式创建的表会出现在表管理的表节点树下，可以通过修改其一级、二级主题来调整出现位置。

2. OTS外部表

OTS外部表建表语句如下。

```
CREATE EXTERNAL TABLE IF NOT EXISTS ots_table_external(
odps_orderkey bigint,
odps_orderdate string,
```

```
odps_custkey bigint,
odps_orderstatus string,
odps_totalprice double
)
STORED BY 'com.aliyun.odps.TableStoreStorageHandler'
WITH SERDEPROPERTIES (
'tablestore.columns.mapping'=':o_orderkey,:o_orderdate,o_custkey,
o_orderstatus,o_totalprice', -- (3)
'tablestore.table.name'='ots_tpch_orders'
'odps.properties.rolearn'='acs:ram::xxxxx:role/aliyunodpsdefaultrole
'
)
LOCATION 'tablestore://odps-ots-dev.cn-shanghai.ots-internal.
aliyuncs.com';
```

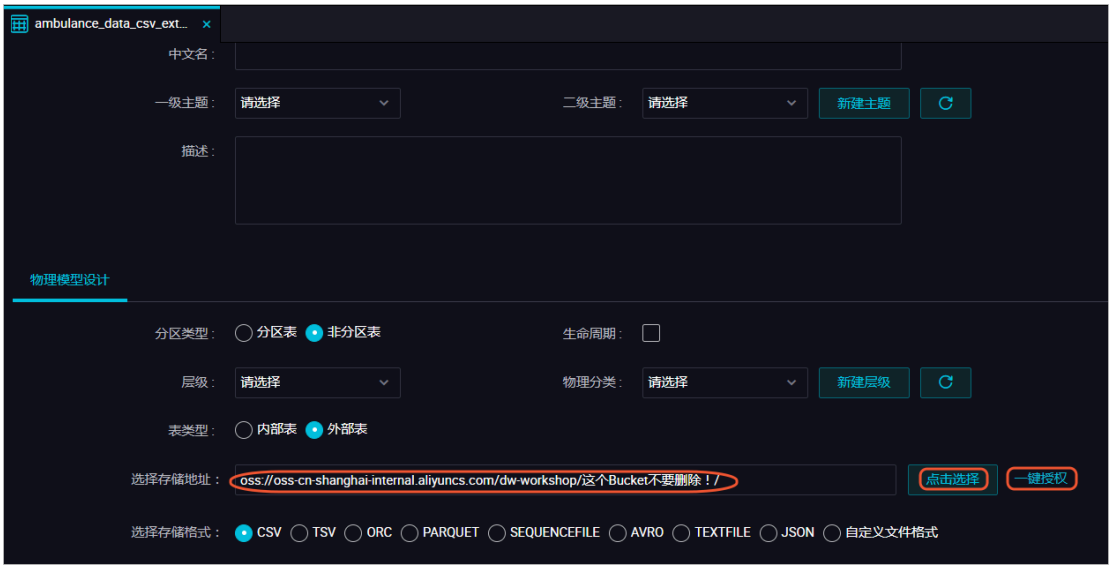
参数说明：

- com.aliyun.odps.TableStoreStorageHandler**是MaxCompute内置的处理TableStore数据的StorageHandler**
- SERDEPROPERTIES**是提供参数选项的接口，在使用TableStoreStorageHandler时，有两个必须指定的选项：tablestore.columns.mapping和 tablestore.table.name。**
  - tablestore.columns.mapping**：必选项，用来描述MaxCompute将访问的Table Store表的列，包括主键和属性 以:打头的用来表示Table Store主键，例如此语句中的:o_orderkey和:o_orderdate，其他的均为属性列。 Table Store支持1-4个主键，主键类型为String、Integer和Binary，其中第一个主键为分区键。 在指定映射时，您必须提供指定Table Store表的所有主键，对于属性列则没有必要全部提供，可以只提供需要通过 MaxCompute来访问的属性列。**
  - tablestore.table.name**：需要访问的Table Store表名。如果指定的Table Store表名错误（不存在），则会报错， MaxCompute不会主动去创建Table Store表。**
- LOCATION**：用来指定Table Storeinstance名字、endpoint等具体信息。**

3. 图形化建表

进入数据开发页面，参见#unique_284进行图形化建表。外部表具有如下属性：

- 基本属性
  - 英文表名（在新建表时输入）
  - 中文表名
  - 一级、二级主题
  - 描述
- 物理模型设计
  - 表类型：请选择为外部表
  - 分区类型：OTS类型外部表不支持分区
  - 选择存储地址：即LOCATION参数。您可以在物理模型设计栏中设置LOCATION参数，如下图所示，该参数可在图形化界面点击选择，完成后可直接进行一键授权。



  - 选择存储格式：根据业务需求进行选择，支持CSV、TSV、ORC、PARQUET、SEQUENCEFILE、RCFILE、AVRO、TEXTFILE和

自定义文件格式。如果您选择了自定义文件格式，需要选择自定义的资源。在提交 资源时，可以自动解析出其包含的类名并可以供用户选取。

- rolearn：如果STS已授权，可不进行填写

· 表结构设计

配置	说明
字段类型	MaxCompute 2.0支持TINYINT、SMALLINT、INT、BIGINT、VARCHAR和STRING类型。
操作	支持新增，修改，删除。
长度/设置	对于VARCHAR类型，可以支持设置长度。对于复杂类型可以直接填写复杂类型的定义。

支持字段类型

外部表支持的简单字段类型如下表所示。外部表支持的复杂字段类型如下表所示。

类型	是否新增	格式举例	描述
TINYINT	是	1Y，-127Y	8位有符号整形，范围 -128 到 127
SMALLINT	是	32767S， -100S	16 位有符号整形， 范围 -32768 到 32767
INT	是	1000，-15645787	32位有符号整形，范围-231到231 -1
BIGINT	否	100000000000L， -1L	64位有符号整形, 范围-263 + 1到263 - 1
FLOAT	是	无	32位二进制浮点型
DOUBLE	否	3.1415926 1E+7	8字节双精度浮点数,64位二进制浮点型
DECIMAL	否	3.5BD， 99999999999.9999999BD	10 进制精确数字类型，整形部分范围-1036+1到1036-1, 小数部分精确到 10-18

类型	是否新增	格式举例	描述
VARCHAR(n)	是	无	变长字符类型，n为长度，取值范围 1 到 65535
STRING	否	“abc”，’ bcd ’，” alibaba”	字符串类型，目前长度限制为 8M
BINARY	是	无	二进制数据类型，目前长度限制为 8M
DATETIME	否	DATETIME ‘2017-11-11 00:00:00’	日期时间类型，使用东八区时间作为系统标准时间。范围从0000年1月1日到9999年12月31日，精确到毫秒
TIMESTAMP	是	TIMESTAMP ‘2017-11-11 00:00:00.123456789’	与时区无关的时间戳类型，范围从0000年1月1日到9999年12月31日 23.59:59.999999999, 精确到纳秒
BOOLEAN	否	TRUE，FALSE	boolean 类型, 取值 TRUE 或 FALSE

类型	定义方法	构造方法
ARRAY	array< int >; array< struct< a:int, b:string >>	array(1, 2, 3); array(array(1, 2); array(3, 4))
MAP	map< string, string >; map< smallint, array< string>>	map(“k1”，“v1”，“k2”，“v2”); map(1S, array(‘a’，‘b’), 2S, array(‘x’，‘y’))
STRUCT	struct< x:int, y:int>; struct< field1:bigint, field2:array< int>, field3:map< int, int>>	named_struct(‘x’ , 1, ‘y’ , 2); named_struct(‘field1’ , 100L, ‘field2’ , array(1, 2), ‘field3’ , map(1, 100, 2, 200)

如需使用MaxCompute 2.0支持的新数据类型（TINYINT、SMALLINT、 INT、FLOAT、VARCHAR、TIMESTAMP 、BINARY或复杂类型），需在建表语句前加上语句 `set odps.sql.type.system.odps2=true;`set语句和建表语句一起提交执行。如需兼容HIVE，建议加上语句`odps.sql.hive.compatible=true;`。

查看和处理外部表

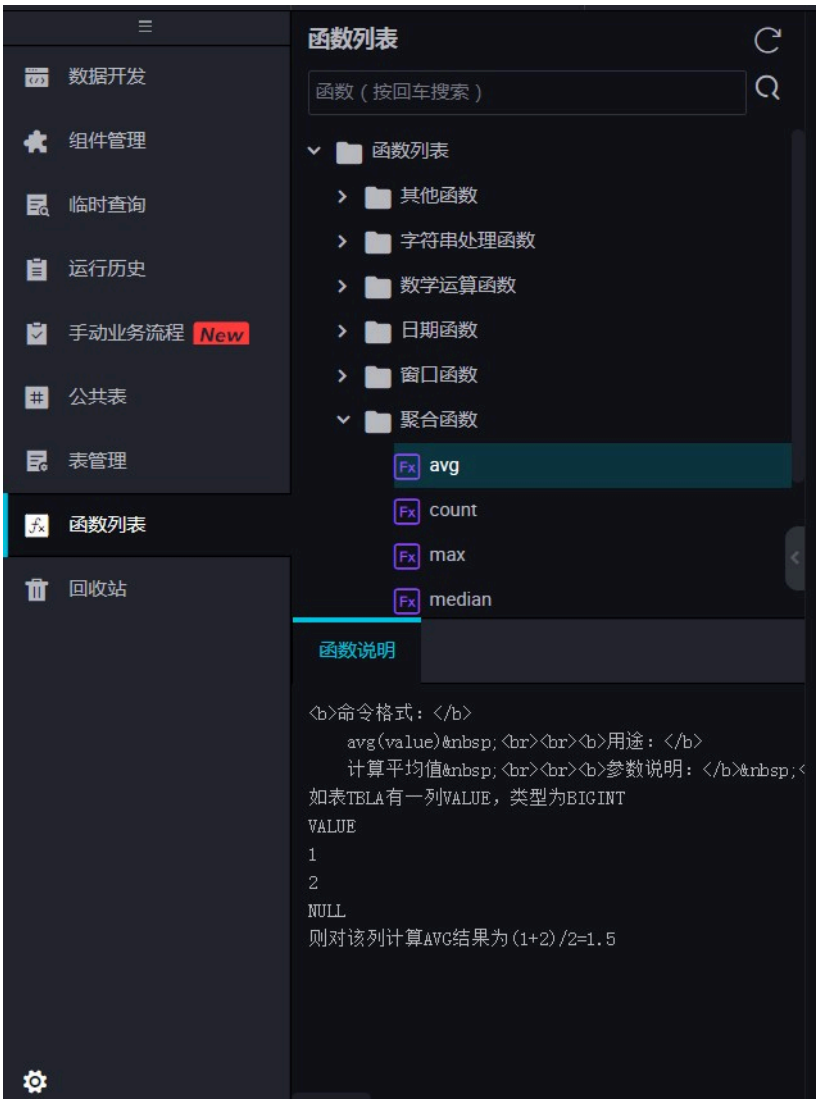
外部表在表管理和#unique_428中可以查询。

处理外部表的方式与内部表基本相同。

3.18 函数列表

函数列表会展示当前可以使用的函数，函数的分类，及函数的使用说明和实例。

函数列表中包括：其他函数、字符串处理函数、数学运算函数、日期函数、窗口函数、聚合函数共计6个部分。这些函数为系统自带的函数，可以通过拖动函数说明，查看函数的说明及使用示例。



3.19 MaxCompute资源

您可通过MaxCompute资源面板，查看在MaxCompute计算引擎中存在的资源、资源的变更历史，并可将资源文件一键添加到数据开发面板的业务流程中。

查看资源

1. **您可通过MaxCompute资源面板，查看在MaxCompute计算引擎对应项目中存在的资源。**

序号	图标	说明
1		单击后，可在弹出的面板中切换工作空间下的项目。 说明: 简单模式的工作空间只有生产环境项目。
2		单击后，可以筛选不同的资源类型。
3		单击后，可以切换条目排序，默认按更新时间倒序排列。

说明:
目前支持FILE、JAR、PYTHON和ARCHIVE四种资源类型。

2. 选中相应的资源，即可查看资源名称、资源类型和资源大小等详细信息。

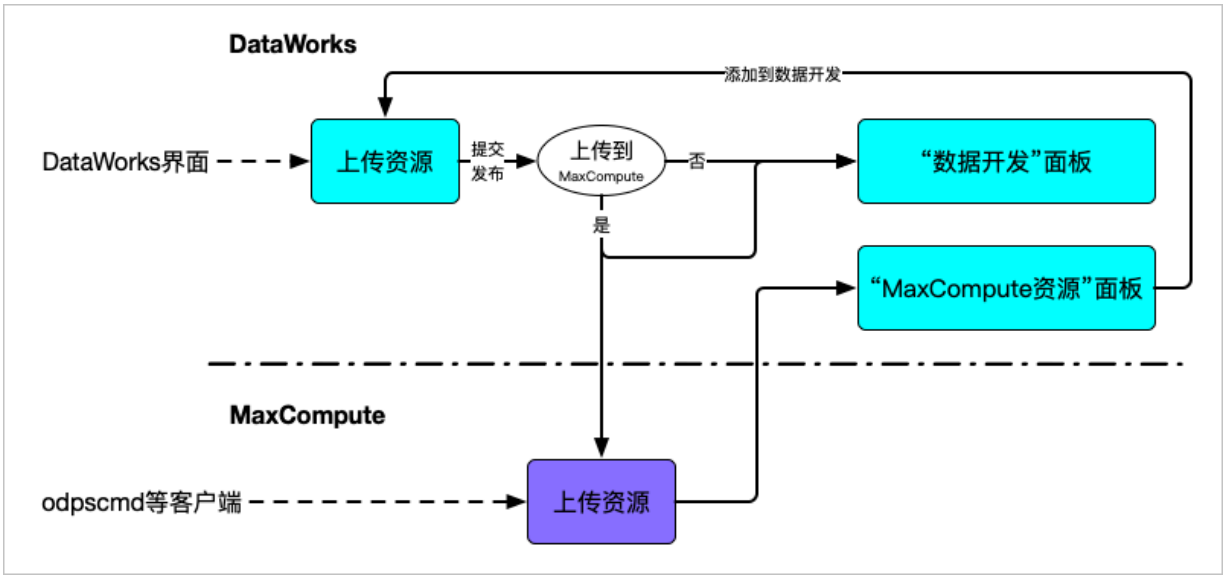
说明:

- MaxCompute资源面板中所列的资源，并非一定与数据开发面板中的资源一致。
  - 数据开发面板中的资源只有同时上传到MaxCompute，并提交/发布后，才会出现在MaxCompute资源面板的开发/生产环境中。
  - 通过odpscmd、MaxCompute Studio等非DataWorks渠道上传的资源文件，不会在数据开发面板显示，但会出现在MaxCompute资源面板中。
- 使用MaxCompute资源面板中所列的资源时，需注意与数据开发面板中资源的区别。

使用场景	数据开发	MaxCompute资源
在ODPS SQL节点中使用	是（需同时上传到ODPS）	是
在ODPS MR节点中使用	是（需同时上传到ODPS）	否
在Shell节点中使用	是	否
在临时查询中使用	是（需同时上传到ODPS）	是
在业务流程中创建函数	是（需同时上传到ODPS）	否

添加资源到数据开发面板

添加资源到数据开发面板的流程，如下图所示。



1. 找到需要的资源后，单击添加到数据开发。

   您可通过此操作，快速将MaxCompute资源面板中的的资源文件同步至数据开发的业务流程中。

2. 填写新建资源对话框中的配置。

上传过程中，您可以进行如下操作：

- 重命名资源名称。
- 选择目标文件夹，即修改资源所处的业务流程。

上传过程中，您不可以进行如下操作：

- 更改资源类型。
- 选择是否上传为ODPS资源。
- 重新上传文件。

3. 单击确定，即可完成资源的创建。

说明：

- 创建完成后，您需要手动完成保存、提交、发布等操作，与业务流程中对资源进行的操作相同。
- 资源提交、发布过程中，会同样上传到开发、生产环境的MaxCompute，同时更新您在MaxCompute资源面板中的资源文件。
- 由于资源在MaxCompute项目中的唯一性，如果在同一项目中有同名资源，添加过程将会覆盖原有函数。如果原有函数处于不同业务流程，将会在新业务流程下完成覆盖。

查看资源的变更历史

1. 单击查看变更历史。

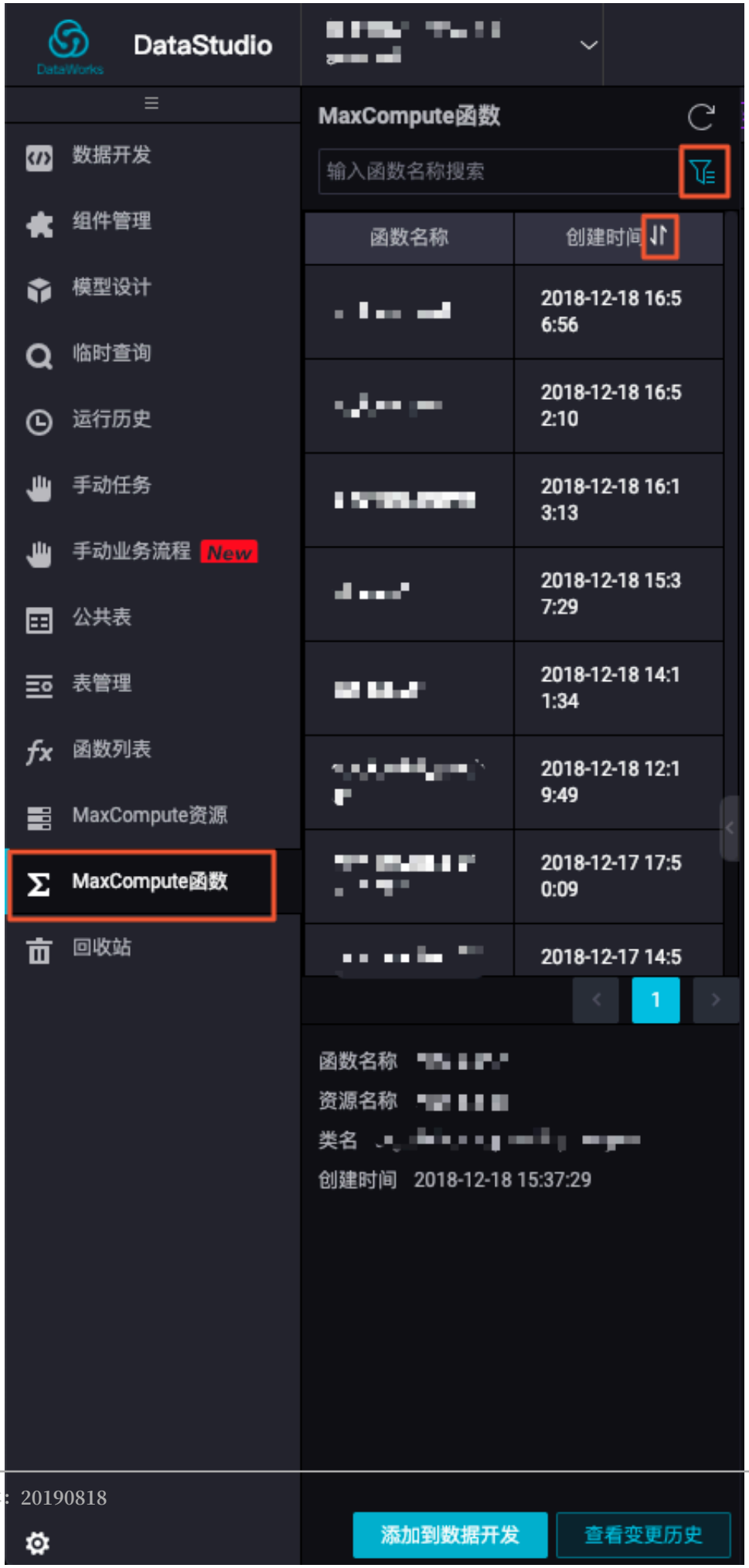
2. 查看资源文件的创建、修改记录。

3.20 MaxCompute函数

MaxCompute函数面板可以用来查看在MaxCompute计算引擎中存在的函数（UDF）、回溯函数的变更历史，以及将函数一键添加到数据开发面板的业务流程中。

查看函数

1. 您可通过MaxCompute函数面板，查看在MaxCompute计算引擎对应项目中存在的自定义函数（UDF）。
   - 单击  按钮，可在弹出的面板中切换工作空间下的项目（简单模式的工作空间只有生产环境项目）。
   - 单击  按钮，可以切换条目排序，默认按照创建时间倒序排列。


DataStudio
数据开发
组件管理
模型设计
临时查询
运行历史
手动任务
手动业务流程 New
公共表
表管理
函数列表
MaxCompute资源
MaxCompute函数
回收站
MaxCompute函数
输入函数名称搜索
函数名称
创建时间
2018-12-18 16:56:56
2018-12-18 16:52:10
2018-12-18 16:13:13
2018-12-18 15:37:29
2018-12-18 14:11:34
2018-12-18 12:19:49
2018-12-17 17:50:09
2018-12-17 14:5
1
函数名称
资源名称
类名
创建时间 2018-12-18 15:37:29
添加到数据开发
查看变更历史

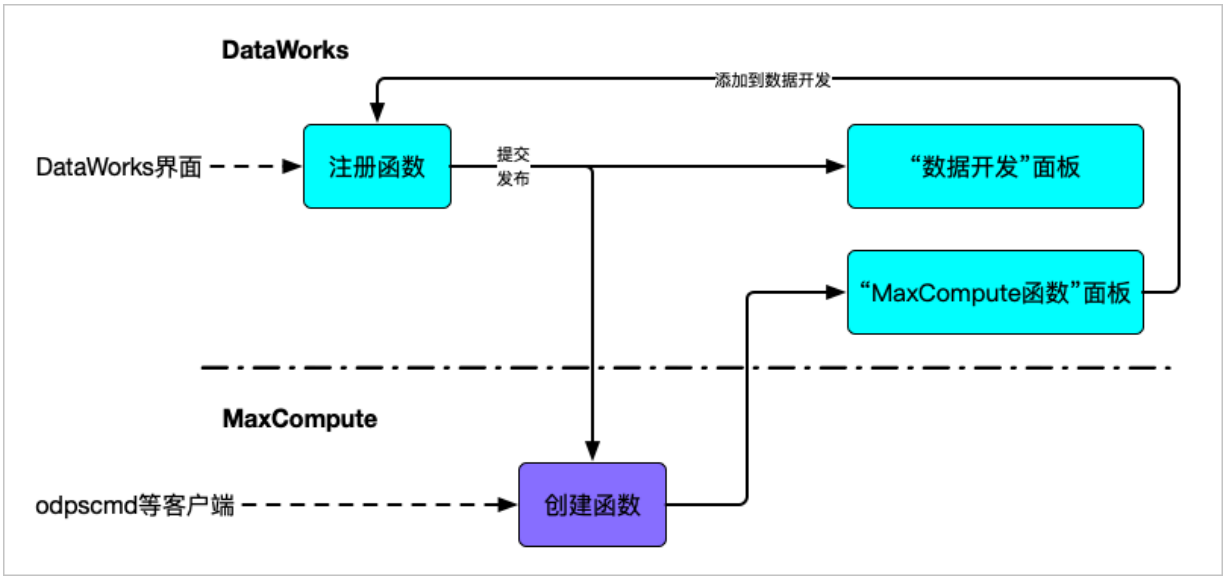
2. 选中某项函数，即可查看其详细信息。

删除函数

如果您需要删除函数，可切换至数据开发面板，右键单击相应业务流程下的函数名称，选择删除。

添加函数到数据开发面板

添加函数到数据开发面板的流程，如下图所示。



操作步骤

1. **进入MaxCompute函数面板，单击添加到数据开发。可以快速将MaxCompute函数面板的函数同步到数据开发面板的业务流程中。**

2. 在新建函数对话框中填写函数名称，并选择目标文件夹。

在上传过程中，您可以重命名函数名称、选择目标目标文件夹（即修改函数所处的业务流程）。但不可以修改函数定义。

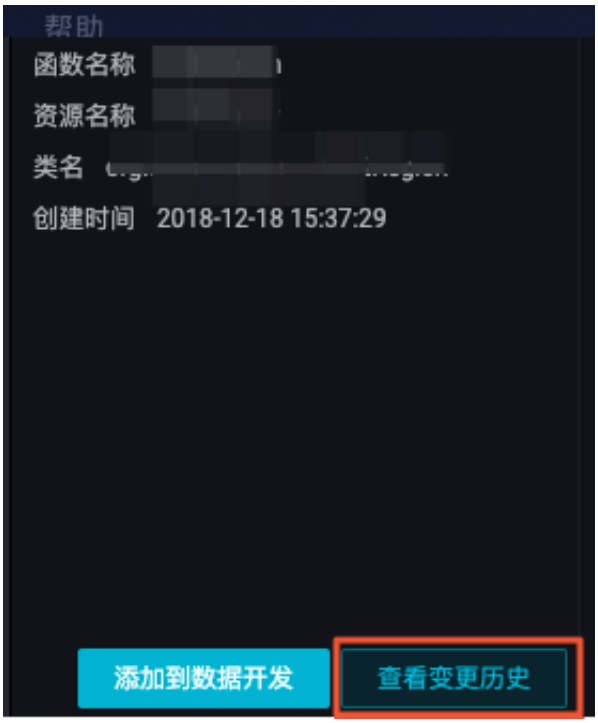
3. 单击提交。

说明：

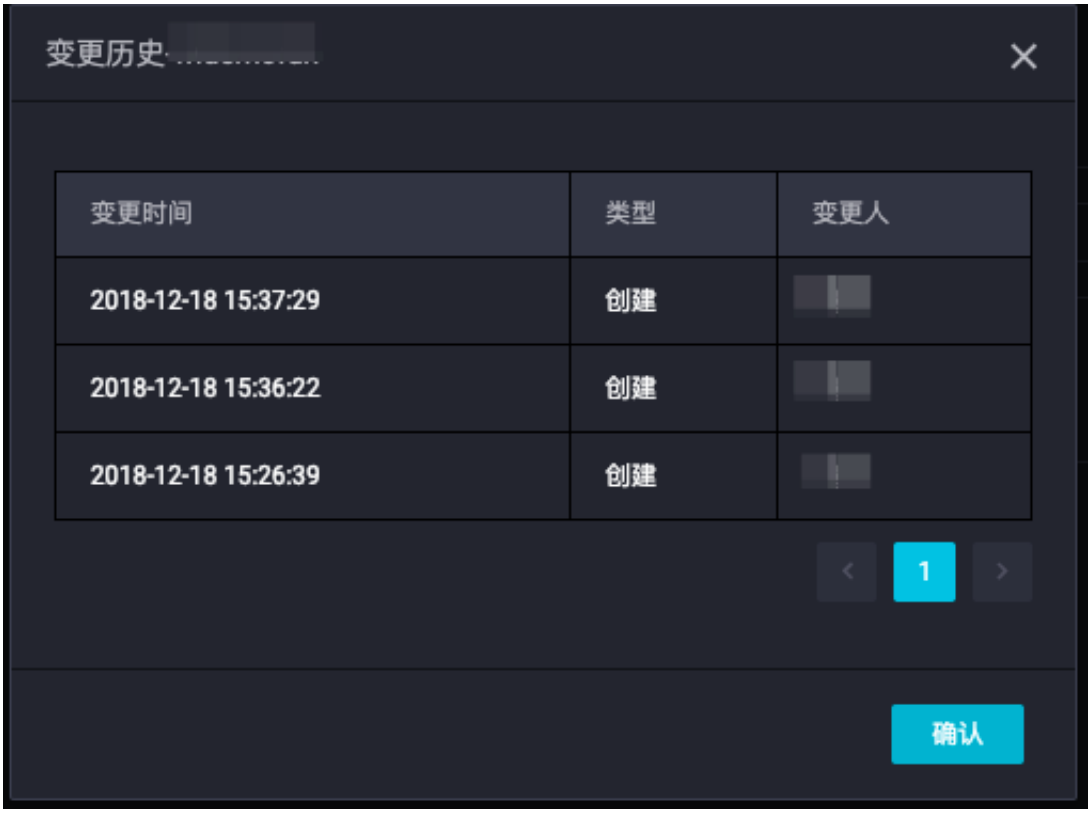
- 创建完成后，您还需要手动完成保存、提交、发布等过程，与在业务流程中的注册函数相同。
- 函数提交、发布过程中，会同样上传到开发、生产环境的MaxCompute，也会同时更新您在MaxCompute函数面板中的自定义函数。
- 由于资源在MaxCompute项目中的唯一性，如果在同一项目中有同名资源，添加过程将会覆盖原有函数。若原有函数处于不同业务流程，将会在新业务流程下完成覆盖。

查看函数的变更历史

1. 单击查看变更历史。



2. 查看函数的创建、修改记录。



说明:

在MaxCompute中，函数无法被修改，因此对于函数的历史操作类型统一为创建。

3.21 编辑器快捷键列表

代码编辑器和常用快捷键。

Windows的Chrome版本下

`Ctrl + S` 保存

`Ctrl + Z` 撤销

`Ctrl + Y` 重做

`Ctrl + D` 同词选择

`Ctrl + X` 剪切一行

`Ctrl + Shift + K` 删除一行

`Ctrl + C` 复制当前行

`Ctrl +i` 选择行

`Shift + Alt +` 鼠标拖动 列模式编辑,修改一整块内容

`Alt +` 鼠标点选 多列模式编辑,多行缩进

`Ctrl + Shift + L` 为所有相同的字符串实例添加光标,批量修改

`Ctrl + F` 查找

`Ctrl + H` 替换

`Ctrl + G` 定位到指定行

`Alt + Enter` 选中所有查找匹配上的关键字

`Alt↓ / Alt↑` 下／上移动当前行

`Shift + Alt + ↓ / Shift + Alt + ↑` 下／上复制当前行

`Shift + Ctrl + K` 删除当前行

`Ctrl + Enter / Shift + Ctrl + Enter` 光标移入下／上一行

`Shift + Ctrl + \` 光标跳到匹配的括号

`Ctrl + ] / Ctrl + [` 增加／减小缩进

`Home / End` 移到当前行最前／最后

`Ctrl + Home / Ctrl + End` 移到当前文件最前／最后

`Ctrl + → /Ctrl + ←` 右／左按单词移动光标

`Shift + Ctrl + [ / Shift + Ctrl + ]` 折叠／展开光标在所在区域

`Ctrl + K + Ctrl + [ / Ctrl + K + Ctrl + ]` 折叠／展开光标所在区域子区域

`Ctrl + K + Ctrl + 0 / Ctrl + K + Ctrl + j` 折叠／展开所有区域

`Ctrl + /` 注释 / 解除注释 光标所在行或代码块

Mac的Chrome版本下

`cmd + S` 保存

`cmd + Z` 撤销

`cmd + Y` 重做

`Cmd+D` 同词选择

`cmd + X` 剪切一行

`Cmd+Shift+K` 删除一行

`cmd + C` 复制当前行

`cmd +i` 选择当前行

`cmd + F` 查找

`cmd + alt + F` 替换

`alt↓ / alt↑` 下／上移动当前行

`shift + alt + ↓ / shift + alt + ↑` 下／上复制当前行

`shift + cmd + K` 删除当前行

`cmd + Enter / shift + cmd + Enter` 光标移入下／上一行

`shift + cmd + \` 光标跳到匹配的括号

`cmd + ] / cmd + [` 增加／减小缩进

`cmd + ← / cmd + →` 移到当前行最前／最后

`cmd + ↑ / cmd + ↓` 移到当前文件最前／最后

alt + → /alt + ← 右／左安单词移动光标

alt + cmd + [/ alt + cmd +] 折叠／展开光标在所在区域

cmd + K + cmd + [/ cmd + K + cmd +] 折叠／展开光标所在区域子区域

cmd + K + cmd + 0 / cmd + K + cmd + j 折叠／展开所有区域

cmd + / 注释 / 解除注释 光标所在行或代码块

多光标／选择

alt + 点击鼠标 插入光标

alt + cmd + ↑/↓ 向上／下插入光标

cmd + U 撤消最后一个光标操作

shift + alt + I 向选中的代码块每一行最后插入光标

cmd + G／shift + cmd + G 查找下／上一个

cmd + F2 选中所有鼠标已选择的字符

shift + cmd + L 选中所有鼠标已选择部分

alt + Enter 选中所有查找匹配上的关键字

shift + alt + 拖拽鼠标 选择多列编辑

shift + alt + cmd + ↑ / ↓ 上下选择多列编辑

shift + alt + cmd + ← / → 左右选择多列编辑

3.22 回收站

DataWorks拥有自己的回收站，用于存放当前项目下所有删除的节点，您可以对节点进行恢复或彻底删除。

单击左侧导航栏中的回收站即可进入回收站页面。

您可在回收站中看到当前项目下所有删除的节点，右键单击选中的节点，可以选择还原此节点或者彻底删除此节点。

说明:

- **回收站仅显示100个节点，如果多于100个，只能彻底删除展示靠前的节点。**
- **组合节点被删除后不会显示在回收站中。**

单击右上角的我的文件，即可查看该项目被删除的节点。

说明:

如果在回收站彻底删除此节点，将无法恢复。回收站目前只能针对节点任务生效，不能回收针对业务流程、表、资源等。

4 运维中心

4.1 运维中心概述

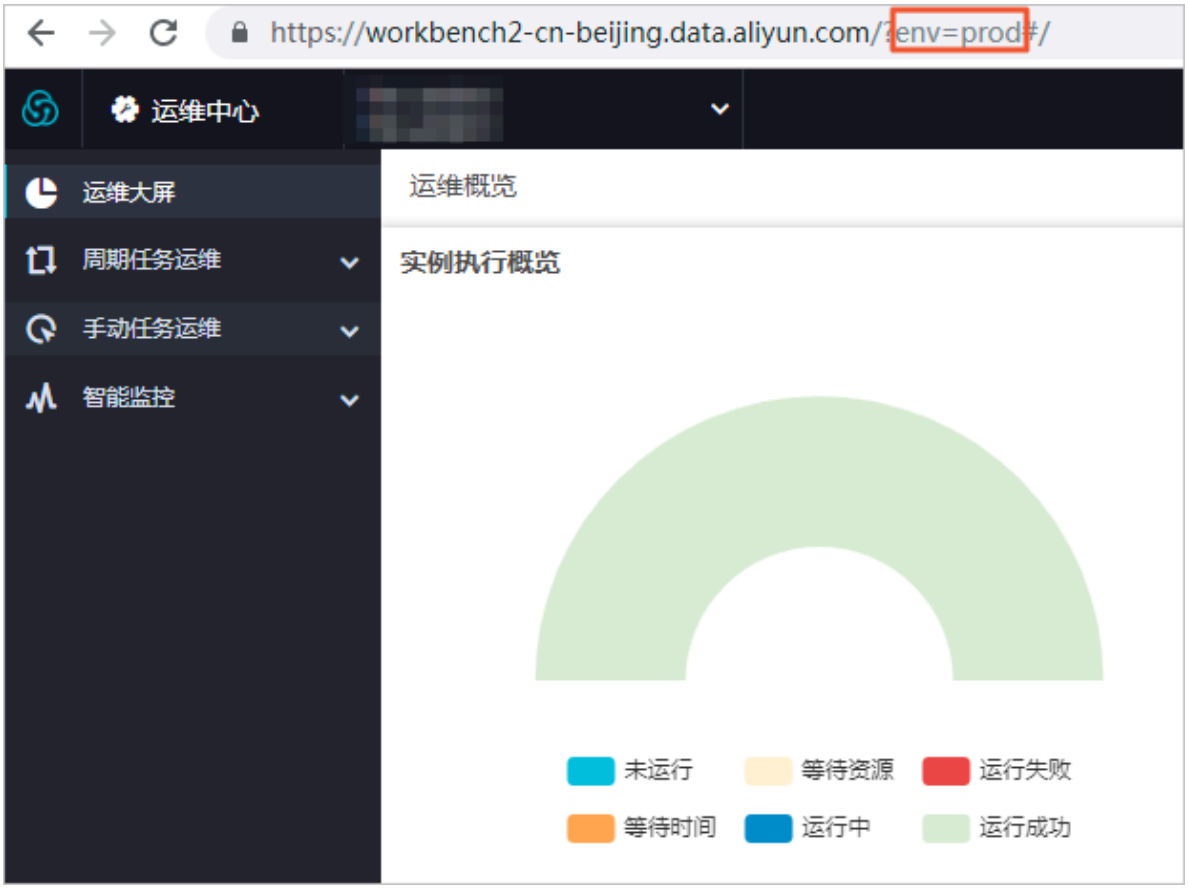
运维中心包括运维大屏、周期任务运维、手动任务运维和智能监控四大模块。

模块	说明
运维大屏	运维大屏主要对任务的运行情况进行报表展示。
周期任务运维	周期任务运维为您展示任务提交到调度系统后，经过调度系统运行后的生产实例，包括周期任务、周期实例、补数据实例和测试实例。
手动任务运维	任务运维为您展示任务提交到调度系统后，经过手动触发运行后的生产实例，包括手动任务和手动实例。
智能监控	智能监控主要对任务的运行情况进行监控。如果被监控的任务异常，您将会收到提示信息。详情请参见智能监控。

应用场景

- 您可以在运维中心查看您的任务和实例，并对展示的任务进行测试、补数据等操作。

- 如果您使用的是标准模式的工作空间，可以通过替换URL，切换生产环境和开发环境的运维中心。当`env=dev`时，为开发环境。当`env=prod`时，为生产环境。切换下图中的框内部分，即可跳转至对应环境的运维中心页面。



- 在任务运维中，您可以看到您所有任务的实例，可以对展示的实例进行终止、重跑等操作。

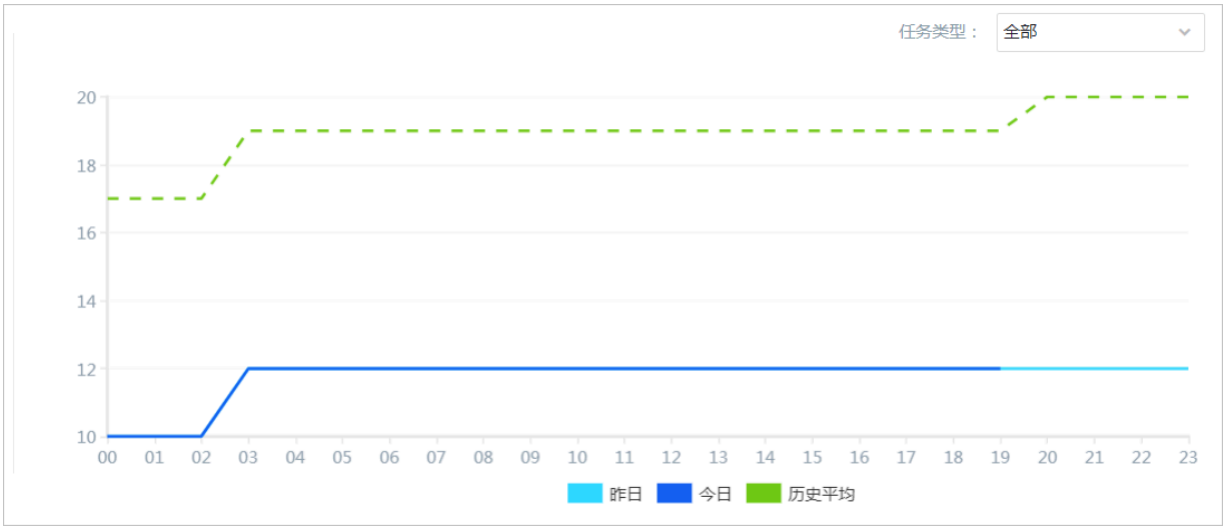
> **说明:**
> 实例是在调度系统中的任务经过调度系统后，触发运行生成的。实例代表了某个任务在某时某刻执行的一个快照，实例中会有任务的运行时间、运行状态、运行日志等信息。

4.2 运维大屏

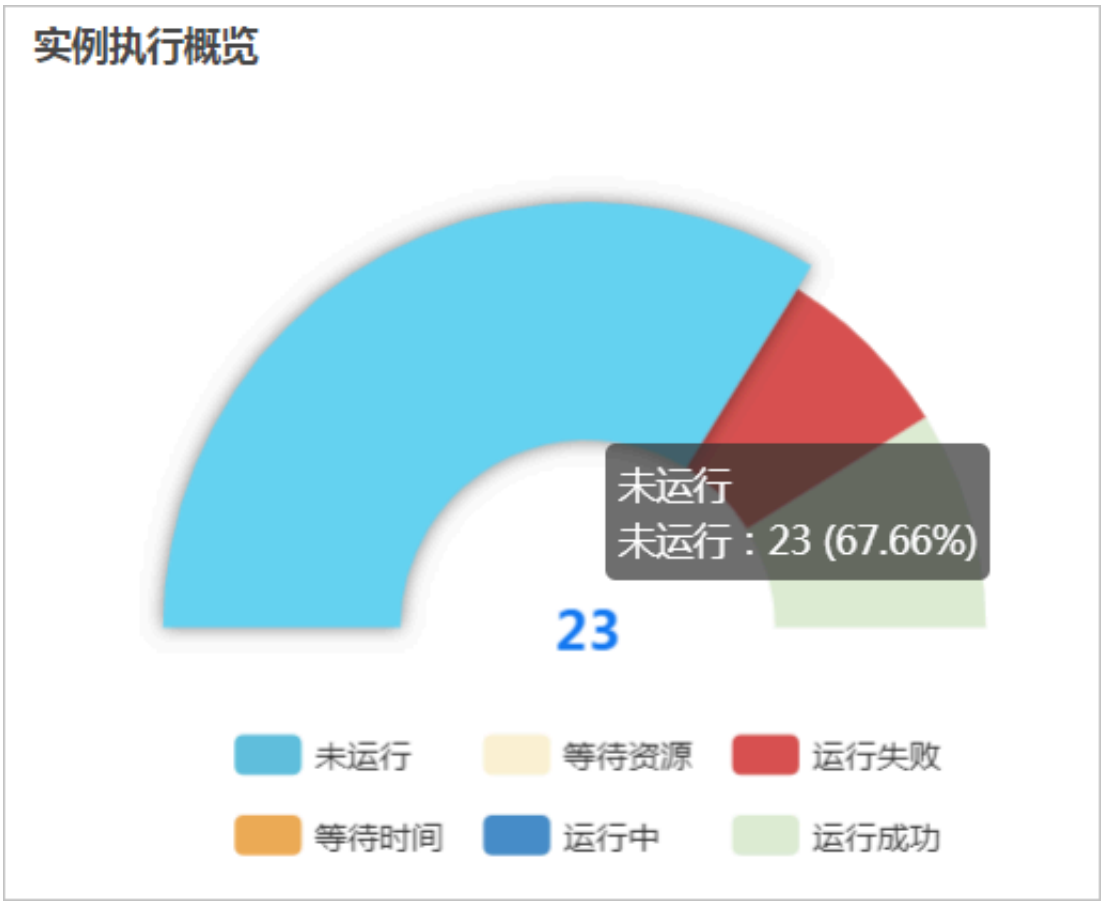
运维大屏可以帮助您从宏观上了解任务运行情况、调度任务数量趋势、任务节点执行时长和出错信息。

实例执行概览

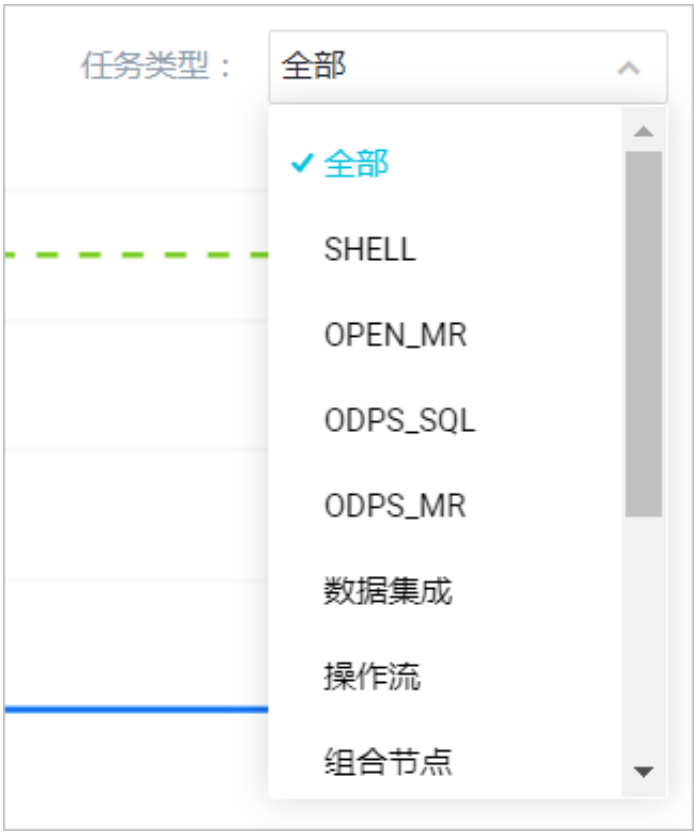
实例执行概览模块主要针对正常周期性调度今天、昨天与历史平均水平的任务完成情况进行对比统计。如果三条曲线偏移过多，则表示在某个时间段内有异常情况出现，需要进行进一步的检查与分析 。



如上述折线统计图所示，分别以三种不同颜色折线显示对当天00:00~23:00时间段内，当前工作空间中所有类型任务完成进度的统计，包括今天的任务完成情况、昨天的任务完成情况和历史平均水平的完成情况。您还可以通过左侧的总览图查看当前各类实例的数量和比例。

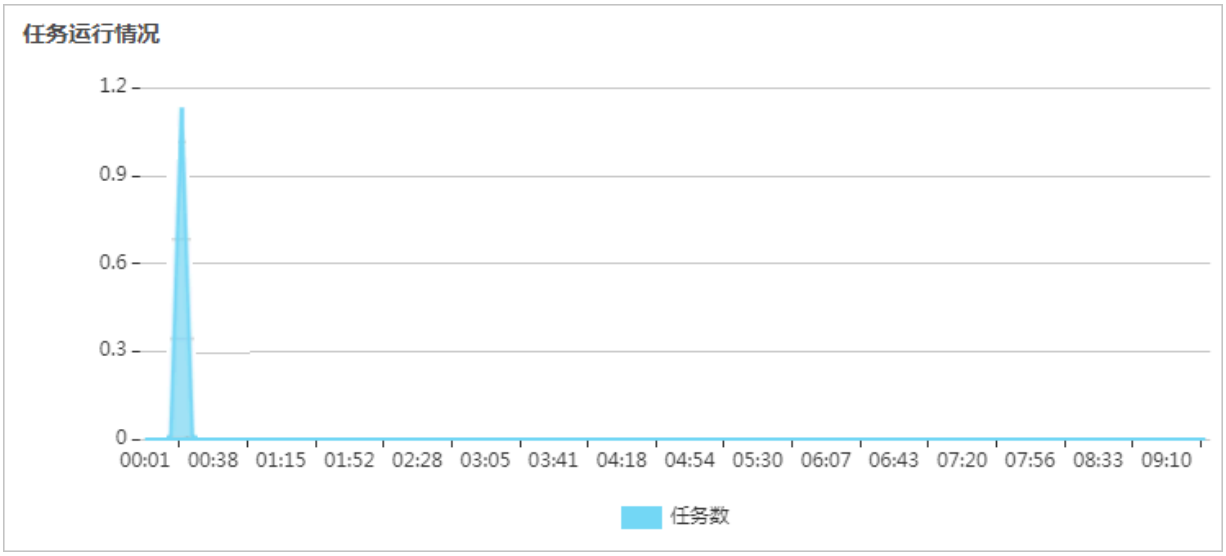


单击统计图右上角的任务类型，可以选择不同类型的任务进行查看。



任务运行情况

任务运行情况模块按照时间点展示当前正在运行的任务的数量，您可以整体查看到某个时间点的任务并发峰值数，以决定是否需要避开并发高峰期，便于及时调整调度的运行时间 。



任务执行时长排行

任务执行时长排行模块为您展示当前工作空间在业务日期内任务执行时长的排行榜单，默认按照执行时长由长到短的顺序排出前十名。您可以查看具体的任务名称、责任人和执行时长 。

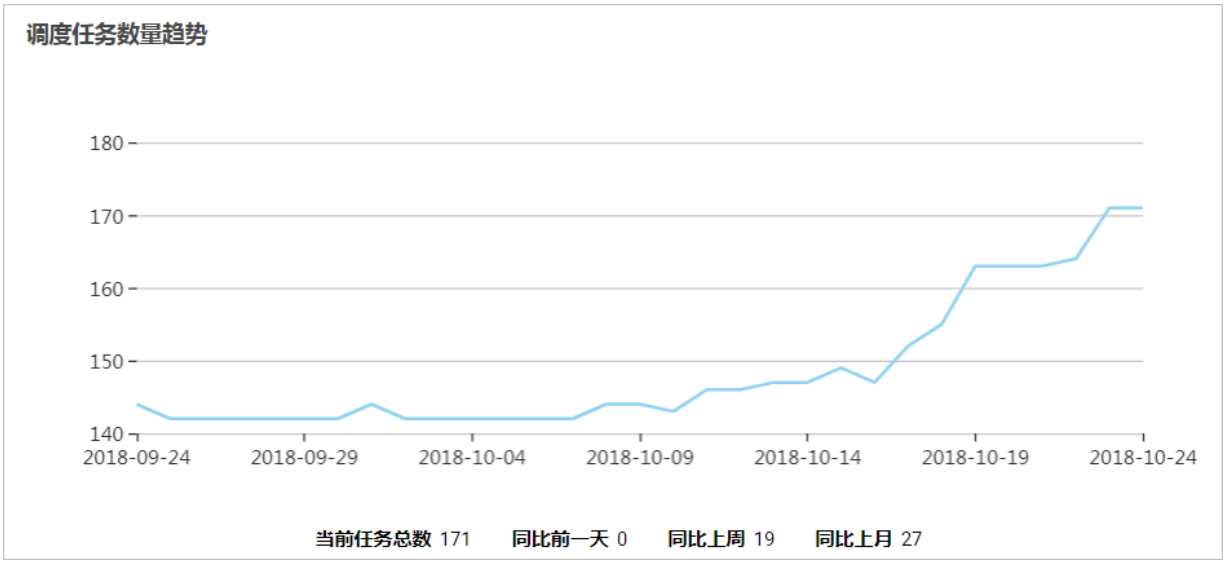
近一月出错排行

近一月出错排行模块主要对最近一个月的任务出错情况进行统计，显示任务出错次数排行榜的前十名。您可以查看任务名称、责任人和出错次数。

单击具体某个节点ID，可以跳转至任务详情页。

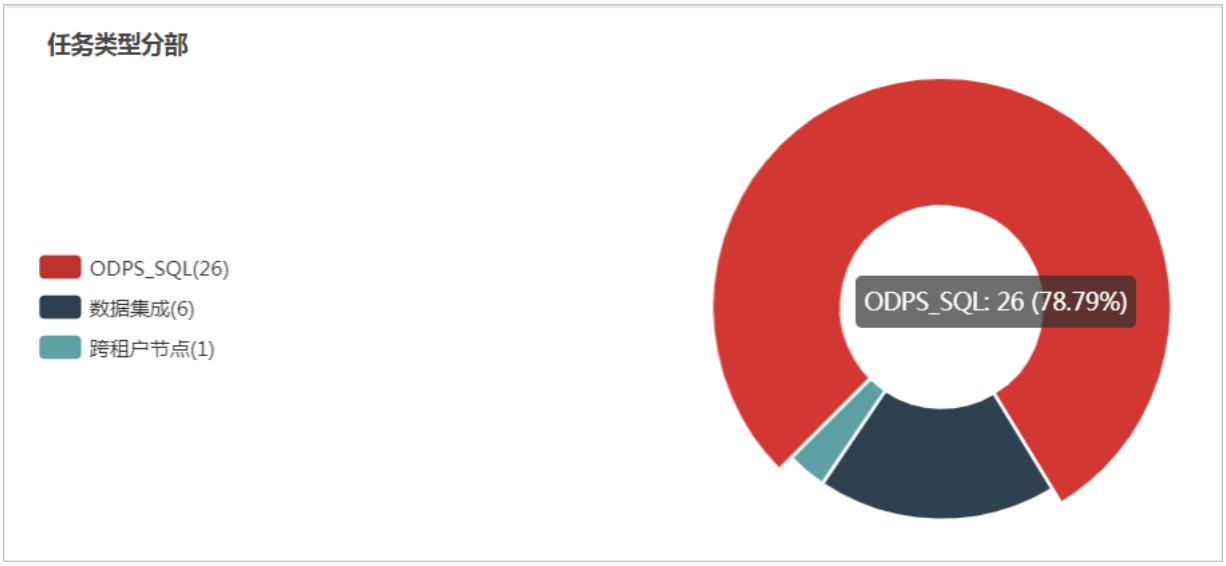
调度任务数量趋势

调度任务数量趋势模块为您展示当前任务总数，同比昨天任务数量的浮动情况、同比上周任务数量的浮动情况、同比上个月任务数量的浮动情况。



任务类型分布

鼠标放在某一扇形区域上，可以显示该任务类型的具体任务数量和占比。



4.3 周期任务运维

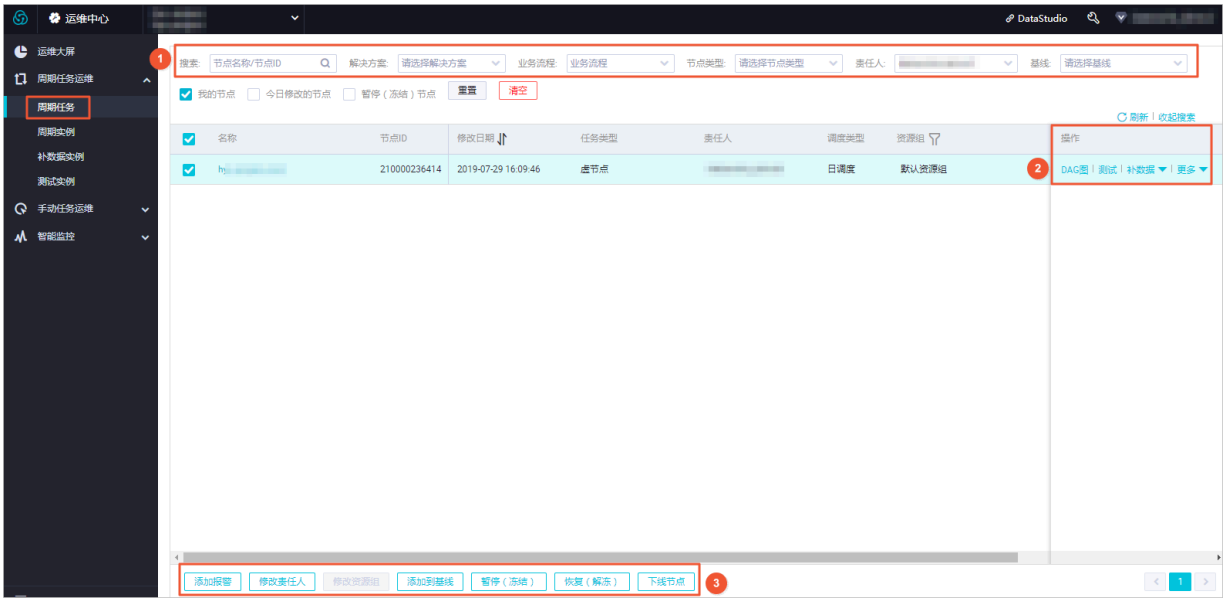
4.3.1 周期任务

周期任务是指调度系统按照调度配置自动定时执行的任务。

> 说明:
>
> - 周期任务列表默认展示当前登录账号下的工作流任务。
> - 任务提交后，将会在第二天23:30自动生成实例来运行任务。如果是在23:30以后提交的任务，则第三天才会开始生成实例来自动运行任务。
> - 请勿操作project_etl_start节点，此节点为项目根节点，周期任务的实例均依赖于此节点。如果冻结此节点，周期任务实例将不会运行。

周期任务列表

周期任务页面以列表的形式展示已提交的周期任务。

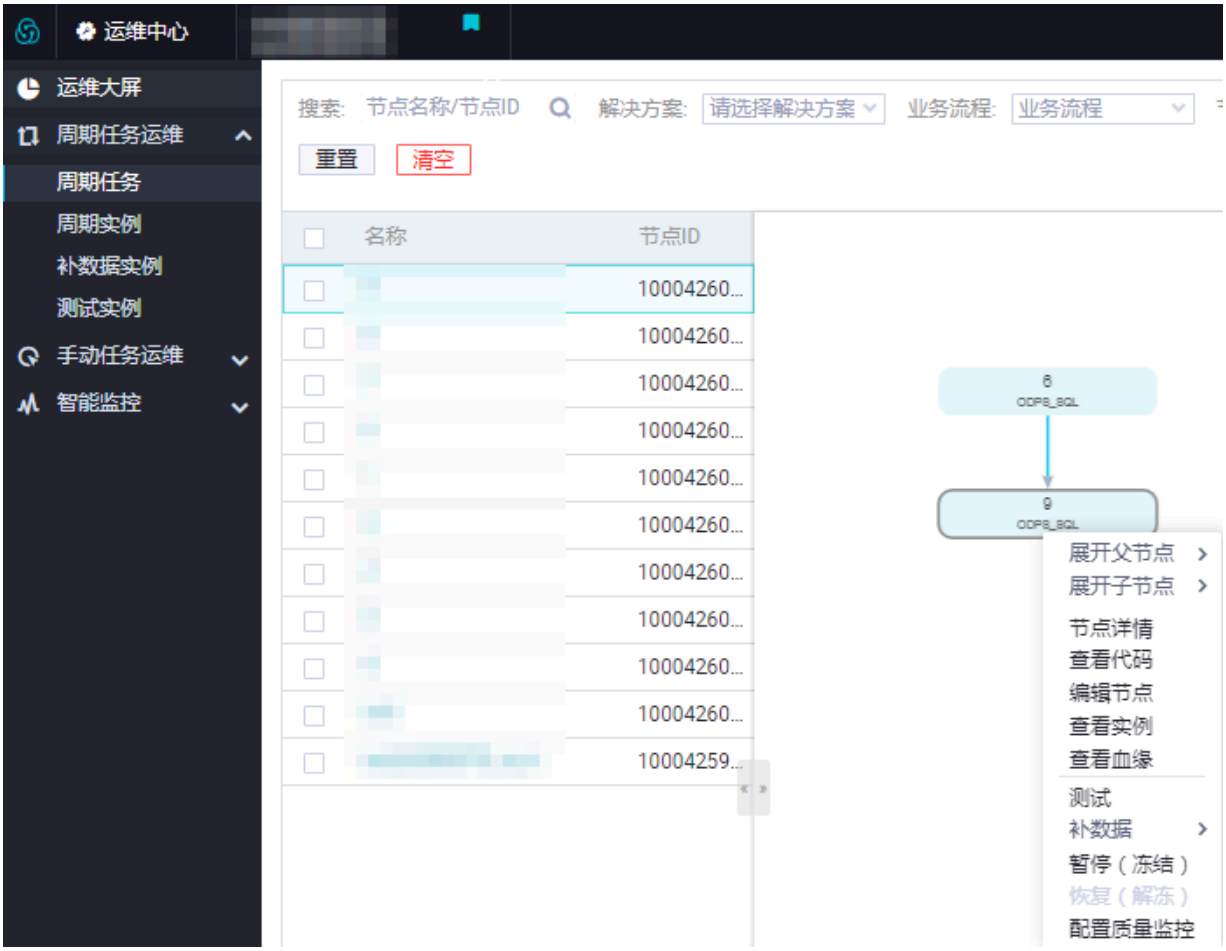


操作	说明
筛选	如上图中的模块1，通过筛选条件过滤出要查询的任务。您可以根据节点名称、解决方案、业务流程、节点类型、责任人、基线、今日修改的节点等条件进行精确筛选。 **说明:** 任务名搜索的结果，会受到其他筛选条件的影响，只有同时满足所有筛选条件的结果才会展示出来。
DAG图	单击操作栏中的DAG图，即可打开此节点的DAG图。您可以在DAG图中查看节点的属性、操作日志、代码等信息。
测试	单击操作栏中的测试，即可对当前节点进行测试，详情请参见#unique_440。
补数据	单击操作栏中的补数据，即可对当前节点进行补数据，详情请参见#unique_441。

操作	说明
更多	单击操作栏中的更多，可以进行暂停、恢复、查看实例等更多操作。 · 单击暂停（冻结），即可将当前节点置为暂停（冻结）状态，并停止调度。当节点状态为暂停时，在节点名称后会出现 图标。 · 单击恢复（解冻），即可将暂停（冻结）的节点恢复调度。 · 单击查看实例，即可查看此节点的周期实例。 · 单击添加报警，即可为节点配置报警。 · 单击修改责任人，即可修改节点责任人。 · 单击添加到基线，即可将当前节点添加到基线。 · 如果工作空间存在多个资源组，单击修改资源组，即可修改节点的资源组。 · 单击配置质量监控，即可配置数据质量，对数据进行校验。 · 单击查看血缘，即可查看节点的血缘关系图。 · 单击上下游，即可跳转至节点基本信息页面，查看节点的上游列表和下游列表。
批量操作	如上图中的模块3，您可以批量选择任务，进行添加报警、修改责任人、修改资源组、添加到基线、暂停（冻结）、恢复（解冻）和下线节点等操作。

周期任务DAG图

单击任务名或操作栏中的DAG图，即可打开此节点的DAG图。您可以在DAG图中，右键单击节点，进行相关操作。



操作	说明
展开父节点/子节点	当一个工作流有3个节点及以上时，运维中心展示任务时会自动隐藏节点。您可以通过展开父子层级，来看到更多的节点依赖关系，层级越大，展示越全面。
节点详情	单击后，即可跳转至节点基本信息页面，查看当前节点的输入表、输出表、上游列表和下游列表等信息。
查看代码	查看当前节点的代码。
编辑节点	单击后，即可跳转至数据开发页面，对当前节点的内容进行修改。
查看实例	查看当前节点的周期实例。
查看血缘	查看当前节点的血缘关系图。
测试	单击后，您需要在冒烟测试对话框中，填写冒烟测试名称并选择业务日期，单击确定后，即可跳转至测试实例页面。
补数据	包括当前节点、当前节点及下游节点和海量节点模式。
暂停（冻结）	将当前节点置为暂停（冻结）状态，并停止调度。
恢复（解冻）	恢复暂停（冻结）的节点的调度。
配置质量监控	配置当前节点的数据质量，对数据进行校验。

4.3.2 周期实例

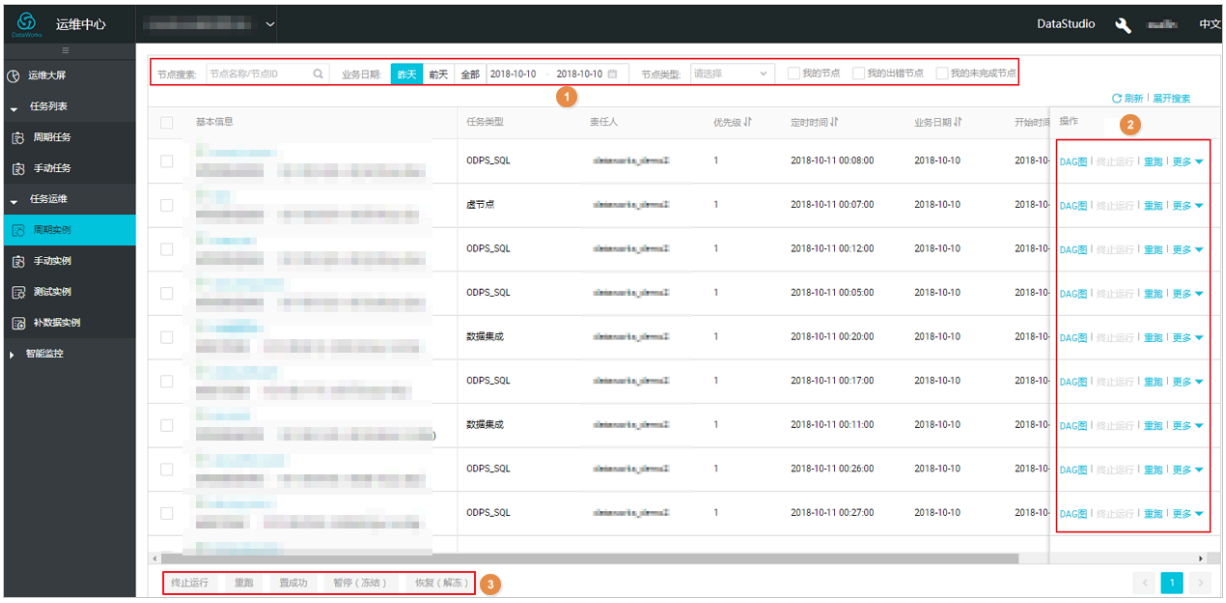
周期实例是周期任务达到启用调度所配置的周期性运行时间时，被自动调度的实例快照。

周期任务每调度一次，便生成一个实例工作流。您可以对已调度的实例任务进行日常的运维管理，如查看运行状态，对任务进行终止、重跑、解冻等操作。

> 说明:
>
> - 周期任务定时生成周期实例，实例会按最新的代码运行任务。如果您的任务在实例生成后修改了代码并重新提交发布，则未运行的实例会拉取最新的代码运行任务。
> - 如果任务失败未报警，请首先检查是否已在个人信息页面配置了您的手机号码与邮箱地址。

周期实例列表

周期实例列表以列表形式对被调度的任务进行运维及管理，包括检查运行日志、重跑任务、终止正在运行的任务等。

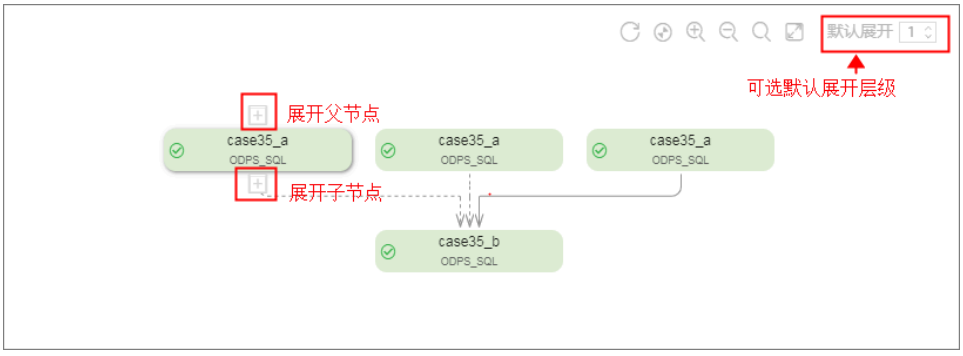


操作	说明
筛选	如上图中的模块1，有丰富的筛选条件，默认筛选业务日期是当前时间前一天的工作流任务。您可添加节点名称、业务日期、节点类型等条件进行更精确的筛选。
终止运行	只可对等待运行、运行中状态的实例进行终止运行操作，进行此操作后，该实例将为失败状态。
重跑	可以重跑某任务，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理出错节点和漏跑节点。 说明: 只能重跑未运行、成功、失败状态的任务。

操作	说明
重跑下游	可以重跑某任务及其下游任务，需要您自定义勾选，勾选的任务将被重跑，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理数据修复。 说明: 只能勾选未运行、完成、失败状态的任务，如果勾选了其他状态的任务，页面会提示已选节点中包含不符合运行条件的节点，并禁止提交运行。
置成功	将当前节点状态改为成功，并运行下游未运行状态的任务。常用于处理出错节点。 说明: 只有失败状态的任务能被置成功，工作流任务不能置成功。
冻结	周期实例中的冻结只针对当前实例，且正在运行中的实例。
解冻	可以将冻结状态的实例解冻。 · 如果该实例还未运行，则上游任务运行完毕后，会自动运行。 · 如果上游任务都运行完毕，则该任务会直接被置为失败，需要手动重跑后，实例才会正常运行。
批量操作	如上图中的模块3，批量操作包括：终止运行、重跑、置成功、冻结和解冻5个功能。

实例DAG图

单击实例名或操作栏中的DAG图，即可打开该实例的DAG图。您可以在DAG图中，右键单击实例，进行相关操作。

操作	说明
展开父节点/子节点	当一个工作流有3个节点及以上时，运维中心展示任务时会自动隐藏节点。您可以通过展开父子层级，来看到全部节点的内容。 
查看运行日志	查看当前实例正在运行、成功、失败等状态的运行日志。
查看代码	查看当前实例的代码。
编辑节点	单击后，即可跳转至数据开发页面，对当前节点的内容进行修改。

操作	说明
查看节点影响	单击后，即可跳转至实例基本信息页面，查看当前实例的基本信息、影响基线信息和运行信息。
查看血缘	查看当前实例的血缘关系。
查看更多详情	单击后，即可查看当前节点的属性、上下文、运行日志、操作日志和代码等信息。
终止运行	仅等待运行、运行中状态的实例可以进行终止运行的操作。进行此操作后，该实例将为失败状态。
重跑	可以重跑某任务，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理出错节点和漏跑节点。  说明: 只能重跑未运行、成功、失败状态的任务。
重跑下游	可以重跑某任务及其下游任务，需要您自定义勾选，勾选的任务将被重跑，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理数据修复。  说明: 只能勾选未运行、完成、失败状态的任务，如果勾选了其他状态的任务，页面会提示已选节点中包含不符合运行条件的节点，并禁止提交运行。
置成功	将当前实例的状态改为成功，并运行下游未运行状态的任务。常用于处理出错节点。  说明: 只有失败状态的任务能被置成功，工作流任务不能置成功。
续跑	任务执行失败后，可以续跑此任务。
紧急操作	当前实例在非常紧急的情况下的操作，紧急操作只对当前节点本次有效。 选择去除依赖，即可解除当前节点的依赖关系。常用于上游失败并与此实例没有数据关系时，启动此节点。
暂停（冻结）	周期实例中的冻结仅针对当前实例，且正在运行中的实例。
恢复（解冻）	可以将冻结状态的实例解冻。 · 如果该实例还未运行，则上游任务运行完毕后，会自动运行。 · 如果上游任务都运行完毕，则该任务会直接被置为失败。需手动重跑后，方会正常运行。

实例状态说明

序号	状态类型	状态标识
1	运行成功状态	
2	未运行状态	
3	运行失败状态	
4	正在运行状态	
5	等待状态	
6	冻结状态	

4.3.3 补数据实例

补数据实例是对周期任务进行补数据时产生的实例，可以对补数据任务实例进行运维管理。例如查看运行状态，对任务实例进行终止、重跑和解冻等操作。

 说明:

补数据实例的限制与说明如下：

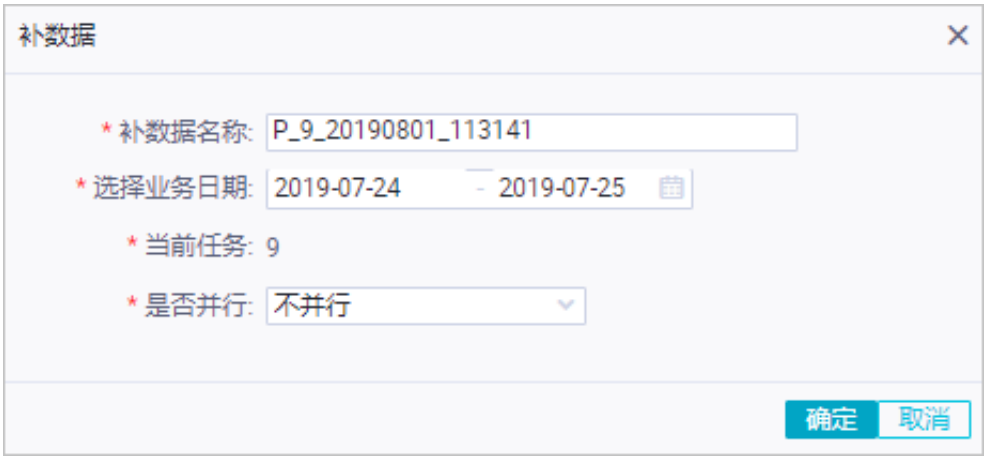
- 如果是补一个区间的数据任务，在第一天有一个任务实例失败了，则当天的补数据实例会被置为失败，第二天的任务实例也不会开始运行（只有当天的全部任务实例都成功，第二天的任务实例才会开始运行）。
- 自依赖的周期任务补数据，如果补数据第一个实例前一天的周期实例没有运行，则该补数据任务也无法触发运行。如果补数据的第一个实例前一天没有周期实例，则补数据直接触发运行。
- 目前仅有周期实例在任务失败时有报警。手动实例、补数据实例和测试实例任务失败均无报警。
- 如果当前任务的周期实例正在运行中，补数据和测试实例必须等周期实例完成才能开始运行。
- 如果周期实例和补数据实例同时都在运行，为了保证周期实例的正常运行，需要终止补数据实例的运行。

补数据

在周期任务列表中，单击相应周期实例后的补数据，选择当前节点、当前节点及下游节点或海量节点模式。

您可以选择对当前节点进行补数据或者为当前节点及下游节点进行补数据。

完成补数据节点的选择后，填写补数据对话框中的配置，单击确定。



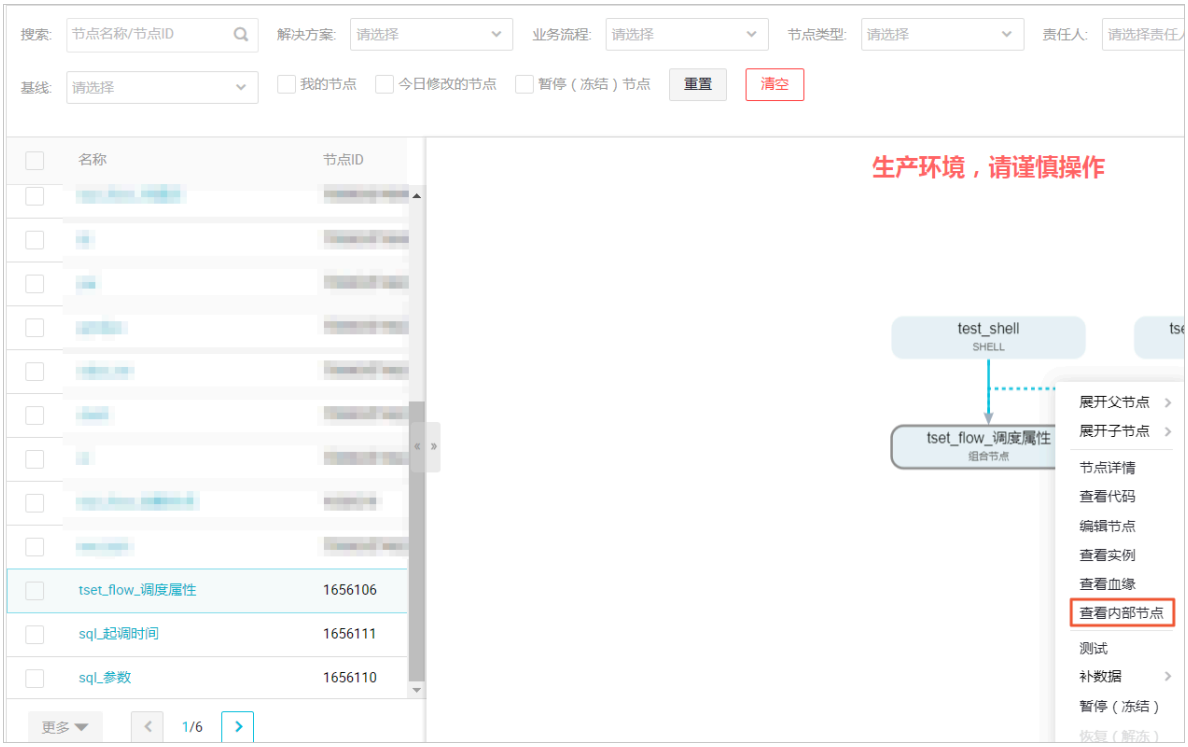
配置	说明
补数据名称	填写补数据任务的名称。
选择业务日期	选择补数据任务的业务日期。
当前任务	需要进行补数据的节点名称。

配置	说明
是否并行	您可以通过选择是否并行，控制同时生成多少个补数据实例来进行补数据。 · 选择不并行，只有一个补数据实例。 · 选择并行，您可以设置同时使用2组、3组、4组或5组等多个补数据实例进行补数据。  说明: · 不并行：一个补数据实例下的多个业务日期串行执行。 · 多个补数据实例下的多个业务日期： - 如果业务日期的跨度时间少于选择的并行组数，则并行执行。例如业务日期是1月11日~1月13日，并行组数选择的是4组，则只会生成3个补数据实例（每个补数据实例对应一个业务日期），三个实例同时并发执行。 - 如果业务日期的跨度大于选择的并行组数，则可能兼有串行和并行。例如业务日期是1月11日~1月13日，并行组数选择是2组，则会生成2个补数据实例（其中一个补数据实例会有两个业务日期，这两个业务日期对应的任务串行执行），两个补数据实例并行执行。

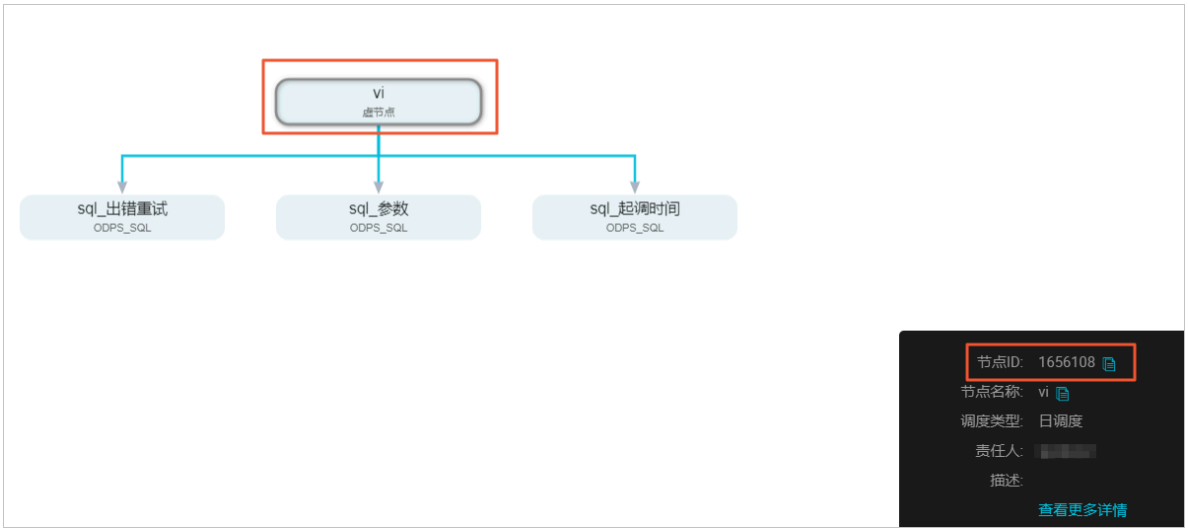
为组合节点中的特定节点补数据

您在DataWorks V1.0中使用的工作流，在您升级DataWorks V2.0后会在运维中心自动转换为组合节点。如果您需要为组合节点中的特定节点进行补数据，需遵照以下流程：

1. 在您的周期任务节点中找到组合节点对应的任务，在DAG图中右键选择查看内部节点。

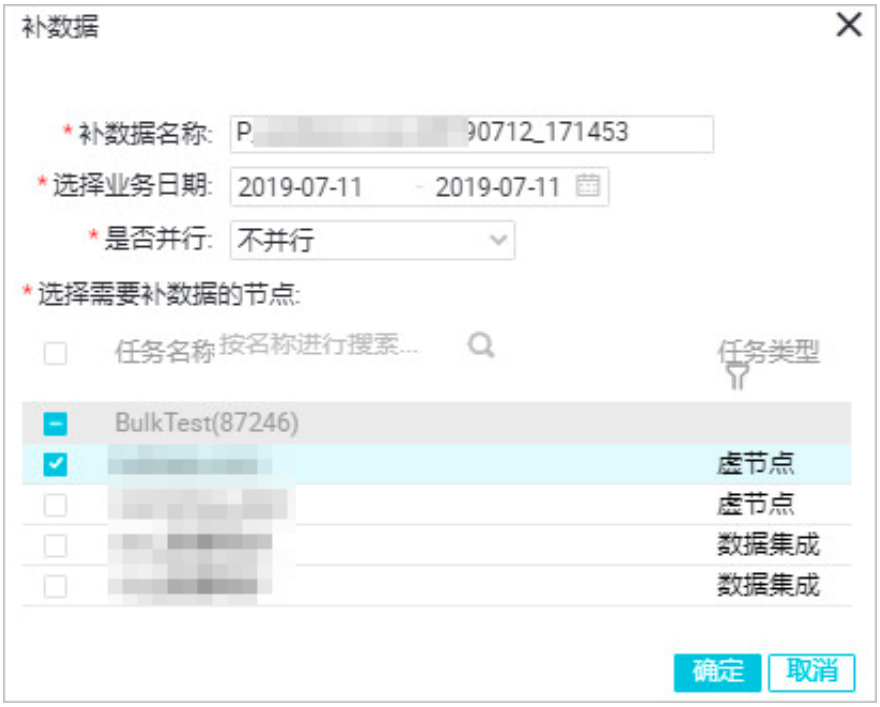


2. 在弹框中找到组合节点中您需要补数据的节点的上游节点，复制其节点ID。



3. 回到周期任务页面，搜索您刚刚获取的内部节点ID，右键点击搜索结果的DAG，选择补数据 > 当前节点及下游节点。

4. 选择您需要补数据的组合节点内的特定节点。



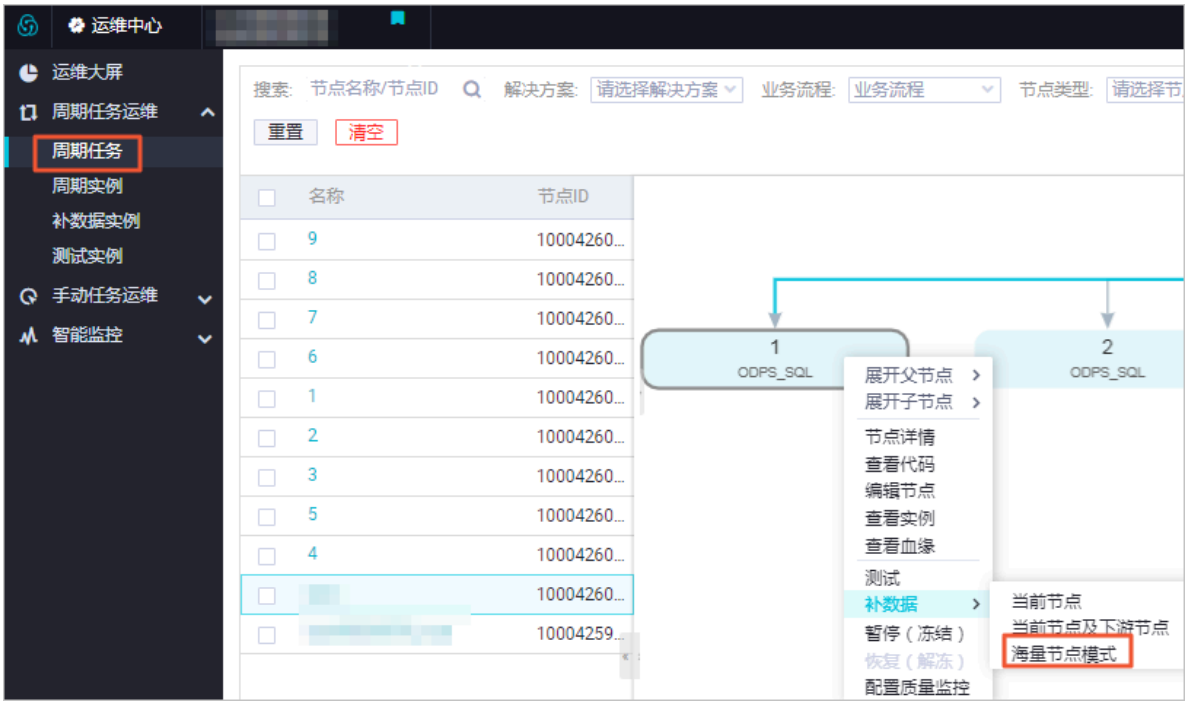
说明:

当前支持从组合节点查找到内部节点，但是不支持从内部节点查找到组合节点。

海量补数据

1. 在周期任务列表中，单击相应周期实例后的补数据，选择海量节点模式。

   您也可以打开DAG图，右键单击实例名称，选择补数据 > 海量节点模式。

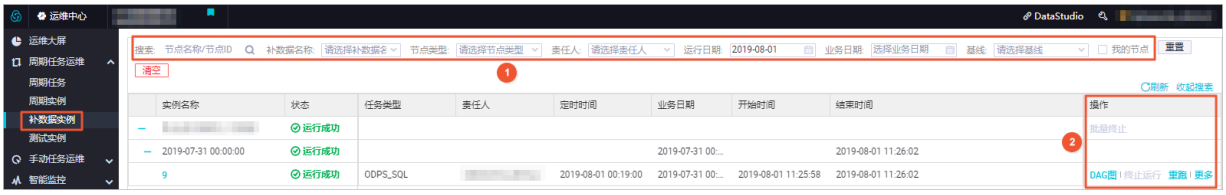


2. 填写补数据对话框中的配置。

配置	说明
补数据名称	通常会根据的您的节点自动生成一个补数据名称。
选择业务日期	选择您的业务时间，但不能选择当天的时间。 说明: 建议不要补太长时间的数据，以免出现任务需要等待资源的情况。
是否并行	您可以根据自身情况选择不并行、2组、3组、4组或5组。如果选择不并行，只能看见一个补数据实例。如果选择分组，则分几组便可看到几个补数据实例。

配置	说明
选择需要补数据的节点	· 如果勾选了包含当前节点，则补的是此节点及下游的数据。 · 如果没有勾选包含当前节点，则当前节点为空跑，补此节点下的下游数据。
选择工作空间	通常展示您的所有项目和补数据的项目，可以通过刷新选择相关的项目（此处支持模糊查询），然后将其加入补数据项目列表。
节点白名单	添加选中项目外仍需要进行补数据的节点。 说明: 目前仅支持搜索节点ID。
节点黑名单	添加选中项目中不需要进行补数据的节点。 说明: 目前仅支持搜索节点ID。

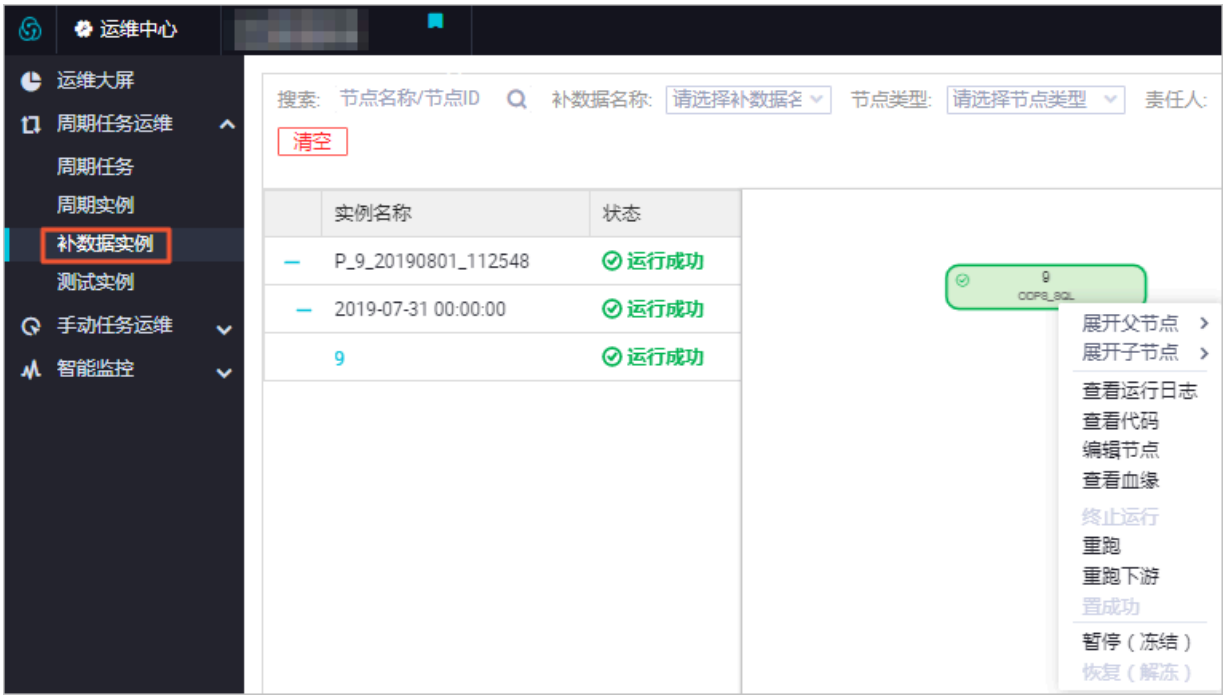
实例列表



操作	说明
筛选	如上图中的模块1，有丰富的筛选条件。您可以添加补数据名称、节点类型、责任人、运行日期等条件进行更精确的筛选。
DAG图	可以打开当前节点的DAG图，查看实例运行的结果。
终止运行	如果实例在运行中，可以单击终止运行，停止任务运行。
重跑	重新调度此实例。
重跑下游	重跑当前节点的下游任务。
暂停（冻结）	将当前节点置为暂停（冻结）状态，并停止调度。当节点状态为暂停时，在节点名称后会出现 图标。
恢复（解冻）	将暂停（冻结）的节点恢复调度。
查看血缘	查看节点的血缘关系图。

实例DAG图

单击实例名称或操作栏中的DAG图，即可打开该实例的DAG图。您可以在DAG图中，右键单击实例，进行相关操作。



说明:
右上角的刷新按钮只能刷新实例DAG状态，不能刷新实例的运行日志。

操作	说明
展开父节点/子节点	当一个工作流有3个节点及以上时，运维中心展示任务时会自动隐藏节点。您可以通过展开父子层级，来看到全部节点的内容。
查看运行日志	查看当前实例正在运行、成功、失败等状态的运行日志。
查看代码	查看当前实例的代码。
编辑节点	单击后，即可跳转至数据开发页面，对当前节点的内容进行修改。
查看血缘	查看当前实例的血缘关系。
终止运行	终止当前实例的运行。
重跑	失败的任务或状态异常的任务重跑实例。
重跑下游	当前节点的下游重跑实例，如果存在多个下游实例，会将这些实例全部重跑。
置成功	修改当前实例的状态为成功。 **说明:** 只有失败状态的任务能被置成功，工作流任务不能置成功。

操作	说明
暂停（冻结）	将当前实例置为暂停（冻结）状态，并停止调度。
恢复（解冻）	恢复暂停（冻结）的节点的调度。

实例状态说明

序号	状态类型	状态标识
1	运行成功状态	
2	未运行状态	
3	运行失败状态	
4	正在运行状态	
5	等待状态	
6	冻结状态	

4.3.4 测试实例

当周期任务达到启用调度所配置的周期性运行时间时，被自动调度的实例快照即为周期实例。

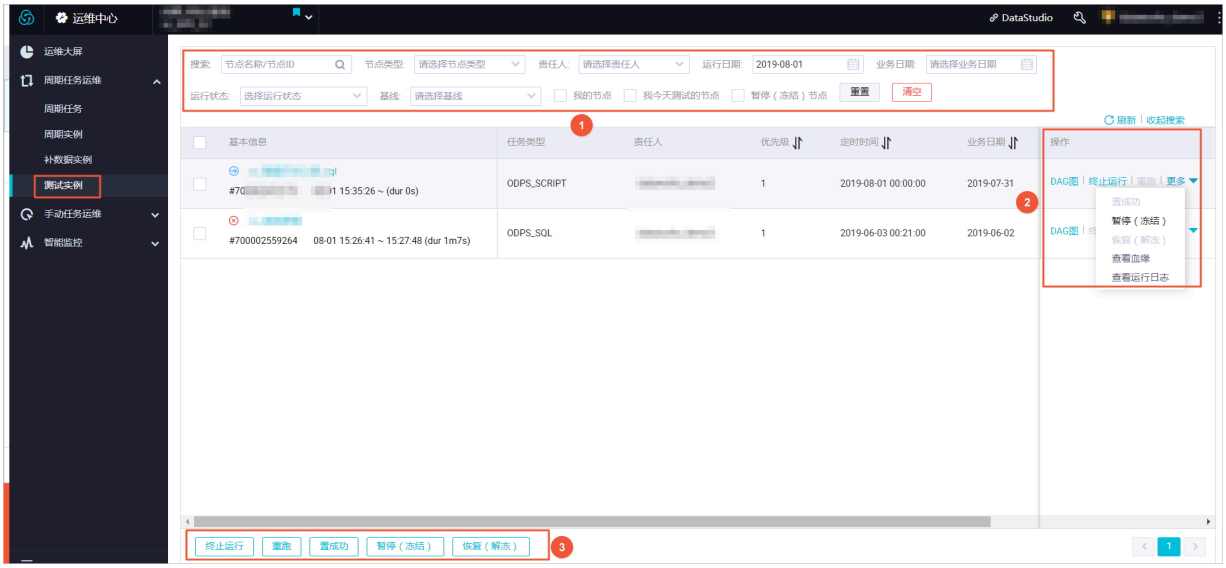
周期实例每调度一次，则生成一个实例工作流。您可以对已调度的实例任务进行日常的运维管理，例如查看运行状态，对任务进行终止、重跑和解冻等操作。

 说明:

目前仅有周期实例在任务失败时有报警。手动实例、补数据实例和测试实例任务失败均无报警。

测试实例列表

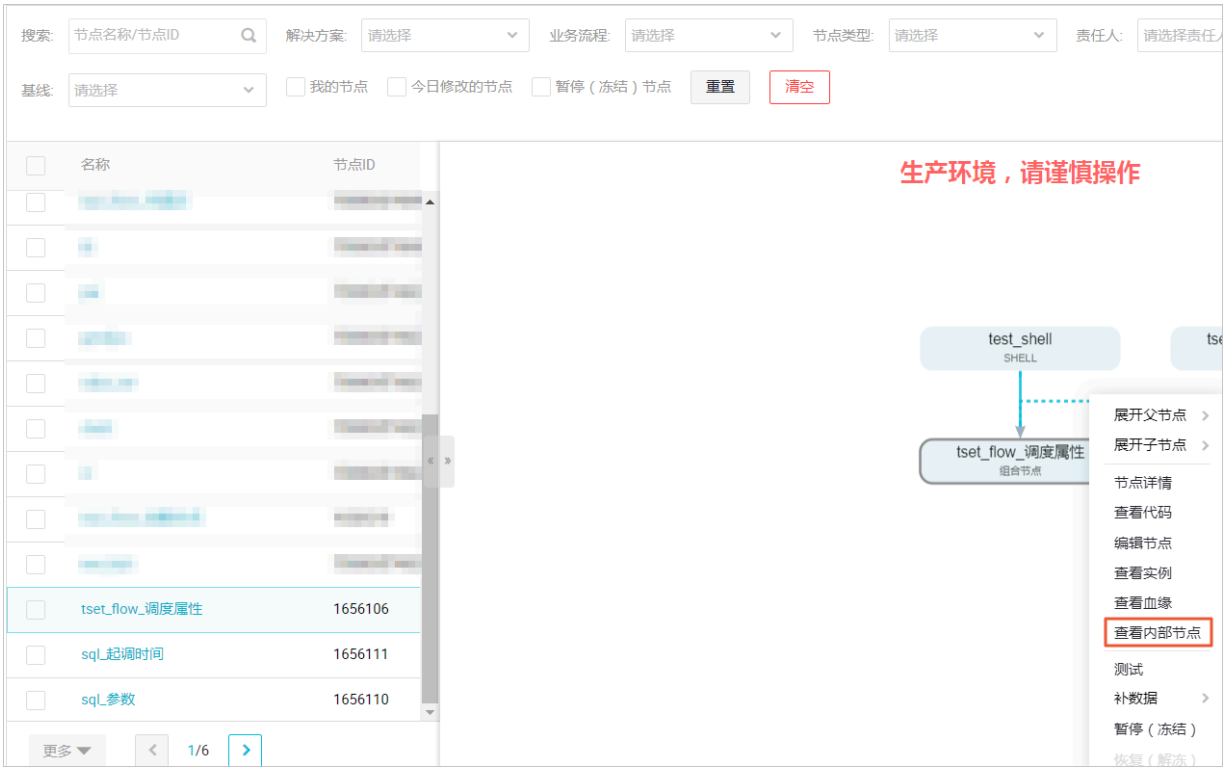
测试实例列表以列表形式对被调度的任务进行运维及管理，包括检查运行日志、重跑任务、终止正在运行的任务等。



操作	说明
筛选	如上图中的模块1，有丰富的筛选条件，默认筛选业务日期是当前时间前一天的工作流任务。您可以添加任务名称、运行时间、责任人等条件进行更精确的筛选。
终止运行	仅等待运行、运行中状态的实例可以进行终止运行的操作。进行此操作后，该实例将为失败状态。
重跑	可以重跑某任务，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理出错节点和漏跑节点。 说明: 只能重跑未运行、成功、失败状态的任务。
更多	单击操作栏中的更多，可以进行置成功、暂停（冻结）、恢复（解冻）、查看血缘和查看运行日志等更多操作。
批量操作	如上图中的模块3，批量操作包括终止运行、重跑、置成功、暂停（冻结）和恢复（解冻）。

实例DAG图

单击实例名或操作栏中的DAG图，即可打开该实例的DAG图。您可以在DAG图中，右键单击实例，进行相关操作。



操作	说明
查看运行日志	查看当前实例正在运行、成功、失败等状态的运行日志。
查看代码	查看当前实例的代码。
编辑节点	单击后，即可跳转至数据开发页面，对当前节点的内容进行修改。
查看血缘	查看当前实例的血缘关系。
终止运行	仅等待运行、运行中状态的实例可以进行终止运行的操作。进行此操作后，该实例将为失败状态。
重跑	可以重跑某任务，任务执行成功后可以触发下游未运行状态任务的调度。常用于处理出错节点和漏跑节点。 说明: 只能重跑未运行、成功、失败状态的任务。
置成功	将当前实例的状态改为成功，并运行下游未运行状态的任务。常用于处理出错节点。 说明: 只有失败状态的任务能被置成功，工作流任务不能置成功。
暂停（冻结）	周期实例中的冻结仅针对当前实例，且正在运行中的实例。

操作	说明
恢复（解冻）	可以将冻结状态的实例解冻。 · 如果该实例还未运行，则上游任务运行完毕后，会自动运行。 · 如果上游任务都运行完毕，则该任务会直接被置为失败。需手动重跑后，方会正常运行。

实例状态说明

序号	状态类型	状态标识
1	运行成功状态	
2	未运行状态	
3	运行失败状态	
4	正在运行状态	
5	等待状态	
6	冻结状态	

4.4 手动任务运维

4.4.1 手动任务

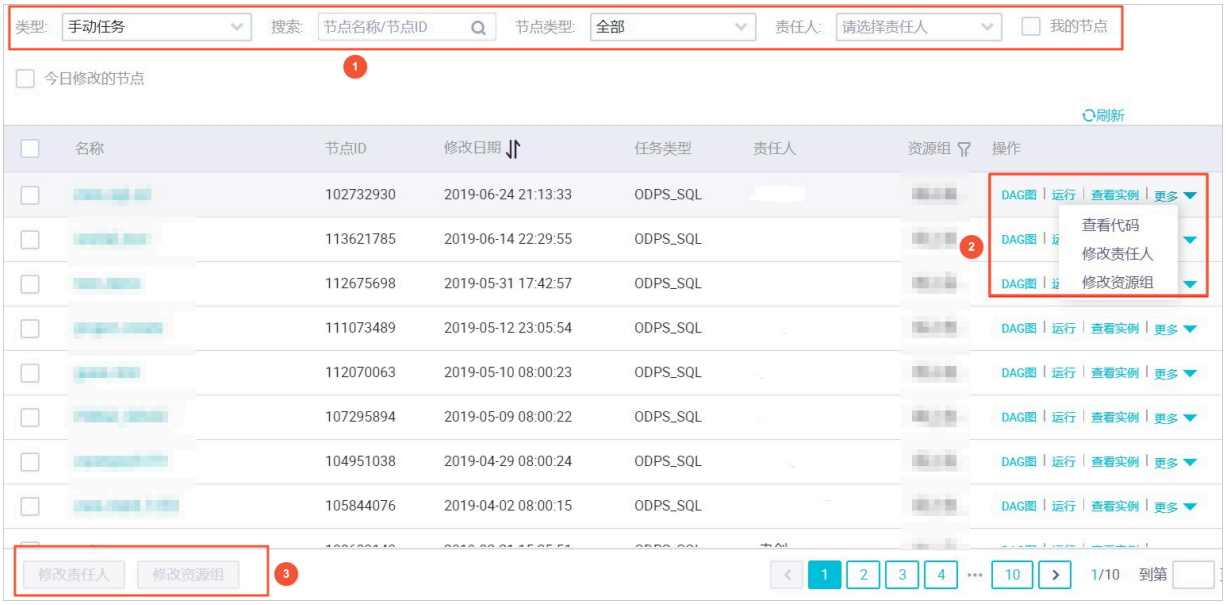
手动任务是指新建任务时，调度类型选择手动任务后，提交至调度系统的任务。

 说明:

· 手动任务提交至调度系统后，不会自动运行，只有手动触发才会运行。
· 目前DataWorks V1.0创建的手动任务显示在手动任务选项下，DataWorks V2.0创建的手动任务显示在手动业务流程选项下。

手动任务列表

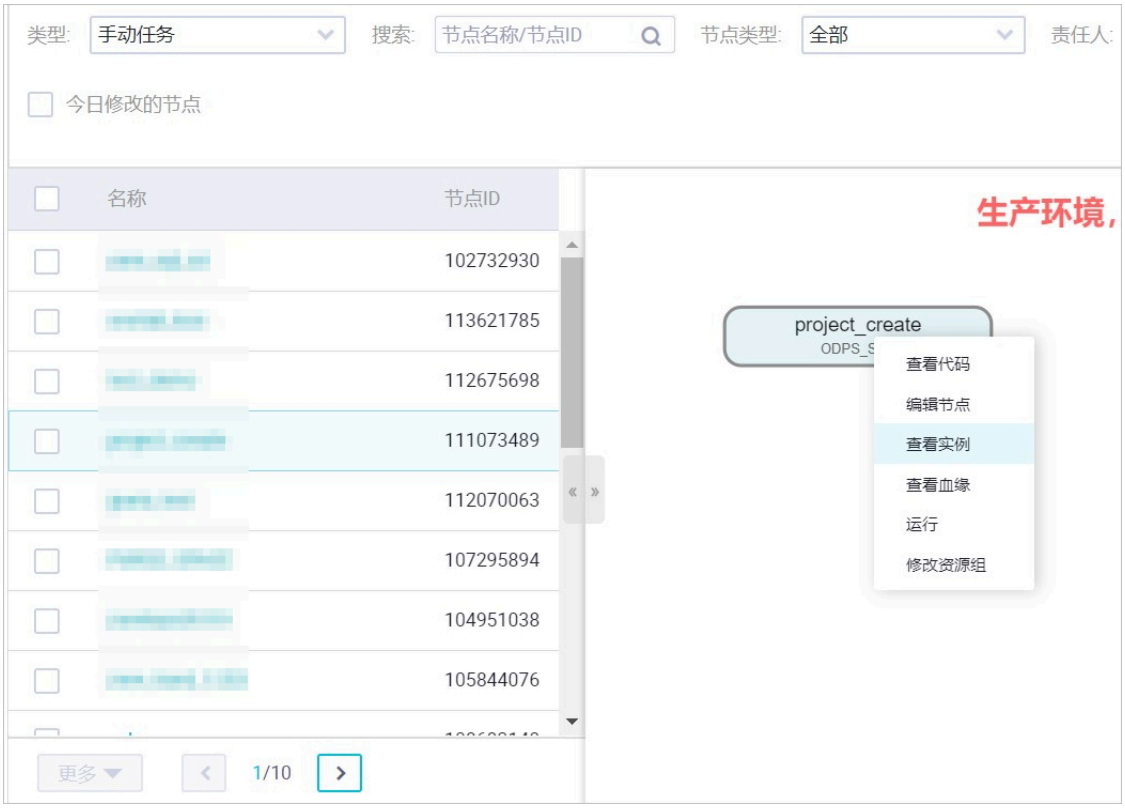
手动任务列表以列表的形式展示已提交的手动任务。



操作	说明
筛选	如上图中的模块1，通过筛选条件过滤出要查询的任务。您可以根据类型、节点类型、责任人和今日修改的节点等条件进行精确筛选。 说明: 任务名搜索的结果，会受到其他筛选条件的影响，只有同时满足所有筛选条件的结果才会展示出来。
DAG图	单击操作栏中的DAG图，即可打开此节点的DAG图。您可以在DAG图中查看节点的代码、血缘等信息。
运行	单击操作栏中的运行，即可运行此手动任务，产生手动实例。
查看实例	单击操作栏中的查看实例，即可跳转至手动实例页面，查看手动任务的运行结果。
更多	单击操作栏中的更多，可以修改责任人或资源组。 · 单击修改责任人，即可修改当前手动任务的节点责任人。 · 单击修改资源组，即可修改当前手动任务所在的资源组。
批量操作	如上图中的模块3，您可以批量选择任务，进行修改责任人和修改资源组的操作。

手动任务DAG图

单击任务名或操作栏中的DAG图，即可打开此节点的DAG图。您可以在DAG图中，右键单击节点，进行相关操作。



操作	说明
查看代码	查看当前节点的代码。
编辑节点	单击后可跳转至数据开发页面，对节点内容进行修改。
查看实例	查看当前节点的周期实例。
查看血缘	查看当前节点的血缘关系图。
运行	运行当前手动任务，产生手动实例。
修改资源组	修改当前节点所在的资源组。

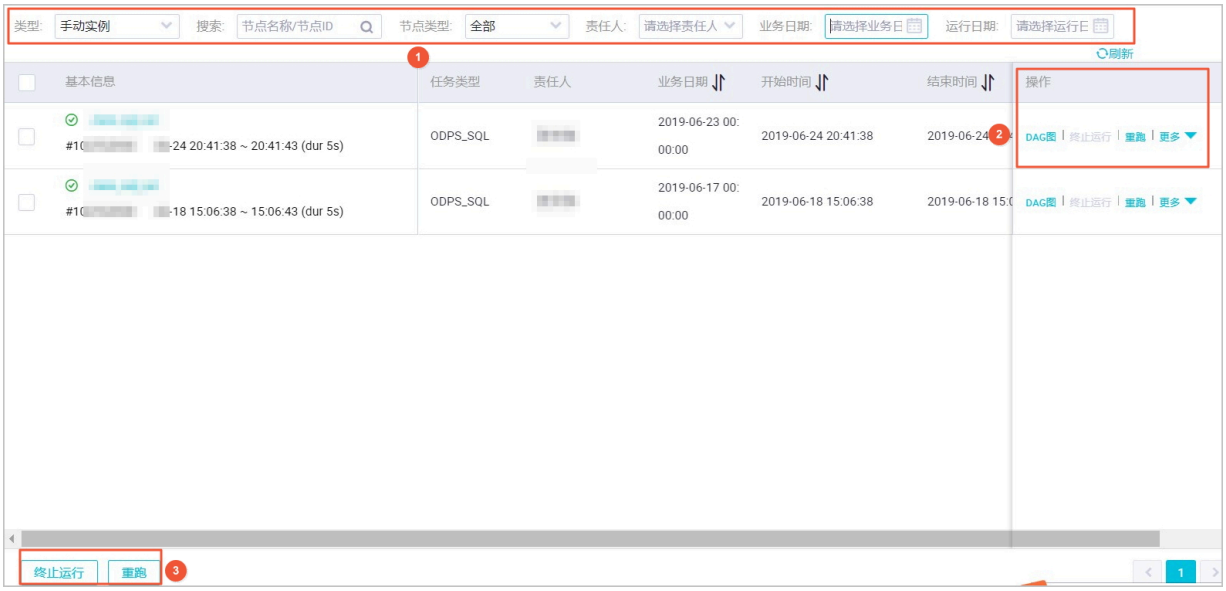
4.4.2 手动实例

手动实例是指手动任务产生的实例，手动任务的特点是没有调度依赖，只需要手动触发即可。

说明:

目前仅在周期实例任务失败时报警。手动实例、补数据实例和测试实例任务失败均无报警。

手动实例列表



操作	说明
筛选	如上图中的模块1，有丰富的筛选条件。您可以添加节点类型、责任人、业务日期等条件进行更精确的筛选。
DAG图	可以打开此节点的DAG图，查看实例运行的结果。
终止运行	如果实例在运行中，可以单击终止运行，停止任务运行。
重跑	重新调度当前实例。
批量操作	如上图中的模块3，批量操作包括终止运行和重跑。

手动实例DAG图

单击实例名称或操作栏中的DAG图，即可打开该实例的DAG图。您可以在DAG图中，右键单击实例，进行相关操作。

说明:

手动任务没有依赖关系，所以DAG图中只会显示当前实例。

操作	说明
查看运行日志	查看当前实例正在运行、成功、失败等状态的运行日志。
查看代码	查看当前节点的代码。
编辑节点	单击后，即可跳转至数据开发页面，对当前节点的内容进行修改。
查看血缘	查看当前实例的血缘关系。
终止运行	终止任务的运行，仅对当前实例有效。
重跑	失败的任务或状态异常的任务重跑实例。

4.5 智能监控

4.5.1 智能监控概述

智能监控是DataWorks任务运行的监控及分析系统，根据监控规则和任务运行情况，智能监控决策是否报警、何时报警、如何报警以及给谁报警。智能监控会自动选择最合理的报警时间，报警方式以及报警对象。

智能监控旨在：

- 降低您的配置成本。
- 杜绝无效报警。
- 自动覆盖所有重要任务。

通常情况下，监控系统只需您配置相应的监控规则即可，但这样无法满足DataWorks的需求，原因如下：

- DataWorks的任务极多，您无法准确地梳理出需要被监控的任务。部分DataWorks业务任务量较大，且任务之间的依赖较为复杂。即使您知道最重要的任务是什么，也很难查找任务的所有上游并全部进行监控。在这样的背景下，如果您直接监控所有任务，会触发较多无用的报警，从而导致有用报警被忽略，效果等同于没有监控。
- 针对每个任务监控的报警方式不同：部分监控需要任务运行超过1个小时报警，而有些监控需要任务运行超过2个小时报警。如果单独对每个任务设置监控极为繁琐，并且很难预估每个任务应该设置的报警阈值。
- 针对每个任务监控的报警时间不同：例如不重要的任务可以放到早上上班后再报警，而重要任务则需要夜间立刻报警，常用的监控系统无法区分每个任务的重要性。
- 报警如何关闭问题：如果报警一直进行提醒，需要在您响应时提供关闭报警的入口。

智能监控拥有一整套的监控报警逻辑，您只需要提供所关注业务的重要任务名称，智能监控即可监控整体任务的产出过程，并生成对应的标准统一的报警机制。智能监控同时也提供了轻量级的自助配置监控功能，使您可以根据自己的需求定义报警规则。

智能监控当前已承担了阿里巴巴集团所有重要业务的任务监控，智能监控的全路径监控功能保障了阿里巴巴集团所有重要业务的任务整体产出链路，智能监控的上下游路径分析功能可以及时发现风险并为业务部门提供运维信息。在智能监控的分析体系下，阿里巴巴集团业务保持了长期的高稳定性。

4.5.2 功能介绍

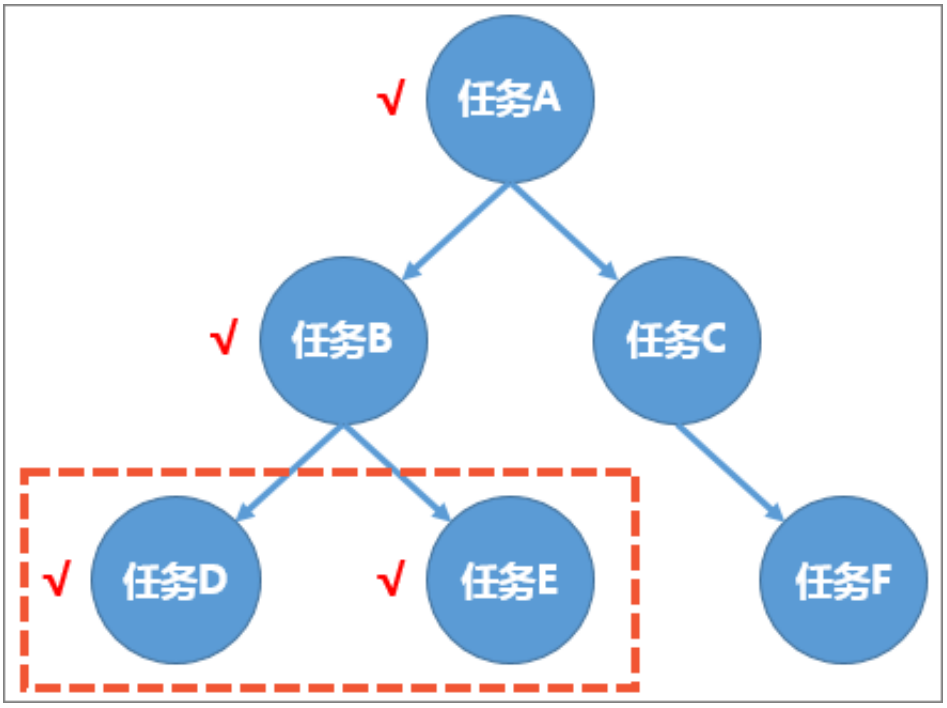
4.5.2.1 基线预警与事件告警

本文将从监控范围、捕获任务、判断报警时机、基线报警、报警方式和甘特图功能等方面，为您介绍基线预警与事件告警的功能逻辑。

监控范围

基线是一组节点的管理单位，即节点分组，您可以通过基线来设置任务纳入监控的方式。

一条基线纳入监控后，该基线及基线上游的所有任务都会被监控。此时智能监控不默认监控所有任务，被监控的任务下游必须有被纳入监控基线的任务。如果下游没有任务被纳入监控基线，即使该任务出错，智能监控也不会报警。



如上图所示，假设整个DataWorks只有6个任务节点，任务D和E属于纳入基线上的节点。任务D和E及它们所有的上游节点，都会被纳入监控范围。即上图中A、B、D、E任务出现异常（出错或变慢），也会被智能监控察觉，而任务C和任务F不受智能监控的监控。

捕获任务

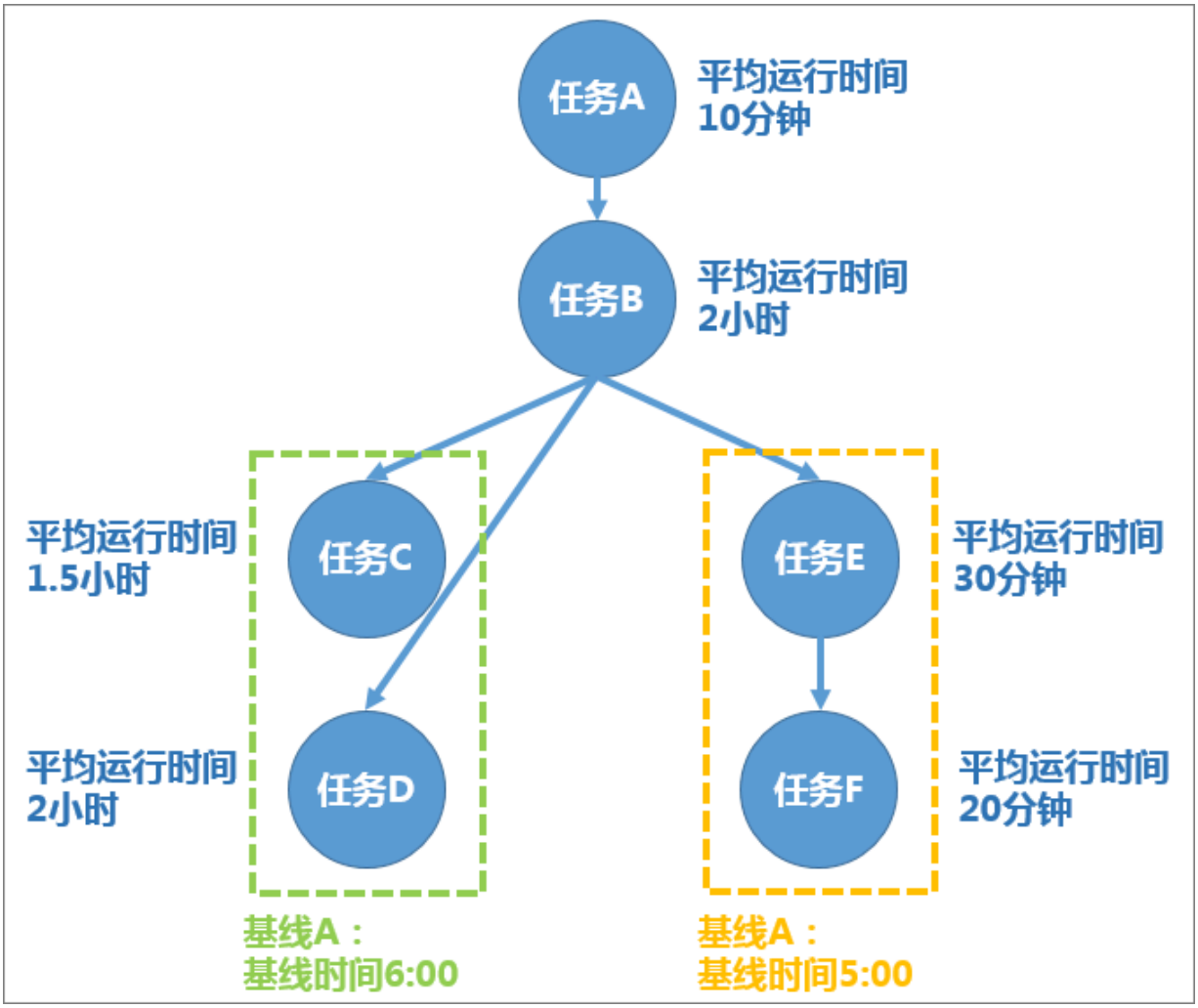
确定监控范围后，如果在监控范围内的任务出现异常，智能监控会生成一个事件，所有的报警决策都是基于对这个事件的分析。任务的异常包括两种类型，您可以导航至事件管理 > 事件类型页面进行查看。

- 出错：任务运行失败。
- 变慢：任务本次运行时间和过去一段时间内的平均运行时间相比，明显变长。

> 说明：
> 如果一个任务先超时，再出错，会生成两个事件。

判断报警时机

余量是智能监控中的一个重要概念，指任务可以被允许拖延开始的最长时间。任务的最晚开始时间=基线时间–平均运行时间。



如上图所示，要满足基线A的基线时间5:00，向上倒推，则要求任务E的最晚开始时间为5:00减任务F的运行时间20分钟，再减任务E的运行时间30分钟，即为4:10。该时间也是满足基线A的任务B的最晚完成时间。

要满足基线B的基线时间6:00，向上倒推，则要求任务B的最晚完成时间为6:00减任务D的运行时间2小时，即为4:00，早于4:10。即任务B的最晚完成时间为4:00，方可同时满足基线A和基线B。

任务A的最晚完成时间为4:00减任务B的运行时间2小时，即为2:00，最晚开始时间为2:00减任务A的运行时间10分钟，即为1:50。如果A无法在1:50分开始运行，则基线A容易破线。

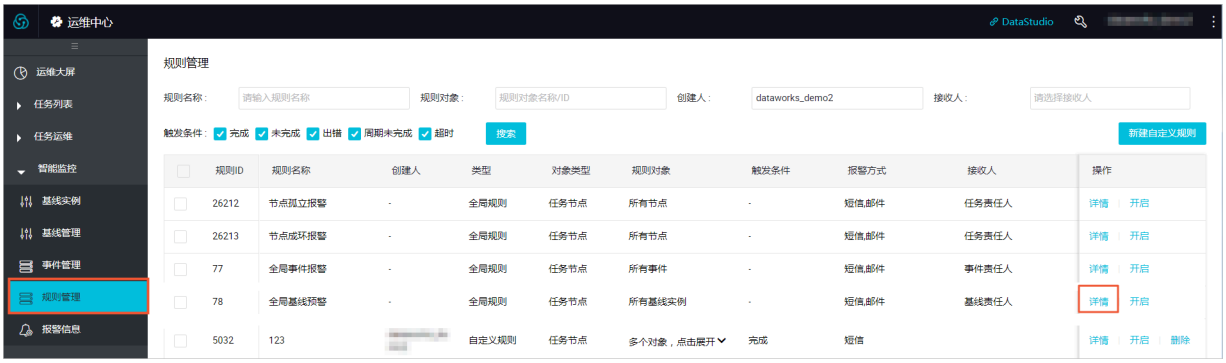
假设1点时任务A运行出错，则此时任务A的余量时间即为1:50和1:00之间的差值，即50分钟。由该示例可见，余量是一个任务异常的警戒程度的体现。

基线报警

基线报警是针对已经开启基线开关的基线的一个附加功能，每个基线都必须提供预警余量和承诺时间。基线报警指当某个时间，智能监控预测基线的完成时间超过预警余量，则会直接通知设置的报警对象，共3次，每次间隔30分钟。

报警方式

基线报警目前默认发送给基线责任人。您可以根据自身需求，在规则管理页面找到全局基线预警，选择详情，修改报警触发方式和报警行为。



甘特图功能

甘特图功能属于智能监控基线实例模块，反映的是一个任务的关键路径情况。

> 说明:
> 关键路径是指导致该任务在该时间点完成的上游最慢链路。

4.5.2.2 自定义提醒

自定义提醒是智能监控较为轻量级的监控功能，设计理念符合常规概念中的监控系统。

您可以自行设置所有的监控报警规则，配置内容如下所示：

- 规则对象：包括节点、基线和工作空间。
- 触发条件：包括完成、未完成、出错、周期未完成和超时。
- 报警方式：包括邮件和短信。
- 最大报警次数：报警的最大次数，超过设置的次数后，不再产生报警。
- 最小报警间隔：两次报警之间的时间间隔。
- 免打扰时间：在设置的时间段内不会发送报警。
- 接收人：可以设置为责任人或其他接收人。

监控规则包括完成、未完成、出错、周期未完成和超时5种触发条件。

- 完成

  可以设置针对任务、基线或工作空间的完成报警。一旦设置的对象上所有的任务均完成，则会发送报警。例如设置的是基线完成报警，则基线上所有的任务完成时，便会发送报警。

- 未完成

  可以设置针对任务、基线或工作空间在某个时间点没有完成的报警。例如设置某条基线在10:00完成，则10点只要基线上有一个任务没有完成，便会发送报警，并推送没有完成的任务列表。

- 出错

  可以设置针对任务、基线或工作空间的出错报警。任务一旦出错，则报警给设置的报警对象，并推送详细的任务出错信息。

- 周期未完成

  针对小时任务的监控规则，可以单独指定不同周期的未完成时间点。

- 超时

  可以设置针对任务、基线或工作空间的超时报警。一旦设置的对象中，发现有被监控的任务在指定时间内未完成，则会发送报警。

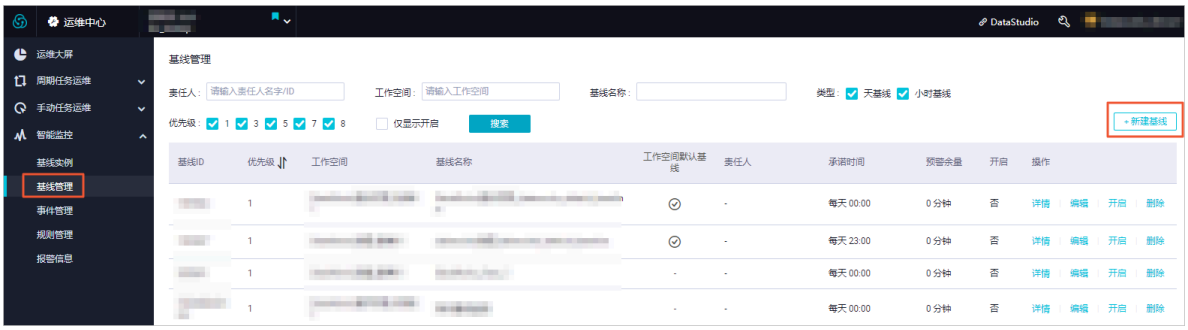
4.5.3 使用指导

4.5.3.1 基线管理

基线管理主要用于创建和定义基线。

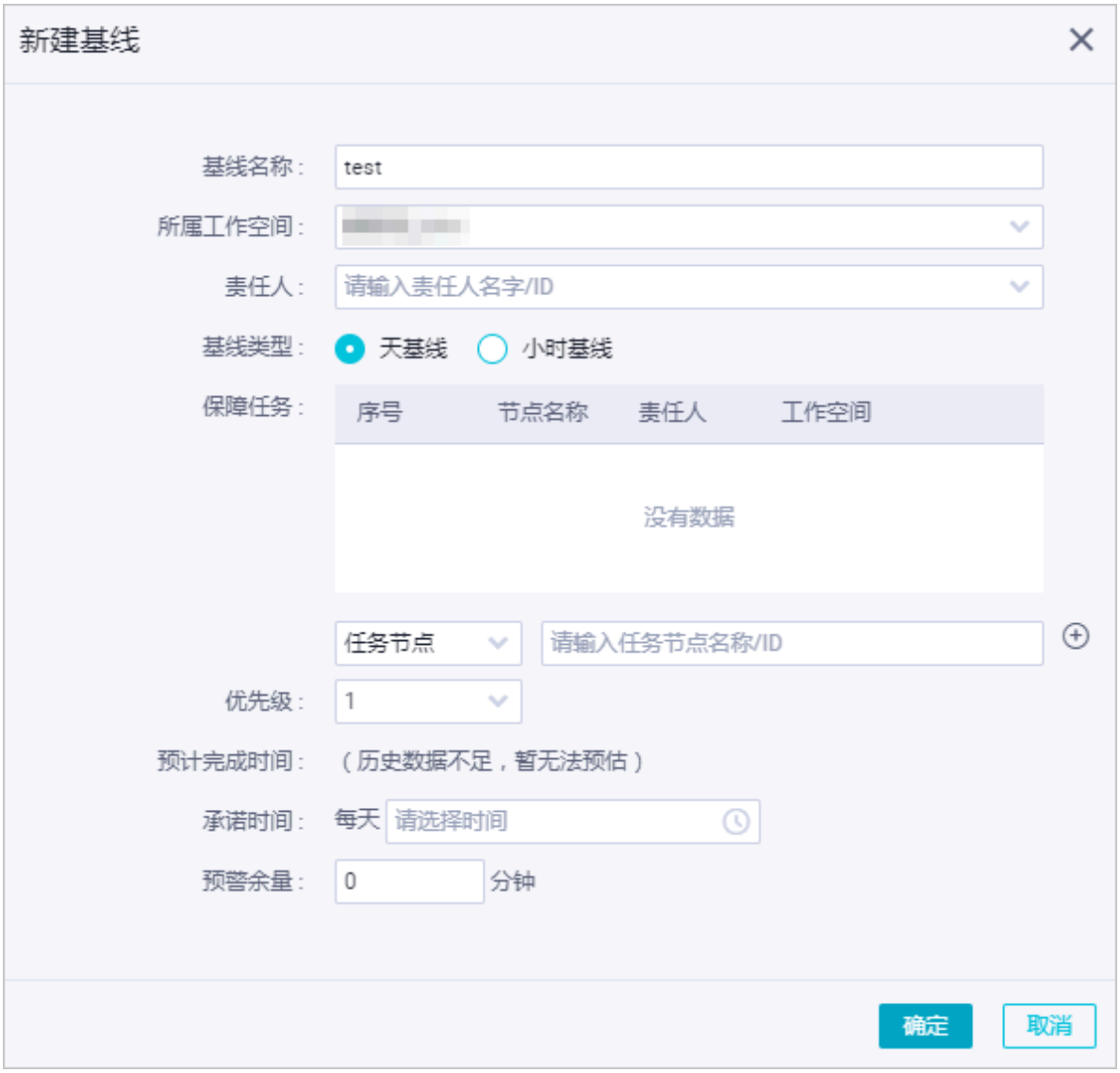
创建基线

1. 进入运维中心页面，选择智能监控 > 基线管理。
2. 单击基线管理页面右上角的新建基线，即可创建基线。



> **说明:**
> 目前仅项目管理员可以创建基线。

3. 填写新建基线对话框中的配置，单击确定。



配置	说明
基线名称	填写基线的名称。
所属工作空间	基线关联的任务所属的工作空间。
责任人	可以根据责任人名称和ID进行搜索。
基线类型	包括天基线和小时基线，决定基线是按天还是按小时进行检测。 · 天基线：对应调度配置中的天调度任务。 · 小时基线：对应调度配置中的小时调度任务。
保障任务	· 任务节点：基线具体关联的任务节点，输入任务节点名称或ID后，单击右侧图标进行添加，可以添加多个任务节点。 · 业务流程：输入业务流程名称或ID后，单击右侧图标进行添加。建议仅添加工作流最下游的节点任务，无需添加所有的任务。

配置	说明
优先级	优先调度数值较高的基线，目前仅一个优先级1。
预计完成时间	根据任务节点之前周期调度完成的平均时间进行预估。如果没有历史数据，会提示历史数据不足，暂无法预估。
承诺时间	如果实际完成时间晚于承诺时间-预警余量时间，则会触发报警。
预警余量	例如设置承诺完成时间为3:30，预警余量为10分钟，如果3:20任务没有完成，便会告警。假设此任务的平均运行时间是30分钟，如果2:50此节点仍未开始运行，便会告警。 说明: 根据最近15天的平均值，可以推算该任务的平均运行时间为30分钟。

4. 完成基线的创建后，单击操作栏中的开启，即可开启基线开关。

您可以单击相应基线后的详情、编辑、开启/关闭和删除进行相关操作。

- 详情：单击详情，即可查看基线任务的基本情况。
- 编辑：单击编辑，即可直接修改基线任务。
- 开启/关闭：控制基线任务的状态，开启才能生成周期实例。
- 删除：单击删除，即可直接删除基线任务。

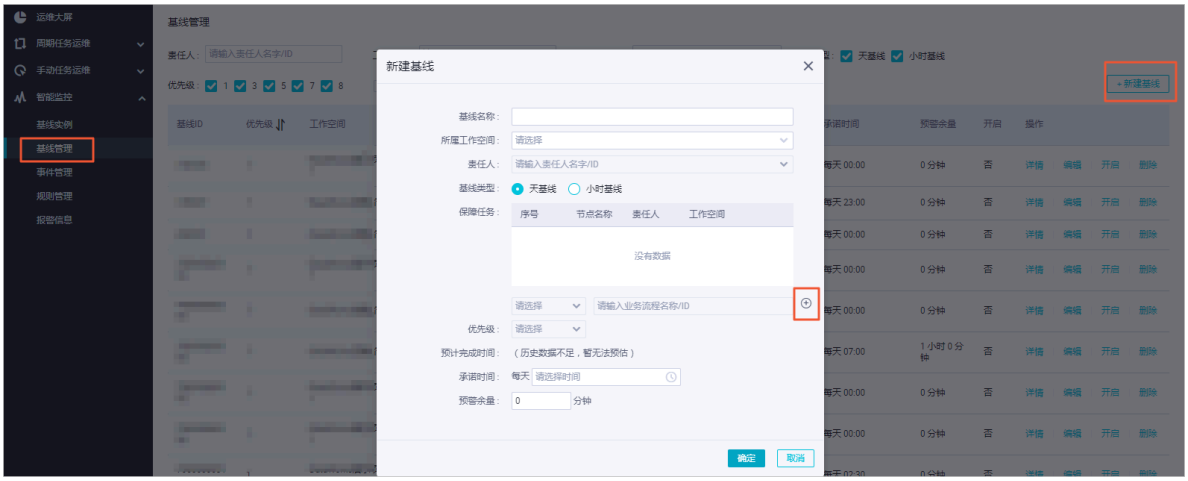
添加任务

生产环境的任务默认都在项目默认基线上，添加基线实际是将任务从默认基线转移至您新添加的基线上。

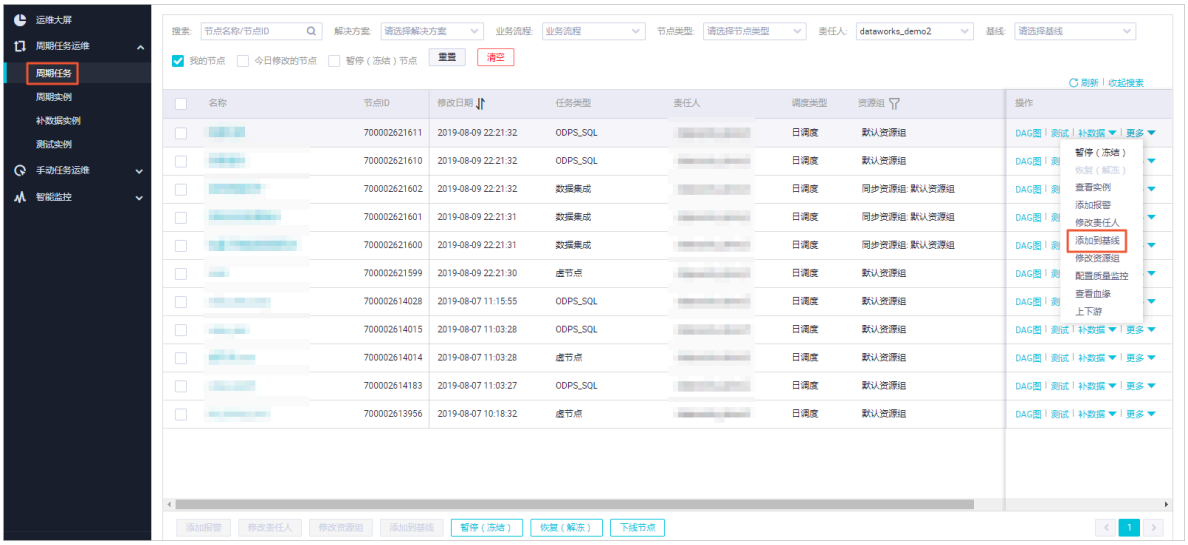
说明:
由于任务必须在基线上，所以不可以删除默认基线上的任务。删除自定义基线（您自行添加的基线）上的任务，实际是把您的任务从自定义基线转移至默认基线上。

您可以通过以下2种方式修改任务基线：

- 进入基线管理页面，单击右上角的新建基线进行添加。



- 进入周期任务页面，选择相应任务后的更多 > 添加到基线。



4.5.3.2 基线实例

基线实例主要用于查看基线的相关信息。

基线实例

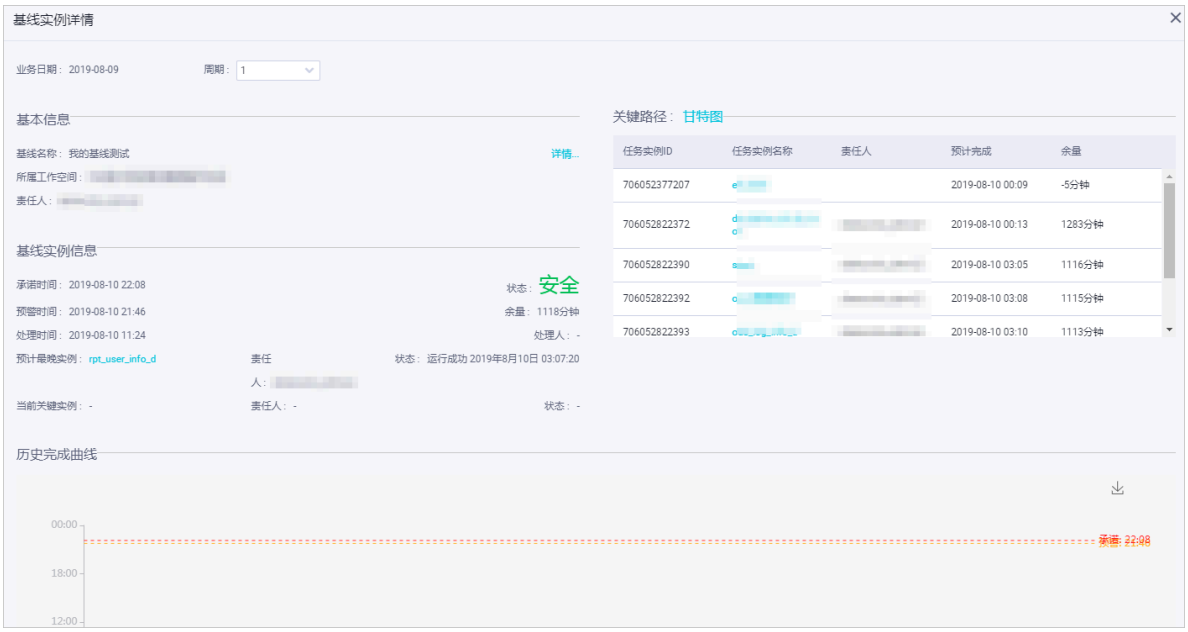
基线创建完成后，需要开启基线开关才会生成基线实例。在基线实例页面，您可以通过业务日期、责任人、相关事件ID、工作空间和基线名称等搜索对应实例，并进行查看详情、处理和查看甘特图等操作。

基线包括以下4种状态：

- 安全：任务在预警时间之前完成。
- 预警：任务在预警时间之后未完成，但还未到达承诺时间。
- 破线：任务在承诺时间之后仍未完成。
- 其他：基线所有任务处于暂停状态或基线没有关联任务。

基线对应的操作栏下包括详情、处理和甘特图。

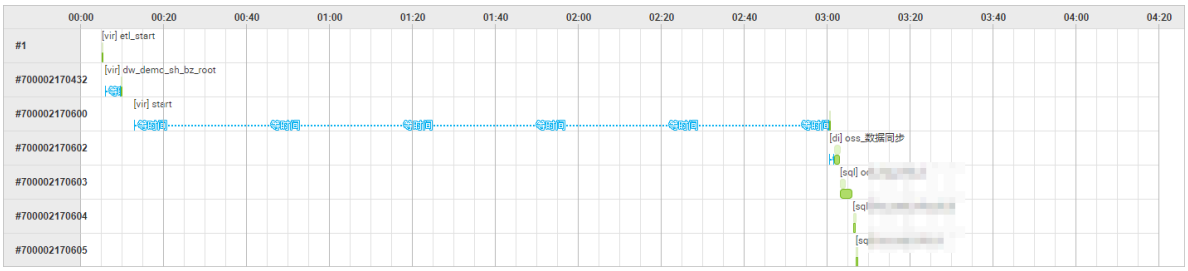
- 详情：单击详情，即可查看基线实例详情基线实例详情。



详情页面包括基本信息、关键路径、基线实例信息、历史完成曲线和相关事件。

> **说明：**
> 业务时间是系统时间-1天，仅小时基线有周期。

- 处理：报警的基线在处理时间内停止报警。
- 甘特图：单击甘特图，即可查看任务的关键路径情况。



4.5.3.3 事件管理

您可以在事件管理页面查看目前所有变慢和出错的事件。

进入事件管理页面，您可以通过责任人、发现时间、事件状态、事件类型、任务节点或任务实例的名称/ID等条件进行搜索。

对于搜索出的结果，每一行代表一个事件（即关联到一个异常的任务）。最差基线代表该事件所影响到的基线中余量时间最少的基线。

- **单击相应事件操作栏中的详情，即可查看事件发生时间、告警时间、恢复事件、任务的过往运行记录及详细的任务日志。**

  其中，实际报警接收人为指派给的人，单击报警信息即可跳转至事件对应的报警详情页面。基线影响会显示该事件对应的任务影响的所有下游基线，通过观察对应的下游基线和破线程度，结合任务日志，可以判断该事件发生的具体原因。
- **单击处理后，事件的处理操作记录会被记录，并且在操作期间暂停报警。**
- **单击忽略后，事件的忽略记录会被记录，并且永久停止报警。**

4.5.3.4 规则管理

本文将为您介绍如何在规则管理页面自定义报警规则。

1. **选择左侧菜单栏中的智能监控 > 规则管理，进入规则管理页面。**
2. **单击右上角的新建自定义规则。**

3. 填写新建自定义规则对话框中的配置。

配置	说明
规则名称	填写新建自定义规则的名称。
对象类型	控制监控的粒度，包括任务节点和业务流程。
规则对象	输入任务节点或业务流程的名称/ID后，单击右侧的图标即可添加对象。

配置	说明
触发条件	包括完成、未完成、出错、周期未完成和超时。
最大报警次数	报警的最大次数，超过设置的次数后，不再产生报警。
最小报警间隔	两次报警之间的时间间隔。
免打扰时间	在设置的时间段内不会发送报警。
报警方式	包括邮件和短信。
接收人	报警的对象，可以设置为任务责任人或其他接收人。
钉钉群机器人	可以添加钉钉群机器人接收报警。

4. 单击确定，即可生成规则。

   您可以在规则管理页面，单击相应规则后的详情，查看规则的具体内容。

4.5.3.5 报警信息

您可以在智能监控模块查看所有的报警信息。

进入智能监控 > 报警信息页面，您可以通过规则ID/名称、接收人、报警时间、报警方式和规则类型等信息进行搜索。

您可以查看相应报警的报警方式、发送状态等信息。单击操作栏中的详情，即可查看报警的详细内容。

4.5.4 智能监控常见问题

4.5.4.1 我的报警为什么报给了别人？

报警发送的邮箱手机，您可以进入数加控制台>个人信息页面，查看报警发送的邮箱地址和手机。

报警的发送逻辑为：报警发送给指定的人，接收人为子账号。如果子账号没有配置手机信息，会根据基本接收管理中产品的欠费、停服、即将释放等相关信息，通知子类中配置的接收人去发送。

4.5.4.2 不想接受不重要的任务的报警，该怎么办?

单击事件管理页面中的详情，即可查看任务影响的下游基线。这些基线的范围内如果出现问题，可能导致任务报警，请联系相应的基线负责人。

4.5.4.3 为什么开启的基线破线未报警?

基线开关开启的基线监控是针对任务的。如果所有的任务都正常，即使破线也不会报警，因为所有的任务都运行正常，无法判断出哪个任务出错。

任务都正常但基线仍破线的原因，通常有以下2个原因：

- 设置的基线时间不合理。
- 任务的依赖有问题，即使基线破线也不报警。

4.5.4.4 变慢的任务是否可以不报警?

任务变慢报警一定要满足以下2个条件:

- 任务处于重要的基线上游。
- 任务和往常比较，确实存在变慢的情况。

如果任务变慢的影响不大，可以选择忽略，请和下游监控了任务的基线方确认（您可以在事件管理页面查看下游基线信息）。如果确认要为下游方负责，请维护好任务。

4.5.4.5 为什么未收到出错任务的报警?

并不是所有任务出错后都会报警，任务需要满足下述条件之一，才会在出错后进行报警:

- 处于某条基线开关开启的基线的上游。
- 设置了相关的自定义提醒规则。

4.5.4.6 夜间收到了报警怎么办?

夜间收到报警，可以登录事件页面关闭事件报警一段时间。

但只能关闭报警一段时间，收到报警后还应及时处理问题。

5 工作空间管理

5.1 工作空间配置

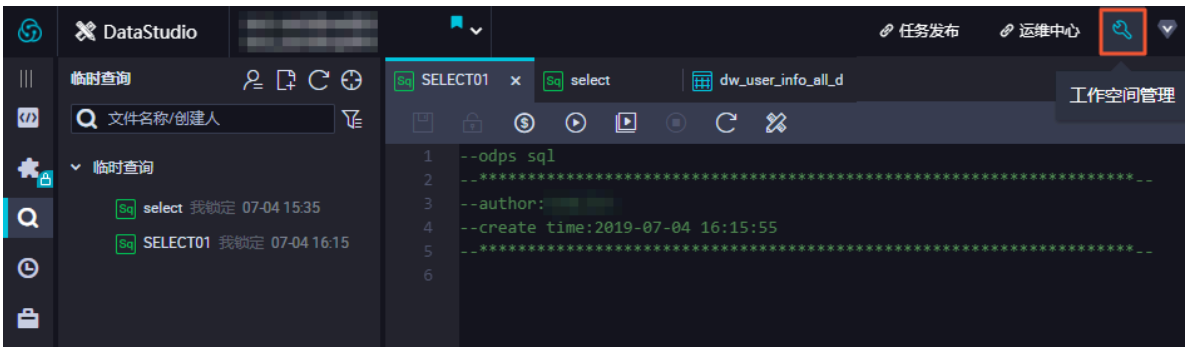
您可以在工作空间配置页面，对当前工作空间的属性进行管理和配置。

操作步骤

1. **登录DataWorks控制台，进入工作空间列表页面 。**
2. **单击对应工作空间后的工作空间配置。**

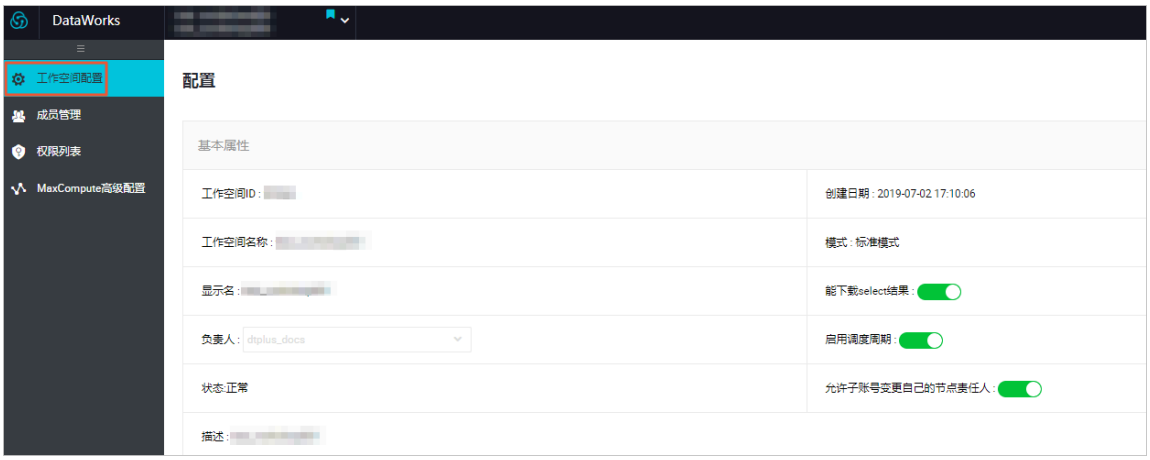
3. 单击工作空间配置对话框中的更多配置，即可进入工作空间配置页面。

您也可以进入数据开发页面，单击右上角的工作空间管理，进入工作空间配置页面。



4. 根据自身需求，在该页面进行基本属性、沙箱白名单和计算引擎信息等配置。

   · 基本属性



配置	说明
工作空间ID	当前工作空间的ID。
工作空间名称	当前工作空间的名称，仅支持字母或者数字（必须字母开头），不区分大小写。它是该工作空间的唯一标识，创建后无法修改。
显示名	当前工作空间的显示名称，用于标识工作空间，支持字母、数字或中文，可以修改。
负责人	当前工作空间的所有者，拥有删除、禁用工作空间的权限，并且该身份无法变更。
状态	工作空间分为初始化中、初始化失败、正常和禁用四种状态。 - 工作空间新建时状态为初始化中。 - 新建失败时状态为初始化失败，可以重试。 - 正常的工作空间可以被管理员禁用，禁用后该工作空间所有功能无法使用，数据保留，已经提交的任务正常执行。 - 被禁用的工作空间可以通过恢复的功能，将工作空间重新置于正常状态。
创建日期	当前工作空间的创建日期，中国站以东八区为准，无法变更。
模式	分为简单模式和标准模式。
能下载select结果	可以设置是否能下载select结果。
启用调度周期	控制当前工作空间是否启用调度系统，如果关闭则无法周期性调动任务。

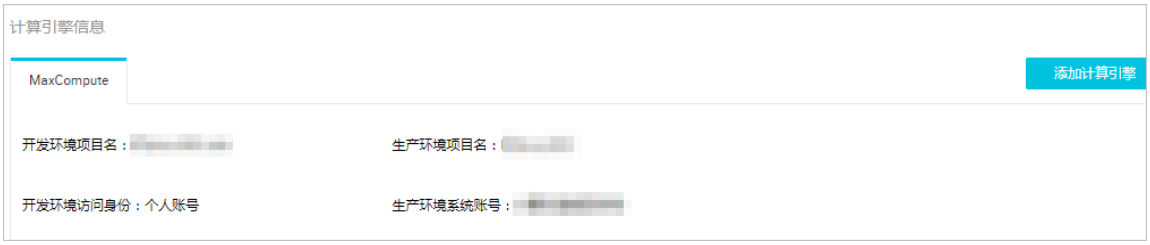
配置	说明
允许子账号变更自己的节点责任人	可以设置是否允许子账号变更自己的节点责任人。
描述	当前工作空间的描述信息，用于备注工作空间的相关信息，可以编辑。44支持128位中文、字母、符号或数字。

- 沙箱白名单（配置Shell任务可以访问的IP地址或域名）

  在此处配置即使Shell任务运行在默认资源组上，也可以直接访问的IP（此处白名单可以配置IP和域名）。

> 说明:
> 必须填写能被访问到的公网地址或域名。如果是内部服务， 建议使用独享资源保证网络可达，详情请参见独享资源模式。

- 计算引擎信息



配置	说明
开发环境项目名	当前DataWorks工作空间底层使用的MaxCompute开发环境的项目名称。 说明: 该MaxCompute项目是计算和存储资源。
开发环境访问身份	默认为个人账号，不可修改。
生产环境项目名	当前DataWorks工作空间底层使用的MaxCompute生产环境的项目名称。

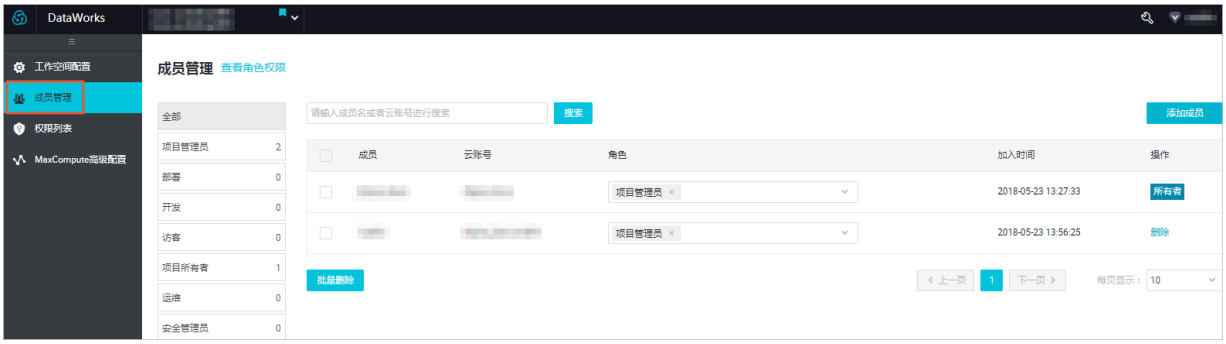
配置	说明
生产环境系统账号	默认选择系统账号。 项目负责人账号执行SQL使用的是主账号的AccessKey，个人账号执行SQL使用的是子账号的AccessKey。 系统账号可以操作该账号下所有工作空间的表，个人账号只能操作有权限的表。 说明: 当生产环境系统账号使用的是个人账号时，在生产环境运行的任务可能因为权限不足而大批量出错，请谨慎操作。

5.2 成员管理

您可以在成员管理页面，对当前工作空间的成员进行管理和配置。

页面说明

进入工作空间配置页面后，单击左侧导航栏中的成员管理，即可进入成员管理页面。



列表项	说明
成员	登录者云账号的显示名。
云账号	登录者的云账号。
成员角色	成员作为当前DataWorks工作空间成员，在工作空间中拥有的角色（项目管理员、部署、开发、访客、运维和安全管理员）。不同成员角色的具体权限，请单击权限列表进行查看。
加入时间	显示该成员加入当前工作空间的时间。
操作	当前用户可以对成员进行的操作。单击操作栏下的删除，即可将该成员移出当前工作空间（仅项目管理员角色拥有该权限）。

单击右上角的添加成员，系统可以为您全量同步主账号下全量子用户账号，并提供搜索筛选的功能。

您可以选中一个或多个搜索结果并批量设置角色，确定后成员添加至工作空间中。被添加的成员可以进入当前空间进行操作，详情请参见#unique_469。

说明:

如果在添加成员列表中没有找到要添加的成员账号，可以单击刷新，将子账号同步至DataWorks。

刷新成功后，选中子账号的勾选框，将子账号转移至右侧已添加的账号栏下，选中底部需要授予的角色，单击确认即可完成添加操作。

查看权限

您可以在MaxCompute_SQL任务中，执行如下语句，查询自己的权限信息。

```
show grants --查看当前用户自己的访问权限
show grants for <username> --查看指定用户的访问权限，仅由项目管理员才有执行权限。
```

更多查看权限的命令请参见#unique_470。

5.3 权限列表

DataWorks提供项目所有者（不可授权）、项目管理员、开发、运维、部署、访客和安全管理员7种角色，本文将为您介绍具体角色的权限说明。

数据管理

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
自己创建的表删除	√	√	√	无	无	无	无
自己创建的表类目设置	√	√	√	无	无	无	无
自己收藏的表查看	√	√	√	无	无	无	无
新建表	√	√	√	无	无	无	无
自己创建的表取消隐藏	√	√	√	无	无	无	无
自己创建的表结构变更	√	√	√	无	无	无	无
自己创建的表查看	√	√	√	无	无	无	无
自己申请的权限内容查看	√	√	√	无	无	无	无
自己创建的表隐藏	√	√	√	无	无	无	无
自己创建的表生命周期设置	√	√	√	无	无	无	无
非自己创建的表数据权限申请	√	√	√	无	无	无	无
更新表	√	√	√	√	√	无	无
删除表	√	√	无	无	无	无	无

发布管理

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
创建发布包	√	√	√	√	无	无	无
查看发布包列表	√	√	√	√	√	√	无
删除发布包	√	√	√	√	无	无	无
执行发布	√	√	无	√	√	无	无
查看发布包内容	√	√	√	√	√	√	无

按钮控制

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
按钮 -- 停止	√	√	√	无	无	无	无
按钮 -- 格式化	√	√	√	无	无	无	无
按钮 -- 编辑	√	√	√	无	无	无	无
按钮 -- 运行	√	√	√	无	无	无	无
按钮 -- 放大	√	√	√	无	无	无	无
按钮 -- 保存	√	√	√	无	无	无	无
按钮 -- 展开/收起	√	√	√	无	无	无	无
按钮 -- 删除	√	√	√	无	无	无	无

代码开发

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
保存提交代码	√	√	√	无	无	无	无
查看代码内容	√	√	√	√	√	√	无
创建代码	√	√	√	无	无	无	无
删除代码	√	√	√	无	无	无	无
查看代码列表	√	√	√	√	√	√	无
运行代码	√	√	√	无	无	无	无
修改代码	√	√	√	无	无	无	无
文件下载	√	√	√	无	无	无	无
文件上传	√	√	√	无	无	无	无

函数开发

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
查看函数详情	√	√	√	√	√	√	无
创建函数	√	√	√	无	无	无	无
查询函数	√	√	√	√	√	√	无
删除函数	√	√	√	无	无	无	无

节点类型控制

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
节点 -- PAI	√	√	√	无	无	无	无
节点 -- MR	√	√	√	无	无	无	无
节点 -- CDP	√	√	√	无	无	无	无
节点 -- SQL	√	√	√	无	无	无	无
节点 -- XLIB	√	√	√	√	√	√	无
节点 -- Shell	√	√	√	无	无	无	无
节点 -- 虚拟节点	√	√	√	√	√	√	无
节点 -- script_seahawks	√	√	√	无	无	无	无
节点 -- dtboost_analytic	√	√	√	无	无	无	无
节点 -- dtboost_recommand	√	√	√	无	无	无	无
节点 -- pyodps	√	√	√	无	无	无	无

资源管理

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
查看资源列表	√	√	√	√	√	√	无
删除资源	√	√	√	无	无	无	无
创建资源	√	√	√	无	无	无	无
上传jar文件	√	√	√	无	无	无	无
上传taxt文件	√	√	√	无	无	无	无
上传archive文件	√	√	√	无	无	无	无

工作流开发

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
运行/停止工作流	√	√	√	无	无	无	无
保存工作流	√	√	√	无	无	无	无

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
查看工作流内容	√	√	√	√	√	√	无
提交节点代码	√	√	√	无	无	无	无
修改工作流	√	√	√	无	无	无	无
查看工作流列表	√	√	√	√	√	√	无
修改owner属性	√	√	无	无	无	无	无
打开节点代码	√	√	√	无	无	无	无
删除工作流	√	√	√	无	无	无	无
创建工作流	√	√	√	无	无	无	无
创建文件夹	√	√	√	无	无	无	无
删除文件夹	√	√	√	无	无	无	无
修改文件夹	√	√	√	无	无	无	无

数据集成

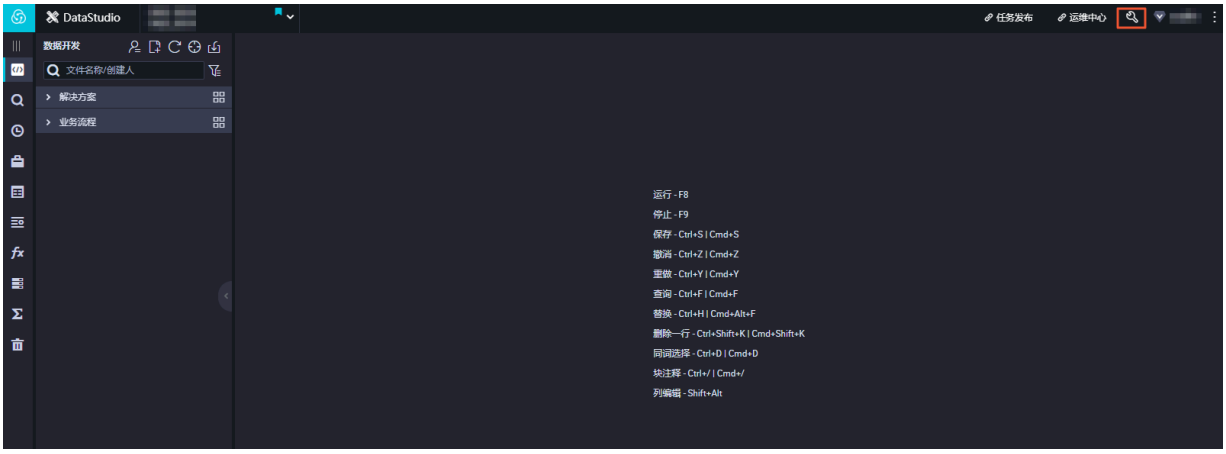
权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
数据集成 - 节点编辑	√	√	√	无	无	无	无
数据集成 - 节点查看	√	√	√	无	无	无	无
数据集成 - 节点删除	√	√	√	无	无	无	无
项目资源消耗监控菜单	√	√	无	无	无	无	无
项目同步资源管理菜单	√	√	√	√	√	无	无
项目同步资源组列表	√	√	√	√	√	√	无
项目同步资源组创建	√	√	√	√	√	无	无
项目同步资源组管理机器列表	√	√	√	√	√	无	无
项目同步资源组添加机器	√	√	√	√	√	无	无
项目同步资源组删除机器	√	√	√	√	√	无	无
项目同步资源组修改机器	√	√	√	√	√	无	无
项目同步资源组获取资源组ak	√	√	√	√	√	无	无
项目同步资源组删除	√	√	√	√	√	无	无

权限点	项目所有者	项目管理员	开发	运维	部署	访客	安全管理员
项目资源消耗监控	√	√	无	无	无	无	无
运维中心任务修改资源组	√	√	√	√	√	无	无
同步任务列表菜单	√	√	√	√	√	无	无
任务转脚本	√	√	√	√	√	无	无
获取项目成员列表	√	√	√	√	√	无	无
新建代码接口	√	√	√	√	√	无	无
保存/更新代码接口	√	√	√	√	√	无	无
根据fileId获取代码接口	√	√	√	√	√	√	无
获取数据集成节点列表	√	√	√	√	√	无	无
搜表接口	√	√	√	√	√	无	无
搜字段接口	√	√	√	√	√	无	无
查询数据源列表接口	√	√	√	√	√	√	无
新建数据源接口	√	√	无	无	无	无	无
查询数据源详情接口	√	√	√	√	√	无	无
更新数据源接口	√	√	无	无	无	无	无
删除数据源接口	√	√	无	无	无	无	无
测试连通性	√	√	√	√	√	无	无
数据预览	√	√	√	√	√	无	无
检查是否开通OTS Stream	√	√	√	√	√	无	无
开通OTS	√	√	√	√	√	无	无
查询ODPS建表语句	√	√	√	√	√	无	无
新建ODPS表	√	√	√	√	√	无	无
查询ODPS建表状态	√	√	√	√	√	无	无
迁移数据库表	√	√	无	无	无	无	无

5.4 MaxCompute高级配置

您可以通过工作空间管理中的MaxCompute高级配置操作，对当前工作空间的MaxCompute属性进行管理和配置。

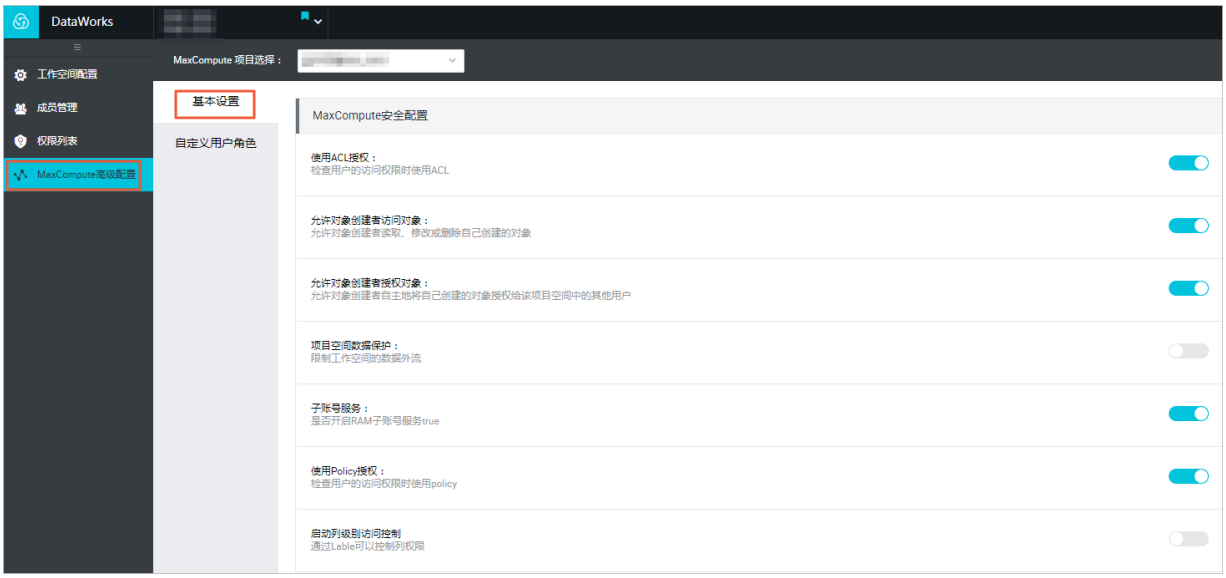
单击数据开发页面右上角的工作空间管理，即可进入工作空间配置页面。



单击左侧导航栏中的MaxCompute高级配置，即可进入MaxCompute高级配置页面。该页面包括基本设置和自定义用户角色两个模块。

基本设置

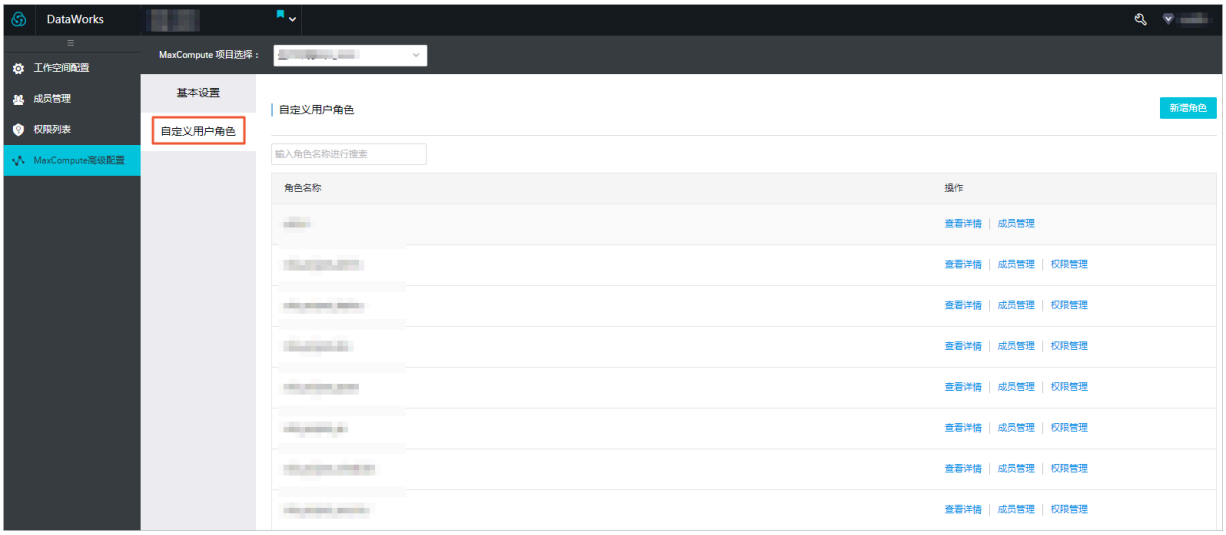
您可以在基本设置模块，进行MaxCompute安全配置。



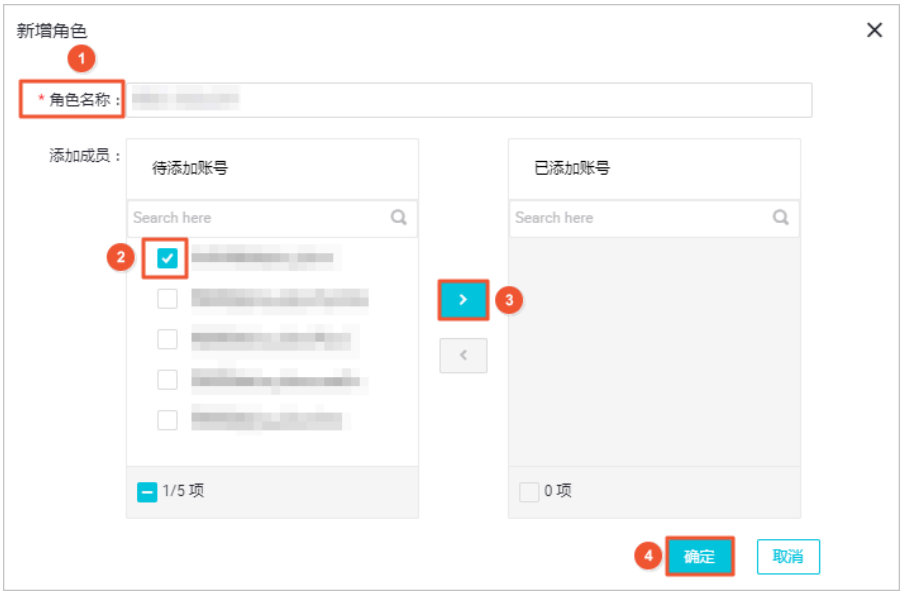
MaxCompute安全设置：涉及底层MaxCompute相关的权限及安全设置，详情请参见#unique_473。

配置	说明
使用ACL授权	激活/冻结该开关，相当于Owner账号在MaxCompute Project中执行`set CheckPermissionUsingACL=true/false`操作，默认激活。
允许对象创建者访问对象	激活/冻结该开关，相当于Owner账号在MaxCompute Project中执行`set ObjectCreatorHasAccessPermission=true/false`操作，默认激活。
允许对象创建者授权对象	激活/冻结该开关，相当于Owner账号在MaxCompute Project中执行`set ObjectCreatorHasGrantPermission=true/false`操作，默认激活。
项目空间数据保护	激活/冻结该开关，相当于Owner账号在MaxCompute Project中执行`set ProjectProtection=true/false`，默认冻结。
子账号服务	激活/冻结该开关，控制子账号可以访问/禁止访问该MaxCompute Project，默认激活。
使用Policy授权	激活/冻结Policy授权机制，相当于Owner账号在MaxCompute project中执行`set CheckPermissionUsingPolicy=true/false`操作，默认激活。
启动列级别访问控制	项目空间中的LabelSecurity安全机制默认关闭，ProjectOwner可以自行开启，相当于Owner账号在MaxCompute Project中执行`Set LabelSecurity=true/false`。

自定义用户角色



配置	说明
角色名称	MaxCompute项目中的角色名称。

配置	说明
操作	· 查看详情：查看当前角色中包含的成员列表，以及当前角色对表或项目的权限。 · 成员管理：添加或删除当前角色中的成员。 · 权限管理：设置并管理当前角色对表/项目的权限，详情请参见#unique_474。 · 删除：删除当前角色。
新增角色	单击右上角的新增角色，在新增角色对话框中填写角色名称，在待添加账号处勾选需要添加的成员账号，单击>，将需要添加的账号移动至已添加的账号中，单击确定，即可添加成功。 

说明:
此处自定义角色所设置的权限，将与原有的系统权限取并集。

5.5 项目模式升级

DataWorks V2.0推出了标准项目模式。标准项目模式下，一个DataWorks空间对应两个MaxCompute项目，将开发环境和生产环境隔离，增加任务的发布流程，保证任务代码的正确性。

标准模式优势

在过去老版本DataWorks中，您创建的都是一个DataWorks工作空间对应一个MaxCompute项目，即DataWorks V2.0中的简单模式。简单模式直接导致了表权限不可控，例如：只想允许工作空间中的某些人查询部分表，这个场景在简单模式下无法直接实现。因为一个DataWorks工作空间对应一个MaxCompute项目，DataWorks的开发角色权限在MaxCompute项目下有全部表的操

作权限，所以无法对表权限进行精准控制。这种情况下，您必须要新建一个单独的DataWorks工作空间，使用工作空间隔离的方式，完成数据的隔离。

DataWorks V1.0针对表权限控制的场景，衍生了一个方案：手动给两个DataWorks工作空间进行发布绑定。将A工作空间设置为B工作空间的发布工作空间，A工作空间即可接收到B工作空间中发布过来的任务，无需直接开发代码。这使得A工作空间成为类似生产环境的工作空间，B工作空间则是类似开发环境的工作空间。

在两个DataWorks工作空间绑定的模式下，也会有一些漏洞：A工作空间是一个正常的DataWorks工作空间，可以直接在数据开发模块中进行任务开发的，导致（生产）环境的代码更新入口不唯一，整个开发流程上会有逻辑漏洞。

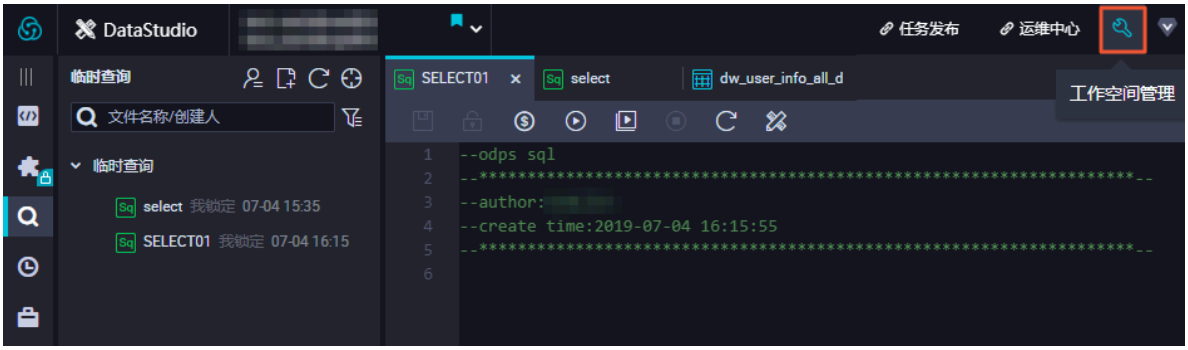
针对上述问题，DataWorksV2.0推出了标准模式。在标准模式下，能给数据开发者们带来几个好处：

1. 一个DataWorks工作空间对应两个MaxCompute项目，可以将开发和生产的计算引擎分开。项目成员只拥有开发环境的权限，默认无权限操作生产环境的表，提升了生产环境数据的安全性。
2. 标准模式下，数据开发界面默认操作开发环境的任务，生产环境的任务都是通过发布功能将任务发布至生产的。保证生产环境代码编辑入口的唯一性，提升生产环境代码的安全性。
3. 标准模式下，开发环境默认不进行周期调度，可以减少账号下的计算资源消耗，保障生产环境任务运行的资源。

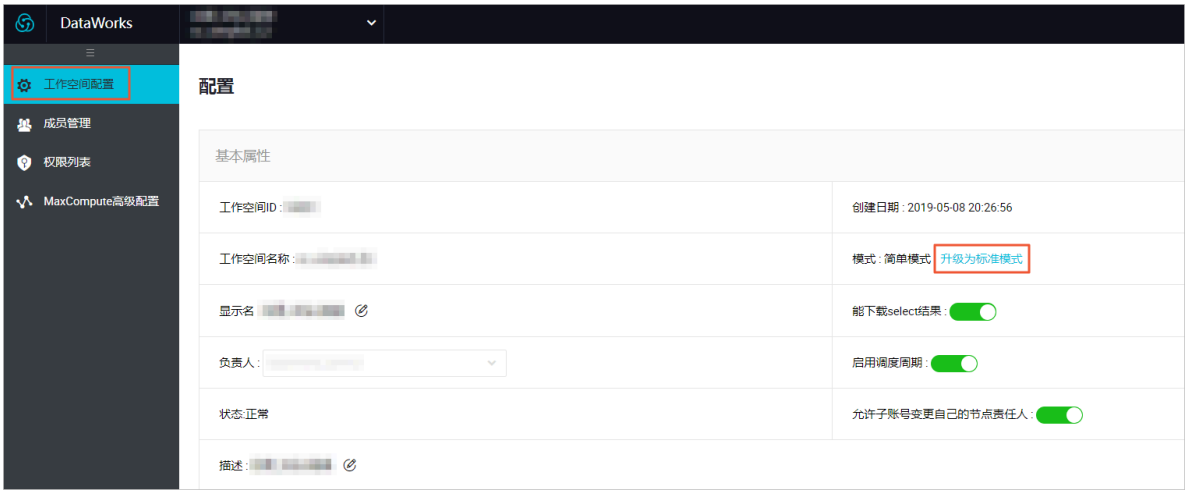
项目模式升级

在DataWorks V1.0中，创建的都是简单模式工作空间。主账号可以通过下述操作，将简单模式的工作空间升级为标准模式。

1. 进入数据开发页面，单击右上角的工作空间管理。



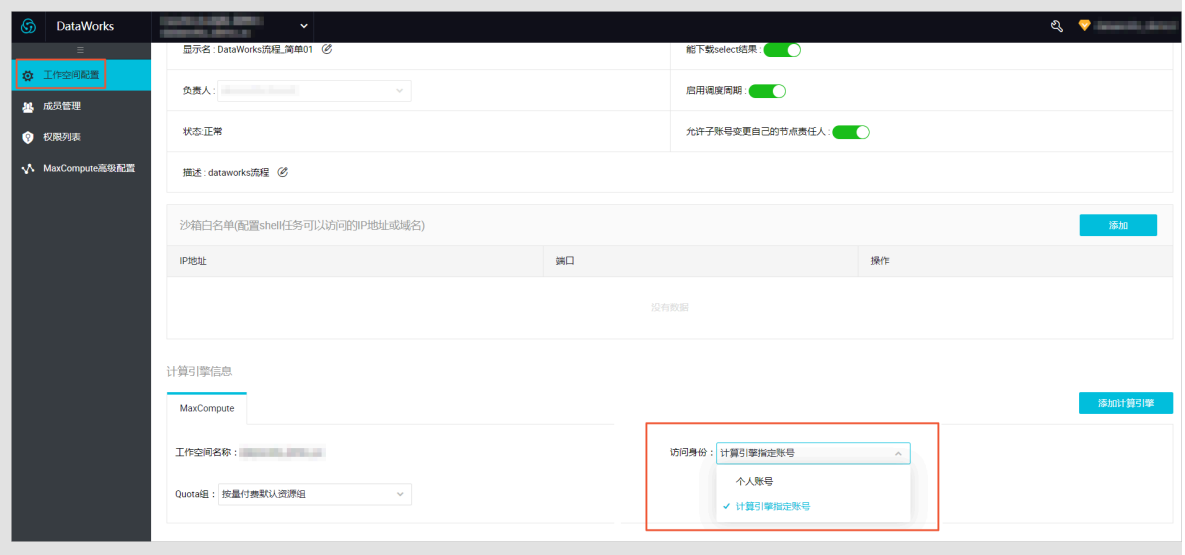
2. 在工作空间配置页面，单击模式后的升级为标准模式。



3. 在升级为标准模式对话框中，填写开发环境下的MaxCompute项目名称，并勾选确认要升级此工作空间，单击确定。

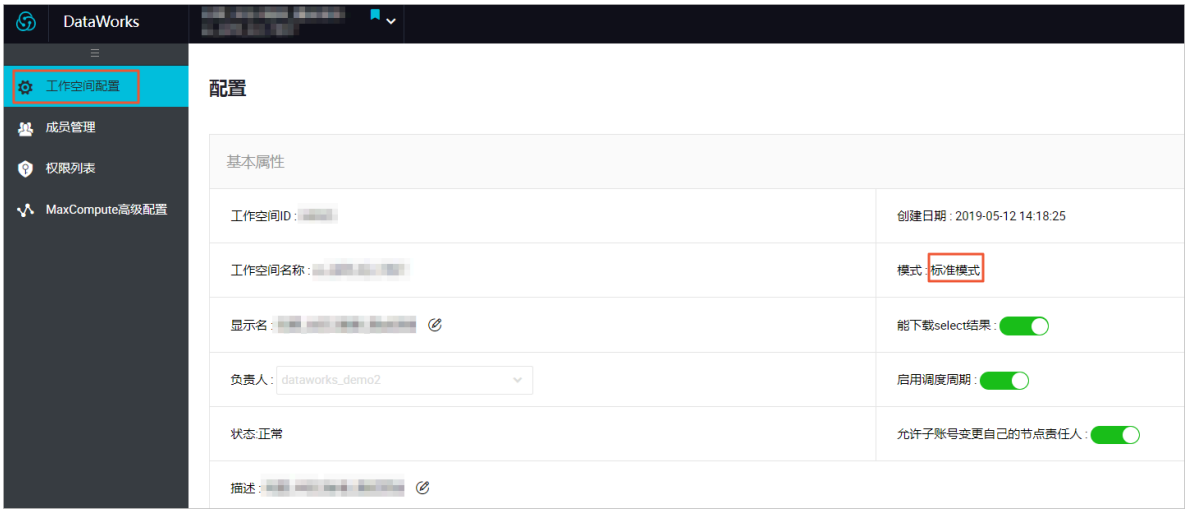
说明:

