

大数据开发门户帮助手册

最新修改时间：2019/03/25

产品介绍

1 什么是大数据开发门户

1.1 开发平台

1.1.1 开发中心

1.1.2 发布管理

1.1.3 运维中心

1.1.4 监控管理

1.1.5 数据管理

1.2 标签管理

1.3 数据服务

1.4 幻数

1.5 审计日志

1.6 基础组件

1.6.1 用户关系管理

1.6.2 资源管理

1.6.3 数据源管理

1.6.4 任务管理

2 产品架构图

3 名词解释

快速入门

1 注册

2 登录

3 管理控制台

4 进入开发

5 数据准备

6 新建资源

7 新建表

8 数据导入

9 数据加工

用户指南

1 注册登录

1.1 注册

1.2 登录

1.3 进入控制台

2 用户中心

2.1 基本信息

2.2 账户安全

3 用户关系管理

4 资源管理

4.1 资源组管理

4.2 计算引擎管理

5 数据源管理

- 5.1 MySQL数据源
- 5.2 HDFS数据源
- 5.3 Kafka数据源
- 6 开发中心
 - 6.1 项目管理
 - 6.1.1 新建项目
 - 6.1.2 资源授权
 - 6.1.3 初始化项目
 - 6.1.4 项目配置
 - 6.2 进入开发
 - 6.3 工作流
 - 6.3.1 新建
 - 6.3.2 属性配置
 - 6.3.3 删除
 - 6.3.4 重命名
 - 6.3.5 运行
 - 6.3.6 提交
 - 6.4 工作流（可视化组件模式）
 - 6.5 离线任务
 - 6.5.1 新建任务
 - 6.5.2 复制任务
 - 6.5.3 删除任务
 - 6.5.4 格式化
 - 6.5.5 代码检查
 - 6.5.6 运行
 - 6.5.7 属性配置
 - 6.5.8 提交
 - 6.5.9 任务提交发布状态
 - 6.6 实时任务
 - 6.7 临时任务
 - 6.8 资源文件
 - 6.9 函数开发
 - 6.10 表管理
 - 6.11 任务类型
 - 6.11.1 Shell
 - 6.11.2 数据同步
 - 6.11.3 非结构化数据同步
 - 6.11.4 Python
 - 6.11.5 Perl
 - 6.11.6 Hive2
 - 6.11.7 PySpark
 - 6.11.8 Spark
 - 6.11.9 SparkSQL
 - 6.11.10 Presto
 - 6.11.11 Storm
 - 6.11.12 MapReduce
 - 6.11.13 Hbase
 - 6.11.14 MySQL
 - 6.11.15 Oracle

- 6.11.16 GreenPlum
- 6.11.17 SQL Server
- 6.11.18 PPAS
- 6.11.19 Tensorflow
- 6.11.20 FlinkSQL
- 6.11.21 Flink
- 6.11.22 实时数据同步
- 6.11.23 Spark Streaming
- 7 发布管理
 - 7.1 进入发布管理
 - 7.2 创建发布包
 - 7.3 审核发布包
- 8 运维中心
 - 8.1 运行总览
 - 8.2 工作流实例
 - 8.3 离线实例
 - 8.4 离线任务
 - 8.5 实时任务
- 9 监控管理
 - 9.1 基线管理
 - 9.2 基线告警
 - 9.3 数据质量告警
- 10 数据管理
 - 10.1 全局概览
 - 10.2 元数据管理
 - 10.3 数据权限
 - 10.4 数据质量
 - 10.5 数据目录
 - 10.6 术语项管理
 - 10.7 Topic管理
 - 10.8 HDFS管理
 - 10.9 Hbase查询
- 11 标签管理
 - 11.1 后台类目
 - 11.1.1 进入后台类目
 - 11.1.2 创建类目
 - 11.1.3 创建标签
 - 11.1.4 绑定数据
 - 11.2 前台类目
 - 11.2.1 进入前台类目
 - 11.2.2 创建类目
 - 11.2.3 选择标签
 - 11.3 标签总览
- 12 数据服务
 - 12.1 创建数据服务
 - 12.2 数据配置
 - 12.2.1 选择前台类目
 - 12.2.2 数据同步
 - 12.3 服务开发

12.3.1 创建仪表盘

12.3.2 创建图表

12.3.3 配置图表

12.3.4 操作仪表盘

13 幻数

13.1 项目管理

13.1.1. 功能说明

13.1.2. 创建项目

13.1.3. 编辑项目

13.1.4. 删除项目

13.2 标签表

13.2.1 功能说明

13.2.2 添加标签数据表

13.2.3 删除标签数据表

13.2.4 搜索标签数据表

13.2.5 数据预览

13.2.6. 标签预览

13.3 仪表盘

13.3.1 功能说明

13.3.2 添加仪表盘

13.3.3 编辑仪表盘

13.3.4 删除仪表盘

13.3.5 添加目录

13.3.6 重命名目录

13.3.7 删除目录

13.3.8 添加图表

13.3.9 编辑图表

13.3.10 删除图表

13.3.11 导出图片

13.3.12 导出Excel

13.3.13 全屏查看

13.3.14 表格组件

13.3.15 指标卡组件

13.3.16 条形图组件

13.3.17 折线图组件

13.3.18 饼图组件

13.3.19 双轴图组件

13.3.20 批量选择标签

13.3.21 维度对比

13.3.22 设置数值格式

13.3.23 图表下钻

13.3.24 图表预警

13.3.25 图表大小设置

13.3.26 图内筛选器

13.3.27 筛选器

13.4 图表授权

13.4.1 功能说明

13.4.2 添加图表授权

13.4.3 编辑图表授权

13.4.4 删除图表授权

14 审计日志

15 任务管理

- 15.1 (创建者) 创建任务
- 15.2 (接收者) 接受任务
- 15.3 (接收者) 提交任务
- 15.4 (创建者) 重新下发任务
- 15.5 (创建者) 编辑、删除任务
- 15.6 (创建者) 查看操作日志

案例实战

1 概述

- 1.1 背景介绍
- 1.2 操作流程

2 开发平台

- 2.1 项目列表
- 2.2 进入开发
- 2.3 数据维度(DIM)
- 2.4 原始数据层(ODS)
- 2.5 数据仓库层(DWD)
- 2.6 数据标签层(TDM)
- 2.7 数据应用层(ADM)
- 2.8 配置基线 (可跳过)
- 2.9 运行
- 2.10 提交发布

3 运维中心 (可跳过)

4 数据管理 (可跳过)

- 4.1 配置数据质量
- 4.2 配置生命周期
- 4.3 查看数据血缘

5 数据源授权

6 标签管理

- 6.1 后台类目
 - 6.1.1 进入后台类目
 - 6.1.2 创建类目
 - 6.1.3 创建标签
 - 6.1.4 绑定数据
- 6.2 前台类目
 - 6.2.1 进入前台类目
 - 6.2.2 创建类目
 - 6.2.3 选择标签

7 数据服务

- 7.1 创建数据服务
- 7.2 数据配置
- 7.3 服务开发
 - 7.3.1 创建仪表盘
 - 7.3.2 创建图表
 - 7.3.3 配置图表

The screenshot shows a data dashboard with a sidebar on the left containing navigation icons and a list of tags. The main area features a filter configuration section with dropdown menus for dimensions (业主ID, 业主姓名, 业主年龄, 业主小区, 城市, 投诉位置) and numerical values (业主投诉次数). Below the filters is a data table with 7 columns: 业主ID, 业主姓名, 业主年龄, 业主小区, 城市, 投诉位置, and 业主投诉次数(求和). The table contains 3 rows of data.

业主ID	业主姓名	业主年龄	业主小区	城市	投诉位置	业主投诉次数(求和)	业主投诉次数(求和)
1	山强	27	汪街H座	厦门	卧室	4	4
1	山强	27	汪街H座	厦门	厨房	4	4
1	山强	27	汪街H座	厦	浴室	4	4

7.3.4 查看API

8 幻数

8.1 创建项目

8.2 添加标签数据表

8.3 添加仪表盘

8.4 添加图表

权限说明

常见问题(FAQ)

- 1 平台推荐使用哪种浏览器?
- 2 任务一直卡在调度中怎么处理?
- 3 创建资源时提示名称重复
- 4 SQL代码检查通过了, 运行却报错
- 5 如何创建全局变量?
- 6 代码中的参数未定义在运行参数中
- 7 什么任务可以被设为上游任务?
- 8 代码如何批量注释?

产品介绍

1 什么是大数据开发门户

大数据开发门户提供从数据同步、数据开发、数据服务的一站式大数据全链路解决方案, 支持Hive、Spark、Hbase、GreenPlum等业界流行的大数据计算引擎, 并可快速扩展。通过此平台可构建PB级数据仓库及进行数据价值的深度挖掘, 来实现超大规模数据的资产化。平台采用“开箱即用”的使用方式, 让用户无需再关心底层集群的搭建和运维, 能极大提高数据开发和应用的效率。

大数据开发门户提供以下产品:

1.1 开发平台

1.1.1 开发中心

开发中心是大数据集成开发工具, 可供开发人员进行任务开发、调试和发布, 可支持多人协同开发。

1.1.2 发布管理

发布管理是将开发环境中的工作流和任务发布到生产环境中, 开发人员可批量发布任务, 管理员可对发布包进行审核。

1.1.3 运维中心

运维中心供运维人员进行各种数据运维操作，例如重跑、重跑下游、补数据、置成功、补下游等。

1.1.4 监控管理

监控管理主要工作是支持对开发任务进行全流程监控，对生产 workflow 进行管控，支持复杂的监控策略配置。

1.1.5 数据管理

数据管理包括元数据管理、数据权限、数据质量、数据目录、术语项管理和 Topic 管理，可对生产过程中表数据进行查看、分析、监控等操作，并对用户权限进行审核管理。

1.2 标签管理

针对于每一个行业可以定义一套行业通用的标准数据类目，基于标准数据类目可以通过沉淀的模型生成标准的标签类目，沉淀行业典型场景标签体系，完成数据资产化，助力客户快速建设自有数据资产。可实时了解标签的使用情况，体现数据资产的应用价值，实现数据业务化的目标。

1.3 数据服务

企业根据已有的标签资产结合不同的引擎，生成不同的服务来应用到不同的业务场景中。提供可视化交互式的操作界面快速生成服务 API，降低使用门槛给用户流畅简洁的操作体验，旨在让用户快速创建自己的数据应用，实现数据业务化的目标。

1.4 幻数

幻数是数据化运营平台，致力于帮助企业快速实现数据资产可视化，为企业数据化运营赋能，提升企业竞争力。

1.5 审计日志

通过审计日志，用户可以记录与产品服务相关的操作事件，可用于支撑安全分析、资源跟踪和问题定位等常见应用场景。

1.6 基础组件

1.6.1 用户关系管理

用户关系管理用来管理公司的部门和人员，同时可对人员进行产品使用授权。

1.6.2 资源管理

资源管理是对开发门户中的执行代理服务和底层计算引擎的统一管理，并可对资源进行授权。

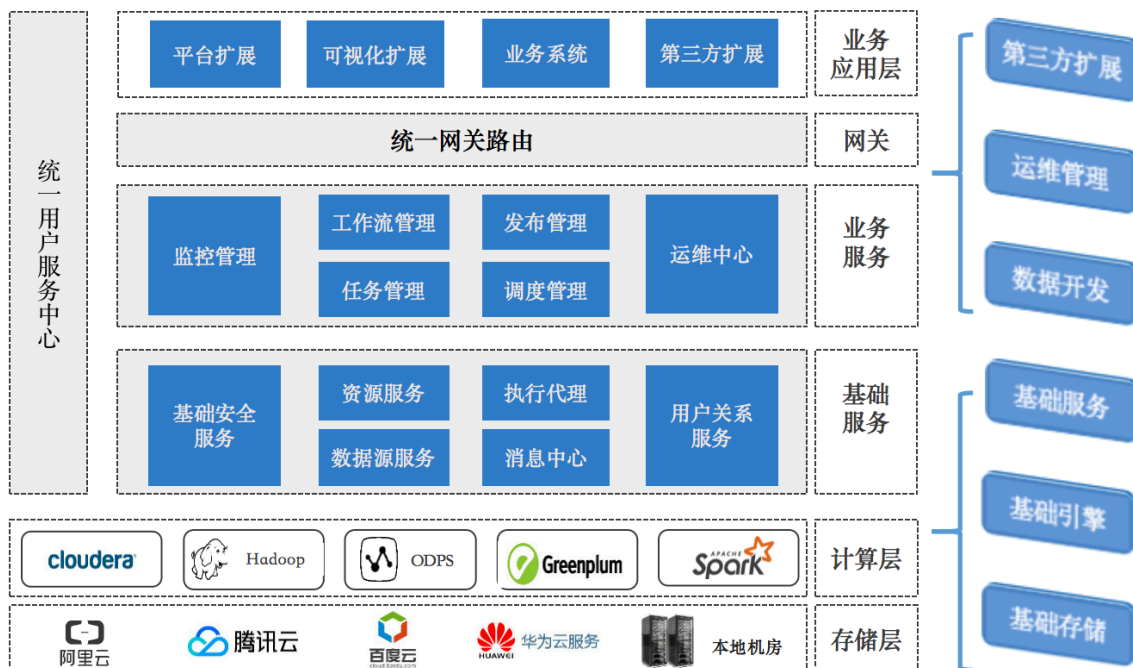
1.6.3 数据源管理

数据源管理支持对数据源进行集中管理，并可对数据源进行授权。

1.6.4 任务管理

任务管理是创建者通过分派任务的方式，将产品任务、数据需求传达给其他团队或同事，所有参与人员都能及时看到任务最新的进展和操作记录，是一种提高团队间协作效率工具类的产品。

2 产品架构图



3 名词解释

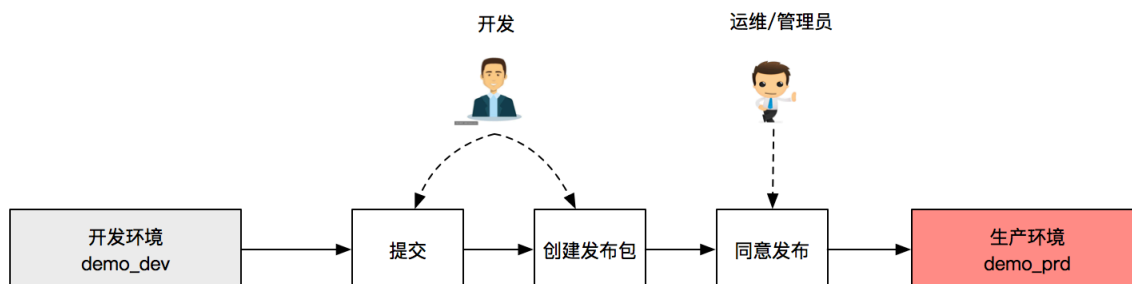
- 项目

项目(Project)是一组任务、资源、函数的集合，同一租户下，对于不同的业务建议分别创建不同的项目。例如对于地产和金融两个业务，可分别新建estate、finance两个项目。

- 环境

同一项目下，可创建多个环境进行数据开发，每个环境拥有独立的Hive数据库、Yarn调度队列，甚至不同的Hadoop集群。

在新建项目时，平台默认创建开发和生产两个环境，开发环境用于用户开发、调试任务，生产环境也即线上环境，系统默认会按天进行周期调度执行任务。生产环境不允许用户直接操作任务、资源和函数，必须在开发环境下进行新建、修改或删除，再经过提交、创建发布包、同意发布三个操作后，才可同步到生产环境。



- 资源组

一个资源组由一个或多个服务器组成，在这些服务器上部署了数据源服务、数据开发服务、算法开发服务、notebook服务、实时数据同步服务。开发平台通过访问服务器调用服务才能完成相关任务。

- 连接信息

连接信息是由计算引擎地址、数据库、用户名、密码生成，可通过配置一个连接来指定一个任务进行数据加工所使用的数据库。

- 计算引擎

目前平台支持的计算引擎类型包括：Hive、LibrA、MySQL、GreenPlum、Oracle等，计算引擎代表一个项目的计算类型，通过配置计算引擎和数据库，可为一个项目指定其开发环境。

- 调度队列

yarn通过设置多个调度队列来实现资源隔离，本平台允许多个项目空间共享整个集群，可以通过为每个项目设置调度队列来获得集群的一部分计算能力。

- 工作流

工作流(Flow)是完成特定业务的一组任务集合，工作流内部的任务可互相依赖，组成有向无环图(DAG)。

- 任务

任务(Task)是平台的最小运行单元，目前支持Shell、Hive、Spark、Presto、Flink等11种任务类型。任务包含离线任务、实时任务、临时任务，其中临时任务是一种特殊的任务，用于一次性查询或分析数据，因此不支持调度和基线配置。

- 资源

资源(Resource)是指用户上传的文件，可被用于任务、函数开发中。目前支持三种类型的资源文件：jar、txt、python。

- 函数

函数(Function)指Hive、Spark、Presto、Flink等系统中的函数。除计算引擎内置的函数外，平台支持用户创建基于Hive的自定义函数(User Defined Function，以下简称UDF)，并可直接用于Hive或SparkSQL类型的任务中。

- 实例

实例(Instance)指的是任务实例，任务的每次运行都会产生一个新实例。实例正常运行会经历待运行(Waiting)、运行(Running)、结束(Finished)三个阶段。

例如对于任务A，两次运行产生的实例ID分别是：T_630_20180301115903046_1、T_630_20180301120009801_1。

- 元数据(meta)

元数据是描述数据的数据，在传统的数据仓库中天然包含元数据库，在大数据平台中，元数据整合各个业务系统中的元信息，例如，库的基本信息，表的字段、分区、存储量、变更、新增、删除等信息。

- 血缘

血缘指的是数据表之间的上下游依赖关系，大数据开发门户平台中通过一系列的数据处理任务流程解决某种业务问题，这个过程中生成了一系列表，数据血缘就是用来描述这种表之间的依赖关系的。

- 生命周期

在大数据开发门户平台中，生命周期的意义类似于数据仓库，用来管理表或者表分区的保存时间。通过生命周期管理，可以管理表的存储。

- 类目

接入数据应用的规范的数据标准，对标签进行分类管理。通过前后台类目的分离管理，来面向不同用户的类目管理需求。

- 后台类目

后台类目是根据数据资产化过程所总结出的类目结构，相对稳定。

- 前台类目

前台类目则主要面向业务运营人员，更适用于瞬息万变的各应用场景，更具业务属性，相对变化性大。

- 标签

数据通过模型或者人工配置之后，生成的数据资产叫做标签。

- 引擎

将标签作用于实际场景的发动机，如标签通过分析引擎能实现各种聚合的报表，通过查询引擎可得到查询标签的其他标签信息等等。

- 服务

配置好的标签以及引擎的组合，能够对外提供使用满足一个特定的场景需求点。

- 系统参数

系统参数为系统预先定义好的参数，当任务发布到生产环境后，会根据当前时间自动替换系统参数的值。

目前支持的系统参数包括：cycTime, bizDate, bizMonth, bizQuar, lastDayofMon, 分别代表当前时间, 前一天, 前一个月, 前一个季度和上月最后一天。例如当前时间为2019/03/23 12:00:00, 则替换的结果如下：

`${cycTime}:20190323120000`

`${bizDate}:20190322`

`${bizMonth}:201902`

`${bizQuar}:2018_04`

`${lastDayofMon}:20190228`

快速入门

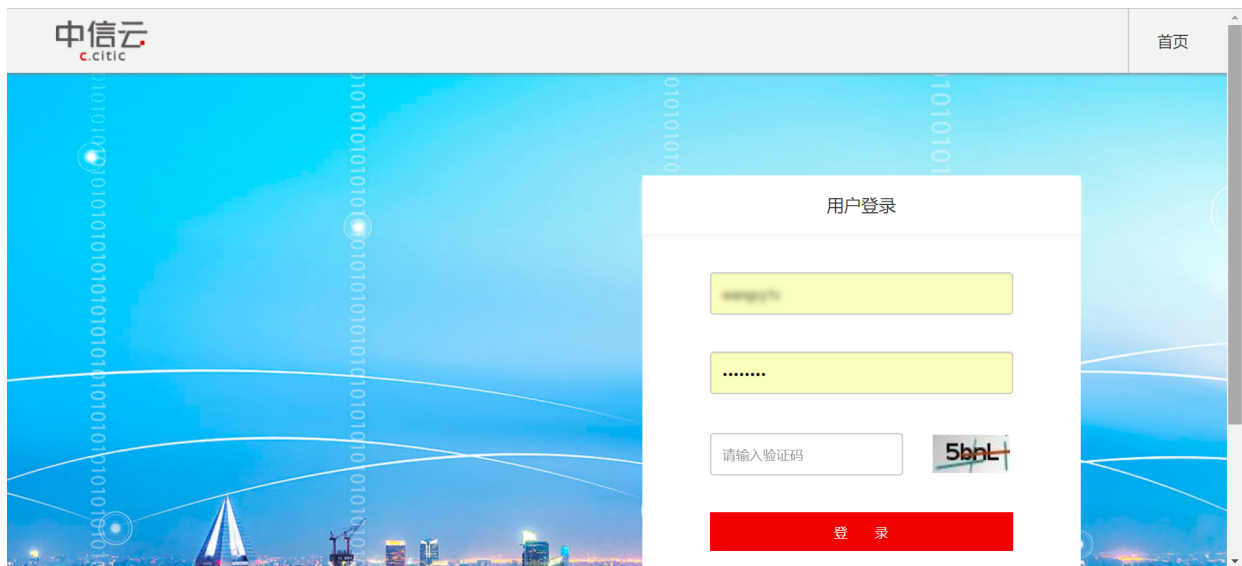
此部分将引导用户在开发中心中完成一个 `quick_start` 案例，即把学生基础数据导入到Hive表并对其进行加工处理，产出简单数据标签。

1 注册

用户在中信云网购买开发门户成功后，将在开发门户上自动注册以该用户为owner的租户账号，该账号具有控制台管理员权限。

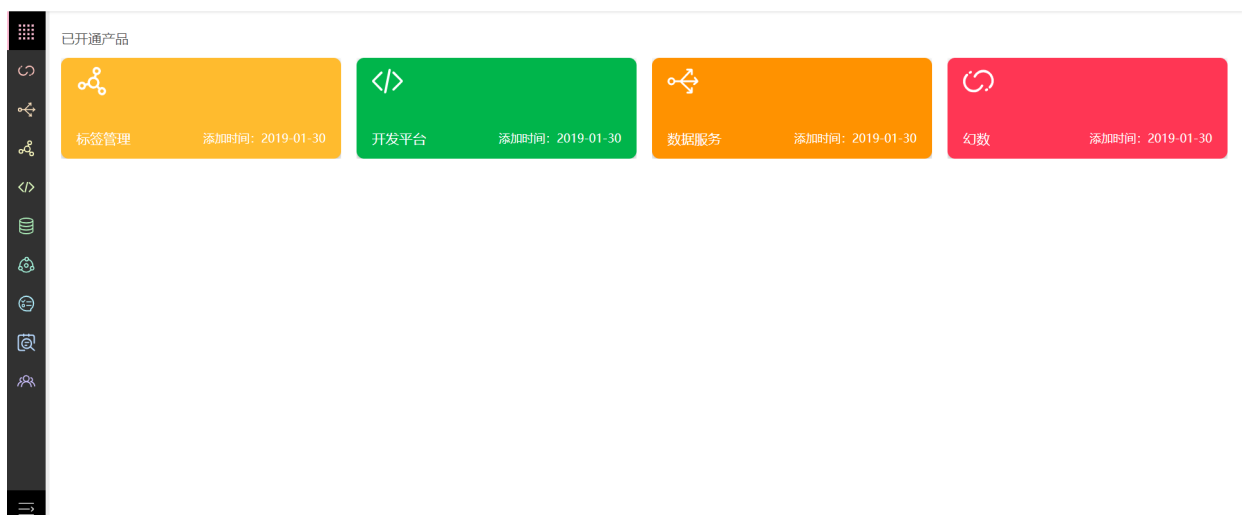
2 登录

通过浏览器访问开发门户，进入登录页面，填写手机号和密码完成登录，进入开发门户的控制台。



3 管理控制台

登录管理控制台后，显示产品如下图所示。



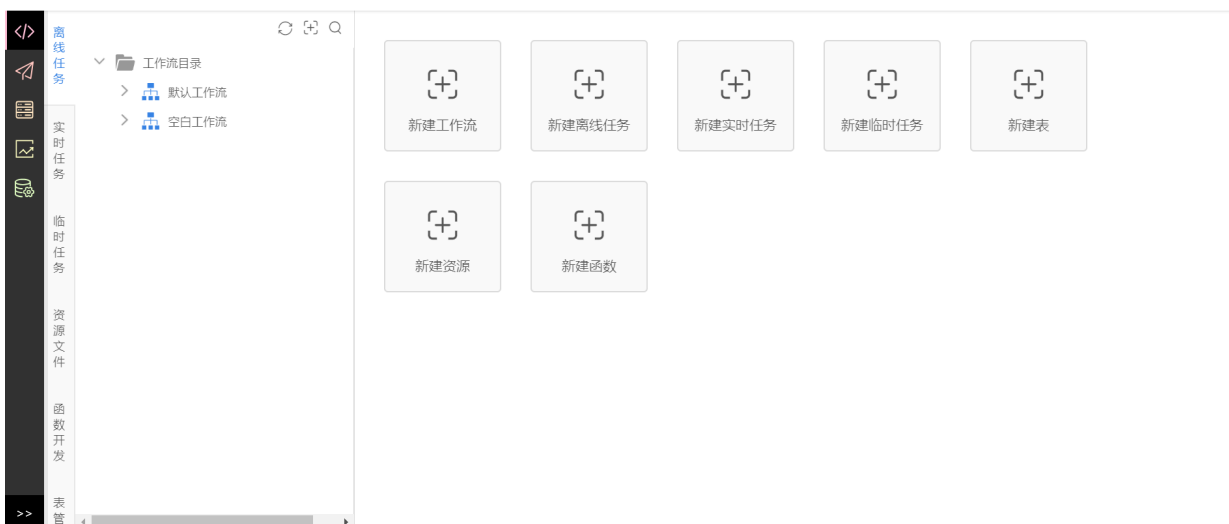
4 进入开发

在控制台中点击【开发平台】进入开发平台项目列表，本平台已为您部署好一个Hive类型的项目 `quick_demo`，如下图所示。

若没有，请先创建一个项目，并做好配置，详情请参考【[用户指南-项目管理](#)】部分。



点击进入开发快捷入口，进入开发中心页面。



5 数据准备

学生的数据信息如下，每列分别表示: 用户ID、姓名、年龄、体重，各列之间用","分割。在本地新建文本文件 `student_info.txt`，将下列数据拷贝至 `student_info.txt` 后保存。

在 `source` 文件夹中已准备好该txt文件：`student_info.txt`

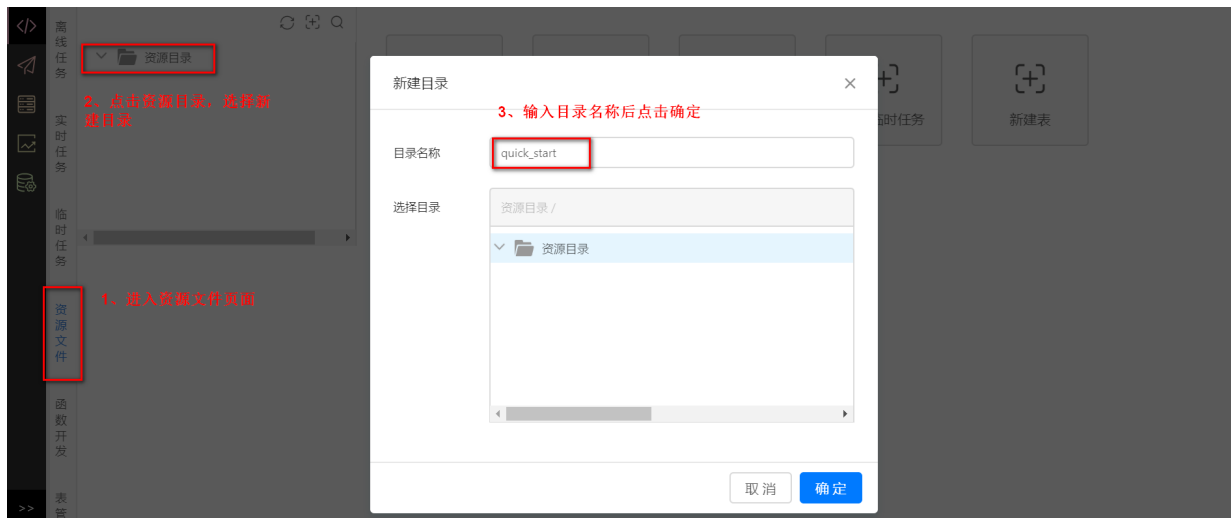
```
1,赵晓丽,23,50
2,王明,25,60
3,王勇,22,55
4,杜孟娟,21,50
5,李志刚,22,56
6,张林静,23,51
```

6 新建资源

(1) 新建资源目录

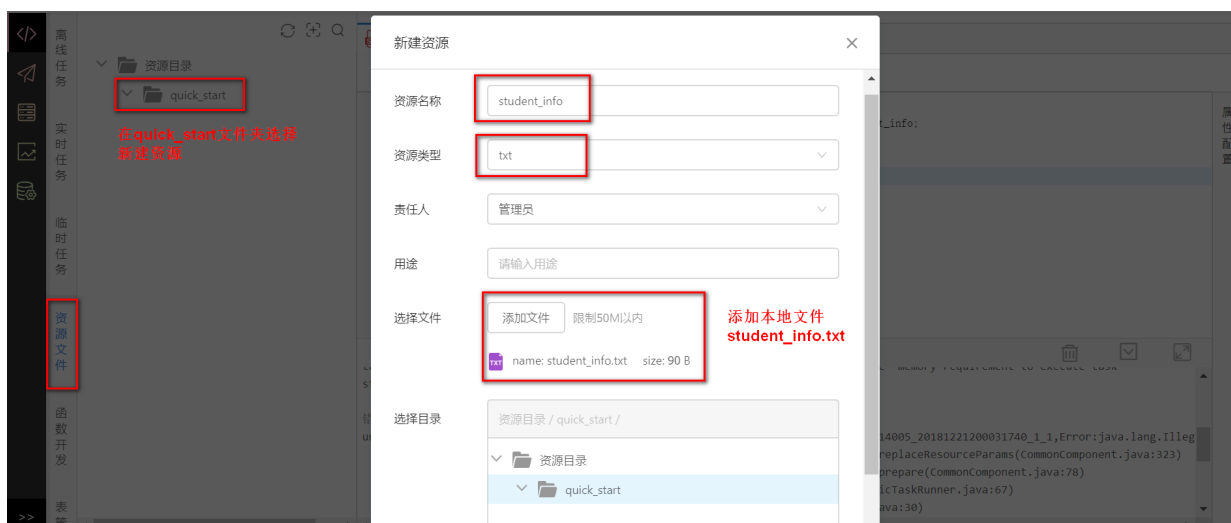
在开发中心下的【资源文件】页面中，右击左侧的【资源目录】，选择【新建目录】，输入目录名称：`quick_start`，点击确定完成。

注意：在开发环境删除发布过任务或资源的目录之后，需要用户手动在生产环境删除对应的目录，以保持开发生产目录结构的一致



(2) 新建资源

右键点击上步创建的 `quick_start` 目录，选择【新建资源】，输入资源名称为 `student_info`，选择资源类型为 `txt`，点击【添加文件】按钮，选择保存在本地的 `student_info.txt` 文件，点击确定。



7 新建表

(1) 新建 workflow

在开发中心下的【离线任务】页面中，点击【+】按钮，选择【新建 workflow】，输入 workflow 名称为 `quick_start`，workflow 模板选择【空白模板】。



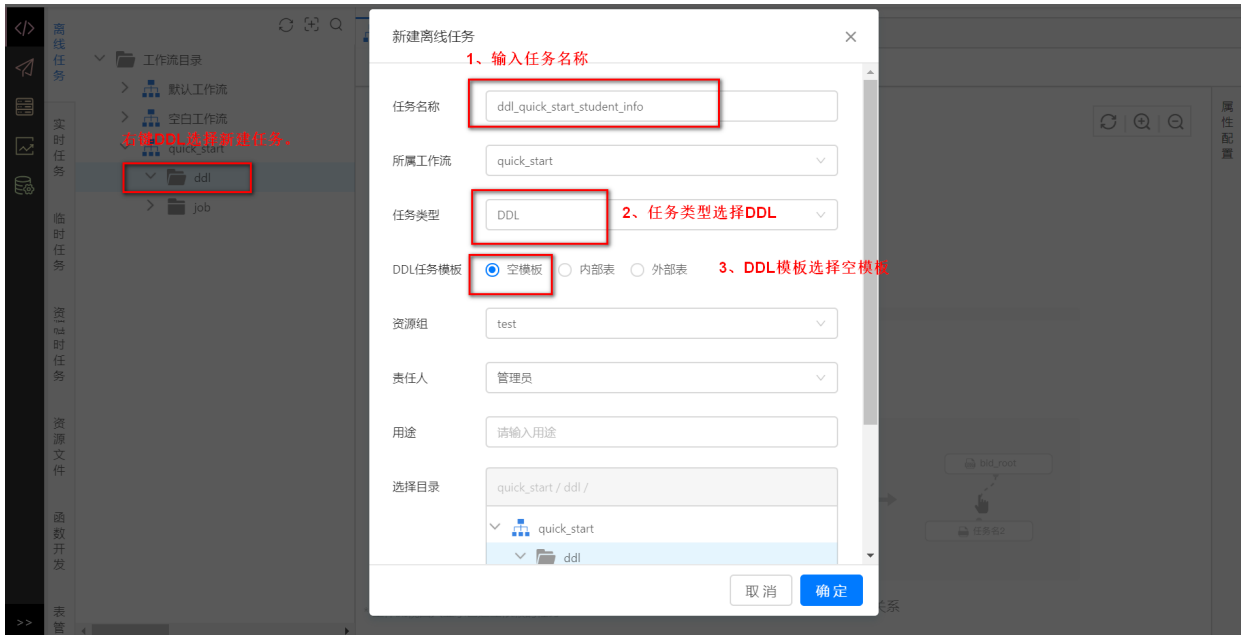
(2) 创建ddl与job目录

右击 `quick_start` 选择【新建目录】，分别创建**ddl**和**job**两个目录。其中，**ddl**目录用于存放创建表的DDL类型任务，**job**目录用于存放其他类型的任务。



(3) 创建DDL任务

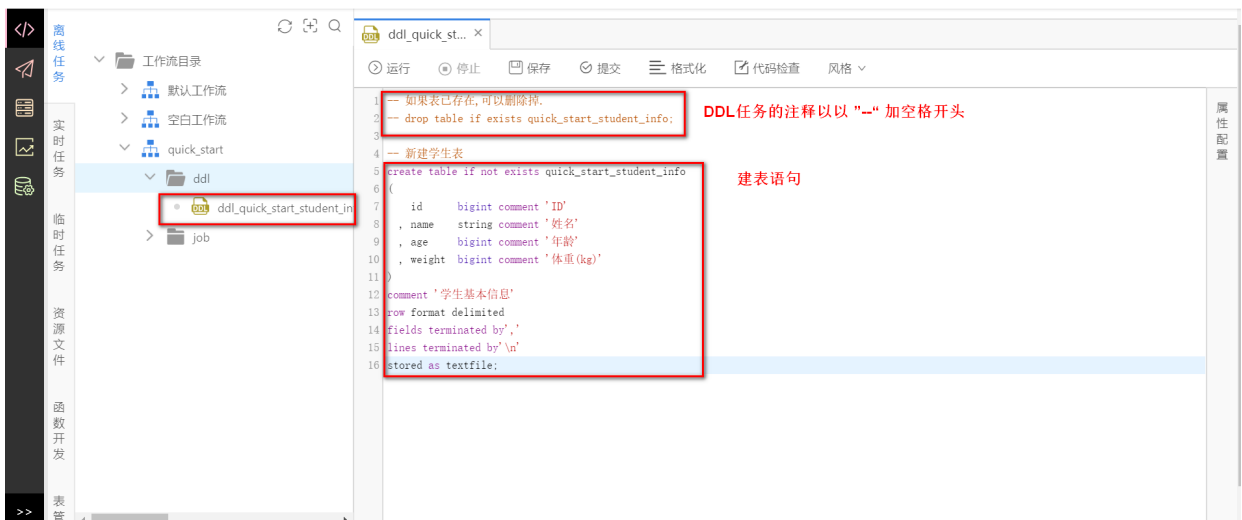
右击**ddl**目录，选择【新建离线任务】，输入任务名称：`ddl_quick_start_student_info`，任务类型：**DDL**，DDL任务模板：**空模板**。



在 `ddl_quick_start_student_info` 任务中，输入以下建表语句。其中代码注释必须要以 "--" 加空格开头，且为单独一行。

```
-- 如果表已存在,可以删除掉。
-- drop table if exists quick_start_student_info;

-- 新建学生表
create table if not exists quick_start_student_info
(
    id      bigint comment 'ID'
  , name   string comment '姓名'
  , age    bigint comment '年龄'
  , weight bigint comment '体重(kg)'
)
comment '学生基本信息'
row format delimited
fields terminated by ','
lines terminated by '\n'
stored as textfile;
```



点击 **运行** 按钮，DDL任务将会在Hive库中新建表 `quick_start_student_info`。运行日志如下图所示，如果显示 **任务运行成功(Finished)** 则表示此DDL任务运行成功。



8 数据导入

把 `student_info.txt` 文件资源中的数据导入到Hive表 `quick_start_student_info` 中。

(1) 创建导入数据任务

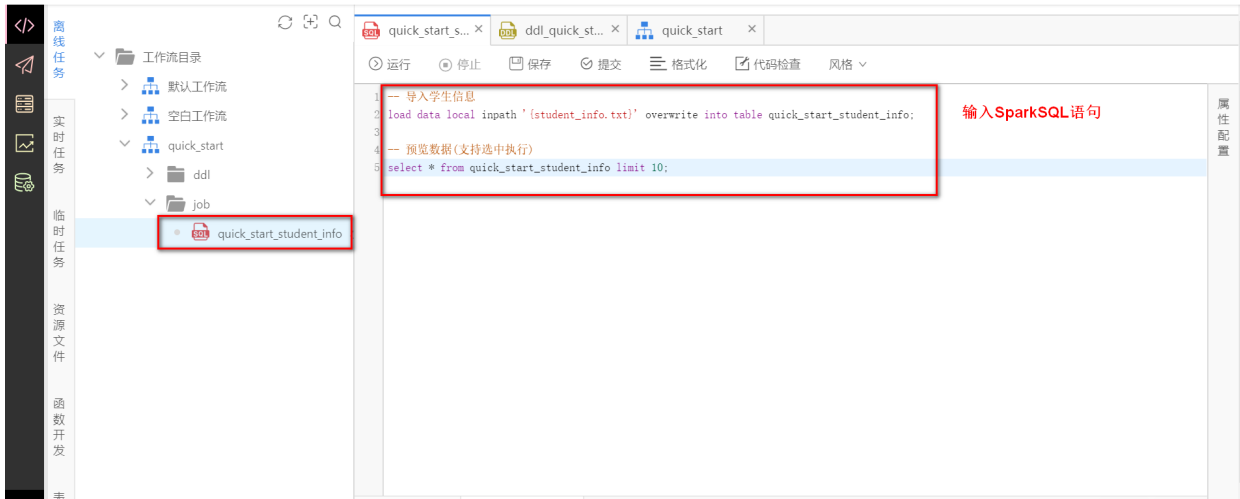
在开发中心的 **离线任务** 里，右击 `quick_start` 目录下的 **job** 目录，选择 **【新建离线任务】**，任务名为 `quick_start_student_info`，任务类型为 **SparkSQL**，点击确定完成。



任务创建成功后，输入如下导入数据语句：

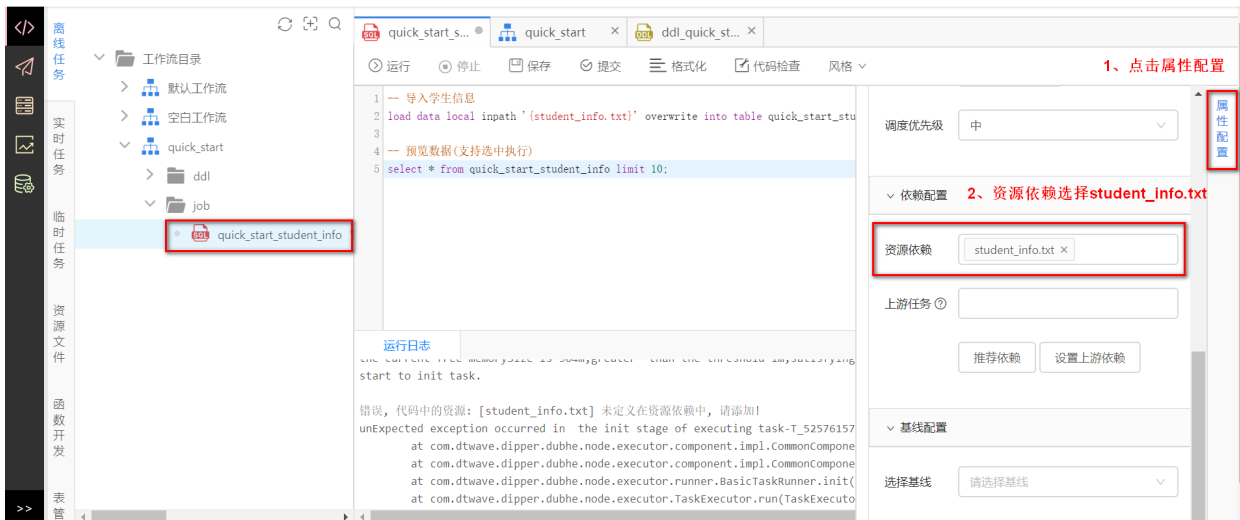
```
-- 导入学生信息
load data local inpath '{student_info.txt}' overwrite into table
quick_start_student_info;

-- 预览数据(支持选中执行)
select * from quick_start_student_info limit 10;
```

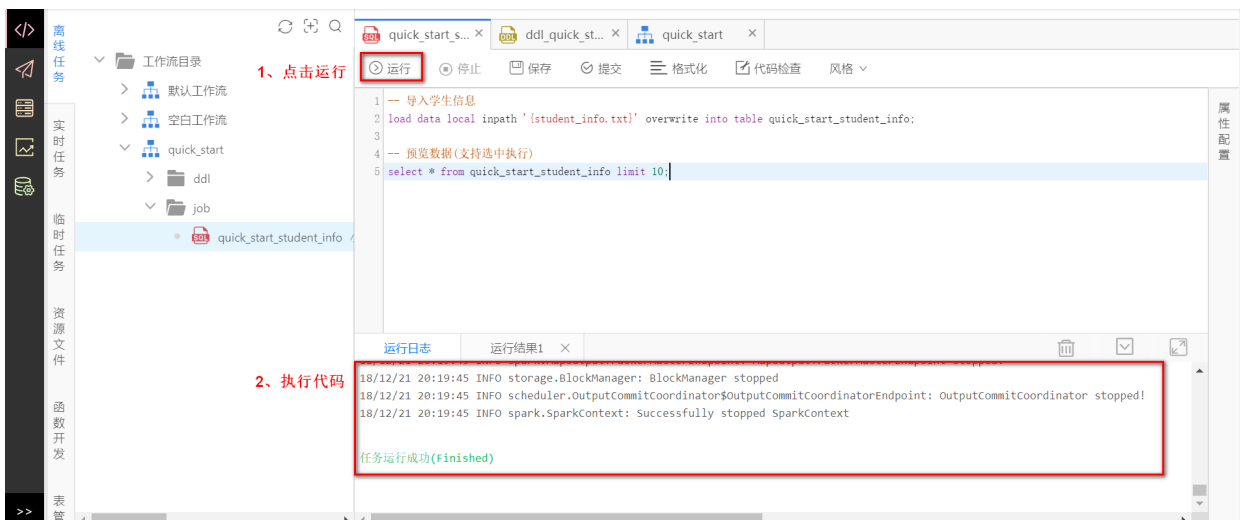
(2) 配置任务属性

点击任务右侧的【属性配置】按钮，在 依赖配置 中的 资源依赖 输入框中，选择 `student_info.txt` 资源。

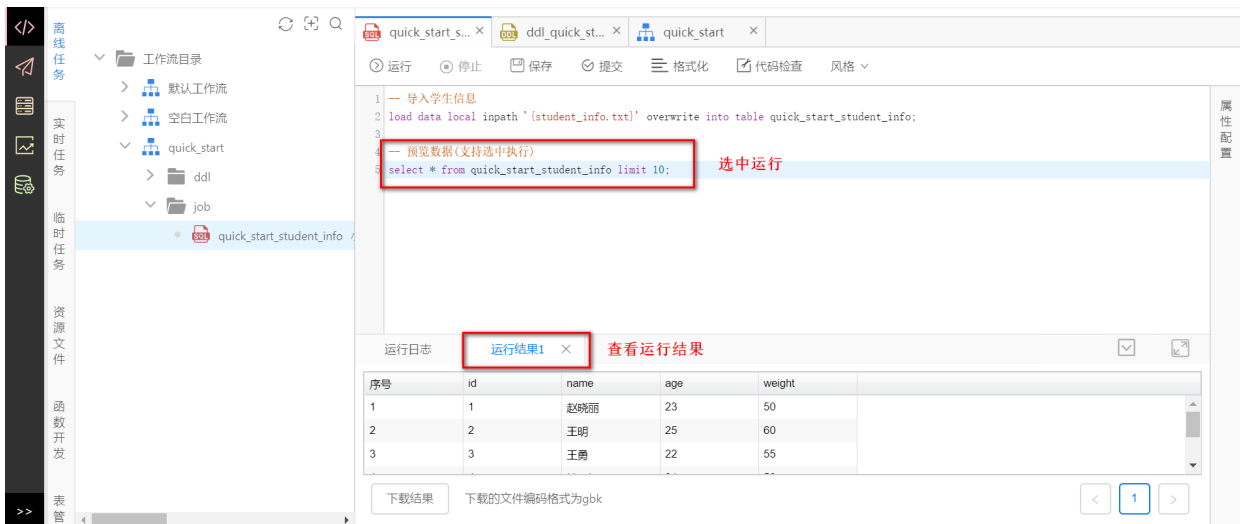


(3) 运行任务

点击【运行】按钮，运行SparkSQL任务。运行日志如图所示，如果日志中出现 任务运行成功 (Finished)，则表示导入数据成功。



导入成功后，也可只选中上述的select语句（选中整行）后，点击运行来预览数据。



导入成功后，可在表管理中查看表 `quick_start_student_info` 中的预览数据。



注意：运行时可能会出现如下错误，这是由于没有该库权限导致的。

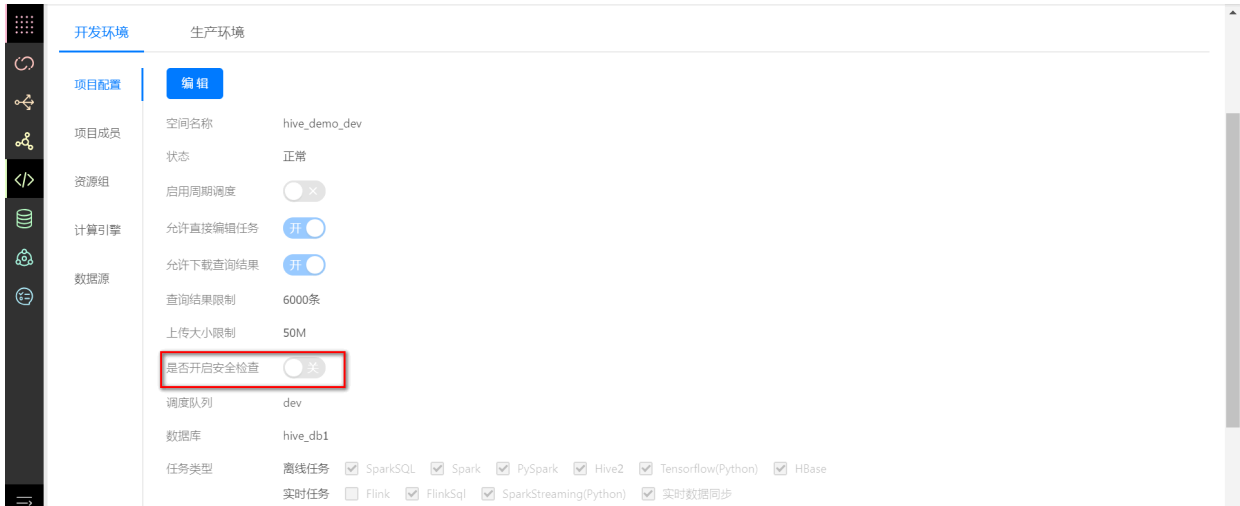
```

错误! 鉴权不通过! 请检查权限! 不通过原因:
command:load data local inpath '/data/dubhe/codes/T_5348218791107092_20190116150942716_2_1/resources/student_info.txt' overwrite into table quick_start_student_info
failedReasonList:["【EntityPermission】鉴权结果: 不含有库名: hive_db1, 表: quick_start_student_info 的 INSERT 权限"]
错误! 鉴权不通过! 请检查权限! 不通过原因:
command:select * from quick_start_student_info limit 10
failedReasonList:["【fieldPermission】鉴权结果: 不含有库名: hive_db1, 表: quick_start_student_info , 字段[id, name, age, weight] 的 SELECT 权限"]
unexpected exception occurred in the init stage of executing task-T_5348218791107092_20190116150942716_2_1,Error:com.dtwave.common.exception.BizException: 鉴权失败原因为:
[{"failedReasonList":["【EntityPermission】鉴权结果: 不含有库名: hive_db1, 表: quick_start_student_info 的 INSERT 权限"],"command":"load data local inpath '/data/dubhe/codes/T_5348218791107092_20190116150942716_2_1/resources/student_info.txt' overwrite into table quick_start_student_info"}]
at com.dtwave.dipper.dubhe.node.executor.component.impl.CommonComponent.authorityCheck(CommonComponent.java:617)
at com.dtwave.dipper.dubhe.node.executor.component.impl.CommonComponent.authorityCheck(CommonComponent.java:519)
at com.dtwave.dipper.dubhe.node.executor.component.impl.CommonComponent.prepare(CommonComponent.java:93)
at com.dtwave.dipper.dubhe.node.executor.runner.BasicTaskRunner.init(BasicTaskRunner.java:70)
at com.dtwave.dipper.dubhe.node.executor.TaskExecutor.run(TaskExecutor.java:30)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)

任务运行失败(Failed)

```

解决方法1：在项目配置中关闭安全检查。

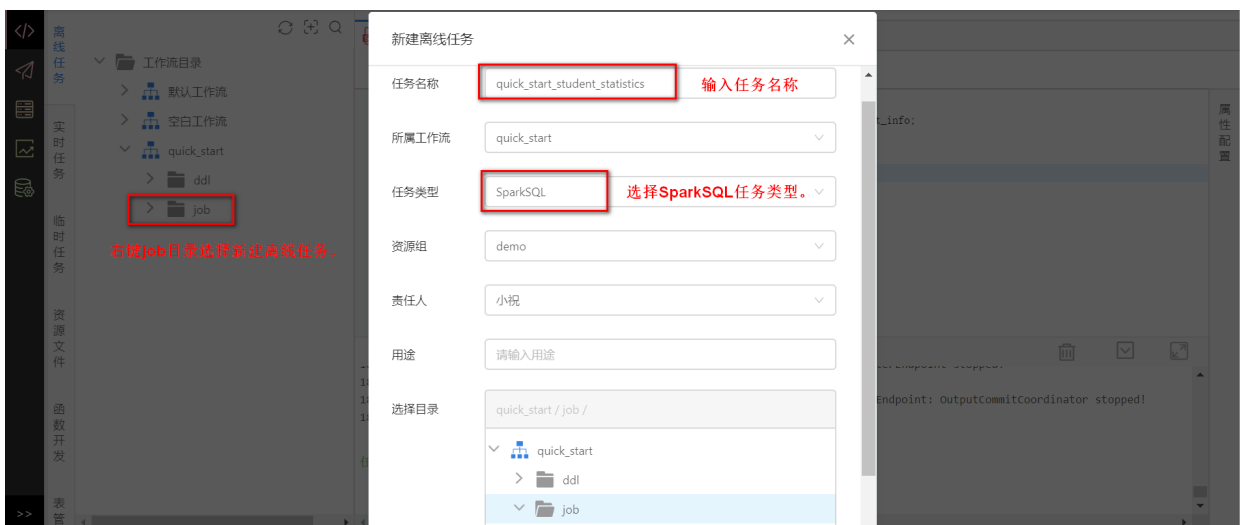


解决方法2：在【数据管理-数据权限】中申请该库权限，详情请查看【[用户指南-数据管理-数据权限](#)】的权限申请部分。

9 数据加工

(1) 创建任务

在开发中心的【离线任务】里，右击 `quick_start` 目录下的 `job` 目录，选择【新建离线任务】。任务名称 填写 `quick_start_student_statistics`，任务类型 选择 **SparkSQL**，点击确定完成。



在 `quick_start_student_statistics` 任务中，输入如下的数据加工语句：

```
-- 1. 查询学生基本信息
select
    id
    , name
    , age
    , weight
from quick_start_student_info
order by age;

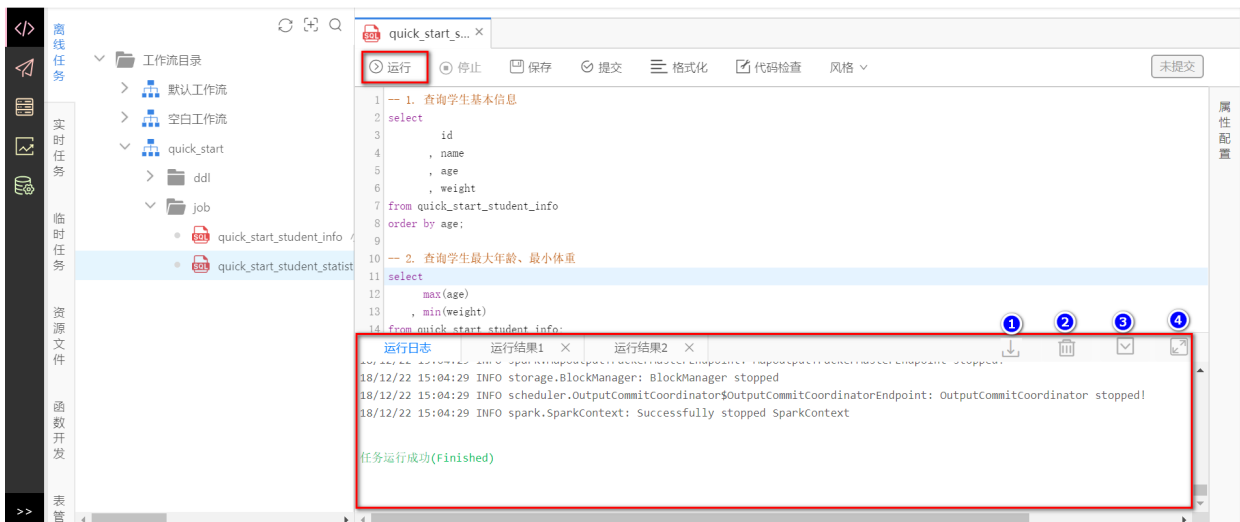
-- 2. 查询学生最大年龄、最小体重
select
```

```
max(age)
, min(weight)
from quick_start_student_info;
```

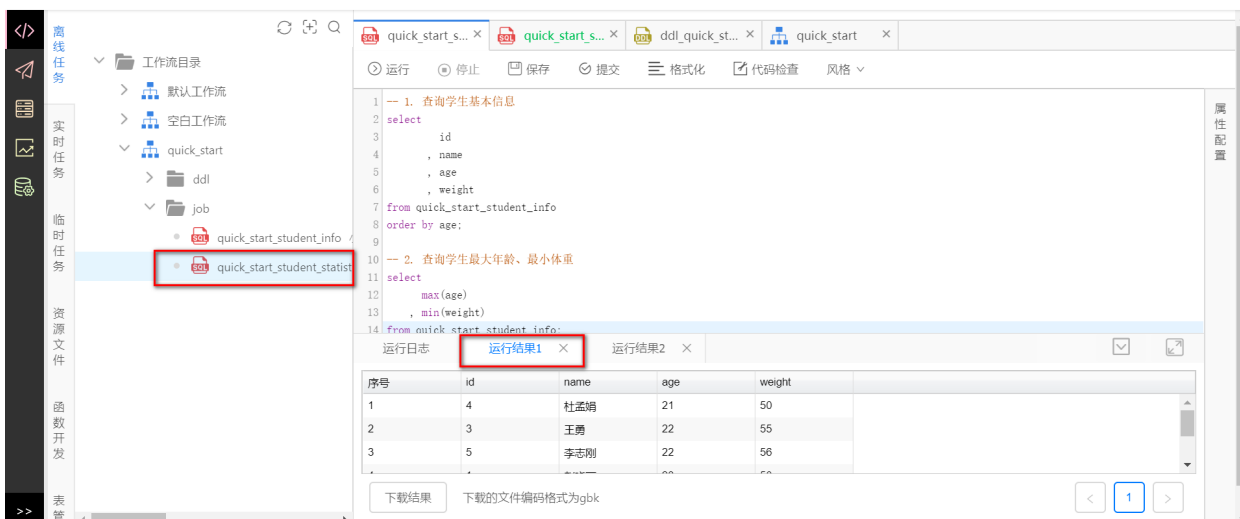
(2) 运行任务

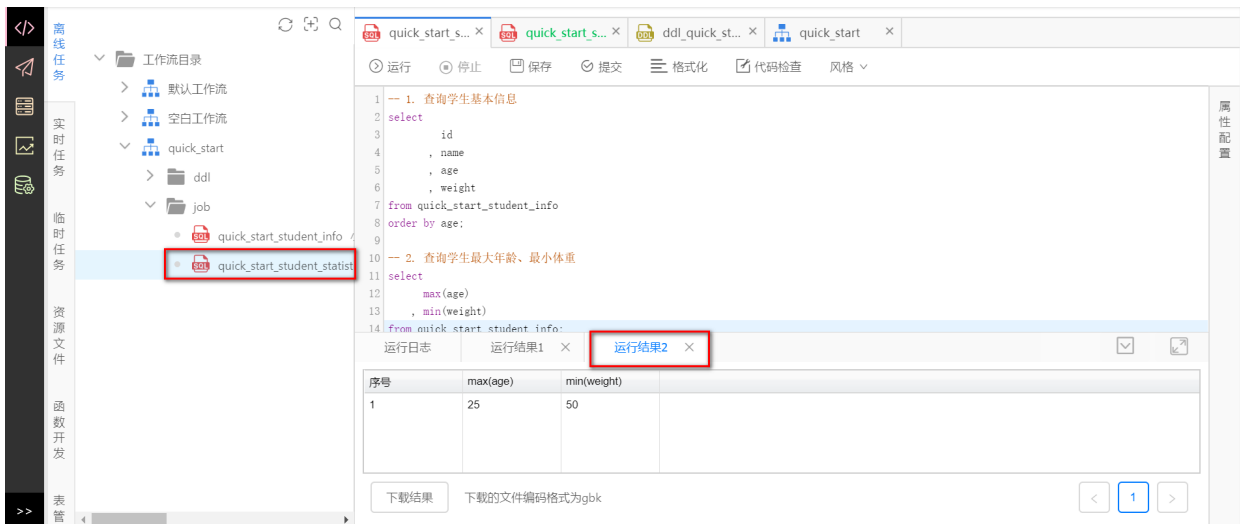
点击 **运行** 按钮运行任务，运行日志如图所示，如果出现 **任务运行成功(Finished)**，则表示数据加工任务运行成功。

每条select语句会产生一个运行结果，因此再未选中运行的情况下（相当于全选运行），运行后会产生两个运行结果。



点击 **运行结果1** 和 **运行结果2** 分别查看数据加工任务的运行结果。





用户指南

此部分将详细介绍开发门户各个模块及使用流程。

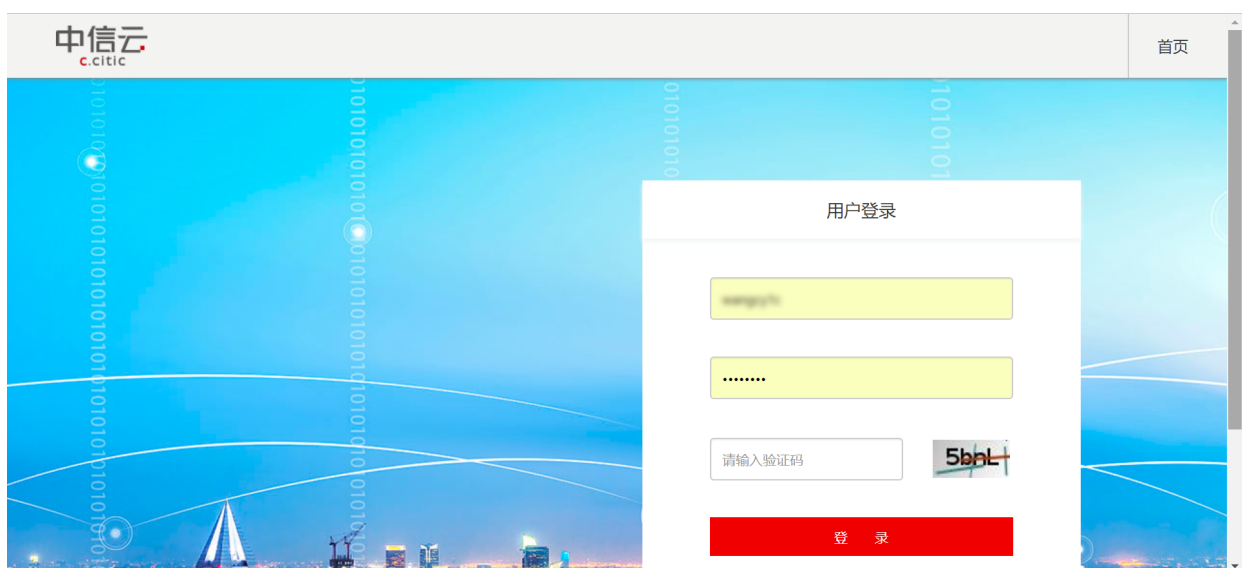
1 注册登录

1.1 注册

用户在中信云购买开发门户成功后，将在开发门户上自动注册以该用户为owner的租户账号，该账号具有平台管理员权限。

1.2 登录

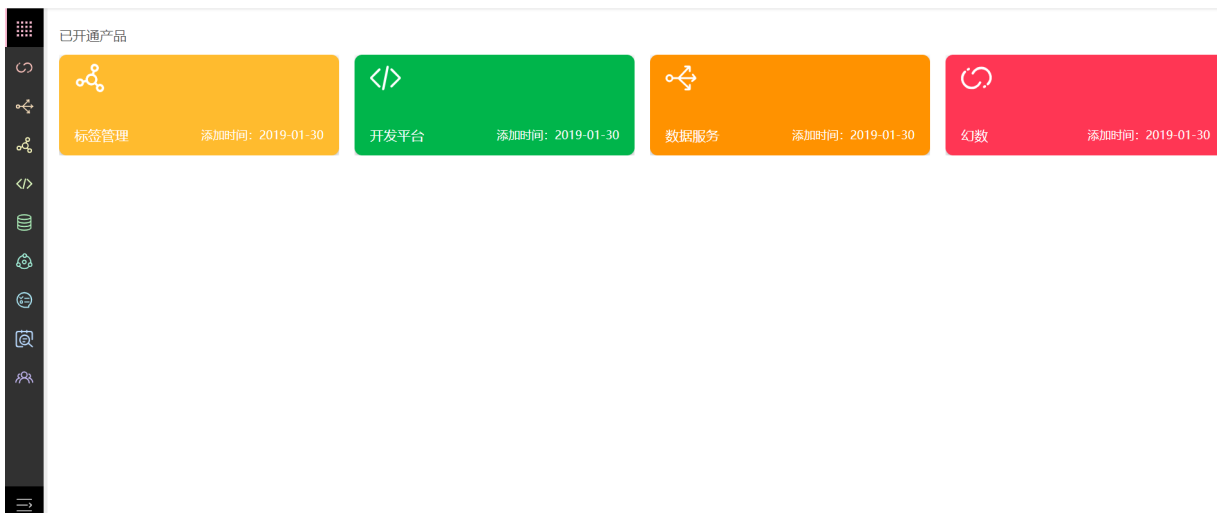
- (1) 通过浏览器访问开发门户
- (2) 输入用户名、密码、验证码登录开发门户。



1.3 进入控制台

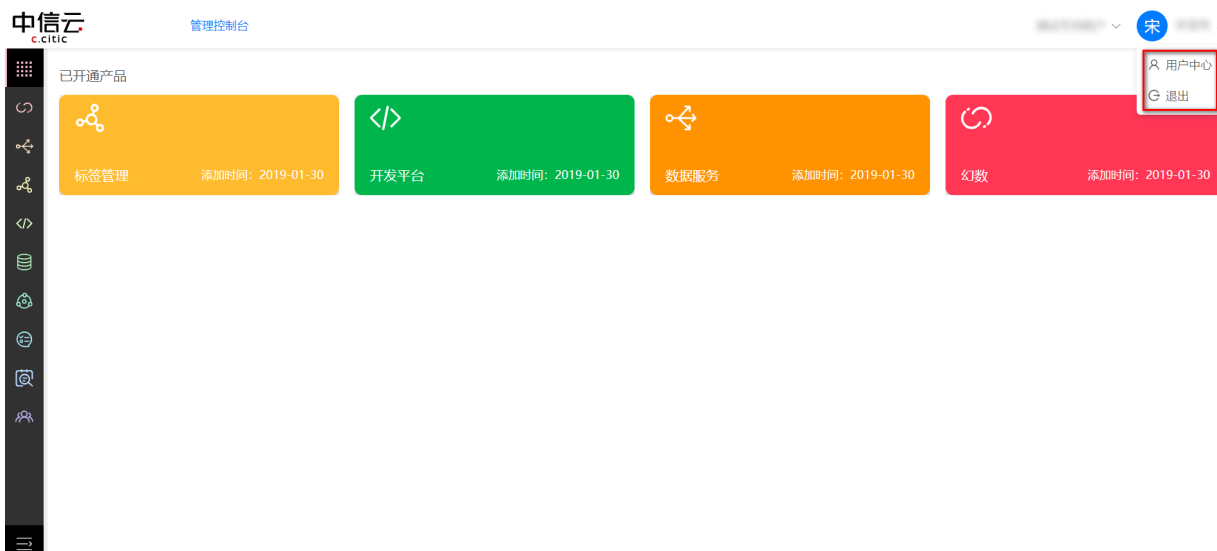
登录成功后进入控制台页面。若购买的是高级版，在已开通产品中显示：标签管理、开发中心、数据服务和幻数。

若购买的是基础版，在已开通产品中显示：开发中心、标签管理；在未开通产品中显示：数据服务、幻数。点击未开通服务右上角的【+】号，可跳转到购买页面。



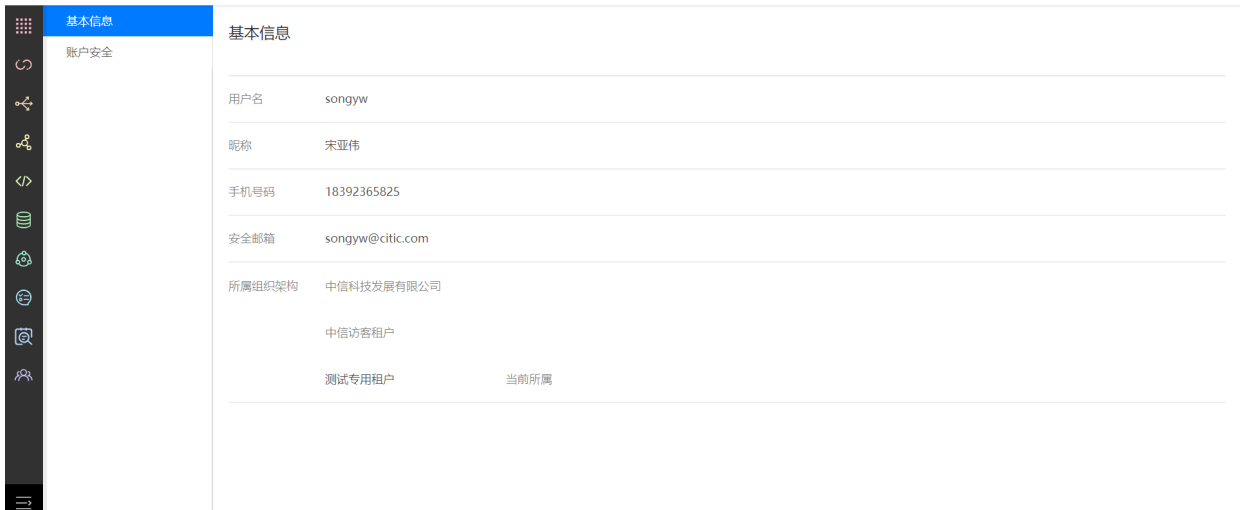
2 用户中心

点击右上角昵称-用户中心，进入用户中心页面。



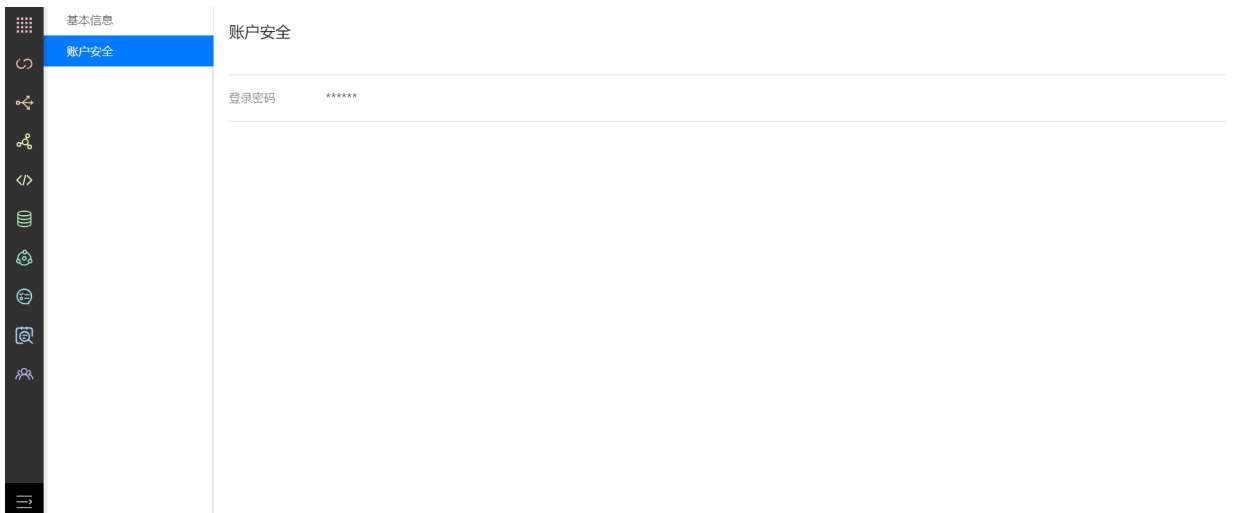
2.1 基本信息

点击【基本信息】，进入基本信息页面。



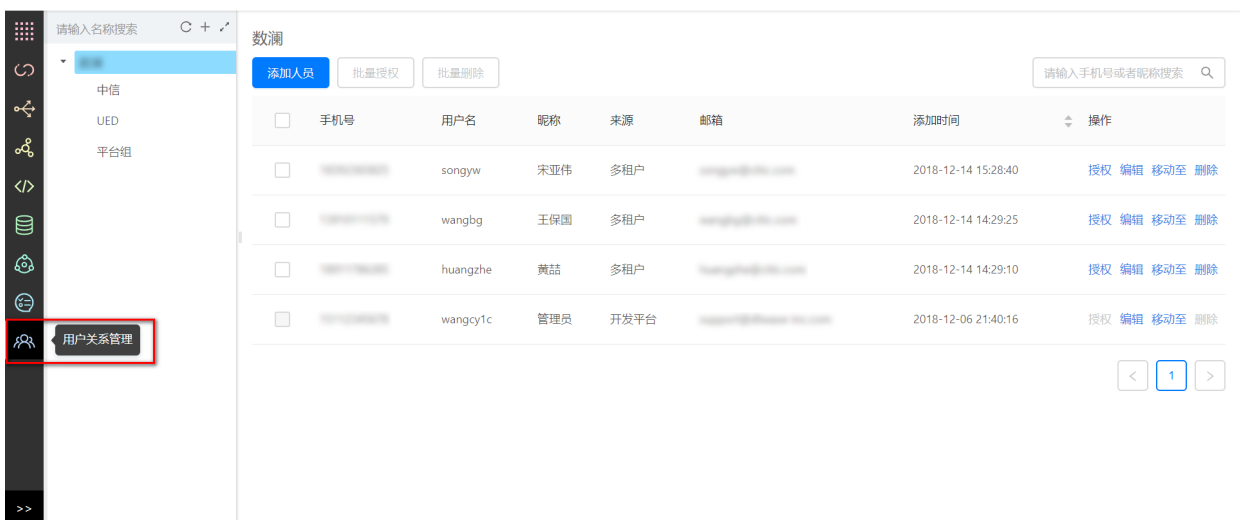
2.2 账户安全

点击【账户安全】，进入账户安全页面。

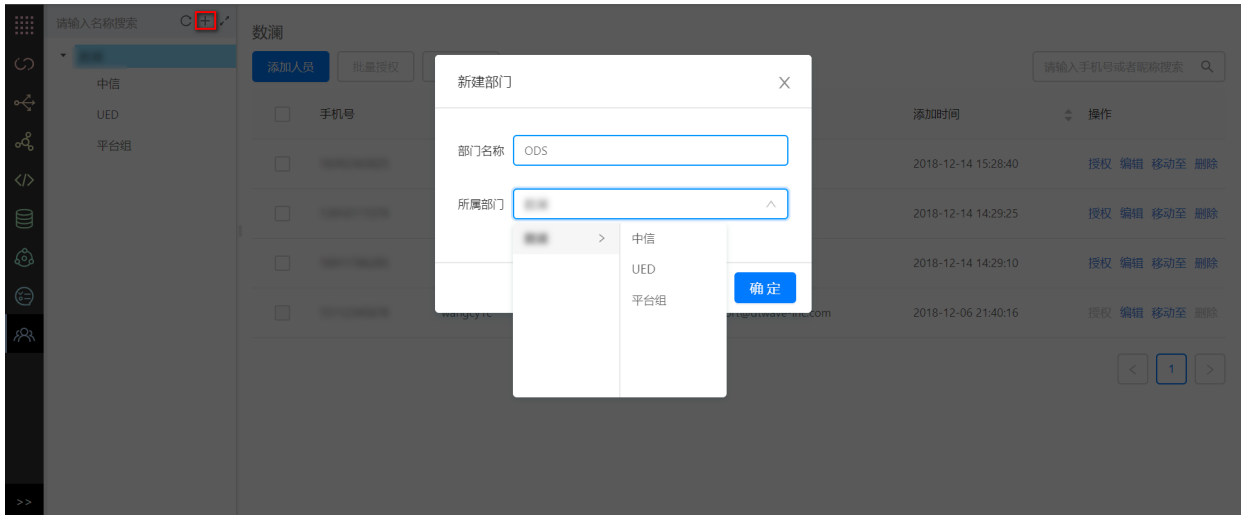


3 用户关系管理

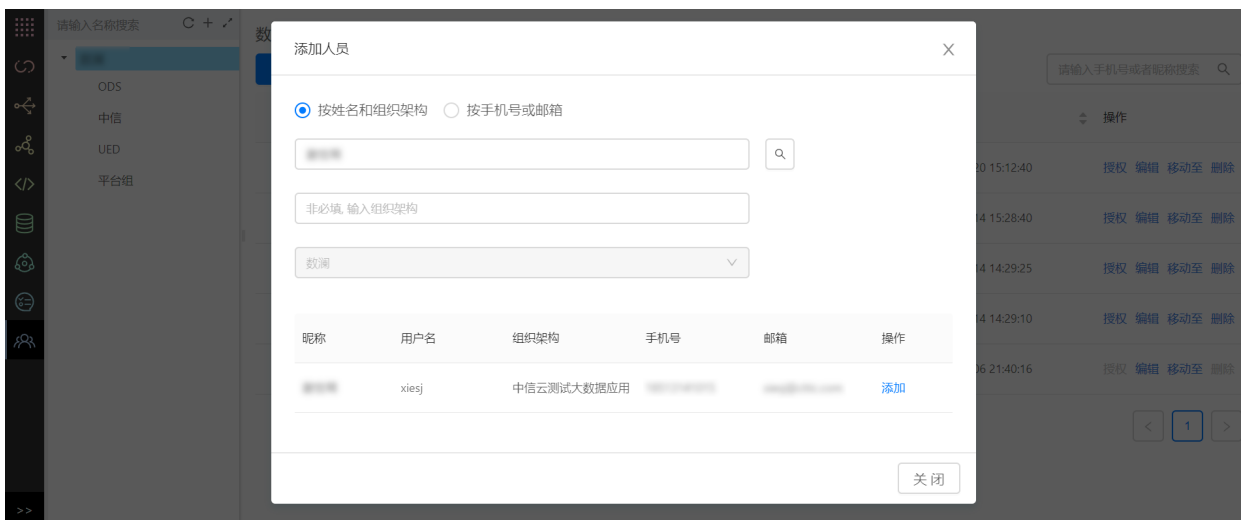
点击左侧【用户关系管理】图标，进入用户关系管理。



(1) 新建部门：点击【+】图标，在弹窗中输入部门名称，选择所属部门后，点击确定完成新建。



(2) 添加人员：先选择部门，再点击【添加人员】，可通过按姓名和组织架构/按手机号或邮箱搜索到用户后，点击【添加】完成添加。



(3) 人员授权：点击【授权】，可将已添加的人员授权。权限详情请查看【[权限说明](#)】



4 资源管理

4.1 资源组管理

点击左侧【资源组】图标，进入资源组管理。

本平台已准备好一个资源组：默认资源组



(1) 新建资源组：点击【新建资源组】，在弹窗中输入名称和描述，点击确定完成新建名为**资源组_demo**的资源组。



(2) 添加服务器：点击资源组名称，为该资源组添加服务器，点击【添加服务器】，在弹窗中输入服务器名称、IP地址、并发数，点击确定完成添加一个名为**node1**的服务器。



(3) 配置服务：点击配置，在弹窗中开启所需服务，并输入端口号，点击【测试连接】，连接通过后，点击确定完成配置服务。



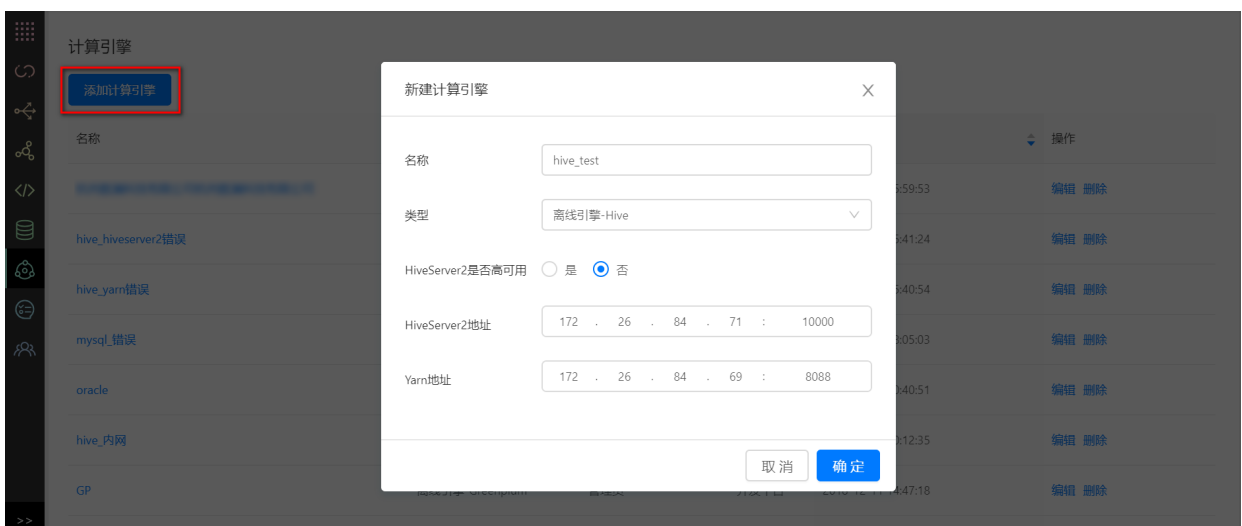
4.2 计算引擎管理

点击左侧【计算引擎】，进入计算引擎管理。

本平台已准备好一个计算引擎：默认计算引擎_Hive



(1) 添加计算引擎：点击添加计算引擎，在弹窗中输入引擎名称，选择类型，填入配置信息，点击确定完成添加。下图为添加一个hive引擎的示例。



(2) 操作：可对已经添加的计算引擎进行操作，包括编辑和删除。

5 数据源管理

点击左侧【数据源管理】，进入数据源管理。

来源包括开发平台和多租户。注意：若将来源于多租户的数据源授权给其他项目，需要在多租户平台中对该数据源进行授权，否则将不起作用。



数据源管理

添加数据源

类型: 全部

请输入名称进行搜索

名称	创建人	类型	关联资源组	来源	创建时间	修改时间	操作
mysql_wzq_DTWAVE_TEST_dev_394153	迭代测试	MySQL	demo资源组	开发平台	2019-03-23 14:21:02	2019-03-23 14:21:02	编辑 授权 删除
mysql_wzq_DTWAVE_TEST_prd_378389	迭代测试	MySQL	demo资源组	开发平台	2019-03-23 14:21:02	2019-03-23 14:21:02	编辑 授权 删除
hive_bigdatadevportal_dev_535953	迭代测试	Hive	246	开发平台	2019-03-22 21:33:52	2019-03-22 21:33:52	编辑 授权 删除
hive_bigdatadevportal_prd_848859	迭代测试	Hive	246	开发平台	2019-03-22 21:33:52	2019-03-22 21:33:52	编辑 授权 删除
trainDemo_oracle	迭代测试	Oracle	trainDemo资源组	开发平台	2019-03-19 20:36:51	2019-03-19 20:36:51	编辑 授权 删除
trainDemo_GP_gp84_dev_145117	迭代测试	Greenplum	trainDemo资源组	开发平台	2019-03-19 20:12:05	2019-03-19 20:12:05	编辑 授权 删除
trainDemo_GP_gp84_prd_860364	迭代测试	Greenplum	trainDemo资源组	开发平台	2019-03-19 20:12:05	2019-03-19 20:12:05	编辑 授权 删除

5.1 MySQL数据源

(1) 添加MySQL数据源。

点击【添加数据源】，在弹窗中选择MySQL类型，填入连接信息，点击测试连接，连接通过后点击确定完成添加。



数据源管理

添加数据源

名称

hive_default_hive_demo_dev

hive_default_hive_demo_prd

新建数据源

名称: mysql_demo

资源组: 资源组_demo X

类型: MySQL

地址: 47.92.38.137:8000

数据库: demo

用户名: admin

密码:

描述:

修改时间

操作

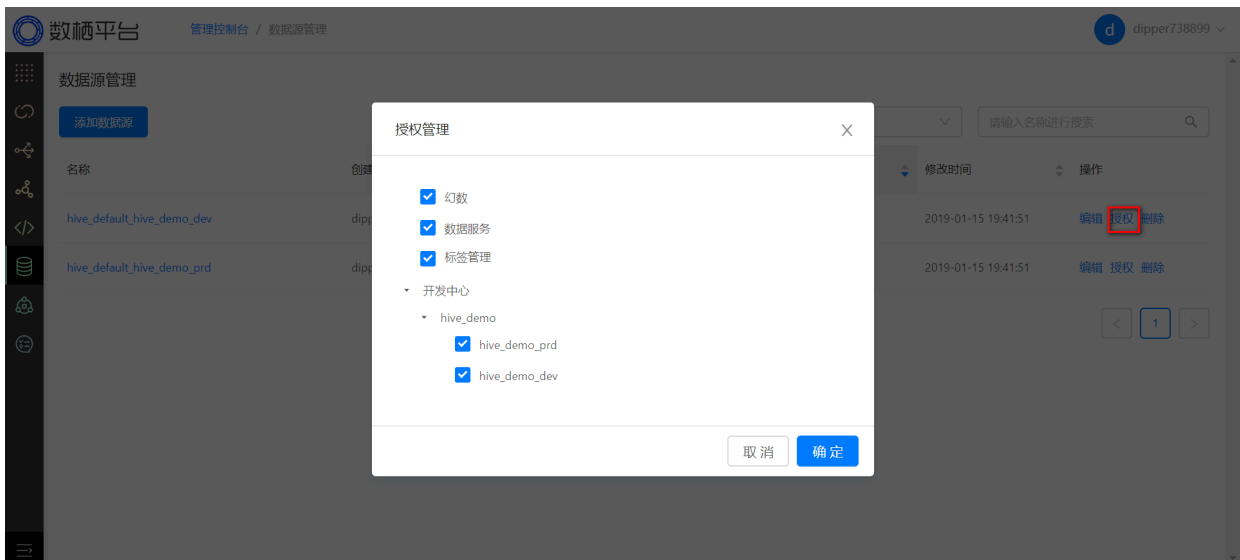
2019-01-15 19:41:51 编辑 授权 删除

2019-01-15 19:41:51 编辑 授权 删除

< 1 >

(2) 授权

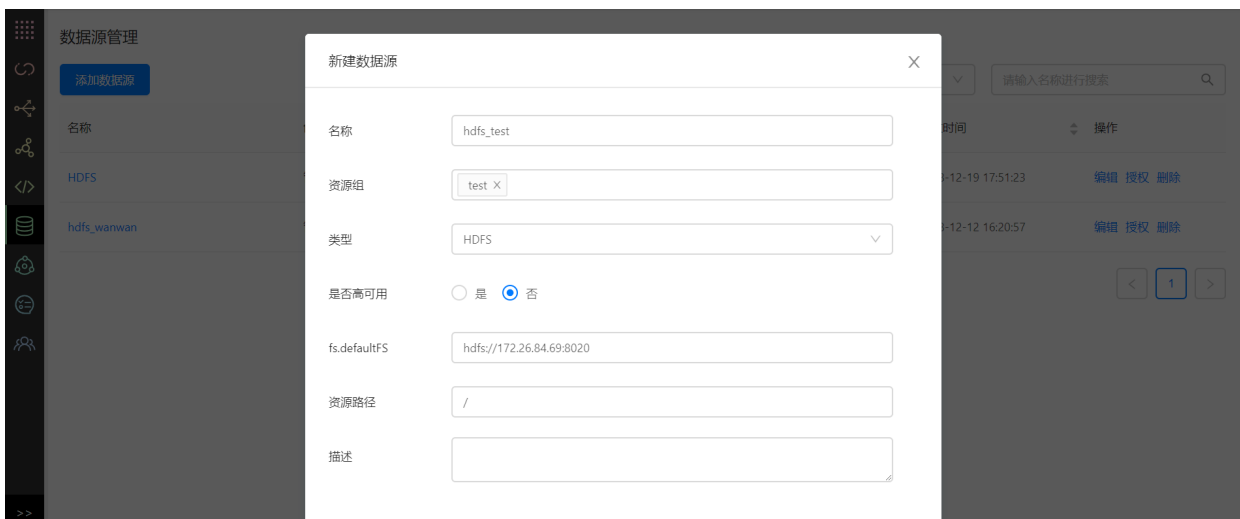
可对已经添加的数据源进行操作，包括编辑、删除、或授权。点击授权可将数据源授权到指定产品、项目或环境。



5.2 HDFS数据源

(2) 添加HDFS数据源。

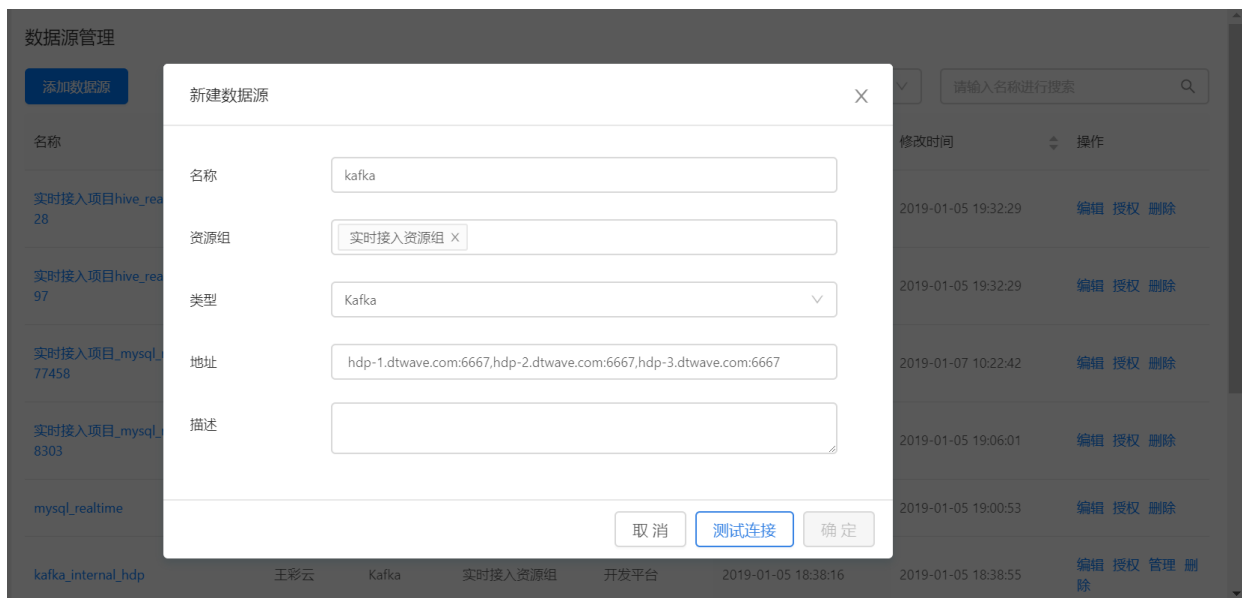
点击【添加数据源】，在弹窗中选择HDFS类型，输入地址、路径等信息，点击【测试连接】，连接通过后点击确定完成添加。



(2) 可对已经添加的数据源进行操作，包括编辑、删除、或授权。

5.3 Kafka数据源

(1) 添加Kafka数据源



(2) 可对已经添加的Kafka数据源进行操作，包括编辑、删除、授权或管理。

点击【管理】，进入该Kafka的管理页面。

数据源管理

添加数据源

类型: 全部

请输入名称进行搜索

名称	创建人	类型	关联资源组	来源	创建时间	修改时间	操作
kafka	王彩云	Kafka	实时接入资源组	开发平台	2019-01-07 14:38:09	2019-01-07 14:38:09	编辑 授权 管理 删除
实时接入项目hive_realtime_dev_853028	王彩云	Hive	实时接入资源组	开发平台	2019-01-05 19:32:29	2019-01-05 19:32:29	编辑 授权 删除
实时接入项目hive_realtime_prd_552497	王彩云	Hive	实时接入资源组	开发平台	2019-01-05 19:32:29	2019-01-05 19:32:29	编辑 授权 删除
实时接入项目_mysql_realtime_dev_477458	王彩云	MySQL	实时接入资源组	开发平台	2019-01-05 19:06:01	2019-01-07 10:22:42	编辑 授权 删除
实时接入项目_mysql_realtime_prd_978303	王彩云	MySQL	实时接入资源组	开发平台	2019-01-05 19:06:01	2019-01-05 19:06:01	编辑 授权 删除
mysql_realtime	王彩云	MySQL	实时接入资源组	开发平台	2019-01-05 19:00:53	2019-01-05 19:00:53	编辑 授权 删除
kafka_internal_hdp	王彩云	Kafka	实时接入资源组	开发平台	2019-01-05 18:38:16	2019-01-05 18:38:55	编辑 授权 管理 删除
oor_sync_1111_1_dev_145151	王彩云	Hive	开发环境通用_勿改	开发平台	2019-01-04 21:07:03	2019-01-04 21:07:03	编辑 授权 删除

进入Kafka管理，可看到该kafka的topic列表和broker列表。

Topic

[新增Topic](#) [批量操作](#)

<input type="checkbox"/>	名称	副本数	分区数	日志保留时长 (天)	描述	操作
<input type="checkbox"/>	bas_broker	3	6	7		编辑 删除
<input type="checkbox"/>	ambari_kafka_service_check	1	1	7		编辑 删除

共2条 [<](#) [1](#) [>](#) 10条/页 跳至 页

Broker

ID	Host	Port	Status
1001	hdp-3.dtwave.com	6667	● 正常
1002	hdp-1.dtwave.com	6667	● 正常
1003	hdp-2.dtwave.com	6667	● 正常

点击新增topic，可为该kafka新增一个topic。也可对已新增的topic进行编辑、删除等基础操作。

kafka

Topic

[新增Topic](#) [批量操作](#)

<input type="checkbox"/>	名称	副本数	分区数	日志保留时长 (天)	描述	操作
<input type="checkbox"/>	bas_broker	3	6	7		编辑 删除
<input type="checkbox"/>	ambari_kafka_service_check	1	1	7		编辑 删除

共2条 [<](#) [1](#) [>](#) 10条/页 跳至 页

Broker

ID	Host	Port	Status
1001	hdp-3.dtwave.com	6667	● 正常
1002	hdp-1.dtwave.com	6667	● 正常
1003	hdp-2.dtwave.com	6667	● 正常

新建 Topic

名称:

分区数目:

副本数:

描述:

[取消](#) [保存](#)

6 开发中心

6.1 项目管理

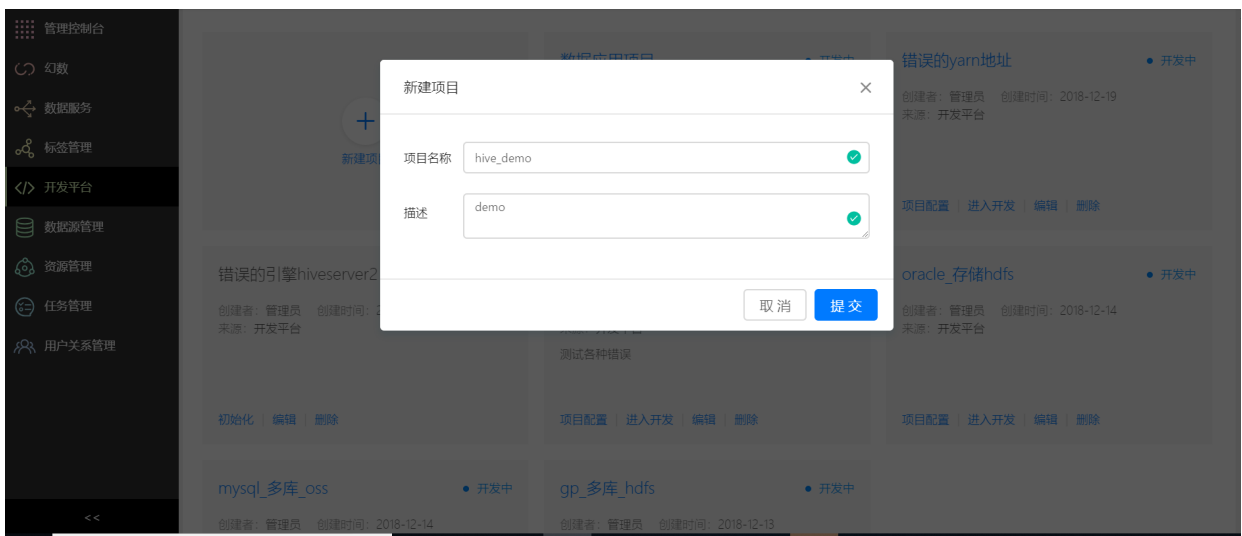
点击左侧【开发平台】图标，进入开发平台的项目列表。

项目来源包括开发平台和多租户，来源开发平台指手动在平台上创建项目，来源多租户指从多租户平台自动同步项目。



6.1.1 新建项目

点击新建项目，在弹窗中输入项目名称和描述，点击提交完成一个名为 `hive_demo` 的项目。



6.1.2 资源授权

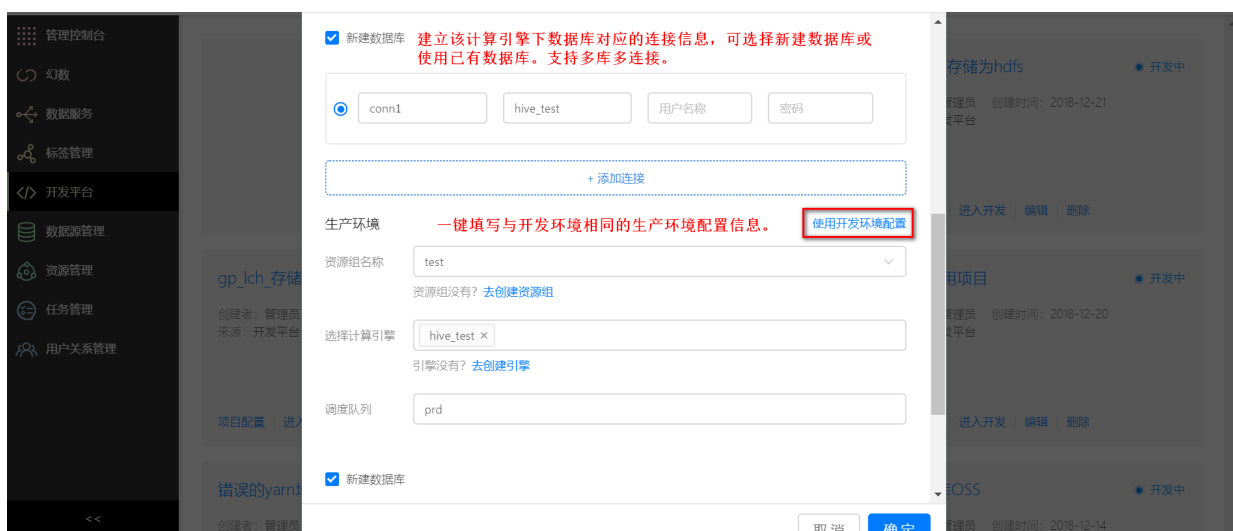
在数据源管理中，将已经建好的数据源授权给项目的环境，点击确定完成授权。



6.1.3 初始化项目

点击【初始化】，在弹窗中填写项目初始化信息。

每个项目默认包含两个环境，一个开发环境，一个生产环境。初始化项目时，需要分别配置好开发环境和生产环境中的资源组、计算引擎、调度队列和数据库。



6.1.4 项目配置

项目初始化完成后，点击【项目配置】进入项目配置，项目配置页面可分别编辑开发环境、生产环境的配置。

(1) 项目配置

进入项目配置后默认显示项目配置页面。

开发环境

生产环境

项目配置

编辑

项目成员

状态 正常

启用周期调度

资源组

启用安全提交机制

计算引擎

允许直接编辑任务

数据源

允许下载查询结果

连接信息

查询结果限制 6000条

上传大小限制 50M

是否开启安全检查

调度队列 dev

数据库 hive_test

任务类型 离线任务 Hive MapReduce SparkSQL Spark PySpark Hive2 Tensorflow(Python) HBase实时任务 Flink FlinkSql SparkStreaming(Python) 实时数据同步 Storm同步任务 数据同步建表任务 DDL其他任务 Shell Python Perl

描述 hive_demo开发空间

存储目的源 hdfs_test

存储路径 /

- 启用周期调度：开启后，项目中的任务会按一定的周期自动运行任务。目前只有生产环境支持启用周期调度。
- 启用安全提交机制：开启后，任务需要运行成功后才可提交。
- 允许直接编辑任务：开启后，可直接在此环境中编辑任务。只支持在开发环境直接编辑任务。
- 允许下载查询结果：开启后，可以下载任务运行过程中得到的查询结果。只支持在开发环境下载查询结果。
- 是否开启安全检查：开启后，项目成员需要项目管理员开通权限后才能访问指定的数据库、表和字段。平台会自动为项目成员中的开发、运维和管理员角色开通本项目数据库的所有权限。
- 查询结果限制：任务对数据库的查询结果条数限制。
- 上传大小限制：开发过程中对上传资源的文件大小限制。
- 调度队列：此环境所使用的计算引擎的调度队列名称。
- 数据库：此环境所使用的数据库名称。
- 任务类型：此环境可使用的任务类型。

(2) 项目成员

在【项目配置】中点击项目的【项目成员】进入项目成员配置页面，需要为开发环境和生产环境分别添加成员。点击【添加成员】按钮，选择用户名和角色，为该环境添加成员并授权。添加完成后，该成员会收到平台发送的项目邀请邮件。



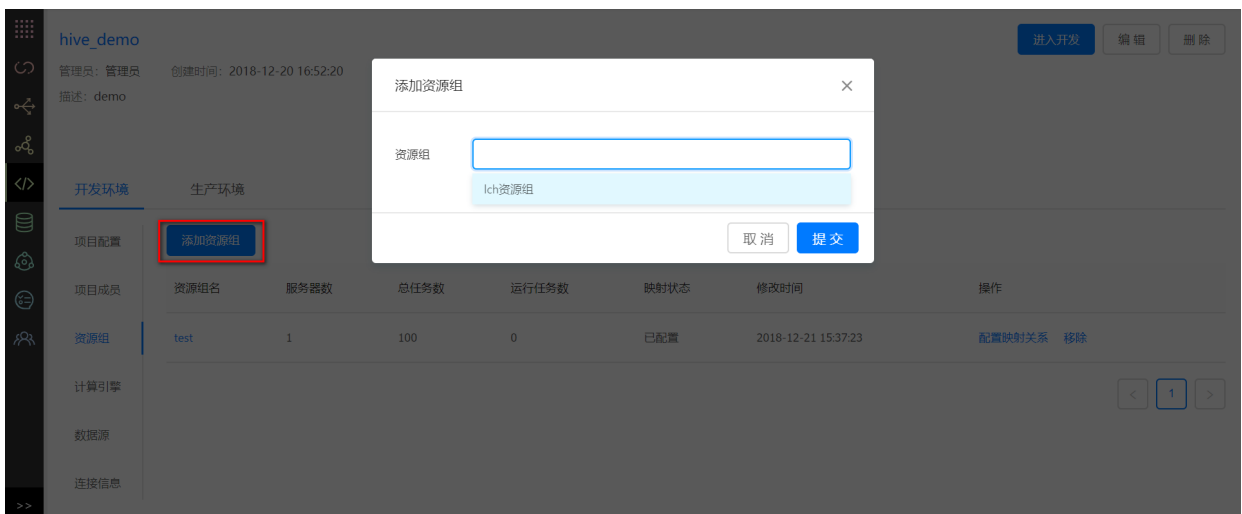
不同权限划分，请查看【[权限说明](#)】。

(3) 资源组

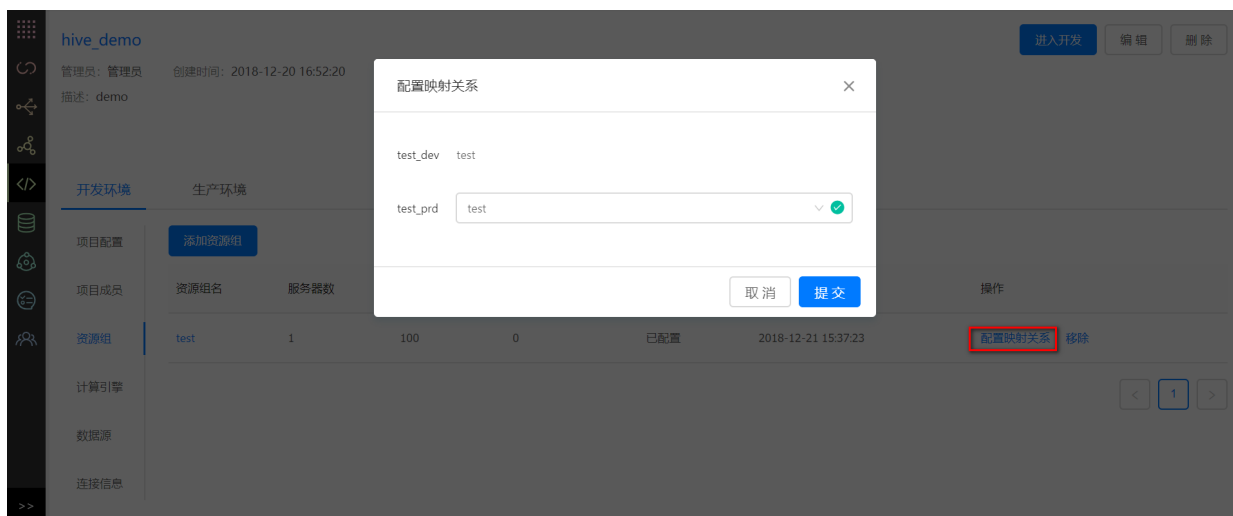
在【项目配置】中点击项目的【资源组】进入资源组配置页面。



点击添加资源组，可在弹窗中选择被授权的资源组，确定。

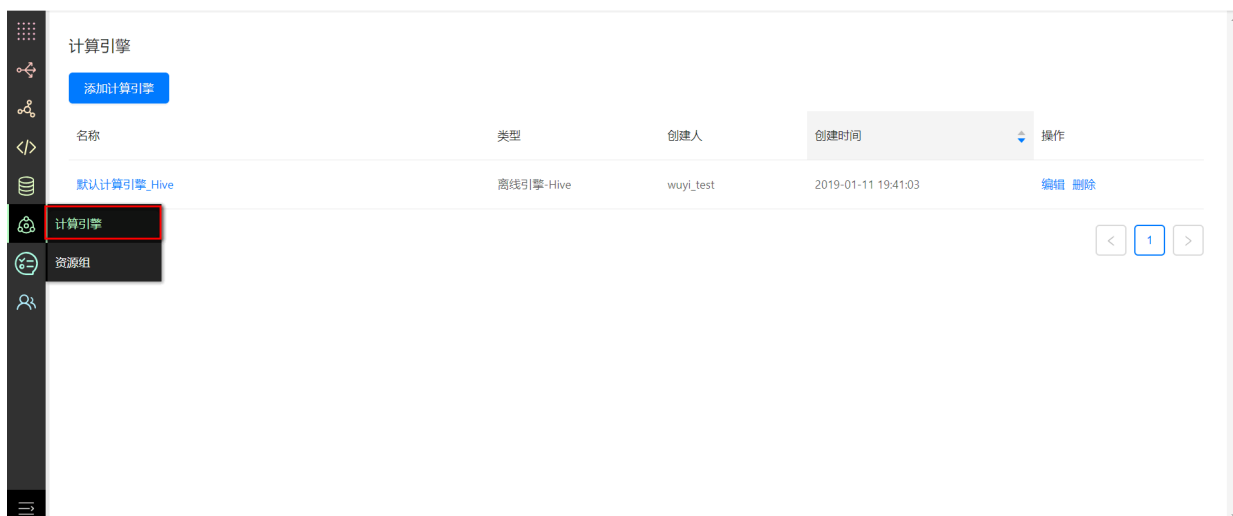


开发环境的任务只可使用开发环境的资源组，生产环境的任务只可使用生产环境的资源组。配置资源组时，需要分别配置开发环境和生产环境的资源组。**开发环境的资源组一般需要与生产环境的执行代理组进行一一映射。**当开发环境的任务开发完成需要发布到生产环境时，该任务所依赖的执行代理组必须在生产环境的执行代理组有一个映射关系，否则该任务将不能发布到生产环境。在资源组右侧点击**配置映射关系**，选择该资源组在生产环境对应的资源组。



