

阿里云 物联网数据分析服务 数据开发

文档版本：20190904

法律声明

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

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

通用约定

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

目录

法律声明.....	I
通用约定.....	I
1 数据开发简介.....	1
2 数据开发.....	2
2.1 开发任务.....	2
2.2 表管理和函数列表.....	7
2.3 使用示例.....	13
3 API服务.....	17
3.1 API服务开发.....	17
3.2 JAVA SDK调用示例.....	23
3.3 错误码.....	25

1 数据开发简介

数据开发提供云上交互式查询服务，无需数据预处理过程，直接使用标准的SQL语句对设备进行数据分析。

可以对以下三个数据来源中的设备数据进行分析：

- 物联网平台的系统数据
- 基于产品能力定义的设备数据
- 用户授权的业务数据

数据开发可以在设备监控、报表统计、设备行为分析等无需实时处理数据的场景皆可使用。数据从设备上报到物联网平台，完成SQL数据分析大概有5分钟左右的时延。如果需要使用实时数据分析，建议您使用流数据分析功能。

数据开发架构示意图如下所示：

2 数据开发

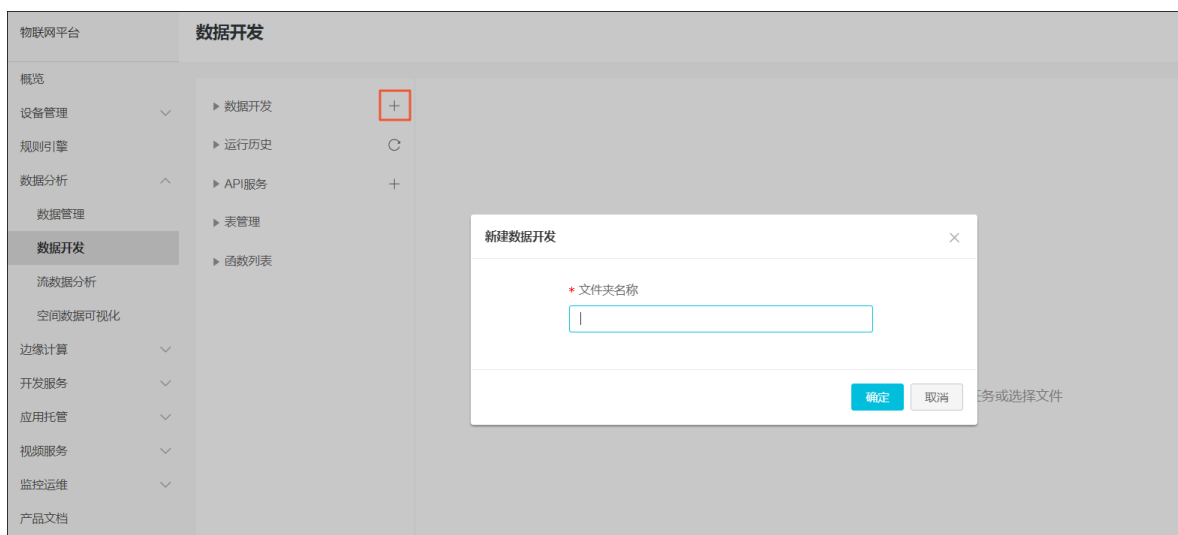
2.1 开发任务

数据开发任务是一个专门为物联网平台开发的SQL数据分析工具，开放物联网平台上的常用设备数据表，方便您进行自主数据分析，无需购买或部署额外的服务器。

操作步骤

1. 登录[物联网平台控制台](#)。
2. 左侧导航栏选择 数据分析 > 数据开发。
3. 在数据开发页面，单击数据开发后的“+”号图标新增数据开发文件夹。

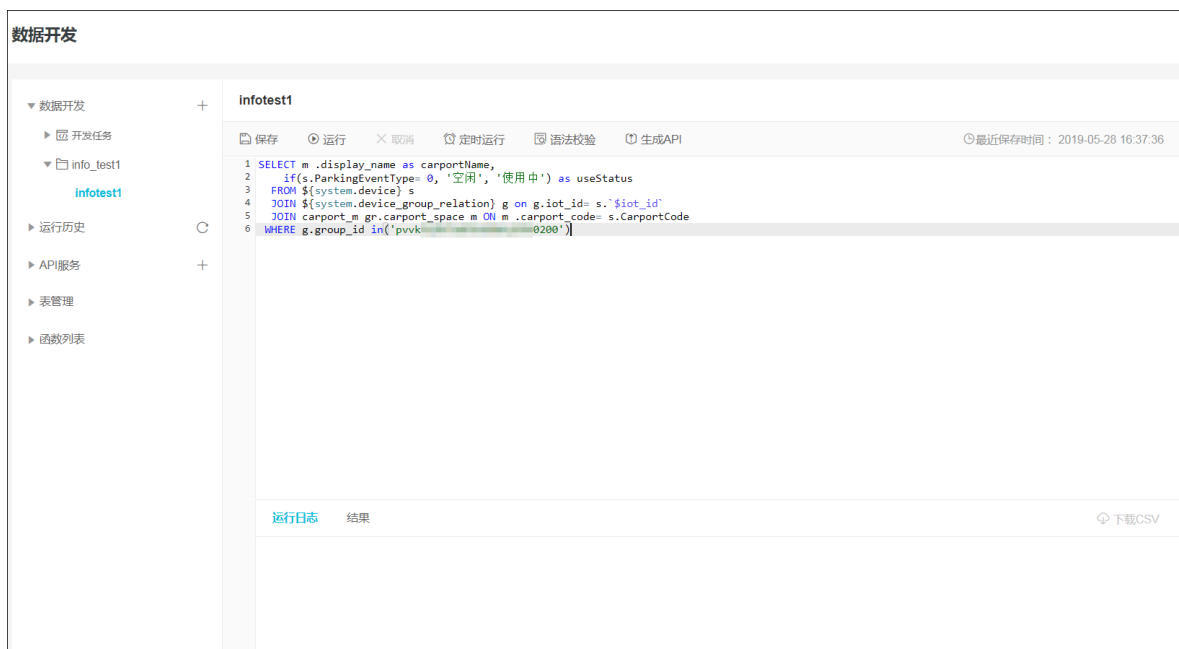
支持在数据开发目录下新建一级文件夹，文件夹用于保存开发任务，可以添加、删除和编辑。