此处您需要新建一个简单模式工作空间作为开发环境下的MaxCompute项目，并可以在工作空间配置 > 计算引擎信息页面，选择使用个人账号或计算引擎指定账号操作生产环境的数据。



4. 单击确认升级提示框中的确认。

完成上述操作后，即可返回工作空间配置页面，查看该工作空间的模式已显示为标准模式。



5. 在升级为标准模式对话框中，填写开发环境的项目名称，单击确定。

简单模式升级为标准模式的影响

升级为标准模式后，DataWorks会将原项目中的成员加入新建的MaxCompute开发项目中，并将原项目的成员和角色都保留。但会撤销项目成员在生产项目的权限，只有项目所有者具有生产项目的权限。

例如，某公司在DataWorks上有一个A项目，单击升级为标准模式后，创建了一个开发环境的项目A_dev。原来A项目中的成员、角色、表、资源都会在A_dev项目下创建一份（只创建表，不会把表数据也克隆一份）。原A项目下的成员A1（开发角色）、B1（运维角色），同时也会加入到A_dev项目下，并保留角色权限。A项目会变成生产项目，A1和B1用户在A项目中的数据权限会被撤销，默认没有表的select和drop权限，生产项目的数据会直接受到保护。

在DataStudio（数据开发）界面，默认操作的MaxCompute项目是A_dev。如果要在数据开发界面查询生产环境的数据，需要使用项目名.表名的方式。数据开发界面只能编辑A_dev环境的代码，要更新A项目中的代码只能通过A_dev提交任务至调度系统，发布至生产环境的方式进行更新。增加了一个任务发布（审核）的流程，保障了生产环境代码的正确性。

说明:

项目模式升级后，无法直接访问原项目的数据，需要申请角色权限。在数据开发界面查询的表，默认是开发环境的表。如果要访问生产表，需要申请完角色权限后，使用项目名.表名的方式访问。

升级到标准项目模式后，会将该子账号之前的角色清除。如果您的代码中写了某个账号的AK，可能会报错没有权限的问题。

6 数据质量

6.1 数据质量概述

DataWorks数据质量（DQC）是支持多种异构数据源的质量校验、通知、管理服务的一站式平台。

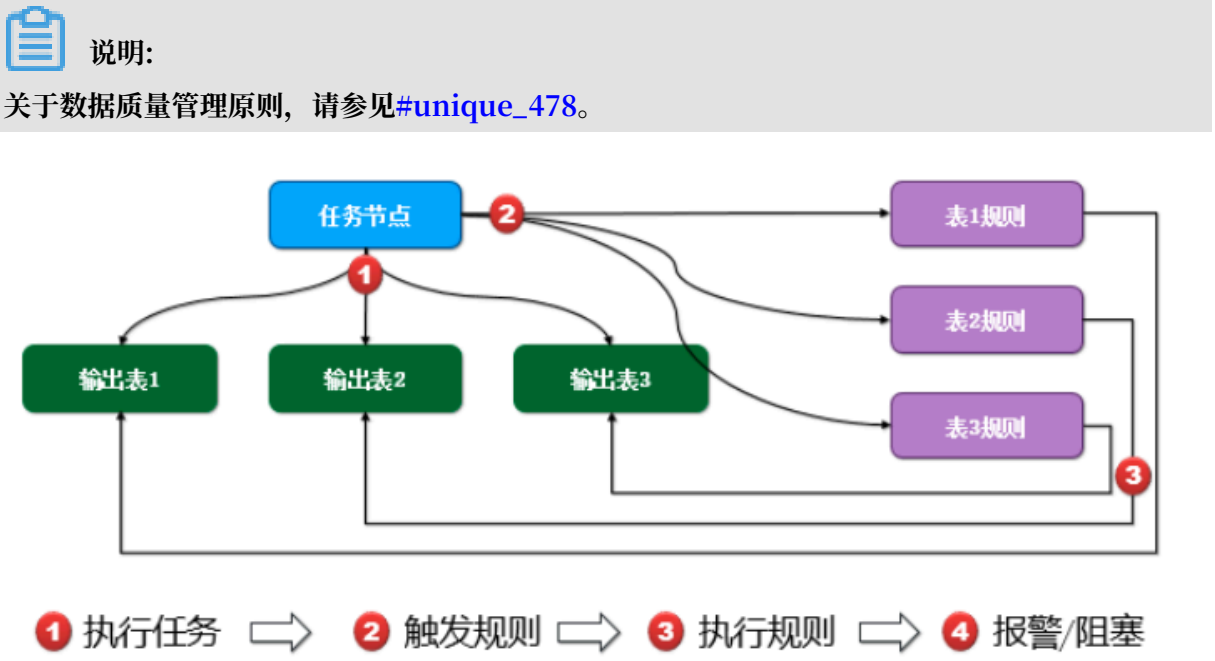
DataWorks数据质量依托DataWorks平台，为您提供全链路的数据质量方案，包括数据探查、数据对比、数据质量监控、SQL扫描和智能报警等功能。

数据质量监控可全程监控数据加工流水线，根据质量规则及时发现问题，通过报警通知负责人及时处理。

数据质量以数据集（DataSet）为监控对象。目前，数据质量支持MaxCompute数据表和DataHub实时数据流的监控。当MaxCompute离线数据发生变化时，数据质量会对数据进行校验，并阻塞生产链路，以避免问题数据污染扩散。同时，数据质量提供了历史校验结果的管理，以便您对数据质量进行分析和定级。

在流式数据场景下，数据质量能够基于Datahub数据通道进行监控和断流，第一时间告警给订阅用户。数据质量还支持橙色、红色告警等级以及告警频次的设置，最大限度地减少冗余报警。

数据质量监控的流程如下图所示。



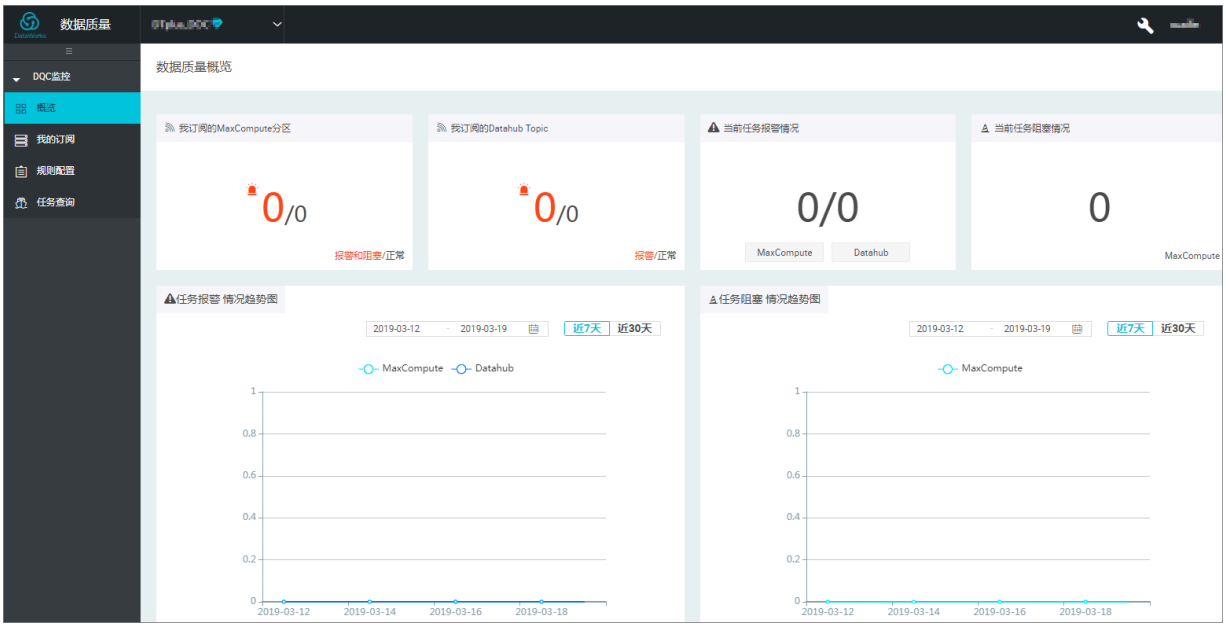
数据质量主要对MaxCompute和DataHub数据集的质量进行监控。因此，您需要先创建表，并在表中写入数据后才能使用数据质量功能。

您可以通过MaxCompute客户端或DataWorks控制台创建MaxCompute表并写入数据。

6.2 功能介绍

6.2.1 首页概览

数据质量首页的概览页面为您展示订阅数据的报警以及任务的阻塞情况，您可以查看经过筛选的结果。



模块	说明
我订阅的MaxCompute分区	显示当天订阅的MaxCompute分区的报警和阻塞、正常两种情况。单击此模块可快速跳转至MaxCompute数据源的任务查询页面，查看报警详情。
我订阅的DataHub Topic	显示当天订阅的DataHub数据源报警、正常两种情况，单击此模块可快速跳转至DataHub数据源的任务查询页面，查看报警详情。
当前任务报警情况	显示当天、当前应用下的MaxCompute和DataHub两种数据源的任务报警情况。
当前任务阻塞情况	显示当天、当前应用下MaxCompute数据源的任务阻塞情况。
任务报警情况趋势图	可选7天、30天以及自定义时间区间，支持日期范围为近三个月内的MaxCompute和DataHub数据源的任务报警趋势图。
任务阻塞情况趋势图	可选7天、30天以及自定义时间区间，支持日期范围为近三个月内的MaxCompute的任务阻塞情况。

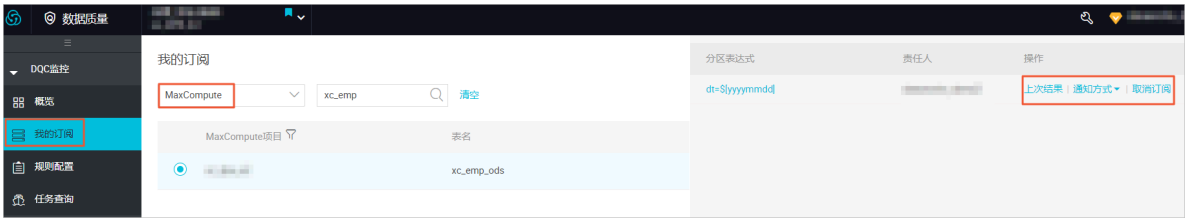
6.2.2 我的订阅

我的订阅页面为您展示当前账号订阅的所有任务。

当前，数据质量支持MaxCompute监控和Datahub监控，您可以在我的订阅页面选择相应的数据源，查找自己订阅的任务。

您可以选择以下两种数据源：

- MaxCompute数据源



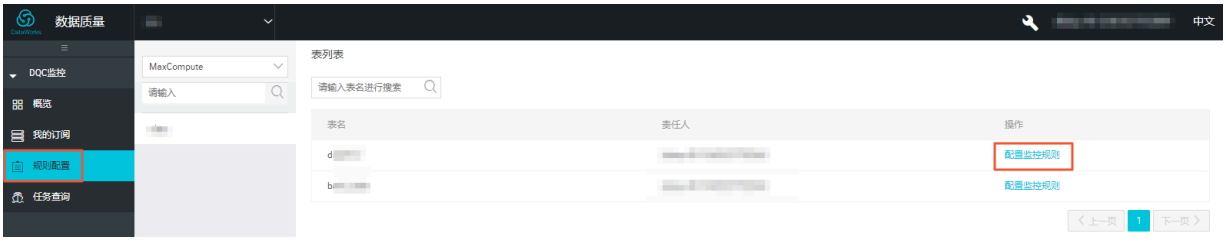
  - 单击右侧相应的分区表达式，即可进入规则配置页面，详情请参见MaxCompute监控。
  - 单击上次结果，可跳转至任务查询页面。
  - 目前支持邮件通知、邮件和短信通知、钉钉群机器人和钉钉群机器人@ALL四种通知方式。
  - 单击取消订阅，即可删除相应的订阅信息。
- DataHub数据源

  - 单击对应Topic操作栏下的配置监控规则，即可进入规则配置页面，详情请参见DataHub监控。
  - 单击对应Topic操作栏下的取消订阅，即可取消已订阅的Topic。

6.2.3 规则配置

目前数据质量支持离线的MaxCompute监控和DataHub监控，本文将为您介绍如何配置MaxCompute规则。

- 选择MaxCompute数据源，即可显示当前数据源下所有的表。您也可以使用搜索功能，快速定位至其他数据源下查看表。
- 选择Datahub数据源，即可显示当前数据源下所有的Topic。您也可以使用搜索功能，快速定位至其他数据源下查看Topic。



单击配置监控规则，即可进入规则配置页面。

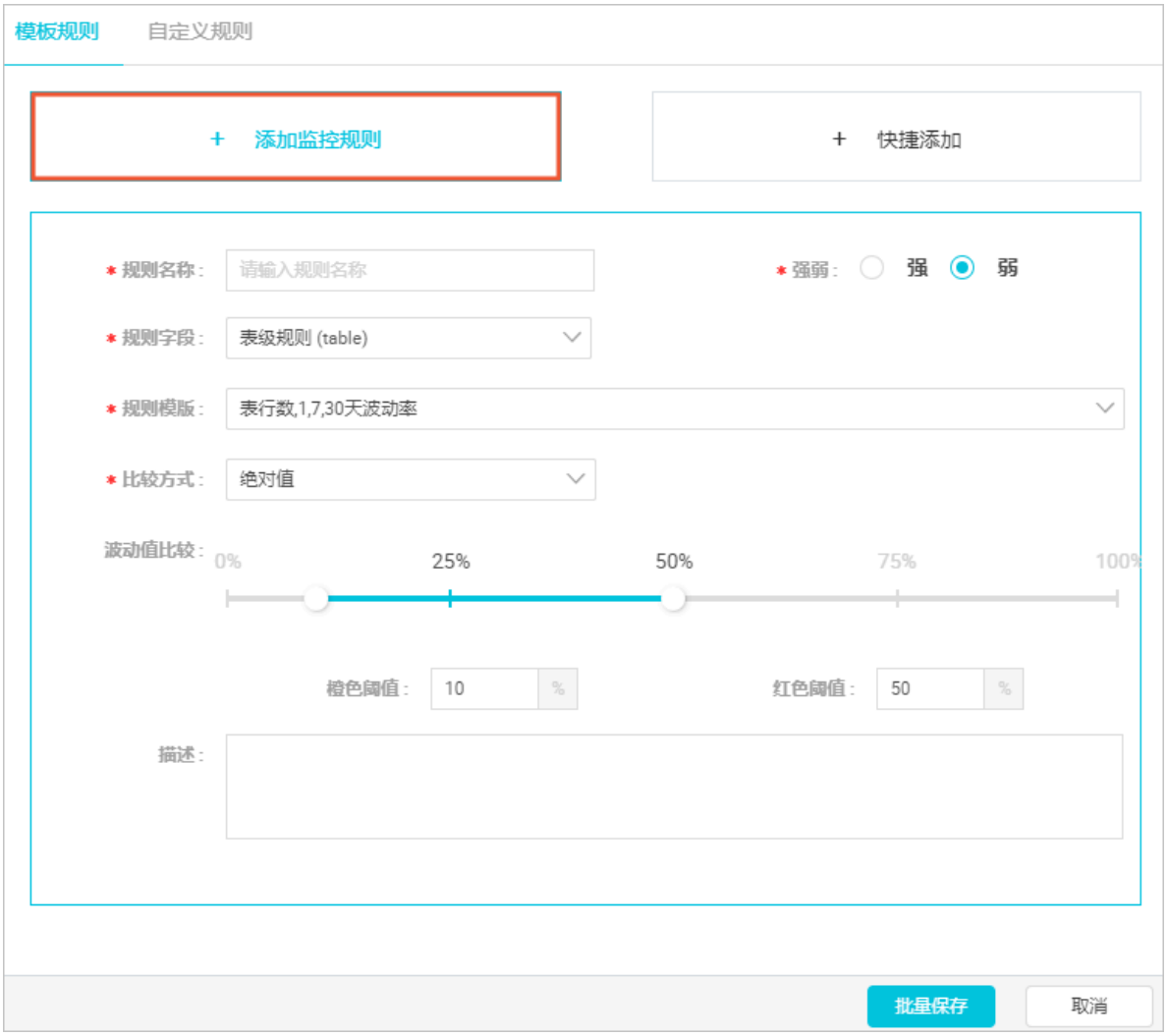
目前数据质量规则配置包括模板规则和自定义规则。

> **说明:**
> 配置模板规则前，需首先配置分区表达式，详情请参见#unique_485/unique_485_Connect_42_section_bby_bb3_r2b。

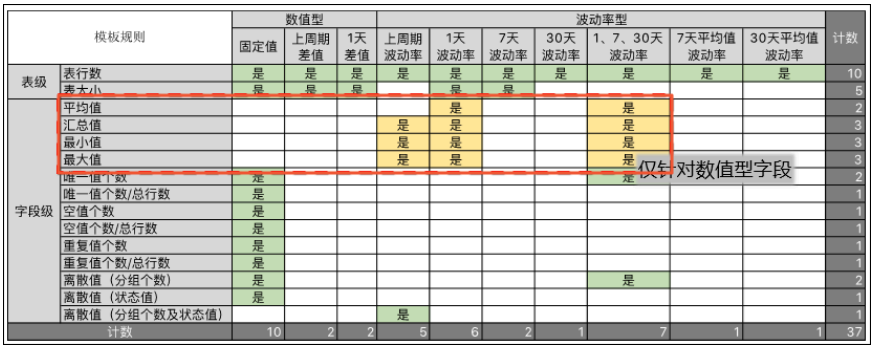
模板规则

您可以通过添加监控规则和快捷添加2种方式创建模板规则。

- 添加监控规则



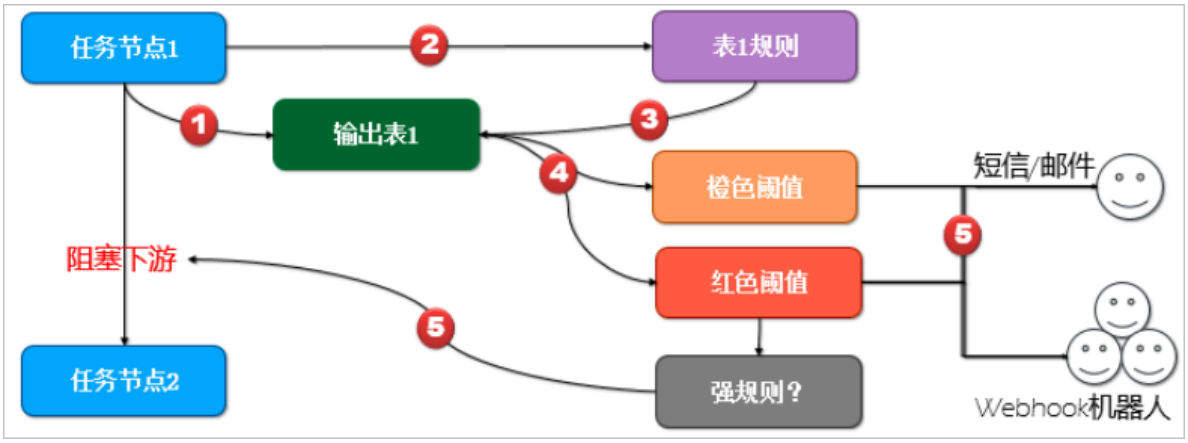
配置	说明
规则名称	请输入规则名称。
强弱	设置强规则或弱规则： - 如果设置强规则，红色异常报警并阻塞下游任务节点，橙色异常报警不阻塞。 - 如果设置弱规则，红色异常报警不阻塞下游任务节点，橙色异常不报警不阻塞。
规则字段	包括表级规则和字段级规则，字段级规则包括数据类型和非数据类型。

配置	说明
规则模板	目前共有37种规则，不支持的规则模板将不能被选择。 您可以单击下拉框选择相关的模板，支持的模板详情请参见下表。  仅针对数值型字段 **说明:** 目前平均值、汇总值、最小值和最大值仅对数值型字段生效。
比较方式	包括绝对值、上升和下降。

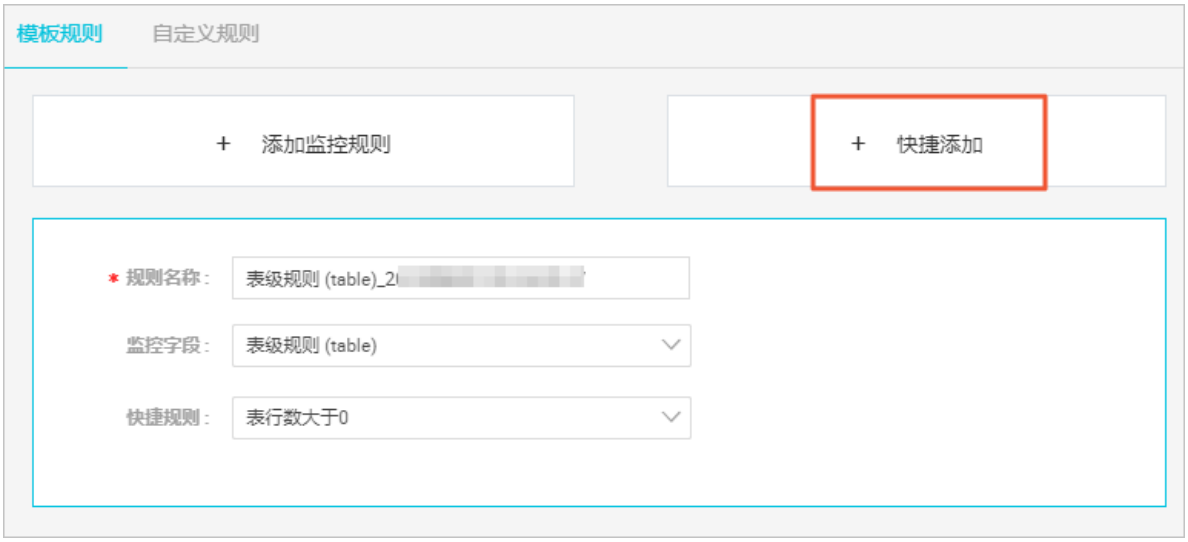
模板规则		数值型			波动率型							计数
		固定值	上周期差值	1天差值	上周期波动率	1天波动率	7天波动率	30天波动率	1、7、30天波动率	7天平均值波动率	30天平均值波动率	
表级	表行数	是	是	是	是	是	是	是	是	是	是	10
	表大小	是	是	是		是	是					5
字段级	平均值					是			是			2
	汇总值				是	是			是			3
	最小值				是	是			是			3
	最大值				是	是			是			3
	唯一值个数	是							是			2
	唯一值个数/总行数	是										1
	空值个数	是										1
	空值个数/总行数	是										1
	重复值个数	是										1
	重复值个数/总行数	是										1
	离散值（分组个数）	是							是			2
	离散值（状态值）	是										1
	离散值（分组个数及状态值）				是							1
计数		10	2	2	5	6	2	1	7	1	1	37

配置	说明
波动值比较	- 计算波动率，您可以根据波动率计算公式计算结果。 波动率=（样本-基准值）/基准值。 - 计算方差波动 （当次样本-历史N天平均值）/标准差，仅BIGINT和DOUBLE等数值类型可以使用方差。 说明: 样本和基准值的名词解释如下： - 样本：当天采集的具体的样本的值。例如对于SQL任务表行数，1天波动检测，则样本是当天分区的表行数。 - 基准值：历史样本的对比值。 ■ 如果规则是SQL任务表行数，1天波动检测，则基准值是前一天分区产生的表行数。 ■ 如果规则是SQL任务表行数，7天平均值波动检测，则基准值是前7天的表行数据的平均值。 您可以设置橙色阈值和红色阈值，对不同严重程度的问题进行监控。 - 如果校验值的绝对值小于或等于橙色阈值，则返回正常。 - 如果校验值的绝对值不满足第1种情况，且小于或等于红色阈值，则返回橙色报警。 - 如果校验值不满足第2种情况，则返回红色报警。 - 如果没有橙色阈值，则只有红色报警和正常2种情况。 - 如果没有红色阈值，则只有橙色报警和正常2种情况。 - 两个都不填，则红色报警（通常禁止两个阈值都不填，阈值校验会默认橙色10%，红色50%）。

下图为报警与阻塞的实现逻辑。



- 快捷添加



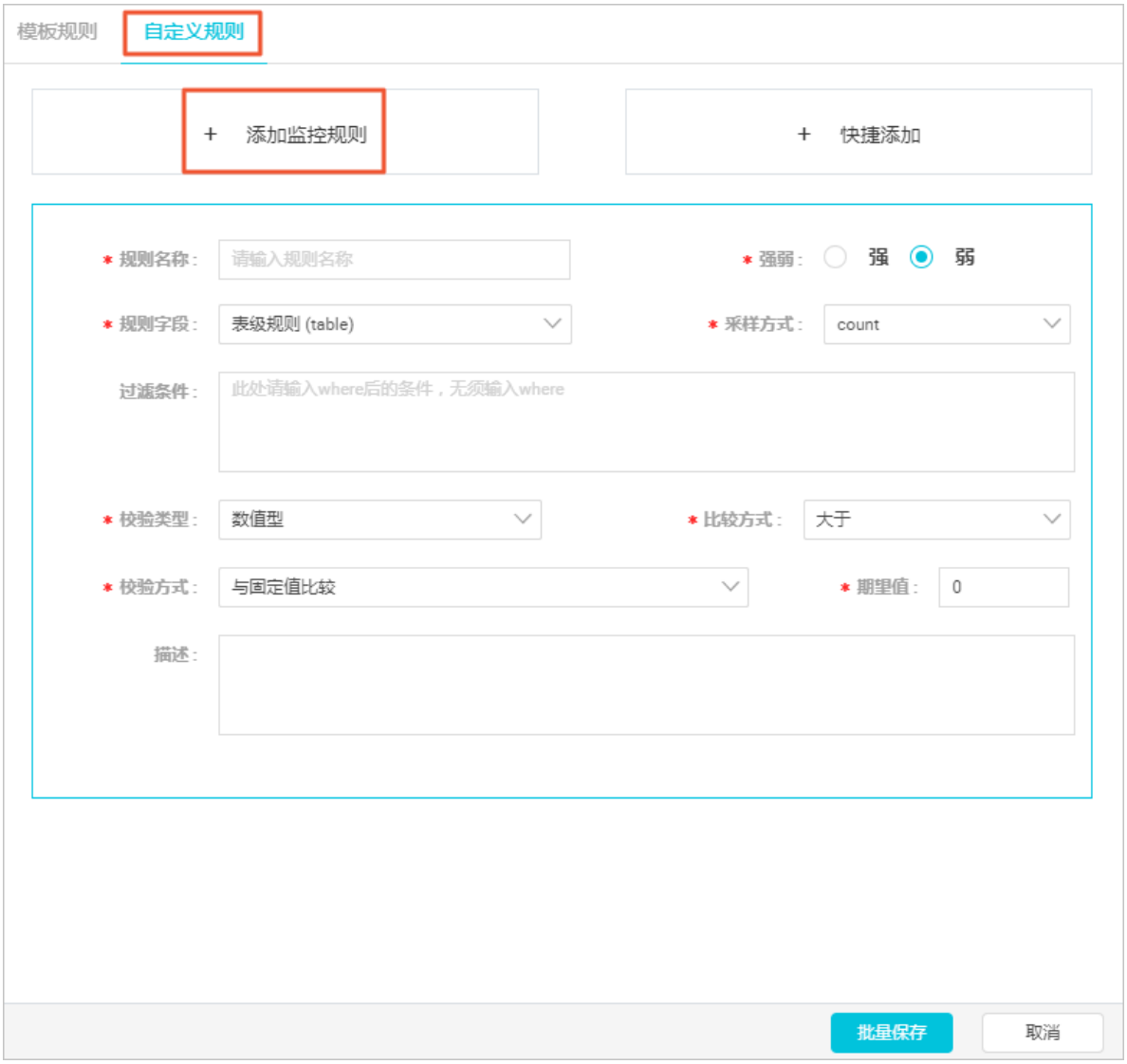
配置	说明
规则名称	请输入规则名称。
监控字段	包括表级规则和字段级规则，字段级规则包括数据类型和非数据类型。
快捷规则	默认表行数大于0。

自定义规则

如果模板规则不能满足您对分区表达式中数据质量的监控需求，您还可以通过创建自定义规则来满足个性化的监控需求。

您可以通过添加监控规则和快捷添加2种方式创建自定义规则。

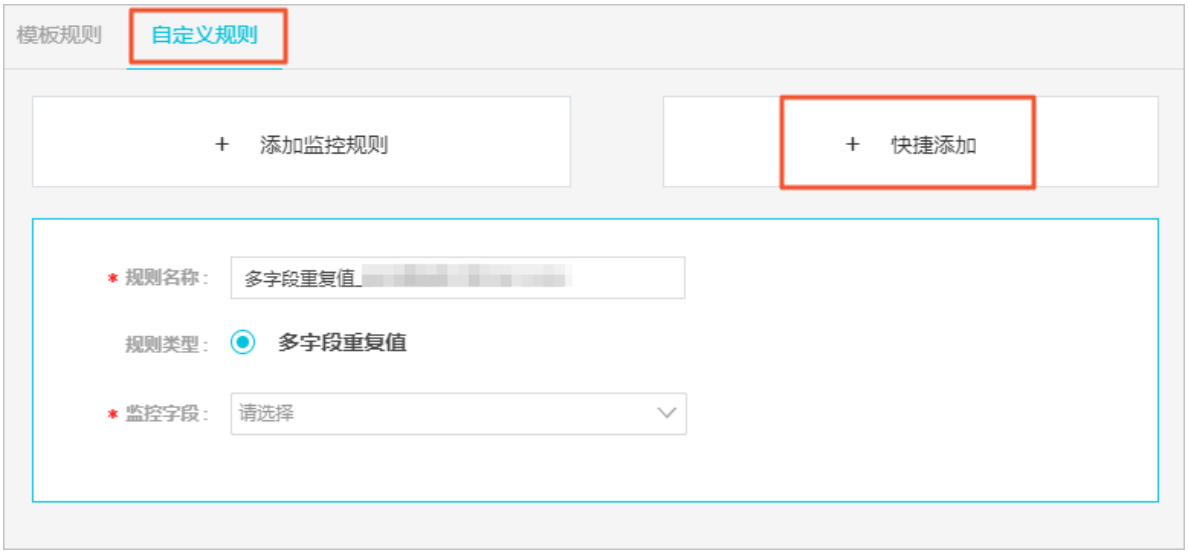
- 添加监控规则



配置	说明
规则名称	请输入规则名称。
规则字段	支持表级规则、自定义SQL和字段级规则。 - 表级、字段级自定义规则，支持根据业务属性自定义where过滤条件。 - 自定义SQL支持自定义SQL逻辑（单行单列输出）。
强弱	设置强规则或弱规则： - 如果设置强规则，红色异常报警并阻塞下游任务节点，橙色异常报警不阻塞。 - 如果设置弱规则，红色异常报警不阻塞下游任务节点，橙色异常不报警不阻塞。
采样方式	支持count和count/table_count2种方式。

配置	说明
过滤条件	例如：您需要查询业务日期下表的分区，可以将where条件设置为`pt=$[yyyymmdd-1]`。 
校检类型	支持数值型和波动率型2种类型。
比较方式	包括大于、大于等于、等于、不等于、小于和小于等于6种比较方式。
校检方式	目前支持与固定值比较。
期望值	设置期望值。
描述	对创建的自定义规则进行描述。

- 快捷添加



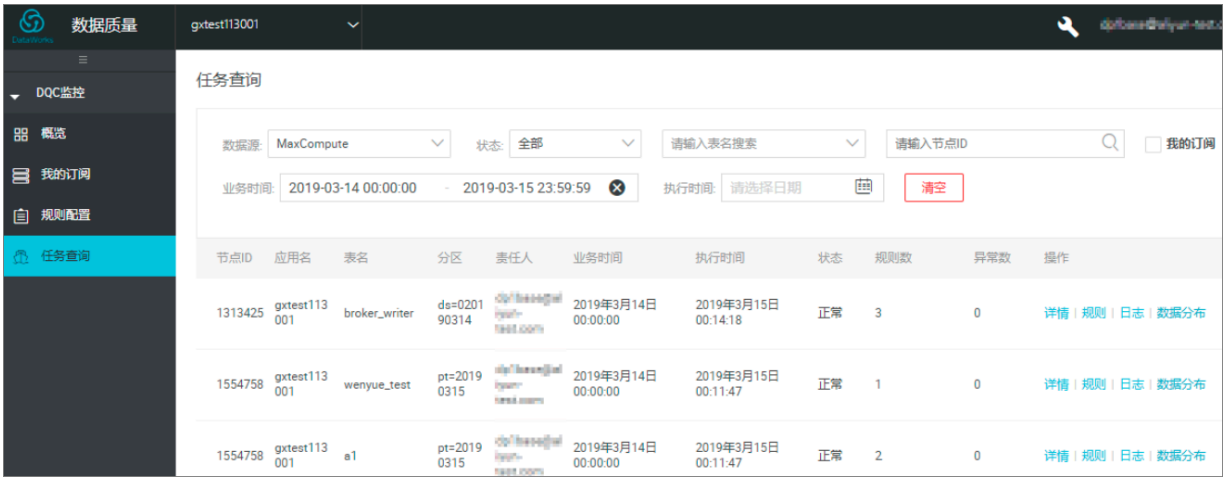
配置	说明
规则名称	请输入规则名称。
规则类型	仅支持多字段重复值。
监控字段	设置监控字段。

6.2.4 任务查询

任务查询模块主要显示规则校验结果，规则运行后，即可在任务查询模块查看运行记录。

查看MaxCompute监控任务

您可以在任务查询页面，根据状态、表名、节点ID等信息进行搜索，查看任务运行情况。右侧对应的蓝色字体可以链接到其他页面，方便您进行查看和修改。

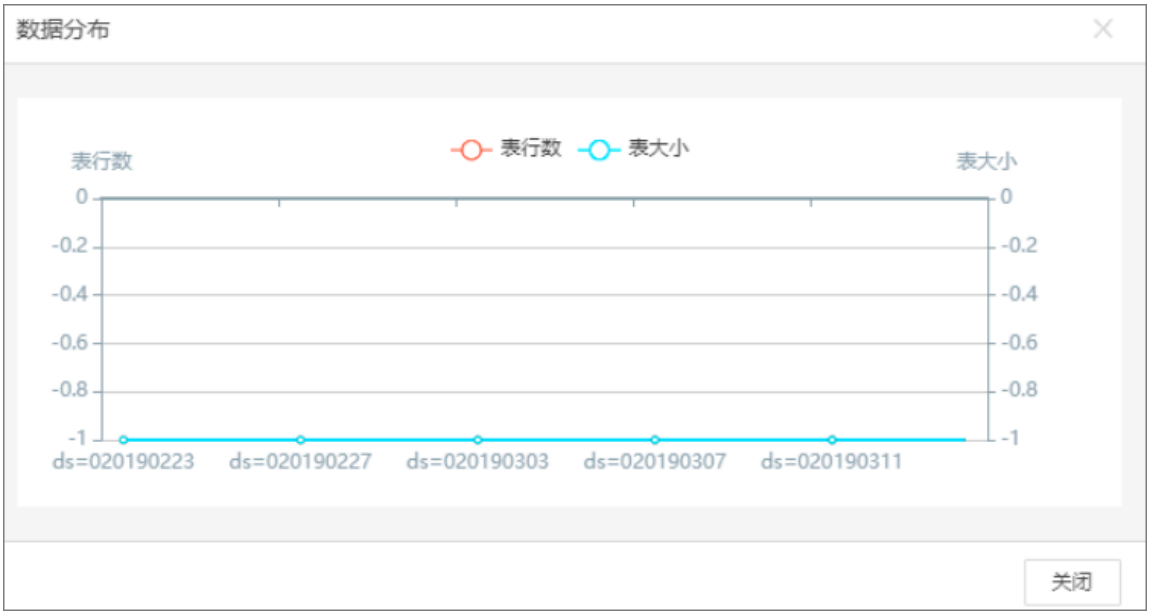


- **节点ID：触发规则的任务节点。**
- **执行时间：规则执行时间。**
- **状态：执行结果的状态，如果是报警或阻塞状态，需要多加注意。**

- 操作：
  - 详情

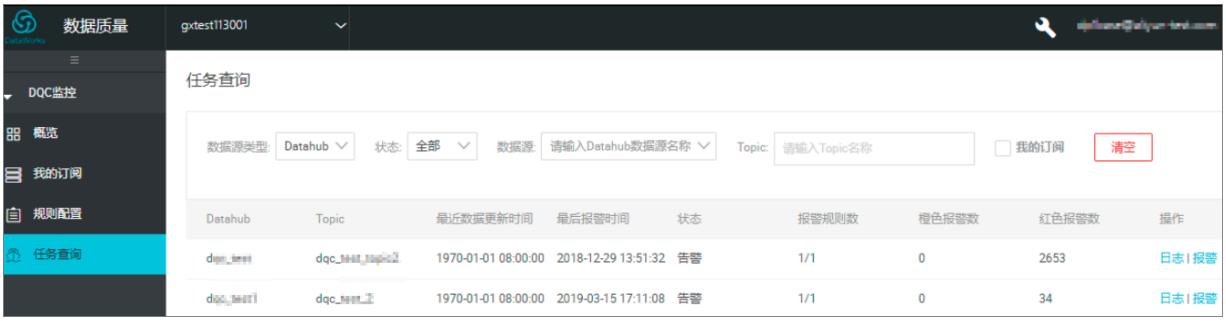
    ■ 单击更多，即可查看数据源、应用名、节点ID和责任人等信息。

    ■ 单击对应字段后的查看历史记录，即可查看每次调度后的运行记录。
  - 规则：单击即可跳转至规则配置页面。可在此查看之前创建的分区表达式和规则，并进行相应的修改。详情请参见#unique_482。
  - 日志：查看规则运行日志（高级）。
  - 数据分布：一键探查数据量、表行数，单击对应任务后的数据分布，即可查看该任务从创建至今，每次运行的情况。



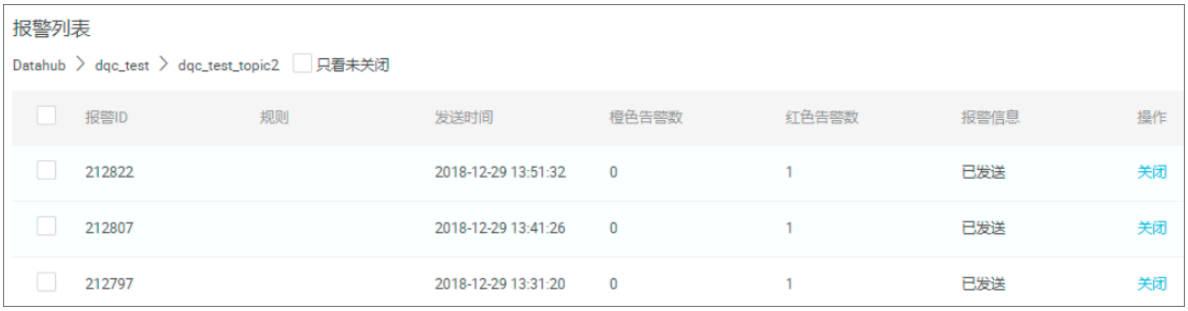
查看DataHub监控任务

您可以在任务查询页面，根据状态、表名、节点ID等信息进行搜索，查看任务运行情况。右侧对应的蓝色字体可以链接到其他页面，方便您进行查看和修改。



操作：

- **日志：查看规则运行日志（高级）。**
- **报警：单击相应Topic右侧的报警，可查看任务运行报警详情，您也可以在详情页面关闭报警。**



6.3 使用指南

6.3.1 DataHub监控

规则配置模块是数据质量（DQC）中最核心的部分，当前支持MaxCompute监控和DataHub监控。本文将为您介绍如何配置DataHub监控。

DataHub实时数据监控当前支持以下功能：

- **支持数据断流、数据延迟两种监控模板。**
- **自定义Flink SQL、维表join、多流join以及窗口函数等流计算特性。**

添加数据源

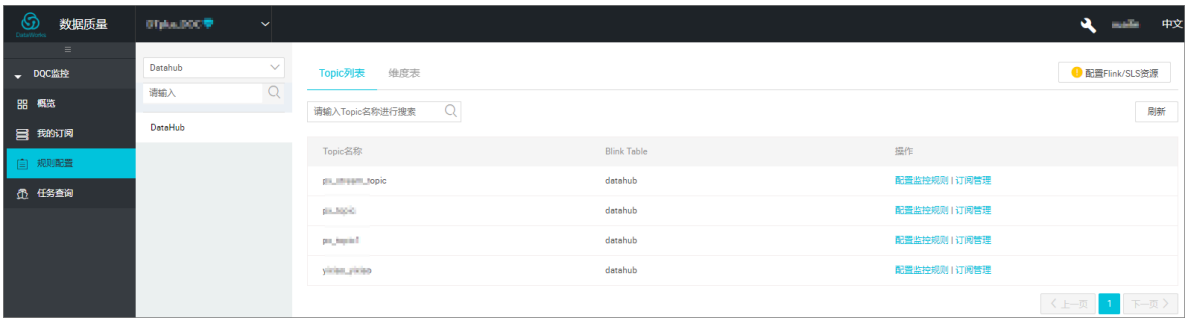
您需要首先进入数据集成页面添加数据源，详情请参见配置Datahub数据源。新建数据源成功后，即可进入数据质量（DQC）页面进行规则的配置。

选择数据源

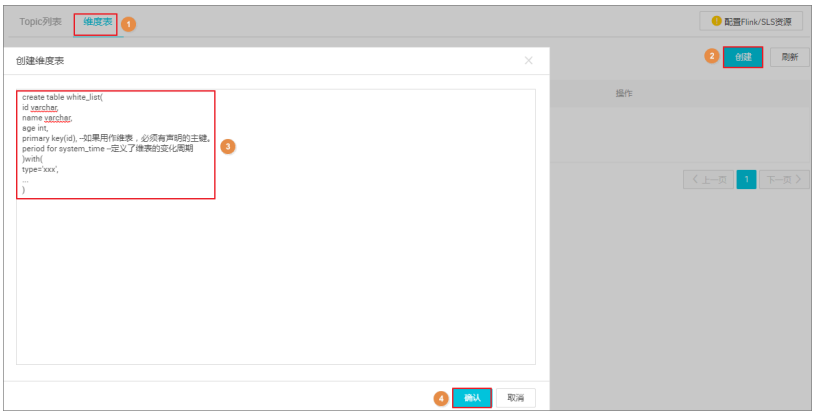
1. 单击左侧导航栏的规则配置，进入规则配置页面。

2. 选择DataHub数据源，即可显示当前数据源下所有的Topic。

   您也可以通过搜索功能，快速定位到其他数据源下查看Topic。



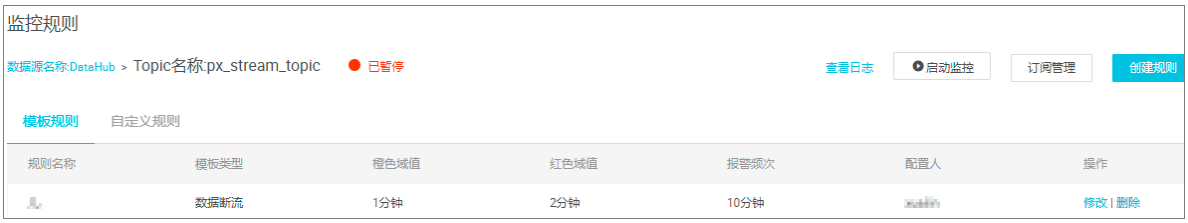
配置	说明
配置Flink/SLS规则	添加数据源后Flink/SLS资源会根据数据源拉取相关的信息。
Topic列表	DataHub数据源下所有Topic的名称，您可以在相应Topic后进行下述操作。 · 配置监控规则：对当前Topic创建规则，支持创建模板规则和自定义规则。 · 订阅管理：查看当前Topic的订阅人，可以快捷修改订阅人和报警方式，也可以配置钉钉群报警，这里修改的报警方式对所有订阅人有效。

配置	说明
维度表	对Topic创建自定义规则join时使用。如果采集到的数据有限，则需要对数据流补齐字段。进行数据分析前，将所需的维度信息进行补全，此时需要在数据质量中对这张维度表进行声明。 DataHub支持AliHBase维表、Lindorm维表、RDS维表、OTS（TableStore）维表、TDDL维表、ODPS维表。  Flink SQL中没有专门为维表设计的DDL语法，使用标准的create table语法即可。但需要额外增加一行`period for system_time`的声明，此行声明定义了维表的周期，即表明该表是一张会变化的表。 **说明:** 声明一张维表时，必须指明唯一键。维表join时，on的条件必须包含所有唯一键的等值条件。

3. 在Topic列表页面选择需要配置的Topic，单击右侧的配置监控规则。

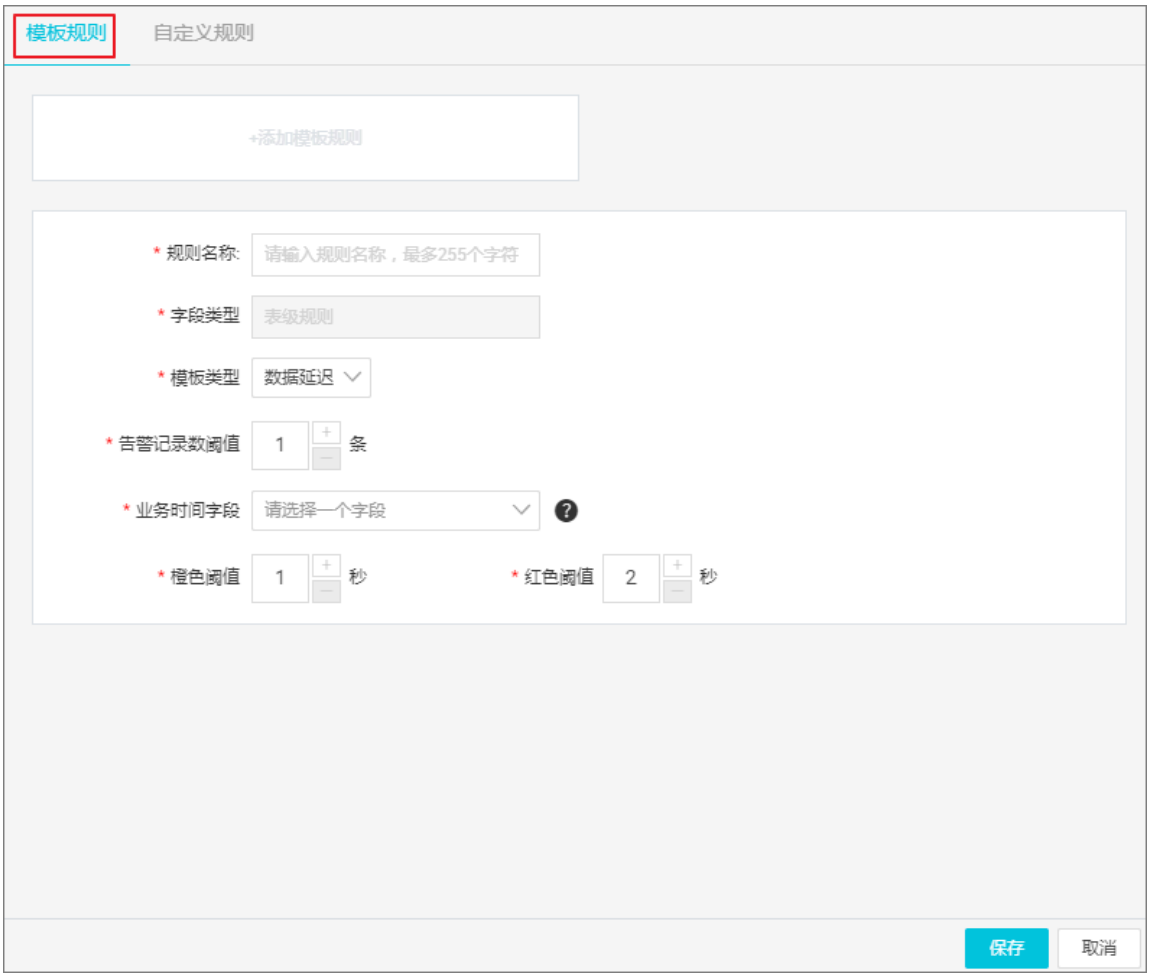
配置监控规则

1. 单击监控规则页面右上角的创建规则，当前支持模板规则和自定义规则两种类型。



- 单击添加模板规则，目前模板类型包括数据延迟和数据断流。

  例如选择模板类型为数据延迟。

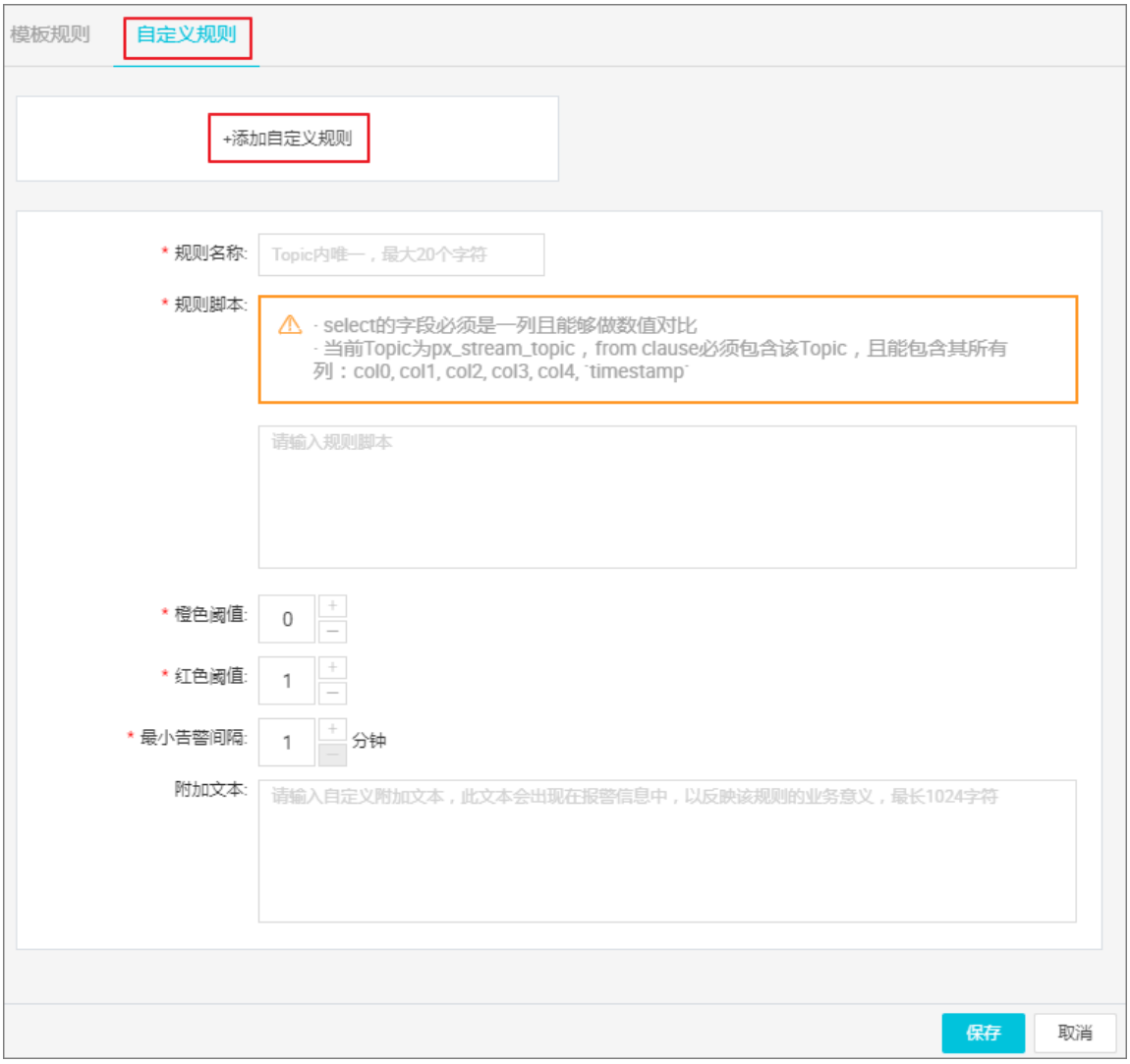


配置	说明
规则名称	输入规则名称，最多255个字符。
字段类型	默认为表级规则。

配置	说明
模板类型	- 数据延迟：记录业务时间字段内，数据产生于流入DataHub通道的时间差，超过设定时间立即报警。 说明: 业务时间字段支持TIMESTAMP和STRING（yyyy -MM -dd H dd HH:mm:ss）两种类型。 - 数据断流：允许在某一时间段内没有数据流入，当超过允许时间，则触发告警。 说明: 配置数据断流前，请首先在Flink中购买服务并创建项目。
告警记录数阈值	允许出现数据延迟的数量上限，超过上限触发数据质量告警，只有模板选择数据延迟才会有此参数。
业务时间字段	Topic中时间字段的字段名称，支持TIMESTAMP和STRING（yyyy-MM-dd HH:mm:ss）两种类型，只有模板选择为数据延迟时才需配置本参数。
告警频次	告警频次包括10分钟、30分钟、1小时和2小时。
橙色阈值	以秒为单位，仅支持输入整数，且必须小于红色阈值。

配置	说明
红色阈值	以秒为单位，仅支持输入整数，且必须大于橙色阈值。

- 如果对DataHub规则有其他的使用方式，可单击添加自定义规则进行创建。



说明:

- select的字段必须是一列且能够与橙色阈值和红色阈值进行数值对比。
- 自定义规则下，from子句必须包含该Topic，且包含此Topic中所有的列。

配置	说明
规则名称	输入规则名称，需在Topic内唯一，最多支持20个字符。

配置	说明

规则脚本	自定义编写SQL来设定规则，select的结果字段必须唯一。
	- 示例一：简单SQL。
	`select id as a from zmr_tst02;`
	- 示例二：与维表join查询，维表名称test_dim。
	`select e.id as eid`
	`from zmr_test02 as e`
	`join test_dim for system_time as of`
	`proctime() as w`
	`on e.id=w.id`
	- 示例三：两个Topic进行join查询，另一个Topic名称dp1test_zmr01。
	`select count(newtab.biz_date) as aa`
	`from (select o.*`
	`from zmr_test02 as o`
	`join dp1test_zmr01 as p`
	`on o.id=p.id)newtab`
	`group by id.biz_date,biz_date_str,`
	`total_price,'timestamp'`

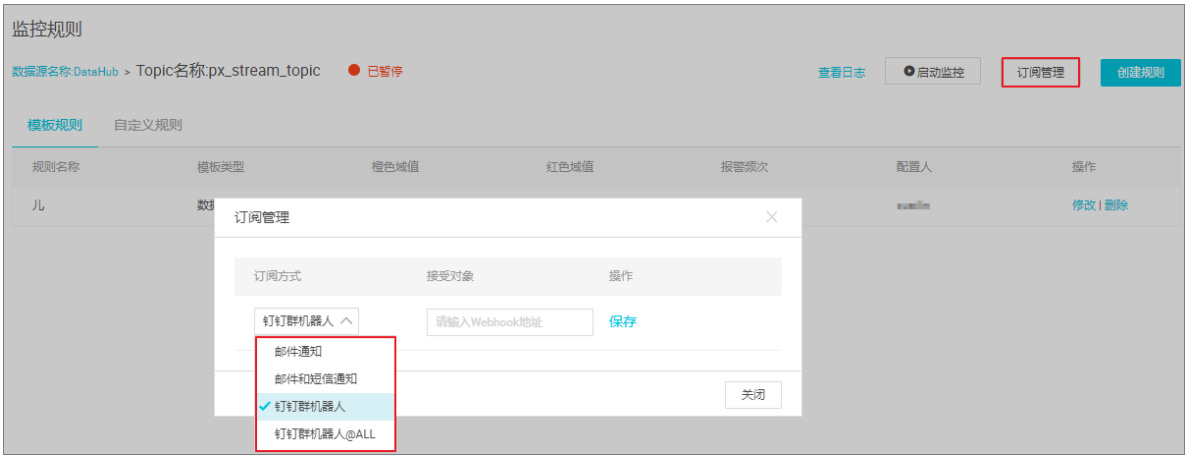
橙色阈值	以分钟为单位，仅支持输入整数，且必须小于红色阈值。
红色阈值	以分钟为单位，仅支持输入整数，且必须大于橙色阈值。
最小告警间隔	允许告警的最小时间差，以分钟为单位。
附加文本	对当前自定义Topic的描述。

2. 配置完成后，单击保存，将创建的规则添加到Topic中。

更多操作

- 单击查看日志，查看当前规则的运行日志。

- 单击订阅管理，您可以在该页面查看、修改当前规则的订阅人，也可以修改告警通知方式，对所有订阅人生效。



您可以将任务报警添加到钉钉群中，支持邮件通知、邮件和短信通知、钉钉群机器人和钉钉群机器人@ALL四种方式。

钉钉群机器人：先添加一个机器人到钉钉群中，通过详情页获取Webhook，然后将Webhook地址复制到订阅管理中，即可添加成功。

6.3.2 MaxCompute监控

规则配置模块是数据质量（DQC）的核心，目前数据质量支持离线的MaxCompute监控和DataHub监控，本文将为您介绍如何配置离线MaxCompute监控。

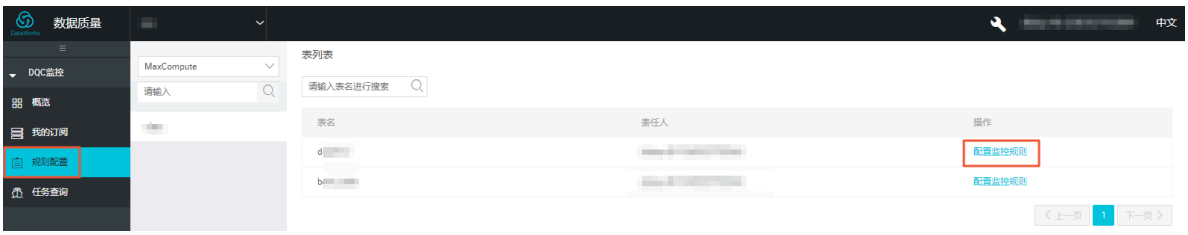
添加数据源

您需要首先进入数据集成页面添加数据源，详情请参见配置MaxCompute数据源。新建成功后，即可进入数据质量（DQC）页面进行规则的配置。

选择数据源

1. 单击左侧导航栏的规则配置，进入规则配置页面。
2. 选择MaxCompute，即可显示当前数据源下所有的表。

   您也可以输入对象表名（支持表名首字母模糊搜索），找到对应的表。
3. 单击右侧的配置监控规则。



配置分区表达式

数据质量用分区表达式来确定需要配置哪条规则。

> **说明:**
>
> - 如果您的检查对象为非分区表，则此处可填写为NOTAPARTITIONTABLE 。
> - 如果您的表为分区表，则可以配置为业务日期的表达式（如$[yyyymmdd]），同时也可以配置为正则表达式。

进入数据表的规则配置页面，单击左上角的+，添加分区表达式。

- 新建分区的表达式：单击左上角的+，会弹出分区配置窗口，您可以根据自身需求编辑符合语法的分区表达式。非分区表可以直接选择推荐的分区表达式中的NOTAPARTITIONTABLE。
  - 一级分区的表达式格式：分区名=分区值，分区值可以是固定值，也可以是内置参数表达式。分区表必须配置到最后一级分区。
  - 多级分区表达式格式：1级分区名=分区值/2级分区名=分区值/N级分区名=分区值，分区值可以是固定值，也可以是内置参数表达式。参数必须使用中括号表示，例如$[yyyymmdd-N]。

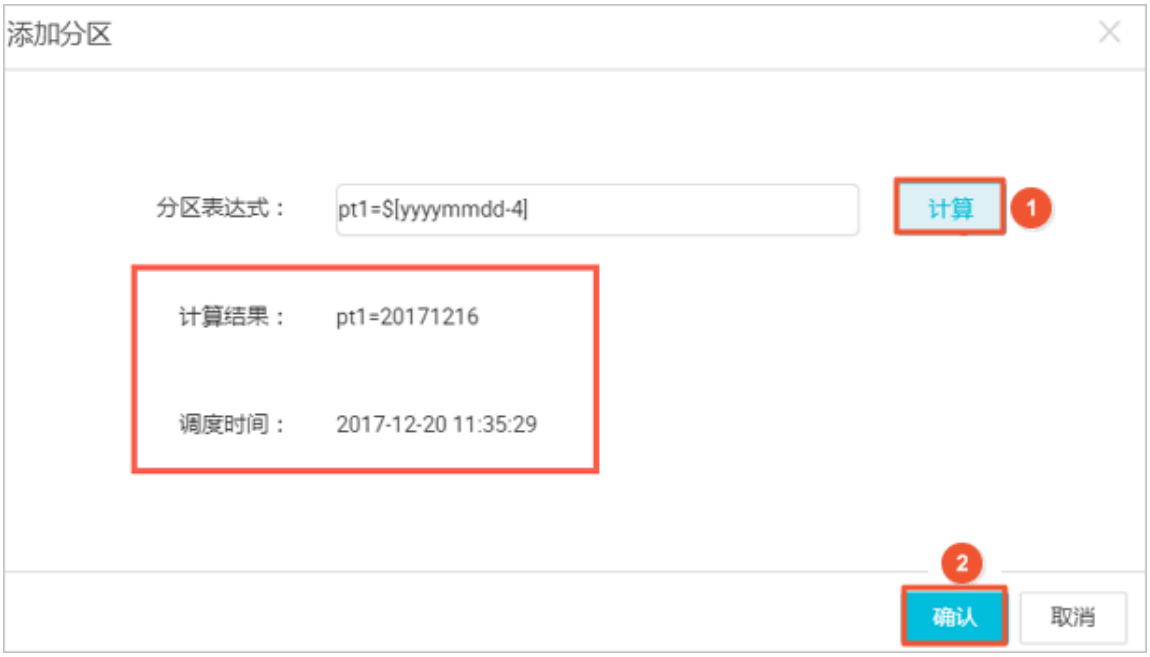
  分区表达式周期由配置的业务日期决定，例如配置运行时间为前5天，则周期为每5天调度一次。支持的分区表达式如下表所示。

分区表达式	说明
dt=$[yyyymmdd-N]	代表前N天。
dt=$[yyyymm01-1]	代表每月1日。
dt=$[yyyymm01-Nm]	代表N月前1日。
dt=$[yyyymmld-1]	代表每月最后一天。
dt=$[yyyymmld-1m]	代表N月前最后一天。
dt=$[hh24miss-1/24]	代表一个小时前。
dt=$[hh24miss-30/24/60]	代表半个小时前。
$[yyyymmdd]	调度日期。

分区表达式	说明
$[yyyymmdd-1]	格式为`yyyymmddmiss-1`，默认为当前实例运行的业务日期的前一天。
$[yyyymmddhh24miss]	格式为`yyyymmddhh24miss`，当前实例运行的业务日期。 - yyyy表示4位数年份 - mm表示2位数月份 - dd表示2位数天 - hh24表示24小时制的时 - mi表示2位数分钟 - ss表示2位数秒
NOTAPARTITIONTABLE	非分区表可以选择此分区表达式。

- 推荐的分区表达式：下文将以分区名dt为例，为您介绍推荐的分区表达式。动态分区表建议使用含有正则的分区表达式。

  1. 单击输入表达式的窗口，会显示数据质量为您推荐的分区表达式。
     - 如果有符合预期的表达式，单击该行，则会自动同步到输出窗口。
     - 如果没有满足需求的分区表达式，您可以根据需求自己输入。
  2. 输入分区表达式后，单击计算。数据质量会按照当前时间（调度时间）计算出分区表达式的计算结果，以便验证分区表达式的正确性。



  3. 单击确认。

- 删除已添加分区表达式：不需要的分区表达式可以删除。如果该分区表达式已经配置有规则，删除时会删除该表达式下的所有规则。

关联调度

如果要在生产链路上监控离线数据质量，需要将数据质量关联调度。

说明：

- 关联界面仅能找到已经提交的节点，且关联调度支持跨项目的关联。
- 关联前，请确保您在关联的两个项目中，同时拥有管理员、开发或运维中至少一个角色。

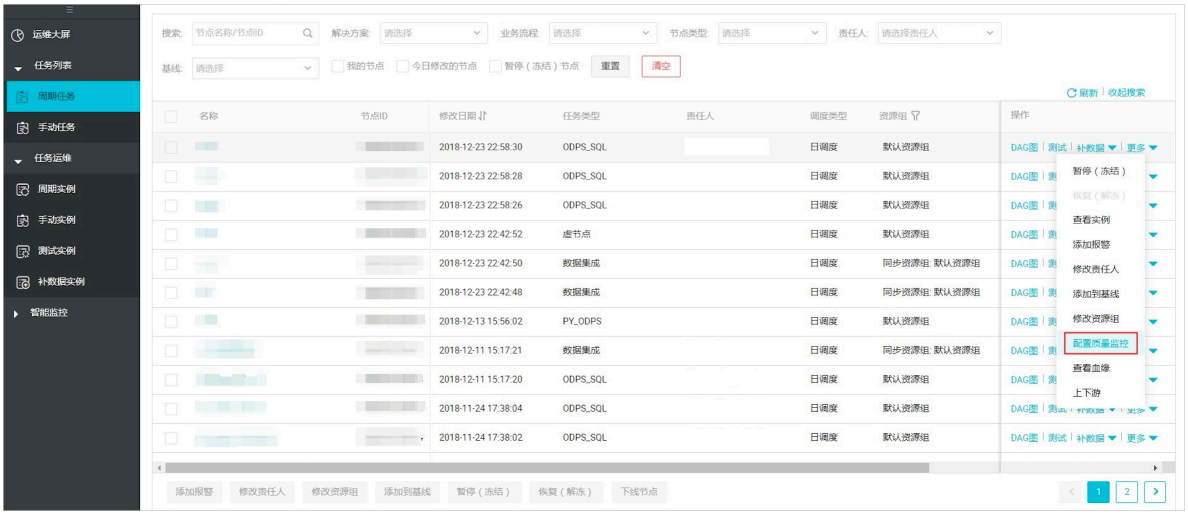
数据质量的关联调度可以关联单个或多个节点任务，关联调度完成后，离线数据质量监控任务可以自动运行。

说明：

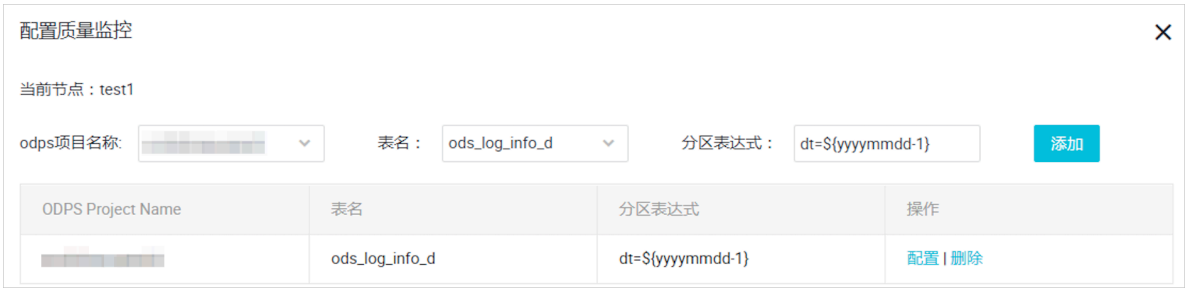
数据质量的关联可以灵活配置，您关联的任务并非一定要与您的表有关系。

关联配置步骤如下：

1. 进入运维中心 > 周期任务页面。
2. 单击对应任务后的更多，选择配置质量监控。



3. 输入对应项目名称、生产环境的表名进行搜索。完成搜索后，单击相应分区表达式后的配置（您也可自行添加分区表达式）。



创建规则

创建规则是数据质量模块的核心内容，您可以根据表的实际需要创建规则。

目前创建规则的方式包括模板规则和自定义规则，您可以根据自身需求选择相应方式。两种规则又分为添加监控规则和快捷添加两部分，详情请参见规则配置。

创建完成后单击批量保存，即可将创建的所有规则保存到已建好的分区表达式。

添加方式	配置	说明
添加监控规则	规则名称	输入规则名称。
	规则字段	包括表级规则和字段级规则。字段级规则可以针对表中的具体字段配置监控规则。此处选择为表级规则，页面中其他设置项对应为表级规则配置项。
	规则模板	系统内置的表级监控规则模块。
	比较方式	比较方式包括绝对值、上升和下降三种类型。

添加方式	配置	说明
	强弱	配置规则的强弱。当勾选强，任务运行时若触发红色阈值，则会将任务置为失败状态。 · 勾选强时，如果触发红色阈值，则报警且任务置为失败状态。如果触发橙色阈值，则报警且任务置为成功状态。 · 勾选弱时，如果触发红色阈值，则报警且任务置为成功状态。如果触发橙色阈值，则不报警且任务置为成功状态。
	波动值比较	设置波动值的橙色阈值和红色阈值。您可以通过拖动进度条来设置，也可以直接输入阈值。
	描述	对配置的规则进行简单描述。
快捷添加	规则名称	输入规则名称。
	监控字段	包括表级规则和字段级规则。字段级规则可以针对表中具体字段进行配置监控规则。
	快捷规则	· 选择表级规则，快捷规则仅支持表行数大于0。 · 选择字段级规则，快捷规则可以选择字段重复值或字段空值。

试跑

成功配置规则后，可针对某个分区表达式下所有规则进行试跑，并查看试跑的校验结果。

 说明:

通过试跑，可以测试规则配置的正确性、测试订阅发送渠道，它是手动运行监控规则的一种方式，您可以根据自身需求选择是否进行试跑。

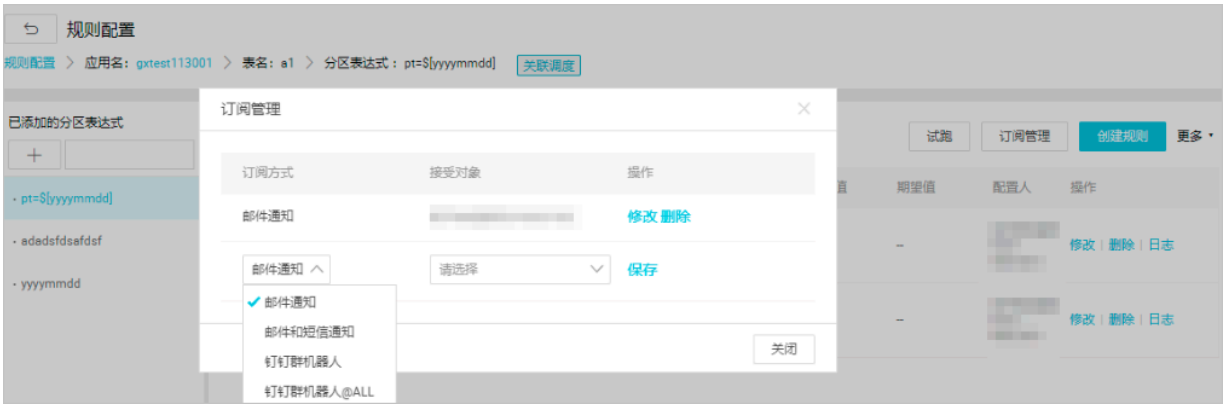
1. 选择需要试跑的调度日期，单击试跑，即可进行试跑。

配置	说明
试跑分区	实际分区会随着业务日期变化而改变。如果为NOPARTITIONTABLE，则会自动添加实际分区。
调度时间	默认为当前时间。

2. 单击试跑成功！点击查看试跑结果，即可跳转至任务查询页面，查看校验结果。

订阅管理

订阅管理默认通知创建者，如果想通知其他用户，您可以手动添加，支持邮件通知、邮件和短信通知、钉钉群机器人和钉钉群机器人@ALL。



转交责任人

当责任人离职或者转岗，可以将分区表达式负责人转交给其他项目成员。默认分区表达式负责人为创建人。

当悬停在责任人上时，会在后面显示一个隐藏按钮，单击可修改责任人，输入交接人的名称，单击确认即可提交成功。

更多

更多选项中包括分区操作日志、上一次校验结果和复制规则。

操作	说明
分区操作日志	显示对当前分区表达式所有的规则设置的记录。
上一次校验结果	跳转到任务查询页面，查看当前分区表达式下的运行结果情况，您还可在此查看历史结果。
复制规则	可将当前设置的规则复制到目标表达式中，还可同步订阅人。

7 可视化搭建-组件接口数据格式

7.1 接口数据格式-数据矩阵模板

```
{
  "code": "200",
  "data": [
    [
      [
        21.22, 2, 3, 40, 22, 23, 44
      ],
      [
        11.22, 2, 30, 11, 22, 13, 44
      ],
      [
        41.22, 2, 3, 11, 12, 33, 14
      ],
      [
        1.22, 12, 3, 11, 22, 13, 64
      ]
    ],
    [
      [
        21.22, 2, 3, 40, 22, 23, 44
      ],
      [
        11.22, 2, 30, 11, 22, 13, 44
      ],
      [
        41.22, 2, 3, 11, 12, 33, 14
      ],
      [
        1.22, 12, 3, 11, 22, 13, 64
      ]
    ],
    [
      [
        21.22, 2, 3, 40, 22, 23, 44
      ],
      [
        11.22, 2, 30, 11, 22, 13, 44
      ],
      [
        41.22, 2, 3, 11, 12, 33, 14
      ],
      [
        1.22, 12, 3, 11, 22, 13, 64
      ]
    ],
    [
      [
        21.22, 2, 3, 40, 22, 23, 44
      ],
      [
        11.22, 2, 30, 11, 22, 13, 44
      ],
      [
        41.22, 2, 3, 11, 12, 33, 14
```

```
      ],
      [
        1.22, 12, 3, 11, 22, 13, 64
      ]
    ]
  ]
}
```

说明:

真实比赛数据应为61*61*7的三维数组。此为符合矩阵模板组件的简单数据格式示例。

7.2 接口数据格式-表格组件

```
{
  code: "200",
  data: {
    items: [
      {
        id: 100,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 3,
        productMethod: "jdjd",
        productArea: "gfagfa"
      },
      {
        id: 101,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 2,
        productMethod: "jdjd",
        productArea: "gfagfa"
      },
      {
        id: 102,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 2,
        productMethod: "jdjd",
        productArea: "gfagfa"
      },
      {
        id: 103,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 2,
        productMethod: "jdjd",
        productArea: "gfagfa"
      },
      {
```

```
        id: 104,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 1,
        productMethod: "jdjd",
        productArea: "gfagfa"
      },
      {
        id: 105,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 1,
        productMethod: "jdjd",
        productArea: "gfagfa"
      },
      {
        id: 106,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 3,
        productMethod: "jdjd",
        productArea: "gfagfa"
      },
      {
        id: 107,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 2,
        productMethod: "jdjd",
        productArea: "gfagfa"
      },
      {
        id: 108,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 1,
        productMethod: "jdjd",
        productArea: "gfagfa"
      },
      {
        id: 109,
        image: "http://ifishonline.org/ifishimage/aquatic/thumbnail/
1703730305_THUMBNAIL.png",
        name: "ajkoajkoajkoajkoajkoajkoajkoajkoajko",
        enName: 50,
        grade: 1,
        productMethod: "jdjd",
        productArea: "gfagfa"
      }
    ],
    totalCount: 100
  }
```

```
}
```

说明:

上面是大赛需要的数据格式。data是对象，里面包含items对象数组，和totalCount字段。items中每个对象包含表格各列的字段，及对应数据；totalCount是数据总数，用于分页，默认每页10条数据。

7.3 接口数据格式-数据图表模板

```
{
  "code": "200",
  "data": [
    [
      92,
      32,
      8,
      85,
      34,
      1,
      53
    ],
    [
      33,
      72,
      80,
      76,
      76,
      78,
      13
    ],
    [
      63,
      98,
      89,
      14,
      37,
      31,
      17
    ],
    [
      21,
      38,
      64,
      63,
      68,
      65,
      20
    ],
    [
      41,
      36,
      20,
      46,
      68,
      29,
      32
    ],
```

```
    [
      68,
      78,
      3,
      51,
      26,
      81,
      0
    ],
    [
      45,
      13,
      14,
      16,
      32,
      30,
      0
    ],
    [
      37,
      36,
      95,
      49,
      40,
      7,
      51
    ],
    [
      70,
      48,
      96,
      64,
      17,
      81,
      86
    ],
    [
      47,
      19,
      71,
      9,
      75,
      54,
      86
    ]
  ]
}
```

说明:

真实比赛数据为3721*7的二维数组。此为适配模板的数据示例。

7.4 接口数据格式-数据地图模板

```
{
  "data": [
    {
      "name": "8月1日",
      "data": [
```

```
        [6, 30, 20, 50, 52, 43, 7, 21, 46, 40, 95, 67, 53, 1, 92, 64,
36, 66, 77, 39, 63, 93, 97, 18, 65, 24, 24, 90, 31, 19, 7, 40, 54, 67
, 70, 24, 45, 95, 53, 18, 47, 69, 77, 89, 11, 18, 71, 18, 69, 65, 81,
61, 70, 62, 54, 85, 93, 62, 47, 40, 33],
        // ...
      ]
    },
    {
      "name": "8月2日",
      "data": [
        [28, 44, 61, 88, 68, 21, 0, 19, 62, 5, 95, 81, 32, 13, 6, 39,
87, 37, 39, 46, 53, 0, 1, 6, 44, 74, 38, 13, 70, 74, 33, 81, 49, 57,
29, 33, 64, 59, 54, 76, 13, 31, 56, 14, 71, 24, 90, 28, 77, 87, 21, 34
, 87, 34, 6, 34, 22, 95, 86, 38, 0],
        // ...
      ]
    },
    {
      "name": "8月3日",
      "data": [
        [6, 30, 20, 50, 52, 43, 7, 21, 46, 40, 95, 67, 53, 1, 92, 64,
36, 66, 77, 39, 63, 93, 97, 18, 65, 24, 24, 90, 31, 19, 7, 40, 54, 67
, 70, 24, 45, 95, 53, 18, 47, 69, 77, 89, 11, 18, 71, 18, 69, 65, 81,
61, 70, 62, 54, 85, 93, 62, 47, 40, 33],
        // ...
      ]
    },
    {
      "name": "8月4日",
      "data": [
        [28, 44, 61, 88, 68, 21, 0, 19, 62, 5, 95, 81, 32, 13, 6, 39,
87, 37, 39, 46, 53, 0, 1, 6, 44, 74, 38, 13, 70, 74, 33, 81, 49, 57,
29, 33, 64, 59, 54, 76, 13, 31, 56, 14, 71, 24, 90, 28, 77, 87, 21, 34
, 87, 34, 6, 34, 22, 95, 86, 38, 0],
        // ...
      ]
    },
    {
      "name": "8月5日",
      "data": [
        [6, 30, 20, 50, 52, 43, 7, 21, 46, 40, 95, 67, 53, 1, 92, 64,
36, 66, 77, 39, 63, 93, 97, 18, 65, 24, 24, 90, 31, 19, 7, 40, 54, 67
, 70, 24, 45, 95, 53, 18, 47, 69, 77, 89, 11, 18, 71, 18, 69, 65, 81,
61, 70, 62, 54, 85, 93, 62, 47, 40, 33],
        // ...
      ]
    },
    {
      "name": "8月6日",
      "data": [
        [28, 44, 61, 88, 68, 21, 0, 19, 62, 5, 95, 81, 32, 13, 6, 39,
87, 37, 39, 46, 53, 0, 1, 6, 44, 74, 38, 13, 70, 74, 33, 81, 49, 57,
29, 33, 64, 59, 54, 76, 13, 31, 56, 14, 71, 24, 90, 28, 77, 87, 21, 34
, 87, 34, 6, 34, 22, 95, 86, 38, 0],
        // ...
      ]
    },
    {
      "name": "8月7日",
      "data": [
        [28, 44, 61, 88, 68, 21, 0, 19, 62, 5, 95, 81, 32, 13, 6, 39,
87, 37, 39, 46, 53, 0, 1, 6, 44, 74, 38, 13, 70, 74, 33, 81, 49, 57,
29, 33, 64, 59, 54, 76, 13, 31, 56, 14, 71, 24, 90, 28, 77, 87, 21, 34
, 87, 34, 6, 34, 22, 95, 86, 38, 0],
```

```
        // ...
      ]
    }
  ]
}
```

说明:

地图数据格式为对象数组。一共7个对象，每个对象中name字段为日期，data为61*61的二维数组。

7.5 接口数据格式-柱状图折线图饼图雷达图

```
{
  "code": "200",
  "data": [
    {
      "name": "降雨量",
      "data": [
        {
          "name": "day1",
          "value": 30
        },
        {
          "name": "day2",
          "value": 20
        },
        {
          "name": "day3",
          "value": 15
        },
        {
          "name": "day4",
          "value": 40
        },
        {
          "name": "day5",
          "value": 31
        },
        {
          "name": "day6",
          "value": 26
        },
        {
          "name": "day7",
          "value": 17
        }
      ]
    }
  ],
}
```

说明:

最外层数据为数组，里面包含1个对象。对象中包含name字段，和data数组。name表示图的系列名，data表示具体7天降雨量数据。

8 数据管理

8.1 数据管理概述

您可在数据管理模块进行组织内全局数据视图的查看、分权管理、元数据信息详情、数据生命周期管理、数据表/资源/函数权限管理审批等操作。

您可以通过点击左上方按钮切换到数据管理页面。

数据管理模块的具体功能，请参见下述文档。

数据搜索

#unique_500

新建表

收藏表修改生命周期

修改表结构

隐藏表

修改表负责人

删除表

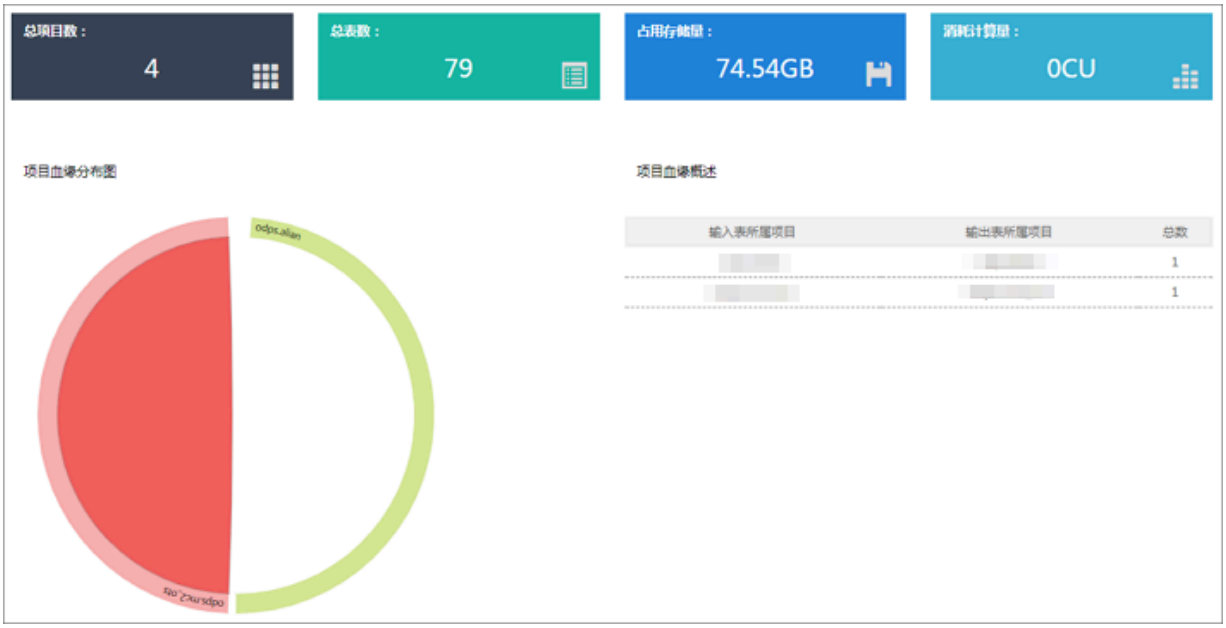
查看表详情

类目导航配置

8.2 全局概览

您可通过全局概览页面查看项目空间的整体情况。

选择数据管理 > 全局概览，进入全局概览页面，该页面中的统计都是在整个组织的前提下进行统计的，同时整个页面的数据信息都是离线产生，即页面中的数据信息为昨天的统计数据。



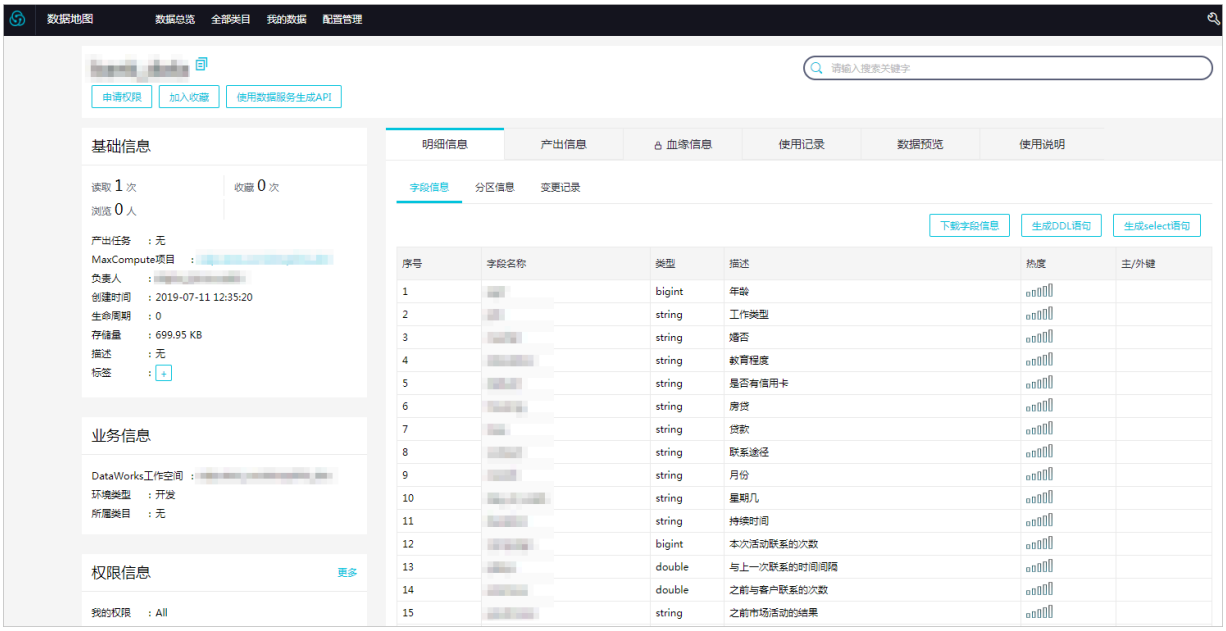
列表项	说明
总项目数、总表数、占用存储量、消耗计算量	在组织视角下，统计的项目空间个数、数据表个数、数据表占用的存储和任务运行时所消耗的计算量（CPU / minute或second等）。

列表项	说明
项目血缘分布图	在组织视角下，以网络来刻画项目空间之间的血缘关系，图中弧形代表项目空间，若存在血缘关系则两个项目空间之间进行连线。
项目血缘概述	在组织视角下，左侧为上游表所在项目空间，右侧为下游表所属项目空间，总数表示两个项目空间存在的血缘关系的数量。
项目占用存储Top	在组织视角下，各个项目空间所占用的存储排行榜，取前十名。
表占用存储Top	在组织视角下，展示数据表占用存储量前十名，并可单击具体表名跳转至表详情页。 说明: 项目存储及表占用的逻辑存储显示T+1用量，并且显示的为逻辑存储大小。项目存储量除了表存储量外，还会计算包括资源存储量、回收站存储量及其他系统文件存储量等在内，因此会大于表存储量。表的存储计费计算的是表的逻辑存储而非物理存储。
热门表	在组织视角下，被引用次数最多的数据表排行，显示前十名，并可单击具体表名跳转至表详情页。

8.3 表详情页介绍

您可以通过表详情页面，查看表的具体信息。

单击数据地图模块任意列表中的数据表名称，即可跳转至表详情页面。



表详情页面为您展示表的基础信息、业务信息、权限信息、技术信息、明细信息（包括字段信息、分区信息和变更记录）、产出信息、血缘信息、使用记录、数据预览和使用说明。

申请表权限

申请表权限的详情请参见申请表权限。

收藏表

单击表名称下方的加入收藏，即可收藏该表。表被收藏后，您可以进入数据地图 > 我的数据 > 我的收藏页面进行查看。

使用数据服务生产API

单击表名称下方的使用数据服务生产API，直接跳转至数据服务页面，详情请参见数据服务概览。

基础信息

表的基础信息包括表的读取次数、收藏次数、浏览人数、产出任务、MaxCompute项目、负责人、创建时间、生命周期、存储量、描述和标签等信息。

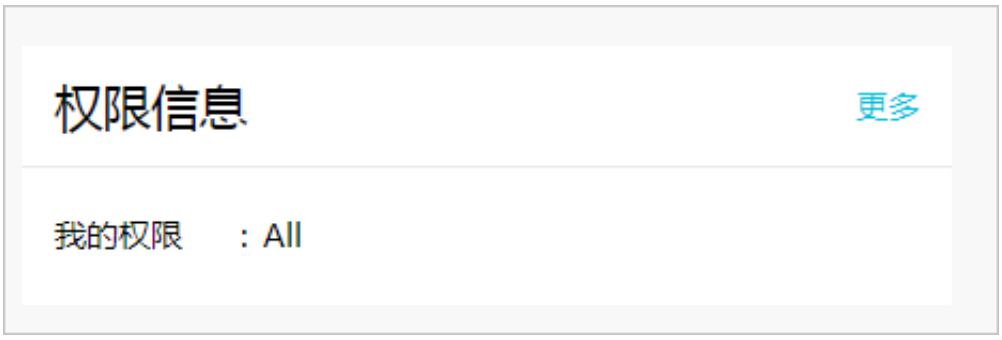
- 单击MaxCompute项目名称，即可跳转至项目详情页。
- 单击标签后面的 + ，即可为该表添加标签。

业务信息

表的业务信息主要包括DataWorks工作空间、环境类型和所属类目。

权限信息

表的权限信息为您展示您当前拥有的权限。单击右侧的更多，即可跳转至表权限申请页面。



技术信息

表的技术信息主要包括技术类型、DDL最后变更时间、最后数据变更时间、最后数据查看时间和计算引擎信息等信息。

- 时间格式默认为`yyyy-mm-dd hh:ss:mm`。
- 单击计算引擎信息后的点击查看按钮，即可查看计算引擎信息弹出框中的信息。

明细信息

表的明细信息是以表的元数据信息为主，显示字段定义、热度、等级、主外键及表结构变更情况。

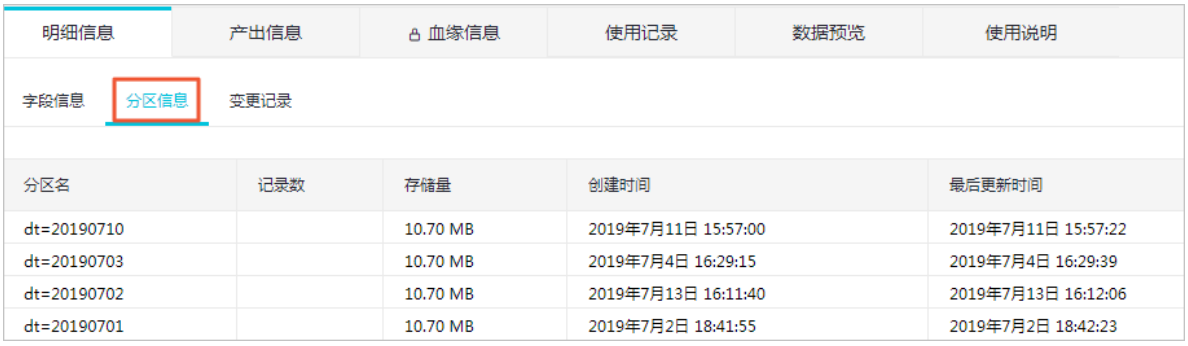
- 字段信息

  表的字段信息包括字段名称、类型、描述、热度和主/外键等信息。

操作	说明
下载字段信息	单击后，即可直接下载相应的字段信息。
生成DDL语句	单击后，即可在弹出框中显示相应的建表语句。
生成select语句	单击后，即可在弹出框中显示相应的select语句。

- 分区信息

  通过表的分区信息模块，您可以查看表当前的分区，包括分区名、记录数、存储量、创建时间和最后更新时间等信息。



- 变更记录

  表的变更记录包括分区名、变更类型、粒度、时间和操作人等信息。

  您可以在下拉框中选择不同的变更类型，来过滤、筛选变更记录。

产出信息

如果表的数据会随着对应的任务周期性发生变化，您可以在该模块查看变化的情况、持续更新的数据等信息。

任务名称	操作	节点ID	执行时间(秒)	等待时间(秒)	开始时间	结束时间
[redacted]	代码 日志	700000624	8	6	2019-04-17 00:06:07	2019-04-17 00:06:15
[redacted]	代码 日志	700000624	103	3	2019-04-17 00:29:05	2019-04-17 00:30:48
[redacted]	代码 日志	700000624	10	25	2019-04-18 00:06:27	2019-04-18 00:06:37
[redacted]	代码 日志	700000624	100	8	2019-04-18 00:29:10	2019-04-18 00:30:50
[redacted]	代码 日志	700000624	11	5	2019-04-19 00:06:06	2019-04-19 00:06:17
[redacted]	代码 日志	700000624	100	3	2019-04-19 00:29:05	2019-04-19 00:30:45
[redacted]	代码 日志	700000624	10	57	2019-04-20 00:07:26	2019-04-20 00:07:36
[redacted]	代码 日志	700000624	100	7	2019-04-20 00:29:09	2019-04-20 00:30:49
[redacted]	代码 日志	700000624	12	4	2019-04-21 00:06:05	2019-04-21 00:06:17
[redacted]	代码 日志	700000624	97	6	2019-04-21 00:29:08	2019-04-21 00:30:45

血缘信息

表的血缘信息可以为您清晰地展示数据的来源和去向，并提供便捷的交互。

说明:

仅购买DataWorks标准版及以上版本的用户，可以查看表的血缘信息。

- **表血缘：您可以根据GUID搜索表的上下游表。**

- **字段血缘：您可以根据字段进行过滤。**

使用记录

- **频繁关联：频繁关联模块为您展示有多少人在使用当前的数据。**

- **访问统计：访问统计模块，以线型方式为您展示表的使用记录。**

数据预览

您可以通过表的数据预览模块，预览当前表的数据信息。

	A	B	C	D	E	F	G	H
字段	empno	firstnme	midinit	lastname	workdept	phoneno	hiredate	job
类型	STRING	STRING	STRING	STRING	STRING	STRING	DATETIME	STRING
3	000010	CH E	I	HA	A00	3978	1995-01-01 00:00:00	PRES
4	000020	MI	L	TH ON	B01	3476	2003-10-10 00:00:00	MANAGER
5	000030	SA	A	KV	C01	4738	2005-04-05 00:00:00	MANAGER
6	000050	JO	B	GE	E01	6789	1979-08-17 00:00:00	MANAGER
7	000060	IR	F	ST	D11	6423	2003-09-14 00:00:00	MANAGER
8	000070	EV	D	PU	D21	7831	2005-09-30 00:00:00	MANAGER
9	000090	EI	W	HE SON	E11	5498	2000-08-15 00:00:00	MANAGER
10	000100	TH E	Q	SP	E21	0972	2000-06-19 00:00:00	MANAGER
11	000110	VI	G	LU SSI	A00	3490	1988-05-16 01:00:00	SALESREP
12	000120	SE		O' ELL	A00	2167	1993-12-05 00:00:00	CLERK
13	000130	DE	M	QU NA	C01	4578	2001-07-28 00:00:00	ANALYST
14	000140	HE	A	NI S	C01	1793	2006-12-15 00:00:00	ANALYST
15	000150	BR		AD ON	D11	4510	2002-02-12 00:00:00	DESIGNER
16	000160	ELIZABETH	R	PI	D11	3782	2006-10-11 00:00:00	DESIGNER

说明:

您需要拥有权限，方可预览生产环境的表。如果没有权限，单击去申请，前往申请页面进行申请。

使用说明

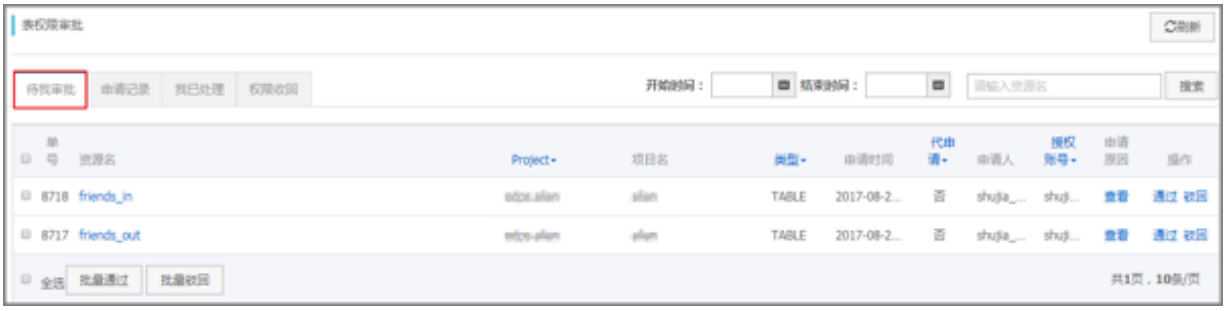
您可以编辑此界面，查看相关的历史版本和Markdown语法。您也可以根据数据的业务说明了解相关的信息。

8.4 权限管理

权限管理模块主要对待我审批、申请记录、我已处理、权限回收等相关表、资源、函数的权限申请进行管理。

待我审批

当前访问账号为管理员时，可通过待我审批模块，查看并审批其所有项目下的待审批记录，包括表、资源和函数等权限申请。



申请记录

当前访问账号可通过申请记录模块，查看其权限申请记录。

我已处理

当前账号为管理员时，可通过我已处理模块，查看其所有项目下已经处理过的权限申请记录，包括表、资源和函数等权限申请。

权限回收

当前账号为管理员时，可通过权限回收模块，查看并回收其所有项目下的已经通过的权限申请记录，包括表、资源和函数等权限申请。

8.5 数据权限申请

本文将为您介绍如何申请数据权限。

DataWorks中有以下三种数据类型。

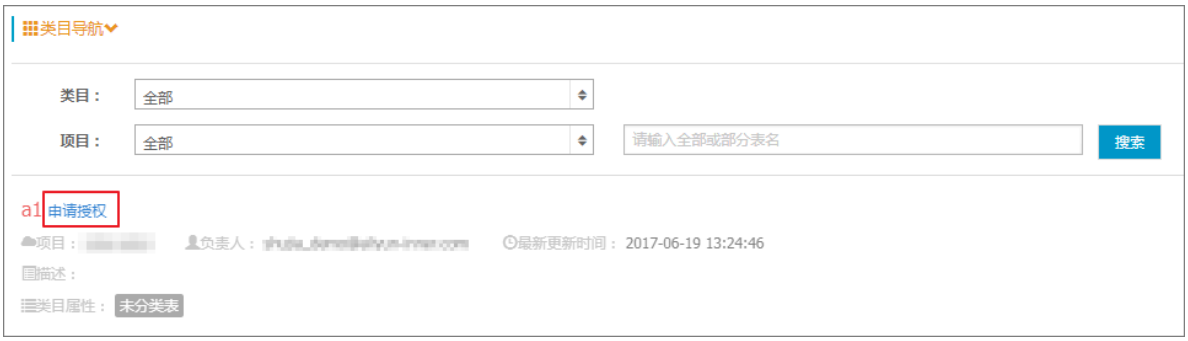
- 表：即数据表。
- 函数：即UDF，可在SQL中使用的函数。

· 资源：如文本文件、MapReduce的Jar文件等。

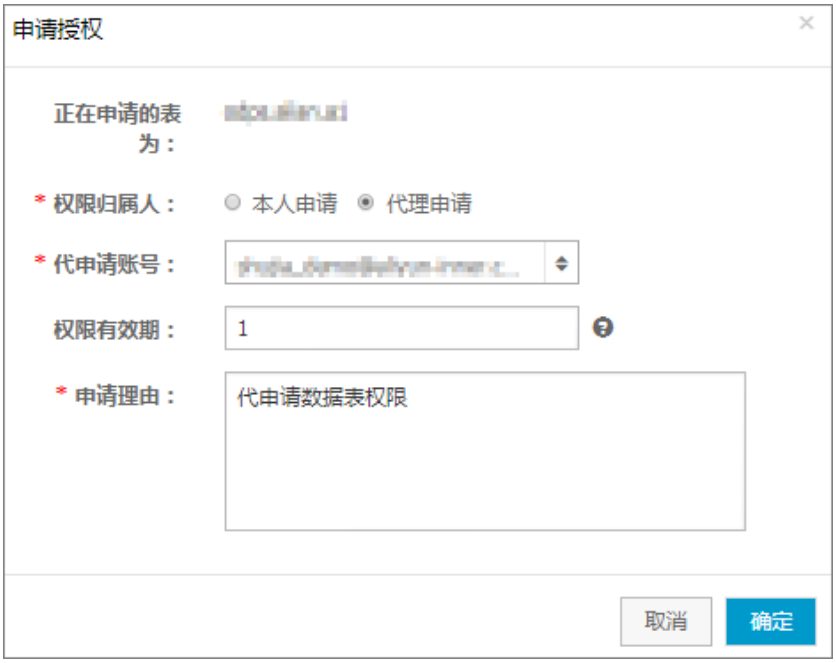
这三种数据类型有着严格的权限控制机制，您需要申请权限后才能使用。

表权限申请（只读权限）

1. 通过数据管理 > 查找数据页面找到需要申请权限的数据表。
2. 单击数据表操作栏中的申请权限。



3. 填写申请授权弹出框中的各配置项。

配置	说明
权限归属人	支持本人申请和代理申请。 · 本人申请：选择该项，审批通过后权限归属于当前用户。 · 代理申请：选择代理申请，需填写代申请账号（系统右上角显示的登录名）。审批通过后权限归属于被代理人。 
权限有效期	申请表权限的时长，单位为天，不填则默认为永久。超过申请权限时长时，该权限将被系统自动回收。

配置	说明
申请理由	请简要填写申请理由以便更快地通过审批。

4. 单击确定后提交，然后等待审批。您可在权限管理 > 申请记录中查看申请状态。

函数和资源权限申请

1. 进入数据管理 > 查找数据页面。
2. 单击列表右上方的申请数据权限。

3. 填写申请授权对话框中的配置。

配置	说明
申请类型	支持函数和Resource资源两种类型。

配置	说明
权限归属人	支持本人申请和代理申请。 · 本人申请：选择该项，审批通过后权限归属于当期用户。 · 代理申请：选择代理申请，需填写代申请账号。审批通过后权限归属于被代理人。
项目名	选择所需申请函数或者资源所在的项目名称（对应的MaxCompute项目名称），支持本组织范围内模糊匹配查找。
函数名称或资源名称	输入项目中的函数或资源名称。资源名称请填写完整，包括文件后缀，如 my_mr.jar。
权限有效期	申请表权限的时长，单位为天，不填则默认为永久。超过申请权限时长时，该权限将被系统自动回收。
申请理由	请简要填写申请理由以便更快地通过审批。

4. 单击确定后提交，然后等待审批。您可通过权限管理 > 申请记录查看申请状态。

8.6 管理配置

当前用户（需组织管理员权限）可在类目导航配置页面中进行新建表所属类目的配置。

操作步骤

1. 以开发者身份进入DataWorks 管理控制台，单击项目列表下对应项目后的进入数据开发。

2. 单击顶部菜单栏中的数据管理。

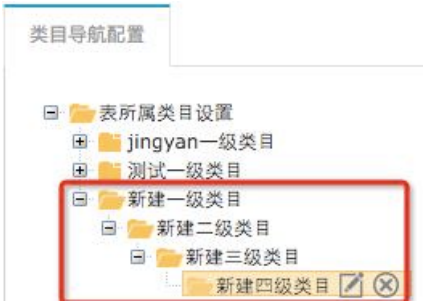
3. 在数据管理页面，单击左侧导航栏中的管理配置。

4. 单击表所属类目设置后的⊕，添加一级类目。

5. 单击一级类目后的⊕，添加所属二级类目。

以此类推，最高可支持四级类目的设置。其中表示编辑该类目名称，表示删除该类目。

类目配置完成后，即可在新建表页面中选择已配置类目。

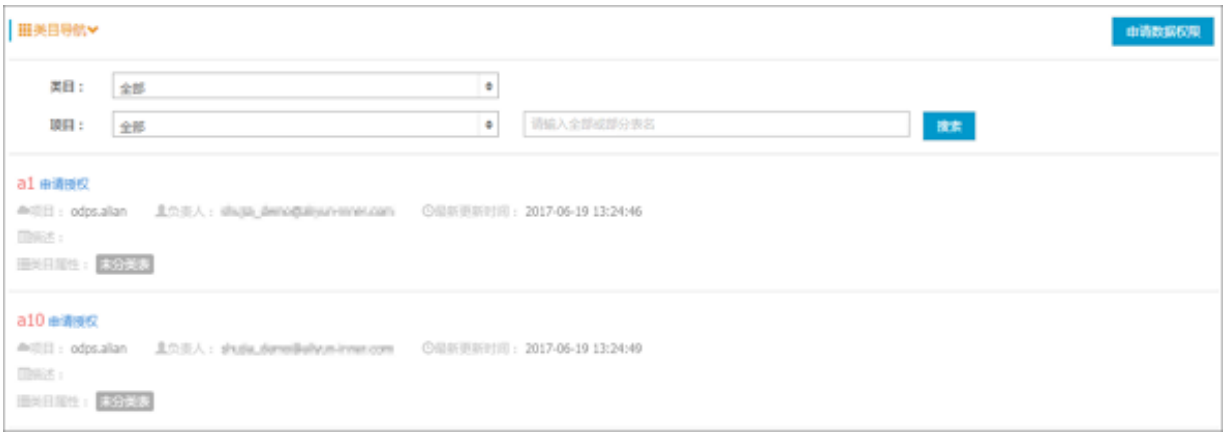


新建表的所属类目，如下图所示：

8.7 查找数据

本文将为您介绍如何查找数据。

您可进入数据管理 > 查找数据 页面，对本组织范围内（多项目空间）的数据表进行搜索。在全部数据页面中，通过选择数据类目导航 + 搜索框中输入表名进行模糊匹配的方式快速查找需要的表。



您可通过以下三种方式对数据表进行搜索。

- 类目导航选择：查看当前类目下所有表。
- 项目名称选择：查看当前项目下所有表，可与类目筛选条件一起使用。
- 搜索条件：在搜索框里输入表名（支持表名模糊搜索），同时支持备注搜索。

8.8 数据表管理

数据表管理模块对数据表进行分类，并为各分类提供不同的表信息以及表操作管理功能，以便您管理自己的数据表。

在数据表管理中，您可对表进行生命周期设置、表管理（包括修改表的类目、描述、字段、分区等）、表隐藏/取消隐藏和表删除等操作。

表的分类

- 我收藏的表

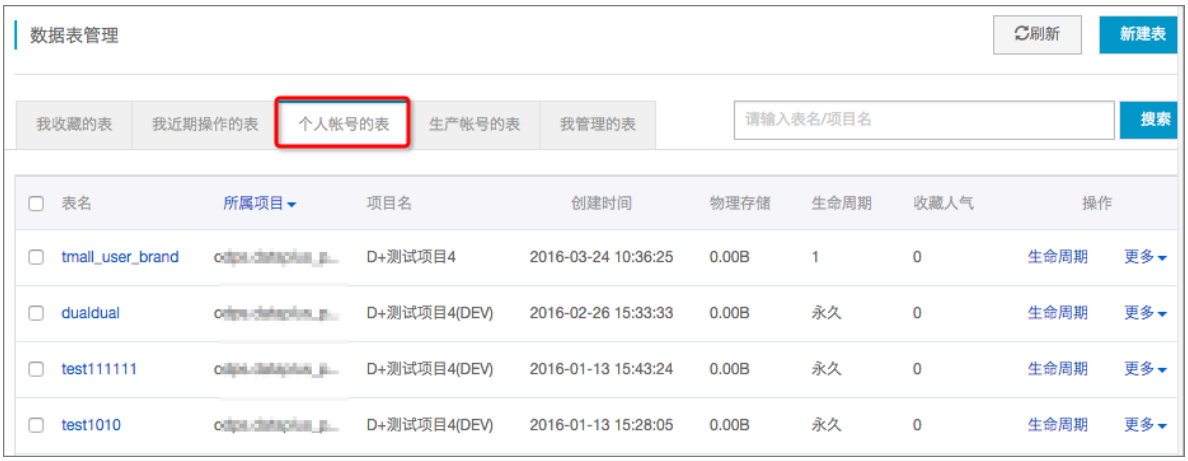
  展示了您所收藏的数据表列表，在此您也可以进行取消收藏操作。

- 我近期操作的表

  展示了您近期操作过的表，在此您可以进行表生命周期设置、表管理（包括修改表的类目、描述、字段、分区等）、表隐藏/取消隐藏、表删除等操作。

- 个人账号的表

  展示了您在组织内所创建的数据表列表，即Owner为当前用户的表。



本模块支持通过表名称模糊匹配查找，同时也可通过所属项目筛选查找表。您可在此对表进行的操作同我近期操作的表。

- 生产账号的表

  展示了表Owner为MaxCompute访问身份配置为计算引擎指定账号（即生产账号）的表。您可在此对表进行的操作同我近期操作的表。

- 我管理的表

  若您为项目管理员，将在此页面显示其所管理的项目空间中的所有数据表，管理员可在此可对表进行各类操作，包括修改表Owner。

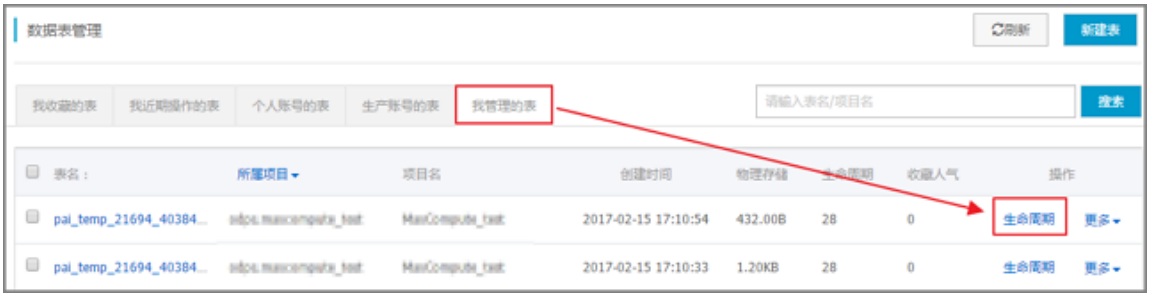
表的管理操作

- 收藏表

  数据管理模块提供收藏表功能，您在表详情页中单击收藏即可，也可在我收藏的表页面中取消收藏。

- 修改生命周期

  1. 单击列表的操作栏中的生命周期。

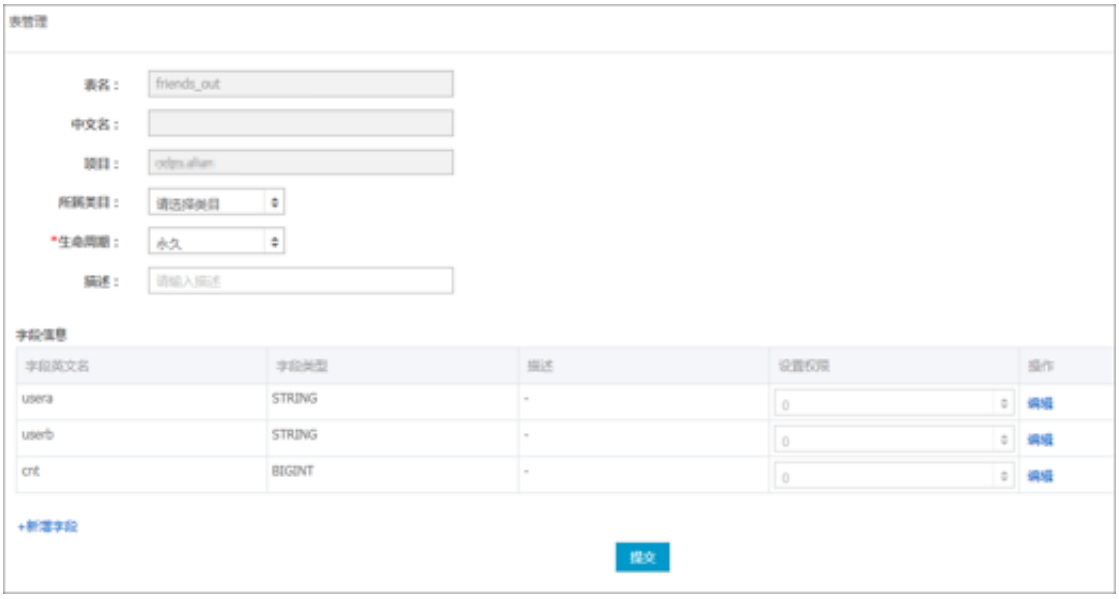


  2. 修改生命周期弹出框中表的生命周期。

- 修改表结构

1. 单击列表操作栏中的更多选项，选择表管理进行表结构修改。

2. 在打开的表管理页面中修改相关信息。



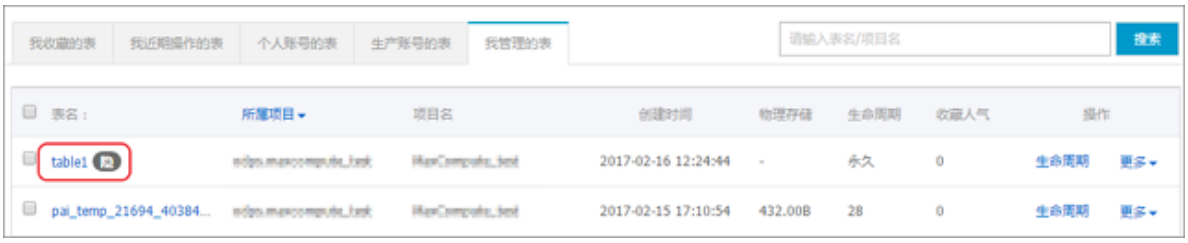
3. 修改完成后，单击提交。

- 隐藏表

  隐藏表功能即表负责人或项目管理员可对表进行隐藏，让其他成员不可见。

  单击列表操作栏中 更多选项，选择隐藏即可隐藏表。隐藏后的表，也可在此单击取消隐藏。

  被隐藏的表名后将有隐藏标识。



  说明:

  主账号隐藏的表子账号不能查看隐藏表内容，单击会报相应的提示：表被隐藏，请联系管理员或owner，子账号隐藏表主账号可以查询表内容。

- 修改表负责人（表Owner）

  项目管理员可修改表负责人，具体操作如下：

  1. 进入我管理的表模块，单击列表操作栏中的更多，选择修改owner。

  2. 在修改表owner弹出框中输入新Owner的云账号名称，该Owner必须为本项目的成员。
  3. 修改完成后，单击提交。

- 删除表

  1. 单击列表操作栏中的更多，选择删除。

  2. 单击确定，确认删除操作。

> 说明:
> 数据表一旦删除，该表的结构信息及表的所有数据均不可恢复，请谨慎操作。

8.9 创建表

通常数据开发过程中需要创建表来存储数据同步、数据加工的结果数据，本文将为您介绍可视化建表、DDL建表两种创建表的方式。

> 说明:
> 通过数据地图（原数据管理）模块创建的表可以进行业务类目划分。当组织内业务很多时，给数据表划分类目可以方便元数据的管理。

前提条件

- 实名认证云账号，生成AccessId和Accesskey。

  建表所用的云账号都是当前登录人账号，必须有AccessId和Accesskey才能发请求到MaxCompute进行建表，所以该云账号必须要实名认证生成AccessId和Accesskey。详情请参见#unique_30。

- 给云账号赋建表权限。

  如果您需要创建表，必须先给建表的云账号授权。MaxCompute项目的owner直接运行授权语句进行授权。

```
use projectname; --打开项目空间
add user aliyun$云账号;  --添加用户
grant CreateInstance,CreateTable,List ON PROJECT projectname TO
aliyun$云账号; --给用户赋权
```

> **说明:**
> 因为此处建表都是用当前登录的云账号创建，因此表的owner即为当前登录账号。

可视化建表

1. 以开发者身份登录DataWorks控制台，单击相应工作空间后的进入数据开发。
2. 单击左上角的图标，选择全部产品 > 数据地图（数据管理），即可进入数据地图页面。
3. 单击左侧菜单栏中的数据表管理，然后单击右上角的新建表。

4. 填写新建表页面的基础信息，单击下一步。

配置	说明
项目名	列表中显示当前用户已加入的MaxCompute项目空间。
表名	以字母、数字、下划线组成。
别名	表的中文名称。
所属类目	当前表所处的类目，最多支持四级类目导航，详情请参见#unique_504。
描述	当前表的简要说明。
生命周期	即MaxCompute的生命周期功能。填写一个数字表示天数，那么该表（或分区）超过一定天数未更新的数据会被清除。 支持1天、7天、32天、永久和自定义5种选项。

5. 填写新建表页面的字段和分区信息。

   - 添加字段信息设置。
   - 设置分区。

配置	说明
字段英文名	字段英文名，由字母、数字、下划线组成。
字段类型	MaxCompute数据类型（STRING、BIGINT、DOUBLE、DATETIME和BOOLEAN）。
描述	字段详细描述。
操作	上移、下移和删除。
是否设置分区	如果选择设置分区，需配置分区键的具体信息，支持STRING和BIGINT类型。

6. 单击提交。

   新建表提交成功后，系统将自动跳转至数据表管理页面，单击我管理的表，即可查看新建的表。

DDL建表

1. 以开发者身份登录DataWorks控制台，单击相应工作空间后的进入数据开发。
2. 单击左上角的图标，选择全部产品 > 数据地图（数据管理），即可进入数据地图页面。
3. 单击左侧菜单栏中的数据表管理，然后单击右上角的新建表。
4. 单击新建表对话框中的DDL建表。

5. 填写MaxCompute SQL建表语句。

```
create table if not exists table2
(
 id string comment'用户ID',
 name string comment'用户名称'
) partitioned by(dt string)
```

```
LIFECYCLE 7;
```

6. 单击提交，出现如下页面。

基础信息页面除表别名、所属类目和生命周期配置项外，都会自动填充。而字段和分区信息页面中字段的中文名称以及字段的安全等级等，需要您进行编辑添加。

7. 补充新建表基础信息页面中的配置项，单击下一步。

8. 单击提交。

新建表提交成功后，系统将自动跳转至数据表管理页面，单击我管理的表，即可查看新建的表。

9 数据地图

9.1 数据管理升级为数据地图

本文将为您及时同步数据管理升级到数据地图的进展和计划。

DataWorks数据地图模块第一批发布计划

- 发布版本：发布DataWorks数据地图模块，取代数据管理模块。
- 发布时间：
  - 2019年6月25日~6月28日20:00~20:20，发布数据地图至各区域。
  - 2019年7月1日20:00，下线所有区域的数据管理模块。
- 发布内容：DataWorks数据地图模块在DataWorks数据管理模块的基础上，根据角色区分对应的功能，控制数据预览权限、新建表权限等，帮助您更好地构建企业级数据信息知识库。
- 已发布：华东2（上海）、华北2（北京）、华东1（杭州）、华南1（深圳）、中国（香港）和亚太东南1（新加坡）。
- 待发布：
  - 2019年6月25日20:00~20:20：发布数据地图至亚太东南2（悉尼）、亚太东南3（吉隆坡）和亚太南部1（孟买）。
  - 2019年6月26日20:00~20:20：发布数据地图至亚太东南5（雅加达）、欧洲中部1（法兰克福）和英国（伦敦）。
  - 2019年6月27日20:00~20:20：发布数据地图至美国东部1（弗吉尼亚）、美国西部1（硅谷）和中东东部1（迪拜）。
  - 2019年7月1日20:00~20:20：下线所有区域的数据管理模块。

DataWorks数据地图与数据管理功能对比一览表

DataWorks数据地图相比于数据管理，在整体视觉交互和角色区分等方面提升使用体验，详情如下表所示。

对比项	数据管理	数据地图	改进效果
页面功能和角色关联	按功能类型组织，分为总览和分页面如下： · 全局概览 · 查找数据 · 数据表管理 · 权限管理 · 管理配置	· 按角色和功能的关系组织页面如下： - 数据总览 - 数据地图首页和全部类目：针对数据搜索。 - 我的数据： 针对个人的表管理、收藏与权限申请和审批。 - 配置管理：针对工作空间管理或类目管理。 · 专为使用实际数据的使用者强化搜索能力、提供按类目检索的能力。	· 用户群体中，人数占比最多的数据实际使用者，能够更方便快捷地查找自己想要的数据。 · 数据管理者，包括个人作为表负责人、工作空间管理者、类目管理者，根据角色选择页面，权责更加清晰，相关功能更易用。
个人操作更集中	表管理、收藏管理、权限管理、资源和函数的权限申请等分散在各处。	· 资源和函数的权限申请入口以及表、资源、函数的审批入口，全部归属到我的数据页面。 · 个人对表的操作入口、表收藏功能等，也移动到我的数据页面。	个人对数据的操作更加集中，方便查找和使用。
表的预览权限收拢	任意用户可以看到表的数据预览，表的负责人或工作空间管理员无法控制。	· 表的预览权限归属于表的负责人所有，其他用户需要申请表的权限才能查看数据。 · 工作空间管理员可以通过开关配置开发表、生产表的数据预览权限对其他用户开放或关闭。 - 开发表默认开放对其他用户的数据预览权限。 - 生产表默认关闭对其他用户的数据预览权限。	· 使表的负责人对于自己所拥有的表的权限控制粒度更细致。 · 工作空间管理员可以方便地控制空间内表的数据预览权限。
新建表的入口收拢	可以在数据管理页面直接新建表，没有权限控制。	· 新建表的入口暂时关闭。 · 建议您使用数据开发的表管理或手动任务来新建和修改表结构，可以复用数据开发模块的开发、运维角色控制。 · 如果需要类目关联，您可以在数据地图 > 我的数据页面快速关联自己所拥有的表所属的类目。	· 使表的新建和修改的入口统一在数据开发模块，并有角色权限控制。 · 控制表的结构变化归属于数据开发者，避免无权限人员随意创建表或关联类目。

数据地图问题反馈途径

如您对DataWorks数据地图模块存在任何产品使用方面的问题，请随时搜索钉钉群号（23182329）或通过扫码进入DataWorks产品交流群，我们将竭诚为您答疑。

9.2 数据地图概述

数据地图主要包括全局查找数据、个人账号管理数据和管理员配置。

单击左上角的图标，选择全部产品 > 数据地图，即可进入数据地图页面。


DataStudio
全部产品
数据集成
DataStudio(数据开发)
运维中心(工作流)
任务发布
数据地图(数据管理)
DataStudio(数据开发)
数据服务
Function Studio
New
数据分析
任务运维
运维中心(工作流)
任务发布
跨项目克隆
数据治理
数据质量
数据保护伞
数据地图(数据管理)


数据地图
数据总览
全部类目
我的数据
配置管理
请输入搜索关键字
表
biao
近期浏览的表
近期读取的表
热门浏览
热门读取

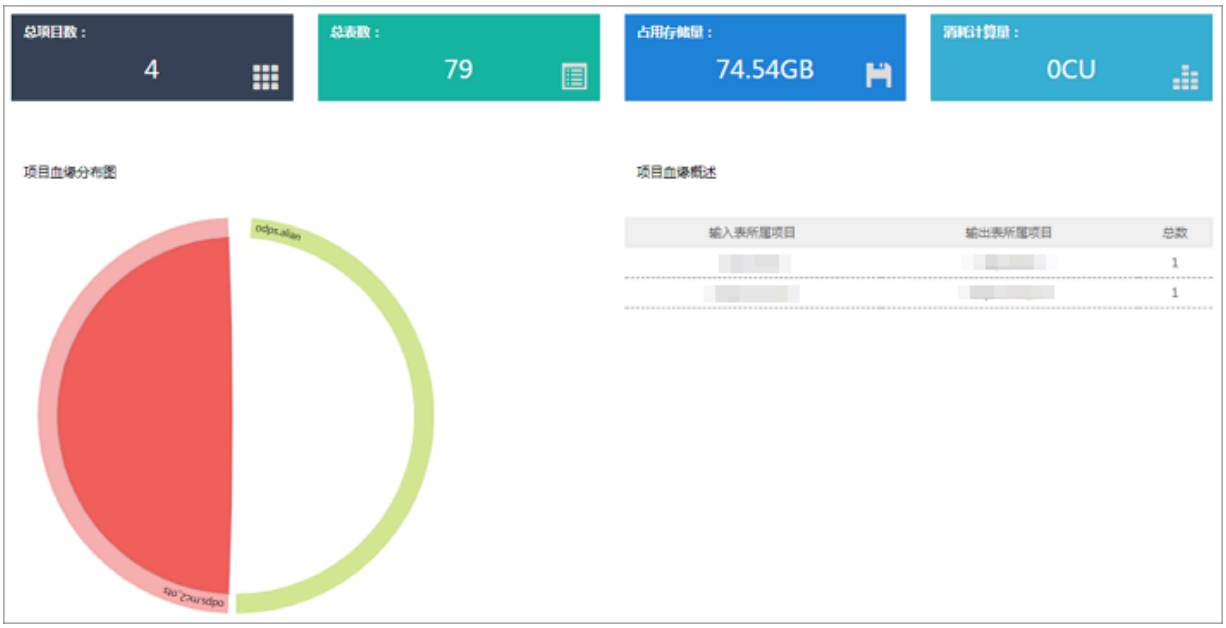
- 如果您更习惯使用强大的搜索引擎，请单击左上方的数据地图，进入首页开始搜索。

  目前输入关键字匹配更准确，同时还提供更多的搜索对象。例如，如果您经常使用数据地图，在数据地图的首页，为您提供了近期浏览和近期读取的表，以及基于您的访问记录推荐的热门浏览和热门读取。
- 如果您需要根据路径找到您想要的表，请单击全部类目。该页面将按照类目结构，为您展示各级类目及对应的表数量。
- 如果您有一些个人事务需要处理，例如修改您拥有的表或者使用小工具，请单击我的数据。
- 如果您是类目管理者或项目管理者，需要修改项目级配置或全局类目，请单击配置管理。

9.3 数据总览

您可以通过数据总览页面，查看工作空间的整体情况。

选择数据地图 > 数据总览，进入数据总览页面。该页面在整个组织的前提下进行统计，同时离线产生整个页面的数据信息，即页面中的数据信息为昨天的统计数据。



列表项	说明
总项目数	在组织视角下，所有工作空间的数量。
总表数	在组织视角下，所有表的数量。
占用存储量	在组织视角下，所有表存储量的总和。
消耗计算量	在组织视角下，该组织累计一天的计算消耗。一个CPU核满载运行一天为1CU。
项目血缘分布图	在组织视角下，以网络来刻画工作空间之间的血缘关系，图中弧形代表工作空间。如果存在血缘关系，则两个工作空间之间进行连线。
项目血缘概述	在组织视角下，左侧为上游表所在工作空间，右侧为下游表所属工作空间，总数表示两个工作空间存在的血缘关系的数量。
项目占用存储Top	在组织视角下，各个工作空间所占用的存储排行榜，取前十名。
表占用存储Top	在组织视角下，展示数据表占用存储量前十名，并可单击具体表名跳转至表详情页。 说明: 项目存储及表占用的逻辑存储显示T+1用量，并且显示的为逻辑存储大小。项目存储量除了表存储量外，还会计算包括资源存储量、回收站存储量及其他系统文件存储量等在内，因此会大于表存储量。表的存储计费计算的是表的逻辑存储而非物理存储。
热门表	在组织视角下，被引用次数最多的数据表排行，显示前十名。并可以单击具体表名跳转至表详情页。

9.4 我的数据

本文将为您介绍数据地图中我的数据模块的功能。

我拥有的数据

您可以在我拥有的数据页面，根据表名、环境、项目/数据库和可见范围等信息进行搜索，查看相应表的具体信息并进行相关操作。

配置	说明
表名	单击表名，即可跳转至表详情页面。
中文名	您可以编辑表的中文名。
项目名/数据库	如果您的表是在不同的环境，会有相关的后缀。例如，_dev表示开发环境。
环境类型	环境类型通常包括开发和生产。
存储量	显示您存储的数据量。
生命周期	和您创建表时设置的生命周期一致。
操作	您可以在相应表后的操作栏下，进行删除、修改类目、隐藏和自动探查等操作。 **说明：** 如果您隐藏了表，在表详情页将看不到申请权限的入口。

我管理的数据

您可以在我管理的数据页面，根据表名、环境和项目/数据库等信息进行搜索，查看相应表的具体信息并进行相关操作。

配置	说明
表名	单击表名，即可跳转至表详情页面。
中文名	您可以编辑表的中文名。

配置	说明
项目名/数据库	如果您的表是在不同的环境，会有相关的后缀。例如，_dev表示开发环境。
环境类型	环境类型通常包括开发和生产。
存储量	显示您存储的数据量。
生命周期	和您创建表时设置的生命周期一致。
操作	您可以在相应表后的操作栏下，进行删除和修改类目等操作。

生产账号的数据

您可以在我管理的数据页面，根据表名、环境和项目/数据库等信息进行搜索，查看相应表的具体信息。

配置	说明
表名	单击表名，即可跳转至表详情页面。
中文名	您可以编辑表的中文名。
项目名/数据库	如果您的表是在不同的环境，会有相关的后缀。例如，_dev表示开发环境。
环境类型	环境类型通常包括开发和生产。
存储量	显示您存储的数据量。
生命周期	和您创建表时设置的生命周期一致。
收藏人次	用户收藏该表的次数。
30天浏览人次	用户30天内浏览该表的次数。
创建时间	该表的创建时间。

我的收藏

收藏表后，你可以在我的收藏页面进行查看。

您可以单击相应表后的取消收藏，取消对该表的收藏。

权限管理

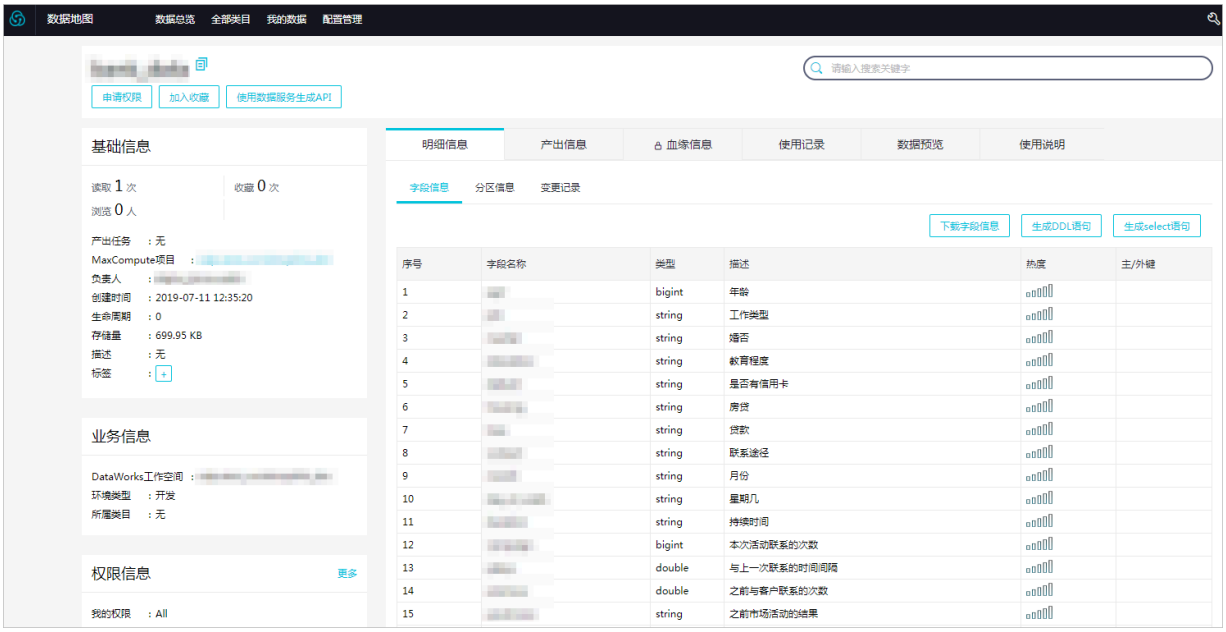
选择数据地图 > 权限管理，即可进入权限管理功能模块。

- 权限管理模块的功能详情请参见权限管理。
- 单击右上方的申请数据权限按钮，可以申请函数和资源权限，详情参见#unique_507/unique_507_Connect_42_section_cjr_fz5_q2b。

9.5 表详情页介绍

您可以通过表详情页面，查看表的具体信息。

单击数据地图模块任意列表中的数据表名称，即可跳转至表详情页面。



表详情页面为您展示表的基础信息、业务信息、权限信息、技术信息、明细信息（包括字段信息、分区信息和变更记录）、产出信息、血缘信息、使用记录、数据预览和使用说明。

申请表权限

申请表权限的详情请参见申请表权限。

收藏表

单击表名称下方的加入收藏，即可收藏该表。表被收藏后，您可以进入数据地图 > 我的数据 > 我的收藏页面进行查看。

使用数据服务生产API

单击表名称下方的使用数据服务生产API，直接跳转至数据服务页面，详情请参见数据服务概览。

基础信息

表的基础信息包括表的读取次数、收藏次数、浏览人数、产出任务、MaxCompute项目、负责人、创建时间、生命周期、存储量、描述和标签等信息。

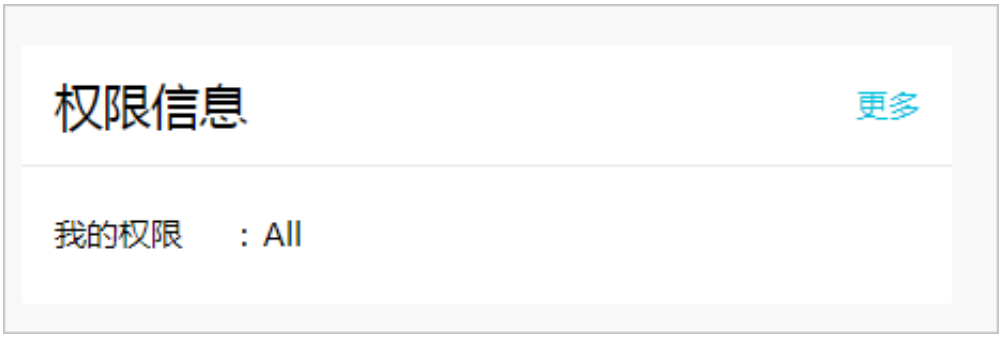
- 单击MaxCompute项目名称，即可跳转至项目详情页。
- 单击标签后面的 + ，即可为该表添加标签。

业务信息

表的业务信息主要包括DataWorks工作空间、环境类型和所属类目。

权限信息

表的权限信息为您展示您当前拥有的权限。单击右侧的更多，即可跳转至表权限申请页面。



技术信息

表的技术信息主要包括技术类型、DDL最后变更时间、最后数据变更时间、最后数据查看时间和计算引擎信息等信息。

- 时间格式默认为`yyyy-mm-dd hh:ss:mm`。
- 单击计算引擎信息后的点击查看按钮，即可查看计算引擎信息弹出框中的信息。

明细信息

表的明细信息是以表的元数据信息为主，显示字段定义、热度、等级、主外键及表结构变更情况。

- 字段信息

  表的字段信息包括字段名称、类型、描述、热度和主/外键等信息。

操作	说明
下载字段信息	单击后，即可直接下载相应的字段信息。
生成DDL语句	单击后，即可在弹出框中显示相应的建表语句。
生成select语句	单击后，即可在弹出框中显示相应的select语句。

- 分区信息

  通过表的分区信息模块，您可以查看表当前的分区，包括分区名、记录数、存储量、创建时间和最后更新时间等信息。

- 变更记录

  表的变更记录包括分区名、变更类型、粒度、时间和操作人等信息。

  您可以在下拉框中选择不同的变更类型，来过滤、筛选变更记录。

产出信息

如果表的数据会随着对应的任务周期性发生变化，您可以在该模块查看变化的情况、持续更新的数据等信息。

任务名称	操作	节点ID	执行时间(秒)	等待时间(秒)	开始时间	结束时间
[redacted]	代码 日志	700000624	8	6	2019-04-17 00:06:07	2019-04-17 00:06:15
[redacted]	代码 日志	700000624	103	3	2019-04-17 00:29:05	2019-04-17 00:30:48
[redacted]	代码 日志	700000624	10	25	2019-04-18 00:06:27	2019-04-18 00:06:37
[redacted]	代码 日志	700000624	100	8	2019-04-18 00:29:10	2019-04-18 00:30:50
[redacted]	代码 日志	700000624	11	5	2019-04-19 00:06:06	2019-04-19 00:06:17
[redacted]	代码 日志	700000624	100	3	2019-04-19 00:29:05	2019-04-19 00:30:45
[redacted]	代码 日志	700000624	10	57	2019-04-20 00:07:26	2019-04-20 00:07:36
[redacted]	代码 日志	700000624	100	7	2019-04-20 00:29:09	2019-04-20 00:30:49
[redacted]	代码 日志	700000624	12	4	2019-04-21 00:06:05	2019-04-21 00:06:17
[redacted]	代码 日志	700000624	97	6	2019-04-21 00:29:08	2019-04-21 00:30:45

血缘信息

表的血缘信息可以为您清晰地展示数据的来源和去向，并提供便捷的交互。

说明:

仅购买DataWorks标准版及以上版本的用户，可以查看表的血缘信息。

- **表血缘：您可以根据GUID搜索表的上下游表。**

- **字段血缘：您可以根据字段进行过滤。**

使用记录

- **频繁关联：频繁关联模块为您展示有多少人在使用当前的数据。**

- **访问统计：访问统计模块，以线型方式为您展示表的使用记录。**

数据预览

您可以通过表的数据预览模块，预览当前表的数据信息。

说明:

您需要拥有权限，方可预览生产环境的表。如果没有权限，单击去申请，前往申请页面进行申请。

使用说明

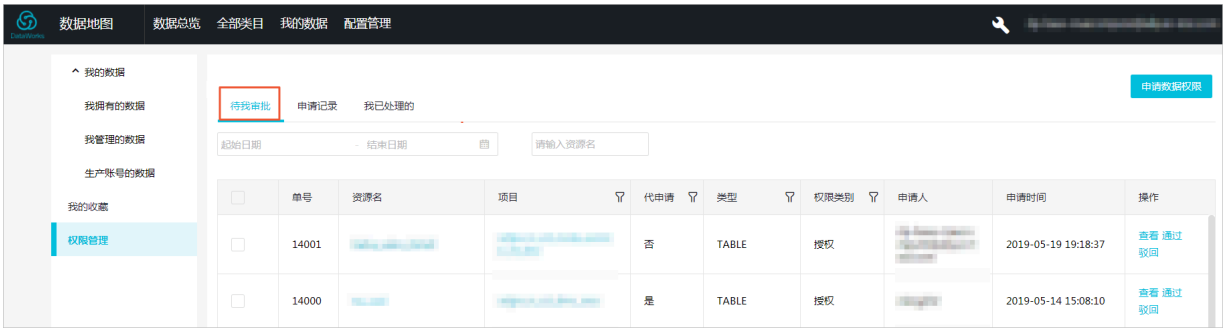
您可以编辑此界面，查看相关的历史版本和Markdown语法。您也可以根据数据的业务说明了解相关的信息。

9.6 权限管理

权限管理模块主要对待我审批、申请记录、我已处理、权限回收等相关表、资源、函数的权限申请进行管理。

待我审批

当前访问账号为管理员时，可通过待我审批模块，查看并审批其所有项目下的待审批记录，包括表、资源和函数等权限申请。



申请记录

当前访问账号可通过申请记录模块，查看其权限申请记录。

我已处理的

当前账号为管理员时，可通过我已处理的模块，查看其所有项目下已经处理过的权限申请记录，包括表、资源和函数等权限申请。

9.7 申请数据权限

本文将为您介绍如何申请数据权限。

DataWorks中有以下三种数据类型。

- 表：即数据表。
- 函数：即UDF，可以在SQL中使用的函数。
- 资源：例如文本文件、MapReduce的Jar文件等。

上述三种数据类型具有严格的权限控制机制，您需要申请权限后才能使用。

申请表权限（表数据预览权限）

1. 单击左上角的图标，选择全部产品 > 数据地图，即可进入数据地图页面。

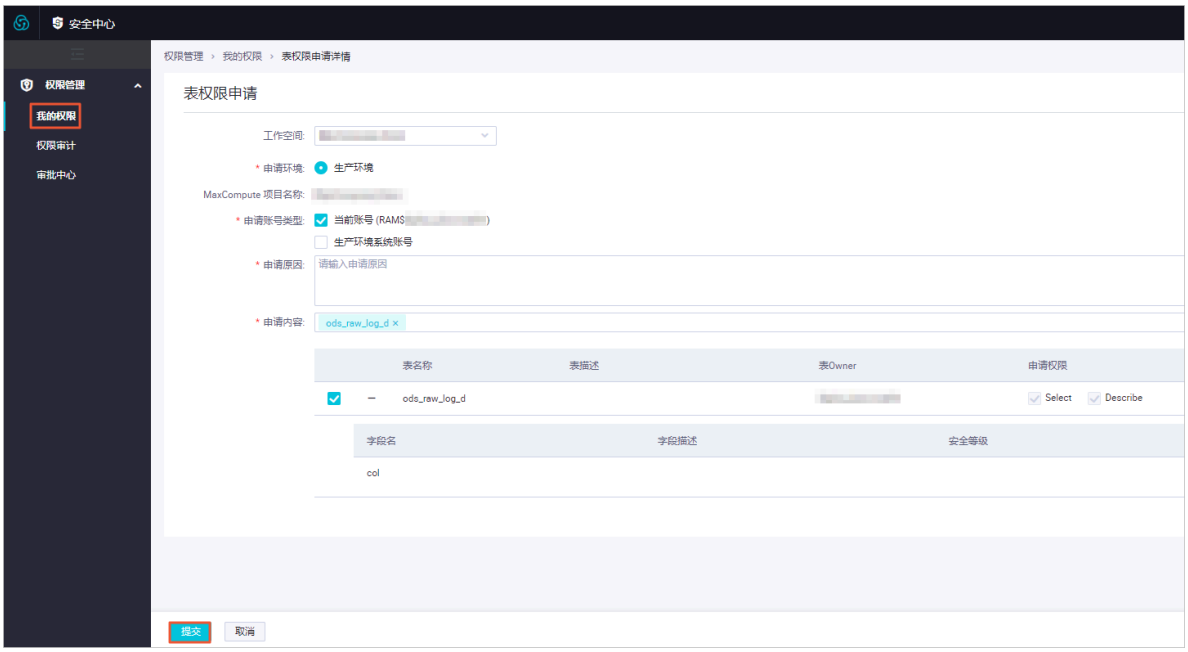
2. 在首页搜索需要申请权限的数据表，单击数据表名称进入表详情页面。

3. 在表详情页面中，单击申请权限，即可跳转至安全中心 > 我的权限页面。

说明：

如果您将表隐藏，便不会显示申请权限按钮。关于如何隐藏表，详情参见#unique_523/unique_523_Connect_42_section_6uh_f1o_4n9。

4. 填写表权限申请页面的各配置项。



配置	说明
工作空间	需要申请的表所在的工作空间。
申请环境	标准模式的工作空间包括开发环境和生产环境，简单模式的工作空间仅生产环境。

配置	说明
MaxCompute项目名称	选择的DataWorks工作空间对应的MaxCompute项目名称，默认不可修改。
申请账号类型	包括当前账号和生产环境系统账号。
申请原因	请简要填写申请原因，以便更快地通过审批。
申请内容	勾选需要申请的内容。

5. 配置完成后，单击提交，待审批通过后即可预览表数据。

> 说明:
> 权限申请提交后，您可以进入数据地图 > 我的数据 > 权限管理 > 申请记录页面，查看申请状态。

申请函数和资源权限

1. 进入数据地图 > 我的数据 > 权限管理页面。
2. 单击右上方的申请数据权限。

3. 填写申请授权对话框中的配置。

配置	说明
申请类型	支持函数和资源两种类型。
权限归属人	支持本人申请和代理申请。 · 本人申请：选择该项，审批通过后权限归属于当期用户。 · 代理申请：选择代理申请，需填写对方用户名。审批通过后权限归属于被代理人。
项目名	选择所需申请函数或者资源所在的项目名称（对应的MaxCompute项目名称），支持本组织范围内模糊匹配查找。
函数名称/资源名称	输入项目中的函数或资源名称。资源名称请填写完整，包括文件后缀，例如my_mr.jar。
权限有效期	申请表权限的时长，单位为天，不填则默认为永久。超过申请权限时长时，该权限将被系统自动回收。
申请理由	请简要填写申请理由，以便更快地通过审批。

4. 填写完成后单击提交，然后等待审批。您可以通过数据地图 > 我的数据 > 权限管理 > 申请记录查看申请状态。

9.8 配置管理

本文为您介绍数据地图的配置管理功能模块。

1. 以开发者身份进入DataWorks管理控制台，单击项目列表下对应项目后的进入数据开发。
2. 单击左上方菜单栏中的数据地图。
3. 在数据地图页面，单击顶部菜单栏中的配置管理。

   配置管理页面包括类目导航配置和项目管理配置两个模块。

类目导航配置

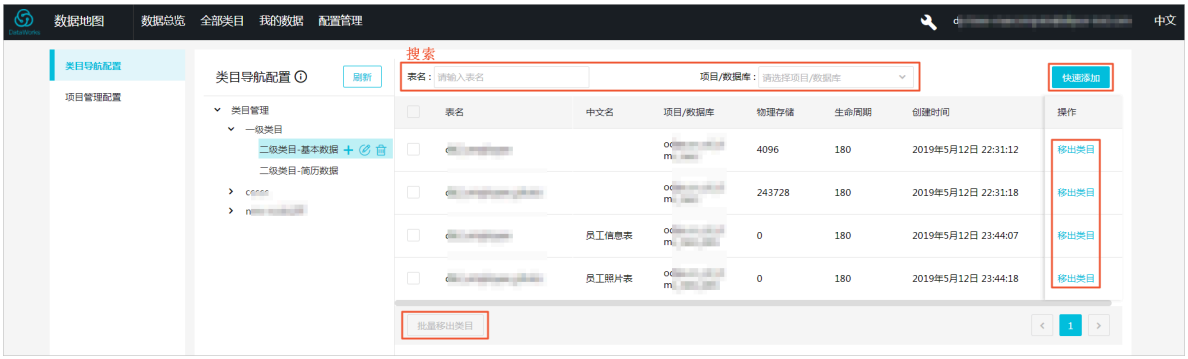
类目导航配置页面主要是可以创建类目和添加相关表到类目里。添加类目的目的是方便您更好的管理表。

1. 单击类目管理后的 ＋，添加一级类目。

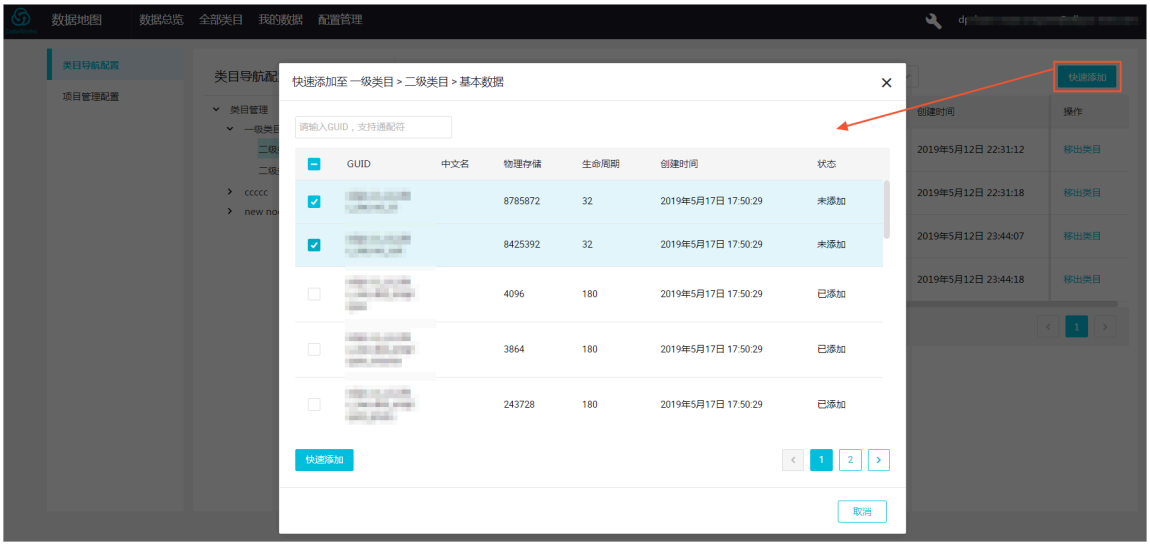
2. 单击一级类目后的 ，添加所属二级类目。

以此类推，最多可支持四级类目的设置。其中 表示编辑该类目名称， 表示删除该类目。

3. 类目配置完成后，即可进行如下操作：



- 快速添加：只能选择未添加的表。已经添加的表如果移出类目，也可以重新选择添加至类目。



- 搜索：可以通过表名和项目/数据库来搜索展示您的表信息。
- （批量）移出类目：已添加的表可以移出类目。

项目管理配置

- 线上自动化测试项目： MaxCompute表管理可以操作预览开发表的数据和预览生产表的数据两个按钮。
- Dataworks调度系统预发项目：MaxCompute表管理可以操作预览开发表的数据和预览生产表的数据两个按钮。

10 数据分析

10.1 数据分析概述

数据分析是MaxCompute数据的取数据工具，主要提供临时查询、个人表数据上传下载和计划任务等核心功能。拥有分析师角色的DataWorks项目成员方可使用数据分析。

功能概述

单击数据开发页面左上角的图标，选择全部产品 > 数据分析，即可进入数据分析功能模块。

数据分析的核心功能如下所示：

- 临时查询：仅支持MaxCompute SQL类型。
- 个人表：
  - 支持SQL方式和可视化方式建表。
  - 支持上传CSV和TXT两种格式到个人表，最大可上传200M。如果文件过大，请拆分后再上传。
- 公共表：MaxCompute表元数据，即在数据地图页面查看的MaxCompute表。详情请参见#unique_527。
- 函数：MaxCompute内置函数。

> **说明:**
> 各功能的详细使用方式请参见数据开发模块。数据开发详情请参见#unique_11。

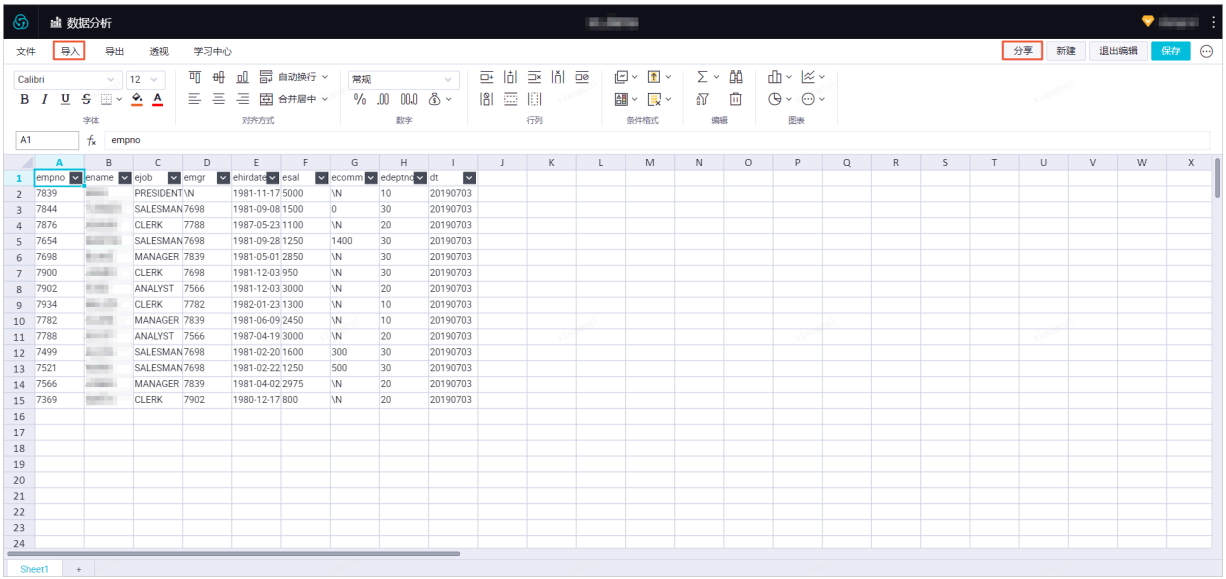
10.2 电子表格

本文将为您介绍数据分析的电子表格功能。

简介

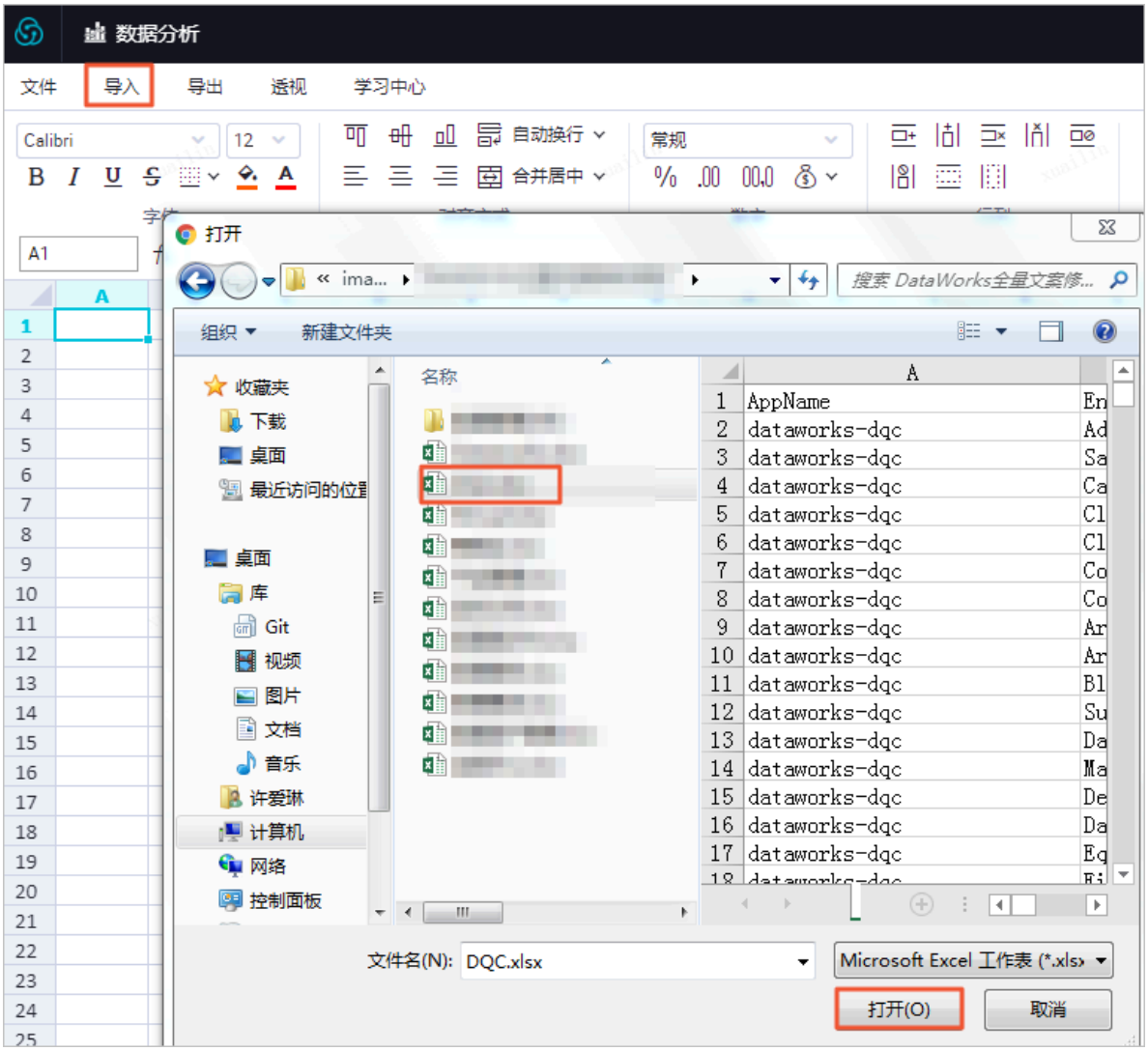
DataWorks的电子表格功能更方便您对数据结果进行操作。数据分析支持三种获取数据的方式，并且加入了数据导入和分享的独特功能。支持很多Excel的功能，例如用Sheet表示创建新的页面，并提供相关函数的功能。

数据导入和分享

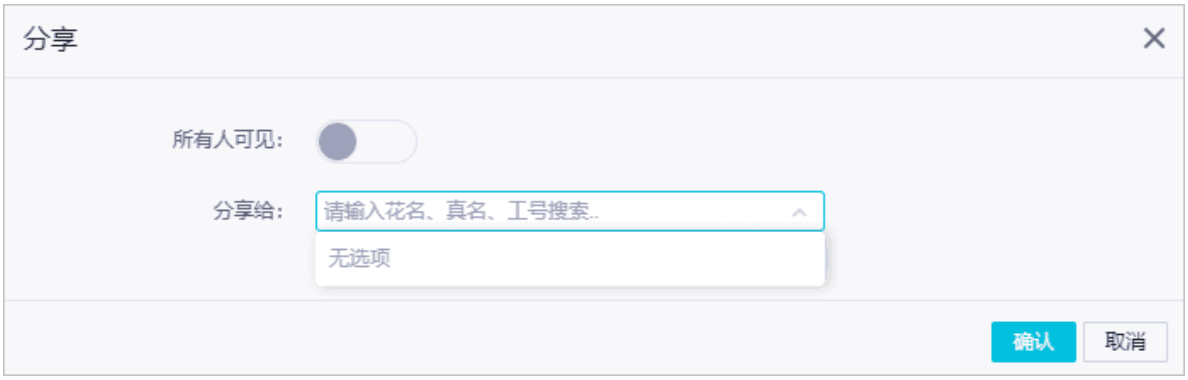


empno	ename	ejob	emgr	ehirdate	esal	ecomm	edeptno	dt
7839	[illegible]	PRESIDENT	\N	1981-11-17	5000	\N	10	20190703
7844	[illegible]	SALESMAN	7698	1981-09-08	1500	0	30	20190703
7876	[illegible]	CLERK	7788	1987-05-23	1100	\N	20	20190703
7654	[illegible]	SALESMAN	7698	1981-09-28	1250	1400	30	20190703
7698	[illegible]	MANAGER	7839	1981-05-01	2850	\N	30	20190703
7900	[illegible]	CLERK	7698	1981-12-03	950	\N	30	20190703
7902	[illegible]	ANALYST	7566	1981-12-03	3000	\N	20	20190703
7934	[illegible]	CLERK	7782	1982-01-23	1300	\N	10	20190703
7782	[illegible]	MANAGER	7839	1981-06-09	2450	\N	10	20190703
7788	[illegible]	ANALYST	7566	1987-04-19	3000	\N	20	20190703
7499	[illegible]	SALESMAN	7698	1981-02-20	1600	300	30	20190703
7521	[illegible]	SALESMAN	7698	1981-02-22	1250	500	30	20190703
7566	[illegible]	MANAGER	7839	1981-04-02	2975	\N	20	20190703
7369	[illegible]	CLERK	7902	1980-12-17	800	\N	20	20190703

- 导入：将本地数据导入电子表格。



- 分享：可以设置所有人可见或分享给指定人员。分享给指定的人员时，您可以输入花名、真名、工号搜索人员。

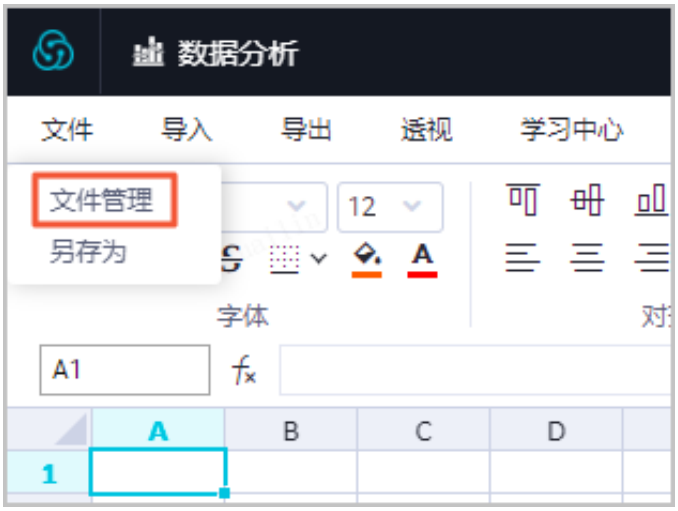


说明:

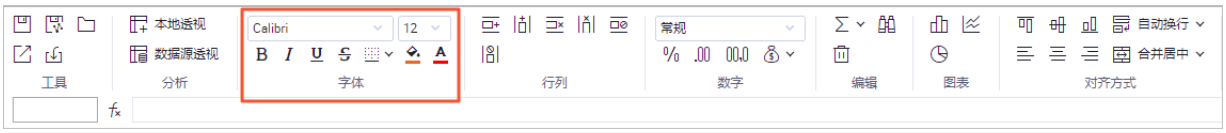
您需要将文件进行保存后，才会显示分享按钮。

- 保存：快捷键为Ctrl+s。

鼠标悬浮至文件，单击文件管理，即可查看保存的文件。单击另存为，可以将文件保存至新的目录。



字体

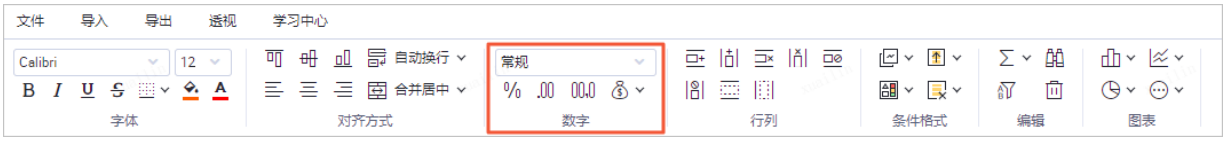


- 加粗：将文本加粗。
- 倾斜：将文字变为斜体。
- 下划线：给文字添加下划线。
- 中划线：给文字添加中划线。
- 边框：给文本添加边框。

列表

- 插入行：在工作簿中添加新的单元行。
- 插入列：在工作簿中添加新的单元列。
- 删除行：在工作簿中删除选中的单元行。
- 删除列：在工作簿中删除选中的单元列。
- 锁定行：在工作簿中锁定选中的单元行。
- 锁定列：在工作簿中锁定选中的单元列。
- 隐藏行：在工作簿中隐藏选中的单元行。
- 隐藏列：在工作簿中隐藏选中的单元列。

数字

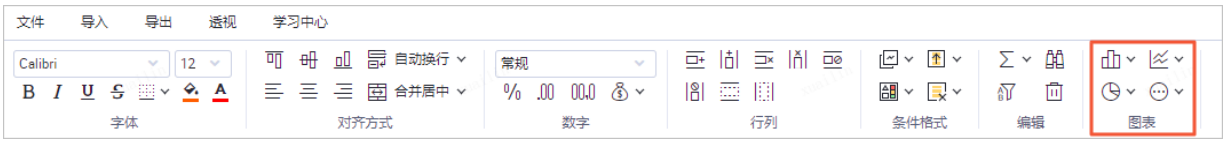


- 数据类型：选择单元格格式，例如数字、货币、短日期、长日期、时间、百分比、分数、科学计数和文本等。
- 百分比：将单元格的数据类型设置为百分比类型。
- 两位小数：使单元格的数据保留两位小数。
- 千位分割：将单元格数据的千位以逗号形式分割，例如1,005。
- 货币：将单元格的数据类型设置为货币类型，例如人民币、美元、英镑、欧元和法郎等。

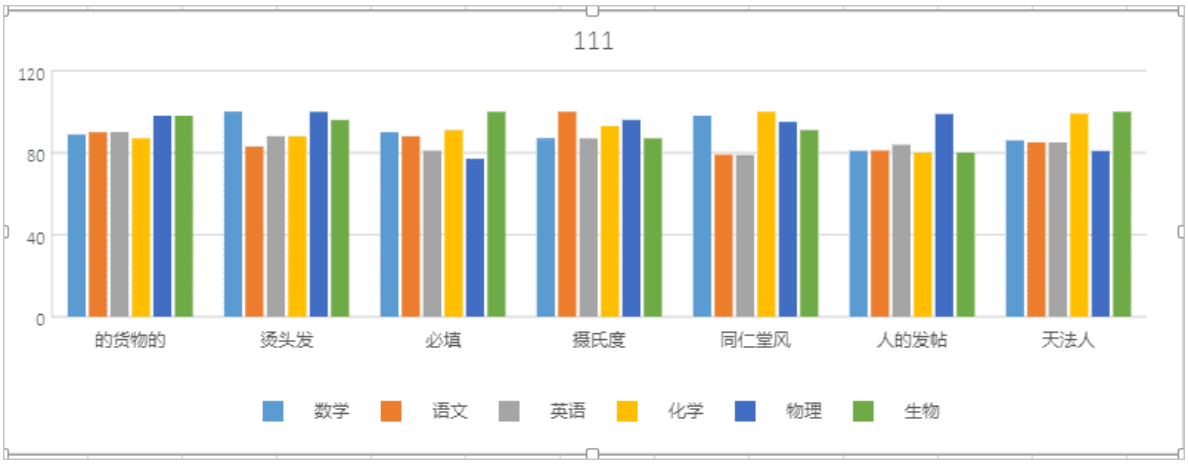
编辑

- 自动求和：支持求和、平均值、计数、最大值、最小值五种类型。
- 查找：您可以直接单击查找，也可以使用快捷键Ctrl+F即会弹出相应的输入框。
- 筛选和排序：您可以筛选数据进行排序或降序处理。
- 清除：将选中的内容直接删除。

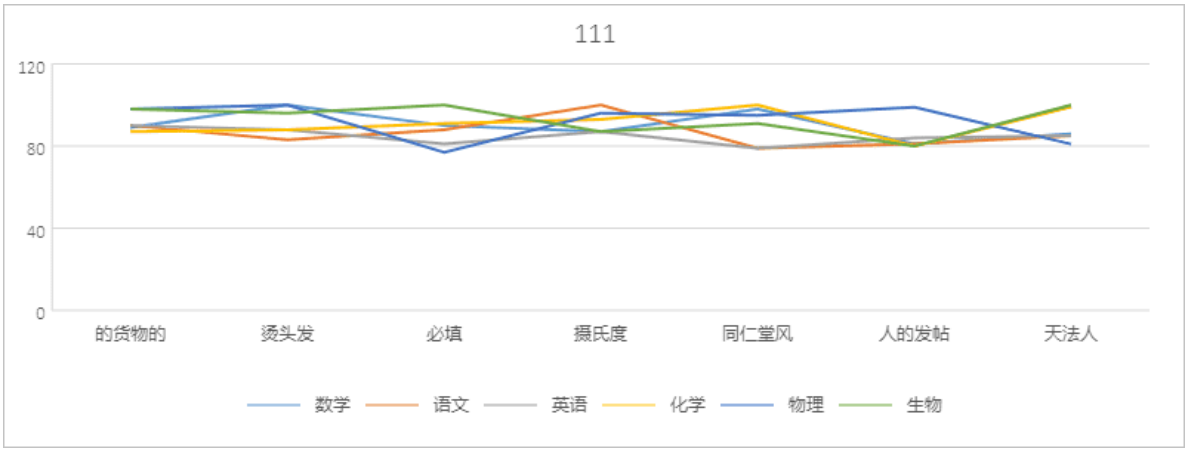
图表



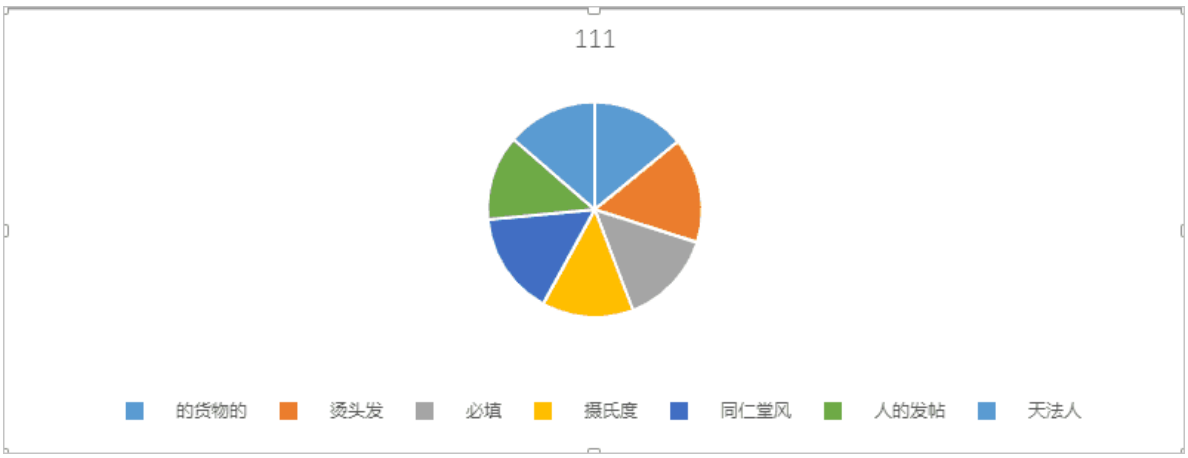
- 柱状图



- 折线图



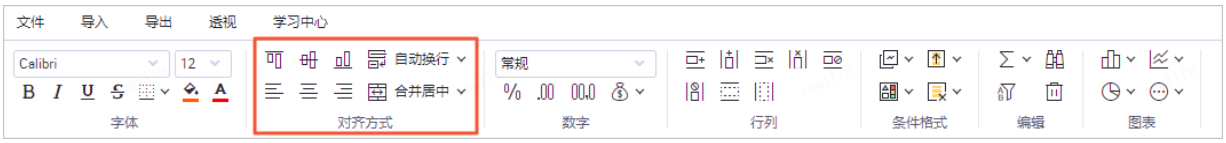
- 饼图



- 更多

  您还可以选择面积图、条形图和散点图。

对齐方式



- 顶端对齐：沿顶端对齐文字。
- 垂直居中：对齐文本，使其在单元格中上下居中。
- 底端对齐：沿底端对齐文字。
- 自动换行：多行显示超长文本，便于看到所有内容。
- 左对齐：将文本靠左对齐。
- 水平居中：将文本居中对齐。
- 右对齐：将文本靠右对齐。
- 合并居中：将选择的多个单元格合并成一个较大的单元格，并将新单元格内容居中。

10.3 透视

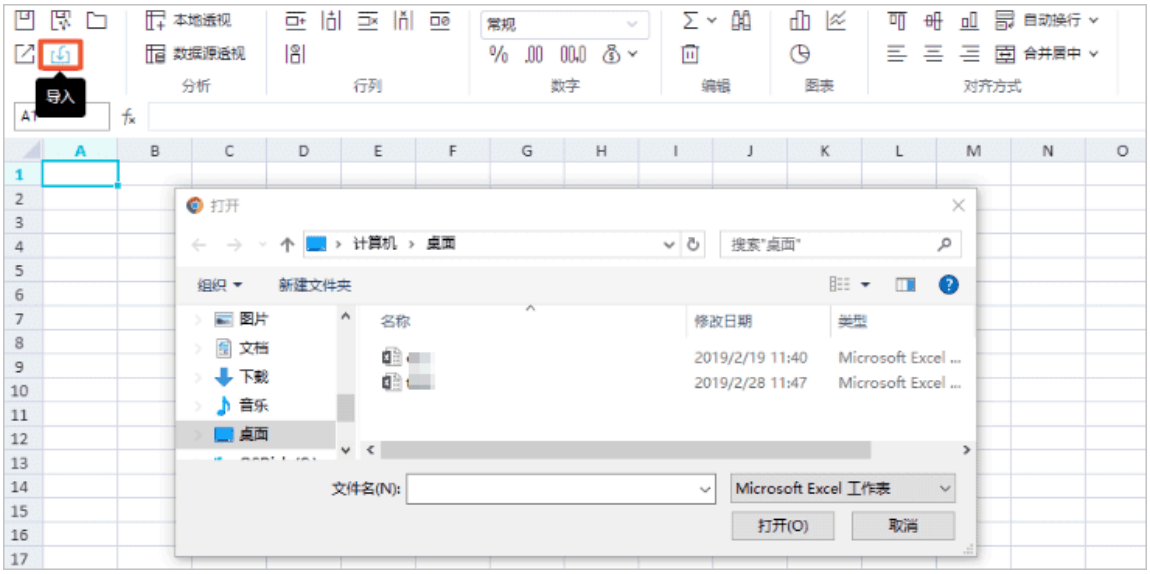
本文为您介绍数据分析组件的两种数据来源场景，以及两种场景下的透视功能案例。

数据来源

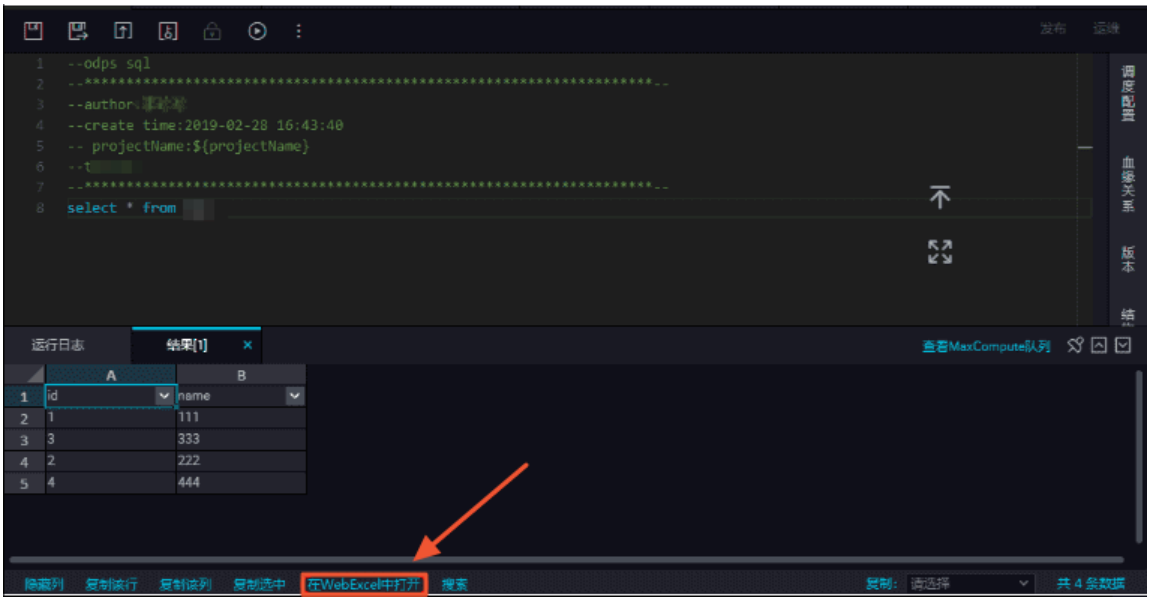
数据分析组件的数据来源一般分为如下两个场景：

- 场景一

1. 导入数据。一般是导入本地的Excel文件。



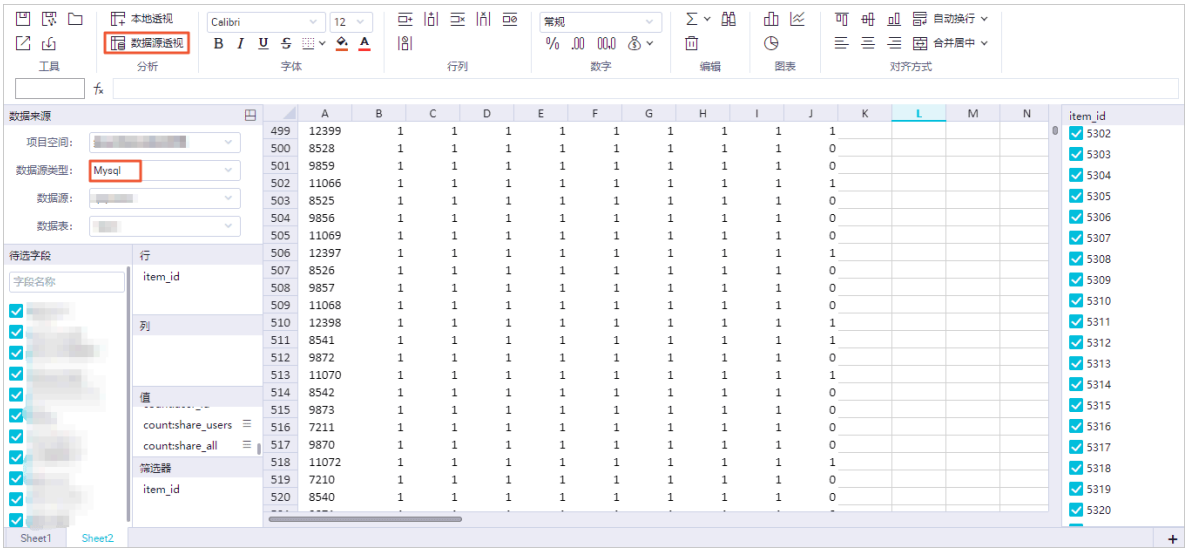
2. 在数据开发界面创建#unique_530，查询出结果后，会直接以电子表格的形式展示。您可以在DataWorks中执行操作，或者在WebExcel中打开，也可以自由拷贝内容粘贴到本地Excel文件中。