(4) 计算引擎

点击左侧【计算引擎】，进入计算引擎管理。



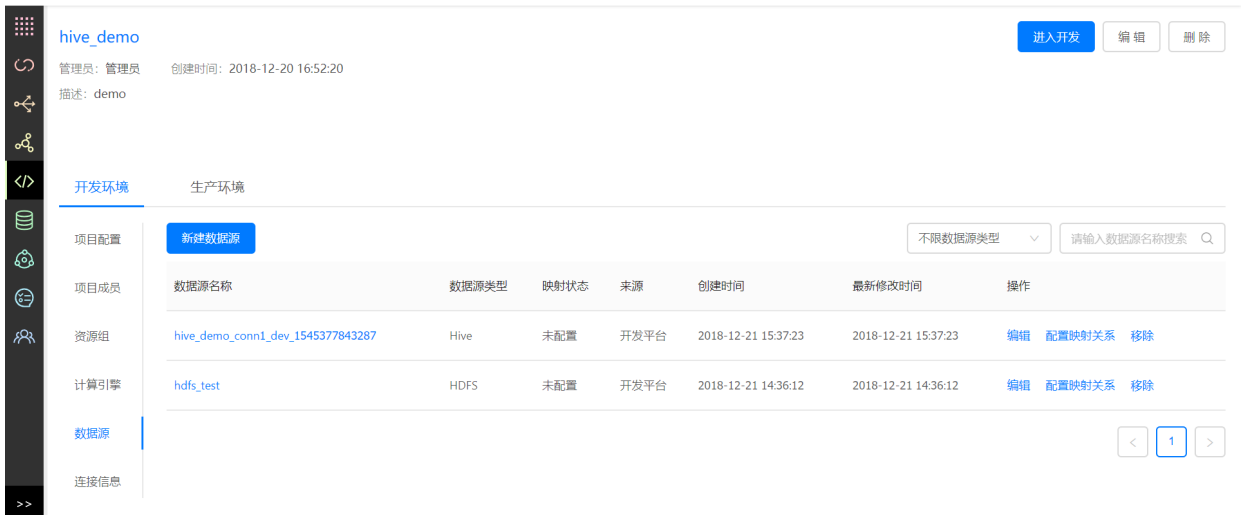
添加计算引擎：点击添加计算引擎，在弹窗中输入引擎名称，选择类型，填入配置信息，点击确定完成添加。下图为添加一个hive引擎的示例。



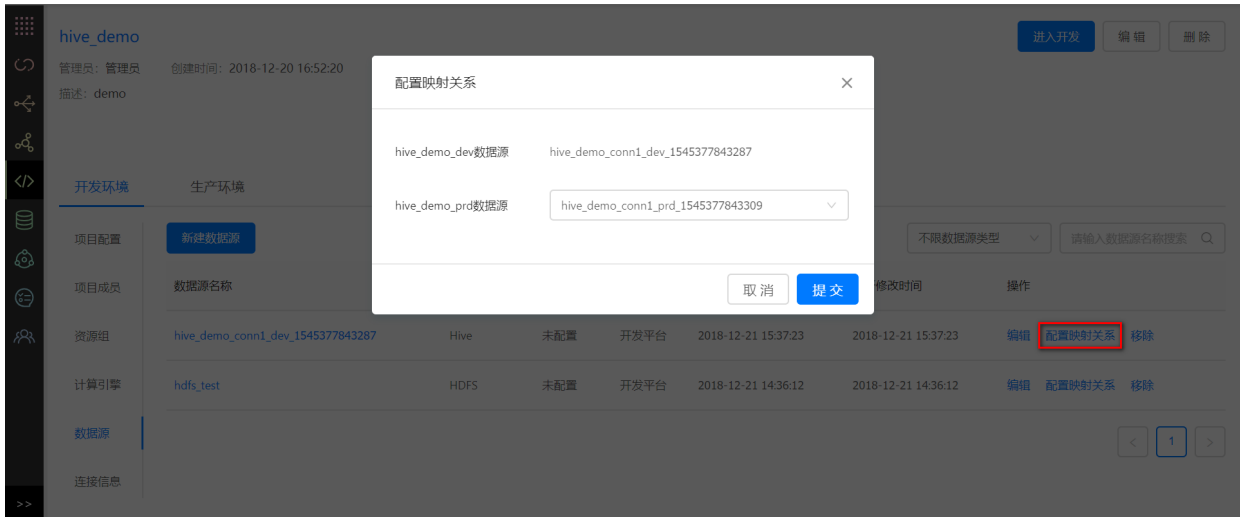
(5) 数据源

数据源一般用于**数据同步任务**，在【项目配置】中点击【数据源】进入数据源配置页面，开发环境的任务只可使用开发环境的数据源，生产环境的任务只可使用生产环境的数据源。配置数据源时，需要分别配置开发环境和生产环境的数据源。

项目将默认创建初始化项目时中连接信息所对应的数据源。



开发环境的数据源一般需要与生产环境的数据源进行**一一映射**。当开发环境的数据同步任务开发完成需要发布到生产环境时，该任务所依赖的数据源必须在生产环境的数据源有一个映射关系，否则该任务将不能发布到生产环境。在数据源右侧点击**配置映射关系**，选择该数据源在生产环境对应的数据源，点击提交完成。



(6) 连接信息

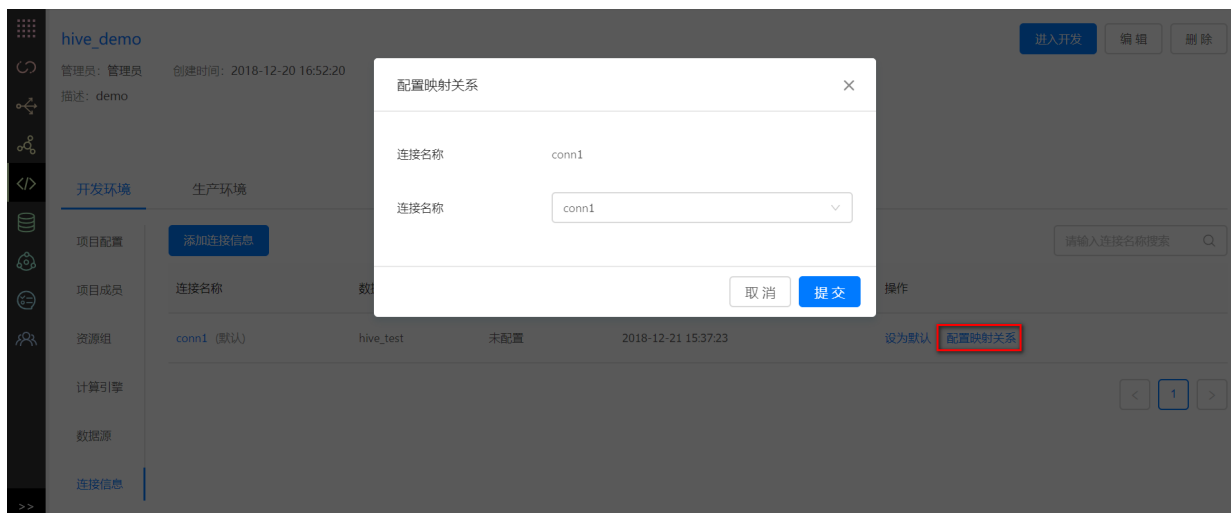
连接信息一般用于数据开发任务，在【项目配置】中点击【连接信息】进入连接信息配置页面，开发环境的任务只可使用开发环境的连接信息，生产环境的任务只可使用生产环境的连接信息。配置连接信息时，需要分别配置开发环境和生产环境的连接信息。



点击【添加连接信息】，在弹窗中填写配置信息，可为该项目添加一个连接。

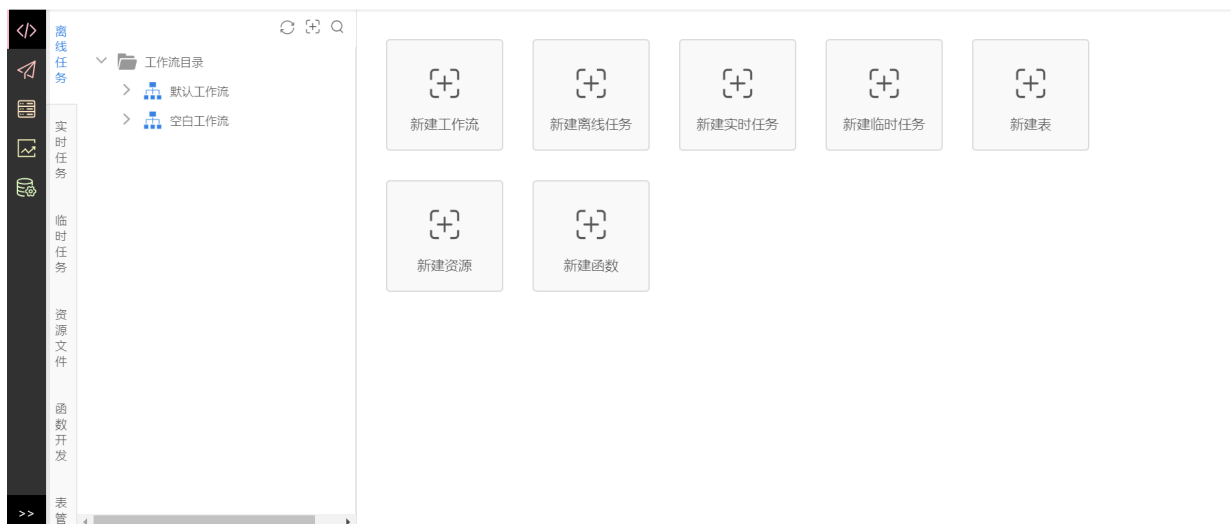


开发环境的连接信息类似数据源，也需要与生产环境的数据源进行一一映射。当开发环境的数据开发任务开发完成需要发布到生产环境时，该任务所依赖的连接信息必须在生产环境的连接信息有一个映射关系，否则该任务将不能发布到生产环境。在连接信息右侧点击配置映射关系，选择该数据源在生产环境对应的数据源，点击提交完成。



6.2 进入开发

在控制台 -> 开发平台页面，点击某个项目的【进入开发】，进入该项目的开发中心。



6.3 工作流

工作流(Flow)是完成特定业务的一组任务的集合。

6.3.1 新建

在开发中心下，选择【+】图标右击，选择新建工作流，输入自定义工作流名称，选择工作流模板，输入用途后并选择目录后，点击确定，完成工作流创建。



6.3.2 属性配置

属性配置包括基本属性、调度配置和依赖配置。

- 基本属性

基本属性包括工作流名称、工作流类型、工作流模板、责任人和用途，可修改用途，点击保存。

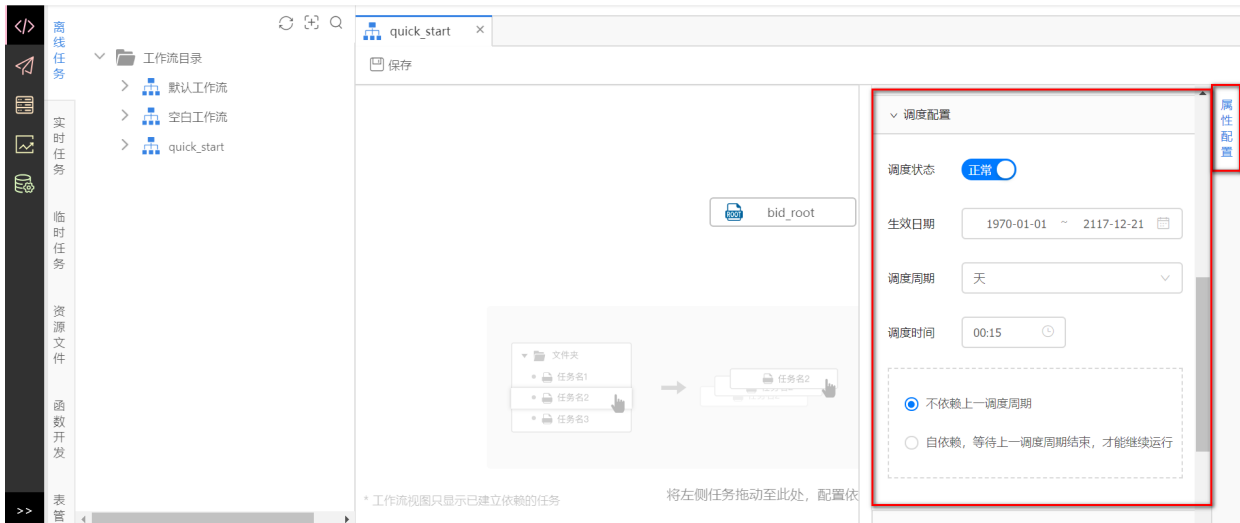


- 调度配置

调度状态包括正常和暂停。

正常调度 下包含分钟、小时、天、周、月、季度六种粒度的调度周期，不同粒度调度周期之间支持相互依赖。其中分钟和小时调度支持**自依赖**。调度时间即工作流开始运行时间，配置原则通常建议为集群常规空闲时间，一般为凌晨2点到8点。**当工作流到达调度时间时，如果上游有未完成的工作流，则此工作流不会被调度。**只有当所有上游工作流都完成后，此工作流才会被调度开始运行。

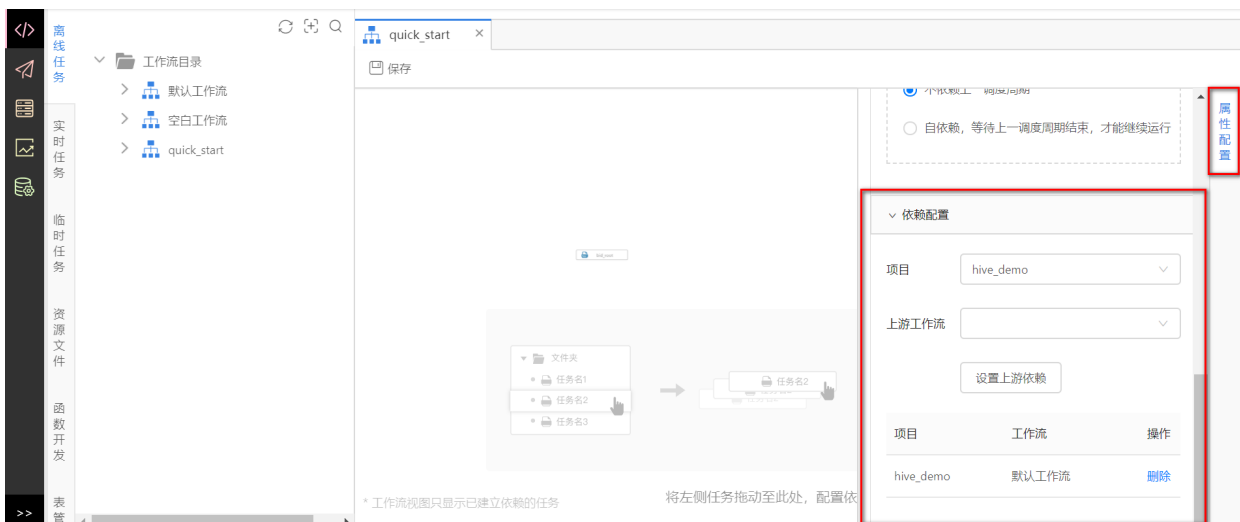
自依赖 定义: 选择添加自依赖后，每天的实例会依赖于前一天的实例(串行关系)，只有前一天的实例运行完成后，当天的实例才会开始执行。如果未选择自依赖，每天的实例是并行的，会同时开始调度运行。



● 依赖配置

每个工作流可以配置一个或多个上游工作流，当所有的上游工作流都运行完成且到达当前任务的调度时间后，此工作流才会被开始调度。工作流依赖支持跨项目依赖，依次选择项目，上游工作流后，下方显示上游工作流列表为添加的工作流。

注意：工作流依赖不能形成闭环。

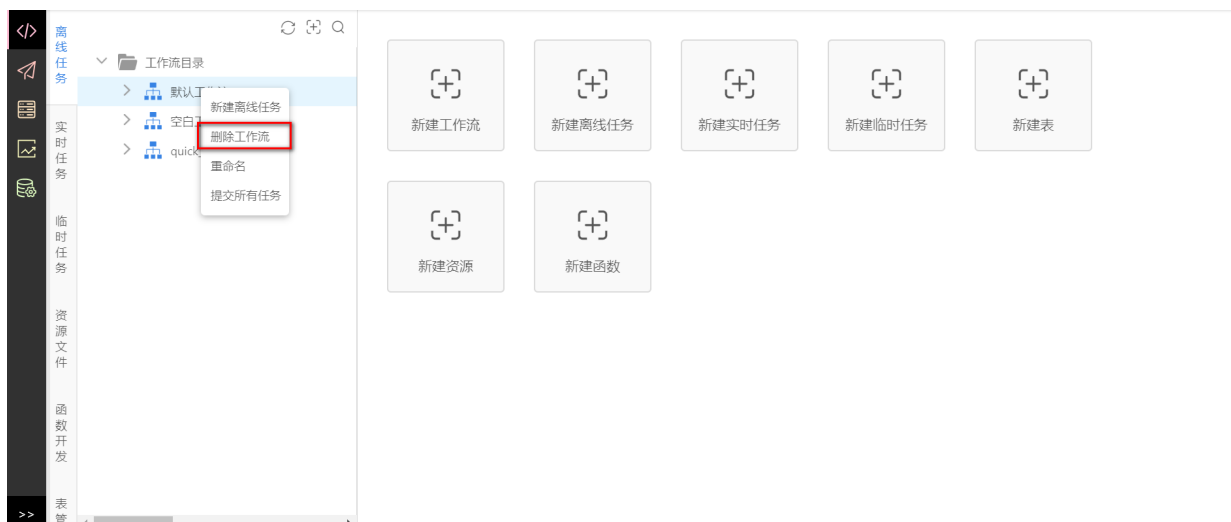


配置上游工作流，也可点击**设置上游工作流依赖**，采用图形化的界面配置此工作流上游依赖，在搜索框中分别选择项目和工作流，自动设为上游工作流，点击确定完成。



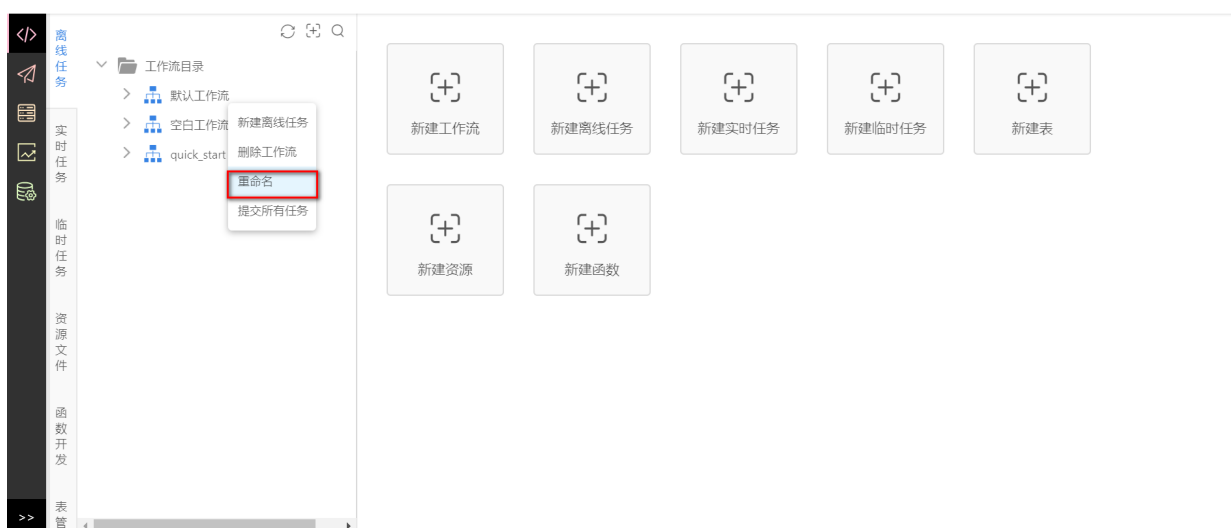
6.3.3 删除

右击点击要删除的工作流，选择**删除工作流**，点击**确定**，完成删除。删除时会检测该工作流的依赖，若存在依赖，则无法删除该工作流。



6.3.4 重命名

右键点击要重命名的工作流，选择**重命名**，输入新名称，新名词不能与其他工作流重名，点击**确定**，完成重命名。

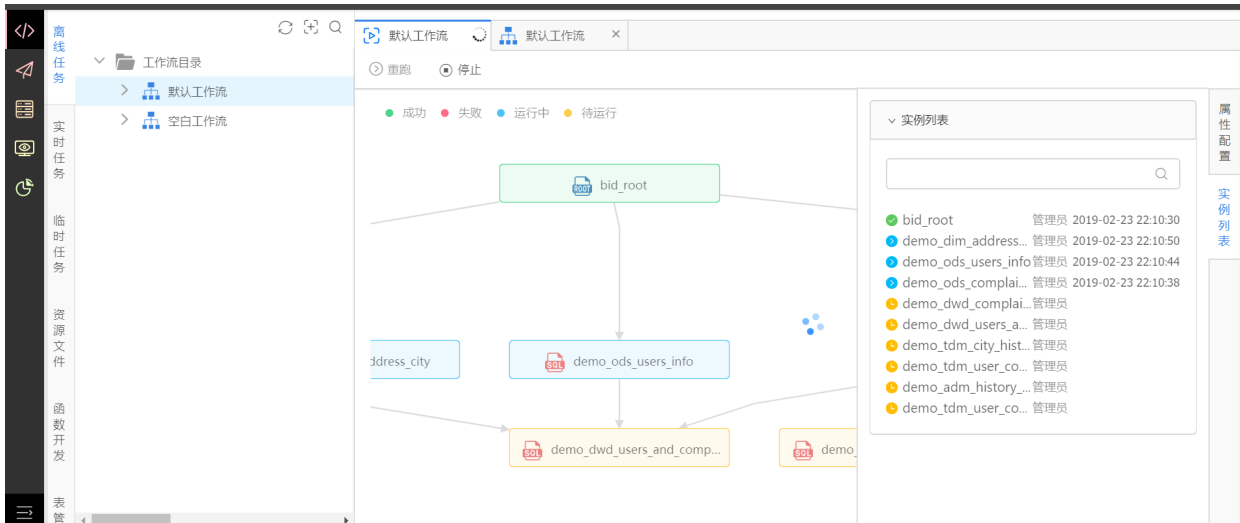


6.3.5 运行

点击运行按钮，弹出一个工作流实例页面，该页面动态显示工作流整体运行情况。若节点数量<300个，一次显示全部节点，否则只展示 `bid_root` 节点及下一层。工作流运行状态分为成功、失败、运行中、待运行，以不同颜色区分。

操作：右键某个任务节点，可选择展开父节点、展开子节点、终止、重跑、重跑下游、置成功。

实例列表：点击右侧实例列表，显示工作流实例列表，可以根据实例名称进行筛选。点击某个实例，可定位到该实例。

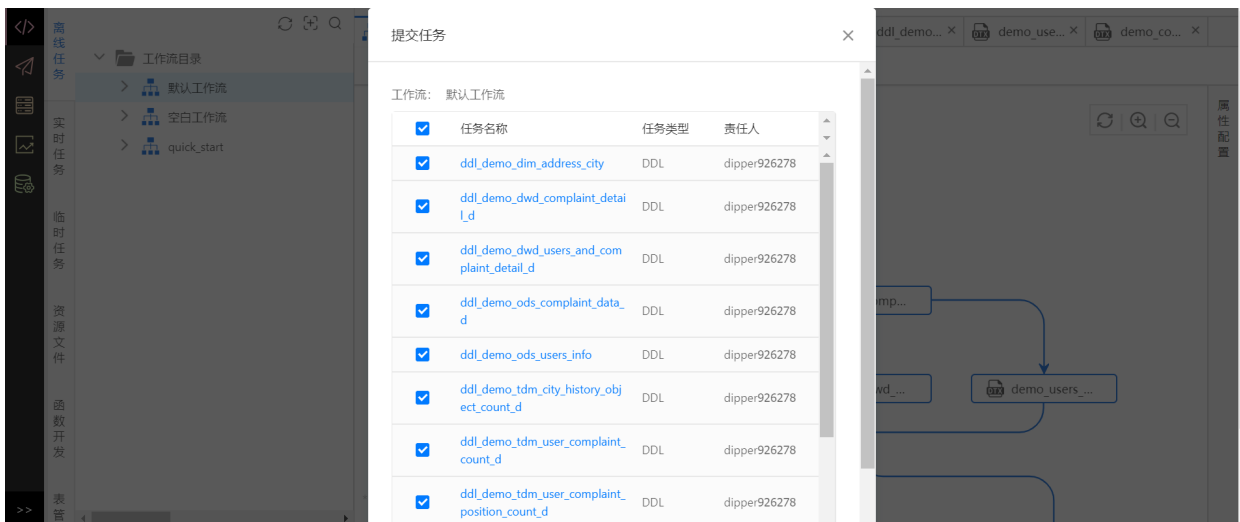


6.3.6 提交

点击提交，弹出提交窗口，支持批量提交任务，显示任务名称、任务类型、修改人和修改类型。默认全选任务，点击确定，批量提交任务。

修改类型包括新增和修改。删除的任务会自动提交到发布管理中。

列表显示依赖于bid_root或其子节点的作业，且这些任务存在新增或修改。



6.4 工作流（可视化组件模式）

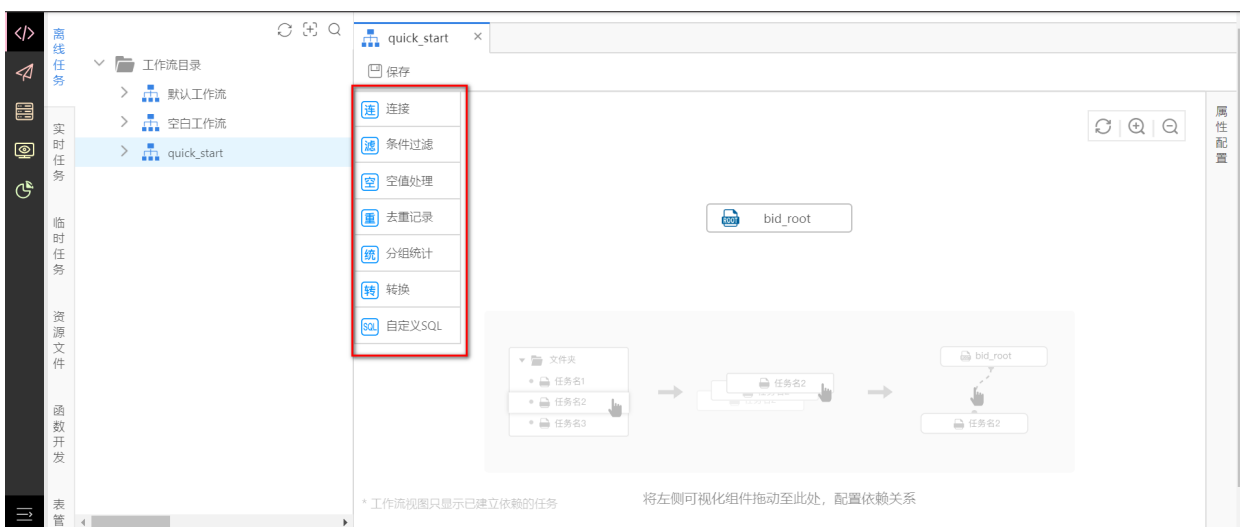
工作流(可视化组件模式)主要用来可视化ETL加工。可直接通过简单的拖拽组件，配置组件依赖关系和基本信息，完成数据加工。

(1) 新建

在开发中心下，选择+图标右击，选择新建工作流（可视化组件模式），输入自定义工作流名称，输入用途后并选择目录后，点击确定，完成工作流创建。

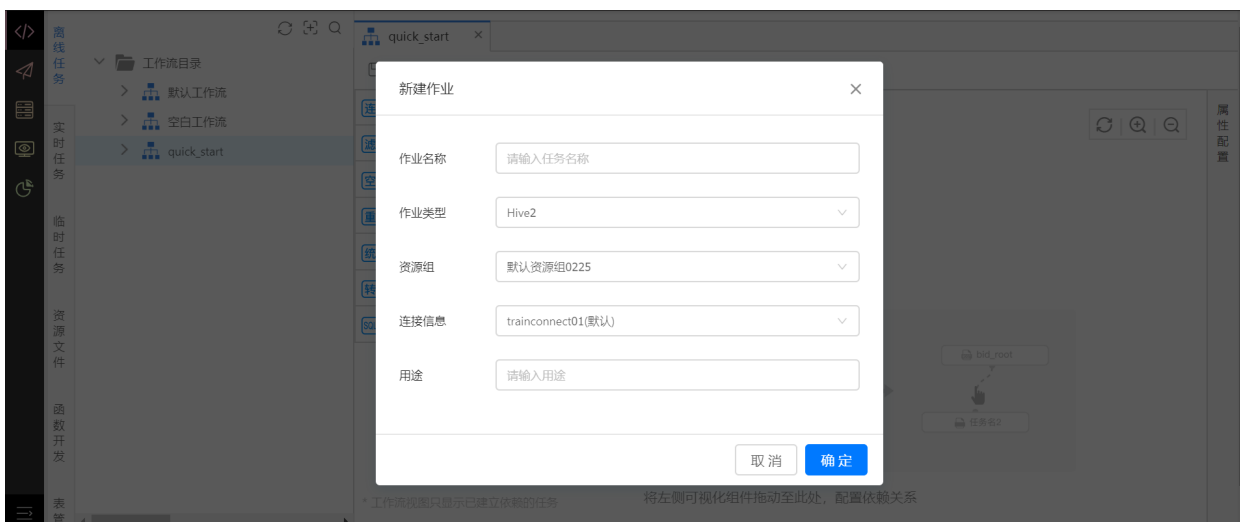


(2) 新建成功后，可看到7种工作类型节点，包括连接、条件过滤、空值处理、去重处理、分组统计、转换和自定义SQL。



(3) 组件拖入界面中，弹出表单页，填入作业名称，选择资源组和连接信息，点击确定完成。

作业类型由当前项目计算引擎类型决定。



(4) 完成后将在右侧弹出组件配置，不同组件配置信息不同，配置好后点击保存。



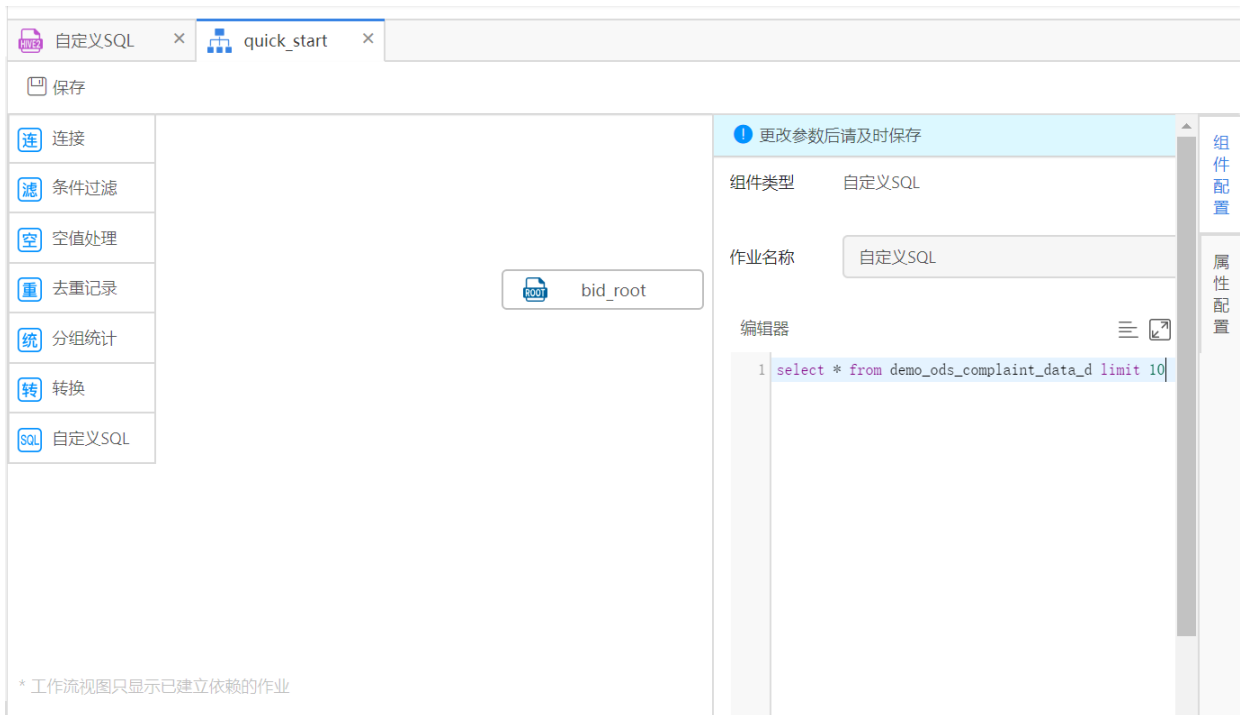
以下将对各个组件做详细介绍：

- 自定义SQL

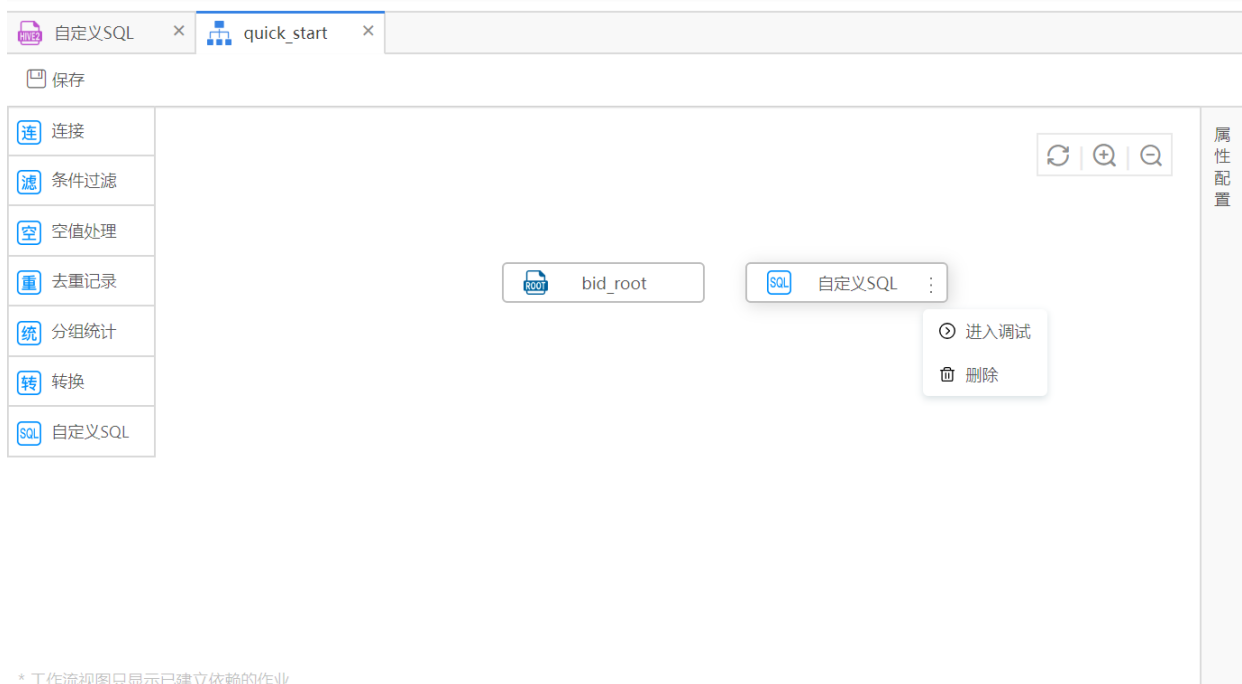
自定义SQL用来编写自定义SQL语句，将 **自定义SQL** 组件拖入到面板中，输入作业名称为**自定义SQL**，点击确定。

在编辑框中输入自定义代码，点击保存。

```
select * from demo_ods_complaint_data_d limit 10;
```



点击 **进入调试**，可对组件进行调试，测试代码是否正确，若不成功，可查看日志。



- 转换

转换组件支持将字段进行转换，如将字段内容进行替换，转换大小写等。将 **转换** 组件拖入到面板中，弹出表单如图所示，输入作业名称：**转换**，点击确定。

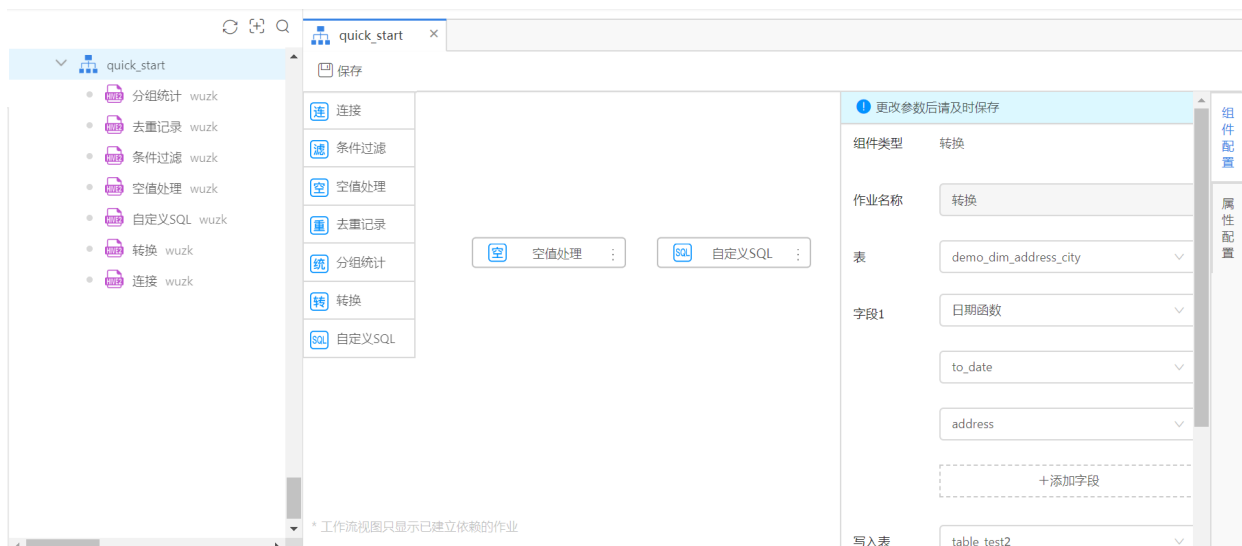
右侧弹出组件配置，配置表，字段，函数类型、函数、写入表等信息。函数类型包括日期函数、字符串函数和转换函数。当任务类型为hive时，

日期函数包括：datediff、to_date、year、month、day、weekofyear、date_sub、date_add、from_unixtime、unix_timestamp。

字符串函数包括：length、trim、ltrim、 rtrim、upper、lower、substr、substring、concat、concat_ws

任务转换函数包括：int、tinyint、smallint、float、double、decimal、string、varchar、char、binary、timestamp、date。

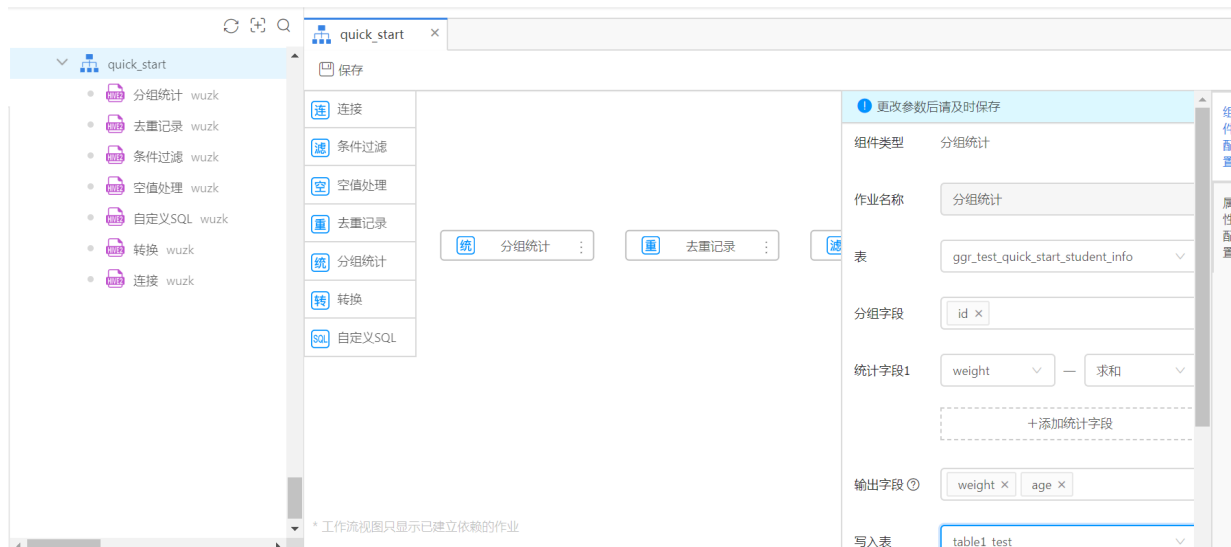
下图中选择了日期函数，to_date，可以返回日期时间字段中的日期部分。选择字段、写入表后点击保存完成。



- 分组统计

分组统计支持将字段进行分组并进行统计。将 **分组** 组件拖入到面板中，弹出表单，输入作业名称为**分组确定**。

右侧弹出组件配置，配置表，分组字段，统计字段、统计方式、输出字段、写入表等信息。统计方式包括求和、计数和去重计数。完成后点击保存。



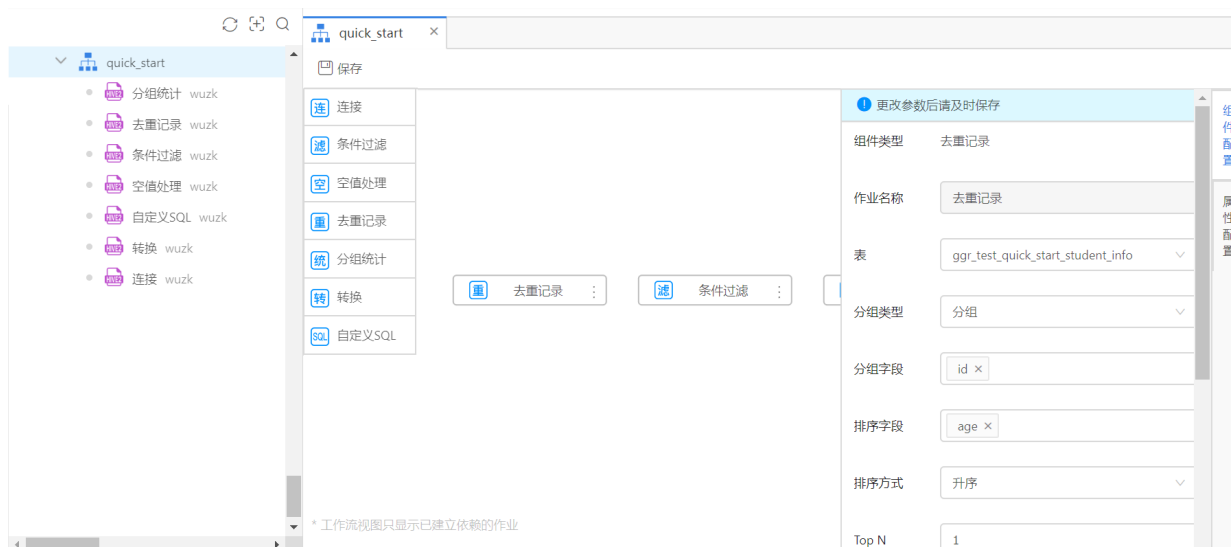
- 去重记录

去重记录可按分组进行去重，将 **去重记录** 组件拖入到面板中，弹出表单，输入作业名称为**去重记录**，点击确定。

右侧弹出组件配置，配置分组类型、写入表等信息。分组类型包括全选和分组。

若选择**全选**，去重完全相同的记录。

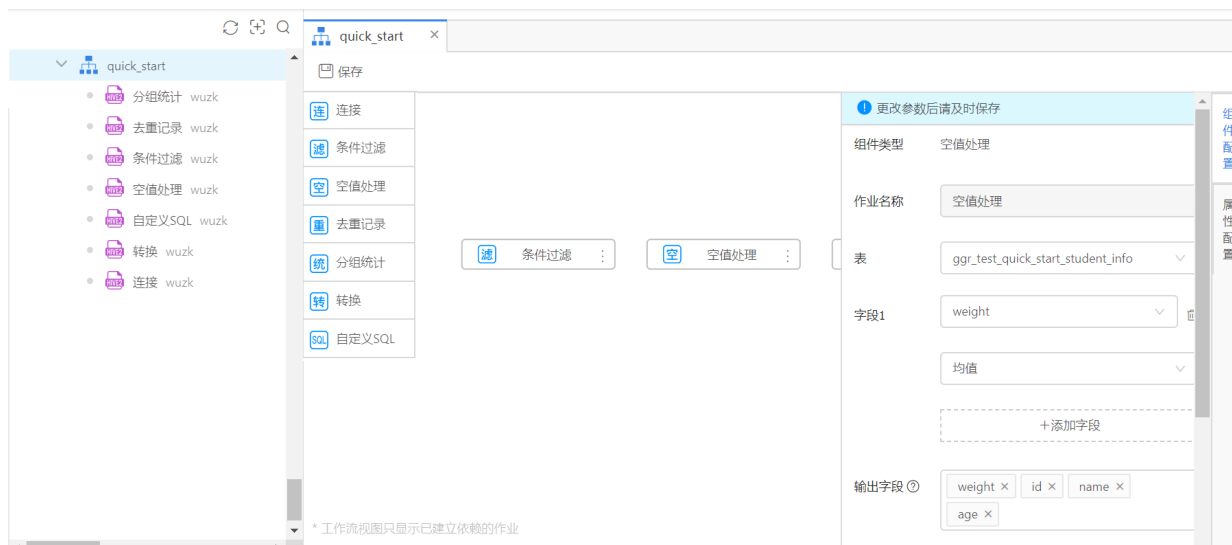
若选择**分组**，选择分组字段，排序字段、排序方式和TopN，将按排序字段包括分组字段中的前N条记录。



- 空值处理

空值处理支持将空值进行自定义处理，将 **空值处理** 组件拖入到面板中，输入作业名称为**空值处理**，点击**确定**。

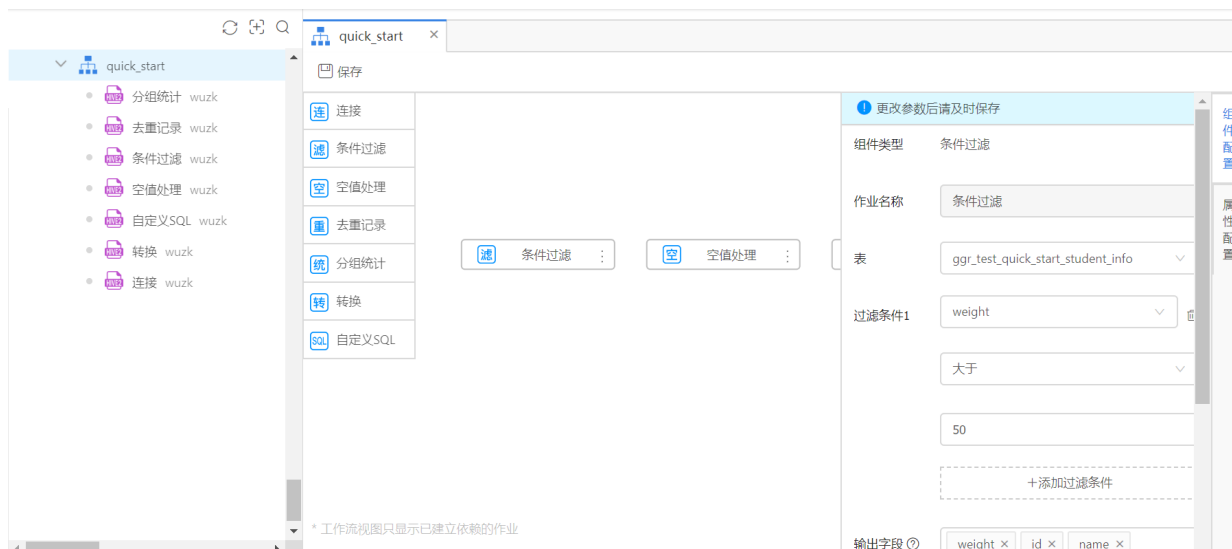
右侧弹出组件配置，配置字段、处理方式、输出字段、写入表等信息。处理方式包括均值、默认值和众数。完成后点击保存。



● 条件过滤

条件过滤支持对记录进行过滤，将 **条件过滤** 组件拖入到面板中，弹出表单，输入作业名称为**条件过滤**，点击 **确定**。

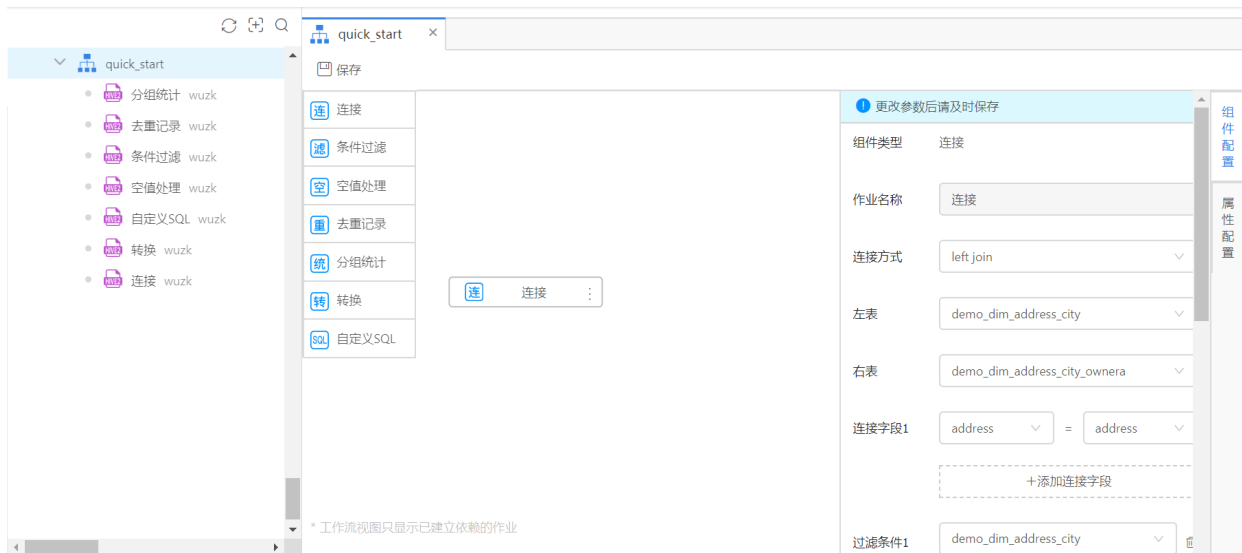
右侧弹出组件配置，配置字段、处理方式、输出字段、写入表等信息。处理方式支持：大于、小于、等于、介于、不等于。完成后点击保存。



● 连接

连接支持对两张表的连接，将 **连接** 组件拖入到面板中，弹出表单，输入作业名称为**连接**，点击 **确定**。

右侧弹出组件配置，配置连接方式、当任务类型为hive时，包括left join、right join 和full join，左表、右表、连接字段、过滤条件、输出字段和写入表等信息。



6.5 离线任务

6.5.1 新建任务

在开发中心下，选择 workflow 或 workflow 下的目录右击，选择新建离线任务，输入自定义任务名称，选择任务类型后，点击确定，完成任务创建。



新建 SparkSQL 类型的任务 `quick_start_student_statistics`，然后在代码框中输入以下代码。

```
-- 1. 查询学生基本信息
select
    id
    , name
    , age
    , weight
from quick_start_student_info
order by age;

-- 2. 查询学生最大年龄、最小体重
select
    max(age)
    , min(weight)
```

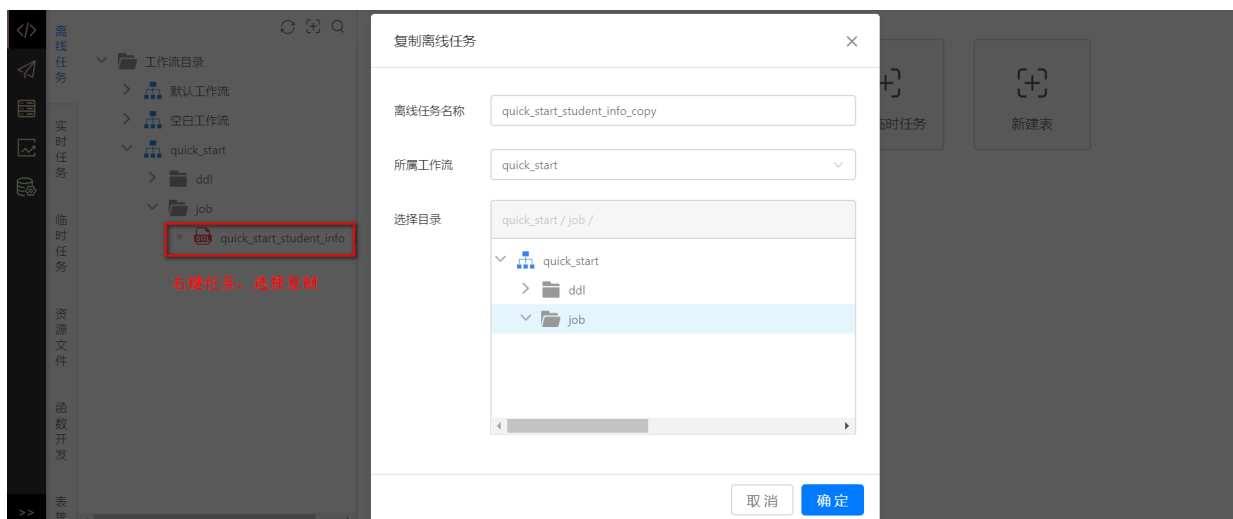


```
from quick_start_student_info;
```

6.5.2 复制任务

在指定任务上右键单击后，选择复制，新任务名默认为 `原任务名_copy`，用户也可自定义任务名、工作流以及目录。

当任务被复制到其他工作流后，新任务的属性配置-上游任务会清空；同一个工作流内的任务复制，新任务的属性配置-上游任务不会被清空。

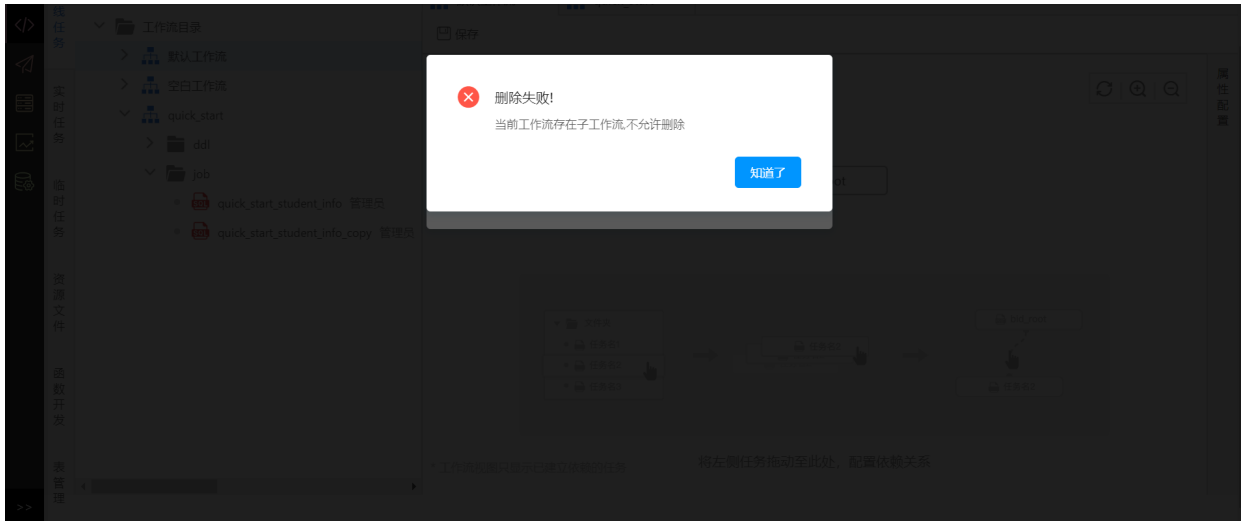


6.5.3 删除任务

右击要删除的任务，选择删除离线任务，点击确定，完成删除。



如果被删除的任务正在被其他任务所依赖（被删除任务是其他任务的上游任务），则无法删除该任务。此时，需要配置其所有子任务不依赖当前任务后，才能删除此任务。



6.5.4 格式化

格式化语句，选中想要格式化的一条完整SQL语句，点击【格式化】。

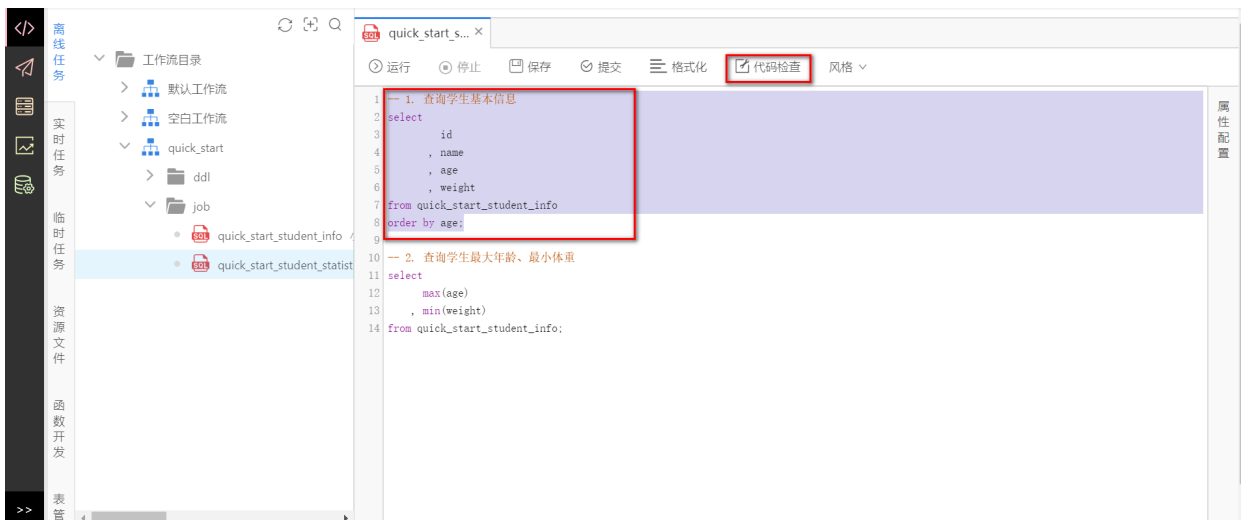
未选中代码相当于全选。



6.5.5 代码检查

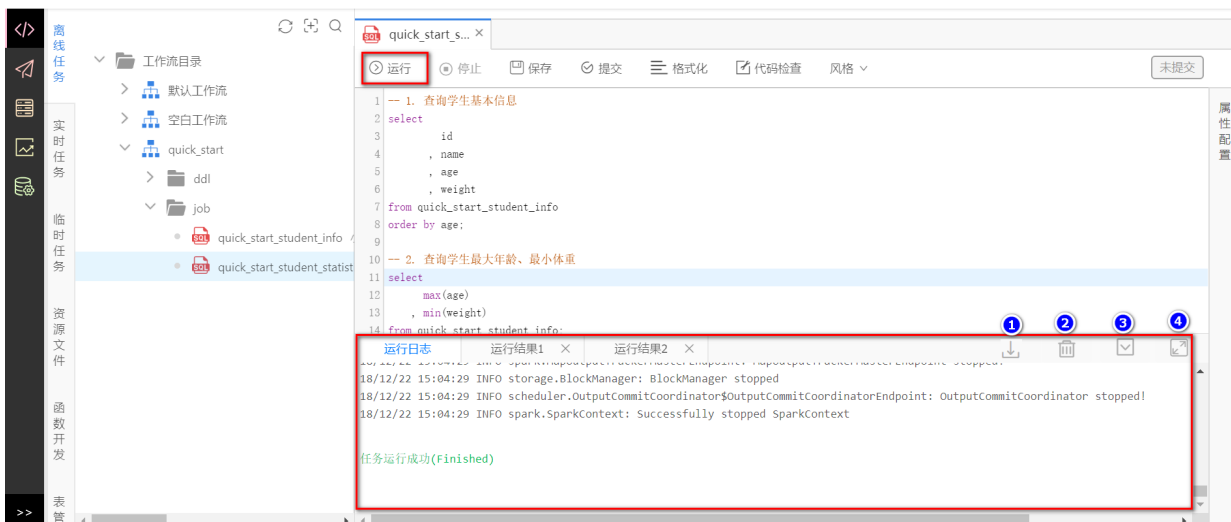
选中需要检查语句，点击【代码检查】，进行语法校验。

目前仅支持基本的SQL语法检查。未选中代码相当于全选。



6.5.6 运行

直接点击【运行】按钮会运行所有语句。当需要运行某一语句时，选中要执行的语句，点击【运行】即可。



- (1) 下载：点击下载按钮，下载日志到本地。
- (2) 清空已有日志：点击清空按钮，清空已有日志。
- (3) 展开/缩起：点击展开/缩起按钮，可展开/缩起日志窗口。
- (4) 全屏：点击全屏按钮，可全屏显示日志信息。

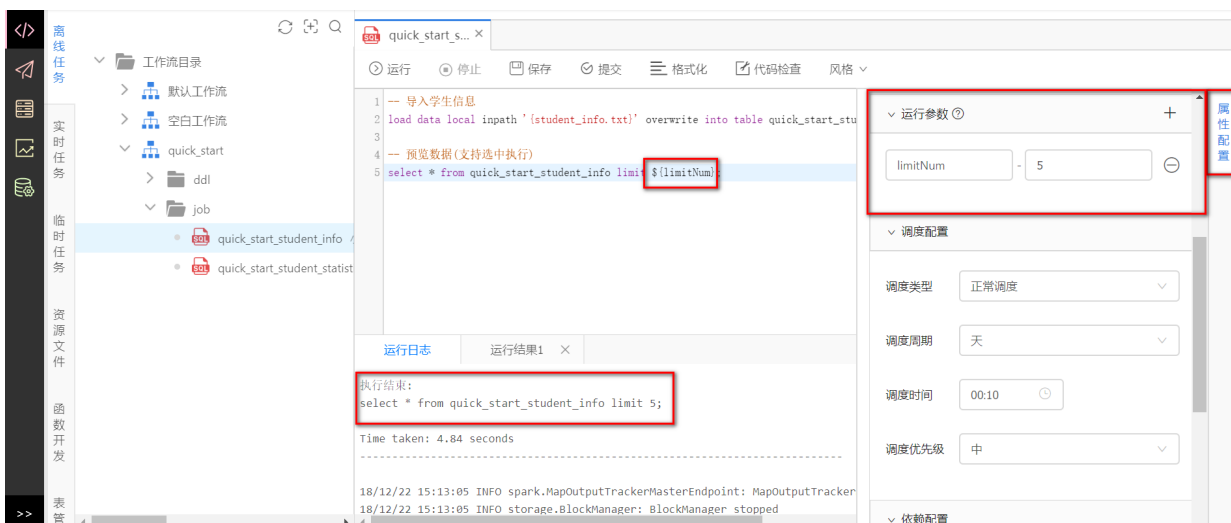
6.5.7 属性配置

- 运行参数

用户自定义参数

在属性配置->运行参数中自定义任务参数，左侧是变量名，右侧是变量值。然后在代码中使用 $\${变量名}$ 即可，运行时会自动把变量名替换为变量值。

如下图所示，定义变量名limitNum的值是5，运行任务 quick_start_student_info 时自动会把 $\${limitNum}$ 替换为5。



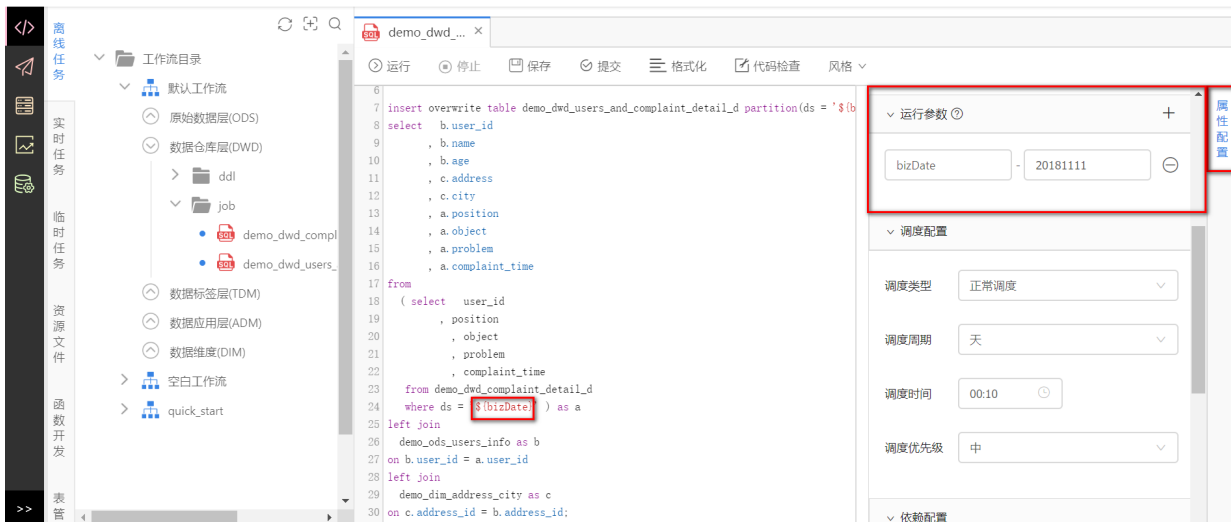
- 系统参数

系统参数是系统自带的默认参数，目前支持参数cyctime和 bizDate。

cyctime：定义为一个实例的定时运行时间，默认格式为：yyyymmddhh24miss。

bizDate：是指业务日期(前一天)，默认格式为 yyyymmdd 。例如：今天日期为"2018-02-21"，则业务日期bizDate="20180220"。同时平台支持\${bizDate+n}或\${bizDate-n}的写法，如\${bizDate+1}，则自动替换为“20180221”。

在开发环境中，手动配置bizDate的参数值；在生产环境中，当任务在每天的周期调度过程中，系统自动会把bizDate替换为前一天的值。

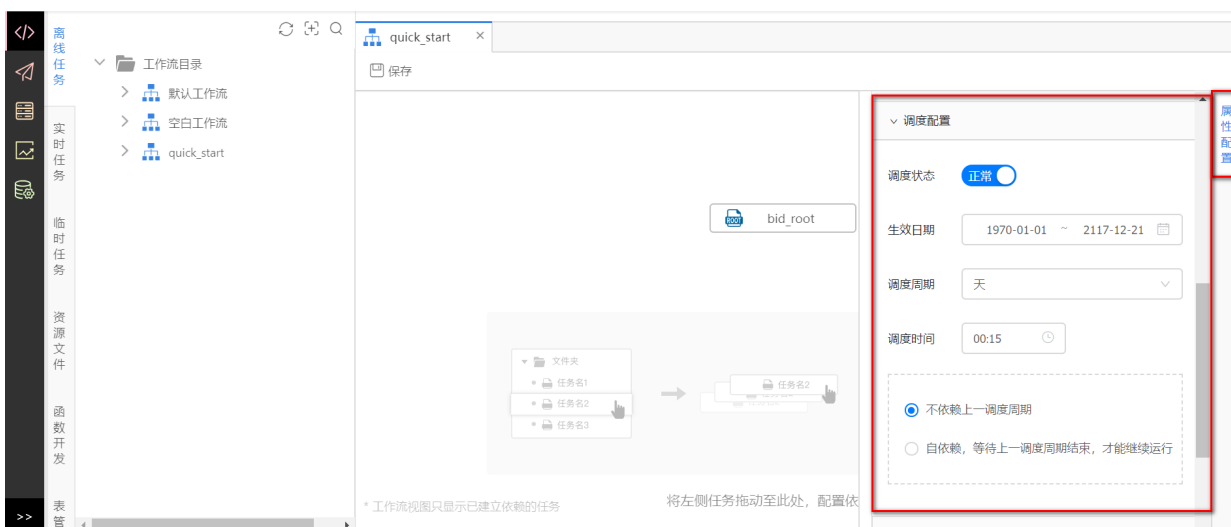


● 调度配置

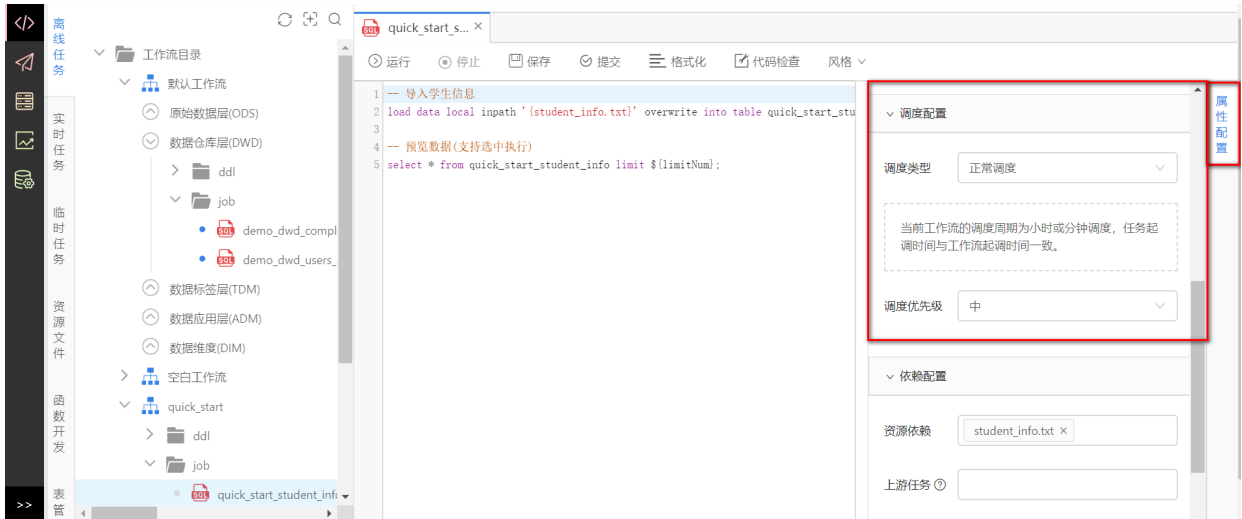
调度类型分为 正常调度、空跑 和 暂停调度 3中类型。

正常调度：正常调度下可选天、周、月三种粒度的调度周期，不同粒度调度周期之间支持相互依赖。调度时间即任务开始运行时间，配置原则通常建议为集群常规空闲时间，一般为凌晨2点到8点。当任务到达调度时间时，如果上游有未完成任务，则此任务不会被调度。只有当所有上游任务都完成后，此任务才会被调度开始运行。

当 workflow 调度周期为小时、分钟，任务调度与 workflow 一致 当 workflow 调度周期为周、月，任务调度与 workflow 一致，可配置调度时间 当 workflow 调度周期为天，任务调度周期支持天，周，月



若该任务所依赖的 workflow 为小时或分钟调度，则任务调度不可配置，只能与 workflow 保持一致。



空跑：任务调度类型配置为空跑时，不会真正运行任务，而是直接将任务置为成功。

暂停调度：任务调度类型配置为暂定调度后，则在每次的周期调度中，此任务不会被系统所调度。任务配置 **暂定调度** 时，要求此任务不能有下游任务。

目前开发环境默认未开启周期调度，因此此配置对开发环境并不生效。当任务经过提交、发布到生产环境后，此任务会被周期调度执行。

- 依赖配置

依赖配置包含 **资源依赖** 和 **上游任务** 两种，前者选择此任务需要用到的资源，后者则是配置此任务的上游任务。在代码中使用 **{资源名}** 便可引用此资源，例如选择的资源依赖是 `student_info.txt`（支持多个资源依赖），则在代码中使用 `{ student_info.txt }` 即可。

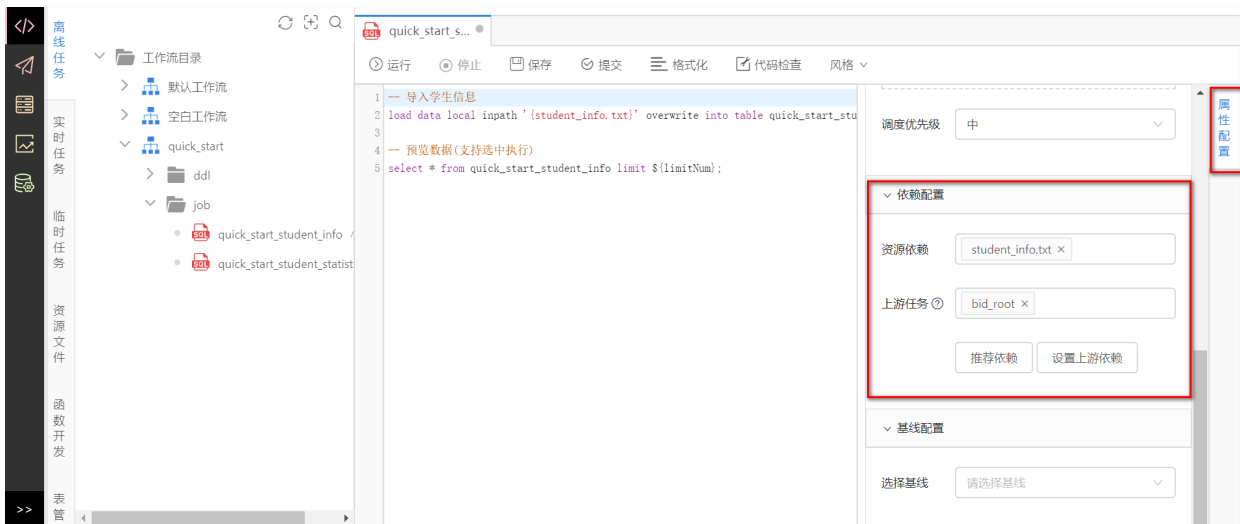
```
-- 导入学生信息
```

```
load data local inpath '{student_info.txt}' overwrite into table  
quick_start_student_info;
```

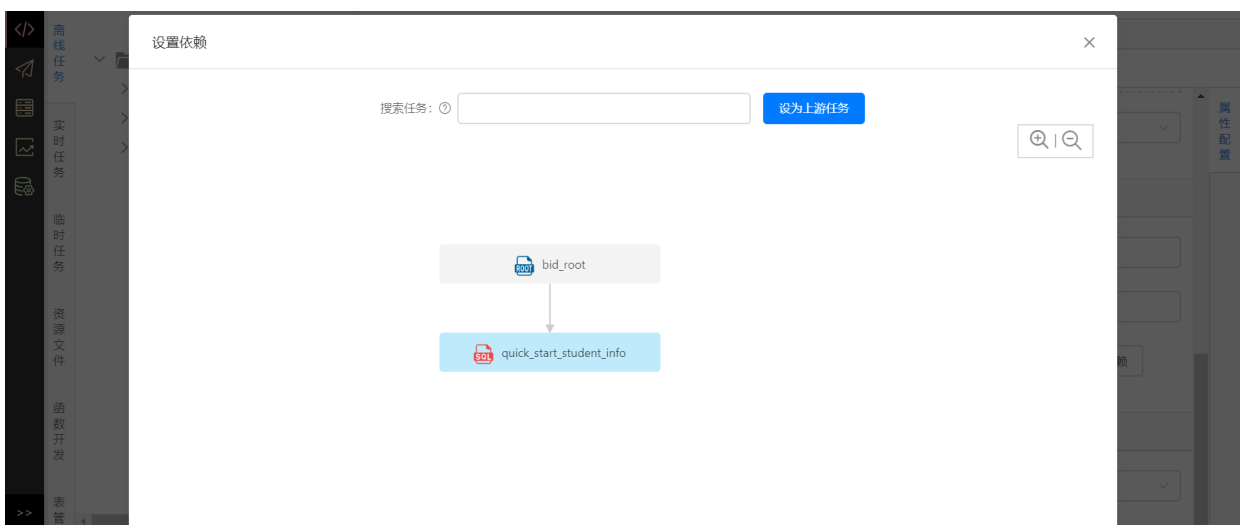
每个任务可以配置一个或多个上游任务，当所有的上游任务都运行完成且到达当前任务的调度时间后，此任务才会被开始调度。**每个任务在提交前，必须至少配置一个上游任务**，每个工作流默认会自带一个根任务 `bid_root`。

平台为DDL类型的任务默认配置了一个上游任务，是任务所属工作流的根任务 `bid_root`。

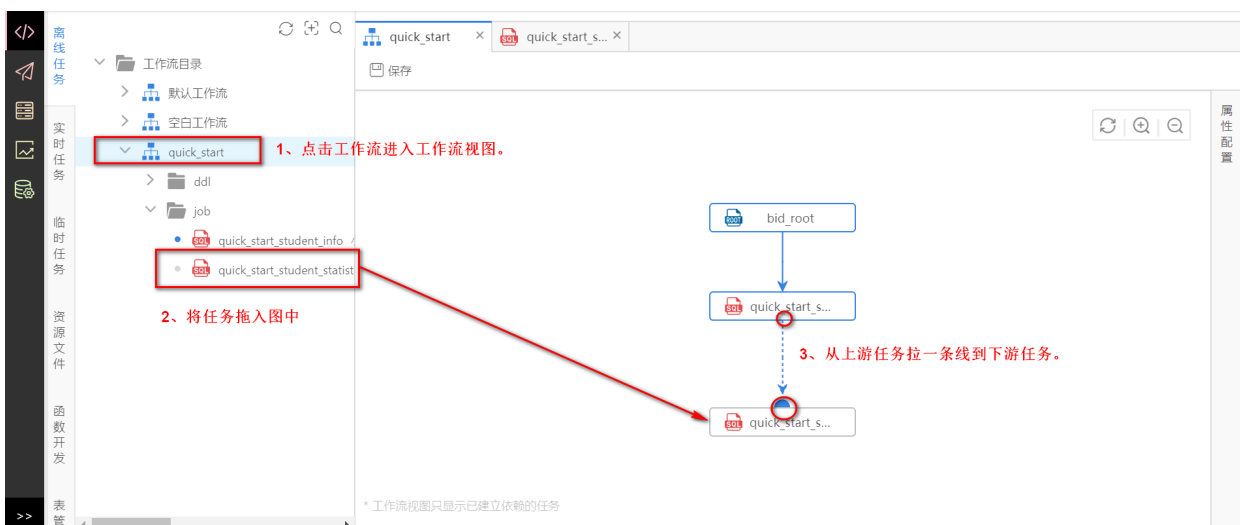
资源依赖、上游任务的配置方式如图所示。



配置上游任务，也可点击**设置上游任务依赖**，采用图形化的界面配置此任务上游任务依赖，在搜索任务框中选择任务，点击**设为上游任务**即可。



在工作流的可视化界面中也可编辑任务直接的依赖关系，直接将工作流中的任务拖动到右侧，从任务下方拉一个箭头指向另一个任务，即可完成任务的依赖编辑。



● 基线配置

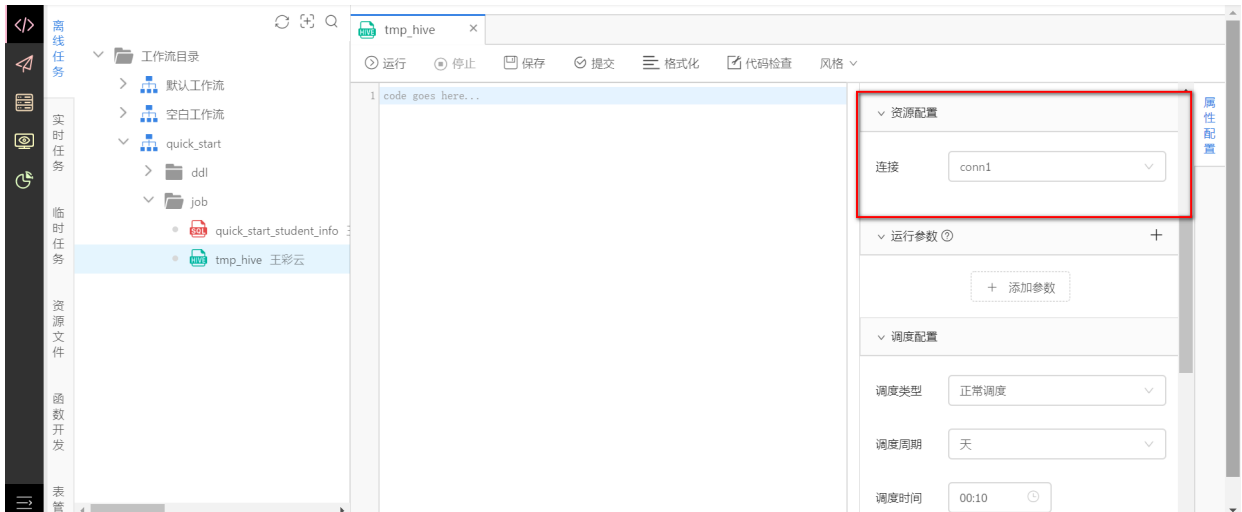
基线介绍详见**基线管理**一节。在任务的**属性配置**中的**基线配置**下可以选择设定好的基线。



● 资源配置

当任务类型为Hive2、GreenPlum、DDL、MySQL、LibrA、Oracle、SQLServer时，可为这些任务配置连接。点击右侧属性配置，选择连接。

若没有想要选择的连接，请进入【项目配置-连接信息】中添加连接。



6.5.8 提交

在开发环境下完成任务的编写和调试后，可以将这些任务发布到该项目的生产环境。发布前，需要先将任务提交到发布管理中，然后将多个任务、资源和函数组合在一起创建发布包。

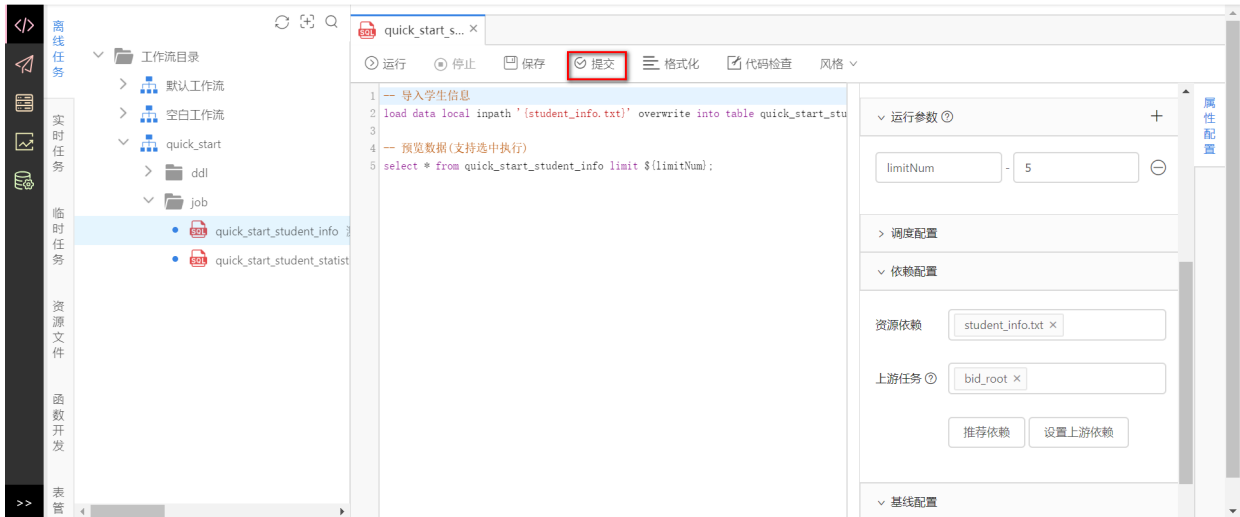
(1) 提交任务

当任务调试完成后，点击提交按钮，填写相应的提交说明后，点击确定。

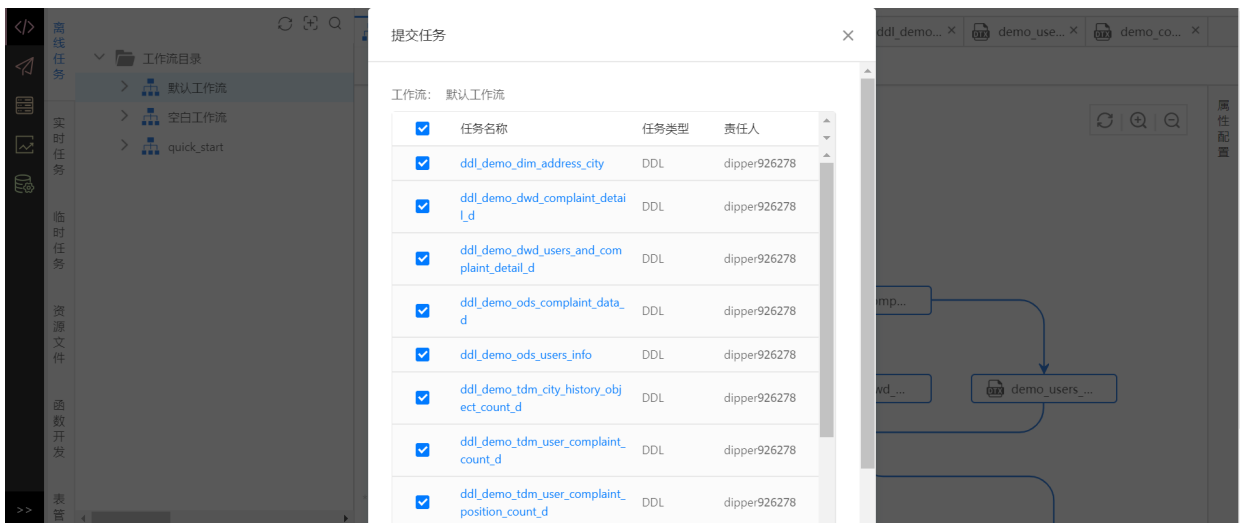
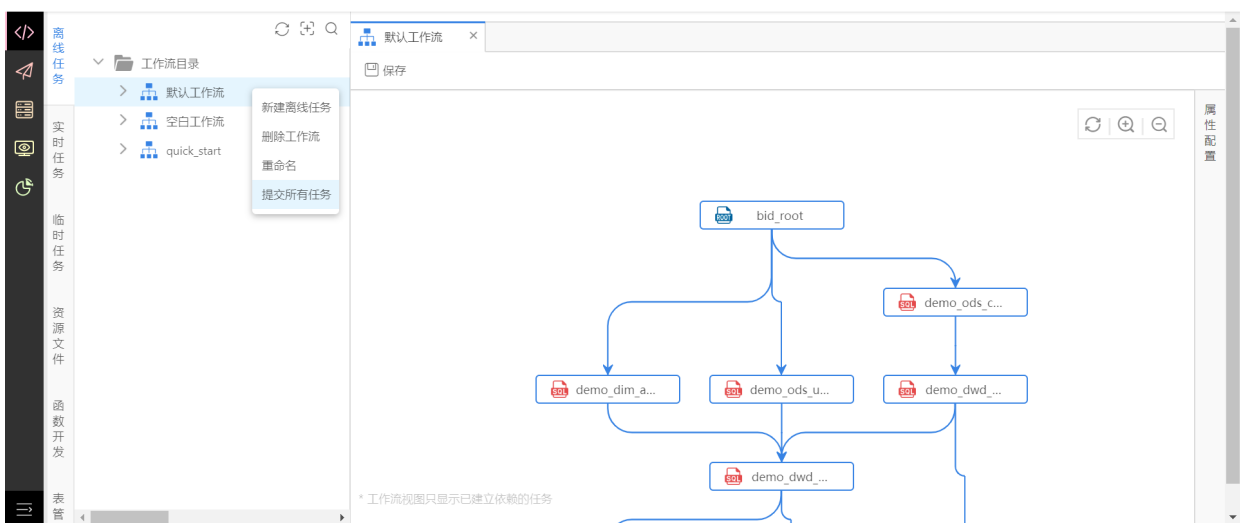
资源文件和函数会默认提交。ddl任务不需要上游依赖。

在提交任务前，请确保完成了任务的属性配置，包括运行参数配置、资源依赖和上游任务的配置。

任务的配置。



当任务较多时，也可右键 workflow 选择提交所有任务，进行提交任务。



6.5.9 任务提交发布状态

任务右上角显示任务提交发布状态，状态分为未提交、已提交+版本号、已提交（已修改）+版本号、已发布+版本号、已发布（已修改）+版本号。



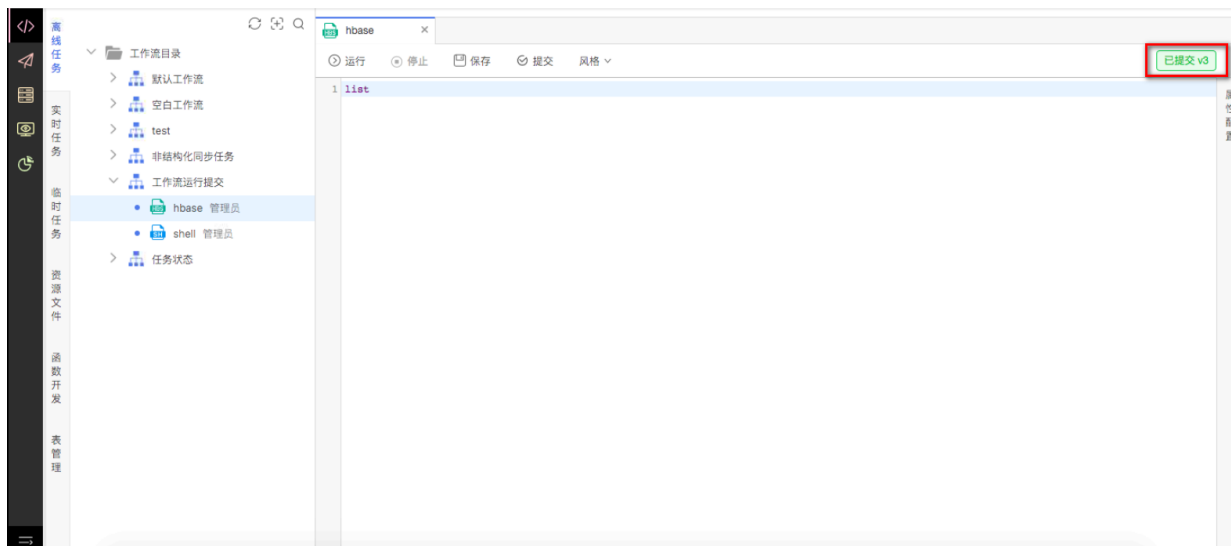
未提交：代表任务未提交。

已提交vN：代表该任务已经提交到发布管理中，版本号为N。

- 已提交vN：代表该任务已经提交到发布管理中，版本号为N，当前任务和提交的任务相比存在修改。

已发布vN：代表该任务已经发布到生产环境中，版本号为N。

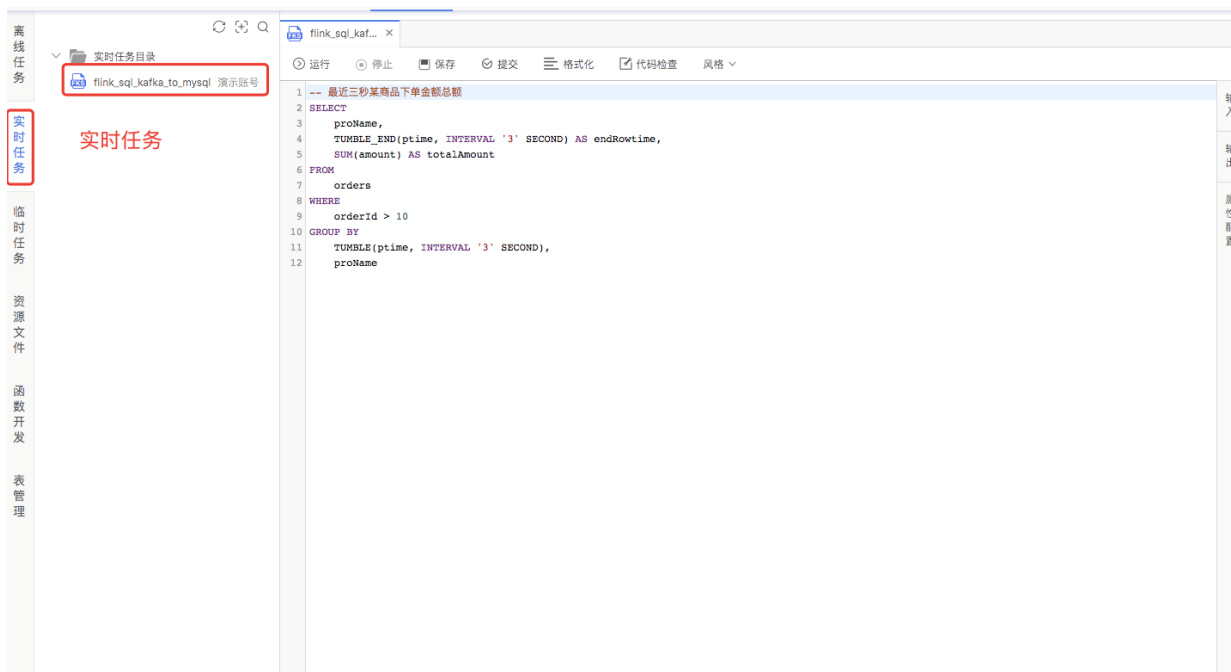
- 已发布vN：代表该任务已经发布到生产环境中，版本号为N，当前任务和发布的任务相比存在修改。



6.6 实时任务

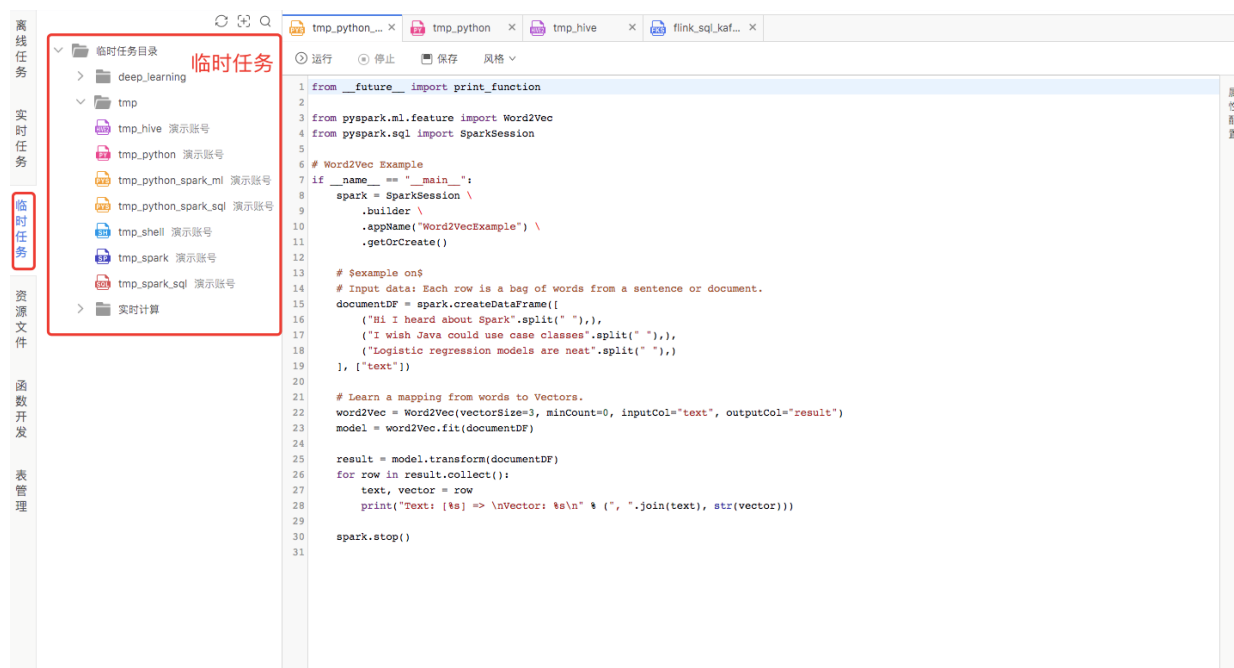
实时任务目前支持Flink、FlinkSql、Storm，sparkstreaming和实时数据同步类型的任务。实时任务的使用方法，与离线任务的用法基本一致，请参考[离线任务](#)中的内容。

实时任务日志下载需指定时间下载。



6.7 临时任务

临时任务下用于存放临时任务，使用方法与离线任务的用法基本一致，请参考离线任务中的内容。



6.8 资源文件

资源文件是指用户上传的文件，可被用于任务、函数开发中。

在开发中心的资源文件下，右击目录，选择新建资源，填写资源名称后，选中相应的资源类型，点击【添加文件】按钮来上传资源文件。



6.9 函数开发

函数(Function)指Hive、Spark、Presto、Flink等系统中的函数。除计算引擎内置的函数外，平台支持用户创建基于Hive的自定义函数(User Defined Function, 以下简称UDF)，并可直接用于Hive或SparkSQL类型的任务中。

在libs文件夹中已准备好已编译好的函数jar包：**extract_complaint.jar**。

(1) 将**extract_complaint.jar**上传到资源文件中。

(2) 在开发中心的函数开发下，右击目录，选择【新建函数】。函数名称请填写自定义的函数名；类名请填写类名全路径；资源依赖请选择该函数所依赖的jar包：**extract_complaint.jar**；用途、命令格式和参数说明选填。



6.10 表管理

表管理中可以查看表的字段信息、分区信息和数据预览。在开发中心下的表管理中，点击一个想要查看的表后，可以看到此表相关信息。



6.11 任务类型

下面每种任务类型仅供测试使用，不需要每天周期调度执行，因此统一存放到开发中心 -> 临时任务中的tmp目录下，且所有任务命名以 tmp_ 开头。

6.11.1 Shell

Shell类型用于书写shell。新建tmp_shell临时任务，输入下述测试代码，并配置运行参数name。

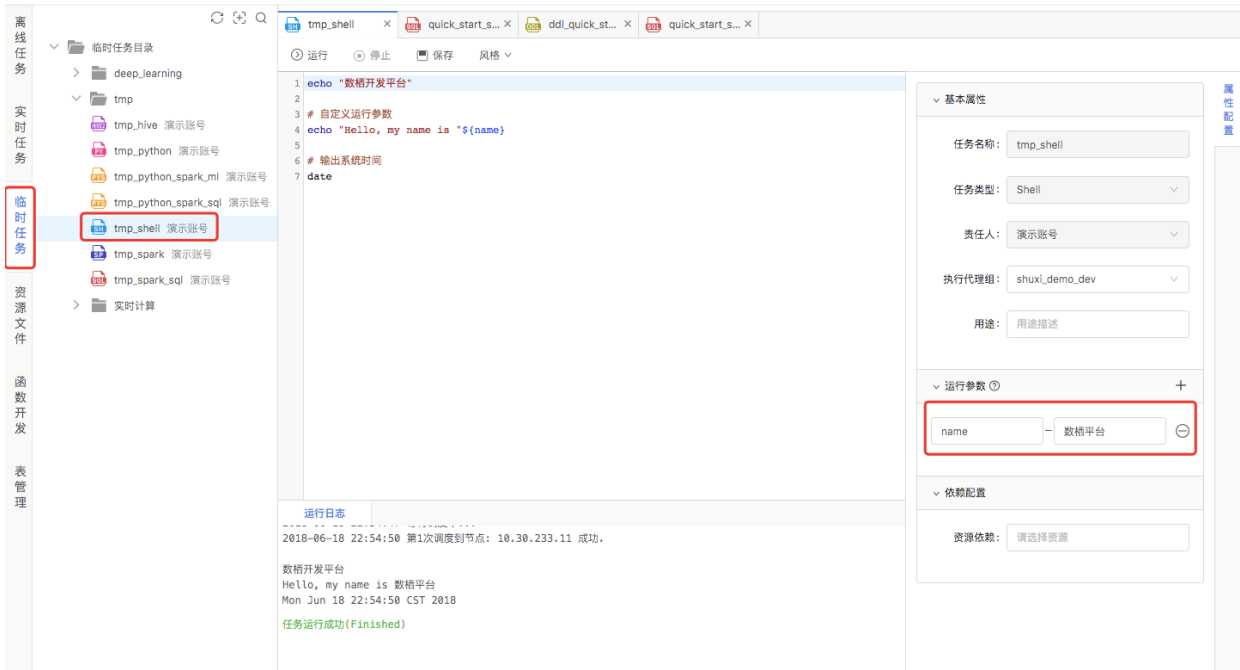
```

echo "开发门户"

# 自定义运行参数
echo "Hello, my name is "${name}

# 输出系统时间
date

```



6.11.2 数据同步

数据同步任务用于数据同步或数据交换，目的是把数据从一个数据源中同步到另一个数据源。

使用数据同步任务前需要先创建数据源，创建数据源的方式请查看[数据源管理](#)部分。

(1) 数据源配置

首先创建一个数据同步任务 `tmp_syncn`，配置源头表信息和目的表信息。

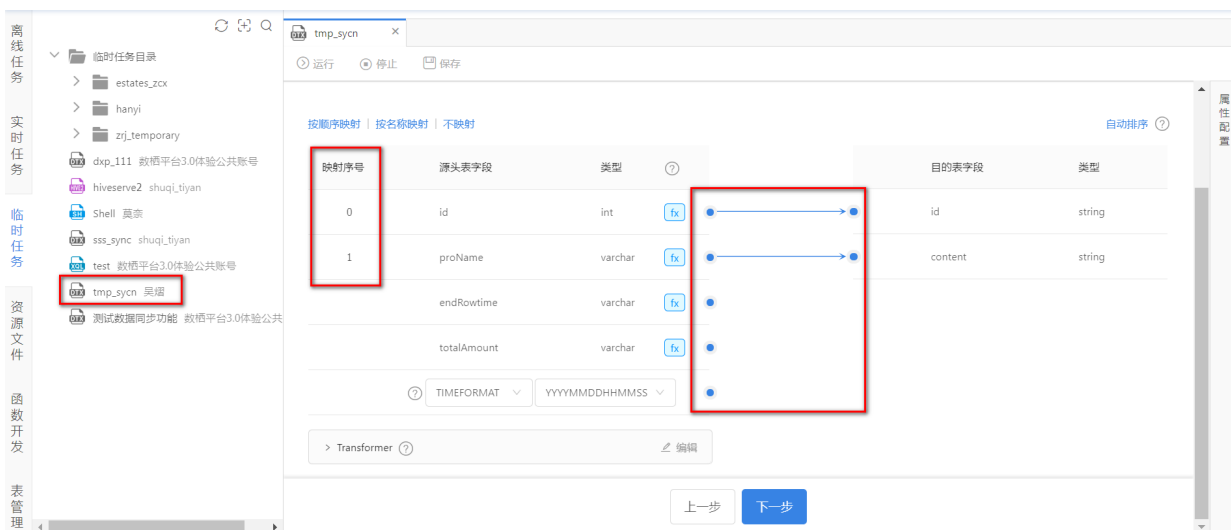


支持可变表名，可手动输入表名并配合系统参数使用，如 `${bizDate}`，`${bizMonth}`，`${bizQuar}`，`${lastDatyofMon}`。如图所示，在开发环境中，点击下一步时，将自动以 `student_201903` 作为表名称去获取该表信息，在生产环境中，每次调度时将自动根据当前调度时间替换表名。

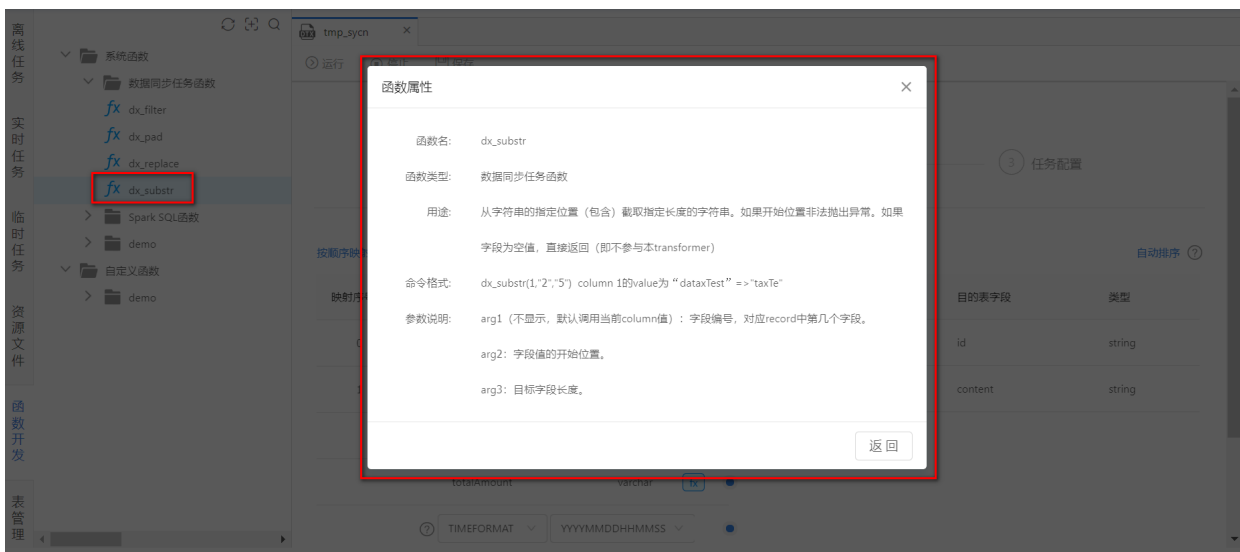
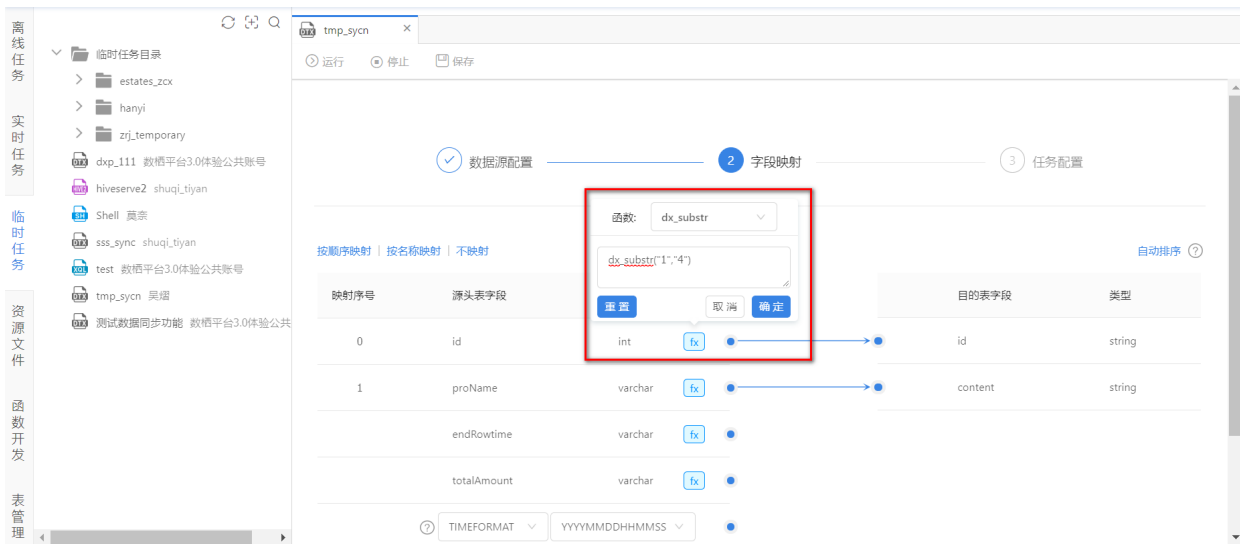