4. 在已创建的文件夹名称后面，单击“+”号图标。
5. 在弹出窗口中输入新建任务的名称，并单击确定。

任务创建完成后，可修改任务名称、或删除任务。


6. 单击任务名称，打开任务SQL开发页面，编辑SQL语句。



编写SQL时，若需要使用系统数据表（平台系统表、平台设备数据表）或外部数据源（用户数据），单击表管理，从相应表目录下获取信息。系统数据表和用户数据表详细说明请参见#unique_6。



编写完成SQL语句后，可对SQL语句做如下操作：

操作	说明
保存	保存已编写的SQL语句。保存不检验语法错误，会直接保存内容。
运行	<p>运行已编写的SQL语句。</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p> 说明： 保证语句成功运行，建议您在运行语句前进行语法校验。</p> <ul style="list-style-type: none"> · 如果SQL语法正确并且存在查询结果时，结果信息将会在结果框中展示，其中字段名会展示在第一行。 · 如果SQL语法存在错误，则系统提示执行失败。 </div>
取消	取消已运行的SQL语句。此按钮在运行SQL语句后可生效。

操作	说明
定时运行	<p>定时执行SQL，运行离线任务。单击定时运行后在弹出窗口中打开定时开关。</p> <ul style="list-style-type: none"> · 生效时间：定时运行的生效时间区间。 · 调度周期：定时运行的调度周期，可以选择分钟、小时、日、周、月。 · 指定时间：指定定时运行具体在哪个日期运行。 · 具体时间：定时运行任务在指定时间的具体哪个时间段运行。
语法校验	对已编写的SQL语句进行语法校验。

执行SQL语句后：

- 可在运行日志页面查看SQL语句运行信息。

运行日志示例和每一行信息说明如下：

```

2019-04-30 17:59:23 -- startPoint:[] #执行参数信息
2019-04-30 17:59:23 -- updatePipelineToActive:success #Pipeline
激活状态
2019-04-30 17:59:23 -- checkWorkFlow:workflow is INTERACTIVE #检
查WorkFlow类型
2019-04-30 17:59:23 -- checkTransformer:transformer is 99d*****
724b1a5 #执行的TransformerUid
2019-04-30 17:59:23 -- dispatchToTaskExecutor:success #获取对应计
算平台的执行器
2019-04-30 17:59:23 -- addMaxLimit:success #补全分页限制
2019-04-30 17:59:23 -- callType:sync #同步执行/异步执行
2019-04-30 17:59:25 -- updateWorkflowToStandBy:success #成功后更
新WorkFlow状态

```

- 若执行成功，可在结果页面查看语句执行结果。结果框中最多返回200条查询结果，如果您有更多数据查询需求，请单击右侧下载CSV，下载任务的1000条查询结果。



说明：

如果您有更多的数据查询需求（超过1000条查询结果），请提交工单申请需求。

- 若执行失败，可在错误日志页面查看错误信息。

错误日志示例和每一行信息说明如下：

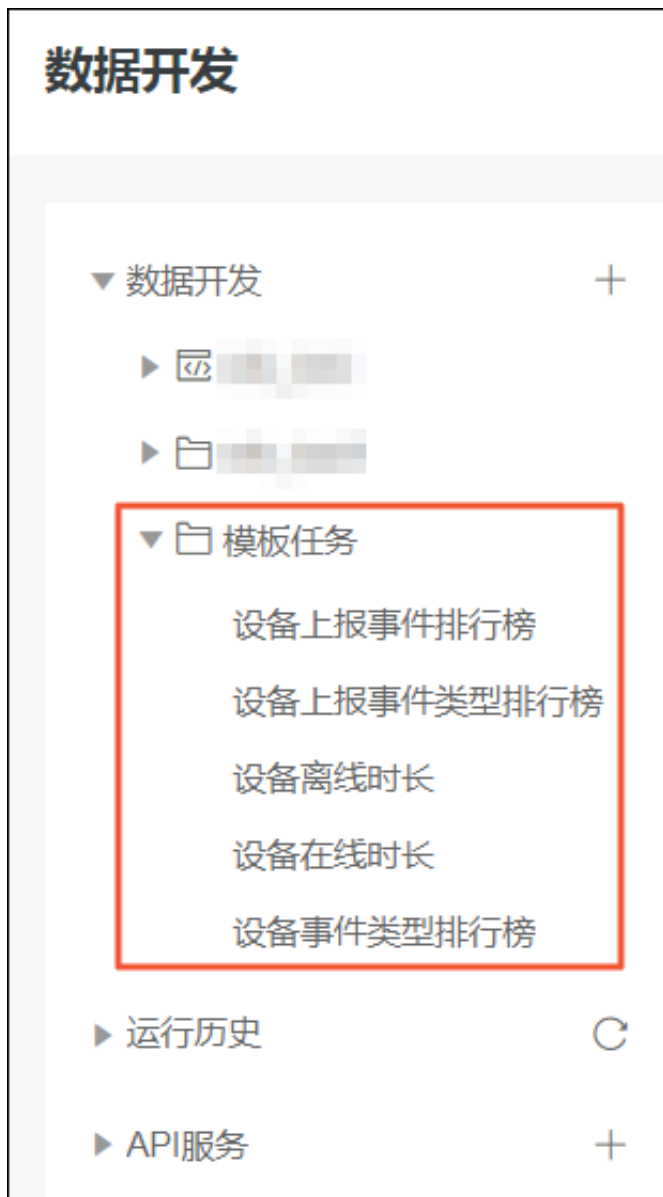
```

2019-04-30 15:31:33 -- startPoint:[] #执行参数信息
2019-04-30 15:31:33 -- updatePipelineToActive:success #Pipeline激
活状态
2019-04-30 15:31:33 -- checkWorkFlow:workflow is INTERACTIVE #检查
WorkFlow类型
2019-04-30 15:31:33 -- checkTransformer:transformer is 426*****
5286502 #执行的TransformerUid
2019-04-30 15:31:33 -- dispatchToTaskExecutor:success #获取对应计算
平台的执行器
2019-04-30 15:31:33 -- addMaxLimit:success #补全分页限制
2019-04-30 15:31:33 -- callType:sync #同步执行/异步执行

```

```
2019-04-30 15:31:33 -- taskExecuted:engine error, traceId 644*****  
03d05dd #提示执行错误并输出了此次调用的traceId  
2019-04-30 15:31:33 -- taskExecuted:INSERT statement has mismatched  
column types #具体错误原因
```