- 场景二

  数据源：在选择数据源透视之前，要先添加MySQL数据源，如下图所示。

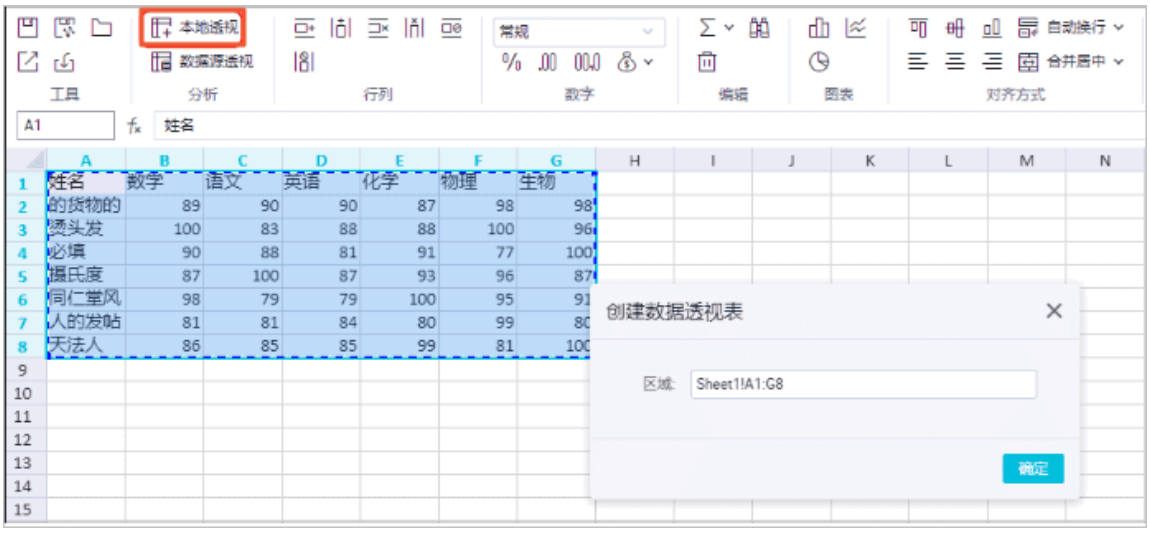
  > **说明：**
  > 目前仅支持MySQL数据源。



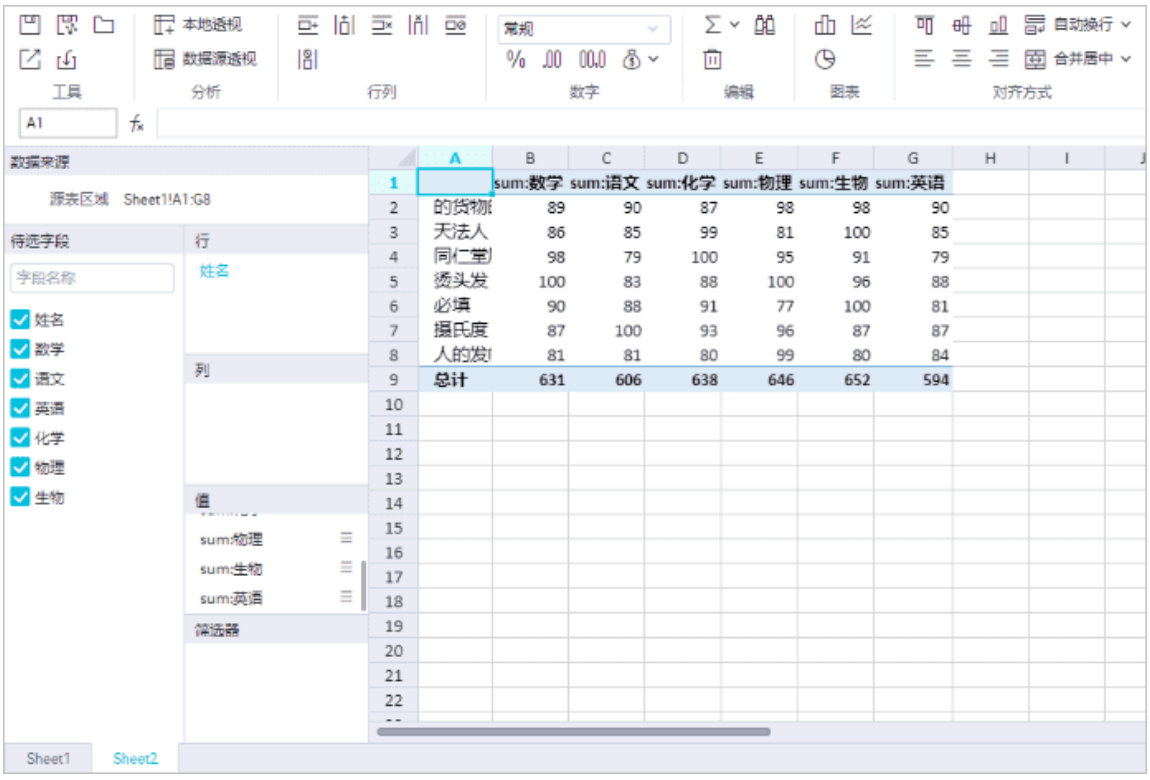
透视功能案例

- 本地透视

1. 选中您需要透视的数据，单击本地透视按钮会弹出创建数据透视表界面。

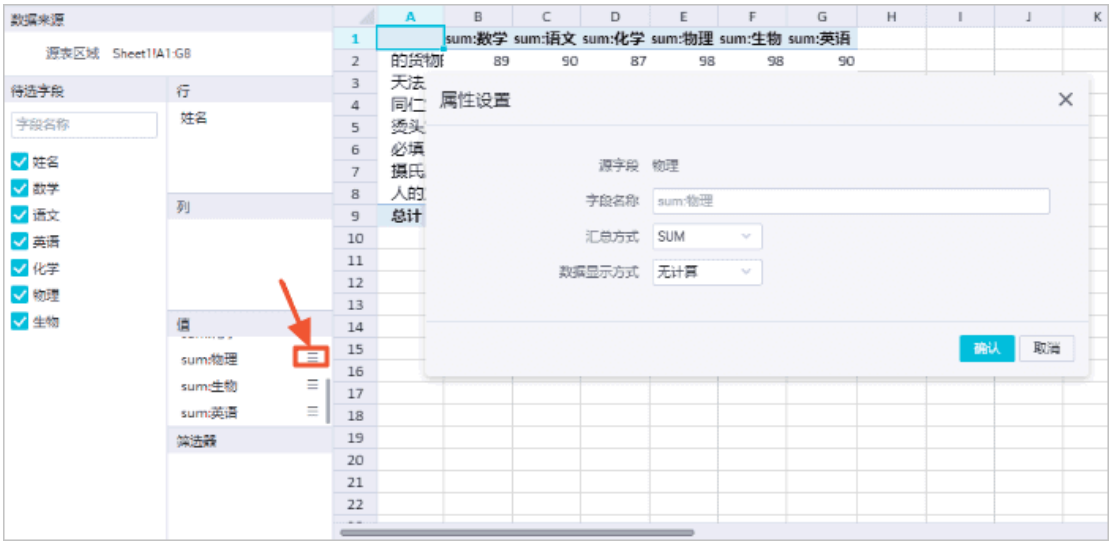


2. 单击确定后，会加载透视界面。



- 数据来源：是指您在前面上传的Excel表格里选中的数据。
- 待选字段：是指您上传数据的横坐标。
- 行：将您选择的字段拖到行文本框，行字段里的每个值都会以行的形式展现。
- 列：将您选择的字段拖到列文本框，列字段里的每个值都会以列的形式展现。

- 值：单击值里面的属性设置按钮，您可以设置汇总方式和数据显示方式字段名称默认不允许修改。



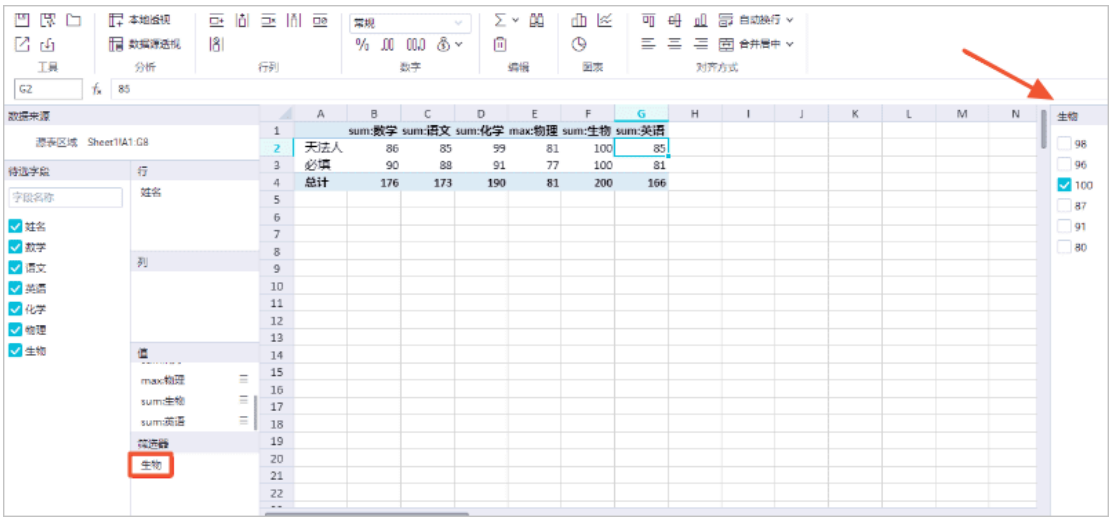
■ 源字段：上传的Excel表格的列名。

■ 字段名称：是汇总方式：源字段的组合。

■ 汇总方式：包括SUM、COUNT、MAX、MIN、AVG。

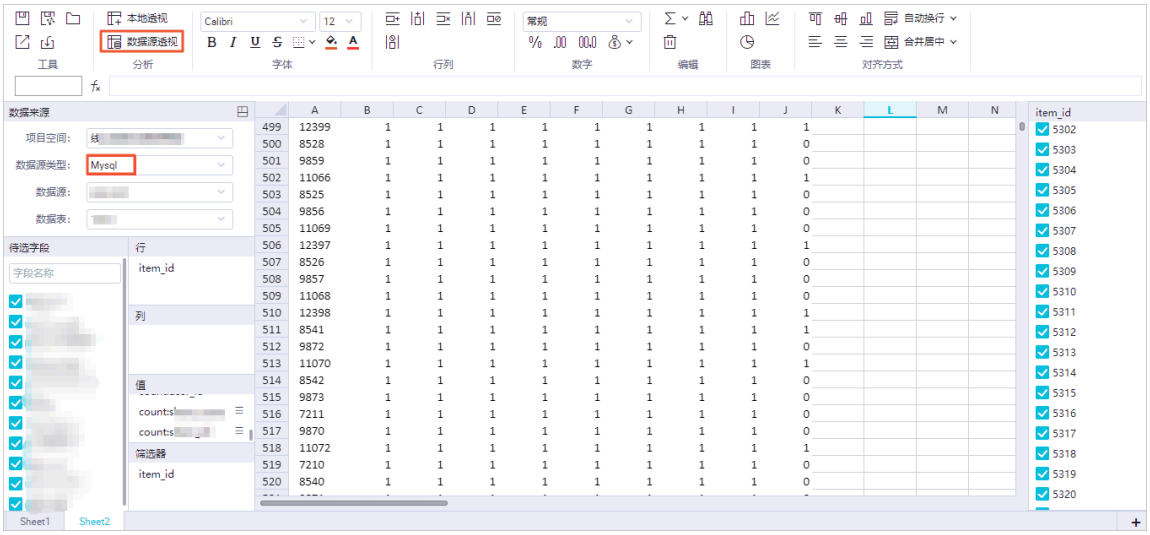
■ 数据显示方式：单击下拉框有两种方式供选择，即无计算和总计的百分比。

- 筛选器：将需要筛选的列拖到筛选框，您可以根据右边的内容进行筛选。

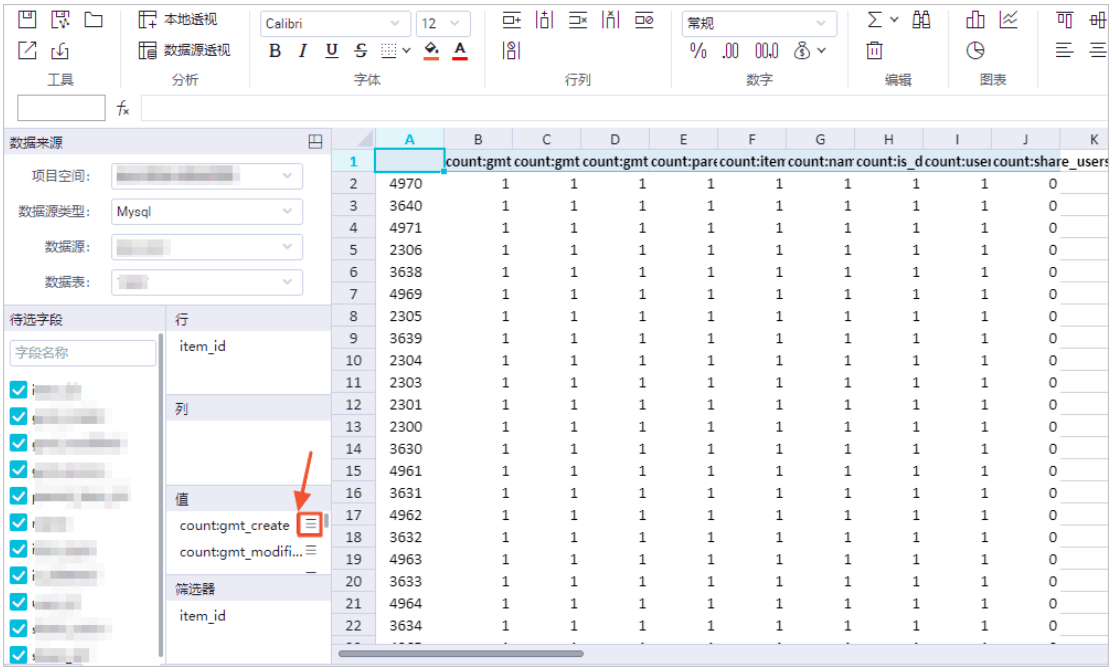


- 数据源透视

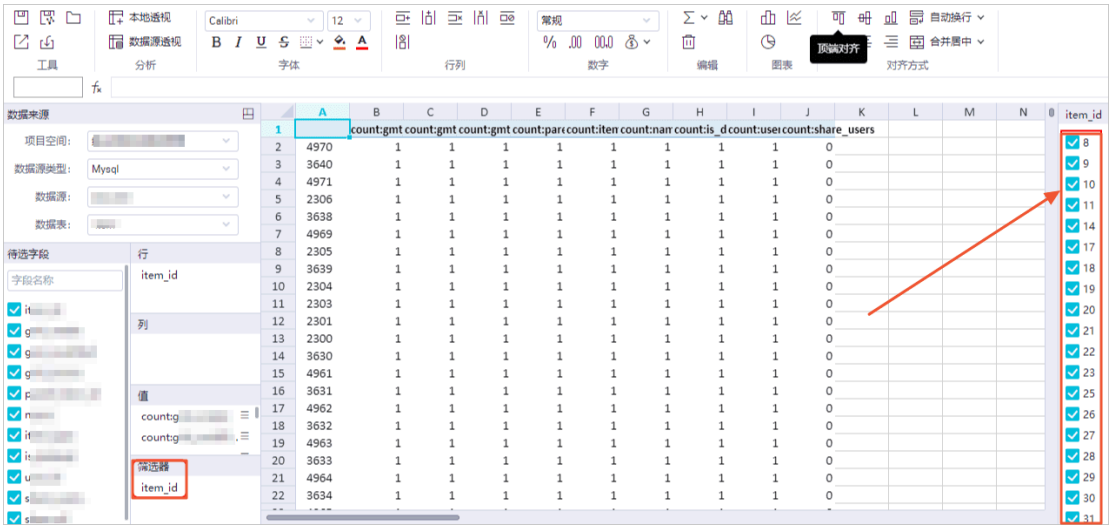
1. #unique_531。
2. 单击数据源透视，选择项目空间、数据源类型、数据源和数据表。



- 项目空间：用户在同一个工程（或者同一个事务）中工作环境的集合。
- 数据源类型：选择数据源类型。
- 数据源：在数据集成界面添加数据源成功后能从下来框搜索出来。
- 数据表：选择数据库中的表。
- 待选字段：将您选择的数据表里的所有字段都展现出来。您可以选择全部字段，也可以选择部分字段。
- 行：将您选择的字段拖到行文本框，行字段里的每个值都会以行的形式展现。
- 列：将您选择的字段拖到列文本框，列字段里的每个值都会以列的形式展现。
- 值：单击值里面的属性设置按钮，您可以设置汇总方式和数据显示方式。字段名称默认不允许修改。



- 筛选器：将需要筛选的列拖到筛选框，您可以根据右边的内容进行筛选。



10.4 图表使用说明

10.4.1 柱形图

本文为您介绍柱形图的使用说明。

在数据分析领域，柱形图是最为常见也是应用最为广泛的。在worksheet中以行或者列来组织的数据，可以用柱形图来绘制和表示。

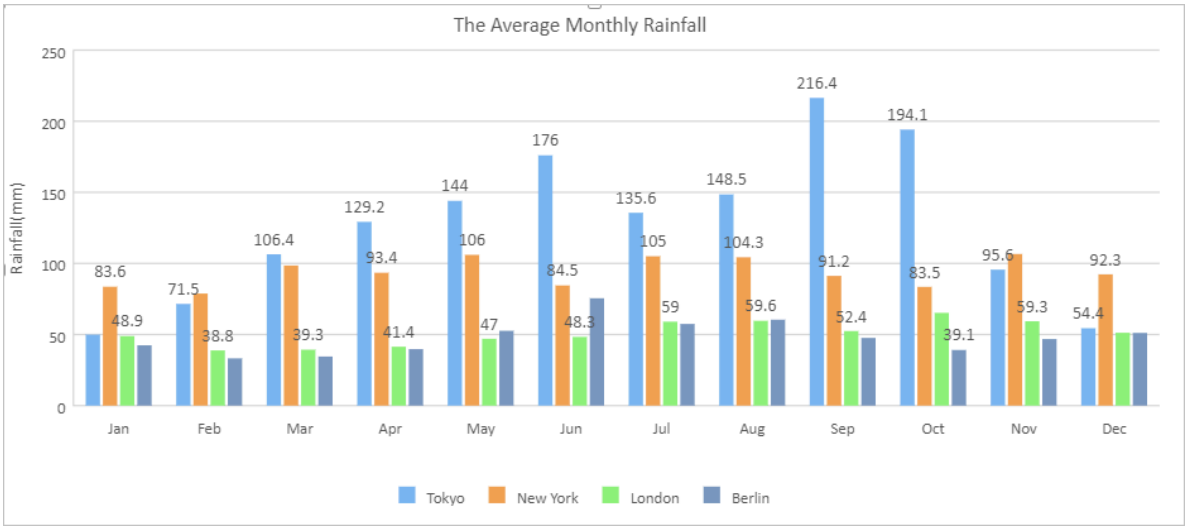
柱状图用高度反映数据差异，可以展示有多少项目（频率）会落入一个具有一定特征的数据段中。例如，分析公司人员构成是否存在老龄化现象，可以通过柱形图看到25岁以下的员工有多少、25岁到35岁之间的员工有多少等这种年龄的分布情况。同时，柱图还可以用来表示含有较少数据值的趋势变化关系。

簇状柱形图

- 数据示例：

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Tokyo	49.9	71.5	106.4	129.2	144	176	135.6	148.5	216.4	194.1	95.6	54.4
New York	83.6	78.8	98.5	93.4	106	84.5	105	104.3	91.2	83.5	106.6	92.3
London	48.9	38.8	39.3	41.4	47	48.3	59	59.6	52.4	65.2	59.3	51.2
Berlin	42.4	33.2	34.5	39.7	52.6	75.5	57.4	60.4	47.6	39.1	46.8	51.1

- 图形示例：

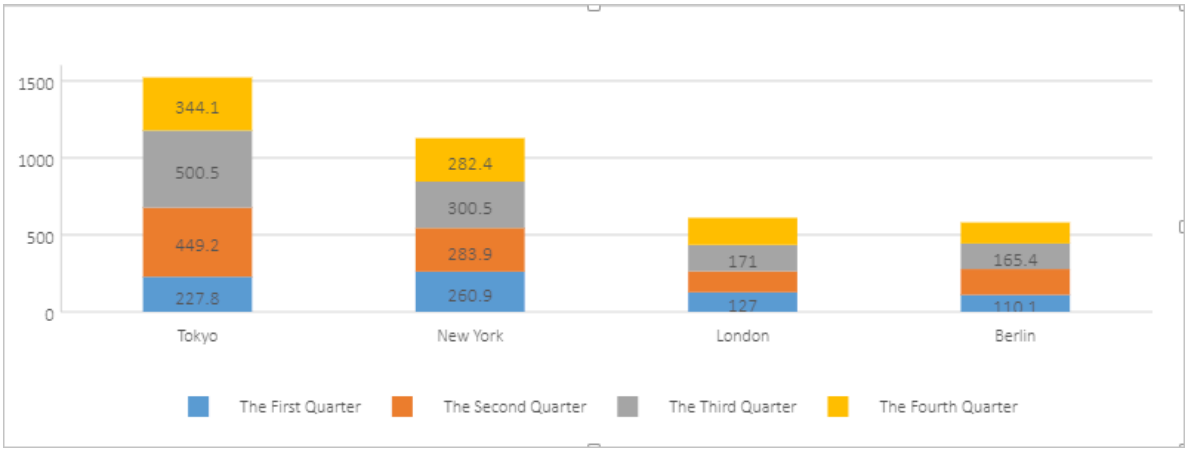


堆积柱形图

- 数据示例：

	Tokyo	New York	London	Berlin
The First Quarter	227.8	260.9	127	110.1
The Second Quarter	449.2	283.9	136.7	167.8
The Third Quarter	500.5	300.5	171	165.4
The Fourth Quarter	344.1	282.4	175.7	137

- 图形示例：

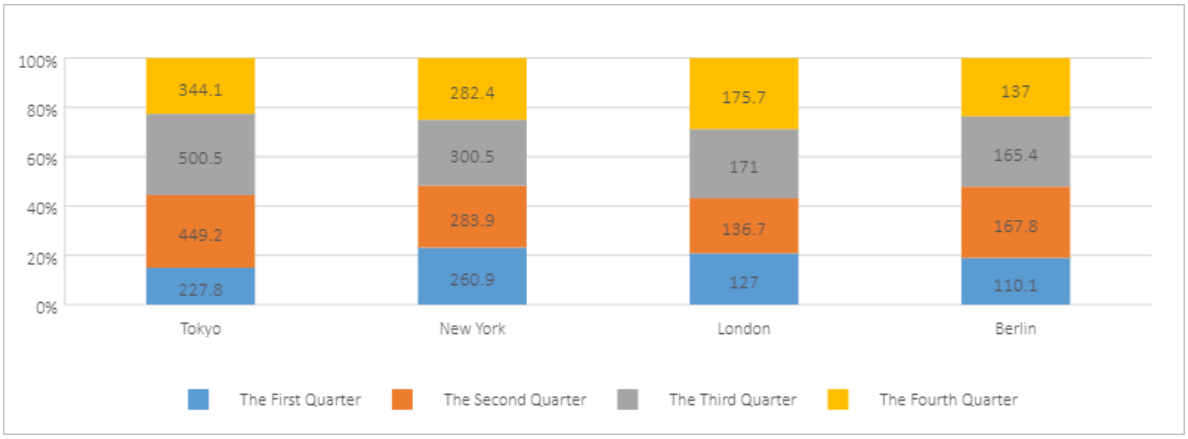


百分比堆积柱形图

- 数据示例：

	Tokyo	New York	London	Berlin
The First Quarter	227.8	260.9	127	110.1
The Second Quarter	449.2	283.9	136.7	167.8
The Third Quarter	500.5	300.5	171	165.4
The Fourth Quarter	344.1	282.4	175.7	137

- 图形示例：



10.4.2 折线图

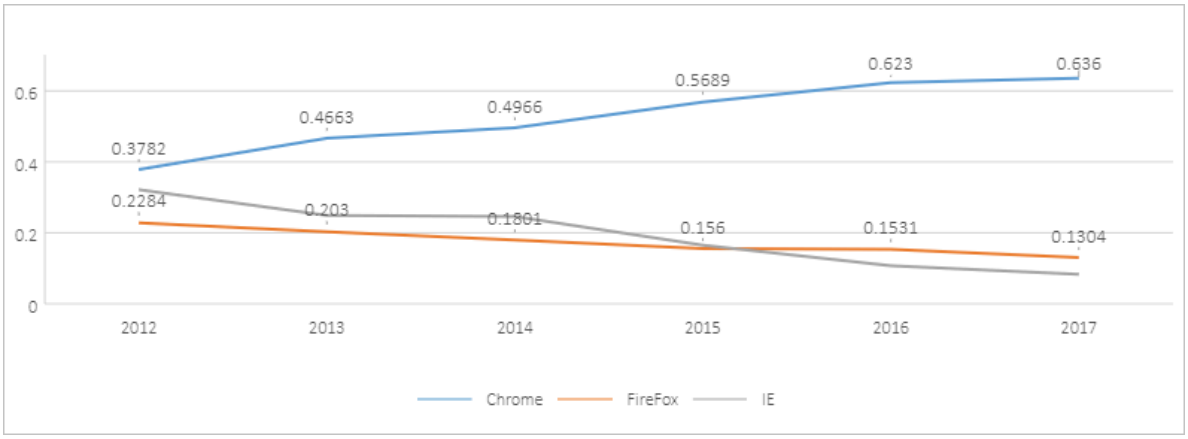
本文为您介绍折线图的使用说明。

在折线图中，分类数据沿着水平轴均匀分布，值数据沿着垂直轴均匀分布。折线图可以用于反映随时间变化而变化的关系，尤其是在数据趋势比单个数据点更重要的场合，因此非常适合用于显示相等的时间间隔里（例如月、季度或者财年）数据的变化趋势。

· 数据示例：

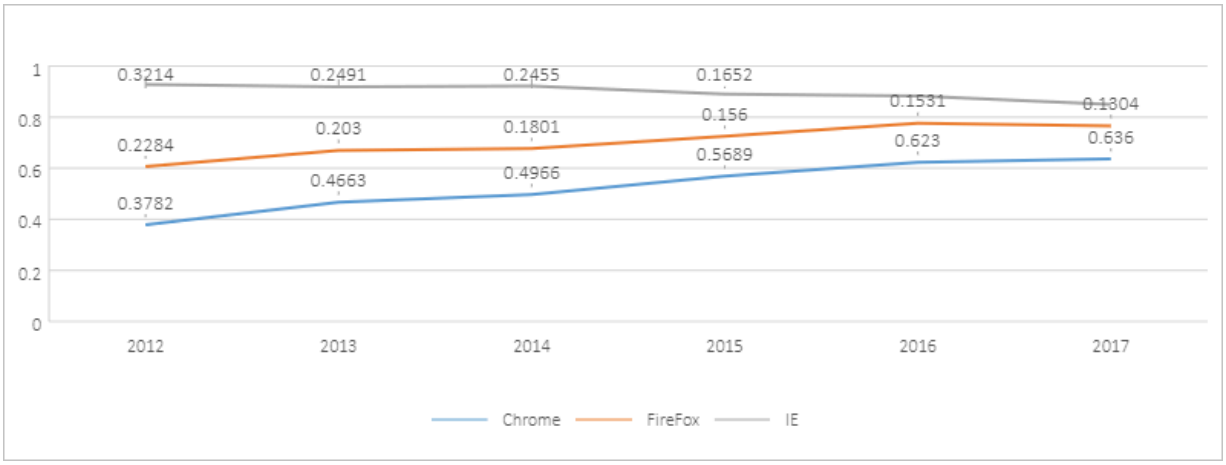
	2012	2013	2014	2015	2016	2017
Chrome	0.3782	0.4663	0.4966	0.5689	0.623	0.636
Firefox	0.2284	0.203	0.1801	0.156	0.1531	0.1304
IE	0.3214	0.2491	0.2455	0.1652	0.1073	0.0834

· 图形示例：



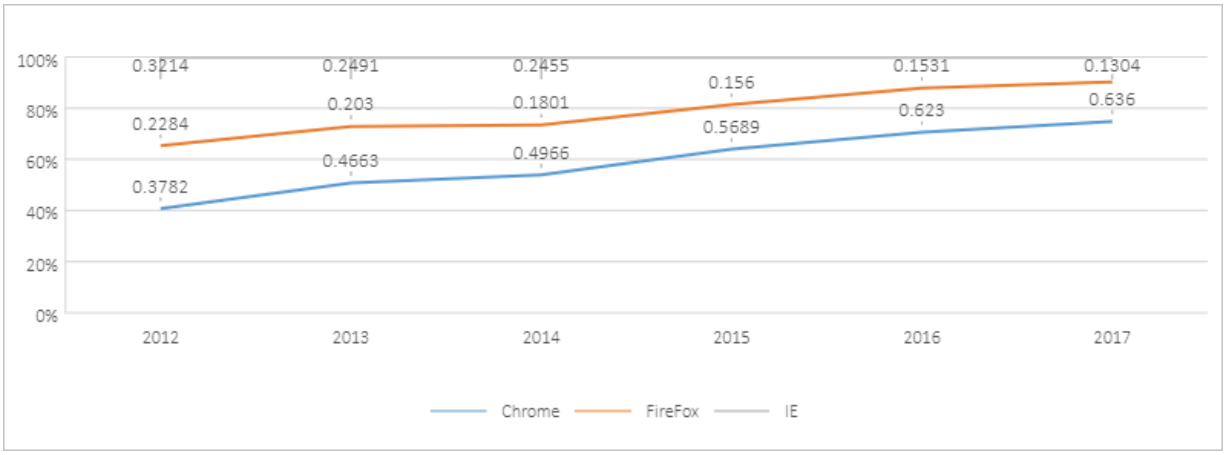
堆积折线图

图形示例：



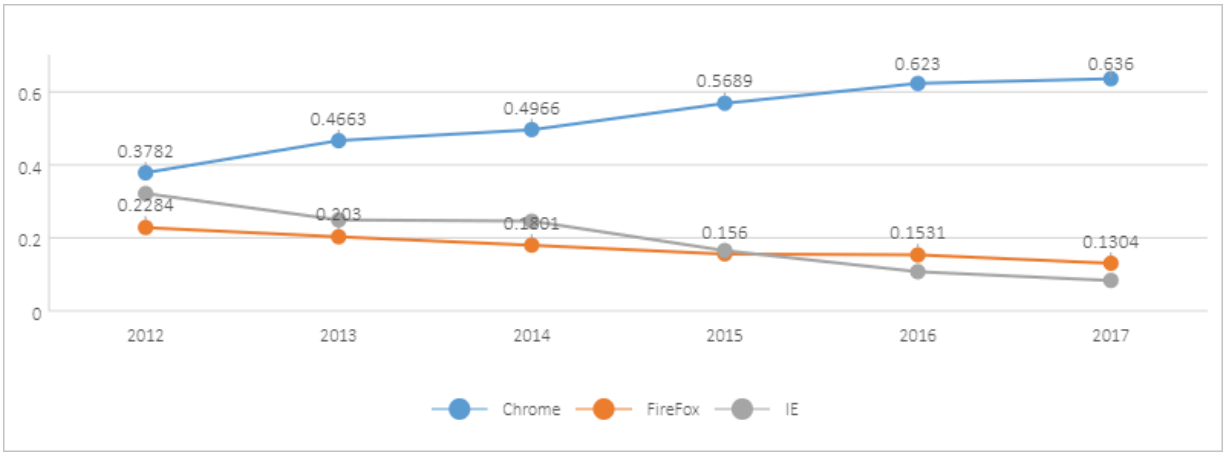
百分比堆积折线图

图形示例：



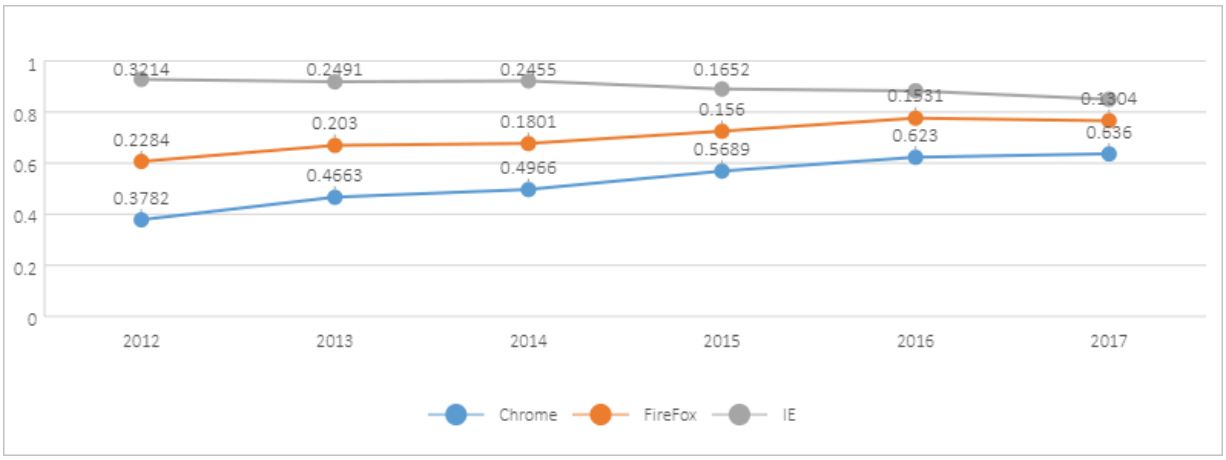
数据点折线图

图形示例：



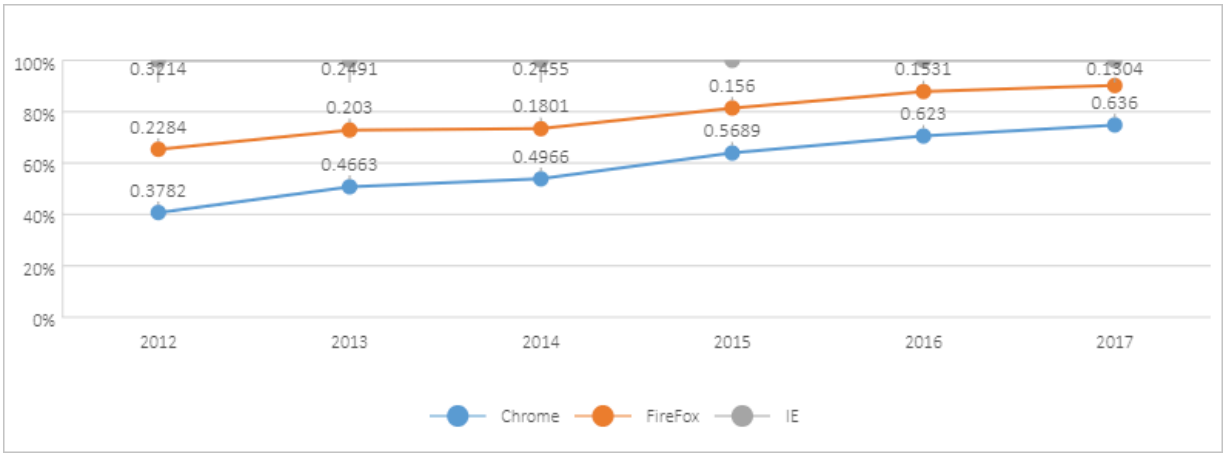
数据点堆积折线图

图形示例：



数据点百分比堆积折线图

图形示例：



10.4.3 饼形图和圆环图

本文为您介绍饼形图和圆环图的使用说明。

饼形图

在工作表中以列或行的形式排列的数据可以绘制为饼形图。

显示构成比例关系时，推荐使用饼形图，可以展示每一部分所占整体的百分比。例如，产品A预计销售额占所有产品销售额的最大份额。

为了最大限度地发挥饼形图的展示效果，在使用饼形图时不宜超过七种成分。为了便于观察，建议您将各种数据成分按顺时针方向排列，并将最重要的数据成分放置在饼形图里靠近12点钟的位置。如果每种数据成分同等重要，或者没有重要性的区分，您可以将它们按照从大到小的顺序排列。

当一个饼形图中各指标所占比例接近时，无法直观判断面积的大小，此时您可以选择圆环图来呈现，会使规律更加清晰。

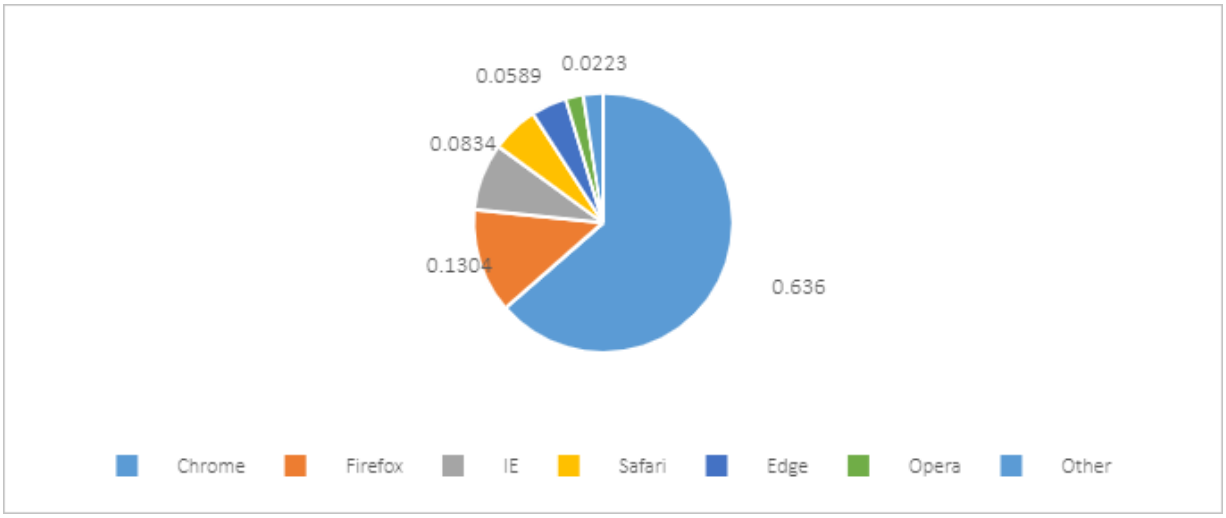
目前我们支持饼形图和圆环图。

- 饼形图仅显示一个数据系列。
- 圆环图以圆环的形式显示数据。圆环图可以包含多个数据系列，其中每个圆环分别代表一个数据系列。

数据示例：

	Chrome	Firefox	IE	Safari	Edge	Opera	Other
2017	0.636	0.1304	0.0834	0.0589	0.0443	0.0223	0.0246

图形示例：

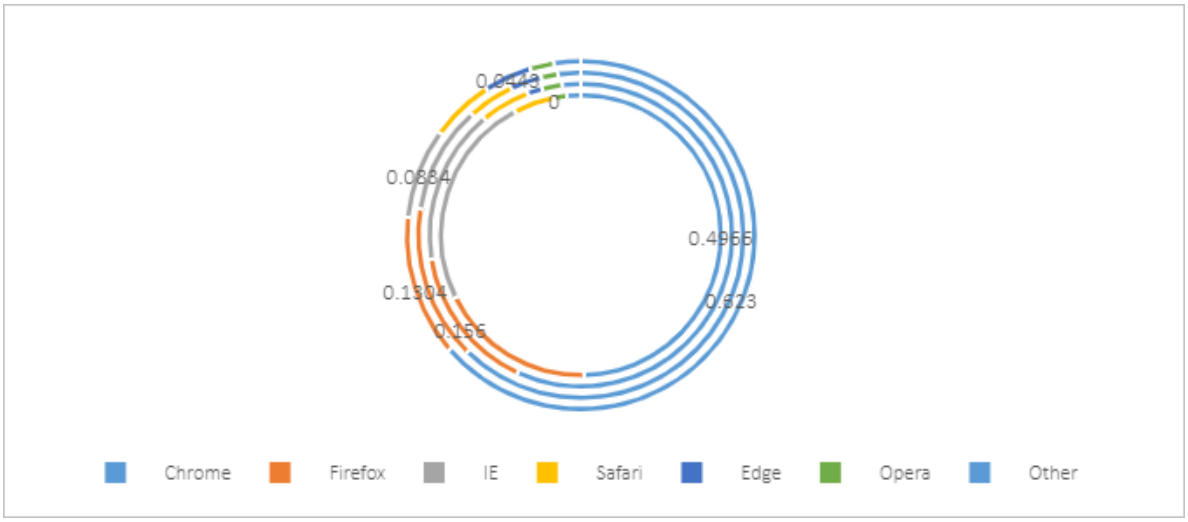


圆环图

- **数据示例：**

	Chrome	Firefox	IE	Safari	Edge	Opera	Other
2014	0.4966	0.1801	0.2455	0.047	0	0.015	0.0158
2015	0.5689	0.156	0.1652	0.0529	0.0158	0.022	0.0192
2016	0.623	0.1531	0.1073	0.0464	0.0311	0.0166	0.0225
2017	0.636	0.1304	0.0834	0.0589	0.0443	0.0223	0.0246

- **图形示例：**



10.4.4 面积图

本文将为您介绍面积图的使用说明。

面积图可以用于绘制随着时间变化而变化的图，主要突出数值之和的总体趋势。

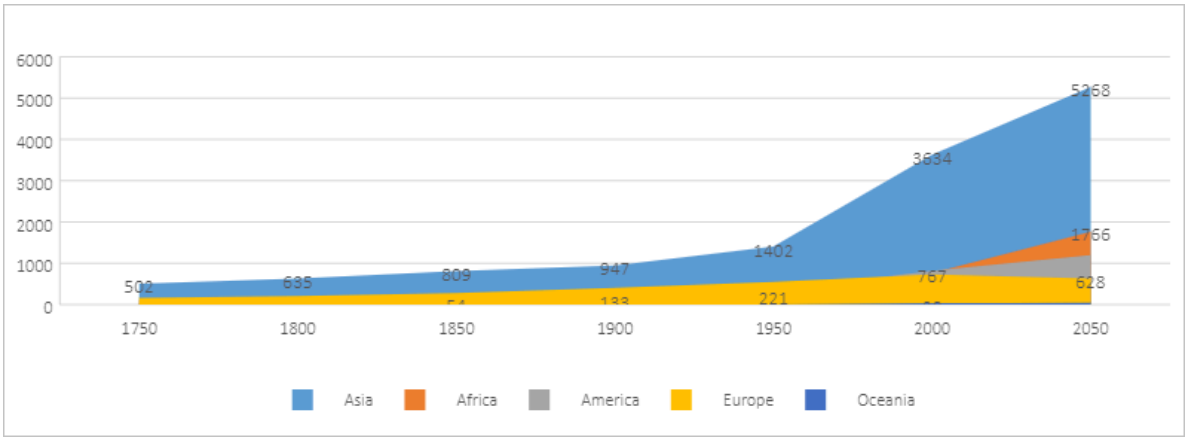
与折线图较为类似，面积图强调变量随时间而变化的程度，也可以用于引起人们对总值趋势的关注。面积图通过填充颜色或图案的面积来显示数据，面积片数不宜超过5片。

目前支持堆积面积图和百分比堆积面积图。

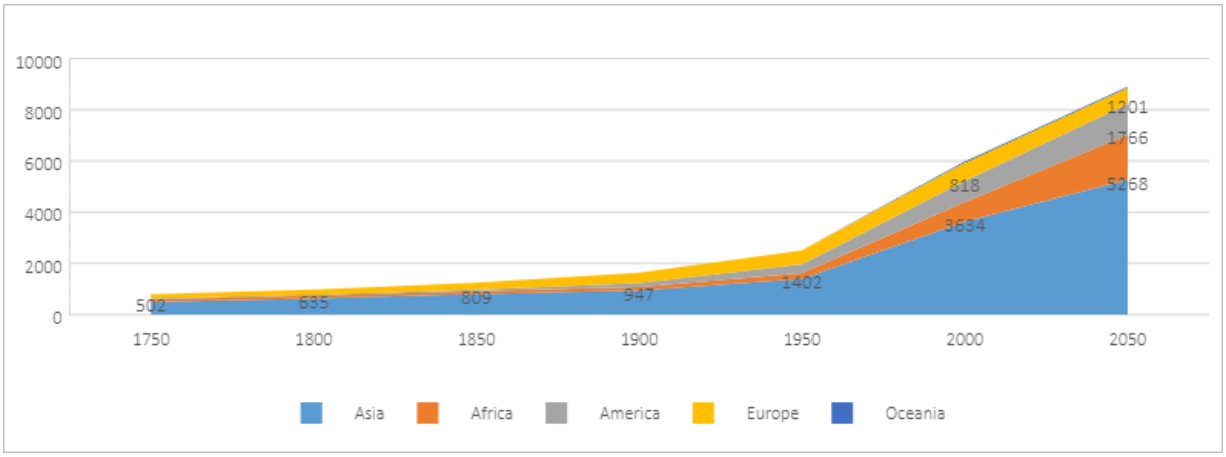
- 数据示例：

-	1750	1800	1850	1900	1950	2000	2050
Asia	502	635	809	947	1402	3634	5268
Africa	106	107	111	133	221	767	1766
America	18	31	54	156	339	818	1201
Europe	163	203	276	408	547	729	628
Oceania	2	2	2	6	13	30	46

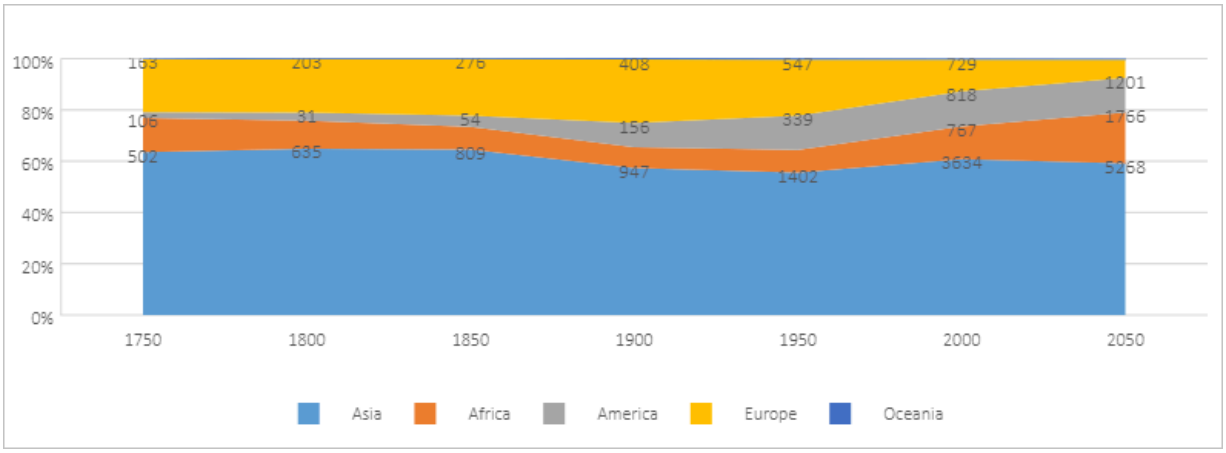
- 图形示例：



堆积面积图



百分比堆积面积图



10.4.5 条形图

本文为您介绍条形图的使用说明。

条形图用于对比各个项目的内容。在条形图里，类别通常沿着纵轴展示，数值沿着水平轴展示。条形图表达对比关系，可以按照强调的方式排列任何顺序，适用于高亮显示Top3或Top5数据。例如，在零售行业中统计畅销品的销售情况。

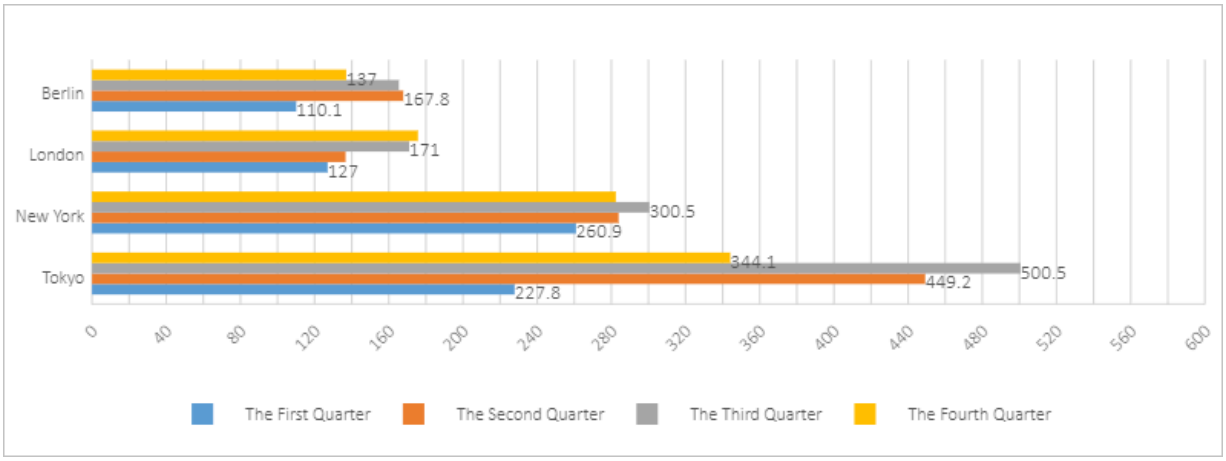
目前我们支持簇状条形图、堆积条形图和百分比堆积条形图。

数据示例：

	Tokyo	New York	London	Berlin
The First Quarter	227.8	260.9	127	110.1
The Second Quarter	449.2	283.9	136.7	167.8
The Third Quarter	500.5	300.5	171	165.4
The Fourth Quarter	344.1	282.4	175.7	137

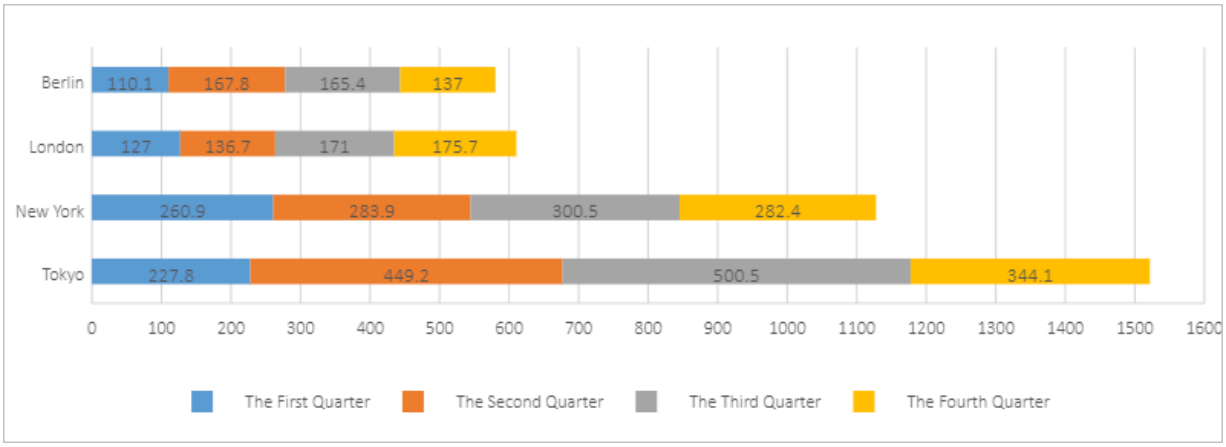
簇状条形图

图形示例：



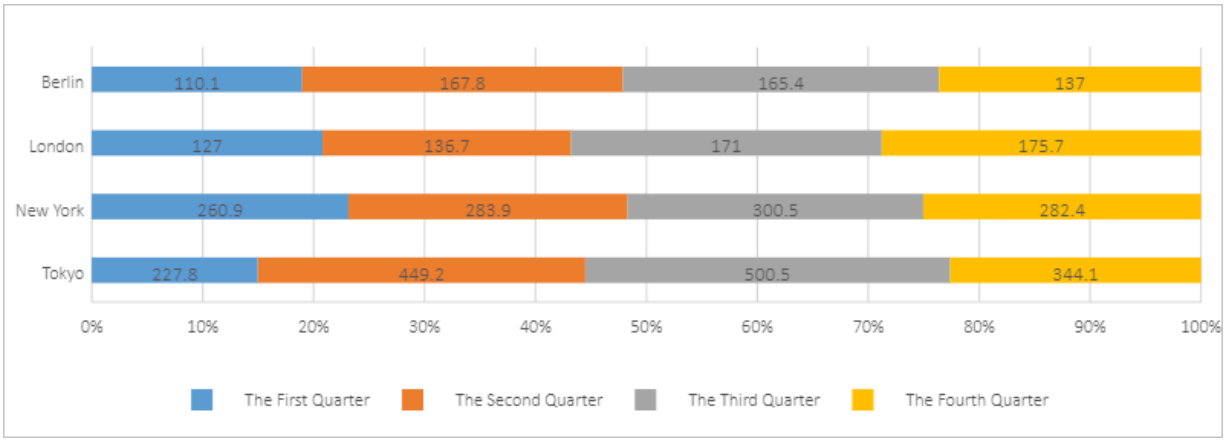
堆积条形图

图形示例：



百分比堆积条形图

图形示例：



10.4.6 散点图

本文为您介绍散点图的使用说明。

散点图通常用于寻找x与y之间是否有关联。在折线图中，x轴表示不同的类别。但在散点图中，x轴表示变量的实际值。

散点图有两个值轴，即水平(x)和垂直(y)值轴。散点图将x和y值合并为单个数据点，并以不规则的间隔或集群显示这些数据点。散点图通常用于显示和比较数值，例如科学、统计和工程数据。

通过散点图可以判断两个变量之间是否存在某种关系，并可以反映五维数据。每个点的不同颜色或标签，以及点的大小等，都可以反映一个维度，通常使用率为10%。

散点图包括以下几种类型：

- 基础散点图

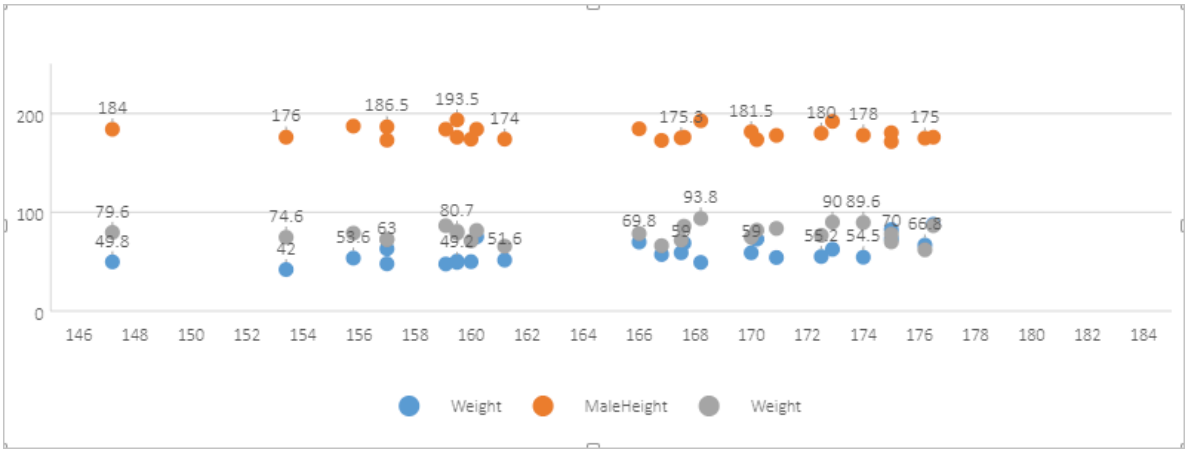
  这类图表显示了基于选定数据范围的点，通过这些点可以观察变量X和y之间是否有关系。

  数据示例：

Female		Male	
Height	Weight	Height	Weight
161.2	51.6	174	65.6
167.5	59	175.3	71.8
159.5	49.2	193.5	80.7
157	63	186.5	72.6
155.8	53.6	187.2	78.8
170	59	181.5	74.8
159.1	47.6	184	86.4
166	69.8	184.5	78.4
176.2	66.8	175	62
160.2	75.2	184	81.6
172.5	55.2	180	76.6
170.9	54.2	177.8	83.6
172.9	62.5	192	90
153.4	42	176	74.6
160	50	174	71
147.2	49.8	184	79.6

Female		Male	
Height	Weight	Height	Weight
168.2	49.2	192.7	93.8
175	73.2	171.5	70
157	47.8	173	72.4
167.6	68.8	176	85.9
159.5	50.6	176	78.8
175	82.5	180.5	77.8
166.8	57.2	172.7	66.2
176.5	87.8	176	86.4
170.2	72.8	173.5	81.8
174	54.5	178	89.6

图形示例：



- **带平滑线和数据标记的散点图**

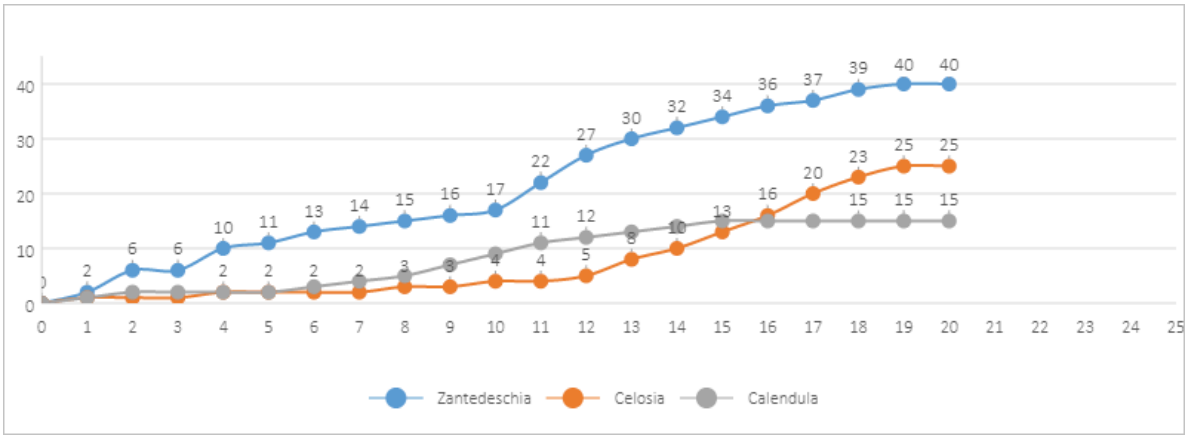
  与基础散点图相比，这类图表显示了一条平滑的曲线。

  数据示例：

Period	Zantedeschia	Celosia	Calendula
0	0	0	0
1	2	1	1
2	6	1	2
3	6	1	2
4	10	2	2

Period	Zantedeschia	Celosia	Calendula
5	11	2	2
6	13	2	3
7	14	2	4
8	15	3	5
9	16	3	7
10	17	4	9
11	22	4	11
12	27	5	12
13	30	8	13
14	32	10	14
15	34	13	15
16	36	16	15
17	37	20	15
18	39	23	15
19	40	25	15
20	40	25	15

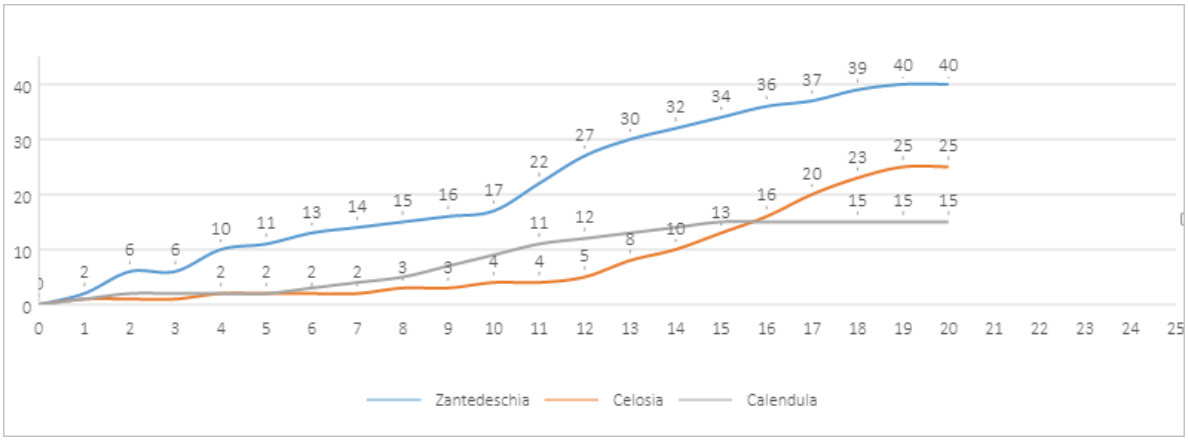
图形示例：



- **带平滑线的散点图**

  与基础散点图相比，这类图表显示了数据连接点的平滑曲线，连接了数据点并删除了所有的点。

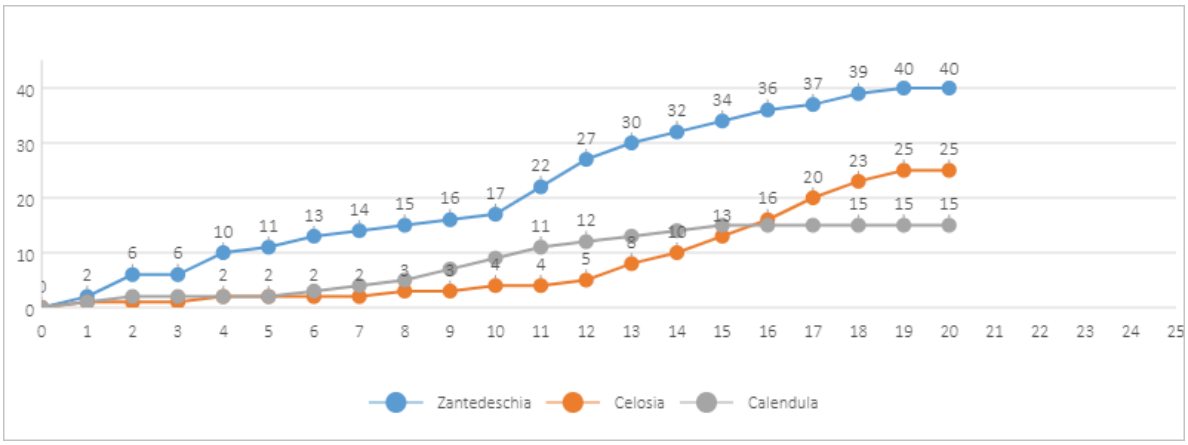
  图形示例：



- **带直线和数据标记的散点图**

  与基础散点图相比，这类图表显示了数据连接点的直线。

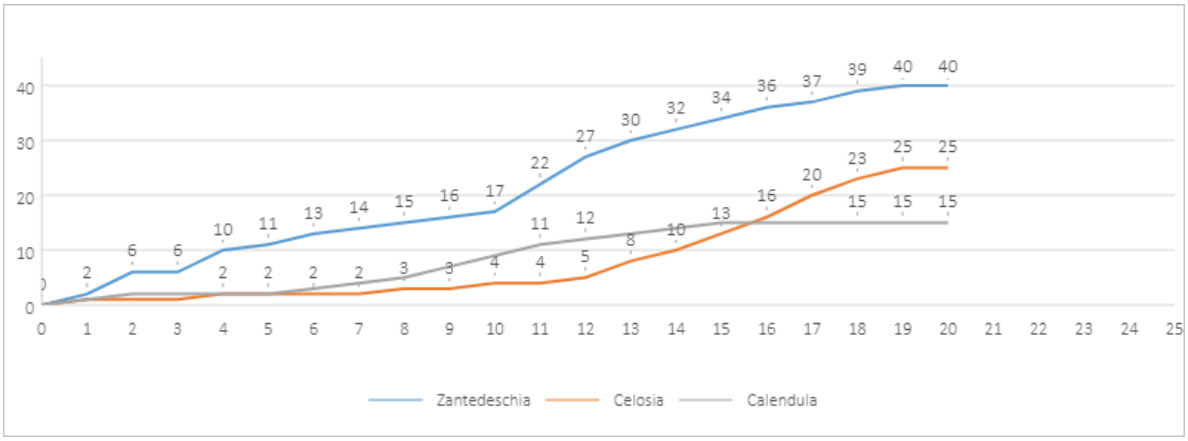
  图形示例：



- 带直线的散点图

  与基础散点图相比，这类图表显示了一条直线，连接了数据点并删除了所有的点。

  图形示例：



- 气泡图

  这是一个特殊的散点图。

  气泡图是散点图的一个变化版本。在气泡图中，数据点被气泡取代，而数据的另一个维度以气泡的大小表示。与散点图类似，气泡图也使用范围轴，即水平轴和垂直轴都是值轴。此外，x值和y值是在散点图上绘制的，一个气泡图包含了x值、y值和z(大小)值。

  如果数据包含三个数据序列，每个序列都包含一组值，那么您可以使用一个气泡图表而不是散点图。气泡的大小是由第三个数据系列中的值决定的。气泡图经常被用于展示金融数据，不同的气泡大小对于引起视觉冲击是很有用的。

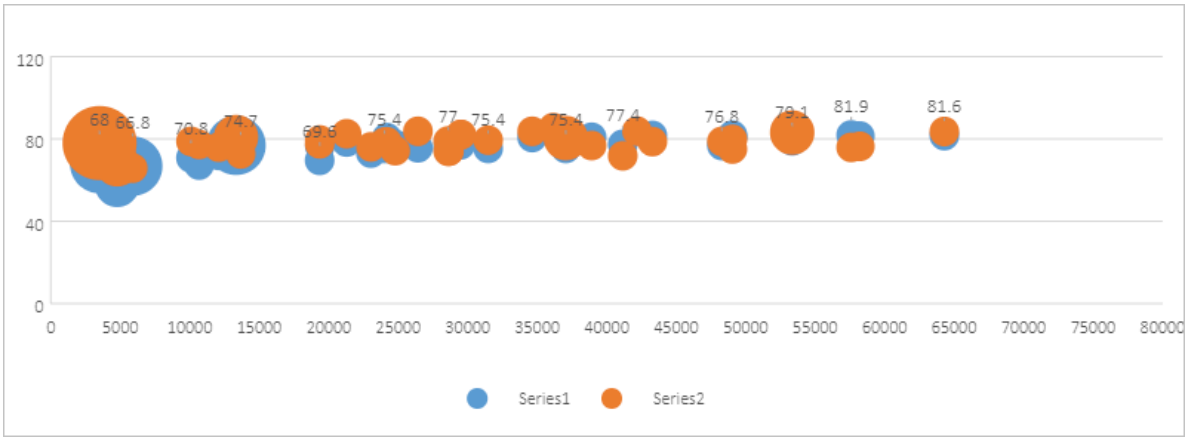
  数据示例：

28604	77	17096869	74	67096869
41163	77.4	27662440	71.8	47662440
3516	68	1154605773	78	1654605773
13670	74.7	10582082	72.7	69582082
28599	75	4986705	79	1986705
29476	77.1	56943299	82.1	26943299
31476	75.4	78958237	79.4	98958237
28666	78.1	254830	74.1	954830
4777	57.7	870601776	67.6	570601776
29550	79.1	122249285	82.1	22249285
5076	67.9	20194354	64.9	40194354

28604	77	17096869	74	67096869
12087	72	42972254	76	342972254
24021	75.4	3397534	78.4	1397534
48296	76.8	4240375	78.8	14240375
1088	70.8	38195258	78.7	18195258
19349	69.6	147568552	77.6	234568552
10670	67.3	53994605	77.3	83994605
26424	75.7	57110117	83.7	86110117
37062	75.4	252847810	80.4	652847810
49056	81.8	23968973	79.8	63968973
43294	81.7	35939927	78.7	15939927
13334	76.9	1376048943	80.9	976048943
21291	78.5	11389562	82.5	151389562
38923	80.8	5503457	76.8	1503457
57599	81.9	64395345	75.9	34395345
49053	81.1	80688545	75.1	20688545
42182	82.8	329425	83.8	1329425
5903	66.8	1311050527	65.8	311050527
36162	83.5	126573481	85.5	326573481
4390	71.4	25155317	77.4	55155317
34644	80.7	50293439	83.7	20293439
24186	80.6	4528526	78.6	13528526
64304	81.6	5210967	83.6	3210967
24787	77.3	38611794	74.3	88611794
23038	73.13	143456918	76.13	83456918
19360	76.5	78665830	79.5	58665830
58225	81.4	64715810	76.4	84715810

28604	77	17096869	74	67096869
53354	79.1	321773631	83.1	721773631

图形示例：



11 数据服务

11.1 数据服务概览

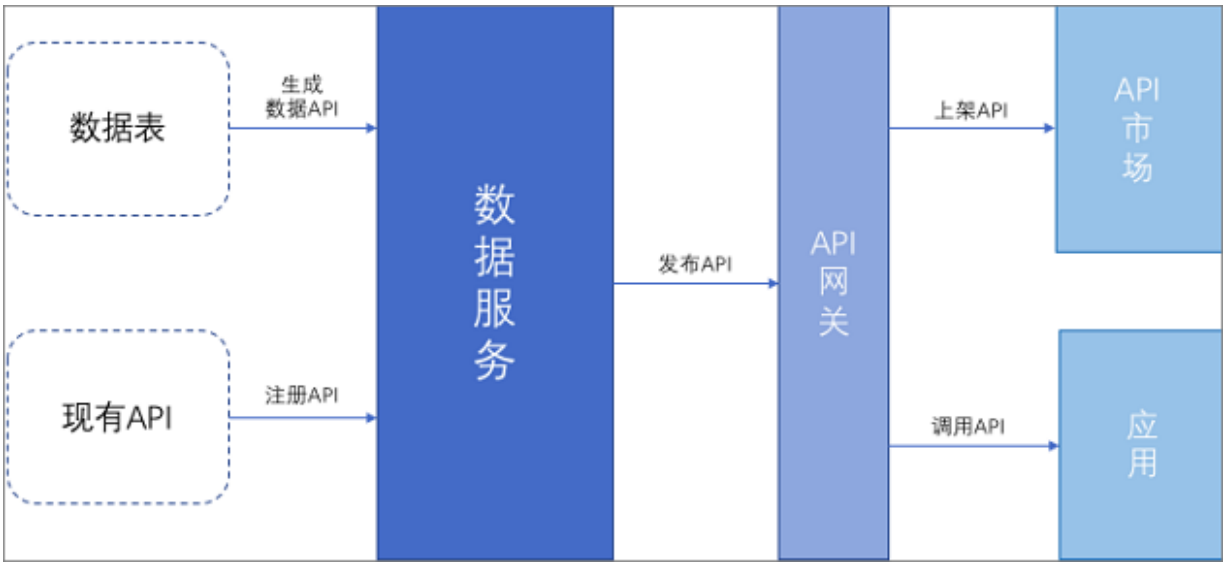
本文将从数据API生成、API注册、API网关和API市场等方面，为您介绍数据服务功能。

> 说明:
> DataWorks数据服务功能已全面开放公测，目前开放region：华东2，华北2，华东1，华南1。如使用数据服务有疑惑，请加入数据服务答疑群：21993540。
>
> 华东2（上海）：http://ds-cn-shanghai.data.aliyun.com
>
> 华东1（杭州）：http://ds-cn-hangzhou.data.aliyun.com
>
> 华北2（北京）：http://ds-cn-beijing.data.aliyun.com
>
> 华南1（深圳）：http://ds-cn-shenzhen.data.aliyun.com

DataWorks数据服务旨在为企业搭建统一的数据服务总线，帮助企业统一管理对内对外的API服务。数据服务为您提供快速将数据表生成数据API的能力，同时支持您将现有的API快速注册到数据服务平台以统一管理和发布。

数据服务已与API网关（API Gateway）打通，支持将API服务一键发布至API网关。数据服务与API网关为您提供了安全稳定、低成本、易上手的数据开放共享服务。

数据服务采用Serverless架构，您只需关注API本身的查询逻辑，无需关心运行环境等基础设施，数据服务会为您准备好计算资源，并支持弹性扩展，零运维成本。



数据API生成

数据服务目前支持将关系型数据库和NoSQL数据库的表通过可视化配置的向导模式快速生成数据API，您无需具备编码能力，即可在几分钟之内配置好一个数据API。

同时为了满足高阶用户的个性化查询需求，数据服务也提供自定义SQL的脚本模式，允许您自行编写API的查询SQL，并支持多表关联、复杂查询条件以及聚合函数等能力。

API注册

数据服务支持将您现有的API服务注册上来，与通过数据表生成的API统一管理。目前支持Restful风格的API注册，包含GET、POST、PUT和DELETE四种常见请求方式，支持表单、JSON和XML三种数据格式。

API网关

API网关（API Gateway），提供API托管服务，涵盖API发布、管理、运维、售卖的全生命周期管理。帮助您简单、快速、低成本、低风险地实现微服务聚合、前后端分离、系统集成，向合作伙伴、开发者开放功能和数据。

数据服务已与API网关产品一键打通，在数据服务中配置生成以及注册的API都可以一键发布到API网关，并通过API网关来管理API的授权鉴权、流量控制、计量等服务。

API市场

阿里云API市场是国内最为全面的综合API交易市场，涵盖了金融理财、人工智能、电子商务、交通地理、生活服务、企业管理和公共事务八大类目，目前已有数千款API产品在线售卖。

数据服务生成和注册的API，发布到API网关之后，可以一键上架到阿里云API市场售卖，帮助企业快速实现数据价值变现，最终形成商业闭环。

11.2 名词解释

本文为您介绍数据源、生成API、注册API、向导模式、脚本模式、API分组、API网关和API市场等名词。

名词	解释
数据源	即数据库的连接信息，数据服务通过数据源来访问数据，数据源需要在DataWorks中的数据集成产品中配置。
生成API	将数据表通过一系列配置生成数据API。
注册API	将您已有的API直接注册到数据服务中，以便与生成的API统一管理。
向导模式	通过Step-By-Step的可视化配置生成API，无需编写任何代码，适合简单的API以及初学者。

名词	解释
脚本模式	通过自定义SQL脚本的方式配置生成API，支持多表关联、复杂查询条件、聚合函数等功能，适合复杂的API及有编程经验的开发者。
API分组	一个特定功能或场景的API集合，是数据服务中API的最小组织单元，也是API网关中的最小管理单元，应对于阿里云API市场的一个API商品。
API网关	阿里云提供的一个API托管的应用服务，提供了API全生命周期的管理、权限控制、访问控制、流量控制等功能。
API市场	阿里云API市场是搭建在云市场体系之内，是国内最全面的综合API交易平台。

11.3 生成API

11.3.1 配置数据源

数据服务可通过数据源获取数据表的Schema信息以及执行数据API的查询请求。

说明:

使用数据API生成服务前，您需要提前配置好数据源。

您可进入数据集成 > 数据源页面配置数据源，不同数据源类型的支持情况及配置方式如下表所示。

数据源名称	向导模式生成数据API	脚本模式生成数据API	配置方法
RDS	支持	支持	RDS包括MySQL、PostgreSQL和SQL Server
DRDS	支持	支持	#unique_56
MySQL	支持	支持	#unique_43
PostgreSQL	支持	支持	#unique_49
SQL Server	支持	支持	#unique_46
Oracle	支持	支持	#unique_52
AnalyticDB（ADS）	支持	支持	#unique_75
Table Store（OTS）	支持	不支持	#unique_96
MongoDB	支持	不支持	#unique_89

说明:

- 数据服务生成API当前不支持MaxCompute（ODPS）类型数据源，您需要通过Lightning的方式实现，且Lightning中需要配置为PostgreSQL类型。
- 由于数据服务通过Lightning查询MaxCompute，Lightning兼容PostgreSQL，而PostgreSQL中没有Datetime数据类型，所以如果您的数据有Datetime类型，数据服务中会映射为PostgreSQL的Timestamp类型进行查询。

11.3.2 生成API功能概览

本文将为您介绍通过向导模式或脚本模式生成API功能的区别。

数据服务目前支持将关系型数据库和NoSQL数据库的表通过可视化配置的向导模式快速生成数据API，您无需具备编码能力，即可在几分钟之内配置好一个数据API。

同时为了满足高阶用户的个性化查询需求，数据服务也提供了自定义SQL的脚本模式，允许您自行编写API的查询SQL，并支持多表关联、复杂查询条件以及聚合函数等能力。

向导模式与脚本模式的功能区别详见下表。

功能分类	功能点	向导模式	脚本模式
查询对象	单数据源单数据表查询	支持	支持
	单数据源多数据表关联查询	不支持	支持
查询条件	数值型等值查询	支持	支持
	数值型范围查询	不支持	支持
	字符型精确匹配	支持	支持
	字符型模糊匹配	支持	支持
	必选参数与可选参数	支持	支持
查询结果	字段值原样返回	支持	支持
	字段值进行数学运算	不支持	支持
	字段值进行聚合函数运算	不支持	支持
	返回结果分页	支持	支持

11.3.3 向导模式生成API

本文将为您介绍如何通过向导模式生成API。

使用向导模式生成数据API简单易学，您不需编写任何代码，通过产品界面进行勾选配置即可快速生成API。推荐对API功能的要求不高或者无代码开发经验的用户使用。

说明:

配置API前，请首先在数据集成 > 数据源页面配置好数据源。

配置API基础信息

1. 进入API服务列表 > 生成API页面，单击向导模式。

2. 单击向导模式，填写API基础信息。

配置	说明
API名称	支持中文、英文、数字、下划线，且只能以英文或中文开头，4-50个字符。
API分组	API分组是指针对某一个功能或场景的API集合，也是API网关对API的最小管理单元。在阿里云API市场中，一个API分组对应于一个API商品。您可单击新建API分组进行新建。

配置	说明
API Path	API存放的路径，如/user。
协议	目前，生成API仅支持HTTP协议。
请求方式	目前，生成API仅支持GET请求方式。
返回类型	目前，生成API仅支持JSON返回类型。
描述	对API进行简要描述。

 说明:

API分组的设置示例如下：

例如您要配置一个天气查询的API产品，天气查询API由城市名称查询天气API、景点名称查询天气API和邮编查询天气API三种API组成，那么就可以创建一个名为天气查询的API分组，并把以上三种API放在这个分组中。然后把这个API上架到API市场中销售时，就会呈现为一个天气查询的API产品。

当然，如果您生成的API在自己的APP中使用，则可以把分组当作分类来使用。

3. 填写好API基础信息后，单击确认，即可进入API参数配置页面。

配置API参数

1. 进入数据源类型 > 数据源名称 > 数据表页面，选择需要配置的表。

 说明:

- 您需提前在数据集成中配置好数据源，数据表下拉框支持表名搜索。
- 创建好API后，会自动跳转至数据表配置页面，您可以直接进行配置。

2. 配置参数字段。

选择好数据表之后，左侧会自动列出这个表的所有字段，分别勾选需要设置为请求参数和返回参数的字段，分别添加到请求参数和返回参数列表当中。

3. 编辑请求参数信息。

   单击右侧的请求参数，设置参数的名称、参数类型、操作符、是否必填、示例值、默认值和描述。

4. 编辑返回参数信息。

单击右侧的返回参数，设置参数的名称、参数类型示例值和描述，并可进行返回结果分页和使用过滤器等高级配置。

配置过程中需要注意返回结果分页的设置。

- 如果不开启返回结果分页，则API默认最多返回500条记录。
- 如果返回结果可能超过500条，请开启返回结果分页功能。

开启返回结果分页后，会自动增加以下公共参数。

- 公共请求参数
  - pageNum：当前页号。
  - pageSize：页面大小，即每页记录数。
- 公共返回参数
  - pageNum：当前页号。
  - pageSize：页面大小，即每页记录数。
  - totalNum：总记录数。

说明：

- 请求参数仅支持等值查询，返回参数仅支持字段值原样输出。
- 尽量将有索引的字段设置为请求参数。
- API允许不设置请求参数，当无请求参数时，则必须开启返回结果分页。
- 为方便API调用者了解API详情，请尽量设置API参数的示例值、默认值、描述等信息。
- 单击已配置的API，可以查看当前的表已经生成的API列表，请避免生成同样的API。

API测试

完成API参数的配置并保存后，单击右上角的测试，即可进入API测试环节。

填写参数值，单击开始测试，即可在线发送API请求，在右侧可以看到API请求详情及返回内容。如果测试失败，请仔细查看错误提示并做相应的修改重新测试。

配置过程中需要注意正常返回示例的设置。配置好API之后，系统会自动生成异常返回示例和错误码，但没办法自动生成正常返回示例。需要在测试成功后，单击保存为正常返回示例，将当前的测试结果保存为正常返回示例。如果返回结果中有敏感数据需要脱敏，可以手动编辑修改。

说明:

- 正常返回示例对于API的调用者来说，具有非常重要的参考意义，请务必配置。
- API调用延迟是本次API请求的延迟，供您评估的API性能。如果延迟较大，则要考虑进行数据库优化。

完成API测试之后，单击完成，即成功生成了一个数据API。

API详情查看

回到API服务列表页面，右键单击相应API服务，选择详情，即可查看API的详情信息。API详情页面以调用者的视角展示了API的详细信息。

11.3.4 脚本模式生成API

本文将为您介绍如何通过脚本模式生成API。

为了满足高阶用户的个性化查询需求，数据服务也提供了自定义SQL的脚本模式，允许您自行编写API的查询SQL，并支持多表关联、复杂查询条件以及聚合函数等能力。

配置API基础信息

1. 进入API服务列表 > 生成API页面，单击脚本模式。

2. 单击脚本模式，填写API基础信息。

配置	说明
API名称	支持中文、英文、数字、下划线，且只能以英文或中文开头，4-50个字符。
API分组	API分组是指针对某一个功能或场景的API集合，也是API网关对API的最小管理单元。在阿里云API市场中，一个API分组对应于一个API商品。您可单击新建API分组进行新建。

配置	说明
API Path	API存放的路径，如/user。
协议	目前，生成API仅支持HTTP协议。
请求方式	目前，生成API仅支持GET请求方式。
返回类型	目前，生成API仅支持JSON返回类型。
描述	对API进行简要描述。

 说明:

API分组的设置示例如下：

例如您要配置一个天气查询的API产品，天气查询API由城市名称查询天气API、景点名称查询天气API和邮编查询天气API三种API组成，那么就可以创建一个名为天气查询的API分组，并把以上三种API放在这个分组中。然后把这个API上架到API市场中销售时，就会呈现为一个天气查询的API产品。

当然，如果您生成的API是在自己的APP中使用的，那可以把分组当作分类来使用。

3. 填写好API基础信息后，单击确认，即可进入API参数配置页面。

配置API查询SQL及参数

1. 选择数据源和表。

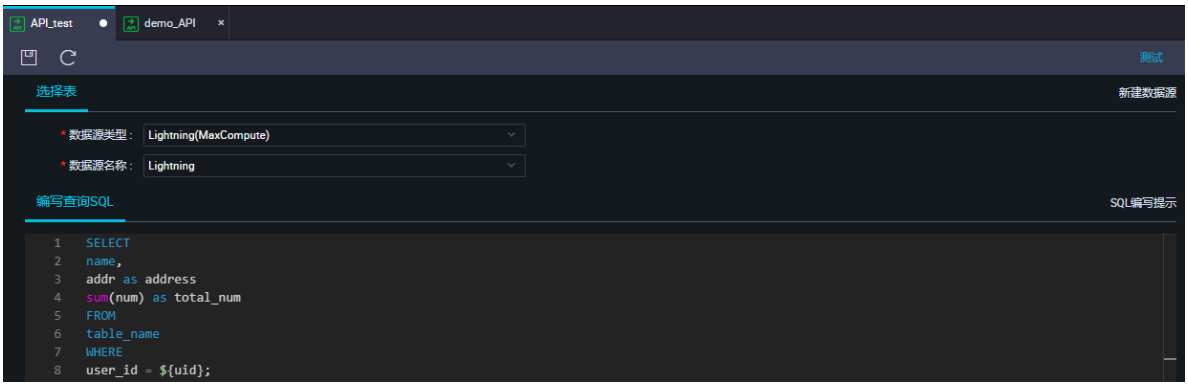
   进入数据源类型 > 数据源名称 > 数据表页面，单击数据表列表中相应的表名，可查看该表的字段信息。

 说明:

- 您需要提前在数据集成中配置好数据源。
- 必须先选择一个数据源，并且只支持同一个数据源的多表关联查询，不支持跨数据源的关联查询。

2. 编写API查询SQL。

   在代码编辑区中输入SQL代码。



> **说明：**
> SELECT查询的字段为API的返回参数，WHERE条件处的参数为API的请求参数，请求参数请使用${}标识。

3. 编辑请求参数信息。

   编写好API查询SQL后，单击右侧的请求参数，设置参数的名称、参数类型、示例值、默认值和描述。

> **说明：**
> 为了帮助API的调用者更全面地了解API，请尽量全面地填写API的参数信息。

4. 编辑返回参数信息。

单击右侧的返回参数，设置参数的名称、参数类型示例值和描述，并可进行返回结果分页和使用过滤器等高级配置。

配置过程中需要注意返回结果分页的设置。

- 如果不开启返回结果分页，则API默认最多返回500条记录。
- 如果返回结果可能超过500条，请开启返回结果分页功能。

开启返回结果分页后，会自动增加以下公共参数。

- 公共请求参数
  - pageNum：当前页号。
  - pageSize：页面大小，即每页记录数。
- 公共返回参数
  - pageNum：当前页号。
  - pageSize：页面大小，即每页记录数。
  - totalNum：总记录数。

说明:

SQL规则提示。

- 仅支持一条SQL语句，不支持多条SQL语句。
- 支持SELECT，不支持非SELECT语法，如INSERT、UPDATE、DELETE等。
- SELECT的查询字段为API的返回参数，WHERE条件中的${param}内的变量param为API的请求参数。
- 不支持`SELECT \*`，必须明确指定查询的列。
- 支持同一数据源下的单表查询，多表关联查询，嵌套查询。
- 如果SELECT查询列的列名带有表名前缀（如t.name），则必须取别名作为返回参数名（如t.name as name）。

- 如果使用聚合函数（min/max/sum/count等），必须取别名作为返回参数名（如sum(num) as total_num）。
- SQL中的${param}统一当请求参数进行替换，包含字符串中的${param}。当${param}前包含转义符\时，不做请求参数处理，作为普通字符串处理。
- 不支持将${param}放在引号中，如'${id}'、'abc${xyz}123'，如果需要可以通过concat('abc', ${xyz}, '123’)实现。
- 不支持将参数设置为可选。

API测试

完成API参数的配置并保存后，单击右上角的测试，即可进入API测试环节。

填写好参数值，单击开始测试，即可在线发送API请求，在右侧可以看到API请求详情及返回内容。如果测试失败，请仔细查看错误提示并做相应的修改重新测试。

配置过程中需要注意正常返回示例的设置。配置好API之后，系统会自动生成异常返回示例和错误码，但没办法自动生成正常返回示例。需要在测试成功后，单击保存为正常返回示例，将当前的测试结果保存为正常返回示例。如果返回结果中有敏感数据需要脱敏，可以手动编辑修改。

说明:

- 正常返回示例对于API的调用者来说，具有非常重要的参考意义，请务必配置。
- API调用延迟是本次API请求的延迟，供您评估的API性能。如果延迟较大，则要考虑进行数据库优化。

完成API测试之后，单击完成，即成功生成了一个数据API。

API详情查看

回到API服务列表页面，右键单击相应API服务，选择详情，即可查看API的详情信息。API详情页面以调用者的视角展示了API的详细信息。

11.3.5 使用过滤器

本文为您介绍如何使用过滤器，对API生成结果进行深入处理。

什么是过滤器

数据服务过滤器是用于对API生成结果进一步处理的动态函数：通过指定一个或者多个过滤器，您可以实现自定义API返回结构，对API的数据进行进一步的加工。

- 过滤器目前只支持Python3的语法。
- 过滤器目前只支持 `import json,time,random,pickle,re,math`
- 过滤器限定函数名 `def filter(context,data):`

函数结构说明

系统自带的模板函数如下。

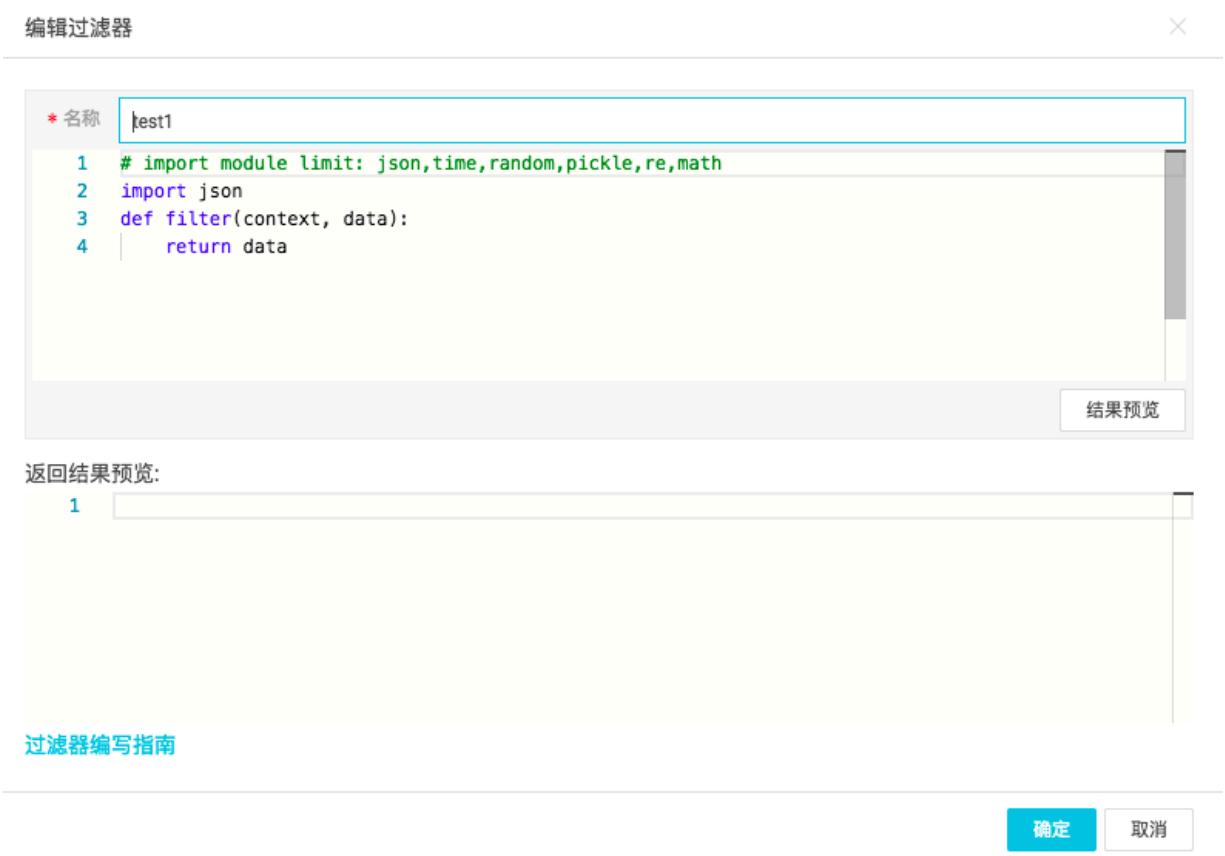
```
# import module limit: json,time,random,pickle,re,math
import json
def filter(context, data):
    return data
```

您可以基于该函数进行修改，函数的入参名称不是强制要求，可以按自己需要修改名称入参，如下所示。

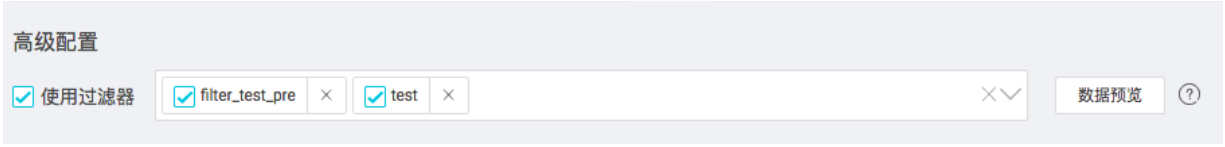
```
参数1[context]：字符串类型，包含API执行的上下文环境，目前为空，暂未启用
```

参数2[data]：字符串类型，包含API执行的结果，或者是上一个过滤器处理后的结果

使用说明



- 名称：过滤器名称，不超过64字符。
- 数据预览：在编写函数时可以随时进行调试，通过数据预览可以实时将API SQL执行结果同正在开发的函数结合起来。
- 返回结果预览：当前函数处理后的结果数据。



- 过滤器支持多级编排，可以随时调整顺序，用于进行细粒度拆分。
- 可以点击数据预览来调试你编排的过滤器的结果。
- 可以随时通过勾选使用过滤器来确定需要执行的函数。

11.4 注册API

本文将为您介绍如何注册API，并与通过数据表生成的API统一管理和发布到API网关。

目前数据服务支持Restful风格的API注册，包含GET、POST、PUT、DELETE四类常见请求方式，支持表单、JSON、XML三种数据格式。

配置API基础信息

1. 进入API服务列表 > 注册API页面。

2. 配置API基础信息。

配置	说明
API名称	支持中文、英文、数字、下划线，且只能以英文或中文开头，4-50个字符。
API分组	API分组是指针对某一个功能或场景的API集合，也是API网关对API的最小管理单元。在阿里云API市场中，一个API分组对应于一个API商品。您可单击新建API分组进行新建。
API Path	后台服务Path的别名，为了支持相同的后台服务Host和Path的API注册为多个API。 如果后台服务Path中定义了参数，则API Path中需要定义同样的参数，参数也放在[]中。
协议	目前仅支持HTTP协议。
请求方式	支持GET、POST、PUT和DELETE，不同的请求方式后续的配置项会略有不同。
返回类型	目前支持JSON和XML返回类型。
描述	对API进行简要描述。

3. 填写好API基础信息后，单击确认，即可进入API参数配置页面。

配置API参数

配置API基础信息后即可配置API参数。这里将配置API的后端服务定义、请求参数定义、返回内容定义和错误码定义。

配置	说明
后端服务定义	· 后台服务Host：待注册API服务的Host，以`http://`或`https://`开头，并且不包含Path。 · 后台服务Path：待注册API服务的Path，Path中支持参数，参数要放在`[]`中，如`/user/[userid]`。 定义了Path中的参数后，在注册API向导的第二步API参数配置环节，系统会自动在请求参数列表添加Path位置的参数。 · 后端超时：设置后端超时时间。

配置	说明
请求参数定义	· 参数位置：请求参数位置支持Path、Header、Query和Body，不同的请求方式所支持的可选参数位置不一样，请根据产品上提供的可选项按需选择。 · 常量参数：常量参数即参数值是固定的参数，对调用者不可见，API调用时不需传入常量参数，但后台服务始终接收这里定义好的常量参数及参数值。适用于当您希望把API的某个参数的取值固定为某个值以及要对调用者隐藏参数的场景。 · 请求Body定义：请求Body定义只在请求方式为POST和PUT时出现。请求Body定义支持输入Body内容描述，即相当于一个请求Body的示例，以供API调用者参考格式。请求Body的内容类型（Content-Type）支持JSON和XML两种。 说明: 当定义了请求Body，如果您同时在请求参数定义中定义了Body位置的参数，那么Body位置的参数就无效了，系统会以请求Body为准。
返回内容定义	支持填写正常返回示例和异常返回示例，以供API调用者参考和编写API返回结果解析代码。
错误码定义	这里填写API调用时的错误信息及解决方案，以帮助API调用者在遇到错误时能够自行查找错误原因并解决。 说明: 为了让API更容易被调用者使用，请尽可能完整的填写API的参数信息，尤其是参数的示例值、默认值以及返回示例等。

API测试

完成API查询SQL及参数的配置后，即可进行API测试。

填写好参数值，单击开始测试，即可在线发送API请求，在右侧可以看到API请求详情及返回内容。如果测试失败，请仔细查看错误提示并做相应的修改重新测试。

配置过程中需要注意正常返回示例的设置。配置好API之后，系统会自动生成异常返回示例和错误码，但没办法自动生成正常返回示例。需要在测试成功后，单击保存为正常返回示例，将当前的测试结果保存为正常返回示例。如果返回结果中有敏感数据需要脱敏，可以手动编辑修改。

说明:

- 正常返回示例对于API的调用者来说，具有非常重要的参考意义，请务必配置。
- API调用延迟是本次API请求的延迟，供您评估的API性能。如果延迟较大，则要考虑进行数据库优化。

完成API测试之后，单击完成，即成功生成了一个数据API。

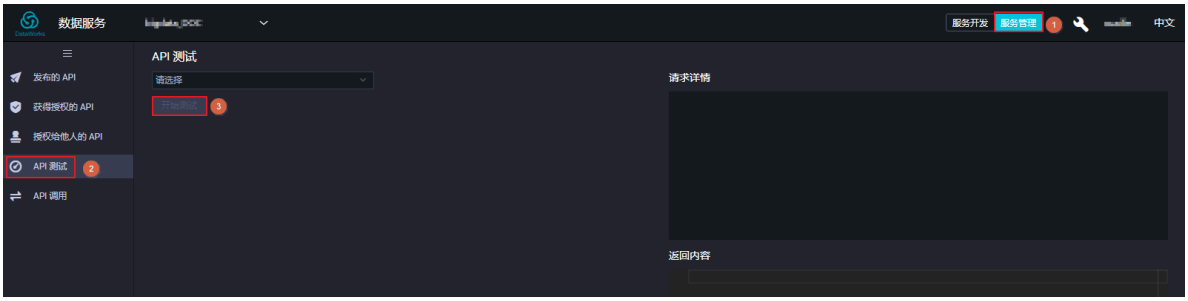
11.5 测试API

本文将为您介绍如何进行API服务测试。

在生成API和注册API时，系统提供了API测试的功能，详情请参见#unique_550。

同时系统也提供了一个独立的API服务测试功能，以方便您日常在线测试API。

1. 单击页面上方的服务管理。
2. 单击左侧导航栏中的API测试。
3. 选择需要测试的API，填写参数值，单击开始测试。



说明:

API服务测试页面仅提供API在线测试功能，不提供API正常返回示例的更新保存功能。如果您需要更新API的正常返回示例，请在API列表页中单击编辑进入API的编辑模式，再在API测试环节更新正常返回示例的内容。

11.6 发布API

本文将为您介绍如何将数据服务中的API发布到API网关，并上架到API市场。

API 网关（API Gateway）提供API托管服务，涵盖API发布、管理、运维、售卖的全生命周期管理。帮助您简单、快速、低成本、低风险地实现微服务聚合、前后端分离、系统集成，向合作伙伴、开发者开放功能和数据。

API网关是API对外开放或者在自己的应用中调用的最后一道防线，提供了权限管理、流量控制、访问控制、计量等服务，因此在数据服务中生成的API以及注册的API，为了安全起见，一般来说都需要发布到API网关中才能对外提供服务。数据服务与API网关产品一键打通，支持将API一键发布到API网关。

数据服务中的API发布到API网关

说明:

您要想发布API，首先必须开通API网关服务。

开通API网关之后，单击API服务列表操作列中的发布，即可将API一键发布到API网关中。发布过程中，系统会自动将API注册到API网关中。系统会以API分组的名称在API网关中创建一个同名的分组，并将此API发布到这个分组下面。

发布完成后，您可以进入到API网关控制台，查看API信息，也可以进一步在API网关设置流量控制、访问控制等功能。

如果您的API是为了供自己的应用程序调用，那么需要在API网关中创建应用，将API授权到应用中，然后通过AppKey和AppSecret加密签名调用，详情请参见调用API。同时API网关也提供了主流编程语言的SDK，您可快速将API集成到自己的应用中，详情请参见SDK下载及使用指南。

将API上架到阿里云API市场

阿里云API市场是国内最为全面的综合API交易市场，涵盖了金融理财、人工智能、电子商务、交通地理、生活服务、企业管理和公共事务八大类目，目前已有数千款API产品在线售卖，是快速帮您实现数据变现的平台。

数据服务生成和注册的API，在发布到API网关之后，可以一键上架到阿里云API市场售卖，帮助企业快速实现数据价值变现，最终形成商业闭环。

在将API上架到阿里云API市场中销售之前，首先要以服务商的身份入驻阿里云云市场，流程详见服务商入驻引导。

说明:

入驻的时候，选择加入API市场。必须是企业才可以入驻阿里云API市场。

1. **进入阿里云服务商平台。**
2. **单击商品管理 > 发布商品，选择接入类型为API服务。**
3. **选择要上架的API分组（一个分组对应一个API商品）。**

4. 配置商品信息，并提交审核。

API商品通过上架审核后，即可在阿里云API市场中看见此API商品，国内外的用户都可以购买您的API服务产品。

11.7 删除API

本文将为您介绍如何删除API。

进入API服务列表页面，选择操作栏下的更多 > 删除，即可删除一个API。

说明:

- API只有是非上线状态才允许删除，当API是上线状态时，请先下线后再执行删除操作。
- 删除操作不可逆，为了您的数据，请谨慎进行删除操作。

11.8 调用API

本文将为您介绍API发布到API网关后，如何进行调用。

API网关提供了API授权、API调用SDK。您可以将API授权给自己或企业内的人员使用，也可将API授权给第三方使用。如果您想调用API，需要进行下述操作。

调用API的三要素

您要调用API，需要以下三个基础条件。

- API：您即将要调用的API，明确API参数定义。
- 应用APP：作为您调用API时的身份，有AppKey和AppSecret用于验证您的身份。
- API和APP的权限关系：APP想调用某个API，需要具有该API的权限，这个权限通过授权的功能来建立。

操作步骤

1. 获取API文档。

   根据您获取API的渠道不同，获取方式略有差异。一般分为从数据市场购买的API服务和不需购买，由提供方主动授权两种方式。详情请参见获取API文档。

2. 创建应用。

   应用APP是您调用API服务时的身份。每个APP有一组Key和Secret，您可以理解为账号密码。详情请参见创建应用。

3. 获取权限。

   授权是指授予APP调用某个API的权限。您的APP需要获取API的授权才能调用该API。

   由于获取API的渠道不同，建立授权的方式也不同。详情请参见获取授权。

4. 调用API。

   您可以直接用API网关控制台为您提供的多语言调用示例来测试调用，可以自行编辑HTTP（S）请求来调用API。详情请参见调用API。

11.9 工作流程

数据服务的工作流程提供了拖拽式可视化工作流编排能力，您可以将多个API及函数服务按照业务逻辑以串行、并行、分支等结构编排成工作流。工作流程整体可以形成一个复合API服务。工作流程又称服务编排功能，本文为您介绍工作流程的基本特性和用方法。

当您调用工作流程API服务时，系统将根据设定依次执行各个服务节点、传递服务节点参数并自动管理每个服务节点的状态转换。工作流程（服务编排）功能极大简化了多个服务之间组合调用的开发和运维成本，让您可以专注于业务本身。

说明:

当前工作流程功能仅华东2区域支持，且需要您开通DataWorks企业版，关于企业版详情请参见#unique_555。

产品优势

- 降低API服务开发成本

  通过拖拽式、可视化的方式进行工作流程编排，无需额外编写代码即可完成多个API服务的串行、并行和分支调用，大大降低了API服务的开发成本。

- 提升服务调用性能

  多个API或函数服务的调用在同一个容器实例内完成，相比您自行编写和搭建工作流服务可有效降低服务调用的网络开销，显著提升服务调用性能。

- 使用Serverless架构

  服务务编排采用Serverless架构。Serverless架构能够实现动态伸缩，您无需关注底层运行环境，只需关注业务逻辑本身。

输入与输出规则

数据服务参数取值规则基于JSONPath。JSONPath是一种信息抽取类库，用于JSON文件中抽取指定信息，数据服务参数可完全参考JSONPath语法。

例如，对于A>B>C这3个顺序节点，节点C需要取节点A、B的输出值：

- A节点输出：`{"namea":"valuea"}`。

  取A节点输出：`${A.namea}`。

- B节点输出：`{"nameb":"valueb"}`。

  取B节点输出：`$.nameb`或`${B.nameb}`。

系统内置开始节点作为整个工作的流程入参。例如工作流程的入参为{"namewf":"valuewf"}，则任意一个节点可通过${START.namewf}获取对应入参值。

参数说明

节点请求参数：

- 如果您不设置参数值，系统会默认匹配上一个节点的输出结果JSON的第一层的同名字段值，即同名映射。第一个节点则与工作流请求参数进行同名映射。
- 如果您设置了参数值，系统会使用您设置的值。
- 如果您要设置为上游指定节点的指定参数值，则需要使用JSONPath表达式获取参数。

常用获取参数的JSONPath表达式：

- $.：获取上一个节点的输出。
- $.param：获取上一个节点的输出中的param参数值。为方便获取上游任意节点的参数值，数据服务扩展了JSONPath表达式。
- ${NODEID1}：获取ID为NODEID1的节点的输出。
- ${START}：获取工作流的请求参数，即开始节点的输出。
- ${NODEID1.param}：获取NODEID1节点输出中的param参数值。

节点设置输出结果：

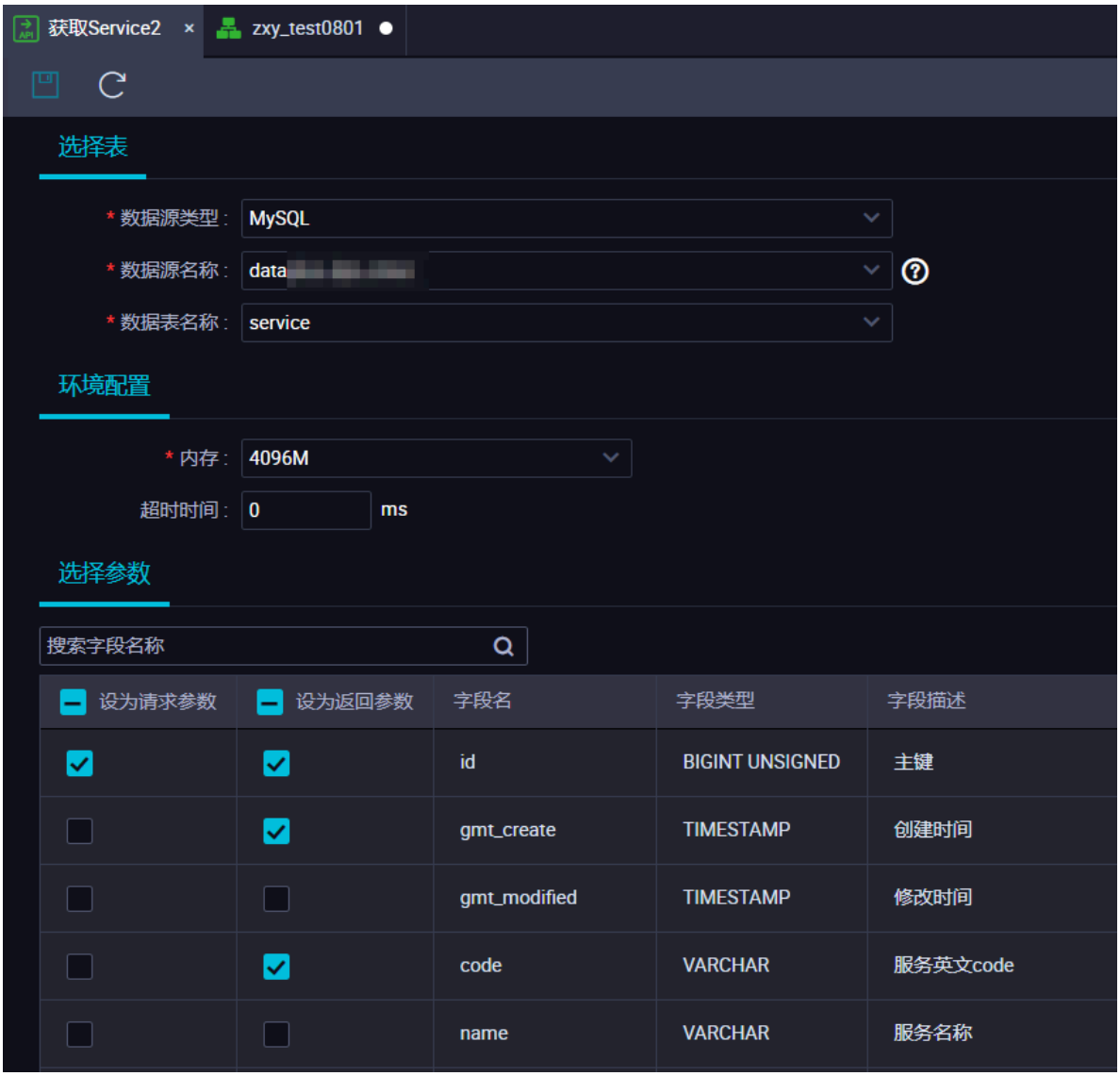
- $.：当前节点的输出结果。
- $.param：当前节点的输出中的param参数值。
- ${NODEID1.param}：获取NODEID1节点输出中的param参数值。

使用示例

请您首先完成数据源的添加，参见配置数据源。undefined本例使用的是MySQL类型数据源。

1. 注册API。

   本例中使用注册API的方式生成新的API，关于注册API的使用方法请参见注册API。



2. 注册函数。

本例中新建一个Python函数用于分支节点之后的结果处理。

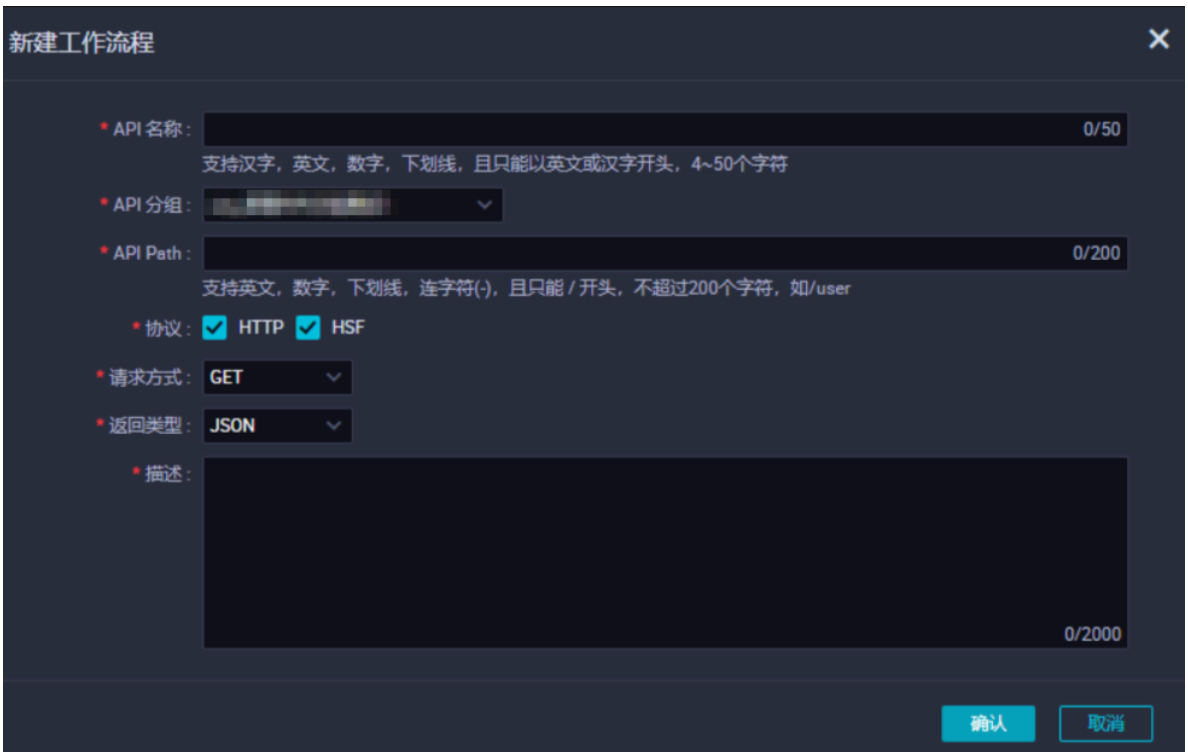
函数代码如下。

```
# -*- coding: utf-8 -*-
# event (str) : in filter it is the API result, in other cases, it
is your param
# context : some environment information, temporarily useless
# import module limit: json,time,random,pickle,re,math
import json
def handler(event,context):
    # load str to json object
    obj = json.loads(event)
    # add your code here
    # end add
    return obj
```

3. 在服务开发页面新建工作流程。

4. 填写工作流程各项参数如下：

   - 协议：HTTP、HSF。
   - 请求方式：支持GET、POST。
   - 返回类型：JSON。



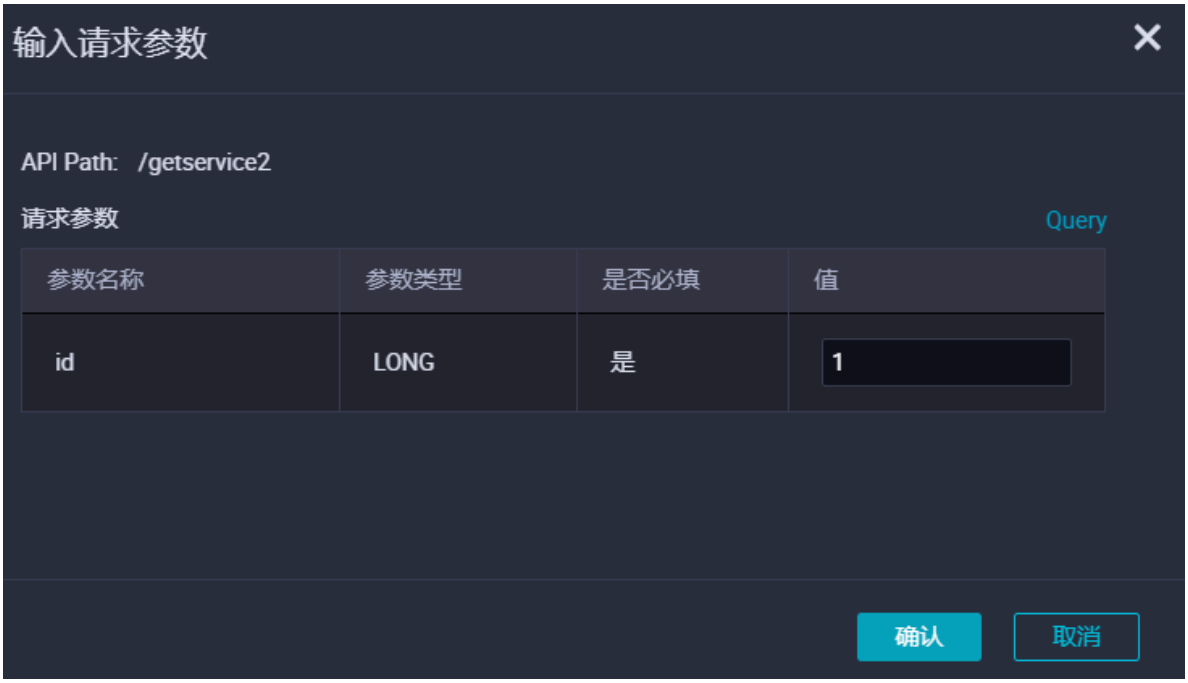
5. 完成工作流程新建后，按照如图所示顺序拖拽对应模块并连线。

6. 通过双击API1对节点进行编辑，选择API为您刚注册的API。

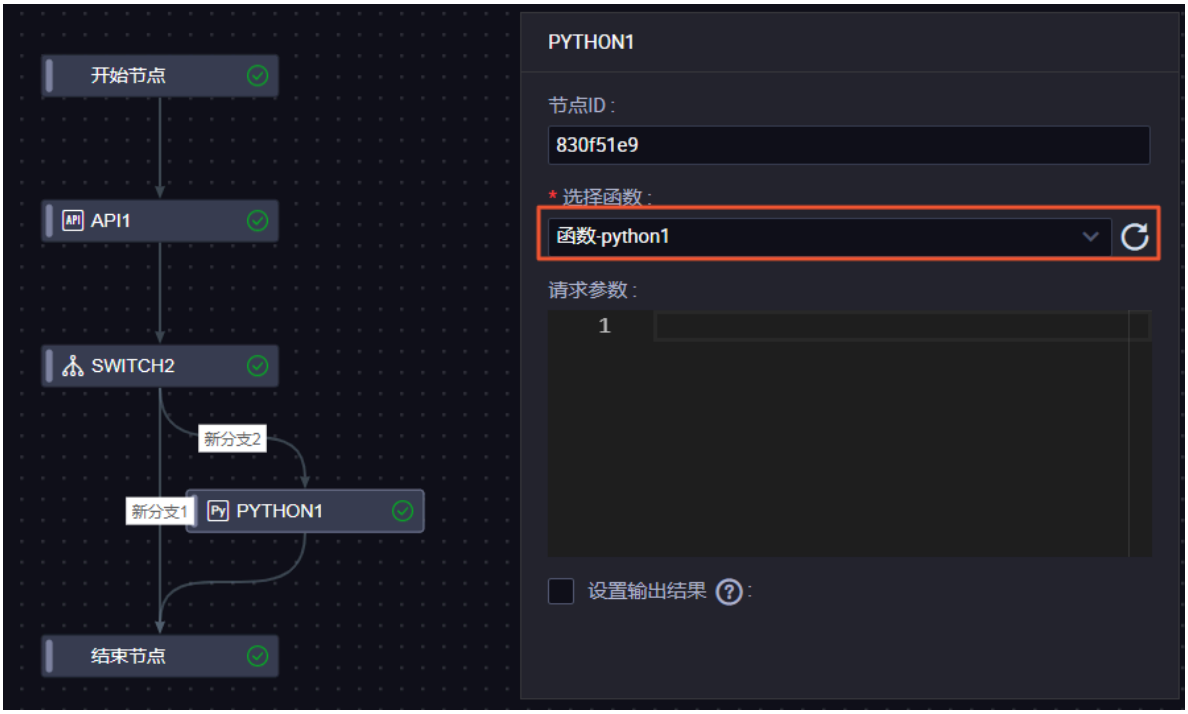
   设置输出结果为`{"user_id":"$.data[0].id"}`。

   输出结果设置规则：使用JSONPath来进行处理，其中获取节点`${NodeA.namea}`语法同入参规则一致。`{"user_id":"$.data[0].id"}`表示把当前节点的处理结果中的`data`数组的第一个值的`id`赋值给`user_id`。然后，输出`{"user_id":"value"}`形式的JSON数据。

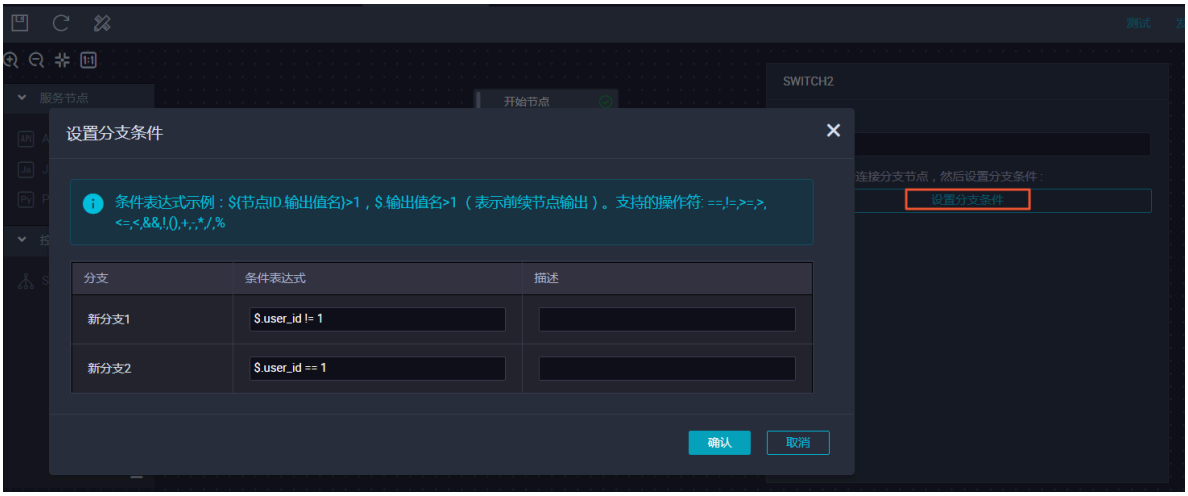
   设置输入请求参数如图所示。



7. 通过双击PYTHON1对节点进行编辑，选择函数为您刚注册的函数名称。



8. 通过双击SWITCH2对节点进行编辑，您需要设置分支条件。条件表达式考察上一个节点的输出，示例：`${节点ID.输出值名}>1`或`$.输出值名>1`。条件表达式支持的操作符包含：==、!=、>=、,><=、<、&&,、!、()、+、-、*、/、%。



本例中，`user_id`是上游输出结果，作为下游的请求参数。

```
新分支1：$.user_id != 1    分支1中，上游的输出结果不等于1。
```

新分支2：$.user_id == 1　　分支2中，上游的输出结果等于1。

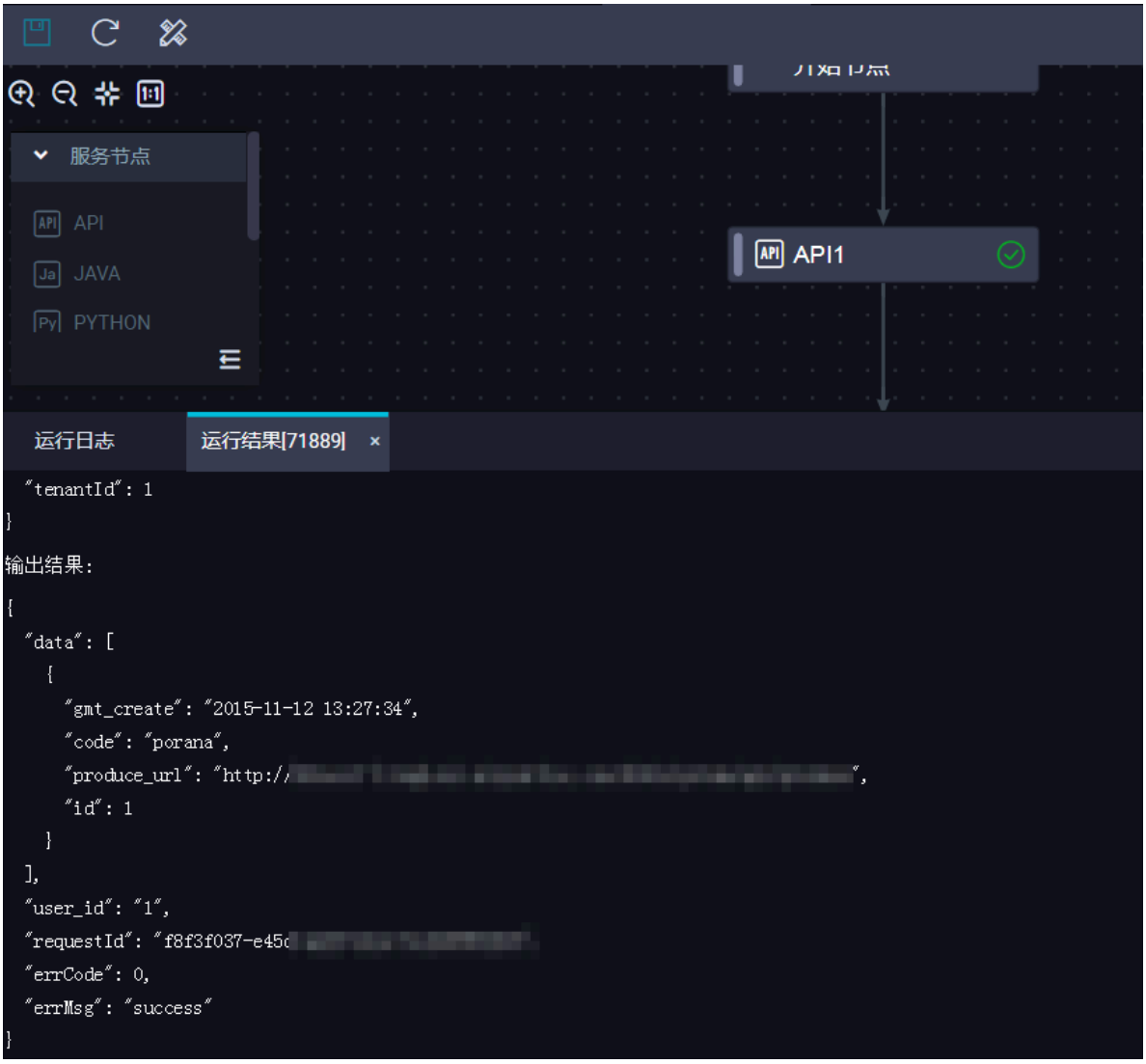
9. 双击结束节点后单击返回参数。

配置返回参数如下所示。

10.单击测试。

输入测试参数。

成功后您可以看到测试结果。



11.10 常见问题

本文将为您介绍数据服务白名单、开通API网关、配置数据源、生成API、API分组、请求方式和协议等方面的常见问题及解决办法。

- **Q：为什么配置数据集成白名单后，数据服务仍不通？**
- **A：因为您的数据库设置了白名单限制，需要加上以下对应区域的白名单。**

区域	白名单
华北2	11.193.100.0/24,11.193.199.0/24
华东2	11.193.96.0/24,11.193.48.0/24,11.193.108.0/24
华东1	11.197.246.0/24,11.193.55.0/24
华南1	11.193.103.0/24,11.193.94.0/24

区域	白名单
成都	11.195.52.0/24
日本	11.199.250.0/24
美国西部	11.193.216.0/24
新加坡	11.197.188.0/24,11.197.227.0/24
上海公安	11.193.98.0/24,11.193.115.0/24
香港（中国）	11.193.200.0/24,11.193.12.0/24
德国	11.199.93.0/24

- Q：是否必须开通API网关?

  A：API网关提供了API托管服务，如果您的API计划对外开放调用，则必须先开通API网关服务。
- Q：在哪里配置数据源?

  A：数据源需要在DataWorks数据集成数据源中进行配置。配置好后，数据服务会自动读取数据源信息。
- Q：向导模式生成API与脚本模式有何功能差异?

  A：脚本模式的功能更为强大，详情请参见#unique_557。
- Q：数据服务中的API分组是干什么用的，与API网关中的分组有什么关联?

  A：API分组一个特定功能或场景的API集合，是数据服务中API的最小组织单元，对应于API网关中的分组概念。简单来说，二者是等同的。数据服务中的API发布到API网关时，系统会在API网关中自动创建一个同名的分组。
- Q：如何设置API分组比较合理?

  A：通常将解决同一个问题或者相似功能的API放在一个分组当中。例如根据城市名称查看天气API、根据经纬度查询天气API可以放在一个名为天气查询的API分组中。
- Q：最多可以创建多少个API分组?

  A：目前一个云账号下最多可创建100个API分组。
- Q：什么情况下要开启API返回结果分页功能?

  A：默认情况下，API最多只会返回500条查询结果。因此当API返回结果有可能超过500条时，请开启返回结果分页功能。当API无请求参数时，一般返回结果会比较多，系统会强制开启返回结果分页。
- Q：生成API是否支持POST请求?

  A：生成API当时仅支持GET请求。

- **Q：是否支持HTTPS协议？**

  A：当前尚不支持HTTPS，在后续的版本迭代中有可能会支持HTTPS，敬请期待。

12 Stream Studio

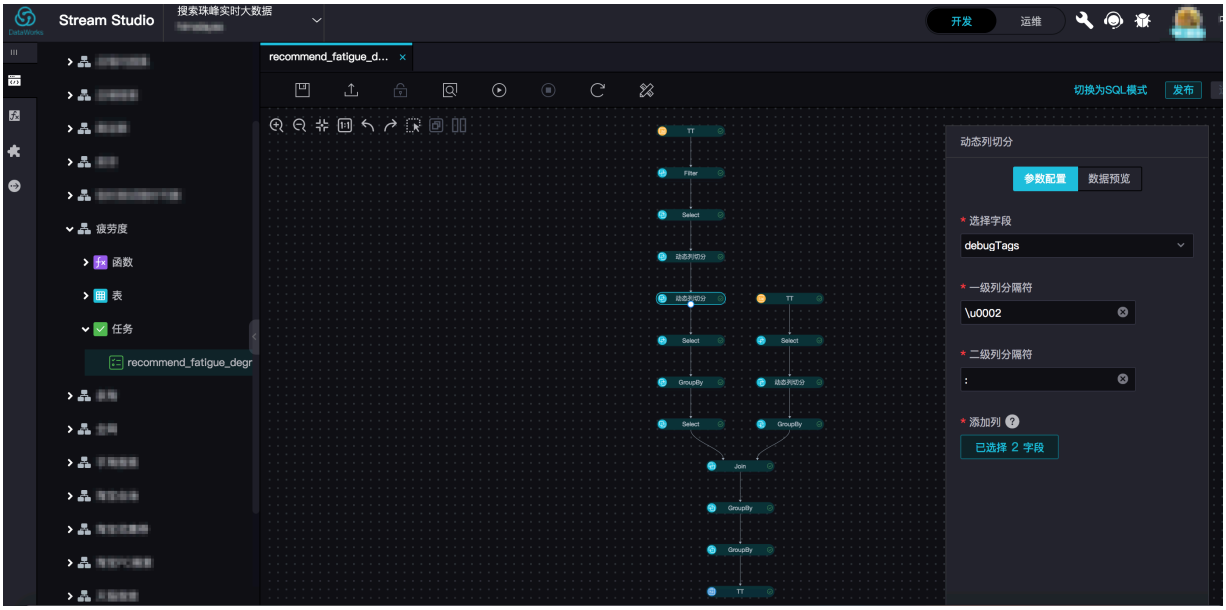
12.1 Stream Studio概述

Stream Studio是DataWorks旗下的一站式实时计算开发平台，请加入钉钉DataWorks Stream Studio交流群（群号：23359532）获取更多支持。

Stream Studio基于阿里云实时计算引擎（基于Flink）构建，集DAG和SQL两种开发模式为一体，并且支持DAG与SQL两种模式互相转换，通过可视化拖拽您就可以轻松实现实时计算作业开发。

Stream Studio是您开发实时计算作业的理想平台，核心功能特性如下：

- 支持DAG模式，通过可视化拖拽即可实现实时计算作业开发。
- 支持Flink SQL模式，您可以选择单纯通过SQL开发实时计算作业。
- 支持DAG与Flink SQL互转，方便查看SQL的算子结构。
- 支持Function Studio在线开发UDF，支持一键发布UDF（仅独享模式支持）。
- 支持作业智能诊断，方便排查线上作业问题。



12.2 绑定实时计算项目

Stream Studio支持共享模式和独享模式的实时计算服务。

前提条件

- 使用Stream Studio前，必须先购买实时计算服务，并创建好项目，然后将实时计算项目绑定至DataWorks的工作空间中。
- 必须购买DataWorks标准版及以上版本，方可绑定实时计算项目。

如果您还未购买实时计算服务，请单击购买实时计算服务进行购买。您可以根据自身需求选择共享模式或独享模式。推荐您使用独享模式以支持更多功能。

背景信息

完成购买后，请进入实时计算控制台的项目管理页面创建项目。详细步骤请参见购买实时计算与创建项目指南。

创建好实时计算项目后，需要与DataWorks工作空间进行绑定。

进入DataWorks控制台，您可以新建一个工作空间，也可以选择已有工作空间进行绑定。

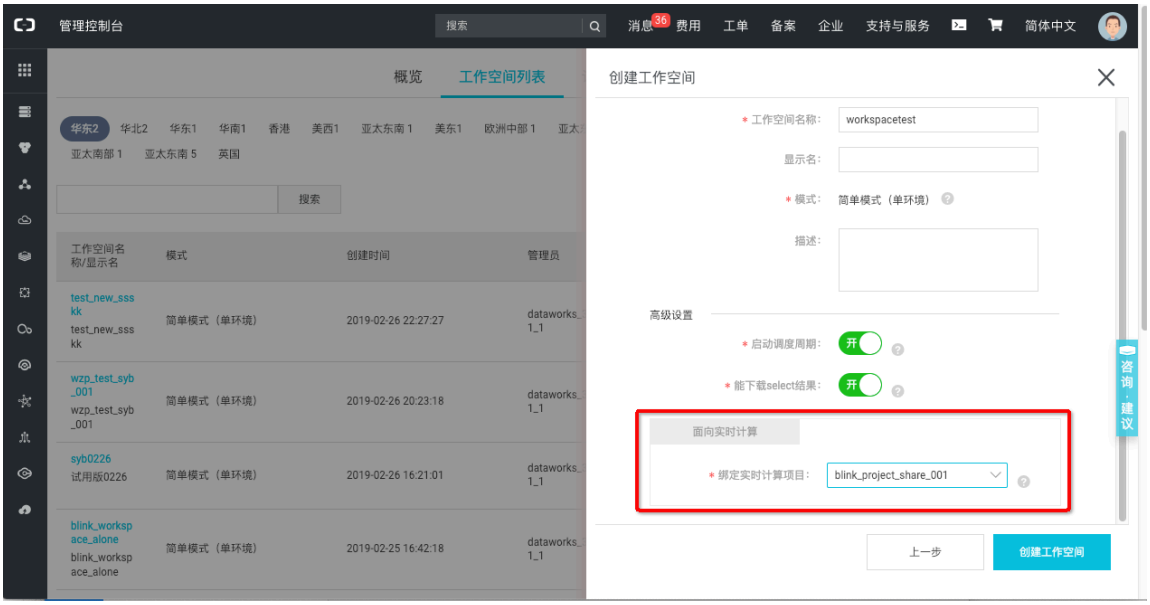
操作步骤

1. 新建工作空间并绑定实时计算项目。

   a) 选择您所在的区域，单击创建工作空间，在选择计算引擎服务中选择实时计算。

   b) 根据您购买的实时计算类型，选择共享模式或独享模式。

   c) 在下一步中绑定实时计算项目。对于共享模式，直接选择要绑定的项目即可。对于独享模式，先选择集群，再选择要绑定的项目。



说明：

每个实时计算项目只能绑定一个工作空间。

2. 已有的工作空间绑定实时计算项目。

选择您所在的区域，然后在工作空间列表中单击修改服务，进行实时计算项目的绑定。

绑定好实时计算项目后，即可在DataWorks中正常使用Stream Studio进行实时计算任务开发。

12.3 快速入门

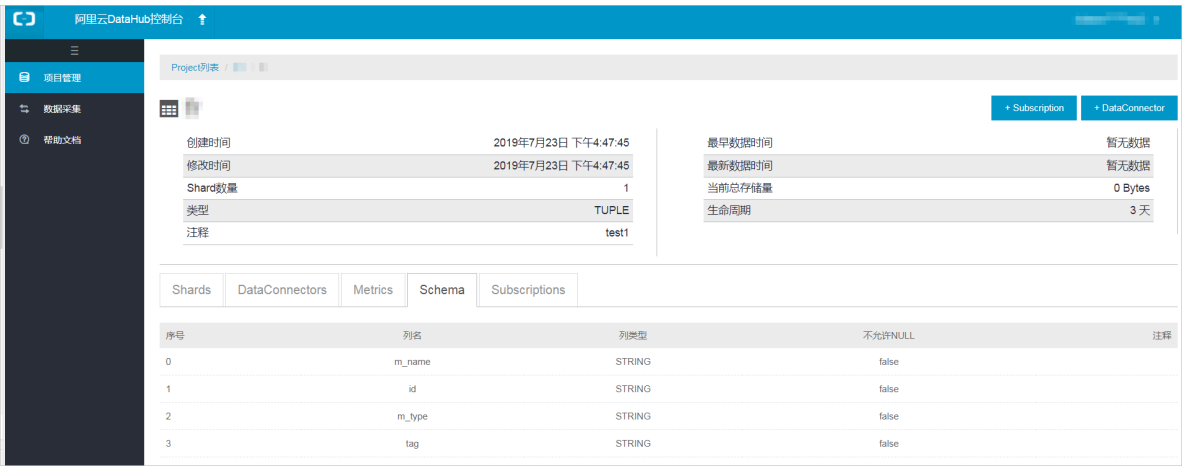
本文将以一个简单的案例，为您介绍如何使用Stream Studio进行实时计算（原流计算）任务的开发和管理，完成一个实时计算任务的新建、开发、发布、启动、停止和下线等所有操作。

前提条件

在您开始快速入门前，请首先绑定实时计算项目，并至少开通DataWorks标准版。

背景信息

- 数据源：1个DataHub表（Topic），包括`m_name`，`id`，`m_type`，`tag`4个字段。



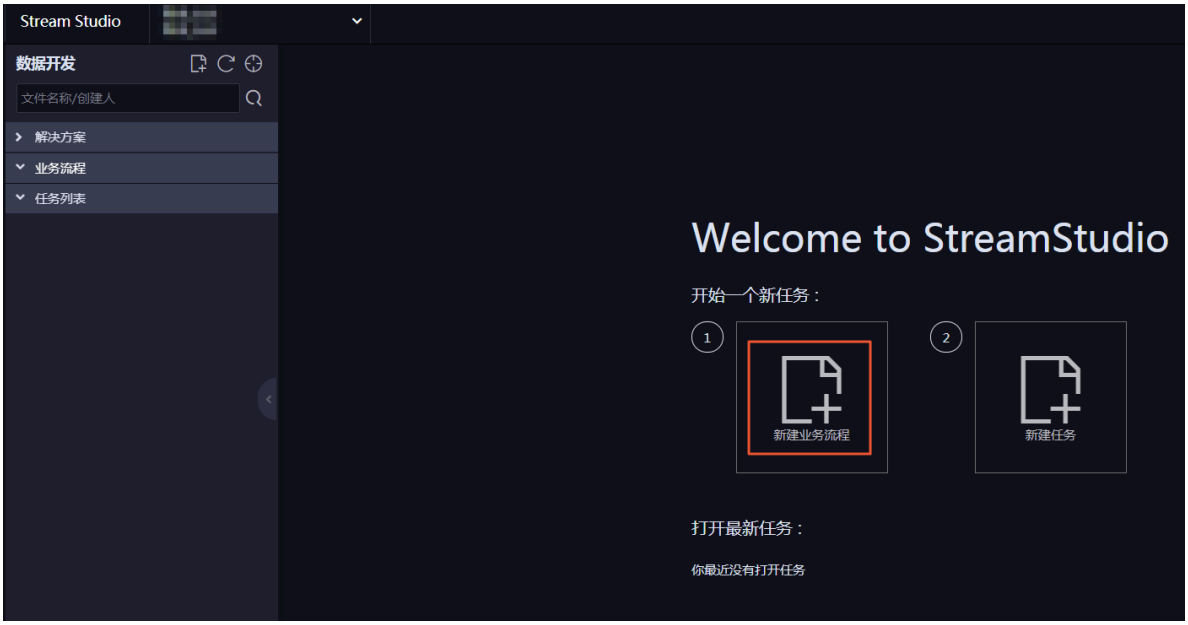
> **说明：**
> 您需要首先创建好Datahub topic，创建方法请参见Datahub Web控制台介绍。

- 数据处理：针对tag字段进行分割，分割符为`;`，分割后成为`color`、`mode`和`weight`3个字段。
- 数据结果输出：最后选择`id`，`m_type`，`weight`写入至日志服务（SLS）表中。

> **说明：**
> 您需要首先创建好日志服务项目与Logstore，创建方法请参见#unique_566。

操作步骤

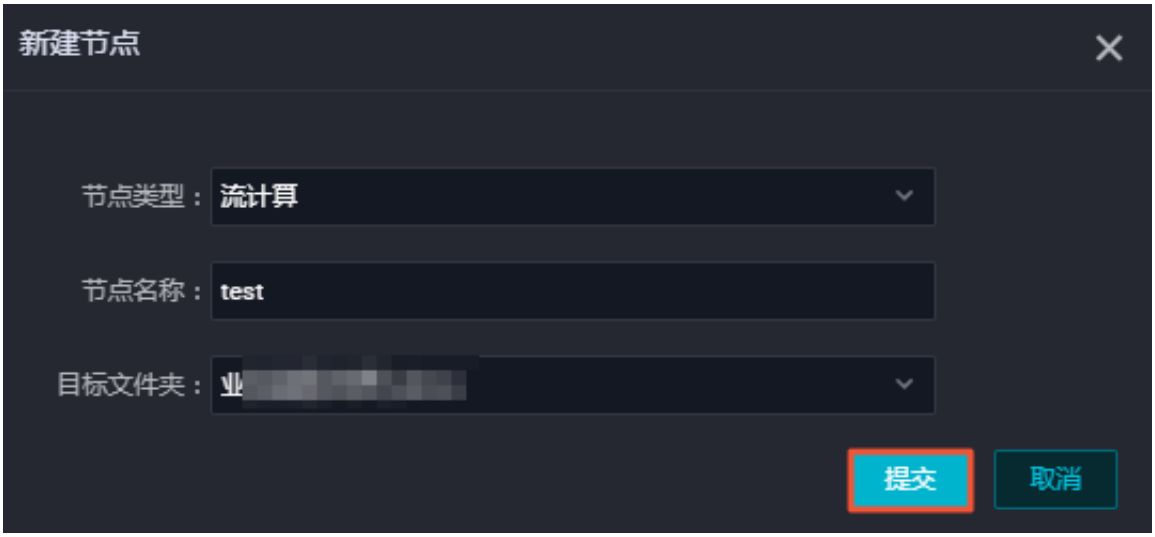
1. 新建业务流程。



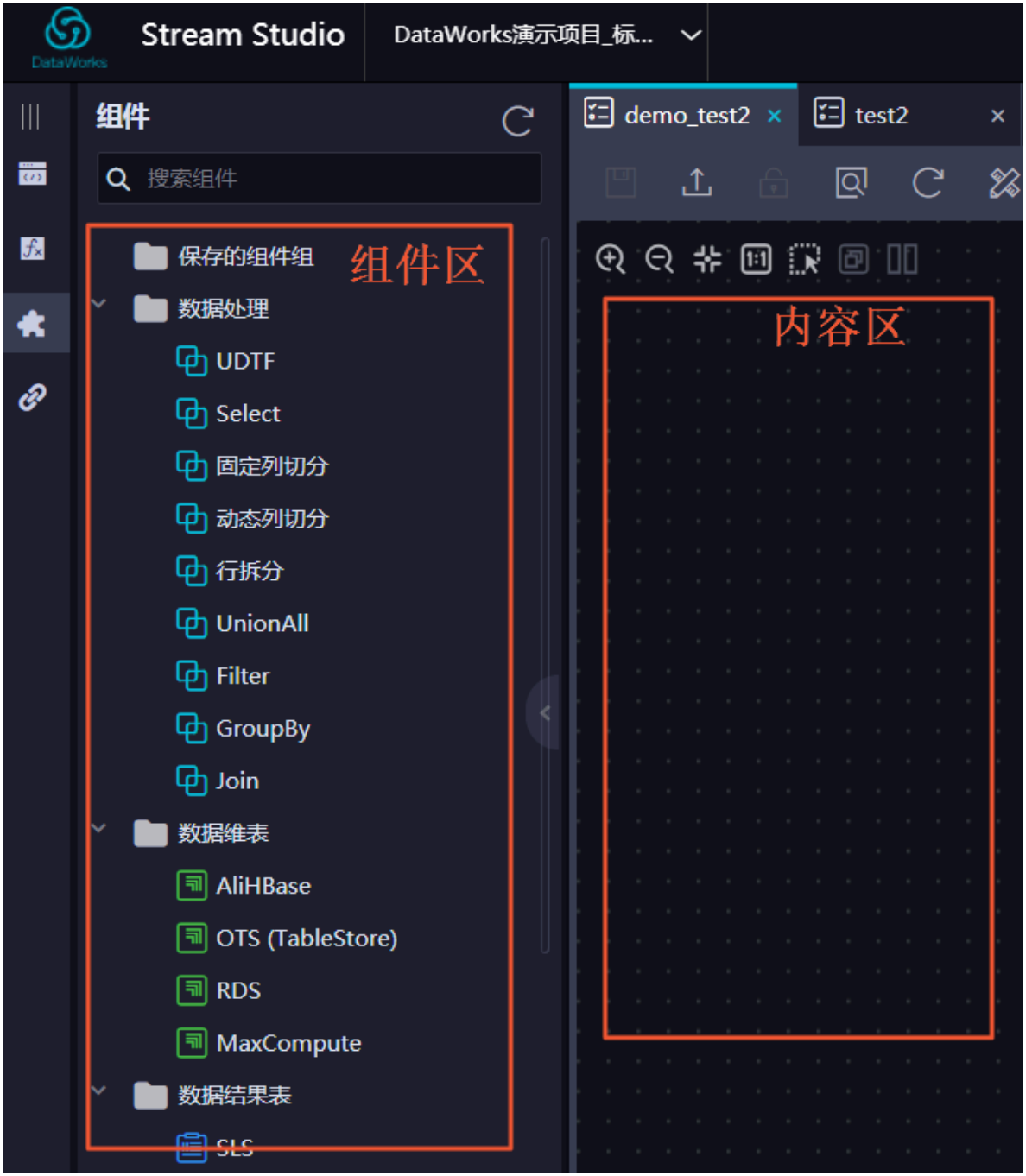
2. 新建实时计算任务。

进入Stream Studio后，在数据开发中，单击新建按钮，选择任务 > 流计算。

输入节点名称，选择目标文件夹。

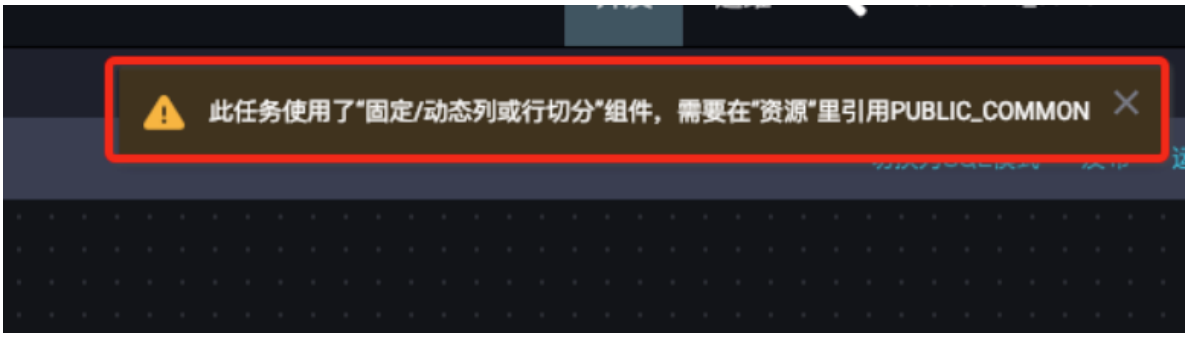


单击提交，即可看到组件的界面。



在资源引用中勾选PUBLIC_COMMON。

如果不勾选，后续使用固定列切分组件时会收到如下提示：

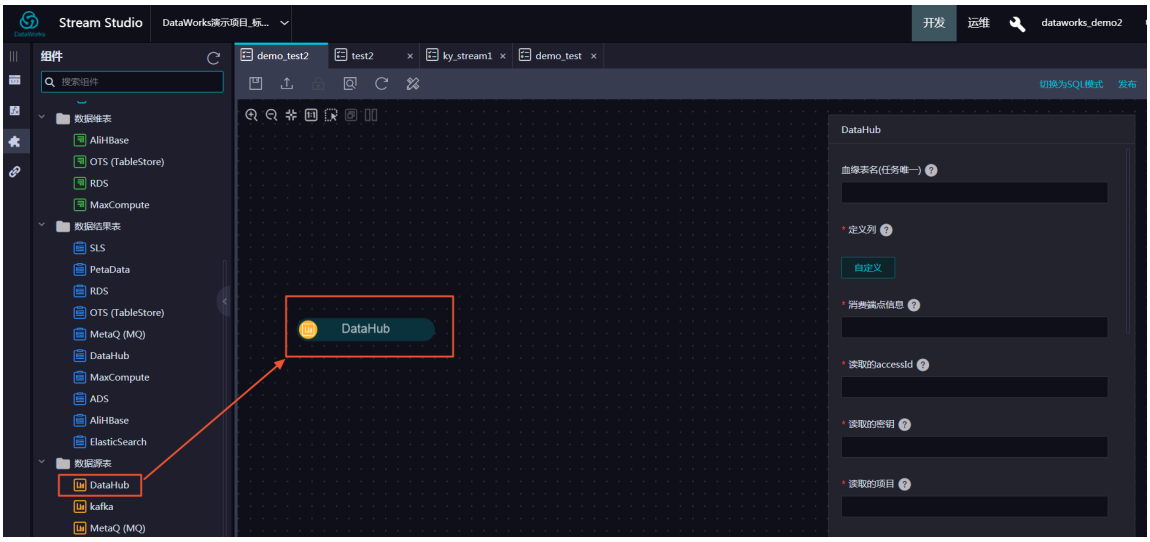


您也可以在出现上述提示后再勾选PUBLIC_COMMON。

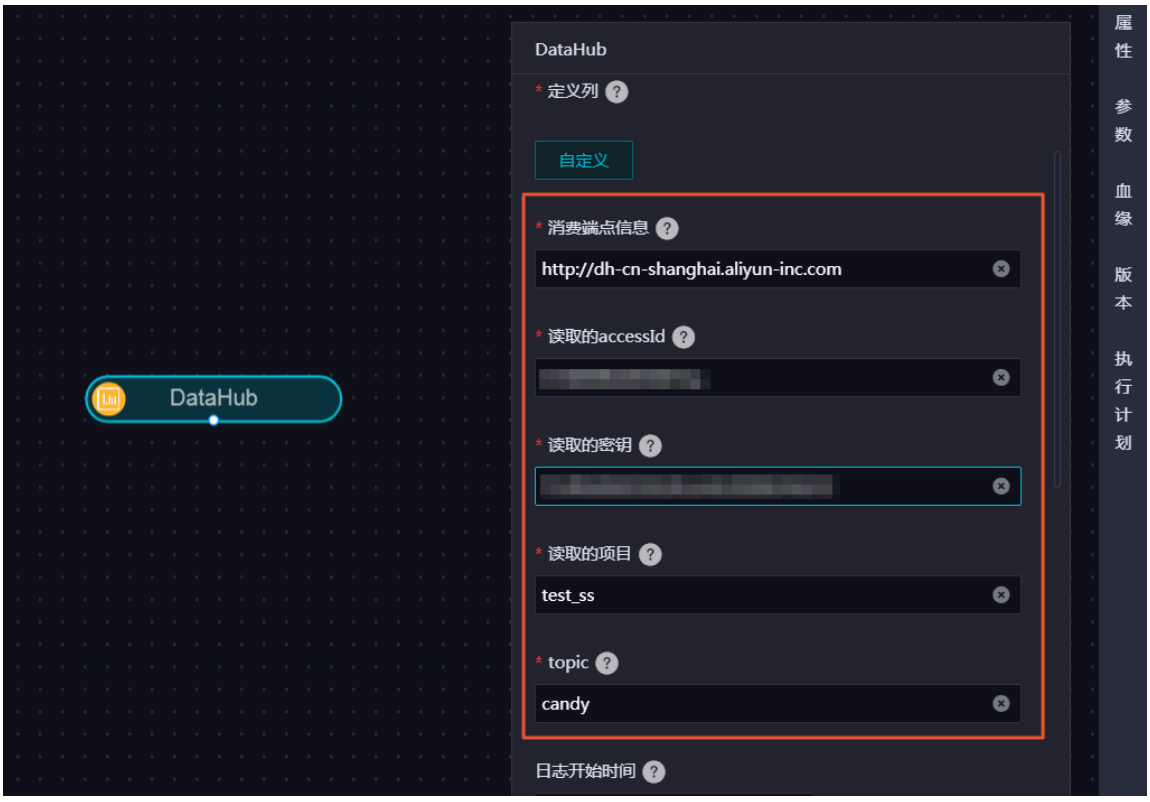
3. 编辑实时计算任务。

a) 在组件页面新增数据源。

拖拽DataHub节点到面板。



双击DataHub组件节点，填写参数。



配置	说明
Endpoint	· 共享集群选择：经典网络ECS，`http://dh-cn-shanghai.aliyun-inc.com`。 · 独享集群选择：VPC ECS，`http://dh-cn-shanghai-int-vpc.aliyuncs.com`。
读取的accessID	填写您Datahub账号的AccessID。
读取的秘钥	填写您Datahub账号的AccessKey。
project	填写相应的项目名称。

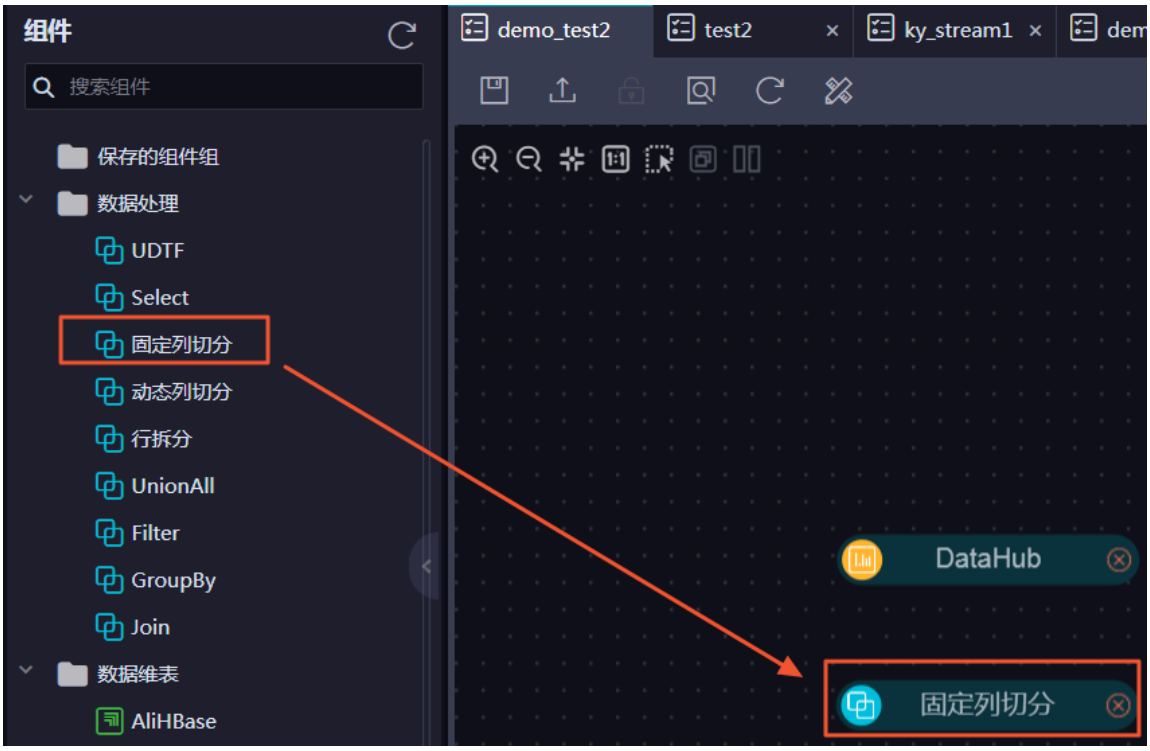
配置	说明
topic	填写相应的topic名称。

添加输出列：选择定义列 > 自定义，单击添加。

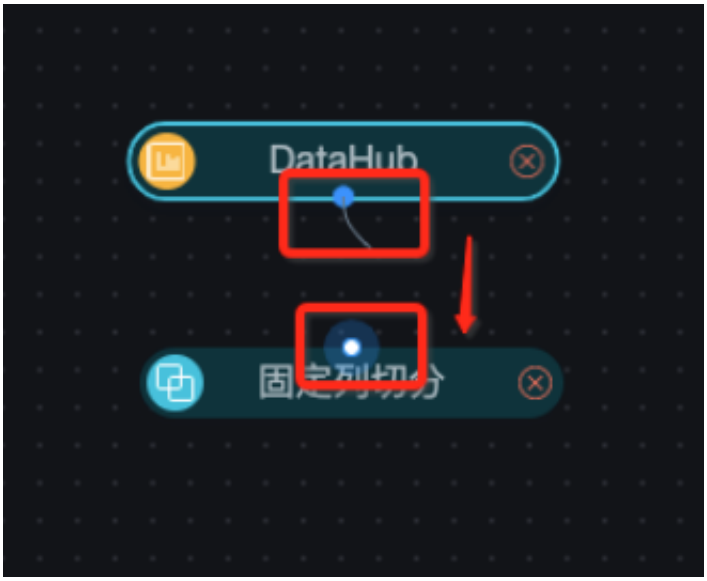
输入名称和类型（VARCHAR），单击确认。

b) 数据处理。

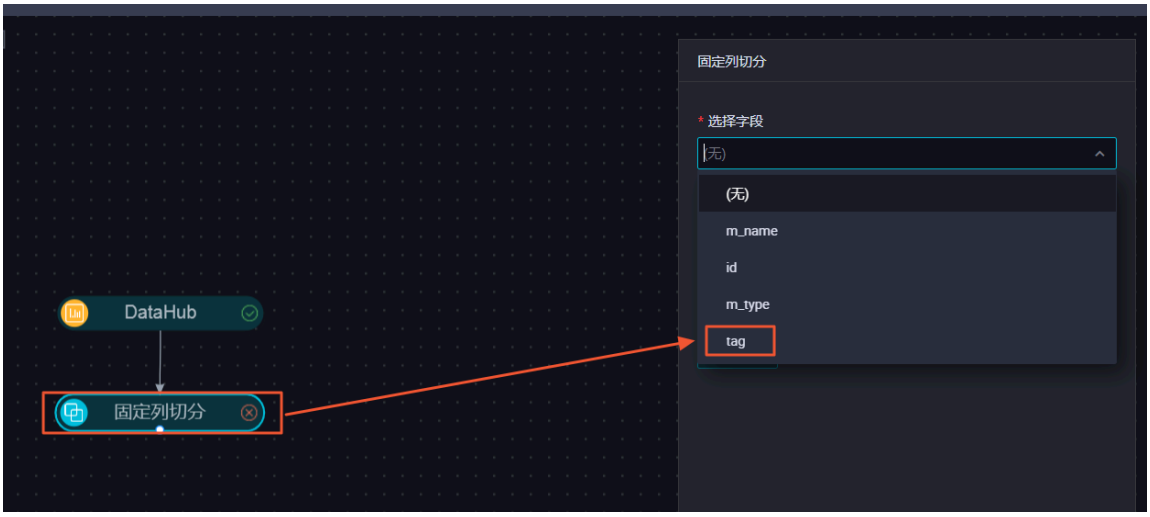
首先进行字段分割。拖拽一个固定列切分组件到面板中。



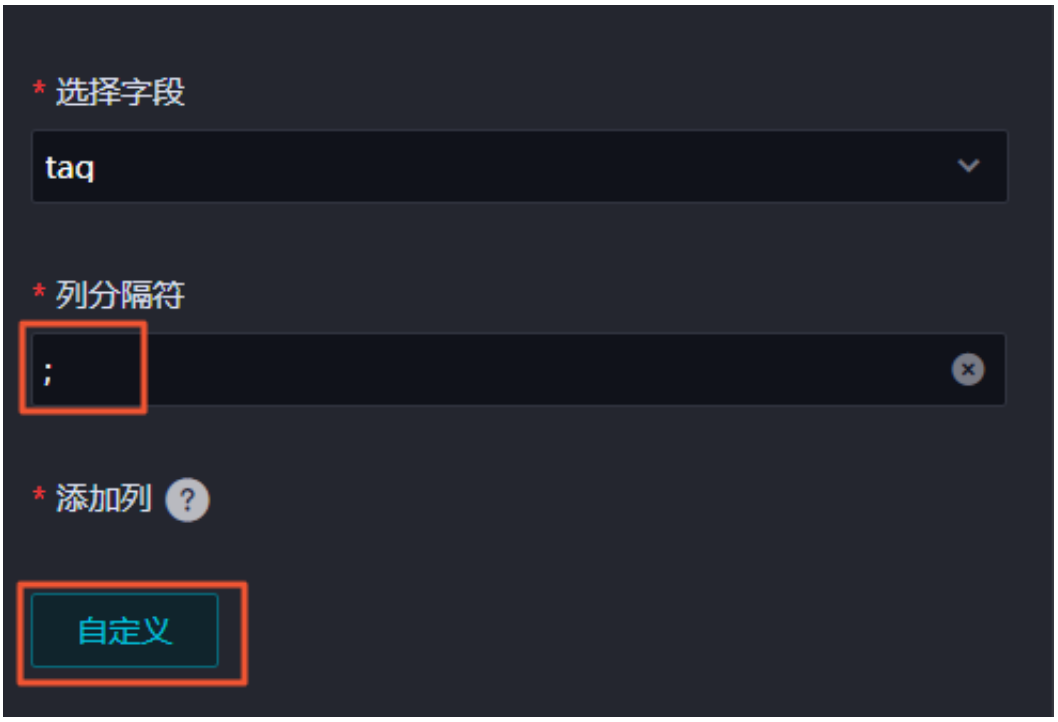
将DataHub组件连线到固定列切分组件。



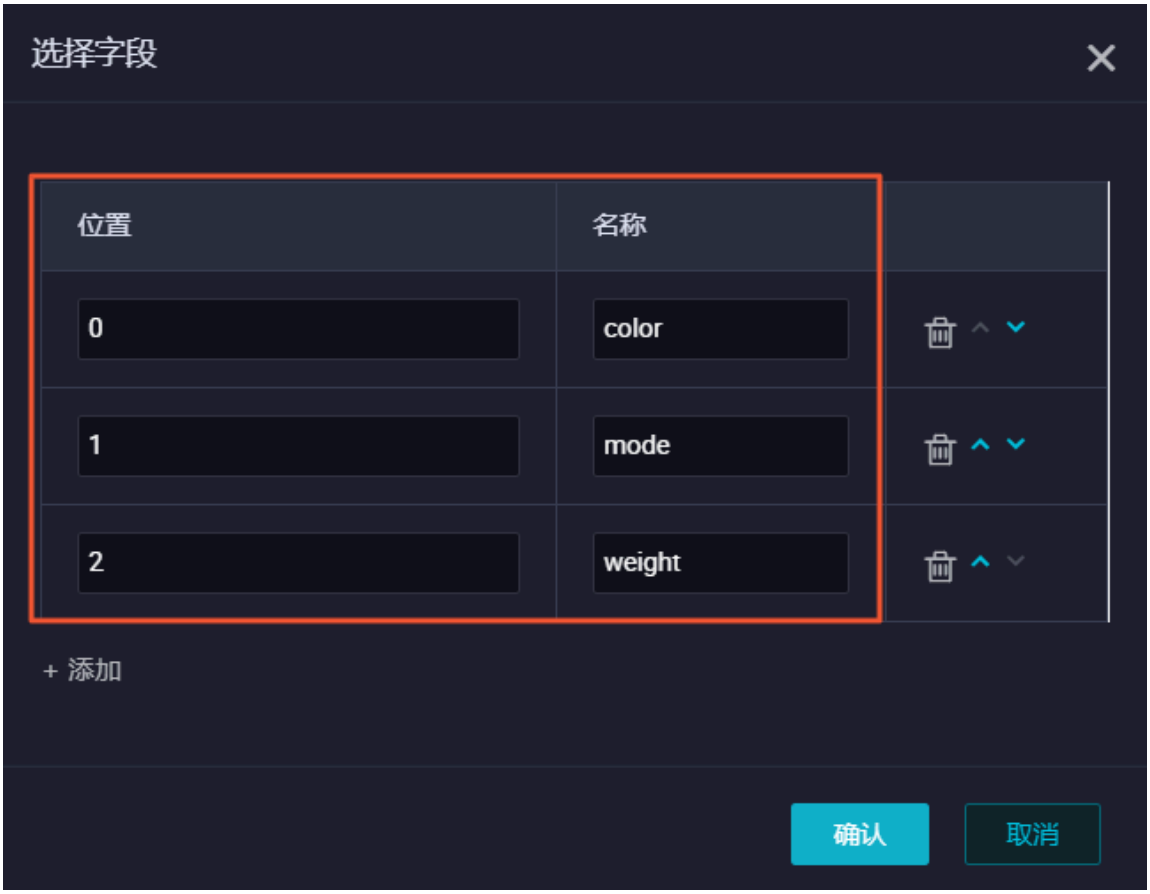
双击固定列切分组件，打开属性面板，选择要分割的字段为tag。



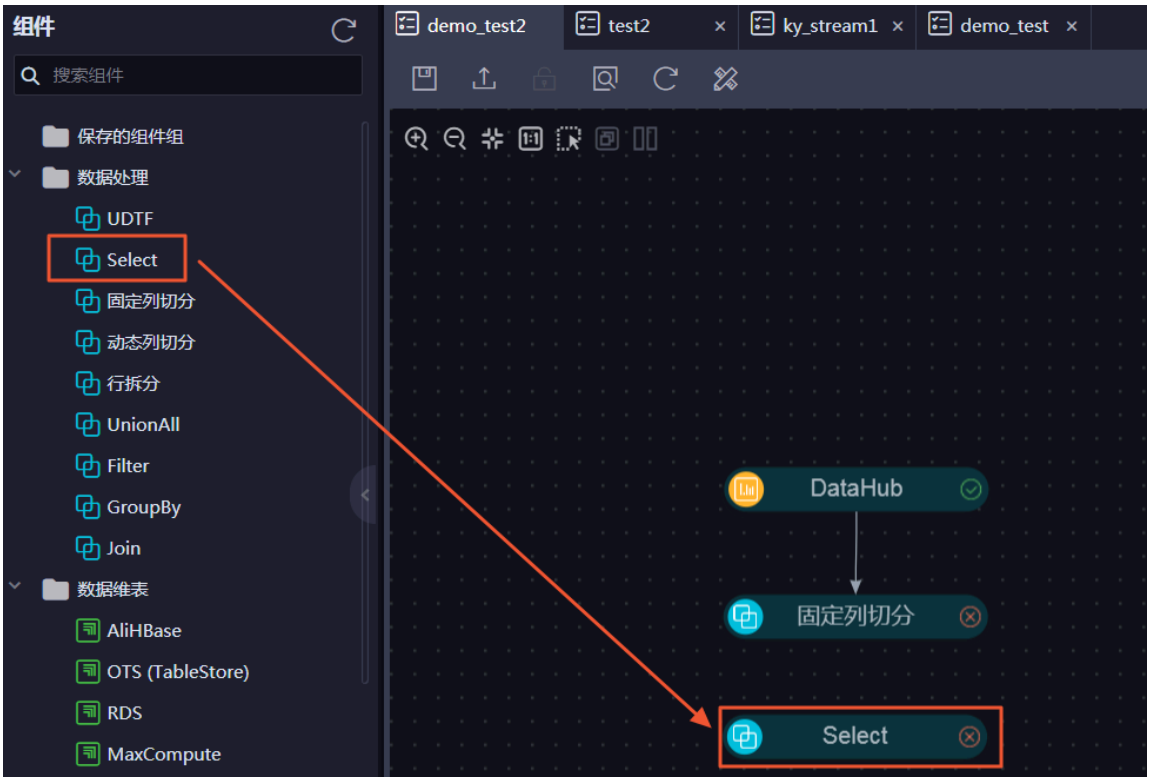
分隔符修改为;。单击自定义，定义输出列。



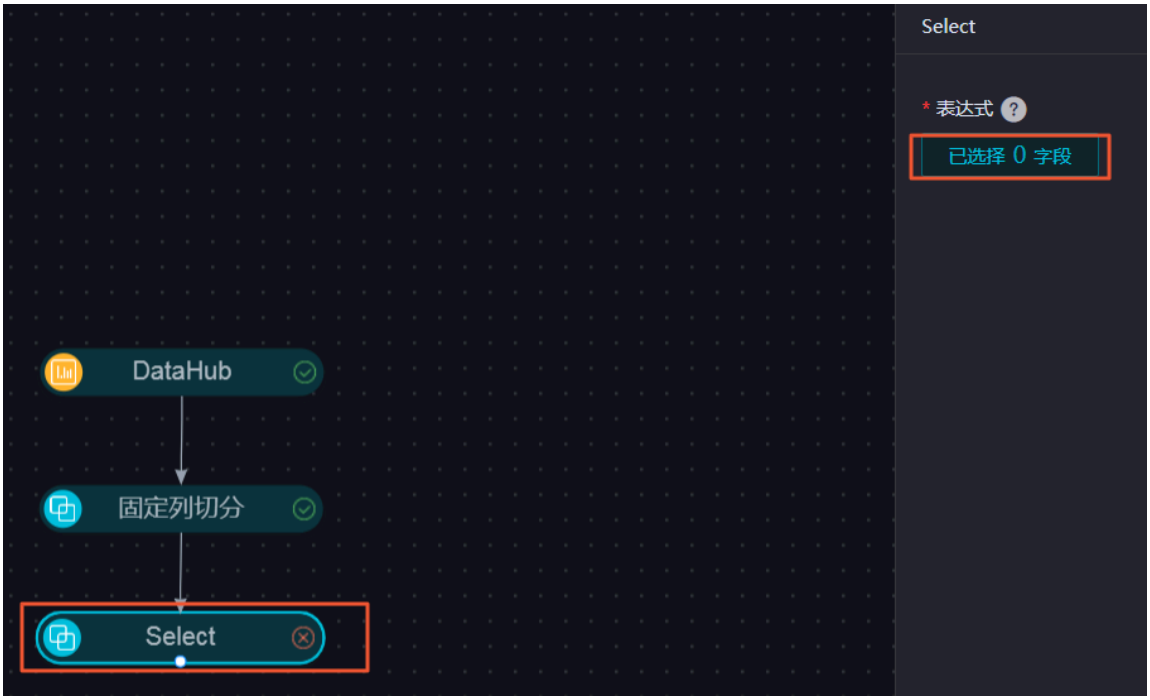
输入分割后字段的位置和名称，单击确认。



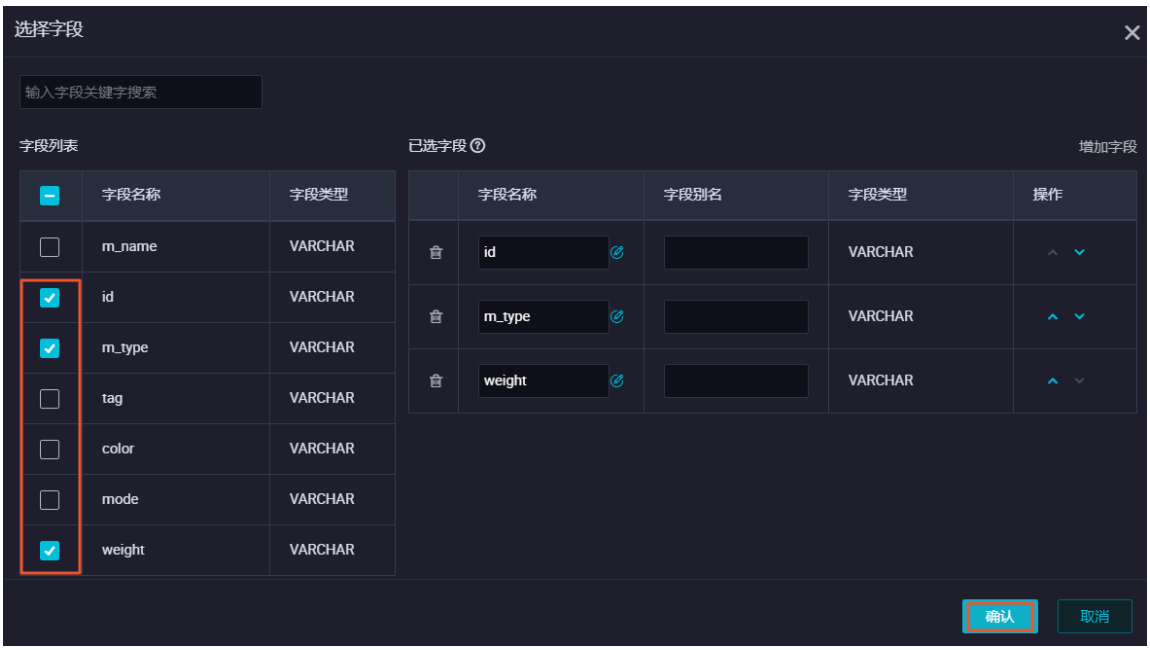
选择要输出的字段：拖拽出Select组件。



连接固定列切分组件与Select组件。双击Select节点，在面板中单击已选择0字段。

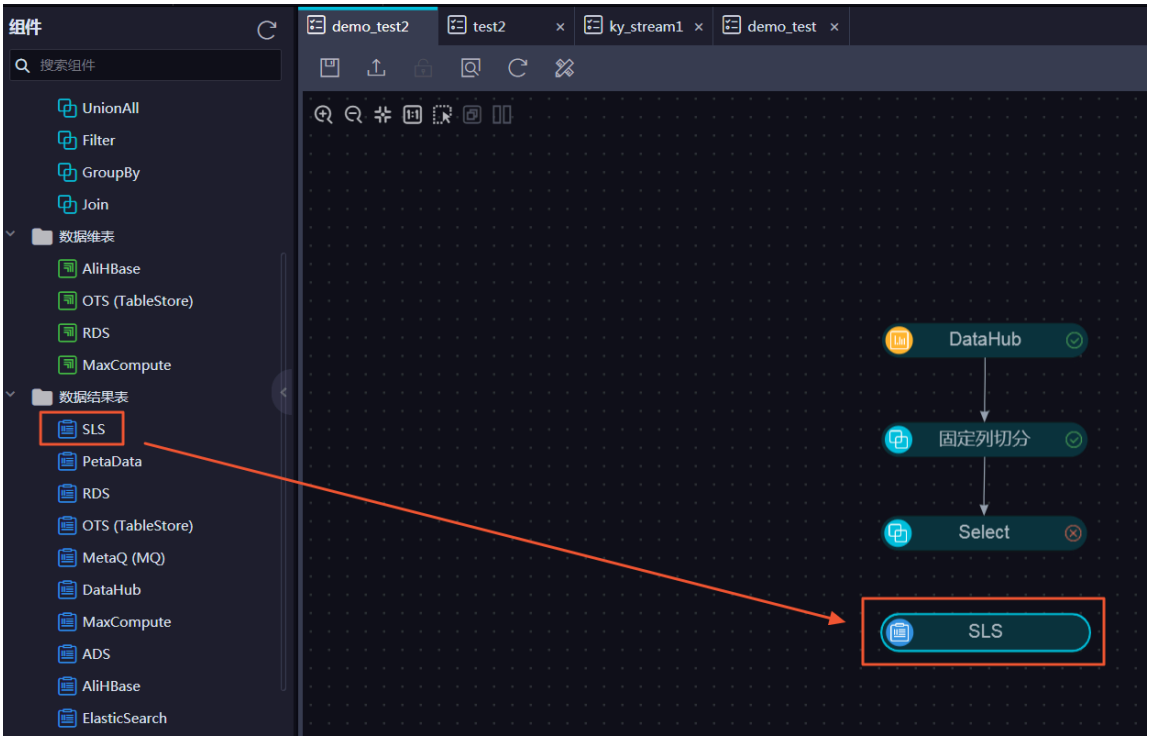


在弹出的选择字段对话框中，勾选`id`，`m_type`，`weight`这三个字段，单击确认。

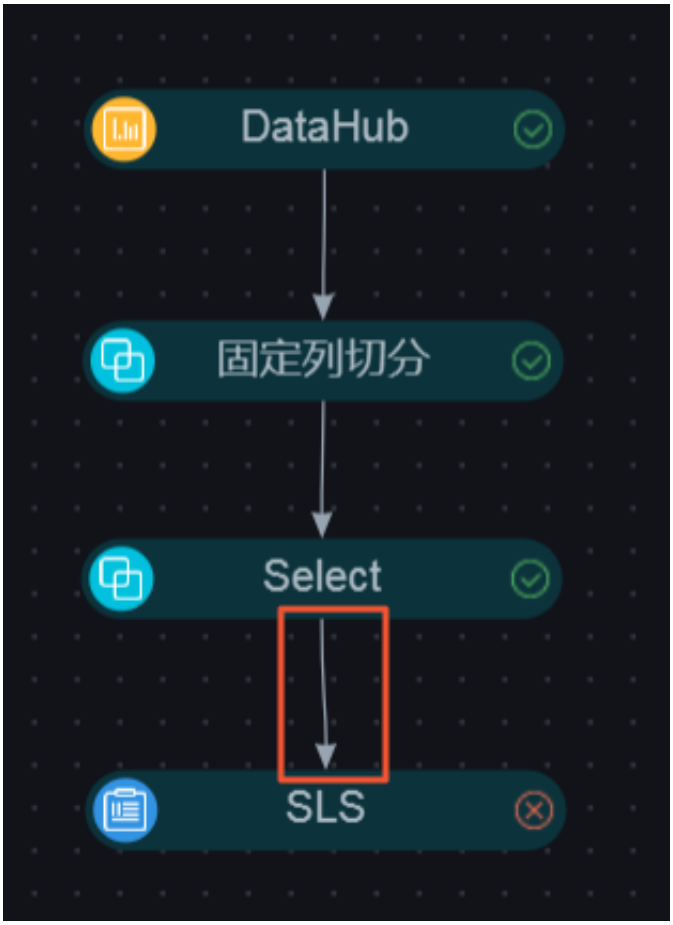


c) 数据结果输出。

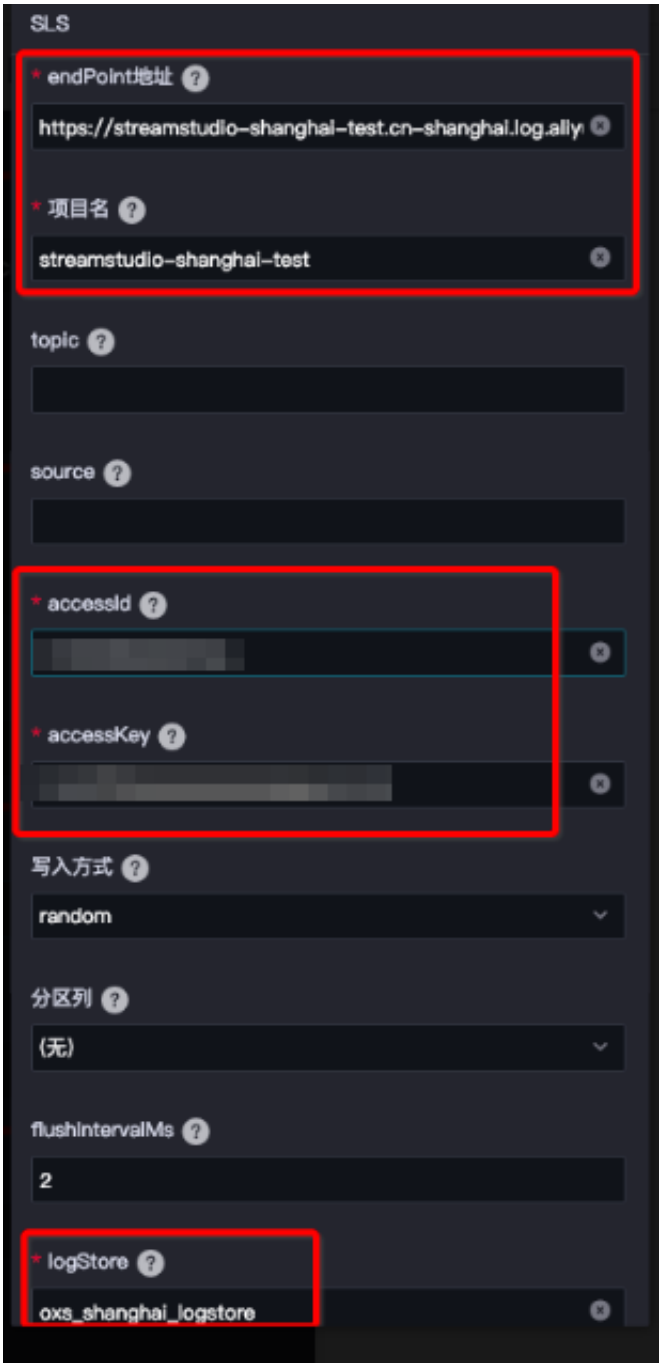
本例中，将上面处理后的数据结果输出到SLS中。在组件面板中，拖出SLS组件。



连接Select组件和SLS组件。



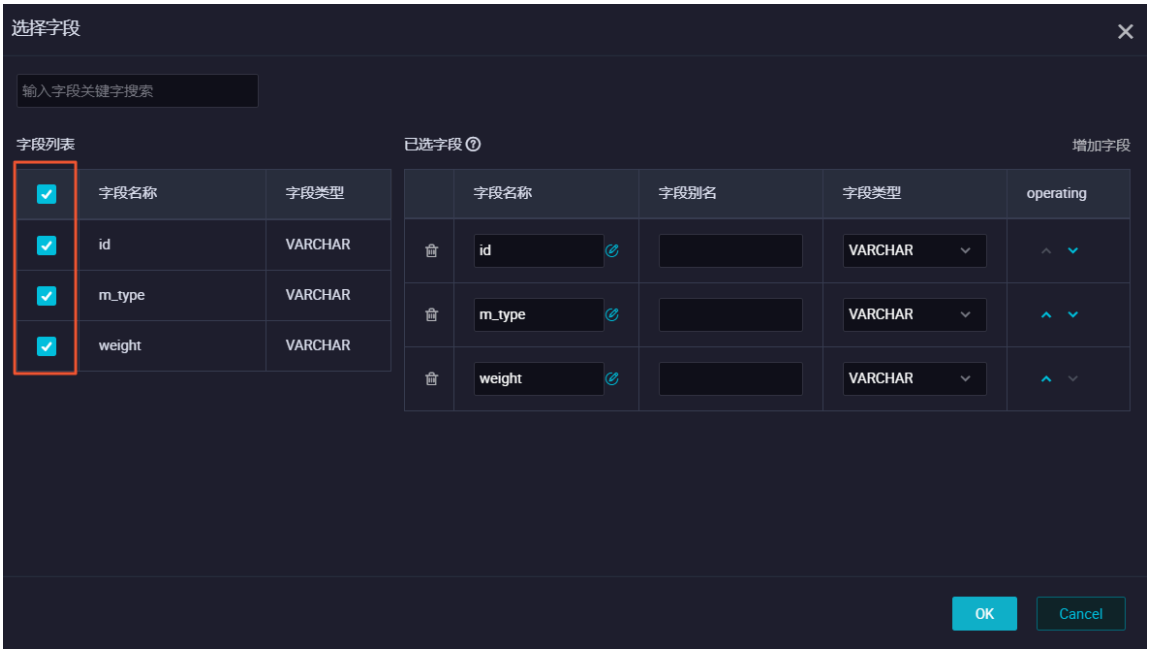
单击SLS组件，填写组件参数。



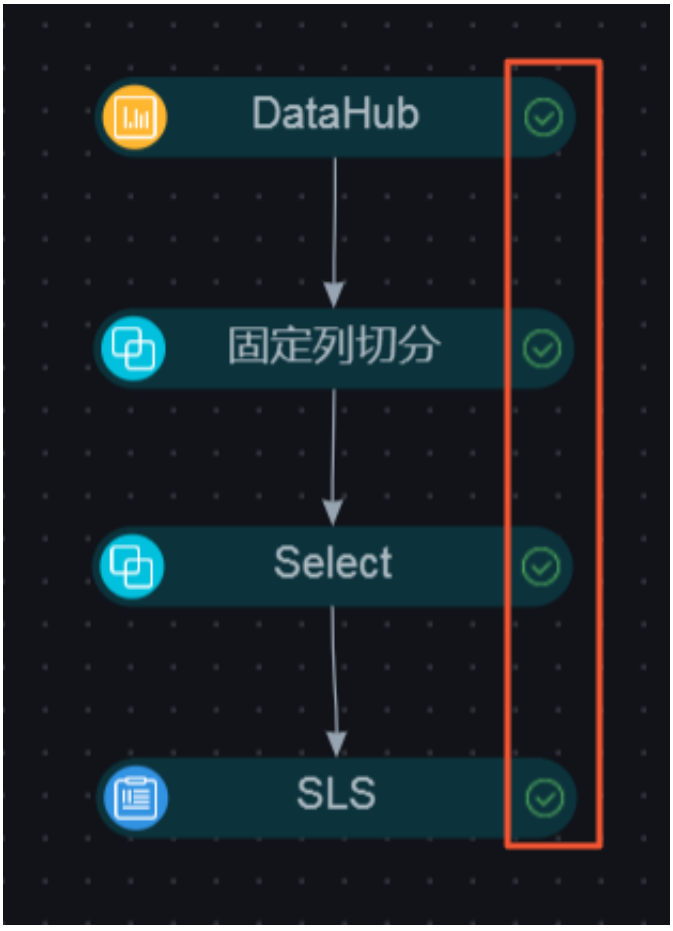
配置	说明
endPoint地址	`https://streamstudio-shanghai-test.cn-shanghai.log.aliyuncs.com`。
读取的accessId	填写SLS账号的AccessID，在本文中使用的是同一个AccessID。
读取的秘钥	填写SLS账号的AccessKey，在本文中使用的是同一个AccessKey。
project	填写相应的项目名称。

