

设备接入

最佳实践

文档版本 1.30
发布日期 2022-08-30



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

| | |
|--|------------|
| 1 概述..... | 1 |
| 2 设备接入..... | 6 |
| 2.1 在线开发 MQTT 协议的智慧路灯..... | 6 |
| 2.2 基于 NB-IoT 小熊派开发智慧路灯..... | 20 |
| 2.3 基于 NB-IoT 小熊派的智慧烟感..... | 38 |
| 2.4 设备模拟器快速接入..... | 57 |
| 2.5 通过协议转换网关实现泛协议设备接入..... | 65 |
| 2.6 恒温空调..... | 68 |
| 2.7 设备按需选网..... | 71 |
| 2.8 设备自定义 TOPIC 迁移上云..... | 78 |
| 2.9 模拟 NB 设备的接入与调试..... | 88 |
| 2.10 利用工业网关（包含 RS485、AI、DO 口设备）进行环境监测..... | 112 |
| 2.11 存量设备迁移最佳实践..... | 129 |
| 2.12 MQTT 协议设备 OTA 升级实践..... | 135 |
| 2.13 设备通过微信小程序接入到平台..... | 146 |
| 2.14 基于 MQTT.fx 的 X.509 证书接入指导..... | 157 |
| 2.15 基于 OpenHarmony3.0 对接 IoTDA..... | 176 |
| 2.16 设备级消息上下行保序方案..... | 185 |
| 2.17 通过设备发放实现批量设备自动注册..... | 186 |
| 2.18 使用自定义 Topic 进行通信..... | 198 |
| 2.19 基于 Jmeter 工具进行 MQTT 性能测试..... | 201 |
| 3 数据转发..... | 207 |
| 3.1 零代码搭建物联网监控大屏..... | 207 |
| 3.2 IoTDA 结合 DLI+DWS+Astro 实现大数据分析..... | 221 |
| 3.3 IoTDA 结合 ModelArts 实现预测分析..... | 233 |
| 4 设备联动..... | 246 |
| 4.1 设备触发告警并邮件或短信通知..... | 246 |
| 4.2 温度过高时自动关闭设备..... | 252 |
| 4.3 煤气浓度过高自动打开推窗器..... | 255 |
| 4.4 监测设备状态变化情况并发送通知..... | 261 |
| 5 设备发放..... | 270 |
| 5.1 设备厂商多实例接入方案..... | 270 |

| | |
|---------------------------------|-----|
| 5.2 结合函数服务通过自定义策略发放证书认证的设备..... | 277 |
| 5.3 共享实例存量设备同步..... | 288 |
| 5.4 设备自动注册安全接入示例..... | 291 |

1 概述

现在您应该对物联网平台有了初步的认识，那也许您会问，如何给您带来价值？哪些业务场景可以通过平台来实现？怎样接入到平台？接下来，通过最佳实践，我们把平台的业务流转串起来，把物模型等概念引进来，通过业务场景示例来呈现平台能力和价值，供您参考。

如果您想了解具体的业务场景描述，可以参阅[快速玩转平台](#)。

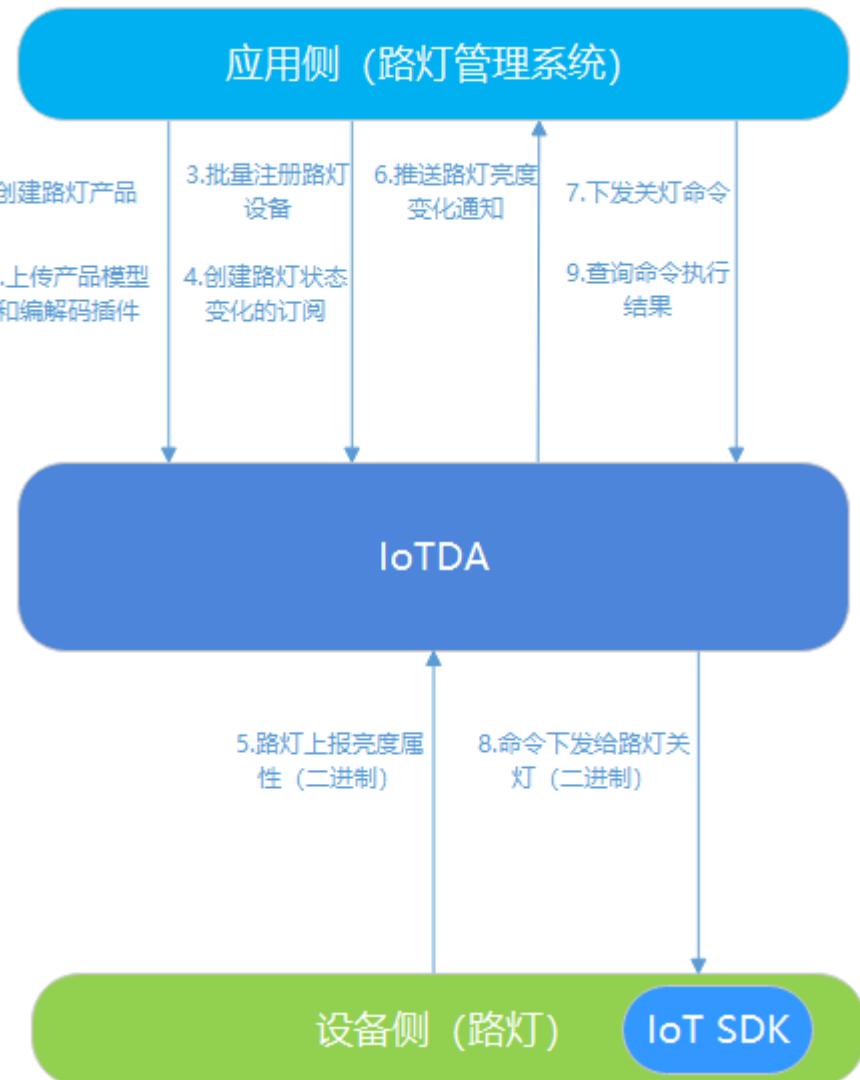
场景示例—智慧路灯

场景简介：路灯管理系统通过接入物联网平台，监控路灯状态，控制路灯开关（集成了NB模组的智慧路灯）。

该场景主要描述的是设备可以通过LWM2M协议与物联网平台进行交互，应用侧可以到物联网平台订阅设备侧变化的通知，也可以给设备侧下发命令。

核心知识点：产品模型、编解码插件、订阅推送、属性上报、命令下发。

最佳实践参考：[基于小熊派开发智慧路灯](#)。



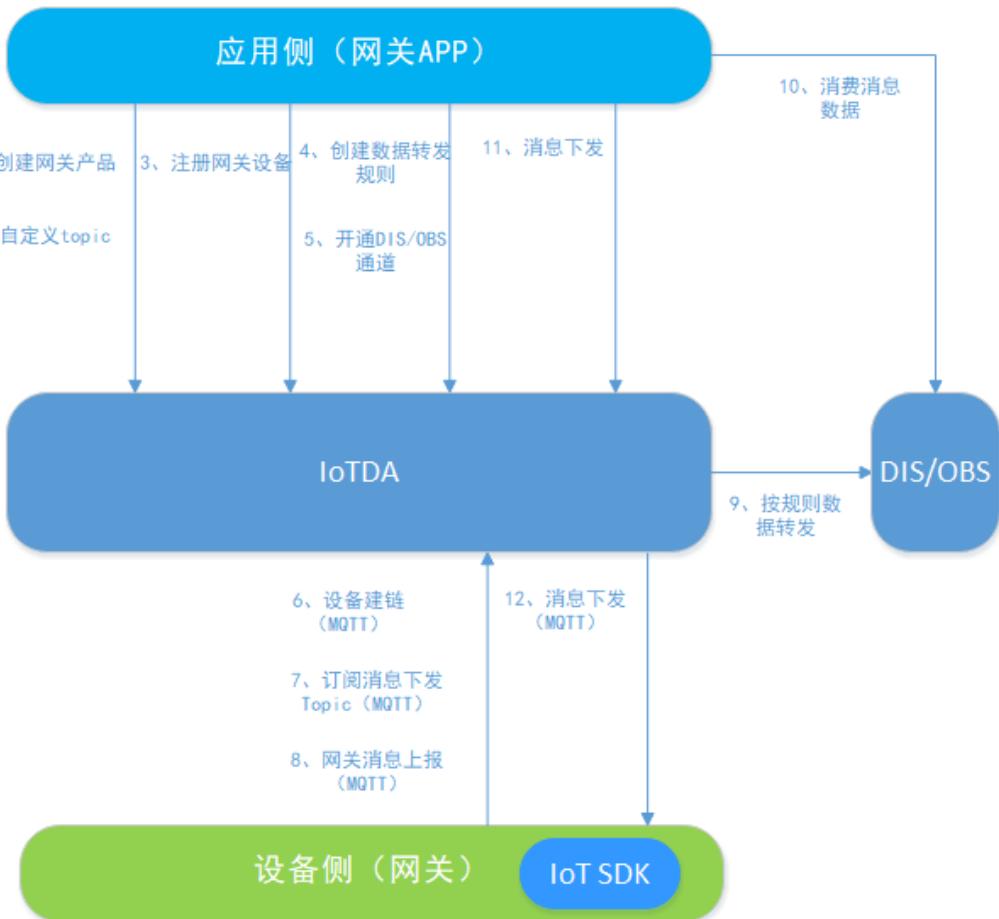
场景示例—智能网关

场景简介：已对接存量设备的网关，不用迁移就可以实现对既有设备和新设备的兼容控制。

该场景主要描述的是设备可以通过MQTT协议与物联网平台进行交互，用户可以在控制台产品详情中自定义Topic，通过应用侧接口或控制台创建数据转发规则，把设备上报的消息转发给其他华为云服务，供应用侧用户主动去消费。

核心知识点：产品模型、消息上报、消息下发、MQTT协议、数据转发规则、自定义Topic。

最佳实践参考：[使用自定义Topic进行通信](#)



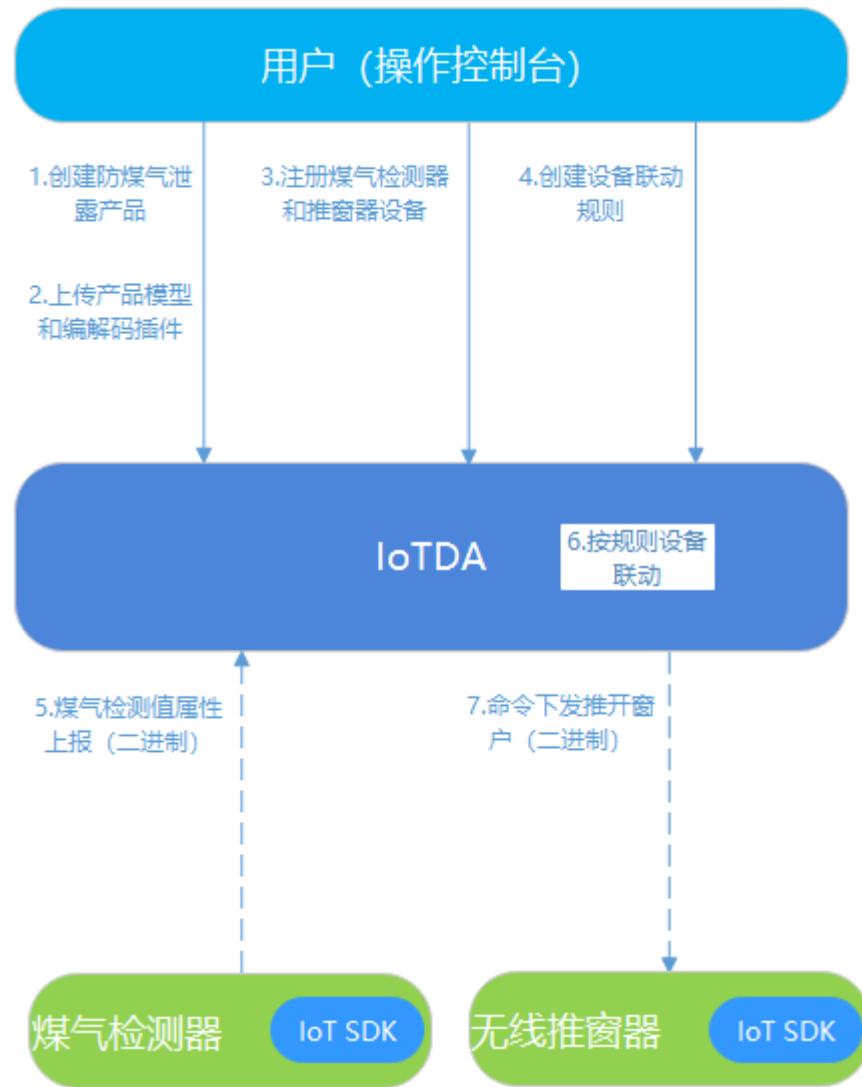
场景示例—智能家居煤气检测联动

场景简介：煤气检测器发现煤气浓度超标时，自动关联无线推窗器把窗户打开，起到通风的作用。

该场景主要描述的是设备可以通过MQTT协议与物联网平台进行交互，用户可以在控制台或通过应用侧接口创建设备联动规则，把设备上报的属性转发，通过物联网平台规则引擎转变成命令下发给其他指定设备。

核心知识点：产品模型、属性上报、命令下发、MQTT协议、设备联动规则。

最佳实践参考：[煤气浓度过高自动打开推窗器](#)。



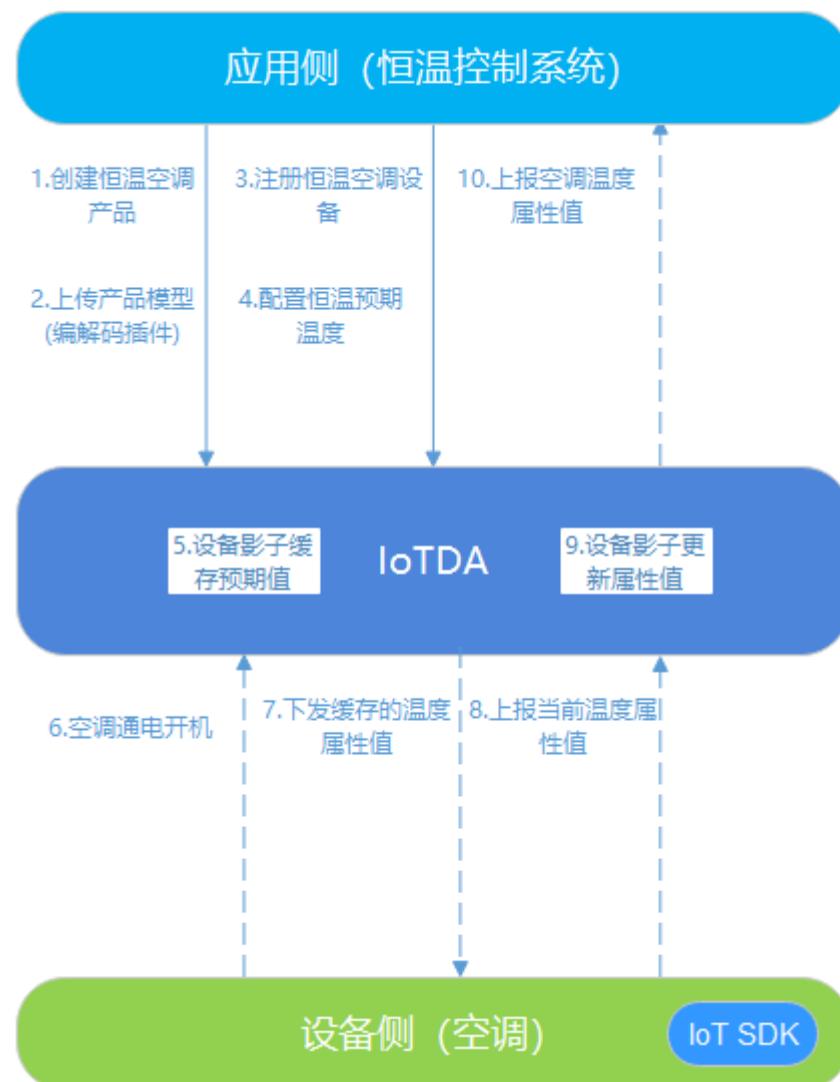
场景示例一恒温空调

场景简介：通过恒温控制系统，不论空调是否开机，都可以调整空调默认温度，待空调上电开机后，自动按默认温度调节。

该场景主要描述的是应用侧或控制台可以给离线设备下发缓存命令，待设备上线后执行，控制台通过设备影子配置属性，待设备上线后上报的属性不匹配的话，会自动下发属性修改的指令，直至修改成功。

核心知识点：产品模型、编辑码插件、设备影子、属性上报、属性修改。

最佳实践参考：[恒温空调](#)。



2 设备接入

2.1 在线开发 MQTT 协议的智慧路灯

场景说明

本文以“智慧路灯”为示例，通过MQTT.fx的设备模拟器替代真实的设备，带您快速体验设备上报数据到物联网平台和远程下发控制命令到设备的全过程。

假设：

路灯设备上报一条光照强度（luminance）消息；支持远程控制路灯开关状态的命令（switch），上报数据的格式为JSON格式。

前提条件

- 已注册华为云官方帐号。未注册可单击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击[实名认证](#)完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“立即使用”后开通该服务。

业务流程

基于MQTT.fx体验平台功能是指以在MQTT.fx模拟器为例，介绍数据上报、命令下发等业务。

- 步骤1** [创建产品](#)。创建一个MQTT协议的产品。
- 步骤2** [开发产品模型](#)。通过定义产品模型，在物联网平台构建一款路灯设备，支持上报光照强度、下发路灯开关状态命令。
- 步骤3** [注册设备](#)。注册一个MQTT协议的设备，体验数据上报业务。
- 步骤4** [连接鉴权](#)。使用MQTT.fx激活在物联网平台上注册的设备。
- 步骤5** [数据上报](#)。使用MQTT.fx向物联网平台上报数据。

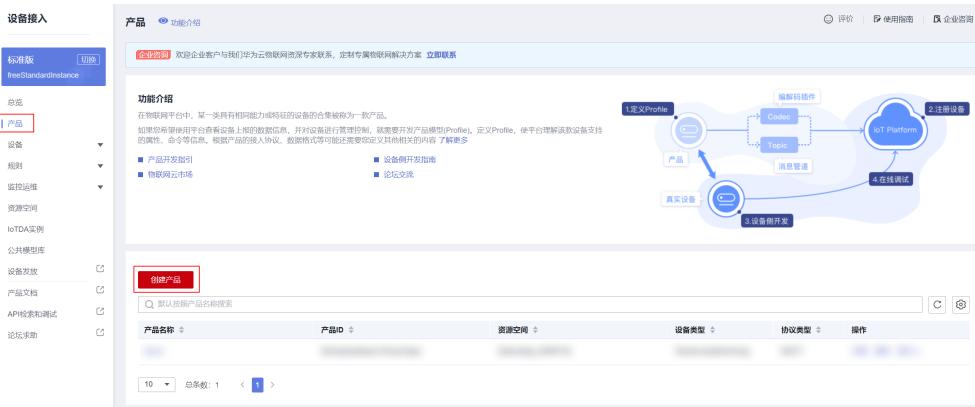
步骤6 命令下发。在管理控制台下发命令，远程控制设备。

----结束

创建产品

步骤1 登录管理控制台，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

图 2-1 产品-创建产品



步骤2 创建一个协议类型为MQTT协议、设备类型为StreetLamp的产品，参考页面提示填写参数后，单击“确定”。

图 2-2 创建产品-MQTT

创建产品

* 所属资源空间 (?)

如需创建新的资源空间, 您可前往当前实例详情创建

* 产品名称 Test_1

协议类型 (?) MQTT

* 数据格式 (?) JSON

设备类型选择 标准类型 自定义类型

* 设备类型 (?) StreetLamp

高级配置 ▲ 定制ProductID | 备注信息

产品ID (?)

产品描述

0/128

确定 取消

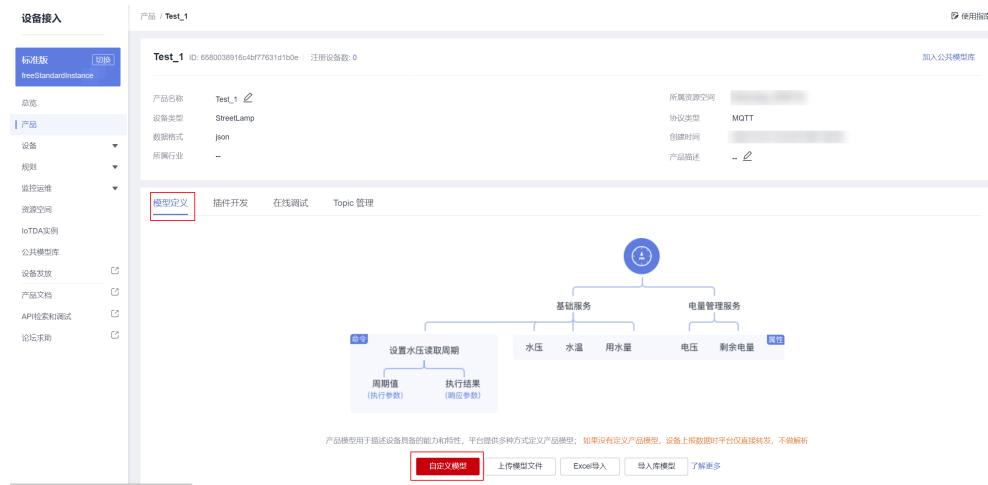
----结束

开发产品模型

步骤1 找到新增的产品，单击产品进入产品界面。

步骤2 在产品详情“模型定义”页面，单击“自定义模型”，配置产品的服务。

图 2-3 自定义模型-MQTT



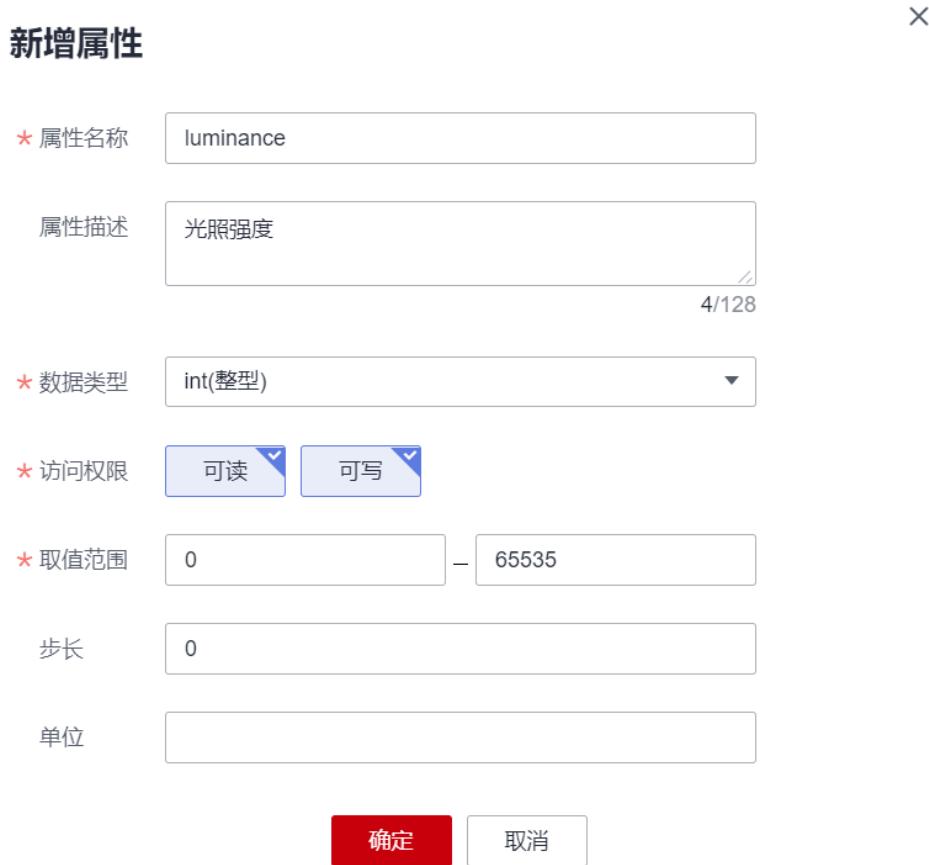
步骤3 新增服务类型“BasicData”。

1. 在“添加服务”页面，根据页面提示填写“服务ID”、“服务类型”和“服务描述”，单击“确定”。

图 2-4 添加服务-BasicData

2. 在“**BasicData**”服务列表右侧区域，单击“新增属性”，填写相关信息后，单击“确定”。

图 2-5 新增属性-luminance

**步骤4 新增服务类型“LightControl”。**

1. 在“模型定义”下单击“添加服务”，根据页面提示填写后，单击“确定”。
 - “服务ID”：LightControl
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：路灯开关控制
2. 在“LightControl”的下拉菜单下单击“添加命令”，输入命令名称“Switch”。

图 2-6 新增命令-Switch

新增命令

| | | | |
|------|--------|----|----|
| 命令名称 | Switch | | |
| 下发参数 | 新增输入参数 | | |
| 参数名称 | 数据类型 | 描述 | 操作 |

暂无表格数据

响应参数

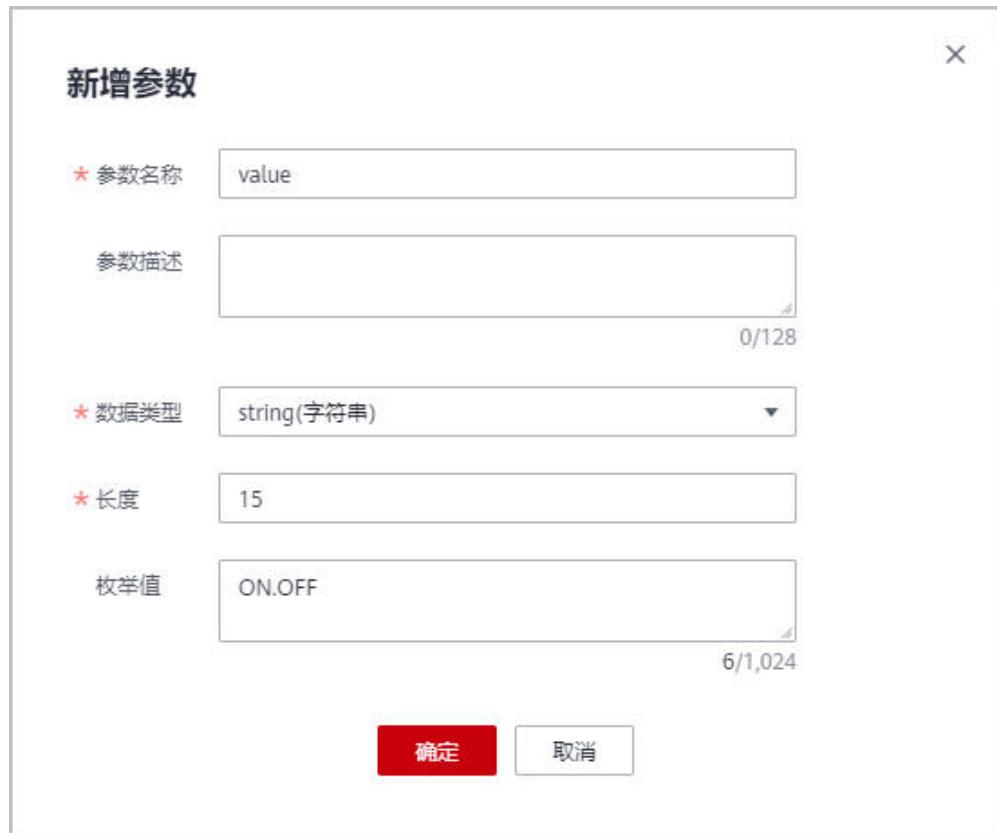
| | | | |
|--------|------|----|----|
| 新增响应参数 | | | |
| 参数名称 | 数据类型 | 描述 | 操作 |

暂无表格数据

确定 取消

3. 在“新增命令”页面，单击“新增输入参数”，填写相关信息后，单击“确定”。

图 2-7 新增命令参数-value



----结束

注册设备

步骤1 在设备接入控制台页面，选择左侧导航栏“设备 > 所有设备”，单击页面右上角的“注册设备”。

图 2-8 所有设备-注册设备

步骤2 根据页面提示信息填写参数，然后单击“确定”。

| 参数名称 | 说明 |
|--------|-----------------------------------|
| 所属资源空间 | 确保和所属产品归属在同一个资源空间。 |
| 所属产品 | 选择对应产品。 |
| 设备标识码 | 即nodeID，设备唯一物理标识。可自定义，由英文字母和数字组成。 |

| 参数名称 | 说明 |
|--------|--------------------|
| 设备名称 | 即device_name，可自定义。 |
| 设备认证类型 | 选择“密钥”。 |
| 密钥 | 此处如不填写，物联网平台会自动生成。 |

图 2-9 单设备注册-MQTT

单设备注册

* 所属资源空间 (?)

* 所属产品 Test_1
MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

* 设备标识码 (?) Test_1

设备名称

设备ID (?)

设备描述
0/2,048

设备认证类型 (?) 密钥 X.509证书

密钥

确认密钥

确定 取消

步骤3 成功注册设备后，平台会自动生成设备ID和密钥，请妥善保管好设备ID（deviceId）和密钥（deviceSecret），用于设备接入。

图 2-10 注册设备成功

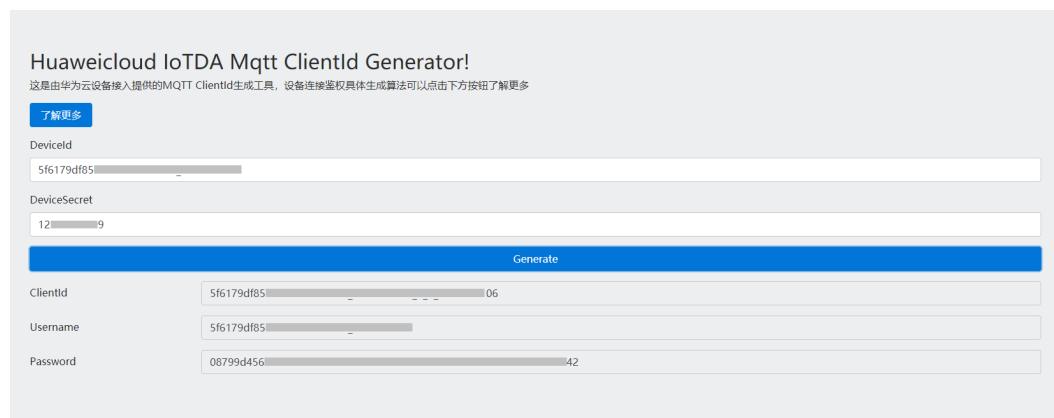


----结束

连接鉴权

使用MQTT.fx工具激活在物联网平台上注册的设备。

- 步骤1 下载[MQTT.fx](#)（默认是64位操作系统，如果是32位操作系统，单击此处下载[MQTT.fx](#)），安装MQTT.fx工具。
- 步骤2 访问[这里](#)，填写注册设备后生成的设备ID（DeviceId）和密钥（DeviceSecret），生成连接信息（ClientId、Username、Password）。



| 参数 | 必选/ 可选 | 类型 | 参数描述 |
|----------|-----------|-------------|--|
| ClientId | 必选 | String(256) | <p>一机一密的设备clientId由4个部分组成：设备ID、设备身份标识类型、密码签名类型、时间戳，通过下划线“_”分隔。</p> <ul style="list-style-type: none">• 设备ID：指设备在平台成功注册后生成的唯一设备标识，通常由设备的产品ID和设备的NodeId通过分隔符“_”拼装而来。• 设备身份标识类型：固定值为0，表示设备ID。• 密码签名类型：长度1字节，当前支持2种类型：<ul style="list-style-type: none">- “0”代表HMACSHA256不校验时间戳。- “1”代表HMACSHA256校验时间戳。• 时间戳：为设备连接平台时的UTC时间，格式为YYYYMMDDHH，如UTC时间2018/7/24 17:56:20则应表示为2018072417。 |
| Username | 必选 | String(256) | 设备ID。 |
| Password | 必选 | String(256) | 加密后的设备密钥。Password的值为使用“HMACSHA256”算法以时间戳为密钥，对secret进行加密后的值。 secret为注册设备时平台返回的secret。 |

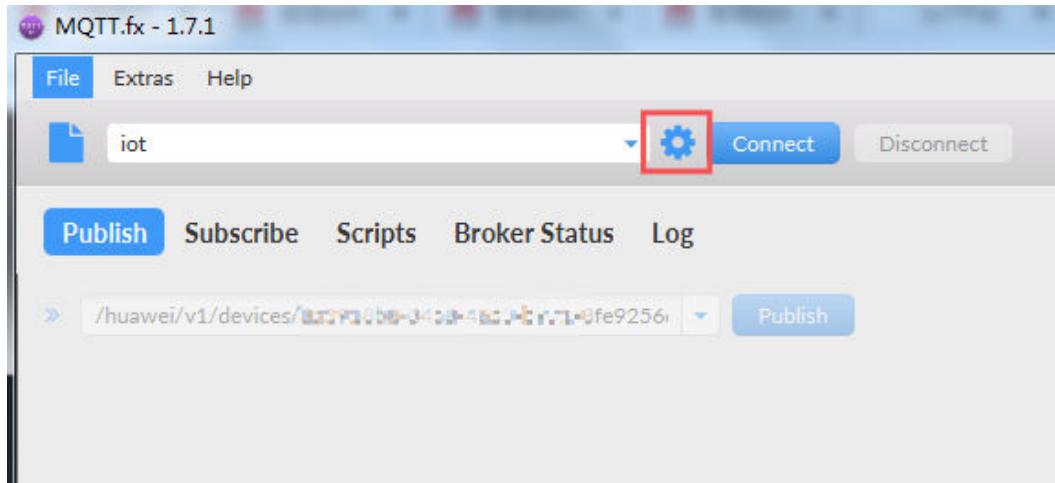
设备通过MQTT协议的connect消息进行鉴权，对于构造clientId的各个部分信息都必须包括进去，平台收到connect消息时，会判断设备的鉴权类型和密码摘要算法。

使用生成工具生成的clientId格式，默认不校验时间戳：设备ID_0_0_时间戳。

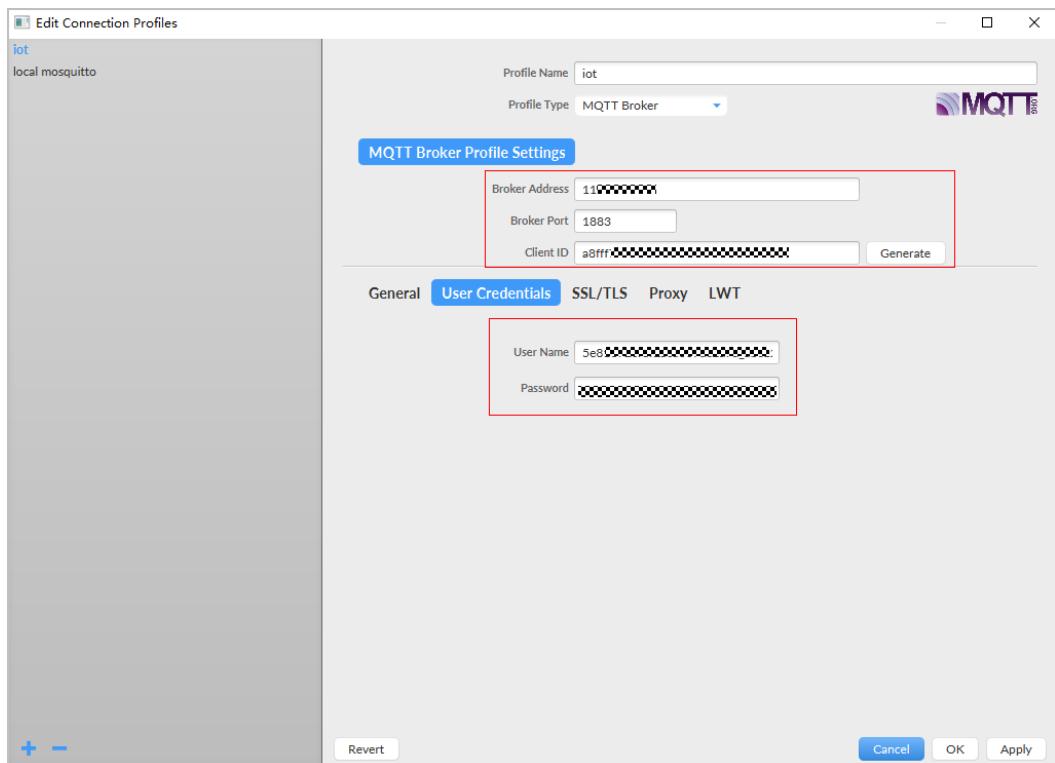
- 当采用“HMACSHA256”校验时间戳方式时，会先校验消息时间戳与平台时间是否一致，再判断密码是否正确。
- 当采用“HMACSHA256”不校验时间戳方式时，鉴权消息也必须带时间戳，但不检验时间是否准确，仅判断密码是否正确。

connect消息鉴权失败时，平台会返回错误，并自动断开MQTT链路。

步骤3 打开MQTT.fx软件，单击设置图标。



步骤4 参考下表配置鉴权参数，然后单击“Apply”。



| 参数名称 | 说明 |
|----------------|--|
| Broker Address | 填写从设备接入服务控制台获取的 平台对接信息 ，此接入地址为域名信息。不能通过域名接入的设备，通过在cmd命令框中执行“ping 域名”获取IP地址，用IP地址接入平台。由于IP地址不固定，您需要将IP地址做成可配置项。 |
| Broker Port | 默认为1883。 |
| Cliend ID | 设备clientID，请参考 2 中获取。 |
| User Name | 即设备ID，请参考 2 中获取。 |

| 参数名称 | 说明 |
|----------|-------------------------------------|
| Password | 加密后的设备密钥，请参考 2中获取 。 |

步骤5 单击“Connect”，设备鉴权成功后，在物联网平台可以看到设备处于在线状态。

图 2-11 设备在线

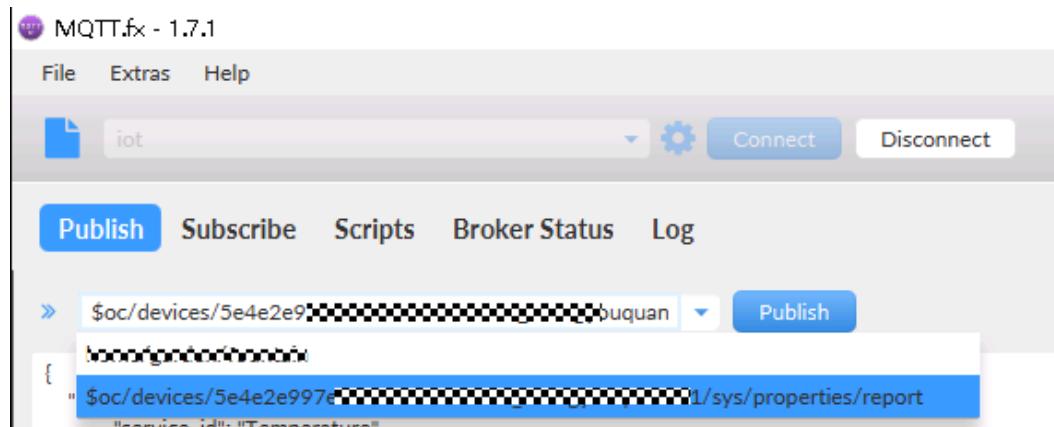


----结束

数据上报

使用MQTT.fx工具向物联网平台上报数据。设备若通过MQTT通道上报数据，需要发给指定的topic，上报消息的topic格式为：\$oc/devices/{device_id}/sys/properties/report，其中对于一机一密设备，使用deviceID接入时需填写为设备注册成功后返回的deviceID值。

步骤1 填写接口地址，此处以\$oc/devices/{device_id}/sys/properties/report为例，如：\$oc/devices/5e4e2e92ac-164aefaa8fouquan1/sys/properties/report。



步骤2 在工具中间的空白处填写上报的数据后，单击“Publish”。

表 2-1 设备服务数据列表

| 字段名 | 必选/可选 | 类型 | 参数描述 |
|----------|-------|-----------------------|--------------------------------------|
| services | 必选 | List<ServiceProperty> | 设备服务数据列表（具体结构参考下表ServiceProperty定义表） |

表 2-2 ServiceProperty 定义表

| 字段名 | 必选/可选 | 类型 | 参数描述 |
|------------|-------|--------|---|
| service_id | 必选 | String | 设备服务的ID。 |
| properties | 必选 | Object | 设备服务的属性列表，具体字段在设备关联的产品模型中定义。 |
| eventTime | 可选 | String | 设备采集数据UTC时间（格式：yyyyMMddTHHmmssZ），如：20161219T114920Z。 设备上报数据不带该参数或参数格式错误时，则数据上报时间以平台时间为准。 |

请求示例

```
{  
    "services": [{  
        "service_id": "BasicData",  
        "properties": {  
            "luminance": 30  
        }  
    }  
}]
```

步骤3 可以在设备详情页中查看设备是否成功上报数据。

图 2-12 查看上报数据-MQTT

The screenshot shows the 'Device Management / Device Details' interface. The top navigation bar includes links for 'Device Information', 'Cloud Operation Log', 'Cloud Download', 'Device Shadow', 'Message Trace', 'Device Monitoring', 'Sub-device', 'Tag', and 'Group'. The main content area displays device details for '所属产品 Test_1'. On the left, there's a sidebar with sections like '所需资源空间', '设备标识码', '注册时间', '固件版本', '设备描述', and '最近离线时间'. On the right, there are fields for '设备ID', '认证类型', '节点类型', '软件版本', '激活时间', and 'MQTT连接参数'. Below these, there's a section titled '物模型数据' (Thing Model Data) which says '当产品下定义物模型，设备按照物模型定义呈现上报的属性数据'. It lists services: '请输入服务名称' (BasicData, highlighted in red), '最新更新时间' (redacted), and '请输入属性名称' (luminance, highlighted in red). The value '30' is shown under 'luminance'. At the bottom, there are pagination controls: '16' (dropdown), '总条数: 1', and navigation arrows.

----结束

命令下发

在管理控制台下发命令，远程控制设备。

步骤1 选择“设备 > 所有设备”，找到新建的设备，单击“详情”进入设备详情页面。

步骤2 单击“云端下发”页签，下发同步命令。

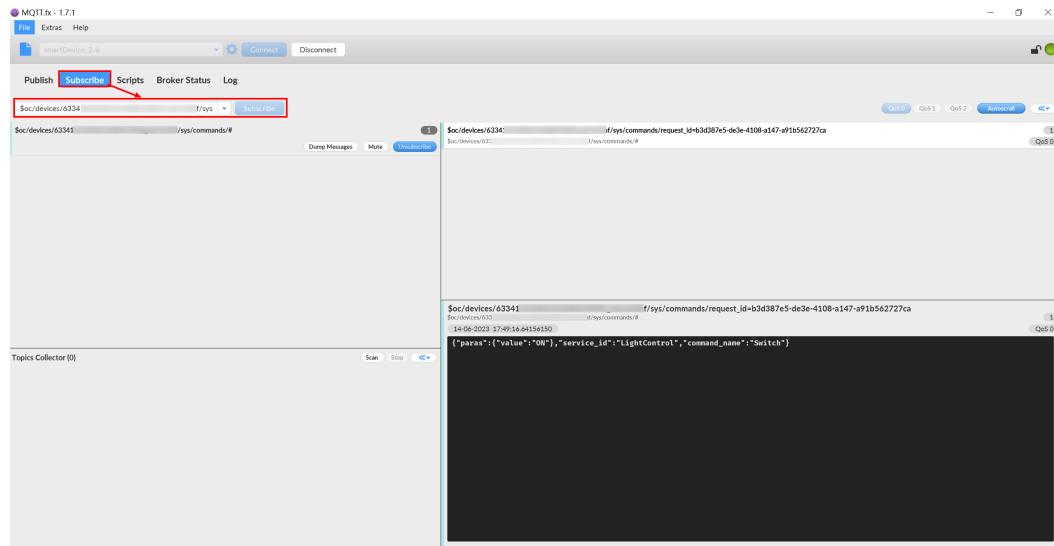
图 2-13 命令下发-MQTT



说明

MQTT协议设备仅支持同步命令下发，NB-IoT设备仅支持异步命令下发。

步骤3 MQTT.fx模拟器中选择“Subscribe”，输入命令下发Topic，订阅后则可以查看到下发的命令参数。



说明

- 通过MQTT.fx模拟器查看下发的命令参数，命令下发Topic格式为：\$oc/devices/{device_id}/sys/commands/#，其中{device_id}为设备注册成功后返回的deviceId值。
- 界面提示命令下发失败，是因为下发的同步命令需要设备侧及时响应，响应内容请参考[平台命令下发](#)。

----结束

进阶体验

按照本页面的指导，使用MQTT.fx模拟MQTT设备接入物联网平台后，您应该已经基本了解MQTT设备调用物联网平台开放的接口与平台交互。

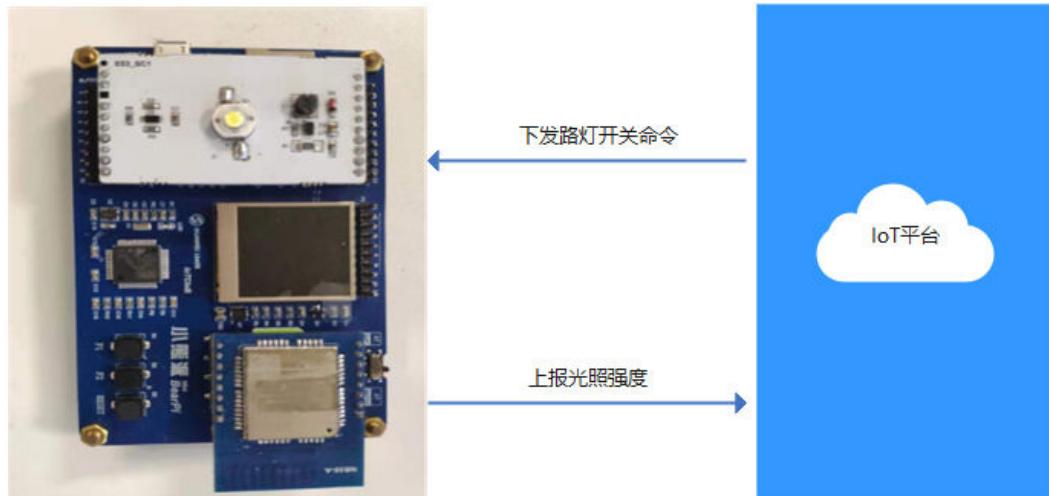
若您想要进一步体验设备接入服务，可参考[开发指南](#)开发真实应用和真实设备，并接入物联网平台，体验更多功能。

2.2 基于 NB-IoT 小熊派开发智慧路灯

场景说明

智慧路灯是城市智能化道路上重要一环，智慧路灯的实施具有节约公共照明能耗、减少因照明引起的交通事故等多种社会意义。路灯也是大家在日常生活中可以强烈感知到的公共设施，更易理解其智能化的场景。

在该文档中，我们基于华为一站式开发工具平台（VS code工具IoTlink插件），从设备（小熊派开发套件）、平台（IoTDA华为物联网平台）、端到端构建一款智慧路灯解决方案样例，带您体验十分钟快速上云。实现智慧路灯检测并上报光照强度，显示在IoTDA控制台，并在IoTDA控制台远程控制LED灯开关的功能。

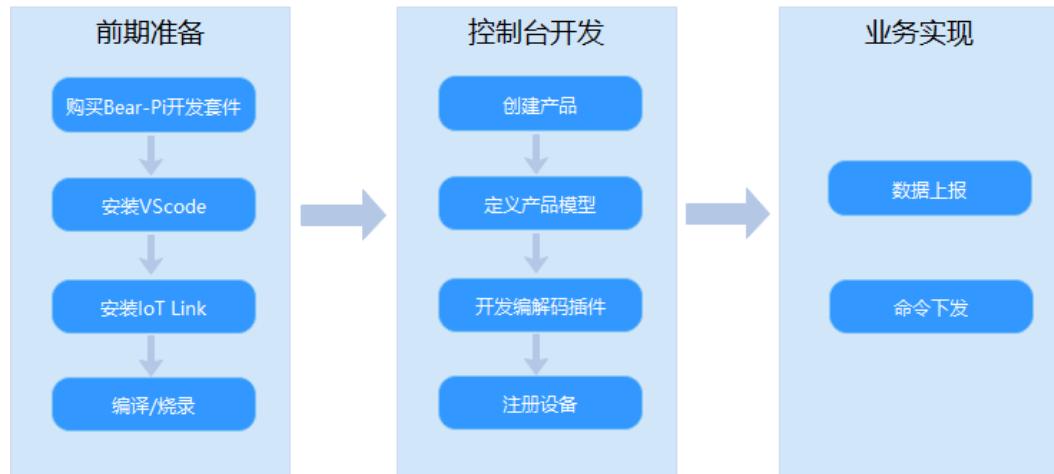


开发环境

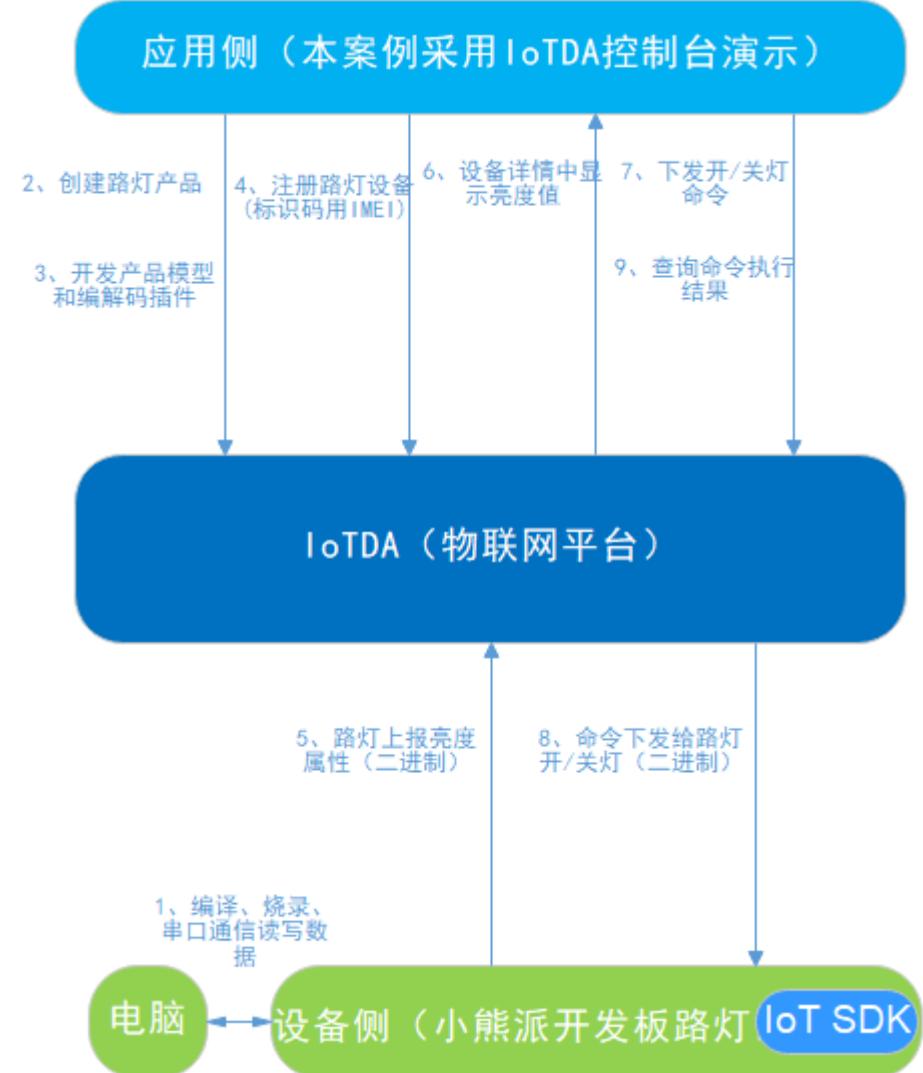
- 硬件：小熊派BearPi-IoT开发套件（包含NB卡、NB模组、智慧路灯功能模块、USB数据线等）。
- 软件：VScode, IoT Link插件、华为云（已开通[设备接入服务](#)）、win7及以上版本的64位操作系统（本文以windows 10 64-bit系统调试）。