数据开发 > 模板任务下的任务是实现#unique_7中各类数据分析结果展示的模板SQL任务，关于SQL的编写也可参考此处的模板任务。



2.2 表管理和函数列表

表管理主要对系统数据表（平台系统表、平台设备数据表）和其他数据源中的用户数据进行管理；函数列表列出的是物联网数据分析支持的函数以及函数说明。

表管理

数据开发使用标准的SQL语法，但平台系统表名、平台设备数据表名等存在特殊性，保留以\${}命名的规则，因此通过表管理管理这些数据。

在数据分析 > 数据开发页面，打开表管理下的各类表内容，鼠标悬停在具体表名上可以显示“ ”图标，单击图标可直接复制表名，方便您使用表名。

平台系统表

平台系统表是开发者在物联网平台创建的产品、设备、设备分组等的数据库。

平台系统表包含设备分组关系表、产品表、设备表和设备分组表。每个表有如下两个页面可查看信息：

- 在表结构页面可查看表的字段说明。
- 在数据概览页面可查看实际的产品、设备、设备分组等的数据库。

平台系统表中，各个表的字段说明如下所示：

- 产品表

字段名称	字段类型	说明
access_method	Integer	接入方式。
data_format	Bigint	数据格式。
description	Varchar	产品描述。
gmt_create	Timestamp	产品创建时间。
gmt_modified	Timestamp	修改产品信息时间。
item_name	Varchar	产品正式名称。
name	Varchar	产品名称。
node_type	Integer	节点类型。 - 0: DEVICE - 1: GATEWAY
product_key	Varchar	产品唯一标识符ProductKey。

字段名称	字段类型	说明
status	Bigint	产品状态。 - 0: 开发中 - 1: 已发布

产品表信息主要对应控制台产品管理页面数据和产品详情页面数据，即您在物联网平台创建的产品相关信息。

· 设备表

字段名称	字段类型	说明
active_time	Timestamp	设备激活时间。
firmware_version	Varchar	固件版本号。
gmt_create	Timestamp	设备创建时间。
gmt_modified	Timestamp	修改设备信息时间。
iot_id	Varchar	设备唯一标识符。
name	Varchar	设备名称。
net_address	Varchar	设备IP地址。
nickname	Varchar	昵称。
product_key	Varchar	设备所属产品Productkey。
status	Tinyint	设备状态。 - 0: 未激活 - 1: 在线 - 3: 离线 - 8: 已禁止

设备表信息主要对应控制台设备管理页面数据和设备详情页面数据，即您在物联网平台创建的设备相关信息。

· 设备分组表

字段名称	字段类型	说明
description	Varchar	设备分组描述。

字段名称	字段类型	说明
gmt_create	Timestamp	设备分组创建时间。
gmt_modified	Timestamp	设备分组信息修改时间。
group_id	Varchar	设备分组ID。
group_type	Varchar	设备分组类型。
name	Varchar	设备分组名称。
status	Smallint	设备分组状态。

设备分组表信息主要对应控制台分组管理页面数据和分组详情 > 分组信息页面数据，即您在物联网平台创建的设备分组相关信息。

· 设备分组关系表

字段名称	字段类型	说明
device_name	Varchar	设备名称。
gmt_create	Timestamp	设备创建时间。
gmt_modified	Timestamp	设备信息修改时间。
group_id	Varchar	设备分组ID。
group_type	Varchar	设备分组类型。
iot_id	Varchar	设备唯一标识符。
product_key	Varchar	设备所属产品Productkey。
status	Varchar	设备分组状态。

设备分组关系表信息主要对应控制台分组详情 > 设备列表页面数据和分组详情 > 子分组列表页面数据，即您在物联网平台创建的设备分组相关信息。

平台设备数据表

平台设备数据表包含设备数据表和设备快照表。每个表有如下两个页面可查看信息：

- 在表结构页面可查看表的字段说明。
- 在数据概览页面可查看实际的产品、设备、设备分组等的的数据。

表说明如下所示：

· 设备数据表

设备产生的历史运行数据。物联网平台默认为每个产品免费赠送一个月数据存储，因此设备数据表保存了每个设备的最新（一个月）运行数据。



说明:

如果您在数据分析 > 数据管理 > IoT数据存储中对产品创建了更长久的数据存储周期，假设6个月，那么可以查询设备近6个月的历史数据。

· 设备快照表

设备快照表存储。在数据概览页面，可查看物联网平台创建的某一个产品下所有设备的运行信息。



说明:

如果产品有多个属性，而在某个时间只上报了一个属性，数据概览页面只显示最新上报的一个属性，其他属性会以历史数据方式存储。

数据开发

数据开发			
		表结构	数据概览
▶ 数据开发	+	概览 默认查询展示最新的数据。	
▶ 运行历史	○		
▶ API服务	+		
▼ 表管理			
▶ 平台系统表			
▼ 平台设备数据表	○		
▶ 设备数据表			
▼ 设备快照表			
Weather			
▶ 设备事件表			
▶ 用户数据			
▶ 函数列表			

sdevice_name	sproduct_key	slot_id	stype
weather1			DEVICE

用户数据

用户数据记录的是物联网平台中数据分析 > 数据管理模块下其他数据源的数据信息。您可以配置业务数据库账号，用于数据开发时进行跨域数据分析。



说明:

其他数据源目前支持的数据库类型为RDS for MySQL，暂不支持自建数据库，后续将会支持多种类型数据库云产品。

函数列表

函数列表列出的是物联网数据分析支持的函数以及函数说明。开发任务中需要使用函数，可通过函数列表查看函数信息。

单击函数名称，可查看该函数的说明：

数据开发

- ▶ 数据开发 +
- ▶ 运行历史 C
- ▶ API服务 +
- ▶ 表管理
- ▼ 函数列表
 - ▼ IP地址解析函数
 - ip2long**
 - ip2region
 - ip_country/ip_provin...
 - long2ip
 - ▶ 二进制
 - ▶ 位运算函数

函数说明

ip2long(ip)

函数说明：把IP地址转换成一个数字
返回值类型：BIGINT

示例：

```
select ip2long('115.239.210.27')
```

+-----+
1945096731

2.3 使用示例

本文以一个停车场智能化改造为例，描述如何使用数据开发实现业务场景。

背景信息

停车场智能化改造最关注的三个问题是停车位使用状态、停车场剩余可用车位情况以及当日停车场收入计算。

假设停车场使用地磁感应器来采集停车位状态信息，那么可以通过物联网平台创建模拟地磁感应器产品，并定义好产品功能，结合物联网数据分析服务中的数据开发功能，可以对停车场现状、排队数据、和收入进行分析计算。