(2) 字段映射

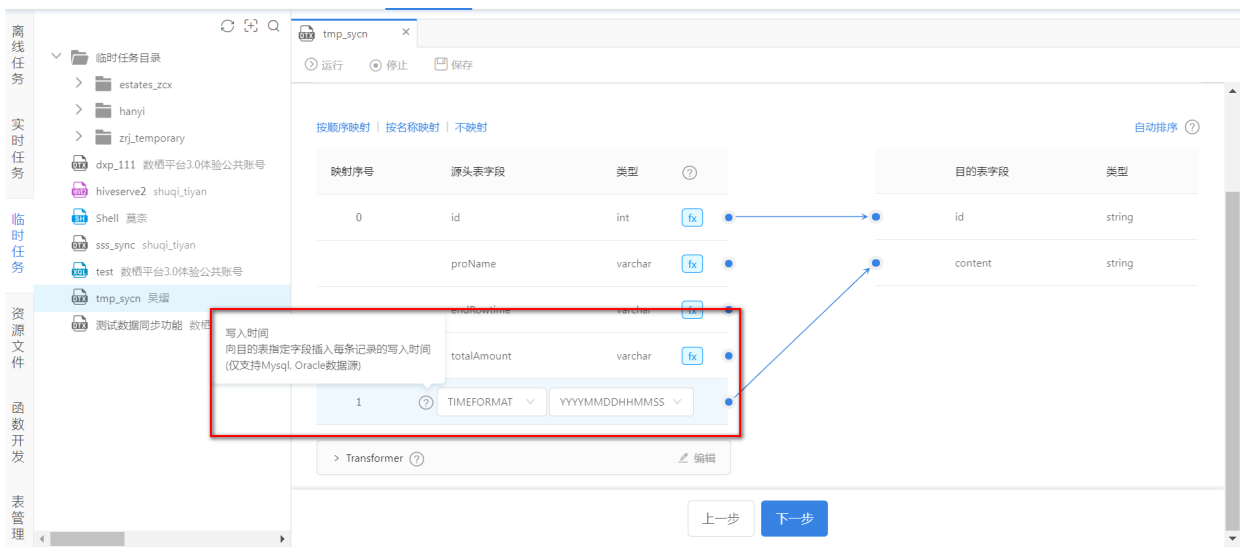
配置源头表和目的表的字段映射关系，映射序号为连线的顺序。



点击fx图标可以为该字段设置转换函数，关于函数的使用方式可在函数开发-系统函数-数据同步任务函数中查看。



在字段映射时, 可选择将写入时间做为源信息写入到目的表的某个字段。目前支持MySQL, Oracle和SQLServer。

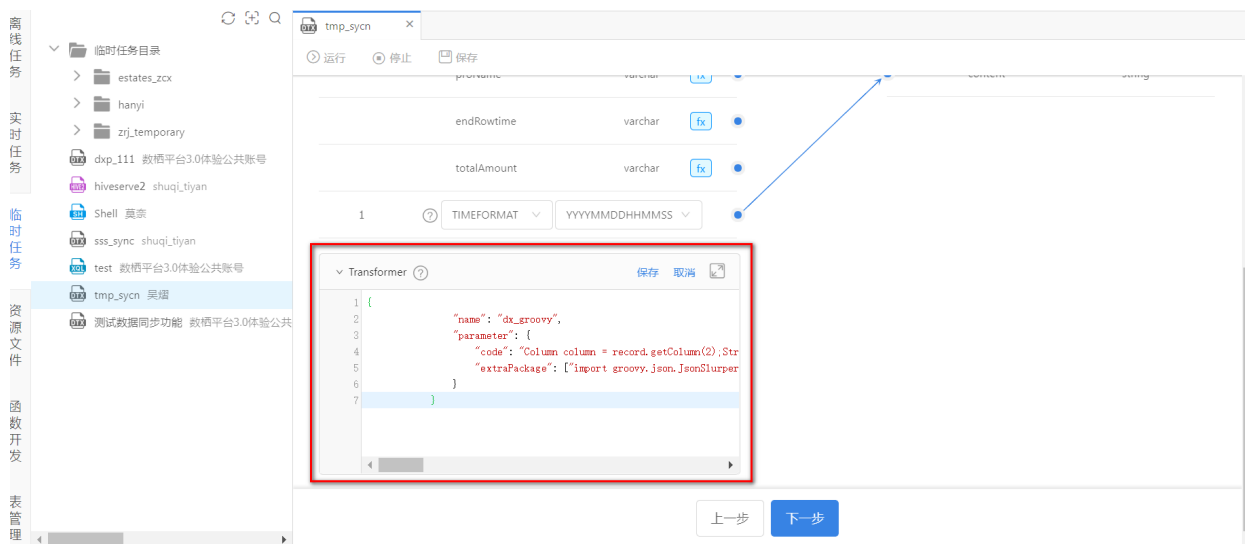


字段映射时, 当遇到复杂的使用场景时, 可通过Transformer来做字段转换功能。代码示例如下, 注意: "code"中的代码不能有换行。

```

{
    "name": "dx_groovy",
    "parameter": {
        "code": "Column column = record.getColumn(2);String
oriValue = column.asString();def jsonSlurper = new JsonSlurper();def object
= jsonSlurper.parseText(oriValue);boolean isWatched = object.isMatched;long
paidNum = 0;long noPaidNum = 0;def list = object.details;list.each {boolean
settlement = it.settlement;if(settlement)
{paidNum++;}else{noPaidNum++;}};record.setColumn(2, new
BoolColumn(isWatched));record.setColumn(3, new
LongColumn(paidNum));record.setColumn(4, new LongColumn(noPaidNum));return
record;",
        "extraPackage": ["import groovy.json.JsonSlurper;"]
    }
}

```



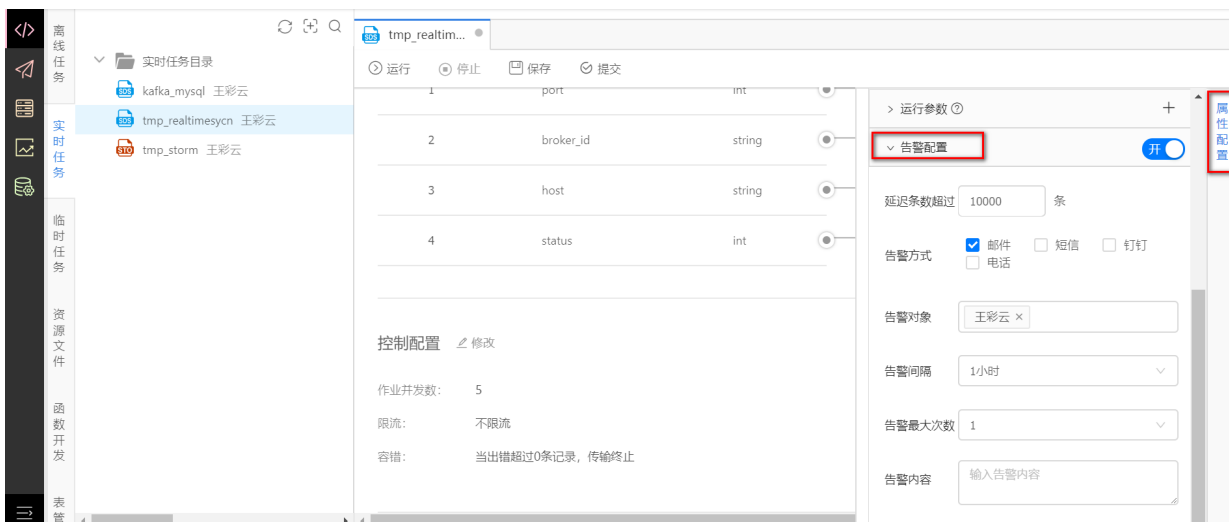
(3) 任务配置

配置最大传输速度和容错。



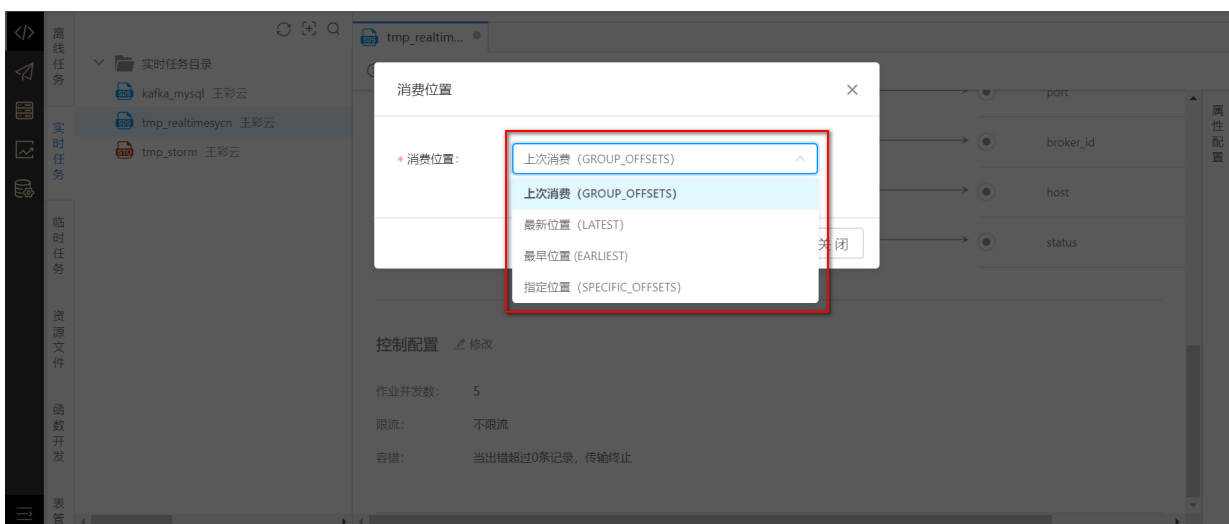
(4) 告警配置

点击属性配置，可开启告警配置，输入延迟条数等信息后，保存属性配置。



(4) 运行

点击运行，可选择消费位置，可选择上次消费位置、最新位置、最早位置或自己指定位置。点击确定后开始运行。



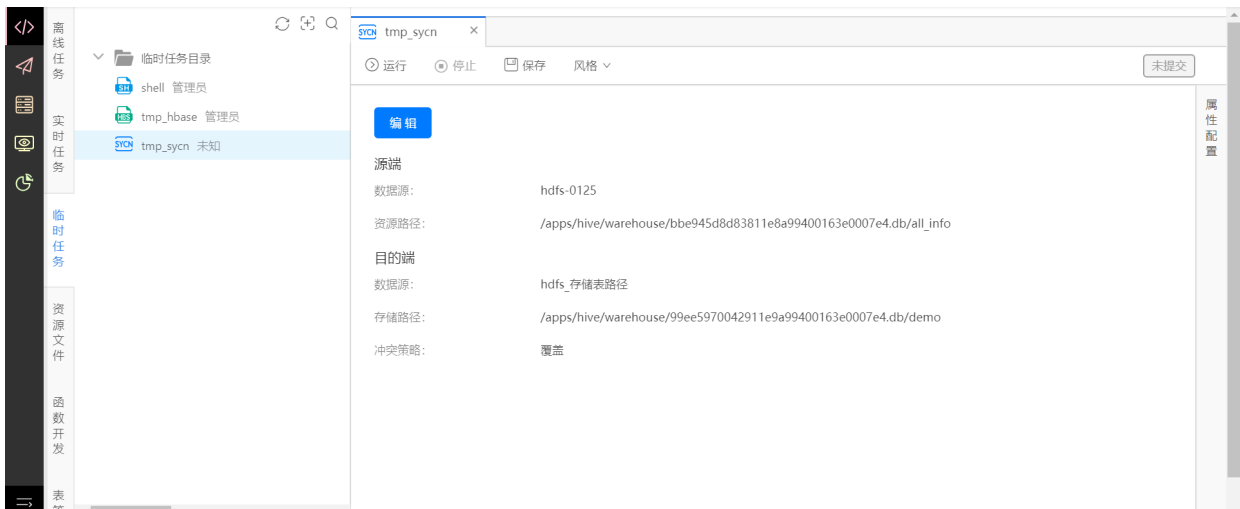
6.11.3 非结构化数据同步

非结构化数据同步任务用于同步文件或文件夹，目的是把文件资源从一个数据源中同步到另一个数据源。

使用非结构化数据同步任务前需要先创建数据源，创建数据源的方式请查看[数据源管理](#)部分。

(新建任务)

创建一个非结构化数据同步任务 `tmp_syncn`，配置源端和目的端信息。



(1) 源端

源端数据源类型支持OSS、FTP和HDFS

选择数据源后，可选填资源路径，当鼠标置于输入框中，显示下一层目录或文件，可在输入框中选择目录或文件。支持模糊搜索，支持通配符*。

点击添加，可增加一行输入框，用于支持批量传输。

通配符*只能用于目的文件夹或文件，不可用于目录中。

(2) 目的端

目的端数据源支持FTP和HDFS。

选择数据源后，可选填存储路径，当鼠标置于输入框中，显示下一层目录。

冲突策略分为覆盖、重命名、跳过、报错。

覆盖：当文件重名时，新文件直接覆盖旧的文件。

重命名：当文件重名时，新文件名称拼接timestamp做为文件名。

跳过：当文件重名时，忽略新文件。

报错：当文件重名时，停止任务并报错。

6.11.4 Python

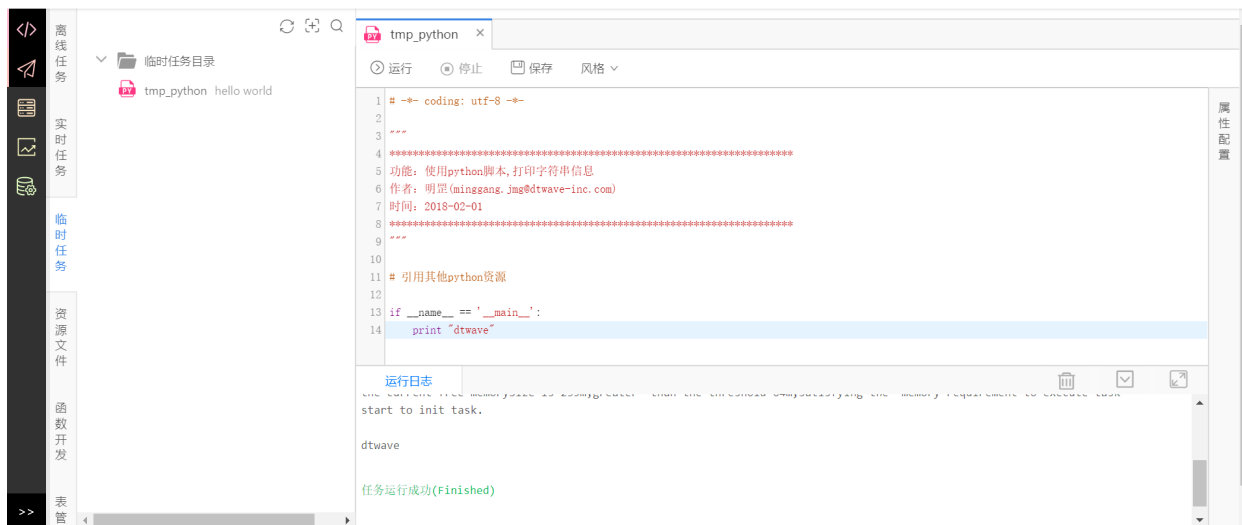
Python类型用于书写Python代码。新建 tmp_python 临时任务，输入以下代码。

```
# -*- coding: utf-8 -*-

"""
*****
功能：使用python脚本,打印字符串信息
作者：明昱
时间：2018-02-01
*****
"""
```

```
# 引用其他python资源
```

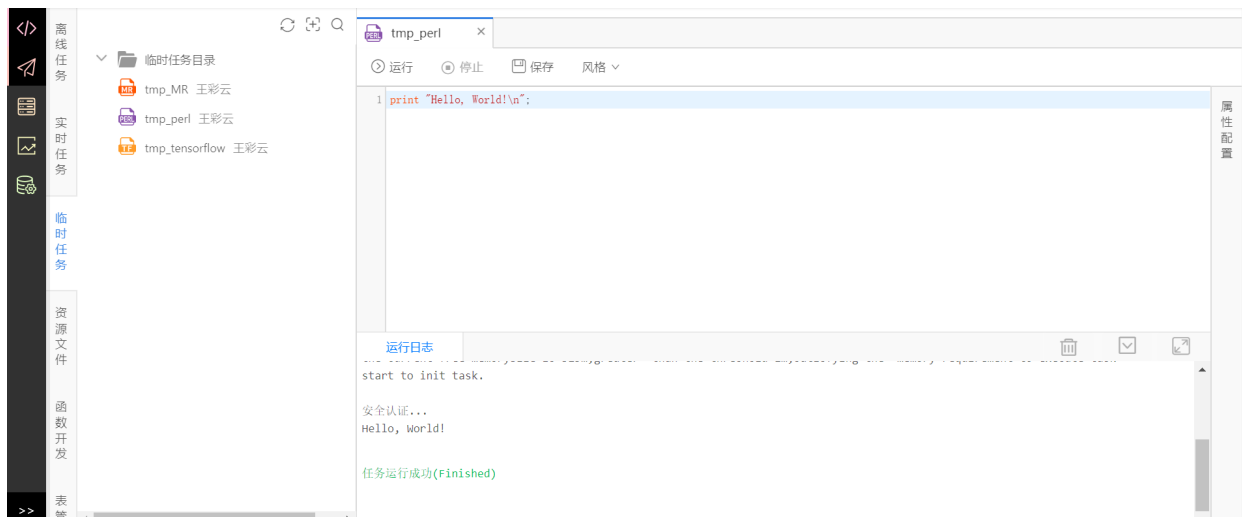
```
if __name__ == '__main__':  
    print "citic"
```



6.11.5 Perl

Perl类型用于书写Perl代码。新建 `tmp_perl` 临时任务，输入以下代码，打印语句。

```
print "Hello, World!\n";
```



6.11.6 Hive2

Hive2类型用于书写Hive SQL。新建 `tmp_hive` 临时任务，输入以下代码。

```
-- 1. 查询用户信息  
select  
    user_id  
    , name  
    , age  
    , address
```

```
from demo_ods_users_info
order by age;
```

-- 2. 查询用户最大年龄

```
select
    max(age)
from demo_ods_users_info;
```

The screenshot shows a web-based IDE interface. On the left, a sidebar lists various tasks under '临时任务' (Temporary Tasks), with 'tmp_hive 演示账号' (tmp_hive demo account) highlighted. The main workspace contains a code editor with the following SQL code:

```
1 -- 1. 查询用户信息
2 select
3     user_id
4     , name
5     , age
6     , address
7 from demo_ods_users_info
8 order by age;
9
10 -- 2. 查询用户最大年龄
11 select
12     max(age)
13 from demo_ods_users_info;
14
```

Below the code editor is a '运行日志' (Execution Log) pane showing the execution progress:

```
正在提交...
waiting...
2018-06-18 23:02:30 调度器收到实例: T_205_20180618230230266_1, 加入到调度队列.
2018-06-18 23:02:30 等待调度中...
2018-06-18 23:02:35 第1次在节点: 10.30.233.11 上获取Yarn资源信息.
2018-06-18 23:02:35 Yarn:10.20.44.126:8080, 队列:dev 的剩余资源是:"VCores: 5, Memory: 8632M".
2018-06-18 23:02:35 获取Yarn资源结束,开始下载到Node上执行.
2018-06-18 23:02:35 第1次调度到节点: 10.30.233.11 成功.

开始执行sql: select
    user_id
    , name
    , age
    , address
from demo_ods_users_info
order by age
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark,
开始执行sql: select
    user_id
    , name
    , age
    , address
from demo_ods_users_info
order by age
```

6.11.7 PySpark

PySpark类型用于书写Python Spark 代码。新建 tmp_PySpark 临时任务，输入下述测试代码。

```
from __future__ import print_function

from pyspark.ml.feature import Word2Vec
from pyspark.sql import SparkSession

# Word2Vec Example
if __name__ == "__main__":
    spark = SparkSession \
        .builder \
        .appName("Word2VecExample") \
        .getOrCreate()

    # $example on$
    # Input data: Each row is a bag of words from a sentence or document.
    documentDF = spark.createDataFrame([
        ("Hi I heard about Spark".split(" ")),
        ("I wish Java could use case classes".split(" ")),
        ("Logistic regression models are neat".split(" ")),
    ], ["text"])
```

```

# Learn a mapping from words to Vectors.
word2Vec = Word2Vec(vectorSize=3, minCount=0, inputCol="text",
outputCol="result")
model = word2Vec.fit(documentDF)

result = model.transform(documentDF)
for row in result.collect():
    text, vector = row
    print("Text: [%s] => \nVector: %s\n" % (" ".join(text),
str(vector)))

spark.stop()

```

The screenshot shows a Databricks workspace interface. On the left, a sidebar lists various tasks and resources, with '临时任务' (Temporary Tasks) highlighted. The main area displays a Python notebook with the following code:

```

1 from __future__ import print_function
2
3 from pyspark.ml.feature import Word2Vec
4 from pyspark.sql import SparkSession
5
6 # Word2Vec Example
7 if __name__ == "__main__":
8     spark = SparkSession \
9         .builder \
10        .appName("Word2VecExample") \
11        .getOrCreate()
12
13 # $example on $
14 # Input data: Each row is a bag of words from a sentence or document.
15 documentDF = spark.createDataFrame([
16     ("Hi I heard about Spark".split(" "),),
17     ("I wish Java could use case classes".split(" "),),
18     ("Logistic regression models are neat".split(" "),)
19 ], ["text"])
20
21 # Learn a mapping from words to Vectors.
22 word2Vec = Word2Vec(vectorSize=3, minCount=0, inputCol="text", outputCol="result")

```

The execution output shows the following results:

```

Text: [I, wish, Java, could, use, case, classes] =>
Vector: [0.0465089769235679,-0.013170594815164804,0.07211566104420593]

Text: [Logistic, regression, models, are, neat] =>
Vector: [0.03593230834230781,-0.02349570980295539,0.01933983787894249]

```

The bottom of the screenshot shows the task execution log, indicating that the task completed successfully.

6.11.8 Spark

Spark类型用于书写Spark代码，语言支持Scala和Java。如果新建任务时没有Spark任务类型，则需要管理员角色可在项目配置中为该项目启用Spark任务。

由于平台暂不支持在线编译代码，因此书写Spark任务的步骤如下：

- 在Idea或者Eclipse中书写如下Scala或Java代码：SparkSqlDemo.scala，并生成jar包：`spark_sql_demo.jar`（在Idea中建议安装PackageJars插件，可直接在package上进行打包）

在libs文件夹中已准备好该jar包：`spark_sql_demo.jar`

```

package com.dtwave.spark.sql

import org.apache.log4j.{Level, Logger}
import org.apache.spark.sql.SparkSession

```

```

/**
 *
 * SparkSQL定义临时函数、操作Hive表
 *
 * 输出Jar包路径: libs/spark_sql_demo.jar
 *
 * @author baisong
 * @date 18/2/1
 */
object SparkSqlDemo {

    //设置日志级别
    Logger.getLogger("org").setLevel(Level.ERROR)

    def main(args: Array[String]): Unit = {

        if (args.length < 2) {
            System.err.println("Usage: SparkSqlDemo <dbName> <tableName>")
            System.exit(-1)
        }
        //库名
        val dbName = args(0)
        // 表名
        val tableName = args(1)

        /**
         * 此处不用设置AppName和Master参数，平台提交作业会自动添加。
         *
         * AppName命名格式：人物名_用户名_实例名
         */
        val spark = SparkSession
            .builder()
            .enableHiveSupport()
            .getOrCreate()

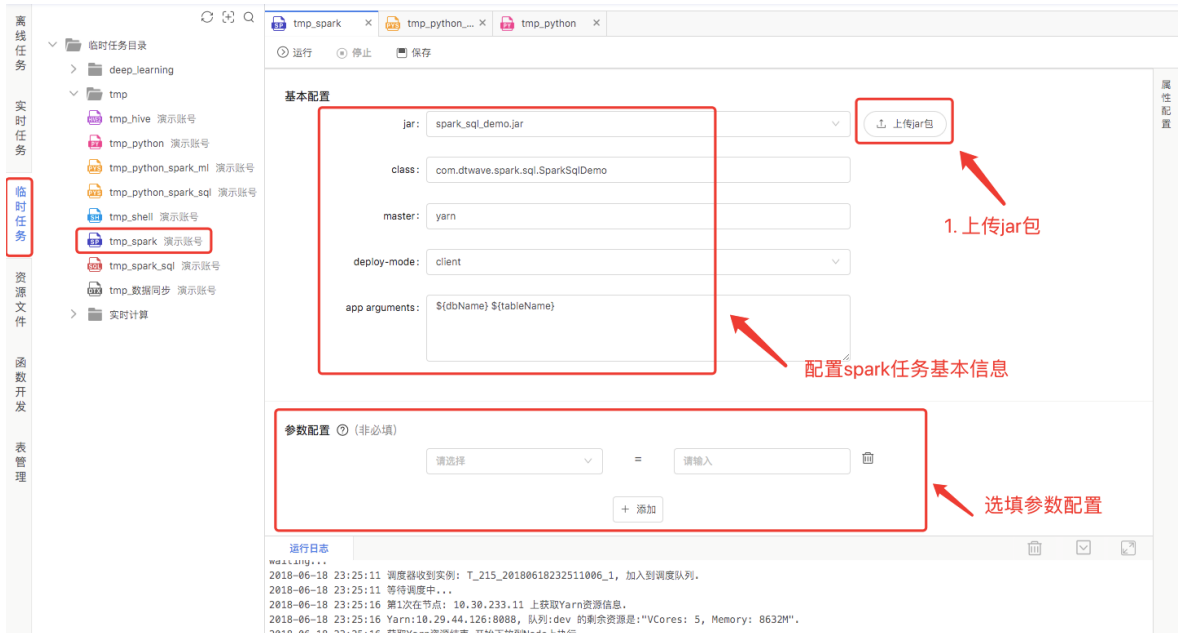
        // 注册临时UDF
        // spark.udf.register("demo_udf", (x: String) => "Hello," + x)

        // 查询Hive表数据
        val df = spark.sql(s"select * from ${dbName}.${tableName} limit 3")

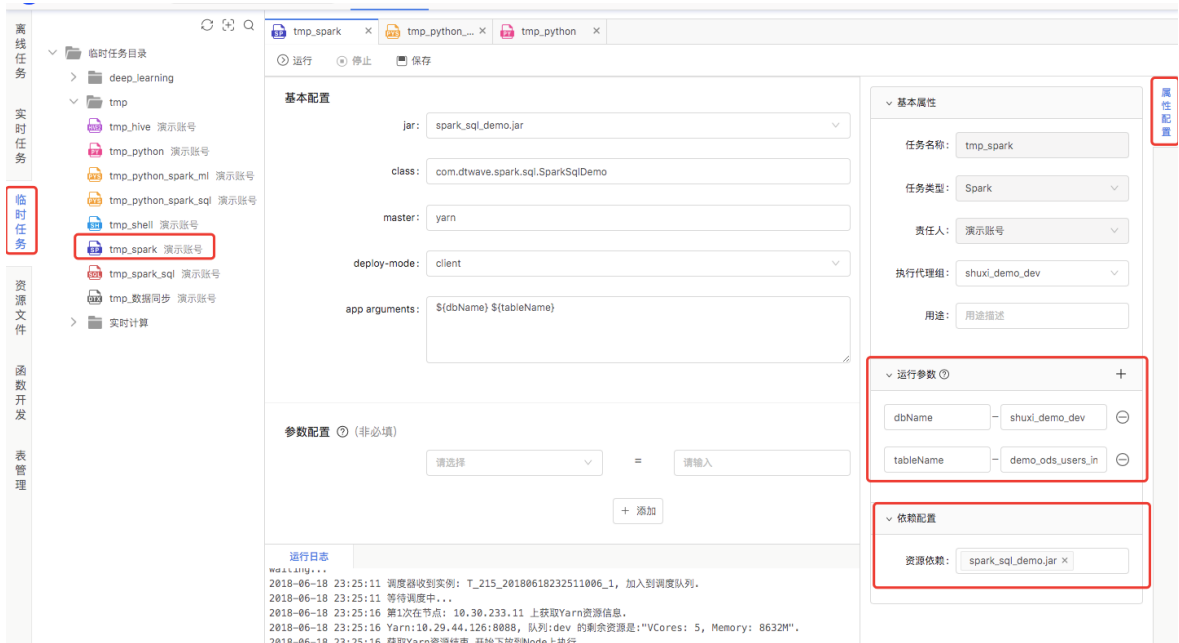
        // 打印输出数据
        df.show(false)
    }
}

```

- 新建Spark类型任务，上传 `spark_sql_demo.jar` 到资源文件，配置好spark任务的基本信息可参数配置：class为 `com.dtwave.spark.sql.SparkSqlDemo`，master为 `yarn`，deploy-mode为 `client`，app arguments为 `${dbName} ${tableName}`



- 配置任务的属性配置信息，包括任务参数和资源依赖。任务参数增加一个dbName，值为当前环境所使用的数据库名 `shuxi_demo_dev`；tableName为 `quick_start_student_info`。



6.11.9 SparkSQL

Spark SQL类型用于书写Spark SQL。新建 `tmp_sparkSQL` 临时任务，输入以下代码。

-- 1. 查询用户信息

```
select
    user_id
  , name
  , age
  , address
from demo_ods_users_info
order by age;
```

-- 2. 查询用户最大年龄

```
select
    max(age)
from demo_ods_users_info;
```

The screenshot shows a web-based interface for managing tasks. On the left, a sidebar lists various task types, with '临时任务' (Temporary Task) highlighted. The main area contains a SQL editor with two queries. The first query is highlighted with a red box. Below the editor, the execution log shows the query completed successfully in 0.755 seconds.

6.11.10 Presto

注意：运行Presto任务需要当前项目空间配置即席引擎，即席引擎配置方法请查看 [资源管理-计算引擎](#) 的帮助手册。

Presto类型用于书写Presto SQL，目前只能用于临时任务。新建 `tmp_presto` 任务，输入以下代码。

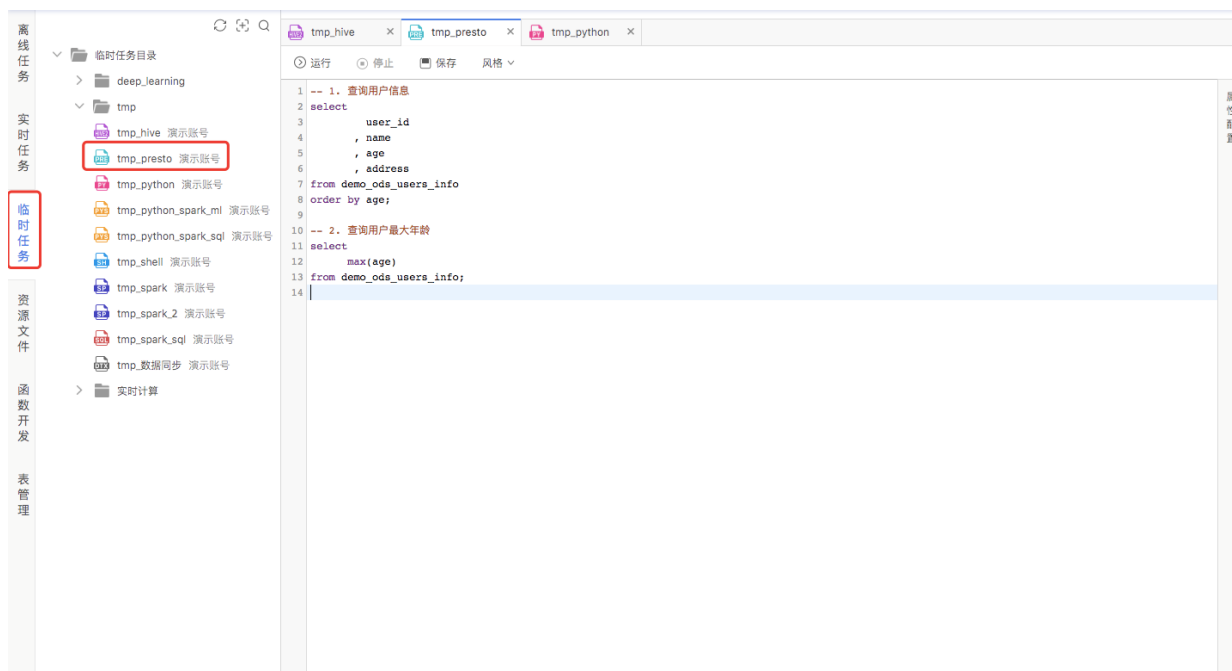
-- 1. 查询用户信息

```
select
    user_id
  , name
  , age
  , address
from demo_ods_users_info
order by age;
```

-- 2. 查询用户最大年龄

```
select
```

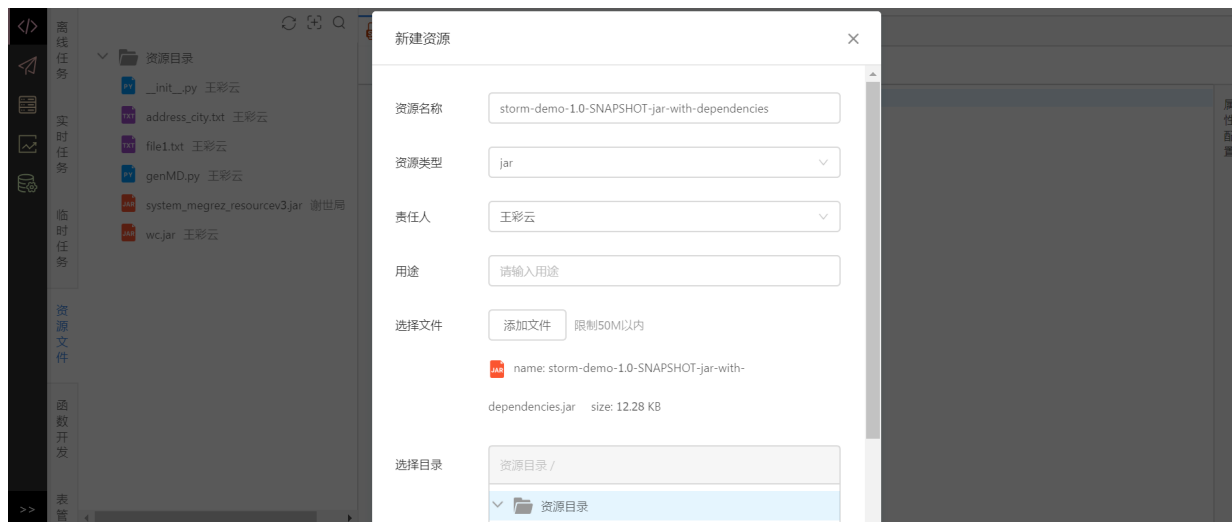
```
max(age)
from demo_ods_users_info;
```



6.11.11 Storm

在libs文件夹中已准备好jar包：`storm-demo-1.0-SNAPSHOT-jar-with-dependencies.jar`

将 `storm-demo-1.0-SNAPSHOT-jar-with-dependencies.jar` 上传到资源文件中。



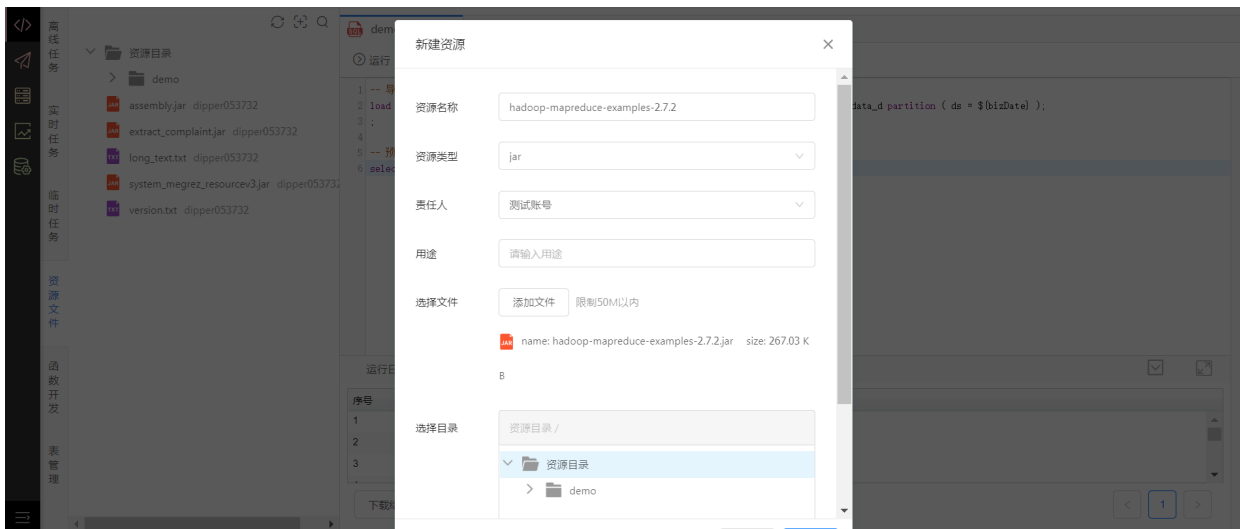
新建storm类型实时任务，输入如下代码并配置好资源依赖：

```
storm jar {storm-demo-1.0-SNAPSHOT-jar-with-dependencies.jar}
storm.starter.WordCountTopology WordCountTopology
```

6.11.12 MapReduce

在libs文件夹中已准备好jar包：`hadoop-mapreduce-examples-2.7.2.jar`

将 `hadoop-mapreduce-examples-2.7.2.jar` 上传至资源文件中。



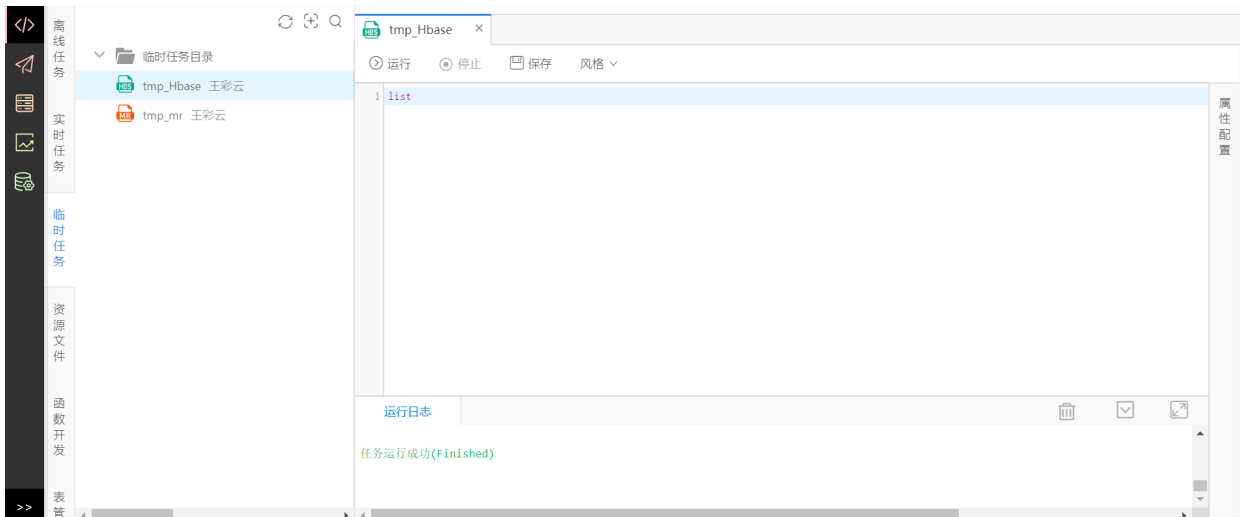
创建MapReduce类型任务用于书写MapReduce的命令，新建 `tmp_mapreduce` 临时任务，输入以下代码并配置资源依赖：

```
hadoop jar {hadoop-mapreduce-examples-2.7.2.jar} pi 10 20
```

6.11.13 Hbase

Hbase类型用于书写Hbase的命令，新建 `tmp_Hbase` 临时任务，输入以下代码查询有哪些表。

```
list
```



6.11.14 MySQL

MySQL类型用于书写MySQL。新建 `tmp_mysql` 临时任务，输入以下代码查看当前库下的表。

```
SHOW TABLES;
```

6.11.15 Oracle

Oracle类型用于书写MySQL。新建 `tmp_oracle` 临时任务，输入以下代码查看当前库下的表。

```
SHOW TABLES;
```

6.11.16 GreenPlum

GreenPlum类型用于书写MySQL。新建 `tmp_gp` 临时任务，输入以下代码查看当前库下的表。

```
SHOW TABLES;
```

6.11.17 SQL Server

SQL Server类型用于书写SQL Server。新建 `tmp_sqlserver` 临时任务，输入以下代码查看当前库下的表。

支持在GreenPlum项目下创建SQL Server任务

```
SHOW TABLES;
```

6.11.18 PPAS

PPAS类型用于书写PPAS。新建 `tmp_ppas` 临时任务，输入以下代码查看当前库下的表。

支持在GreenPlum项目下创建PPAS任务

```
SHOW TABLES;
```

6.11.19 Tensorflow

Tensorflow类型用于书写TensorFlow代码，代码如下。

```
# -*- coding: UTF-8 -*-

import tensorflow as tf
import numpy as np

# 样本，输入列表，正太分布(Normal Distribution)，均值为1，均方差为0.1，数据量为
100个
x_vals = np.random.normal(1, 0.1, 100)
# 样本输出列表， 100个值为10.0的列表
y_vals = np.repeat(10.0, 100)

x_data = tf.placeholder(shape=[1], dtype=tf.float32)
y_target = tf.placeholder(shape=[1], dtype= tf.float32)

A = tf.Variable(tf.random_normal(shape=[1]))

# 我们定义的模型，是一个线型函数，即  $y = w * x$ ，也就是my_output = A * x_data
# x_data将用样本x_vals。我们的目标是，算出A的值。
# 其实已经能猜出，y都是10.0的话，x均值为1，那么A应该是10。哈哈
my_output = tf.multiply(x_data, A)
```

```

# 损失函数，用的是模型算的值，减去实际值，的平方。y_target就是上面的y_vals。
loss = tf.square(my_output - y_target)

sess = tf.Session()
init = tf.global_variables_initializer()#初始化变量
sess.run(init)

# 梯度下降算法，学习率0.02，可以认为每次迭代修改A，修改一次0.02。比如A初始化为20，发现不好，于是猜测下一个A为20-0.02
my_opt = tf.train.GradientDescentOptimizer(learning_rate=0.02)
train_step = my_opt.minimize(loss)#目标，使得损失函数达到最小值

for i in range(100):#0到100,不包括100
    # 随机从样本中取值
    rand_index = np.random.choice(100)
    rand_x = [x_vals[rand_index]]
    rand_y = [y_vals[rand_index]]
    #损失函数引用的placeholder(直接或间接用的都算)，x_data使用样本rand_x,
    y_target用样本rand_y
    sess.run(train_step, feed_dict={x_data: rand_x, y_target: rand_y})
    #打印
    if i%5==0:
        print('step: ' + str(i) + ' A = ' + str(sess.run(A)))
        print('loss: ' + str(sess.run(loss, feed_dict={x_data: rand_x,
y_target: rand_y})))

```

The screenshot shows a Jupyter Notebook environment. The code in the cell is as follows:

```

1 # -*- coding: UTF-8 -*-
2
3 import tensorflow as tf
4 import numpy as np
5
6 # 样本，输入列表，正态分布(Normal Distribution)，均值为1，均方差为0.1，数据量为100个
7 x_vals = np.random.normal(1, 0.1, 100)
8 # 样本输出列表，100个值为10.0的列表
9 y_vals = np.repeat(10.0, 100)
10
11 x_data = tf.placeholder(shape=[1], dtype=tf.float32)
12 y_target = tf.placeholder(shape=[1], dtype=tf.float32)
13
14 A = tf.Variable(tf.random_normal(shape=[1]))
15
16 # 我们定义的模型，是一个线性函数，即 y = w * x，也就是my_output = A * x_data
17 # x_data将用样本x_vals，我们的目标是，算出A的值。
18
19

```

The execution output (运行日志) is:

```

loss: [1.5007801]
step: 90 A = [9.765769]
loss: [0.25166214]
step: 95 A = [9.93642]
loss: [0.00707662]

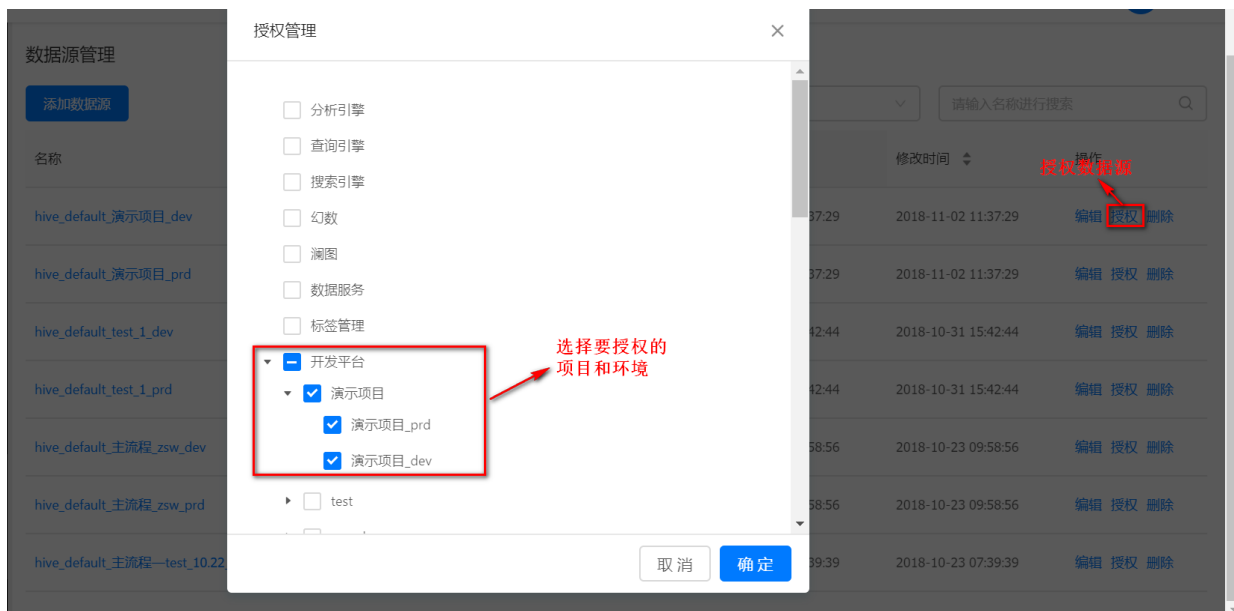
```

The notebook status is "任务运行成功(Finished)".

6.11.20 FlinkSQL

FlinkSQL类型用于书写Flink SQL，直接使用SQL来描述流计算的逻辑，并在任务属性配置中配置输入及输出后，即可完成一个流计算任务的开发。

(1) 首先需要管理员在数据源管理中为该项目授权两个数据源或在项目配置中直接创建数据源，以一个Kafka数据源和一个MySQL数据源为例，创建数据源的方式请查看数据源管理部分。

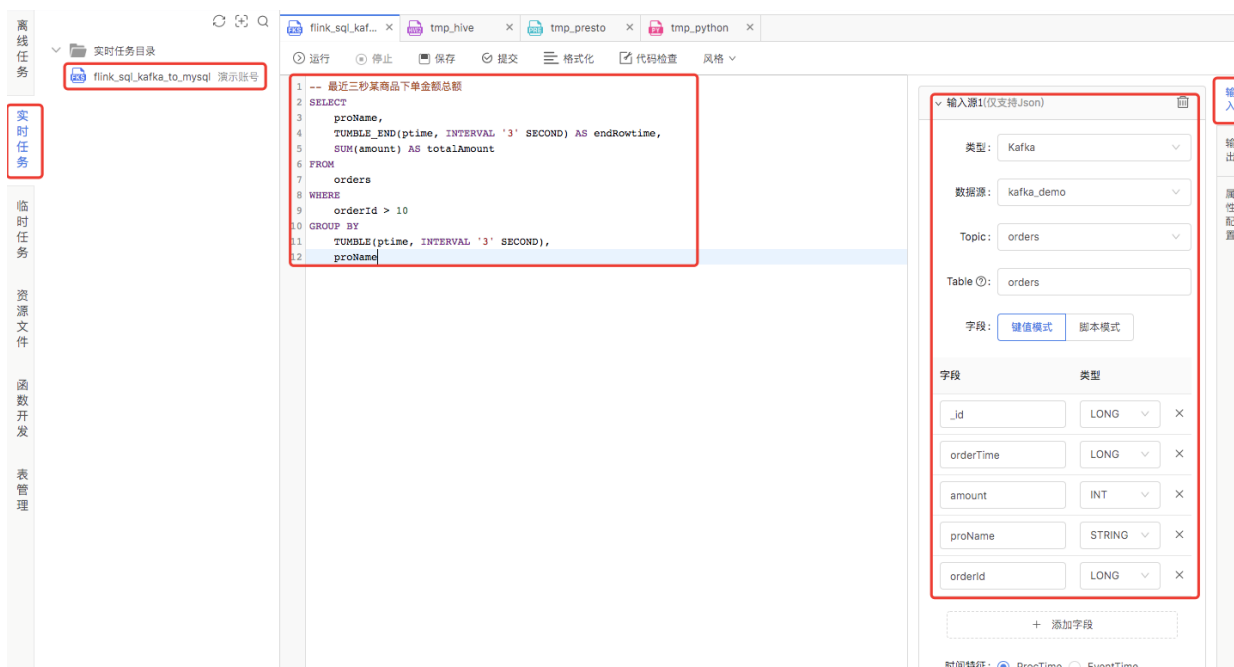


(2) 在实时任务中新建一个FlinkSQL类型的任务，输入以下代码。

-- 最近三秒某商品下单金额总额

```

SELECT
    proName,
    TUMBLE_END(pTime, INTERVAL '3' SECOND) AS endRowtime,
    SUM(amount) AS totalAmount
FROM
    orders
WHERE
    orderId > 10
GROUP BY
    TUMBLE(pTime, INTERVAL '3' SECOND),
    proName
  
```



(3) 配置输入源信息

1 -- 最近三秒某商品下单金额总额
2 SELECT
3 proName,
4 TUMBLE_END(ptime, INTERVAL '3' SECOND) AS endRowtime,
5 SUM(amount) AS totalAmount
6 FROM
7 orders
8 WHERE
9 orderId > 10
10 GROUP BY
11 TUMBLE(ptime, INTERVAL '3' SECOND),
12 proName

输入源1(仅支持Json)

类型: Kafka

数据源: kafka_demo

Topic: orders

Table: orders

字段: 键值模式 脚本模式

字段	类型
_id	LONG
orderTime	LONG
amount	INT
proName	STRING
orderid	LONG

时间特征: ProcTime EventTime

别名: ptime

(4) 配置输出源信息

1 -- 最近三秒某商品下单金额总额
2 SELECT
3 proName,
4 TUMBLE_END(ptime, INTERVAL '3' SECOND) AS endRowtime,
5 SUM(amount) AS totalAmount
6 FROM
7 orders
8 WHERE
9 orderId > 10
10 GROUP BY
11 TUMBLE(ptime, INTERVAL '3' SECOND),
12 proName

输出源1

存储类型: RDS

数据源: mysql_demo

表: demo_kafka_output

(5) 配置属性信息

至此，FlinkSQL任务新建完成，但输入源Kafka中仍无数据。下面步骤提供两种方法来模拟Kafka数据生成。

- 模拟kafka数据生成

本文档已提供Kafka发送消息参考代码：`KafkaDataSimulation.java`。

```
package com.dtwave.flink;

import com.fasterxml.jackson.databind.ObjectMapper;
import org.apache.kafka.clients.producer.Callback;
import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerRecord;
import org.apache.kafka.clients.producer.RecordMetadata;

import java.io.Serializable;
import java.util.Properties;

/**
 * Kafka模拟数据生成
 * <p>
 * 数据格式：
 *
 * {"_id":1,"orderId":18,"proName":"prt3","amount":1,"orderTime":1512974707597}
 *
 * {"_id":2,"orderId":19,"proName":"prt4","amount":1,"orderTime":1512974708601}
 *
 * {"_id":3,"orderId":20,"proName":"prt0","amount":1,"orderTime":1512974709607}
 *
 */
```

```

* java -cp target/shuxi-0.0.1-jar-with-dependencies.jar
com.dtwave.flink.KafkaDataSimulation mq250:9092,mq221:9092,mq164:9092
*
* @author hulb
* @date 2017/12/11 下午2:43
*/
public class KafkaDataSimulation {

    public static void main(String[] args) throws Exception {

        if (args.length < 1) {
            System.err.println("Usage: KafkaDataSimulation
<bootstrap.servers>\n" +
                "Example : mq1:9092,mq2:9092,mq3:9092");
            System.exit(-1);
        }
        Integer messageCount = 10000;
        Properties props = new Properties();
        props.put("bootstrap.servers", args[0]);
        props.put("acks", "all");
        props.put("retries", 0);
        props.put("batch.size", 16384);
        props.put("linger.ms", 1);
        props.put("request.timeout.ms", 45000);
        props.put("buffer.memory", 33554432);
        props.put("compression.type", "snappy");
        props.put("key.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
        props.put("value.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
        KafkaProducer<String, String> producer = new KafkaProducer<>
(props);

        for (int i = 1; i <= messageCount; i++) {
            Order order = new Order();
            order.set_id(Integer.toUnsignedLong(i));
            order.setOrderId(Integer.toUnsignedLong(i));
            order.setProName("prt_" + Math.floorMod(i, 5));
            order.setAmount(1);
            long currentTime = System.currentTimeMillis();
            order.setOrderTime(currentTime);
            String adsContent = new
ObjectMapper().writeValueAsString(order);
            System.out.println(adsContent);
            ProducerRecord<String, String> adsProduceRecord = new
ProducerRecord<>("orders", adsContent);
            producer.send(adsProduceRecord, new Callback() {
                @Override

```

```

        public void onCompletion(RecordMetadata recordMetadata,
Exception e) {
            if (e != null) {
                System.err.println("failed to send the record,"
+ e.getMessage());
            } else {
                System.out.println("send successfully");
            }
        }
    });
    Thread.sleep(1000L);
}
}

class Order implements Serializable {
    private Long _id;
    private Long orderId;
    private String proName;
    private Integer amount;
    private Long orderTime;

    public Long get_id() {
        return _id;
    }

    public void set_id(Long _id) {
        this._id = _id;
    }

    public Long getOrderId() {
        return orderId;
    }

    public void setOrderId(Long orderId) {
        this.orderId = orderId;
    }

    public String getProName() {
        return proName;
    }

    public void setProName(String proName) {
        this.proName = proName;
    }

    public Integer getAmount() {
        return amount;
    }
}

```



```

    }

    public void setAmount(Integer amount) {
        this.amount = amount;
    }

    public Long getOrderTime() {
        return orderTime;
    }

    public void setOrderTime(Long orderTime) {
        this.orderTime = orderTime;
    }
}

```

○ 方法一

在libs文件夹中已准备好jar包 `flink_demo.jar`。

把flink_demo.jar上传到资源文件中。

新建一个Shell任务，并输入如下代码，配置资源依赖：

`flink_demo.jar`，`kafka_param` 参数是Kafka的服务器地址，例如 "mq1:9092,mq2:9092,mq3:909"。点击运行后，便会启动kafka消息生成器往orders这个Topic持续发送消息。

```

java -cp {flink_demo.jar} com.dtwave.flink.KafkaDataSimulation
${kafka_param}

```

○ 方法二

下载源码后,修改pom依赖provided级别为compile,设置程序入参后,直接运行KafkaDataSimulation类即可，成功运行会打印如下输出：

```

{"_id":1,"orderId":1,"proName":"prt_1","amount":1,"orderTime":1519
975807938}
send successfully

```

- 启动消息发送程序后，即可开始运行Flink SQL任务。

6.11.21 Flink

如果新建任务时没有Flink任务类型，则需要管理员角色可在项目配置中为该项目启用Flink任务。

- 本文档提供测试代码 `FlinkDemo.scala`，在Idea或者Eclipse中书写Scala或Java代码并打成Jar包：`flink_demo.jar`。

在libs文件中已准备好该jar包：`flink_demo.jar`。

```

package com.dtwave.flink

```

```

import java.util.Properties

import org.apache.flink.api.common.typeinfo.TypeInformation
import org.apache.flink.streaming.api.scala._
import org.apache.flink.streaming.connectors.kafka.Kafka010JsonTableSource
import org.apache.flink.table.api.{Table, TableEnvironment, Types}

/**
 * FlinkDemo 演示程序。
 * 消费kafka orders topic的内容。
 * 输出到控制台。
 */
object FlinkDemo {
  def main(args: Array[String]): Unit = {

    if (args.length < 1) {
      System.err.println("Usage: FlinkDemo <bootstrap.servers>\n" +
        "Example : mq1:9092,mq2:9092,mq3:9092")
      System.exit(-1)
    }

    val env=StreamExecutionEnvironment.getExecutionEnvironment
    val tableEnv=TableEnvironment.getTableEnvironment(env)

    //设置检查点
    env.enableCheckpointing(5000L)
    //与kafka集成
    val props = new Properties()
    props.setProperty("bootstrap.servers", args.apply(0))
    props.setProperty("group.id", "flink-group")

    val typeInfo=Types.ROW(
      Array[String]("_id","orderId","proName","amount","orderTime"),
      Array[TypeInformation[_]]
    (Types.LONG,Types.LONG,Types.STRING,Types.INT,Types.LONG)
    )
    //kafka source 选择消费orders这个topic
    val kafkaSource=new Kafka010JsonTableSource("orders",props,typeInfo)
    //注册为表
    tableEnv.registerTableSource("orders",kafkaSource)
    //appended table
    val appendedTable:Table=tableEnv.sql("select orderId, proName, amount
from orders")
    val appendedStream=tableEnv.toAppendStream[(Long,String,Int)]
    (appendedTable)

    //输出到标准输出
    appendedStream.print()
  }
}

```

```
//开始执行
env.execute("orders in sql")
}
case class Order(orderId:Int,proName:String,amount:Int)
}
```

- 将 `flink_demo.jar` 上传到资源文件中，并在实时任务中新建Flink类型的实时任务，输入以下代码，配置资源依赖为 `flink_demo.jar`，运行参数新增 `kafka_param`，值为Kafka的服务器地址，例如 "mq1:9092,mq2:9092,mq3:909"

```
flink run -m yarn-cluster -yn 2 -c com.dtwave.flink.FlinkDemo
{flink_demo.jar} ${kafka_param}
```

- 启动Kafka消息发送程序后，即可启动Flink任务。

6.11.22 实时数据同步

实时数据同步任务用于实时数据同步或实时数据交换，目的是把实时数据从一个数据源中同步到另一个数据源。

使用数据同步任务前需要先创建Kafka数据源、Topic和Schema，创建数据源和Topic的方式请查看[5.3 Kafka数据源](#)部分，创建schema的方式请查看[10.7 Topic](#)管理部分

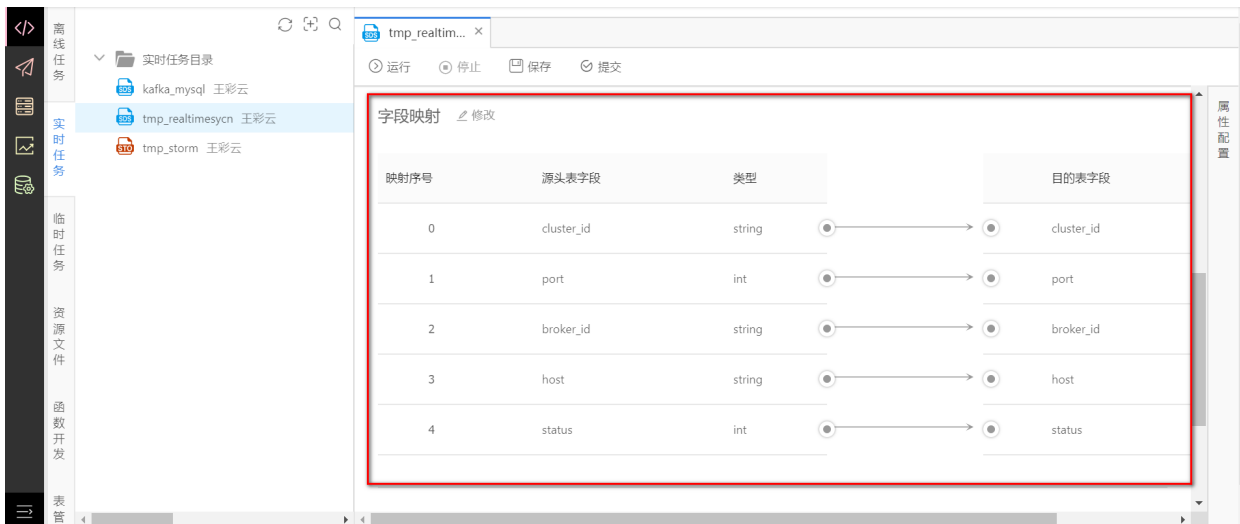
(1) 数据源配置

首先创建一个实时数据同步任务 `tmp_realtimesyncn`，配置源头表信息和目的表信息。



(2) 字段映射

配置源头表和目的表的字段映射关系，映射序号为连线的顺序。



6.11.23 Spark Streaming

Spark Streaming类型用于书写Spark Streaming代码。新建 `tmp_sparkstreaming` 临时任务，输入下述测试代码。

```
# -*- coding: utf-8 -*-
from __future__ import print_function
import sys
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

if __name__ == "__main__":

    sc = SparkContext(appName="PythonStreamingNetworkWordCount")
    ssc = StreamingContext(sc, 1)

    lines = ssc.socketTextStream("localhost", int(9999))
    counts = lines.flatMap(lambda line: line.split(" "))\
        .map(lambda word: (word, 1))\
        .reduceByKey(lambda a, b: a+b)
    counts.pprint()

    ssc.start()
    ssc.awaitTermination()
```

```

1 # -*- coding: utf-8 -*-
2 from __future__ import print_function
3 import sys
4 from pyspark import SparkContext
5 from pyspark.streaming import StreamingContext
6
7 if __name__ == "__main__":
8
9     sc = SparkContext(appName="PythonStreamingNetworkWordCount")
10    ssc = StreamingContext(sc, 1)
11
12    lines = ssc.socketTextStream("localhost", int(9999))
13    counts = lines.flatMap(lambda line: line.split(" "))\
14                  .map(lambda word: (word, 1))\
15                  .reduceByKey(lambda a, b: a+b)
16    counts.pprint()
17

```

运行日志

```

D.scala:48), which has no missing parents
19/01/05 17:54:50 INFO SparkContext: Created broadcast 135 from broadcast at DAGScheduler.scala:1012
19/01/05 17:54:50 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 268 (PythonRDD[530] at RDD at PythonRDD.scala:48)
19/01/05 17:54:50 INFO YarnScheduler: Adding task set 268.0 with 1 tasks
19/01/05 17:54:50 INFO TaskSetManager: Starting task 0.0 in stage 268.0 (TID 203, node37, partition 0, PROCESS_LOCAL, 5098 bytes)
19/01/05 17:54:50 INFO MapOutputTrackerMasterEndpoint: Asked to send map output locations for shuffle 67 to 10.168.229.37:47578
19/01/05 17:54:50 INFO MapOutputTrackerMaster: Size of output statuses for shuffle 67 is 83 bytes
19/01/05 17:54:50 INFO TaskSetManager: Finished task 0.0 in stage 268.0 (TID 203) in 114 ms on node37 (1/1)

```

7 发布管理

7.1 进入发布管理

当任务提交完成后，点击【发布管理】图标进入发布管理页面，如图所示。

创建发布包

发布管理

对象类型 全部 所属工作流 全部 责任人 全部

修改时间 请选择日期 对象名称 输入对象名称进行查询

刷新

<input type="checkbox"/>	对象名称	工作流	对象类型	版本号	变更类型	责任人	修改时间	操作
<input type="checkbox"/>	demo_dim_address_city	默认工作流	离线任务	2	未删除	小祝	2018-11-20 10:02:32	查看 添加 删除
<input type="checkbox"/>	ddl_demo_dim_address_city	默认工作流	离线任务	2	未删除	小祝	2018-11-20 10:02:27	查看 添加 删除
<input type="checkbox"/>	demo_adm_history_complaint_d	默认工作流	离线任务	2	未删除	小祝	2018-11-20 10:02:22	查看 添加 删除
<input type="checkbox"/>	ddl_demo_adm_hist	默认工作流	离线任务	2	未删除	小祝	2018-11-20 10:02:18	查看 添加 删除

待发布对象 0 个

请在左侧添加要发布的对象

创建发布包

7.2 创建发布包

在创建发布包列表页中勾选要发布的任务、资源和函数后，点击【创建发布包】，填入发布包名称和描述，点击确定提交发布包。

创建发布包

对象类型 全部

修改时间 请选择日期

刷新

对象名称 工作流

address_city.txt

spark_sql_demo_lrb.jar

spark_sql_demo.jar

共 3 条 1 10 条/页 页

待发布对象 2 个

address_city.txt 类型: 资源

spark_sql_demo_lrb.jar 类型: 资源

创建发布包

创建发布包

发布包名称 test_demo

发布描述 test

责任人 管理员

发布到项目 主流程_zsw_prd

填写发布包名称和描述

取消 确定

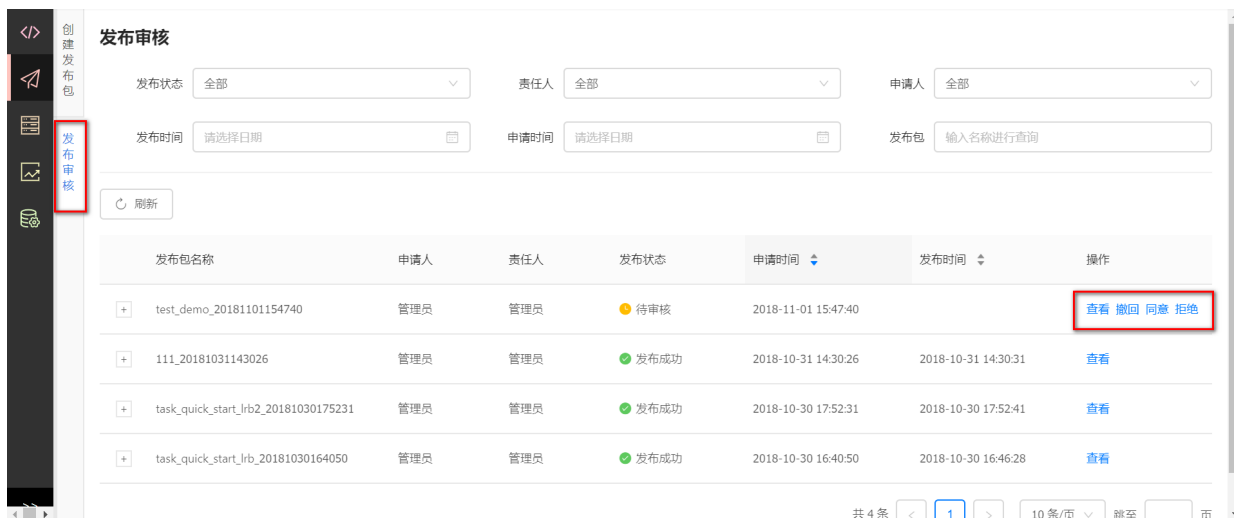
7.3 审核发布包

只有开发平台管理员、项目的管理员和运维角色才有权限审核发布包。

开发人员提交的发布包，需要管理员或者运维人员审核通过后才能发布到生产环境。

在发布审核列表页中，管理员或运维人员点击**同意后**，发布包中任务、资源、函数将会发布到生产环境，其中任务将按周期执行。点击**拒绝**发布以后，任务会再次回到**创建发布包**页面下。

开发人员可**撤回**发布包，点击**撤回**后，任务或资源会再次回到**创建发布包**页面下。



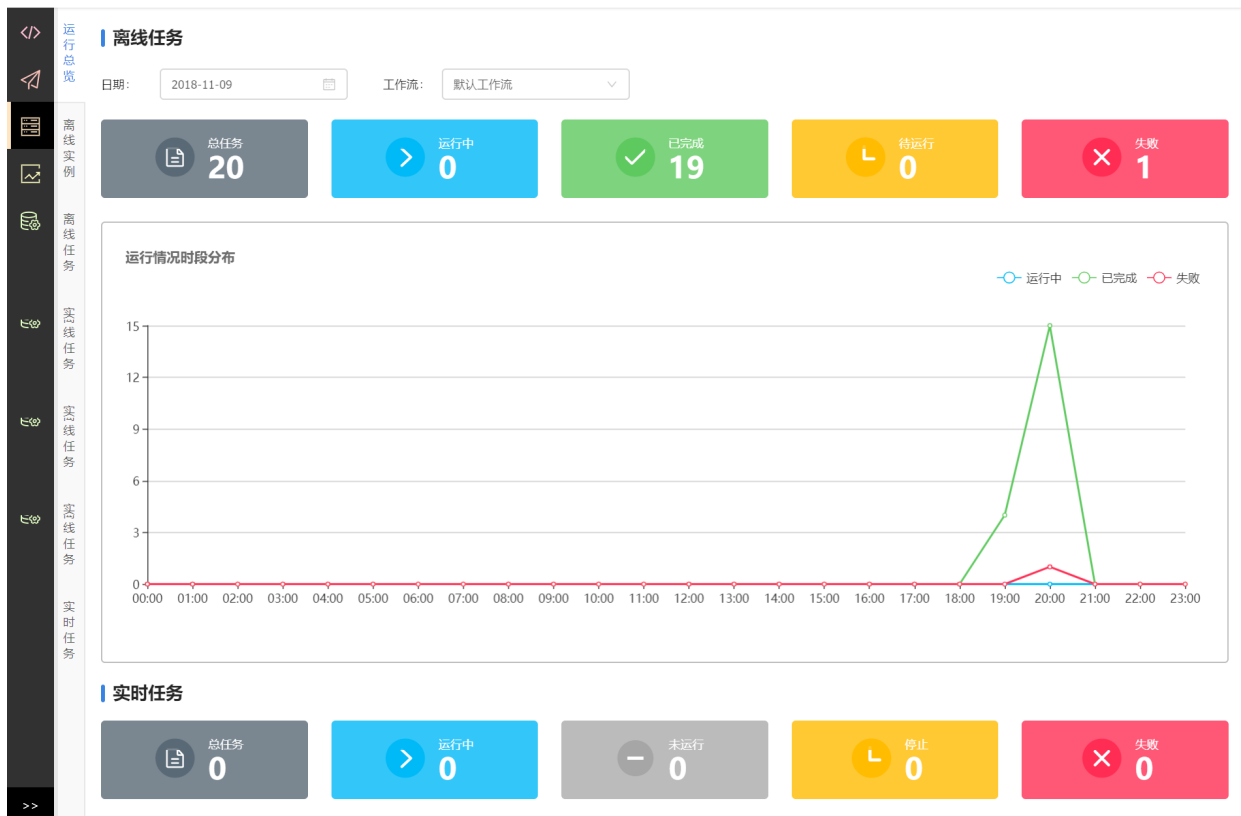
The screenshot displays the '发布审核' (Release Review) interface. It includes a sidebar with navigation options like '创建发布包' and '发布审核'. The main area features search filters for '发布状态' (Release Status), '责任人' (Responsible Person), and '申请人' (Applicant), along with date pickers for '发布时间' (Release Time) and '申请时间' (Application Time). A table lists release packages with columns: '发布包名称' (Release Package Name), '申请人' (Applicant), '责任人' (Responsible Person), '发布状态' (Release Status), '申请时间' (Application Time), '发布时间' (Release Time), and '操作' (Action). The first row shows a package with ID 'test_demo_20181101154740' in '待审核' (Pending Review) status, with a '查看' (View) button highlighted in red. The bottom of the page shows pagination information: '共 4 条' (Total 4 items), page '1', and '10 条/页' (10 items per page).

8 运维中心

进入开发后，点击【运维中心】图标，可进入运维中心。运维中心可查看 workflow 和任务的运行总览、每个实例的运行状态和日志，还有任务及 workflow 之间的依赖关系。同时对于生产环境下的任务提供重跑、重跑下游、置成功、补数据、补下游等操作。运维中心模块一般用于生产环境中。

8.1 运行总览

进入运维中心后，默认显示运行总览页面，**运行总览**页面可查看每天离线任务和实时任务的运行统计信息，离线任务的运行统计如下图所示。点击某个状态可自动跳转到相应的工作流或离线实例中。



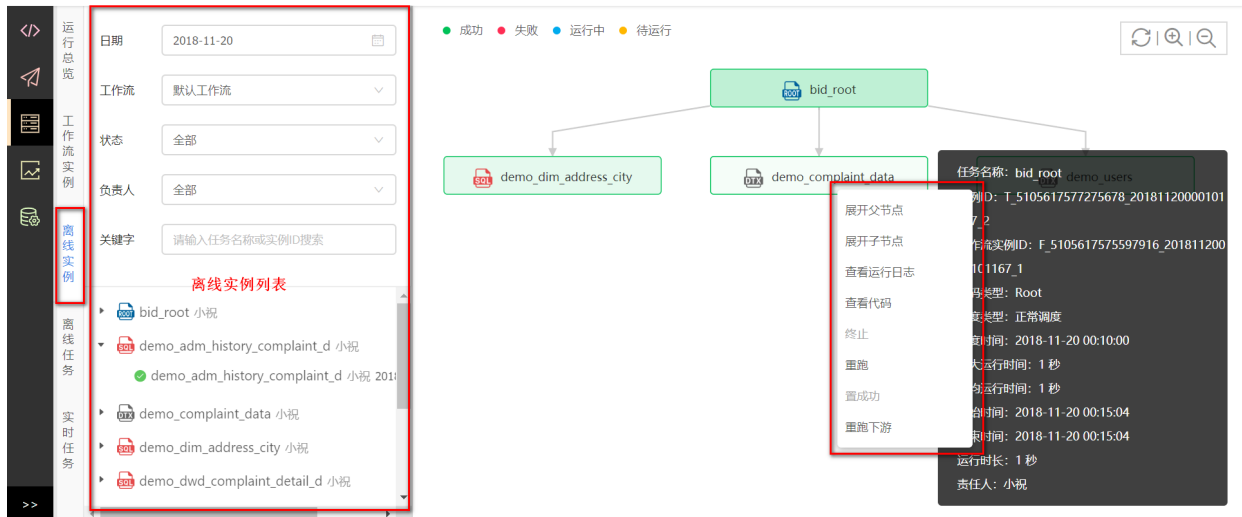
8.2 工作流实例

工作流实例页面分为两块内容，左侧是工作流实例列表，可以根据日期、状态、负责人、关键字进行筛选；右侧是实例信息，包含实例上下游、运行状态、基本信息等。右击任务实例，可以看到实例操作列表，包含: 展开父节点、展开子节点、终止、重跑、重跑下游、置成功。点击工作流实例，可直接跳转到该工作流的离线实例。



8.3 离线实例

离线实例页面分为两块内容，左侧是实例列表，可以根据日期、状态、负责人、名称进行筛选；右侧是实例信息，包含实例上下游、运行状态、基本信息等。右击任务实例，可以看到实例操作列表，包含: 展开父节点、展开子节点、查看运行日志、查看代码、终止、重跑、重跑下游、置成功。



- 展开父节点

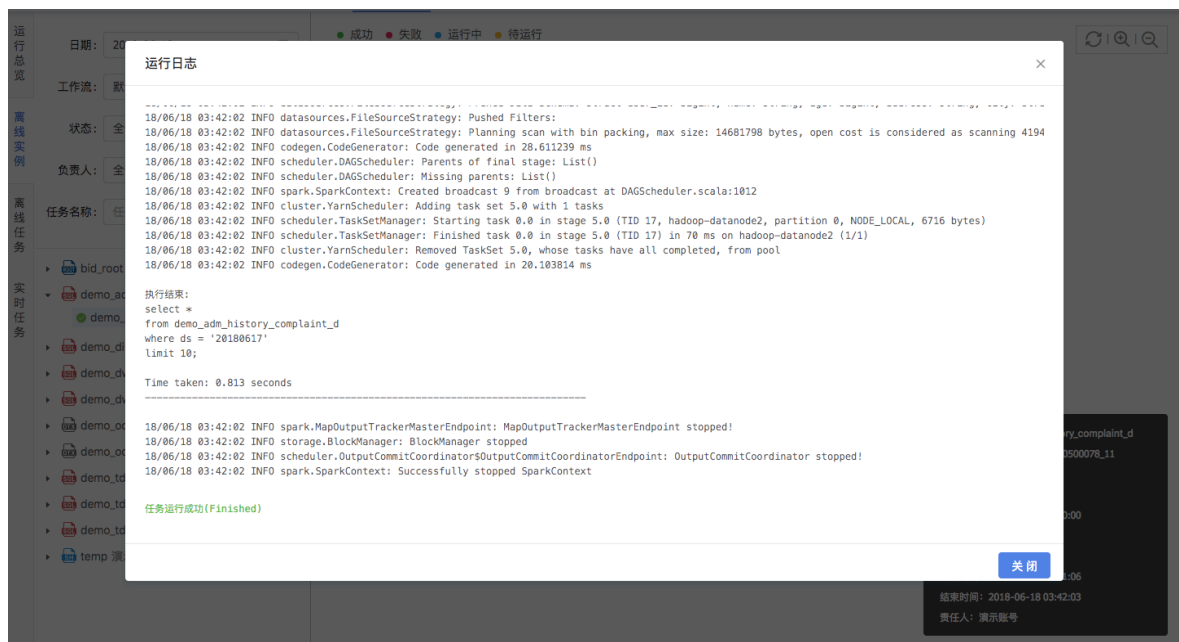
点击任务实例，右击可以看到功能操作列表，点击展开父节点，在实例上方即可看到任务实例父节点信息。

- 展开子节点

点击任务实例，右击可以看到功能操作列表，点击展开子节点，在实例下方即可看到任务实例子节点信息。

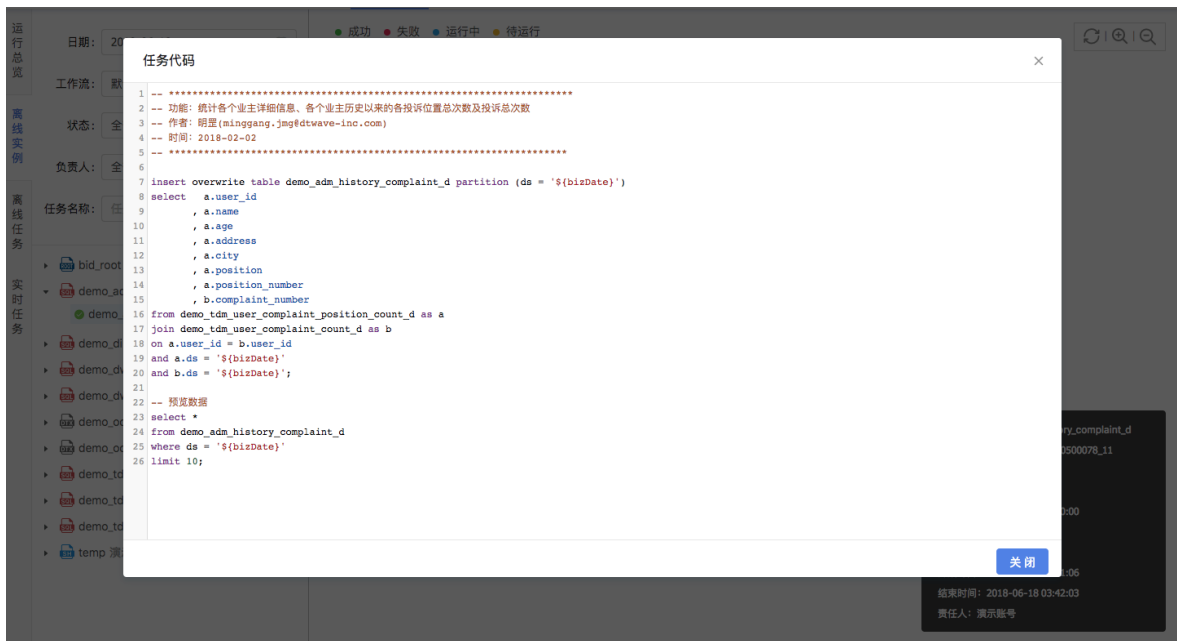
- 查看运行日志

点击任务实例，右击可以看到功能操作列表，点击查看运行日志，弹出运行日志窗口，查看实例运行日志。



- 查看代码

点击任务实例，右击可以看到功能操作列表，点击查看代码，弹出任务代码窗口，查看实例运行真实代码。任务代码页如下图所示。



- 终止

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。如果有下游实例，当前实例被终止后，下游所有实例都将会被停止，状态是STOPPED。

- 重跑

重跑即实例重新运行一遍，不会再次进行实例化。当任务状态处于成功、失败时可以进行重跑。

- 重跑下游

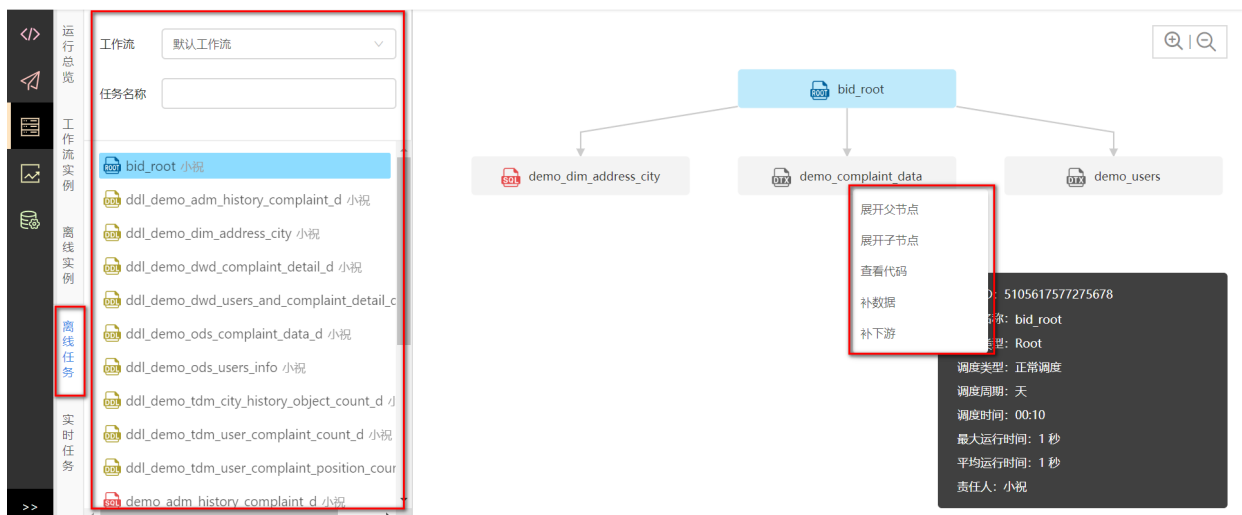
重跑下游会把实例和所有下游实例节点都重新运行一遍。当任务状态处于成功、失败时，可以进行重跑下游。

- 置成功

将当前节点状态改为成功，并重跑下游所有的实例。当任务状态处于失败时，可以进行置成功操作。

8.4 离线任务

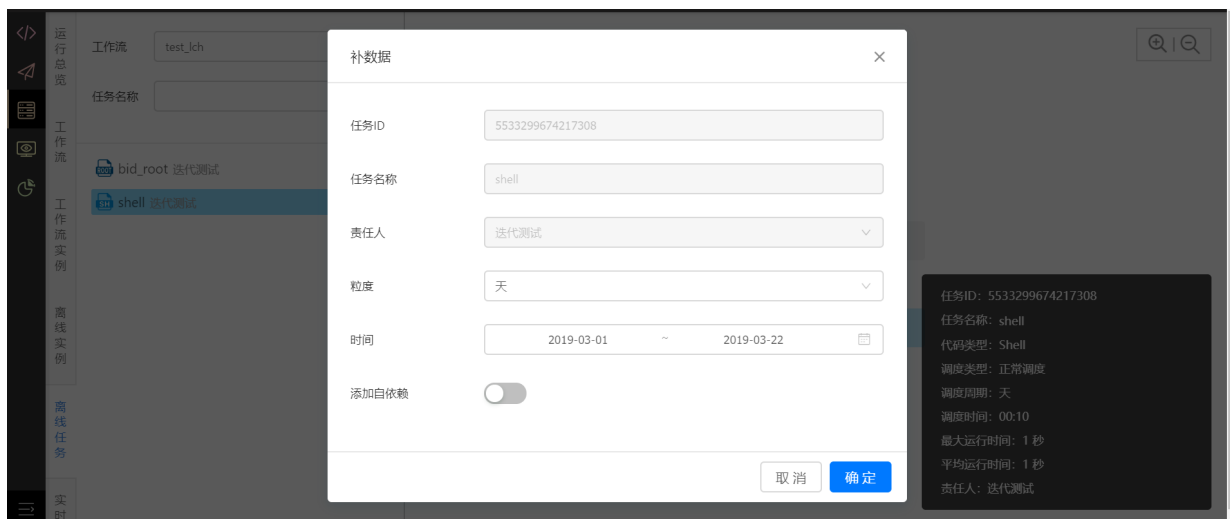
离线任务页面分为两块内容，左侧为离线任务列表，可以根据任务名称进行筛选；右侧为具体任务信息，包含任务上下游依赖、基本信息等。



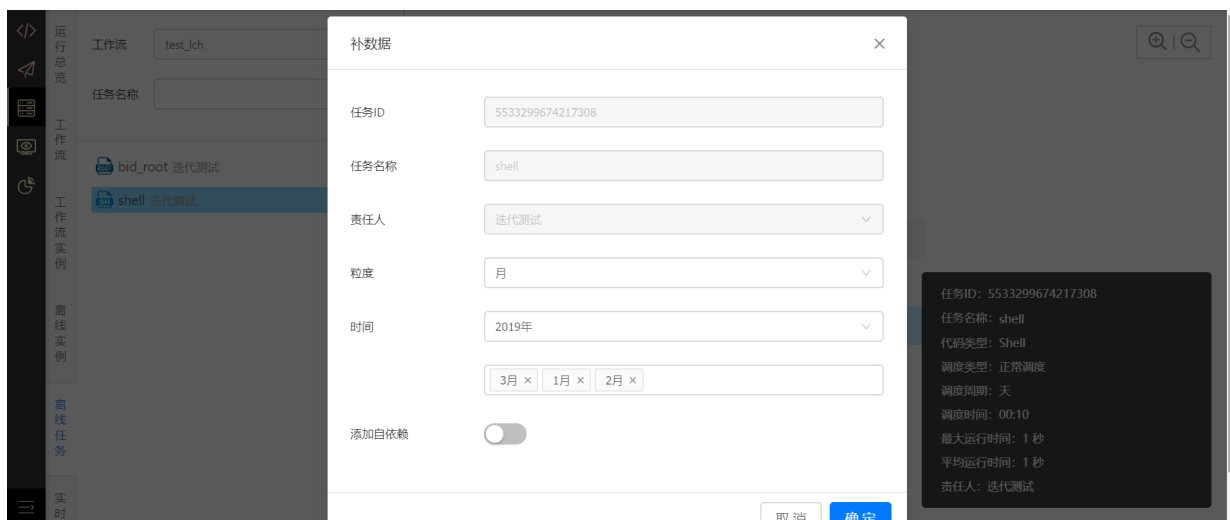
- 补数据

生产环境下的任务才可以进行补数据操作，补数据支持时间粒度为天，周，月。

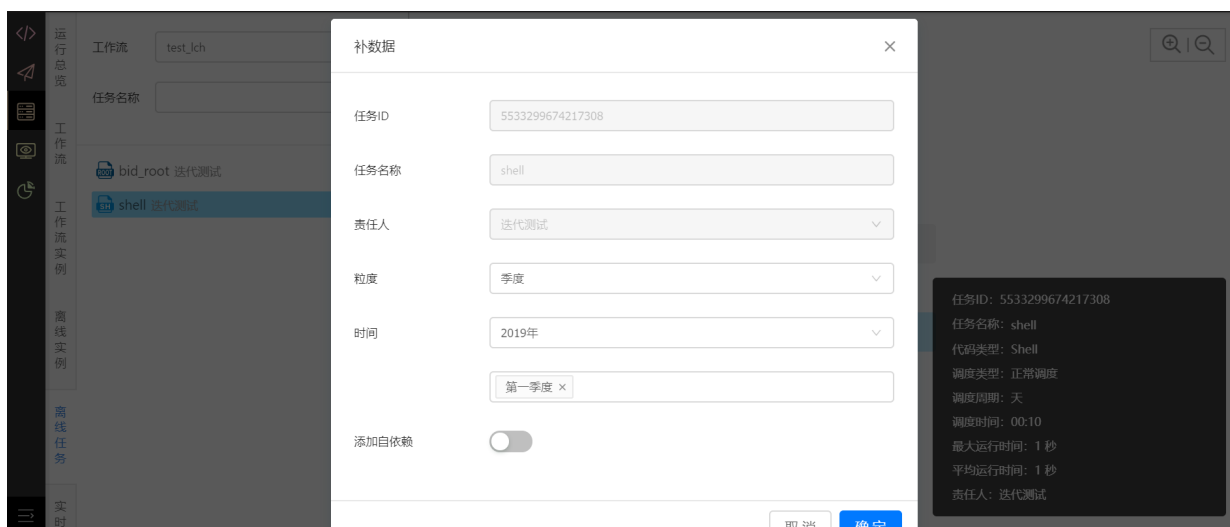
当选择时间粒度为天，如果任务选择连续补多天数据时，每天会生成一个实例。



当选择时间粒度为月，年可选前10年~今年，月可选1月~当前月，选择几个月将生成几个实例。



当选择时间粒度为季度，年可选前10年~今年，月可选第1季度~当前季度，选择几个季度将生成几个实例。



- 补下游

生产环境下任务才可以进行补下游操作，即补该任务及其所有下游任务，操作请参考补数据。

8.5 实时任务

(1) 实时任务页面可以查看项目空间下实时任务列表，可对实时任务进行启动、停止和查看日志、下载日志等操作。

The screenshot shows the '实时任务' (Real-time Task) management page. It includes a sidebar with navigation options like '运行总览', '工作流', '工作流实例', '高线实例', '高线任务', and '实时'. The main area has filters for '全部状态' (All Status), '责任人' (Responsible Person), and '启动时间' (Start Time). A search box for '任务名称' (Task Name) is present. Below the filters is a table of tasks:

任务名称	类型	状态	描述	责任人	启动时间	操作
hiveTOKafka	StreamDataSync	● 运行中	--	租户测试	2019-03-22 10:41:47	停止 查看日志 下载日志
saprkstreaming	SparkStreaming	● 失败	--	租户测试	--	启动 查看日志 下载日志
kafka	StreamDataSync	● 未运行	--	迭代测试	--	启动
kafka_hive	StreamDataSync	● 未运行	--	迭代测试	--	启动

At the bottom, there are pagination controls showing '共 4 条' (Total 4 items) and '10 条/页' (10 items per page).

- 点击停止，弹出二次确认窗口，点击确定，可停止该实时任务。
- 点击启动，点击启动，选择消费位置后重新启动。

This screenshot shows the same '实时任务' page, but with a context menu open over the 'hiveTOKafka' task. The menu options are:

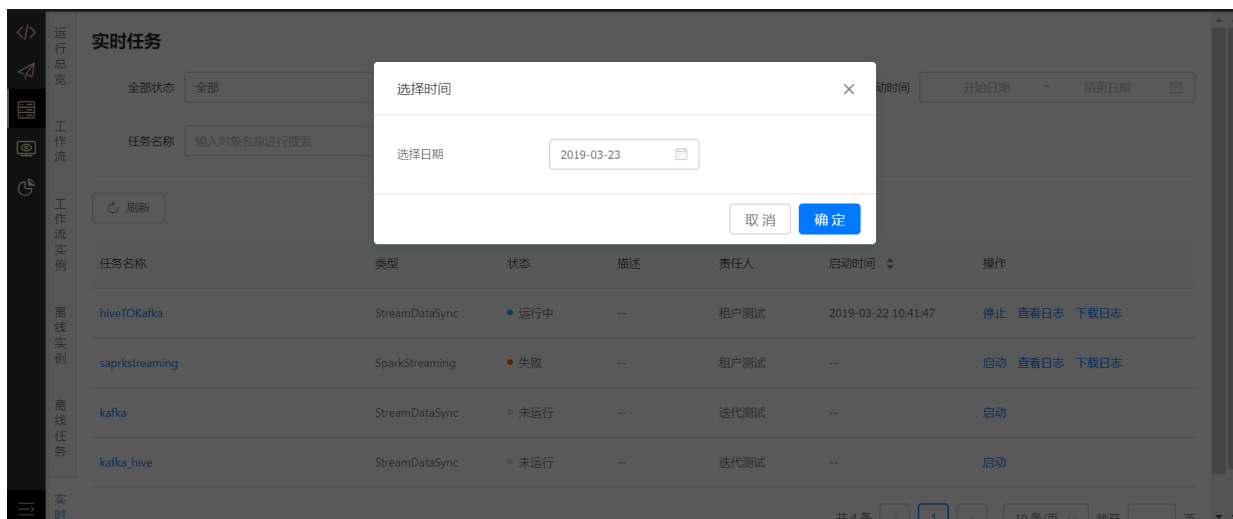
- 从上次消费位置启动 (GROUP_OFFSETS)
- 从最新位置启动 (LATEST)
- 从最早位置启动 (EARLIEST)
- 从指定位置启动 (SPECIFIC_OFFSETS)

The '启动' (Start) button in the task's action column is highlighted, and the context menu is positioned over it.

- 点击查看日志，弹出当前日志信息。

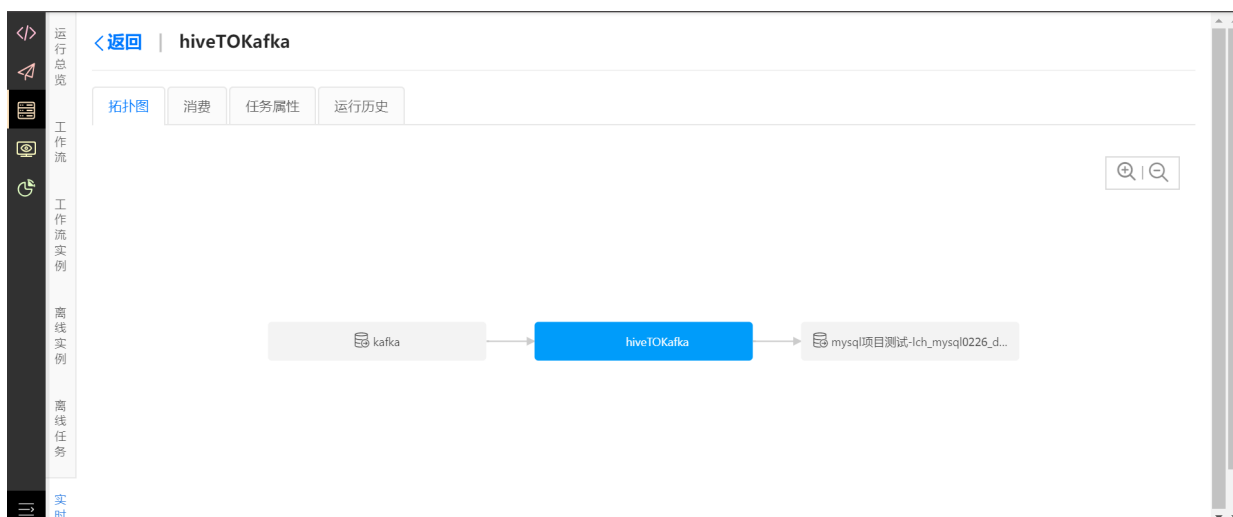
The screenshot shows a '日志' (Log) window. The log content consists of multiple lines of text, each representing a task's progress or status. The text is repetitive, showing task IDs and their current consumption positions. At the bottom right of the window, there is a '关闭' (Close) button.