整体流程

智慧路灯案例的端到端整体流程如下图所示。



该场景主要描述的是设备可以通过LWM2M协议（NB卡）与物联网平台进行交互，应用侧可以查看设备侧属性变化，也可以给设备侧下发命令。



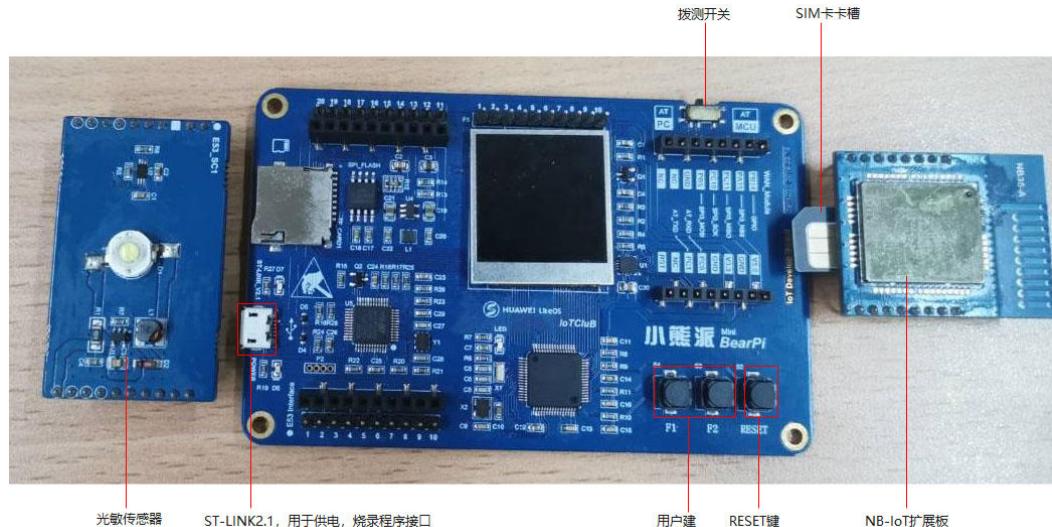
小熊派开发板简介

开发板在物联网系统架构中属于感知设备，该类设备通常由传感器、通信模组、芯片以及操作系统组成。为增加开发板的可扩展性，小熊派开发板没有采用传统的板载设计，而是使用了可更换传感器扩展板以及可更换通信模组扩展板设计，通信模组是数据传输的出入口，常用的通信模组包括NB-IoT，Wifi以及4G等，芯片是设备的主控设备，该开发板内置了一个低功耗的STM32L431单片机作为主控芯片，即MCU。操作系统使用的是华为的LiteOS操作系统，其提供了丰富的端云互通组件。

为了便于开发调试，如图1所示，该开发板板载了2.1版本的ST-Link，它具有在线调试烧录，拖拽下载以及虚拟串口等功能。开发板中间板载一块分辨率为240 * 240的LCD屏幕，其主要用于显示传感器数据以及调试日志。LCD屏幕下方是主控芯片。

开发板右上角具有一个拨码开关，将其拨至左侧AT-PC模式，通过电脑端的串口助手，发送AT指令调试通信模组。右侧AT-MCU模式，通过MCU发送AT指令与通信模组进行交互，将采集到的传感器数据通过通信模组发送到云端。

图 2-14 小熊派开发板



硬件连接

1. 将NB卡插入到NB-IoT扩展板的SIM卡卡槽，确保插卡的时候卡的缺口朝外插入，如图2所示。
2. 将光敏传感器以及NB-IoT扩展板插入到开发板上，注意安装方向，然后用USB数据线将小熊派开发板与电脑连接起来。显示屏有显示，电源灯被点亮，说明开发板通电成功。

图 2-15 硬件连接



安装 IoT Link Studio 插件

IoT Link Studio是针对IoT物联网端侧开发的IDE环境，提供了编译、烧录、调试等一站式开发体验，支持 C、C++、汇编等多种开发语言，让您快速、高效地进行物联网开发。

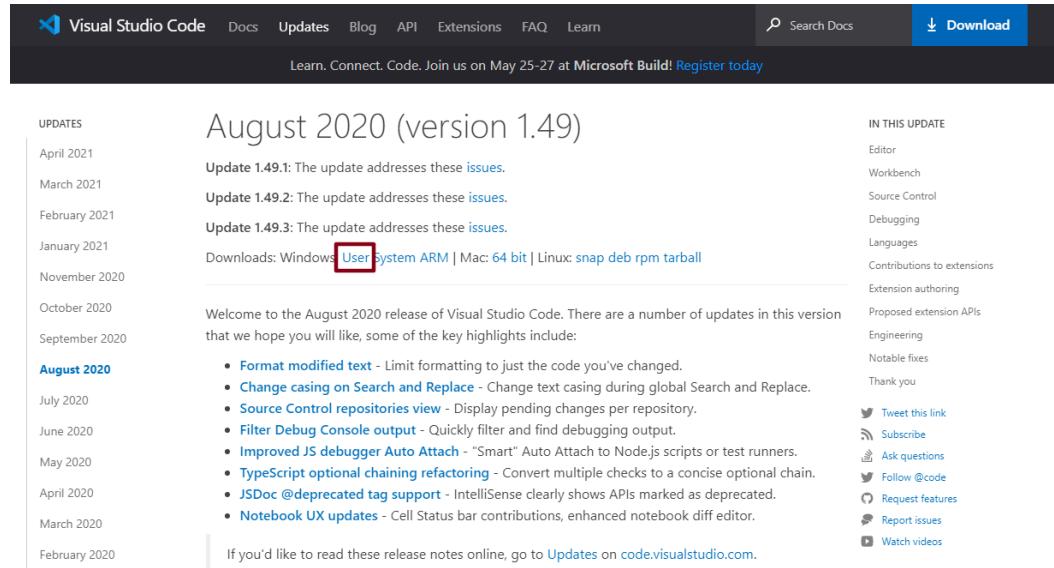
步骤1 以win10为例，获取电脑的系统配置：在运行窗口中输入pc，然后单击“属性”，查看系统配置，因为后续安装软件要选择操作系统。

图 2-16 获取系统配置



步骤2 访问 [这里](#)，根据您电脑系统配置，下载匹配的Visual Studio Code并安装。（本文以 windows 10 64-bit系统Visual Studio Code为例。请下载1.49版本，其他版本不支持 IoT Link）。

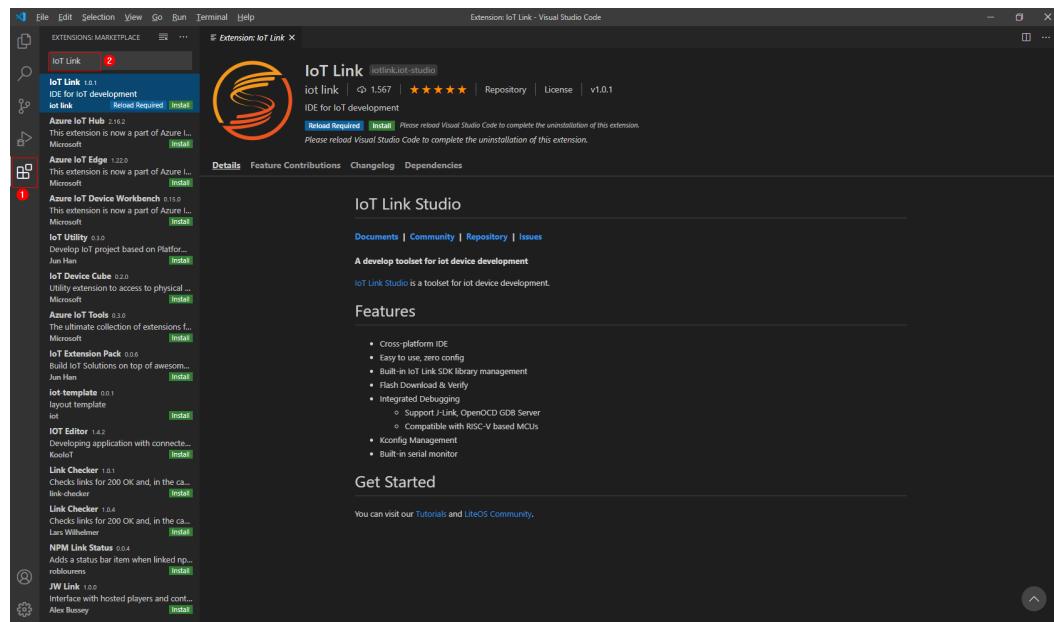
图 2-17 下载 Visual Studio Code



注：Visual Studio Code不支持苹果Mac系统。

步骤3 安装成功后，打开VSCode 插件应用商店，搜索 IoT Link 找到IoT Link，然后单击安装。

图 2-18 安装



步骤4 首次启动配置。

IoT Link Studio 首次启动时会自动从网络下载最新的SDK包以及gcc依赖环境，请确保您的网络可用。安装过程中请不要关闭窗口，耐心等待。安装完成后重启VSCode使插件生效。

说明

若您的网络需要配置代理，请在VSCode主页，选择左下角设置-应用程序-代理服务器，在Proxy Support下拉框中选择“on”。

步骤5 假如下载SDK包及gcc依赖环境失败，请手动下载[SDK包](#)，放到C:\Users\\$用户名\$\.iotlink\ sdk目录下,文件名修改为IoT_LINK。放置完后，重新打开VSCode即可。目录格式如下图：

图 2-19 SDK 下载失败

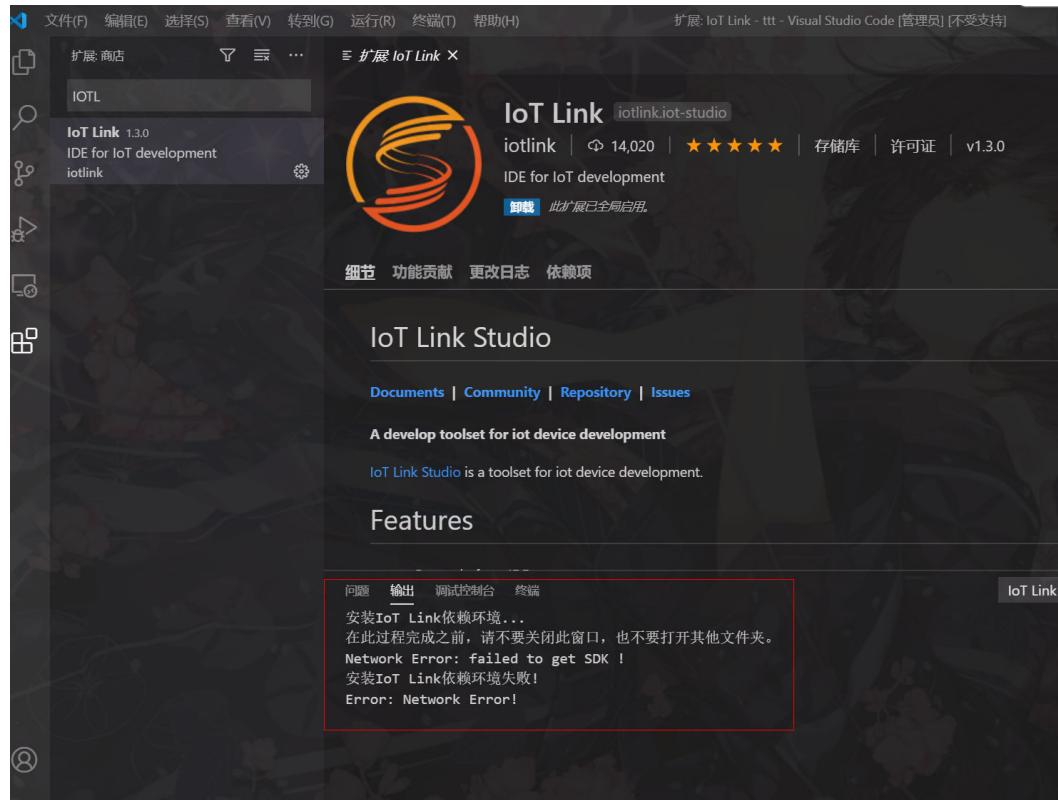


图 2-20 SDK 存放目录

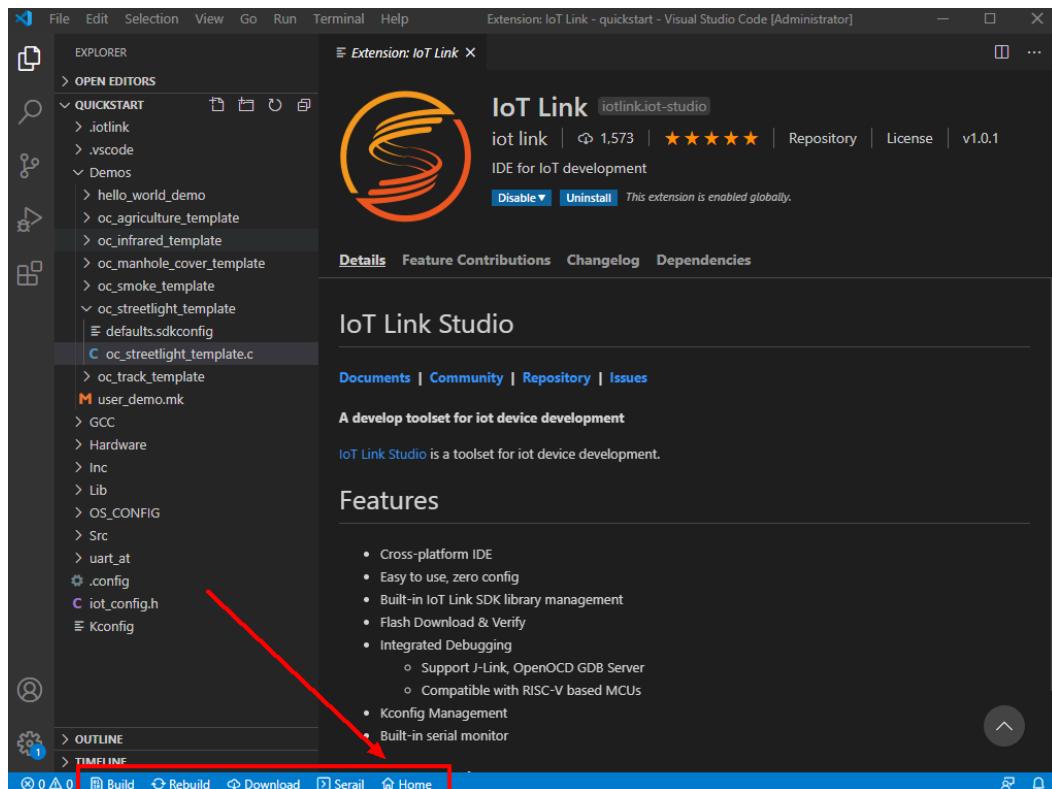
| 名称 | 修改日期 | 类型 |
|----------|------------------|-----|
| .github | 2023/10/26 20:03 | 文件夹 |
| doc | 2023/10/26 20:03 | 文件夹 |
| drivers | 2023/10/26 20:03 | 文件夹 |
| FAQ | 2023/10/26 20:03 | 文件夹 |
| iot_link | 2023/10/26 20:05 | 文件夹 |

----结束

配置 IoT link Studio 工程

步骤1 单击VSCode底部工具栏的“Home”。

- Home: 管理IoT Link工程;
- Serial: 输入AT指令检查开发板状态;
- Build: 编译示例代码（步骤3后可见）;
- Download: 把编译后的代码烧录到MCU（步骤3后可见）。



步骤2 配置交叉编译工具链。在弹出界面中单击“IoT Link设置”，选择工具链，若GCC工具目录或文件不存在，点击[下载安装](#)。

图 2-21 IoT Link 设置

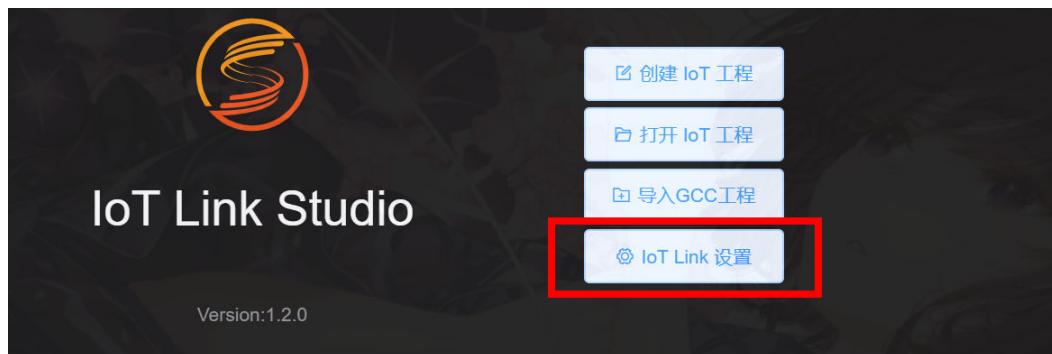
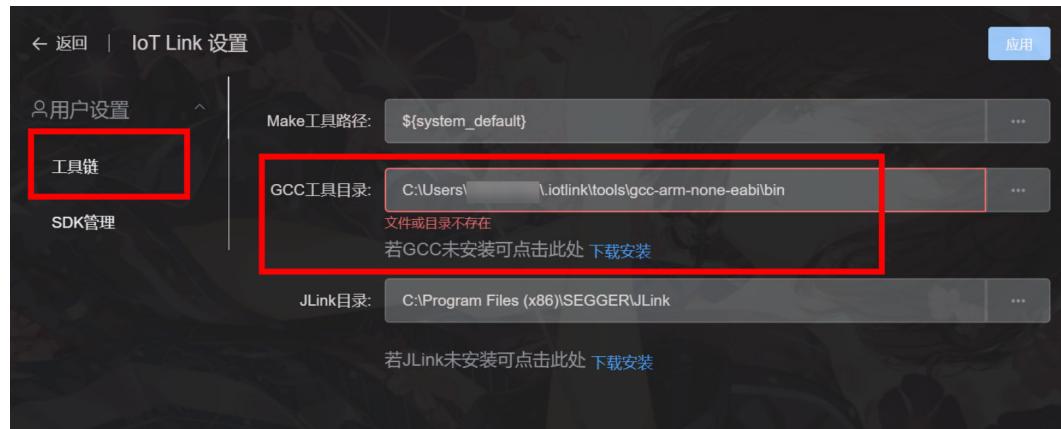


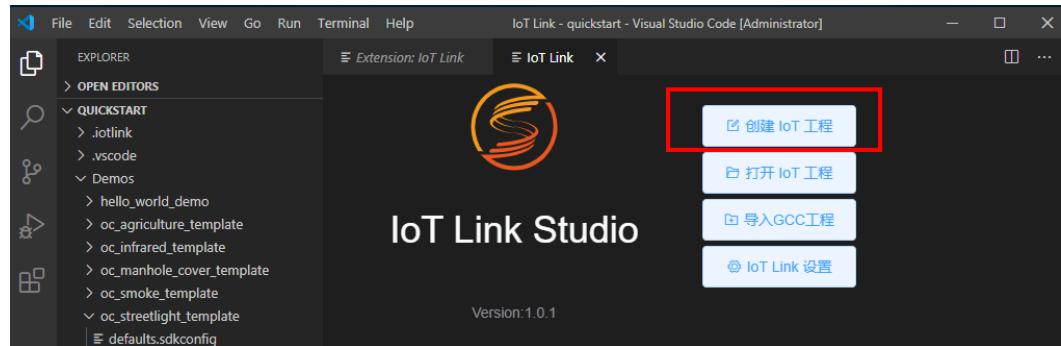
图 2-22 交叉编译工具链

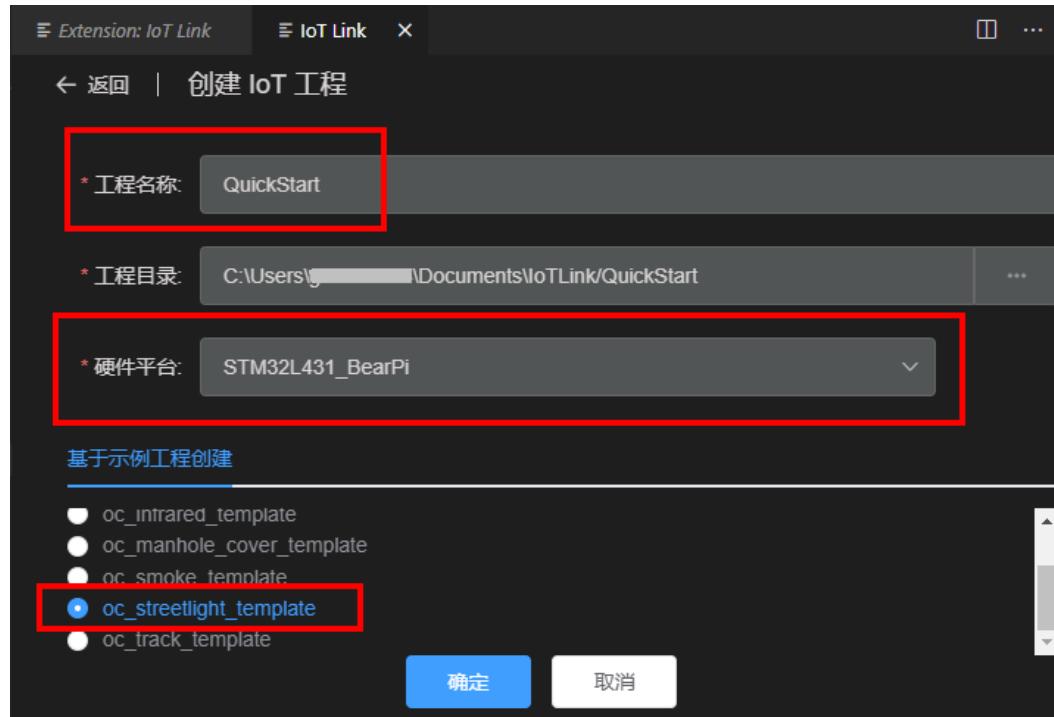


说明

小熊派STM32431下载的编译工具链版本为win32.zip。

步骤3 在弹出界面中单击“创建 IoT 工程”，输入工程名称，工程目录，并选择开发板的硬件平台和示例工程模板。





- 工程名称：自定义，如QuickStart。
- 工程目录：可以使用工具安装的默认路径，也可以选择系统盘以外的其他盘，如D:\。
- 硬件平台：当前提供的demo只适配STM32L431_BearPi硬件平台，请选择STM32L431_BearPi。
- 示例工程：本示例中，我们以智慧路灯为例，请选择oc_streetlight_template，否则烧录的demo样例和在控制台定义的产品模型不匹配，无法上报数据。如果您需要适配智慧烟感等其他场景，请选择对应的oc_smoke_template demo样例。

步骤4 单击“确定”，导入完成。

----结束

编译并烧录代码

由于提供的demo样例中已配置好和华为云物联网平台的对接信息，您可以直接编译（代码不用做任何修改），并烧录到小熊派开发板MCU，节省开发时间。

步骤1 单击VSCode底部工具栏的“Build”，等待系统编译完成。编译成功后，界面显示“编译成功”。

步骤2 使用USB数据线，将小熊派开发与电脑连接，开发板右上角的拨测开关拨到右侧“AT-MCU模式”。

步骤3 单击VSCode底部工具栏的“Download”，等待系统烧录完成。烧录成功后，界面显示“烧录成功”。

说明

如果显示烧录失败，可能是开发板没有驱动导致与电脑无法串口通信，请参考[2检查ST-Link驱动是否安装成功](#)，如果驱动未安装，则参考[这里](#)下载并安装ST-Link驱动。

步骤4（可选）安装ST-Link驱动。

1. 访问[ST官网](#)，下载ST-Link驱动，双击stlink_winusb_install.bat文件进行自动安装。（本文以Windows10-64bit ST-Link 2.0.1为例）。

| Name | Size | Packed | Type |
|----------------------------|---------|---------|----------------|
| .. | | | 文件夹 |
| amd64 | | | 文件夹 |
| x86 | | | 文件夹 |
| dpinst_amd64.exe | 680,440 | 242,269 | 应用程序 |
| dpinst_x86.exe | 552,328 | 231,230 | 应用程序 |
| readme.txt | 391 | 250 | 文本文档 |
| stlink_bridge_winusb.inf | 2,853 | 1,093 | 安装信息 |
| stlink_dbg_winusb.inf | 4,373 | 1,347 | 安装信息 |
| stlink VCP.inf | 2,467 | 871 | 安装信息 |
| stlink_winusb_install.bat | 412 | 259 | Windows 批处理... |
| stlinkbridgewinusb_x64.cat | 11,004 | 5,890 | 安全目录 |
| stlinkbridgewinusb_x86.cat | 11,004 | 5,892 | 安全目录 |
| stlinkdbgwinusb_x64.cat | 10,997 | 5,891 | 安全目录 |
| stlinkdbgwinusb_x86.cat | 10,998 | 5,892 | 安全目录 |
| stlinkvcp_x64.cat | 9,248 | 5,474 | 安全目录 |
| stlinkvcp_x86.cat | 9,247 | 5,470 | 安全目录 |

注：您也可以使用适配您系统版本的exe文件进行安装。

2. 打开PC设备管理器可查看对应的驱动是否安装成功。若下图所示，表明驱动安装成功。



----结束

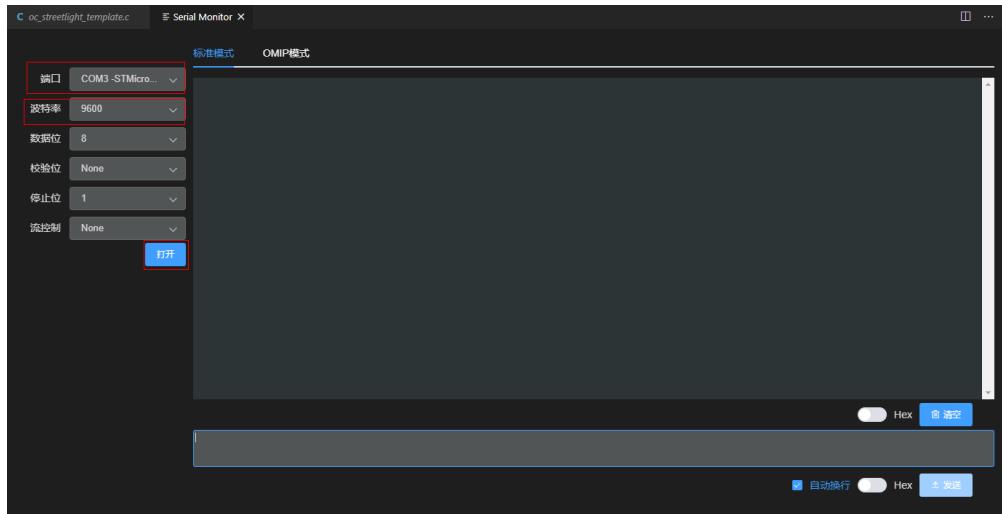
使用 AT 指令定位模组通信问题

IoT Link在与物联网平台连接使用时，可使用AT指令快速定位模组与云端连通性问题，提高开发效率。本节以“小熊派开发板”为例，介绍如何使用AT指令检测通信模组常见问题，如设备无法上线，数据上报不成功等。

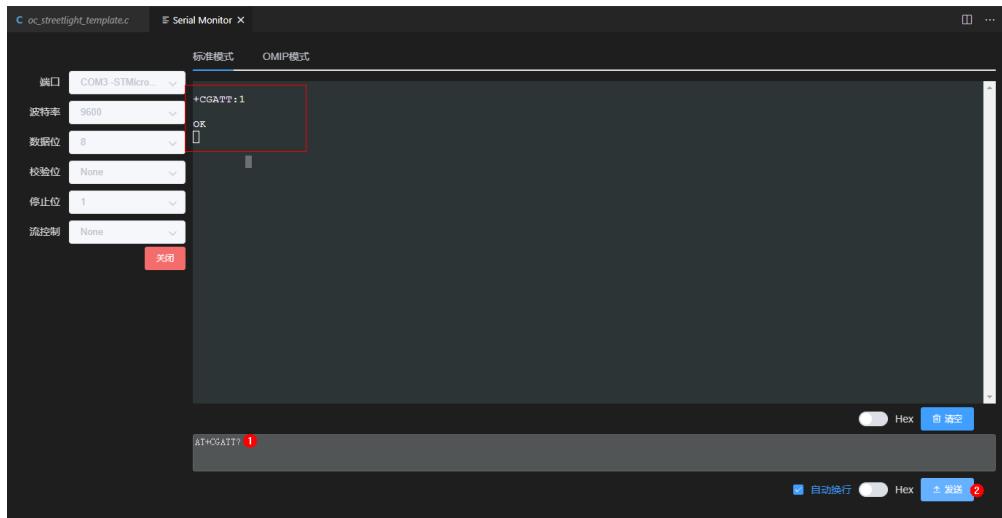
1. 小熊派开发板和电脑已正常连接（确保驱动已安装），并将开发板右上角的拨测开关切换到AT-PC模式。
2. 单击VSCode底部工具栏的“Serial”。



3. 选择2中查看的端口号，波特率设置为9600，然后单击“打开”。



4. 输入“AT+CGATT?”，然后单击“发送”，若返回“+CGATT:1”，表示网络附着成功（附着成功代表NB-IoT联网正常），返回“+CGATT:0”表示网络附着失败（附着失败代表NB-IoT联网异常），请查看SIM卡是否插入正确，或联系运营商检查网络状态。



□ 说明

使用AT指令检测完模组通信后，请将拨测开关拨到AT-MCU模式，以便完成控制台的开发后，将采集到的传感器数据通过通信模组发送到平台。

PC模式是开发板与电脑串口通信，AT指令读写开发板的状态等数据；MCU模式是开发板通过模组上插的SIM连接网络，实现NB-IoT通信。

5. “AT+CSQ<CR>”指令用于检查网络信号强度和SIM卡情况。输入“AT+CSQ<CR>”，然后单击“发送”，返回“+CSQ:##,##”。其中##应在 10 到 31 之间，数值越大表明信号质量越好，##为误码率,值在 0 到 99 之间。若返回的值不在这个范围，应检查天线或 SIM 卡是否正确安装。

注：本文仅列举两个通用的AT指令，用于检测模组的网络状况。更多AT指令，您可以参考小熊派模组的说明书，进行调测。

控制台操作概览

在完成实体设备硬件连接和代码编译烧录之后，我们需要在设备接入服务控制台创建产品、定义产品模型、开发编解码插件和注册设备。

- 创建产品：在物联网平台上规定好某一款产品的协议类型、数据格式、厂商名称、设备类型。此处我们需要按照智慧路灯的特征，在控制台创建智慧路灯的产品。
- 定义产品模型：产品模型是用来描述设备能力的文件，通过JSON的格式定义了设备的基本属性、上报数据和下发命令的消息格式。定义产品模型，即在物联网平台构建一款设备的抽象模型，使平台理解该款设备支持的属性信息。此处，我们需要在控制台上，定义开关灯控制、光照强度、信号质量等。
- 编解插件开发：编解码插件是供物联网平台调用，完成二进制格式和JSON格式相互转换的。它将设备上报的二进制数据解码为JSON格式供应用服务器“阅读”，将应用服务器下行的JSON格式命令编码为二进制格式数据供终端设备（UE）“理解执行”。智慧路灯的数据格式是二进制，因此我们需要开发编解码插件，让物联网能够理解智慧路灯上报的数据，智慧路灯也能理解物联网平台下发的命令。
- 注册设备：将小熊派智慧路灯注册到物联网平台。

创建产品

某一类具有相同能力或特征的设备的集合称为一款产品。除了设备实体，产品还包含该类设备在物联网能力建设中产生的产品信息、产品模型（Profile）、编解码插件等资源。下面，我们按照小熊派智慧路灯的基本特征，在控制台上创建智慧路灯产品。

步骤1 登录[管理控制台](#)，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

图 2-23 产品-创建产品



步骤2 创建一个协议类型为LwM2M/CoAP协议、设备类型为StreetLamp的产品，参考页面提示填写参数后，单击“确定”。

图 2-24 创建产品-CoAP



----结束

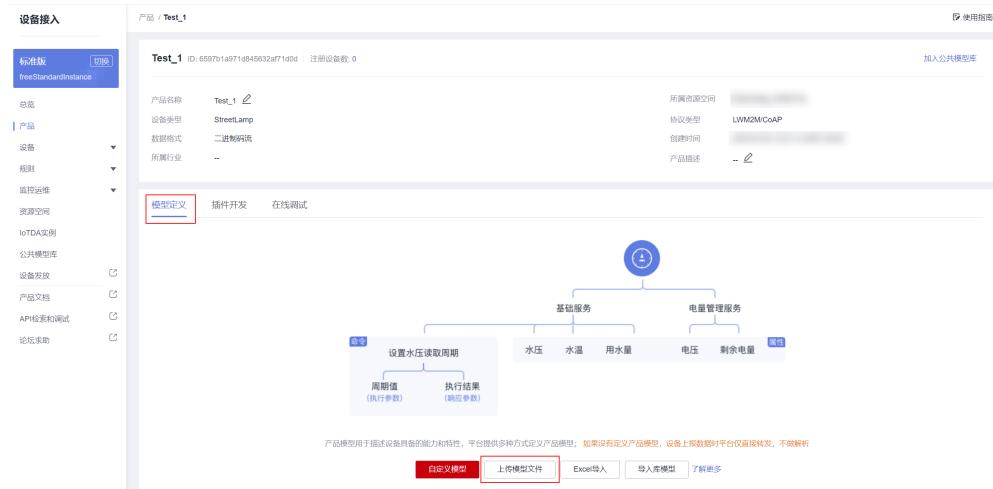
上传产品模型

产品模型是用来描述设备能力的文件，通过JSON的格式定义了设备的基本属性、上报数据和下发命令的消息格式。定义产品模型，即在物联网平台构建一款设备的抽象模型，使平台理解该款设备的功能。我们已提供开发好的产品模型，帮助您快速体验上云流程，如果您想体验产品模型的开发流程，可参考[开发产品模型](#)。

操作步骤：

- 步骤1 找到新增的产品，单击产品进入产品界面。
- 步骤2 在产品详情“模型定义”页面，单击“上传模型文件”。

图 2-25 上传产品模型-CoAP



步骤3 在弹出的页面中，上传提供的产品模型（[本示例的产品模型](#)），然后单击“确认”。

图 2-26 上传模型文件-CoAP



----结束

注册设备

本文介绍集成NB模组设备的注册方法，将小熊派智慧路灯在物联网平台注册。

步骤1 在产品详情页面，选择“在线调试”，单击“新增测试设备”，此处新增的是非安全的NB-IoT设备。

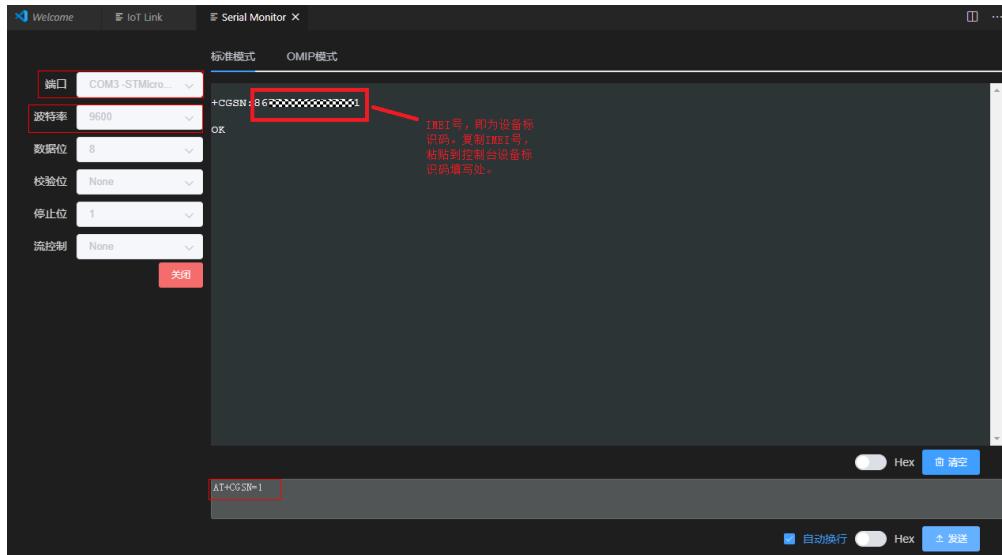
步骤2 在新增测试设备页面，选择“真实设备”，完成设备参数的填写后，单击“确定”。

图 2-27 在线调试-新增测试设备



- 设备名称：自定义。
- 设备标识码：设备的IMEI号，用于设备在接入物联网平台时携带该标识信息完成接入鉴权，可在NB模组上查看。您也可以将拨测开关拨到AT-PC模式，选择STM的端口，波特率设置为9600，输入指令“AT+CGSN=1”获取IMEI号。

注：获取IMEI注册完设备后，需要将开发板的拨测开关拨到“AT-MCU”模式，因为开发板在MCU模式下才会通过NB卡连接网络。



- 设备注册方式：不加密。

步骤3 设备创建成功，可在页面看到创建的设备。

----结束

数据上报

平台和开发板建立连接后，根据烧录至开发板的代码，小熊派智慧路灯每隔2秒（上报频率可根据业务需要在demo中自行设置）上报光照传感器数据。您可以通过用手遮挡光线改变光照强度，查看设备上报给平台的光照强度数据的实时变化。

注：先检查确认开发板的拨测开关是“AT-MCU”模式。

步骤1 登录**设备接入服务控制台**，选择“设备 > 所有设备”。

步骤2 选择此处注册的设备，单击“查看”进入设备详情，查看上报到平台的数据。

The screenshot shows the 'Device Management' page. At the top, it displays basic device information:所属资源空间 (Resource Space), 设备标识码 (IMEI), 注册时间 (Registration Time), 固件版本 (Firmware Version), and 设备描述 (Device Description). To the right, it shows the device ID (IMEI), 认证类型 (Authentication Type), 节点类型 (Node Type), and 软件版本 (Software Version). Below this, the 'Latest Report Data' section shows a table with columns for service name, update time, and value. The table includes rows for Button, LED, Sensor, and Connectivity, with the Sensor row showing 'luminance' with a value of '206'. There are also search and filter options for the data table.

----结束

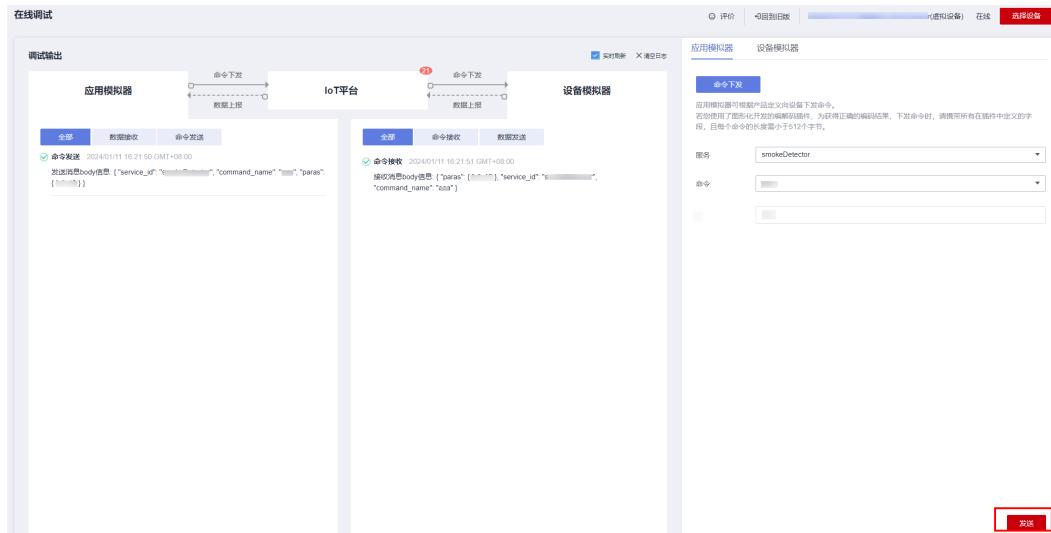
命令下发

步骤1 登录**设备接入服务控制台**，选择之前创建的产品，单击产品进入产品界面。

步骤2 选择“在线调试”，单击注册的设备进入调试页面。

步骤3 设置命令参数后，单击“命令发送”。

图 2-28 命令下发调试-SmokeDetector



步骤4 小熊派板子灯被点亮，下发OFF命令后，灯熄灭。



----结束

至此，基于NB-IoT小熊派开发智慧路灯的最佳实践，端到端流程就全部体验完成了。

更多参考

- **开发产品模型**

在产品详情“模型定义”页面，单击“自定义模型”，配置产品的服务。

产品模型设计思路：

表 2-3 设备服务列表

| 服务类型 (ServiceID) | 服务描述 |
|------------------|----------|
| Button | 实时按键检测 |
| LED | LED灯控制 |
| Sensor | 实时检测光照强度 |
| Connectivity | 实时检测信号质量 |

服务能力如下表所示。

表 2-4 Button

| 能力描述 | 属性名称 | 数据类型 | 数据范围 |
|------|--------|------|-----------|
| 属性列表 | toggle | int | 0 ~ 65535 |

表 2-5 LED 命令列表

| 能力描述 | 命令名称 | 命令字段 | 字段名称 | 类型 | 数据长度 | 枚举 |
|------|---------|------|-------------|--------|------|--------|
| 命令列表 | Set_LED | 下发命令 | LED | string | 3 | ON、OFF |
| | | 响应命令 | Light_state | string | 3 | ON、OFF |

表 2-6 Sensor

| 能力描述 | 属性名称 | 数据类型 | 数据范围 |
|------|-----------|------|-----------|
| 属性列表 | luminance | int | 0 ~ 65535 |

表 2-7 Connectivity

| 能力描述 | 属性名称 | 数据类型 | 数据范围 |
|------|-------------|------|------------|
| 属性列表 | SignalPower | int | -140 ~ -44 |
| | ECL | int | 0 ~ 2 |
| | SNR | int | -20 ~ 30 |
| | CellID | int | 0 ~ 65535 |

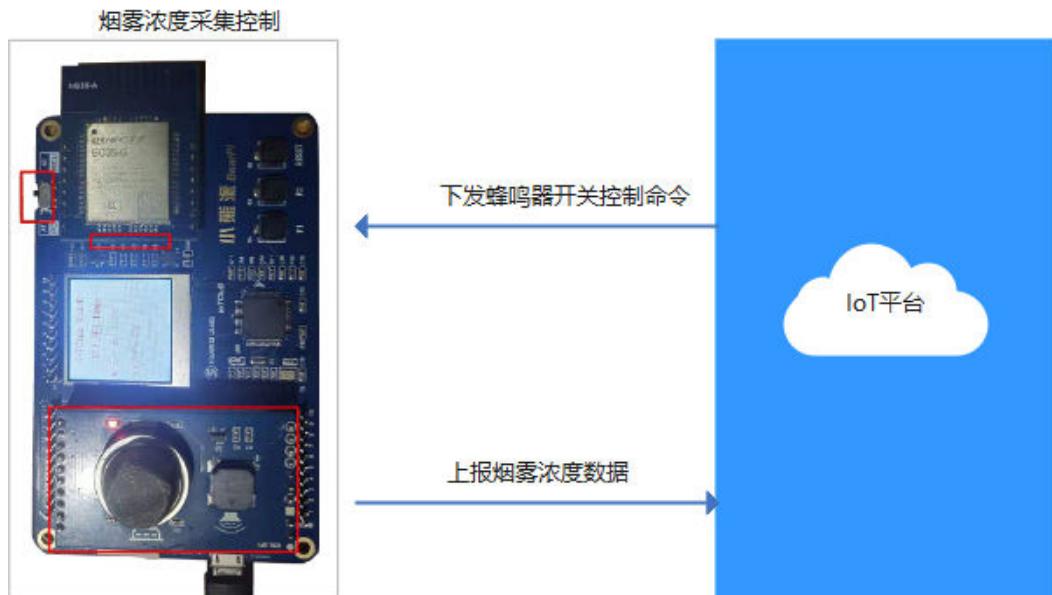
2.3 基于 NB-IoT 小熊派的智慧烟感

场景说明

近年来，火灾频发，造成很大的人员及财产损失，国家高度重视，陆续出台相关文件，明确相关要求，落实消防安全责任。同时，人民消防意识逐渐提高，独立烟感得到一定程度的普及，在防火减灾方面起到一定的作用。但由于独立烟感产品存在一定的局限，例如人不在场收不到报警信息，工作状态不能实时掌握，独立烟感没有完全解决这些问题。

NB-IoT智慧烟感解决了传统烟感器布线难、电池使用周期短、维护成本高、无法与业主及消防机构交互等缺点。智慧烟感采用无线通信，具备即插即用、无需布线、易于安装等特点。

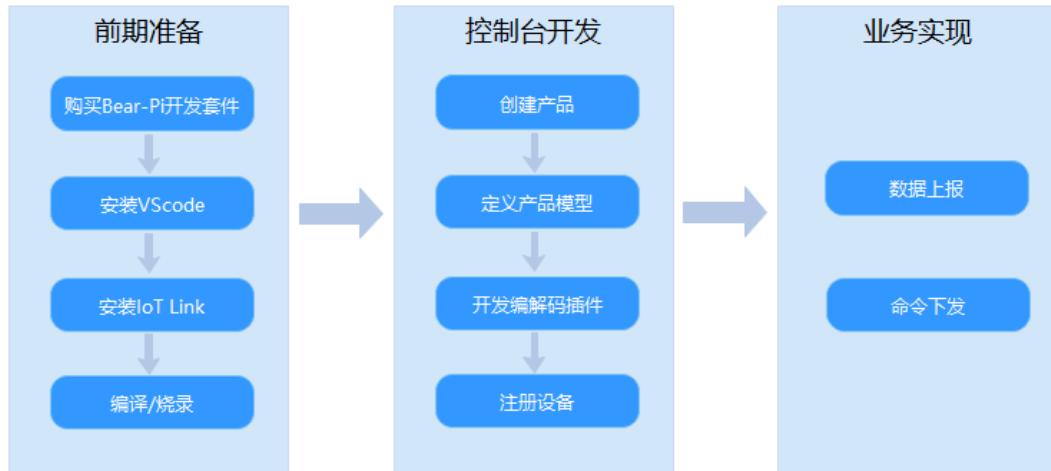
在该文档中，我们基于华为一站式开发工具平台（VS code工具IoTlink插件），从设备（小熊派开发套件）、平台（IoTDA华为物联网平台）、端到端构建一款智慧烟感解决方案样例，带您体验十分钟快速上云。实现智慧烟感检测并上报烟雾浓度值，显示在IoTDA控制台，并在IoTDA控制台远程控制蜂鸣器开关的功能。



开发环境

- 硬件：小熊派BearPi-IoT开发套件（包含NB卡、NB模组、智慧路灯功能模块、USB数据线等）。
- 软件：VScode, IoT Link插件、华为云（已开通设备接入服务）、win7及以上版本的64位操作系统（本文以windows 10 64-bit系统调试）。

整体流程



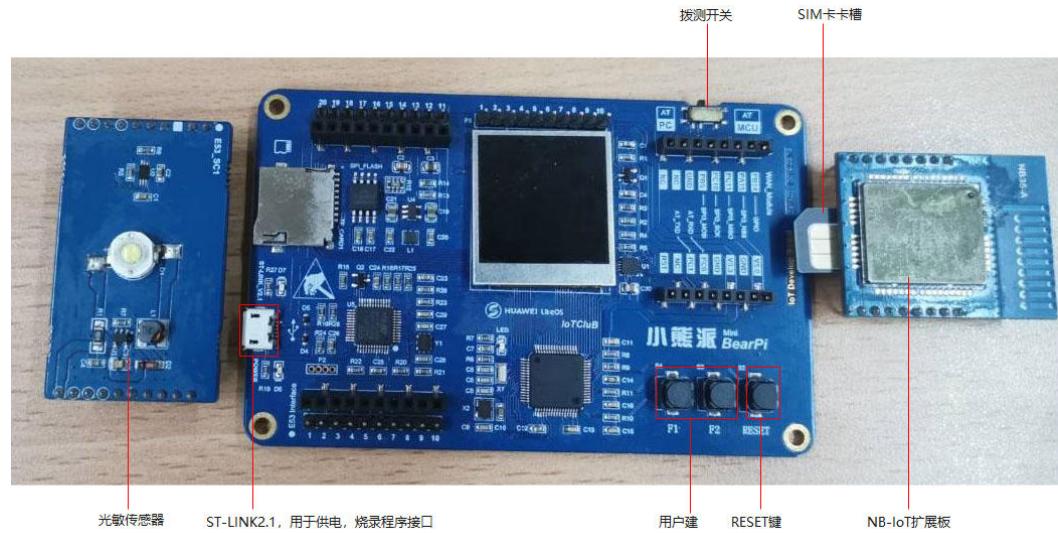
小熊派开发板简介

开发板在物联网系统架构中属于感知设备，该类设备通常由传感器、通信模组、芯片以及操作系统组成。为增加开发板的可扩展性，小熊派开发板没有采用传统的板载设计，而是使用了可更换传感器扩展板以及可更换通信模组扩展板设计，通信模组是数据传输的出入口，常用的通信模组包括NB-IoT, Wifi以及4G等，芯片是设备的主控设备，该开发板内置了一个低功耗的STM32L431单片机作为主控芯片，即MCU。操作系统使用的是华为的LiteOS操作系统，其提供了丰富的端云互通组件。

为了便于开发调试，如图1所示，该开发板板载了2.1版本的ST-Link，它具有在线调试烧录，拖拽下载以及虚拟串口等功能。开发板中间板载一块分辨率为240 * 240的LCD屏幕，其主要用于显示传感器数据以及调试日志。LCD屏幕下方是主控芯片。

开发板右上角具有一个拨码开关，将其拨至左侧AT-PC模式，通过电脑端的串口助手，发送AT指令调试通信模组。右侧AT-MCU模式，通过MCU发送AT指令与通信模组进行交互，将采集到的传感器数据通过通信模组发送到云端。

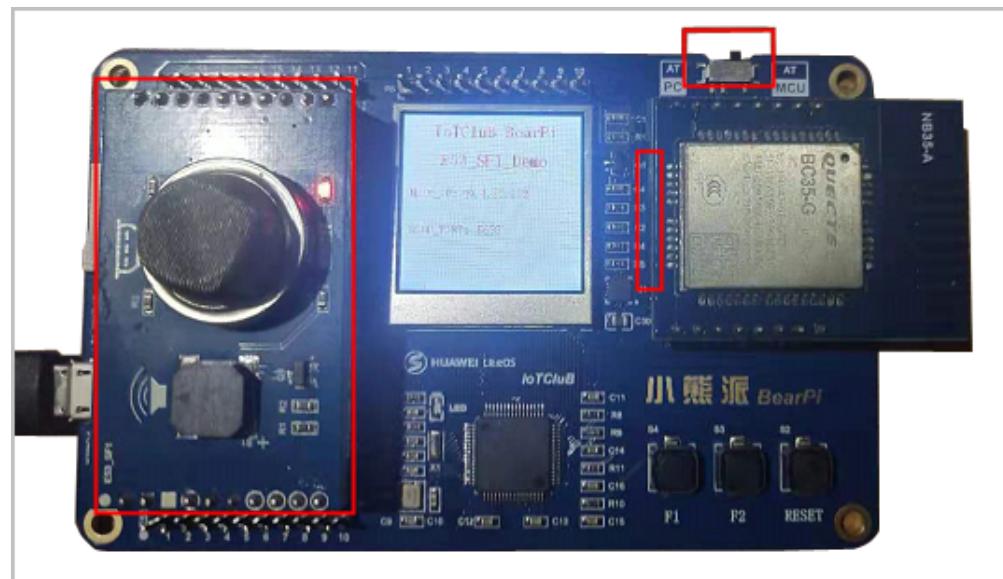
图 2-29 小熊派开发板



硬件连接

1. 将NB卡插入到NB-IoT扩展板的SIM卡卡槽，确保插卡的时候卡的缺口朝外插入，如图2所示。
2. 将烟雾浓度采集控制板和NB-IoT扩展板插入到开发板上，注意安装方向，然后用USB数据线将小熊派开发板与电脑连接起来。显示屏有显示，电源灯被点亮，说明开发板通电成功。

图 2-30 硬件连接



安装 IoT Link Studio 插件

IoT Link Studio是针对IoT物联网端侧开发的IDE环境，提供了编译、烧录、调试等一站式开发体验，支持 C、C++、汇编等多种开发语言，让您快速、高效地进行物联网开发。

步骤1 以win10为例，获取电脑的系统配置：在运行窗口中输入pc，然后单击“属性”，查看系统配置，因为后续安装软件要选择操作系统。

图 2-31 获取系统配置



步骤2 访问 [这里](#)，根据您电脑系统配置，下载匹配的Visual Studio Code并安装。（本文以 windows 10 64-bit 系统 Visual Studio Code 为例。请下载 1.49 版本，其他版本不支持 IoT Link）。

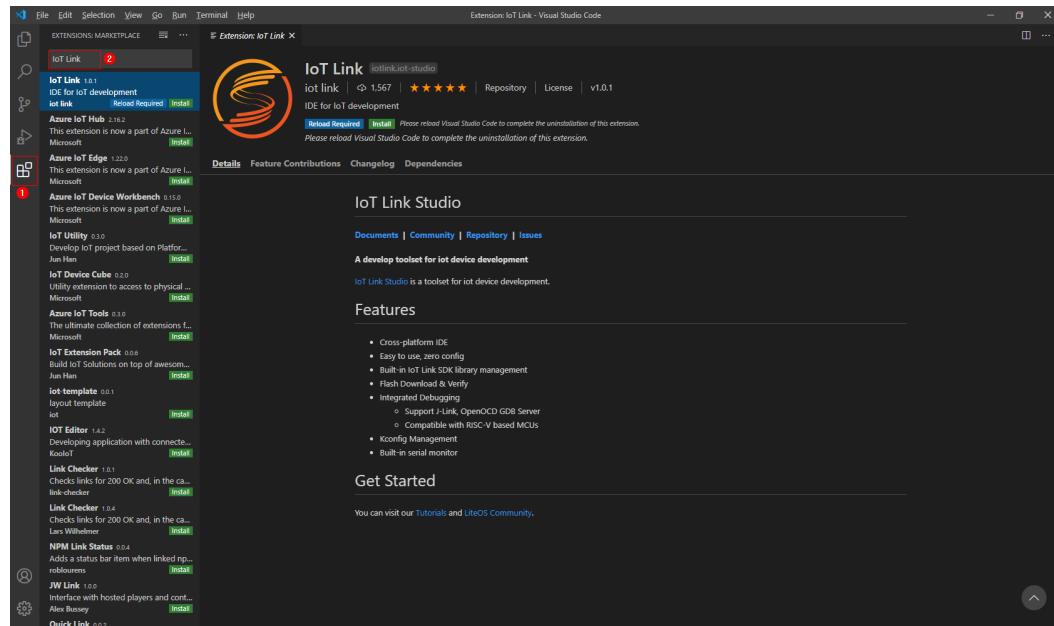
图 2-32 下载 Visual Studio Code

The image is a screenshot of the Visual Studio Code website. At the top, there is a navigation bar with links for Docs, Updates, Blog, API, Extensions, FAQ, and Learn. To the right of the navigation bar are 'Search Docs' and 'Download' buttons. Below the navigation bar, there is a banner with the text 'Learn. Connect. Code. Join us on May 25-27 at Microsoft Build: [Register today](#)'. The main content area is titled 'August 2020 (version 1.49)'. It includes a sidebar with a list of previous releases from April 2021 down to February 2020. The main content area contains several sections: 'Updates' (listing changes for version 1.49.1, 1.49.2, and 1.49.3), 'Downloads' (listing Windows, User, System ARM, Mac, and Linux options), 'Welcome' (highlighting key features like Format modified text, Change casing on Search and Replace, Source Control repositories view, Filter Debug Console output, Improved JS debugger Auto Attach, TypeScript optional chaining refactoring, JSDoc @deprecated tag support, and Notebook UX updates), and a note about reading online release notes. To the right of the main content is a sidebar titled 'IN THIS UPDATE' with a list of items such as Editor, Workbench, Source Control, Debugging, Languages, Contributions to extensions, Extension authoring, Proposed extension APIs, Engineering, Notable fixes, Thank you, Tweet this link, Subscribe, Ask questions, Follow @code, Request features, Report issues, and Watch videos.