下文将详细描述使用数据开发模块实现停车场智能化地操作。

操作步骤

1. 以阿里云账号登录[物联网控制台](#)。
2. 参考[#unique_10](#)，创建地磁感应器产品。

其中，部分参数必须按如下说明设置。

参数	描述
所属分类	选择智能城市 > 公共服务 > 地磁检测器。
节点类型	选择设备。

3. 参考[#unique_11](#)，在产品详情页，为刚创建的地磁感应器产品添加物模型。

- 添加事件类型属性

其中，取值范围中的0代表车位空闲，1表示车位使用中。

- 添加感应器编码属性

埋藏在停车位下的地磁感应器，在汽车停入和驶离时会发送停车位事件（ParkingEventType）和当前地磁感应器的编码（CarportCode），这样地磁感应器的状态就可以反映车位的状态。

4. 参考[#unique_12](#)，为地磁感应器产品添加设备。

5. 参考[#unique_13](#)，创建一个设备分组，并将上一步创建的地磁感应器设备添加到该分组中。

6. 参考#unique_14, 创建一个文件夹, 并在该文件夹下创建3个数据开发任务。

分别在3个数据开发任务中, 输入如下SQL语句, 校验语法并保存执行。

- 查询车位使用状态SQL如下:

```
-- 查询停车场使用状态
SELECT m.display_name as carportName,
       if(s.ParkingEventType= 0, '空闲', '使用中') as useStatus
FROM   ${snapshot.your_productkey} s -- 关联设备快照数据表, 它保存了地
磁感应器属性的最新数据
JOIN   ${system.device_group_relation} g on g.iot_id= s.`$iot_id`
-- 关联平台系统表, 设备分组表
JOIN   carport_mgr.carport_space m ON m.carport_code= s.CarportCode
-- 关联业务表, 车位可读名
WHERE  g.group_id in('your_device_group_id') -- your_device_group_id为该停车场设备分组id
```

其中,

- 将`${snapshot.your_productkey}`替换为地磁感应器设备 (sensor_1) 的设备快照数据表名。
- 将`your_device_group_id`替换为已创建的设备分组 (test_device_group) ID。
- 查询停车场使用率SQL如下:

```
-- 查询停车场使用率
CREATE VIEW stat_used_carport as
SELECT COUNT(*) as used_count
FROM   ${snapshot.your_productkey} r JOIN ${system.device_group_relation} gr ON
gr.iot_id= r.`$iot_id`
WHERE  gr.group_id in('your_device_group_id')
AND    r.ParkingEventType is not null
AND    r.CarportCode is not null
AND    r.ParkingEventType = 1; -- 使用中
CREATE VIEW stat_total_carport as
SELECT COUNT(*) as total_count
FROM   ${snapshot.your_productkey} r JOIN ${system.device_group_relation} gr ON
gr.iot_id= r.`$iot_id`
WHERE  gr.group_id in('your_device_group_id')
AND    r.ParkingEventType is not null
AND    r.CarportCode is not null;
SELECT if(b.total_count=0, 0, (a.used_count / b.total_count)) as
usage_ratio
```

```
FROM stat_used_carport a, stat_total_carport b;
```

其中,

- 将`${snapshot.your_productkey}`替换为地磁感应器设备 (sensor_1) 的设备快照数据表名。
- 将`your_device_group_id`替换为已创建的设备分组 (test_device_group) ID。
- 查询停车场收入SQL如下:

```
-- 查询停车场收费
create view no_used_carport as
SELECT t.CarportCode as carportCode,
       t.`$event_time` as eventTime,
       t.ParkingEventType as eventType,
       t.`$iot_id`
FROM ${pk.your_productkey} t -- 设备数据表 (历史数据)
WHERE t.CarportCode is not null
      and t.ParkingEventType is not null
      and t.ParkingEventType = 0 -- 空闲
group by t.`$event_time`,
         eventType,
         carportCode,
         t.`$iot_id`
order by t.`$event_time` desc;

create view used_carport as
SELECT t.CarportCode as carportCode,
       t.`$event_time` as eventTime,
       t.ParkingEventType as eventType,
       t.`$iot_id`
FROM ${pk.your_productkey} t -- 设备数据表 (历史数据)
WHERE t.CarportCode is not null
      and t.ParkingEventType is not null
      and t.ParkingEventType = 1 -- 使用中
group by t.`$event_time`,
         eventType,
         arportCode,
         t.`$iot_id`
order by t.`$event_time` desc;

create view used_time as
SELECT no_used_carport.eventTime - used_carport.eventTime as
duration
FROM no_used_carport INNER JOIN used_carport
on no_used_carport.carportCode = used_carport.carportCode
WHERE no_used_carport.eventType = used_carport.eventType - 1;
create view total_used_time as
SELECT (sum (if(used_time.duration< 3*1000/60/60, 0, used_time.
duration)) /1000/60/60)
as duration_hour FROM used_time;

-- 停车费用计算
SELECT if(duration_hour= 0, 0, duration_hour*5) as fee FROM
total_used_time;
```

其中,

- 将`${pk.your_productkey}`替换为地磁感应器设备 (sensor_1) 的设备数据表名。

7. (可选) 在真实的地磁感应器上做开发前, 您可以通过物联网平台的在线调试功能, 模拟出设备信息, 可以让硬件开发与应用开发独立进行。

参考[虚拟设备调试](#), 选择地磁感应器产品和设备, 设置相应的ParkingEventType和CarportCode属性值, 单击推送, 可以模拟地磁感应器的信号。