- 点击下载日志，选择指定日期，点击确定下周该天日志到本地。

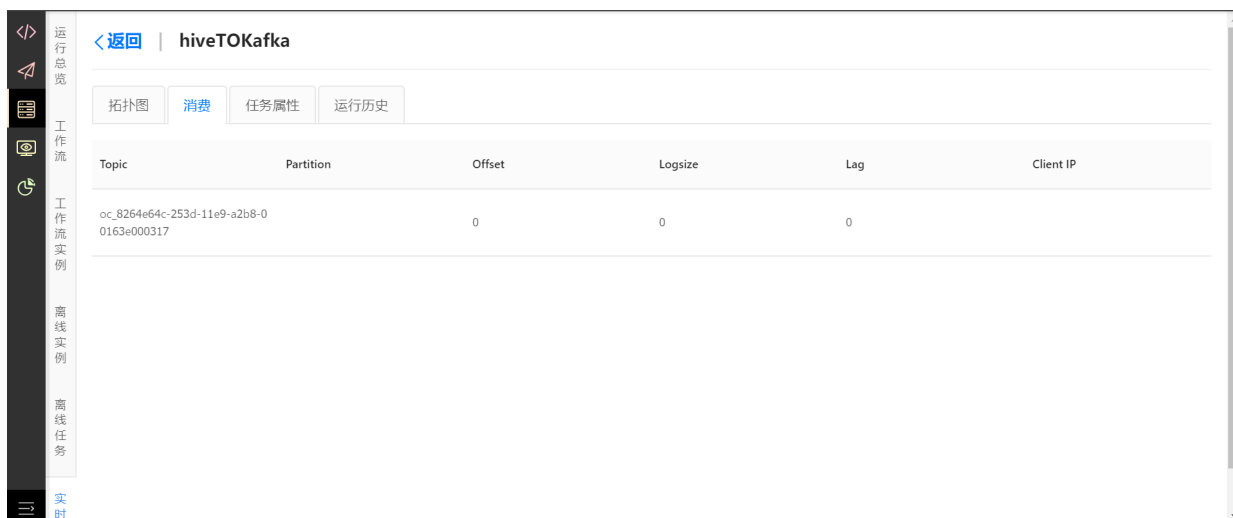


(2) 点击实时任务名称，可查看实时任务的拓扑图、消费、属性、运行历史。

- 点击拓扑图tab页，可查看到实时任务的拓扑图。



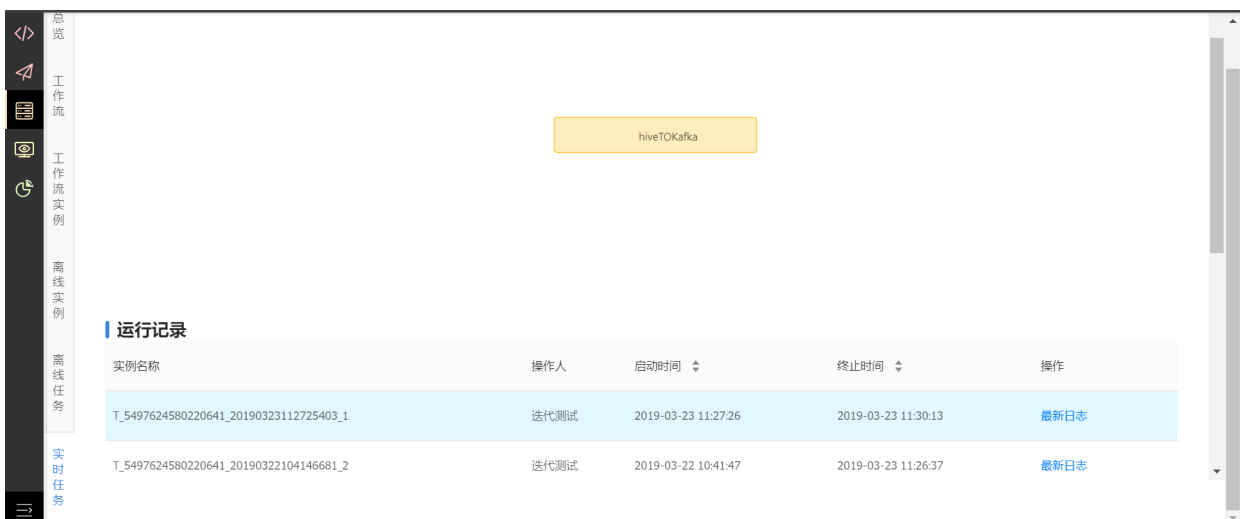
- 点击消费，可查看该任务的__consumer_offsets信息。



- 点击属性，可查看该任务属性。

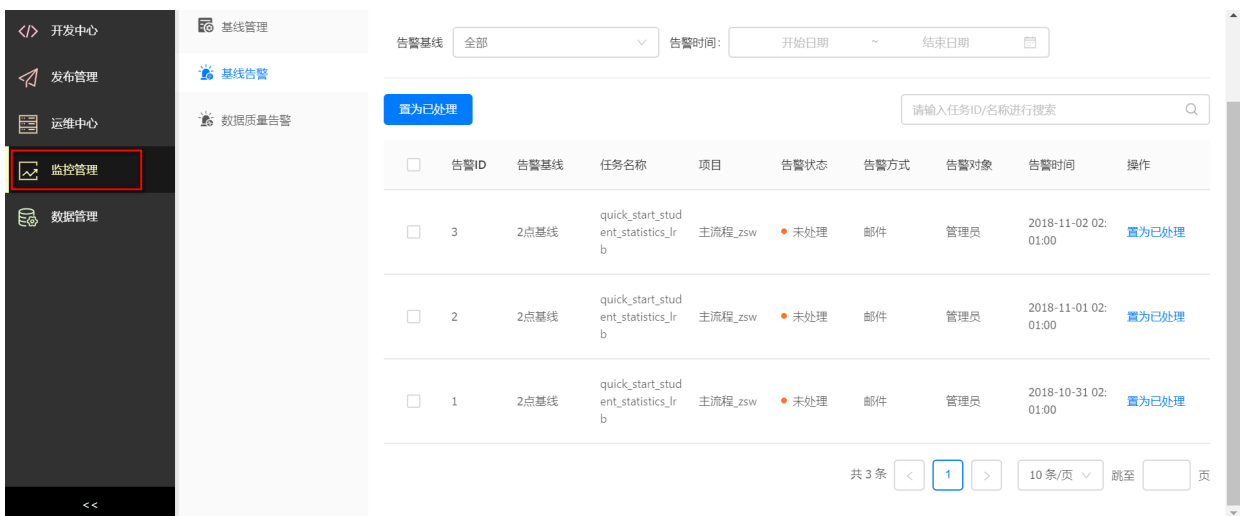


- 点击运行历史，可查看该实例的当前状态和运行实例历史。



9 监控管理

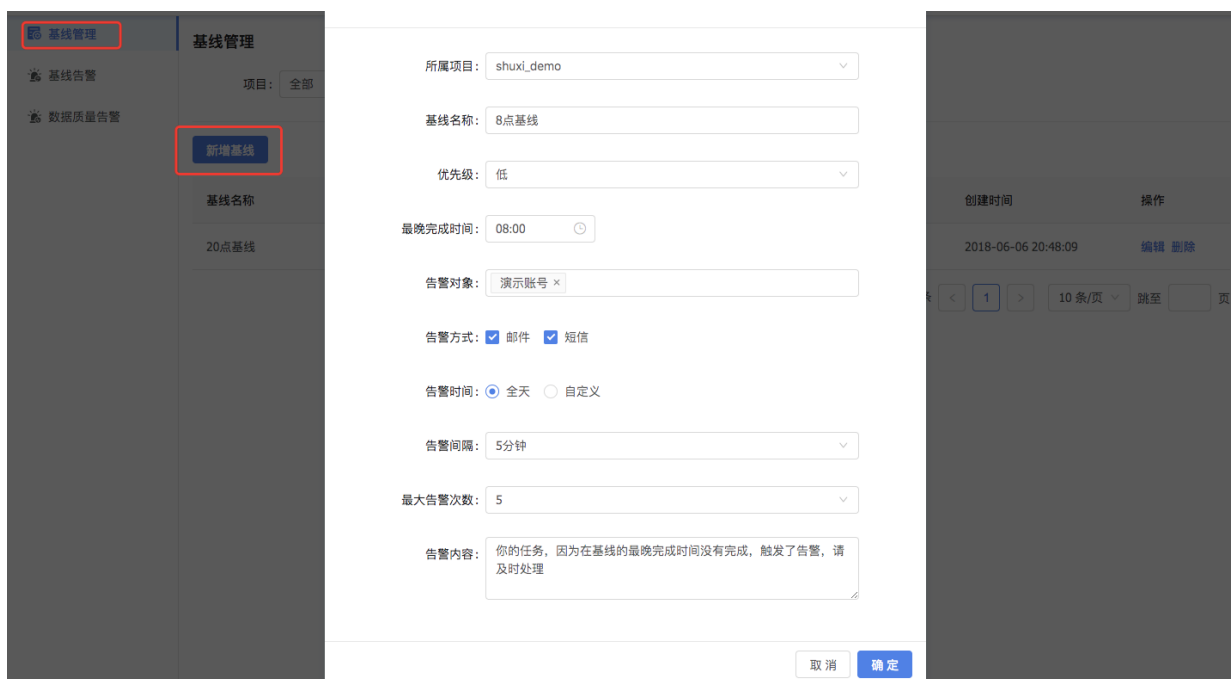
进入开发后，点击【监控管理】图标进入监控管理页面。



9.1 基线管理

在基线中定义最晚完成时间、告警方式及间隔，当任务配置基线后，如果在基线的最晚完成时间前未完成或者是失败状态，则会触发告警，给基线配置中的告警对象进行通知。下面演示新建基线。

(1) 进入**监控管理**—>**基线管理**界面，点击【**新增基线**】。



(2) 选择基线所属项目、基线名称、最晚完成时间和告警方式等。

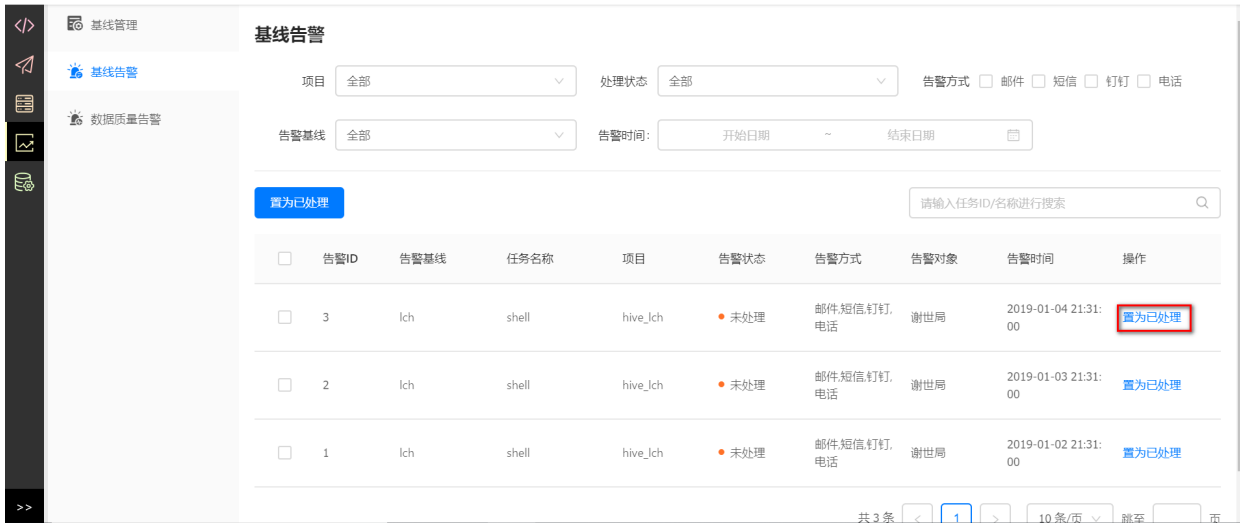
(3) 点击确定，出现系统的操作成功提示，即新建完成。

(4) 在开发套件中为任务配置基线后提交发布到生产环境，当任务未按时运行完成时就会触发基线告警。



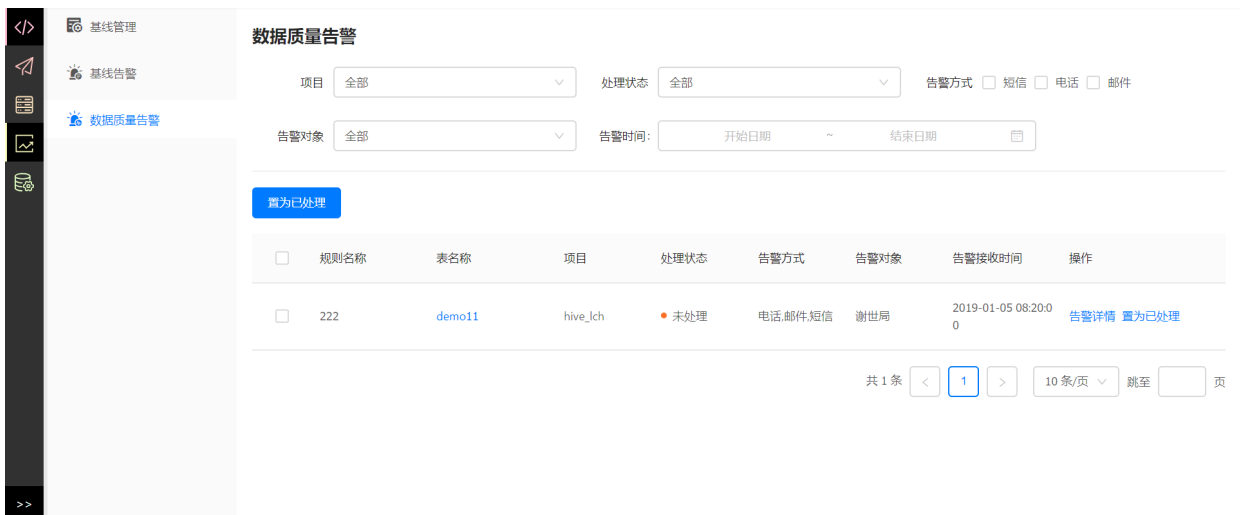
9.2 基线告警

基线告警页面以列表形式显示已触发基线的所有任务列表，当把任务处理(一般是重跑)后，可手动把状态设置为已处理。



9.3 数据质量告警

数据质量详细介绍见本文数据质量，数据质量告警页面以列表形式显示已触发数据质量监控规则的所有表。



10 数据管理

数据管理是对生产过程中表数据进行查看、分析、监控等操作，主要包含全局概览、元数据管理、数据权限、数据质量、数据目录、术语项管理等六个子功能。进入开发后，点击【数据管理】图标进入数据管理页面，进入后默认显示全局概览页面。

10.1 全局概览

全局概览展示租户下的元数据、数据质量、数据库和表等信息。



10.2 元数据管理

(1) 元数据管理中以列表形式展示当前管理的库和表的信息。

元数据管理

类型: 全部 | 数据源: 全部 | 类目: 全部

数据库: 全部 | 表名称: 请输入表名称进行搜索 | 字段名称: 请输入字段名称进行搜索

表名称	类型	数据库	数据源	类目	存储大小	数据更新时间	操作
performance_test_cl_t mp_1	Greenplum	gp_uniqlq.public	gp_多库_gp_uniqlq_de v_1544752471251	--	95.09 MB	--	详情 血缘
performance_test_cl_t mp_1	Greenplum	gp_lch_oss.public	gp_lch_存储为hdfs_gp_lch_oss_dev_15453702 11222	--	95.09 MB	--	详情 血缘
performance_test_cl_t mp_1	Greenplum	gp_conn_lch3.public	mysql_多库_gp_conn_lch3_dev_15447767821 44	--	95.09 MB	--	详情 血缘
performance_test_cl_t	Greenplum	gp_lch1213.public	gp_多库_gp_lch1213新 库 prd_154470881685	--	95.09 MB	--	详情 血缘

(2) 元数据管理表详情，点击详情进入到表的详细信息界面。主要包括以下信息：

- 基本信息，包括中文名、表创建时间、DDL更新时间、数据更新时间、所属类目和描述。
- 存储信息，包括总存储量、昨日新增量、生命周期、存储方式。
- 明细信息，包含字段信息、分区信息、产出信息、数据预览。

test_1109_prd.demo_tdm_user_complaint_position_count_d [查看血缘关系](#)

基本信息

中文名: [编辑](#)

责任人: [编辑](#)

表类型: 公开表 [编辑](#)

表创建时间: 2018-11-09 21:10:34

DDL更新时间: 2018-11-09 21:10:34

数据更新时间:

所属类目: [修改类目](#)

描述: 历史以来业主投诉位置的次数统计 [编辑](#)

存储信息

数据库类型: Hive

总存储量: 625.15 KB

昨日新增量: 52.63 KB

生命周期: 永久 [编辑](#)

存储方式: 分区

明细信息

[字段信息](#) | [分区信息](#) | [产出信息](#) | [数据预览](#)

- 生命周期

生命周期目前只能针对分区表，在表**存储信息**中可设置生命周期: 7天、永久、30天和366天，默认为永久。如果表的生命周期设置为7天，则只保存最近7天的数据，其他分区会被平台自动删除。

只有hive类型且有分区的表才可修改生命周期。



● 血缘关系

平台会根据任务间的依赖关系，自动解析出表的血缘关系。因为只有生产环境的任务会按照依赖关系进行调度执行，因此血缘信息只针对于生产环境的库对应的表。血缘信息展示可以从任意一张表出发，向上追溯到数据的源头，可以向下追溯到依赖该表数据的最新产出的表的信息。



10.3 数据权限

数据权限是对当前租户下的数据库、队列、HDFS操作权限进行管理，当一个用户没有对某个库、表或者字段的操作权限时，执行任务会提示鉴权失败。

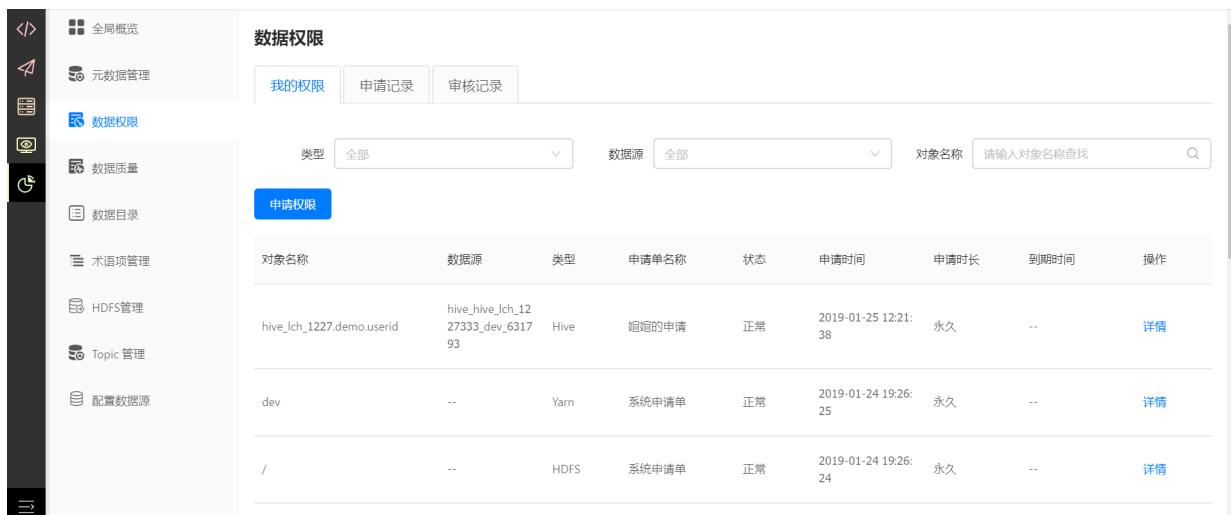
注意：只有管理员在项目管理中为该项目开启安全检查时，数据权限检查才会生效，开启安全检查的方式可查看**项目配置**部分。

开发角色只可查看自己当前的权限和权限申请记录；运维角色所有角色当前的权限和权限申请记录。

当管理员为用户授权为项目的开发、运维或管理员角色时，系统会自动为该成员开通当前项目数据库的整库权限。

(1) 我的权限

我的权限页面可以查看当前用户当前的权限列表和权限详情。



(2) 申请权限

- Hive

在类型中选择Hive，如果需要申请整库权限，在**是否整库**处勾选**是**，然后选择相应的数据库以及表操作权限，提交后等待管理员审核。

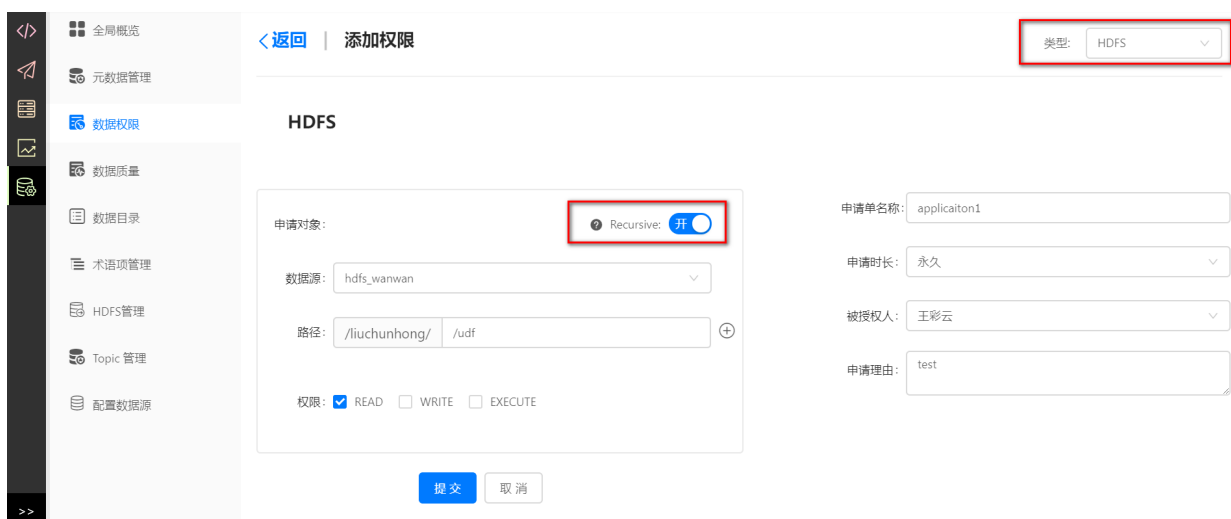


如果需要申请某张表的权限，在**是否整库**处不勾选，然后选择对应的数据库、表和操作权限。只有操作权限选择SELECT时，才可选择字段值。



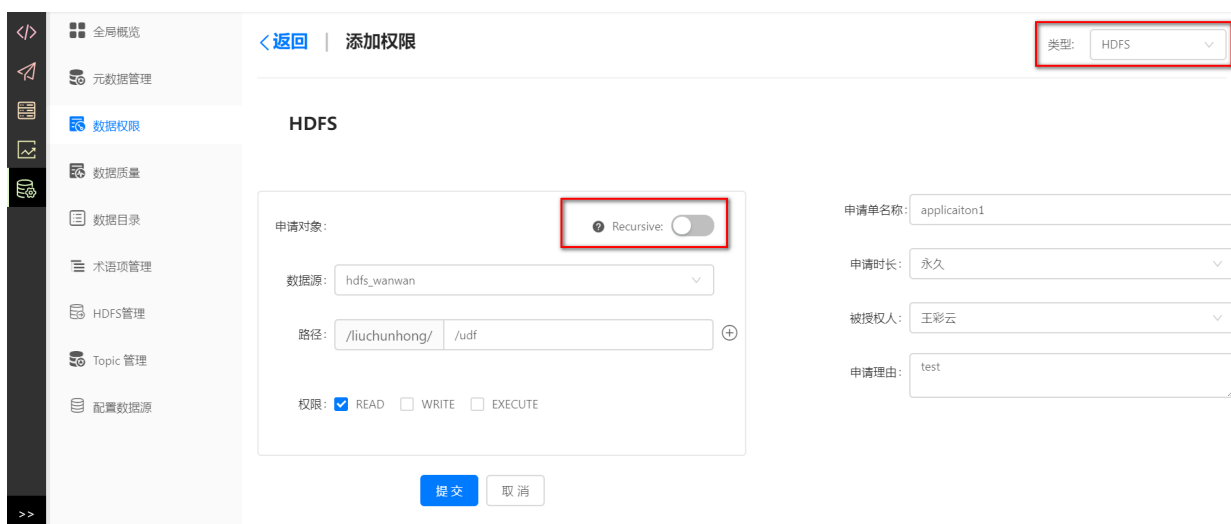
- HDFS

在类型中选择HDFS，如果需要目录及所有子目录权限，开启Recursive，然后选择相应的数据源并填写路径，提交后等待管理员审核。



The screenshot shows the '添加权限' (Add Permission) interface for HDFS. The '类型' (Type) dropdown is set to 'HDFS'. The '申请对象' (Request Object) field is empty. The 'Recursive' checkbox is checked and highlighted with a red box. The '数据源' (Data Source) is 'hdfs_wanwan'. The '路径' (Path) is '/liuchunhong/ /udf'. The '权限' (Permissions) are 'READ', 'WRITE', and 'EXECUTE'. The '申请单名称' (Request Name) is 'applicaiton1', '申请时长' (Request Duration) is '永久', '被授权人' (Authorized Person) is '王彩云', and '申请理由' (Request Reason) is 'test'. The '提交' (Submit) button is highlighted.

如果仅申请当前目录的权限，关闭Recursive，然后选择相应的数据源并填写路径，提交后等待管理员审核。



The screenshot shows the '添加权限' (Add Permission) interface for HDFS. The '类型' (Type) dropdown is set to 'HDFS'. The '申请对象' (Request Object) field is empty. The 'Recursive' checkbox is unchecked and highlighted with a red box. The '数据源' (Data Source) is 'hdfs_wanwan'. The '路径' (Path) is '/liuchunhong/ /udf'. The '权限' (Permissions) are 'READ', 'WRITE', and 'EXECUTE'. The '申请单名称' (Request Name) is 'applicaiton1', '申请时长' (Request Duration) is '永久', '被授权人' (Authorized Person) is '王彩云', and '申请理由' (Request Reason) is 'test'. The '提交' (Submit) button is highlighted.

- Hbase

在类型中选择Hbase，选择数据源、对象和权限，对象可选命名空间和列族，填写表单信息后提交等待审核。

Hbase

申请对象:

数据源: hbase

命名空间: default.t1

权限: READ WRITE CREATE

提交 取消

申请单名称: application1

申请时长: 7天

被授权人: 王彩云

申请理由: test

• Yarn

在类型中选择Yarn，选择计算引擎和队列，填写表单信息后提交等待审核。

Yarn

申请单名称: application2

计算引擎: hive_test

队列: dev

申请时长: 7天

被授权人: 王彩云

申请理由: 请输入申请理由

(3) 申请记录

在申请记录中可以看到当前用户历史权限申请记录和申请详情。

数据权限

我的权限 申请记录 审核记录

类型: 全部 审批状态: 全部 申请单名称: 请输入申请单名称

申请单名称	类型	申请时间	审批人	状态	审批意见	审批时间	操作
yarn_1_1	Yarn	2019-01-05 12:57:13	--	待审核	--	--	详情 取消申请
整库的select权限	Hive	2019-01-04 14:12:24	谢世局	审核通过	test	2019-01-04 14:13:03	详情
hdfs没有输入路径的read权限	HDFS	2019-01-04 11:50:54	谢世局	审核通过	ccc	2019-01-04 11:51:03	详情
ooo	HDFS	2019-01-04 11:46:06	--	待审核	--	--	详情 取消申请

(4) 审核记录

管理员在审核记录中可查看所有的权限申请记录，点击**同意**后该用户将获得相应的库或者表操作权限，点击**拒绝**后该用户将无法获得相应的权限；点击**收回权限**后该用户原有的权限将会收回。

申请单名称	类型	申请人	申请时间	审批人	审批时间	操作
yarn_1_1	Yarn	王彩云	2019-01-05 12:57:13	--		同意 拒绝
系统申请单	Yarn	谢世局(同步)	2019-01-05 10:11:23	王彩云	2019-01-05 10:11:23	查看详情 收回权限
系统申请单	Hive	谢世局(同步)	2019-01-05 10:11:22	王彩云	2019-01-05 10:11:23	查看详情 收回权限
系统申请单	HDFS	谢世局(同步)	2019-01-05 10:11:22	王彩云	2019-01-05 10:11:23	查看详情 收回权限
系统申请单	Hive	谢世局(同步)	2019-01-05 10:11:21	王彩云	2019-01-05 10:11:22	查看详情 收回权限
系统申请单	HDFS	谢世局(同步)	2019-01-05 10:11:21	王彩云	2019-01-05 10:11:22	查看详情 收回权限

10.4 数据质量

数据质量主要是对分区表数据的准确性和数据量进行校验，按照通用和自定义的规则进行校验和检查，并有可视化的工具对问题数据和任务进行记录和展示。

表名称	数据库	数据库类型	监控类型	状态	数据更新时间	操作
RUBIN	xe.LCH	Oracle	--	--	2019-01-04 13:35:52	配置规则
LCH	xe.LCH	Oracle	--	--	2018-12-28 13:39:45	配置规则
TEST	xe.SYSTEM	Oracle	--	--	2018-11-13 07:46:18	配置规则
HELP	xe.SYSTEM	Oracle	--	--	2011-08-28 22:18:14	配置规则
SQLPLUS_PRODUCT_PROFILE	xe.SYSTEM	Oracle	--	--	2011-08-28 22:18:13	配置规则
DEFS_AQERROR	xe.SYSTEM	Oracle	--	--	2011-08-28 22:16:58	配置规则

点击**配置规则**后，可配置表监控或字段监控。表级监控包括记录波动、每日新增存储量和总存储量。字段级监控包括字段规范性和字段值两种。

规则名称: 请输入规则名称

规则类型: 记录波动

对比对象: 每日新增存储量

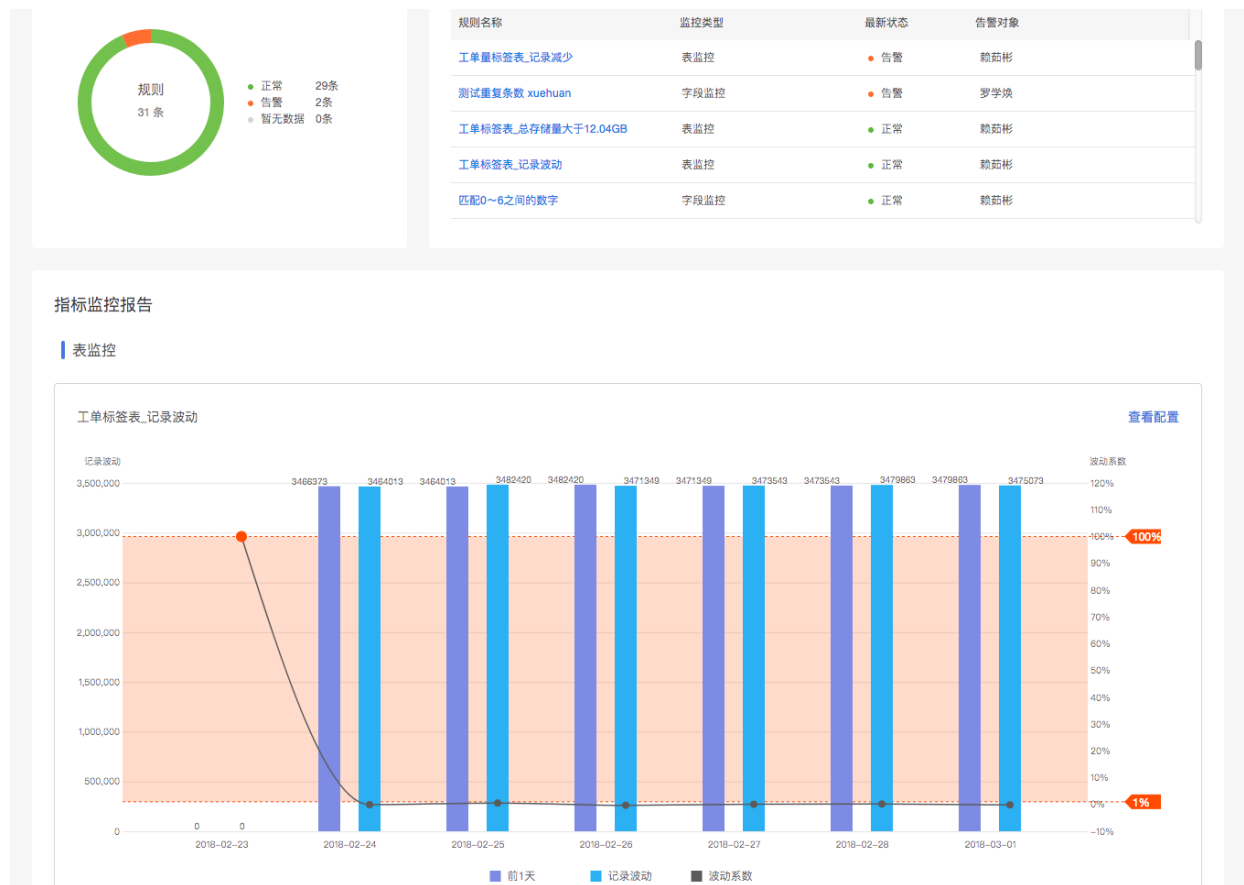
监控策略: 总存储量

警告对象: 王彩云

警告方式: 短信 电话 邮件



数据质量报告。规则配置好后，第二天可以查看到数据质量报告。



10.5 数据目录

数据目录支持用户按照目录层级关系组织表，数据目录包含数据数目和标签类目两部分。数据类目一般针对于原始表，例如用户上传的数据表、数据同步后的表；标签类目一般针对于加工后的表，例如加工后学生兴趣属性表、学生成绩排序表等。

可右键【添加子类目】来建立数据类目。



在表详情中点击【修改类目】为该表指定类目。



10.6 术语项管理

术语项管理支持用户按照目录层级关系组织表字段，每个术语项可绑定多个表的字段。

可右键【添加子术语】来建立数据类目。



在表详情中点击【修改类目】为该表指定类目。



10.7 Topic管理

Topic管理支持查看topic详情以及管理schema，点击【Topic管理】，默认显示第一个kafka的topic列表。

Topic 管理

Kafka 数据源: kafka_internal_hdp | Topic 名称: 请输入名称进行搜索 | Schema 名称: 请输入名称进行搜索

Topic	Kafka 数据源	Schema	操作
bas_broker	kafka_internal_hdp	bas_broker	管理
ambari_kafka_service_check	kafka_internal_hdp	--	管理

共2条 | 1 | 10 条/页 | 跳至 | 页

(1) 管理topic：点击管理，显示该topic中的基本信息、schema信息、明细信息和数据预览。

< 返回 | bas_broker

基本信息

名称: bas_broker
创建时间: 2019-01-05 22:09:12
更新时间: 2019-01-05 22:09:12
日志保留时长: 7
副本数: 3
分区数: 6
描述:

Schema 信息

名称: bas_broker
类型: avro
schema 组: realtime
描述:

明细信息

Schema | Partition

编辑 | version 1

```
1 {
2   "type": "record",
3   "name": "bas_topic",
4   "namespace": "com.dtwave.meteor",
5   "fields": [
6     {
7       "name": "cluster_id",
8       "type": "string"
9     },
10    {
11      "name": "broker_id",
12      "type": "string"
13    },
14    {
15      "name": "host",
16      "type": [
17        "null",
18        "string"
19      ]
20    },
21    {
22      "name": "port",
23      "type": [
24        "null",
25        "int"

```

序号	列名	列类型	是否允许为空
1	cluster_id	string	否
2	port	int	是
3	broker_id	string	否
4	host	string	是
5	status	int	是

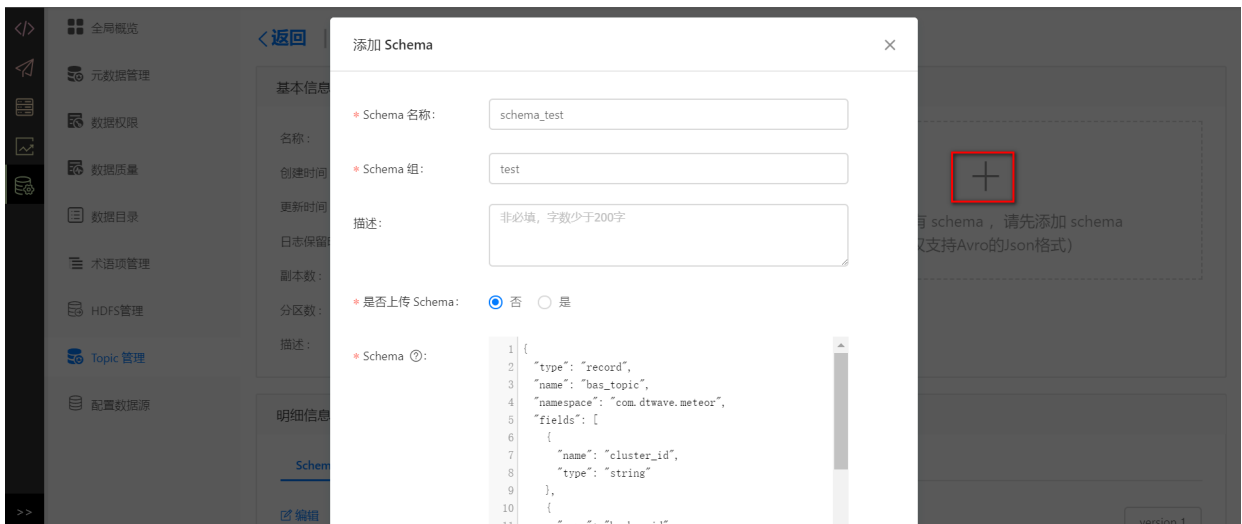
数据预览

显示最新 条数据 刷新

cluster_id	broker_id	host	port	status
8d26be6b02c547caa141936ff6de5191	abc0	192.168.1.2	8801	0
a4ee8f40da5248bba3fa6333c8aeef44	abc1	192.168.1.3	8801	0
1d0c47f349cb425abb4179f1d690f8c6	abc2	192.168.1.4	8801	0
9cc0c8eba89a4ae6b6efcf8c2cfe1bb	abc0	192.168.1.5	8801	0
81c2eb1020e34fa990b25ba910006f40	abc1	192.168.1.6	8801	0
745908b67ca2487fb66c3d9cc426db39	abc2	192.168.1.2	8801	0
eb12294695404394b134784c08aec16f	abc0	192.168.1.3	8801	0
f935430b38b448cca5e83ed1b281e087	abc1	192.168.1.4	8801	0
95ecc1dbdd074314ab614f9e8eb49b9f	abc2	192.168.1.5	8801	0
843a638e4378446e8c9835caca96b519	abc0	192.168.1.6	8801	0

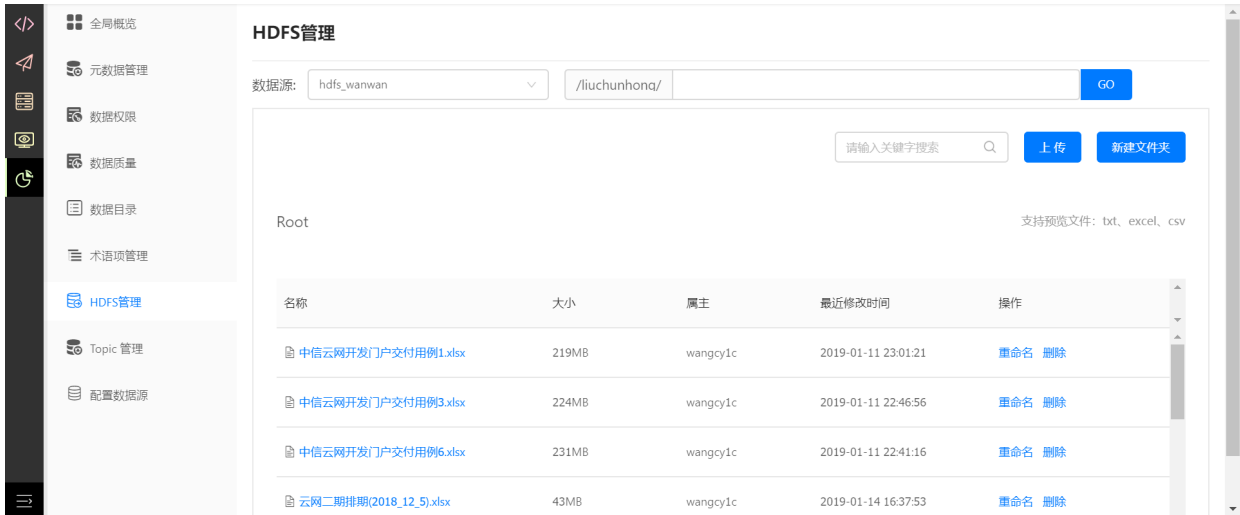
< 1 2 >

(2) 添加topic: 若topic未添加schema, 可为该topic添加一个schema。

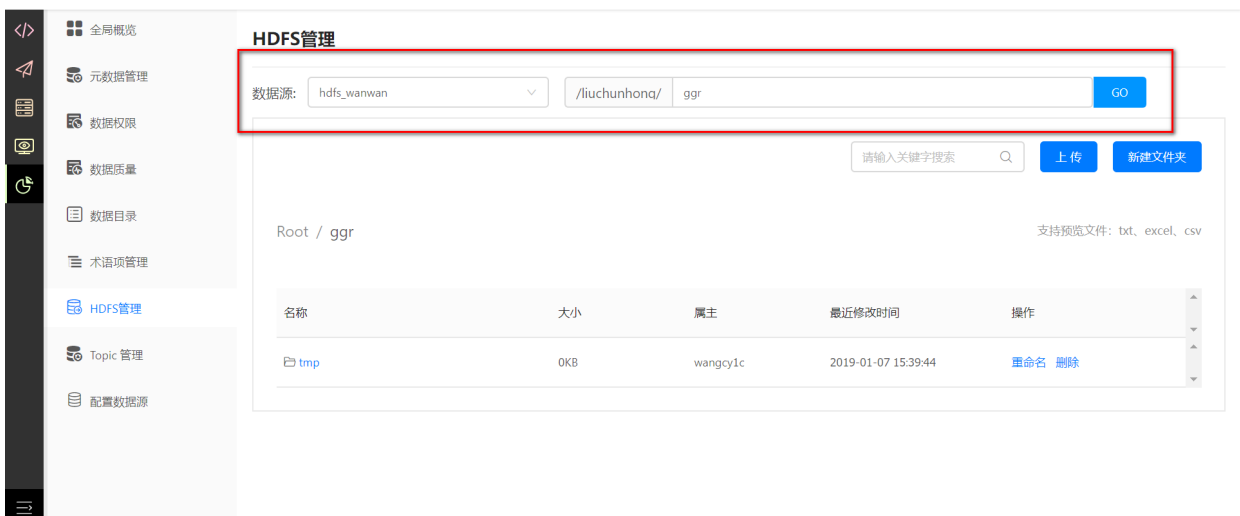


10.8 HDFS管理

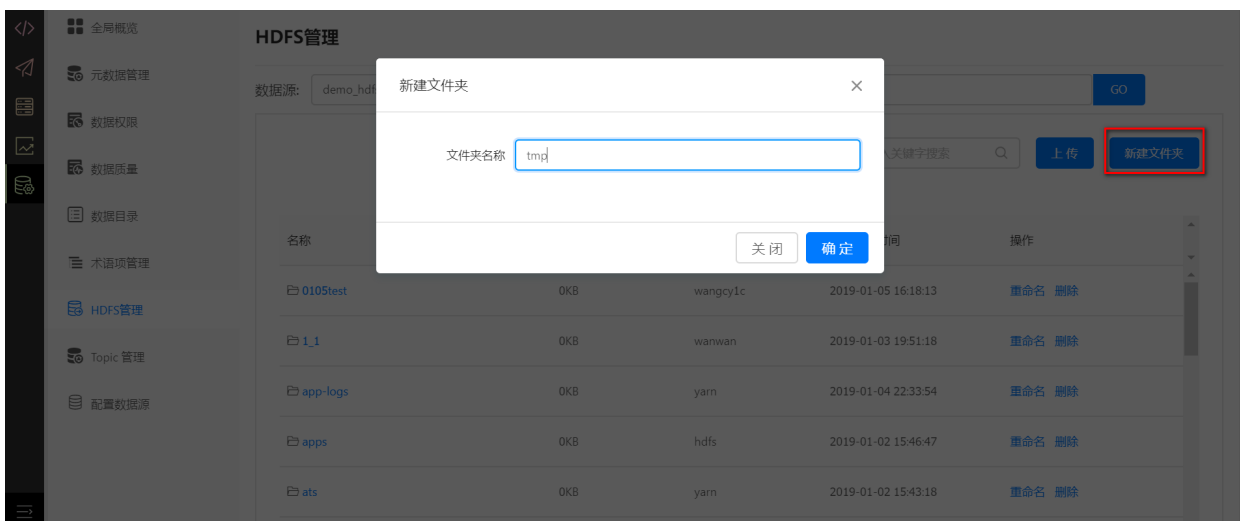
点击【HDFS管理】可对HDFS文件进行管理。



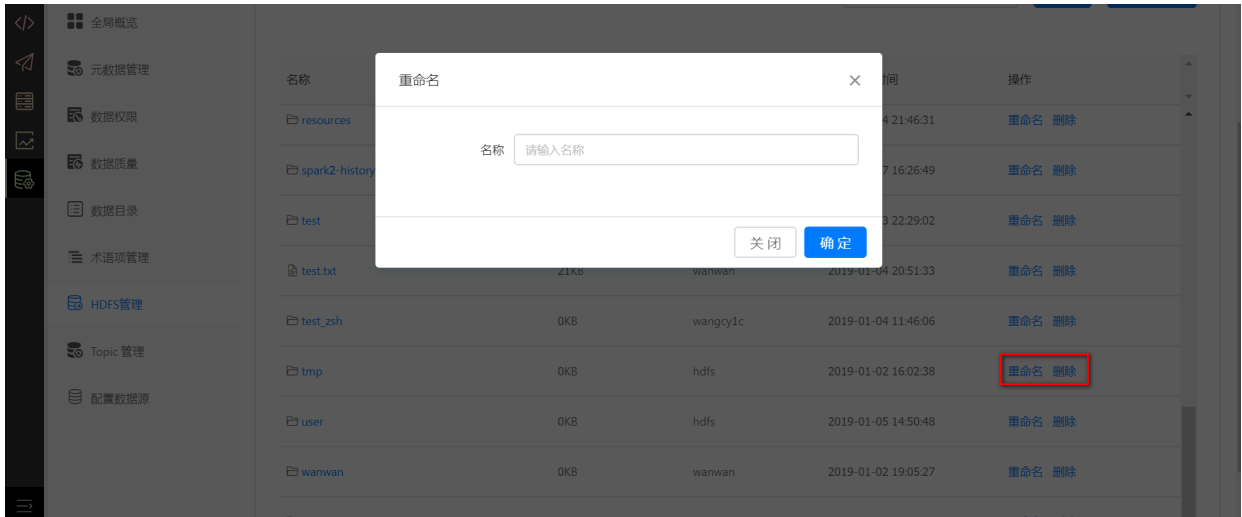
可选择数据源并输入资源目录，点击GO定位到该目录。



可下当前目录下上传文件或新建文件夹。



可删除或重命名文件夹或文件。



10.9 Hbase查询

点击【数据查询】，进入数据查询页面，目前支持Hbase查询。

选择数据源和表名称后，点击搜索可查看该表最新200条数据。可通过时间范围、Rowkey、列族、列来查询数据。可配置最多显示的数据条数，范围为1~2000。



11 标签管理

11.1 后台类目

11.1.1 进入后台类目

点击左侧标签管理-后台类目进入后台类目。



11.1.2 创建类目

后台类目需要创建类目，同样支持最多创建三级类目。类目以树形结构对数据资产进行分类管理、展示和检索，通过类目的形式组织标签可以实现对数据的规范化及标准化管理。让用户根据可理解的分类方式快速查找数据、标签。

点击【+】图标，在创建类目的弹框中，输入类目名称和描述，点击确定即完成类目的创建。

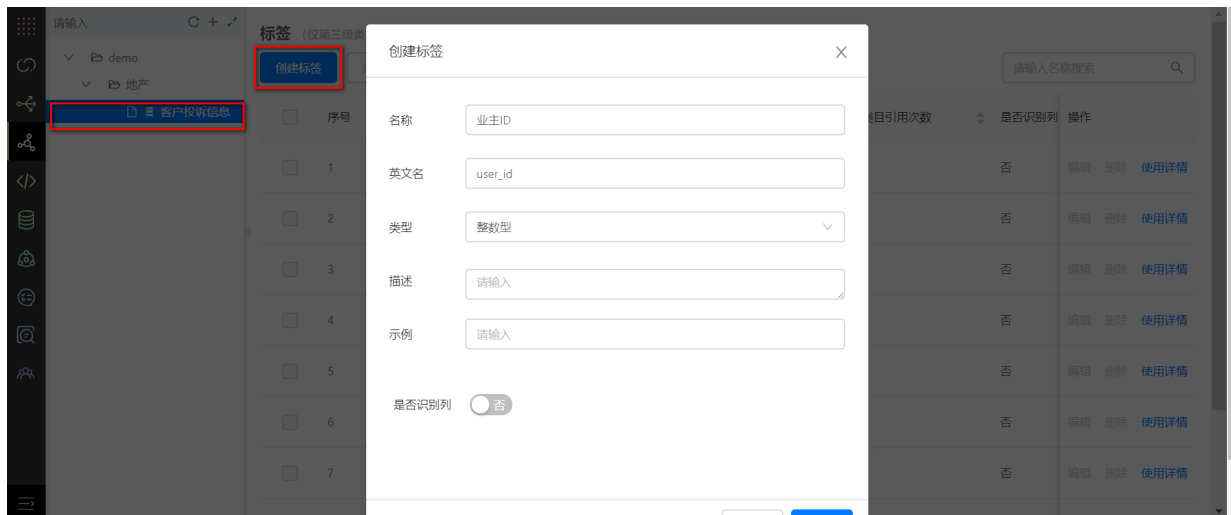


在此类目下可以继续创建二级类目、三级类目。如图所示。



11.1.3 创建标签

选择三级类目，点击【创建标签】，在弹窗表单中填写信息，创建后台标签，信息项包含“标签名”、“英文名”、“类型”等，如下图所示：



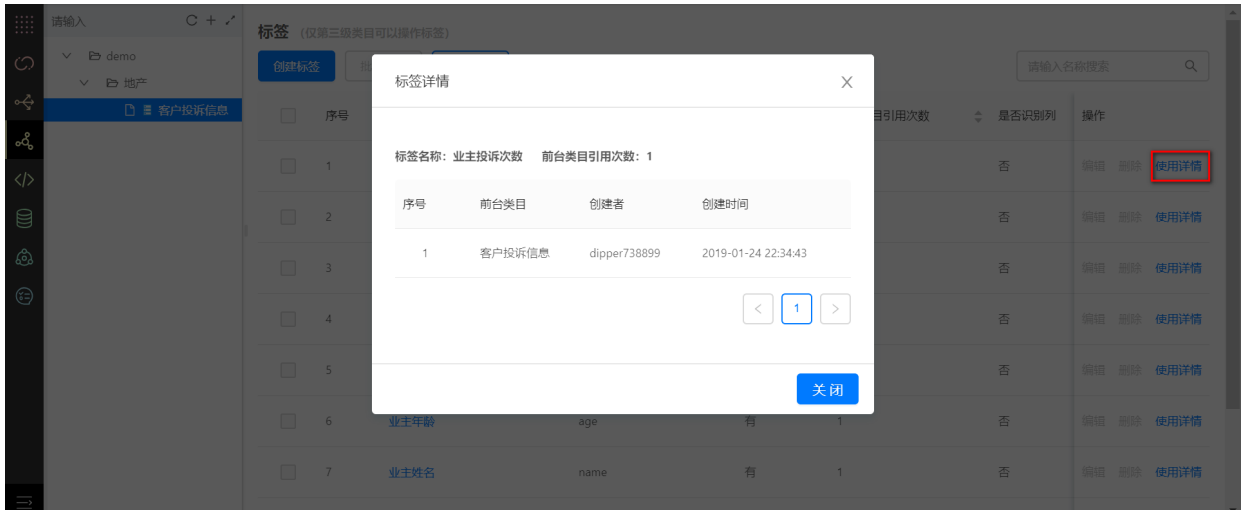
标签创建完成后，可以在标签列表页查看已添加的所有标签。标签列表如下图所示：



点击标签名查看标签详情。

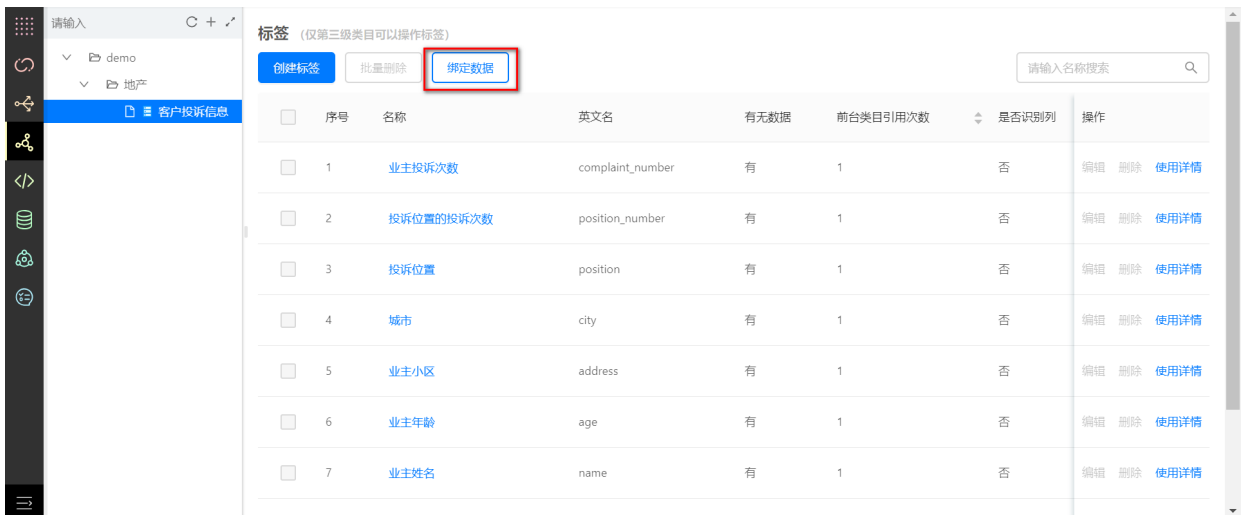


点击使用详情查看标签使用详情。

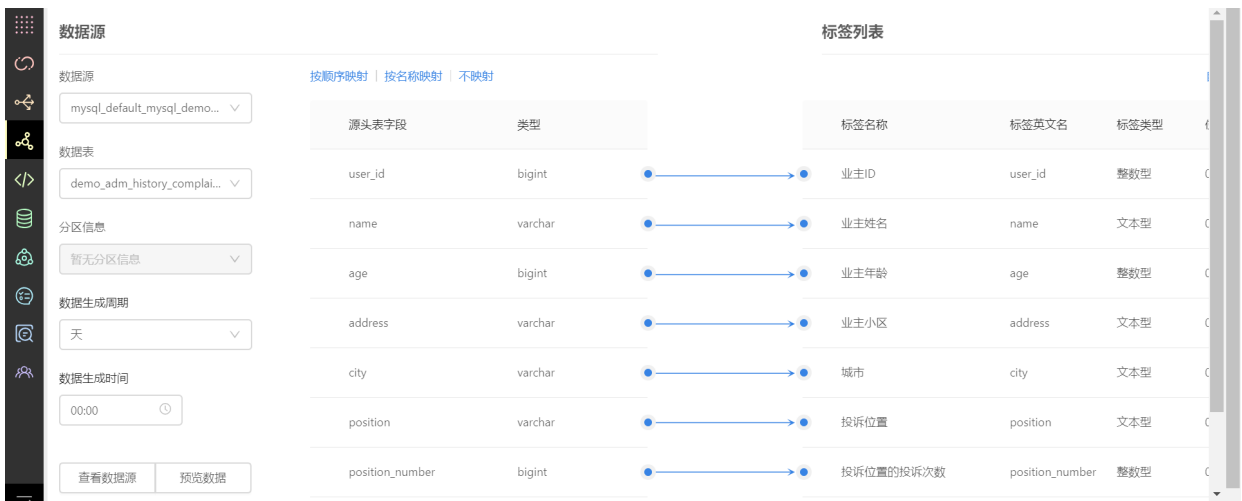


11.1.4 绑定数据

创建完成的标签，需要绑定数据后才能被真正使用起来为业务创造价值。在标签列表页点击“绑定数据”，即进入绑定数据的页面。



绑定数据时需要先选择数据源，即某个数据库下的表（如有分区），并且自定义数据的产生周期（支持天、周、月的粒度），如下图所示：



11.2 前台类目

11.2.1 进入前台类目



11.2.2 创建类目

创建类目时最多支持三级类目。类目以树形结构对数据资产进行分类管理、展示和检索，通过类目的形式组织标签可以实现对数据的规范化及标准化管理。让用户根据可理解的分类方式快速查找数据、标签。

点击【+】图标，在创建类目的弹框中，输入类目名称和描述，点击确定即完成类目的创建。

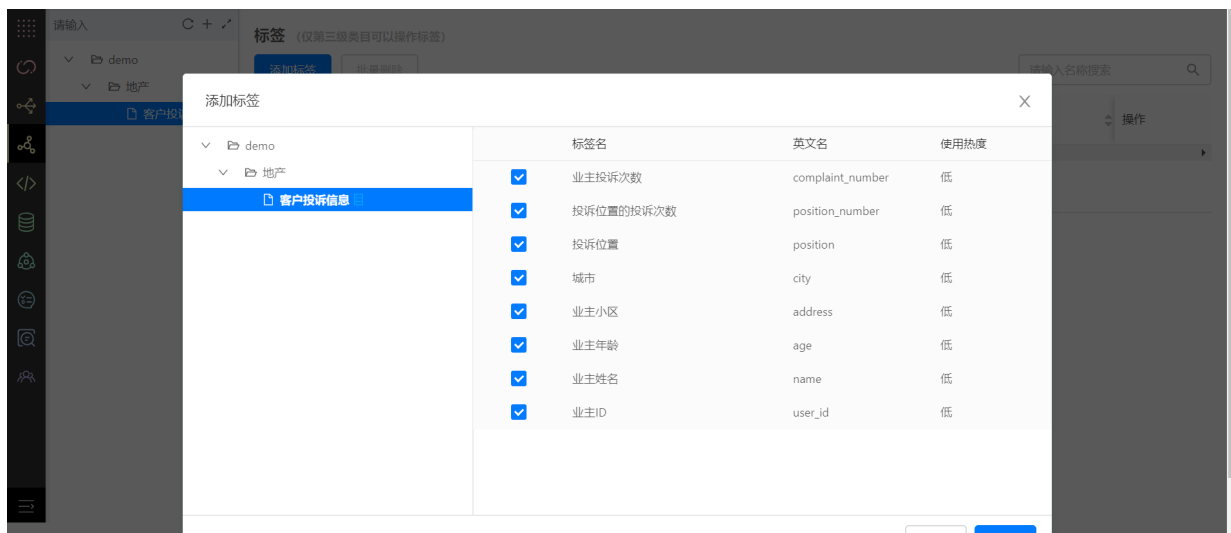


在此类目下可以继续创建二级类目、三级类目。如下图所示。



11.2.3 选择标签

选择三级类目，点击【添加标签】按钮，添加前台类目，挑选后台标签即可。



标签添加完成后，可以在标签列表页查看已添加的所有标签。



点击标签名称，可查看标签详情。



点击使用详情查看标签使用详情。



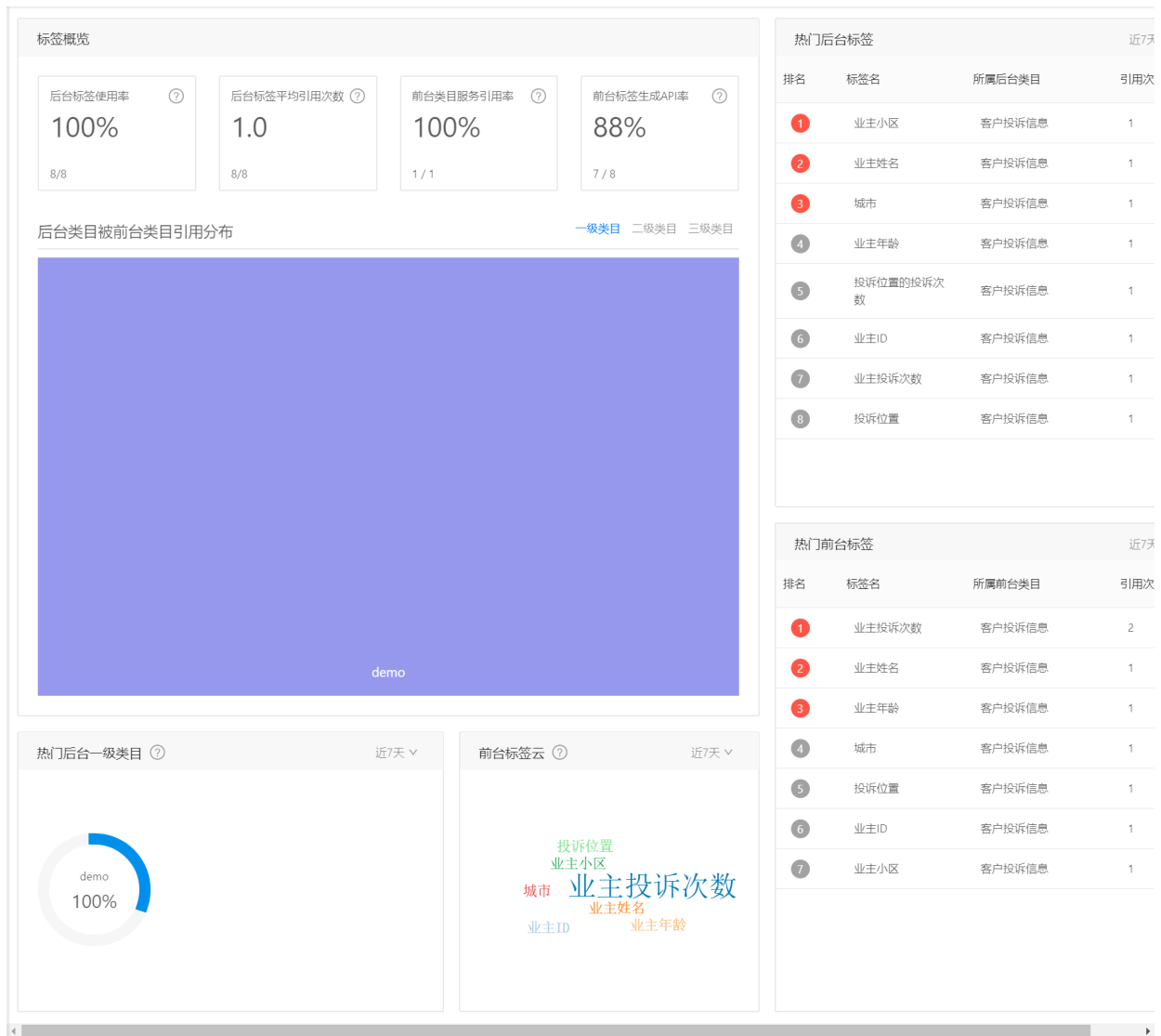
11.3 标签总览

点击左侧导航栏“标签管理-标签总览”，即可进入总览页面，标签总览的入口如下图所示：



在标签总览页面中，用户可以实时监控标签的使用率、引用次数、服务引用率、API生成率、热门标签排名等。

让企业能快速了解标签的总体使用情况，能优化标签设计，提供更精准更符合业务场景需要的标签体系。如下图所示：



12 数据服务

从控制台中进入数据服务，需要完成创建数据服务、数据配置、服务开发，生成服务API流程。

12.1 创建数据服务

首先需要创建服务，填写服务名称、描述、引擎和部署环境。这里根据实际场景需要来选择不同的引擎以及部署环境。这里以[分析引擎]为例，输入“服务名称”、“描述”，选择“分析引擎”，部署环境，创建完成，如下图所示：

若引擎为空，请进入【管理控制台】，添加【分析引擎】。

若部署环境为空，请进入【数据源管理】，将数据源授权给【数据服务】。



12.2 数据配置

12.2.1 选择前台类目

点击【+】图标，弹出表单，选择前台类目，该类目下的标签会自动同步数据，如下图所示：



点击【确定】后，显示所选前台类目和标签列表。



点击三级类目右侧图标，选择删除，可删除该类目。



12.2.2 数据同步

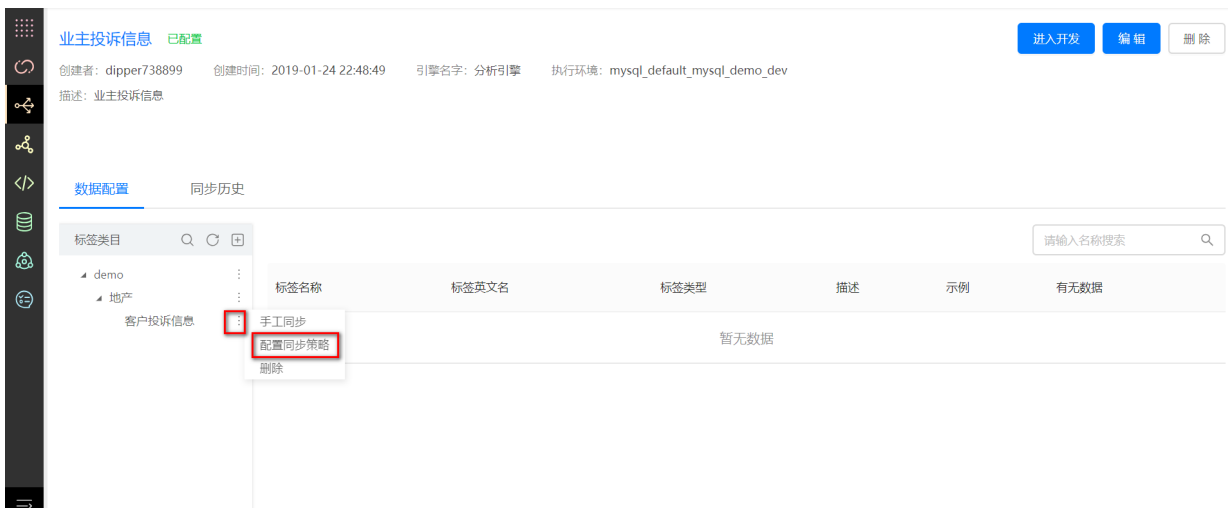
(1) 手工同步

服务的前台类目配置后，点击三级类目右侧图标，可以选择“手工同步”，点击后即触发数据同步任务。



(2) 自动同步

除了手工触发数据同步外，系统支持按照自定义周期进行自动同步。点击三级类目右图标可以选择“配置同步策略”。



弹出配置表单，选择是否调度，配置调度周期和调度时间，目前支持天、周、月级别的周期。如下图所示：



(3) 同步历史

切换到“同步历史”的Tab页，可以看到同步的基本信息，并可以根据一级类目名称搜索同步记录，同时在操作列中可以点击查看同步日志，如下图所示：



12.3 服务开发

12.3.1 创建仪表盘

服务的前台类目配置完成且数据同步成功后，可以点击服务详情页右上角的【进入开发】跳转到分析引擎页面，点击创建仪表盘，输入仪表盘名称，点击确定。



12.3.2 创建图表

点击添加图表，输入图标名称，选择标签组，点击确定。



12.3.3 配置图表

图表创建完成后，即可对图表进行编辑。

左侧标签标识的解释：#表示该标签为数字类型，当作为数值标签时聚合操作有求和、求平均值、计数、去重计数、最大值、最小值。T表示该标签为文本类型，当作为数值标签时聚合操作只有计数、去重计数。在配置维度标签时，点击维度标签，可以选择对该标签的排序方式，方式有升序、降序、默认。

在配置数值标签时，点击数值标签，可以选择对该标签的聚合方式和排序方式，当标签为数字类型时聚合方式有求和、求平均值、计数、去重计数、最大值、最小值，

当标签为文本类型时聚合方式只有计数、去重计数。

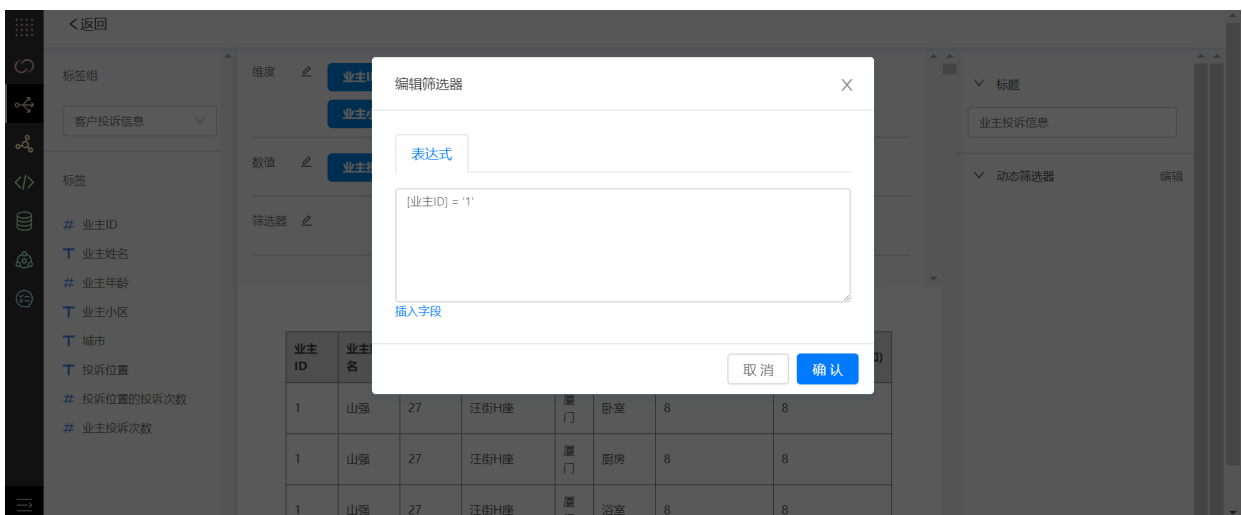
配置如下图所示：



可为图标配置筛选项，点击筛选项编辑按钮，选择标签，点击下一步。



输入表达式，例如：[业主ID]='1'，点击确定。



点击返回，可返回上一级。



12.3.4 操作仪表盘

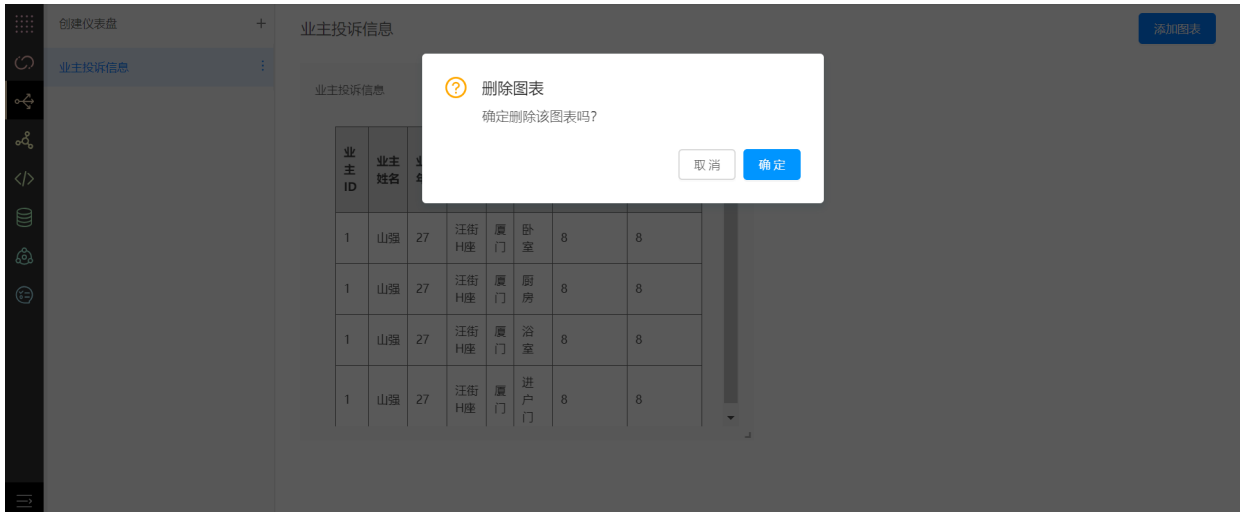
返回上一级后，点击编辑按钮，进入编辑页面，重新配置图表。



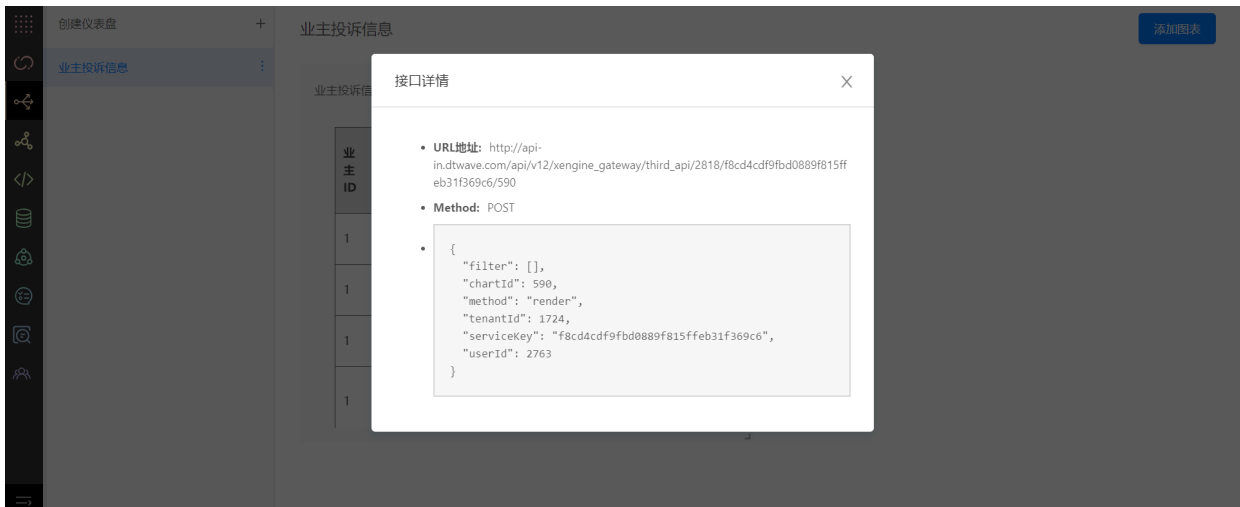
点击更多按钮，显示删除和接口详情按钮。



点击删除按钮，删除数据服务需要二次确认。



点击接口详情，生成该数据服务的API信息，供相应的应用场景使用，如下图所示，即完成数据服务。



13 幻数

13.1 项目管理

13.1.1. 功能说明

项目列表是幻数的核心功能。它包括标签体系和仪表盘两部分。通过标签体系结合可视化图表，生成各种样式的仪表盘。

13.1.2. 创建项目

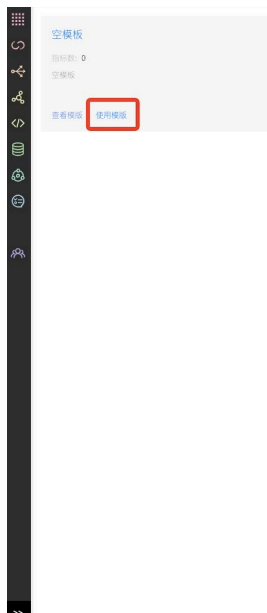
1.在管理控制台中，**点击幻数产品**；或者在左侧菜单栏中，鼠标移入幻数，**点击项目列表**，都可以进入幻数的项目列表页面。



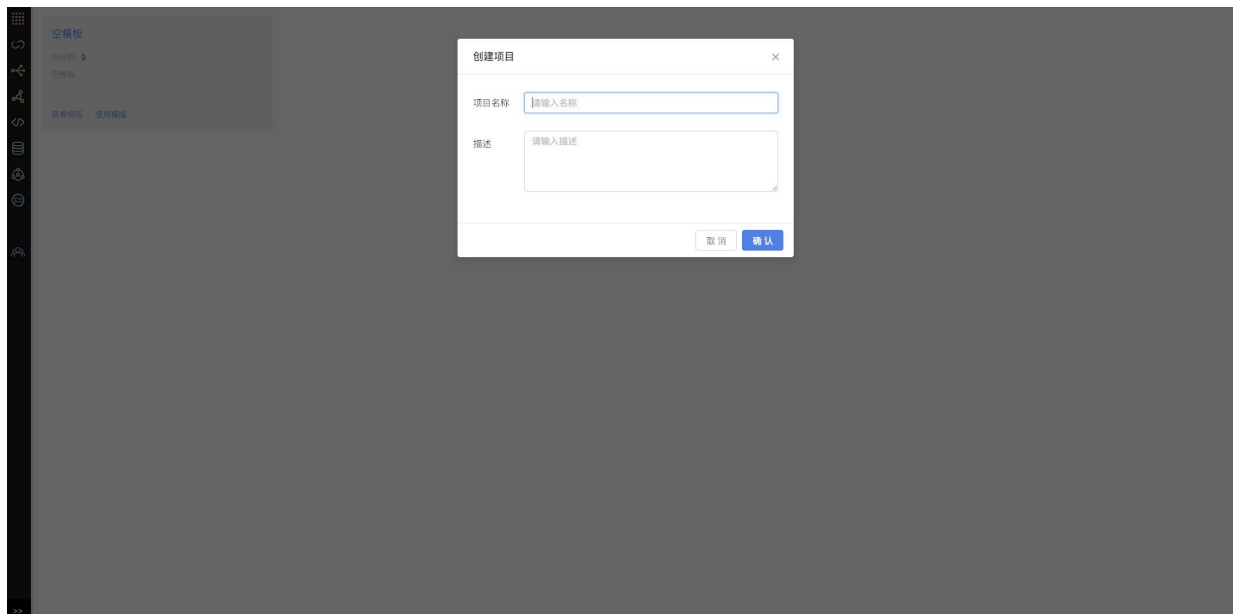
2.在项目列表页面，点击新建项目按钮。



3.选择要使用的模板，可以选择空模板也可以选择非空模板，点击使用模板按钮。



4.选择模板后，需要填写项目名称和描述。填写完成后，点击确定按钮，项目创建成功，可以在项目列表中看到新创建的项目。

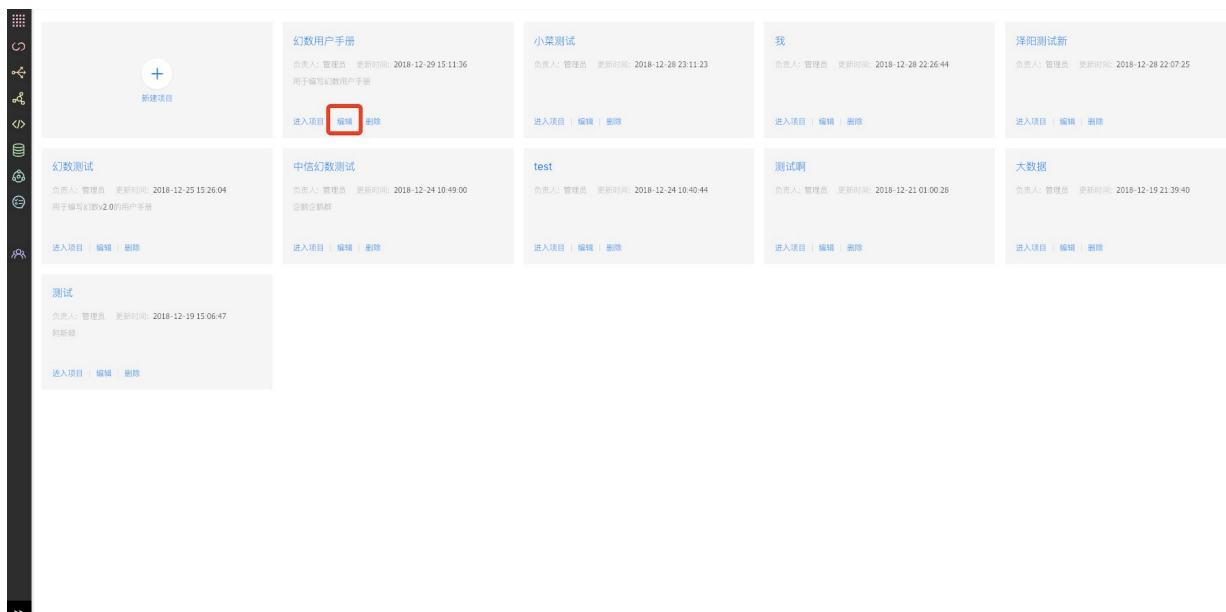


注意：

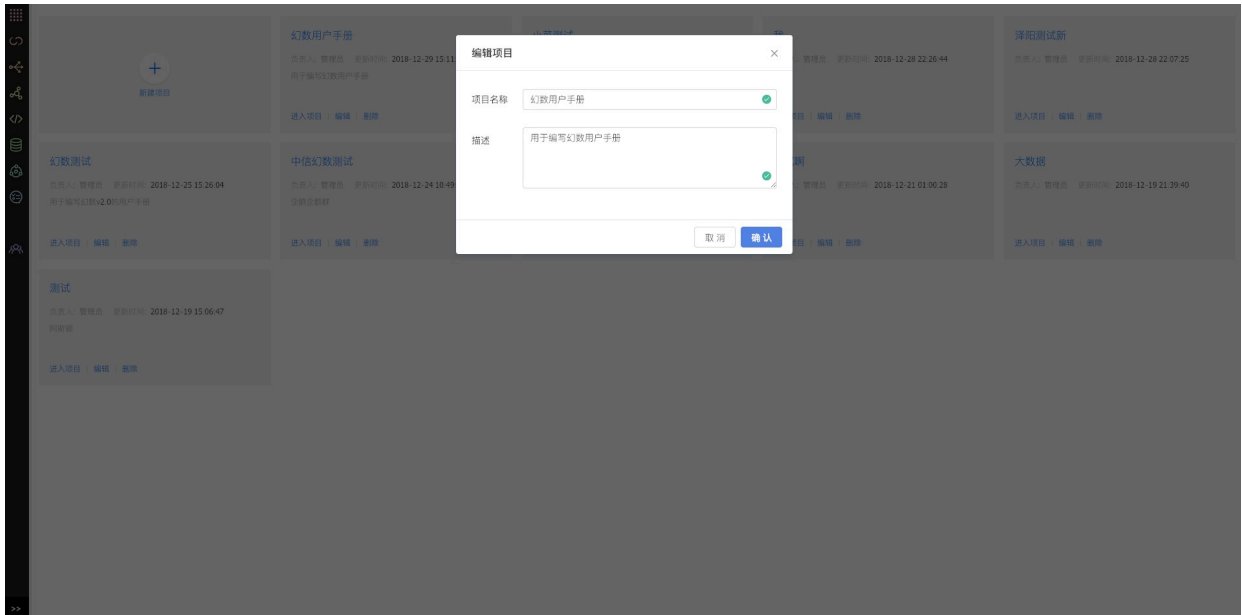
- 1) 使用非空模板，会初始化相应的前后台标签、数据映射关系、数据服务、标签表以及仪表盘。
- 2) 当前只提供空模板。

13.1.3. 编辑项目

1.选择想要编辑的项目，点击编辑按钮。



2.在编辑项目弹窗中，修改相应的信息后，点击确定按钮，编辑项目成功。



13.1.4. 删除项目

1.选择想要删除的项目，点击删除按钮。



2.删除项目需要二次确认，点击确定后，删除项目成功。



注意：

- 1) 项目删除后，该项目下的标签表和仪表盘均会被删除。

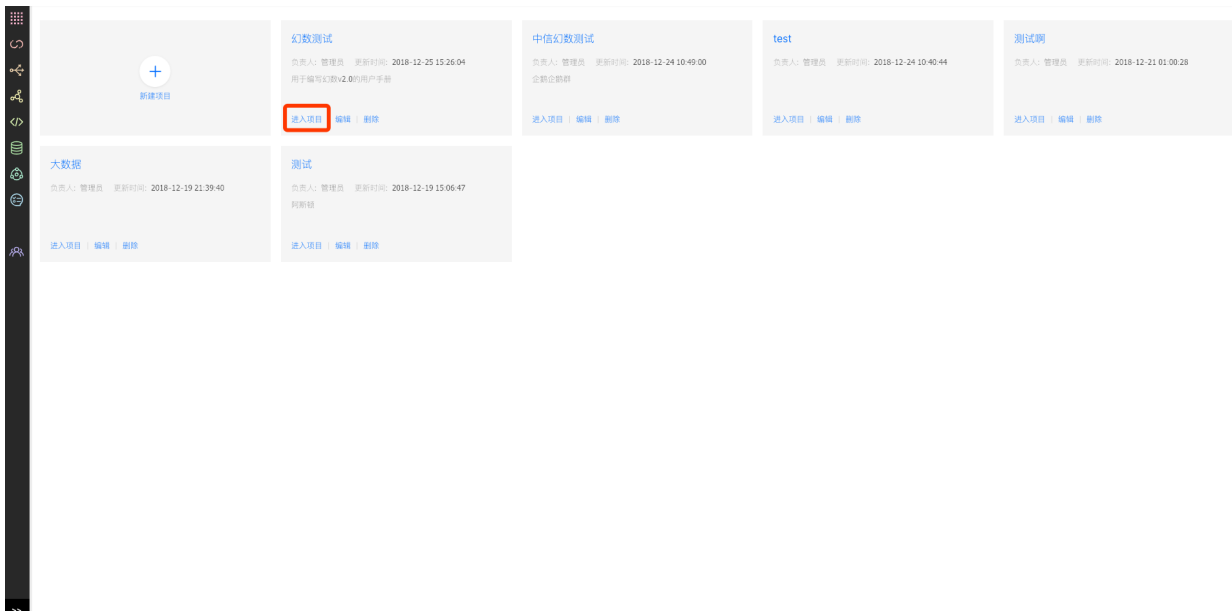
13.2 标签表

13.2.1 功能说明

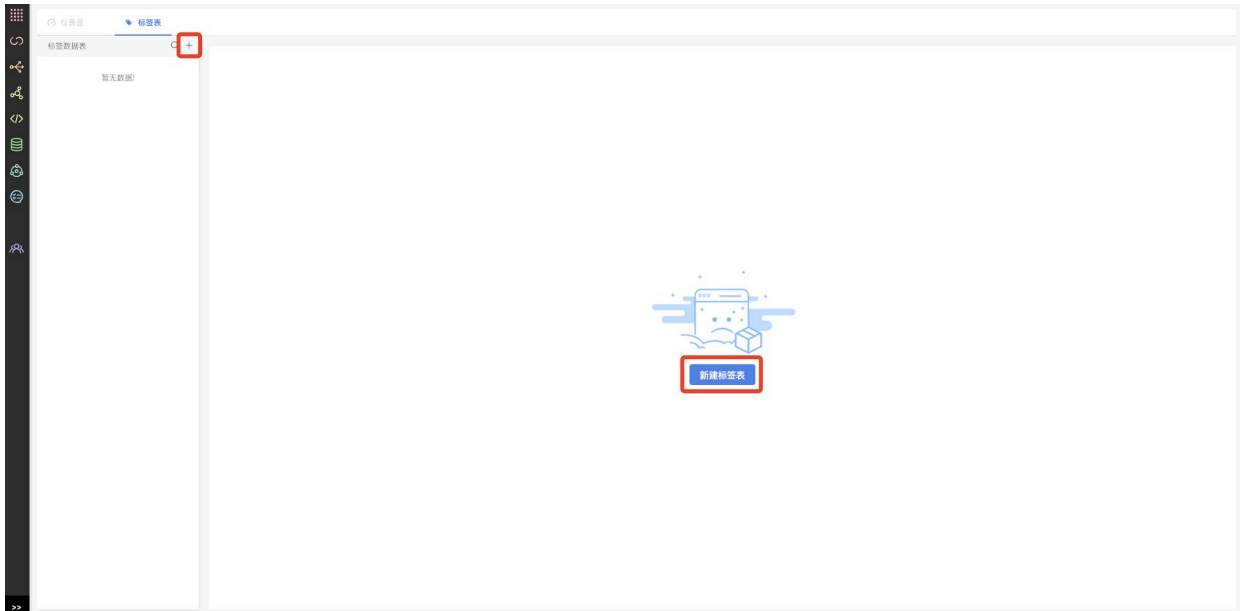
标签表是图表可视化的基础。幻数中的标签表来自于平台数据服务模块的API透出。

13.2.2 添加标签数据表

1. 选择一个要操作的项目，点击进入项目按钮。



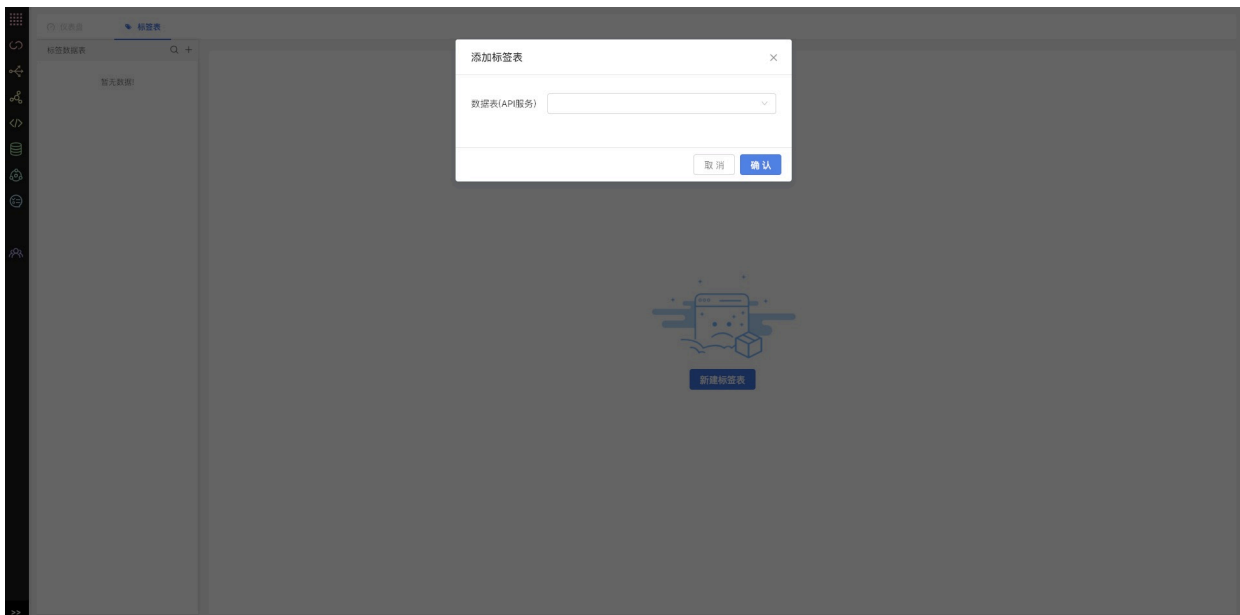
2. 选择标签表选项卡，点击添加标签表按钮。



注意：

1) 如果该项目没有添加标签表，则进入项目时，默认进入标签表页面，且仪表盘选项卡置灰不可点击。只有标签表添加之后，仪表盘才可点击。

3.选择数据服务，点击确定后，添加标签表成功。

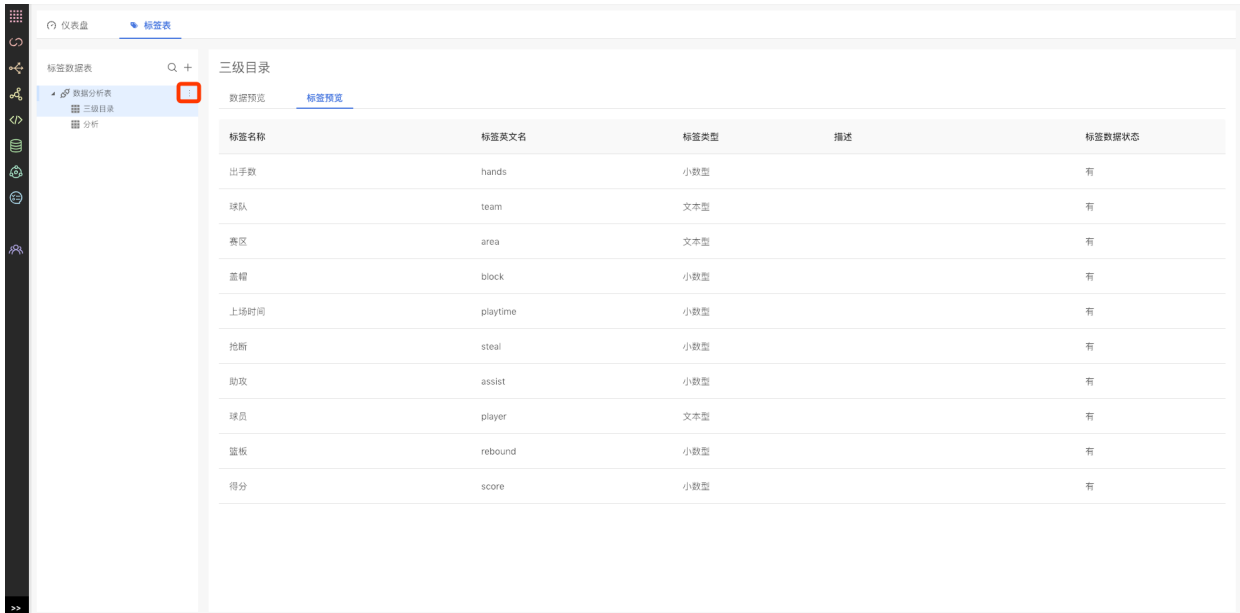


注意：

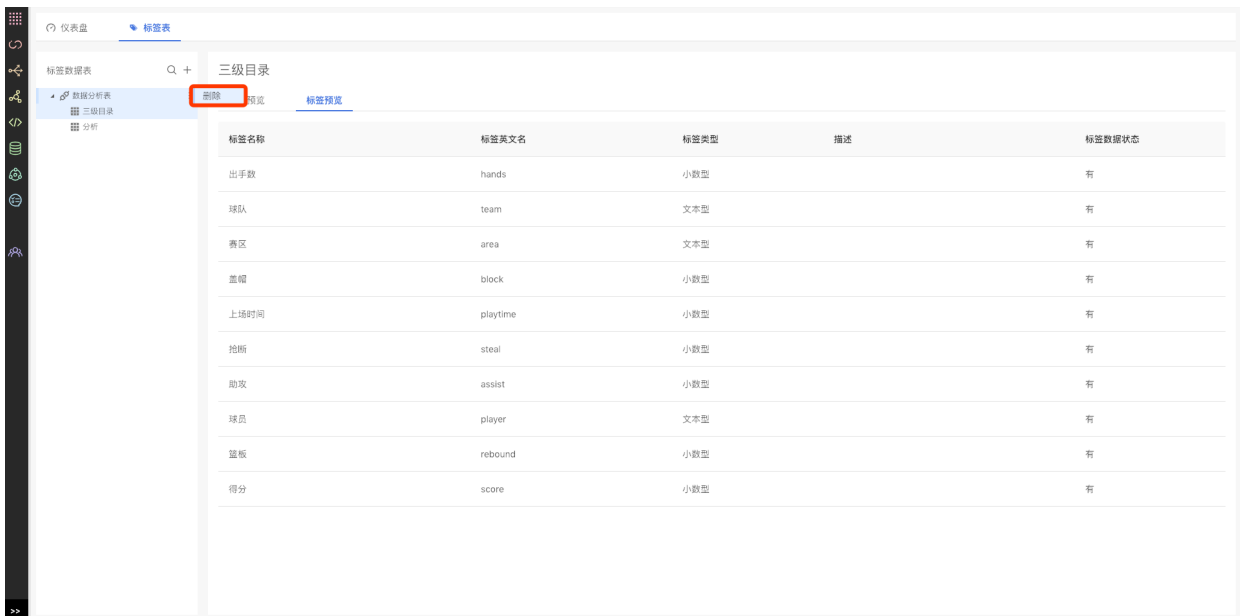
1) 数据服务添加后，该服务下的所有标签表会添加到该项目。

13.2.3 删除标签数据表

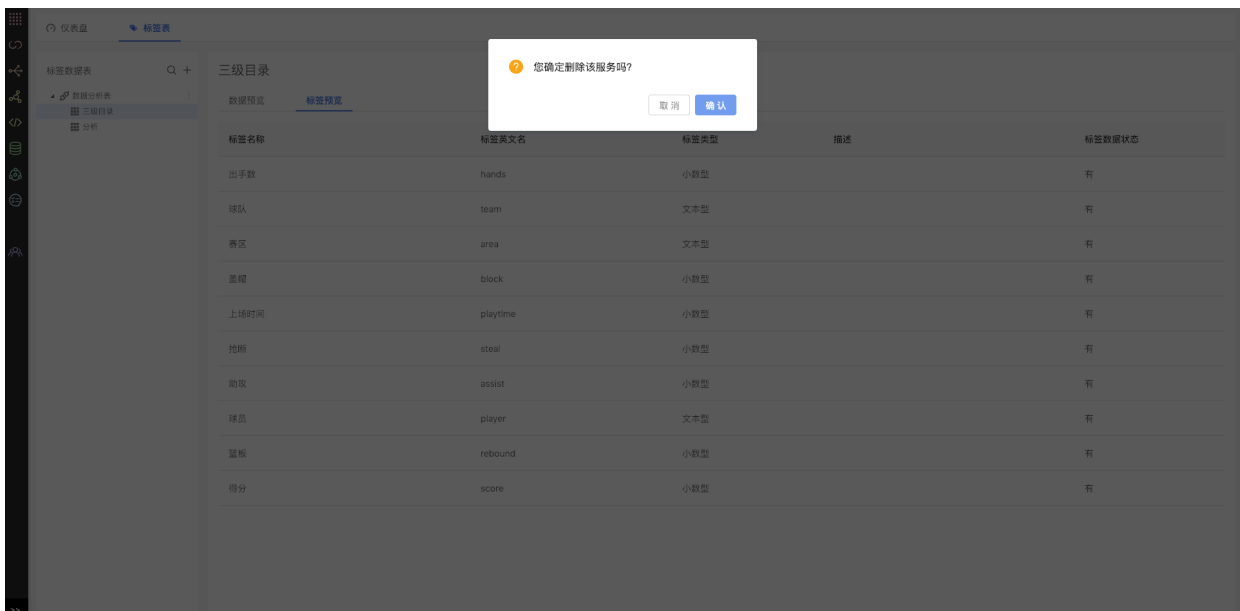
1.鼠标移入要删除的数据服务，点击**更多按钮**。



2. 点击删除按钮。



3. 删除服务需要二次确认，点击确定按钮后，删除服务成功，该服务下的标签表也一起删除。

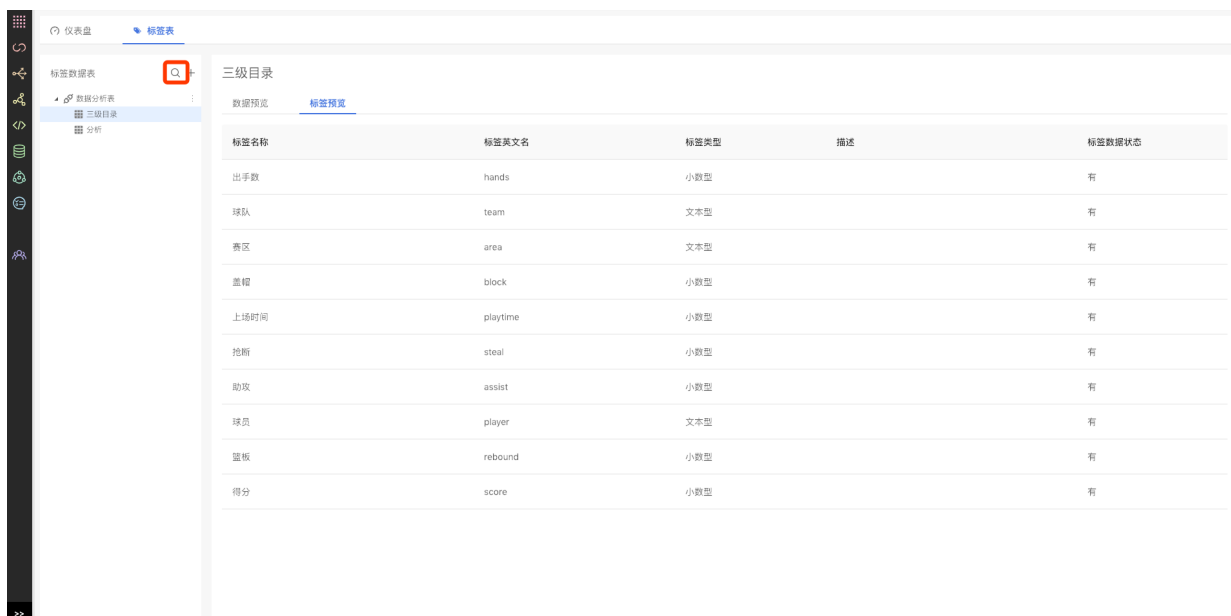


注意：

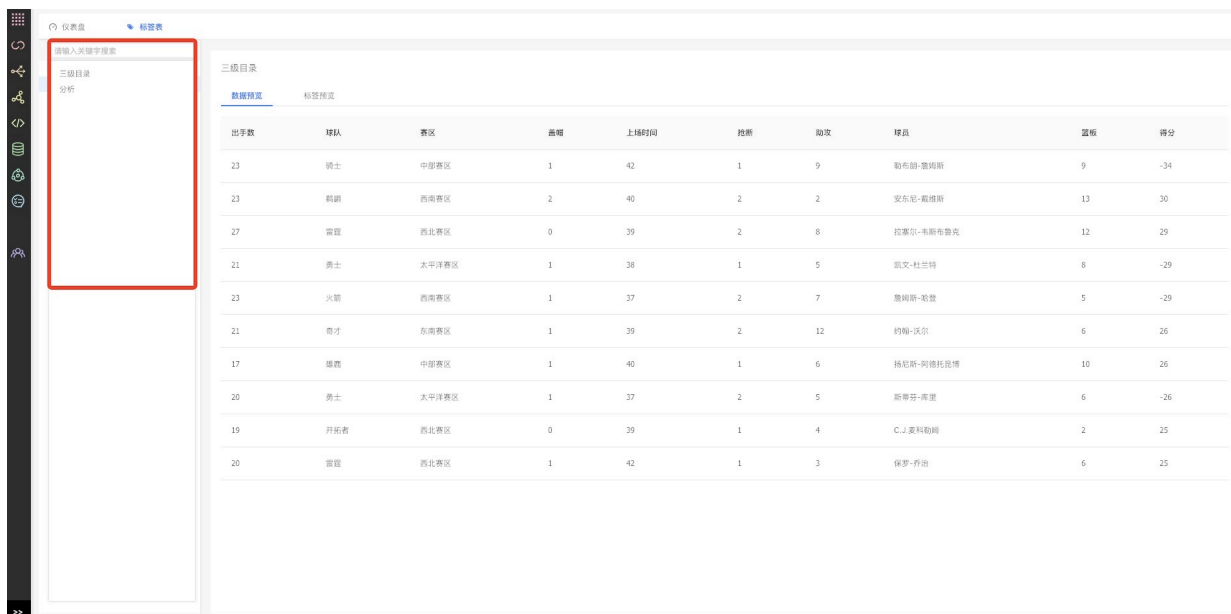
- 1) 数据服务删除后，该服务下的所有标签表会同时删除。
- 2) 若该服务下的标签表有被图表在使用，则无法删除该服务，需要先删除关联的图表。

13.2.4 搜索标签数据表

1. 在标签表列表中，点击**搜索按钮**。



2. 在搜索框中输入要查找的标签表，搜索支持模糊匹配，选择对应标签表名称，查看相应的标签表。



13.2.5 数据预览

1. 在标签表选项卡下，选择要查看的标签表。
2. 选择数据预览选项卡，就可以预览当前标签数据表下的数据。

仪表盘 > 标签表

标签数据表 数据目录 分析

三级目录 数据预览

出手数	球队	赛区	赛期	上报时间	抢断	助攻	球员	篮板	得分
23	骑士	中部赛区	1	42	1	9	勒布朗-詹姆斯	9	-34
23	鹈鹕	西南赛区	2	40	2	2	安东尼-戴维斯	13	30
27	雷霆	西北赛区	0	39	2	8	拉塞尔-威斯布鲁克	12	29
21	勇士	太平洋赛区	1	38	1	5	凯文-杜兰特	8	-29
23	火箭	西南赛区	1	37	2	7	詹姆斯-哈登	5	-29
21	奇才	东南赛区	1	39	2	12	约翰-沃尔	6	26
17	雄鹿	中部赛区	1	40	1	6	扬尼斯-阿德托昆博	10	26
20	勇士	太平洋赛区	1	37	2	5	斯蒂芬-库里	6	-26
19	开拓者	西北赛区	0	39	1	4	C.J.麦科勒姆	2	25
20	雷霆	西北赛区	1	42	1	3	保罗-乔治	6	25

注意：

- 1) 数据预览最多展示10条记录。

13.2.6. 标签预览

1. 在标签表选项卡下，选择要查看的标签表。
2. 选择标签预览选项卡，就可以预览当前标签数据表下的标签。

仪表盘 > 标签表

标签数据表 数据目录 分析

三级目录 数据预览 标签预览

标签名称	标签英文名	标签类型	描述	标签数据状态
出手数	hands	小数值		有
球队	team	文本型		有
赛区	area	文本型		有
赛期	block	小数值		有
上报时间	playtime	小数值		有
抢断	steal	小数值		有
助攻	assist	小数值		有
球员	player	文本型		有
篮板	rebound	小数值		有
得分	score	小数值		有

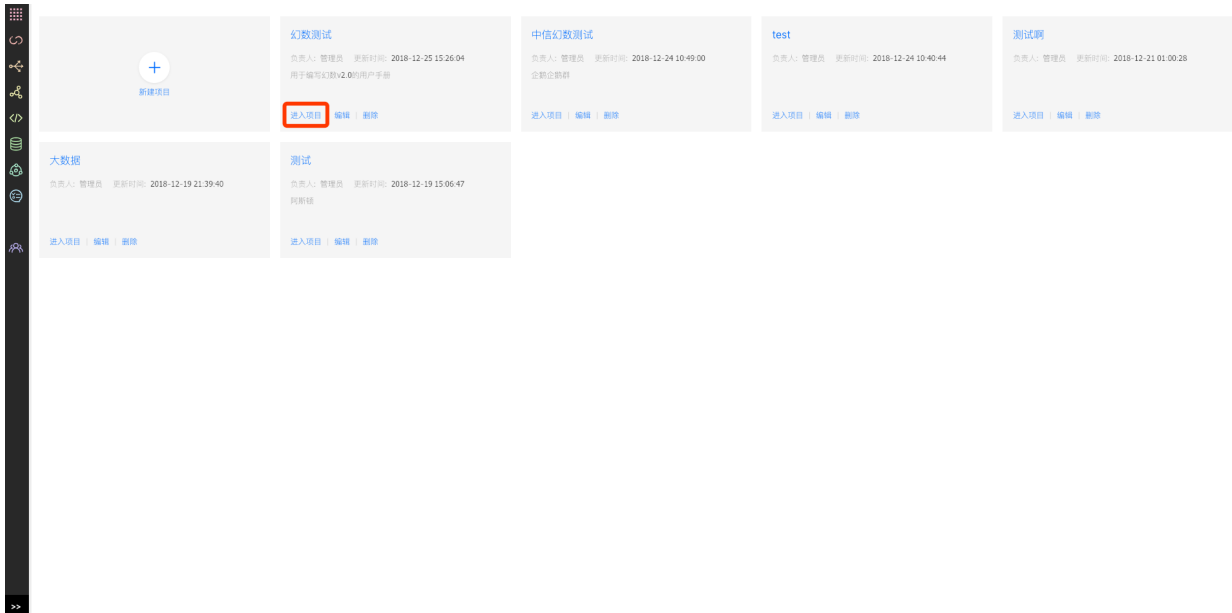
13.3 仪表盘

13.3.1 功能说明

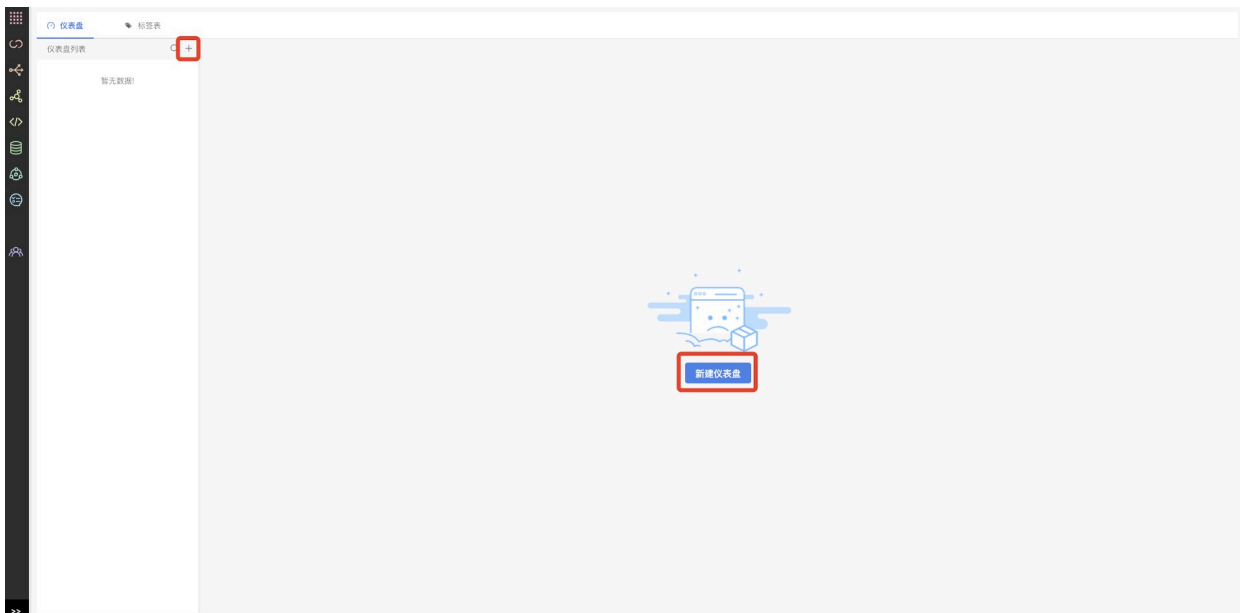
幻数提供丰富的图表组件和个性化配置项，可以帮助用户生成满足企业分析所需的仪表盘，指导企业数据化运营。

13.3.2 添加仪表盘

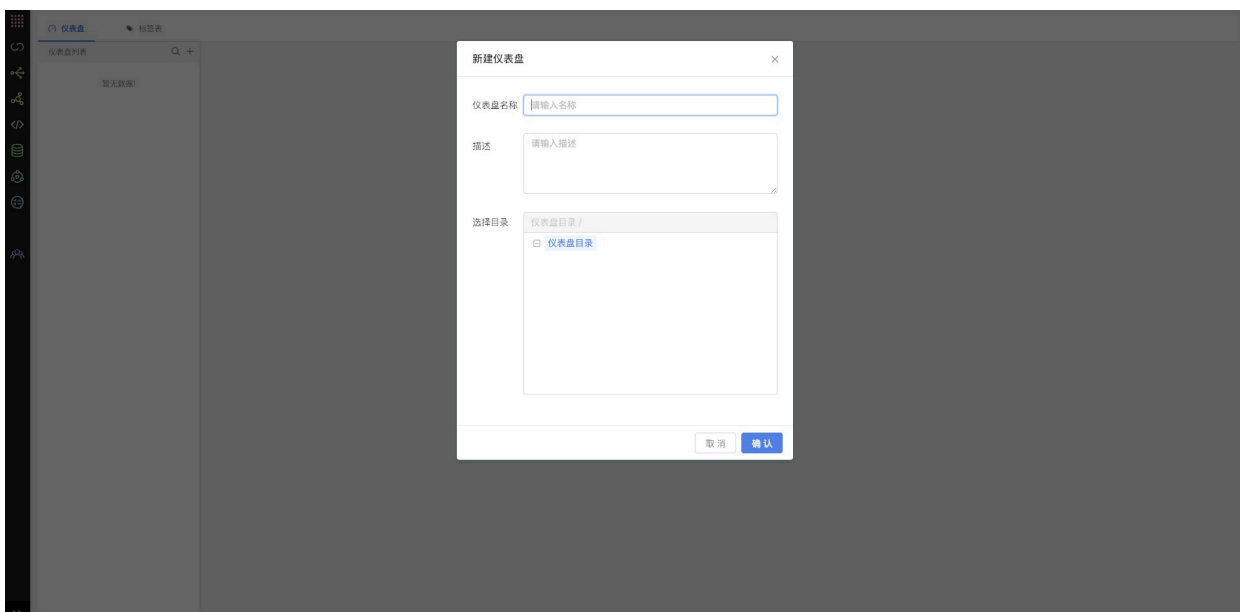
1. 选择一个要操作的项目，点击进入项目按钮。



2.在仪表盘页面，点击**新建仪表盘按钮**。



3.在新建仪表盘弹窗中，**填写名称和描述，选择目录后，点击确定按钮**，添加仪表盘成功。

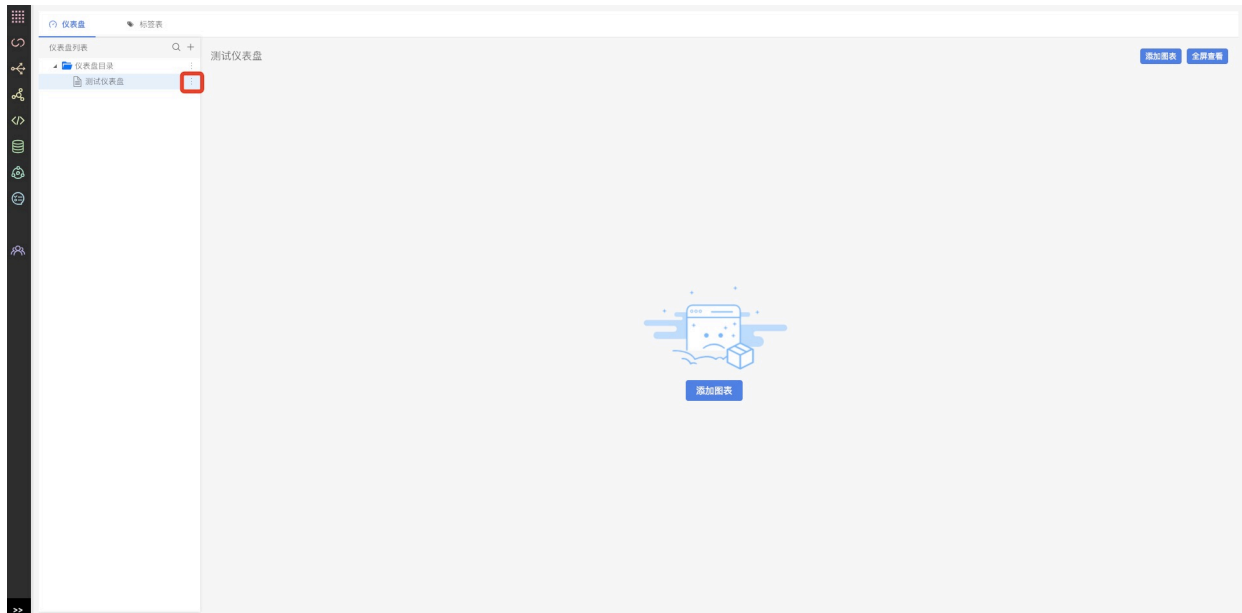


注意：

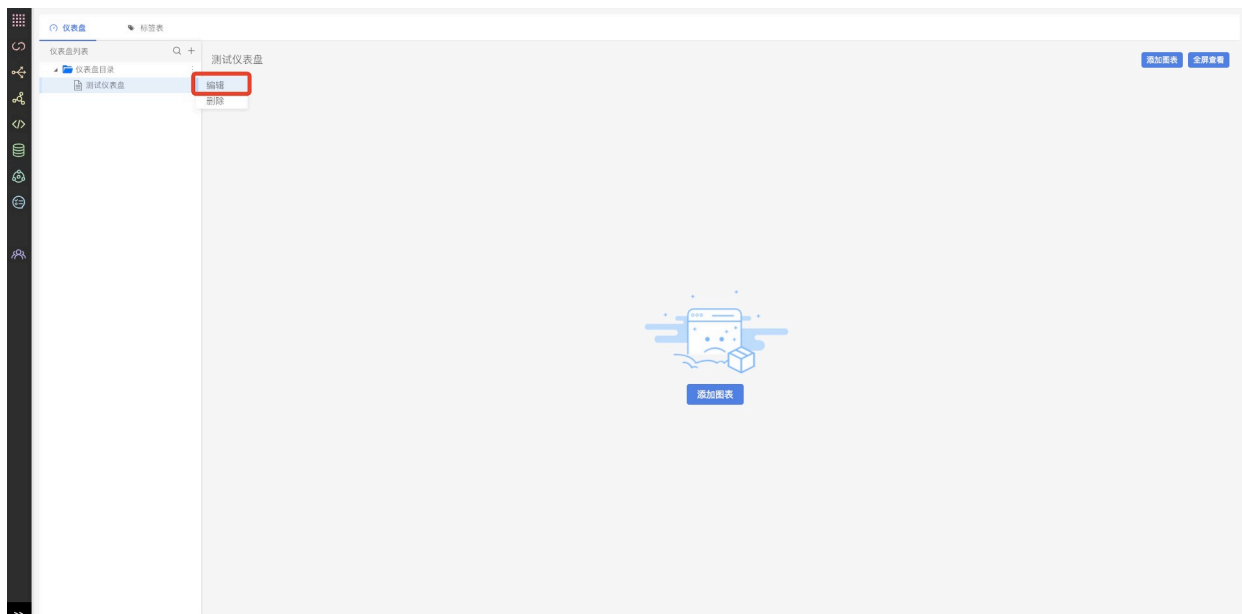
- 1) 仪表盘只有在标签表中添加了标签表之后才可以使用。
- 2) 第一次添加仪表盘或者添加目录，都会在仪表盘根目录下添加。