注：Visual Studio Code不支持苹果Mac系统。

步骤3 安装成功后，打开VSCode 插件应用商店，搜索 IoT Link 找到IoT Link，然后单击安装。

图 2-33 安装



步骤4 首次启动配置。

IoT Link Studio 首次启动时会自动从网络下载最新的SDK包以及gcc依赖环境，请确保您的网络可用。安装过程中请不要关闭窗口，耐心等待。安装完成后重启VSCode使插件生效。

说明

若您的网络需要配置代理，请在VSCode主页，选择左下角设置-应用程序-代理服务器，在Proxy Support下拉框中选择“on”。

步骤5 假如下载SDK包及gcc依赖环境失败，请手动下载SDK包，放到C:\Users\\${用户名}\.iotlink\sdk目录下,文件名修改为IoT_LINK。放置完后，重新打开VSCode即可。目录格式如下图：

图 2-34 SDK 下载失败

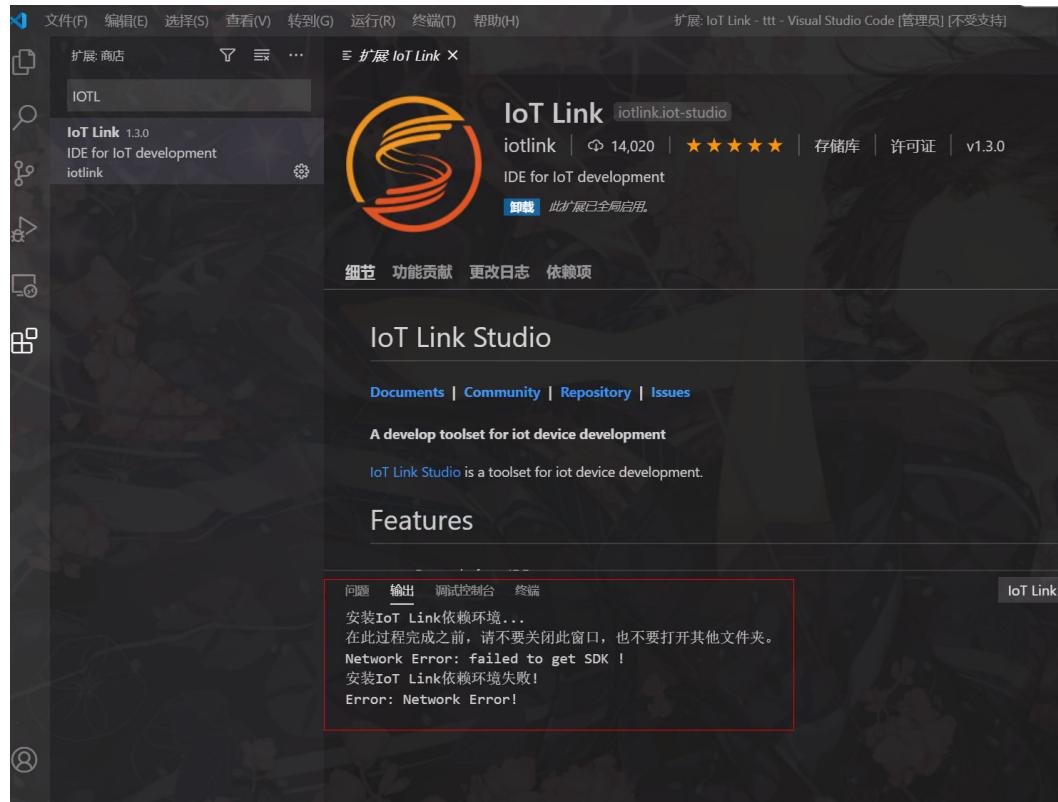
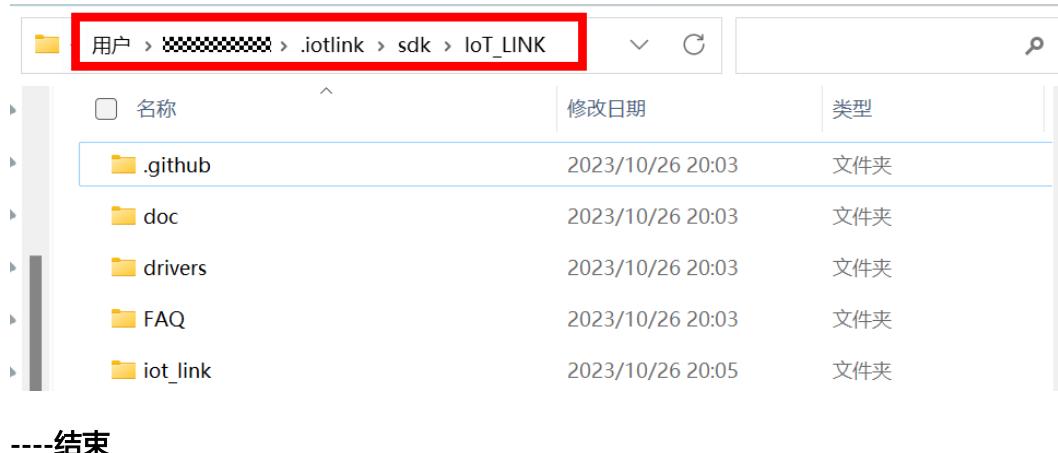


图 2-35 SDK 存放目录

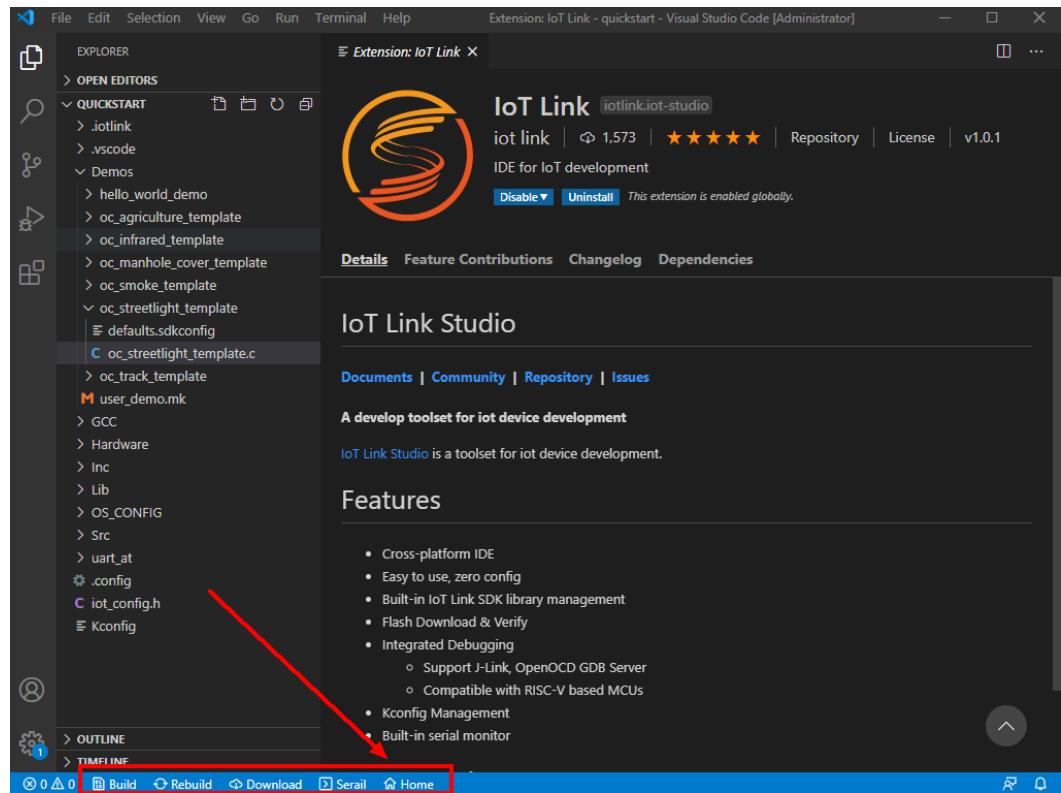


----结束

配置 IoT link Studio 工程

步骤1 单击VSCode底部工具栏的“Home”。

- Home: 管理IoT Link工程；
- Serial: 输入AT指令检查开发板状态；
- Build: 编译示例代码（步骤3后可见）；
- Download: 把编译后的代码烧录到MCU（步骤3后可见）。



步骤2 配置交叉编译工具链。在弹出界面中单击“IoT Link设置”，选择工具链，若GCC工具目录或文件不存在，点击[下载安装](#)。

图 2-36 IoT Link 设置

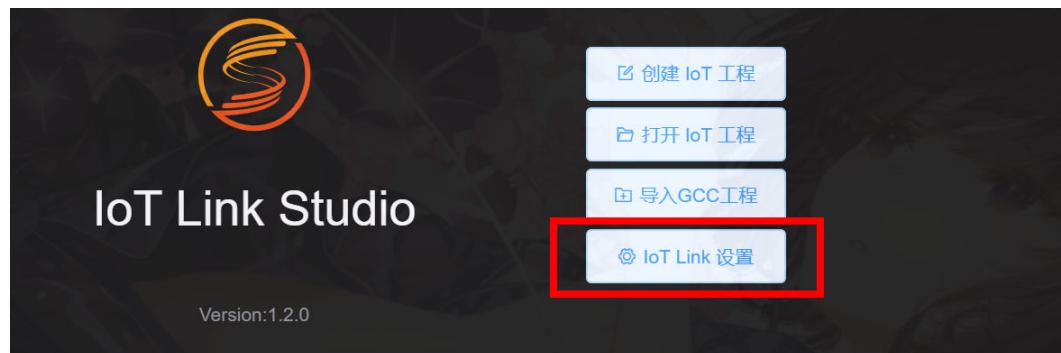
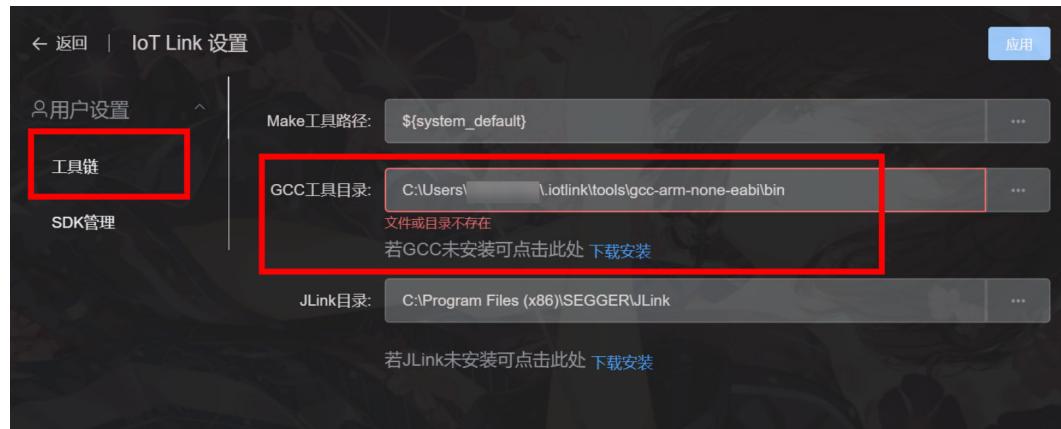


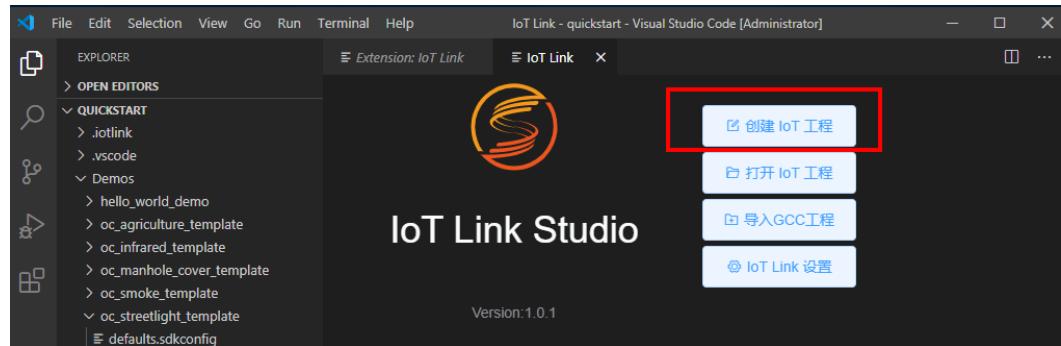
图 2-37 交叉编译工具链

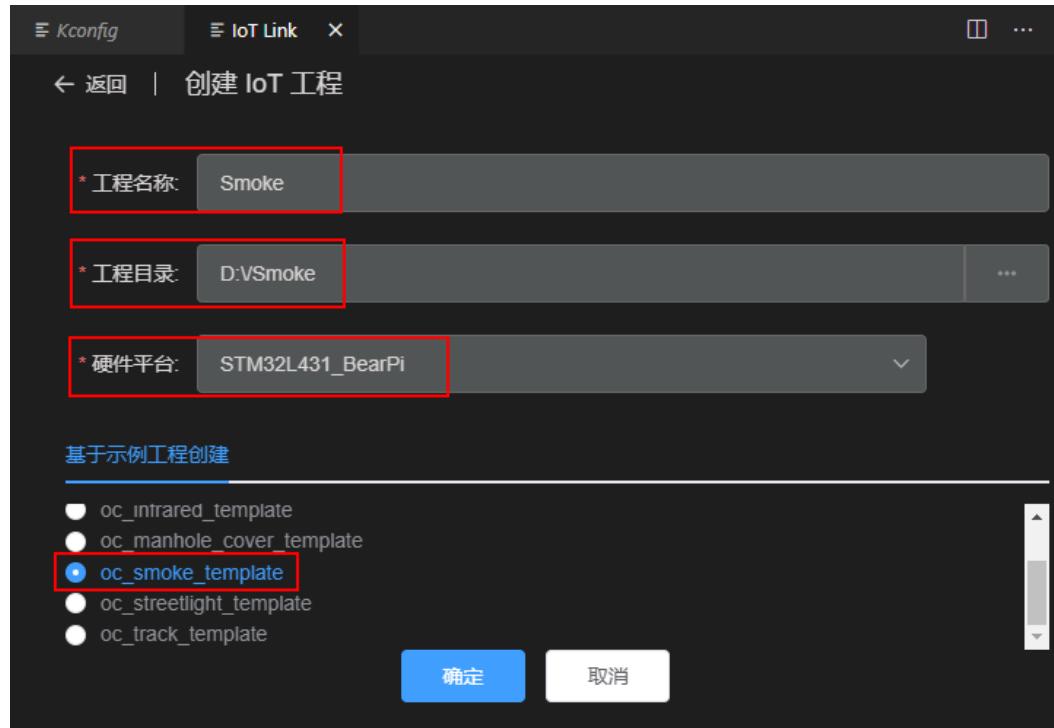


说明

小熊派STM32431下载的编译工具链版本为win32.zip。

步骤3 在弹出界面中单击“创建 IoT 工程”，输入工程名称，工程目录，并选择开发板的硬件平台和示例工程模板。





- 工程名称：自定义，如Smoke。
- 工程目录：可以使用工具安装的默认路径，也可以选择系统盘以外的其他盘，如D:\。
- 硬件平台：当前提供的demo只适配STM32L431_BearPi硬件平台，请选择STM32L431_BearPi。
- 示例工程：本示例中，我们以智慧烟感为例，请选择oc_smoke_template demo，否则烧录的demo样例和在控制台定义的产品模型不匹配，无法上报数据。单击“确定”，导入完成。

步骤4 单击“确定”，导入完成。

----结束

编译并烧录代码

由于提供的demo样例中已配置好和华为云物联网平台的对接信息，您可以直接编译（代码不用做任何修改），并烧录到小熊派开发板MCU，节省开发时间。

步骤1 单击VSCode底部工具栏的“Build”，等待系统编译完成。编译成功后，界面显示“编译成功”。

步骤2 使用USB数据线，将小熊派开发与电脑连接，开发板右上角的拨测开关拨到右侧“AT-MCU模式”。

步骤3 单击VSCode底部工具栏的“Download”，等待系统烧录完成。烧录成功后，界面显示“烧录成功”。

说明

如果显示烧录失败，可能是开发板没有驱动导致与电脑无法串口通信，请参考[2检查ST-Link驱动是否安装成功](#)，如果驱动未安装，则参考[这里](#)下载并安装ST-Link驱动。

步骤4（可选）安装ST-Link驱动。

1. 访问[ST官网](#)，下载ST-Link驱动，双击stlink_winusb_install.bat文件进行自动安装。（本文以Windows10-64bit ST-Link 2.0.1为例）。

| Name | Size | Packed | Type |
|----------------------------|---------|---------|----------------|
| .. | | | 文件夹 |
| amd64 | | | 文件夹 |
| x86 | | | 文件夹 |
| dpinst_amd64.exe | 680,440 | 242,269 | 应用程序 |
| dpinst_x86.exe | 552,328 | 231,230 | 应用程序 |
| readme.txt | 391 | 250 | 文本文档 |
| stlink_bridge_winusb.inf | 2,853 | 1,093 | 安装信息 |
| stlink_dbg_winusb.inf | 4,373 | 1,347 | 安装信息 |
| stlink_VCP.inf | 2,467 | 871 | 安装信息 |
| stlink_winusb_install.bat | 412 | 259 | Windows 批处理... |
| stlinkbridgewinusb_x64.cat | 11,004 | 5,890 | 安全目录 |
| stlinkbridgewinusb_x86.cat | 11,004 | 5,892 | 安全目录 |
| stlinkdbgwinusb_x64.cat | 10,997 | 5,891 | 安全目录 |
| stlinkdbgwinusb_x86.cat | 10,998 | 5,892 | 安全目录 |
| stlinkvcp_x64.cat | 9,248 | 5,474 | 安全目录 |
| stlinkvcp_x86.cat | 9,247 | 5,470 | 安全目录 |

注：您也可以使用适配您系统版本的exe文件进行安装。

2. 打开PC设备管理器可查看对应的驱动是否安装成功。若下图所示，表明驱动安装成功。



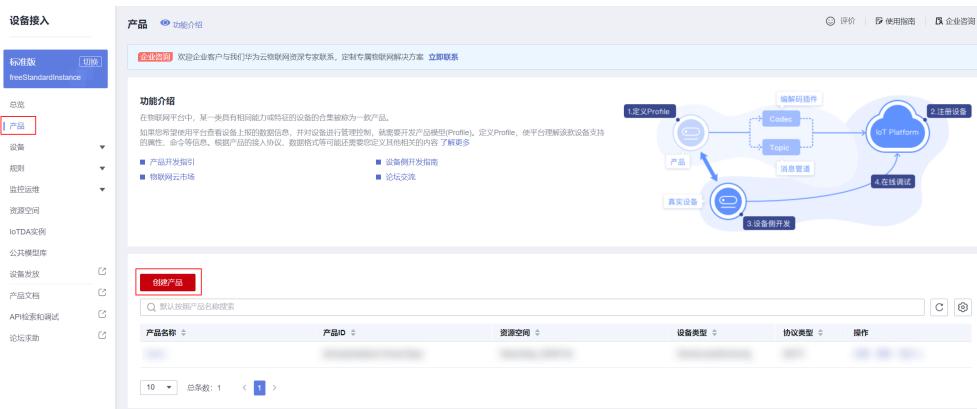
----结束

创建产品

某一类具有相同能力或特征的设备的集合称为一款产品。除了设备实体，产品还包含该类设备在物联网能力建设中产生的产品信息、产品模型（Profile）、插件等资源。

步骤1 登录[管理控制台](#)，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

图 2-38 产品-创建产品



步骤2 创建一个协议类型为LwM2M/CoAP协议、设备类型为SmokeSensors的产品，参考页面提示填写参数后，单击“确定”。

图 2-39 创建产品-CoAP

This is a detailed view of the 'Create Product' dialog for the CoAP protocol. It includes fields for 'Resource Space' (selected), 'Product Name' (Test_1), 'Protocol Type' (LwM2M/CoAP), 'Data Format' (Binary Stream), 'Device Type Selection' (Custom Type), and 'Device Type' (SmokeSensors). Advanced settings like 'Custom Product ID' and 'Remarks' are shown. At the bottom are 'Confirm' and 'Cancel' buttons.

----结束

定义产品模型

在产品详情，“模型定义”页面，单击“自定义模型”，配置产品的服务。具体创建方法可见：

产品模型设计思路：

表 2-8 设备服务列表

| 服务类型 (ServiceID) | 服务描述 |
|------------------|----------|
| Smoke | 实时检测烟雾状况 |

服务能力如表2、表3所示。

表 2-9 Smoke

| 能力描述 | 属性名称 | 数据类型 | 数据范围 |
|------|-------------|------|-----------|
| 属性列表 | Smoke value | int | 0 ~ 65535 |

表 2-10 Smoke 命令列表

| 能力描述 | 命令名称 | 命令字段 | 字段名称 | 类型 | 数据长度 | 枚举 |
|------|--------------------|------|------------|--------|------|--------|
| 命令列表 | Smoke control beep | 下发命令 | beep | string | 3 | ON、OFF |
| | | 响应命令 | beep_state | int | / | / |

新增服务类型Smoke

- 进入“添加服务”页面，填写“服务ID”、“服务类型”和“服务描述”，单击“确认”。
 - “服务ID”：Smoke
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：实时检测烟雾状况

图 2-40 添加服务-Smoke



2. 在“Smoke”的下拉菜单下单击“添加属性”，填写相关信息，单击“确认”。
- 属性名称：Smoke_Value
 - “数据类型”：int（整型）
 - “访问权限”：可读、可写
 - “取值范围”：0~65535
 - “步长”：0
 - “单位”：不填写

图 2-41 新增属性-Smoke_value



3. 在“Smoke”的下拉菜单下单击“添加命令”，填写命令名称“Smoke_Control_Beep”。

图 2-42 新增命令-Smoke_Control_Beep



4. 在“新增命令”弹框，分别单击“新增输入参数”和“新增响应参数”，填写相关信息后，单击“确认”。

图 2-43 新增命令参数-Beep



图 2-44 新增命令参数-Beep_State

新增参数

* 参数名称: Beep_State

参数描述:

* 数据类型: int(整型)

* 取值范围: 0 - 1

步长:

单位:

确定 取消

注册设备

本文介绍集成NB模组设备的注册方法。

- 步骤1** 在产品详情页面，选择“在线调试”，单击“新增测试设备”，此处新增的是非安全的NB-IoT设备。
- 步骤2** 在新增测试设备页面，选择“真实设备”，完成设备参数的填写后，单击“确定”。

图 2-45 在线调试-新增测试设备

新增测试设备

设备类型 真实设备 虚拟设备

* 设备名称: Test

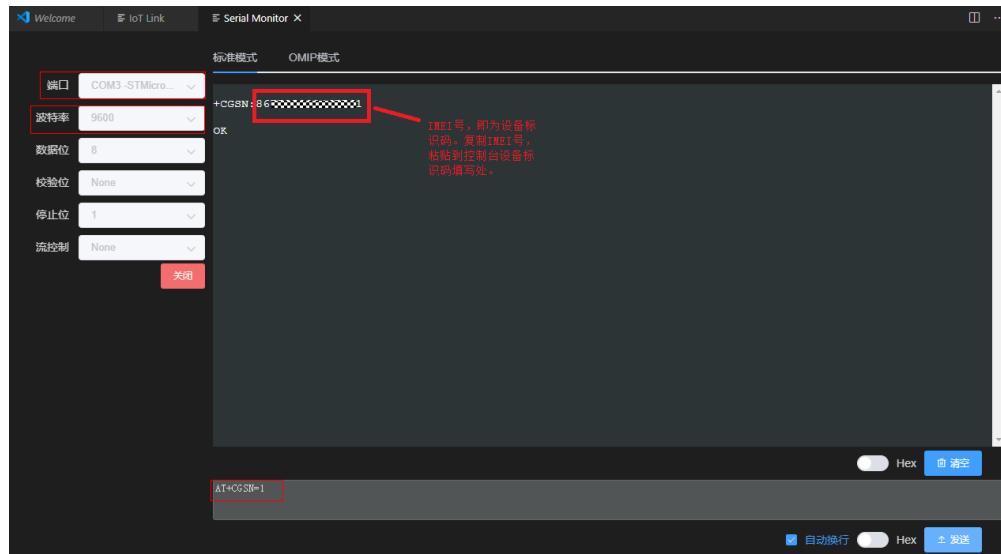
* 设备标识码: 86XXXXXXXXXX1

设备注册方式 不加密 加密

确定 取消

- 设备名称：自定义。
- 设备标识码：设备的IMEI号，用于设备在接入物联网平台时携带该标识信息完成接入鉴权，可在NB模组上查看。您也可以将拨测开关拨到**AT-PC模式**，选择STM的端口，波特率设置为9600，输入指令“AT+CGSN=1”获取IMEI号。

注：获取IMEI注册完设备后，需要将开发板的拨测开关拨到“AT-MCU”模式，因为开发板在MCU模式下才会通过NB卡连接网络。



- 设备注册方式：不加密。

步骤3 设备创建成功，可在页面看到创建的设备。

----结束

数据上报

平台和开发板建立连接后，上报烟雾浓度值。

步骤1 登录**设备接入服务控制台**，选择“设备 > 所有设备”。

步骤2 选择注册的设备，单击“查看”进入设备详情，查看上报到平台的数据。

图 2-46 查看上报数据-Smoke

The screenshot shows the 'Device Details' page for a device named 'Smoke'. At the top, there are sections for '所属资源空间' (Resource Space) and '设备标识码' (Device Identifier). On the right, there are sections for '设备ID' (Device ID), '认证类型' (Authentication Type), '节点类型' (Node Type), and '软件版本' (Software Version). Below this, the '最新上报数据' (Latest Reported Data) section displays a table with a single row for 'Smoke_Value' (Value: 2197). There are also search bars for '请输入服务名称' (Enter service name) and '请输入属性名称' (Enter attribute name).

----结束

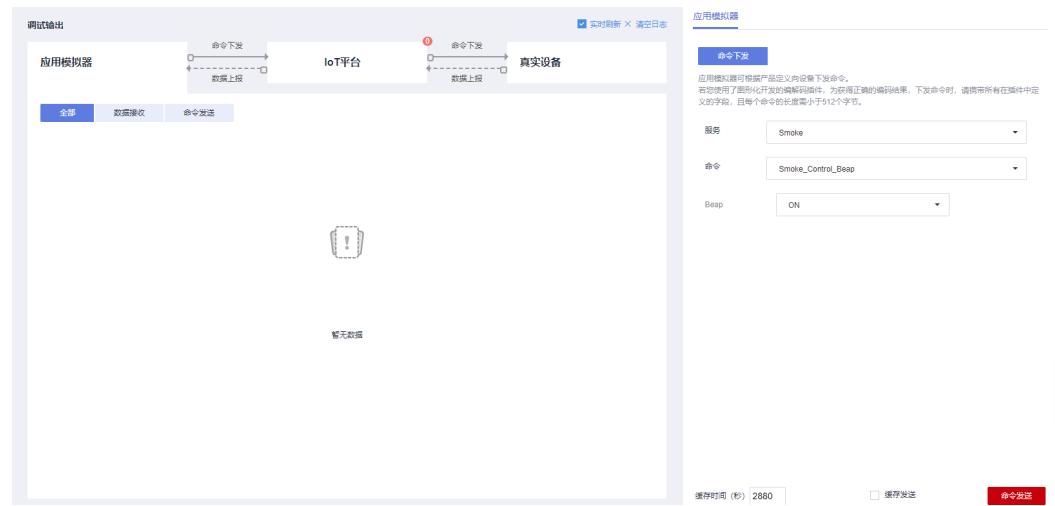
命令下发

步骤1 登录[设备接入服务控制台](#)，选择创建的产品，单击产品进入产品详情页。

步骤2 选择“在线调试”，单击注册的设备进入调试页面。

步骤3 设置命令参数后，单击“命令发送”。

图 2-47 命令下发调试-Smoke



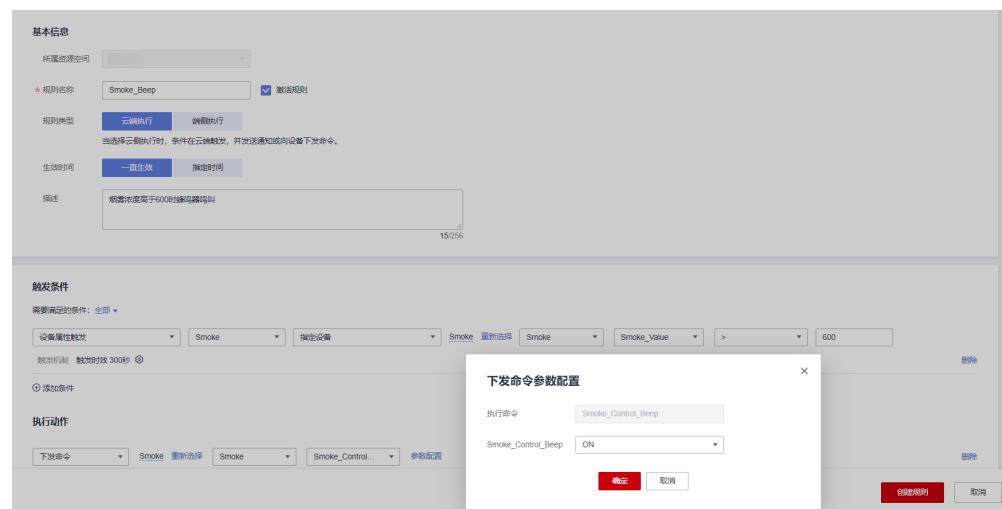
----结束

(可选) 配置设备联动规则

步骤1 选择左侧导航栏的“规则 > 设备联动”，单击页面右上角的“创建规则”。

步骤2 参考下表参数说明，创建设备联动规则。

图 2-48 新建联动规则-Smoke_Beep



| 参数名称 | 说明 |
|------|---|
| 规则名称 | 创建的规则名称，例如“Smoke_Beep”。 |
| 激活规则 | 勾选“激活规则” |
| 生效时间 | 选择“一直生效”。 |
| 描述 | 对该规则的描述，如“烟雾浓度高于600时蜂鸣器鸣叫”。 |
| 触发条件 | 1. 单击“添加条件”； 2. 在下拉框中选择“设备属性触发”，选择 注册设备 中添加的设备； 3. “服务类型”选择“smoke”，“属性”选择“Smoke_Value”，“操作”选择“>”，“值”填写“600”，单击“触发机制”，“触发策略”选择“重复抑制”，“数据时效”填写“300”，然后单击“确认”。 |
| 执行动作 | 1. 单击“添加动作”； 2. “动作类型”选择“下发命令”，下发设备选择 注册设备 创建的设备； 3. “服务类型”选择“smoke”，“命令”选择“Smoke_Control_Beep”，单击“参数配置”，“Beep”的值选择“ON”，然后单击“确认”。 |

----结束

(可选) 验证操作

将降温剂喷在烟雾传感器周围，模拟烟雾浓度。当烟雾浓度大于600时，蜂鸣器鸣叫。

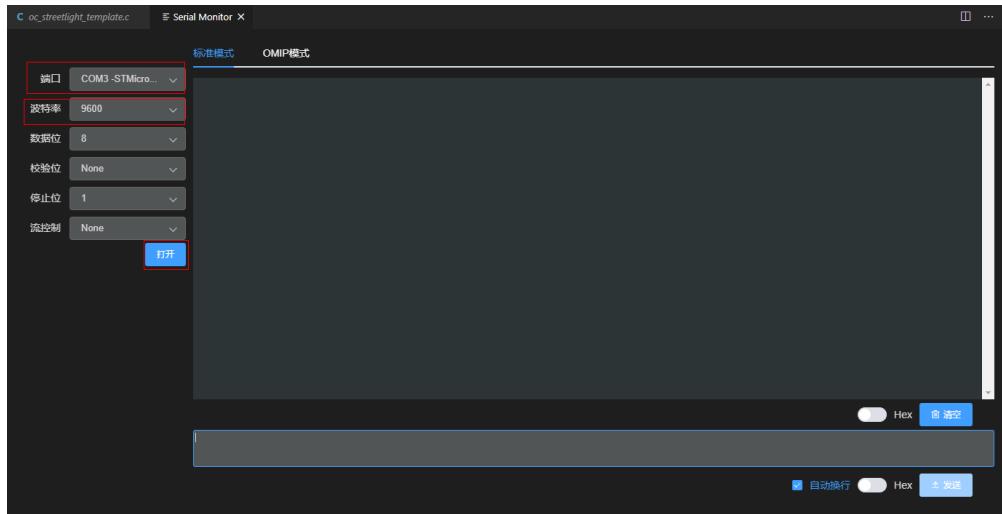
使用 AT 指令定位模组通信问题

IoT Link在与物联网平台连接使用时，可使用AT指令快速定位模组与云端连通性问题，提高开发效率。本节将以“小熊派开发板”为例，介绍如何使用AT指令检测通信模组常见问题，如设备无法上线，数据上报不成功等。

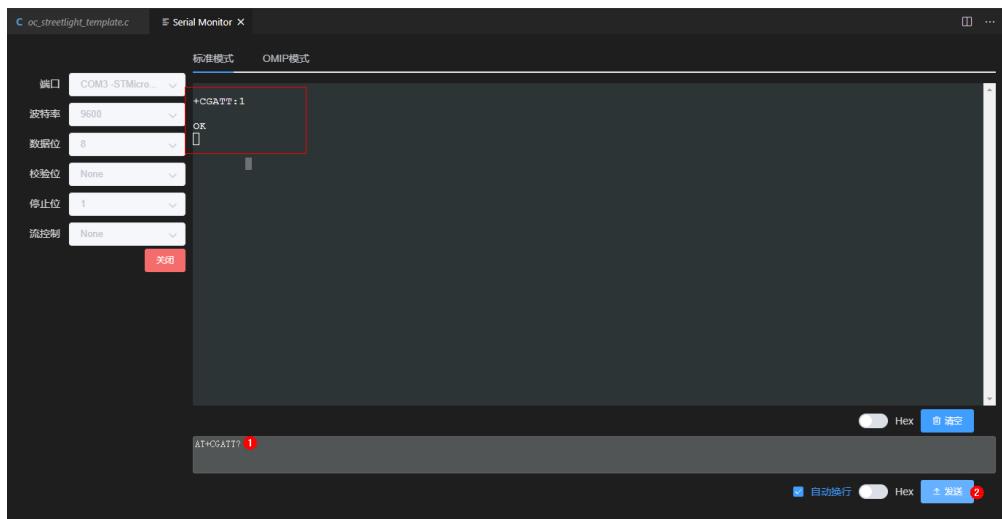
1. 小熊派开发板和电脑已正常连接（确保驱动已安装），并将开发板右上角的拨测开关切换到AT-PC模式。
2. 单击VSCode底部工具栏的“Serial”。



3. 选择[步骤4.2](#)中查看的端口号，波特率设置为9600，然后单击“打开”。



4. 输入“AT+CGATT?”，然后单击“发送”，若返回“+CGATT:1”表示网络附着成功（附着成功代表NB-IoT联网正常），返回“+CGATT:0”表示网络附着失败（附着失败代表NB-IoT联网异常），请查看SIM卡是否插入正确，或联系运营商检查网络状态。



□ 说明

使用AT指令检测完模组通信后，请将拨测开关拨到**AT-MCU**模式，以便完成控制台的开发后，将采集到的传感器数据通过通信模组发送到平台。

PC模式是开发板与电脑串口通信，AT指令读写开发板的状态等数据；MCU模式是开发板通过模组上插的SIM连接网络，实现NB-IoT通信。

2.4 设备模拟器快速接入

本文档以设备接入模拟器为例，介绍以MQTT原生协议接入物联网平台。此模拟器是MQTT客户端，可以快速验证是否可以与物联网平台服务交互发布或订阅消息。

前提条件

- 已注册华为云官方帐号。未注册可点击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上点击[实名认证](#)完成认证，否则无法使用设备接入功能。

- 已开通设备接入服务。未开通则访问[设备接入服务](#)，点击“立即使用”后开通该服务。

获取设备接入信息

在设备接入服务控制台获取设备接入平台所需的信息。

- 步骤1** 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 步骤2** 单击“立即使用”进入控制台，单击左侧导航栏的“总览”，点击“实例基本信息 - 接入信息”查看设备接入信息，记录域名和端口。

图 2-49 获取接入信息

The screenshot shows the 'Device Access' service control console. On the left, there's a navigation bar with 'Standard Edition' selected. The main area has a 'Overview' tab selected. In the center, there's a 'Data Statistics' section showing 25 current devices. To the right, there's a detailed 'Access Information' section with tables for '接入...' (Access), '接入协议 (端口号)' (Access Protocol (Port Number)), '接入地址' (Access Address), and '自定义' (Custom). Below these tables is a chart titled '注册设备数' (Number of Registered Devices) showing data from 12/28 to 12/29. At the bottom, there's a 'Related Documentation/Resources' section.

说明

针对不支持用域名接入的设备，通过在cmd命令框中执行“ping 域名”获取IP地址，用IP地址接入平台。由于IP地址不固定，您需要将IP地址做成可配置项。

结束

创建产品

- 步骤1** 创建MQTT协议产品（如果已有MQTT协议产品，可跳过此步骤）。
- 步骤2** 登录[管理控制台](#)，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

图 2-50 产品-创建产品

The screenshot shows the 'Product Creation' interface in the management console. On the left, there's a navigation bar with 'Standard Edition' selected. The main area has a 'Product Introduction' section with a 'Business Inquiry' button. Below it is a 'Function Introduction' section with a diagram illustrating the product creation process: 1. Define Profile, 2. Register Device, 3. Device Development, and 4. Online Testing. There's also a search bar with '自有产品' (Own Product) highlighted. The bottom of the screen shows a pagination bar with '10' and '1'.

步骤3 创建一个协议类型为MQTT协议、设备类型为StreetLamp的产品，参考页面提示填写参数后，单击“确定”。

图 2-51 创建产品-MQTT

所属资源空间

产品名称

协议类型

数据格式

设备类型选择

设备类型

高级配置

产品ID

产品描述

0/128

确定 取消

----结束

注册设备

步骤1 在设备接入控制台页面，选择左侧导航栏“设备 > 所有设备”，单击页面右上角的“注册设备”。

图 2-52 所有设备-注册设备

步骤2 根据页面提示信息填写参数，然后单击“确定”。

| 参数名称 | 说明 |
|--------|-----------------------------------|
| 所属资源空间 | 确保和所属产品归属在同一个资源空间。 |
| 所属产品 | 选择对应产品。 |
| 设备标识码 | 即nodeID，设备唯一物理标识。可自定义，由英文字母和数字组成。 |
| 设备名称 | 即device_name，可自定义。 |
| 设备认证类型 | 选择“密钥”。 |
| 密钥 | 此处如不填写，物联网平台会自动生成。 |

图 2-53 单设备注册-MQTT

单设备注册

所属资源空间 (②)

所属产品 Test_1

MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

设备标识码 (②) Test_1

设备名称

设备ID (②)

设备描述 0/2,048

设备认证类型 (②) 密钥 X.509证书

密钥

确认密钥

确定 取消

步骤3 成功注册设备后，平台会自动生成设备ID和密钥，请妥善保管好设备ID（deviceId）和密钥（deviceSecret），用于设备接入。

图 2-54 注册设备成功



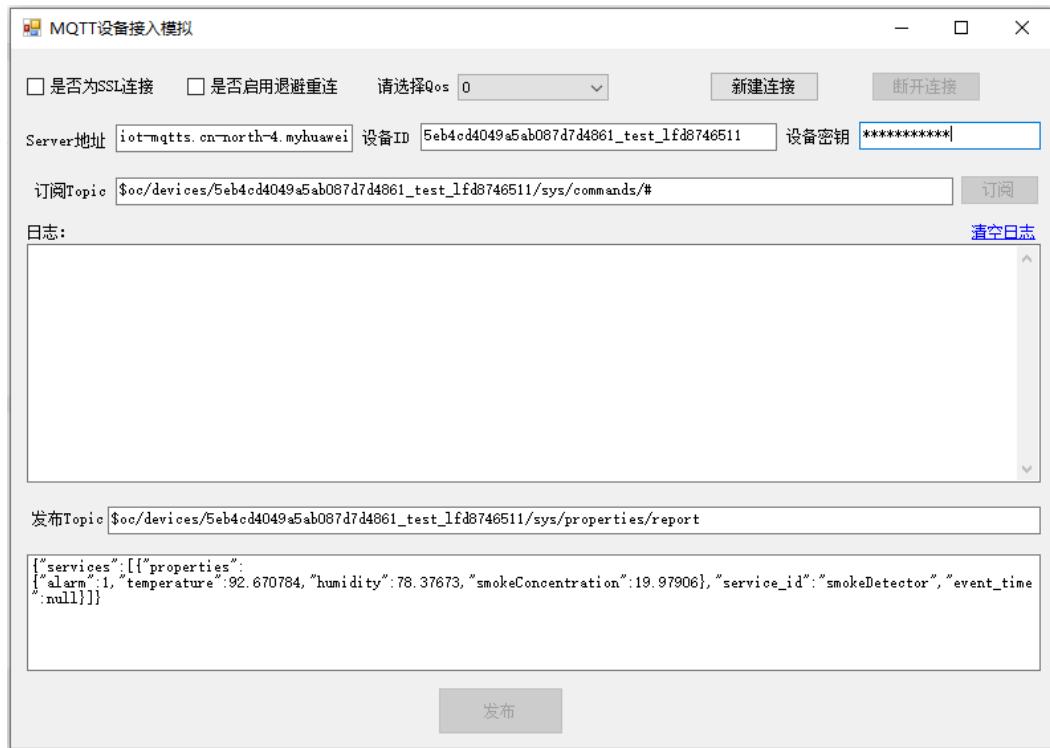
----结束

使用模拟器接入

步骤1 下载[模拟器](#)（默认是64位操作系统），并启动模拟器，如下图：



步骤2 界面展示



1. 模拟器提供了界面显示，填写Server地址、设备ID、设备密钥。请根据实际创建的设备信息填写。
 - Server地址：即域名，参考[平台对接信息](#)获取；
 - 设备ID和设备密钥：在[这里](#)获取。
2. 不同的Server地址SSL加密接入时要与对应的[证书文件匹配使用](#)，证书获取请参考[获取资源](#)，需要替换certificate文件夹下的证书，如下图：



3. 用户可以选择设备侧建链时是否为SSL加密，选择Qos方式是0还是1，当前不支持Qos2，可参考[使用限制](#)。

步骤3 新建连接

设备或网关在接入物联网平台时首先需要和平台建立连接，从而将设备或网关与平台进行关联。开发者通过传入设备信息，将设备或网关连接到物联网平台。点击新建连接按钮，域名、设备ID和秘钥正确的情况下，可以看到日志设备连接成功，可在平台查看设备状态，如下图：

图 2-55 设备在线



步骤4 订阅Topic

订阅某topic的设备才能接收broker发布的关于该topic的消息，关于平台预置topic可参考[Topic定义](#)。

建链后，如果成功订阅Topic，主界面日志栏显示如下信息：

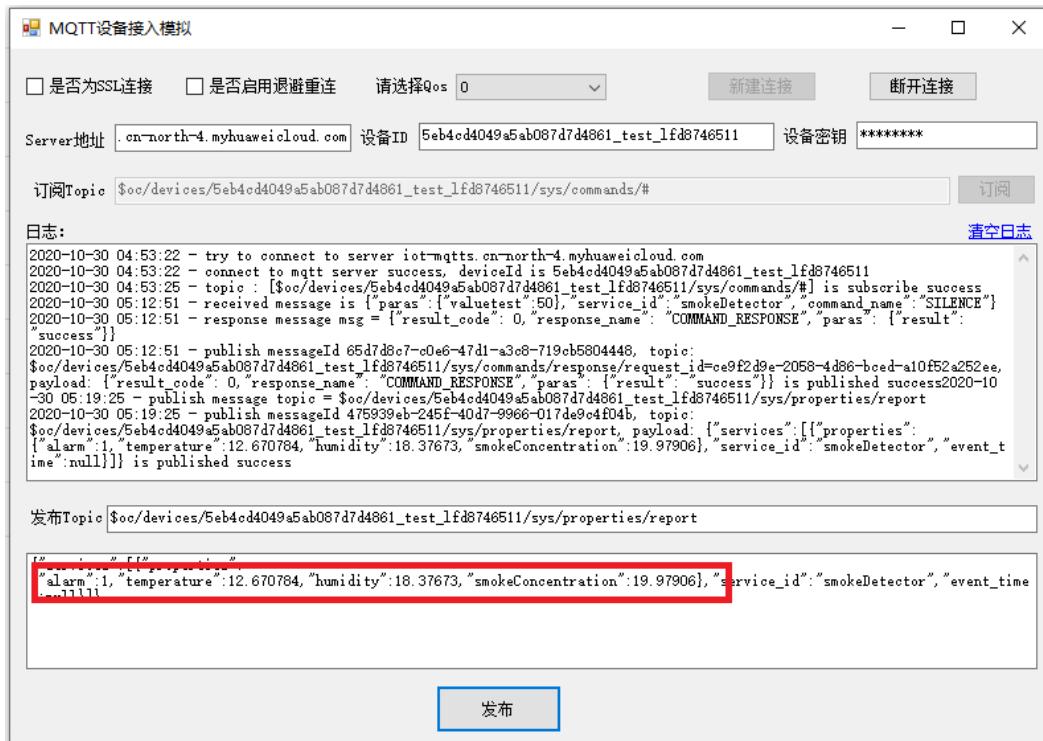


步骤5 发布Topic

发布Topic是指设备主动向平台上报自己的属性或消息，详细见[设备属性上报接口文档](#)。

在模拟器中实现了上报Topic、属性上报功能。

发布Topic后，Demo界面显示如下：



设备上报属性成功后可在“设备详情”页面查看到上报的属性：

图 2-56 查看上报数据-MQTT

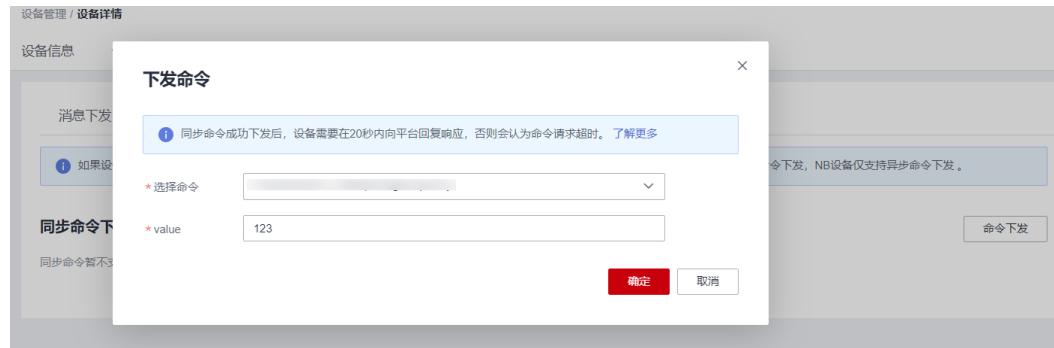


步骤6 接收下发命令

模拟器支持接收平台下发命令的功能，在MQTT建链完成并成功订阅Topic后，可以在**控制台**设备详情中对该设备进行同步命令下发。下发成功后，在MQTT的回调函数中接收到平台下发给设备的命令。

例如下发参数名为smokeDetector: SILENCE，参数值为50的命令。

图 2-57 命令下发-smokeDetector



同步命令下发成功后，Demo界面显示如下：



----结束

2.5 通过协议转换网关实现泛协议设备接入

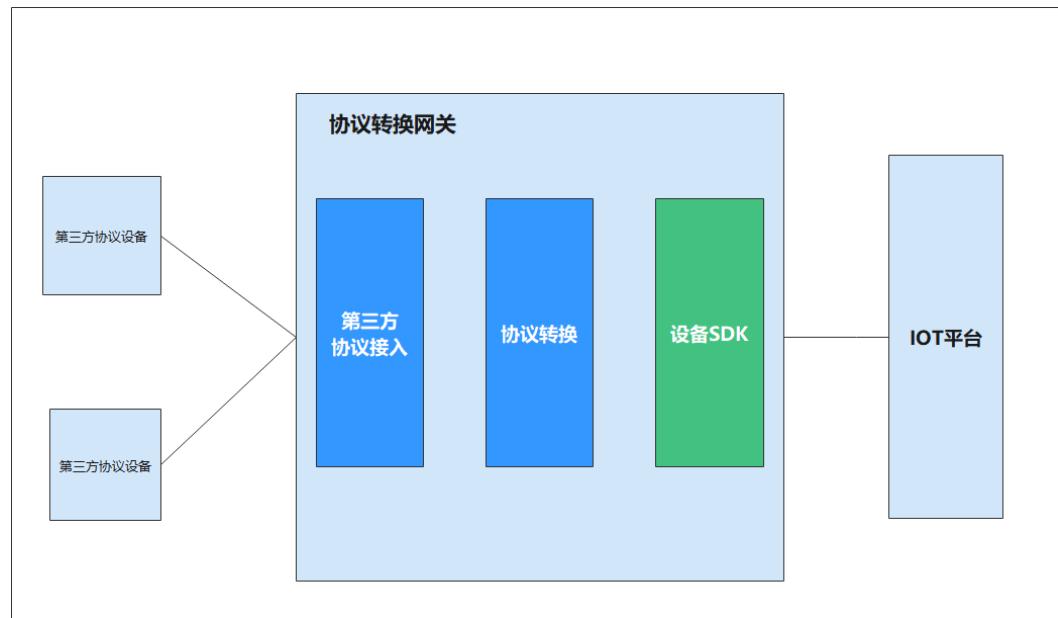
场景说明

目前平台只支持MQTT/HTTP/LWM2M等标准协议接入，如果设备是其他协议（统称为第三方协议），怎么接入平台？

由于第三方协议设备不能直接接入平台，需要在平台外部完成协议转换。我们推荐使用网关来完成协议转换，将第三方协议转成MQTT协议。我们把用来进行协议转换的网关叫做协议转换网关。