3 API服务

3.1 API服务开发

API服务指可以将数据开发中的数据任务（使用SQL开发的数据任务）封装成API，方便开发者调用，既可以直接响应设备端请求，也可以用来做服务端数据对接。

生成API服务的方式有两种，通过数据任务生成API或在API服务目录下开发API服务。

方式一、生成API

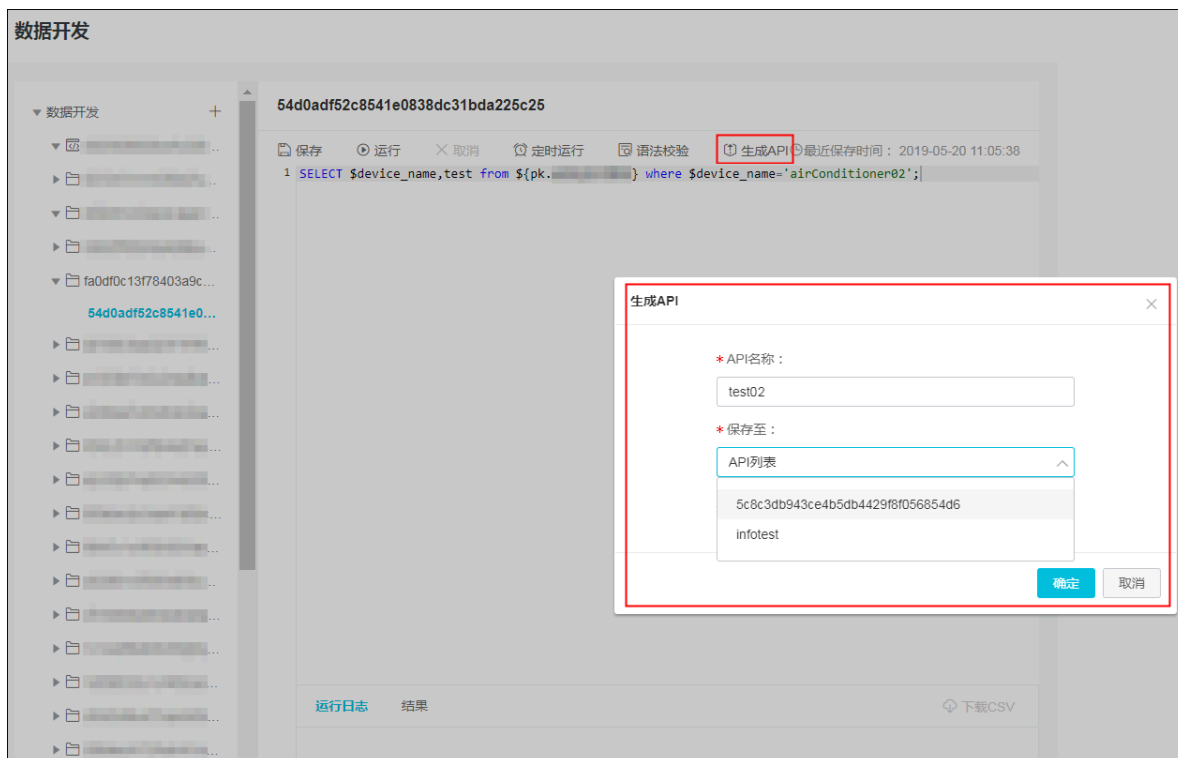
1. 登录[物联网平台控制台](#)。
2. 左侧导航栏选择 数据分析 > 数据开发 > 数据开发。
3. 选择已经运行通过的数据任务，单击上方操作栏中的生成API。



注意：

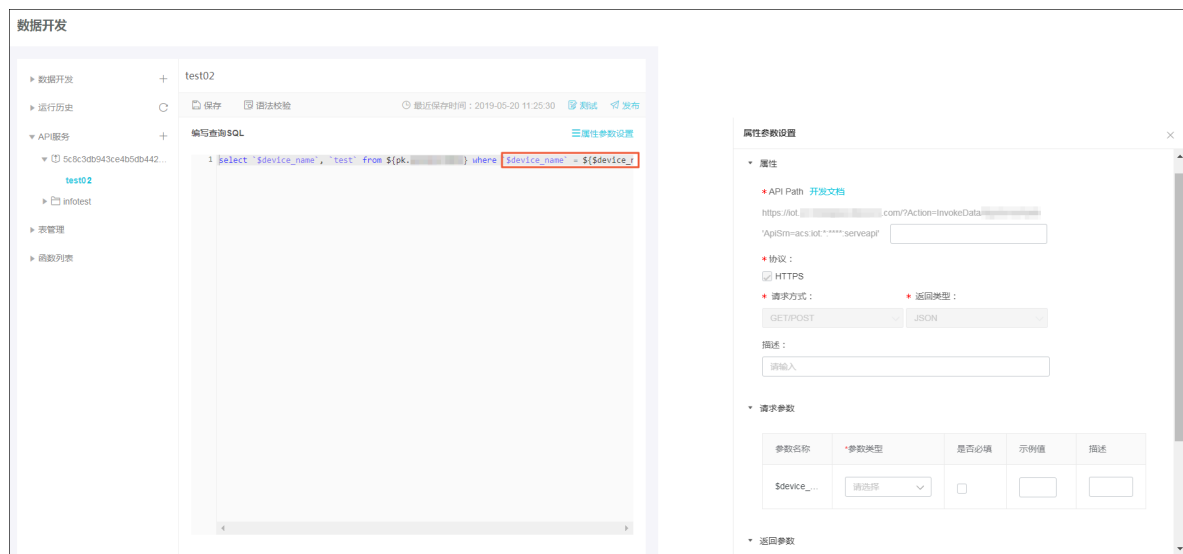
数据任务必须为运行成功任务。若无数据任务，请参考[#unique_17](#)编写SQL语句，完成数据任务的开发。

4. 在弹出页面中设置API名称，选择要保存API的目录（默认选项为API列表，也可以选择已有的自建目录）。



5. 单击确定，生成API。

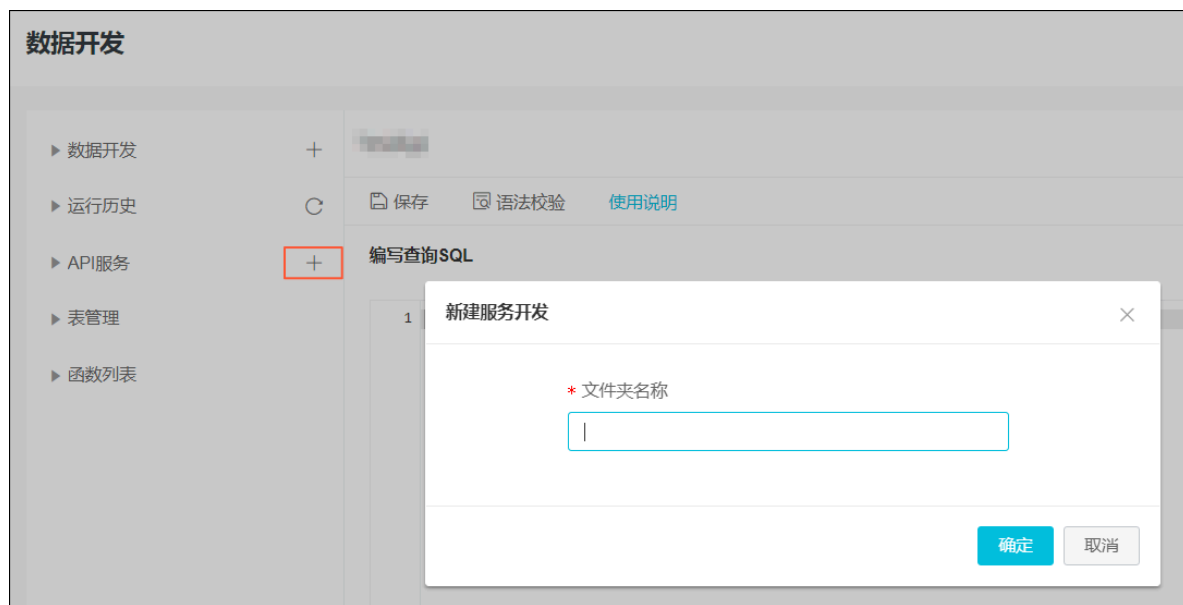
系统跳转至新创建的API任务，SQL已经进行转换，where语句后的查询条件变成参数形式，并默认展开属性参数设置面板。