配置	说明
logStore	填写相应的logStore名称。

选择输出字段，单击已选择0字段，直接全选即可。



单击OK后，此时您应该可以看到所有节点显示正确。

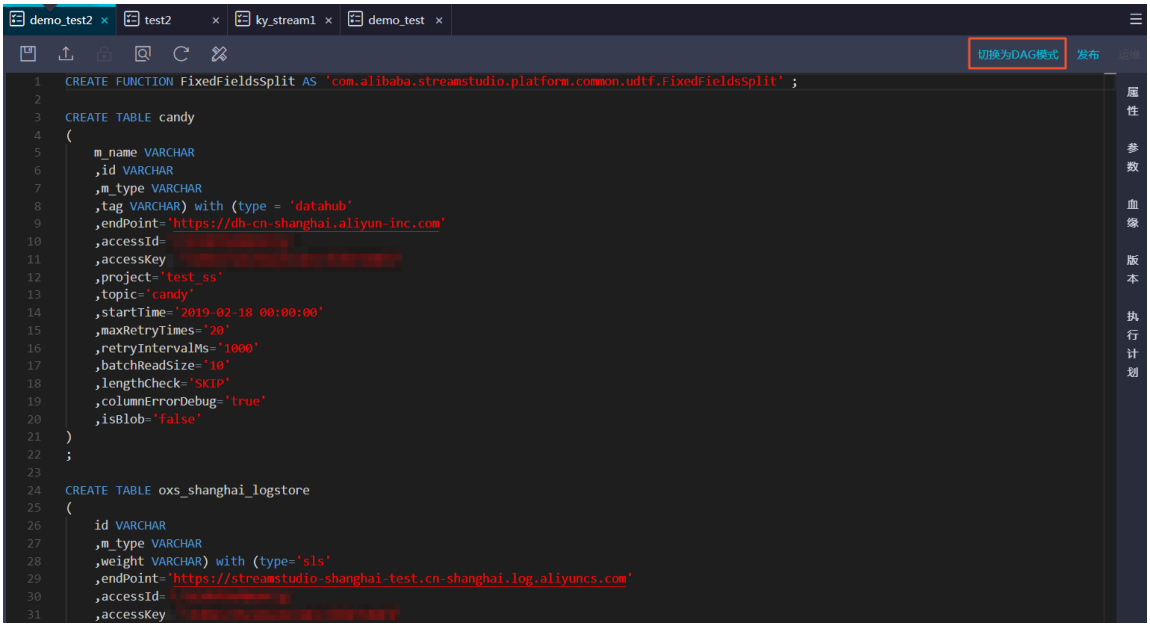


d) 切换DAG模式与SQL模式。

Stream Studio支持DAG模式编辑和SQL模式编辑流任务，两种模式对等，且支持互相转换。

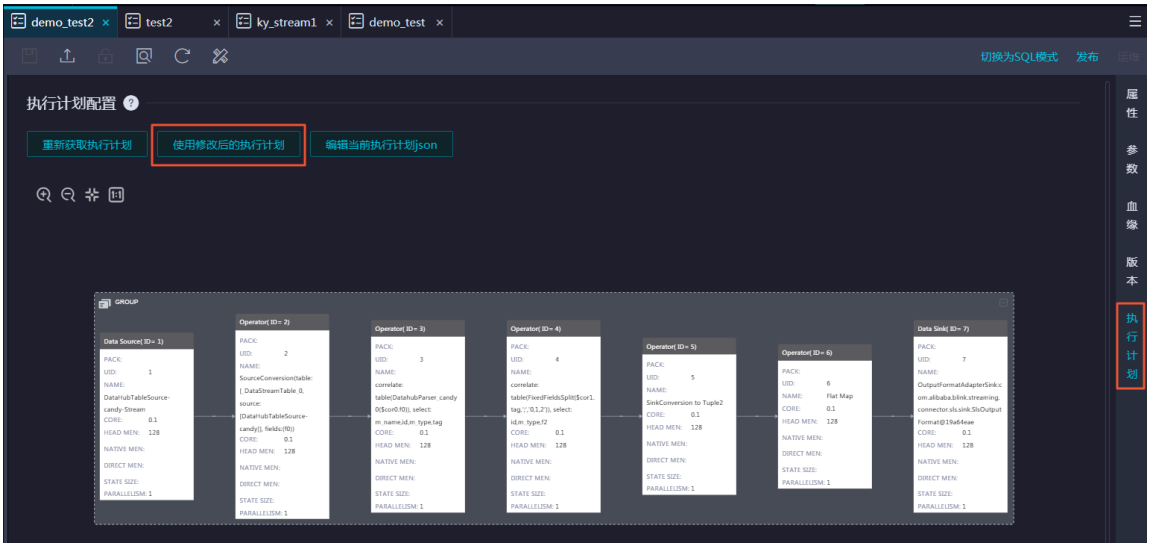
单击切换为SQL模式，即可以将DAG节点转换为对应的SQL。

SQL视图如下所示，单击切换为DAG模式即可再次切换回DAG模式。



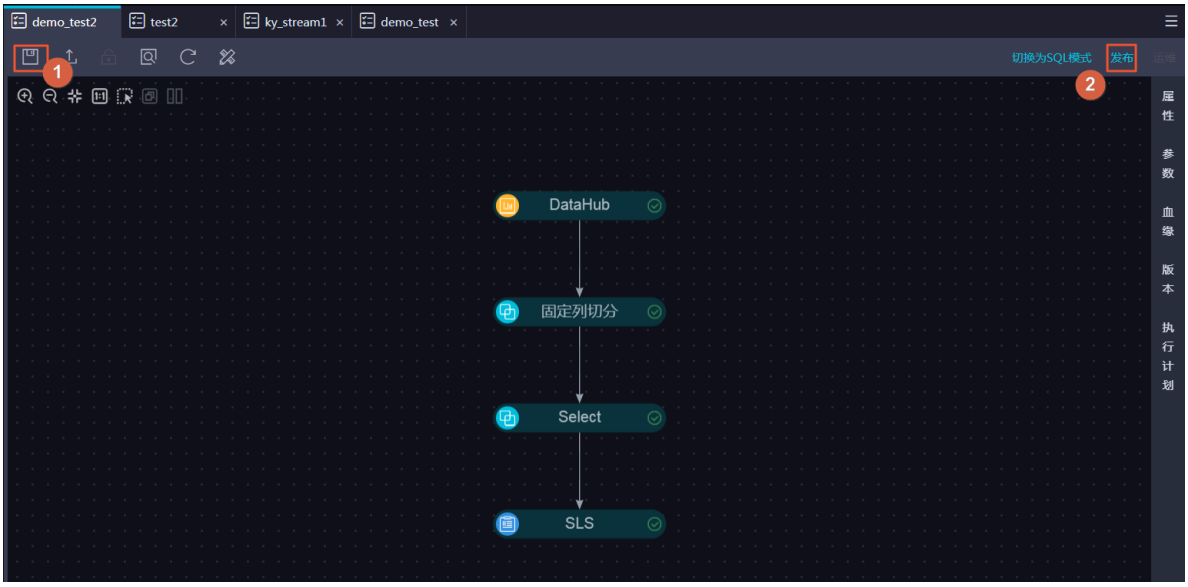
e) 设置执行计划。

单击执行计划，生成执行计划。单击使用修改后的执行计划。

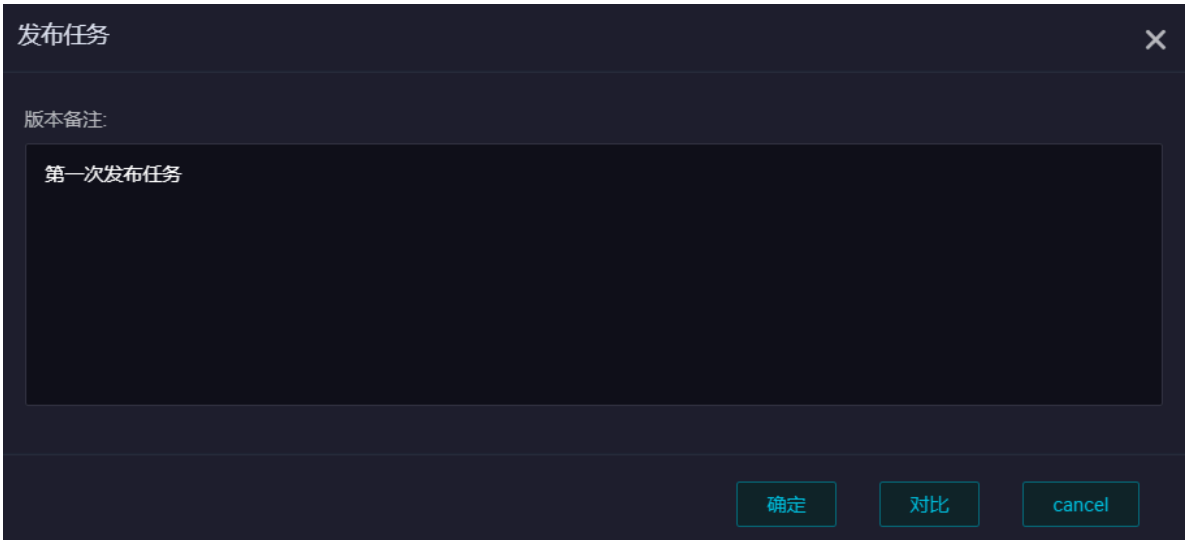

demo_test2
test2
ky_stream1
demo_test
切换为SQL模式
发布
执行计划配置
重新获取执行计划
使用修改后的执行计划
编辑当前执行计划json
属性
参数
血缘
版本
执行计划
GROUP

4. 发布实时计算任务。

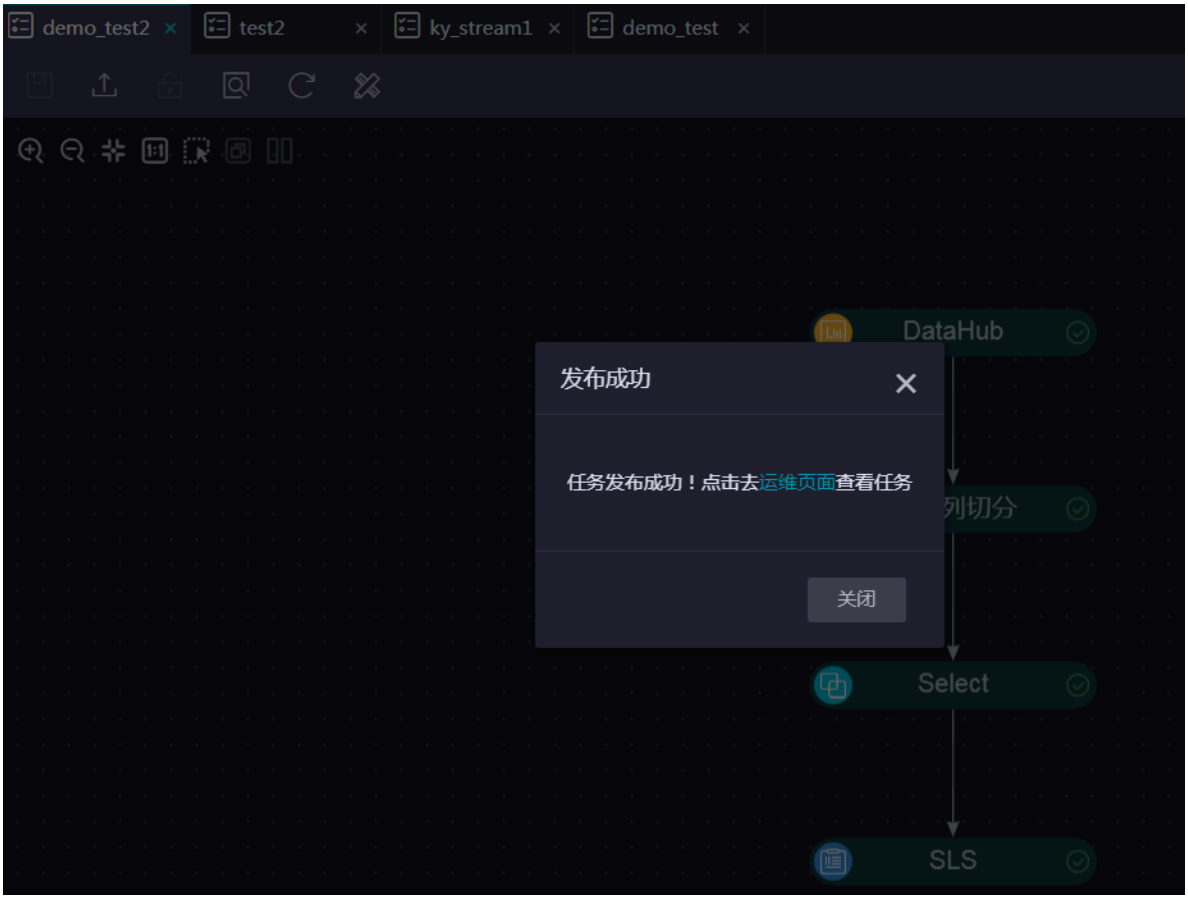
编辑好流任务之后即可进行任务发布。单击保存，然后单击发布（发布时会检查是否保存，如果没有保存也会提示保存）。



发布注释为非必选项，单击确定发布。

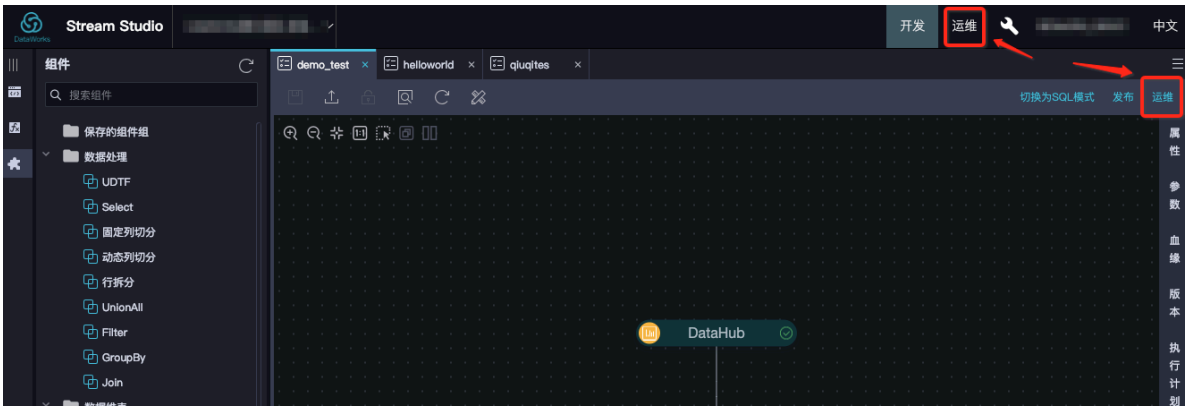


发布完毕后，可以去运维页面中查看任务状态和进行任务管理。


demo_test2
test2
ky_stream1
demo_test
DataHub
发布成功
任务发布成功！点击去运维页面查看任务
关闭
列切分
Select
SLS

5. 运维实时计算任务。

单击进入运维页面。

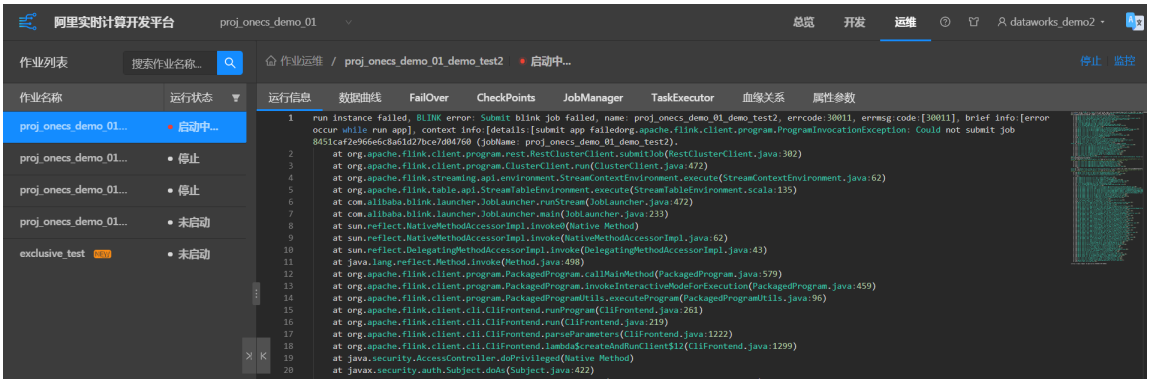


a) 启动任务。

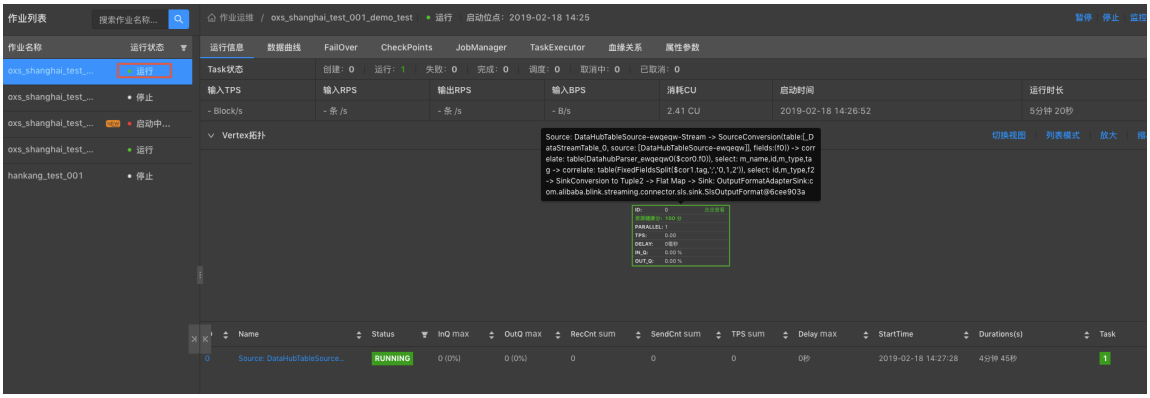
在任务列表中找到我们新建实时计算任务，单击启动，即可启动任务。

根据自己的实际业务需要，设置启动时间。

任务启动后，单击任务名称即可看到详细启动状态。



正常启动完毕即显示运行。



b) 停止和下线任务。

单击停止即可停止任务。

任务停止后，单击下线即可完成任务下线。

预期结果

至此便完成一个实时计算任务的新建、开发、发布、启动、停止和下线等所有操作。

12.4 数据采集

所有的大数据分析系统都基于一个前提，即数据需要经过采集才能进入大数据系统。

为最大化利用您现有的流式存储系统，阿里云实时计算对接了多种上游的流式存储，让您可以不用额外进行数据的采集，即可享受现有的数据流式存储。

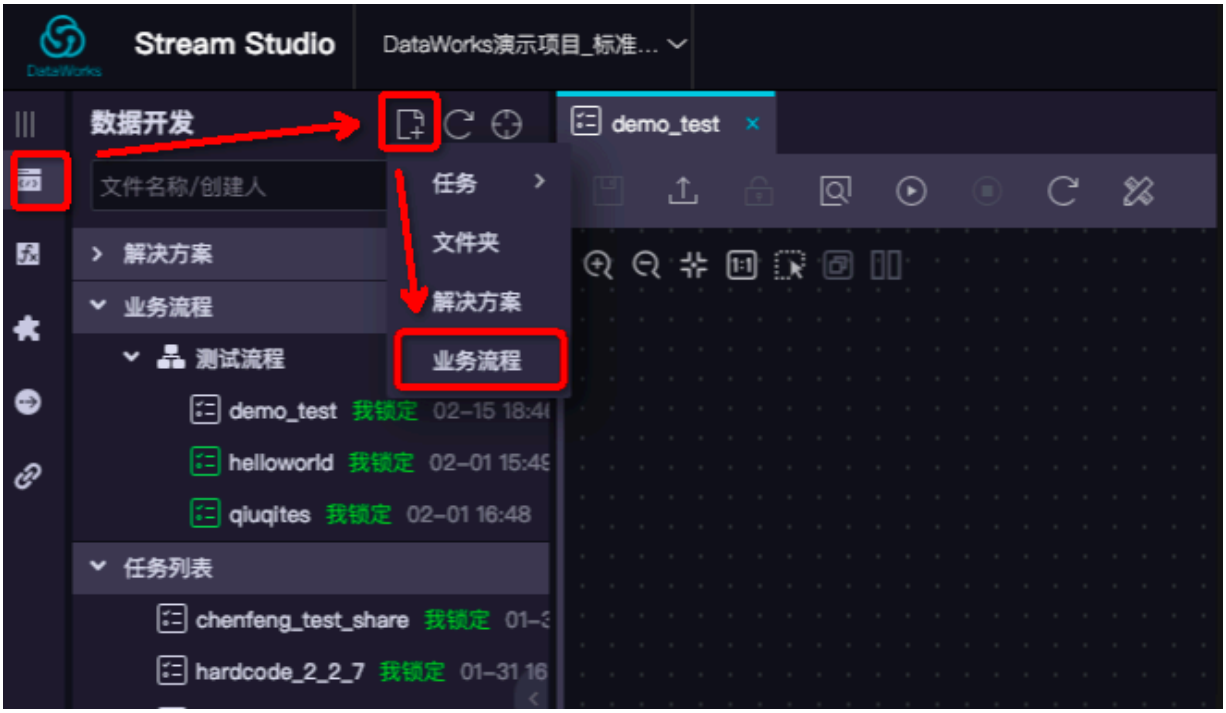
数据采集的详细说明请参考实时计算帮助文档。

12.5 新建实时计算任务

本文将为您介绍如何新建实时计算任务，并通过Stream Studio进行数据开发。

前提条件

如果您还没有业务流程，请先新建一个业务流程。如果您已有业务流程，可以略过。

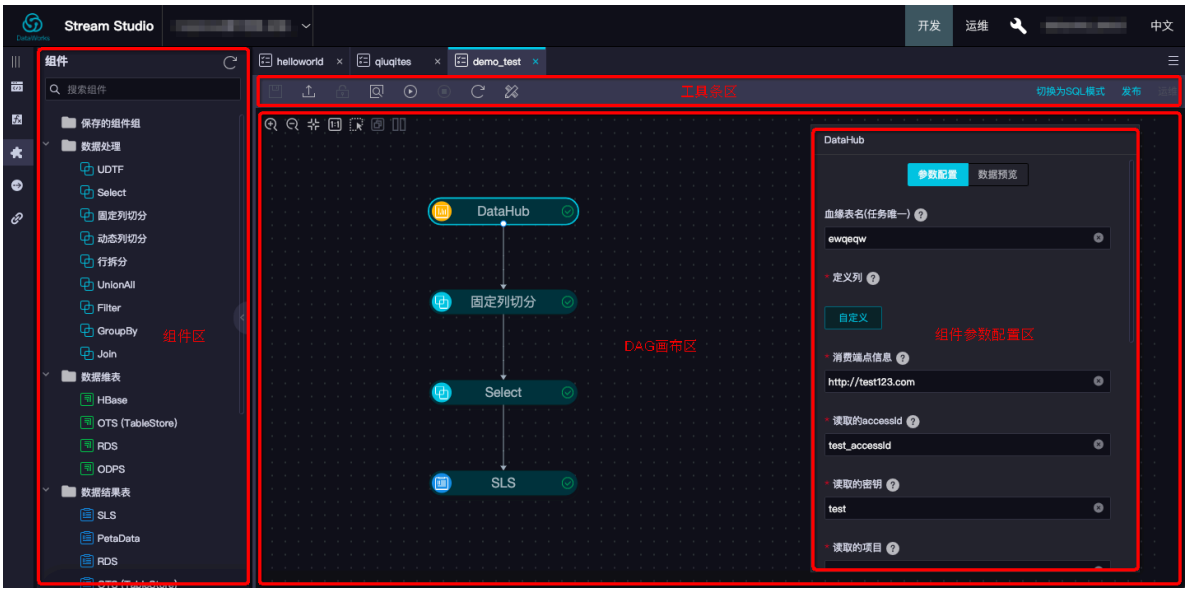


完成业务流程的创建后，在数据开发列表中，单击新建按钮，选择任务 > 实时计算开发，新建实时计算任务。默认以DAG模式进行任务开发。

操作步骤

1. DAG模式开发。

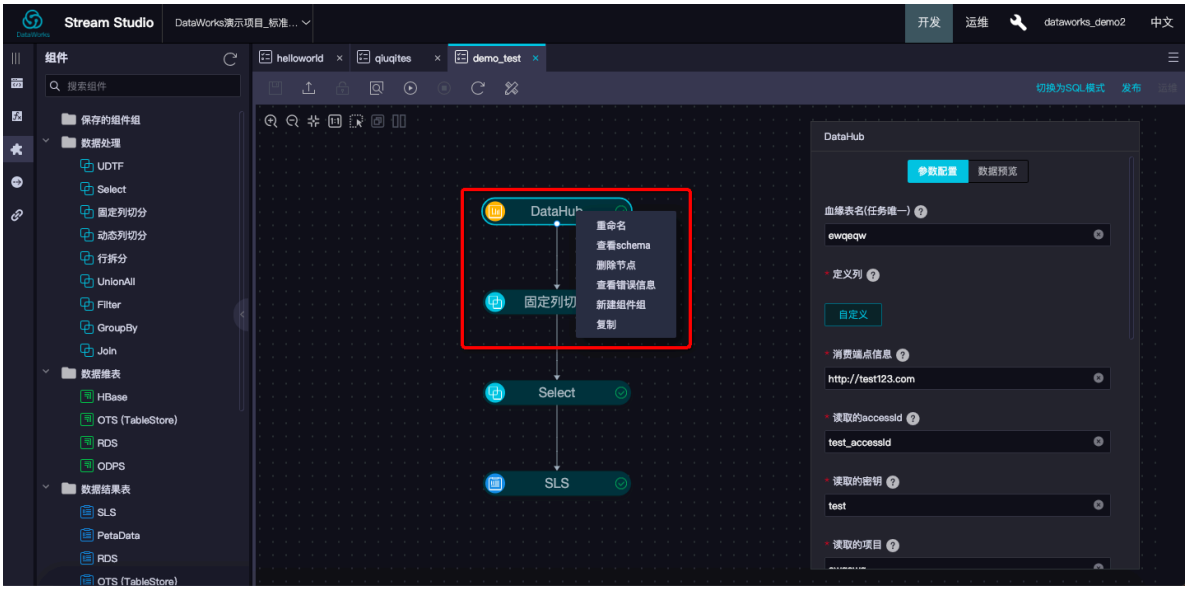
   DAG模式开发的产品界面如下所示。



   整体分为四个区域：

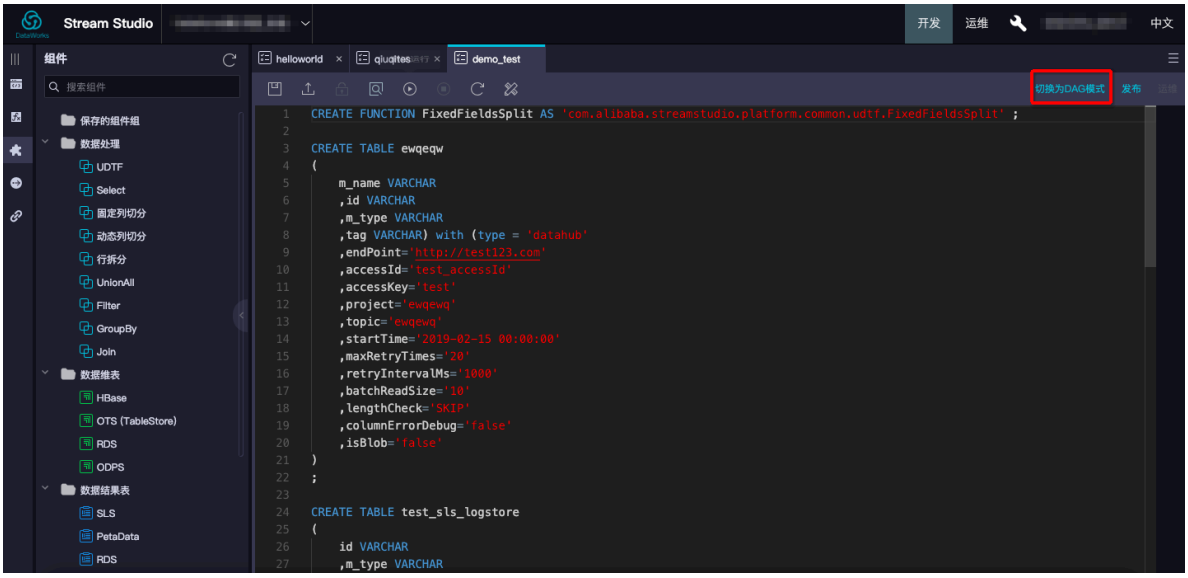
   - 组件区：在左侧导航条中切换到组件，在组件列表中列出了DAG模式可以使用的组件。
   - DAG画布区：右侧工作区为DAG画布，您可以将组件列表中的组件拖拽到DAG画布中，然后连接组件，最终构建一个DAG工作流程图。一个DAG工作流就是一个实时计算任务。
   - 组件参数配置区：在DAG画布中双击组件，在右侧会浮出组件参数配置面板，在这里对组件参数进行配置。若组件参数配置完整，则在组件右侧会显示绿色的对勾图标。如果参数配置不完整则会显示红色的错误图标。
   - 工具条区：在上方是工具条区，在这里可以进行的操作有：保存、提交、偷锁、预编译、测试运行、停止运行、重新加载和格式化。

   在DAG工作流编辑过程中，可以通过右键单击组件弹出右键菜单，进行更多操作：如重命名、删除、查看schema、查看错误信息、新建组件组和复制等。



2. DAG模式与SQL模式的转换。

配置好DAG工作流后，单击右上方的切换为SQL模式，即可将DAG转为SQL。

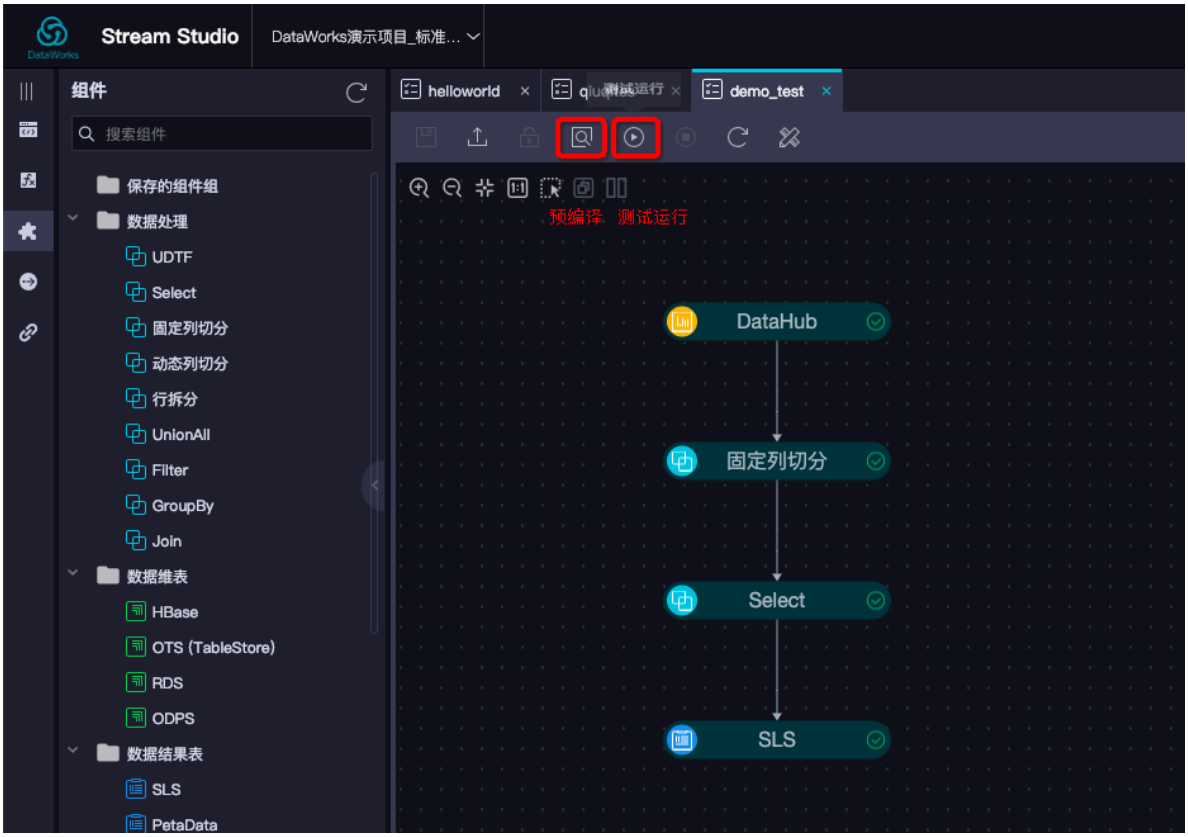


该功能能够将DAG工作流100%转为Flink SQL。在SQL模式中，也可以单击切换为DAG模式，将SQL切换为DAG。

如果您比较喜欢编写SQL，也可以直接在SQL模式中进行SQL任务开发。由于Flink SQL的功能比DAG更为强大，因此可能部分SQL语句无法转成DAG。随着组件增加和完善，未来将支持Flink SQL所有特性都能够转为DAG工作流。

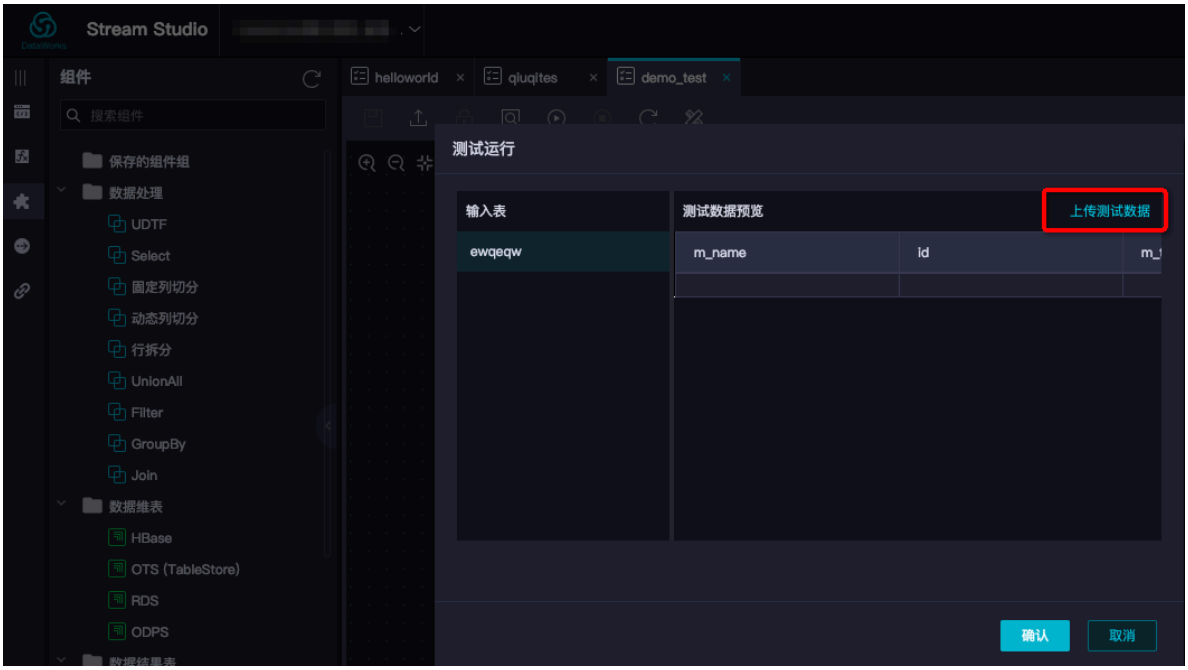
3. 预编译与测试运行。

编辑好DAG或SQL后，可以在工具条中单击预编译，对任务进行预编译，以提前进行错误检查。如果有错误，则会弹出错误信息，您可以根据错误信息对任务进行修改。



预编译通过后，可以进行测试运行，即本地调试。测试运行允许您上传一份样本数据对流任务进行本地测试。

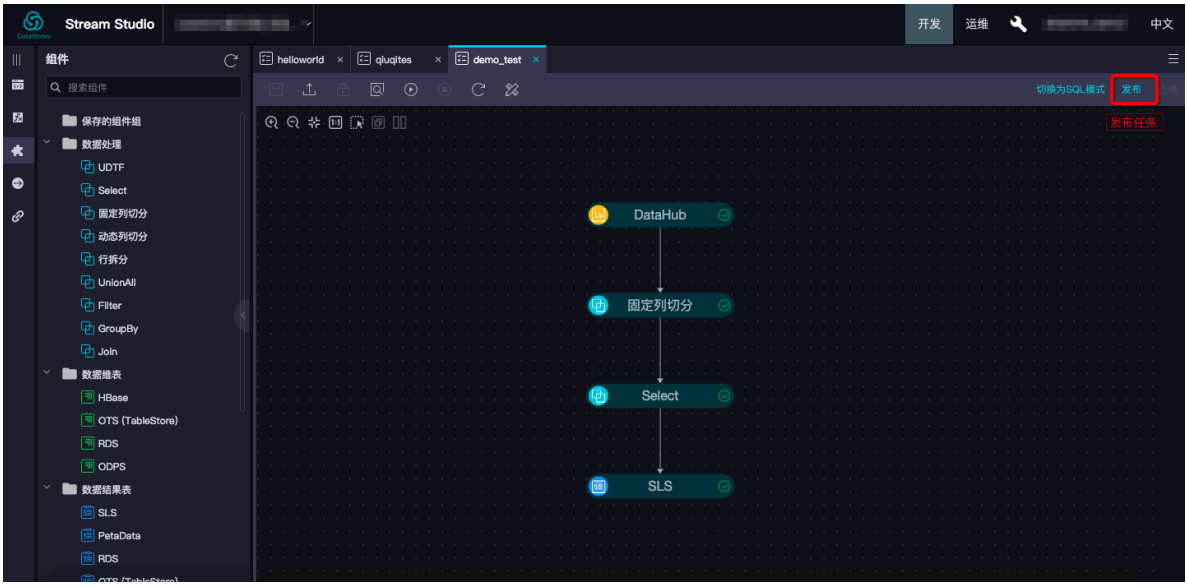
单击工具条中的测试运行按钮。在弹出的对话框中，针对任务中的每一个数据源表及数据维表都上传一份样本数据，然后单击确定，进行任务的本地测试。



如果运行成功，则会在下方显示运行结果。如果运行失败，请查看运行日志对任务进行修改。

4. 发布任务。

   如果任务编辑完成且测试运行通过，则可正式发布任务。单击工具条右边的发布，即可一键发布任务。



任务发布之后，可以进入运维页面，对任务进行启动、暂停、恢复和下线等操作等操作，详情请参见任务运维。

12.6 组件配置

12.6.1 数据源表

12.6.1.1 DataHub

DataHub作为一个流式数据总线，为阿里云数加平台提供了大数据的入口服务。

实时计算通常使用DataHub作为流式数据存储头和输出目的端。同时，上游众多流式数据，包括DTS、IOT等均选择DataHub作为大数据平台的数据入口。DataHub本身是流数据存储，实时计算只能将其作为流式数据输入。更多信息请参见创建数据总线DataHub源表。

配置面板说明

参数	注释说明	备注
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他血缘表重名。	无
定义列	要读取的DataHub内容字段列表，以及属性字段列表。	无
消费端点信息	消费端点信息。	无
读取的AccessID	读取的AccessID。	无
读取的秘匙	读取的秘匙。	无
读取的项目	读取的项目。	无
topic	项目下的具体的topic。	无
日志开始时间	日志开始时间。	格式为yyyy-MM-dd hh:mm:ss
读取最大尝试次数	可以尝试读取的最大次数。	无
重试间隔	重试间隔。	无
单次读取条数	单次读取条数。	无
单行字段条数检查策略	单行字段条数检查策略。	默认为SKIP，其它可选值为EXCEPTION和PAD · SKIP：字段数目不符合时跳过 · EXCEPTION：字段数目不符合时抛出异常 · PAD：按顺序填充，不存在的置为null
调试开关	如果打开调试开关，会打印出解析异常的日志。	无

参数	注释说明	备注
DataHub是否为BLOB类型	DataHub是否为BLOB类型。	无
数据质量	跳转至数据质量页面，查看相关监控任务。	无

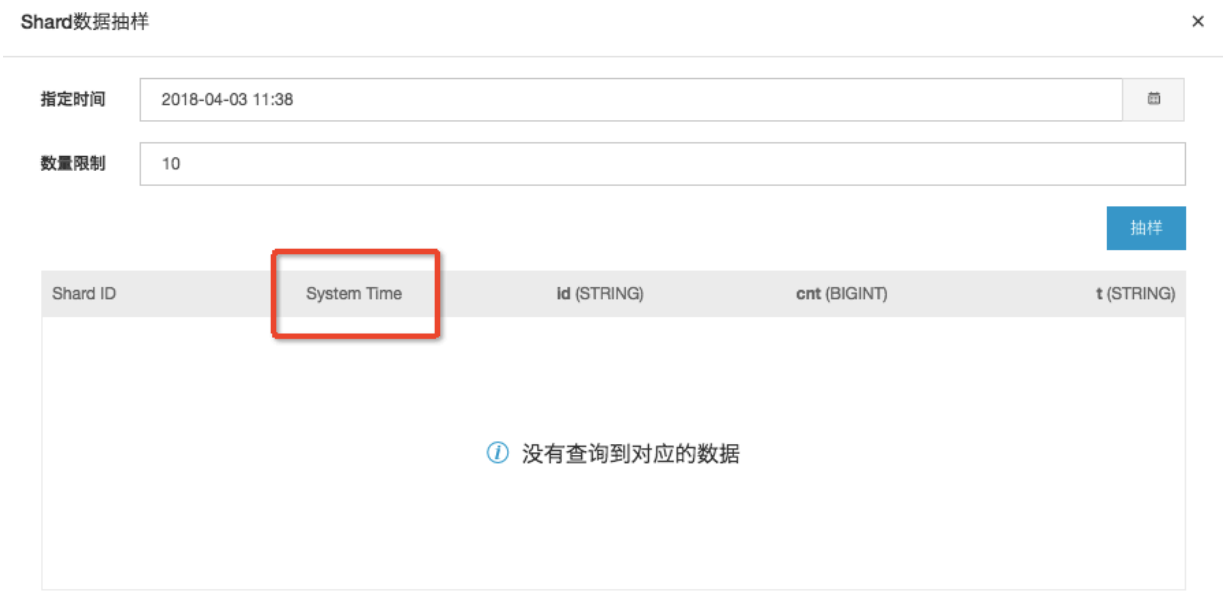
类型映射

DataHub和实时计算字段类型对应关系，建议您使用该对应关系进行DDL声明。

DataHub字段类型	实时计算字段类型
BIGINT	BIGINT
DOUBLE	DOUBLE
TIMESTAMP	BIGINT
BOOLEAN	BOOLEAN
DECIMAL	DECIMAL

属性字段

列定义支持获取DataHub的属性字段，可以记录每条信息写入DataHub的系统时间。



字段名	注释说明
System Time	每条记录入DataHub的System Time。

12.6.1.2 Kafka

实时计算Kafka源表声明基于云栖社区的kafka版本实现。Kafka源表数据解析流程为Kafka Source Table > UDTF > Flink > SINK，从Kakfa读入的数据均为VARBINARY（二进制）格式。读入的每条数据，都需要用UDTF将其解析为格式化数据。

更多详情请参见创建消息队列Kafka源表。

配置面板说明

参数	注释说明	备注
血缘表名	任务中表的唯一标志，不能与任务中其他血缘表重名。	无。
定义列	定义要读取的kafka字段列表。	kafka的列定义必须依次为messageKey varbinary、message varbinary、topic varchar、partition int和offset bigint，顺序不可更改。
kafka对应版本	kafka对应版本。	无。
读取的单个topic	读取的单个topic。	可选项包括kafka08、kafka09、kafka10和kafka11。
读取一批topic的表达式	读取一批topic的表达式。	无。
启动位点	启动位点。	· EARLISET从kafka最早分区开始读取数据。 · Group_OFFSETS根据Group读取数据。 · LATEST从kafka最新位点开始读取数据。 · TIMESTAMP从指定的时间点开始读取数据（kafk010、kafka011支持）。
定时检查是否有新分区产生	检查是否有新分区产生的时间间隔，单位为ms。	无。
group.id	消费组的ID。	可选。
zk链接地址	zk链接地址。	适用于kafka08。
kafka集群地址	kafka集群地址。	适用于kafka09、kafka10和kafka11。

参数	注释说明	备注
添加扩展参数	添加kafka支持的可选配置项。	无。

可选扩展参数

- kafka08

```
"consumer.id","socket.timeout.ms","fetch.message.max.bytes","num.
consumer.fetchers","auto.commit.enable","auto.commit.interval.ms","
queued.max.message.chunks", "rebalance.max.retries","fetch.min.bytes
","fetch.wait.max.ms","rebalance.backoff.ms","refresh.leader.backoff
.ms","auto.offset.reset","consumer.timeout.ms","exclude.internal
.topics","partition.assignment.strategy","client.id","zookeeper.
session.timeout.ms","zookeeper.connection.timeout.ms","zookeeper
.sync.time.ms","offsets.storage","offsets.channel.backoff.ms","
offsets.channel.socket.timeout.ms","offsets.commit.max.retries","
dual.commit.enabled","partition.assignment.strategy","socket.receive
.buffer.bytes","fetch.min.bytes"
```

- kafka09
- kafka010
- kafka011

kafka版本对应关系

type	kafka版本
kafka08	0.8.22
kafka09	0.9.0.1
kafka010	0.10.2.1
kafka011	0.11.0.2

12.6.1.3 MQ

消息队列（Message Queue，简称MQ）是阿里云商用的专业消息中间件、企业级互联网架构的核心产品，基于高可用分布式集群技术，搭建了包括发布订阅、消息轨迹、资源统计、定时（延时）、监控报警等一套完整的消息云服务。

MQ的更多信息请参见创建消息队列MQ源表。

配置面板说明

参数	注释说明	备注
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他血缘表重名。	无

参数	注释说明	备注
定义列	定义要读取的MetaQ字段列表。	无
topic	项目下的具体的topic。	无
订阅消费group名	订阅消费group名。	无
拉取时间间隔	拉取时间间隔，单位为毫秒。	无
消息消费启动的时间点	消息消费启动的时间点，为可选项。	无
跨单元访问app所在单元	跨单元访问时，需指明app所在单元。	无
订阅的标签	订阅的标签。	多个tag时使用，为可选项。
解析message body时的行分隔符	解析message body时的行分隔符。	无
字段分隔符	字段分隔符。	支持以\u开头，接四位十六进制数字表示任意unicode字符，例如\u0001表示Crtl+A。
编码格式	编码格式。	无
单行字段条数检查策略	单行字段条数的检查策略。	· SKIP：字段数目不符合时跳过。 · EXCEPTION：字段数目不符合时抛出异常。 · PAD：按顺序填充，不存在的置为null。

类型映射

MQ字段类型	实时计算字段类型
STRING	VARCHAR

注意事项

MQ实际上是非结构化存储格式，对于数据的Schema不提供强制定义，完全由业务层指定。目前实时计算支持类CSV格式和二进制格式。

12.6.1.4 Log Service（SLS）

日志服务（Log Service，简称LOG/原SLS）是针对实时数据的一站式服务，无需开发即可完成数据采集、消费、投递以及查询分析等功能，帮助提升运维、运营效率，建立DT时代海量日志处理能力。

日志服务本身是流数据存储，实时计算可以将其作为流式数据输入。更多信息请参见创建日志服务LOG源表。

日志服务的数据格式和JSON一致，示例如下。

```
{
    "a": 1000,
    "b": 1234,
    "c": "li"
}
```

配置面板说明

参数	注释说明	备注
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他血缘表重名。	无
列定义	要读取的Log Service内容字段和属性字段列表。	无
消费端点信息	消费端点信息。	#unique_581
AccessKey ID	读取的AccessKey ID。	无
AccessKey Secret	读取的AccessKey Secret。	无
读取的SLS项目	读取的SLS项目。	无
LogStore	项目下的具体的LogStore。	无
消费组名	消费组名。	您可以自定义消费组名（无固定格式）
消费日志开始的时间点	消费日志开始的时间点。	无
消费客户端心跳间隔时间	消费客户端心跳间隔时间，单位ms，为可选项。	无
读取最大尝试次数	可以尝试读取的最大次数。	无
调试开关	如果打开调试开关，会打印出解析异常的日志。	无

类型映射

日志服务和实时计算字段类型对应关系，建议您使用该对应关系进行DDL声明。

日志服务字段类型	实时计算字段类型
STRING	VARCHAR

属性字段

目前默认支持三个属性字段的获取，也支持其他自定义写入的字段。

字段名	注释说明
`__source__`	消息源
`__topic__`	消息主题
`__timestamp__`	日志时间

说明:

- SLS暂不支持MAP类型的数据。
- 字段顺序支持无序（建议字段顺序和表中定义一致）。
- 输入数据源为JSON形式时，注意定义分隔符，并且需要采用内置函数分析JSON_VALUE，否则就会解析失败，报错如下。

```
2017-12-25 15:24:43,467 WARN [Topology-0 (1/1)] com.alibaba.blink
.streaming.connectors.common.source.parse.DefaultSourceCollector
 - Field missing error, table column number: 3, data column number
: 3, data filed number: 1, data: [{"lg_order_code":"LP00000005","
activity_code":"TEST_CODE1","occur_time":"2017-12-10 00:00:01"}]
```

- batchGetSize设置不能超过1,000，否则会报错。
- batchGetSize指明的是logGroup获取的数量。如果单条logItem的大小和batchGetSize都很大，可能会导致频繁的GC，此时需要调小该参数。

12.6.2 数据处理

12.6.2.1 固定列分割

固定列分割是指按照特定分隔符来分割固定的列。

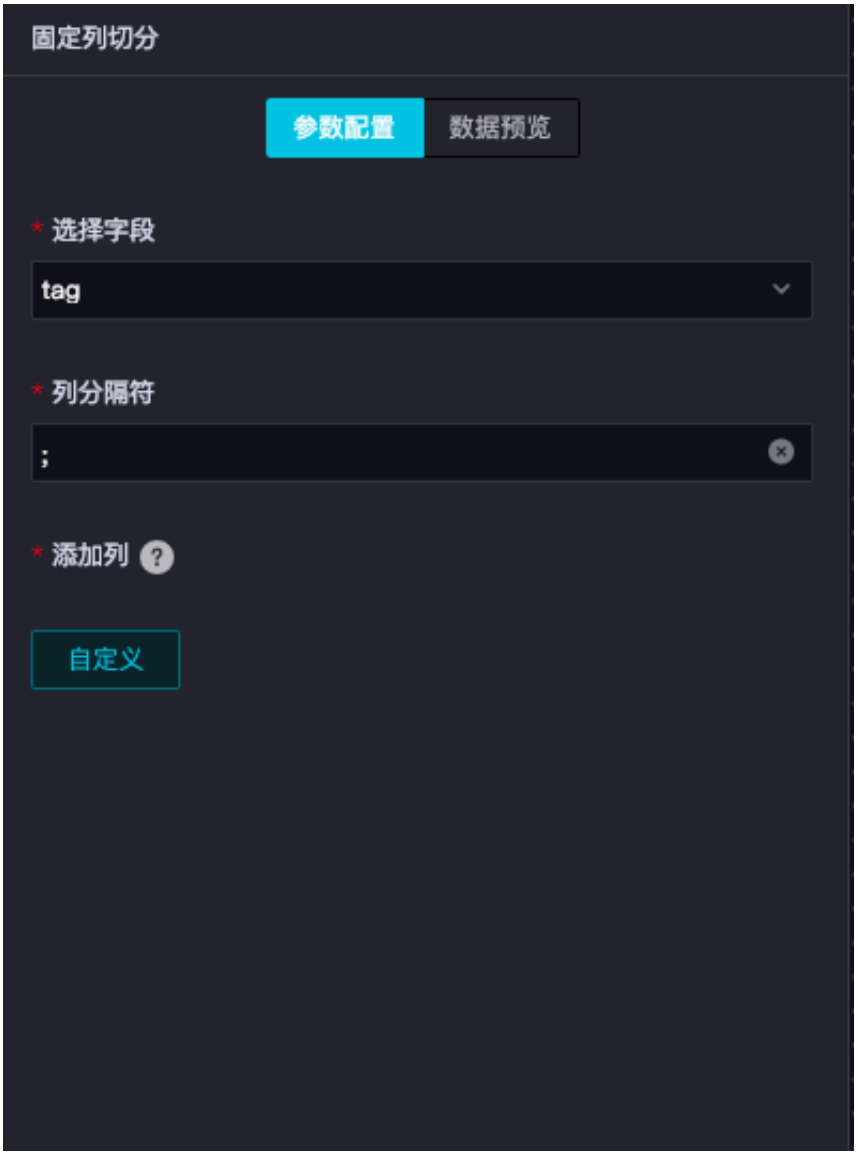
使用示例

您可以使用固定列分割功能，将如下日志按照（,）分割为4列：1111、2222、3333和4444。

```
1111,2222,3333,4444
```

固定列分割适用于列是按指定分隔符固定排列的日志。

配置面板说明



参数	注释说明
选择字段	选择要进行分割的字段名。
列分隔符	填写列分隔符，注意英文和中文。
添加列	添加您需要对外输出的列，此处需要指定列名（即key）和序号，并且可以定义别名。

12.6.2.2 动态列分割

动态列分割适用于当原始日志的列不是固定的，而是动态的情况。

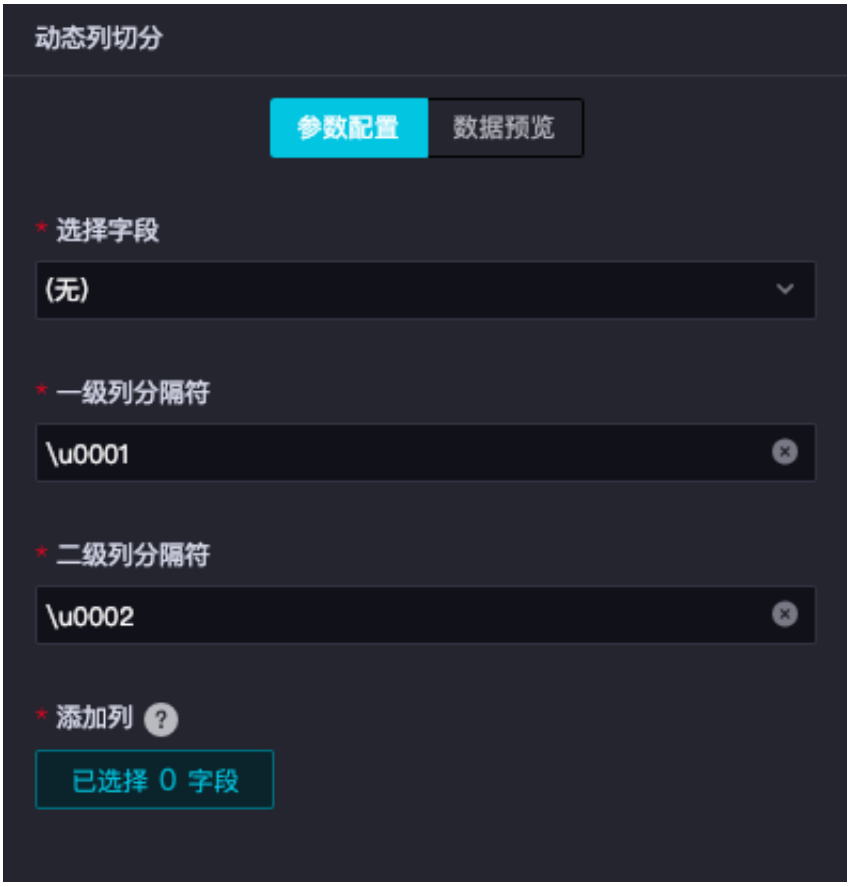
使用示例

例如原始日志为：

```
k1=v1,k2=v2,k3=v3,k4=v4
```

上述示例中的日志是按Key=value键值对的形式存储，每一条所拥有的键值对数量可能会不一样，即列的数量不是固定的。动态列分割可以对此种类型的日志先按一级分隔符（如上例中为“,”）分割出每一个键值对，然后按二级分隔符（如上例中为“=”）分割出key和value。

配置面板说明



参数	说明
选择字段	选择要进行分割的字段名。
一级列分隔符	一级列分隔符，默认为\u0001。
二级列分隔符	二级列分隔符，默认为\u0002。
添加列	添加您需要对外输出的列，此处需要指定列名（即key），并且可以定义别名。

12.6.2.3 行拆分

行拆分是指将一行数据中的某一个字段，按照指定分隔符拆分为多行。

使用示例

原始数据如下所示：

id	num
1	1,2

现将num字段按照（,）拆分为多行，并把拆分后的数据放在新增的new_num列中，则输出结果数据如下所示：

id	num	new_num
1	1,2	1
1	1,2	2

配置面板说明

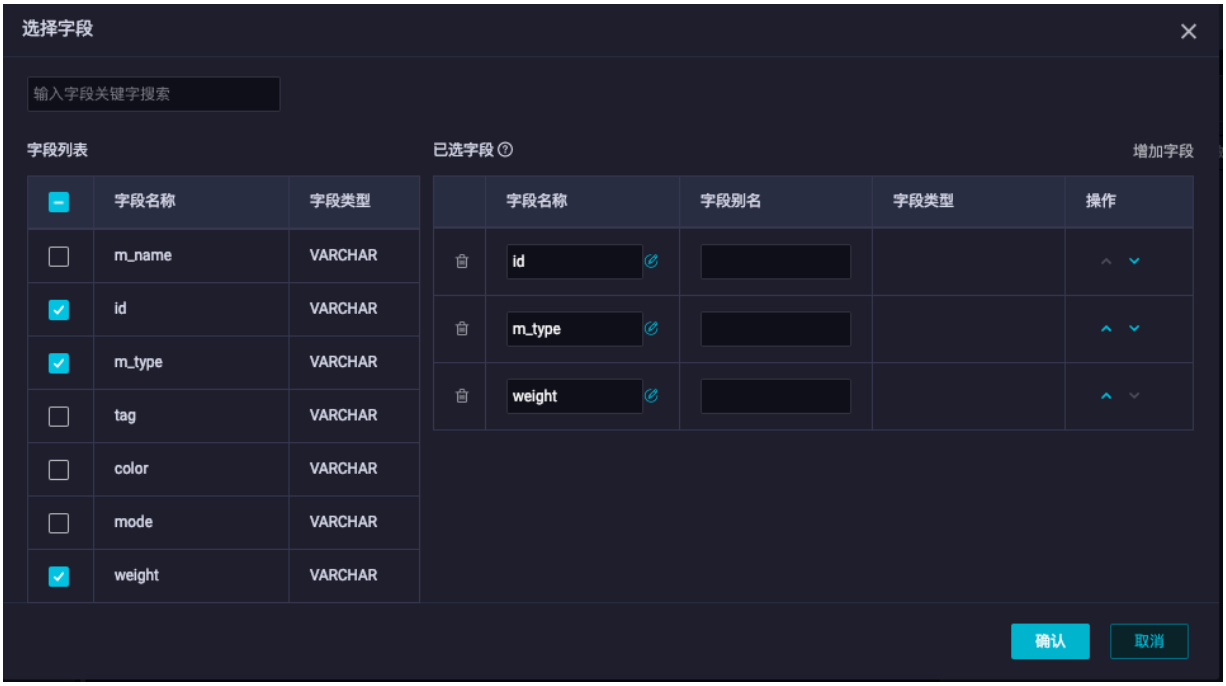
参数	注释说明	备注
选择字段	选择要进行拆分的字段名。	上例中为num。
字段分隔符	填写行拆分的分隔符，默认为（\n）。	上例中为（,）。
添加列	定义存放拆分结果的列名称，此处给出一个新字段名称的定义即可。	上例中为new_num。

12.6.2.4 Select

Select组件即SQL中的Select操作，选择要输出的字段，并且支持字段表达式。

配置面板说明

Select组件的配置为一个选择字段对话框。



您可以在字段列表中选择要输出的字段名，也可以定义字段别名。如果要对字段进行表达式运算，可以单击字段名称后面的按钮，在弹出的输入框中输入SQL表达式。

12.6.2.5 Filter

Filter过滤器，即SQL中的Where子句。

配置面板说明

Filter组件的配置简单，您只需填写一个表达式即可。Filter表达式支持函数以及运算符（=、<>、>、>=、<、<=），参考格式：`city = 'Beijing'`。

12.6.2.6 GroupBy

GroupBy组件即SQL中的Group By子句。

配置面板说明

参数	注释说明
选择分组字段	要Group By的字段，支持选择多个
选择输出字段	要输出的字段，即需要Select的字段，配置方式同Select组件

12.6.2.7 Join

Join组件即SQL中的Join子句。

Join子句的详情请参见Join。

配置面板说明

参数	注释说明
Join模式	选择要采用的JOIN模式，支持INNER JOIN、LEFT OUTER JOIN、RIGHT OUTER JOIN、FULL OUTER JOIN。
表达式	JOIN表达式，只支持等值连接，不支持不等连接，参考格式：leftId = rightId AND limit = 0。
选择字段	选择要输出的字段，相当于要Select的字段。

12.6.2.8 UnionAll

UnionAll组件即SQL中的Union All子句。

配置面板说明

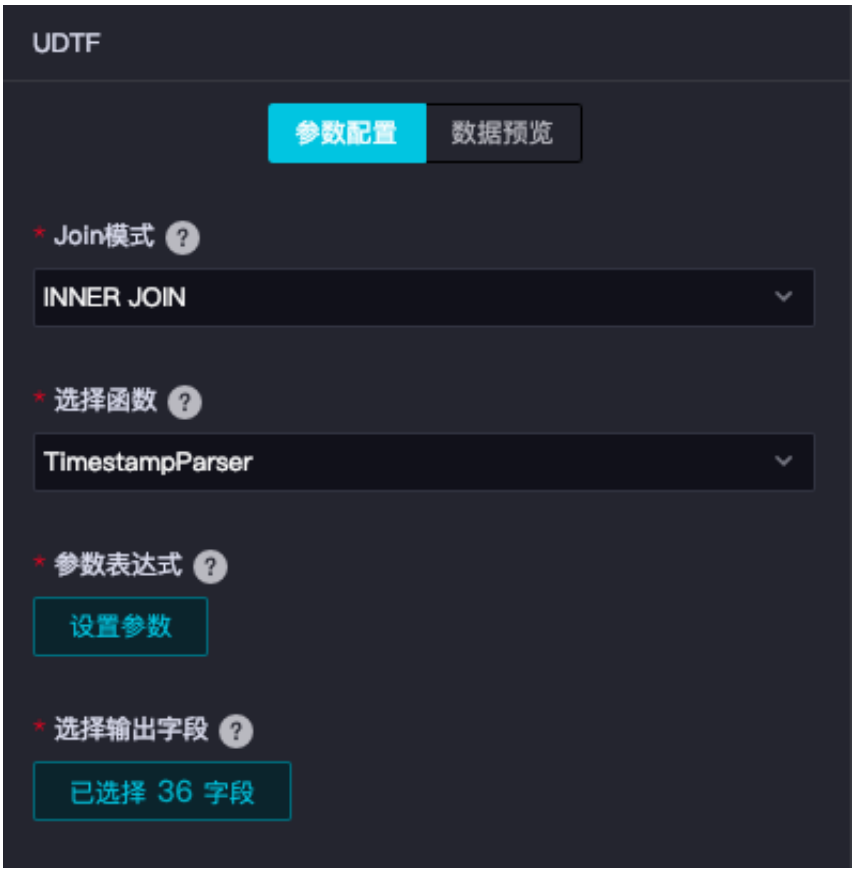
UnionAll组件无需配置任何参数。

12.6.2.9 UDTF

UDTF组件即自定义函数组件，相当于SQL中的UDTF子句。

> 说明:
> 仅实时计算服务的独享模式支持UDTF组件，详情请参见#unique_596。

配置面板说明



参数	注释说明
JOIN模式	支持INNER JOIN和LEFT OUTER JOIN两种模式。 · INNER JOIN：UDTF有结果才返回。 · LEFT OUTER JOIN：UDTF无结果补NULL。
选择函数	选择函数名称。您需要先在资源引用中上传程序包，然后在资源引用面板中勾选程序包，以表明当前任务引用了此程序包。
参数表达式	选择函数后填写对应的输入、输出参数。
选择输出字段	选择要输出的字段，此处可以定义字段名、字段别名和字段表达式。

12.6.3 数据维表

12.6.3.1 HBase

本文将为您介绍实时计算云数据库HBase版维表的配置面板说明。

更多详情请参见实时计算云数据库（HBase）维表。

配置面板说明

参数	注释说明	备注
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他血缘表重名。	无
表名	HBase表名。	无
数据同步	从元数据中心读取指定表名的元数据，帮助填充输出字段和其他元信息表单项。	无
列族名	列族名。	目前仅支持插入同一列族。
Rowkey字段名	指定输出字段中作为主键的字段。	无
HBase集群对应的ZooKeeper地址	HBase集群配置的ZooKeeper地址，以（,）分隔主机列表。	可以在*hbase-site.xml*文件中找到*hbase.zookeeper.quorum*相关配置。
HBase集群配置在zk上的路径	HBase集群配置在ZooKeeper的路径。	可以在*hbase-site.xml*文件中找到*hbase.zookeeper.quorum*的相关配置。
选择输出字段	要输出到HBase表的字段列表。	无
缓存策略	缓存策略。	包括None、LRU和ALL三种缓存策略。
缓存大小	缓存大小。	选择LRU缓存策略后，可以设置缓存大小。
缓存超时时间	缓存超时时间，单位为毫秒。	· 选择LRU缓存策略后，可以设置缓存失效的超时时间，默认不过期。 · 选择ALL策略，则为缓存reload的间隔时间，默认不重新加载。

参数	注释说明	备注
更新时间黑名单	缓存策略选择ALL时启用。更新时间黑名单，防止在此时间内进行cache更新（例如双11场景）。	自定义输入格式如下所示： `2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00` 用逗号（,）分隔多个黑名单，用箭头（->）分隔黑名单的起始结束时间。
cacheScanLimit	缓存策略选择ALL时启用。load全量HBase数据，服务端一次RPC返回给客户端的行数。	无
用户名	用户名。	无
密码	密码。	无
最大尝试次数	最大尝试次数。	默认10次。
partitionedJoin	设置为true后，会用joinKey进行分区，将数据分发到Join节点，提高缓存命中率。	可选，默认关闭。
shuffleEmptyKey	设置为true后，遇到空key会随机往下游进行shuffle，否则往0号下游发。	建议打开。

说明

- 声明维表时，必须要指名主键。维表JOIN的时候，ON的条件必须包含所有主键的等值条件，HBase中的主键即rowkey。
- 目前RDS和DRDS提供None（无缓存）、LRU（最近使用策略缓存）和ALL三种缓存策略。需要配置相关参数：缓存大小（cacheSizecacheTTLMs、cacheTTLMs和cacheReloadTimeBlackList）。
- 使用ALL Cache时，由于会进行异步Reload，需要给维表JOIN的节点增加内存，增加的内存大小为远程表数据量的2倍。

12.6.3.2 TableStore（OTS）

表格存储（Table Store，简称OTS）是根据高可用和数据可靠性的设计标准，构建在阿里云飞天分布式系统之上的分布式NoSQL数据存储服务。

更多详情请参见创建表格存储TableStore维表。

配置面板说明

参数	注释说明
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他血缘表重名。
选择输出字段	要输出至Table Store表的字段列表。
实例名	实例名。
表名	表名。
数据同步	从元数据中心读取指定表名的元数据，帮助填充输出字段和其他元信息表单项。
选择输出字段	从维表中读取输出到下游组件的字段列表。
实例访问地址	实例访问地址。
AccessKey ID	读取的AccessKey ID。
AccessKey Secret	读取的AccessKey Secret。
缓存策略	可选LRU。
缓存大小	当选择LRU缓存策略后，可以设置缓存大小。
缓存超时时间	缓存超时时间，单位为毫秒。
primaryKey	指定输出字段中作为主键的字段。

12.6.3.3 RDS

阿里云关系型数据库（Relational Database Service，简称RDS）是一种稳定可靠、可弹性伸缩的在线数据库服务，详情请参见#unique_603。

配置面板说明

参数	注释说明	备注
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他表血缘表名重名。	无
地址	地址	地址请参见RDS的URL地址。
表名	表名	无
用户名	用户名	无
密码	密码	无
数据集成同步	从元数据中心读取指定表名的元数据，帮助填充输出字段和其他元信息表单项。	无

参数	注释说明	备注
选择输出字段	从维表中读取输出到下游组件的字段列表。	无
最大尝试插入次数	最大尝试插入次数	无
缓存策略	缓存策略	可选None、LRU或ALL。
缓存大小	缓存大小	当选择`LRU`缓存策略后，可以设置缓存大小。
缓存超时时间	缓存超时时间，单位毫秒。	当选择`LRU`缓存策略后，可以设置缓存失效的超时时间，默认不过期。当选择`ALL`策略，则为缓存reload的间隔时间，默认不重新加载。
更新时间黑名单	缓存策略选择`ALL`时启用。更新时间黑名单，防止在此时间内做cache更新（例如双11场景）。	自定义输入格式为 `2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00` 。用逗号`,`来分隔多个黑名单，用箭头`->`来分割黑名单的起始结束时间
primaryKey	指定输出字段中作为主键的字段。	声明一个维表时，必须要指名主键。维表JOIN的时候，ON的条件必须包含所有主键的等值条件。RDS或DRDS的主键可以定义为表的主键或是唯一索引列。

说明

· 目前RDS/DRDS提供如下三种缓存策略：

- None：无缓存。
- LRU：最近使用策略缓存。需要配置相关参数：缓存大小（cacheSize）和缓存超时时间（cacheTTLMs）。
- ALL：全量缓存策略。

Job运行前，会将远程表中所有数据load到内存中，之后所有的维表查询都会通过cache进行。cache命中不到则不存在数据，并在缓存过期后重新加载一遍全量缓存。全量缓存策略适合远程表数据量小、miss key多的场景。全量缓存相关配置：缓存更新间隔（cacheTTLMs），更新时间黑名单（cacheReloadTimeBlackList）。

- 由于异步reload，使用cache all时，需要将维表JOIN的节点增加一些内存，增加的内存大小为远程表两倍的数据量。
- 使用cache all，请特别注意节点的内存，防止内存溢出。

12.6.3.4 MaxCompute（ODPS）

本文将为您介绍实时计算MaxCompute维表的配置面板说明、类型映射和METRIC信息。

更多详情请参见#unique_606。

配置面板说明

参数	注释说明	备注
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他血缘表重名。	无
表名	表名。	无
project	项目。	无
accessId	用户访问ID。	无
accessKey	用户访问密码。	无
和元数据中心同步	从元数据中心读取指定表名的元数据，帮助填充输出字段和其他元信息表单项。	无
选择输出字段	从维表中读取输出到下游组件的字段列表。	无
分区名	分区名。	无
加载最大表限制	加载最大表限制。	加载MaxCompute表最大记录数。
缓存策略	缓存策略。	包括None、LRU和ALL三种缓存策略。
缓存大小	缓存大小。	当选择LRU缓存策略后，可以设置缓存大小，MaxCompute默认缓存值为100,000行。
缓存超时时间	缓存超时时间，单位为毫秒。	· 选择LRU缓存策略后，可以设置缓存失效的超时时间，默认不过期。 · 选择ALL策略，则为缓存reload的间隔时间，默认不重新加载。

参数	注释说明	备注
更新时间黑名单	缓存策略选择ALL时启用。更新时间黑名单，防止在此时间内进行cache更新（例如双11场景）。	可选，默认为空，格式如下： `2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00` 用逗号（,）分隔多个黑名单，用箭头（->）分隔黑名单的起始结束时间。
cacheScanLimit	缓存策略选择ALL时启用。load全量HBase数据，服务端一次RPC返回给客户端的行数。	无
primaryKey	指定输出字段中作为主键的字段。	无

类型映射

MaxCompute	Blink
TINYINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE
BOOLEAN	BOOLEAN
DATETIME	TIMESTAMP
TIMESTAMP	TIMESTAMP
VARCHAR	VARCHAR
STRING	STRING
DECIMAL	DECIMAL
BINARY	VARBINARY

METRIC

维表JOIN可以查看关联率，缓存命中率等METRIC信息。可以通过kmonitor查看。

查询语句	说明
fetch qps	查询维表总qps，包括命中和不命中。对应的Metric Name：`blink.projectName.jobName.dimJoin.fetchQPS`。
fetchHitQPS	维表命中qps，包括换成命中和查询物理维表命中，对应的Metric Name：`blink.projectName.jobName.dimJoin.fetchHitQPS`。
cacheHitQPS	维表缓存命中qps，对应的Metric Name：`blink.projectName.jobName.dimJoin.cacheHitQPS`。
dimJoin.fetchHit	维表关联率，对应的Metric Name：`blink.projectName.jobName.dimJoin.fetchHit`。
dimJoin.cacheHit	维表缓存关联率，对应的Metric Name：`blink.projectName.jobName.dimJoin.cacheHit`。

说明

- 推荐使用实时计算2.1.1及以上版本。
- 使用MaxCompute表作为维表，需要先给MaxCompute账号赋予读权限。
- 声明维表时，必须要指名主键，维表JOIN的时候，ON的条件必须包含所有主键的等值条件。
- MaxCompute维表主键必须有唯一性，否则会被去重。
- 如果是分区表，目前不支持将分区列写入到schema定义中。
- 使用ALL Cache时，由于会进行异步Reload，需要给维表JOIN的节点增加内存，增加的内存大小为远程表数据量的2倍。
- 如果任务出现如下Failover：

```
RejectedExecutionException: Task
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTas
```

通常是由于实时计算1.x版本中维表JOIN存在一定的问题，推荐升级到2.1.1及以上实时计算版本。如果要继续使用原有版本，建议对作业进行暂停操作，根据Failover History中第一次出现failover的具体报错信息进行排查。

12.6.4 数据结果表

12.6.4.1 Log Service (SLS)

日志服务（Log Service）简称LOG，是针对实时数据一站式服务，无需开发就能快捷完成数据采集、消费、投递以及查询分析等功能，帮助提升运维、运营效率，建立DT（Data Technology）时代海量日志处理能力。更多信息请参考#unique_609。

配置面板说明

参数	注释说明	备注
血缘表名(任务唯一)	任务中表的唯一标志，不能与任务中其他表血缘表名重名	无
选择输出字段	输出到log service的字段列表	无
endPoint地址	endPoint地址	#unique_581
项目名	项目名	无
topic表名	topic表名	无
source	日志的来源地，例如产生该日志机器的IP地址。	无
accessId	accessId	无
accessKey	accessKey	无
写入方式	写入方式	可选，默认为`random`模式，选择`partition`则会按分区写入。
分区列	分区列	如果`mode`为`partition`则必填。
logStore	Project下面具体的logStore	无

12.6.4.2 PetaData

云数据库HybridDB for MySQL（原名PetaData）是同时支持在线事务（OLTP）和在线分析（OLAP）的关系型HTAP类数据库。

HTAP是Hybrid Transaction/Analytical Processing的简写，意为将数据的事务处理（TP）与分析（AP）混合处理，从而实现对数据的实时处理分析HybridDB for MySQL兼容MySQL的语法及函数，并且增加了对Oracle常用分析函数的支持，完全兼容TPC-H和TPC-DS测试标准。更多信息请参见创建云数据库HybridDB for MySQL结果表。

配置面板说明

参数	注释说明	备注
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他表血缘表名重名	无
选择输出字段	要输出到云数据库表的字段列表	无
地址	地址	#unique_612
表名	表名	无
用户名	用户名	无
密码	密码	无
最大尝试插入次数	最大尝试插入次数	无
每次写的批次大小	每次写的批次大小	表示每次写多少条
去重的buffer大小，需要指定主键才生效	去重的buffer大小，需要指定主键才生效	无
写超时时间	写超时时间，单位ms	表示数据超过了指定ms，还没有写过，就会将缓存的数据写入一次
是否忽略delete操作	是否忽略delete操作	无
primaryKey	指定输出字段中作为主键的字段	可多选

说明:

- 本文档仅适用于独享模式。
- 实时计算写入PetaData数据库结果表原理：针对实时计算每行结果数据，拼接成一行SQL向目标端数据库进行执行。
- bufferSize默认值是1000，如果到达bufferSize阈值（buffer hashmap size），也会触发写出。因此您配置batchSize的同时还需要配上bufferSize。bufferSize和batchSize大小相同即可。
- 建议设置batchSize='4096'，bathcSize数值不建议设置过大。

12.6.4.3 RDS

阿里云关系型数据库（Relational Database Service，简称RDS）是一种稳定可靠、可弹性伸缩的在线数据库服务。详情请参见#unique_614。

配置面板说明

参数	注释说明	备注
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他表血缘表名重名。	无
选择输出字段	要输出到RDS表的字段列表。	无
地址	地址	地址请参见RDS的URL地址。
表名	表名	无
用户名	用户名	无
密码	密码	无
最大尝试插入次数	最大尝试插入次数	无
每次写的批次大小	每次写的批次大小	表示每次写多少条。
去重的buffer大小	去重的buffer大小，需要指定主键才生效。	表示输入的数据达到指定条就开始输出。
写超时时间	写超时时间，单位ms。	表示数据超过了指定ms，还没有写过，就会将缓存的数据都写一次。
excludeUpdateColumns	对相同key的值更新时排除掉相应的column。	无
是否忽略delete操作	是否忽略delete操作。	无
partitionBy	写入Sink节点前，会根据该值进行hash，数据会流向相应的hash节点。	无

类型映射

RDS字段类型	实时计算字段类型
text	varchar
byte	varchar
integer	int
long	bigint
double	double

RDS字段类型	实时计算字段类型
date	varchar
datetime	varchar
timestamp	varchar
time	varchar
year	varchar
float	float
decimal	decimal
char	varchar

JDBC 连接参数

参数名称	参数说明	缺省值	最低版本要求
useUnicode	是否使用Unicode字符集，如果参数characterEncoding设置为gb2312或gbk，本参数值必须设置为true。	false	1.1g
characterEncoding	当useUnicode设置为true时，指定字符编码。比如可设置为gb2312或gbk。	false	1.1g
autoReconnect	当数据库连接异常中断时，是否自动重新连接。	false	1.1
autoReconnectForPools	是否使用针对数据库连接池的重连策略。	false	3.1.3
failOverReadOnly	自动重连成功后，连接是否设置为只读。	true	3.0.12
maxReconnects	autoReconnect设置为true时，重试连接的次数。	3	1.1
initialTimeout	autoReconnect设置为true时，两次重连之间的时间间隔，单位：秒。	2	1.1

参数名称	参数说明	缺省值	最低版本要求
connectTimeout	和数据库服务器建立socket连接时的超时，单位：毫秒。 0表示永不超时，适用于JDK 1.4及更高版本。	0	3.0.1
socketTimeout	socket操作（读写）超时，单位：毫秒。 0表示永不超时。	0	3.0.1

FAQ

- Q：实时计算的结果数据写入RDS表，是按主键更新的，还是新生成一条记录?

  A：如果在DDL中定义了主键，会采用`insert into on duplicate key update`的方式更新记录，也就意味着对于不存在的主键字段会直接插入，存在的主键字段则更新相应的值。 如果DDL中没有声明primary key，则会用`insert into`方式插入记录，追加数据。

- Q：使用RDS表中的唯一索引进行GROUP BY，需要注意什么?

  A：RDS中只有一个自增主键，实时计算作业中不能声明为Primary Key。如果需要使用RDS表中的唯一索引进行GROUP BY，需要在作业中的Primary Key中声明这些唯一索引。

12.6.4.4 TableStore (OTS)

表格存储（Table Store，简称OTS）是构建在阿里云飞天分布式系统之上的分布式NoSQL数据存储服务。根据99.99%的高可用以及11个9的数据可靠性的标准设计。表格存储通过数据分片和负载均衡技术，实现数据规模与访问并发上的无缝扩展。提供海量结构化数据的存储和实时访问服务。

配置面板说明

参数	注释说明	备注
血缘表名(任务唯一)	任务中表的唯一标志，不能与任务中其他表血缘表名重名	无
选择输出字段	要输出到Table Store表的字段列表	无
实例名	实例名	无
表名	表名	无
实例访问地址	实例访问地址	参见 服务地址 。
访问的id	访问的id	无
访问的键	访问的键	无

参数	注释说明	备注
指定插入的字段列名，多个以逗号分割	指定插入的字段列名，多个字段以逗号分割	插入2个字段的输入为`'ID,NAME'`。
去重的buffer大小	去重的buffer大小	例如配置5000，表示输入的数据达到5000条就开始输出。
写超时时间	写超时时间，单位ms	表示如果超过了指定ms，还没有数据写入TableStore，则会将缓存的数据写入一次
每次写的批次大小	每次写的批次大小	无
重试间隔时间	重试间隔时间，单位ms	无
最大重试次数	最大重试次数	无
是否忽略delete操作	是否忽略delete操作	无

类型映射

OTS字段类型	实时计算字段类型
integer	bigint
string	varchar
boolean	boolean
double	double

12.6.4.5 MQ

消息队列（Message Queue）简称MQ，是阿里云商用的专业消息中间件，是企业级互联网架构的核心产品。消息列队是基于高可用分布式集群技术，搭建了包括发布订阅、消息轨迹、资源统计、定时（延时）、监控报警等一套完整的消息云服务。 实时计算可以将消息队列作为流式数据输入。更多信息请参见#unique_618。

配置面板说明

参数	注释说明	备注
血缘表名(任务唯一)	任务中表的唯一标志，不能与任务中其他表血缘表名重名	无
选择输出字段	要输出到DataHub表的字段列表	无
写入的MetaQ队列名	写入的Message Queue队列名	无

参数	注释说明	备注
写入的群组	地址	公共云内网接入#阿里云经典网络/VPC##华东1、华东2、华北1、华北2、华南1香港的区域endpoint的地址是#onsaddr-internal.aliyun.com:8080。公共云公网接入地址是：http://onsaddr-internet.aliyun.com/rocketmq/nsaddr4client-internet。
accessID	填写自己的ID	无
accessKey	填写自己的Key	无
写入的群组	写入的群组	无
写入的标签	写入的标签	无
字段分割符	字段分割符	支持以\u开头接四位十六进制数字表示任意unicode字符，例如\u0001表示Crtl+A
编码	编码	无
retryTimes	写入重试次数	无
sleepTimeMs	重试间隔时间，单位ms	无

12.6.4.6 DataHub

DataHub作为一款流式数据总线，为阿里云数加平台提供了大数据的入口服务。结合阿里云众多云产品，可以构建一站式的数据处理平台。实时计算通常使用DataHub作为流式数据存储输入源和输出目的端。更多信息请参考#unique_620。

配置面板说明

参数	注释说明	备注
血缘表名(任务唯一)	任务中表的唯一标志，不能与任务中其他表血缘表名重名	
选择输出字段	要输出到DataHub表的字段列表	
endPoint地址	endpoint地址	参见DataHub的Endpoint地址
项目名	项目名	无

参数	注释说明	备注
topic	topic表名	无
accessId	accessId	无
accessKey	accessKey	无
最大尝试插入次数	最大尝试插入次数	无
每次写的批次大小	每次写的批次大小	无
缓存数据的最大超时时间	缓存数据的最大超时时间，单位ms	无
每次写入的最大Block数	每次写入的最大Block数	无
数据质量	跳转数据质量中心查看相关监控任务	无

类型映射

DataHub和实时计算字段类型对应关系，强烈建议用户使用该对应关系进行DDL声明：

DataHub字段类型	实时计算字段类型
BIGINT	BIGINT
DOUBLE	DOUBLE
TIMESTAMP	BIGINT
BOOLEAN	BOOLEAN
DECIMAL	DECIMAL

12.6.4.7 MaxCompute (ODPS)

更多信息请参见#unique_622。

配置面板说明

参数	注释说明	备注
血缘表名(任务唯一)	任务中表的唯一标志，不能与任务中其他表血缘表名重名	无
表名	表名	无
选择输出字段	要输出到MaxCompute表的字段列表	无
endPoint	地址	参见#unique_165。
project	项目	无

参数	注释说明	备注
accessId	填写自己的ID	无
accessKey	填写自己的KEY	无
partition	写入分区	分区表必填，具体分区信息到数据管理查看。例如: 一个表的分区信息为`ds=20180905`，则可以写`partition` `` = 'ds=20180905'``。多级分区之间用逗号分隔，示例：partition`= 'ds=20180912,dt=xxxyyy'`

说明:

实时计算数据写入odps的方式是每次做checkPoint的时候将缓存数据进行输入。

类型映射

MaxCompute	Blink
TINYINT	TINYINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE
BOOLEAN	BOOLEAN
DATETIME	TIMESTAMP
TIMESTAMP	TIMESTAMP
VARCHAR	VARCHAR
STRING	STRING
DECIMAL	DECIMAL
BINARY	VARBINARY

常见问题

Q: Stream模式的ODPS Sink是否支持`isOverwrite`为`true`的情况?

A：`isOverwrite`为`true`，写入sink之前会把结果表或者结果数据清空。具体来说，就是每次启动后和暂停恢复后，写入之前会把原来结果表或者结果分区里的内容删除掉。流上暂停恢复后清空数据会导致丢数据。

12.6.4.8 HBase

更多信息请参见#unique_625。

配置面板介绍

参数	注释说明	备注
血缘表名(任务唯一)	任务中表的唯一标志，不能与任务中其他表血缘表名重名	无
选择输出字段	要输出到HBase表的字段列表	无
hbase集群配置的zk地址	HBase集群配置的zk地址	可以在`hbase-site.xml`文件中找到`hbase.zookeeper.quorum`相关配置。
集群配置在zk上的路径	集群配置在zk上的路径	可以在`hbase-site.xml`文件中找到`hbase.zookeeper.quorum`相关配置。
hbase 表名	HBase表名	无
列族名	列族名	无
用户名	用户名	无
密码	密码	无
partitionBy	设置为true之后会在用joinKey做partition，将数据分发到join节点，提高缓存命中率。	无
shuffleEmptyKey	设置为true之后遇到空key会随机往下游做shuffle，否则往0号下游发。	建议打开
插入尝试次数	插入尝试次数	无
流入多少条数据后进行去重	流入多少条数据后进行去重	无
批次写入大小	批次写入大小	无
最长插入时间	最长插入时间，单位ms	无
是否写入主键值	是否写入主键值	无
是否都按照string插入	是否都按照string插入	无

参数	注释说明	备注
rowKey的分隔符	rowKey的分隔符	无
是否为动态表	是否为动态表	无
primaryKey	从输出字段中选择主键字段	无

说明:

- 本文档只适用于独享模式。
- primary key支持定义多个字段。多个字段会按照rowkeyDelimiter（默认为:）拼接起来作为row_key。
- HBase做撤回删除操作时，如果column定义了多版本，会把所有版本的值清空。

12.6.4.9 ElasticSearch

本文将为您介绍ElasticSearch的配置面板说明。

配置面板说明

参数	注释说明	默认值
血缘表名（任务唯一）	任务中表的唯一标志，不能与任务中其他表血缘表名重名。	无
选择输出字段	要输出到ElasticSearch表的字段列表。	无
endPoint	server地址，例如http://127.0.0.1:9211。	无
accessId	访问实例ID。	无
accessKey	访问实例秘匙。	无
index	索引名称，类似于数据库的名称。	无
typeName	type名称，类似于数据库的表名称。	无
bufferSize	分批写入的记录条数。	1,000
maxRetryTimes	异常重试次数。	30
timeout	读超时，单位为毫秒。	600,000
discovery	是否开启节点发现。如果开启客户端会5分钟刷新一次服务列表。	false

参数	注释说明	默认值
compression	是否使用GZIP压缩request bodies。	true
multiThread	是否开启JestClient多线程。	true
ignoreWriteError	是否忽略写入异常。	false
settings	创建索引的settings配置。	无

说明:

- 本文档仅适用于独享模式。
- ES支持根据primaryKey进行update，primaryKey只能为1个字段。
- 指定primaryKey后，document的id为primaryKey字段的值。
- 未指定primaryKey的document对应的id为随机。
- full更新模式下，后面的doc会完全覆盖之前的doc，不会原地更新字段。
- 所有的更新默认为upsert语义，即insert or update。

更多信息请参见#unique_627。

12.7 任务运维

在Stream Studio顶部导航条中点击运维可进入任务运维页面。目前Stream Studio的任务运维功能直接对接原有实时计算开发平台（Bayes）。

任务运维可直接参考实时计算服务帮助文档中的数据运维部分。

12.8 Stream Studio常见问题

本文为您介绍使用Stream Studio可能会遇到的常见问题。

Q：使用Stream Studio之前需要开通什么计算引擎服务?

A：Stream Studio是开发平台，基于阿里云实时计算服务构建，因此需要首先开通实时计算服务。

Q：Stream Studio支持哪些集群模式的实时计算服务?

A：共享模式和独享模式都支持。

Q：Stream Studio在共享模式和独享模式的支持上是否有功能区别?

A：有区别。出于安全考虑，对于共享模式不支持UDF。独享模式支持UDF。如果你对性能和功能要求较高，建议采用独享模式。

Q：实时计算项目在哪里创建？如何与Stream Studio关联上？

A：实时计算项目在实时计算控制台中创建，创建好之后可以到DataWorks控制台的工作空间列表中绑定到已有工作空间中，或者直接创建新的工作空间并绑定实时计算项目。绑定好项目之后，就可以在Stream Studio中开发任务了。

Q：Stream Studio中的DAG开发模式有什么优势，与SQL有什么异同？

A：Stream Studio支持可视化DAG和SQL两种开发模式。使用DAG开发模式是所见即所得的，无需编写代码，以拖拽组件的方式就可以完成任务开发，简单快捷。同时DAG工作流可以与SQL互转。你可以自由选择。

Q：Stream Studio支持哪种类型的SQL？

A：阿里云实时计算引擎是基于Flink构建的，因此Stream Studio支持Flink SQL。

13 App Studio

13.1 App Studio概述

App Studio提供了丰富的前端组件，通过自由拖拽即可简单快速搭建前端应用。

App Studio是一款数据产品的开发工具，您无需下载安装本地IDE或配置维护环境变量，只需一个浏览器即可编写、运行和调试应用程序，体验和本地IDE一样的编程效果，并且可以在线发布应用。

产品优势

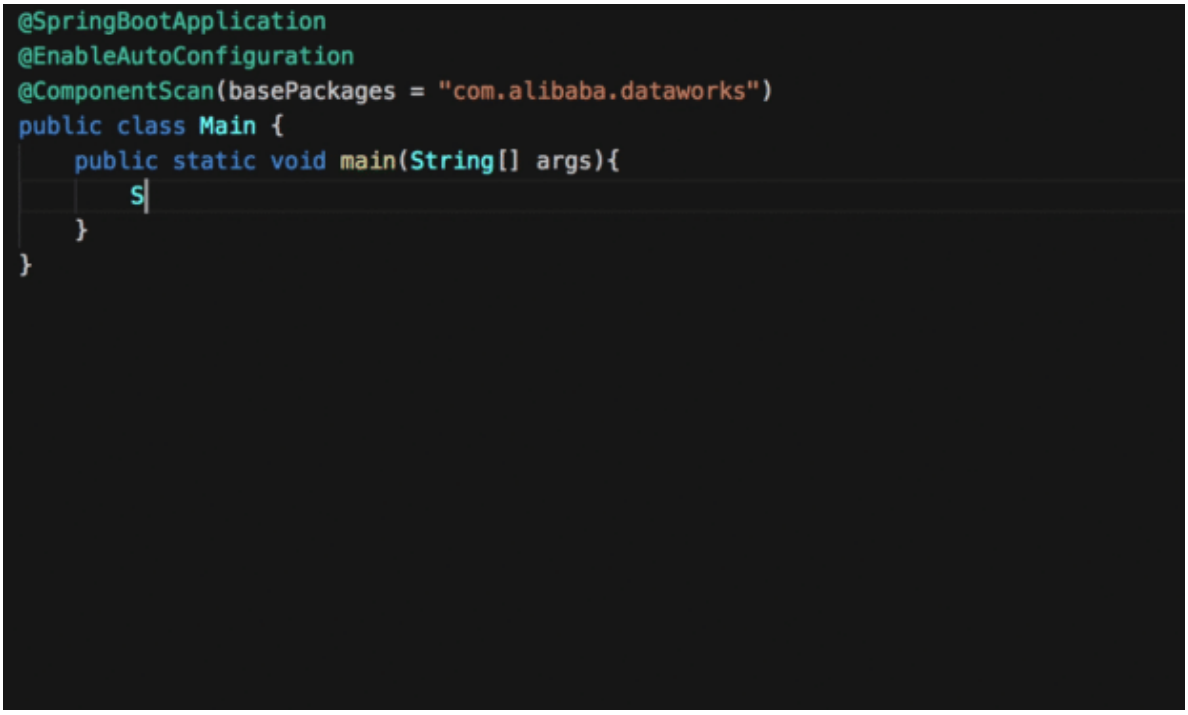
App Studio有以下核心优势：

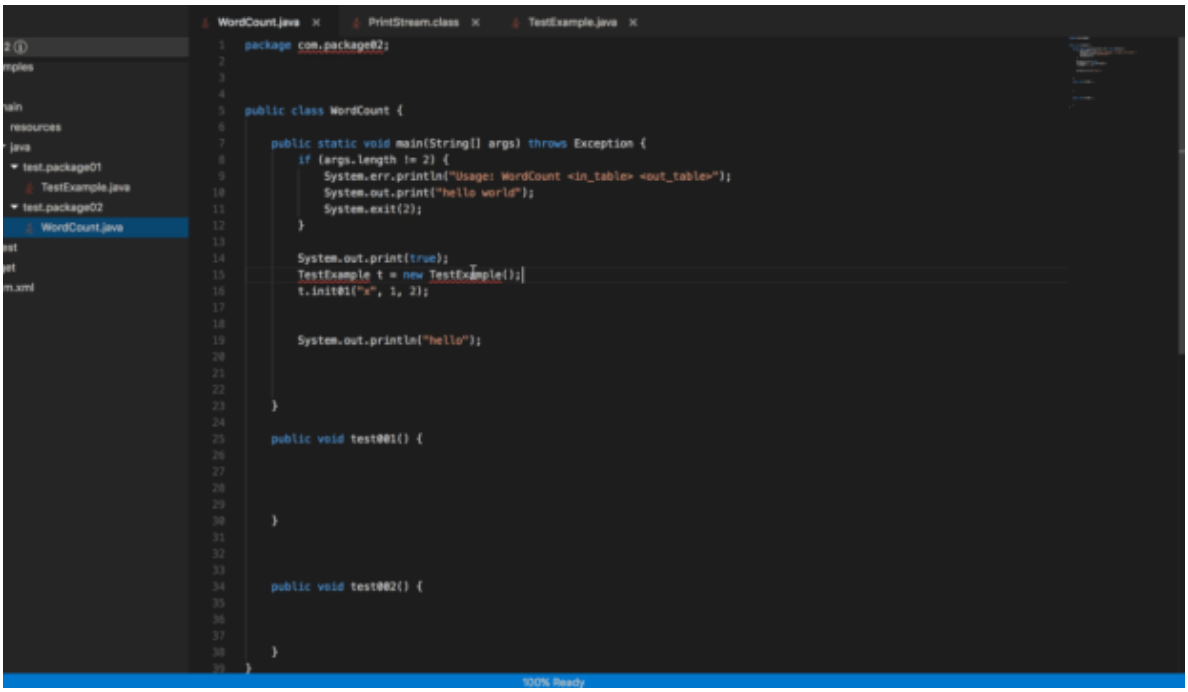
- 随时随地开发

  您无需下载安装本地IDE和配置维护环境变量，只需一个浏览器，即可在办公室、家或任何可以连接网络的地方，进行您的数据开发工作。

- 功能完备的编辑器

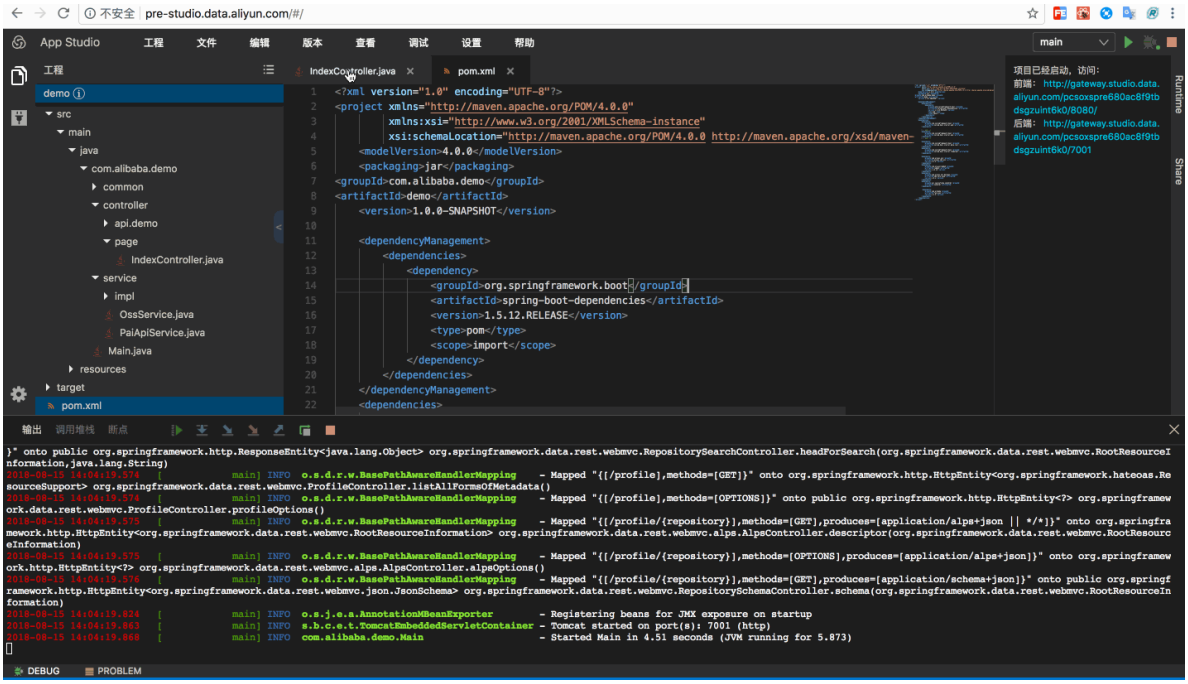
  App Studio提供一个基于浏览器的编辑器，您可以使用它轻松地编写、运行和调试项目。当您输入代码时，App Studio会提供智能提示、补全代码并提供修复建议。您还可以查找方法的引用和定义，自动生成代码。





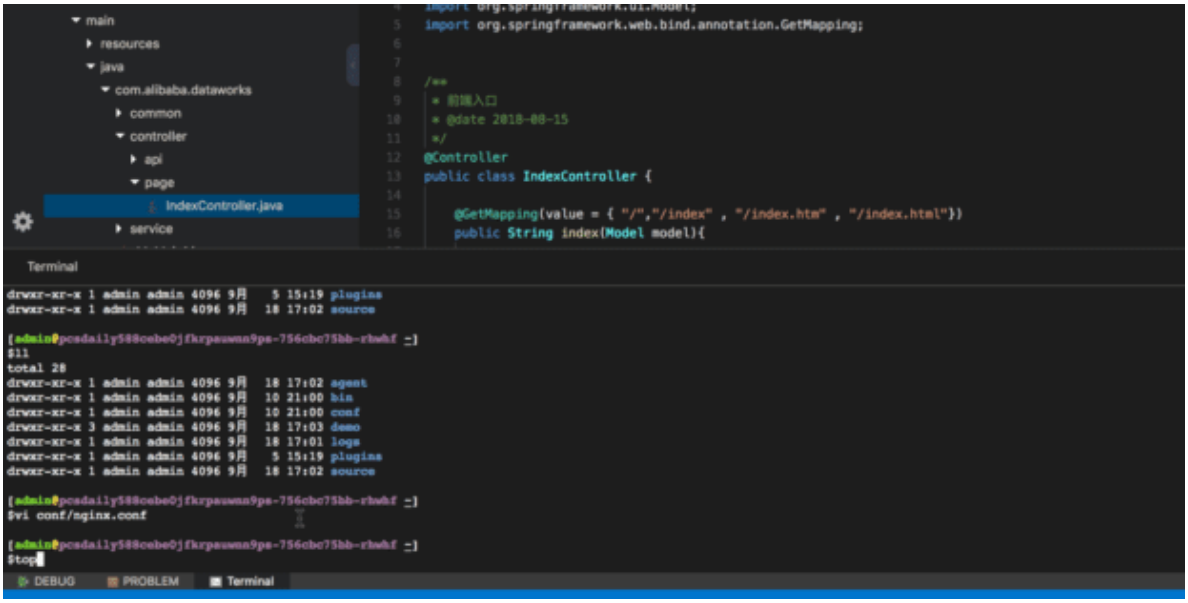
- 在线调试功能

  在线调试具有本地IDE所有的断点类型和断点操作，支持线程切换、过滤，支持变量值查看、监视，支持远程调试和热部署。



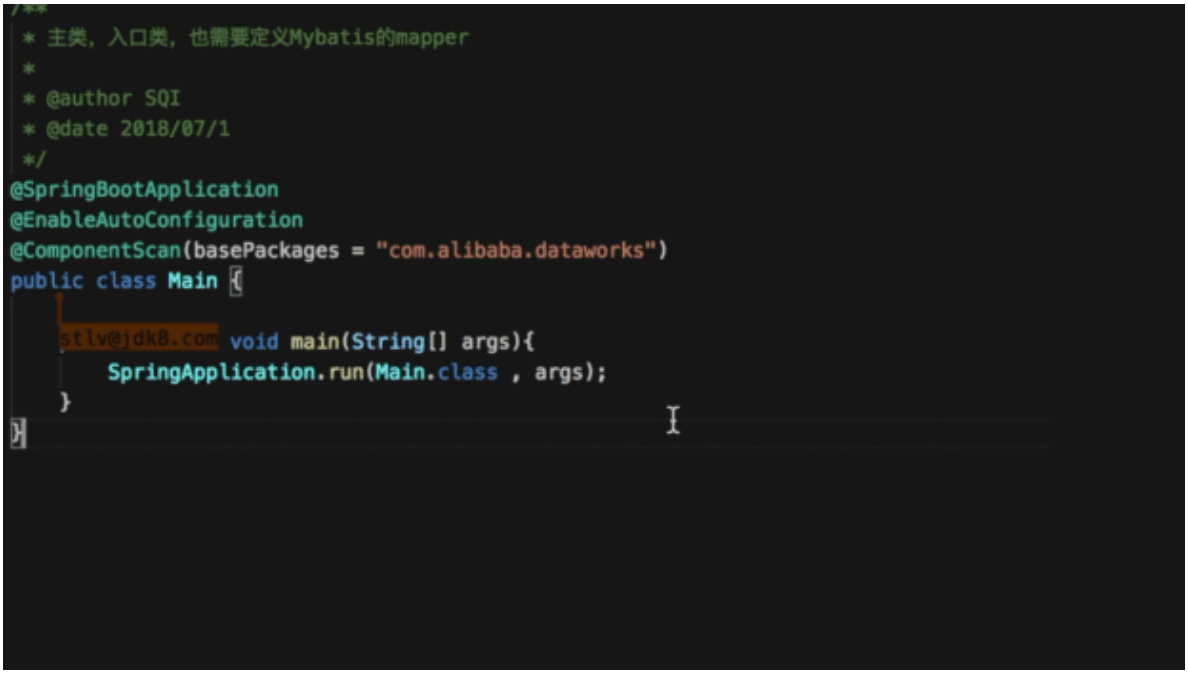
- 多功能终端

  开发者可以直接进入运行环境，目前的运行环境基于CentOS作为基础镜像来构建。终端可以支持任意的bash命令，包括VIM等具有交互功能的命令。



- 协同编辑

  您和您的团队成员可以借助App Studio共享开发环境，进行团队协作编程。目前可支持8人同时在线编辑同一个工程的同一个文件，提高工作效率。后续协同编辑组建还会支持聊天、弹幕、代码批注、视频等功能，让团队合作更加轻松。



- 插件体系

  App Studio支持业务插件、工具插件和语言插件三种插件。

  - 您可以根据业务在App Studio上定制任意的菜单栏，在界面增加业务入口。
  - 您可以定制专属于您业务的项目管理过程、工程类型和模板。
  - 您可以开发通用工具，例如GIT功能增强、代码规则扫描、快捷键、编辑功能增强、代码片段等集成到App Studio中。
  - 您可以通过语言插件扩展App Studio支持的语言，满足自身需求的同时帮助App Studio建设服务更多的语言用户。

- 可视化搭建

  App Studio提供丰富的组件，并且深度打通数据服务和DataStudio。您能且仅能在App Studio中调用DataWorks的Open API，并且通过可视化拖拽、配置的方式快速搭建前端应用，真正实现零代码开发Web应用。

- 丰富灵活的项目管理

  App Studio提供了丰富多样的模板工程，您可以基于模板工程进行再次开发，节省人力提高效率。您也可以将您自己的工程保存为模板，供您自己后续开发使用，或分享给其他人使用。

13.2 App Studio版本历史

本文将为您及时同步App Studio的版本更新。

App Studio 1.0

发布日期：2019年4月3日

发布内容：在Function Studio的基础上实现一个能做应用发布的IDE，核心功能如下所示：

- LSP语言服务

  支持语法高亮、智能提示、智能补全、智能诊断、查找定义、查找引用等本地编辑体验。
- 支持Debug功能

  具有本地IDE所有的断点类型和断点操作，支持线程切换、过滤，支持变量值查看、监视，支持远程调试、热部署和多功能终端。
- 支持基于接口定义的前后端开发

  前端可视化的组件可以通过配置后端接口进行前端联动。
- 前端可视化搭建

  您可以通过拖拽组件搭建前端应用，具有很大的灵活性，支持没有前端经验的用户开发前端应用。支持前端模板管理，支持可视化模式和代码模式互转，可兼顾开发者更高的开发需求。
- 具备代码版本管理能力。
- 可以在线部署和实时预览应用。
- 具备协同编辑功能

  支持8人同时在线编辑同一个工程同一个文件。
- 支持用户自定义工程模板，提供强大的工程管理能力。
- 具有插件开发和集成能力

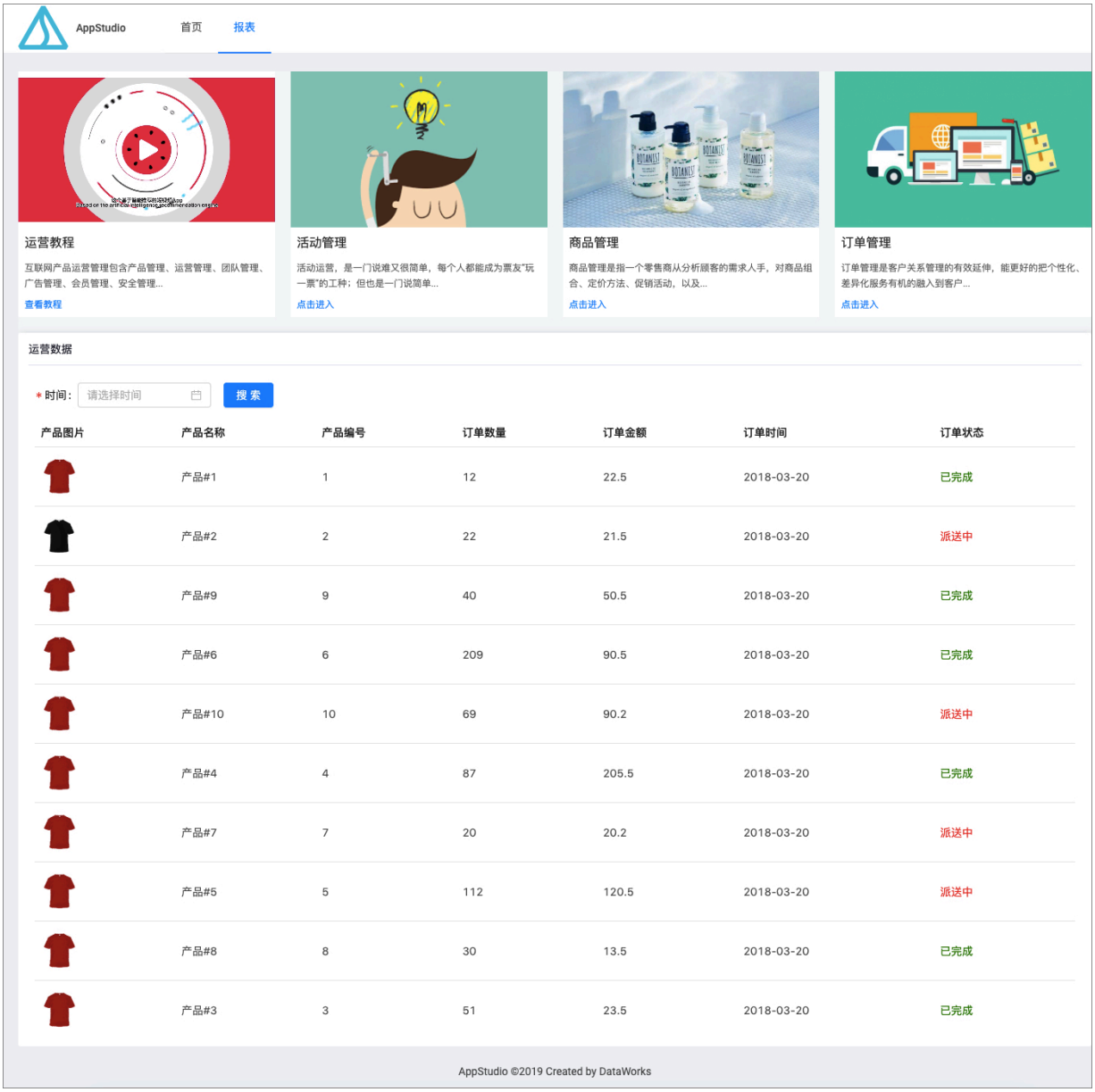
  支持用户开发插件定制专属于业务的IDE（计划在App Studio1.1版本和插件市场一起发布）。
- 支持Java、JS、CSS、HTML和Python多种语言。
- 支持UT自动生成和运行。
- 可设置项目为可分享状态，并通过链接分享给他人（计划在App Studio1.1版本和插件市场一起发布）。
- 支持开发完成的应用在线发布（计划在App Studio1.2版本发布）。

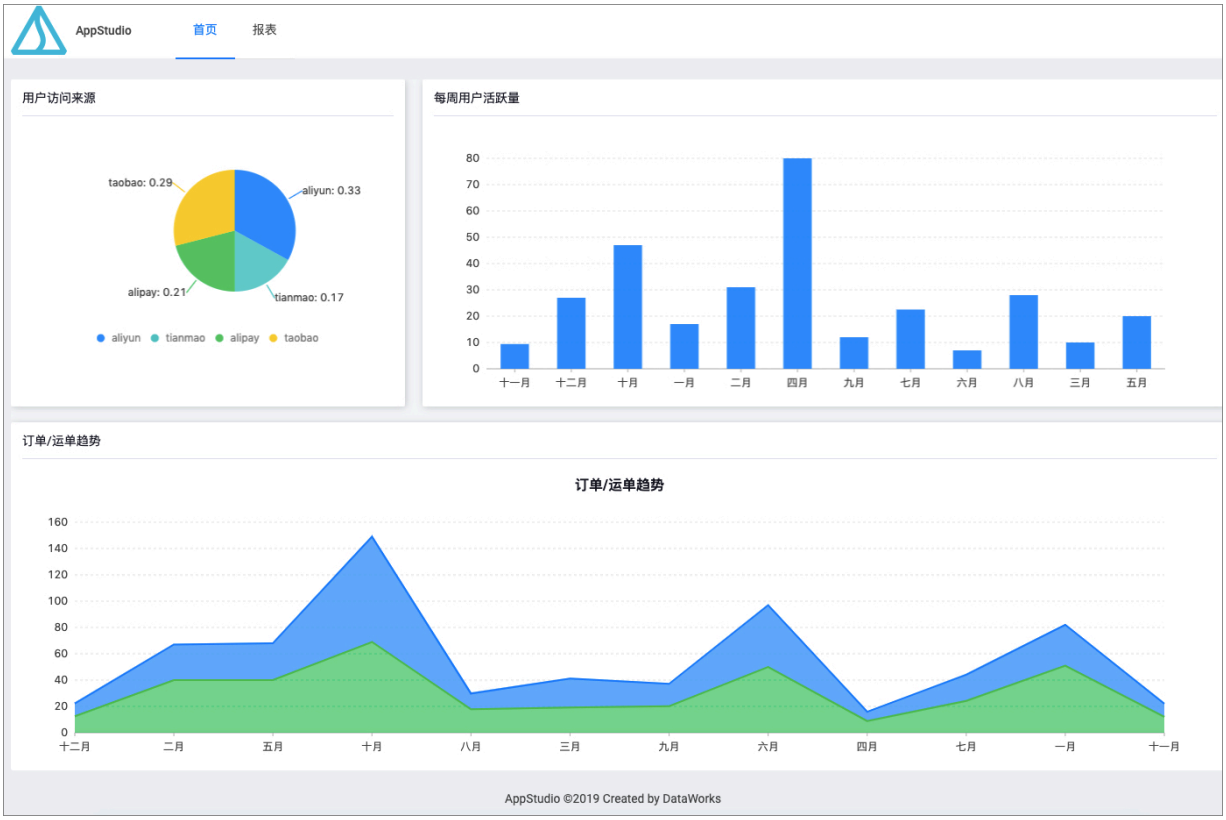
13.3 入门教程

通常，工程师搭建一个数据门户需要开发数据、搭建后端服务和开发前端页面三个环节。本文将为您介绍App Studio的基本功能及如何使用App Studio。

通常，数据工程师在DataWorks进行离线或流式数据开发。随着DataWorks的操作越来越简单，算法工程师、BI分析师、运营、熟悉SQL的产品经理等诸多角色，也逐渐可以在DataWorks进行数据开发。

针对不同种类的用户，App Studio可以助您快速搭建看数据的网页、查数据的App。



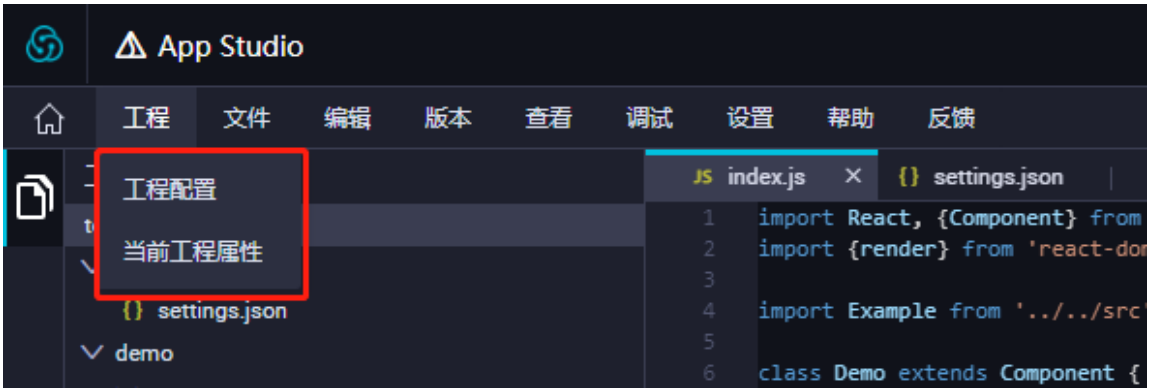

AppStudio
首页
报表
用户访问来源
taobao: 0.29
aliyun: 0.33
alipay: 0.21
tianmao: 0.17
aliyun tianmao alipay taobao
每周用户活跃量
80
70
60
50
40
30
20
10
0
十一月 十二月 十月 一月 二月 四月 九月 七月 六月 八月 三月 五月
订单/运单趋势
订单/运单趋势
160
140
120
100
80
60
40
20
0
十二月 二月 五月 十月 八月 三月 九月 六月 四月 七月 一月 十一月
AppStudio ©2019 Created by DataWorks

了解App Studio

- 菜单栏

  - 工程

    您可以通过工程菜单中的子菜单进行工程配置和查看当前工程属性。当前工程属性中包括工程ID、工程名、工程类型、创建时间和UUID等工程相关信息。



  - 文件

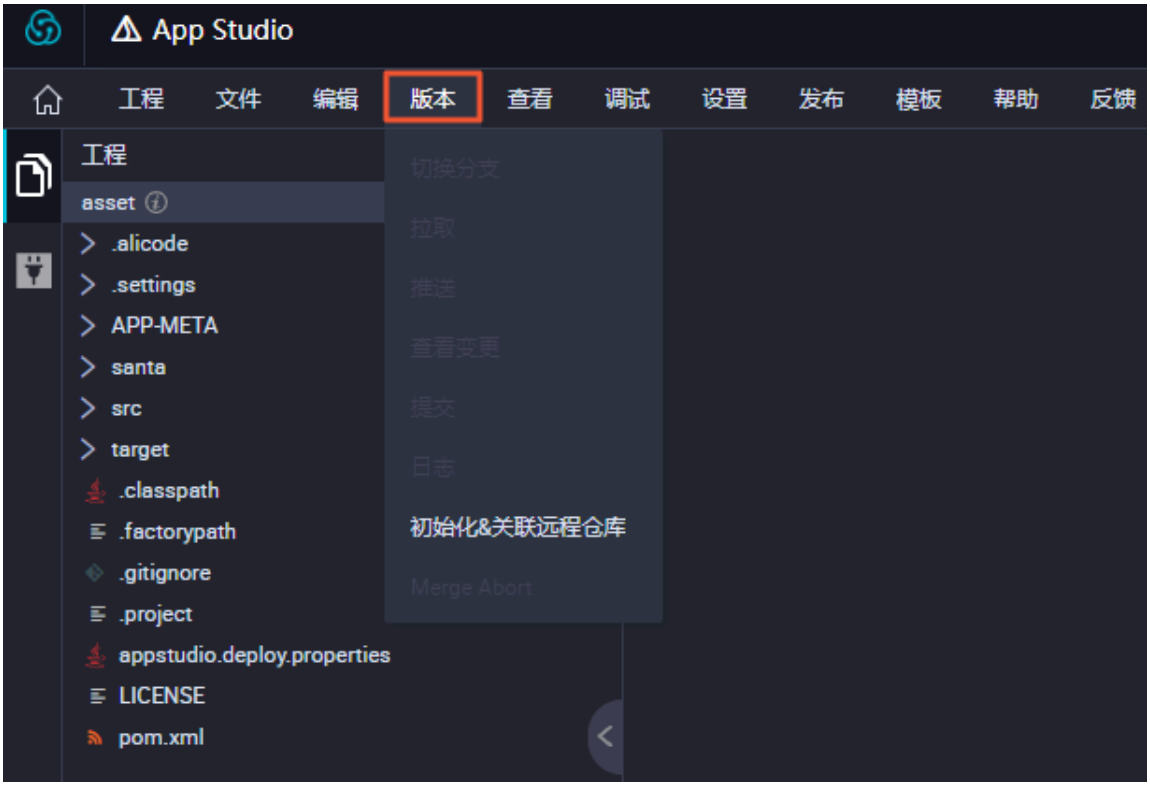
    您可以通过文件菜单中的子菜单新建文件、打开最近的文件。

  - 编辑

    您可以通过编辑菜单栏进行常用的编辑操作，全文搜索是对工程内所有代码内容进行搜索，并可打开相关的文件。全文搜索的详情请参见#unique_634。

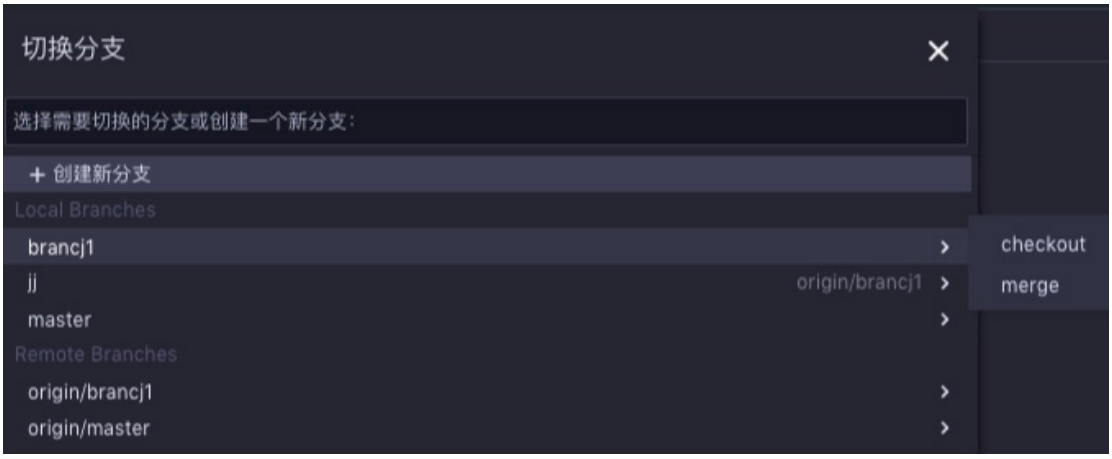
- 版本

  您可以进行切换分支、拉取、推送、查看变更、提交、日志、初始化&关联远程仓库和Merge Abort等操作。

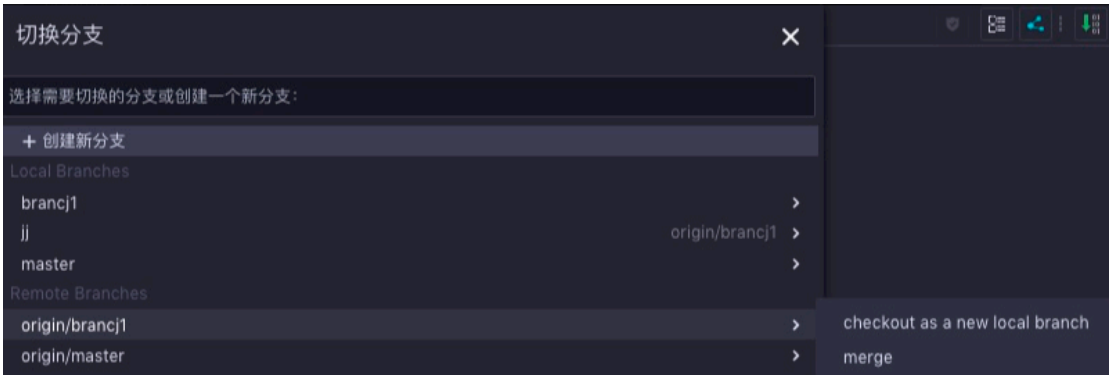


■ 切换分支

您可以通过+创建新分支创建本地新分支，然后推送到远程仓库。您可以选择一个本地分支，单击右边弹出框中的checkout。您也可以通过merge，将选中的分支合并到当前分支。



您可以选择一个远程分支，单击右边弹出框中的check out as a new local branch，将该远程分支checkout到本地并重新命名。您也可以通过merge，将选中的分支合并到当前分支。



■ **拉取**

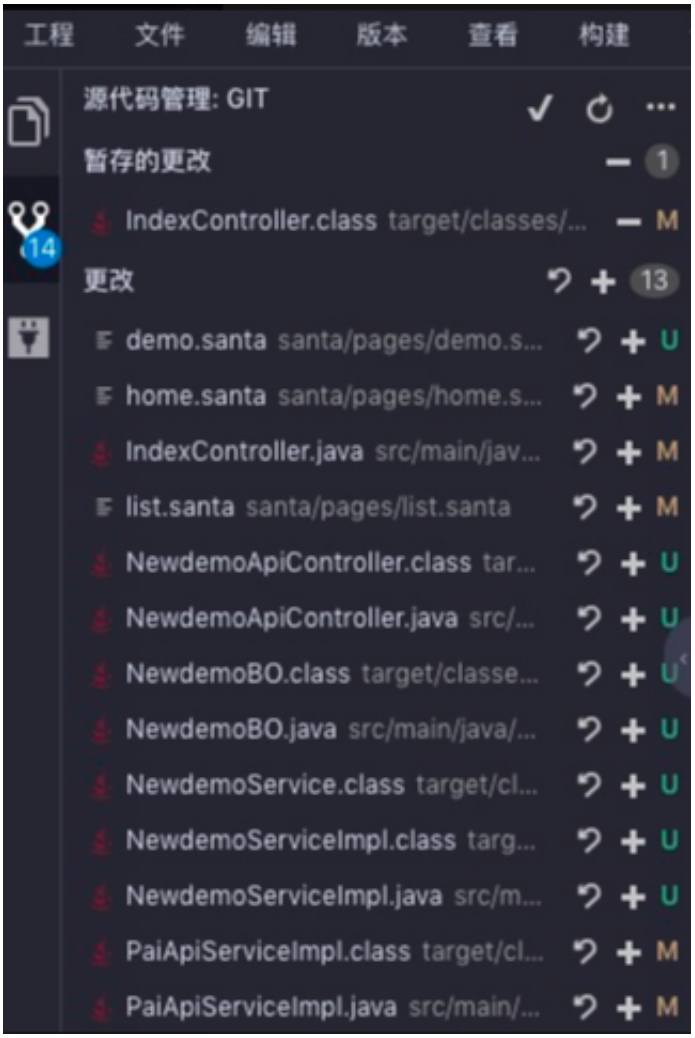
可以将远程分支的代码拉取到本地分支。

■ **推送**

可以将本地分支的变更暂存后推送到远程分支。

■ **查看变更**

单击查看变更后，右侧导航栏会弹出本地变更文件列表。



标识	说明
	代表变更文件的个数。
	代表新增的文件。
	代表变更的文件。
	单击后可以撤销更改。
	单击后可以暂存更改。
	单击后可以撤销暂存。
	单击后可以提交或提交并推送暂存的代码。

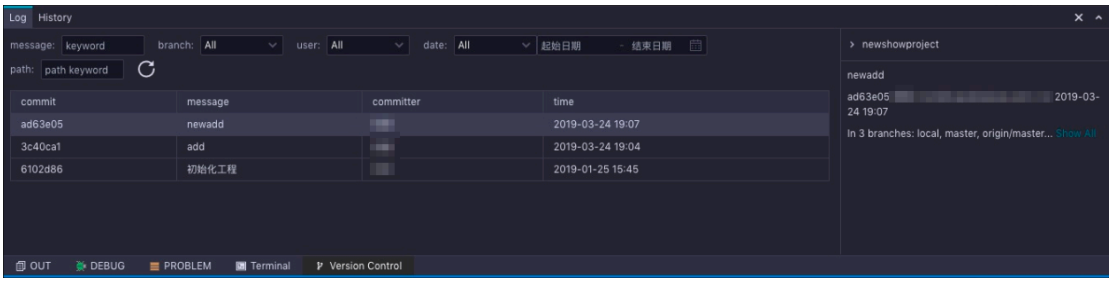
标识	说明
•••	更多中包括推送和拉取操作。

■ 提交

可以将本地分支的变更提交以暂存，需要输入commit信息。

■ 日志

可以查看分支的所有提交记录，并可以筛选。



■ 初始化&关联远程仓库

新建的工程可以关联远程仓库，从而进行版本管理。

- 查看

您可以通过切换全屏将IDE设置成全屏，然后通过Esc键退出全屏。您可以通过切换侧边栏和切换状态栏收起侧边和状态栏。

- 调试

■ 如果您建的是前端工程，调试选项如下所示。

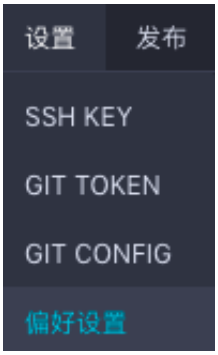
您可以配置运行参数、添加自定义镜像。

■ 如果您建的是后端工程，调试选项如下所示。

App Studio支持Java Debug，在后端工程的调试中，除配置和自定义镜像操作外，还有很多调试相关的操作。同时会有全量构建、增量构建、编译的操作入口。

- 设置

您在开始使用App Studio前，需要配置SSH KEY和GIT CONFIG。您也可以通过偏好设置，设置自己偏好的属性，目前仅支持字体大小，后续会支持颜色、样式、主题、快捷键等。



- **帮助**

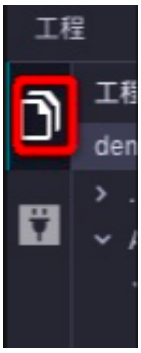
  您可以在帮助中查看产品使用文档、查看快捷键、查看版本历史和清空本地缓存。

- **反馈**

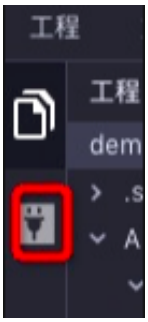
  您可以通过反馈提交问题和需求。

- 左边栏
  - 入口

    单击下图中的图标，即可展开工程区。

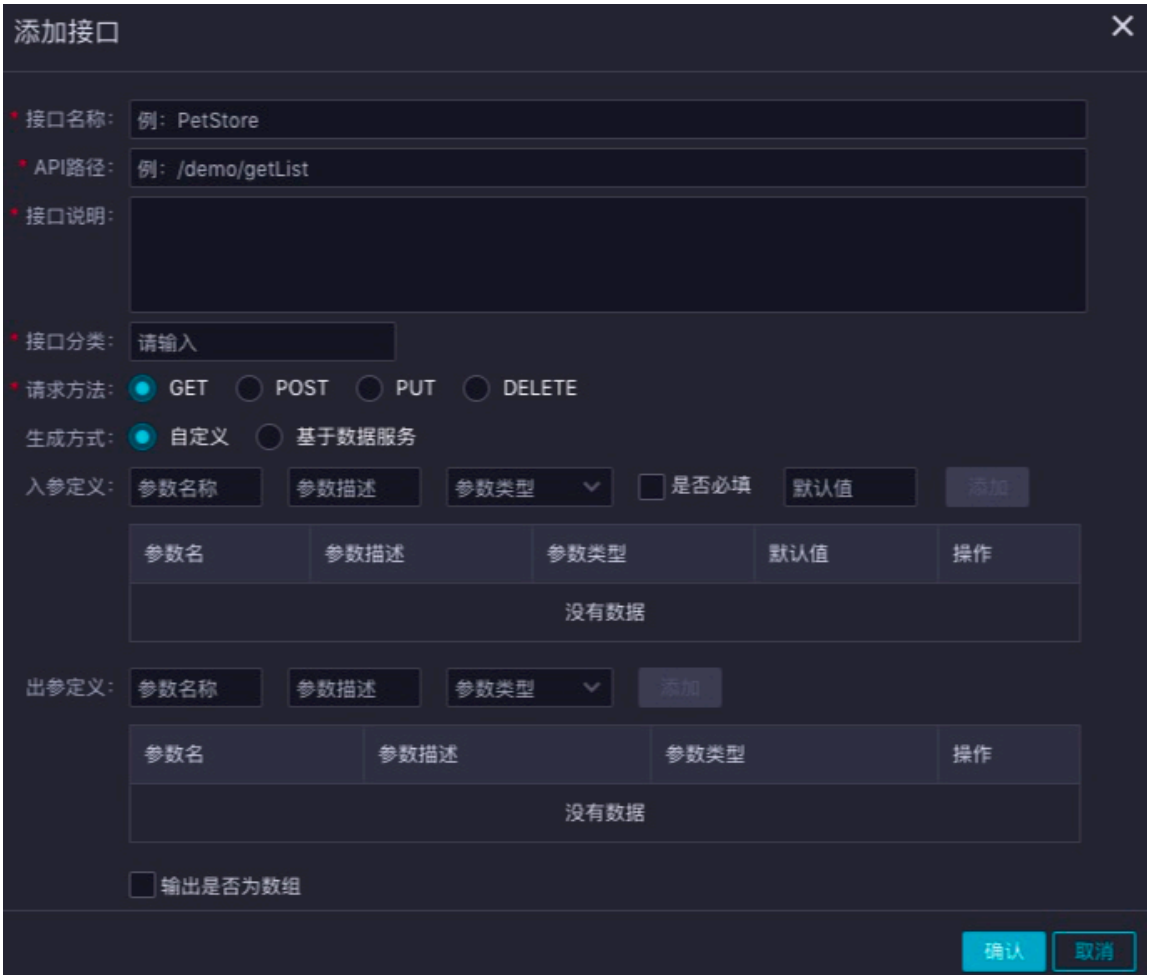
    单击下图中的图标，即可展开接口定义区。

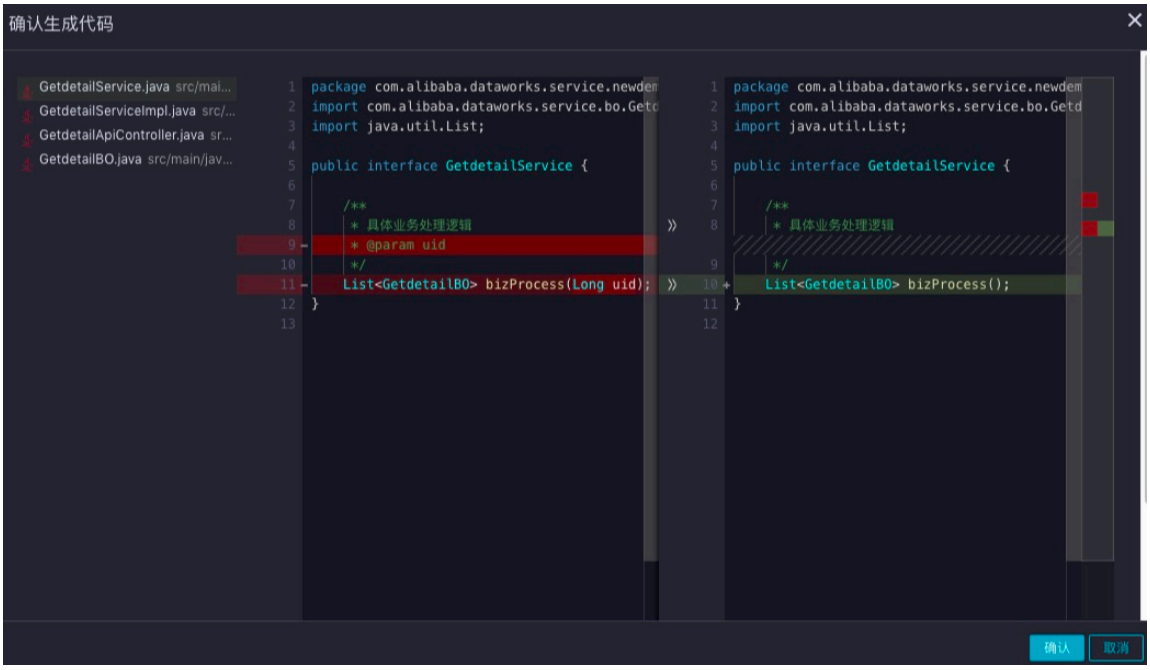
    

  - 接口定义区

    您可以添加接口并自动生成接口类代码，还可通过箭头，将左边的新代码同步到右边的本地代码中。

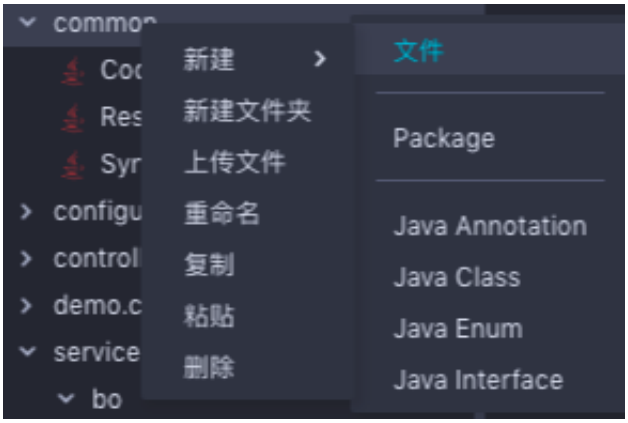




- 工程区

■ 文件夹操作

如果您创建的是后端工程，文件模板新建后，会帮您自动生成一些框架代码。

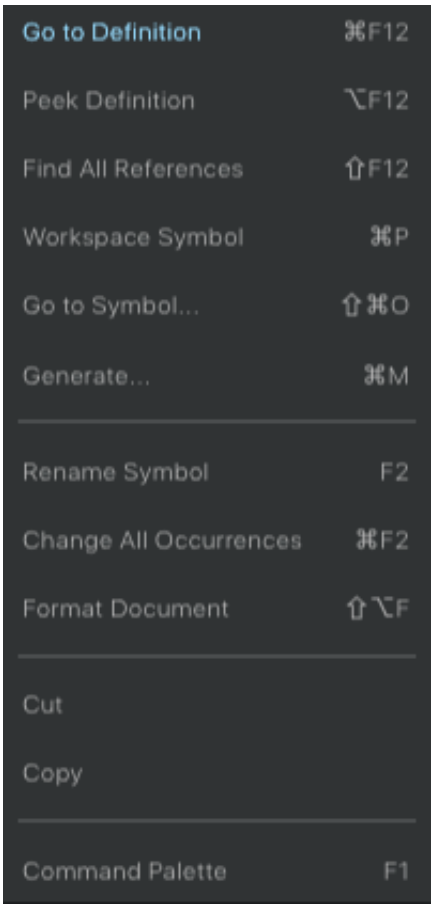


■ 文件操作

如果您创建的是前端工程，则新建操作只有文件一个选项。

您可以重命名、复制和删除文件，也可以查看文件的GIT提交历史并进行版本对比。

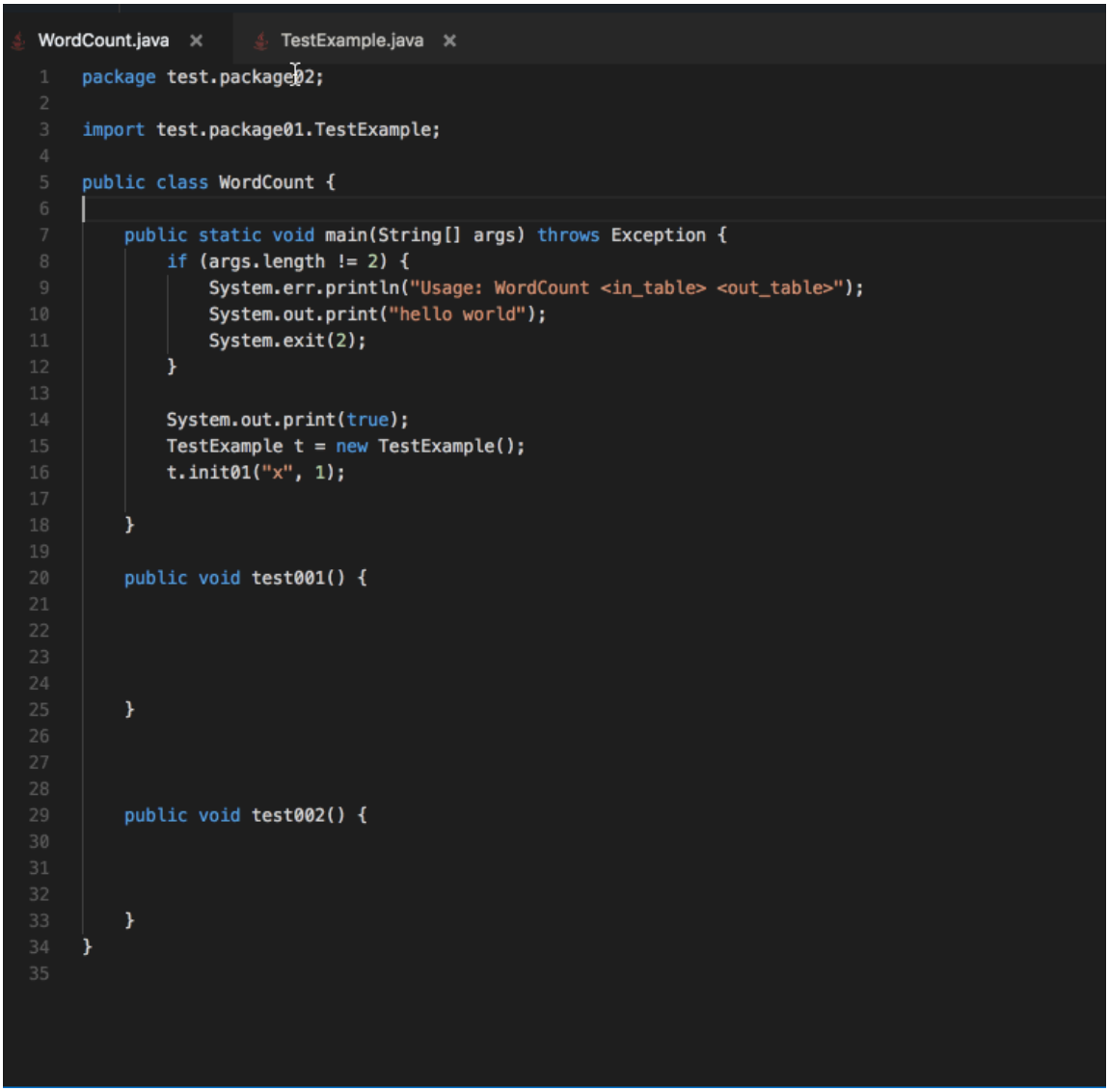
- 编辑区
  - 右键操作



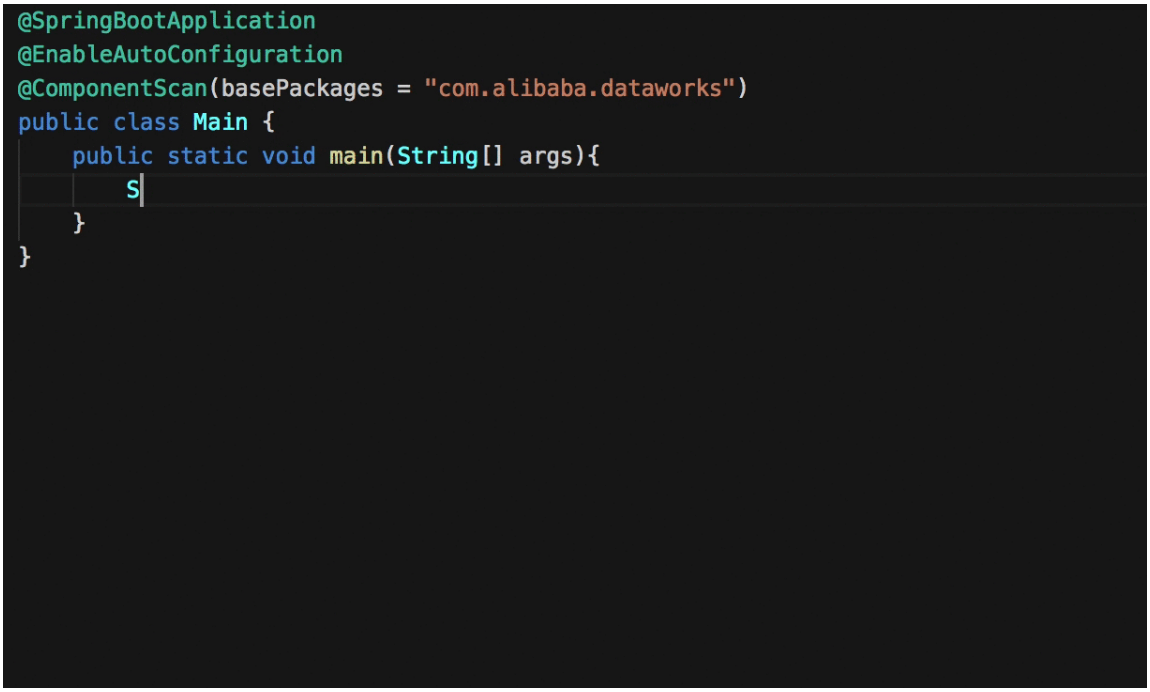
操作	说明
Go to Definition	单击后跳转至定义。
Peek Definition	单击后可以预览定义。
Find All References	单击后可以查找所有引用。
Workspace Symbol	单击后可以在项目中查找符号。
Go to Symbol...	单击后可以跳转至符号。
Generate...	单击后可以生成代码。
Rename Symbol	单击后可以重命名符号。
Change All Occurrences	单击后可以修改当前文件中的所有该符号名字。
Format Document	单击后可以格式化文件。
Cut	剪切。
Copy	复制。

操作	说明
Command Palette	单击后可以进入命令面板。

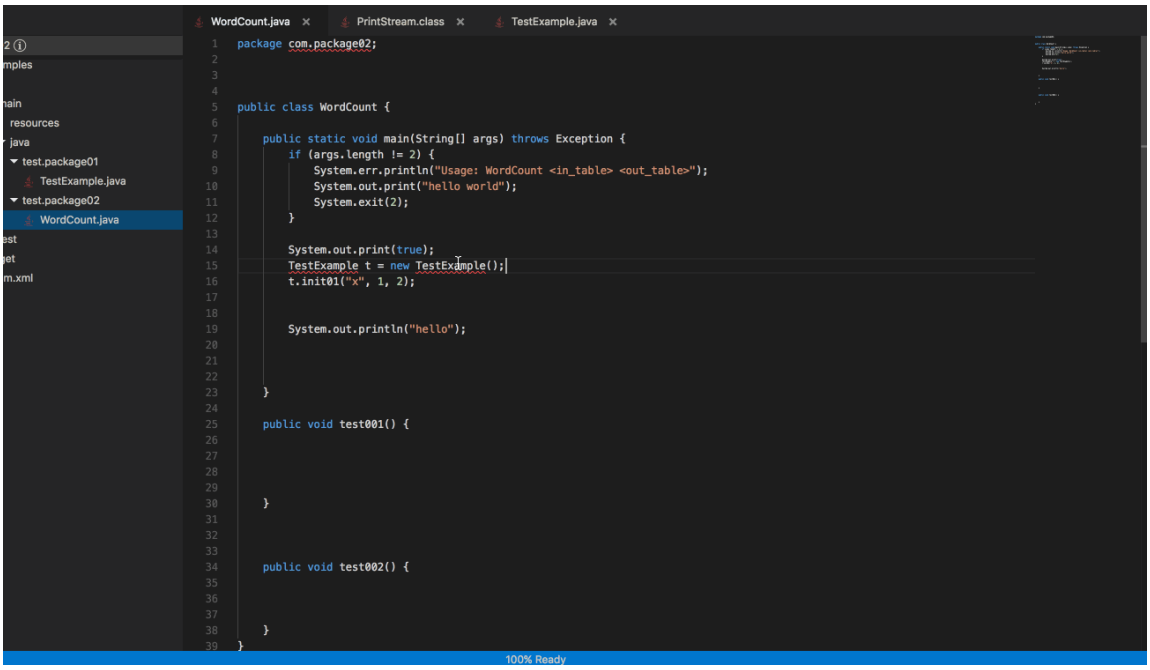
- 智能提示



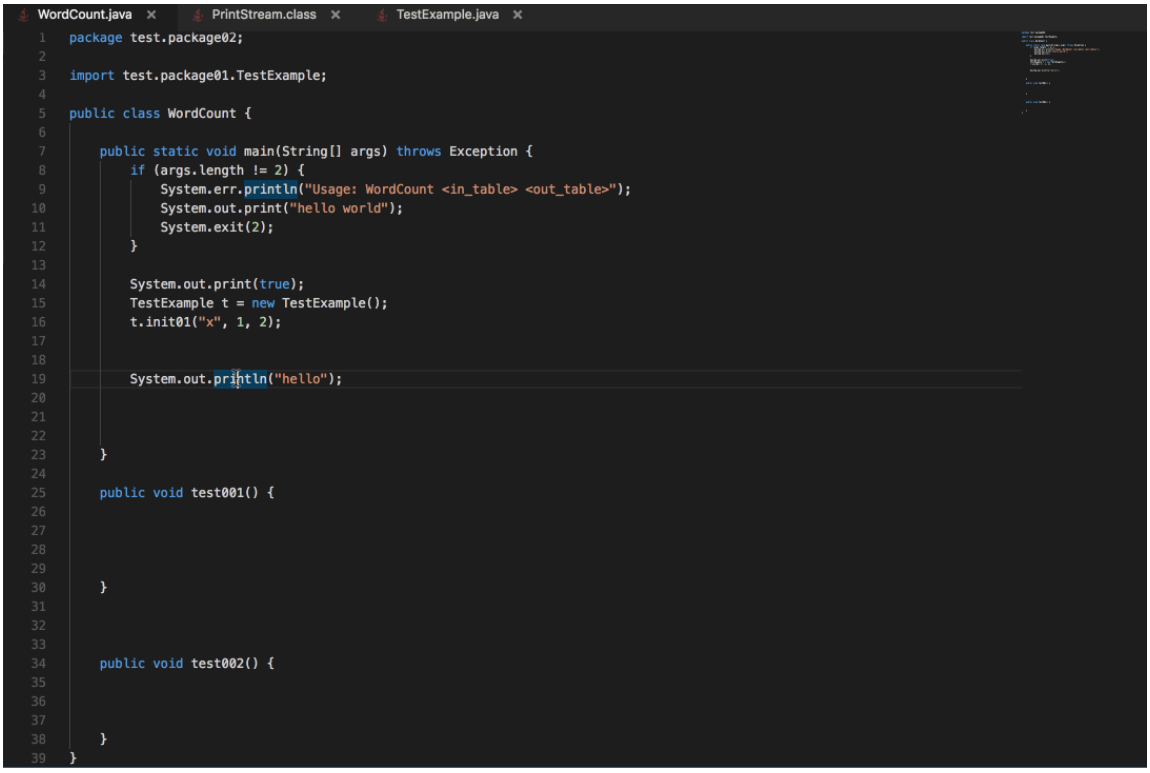
- 智能补全


@SpringBootApplication
@EnableAutoConfiguration
@ComponentScan(basePackages = "com.alibaba.dataworks")
public class Main {
public static void main(String[] args){
S
}
}

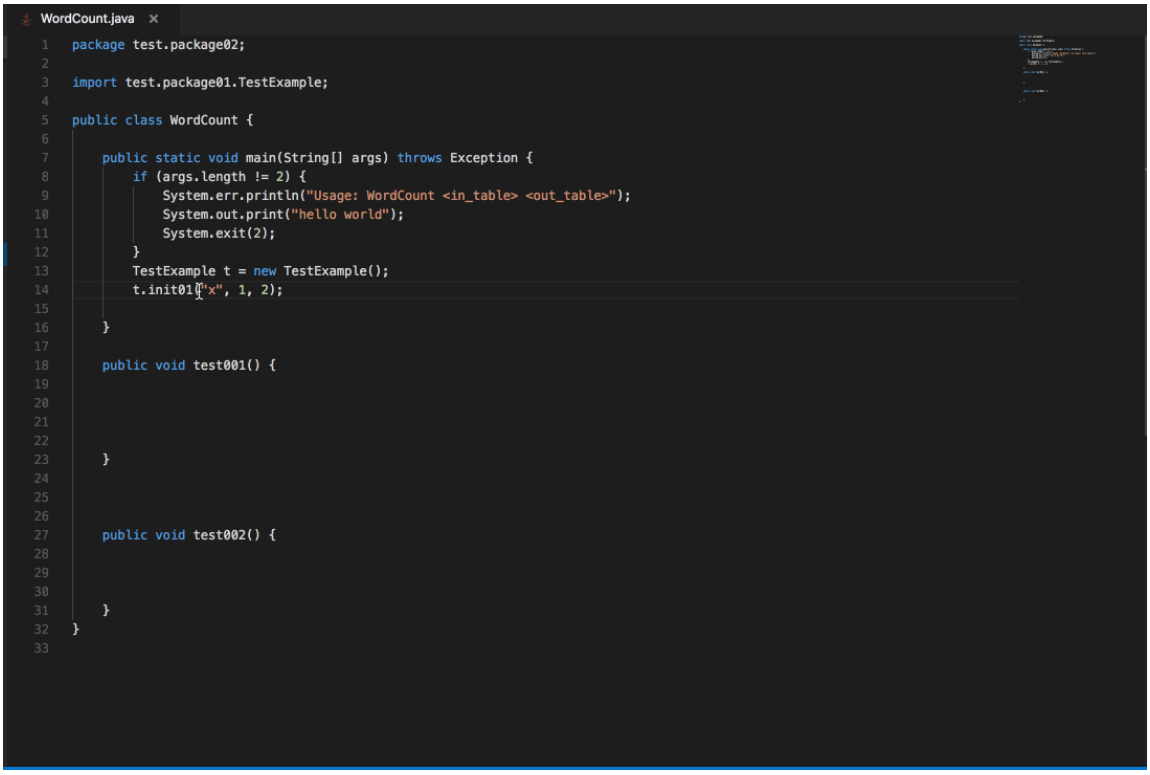
- 智能诊断


WordCount.java
PrintStream.class
TestExample.java
package com.package02;
public class WordCount {
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System.err.println("Usage: WordCount <in_table> <out_table>");
System.out.print("hello world");
System.exit(2);
}
System.out.print(true);
TestExample t = new TestExample();
t.init01("x", 1, 2);
System.out.println("hello");
}
public void test001() {
}
public void test002() {
}
}
100% Ready

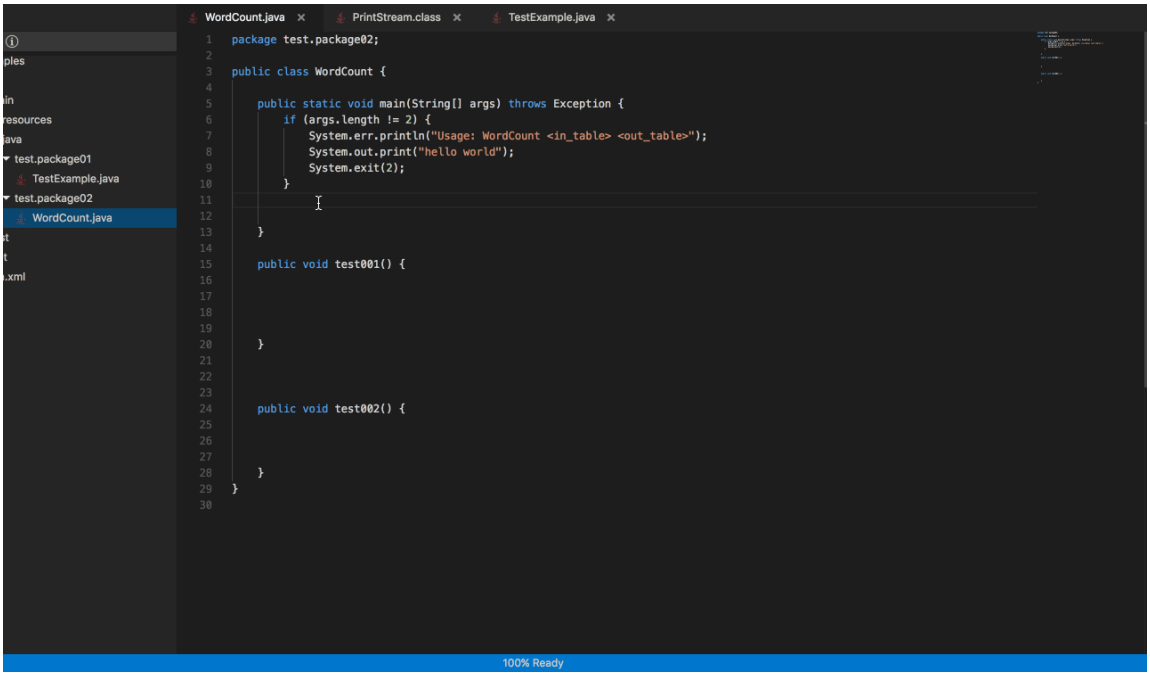
- 查找定义


WordCount.java
PrintStream.class
TestExample.java
package test.package02;
import test.package01.TestExample;
public class WordCount {
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System.err.println("Usage: WordCount <in_table> <out_table>");
System.out.print("hello world");
System.exit(2);
}
System.out.print(true);
TestExample t = new TestExample();
t.init01("x", 1, 2);
System.out.println("hello");
}
public void test001() {
}
public void test002() {
}
}

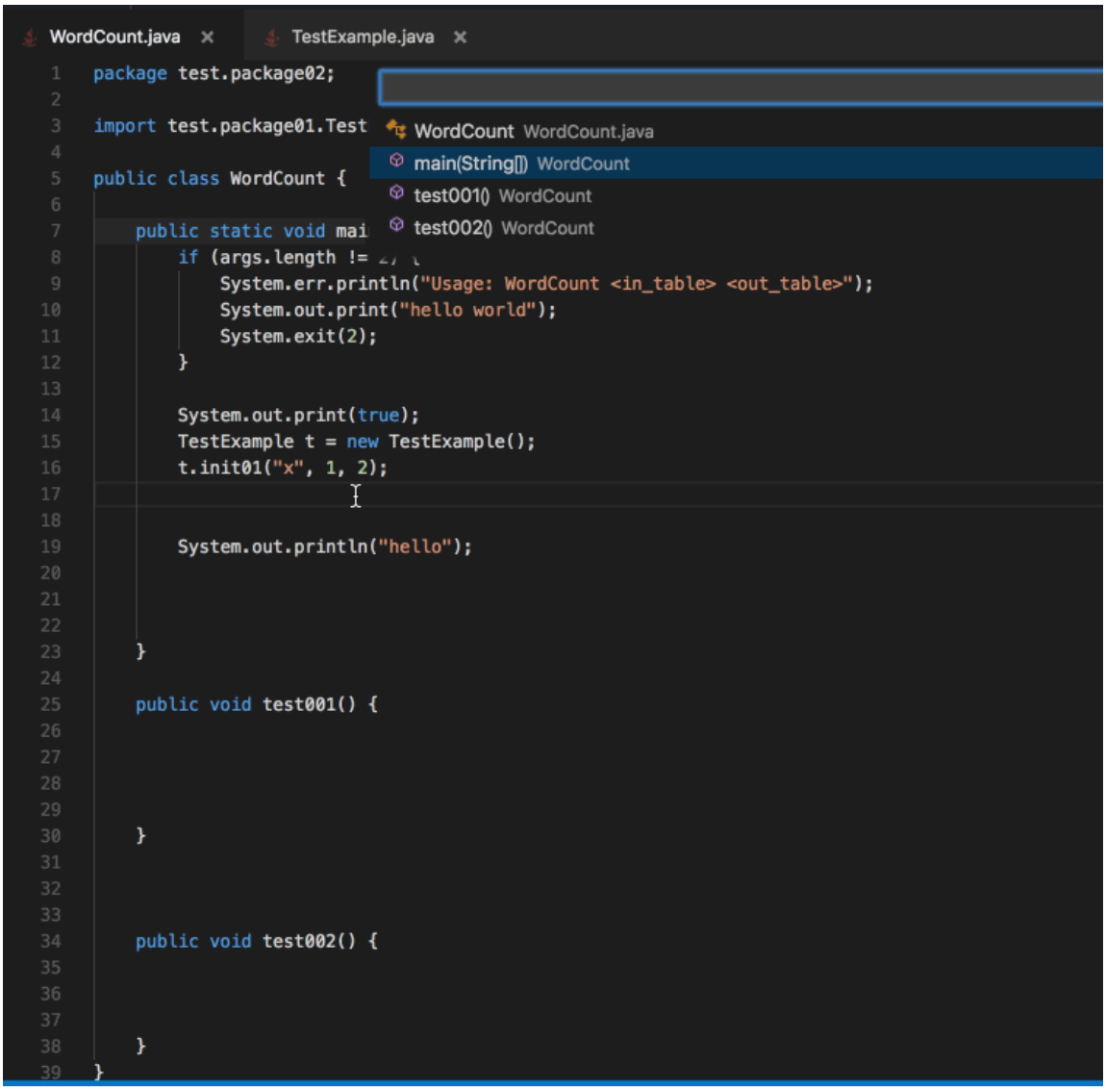
- 查找引用


WordCount.java
package test.package02;
import test.package01.TestExample;
public class WordCount {
public static void main(String[] args) throws Exception {
if (args.length != 2) {
System.err.println("Usage: WordCount <in_table> <out_table>");
System.out.print("hello world");
System.exit(2);
}
TestExample t = new TestExample();
t.init01("x", 1, 2);
}
public void test001() {
}
public void test002() {
}
}

- 自动导入


WordCount.java
PrintStream.class
TestExample.java
test.package01
TestExample.java
test.package02
WordCount.java
100% Ready

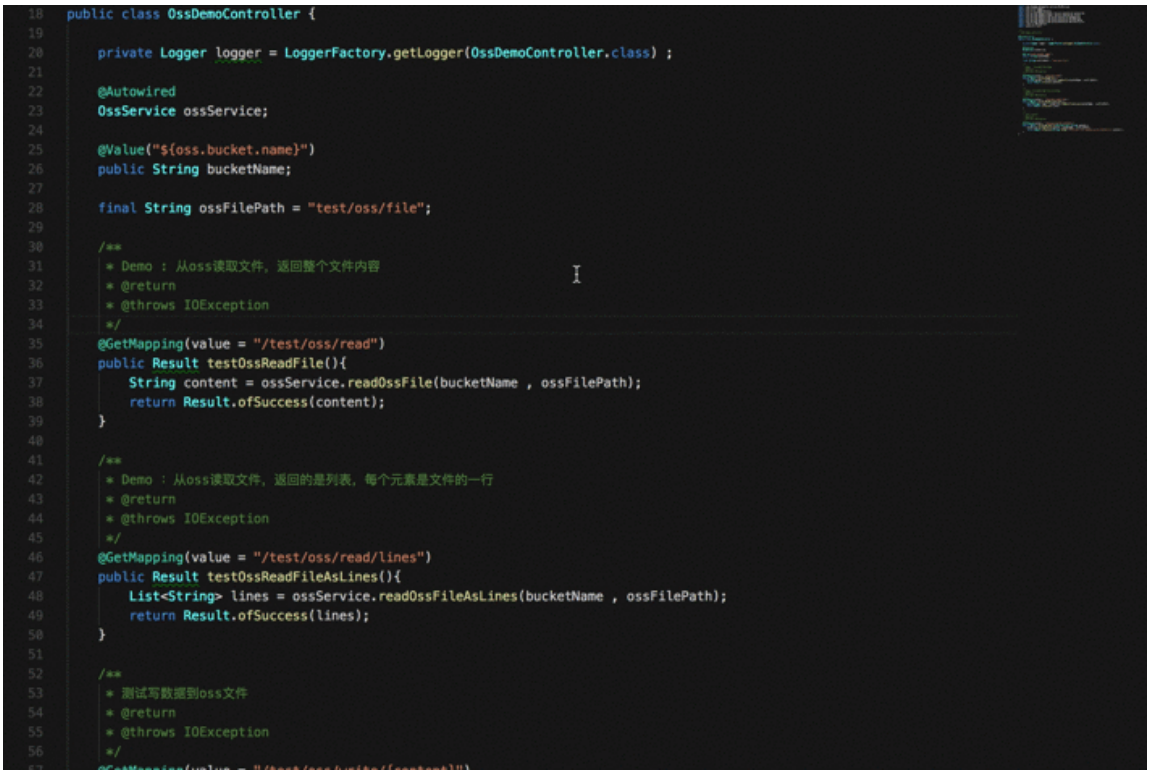
- 查找符号


WordCount.java
TestExample.java
WordCount WordCount.java
main(String[]) WordCount
test001() WordCount
test002() WordCount

- 多光标编辑

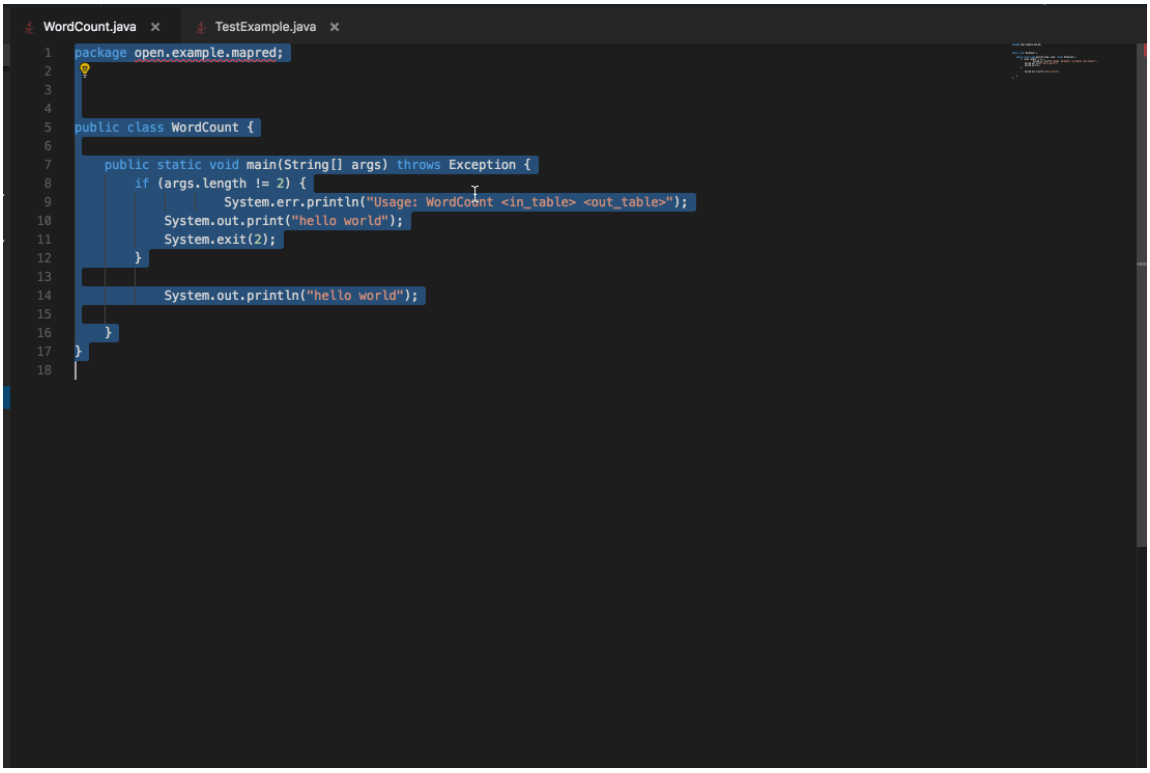
```java
@Controller
public class IndexController {

    @GetMapping(value = { "/","/index" , "/index.htm" , "/index.html"})
    public String index(Model model){
        return "index";
    }

}
```

- 查找、替换

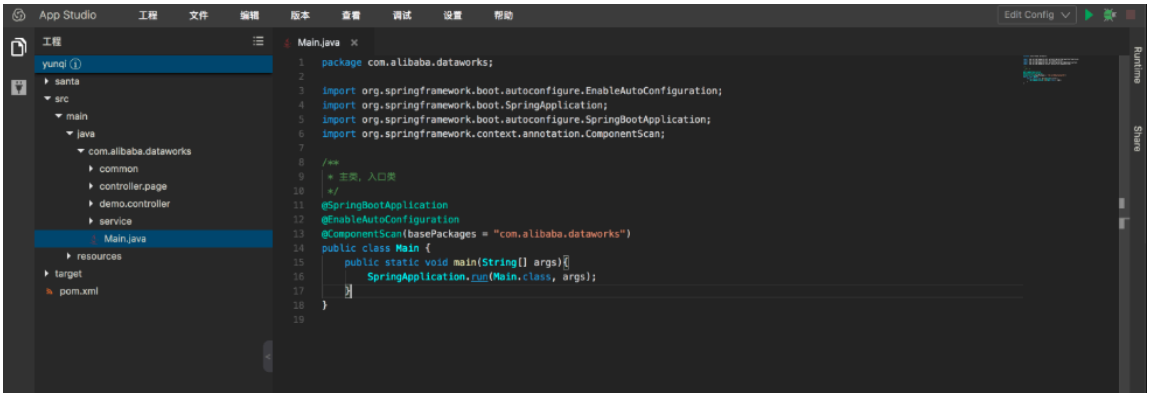


```java
public class OssDemoController {

    private Logger logger = LoggerFactory.getLogger(OssDemoController.class) ;

    @Autowired
    OssService ossService;

    @Value("${oss.bucket.name}")
    public String bucketName;

    final String ossFilePath = "test/oss/file";

    /**
     * Demo : 从oss读取文件，返回整个文件内容
     * @return
     * @throws IOException
     */
    @GetMapping(value = "/test/oss/read")
    public Result testOssReadFile(){
        String content = ossService.readOssFile(bucketName , ossFilePath);
        return Result.ofSuccess(content);
    }

    /**
     * Demo : 从oss读取文件，返回的是列表，每个元素是文件的一行
     * @return
     * @throws IOException
     */
    @GetMapping(value = "/test/oss/read/lines")
    public Result testOssReadFileAsLines(){
        List<String> lines = ossService.readOssFileAsLines(bucketName , ossFilePath);
        return Result.ofSuccess(lines);
    }

    /**
     * 测试写数据到oss文件
     * @return
     * @throws IOException
     */
```

- 代码格式化

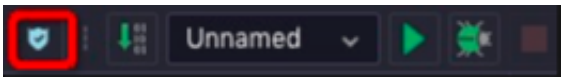


- 括号匹配

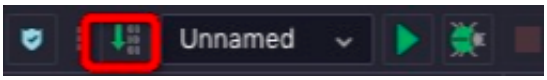


· 右上角图标区

- 编码规约



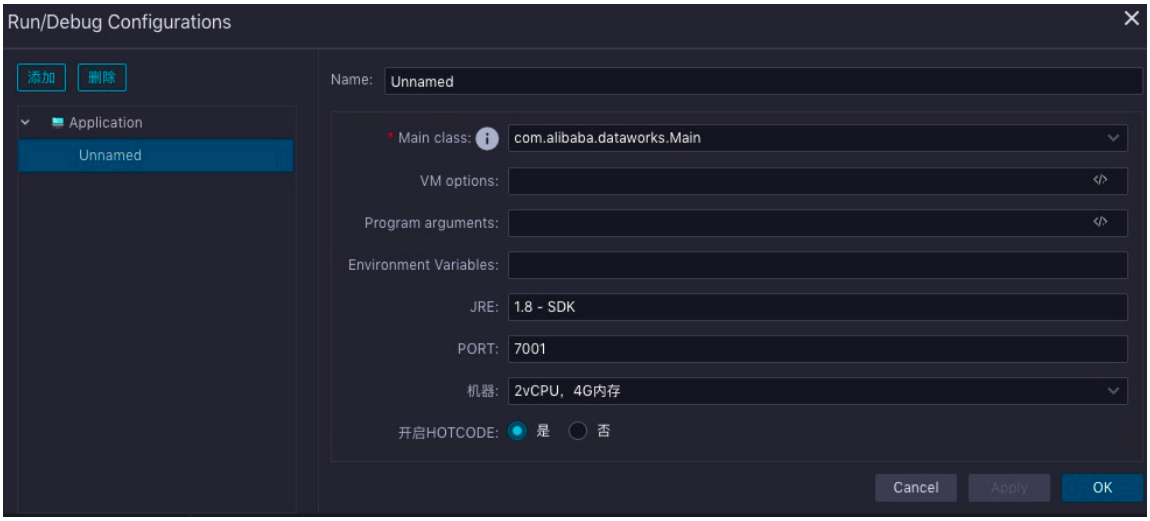
- 构建



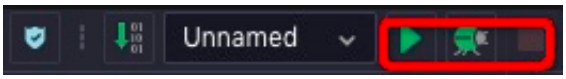
说明:

构建需要在工程运行或者debug时才能进行。

- **Run/Debug Configurations**



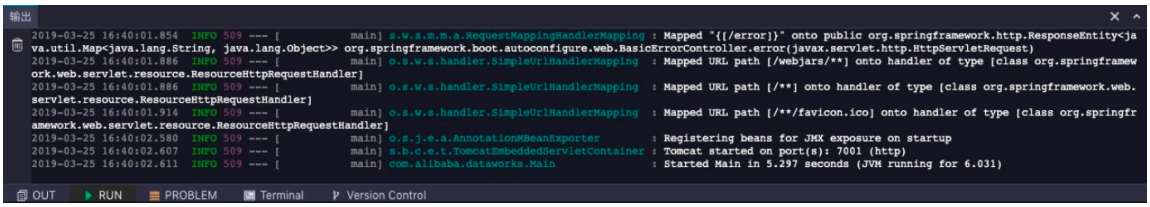
- **Debug入口**



从左到右的图标依次代表运行、Debug和停止工程。

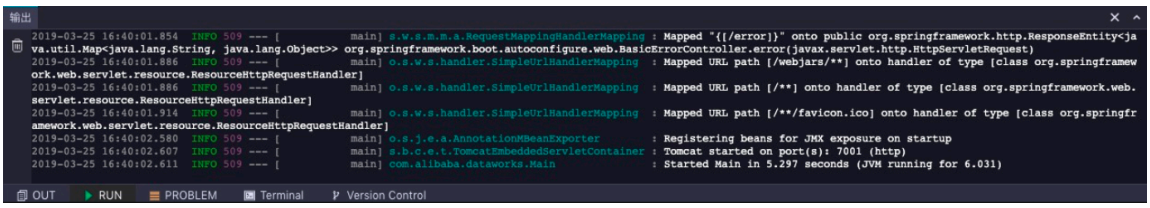
- 底边栏
  - DEBUG/RUN面板

    单击运行或Debug工程，该面板会弹出，展示进度和信息。



  - PROBLEM面板

    当工程有问题时，运行或Debug工程该面板会弹出。



  - TERMINAL面板

    当工程运行或Debug时，可以通过Terminal触达机器进行bash、vim命令操作。

  - VERSION CONTROL面板

    该面板展示Git history和Git log两部分内容。

- 右边栏
  - Runtime

    工程运行完成时会展开这个面板，并展示机器信息和访问链接。

    - 如果是后端工程，仅展示后端访问链接。
    - 如果是前端工程，仅展示前端链接。
    - 如果是可视化搭建工程，可展示前端访问链接和后端访问链接。
  - Share

    您可以邀请他人协同编程，目前支持8人同时编辑同一工程同一文件。

- **Data**

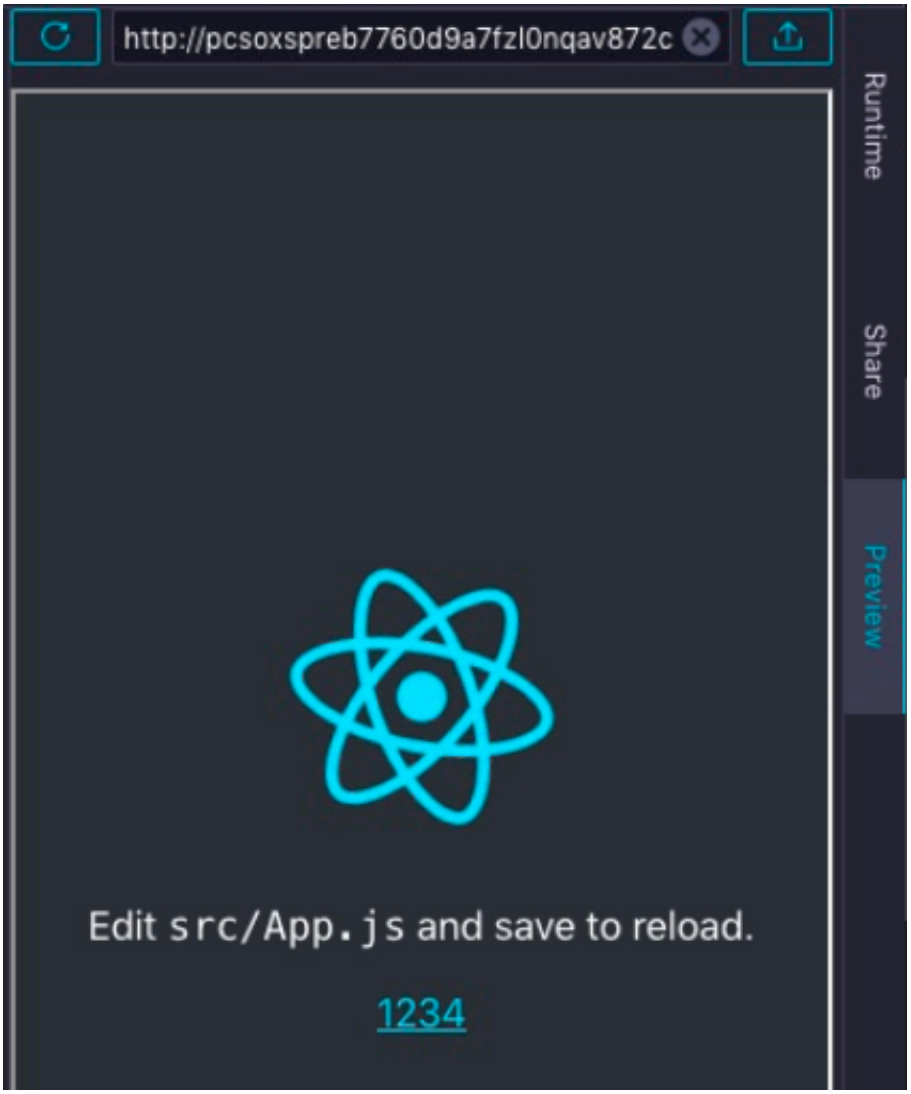
  数据服务是承接DataStudio和App Studio的重要一环。

数据服务在App Studio中有两种使用方式，更多详情请参见数据服务。

- 可以在代码中直接使用，或者对接口结果进行再加工。
- 可以在可视化搭建中直接配置为组件的数据源。

- Preview

  如果您创建的是前端工程，在右边栏会有Preview入口，在运行工程时可以实时预览前端页面。

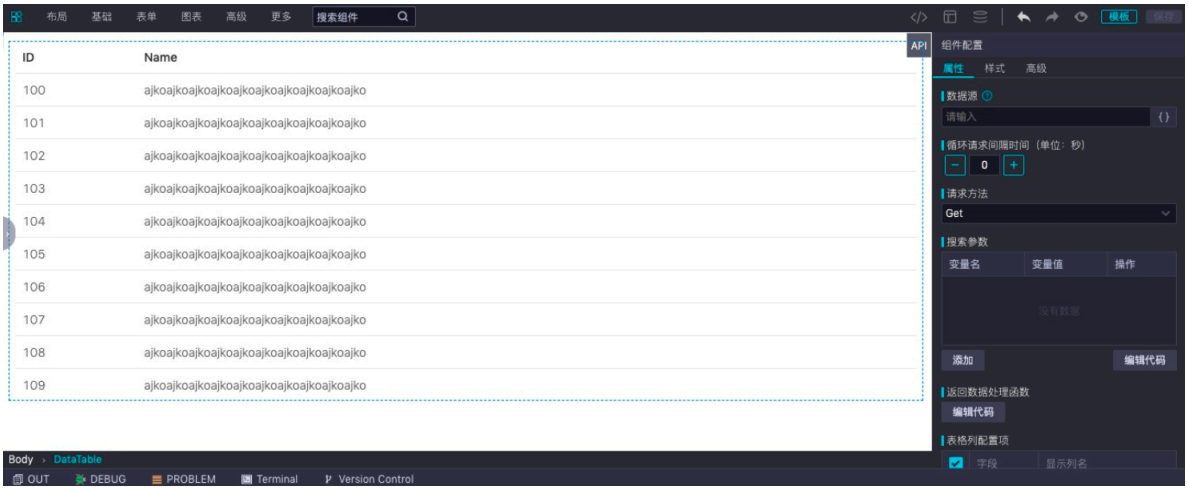

http://pcsoxspreb7760d9a7fzl0nqav872c
Runtime
Share
Preview
Edit src/App.js and save to reload.
1234

- 可视化搭建区

  您的工程必须是可视化搭建工程，在工程中找到santa-pages下的.santa文件，双击打开。

  左上角是组件区，您可以选择需要的组件，也可以通过搜索组件名称选择需要的组件。右上角的图标分别是切换为代码模式、导航配置、全局数据流配置、撤销、重做、预览、保存为模板和保存。

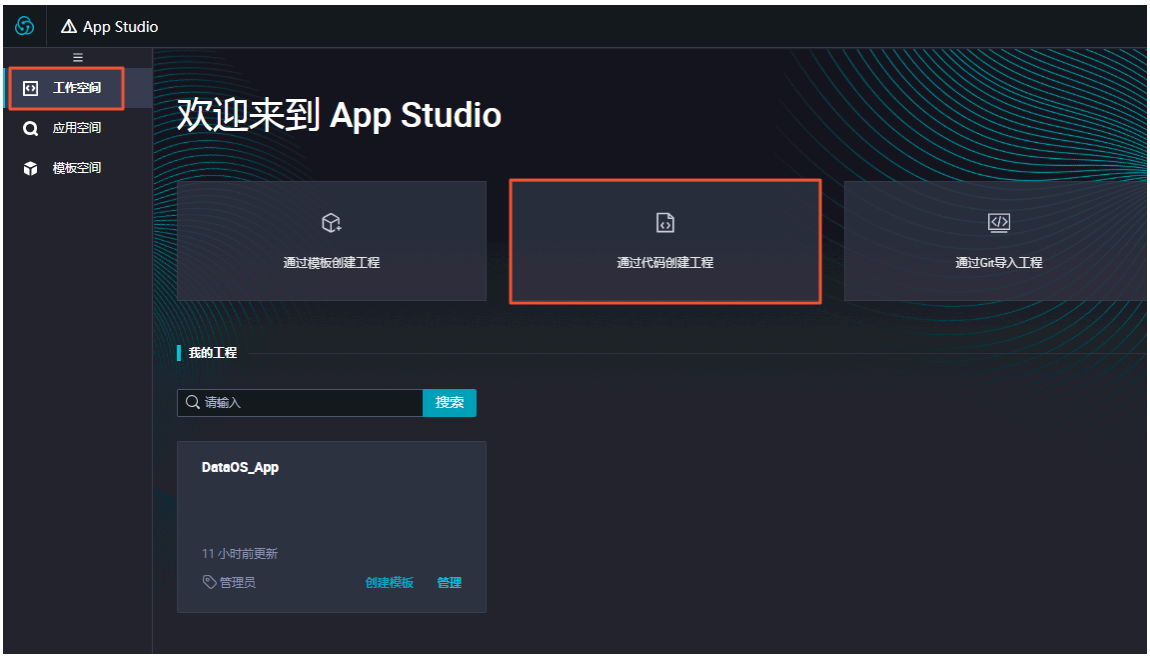
  拖动一个表格组件到画布，单击表格组件，右边会弹出组件配置区，可以对组件的属性、样式进行配置，也可以进行组件联动配置。



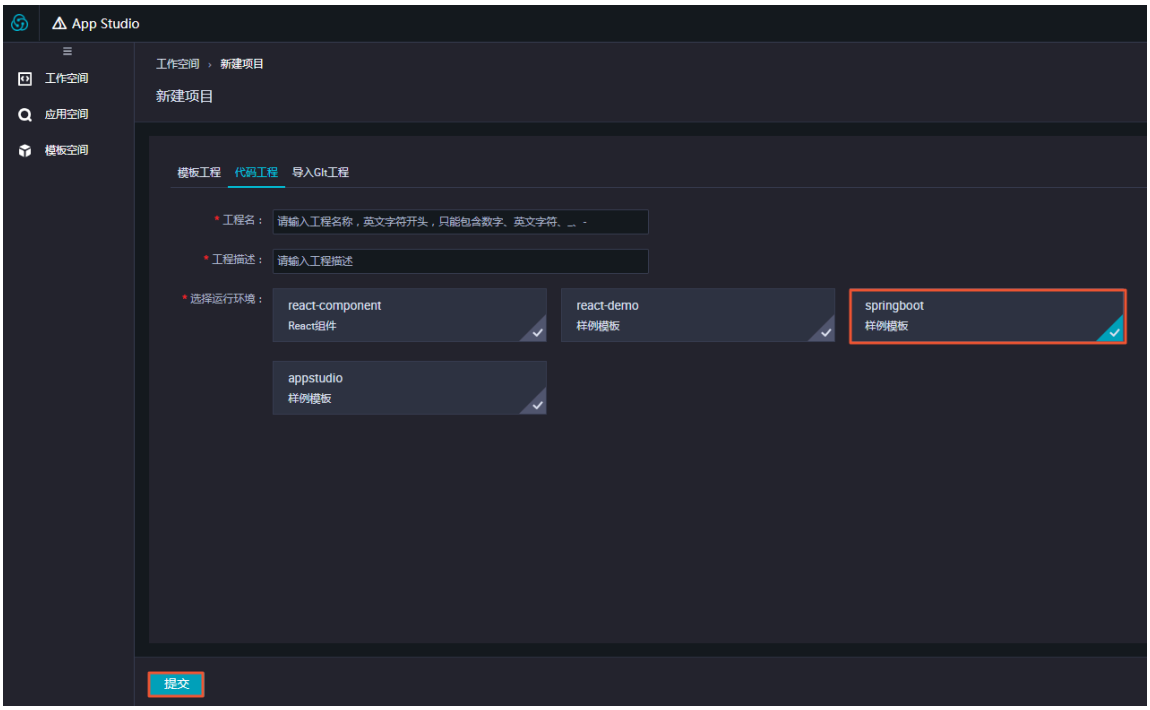
创建后端工程

1. 基于样例工程新建工程。

   a. 进入App Studio页面，单击工作空间页面的通过代码创建工程。



   b. 填写新建项目对话框中的工程名和工程描述，选择运行环境为springboot样例模板。

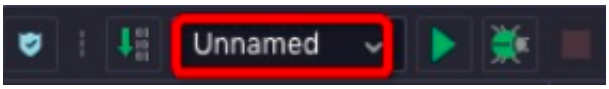


   c. 配置完成后，单击提交。

2. 配置运行参数。

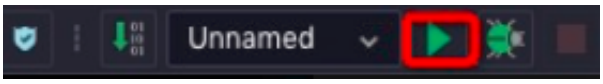
   填写好配置的名称，选择运行的main函数，选择机器规格，单击OK即可完成配置。

   您可以通过左边的添加按钮添加多个配置，运行时选择不同的配置运行。

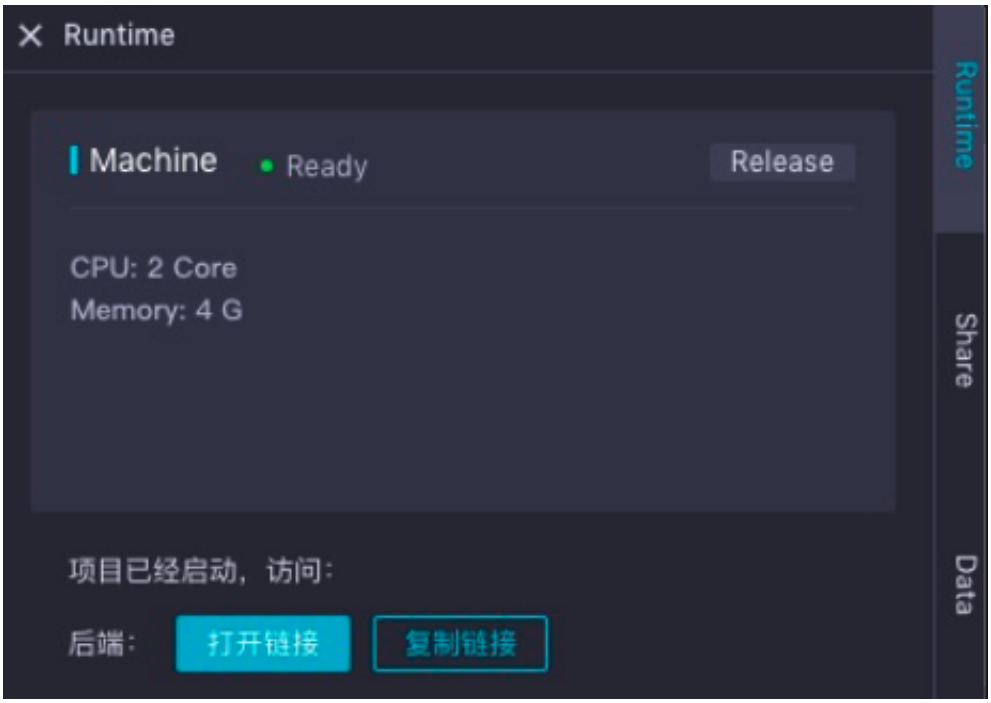


3. 运行工程

单击红框中的运行图标开始运行工程。



第一次运行需要分配机器、初始化语言服务，需要较长时间运行完成，完成后会弹出runtime窗口，展示访问链接。



4. 访问工程

单击打开链接，即可访问工程。

在链接中加上/testapi并刷新页面。

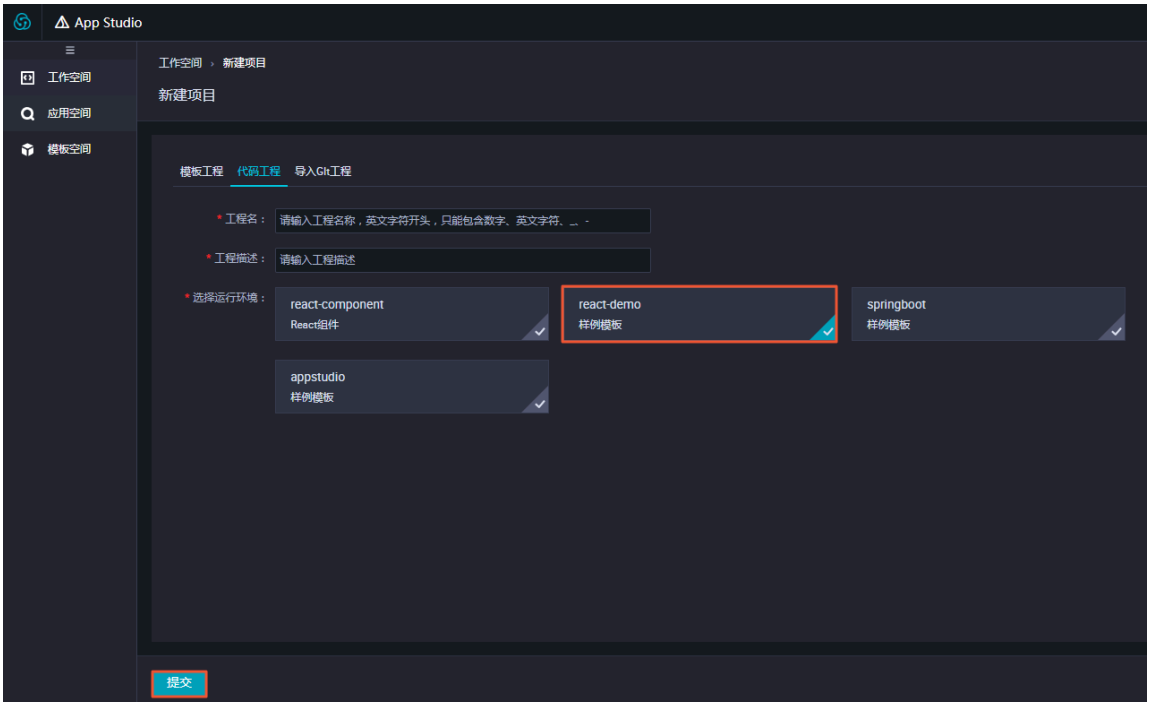
创建前端工程

App Studio提供完善的前端开发能力，支持与本地一致的前端开发体验。您可以在App Studio中创建前端工程，以自己熟悉的方式进行HTML、CSS、JS和React的开发，并且您在开发过程中不需要掌握和理解新的概念。

1. 基于样例工程新建工程。

   a. 进入App Studio页面，单击工作空间页面的通过代码创建工程。

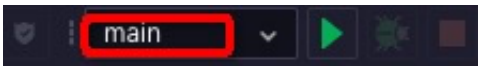
   b. 填写新建项目对话框中的工程名和工程描述，选择运行环境为react-demo样例模板。

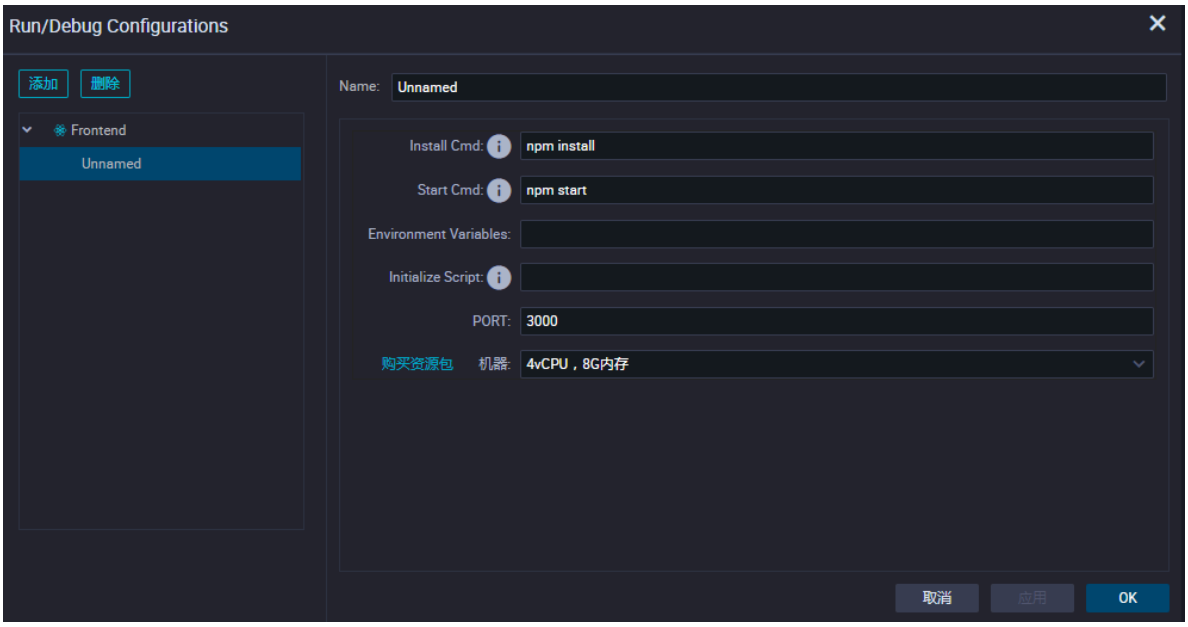


   c. 填写工程名和工程描述，单击确认。

2. 配置运行参数。

   您可以选择机器规格、配置端口，如果没有特殊需求可以直接使用默认的配置，单击OK即可。

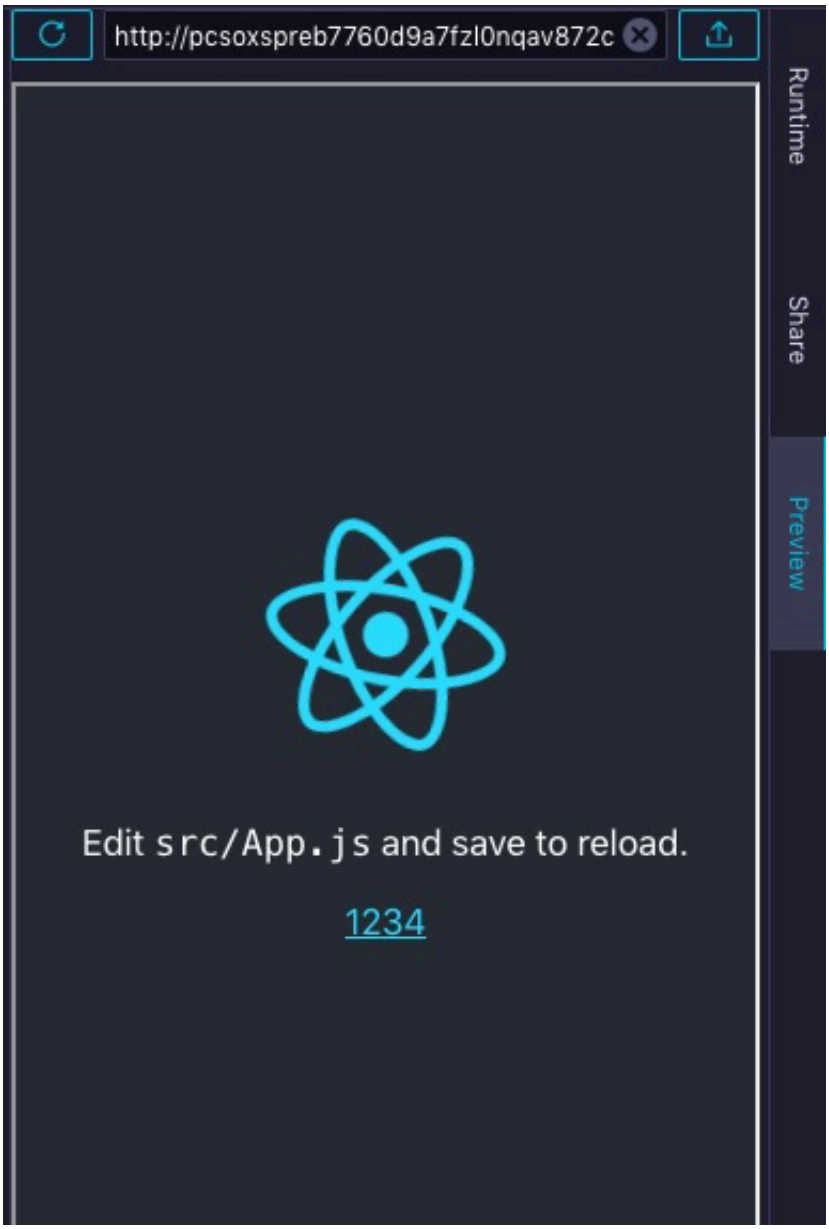




3. 运行工程

单击右上角的运行图标开始运行工程，目前支持以`tnpm start`的方式启动前端工程，配置了*webpack-dev-server*的工程可以无缝对接运行。

启动运行后，开发者可以在日志中看到依赖安装及应用启动的日志，运行完成后右边会弹出页面的预览页面。您可以实时修改代码并进行保存，便可实时生效。



4. 访问工程

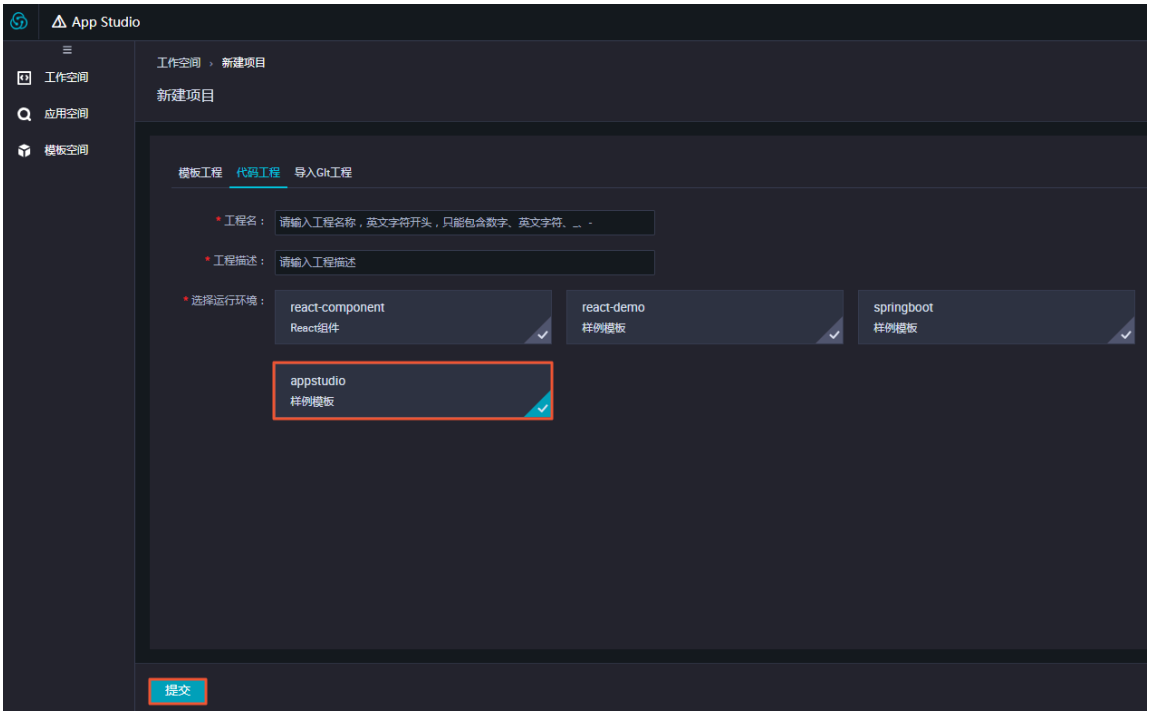
单击链接边的箭头即可打开访问页面，App Studio对于前端工程的编辑开发提供了与本地IDE一致的开发体验，包括HTML、CSS、LESS、SCSS、JavaScript、TypeScript、JSX和TSX等文件的智能补全、函数签名、重构和跳转等功能。同时您不需进行搭建环境、下载依赖等操作，可以在模板基础上进行前端开发。

搭建前端可视化工程

1. 基于样例工程新建工程。

   a. 进入App Studio页面，单击工作空间页面的通过代码创建工程。

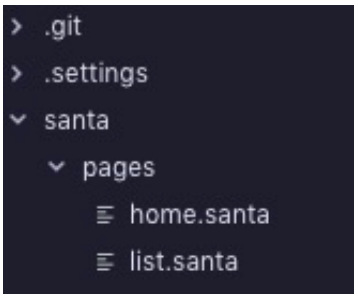
   b. 填写新建项目对话框中的工程名和工程描述，选择运行环境为App Studio样例模板。



   c. 配置完成后，单击提交。

2. 打开home.santa文件。

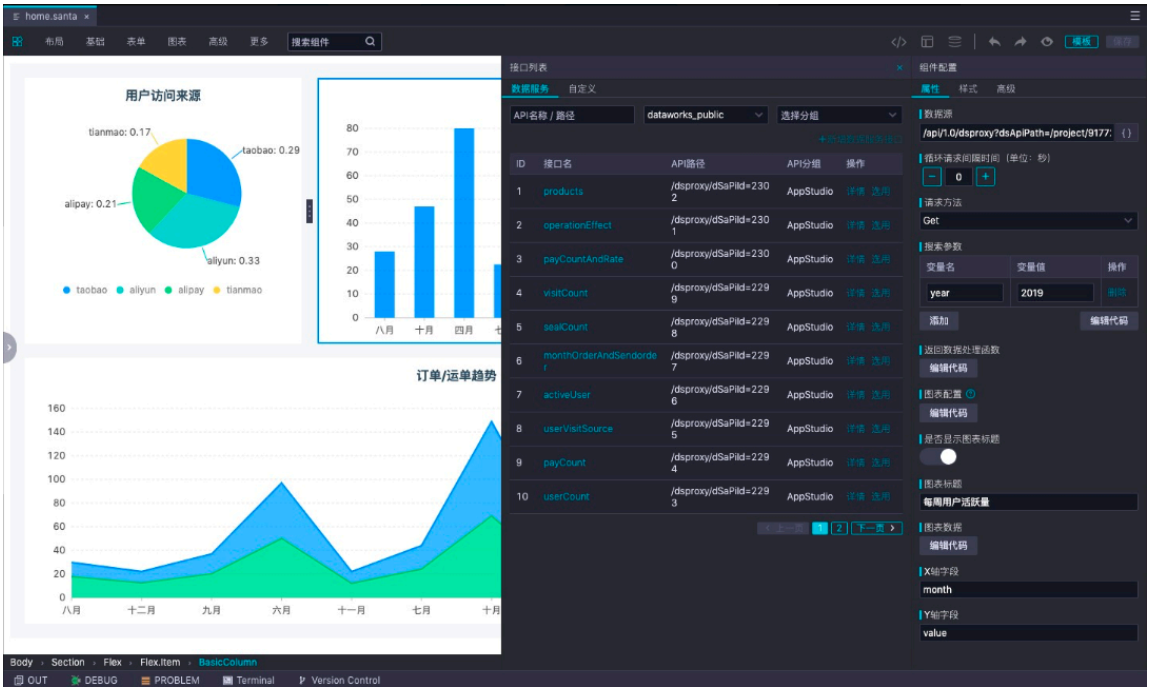
   在santa-pages目录下找到.santa文件，有home和list两个样例页面。



   a. 打开home.santa，是一个简单的报表页面。

   b. 选中一个组件，右边会弹出组件配置。

c. 单击数据源输入框，会弹出接口列表。



App Studio为您提供一些数据服务接口，以便您入门使用。您可以单击+新增数据服务接口前往数据服务中新增接口，通过API路径查看现在的组件对应的接口。

说明:

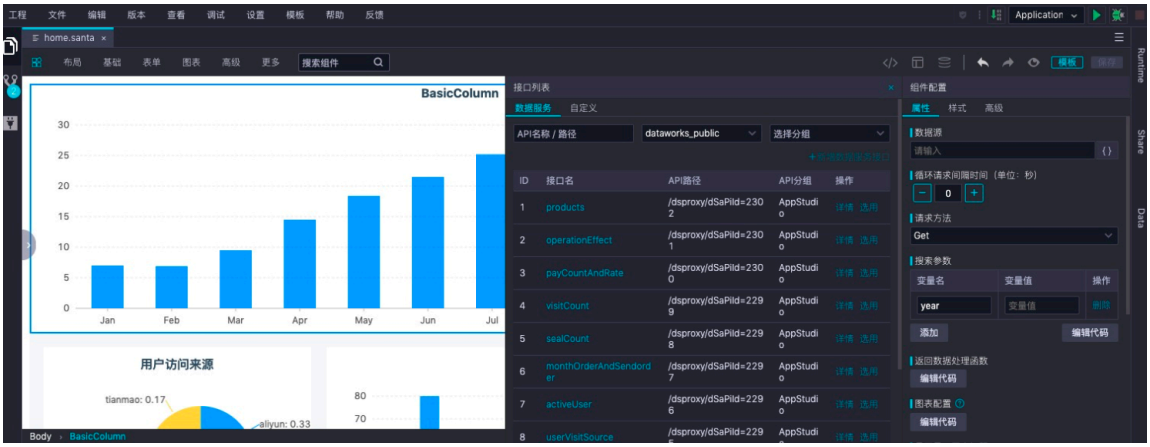
您可以尝试去掉接口自行配置，体验组件配置数据源的效果，也可以对样式进行修改。

3. 添加组件&配置接口。

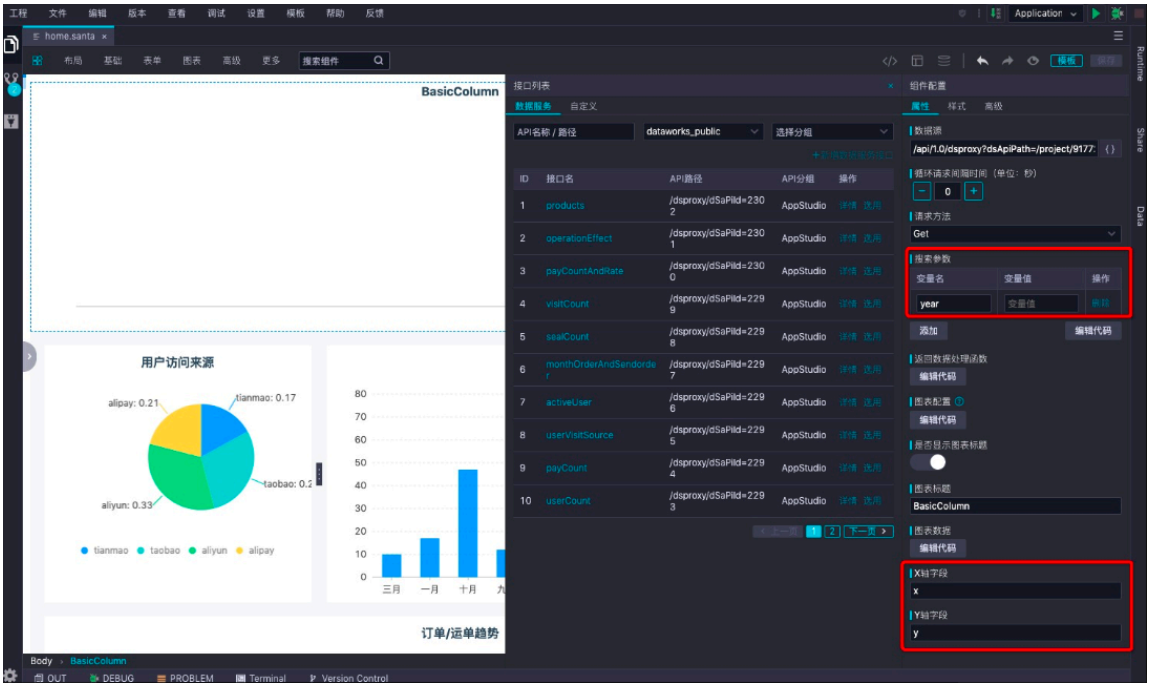
a. 从图表中拖动一个柱状图到画布上。

b. 选中组件，单击弹出的组件配置框中的数据源输入框。

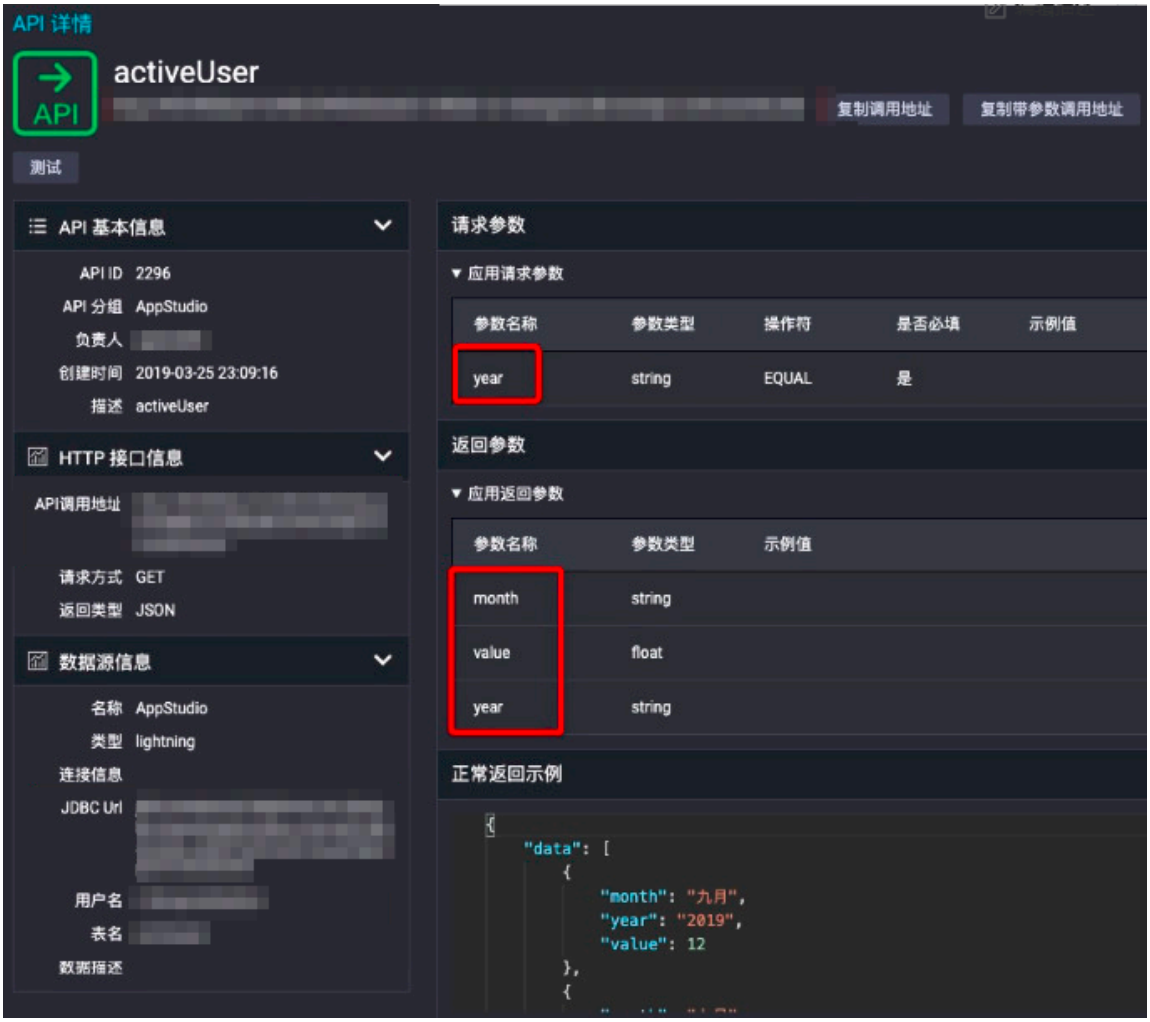
c. 选择第7个接口，单击选用，便成功配置接口。



d. 此时组件中没有返回结果，是因为此接口需要填写入参和返回的列。



您可以单击第7个接口的详情，查看请求和返回的内容。



> **说明:**
> 由于这是示例项目，您会无权访问，建议您搭建可视化工程时，使用自己的账号到数据服务创建接口。

e. 按照下图填写组件配置。

配置完成后，即可看到组件中已有配置好的数据。

4. 打开list.santa文件。

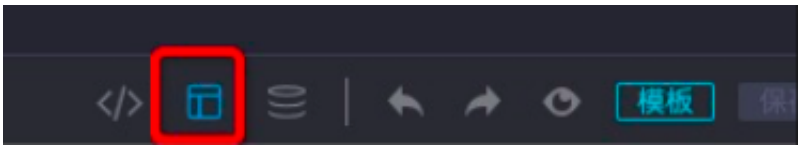
   App Studio可视化搭建不仅可以搭建报表，还可以搭建应用。

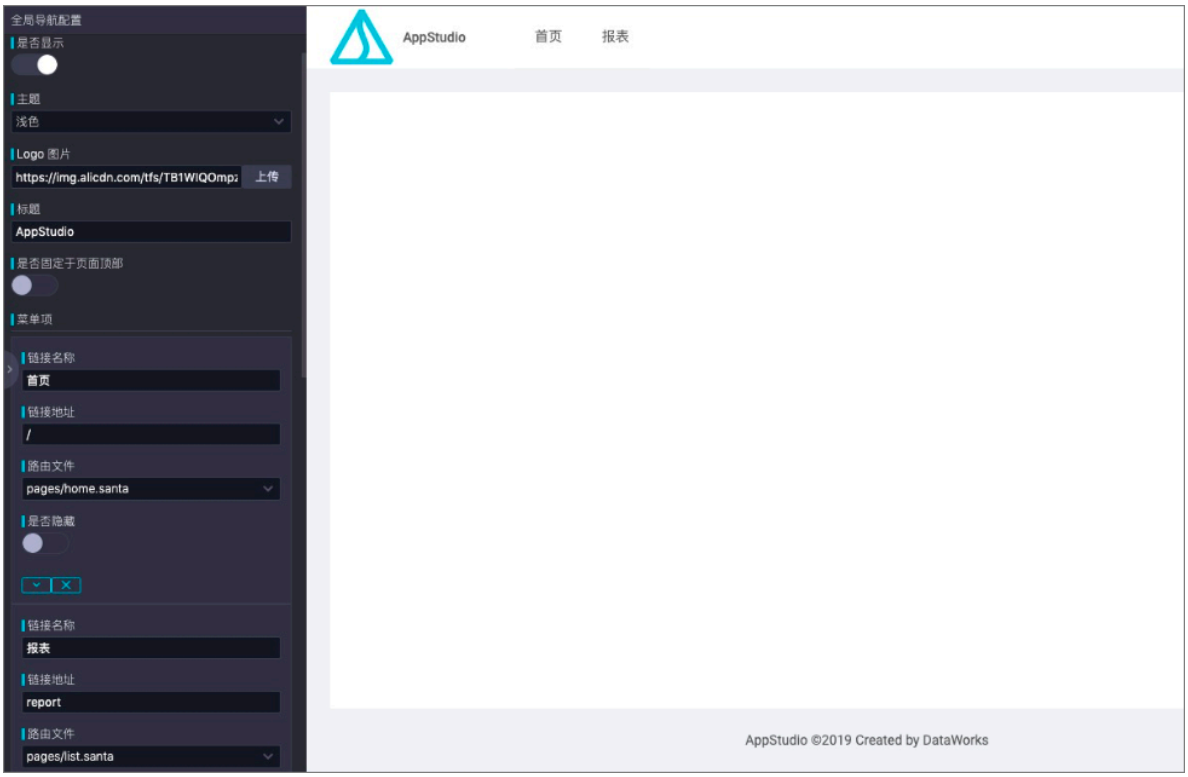
   打开`list.santa`文件，是一个简单的数据应用。其中包括图标、链接、视频、列表和搜索等组件，详情请参见#unique_636。

5. 导航配置。

   您搭建一个应用，通常不会只有一个页面，多个页面之间需要一个导航配置。

   单击右上角的导航配置标识，即可打开导航配置页面。





6. 配置运行参数，可参见搭建后端工程的操作。
7. 运行工程。

   单击右上角的运行标识即可开始运行工程，运行后会弹出Runtime面板，单击里面的前端链接即可访问工程。

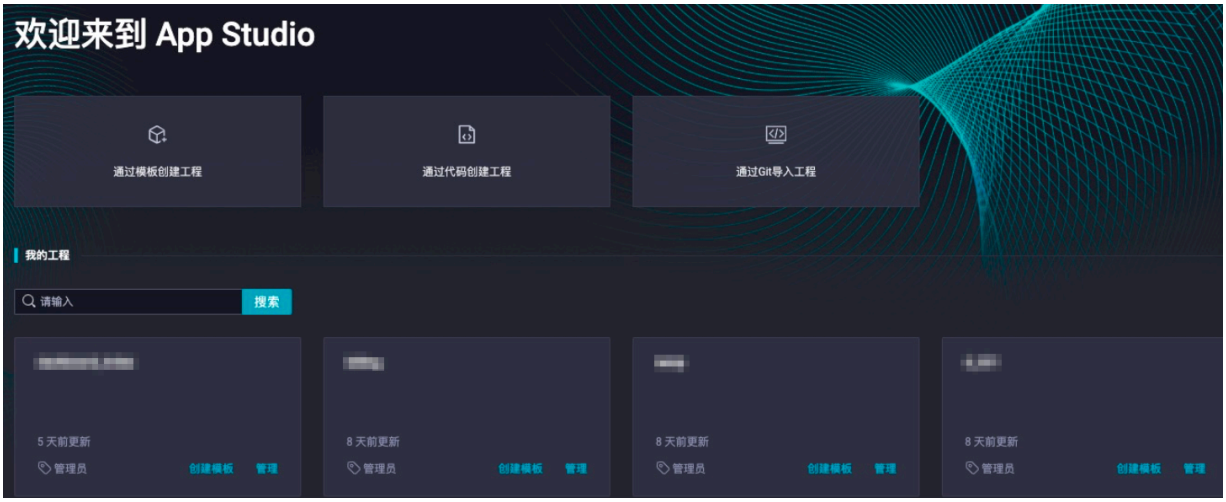
13.4 功能介绍

13.4.1 导航页

13.4.1.1 工作空间

您可在工作空间页面创建和管理工程。

App Studio的工作空间页面，将为您展示当前创建的工程列表，并提供三种创建工程的方式，详情请参见#unique_640。



单击工程卡片，即可进入工程开发页面。您也可单击创建模板或管理，进行相关操作。

创建模板

1. 单击相应工程下的创建模板。
2. 填写生成模板对话框中的配置。

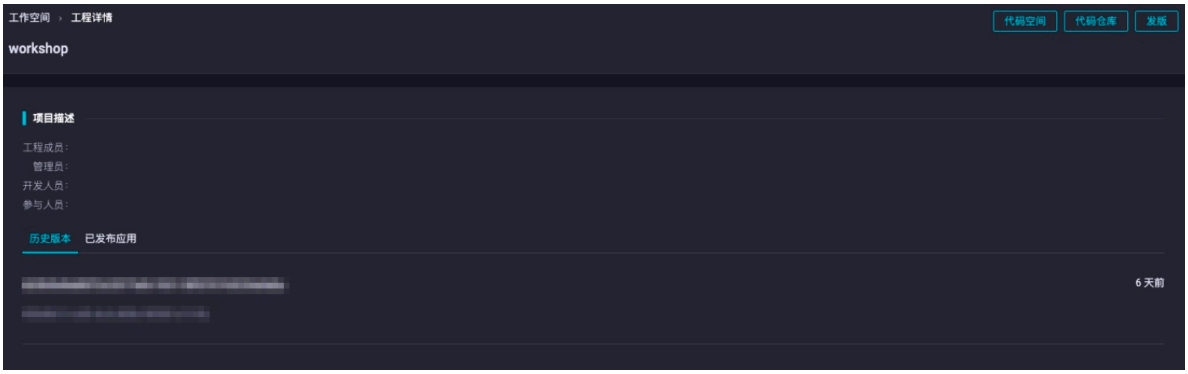
配置	说明
模板名称	输入模板的名称。
模板描述	对模板进行简单描述。
分类	选择模板的分类。

3. 填写完成后，单击生成。

管理

您的工程可以发布为一个应用，为方便您的版本管理，您可将工程发布成不同的版本，然后再进行应用发布。

1. 单击相应工程下的管理，即可进入工程管理页面。



2. 单击右上角的发版，选择要更新版本的应用。

> **说明:**
> 您需要将工程绑定Git仓库后，方可进行发版。

3. 配置完成后，单击更新，即可产生一个新的版本。

13.4.1.2 应用空间

应用空间包括我开发的应用、我分享的应用和第三方应用三大模块。

> **说明:**
> - 仅购买旗舰版的用户，可以查看我分享的应用。
> - 仅购买企业版和旗舰版的用户，可以查看第三方应用。

我开发的应用

我开发的应用页面为您展示已开发的应用，您可以对应用进行发布，也可以通过部署控制台进入应用运维页面。

说明:
分享入口仅对购买旗舰版的用户可见。

部署控制台

单击相应应用下的部署控制台，即可进入运维页面。

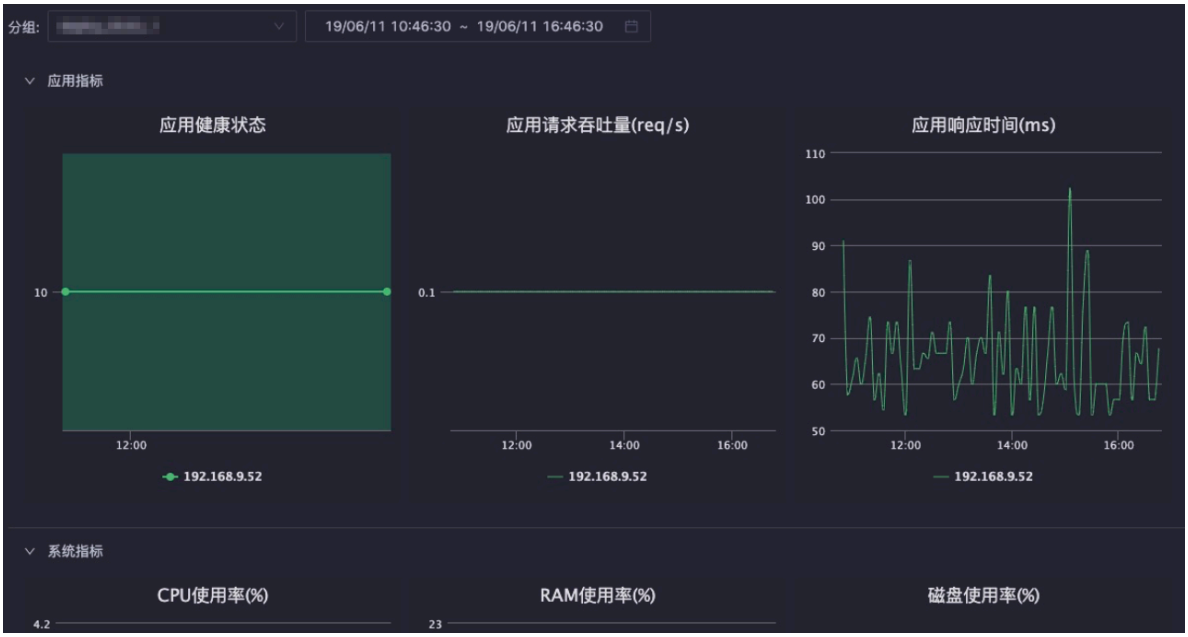
运维页面为您展示所有应用的运维情况，您可以在左侧下拉框选择需要查看的应用。

- **概览**

  概览页面为您展示应用信息、应用状态、分组信息、机器信息、分组列表和机器列表等信息。

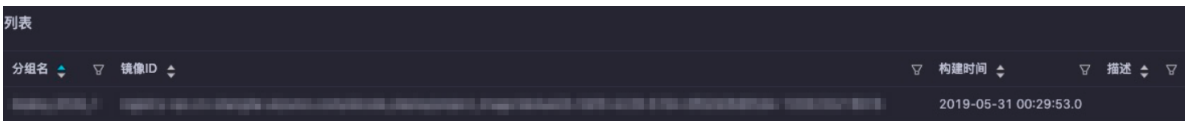
- **监控**

  监控页面为您展示应用的详细运维指标，包括3个应用指标、8个系统指标和7个JVM指标。



- **镜像**

  镜像页面为您展示每个分组使用的镜像和构建时间。



- 变更

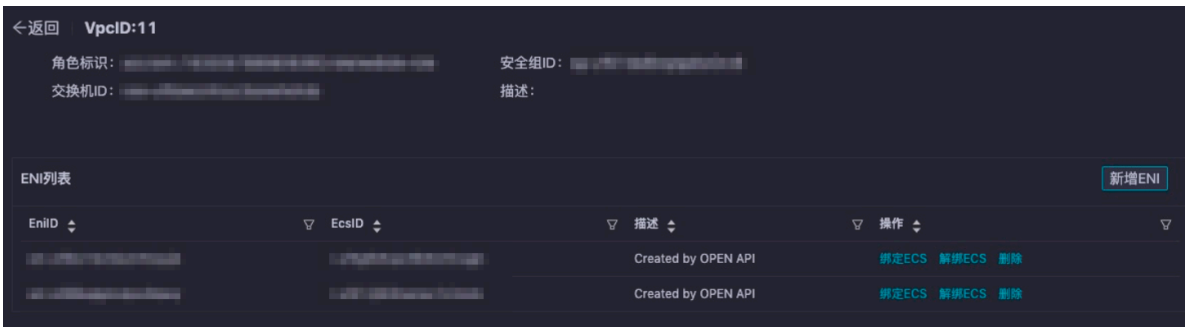
  变更页面为您展示进行的部署、应用扩容或机器下线等操作。单击变更单ID，即可查看变更详情。

  当应用正在部署时，可以在此查看详细的部署信息和日志。

- 资源

  资源页面为您展示您所购买的VPC。购买VPC后，需要在此进行新增操作。

  单击相应的ID，即可进入VPC详情页。



- 操作

  您可以进行应用重启、机器重启、机器下线和应用扩容四项操作。

  - 应用重启

    在应用重启对话框中，对当前操作进行简要描述。单击执行，即可重启整个应用。

  - 机器重启

    在机器重启对话框中，选择分组和机器，并进行简要描述。单击执行，即可对当前应用的某个分组下的某台机器进行重启。

- **机器下线**

  在机器下线对话框中，选择分组和机器，并进行简要描述。单击执行，即可将当前应用的某个分组下的某台机器移除，放回资源池中。

- **应用扩容**

  在应用扩容对话框中，选择扩容分组、可用机器，并进行简要描述。单击执行，即可将您的资源池中的机器，加入到当前应用的某个分组下。

发布

单击相应应用下的发布，即可进行发布操作，详情请参见#unique_642。

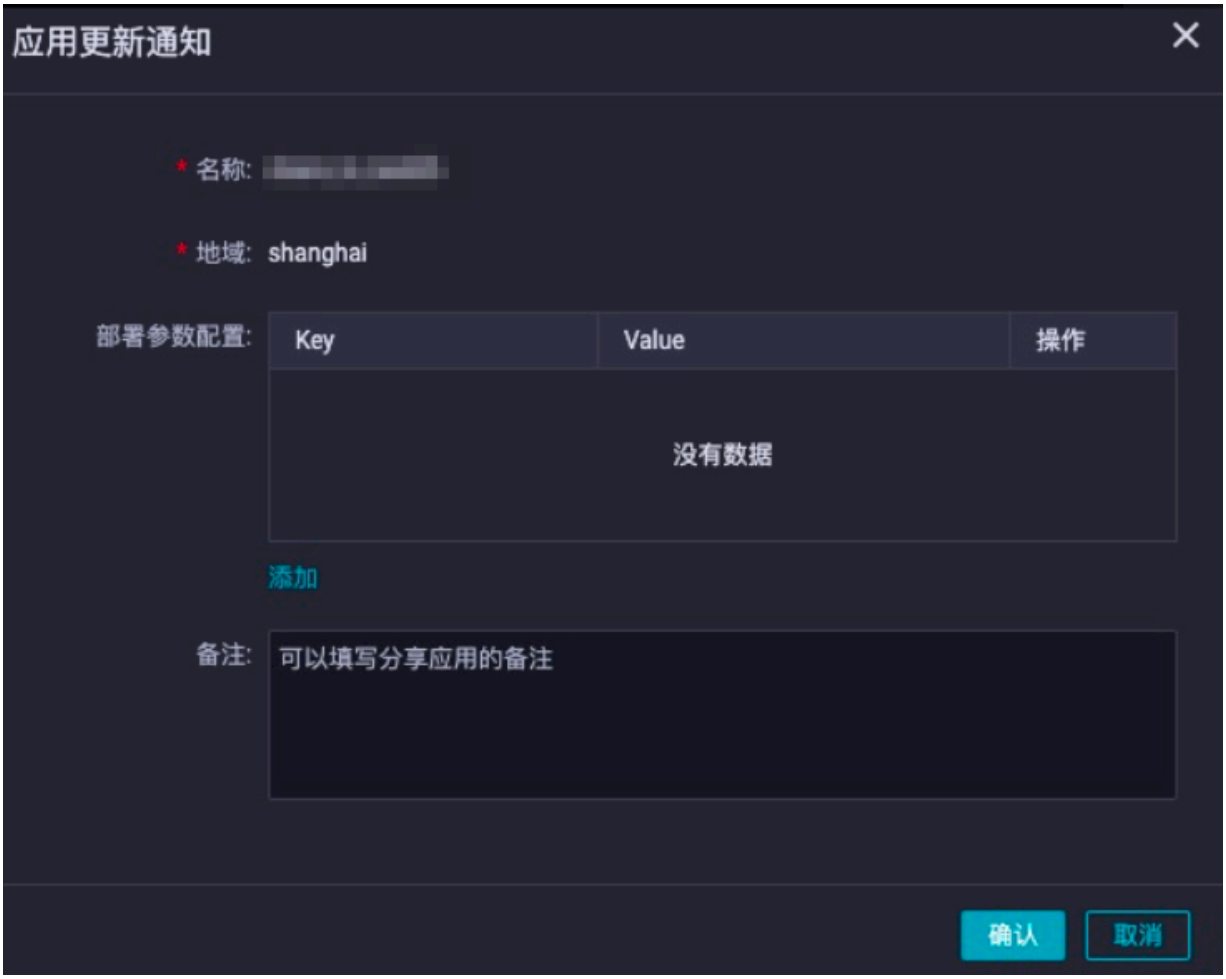
分享

单击相应应用下的分享，购买企业版及以上版本的用户，可以将应用分享给其他用户。分享成功后，您可以在我分享的应用列表中进行查看，对方可以在第三方应用的列表中进行查看。

我分享的应用

进入应用空间 > 我分享的应用页面，即可查看分享过的应用。

单击相应应用下的部署通知，可以将应用的代码更新推送给被分享的用户，进行应用部署。



第三方应用

进入应用空间 > 第三方应用页面，可以查看别人分享给您的应用，并进行部署和发布等操作，操作方式和应用空间一致。

13.4.1.3 模板空间

模板空间为您展示所有通过工程创建的模板。

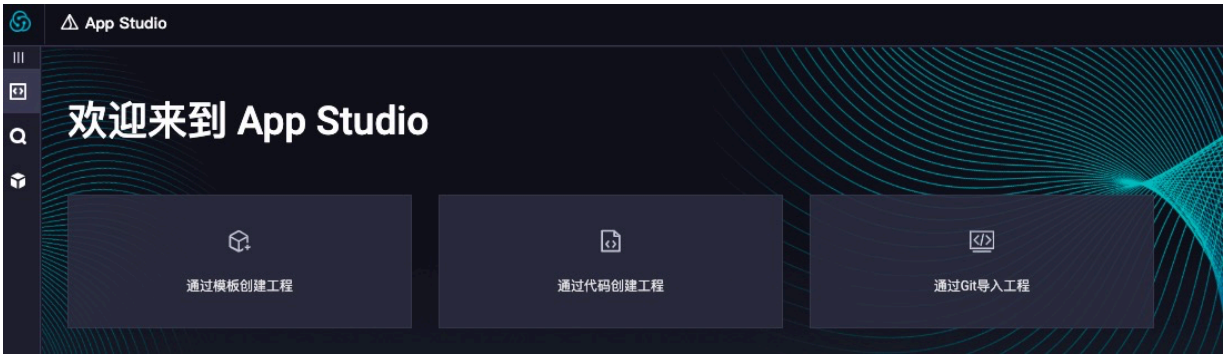
您可以单击相应的模板卡片，进入模板详情页面。然后单击代码空间，即可查看模板相应的工程代码。

您也可以直接单击相应模板下创建工程，即可跳转至通过模板创建工程页面，基于当前模板创建工程。

13.4.2 工程管理

本文将为您介绍如何新建和管理工程。

您可以通过模板、代码和Git导入三种方式新建工程。



通过模板创建工程

1. 进入App Studio页面，单击工作空间页面的通过模板创建工程。

2. 填写新建项目对话框中的工程名和工程描述，选择相应的模板。

说明:

- 您可以选择自己定义的模板，也可以选择系统提供的模板创建工程。
- 通过模板创建的都是可视化工程。

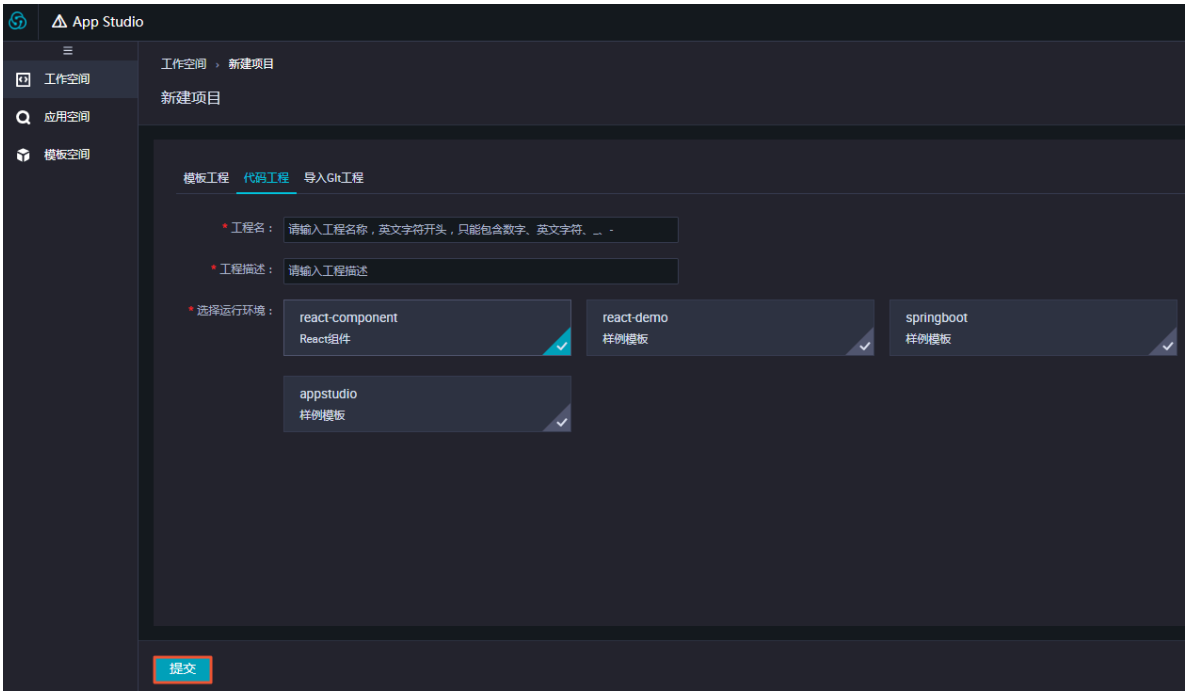
3. 配置完成后，单击提交。

通过代码创建工程

如果想进行纯代码开发的工程，可以通过代码创建工程。App Studio提供了4种运行环境的代码模板，您可以根据自身需求进行选择。

1. 进入App Studio页面，单击工作空间页面的通过代码创建工程。

2. 填写新建项目对话框中的工程名和工程描述，选择相应的模板。

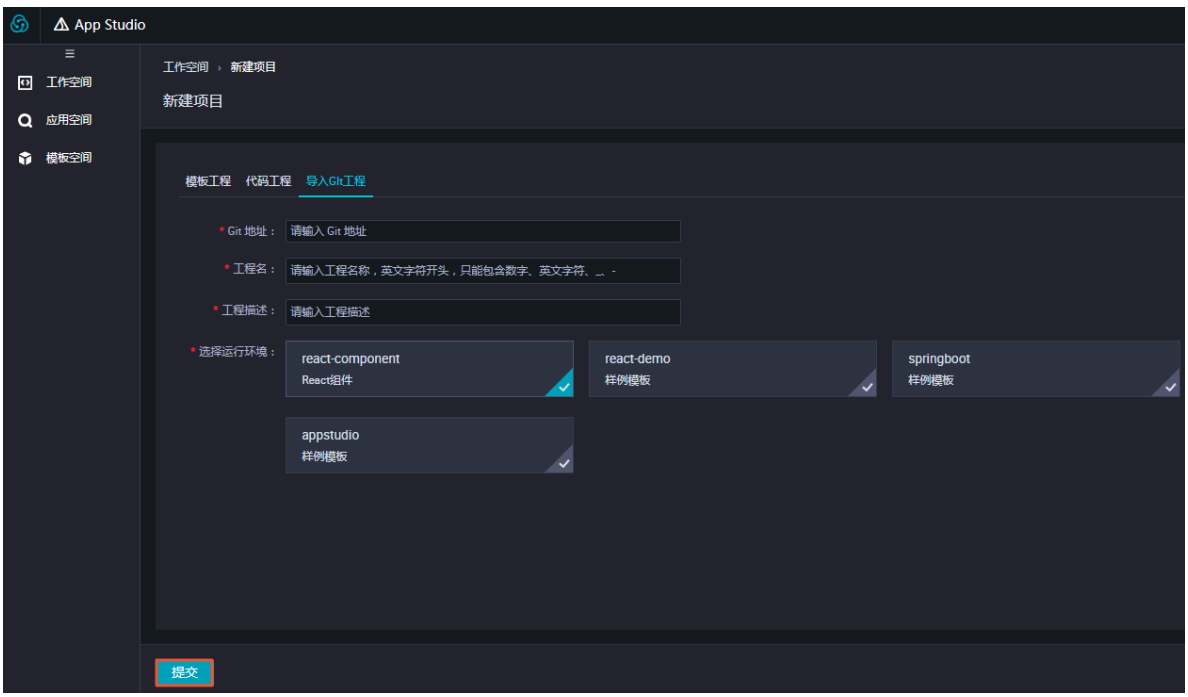


3. 配置完成后，单击提交。

通过Git导入工程

如果您已经有Git代码，可以直接导入Git代码创建工程。此处仅支持Code中的Git代码导入。

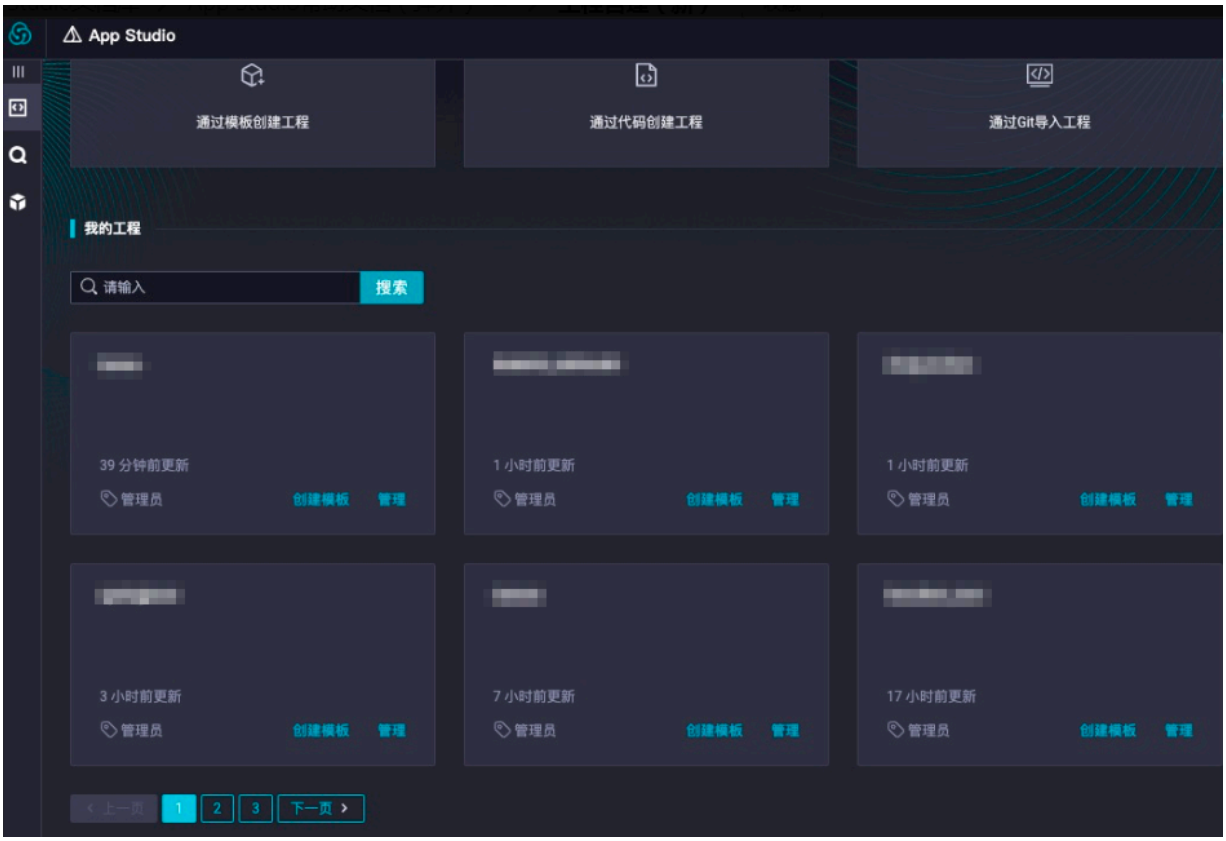
1. 进入App Studio页面，单击工作空间页面的通过Git导入工程。
2. 填写新建项目对话框中的Git地址、工程名和工程描述，选择相应的运行环境。



3. 配置完成后，单击提交。

工程列表

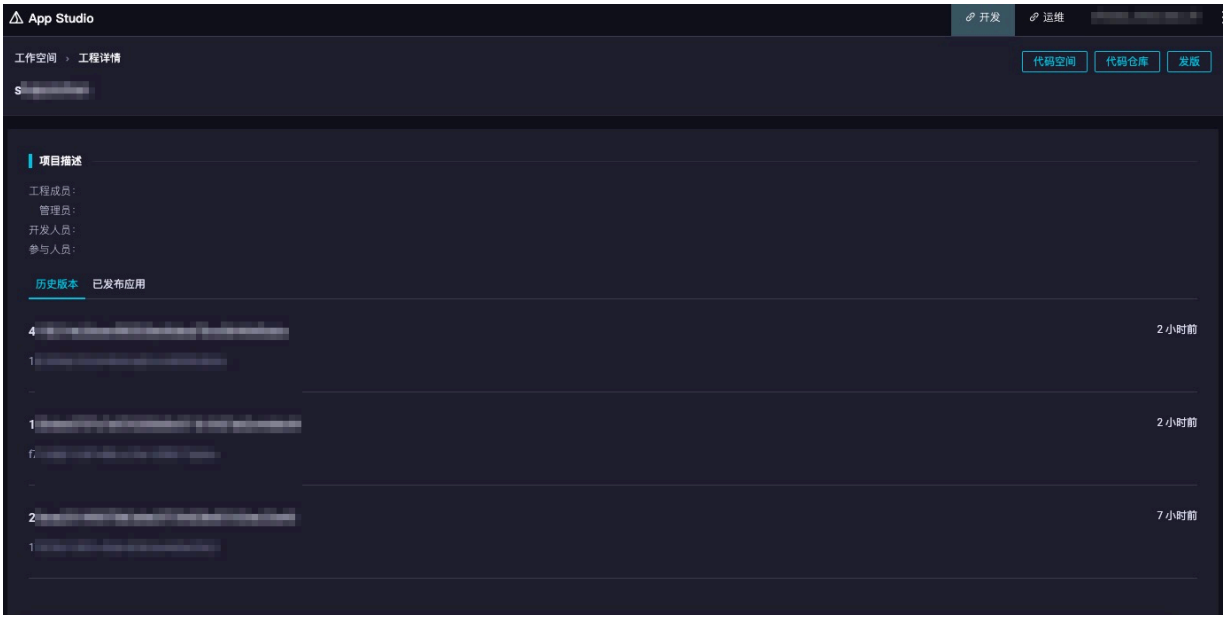
您可以在工作空间页面查看创建的工程。



您可以直接单击相应的工程名称，进入工程编辑页面。也可以单击相应工程下的创建模板，通过工程创建模板。

> **说明:**
> 如果是其他人分享给您的工程，将不能进行创建模板的操作。

App Studio对工程可以进行部署的版本管理，单击相应工程下的管理，即可进入部署版本管理页面。



您可以对工程进行发版，然后进入应用空间页面，部署相应的工程版本。

> **说明:**
> 工程需要关联Git才能进行发版。

13.4.3 版本管理

App Studio集成了通用的Git服务，本文将为您介绍在App Studio中如何使用VCS-git。

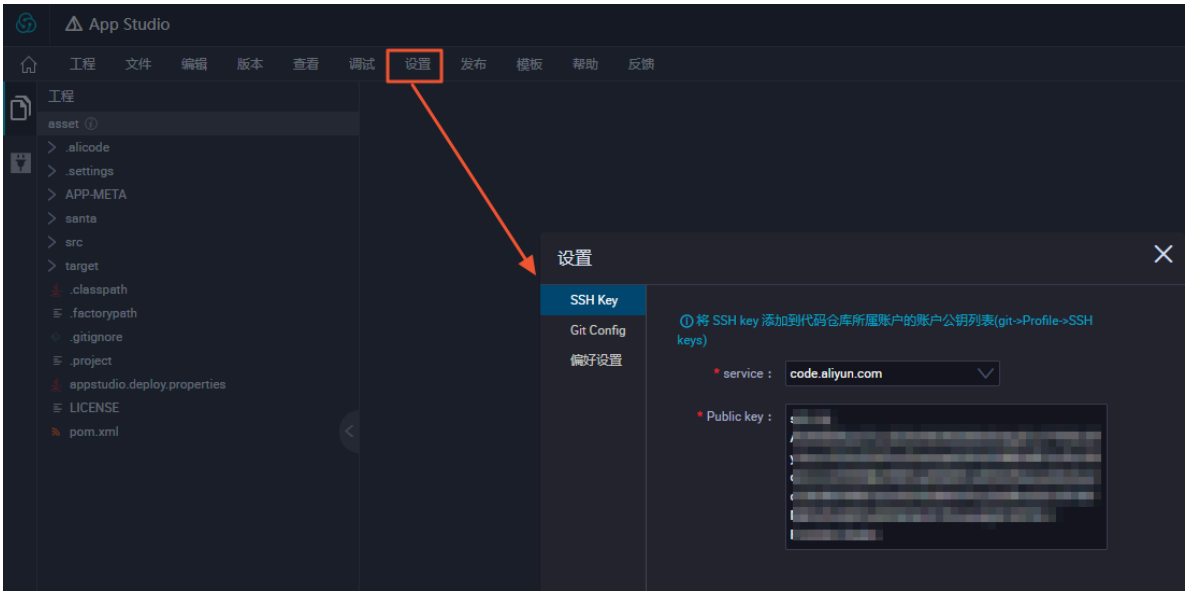
新建工程关联Git系统

1. 新建工程。

2. 录入用户基本信息。

关联Git操作前，需要首先录入用户基本信息。

打开已导入的Git工程，单击菜单栏中的设置，生成一个SSH Key，并根据提示添加到代码仓库所属的账户公钥列表中。



说明:

新创建的工程默认未关联Git服务。如果需要Git服务，请关联当前项目至自己的Git仓库。

3. 新建Git仓库。

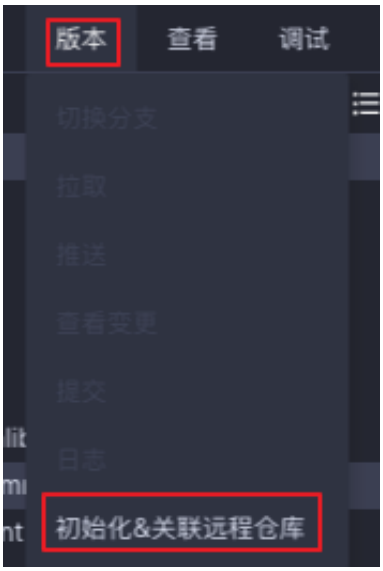
4. 获取当前仓库的SSH地址。

a. 单击SSH，即可获取当前仓库的SSH地址。

b. 单击右侧的复制按钮，即可复制SSH地址至剪贴板。

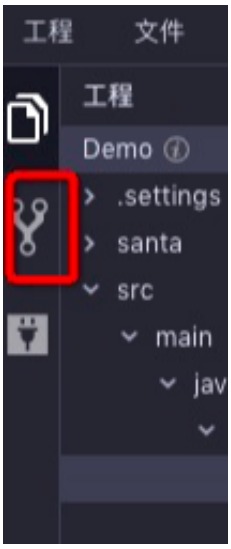
5. 关联Git仓库。

   a. 选择菜单栏中的版本 > 初始化&关联远程仓库。



   b. 填写关联远程仓库对话框中的Git地址，单击提交。

   c. 关联完成后，在App Studio页面的左侧导航栏增加版本控制标签。



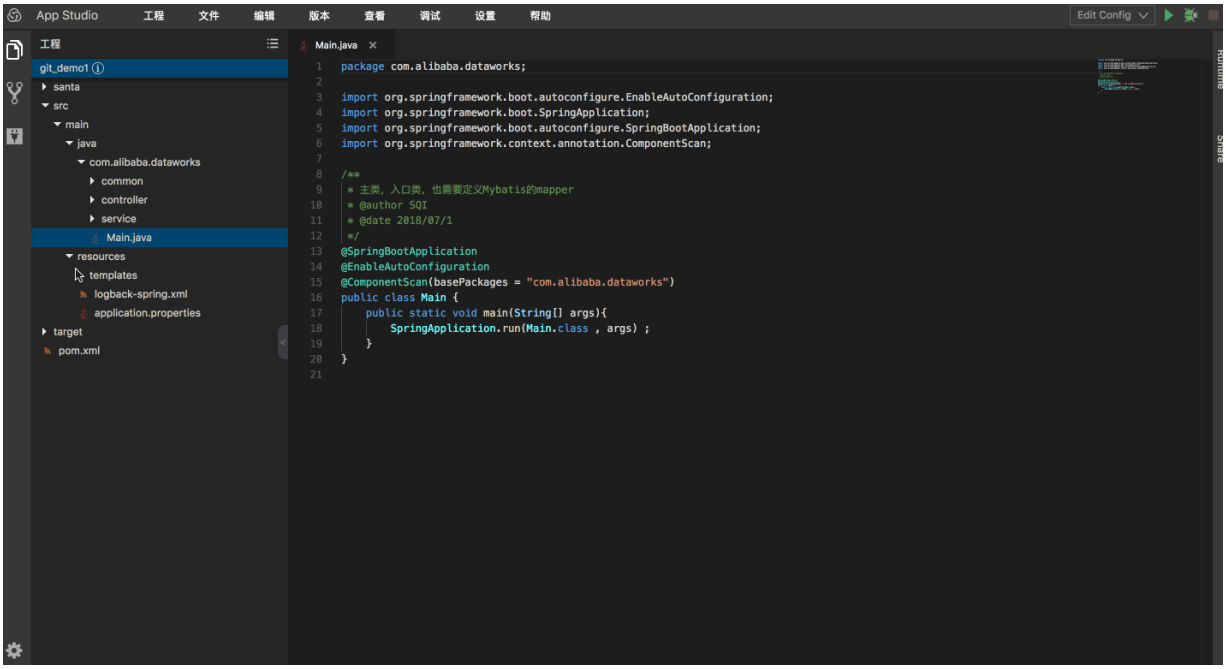
   d. 单击版本下拉列表中的推送，即可将本地代码推送至远程仓库。

Git操作入口

Git相关的操作均集成在左侧的Git面板，以及顶端的版本控制菜单栏中。

Git控制面板

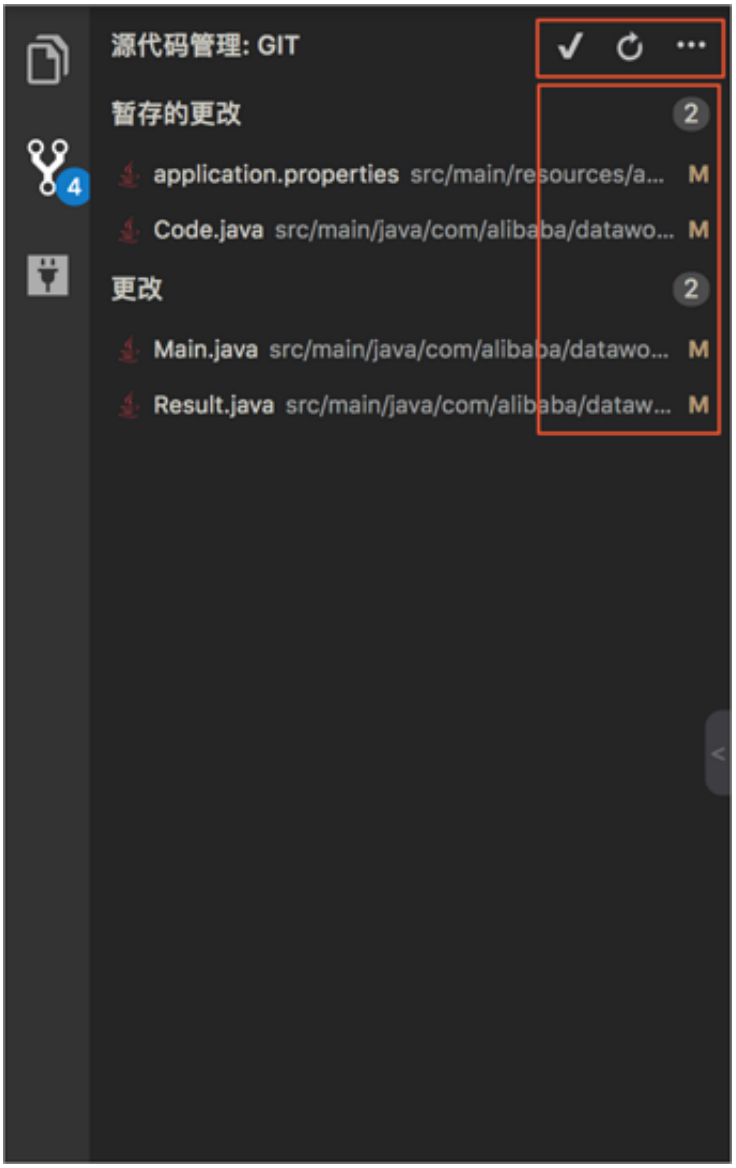
Git控制面板会动态更新文件的编辑状态。



您可以在Git控制面板中，完成基本的`git add/rm/commit/revert`等操作。

Git基本操作

Git面板中以列表形式展示变动的文件，包括文件名、路径、以及右侧支持的基本操作。



如上图所示，红框中包含了可操作按钮，以及文件标识（icon）。

- 源代码管理

  您可在此处进行commit、refresh、pull和push等操作。

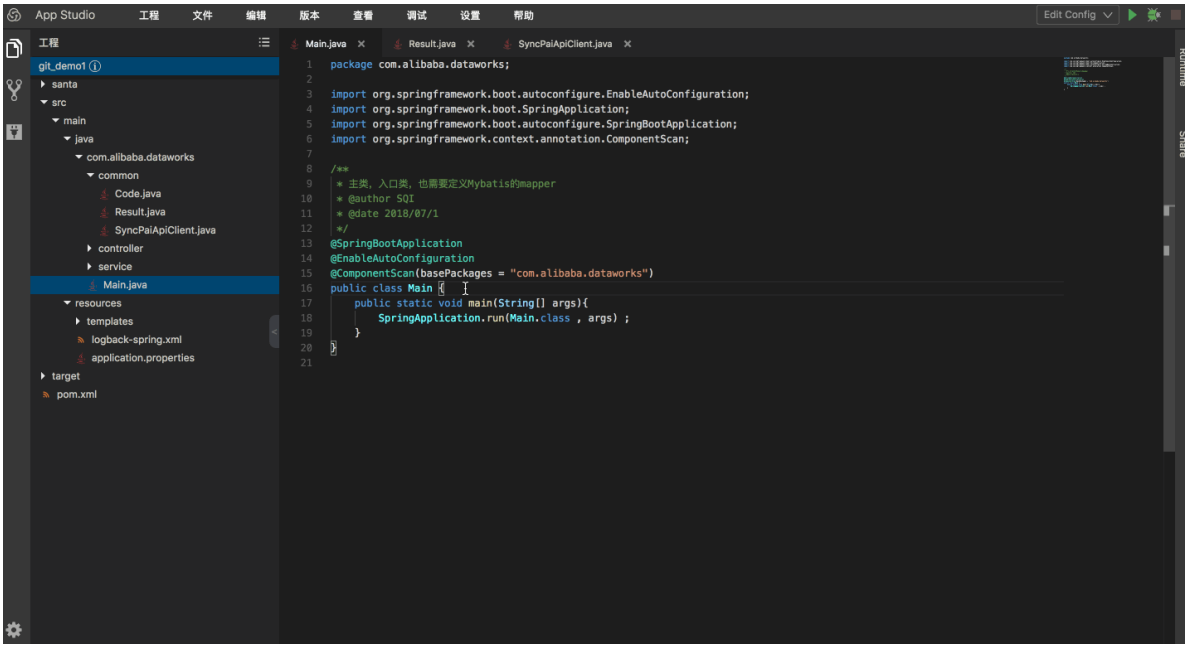
  - commit操作：选择 下的commit&push操作。
  - refresh操作：单击 ，刷新当前控制面板内容，相当于执行`git status`，并刷新界面。
  - pull/push操作：单击 ，根据自身需求选择拉取或推送。

- 暂存的更改
  -  ：放弃所有修改，相当于执行`git reset`。
  -  ：文件数量标签。
  - 

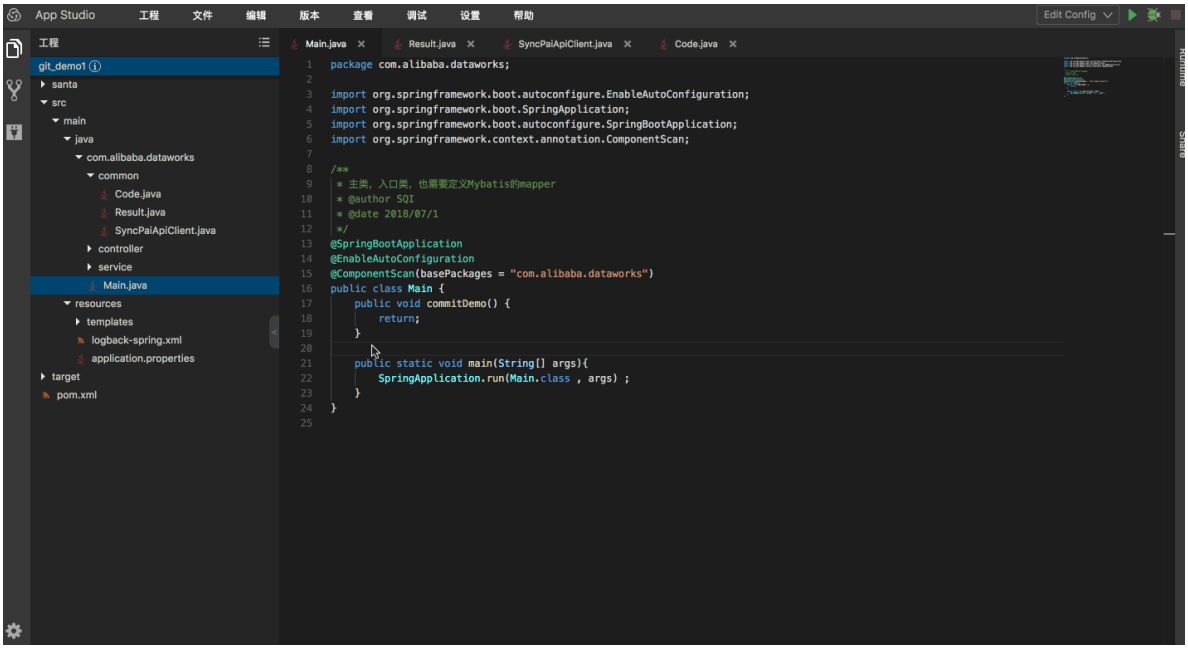
：显示文件更改。
- 更改
  -  ：放弃所有更改。
  -  ：将所有文件添加至缓存区，相当于执行`git add`。
  -  ：文件数量标签。
  - 显示条目包含的操作如下：
    -  ：放弃修改（revert）。
    -  ：暂存更改（add）。
    -  ：文件改动标识（Modified）。

commit/push操作示例

- 示例一



- 示例二

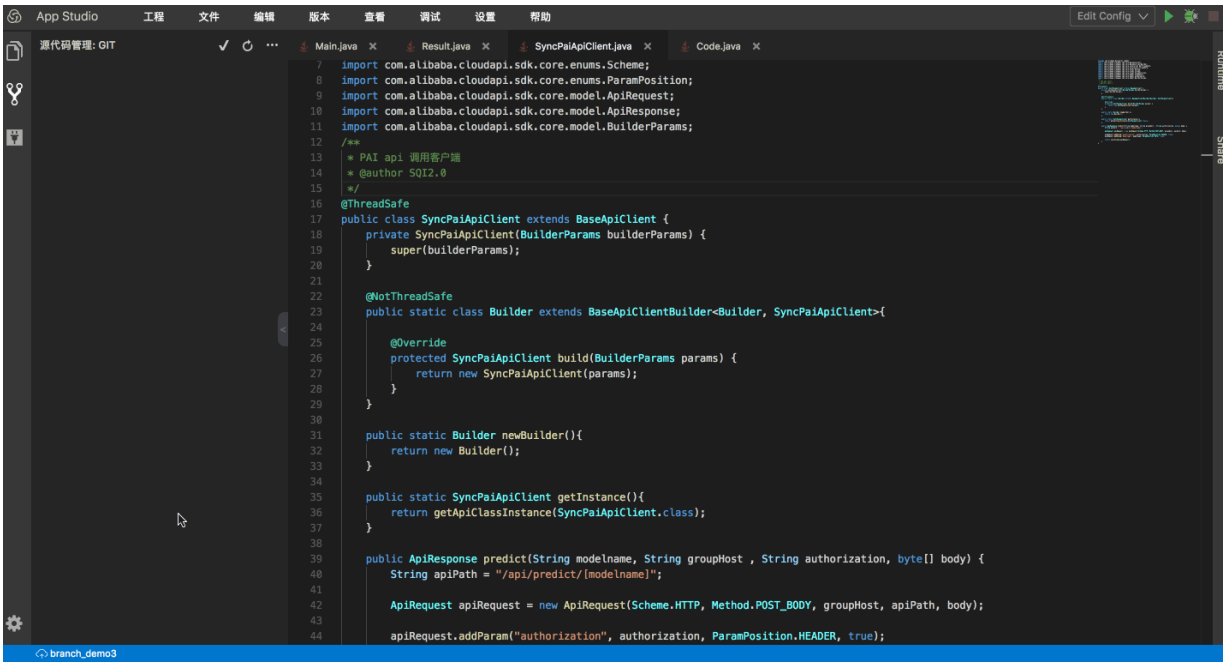


说明:

- Git客户端逻辑一致，您需要主动调用push，本地的代码才会推送至远程仓库。
- 与push同理，您需要主动调用pull，远程仓库的代码才会拉取至本地。

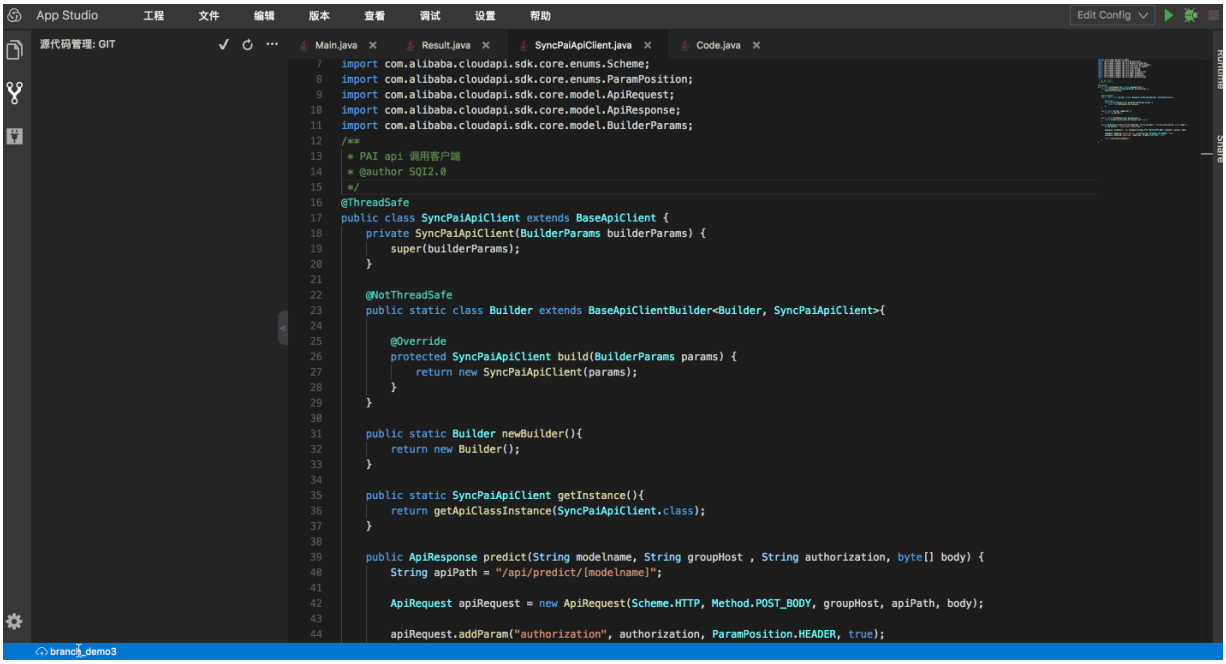
Branch管理

打开分支管理弹窗。



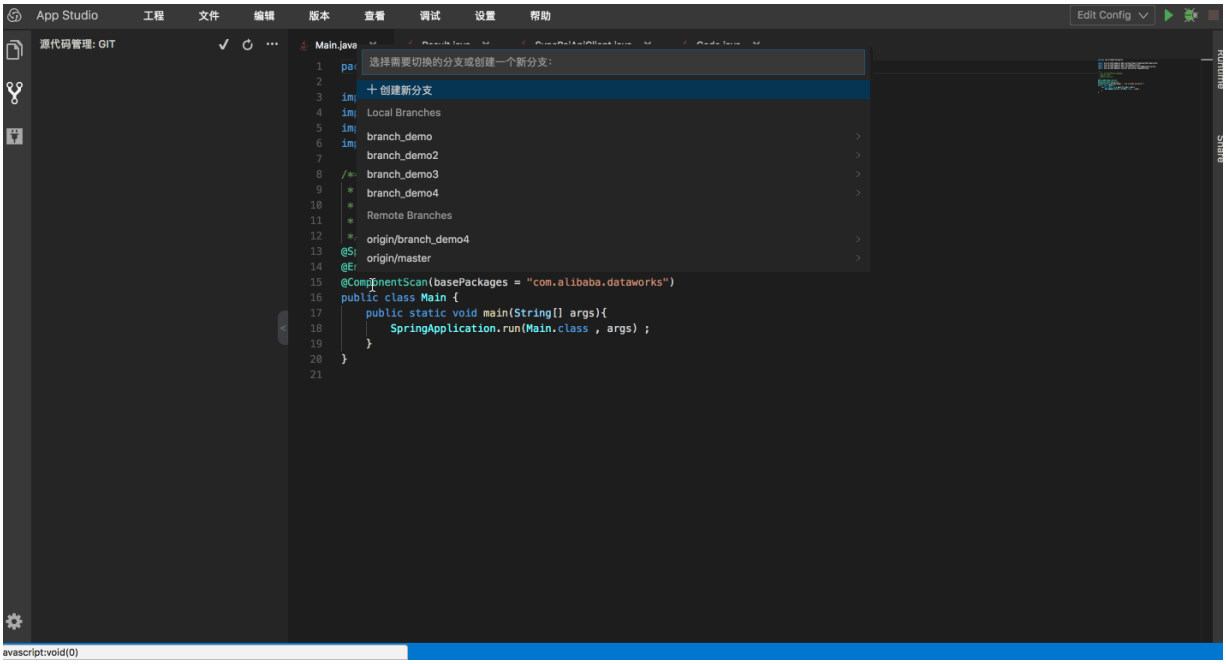
单击窗口下侧状态栏中显示的当前Branch名称，即可弹出Branch管理窗口。

新建本地分支



分支创建后，会自动切换至新创建的分支。

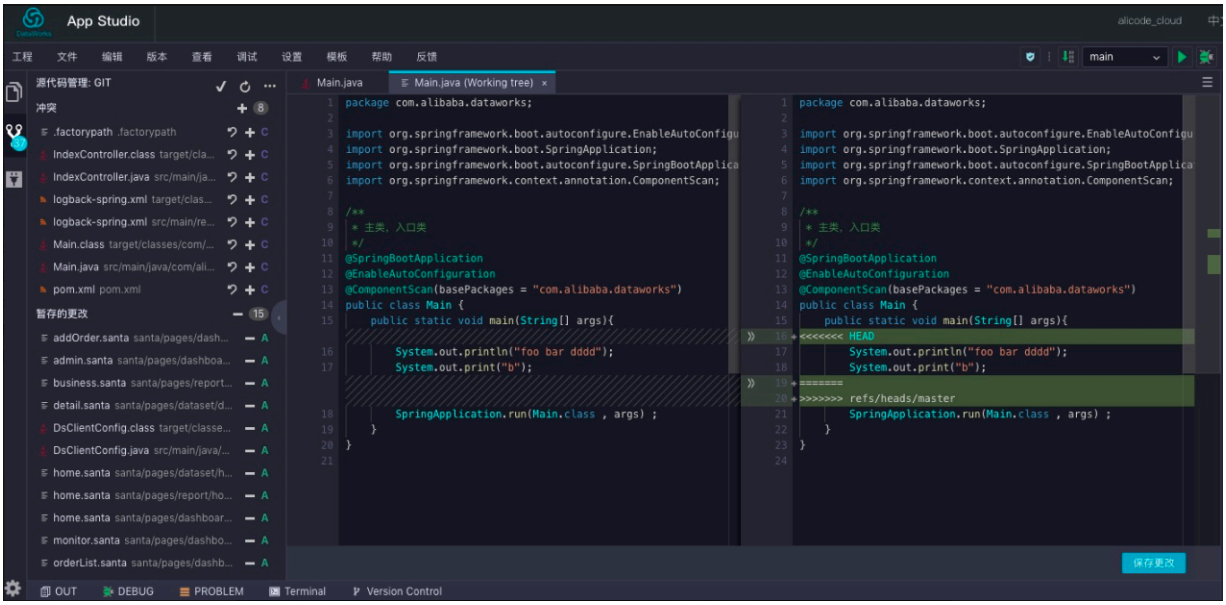
创建/切换/合并分支



说明:

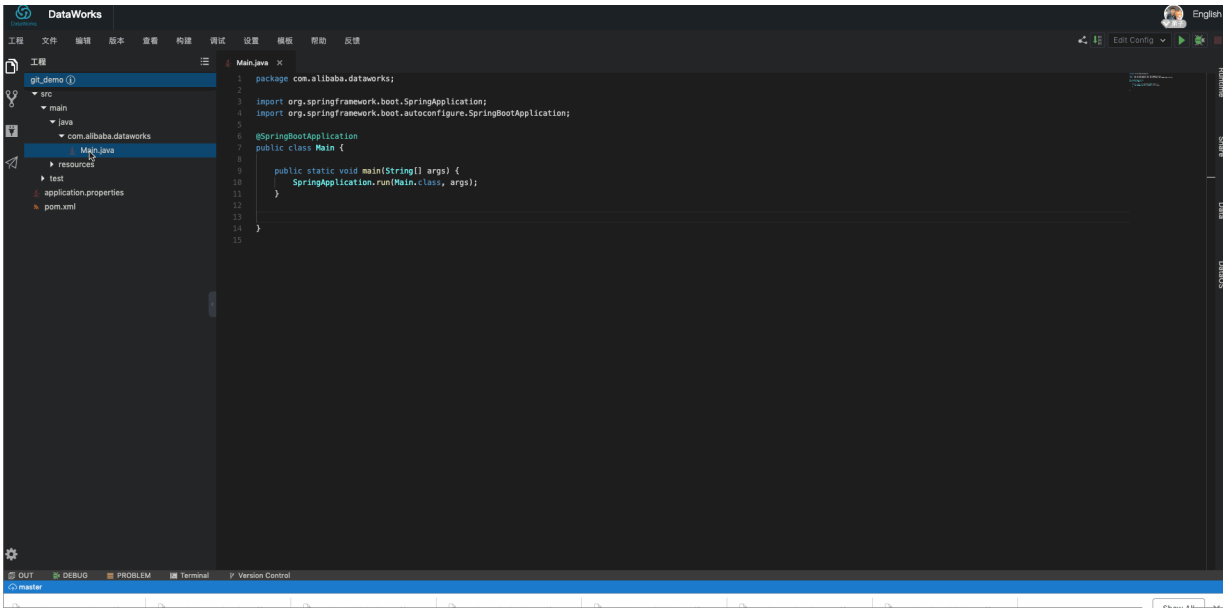
新创建的本地分支，可直接推送至远程，远程分支名与本地一致。

通过Diff页面解决merge conflict



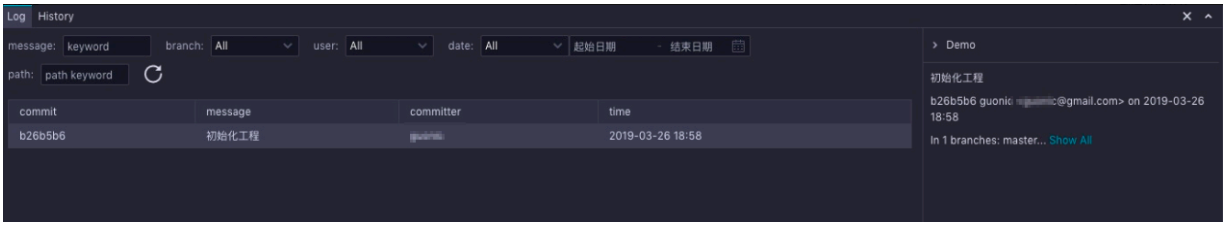
Show History

右键单击文件，选择Git > Show History，即可查看当前文件的历史记录，对特定的commit与当前version进行Diff。



Git Log

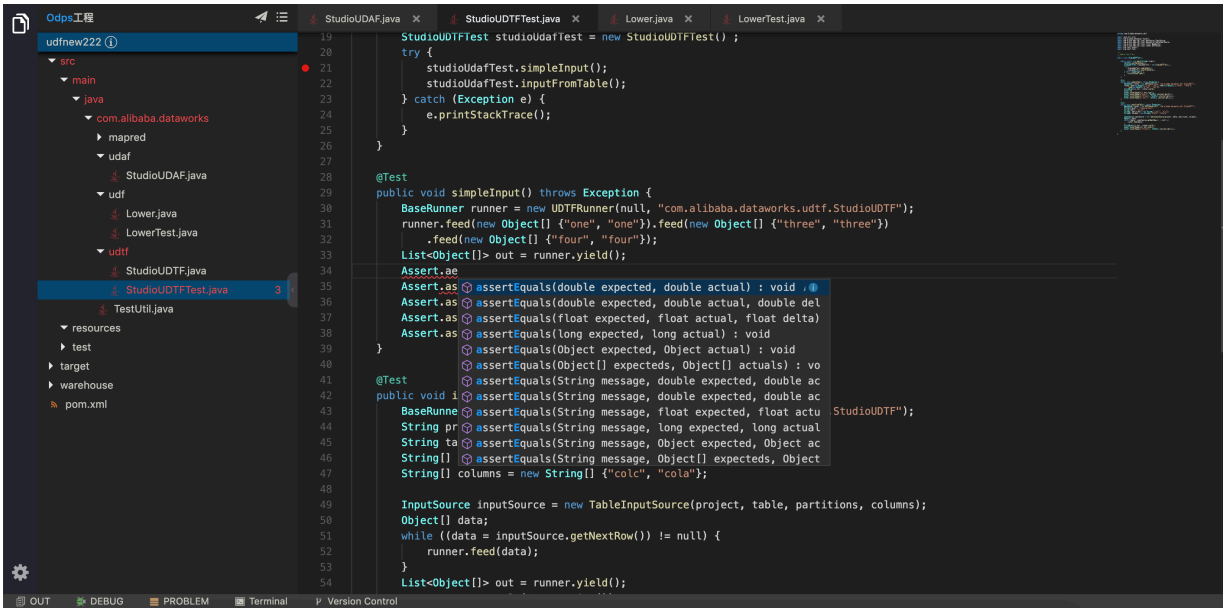
单击菜单栏中的版本 > 日志，打开Git log面板，即可查看提交的信息、时间、作者，您可以通过信息、分支、作者、时间筛选提交日志。



13.4.4 代码编辑

13.4.4.1 代码编辑概述

代码编辑包括自动补全、智能提示、语法诊断和全局内容搜索等常见的IDE具备的功能。



目前语言和对应的功能支持情况，如下表所示。

基本功能	Java	Python	JavaScript/ TypeScript
Completion自动补全	支持	支持	支持
Hover智能提示	支持	支持	支持
Diagnostics语法诊断提示	支持	支持	支持
SignatureHelp函数参数提示	支持	支持	支持
Definition跳转定义	支持	支持	支持
References查找引用	支持	支持	支持
Implementation查找实现类	支持（comming soon）	不支持	不支持

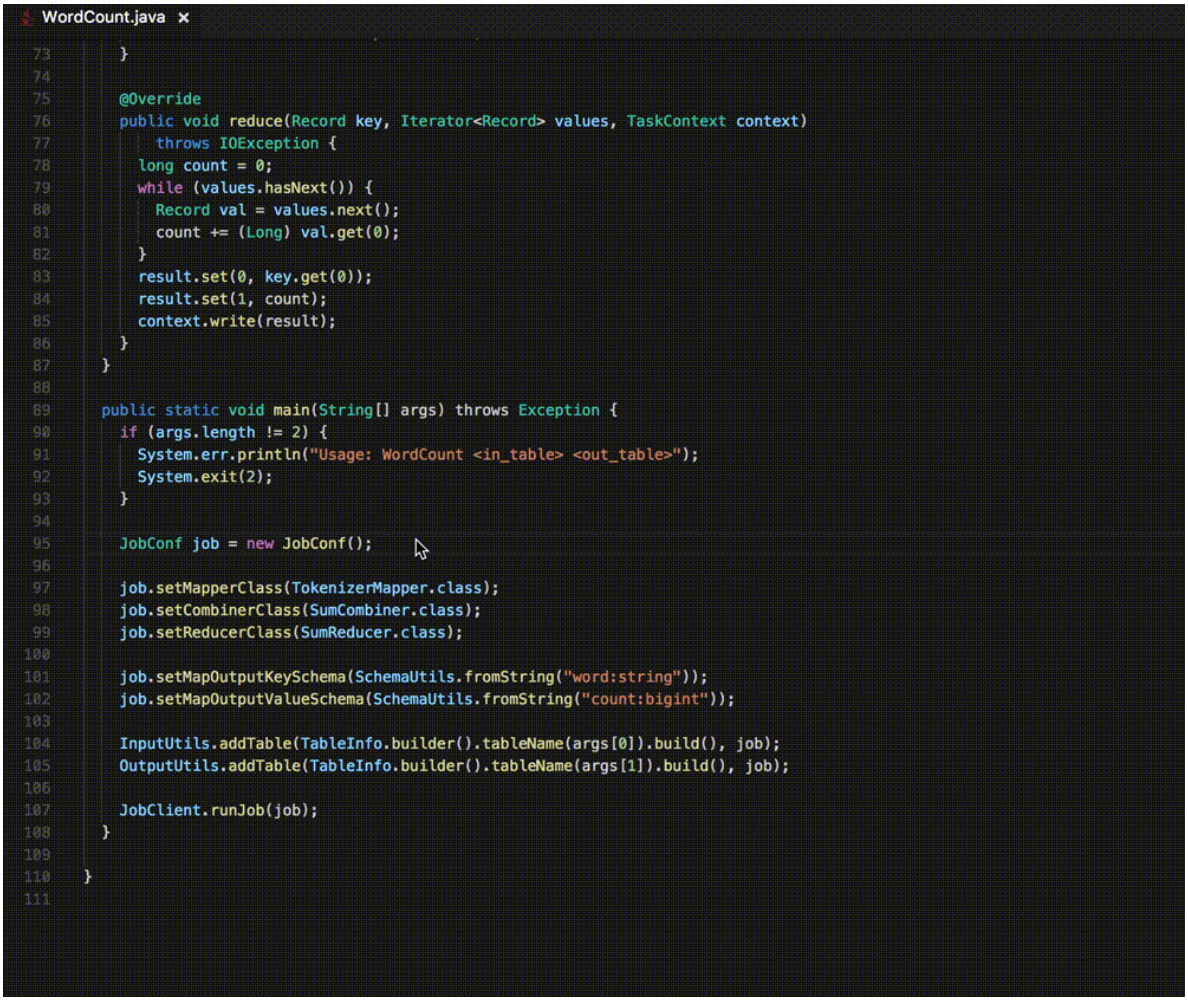
基本功能	Java	Python	JavaScript/ TypeScript
DocumentHighlight变量高亮	支持	支持	支持
DocumentSymbol查找类成员	支持	支持	支持
WorkspaceSymbol全局查找类/函数	支持	支持	支持
CodeAction修复建议	支持（Alibaba Java Guidelines is coming soon）	支持	支持
CodeLens行操作提示	References Implementation	不支持	不支持
Formatting 格式化代码	支持	支持	不支持
RangeFormatting局部格式化	支持	不支持	不支持
FindInPath全局内容搜索	支持	支持	支持

高级功能	Java	Python	JavaScript/ TypeScript
Rename重命名	支持	支持	支持
WorkspaceEdit多文件修改	支持	不支持	不支持
UnitTest单元测试（quickstart）	支持	不支持	不支持
MainClass查找main函数入口	支持	不支持	不支持
MainClassQuickStart快捷运行	不支持	不支持	不支持
ListModules查找所有模块	支持	不支持	不支持

高级功能	Java	Python	JavaScript/ TypeScript
Generate生成代码片段	Constructor Override Getter/Setter Implement	不支持	不支持

基本功能

- 自动补全



- 智能提示

```java
@Resolve({"double->double"})
public class AggrAvg extends Aggregator {
    private static class AvgBuffer implements Writable {
        private double sum = 0;
        private long count = 0;
        @Override
        public void write(DataOutput out) throws IOException {
            out.writeDouble(sum);
            out.writeLong(count);
        }
        @Override
        public void readFields(DataInput in) throws IOException {
            sum = in.readDouble();
            count = in.readLong();
        }
    }
    private DoubleWritable ret = new DoubleWritable();
    @Override
    public Writable newBuffer() {
        return new AvgBuffer();
    }
    @Override
    public void iterate(Writable buffer, Writable[] args) throws UDFException {
        DoubleWritable arg = (DoubleWritable) args[0];
        AvgBuffer buf = (AvgBuffer) buffer;
        if (arg != null) {
            buf.count += 1;
            buf.sum += arg.get();
            if (buf.count > 9) {
                throw new IllegalStateException("只能计算10个数");
            }
        }
    }
```

- 语法诊断

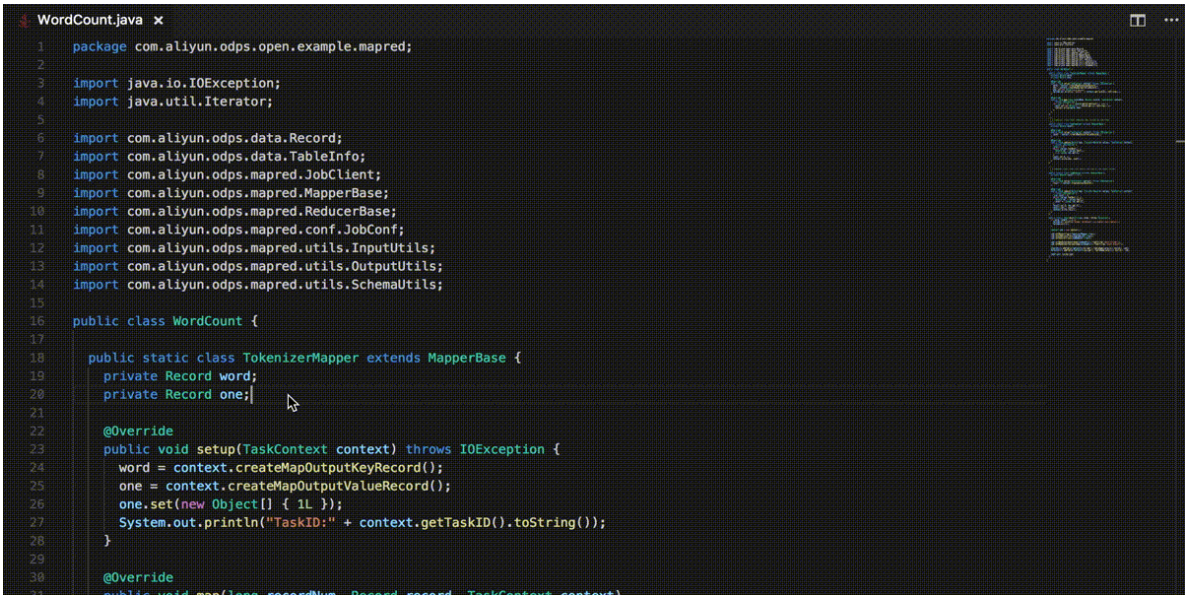
```
@Resolve({"double->double"})
public class AggrAvg extends Aggregator {
    private static class AvgBuffer implements Writable {
        private double sum = 0;
        private long count = 0;
        @Override
        public void write(DataOutput out) throws IOException {
            out.writeDouble(sum);
            out.writeLong(count)
        }
        @Override
        public void readFields(DataInput in) throws IOException {
            sum = in.readDouble();
            count = in.readLong();
        }
    }
    private DoubleWritable ret = new DoubleWritable();
    @Override
    public Writable newBuffer() {
        return new AvgBuffer();
    }
    @Override
```

- 函数参数提示

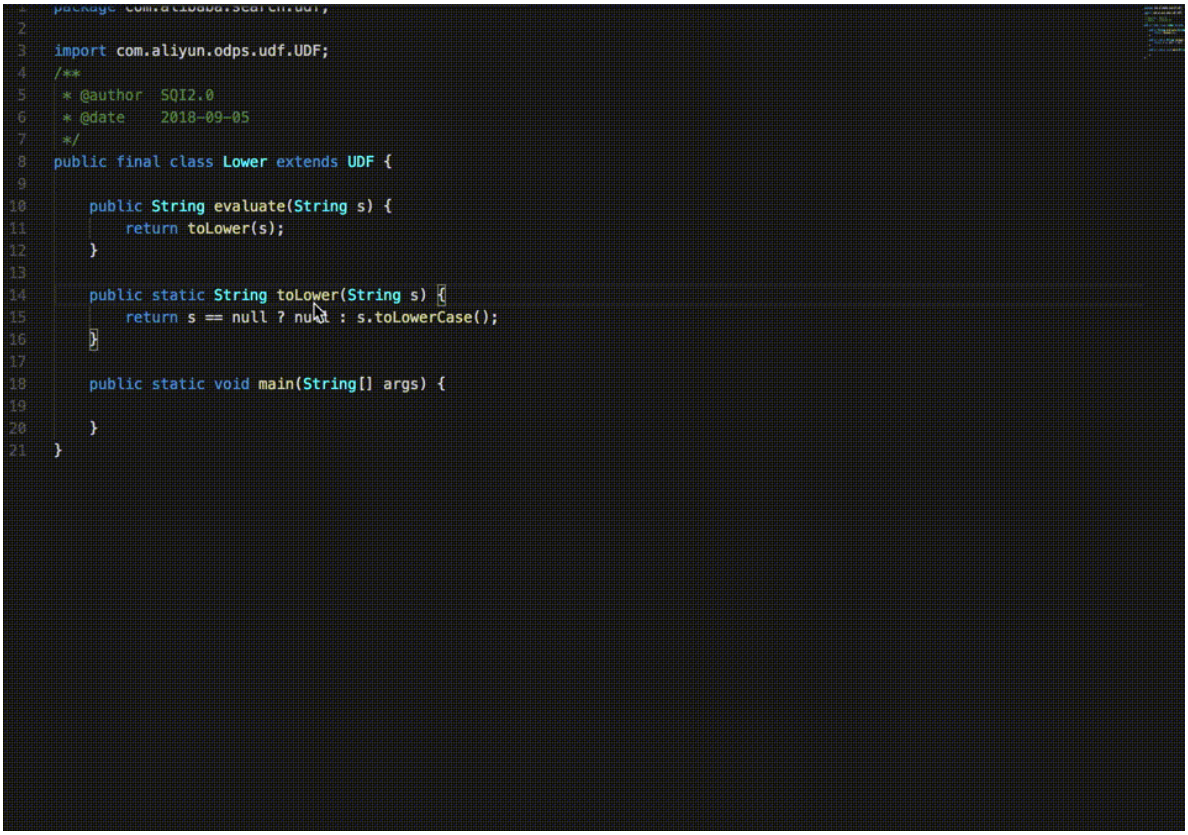
StudioUDAF.java | StudioUDTFTest.java | Lower.java | LowerTest.java

```
/**
 * Sample Java Class
 */
public class StudioUDTFTest {

    public static void main(String[] args){
        TestUtil.initWarehouse();
        StudioUDTFTest studioUdafTest = new StudioUDTFTest() ;
        try {
            studioUdafTest.simpleInput();
            studioUdafTest.inputFromTable();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Test
    public void simpleInput() throws Exception {
        BaseRunner runner = new UDTFRunner(null, "com.alibaba.dataworks.udtf.StudioUDTF");
        runner.feed(new Object[] {"one", "one"}).feed(new Object[] {"three", "three"})
            .feed(new Object[] {"four", "four"});
        List<Object[]> out = runner.yield();

        Assert.assertEquals(3, out.size());
        Assert.assertEquals("one,3", TestUtil.join(out.get(0)));
        Assert.assertEquals("three,5", TestUtil.join(out.get(1)));
        Assert.assertEquals("four,4", TestUtil.join(out.get(2)));
    }
```

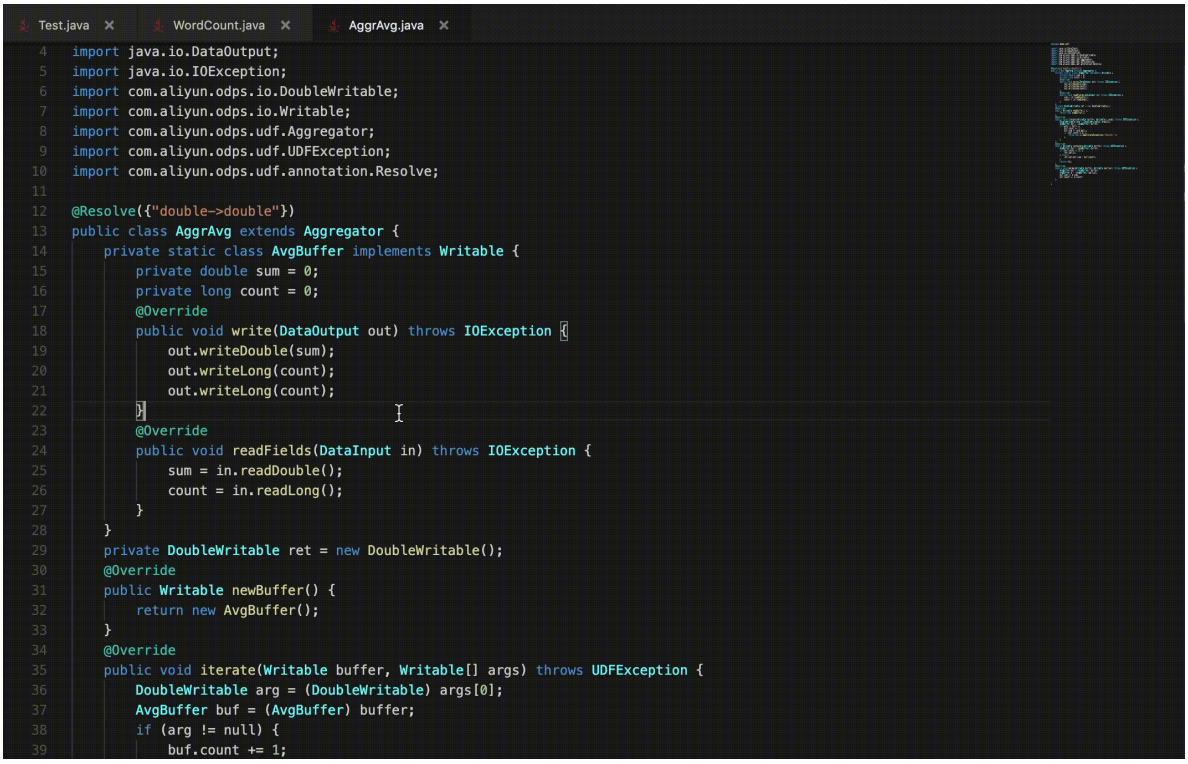
- 跳转定义


WordCount.java
package com.aliyun.odps.open.example.mapred;
import java.io.IOException;
import java.util.Iterator;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.MapperBase;
import com.aliyun.odps.mapred.ReducerBase;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
public class WordCount {
public static class TokenizerMapper extends MapperBase {
private Record word;
private Record one;
@Override
public void setup(TaskContext context) throws IOException {
word = context.createMapOutputKeyRecord();
one = context.createMapOutputValueRecord();
one.set(new Object[] { 1L });
System.out.println("TaskID:" + context.getTaskID().toString());
}
@Override

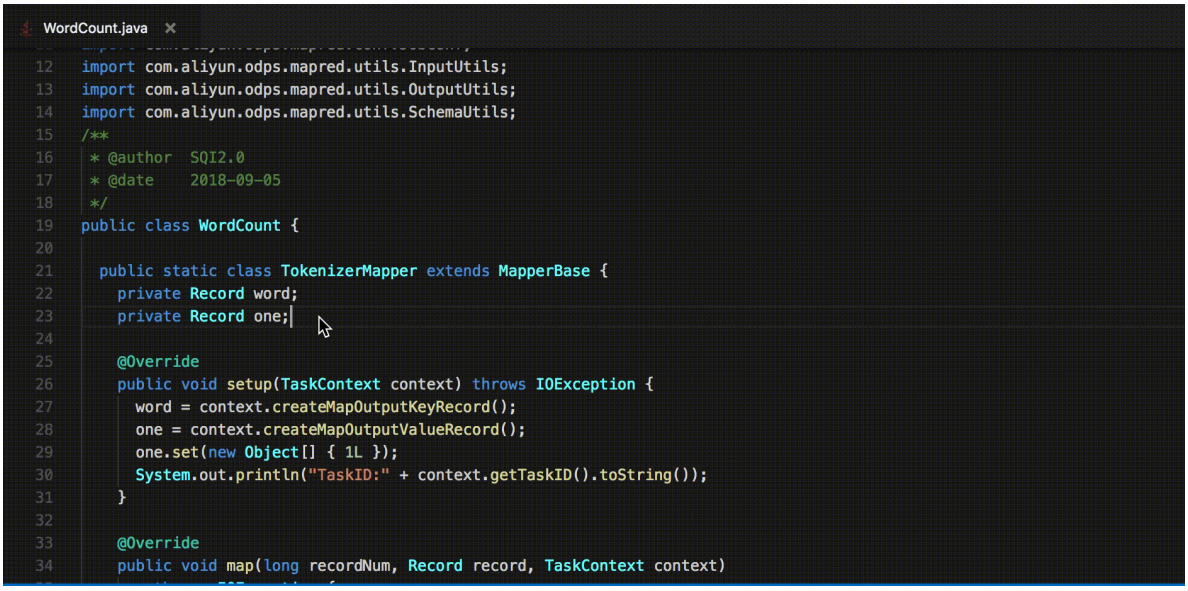
- 查找引用


import com.aliyun.odps.udf.UDF;
/**
* @author SQI2.0
* @date 2018-09-05
*/
public final class Lower extends UDF {
public String evaluate(String s) {
return toLower(s);
}
public static String toLower(String s) {
return s == null ? null : s.toLowerCase();
}
public static void main(String[] args) {
}
}

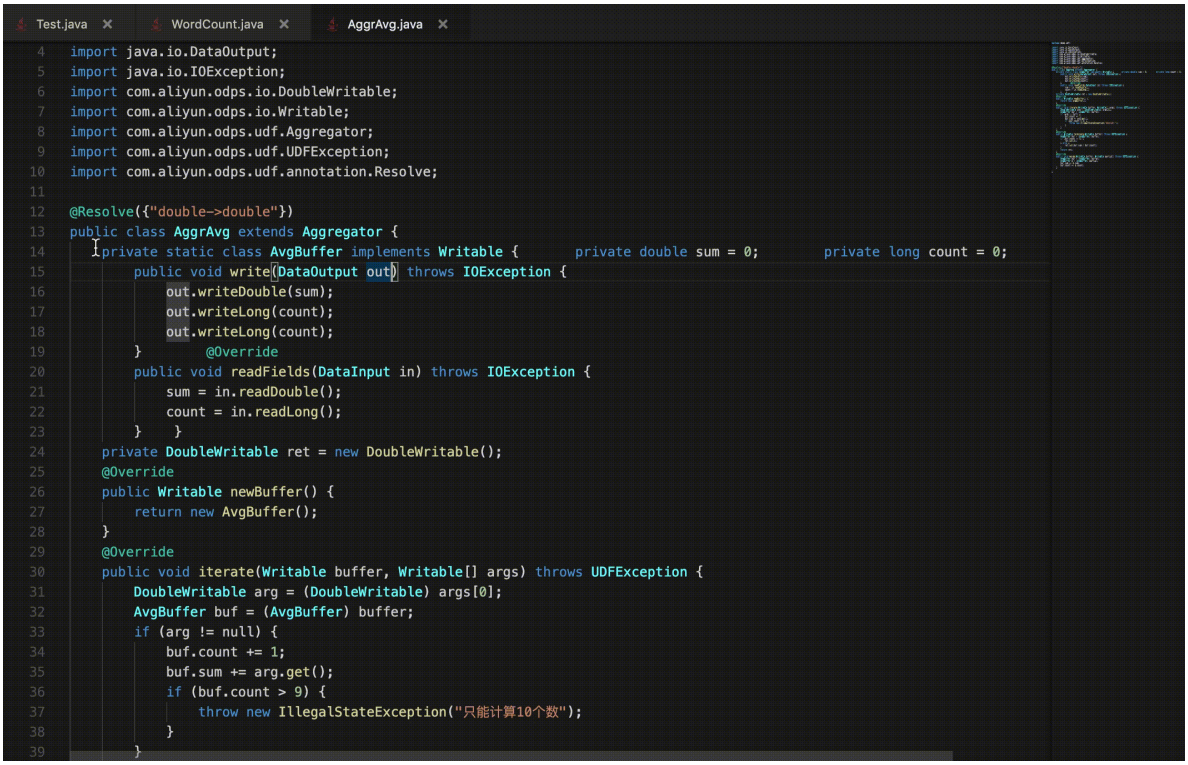
- 查找当前类成员



- 全局查找类/函数



· 代码格式化

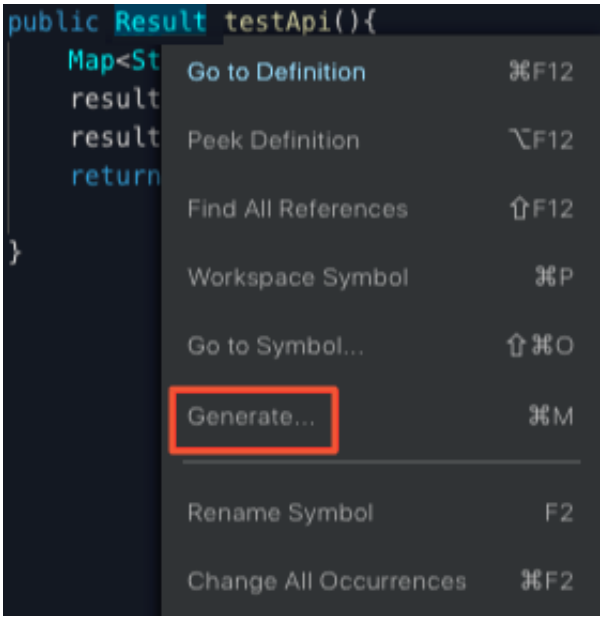


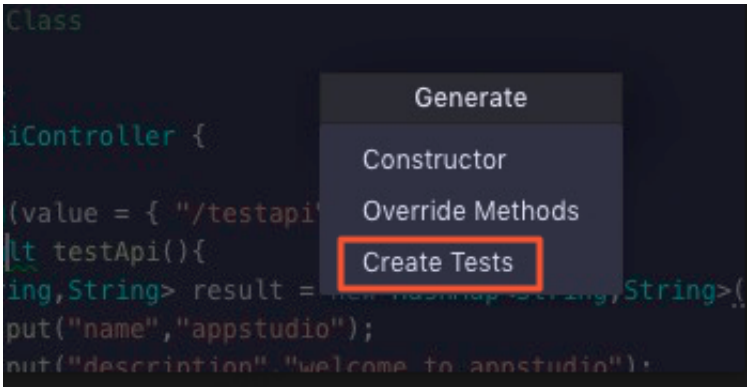
13.4.4.2 UT测试

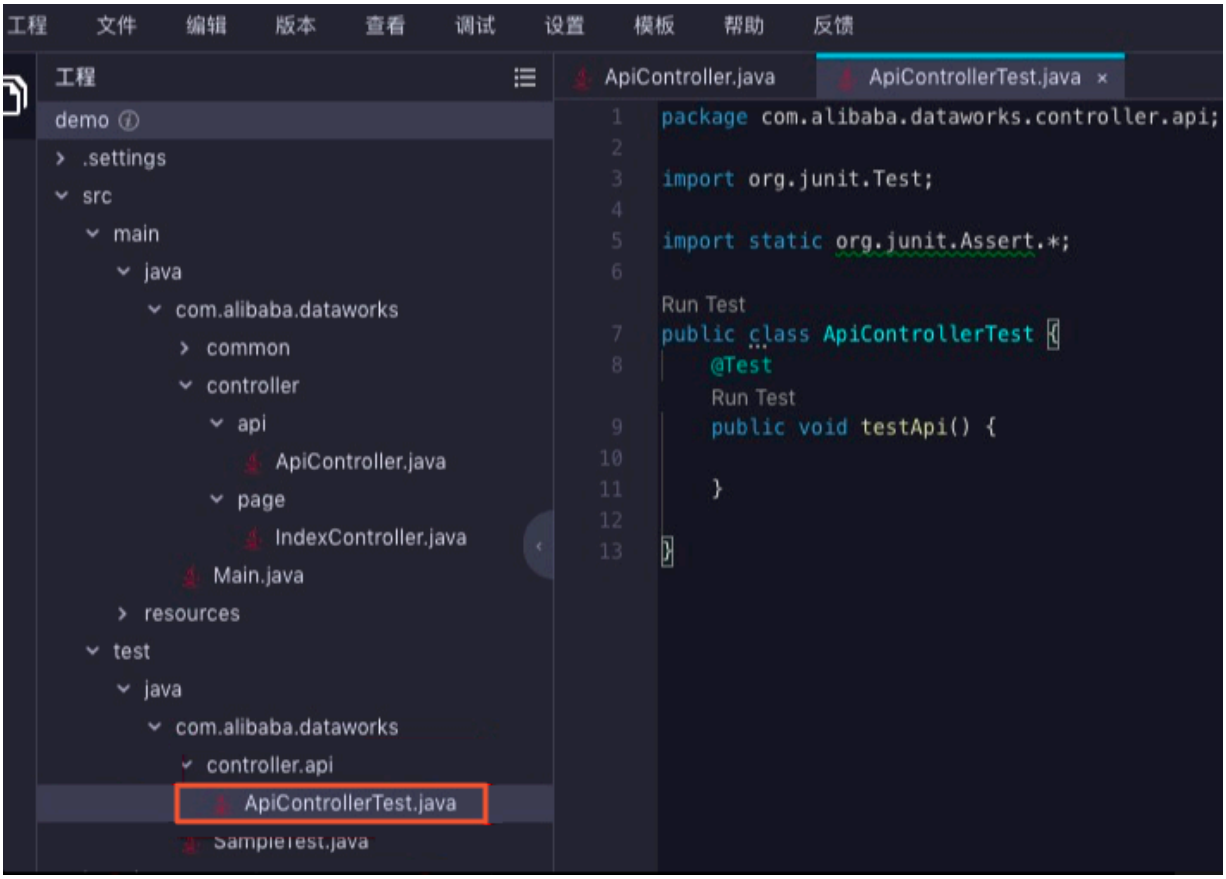
App Studio目前支持自动生成UT代码、检测UT测试入口、运行UT代码和展示运行结果等功能。

自动生成UT代码

打开相应文件后，右键单击代码编辑区，选择Generate > Create Test，即可在Test目录下自动生成测试类和测试代码。





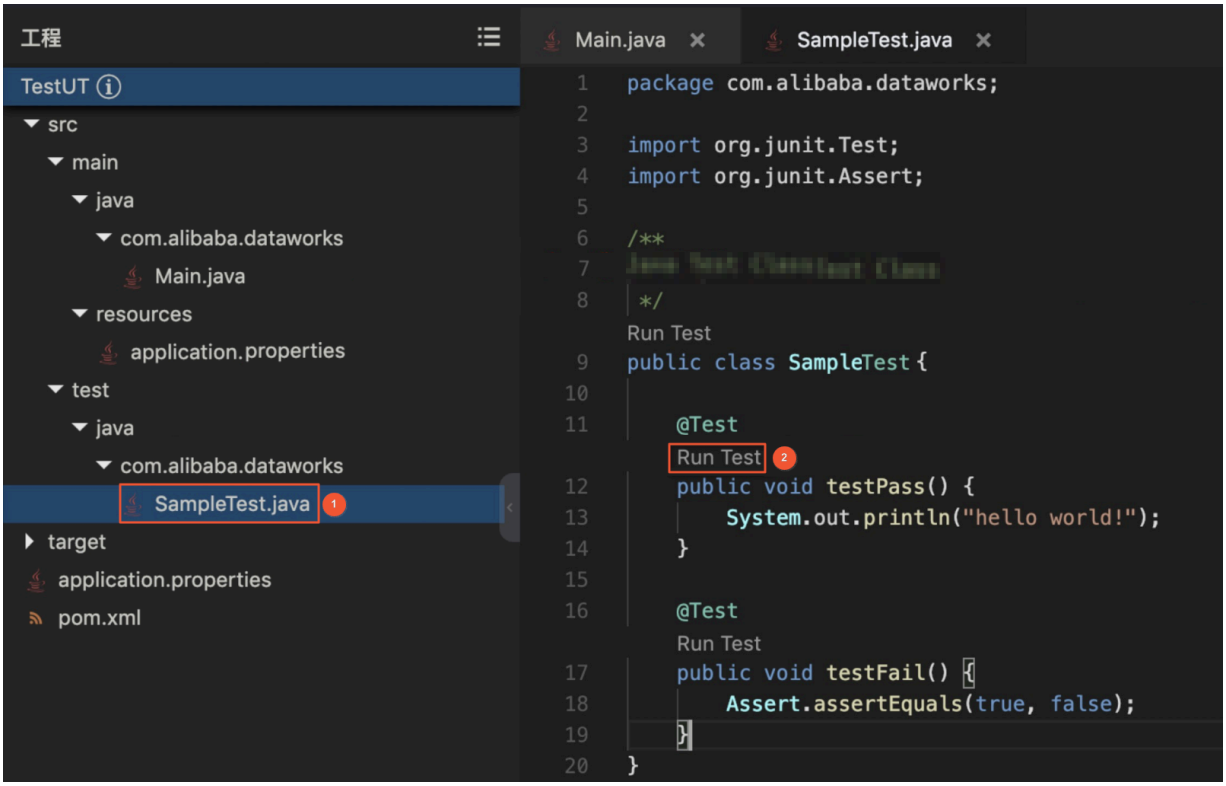


检测UT测试入口

说明:

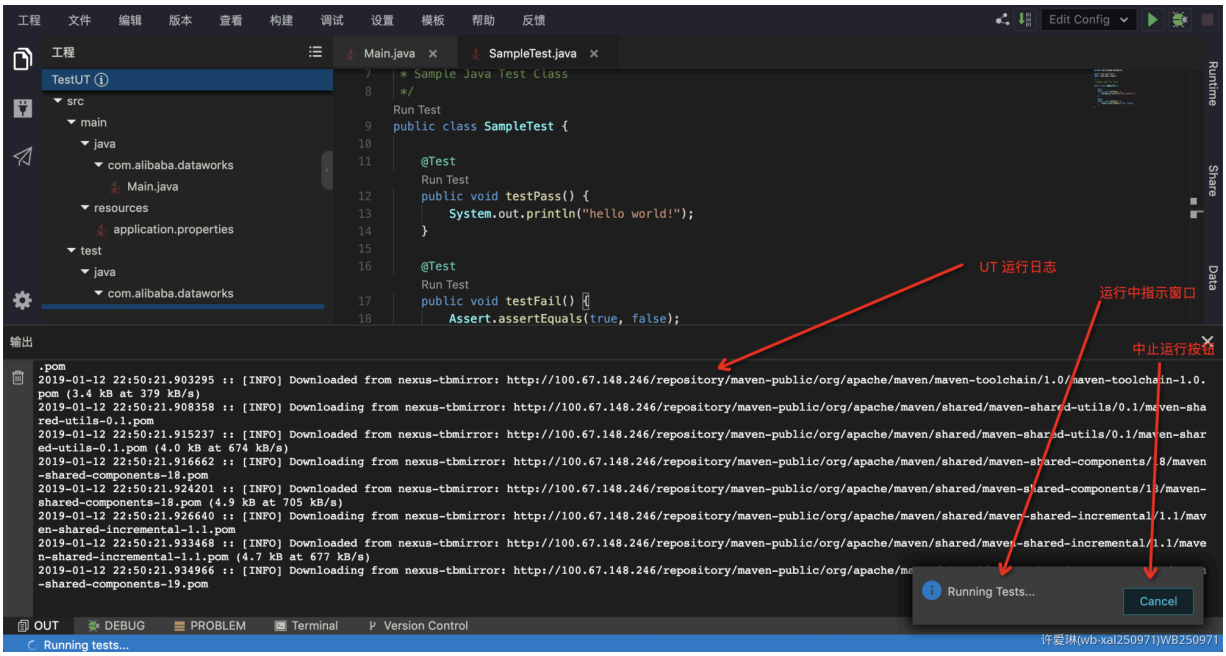
- **UT类需要写在*src/test/java*目录下。如果Java UT类文件不在该目录下，将无法被正常识别成Java UT类。**
- **@Test注解下的方法会出现Run Test的UT运行入口。**

完成Java类的创建后，在对应的测试用例方法上添加`org.junit.Test`的@Test注解即可。

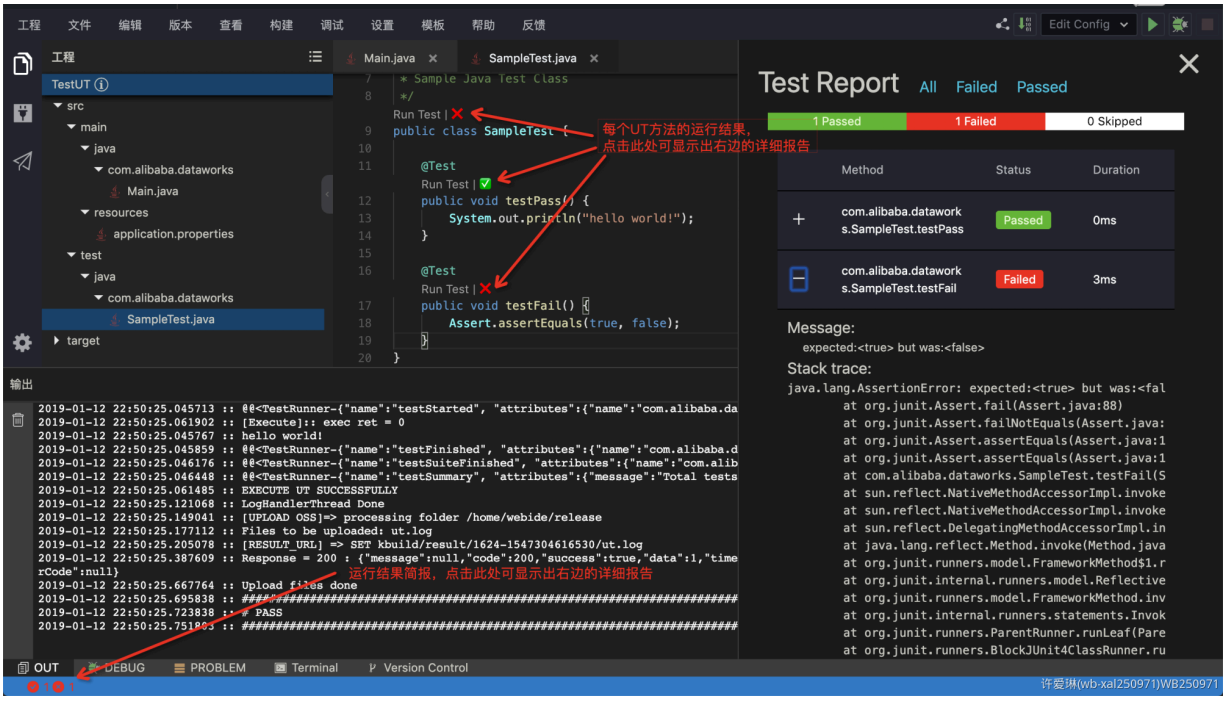


运行UT代码

单击右上角的运行按钮，即可运行UT的测试用例。



展示UT运行结果



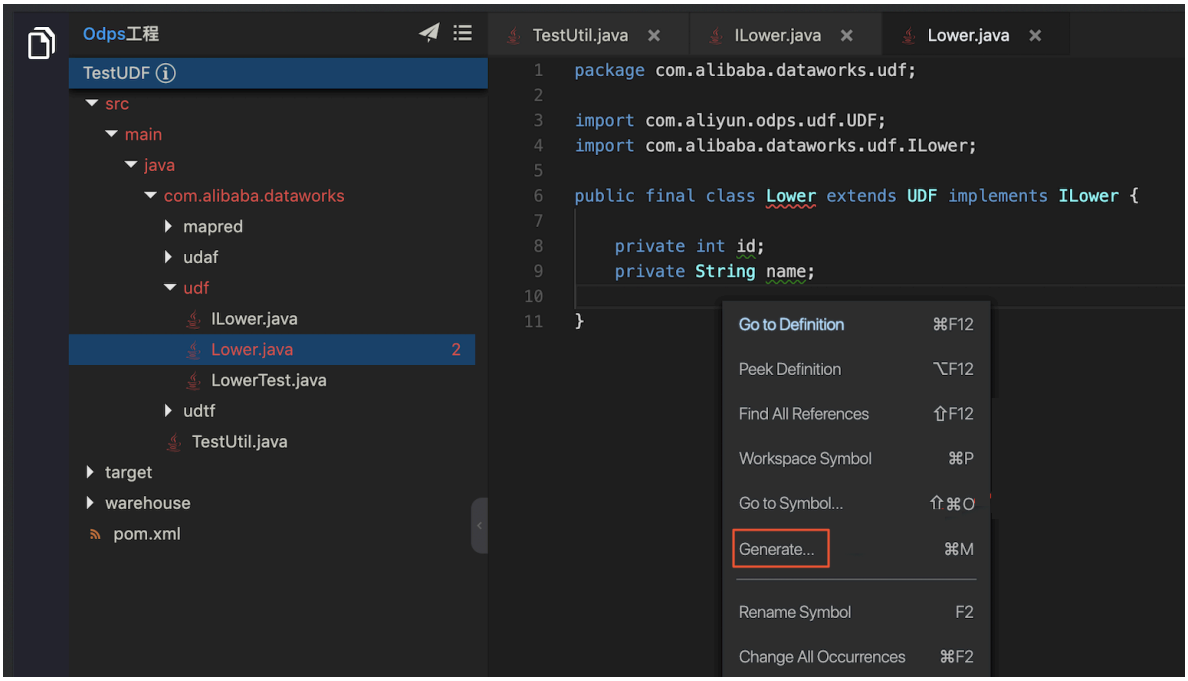
13.4.4.3 生成代码片段

目前App Studio在Java语言中，支持生成Java类的构造函数（Constructor）、Getter函数、Setter函数，也支持生成该类所继承父类的Override方法、所要实现的接口方法等。

功能入口

目前Java的代码生成入口有以下两种：

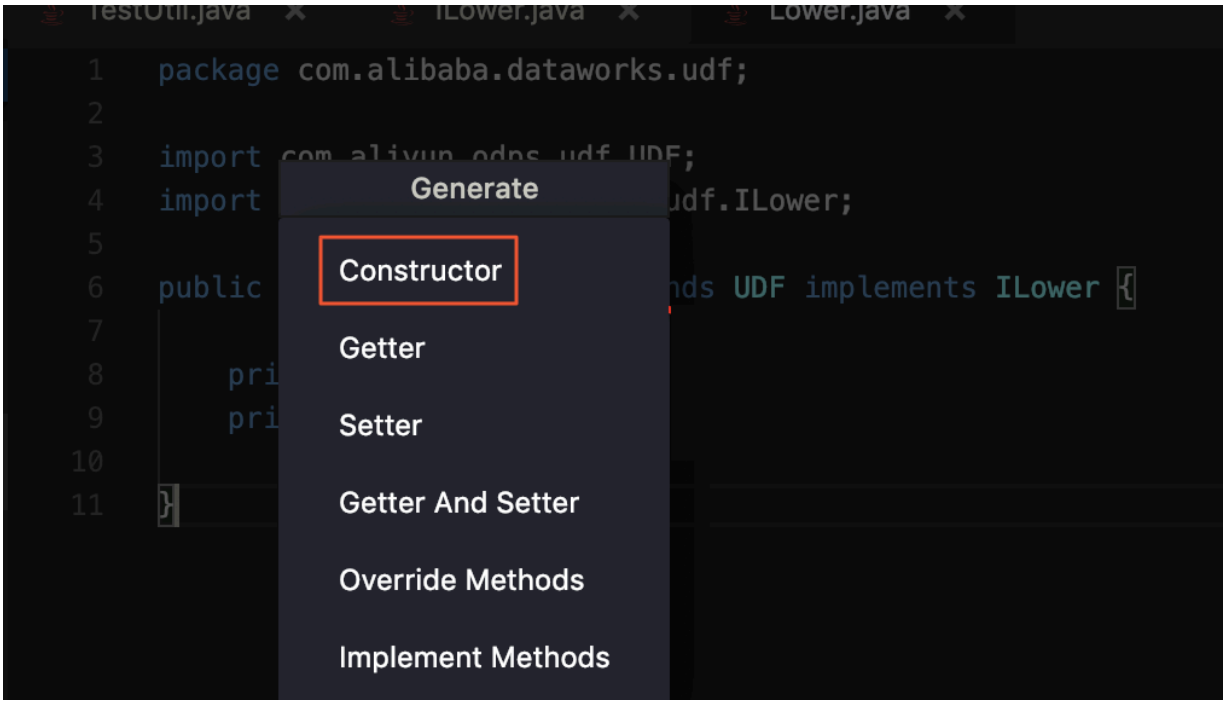
- **鼠标右键单击代码区域，选择Generate。**



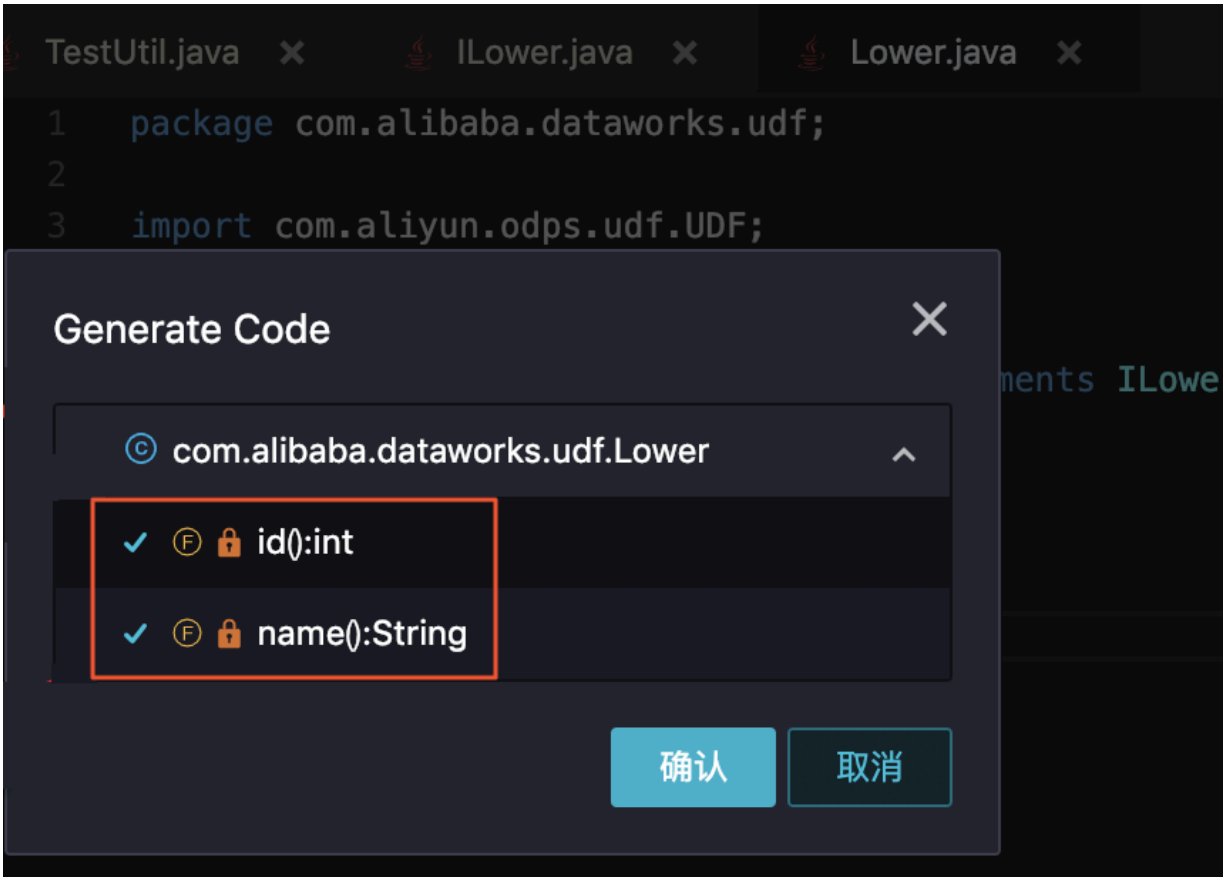
- 通过快捷键cmd+m自动生成Java代码。

Constructor

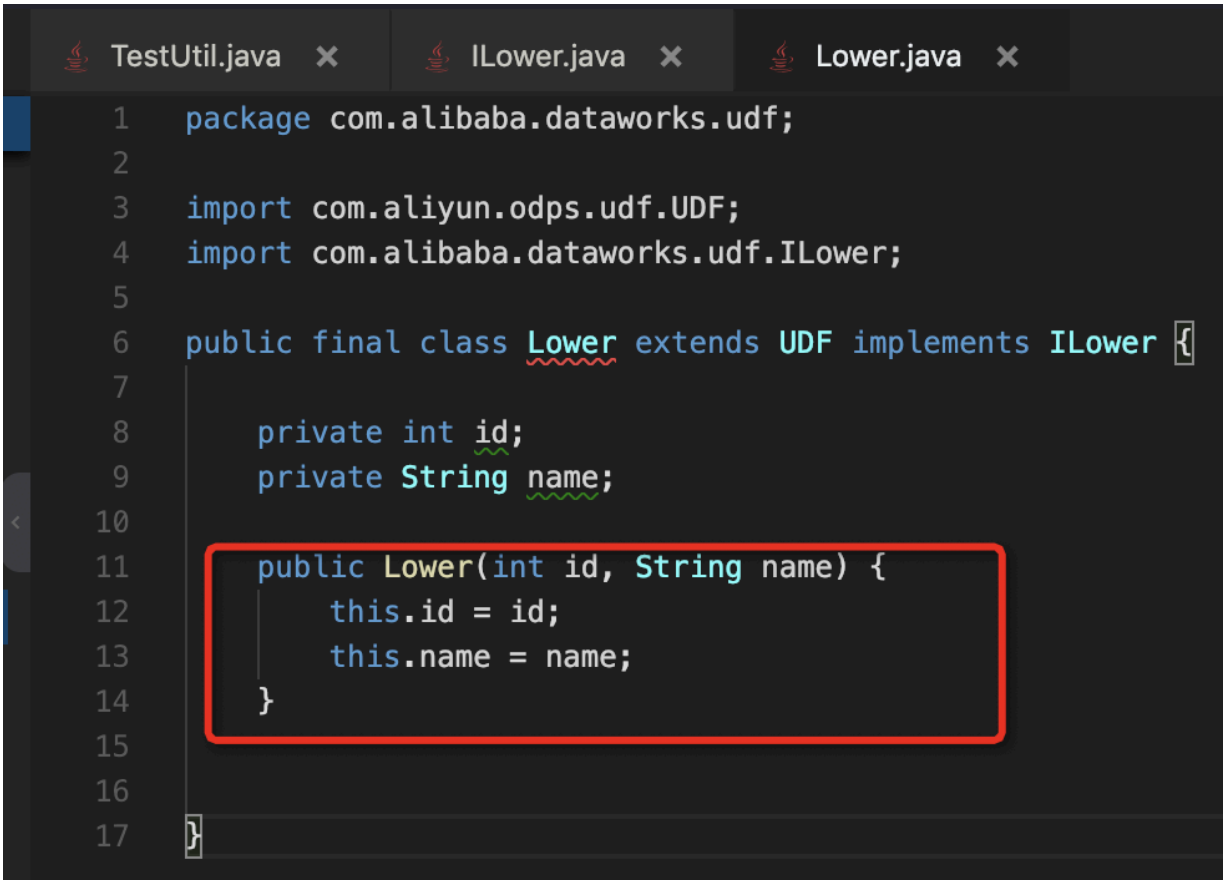
进入Generate Code面板后，选择Constructor，即可生成Constructor。



选择构造函数中要包含的Fields。

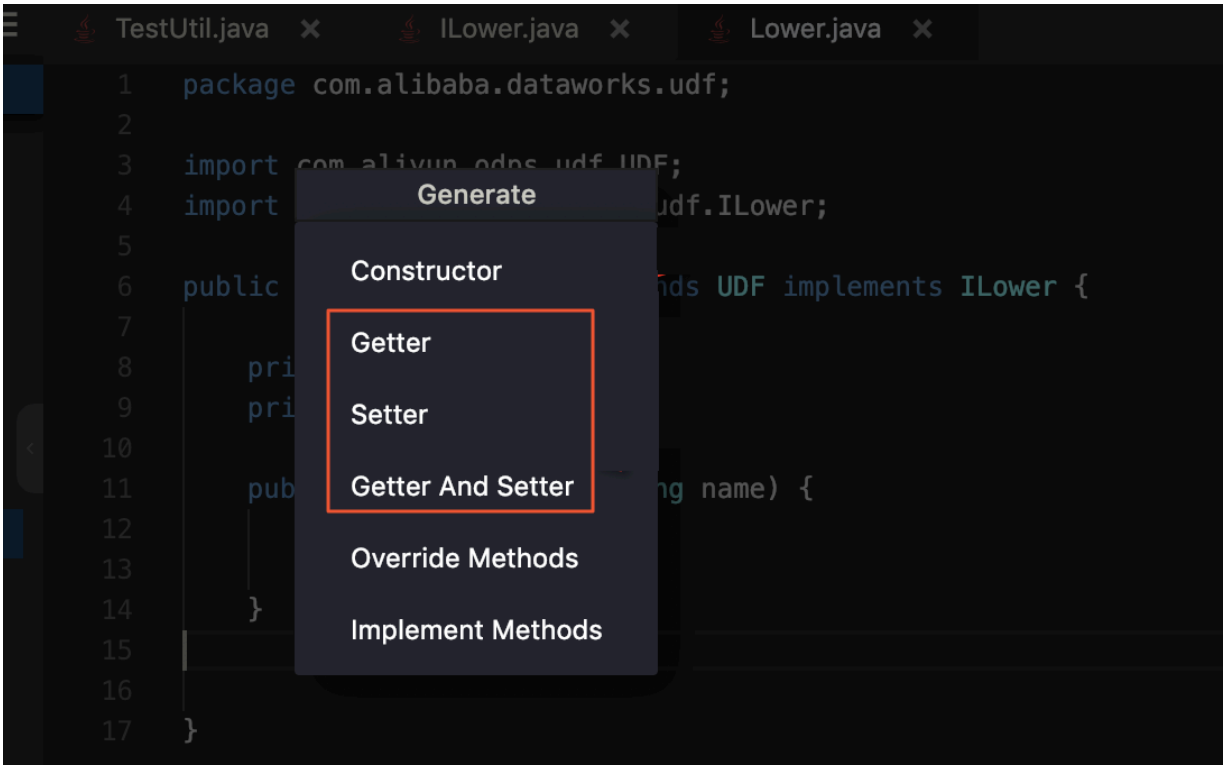


即可生成包含这些Fields初始化语句的构造函数。



Getter&Setter

Getter和Setter函数的生成方式类似于Constructor的生成方式。

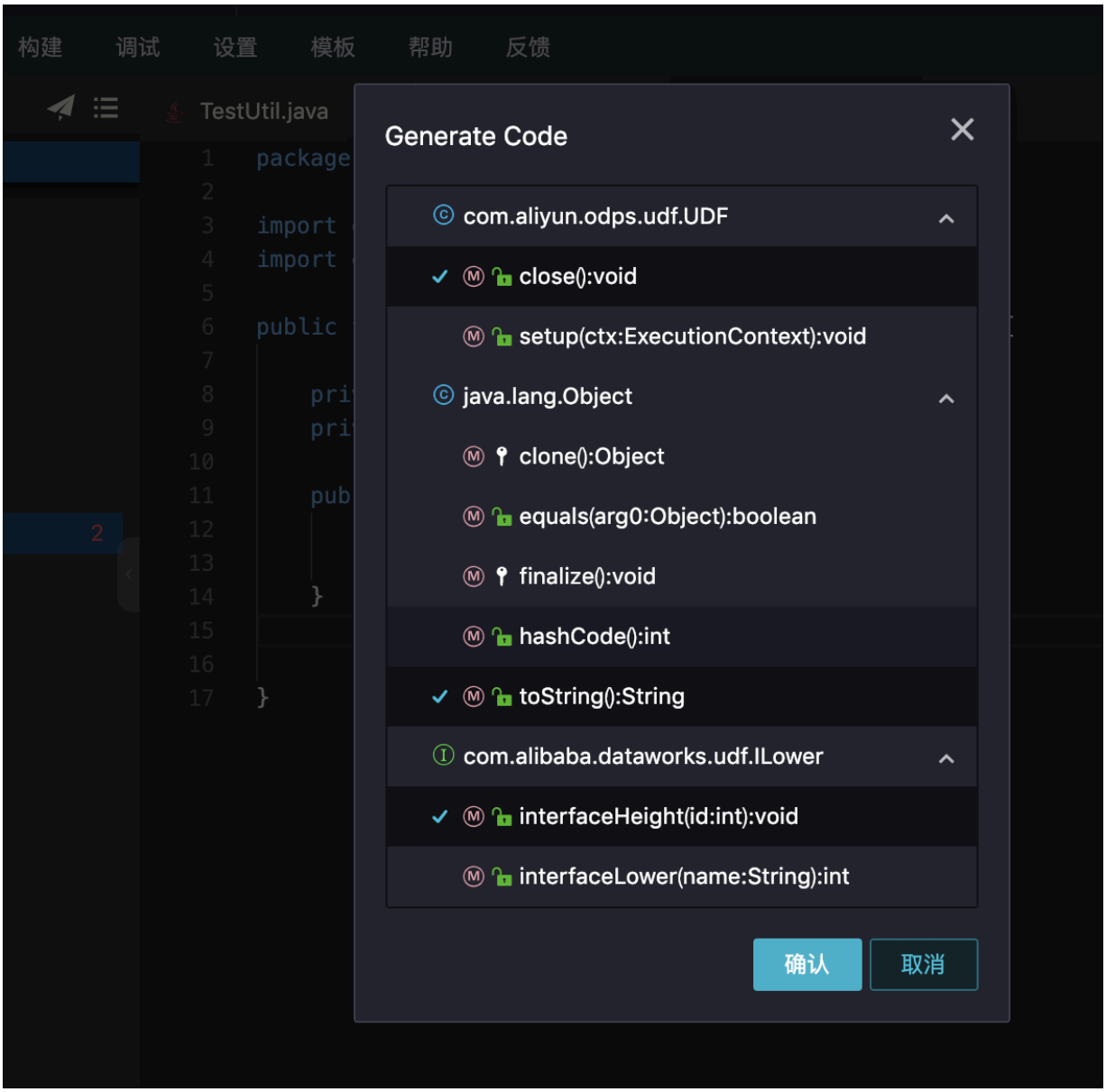


说明:

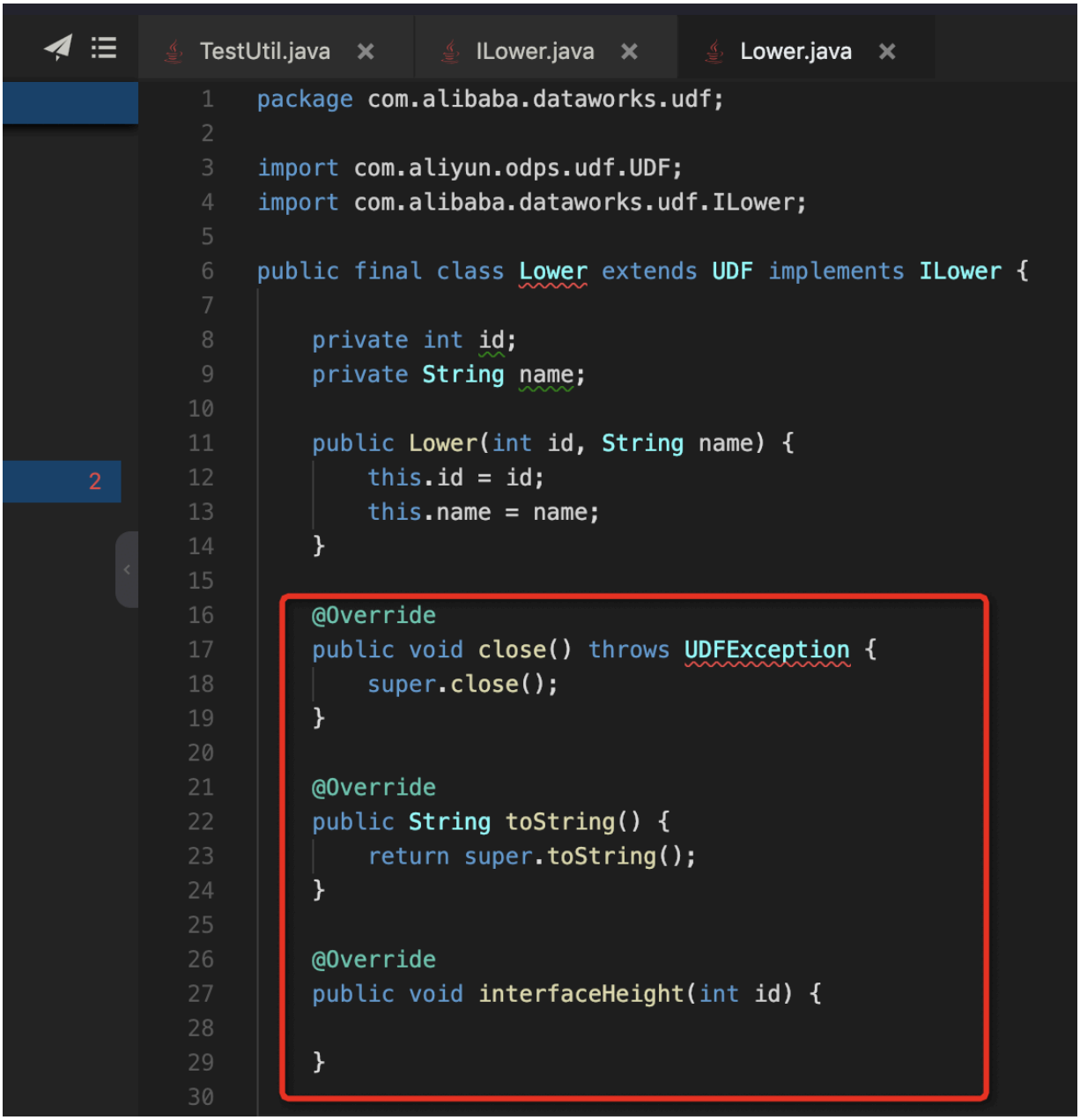
如果该Java类没有任何Fields，或者该Java类已经被lombok的@data注解覆盖，则没有图中的三个选项，因为此时该类不需要生成Getter或Setter函数。

Override Methods

当选择了生成Override Methods的一级菜单后，在二级菜单中会罗列所有可Override的方法。

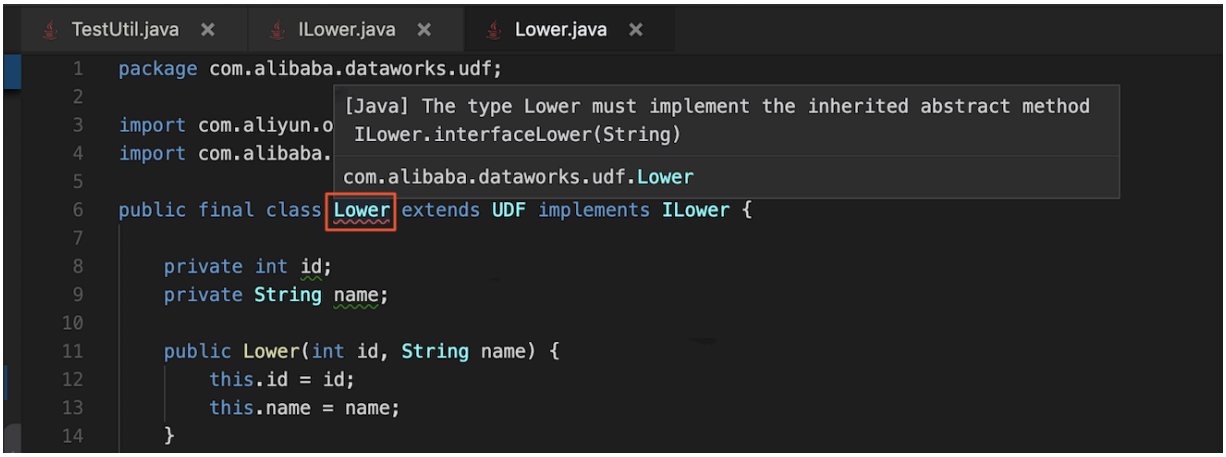


选择之后即可生成对应方法。

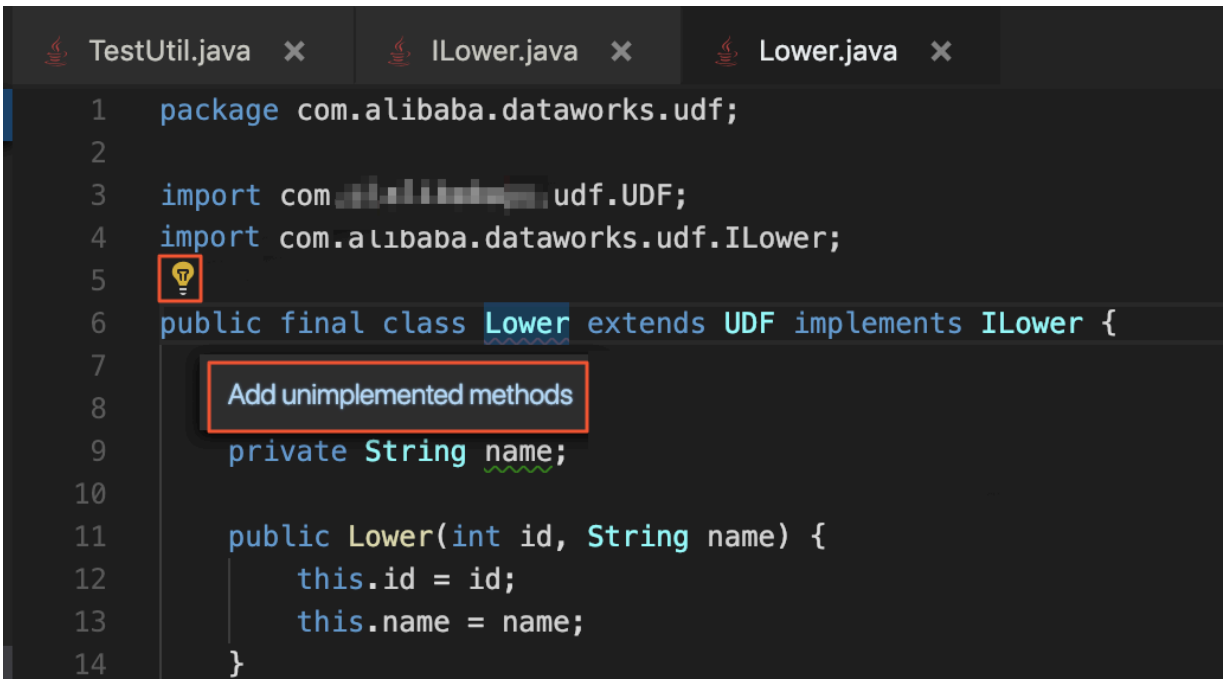


Implement Methods

Implement Methods与Override Methods类似。如果实现接口的方法不实现，会产生Java语法问题，从而出现红色波浪线。



除本文介绍的Generate功能外，您还可以使用智能提示功能达到同样的效果。



生成的代码如下所示。

```
6   public final class Lower extends UDF implements ILower {
7
8       private int id;
9       private String name;
10
11      public Lower(int id, String name) {
12          this.id = id;
13          this.name = name;
14      }
15
16      @Override
17      public int interfaceLower(String name) {
18          return 0;
19      }
20
21      @Override
22      public void interfaceHeight(int id) {
23
24      }
25
```