实现原理

方案总体架构框图如下：

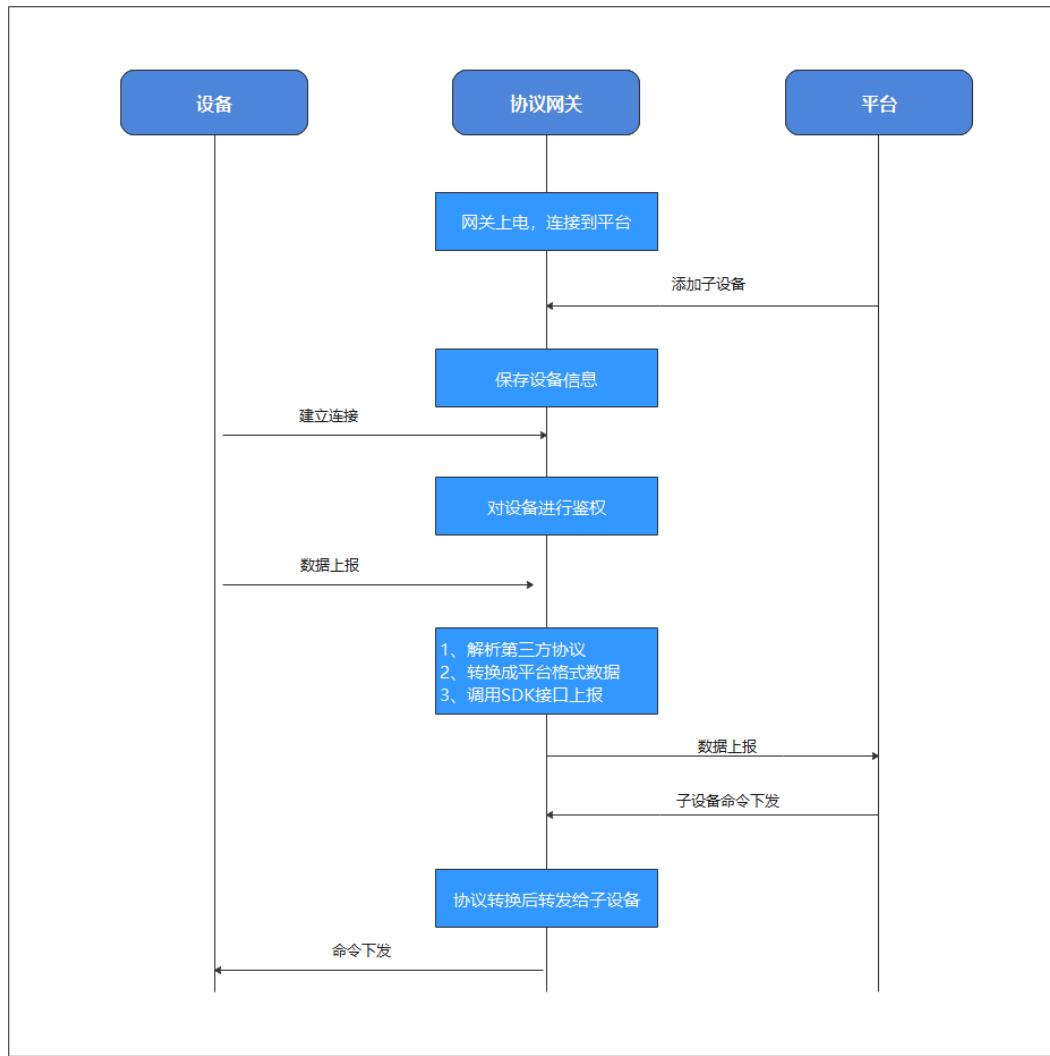


协议转换网关是一个网关，可以部署在云上或者本地。第三方协议设备作为协议转换网关的子设备接入平台。

协议转换网关一般由三部分组成：

1. 第三方协议接入。完成第三方协议的解析，鉴权。
2. 协议转换。负责完成第三方协议数据和平台格式数据的互相转换。
 - 上行：把第三方协议数据转成平台格式数据，并调用设备SDK接口进行上报。
 - 下行：收到平台下行数据时，转换为第三方协议数据转发给第三方协议设备。
3. 设备SDK。即平台提供的设备接入SDK，提供了网关的通用功能实现，用户可以在此基础上实现自己的网关。

业务流程



1. 在物联网平台上注册网关，详细方法请参考[设备注册](#)。
2. 网关上电，连接到平台，连接所需的鉴权参数在[注册网关](#)时获取。
3. 用户在平台上注册子设备时，平台下发添加子设备事件到网关。网关收到后，保存子设备信息到本地并持久化（SDK提供了默认的持久化实现，用户可以自定义扩展）。
4. 第三方协议设备连接到网关。网关根据子设备信息对设备进行鉴权。
5. 设备上报数据到网关。网关转换为平台格式数据后，调用SDK的上报子设备属性/消息的接口上报给平台。
6. 平台向设备下发命令。网关收到后，转换为第三方协议，转发给子设备。设备收到后对命令进行处理。

协议转换网关的具体实现

网关具体实现和使用方法参见 [Java版本](#)、[C版本](#)。

2.6 恒温空调

场景说明

通过恒温控制系统，不论空调是否开机，都可以调整空调默认温度，待空调上电开机后，自动按默认温度调节。空调接入到物联网平台后，用户可以在应用侧或者设备接入控制台设置设备影子，将预置的温度通过设备影子下达属性修改给空调。空调收到修改属性的要求后，自动调节温度。

恒温空调产品开发

- 步骤1** 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 步骤2** 选择左侧导航栏的“产品”，单击右上角下拉框，选择新建产品所属的资源空间。
- 步骤3** 单击左侧的“创建产品”，创建恒温空调产品，填写参数后，单击“确定”。

| 基本信息 | |
|------|-------------------|
| 产品名称 | 自定义，如aircondition |
| 协议类型 | 选择“MQTT” |
| 数据格式 | 选择“JSON” |
| 所属行业 | 自定义 |
| 设备类型 | |

- 步骤4** 产品创建成功后，单击对应的产品进入产品详情页面。
- 步骤5** 在模型定义页面，单击“自定义模型”，参考下表，完成产品模型配置。

| 服务数据 | |
|------|--|
| 服务 | 服务ID：temperature 服务类型：建议和服务ID保持一致 |
| 属性 | 属性名称：temperature 数据类型：int 访问权限：可读、可写 长度：1 |

- 步骤6** 进入“设备 > 设备注册”页面，单击“注册设备”，参考下表填写参数。

图 2-58 单设备注册-MQTT

The screenshot shows the 'Single Device Registration' dialog box. It includes fields for '所属资源空间' (selected), '所属产品' (Test_1), '设备标识码' (Test_1), '设备名称' (Test_1), '设备ID' (Test_1), and a large '设备描述' (Device Description) field with a character limit of 0/2,048. Under '设备认证类型' (Device Authentication Type), '密钥' (Key) is selected. Below are fields for '密钥' and '确认密钥' (Confirm Key). At the bottom are '确定' (Confirm) and '取消' (Cancel) buttons.

| 参数名称 | 说明 |
|--------|--|
| 所属产品 | 选择在步骤 步骤3 中创建的产品。 |
| 设备标识码 | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |
| 设备名称 | 自定义。 |
| 设备认证类型 | 选择“密钥”。 |
| 密钥 | 设备密钥，可自定义，不填写物联网平台会自动生成。 |

----结束

配置设备影子

用户可以选择在应用侧调用[配置设备影子预期数据接口](#)设置设备影子，也可以在设备接入控制台设置设备影子。此处以在设备接入控制台设置设备影子为例。

步骤1 登录[控制台](#)，选择左侧导航栏的“设备”，单击**步骤6**注册的设备进入到恒温空调设备的详情页面。

步骤2 选择“设备影子”页签，单击“属性配置”。

步骤3 在弹出窗口中输入服务属性对应的期望值。此处设置“temperature”的属性值为25。

图 2-59 配置设备影子



----结束

验证操作

方法一：

您可以使用MQTT.fx模拟设备验证。

1. 使用MQTT.fx模拟恒温空调，并连接到物联网平台。操作方法请参考[在线开发 MQTT协议的智慧路灯](#)。
2. 选择“Subscribe”页签，输入订阅设备影子的topic=\$oc/devices/{device_id}/sys/shadow/get/response/#，其中{device_id}与步骤6的设备ID保持一致，单击“Subscribe”按钮。



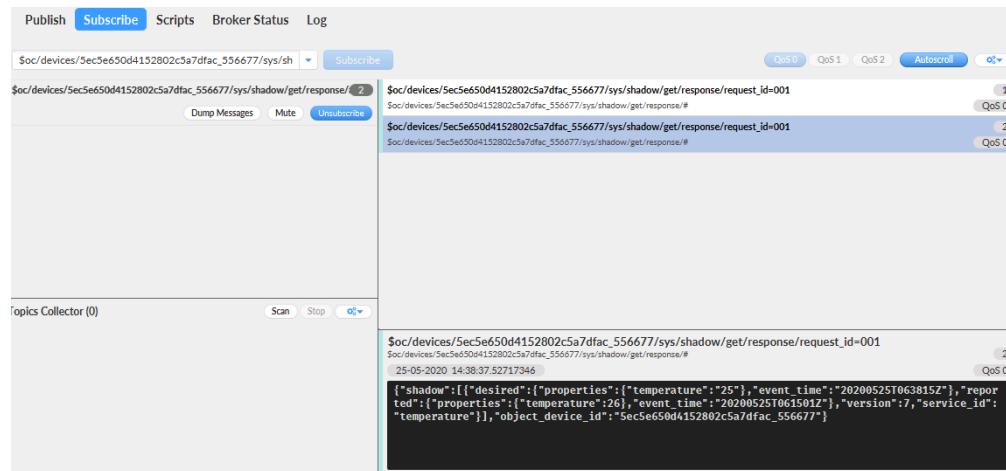
3. 选择“Publish”页签，输入请求设备影子的Topic=\$oc/devices/{device_id}/sys/shadow/get/request_id={request_id}。
4. 输入一条获取设备影子的请求，单击“Publish”按钮。

示例：

```
{  
    "object_device_id": "40fe3542-f4cc-4b6a-98c3-61a49ba1acd4",  
    "service_id": "temperature"  
}
```



5. 选择“Subscribe”页签，看到平台下发的设备影子数据。



方法二：

您可以使用配置设备接入服务时注册的真实设备接入平台，设备会收到平台下发的设备影子，修改空调的预设温度值。

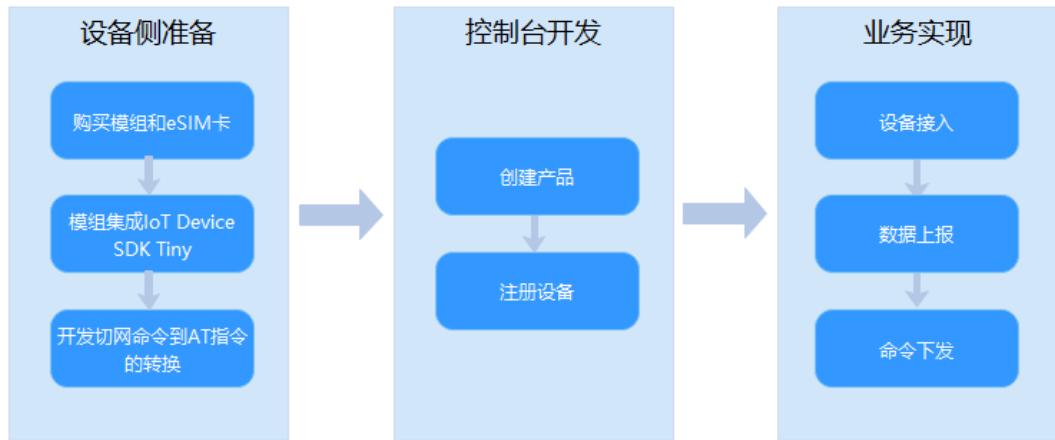
2.7 设备按需选网

场景说明

使用全球SIM联接服务的设备接入到物联网平台时，可能会面临网络信号强度弱，信号有时延等问题。当应用服务器接收到设备上报的网络信号强度数据时，您可以通过命令下发的方式切换运营商网络。

整体流程

按需选网整体流程如下图所示。



设备在集成SDK Tiny后，还需要完成切换网络命令转换成AT指令的开发，以完成设备网络的自动切换。

当设备接入到物联网平台时，便可实现数据上报，命令下发。

购买模组和 eSIM 卡

- 访问[全球SIM联接服务](#)，单击“购买eSIM”，购买模组和eSIM卡。
模组型号选择“移远EC20 CEHDLG”。
- 登录[全球SIM联接控制台](#)，订购套餐，您可以选择单卡订购或者批量订购。
 - 单卡订购：选择左侧导航栏的“SIM卡管理”，单击某个SIM卡右侧的“服务管理”，然后选择套餐服务并单击“立即订购”。
 - 批量订购：选择左侧导航栏的“SIM卡管理”，导出需要订购套餐的SIM卡，下载批量订购的模板，编辑完批量订购文档后，上传，完成批量套餐订购。

集成 IoT Device SDK Tiny

参考[IoT Device SDK Tiny使用指南](#)，完成设备侧的开发。

开发切网命令转换成 AT 指令

设备支持自动切网功能，需要进行设备侧开发。

- 设备收到平台下发的切换网络命令如下：

```
{  
    "paras": [  
        "iccid": "xxxx"  
    ],  
    "service_id": "MobileConnection",  
    "command_name": "SWITCH_NETWORK"  
}
```

注：xxxx为SIM卡标识，其他内容固定不变。

- 设备收到命令后，执行切网指令：

```
AT+HWICCIDENABLE=xxxx
```

注：xxxx为SIM卡标识。

创建产品

- 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。

2. 选择左侧导航栏的“产品”，单击左侧的“创建产品”，参考下表填写产品的基本信息。

| 基本信息 | |
|--------|-----------------------------------|
| 所属资源空间 | 在下拉框选择所属资源空间。 |
| 产品名称 | 自定义填写，如SwitchNetwork |
| 协议类型 | MQTT |
| 数据格式 | JSON |
| 所属行业 | 在下拉框选择相应的行业。 |
| 设备类型 | 使用平台预置的产品模型，会自动关联设备类型，不需要再输入设备类型。 |

3. 进入产品详情，点击导入库模型，在标准模型列表中选择“MobileDevice”，点击确定

| MobileDevice模型中MobileConnection服务的属性说明 | | |
|--|---------------------------|--------------------|
| 属性名称 | 描述 | 示例内容 |
| iccid | SIM卡标识 | 898604421920C0XXXX |
| imsi | 国际移动用户识别码 | 460046240XXXXXX |
| imei | 国际移动设备识别码 | 86761100XXXXXXXX |
| operator | 移动设备的网络运营商 (MCC+MNC) | 46000 |
| rssi | 接收的信号强度 | -40 |
| cellId | 蜂窝小区ID | 1106525 |
| lac | 位置区码 | 12345 |
| networkType | 网络类型 | 2G,3G,4G,5G,NB-IoT |
| modelType | 模组型号 | EC20 CEHDLG |

注册设备

1. 在左侧导航栏，单击“设备 > 所有设备”。
2. 单击右上角“注册设备”，按照如下表格填写参数后，单击“确定”，完成设备的注册。

| 参数名称 | 说明 |
|--------|-------------------------------|
| 所属资源空间 | 在下拉框选择所属资源空间，与步骤3选择的资源空间保持一致。 |

| 参数名称 | 说明 |
|--------|--|
| 所属产品 | 选择步骤3中创建的产品。 |
| 设备标识码 | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |
| 设备名称 | 自定义。 |
| 设备认证类型 | 选择“密钥”。 |
| 密钥 | 设备密钥，可自定义，不填写物联网平台会自动生成。 |

设备接入平台

方法一：

您可以使用MQTT.fx模拟设备接入平台。操作方法请参考[在线开发MQTT协议的智慧路灯](#)。

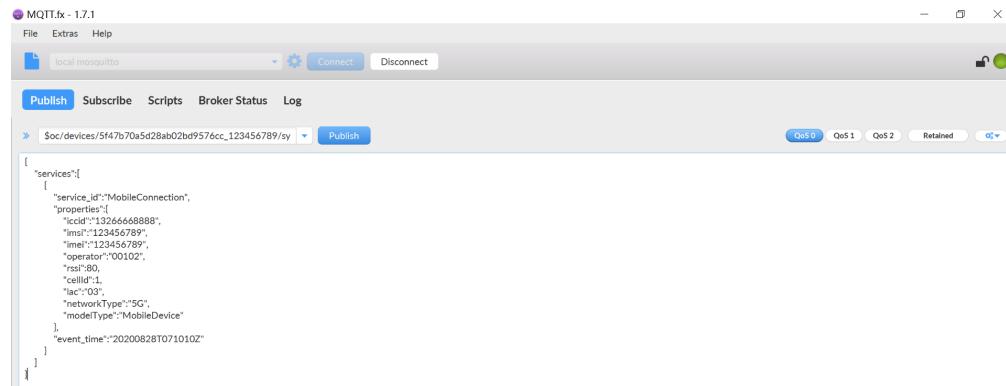
方法二：

您可以使用配置设备接入服务时注册的真实设备接入平台。

属性上报

本文使用MQTT.fx模拟设备上报网络信号强度等数据。

1. 选择Publish页签，输入设备属性上报的topic “\$oc/devices/{device_id}/sys/properties/report”，其中{device_id}与注册设备时获取的设备ID保持一致，您可以使用真实设备上报属性。在下方填写上报的属性，单击Publish按钮。具体可参考[设备属性上报](#)。



2. 在控制台“设备列表”中找到该设备，单击“查看”，进入设备详情页。在设备详情页看到最新上报的数据。您也可以调用API查看最新上报数据，详细请参考[查询设备影子数据](#)。

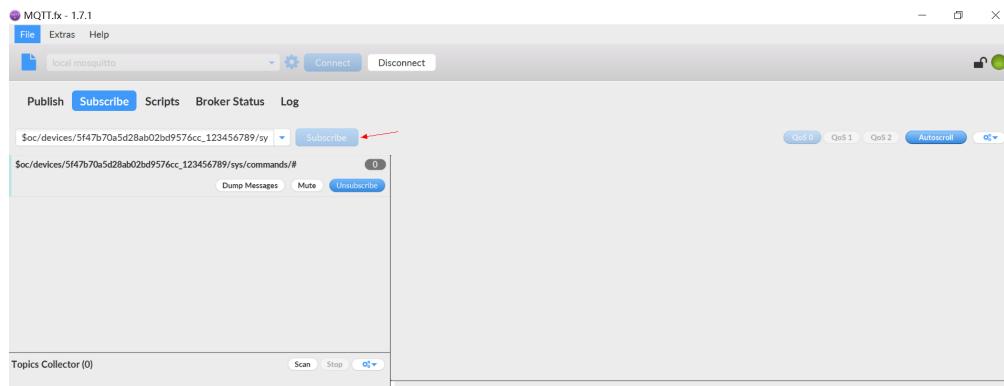
图 2-60 查看上报数据-MQTT

The screenshot shows the 'Device Management / Device Details' page for a device named 'Test_1'. The top navigation bar includes tabs for '设备信息' (Device Information), '云端运行日志' (Cloud Operation Log), '云端下发' (Cloud Push), '设备影子' (Device Shadow), '消息跟踪' (Message Tracking), '设备监控' (Device Monitoring), '子设备' (Sub-device), '标签' (Tags), and '群组' (Groups). The main content area displays device information such as '所属资源空间' (Resource Space), '设备标识码' (Device Identifier), '注册时间' (Registration Time), '固件版本' (Firmware Version), '设备描述' (Device Description), '最近离线时间' (Last Offline Time), '设备ID' (Device ID), '认证类型' (Authentication Type), '节点类型' (Node Type), '软件版本' (Software Version), '激活时间' (Activation Time), and 'MQTT连接参数' (MQTT Connection Parameters). Below this, there is a section titled '物模型数据' (Thing Model Data) which lists properties like 'luminance' (光亮度) with a value of '30'. A red box highlights this section.

下发切换网络的命令

本文使用MQTT.fx模拟设备接收命令。

- 在MQTT.fx，选择Subscribe页签，输入订阅命令的topic “\$oc/devices/{device_id}/sys/commands/#”，单击“Subscribe”。更多命令下发格式请参考[命令topic订阅](#)。（您可以使用真实设备订阅topic）

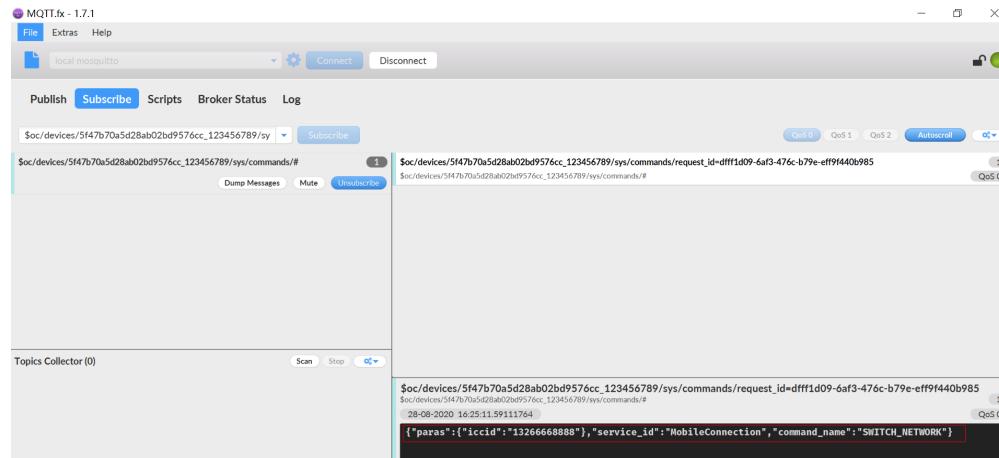


- 在控制台设备详情页，单击“命令”页签，下发同步命令。您也可以调用API下发命令，详细请参考[同步命令下发](#)。

图 2-61 命令下发-MQTT

The screenshot shows the 'Device Management / Device Details' page. The left sidebar has tabs for '基础版' (Basic Edition) and '切换' (Switch). Under '基础版', there are sections for '启停' (Start/Stop), '产品' (Product), '设备' (Device), and '所有设备' (All Devices). The top navigation bar includes tabs for '设备信息' (Device Information), '云端运行日志' (Cloud Operation Log), '云端下发' (Cloud Push), '设备影子' (Device Shadow), '消息跟踪' (Message Tracking), '设备监控' (Device Monitoring), '子设备' (Sub-device), '标签' (Tags), and '群组' (Groups). The main content area has a '消息下发' (Message Push) tab and a '命令下发' (Command Push) tab, which is currently selected. A note states: '如果设备所属产品定义了命令功能，则您可以通过应用调用平台接口或者操作下面的“下发命令”按钮下发命令。当前MQTT设备仅支持同步命令下发，NB设备仅支持异步命令下发。' Below this is a '同步命令下发' (Sync Command Push) section with a '命令下发' (Send Command) button.

- 在MQTT.fx的Subscribe页，接收到平台下发的命令。



4. 使用串口工具发送AT指令完成网络切换。

a. 安装USB驱动：

- 运行exe文件，根据界面提示进行安装。

说明

不同厂家匹配的USB驱动版本不同，请联系厂家获取符合要求的驱动程序。

- 驱动安装成功后，连接开发板的USB接口到PC，并打开电源，可在设备管理器中查看到枚举出的串口设备。



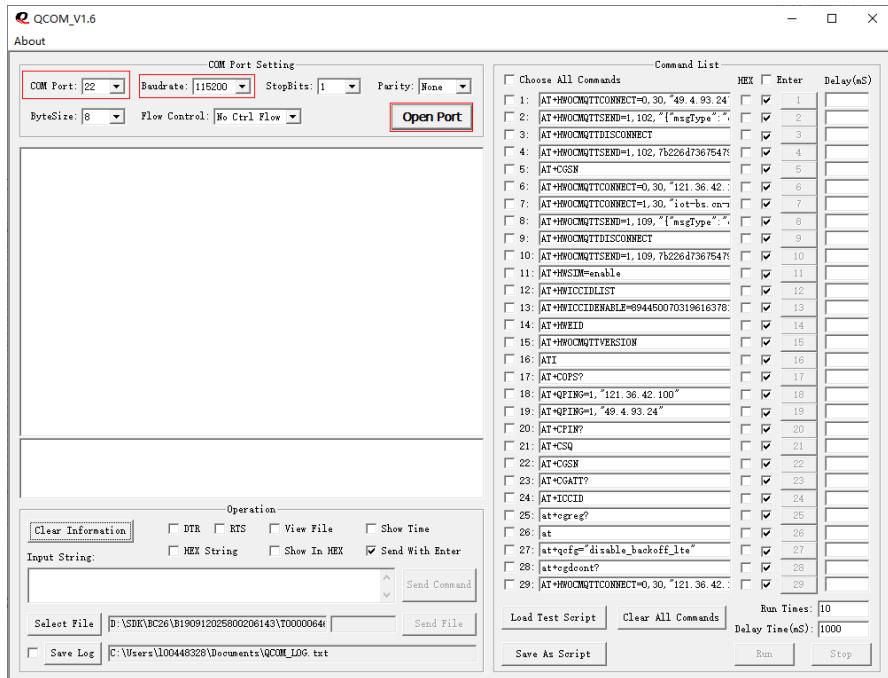
b. 使用串口工具切换网络。

- 运行exe文件，根据界面提示进行安装。

说明

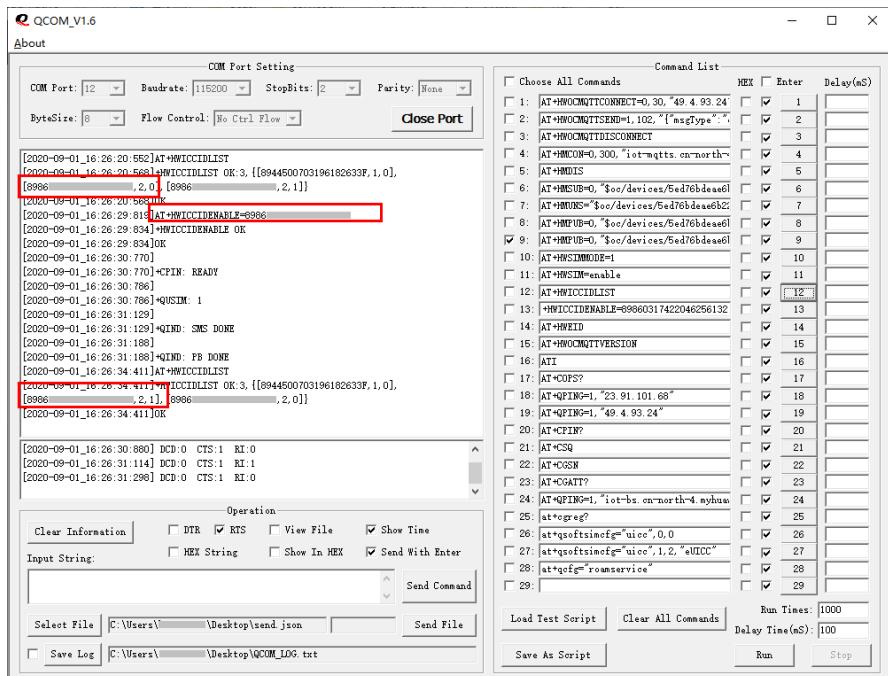
不同厂家匹配的串口工具版本不同，请联系厂家获取符合要求的串口工具。

- 打开串口工具，选择步骤2枚举的AT串口，波特率设置为115200，单击“Open Port”。



注：请确保设置正确，否则AT命令不能够被解析或者解析出错。

- iii. 输入“AT+HWICCIDENABLE”完成网络切换。切换成功后，您可以前往全球SIM联接服务控制台“SIM卡管理”，单击“服务详情”查看正在使用中的套餐。



2.8 设备自定义 TOPIC 迁移上云

场景说明

随着IoT终端设备规模不断增加，企业自建物联网面临服务器扩容，硬件投入和运维成本越来越高等诸多问题。为此，华为云物联网平台提供设备极简迁移上云解决方案，企业设备通过极少的改动，就可以快速将设备迁移到华为云物联网平台。

整体方案

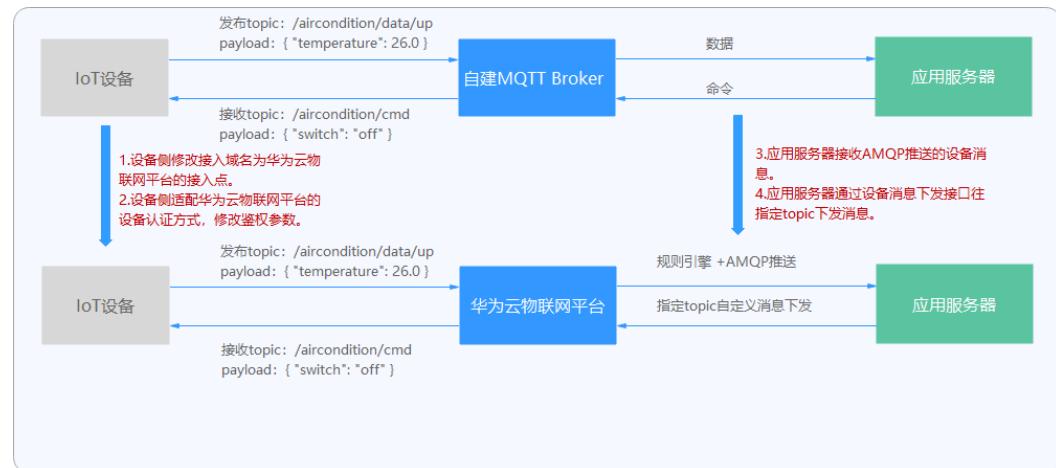
假设企业终端设备接入自建MQTT集群，业务架构如下图所示。



基于MQTT协议的上行数据和下行指令的业务定义如下：

| 业务场景 | 通信Topic | 报文Payload |
|---------|-----------------------|-------------------------|
| 设备上报数据 | /aircondition/data/up | { "temperature": 26.0 } |
| 服务端控制指令 | /aircondition/cmd | { "switch": "off" } |

为减少企业改造成本，华为云物联网平台提供如下迁移方案，设备侧不用改变原有的topic和payload报文格式，就可以快速迁移设备到华为云物联网平台。



企业设备迁移上云有三个核心变更点：

- 设备侧修改接入域名为华为云物联网平台的接入点。
- 配置规则引擎，把设备数据流转到应用服务器、AMQP消费组或华为云第三方云服务产品。

- 应用服务器适配设备消息下发接口往指定topic下发消息。

操作步骤

- 在控制台配置产品、设备和数据流转方案，请参见[云端配置开发](#)。
- 对设备端进行业务开发，请参见[设备端开发](#)。
- 对服务端进行业务开发，实现接收设备数据和下发控制指令，请参见[服务端开发](#)。

云端配置开发

步骤1 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。

步骤2 创建产品。

- 选择左侧导航栏的“产品”。
- 单击左侧的“创建产品”，创建待迁移设备的产品，填写参数后，单击“确定”。

| 基本信息 | |
|------|---------------------------|
| 产品名称 | 自定义，如aircondition |
| 协议类型 | 选择“MQTT” |
| 数据格式 | 选择“JSON” |
| 所属行业 | 无 |
| 设备类型 | 根据实际情况填写，如“aircondition”。 |

步骤3 注册设备。

- 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击页面右上角的“注册设备”。

图 2-62 单设备注册-aircondition

The screenshot shows the 'Single Device Registration' dialog box. It includes fields for '所属资源空间' (selected), '所属产品' (aircondition), '设备标识码' (12345678), '设备ID' (65af802e5b6d3c2d690a7138_12345678), and '设备名称' and '设备描述' (both empty). The '设备认证类型' section shows '密钥' selected over 'X.509证书'. Both '密钥' and '确认密钥' fields contain '*****'. At the bottom are '确定' (Confirm) and '取消' (Cancel) buttons.

| 参数名称 | 说明 |
|--------|--|
| 所属产品 | 选择在步骤2中创建的产品。 |
| 设备标识码 | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |
| 设备名称 | 自定义。 |
| 设备认证类型 | 根据企业设备现有的认证类型，选择对应的认证方式。本文以“密钥”方式为例。 |
| 密钥 | 设备密钥。填写待迁移设备的密钥。 |

2. 填写设备参数后，单击“确定”。

步骤4 创建数据转发规则。

1. 选择左侧导航栏的“规则 > 数据转发”，单击页面左测的“创建规则”。
2. 参考下表填写参数后，单击“创建规则”。

| 参数名 | 参数说明 |
|------|----------|
| 规则名称 | 创建的规则名称。 |

| 参数名 | 参数说明 |
|------|--|
| 规则描述 | 对该规则的描述。 |
| 数据来源 | 选择“设备消息”。设备迁移上云时，设备按原有的topic和payload进行上报，物联网平台默认按设备消息流程进行处理。 |
| 触发事件 | 选择数据来源后，自动匹配触发事件。 |
| 资源空间 | 您可以选择单个资源空间或所有资源空间。 |

- 在设置转发目标页面，单击“添加”，在弹出的页面中参考下表配置完参数后，单击“确认”。

| 参数名 | 参数说明 |
|------|--|
| 转发目标 | 选择数据转发目标，本文以“AMQP推送消息队列”为例。 |
| 消息队列 | <p>单击“选择”，选择消息队列。</p> <ul style="list-style-type: none">- 若没有消息队列，请新建消息队列，队列名称自定义且单个租户名下唯一，长度8-128，只能包含大写字母、小写字母、数字和指定特殊字符（如_-:）。- 若要删除消息队列，单击消息队列右侧的“删除”即可。 <p>说明 已经订阅的队列不允许删除。</p> |

- 完成完整的规则定义后，单击“启动规则”，实现数据转发至AMQP消息队列。

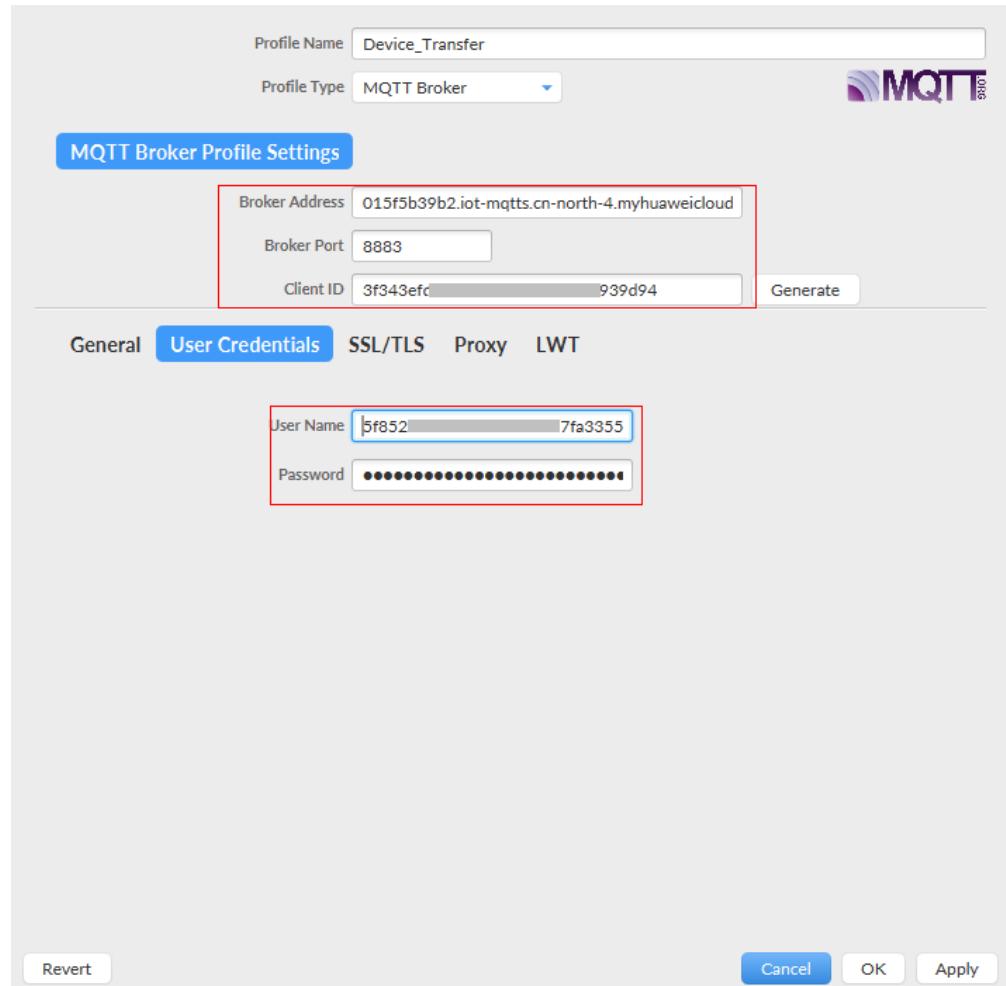
----结束

设备端开发

完成云端配置后，需要进行设备端业务开发。完整的设备开发流程可参考[设备侧开发](#)。本章节以MQTT.fx为例，介绍在设备迁移场景下，设备侧如何在尽量少改动的情况下，实现设备建立MQTT连接、数据上报、指令接收等功能。

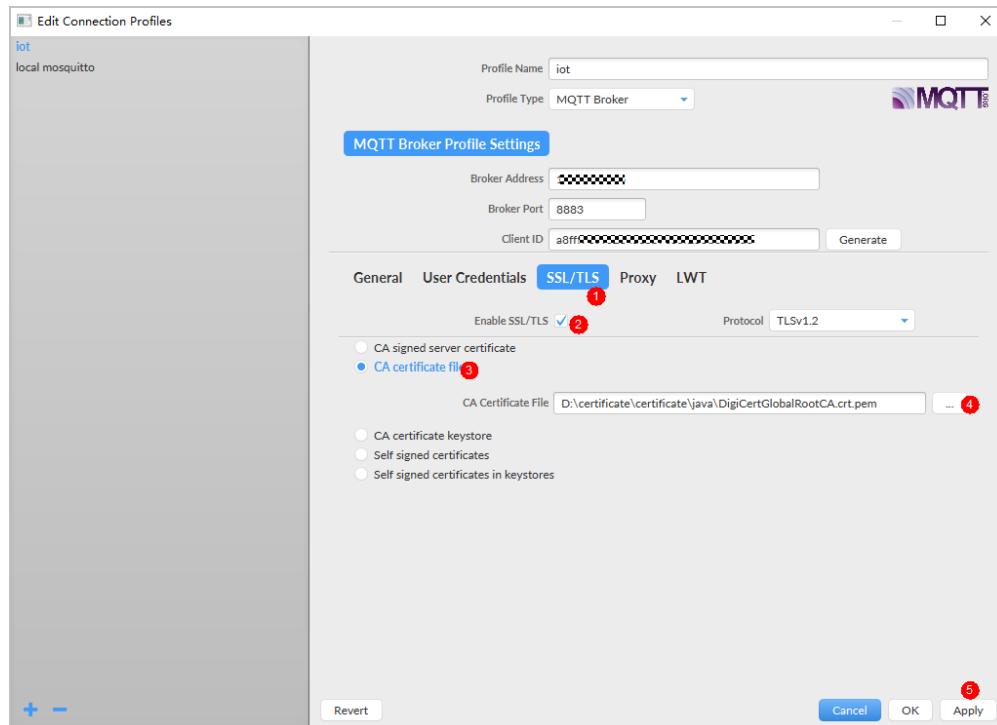
步骤1 设备同物联网平台建立MQTT连接。

- 参考下表配置鉴权参数。



| 参数 | 必选/可选 | 参数描述 |
|----------------|-------|--|
| Broker Address | 必选 | 华为云物联网平台的MQTT协议接入地址, 请参考 此处 获取。 |
| Broker Port | 必选 | 8883。若设备侧MQTT接入端口不是8883, 且无法修改, 可以开通 企业版实例 , 参考 此页面 配置MQTT协议接入端口。 |
| Client ID | 可选 | 使用设备迁移前的Client ID。 |
| User Name | 必选 | 填写步骤3注册设备时生成的设备ID, 默认通过控制台生成的设备ID会添加产品ID前缀。在设备迁移场景, 设备侧User Name参数无法修改时可以调用 创建设备 接口, 指定设备ID参数值同迁移前的User Name参数值保持一致。 |
| Password | 必选 | 加密后的设备密钥。Password的值为使用“HMACSHA256”算法以时间戳为密钥, 对secret进行加密后的值。 secret为注册设备时平台返回的secret。 |

2. 参考下表配置“SSL/TLS”认证参数，然后单击“Apply”。



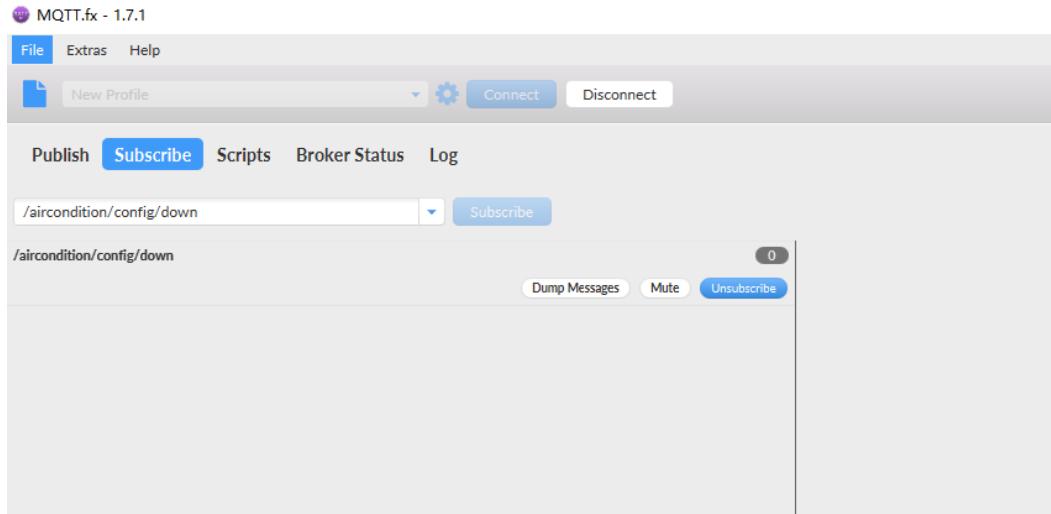
| 参数 | 必选/可选 | 参数描述 |
|---------------------|-------|--|
| Enable SSL/TLS | 必选 | 选择“Enable SSL/TLS” |
| CA certificate file | 必选 | 上传 证书资源 页面获取的CA证书。如果设备侧的CA证书无法修改，可以开通 企业版实例 ，参考 此页面 配置服务端证书。 |

步骤2 设备同物联网平台建立连接后，设备沿用迁移前的topic和payload格式上报数据。物联网平台针对这类非系统预定义的topic，统一按照“设备消息”的处理流程将设备上报的数据转发给第三方应用或者华为云的其他云服务处理。

图 2-63 设备在线



步骤3 根据迁移前的topic进行订阅，接收应用服务器下发的指令。



----结束

服务端开发

对服务端进行业务开发，实现接收设备数据和下发控制指令。本文以Java脚本为例，演示接收设备数据和下发控制指令。

- **业务服务器接收设备数据**

服务器通过AMQP客户端接收消息，详细说明请参见[AMQP客户端接入说明](#)，[Java SDK接入示例](#)。设备迁移场景AMQP收到消息接口参考[设备消息上报通知](#)。

AMQP客户端收到的消息样例，body字段携带设备上报的原始topic和payload。

```
{  
    "resource": "device.message",  
    "event": "report",  
    "event_time": "20201114T034403Z",  
    "notify_data": {  
        "header": {  
            "app_id": "QAksSBSBBQpWYtEKC3LrrOboNk0a",  
            "device_id": "5fae45e358115902ce609882_20201113",  
            "node_id": "20201113",  
            "product_id": "5fae45e358115902ce609882",  
            "gateway_id": "5fae45e358115902ce609882_20201113"  
        },  
        "body": {  
            "topic": "/aircondition/data/up",  
            "content": {  
                "temperature": 26.0  
            }  
        }  
    }  
}
```

```
    }
}
```

JAVA核心代码样例如下：

```
package com.huawei.iot.amqp.jms;

import org.apache.qpid.jms.JmsConnection;
import org.apache.qpid.jms.JmsConnectionFactory;
import org.apache.qpid.jms.JmsConnectionListener;
import org.apache.qpid.jms.message.JmsInboundMessageDispatch;
import org.apache.qpid.jms.transports.TransportOptions;
import org.apache.qpid.jms.transports.TransportSupport;

import javax.jms.*;
import javax.naming.Context;
import javax.naming.InitialContext;
import java.net.URI;
import java.util.Hashtable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.ThreadPoolExecutor;
import java.util.concurrent.TimeUnit;

public class HwlotAmqpJavaClientDemo{
    //异步线程池，参数可以根据业务特点作调整，也可以用其他异步方式来处理。
    private final static ExecutorService executorService = new
    ThreadPoolExecutor(Runtime.getRuntime().availableProcessors(),
        Runtime.getRuntime().availableProcessors() * 2, 60,
        TimeUnit.SECONDS, new LinkedBlockingQueue<>(5000));

    public static void main(String[] args) throws Exception{
        //连接凭证接入键值。
        String accessKey = "${yourAccessKey}";
        long timeStamp = System.currentTimeMillis();
        //UserName组装方法，请参见文档：AMQP客户端接入说明。
        String userName = "accessKey=" + accessKey + "|timestamp=" + timeStamp;
        //连接凭证接入码。
        String password = "${yourAccessCode}";
        //按照qpid-jms的规范，组装连接URL。
        String connectionUrl = "amqps://${UUCID}.iot-amqps.cn-north-4.myhuaweicloud.com:5671?
amqp.vhost=default&amqp.idleTimeout=8000&amqp.saslMechanisms=PLAIN";
        Hashtable<String, String> hashtable = new Hashtable<>();
        hashtable.put("connectionfactory.HwConnectionFactory", connectionUrl);
        //队列名，可以使用默认队列DefaultQueue
        String queueName = "${yourQueue}";
        hashtable.put("queue.HwQueueName", queueName);
        hashtable.put(Context.INITIAL_CONTEXT_FACTORY,
        "org.apache.qpid.jms.jndi.JmsInitialContextFactory");
        Context context = new InitialContext(hashtable);
        JmsConnectionFactory cf = (JmsConnectionFactory) context.lookup("HwConnectionFactory");
        //同一个链接可创建多个queue,与前面queue.HwQueueName作好配对就行
        Destination queue = (Destination) context.lookup("HwQueueName");

        //信任服务端
        TransportOptions to = new TransportOptions(); to.setTrustAll(true);
        cf.setSslContext(TransportSupport.createJdkSslContext(to));

        // 创建连接
        Connection connection = cf.createConnection(userName, password);
        ((JmsConnection) connection).addConnectionListener(myJmsConnectionListener);
        // 创建 Session
        // Session.CLIENT_ACKNOWLEDGE: 收到消息后，需要手动调用message.acknowledge()。
        // Session.AUTO_ACKNOWLEDGE: SDK自动ACK（推荐）。
        Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
        connection.start();
        // 创建 Receiver Link
        MessageConsumer consumer = session.createConsumer(queue);
        //处理消息有两种方式
        // 1, 主动拉数据（推荐），参照receiveMessage(consumer)
```

```
// 2, 添加监听, 参照consumer.setMessageListener(messageListener), 服务端主动推数据给客户端, 但得考虑接受的数据速率是客户能力能够承受住的
    receiveMessage(consumer);
    // consumer.setMessageListener(messageListener);
}

private static void receiveMessage(MessageConsumer consumer) throws JMSEException{
    while (true){
        try{
            // 建议异步处理收到的消息, 确保receiveMessage函数里没有耗时逻辑。
            Message message = consumer.receive(); processMessage(message);
        } catch (Exception e) {
            System.out.println("receiveMessage hand an exception: " + e.getMessage());
            e.printStackTrace();
        }
    }
}

private static MessageListener messageListener = new MessageListener(){
    @Override
    public void onMessage(Message message){
        try {
            // 建议异步处理收到的消息, 确保onMessage函数里没有耗时逻辑。
            // 如果业务处理耗时过程过长阻塞住线程, 可能会影响SDK收到消息后的正常回调。
            executorService.submit(() -> processMessage(message));
        } catch (Exception e){
            System.out.println("submit task occurs exception: " + e.getMessage());
            e.printStackTrace();
        }
    }
};

/**
 * 在这里处理您收到消息后的具体业务逻辑。
 */
private static void processMessage(Message message) {
    try {
        String body = message.getBody(String.class); String content = new String(body);
        System.out.println("receive an message, the content is " + content);
    } catch (Exception e){
        System.out.println("processMessage occurs error: " + e.getMessage());
        e.printStackTrace();
    }
}

private static JmsConnectionListener myJmsConnectionListener = new JmsConnectionListener(){
    /**
     * 连接成功建立。
     */
    @Override
    public void onConnectionEstablished(URI remoteURI){
        System.out.println("onConnectionEstablished, remoteUri:" + remoteURI);
    }

    /**
     * 尝试过最大重试次数之后, 最终连接失败。
     */
    @Override
    public void onConnectionFailure(Throwable error){
        System.out.println("onConnectionFailure, " + error.getMessage());
    }

    /**
     * 连接中断。
     */
    @Override
    public void onConnectionInterrupted(URI remoteURI){
        System.out.println("onConnectionInterrupted, remoteUri:" + remoteURI);
    }
}
```

```
    }

    /**
     * 连接中断后又自动重连上。
     */
    @Override
    public void onConnectionRestored(URI remoteURI){
        System.out.println("onConnectionRestored, remoteUri:" + remoteURI);
    }

    @Override
    public void onInboundMessage(JmsInboundMessageDispatch envelope){
        System.out.println("onInboundMessage, " + envelope);
    }

    @Override
    public void onSessionClosed(Session session, Throwable cause){
        System.out.println("onSessionClosed, session=" + session + ", cause =" + cause);
    }

    @Override
    public void onConsumerClosed(MessageConsumer consumer, Throwable cause){
        System.out.println("MessageConsumer, consumer=" + consumer + ", cause =" + cause);
    }

    @Override
    public void onProducerClosed(MessageProducer producer, Throwable cause){
        System.out.println("MessageProducer, producer=" + producer + ", cause =" + cause);
    }
}
```

- **业务服务器下发控制指令**

服务器通过调用消息下发接口下发控制指令，具体请参考[下发设备消息](#)。

下发设备消息关键参数说明：

| 名称 | 必选/ 可选 | 类型 | 位置 | 说明 |
|-----------------|-----------|-------------|--------|---|
| X-Auth-Token | 必选 | String | Header | 用户Token。通过调用IAM服务 获取IAM用户Token 接口获取，接口返回的响应消息头中“X-Subject-Token”就是需要获取的用户Token。简要的获取方法样例请参见 Token认证 。 |
| Instance-Id | 可选 | String | Header | 实例ID。物理多租下各实例的唯一标识，一般华为云租户无需携带该参数，仅在物理多租场景下从管理面访问API时需要携带该参数。 |
| project_id | 必选 | String | Path | 项目ID。获取方法请参见 获取项目ID 。 |
| device_id | 必选 | String | Path | 填写迁移设备的设备ID。 |
| message | 必选 | String | Body | 设备执行的消息，字符串，具体格式需要应用和设备约定。 |
| topic_full_name | 可选 | String(128) | Body | 迁移设备的topic。 |

接口样例：

```
POST https://{{Endpoint}}/v5/iot/{{project_id}}/devices/{{device_id}}/messages
Content-Type: application/json
X-Auth-Token: *****
Instance-Id: *****

{
    "message": "{\"switch\":\"off\"}",
    "topic_full_name": "/aircondition/cmd"
}
```

JAVA核心代码样例如下：

```
public class DeviceMessage {
    public static void main(String[] args) throws KeyManagementException,
    NoSuchAlgorithmException, IOException {
        String token = Authentication.getToken();
        Map<String, String> headers = new HashMap<String, String>();
        headers.put("Content-Type", "application/json");
        headers.put("X-Auth-Token", token);

        Message message = new Message();
        message.setMessage("{\"switch\": \"off\"}");
        message.setTopic_full_name("/aircondition/cmd");

        String projectId = "11111";
        String deviceld = "5fae45e358115902ce609882_20201113";
        String url = "https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/%s/devices/%s/messages";
        url = String.format(url, projectId, deviceld);
        HttpUtils httpUtils = new HttpUtils();
        httpUtils.initClient();
        StreamClosedHttpResponse httpResponse = httpUtils.doPost(url, headers,
        JsonUtils.Obj2String(message));
        System.out.println(httpResponse.getContent());
    }
}
```

2.9 模拟 NB 设备的接入与调试

场景说明

本文以“智慧路灯”为例，通过设备接入控制台提供的设备模拟器和应用模拟器替代真实的设备和应用，带您快速体验设备上报数据到物联网平台和远程下发控制命令到设备的全过程。

假设：

路灯设备上报一条数据消息，包含路灯的光照强度（light_intensity）和路灯的开关状态（light_status）；支持远程控制路灯开关状态的命令（SWITCH_LIGHT），上报数据的格式为二进制格式。

前提条件

- 已注册华为云官方帐号。未注册可单击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击[实名认证](#)完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“立即使用”后开通该服务。

(可选) 创建资源空间

资源空间是物联网平台提供物联网应用和设备的调测空间，您可以根据场景的不同创建不同资源空间分别调测。

系统为您预置了一个资源空间，您可以直接在预置的资源空间下，在线开发设备的产品模型文件和编解码插件；也可以新建一个资源空间，具体步骤如下。

- 步骤1 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 步骤2 选择左侧导航栏的“资源空间”，单击右上角“新建资源空间”。
- 步骤3 在弹出的页面填写“空间名称”，然后单击“确定”。

----结束

创建产品

基于开发的产品，用户可以在线开发设备的产品模型文件和编解码插件；另外，物联网平台提供了设备和应用模拟器，可以便捷地调测开发的产品模型文件和编解码插件的正确性。

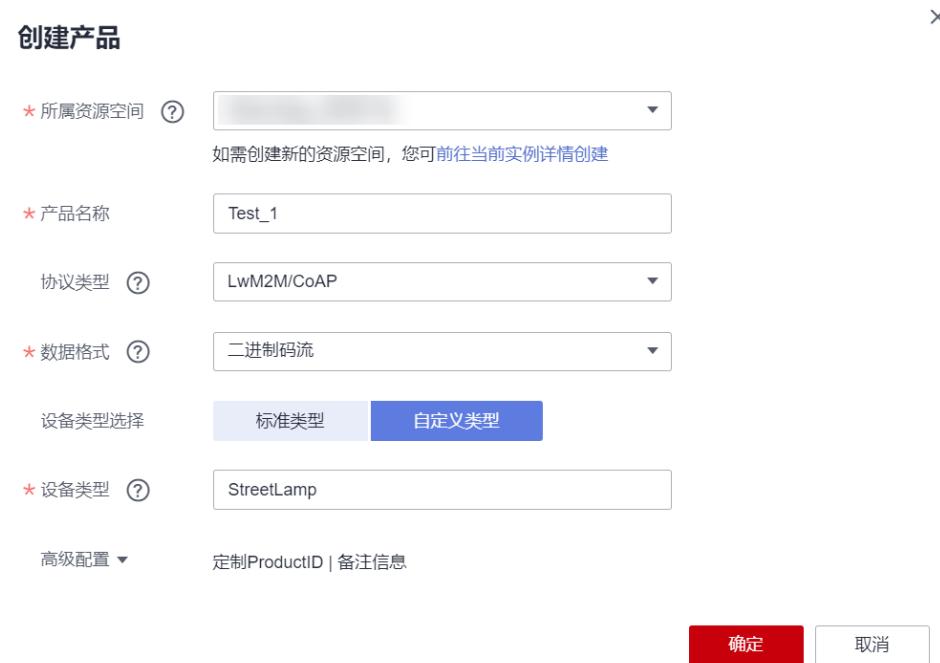
- 步骤1 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 步骤2 选择左侧导航栏的“产品”。