方式二、API服务开发

1. 登录物联网平台控制台。
2. 左侧导航栏选择 数据分析 > 数据开发 > API服务。
3. 单击API服务后的“+”号图标新增服务开发文件夹。

支持在API服务目录下新建一级文件夹，文件夹用于保存开发服务，可以添加、删除和编辑。



4. 在已创建的文件夹名称后面，单击“+”号图标。



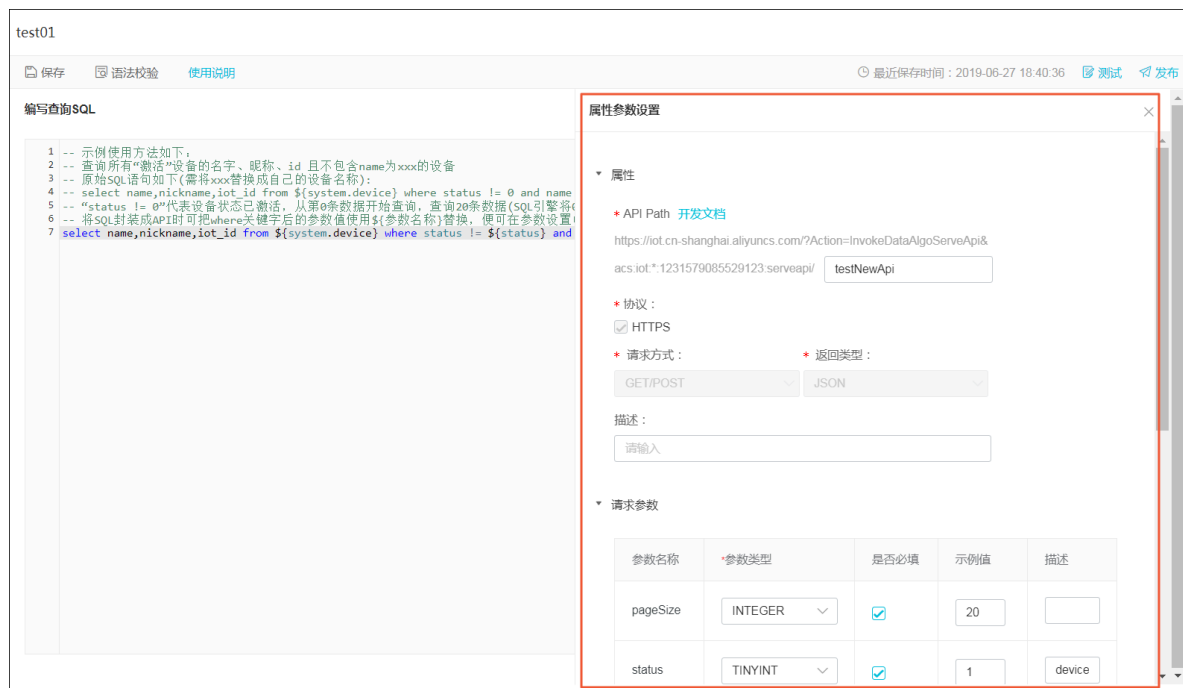
5. 在弹出框中设置参数，并单击确定。

参数	描述
API名称	设置API名称。仅支持中文、英文大小写、数字、下划线（_）、中划线（-）、英文括号（()）和空格，不能为空且长度不超过20个字符。
数据任务拷入	<p>可选参数。</p> <ul style="list-style-type: none"> · 可以选择已运行成功的数据任务，作为生成API的数据模板。 · 也可以不设置该参数，则系统使用默认的SQL模板。默认的SQL模板如下： <pre>select name,nickname,iot_id from \${system.device} where status != \${status} and name != \${name} limit \${pageNo} , \${pageSize}</pre>

创建API完成后，可修改API名称、或删除API。

6. 在编写查询SQL页面中，设置API相关参数。

您可以编写SQL，并根据SQL中设置的变量，单击右上角属性参数设置（新建API完成后属性参数设置页面会默认打开），设置API属性参数。



参数说明如下：

表 3-1: 属性参数配置

参数	是否必填	说明	示例
API Path	是	API路径，用于后续调用API服务的路径。API路径的前一段部分由系统生成，您只需要填写后一段路径。	testNewApi
描述	否	API描述。	无

表 3-2: 请求参数配置

参数	说明
参数类型	根据请求参数的类型，选择对应的参数类型。例如请求参数的类型为String，则选择VARCHAR。
是否必填	指调用API时，此参数是否必填。
示例值	参数值的示例，可以为空。

参数	说明
描述	描述请求参数，可以为空。

表 3-3: 返回参数配置

参数	说明
参数类型	根据返回参数的类型，选择对应的参数类型。
示例值	参数值的示例，可以为空。
描述	描述返回参数，可以为空。

例如，查询所有已激活设备的名称、昵称、ID，有如下两种方式修改模板SQL：

- 固定请求参数。例如从起始页开始查询，每页显示20条已激活设备的记录，则SQL语句如下：

```
select name,nickname,iot_id from ${system.device} where status != 0 limit 0 , 20;
```