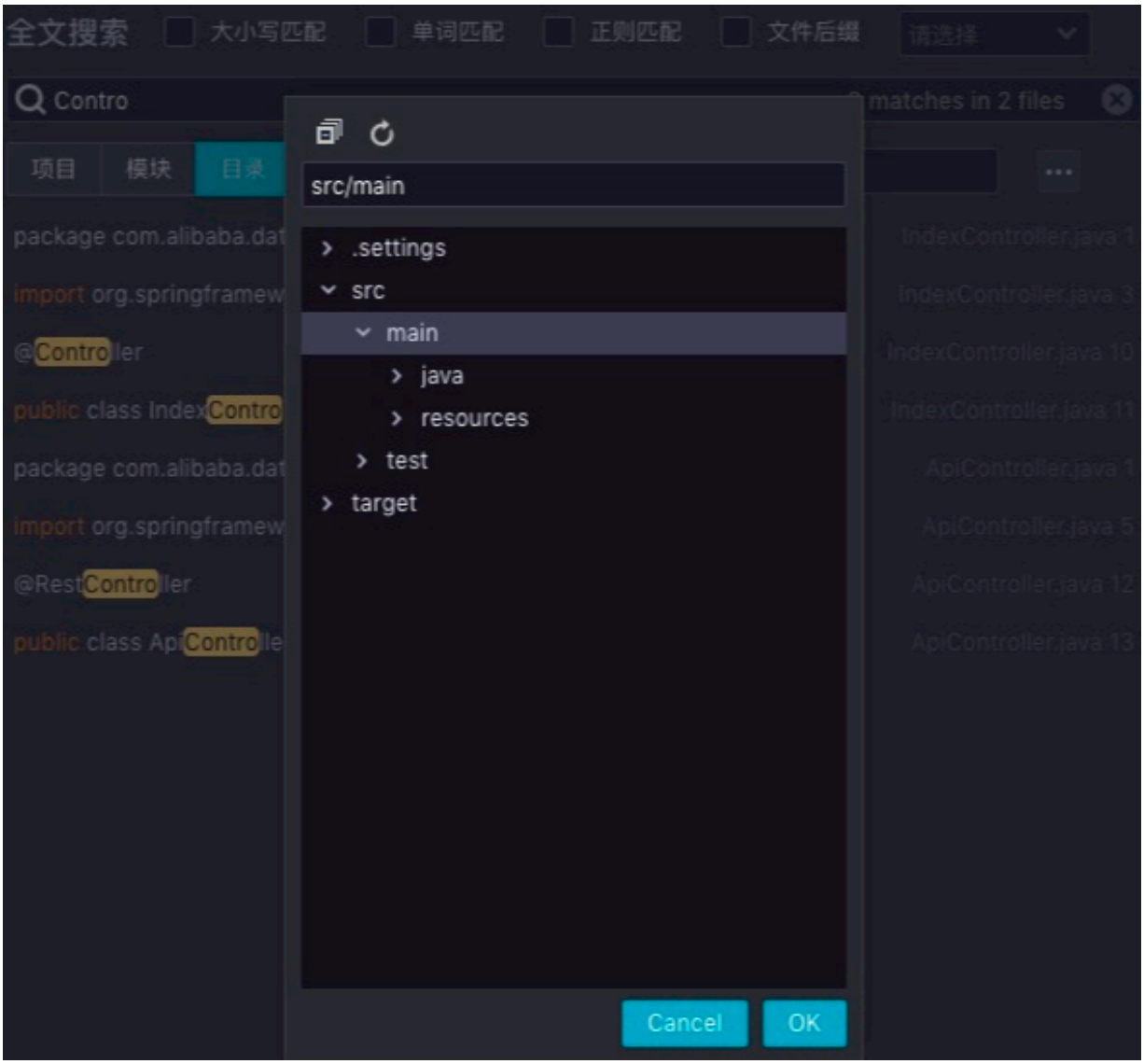
13.4.4.4 全文内容搜索

App Studio支持全文内容搜索功能。

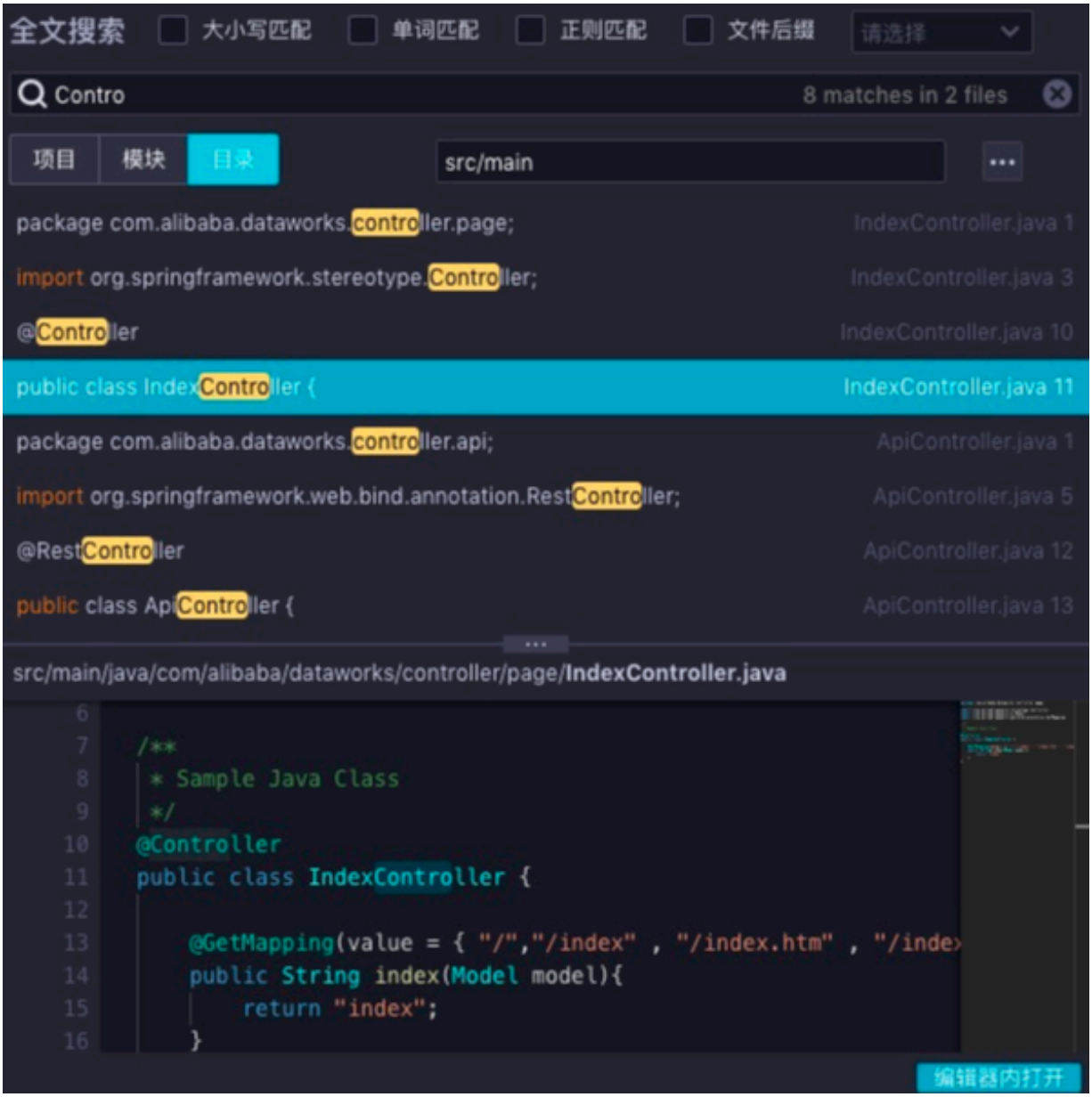
选择菜单栏中的编辑 > 全文搜索。

支持输入小写进行精确匹配、单词精确匹配、正则匹配，支持查找指定的文件类型。

支持根据模块、目录进行查找。



选中文件后，可以定位到文件中的搜索内容，并在编辑器内打开该文件。

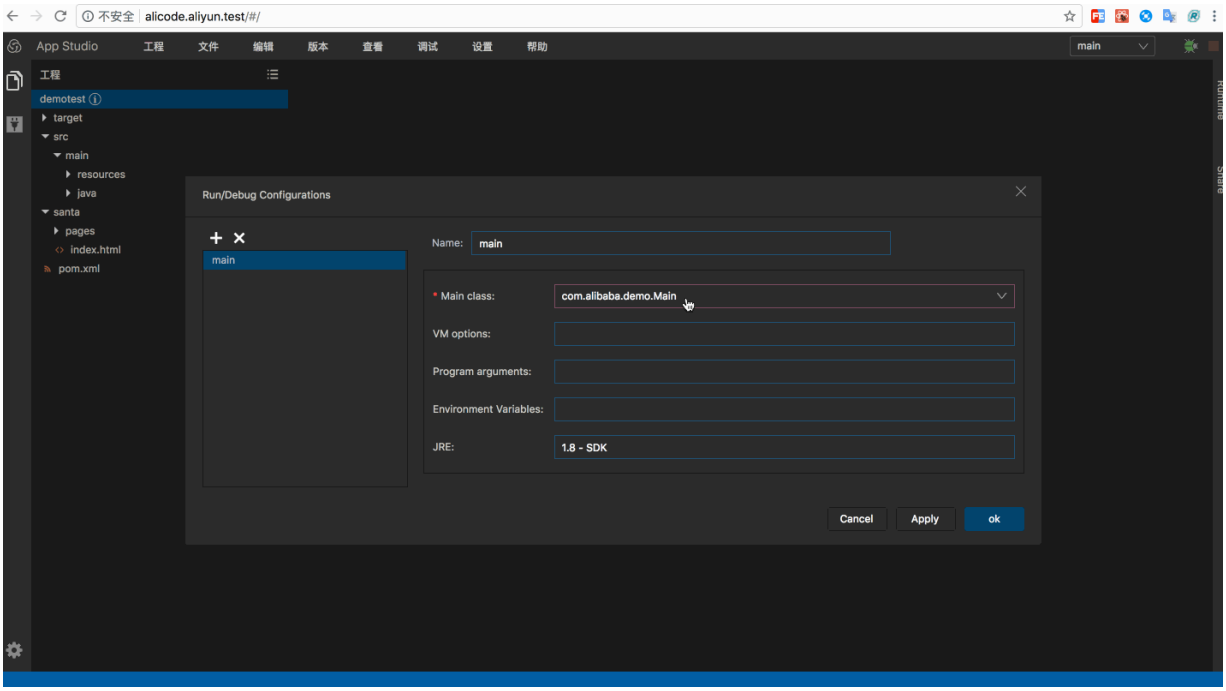

全文搜索
大小写匹配
单词匹配
正则匹配
文件后缀
请选择
Contro
8 matches in 2 files
项目
模块
目录
src/main
package com.alibaba.dataworks.controller.page;
IndexController.java 1
import org.springframework.stereotype.Controller;
IndexController.java 3
@Controller
IndexController.java 10
public class IndexController {
IndexController.java 11
package com.alibaba.dataworks.controller.api;
ApiController.java 1
import org.springframework.web.bind.annotation.RestController;
ApiController.java 5
@RestController
ApiController.java 12
public class ApiController {
ApiController.java 13
src/main/java/com/alibaba/dataworks/controller/page/IndexController.java
/**
* Sample Java Class
*/
@Controller
public class IndexController {
@GetMapping(value = { "/","/index" , "/index.htm" , "/inde
public String index(Model model){
return "index";
}
编辑器内打开

13.4.5 调试

13.4.5.1 Config配置及启动

您可通过配置入口函数，单击调试、断点等步骤，进行程序的调试。

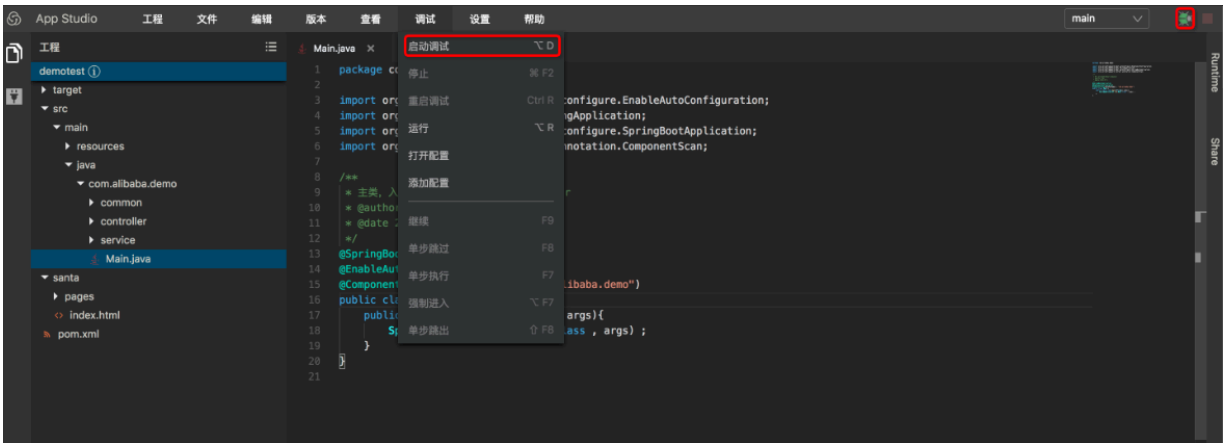
配置入口函数



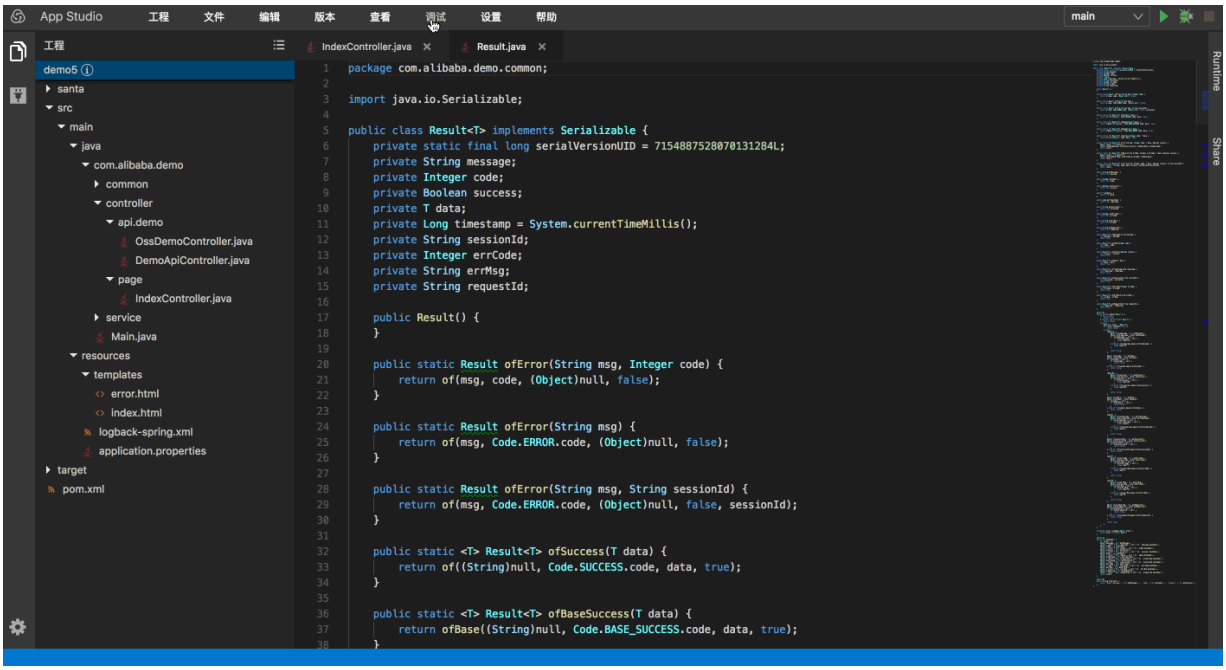
配置	说明
MainClass	您可以从多个配置中选择需要启动的main函数。
VM options	您可以配置在JVM启动时，例如-D -Xms -Xmx等配置。
Program arguments	您可以添加启动参数，此参数会被main函数的args参数接收。
Environment	环境变量参数。
PORT	端口，表示本程序需要暴露的端口信息，例如springboot经典的7001、8080等端口。
机器	您可以选择需要的机器配置进行调试。 2vCPU，4G内存 2vCPU，2G内存 ✓ 2vCPU，4G内存 4vCPU，8G内存
HotCode	此配置仅在run模式下生效，默认使用公司的HotCode2插件进行启动。

启动调试

选择菜单栏中的调试 > 启动调试。



因为需要为您准备运行环境和下载mvn依赖，第一次启动时速度较慢。重启调试会跳过此步骤，启动速度逐渐接近本地编辑器的体验。



13.4.5.2 在线调试

在线调试支持Java Application和基于SpringBoot的Web工程。

进行在线调试前，首先要#unique_654/unique_654_Connect_42_section_j3h_y3p_fhb和#unique_654/unique_654_Connect_42_section_qx3_kjp_fhb，完成上述步骤后，进行后续操作。

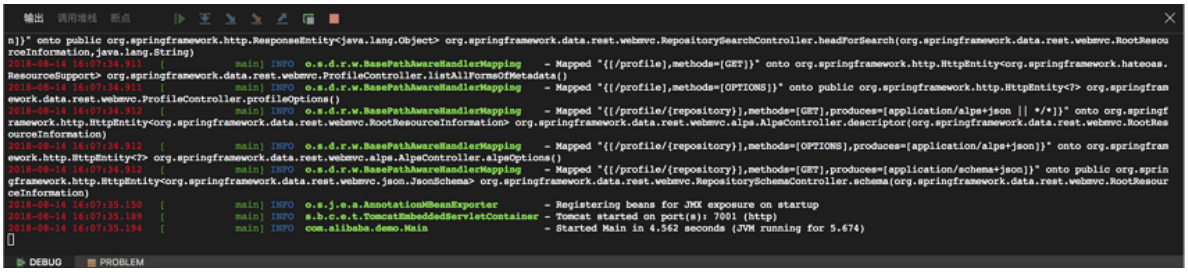
透出服务

程序成功启动后，会提供两个基本服务，您可以单击后端链接，对后端Java代码进行调试。

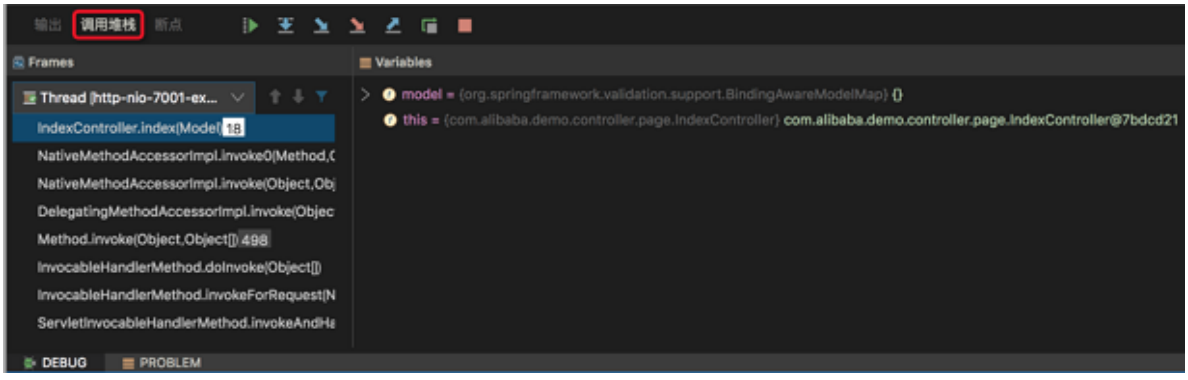
面板介绍

- 输出面板

  输出面板会显示所有程序的标准输出（暂不支持System.in），支持ansi颜色，体验与本地终端基本一致。

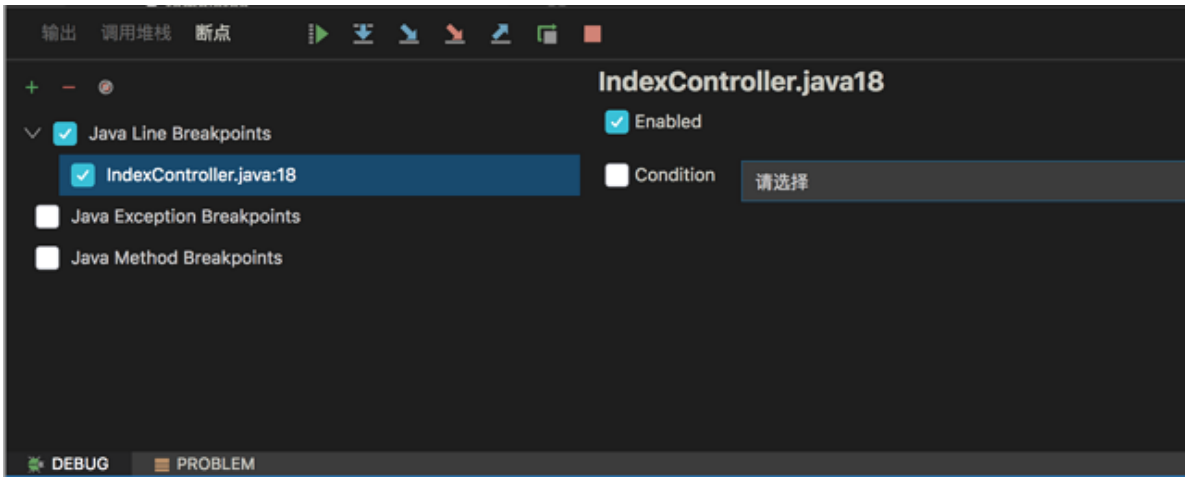


- 调用堆栈



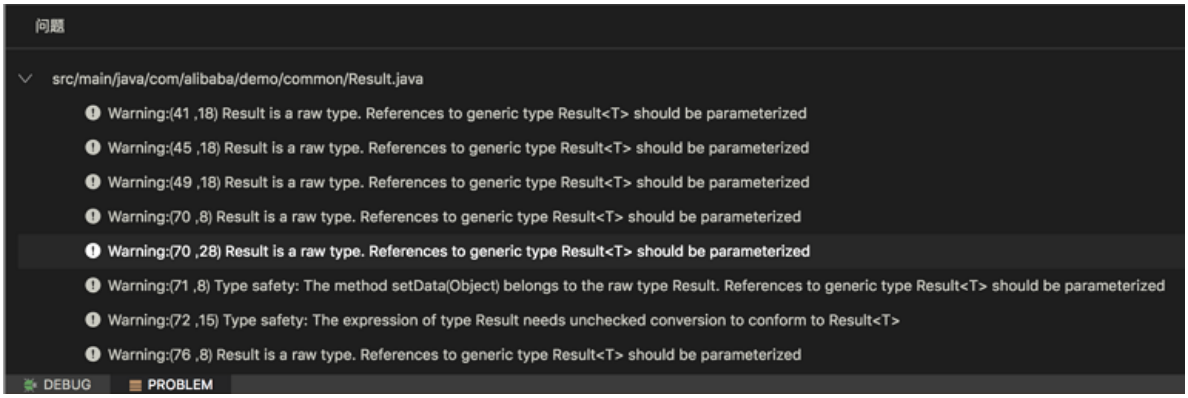
- 断点

  断点面板为您展示当前设置的所有断点，后续将为您介绍断点类型及使用。



- PROBLEM

  如果程序遇到编译问题，会展示在PROBLEM面板上，您可通过单击跳转至对应的文件行。

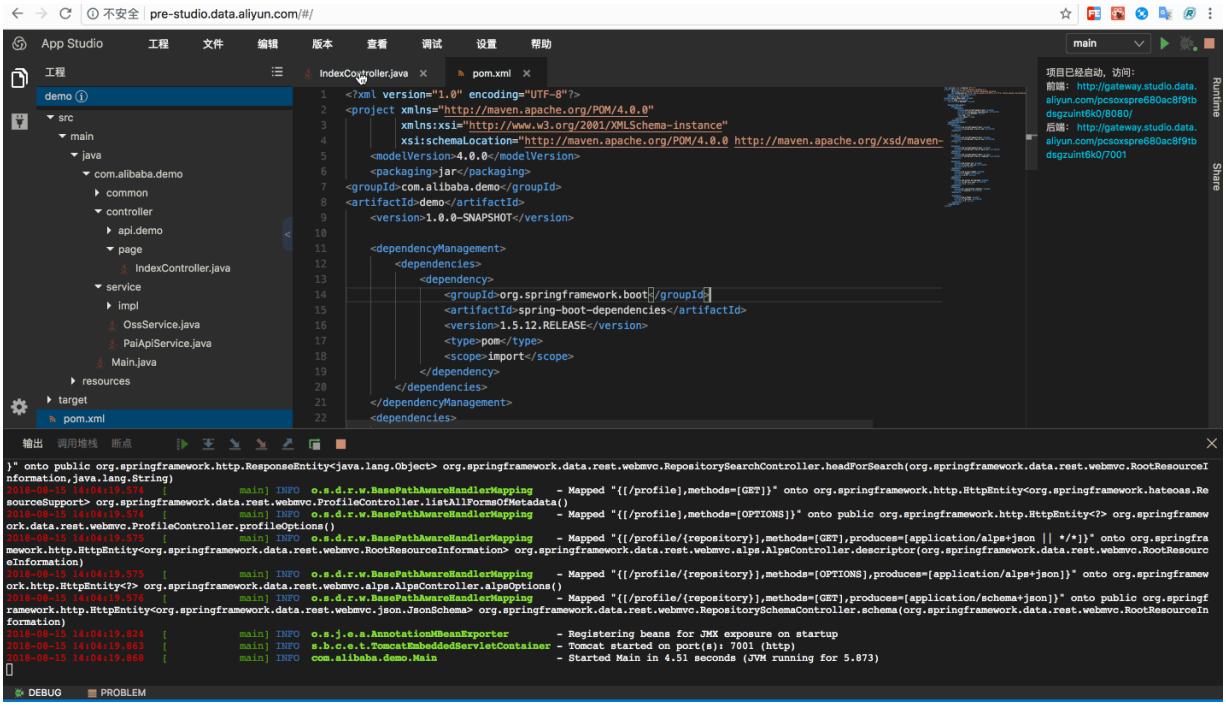


断点介绍

App Studio支持普通行断点、函数断点和异常断点，详情请参见#unique_655。

调试按钮

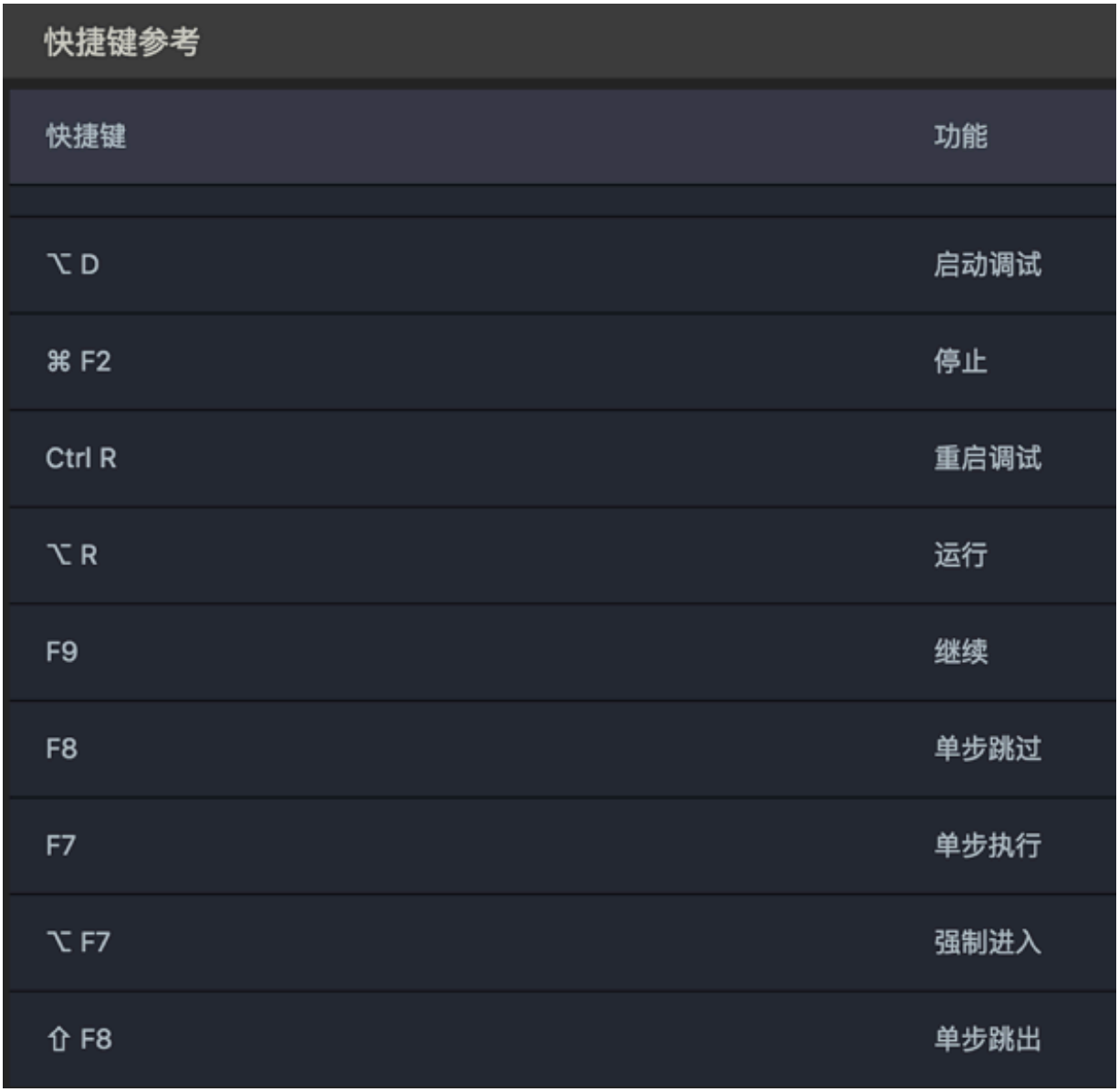
调试界面如下所示：



上图中从左到右的每个按钮所代表的含义如下：

功能	说明
continue	恢复当前断点时，当前线程继续运行。
step over	执行到下一行。
step in	进入函数。
force step in	强制进入函数，与step in的区别在于，它可以引导断点执行到java自带的类库中。
step out	从当前函数跳出。
restart	目前的restart的实现方式较为简单（可能无法完成程序清理等工作），正在优化中。
stop	停止。

操作的快捷键如下：

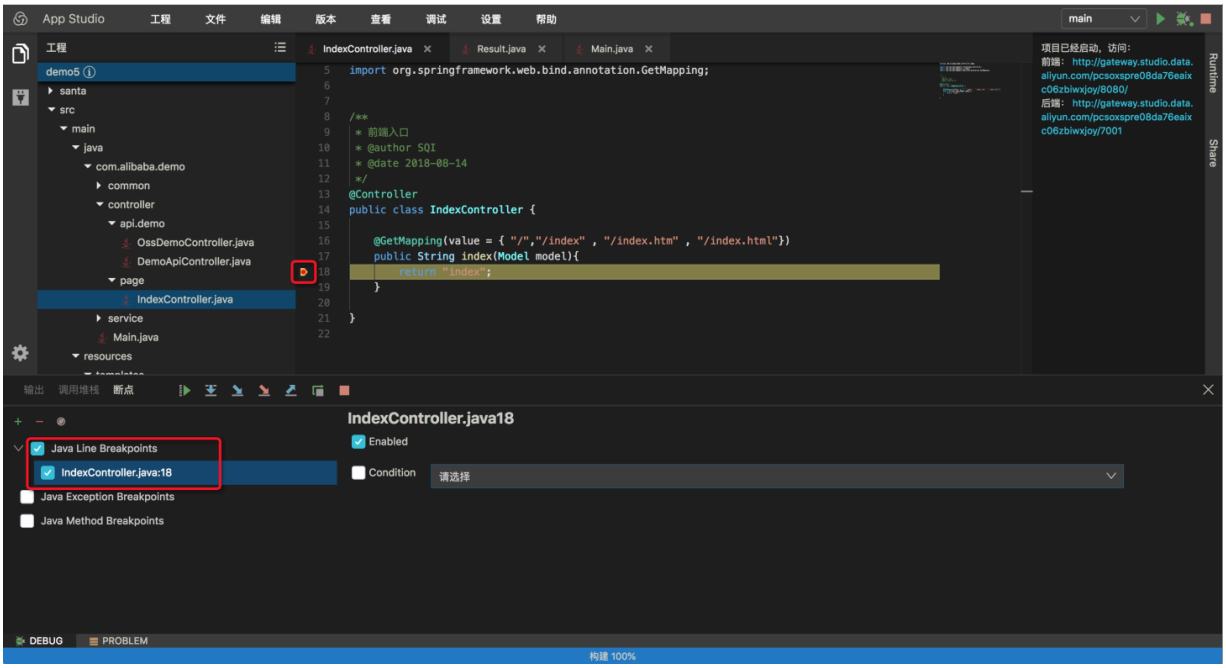


13.4.5.3 断点类型

App Studio支持普通行断点、函数断点和异常断点三种断点类型。

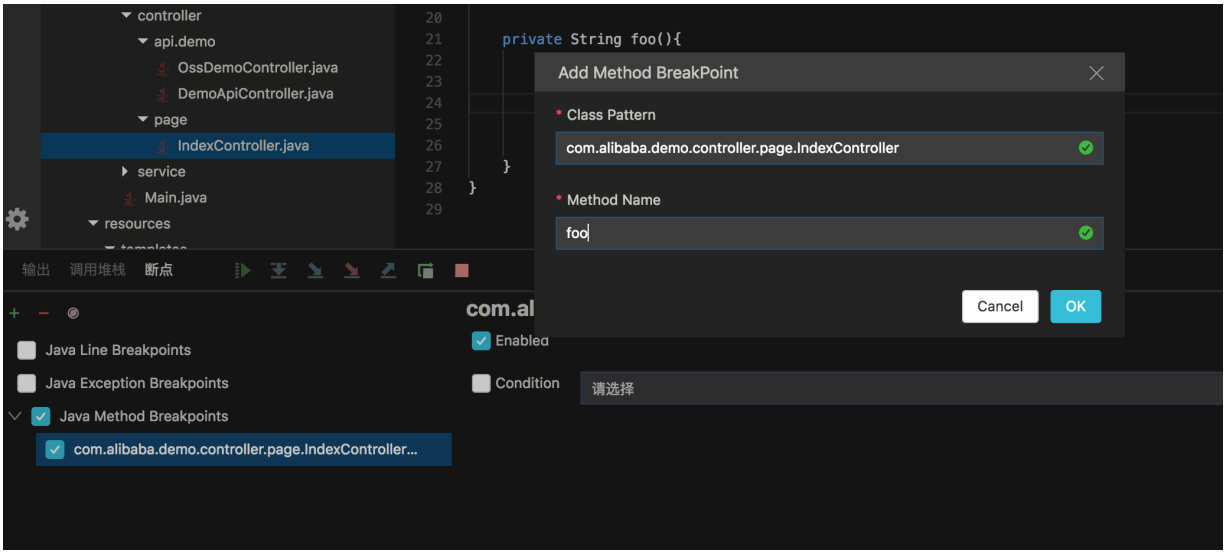
普通行断点

通过单击文件行号前的空白区域，可以生成针对该行的断点，同时断点面板中会显示该断点。

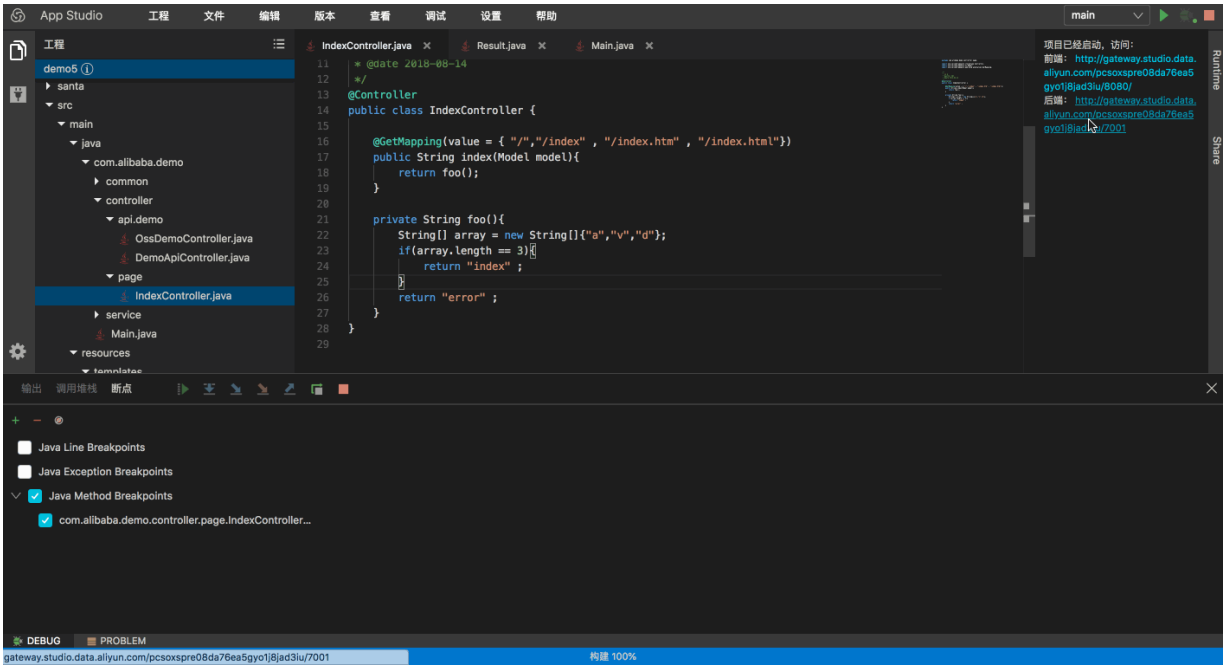


函数断点

函数断点相比异常断点与行断点的不同点为：函数断点会触发两次事件，即entry/exit。您可以手动添加一个函数断点，也可在函数被定义的地方打断点，同样会产生一个函数断点。

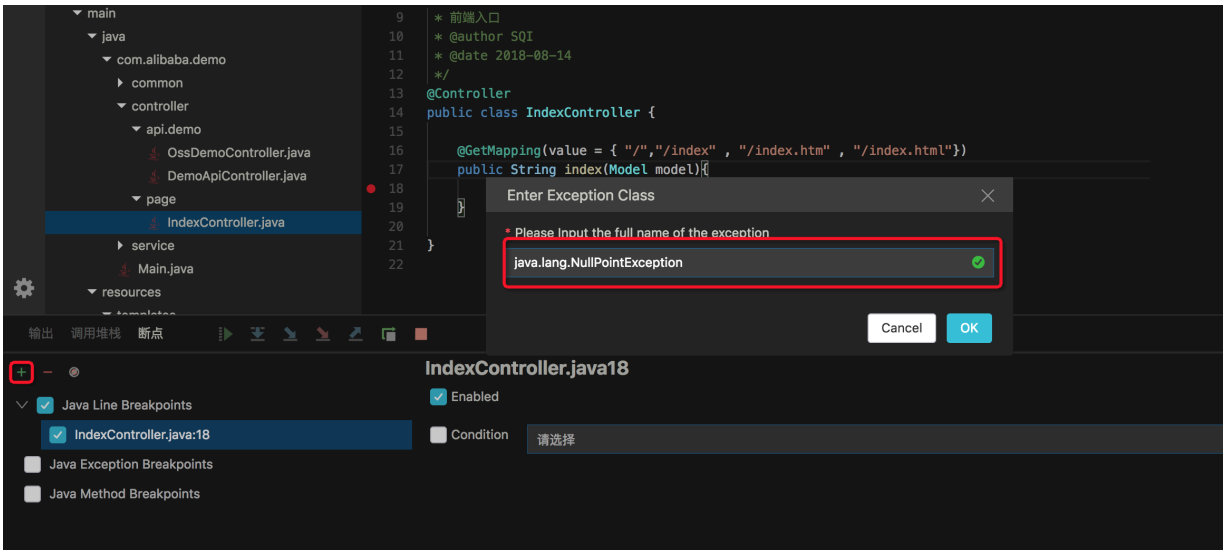


触发它后可以看到，进入该函数时会暂停，即将跳出程序时也会暂停。

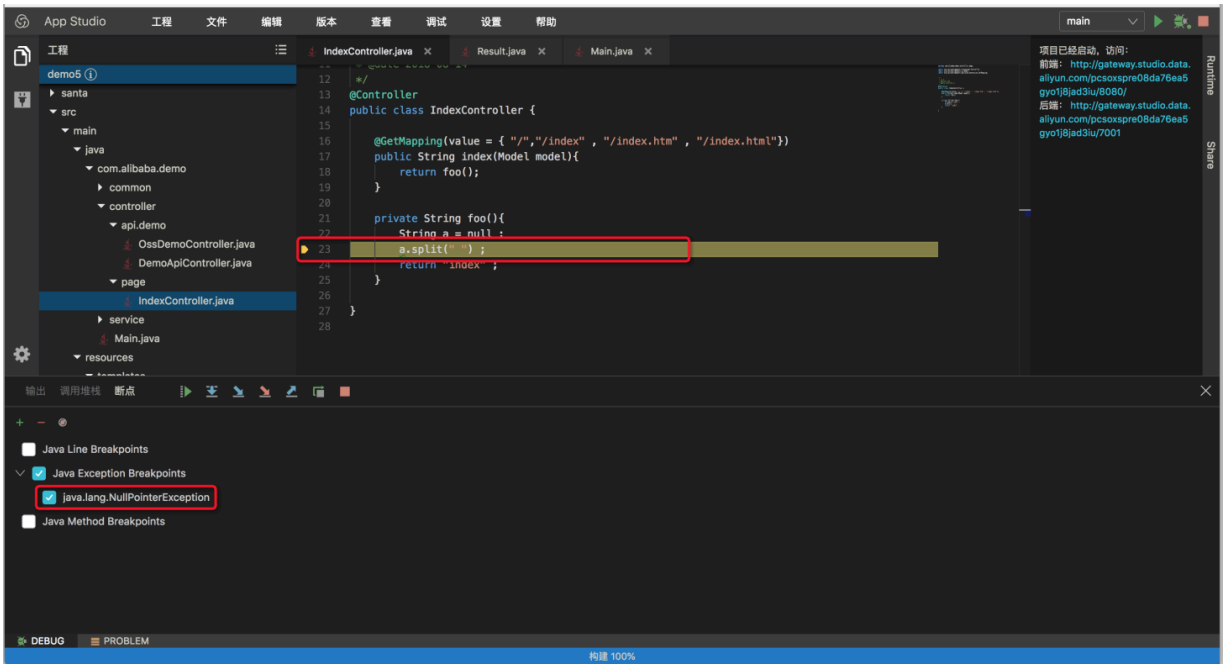


异常断点

如果配置了异常断点，当程序在遇到异常时，会在出现异常的地方进行断点。



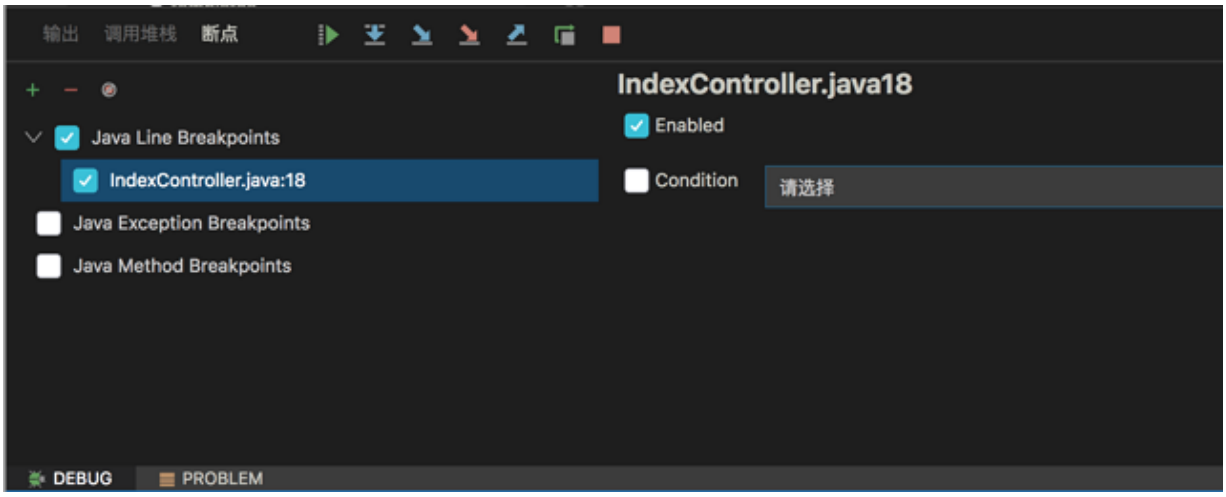
触发index，由于出现了NullPointerException，所以断点在23行。



13.4.5.4 断点及操作

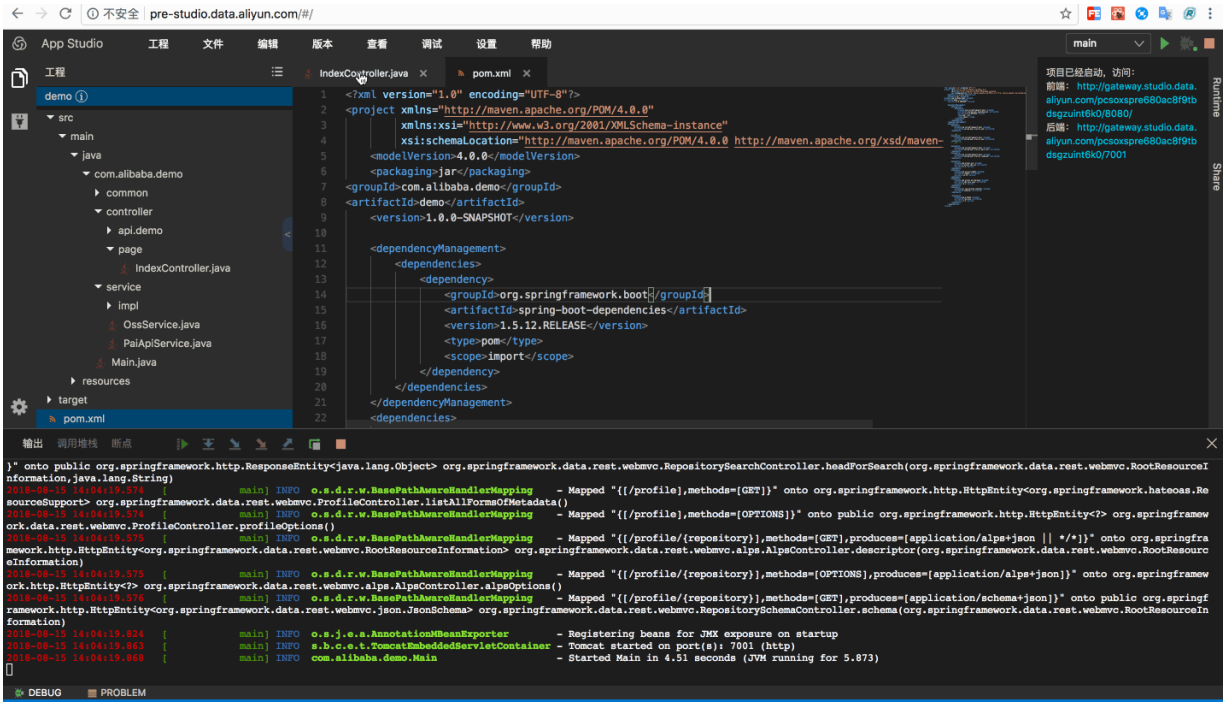
断点面板为您展示当前设置的所有断点，本文将为您介绍断点的操作。

断点包括普通行断点、函数断点和异常断点三种断点类型，详情请参见断点类型。



调试操作

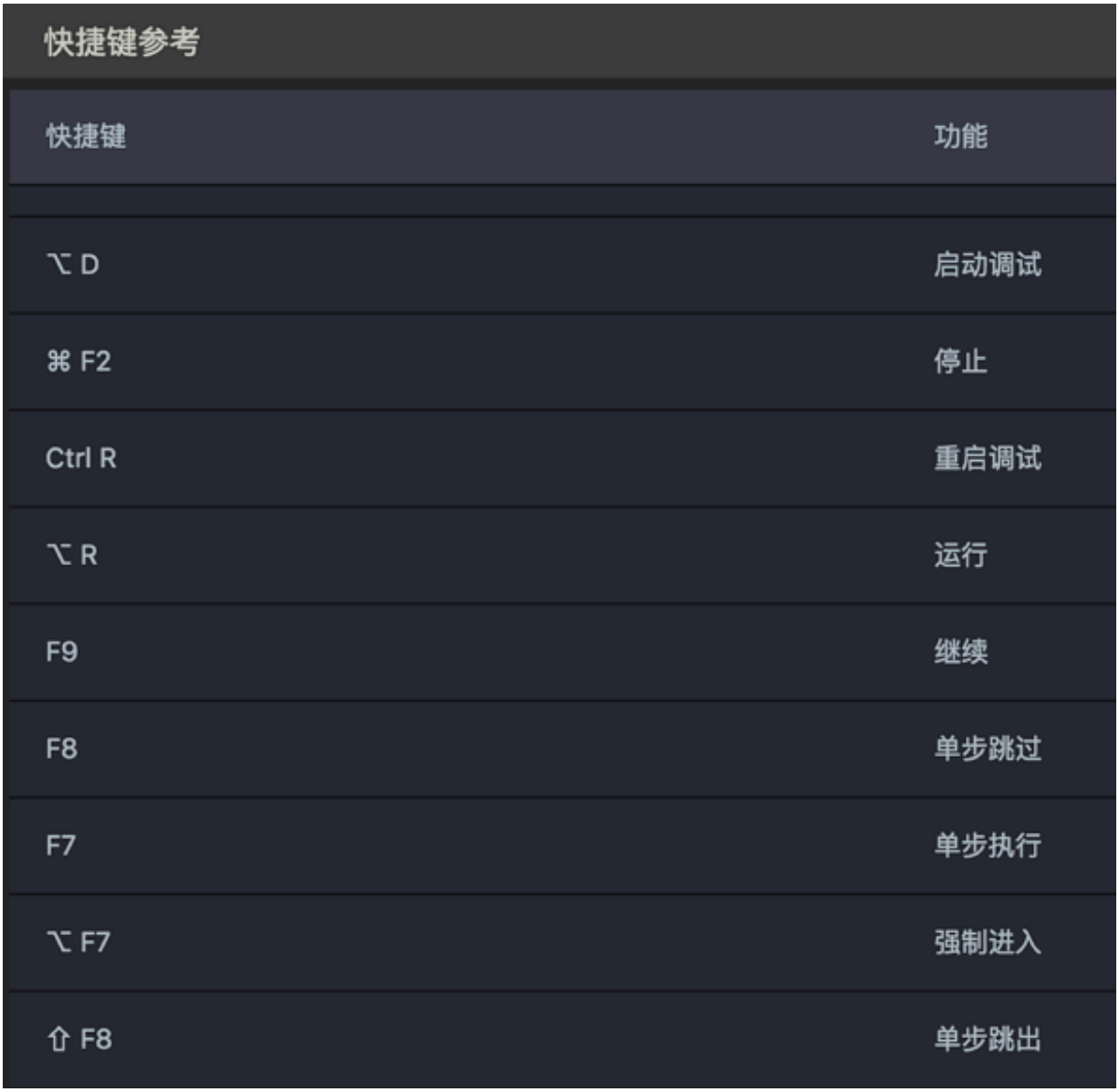
调试界面如下所示。



上图中从左到右的每个按钮所代表的含义如下。

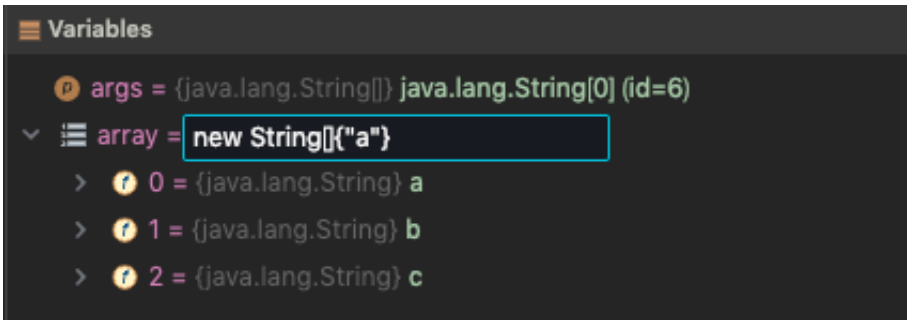
功能	说明
continue	恢复当前断点时，当前线程继续运行。
step over	执行到下一行。
step in	进入函数。
force step in	强制进入函数，与step in的区别在于，它可以引导断点执行到Java自带的类库中。
step out	从当前函数跳出。
restart	目前的restart实现方式可能无法完成程序清理等工作，正在优化中。
stop	停止调试。
Drop Frame	删除当前栈，回退到上一个函数。
Run to Cusor	执行到当前行，可以在某一行打一个临时断点。
计算表达式	可以任意执行一个表达式进行计算。

操作的快捷键如下所示。

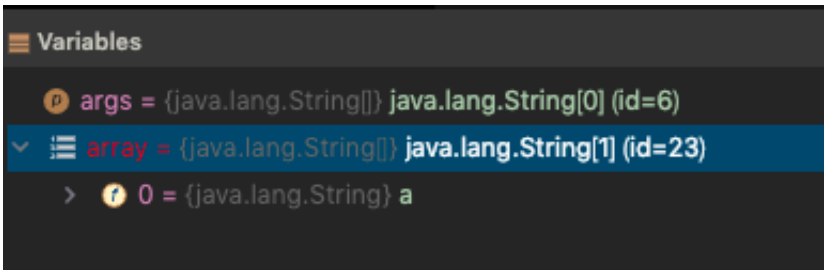

快捷键参考
快捷键
功能
⌥ D
启动调试
⌘ F2
停止
Ctrl R
重启调试
⌥ R
运行
F9
继续
F8
单步跳过
F7
单步执行
⌥ F7
强制进入
⇧ F8
单步跳出

- **对变量进行赋值**

  您可以直接在断点上对一个变量进行赋值。



  双击某个字段，然后构造一个表达式对当前值进行赋值，按enter键生效。

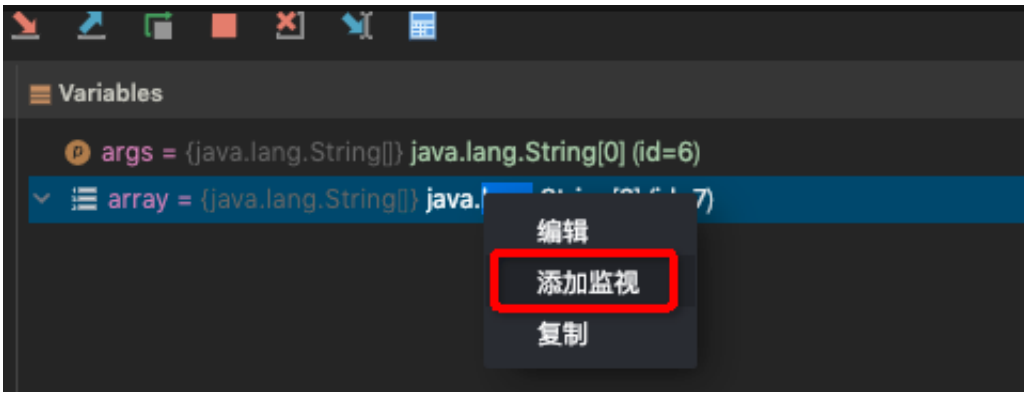


- **表达式计算**

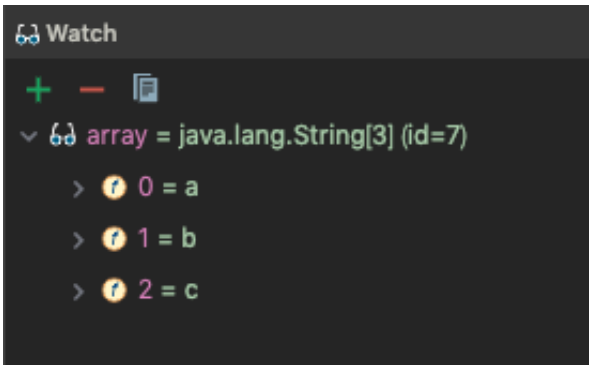
  打开计算表达式面板，输入可执行表达式。

- 变量监视

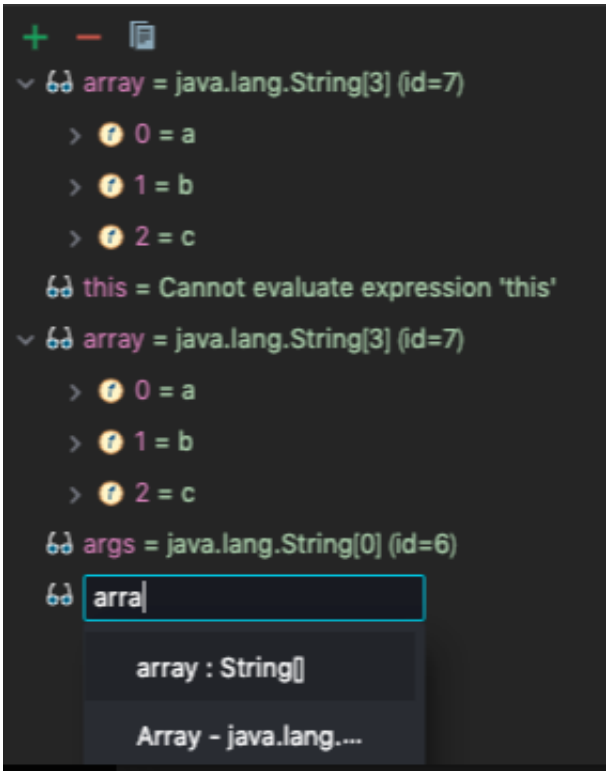
  您可以右键单击变量，选择添加监视。



  添加后，即可在右边面板看到相应的变量监视。

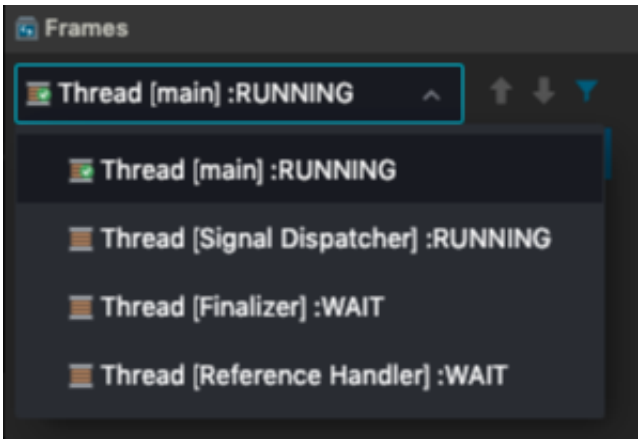


  您也可以在watch上手动新增变量。



- 线程操作

  您可以在调试面板查看线程操作。

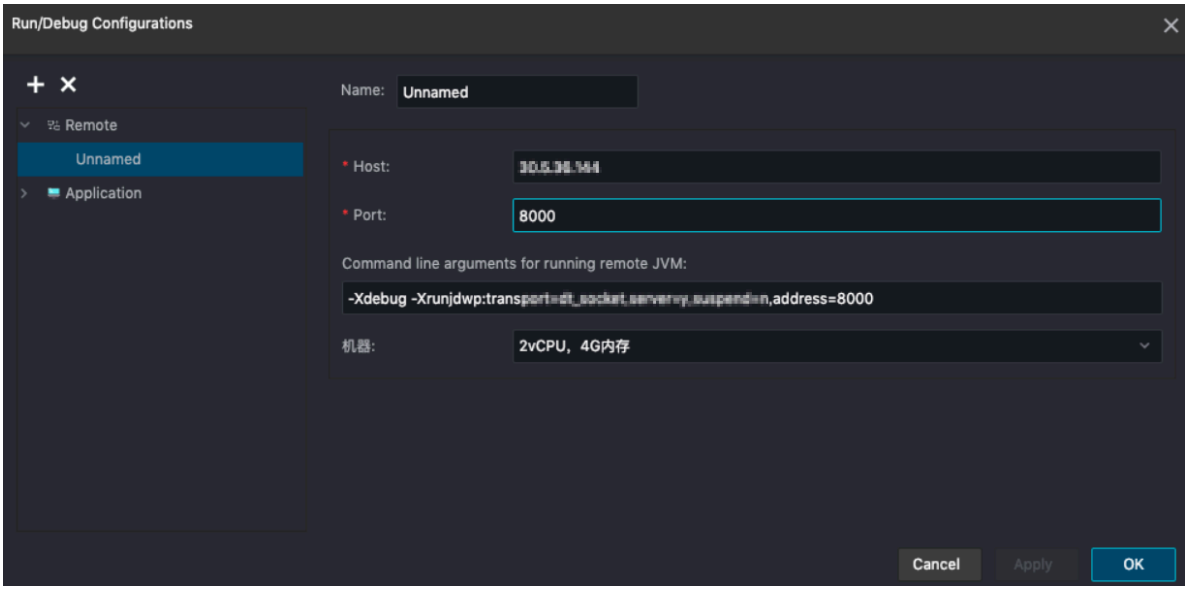


  根据当前线程的运行进度，下拉框中会显示RUNNING或WAIT等不同的信息。当您选中另外的线程时，变量的面板信息也会随之改变。

13.4.5.5 远程调试

由于调试机器在日常环境，因此只能调试日常环境上部署的应用。

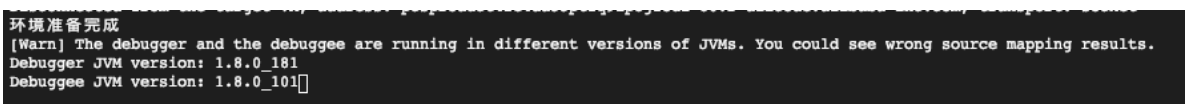
1. 配置调试信息。



   说明:

   您需要填写Host和Port信息，告知JVMTI需要连接哪个远端服务。

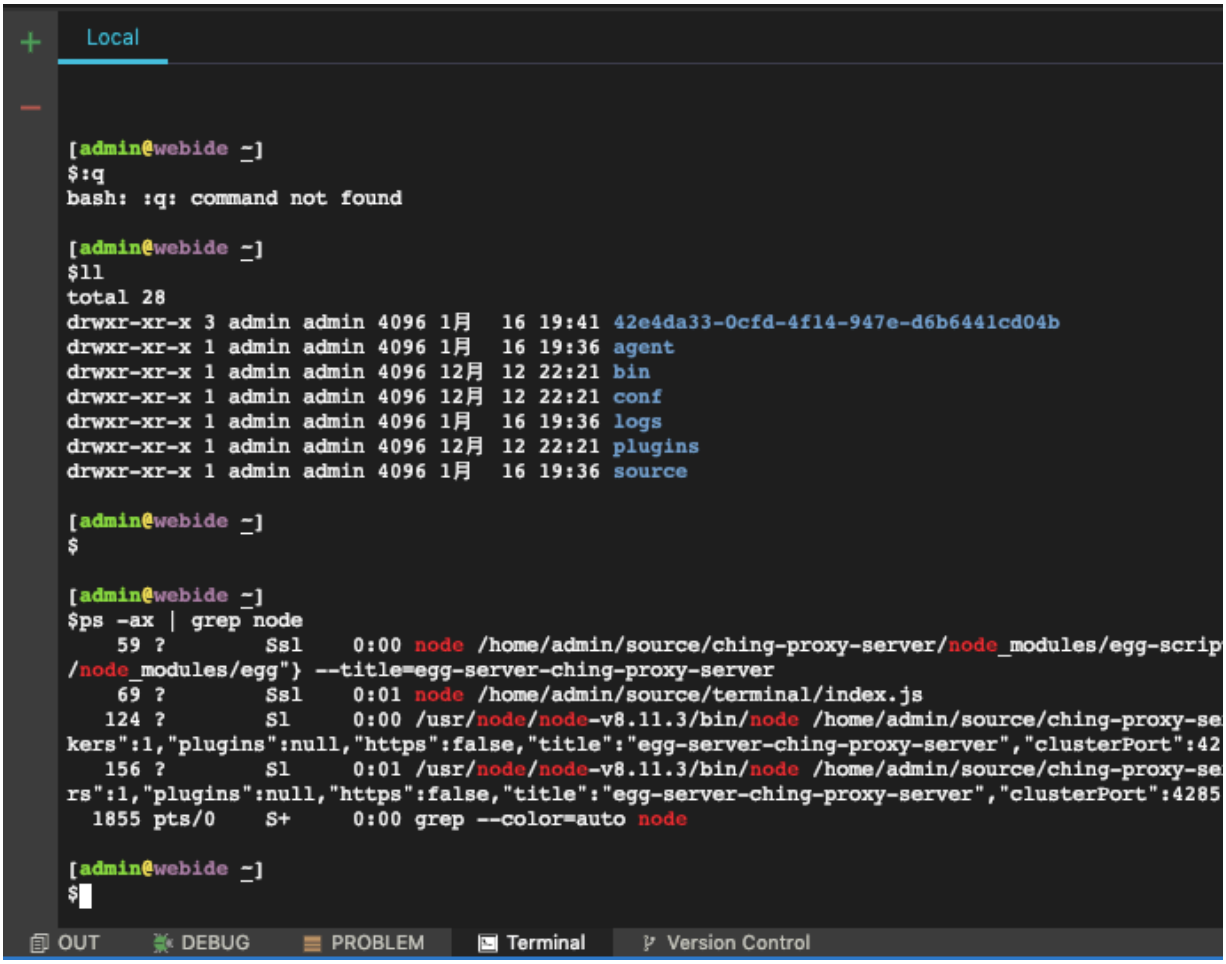
2. 单击调试，遇到Debugger信息时，代表连接成功，即可开始调试。



远程调试使用JVMTI进行Socket连接，本质上Debugger和Debuggee之间仅传输JVM运行信息，不会传输标准输出和错误输出。

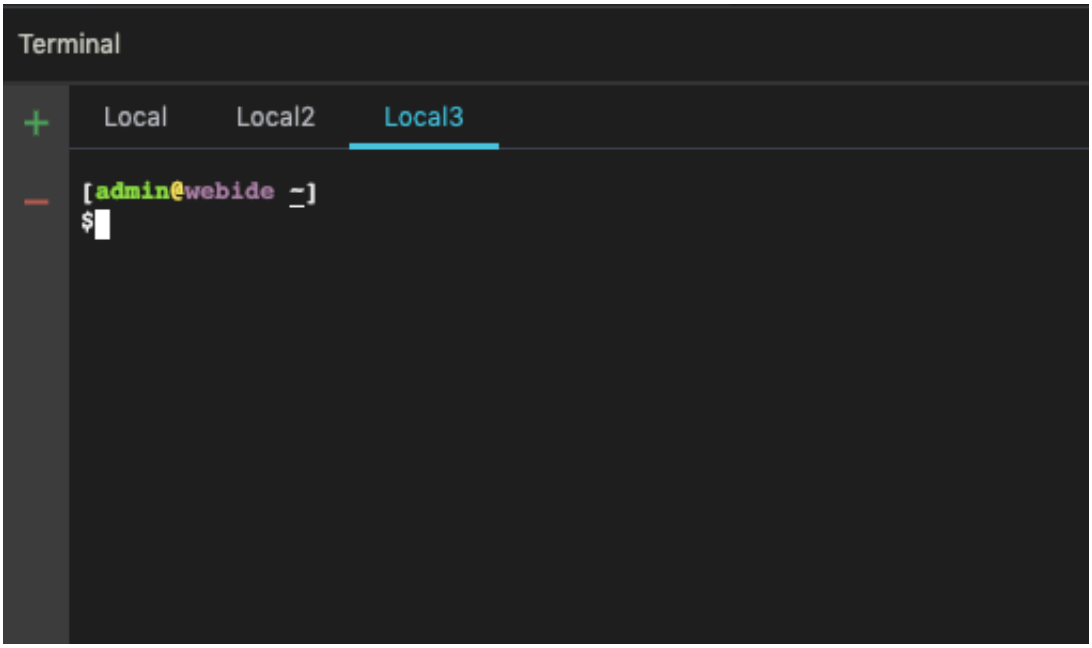
13.4.5.6 终端

Terminal按钮显示在页面底部。



App Studio支持常规的`ls`、`cat`等Shell命令和`vi`、`top`等带有交互的命令。

您可以开启多个终端。

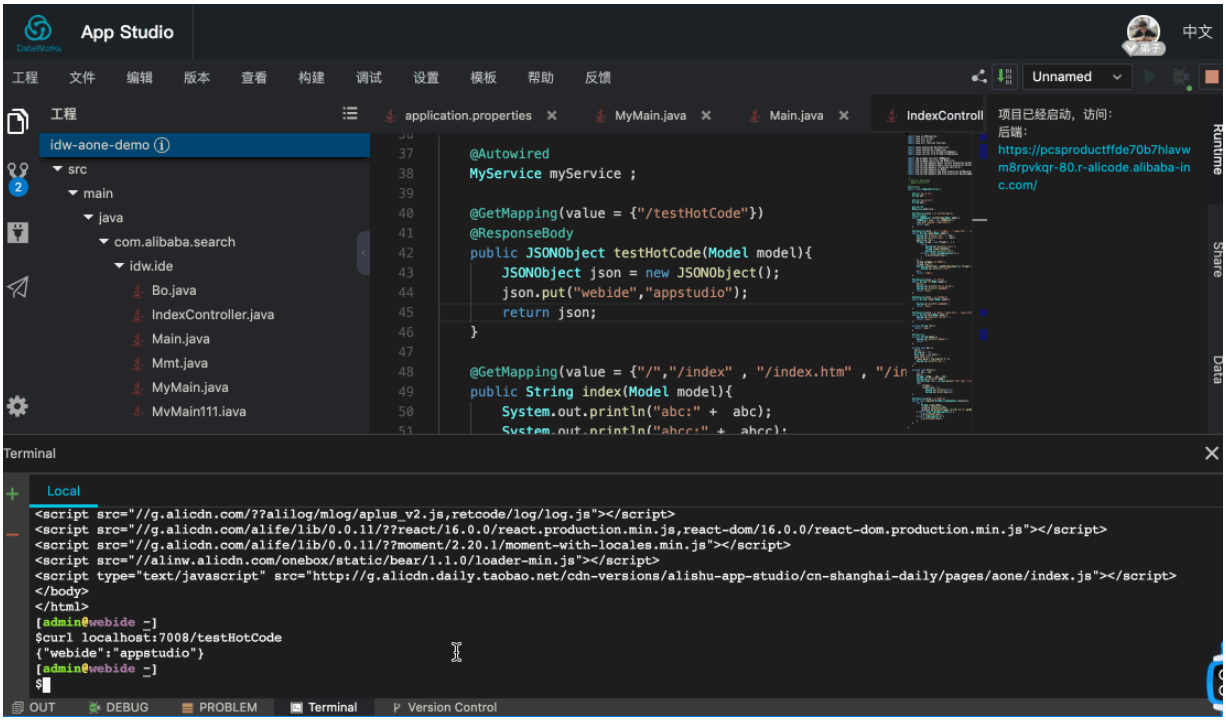


13.4.5.7 热部署

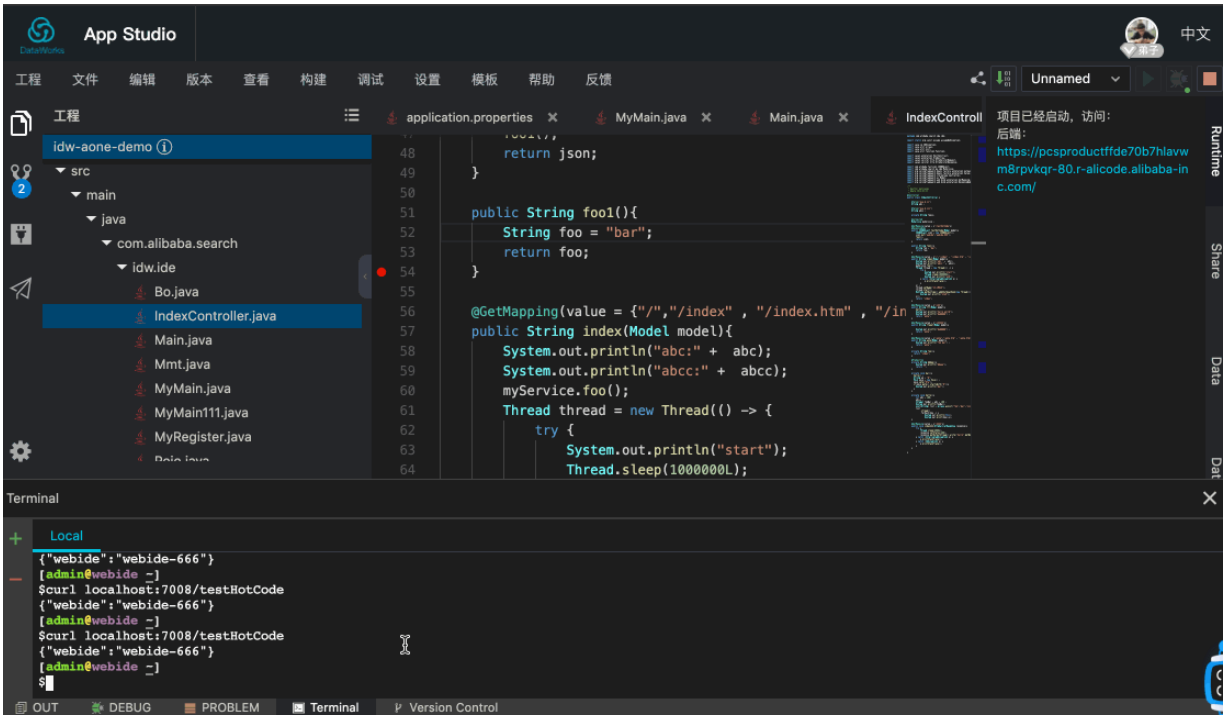
热部署是指代码运行过程中，您手动修改的代码可以在不重启服务的情况下生效。

例如SpringBoot在运行/调试过程中，修改完代码后无需重启，保存即可生效，App Studio已经默认包含此功能。

除调试模式外，运行模式下也支持这项功能。触发热部署无需安装插件和手动编译文件，您只需保存文件即可。

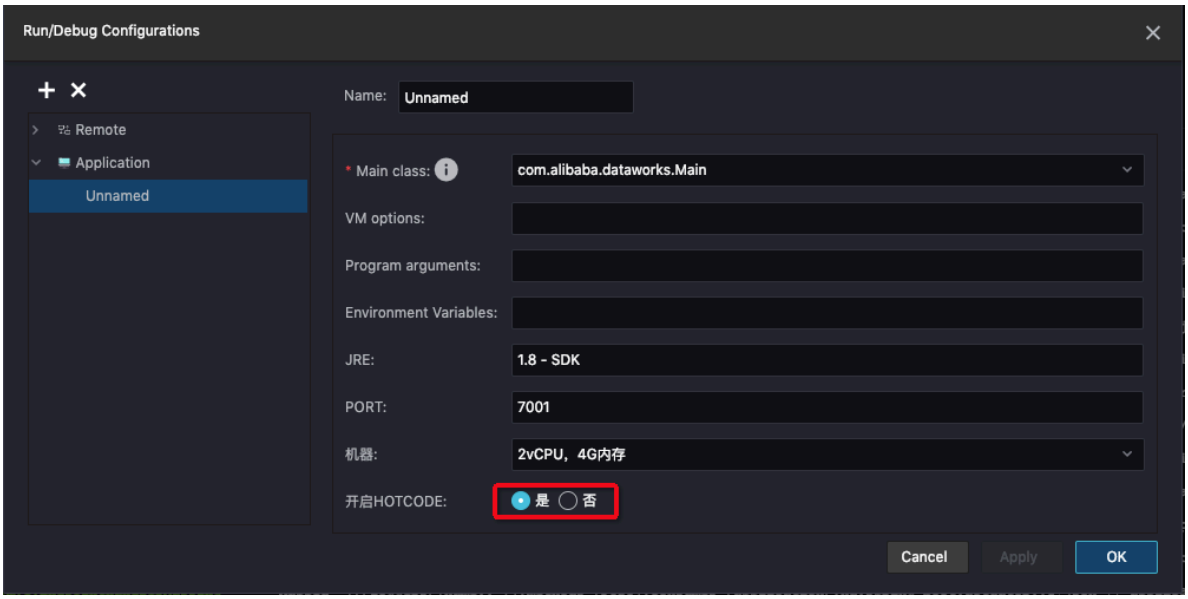


如果您正在Debug中进行代码变动，会自动删除当前运行栈，回退到函数入口。



运行模式下的热部署配置

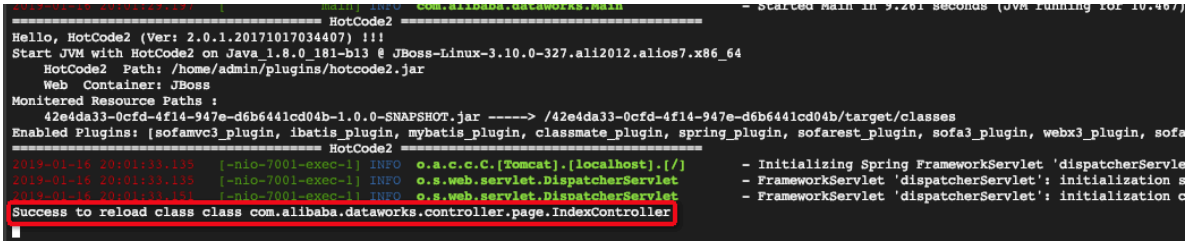
1. 在配置面板上主动开启热部署。



启动后，即可在输出中看到HotCode2的输出信息。

2. 触发热部署。

当您对文件进行修改时，需要手动触发文件保存。



3. 当代码增量同步完成后，控制台显示Reload某个类的输出，则代表热部署生效。代码示例如下：

```
public class IndexController {
    @RequestMapping("/")
    @ResponseBody
    public String index(){
        return "cccc";
    }
}
```

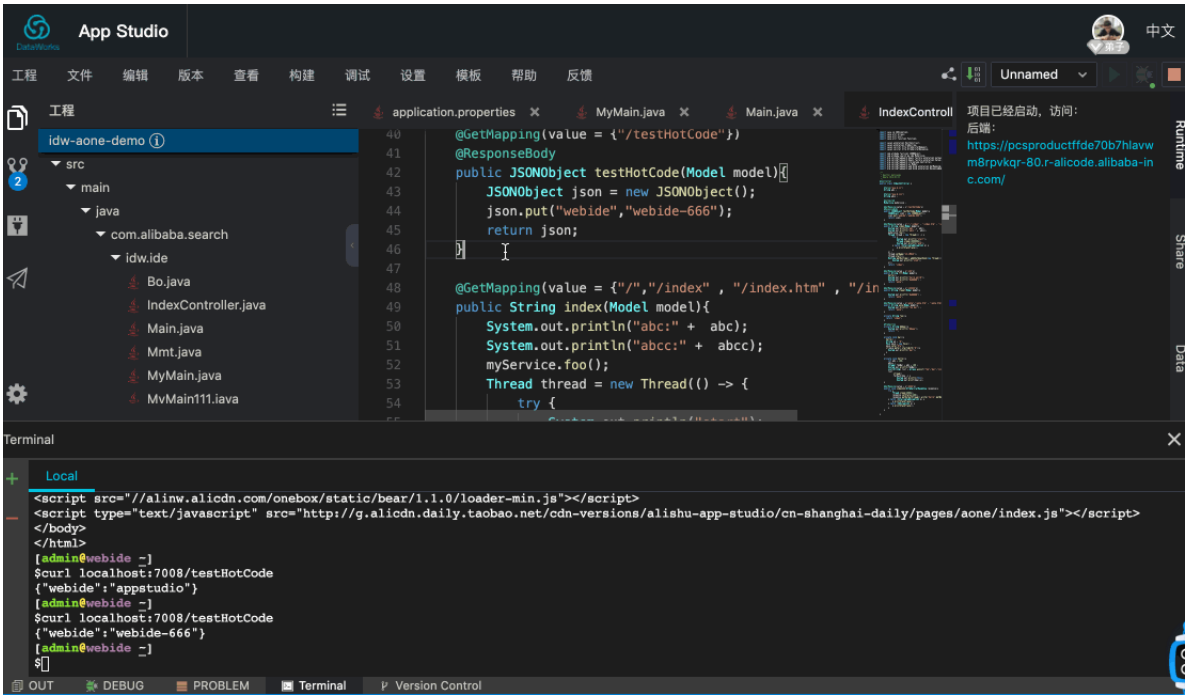
您可以将Return字符串内容改为其它字符串，让其立即生效。

Debug模式下的热部署

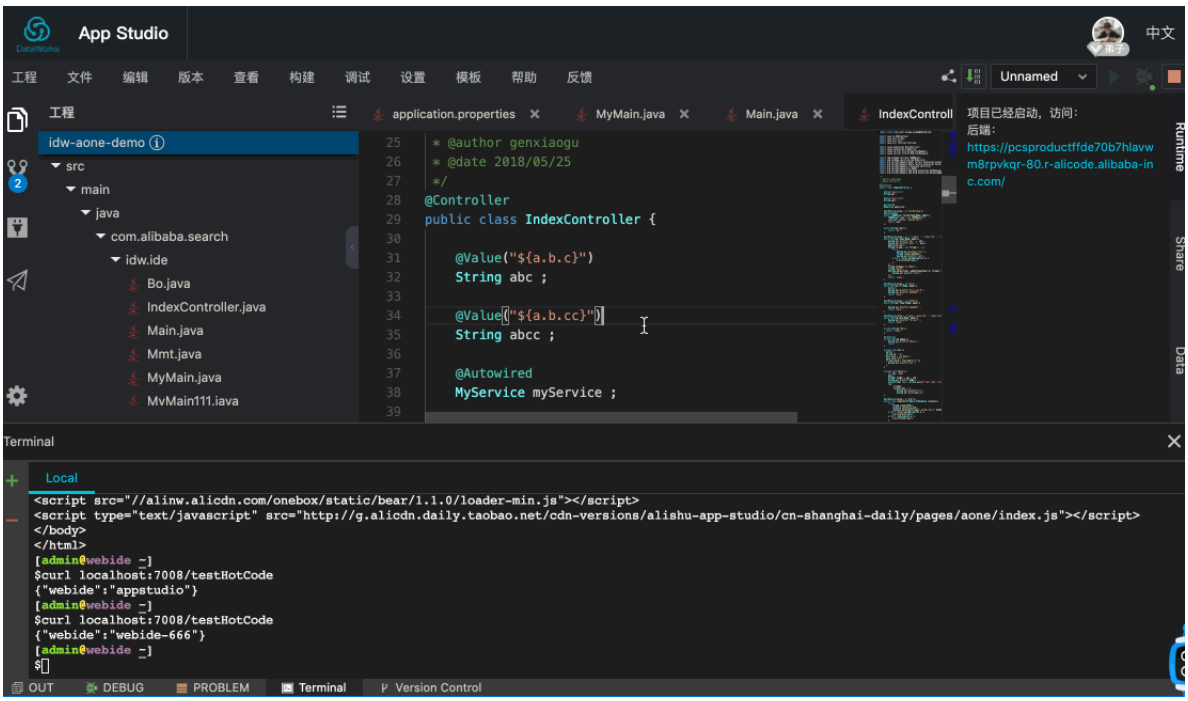
您可以通过JDI原生方法实现Debug模式下的热部署，但由于JVM的限制，在给某个类增加或删除方法时，无法进行热部署。您同样只需保存文件即可触发热部署。

> **说明:**
> JVM原生不支持对类结构进行变动后的热部署，新增或删除类等其他操作都可以支持热部署。

- 新增方法或删除方法。



- 新增字段。



13.4.6 协同编程

本文将从实时协同编辑、邀请协作者、加入写作项目、协作者面板和权限等方面为您介绍协同编辑。

App Studio支持实时协同编辑功能，团队中多个成员可以同时在同一个项目中开发、编写代码，并实时查看其它成员的改动。能够避免同步代码、合并分支的繁琐，显著提升开发效率。

邀请协作者

项目的所有者可以邀请其他开发者加入项目进行协作。

1. 打开要分享的项目。
2. 单击右侧的Share展开协作者面板。
3. 单击右上角的邀请，进入邀请流程。

4. 填写邀请协作者对话框中的各配置项。

配置	说明
用户名	填写邀请的写作者的用户名。
权限	根据自身需求选择只读或读写权限。

5. 单击确认，即可成功邀请。

加入协作项目

当您被邀请加入其他开发者的项目后，可以在打开的工程面板下，选择我参与的，查看您加入的协作项目。单击即可加入项目，开始实时协同编辑。

协作者面板

实时协同编辑时，协作者们可以互相查看当前的状态。

1. 单击页面右侧的Share展开协作者面板。

2. 查看相应协作者的在线状态、正在编辑的文件和拥有的权限。

> 说明:
> 项目所有者可以移除协作者。

权限说明

在协同编辑的过程中，参与的协作者权限分为以下三种：

- 所有者：所有者是项目的创建者，无法变更。所有者可以邀请其他开发者加入项目，也可移除其他的协作者。
- 读写权限：拥有读写权限的协作者可以查看项目中的所有文件，也可以对这些文件进行编辑。
- 只读权限：拥有只读权限的协作者只能查看项目中的文件，但是无法进行编辑。

13.4.7 应用部署

本文将为您介绍如何在App Studio上新建一个应用并部署到生产环境，获得一个可以通过公网访问的应用。

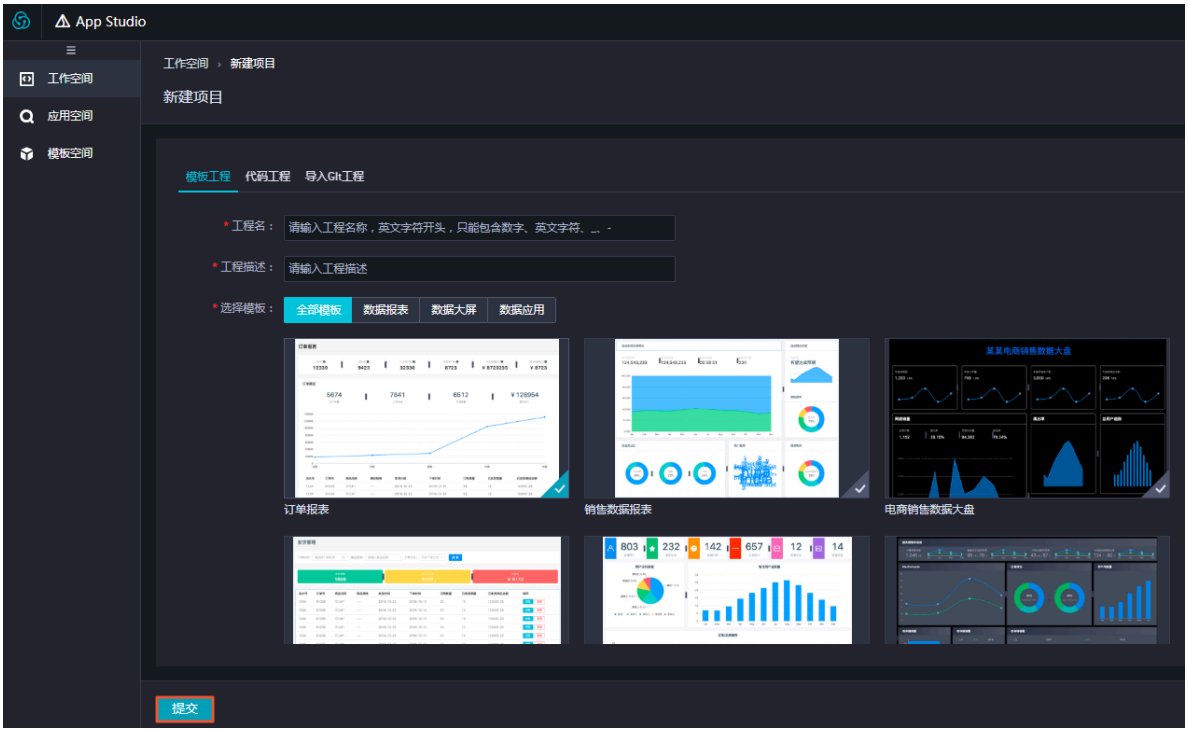
进入App Studio

新建工程

1. 登录DataWorks控制台，单击相应工作空间后的进入数据开发。

2. 单击左上角的图标，鼠标悬停至全部产品，选择应用开发 > App Studio。

3. 进入App Studio页面后，您可以通过模板、代码和Git导入三种方式创建工程。

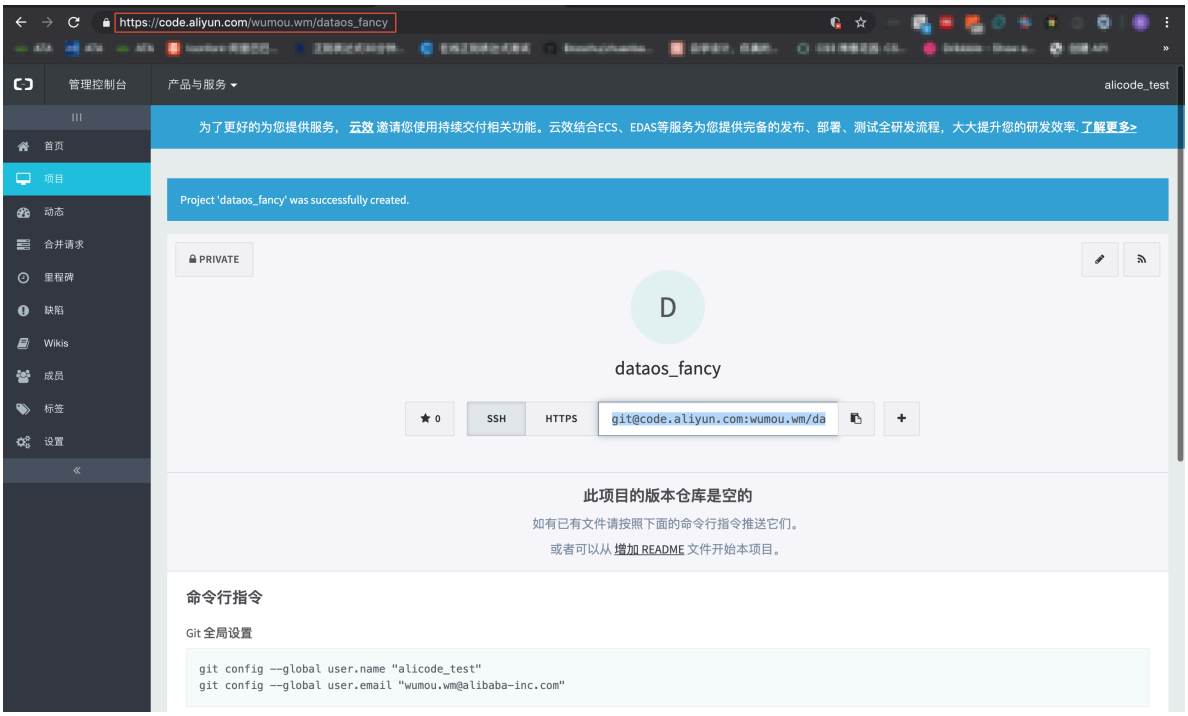


4. 根据自身需求填写配置后，单击提交，即可新建工程。

关联Git

发布应用前，需要初始化Git。

1. 首先在Code页面新建一个repo，并记下仓库的SSH地址。



2. 进入App Studio页面，单击版本，选择初始化&关联远程仓库。

3. 填写关联远程仓库对话框中的配置，单击提交。

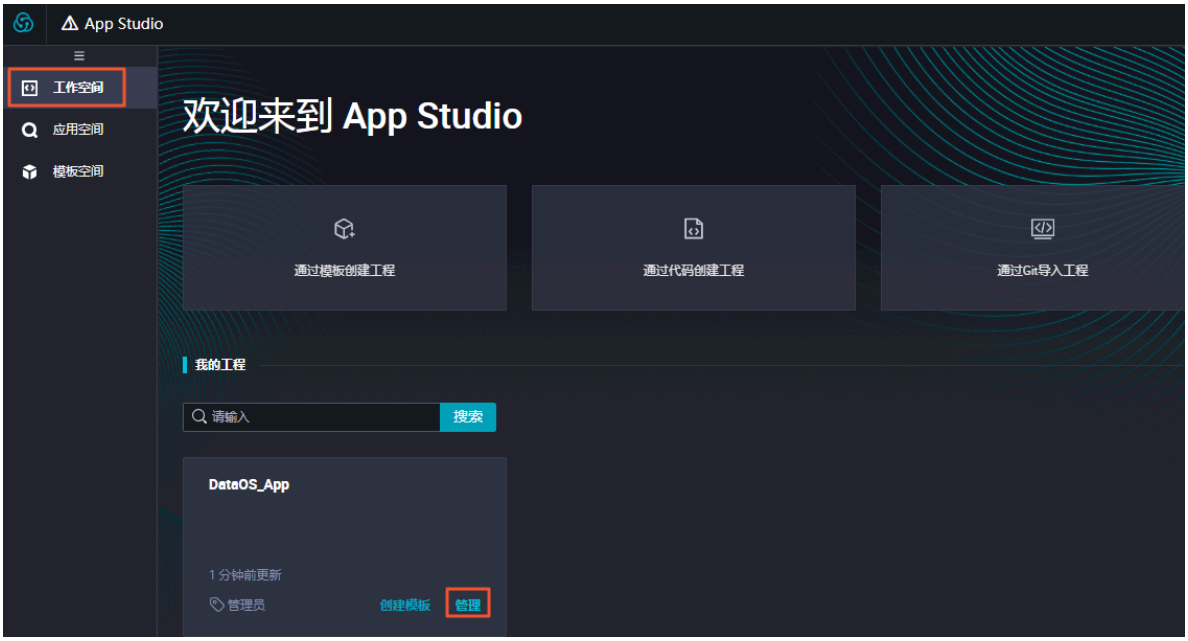
说明:

如果您未绑定SSH KEY或Git用户名邮箱，可根据页面引导进行操作。

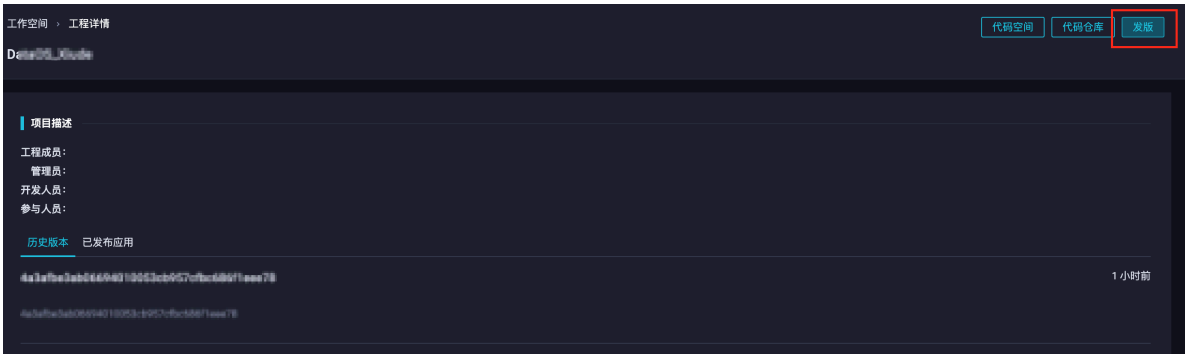
发版

关联Git完成后，即可通过发版创建应用。

1. 返回工作空间页面，单击相应工程下的管理。



2. 单击右上角的发版，填写应用名称。



3. 单击发版。

部署应用

1. 单击发布，出现如下图所示的引导页面。

   您需要根据指引访问购买页面，购买运行空间。然后进入运维平台，创建分组，并将购买的机器加入分组。

2. 单击购买链接，根据指引在相应的Region购买App Studio运行空间。

3. 单击部署控制台，进入运维页面。

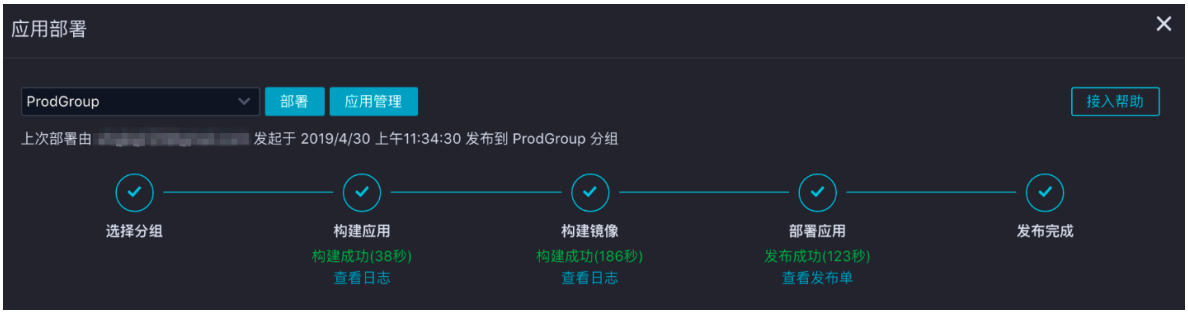
> **说明:**
> 此时需要解绑之前绑定的Host。

4. 单击分组列表下的创建分组，完成分组的创建。

5. 选择操作 > 应用扩容，将刚刚购买的机器加入创建的分组中。

6. 完成后会刷新应用空间，单击部署，将应用发布到默认的分组即可。

出现下图中的状态，代表发布完成。此时应用已经部署到您的ECS，并启动服务。



VPC下沉

VPC下沉是指将VPC加入到用户购买机器的网段。该操作需要在阿里云和App Studio应用运维平台实现，且每个项目仅需执行一次，之后的版本迭代只需执行上面的部署应用即可。

VPC接入授权

App Studio用于发布的ECS通过弹性网卡和用户VPC连通，需要用户给App Studio的服务账号添加网卡权限， 提交给运维平台。

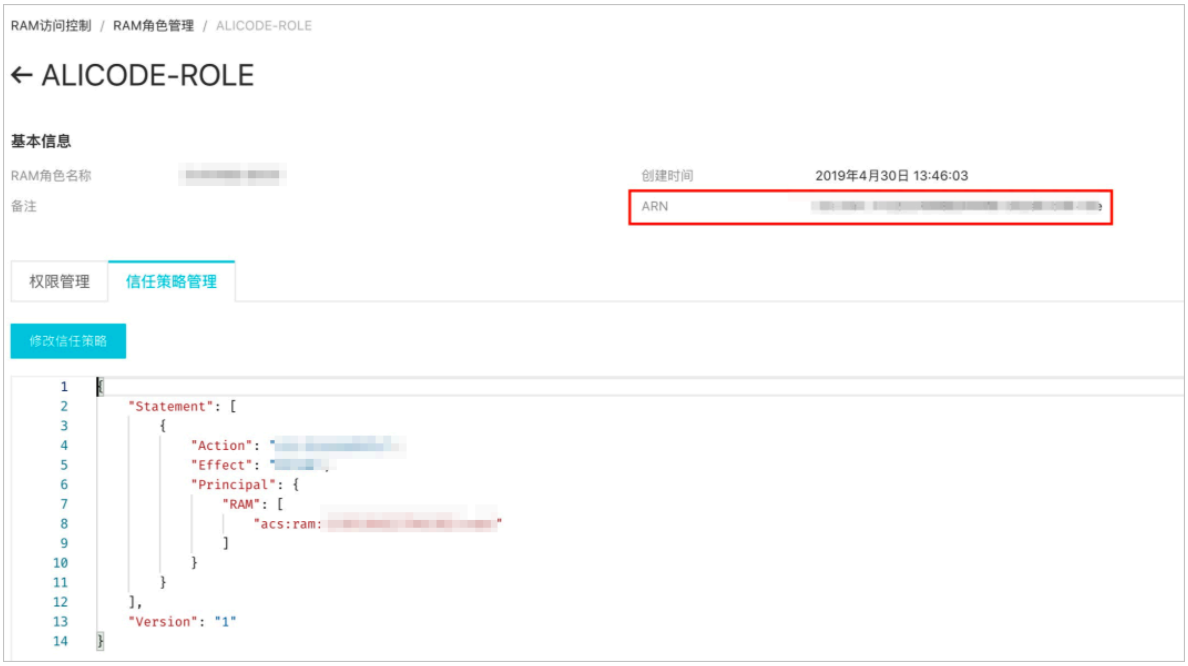
1. 进入角色管理页面，单击新建角色，选择阿里云账号和其他云账号1591568227964362，自行选择角色名称，单击确定。

说明：

此处的其他云账号固定选择为1591568227964362。

2. 单击相应RAM角色后的添加权限，为其添加管理ECS弹性网卡的权限，完成选择后单击确定。

3. 进入相应的RAM角色，查看ARN。



创建专有网络和交换机

创建专有网络和交换机需要在App Studio相同的Region进行，此处以上海Region为例。

进入VPC控制台创建专有网络，具体操作请参见#unique_663。

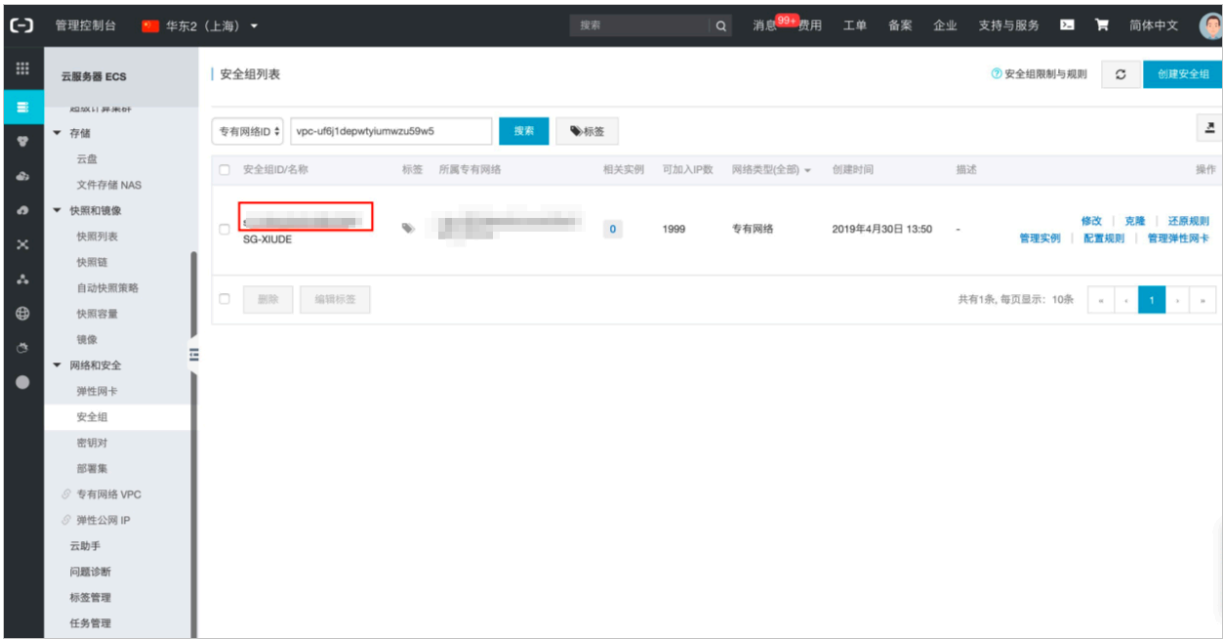
> **说明:**
> 专有网络的IPv4网段需要选择与部署应用前选择的网段不同的网段。

创建完成后，在交换机页面记录下交换机的ID进行备用。

创建安全组

进入ECS控制台创建安全组，详细操作请参见#unique_664。

安全组创建完成后，请记录安全组的ID进行备用。

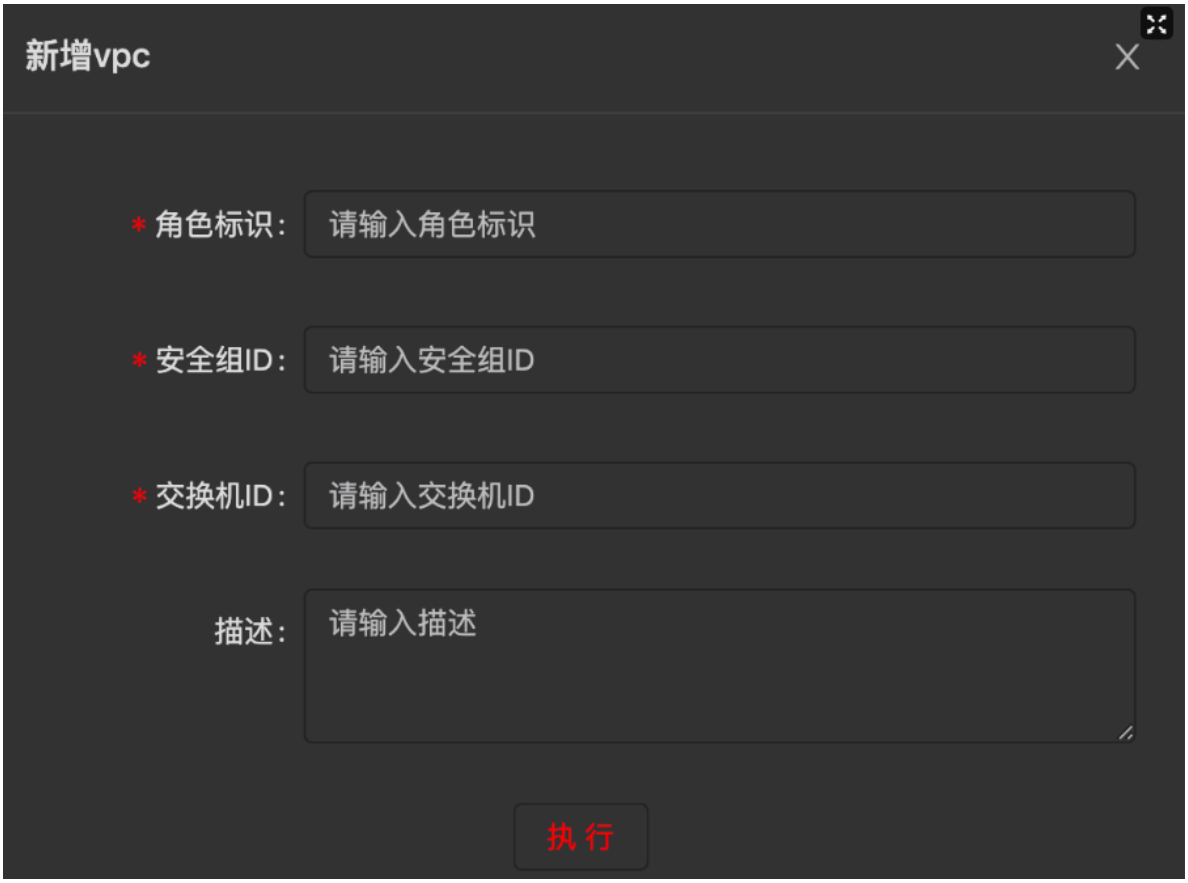


在运维平台添加用户VPC

1. 单击App Studio页面右上角的运维。

2. 进入资源 > VPC页面，单击新增VPC。

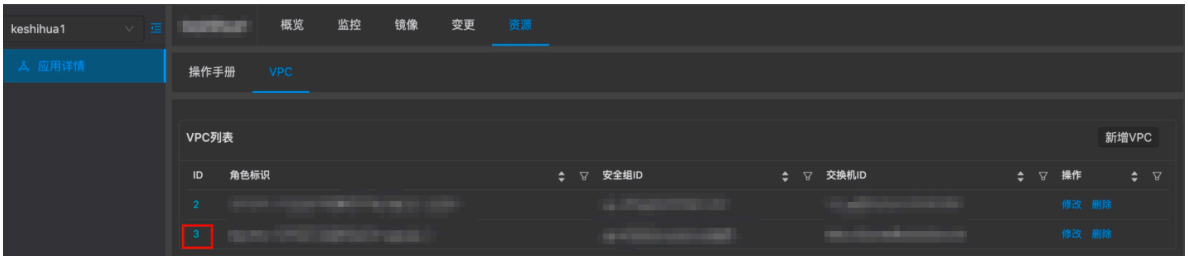
3. 在新增vpc对话框中填写之前记录的角色标识（即ARN）、安全组ID和交换机ID，并进行相应的描述。



4. 配置完成后，单击执行。

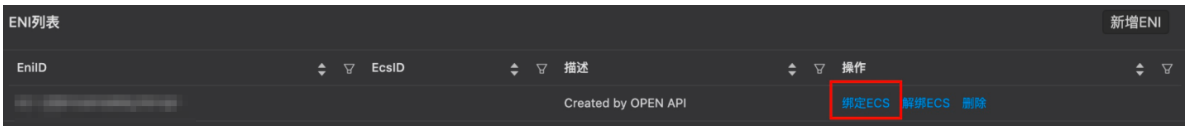
创建弹性网卡并绑定ECS

1. 单击相应VPC的ID，进入ENI管理页面。



2. 单击新增ENI。

3. 新增完成后，单击绑定ECS。



4. 在绑定ecs对话框中选择相应的VpcID、EniID、分组和机器。

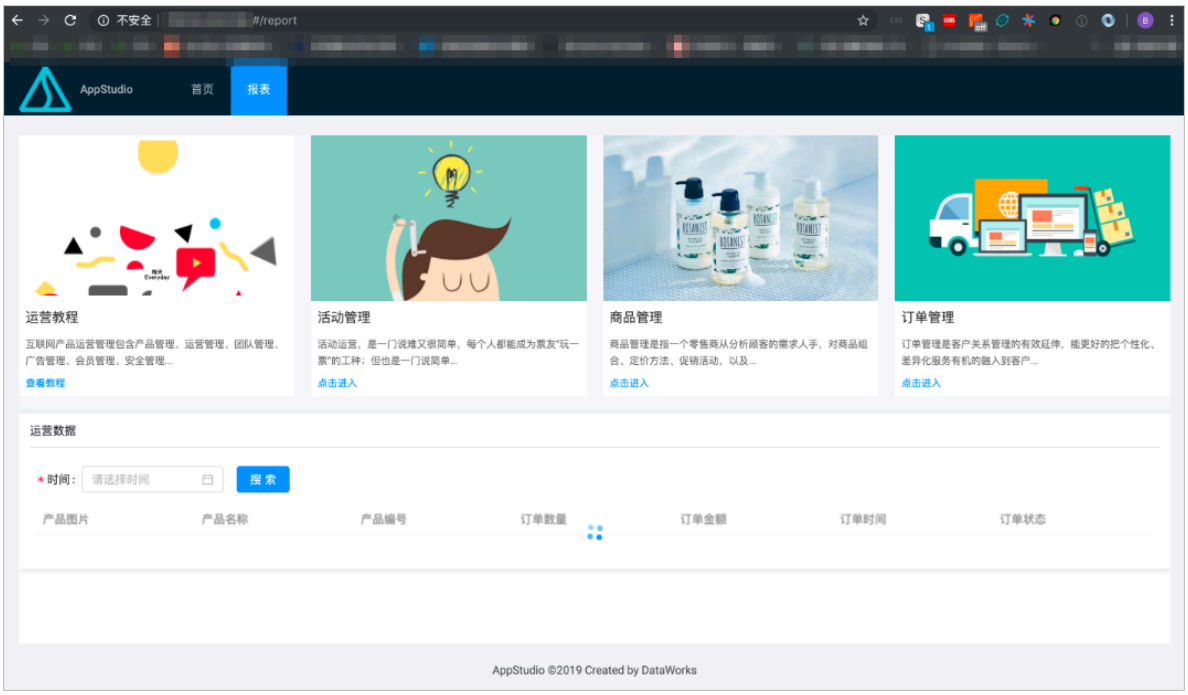
完成上述操作后，App Studio会为您创建弹性网卡，并绑定到机器实例。

公网访问

接下来，您可以通过将弹性网卡绑定至弹性公网IP的方式，将应用透出至公网。您也可以在其中加入负载均衡的服务。

通过弹性公网IP将应用透出至公网的操作，如下所示。

1. 访问VPC控制台购买弹性公网IP，具体操作请参见#unique_665。
2. 绑定弹性网卡，具体操作请参见绑定弹性网卡。
3. 完成上述操作后，即可通过公网IP访问您的服务。



13.4.8 第三方服务接入

13.4.8.1 数据服务

本文将为您介绍如何在App Studio中查看用户有权限调用的数据服务，并通过App Studio生成快速访问数据服务API的代码片。

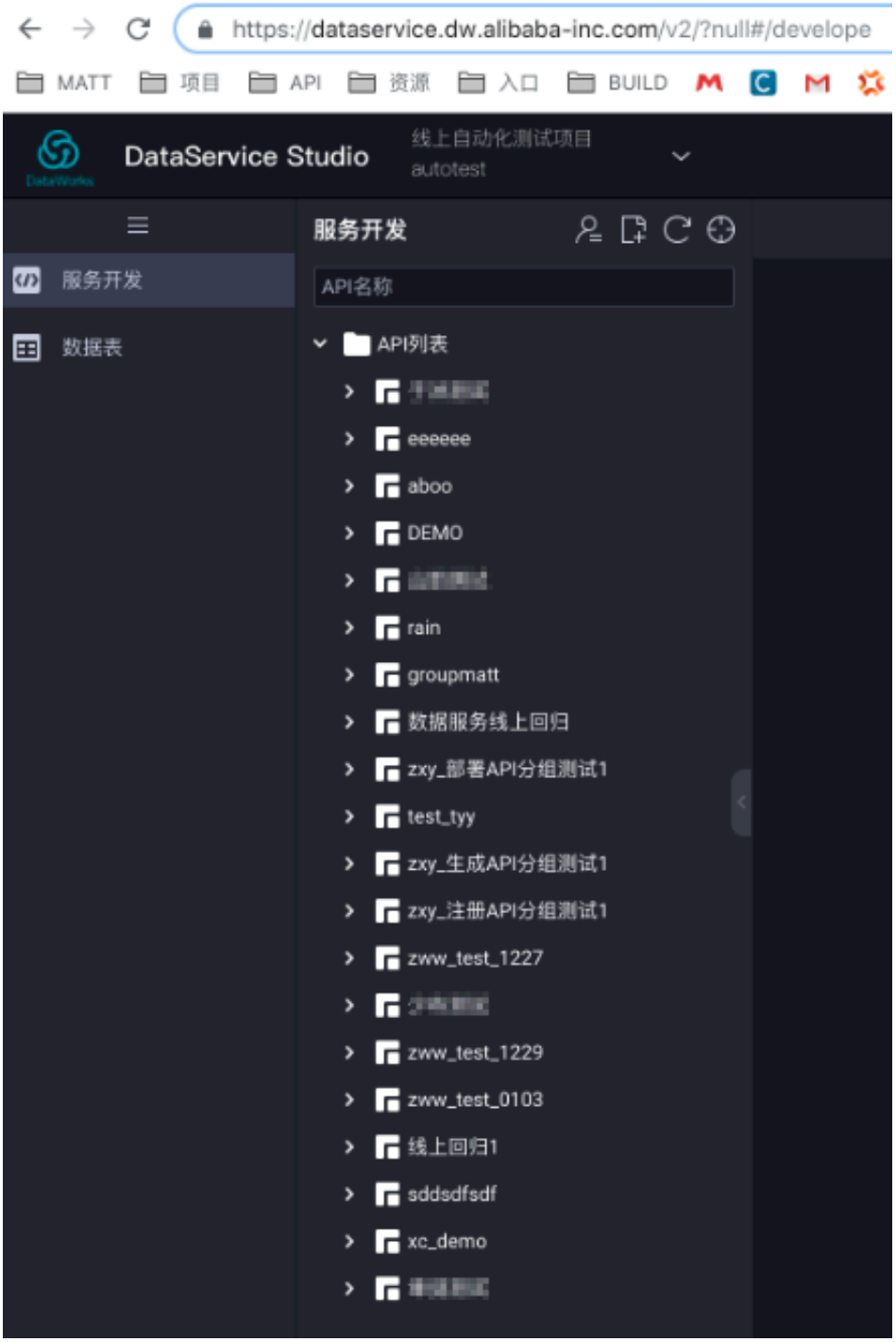
如果您想要获取更多数据服务API的申请、SDK以及调用方法，请参见数据服务。

准备工作

在开始操作前，需要首先准备以下内容。

- 确认在数据服务中有相关工作空间的权限和API。

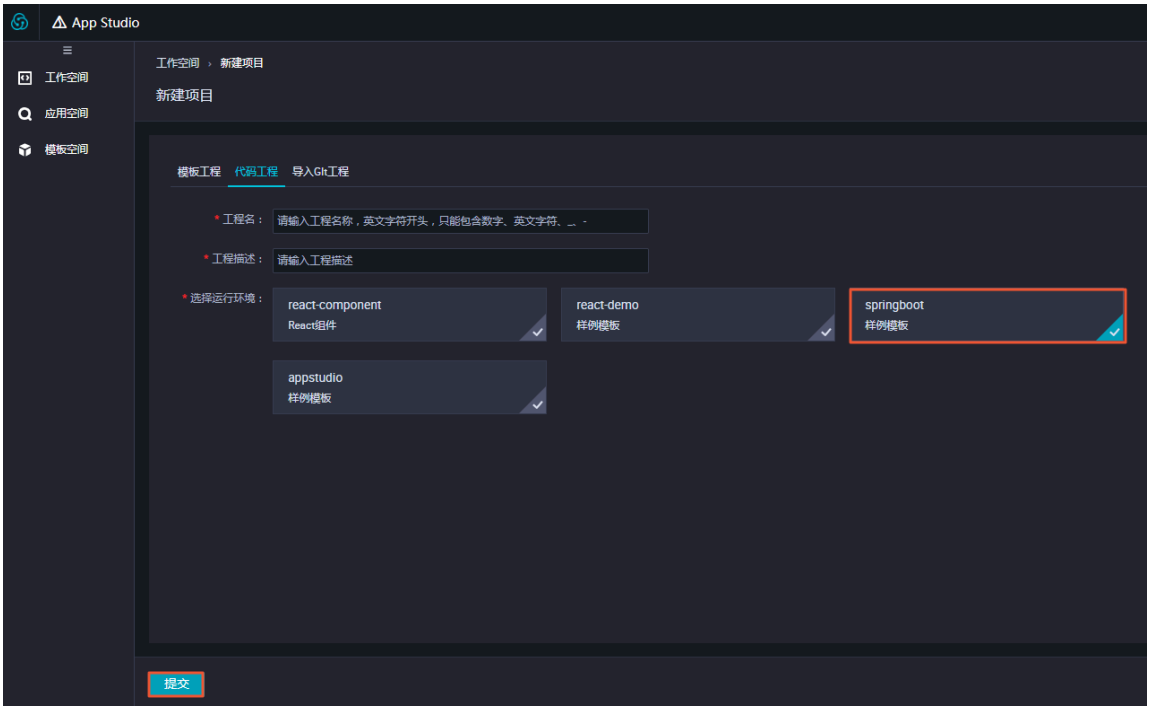
  由于本文内容适用于有权限的数据服务API， 所以请首先进入数据服务页面，查看是否有DataWorks工作空间，并查看相应工作空间下是否存在有权限的API。



- App Studio中准备一个Java项目。

  以springboot类型的项目为例，为您介绍代码片生成的功能。

  1. 进入App Studio页面，单击工作空间页面的通过代码创建工程。
  2. 填写新建项目对话框中的工程名和工程描述，选择运行环境为springboot样例模板。



  3. 配置完成后，单击提交。

  项目创建完成后，请确保pom.xml中有数据服务的依赖，maven坐标： Nexus Repository Manager。

```
<dependency>
  <groupId>com.aliyun.dataworks</groupId>
  <artifactId>aliyun-dataworks-dataservice-java-sdk</artifactId>
  <version>0.0.1-aliyun</version>
</dependency>
```

在App Studio中使用数据服务

您可以直接在代码中使用数据服务，也可以在可视化搭建中使用数据服务。

- 直接在代码中使用数据服务。

  此步骤将为您介绍如何在App Studio中方便地根据关键字、项目和业务分组查看可用的数据服务，同时利用生成代码片的功能快速生成，并调用某个数据服务API的代码。

  1. 查看数据服务API列表。

     打开右侧的Data页面，弹出数据服务API列表，支持按照API名称、项目和服务分组进行筛选。

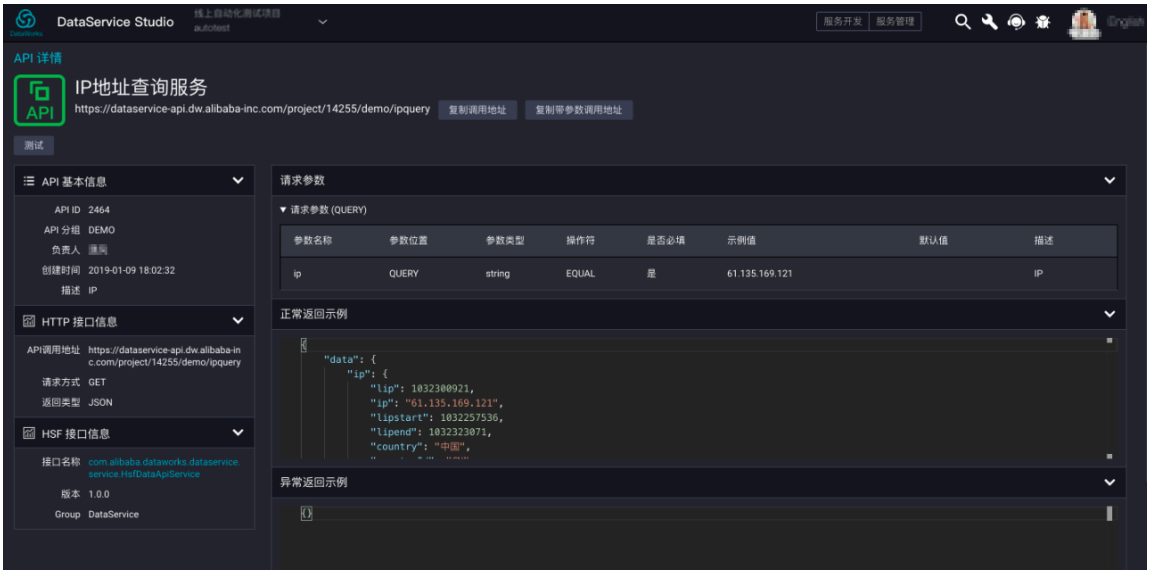
     

  2. 在数据服务页面新增API。

     您可以单击右上角前往数据服务页面新增API，以满足调用API的需求。

  3. 查看数据服务API详情。

     单击单个数据服务API右侧的查看详情，即可跳转至数据服务页面查看API详情。

  4. 快速生成访问代码。

App Studio支持一键生成访问代码的方式， 自动填充appkey、appsecret，生成样例controller代码，方便您直接插入项目。

单击选用链接，即可打开包含样例访问代码的详情页。

完整的controller示例如下所示，仅供参考。 在生成的InvokeApi2252方法中，您访问这个数据服务需要的path、host、key和secret都会被自动填充，ApiRequest2252DTO则包含了访问该服务的所有参数。

```
package com.alibaba.dataworks.dataservice;
import com.aliyun.dataworks.dataservice.model.api.protocol.
ApiProtocol;
import com.aliyun.dataworks.dataservice.sdk.facade.DataApiClient;
import com.aliyun.dataworks.dataservice.sdk.loader.http.Request;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import java.lang.reflect.Field;
import java.util.HashMap;
/**
 * @author ****
 * @date 2019-03-21T17:23:17.040
 * - 使用前，请确保pom.xml包含最新的data-service-client依赖。
 *      <dependency>
 *          <groupId>com.alibaba.dataworks</groupId>
 *          <artifactId>data-service-client</artifactId>
 *          <version>${latest-data-service-version}</version>
 *      </dependency>
 *   - 使用前，确保配置spring config类，需要单独配置，不可与其他config合
并。
```

```
 *     @Configuration
 *     @ComponentScan(basePackageClasses = { DsClientConfig.class
 })
 *     public class DsClientConfig {
 *         @Bean
 *         public BeanRegistryProcessor beanRegistryProcessor(){
 *             return new BeanRegistryProcessor();
 *         }
 *     }
 */
@RestController
public class Test2252Controller {
    private Logger logger = LoggerFactory.getLogger(Test2252Co
ntroller.class);
    @Autowired
    private DataApiClient dataApiClient;
    /**
     * Sample Result:
     * {
     *     "data": {
     *         "totalNum": 1000,
     *         "pageSize": 100,
     *         "rows": [
     *             {
     *                 "pageNum": "...", // 分页默认参数：页编号,
Integer类型。
     *                 "pageSize": "...", // 分页默认参数：页大小,
Integer类型。
     *                 "totalNum": "...", // 分页默认参数：总记录数,
Integer类型。
     *                 "id": "...", // Integer类型。
     *                 "name": "...", // String类型。
     *                 "sex": "...", // String类型。
     *                 "age": "...", // Integer类型。
     *             }
     *             ......
     *         ],
     *         "pageNum": 1
     *     },
     *     "errCode": 0,
     *     "requestId": "478cae2f-0***-42fb-a439-c0***e6f",
     *     "errMsg": "success"
     * }
     */
    private HashMap InvokeApi2252(ApiRequest2252DTO dto) throws
Exception {
        Request request = new Request();
        request.setMethod("GET");
        request.setAppKey("15810204");
        request.setAppSecret("******************************");
        request.setHost("http://0e5e6cd70******5e64****hai.a***pi.
com");
        request.setPath("/test");
        for (Field f : dto.getClass().getDeclaredFields()) {
            try{
                if(f.get(dto)!= null) {
                    request.getBodys().put(f.getName(), f.get(dto
).toString());
                }
            }catch(Exception e){}
        }
        request.setApiProtocol(ApiProtocol.HTTP);
        return dataApiClient.dataLoad(request);
    }
```

```
    /**
     * Response:
     */
    @RequestMapping(value = "/sample/test2252", method =
RequestMethod.POST)
    public HashMap testApi(@RequestBody ApiRequest2252DTO dto)
throws Exception {
        return InvokeApi2252(dto);
    }
}
/**
 * Request
 */
class ApiRequest2252DTO {
    public Integer pageNum;
    public Integer pageSize;
    public Integer id;
    public String name;
    public String sex;
    public Integer age;
}
```

说明:

您可以参考生成的代码样例，也可以直接单击保存，将代码添加到当前代码目录的dataservice包中。

- **在可视化搭建中使用数据服务。**

  可视化搭建的组件和数据服务接口进行了深度的融合，数据服务的返回数据的默认格式，即为可视化组件接收数据的格式。可以实现即配即用，详情请参见可视化搭建。

13.4.8.2 DataOS API

本文将为您介绍DataOS API的功能、输入、输出等详情，以及如何进行配置使用。

CheckMetaTable

- **功能：判断table是否存在。**
- **输入：tableGuid（必选）。**
- **格式：`odps.<project>.<table>`。**
- **输出：true/false。**
- **示例如下：**
  - **输入：`request.setTableGuid("odps.autotest.daily_test");`**
  - **输出：`{"requestId":"0b85c9d915548770462378104e","errMsg":"success","errCode":0,"data":true}`**

GetMetaDB

- **功能：获取MaxCompute项目的信息。**

- 输入：项目GUID（必选）。
- 格式：`odps.<project>`。
- 输出：项目详情。

字段	描述
appGuid	项目唯一标识。
project	项目英文名称。
projectNameCn	项目名称。
comment	备注。
ownerId	owner的id。
createTime	创建时间。
modifyTime	修改时间。

- 示例如下：
  - 输入：`request.setDbGuid("odps.autotest");`
  - 输出：

```
{
    "requestId": "0bfaefec****61500671805e",
    "errMsg": "success",
    "errCode": 0,
    "data": {
        "appGuid": "odps.meta",
        "projectName": "meta",
        "projectNameCn": "ODPS元仓",
        "comment": "",
        "ownerId": "13101879118",
        "createTime": "2014-02-18",
        "modifyTime": "2018-04-16"
  }
}
```

GetMetaTable

- 功能：获取MaxCompute表的信息。
- 输入：tableGuid（必选）。
- 格式：`odps.<project>.<table>`。
- 输出：表的详情。

字段	描述
appGuid	项目唯一标识。
tableGuid	表唯一标识。

字段	描述
tableName	表名称。
id	数据库ID 。
ownerId	owner的ID 。
hasPart	是否为分区表，1为分区表，0为非分区表。
dataSize	表数据的大小。
createTime	表的创建时间。
lastDdlTime	表DDL最后的更新时间。
lastModifyTime	表最后的修改时间。

- 示例如下：
  - 输入：`request.setTableGuid(tableGuid);`
  - 输出：

```
{
    "requestId": "0b8906da****8175861e",
    "errMsg": "success",
    "errCode": 0,
    "data": {
        "appGuid": "odps.meta",
        "tableGuid": "odps.meta.m_table",
        "tableName": "m_table",
        "id": 64809,
        "OwnerId": "dp-base-odps@aliyun-test.com",
        "hasPart": 1,
        "dataSize": 49397610904693,
        "createTime": "2014-12-10 21:20:23",
        "lastDdlTime": "2017-04-18 10:10:06",
        "lastModifyTime": "2019-04-09 20:24:08"
  }
}
```

ListMetaTableColumn

- 功能：获取MaxCompute的列信息。
- 输入：tableGuid（必选）。
- 格式：`odps.<project>.<table>`。
- 输出：列详情。

字段	描述
appGuid	项目唯一标识。
tableGuid	表唯一标识。
tableName	表名称。

字段	描述
columnGuid	列唯一标识，格式为odps.<project>.<table>.<col>。
columnName	列名。
columnType	列类型。
seqNumber	列编号（从1开始）。
isPartitionCol	是否为分区列： 0为非分区列，1为分区列。
comment	备注。
safeLevel	安全等级。

- 示例如下：
  - 输入：`request.setTableGuid(tableGuid);`
  - 输出：

```
{
    "requestId": "0b8906d*****9796824e",
    "errCode": 0,
    "errMsg": "success",
    "columnList": [{
        "appGuid": "odps.meta",
        "tableGuid": "odps.meta.m_table",
        "tableName": "m_table",
        "columnGuid": "odps.meta.m_table.project_name",
        "columnName": "project_name",
        "columnType": "string",
        "seqNumber": 1,
        "isPartitionCol": 0,
        "comment": "Project名称",
        "safeLevel": "C2"
  },
{
    "appGuid": "odps.meta",
    "tableGuid": "odps.meta.m_table",
    "tableName": "m_table",
    "columnGuid": "odps.meta.m_table.name",
    "columnName": "name",
    "columnType": "string",
    "seqNumber": 2,
    "isPartitionCol": 0,
    "isPrimaryKey": 0,
    "isNullable": 0,
    "comment": "表名",
    "safeLevel": "C2"
  } ... ]
}
```

ListMetaTablePartition

- 功能：获取MaxCompute的分区信息。

- 输入：

参数	说明
tableGuid	格式为odps.<project>.<table>。
pageNum	页码。
pageSize	每页最多显示记录数。

- 输出：表分区的详情。

表 13-1:

字段	描述
appGuid	项目唯一标识。
tableGuid	表唯一标识。
tableName	表名称。
partitionGuid	分区唯一标识，格式为odps.<project>.<table>.<partition>。
partitionName	分区名称。
createTime	分区的创建时间。
modifyTime	分区的修改时间。
dataSize	分区的数据大小。
records	分区的记录数。
pageNum	当前分页页码。
pageSize	当前分页大小。
totalNum	总记录数。

- 返回示例如下：

```
{
    "requestId": "0baf3e0*****5025570e",
    "errCode": 0,
    "errMsg": "success",
    "pageNum": 1,
    "pageSize": 10,
    "totalNum": 1101,
    "partitionList": [{
        "appGuid": "odps.meta",
        "tableGuid": "odps.meta.m_table",
        "tableName": "m_table",
        "id": 168504514,
        "partitionGuid": "odps.meta.m_table.ds\u003d20190408",
        "partitionName": "ds\u003d20190408",
        "createTime": "2019-04-08 13:59:52",
        "modifyTime": "2019-04-08 19:54:51",
```

```
        "dataSize": 273248012568,
        "records": 720503170
    } ... ]
}
```

SearchMetaTables

- 功能：模糊查找表。
- 输入：

参数	说明
keyword	表名称的关键字。
pageNum	页码。
pageSize	每页最多显示的记录数。

- 输出：

字段	描述
appGuid	项目唯一标识。
tableGuid	表唯一标识。
tableName	表名称。
ownerId	owner的ID。
createTime	表的创建时间。
lastDdlTime	表DDL最后的更新时间。
lastModifyTime	表最后的修改时间。

- 示例如下：
  - 输入：`request.setKeyword("test");`
  - 输出：

```
{
    "message": null,
    "code": 200,
    "success": true,
    "data": {
        "requestId": "0be41b***22277597924e",
        "errCode": 0,
        "errMsg": "success",
        "pageNum": 1,
        "pageSize": 2,
        "totalNum": 5000,
    "data": [{
        "appGuid": null,
        "tableGuid": "odps.ant_p13n.finance_newsrec_tab_dataset_ds
",
        "tableName": "finance_newsrec_tab_dataset_ds",
        "createTime": "2018-07-06 16:24:41",
```

```
        "lastModifyTime": "2019-04-26 10:49:23",
        "lastDdlTime": null,
        "lastAccessTime": null,
        "ownerId": "163585"
    },
    {
        "appGuid": null,
        "tableGuid": "odps.tbcdm.dws_tm_itm_cate_food_ftr_test_cm
",
        "tableName": "dws_tm_itm_cate_food_ftr_test_cm",
        "createTime": "2017-11-23 17:06:18",
        "lastModifyTime": "2019-04-26 20:34:12",
        "lastDdlTime": null,
        "lastAccessTime": null,
        "ownerId": "108292"
    }]
  },
    "timestamp": 1556452227875,
    "sessionId": null
}
```

使用DataOS API

pom配置如下所示：

```
<!--DataOS Start-->
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-dataworks-enterprise-ultimate</
artifactId>
    <version>0.0.3</version>
</dependency>

<!-- 要求JSON 2.8.5以上版本 -->
<dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.4.0</version>
</dependency>
<!--DataOS End-->
```

host配置如下所示：

```
# from src/main/resources/application.properties
# dataos api configuration
dataworks.dataos.auth.accessId= <indicate user accessid, refer to
aliyun>
dataworks.dataos.auth.accessKey= <indicate user accessid, refer to
aliyun>
dataworks.dataos.region=cn-shanghai
dataworks.dataos.endpoint=dataworks-ee-ue-share.cn-shanghai.aliyuncs.
com
dataworks.dataos.product=dataworks-enterprise-ultimate
```

Java代码如下所示，其中创建IClientProfile时，需要指定云账号的AccessKeyID和AccessKeySecret，详情请参见下文的常见问题。

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.dataworks.model.v20171212.CheckMetaTableRequest;
```

```
import com.aliyuncs.dataworks.model.v20171212.CheckMetaTableResponse;
import com.aliyuncs.dataworks.model.v20171212.GetMetaDBRequest;
import com.aliyuncs.dataworks.model.v20171212.GetMetaDBResponse;
import com.aliyuncs.dataworks.model.v20171212.GetMetaTableRequest;
import com.aliyuncs.dataworks.model.v20171212.GetMetaTableResponse;
import com.aliyuncs.dataworks.model.v20171212.ListMetaTableColumnR
equest;
import com.aliyuncs.dataworks.model.v20171212.ListMetaTableColumnR
esponse;
import com.aliyuncs.dataworks.model.v20171212.ListMetaTablePartiti
onRequest;
import com.aliyuncs.dataworks.model.v20171212.ListMetaTablePartiti
onResponse;
import com.aliyuncs.dataworks.model.v20171212.SearchMetaTablesRequest;
import com.aliyuncs.dataworks.model.v20171212.SearchMetaTablesResponse
;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;
import com.google.gson.Gson;

public class Simple {
  IAcsClient client = null;

  @org.junit.Test
  public void testCheckMetaTable() throws ServerException, ClientExce
ption {
    String tableGuid = "odps.meta.m_table";

    CheckMetaTableRequest request = new CheckMetaTableRequest();
    request.setTableGuid(tableGuid);
    CheckMetaTableResponse response = client.getAcsResponse(request);
    System.out.println(new Gson().toJson(response));
  }

  @org.junit.Test
  public void testGetProject() throws ServerException, ClientException
 {
    String appGuid = "odps.meta";

    GetMetaDBRequest request = new GetMetaDBRequest();
    request.setDbGuid(appGuid);
    GetMetaDBResponse getMetaDBResponse = client.getAcsResponse(
request);
    System.out.println(new Gson().toJson(getMetaDBResponse));
  }

  @org.junit.Test
  public void testGetPartitions() throws ServerException, ClientExce
ption {
    String tableGuid = "odps.meta.m_table";

    ListMetaTablePartitionRequest request = new ListMetaTablePartiti
onRequest();
    request.setTableGuid(tableGuid);
    request.setPageNum(1);
    request.setPageSize(10);
    ListMetaTablePartitionResponse response = client.getAcsResponse(
request);
    System.out.println(new Gson().toJson(response));
  }

  @org.junit.Test
```

```
  public void testSearchTables() throws ServerException, ClientExce
ption {
    SearchMetaTablesRequest request = new SearchMetaTablesRequest();
    request.setKeyword("test");
    request.setPageNum(1);
    request.setPageSize(10);
    SearchMetaTablesResponse response = client.getAcsResponse(request
);
    System.out.println(new Gson().toJson(response));
  }

    @org.junit.Test
    public void testGetColumns() throws ServerException, ClientExce
ption {
        String tableGuid = "odps.meta.m_table";

        ListMetaTableColumnRequest request = new ListMetaTableColumnR
equest();
        request.setTableGuid(tableGuid);
        ListMetaTableColumnResponse response = client.getAcsResponse(
request);
        System.out.println(new Gson().toJson(response));
    }

  @org.junit.Test
  public void testGetTable() throws ServerException, ClientException {
    String tableGuid = "odps.meta.m_table";

    GetMetaTableRequest request = new GetMetaTableRequest();
    request.setTableGuid(tableGuid);
    GetMetaTableResponse response = client.getAcsResponse(request);
    System.out.println(new Gson().toJson(response));
  }

  public Simple() throws ClientException {
    IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou",
 "<!!!!id>",
        "<!!!key>");
    DefaultProfile.addEndpoint("cn-hangzhou", "cn-hangzhou", "
dataworks", "dataworks-share.aliyuncs.com");
    client = new DefaultAcsClient(profile);
  }

}
```

常见问题

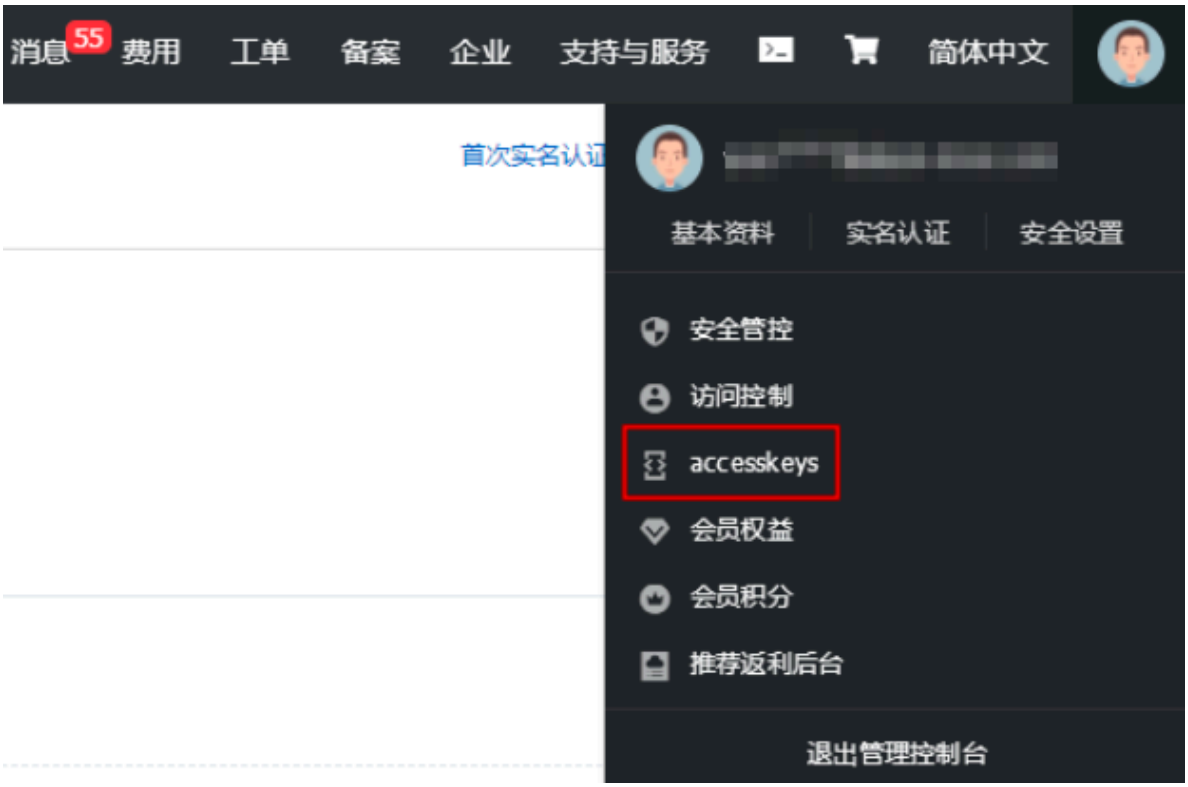
- **无法访问API，错误提示如下所示：**

  ```
  Exception in thread "main" com.aliyuncs.exceptions.ClientException:
  InvalidApi.NotFound : Specified api is not found, please check your
  url and method.
  RequestId : B081CCF1-9F19-473E-9B99-68F202E7572B
  ```

  错误原因：没有获取API权限。

- 如何查询AccessKeyID和AccessKeySecret?

  单击页面右上角账号下的accesskeys，即可进行查询。



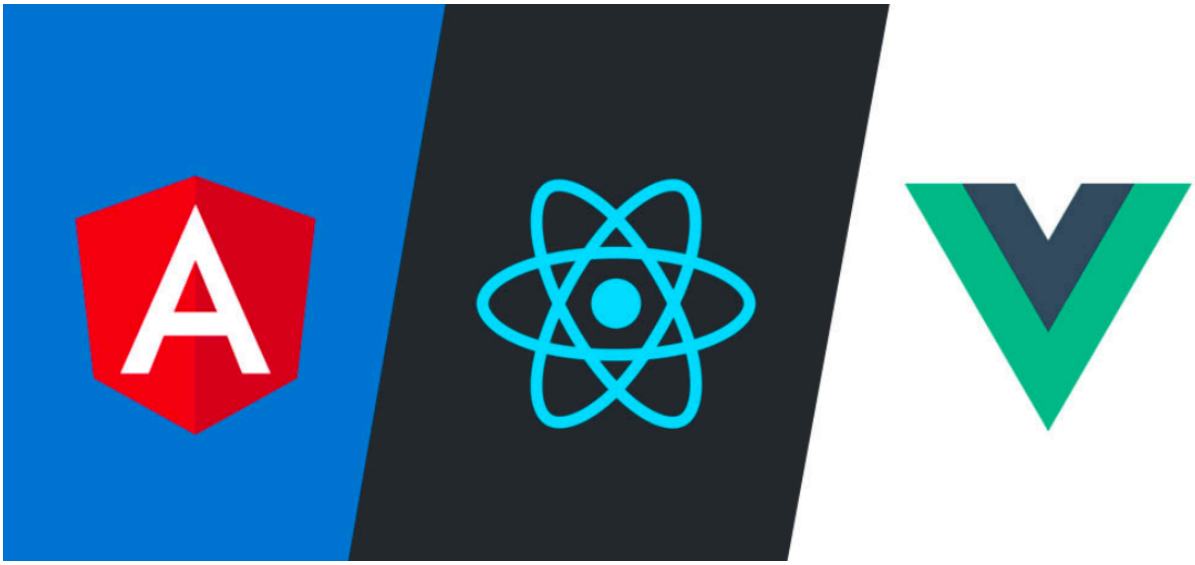
13.4.9 可视化搭建

13.4.9.1 可视化搭建概述

App Studio可视化搭建是辅助生成前端页面的工具，提供了一系列常见的网页组件，让开发者可以通过简单的拖拽，便可生成前端页面。本文将为您介绍App Studio可视化搭建的特点。

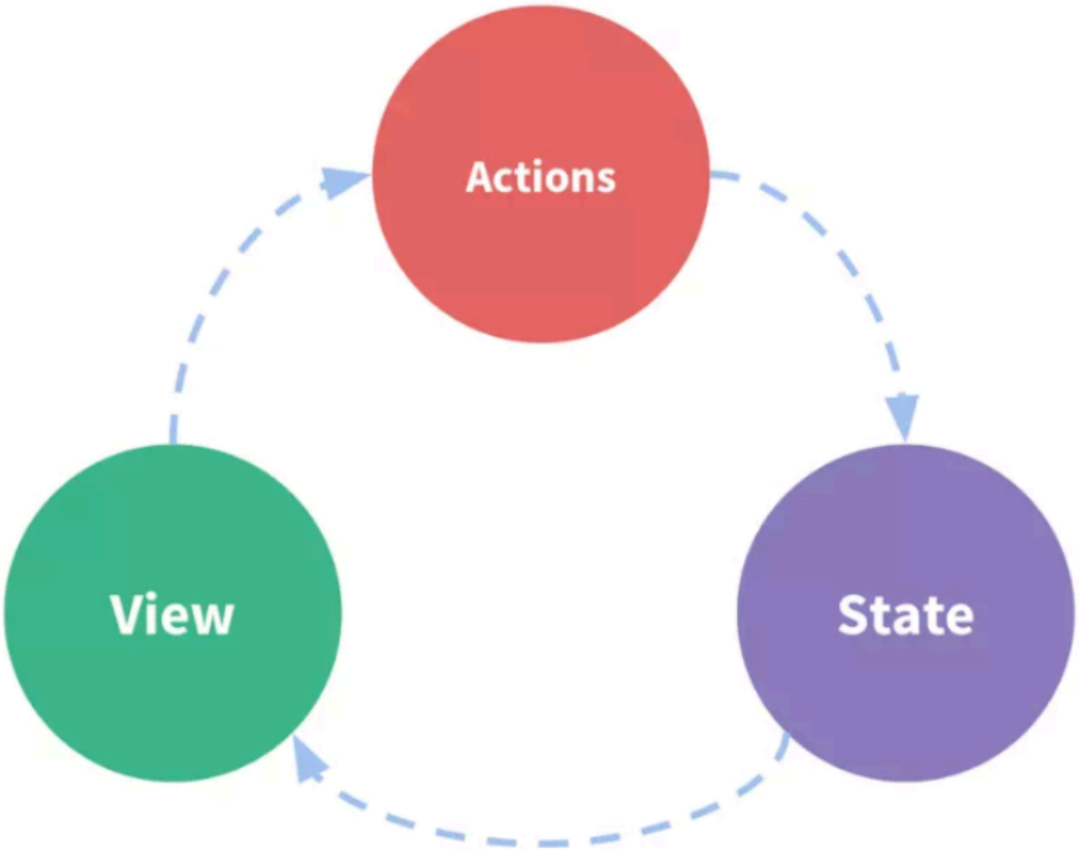
框架无感知

无论您使用React、Angular或Vue，App Studio可视化搭建系统都可以适配，因为底层使用了一种通用的描述语言来描述页面的结构、表现、行为等属性。



集成简单的数据处理来满足复杂交互需求

App Studio集成了一个全局的状态管理方案，来完成页面数据管理以及组件之间的交互。

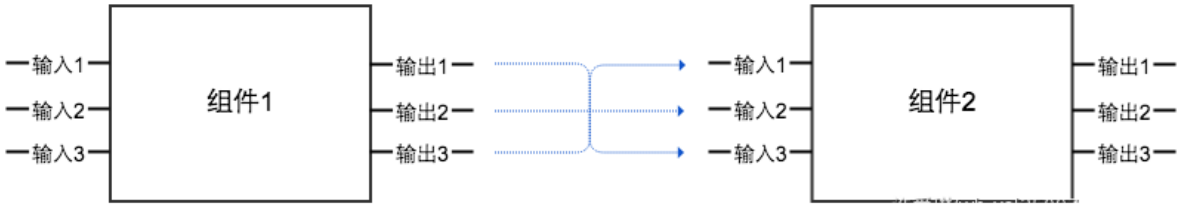


提供代码模式来满足复杂交互页面的搭建

App Studio可视化搭建的底层使用了通用的结构化可描述性语言（DSL）作为中间层。您可以直接基于DSL进行代码模式的修改，实现了代码模式与可视化拖拽模式的互转，对于高阶开发者来说这是一种进阶的使用体验。

可视化方式配置组件联动

App Studio可视化搭建提供了一种非常简单的可视化连线方式，来进行组件之间的交互联动。



无需构建即可直接发布运行

App Studio可以将中间层的DSL在线编译为一份可直接在浏览器中执行的代码，进行页面渲染。

对接DataWorks数据服务，快速集成数据接口

App Studio无缝对接DataWorks数据服务接口，可实时调试接口。

丰富的组件、模板市场

App Studio提供了丰富的组件，同时支持您自定义组件并上传到组件库。

另外，App Studio也提供了丰富的模板，您可以直接基于某一个模板快速生成页面，也可以将页面保存为模板并发布到模板市场供他人使用。

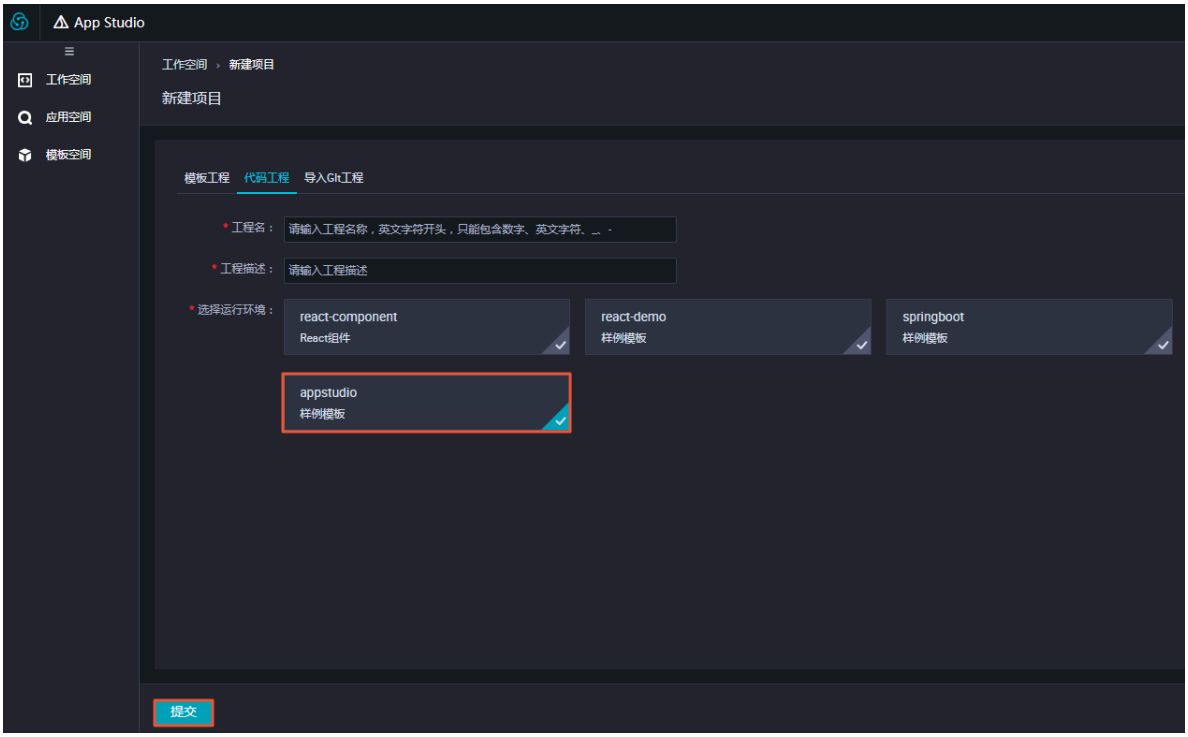
13.4.9.2 基本使用

本文将为您介绍可视化搭建系统的新建工程、可视化搭建等基本操作。

新建工程

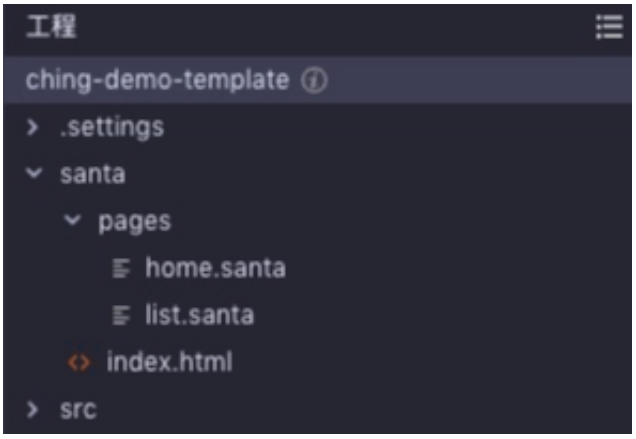
1. 进入App Studio页面，单击工作空间页面的通过代码创建工程。

2. 填写新建项目对话框中的工程名和工程描述，选择运行环境为appstudio样例模板。



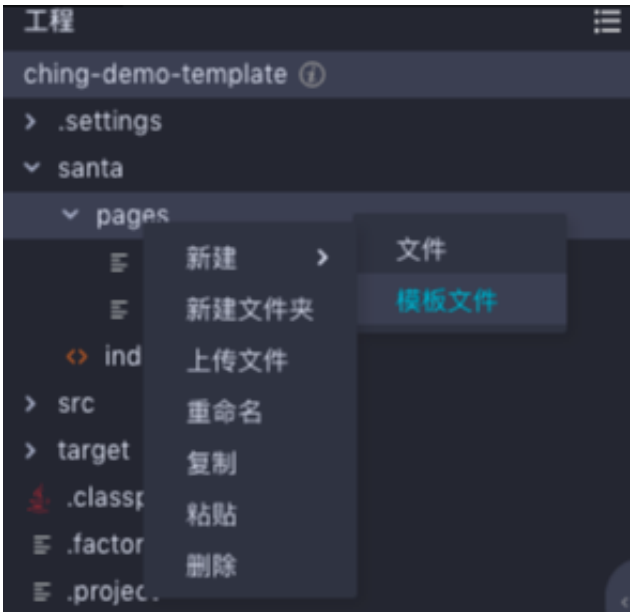
3. 配置完成后，单击提交。

4. 打开*santa/pages*目录。



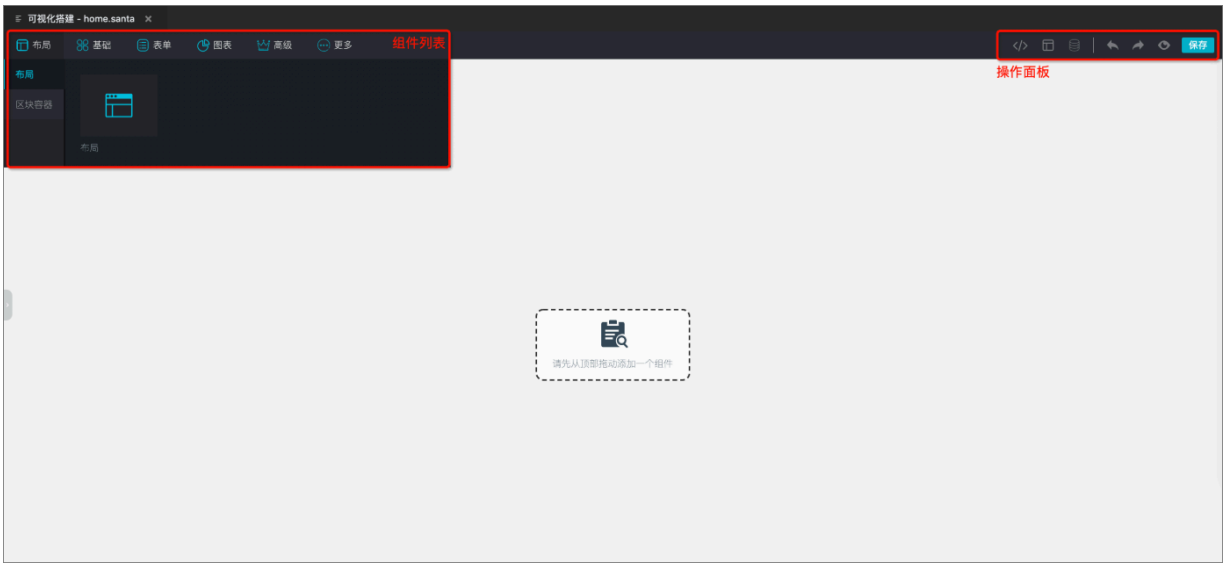
5. 单击任意一个.santa文件进入可视化搭建。

   您也可以右键单击pages，选择新建 > 模板文件，基于模板进行开发。



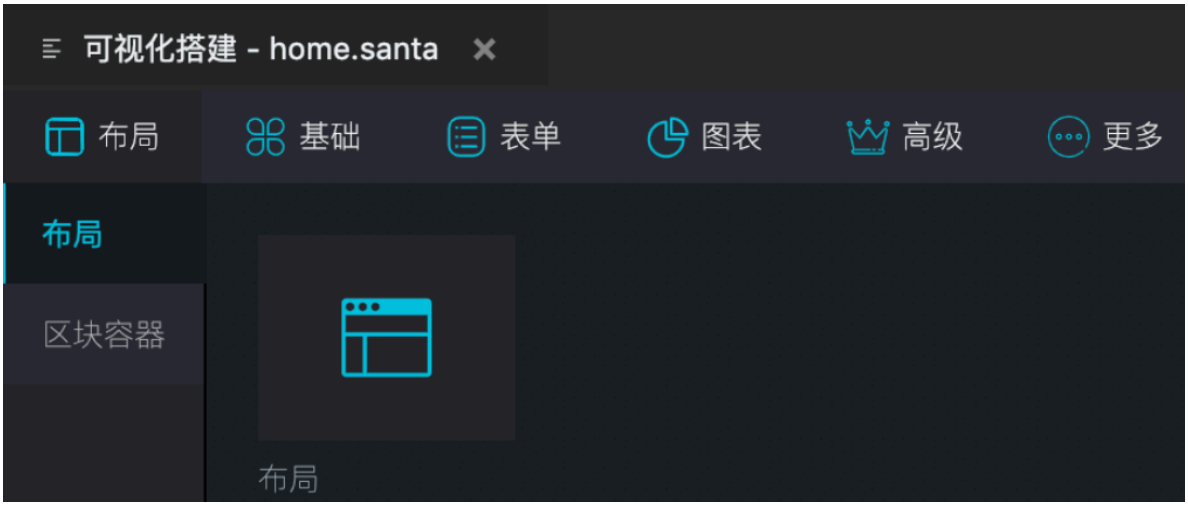
可视化搭建

可视化搭建页面主要由组件列表和操作面板组成。



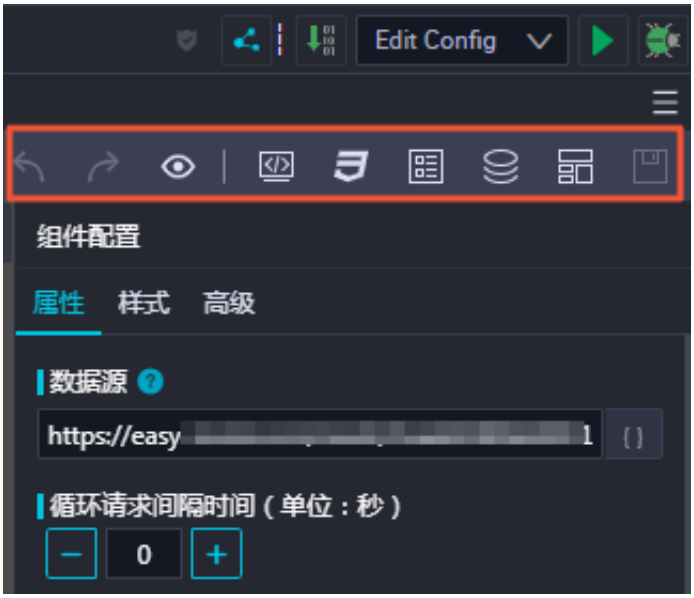
- 组件菜单

  组件菜单为您展示可视化搭建系统中，所有的系统预设组件，包括布局、基础、表单、图表、高级和更多等组件。



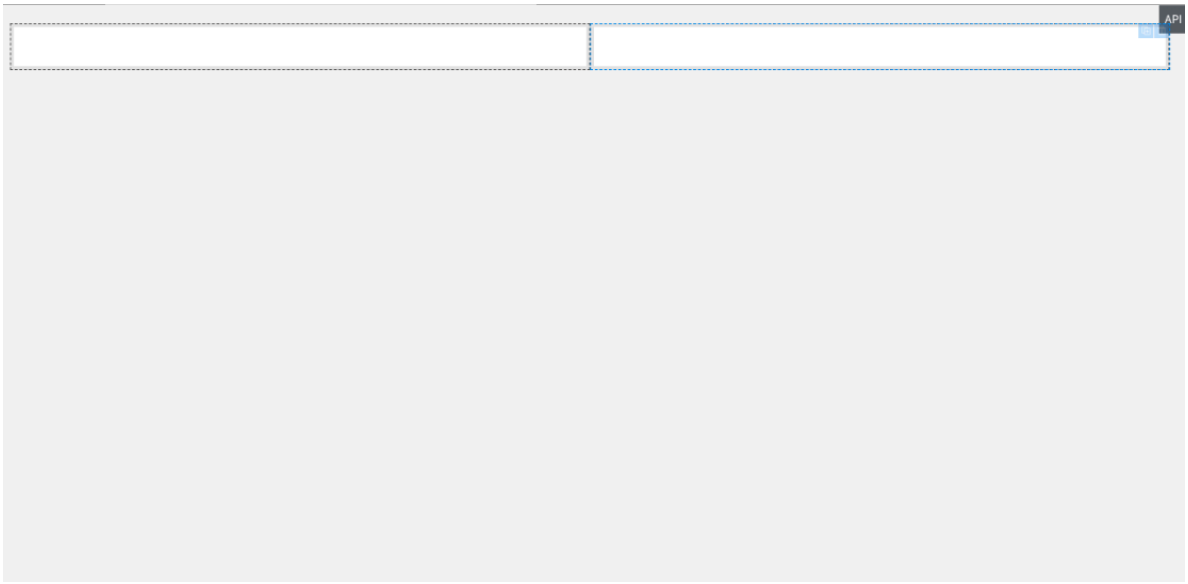
- 操作面板

  操作面板包括代码模式、导航配置、全局数据流配置、撤销、重做、预览和保存。



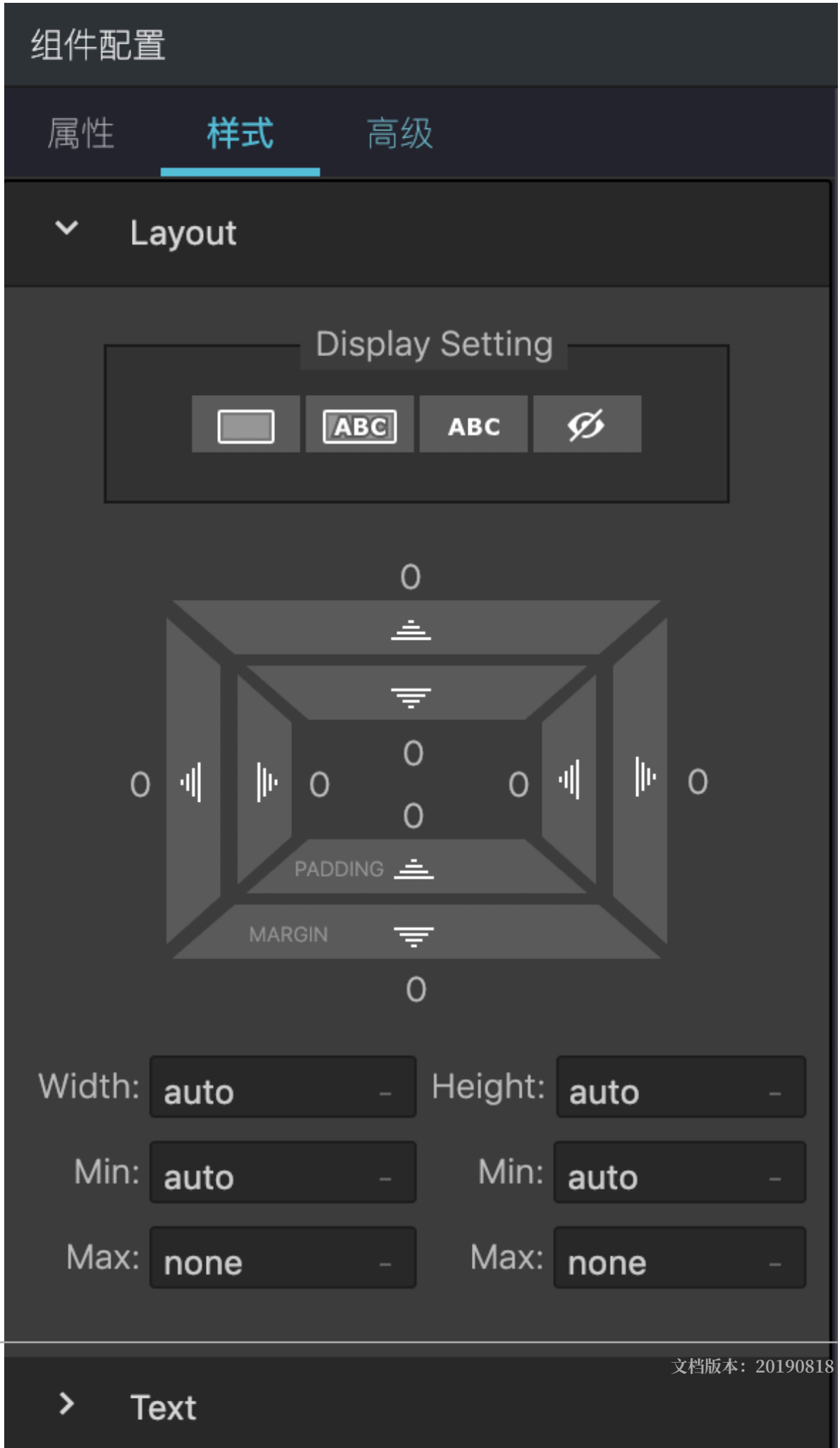
- **可视化操作面板**

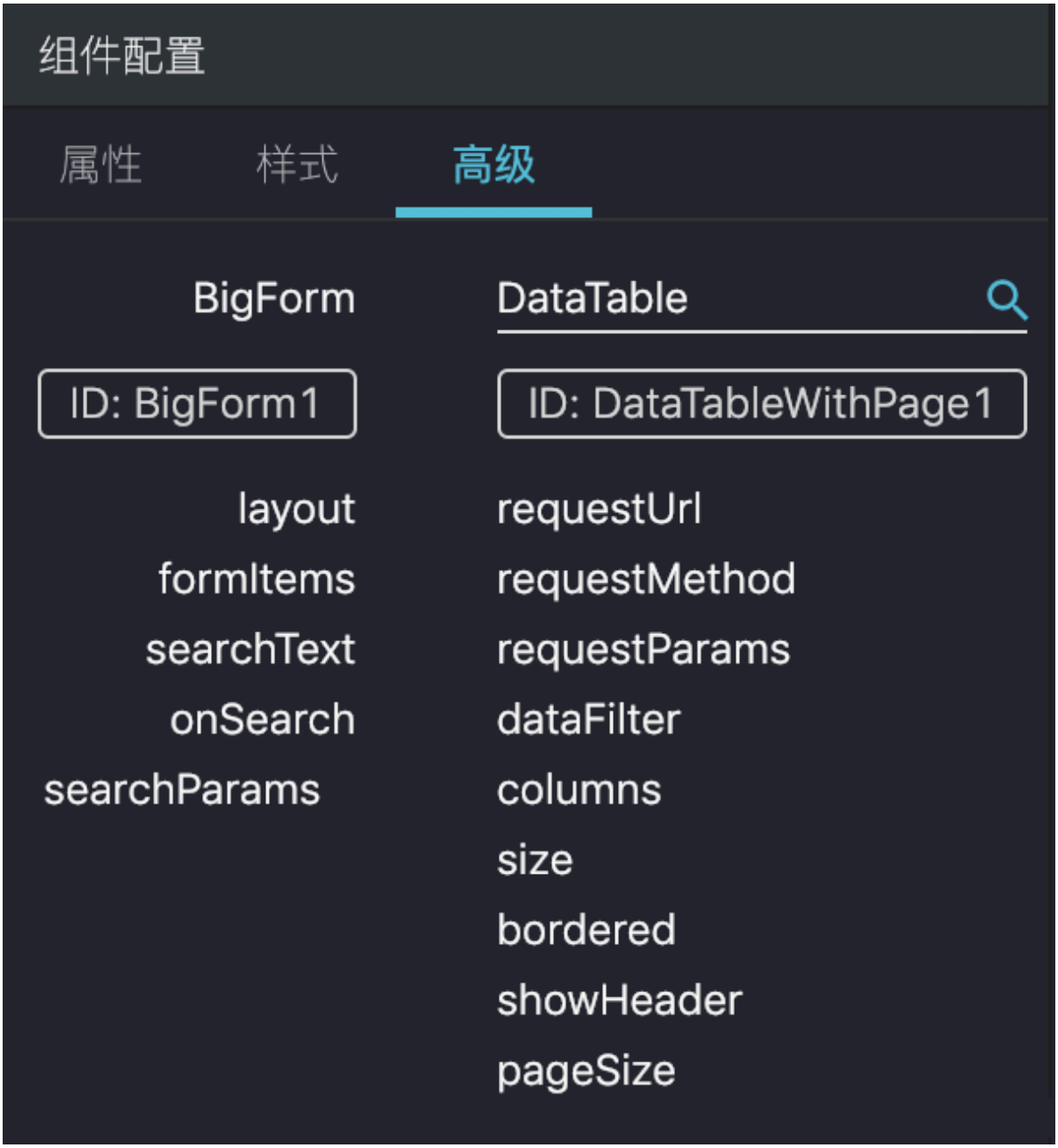
  展开组件菜单，单击某一个组件拖拽至可视化操作面板。



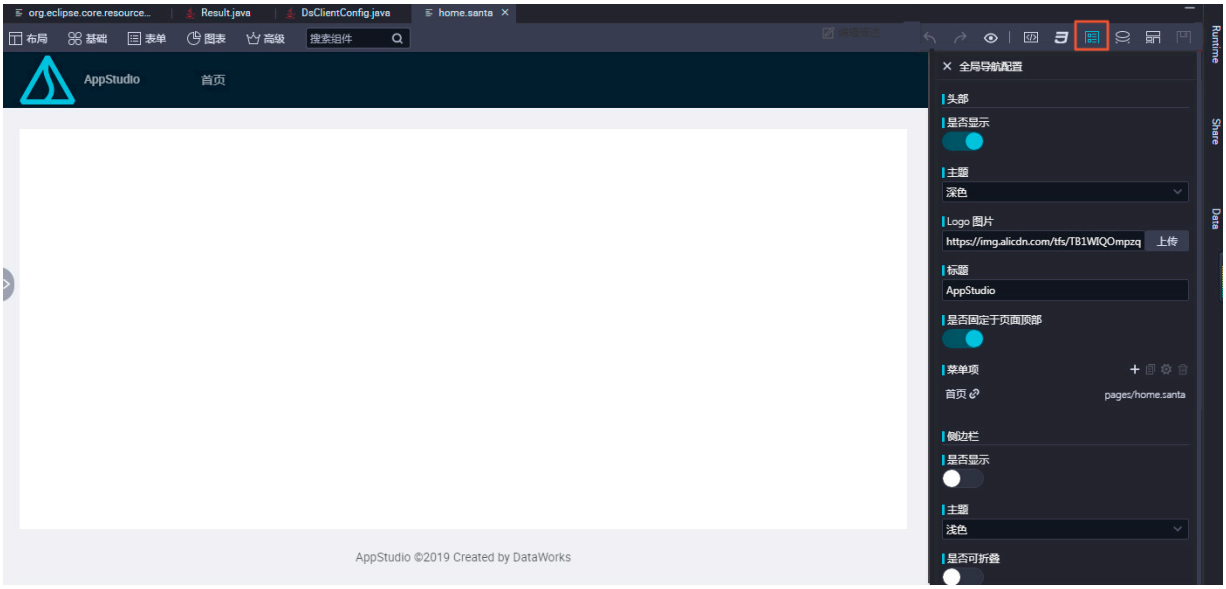
- **组件属性配置面板**

组件属性配置面板分为组件属性面板、组件样式面板和组件联动高级配置面板三部分。


组件配置
属性
样式
高级
Layout
Display Setting
ABC
ABC
0
0
0
0
0
PADDING
MARGIN
0
Width:
auto
Height:
auto
Min:
auto
Min:
auto
Max:
none
Max:
none
Text



单击操作面板中的导航配置按钮，即可打开应用导航配置页面。



导航配置中，您可以配置整个应用的公共头部、侧边栏以及应用菜单。

可视化搭建系统默认给应用添加公共头部、侧边栏，您可以打开导航配置进行自定义配置，例如隐藏侧边栏等。系统支持的配置如下所示：

- 头部支持如下配置：
  - Logo
  - 站点标题
  - 菜单项
  - 是否显示
  - 是否固定于页面顶部
  - 主题：黑色系/白色系
- 侧边栏支持如下配置：
  - 菜单项
  - 是否显示
  - 是否支持收起
  - 主题：黑色系/白色系

全局数据流配置

全局数据流配置详情请参见全局配置。

- 组件属性配置

  组件属性配置面板主要负责可视化的方式配置组件属性。

  根据组件的属性配置规则，组件属性配置面板将会生成一个可视化表单，让您输入组件的属性配置。在组件属性配置表单中更改组件属性后，可视化操作区域将会根据接收到的组件属性，进行重新渲染。您可以实时查看组件不同属性的渲染结果。

- 组件样式配置

  组件样式面板主要负责组件样式的相关设置。

  组件样式配置面板将会生成一个通用的样式配置可视化页面，您可以基于该面板定制组件基本的外观样式，包括布局、文字、背景、边框、效果等常用样式配置。

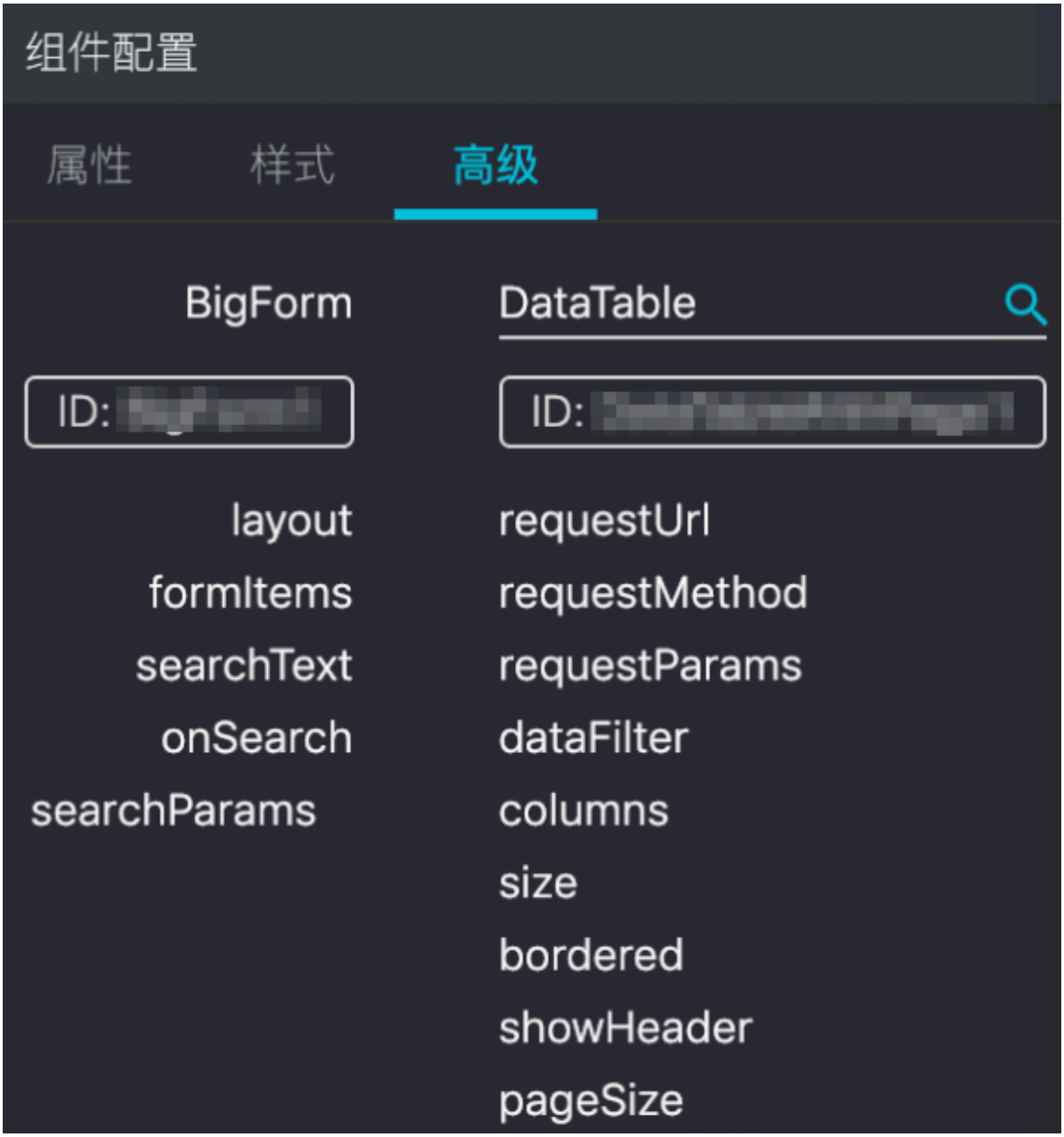
  在组件样式配置面板中添加、修改组件样式，可视化搭建系统将会收集所有的样式设置到组件上，可视化操作区域将会根据新的样式设置重新渲染对应组件，您可以实时查看配置后的组件效果。

- 组件联动高级配置

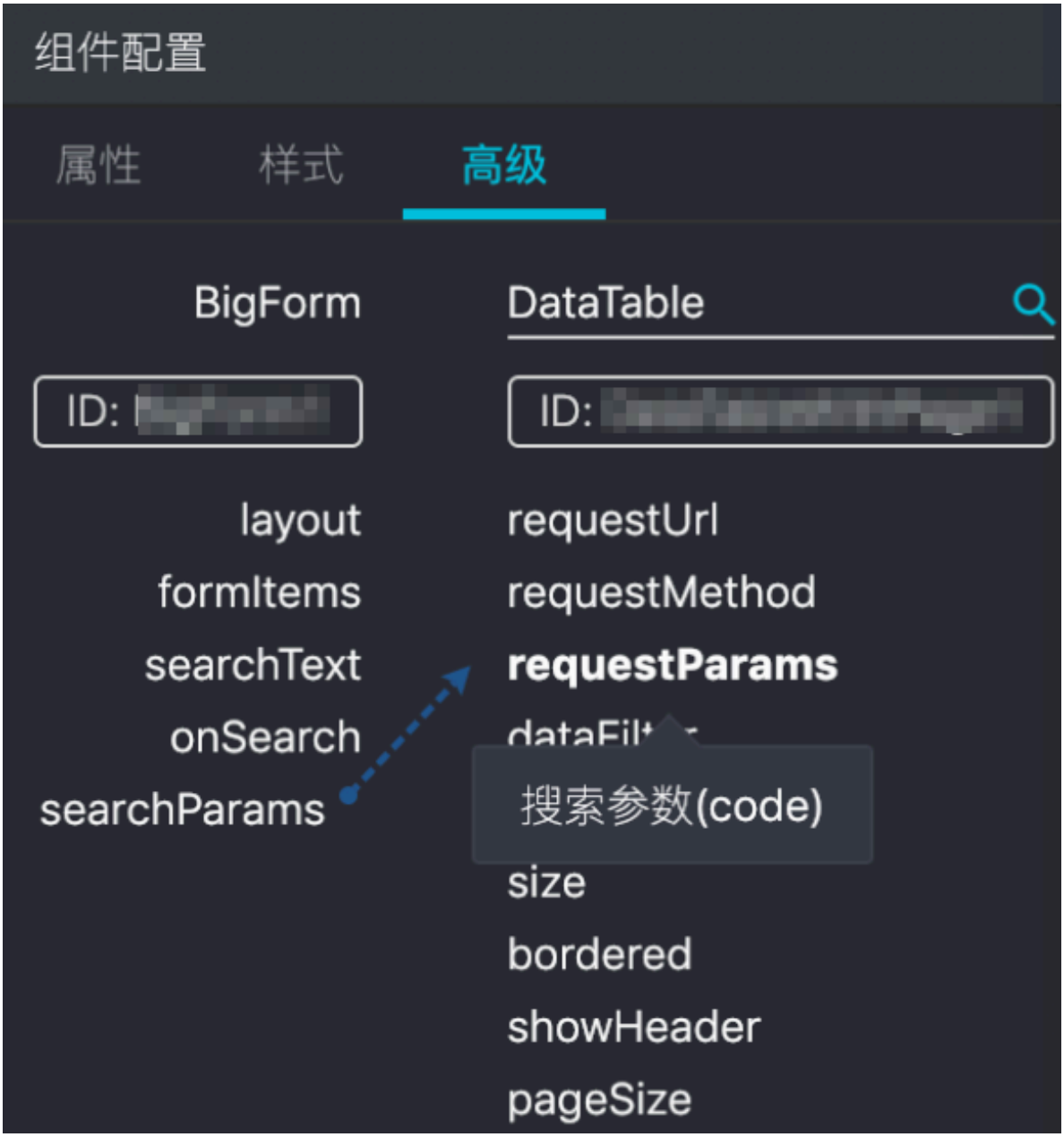
  组件联动高级设置面板主要负责组件之间的联动设置。

  单击可视化操作区域中的某一个组件，选中高级面板，高级设置面板中，将会在左侧列出当前选中组件对应的组件属性，单击右侧的放大镜按钮选择需要关联的另一个组件。

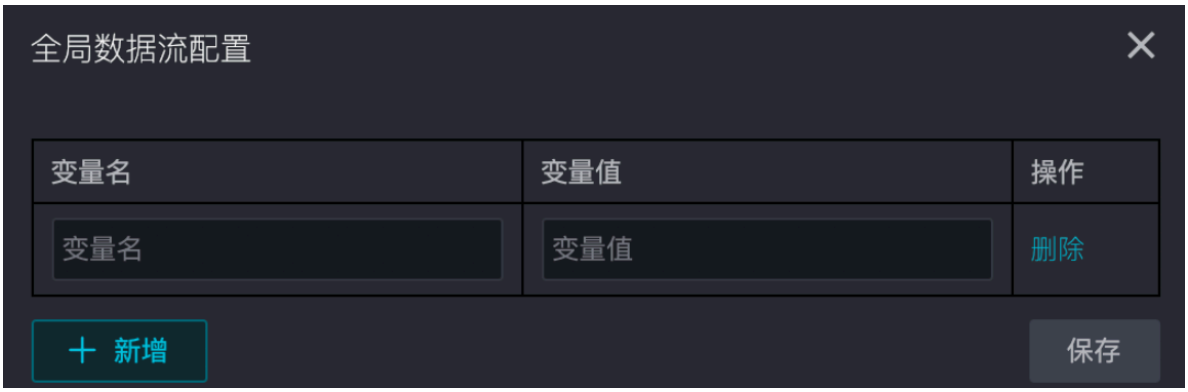
  选中需要关联的另一个组件后，高级设置面板右侧将会出现对应的组件属性。


组件配置
属性
样式
高级
BigForm
DataTable
ID:
ID:
layout
formItems
searchText
onSearch
searchParams
requestUrl
requestMethod
requestParams
dataFilter
columns
size
bordered
showHeader
pageSize

- 单击左侧属性列表中的某一个属性，连线至右侧属性列表中的另一个属性。



该操作将会实现两个组件之间的属性联动，左侧组件的searchParams参数变更将会及时传递到右侧组件的requestParams参数，从而实现两个组件基于属性之间的联动配置。



代码模式

代码模式提供了一种更高级的方式来满足更复杂的交互场景的需求，详情请参见#unique_675。

保存、预览、运行和热部署

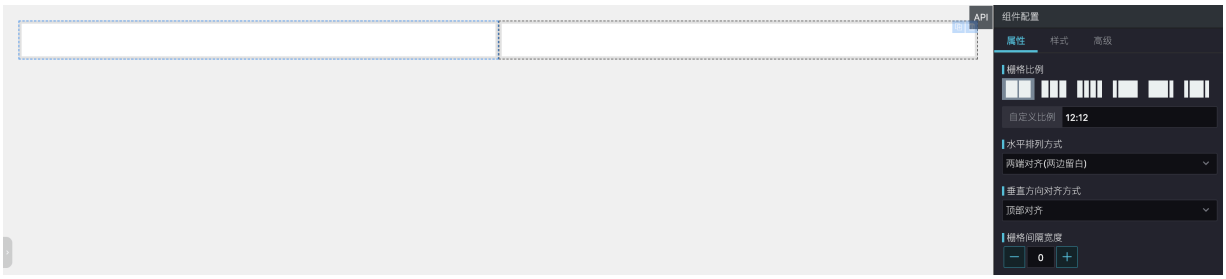
详情请参见保存、预览、运行和热部署。

13.4.9.3 常用组件

APP Studio可视化搭建系统自带80多个组件，可以满足您搭建基本页面的需求。本文将为您介绍可视化搭建系统默认自带的组件。

布局组件

布局组件为您提供一个24栅格系统组件。

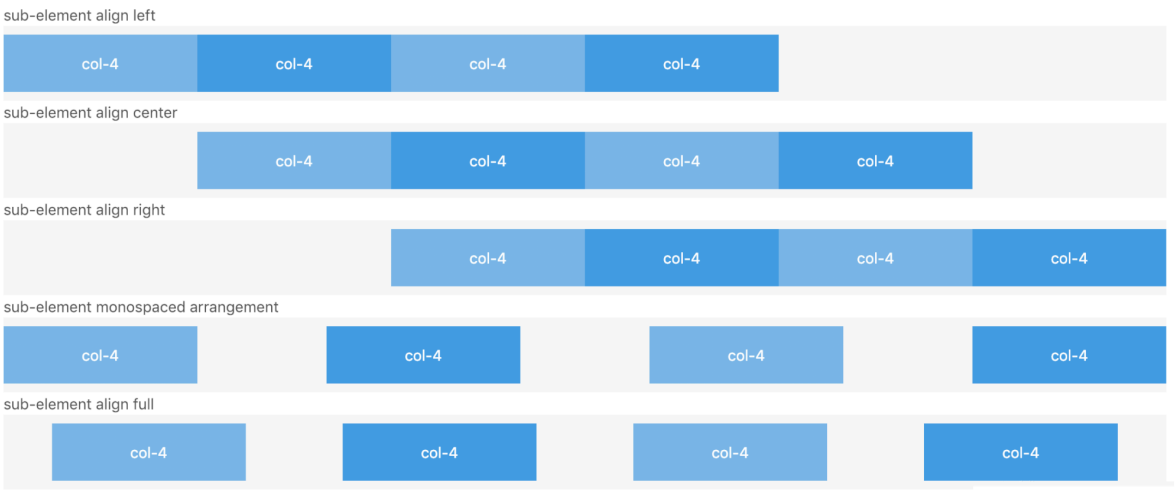


- 栅格比例

  系统默认将24栅格切割成一个12:12的栅格系统，您可以切换至其他常见的栅格比例，也可以自定义栅格比例。只需要保证所有栅格比例加起来是24的总数，布局组件将会根据各个栅格的比例进行布局切割。

- 水平排列方式

  水平排列方式定义了栅格在父节点中的排版方式。



- 垂直排列方式

  垂直排列方式定义了子元素垂直方向的对齐方式。

- 栅格间隔宽度

  栅格常常需要和间隔进行配合，您可以使用该配置来定义栅格间隔。

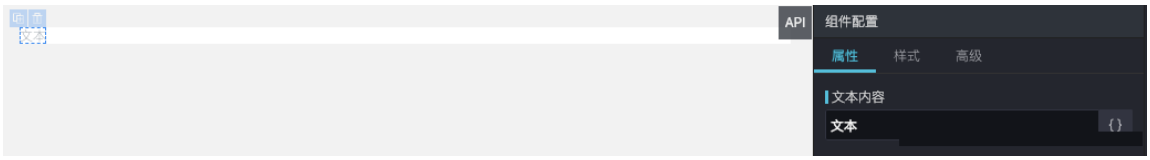
- 区块容器

  区块容器是一个块状的容器组件，区块容器组件可以作为一系列组件的父组件，类似于HTML中的div容器。

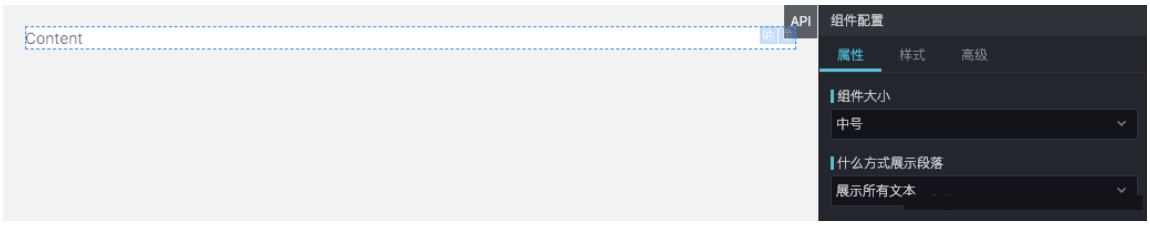
基础组件

基础组件均支持组件相关的常用属性设置。

- 文字
  - 文字



  - 段落



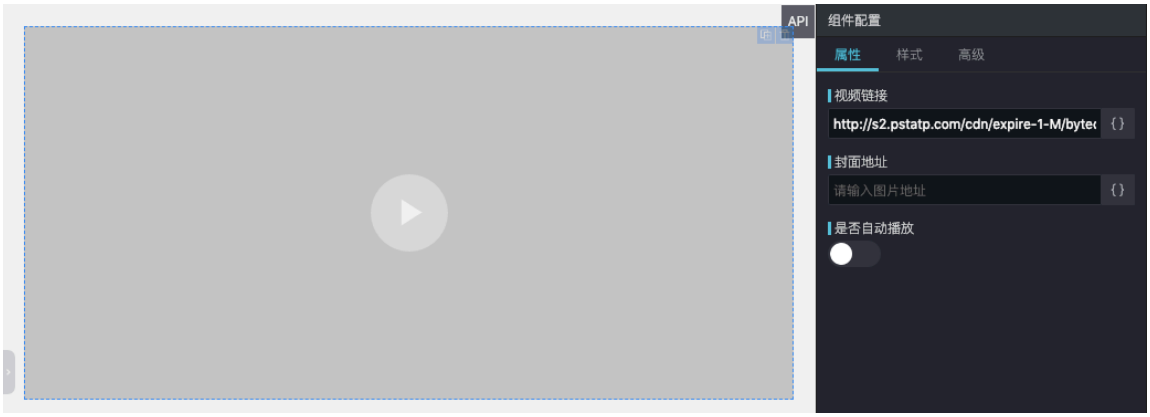
■ 组件大小

定义了段落文字大小。

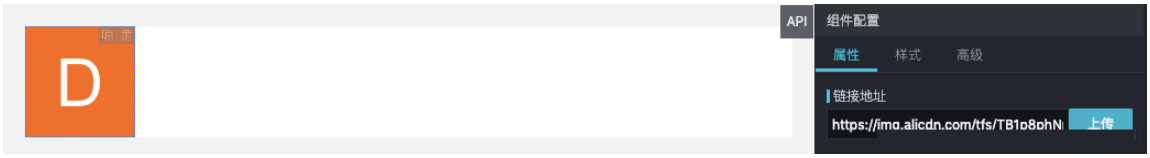
■ 什么方式展示段落

用于区分短文本和长文本，短文本的行间距会更小（通常三行以内）。

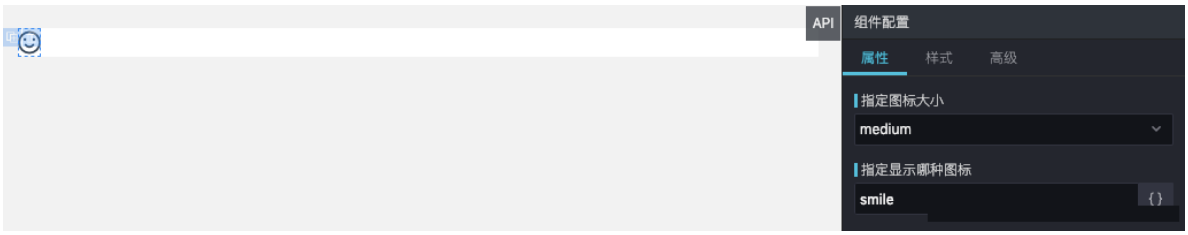
- 媒体
  - 视频



    - 视频链接：需要播放的视频地址。
    - 封面地址：视频封面图片地址。
    - 是否自动播放：是否在组件加载完之后自动播放视频。
  - 图片



    链接地址：显示的图片地址，可以上传图片。
- 图标

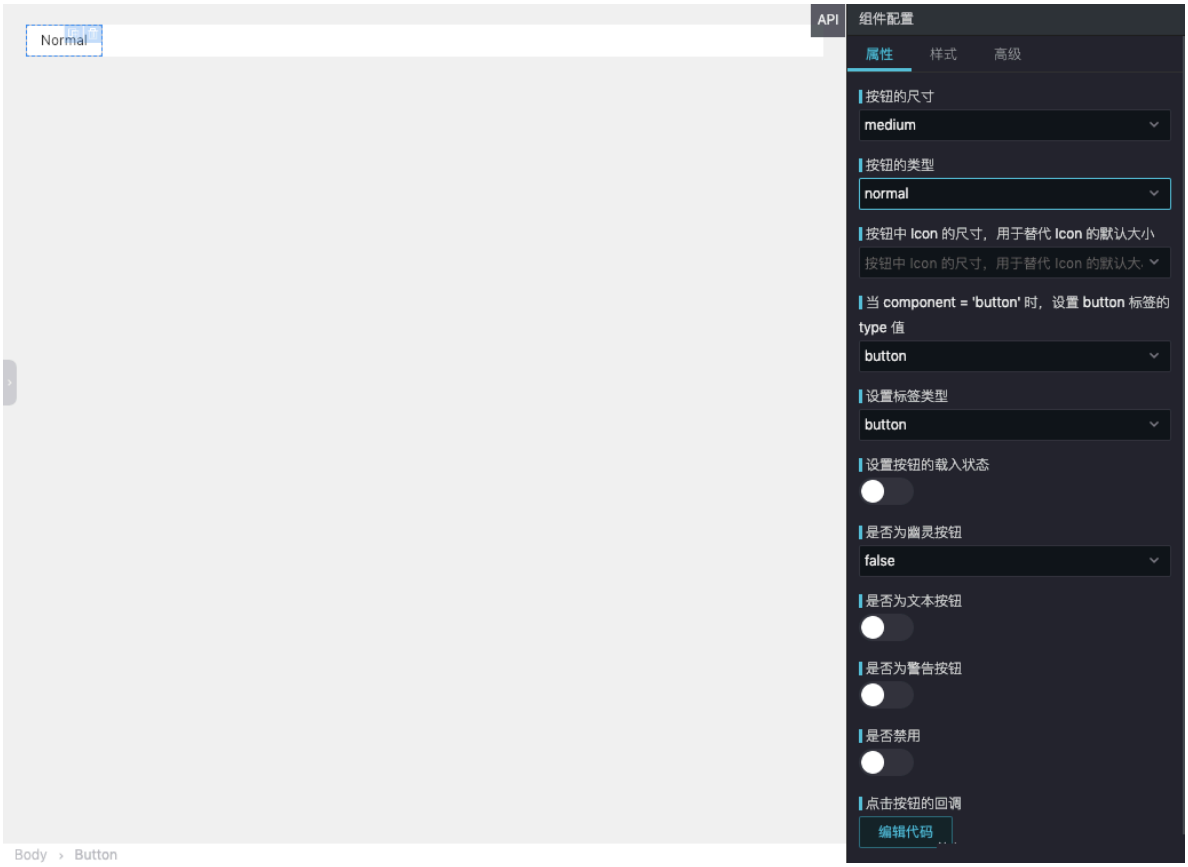


  - 指定图标大小

    指定图标的显示大小。
  - 指定显示哪种图标

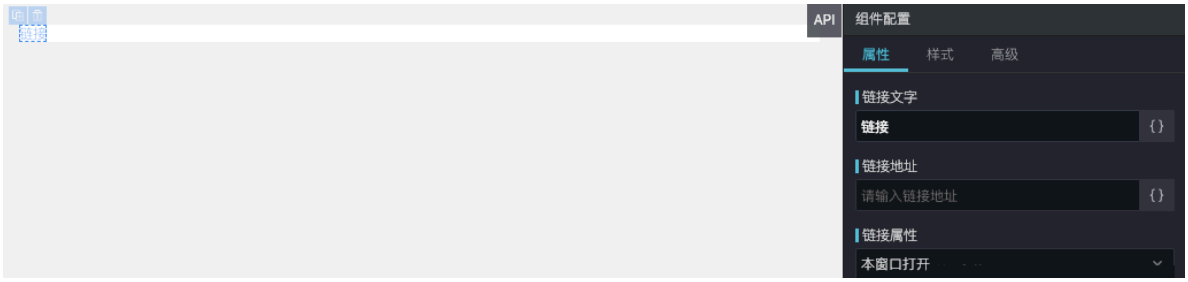
    指定图标的类型。

- 按钮



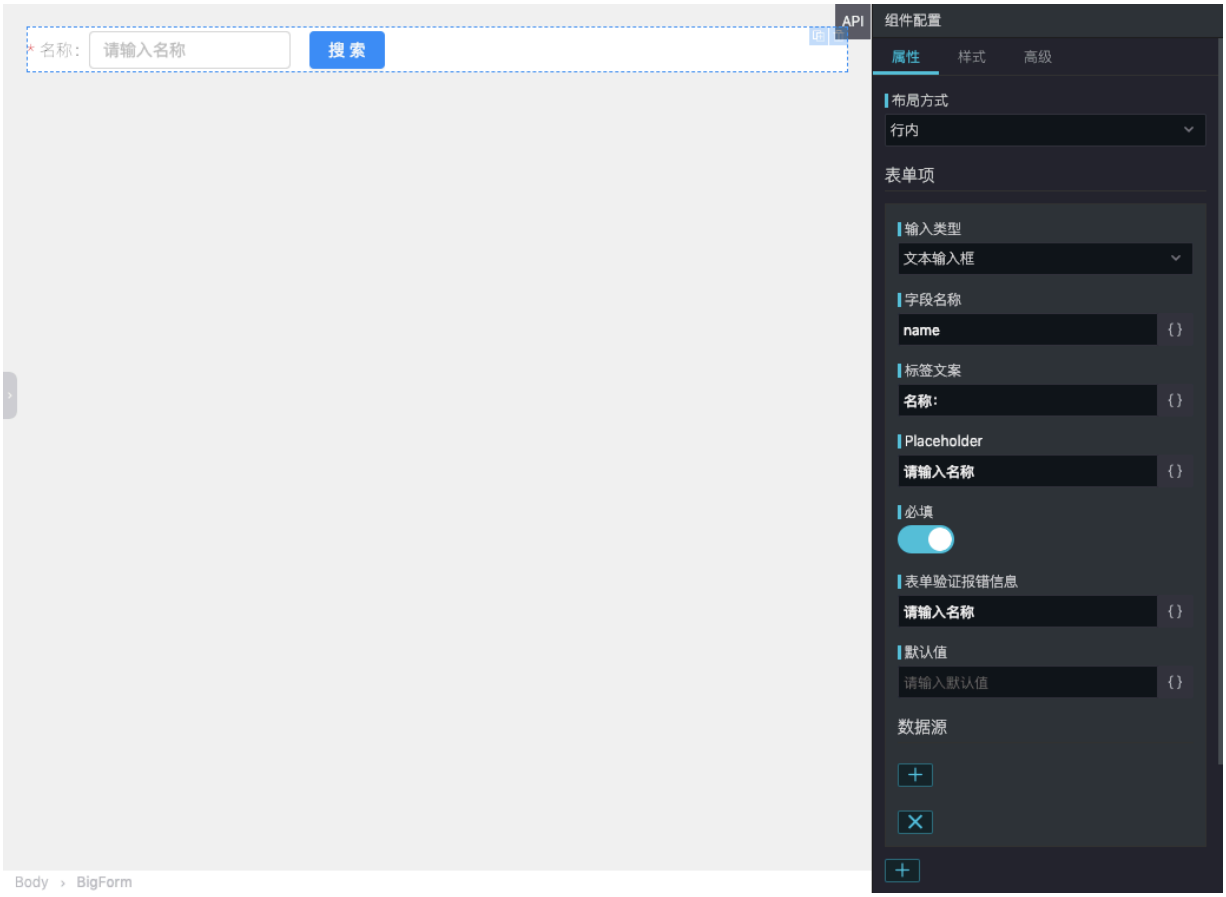
按钮属性的详情请参见按钮文档。

- 链接



  - 链接文字：显示的链接文字。
  - 链接地址：单击链接的跳转地址。
  - 链接属性：在本窗口打开和在新窗口打开。

表单组件



表单包括行内、水平和垂直三种布局方式。

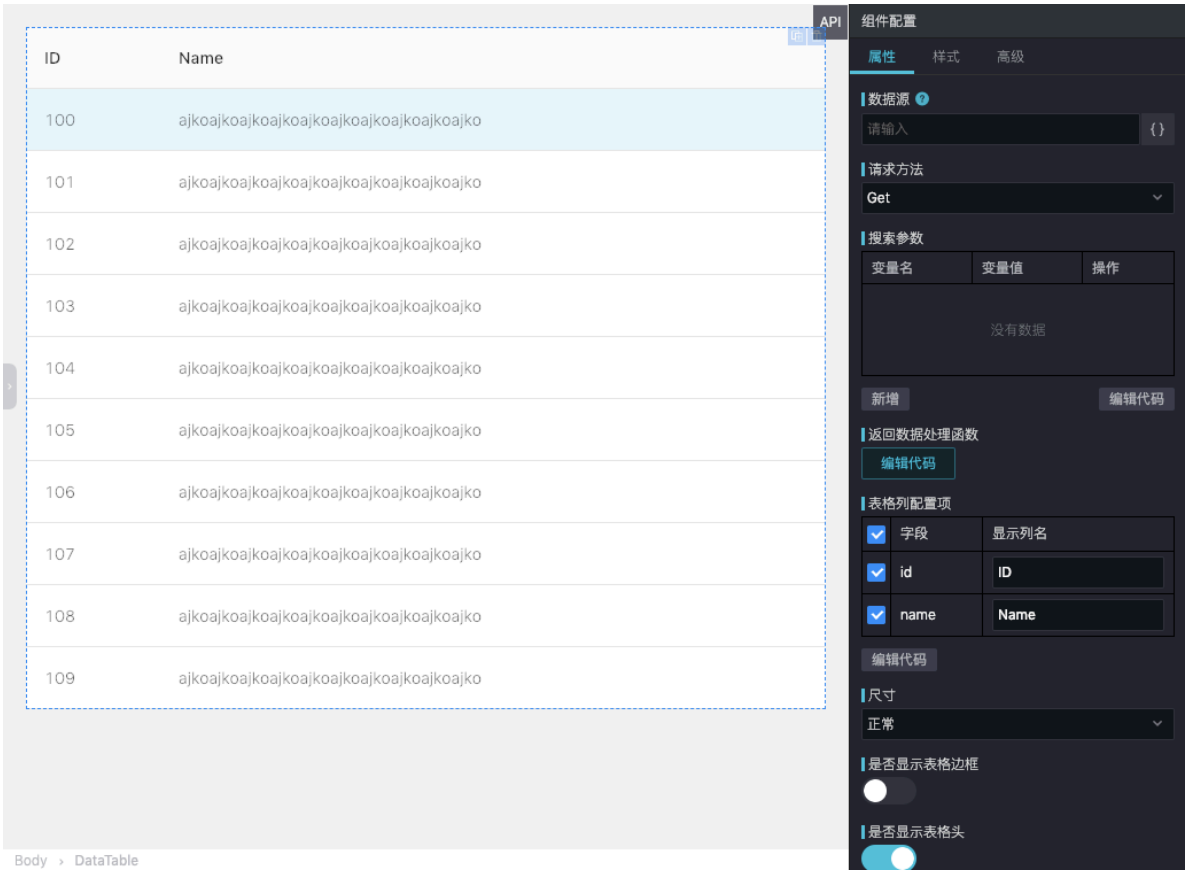
上传图片和附件详情请参见上传附件。

筛选详情请参见搜索。

输入框详情请参见输入框。

图表

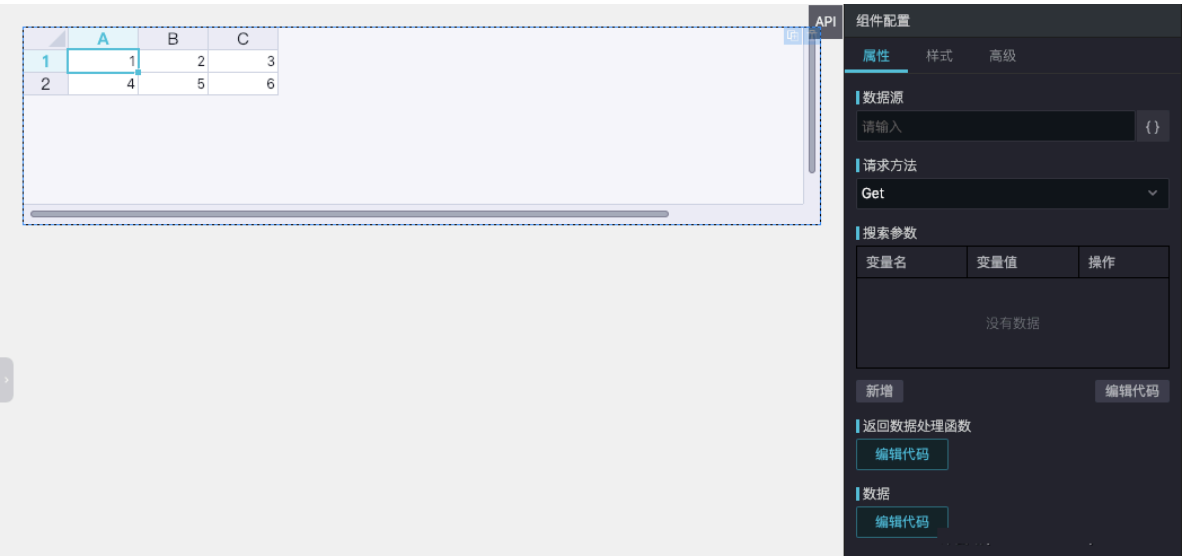
- 数据表格



配置	说明
数据源	请求接口地址。
请求方法	请求方法：Get/Post/Put/Delete。
搜索参数	接口请求参数。
返回数据处理函数	接口数据返回后的数据处理函数。
表格列配置项	定义表格需要显示的表格列。
尺寸	设置表格尺寸。
是否显示表格边框	设置是否显示表格边框。
是否显示表格头	设置是否显示表格头。

带分页的数据表格多了一项每页显示数量的配置项，定义分页中每一页的显示数量。

- Excel



配置	说明
数据源	请求接口地址。
请求方法	请求方法：Get/Post/Put/Delete。
搜索参数	接口请求参数。
返回数据处理函数	接口数据返回后的数据处理函数。
数据	直接配置Excel需要显示的数据。

· 折线图

配置	说明
数据源	请求接口地址。
请求方法	请求方法：Get/Post/Put/Delete。
搜索参数	接口请求参数。
返回数据处理函数	接口数据返回后的数据处理函数。
图表配置	通过代码对图表进行配置。
是否显示图表标题	设置是否显示图表标题。
图表标题	显示图表标题。
图表数据	直接配置图表需要显示的数据。
X轴字段	定义返回数据中显示到X轴的数据字段名。
Y轴字段	定义返回数据中显示到Y轴的数据字段名。

说明:

柱状图、条形图、面积图、饼图、地图、词云和散点图等图表组件的配置，请参见折线图。

高级组件

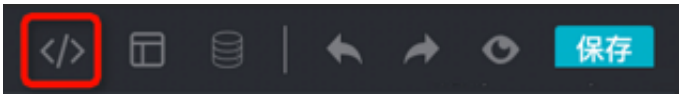
高级组件均支持组件相关的常用属性设置。

- 选择组件包括选择器、复选按钮、 级联选择、单选框、区段选择器、开关组件和评分。
- 交互：您可以通过Tab选项卡，在不同子任务、视图、模式之间切换，它具有全局导航的作用，是全局功能的主要展示和切换区域。详情请参见Tab选项卡。
- 轮播图：轮播组件以幻灯片的方式，在页面中横向展示诸多内容的组件。详情请参见图片轮播。
- 步骤条：默认情况下，Step定义为展示型组件。上层组件可以通过修改传入的current属性值来修改当前的步骤，同时可以设置每个节点的click事件，来自定义回调。详情请参见步骤。
- 进度条：进度指示器可以为您展示操作的当前进度。详情请参见进度指示器。
- 菜单：您可根据自身需求选择相应的菜单，详情请参见菜单。
- 导航：导航包括顶部导航和侧边导航。顶部导航提供全局性的类目和功能，侧边导航提供多级结构来收纳和排列网站架构。详情请参见导航。

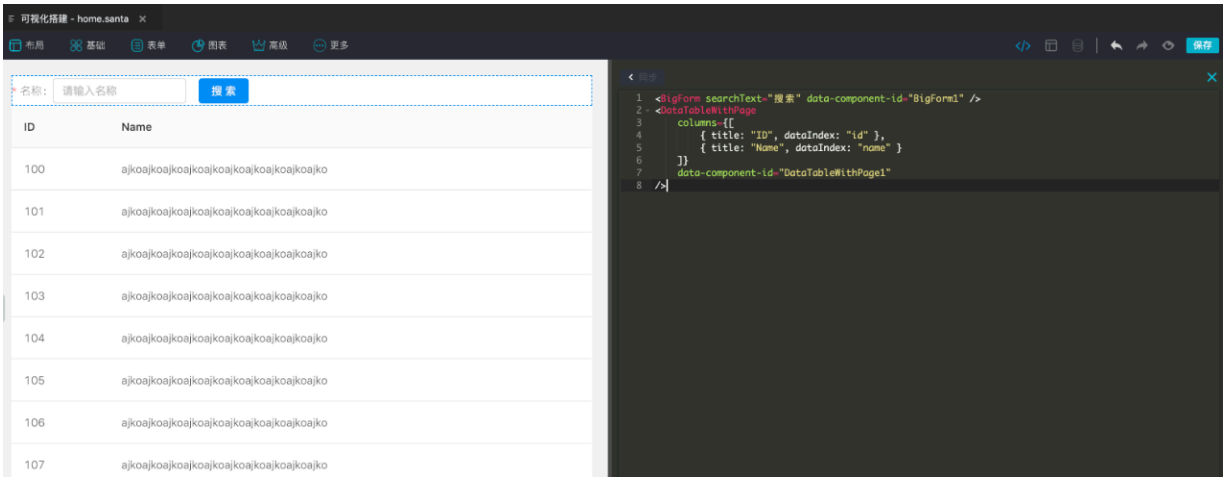
13.4.9.4 代码模式

代码模式提供了一种更高级的方式来满足更复杂的交互场景的需求。

单击操作面板中的代码模式图表，即可打开代码模式。



页面右侧将会出现代码区域。



可视化搭建使用DSL描述语言作为中间层的代码，基于该DSL进行可视化与代码模式的互转。可以简单地将DSL看作简化版的React，语法与React基本一致。

如上图的代码区域所示，DSL将一个组件使用标签进行描述，标签的属性就是组件的Props属性，属性值支持简单的数据类型，例如STRING或NUMBER。属性值也支持表达式，您可以直接输入`state.xxx`来获取全局数据流中的数据。

代码模式具有如下特点：

- 可视化区域中拖拽操作、组件属性配置等，会实时更新到代码。
- 代码中的修改会实时更新到可视化区域。
- 可视化拖拽操作、组件属性配置，与代码模式的修改可以互转。

13.4.9.5 DSL语法

DSL是一种以React JSX与Vue template的语言特性为基础，更符合UI编排的组件化语言。

JSX

DSL语法类似于React.render方法中的JSX部分，JSX的简单理解如下：

- 通过`{}`，将HTML作用域切换为JS作用域。JS作用域可以写任何合法的JS表达式，返回值会输出到页面上，例如`<div>{'Hello' + ' Relim'}</div>`。

  说明：

  `{ }`内可以写任何计算语句或字面量等JS表达式。

- 通过HTML标签，将JS作用域切换为HTML作用域，例如`{<div>Hello Relim</div>}`。
- HTML和JS作用域切换可以嵌套进行，例如`{<div>{'Hello' + ' Relim'}</div>}`。

JSX的更多详情请参见React JSX。

合法的JS表达式