13.3.3 编辑仪表盘

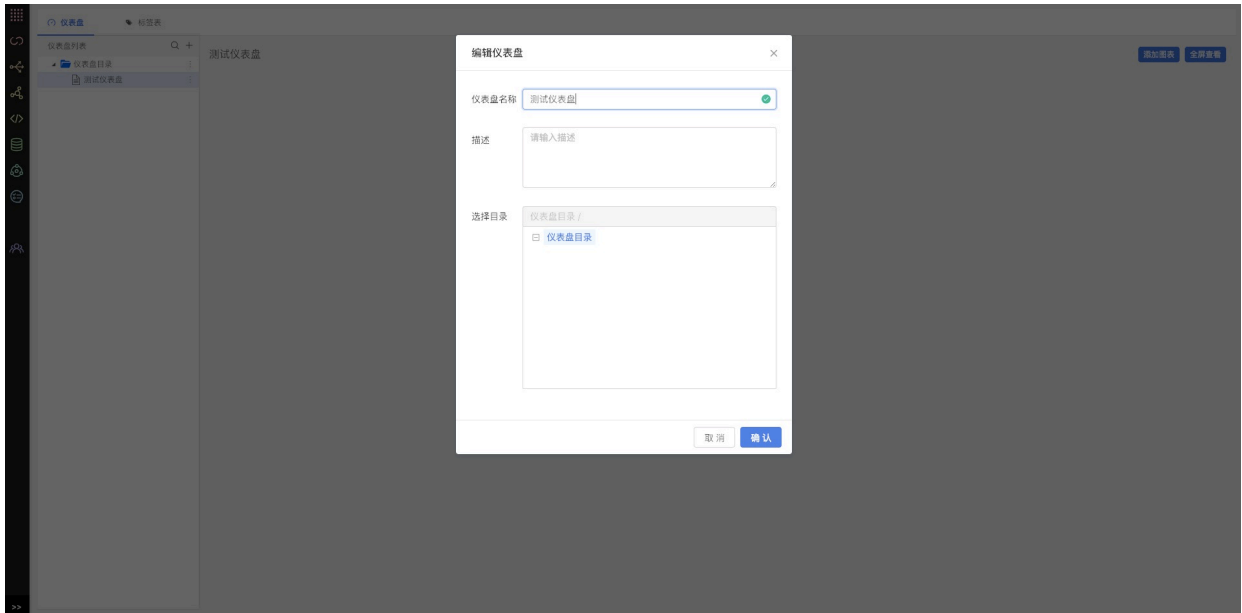
1.在仪表盘页面，选择要编辑的仪表盘，点击**更多按钮**。



2.点击**编辑按钮**。

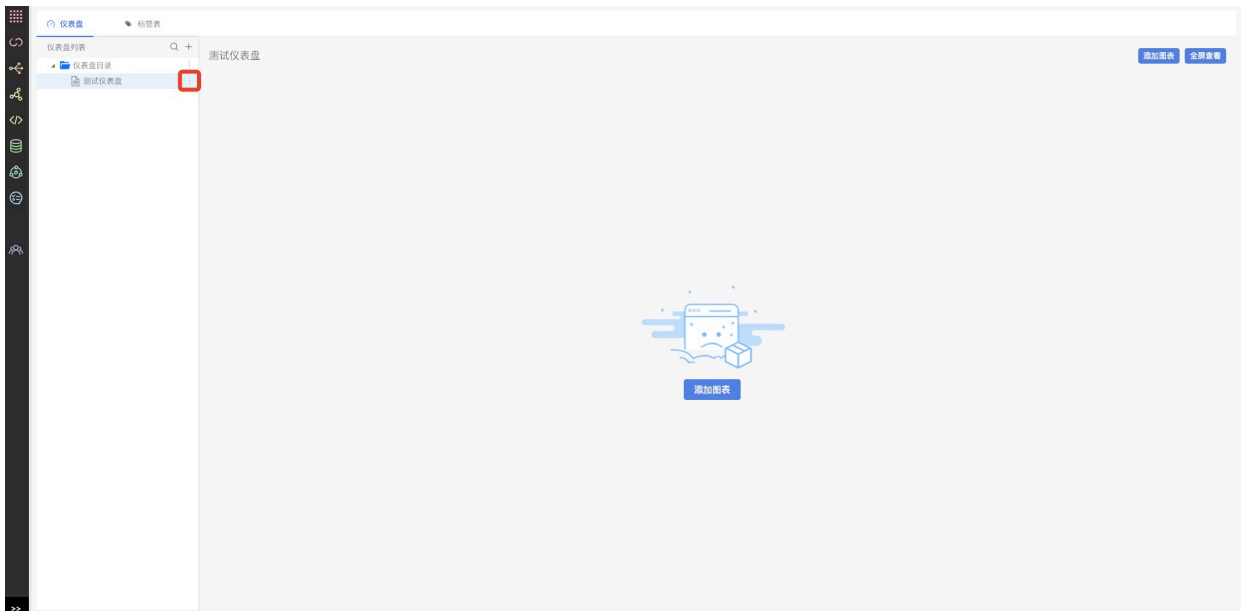


3.在编辑仪表盘弹窗中，填写要修改的信息后，点击**确定按钮**，编辑仪表盘成功。

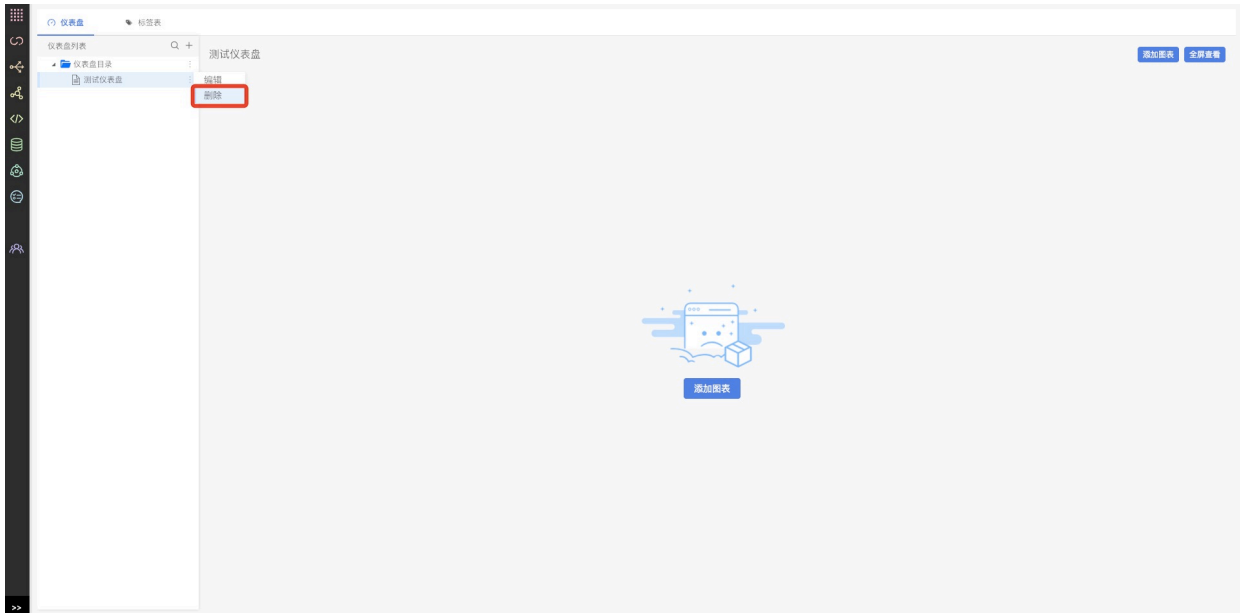


13.3.4 删除仪表盘

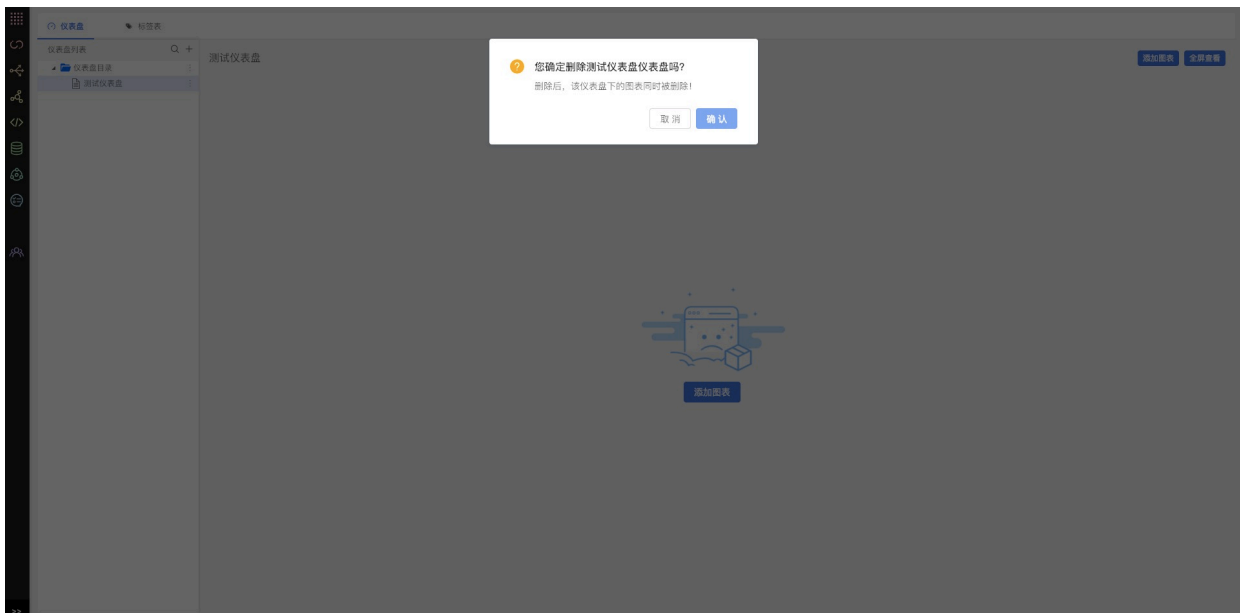
1. 在仪表盘页面，选择要删除的仪表盘，点击**更多**按钮。



2. 点击**删除**按钮。



3.删除仪表盘需要二次确认，点击确定后，仪表盘删除成功。

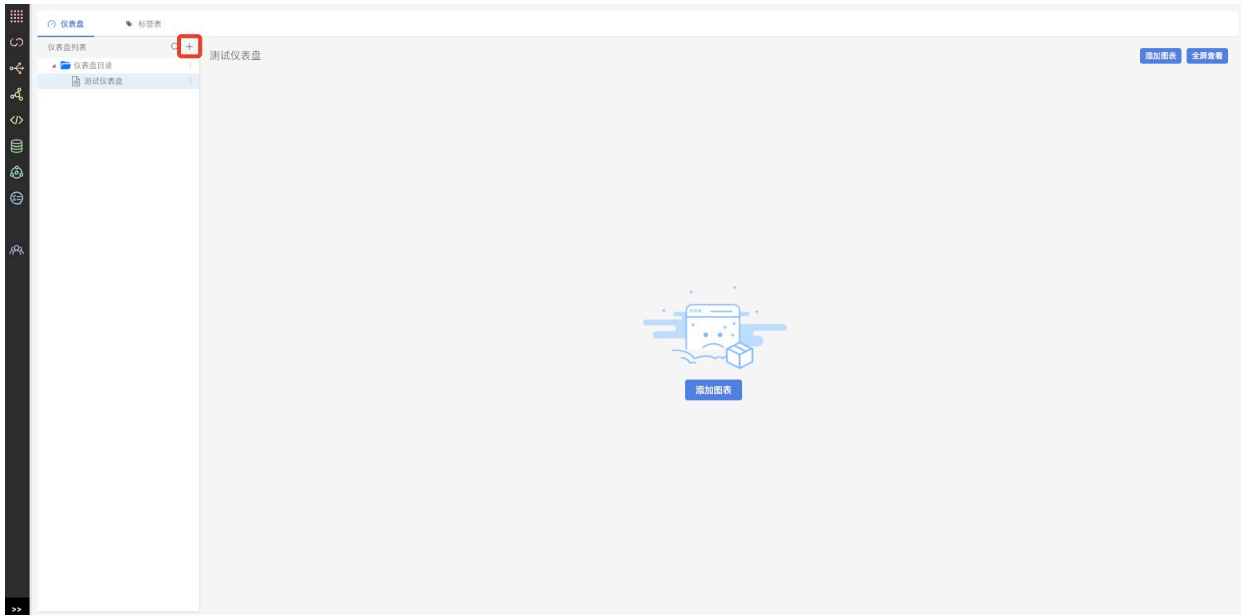


注意：

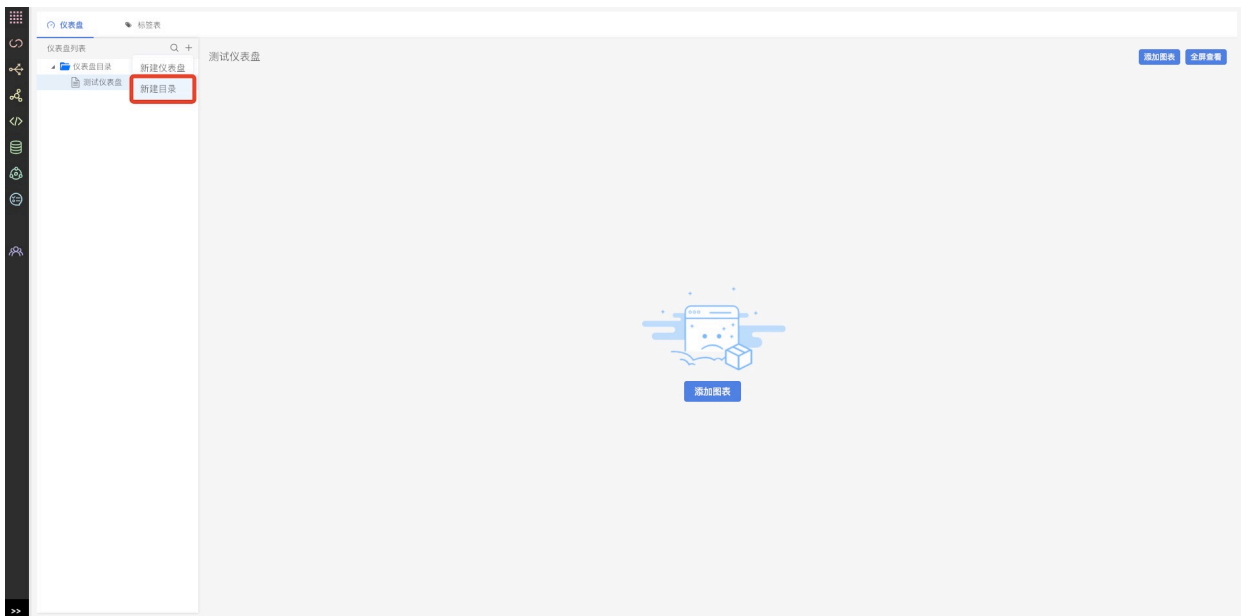
- 1) 仪表盘删除后，仪表盘下图表也会被删除。

13.3.5 添加目录

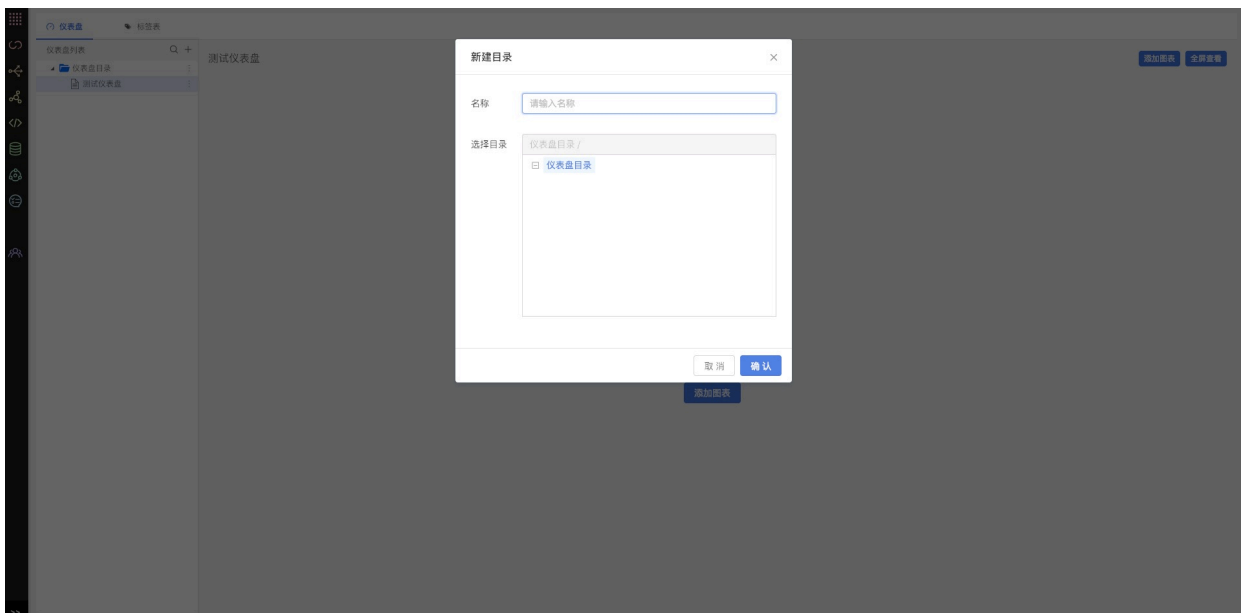
- 1.在仪表盘页面，点击添加按钮。



2. 点击新建目录按钮。



3. 在新建目录弹窗中，填写名称，选择目录后，点击确定按钮，添加目录成功。

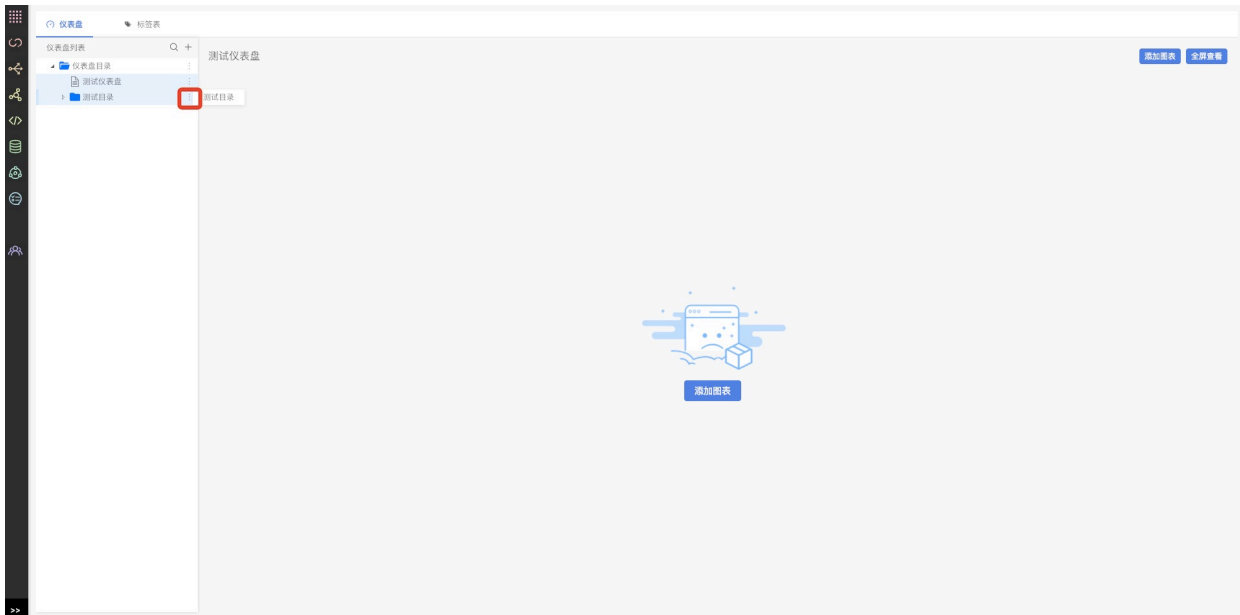


注意：

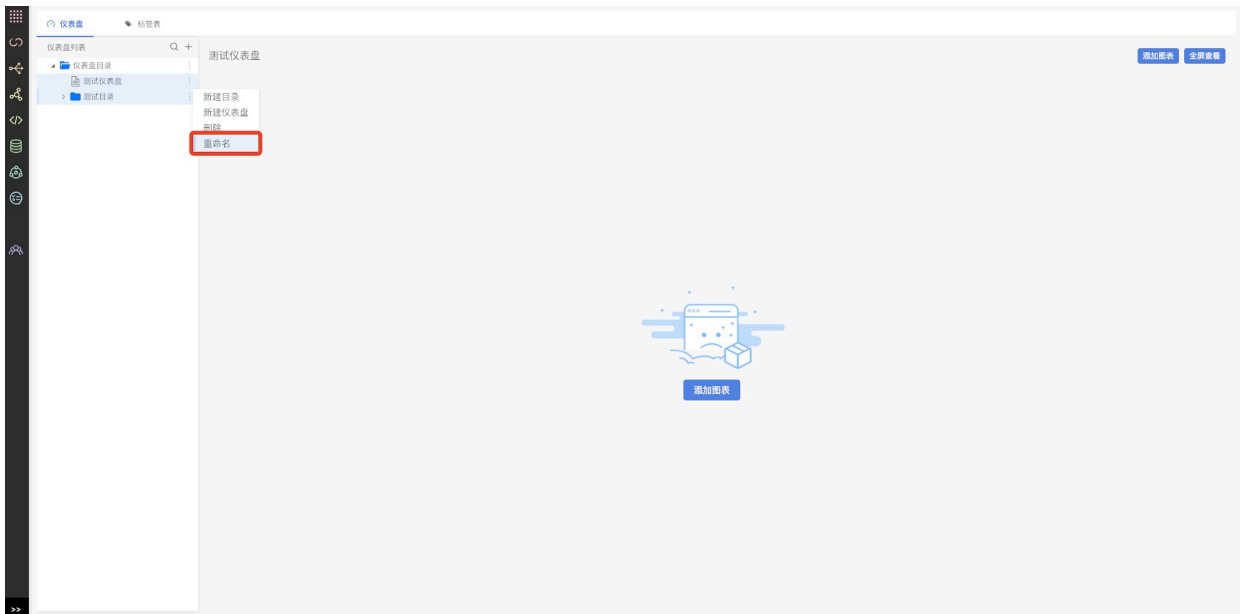
- 1) 根目录无法重命名。

13.3.6 重命名目录

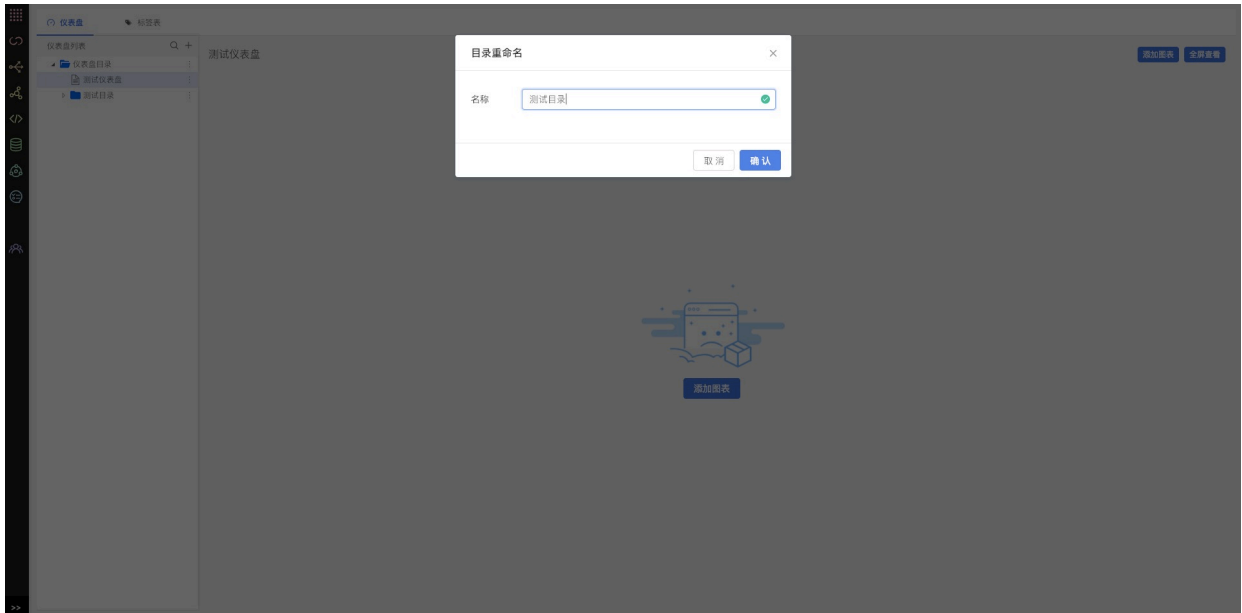
- 1.在仪表盘页面，选择要重命名的目录，点击**更多按钮**。



- 2.点击**重命名按钮**。

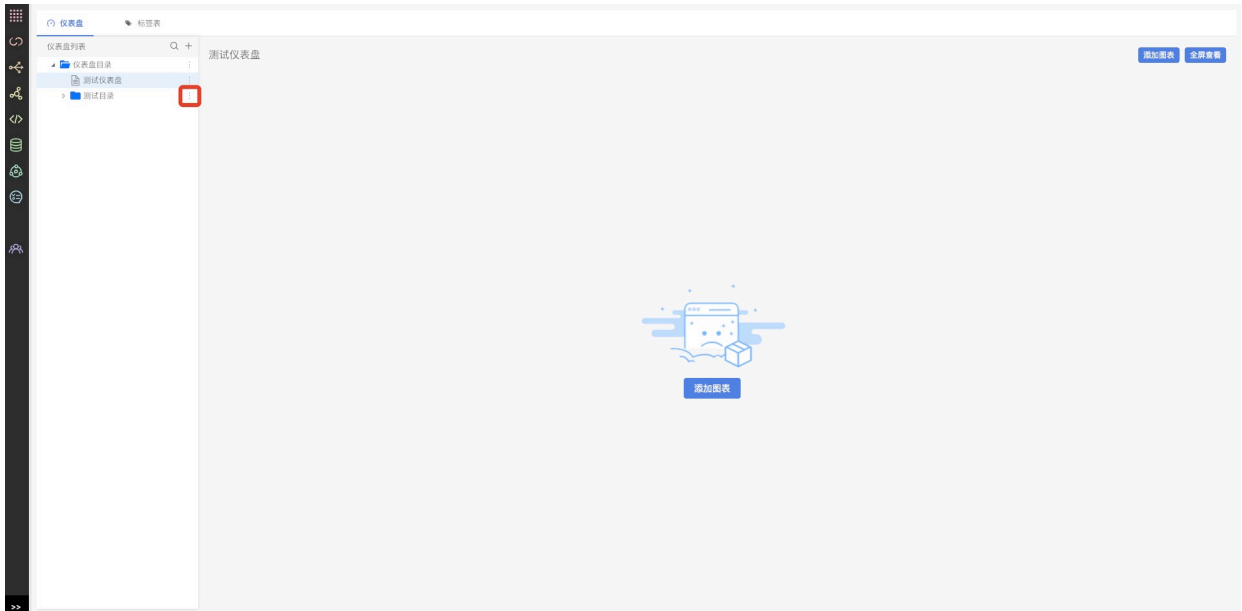


- 3.在目录重命名弹窗中，填写要修改的名称后，点击**确定按钮**，重命名目录成功。

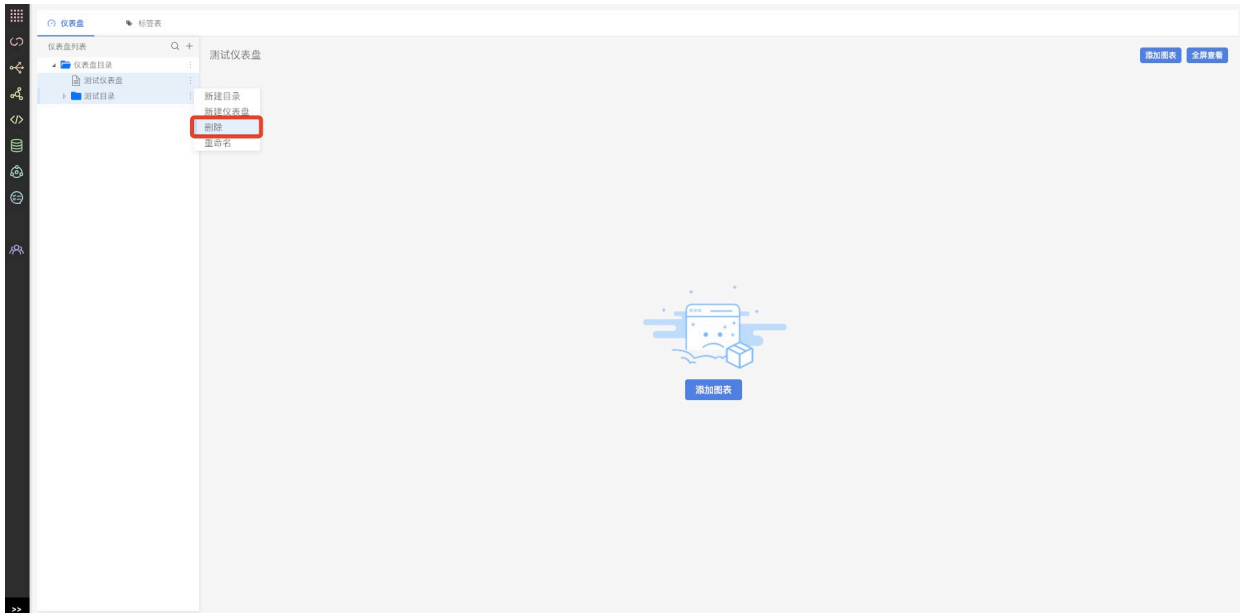


13.3.7 删除目录

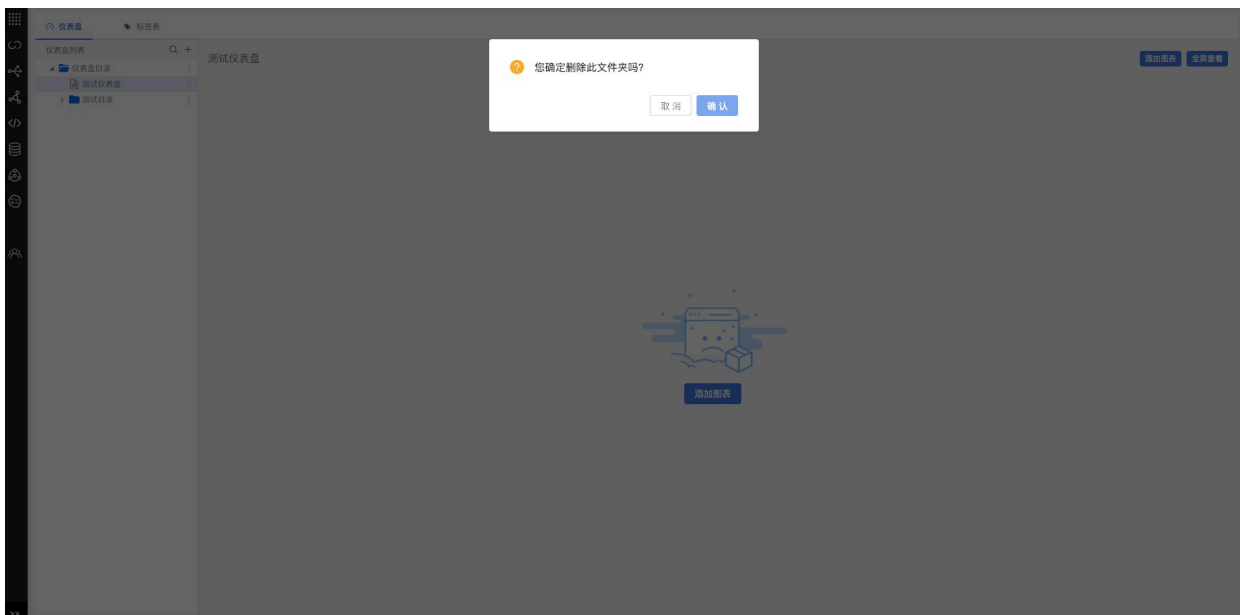
1. 在仪表盘页面，选择要删除的目录，点击**更多**按钮。



2. 点击**删除**按钮。



3.删除目录需要二次确认，点击确定后，目录删除成功。

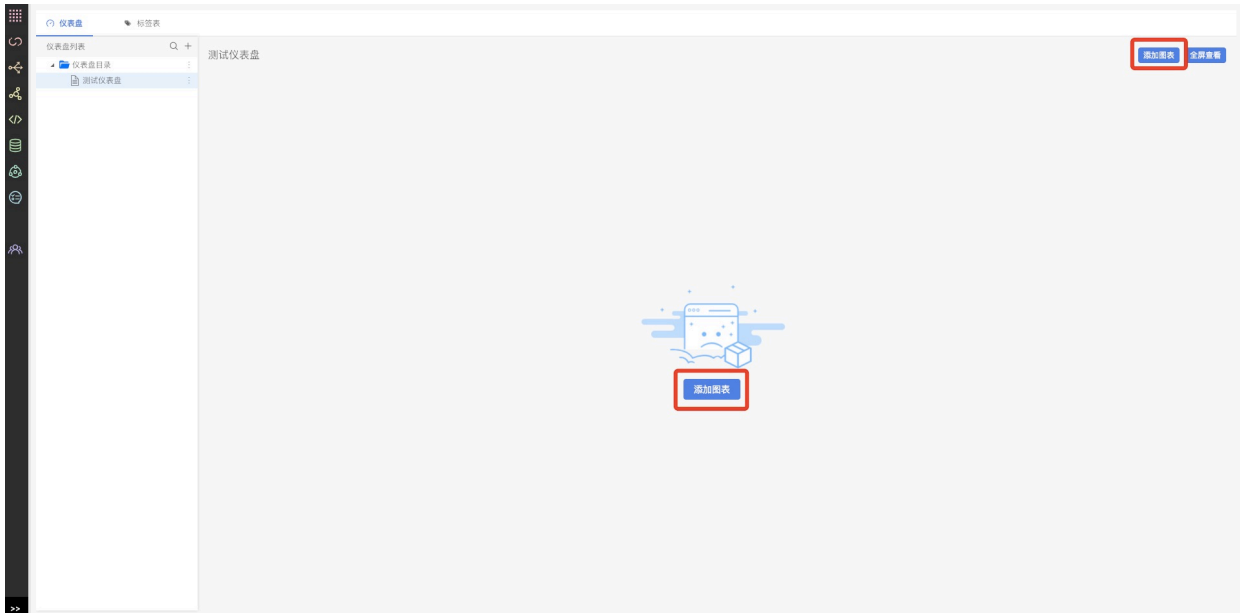


注意：

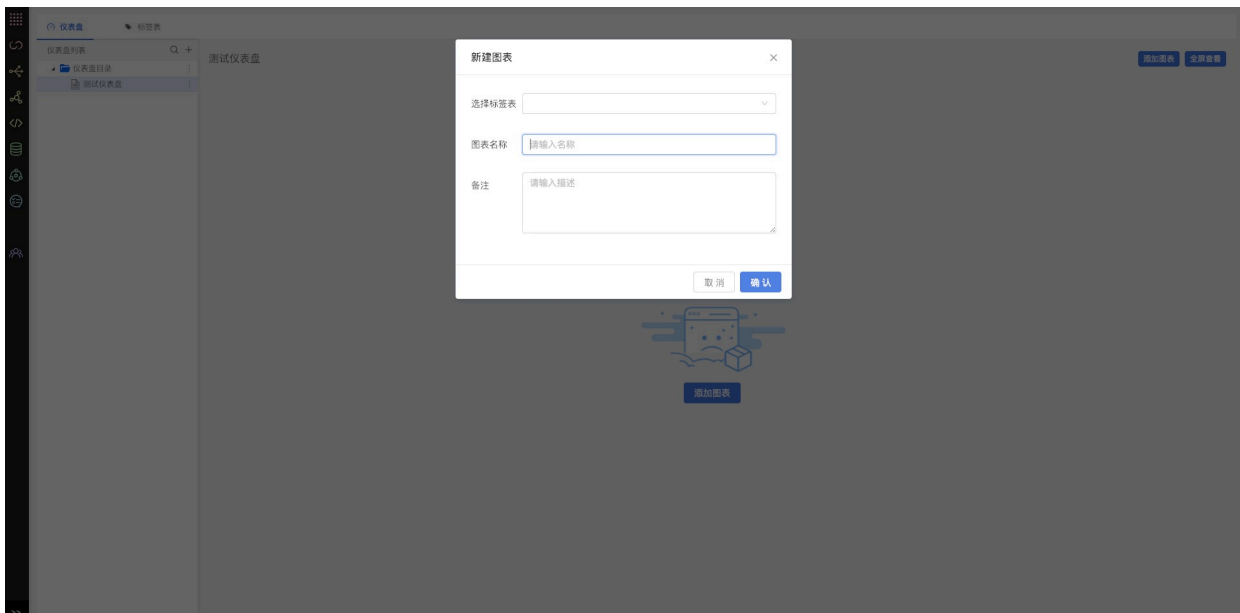
1) 目录下有仪表盘存在时，不允许删除该目录。

13.3.8 添加图表

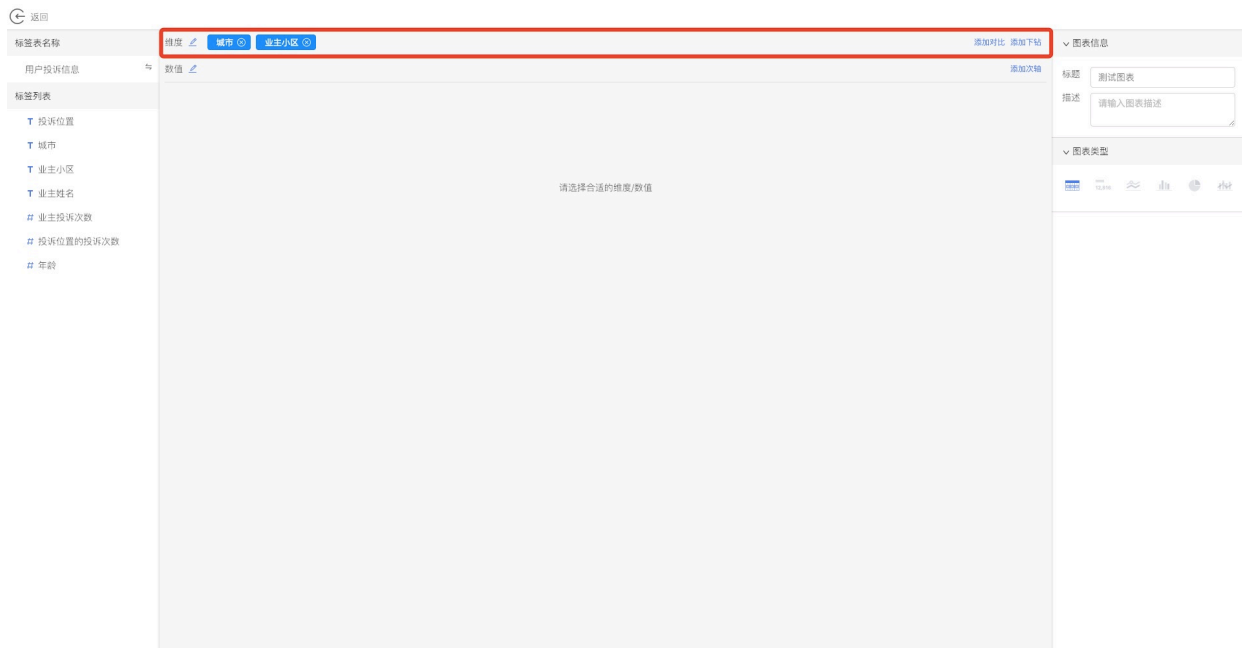
1.在仪表盘页面，选择一张仪表盘，点击添加图表按钮。



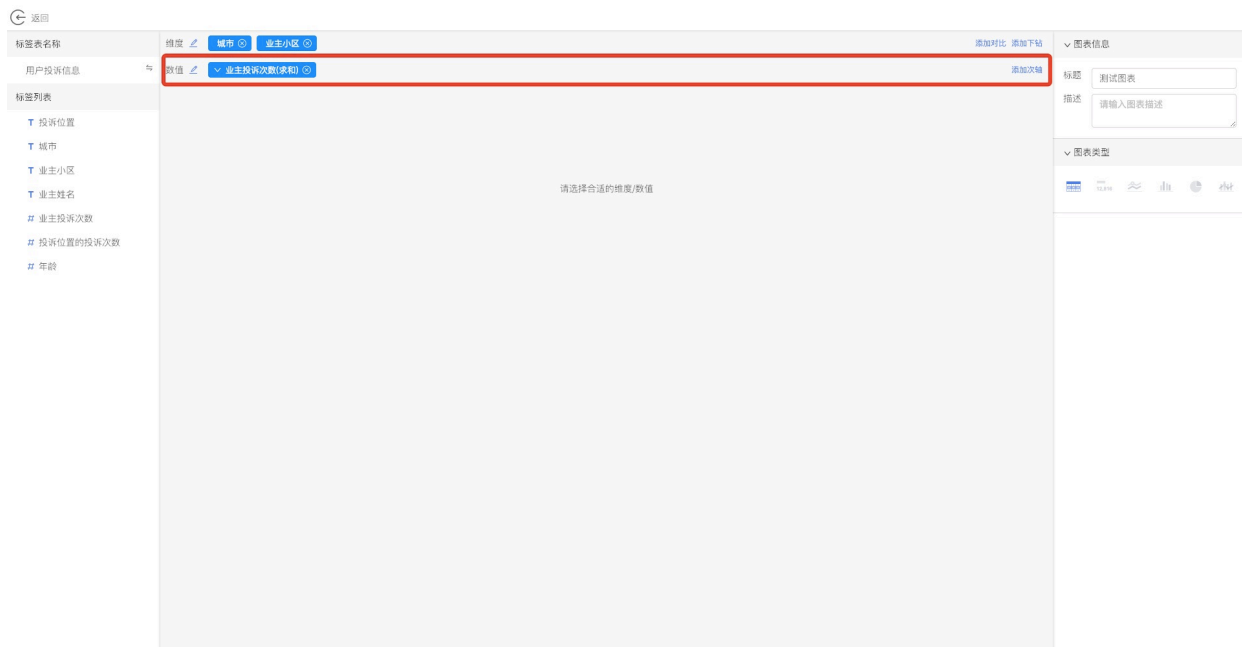
2.选择要使用的标签表，填写图表名称和备注后，点击确认按钮。



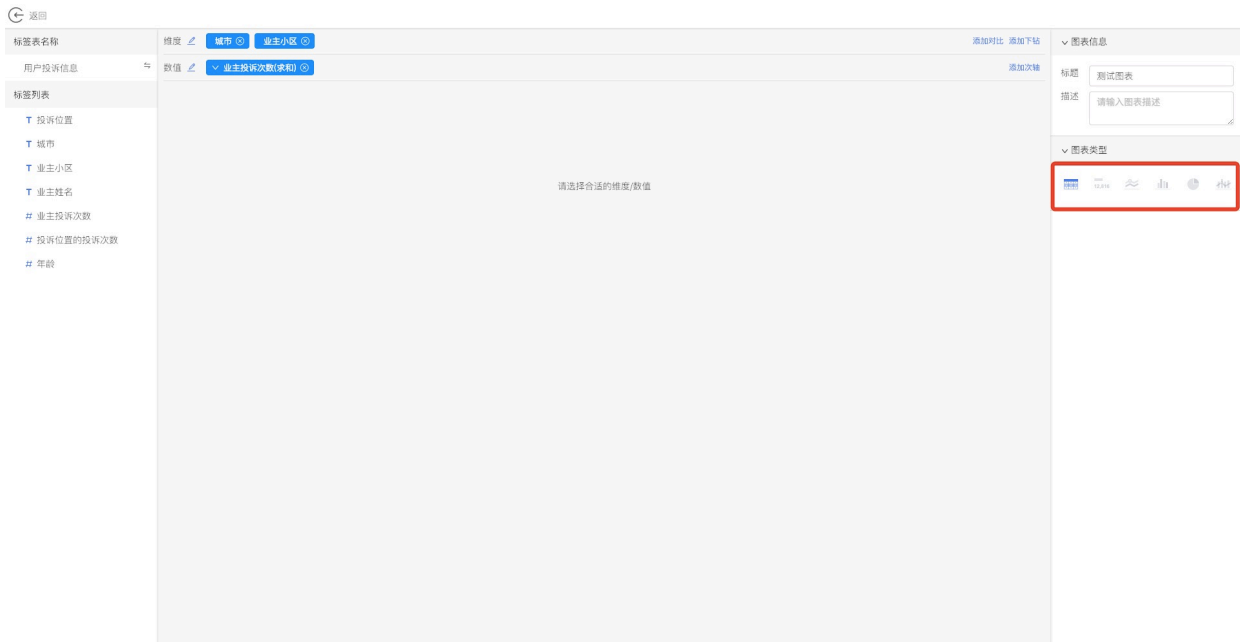
3.从标签列表中选择 一个标签，拖入到维度栏。维度通常是一个不连续的分类字段。



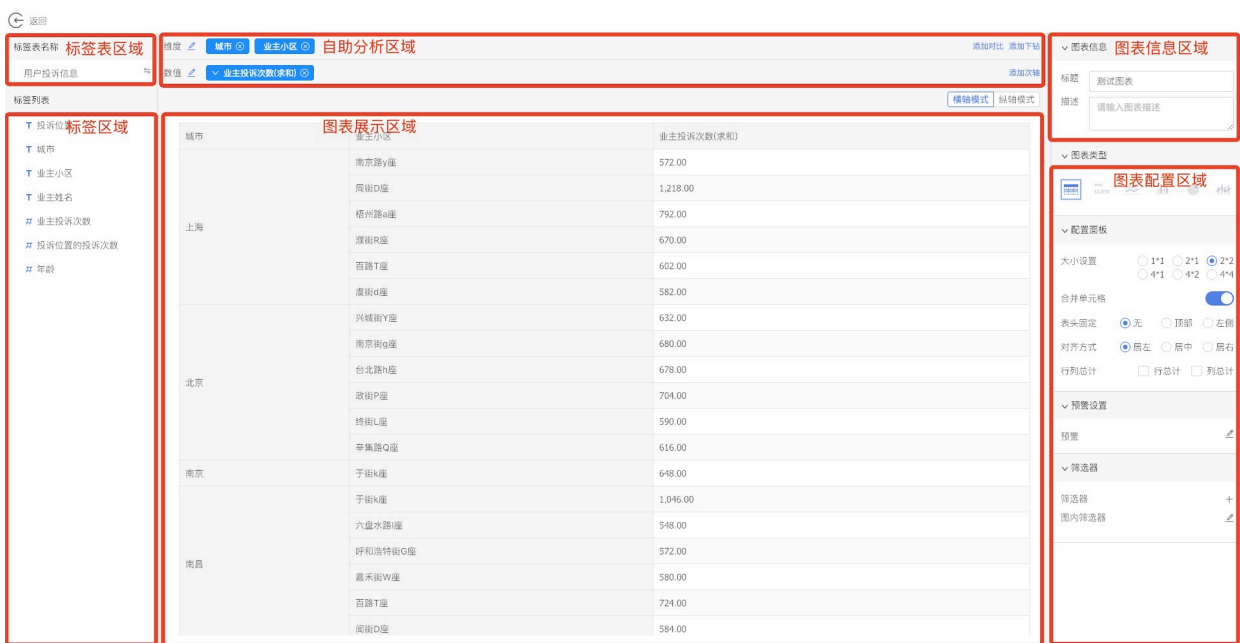
4.从标签列表中选择**一个**标签，拖入到数值栏。数值通常是一个连续的字段。



5.在右侧配置区域**选择**图表一种图表类型。



6.在展现区可以看到已经生成的图表，在配置区根据图表提供的个性化配置项进行调整。



7.点击头部返回按钮，返回仪表盘页面就可以看到配置好的图表。

返回
添加对比 添加下钻

标签表名称

三级目录

标签列表

- T 球员
- T 赛区
- T 球队
- # 得分
- # 篮板
- # 助攻
- # 抢断
- # 上场时间
- # 盖帽
- # 出手数

维度 赛区 球队 球员

数值 出手数(求和) 盖帽(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和)

图内筛选器 请选择赛区 横轴模式 纵轴模式

赛区	球队	球员	出手数(求和)	盖帽(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)
火箭		PJ-塔克	6.00	1.00	34.00	1.00	1.00	7.00
		克林特-卡佩拉	9.00	2.00	31.00	1.00	1.00	12.00
		克里斯-保罗	17.00	0.00	35.00	2.00	6.00	6.00
		埃里克-戈登	13.00	1.00	32.00	1.00	2.00	3.00
		特雷沃-阿里扎	8.00	0.00	34.00	1.00	1.00	4.00
		詹姆斯-哈登	23.00	1.00	37.00	2.00	7.00	5.00
		帕蒂-米尔斯	11.00	0.00	33.00	1.00	3.00	2.00
西南赛区	马刺	德章泰-穆雷	6.00	0.00	19.00	1.00	2.00	4.00
		拉马库斯-阿尔德里奇	16.00	0.00	35.00	1.00	2.00	9.00
		马努-吉诺比利	7.00	0.00	21.00	1.00	3.00	3.00
		鲁迪-盖伊马刺	13.00	0.00	32.00	2.00	2.00	6.00
		乔丹-克劳福德	7.00	0.00	8.00	1.00	1.00	2.00
伊安-克拉克	7.00	0.00	21.00	1.00	1.00	1.00		

图表信息

标题 测试图表

描述 请输入图表描述

图表类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

筛选器 赛区

仪表盘

仪表盘列表

- 仪表盘目录
- 测试仪表盘
- 测试仪表盘1

测试仪表盘1

城市	业主小区	业主投诉次数(求和)
上海	南京路y座	572
	凤雨D座	1,218
	梧州路a座	792
	源街f座	670
	百路T座	602
	虞街d座	582
北京	兴城南Y座	632
	南京街g座	680
	台北路h座	678
	政街P座	704
	终街L座	590

注意：

- 1) 在幻数中，根据标签的数据格式不同，把标签分成三类，分别是日期类标签、文本类标签和数值类标签，用不同的icon表示。
- 2) 所有标签可以随意拖入维度行或者数值行，但根据实际使用场景，建议维度行拖入日期类和文本类标签、数值行拖入数值类标签。
- 3) 日期类标签拖入维度行时，不管之前格式如何，会默认按年/月/日/时/分/秒这样的格式去展示。
- 4) 日期类标签拖入维度行时，日期类标签会出现一个下拉按钮，点击下拉按钮，可以选择按年、按月、按日三种形式去统计数据。比如当前配置的图表显示的是2018年每天的营业额统计，现在通过点击日期下拉，选择按月时，图表数据显示的就是2018年每月的营业额统计。反之，如果表里只有大日期的数据，此时选择按细分日期去统计时，没有多大意义。比如当前配置的图表显示是2018年每月的营业额统计，现在通过点击日期下拉，选择按日时，图表数据显示就是2018年每月1号的营业额统计。

5) 数值类标签拖入数值行时，数值类标签会出现一个下拉按钮，点击下拉按钮，可以选择计算方式，包括求和、求平均、最大值、最小值、计数和去重计数，默认按求和进行计算。

6) 文本类或者日期类标签拖入数值行时，也会出现一个下拉按钮，点击下拉按钮，可以选择计算方式，包括计数和去重计数（因为文本类和日期类标签无法按其他方式进行计算）

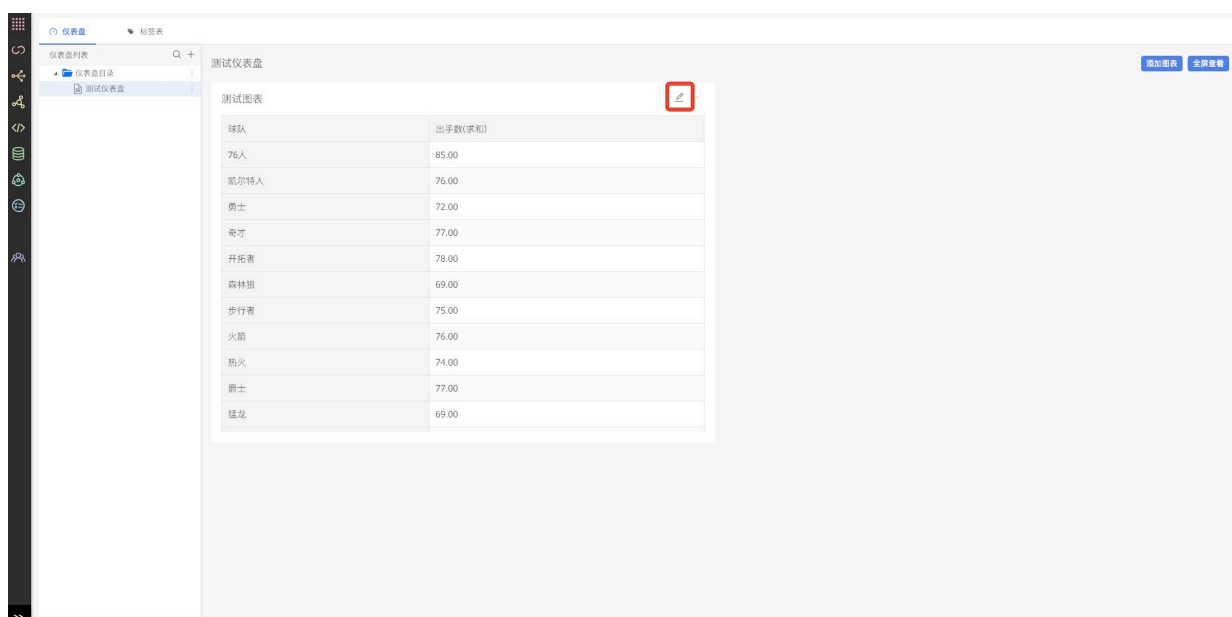
7) 拖入数值行的标签，下拉还可以选择设置数值显示格式，包括设置小数显示还是百分比显示、小数点位数、数值的单位、数值对应的颜色以及是否显示千分位。

8) 图表配置都是实时保存的，操作了即保存。

9) 不是什么图表组件都是可以选择的，只有选择的维度和数值符合组件要求，该组件才可以使用。不符合条件的组件会置灰，处于不可用状态。

13.3.9 编辑图表

1.在仪表盘页面，鼠标移入要编辑的图表，点击编辑按钮。



2.可以更改标签表。



注意：只允许同一个数据服务下的标签表进行切换。

3.可以重新选择维度和数值。

The screenshot shows a data visualization tool interface. At the top, there are two dropdown menus for selecting dimensions and values. The dimension dropdown is set to '球队' (Team) and the value dropdown is set to '出手数(求和)' (Shooting Count (Sum)). A red box highlights these two dropdown menus. Below the dropdowns is a table with the following data:

球队	出手数(求和)
76人	85
凯尔特人	76
勇士	72
奇才	77
开拓者	78
森林狼	69
步行者	75
火箭	76
热火	74
爵士	77
猛龙	69
雄鹿	69
雷霆	67
马刺	53
骑士	65
鹈鹕	87

On the right side of the interface, there is a '图表信息' (Chart Information) panel with fields for '标题' (Title) and '描述' (Description). Below that is a '配置面板' (Configuration Panel) with various settings for chart size, grid, and alignment.

4.可以重新编辑图表信息。

This screenshot is identical to the one above, but with a red box highlighting the '图表信息' (Chart Information) panel on the right. The '标题' (Title) field contains '测试图表' (Test Chart) and the '描述' (Description) field contains '请输入图表描述' (Please enter chart description).

5.可以重新设置图表配置区。

返回

标签表名称 维度 球队 添加对比 添加下钻

三级目录 数值 出手数(求和) 添加滤镜

标签列表 横轴模式 纵轴模式

球队	出手数(求和)
76人	85.00
凯尔特人	76.00
勇士	72.00
奇才	77.00
开拓者	78.00
森林狼	69.00
步行者	75.00
火箭	76.00
热火	74.00
爵士	77.00
猛龙	69.00
雄鹿	69.00
雷霆	67.00
马刺	53.00
骑士	65.00
鹈鹕	87.00

图表信息

标题 测试图表

描述 请输入图表描述

图表类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

筛选器 +

图内筛选器

6.修改好后，点击返回按钮，回到仪表盘页面可以看到已经修改后的图表。

返回
添加对比 添加下钻

标签表名称

三级目录

标签列表

- T 球员
- T 赛区
- T 球队
- # 得分
- # 篮板
- # 助攻
- # 抢断
- # 上场时间
- # 盖帽
- # 出手数

维度 赛区 球队 球员

数值 出手数(求和) 盖帽(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和)

图内筛选器 请选择赛区 横轴模式 纵轴模式

赛区	球队	球员	出手数(求和)	盖帽(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)
火箭		PJ-塔克	6.00	1.00	34.00	1.00	1.00	7.00
		克林特-卡佩拉	9.00	2.00	31.00	1.00	1.00	12.00
		克里斯-保罗	17.00	0.00	35.00	2.00	6.00	6.00
		埃里克-戈登	13.00	1.00	32.00	1.00	2.00	3.00
		特雷沃-阿里扎	8.00	0.00	34.00	1.00	1.00	4.00
		詹姆斯-哈登	23.00	1.00	37.00	2.00	7.00	5.00
西南赛区	马刺	帕蒂-米尔斯	11.00	0.00	33.00	1.00	3.00	2.00
		德章泰-穆雷	6.00	0.00	19.00	1.00	2.00	4.00
		拉马库斯-阿尔德里奇	16.00	0.00	35.00	1.00	2.00	9.00
		马努-吉诺比利	7.00	0.00	21.00	1.00	3.00	3.00
		鲁迪-盖伊马刺	13.00	0.00	32.00	2.00	2.00	6.00
		乔丹-克劳福德	7.00	0.00	8.00	1.00	1.00	2.00
伊安-克拉克	7.00	0.00	21.00	1.00	1.00	1.00		

图表信息

标题 测试图表

描述 请输入图表描述

图表类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

筛选器

赛区

仪表盘
标签表

仪表盘列表

- 仪表盘目录
- 测试仪表盘

测试仪表盘

测试图表

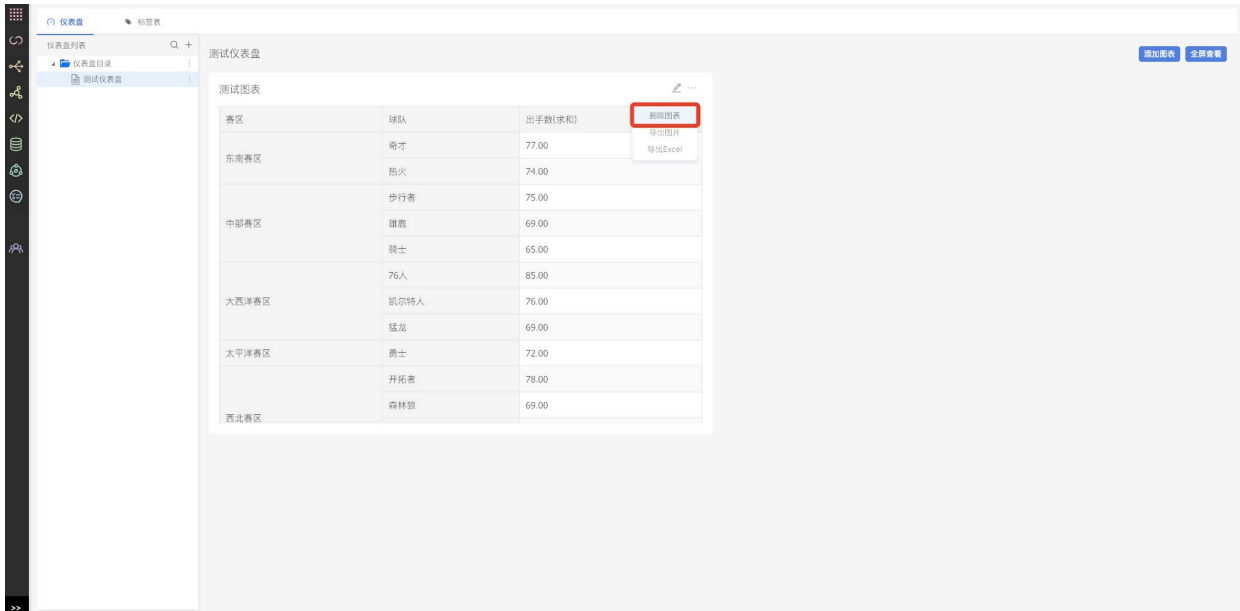
赛区	球队	出手数(求和)
东南赛区	奇才	77.00
	热火	74.00
中部赛区	步行者	75.00
	雄鹿	69.00
	骑士	65.00
大西洋赛区	76人	85.00
	凯尔特人	76.00
	猛龙	69.00
太平洋赛区	勇士	72.00
	开拓者	78.00
西北赛区	森林狼	69.00

注意：

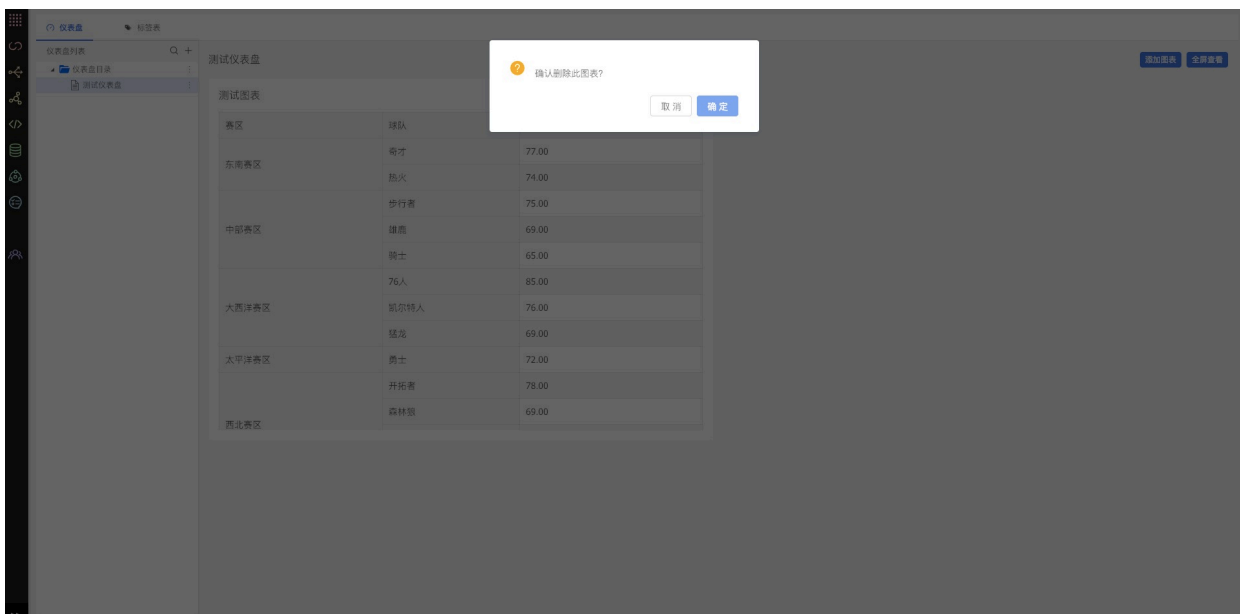
- 1) 切换标签表时，会清空之前配置的图表，需有二次确认。
- 2) 图表配置过程是实时保存的，无法撤销。

13.3.10 删除图表

1.在仪表盘页面，鼠标移入要删除的图表，选择更多按钮，点击删除图表按钮。

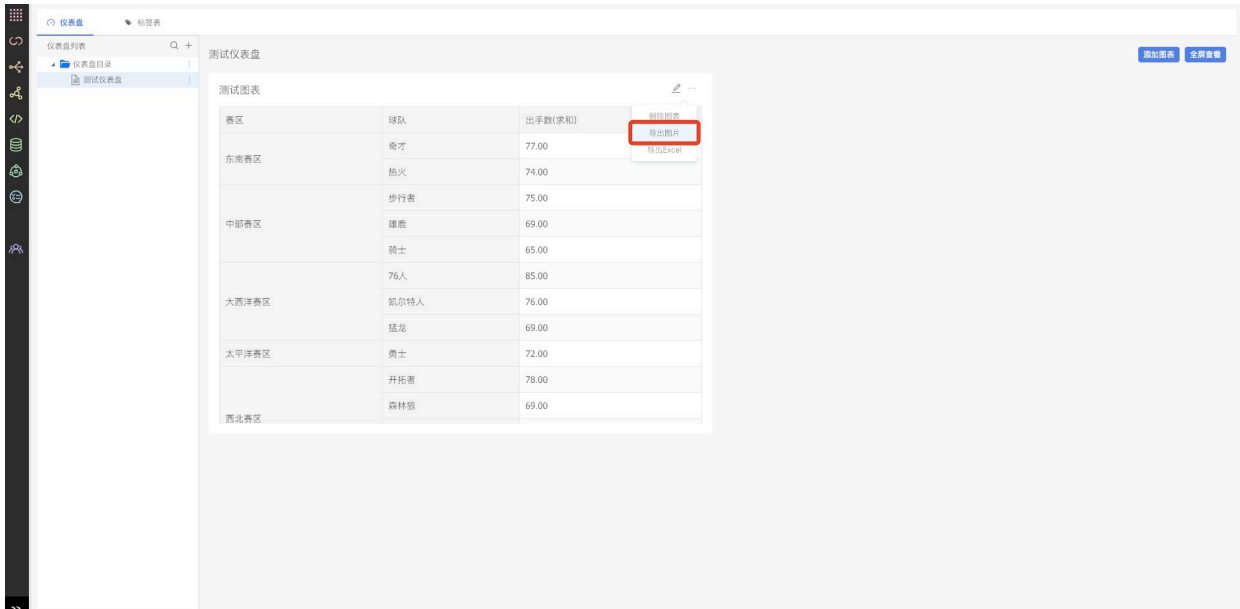


2.删除图表需要二次确认，点击确定按钮后，删除图表成功。



13.3.11 导出图片

1.在仪表盘页面，鼠标移入要导出的图表，选择更多按钮，点击导出图片按钮。



2.选择图片要存储的地址，图片导出成功，可以查看导出的图片。

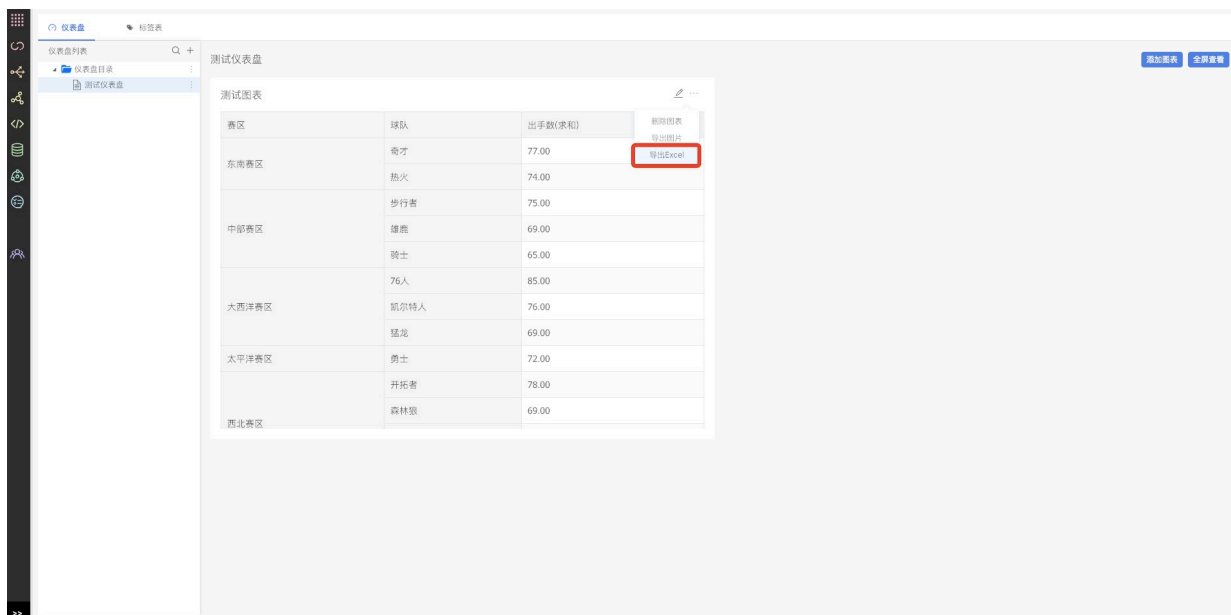


注意：

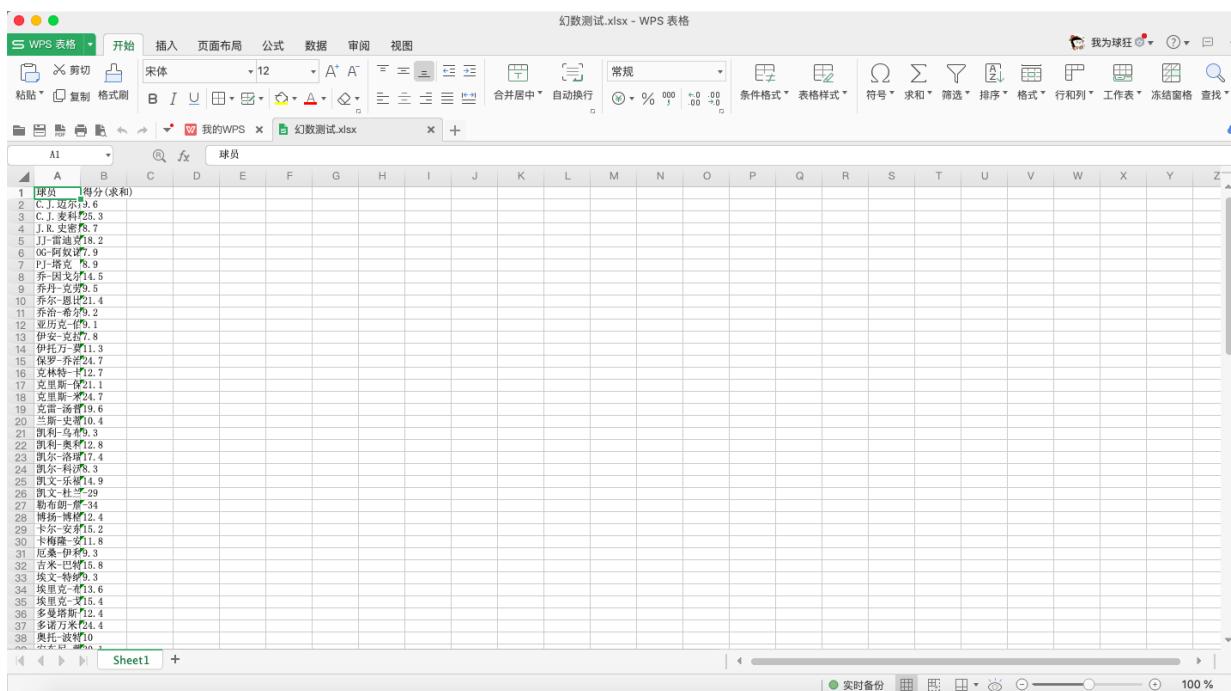
- 1) 幻数的图表组件封装的是e-charts，因为e-charts图表都是位图，所以在导出时，小尺寸图表导出时分辨率不高。
- 2) 表格类图表导出时，只会把当前所能看到的进行导出，也就是说在需要滚动才能看到的部分是无法导出的。

13.3.12 导出Excel

1.在仪表盘页面，鼠标移入要导出的表格类图表，选择更多按钮，点击导出Excel按钮。



2.选择Excel要存储的地址，Excel导出成功，可以查看导出的Excel。

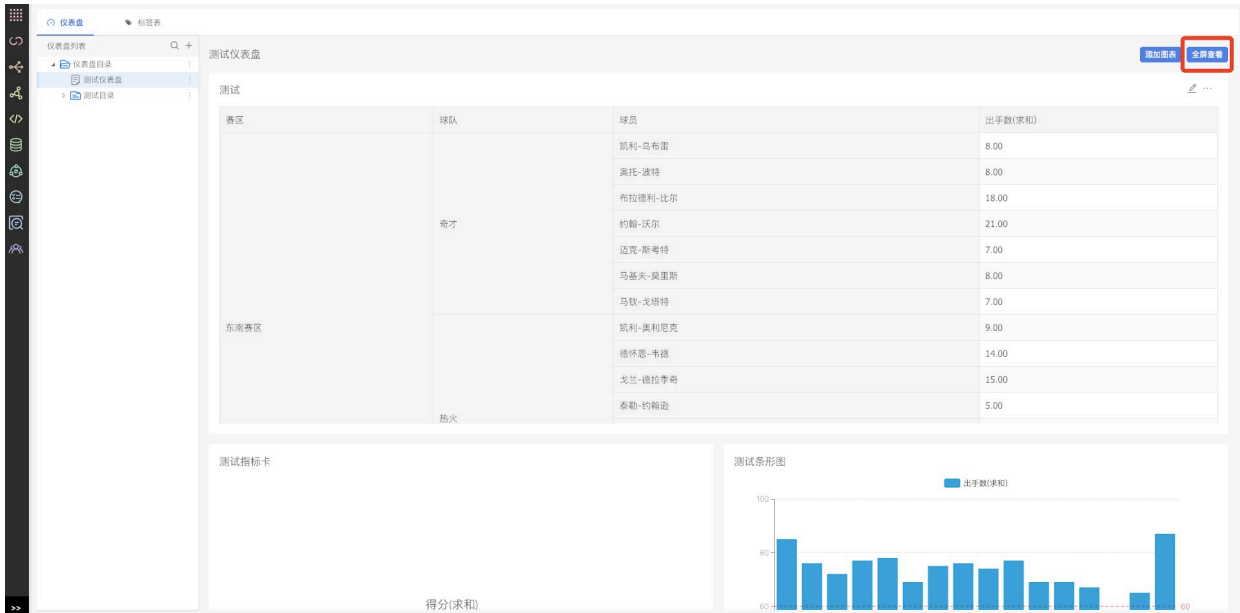


注意：

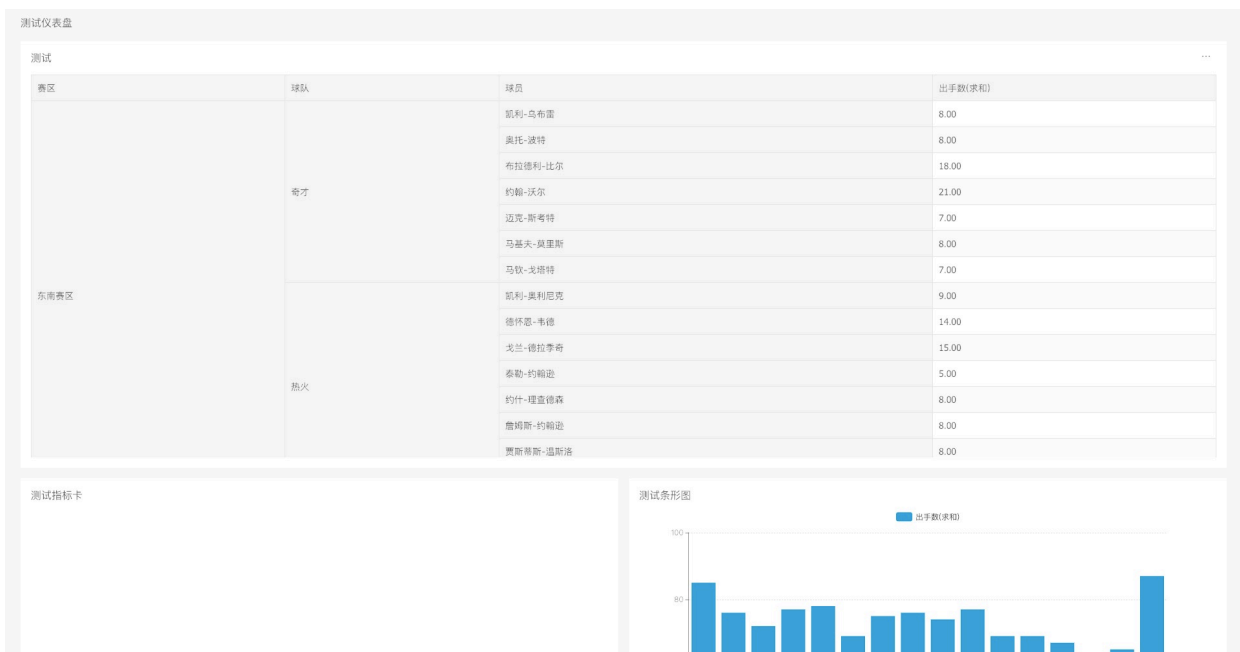
- 1) 只有表格类型的图表才可以导出为Excel。
- 2) 导出Excel最大条数限制为1500条。

13.3.13 全屏查看

1.选择要全屏查看的仪表盘，点击全屏查看按钮。

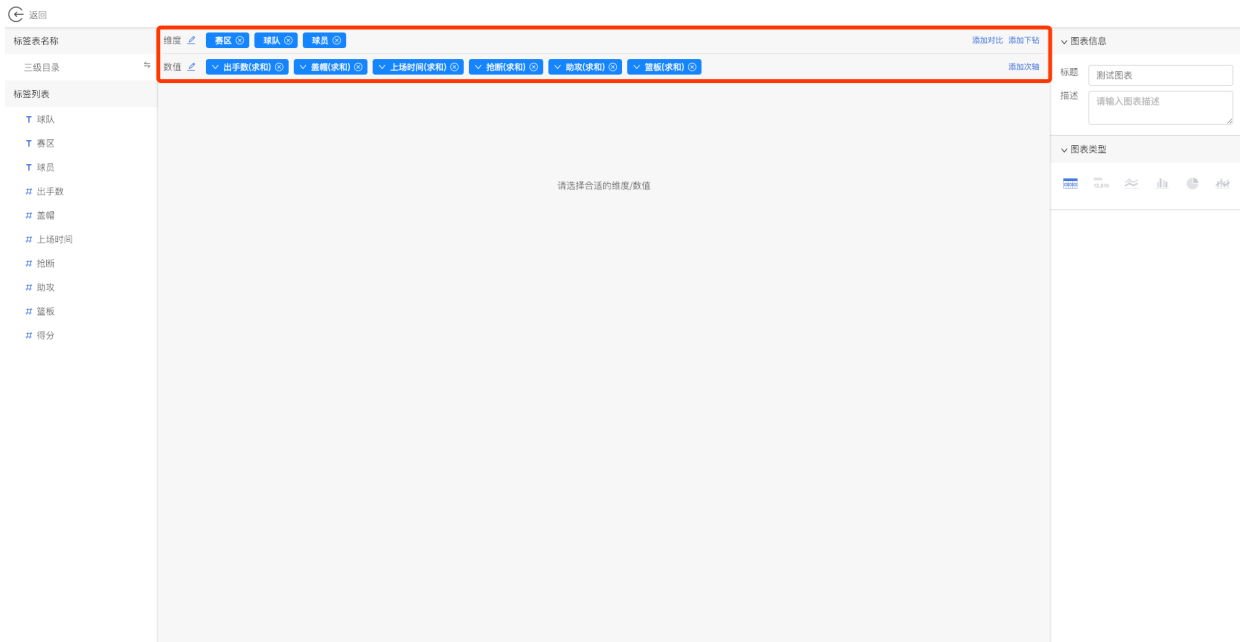


2. 就可以全屏查看报表内容。



13.3.14 表格组件

1. 在图表编辑页，维度行选择0个或多个标签，数值行选择1个或多个标签。



2.在右侧配置区域的图表类型中，选择表格组件。



3.在配置面板中，可以选择是否合并单元格。默认合并单元格。

返回

标签表名称 维度 赛区 球队 球员 添加对比 添加下钻

三级目录 数值 出手数(求和) 篮板(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和) 添加快捷

标签列表 横轴模式 纵轴模式

赛区	球队	球员	出手数(求和)	篮板(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)
东南赛区	奇才	凯利-乌布雷	8.00	1.00	25.00	1.00	1.00	4.00
		奥托-波特	8.00	1.00	32.00	1.00	2.00	5.00
		布拉德利-比尔	18.00	0.00	36.00	1.00	3.00	3.00
		约翰-沃尔	21.00	1.00	39.00	2.00	12.00	6.00
		迈克-斯科特	7.00	0.00	21.00	0.00	1.00	4.00
		马基夫-莫里斯	8.00	1.00	30.00	1.00	2.00	8.00
	热火	马钦-戈塔特	7.00	0.00	27.00	0.00	1.00	6.00
		凯利-奥利尼克	9.00	1.00	29.00	1.00	4.00	5.00
		德怀恩-韦德	14.00	0.00	25.00	1.00	4.00	4.00
		戈兰-德拉季奇	15.00	0.00	31.00	1.00	5.00	3.00
		泰勒-约翰逊	5.00	0.00	16.00	0.00	1.00	2.00
		约什-理查德森	8.00	1.00	26.00	2.00	3.00	3.00
步行者	詹姆斯-温斯洛	8.00	1.00	25.00	1.00	3.00	7.00	
	韦恩-埃灵顿	7.00	0.00	20.00	0.00	1.00	2.00	
	兰斯-史蒂芬森	9.00	0.00	21.00	0.00	3.00	3.00	
	博扬-博格达诺维奇	12.00	0.00	34.00	1.00	2.00	3.00	
	多曼塔斯-萨博尼斯	9.00	0.00	24.00	0.00	1.00	5.00	
	维克托-奥拉迪波	18.00	0.00	37.00	2.00	6.00	8.00	

图表信息

标题 测试图表

描述 请输入图表描述

图表类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

图内筛选器

4.也可以选择固定顶部表头或者左侧表头。

返回

标签表名称 维度 赛区 球队 球员 添加对比 添加下钻

三级目录 数值 出手数(求和) 篮板(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和) 添加快捷

标签列表 横轴模式 纵轴模式

赛区	球队	球员	出手数(求和)	篮板(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)
东南赛区	奇才	凯利-乌布雷	8.00	1.00	25.00	1.00	1.00	4.00
		奥托-波特	8.00	1.00	32.00	1.00	2.00	5.00
		布拉德利-比尔	18.00	0.00	36.00	1.00	3.00	3.00
		约翰-沃尔	21.00	1.00	39.00	2.00	12.00	6.00
		迈克-斯科特	7.00	0.00	21.00	0.00	1.00	4.00
		马基夫-莫里斯	8.00	1.00	30.00	1.00	2.00	8.00
	热火	马钦-戈塔特	7.00	0.00	27.00	0.00	1.00	6.00
		凯利-奥利尼克	9.00	1.00	29.00	1.00	4.00	5.00
		德怀恩-韦德	14.00	0.00	25.00	1.00	4.00	4.00
		戈兰-德拉季奇	15.00	0.00	31.00	1.00	5.00	3.00
		泰勒-约翰逊	5.00	0.00	16.00	0.00	1.00	2.00
		约什-理查德森	8.00	1.00	26.00	2.00	3.00	3.00
步行者	詹姆斯-温斯洛	8.00	1.00	25.00	1.00	3.00	7.00	
	韦恩-埃灵顿	7.00	0.00	20.00	0.00	1.00	2.00	
	兰斯-史蒂芬森	9.00	0.00	21.00	0.00	3.00	3.00	
	博扬-博格达诺维奇	12.00	0.00	34.00	1.00	2.00	3.00	
	多曼塔斯-萨博尼斯	9.00	0.00	24.00	0.00	1.00	5.00	
	维克托-奥拉迪波	18.00	0.00	37.00	2.00	6.00	8.00	

图表信息

标题 测试图表

描述 请输入图表描述

图表类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

图内筛选器

5.也可以选择单元格字体对齐方式，默认左对齐。

返回

标签表名称 维度 赛区 球队 球员 添加对比 添加下钻

三级目录 数值 出手数(求和) 篮板(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和) 添加次轴

标签列表 横轴模式 纵轴模式

赛区	球队	球员	出手数(求和)	篮板(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)
东南赛区	奇才	凯利-乌布雷	8	1	25	1	1	4
		奥托-波特	8	1	32	1	2	5
		布拉德利-比尔	18	0	36	1	3	3
		约翰-沃尔	21	1	39	2	12	6
		迈克-斯科特	7	0	21	0	1	4
		马基夫-莫里斯	8	1	30	1	2	8
	热火	马钦-戈塔特	7	0	27	0	1	6
		凯利-奥利尼克	9	1	29	1	4	5
		德怀恩-韦德	14	0	25	1	4	4
		戈兰-德拉季奇	15	0	31	1	5	3
		泰勒-约翰逊	5	0	16	0	1	2
		约什-理查德森	8	1	26	2	3	3
步行者	詹姆斯-约翰逊	8	1	32	1	5	6	
	贾斯汀-温斯洛	8	1	25	1	3	7	
	韦恩-埃灵顿	7	0	20	0	1	2	
	兰斯-史蒂芬森	9	0	21	0	3	3	
	博扬-博格达诺维奇	12	0	34	1	2	3	
	多曼塔斯-萨博尼斯	9	0	24	0	1	5	
维克托-奥拉迪波	18	0	37	2	6	8		

图信息 标题 测试图表 描述 请输入图表描述

图类型

配置面板 大小设置 1*1 2*1 2*2 4*1 4*2 4*4 合并单元格 表头固定 无 顶部 左侧 对齐方式 居左 居中 居右 行列总计 行总计 列总计

预警设置 预警 筛选器 图内筛选器

6.也可以开启行总计和列总计。默认不开启。

返回

标签表名称 维度 赛区 球队 球员 添加对比 添加下钻

三级目录 数值 出手数(求和) 篮板(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和) 添加次轴

标签列表 横轴模式 纵轴模式

赛区	球队	球员	出手数(求和)	篮板(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)	行总计
东南赛区	奇才	凯利-乌布雷	8	1	25	1	1	4	40
		奥托-波特	8	1	32	1	2	5	49
		布拉德利-比尔	18	0	36	1	3	3	61
		约翰-沃尔	21	1	39	2	12	6	81
		迈克-斯科特	7	0	21	0	1	4	33
		马基夫-莫里斯	8	1	30	1	2	8	50
	热火	马钦-戈塔特	7	0	27	0	1	6	41
		凯利-奥利尼克	9	1	29	1	4	5	49
		德怀恩-韦德	14	0	25	1	4	4	48
		戈兰-德拉季奇	15	0	31	1	5	3	55
		泰勒-约翰逊	5	0	16	0	1	2	24
		约什-理查德森	8	1	26	2	3	3	43
步行者	詹姆斯-约翰逊	8	1	32	1	5	6	53	
	贾斯汀-温斯洛	8	1	25	1	3	7	45	
	韦恩-埃灵顿	7	0	20	0	1	2	30	
	兰斯-史蒂芬森	9	0	21	0	3	3	36	
	博扬-博格达诺维奇	12	0	34	1	2	3	52	
	多曼塔斯-萨博尼斯	9	0	24	0	1	5	39	
维克托-奥拉迪波	18	0	37	2	6	8	71		

图信息 标题 测试图表 描述 请输入图表描述

图类型

配置面板 大小设置 1*1 2*1 2*2 4*1 4*2 4*4 合并单元格 表头固定 无 顶部 左侧 对齐方式 居左 居中 居右 行列总计 行总计 列总计

预警设置 预警 筛选器 图内筛选器

7.也可以通过切换横轴模式和纵轴模式，使表格类型的行列进行互换。

返回

标签表名称 维度 赛区 球队 球员 添加对比 添加下钻

三级目录 数值 出手数(求和) 篮板(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和) 添加次轴

标签列表

模轴模式 纵轴模式

赛区	东南赛区											
球队	奇才											热火
球员	凯利-乌布雷	奥托-波特	布拉德利-比尔	约翰-沃尔	迈克-斯考特	马基夫-莫里斯	马钦-戈塔特	凯利-奥利尼克	德怀恩-韦德	戈兰-德拉季奇	泰勒-约翰逊	约什-理查德森
出手数(求和)	8.00	8.00	18.00	21.00	7.00	8.00	7.00	9.00	14.00	15.00	5.00	8.00
篮板(求和)	1.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	1.00
上场时间(求和)	25.00	32.00	36.00	39.00	21.00	30.00	27.00	29.00	25.00	31.00	16.00	26.00
抢断(求和)	1.00	1.00	1.00	2.00	0.00	1.00	0.00	1.00	1.00	1.00	0.00	2.00
助攻(求和)	1.00	2.00	3.00	12.00	1.00	2.00	1.00	4.00	4.00	5.00	1.00	3.00
篮板(求和)	4.00	5.00	3.00	6.00	4.00	8.00	6.00	5.00	4.00	3.00	2.00	3.00

图表信息

标题 测试图表

描述 请输入图表描述

图表类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

筛选器 +

图内筛选器

8.还可以通过拖动单元格，调整列宽。

返回

标签表名称 维度 赛区 球队 球员 添加对比 添加下钻

三级目录 数值 出手数(求和) 篮板(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和) 添加次轴

标签列表

模轴模式 纵轴模式

赛区	球队	球员	出手数(求和)	篮板(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)
东南赛区	奇才	凯利-乌布雷	8.00	1.00	25.00	1.00	1.00	4.00
		奥托-波特	8.00	1.00	32.00	1.00	2.00	5.00
		布拉德利-比尔	18.00	0.00	36.00	1.00	3.00	3.00
		约翰-沃尔	21.00	1.00	39.00	2.00	12.00	6.00
		迈克-斯考特	7.00	0.00	21.00	0.00	1.00	4.00
		马基夫-莫里斯	8.00	1.00	30.00	1.00	2.00	8.00
	热火	马钦-戈塔特	7.00	0.00	27.00	0.00	1.00	6.00
		凯利-奥利尼克	9.00	1.00	29.00	1.00	4.00	5.00
		德怀恩-韦德	14.00	0.00	25.00	1.00	4.00	4.00
		戈兰-德拉季奇	15.00	0.00	31.00	1.00	5.00	3.00
		泰勒-约翰逊	5.00	0.00	16.00	0.00	1.00	2.00
		约什-理查德森	8.00	1.00	26.00	2.00	3.00	3.00
步行者	詹姆斯-约翰逊	8.00	1.00	32.00	1.00	5.00	6.00	
	贾斯汀-温斯洛	8.00	1.00	25.00	1.00	3.00	7.00	
	韦恩-埃灵顿	7.00	0.00	20.00	0.00	1.00	2.00	
	兰斯-史蒂芬森	9.00	0.00	21.00	0.00	3.00	3.00	
	博扬-博格达诺维奇	12.00	0.00	34.00	1.00	2.00	3.00	
	多曼塔斯-萨博尼斯	9.00	0.00	24.00	0.00	1.00	5.00	
维克托-奥拉迪波	18.00	0.00	37.00	2.00	6.00	8.00		

图表信息

标题 测试图表

描述 请输入图表描述

图表类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

筛选器 +

图内筛选器

9.点击返回按钮，返回仪表盘就可以看到配置的表格图表。

返回
标签表名称
维度 赛区 球队 球员
添加对比 添加下钻

三级目录
数值 出手数(求和) 盖帽(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和)
添加次轴

标签列表
图内筛选器 请选择赛区
横轴模式 纵轴模式

- T 球员
- T 赛区
- T 球队
- # 得分
- # 篮板
- # 助攻
- # 抢断
- # 上场时间
- # 盖帽
- # 出手数

赛区	球队	球员	出手数(求和)	盖帽(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)
火箭		PJ-塔克	6.00	1.00	34.00	1.00	1.00	7.00
		克林特-卡佩拉	9.00	2.00	31.00	1.00	1.00	12.00
		克里斯-保罗	17.00	0.00	35.00	2.00	6.00	6.00
		埃里克-戈登	13.00	1.00	32.00	1.00	2.00	3.00
		特雷沃-阿里扎	8.00	0.00	34.00	1.00	1.00	4.00
		詹姆斯-哈登	23.00	1.00	37.00	2.00	7.00	5.00
西南赛区	马刺	帕蒂-米尔斯	11.00	0.00	33.00	1.00	3.00	2.00
		德章泰-穆雷	6.00	0.00	19.00	1.00	2.00	4.00
		拉马库斯-阿尔德里奇	16.00	0.00	35.00	1.00	2.00	9.00
		马努-吉诺比利	7.00	0.00	21.00	1.00	3.00	3.00
		鲁迪-盖伊马刺	13.00	0.00	32.00	2.00	2.00	6.00
		乔丹-克劳福德	7.00	0.00	8.00	1.00	1.00	2.00
伊安-克拉克	7.00	0.00	21.00	1.00	1.00	1.00		

标题 测试图表

描述 请输入图表描述

图例信息

图例类型

配置面板

大小设置

合并单元格

表头固定

对齐方式

行列总计

预警设置

预警

筛选器

筛选器

赛区

仪表盘
标签表

- 仪表盘列表
- 仪表盘目录
- 测试仪表盘
- 测试目录
- 测试仪表盘1

测试仪表盘1
添加图表 全屏查看

各地区投诉次数分布

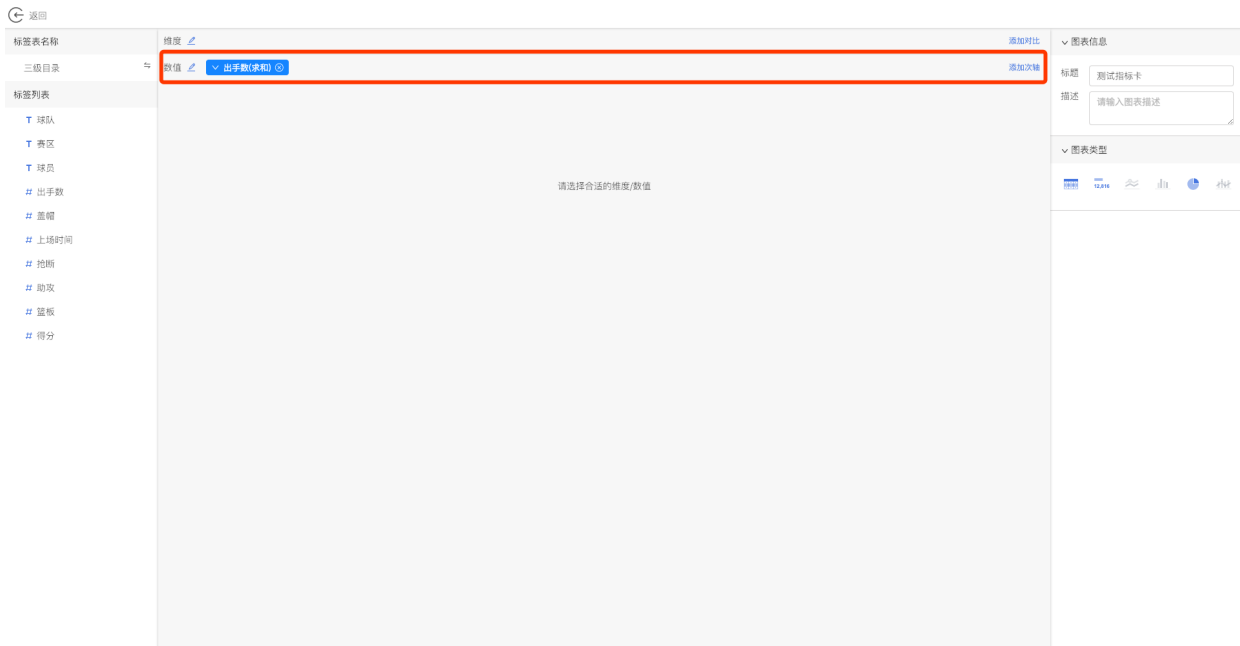
城市	业主小区	业主投诉次数(求和)
上海	南京路y座	572
	凤雨D座	1218
	梧州路a座	792
	源街f座	670
	百路T座	602
	虞街d座	582
北京	兴城南Y座	632
	南京街g座	680
	台北路h座	678
	政街P座	704
	终街L座	590

注意：

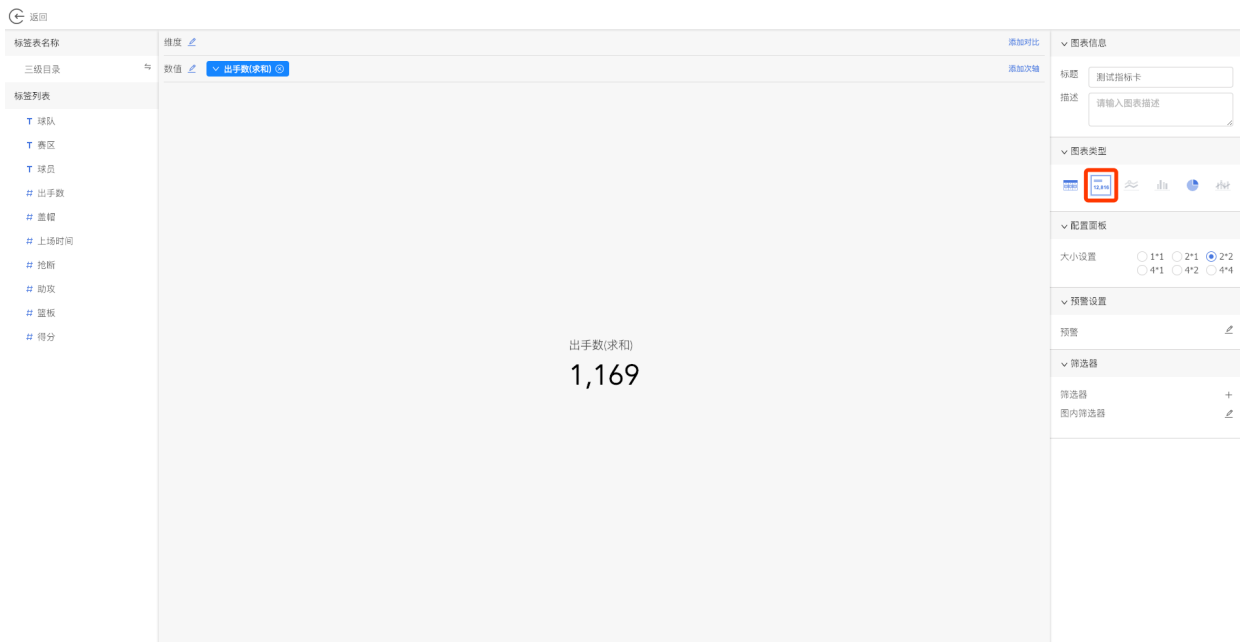
- 1) 合并单元格，对维度行标签顺序有要求，大分类标签在前，细分标签在后（因为取数时先按第一个维度标签group by，再取对应的第二个维度标签，所以大分类在后的无法合并）
- 2) 关于固定表头，对于只有维度标签，没有数值标签时，这个时候选择固定表头，会造成图表无法滚动。（因为整个表头都被固定了）
- 3) 为防止所有数据被拉取，受分析引擎限制，表格组件展示的最大条数为1500条，超过部分不展示。

13.3.15 指标卡组件

1.在编辑图表页面，维度行为空，数值行选择1个或2个标签。



2.在右侧配置区的图表类型中，选择指标卡组件。



3.点击返回按钮，返回仪表盘，可以看到指标卡配置成功。

返回

标签表名称 维度 [添加对比](#)

三级目录 数值 [出手数\(求和\)](#) [添加次轴](#)

标签列表

- T 球员
- T 赛区
- T 球队
- # 得分
- # 篮板
- # 助攻
- # 抢断
- # 上场时间
- # 盖帽
- # 出手数

出手数(求和)
1,169

图表信息

标题 测试指标卡

描述 请输入图表描述

图类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

预警设置

预警 [编辑](#)

筛选器

筛选器 [+](#)

图内筛选器 [编辑](#)

仪表盘 标签表

仪表盘列表

- 仪表盘目录
- 测试仪表盘

测试仪表盘 [添加仪表](#) [全屏查看](#)

测试图表

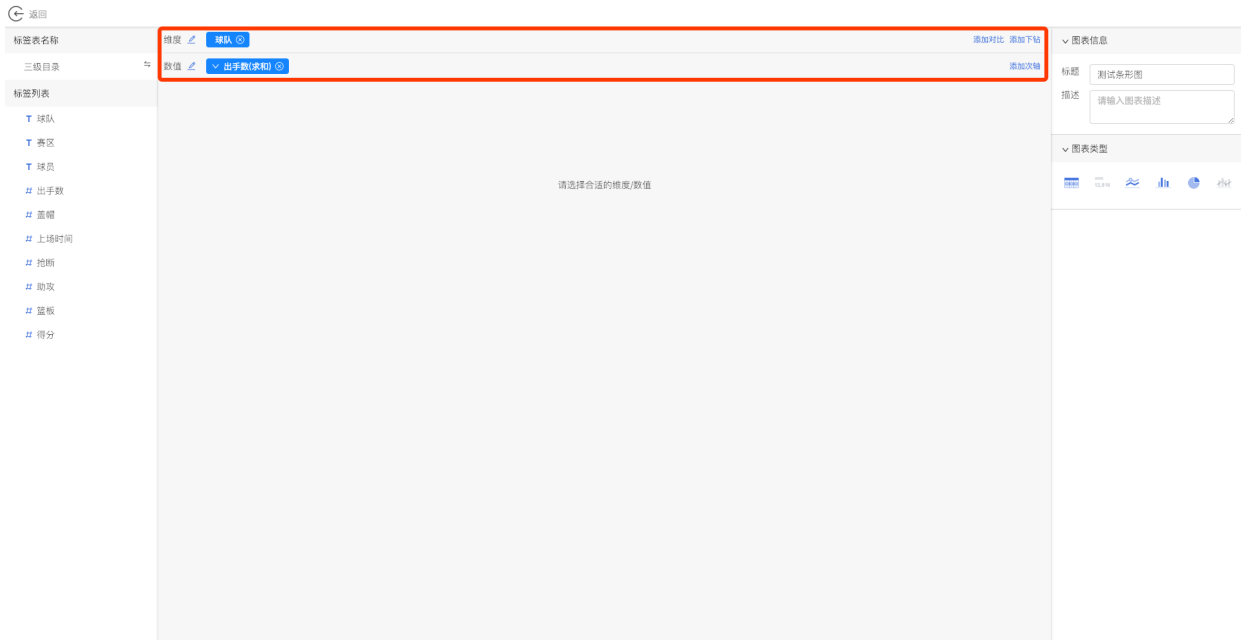
赛区	球队	球员	出手数(求和)	盖帽(求和)	上场时间(求和)	抢断(求和)	助攻(求和)
奇才		凯利-乌布雷	8.00	1.00	25.00	1.00	1.00
		奥托-波特	8.00	1.00	32.00	1.00	2.00
		布拉德利-比尔	18.00	0.00	36.00	1.00	3.00
		约翰-沃尔	21.00	1.00	39.00	2.00	12.00
		迈克-斯科特	7.00	0.00	21.00	0.00	1.00
		马基夫-莫里斯	8.00	1.00	30.00	1.00	2.00
		马钱-戈塔特	7.00	0.00	27.00	0.00	1.00
东南赛区		凯利-奥利尼克	9.00	1.00	29.00	1.00	4.00
		德怀恩-韦德	14.00	0.00	25.00	1.00	4.00
		戈兰-德拉季奇	15.00	0.00	31.00	1.00	5.00
		泰勒-约翰逊	5.00	0.00	16.00	0.00	1.00
热火							