说明：

status != 0表示设备状态为已激活。

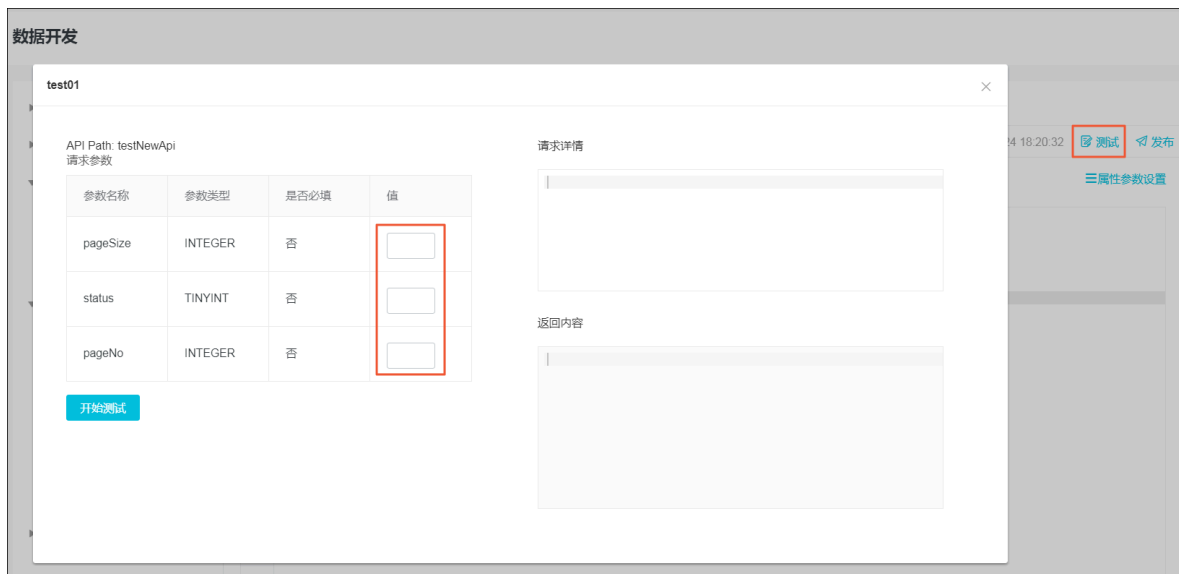
此时，在属性参数设置页面中，仅需设置属性和返回参数。

- 直接封装模板SQL，通过属性参数设置设置属性、请求参数和返回参数。

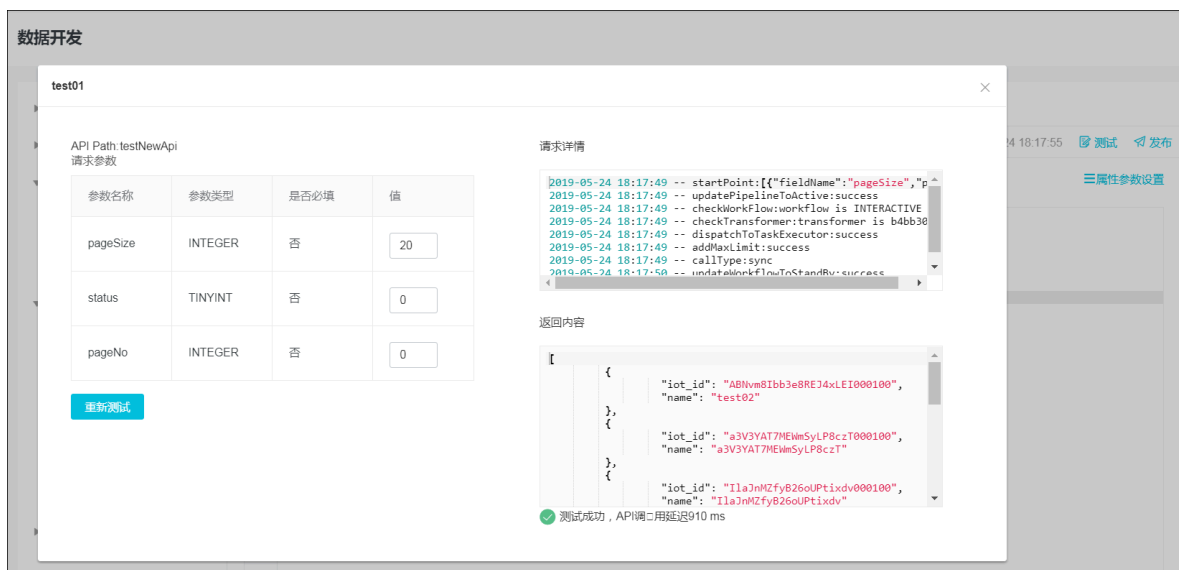
7. 配置参数完成后，单击保存，并进行语法校验且需要校验成功。

8. 单击服务开发页面右上角测试，进行API测试。

在API测试页面，若没有固定请求参数，则可以填写请求参数值，然后单击开始测试。



查看是否成功获取返回内容。



9. 测试成功后，在服务开发页面右上角单击发布，发布API服务。

发布成功后，可下载SDK，调用API服务。



说明:

API发布成功后，服务开发页面将不可编辑。

10. API发布成功后，参考#unique_18调用API。

3.2 JAVA SDK调用示例

本文描述如何通过Java SDK，调用物联网平台数据分析 > 数据开发中生成的API。物联网平台的Java SDK让开发人员可以方便地使用Java程序操作物联网平台。开发者可以使用Maven依赖添加SDK，也可以下载安装包到本地直接安装。

安装SDK

1. 安装Java开发环境。您可以从[Java官方网站](#)下载，并按说明安装Java开发环境。
2. 安装IoT Java SDK。
 - a. 访问[Apache Maven官网](#)，下载Maven软件。
 - b. 添加Maven项目依赖。IoT Java SDK的Maven依赖包坐标如下：

```
<!-- https://mvnrepository.com/artifact/com.aliyun/aliyun-java-sdk-iot -->
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-iot</artifactId>
  <version>6.10.0</version>
</dependency>
```

- c. 依赖公共包如下：

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.3.5</version>
</dependency>
```

初始化SDK

参考如下内容，初始化SDK。



说明：

以下示例以华东2地域及其服务接入地址为例。您在设置时，需使用您自己的物联网平台地域和对应的服务接入地址。

```
String accessKey = "<your accessKey>";
String accessSecret = "<your accessSecret>";
DefaultProfile.addEndpoint("cn-shanghai", "cn-shanghai", "Iot", "iot.cn-shanghai.aliyuncs.com");
IClientProfile profile = DefaultProfile.getProfile("cn-shanghai", accessKey, accessSecret);
```