图 2-64 产品-创建产品



- 步骤3 单击左侧的“创建产品”，创建一个基于LwM2M/CoAP协议的产品，填写参数后，单击“确定”。

图 2-65 创建产品-CoAP



| 基本信息 | |
|--------|--|
| 所属资源空间 | 选择新建产品所属的资源空间。 |
| 产品名称 | 自定义, 如Test_1。 |
| 协议类型 | 选择“LwM2M/CoAP”。 |
| 数据格式 | 选择“二进制码流”。 说明 当“数据格式”配置为“二进制码流”时, 该产品下需要进行编解码插件开发; 当“数据格式”配置为“JSON”时, 该产品下不需要进行编解码插件开发。 |
| 所属行业 | 无 |
| 设备类型 | streetlamp |

----结束

定义产品模型

步骤1 找到**步骤3**创建的产品, 单击产品进入产品界面。

步骤2 在产品详情模型定义页面, 单击“自定义模型”, 配置产品的服务, 然后单击“确定”。

- “服务ID” : StreetLight
- “服务类型” : 建议和服务ID保持一致

- “服务描述”：路灯上报的环境光强度和路灯开关状态的属性。

步骤3 单击步骤2新增的服务ID，在展开的页面单击“添加属性”，定义一条属性为路灯采集的当前环境的光照强度。

- “属性名称”：light_intensity
- “数据类型”：int（整型）
- “访问权限”：可读、可写
- “取值范围”：0~100，光照强度范围。

图 2-66 添加属性-light_intensity



步骤4 单击“添加属性”，定义一条属性为路灯当前的开关灯状态。

- “属性名称”：light_status
- “数据类型”：int（整型）
- “访问权限”：可读、可写
- “取值范围”：0~1，0代表关闭，1代表打开状态

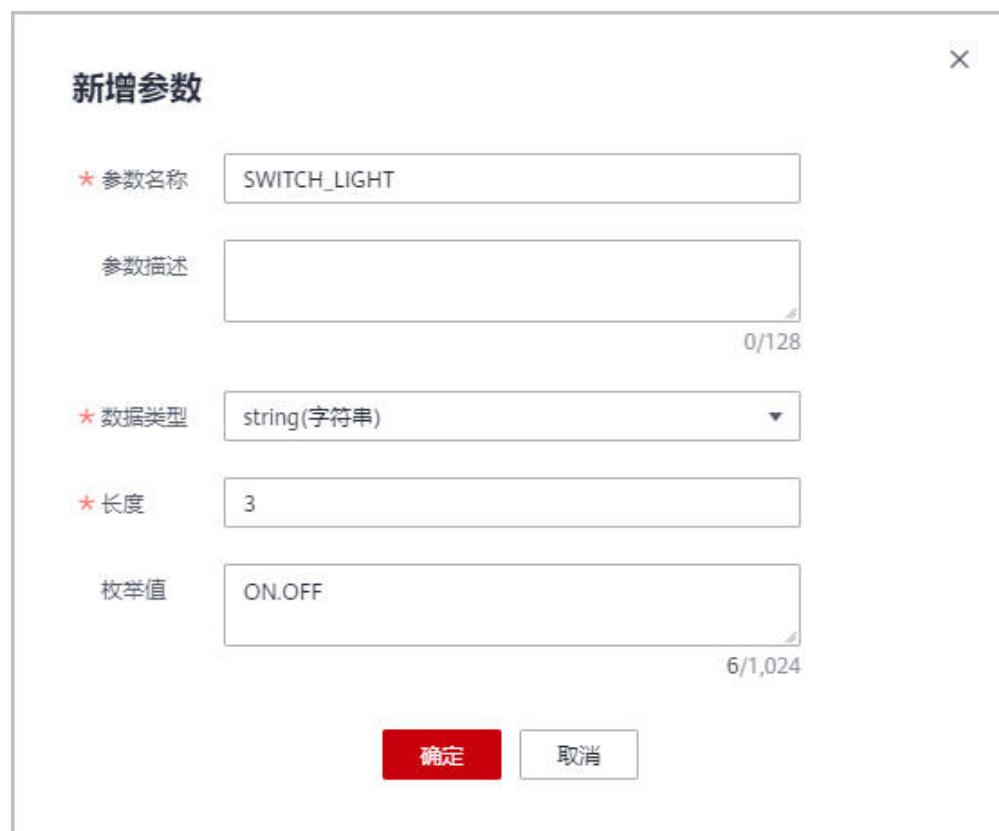
图 2-67 添加属性-light_status



步骤5 接下来定义远程控制开关灯状态的命令。

1. 单击“添加命令”，定义命令名称：SWITCH_LIGHT。
2. 单击“新增输入参数”，参数名称：SWITCH_LIGHT，数据类型为：string，长度为：3个字符，枚举值为：ON,OFF。

图 2-68 添加属性-SWITCH_LIGHT



3. 单击“新增响应参数”，参数名称：result，数据类型：int。

图 2-69 新增响应参数-result



步骤6 单击“确定”，完成该路灯的产品模型文件开发。

----结束

在线开发插件

通常情况下设备为了省电，设备会采用“二进制”格式上报数据，编解码插件的作用就是将设备上报的“二进制”格式数据，按照产品模型文件的定义的属性转换为“JSON”格式数据，便于物联网平台和应用服务器识别。同时，用户远程下发控制命令时，物联网平台会将“JSON”格式的命令转换为“二进制”格式数据下发给设备。

说明

如果设备本身上报的是JSON格式数据，则不需要定义编解码插件。

步骤1 在智慧路灯产品的开发空间，选择“插件开发”，单击“图形化开发”。

步骤2 单击“新增消息”，配置数据上报消息，上报路灯的数据消息。

- 消息名：LightData
- 消息类型：数据上报
- 添加响应字段：是。添加响应字段后，物联网平台在收到设备上报的数据后，会下发用户设置的响应数据到设备。
- 响应数据：AAAA0000（默认）

图 2-70 插件开发-新增消息



- 单击“添加字段”，添加**messageId**字段，表示消息种类。

图 2-71 插件开发-添加字段 messageId



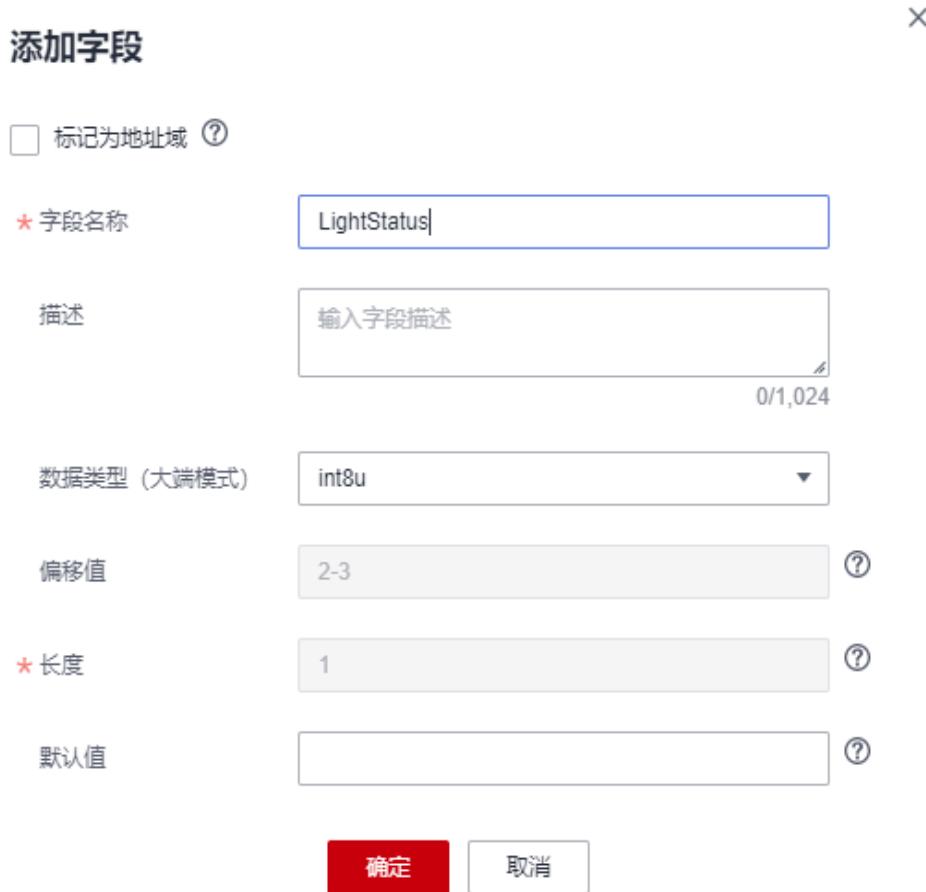
2. 添加**LightIntensity**字段，表示光照强度，数据类型：int8u（8位无符号整型），长度：1字节。

图 2-72 插件开发-添加字段 LightIntensity



3. 添加**LightStatus**字段，表示路灯上报的路灯开关状态数据。数据类型：int8u（8位无符号整型），长度：1字节。

图 2-73 插件开发-添加字段 LightStatus



步骤3 再次单击“新增消息”，定义远程控制路灯开关的命令对应的编解码插件消息。

- 消息名：SwitchStatus
- 消息类型：命令下发
- 添加响应字段：是。添加响应字段后，设备在接收命令后，会上报命令执行结果。

图 2-74 插件开发-新增消息 SwitchStatus



- 单击“添加字段”，添加**messageId**字段，表示消息种类。

图 2-75 插件开发-添加字段 messageId



2. 添加mid字段，用于将下发的命令和命令执行结果进行关联。

图 2-76 插件开发-添加字段 mid



3. 添加**SwitchStatus**字段，数据类型：string（字符串类型），长度：3字符。

图 2-77 插件开发-添加字段 SwitchStatus



4. 单击“添加响应字段”，配置命令下发响应消息。
 - 添加**messageld**，表示消息种类。命令执行结果为上行消息，需要通过 messageld和数据上报消息进行区分。

图 2-78 插件开发-添加响应字段 messageId



- 添加mid字段，用于将下发的命令和命令执行结果进行关联。

图 2-79 插件开发-添加响应字段 mid



- 添加errcode字段，用于表示命令执行状态：00表示成功，01表示失败，如果未携带该字段，则默认命令执行成功。

图 2-80 插件开发-添加响应字段 errcode



- 添加result字段，用于表示命令执行结果。

图 2-81 插件开发-添加响应字段 result



步骤4 拖动右侧“设备模型”区域的属性字段和命令字段（产品模型文件定义的字段），与编解码插件定义的数据上报消息和命令下发消息的相应字段建立映射关系。

图 2-82 插件开发-在线开发插件



步骤5 编解码插件与产品模型文件建立映射完成后，单击右上角“保存”，并单击“部署”，完成编解码插件的部署。

----结束

使用真实设备在线调试

模拟器在线调测具备设备模拟和应用模拟功能，可以对定义的产品模型文件和编解码插件进行调试，用户可以直观的感受设备上报数据到物联网平台，以及使用物联网平台下发远程控制命令。

步骤1 在智慧路灯产品的开发空间，选择“在线调试”，单击“新增测试设备”。

步骤2 在新增测试设备页面，完成设备参数的填写后，单击“确定”。

- 设备类型：真实设备。
- 设备名称：自定义。
- 设备标识码：设备的IMEI号，用于设备在接入物联网平台时携带该标识信息完成接入鉴权，可在NB模组上查看。
- 设备注册方式：不加密。如果使用DTLS传输层安全协议接入时，设备注册方式选择“加密”，且请妥善保存密钥。

图 2-83 在线调试-创建真实设备

The dialog box for adding a test device has the following fields:

- 设备类型:** 真实设备 (Real Device) is selected.
- 设备名称:** test
- 设备标识码:** (IMEI number field)
- 设备注册方式:** 不加密 (Unencrypted) is selected.
- Buttons:** 确定 (Confirm) and 取消 (Cancel).

说明书

新添加的设备处于未激活状态，此时不能进行在线调试，可参考[连接鉴权](#)，待设备接入平台后，进行调试。

步骤3 单击“调试”，进入调试界面。

图 2-84 进入调试界面



步骤4 模拟远程下发控制命令场景，在应用模拟器中，选择服务：StreetLight，命令：SWITCH_LIGHT，命令取值为：ON，单击“立即发送”，我们可以看到路灯被点亮。

----结束

使用虚拟设备在线调试

步骤1 在智慧路灯产品的开发空间，选择“在线调试”，单击“新增测试设备”。

步骤2 在新增测试设备页面，选择“虚拟设备”，然后单击“确定”，创建一个虚拟设备。虚拟设备名称包含“Simulator”字样，每款产品下只能创建一个虚拟设备。

图 2-85 在线调试-创建虚拟设备



步骤3 单击“调试”，进入调试界面。

图 2-86 进入调试界面



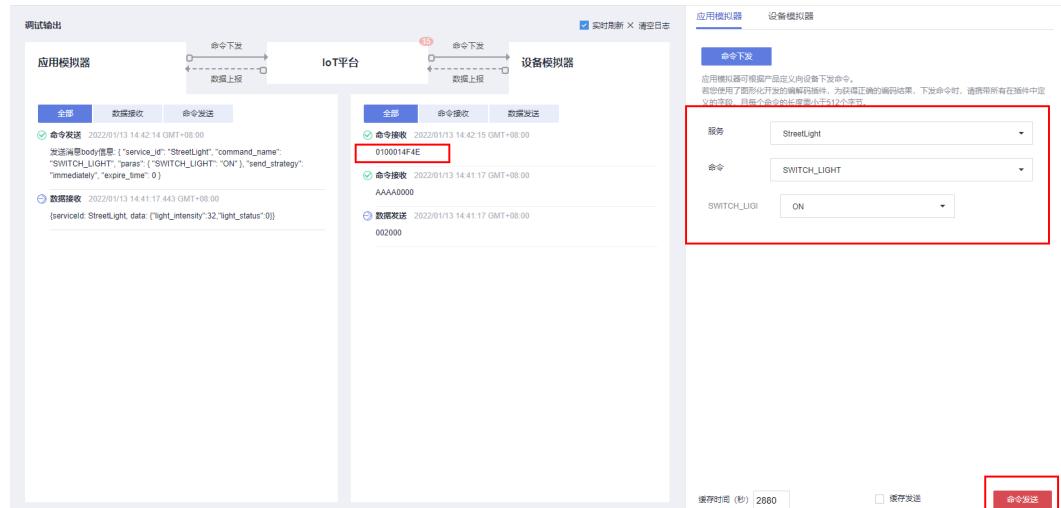
步骤4 模拟设备数据上报场景，在设备模拟器中，输入十六进制码流：002000（00为地址域meaasgeID，长度为1个字节；20表示光照强度，长度为1个字节；00表示路灯开关状态为关闭状态，长度为1个字节），然后单击“发送”，我们可以在应用模拟器中看到转换为JSON格式的数据为："Light_Intensity":32, "Light_Status":0。

图 2-87 模拟数据上报



步骤5 模拟远程下发控制命令场景，在应用模拟器中，选择服务：StreetLight，命令：SWITCH_LIGHT，命令取值为：ON，单击“立即发送”，我们可以在设备模拟器中看到转换为十六进制的码流：0100014F4E（经ASCLL码转为为十六进制）。

图 2-88 模拟命令下发



----结束

使用离线模拟器调试

NB-IoT设备模拟器用于模拟以LwM2M/CoAP协议接入物联网平台的NB设备，实现数据上报和命令下发功能。

步骤1 获取**2**中注册设备时的设备标识码，密钥。

步骤2 下载并解压NB-IoT设备模拟器，然后双击“NB-IoTDeviceSimulator_zh.jar”，运行模拟器。

说明

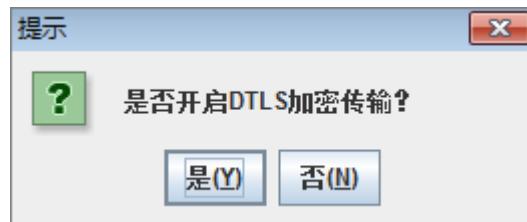
- 推荐在Windows系统下运行，暂不支持MAC OS。
- 请确保已经安装jdk，否则，无法运行jar文件。

| Name | Size | Packed | Type | Modified | CRC32 |
|------------------------------|-----------|-----------|---------------------|-----------------|-----------|
| .. | | | 文件夹 | | |
| images | | | 文件夹 | 2020/10/26 9... | |
| NB-IoTDeviceSimulator_zh.jar | 4,464,721 | 4,080,631 | Executable Jar File | 2017/8/7 14:07 | B47CBF61 |
| Californium.properties | 1,203 | 613 | PROPERTIES 文件 | 2017/7/29 12... | 01A69483 |
| setting.properties | 111 | 108 | PROPERTIES 文件 | 2017/8/7 15:21 | 3CEB9C... |

NB-IoT设备模拟器的文件说明如下：

- **NB-IoTDeviceSimulator_zh.jar**: 中文版模拟器。
- **Californium.properties**: 模拟器的配置文件。
- **setting.properties**: 设备模拟器接入物联网平台的配置文件。

步骤3 模拟器启动后，会提示“是否开启DTLS加密传输？”若**注册设备**时，没有填写密钥，单击“否”，若**注册设备**时，已填写密钥，单击“是”。



步骤4 填写“IP地址”、“VerifyCode”和“PSK”，单击“注册设备”，将模拟器与物联网平台进行绑定。

注：如果未开启DTLS加密传输，则无需填写密钥。

如下信息根据实际情况填写：

- IP地址：**物联网平台**的接入域名（也可以使用IP，通过ping命令“ping 域名”可以获得IP地址）。
- VerifyCode：设备标识码，如：aaaaa11111。
- PSK：注册设备时预置的密钥，如：aaaaa11111aaaaa。



设备绑定成功，可以回到管理控制台，选择“设备 > 所有设备”，看见设备“aaaaaa11111”显示“在线”，表示模拟器绑定成功。

- 步骤5** 模拟设备数据上报场景，则在NB-IoT设备模拟器中，输入十六进制码流：001400（00为地址域meaasgeID，长度为1个字节；14表示光照强度，长度为1个字节；00表示路灯开关状态为关闭状态，长度为1个字节），然后单击“发送数据”。

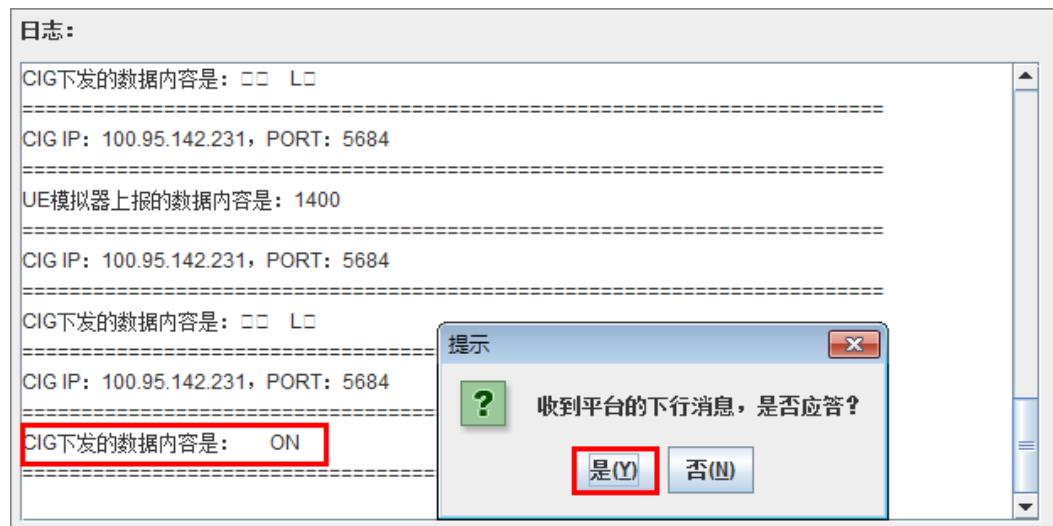


数据上报成功，可以回到管理控制台，在“aaaaaa11111”设备详情页面中查看到最新上报数据为：“Light_Intensity”: 20, “Light_Status”: 0。

- 步骤6** 模拟远程下发控制命令场景，在管理控制台，找到对应的产品，单击产品进入产品界面，选择“在线调试”页签，单击设备“aaaaaa11111”右边的“调试”，进入调试界面。

在应用模拟器中，选择服务：StreetLight，命令：SWITCH_LIGHT，命令取值为：ON，单击“立即发送”。

我们可以在“日志”栏看到CIG下发的数据内容：ON，且模拟器会提示“收到平台的下行消息，是否应答？”，单击“是”，可以在管理控制台的应用模拟器看到命令状态为“已送达”。



说明

由于插件中的string类型是按照ascii码解析的，因此需要下发可显示字符，非可显示字符在模拟器中不会显示。

----结束

2.10 利用工业网关（包含 RS485、AI、DO 口设备）进行环境监测

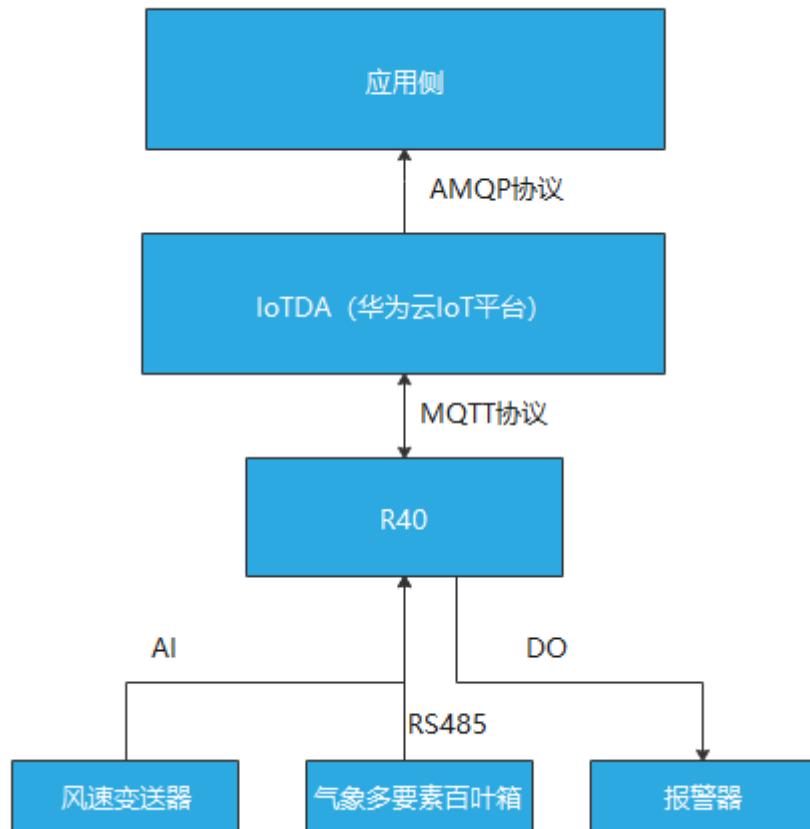
场景说明

微型气象是不可预测的，基本上不能通过人工手段来分析其变化，因此必须运用新技术，对气象进行实时监测，以便采取相应的措施来避免或解决事故的发生。而常规气象环境数据采集容易造成数据损失、人力成本高、数据安全性差、可追踪性差等问题。为解决上述问题，可以通过网关将设备采集的数据上报到华为云IoT平台，实现对现场设备数据的实时监控，您可通过华为云IoT平台实时查看现场温度、湿度、光照、风速等环境数据，并对数据进行转储、分析等处理，同时也能通过IoT平台对设备进行控制。本文以金鸽的R40设备为例进行演示。

整体流程

整体流程如图所示：

图 2-89 整体流程图



前提条件

- 硬件要求：三个输出为12V 1000mA的电源。

设备简介

R40（如图2）负责将传感器的数据收集上来，上报到华为云IoT平台，同时负责将华为云IoT平台下发的控制信息下发给传感器。其中演示的传感器设备有如下三种：

1. 风速变送器（如图3）将采集到的风速通过电流信号输出给R40；
2. 气象多要素百叶箱（如图4）将采集到的噪声、光照、温湿度信息通过RS485口（Modbus协议）输出给R40。
3. 报警器（如图5）通过R40的DO口接收从华为云IoT下发给的控制信息。

图 2-90 R40 介绍



图 2-91 风速变送器



图 2-92 气象多要素百叶箱



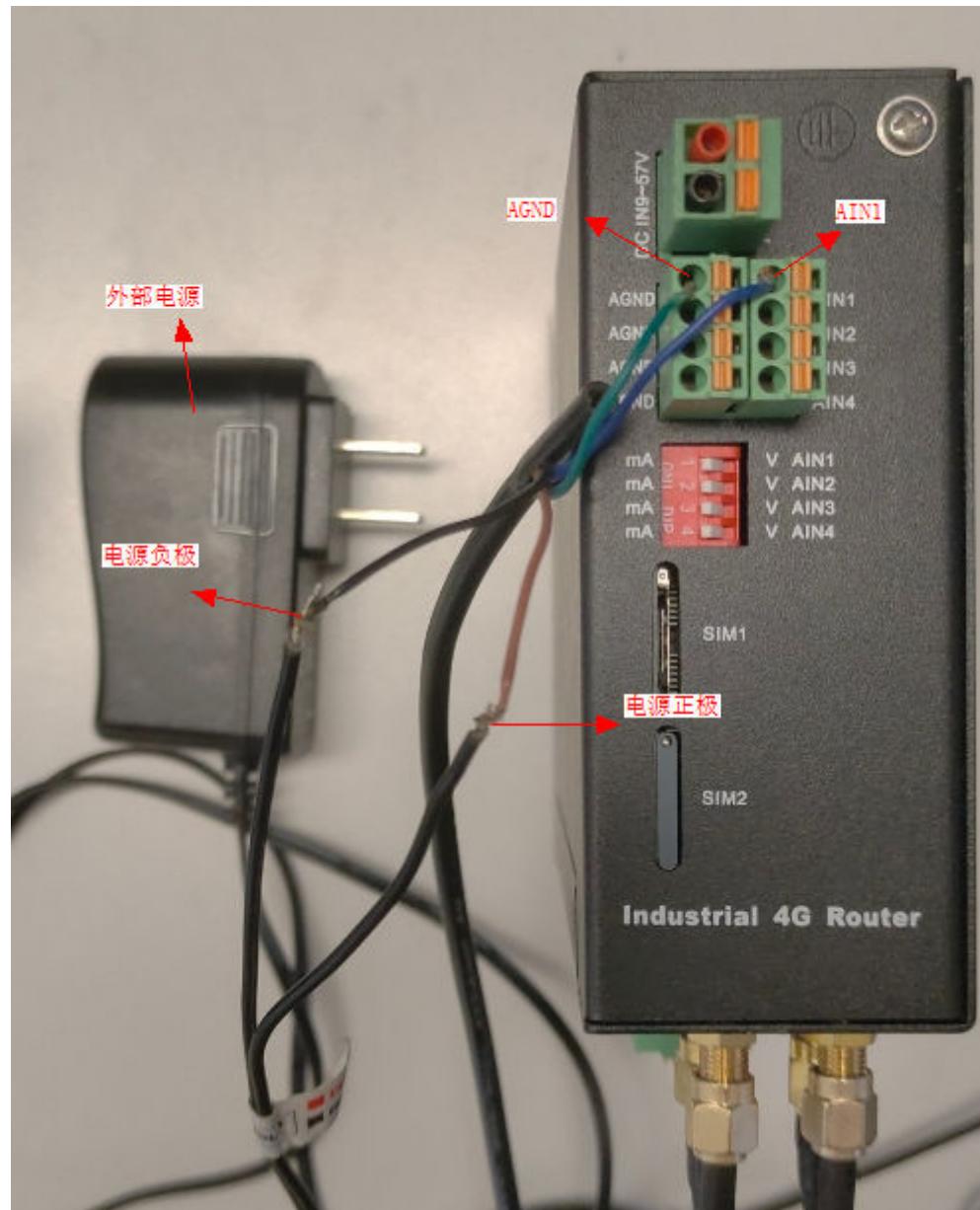
图 2-93 警报器



硬件连接

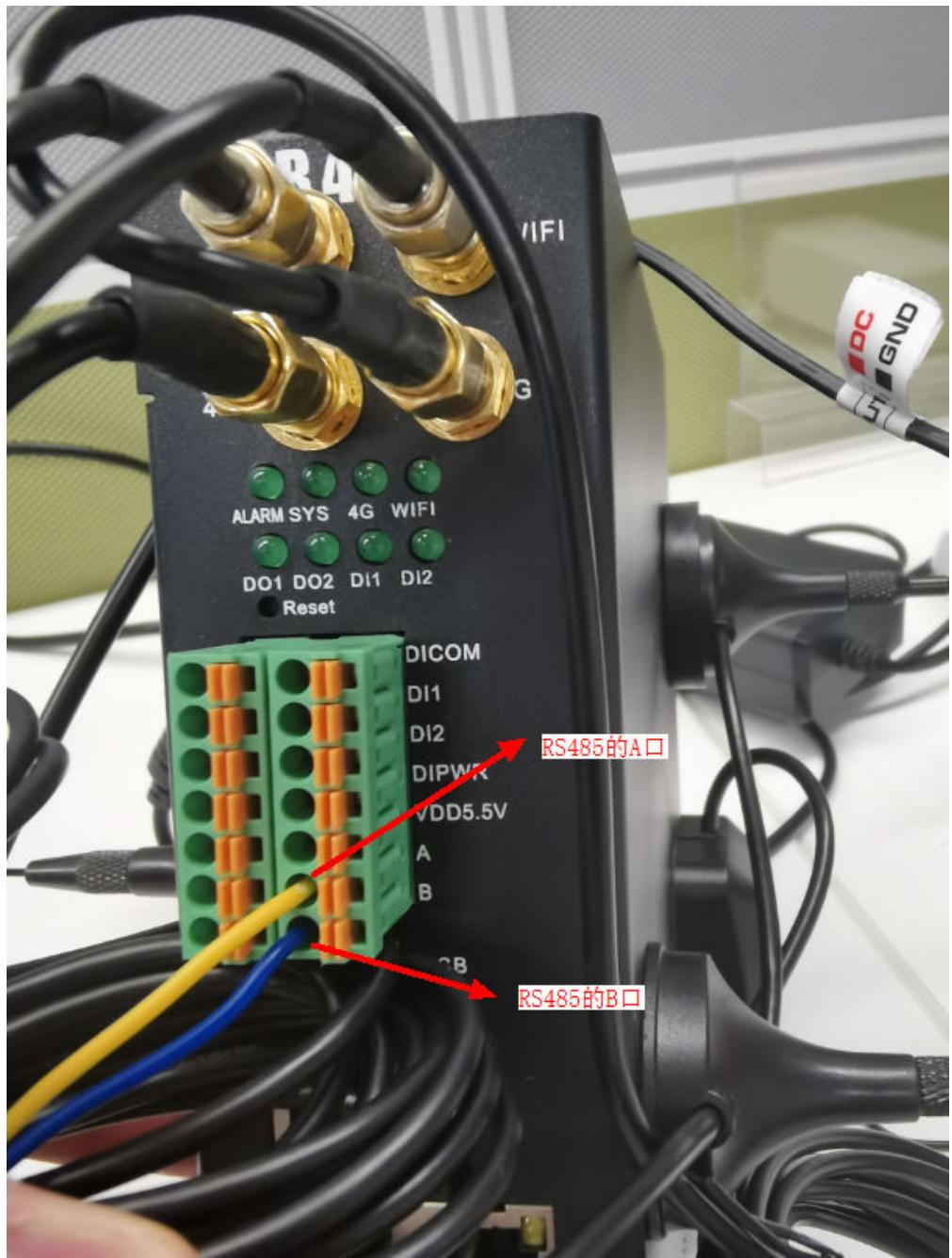
1. 将风速变送器的棕色线和黑色线分别接到外部电源的正级和负极，将蓝色线和绿色线分别接入到R40的AIN1口和AGND口（见下图）。

图 2-94 风速变送器连接 R40



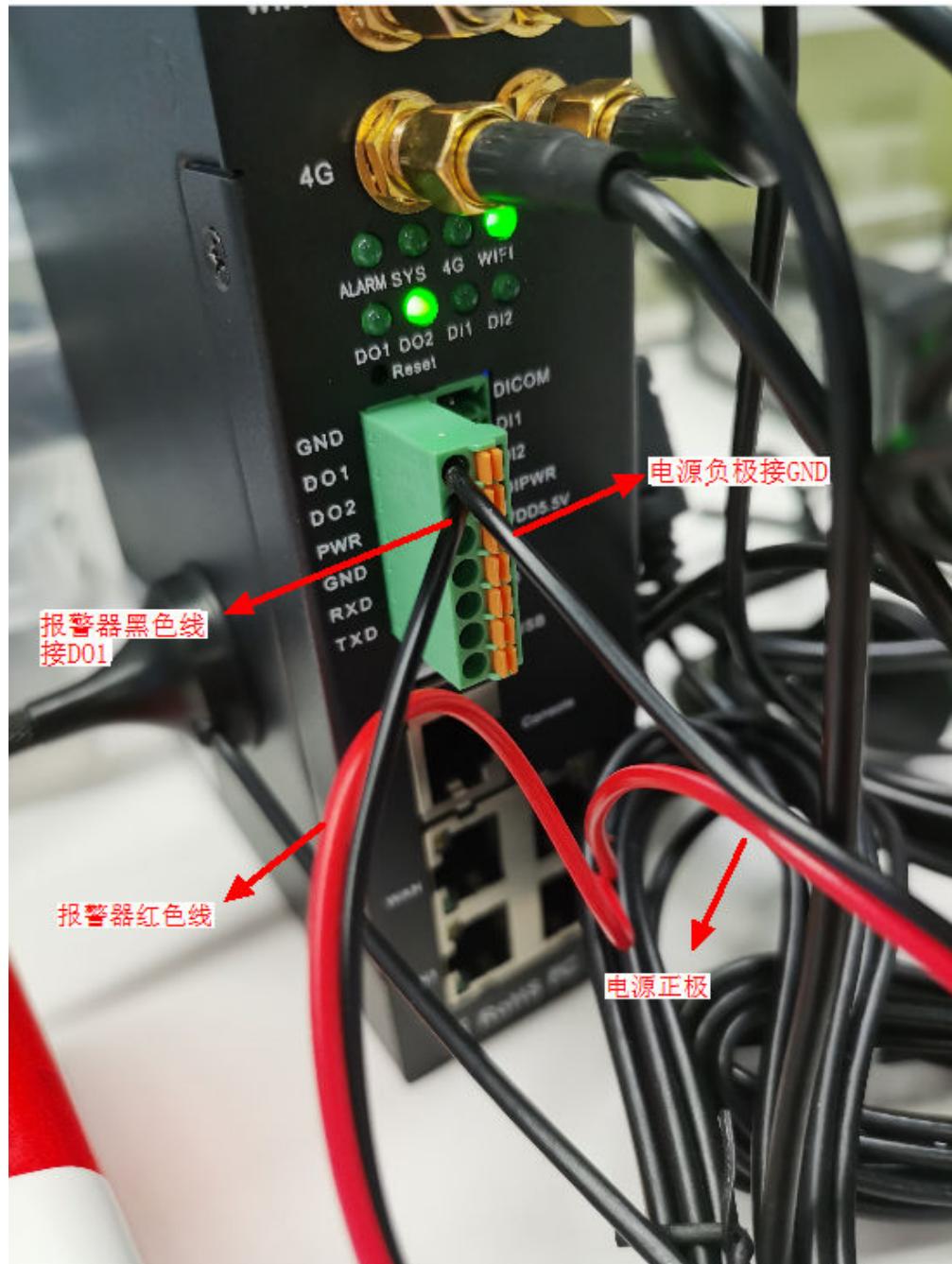
2. 将气象多要素百叶箱的棕色线和黑色线分别接到电源的正极和负极，将黄色线和蓝色线分别接入到R40的485A口和485B口（见下图）。

图 2-95 气象多要素百叶箱连接 R40



3. 将报警器的红色线和黑色线分别接到电源正极和R40的DO2口，将电源的负极接入到R40的DO1口（见下图）。

图 2-96 报警器连接 R40



控制台操作概览

在真实设备连接之前，需要在设备接入服务控制台创建产品、定义产品模型和注册设备。

- **创建产品：**在物联网平台上规定好某一款产品的协议类型、数据格式、厂商名称、设备类型。此处我们需要按照R40的特征，在控制台创建R40的产品。
- **定义产品模型：**产品模型是用来描述设备能力的文件，通过JSON的格式定义了设备的基本属性、上报数据和下发命令的消息格式。定义产品模型，即在物联网平台构建一款设备的抽象模型，使平台理解该款设备支持的属性信息，例如风速、光照强度、温湿度、噪声等。

- 注册设备：将R40设备注册到物联网平台。

创建产品

某一类具有相同能力或特征的设备的集合称为一款产品。除了设备实体，产品还包含该类设备在物联网能力建设中产生的产品信息、产品模型（Profile）等资源。下面，我们按照R40的基本特征，在控制台上创建R40产品。

- 步骤1** 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 步骤2** 选择左侧导航栏的“产品”，然后单击左侧的“创建产品”，创建一个基于MQTT协议的产品，参考下表填写参数后，单击“确定”，完成产品的创建。

| 基本信息 | |
|--------|-------------------------------|
| 所属资源空间 | 选择您需要归属到的资源空间 |
| 产品名称 | 自定义，如“R40” |
| 协议类型 | 选择“MQTT” |
| 数据格式 | 选择“JSON” |
| 所属行业 | 无 |
| 设备类型 | EnvironmentalMonitoringDevice |

- 步骤3** 产品创建成功后，单击“查看详情”按钮进入产品界面，进行后面操作。

----结束

上传产品模型

产品模型是用来描述设备能力的文件，通过JSON的格式定义了设备的基本属性、上报数据和下发命令的消息格式。定义产品模型，即在物联网平台构建一款设备的抽象模型，使平台理解该款设备的功能。我们已提供开发好的产品模型，帮助您快速体验上云流程，如果您想体验产品模型的开发流程，可参考[开发产品模型](#)。

操作步骤：

1. 在产品详情“模型定义”页面，单击“上传模型文件”。
2. 在弹出的页面中，上传提供的产品模型（[本示例的产品模型](#)），然后单击“确认”。

图 2-97 产品-上传模型文件



注册设备

1. 在管理控制台页面，选择左侧导航栏“设备 > 所有设备”，单击页面右上角的“注册设备”。
2. 根据页面提示信息填写参数，然后单击“确定”。

| 参数名称 | 说明 |
|--------|-----------------------------------|
| 所属资源空间 | 确保和之前创建的产品归属在同一个资源空间。 |
| 所属产品 | 选择之前创建的产品。 |
| 设备标识码 | 即nodeID，设备唯一物理标识。可自定义，由英文字母和数字组成。 |
| 设备名称 | 即device_name，可自定义。 |
| 设备认证类型 | 选择“密钥”。 |
| 密钥 | 设备密钥，可自定义，不填写物联网平台会自动生成。 |

图 2-98 所有设备-注册设备

The screenshot shows the 'All Devices' management interface. On the left, there is a navigation sidebar with 'Standard Edition' selected. Under 'Devices', 'All Devices' is highlighted with a red box. The main area displays a table of registered devices with columns: Status, Device Name, Device ID, Resource Space, Product, and Node Type. A search bar at the top allows filtering by device ID. On the right, there are buttons for 'Batch Registration', 'Batch Update', 'Batch Delete', and 'File Upload'. A red box highlights the 'Register Device' button in the top right corner.

成功注册设备后，平台会自动生成设备ID和密钥，请妥善保管好设备ID（deviceld）和密钥（deviceSecret），用于设备接入。

图 2-99 注册设备成功



R40 配置

在R40上需要对传感器的参数进行配置：

风速变速器相关配置：

- R40上电后，在自己的电脑上连接King开头的wifi，连上后在浏览器中输入地址192.168.3.1进入R40的配置界面。
- 输入用户名（默认为admin）和密码（默认无密码）后，点击“登录”按钮。
- 点击上方菜单栏“RTU I/O”，选中“模拟量接入”，进入模拟输入配置界面。

The screenshot shows the R40 configuration interface with the 'RTU I/O' tab selected. Under '模拟输入' (Analog Input), there are two sections: '模拟输入设置' (Analog Input Settings) and '触发设置' (Trigger Settings). In the '模拟输入设置' section, four channels (AIN1-4) are configured with their respective modes: '电流4~20mA' (Current 4~20mA) for AIN1, '电压0~5V' (Voltage 0~5V) for AIN2, AIN3, and AIN4. A context menu is open over the AIN1 row, with '模拟量输入' (Analog Input) highlighted. The '触发设置' section shows a single trigger rule for AIN1: '模拟输入大于门限' (Analog input exceeds threshold) with values 3, 2, 1, and action 'D01断开' (D01 disconnected).

- 在AIN1输入一行（因为风速变速器选择的是该口）中，选择“电流4~20mA”（风速变送器的输出为该值）。并且按照传感器推荐的将最小值设置为0，最大值设置为30，单位设置为m/s。
- 设置完毕后，点击“保存并应用”。
- 通过旋转风速变送器的扇叶，然后刷新一下当前的配置页，可以看到“当前值”那一栏中有数据，说明已经能采集到传感器的数据。

| 输入口 | 模式 | 最小值 | 最大值 | 当前值 | 单位 |
|------|----------|-----|-----|----------|-----|
| AIN1 | 电流4~20mA | 0 | 30 | 1.132147 | m/s |
| AIN2 | 电压0~5V | | | 0.004167 | |
| AIN3 | 电压0~5V | | | 0.004167 | |
| AIN4 | 电压0~5V | | | 0.005449 | |

气象多要素百叶箱相关配置：

- R40上电后，在自己的电脑上连接King开头的wifi，连上后在浏览器中输入地址192.168.3.1进入R40的配置界面。
- 输入用户名（默认为admin）和密码（默认无密码）后，点击“登录”按钮。
- 点击上方菜单栏“应用”，选择“Modbus”主站，进入Modbus主站配置页面。
- 在下方的框中填写“noise”，单击“添加”按钮。添加完成后，点击“编辑”按钮（“删除”按钮旁的那个），进入传感器的配置界面。
- 在配置界面，将别名设置为“噪声监测”，从机地址设置为“2”，寄存器类型设置为“16位数据”，功能码设置为“03”，寄存器起始地址设置为“0”，数据个数设置为“1”，映射地址分配设置为“手动”，映射起始地址设置为“20004”，从机接口设置为“RS485”。

The screenshot shows the 'Modbus主站' (Modbus Master) configuration page. For the 'noise' sensor entry, the following parameters are set:

- 别名 (Alias): 噪声监测
- 从机地址 (Slave Address): 2
- 寄存器类型 (Register Type): 16位数据
- 功能码 (Function Code): 03
- 寄存器起始地址 (Register Start Address): 0
- 数据个数 (Data Length): 1
- 映射地址分配 (Mapping Address Allocation): 手动
- 映射起始地址 (Mapping Start Address): 20004
- 从机接口 (Slave Interface): RS485

At the bottom right, there are '保存' (Save) and '复位' (Reset) buttons.

- 设置完毕后，点击“保存”。然后点击左下角的“返回至概况”。然后点击“保存并应用”。
- 点击“noise”这一行的“查询”按钮，可以看到“数据值”，如果周围噪声发生改变，可以通过刷新按钮来查看值的变化。

说明

从机地址、寄存器类型、功能码以及寄存器起始地址可以从气象多要素百叶箱的配置文件中获取到。

映射地址为R40的寄存器地址，建议改为手动，同时设置映射地址。因为该地址跟物模型中的属性是对应的，属性名称需要设置成“REG”+映射起始地址。这里设置20004是因为物模型中已经设置了REG20004的属性名。

- 从噪声查看页点击“返回至概况”，在Modbus设置界面下方的空白框处填写“lightIntensity”，点击“添加”按钮。添加完成后，点击“编辑”按钮（“删除”按钮旁的那个），进入传感器的配置界面。
- 在配置界面，将别名设置为“光照强度”，从机地址设置为“5”，寄存器类型设置为“32位数据”，功能码设置为“03”，寄存器起始地址设置为“0”，数据个数设置为“1”，映射地址分配设置为“手动”，映射起始地址设置为“20130”，从机接口设置为“RS485”。



- 设置完毕后，点击“保存”。然后点击左下角的“返回至概况”。然后点击“保存并应用”
- 点击“lightIntensity”这一行的“查询”按钮，可以看到“数据值”，如果周围光照强度发生改变，可以通过刷新按钮来查看值的变化。

说明

从机地址、寄存器类型、功能码以及寄存器起始地址可以从气象多要素百叶箱的配置文件中获取到。

映射地址为R40的寄存器地址，建议改为手动，同时设置映射地址。因为该地址跟物模型中的属性是对应的，属性名称需要设置成“REG”+映射起始地址。这里设置20130是因为物模型中已经设置了REG20130的属性名。

- 从光照强度查看页点击“返回至概况”，在Modbus设置界面下方的空白框处填写“shidu”，点击“添加”按钮。添加完成后，点击“编辑”按钮（“删除”按钮旁的那个），进入传感器的配置界面。
- 在配置界面，将别名设置为“湿度”，从机地址设置为“1”，寄存器类型设置为“16位数据”，功能码设置为“03”，寄存器起始地址设置为“500”，数据个数设置为“1”，映射地址分配设置为“手动”，映射起始地址设置为“20015”，从机接口设置为“RS485”。

Modbus主站

Modbus主站

| | |
|---------|-------|
| 别名 | 湿度 |
| 从机地址 | 1 |
| 寄存器类型 | 16位数据 |
| 功能码 | 03 |
| 寄存器起始地址 | 500 |
| 数据个数 | 1 |
| 映射地址分配 | 手动 |
| 映射起始地址 | 20015 |

bool:66-256
16bit:20001-20002,20016-20127
32bit:20128-20129,20132-20254

从机接口

RS485

[返回至概况](#) 保存 复位

- 设置完毕后，点击“保存”。然后点击左下角的“返回至概况”。然后点击“保存并应用”
- 点击“shidu”这一行的“查询”按钮，可以看到“数据值”，如果周围光照强度发生改变，可以通过刷新按钮来查看值的变化。

说明

从机地址、寄存器类型、功能码以及寄存器起始地址可以从气象多要素百叶箱的配置文件中获取到。

映射地址为R40的寄存器地址，建议改为手动，同时设置映射地址。因为该地址跟物模型中的属性是对应的，属性名称需要设置成“REG” + 映射起始地址。这里设置20115是因为物模型中已经设置了REG20115的属性名。

- 从湿度强度查看页点击“返回至概况”，在Modbus设置界面下方的空白框处填写“wendu”，点击“添加”按钮。添加完成后，点击“编辑”按钮（“删除”按钮旁的那个），进入传感器的配置界面。
- 在配置界面，将别名设置为“温度”，从机地址设置为“1”，寄存器类型设置为“16位数据”，功能码设置为“03”，寄存器起始地址设置为“501”，数据个数设置为“1”，映射地址分配设置为“手动”，映射起始地址设置为“20003”，从机接口设置为“RS485”。

Modbus主站

Modbus主站

别名: 温度

从机地址: 1

寄存器类型: 16位数据

功能码: 03

寄存器起始地址: 501

数据个数: 1

映射地址分配: 手动

映射起始地址: 20003

从机接口: RS485

bool:66-256
16bit:20001-20002,20016-20127
32bit:20128-20129,20132-20254

返回至概况 保存 复位

- 设置完毕后，点击“保存”。然后点击左下角的“返回至概况”。然后点击“保存并应用”
- 点击“shidu”这一行的“查询”按钮，可以看到“数据值”，如果周围光照强度发生改变，可以通过刷新按钮来查看值的变化。

说明

从机地址、寄存器类型、功能码以及寄存器起始地址可以从气象多要素百叶箱的配置文件中获取到。

映射地址为R40的寄存器地址，建议改为手动，同时设置映射地址。因为该地址跟物模型中的属性是对应的，属性名称需要设置成“REG” + 映射起始地址。这里设置20003是因为物模型中已经设置了REG20003的属性名。

R40上报到平台的配置：

- 在上方的菜单栏中选择“云平台”，选择“华为云”。“认证方式”选择“设备密钥”，设备ID和密钥在iot平台注册所得，服务ID填写“basicData”，发布周期填写60。
- 配置完后，点击“保存并应用”。设置完毕后，R40会自动连上华为云iot平台，并且每隔60s报一次数据。

云连接设置

云连接设置

启用设置:

认证方式: 设备密钥

设备ID: [REDACTED]

密钥: [REDACTED]

服务ID: basicData

地域(Region ID): 华北-北京四

发布周期(秒): 60

保存并应用 保存 复位

设备上线及数据上报

在华为云IoT平台的“设备”>“所有设备”一栏里，可以看到注册的设备已经上线了。

图 2-100 设备在线

The screenshot shows the 'All Devices' interface. At the top, there are tabs: '设备列表' (Device List), '批量注册' (Batch Registration), '批量更新' (Batch Update), '批量删除' (Batch Delete), '批量添加设备至群组' (Batch Add Device to Group), and '文件上传' (File Upload). Below the tabs, there are two buttons: '注册设备' (Register Device) in red and '删除' (Delete) in grey. A search bar with placeholder text '默认按照设备标识码搜索' (Search by Device Identifier) is present. Under the search bar, there are two filter options: '状态' (Status) with a checkbox and '在线' (Online) with a green dot, both currently unchecked. The main table lists devices with columns: '设备名称' (Device Name) and '设备标识码' (Device Identifier). One device entry is visible.

在上述页面点击该设备右侧的“查看”按钮，可以看到上报的数据。

图 2-101 查看上报数据-工业网关

The screenshot shows the 'View Report Data - Industrial Gateway' interface. At the top, there are tabs: '在线' (Online), '离线' (Offline), and '所属产品' (Associated Product). On the right, there is a link '下数设备历史数据' (Historical Data of Subordinate Devices). The main area displays device details: '所属资源空间' (Resource Space), '设备标识码' (Device Identifier), '节点类型' (Node Type), '软件版本' (Software Version), '注册时间' (Registration Time), '最近上线时间' (Last Online Time), '设备ID' (Device ID), '认证类型' (Authentication Type), '固件版本' (Firmware Version), '设备描述' (Device Description), '激活时间' (Activation Time), and 'MQTT连接参数' (MQTT Connection Parameters). Below this, there is a table titled '物模型数据' (Thing Model Data) with columns: '属性名' (Attribute Name), '最新更新时间' (Latest Update Time), and '值' (Value). The table contains four rows of data. At the bottom, there are pagination controls: '16' (page size), '总条数: 5' (Total items: 5), and navigation arrows.

属性配置下发

单击上图的“设备影子”>“属性配置”>设置DO1为1，然后点击“确定”按钮，即可听到报警器发出报警声，代表属性设置成功。

图 2-102 查看上报数据-工业网关

The screenshot shows a detailed view of a device's report data. At the top, there are tabs for '基础' (Basic), '高级' (Advanced), and '历史' (History). Below the tabs, the device's basic information is listed:

| | |
|----------|-------------------------------|
| 所属资源空间 | ... |
| 设备标识码 | ... |
| 节点类型 | 直接设备 |
| 软件版本 | - |
| 注册时间 | 2024/01/23 17:03:19 GMT+08:00 |
| 最近上线时间 | 2024/01/23 17:43:00 GMT+08:00 |
| 设备ID | ... |
| 认证类型 | 密钥 |
| 固件版本 | - |
| 设备用途 | - |
| 激活时间 | 2024/01/23 17:04:54 GMT+08:00 |
| MQTT连接参数 | 查看 |

Below this, there is a section titled '物模型数据' (Thing Model Data) with a note: '该节点下无父级物模型，设备按照物模型定义呈现上报的属性数据。注意：当上报属性名称和产品模型中配置的属性名称不一致或属性名称中包含特殊字符(., \$, \等十六进制的ASCII码为00时，会无法正常刷新属性数据)'.

A search bar for '请输入服务名称' (Enter Service Name) is present, along with a timestamp '最新更新时间: 2015/12/12 20:12:12 GMT+08:00' and a search bar for '请输入属性名称' (Enter Attribute Name).

The main data area displays four data items under the service name 'basicData':

| | | | |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| AI1 AI1 0 | REQ20003 REQ20003 267 | REQ20004 REQ20004 439 | REQ20130 REQ20130 322 |
| REG20015 REG20015 657 | | | |

At the bottom, there is a pagination control with '16' (current page), '总条数: 5', and navigation arrows.