测试指标卡

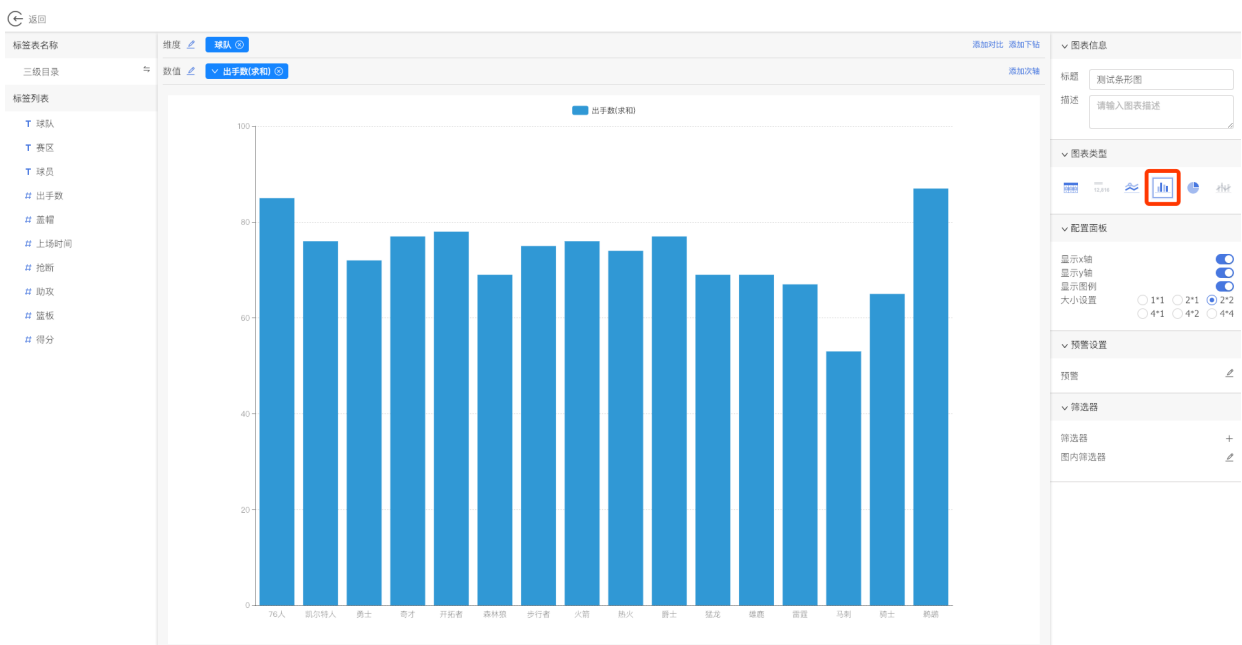
出手数(求和)
1,169

13.3.16 条形图组件

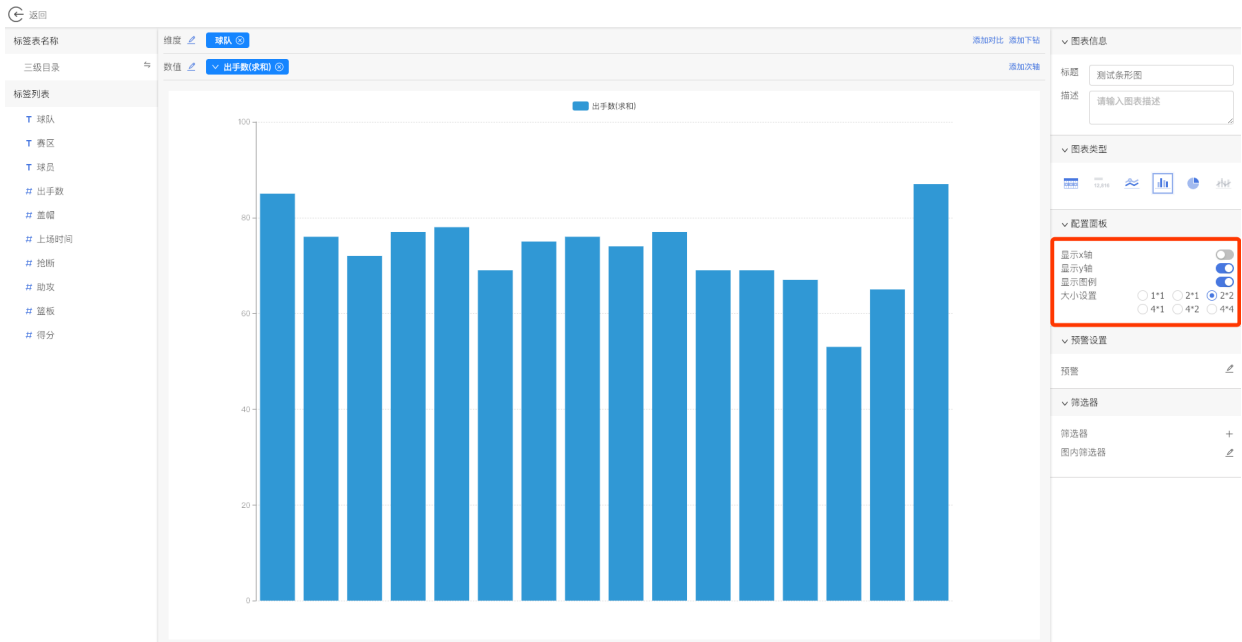
1.在图表编辑页面，维度行添加1个标签，数值行添加1个或多个标签。



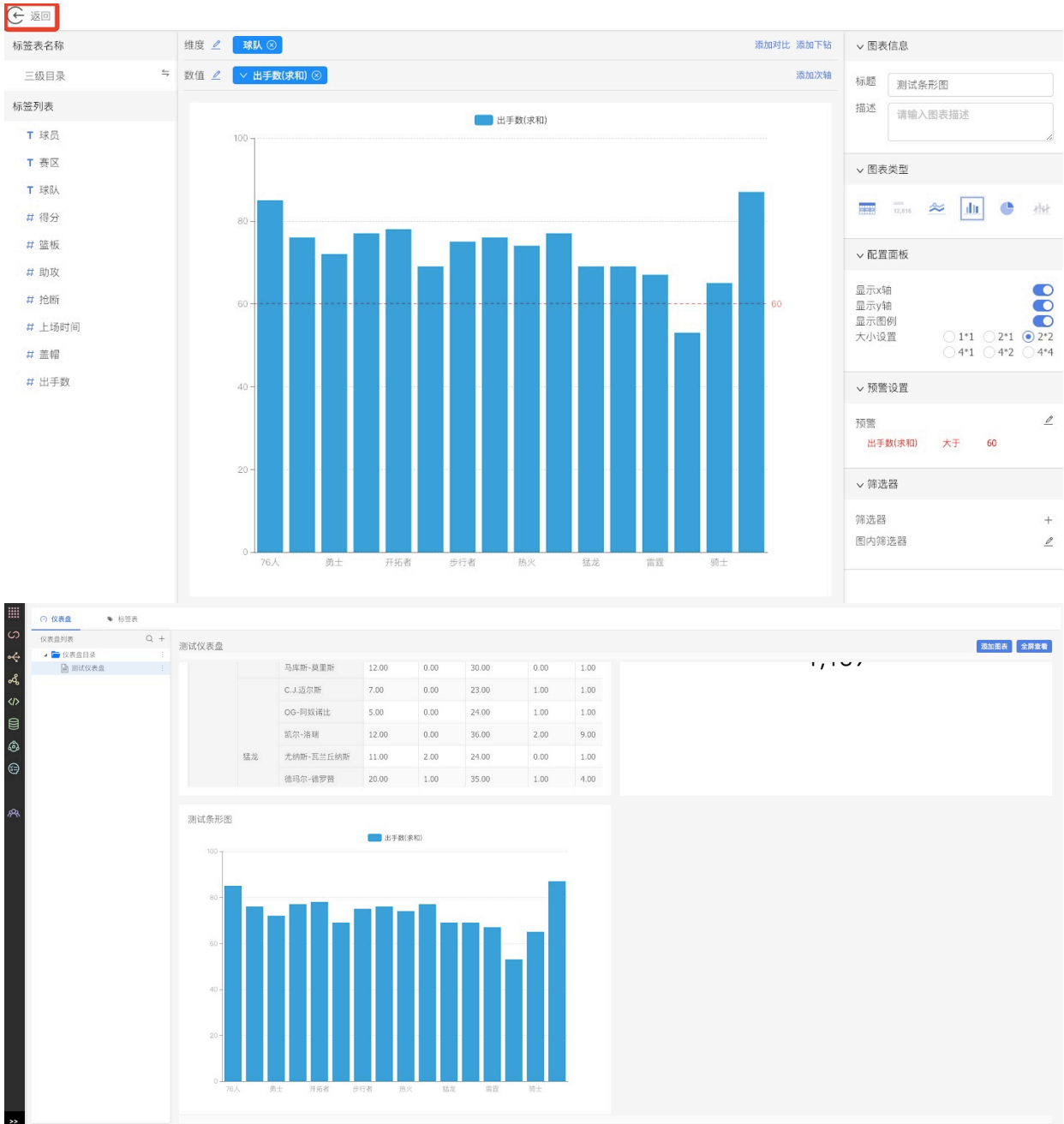
2.在右侧配置区的图表类型中，选择条形图组件。



3.根据需要，在配置面板中对条形图进行调整，包括是否显示x轴、是否显示y轴，是否显示图例等。

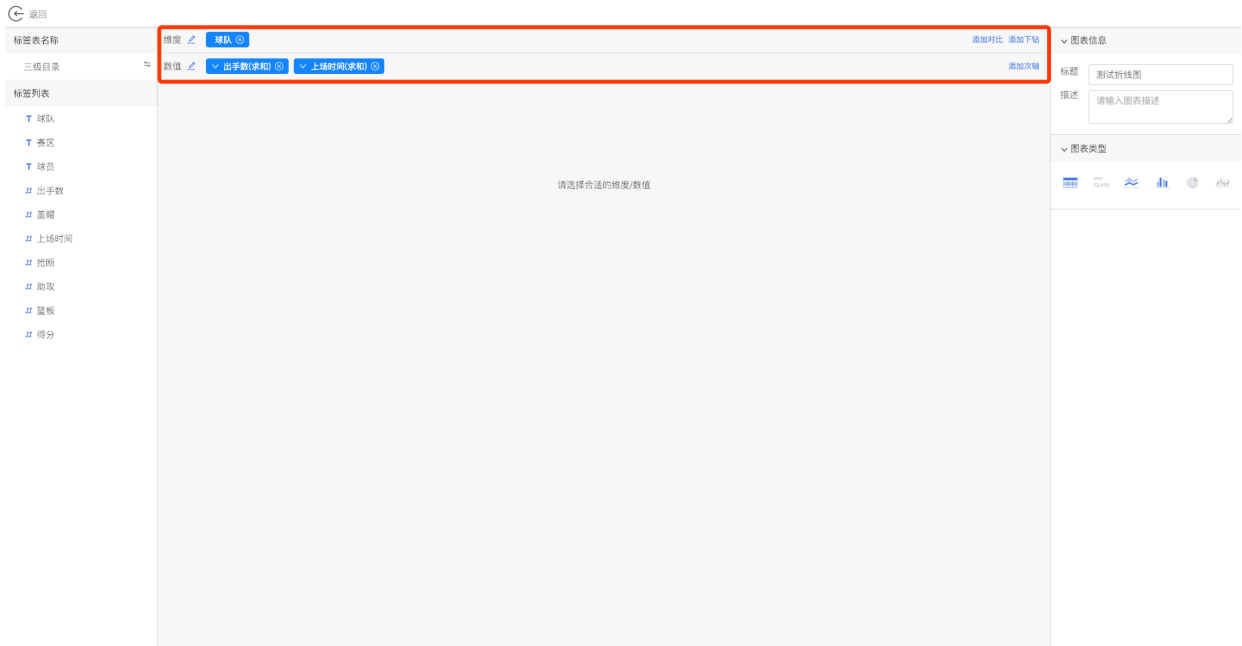


4. 点击返回按钮，返回仪表盘页面，可以看到条形图配置成功。

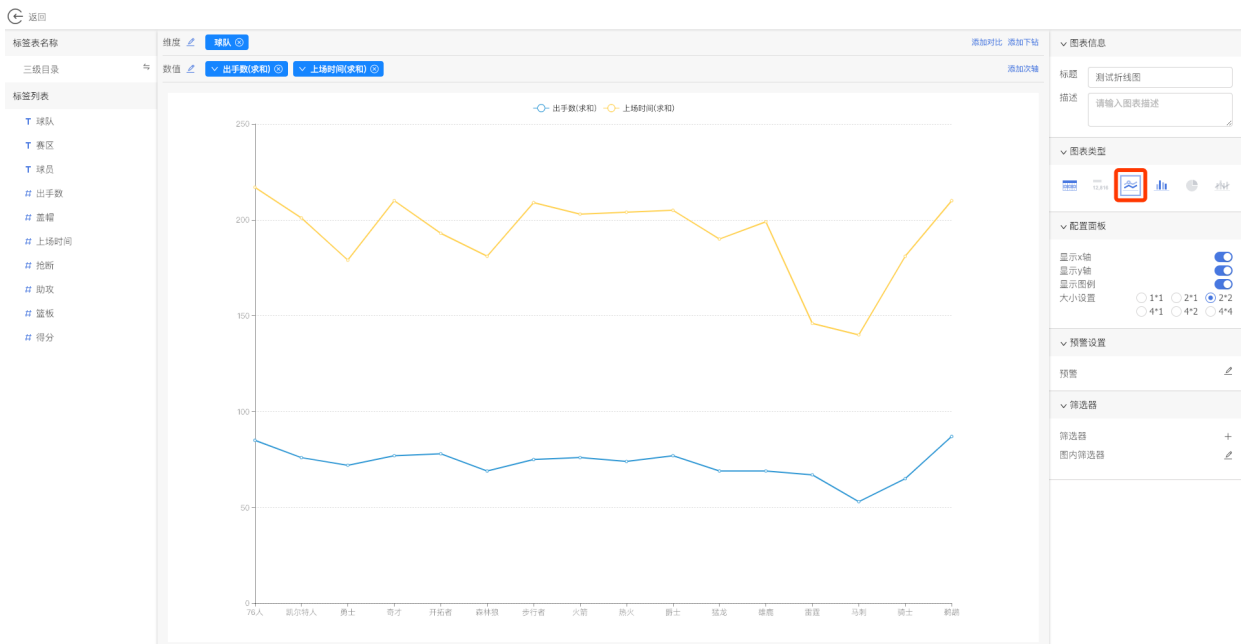


13.9.17 折线图组件

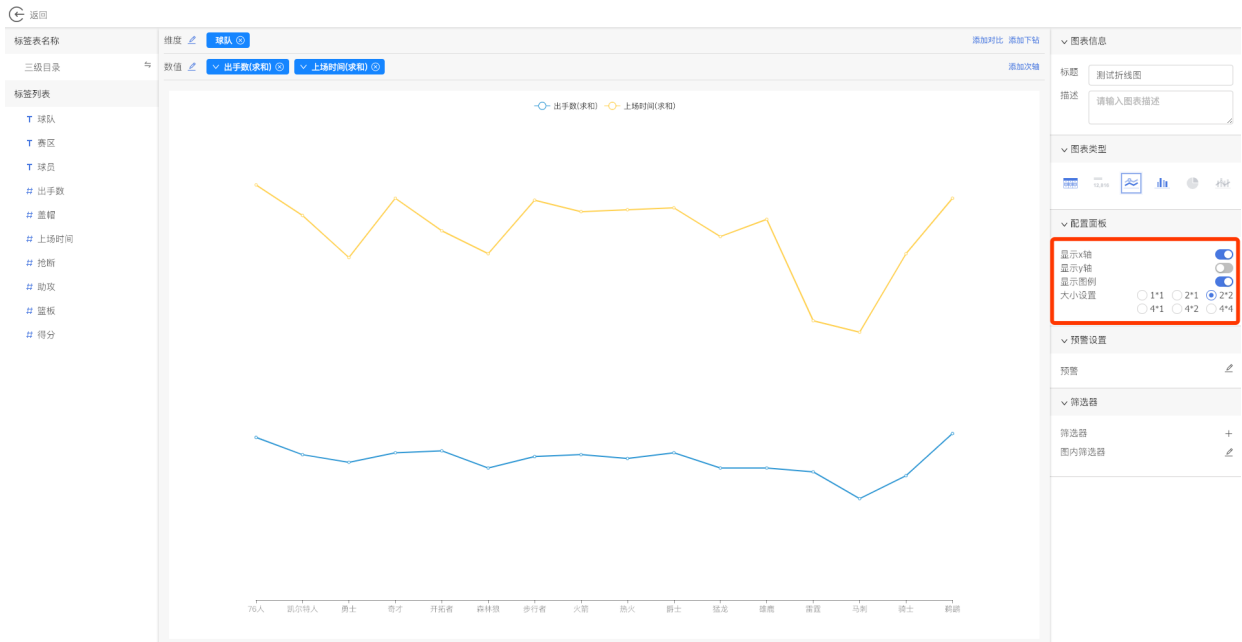
1.在图表编辑页面，维度行选择1个标签，数值行选择1个或多个标签。



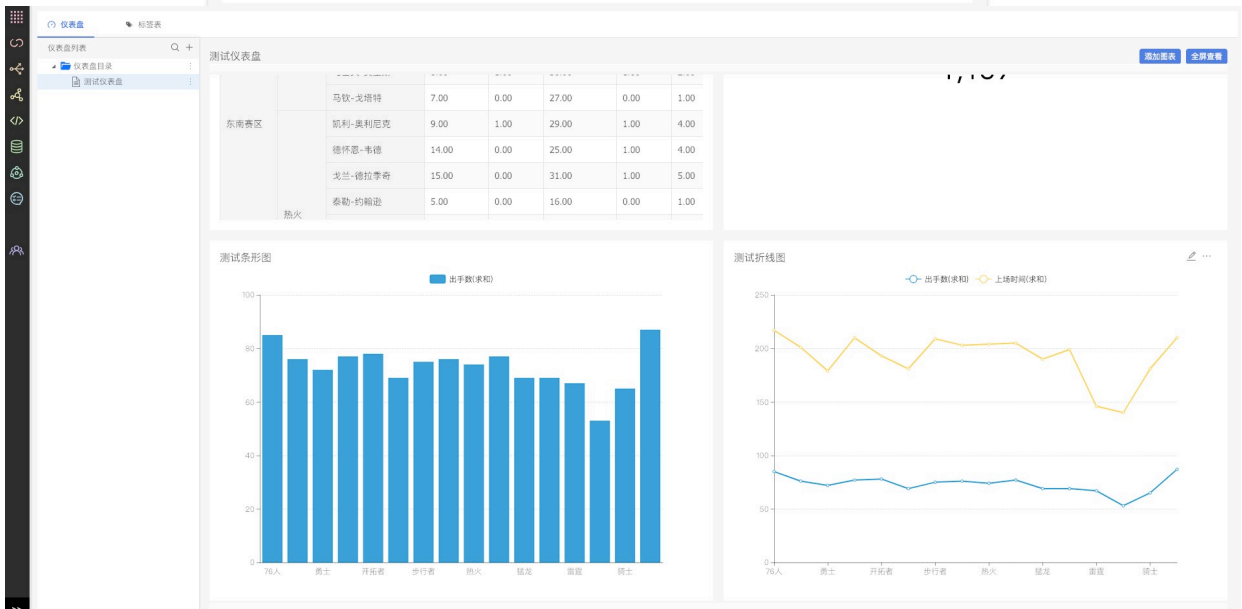
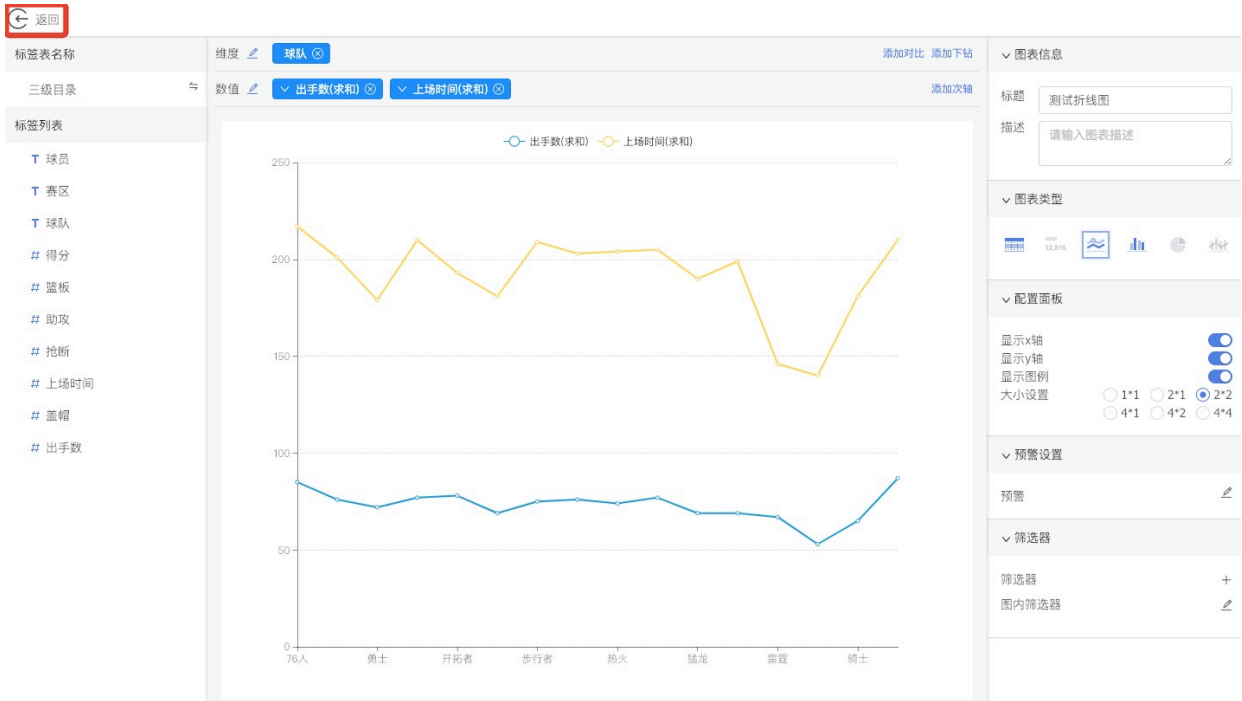
2.在右侧配置区域的图表类型中，选择折线图组件。



3.根据需要，在配置面板中对折线图进行调整，包括是否显示x轴、是否显示y轴，是否显示图例等。

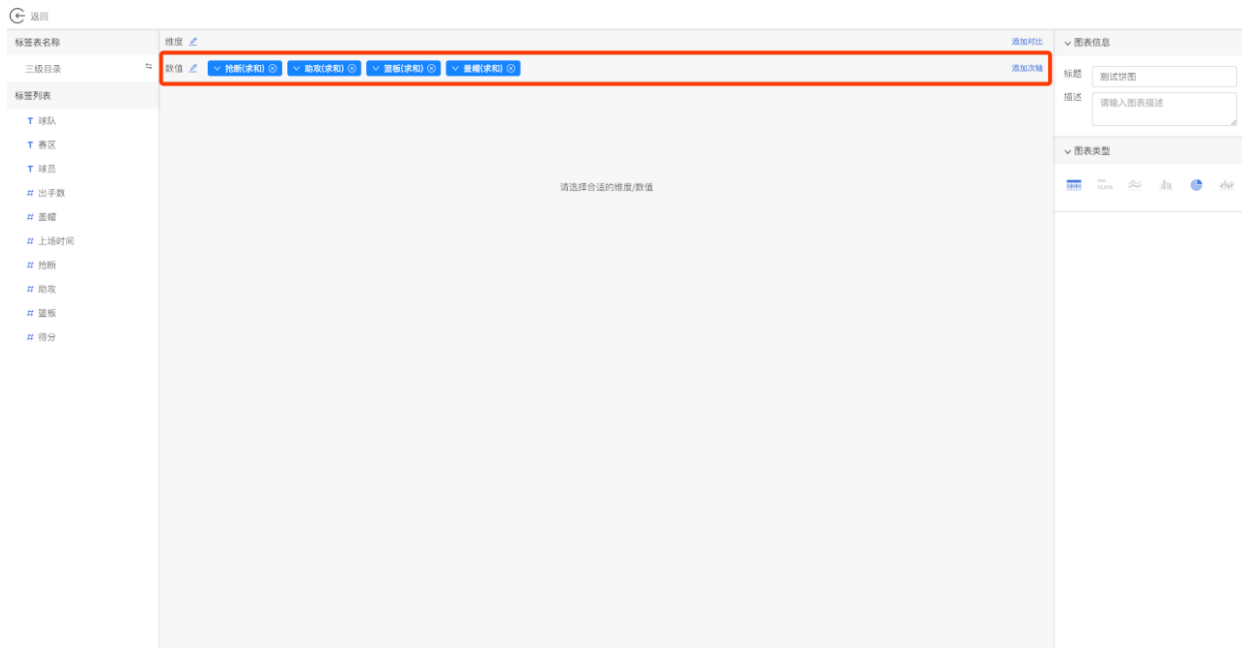


4. 点击返回按钮，返回仪表盘页面，可以看到折线图已经配置完成。

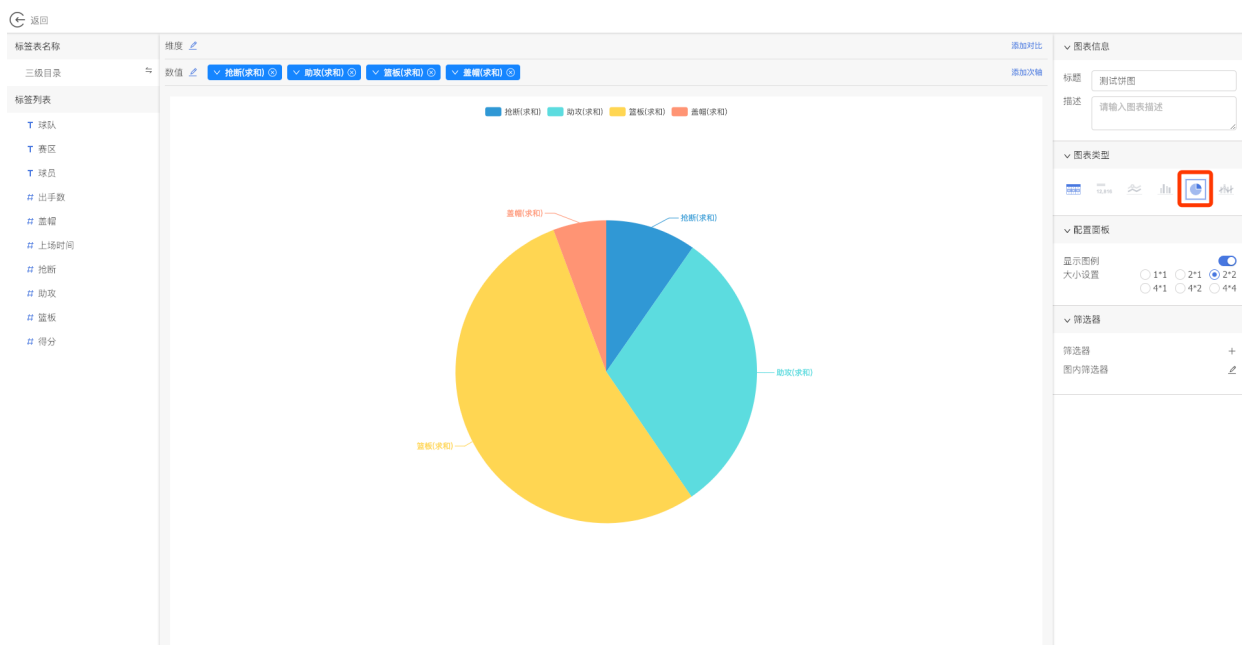


11.3.18 饼图组件

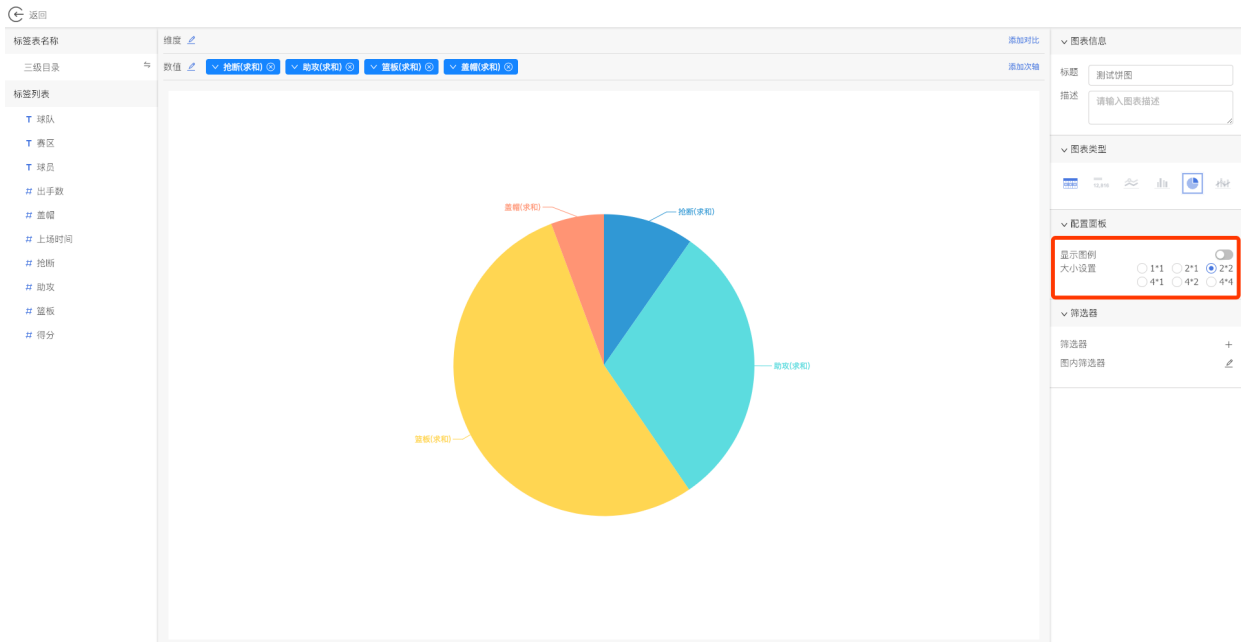
1.在图表编辑页面，维度行选择1个标签，数值行选择1个标签。或者维度行选择0个标签，数值行选择多个标签。



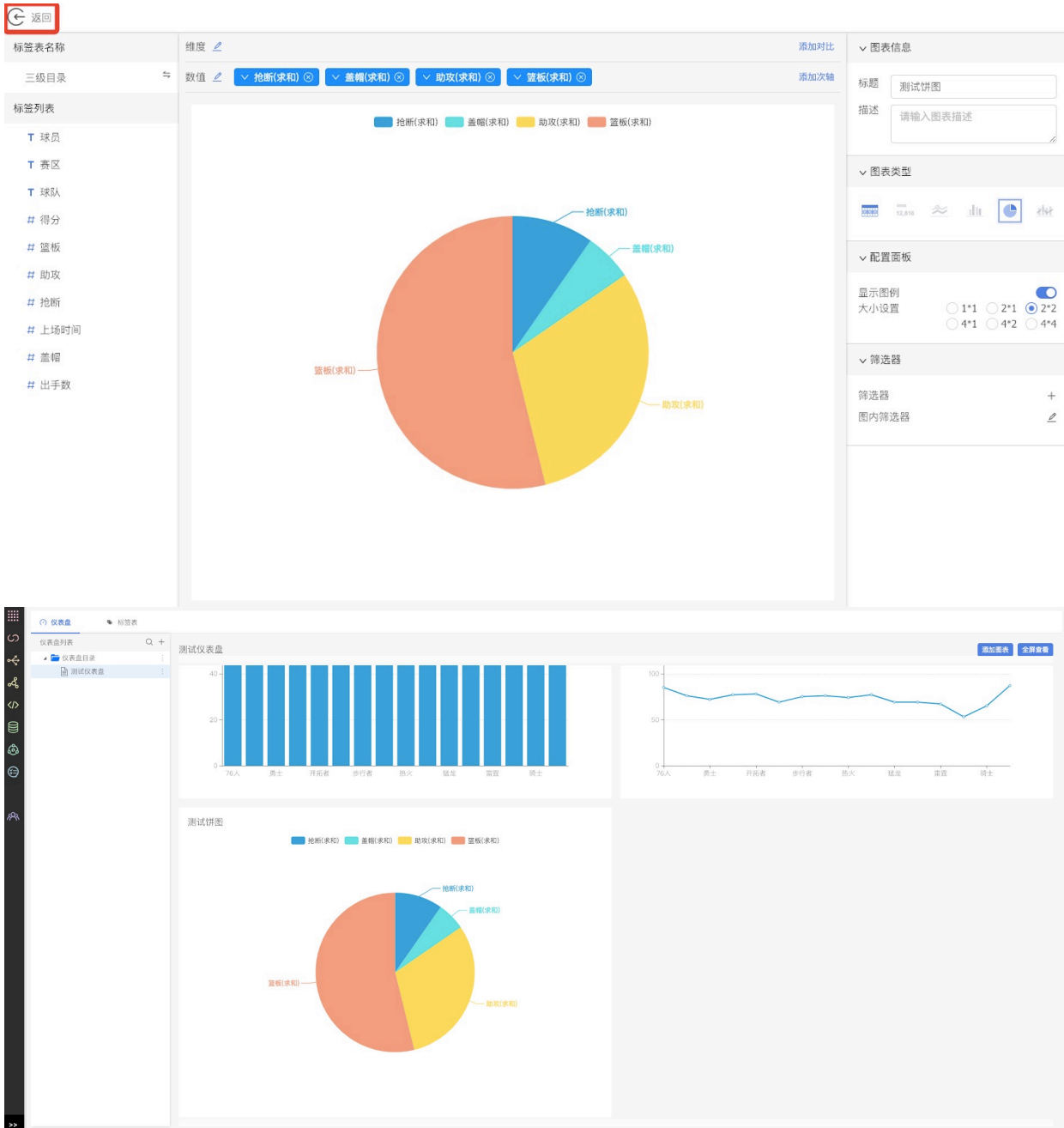
2. 在右侧配置区域的图表类型中，选择饼图组件。



3. 根据需要在配置面板中，可以对图表进行调整，比如是否显示图例等。

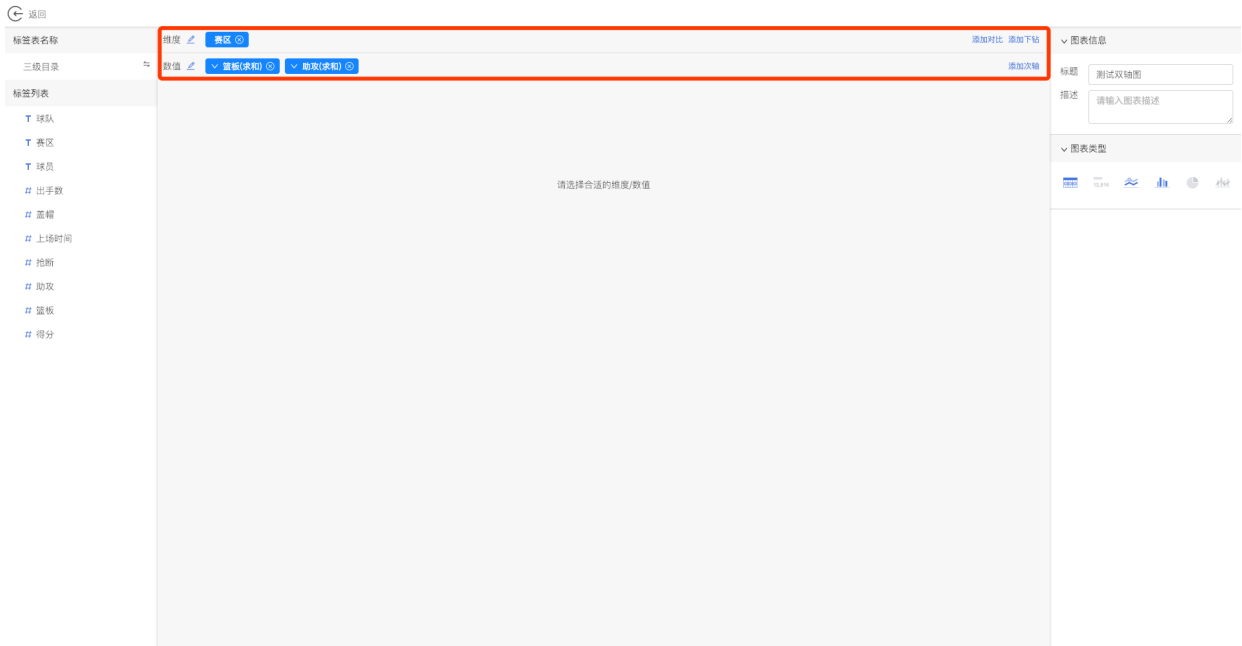


4. 点击返回按钮，返回仪表盘页面，可以看到饼图配置成功。

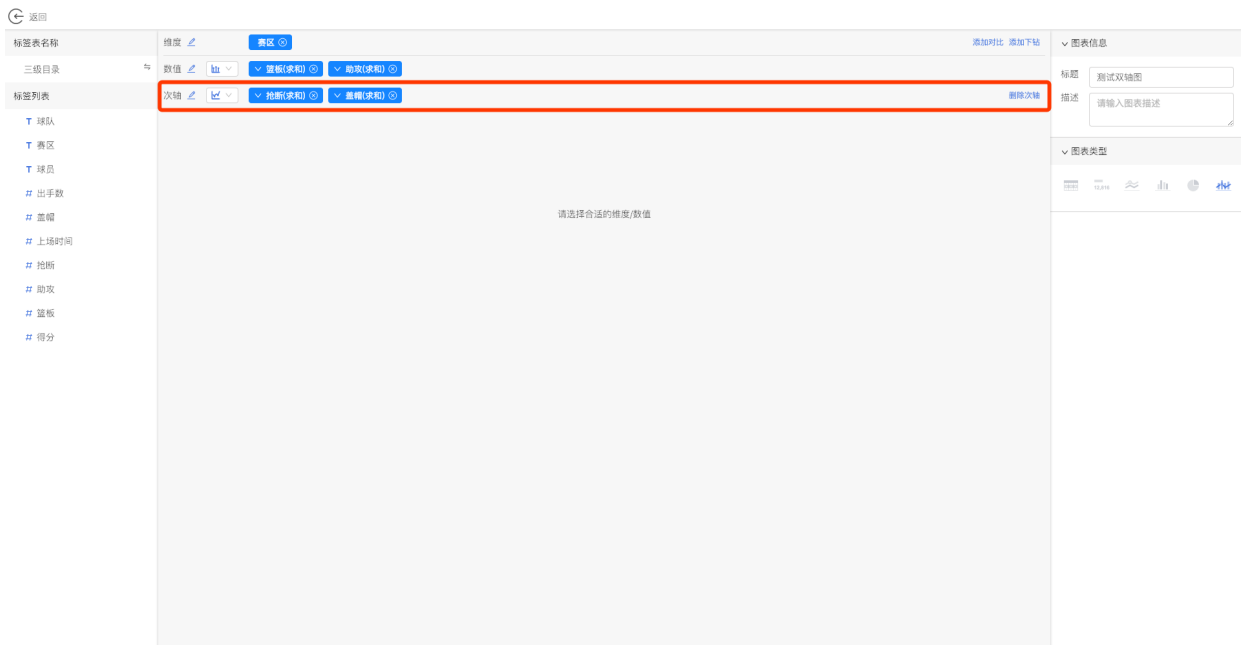


13.3.19 双轴图组件

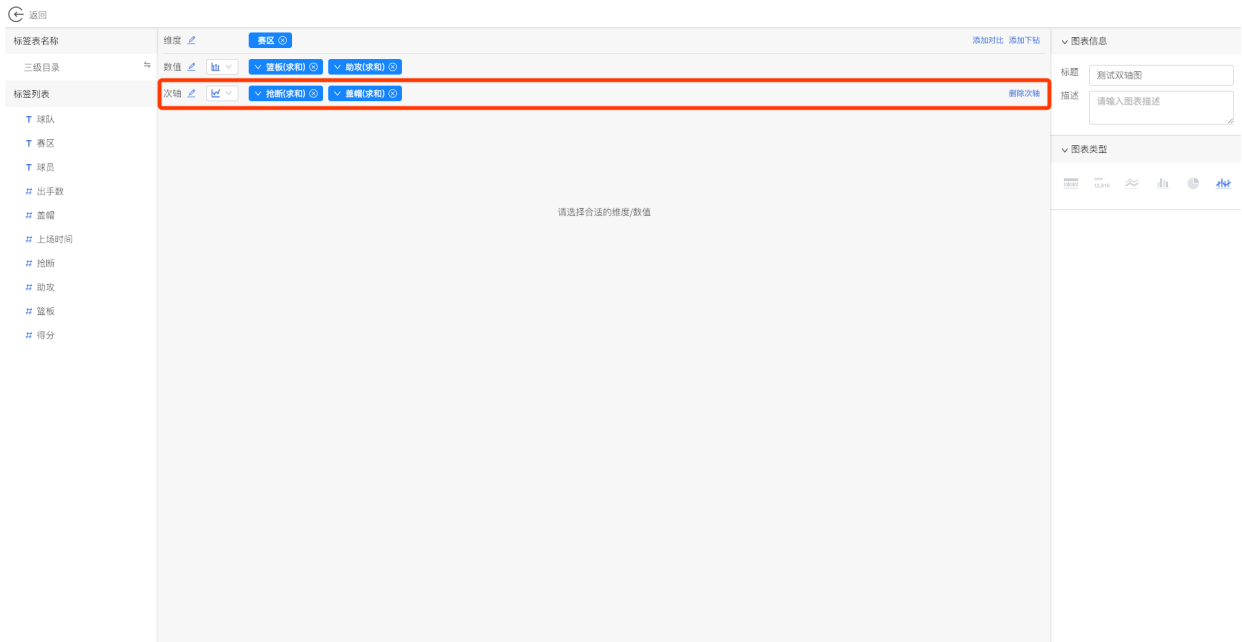
1.在图表编辑页面，维度行选择1个标签，数值行选择1个或多个标签。



2. 点击添加次轴按钮。



3. 在次轴中选择1个或多个标签。



4.在配置区域的图表类型中选择双轴图组件。



5.可以自定义选择纵轴和次轴的图表类型。

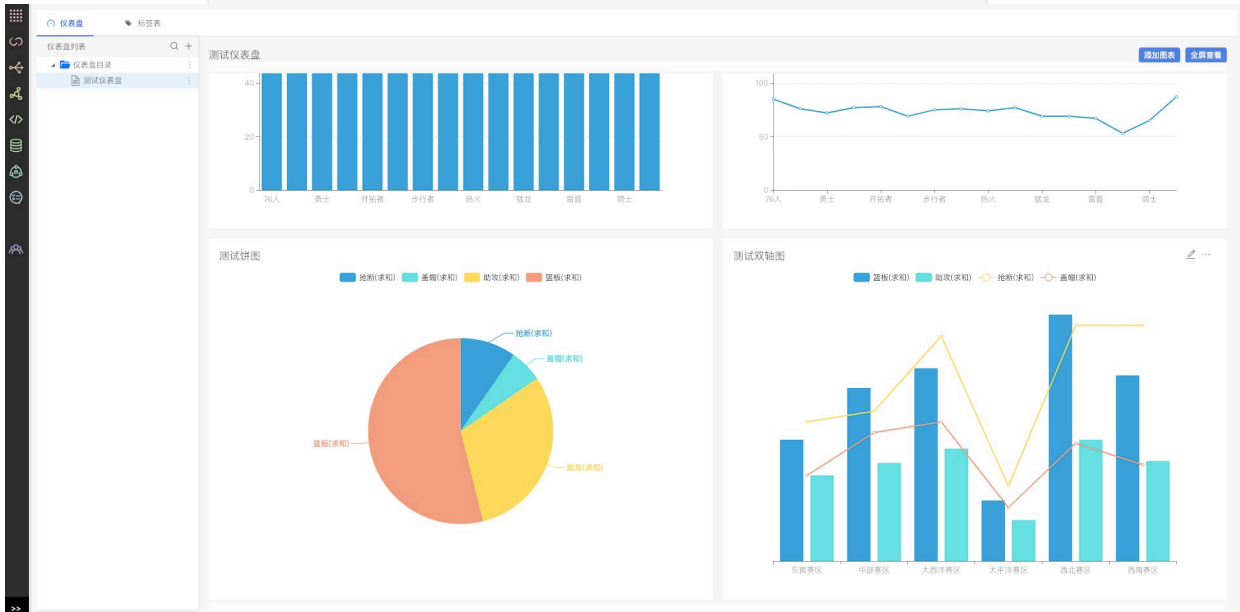


6.根据需要在配置面板中，可以对图表进行调整，比如是否显示x轴、是否显示主轴、是否显示次轴、是否显示图例等。



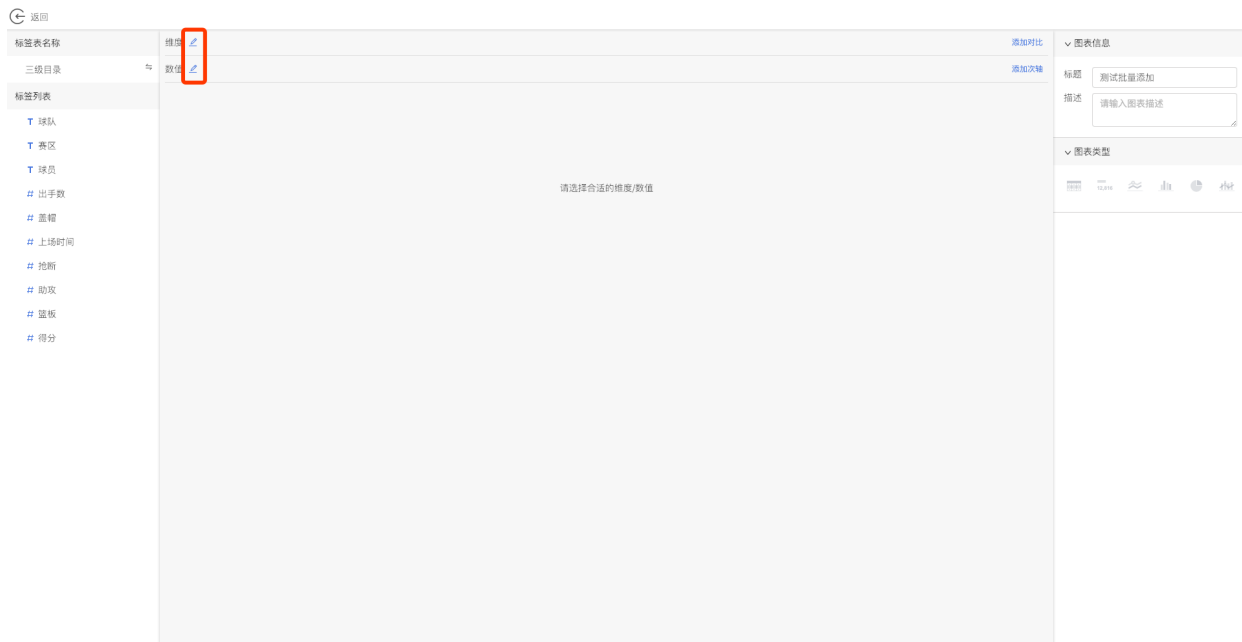
7.点击返回按钮，返回仪表盘页面，可以看到双轴图配置成功。

返回

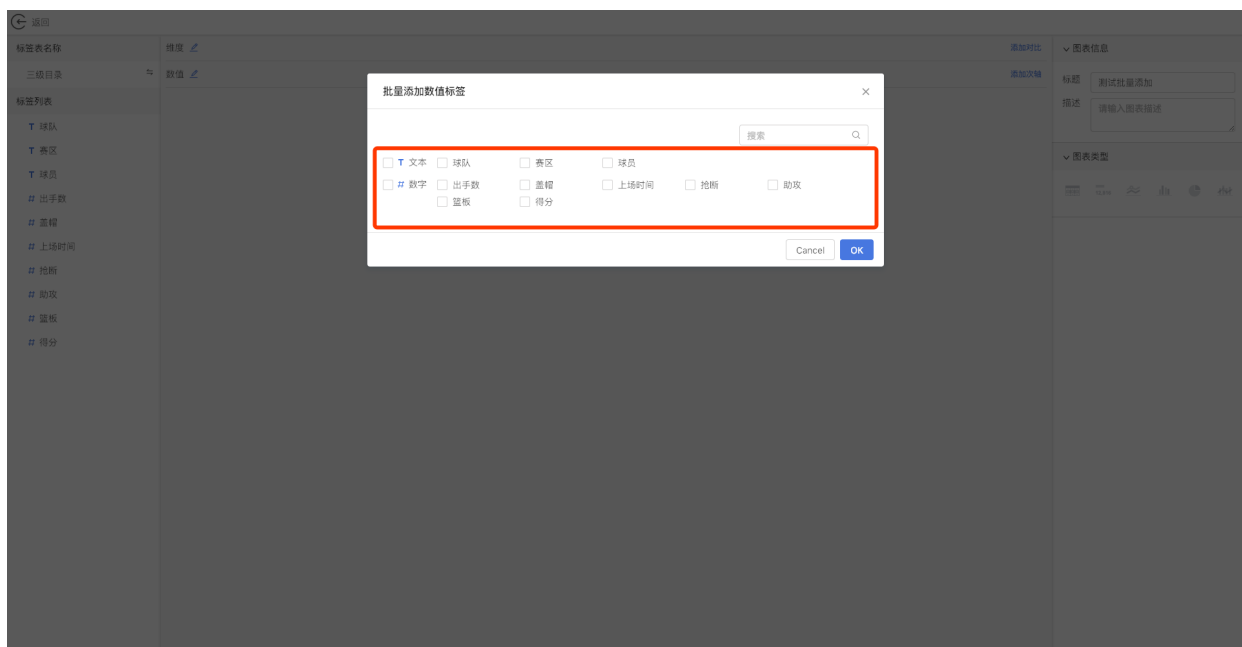


13.3.20 批量选择标签

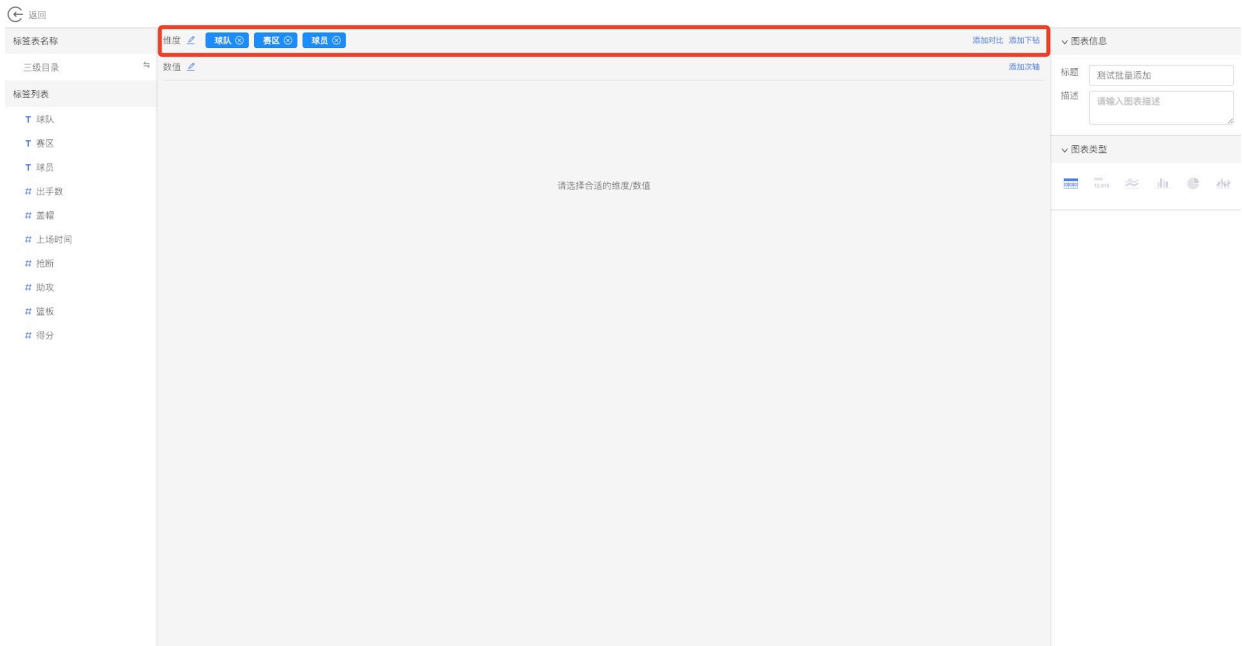
1.在维度行或者数值行，点击批量添加按钮。



2.在批量添加弹窗中，选择要添加的标签。

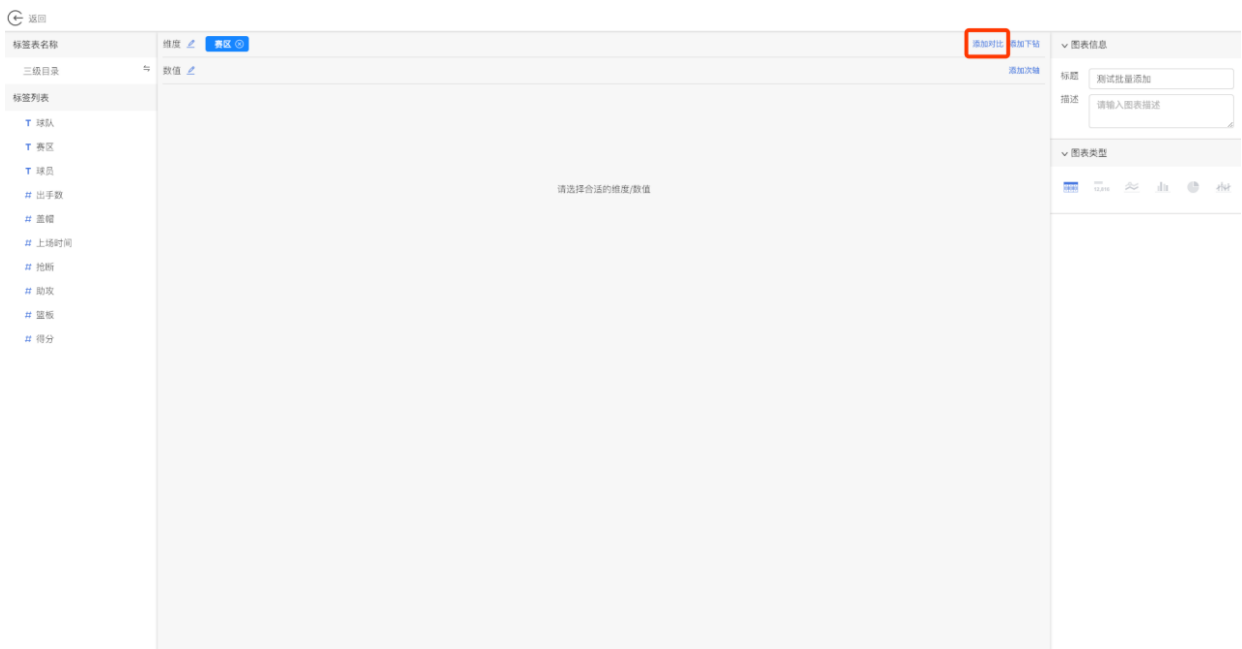


3.点击确定后，可以看到标签批量添加成功。

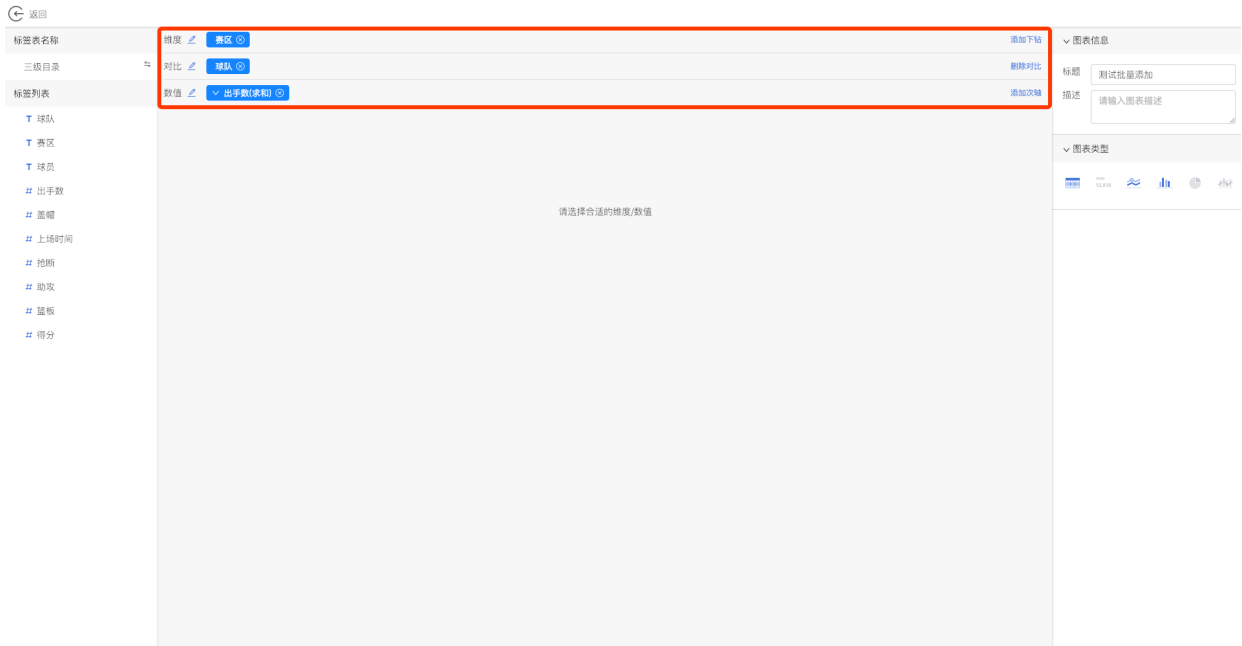


13.3.21 维度对比

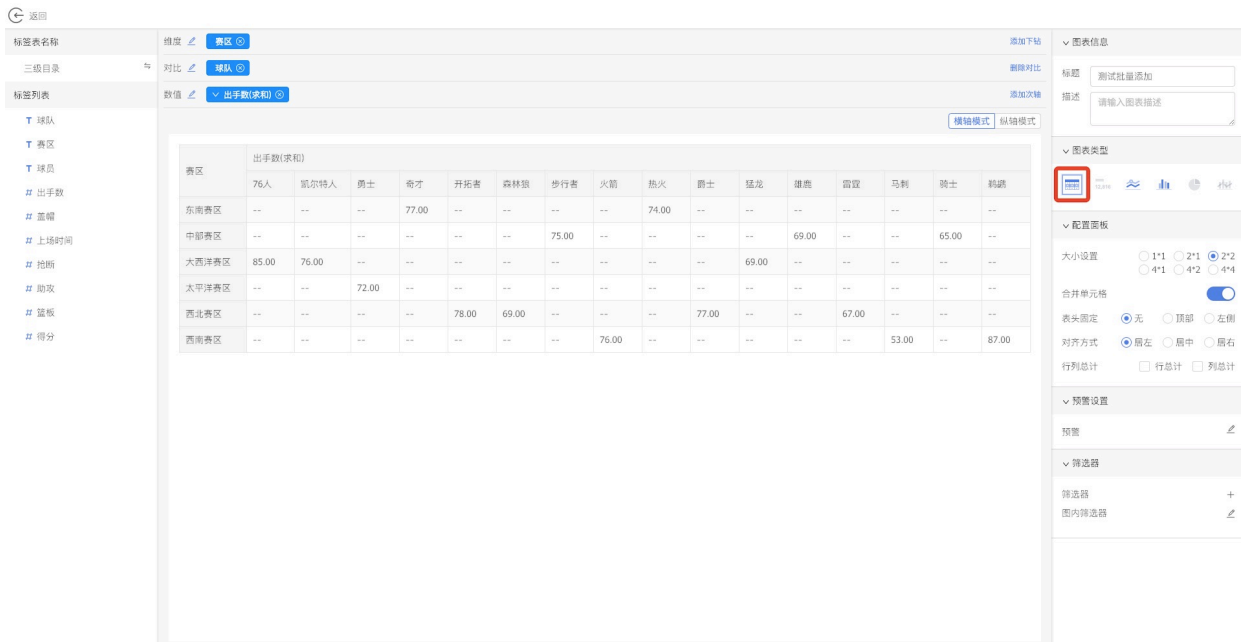
1.在图表编辑页面，拖入维度标签后，点击添加对比按钮。



2.在对比行中选择1个或多个标签。



3.在右侧配置区域的图表类型中，选择一个图表组件，就可以看到配置好的维度对比图表。



13.3.22 设置数值格式

1.编辑图表时，选择要设置格式的数值，点击下拉按钮。

返回

标签表名称 维度 赛区 球队 添加对比 添加下钻

数值 出手数(求和)

横轴模式 纵轴模式

图信息

标题 测试数据添加

描述 请输入图表描述

图类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

筛选器 +

图内筛选器

赛区	球队	出手数(求和)
东南赛区	奇才	77
	热火	74
	步行者	75
中部赛区	雄鹿	69
	骑士	65
	76人	85
大西洋赛区	凯尔特人	76
	猛龙	69
	勇士	72
太平洋赛区	开拓者	78
	森林狼	69
	爵士	77
西北赛区	雷霆	67
	火箭	76
	马刺	53
西南赛区	鹈鹕	87

2.在下拉项中选择数值显示格式。

返回

标签表名称 维度 赛区 球队 添加对比 添加下钻

数值 出手数(求和)

横轴模式 纵轴模式

图信息

标题 测试数据添加

描述 请输入图表描述

图类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

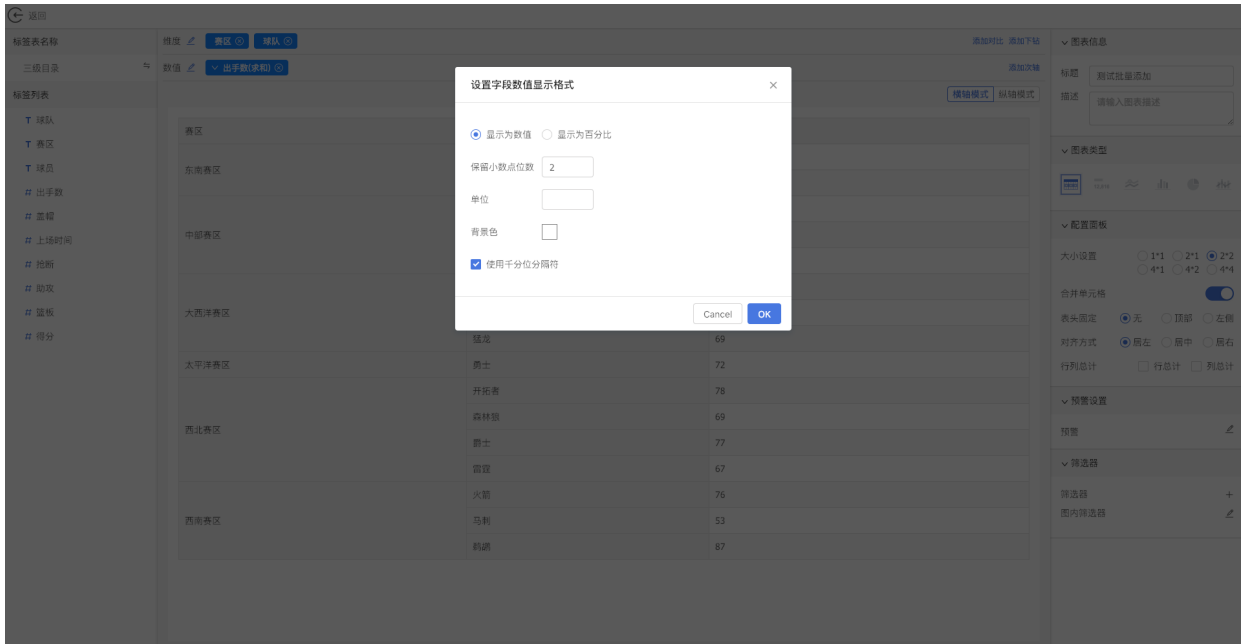
筛选器 +

图内筛选器

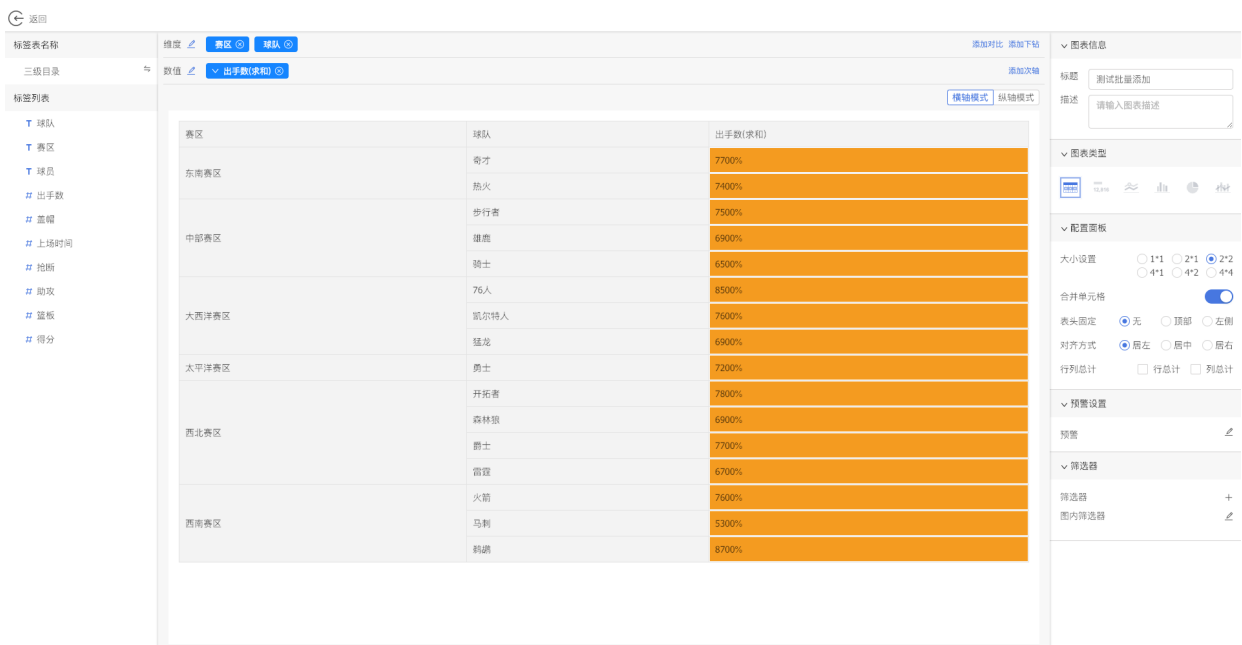
求和
求平均数
计数
去重计数
最大值
最小值
数值显示格式

赛区	球队	出手数(求和)
东南赛区	奇才	77
	热火	74
	步行者	75
中部赛区	雄鹿	69
	骑士	65
	76人	85
大西洋赛区	凯尔特人	76
	猛龙	69
	勇士	72
太平洋赛区	开拓者	78
	森林狼	69
	爵士	77
西北赛区	雷霆	67
	火箭	76
	马刺	53
西南赛区	鹈鹕	87

3.在数值显示格式弹窗中可以设置数值类型、小数点位数、背景色、数值单位、千分位等。



4.设置完成后，点击**确定按钮**，可以看到数值格式设置成功。



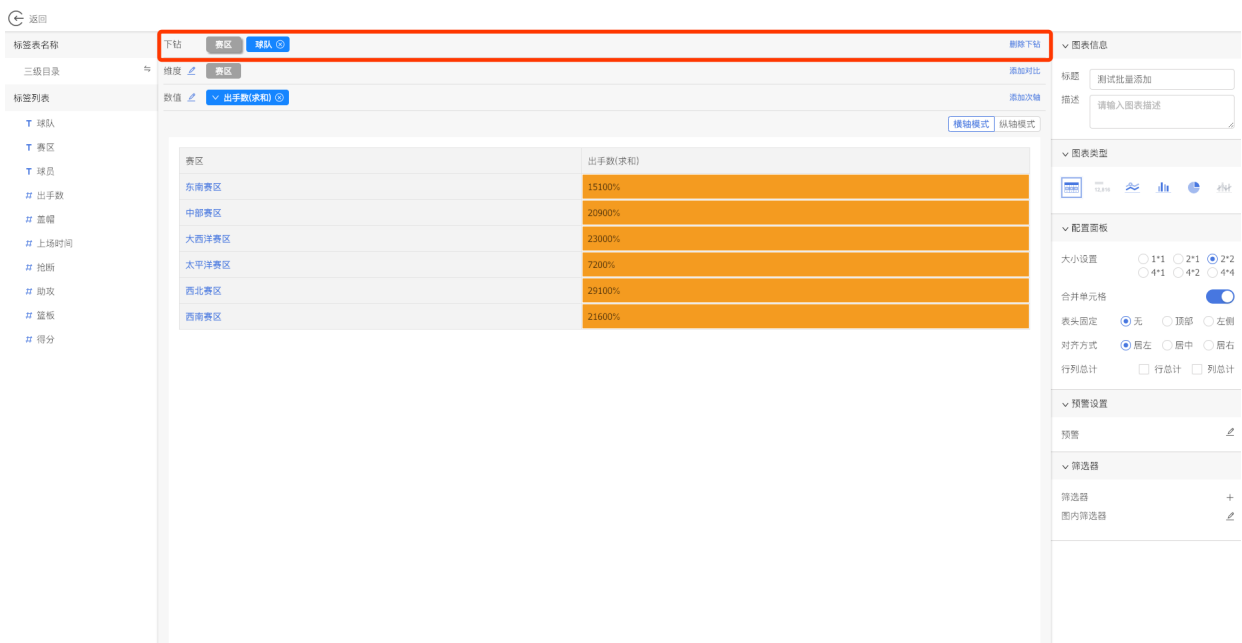
背景色只对表格类图表起作用。

13.3.23 图表下钻

1.在图表编辑页面，在已经配置好的图表基础上，点击**添加下钻按钮**。



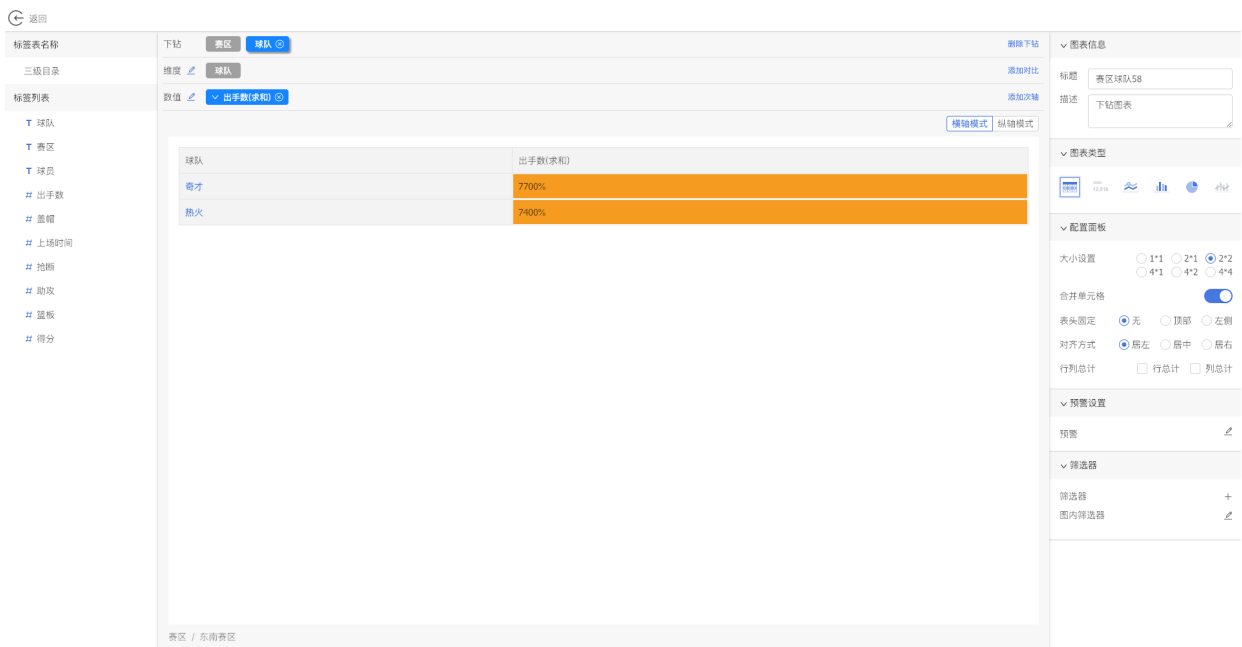
2.在下钻行中拖入要下钻的标签，拖入多个就是多层下钻。



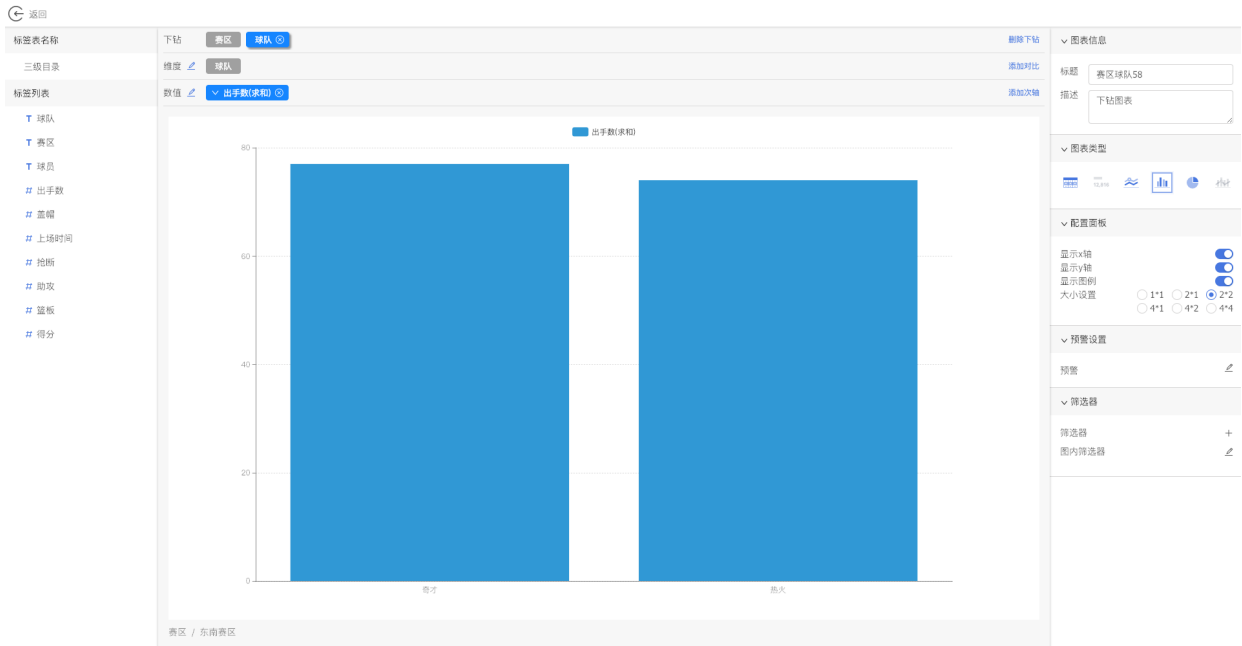
3.点击要查看的下钻部分。



4.就可以看到下钻后的数据。

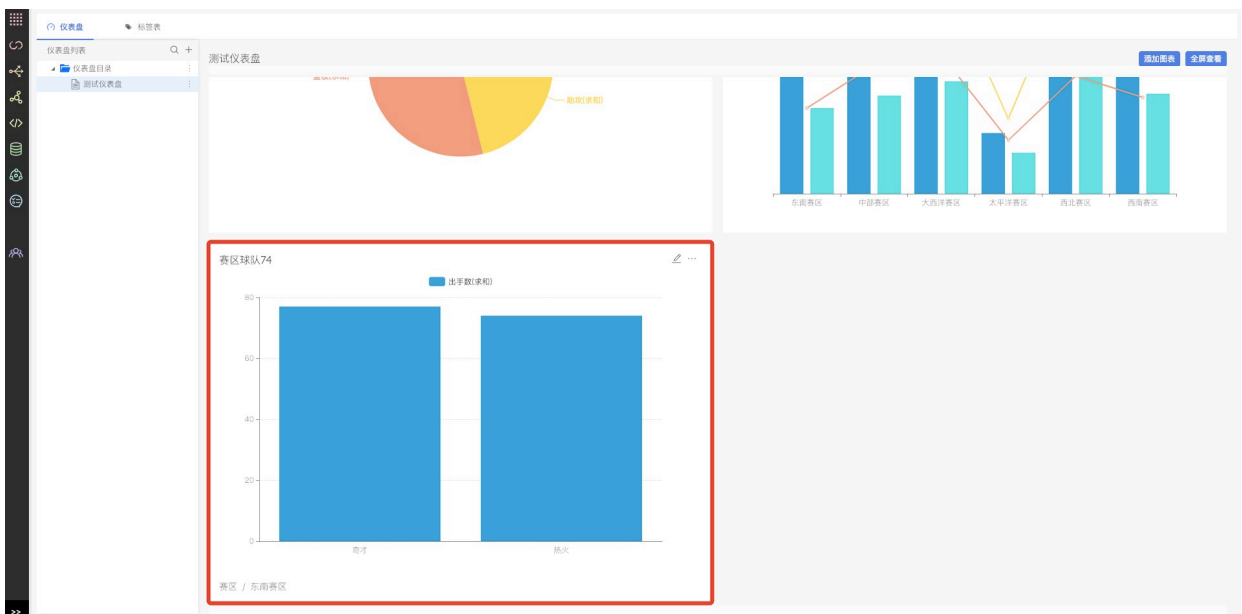
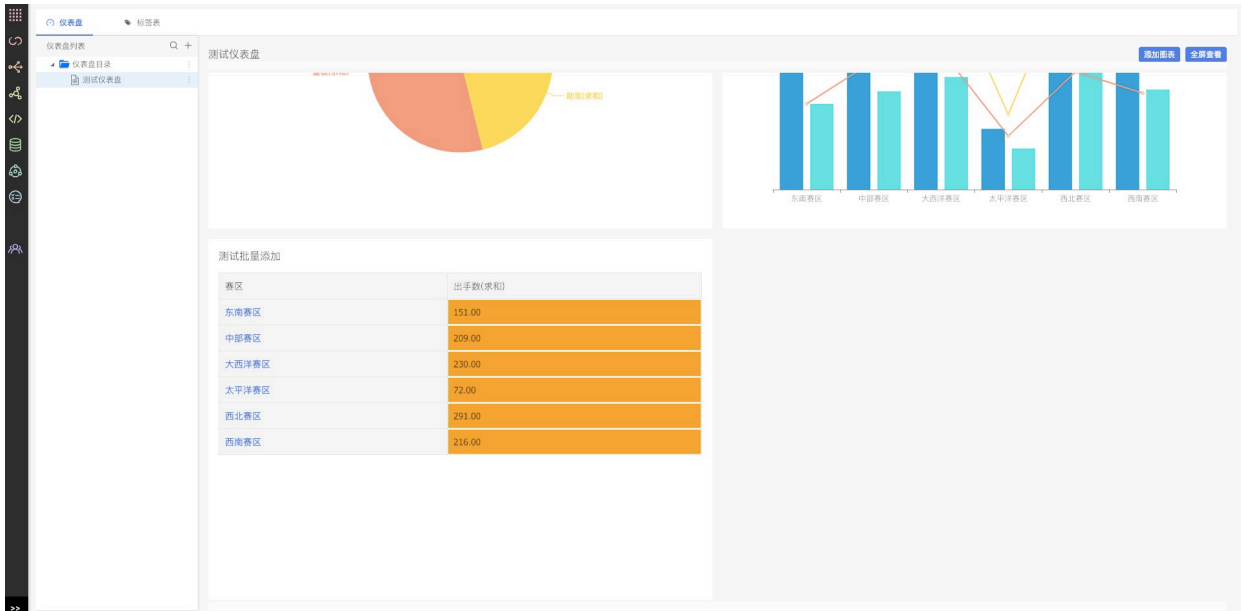


5.下钻后的图表可以重新配置图表类型。



6.返回仪表盘页面，点击要下钻的部分，查看下钻效果。



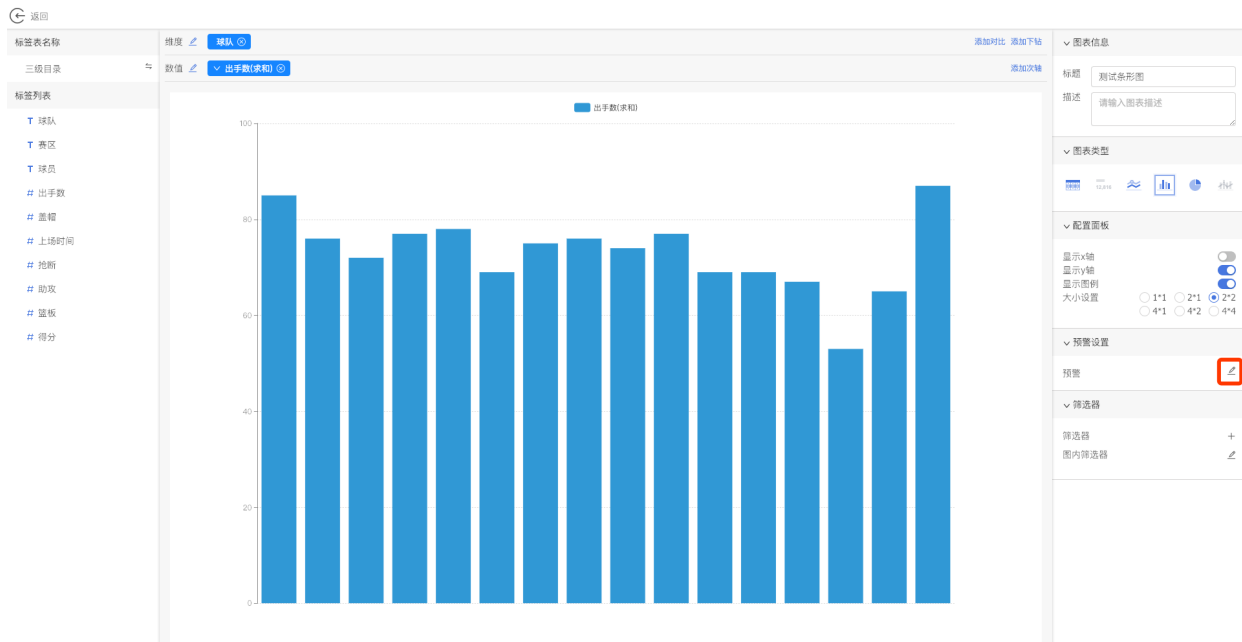


注意：

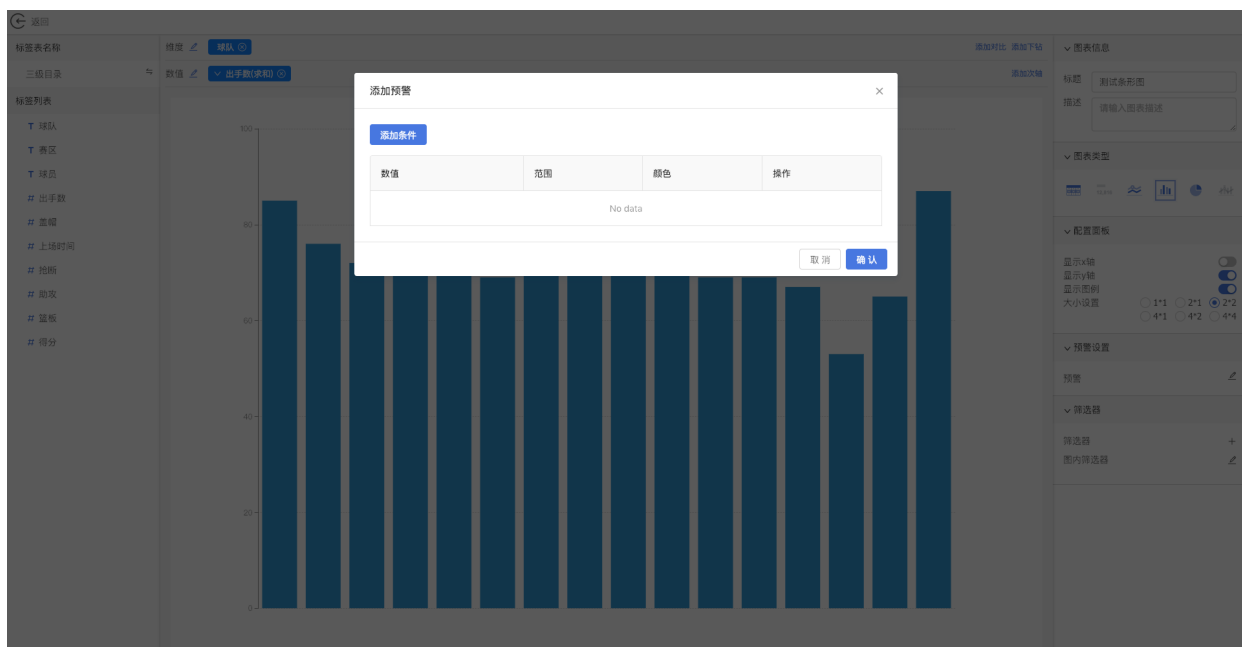
- 1) 在下钻行中拖入多个标签，图表下钻的顺序按标签顺序进行下钻。
- 2) 下钻只能对拖入的第一个维度标签进行下钻，设置下钻的时候，该标签置灰不可改。

13.3.24 图表预警

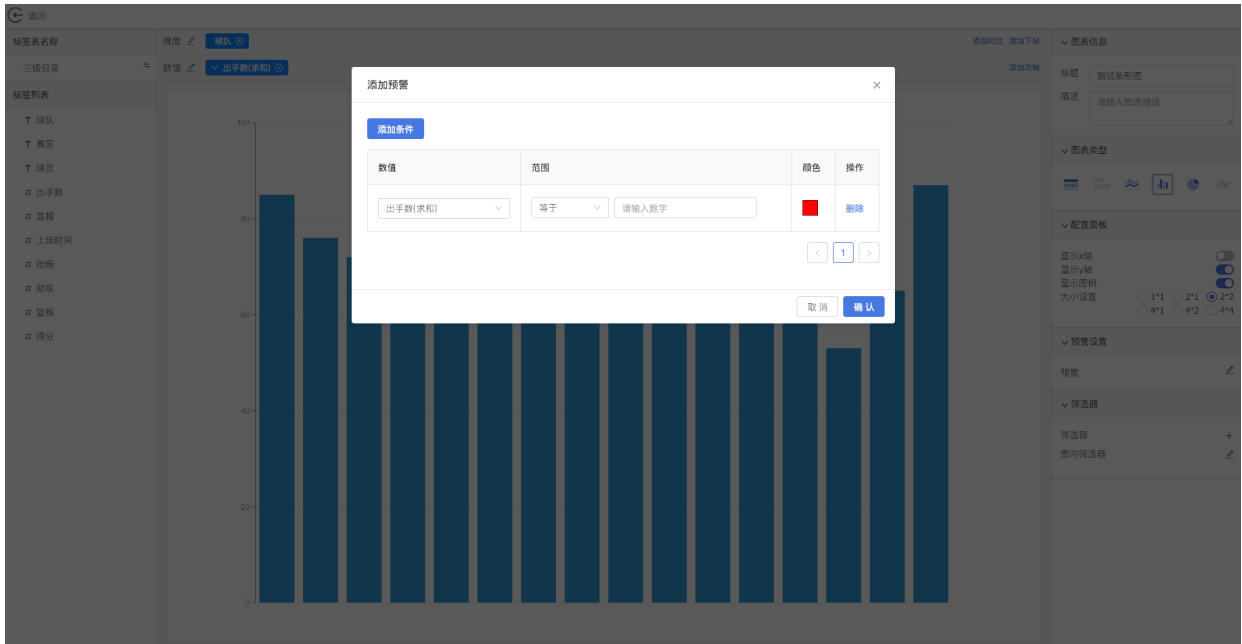
- 1.在编辑图表页面，配置区域的预警设置中，点击编辑预警按钮。



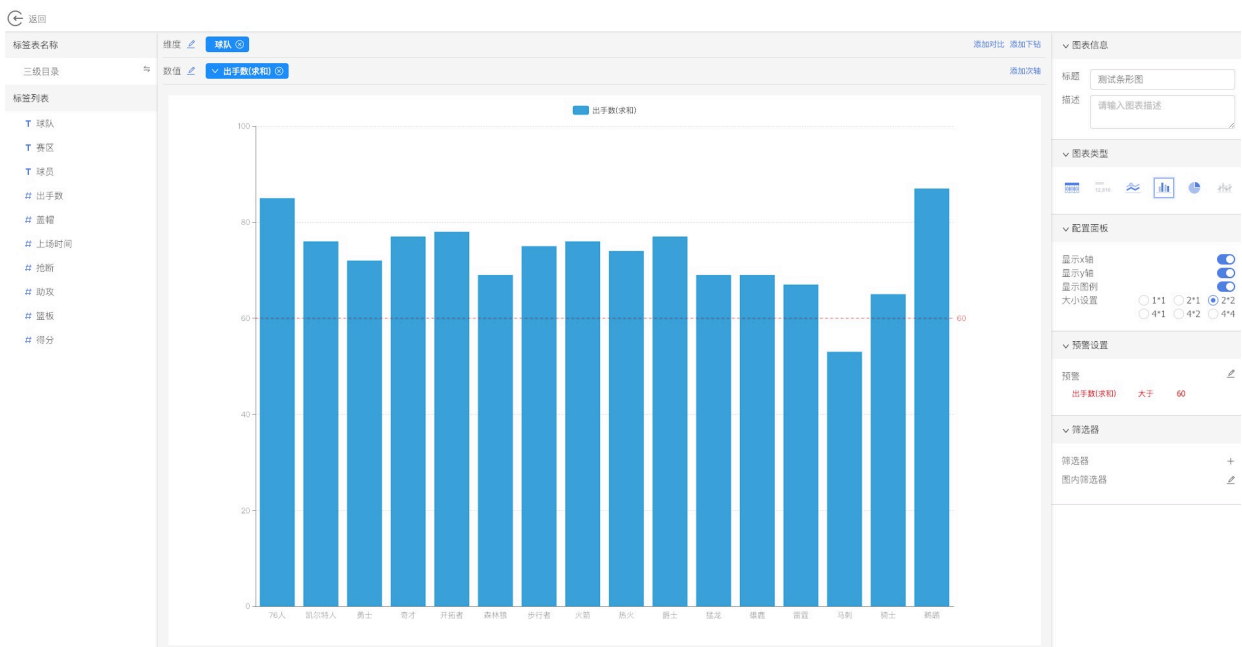
2.在添加预警弹窗中，点击添加条件按钮。



3.设置预警条件和预警色后，点击确定按钮。



4.可以在图上看到预警效果。



注意:

- 1) 极坐标类图表暂不支持预警。
- 2) 表格类图表设置预警后，触发预警规则的数据会以预警色显示。
- 3) 其他图表设置预警后，会在图表上通过预警色标识出预警线。
- 4) 只能对数值类标签设置预警，其他标签不能设置预警规则。数值类标签预警条件包括大于、小于、大于等于、小于等于、等于、不等于和区间这几种方式。
- 5) 当一张图表设置了预警后，这个时候把对应的标签的计算方式改掉或者去掉这个标签。相应的预警规则会失效，在页面中对应的预警规则会有一条删除线。

13.3.25 图表大小设置

1.在编辑图表页面，配置区域的配置面板中，设置图表大小。默认大小是宽×高：2×2。

返回

标签表名称 维度 赛区 球队 球员 添加列比 添加下钻

三级目录 数值 出手数(求和) 篮板(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和) 添加滤镜

标签列表 横轴模式 纵轴模式

赛区	球队	球员	出手数(求和)	篮板(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)	行总计
东南赛区	奇才	凯利-乌布雷	8	1	25	1	1	4	40
		奥托-波特	8	1	32	1	2	5	49
		布拉德利-比尔	18	0	36	1	3	3	61
		约翰-沃尔	21	1	39	2	12	6	81
		迈克-斯科特	7	0	21	0	1	4	33
		马基夫-莫里斯	8	1	30	1	2	8	50
	马钦-诺斯特	7	0	27	0	1	6	41	
	热火	凯利-奥克利克	9	1	29	1	4	5	49
		德怀恩-韦德	14	0	25	1	4	4	48
		戈兰-德拉季奇	15	0	31	1	5	3	55
		泰勒-约翰逊	5	0	16	0	1	2	24
		约什-理查德森	8	1	26	2	3	3	43
詹姆斯-约翰逊		8	1	32	1	5	6	53	
步行者	贾斯佩斯-温斯洛	8	1	25	1	3	7	45	
	韦恩-埃灵顿	7	0	20	0	1	2	30	
	兰斯-史蒂芬森	9	0	21	0	3	3	36	
	博扬-博格达诺维奇	12	0	34	1	2	3	52	
	多曼塔斯-萨博尼斯	9	0	24	0	1	5	39	
	维克托-奥拉迪波	18	0	37	2	6	8	71	

图表信息

标题 测试图表

描述 请输入图表描述

图表类型

配置面板

大小设置 1*1 2*1 2*2 4*1 4*2 4*4

合并单元格

表头固定 无 顶部 左侧

对齐方式 居左 居中 居右

行列总计 行总计 列总计

预警设置

预警

筛选器

图内筛选器

2.返回仪表盘页面，可以看到图表大小已经发生变化。

返回
标签表名称
维度 赛区 球队 球员
添加对比 添加下钻

三级目录
数值 出手数(求和) 盖帽(求和) 上场时间(求和) 抢断(求和) 助攻(求和) 篮板(求和)
添加次轴

图内筛选器 请选择赛区
横轴模式 纵轴模式

赛区	球队	球员	出手数(求和)	盖帽(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)
西南赛区	火箭	PJ-塔克	6.00	1.00	34.00	1.00	1.00	7.00
		克林特-卡佩拉	9.00	2.00	31.00	1.00	1.00	12.00
		克里斯-保罗	17.00	0.00	35.00	2.00	6.00	6.00
		埃里克-戈登	13.00	1.00	32.00	1.00	2.00	3.00
		特雷沃-阿里扎	8.00	0.00	34.00	1.00	1.00	4.00
		詹姆斯-哈登	23.00	1.00	37.00	2.00	7.00	5.00
		帕蒂-米尔斯	11.00	0.00	33.00	1.00	3.00	2.00
西南赛区	马刺	德章泰-穆雷	6.00	0.00	19.00	1.00	2.00	4.00
		拉马库斯-阿尔德里奇	16.00	0.00	35.00	1.00	2.00	9.00
		马努-吉诺比利	7.00	0.00	21.00	1.00	3.00	3.00
		鲁迪-盖伊	13.00	0.00	32.00	2.00	2.00	6.00
		乔丹-克劳福德	7.00	0.00	8.00	1.00	1.00	2.00
伊安-克拉克	7.00	0.00	21.00	1.00	1.00	1.00		

测试图表
添加名称 全屏查看

赛区	球队	球员	出手数(求和)	盖帽(求和)	上场时间(求和)	抢断(求和)	助攻(求和)	篮板(求和)
东南赛区	奇才	凯利-乌布雷	8.00	1.00	25.00	1.00	1.00	4.00
		奥托-波特	8.00	1.00	32.00	1.00	2.00	5.00
		布拉德利-比尔	18.00	0.00	36.00	1.00	3.00	3.00
		约翰-沃尔	21.00	1.00	39.00	2.00	12.00	6.00
		迈克-斯科特	7.00	0.00	21.00	0.00	1.00	4.00
		马基夫-莫里斯	8.00	1.00	30.00	1.00	2.00	8.00
		马钦-戈登	7.00	0.00	27.00	0.00	1.00	6.00
		凯利-奥利尼克	9.00	1.00	29.00	1.00	4.00	5.00
		德怀恩-韦德	14.00	0.00	25.00	1.00	4.00	4.00
		戈兰-德拉季奇	15.00	0.00	31.00	1.00	5.00	3.00
泰勒-约翰逊	5.00	0.00	16.00	0.00	1.00	2.00		

测试指标卡

测试条形图

注意：

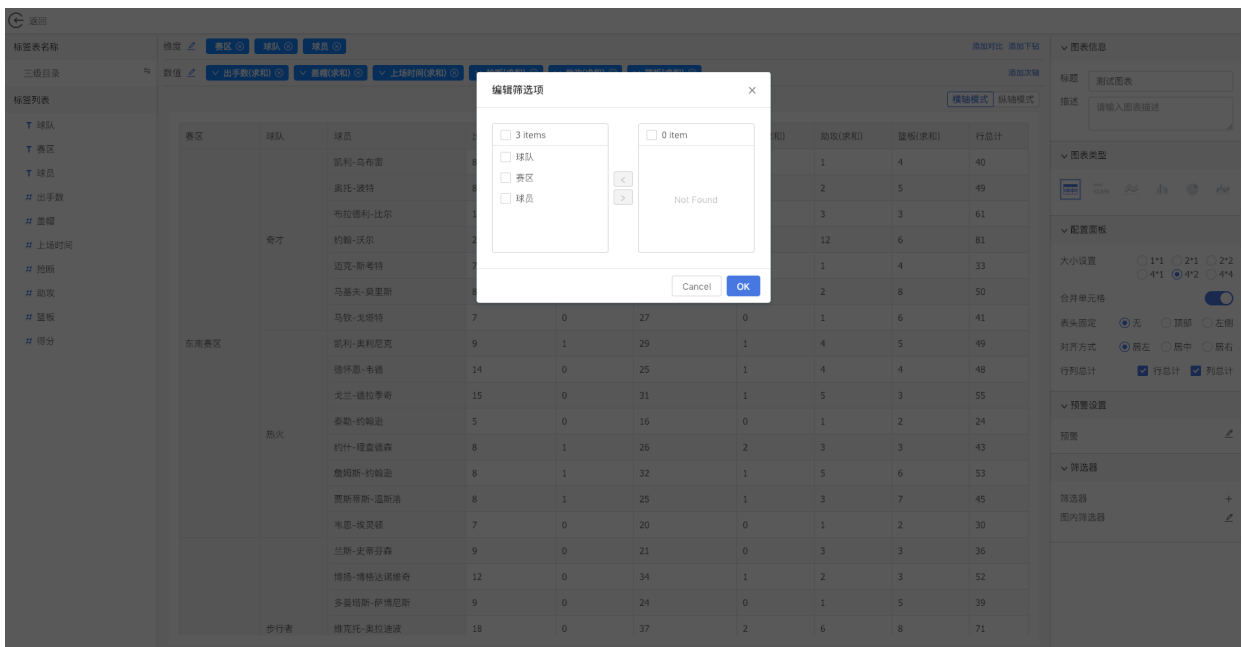
- 1) 图表大小目前不允许随意定制，只能按照等比例放大缩小。
- 2) 当前提供1×1、2×1、2×2、4×1、4×2、4×4六种模式的宽高比。图表默认宽高比是2×2
- 3) 请根据实际需要配置图表大小，否则会导致配置的仪表盘不是很美观。

13.3.26 图内筛选器

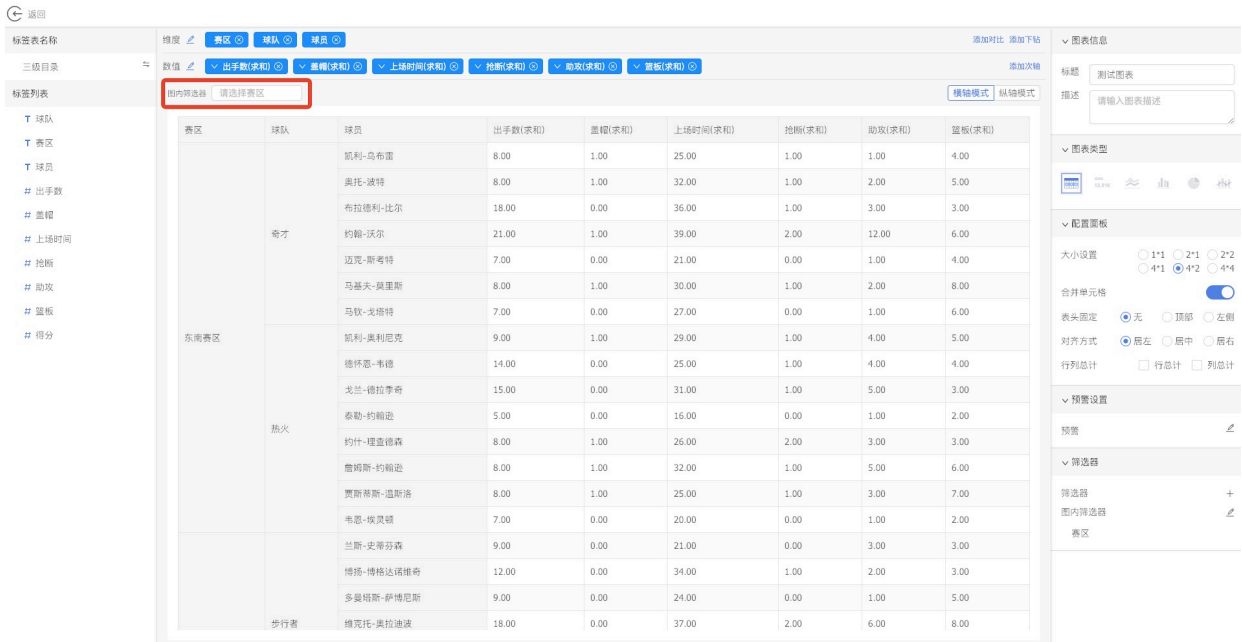
1.在图表编辑页面，配置区的图内筛选器中，点击编辑图内筛选器按钮。



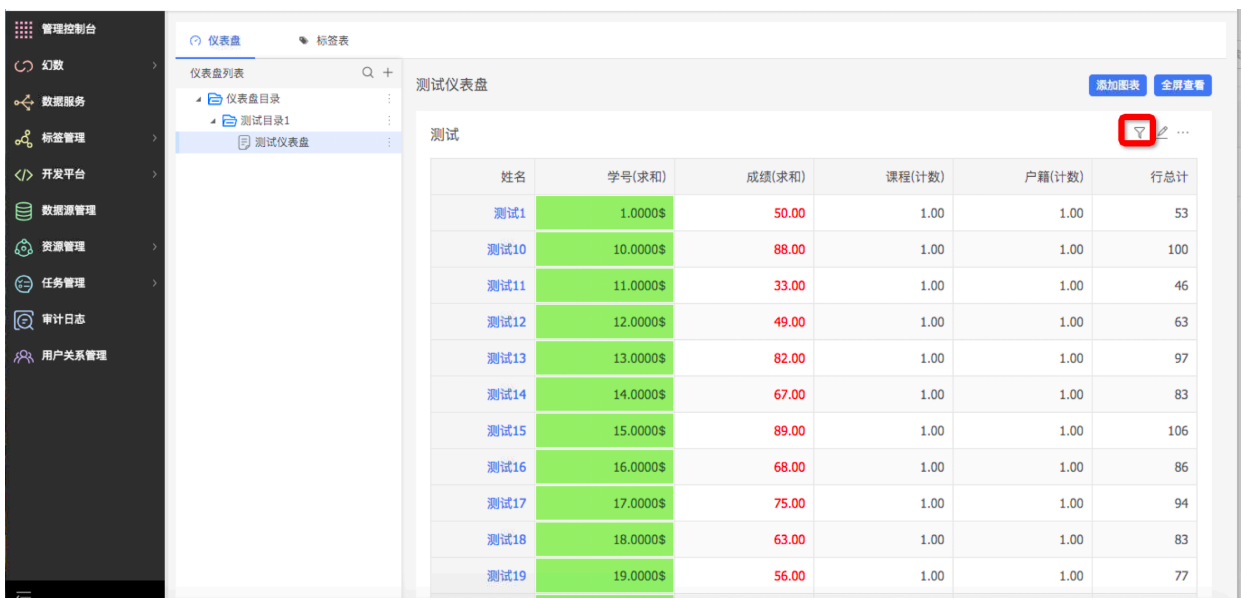
2.选择要筛选的维度，点击添加按钮。



3.点击确定后，可以在图表中看到添加的图内筛选器。



4.返回仪表盘页面。点击图内筛选器按钮，显示筛选项，选择对应的筛选项就可以筛选图表数据。



注意：

- 1) 设置图内筛选器后，为防止所有数据被拉取，分析引擎设置了最多1000个筛选。请选择合适的筛选场景。
- 2) 图内筛选器只支持选择日期类和文本类标签。
- 3) 选择文本类标签时，图内筛选器的下拉项是该标签的标签值（去重）
- 4) 选择日期类标签时，图内筛选器的下拉项包括今天、昨天、近一周、本月、上个月、近90天、近365天以及自定义时间。选择自定义时间时，可以选择一段时间跨度。
- 5) 图内筛选器最多可以选择5个标签。

13.3.27 筛选器

1.在图表编辑页面，配置区的筛选器中，点击编辑筛选器按钮。

在编辑筛选器弹窗中，选择筛选条件。

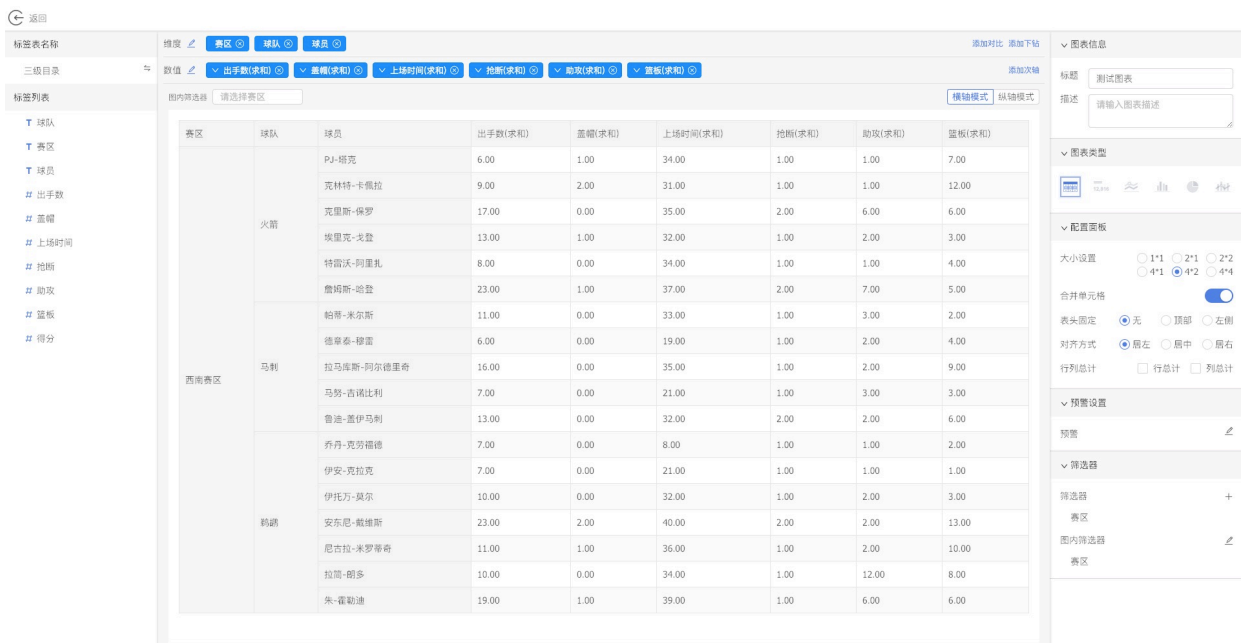
2.在编辑筛选器弹窗中，选择筛选条件。

3.选择要筛选的数据，点击添加按钮。

3.选择要筛选的数据，点击添加按钮。



4. 点击确定后，图表中就只能看到筛选后的数据。



注意：

- 1) 图内筛选器和筛选器的区别：图内筛选器是提供给用户下拉筛选数据，呈现的还是全量数据；筛选器只呈现筛选后的数据。
- 2) 筛选器可以添加多个，也可以重新编辑和删除。
- 3) 筛选器筛选后，图内筛选器中的下拉项也会经过筛选器筛选。
- 4) 筛选器根据选择的标签类型不同，筛选的形式也不同。
- 5) 如果选择的是文本类标签，则需要选择该标签下的标签值（能显示的标签值个数也受分析引擎限制）。比如说文本类标签为部门，选择的筛选项为市场部、产品部、技术部三个标签值，则图表显示的数据就是三个部门所对应的数据。