```
DefaultAcsClient client = new DefaultAcsClient(profile); //初始化SDK客户端
```

其中, `<your accessKey>`为您账号的AccessKeyId, `<your accessSecret>`为AccessKeyId对应的AccessKeySecret。您可在阿里云官网控制台AccessKey管理中创建或查看您的AccessKey。

发起调用

以调用服务端订阅API接口, 查询数据结果为例。

```
String apiSrn = "your_api_srn";

InvokeDataAPIServiceRequest.Param param = new InvokeDataAPIServiceRequest.Param();
// 请求参数名称
param.setParamName("your_param_name");
// 在线状态
param.setParamValue("your_param_value");

InvokeDataAPIServiceRequest request = new InvokeDataAPIServiceRequest();
request.setApiSrn(apiSrn);
request.setParams(Arrays.asList(param));
// 当param为空时用请求方式用GET, 如果不为空是用POST
request.setParams(Arrays.asList(param));
request.setSysMethod(MethodType.POST);

try {
    InvokeDataAPIServiceResponse response = acsClient.getAcsResponse(request);

    System.out.println(response.getSuccess());
    System.out.println(response.getErrorMessage());

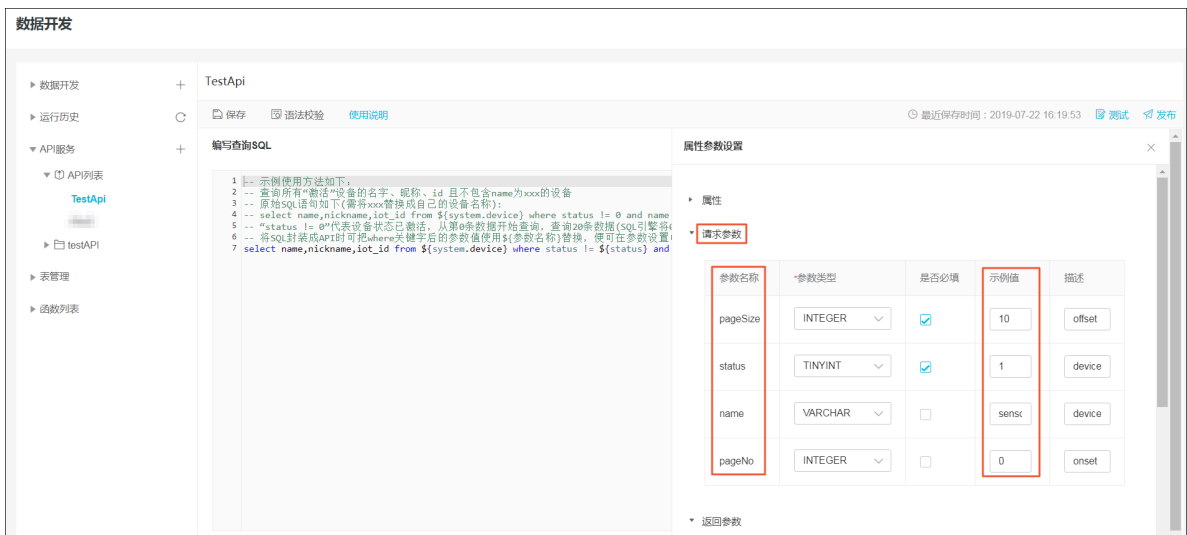
    // 服务API指定的SQL查询结果
    List<Map<Object, Object>> result = response.getData().getResultList();
    System.out.println(result);
} catch (ClientException ce) {
    ce.printStackTrace();
}
```

其中, 部分参数按如下说明替换参数值。

- your_api_srn: API的资源定位符, 请替换为您实际的API资源定位符。



- your_param_name: 待查询数据的参数名称, 请替换为您实际的API参数名称。



- your_param_value: 待查询数据的参数值, 请替换为您实际的API参数值。

3.3 错误码

本文列举调用物联网数据分析 > 数据开发 > API服务中的API出错时, 返回的错误信息。

入参数据格式错误、超出限定值、入参缺少必需参数等错误的修改方法, 请参见API服务文档的请求参数描述。

系统错误码


以iot.system开头的错误码为系统相关错误码。

错误码	描述
iot.system.SystemException	系统异常。 请稍后重试。

数据服务 (DataAlgoServeApi) 相关错误码

以*iot.prod*开头的错误码为数据服务相关错误码。

错误码	描述
<i>iot.dap.noServeJobExit</i>	数据服务API对应的任务不存在。 请查看对应的API任务是否正常运行。若异常，请修改或重新生成API。
<i>iot.dap.serveApiPathRepetition</i>	数据服务API接口地址重复。 请在属性参数设置中，更新API path的值。
<i>iot.dap.serveApiInvalidParam</i>	调用数据服务API的参数检查不通过。 请检查请求参数类型是否有误。
<i>iot.dap.serveApiPublishStatusError</i>	请先通过测试后，再发布任务。
<i>iot.dap.serveApiDeleteStatusError</i>	已发布的任务不可删除。
<i>iot.dap.serveApiPublishedNotEditable</i>	已发布的任务不可编辑。
<i>iot.dap.folderHasServeApiPublished</i>	存在已发布的数据服务API，不可删除文件夹。
<i>iot.dap.serveApiNotPublished</i>	数据服务API不在发布状态，无法回滚。
<i>iot.dap.duplicateTableNameError</i>	资源表名称重复。 请检查同一产品中属性或事件名称（不区分大小写）是否重复。
<i>iot.dap.serveApiAlreadyPublished</i>	数据服务API已发布。
<i>iot.dap.serveApiPathIsEmpty</i>	数据服务API接口地址不能为空。
<i>iot.dap.serveApiSqlTemplateError</i>	SQL模板信息异常。 请校验并更新后再尝试调用服务。
<i>iot.dap.serveApiSqlInvokeParamError</i>	SQL参数错误，参数类型与参数值不匹配。

错误码	描述
iot.dap.syncStartPipelineError	<p>任务启动失败。</p> <p>请根据异常信息修复后重新发起请求，若仍未处理问题可以提交工单反馈。</p> <div style="background-color: #f0f0f0; padding: 5px;"><p> 说明： 工单中请提供请求的requestId。</p></div>
iot.dap.methodTimeout	<p>接口调用超时。</p> <p>请修改SQL语句，减少查询数据量。例如缩小pageNo、pageSize等参数的参数值。</p>