```
//计算语句的情形
{aaa} // √   变量aaa需要有定义
{aaa * 111} // √
{1 == 1 ? 1 : 0} // √
{/^123/.test(aa)} // √
{[1,2,3].join('')} // √
{(()=>{return 1})()} //自执行函数 √

//字面量
{1}
{true}
{[11,22,33]} // √
{{aa:"11",bb:"22"}} // √
{()=>1} //描述一个函数，合法，但无意义 √
```

说明：

如果遇到较为复杂的逻辑，一条计算语句不能实现，需拆分为多条语句的需求。可以将其包装为自执行函数，自执行函数是合法的表达式。示例如下：

```
{(function(){
    //将一个数字数组的偶数为求和。
    var input = [1,2,3,4,5,6,7,8,9,10];
    var temp = input.filter(i => i % 2 == 0)
```

```
    return temp.reduce((buf, cur) => buf + cur, 0)
})()}
```

非法的JS表达式

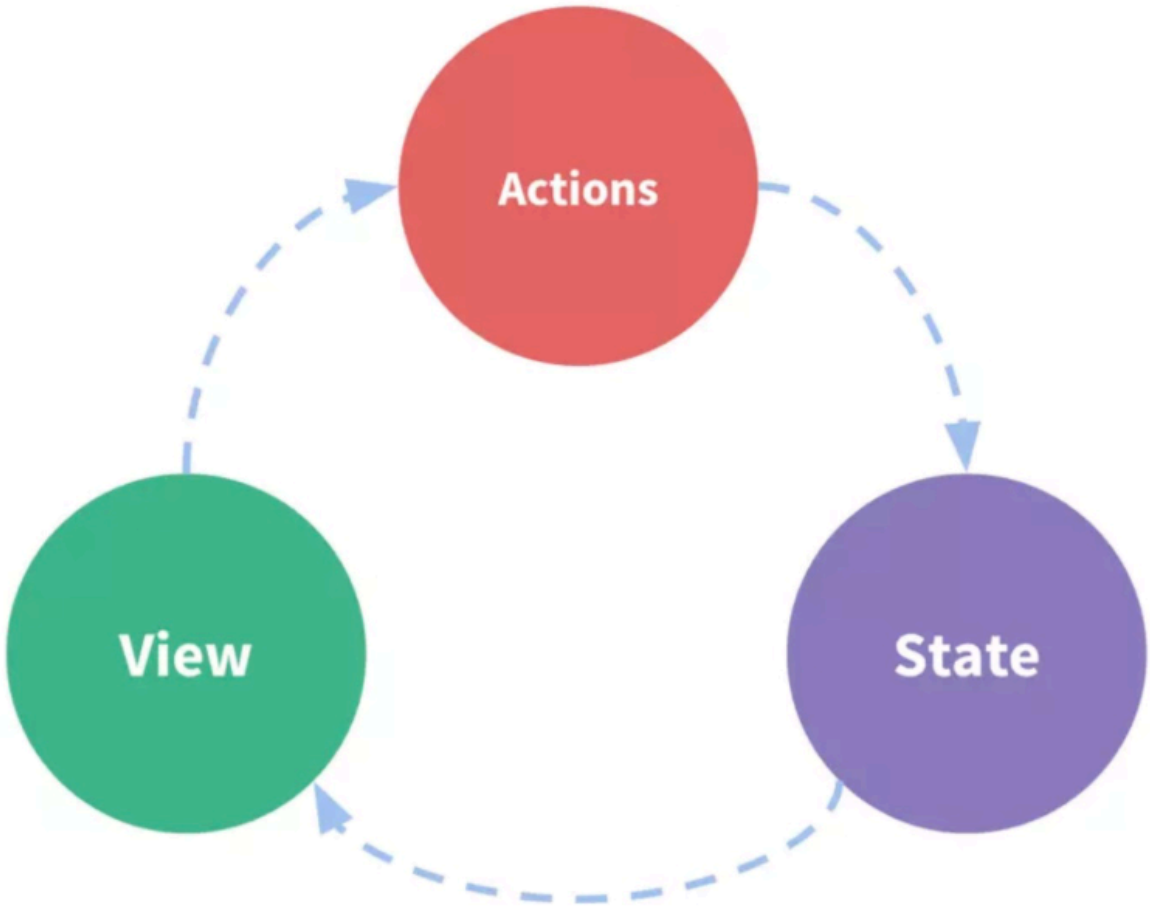
```
{ var a = 1 } // 赋值语句。
{ aaa * 111; 2} // 出现分号的多条语句。
```

13.4.9.6 全局数据流

全局数据流是前端数据管理的概念，多个组件为共享状态时，共享状态和组件间通信较为困难。此时将共享状态抽取出来，用全局数据流的方式使之变得简单。

全局数据流的原理

全局数据流使用了单一的数据流转方式，来实现全局数据的传递。在全局数据中声明的数据，只要变更后便会执行如下图所示的数据流转。



1. 组件触发一个Action（比如通过鼠标点击触发）。
2. Action触发全局数据变更。
3. 全局数据变更会自动触发引用了该全局状态的相关组件的重新渲染。

全局数据流的适用场景

全局数据流适用于页面中两个组件或者多个组件之间的组件联动，可以通过将公共数据提炼到全局数据中进行统一管理，再利用全局数据流机制串联两个或多个组件。

全局数据流的定义

1. 打开全局数据流弹框。

2. 输入变量名和变量值。

- 变量值可以为数字、字符串或JSON串。
- 变量值声明为一个接口地址，接口获取到的数据将会成为变量名对应的值。

全局数据流的使用

- 获取全局数据

  组件中通过`state.name`来获取全局数据。

  ```
  <Input value={state.name} />
  ```

- 修改全局数据

  组件中通过$setState方法修改全局数据。

  ```
  <Input onChange={value => $setState({ name: value })} />
  ```

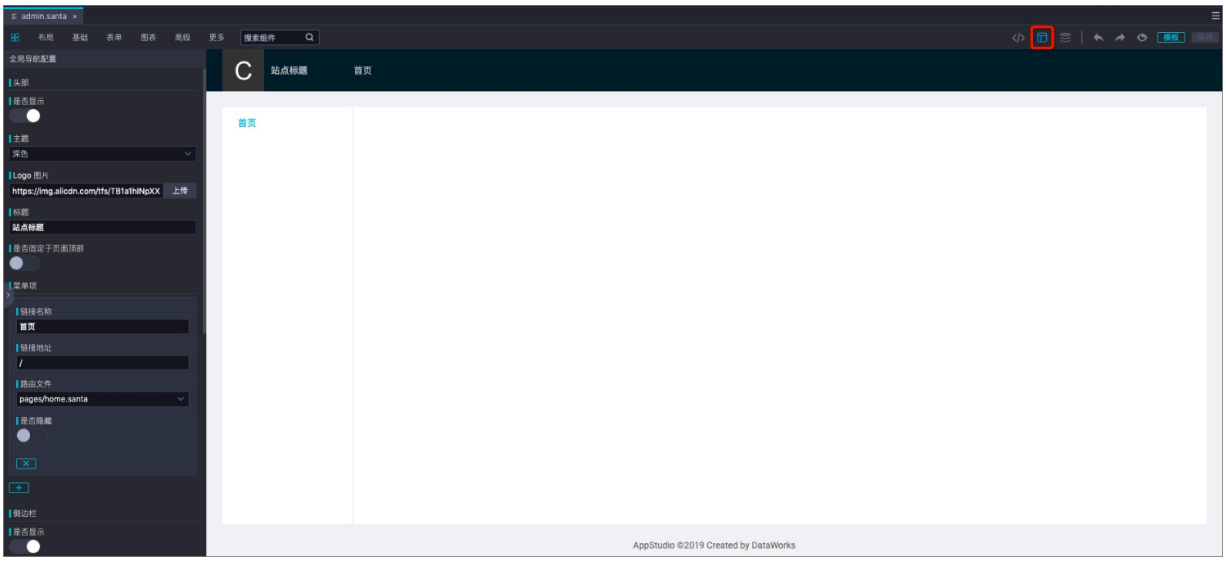
  说明:
  请一定使用$setState方法修改全局数据，使用`state.name = 'new value'`将会无法触发重新渲染。

13.4.9.7 导航配置

本文将为您介绍如何设置可视化搭建站点导航。

App Studio可视化搭建为应用提供了公共的页面头、底部和侧边栏，提供了丰富的菜单配置、主题配置。如果您不需要显示系统提供的公共头和侧边栏，可以进行配置。

单击右上角的导航配置，进入导航配置页面。



公共头部配置

您可以根据自身需求对公共头部进行配置。

配置	说明
是否显示	设置是否显示公共头部。
主题	您可以选择深色或浅色的主题样式。
Logo图片	显示的站点Logo图片，您可以输入一个图片地址，或者选择本地上传一张图片。
标题	设置显示的站点标题。

配置	说明
是否固定于页面顶部	是否让公共头部一定固定于页面顶部（页面滚动时，公共头也将一直位于页面顶部）。
菜单项	您可以定义公共头部可以显示的链接名称、链接地址等菜单项。

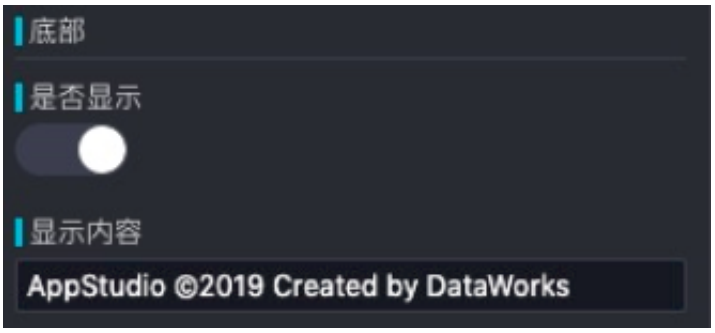
侧边栏配置

您可以根据自身需求对侧边栏进行配置。

配置	说明
是否显示	设置是否显示侧边栏。
主题	您可以选择深色或浅色的主题样式。
标题	设置显示的站点标题。
是否可折叠	设置侧边栏菜单是否具有折叠功能。
菜单项	您可以定义侧边栏可以显示的链接名称、链接地址等菜单项。
是否自动展开所有菜单	设置是否自动展开所有菜单（包括子菜单）。

公共底部配置

您可以根据自身需求对公共底部进行配置。



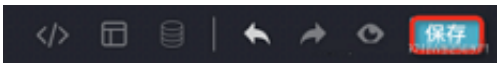
配置	说明
是否显示	设置是否显示公共底部。
显示内容	设置公共底部的显示文案。

13.4.9.8 保存、预览、运行和热部署

可视化搭建系统支持保存、预览、操作和热部署等操作。

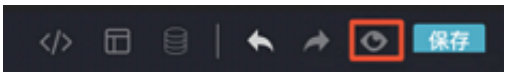
保存

可视化搭建系统会定时保存您的修改，您也可以单击操作面板中的保存按钮，进行保存。



预览

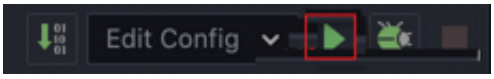
在可视化搭建系统中，可视化操作区域处于编辑的状态。有部分组件针对编辑状态进行特殊处理，只有在正式的运行状态下才能执行正常的渲染逻辑。如果您想查看正常的渲染结果，可以单击操作区域的预览按钮。



运行

可视化搭建系统单次只能打开一个可视化文件进行编辑。如果您想要以整个应用的视角进行查看，可以运行整个应用来查看结果。

您可以单击App Studio Debug面板中的启动按钮，来运行整个应用。



热部署

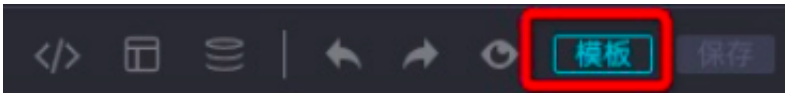
应用启动后，如果您发现页面不符合预期，可以继续返回到可视化搭建系统进行调整。

调整完成后进行保存，您的修改将会支持热部署的方式生效至运行的页面。

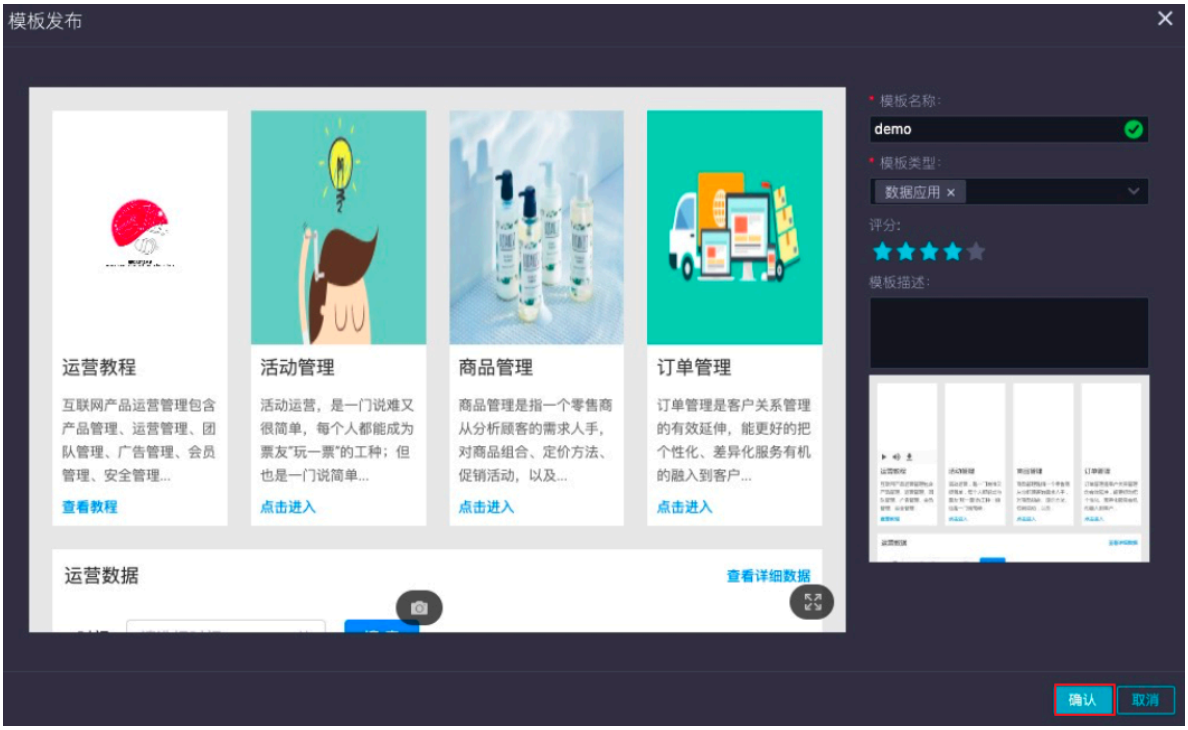
13.4.9.9 保存为模板

您可以将搭建好的前端页面保存为模板，后续可供自己或分享给他人使用。

1. 单击右上角的模板。

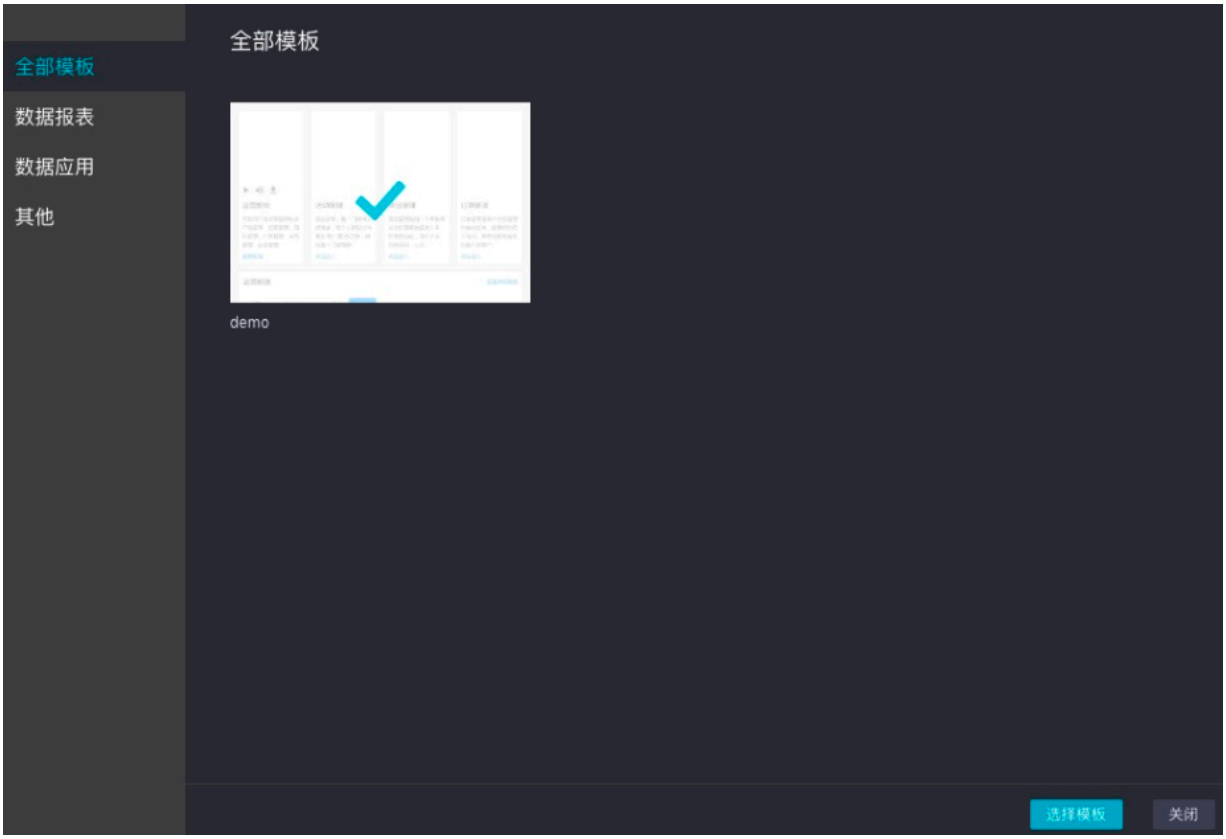


2. 保存模板，单击确认。



3. 右键单击santa > pages，选择新建 > 模板文件。

此时您可看到刚刚保存的模板，您可选择它创建一个页面，并基于模板进行开发。



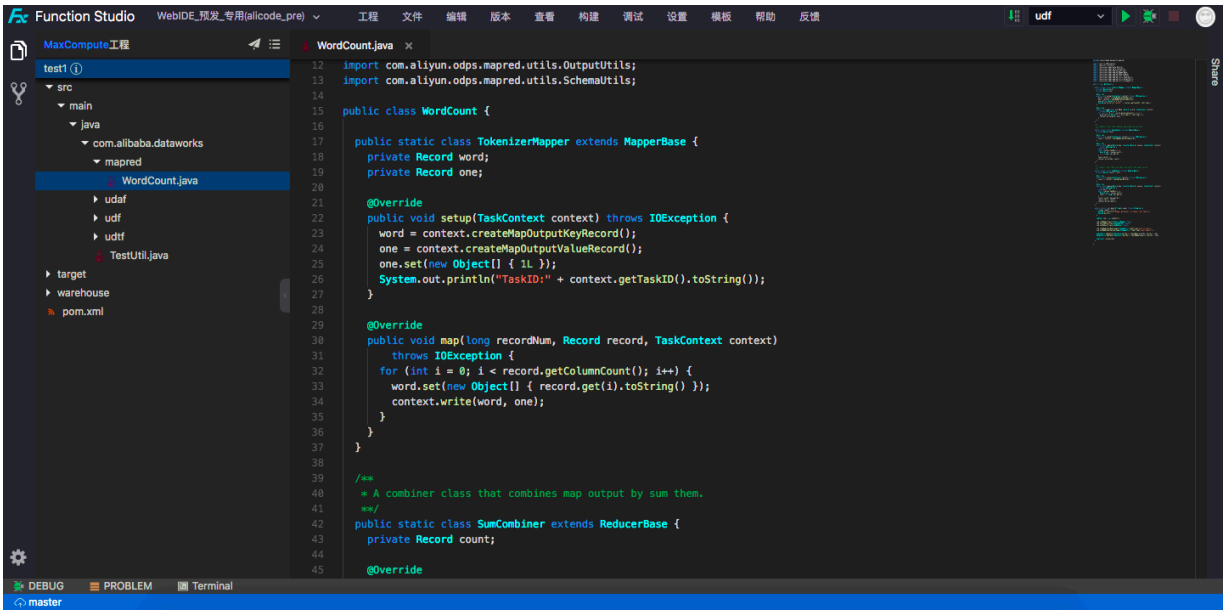
14 Function Studio

14.1 Function Studio简介

Function Studio是由阿里巴巴集团完全自主开发的、面向函数开发场景的Web项目代码编辑开发工具，是DataWorks开发平台的重要组成部分。

基于底层创新性的支撑架构，Function Studio占有资源少、支持高并发，方便、灵活且高效。

Function Studio提供语法高亮、代码自动补全、智能纠错和语法错误提示等功能，并支持在线开发、在线调试，多人协同编辑、一键发布UDF资源和函数到DataWorks。



功能概述

- 支持MaxCompute Java UDF函数的编辑，可编译和一键发布至DataWorks。
- 支持工程下的文件和文件夹对象的各种管理操作。
- 提供上下文相关的智能编辑器，支持智能化的多Java文件同时编辑，支持查找定义、查找引用、智能提示代码补全、语法关键字高亮和实时语法错误提示等功能。
- 支持UDF/UDAF/UDTF等函数模板，并快捷自动发布资源和函数至DataWorks的业务流程中，极大提高了UDF函数的开发效率。
- 开发环境集成了提交、推送等常见的Git操作，实现了代码文件的版本管理。
- 支持从DataStudio一键跳转至Function Studio，查看UDF函数的源代码，方便在线进行UDF的维护和管理。

未来发展

后续Function Studio将支持Python等更多语言， 支持实时计算等更多平台上的函数开发场景。

14.2 Function Studio版本历史

本文将为您介绍Function Studio版本内容的更新情况，基于此您可以了解Function Studio支持的新功能和语法特性，提高项目开发效率。

Function Studio 1.0

发布日期：2018-12-11

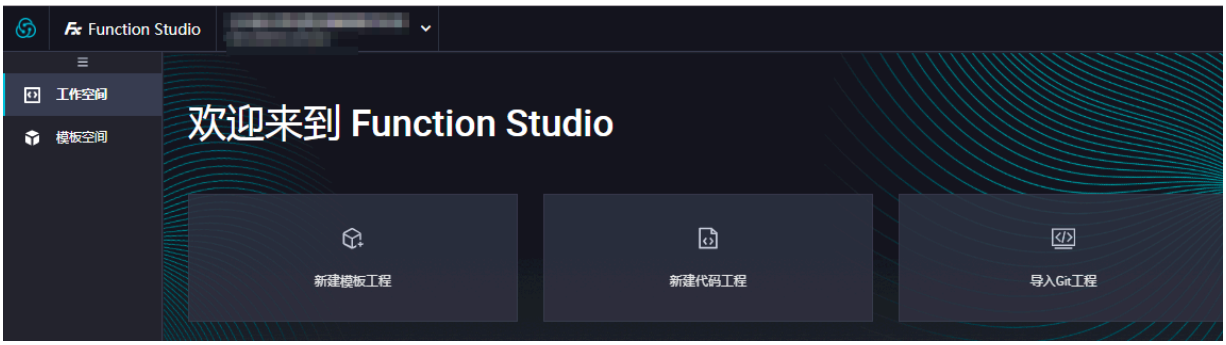
- 全新推出支持在线开发UDF（Java）的IDE产品。
- 支持一站式开发UDF项目，编译发布UDF资源或函数。
- 函数或资源发布后，您可进入Function Studio页面，对其进行维护或二次开发。
- 支持Java的高级编辑功能：代码提示、 跳转和重构。
- 支持Git的各项功能。
- 支持在线Debug，支持Run/Debug模式下的热部署。

14.3 Function Studio快速开始

14.3.1 新建工程

本文将为您介绍如何新建和管理工程。

您可以新建模板工程、新建代码工程和导入Git工程。



新建模板工程

1. 进入Function Studio页面，单击工作空间页面的新建模板工程。

2. 填写新建项目对话框中的工程名和工程描述，选择相应的模板。

> 说明:
> 您可以选择自己定义的模板，也可以选择系统提供的模板创建工程。

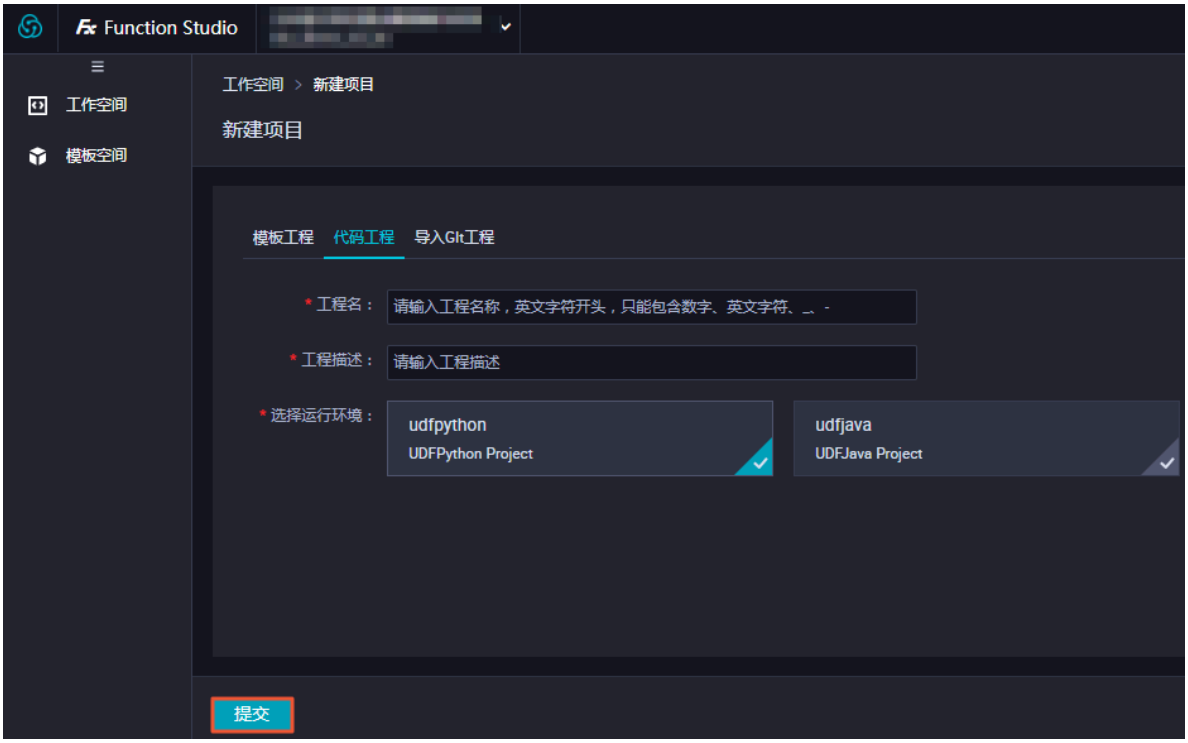
3. 配置完成后，单击提交。

新建代码工程

如果想进行纯代码开发的工程，可以通过代码创建工程。Function Studio提供了2种运行环境的代码模板，您可以根据自身需求进行选择。

1. 进入Function Studio页面，单击工作空间页面的新建代码工程。

2. 填写新建项目对话框中的工程名和工程描述，选择相应的模板。



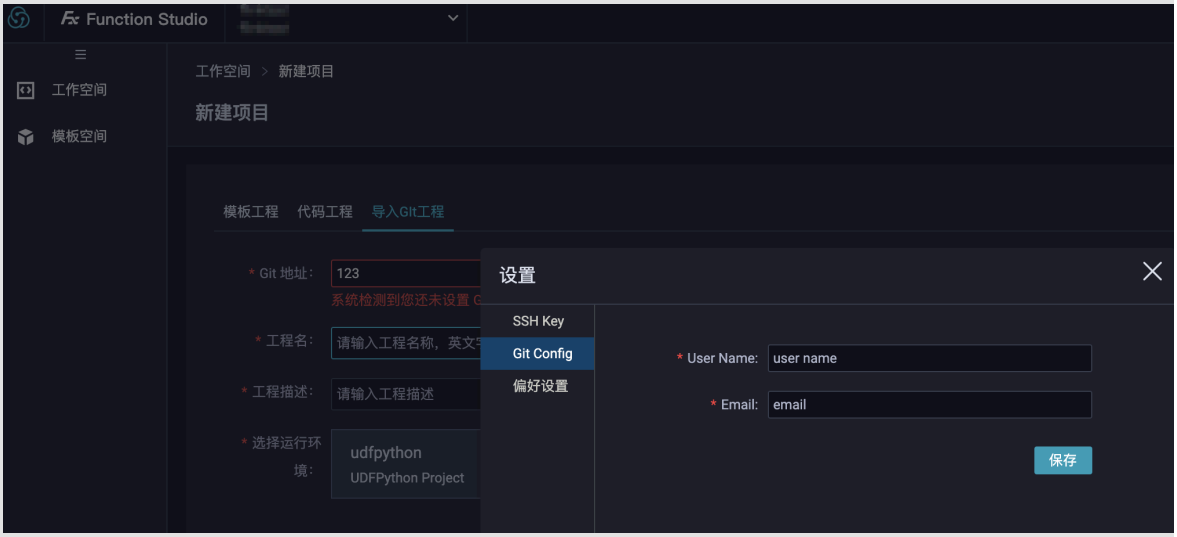
3. 配置完成后，单击提交。

导入Git工程

如果您已经有Git代码，可以直接导入Git代码创建工程。

> **说明:**
>
> · Function Studio支持直接导入Git工程，仅支持http格式（支持将SSH方式转化为http方式）。

- 您在导入Git工程前，需要录入当前用户的用户名、Email、对应Git服务商的SSH等基本信息。如果没有设置，会报错并弹出设置对话框，引导您进行设置。



您也可以进入工程的编辑页面，单击菜单栏中的设置，对SSH Key、Git Config和偏好设置进行修改。

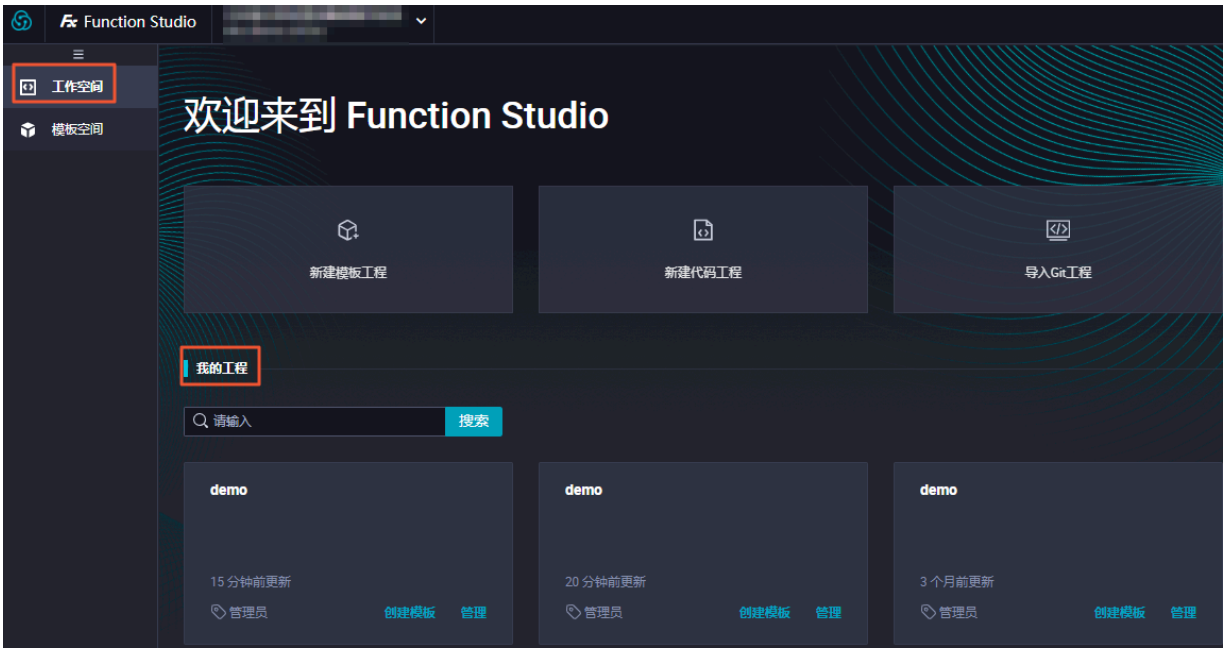
1. 进入Function Studio页面，单击工作空间页面的导入Git工程。

2. 填写新建项目对话框中的Git地址、工程名和工程描述，选择相应的运行环境。

3. 配置完成后，单击提交。

工程列表

您可以在工作空间页面查看创建的工程。



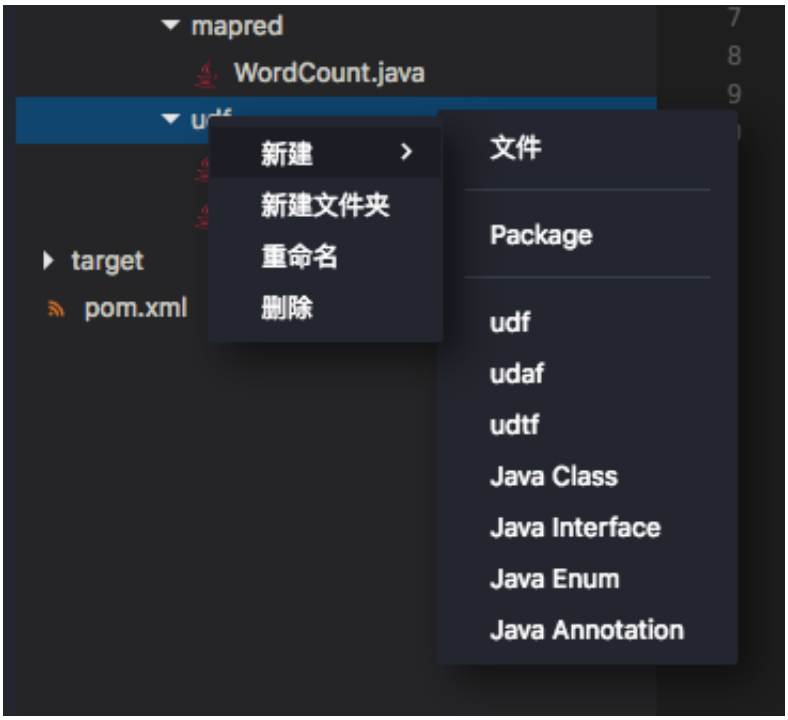
您可以直接单击相应的工程名称，进入工程编辑页面。也可以单击相应工程下的创建模板，通过工程创建模板。

Function Studio对工程可以进行部署的版本管理，单击相应工程下的管理，即可进入部署版本管理页面。

14.3.2 UDF开发

新建工程完成后会自动生成一个框架代码，支持新建UDF、UDAF和UDTF，本文将以新建UDF为例为您介绍如何进行开发。

1. 选择新建 > UDF。



2. 在弹出框中输入类名，单击确认，即可自动生成框架代码。

3. 根据自身需求修改evaluate方法中的内容，来实现UDF的开发。

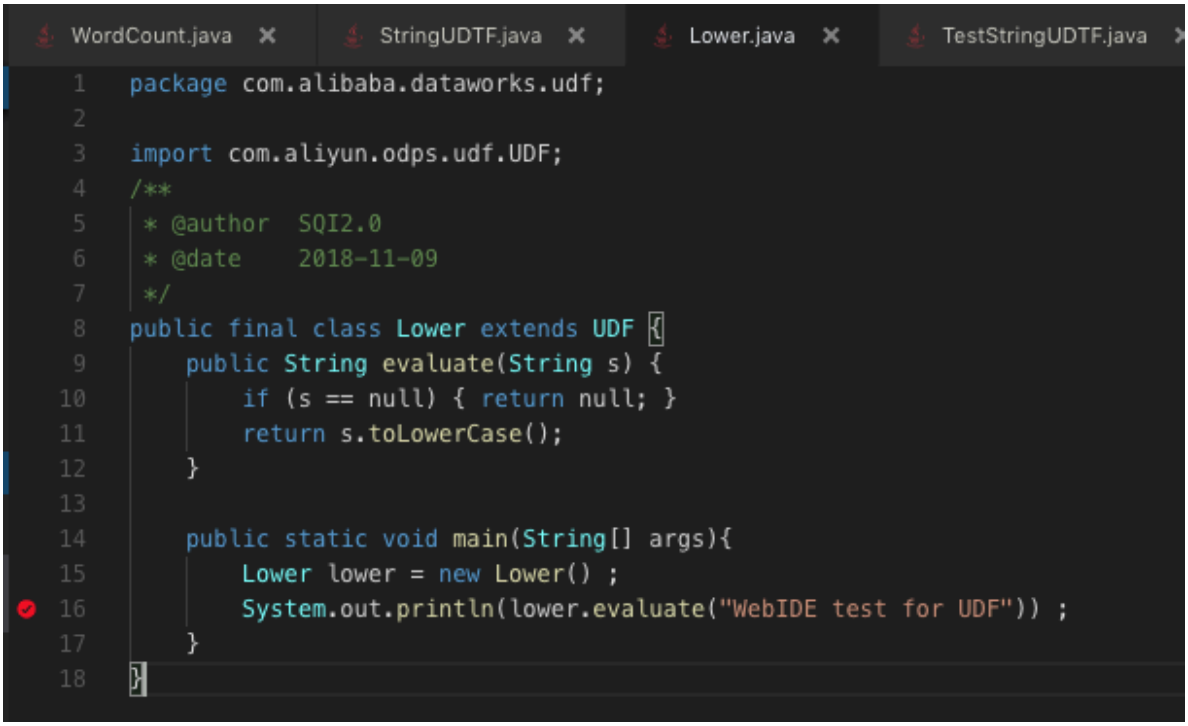
14.3.3 UDF调试

UDF调试包括UDF（仅支持Java）、UDAF和UDTF的调试。

UDF调试（目前仅支持Java）

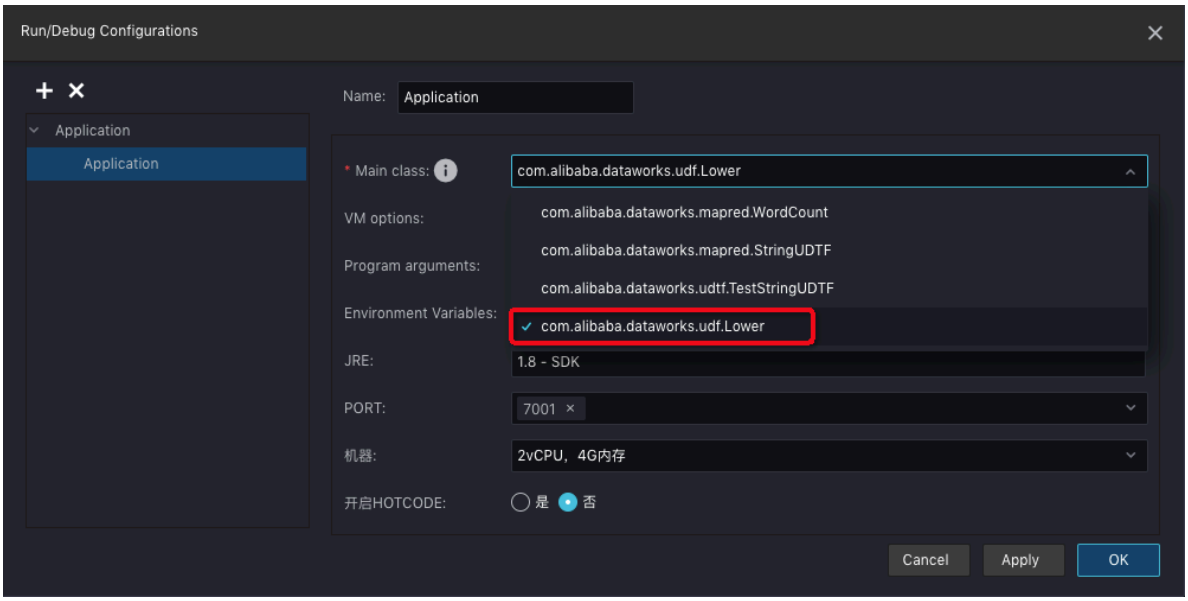
1. 新建main函数。

   Function Studio目前支持通过main函数调用UDF，进行在线调试。



2. 设置Debug配置。

单击右上角的config，进入配置页面。

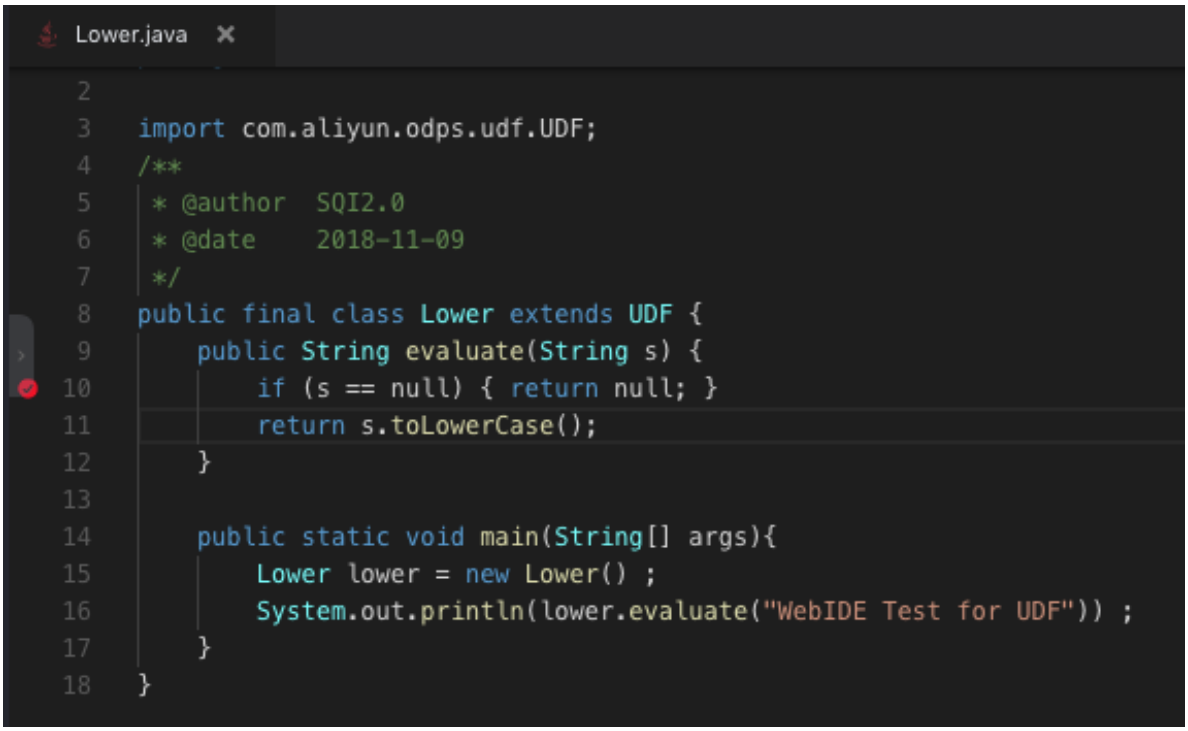


您只需要选择新建的main函数，其他的信息都是自动生成的。

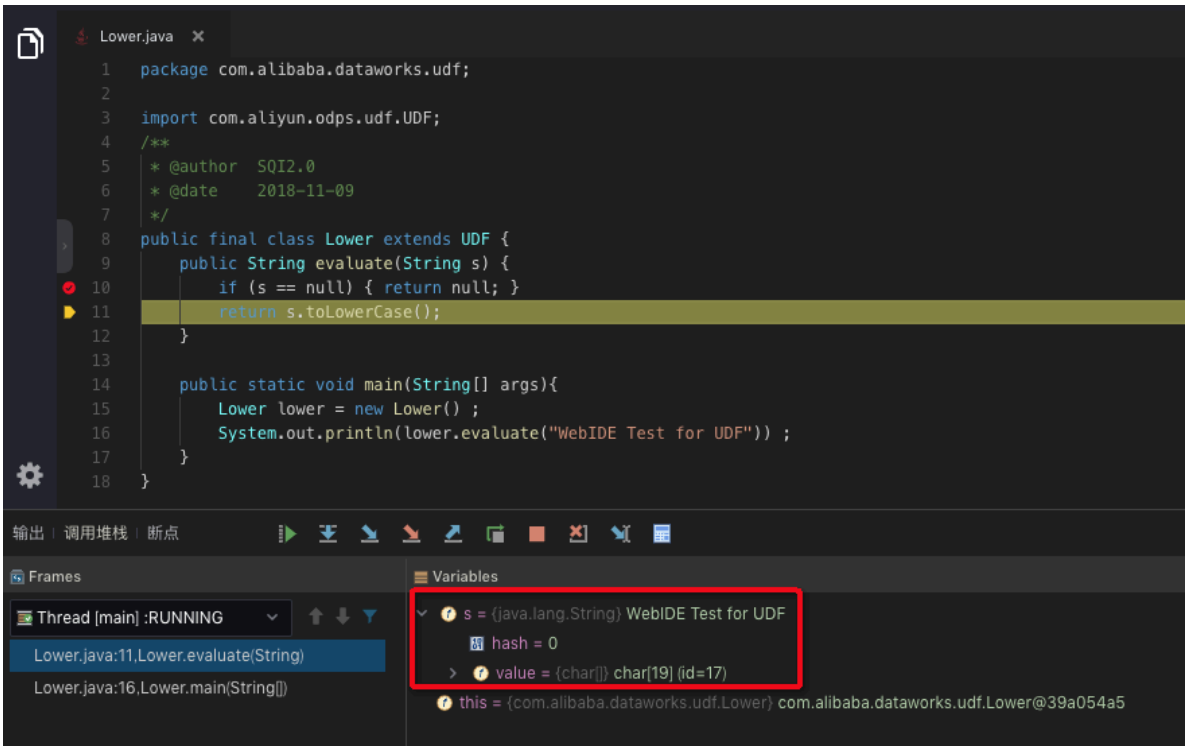
配置	说明
Main class	选择需要调试的入口函数，必填项。
VM options	需要启动的JVM参数，非必填项。
Program Variables	启动参数，非必填项。
JRE	目前仅支持JDK1.8。
PORT	需要开放的HTTP端口，UDF/UDAF/UDTF类的项目此配置为非必填项。
机器	机器规格。
开启HOTCODE	是否需要支持热部署。

3. 启动Debug。

在Lower的evaluate函数上断点后，启动Debug。

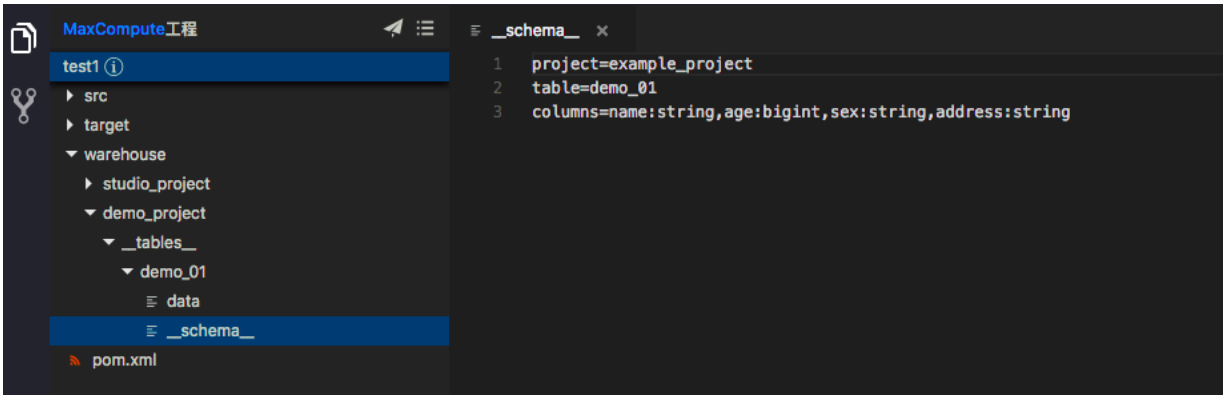


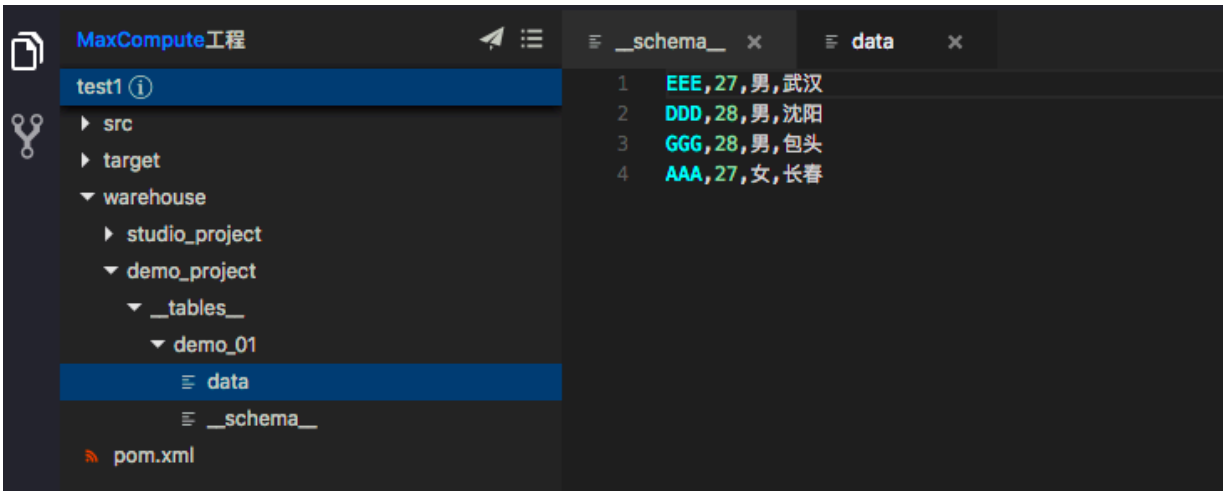
启动成功后即可进行正常的调试。可通过step into进入到UDF方法内，并查看变量值。



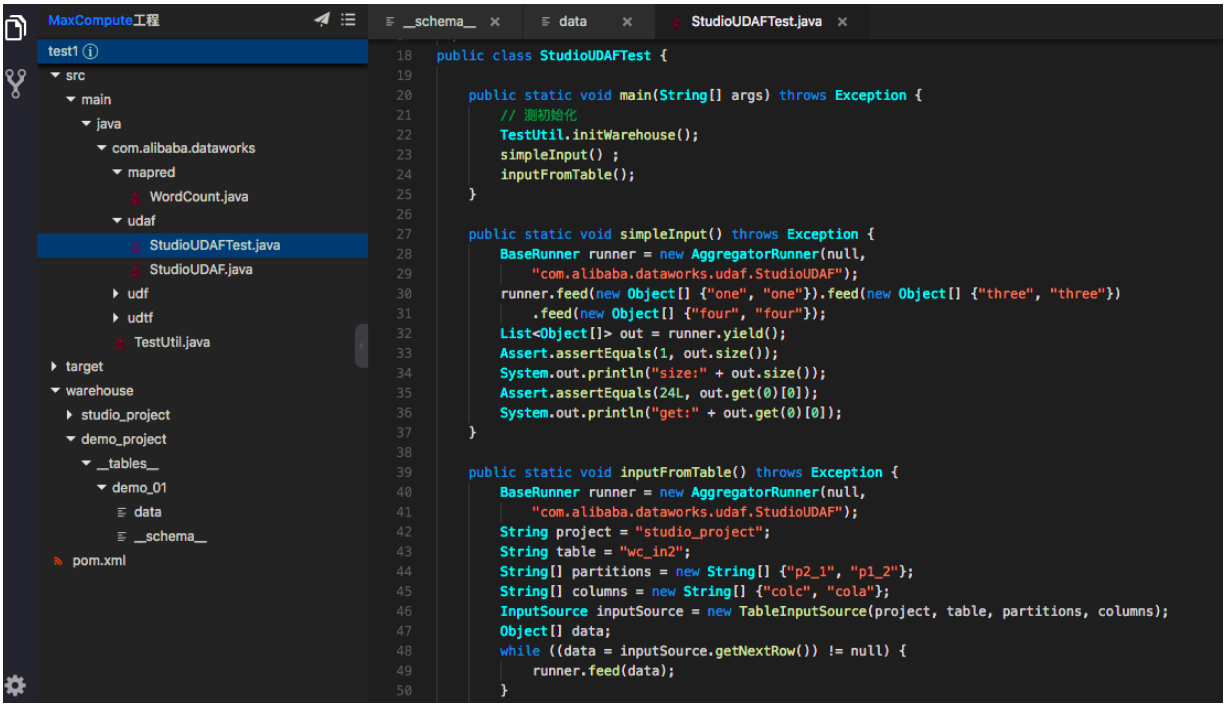
UDAF调试

UDAF的调试需要自己构造相关数据，并且使用warehouse来模拟MaxCompute的数据。warehouse下会保存相关表的Schema以及Data，然后编写相关测试的main函数。



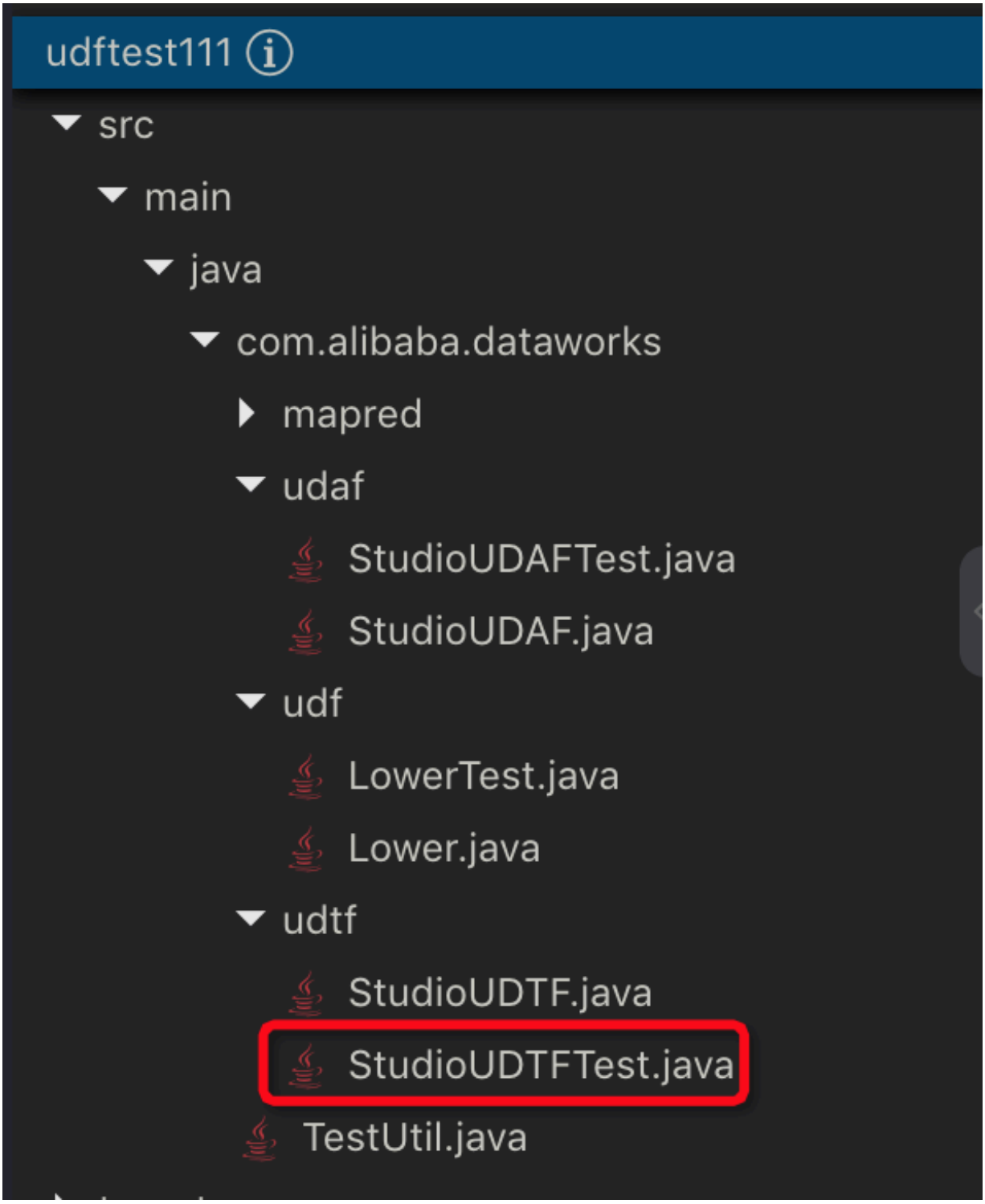


初始化warehouse后，调用相关的UDAF进行测试。



UDTF的调试

UDTF与UDAF的调试一样，初始化工程中已存在UDTF的测试类，直接执行该类，即可模拟真实数据对UDTF进行调试。



单击执行，如果程序没有抛异常，则表示运行通过。

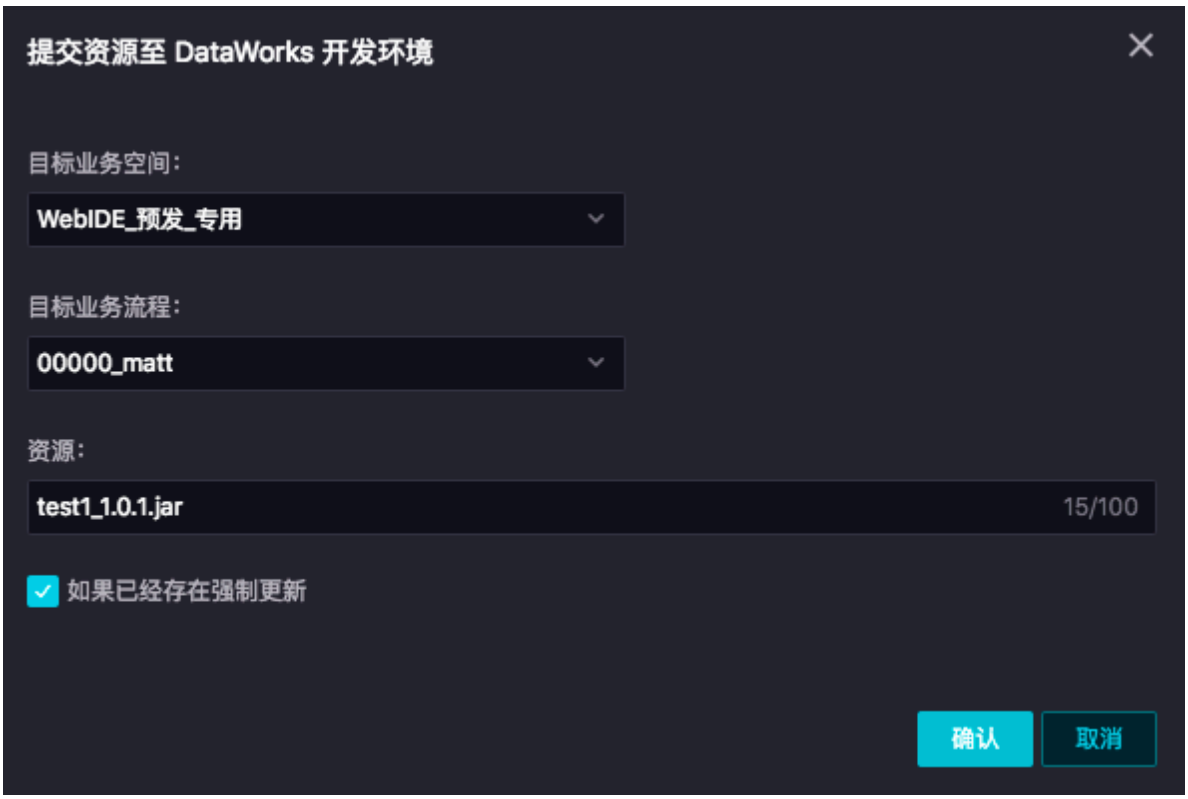
14.3.4 UDF发布

本文将为您介绍如何提交资源或函数到DataWorks开发环境。

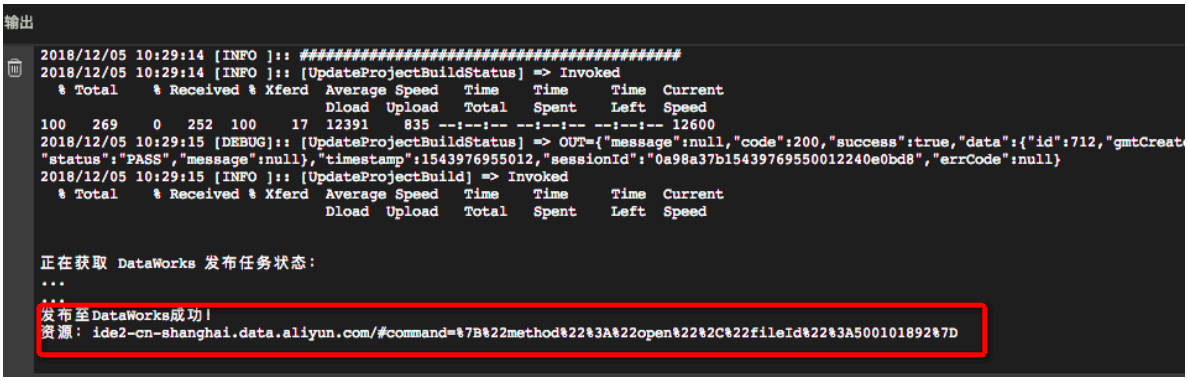
提交资源到DataWorks开发环境

1. 单击工程区右上角的发布标识。

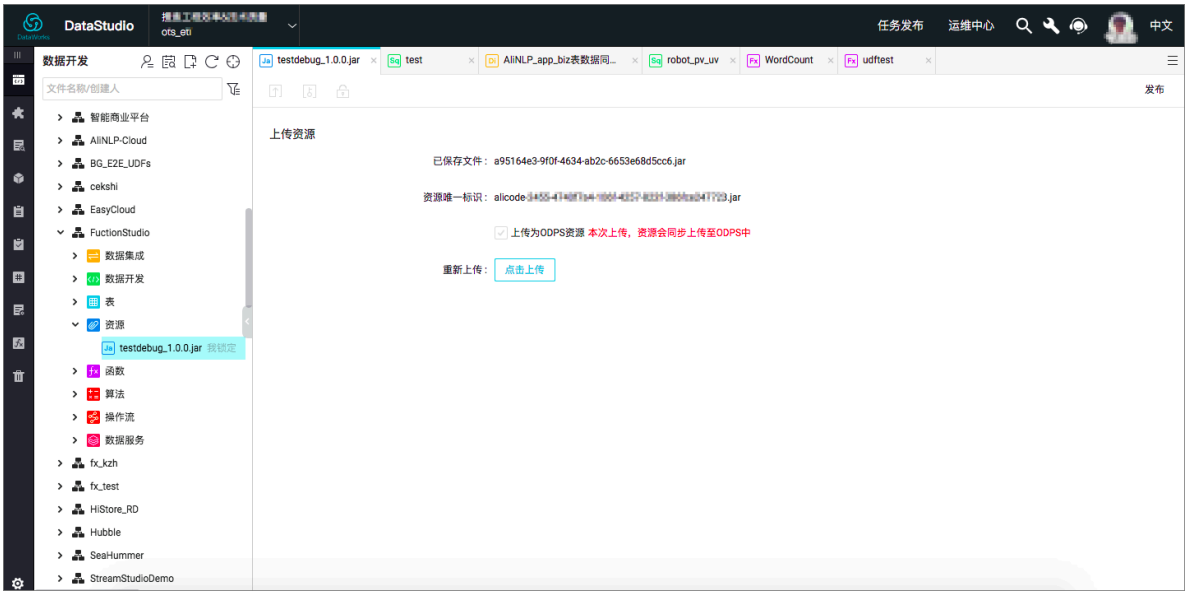
2. 单击提交资源至DataWorks开发环境。



3. 填好弹出框中的配置信息，单击确认即可进行发布，发布完成后会给出资源在DataWorks中的定位链接。



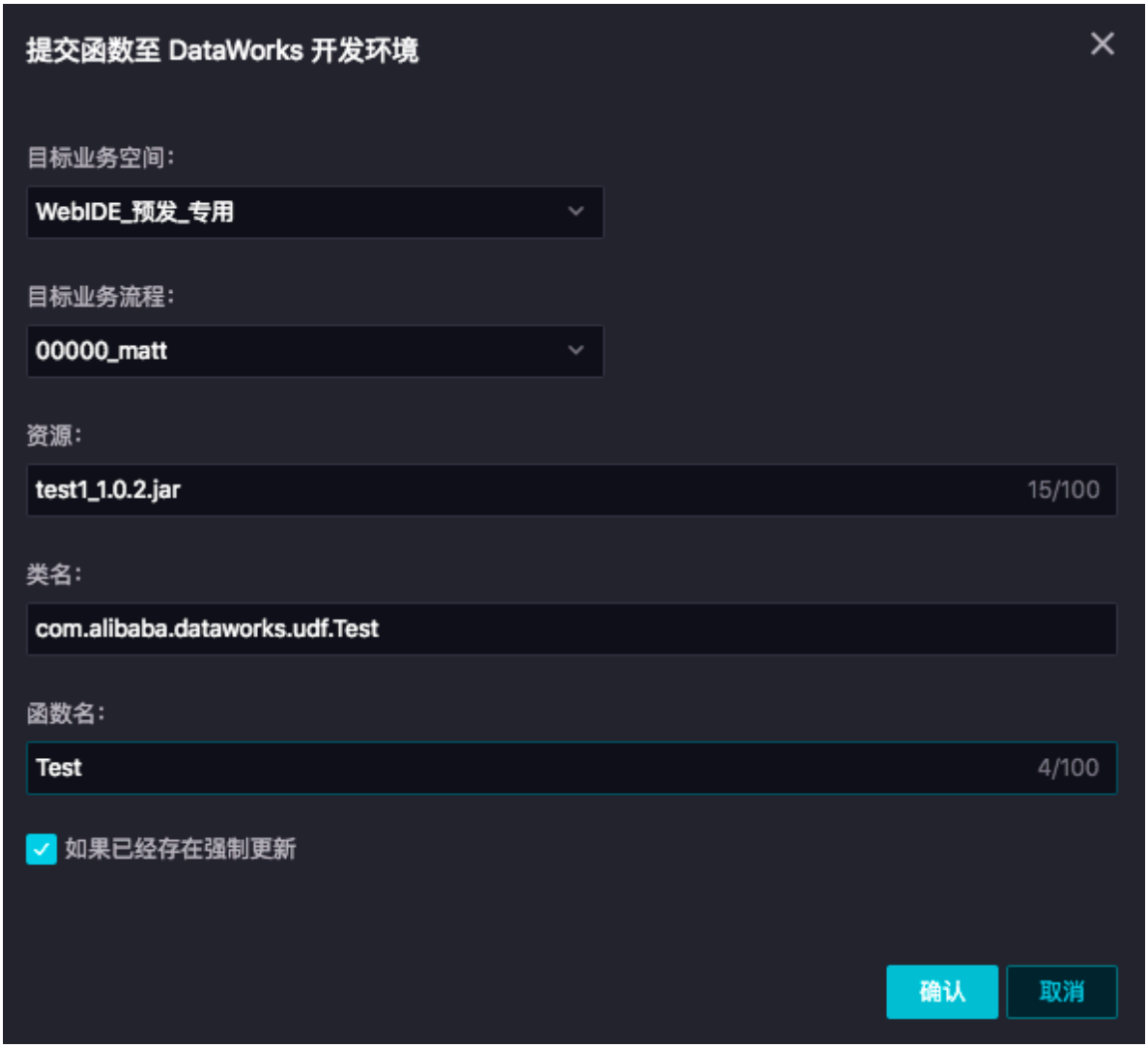
4. 复制链接再打开即可定位到该资源。



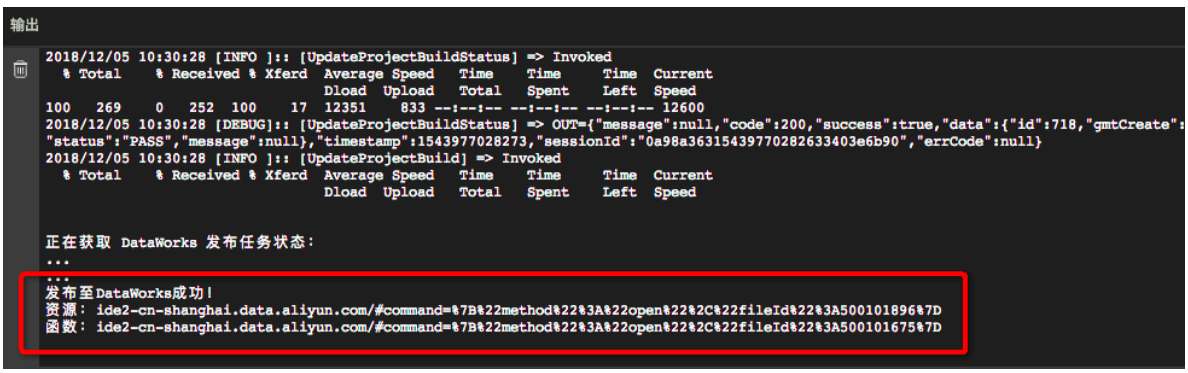
提交函数到DataWorks开发环境

1. 单击工程区右上角的发布标识。

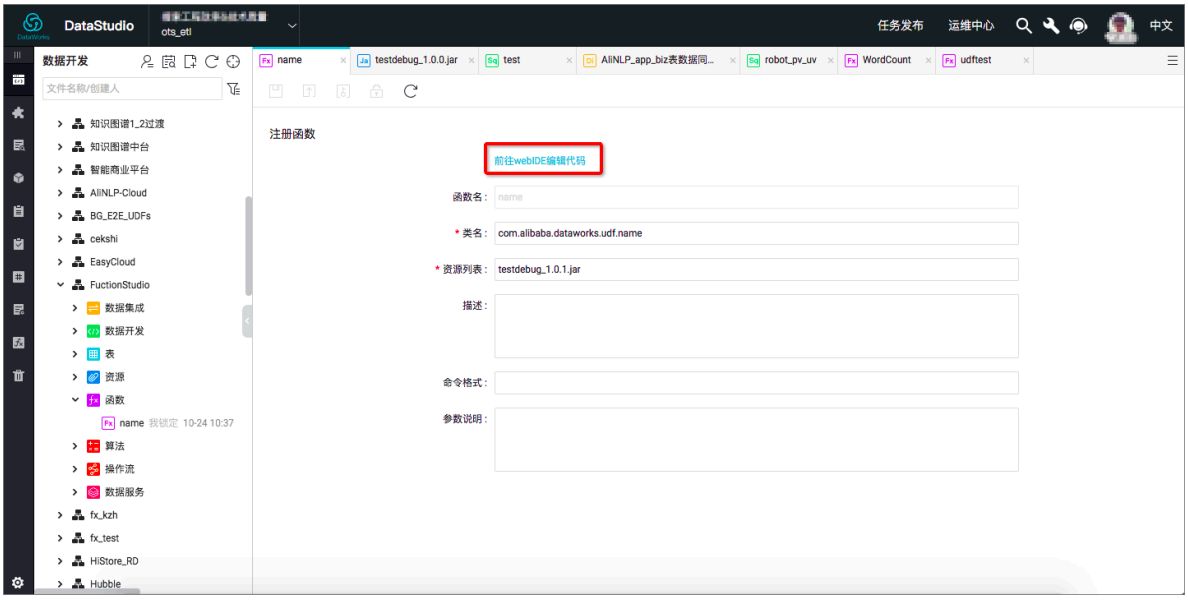
2. 单击提交函数到DataWorks开发环境。



3. 填写弹出框中的配置信息，单击确认即可进行发布，发布完成后会给出函数在DataWorks中的定位链接。

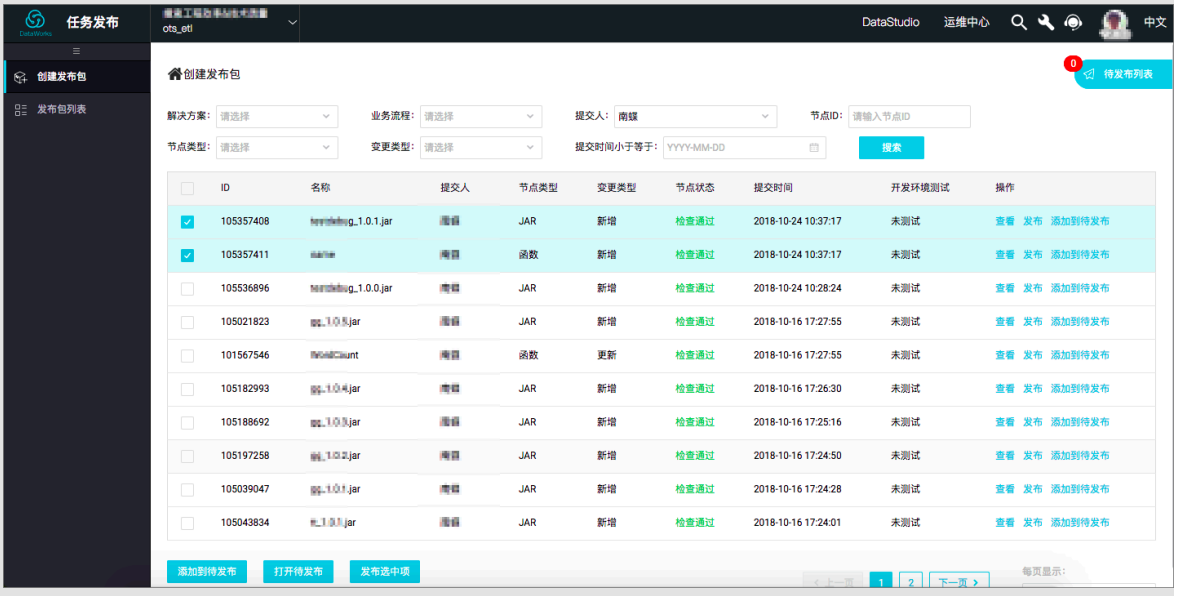


4. 复制链接打开函数，在函数详情页中有链接可前往Function Studio进行函数编辑。



说明：

发布后的资源和函数在开发环境，需要在任务发布中将资源和函数发布到线上才能在线上使用。



14.3.5 MapReduce功能开发

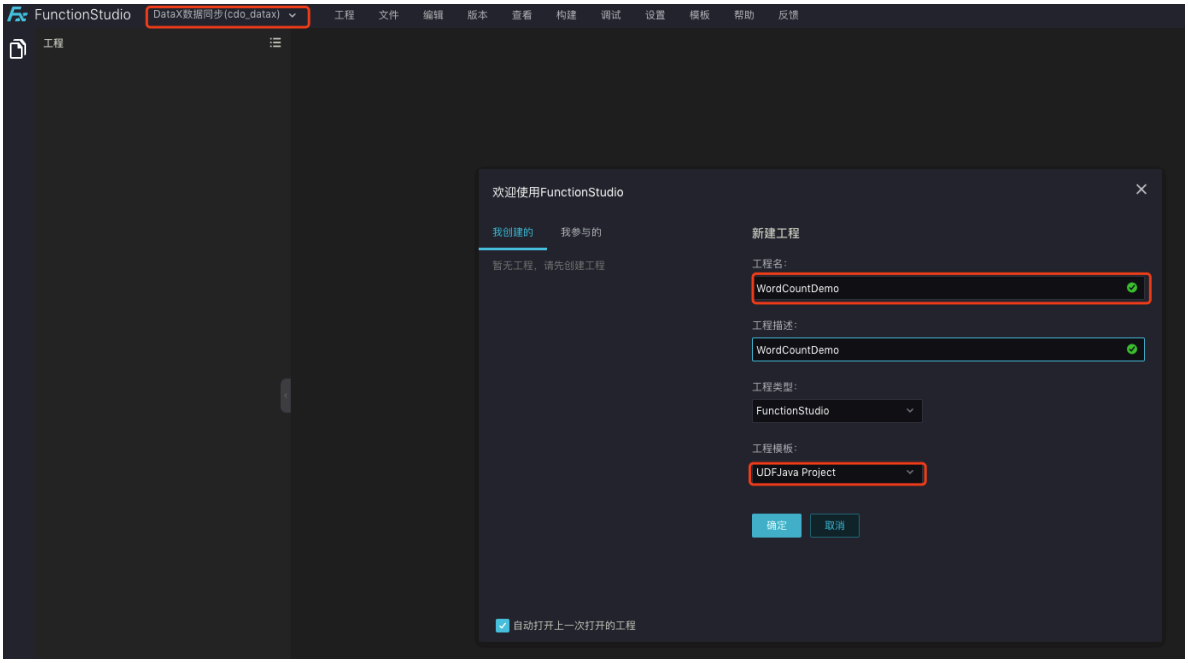
工程创建完成后会自动生成框架代码，支持新建MapReduce任务。本文将以WordCount示例代码为例，为您介绍如何从0开始测试和发布。

新建工程

1. 单击顶部导航栏中的工程，选择新建工程。

2. 填写新建工程对话框中的配置信息。

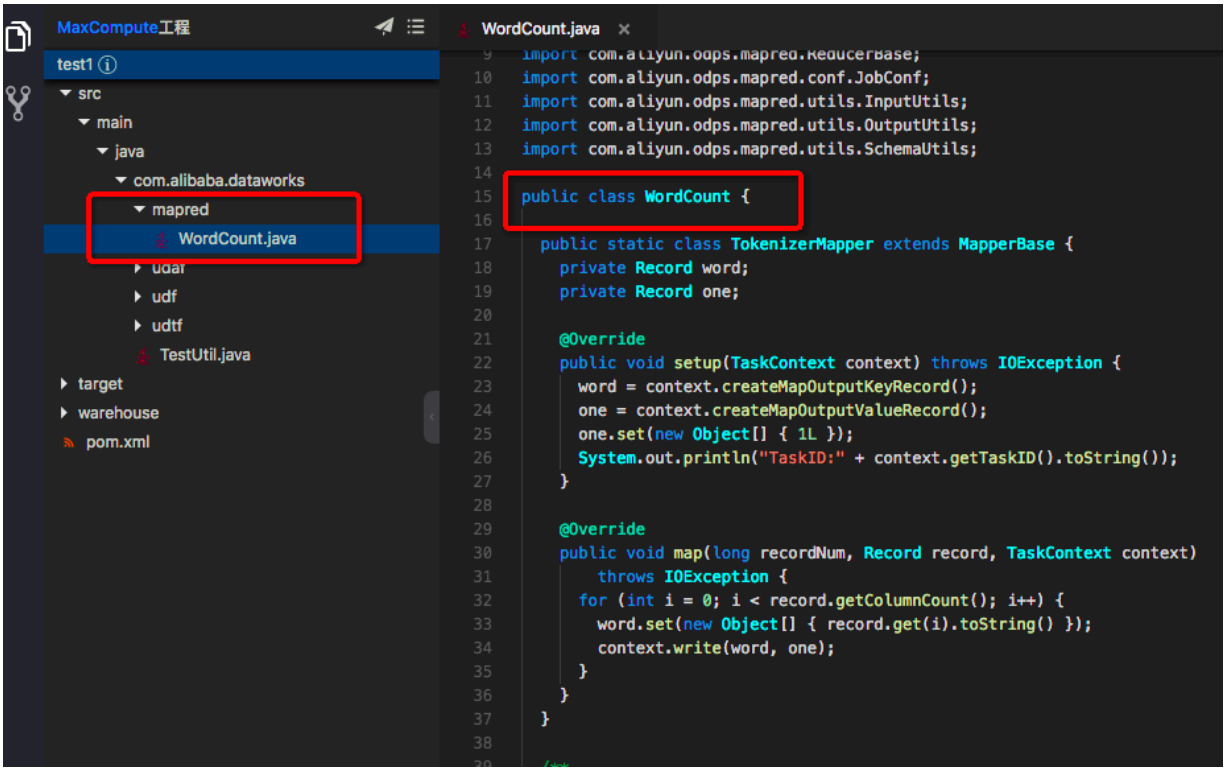
   在指定MaxCompute工程空间（cdo_datax）下创建项目（wordcountDemo），工程模板选择UDFJava Project。



3. 单击确定。

项目开发

在mapred包下，已存在WordCount的MapReduce示例代码。示例代码的功能是对输入表中的单词进行次数统计，将统计结果写入到输出表中，输入输出分别是两个表，详情请参见MapReduce。



项目调试

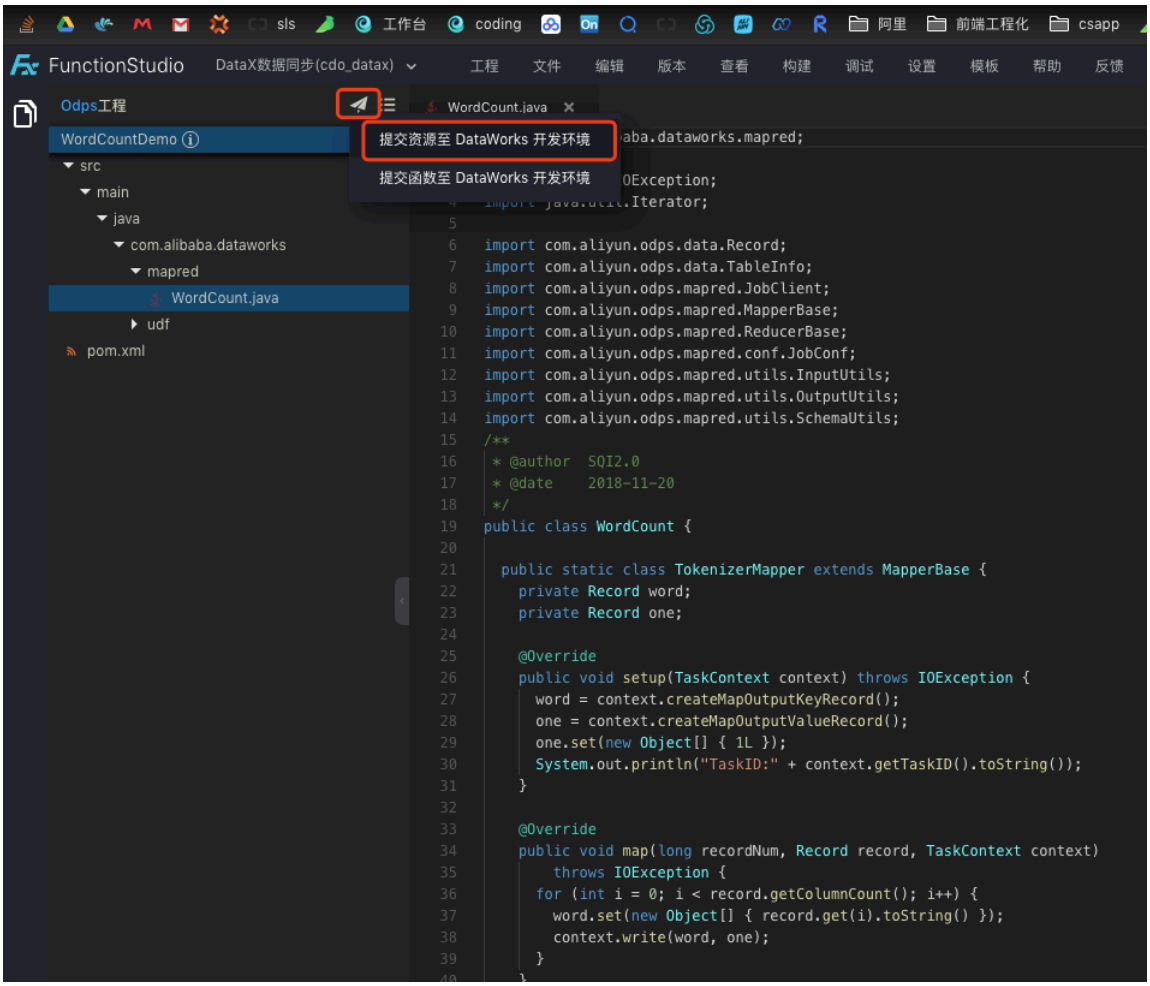
MapReduce项目目前不能在Function Studio中进行Debug，需要将代码发布至DataWork开发环境，然后跳转到DataWorks中进行逻辑验证。

> **说明:**
> Function Studio目前仅支持编码和编译打包两个功能。

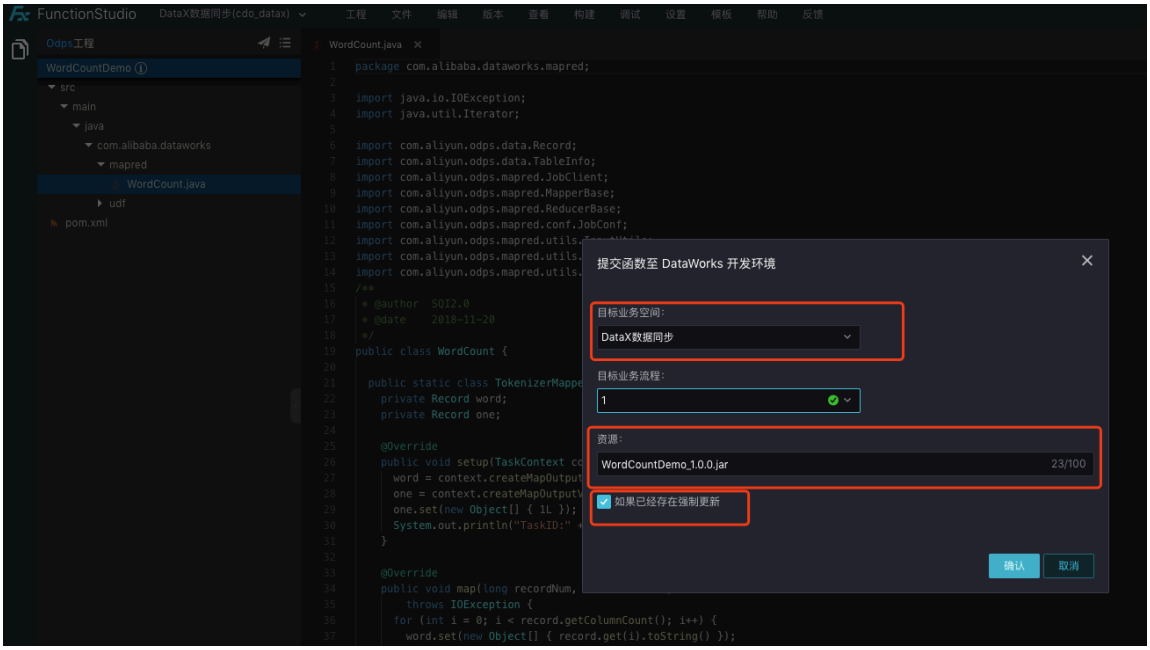
项目发布

1. Function Studio编译打包代码发布到DataWorks开发环境。

   a. 单击发布，选择提交资源到DataWorks开发环境。



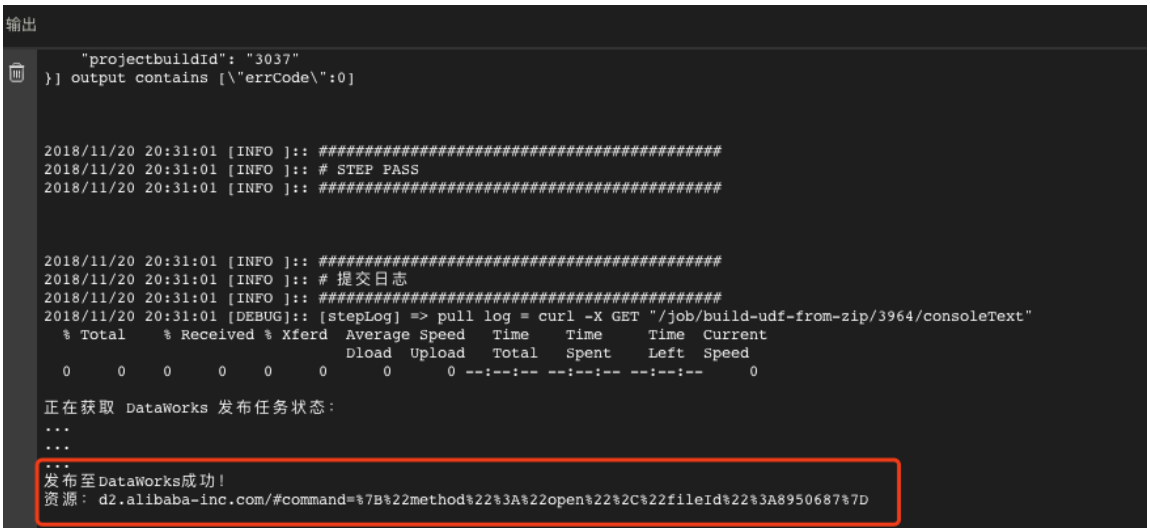
   b. 填写提交函数至DataWorks开发环境对话框中的配置信息。



配置	说明
目标业务空间	发布Jar包的目标业务空间，需要和后续建立的DataWorks计算节点在一个空间中。此处的目标业务空间是DataX数据同步（cdo_datax）。
目标业务流程	选择目标业务流程。
资源	您可以自定义资源名称，在后续的计算节点脚本中会被引用。
如果已经存在强制更新	名称可以和上次发布的一致，勾选如果已经存在强制更新，名称会被覆盖。

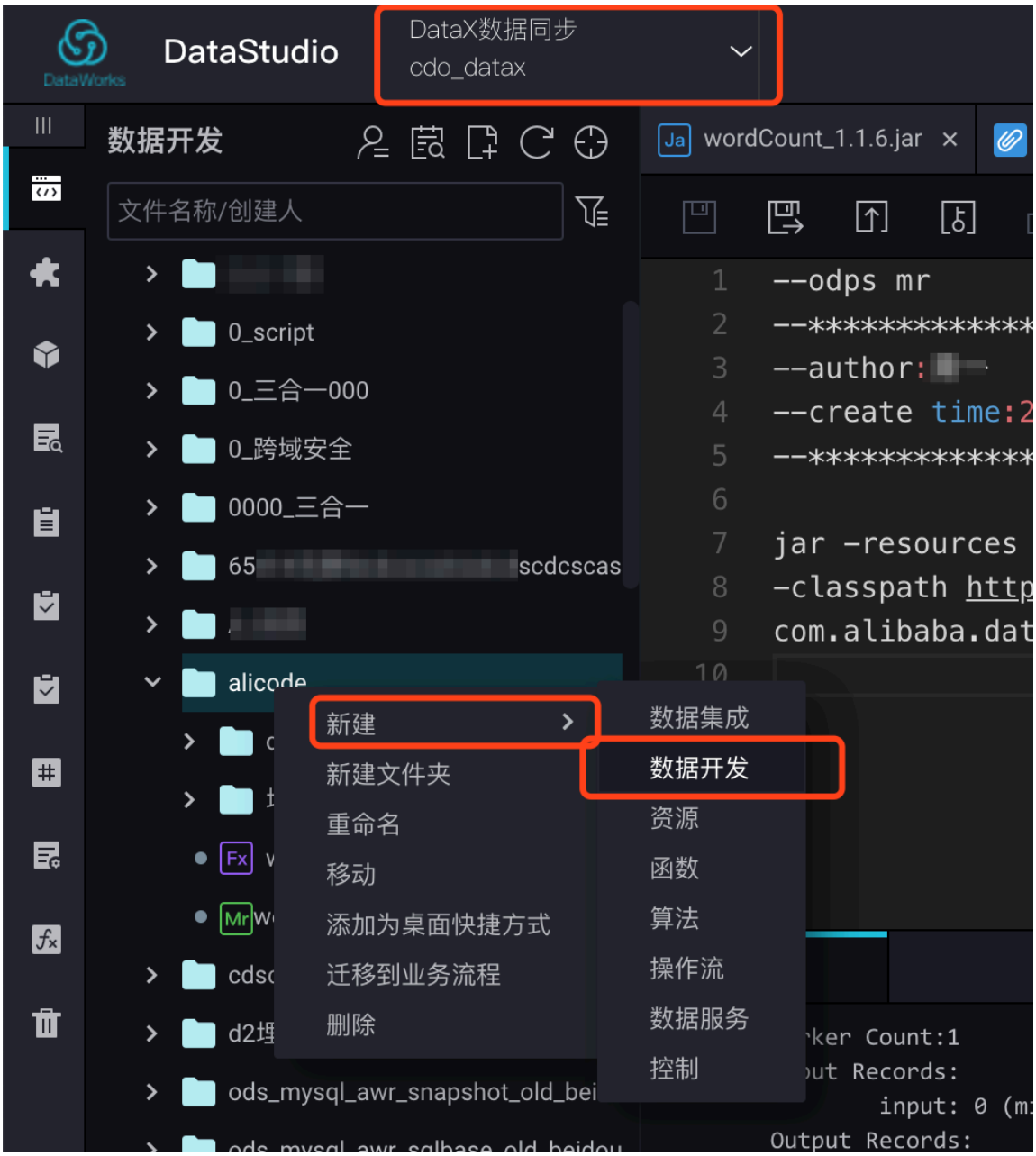
c. 单击确认，编译发布到DataWorks开发环境。

信息提示窗口会输出本次编译发布成功还是失败。

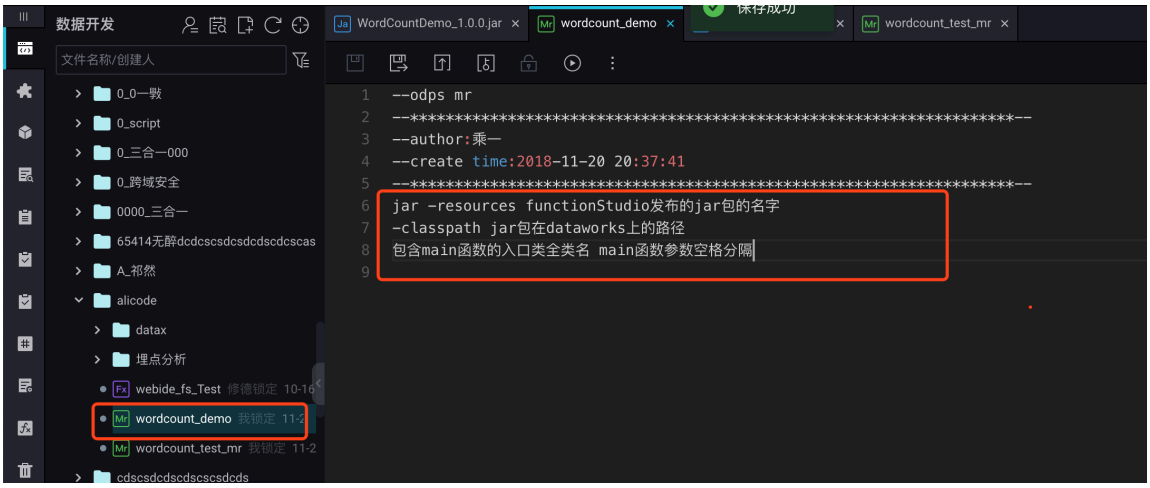


2. 在DataWorks中创建MapReduce节点进行测试。

   a. 打开DataWorks同名工作空间，创建ODPS MR类型节点。



b. 计算节点需要写入以下固定脚本，目前脚本的一些变量还需被手工替换成相关Jar包的信息。



说明：

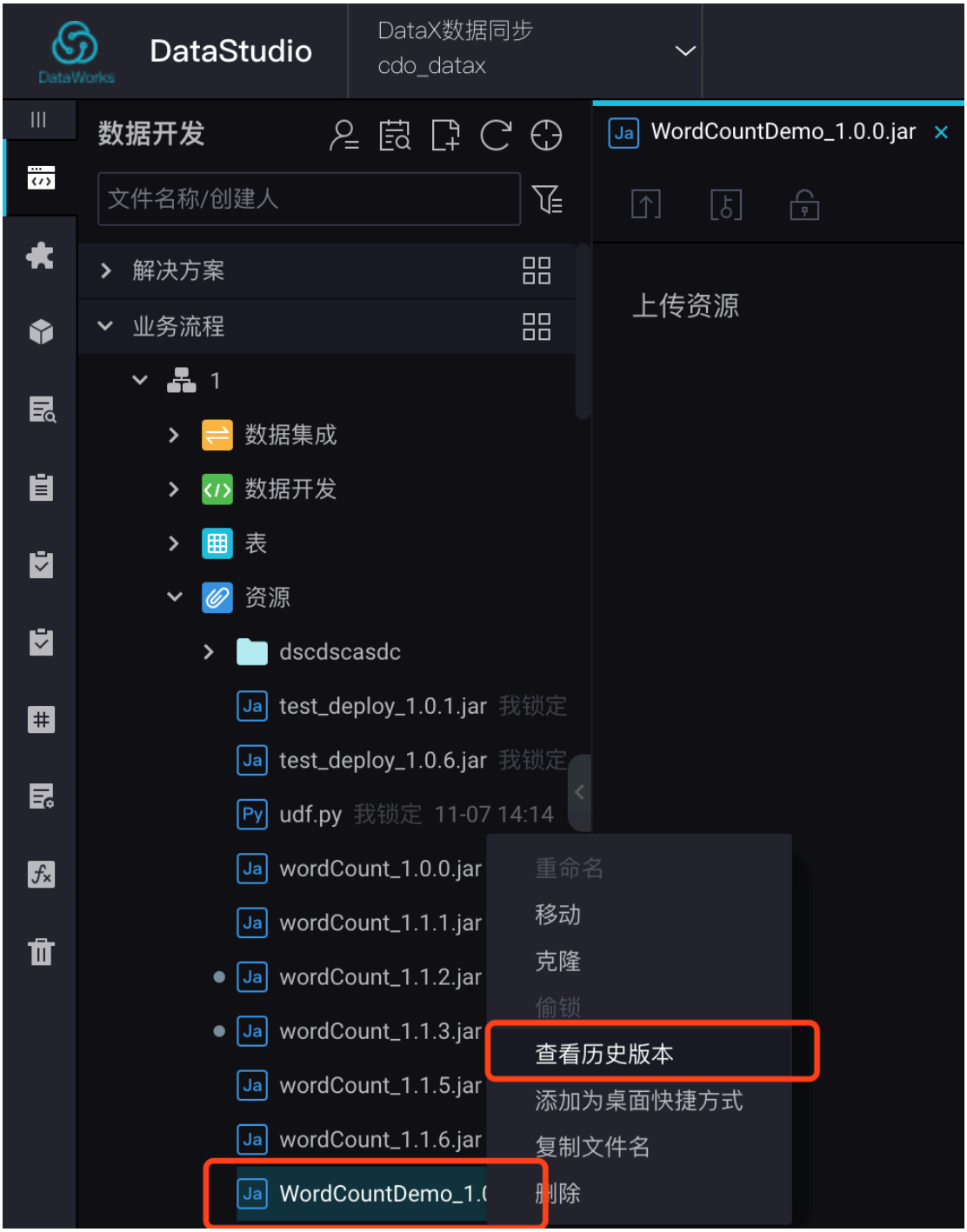
请使用在Function Studio中发布的Jar包的信息来替换脚本中的信息，生成最后的代码。

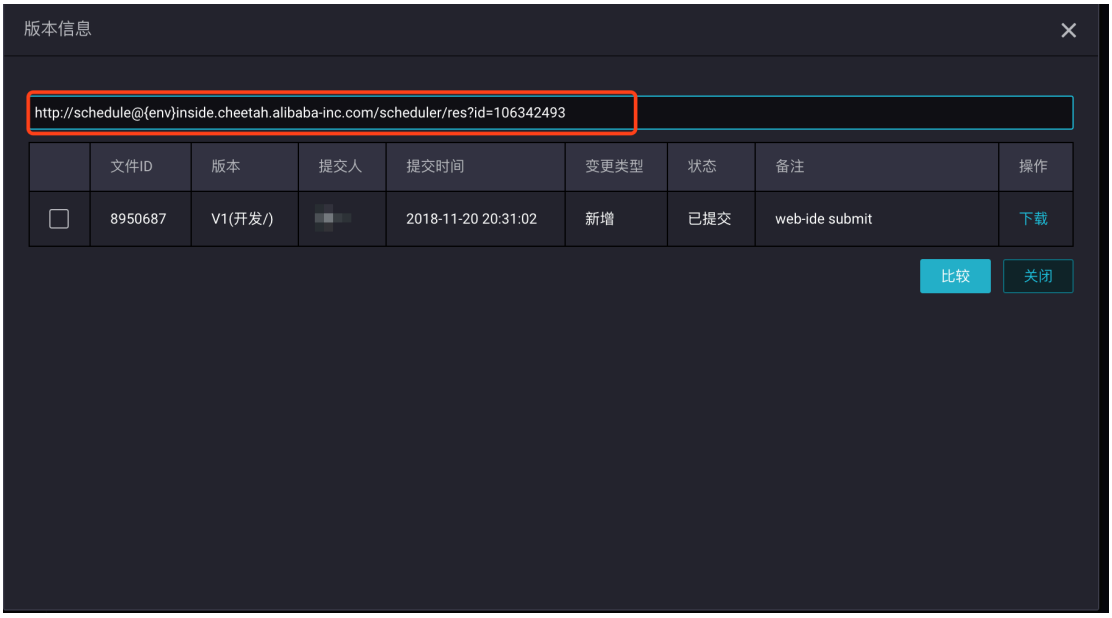
- `jar -resources`：FunctionStudio发布的Jar包名称。
- *-classpath*：Jar包在DataWorks中的路径。

- 包含main函数的入口类全类名main函数参数空格分隔。

c. 选择相应业务流程下的资源，查看在Function Studio中发布的Jar包信息，来替换脚本中的相关信息。

- Jar包的名称为`WordCountDemo_1.0.0.jar`，对应脚本中的`-resource`。
- 右键单击Jar包，选择查看历史版本，即可查看路径的名称`http://schedule@{env}inside.cheetah.alibaba-inc.com/scheduler/res?id=106342493`，对应脚本中的`-classpath`。


DataStudio
DataX数据同步
cdo_datax
数据开发
文件名称/创建人
解决方案
业务流程
1
数据集成
数据开发
表
资源
dscdscasdc
test_deploy_1.0.1.jar 我锁定
test_deploy_1.0.6.jar 我锁定
udf.py 我锁定 11-07 14:14
wordCount_1.0.0.jar
wordCount_1.1.1.jar
wordCount_1.1.2.jar
wordCount_1.1.3.jar
wordCount_1.1.5.jar
wordCount_1.1.6.jar
WordCountDemo_1.(
WordCountDemo_1.0.0.jar
上传资源
重命名
移动
克隆
偷锁
查看历史版本
添加为桌面快捷方式
复制文件名
删除

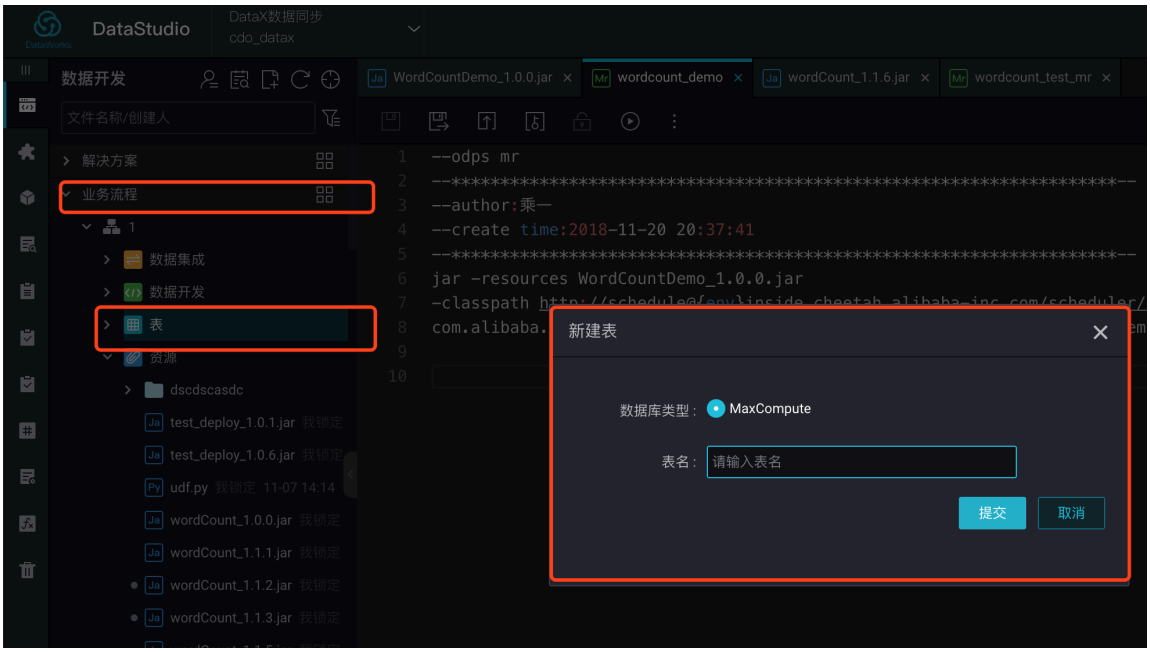


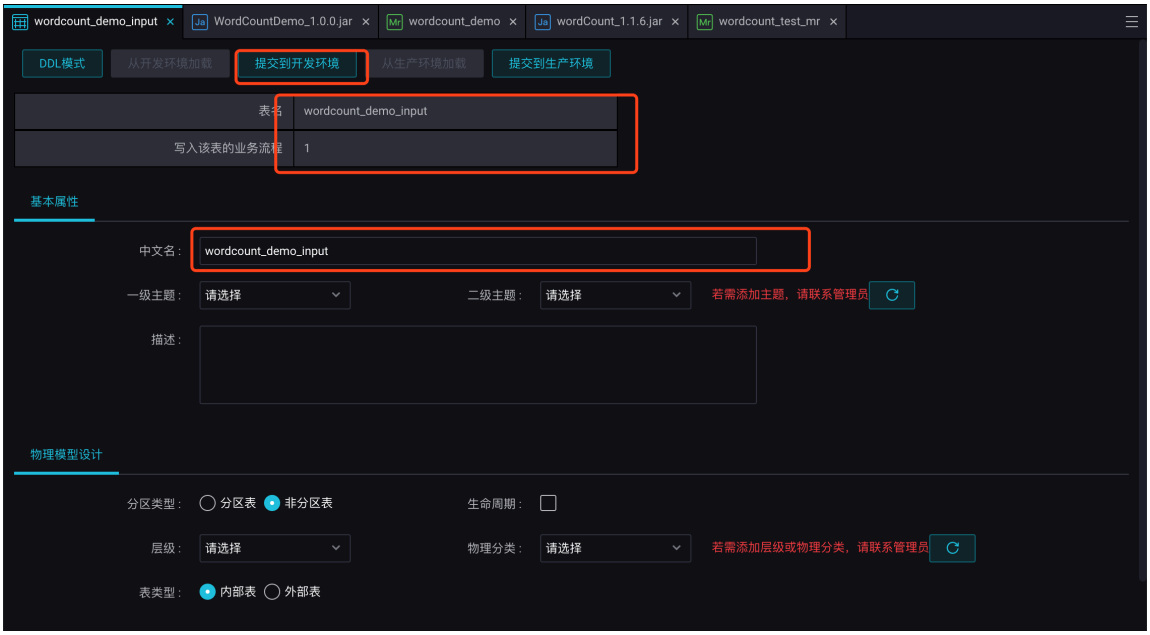
最后的脚本：

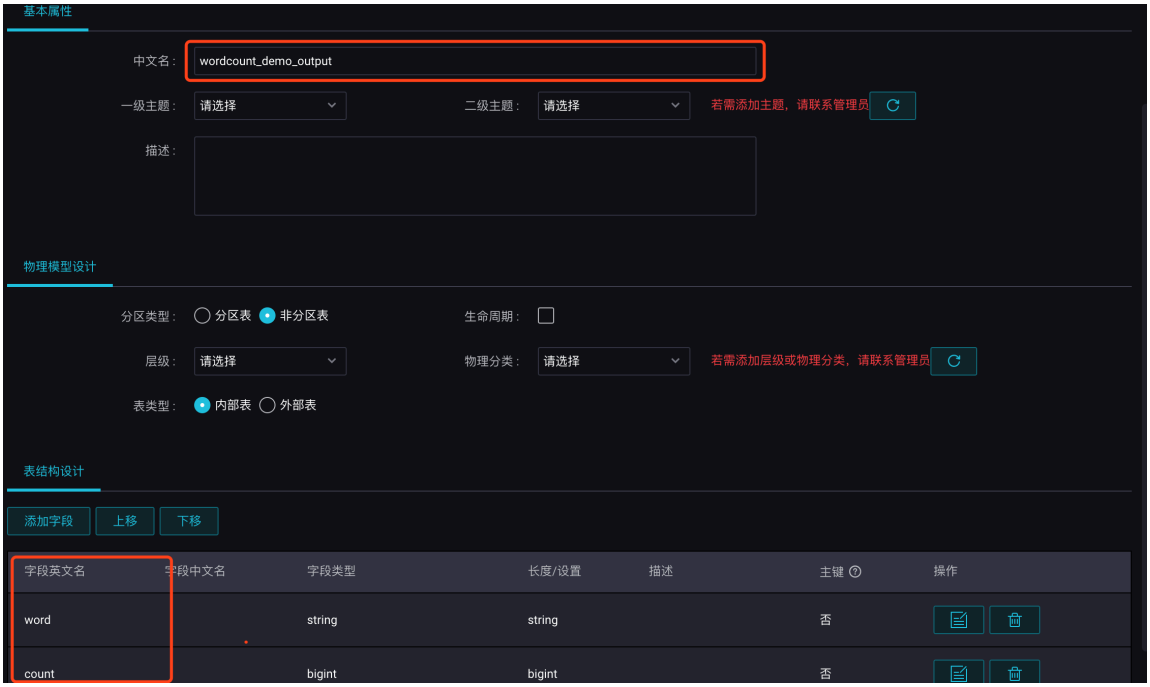
```
# 手工将刚才从发布jar包中的信息填入脚本，脚本完成。
jar -resources WordCountDemo_1.0.0.jar
-classpath http://schedule@{env}inside.cheetah.alibaba-inc.com/
scheduler/res?id=106342493
```

```
com.alibaba.dataworks.mapred.WordCount wordcount_demo_input
wordcount_demo_output
```

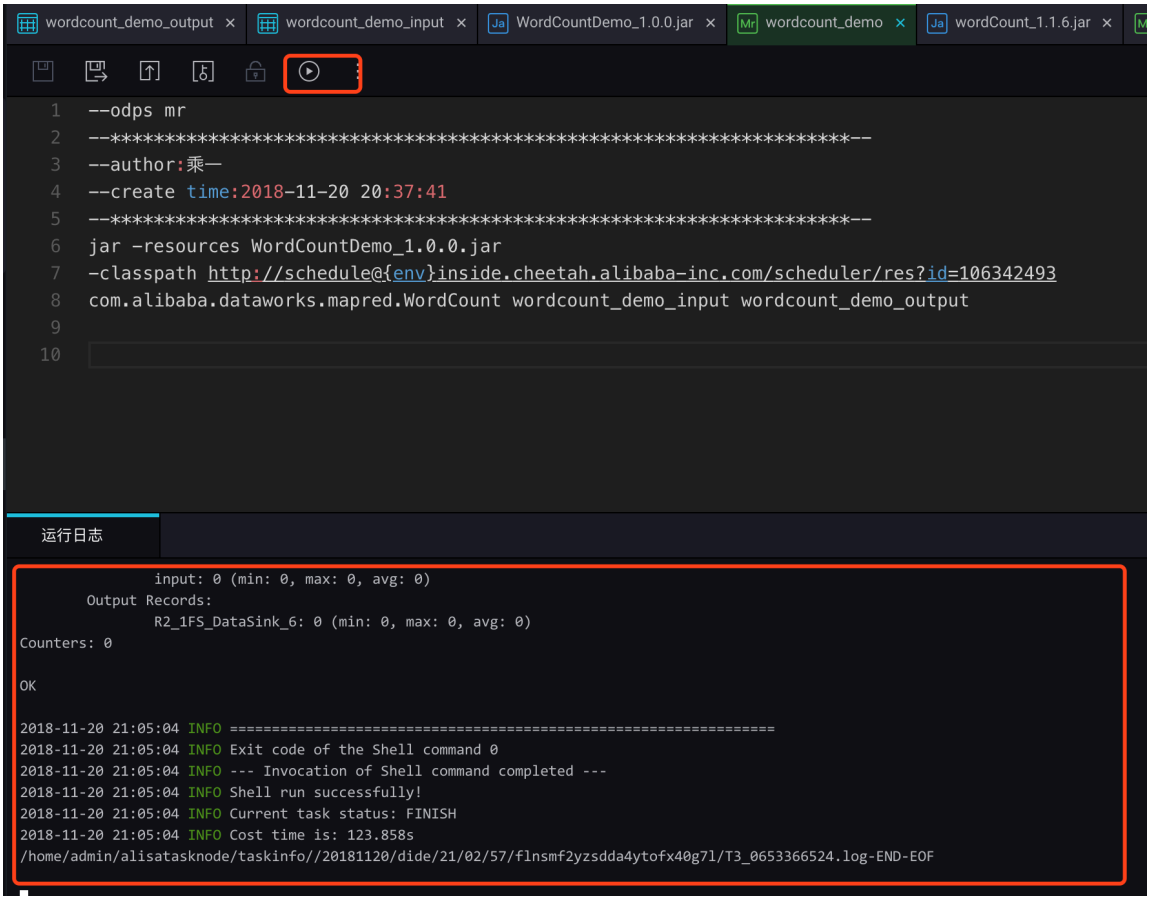
d. 创建测试表和测试数据。







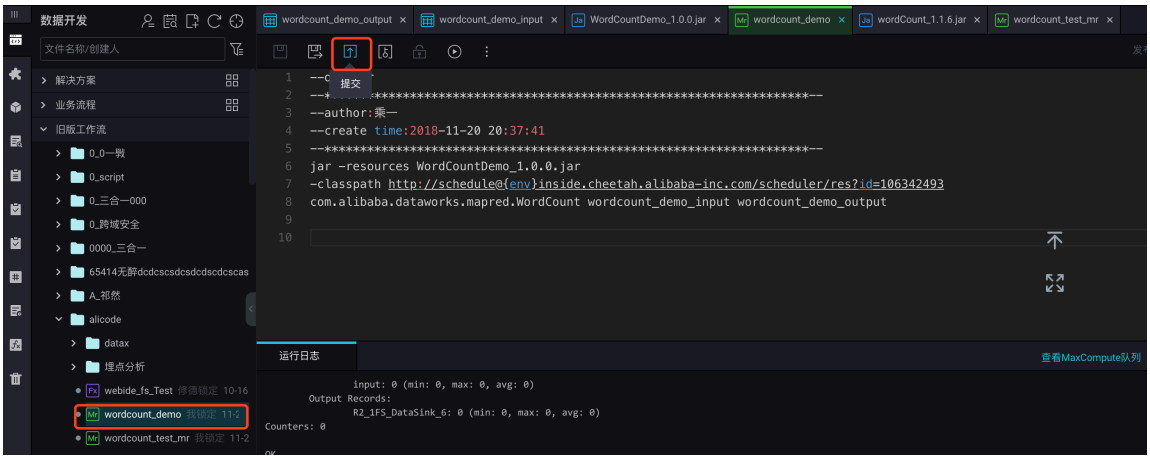
准备好数据后，在开发环境运行脚本。



至此，WordCount在开发环境的测试全部完成。由于WordCount的计算节点、Jar包、输入输出表都在开发环境，所以需要分别发布至正式环境。

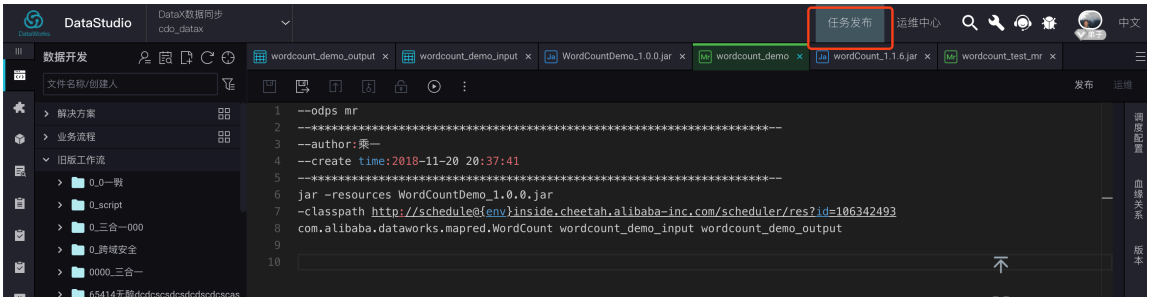
3. 将资源包，数据表，节点分别发布到DataWorks正式环境。

a. 提交计算节点代码。



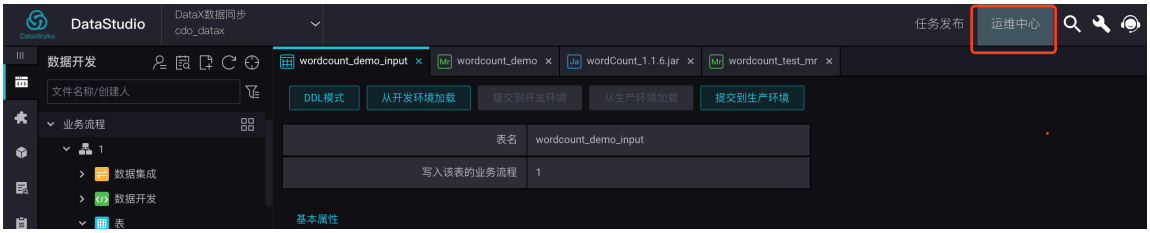
b. 进行发布配置。

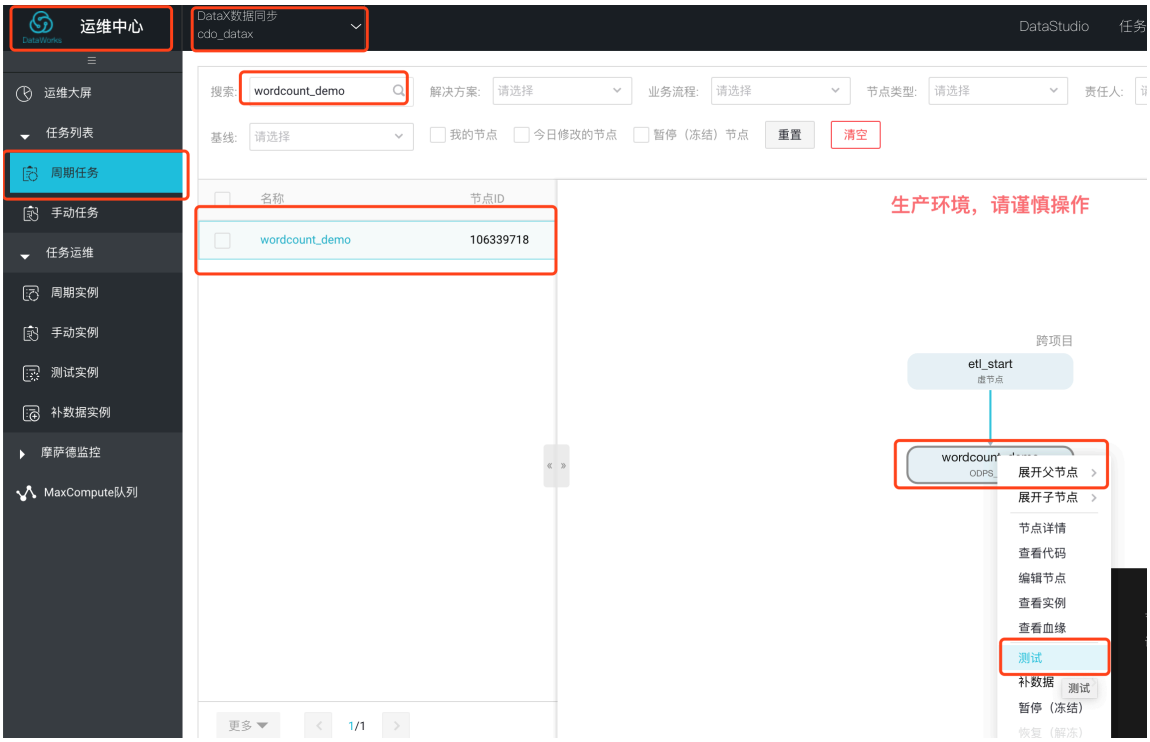
c. 进入发布页面，勾选提交的Jar包和节点进行发布。

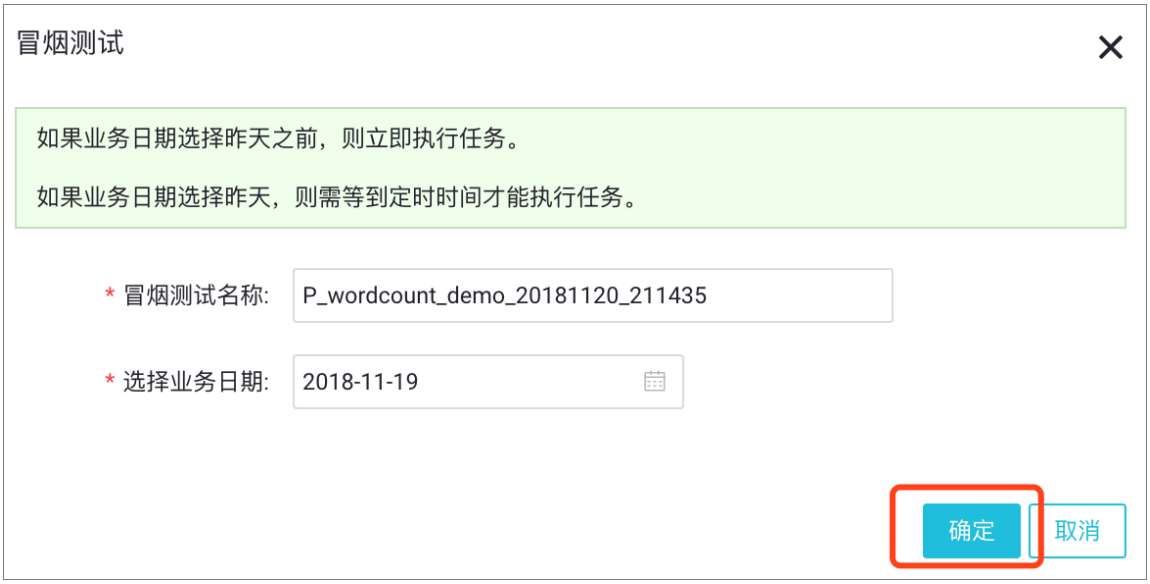


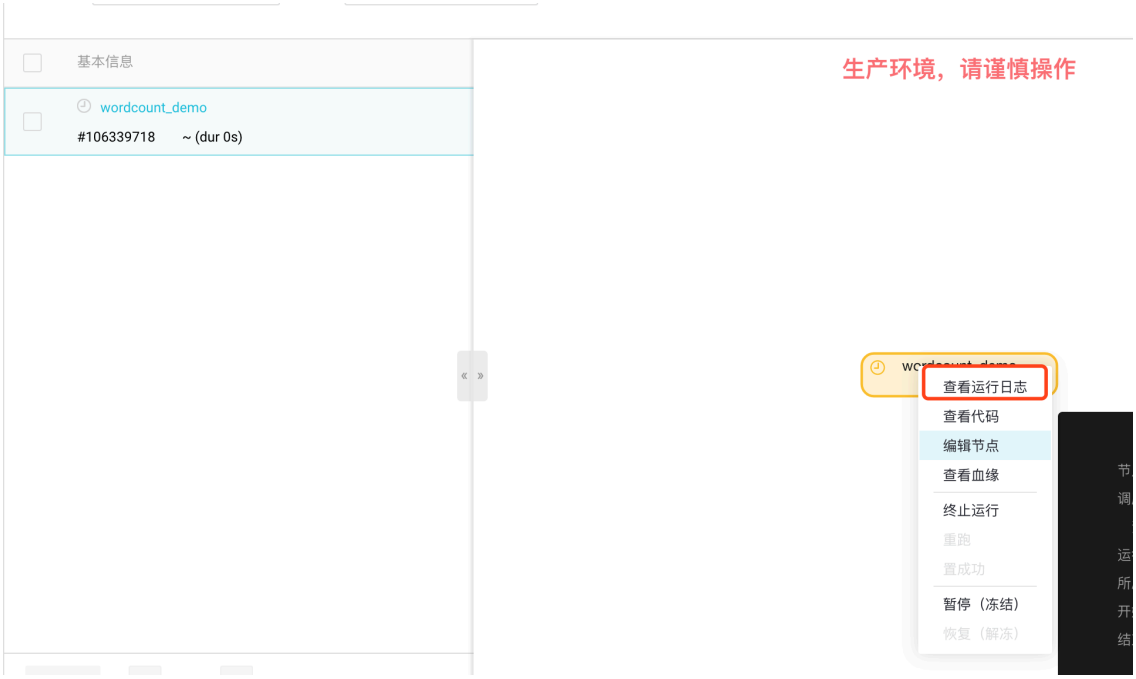
d. 发布数据表。

e. 在运维中心对MapReduce任务进行线上测试。






冒烟测试
如果业务日期选择昨天之前，则立即执行任务。
如果业务日期选择昨天，则需等到定时时间才能执行任务。
* 冒烟测试名称:
P_wordcount_demo_20181120_211435
* 选择业务日期:
2018-11-19
确定
取消


基本信息
wordcount_demo
#106339718 ~ (dur 0s)
生产环境，请谨慎操作
查看运行日志
查看代码
编辑节点
查看血缘
终止运行
重跑
置成功
暂停（冻结）
恢复（解冻）

日志显示成功运行。

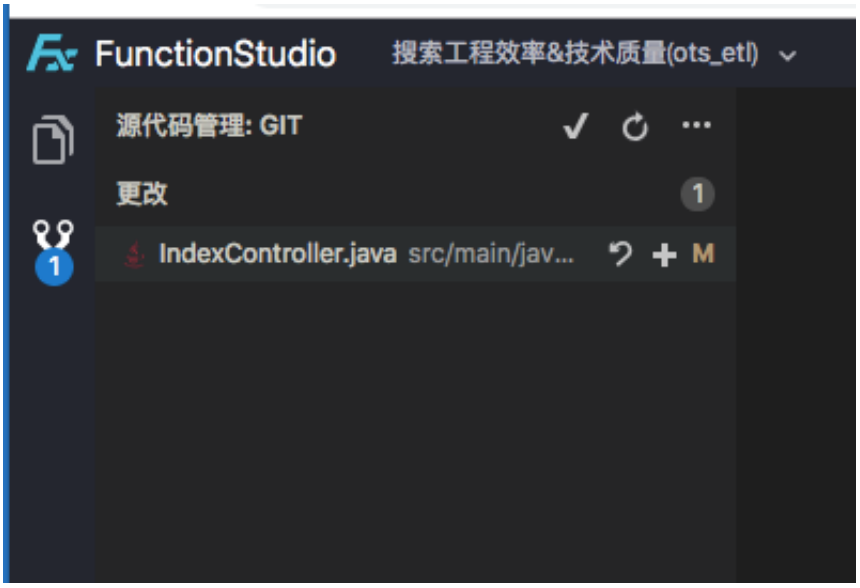
```
...
Inputs:
        cdo_datax.wordcount_demo_input: 0 (0 bytes)
Outputs:
        cdo_datax.wordcount_demo_output: 0 (0 bytes)
M1_cdo_datax_20181120131649978g1nhmk593_LOT_0_0_0_job0:
        Worker Count:1
        Input Records:
                input: 0 (min: 0, max: 0, avg: 0)
        Output Records:
                R2_1: 0 (min: 0, max: 0, avg: 0)
R2_1_cdo_datax_20181120131649978g1nhmk593_LOT_0_0_0_job0:
        Worker Count:1
        Input Records:
                input: 0 (min: 0, max: 0, avg: 0)
        Output Records:
                R2_1FS_DataSink_6: 0 (min: 0, max: 0, avg: 0)
Counters: 0
OK
2018-11-20 21:17:26 INFO ================================================================
2018-11-20 21:17:26 INFO Exit code of the Shell command 0
2018-11-20 21:17:26 INFO --- Invocation of Shell command completed ---
2018-11-20 21:17:26 INFO Shell run successfully!
2018-11-20 21:17:26 INFO Current task status: FINISH
2018-11-20 21:17:26 INFO Cost time is: 52.287s
/home/admin/alisatasknode/taskinfo//20181120/phoenixprod/21/16/29/zm0yabqx3thz9yar6qjt7dks/T3_1546661144.log-END-EOF
```

由此可见，Function Studio可以编码和编译发布代码到DataWorks节点，DataWorks需要手工操作生成计算节点，在开发环境和生产环境分别运行。

14.3.6 Git管理

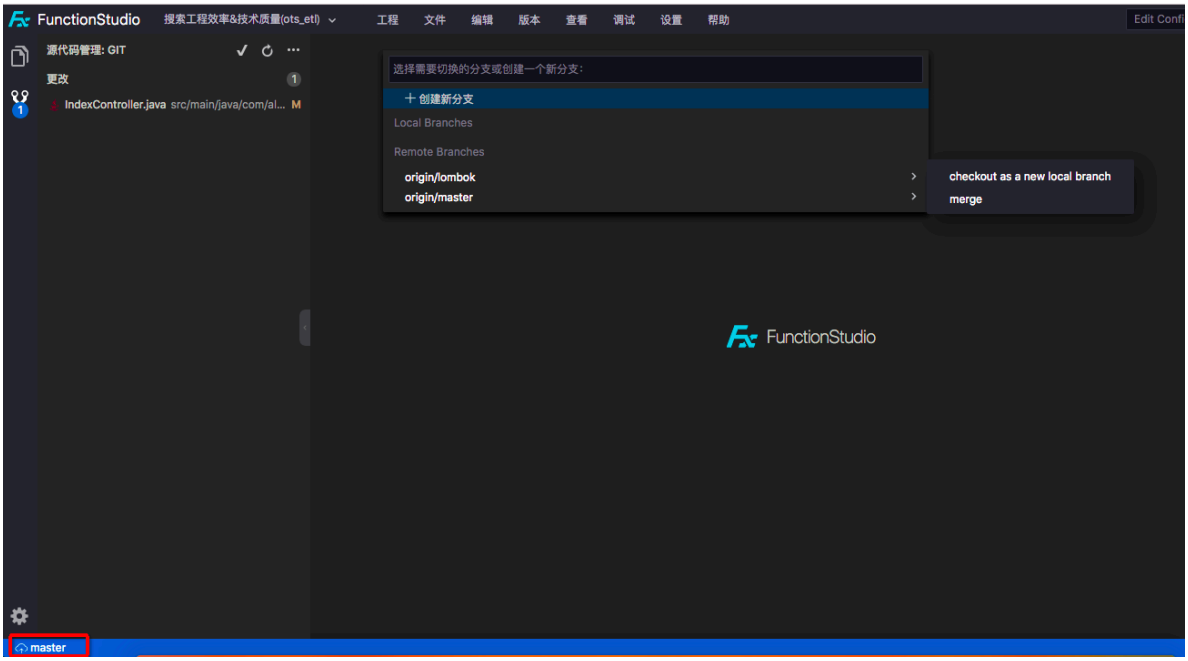
您可将新建的工程和Git进行关联，关联后即可进行常规的Git操作。

1. 选择菜单栏中的版本 > 版本管理。



- 单击文件后的+，可以进行Git add操作。
- 单击√可以进行commit和push操作。
- 单击...可以进行拉取和推送操作。

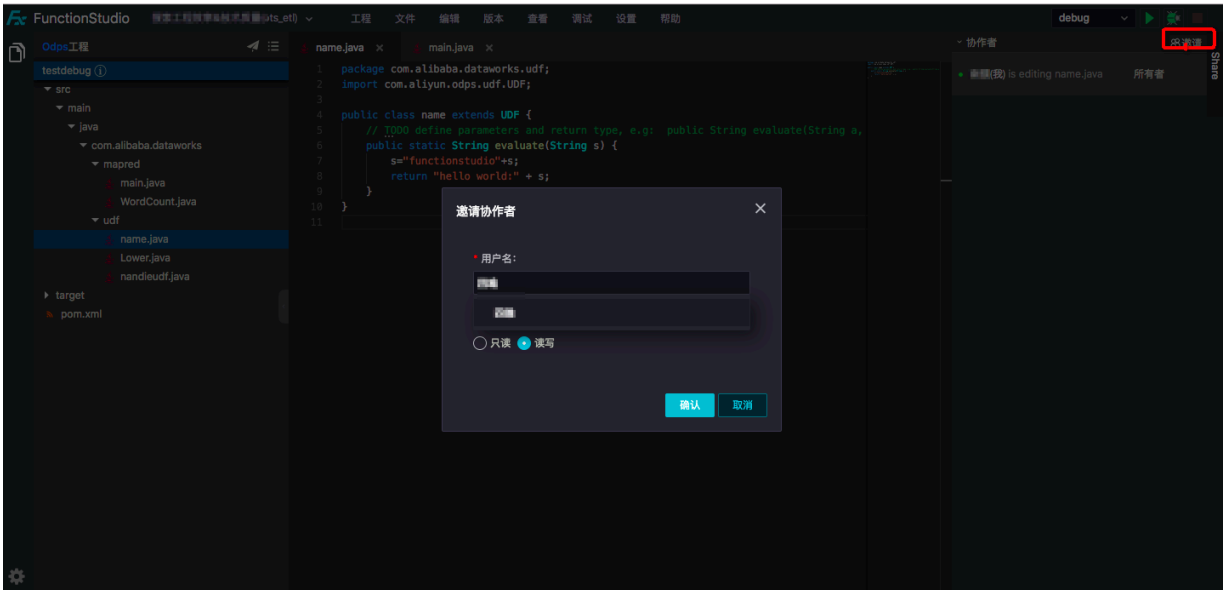
2. 底部的master入口可以关联远程和本地分支，您也可创建分支。



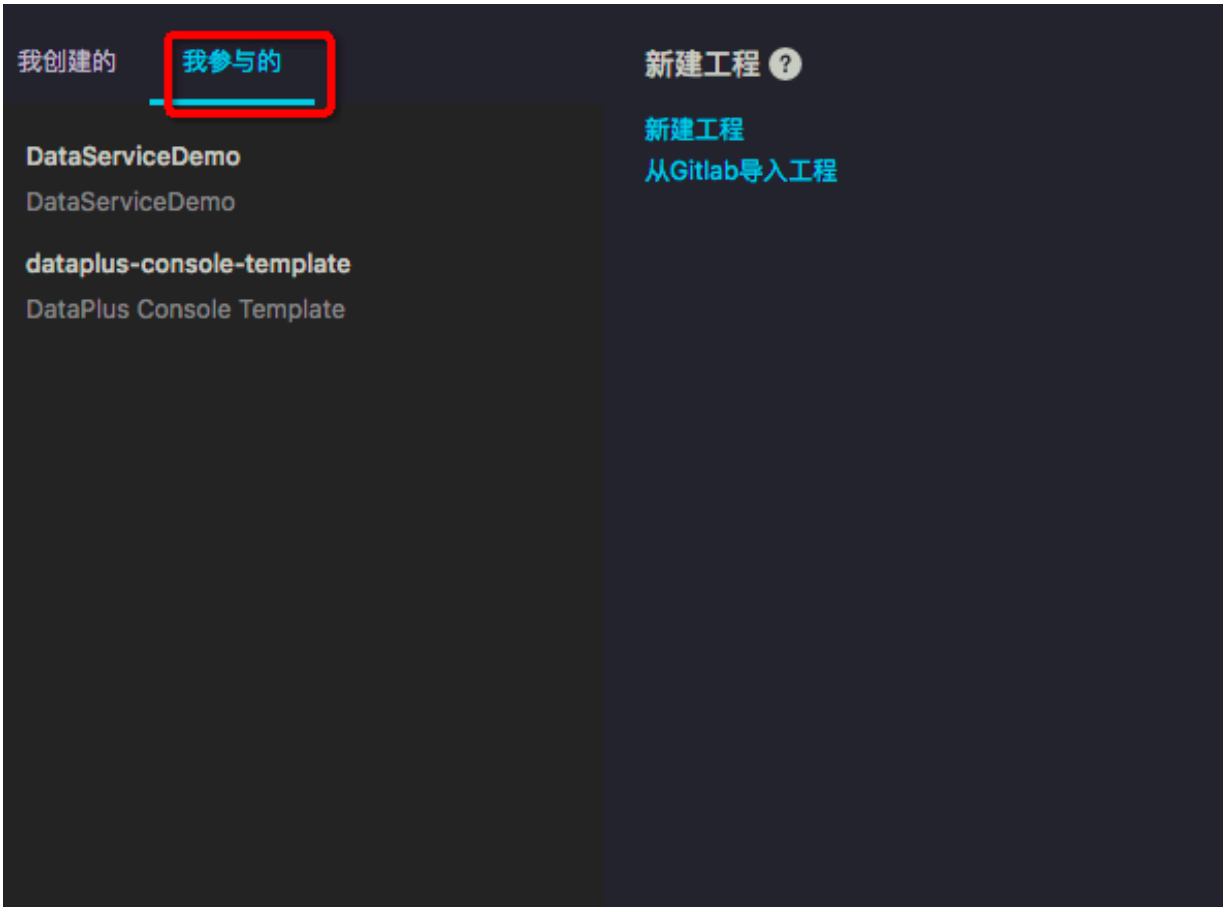
14.3.7 协同编辑

Function Studio支持邀请多人协同编辑同一工程的同一文件。

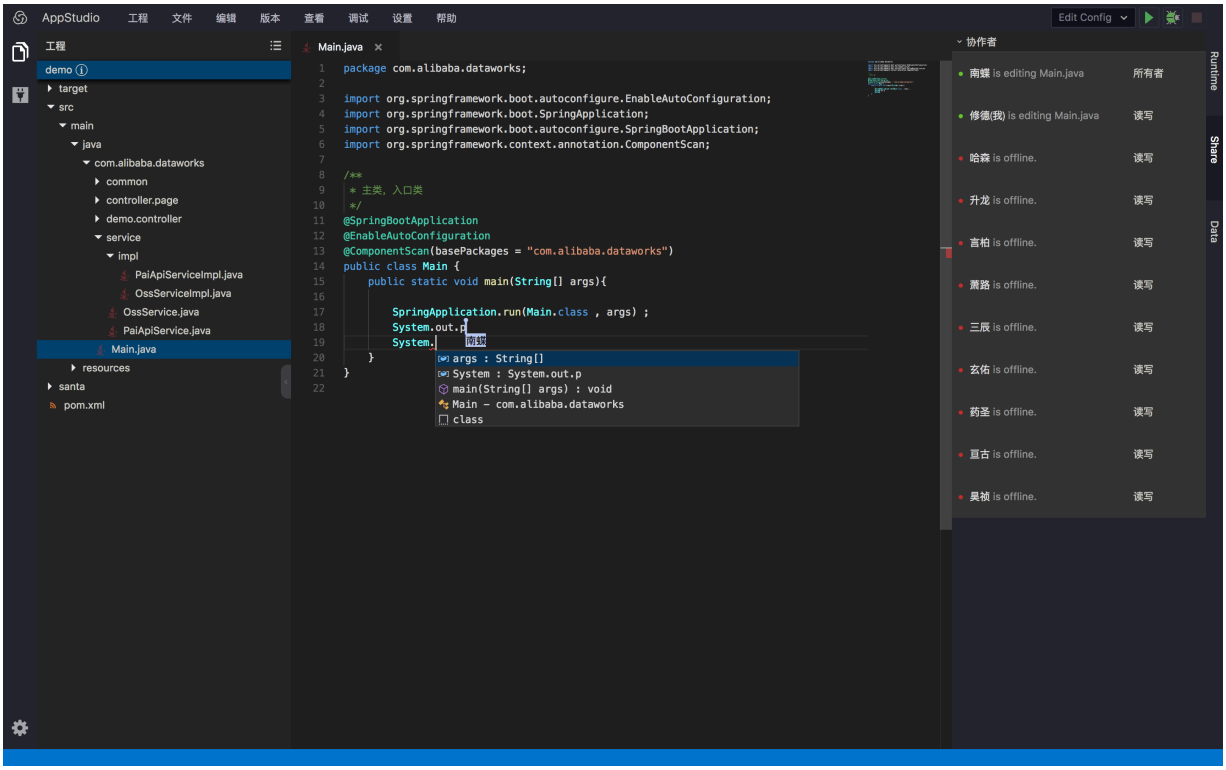
您可单击右上角的share，邀请别人协同编辑。



您可进入我参与的工程列表，查看共享下的工程。



多人同时编辑同一工程的同一文件的效果。



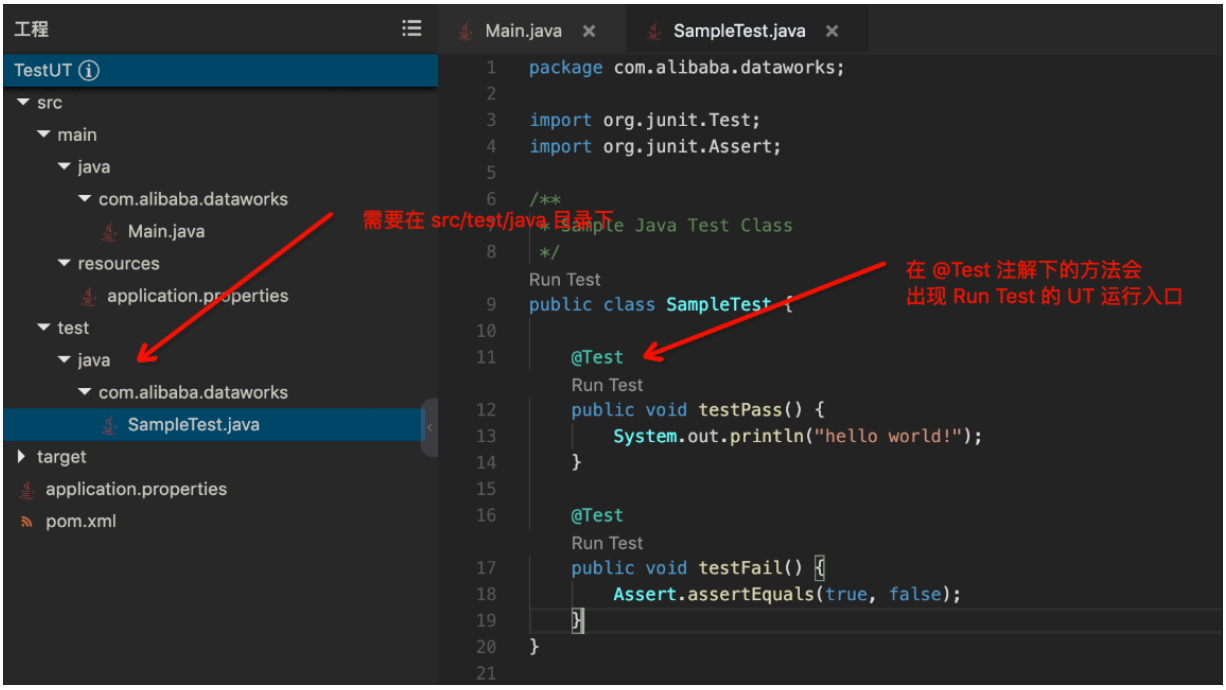
14.3.8 UT测试

Function Studio目前支持UnitTest的运行，包括UT测试的入口检测、运行UT代码和展示运行结果三大功能。

UT测试的入口检测

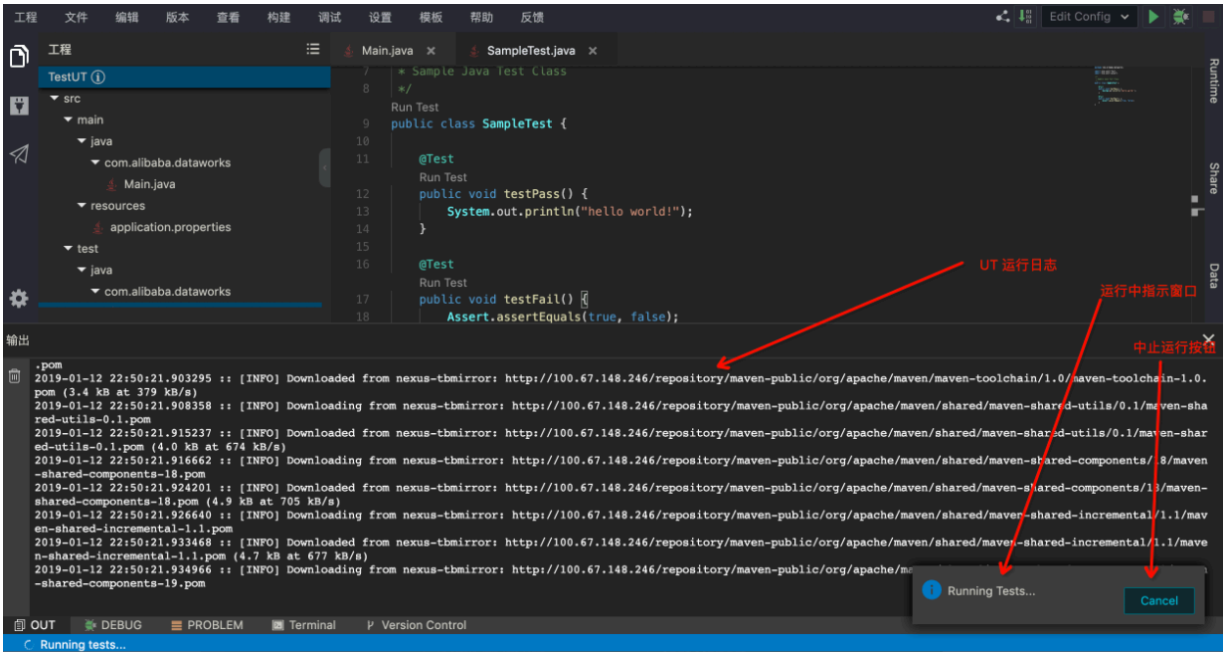
说明:

- UT类需要写在*src/test/java*目录下，如果不在此目录下的Java UT类文件，将无法被正常识别成Java UT类。
- Java类创建完成后，在对应的测试用例方法上添加`org.junit.Test`的`@Test`注解即可。

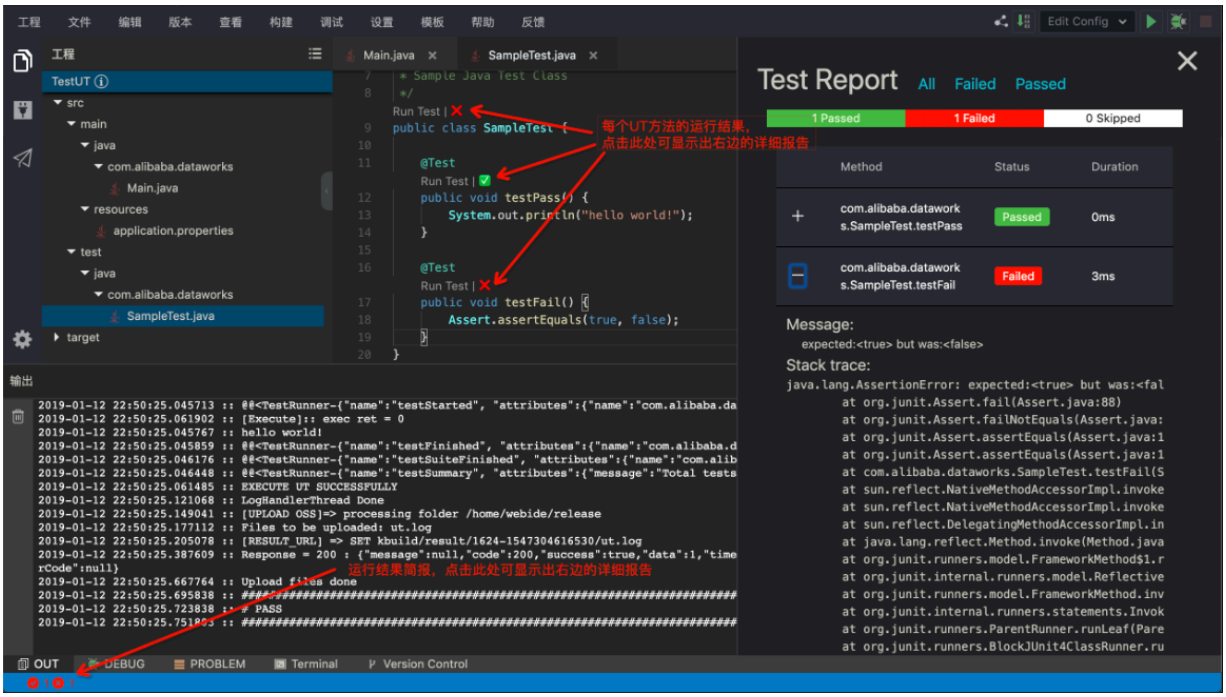


运行UT代码

单击Run test，运行UT代码。



展示运行结果



14.3.9 全文搜索

本文将为您介绍如何在Function Studio中进行全文搜索。

Function Studio支持全文搜索功能，具体操作如下所示。

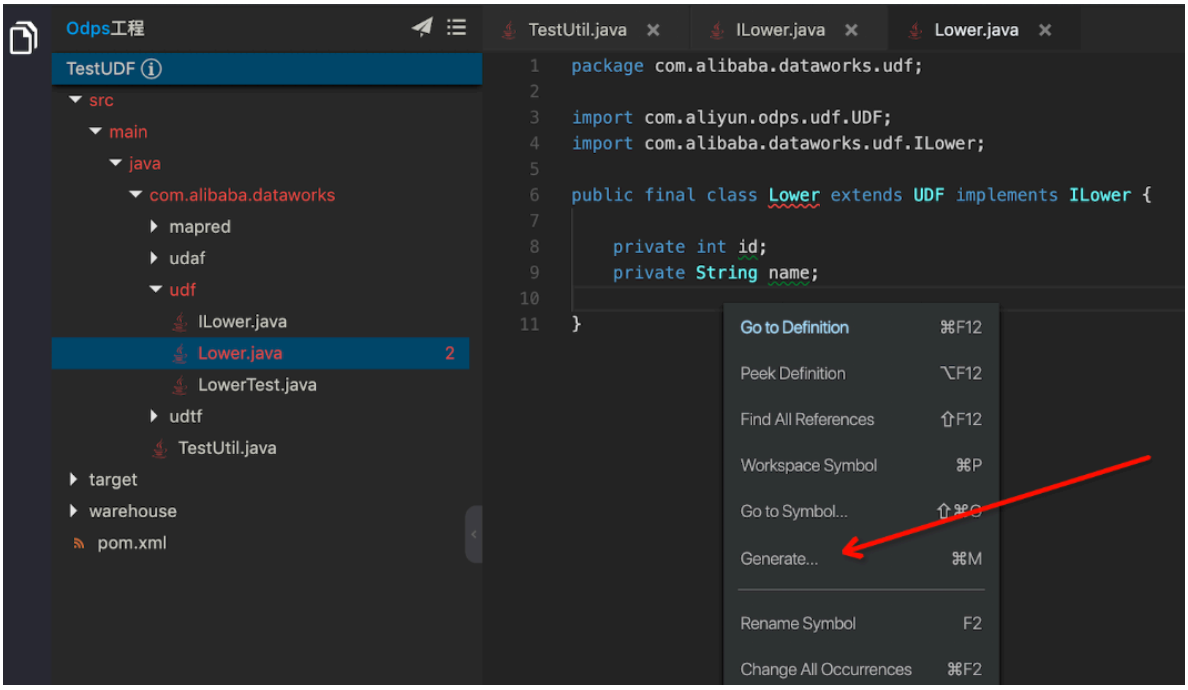
14.3.10 自动代码生成

Function Studio目前在Java中可支持一些常用的代码生成功能，目前已经支持Java类的构造函数（Constructor）、Getter函数、Setter函数、该类所继承父类的Override方法生成和所要实现的接口方法的生成。

功能入口

目前有两个Java代码生成的入口。

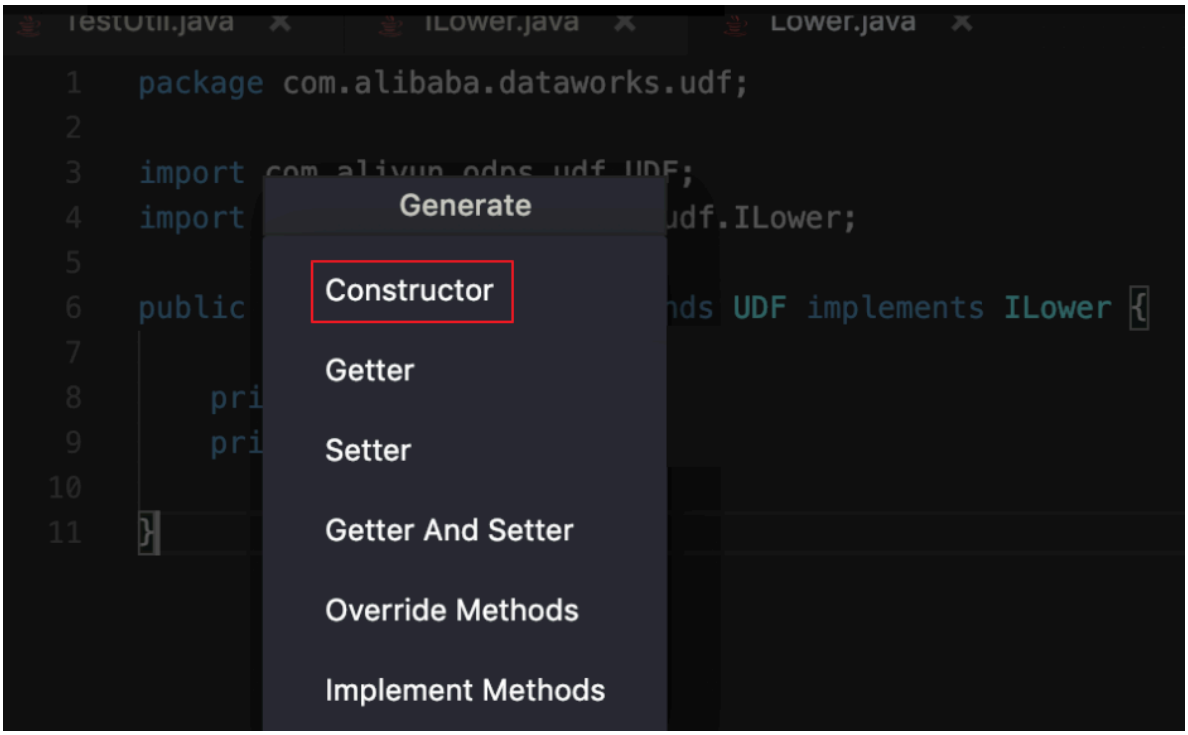
- 鼠标右键，选择Generate。



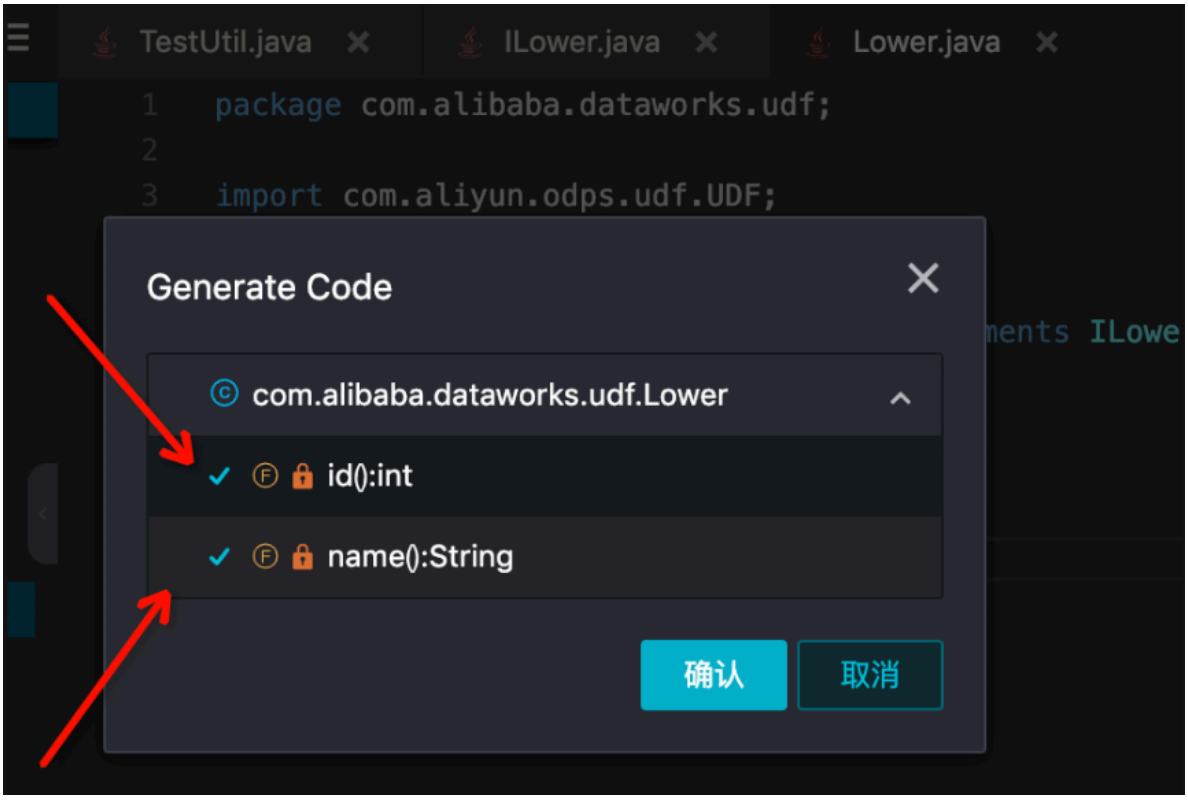
- 通过快捷键cmd+m进入。

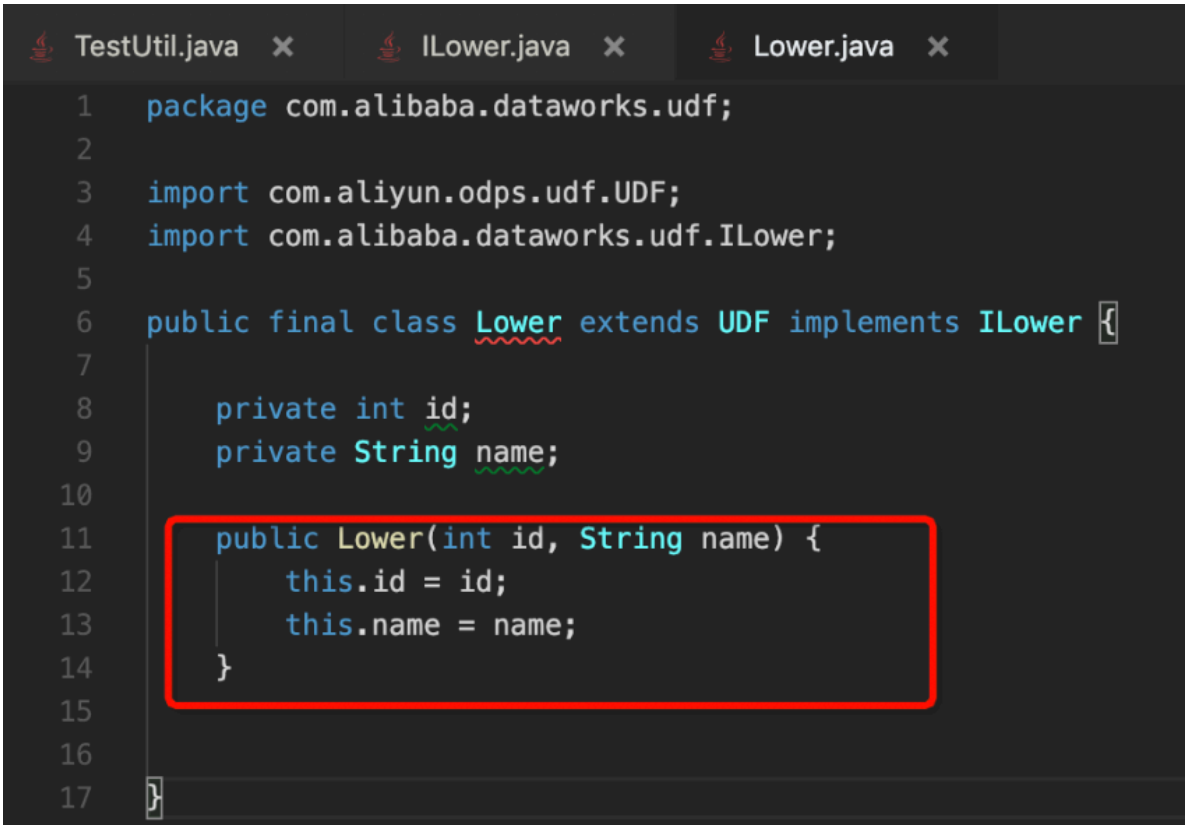
Constructor

1. 进入Generate Code面板，选择Constructor。



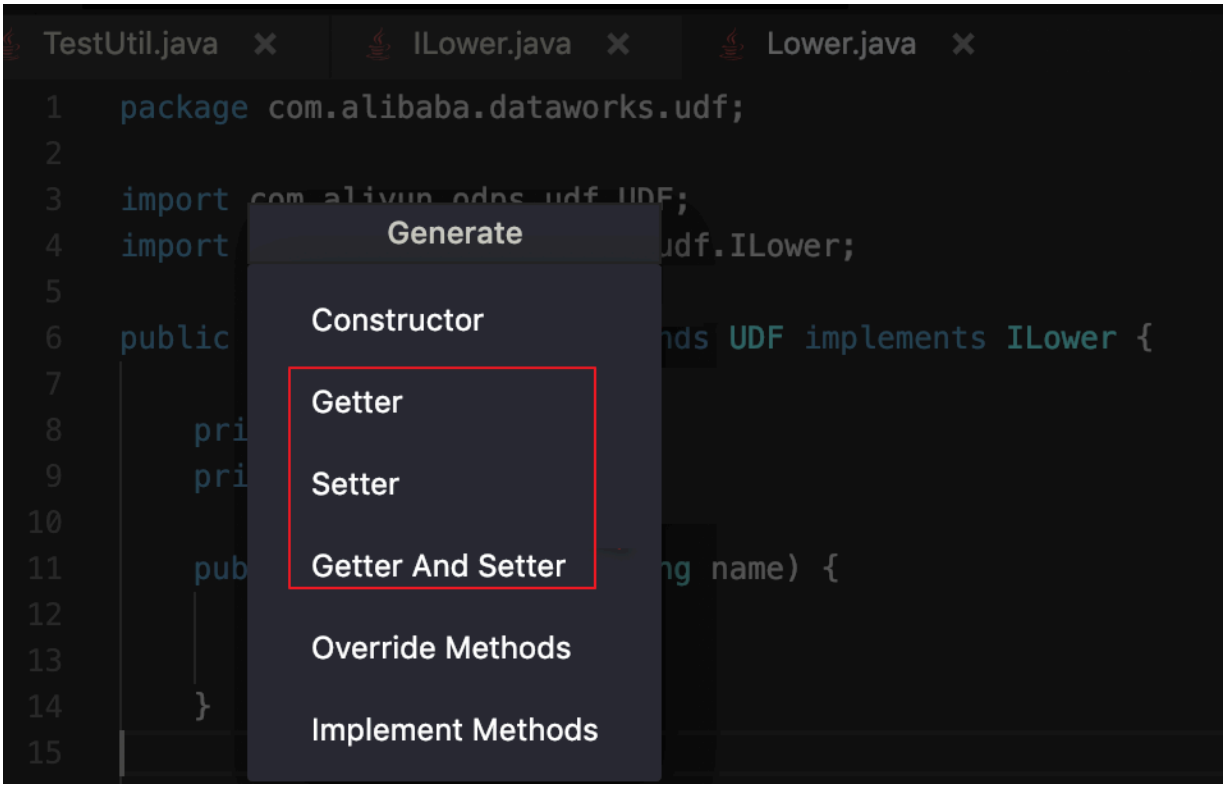
2. 选择构造函数中要包含的字段，即可生成包含这些字段初始化语句的构造函数。





Getter&Setter

您可参见Constructor的生成方式来生成Getter和Setter函数。

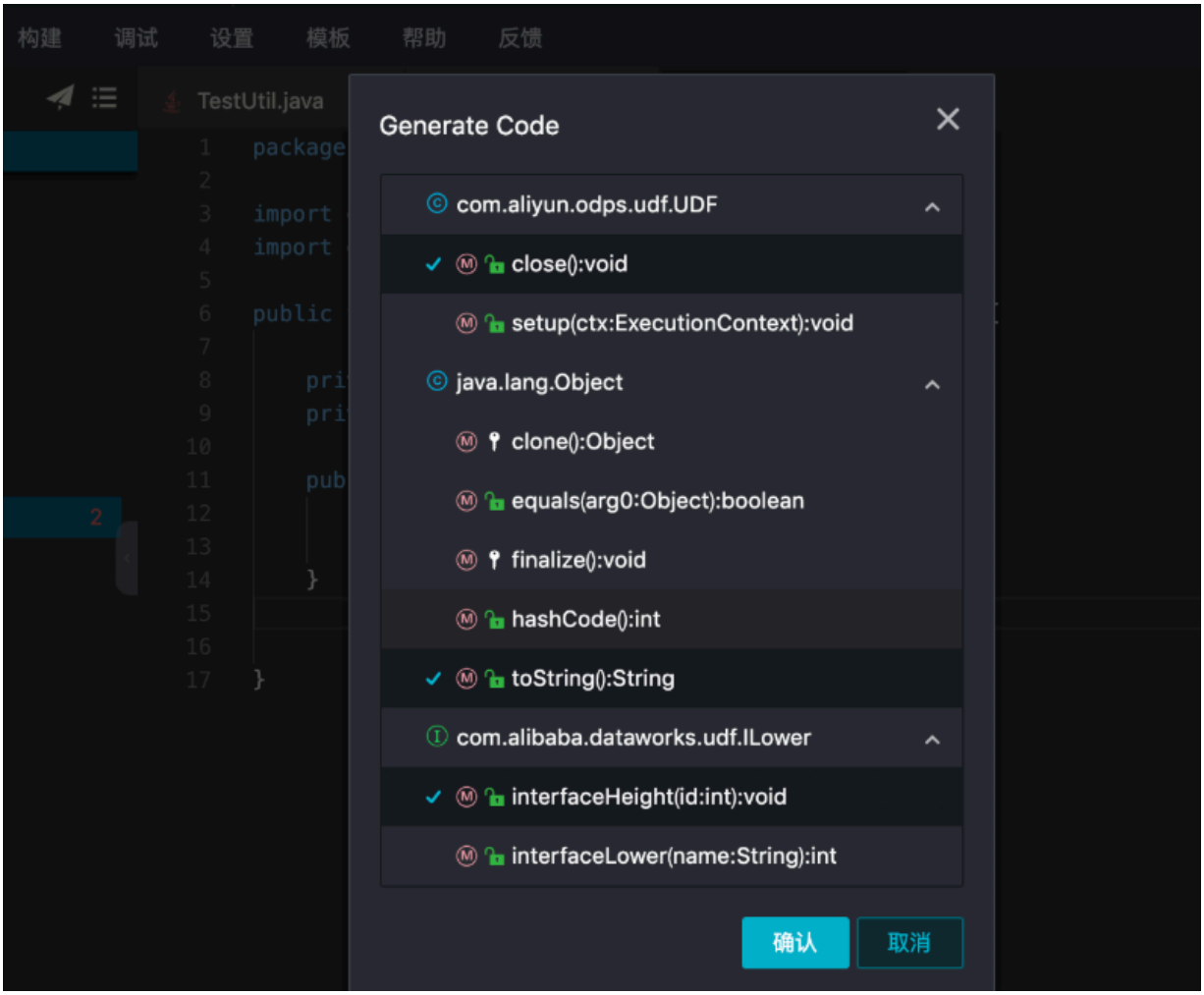


说明:

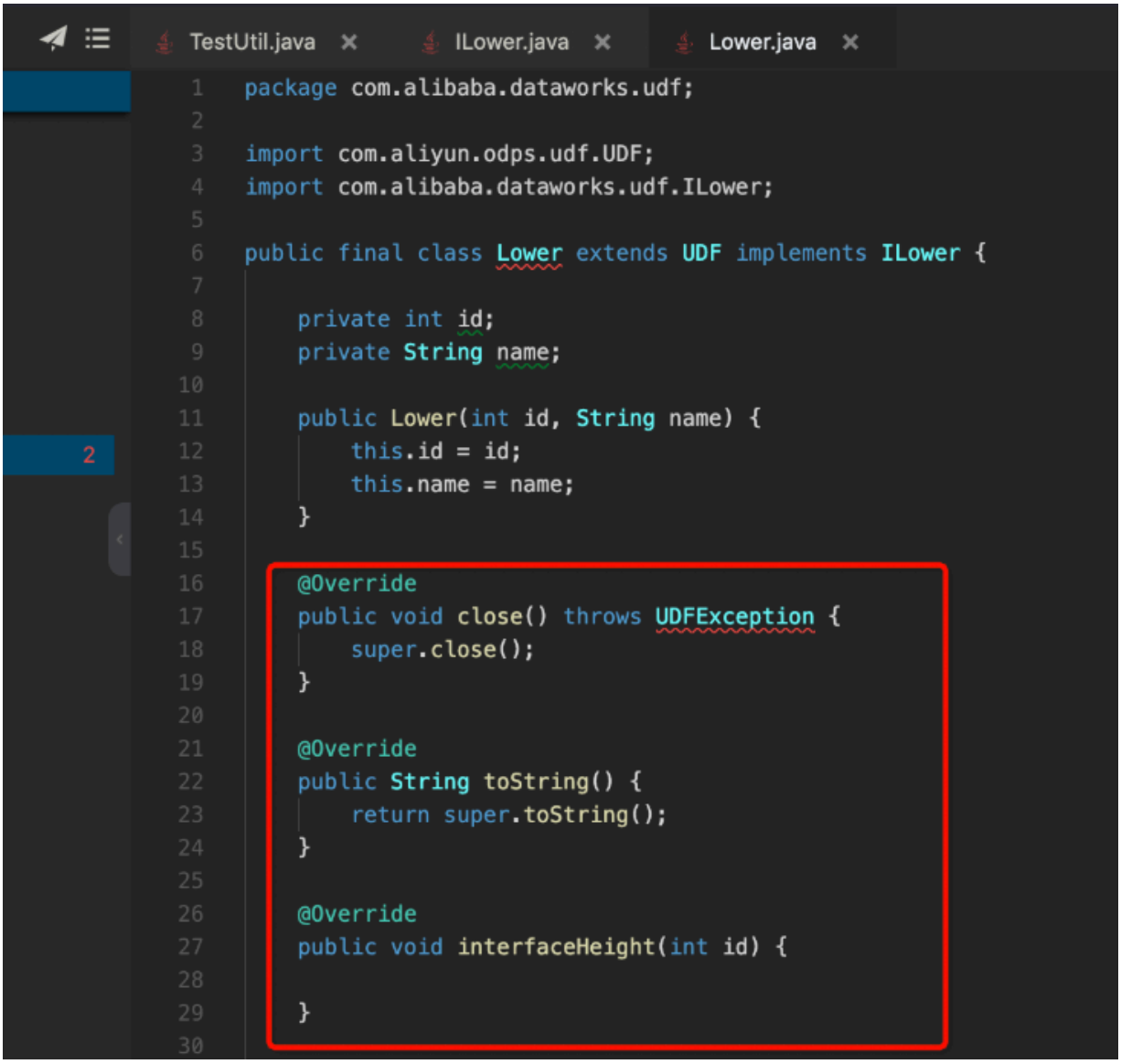
如果该Java类没有任何字段，或者该Java类已经被lombok的@data注解覆盖，则没有上图中的三个选项，因为此时该类不需要生成Getter或Setter函数。

Override Methods

选择生成Override Methods的一级菜单后，在二级菜单中会罗列所有可以Override的方法。



选择后即可生成对应的方法。

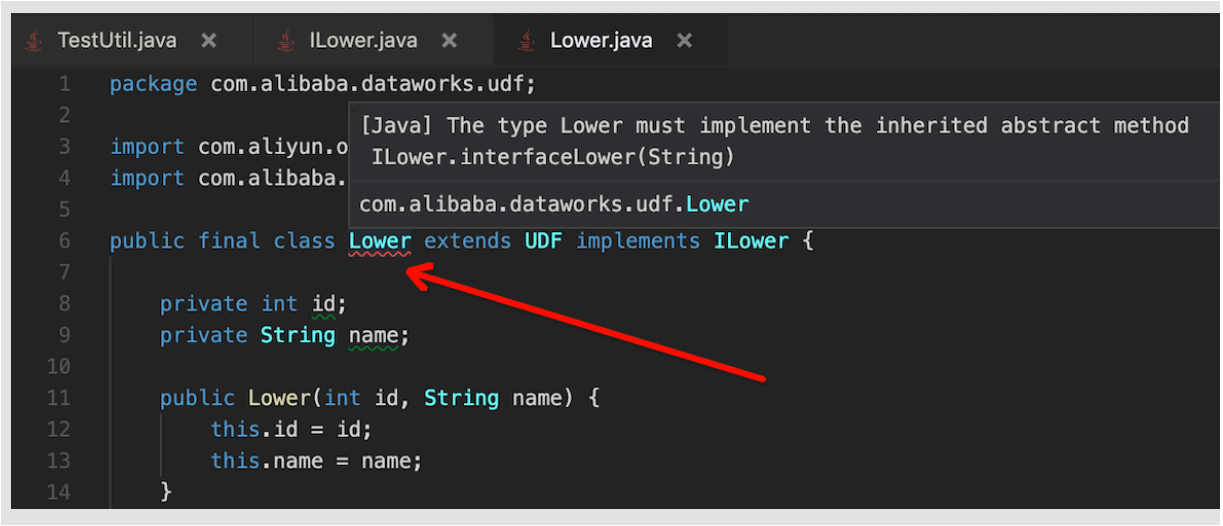


Implement Methods

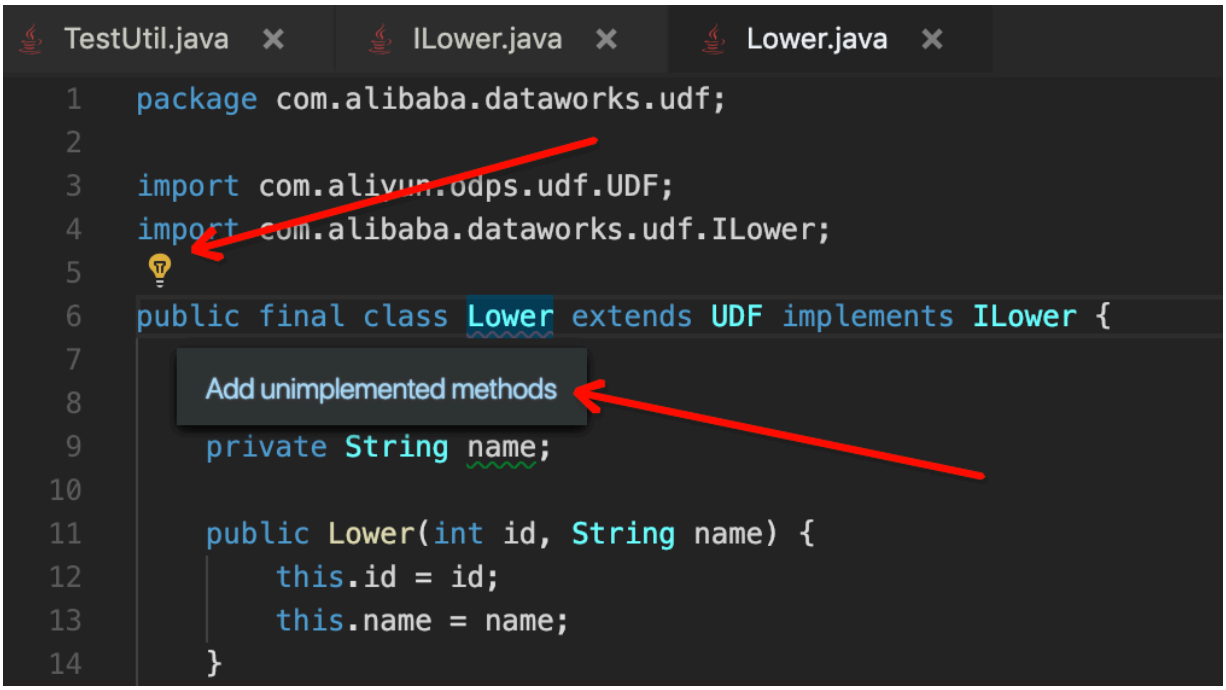
您可参见Override Methods的生成方式来生成Implement Methods。

说明:

Java中如果不实现接口的方法，本身会有语法问题，则会有红色波浪线。



您不仅可以使用上文介绍的Generate功能，也可以使用智能提示功能达到同样的效果。