6) 如果选择的是数值类标签,则需要填写该数值的条件表达式,条件包括大于、小于、大于等于、小于等于、等于、不等于以及区间。比如说数值类标签为销售额,选择的筛选条件为大于1000,则图表显示的数据就是销售额大于1000所对应的数据。

7) 如果选择的是日期类标签,则需要选择日期筛选项,包括今天、昨天、本月、上个月、近一周、近90天、近365天以及自定义时间。比如说日期类为日期,选择的筛选条件为本月,则图表显示的数据就是本月所对应的数据。

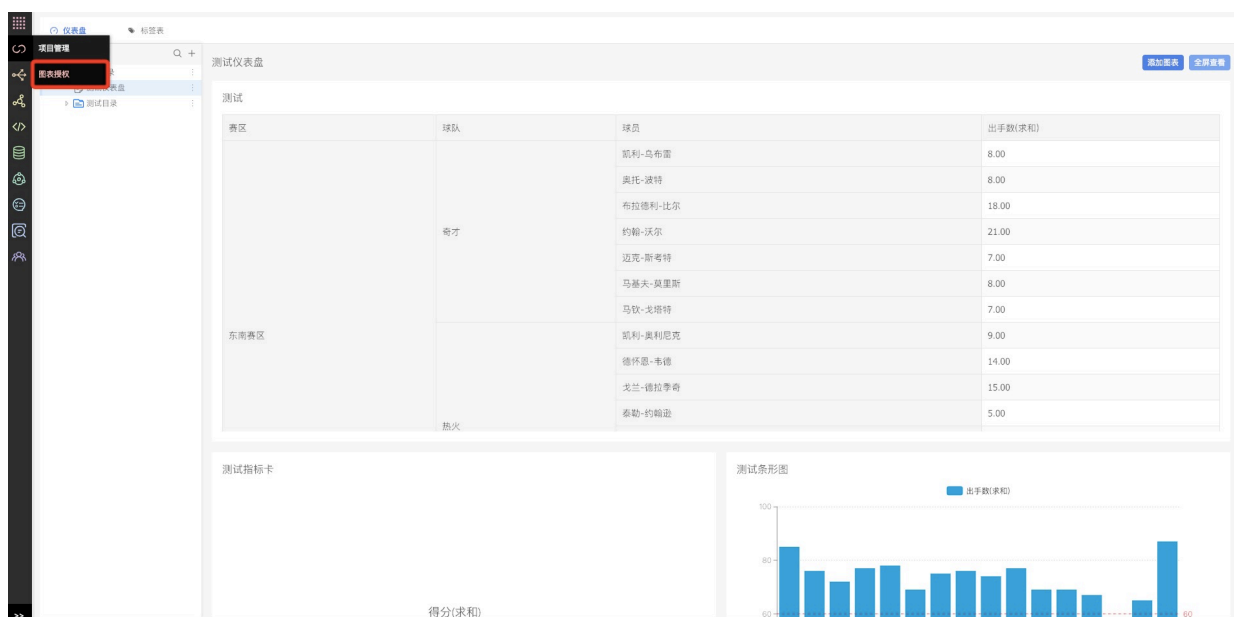
13.4 图表授权

13.4.1 功能说明

用户可以把自己配置好的图表授权给其他用户,其他用户可以查看别人配置的图表。

13.4.2 添加图表授权

1.在左侧菜单栏中,选中幻数二级菜单,图表授权。



2.在图表授权页面,点击>创建授权按钮。

图表授权

创建授权

授权名称	授权描述	创建人	创建时间	操作
最大时代	--	管理员	2018-12-25 11:33:30	编辑 删除
测试授权	--	管理员	2018-12-28 18:03:14	编辑 删除

< 1 >

3.填写授权名称、描述，选择是否开启数据权限、选择要授权的报表后，点击下一步按钮。

创建授权

1 选择授权内容

2 选择授权对象

授权名称

开启数据权限

授权描述

选择报表

可选内容

- 幻数用户手册
 - 测试仪表盘
 - 测试图表
 - 测试指标卡
 - 测试条形图
 - 测试折线图
 - 测试饼图
 - 测试双轴图
 - 测试柱量添加
 - 小菜测试
 - 测试仪表盘

已选内容

返回 下一步

注意：开启数据权限，需要底层的表中包含用户中心部门id，才可以生效。

4.选择要授权给哪些用户查看，选择好后点击确定按钮，授权成功。

创建授权

1 选择授权内容

2 选择授权对象

选择授权对象

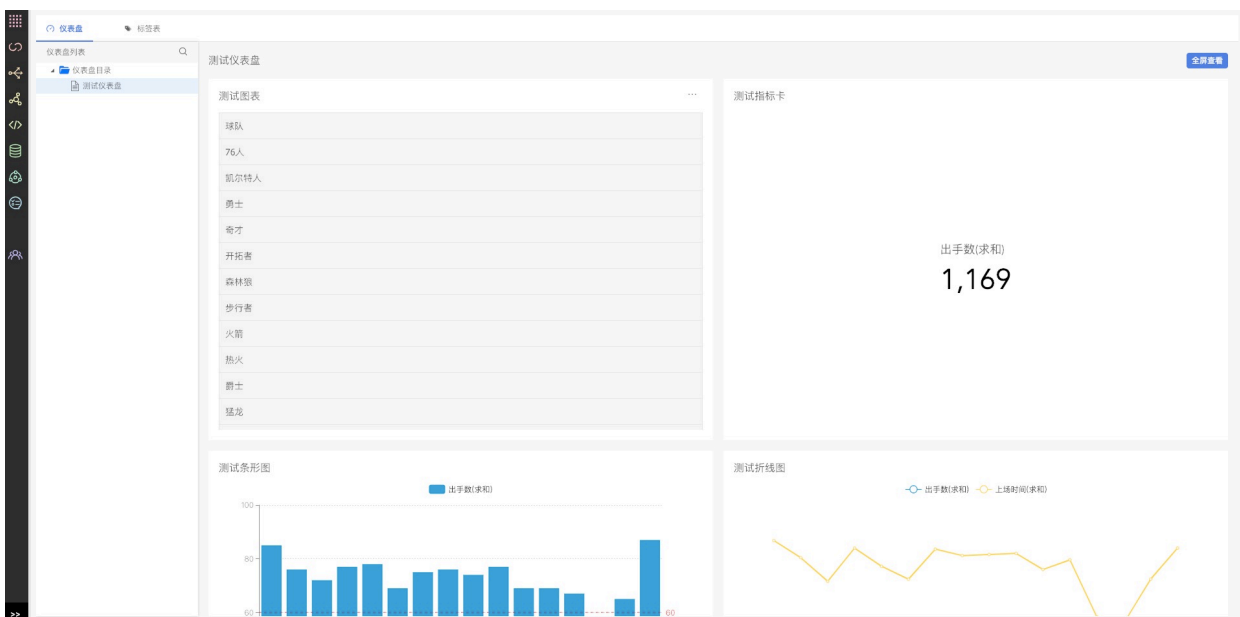
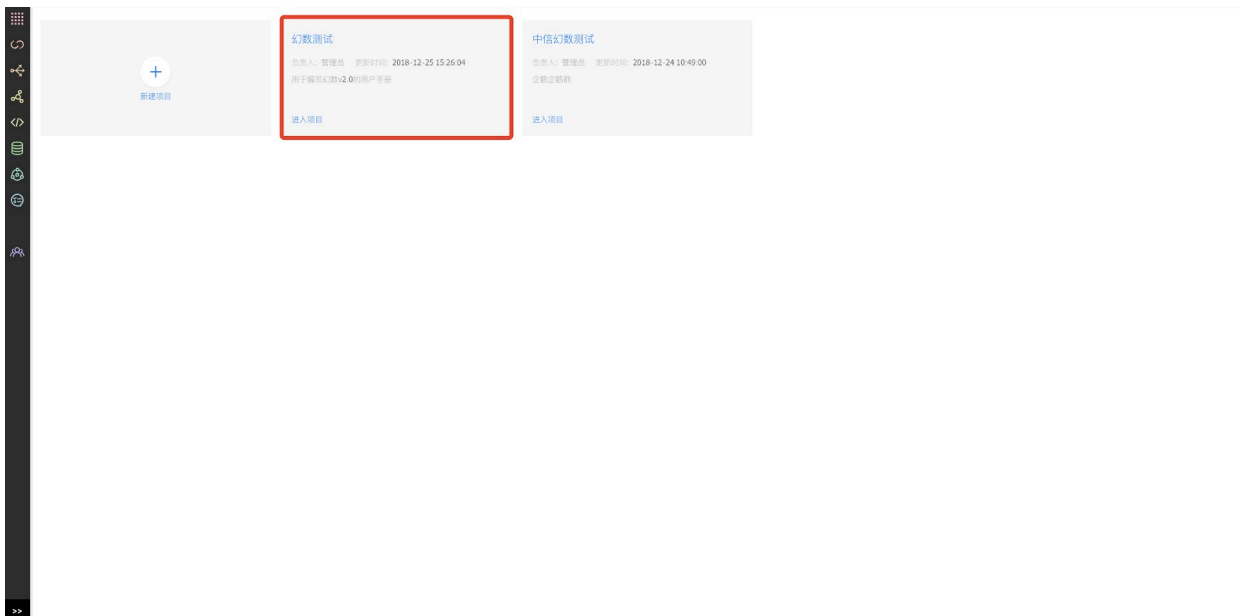
可选内容

- 数据
 - ODS
 - 中信
 - 管理员
 - 谢世局
 - 黄磊
 - 王保国
 - 宋亚伟
 - UED
 - 平台组

已选内容

上一步 确定

5.切换一个被授权的账号登录，可以看到图表授权成功。

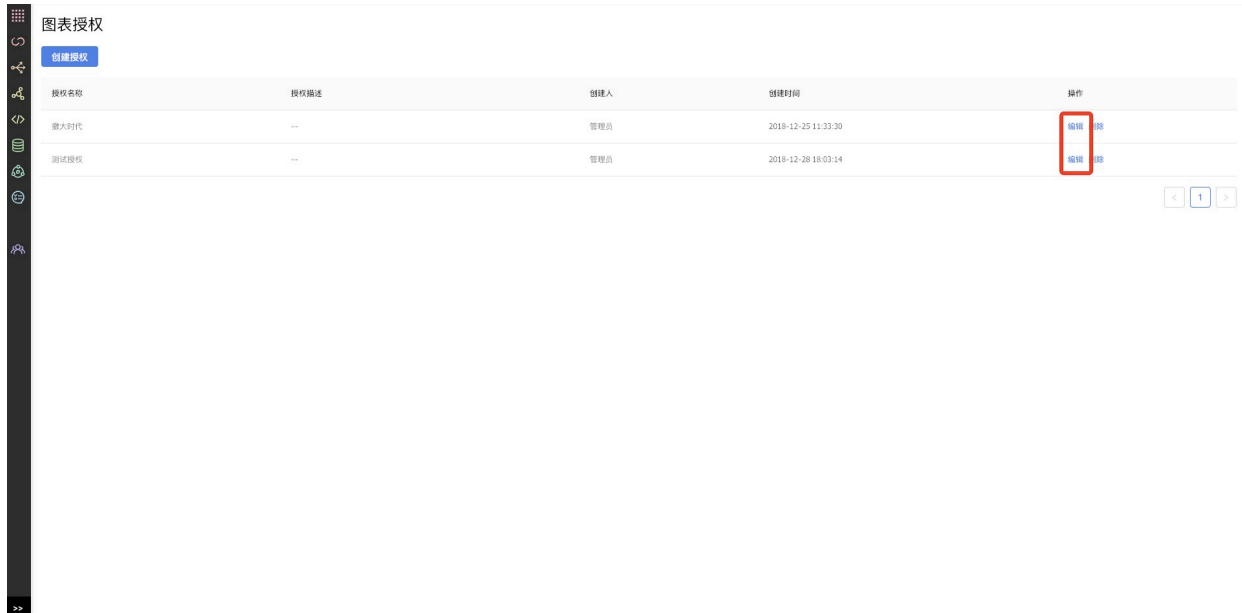


注意：

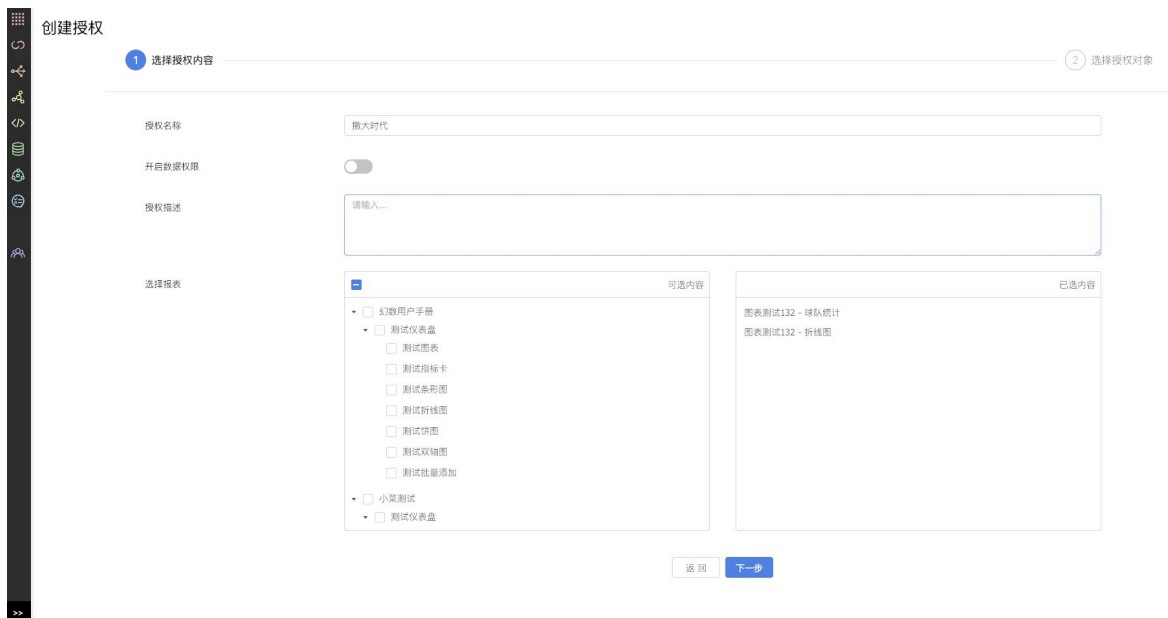
- 1) 授权的用户只能查看图表，不能对图表进行操作。
- 2) 开启数据权限生效的前提，是对应的标签表中有组织架构中的部门ID。这样给对应的部门的用户授权后，用户登录后只能看到自己所在部门的数据。

13.4.3 编辑图表授权

1.在图表授权页面，选择要编辑的授权，点击**编辑按钮**。

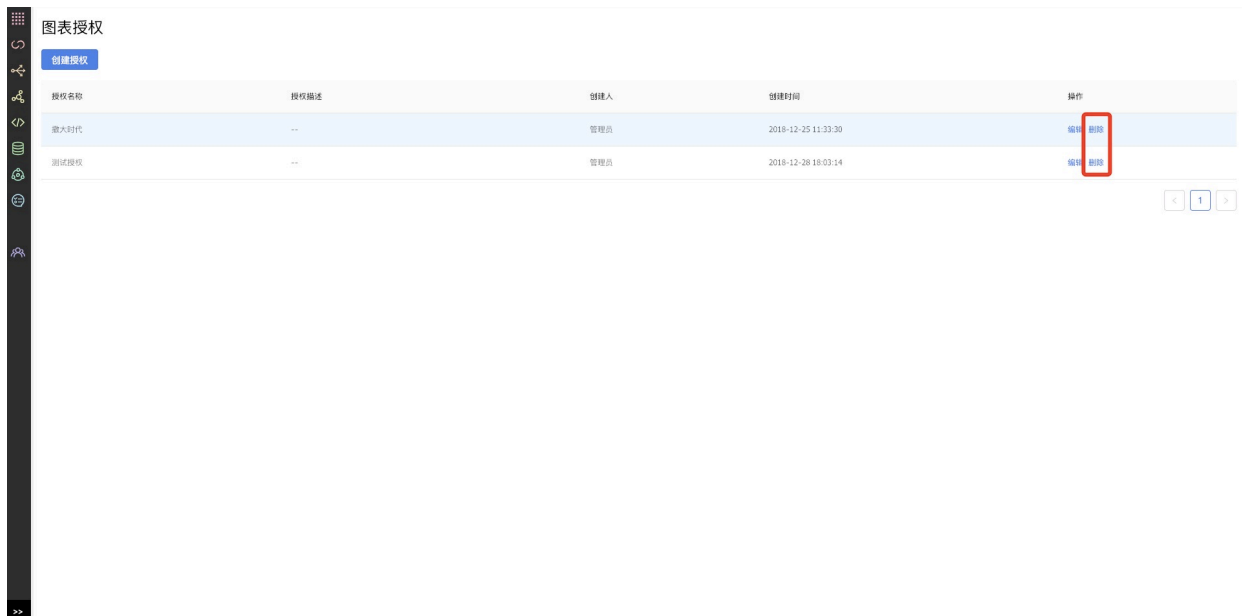


2. 编辑要修改的内容后，点击确定按钮，修改授权成功。

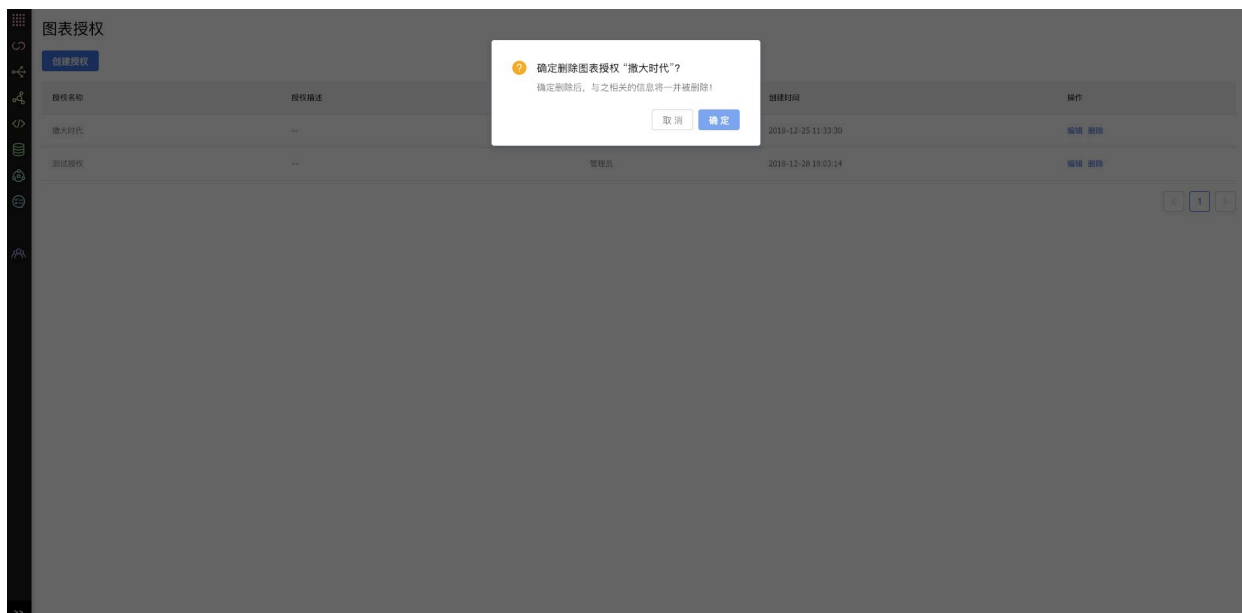


13.4.4 删除图表授权

1. 在图表授权页面，选择要删除的授权，点击删除按钮。



2. 删除图表授权需要二次确认，点击确定后，删除图表授权成功。



注意：

删除授权后，被授权的用户将无法再看到授权的图表。

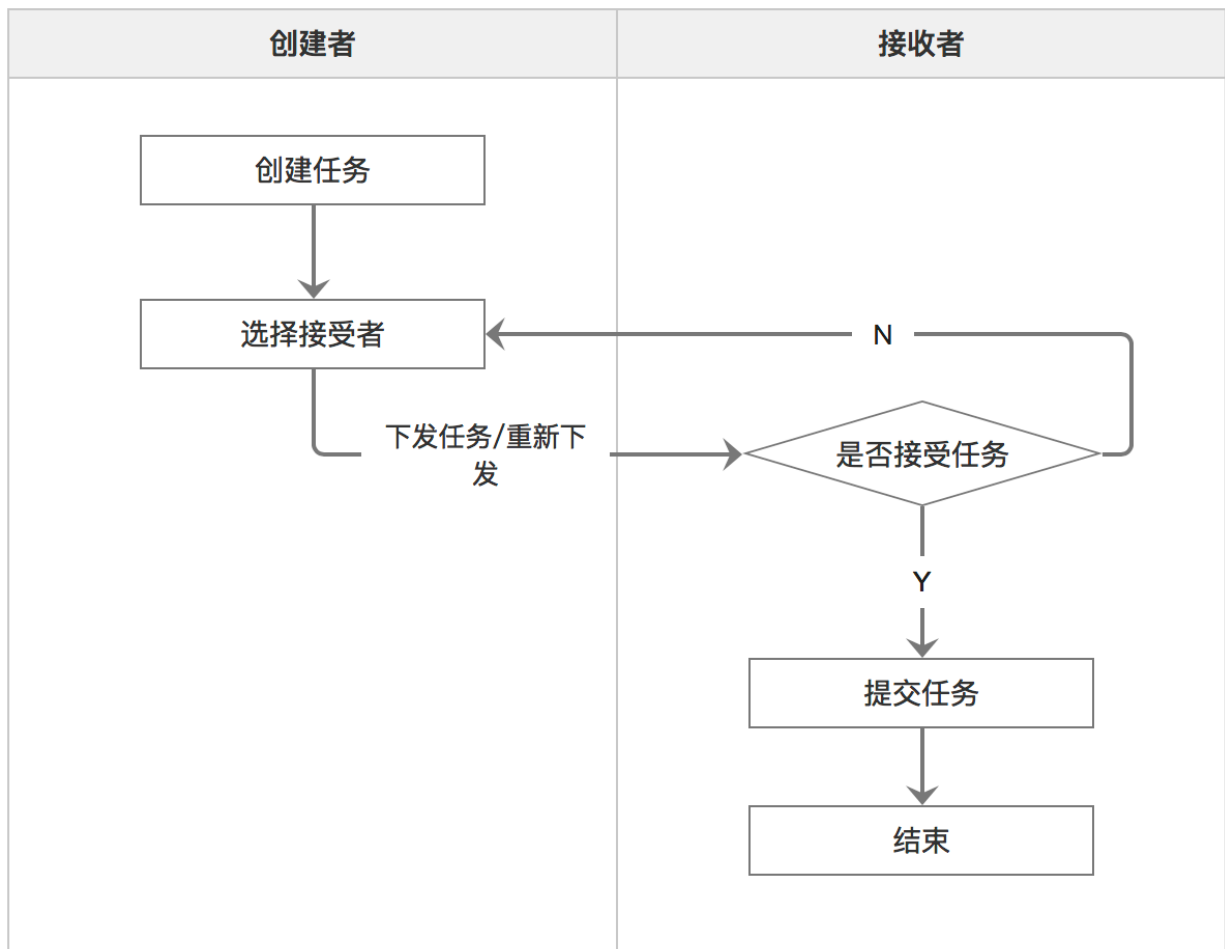
14 审计日志

点击左侧【审计日志】图标进入审计日志，审计日志中记录了用户的行为，可通过筛选项目进行筛选。

审计日志							
服务	全部	类型	全部	项目	全部	时间范围	2019-01-03 ~ 2019-01-03
用户	全部	结果	全部				
用户	操作	类型	服务	项目	时间	源IP	结果
管理员	获取用户项目列表	访问	开发平台		2019-01-03 11:51:43	127.0.0.1	成功
管理员	获取用户项目列表(包含工作空间信息)	访问	开发平台	hive_lch	2019-01-03 11:30:59	127.0.0.1	成功
管理员	获取用户项目列表(包含工作空间信息)	访问	开发平台	hive_lch	2019-01-03 11:20:56	127.0.0.1	成功
管理员	获取用户项目列表(包含工作空间信息)	访问	开发平台	hive_lch	2019-01-03 11:10:56	127.0.0.1	成功
管理员	当前项目空间下的表信息显示(开发空间)	访问	开发平台	hive_lch	2019-01-03 11:00:57	127.0.0.1	成功
管理员	查询租户的资源目录	访问	开发平台	hive_lch	2019-01-03 11:00:56	127.0.0.1	成功

15 任务管理

- 任务流程图



- 任务流程步骤

(创建者)创建任务 > (接收者) 接受任务 > (接收者) 完成任务

- 任务状态

待处理：创建者下发任务后，接收者还未接受或拒绝的状态

处理中：接收者接受任务后，还未提交任务的状态

待配置：创建者创建完任务后，未选择接收者或已选接收者但未下发任务的状态

已拒绝：接收者拒绝任务的状态

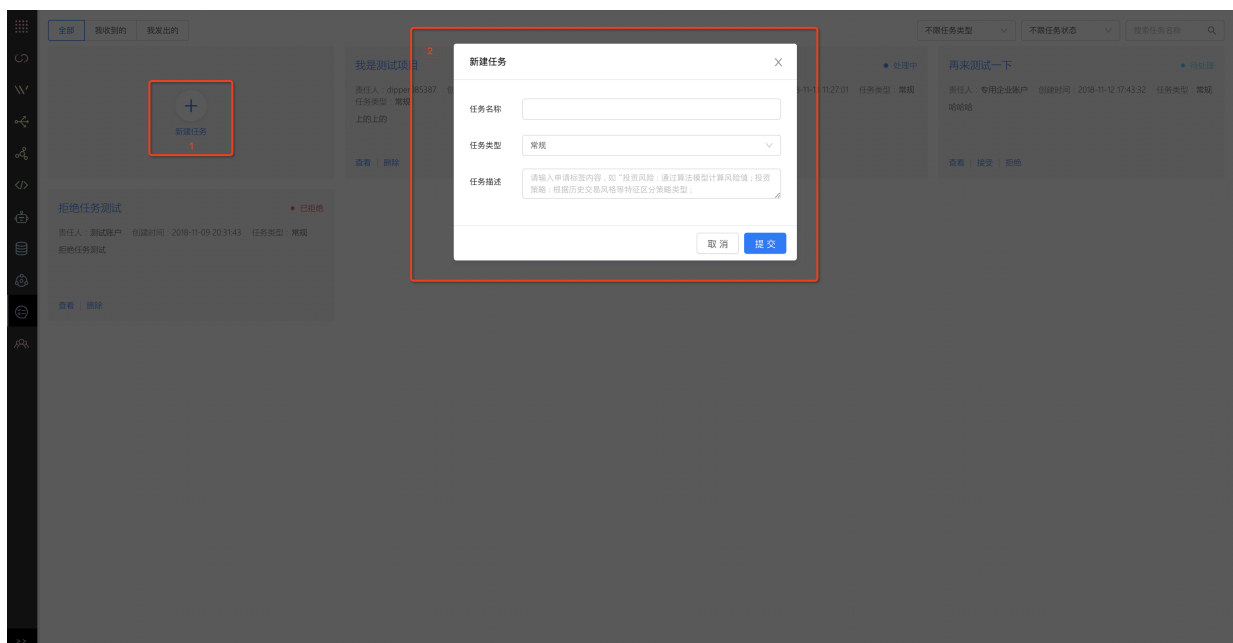
已完成：接收者提交任务之后的状态

具体任务流程如下所示，

15.1（创建者）创建任务

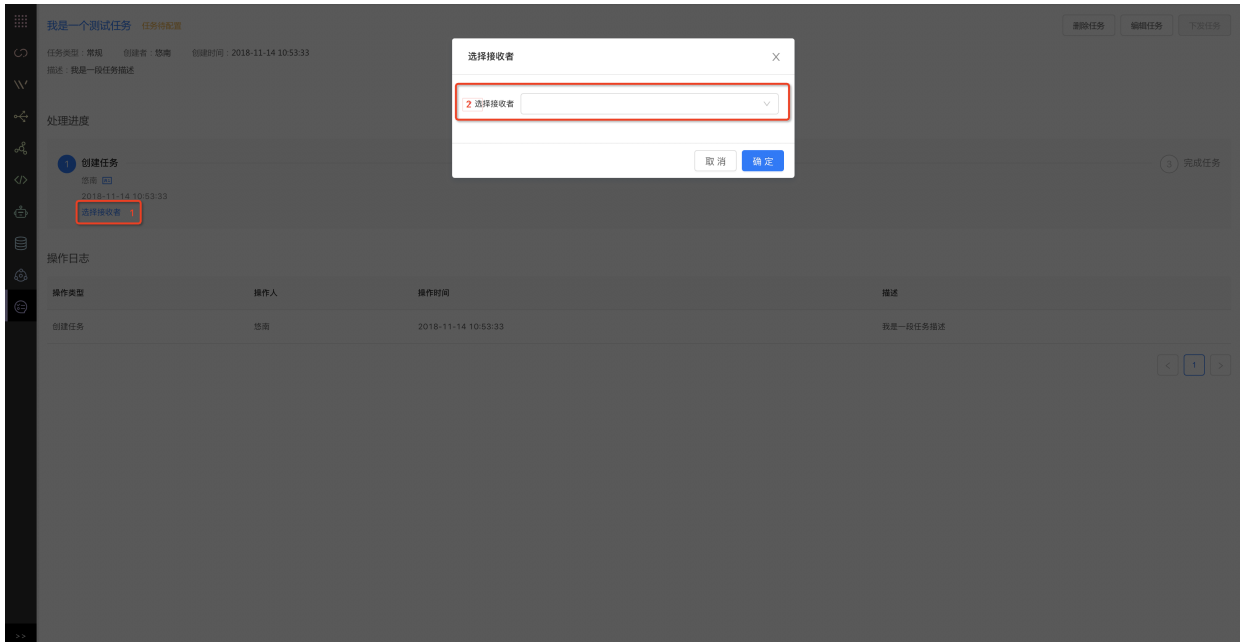
1、创建任务

创建者在任务列表中，点击“新建任务”，在弹出框中填入任务名，在任务类型下拉项中选择“常规”，将任务的目标、要求和内容填入到描述框中后，点击确定完成创建。



2、配置任务

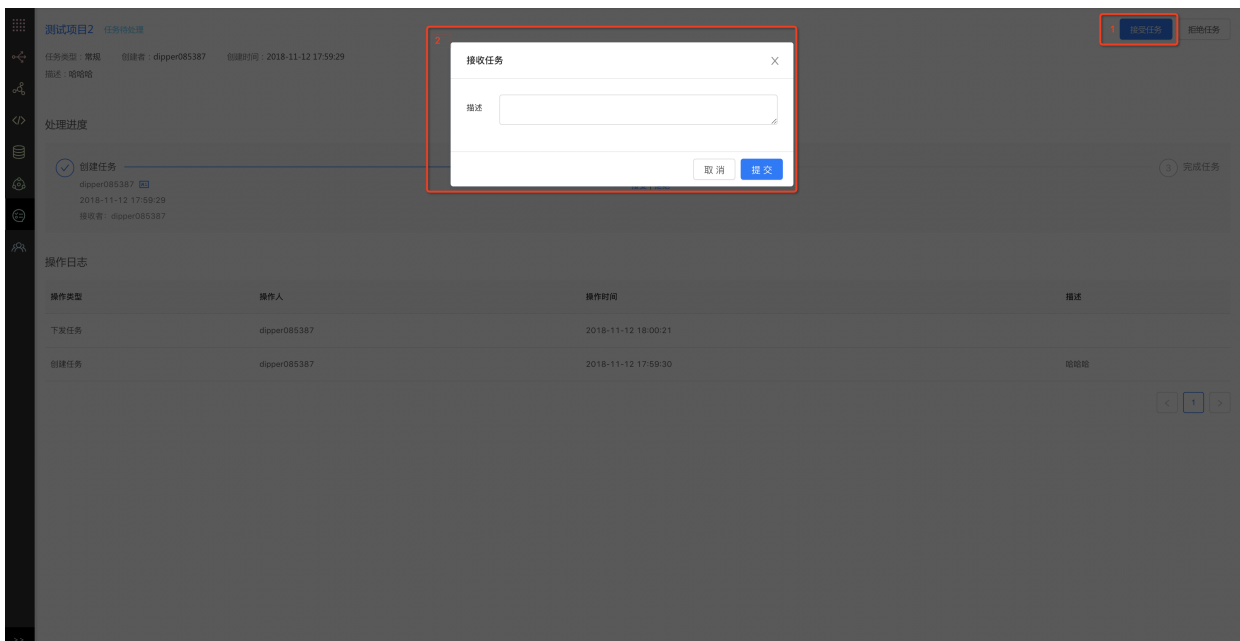
创建完任务之后，进入任务详情，点击选择接收者，在当前所在的组织架构中选择任务的接收者，确认后点击“下发任务”将任务发送给接收者。



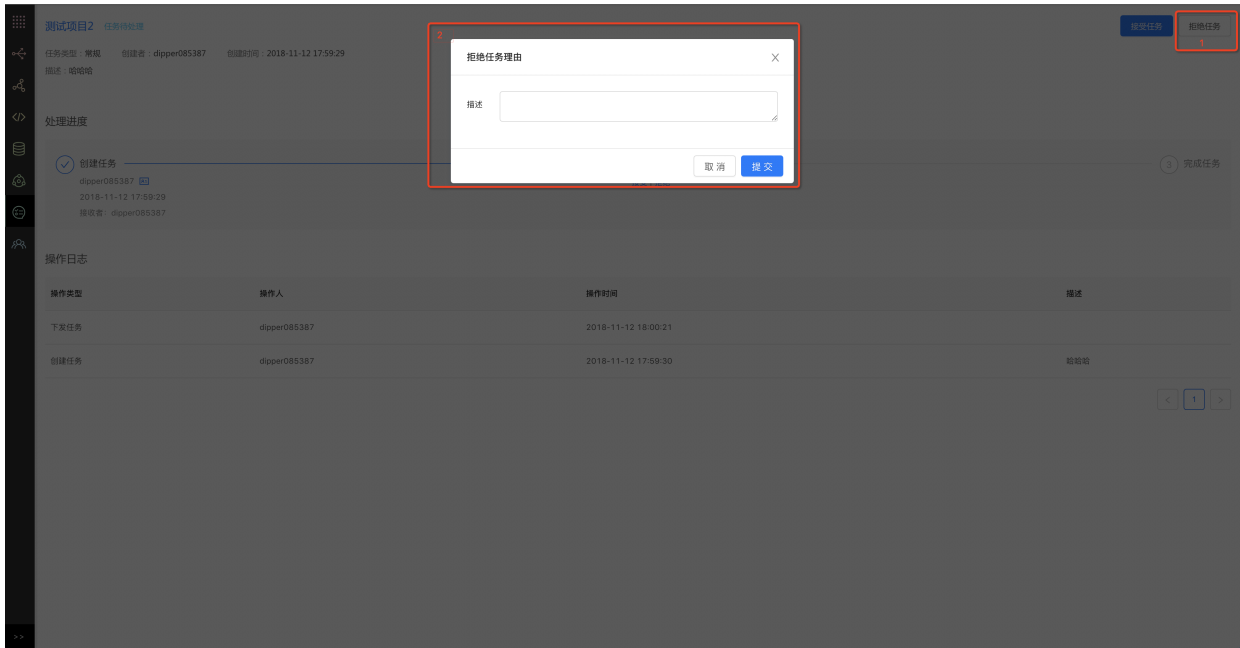
15.2 (接收者) 接受任务

接收者可以在任务列表“我收到的”分类中看到最新收到的任务，进入状态为“待处理”的任务中，在任务环节点上有两种类型的操作，接收者可以根据实际情况选择“接受”或“拒绝”任务。

- **接受任务** 接收者在任务环节点上点击“接受”，弹出确认框，确认接受任务。



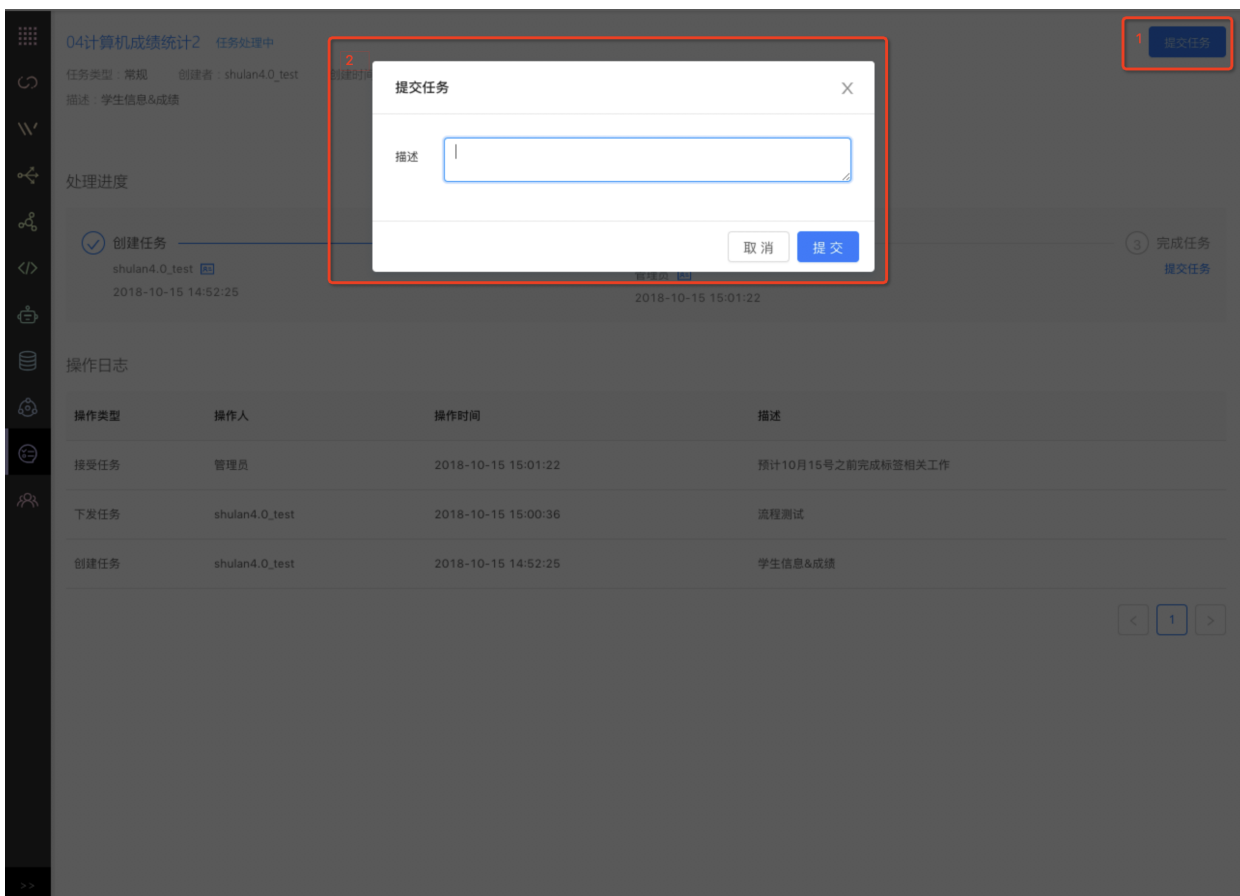
- **拒绝任务** 接收者在任务环节点上点击“拒绝”，在弹出的对话框中，填写拒绝原因，确定拒绝任务。



备注：(创建者)查看任务状态，若任务状态为拒绝,可重新下发任务，具体操作请见[重新下发任务](#)

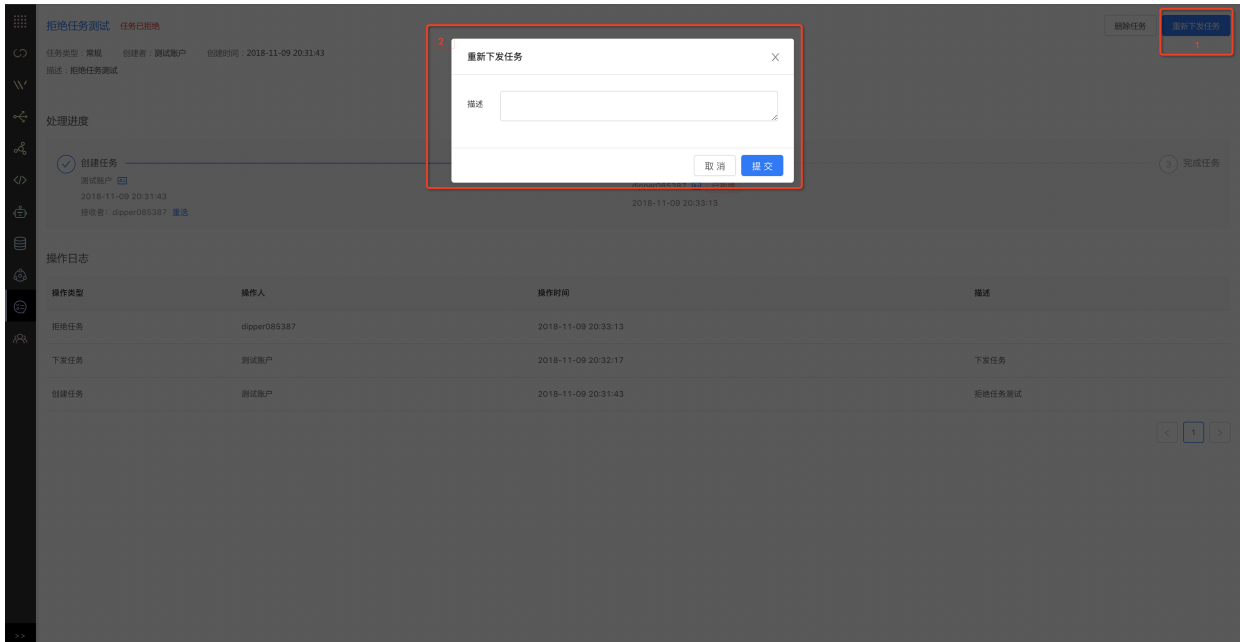
15.3 (接收者) 提交任务

接收者接受任务并在线上或线下完成相应的任务开发后，进入“任务详情页面”，点击“提交任务”，创建者可实时收到该任务的状态变化，至此整个任务流程结束。



15.4 (创建者) 重新下发任务

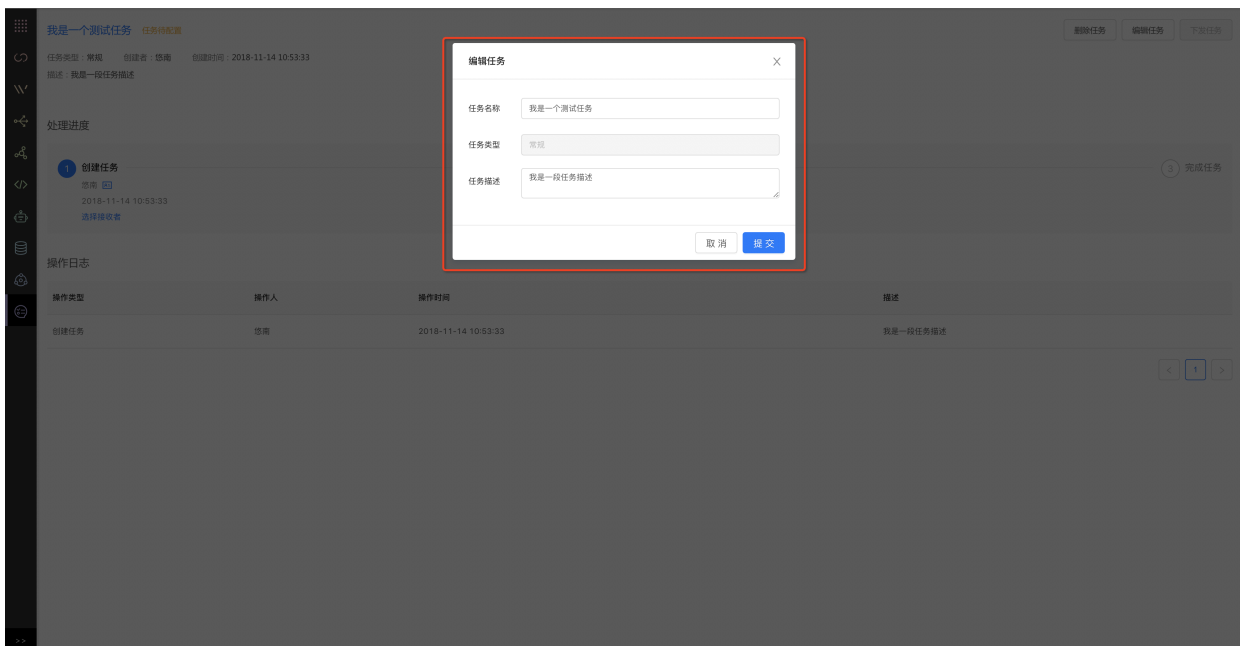
创建者在收到接收者拒绝任务的请求时，可根据拒绝原因的来决定是否重新下发给当前拒绝者或组织架构中的其他成员。重选接收者后，点击“重新下发”将任务下发给此接受者。



15.5 (创建者) 编辑、删除任务

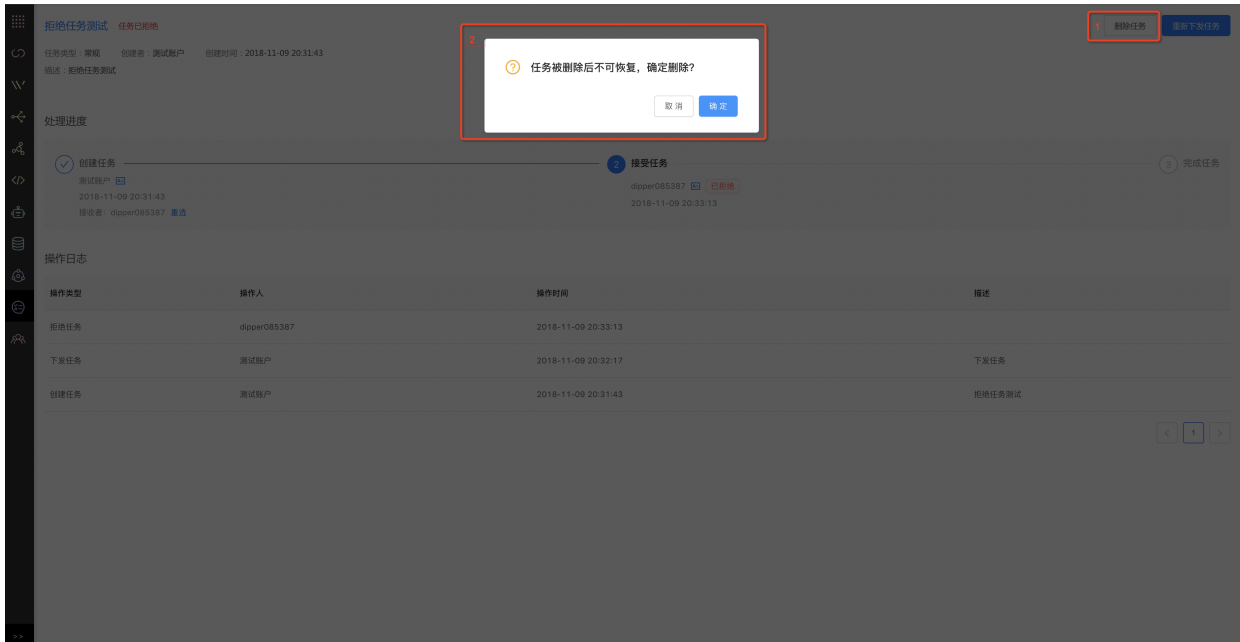
- 编辑任务

只有创建者有编辑任务权限，可以编辑任务名称和描述，任务类型不可修改



- 删除任务

创建者和接受者都有任务权限，但只能删除这条任务在自己账户上的记录，不影响对方的任务记录，不会从后台数据库中真实删除。



15.6 (创建者) 查看操作日志

创建者和接收者对任务每一次的操作、任务信息的修改及时间点，系统均会自动记录形成详细的操作日志，做到有记录可查。在任务详情页里“处理进度”模块下方，可以看到“操作日志”的列表，包含操作类型、操作人、操作时间及描述信息。

- 7种操作类型

创建任务

下发任务

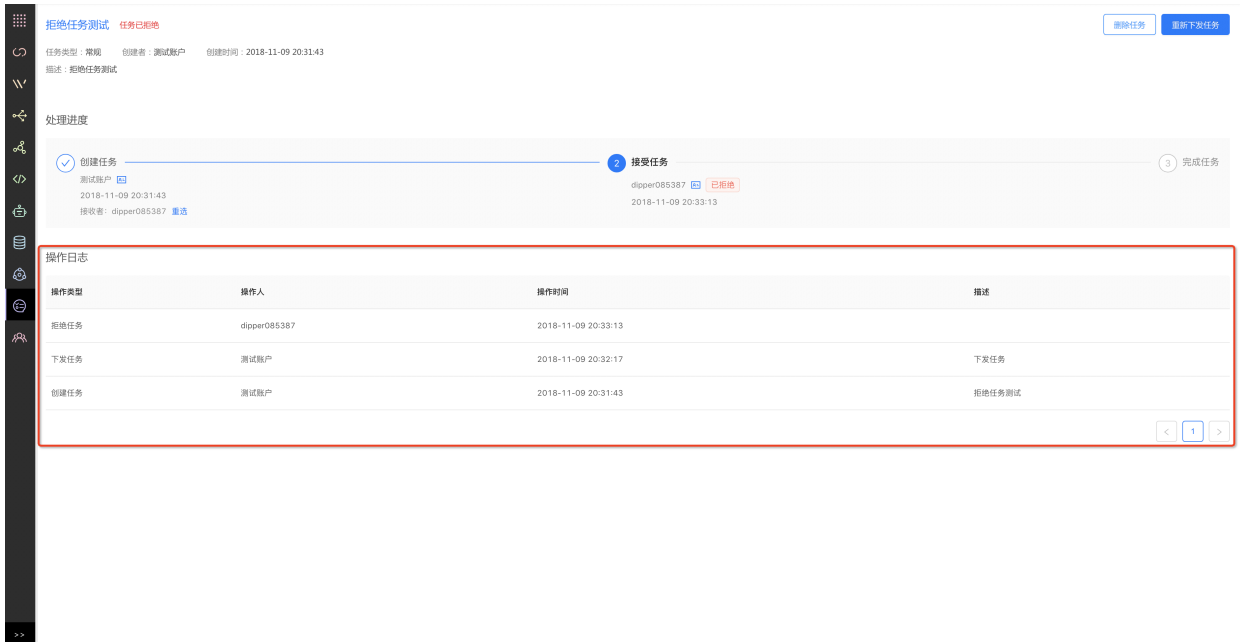
接受任务

完成任务

拒绝任务

重新下发任务

编辑任务



案例实战

1 概述

1.1 背景介绍

本案例介绍如何使用平台每天定时对业主信息和业主投诉内容等原始数据进行加工处理，以此实现对业主投诉信息数据的统计分析，通过数据加工形成标签，配置分析引擎，并通过幻数展示可视化报表。

1.2 操作流程

在本案例中，我们按照以下步骤在平台中构建数据，以便对数据进行处理、加工、挖掘以及可视化。

在案例实战前，请参考 [快速入门](#) 一节内容完成账号注册等步骤。

步骤1：数据开发

步骤2：发布管理

步骤3：运维监控（可跳过）

步骤4：数据管理（可跳过）

步骤5：创建标签

步骤6：创建服务

步骤7：数据分析

2 开发平台

2.1 项目列表

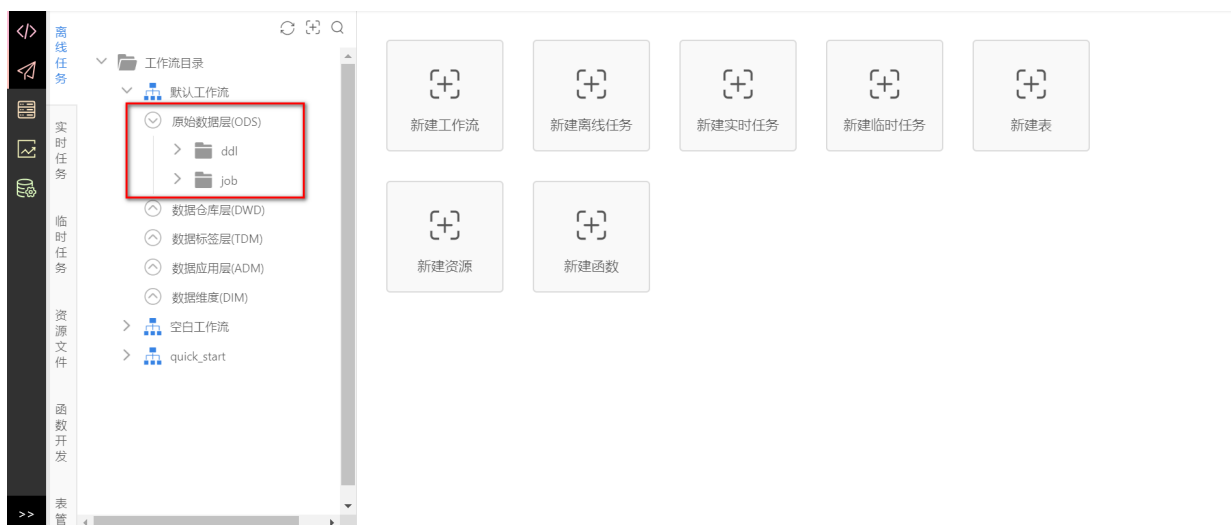
在控制台中点击【开发平台】进入开发平台项目列表，本平台已为您部署好一个Hive类型的项目 quick_demo，

若没有，请先创建一个项目，并做好配置，详情请参考【[用户指南-项目管理](#)】部分。



2.2 进入开发

点击进入开发快捷入口，进入开发中心页面。已经默认创建一个默认工作流和一个空白工作流，默认工作流已自动创建好ods、dwd、tdm、adm、dim层目录。

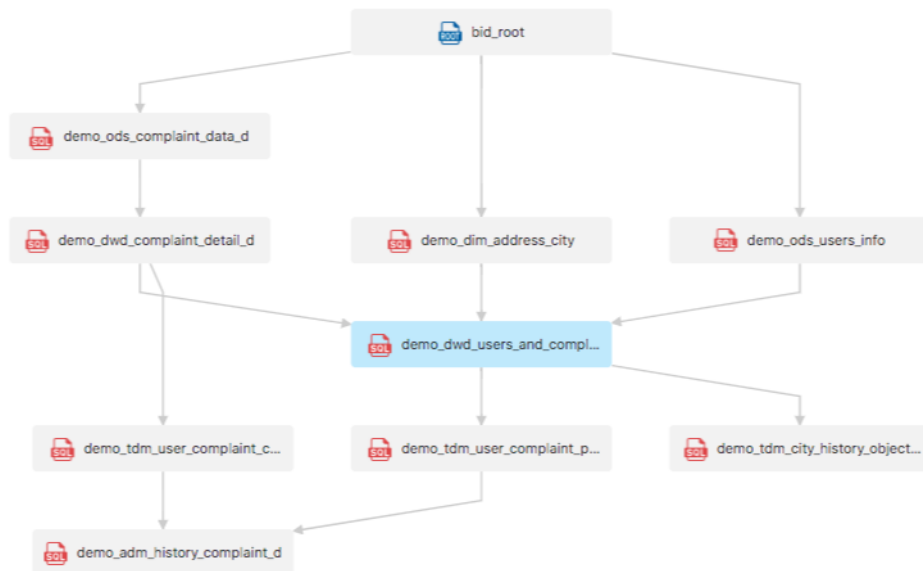


平台推荐使用数据分层的概念进行数据开发，数据层级分为ods、dwd、tdm、adm、dim层。各层任务如下：

- ods层存放原始数据，此案例ods层任务如下：
 - demo_ods_complaint_data_d
 - demo_ods_users_info
- dwd层存放清洗、初加工ods层数据后的数据，此案例dwd层任务如下：
 - demo_dwd_complaint_detail_d
 - demo_dwd_users_and_complaint_detail_d
- tdm层为数据标签层，通常在此层，标签化dwd层数据，此案例tdm层任务如下：
 - demo_tdm_city_history_object_count_d
 - demo_tdm_user_complaint_count_d

- demo_tdm_user_complaint_position_count_d
- adm层为数据应用层，按照特定的业务组织标签数据，此案例adm层任务如下：
 - demo_adm_history_complaint_d
- dim层存放维表数据，此案例dim层任务如下：
 - demo_dim_address_city

案例实战任务依赖关系如图所示。



2.3 数据维度(DIM)

数据维度(DIM)存放维表数据，此层任务可以是定时任务，定时更新维表数据。本案例中dim层存放定时更新的小区 and 城市的映射关系数据，您需要先将维表数据上传到平台，然后将数据导入表中。具体步骤如下：

(1) 将把维表数据存储为txt格式的 `address_city.txt` 文件，且文件后缀名必须为txt。地址数据如下(字段之间以Tab键分割)。

1	南京街g座	北京
2	关街N座	杭州
3	百路T座	南昌
4	六盘水路I座	南昌
5	辛集路Q座	北京
6	百路T座	上海
7	潮州路T座	西安
8	汪街H座	厦门
9	南京路y座	上海
10	于街k座	南京
11	于街k座	南昌
12	虞街d座	上海

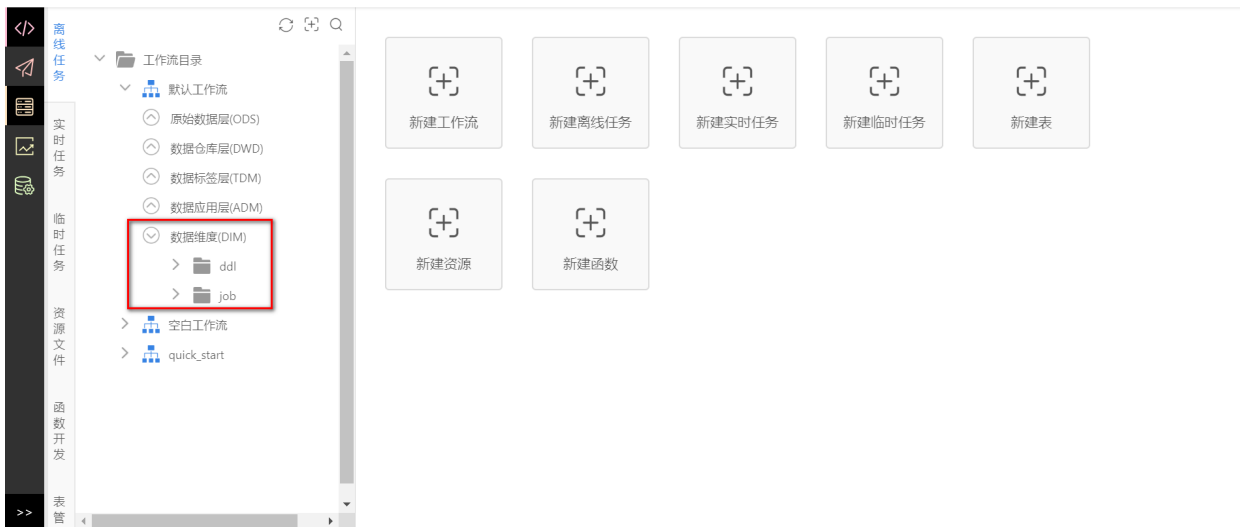
13 于街k座 南昌
 14 濮街R座 上海
 15 梧州路a座 上海
 16 闻街D座 南昌
 17 深圳街M座 杭州
 18 兴城街Y座 北京
 19 辛集路Q座 厦门
 20 乐路Y座 厦门
 21 通辽路r座 厦门
 22 政街P座 北京
 23 台北路h座 北京
 24 周街D座 上海
 25 香港路s座 厦门
 26 南京路z座 深圳
 27 呼和浩特街G座 厦门
 28 呼和浩特街G座 厦门
 29 嘉禾街w座 南昌
 30 呼和浩特街G座 南昌
 31 终街L座 西安
 32 辛集路Q座 厦门
 33 通辽路r座 厦门
 34 周街D座 上海
 35 羊街y座 杭州
 36 西路H座 厦门
 37 兴城街Y座 西安
 38 终街L座 北京
 39 香港路s座 杭州
 40 香港路s座 北京

字段名	类型	描述
addressid	Int	业主小区id
address	string	业主小区
city	string	城市

(2) 点击左侧的资源文件页面，创建demo目录，在demo目录下创建dim目录。在dim目录下新建资源，资源名 `address_city`，资源类型txt，添加需要上传的 `address_city.txt` 文件，完成上传。



(3) 在开发中心的离线任务的数据维度 (DIM) 中, 创建demo workflow(本教程使用默认 workflow 演示), 模板选择默认模板, 在数据纬度(DIM)目录下新建ddl/job两个目录。



(4) 在 数据维度(DIM) -> ddl 目录下创建DDL任务, 任务名为 `ddl_demo_dim_address_city`, 输入下面的建表语句, 运行 任务完成hive表创建工作。

提示: DDL任务调度类型默认为暂停调度, 当任务发布以后, DDL任务立即运行且仅运行一次。

```

-- *****
-- 功能: 创建demo_dim_address_city表
-- 作者: 明罡
-- 时间: 2018-02-02
-- *****

drop table if exists demo_dim_address_city;

create table if not exists demo_dim_address_city
(
    address_id bigint comment '业主小区id'
    ,address string comment '业主小区'
    , city    string comment '城市'
)

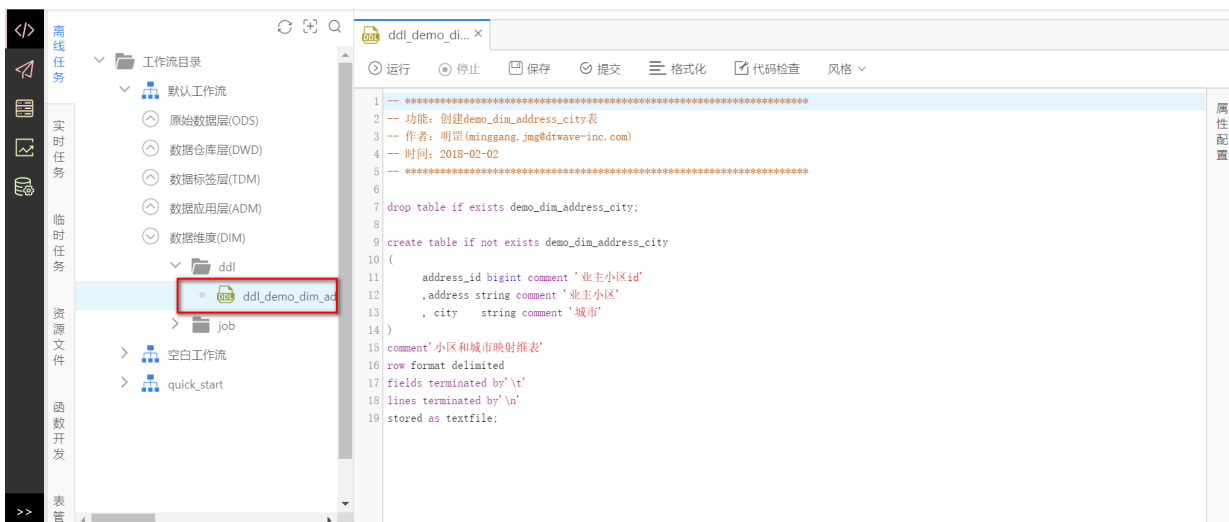
```

```

)
comment '小区和城市映射维表'
row format delimited
fields terminated by '\t'
lines terminated by '\n'
stored as textfile;

```

注意: 用户在数据开发中, 不用指定数据库名。该任务在开发环境中执行的时候, 表对应的库就是开发环境下的Hive库。发布到生产, 在生产环境中执行的时候, 表对应的库就是生产环境的Hive库。开发环境和生产环境对应的Hive库可以从管理门户中的项目配置里查看。



(5) 在 **数据纬度(DIM)** -> **job**目录下创建SparkSQL任务, 任务名为 `demo_dim_address_city`, 在界面右侧**属性配置**中资源依赖里选择上传的资源 `address_city.txt`, 上游任务: `bid_root`, 在新建任务页面输入任务语句, 运行任务完成数据导入。

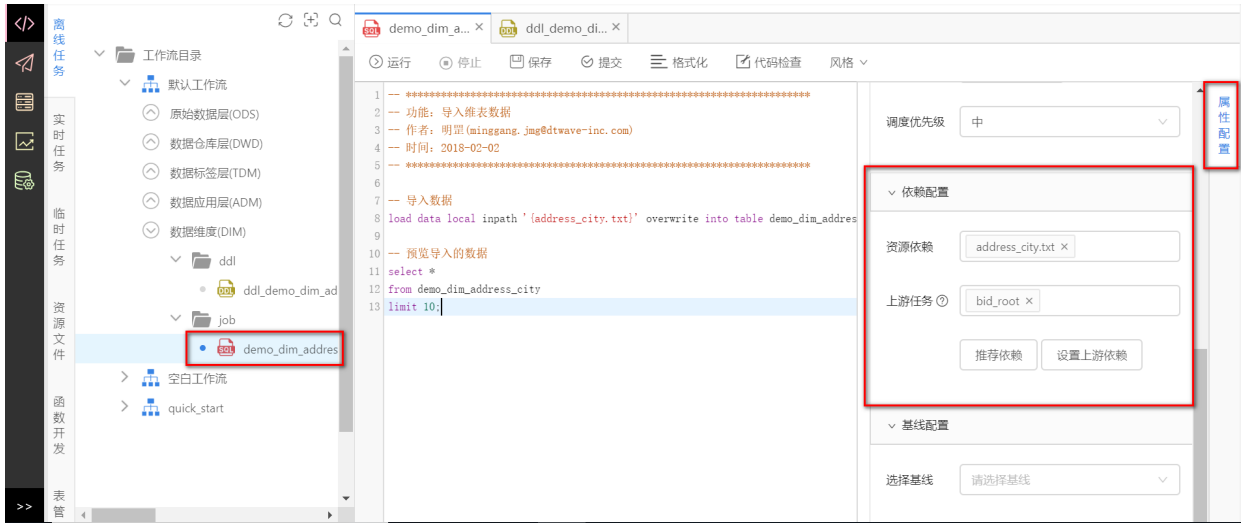
```

-- *****
-- 功能: 导入维表数据
-- 作者: 明昱
-- 时间: 2018-02-02
-- *****

-- 导入数据
load data local inpath '{address_city.txt}' overwrite into table
demo_dim_address_city;

-- 预览导入的数据
select *
from demo_dim_address_city
limit 10;

```



2.4 原始数据层(ODS)

ods层存放原始数据，此层数据保持原貌，不做任何改动。本案例中ods层的数据来源于已经准备好的文件 `demo_complaint_data.txt` 和 `demo_users.txt`，您需要将文件中的数据导入到该项目的 `demo_ods_complaint_data` 表和 `demo_ods_users_info` 表中。

部分样本数据如下：

`demo_complaint_data` 表

```

2046  厨房窗户关不上  2018-11-27
3648  你好,卧室门有污渍,清理不掉  2018-11-28
2480  浴室门上都是划痕  2018-11-20
1408  投诉好久了我家的主卧地板发黑,现在都没人管!  2018-11-26
2154  卫生间水管渗水,房子都不能住了  2018-11-16
2810  客服你好,厨房台面有点开裂  2018-11-20
2404  你好,我家厨房的地漏堵着了  2018-11-25
3128  卫生间马桶下水慢,请快速解决  2018-11-26
3081  客服你好,厨房台面有点开裂  2018-11-21
658  你好,卧室门有污渍,清理不掉  2018-11-25

```

字段名	类型	描述
userid	string	客户id
content	string	投诉内容
complaint_time	string	投诉时间

`demo_users` 表

1	山强	27	8
2	侯海燕	43	34
3	于芳	33	4
4	于志强	46	5
5	父波	27	35
6	苟强	39	31
7	皇秀兰	29	25
8	甄秀华	47	14
9	魏琳	48	2
10	羊霞	43	4

字段名	类型	描述
userid	string	业主id
username	string	业主名字
age	Int	年龄
address_id	Int	小区id

(1) 在开发中心的离线任务的原始数据层(ODS)目录下新建ddl与job目录。



(2) 在 原始数据层(ODS) -> ddl 目录下新建离线DDL任务，任务名

称：`ddl_demo_ods_complaint_data_d`，在新建任务页面输入建表语句，点击运行，完成hive表创建工作。任务语句如下：

```

-- *****
-- 功能：创建demo_ods_complaint_data_d表2
-- 作者：明昱
-- 时间：2018-02-02
-- *****

drop table if exists demo_ods_complaint_data_d;

```

```

create table if not exists demo_ods_complaint_data_d
(
    user_id    bigint comment '业主ID'
    , content   string comment '投诉内容'
    , complaint_time string comment '投诉时间'
)
comment '业主投诉内容表'
partitioned by (ds string comment '按天分区,例如20160620')
row format delimited
fields terminated by'\t'
lines terminated by'\n'
stored as textfile;

```

同样地，在 **原始数据层(ODS)** -> **ddl** 目录下创建DDL任务，任务名

称：`ddl_demo_ods_users_info`。按照上述操作完成 `demo_ods_users_info` 表创建。任务语句如下：

```

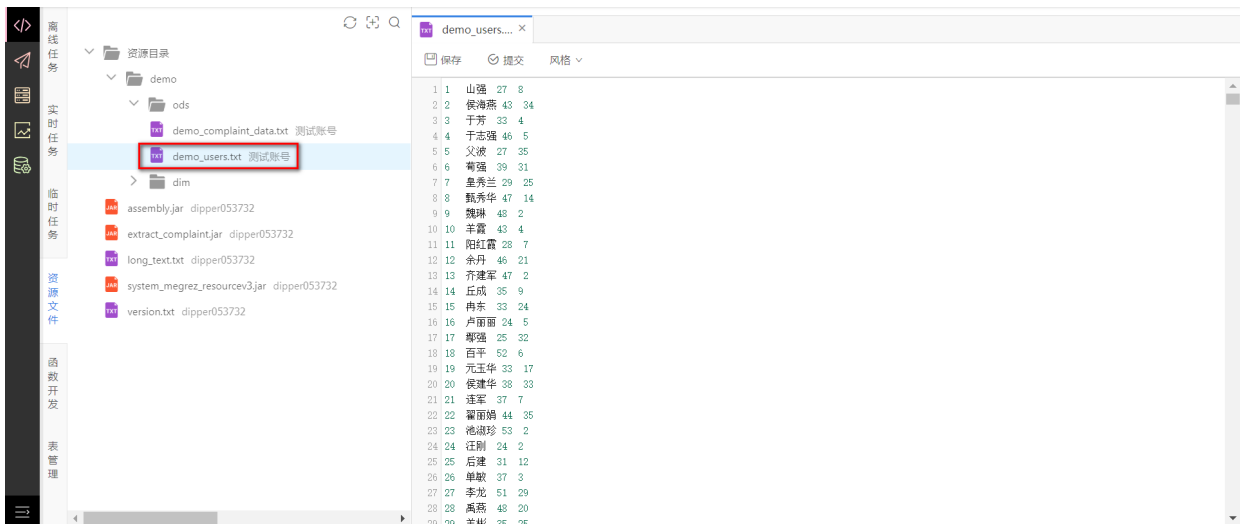
-- *****
-- 功能：创建demo_ods_users_info表
-- 作者：明罡
-- 时间：2018-02-02
-- *****

drop table if exists demo_ods_users_info;

create table if not exists demo_ods_users_info
(
    user_id bigint comment '业主ID'
    , name   string comment '业主名字'
    , age    bigint comment '业主年龄'
    , address_id bigint comment '业主小区id'
)
comment '业主信息表'
row format delimited
fields terminated by'\t'
lines terminated by'\n'
stored as textfile;

```

(3) 新建资源文件，上传source文件夹中的 `demo_users.txt`。



在原始数据层(ODS) -> job 目录下新建离线SparkSQL任务，任务名称：`demo_ods_users_info`，在界面右侧属性配置中资源依赖里选择上传的资源 `demo_users.txt`，上游任务：`bid_root`。

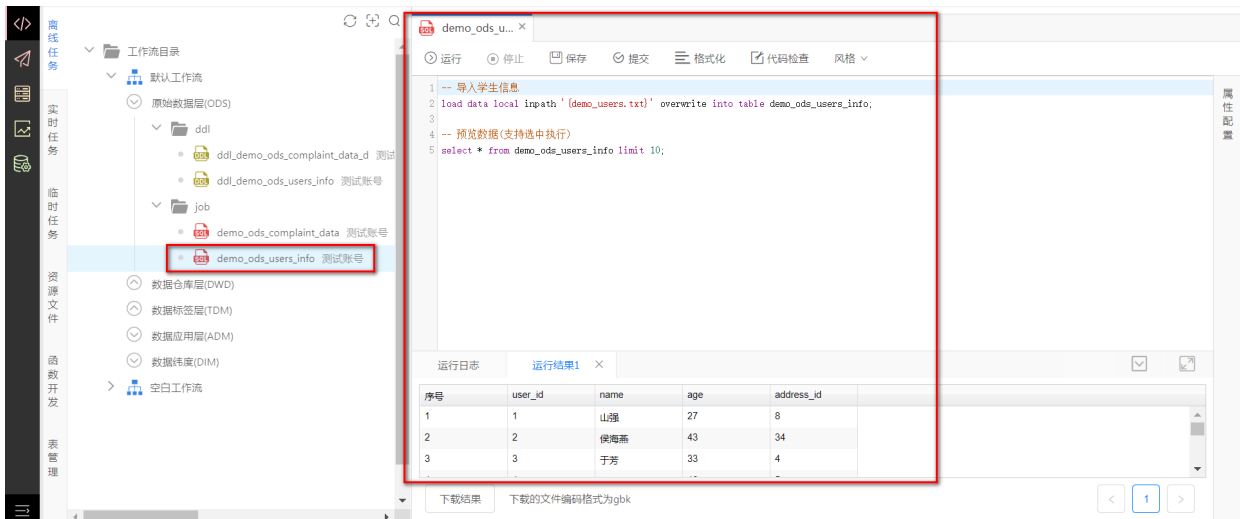
在新建任务页面输入代码，点击运行，向表中导入数据。任务语句如下：

-- 导入学生信息

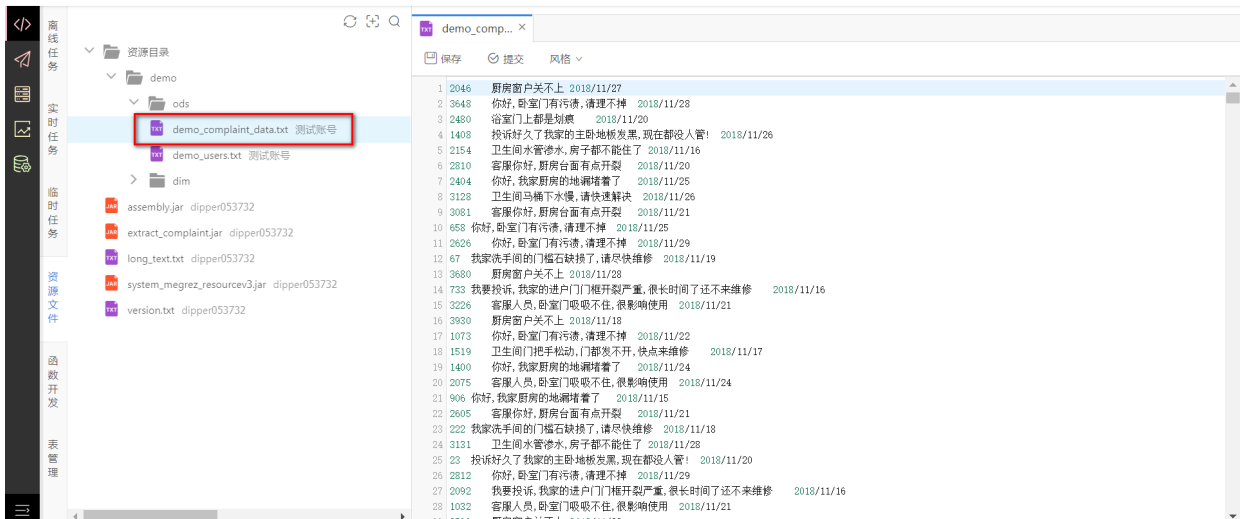
```
load data local inpath '{demo_users.txt}' overwrite into table
demo_ods_users_info;
```

-- 预览数据(支持选中执行)

```
select * from demo_ods_users_info limit 10;
```



(4) 新建资源文件，上传source文件夹中的demo_complaint_data。



在原始数据层(ODS) -> job 目录下新建离线SparkSQL任务，任务名称：`demo_ods_complaint_data_d`，在界面右侧属性配置中运行参数里填写参数名为bizDate,参数值填今天时间，如：`bizDate=20181111`，资源依赖：`demo_complaint_data.txt`，上游任务：`bid_root`。

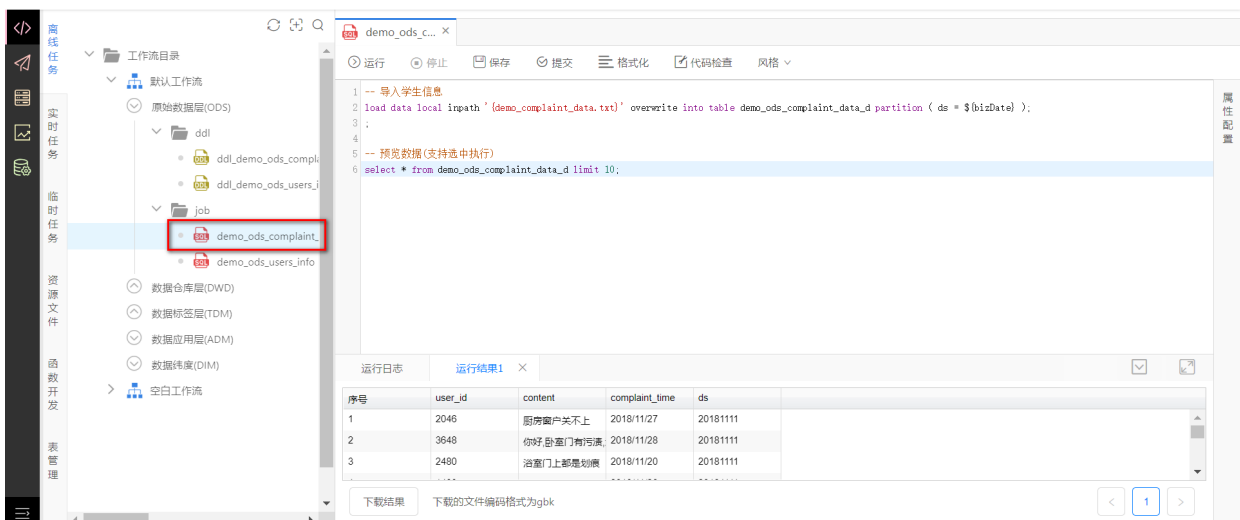
在新建任务页面输入代码，点击运行，向表中导入数据。任务语句如下：

```

-- 导入学生信息
load data local inpath '{demo_complaint_data.txt}' overwrite into table
demo_ods_complaint_data_d partition ( ds = ${bizDate} );
;

-- 预览数据(支持选中执行)
select * from demo_ods_complaint_data_d limit 10;

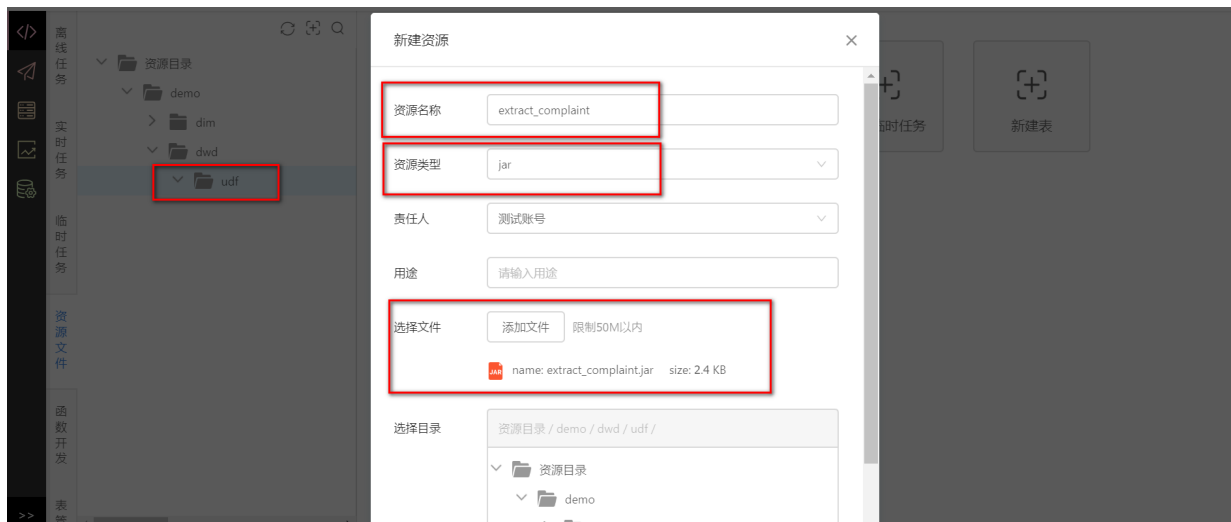
```



2.5 数据仓库层(DWD)

原始数据层(ODS)的数据经过清洗，初加工以后存放在数据仓库层(DWD)。本案例需要使用算法模型从业主投诉信息中获取投诉区域、投诉对象、投诉问题，然后合并dim层的数据，获得以业主对象的所有信息，具体步骤如下：

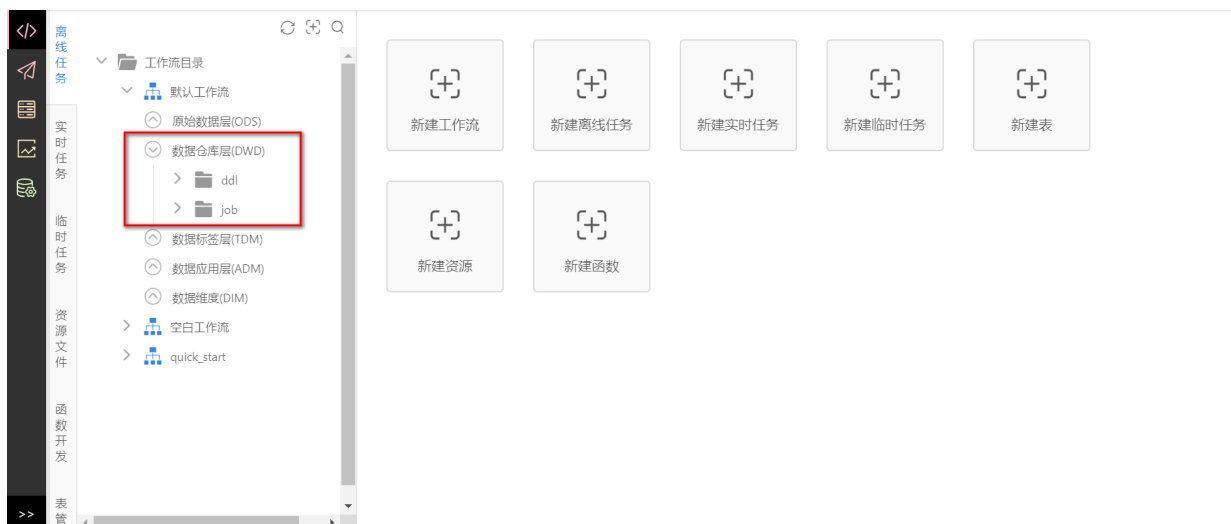
(1) 将libs文件夹中已编译好的udf的jar包上传到开发中心的资源开发页面下的demo/dwd/udf目录下面，目录如果不存在则需创建，资源名：`extract_complaint`，资源类型：`jar`。



(2) 在开发中心的函数开发页面下的自定义函数中，创建demo/udf目录。点击udf目录，选择新建函数，函数名：`extract_complaint`，类名(类名全路径)：`com.dtwave.hive.ExtractComplaintUDF`，资源依赖：`extract_complaint.jar`。用途、命令格式、参数说明这三个参数选填。



(3) 在开发中心的离线任务的数据仓库层(DWD)目录下新建ddl与job两个目录。

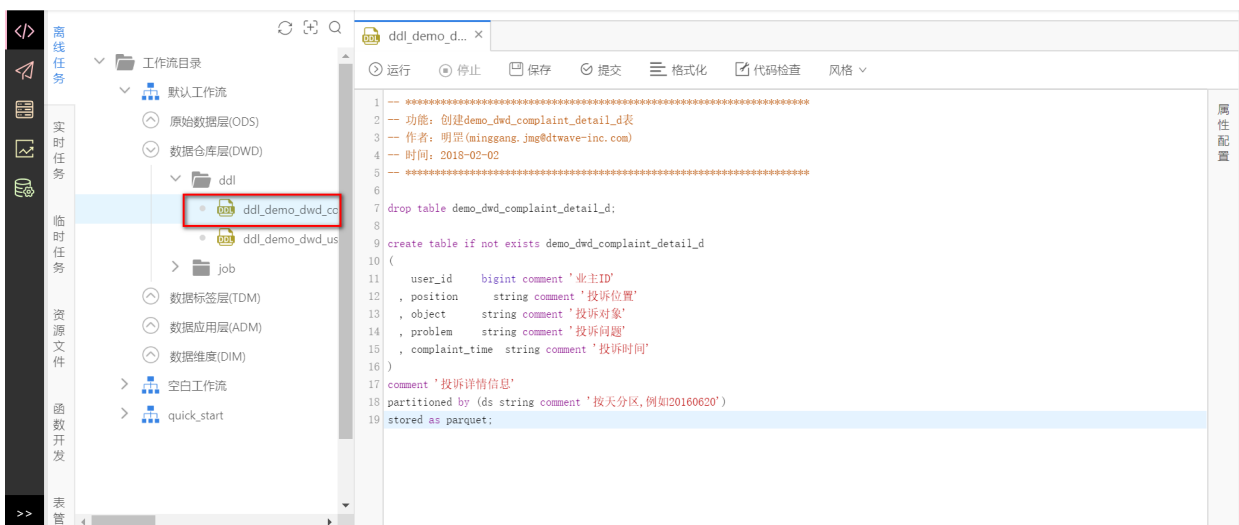


(4) 在 **数据仓库层(DWD)** -> **ddl** 目录下新建**离线DDL任务**，任务名为 `ddl_demo_dwd_complaint_detail_d`，在任务页面输入建表语句，运行完成hive表创建工作。任务语句如下：

```
-- *****
-- 功能：创建demo_dwd_complaint_detail_d表
-- 作者：明昱
-- 时间：2018-02-02
-- *****

drop table demo_dwd_complaint_detail_d;

create table if not exists demo_dwd_complaint_detail_d
(
    user_id      bigint comment '业主ID'
  , position    string comment '投诉位置'
  , object      string comment '投诉对象'
  , problem     string comment '投诉问题'
  , complaint_time string comment '投诉时间'
)
comment '投诉详情信息'
partitioned by (ds string comment '按天分区,例如20160620')
stored as parquet;
```



(5) 同样地，在 **数据仓库层(DWD)** -> **ddl** 目录下创建 `ddl_demo_dwd_users_and_complaint_detail_d` 任务，任务类型为DDL，按照上述操作创建 `demo_dwd_users_and_complaint_detail_d` 表。任务语句如下：

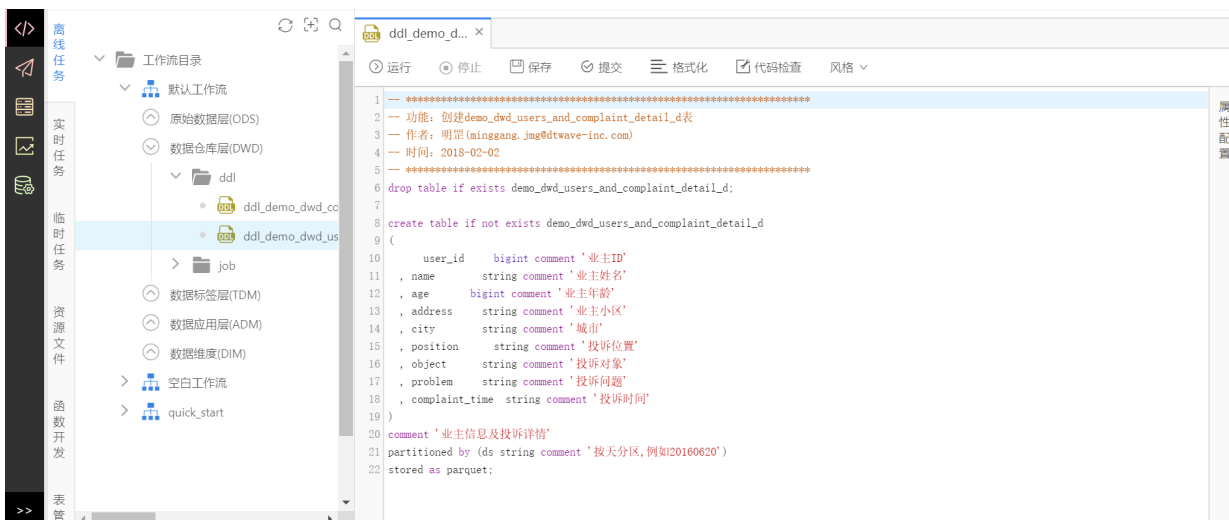
```
-- *****
-- 功能：创建demo_dwd_users_and_complaint_detail_d表
-- 作者：明昱
-- 时间：2018-02-02
-- *****

drop table if exists demo_dwd_users_and_complaint_detail_d;
```

```

create table if not exists demo_dwd_users_and_complaint_detail_d
(
    user_id      bigint comment '业主ID'
    , name       string comment '业主姓名'
    , age        bigint comment '业主年龄'
    , address    string comment '业主小区'
    , city       string comment '城市'
    , position   string comment '投诉位置'
    , object     string comment '投诉对象'
    , problem    string comment '投诉问题'
    , complaint_time string comment '投诉时间'
)
comment '业主信息及投诉详情'
partitioned by (ds string comment '按天分区,例如20160620')
stored as parquet;

```



(6) 在数据仓库层(DWD) -> job 目录下新建离线SparkSQL任务，任务名为 demo_dwd_complaint_detail_d，运行参数：bizDate = 20181111，上游任务：demo_ods_complaint_data_d，在界面右侧属性配置中运行参数里填写参数名为bizDate,参数值填今天时间，如：bizDate = 20181111，上游任务：demo_ods_complaint_data_d。

在此任务中使用创建的udf函数解析投诉信息，获取投诉位置、投诉对象、投诉问题。任务语句如下：

```

-- *****
-- 功能：采用算法模型解析投诉信息,获取投诉位置,投诉对象,投诉问题!
-- 函数：extract_complaint('我家里卫生间的马桶漏水了') 返回 {"problem":"漏水","position":"卫生间","object":"马桶"}
-- 函数：get_json_object(extract_complaint('我家里卫生间的马桶漏水了'),'$.position') 返回 卫生间
-- 作者：明罡
-- 时间：2018-02-02
-- *****

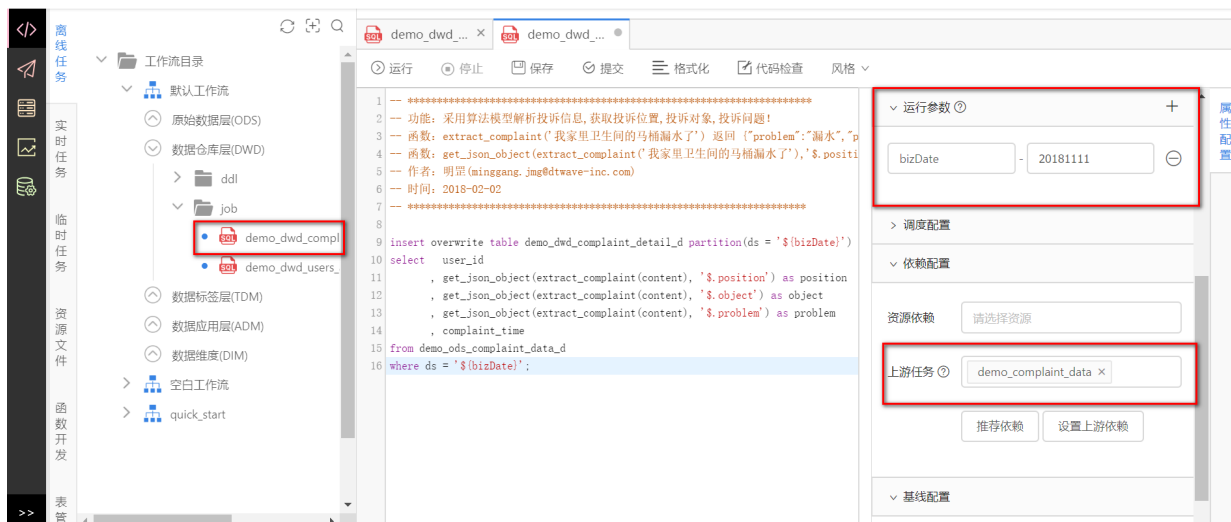
insert overwrite table demo_dwd_complaint_detail_d partition(ds =
'${bizDate}')
select  user_id

```

```

    , get_json_object(extract_complaint(content), '$.position') as
position
    , get_json_object(extract_complaint(content), '$.object') as object
    , get_json_object(extract_complaint(content), '$.problem') as
problem
    , complaint_time
from demo_ods_complaint_data_d
where ds = '${bizDate}';

```



(7) 同样地，在数据仓库层(DWD) -> job 目录下创建

demo_dwd_users_and_complaint_detail_d 任务，运行参数：bizDate = 20181111，上游任务：demo_ods_users_info、demo_dim_address_city、demo_dwd_complaint_detail_d。任务类型为SparkSQL，在此任务中通过多表join获得以业主为对象的所有信息。任务语句如下：

```

-- *****
-- 功能：通过多表join获取业主信息及投诉详情
-- 作者：明盟
-- 时间：2018-02-02
-- *****

insert overwrite table demo_dwd_users_and_complaint_detail_d partition(ds =
'${bizDate}')
select    b.user_id
        , b.name
        , b.age
        , c.address
        , c.city
        , a.position
        , a.object
        , a.problem
        , a.complaint_time

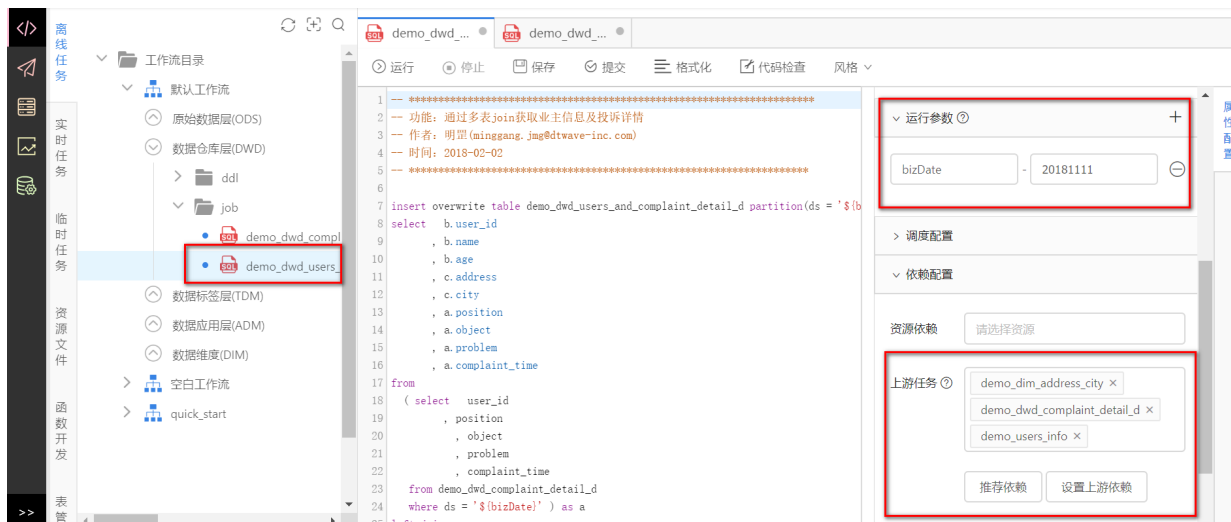
from
    ( select    user_id
              , position

```

```

, object
, problem
, complaint_time
from demo_dwd_complaint_detail_d
where ds = '${bizDate}' ) as a
left join
demo_ods_users_info as b
on b.user_id = a.user_id
left join
demo_dim_address_city as c
on c.address_id = b.address_id;

```



2.6 数据标签层(TDM)

数据标签层(TDM)，通常在此层标签化数据仓库层(DWD)数据。本案例需要获得标签：每个城市历史以来的各投诉对象总次数、各个业主历史以来的投诉总次数、各个业主详细信息及历史以来的投诉位置总次数。具体步骤如下：

(1) 在开发中心的离线任务的数据标签层(TDM)目录下新建ddl与job两个目录。

(2) 创建标签表 `demo_tdm_city_history_object_count_d`（每个城市历史以来的各投诉对象总次数）。在数据标签层(TDM) -> ddl 目录下新建离线DDL任务，任务名为 `ddl_demo_tdm_city_history_object_count_d`，在新建任务页面输入建表语句。任务语句如下：

```

-- *****
-- 功能：创建demo_tdm_city_history_object_count_d表
-- 作者：明昱
-- 时间：2018-02-02
-- *****

-- drop table demo_tdm_city_history_object_count_d;

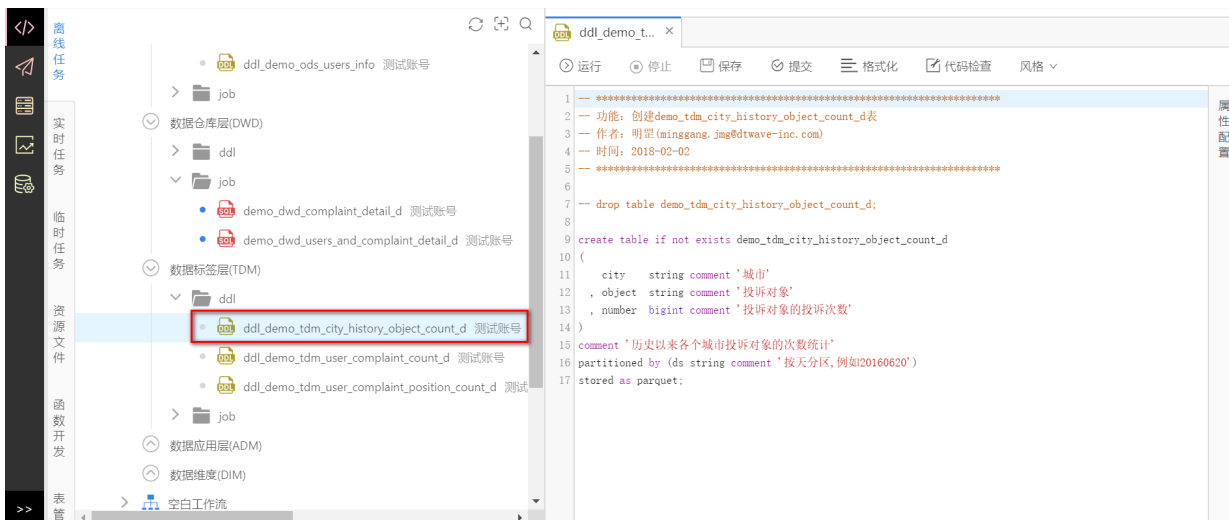
create table if not exists demo_tdm_city_history_object_count_d
(
    city    string comment '城市'

```

```

, object string comment '投诉对象'
, number bigint comment '投诉对象的投诉次数'
)
comment '历史以来各个城市投诉对象的次数统计'
partitioned by (ds string comment '按天分区,例如20160620')
stored as parquet;

```



同样，创建标签表 `demo_tdm_user_complaint_count_d`（各个业主历史以来的投诉总次数），在数据标签层(TDM) -> ddl 目录下创建 `ddl_demo_tdm_user_complaint_count_d` 任务，任务类型为DDL。任务语句如下：

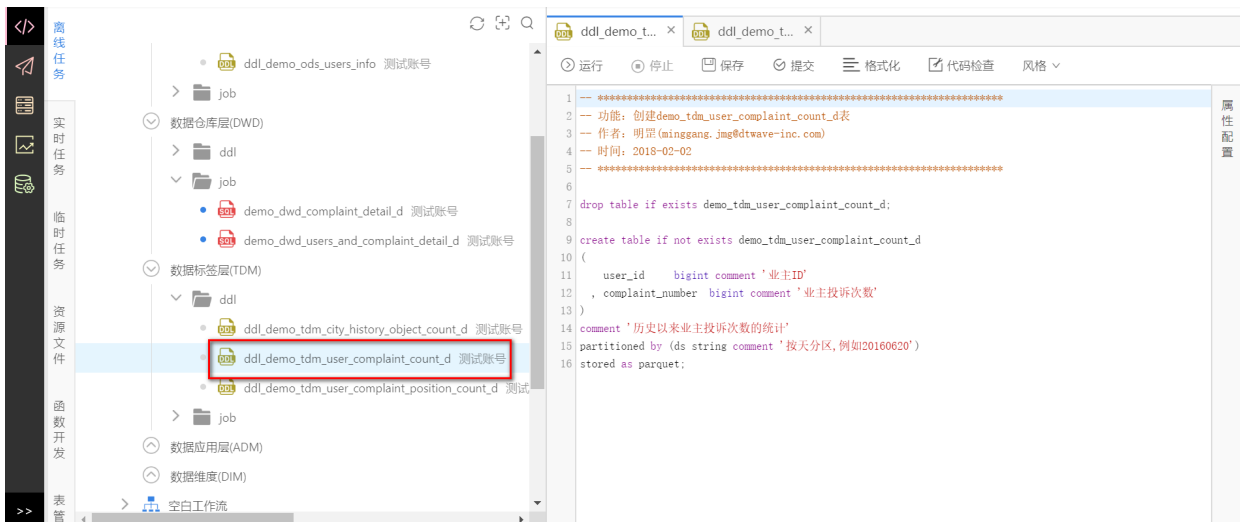
```

-- *****
-- 功能：创建demo_tdm_user_complaint_count_d表
-- 作者：明昱
-- 时间：2018-02-02
-- *****

-- drop table demo_tdm_user_complaint_count_d;

create table if not exists demo_tdm_user_complaint_count_d
(
    user_id      bigint comment '业主ID'
    , complaint_number  bigint comment '业主投诉次数'
)
comment '历史以来业主投诉次数的统计'
partitioned by (ds string comment '按天分区,例如20160620')
stored as parquet;

```



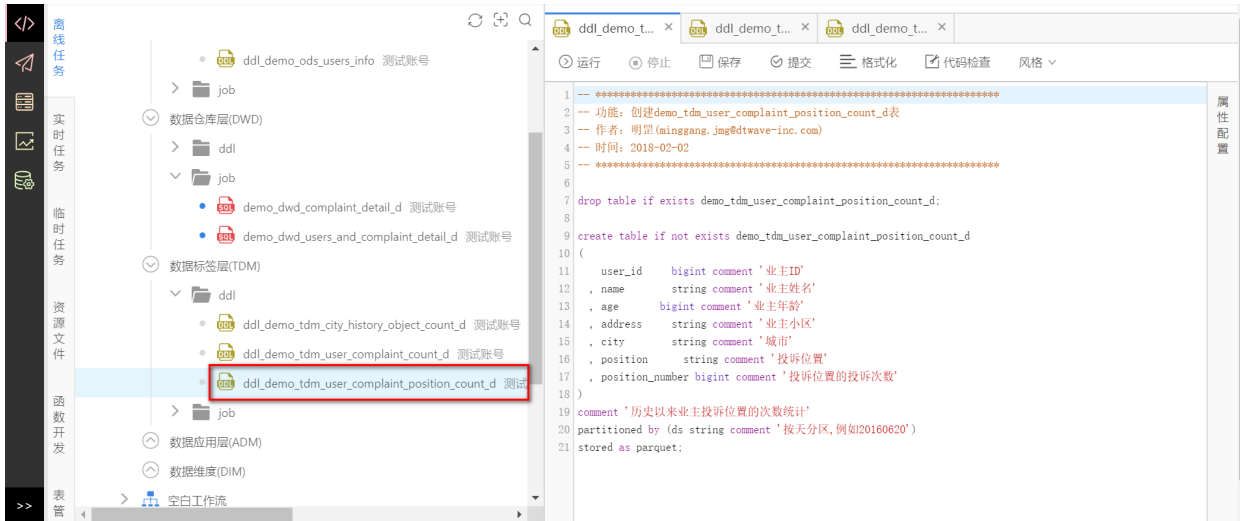
创建标签表 `demo_tdm_user_complaint_position_count_d` (各个业主详细信息及历史以来的投诉位置总次数)，在 **数据标签层(TDM)** -> **ddl** 目录下创建

`ddl_demo_tdm_user_complaint_position_count_d` 任务，任务类型为DDL。任务语句如下：

```
-- *****
-- 功能: 创建demo_tdm_user_complaint_position_count_d表
-- 作者: 明罡
-- 时间: 2018-02-02
-- *****

-- drop table demo_tdm_user_complaint_position_count_d;

create table if not exists demo_tdm_user_complaint_position_count_d
(
    user_id      bigint comment '业主ID'
    , name       string comment '业主姓名'
    , age        bigint comment '业主年龄'
    , address    string comment '业主小区'
    , city       string comment '城市'
    , position   string comment '投诉位置'
    , position_number bigint comment '投诉位置的投诉次数'
)
comment '历史以来业主投诉位置的次数统计'
partitioned by (ds string comment '按天分区,例如20160620')
stored as parquet;
```



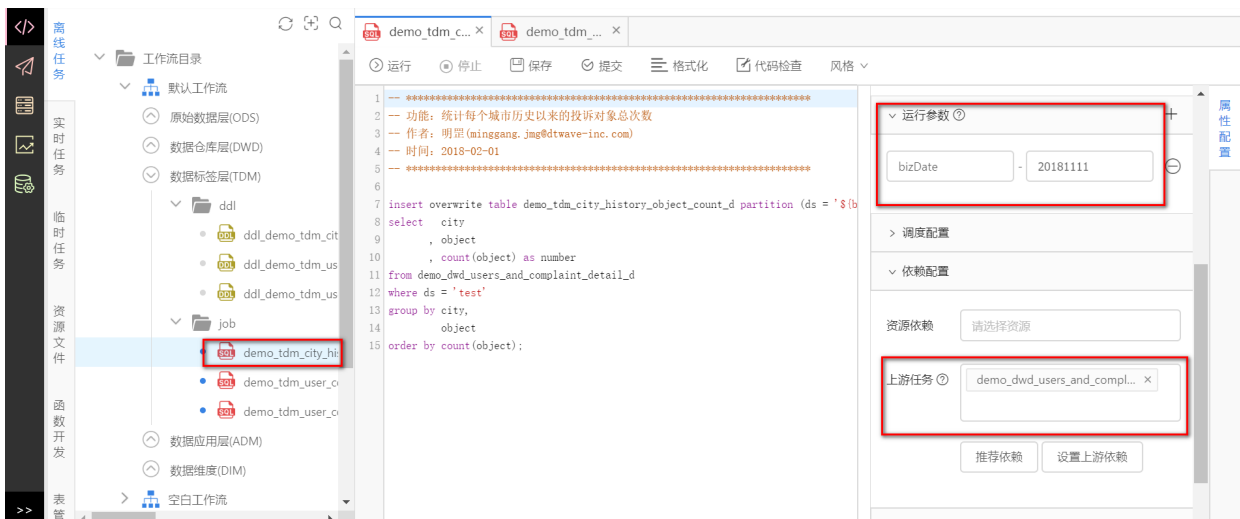
(3) 统计每个城市历史以来的各投诉对象总次数。在 **数据标签层(TDM)** -> **job** 目录下创建SparkSQL任务，任务名为 `demo_tdm_city_history_object_count_d`，运行参数：`bizDate = 20181111`，上游任务：`demo_dwd_users_and_complaint_detail_d`。任务语句如下：

```

-- *****
-- 功能：统计每个城市历史以来的投诉对象总次数
-- 作者：明罡
-- 时间：2018-02-01
-- *****

insert overwrite table demo_tdm_city_history_object_count_d partition (ds =
'${bizDate}')
select  city
      , object
      , count(object) as number
from demo_dwd_users_and_complaint_detail_d
where ds <= '${bizDate}'
group by city,
        object
order by count(object);

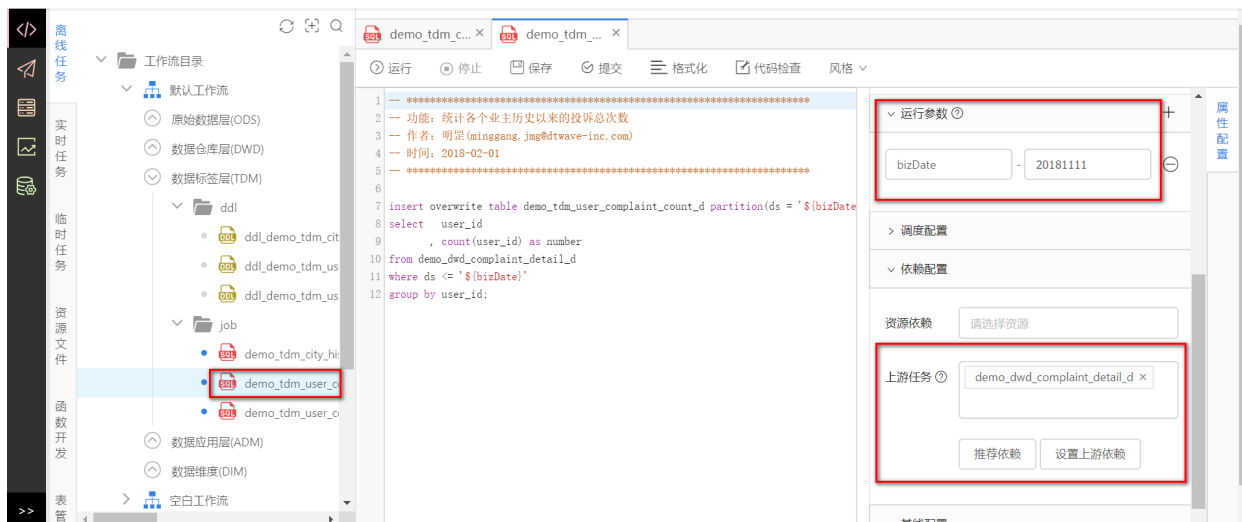
```



(4) 统计各个业主历史以来的投诉总次数。在 **数据标签层(TDM)** -> **job** 目录下创建SparkSQL任务, 任务名为 `demo_tdm_user_complaint_count_d`, 运行参数: `bizDate = 20181111`, 上游任务: `demo_dwd_complaint_detail_d`。任务语句如下:

```
-- *****
-- 功能: 统计各个业主历史以来的投诉总次数
-- 作者: 明罡
-- 时间: 2018-02-01
-- *****

insert overwrite table demo_tdm_user_complaint_count_d partition(ds =
'${bizDate}')
select  user_id
        , count(user_id) as number
from demo_dwd_complaint_detail_d
where ds <= '${bizDate}'
group by user_id;
```



(5) 获取各个业主详细信息及历史以来的投诉位置总次数。在 **数据标签层(TDM)** -> **job** 目录下创建 SparkSQL任务, 任务名为 `demo_tdm_user_complaint_position_count_d`, 运行参数: `bizDate = 20181111`, 上游任务: `demo_dwd_users_and_complaint_detail_d`。任务语句如下:

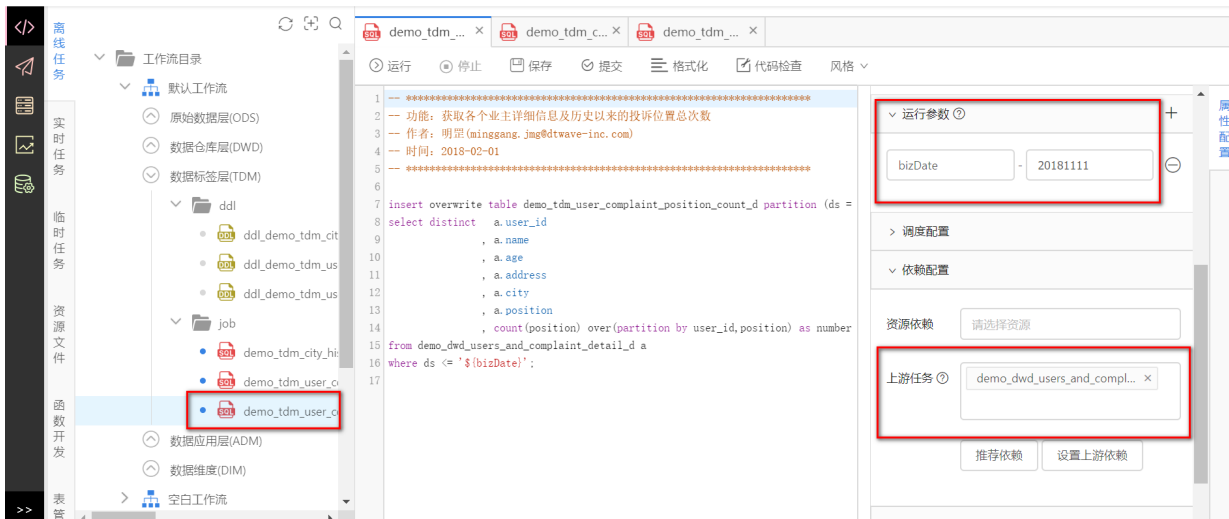
```
-- *****
-- 功能: 获取各个业主详细信息及历史以来的投诉位置总次数
-- 作者: 明罡
-- 时间: 2018-02-01
-- *****

insert overwrite table demo_tdm_user_complaint_position_count_d partition
(ds = '${bizDate}')
select distinct  a.user_id
                 , a.name
                 , a.age
                 , a.address
                 , a.city
```

```

, a.position
, count(position) over(partition by user_id,position) as
number
from demo_dwd_users_and_complaint_detail_d a
where ds <= '${bizDate}';

```



2.7 数据应用层(ADM)

数据应用层(ADM)，按照特定的业务组织标签数据。本案例中此层演示如何统计各个业主详细信息、各个业主历史以来的各投诉位置总次数及投诉总次数，具体步骤如下：

(1) 在开发中心的离线任务的数据应用层(ADM)目录下新建ddl与job两个目录。

(2) 创建标签表 `demo_adm_history_complaint_d`。在数据应用层(ADM) -> `ddl` 目录下创建DDL任务，任务名为 `ddl_demo_adm_history_complaint_d`，在新建任务页面输入建表语句。任务语句如下：

```

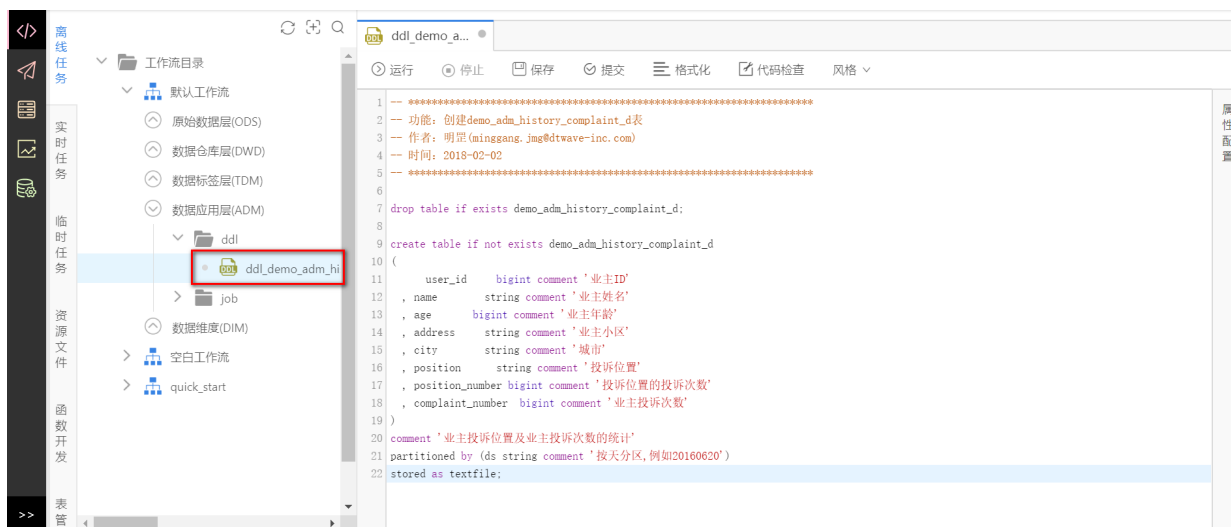
-- *****
-- 功能：创建demo_adm_history_complaint_d表
-- 作者：明昱
-- 时间：2018-02-02
-- *****

-- drop table demo_adm_history_complaint_d;

create table if not exists demo_adm_history_complaint_d
(
    user_id      bigint comment '业主ID'
, name         string comment '业主姓名'
, age         bigint comment '业主年龄'
, address     string comment '业主小区'
, city       string comment '城市'
, position    string comment '投诉位置'
, position_number bigint comment '投诉位置的投诉次数'
, complaint_number bigint comment '业主投诉次数'
)

```

```
)
comment '业主投诉位置及业主投诉次数的统计'
partitioned by (ds string comment '按天分区,例如20160620')
stored as parquet;
```



(5) 统计各个业主详细信息、各个业主历史以来的各投诉位置总次数及投诉总次数，在数据应用层(ADM) -> job 目录下创建SparkSQL任务，任务名为 demo_adm_history_complaint_d，运行参数：bizDate = 20181111，上游任

务：demo_tdm_user_complaint_position_count_d、demo_tdm_user_complaint_count_d

。

任务语句如下：

```
-- *****
-- 功能：统计各个业主详细信息、各个业主历史以来的各投诉位置总次数及投诉总次数
-- 作者：明昱
-- 时间：2018-02-02
-- *****

insert overwrite table demo_adm_history_complaint_d partition (ds =
'${bizDate}')
select  a.user_id
        , a.name
        , a.age
        , a.address
        , a.city
        , a.position
        , a.position_number
        , b.complaint_number
from demo_tdm_user_complaint_position_count_d as a
join demo_tdm_user_complaint_count_d as b
on a.user_id = b.user_id
and a.ds = '${bizDate}'
and b.ds = '${bizDate}';
```

-- 预览数据

```
select *
from demo_adm_history_complaint_d
where ds = '${bizDate}'
limit 10;
```

The screenshot shows a workflow editor with a task named 'demo_adm_history' selected. The task's configuration is displayed in the main area, including a SQL script and a '运行参数' (Run Parameters) section. The '运行参数' section is highlighted with a red box, showing a parameter 'bizDate' with a value of '20181111'. Below it, the '上游任务' (Upstream Tasks) section is also highlighted, showing dependencies on 'demo_tdm_user_complaint_p...' and 'demo_tdm_user_complaint_c...'.