说明

其中的服务名称basicData是在产品模型中自定义的，可以修改为自定义的内容，同时上报数据时也需要同步修改。

2.11 存量设备迁移最佳实践

场景说明

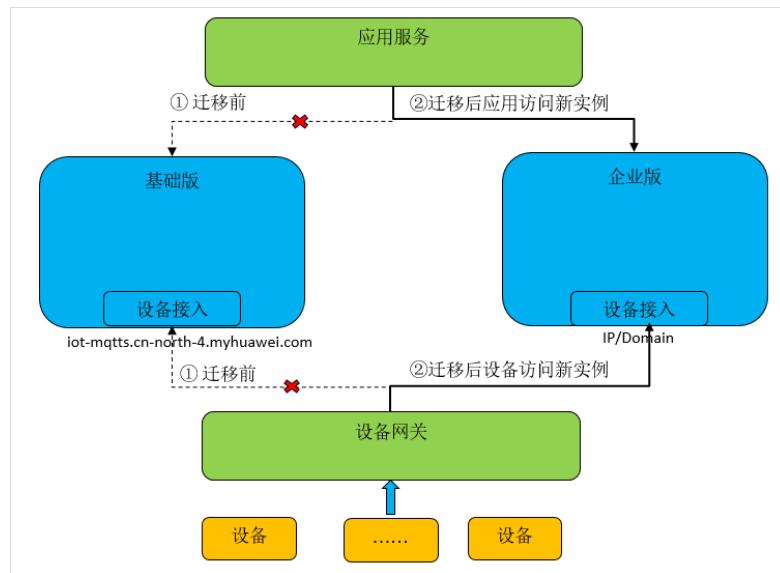
用户在基础版/标准版有一定量的设备，随着设备数量不断增多，购买了IoTDA企业版以提供完全独享的实例，并支持更高的性能规格。支持以下几种场景的设备迁移，本节以**设备从基础版迁移到企业版的迁移场景为例**，介绍如何把设备从基础版迁移到企业版，在新实例上统一管理设备和发放设备。

- 设备从基础版迁移到企业版
- 设备从标准版迁移到企业版
- 设备从基础版迁移到标准版

设备迁移整体流程

通过设备发放服务将基础版的设备数据迁移到企业版，设备数据迁移完成后，应用侧和设备侧访问新实例的地址，往新实例上注册设备、下发命令和上报数据等，在新实例上操作设备和管理设备。

图 2-103 设备迁移整体流程



迁移前提

- 已开通IoTDA基础版，并购买了IoTDA企业版/标准版。
- 用户的应用平台访问IoTDA的地址可以修改，应用平台访问IoTDA从基础版地址改成企业版实例的地址。
- 用户的设备访问IoTDA的地址可以修改，设备访问IoTDA从基础版地址改成企业版实例的地址。如果设备访问IoTDA的地址不能修改(例如设备不支持固件升级)，可以[提交工单](#)说明您的诉求。

⚠ 注意

设备数据迁移时，如果是指定资源空间和设备迁移，迁移数据仅包括设备连接新实例认证需要的身份信息，包括资源空间、产品、证书、设备基本信息和设备状态；暂不包括基础版中的设备分组、服务端订阅、规则引擎和运行日志配置。

操作步骤：

以设备数据从基础版迁移到企业版为例，流程如下：

图 2-104 用户设备迁移操作流程



步骤1 购买企业版或标准版

- 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 在左侧导航栏，单击“IoTDA实例”，查看是否购买了IoTDA企业版或者标准版，如果您未开通新实例，参考[实例开通管理](#)

步骤2 设备数据从基础版迁移到企业版

1. 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
2. 在左侧导航栏，单击“设备发放”，如果您是首次进入设备发放页面，单击右上角“授权开通”对服务进行授权。

图 2-105 设备发放总览



3. 单击“设备数据迁移”进入界面，再单击右上角的“创建同步任务”。

图 2-106 数据同步页面



4. 迁移方式选择“按资源空间整体迁移”，迁移源实例选择北京四基础版，目标实例选择步骤1中购买的企业版实例，单击“确认创建”即可，设备发放会自动把对应资源空间的产品、证书和设备数据迁移过去。

图 2-107 创建迁移任务页面



说明

如果您的设备量较小，可以选择“指定设备迁移”。

- 在“迁移任务创建完成后”，你可以在“设备数据迁移”页面单击“任务ID”或者“详情”，查看迁移任务详情，并且可以使用“高级搜索”对迁移设备进行过滤查询操作。

图 2-108 迁移任务详情页面

| 迁移设备列表 | | | | | |
|--------|-----------------------|--|--------------------------|----------------|-------------------------------|
| 状态 | 设备名称 | 设备ID | 所属产品 | 所属资源空间 | 迁移时间 |
| 已发放 | deviceName_vGHuGRIT2W | 60627e79baek802d73a3a22_xCTC60Mq | WaterMeter_mqtt_0kRgNFT3 | app4mqtt_aiPFC | 2021/04/07 11:02:25 GMT+08:00 |
| 已发放 | deviceName_6uaIy3x1e | 60627e79baek802d73a3a22_K05PhG1ao | WaterMeter_mqtt_0kRgNFT3 | app4mqtt_aiPFC | 2021/04/07 11:02:25 GMT+08:00 |
| 已发放 | deviceName_jNBvXmM4D | 60627e79baek802d73a3a22_p07m7MCPj | WaterMeter_mqtt_0kRgNFT3 | app4mqtt_aiPFC | 2021/04/07 11:02:25 GMT+08:00 |
| 已发放 | deviceName_Mt4Qy4ZLzL | 60627e79baek802d73a3a22_yetTBqAml | WaterMeter_mqtt_0kRgNFT3 | app4mqtt_aiPFC | 2021/04/07 11:02:24 GMT+08:00 |
| 已发放 | deviceName_V0WbLaRvK | 60627e79baek802d73a3a22_p0l0ru0Ec | WaterMeter_mqtt_0kRgNFT3 | app4mqtt_aiPFC | 2021/04/07 11:02:24 GMT+08:00 |
| 已发放 | deviceName_V3sgt8ADyx | 60627e79baek802d73a3a22_zbd1Mrt0f | WaterMeter_mqtt_0kRgNFT3 | app4mqtt_aiPFC | 2021/04/07 11:02:24 GMT+08:00 |
| 已发放 | deviceName_hq4LUzawbF | 60627e79baek802d73a3a22_xTEHko4ZM6 | WaterMeter_mqtt_0kRgNFT3 | app4mqtt_aiPFC | 2021/04/07 11:02:24 GMT+08:00 |
| 已发放 | deviceName_Eg4zr | 60627e79baek802d73a3a22_xTEHko4ZM6_Devicelogftb... | WaterMeter_mqtt_W0593800 | app4mqtt_aiPFC | 2021/04/07 11:02:24 GMT+08:00 |
| 已发放 | deviceName_3MahH856p4 | 60627e79baek802d73a3a22_zzKHqHq51C | WaterMeter_mqtt_0kRgNFT3 | app4mqtt_aiPFC | 2021/04/07 11:02:24 GMT+08:00 |
| 已发放 | deviceName_03GqjQmTj | 60627e79baek802d73a3a22_zsJqyV9ku0 | WaterMeter_mqtt_0kRgNFT3 | app4mqtt_aiPFC | 2021/04/07 11:02:24 GMT+08:00 |

说明

当迁移任务类型为“指定设备迁移”时，此类任务下所有迁移设备均属于同一应用、同一产品。因此在进行迁移设备列表的“高级搜索”时，搜索项“所属产品”和“所属资源空间”均为空。

- 在设备迁移详情页，可以查看单个设备的信息和设备的发放记录。

图 2-109 设备迁移详情页面

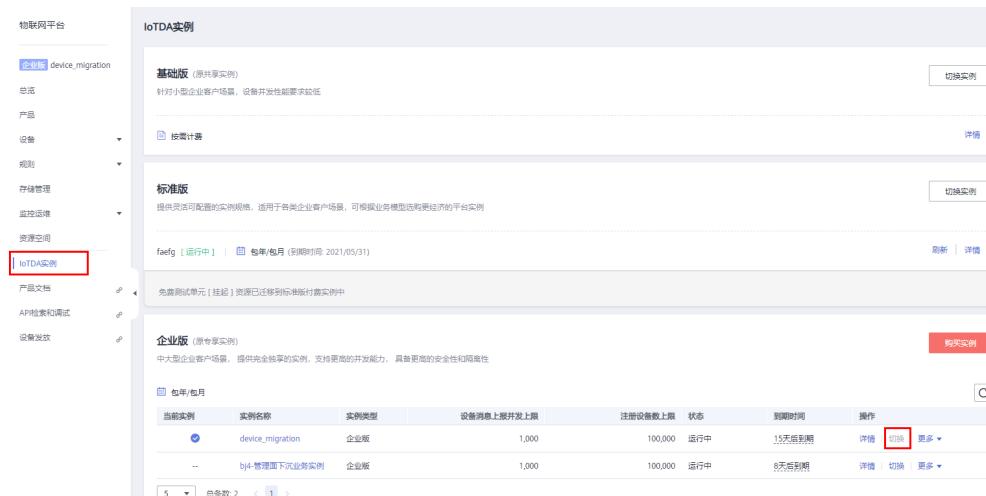


步骤3 企业版配置检查

由于设备迁移任务不会迁移规则引擎和服务端订阅证书，如果你在基础版使用了“规则”和“服务端证书”，需要在企业版再配置一遍。

1. 访问**设备接入服务**，单击“立即使用”进入设备接入控制台。
2. 在左侧导航栏，单击“IoTDA实例”，在企业版实例里单击“切换”，切换到企业版实例。

图 2-110 切换企业版实例



3. 访问**设备接入服务**，单击“立即使用”进入设备接入控制台，在左侧导航栏选择“规则 > 数据转发”，单击左侧“创建规则”配置您需要用到的数据转发规则，详情请参考**数据转发**。

图 2-111 规则配置页面



4. 访问**设备接入服务**，单击“立即使用”进入设备接入控制台，在左侧导航栏选择“规则 > 服务端证书”，单击右上角“上传证书”，详情请参考**加载推送证书**。

图 2-112 上传服务端证书



步骤4 应用侧和设备侧修改访问IoTDA平台的地址，对接到新实例

1. 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
2. 单击设备接入服务左侧导航栏“IoTDA实例”，单击实例详情，获取应用和设备的接入地址。

图 2-113 实例详情页面



3. 修改应用侧和设备侧访问IoTDA的地址。

步骤5 业务拨测和观察。应用侧和设备侧对接到企业版后，可以结合你的业务做一些基本功能验证。例如

1. 通过应用侧注册设备，注册设备接口参考[注册设备](#)
2. 通过设备侧属性上报，设备属性上报参考[设备属性上报](#)
3. 通过应用侧下发命令，下发命令接口参考[平台命令下发](#)

步骤6 增量设备迁移。设备迁移过程中基础版新增的设备不会迁移到企业版，需要对这部分设备做增量迁移，可以参考[步骤2](#)，选择“增量设备迁移”迁移指定的设备

----结束

2.12 MQTT 协议设备 OTA 升级实践

场景说明

MQTT (Message Queuing Telemetry Transport) 是一个基于客户端-服务器的消息发布/订阅传输协议，主要应用于计算能力有限，且工作在低带宽、不可靠的网络的远程传感器和控制设备，适合长连接的场景，如智能路灯等。固件升级功能是网络设备所必须支持的基础服务。在设备固件出现安全漏洞、软件bug维护、自身功能优化、以及设备性能改良等需更新固件时，在线升级功能就尤为重要。本文档通过MQTT.fx模拟固件设备，使用华为云平台对该固件进行设备升级。

说明

软件升级流程和固件升级流程一样，只是在设备上报版本号时对应参数字段不一样。固件升级上报版本号对应参数字段为fw_version，软件升级上报版本号对应参数字段为sw_version,请参考[设备上报版本号](#)

整体流程

图 2-114 升级实践流程

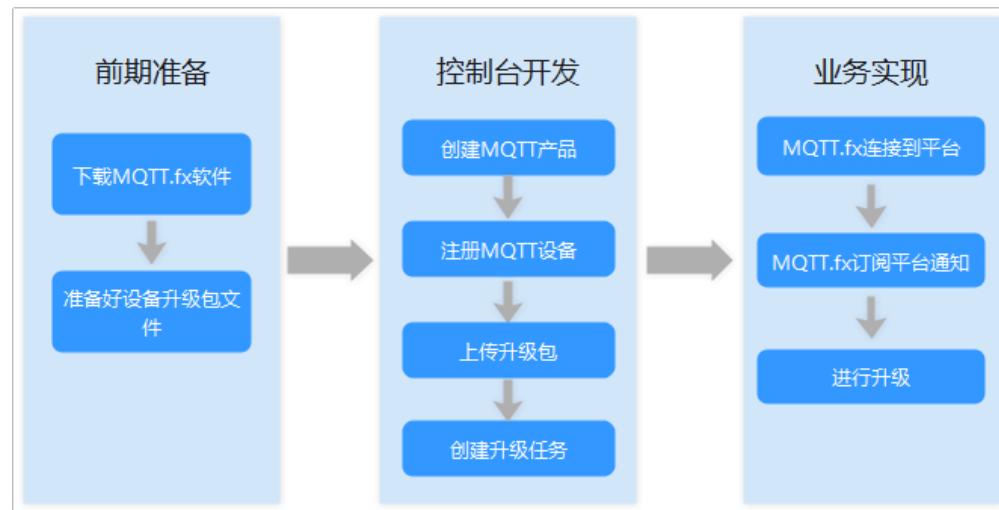
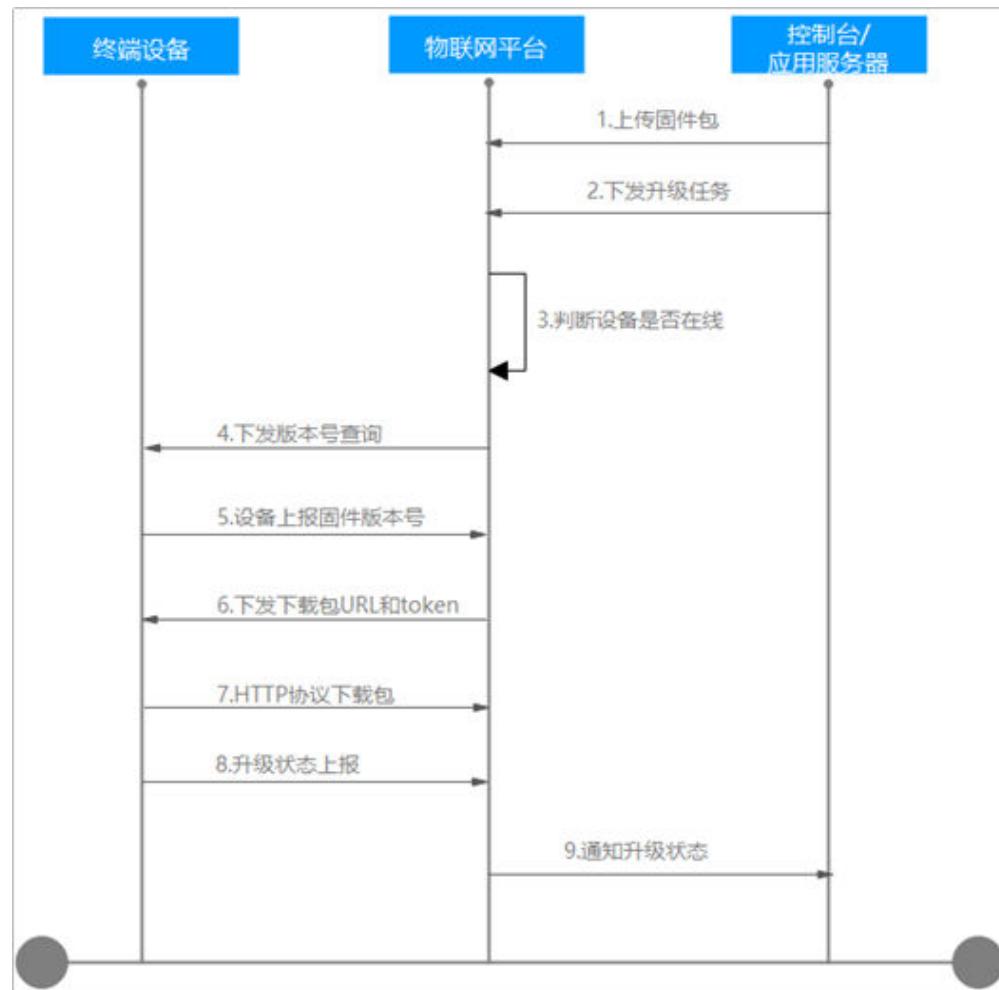


图 2-115 MQTT 设备升级流程



开发环境

- 软件：华为云（已开通[设备接入服务](#)）、win7及以上版本的64位操作系统（本文以windows 10 64-bit系统调试）、[MQTT.fx](#)模拟器。

前提条件

- 已注册华为云官方帐号。未注册可点击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上点击[实名认证](#)，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，点击“立即使用”后开通该服务。

前期准备

- 下载并安装MQTT.fx(应使用1.7.1及以上版本)MQTT.fx安装请参考[使用MQTT.fx调测](#)。
- 升级包文件准备。固件升级包及其版本号需向模组厂家获取。本次演示使用临时文件代替。
- [创建产品](#)，创建MQTT协议产品（如果有MQTT协议产品可直接使用）。

- **注册单个设备。**
- **设备群组**, 将注册的设备加入一个设备群组(创建升级任务时会用到)。

上传升级包

- 步骤1** 访问**设备接入服务**, 单击“立即使用”进入设备接入控制台。
- 步骤2** 在左侧导航栏选择“设备 > 软固件升级”。
- 步骤3** 选择“资源包管理”页签, 选择“固件列表”。
- 步骤4** 单击“上传固件”, 在弹出的页面中有两种上传方式, OBS文件上传和本地文件上传。

图 2-116 上传升级包-OBS 文件

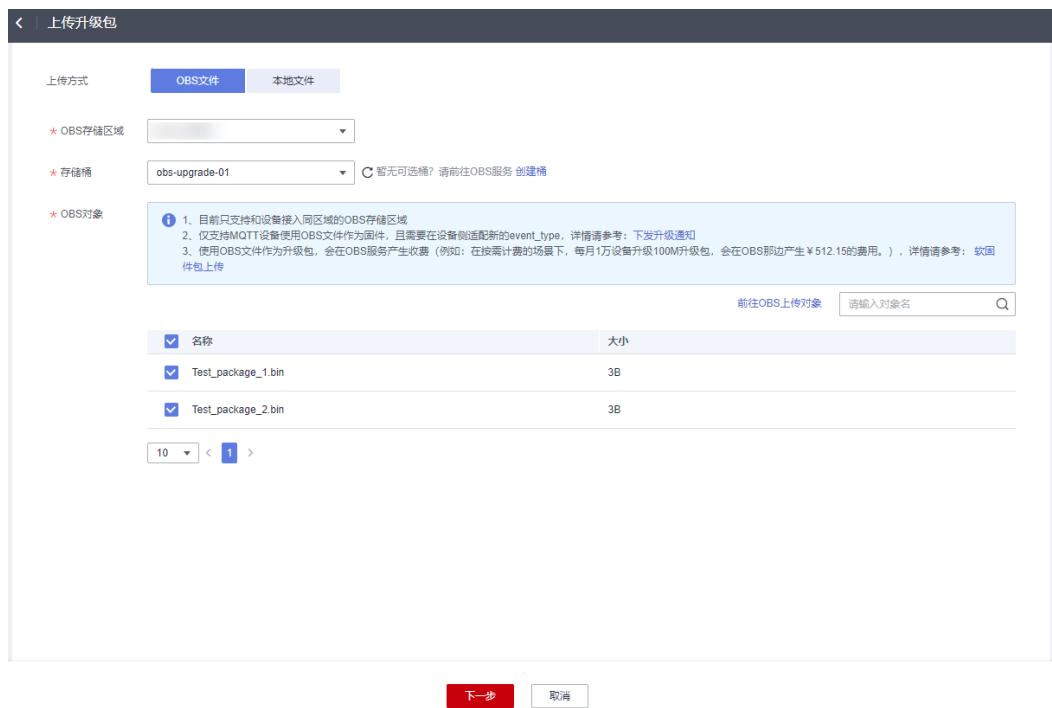


图 2-117 上传升级包-本地文件



步骤5 上传升级包后, 填写升级包参数信息, 单击“确定”完成升级包上传。

图 2-118 上传升级包-OBS 文件参数

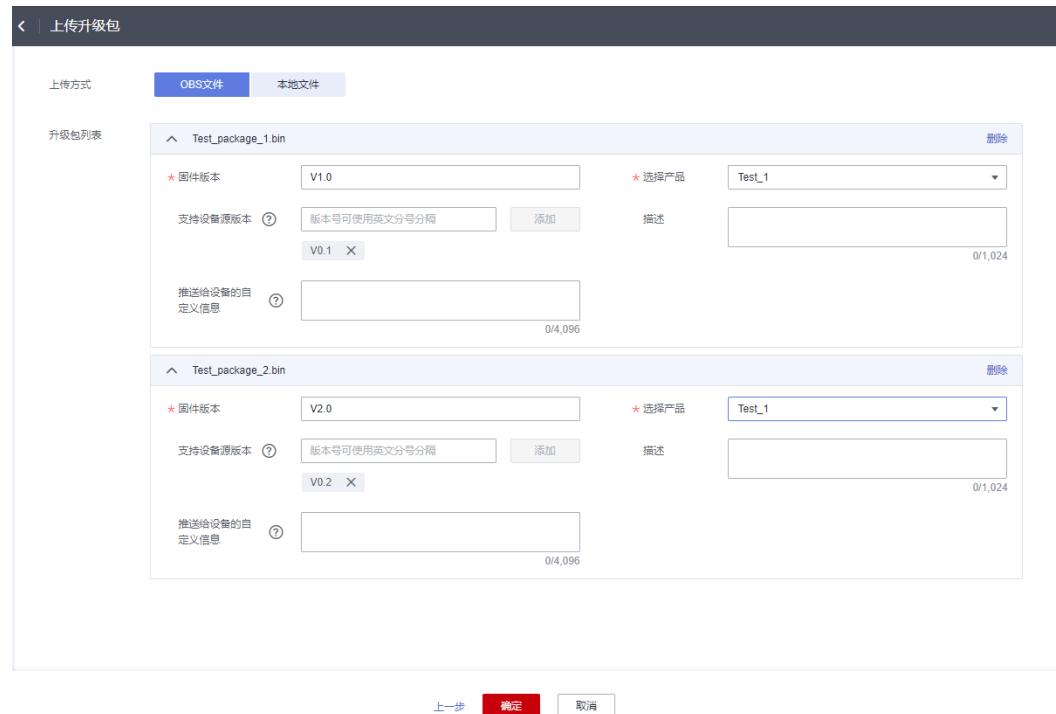
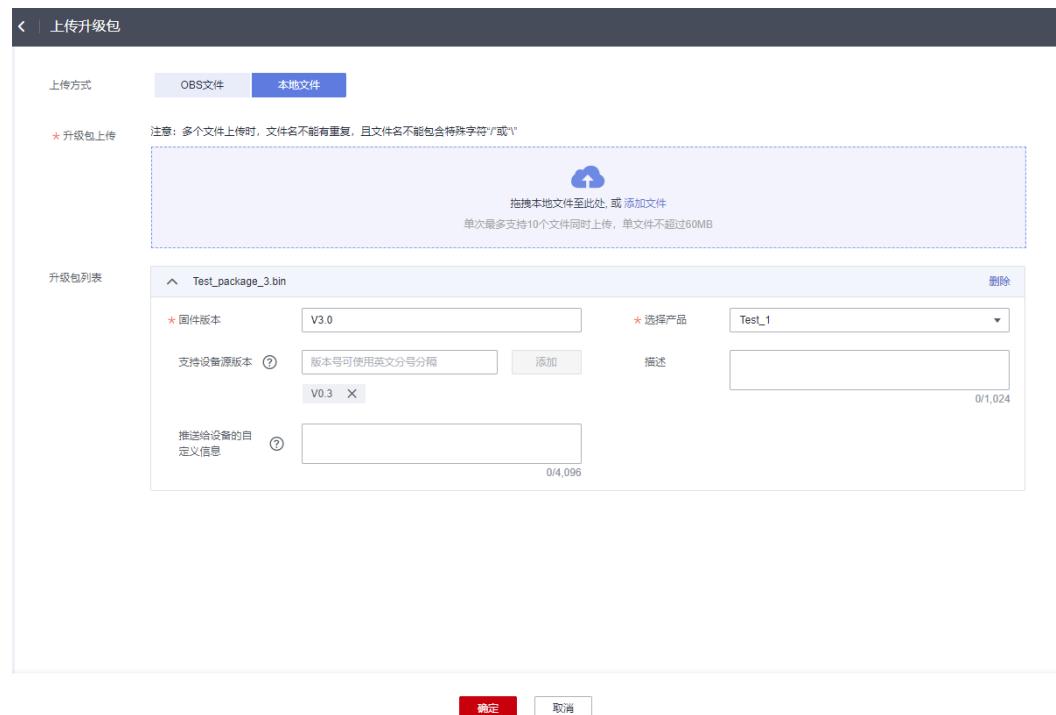


图 2-119 上传升级包-本地文件参数



步骤6 升级包上传完成后会在"固件列表"中显示出来。

图 2-120 软件升级-固件列表

The screenshot shows a software interface titled 'Firmware Upgrade'. At the top, there are tabs for 'Resource Management' and 'Upgrade Task'. Below the tabs, there are two sub-tabs: 'Firmware List' and 'Software List'. A note states: '固件一般是指设备硬件的底层驱动程序，承担将一个系统最基础最底层工作的软件，比如计算机主板上的基本输入输出系统BIOS (Basic Input/output System)'.

| 固件ID | 固件名称 | 版本 | 产品ID | 产品名称 | 上传时间 | 操作 |
|------------|--------------------|------|--------|--------|-------------------------------|----------------------------|
| [Redacted] | Test_package_3.bin | V3.0 | Test_1 | Test_1 | 2024/01/12 10:33:45 GMT+08:00 | [View] [Download] [Delete] |
| [Redacted] | Test_package_2.bin | V2.0 | Test_1 | Test_1 | 2024/01/12 10:32:36 GMT+08:00 | [View] [Download] [Delete] |
| [Redacted] | Test_package_1.bin | V1.0 | Test_1 | Test_1 | 2024/01/12 10:32:36 GMT+08:00 | [View] [Download] [Delete] |

At the bottom left, there are buttons for 'Upload Firmware' and 'Search'. On the right, there are filter and search icons.

步骤7 选择“升级任务”，在“固件升级”页签，单击“新建任务”按钮，填写参数信息，选择升级包，选择需要升级的设备，单击“立刻创建任务”。

图 2-121 新建固件升级任务-Test_upgrade_1

The screenshot shows a dialog box titled 'Create New Upgrade Task'. It has two main sections: 'Basic Information' and 'Select Upgrade Package'.

Basic Information:

- 任务名称: Test_upgrade_1
- 执行时间: 立刻执行 (selected)
- 单设备任务延时时间: 30 天
- 启用重试:
- 单设备升级延时时间:

Select Upgrade Package:

请选择固件以及升级目标版本

| 高级搜索 | | | | | | | |
|------------|--------------------|------|-------|--------|----|------|-------------|
| 固件ID | 固件名称 | 版本 | 来源 | 所属产品 | 描述 | 源版本 | 推送至设备的自定... |
| [Redacted] | Test_package_3.bin | V3.0 | IoTDA | Test_1 | | V0.3 | |
| [Redacted] | Test_package_2.bin | V2.0 | OBS | Test_1 | | V0.2 | |
| [Redacted] | Test_package_1.bin | V1.0 | OBS | Test_1 | | V0.1 | |

At the bottom right, there are buttons for '立即创建任务' (Create Task Now) and '取消' (Cancel).

说明

创建升级任务时，可以选择多个升级包（最多10个），选择的升级包必须满足支持的源版本没有重复。如果某个升级包没有指定支持源版本，则默认该升级包支持所有源版本设备升级。

步骤8 任务创建成功后可以在任务列表中查询到该任务，点击详情可以查看当前任务详情。在任务详情页面可以批量停止（一次最多100个设备）或单个停止正在执行中的设备升级，也可以批量重试（一次最多100个设备）或单个重试失败的设备升级。点击“全部重试”可以重试该批量升级任务中所有失败的设备升级。

图 2-122 任务详情-Test_upgrade_1

任务详情

基本信息

| | | | |
|-----------|-------------------------------|-----------|--------------------|
| 任务名称 | Test_upgrade_1 | 状态 | 执行中 |
| 执行时间 | 立刻执行 | 任务类型 | 固件升级 |
| 开始时间 | 2024/01/12 10:35:59 GMT+08:00 | 重试策略 | 重试0次 重试间隔 (分钟) 0 |
| 单设备升级超时时间 | 30分钟 | 单设备任务超时时间 | 30天 |

执行详情 已选升级包

| | | | | | | | |
|--------|---|-------|---|-------|---|----|---|
| 升级设备总数 | 1 | 升级成功数 | 0 | 升级失败数 | 0 | 其他 | 1 |
|--------|---|-------|---|-------|---|----|---|

| <input type="checkbox"/> 状态 | 设备ID | 升级描述 | 操作 |
|---|--------------------|-----------------|---------|
| <input checked="" type="checkbox"/> 执行中 | Test_1_Test_MQTT_1 | QueryingVersion | 重试 停止 |

10 总条数: 1 < 1 >

----结束

业务实现

步骤1 使用MQTT.fx模拟设备接入平台，详情请参考[连接鉴权](#)。

步骤2 用MQTT.fx订阅平台下行信息Topic，MQTT.fx会收到平台服务端下发的查询版本命令。

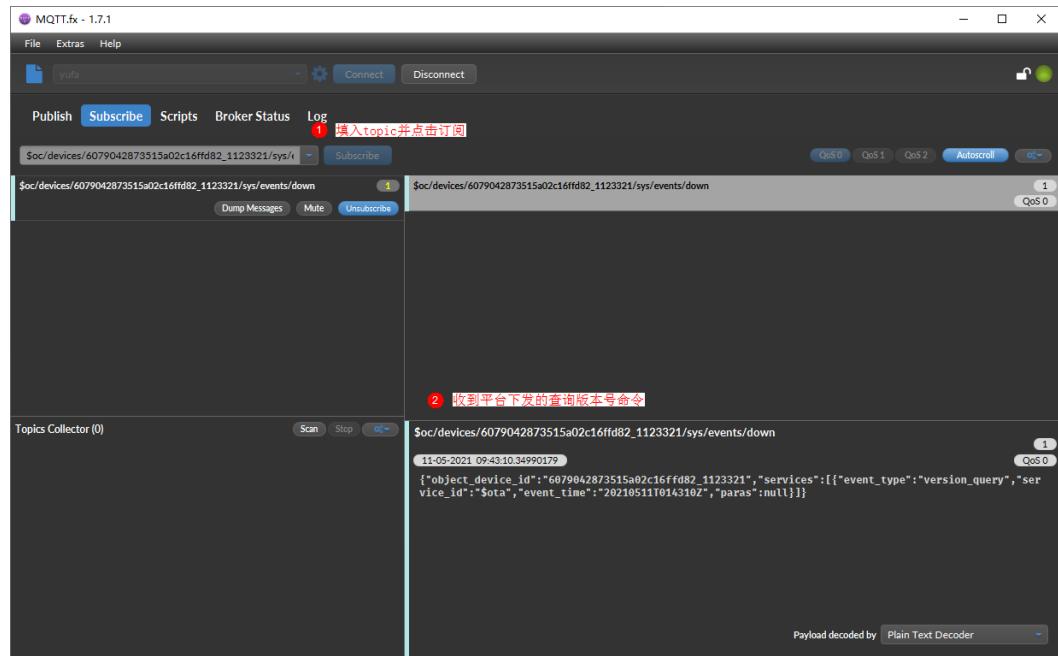
Topic

下行: \$oc/devices/{device_id}/sys/events/down

参数说明

请参考[平台下发获取版本信息通知](#)

图 2-123 订阅通知



步骤3 通过MQTT.fx上报软固件版本信息。

Topic

上行: \$oc/devices/{device_id}/sys/events/up

参数说明

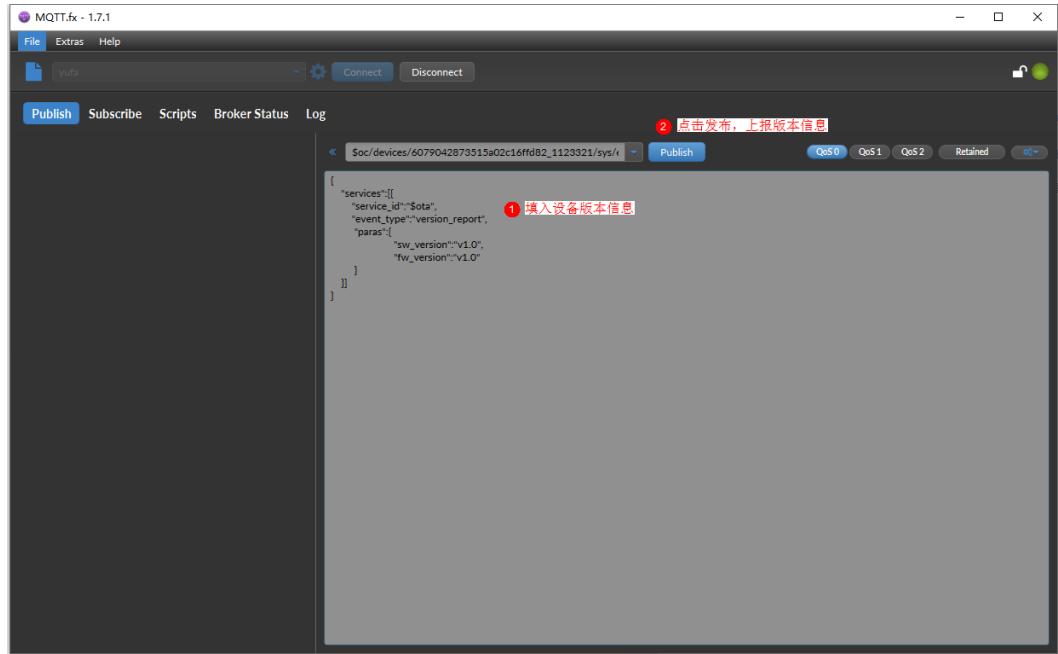
请参考[设备上报软固件版本](#)

示例

Topic: \$oc/devices/{device_id}/sys/events/up

数据格式: {
 "object_device_id": "{object_device_id}",
 "services": [
 {
 "service_id": "\$ota",
 "event_type": "version_report",
 "event_time": "20151212T121212Z",
 "paras": {
 "sw_version": "v1.0",
 "fw_version": "v1.0"
 }
 }
]
}

图 2-124 上报版本号



步骤4 上报完版本号后，模拟器会接收到平台下发的升级通知，通知信息如下：

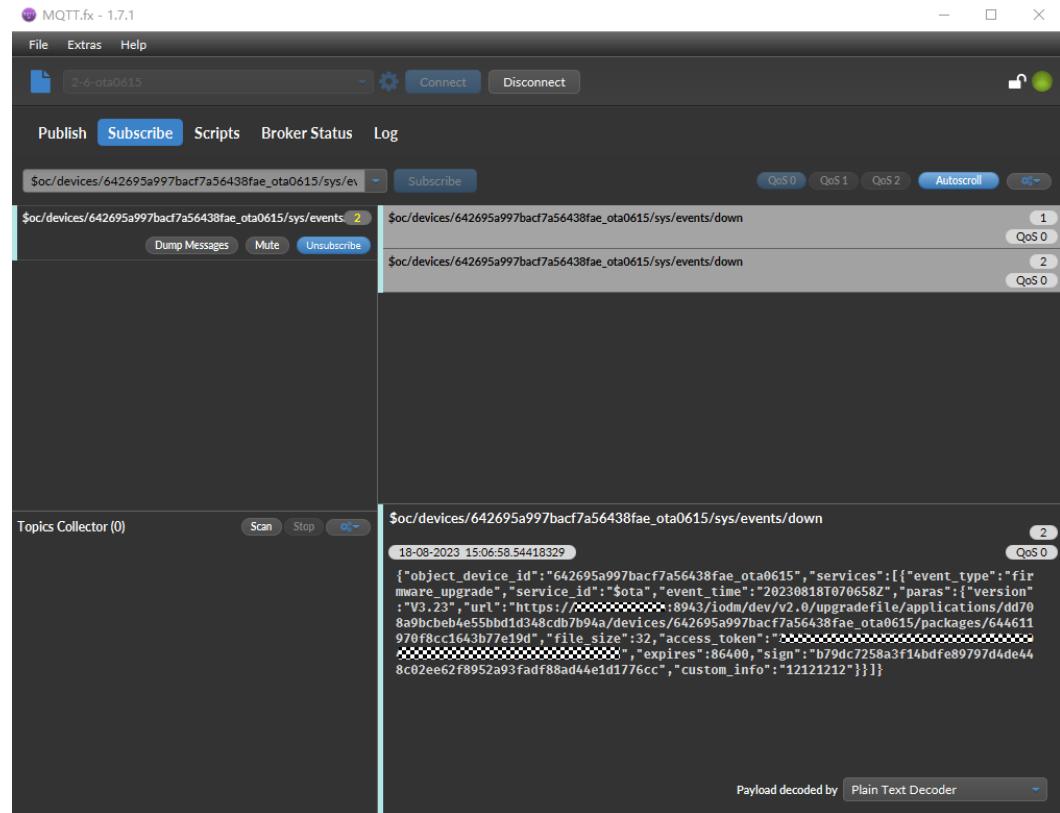
Topic

上行: \$oc/devices/{device_id}/sys/events/down

参数说明

请参考[平台下发升级通知](#)

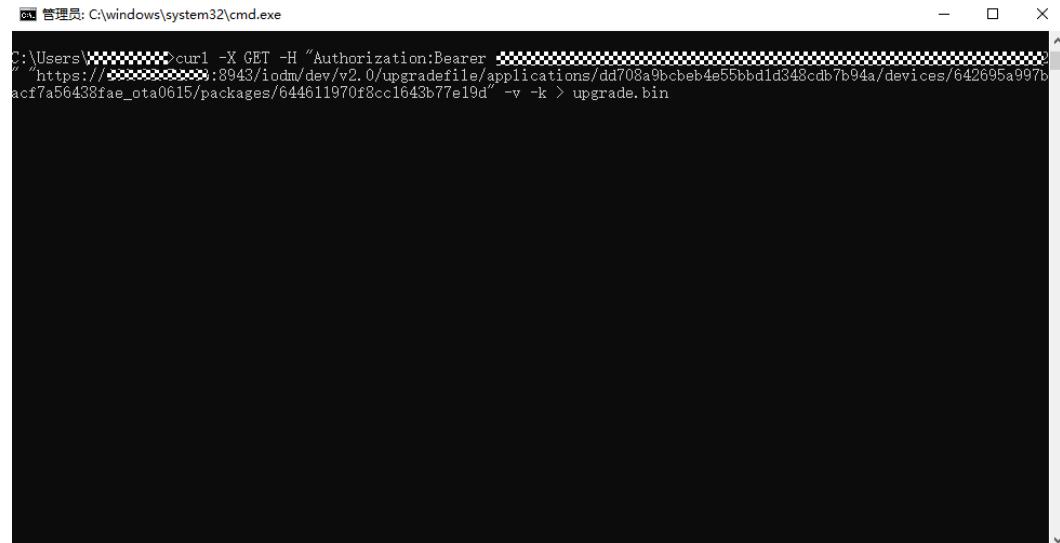
图 2-125 获取升级通知



步骤5 设备收到升级通知后，可通过Http请求下载升级包。

本次演示使用Curl命令代替。

图 2-126 下载升级包



示例

```
curl -X GET -H "Authorization:Bearer *****" "https://100.93.28.202:8943/iodm/dev/v2.0/upgradefile/applications/ddecccc223574aee9466fe8f7e16a205/devices/6079042873515a02c16ffd82_123456789/packages/4f201f38c281ca5d40794a3f" -v -k
```

⚠ 注意

- 应在http附加请求消息头（ header ）字段附加Authorization信息，其值为“Bearer {access_token}”，其中{access_token}为收到的升级通知中的access_token。Bearer和{access_token}之间有一个空格。
- 如果event_type为firmware_upgrade_v2、software_upgrade_v2，则在请求下载软件包时不需要携带请求头。请求示例如下：

```
GET https://*****.obs.cn-north-4.myhuaweicloud.com:443/test.bin?  
AccessKeyId=DX5G7W*****
```

步骤6 设备上报升级状态**Topic**

上行: \$oc/devices/{device_id}/sys/events/up

参数说明

请参考[设备上报升级状态](#)

示例

```
Topic: $oc/devices/{device_id}/sys/events/up  
数据格式:  
{  
    "object_device_id": "{object_device_id}",  
    "services": [  
        {"service_id": "$ota",  
         "event_type": "upgrade_progress_report",  
         "event_time": "20151212T121212Z",  
         "paras": {  
             "result_code": 0,  
             "progress": 50,  
             "version": "V1.0",  
             "description": "upgrade processing"  
         }  
     ]  
}
```

如下图所示上报升级进度为50%，可以在平台查询到升级进度为50%。

图 2-127 上报升级进度 50%

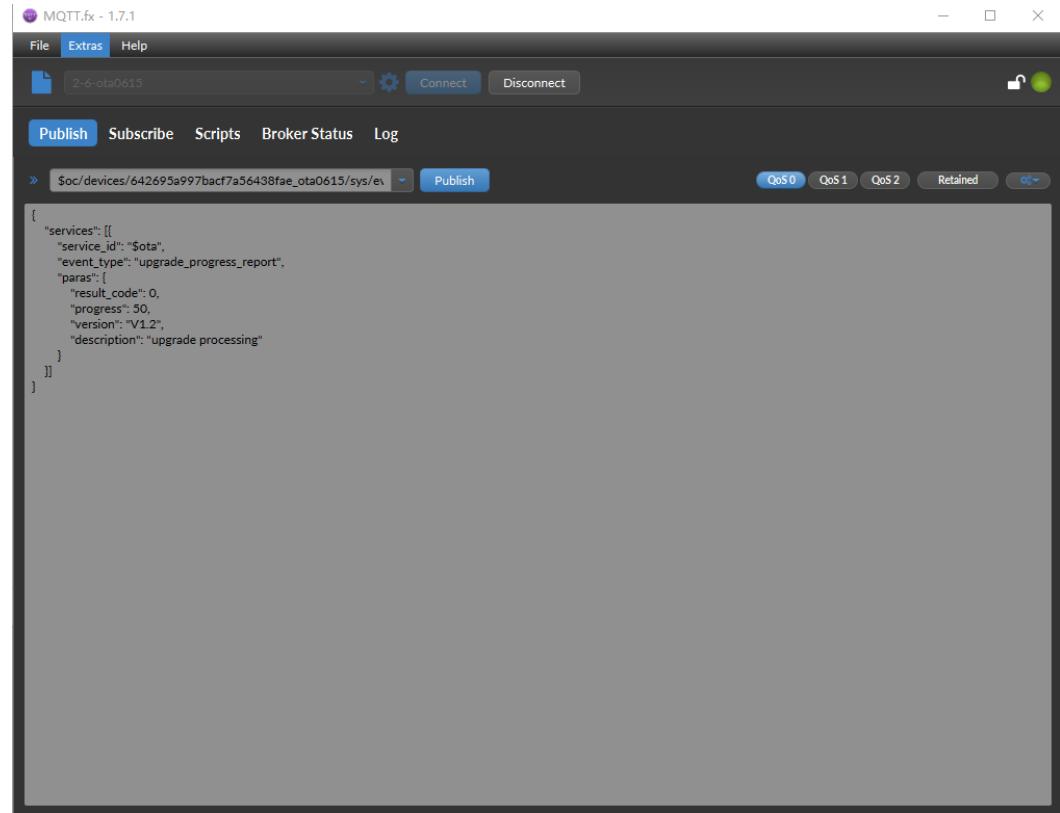


图 2-128 任务详情-Test_upgrade_1-升级进度

| 基本信息 | 任务名称 | 状态 | 操作 |
|----------------|-------------------------------|-----------|--------------------|
| Test_upgrade_1 | 执行中 | 固件升级 | |
| 执行时间 | 立刻执行 | 任务类型 | |
| 开始时间 | 2024/01/12 10:35:59 GMT+08:00 | 重试策略 | 重试0次 重试间隔 (分钟) 0 |
| 单设备升级超时时间 | 30分钟 | 单设备任务超时时间 | 30天 |

| 执行详情 | 已选升级包 | | |
|---|--|-----------------------------|---------------------------------------|
| | 升级设备总数: 1 升级成功数: 0 升级失败数: 0 其他: 1 | | |
| <input type="button" value="全部重试"/> <input type="button" value="批量重试"/> <input type="button" value="批量停止"/> | <input type="button" value="所有状态"/> <input type="text" value="请输入设备ID"/> <input type="button" value="搜索"/> <input type="button" value="清空"/> | | |
| 状态 | 设备ID | 升级描述 | 操作 |
| <input type="checkbox"/> 执行中 | Test_1_Test_MQTT_1 | downloading progress is 50% | 重试 停止 |

总条数: 1 < 1 >

步骤7 完成升级

上报升级进度为100%，且当前版本为升级的目标版本，可以在平台查询到升级完成。

图 2-129 上报升级进度 100%

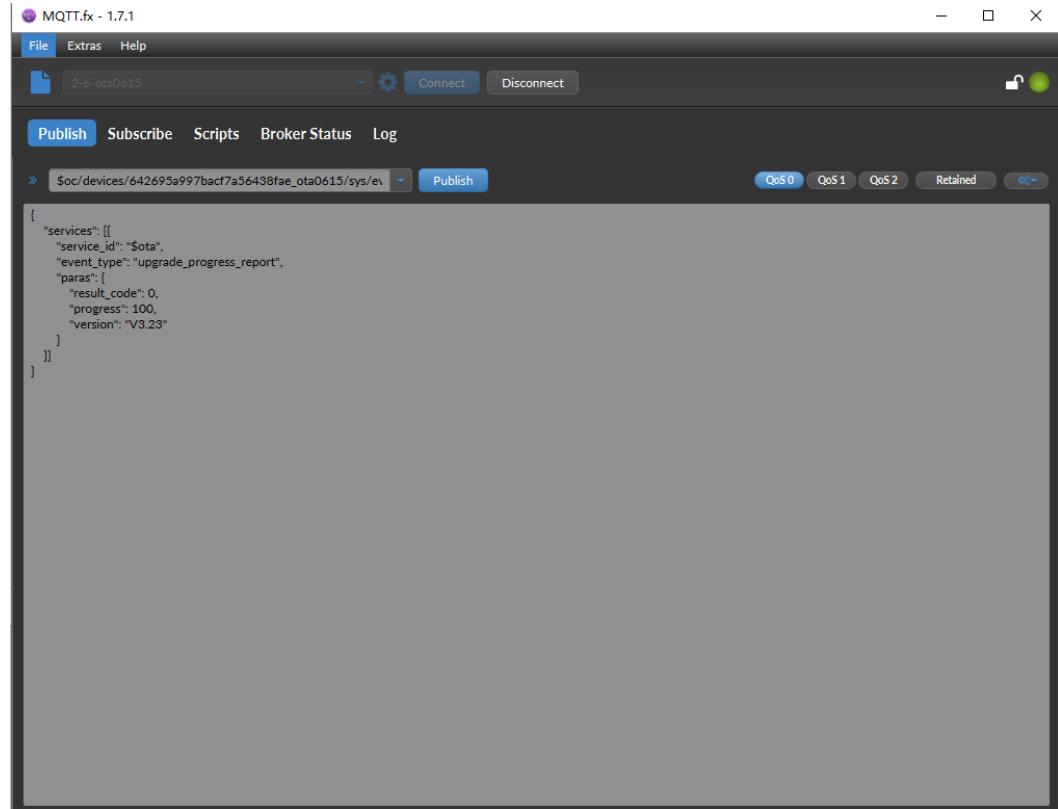


图 2-130 软固件升级-固件升级



----结束

2.13 设备通过微信小程序接入到平台

场景说明

平台支持基于WebSocket的MQTT协议，使基于浏览器的应用程序或者小程序可以快速接入到物联网平台。本文以“智慧路灯”为例，通过微信小程序模拟设备，带您体验设备通过MQTT over WebSocket协议连接到物联网平台、上报数据和接收命令的全过程。

假设：

路灯设备上报一条包含光照强度（luminance）的消息；路灯支持远程控制开关状态的命令（switch），路灯上报的数据格式为JSON格式。

前提条件

- 已注册华为云官方帐号。未注册可单击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击[实名认证](#)完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“立即使用”后开通该服务。

业务流程

具体步骤如下：

步骤1：创建产品。 创建一个MQTT协议的产品。

步骤2：开发产品模型。 通过定义产品模型，在物联网平台构建一款路灯设备，支持上报光照强度、下发路灯开关状态命令。

步骤3：注册设备。 注册一个MQTT协议的设备，体验数据上报业务。

步骤4：使用微信小程序接入。 使用微信小程序将设备连接到物联网平台。

创建产品

产品是设备的合集，您可以将相同能力或特征的设备归属在同一个产品下。

步骤1 登录[管理控制台](#)，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

步骤2 根据页面提示填写参数，然后单击“立即创建”。

| 基本信息 | |
|--------|--|
| 所属资源空间 | 平台自动将新创建的产品归属在默认资源空间下。如需归属在其他资源空间下，下拉选择所属的资源空间。如无对应的资源空间，请先创建 资源空间 。 |
| 产品名称 | 自定义，如Streetlight。支持字母、数字、下划线（_）、连字符（-）的字符组合。 |
| 协议类型 | 选择MQTT。 |
| 数据格式 | 选择JSON。 |
| 所属行业 | 请根据产品模型的所属行业填写。 |
| 设备类型 | Streetlight |

----结束

开发产品模型

步骤1 找到[创建产品](#)章节新增的产品，单击产品进入产品界面。

步骤2 在产品详情“模型定义”页面，单击“自定义模型”，配置产品的服务。

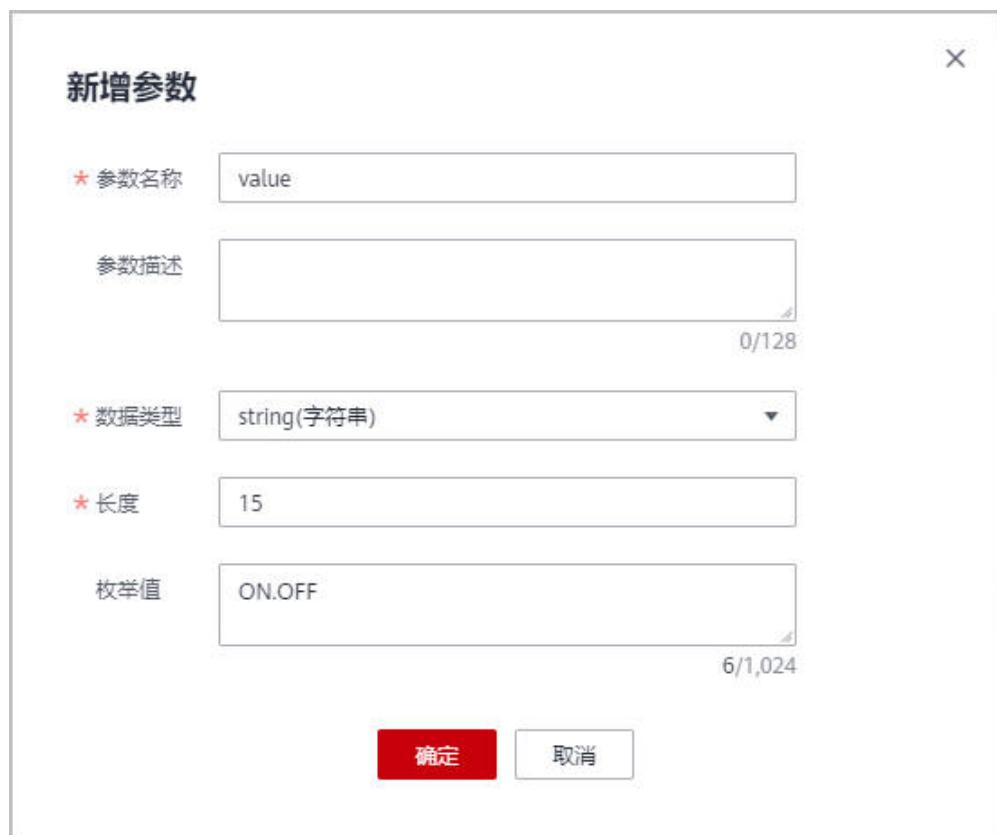
步骤3 添加服务“BasicData”。

1. 在“添加服务”页面，填写“服务ID”、“服务类型”和“服务描述”，单击“确定”。
 - “服务ID”：BasicData
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：上报路灯数据
2. 在“**BasicData**”的下拉菜单下单击“添加属性”，填写相关信息后，单击“确定”。
 - “属性名称”：luminance
 - “数据类型”：int（整型）
 - “访问权限”：可读、可写
 - “取值范围”：0~65535
 - “步长”：0
 - “单位”：不填写

步骤4 添加服务类型“LightControl”。

1. 在“模型定义”下单击“添加服务”，填写“服务ID”、“服务类型”和“服务描述”，单击“确定”。
 - “服务ID”：LightControl
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：路灯开关控制
2. 在“**LightControl**”的下拉菜单下单击“添加命令”，输入命令名称“Switch”。
3. 在“新增命令”页面，单击“新增输入参数”，填写相关信息后，单击“确定”。