```java
public final class Lower extends UDF implements ILower {

    private int id;
    private String name;

    public Lower(int id, String name) {
        this.id = id;
        this.name = name;
    }

    @Override
    public int interfaceLower(String name) {
        return 0;
    }

    @Override
    public void interfaceHeight(int id) {

    }
```

15 数据保护伞

15.1 进入数据保护伞

本文将为您介绍如何进入、授权并开通数据保护伞功能。

进入引导页面

登录DataWorks控制台，单击左上角导航栏中的数据保护伞，即可进入数据保护伞页面。

如果您是首次登录数据保护伞，会出现引导页面为您介绍数据保护伞的核心功能及使用流程，帮助您对数据保护伞进行初步了解。

单击立即体验即可进入数据保护伞授权页面。

说明:

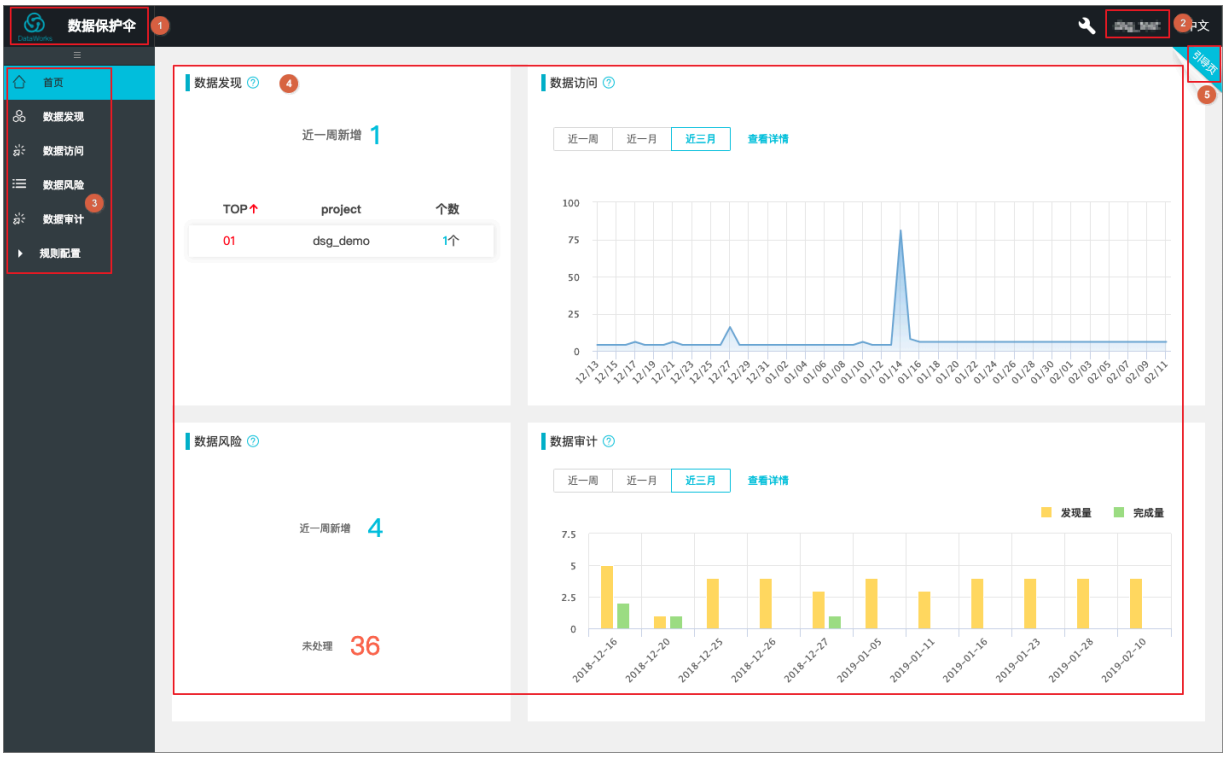
如果租户管理员已授权，则直接进入数据保护伞首页。

进入授权页面

只有租户管理员才能进行授权，开通数据保护伞。

登录数据保护伞

登录数据保护伞平台，首页如下所示：



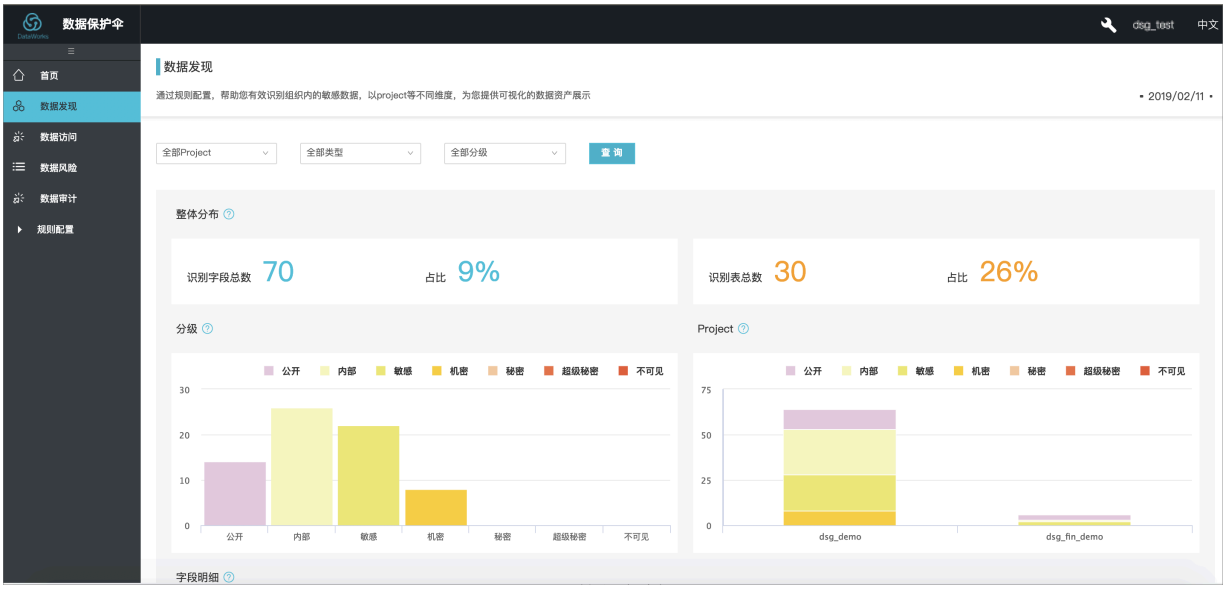
序号	名称	说明
1	功能菜单栏	当前用户有权可见的功能模块，包括数据开发、数据质量、数据集成、数据服务、运维中心、数据管理和数据保护伞。

序号	名称	说明
2	用户信息	当前登录用户，可查看并编辑用户信息，包括邮箱、手机、AccessKeyID和AccessKeySecret。
3	导航栏	对应功能菜单的导航栏，不同的功能模块对应不同的左侧导航栏。
4	数据保护伞首页	· 该租户近一周新识别数据。 · 所有识别数据近一周、近一月、近三月的访问趋势。 · 近一周新增数据风险量。 · 所有风险近一周、近一月、近三月的发现量以及完成量。
5	引导页切换	单击引导页即可切换到引导页面，查看产品的信息。

15.2 数据发现

数据安全管理员在完成敏感数据规则配置T+1后，即可在识别数据分布中查看数据分布情况。

数据分布分为整体分布、按等级分布和字段明细，您可根据自己的查询需要，按照Project、规则名、规则类型、风险等级（即分级）等进行过滤选择。



字段明细

Project	规则名(全部)	类型(全部)	分级(全部)	表数量	字段数量	操作
dsg_demo	姓名	姓名	内部	8	14	明细
dsg_demo	邮箱	邮箱	敏感	7	11	明细
dsg_demo	手机号	手机号	内部	8	10	明细
dsg_demo	地址	地址	敏感	6	8	明细
dsg_demo	身份证号	身份证号	机密	4	6	明细
dsg_demo	mac地址	mac地址	公开	4	4	明细
dsg_demo	车牌号	车牌号	公开	3	3	明细
dsg_demo	IP	IP	公开	2	2	明细
dsg_fin_demo	邮箱	邮箱	敏感	1	2	明细
dsg_demo	交易金额	交易金额	机密	2	2	明细

1 2

15.3 数据访问

本文将为您介绍数据访问的访问行为和导出行为。

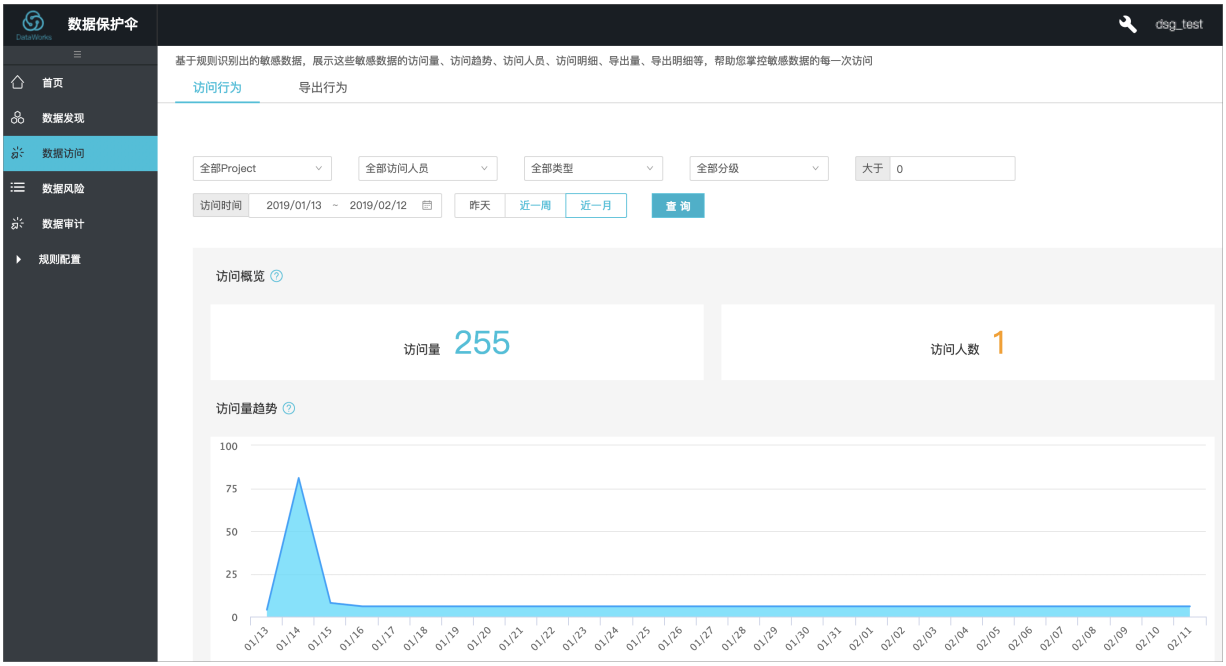
数据访问包括访问行为和导出行为两个模块。

- 访问行为：包含Create、Insert操作，但不包括访问失败的行为。
- 导出行为：数据从MaxCompute导出的行为。

访问行为

数据安全管理员在完成敏感数据规则配置T+1后，即可在数据访问行为中查看数据使用情况，包括访问概览、访问趋势以及访问明细。

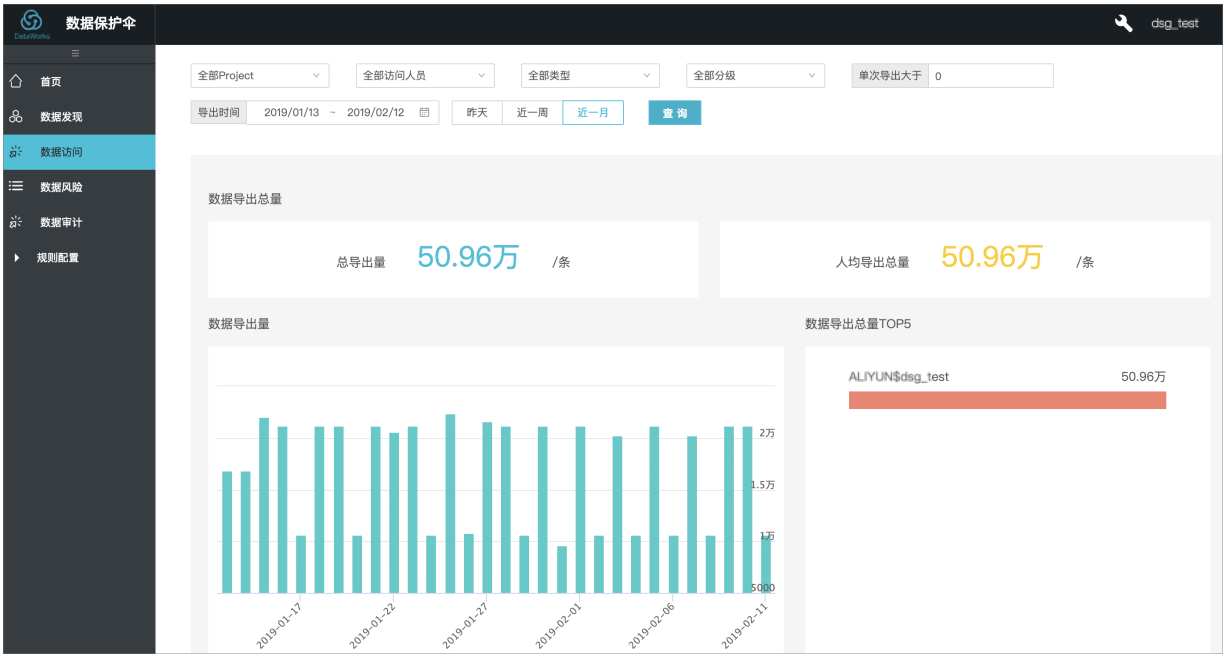
您可根据自己的查询需要，按照Project、规则名、规则类型、风险等级（即分级）、访问人员等进行过滤选择。



风险识别_数据导出

数据安全管理员在完成敏感数据规则配置T+1后，即可在数据导出中查看用户从MaxCompute中把数据导出到外部的情况，包括数据导出总量、TOP导出用户和导出明细。

您可根据自己的查询需要，按照规则名、规则类型、导出量等进行过滤选择。



单次导出大于 0

导出时间 2019/01/13 ~ 2019/02/12

数据导出总量

总导出量 50.96万 /条

人均导出总量 50.96万 /条

数据导出量

数据导出总量TOP5

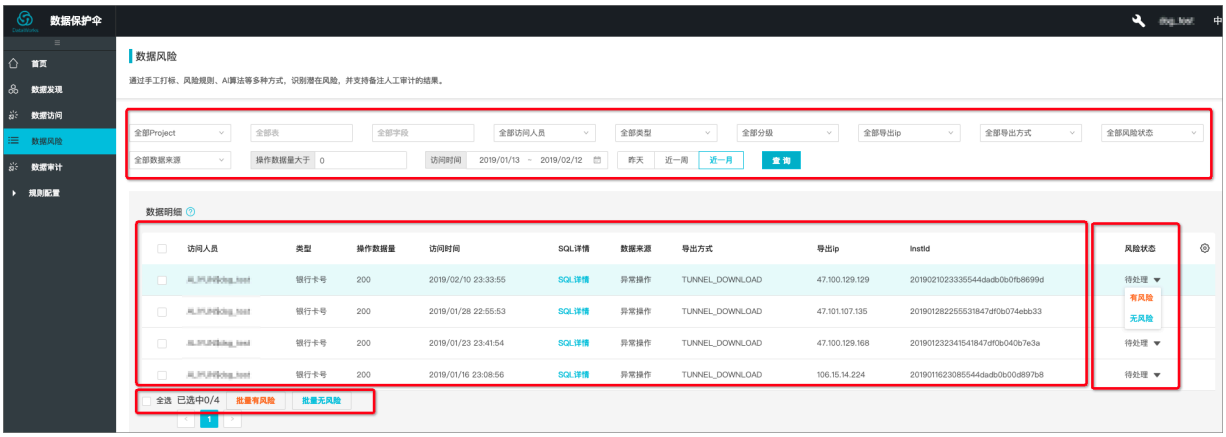
ALIYUN$dsg_test 50.96万

数据导出明细

导出人(全部)	导出账号(全部)	导出ip(全部)	IP所在地(全部)	导出量	导出方式	导出时间	操作
ALIYUN$dsg_test	ALIYUN$dsg_test	47.100.129.87	数据保护伞扫描任务	200	Tunnel下载	2019/02/11 15:19:55	明细
ALIYUN$dsg_test	ALIYUN$dsg_test	47.100.129.87	数据保护伞扫描任务	200	Tunnel下载	2019/02/11 15:19:54	明细
ALIYUN$dsg_test	ALIYUN$dsg_test	47.100.129.87	数据保护伞扫描任务	200	Tunnel下载	2019/02/11 15:19:54	明细
ALIYUN$dsg_test	ALIYUN$dsg_test	47.100.129.87	数据保护伞扫描任务	200	Tunnel下载	2019/02/11 15:19:54	明细
ALIYUN$dsg_test	ALIYUN$dsg_test	47.100.129.87	数据保护伞扫描任务	200	Tunnel下载	2019/02/11 15:19:54	明细
ALIYUN$dsg_test	ALIYUN$dsg_test	47.100.129.87	数据保护伞扫描任务	200	Tunnel下载	2019/02/11 15:19:54	明细
ALIYUN$dsg_test	ALIYUN$dsg_test	47.100.129.87	数据保护伞扫描任务	200	Tunnel下载	2019/02/11 15:19:53	明细
ALIYUN$dsg_test	ALIYUN$dsg_test	47.100.129.87	数据保护伞扫描任务	200	Tunnel下载	2019/02/11 15:19:53	明细

15.4 数据风险

数据风险页面提供通过手工风险数据识别、风险识别管理（风险规则配置识别、AI识别）产生的风险数据清单，同时可对这些风险数据进行审计备注。

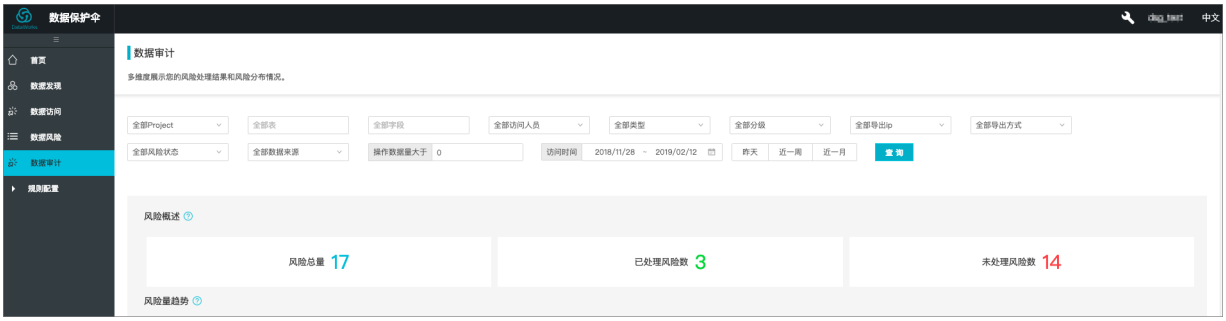


页面说明如下：

- 查询风险数据条件：可供筛选的条件包括Project、表名、字段、访问人员、规则类型、规则名、分级、导出IP、导出风险、风险状态和风险数据类型。
- 风险数据明细：可根据查看指标的需要在标题栏处的设置按钮中选择审计备注，对风险数据进行有/无风险标注，支持增加标签、增加详细备注信息。
- 批量审计处理：分为批量有/无风险标注以及详细信息备注。

15.5 数据审计

数据审计页面是数据风险统计信息的汇总，包括风险数据概览、每日风险趋势和风险维度分析。



15.6 规则配置

本文将为您介绍如何定义敏感数据，新建规则并进行配置。

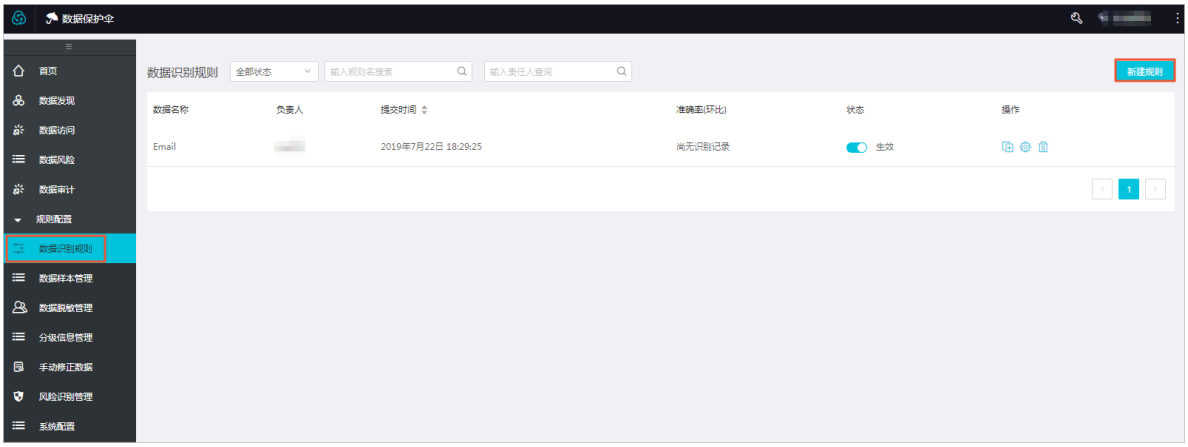
定义敏感数据

数据安全管理员的操作步骤如下所示：

1. 单击左上角的图标，选择全部产品 > 数据保护伞。

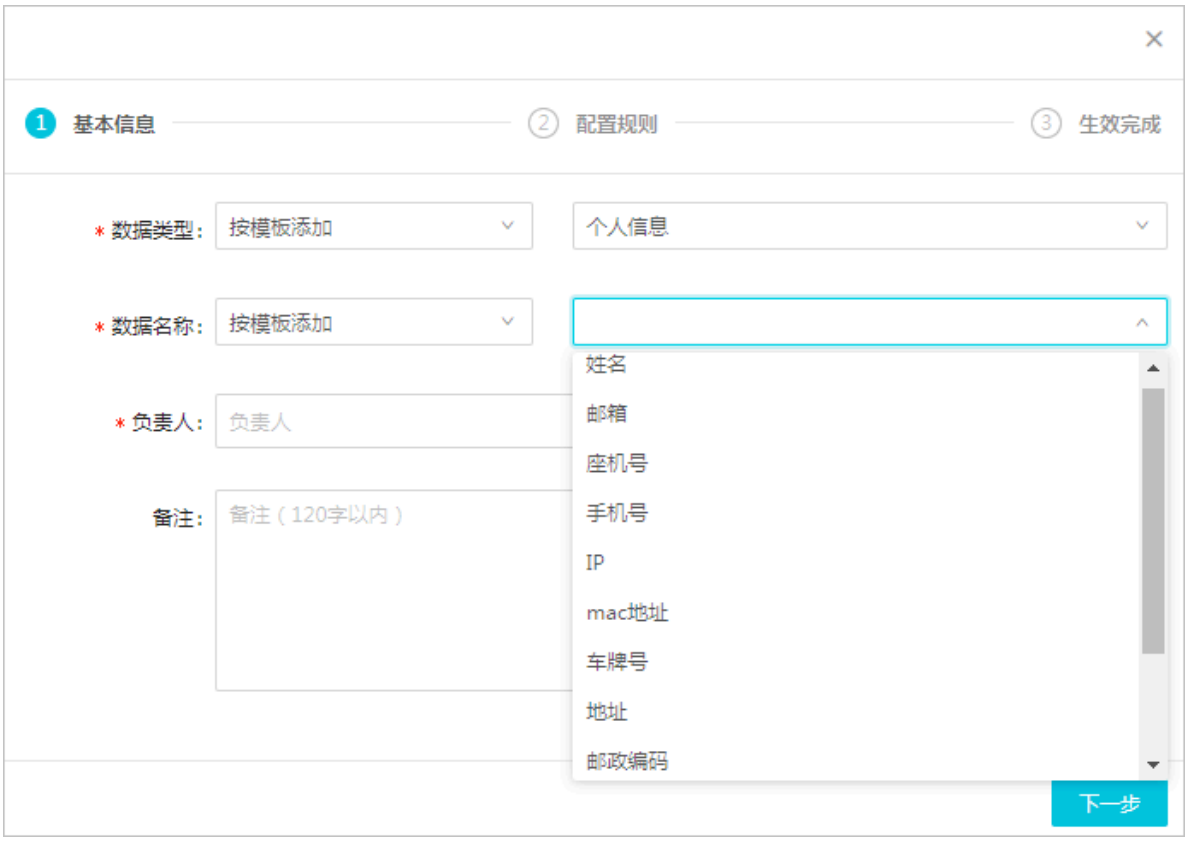
2. 单击数据保护伞产品页面的立即体验，即可进入数据保护伞控制台。

3. 导航至规则配置 > 识别数据规则，单击新建规则。



4. 填写对话框中的基本信息，单击下一步。

您可以通过按模板添加和自定义添加两种方式新建规则。



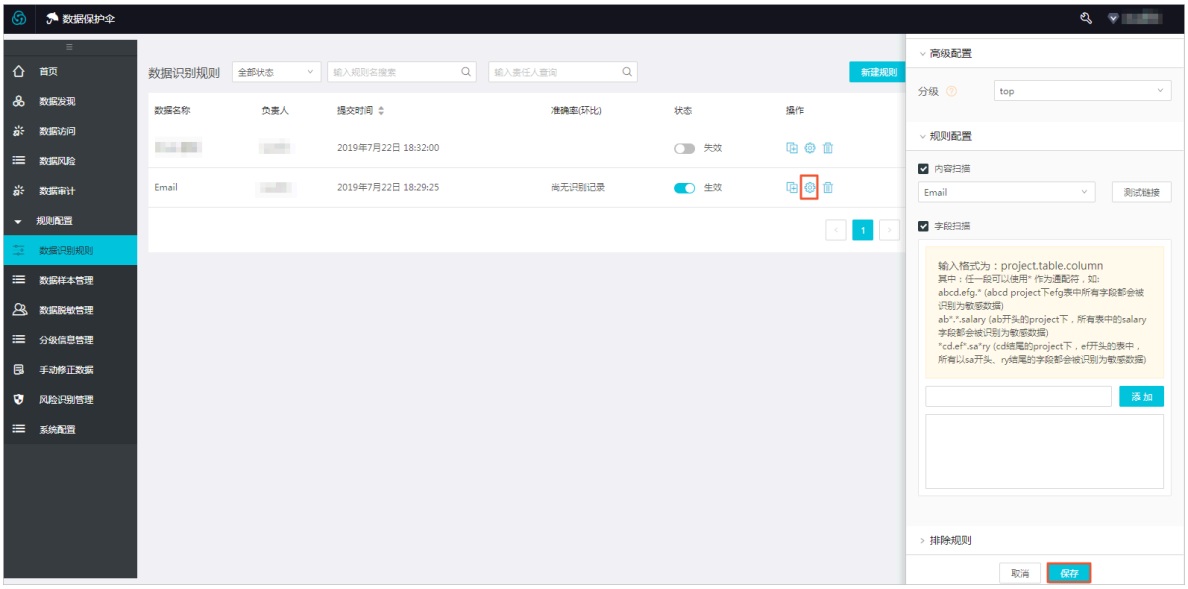
配置	说明
数据类型	即规则所属分类，支持按模板添加或自定义添加。 · 如果选择按模板添加，可以选择个人信息、商户信息和公司信息。 · 如果选择自定义添加，您可以自行填写数据类型。
数据名称	· 如果选择按模板添加，系统内置姓名、邮箱、座机号、手机号、IP、mac地址、车牌号、地址、邮政编码、身份证号、银行卡号和公司名12种敏感数据识别定义模板。 · 如果选择自定义添加，您可以自行填写数据名称。
责任人	规则设置人员信息。
备注	对当前规则进行简单描述。

5. 单击对话框中的配置规则，单击下一步。

配置	说明
分级	对配置的数据进行等级划分，如果现有等级不满足需求，请在分级信息管理处进行设置。
内容扫描	提供的数据识别方式之一，系统内部12种数据识别模板皆为内容扫描。 · 如果选择模板则无法更改识别规则，但提供规则准确性验证的通道，同时对识别不准的情况可进行手动修正。 · 如果选择正则式匹配，则自定义识别规则。
字段扫描	提供字段名精确匹配和模糊匹配方式，支持多个字段匹配，各字段间为或关系。

6. 确认配置无误后，单击保存并生效。

如果需要修改已有规则，需要将该规则置为失效状态，然后单击操作下的规则配置，在右侧对话框进行高级信息的配置和修改。



规则配置修改完成后，单击保存。确认规则无误后，更改状态为生效。

说明:

定义敏感数据时，需遵循如下规则：

- 规则名必须唯一。

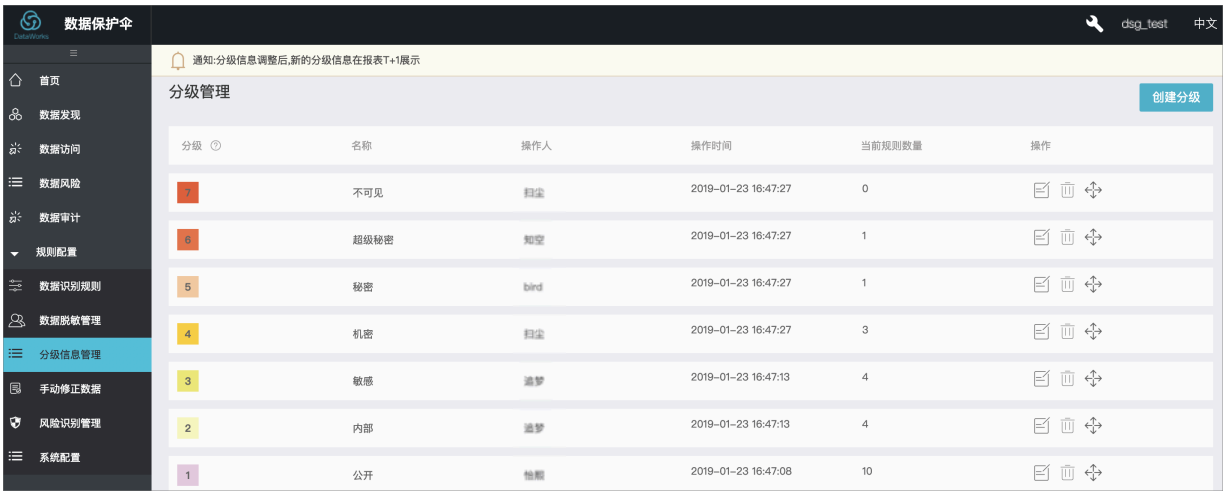
- 不同规则的内容扫描或者字段扫描必须唯一。
- 规则识别数据，T+1在报表展现。

已定义敏感数据

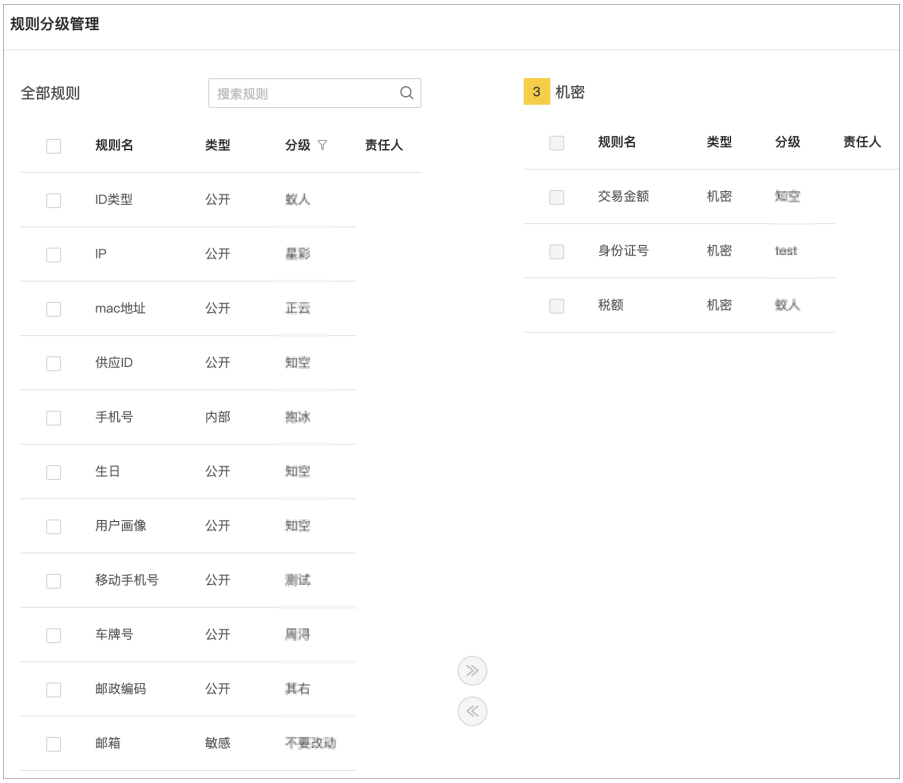
如果您已经定义敏感数据，请直接跳转至数据发现、数据访问和数据风险模块进行查看。

15.7 分级信息管理

当规则配置中的分级选择无法满足您的需求时，可在分级页面管理中进行设置，该页面提供新建分级、删除分级、分级优先级调整和规则分级调整的功能。

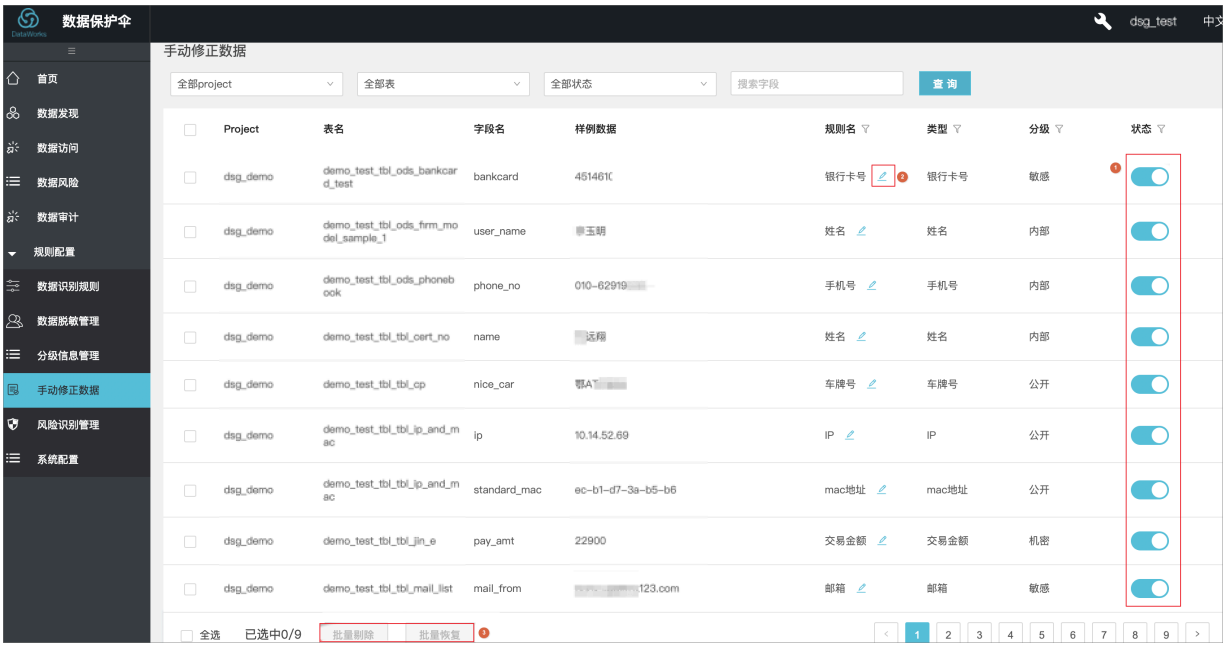


配置	说明
创建分级	单击创建分级增加新的分级，填写名称和操作人。

配置	说明
	调整规则分级，单击后可进行规则的选择与调整。 
	调整分级优先级，单击后向下（降低优先级）或者向上拖动（提高优先级）。
	删除分级，单击后可删除不需要的分级。

15.8 手动修正数据

手动修正页面提供对规则识别的敏感数据不准确的情况进行手动修正的功能，包括剔除识别错误数据、更改识别数据类型和批量处理。



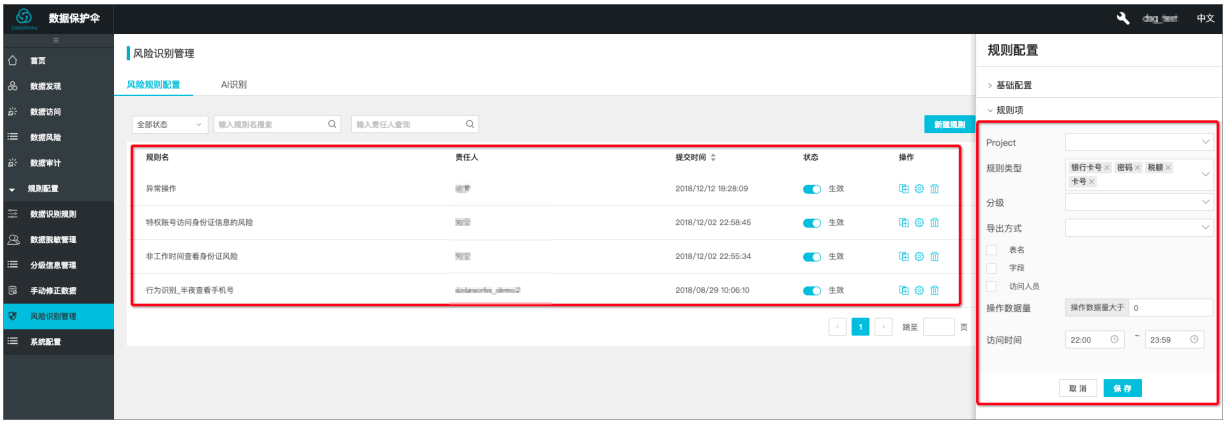
操作	说明
剔除识别错误数据	滑动状态一栏下的按钮，即可更改为已剔除状态，已剔除的数据可恢复。
	更改识别数据类型。如果识别为邮箱，实际为车牌号，则单击邮箱右边的 进行更改，只能选择已配置的规则名称。
批量处理	包括批量剔除和批量恢复，选择需操作的数据，单击数据左方的复选框，再单击相应的操作。

说明:

手动修正数据时需要遵循剔出以及更改数据名称类型均T+1生效于识别数据分布、数据访问行为、数据导出页面的规则。

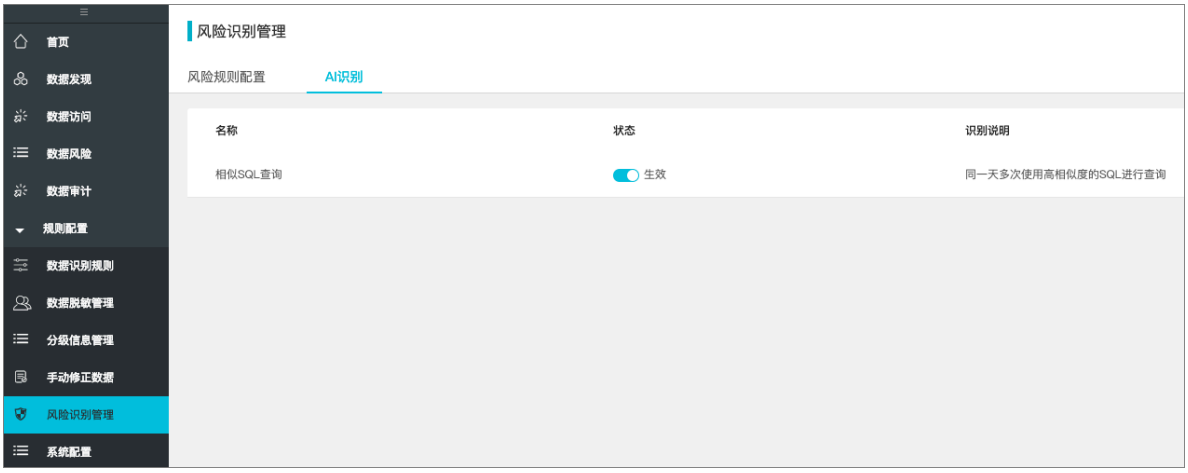
15.9 风险识别管理

风险识别管理页面提供风险数据规则配置，您可以识别日常访问中的风险以及启动AI识别自动识别数据风险，识别后的风险数据统一在数据风险页面进行展示和审计操作，同时也会在数据访问页面处的数据打上识别标志。



页面说明如下：

- 风险识别管理：分为风险规则配置和AI识别。AI识别的页面包括个人信息查询、相似SQL查询以及这两块内容的识别介绍，您只需在状态列启动即可，同时启动后也可关闭（不删除之前识别的数据）。



- 风险规则配置_新建规则：在弹出框中录入规则名称、责任人和规则备注信息后，规则基本信息创建完成。

- 风险规则配置_操作：提供复制规则、编辑风险规则项和删除规则功能。
- 风险规则配置_规则项配置：提供Project（支持多选）、类型（支持多选）、规则（支持多选）、分级（支持多选）、导出方式（支持多选）、表（支持模糊/精确匹配）、字段（支持模糊/精确匹配）、访问人员（支持模糊/精确匹配）、操作数据量和访问时间条件配置。
- 风险规则配置_状态：您在配置完规则后，需在状态列启动规则后生效。

说明:
风险识别管理数据需要遵循规则配置以及AI识别启动后，数据均T+1生效于数据风险页面的规则。

15.10 设置并查询自定义脱敏

DataWorks目前支持动态脱敏，本文将为您介绍如何设置数据保护伞自定义脱敏，并在DataWorks中进行脱敏查询。

数据保护伞自定义脱敏设置

说明:
您需要首先开通数据保护伞服务，方可使用数据保护伞自定义脱敏。

1. 登录数据保护伞平台。
2. 进入规则配置 > 数据脱敏管理页面。

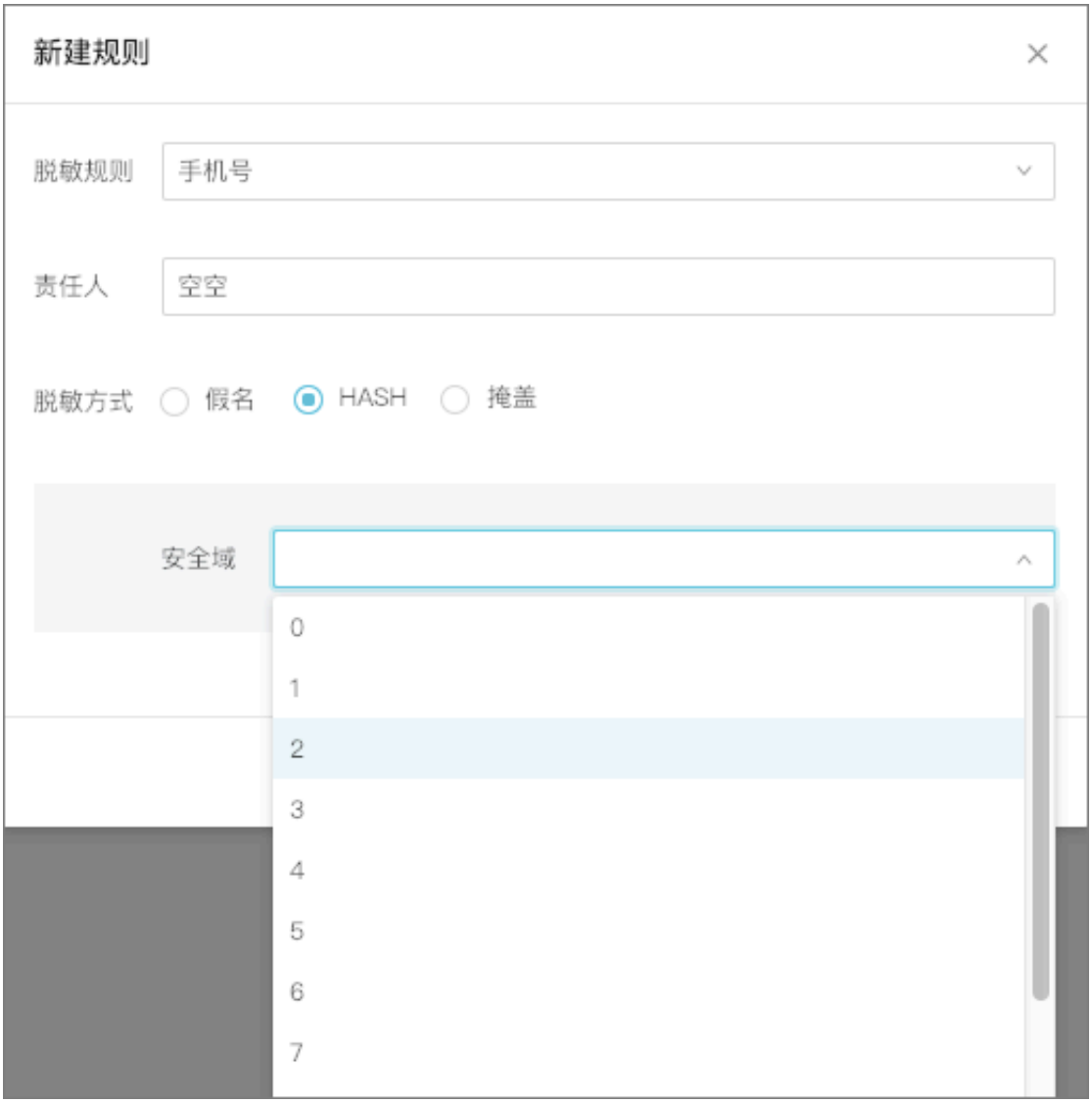
3. 在脱敏场景中选择默认场景（_default_scene_code），然后单击新建规则，添加需要进行自定义脱敏的规则。

4. 在新建规则对话框中选择需要设置的脱敏规则、责任人、脱敏方式和安全域。

目前数据保护伞提供了HASH、掩盖和假名三种脱敏方式。

· HASH

HASH脱敏需要选择一个安全域，相同的值在不同的安全域HASH脱敏后的值不一样。



- **掩盖**

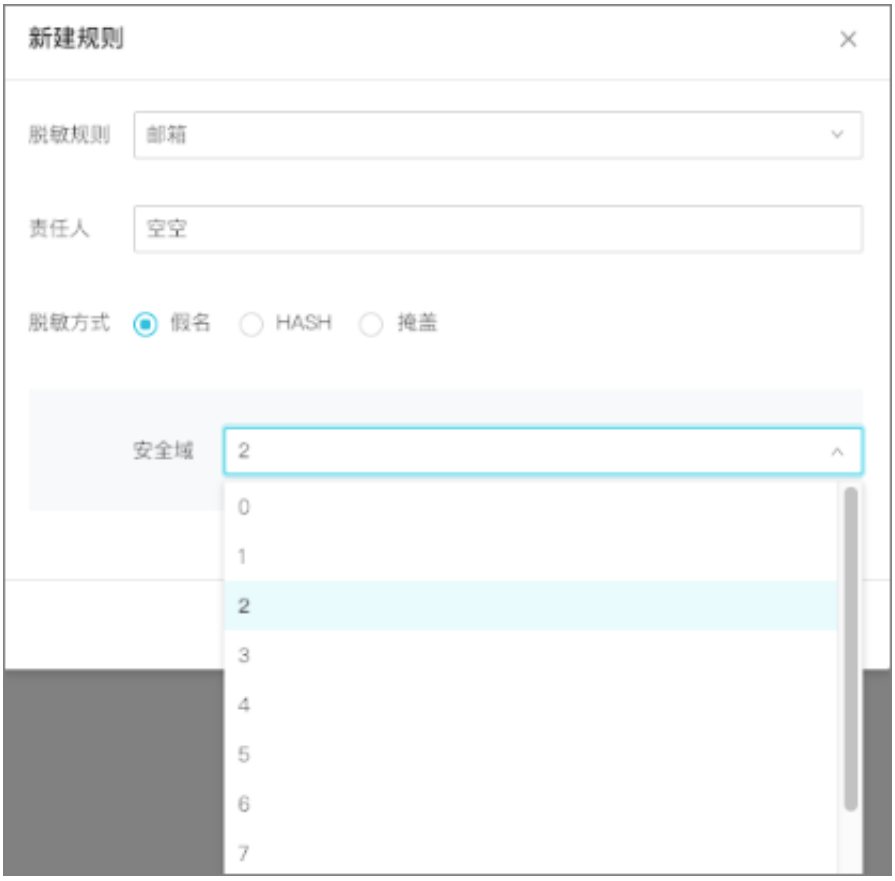
  掩盖脱敏是使用*对部分信息进行掩盖，达到脱敏的效果，是一种比较常用的脱敏方式。

配置	说明
推荐方式	为身份证、银行卡等常用的数据类型提供掩盖脱敏策略。

配置	说明
自定义	自定义设置提供了更加灵活的设置方式，可以在前中后三段上设置是否脱敏，以及需要脱敏（或者不脱敏）的字符长度。 

- 假名

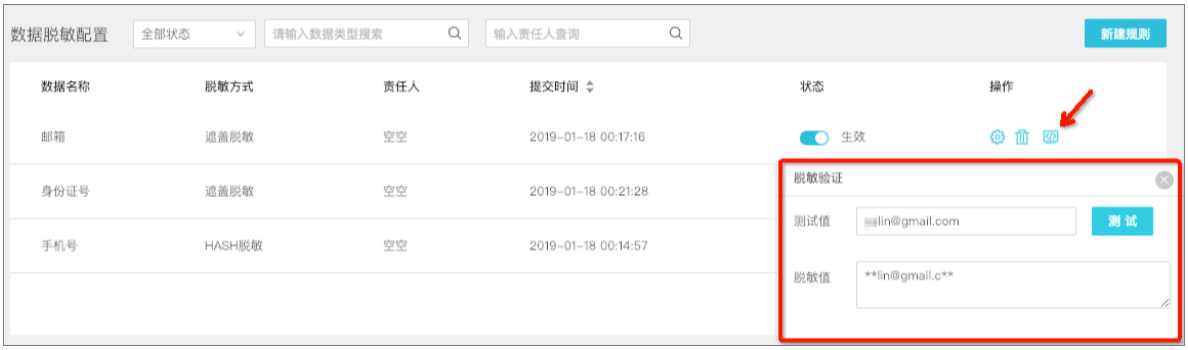
  假名脱敏会将一个值替换成一个具有相同特征的脱敏信息，使用假名脱敏时，也需要选择安全域，相同的值在不同的安全域脱敏出来的假名信息不同。



5. 设置成功后，单击确认，跳转至数据脱敏管理页面。

6. 在数据脱敏配置下，对脱敏策略进行生效或失效操作。

设置成功后，您可单击相应脱敏规则后的 图标，对脱敏效果进行预览。

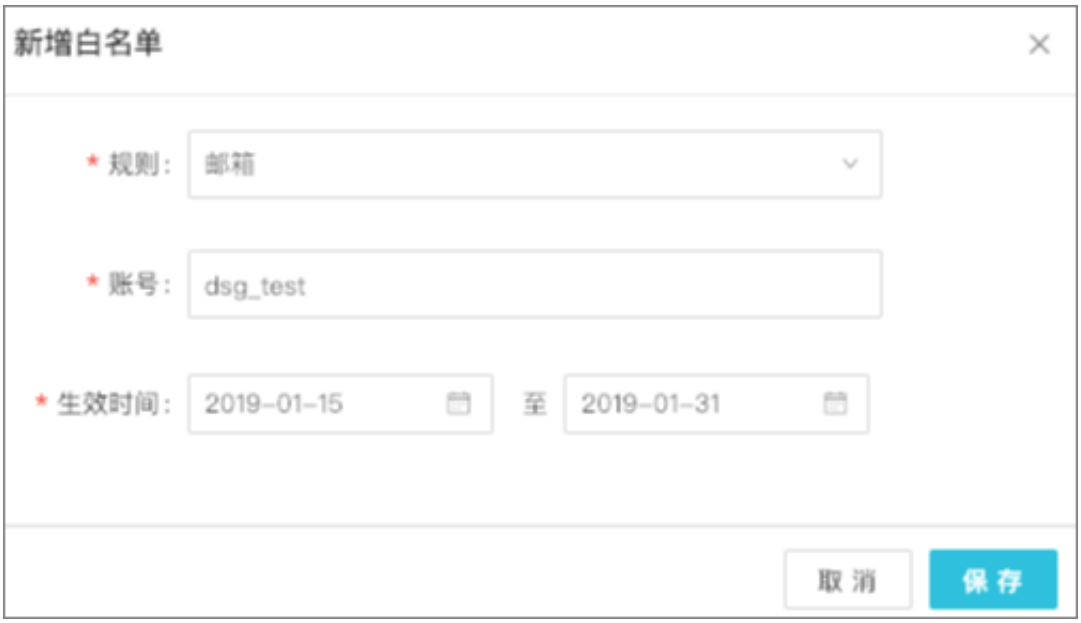


7. 进入白名单配置管理页面，单击新增白名单。

8. 在新增白名单对话框中设置规则、账号和生效时间，并单击保存。

   这里的账号支持主账号和RAM子账号。

   · 主账号



   · 子账号

   说明:

   设置白名单生效时间后，如果不在白名单脱敏时间的区间内，该用户在查询该敏感信息时将会继续脱敏。

DataWorks中进行脱敏查询

成功新建脱敏规则并进行配置后，您可在DataWorks中进行脱敏查询。

说明:

您需要首先开启DataWorks项目空间的查询脱敏功能，详情请参见#unique_370。

16 安全中心

16.1 安全中心概述

安全中心为您提供便捷的权限管控能力，提供可视化申请审批流程，并可进行权限的审计和管理，提高数据安全的同时还可方便您进行数据权限管控。

> 说明:
> 安全中心已开放内测邀请。

您可以通过单击左上方的DataWorks图标，切换至安全中心页面。

安全中心模块包括我的权限、权限审计和审批中心三大模块。目前安全中心具备的功能如下所示。

- 权限自助申请：您可选择自己需要权限的数据表，在线上快速发起申请，改变原有线下联系管理员的模式，提高工作效率。
- 权限审计/交还：管理员可以快速方便地查看数据库表权限对应人员，进行审计管理，用户也可主动交还不再需要的权限。
- 权限审批管理：将以前管理员直接授权的模式改为审批授权模式，提供可视化、流程化的管理授权机制，并可对审批流程进行事后追溯。

您可在安全中心模块进行组织内全局数据表权限的查看、表权限管理、数据表权限申请/审批等操作。

安全中心的各项操作支持同一租户下的所有工作空间，包括标准模式（开发环境&生产环境）及简单模式。

16.2 快速入门

本文将通过一个简单示例，为您介绍安全中心各功能的使用。

前提条件

使用安全中心前请注意以下事项。

- 关于字段级别授权与LabelSecurity

  未开启LabelSecurity机制的工作空间将无法进行字段级别的授权，只能进行整表授权，同时无法在申请时设置有效期。LabelSecurity的详情请参见#unique_713。

- 关于权限有效期

  为保证字段级别的授权在有效期正常运行，除开启LabelSecurity外，还请务必保证各字段的LabelSecurity大于账号的LabelSecurity。

  如果部分字段的LabelSecurity为空或不大于账号的LabelSecurity，账号申请该表权限时虽然可以选择有效期，但申请审批通过后将自动获取这部分字段的权限，同时有效期为永久，并且无法单独交还/回收这部分字段权限。

- 关于申请及展示权限

  目前安全中心仅展示通过ACL授权方式所获取的权限，不展示通过用户角色等其他方式获取的权限。例如作为项目开发角色的账号可以访问项目中所有的表，但在安全中心中，并不显示对项目中的表具有权限。当发现安全中心未展示具有权限但实际可以访问的情况，请与系统管理员确认是否通过用户角色等其他方式获取到了相应权限。

用户角色对应功能介绍

- 子账号-普通用户
  - 我的权限：支持查看权限、申请权限、交还操作权限和交还字段权限等操作。
  - 审批中心 > 我的申请：支持查看申请单进度和查看历史申请等操作。
- 子账号-表Owner
  - 我的权限：支持查看权限、申请自己不是表Owner的表权限、交还自己不是表Owner的表操作权限、交还自己不是表Owner的表字段权限等操作。
  - 审批中心 > 我的申请：支持查看申请单进度和查看历史申请等操作。
  - 审批中心 > 待我审批：支持审批等待自己处理的申请单。
  - 审批中心 > 我已审批：支持查看自己处理过的申请单。

- 子账号-项目管理员
  - 权限审计：支持查看项目成员和回收成员权限等操作。
  - 审批中心 > 我的申请：支持查看申请单进度和查看历史申请等操作。
  - 审批中心 > 待我审批：支持审批等待自己处理的申请单。
  - 审批中心 > 我已审批：支持查看自己处理过的申请单。
- 主账号
  - 权限审计：支持查看项目成员和回收成员权限等操作。
  - 审批中心 > 我的申请：支持查看申请单进度和查看历史申请等操作。
  - 审批中心 > 待我审批：支持审批等待自己处理的申请单。
  - 审批中心 > 我已审批：支持查看自己处理过的申请单。

示例介绍

本示例的角色分类有：普通用户、表Owner和项目管理员。

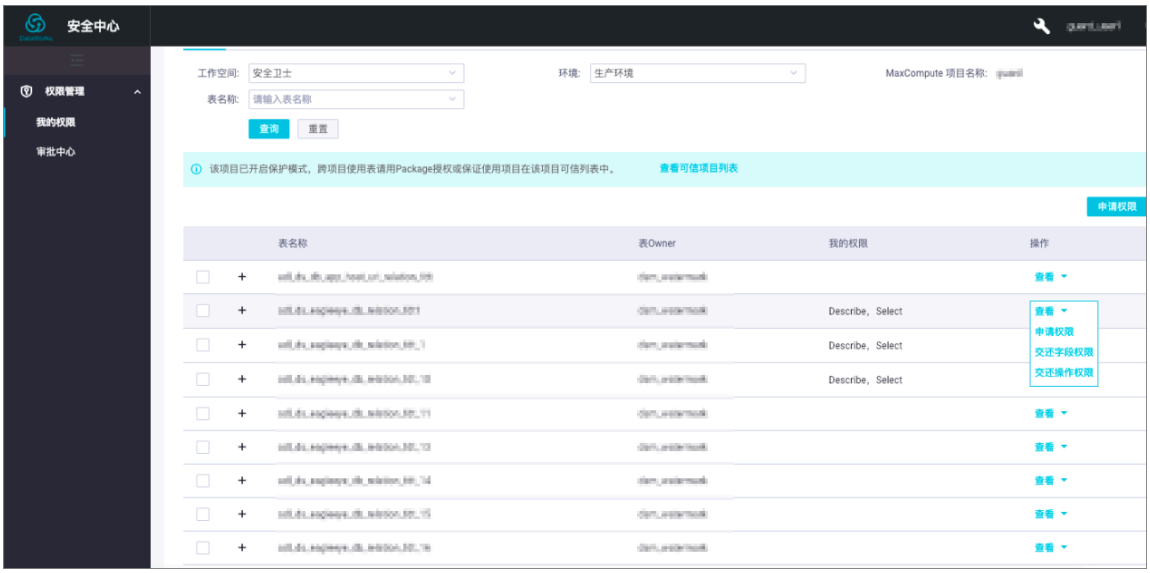
本示例需要实现以下场景。

- 子账号甲作为普通用户查看自己当前具有的权限，然后申请A、B这两张目前无权限的数据表的权限。
- 子账号乙作为A表的表Owner进行审批，通过A表申请。
- 主账号作为项目管理员进行审批，通过B表申请。
- 子账号甲交还A表部分字段无需使用，交还部分字段权限。
- 子账号甲不再需要使用A表，交还A表权限。
- 主账号回收子账号B表的权限。

普通用户

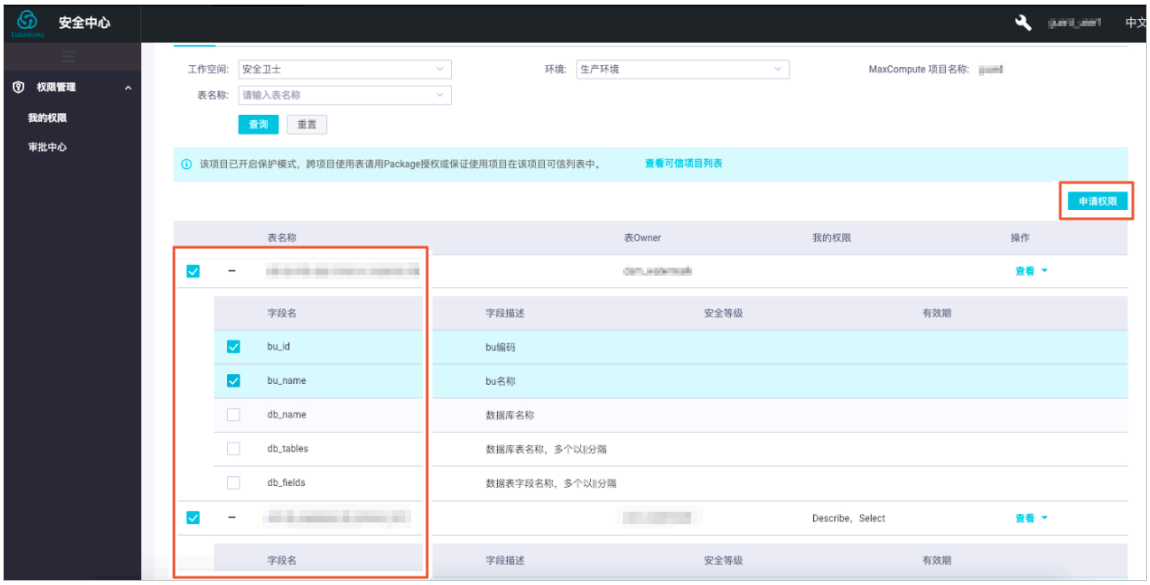
- 子账号甲查看自己在项目中的权限

1. 子账号登录控制台，进入安全中心 > 我的权限 > 表权限页面，对本组织范围内（多工作空间）的数据表进行查看和搜索。
2. 在表权限页面，通过选择工作空间 + 环境（标准模式下）展示该工作空间对应环境下所有表， 查看自己对哪些表具有权限。

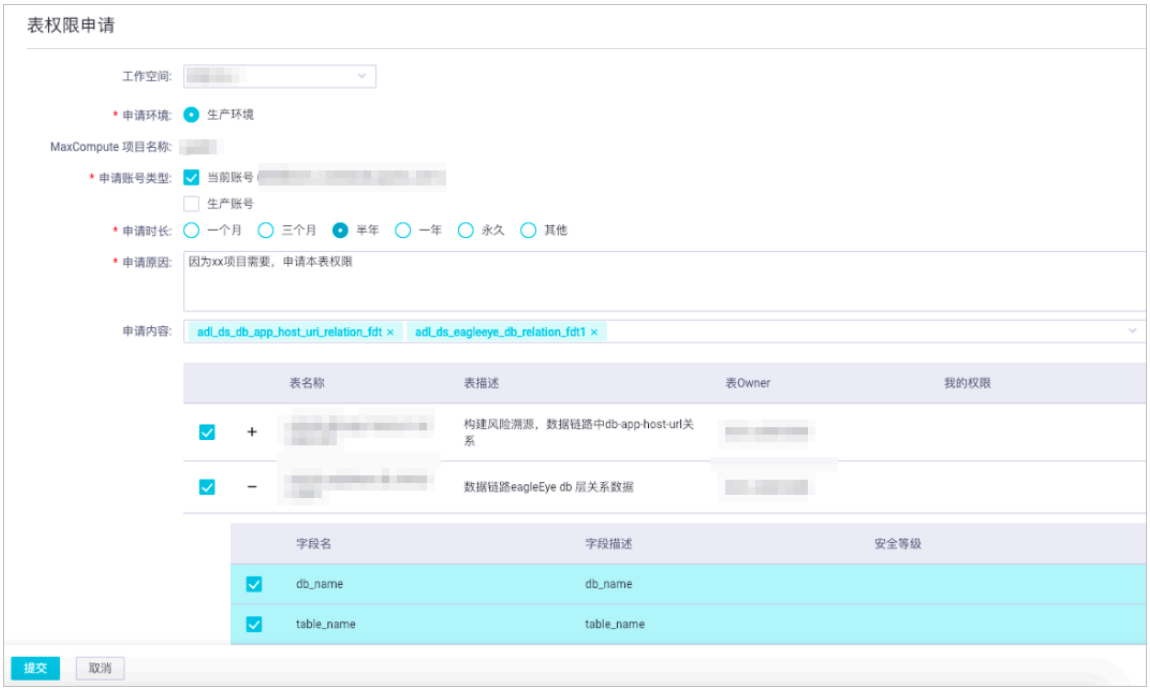


- 申请A、B两张表的权限

1. 子账号登录控制台，进入安全中心 > 我的权限 > 表权限页面。
2. 勾选A、B两张表需要申请权限的字段，单击申请权限。



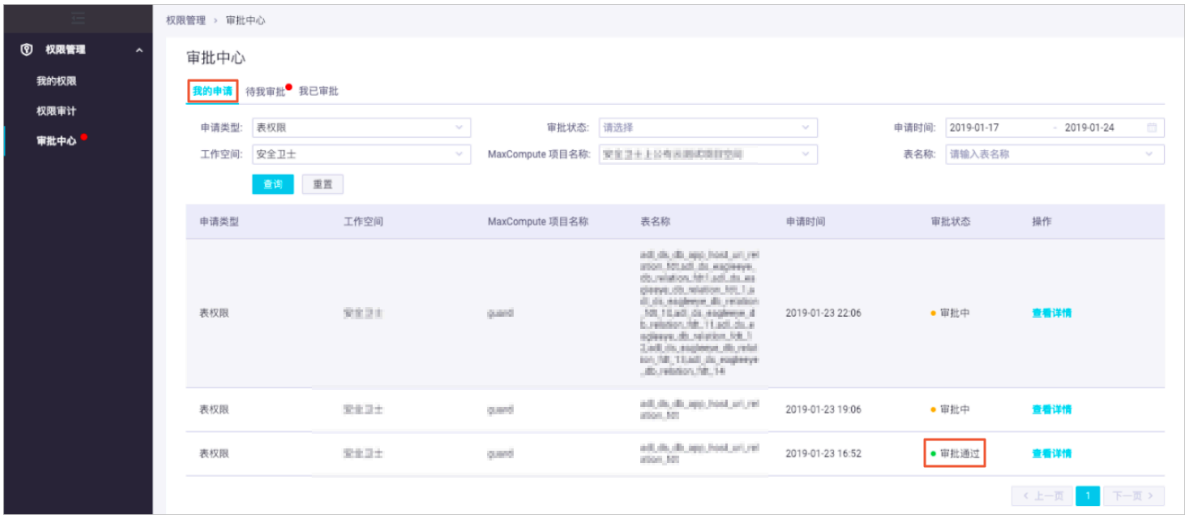
3. 填写表权限申请对话框中的信息。



4. 单击提交。

- 等待审批人审批通过

  进入安全中心 > 审批中心 > 我的申请页面，查看之前申请的进度。当审批状态为审批通过时，代表已具有相应的表权限，可以进行相关操作。

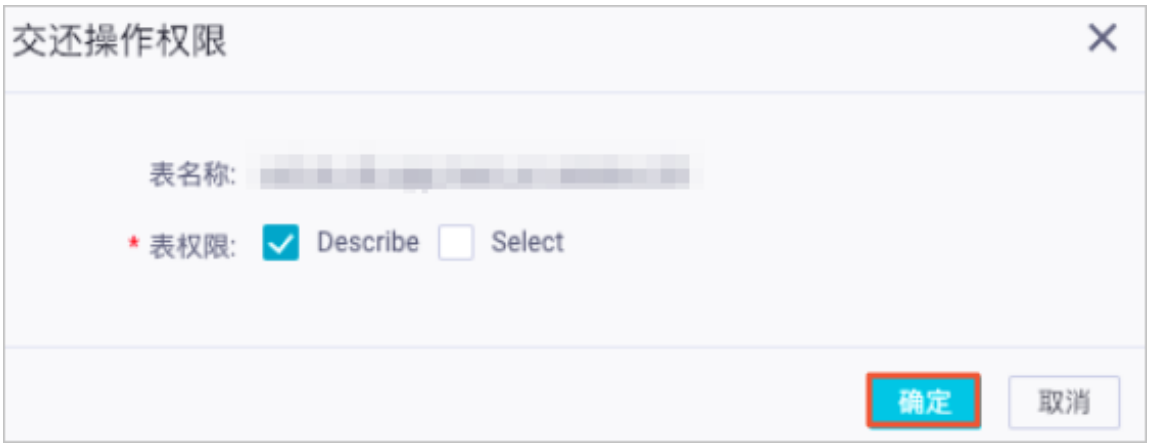


- **交还A表部分字段的权限**

  1. **子账号登录安全中心 > 我的权限 > 表权限页面。**
  2. **单击A表操作栏中的点击查看，选择交还字段权限。**
  3. **勾选交还字段权限对话框中需要交还的字段，单击确定。**

- 交还A表操作权限

  1. 子账号登录安全中心 > 我的权限 > 表权限页面。
  2. 单击A表操作栏中的点击查看，选择交还操作权限。
  3. 勾选交还操作权限对话框中需要交还的权限，单击确定。

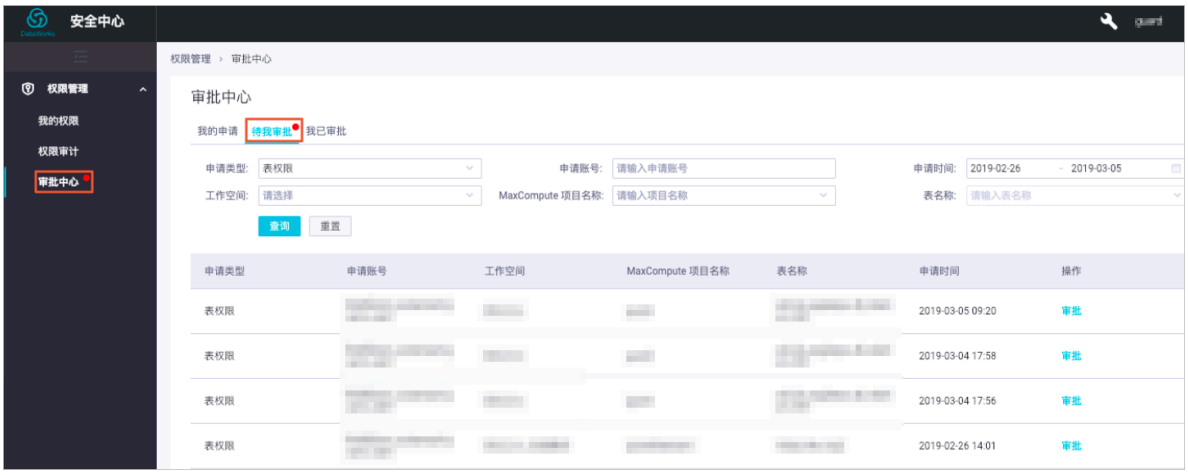


表Owner

子账号乙作为A表的表Owner进行审批，通过A表的申请。

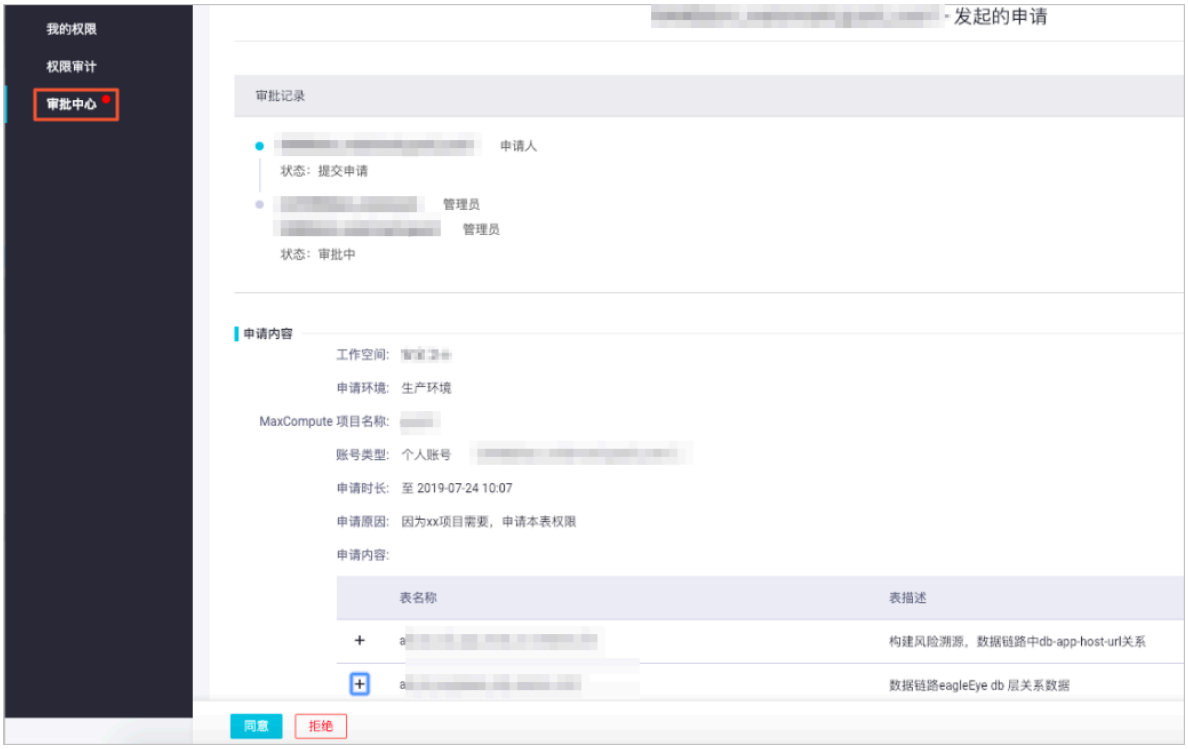
子账号乙作为表Owner，除能使用子账号甲作为普通用户的功能外，还可进行自己作为表Onwer的相关表的审批。

1. 进入安全中心 > 审批中心 > 待我审批页面。



2. 单击子账号刚刚提交的申请单后的审批，即可进入申请单详情页面查看审批记录和申请内容。

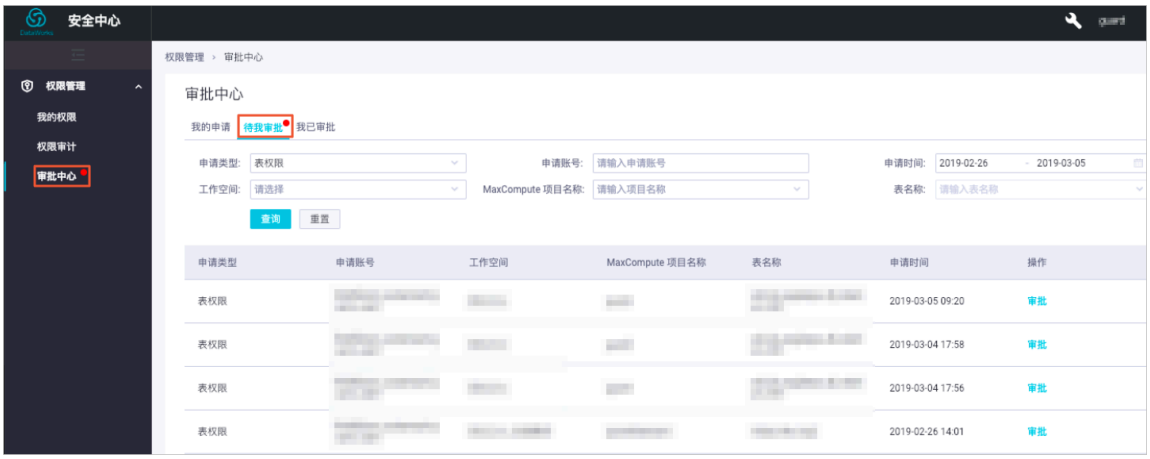
3. 填写审批意见，并单击同意审批通过该申请。



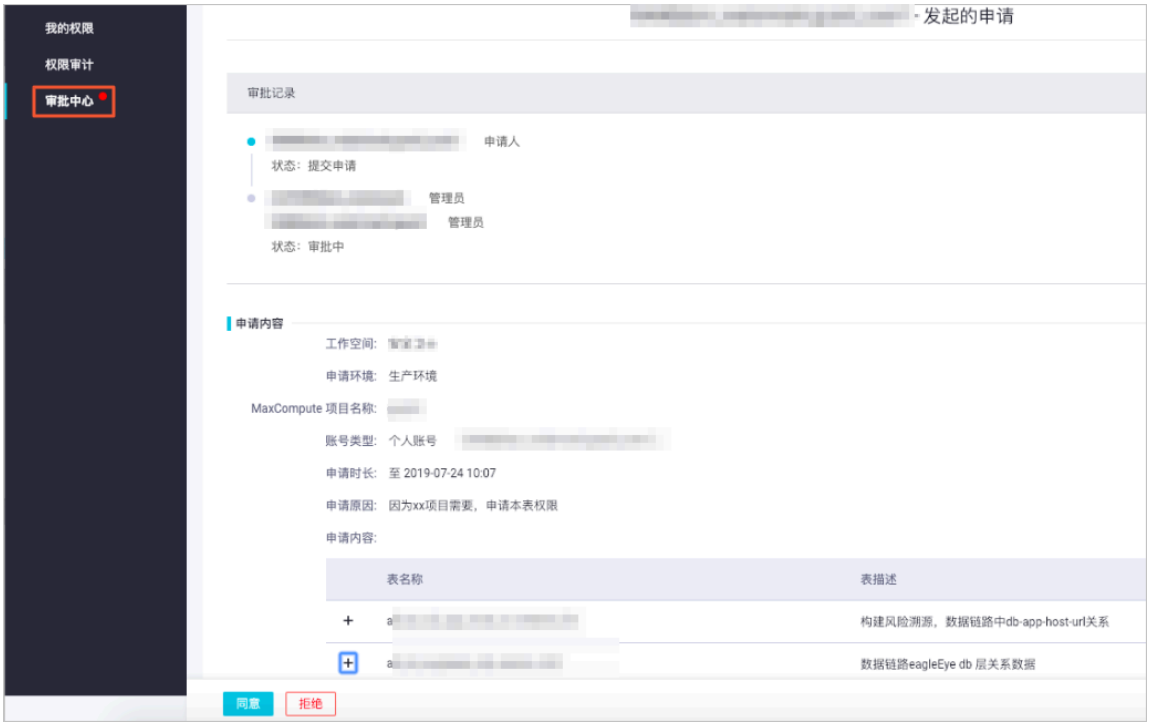
项目管理员

- 主账号作为项目管理员进行审批，通过B表申请。

1. 主账号进入安全中心 > 权限审计 > 表权限页面。

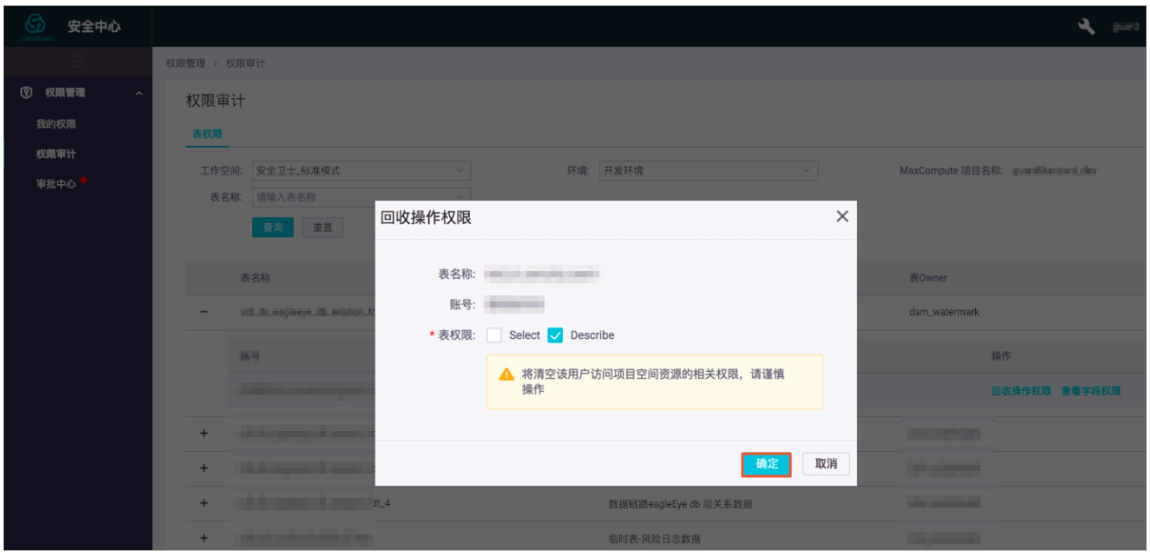


2. 单击子账号刚刚提交的申请单后的审批，即可进入申请单详情页面查看审批记录和申请内容。
3. 填写审批意见，并单击同意审批通过该申请。



- 回收子账号B表的权限

  1. 主账号进入安全中心权限审计页面。
  2. 找到子账号的B表，单击查看成员列表。
  3. 单击子账号操作栏中的回收操作权限。
  4. 勾选回收操作权限对话框中需要回收的权限，单击确定，即可回收该账号拥有的当前数据表的权限。



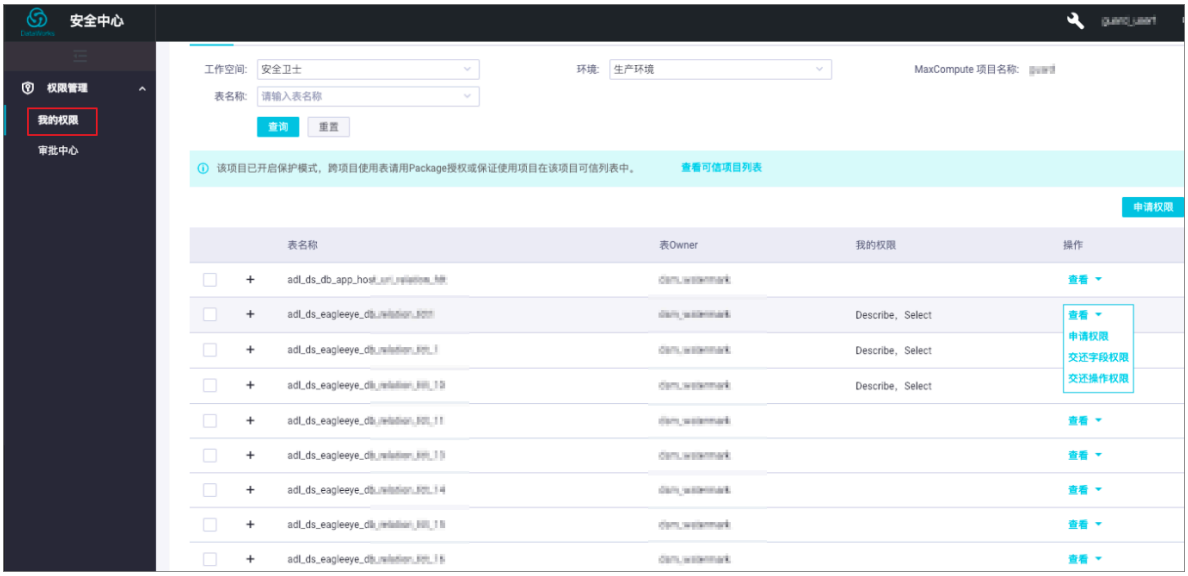
16.3 我的权限

您可在我的权限页面，查看工作空间内自己拥有的表/字段权限，并对表/字段的权限进行申请或交还。

查看表/字段权限

1. 进入安全中心 > 我的权限 > 表权限页面。

2. 在表权限页面，您可通过选择工作空间 + 环境（标准模式下）查看该工作空间对应环境下所有的表， 也可通过搜索框中输入表名进行模糊匹配的方式快速查找需要的表。



您可在此页面查看当前工作空间的表名称、表Owner和我的权限，并进行申请权限、交还字段权限和交还操作权限的操作。

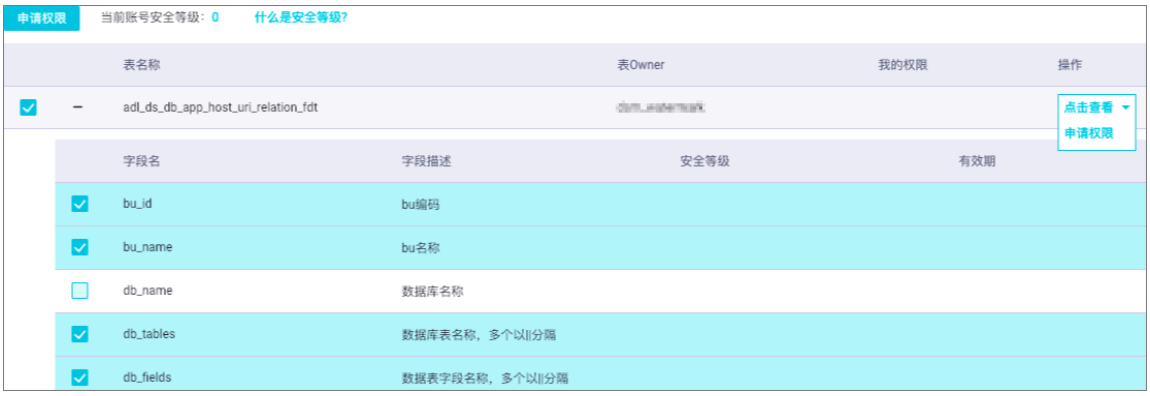
申请表/字段权限

1. 选择需要申请的表/字段权限。

   · 申请单张表/字段权限

   您可勾选自己当前无权限但需要使用的表的具体字段后，选择相应操作栏中的点击查看 > 申请权限。

   您也可不勾选具体字段，直接选择相应表后操作栏中的点击查看 > 申请权限，则默认申请该表的所有字段权限。



说明:

未开启LabelSecurity机制的工作空间将无法进行字段级别的授权，只能进行整表授权。

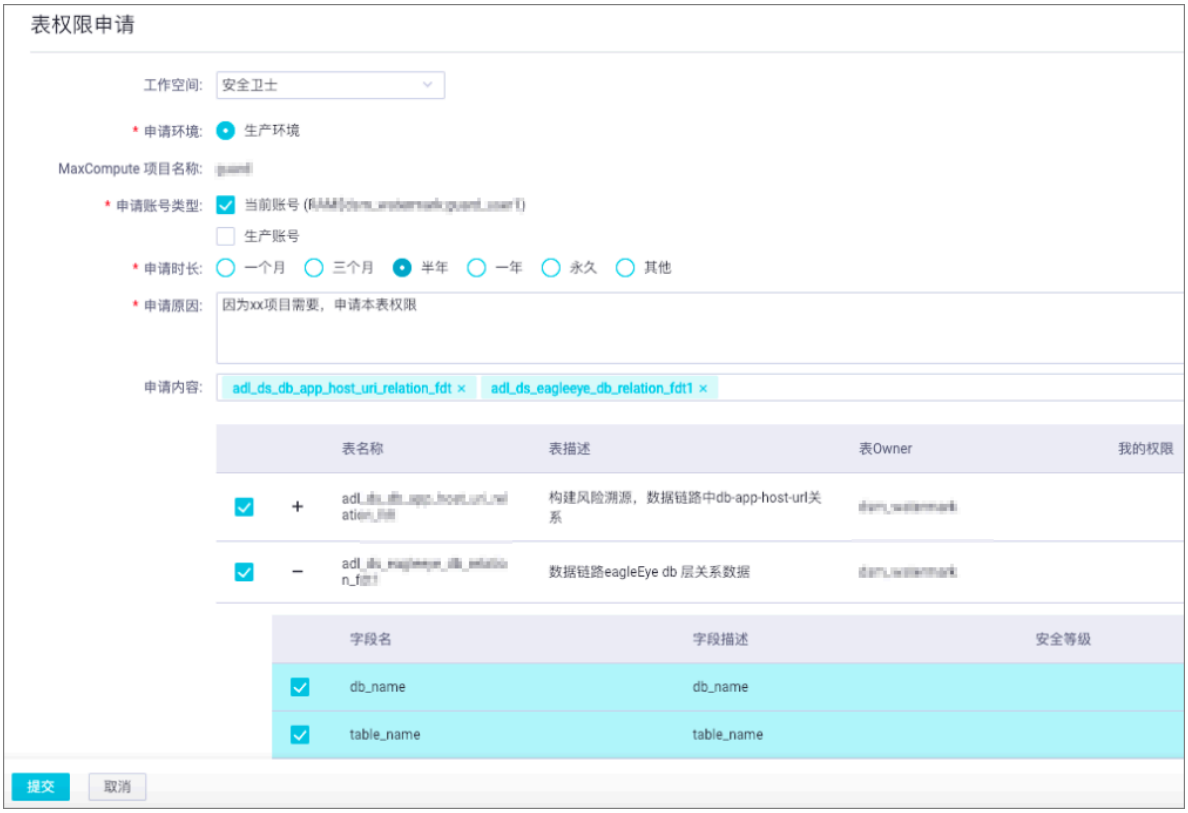
- 申请多张表/字段权限

  勾选需要申请的所有表/字段后，单击申请权限进行批量申请。

说明:

您也可直接单击申请权限，然后再申请详情中填写需要申请的内容。

2. 填写表权限申请对话框中的配置。



配置	申请
工作空间	工作空间会根据我的权限页面中的信息自动填写，您也可以进行修改。
申请环境	申请的工作空间的环境。
MaxCompute项目名称	申请的MaxCompute项目名称。
申请账号类型	您可以为当前账号申请权限，也可以替自己加入的其他工作空间的生产账号申请权限。
申请时长	您可选择一个月、三个月、半年、一年、永久或其他。
申请原因	简单说明申请权限的原因。
申请内容	填写表名称，可在原有基础上进行增加和删除。

3. 配置项填写完成后，单击提交。如果不想申请，可单击取消。

交还权限

交还权限包括交还字段权限和交还操作权限。

- 交还字段权限。

> 说明:

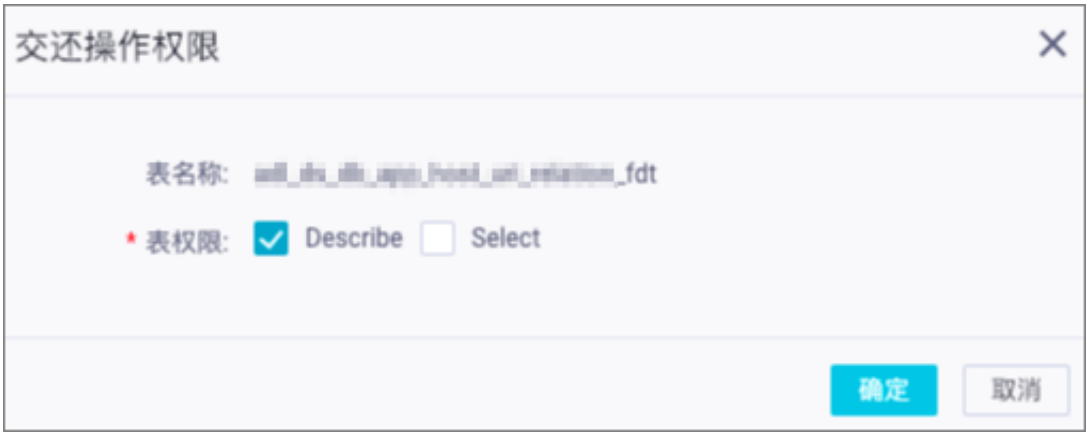
- 交还字段权限仅在已开启LabelSecurity的工作空间中可见。
- 如果希望交还整张表的权限，请使用交还操作权限功能释放所有权限。

1. 选择想要交还权限的表后的操作 > 点击查看 > 选择字段权限。
2. 在交还字段权限对话框中，勾选需要交还权限的字段。

3. 单击确定。

- 交还操作权限。

  1. 选择想要交还权限的表后的操作 > 点击查看 > 选择操作权限。
  2. 在交还操作权限对话框中，勾选需要交还的表权限。



  3. 单击确定。

16.4 权限审计

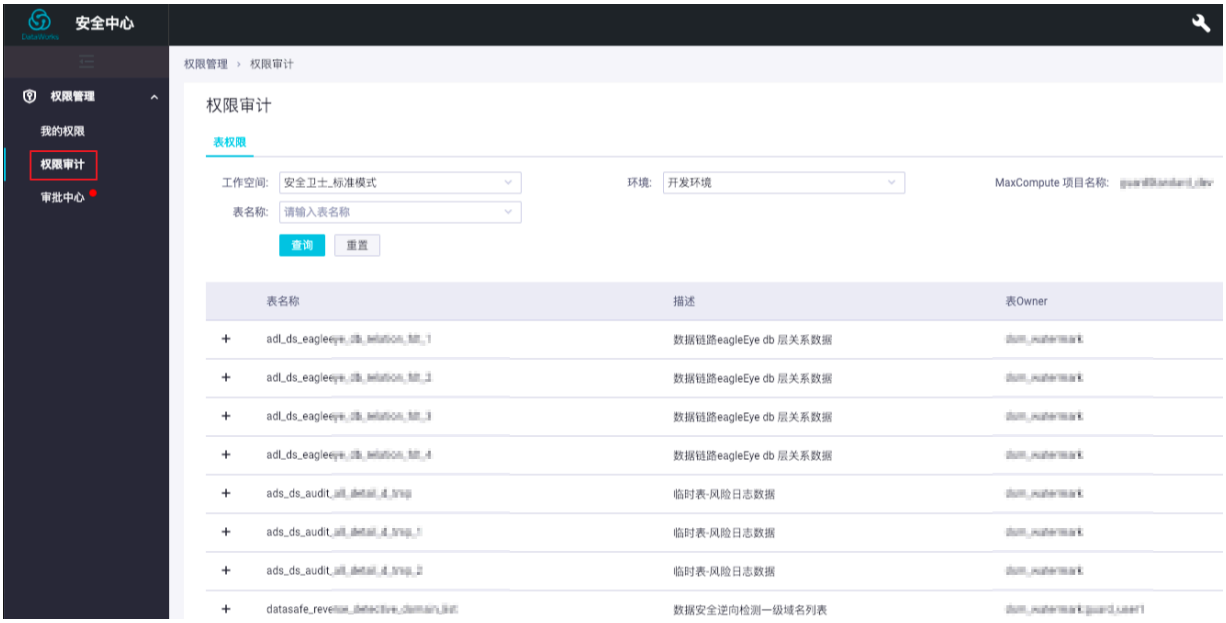
项目管理员可在权限审计页面查看各个工作空间内，分别有哪些账号拥有表和字段的权限，并可回收不必要的表/字段权限。

您可进入安全中心 > 权限审计 > 表权限页面，对本组织范围内（多工作空间）的数据表进行查看和搜索。

在表权限页面中，您可通过选择工作空间+环境（标准模式下）展示该工作空间对应环境下的所有表， 也可以通过搜索框中输入表名进行模糊匹配的方式快速查找需要的表。

查看拥有表权限的账号

您可通过权限审计 > 表权限查看拥有该表权限的所有账号。

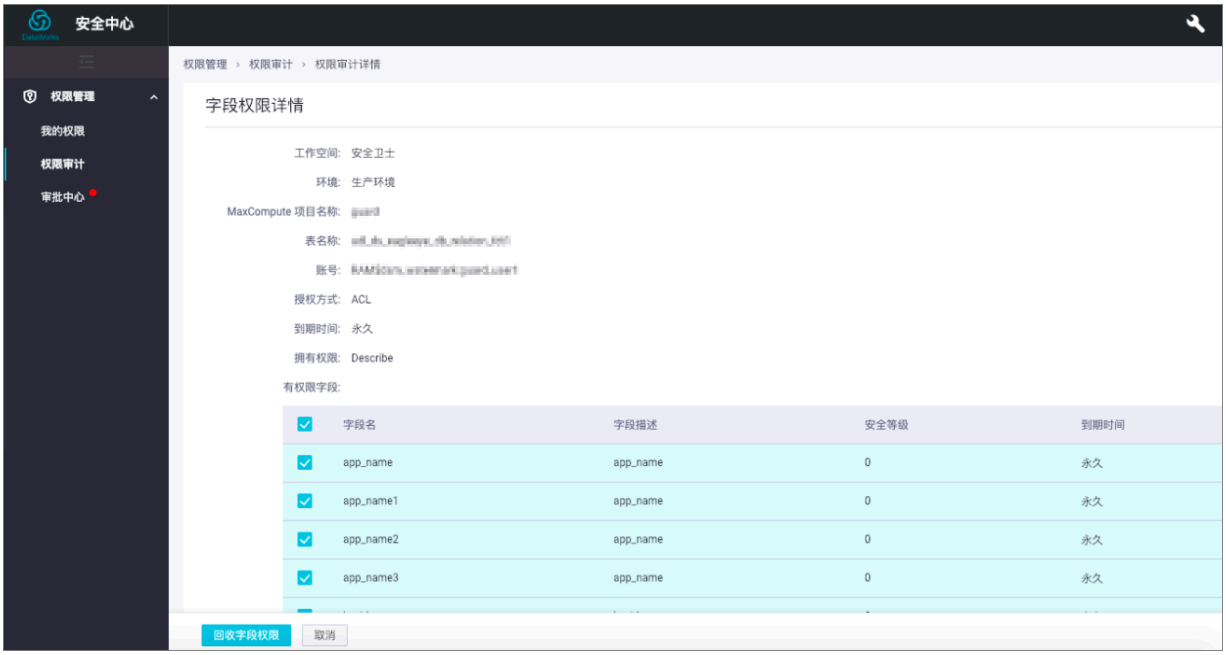


回收表权限

单击相应账号后操作栏中的回收操作权限，即可回收该账号拥有的当前数据表的权限。

查看字段权限

单击相应账号后操作栏中的查看字段权限，即可查看该账号拥有的当前数据表的字段权限。



回收字段权限

对于已经开启LabelSecurity的项目，可以在字段权限详情页面，勾选需要回收的字段，单击回收字段权限，即可进行回收。

16.5 审批中心

您可在审批中心页面，查看自己提交的申请及进度，查看待自己审批的申请并进行审批，也可查看自己以前完成的审批任务。

我的申请

1. 进入安全中心 > 审批中心 > 我的申请页面。

   您可在此页面查看自己提交的申请，包括申请类型、工作空间、MaxCompute项目名称、表名称、申请时间、审批结果等信息。

说明:

同一个申请单提交的多张表权限申请，会按照表Owner的不同，可能会自动拆分成多个申请。

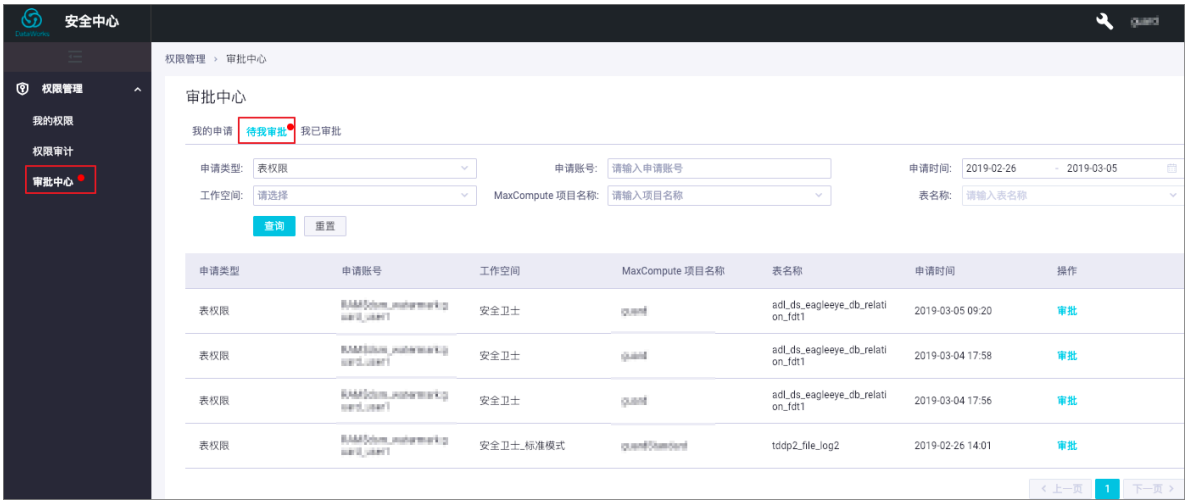
2. 单击操作栏中的查看详情，可以查看申请单的详细信息。

待我审批

1. 进入安全中心 > 审批中心 > 待我审批页面。

   您可在此页面查看当前需要您进行审批的申请单。如果存在需要您审批的申请，则审批中心和待我审批页面会显示红点进行提醒。

   您可查看待审批的申请类型、申请账号、工作空间、MaxCompute项目名称、表名称、申请时间等信息。



2. 单击操作栏中的审批，可以查看申请单的详细信息并进行审批，审批详情页面会为您展示审批记录和申请内容。

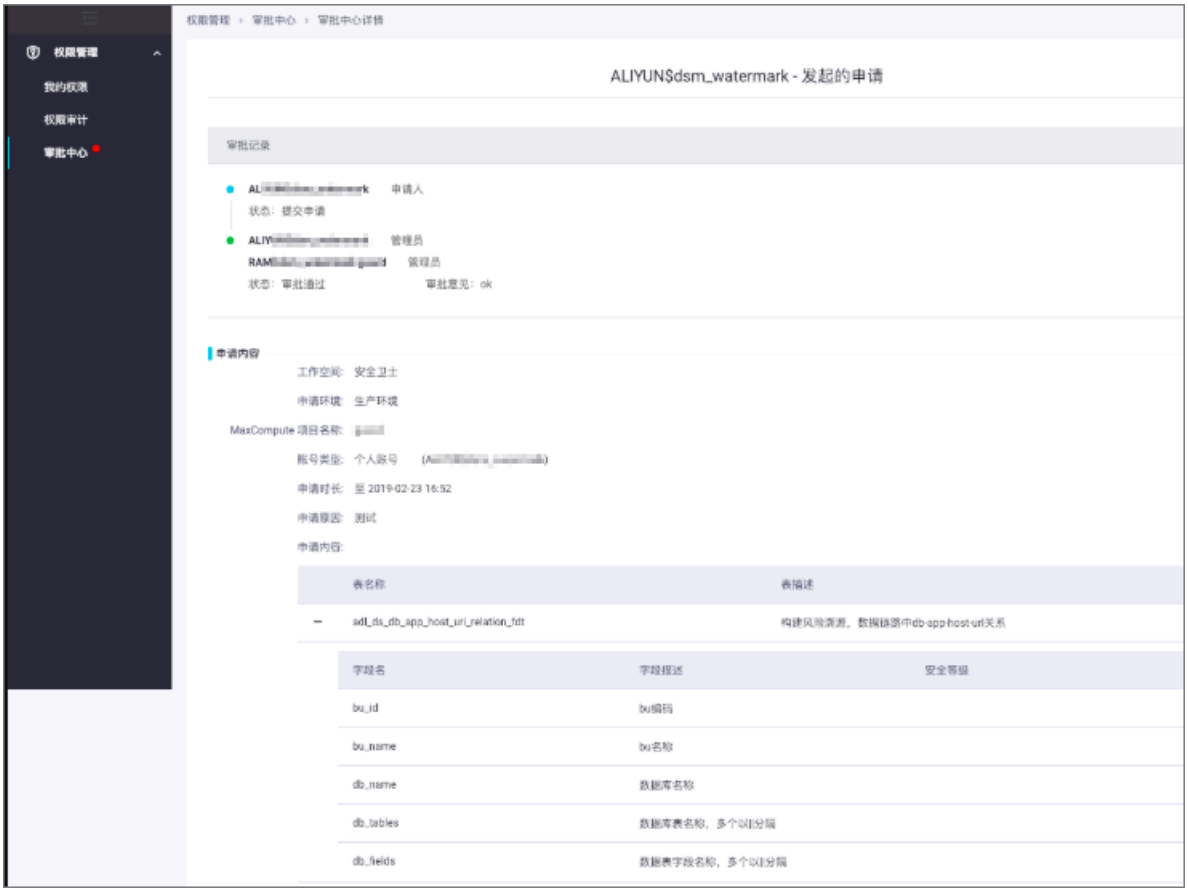
3. 填写审批意见后，您可根据实际情况同意或拒绝该申请。

我已审批

1. 进入安全中心 > 审批中心 > 我已审批页面。

   您可在此页面查看以前审批过的申请单，包括已审批的申请类型、申请账号、工作空间、MaxCompute项目名称、表名称、申请时间等信息。

2. 单击操作栏中的查看详情，即可查看申请单的审批记录和申请内容等详细信息。



16.6 常见问题

本文将为您介绍DataWorks安全中心模块的常见问题和解决方法。

- Q：通过安全中心可以申请什么权限?

  A：您可通过安全中心页面申请DataWorks工作空间内的表权限，包括开发环境和生产环境。

- Q：数据管理和安全中心是什么关系?

  A：安全中心是数据管理中与权限和安全相关功能的升级替代产品。此前已经在数据管理模块中申请的权限和通过`odpscmd grant`命令授权的权限，仍可在安全中心 > 我的权限中显示。

  如果您需通过可视化的方式进行新的权限申请、审批操作，请进入安全中心进行操作，数据管理模块后续将不再支持权限的申请和审批。

- Q：为什么在申请时，有时可以选择字段，有时不可以选择?

  A：如果该工作空间开启了LabelSecurity，即可在申请时选择字段，未开启则只能整表申请。

- Q：提交申请后，需要谁进行审批?

  A：提交的申请需要项目管理员或表Owner进行审批，其中任何一个审批通过/拒绝，则审批完成。

- Q：为什么提交了一个申请，在我的申请中却看到两个申请单？

  A：因为您的申请单中包含的数据表的表Owner不同，安全中心会按照表Owner对于申请单自动进行拆分。
- Q：为什么有的字段只申请1个月权限，审批完成后查看变为永久？

  A：说明字段的安全等级为0或者小于等于您账号的安全等级。
- Q：为什么有的表和字段没有申请权限，但能看到有权限？

  A：出现此情况有以下两种可能：

  - 除安全中心外，管理员还可通过控制台命令行给您授权。
  - 如果您是通过安全中心进行了申请，则说明字段的安全等级为0或者小于等于您账号的安全等级。
- Q：为什么并没有审批某个待我审批中的申请单，却没有了？

  A：因为申请单由项目管理员或表Owner进行审批，其他项目管理员或表Owner在您之前完成了审批，因此该申请单已成为完结状态，便从您的待我审批中消失了。
- Q：查询某个工作空间和环境，提示MaxCompute项目异常，无法进行操作，该如何处理？

  A：请将提示框及框内的错误编码发给项目管理员，由他进行问题的排查及解决。
- Q：为什么交还/回收某个字段权限却无效？

  A：只能交还/回收字段的安全等级大于账号安全等级的字段，对于安全等级为0或者小于等于账号安全等级的字段，无法进行字段权限的交还/回收。
- Q：为什么主账号不能申请权限？

  A：主账号默认具有所有权限，无需单独申请权限，因此对主账号无需的操作，如申请权限等进行隐藏，不会影响到主账号的正常使用。
- Q：是否可以在安全中心页面查看以前在数据管理页面的申请/审批记录？

  A：目前安全中心和数据管理中的申请/审批记录没有进行关联，如果需要查看在数据管理进行申请/审批的历史记录，请跳转至数据管理页面进行查看。
- Q：是否可以通过安全中心的申请记录来回收权限？

  A：目前安全中心并非唯一的授权渠道，为了最大程度地支持权限回收，权限审计中列出的是所有用户的ACL权限，不区分授权渠道。您可以基于目前的权限现状进行回收，不必通过申请记录来操作。
- Q：之前在数据管理提交的权限申请，仍未通过审批，是否需要重新申请？

  A：安全中心和数据管理中的申请审批记录未进行关联，您需要重新申请。

- **Q：如何设置字段的LabelSecurity?**

  A：您需前往数据地图设置字段的LabelSecurity。

17 需求管理

17.1 需求管理概述

需求管理以工具化、产品化的方式，帮助阿里云大数据用户、企业以最低的成本实现规范性数据研发流程。

在大数据时代，规范地进行数据研发尤为重要，作用如下：

- 简化、规范日常工作流程。
- 减少无效和冗余工作，提高工作效率。
- 保障数据研发工作有条不紊地运作。

产品优势

- 更专业的需求管理方式：基于通用的数据研发场景增加里程碑功能，呈现每个数据需求所处的阶段，打通从数据需求提出到交付的闭环流程。
- 快速构建规范性研发流程：基于需求管理的功能，您可以全面控制数据需求的评审、设计、开发、测试、发布和验收等环节的生命周期。

基础功能

- 创建需求：阿里云主/子账号登录需求管理模块后，即可进入创建需求页面，填写并提交需求，同时将需求指派给主账号下的其他成员作为责任人。
- 管理需求：需求责任人、被抄送人可以根据实际情况，在需求详情页面进行添加评论、修改优先级、修改需求状态和上传附件等操作。
- 搜索需求：需求管理为您提供普通搜索、条件搜索和高级搜索三种方式查找所需要的已提交过的需求。
- 需求视图：使用者在需求列表中可以将搜索（过滤）后的结果保存为视图，方便后续的筛选与过滤。

特色功能

- 里程碑：根据规范性数据研发流程为用户定制6大里程碑，助力数据研发工作的进行。
- 关联开发任务：每个需求可以关联对应的开发任务，同时可视化展示被关联任务的发布进度，并与里程碑相关联，及时透明地为业务方呈现需求的总体进度。同时可以查看需求对应的代码，便于开展审计工作。

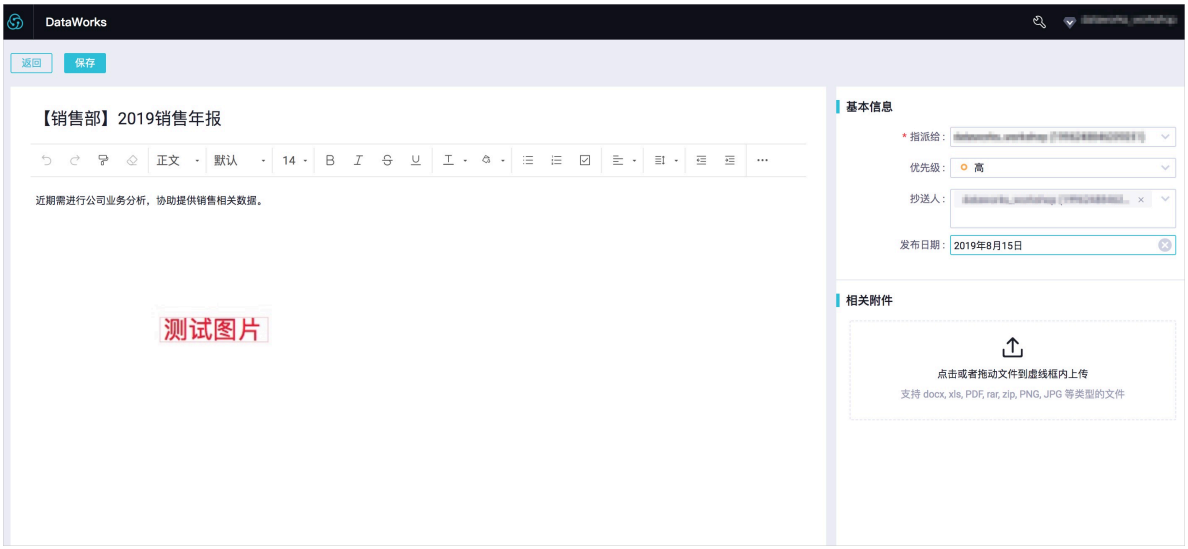
17.2 新建需求

阿里云主/子账号登录需求管理模块后，即可新建需求，并将需求指派给主账号下的其他成员作为责任人。

操作步骤

1. 登录DataWorks控制台，单击相应工作空间后的进入数据开发。
2. 单击左上角的图标，选择全部产品 > 需求管理。
3. 单击新建需求，进入新建需求页面。

4. 填写需求名称、需求内容，设置基本信息并上传相关附件，单击保存。



说明：

> 必须指派需求的负责人，可以选择当前阿里云主账号下任意一个主/子账号，无论其是否被加入到DataWorks工作空间中。

5. 保存新建的需求后，即可跳转至需求列表进行查看。

17.3 搜索需求

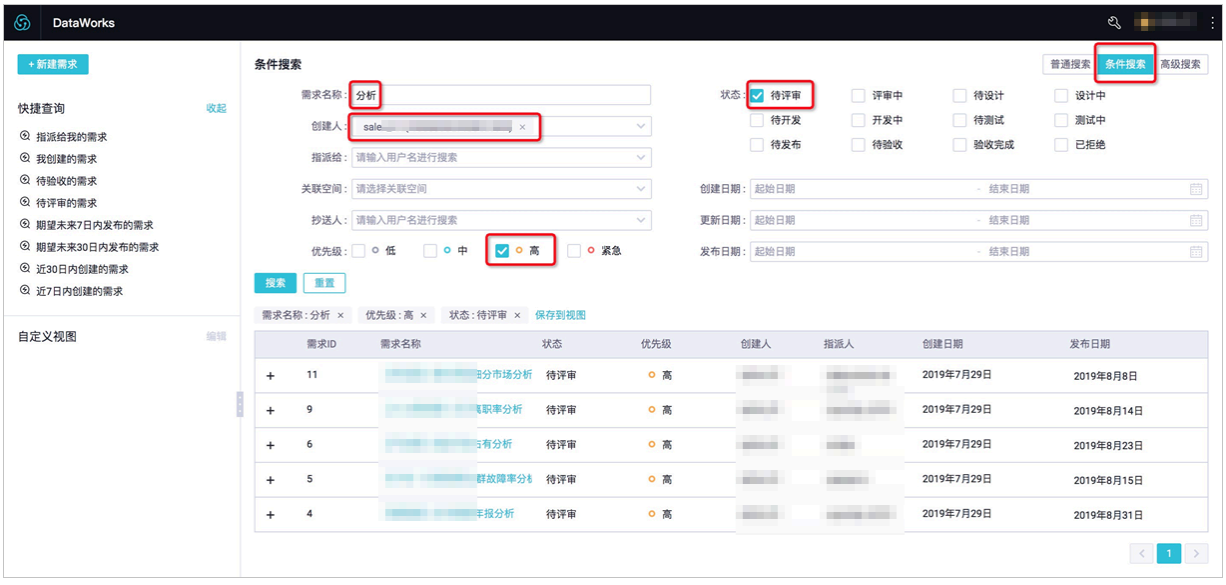
您可以通过普通搜索、条件搜索和高级搜索，查找所需要的已创建的需求。

普通搜索

单击右上角的普通搜索，即可通过输入需求名称进行搜索。

条件搜索

单击右上角的条件搜索，即可通过筛选需求的基本信息进行搜索。



高级搜索

单击右上角的高级搜索，即可对多个搜索条件进行且、或关系配置，来解决需求数量庞大、搜索条件复杂的搜索场景。

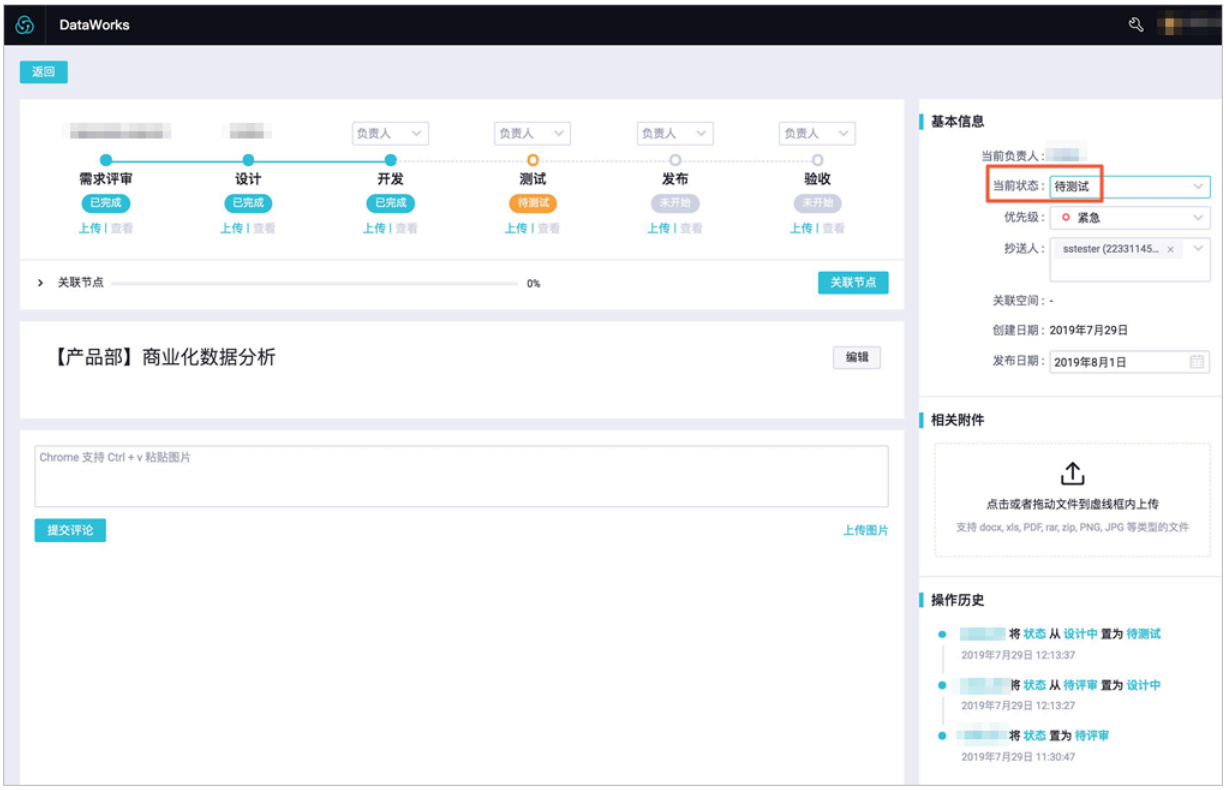
功能	说明
+规则	单击+规则，即可在当前搜索层级下，添加一项搜索条件。
+组	单击+组，即可在当前搜索层级的下一级，添加一项搜索条件。
且/或	决定当前层级下各搜索条件之间的关系。选择且，表示取每个搜索条件所得结果的交集。选择或则表示取每个搜索条件所得结果的并集。 例如图中4、5两个条件间的关系为且，1、2和3的关系也为且，则结果为1、2、4和5的交集。 如果有更复杂的搜索场景，您可以通过添加更多组，并设置合理的且/或关系来实现。

17.4 管理需求

需求创建完成后，被指派责任人、被抄送人可以根据实际情况，在需求详情页对需求进行管理。

里程碑

您可以根据当前需求所处的实际阶段，选择当前状态，来展示该需求当前的状态。



当前状态包括需求评审、设计、开发、测试、发布和验收个阶段，请参见数据仓库研发规范来设置需求的状态。

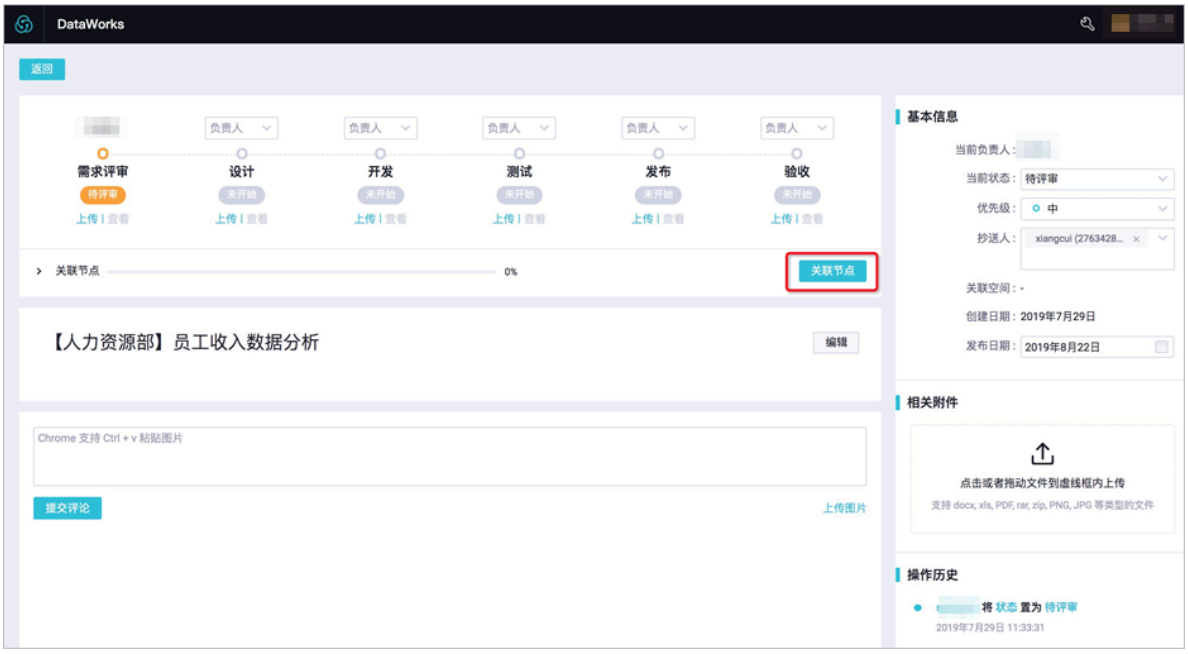
同时，您可以指定不同人员作为需求在每一阶段的负责人，将人员和阶段职责一一对应，便于分工协作。

关联节点

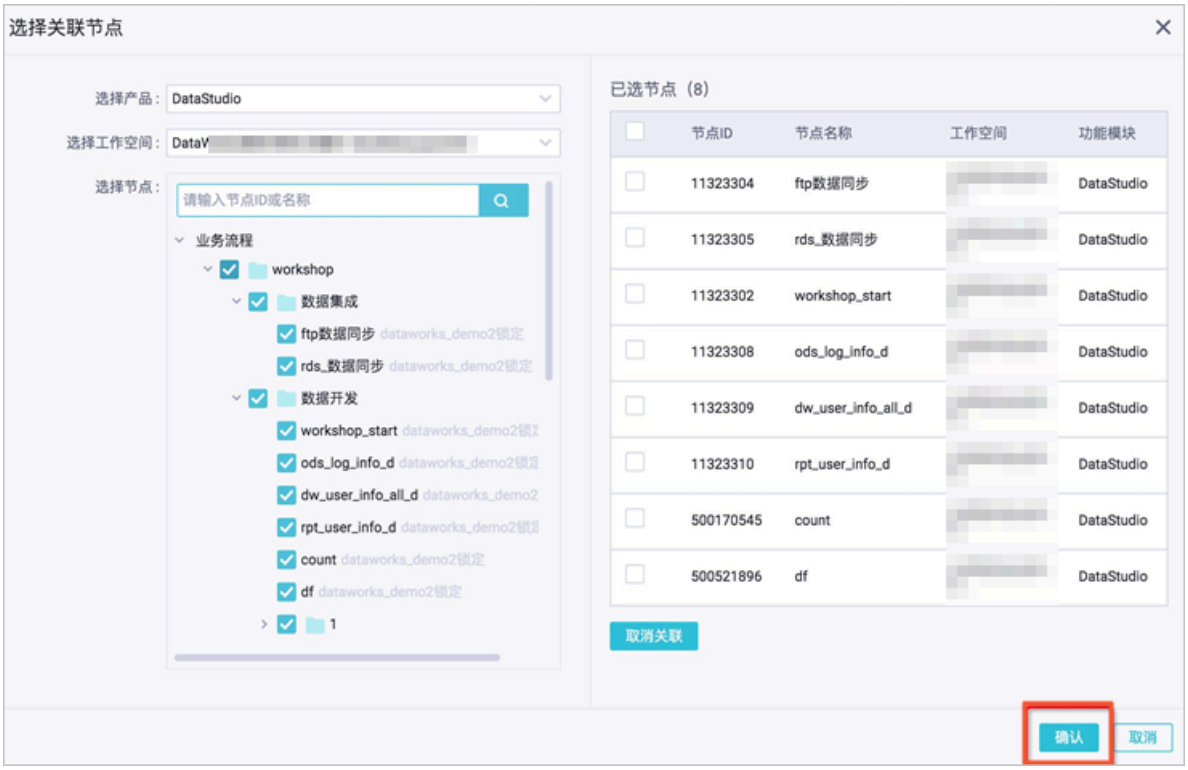
如果该需求在实现过程中，需要涉及在DataWorks中完成相关数据开发工作，您可以将数据开发模块中的任务、机器学习中的实验等对象关联至该需求。

> **说明:**
> 关联节点后，必须保证所有节点达到已发布的状态，需求状态才可以切换至待验收或验收完成的状态。

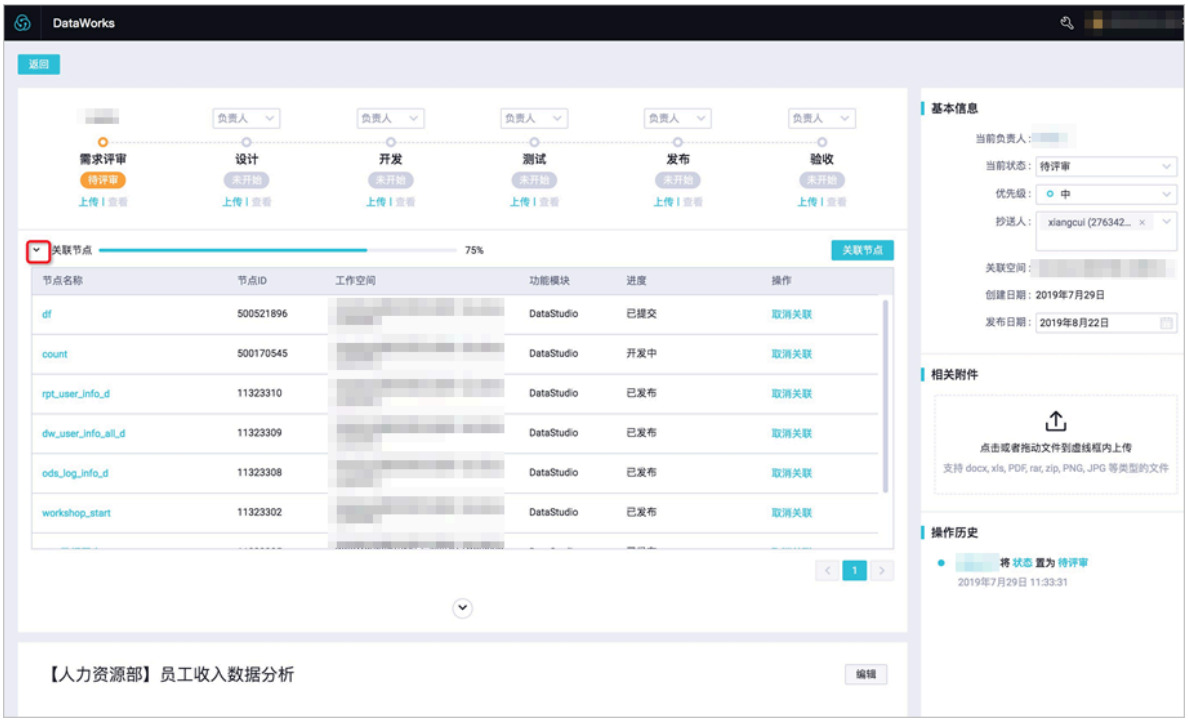
1. 单击关联节点。



2. 选择需要关联至需求的节点，单击确认。



3. 单击左侧的箭头，即可查看关联节点的发布进度。

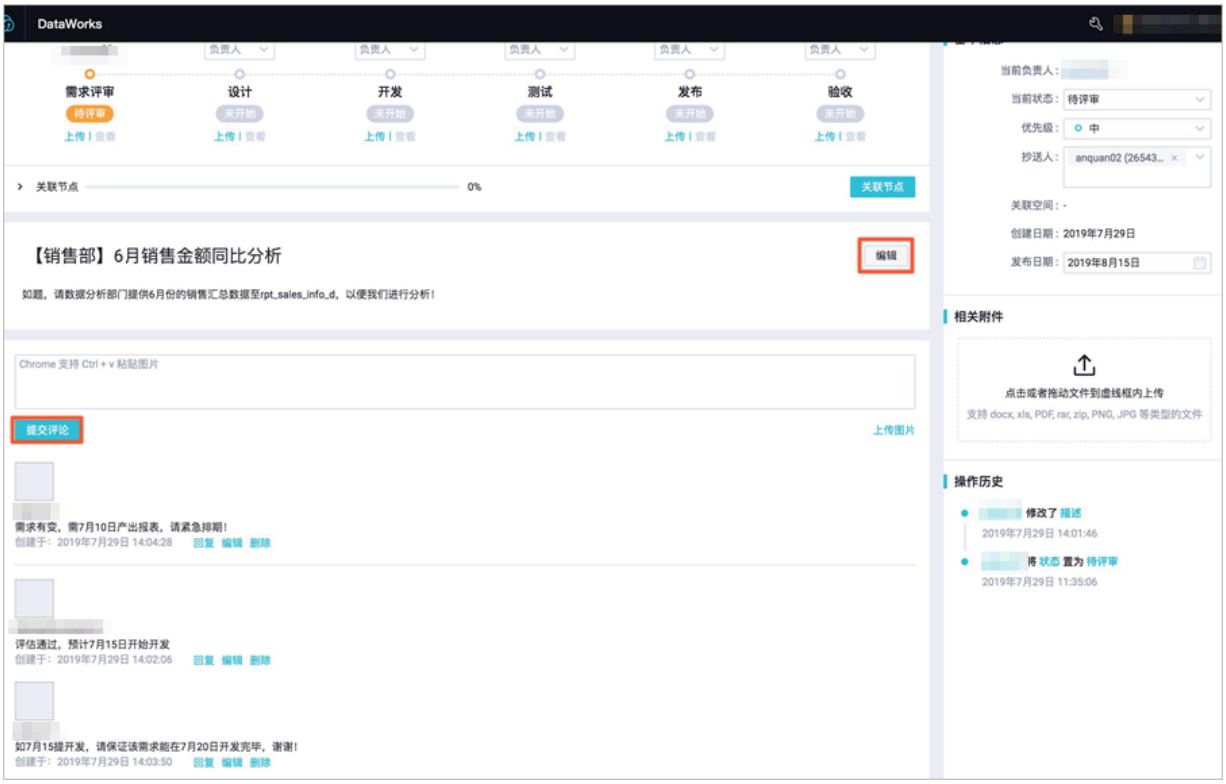


说明:

- 仅支持选择当前登录的阿里云主/子账号所在DataWorks工作空间的节点。
- 如果关联的是DataWorks简单模式工作空间的节点，且在数据开发模块已提交，则此处的状态为已发布。
- 如果关联的是DataWorks标准模式工作空间的节点，且在数据开发模块已发布，则此处的状态为已发布。
- 如果关联的是来自机器学习平台的实验，则默认为已发布状态。
- 对于已发布至运维中心（调度系统）的节点，如果进行下线操作，则此处状态默认退回至已提交。

编辑与评论需求

需求提交后，您仍可以对需求的正文进行修改或提交相关评论，便于信息的补充与传递。


DataWorks
需求评审
设计
开发
测试
发布
验收
关联节点
【销售部】6月销售金额同比分析
编辑
如题，请数据分析部门提供6月份的销售汇总数据至rpt_sales_info_d，以便我们进行分析！
提交评论
上传图片
需求有变，需7月10日产出报表，请紧急排期！
评估通过，预计7月15日开始开发
如7月15提开发，请保证该需求能在7月20日开发完毕，谢谢！
当前负责人：
当前状态：待评审
优先级：中
抄送人：
关联空间：-
创建日期：2019年7月29日
发布日期：2019年8月15日
相关附件
点击或者拖动文件到虚线框内上传
操作历史

18 资源优化

18.1 资源优化概述

资源优化从数据存储、数据计算和数据采集3个领域进行扫描，帮您扫描出可以优化的表和节点，从而合理、高效地运行DataWorks上的任务。

> 说明:
> 目前仅华东2（上海）地域支持资源优化，正在内部邀测中。如果您有相关需求，请提交工单进行申请。

资源优化以列表的形式为您展示不规范的使用问题，您可以根据对应优化项的解读，来进行优化。

资源优化页面主要面向个人和管理员2种用户群体：

- 个人资产优化页面主要供个人使用，为您展示个人名下的总任务数、总表数、优化趋势和个人资产优化模块。
- 工作空间资产优化页面主要供管理员使用，为管理员展示当前工作空间下的总任务数、总表数、优化趋势、可优化计算管理个人、可优化存储管理个人和工作空间资产优化模块。

  管理员可以通过可优化计算管理个人和可优化存储管理个人的排行信息，联系对应的资产责任人进行优化。

> 说明:
> 个人资产和工作空间管理这两个查看维度，需要添加一下数据的更新时间，目前个人资产优化和工作空间资产优化页面的总实例数、总表数的数据是离线（T+1）更新的，会存在数据延迟的情况。

18.2 个人资产优化

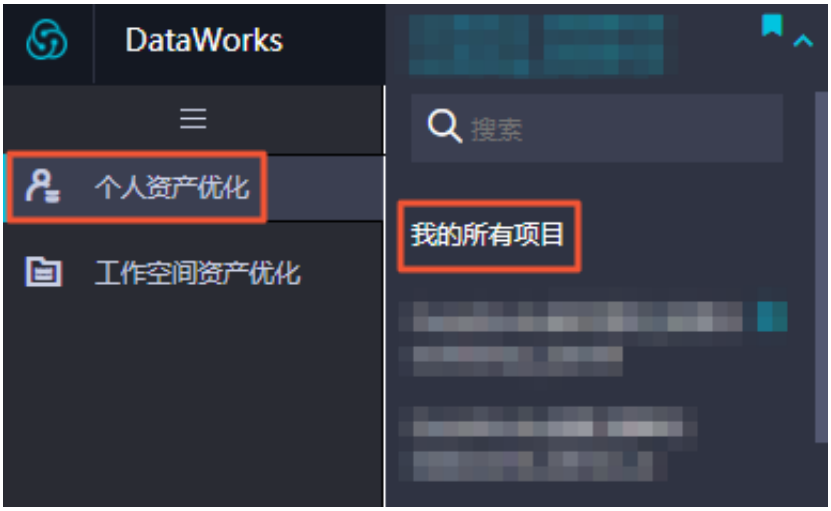
个人资产优化页面主要供个人使用，为您展示个人名下可优化的任务和表。

进入个人资产优化页面

1. 登录DataWorks控制台，单击对应工作空间操作栏中的进入数据开发。
2. 单击左上角的图标，选择全部产品 > 资源优化。

3. 单击左侧菜单栏中的个人资产优化，即可进入个人资产优化页面，查看个人名下的全部任务和表的总数。

   您可以单击顶部工作空间下拉框，选择相应的工作空间，也可以选择我的所有项目。



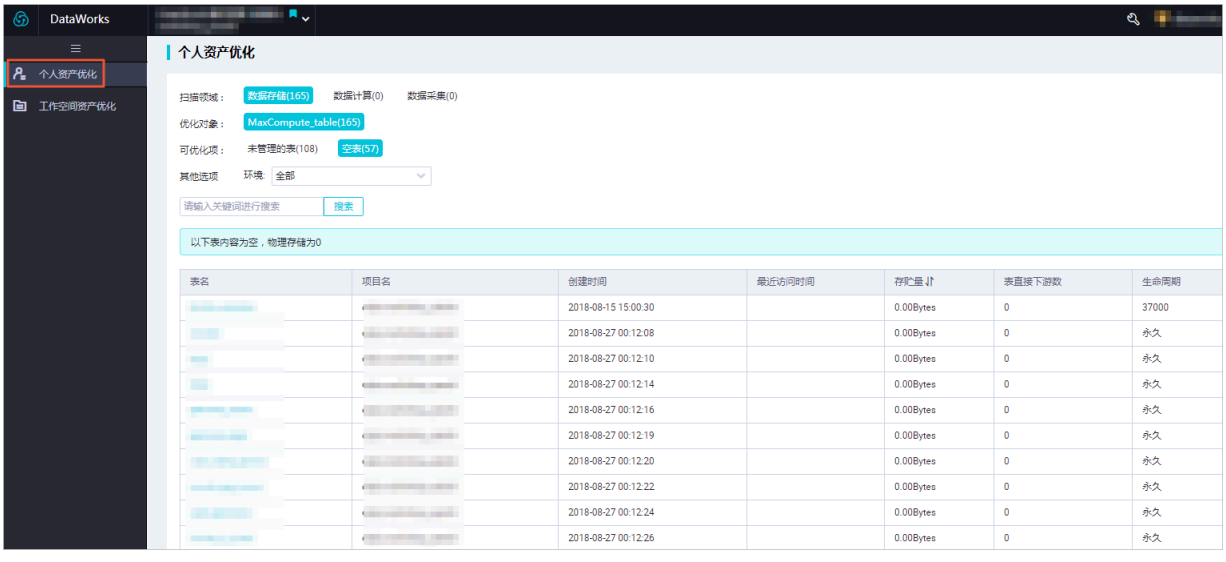
查看个人资产优化页面

优化趋势为您展示最近10天内可优化项的变化趋势，您可以在此查看最近完成可优化项的数量。

说明:

优化趋势中的数据是离线计算生成的，您可以查看最新的日期，以获取数据的最近更新时间。

个人资产优化从数据存储、数据计算和数据采集3个领域进行扫描。目前，数据存储和数据计算扫描的是MaxCompute的表和SQL任务。数据采集扫描的是写入至MaxCompute中的同步任务。



 **说明:**
DataWorks支持开发环境和生产环境隔离的标准工作空间模式，即一个DataWorks工作空间支持底层有2个MaxCompute项目，此时您可以通过环境进行筛选。

扫描领域	优化对象	可优化项	说明
数据存储	MaxCompute_Table	未管理的表	未管理的表需要满足以下2个校验条件： · 未设置生命周期的表。 · 最近一个月未在DataWorks上访问的非分区表。 同时满足上述条件的表，会被扫描出来。针对上述扫描条件，您可以通过设置表的生命周期，解决上述扫描问题。表的生命周期详情请参见#unique_727。  **说明:** 表的生命周期到期后，会回收表数据，请谨慎操作。
		空表	存储量为0的表即为空表。不建议您直接删除表，推荐您根据表的创建时间，对早期创建的表进行审计。
数据开发	MaxCompute任务	冲突任务	扫描出与其它任务写入同一张表，会导致非预期结果的任务。
		数据倾斜	扫描出存在部分实例的处理数据量，及时间超过其它实例的情况，会导致整体任务执行时间变长的任务。
数据采集	数据同步节点	持续导入一致	扫描出连续15天导入数据量持续一致的数据同步节点，请关注源数据是否不再更新。
		导入为空	扫描出导入数据量持续为0的数据同步节点，您可以暂停或下线该节点。
		同源导入	扫描出有相同的数据源，存在重复导入MaxCompute的情况，会导致资源浪费的数据同步节点。您可以合并作业。

18.3 工作空间资产优化

工作空间资产优化页面主要以项目管理员为维度，为项目管理员展示指定工作空间下的可优化项信息，以及可优化计算管理个人和可优化存储管理个人的排行信息。

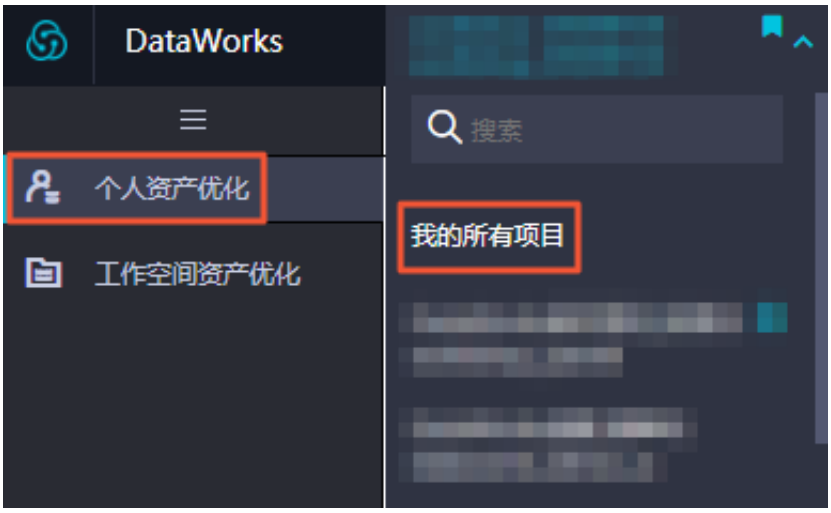
说明:

可优化计算管理个人和可优化存储管理个人最多展示10行信息。

进入工作空间资产优化页面

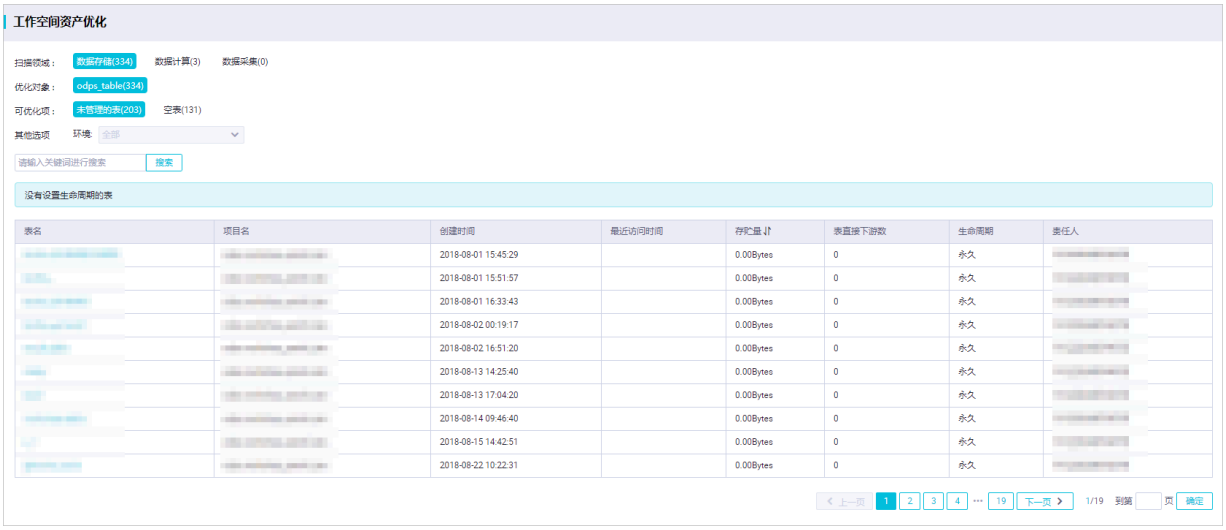
1. 登录DataWorks控制台，单击对应工作空间操作栏中的进入数据开发。
2. 单击左上角的图标，选择全部产品 > 资源优化。
3. 单击左侧菜单栏中的工作空间资产优化，即可进入工作空间资产优化页面，查看个人名下的全部任务和表的总数。

   您可以单击顶部工作空间下拉框，选择相应的工作空间，也可以选择我的所有项目。



查看工作空间资产优化页面

工作空间资产优化和个人资产优化的可优化项基本一致，只是查看的视角不同。



扫描领域	优化对象	可优化项	说明
数据存储	MaxCompute_Table	未管理的表	未管理的表需要满足以下2个校验条件： · 未设置生命周期的表。 · 最近一个月未在DataWorks上访问的非分区表。 同时满足上述条件的表，会被扫描出来。针对上述扫描条件，您可以通过设置表的生命周期，解决上述扫描问题。表的生命周期详情请参见#unique_727。 说明: 表的生命周期到期后，会回收表数据，请谨慎操作。
		空表	存储量为0的表即为空表。不建议您直接删除表，推荐您根据表的创建时间，对早期创建的表进行审计。
数据开发	MaxCompute任务	冲突任务	扫描出与其它任务写入同一张表，会导致非预期结果的任务。
		数据倾斜	扫描出存在部分实例的处理数据量，及时间超过其它实例的情况，会导致整体任务执行时间变长的任务。
数据采集	数据同步节点	持续导入一致	扫描出连续15天导入数据量持续一致的数据同步节点，请关注源数据是否不再更新。
		导入为空	扫描出导入数据量持续为0的数据同步节点，您可以暂停或下线该节点。

扫描领域	优化对象	可优化项	说明
		同源导入	扫描出有相同的数据源，存在重复导入MaxCompute的情况，会导致资源浪费的数据同步节点。您可以合并作业。

19 MaxCompute管家

19.1 MaxCompute预付费资源监控工具-CU管家

MaxCompute管家为系统运维人员提供系统状态、资源组分配和任务监控三个功能，本文将为您介绍Maxcompute管家的使用方法。

前提条件

使用MaxCompute管家前，您需要购买MaxCompute预付费CU资源，适用于60CU以上的用户。

> **说明:**
>
> 当前政务云不支持CU管家。
>
> CU过小无法发挥计算资源及管家的优势。如果禁用主账号的AK，会导致无法用相应的子账号管理CU管家。

满足上述条件，您即可登录DataWorks控制台，选择我的资源 > CU管理。

系统状态

您可以通过系统状态了解CU计算资源和存储的消耗情况。

操作	说明
Quota选择	可以选择所查看的资源组，根据选择的资源组，展示当前资源组的消耗信息和当前存储量。
选择时间段	可以选择查看所选资源组的时间区间，选择的区间不同，资源组数据展示的粒度不同（计算资源CU每6分钟采集一次/存储每一小时采集一次）。

Quota设置

Quota指资源组。例如您购买了100CU，表示您全部Quota的额度是100CU。您可以通过大数据管家来新建Quota，这样就可以对Quota进行资源分配。运维人员可以很方便地将各个项目的资源隔离，保证重要项目的计算资源充沛。

操作	说明
新建Quota	新建一个Quota组，建好后可以通过移动项目功能，来将项目移动到Quota下。创建的Quota可以删除，但如果当前Quota下有项目，则无法删除。

操作	说明
修改CU消耗	建好的Quota支持修改CU最小消耗值。
移动项目	支持将当前Quota下的项目移动至其他Quota下，新建的Quota即可通过这个移动项目这个功能来做到资源隔离。
删除	支持删除Quota组，如果当前Quota下有项目，则无法删除。

说明:

计算资源CU升级和降配时，默认Quota组Max、Min会相应变化，其他组不会改变配置。如果降配时，剩余默认Min组小于降配额度，会降配失败。Max为最大分配的资源，Min为最小保障资源。

Quota组配置示例

例如60CU由两个部门使用。

- 资源组独享：【MaxCU,MinCU】，A组【40,40】，B组【20,20】。
- 资源组倾斜：【MaxCU,MinCU】，A组【60,40】，B组【40,20】。

不同Quota组之间的调度顺序

目前不同的Quota组暂不支持调度优先级设置，Quota的使用遵循先到先得、不抢占的原则。例如60CU由两个部门使用，分配如下。

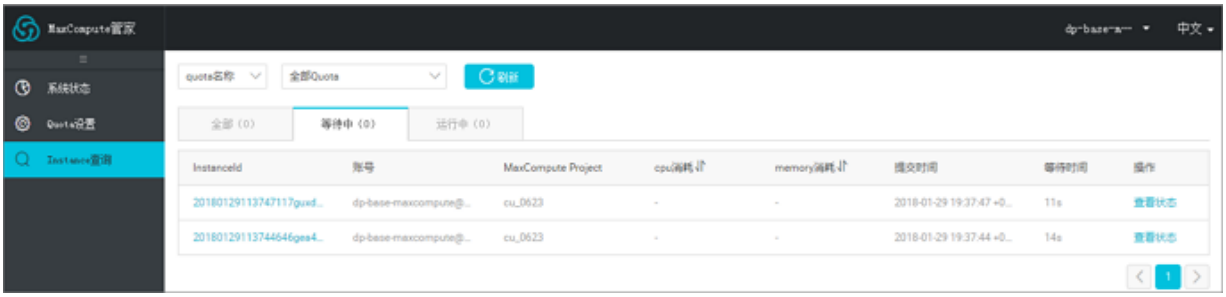
【MaxCU,MinCU】，A组【40,20】，B组【30,10】

假设A组先使用，占用了40CU的资源，则后使用的B组只能使用20CU的资源，此时B组无法抢占A组已抢占的资源。

假设A组在使用一段时间后，释放了40CU中的10CU资源，则此时B组可以占用30CU资源。

Instance查询

您可以通过计算任务监控，了解当前任务排队状态，资源被哪些任务抢占，然后对任务进行分析或停止。



可以根据Quota组名称和项目名称两个维度来筛选，精确搜索。

- InstanceId：每个MaxCompute任务都会有一个Instance，通过单击InstanceId可以跳转到Logview页面，查看具体的任务进度。

  查看Logview的方法请参见Logview查看。
- 账号：运行这个MaxCompute任务的操作人，可以根据这个账号信息找到任务所属的责任人，如果该任务占用太多资源而影响其他任务的运行，可以与该责任人联系，是否停止该任务。

  停止任务的方法请参见实例操作中的Kill Instance。
- MaxCompute Project：当前Instance所属的项目名称。
- cpu消耗：当前Quota组实际使用的CPU资源比例。
- memory消耗：当前Quota组实际使用的内存资源比例。
- 提交时间：当前Instance的提交时间。
- 等待时间：等待运行资源的时长。
- 操作：可以在此处查看当前Instance的状态，此处会显示当前状态和历史状态。