2.8 配置基线（可跳过）

该步骤可省略，若需要监控任务延迟情况，可通过下述步骤配置。

基线介绍详见本文 [【用户指南-监控管理】](#) 部分，案例配置基线步骤如下：

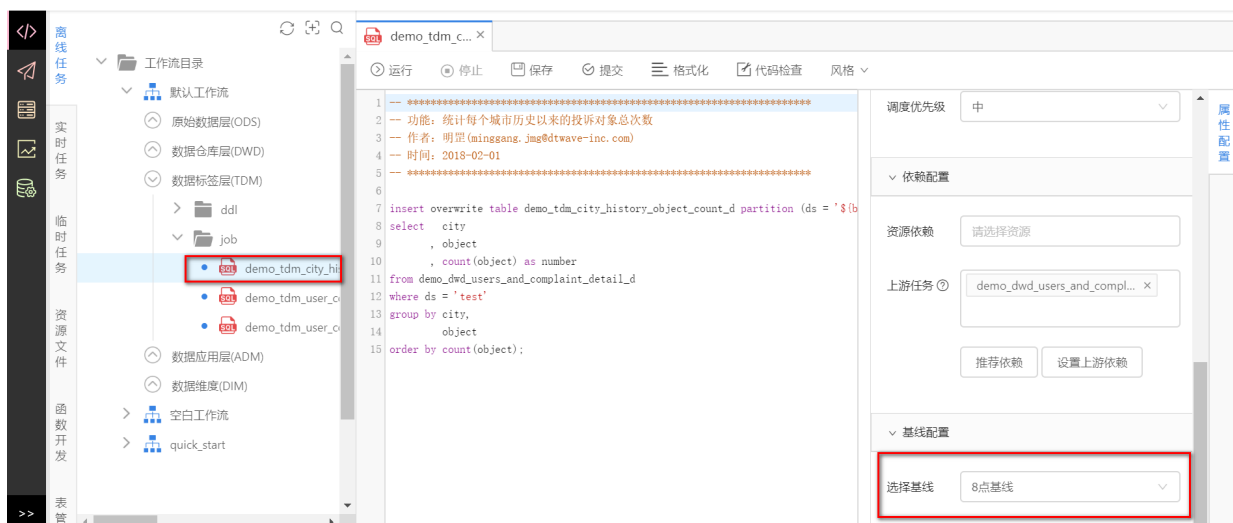
- (1) 创建基线，名称为“8点基线”，告警方式为邮件、短信。

The screenshot shows the 'Baseline Management' (基线管理) configuration interface. The '新增基线' (Add Baseline) button is highlighted. The configuration form shows fields for '所属项目' (shuxi_demo), '基线名称' (8点基线), '优先级' (低), '最晚完成时间' (08:00), '告警对象' (演示账号), '告警方式' (邮件, 短信), '告警时间' (全天), '告警间隔' (5分钟), '最大告警次数' (5), and '告警内容' (你的任务, 因为在基线的最晚完成时间没有完成, 触发了告警, 请及时处理). The '确定' (Confirm) button is highlighted.

- (2) 配置基线

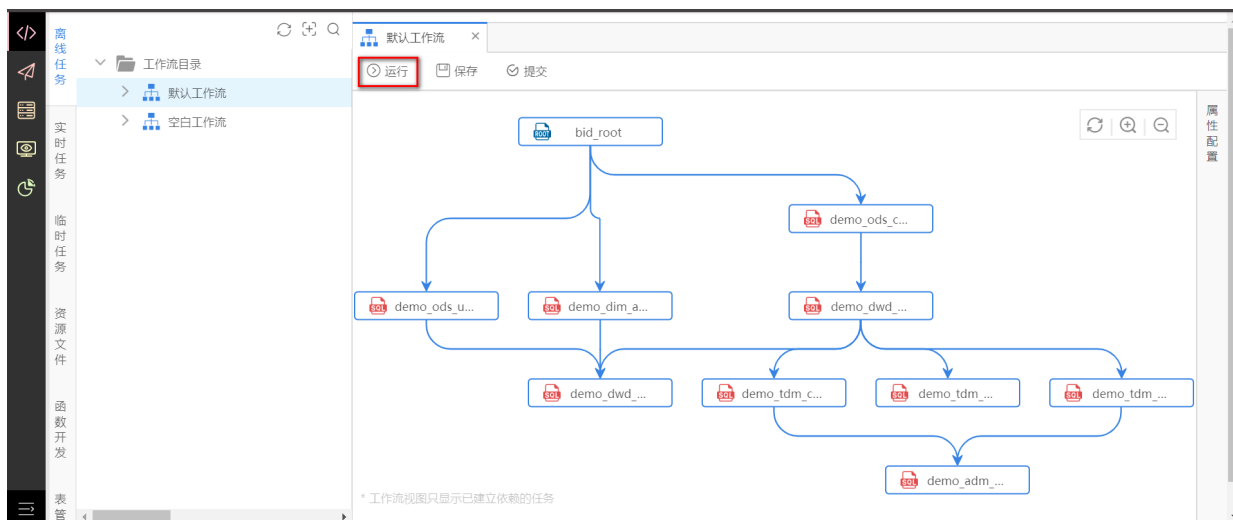
一般只需对项目关键任务配置基线，本案例中对tdm层的

demo_tdm_city_history_object_count_d、demo_tdm_user_history_object_count_d、demo_tdm_user_complaint_position_count_d 三个任务配置基线，在任务的左侧属性配置->基线配置中选择刚才创建的基线。下图为配置基线示例。

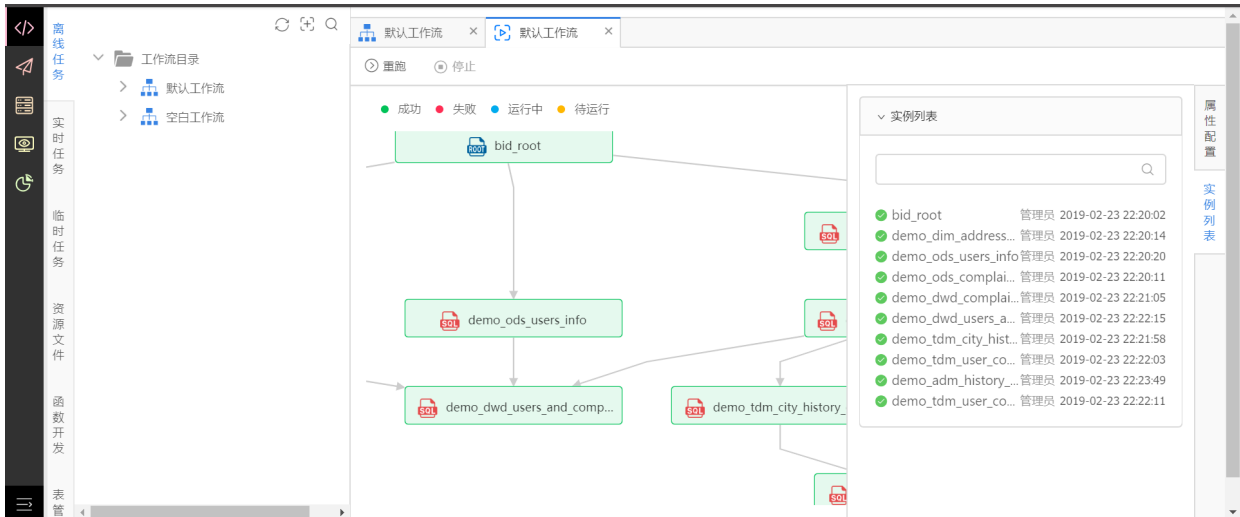


2.9 运行

将所有ddl任务运行成功后，点击工作流，点击运行，可一键运行所有依赖 bid_root 的节点或子节点。



点击运行后，弹出工作流实例运行界面，点击右侧实例列表可定位到具体实例。



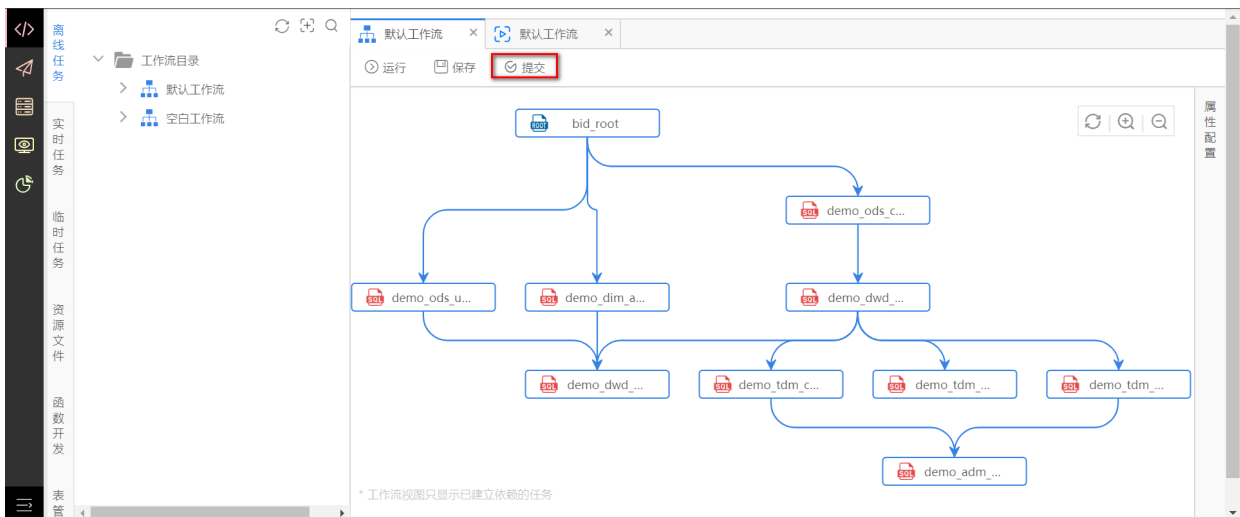
2.10 提交发布

在开发环境下完成的任务、上传的资源、新建的udf函数，测试无误后，可以发布到生产环境依据设置的调度时间进行周期调度运行，DDL任务也需要发布，当DDL任务发布以后，DDL任务立即运行且仅运行一次。步骤如下：

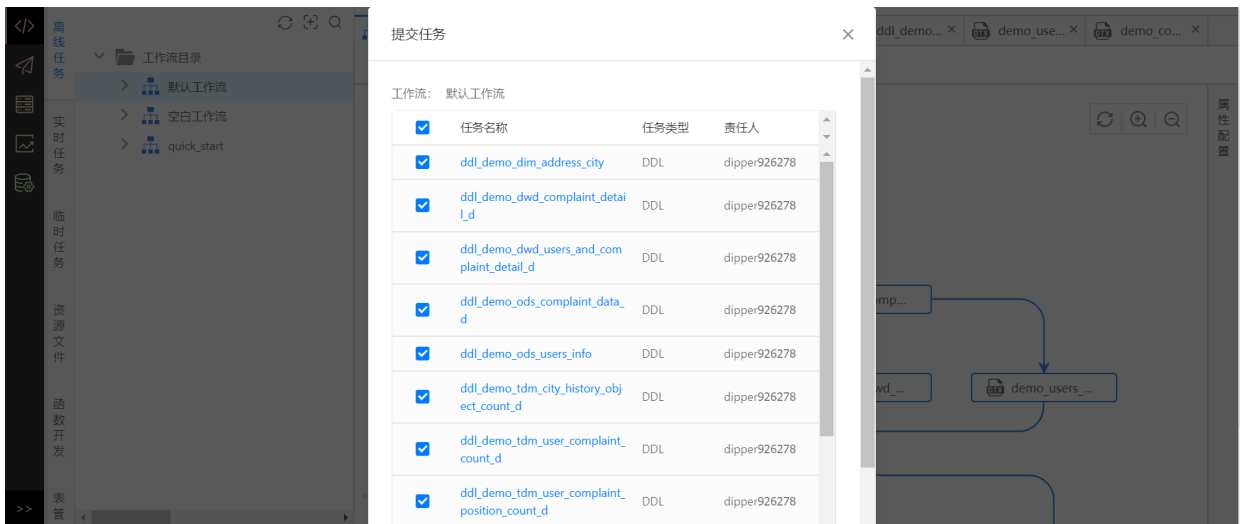
(1) 提交任务

任务在提交之前确保已经配置完成所需要的运行参数、调度配置、依赖配置、基线配置（如果任务不需要配置某项参数则不用配置，DDL任务不需要配置上游任务），且测试无误后，在 workflow 页面，点击提交按钮。

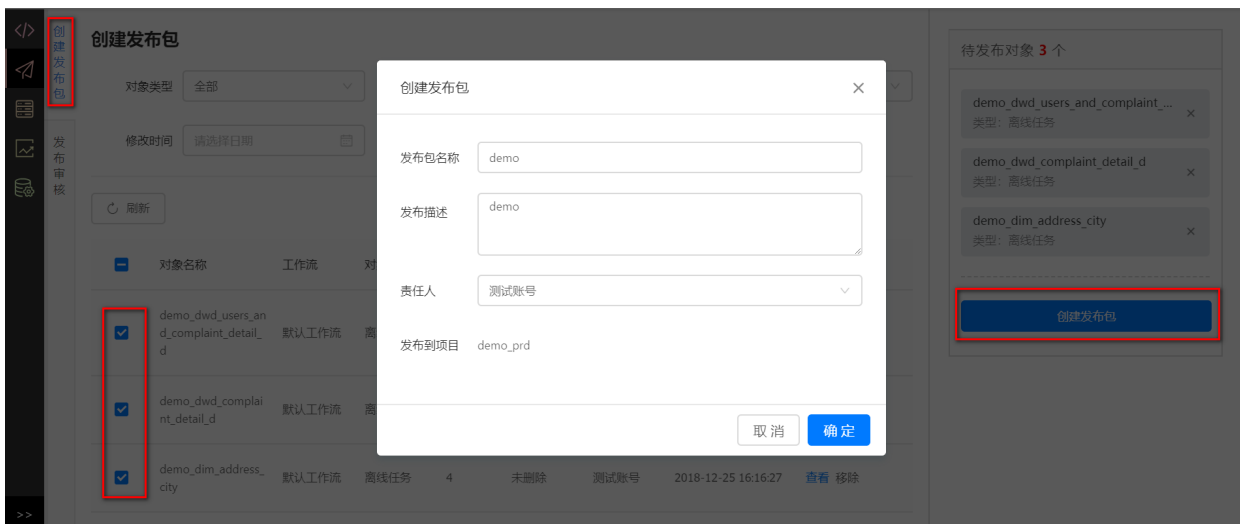
注意：对于没有任务依赖的任务，上游任务设置为 `bid_root`；对于资源、udf函数，系统会默认提交到发布包列表页面。



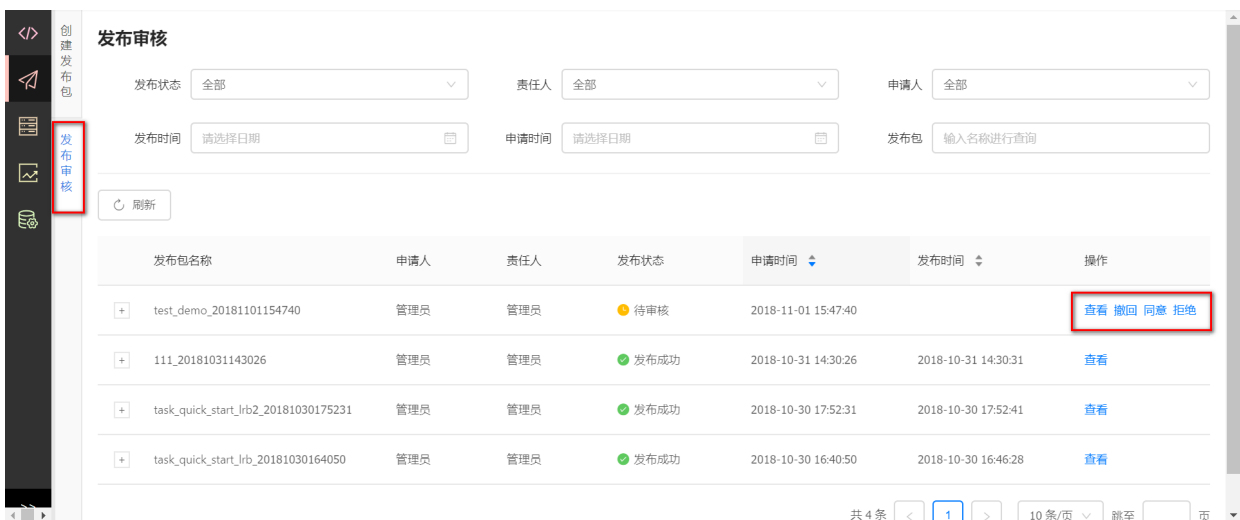
默认全选任务，点击确定完成提交。



(2) 在发布中心的创建发布包页面下，选中我们提交的任务、资源、函数，这时在右侧的待发布对象中会看到我们选中的任务、资源、函数，然后点击创建发布包，填写发布包名称、发布描述，完成创建发布包。



(2) 具有发布权限的管理员或运维人员，可在运维门户->发布审核下同意发布。



3 运维中心（可跳过）

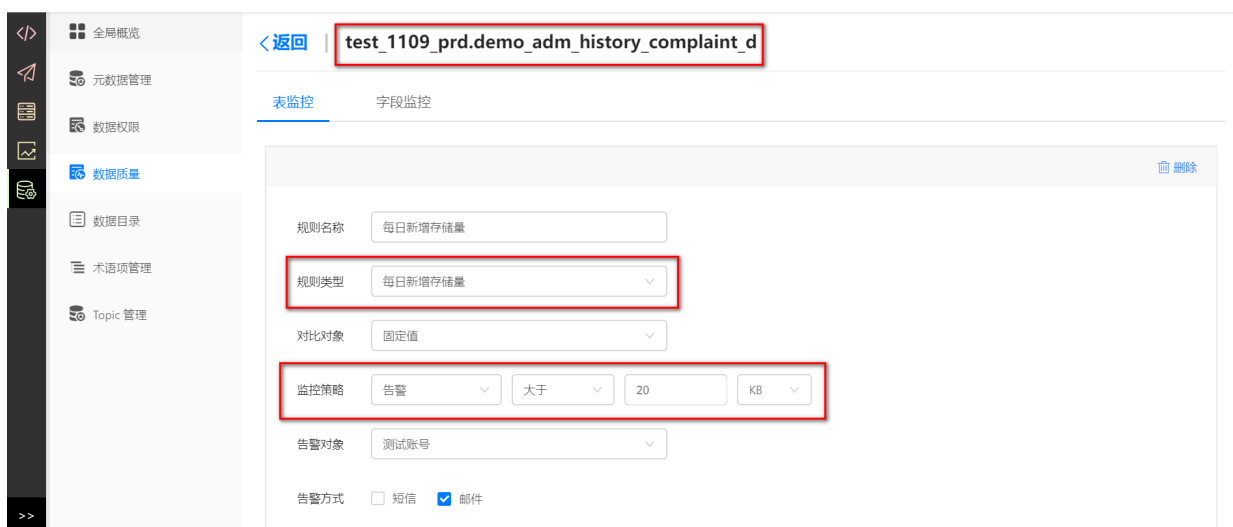
在生产环境的运维中心下可以通过运行总览页可以查看离线任务、流式任务的统计信息，包含运行总任务数、当前时间任务运行数、已完成的任务数、待运行任务数、失败任务数等信息。还可以在离线实例页面查看实例上下游依赖、实例运行状态、实例基本信息等，进行展开父节点、展开子节点、查看运行日志、查看代码、终止、重跑、重跑下游、置成功等操作。

具体操作介绍详见本文 [8 运维中心](#)。

4 数据管理（可跳过）

4.1 配置数据质量

在数据质量中对生产环境库中 `demo_adm_history_complaint_d` 表进行每日新增存储量进行监控配置，配置如下图所示，其他数据质量规则配置详见 [【用户指南-数据管理-数据质量】](#) 部分。



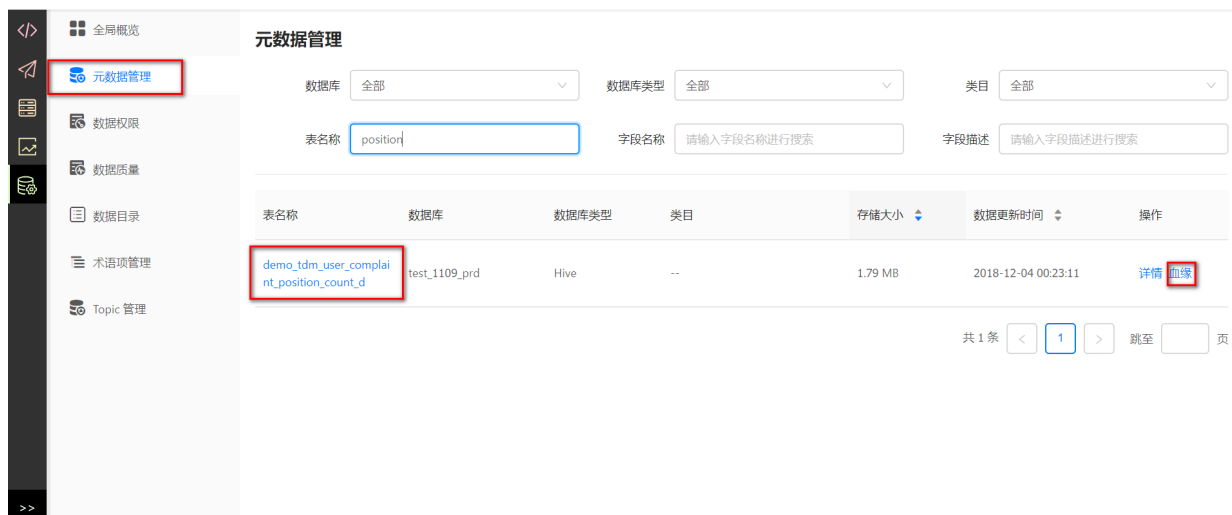
4.2 配置生命周期

在表详情中生命周期信息中，选择需要的周期，点击“确定”按钮，出现系统提示“操作成功”，即设置成功。



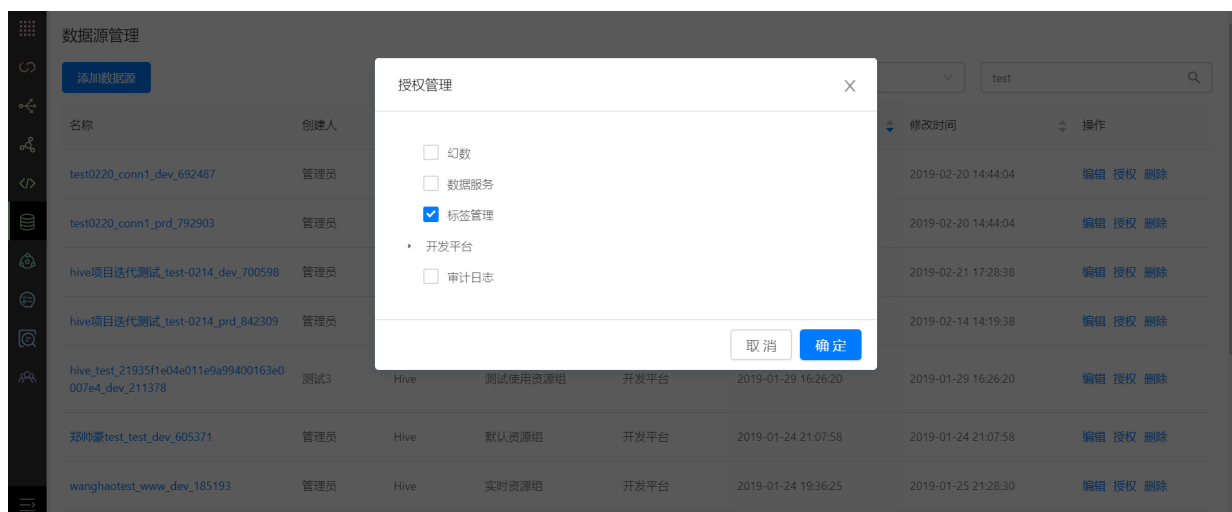
4.3 查看数据血缘

在元数据管理中，查看生产环境中 `demo_tdm_user_complaint_position_count_d` 表的血缘信息。



5 数据源授权

进入数据源管理中，找到本案例hive项目初始化时生成的数据源，并授权到标签管理。并将一个MySQL或GreenPlum数据源授权到数据服务。作为分析引擎的数据源并同步hive数据源中的数据。



6 标签管理

6.1 后台类目

6.1.1 进入后台类目

点击左侧标签管理-后台类目进入后台类目。



6.1.2 创建类目

在左侧目录树部分创建3级类目，如图所示。

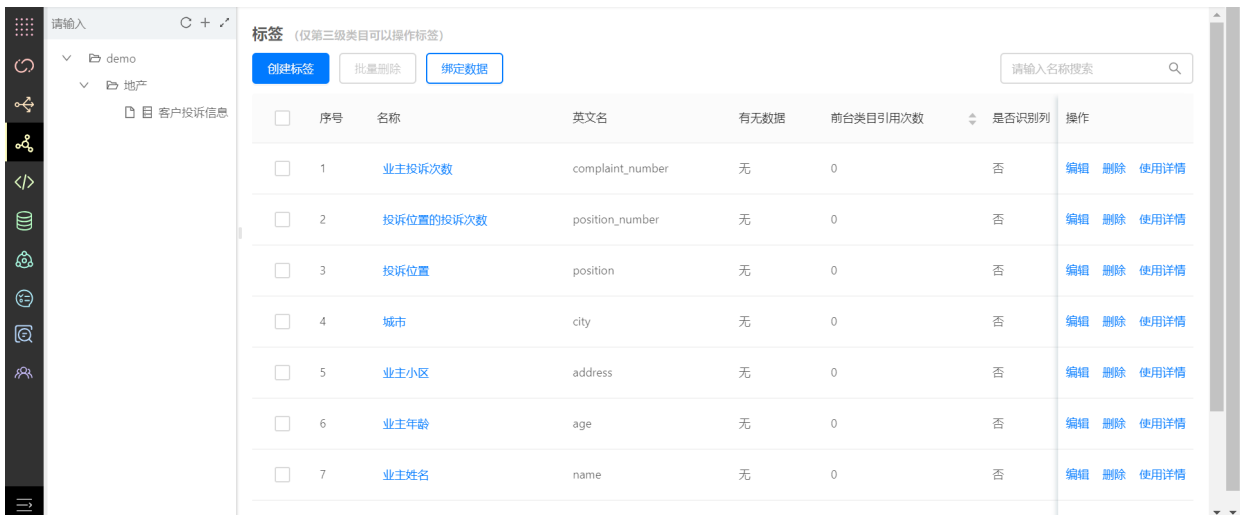


6.1.3 创建标签

创建后台标签，信息项包含“标签名”、“英文名”、“类型”等，如下图所示：

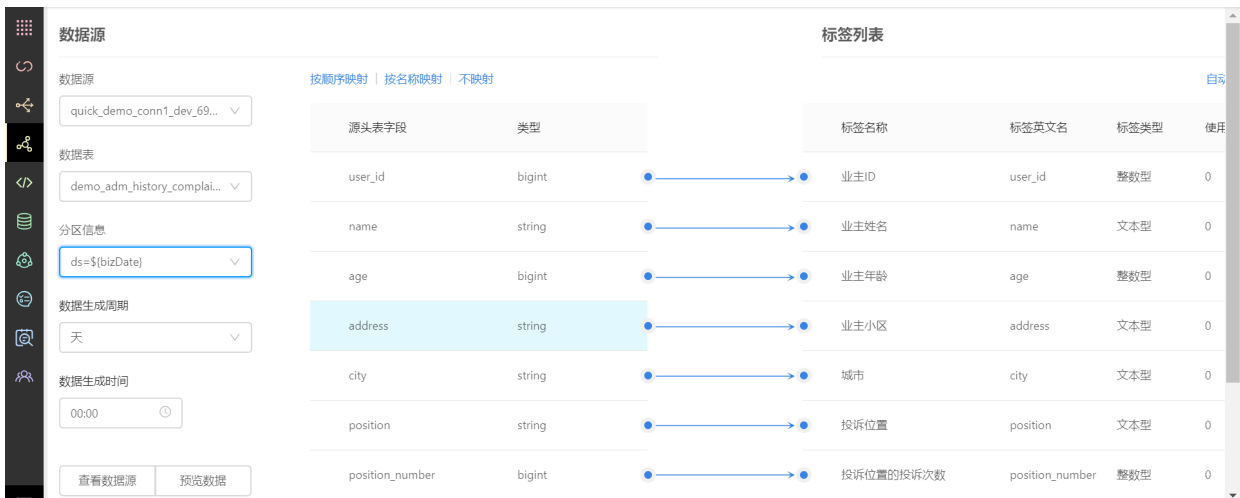


标签创建完成后，如下图所示：



6.1.4 绑定数据

将标签中的字段与数据源表中的字段一一映射，并配置调度周期。



6.2 前台类目

6.2.1 进入前台类目



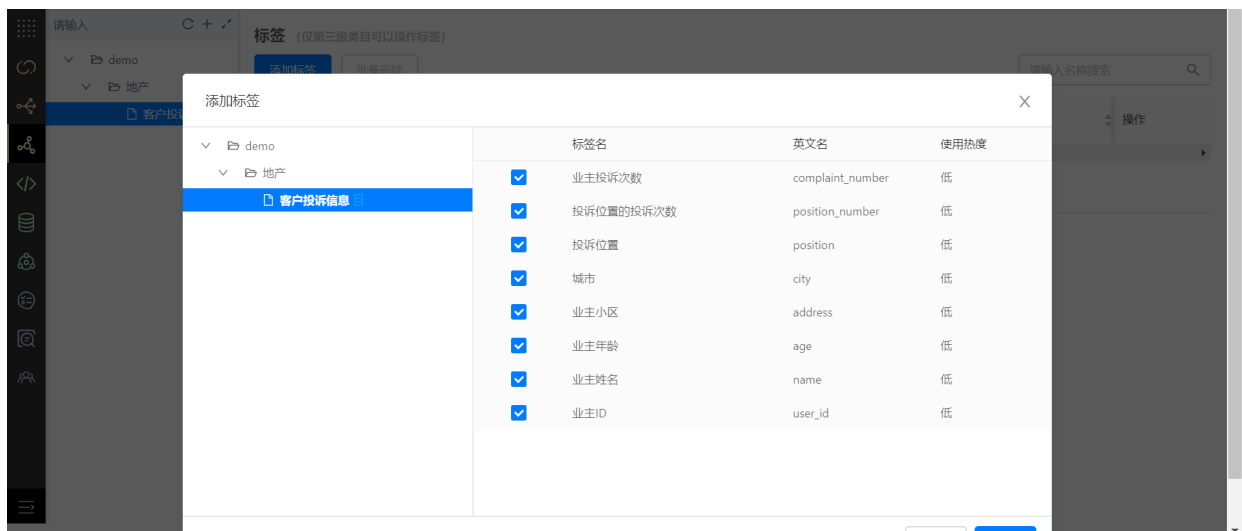
6.2.2 创建类目

在界面左侧创建3级类目，如下图所示。



6.2.3 选择标签

添加前台类目，挑选后台标签即可。



7 数据服务

从控制台中进入数据服务，需要完成创建数据服务、数据配置、服务开发，生成服务API流程。

7.1 创建数据服务

输入“服务名称”、“描述”，选择“分析引擎”，部署环境，创建完成，如下图所示：

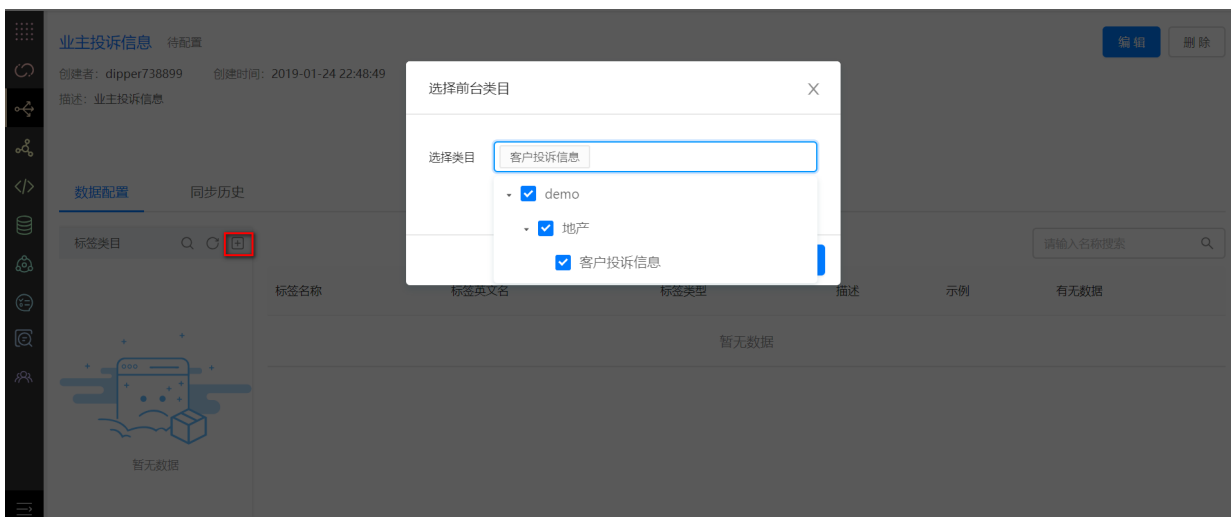
若引擎为空，请进入【管理控制台】，添加【分析引擎】。

若部署环境为空，请进入【数据源管理】，将指定数据源授权给【数据服务】。



7.2 数据配置

选择前台类目，该类目下的标签会自动同步数据，如下图所示：





7.3 服务开发

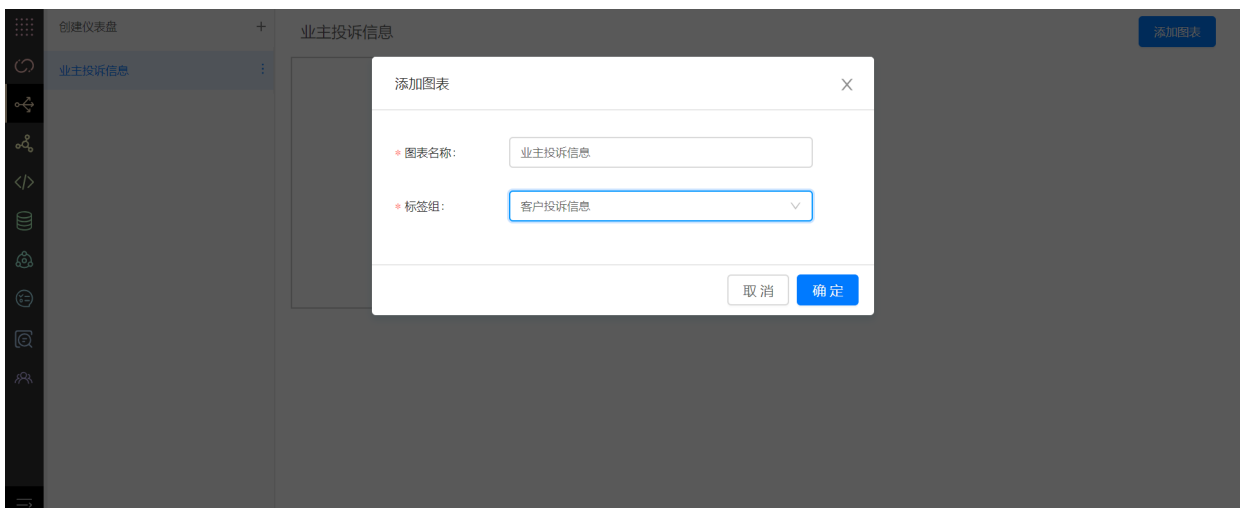
7.3.1 创建仪表盘

点击进入开发，点击创建仪表盘，输入仪表盘名称，点击确定。



7.3.2 创建图表

点击添加图表，输入图标名称，选择标签组，点击确定。



7.3.3 配置图表

设置维度、指标，可对指标进行加工处理，如去重计算、求平均值等，若需要提供筛选项，挑选筛选项对象，即可完成筛选项配置。

The screenshot shows the configuration interface for a chart titled '业主投诉信息'. On the left, there is a sidebar with a '返回' (Return) button and a list of tags including '业主ID', '业主姓名', '业主年龄', '业主小区', '城市', '投诉位置', '投诉位置的投诉次数', and '业主投诉次数'. The main area is divided into sections for '维度' (Dimensions), '数值' (Metrics), and '筛选器' (Filters). Under '维度', '业主ID', '业主姓名', '业主年龄', '业主小区', '城市', and '投诉位置' are selected. Under '数值', '业主投诉次数(求和)' is selected. Below these settings is a data table with 7 columns: '业主ID', '业主姓名', '业主年龄', '业主小区', '城市', '投诉位置', and '业主投诉次数(求和)'. The table contains 3 rows of data. On the right, there is a '标题' (Title) field containing '业主投诉信息' and a '动态筛选器' (Dynamic Filter) section with an '编辑' (Edit) button.

业主ID	业主姓名	业主年龄	业主小区	城市	投诉位置	业主投诉次数(求和)	业主投诉次数(求和)
1	山强	27	汪街H座	厦门	卧室	4	4
1	山强	27	汪街H座	厦门	厨房	4	4
1	山强	27	汪街H座	厦	浴室	4	4

7.3.4 查看API

返回上一级，点击接口详情，生成该数据服务的API信息，供相应的应用场景使用，如下图所示，即完成数据服务。

The screenshot shows the '接口详情' (API Details) dialog box. The dialog has a title bar with '删除' (Delete) and '接口详情' (API Details) buttons. The main content area displays the following information:

- URL地址:** `http://api-in.dtwave.com/api/v12/xengine_gateway/third_api/2818/f8cd4cdf9fd0889f815ffe31f369c6/590`
- Method:** POST
- Request Body:**

```
{  "filter": [],  "chartId": 590,  "method": "render",  "tenantId": 1724,  "serviceKey": "f8cd4cdf9fd0889f815ffe31f369c6",  "userId": 2763}
```

This screenshot is a closer view of the '接口详情' (API Details) dialog box, showing the request body in a code editor. The request body is a JSON object with the following structure:

```
{  "filter": [],  "chartId": 590,  "method": "render",  "tenantId": 1724,  "serviceKey": "f8cd4cdf9fd0889f815ffe31f369c6",  "userId": 2763}
```

8 幻数

8.1 创建项目

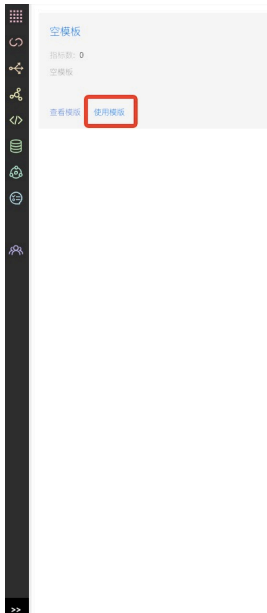
1.在管理控制台中，**点击幻数产品**；或者在左侧菜单栏中，鼠标移入幻数，**点击项目列表**，都可以进入幻数的项目列表页面。



2.在项目列表页面，**点击新建项目按钮**。



3.选择要使用的模板，可以选择空模板也可以选择非空模板，**点击使用模板按钮**。



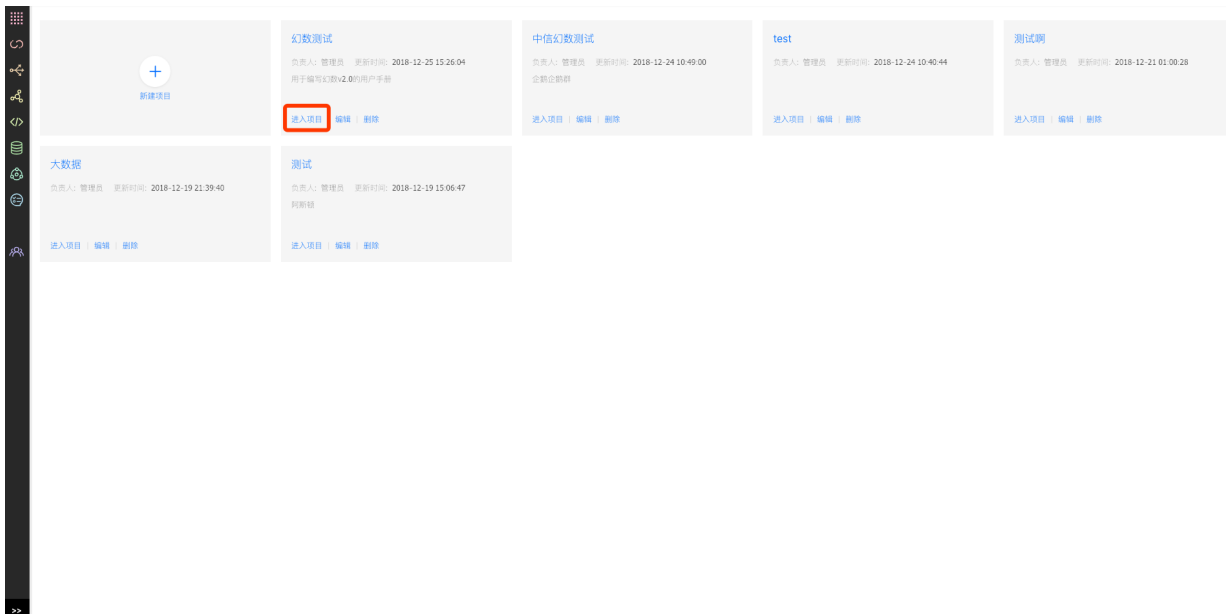
4.选择模板后，需要填写项目名称和描述。填写完成后，点击**确定按钮**，项目创建成功，可以在项目列表中看到新创建的项目。

注意：

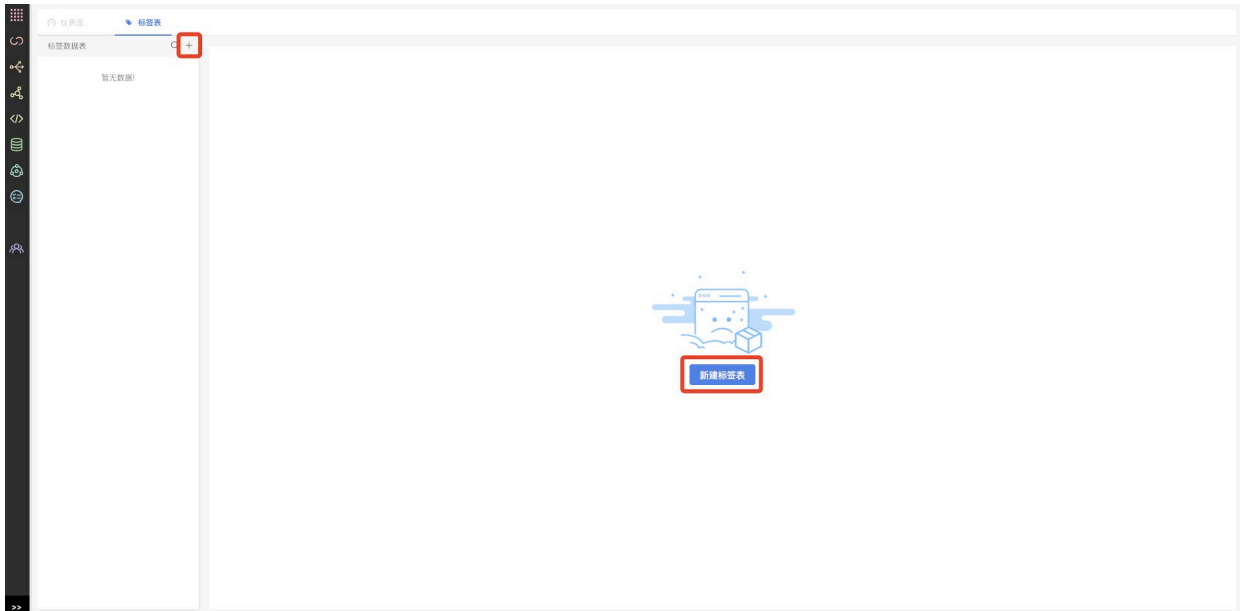
使用非空模板，会初始化相应的前后台标签、数据映射关系、数据服务、标签表以及仪表盘。

8.2 添加标签数据表

1.选择一个要操作的项目，点击**进入项目按钮**。



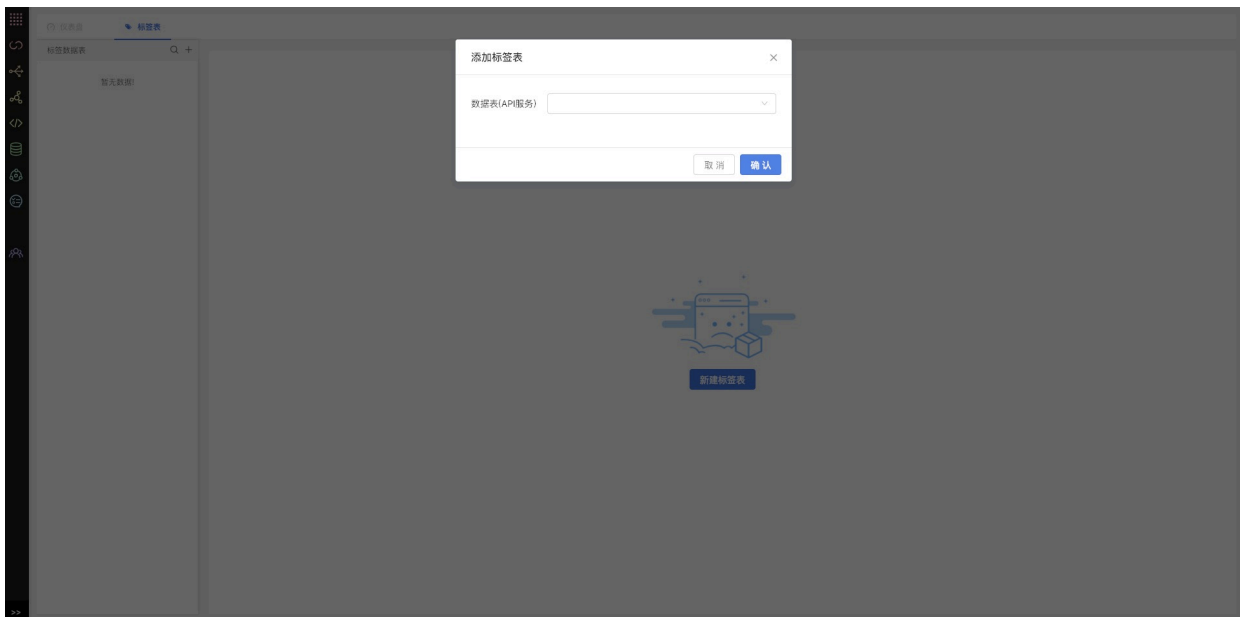
2.选择标签表选项卡，点击**添加标签表按钮**。



注意：

如果该项目没有添加标签表，则进入项目时，默认进入标签表页面，且仪表盘选项卡置灰不可点击。只有标签表添加之后，仪表盘才可点击。

3.选择数据服务：业主投诉，点击**确定**后，添加标签表成功。

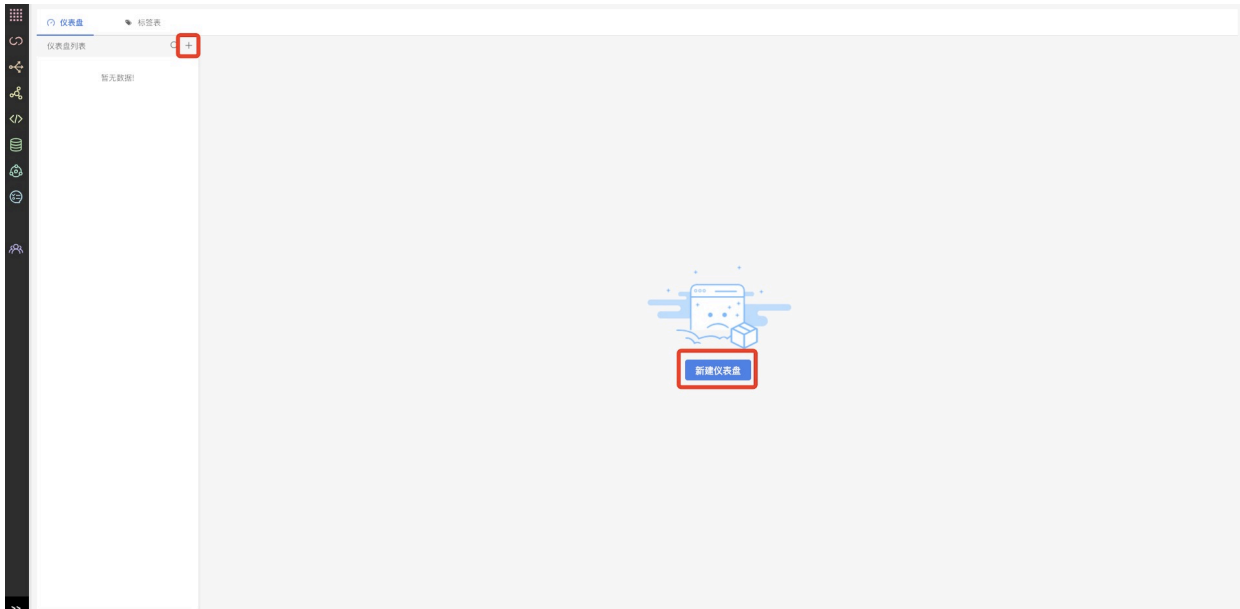


注意：

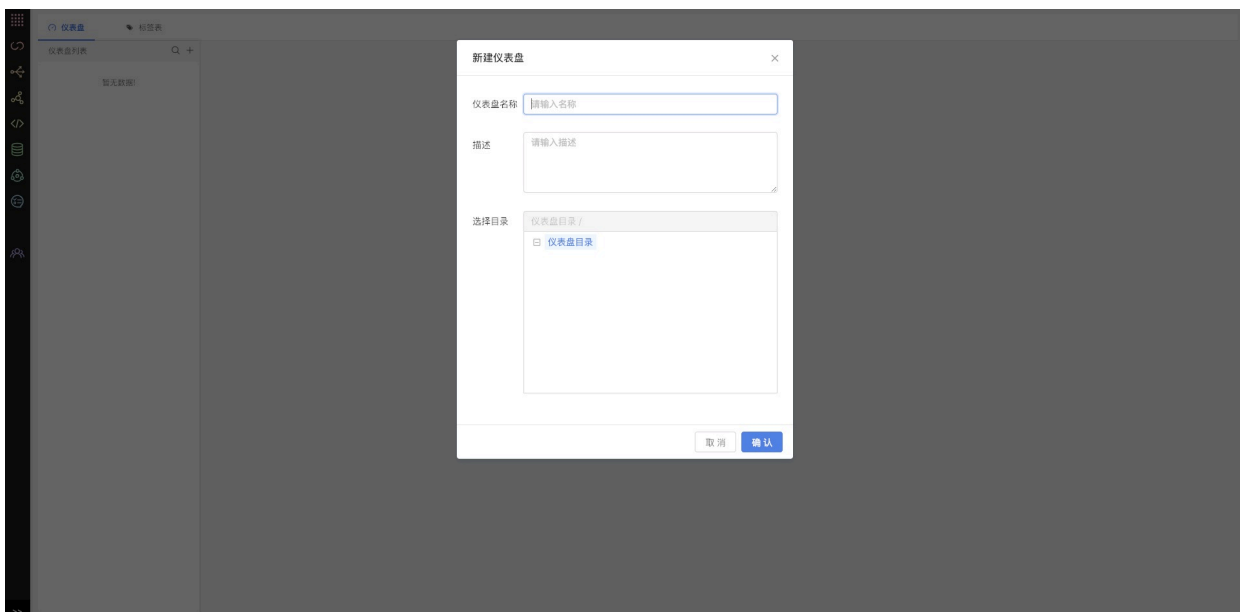
数据服务添加后，该服务下的所有标签表会添加到该项目。

8.3 添加仪表盘

1.切换头部选项卡，选择仪表盘，在仪表盘页面，点击**新建仪表盘**按钮。



2. 在新建仪表盘弹窗中，填写名称和描述，选择目录后，点击确定按钮，添加仪表盘成功。

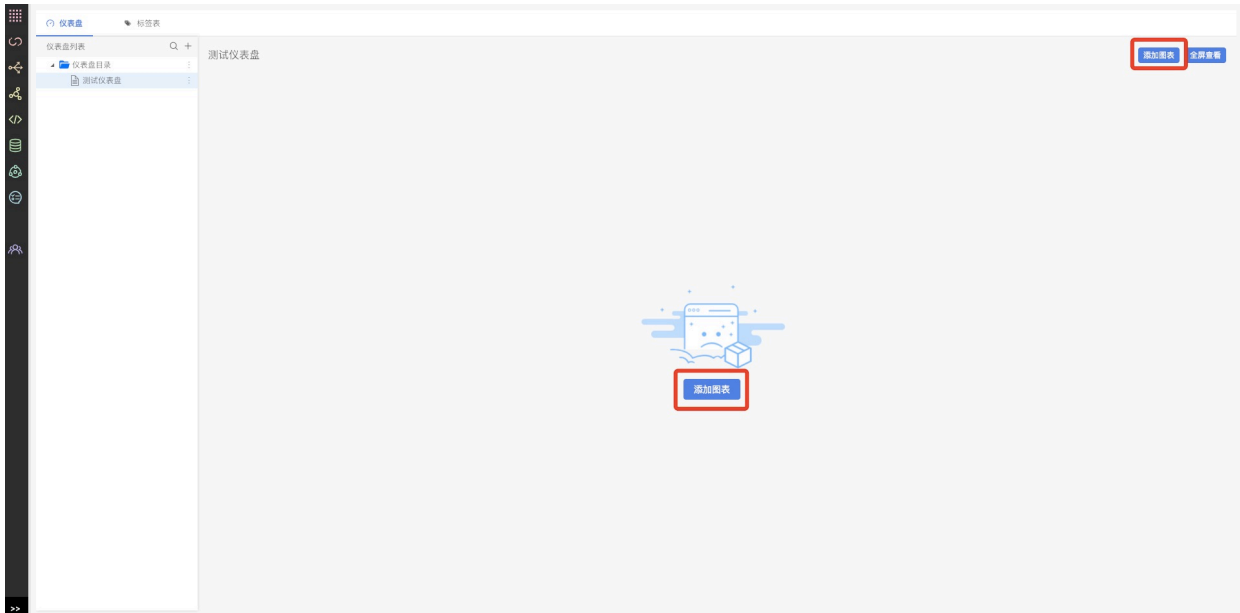


注意：

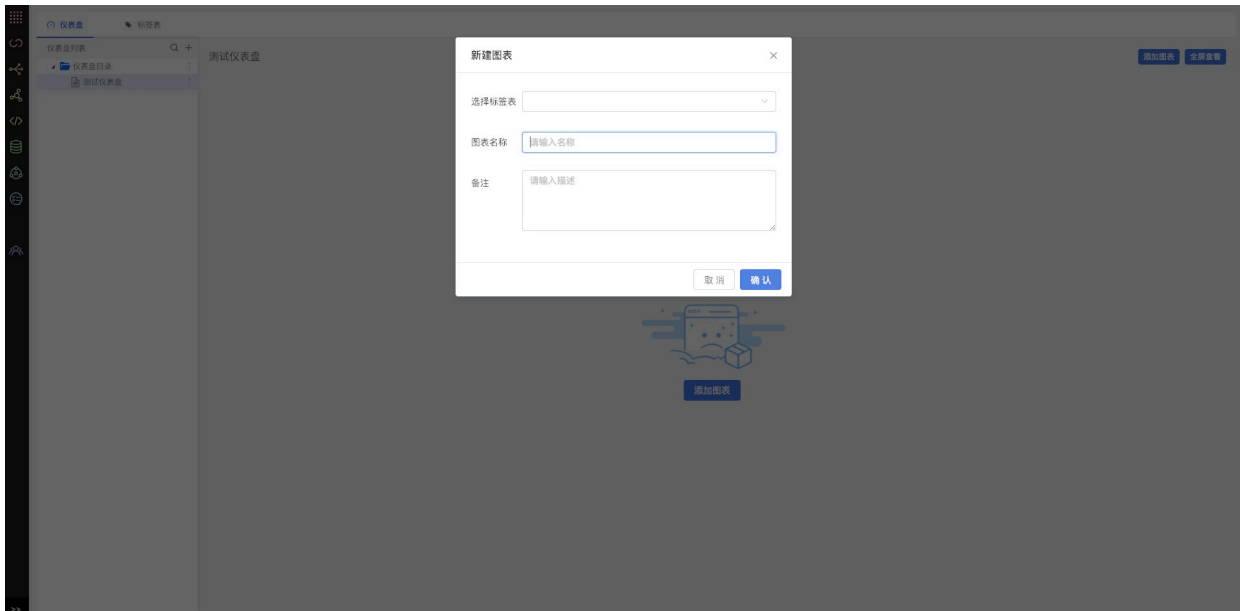
仪表盘只有在标签表中添加了标签表之后才可以使用。

8.4 添加图表

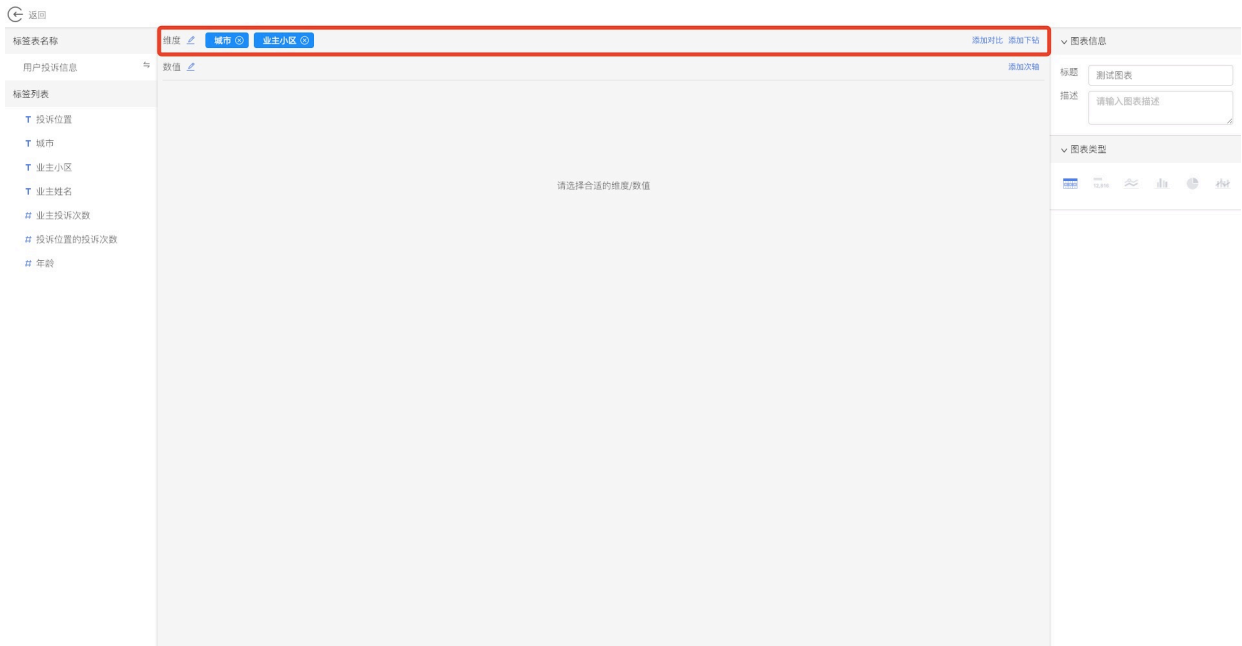
1. 在仪表盘页面，选择一张仪表盘，点击添加图表按钮。



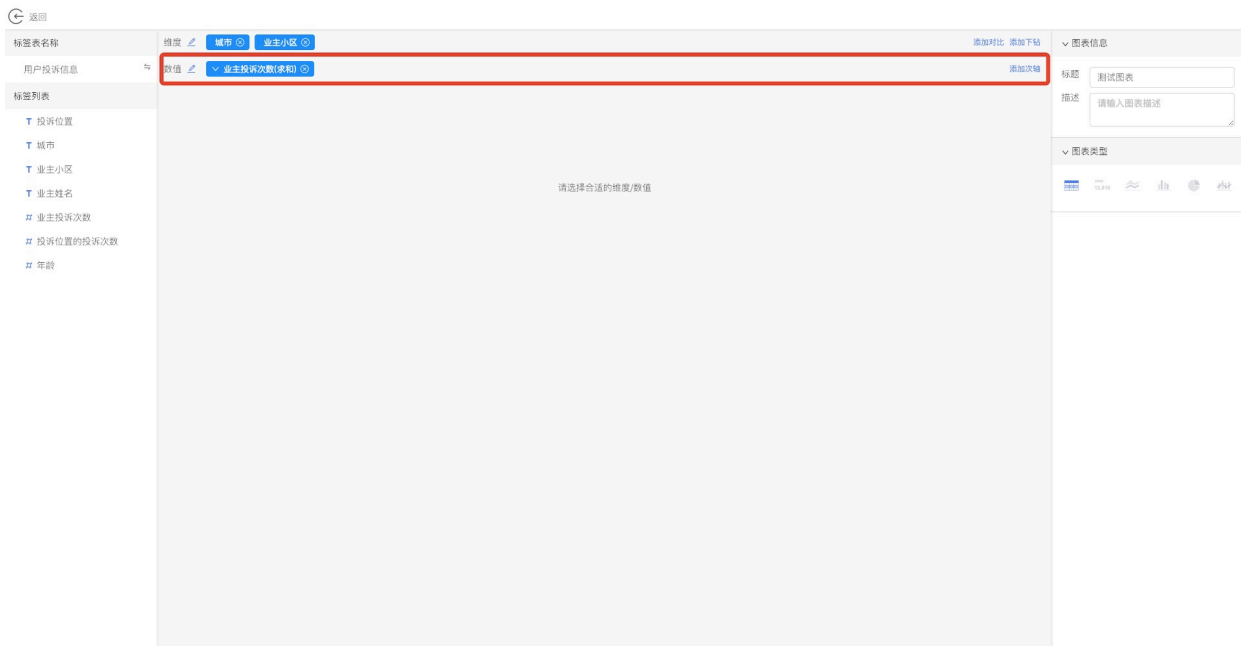
2.选择要使用的标签表：这里我们选择用户投诉信息这张标签表，填写图表名称和备注后，点击确认按钮。



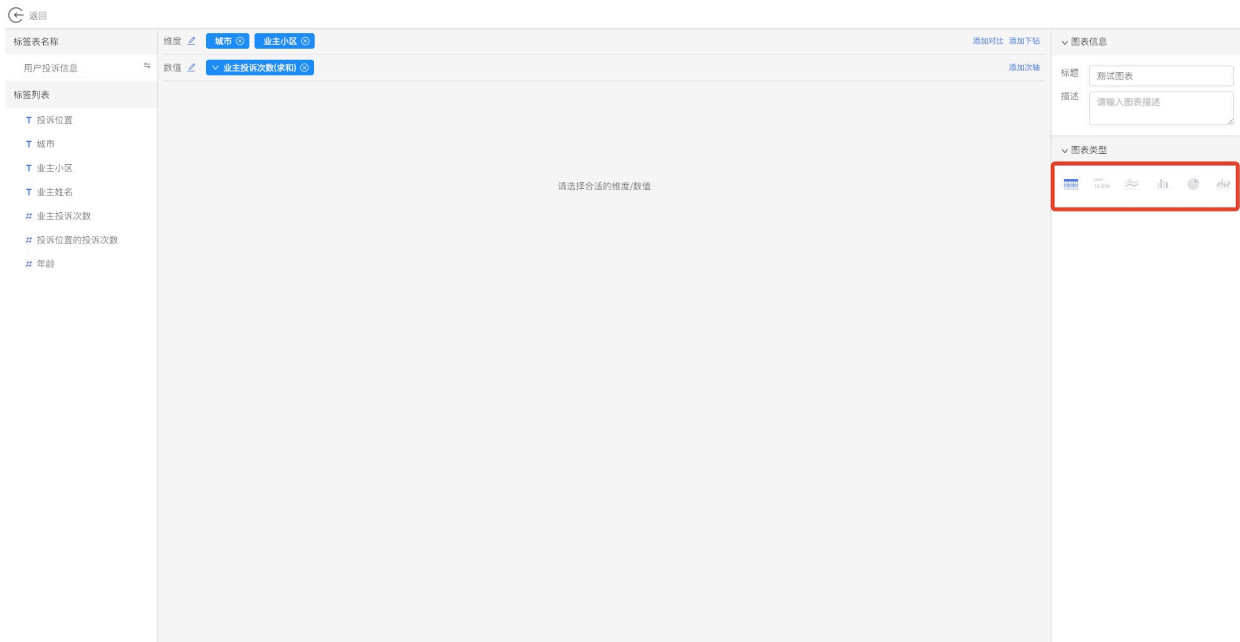
3.从标签列表中选择 一个标签，拖入到维度栏：这里我们拖入城市和业务小区两个标签。维度通常是一个不连续的分类字段。



4.从标签列表中选择**一个**标签，拖入到数值栏：这里我们拖入**业主投诉次数**这个标签。数值通常是一个连续的字段。



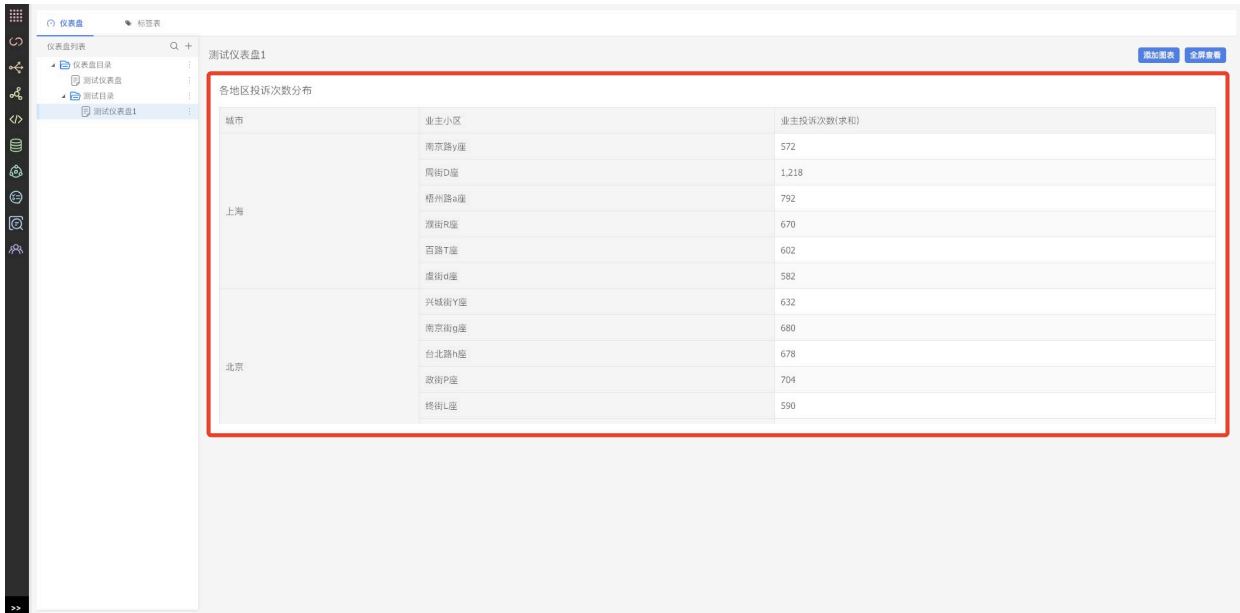
5.在右侧配置区域**选择**图表**一种**图表类型：这里我们选择**表格**类型。



6.在展现区可以看到已经生成的图表，在配置区根据图表提供的个性化配置项进行调整。



7.点击头部返回按钮，返回仪表盘页面就可以看到配置好的图表。



注意：

- 1) 图表配置都是实时保存的，操作了即保存。
- 2) 不是什么图表组件都是可以选择的，只有选择的维度和数值符合组件要求，该组件才可以使用。不符合条件的组件会置灰，处于不可用状态。

权限说明

大数据开发门户平台的角色分为大数据开发门户管理员（租户管理员），（产品）管理员，和产品下的角色。

owner为开通租户的用户，权限等同开发门户管理员，开发门户管理员拥有所有权限。具体权限说明如下：

√：有权限

×：没有权限

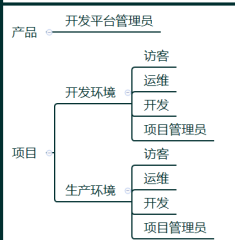
-：若为责任人则有权限，否则没有

大数据开发门户权限说明					
功能模块	角色	权限点划分			
管理控制台 用户中心 用户关系管理 任务管理 资源管理 数据源管理 审计日志 开发平台 标签管理 数据服务 幻数	开发门户 非大数据开发门户管理员 大数据开发门户管理员	开发门户			
		管理控制台	查看产品列表	√	√
		用户中心	查看信息	√所有权限	√所有权限
			修改昵称		
			修改手机号码		
			修改邮箱		
			修改密码		
		用户关系管理	查看部门列表	√所有权限	√
			查看人员列表		√
			新建部门		×
			重命名部门		×
			删除部门		×
			添加人员		×
			删除人员		×
修改人员手机号			×		
修改人员邮箱		×			
修改人员部门		×			

开发门户

	授权人员 批量删除人员 批量授权人员		● ● ●
任务管理	新建任务 查看任务 编辑任务 删除任务 配置任务 下发任务 置完成	●所有权限	●所有权限
资源管理	查看计算引擎列表 新建计算引擎 删除计算引擎 编辑计算引擎 查看计算引擎 查看资源组列表 新建资源组 编辑资源组 删除资源组 授权资源组 查看服务器列表 查看服务器详情 暂停/启动服务器 配置服务器 刷新服务器 编辑服务器 删除服务器	●所有权限	● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
数据源管理	查看数据源列表 新建数据源 编辑数据源 删除数据源 授权数据源	●所有权限	无此模块
审计日志	查看审计日志	●所有权限	●产品管理员有所有权限 ●非产品管理员无权限
开发平台	项目管理 开发中心 发布管理 运维中心 监控管理 数据管理	●所有权限	权限详情请查开发平台
标签管理	标签总览 前台类目 后台类目	●所有权限	●产品管理员有所有权限 ●非产品管理员无权限
数据服务	新建服务 配置服务 进入开发	●所有权限	●产品管理员有所有权限 ●非产品管理员无权限
幻数	项目管理 图表授权 邮件通知	●所有权限	●产品管理员有所有权限 ●非产品管理员无权限

开发中心
发布管理
运维中心
监控管理
数据管理



项目管理						
	权限点	访客	运维	开发	项目管理员	开发平台管理员
项目	新建项目	●	●	●	●	●所有权限
	编辑项目	●	●	●	●	
	初始化项目	●	●	●	●	
	配置项目	●	●	●	●	
	删除项目	●	●	●	●	
	进入开发	●	●	●	●	

开发环境						
	权限点	访客	运维	开发	项目管理员	开发平台管理员
任务	查看任务	●	●	●所有权限	●所有权限	●所有权限
	查看任务属性	●	●			
	运行任务	●	●			
	新建任务	●	●			
	重命名任务	●	●			
	更新任务代码	●	●			
	配置任务属性	●	●			
	停止任务	●	●			
	删除任务	●	●			
	提交任务	●	●			
目录	查看目录	●	●	●所有权限	●所有权限	●所有权限
	删除目录	●	●			
	新建目录	●	●			
	重命名目录	●	●			
工作流	查看工作流依赖	●	●	●所有权限	●所有权限	●所有权限
	查看工作流属性	●	●			
	新建工作流	●	●			
	配置工作流属性	●	●			
	重命名工作流	●	●			
	保存工作流	●	●			
	运行工作流	●	●			

	提交 workflow	●	●			
资源文件	查看资源	●	●	●所有权限	●所有权限	●所有权限
	新建资源	●	●			
	删除资源	●	●			
	编辑资源	●	●			
	提交资源	●	●			
函数	查看函数	●	●	●所有权限	●所有权限	●所有权限
	新建自定义函数	●	●			
	配置自定义函数	●	●			
	删除函数	●	●			
	提交函数	●	●			
表	查看表	●所有权限	●所有权限	●所有权限	●所有权限	●所有权限

生产环境

	权限点	访客	运维	开发	项目管理员	开发平台管理员
任务	查看任务	●	●	●	●	●
	查看任务属性	●	●	●	●	●
	运行任务	●	●	●	●	●
	新建任务	●	●	●	●	●
	重命名任务	●	●	●	●	●
	更新任务代码	●	●	●	●	●
	配置任务属性	●	●	●	●	●
	停止任务	●	●	●	●	●
	删除任务	●	●	●	●	●
	提交任务	●	●	●	●	●
	删除空目录	●	●	●	●	●
目录	新建目录	●	●	●	●	●
	重命名目录	●	●	●	●	●
Workflow	查看 workflow drag图	●	●	●	●	●
	查看 workflow 列表	●	●	●	●	●
	查看 workflow 属性	●	●	●	●	●
	新建 workflow	●	●	●	●	●
	配置 workflow 属性	●	●	●	●	●
	重命名 workflow	●	●	●	●	●
	保存 workflow	●	●	●	●	●
	运行 workflow	●	●	●	●	●
	提交 workflow	●	●	●	●	●
资源文件	查看资源	●	●	●	●	●
	新建资源	●	●	●	●	●
	删除资源	●	●	●	●	●
	编辑资源	●	●	●	●	●
	提交资源	●	●	●	●	●
函数	查看函数	●	●	●	●	●
	新建自定义函数	●	●	●	●	●
	配置自定义函数	●	●	●	●	●
	删除函数	●	●	●	●	●
	提交函数	●	●	●	●	●

发布管理

	权限点	访客	运维	开发	项目管理员	开发平台管理员
创建发布包	查看未发布对象列表	无此模块	●	●所有权限	●所有权限	●所有权限
	查看未发布对象详情		●			
	添加对象		●			
	删除对象		●			
	移除对象		●			
	创建发布包		●			
发布审核	查看发布包列表	●	●	●	●所有权限	●所有权限
	查看发布包详情	●	●	●		
	同意发布	●	●	●		
	拒绝发布	●	●	●		
	撤回发布包	●	●	●		

运维中心

	权限点	访客	运维	开发	项目管理员	开发平台管理员
运行总览	查看运行概览	●所有权限	●所有权限	●所有权限	●所有权限	●所有权限
离线实例	查看离线实例列表	●	●所有权限	●	●所有权限	●所有权限
	查看运行日志	●		●		
	查看代码	●		●		
	展开父节点	●		●		
	展开子节点	●		●		
	重跑	●		●		
	终止	●		●		
离线任务	查看离线任务列表	●所有权限	●所有权限	●所有权限	●所有权限	●所有权限
	展开父节点					
	展开子节点					
	查看代码					
实时任务	查看实时任务列表	●	●所有权限	●	●所有权限	●所有权限
	查看运行记录	●		●		
	查看代码	●		●		
	查看日志	●		●		
	启动	●		●		
	停止	●		●		

开发平台

生产环境						
	权限点	访客	运维	开发	项目管理员	开发平台管理员
运行总览	查看运行概览	●所有权限	●所有权限	●所有权限	●所有权限	●所有权限
工作流实例	展开父节点	●	●所有权限	●	●所有权限	●所有权限
	展开子节点	●		●		
	重跑	●		●		
	终止	●		●		
	置成功	●		●		
	重跑下游	●		●		
离线实例	查看运行日志	●	●所有权限	●	●所有权限	●所有权限
	查看代码	●		●		
	展开父节点	●		●		
	展开子节点	●		●		
	重跑	●		●		
	终止	●		●		
	置成功	●		●		
重跑下游	●		●			
离线任务	展开父节点	●	●所有权限	●	●所有权限	●所有权限
	展开子节点	●		●		
	查看代码	●		●		
	补数据	●		●		
实时任务	查看任务列表	●	●所有权限	●	●所有权限	●所有权限
	查看运行记录	●		●		
	查看代码	●		●		
	查看日志	●		●		
	启动	●		●		
	停止	●		●		

监控管理						
	权限点	访客	运维	开发	项目管理员	开发平台管理员
基线管理	查看基线列表	●	●所有权限	●	●所有权限	●所有权限
	新增基线	●		●		
	编辑基线	●		●		
	删除基线	●		●		
基线告警	告警列表	●	●所有权限		●所有权限	●所有权限
	置为已处理	●		●		
数据质量告警	查看告警列表	●	●所有权限	●	●所有权限	●所有权限
	查看表详情	●		●		
	告警详情	●		●		
	置为已处理	●		●		

数据管理						
	权限点	访客	运维	开发	项目管理员	开发平台管理员
全局概览	查看概览	●所有权限	●所有权限	●所有权限	●所有权限	●所有权限
元数据管理	查看表详情	●	●所有权限	●	●所有权限	●所有权限
	查看表血缘	●		●		
	展开上游表	●		●		
	展开下游表	●		●		
	修改表类型	●		●		
	修改责任人	●		●		
	修改类目	●		●		
	修改生命周期	●		●		
	修改类目	●		●		
	修改术语项	●		●		
数据权限	我的权限	申请权限	无此模块	●	●	●
		权限详情		●	●	●
		查看权限列表		●	●	●
	申请记录	查看申请列表		●	●	●
		查看申请详情		●	●	●
	审核记录	查看审核记录		无审核记录tab页签	无审核记录tab页签	无审核记录tab页签
		查看申请单详情				
收回权限						
数据质量	查看列表	●	●全部权限	●全部权限	●全部权限	●全部权限
	查看表详情	●				
	配置规则	●				
	查看质量报告	●				
	查看规则	●				
数据目录	查看数据目录	●	●全部权限	●全部权限	●全部权限	●全部权限
	查看标签类目	●				
	查看表详情	●				
	修改类目	●				
	重命名	●				
	删除	●				
	修改类目	●				
	添加子类目	●				
	批量修改类目	●				
术语项管理	查看术语项目录	●	●全部权限	●全部权限	●全部权限	●全部权限
	查看表详情	●				
	修改术语项	●				
	重命名	●				
	删除	●				

			修改术语 添加术语 批量修改术语	●				
			HDFS管理 查看文件或文件夹列表 上传文件 新建文件夹 删除文件或文件夹 重命名文件或文件夹 下载文件 查看文件内容	●全部权限	●全部权限	●全部权限	●全部权限	●全部权限
			Topic管理 查看topic列表 进入topic管理页 查看schema 添加schema 编辑schema 查看partition 数据预览	● ● ● ● ● ● ●	● ● ● ● ● ● ●	●全部权限	●全部权限	●全部权限
			配置数据源 添加数据源 查看数据源 移除数据源	● ● ● ●	● ● ● ●	● ● ●	●全部权限	●全部权限
标签管理	标签管理	产品 <input type="text" value="标签管理管理员"/>	●所有权限					
数据服务	数据服务	产品 <input type="text" value="数据服务管理员"/>	●所有权限					
幻数	幻数	产品 <input type="text" value="幻数管理员"/>	●所有权限					
审计日志	审计日志	产品 <input type="text" value="审计日志管理员"/>	●所有权限					

常见问题(FAQ)

1 平台推荐使用哪种浏览器?

我们推荐您使用Chrome浏览器。

2 任务一直卡在调度中怎么处理?

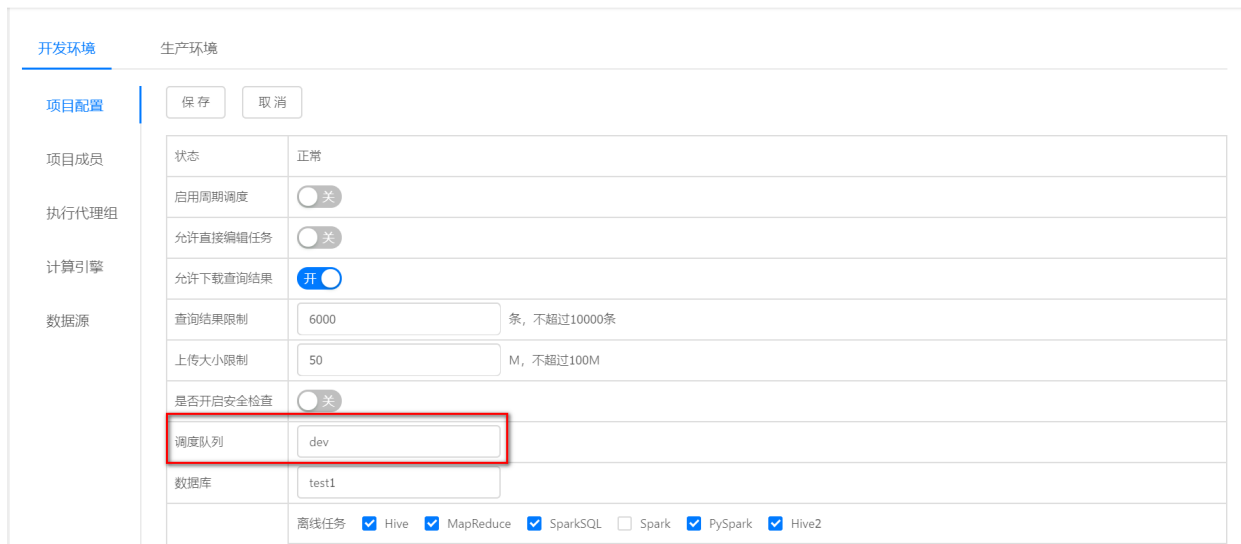
问题描述

r任务运行时一直显示没有可用yarn资源，如下图。



解决方案

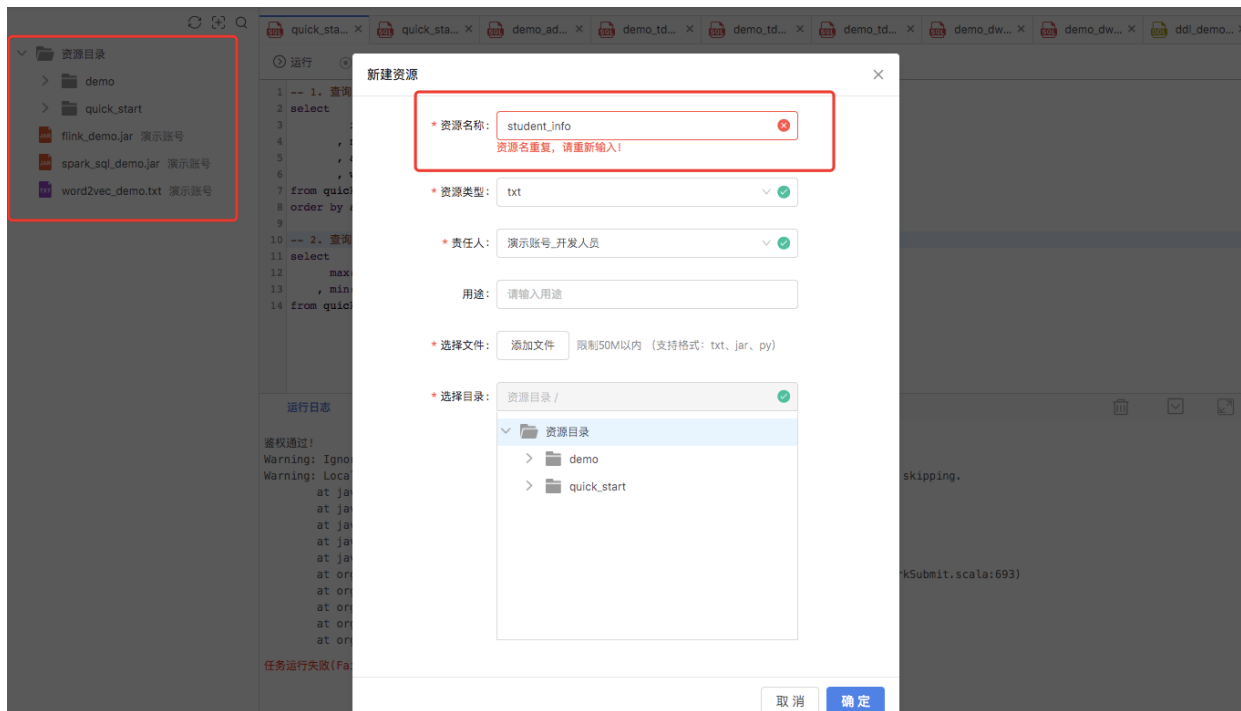
可在项目配置中修改调度队列，为该项目配置有资源的调度队列。



3 创建资源时提示名称重复

问题描述

开发中心中上传资源时，子目录里没有这个资源，却提示资源名称重复，如图所示。



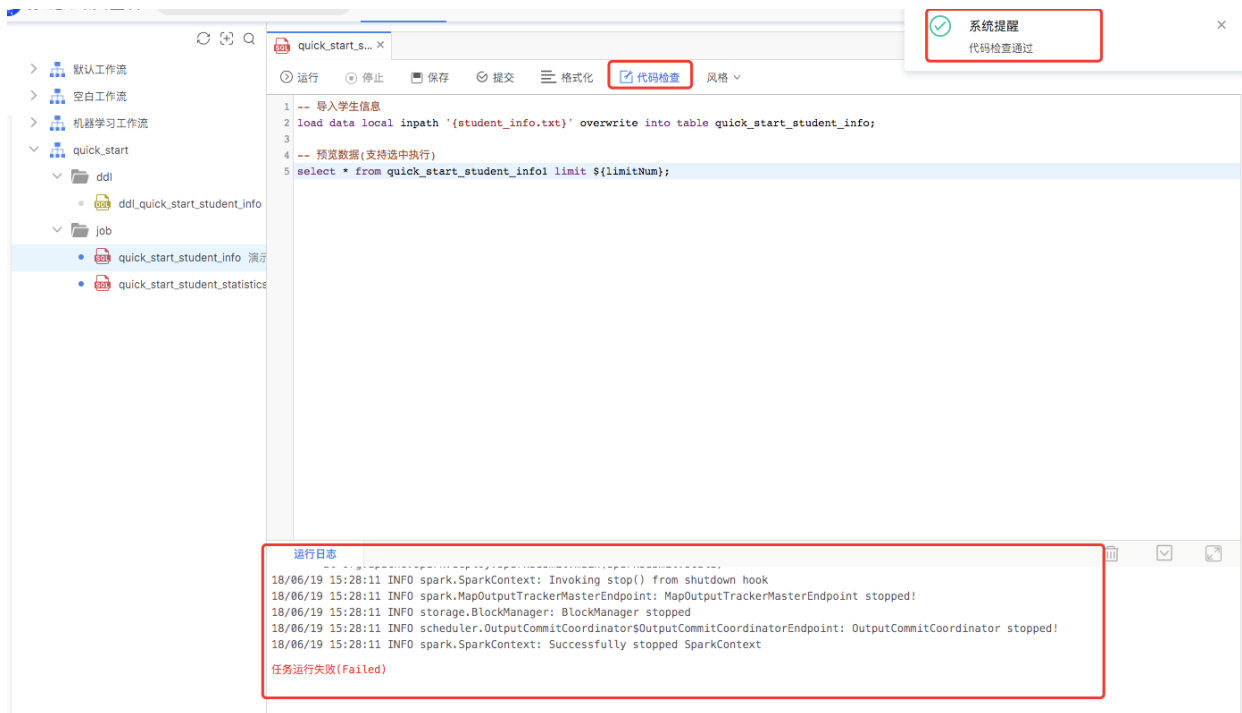
解决方案

资源名称在整个项目下唯一，这个子目录中虽然没有这个资源名，但这个项目下已经有了这个资源名的话也不能重复；建议通过左侧的搜索按钮搜索一下当前项目中是否已有同名的资源。

4 SQL代码检查通过了，运行却报错

问题描述

使用代码检查功能检查SQL代码，检查通过了，但是运行时却报错了。



解决方案

一个SQL的执行分为解析SQL生成语法树、查询计划、物理计划三个阶段。页面上的“代码检查”只是第一阶段。所以会出现语法检查通过，运行时出现异常的现象。

5 如何创建全局变量？

问题描述

运行参数只是针对一个任务的局部参数定义，如何定义全局变量？

目前只有一个系统参数，bizDate='20180220'(业务日期，即前一天的日期)，且不能修改。如果要定义其他参数呢？

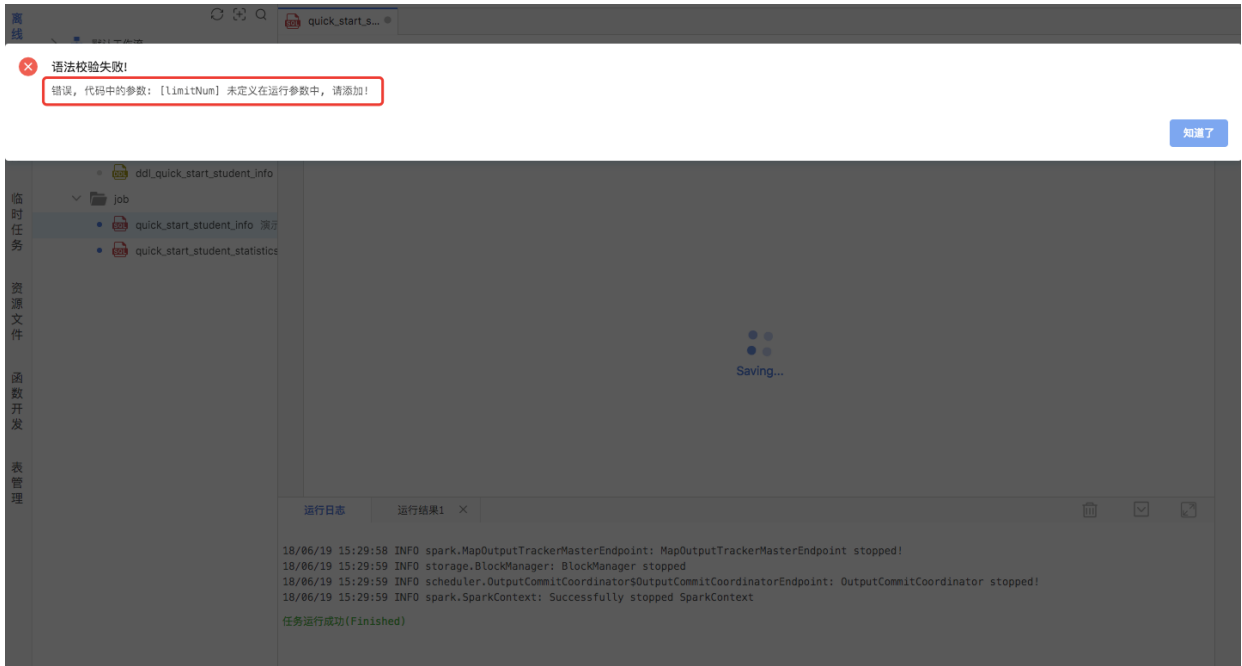
解决方案

目前不支持全局变量，因为一个项目下可能有上千个任务，多个人定义全局变量时容易出现变量名冲突或被覆盖问题。

6 代码中的参数未定义在运行参数中

问题描述

语法校验失败，代码中的参数: [xxx] 未定义在运行参数中，请添加！



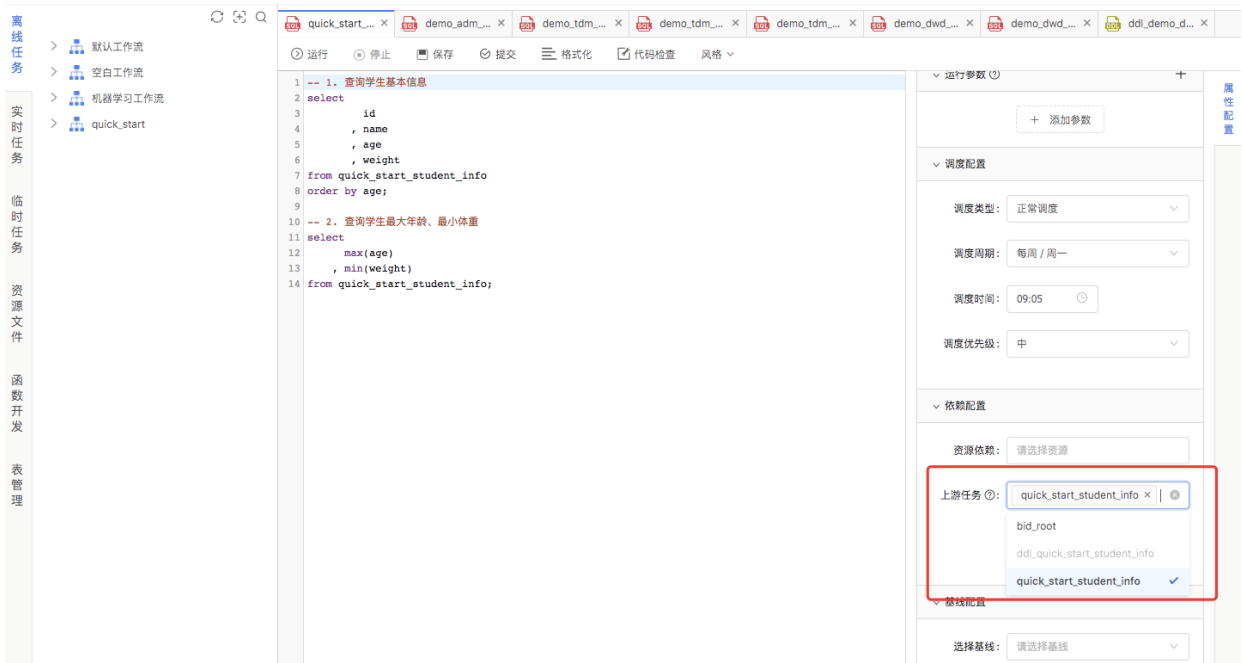
解决方案

所有任务中使用到的参数 `{变量名}`，必须在运行参数中预先定义。

7 什么任务可以被设为上游任务？

问题描述

属性配置--设置上游任务：有的任务置灰-不可选，有的任务不置灰-可选，什么任务可以被设为上游任务？



解决方案

根任务或根任务可达的某个任务可以被依赖，当可以被依赖时，在任务列表中小点图表显示蓝色。

离线任务

- 默认 workflow
- 空白 workflow
- 机器学习 workflow

实时任务

- quick_start
 - ddl
 - job
 - quick_start_student_info 演示
 - quick_start_student_statistics

临时任务

资源文件

```
1 -- 1. 查询学生基本信息
2 select
3     id
4     , name
5     , age
6     , weight
7 from quick_start_student_info
8 order by age;
9
10 -- 2. 查询学生最大年龄、最小体重
11 select
12     max(age)
13     , min(weight)
14 from quick_start_student_info;
```

蓝色表述已配置上游任务, 可被依赖

8 代码如何批量注释?

问题描述

开发套件中的任务代码, 如何注释?

解决方案

Mac系统批量注释快捷键为 `command + /`, Windows系统批量注释为 `ctrl + /`。