图 2-131 新增命令参数-value



----结束

注册设备

- 步骤1** 在管理控制台页面，选择左侧导航栏“设备 > 所有设备”，单击页面右上角的“注册设备”。
- 步骤2** 根据页面提示信息填写参数，然后单击“确定”。

| 参数名称 | 说明 |
|--------|-----------------------------------|
| 所属资源空间 | 确保和创建的产品归属在同一个资源空间。 |
| 所属产品 | 选择 步骤2 创建的产品。 |
| 设备标识码 | 即nodeID，设备唯一物理标识。可自定义，由英文字母和数字组成。 |
| 设备名称 | 即device_name，可自定义，如streetlight。 |
| 设备认证类型 | 选择“密钥”。 |
| 密钥 | 此处不填写，物联网平台会自动生成。 |

成功注册设备后，平台会自动生成设备ID和密钥，请妥善保管好设备ID（deviceId）和密钥（deviceSecret），用于设备接入。

图 2-132 注册设备成功



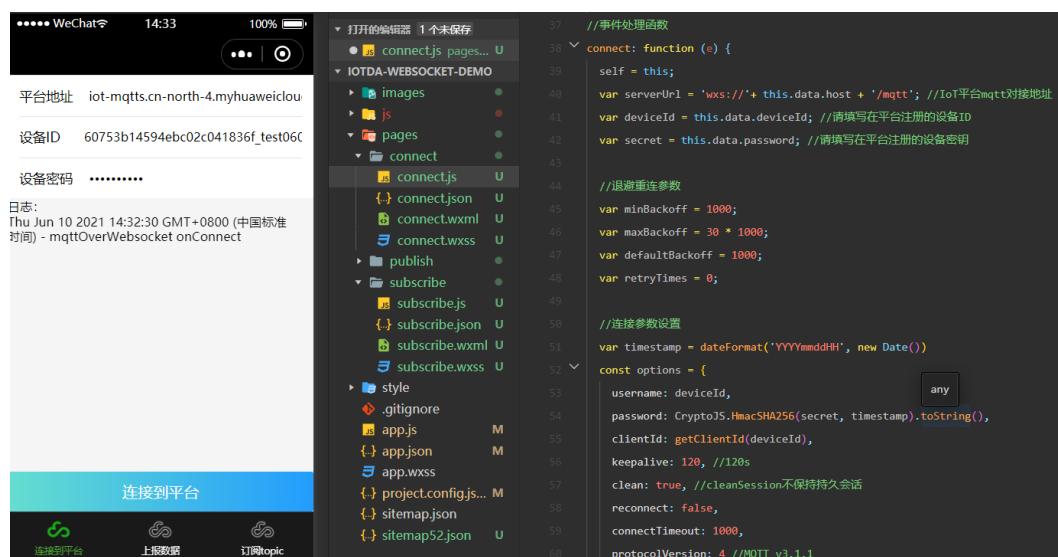
----结束

使用微信小程序接入

步骤1 下载[微信小程序示例工程](#)。

步骤2 下载并安装[微信开发者工具](#)，使用微信开发者工具打开微信小程序示例工程，如下图所示。

图 2-133 微信小程序示例工程



小程序提供了界面显示，填写Server地址、设备ID、设备密钥。请根据实际创建的设备信息填写。

- Server地址：即域名，参考[平台对接信息](#)获取；
- 设备ID和设备密钥：在[这里](#)获取。

步骤3 微信小程序连接到物联网平台

开发者通过传入设备信息，将设备连接到物联网平台。

说明

真机调测时请在微信开发者工具-详情-本地设置-不校验合法域名，将其取消勾选，再在微信公众平台-小程序管理员界面-开发管理-服务器域名中添加socket合法域名（'wss://'+MQTT接入地址）。

图 2-134 连接到平台



单击连接到平台按钮，在域名、设备ID和密钥正确的情况下，可以看到日志设备连接成功，可在平台查看设备状态，如下图：

图 2-135 设备在线



步骤4 发布Topic

发布Topic是指设备主动向平台上报自己的属性或消息，详细见[设备属性上报接口](#)文档。

在微信小程序中实现了上报属性数据的功能。

发布Topic后，Demo界面显示如下：

图 2-136 上报数据



设备上报属性成功后可在[设备详情](#)页面查看到上报的属性：

图 2-137 查看上报数据-MQTT



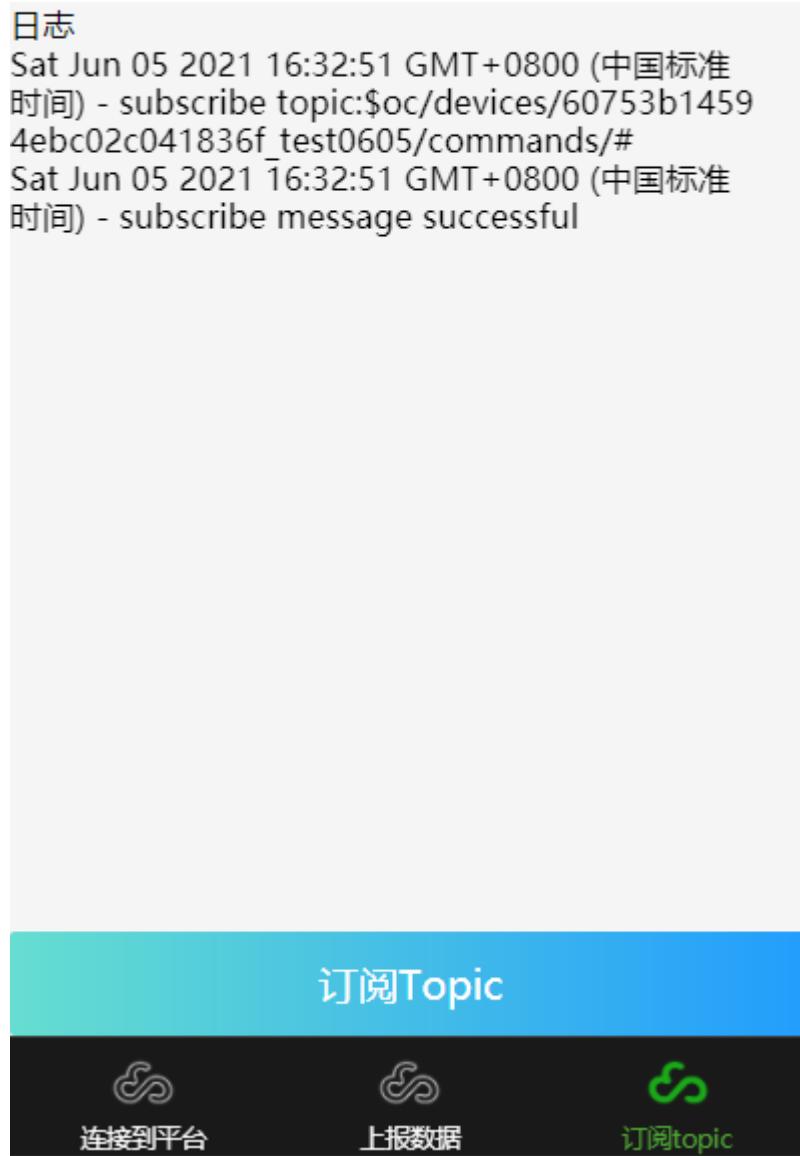
步骤5 订阅Topic

订阅某topic的设备才能接收平台发布的关于该topic的消息，关于平台预置topic可参考[Topic定义](#)。

建链后，如果成功订阅Topic，主界面日志栏显示如下信息：

图 2-138 订阅 topic 界面

topic \$oc/devices/60753b14594ebc02c041



步骤6 接收下发命令

小程序支持接收平台下发命令的功能，在MQTT建链完成并成功订阅Topic后，可以在[控制台](#)设备详情中对该设备进行同步命令下发。下发成功后，在MQTT的回调函数中接收到平台下发给设备的命令。

例如下发参数名为LightControl: Switch，参数值为ON的命令。

图 2-139 命令下发-MQTT



同步命令下发成功后，Demo界面显示如下

图 2-140 接收命令界面



----结束

2.14 基于 MQTT.fx 的 X.509 证书接入指导

本文档以MQTT.fx为例，介绍以MQTT原生协议接入物联网平台。MQTT.fx是目前主流的MQTT客户端，可以快速验证是否可以与物联网平台服务交互发布或订阅消息。

X.509是一种用于通信实体鉴别的数字证书，物联网平台支持设备使用自己的X.509证书进行认证鉴权。使用X.509认证技术时，设备无法被仿冒，避免了密钥被泄露的风险。

前提条件

- 已注册华为云官方帐号。未注册可单击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击[实名认证](#)完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击[管理控制台](#)后开通该服务。

限制说明

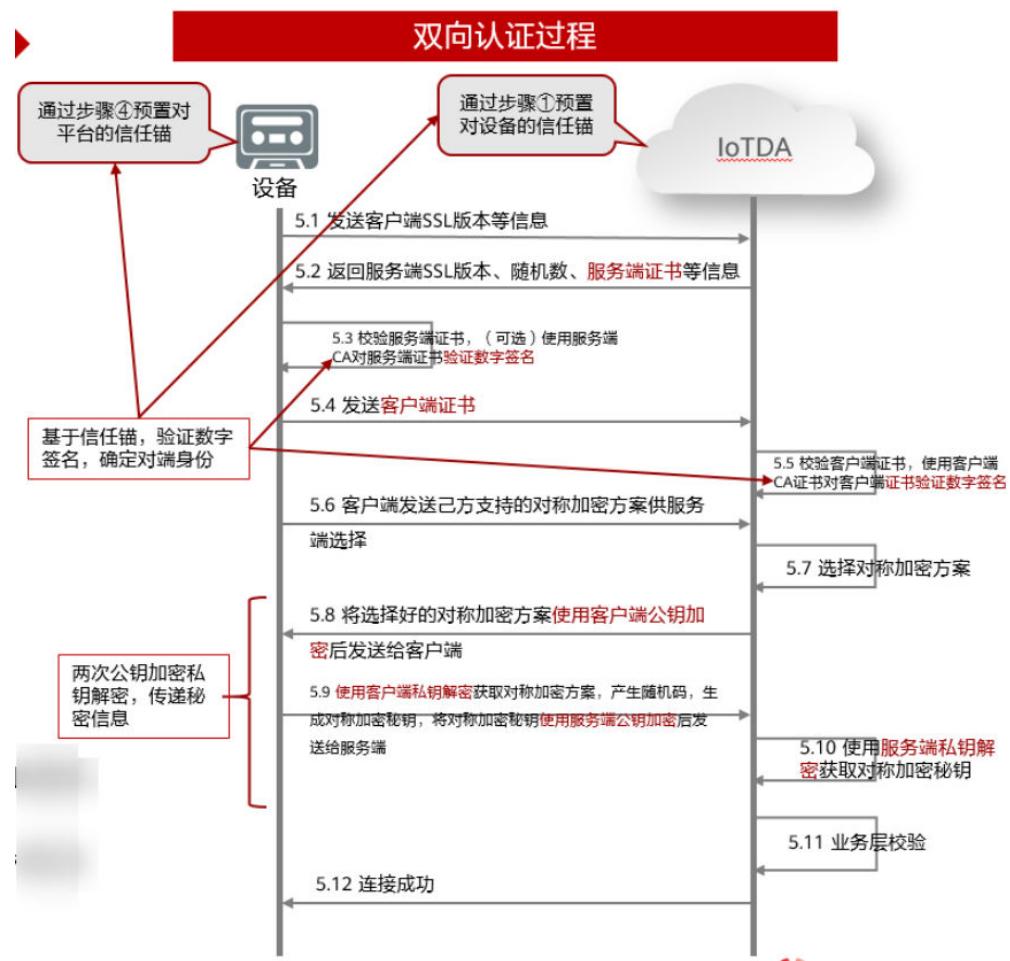
- 当前只有通过MQTT接入的设备支持使用X.509证书进行设备身份认证。
- 每个用户最多上传100个设备CA证书。

X509 认证步骤

- 证书双向认证完整业务流程为：



- 证书双向认证过程为：



获取设备接入信息

在设备接入服务控制台获取设备接入平台所需的信息。

步骤1 访问**设备接入服务**，单击**管理控制台**进入控制台。

步骤2 单击**管理控制台**进入控制台，单击左侧导航栏的“总览”，点击“实例基本信息 - 接入信息”查看设备接入信息，记录域名和端口。

图 2-141 进入平台接入地址



说明

针对不支持用域名接入的设备，通过在cmd命令框中执行“ping 域名”获取IP地址，用IP地址接入平台。由于IP地址不固定，您需要将IP地址做成可配置项。

----结束

创建产品

步骤1 登录[管理控制台](#)，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

步骤2 根据页面提示填写参数，然后单击“确定”。

| 基本信息 | |
|--------|--|
| 所属资源空间 | 平台自动将新创建的产品归属在默认资源空间下。如需归属在其他资源空间下，下拉选择所属的资源空间。如无对应的资源空间，请先创建 资源空间 。 |
| 产品名称 | 自定义。支持字母、数字、下划线（_）、连字符（-）的字符组合。 |
| 协议类型 | 选择MQTT。 |
| 数据格式 | 选择JSON。 |
| 所属行业 | 请根据实际情况填写。若使用平台预置的产品模型，请根据产品模型所属的行业填写。没有填无。 |
| 设备类型 | 请根据实际情况填写。若使用平台预置的产品模型，会自动关联设备类型，不需要再输入设备类型。 |
| 高级配置 | |
| 产品ID | 定制ProductID，用于唯一标识一个产品。如果携带此参数，平台将产品ID设置为该参数值；如果不携带此参数，产品ID在物联网平台创建产品后由平台分配获得。 |

图 2-142 创建产品

The screenshot shows the 'Create Product' dialog box. At the top right is a close button (X). The title bar says '创建产品'. The form fields include:

- '所属资源空间' (必填) dropdown.
- '产品名称' (必填) input field.
- '协议类型' (必填) dropdown set to 'MQTT'.
- '数据格式' (必填) dropdown set to 'JSON'.
- '设备类型选择' section with two buttons: '标准类型' (highlighted in blue) and '自定义类型'.
- '所属行业' (必填) dropdown with placeholder '在设备类型下选择设备后可自动匹配'.
- '所属子行业' dropdown.
- '设备类型' (必填) dropdown.
- '高级配置' dropdown with placeholder '定制ProductID | 备注信息'.

At the bottom right are '确定' (Confirm) and '取消' (Cancel) buttons.

----结束

开发产品模型

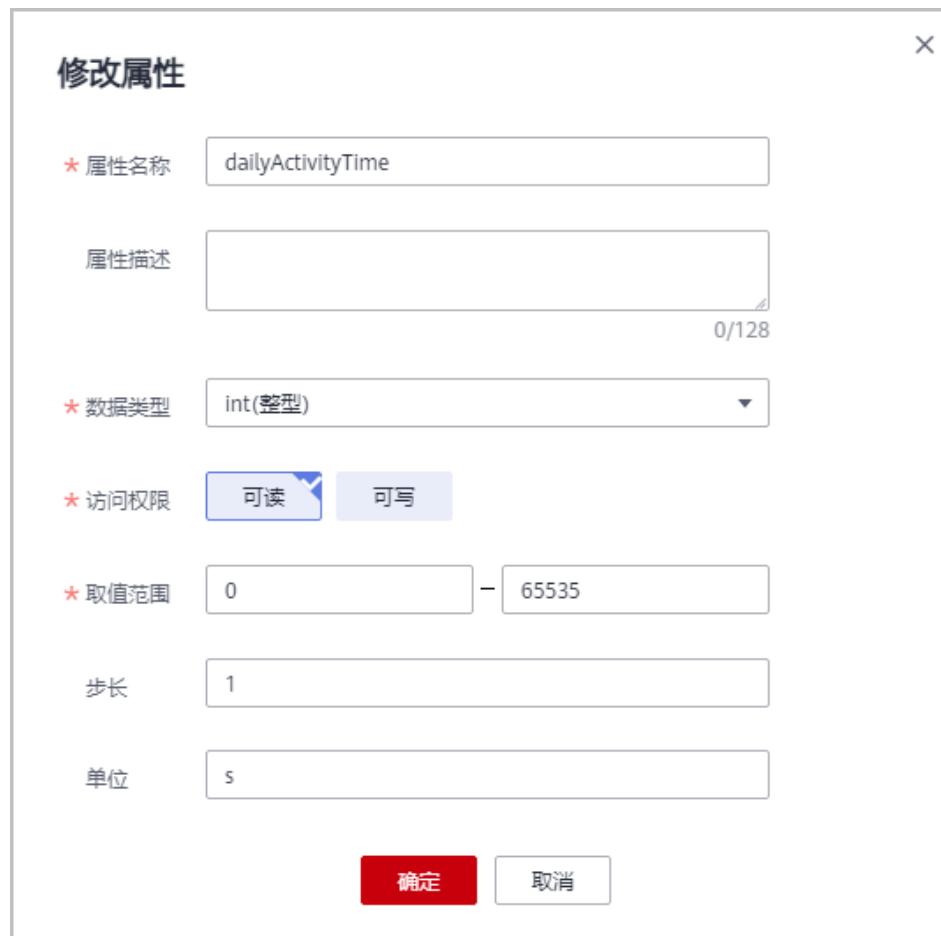
- 步骤1** 找到新增的产品，单击“产品”进入产品界面。
- 步骤2** 在产品详情“模型定义”页面，单击“自定义模型”，配置产品的服务。
- 步骤3** 新增服务类型“Connectivity”。
 1. 在“添加服务”页面，填写“服务ID”、“服务类型”和“服务描述”，单击“确定”。
 - “服务ID”：Connectivity
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：Connectivity

图 2-143 添加服务



2. 在“Connectivity”的下拉菜单下单击“添加属性”，填写相关信息后，单击“确定”。
 - “属性名称”：dailyActivityTime
 - “数据类型”：int（整型）
 - “访问权限”：可读
 - “取值范围”：0~65535
 - “步长”：1
 - “单位”：s

图 2-144 添加属性

**步骤4 新增服务类型“Battery”。**

1. 在“模型定义”下单击“添加服务”，填写“服务ID”、“服务类型”和“服务描述”，单击“确定”。
 - “服务ID”：Battery
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：Battery

图 2-145 添加服务



2. 在““Battery”” 的下拉菜单下单击“添加属性”，填写相关信息后，单击“确定”。
 - “属性名称”：batteryLevel
 - “数据类型”：int（整型）
 - “访问权限”：可读
 - “取值范围”：0-100
 - “步长”：1
 - “单位”：%

图 2-146 添加属性



----结束

上传设备 CA 证书

- 步骤1 在左侧导航栏选择“设备 > 设备CA证书”，单击右上角的“上传证书”。
- 步骤2 在弹出的对话框中，单击“添加文件”，然后单击“确定”。

图 2-147 上传证书



说明

- 设备CA证书由设备厂商提供，调测时可[自行制作调测证书](#)，商用时建议更换为商用证书，否则会带来安全风险。
- CA证书具有一个过期日期，在该日期后，这些证书将无法用于验证服务器的证书；请在CA证书的过期日期前替换这些证书，以确保设备可以正常的连接到IoT平台。

----结束

制作设备 CA 调测证书

本文以Windows环境为例，介绍通过OpenSSL工具制作调测证书的方法，生成的证书为PEM编码格式的证书。

- 在浏览器中访问[这里](#)，下载并进行安装OpenSSL工具。
- 以管理员身份运行cmd命令行窗口。
- 执行cd c:\openssl\bin（请替换为openssl实际安装路径），进入openssl命令视图。
- 执行以下命令生成密钥对。
openssl genrsa -out rootCA.key 2048
- 执行以下命令，使用密钥对中的私有密钥生成CA证书。
openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem
系统提示您输入如下信息，所有参数可以自定义。
 - Country Name (2 letter code) [AU]: 国家，如CN。
 - State or Province Name (full name) []: 省份，如GD。
 - Locality Name (for example, city) []: 城市，如SZ。
 - Organization Name (for example, company) []: 组织，如Huawei。
 - Organizational Unit Name (for example, section) []: 组织单位，如IoT。
 - Common Name (e.g. server FQDN or YOUR name) []: 名称，如zhangsan。
 - Email Address []: 邮箱地址，如1234567@163.com。

在openssl安装目录的bin文件夹下，获取生成的CA证书（rootCA.pem）。

上传验证证书

如果上传的是调测证书，上传后证书状态显示为“未验证”，您需要上传验证证书，来证明您拥有该CA证书。

图 2-148 设备 CA 证书



验证证书是由设备CA证书对应的私钥创建的，请参考如下操作制作验证证书。

步骤1 获取验证证书的验证码

图 2-149 选择上传验证证书



图 2-150 获取验证码

**步骤2** 执行如下命令为私有密钥验证证书生成密钥对。

```
openssl genrsa -out verificationCert.key 2048
```

步骤3 执行如下命令为私有密钥验证证书创建CSR (Certificate Signing Request)。

```
openssl req -new -key verificationCert.key -out verificationCert.csr
```

系统提示您输入如下信息，**Common Name**填写为验证证书的验证码，其他参数自定义。

- Country Name (2 letter code) [AU]: 国家，如CN。
- State or Province Name (full name) []: 省份，如GD。
- Locality Name (for example, city) []: 城市，如SZ。
- Organization Name (for example, company) []: 组织，如Huawei。
- Organizational Unit Name (for example, section) []: 组织单位，如IoT。
- Common Name (e.g. server FQDN or YOUR name) []: 验证证书的验证码，请参考**步骤1**获取。
- Email Address []: 邮箱地址，如1234567@163.com。
- Password[]: 密码，如1234321。
- Optional Company Name[]: 公司名称，如Huawei。

步骤4 执行以下命令使用CSR创建私有密钥验证证书。

```
openssl x509 -req -in verificationCert.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out verificationCert.pem -days 500 -sha256
```

在openssl安装目录的bin文件夹下，获取生成的验证证书（ verificationCert.pem ）。

步骤5 选择对应证书，单击 然后单击“上传验证证书”。

图 2-151 上传验证证书

| 验证状态 | 证书ID | 证书所有者 | 证书创建日期 | 证书失效日期 | 操作 |
|------|----------------|----------------|----------------------------|----------------------------|----|
| 未验证 | XXXXXXXXXXXXXX | XXXXXXXXXXXXXX | 2021/04/14 11:13:53 GMT... | 2024/02/02 11:08:30 GMT... | 删除 |

证书ID: XXXXXXXX
证书所有者: XXXXXXXX
证书生效日期: 2021/04/14 11:08:30 GMT+08:00
验证证书 ②

步骤6 在弹出的对话框中，单击“添加文件”，然后单击“确定”。

图 2-152 完成验证



上传验证证书后，证书状态变为“已验证”，表明您拥有该CA证书。

----结束

预置 X.509 证书

在注册X.509设备之前，您需要在设备侧预置CA机构签发的X.509证书。

说明

X.509证书由CA机构签发，若没有CA机构签发的商用证书，您可以自己[制作设备CA调测证书](#)。

制作X.509调测证书

- 以管理员身份运行cmd命令行窗口，执行cd c:\openssl\bin（请替换为openssl实际安装路径），进入openssl命令视图。
- 执行如下命令生成密钥对。
openssl genrsa -out deviceCert.key 2048
- 执行如下命令为设备证书创建CSR（Certificate Signing Request）。
openssl req -new -key deviceCert.key -out deviceCert.csr
系统提示您输入如下信息，所有参数可以自定义。
 - Country Name (2 letter code) [AU]: 国家，如CN。

- State or Province Name (full name) []: 省份, 如GD。
 - Locality Name (for example, city) []: 城市, 如SZ。
 - Organization Name (for example, company) []: 组织, 如Huawei。
 - Organizational Unit Name (for example, section) []: 组织单位, 如IoT。
 - Common Name (e.g. server FQDN or YOUR name) []: 名称, 如zhangsan。
 - Email Address []: 邮箱地址, 如1234567@163.com。
 - Password[]: 密码, 如1234321。
 - Optional Company Name[]: 公司名称, 如Huawei。
4. 执行以下命令使用CSR创建设备证书。
- ```
openssl x509 -req -in deviceCert.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out deviceCert.pem -days 500 -sha256
```
- 在openssl安装目录的bin文件夹下, 获取生成的设备证书 (deviceCert.pem)。

## 注册 X.509 证书认证的设备

**步骤1** 访问[设备接入服务](#), 单击[管理控制台](#)进入设备接入控制台。

**步骤2** 在左侧导航栏选择“设备 > 所有设备”, 单击右上角“注册设备”, 按照如下表格填写参数后, 单击“确定”。

图 2-153 单设备注册



| 参数名称   | 说明           |
|--------|--------------|
| 所属资源空间 | 选择设备所属的资源空间。 |

| 参数名称   | 说明                                                                                                                                                                                                                                                                                                                                                                              |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 所属产品   | 选择设备所属的产品。<br>如没有, 请先 <a href="#">创建产品</a> 。                                                                                                                                                                                                                                                                                                                                    |
| 设备标识码  | 即node_id, 填写为设备的IMEI、MAC地址或Serial No; 若没有真实设备, 填写自定义字符串, 由英文字母和数字组成                                                                                                                                                                                                                                                                                                             |
| 设备名称   | 即device_name, 可自定义。                                                                                                                                                                                                                                                                                                                                                             |
| 设备认证类型 | X.509证书: 设备使用X.509证书验证身份。                                                                                                                                                                                                                                                                                                                                                       |
| 指纹     | 当“设备认证类型”选择“X.509证书”时填写, 导入 <a href="#">设备侧预置的设备证书</a> 对应的指纹, 在OpenSSL执行 <a href="#">openssl x509 -fingerprint -sha256 -in deviceCert.pem</a> 命令可查询。注: 填写时需要删除冒号。<br><pre>[root@k8s-iot-wl2-2 ~]# openssl x509 -fingerprint -sha256 -in deviceCert.pem SHA256 Fingerprint=F7:91:90:45:BB:88:37:E6:A7:E7:70:4A:90:75:F3:87:DA:27:5B:7C:49:3E:FF:59:7A:6F:4F:08:4D:F8:54:E8</pre> |

----结束

## 连接鉴权

参考[连接鉴权](#)接口文档, 使用MQTT.fx工具激活在物联网平台上注册的设备。

**步骤1** 下载[MQTT.fx](#) (默认是64位操作系统, 如果是32位操作系统, 单击此处下载[MQTT.fx](#)), 安装MQTT.fx工具。

### 说明

- 安装最新版MQTT.fx工具, 可点击此处[下载](#)。
- MQTT.fx 1.7.0及旧版本对带有\$的主题 (Topic) 处理存在问题, 请使用最新版本进行测试。

**步骤2** 访问[这里](#), 填写[注册设备](#)后生成的设备ID (DeviceId) 和密钥 (DeviceSecret), 生成连接信息 (ClientId、Username、Password)。

### 说明

DeviceSecret在此场景下可随意填写, 如12345678。

图 2-154 获取 ClientId

The screenshot shows the "HuaweiCloud IoTDA Mqtt ClientId Generator!" tool. It has input fields for DeviceId (5f6179df85) and DeviceSecret (12345678). A large blue "Generate" button is centered. Below it, the generated ClientId (5f6179df8506), Username (5f6179df85), and Password (08799d45642) are displayed in separate input fields.

| 参数       | 必选/<br>可选 | 类型          | 参数描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------|-----------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ClientId | 必选        | String(256) | <p>一机一密的设备clientId由4个部分组成：设备ID、设备身份标识类型、密码签名类型、时间戳，通过下划线“_”分隔。</p> <ul style="list-style-type: none"><li>● <b>设备ID：</b>指设备在平台成功注册后生成的唯一设备标识，通常由设备的产品ID和设备的NodeID通过分隔符“_”拼装而来。</li><li>● <b>设备身份标识类型：</b>固定值为0，表示设备ID。</li><li>● <b>密码签名类型：</b>长度1字节，当前支持2种类型：<ul style="list-style-type: none"><li>- “0”代表HMACSHA256不校验时间戳。</li><li>- “1”代表HMACSHA256校验时间戳。</li></ul></li><li>● <b>时间戳：</b>为设备连接平台时的UTC时间，格式为YYYYMMDDHH，如UTC时间2018/7/24 17:56:20则应表示为2018072417。</li></ul> |
| Username | 必选        | String(256) | 设备ID。                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

设备通过MQTT协议的connect消息进行鉴权，对于构造clientId的各个部分信息都必须包括进去，平台收到connect消息时，会判断设备的鉴权类型和密码摘要算法。

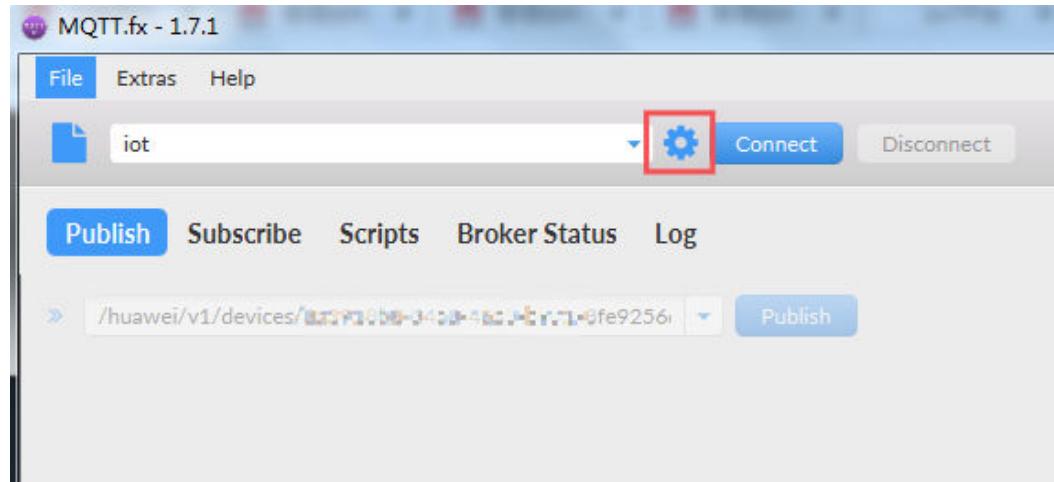
使用生成工具生成的clientId格式，默认不校验时间戳：设备ID\_0\_0\_时间戳。

- 当采用“HMACSHA256”校验时间戳方式时，会先校验消息时间戳与平台时间是否一致，再判断密码是否正确。
- 当采用“HMACSHA256”不校验时间戳方式时，鉴权消息也必须带时间戳，但不检验时间是否准确，仅判断密码是否正确。

connect消息鉴权失败时，平台会返回错误，并自动断开MQTT链路。

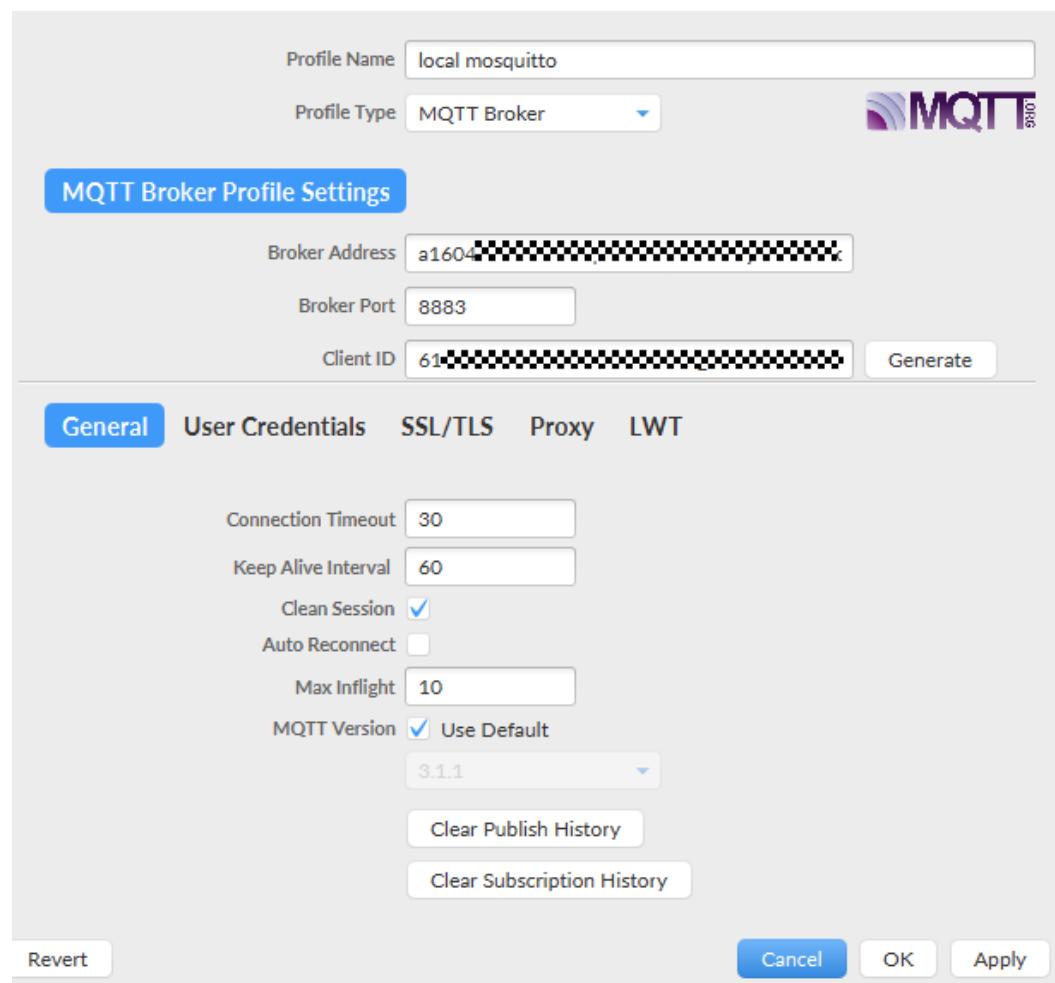
**步骤3** 打开MQTT.fx软件，单击“设置”图标。

图 2-155 设置



步骤4 填写“Connection Profile”相关信息。

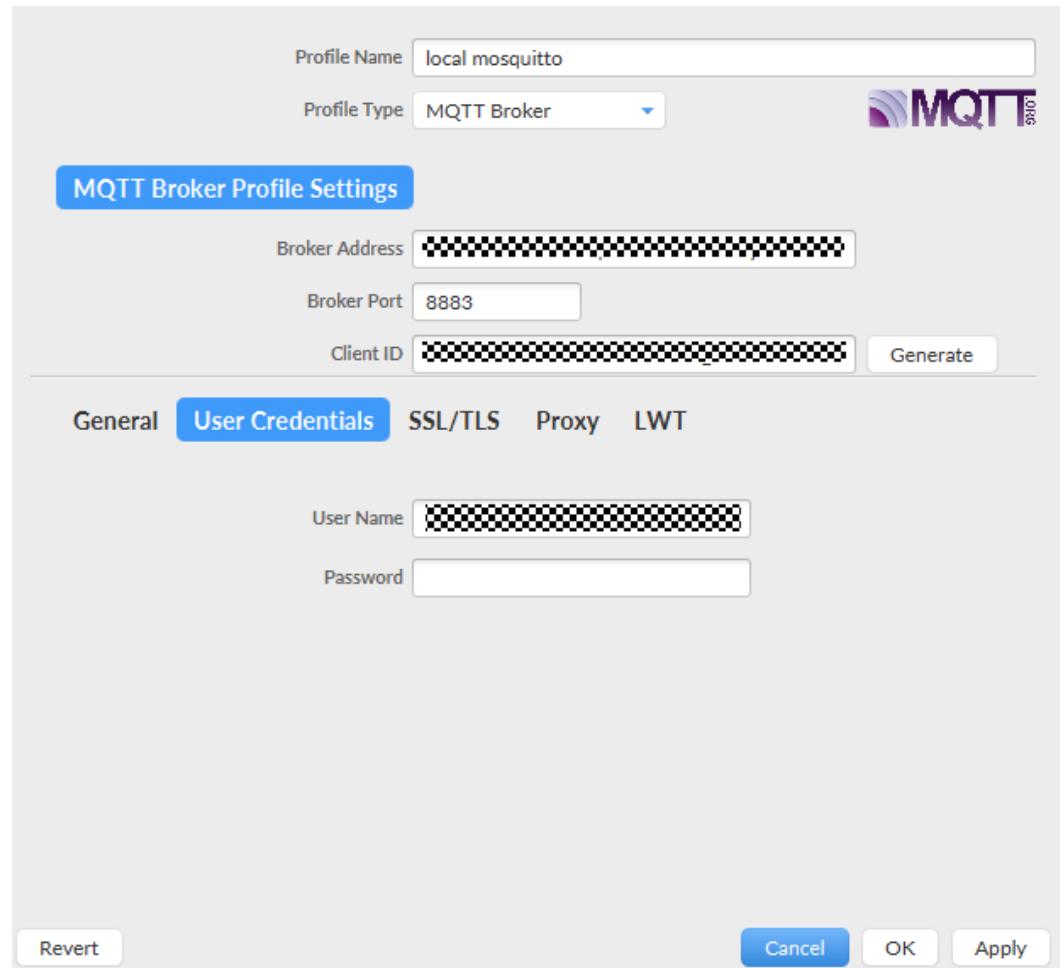
图 2-156 “General”可以使用工具默认信息



| 参数名称           | 说明                                                                     |
|----------------|------------------------------------------------------------------------|
| Broker Address | 填写从设备接入服务控制台获取的 <b>设备对接地址</b> ，此地址为域名信息。不能通过域名接入的设备，填写IP地址，详细请参考2中获取的。 |
| Broker Port    | 为8883。                                                                 |
| Client ID      | 设备clientID，请参考2中获取。                                                    |

步骤5 单击“User Credentials”填写“User Name”。

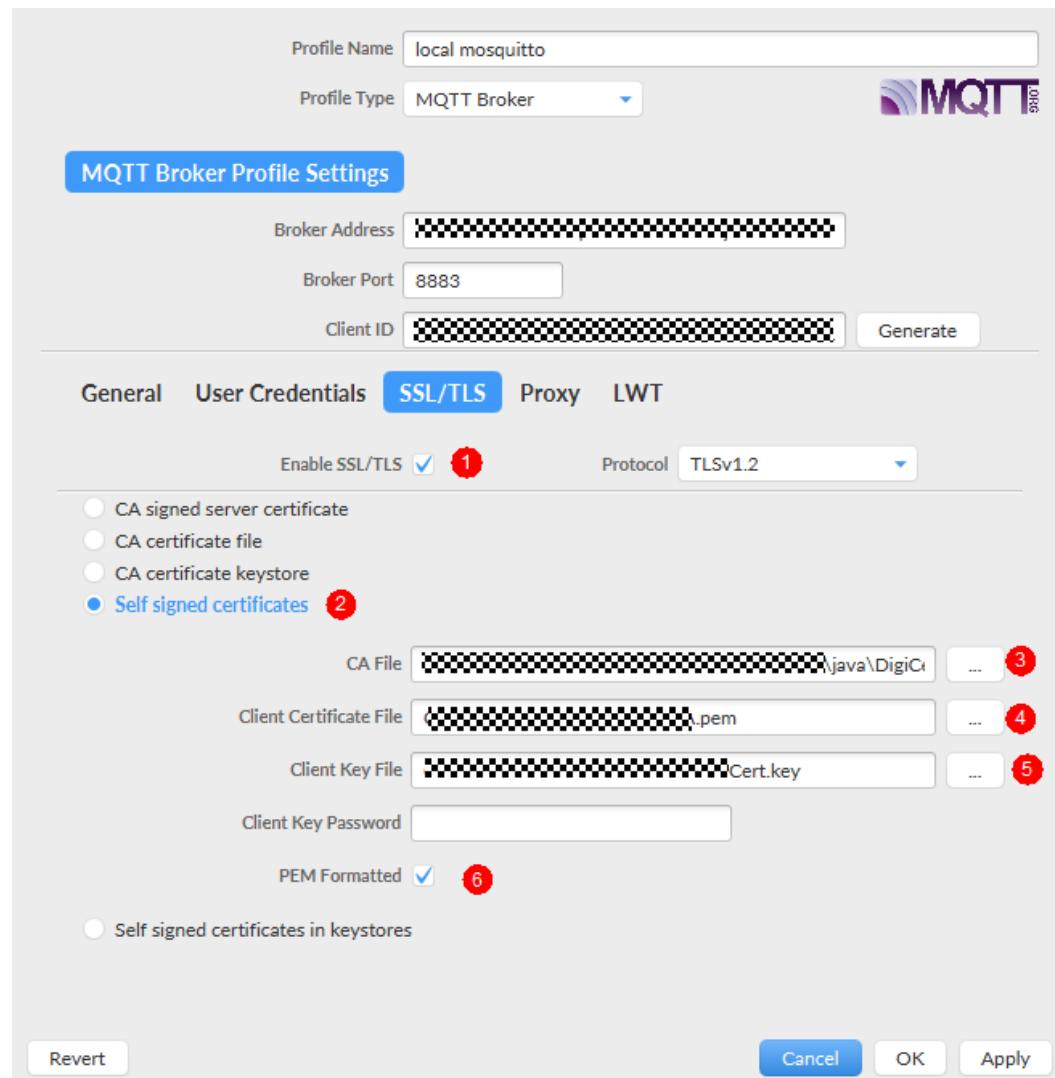
图 2-157 填写设备 ID



| 参数名称      | 说明                 |
|-----------|--------------------|
| User Name | 即设备ID，请参考2中获取。     |
| Password  | 使用X.509证书认证时不需要填写。 |

**步骤6** 单击“SSL/TLS”配置鉴权参数，然后单击““Apply””。选择开启“SSL/TLS”，勾选“Self signed certificates”，配置相关证书内容。

图 2-158 填写“SSL/TLS”相关参数



### 说明

CA File为对应的CA证书。下载并获取证书（加载pem格式的证书），获取证书请根据需要在[资源获取](#)里下载。

Client Certificate File为设备的设备证书（deviceCert.pem）。

Client Key File为设备的私钥（deviceCert.key）。

**步骤7** 单击“Connect”，设备鉴权成功后，在物联网平台可以看到设备处于在线状态。

图 2-159 设备状态

| 设备列表 |            |        |                |      |      |          | 想洞悉设备数据深层次的信息？开始 分析历史数据 |      |    |
|------|------------|--------|----------------|------|------|----------|-------------------------|------|----|
| 状态   | 设备名称       | 设备标识码  | 所属资源空间         | 所属产品 | 节点类型 | 操作       | 所有状态                    | 设备名称 | 查询 |
| 在线   | xxxxxxxxxx | xxxxxx | xxxxxxxxxxxxxx | xxxx | 直连设备 | 查看、删除、冻结 |                         |      |    |

----结束

## 上报数据

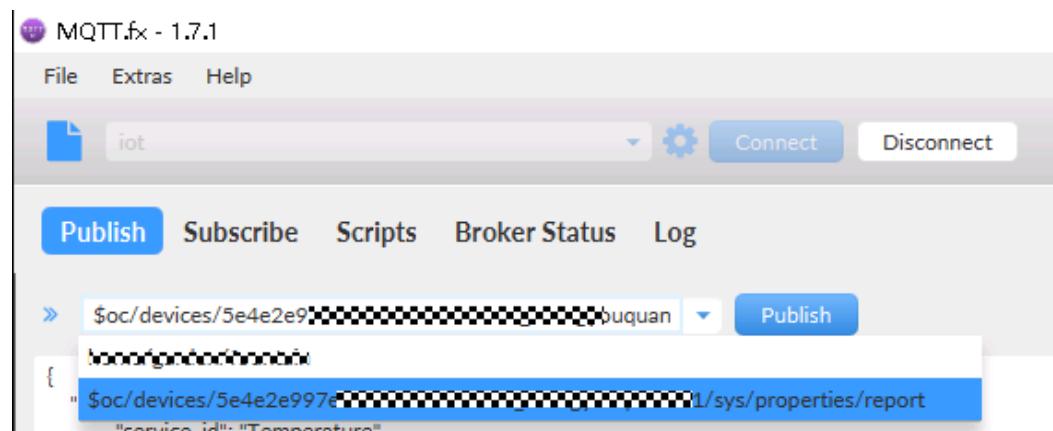
参考[设备属性上报](#)接口文档，使用MQTT.fx工具向物联网平台上报数据。

设备若通过MQTT通道上报数据，需要发给指定的topic，上报消息的topic格式为：

“\$oc/devices/{device\_id}/sys/properties/report”，其中“device\_id”的值，为设备注册成功后返回的“device\_id”值。

- 步骤1** 填写接口地址，此处以“\$oc/devices/{device\_id}/sys/properties/report”为例，如“\$oc/devices/5e4e2e92ac-164aefaa8fouquan1/sys/properties/report”。

图 2-160 填写接口地址



- 步骤2** 填写上报的数据。

### 请求参数

| 字段名      | 必选/可选 | 类型                    | 参数描述                                 |
|----------|-------|-----------------------|--------------------------------------|
| services | 必选    | List<ServiceProperty> | 设备服务数据列表（具体结构参考下表ServiceProperty定义表） |

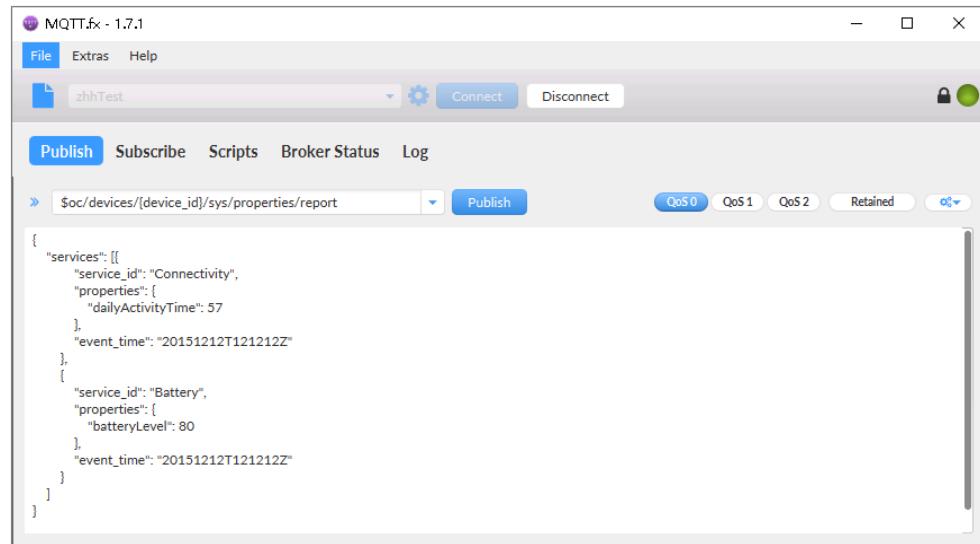
ServiceProperty定义表：

| 字段名        | 必选/可选 | 类型     | 参数描述                                                                                        |
|------------|-------|--------|---------------------------------------------------------------------------------------------|
| service_id | 必选    | String | 设备服务的ID。                                                                                    |
| properties | 必选    | Object | 设备服务的属性列表，具体字段在设备关联的产品模型中定义。                                                                |
| event_time | 可选    | String | 设备采集数据UTC时间（格式：yyyyMMddTHHmmssZ），如：20161219T114920Z。<br>设备上报数据不带该参数或参数格式错误时，则数据上报时间以平台时间为准。 |

## 请求示例

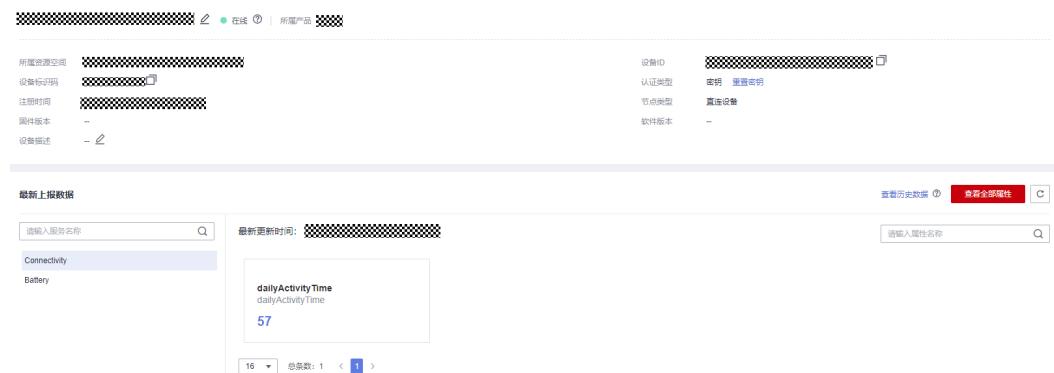
```
{
 "services": [{
 "service_id": "Connectivity",
 "properties": {
 "dailyActivityTime": 57
 },
 "event_time": "20151212T121212Z"
 },
 {
 "service_id": "Battery",
 "properties": {
 "batteryLevel": 80
 },
 "event_time": "20151212T121212Z"
 }
]}
```

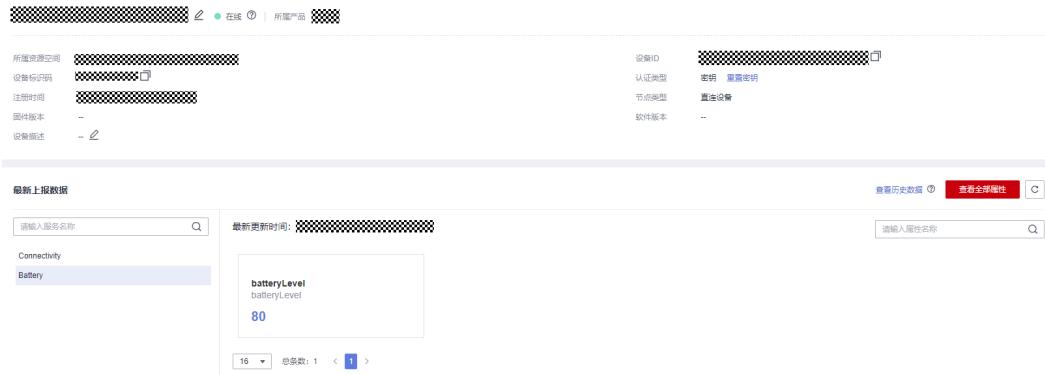
图 2-161 请求示例



步骤3 单击“Publish”，可以在物联网平台上查看设备是否成功上报数据。

## 图 2-162 上报数据





----结束

## 2.15 基于 OpenHarmony3.0 对接 IoTDA

本文主要介绍基于OpenHarmony 3.0来接入IoTDA，以BearPi-HM\_Nano开发板为例，使用huaweiCloud\_iot\_link SDK对接华为云物联网平台的简单流程。基于OpenHarmony 3.1及更高版本参考[链接](#)。

### 前提条件

- 已注册华为云官方帐号。未注册可单击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击[实名认证](#)完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“管理控制台”后开通该服务。

### 硬件环境

BearPi-HM\_Nano开发板、E53\_IA1扩展模块、Type-C数据线、PC。

### 软件环境

参考[链接](#)单击[这里](#)，搭建环境可以选择基于IDE方式或者基于命令行方式，其中BearPi-HM\_Nano主控芯片为Hi3861，因此搭建环境时需要安装Hi3861对应环境。

图 2-163 软件环境创建步骤

## 快速入门

- 快速入门概述
- 1. 基于IDE入门
  - 搭建开发环境
    - 搭建Windows环境
    - 搭建Ubuntu环境
    - 配置远程访问环境
  - 创建工程并获取源码
  - 轻量系统（基于Hi3861开发板）
    - 编写“Hello World”程序
    - 编译
    - 烧录
    - 运行
- 2. 基于命令行入门
  - 搭建开发环境
    - 准备开发环境
    - 安装库和工具集
    - 获取源码
    - 安装编译工具
  - 轻量系统（基于Hi3861开发板）
    - 安装Hi3861开发板特有环境
    - 编写“Hello World”程序
    - 编译
    - 烧录

### 说明

在“安装Hi3861开发板特有环境”的安装gcc\_riscv32步骤时，强烈建议直接下载gcc\_riscv32镜像，否则可能出现某些插件下载或安装失败，具体操作如下图2所示。

### 须知

开源代码量较大，下载时间可能较长，需预留好充足时间。

图 2-164 直接下载 gcc\_riscv32 镜像

**安装gcc\_riscv32 (WLAN模组类编译工具链)**

须知:

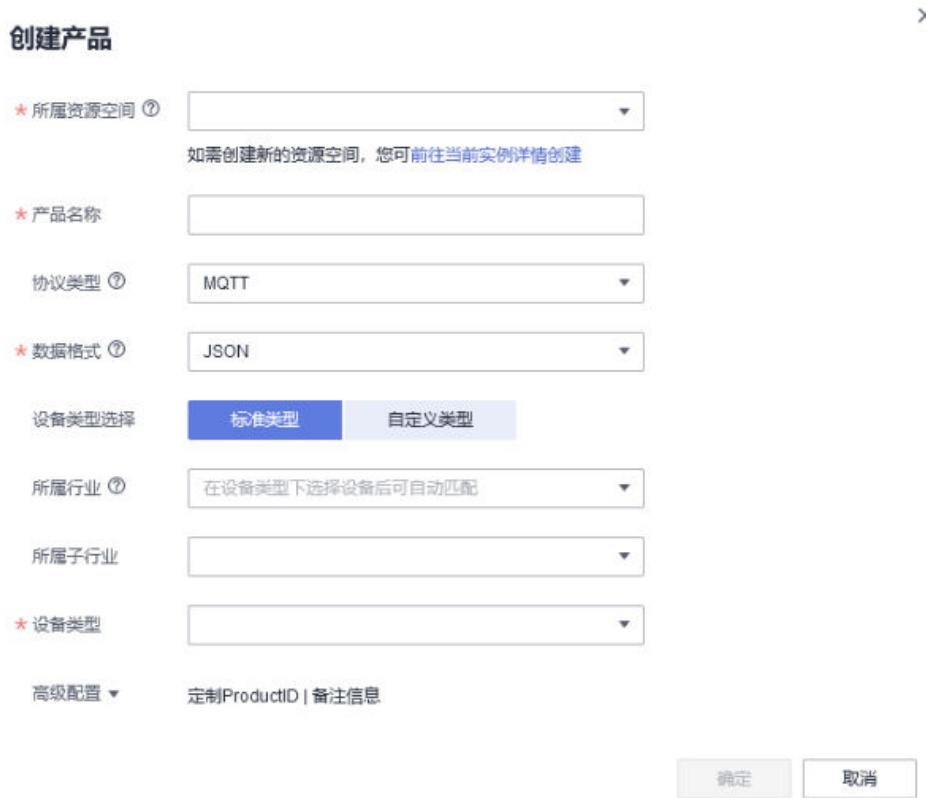
- Hi3861开发板平台仅支持使用libgcc运行时库的静态链接，不建议开发者使用libgcc运行时库的动态链接，以免产品需遵从GPLV3许可证。
- 通过下述步骤2-14，编译好了gcc\_riscv32镜像，提供给开发者[直接下载](#)使用。直接下载gcc\_riscv32镜像的开发者可省略下述2-14步。

## 创建产品

**步骤1** 登录[管理控制台](#)，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。**步骤2** 根据页面提示填写参数，然后单击“确认”，完成产品的创建。

| 基本信息   |                                                                                |
|--------|--------------------------------------------------------------------------------|
| 所属资源空间 | 平台自动将新创建的产品归属在默认资源空间下。如需归属在其他资源空间下，下拉选择所属的资源空间。如无对应的资源空间，请先创建资源空间。             |
| 产品名称   | 为产品命名。产品名称在相同资源空间有唯一性。长度不超过64，只允许中文、字母、数字、以及_?#().,&%@!-等字符的组合。                |
| 协议类型   | 选择MQTT。                                                                        |
| 数据格式   | 选择JSON。                                                                        |
| 所属行业   | 请根据实际情况选择。                                                                     |
| 设备类型   | 请根据实际情况选择。                                                                     |
| 高级配置   |                                                                                |
| 产品ID   | 定制ProductID，用于唯一标识一个产品。如果携带此参数，平台将产品ID设置为该参数值；如果不携带此参数，产品ID在物联网平台创建产品后由平台分配获得。 |

图 2-165 创建产品



----结束

## 开发产品模型

**步骤1** 找到新增的产品，单击“查看”进入产品界面。

**步骤2** 在产品详情“模型定义”页面，直接上传模型文件[BearPi\\_Agriculture.zip](#)。

### 说明

- 在产品列表中，单击对应的产品，进入产品详情页。您可以查看产品ID、产品名称、设备类型、数据格式、厂商名称、所属资源空间、协议类型等产品基本信息。
- 产品创建成功后，您可以单击“删除”删除不再使用的产品。删除产品后，该产品下的产品模型、编解码插件等资源将被清空，请谨慎操作。

----结束

## 注册设备

**步骤1** 访问[设备接入服务](#)，单击[管理控制台](#)进入设备接入控制台。

**步骤2** 在左侧导航栏选择“设备 > 所有设备”，单击右上角“注册设备”，按照如下表格填写参数后，单击“确定”。

图 2-166 注册单个设备

单设备注册

所属资源空间

所属产品

MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

设备标识码

设备ID

设备名称

设备描述

0/2,048

设备认证类型

密钥 X.509证书

密钥

确认密钥

确定 取消

| 参数名称   | 说明                                                                                              |
|--------|-------------------------------------------------------------------------------------------------|
| 所属资源空间 | 选择设备所属的资源空间。                                                                                    |
| 所属产品   | 选择设备所属的产品。如没有, 请先 <a href="#">创建产品</a> 。                                                        |
| 设备标识码  | 即node_id, 填写为设备的IMEI、MAC地址或Serial No; 若没有真实设备, 填写自定义字符串, 长度为4至64个字符, 只允许字母、数字、下划线(_)、连接符(-)的组合。 |
| 设备名称   | 即device_name, 可自定义。                                                                             |
| 设备认证类型 | 选择“密钥”。                                                                                         |
| 密钥     | 设备密钥, 可自定义, 不填写物联网平台会自动生成。                                                                      |

图 2-167 获取设备密钥



### 说明

1. 设备注册成功后，请妥善保管好设备ID和密钥，用于设备接入平台认证。
2. 若密钥丢失，只能通过[重置密钥](#)生成新的密钥，无法找回注册设备时生成的密钥。
3. 用户可在[设备列表](#)删除不再使用的设备。删除设备不支持撤回，请谨慎操作。

----结束

## Huaweicloud\_iotlink SDK 使用

**步骤1** 下载源码：[oh3.0\\_hwcloud\\_iotlink-master.zip](#)。

**步骤2** 将上述源码直接拷贝到OpenHarmony源码根目录下的src->third\_party目录中，此处需要注意的是OpenHarmony和Huaweicloud\_iotlink SDK共同具备的三方库直接使用OpenHarmony的库文件，如CJSON和mbedtls。

**步骤3** 在OpenHarmony 3.0源码device\bearpi\bearpi\_hm\_nano\app\BUILD.gn中添加如下代码，根据#是否注释来决定具体编译demo的选择。

```
//third_party/hwcloud_iotlink/demos/mqtt_tiny_v5_agriculture_demo:mqtt_tiny_v5_agriculture_demo,
```

图 2-168 demo 的选择



```
device > bearpin > bearpin_hm_nano > app > BUILD.gn
12 # limitations under the License.
13
14 import("//build/lite/config/component/lite_component.gni")
15
16 lite_component("app") {
17 features = []
18
19 1 "# //third_party/hwcloud_iotlink/demos/test_iotlink:test_iotlink",
20
21 2 "//third_party/hwcloud_iotlink/demos/mqtt_tiny_v5_agriculture_demo:mqtt_tiny_v5_agriculture_demo",
22
23 # A1_kernel_thread:thread_example",
24 # A2_kernel_timer:timer_example",
25 # ...
26 }
```

说明

图6中2为使用MQTT对接华为云的demo，看一下该demo的BUILD.gn文件，具体如图7所示，其中A区域包含开发板硬件和wifi相关的库文件，B区域包括MQTT对接华为云必须的一些库文件，如 cJSON，mqtt相关、osal相关以及配置库文件，C区域表示编译该文件时需要编译 hwcloud\_iotlink库，编译是会按照指定路径找到该文件所必要的库以及C文件进行编译。

图 2-169 代码编译文件

```
git workspace
> grpc
> gstreamer
> harfbuzz
< hwcloud_iotlink
> at
> cJSON
> crc
< demos
> hello_world...
< mqtt_tiny_v5_agriculture_demo
> include
> src
BUILD.gn
C mqtt_v5_o...
C oc_agricultu...
C oc_mqtt_de...
< test_iotlink
BUILD.gn
C test_main.c
C app_demo...
M demos.mk
< driver
BUILD.gn
C dev_fs_test.c
C dev_fs.c
C dev_test.c
C driver.c
C driver.h
M driver.mk
> fs
< hwcloud_iotli...
BUILD.gn
C iot_config.h
C iot_link_con...
C link_main.c
C link_main.h
> inc
> link_log

... BUILD.gn ... \app M ... BUILD.gn ... \mqtt_tiny_v5_agriculture_demo M x C wifi_connect.c U C iot_config.h

third_party > hwcloud_iotlink > demos > mqtt_tiny_v5_agriculture_demo > BUILD.gn
4 # You may obtain a copy of the License at
5 #
6 # http://www.apache.org/licenses/LICENSE-2.0
7 #
8 # Unless required by applicable law or agreed to in writing, softw
9 # distributed under the License is distributed on an "AS IS" BASIS
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
See the License for the specific language governing permissions
limitations under the License.

static_library("mqtt_tiny_v5_agriculture_demo") {
 sources = [
 "mqtt_v5_oc_agriculture_sample.c",
 "src/E53_I1.c",
 "src/wifi_connect.c",
]

 include_dirs = [
 "//base/iot_hardware/peripheral/interfaces/kits",
 "//device/bearpi/bearpi_hm_nano/iot_hardware_hals/include",
 "//foundation/communication/wifi_lite/interfaces/wifiservice",
 // third-party dependencies
 "//third_party/cJSON",
 "//third_party/hwcloud_iotlink/oc/oc_mqtt/oc_mqtt_all",
 "//third_party/hwcloud_iotlink/oc/oc_mqtt/oc_mqtt_profile_v5",
 "//third_party/hwcloud_iotlink/inc",
 "//third_party/hwcloud_iotlink/os/osal",
 "//third_party/hwcloud_iotlink/os/ohos",
 "include",
 "//third_party/hwcloud_iotlink/hwcloud_iotlink_config",
]
}

deps = [
 // hwcloud_iotlink dependency
 "//third_party/hwcloud_iotlink:hwcloud_iotlink",
]
```

图 2-170 demo 的主函数文件

```
> crc
> cJSON
> demos
> hello_world...
> mqtt_tiny_v...
> include
> src
> BUILD.gn
C mqtt_v5_... 1, M
> oc_agricultu...
> oc_mqtt.de...
> test_iotlink
> BUILD.gn
C test_main.c
C app_demo_...
M demos.mk
> driver
> BUILD.gn
C dev_fs_test.c
> ...

22 #include <dtls_al.h>
23 #include <mqtt_al.h>
24 #include <oc_mqtt_al.h>
25 #include <oc_mqtt_profile.h>
26 #include "E53_IA1.h"
27 #include "wifi_connect.h"

355 static void IoTMainTaskEntry(void)
356 {
357 link_main_task_entry();
358 (void) osal_task_create("CloudMainTaskEntry", CloudMainTaskEntr
359 (void) osal_task_create("SensorTaskEntry", SensorTaskEntry, NULL
360
361 }
362
363 APP_FEATURE_INIT(IotMainTaskEntry);
```

## 说明

图8中A区域表示MQTT对接华为云所必须的一些库文件以及dtls的库文件，入口函数IoTMainTaskEntry()中务必先调用link\_main\_task\_entry()函数，来确保osal的安装以及其他配置的初始化。

## 步骤4 设置参数。

图 2-171 参数修改

```
> at
> cJSON
> crc
> demos
> hello_world_demo
> mqtt_tiny_v5_agriculture_demo
> include
> src
C E53_IA1.c
C wifi.connect.c
> BUILD.gn
C mqtt_v5_oc_agriculture_sample.c 9+
> oc_agriculture_template
> oc_mqtt_demo
> test_iotlink
> app_demo_main.c

30 #define CONFIG_WIFI_SSID "your wifi hotspot name" // 修改为自己的WiFi 热点账号
31
32 #define CONFIG_WIFI_PWD "your wifi hotspot password" // 修改为自己的WiFi 热点密码
33
34 #define CONFIG_APP_SERVERIP "Access Address" // 替换为自己的对接地址
35
36 #define CONFIG_APP_SERVERPORT "1883"
37
38 #define CONFIG_APP_DEVICEID "your device id" // 替换为注册设备后生成的deviceid
39
40 #define CONFIG_APP_DEVICEPWD "your device password" // 替换为注册设备后生成的密钥
41
42 #define CONFIG_APP_LIFETIME 60 // seconds
43
44 #define CONFIG_QUEUE_TIMEOUT (5 * 1000)
45
46 #define MSGQUEUE_COUNT 16
47 #define MSGQUEUE_SIZE 10
```

## 注意

- (1) 如果需要对接上云需要根据自己的设备修改配网信息，wifi热点账号以及密码以及自己在云端注册的设备ID和设备秘钥，注意此设备仅支持2.4G频段的wifi。
- (2) 对接地址修改为[管理控制台](#)界面“总览”的“接入信息”设备接入MQTT协议地址。
- (3) 编译和烧录，分为IDE方式和命令行方式，这里不一一介绍。可以参考[图2-163](#)对应链接。
- (4) 烧录后，按下开发板RESET复位键，设备将启动。
- (5) 上述代码基于OpenHarmony3.0版本，若为其他版本按需修正相应BUILD.gn文件中引入的OpenHarmony源码路径。

## ----结束

## 设备接入平台

代码烧录到设备后重启设备（首次使用需重启两次，第一次烧录时可能需要配置一下内部信息），第二次设备重启之后可以正常对接华为云。

登录平台可以看见数据上报的具体信息，也可以进行命令下发等操作控制设备，具体如下图10-12。

图 2-172 数据上报

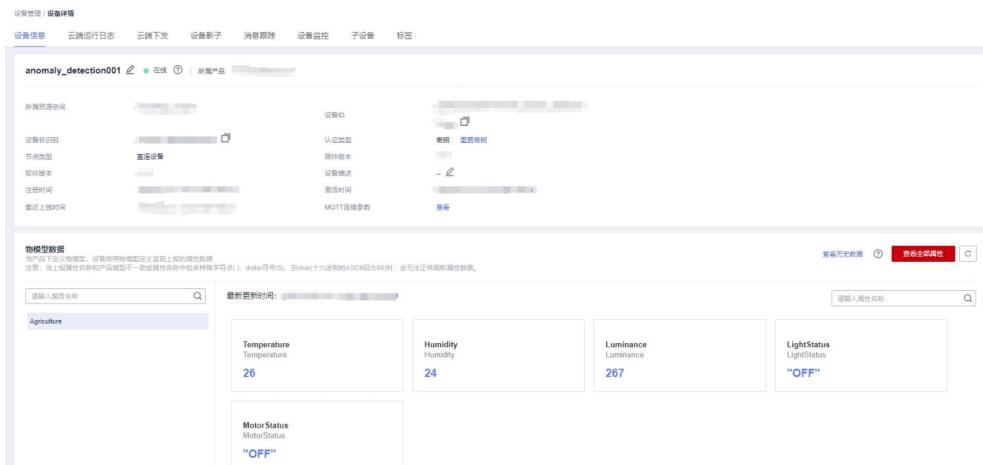


图 2-173 命令下发



图 2-174 日志信息

```
clients <1> :
mac_idx mac addr state lease tries rto
0 201131001a50 [REDACTED] 10 0 1 4
[DEBUG] [3690] [hub_step] hub_step:enter
[DEBUG] [3690] [dmp_connect] oc_mqtt_connect:server: [REDACTED] port:
[DEBUG] [3700] [dmp_connect] oc_mqtt_connect:client_id:
[DEBUG] [3710] [dmp_connect] oc_mqtt_connect:user:
password:
[DEBUG] [3720] [dtls_ssl_new] setting up the SSL structure
[DEBUG] [3730] [dtls_ssl_new] set SSL structure succeed
[DEBUG] [3730] [dtls_shakehand] connecting to server
[DEBUG] [3930] [dtls_shakehand] performing the SSL/TLS handshake
[DEBUG] [4370] [dtls_shakehand] handshake succeed
[DEBUG] [4470] [dmp_connect] oc_mqtt_connect:retcode:0 :SUCCESS
[DEBUG] [4470] [dmp_subscribe] oc_mqtt_subscribe:start
[DEBUG] [4480] [dmp_subscribe] oc_mqtt_default_subscribe:topic:$oc/devices/6399432cf5ceb911701elf63_anomaly_detection001bzm/sys/commands/#
[DEBUG] [5590] [dmp_subscribe] oc_mqtt_default_subscribe:retcode:0:success
[DEBUG] [5590] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701elf63_anomaly_detection001bzm/sys/messages/down
[DEBUG] [6600] [dmp_subscribe] oc_mqtt_subscribe:retcode:0:success
[DEBUG] [7601] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701elf63_anomaly_detection001bzm/sys/properties/set/#
[DEBUG] [7610] [dmp_subscribe] oc_mqtt_subscribe:retcode:0:success
[DEBUG] [7810] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701elf63_anomaly_detection001bzm/sys/properties/get/#
[DEBUG] [8940] [dmp_subscribe] oc_mqtt_subscribe:retcode:0:success
[DEBUG] [8940] [dmp_subscribe] topic:$oc/devices/6399432cf5ceb911701elf63_anomaly_detection001bzm/sys/shadow/get/response/#
[DEBUG] [10070] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701elf63_anomaly_detection001bzm/sys/events/down
[DEBUG] [10070] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701elf63_anomaly_detection001bzm/sys/events/up
[DEBUG] [11090] [dmp_subscribe] oc_mqtt_subscribe:retcode:0:success
[DEBUG] [11090] [hub_step] hub_step:ok exit
ret = 0
oc_mqtt_profile_connect succed!
```

## 2.16 设备级消息上下行保序方案

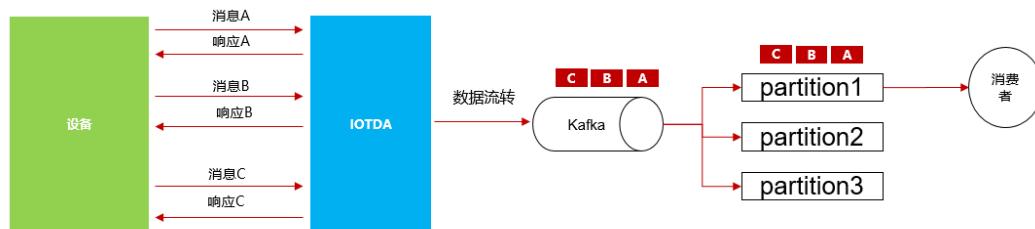
华为云IoTDA服务在满足高可靠、高并发的同时可按需开启设备维度的消息上下行保序功能。本文以MQTT协议为例介绍基于IoTDA的上、下行消息的保序方案。

### 场景说明

典型如车联网场景中：高速行驶中的车辆会实时上报自身的GPS位置数据（通常1~5秒/条数据），IoT平台接收该数据并转发给云端应用，应用侧根据上报的位置数据绘制实时轨迹地图，车辆端上报的轨迹点依次为：A点——>B点——>C点，在乱序情况下，应用侧接收到的轨迹点可能会为：B点——>A点——>C点，从而导致绘制的轨迹地图与真实轨迹不符。

### 上行消息保序

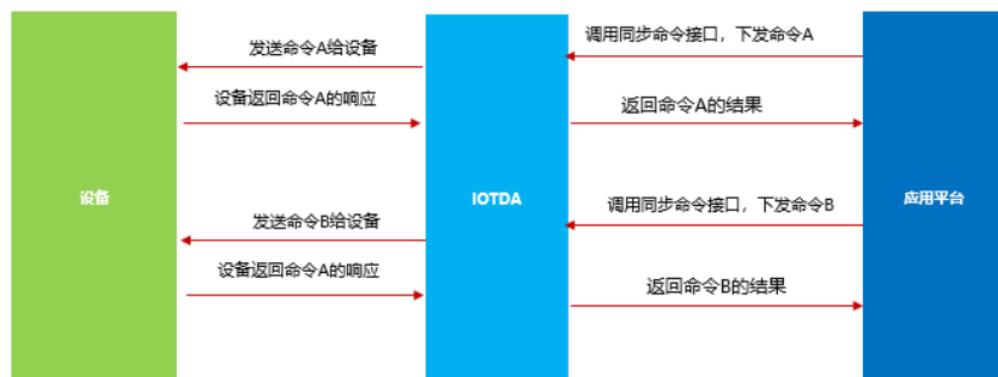
在需要上行消息保序的场景，可以参考如下图的方案进行设计：



- 设备-平台：设备侧上报消息时，Qos值设置为1，上报消息A后，等收到消息A的Qos响应后，再依次上报消息B和消息C，从而保证设备与平台侧的消息上行保序。
- 平台-应用：IoTDA配置数据流转规则为流转到Kafka，当开启IoTDA的消息保序功能后，IoTDA会将同一个设备的消息推送到Kafka的同一个partition，从而实现平台与应用端的转发保序。
- 应用服务器消费Kafka数据时，同一个设备的消息会分发到同一个消费者，消费者可以按顺序消费设备依次上传上来的数据。

### 下行消息保序

在需要下行消息保序的场景，可以参考如下图的方案进行设计：



- 应用平台调用**同步命令**接口下发命令A。
- IoTDA收到同步命令A的请求后，会将命令转换为MQTT消息下发给设备，下发请求后平台会阻塞等待，等待设备返回命令A的执行结果。
- 设备收到命令A的请求，执行完对应操作后，返回命令A的响应。
- IoTDA收到命令A的响应后返回命令结果给应用侧。
- 应用平台收到命令A的响应后再下发命令B。

### 须知

通过物联网平台实现消息保序会对整体性能略有影响，IoTDA默认关闭该功能，如有需求请[提交工单](#)咨询，当前仅企业版支持开启该功能。

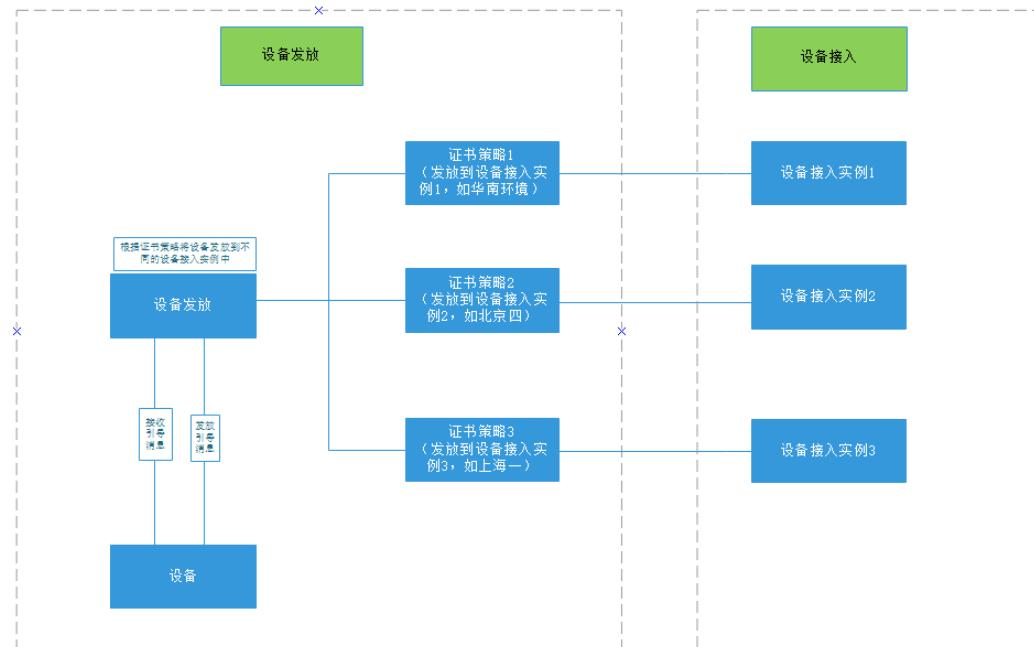
## 2.17 通过设备发放实现批量设备自动注册

### 场景说明

当前物联网平台基于安全考虑，需要将设备的基本信息（例如设备ID、鉴权信息）注册到平台后，设备才能连接上平台。当用户的设备数不断增多时，如何安全且快速的注册设备尤为关键，需要采用安全的批量注册的场景。本文基于设备发放注册组功能，使用证书策略，带您快速体验设备批量自注册场景，并成功发放到指定的设备接入区域（本文以“华南-广州-友好用户环境”为例）。

### 整体流程

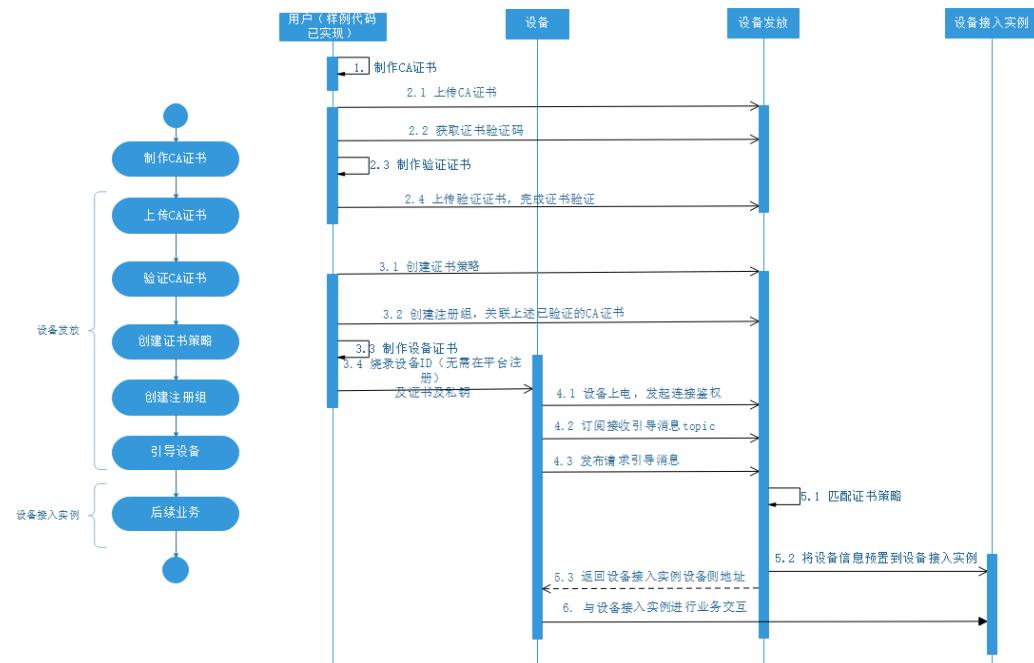
图 2-175 整体流程图



如上图所示，用户可以在设备发放里设置证书策略，不同的证书策略可以将设备发放到不同的设备接入服务实例中，当设备携带了对应证书并成功接入到设备发放时，可以主动发起发放引导流程，从而将该设备按照证书的策略发放到对应的设备接入实例。

## 详细业务流程

图 2-176 业务流程图



如上图所示，用户需要按照上图的设备发放步骤操作，然后才能通过证书策略发放到对应设备接入实例，当前样例代码已将全流程自动化，用户可以参考后续的步骤来完成批量自注册流程。

## 前提条件

1. 已开通IoTDA的“华南-广州-华南友好环境”。
2. 进入“IoTDA华南-广州-华南友好环境”控制台，选择左侧导航栏“设备发放”服务，开通设备发放服务。

图 2-177 华南-广州-华南友好环境控制台

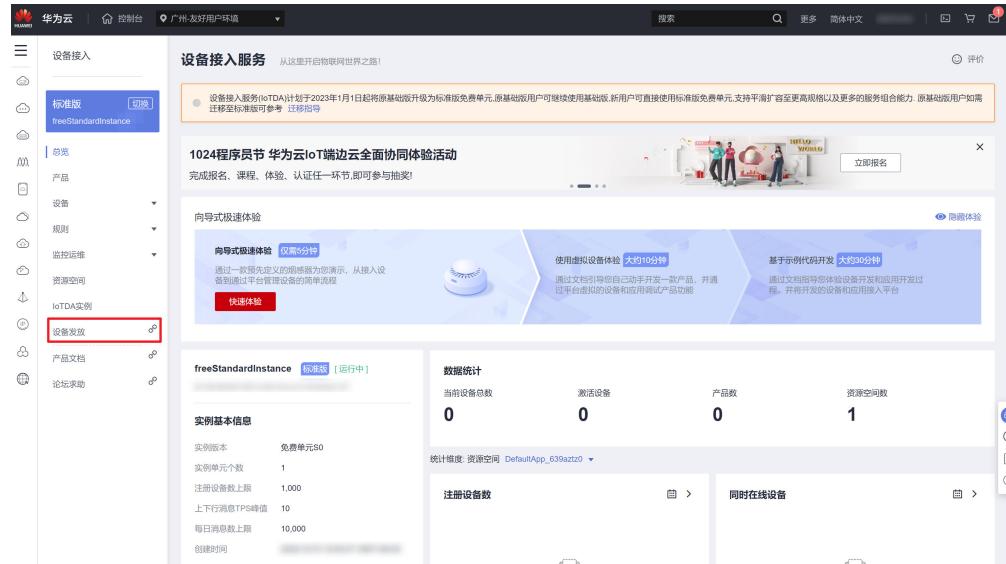
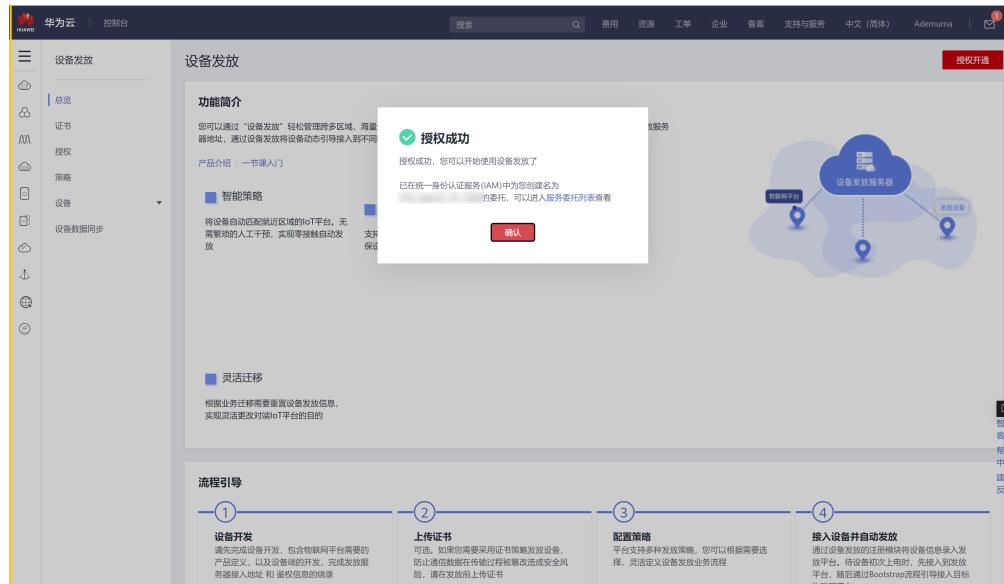


图 2-178 授权开通



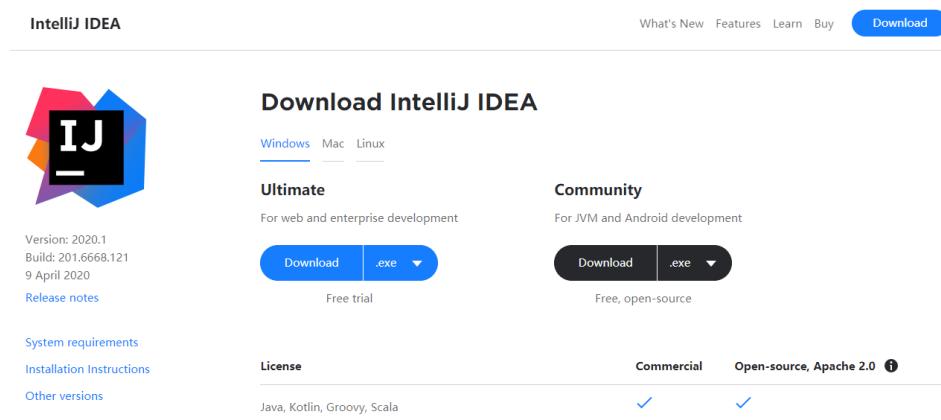
图 2-179 授权成功



### 3. 安装IntelliJ IDEA

- 访问[IntelliJ IDEA官网](#)，选择合适系统的版本下载。（本文以windows 64-bit系统IntelliJ IDEA 2019.2.3 Ultimate为例）。

图 2-180 IntelliJ IDEA 官网



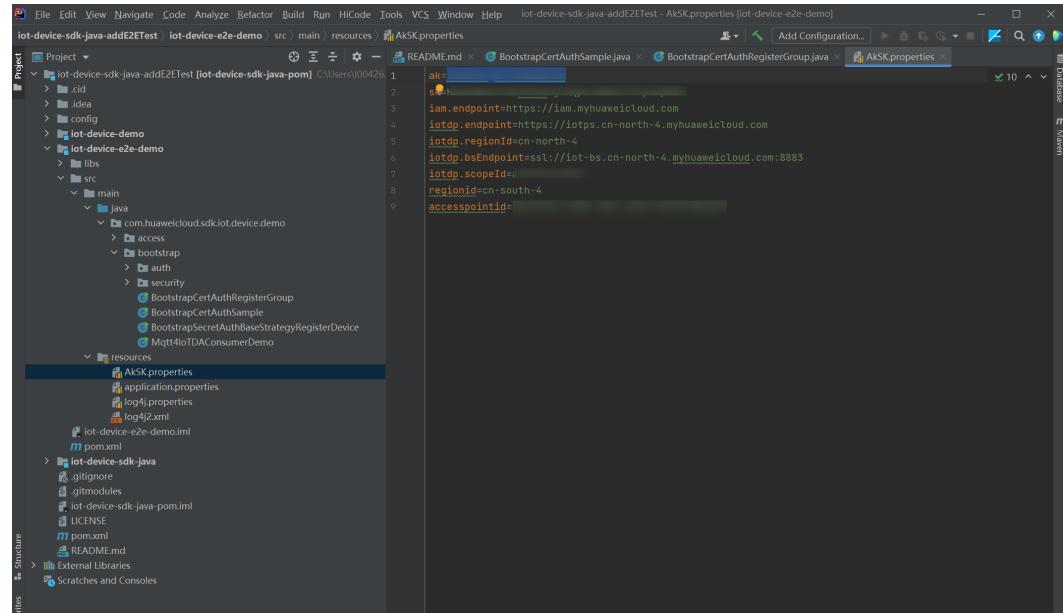
- 下载完成后，运行安装文件，根据界面提示安装。
- 安装Maven软件，通过 Maven 安装项目依赖是使用 Java SDK 的推荐方法，首先您需要[下载并安装 Maven](#)，安装完成后您只需在 Java 项目的 pom.xml 文件加入相应的依赖项即可。

## 导入批量自注册样例代码

**步骤1** 下载[JAVA批量自注册样例文档](#)。

**步骤2** 解压后选择样例代码主目录下的pom.xml文件，并用IDEA开发者工具打开后，可以看到工程目录如下：

图 2-181 工程目录

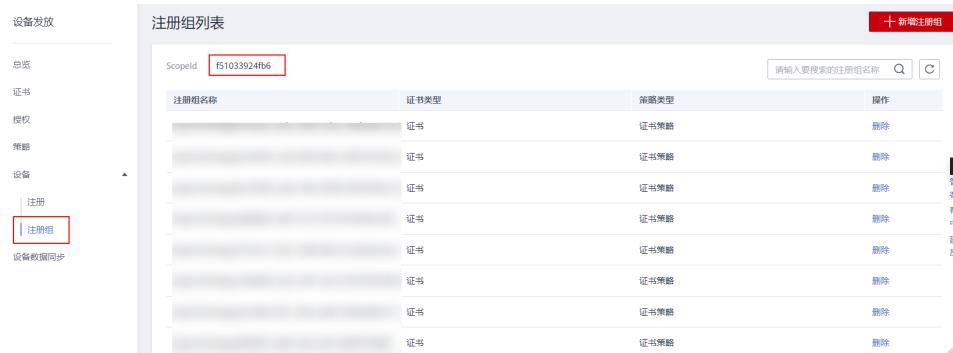


**步骤3** 修改AKSK.properties中的ak/sk参数，scopeId参数，accesspointid参数

## 说明

1. 用户可以在[此处](#)获取ak/sk。
2. scopeId可以在[此处](#)获取。

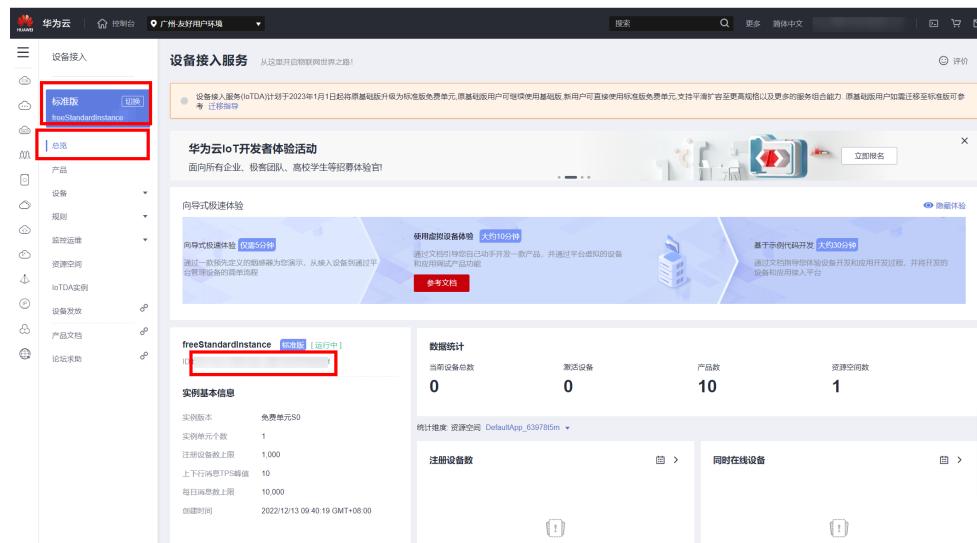
图 2-182 获取注册组 scopeId



| 注册组名称 | 证书类型 | 策略类型 | 操作 |
|-------|------|------|----|
| 证书    | 证书策略 | 删除   |    |

3. accesspointid可以在[IoTDA的“华南-广州-友好用户环境”](#)的总览界面获取到。

图 2-183 accesspointid 获取



设备接入服务

华为云IoT开发者体验活动

实例基本信息

|             |                               |
|-------------|-------------------------------|
| 实例版本        | 免费单元50                        |
| 实例帐号数       | 1                             |
| 注册设备数上限     | 1,000                         |
| 上下行消息TFPS峰值 | 10                            |
| 每日消息数上限     | 10,000                        |
| 创建时间        | 2022/12/13 09:40:19 GMT+08:00 |

数据统计

|        |    |
|--------|----|
| 当前设备总数 | 0  |
| 测试设备   | 0  |
| 产品数    | 10 |
| 逻辑空间数  | 1  |

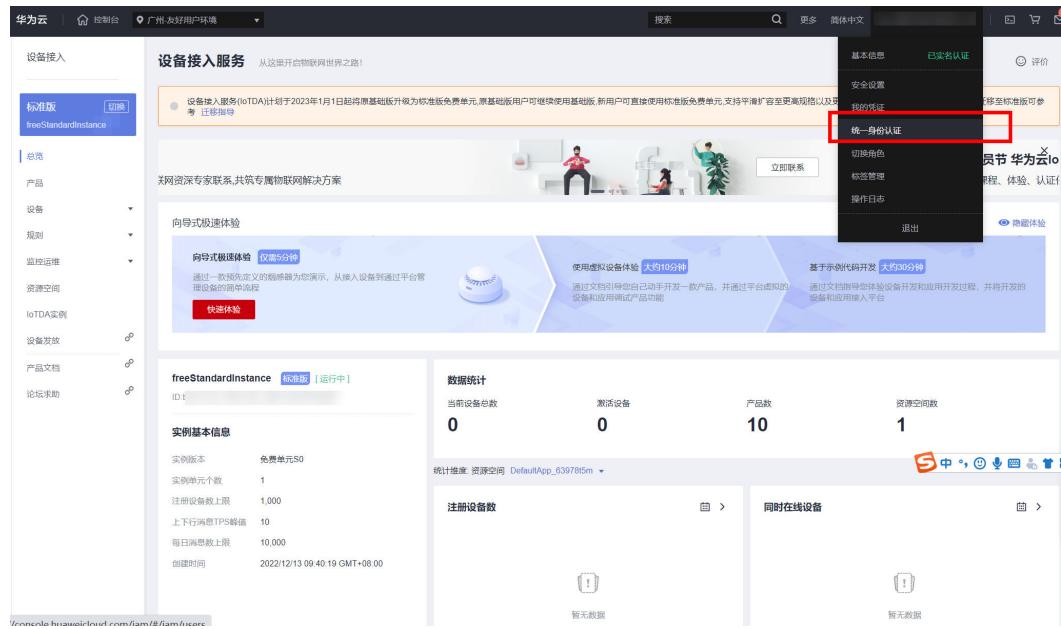
步骤4 上述步骤替换后，代码导入和修改完毕。

----结束

## 权限配置

步骤1 单击设备接入[控制台页面](#)右上角的账号，选择“统一身份验证”。

图 2-184 统一身份认证



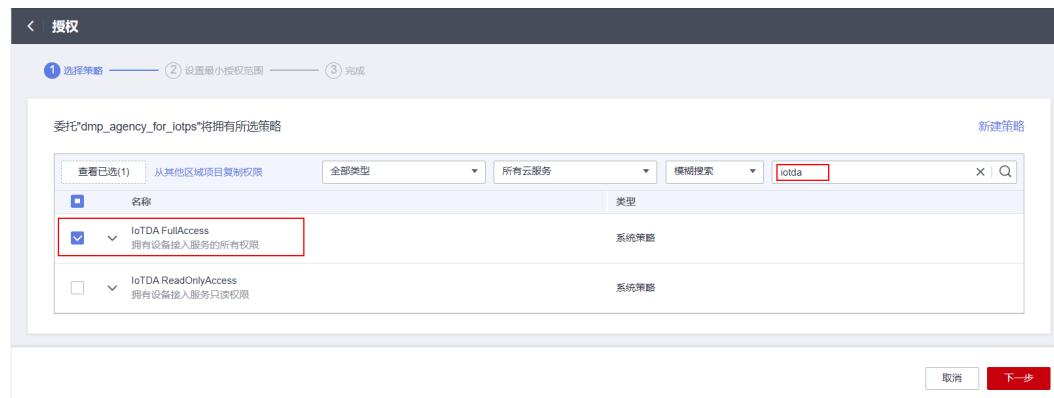
步骤2 进入统一身份认证服务页面后，单击左侧“委托”，然后在右上角的输入框中输入“iotps”，再单击一下iotps这个委托名称右侧的“授权”按钮。

图 2-185 创建委托授权



步骤3 进入授权页面后，在右上角的输入框中“iotda”，然后选择“IoTDA FullAccess”，再单击“下一步”。

图 2-186 选择策略



步骤4 跳转到如下页面后，直接单击右下角的“确定”即可。

图 2-187 设置最小授权范围



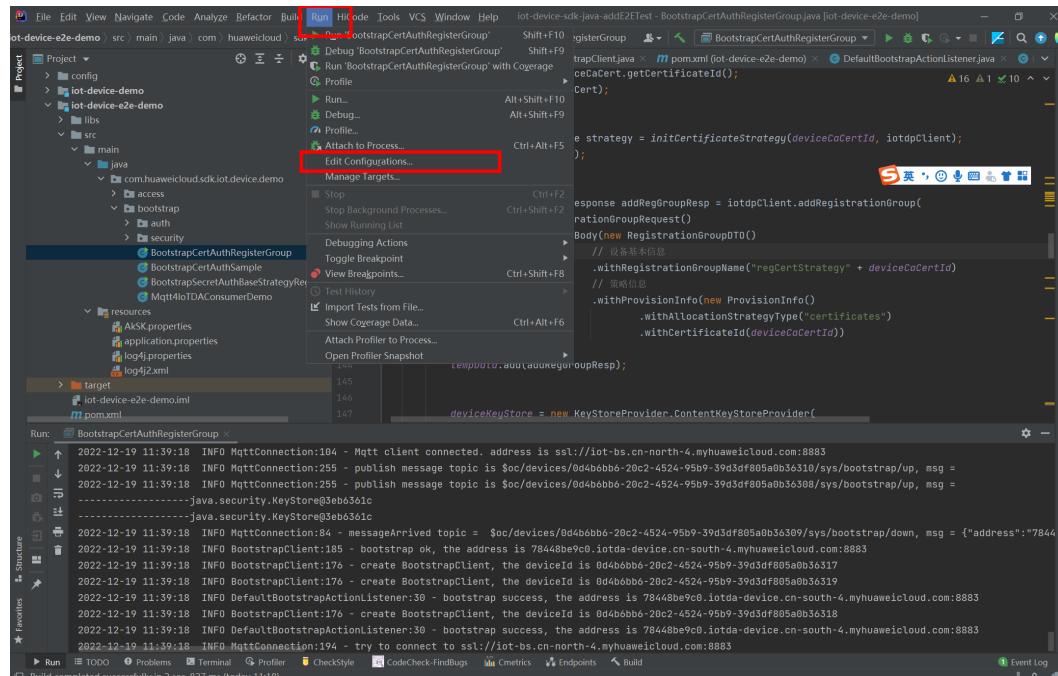
----结束

## 开始批量自注册引导

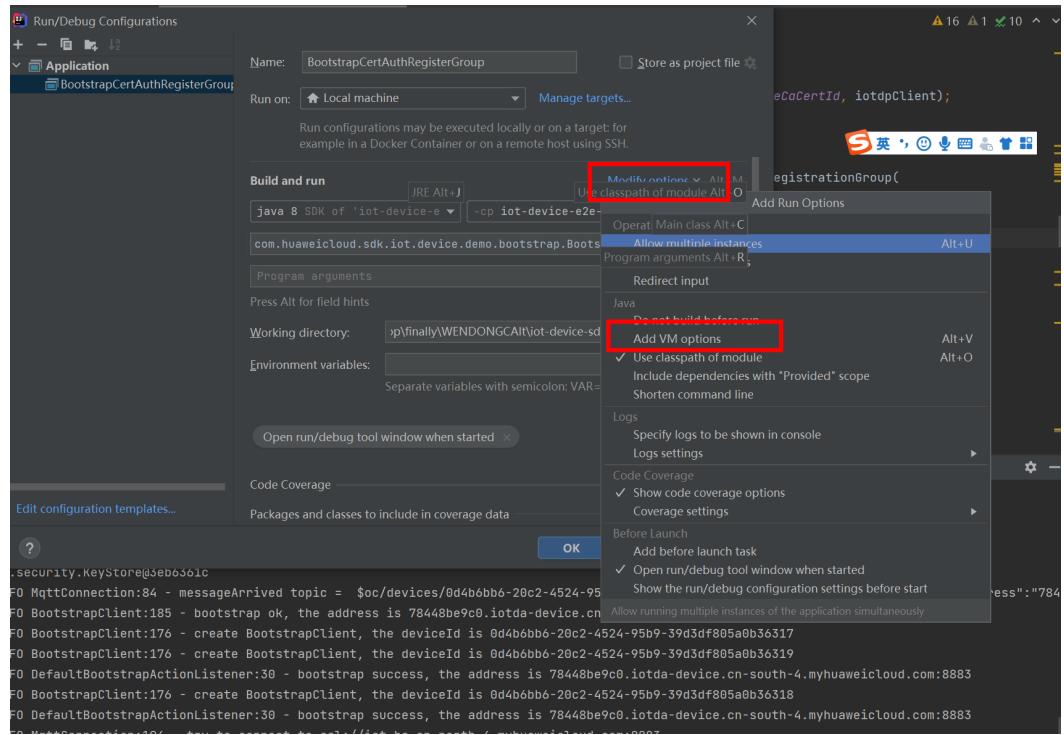
上述内容修改完后，可以运行iot-device-e2e-demo下的BootstrapCertAuthRegisterGroup.java类，右键单击运行后，SDK会自动开始批量自注册过程。当前默认自注册大约1000设备，大约10TPS。

由于mqtt链接比较耗资源，大家可以把jvm最大内存调大一些。

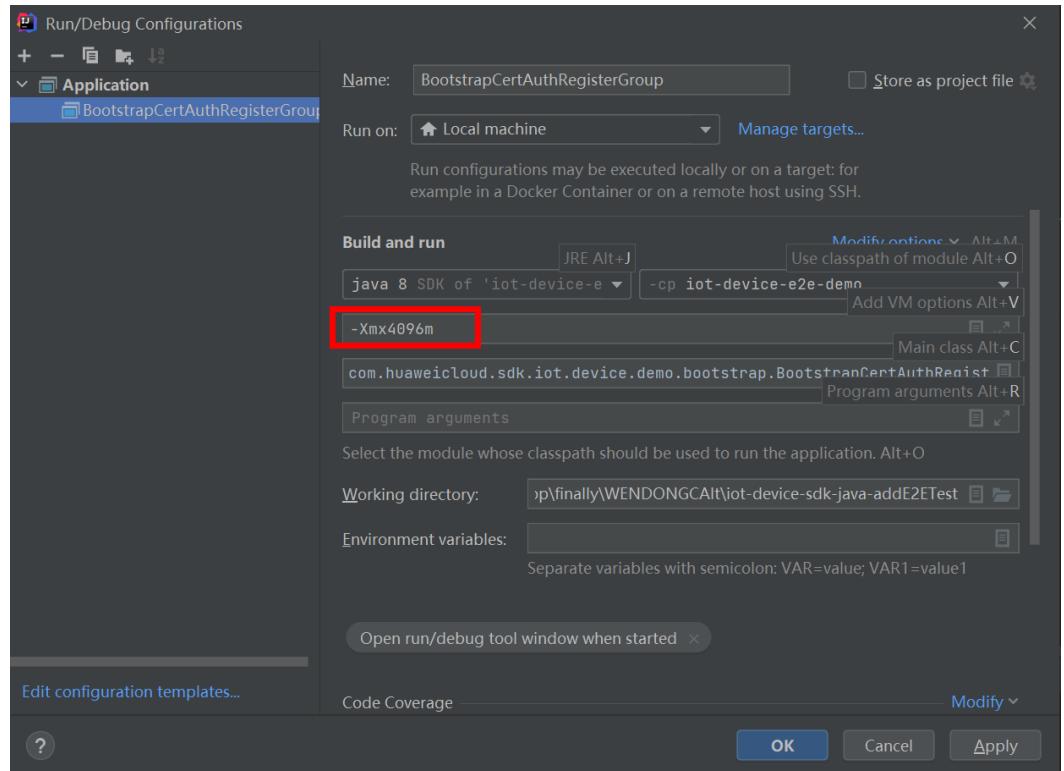
### 1、点击IDEA菜单栏上方的“RUN” - “Edit Configurations”



### 2、点击“Modify options” - “Add VM options”

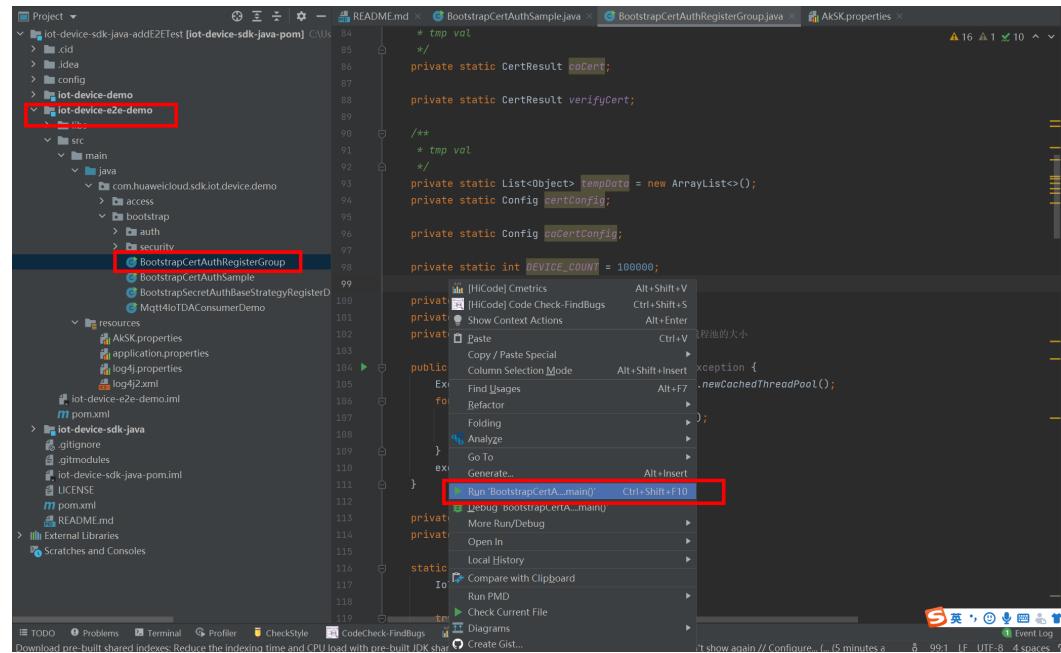


3、在下图的框中输入“-Xmx4096m”



4、运行程序，如下图所示：

图 2-188 批量自注册



## 结果验证

- 在自动开始批量自注册之后，用户可以在IDEA的控制台看到自注册的日志。

图 2-189 自注册日志

```
2022-12-13 21:01:00 INFO BootstrapClient:176 - create BootstrapClient, the deviceId is 7
2022-12-13 21:01:00 INFO MqttConnection:84 - messageArrived topic = $oc/devices/7
2022-12-13 21:01:00 INFO MqttConnection:183 - bootstrap ok, the address is 7/sys/bootstrap/down, msg = {"address":"iot-bs.cn-north-4.myhuaweicloud.com:8883"}
2022-12-13 21:01:00 INFO BootstrapClient:183 - bootstrap ok, the address is 7/sys/bootstrap/down, msg = {"address":"iot-bs.cn-north-4.myhuaweicloud.com:8883"}
2022-12-13 21:01:00 INFO MqttConnection:193 - try to connect to ssl://iot-bs.cn-north-4.myhuaweicloud.com:8883
2022-12-13 21:01:00 INFO MqttConnection:214 - connect success, the uri is ssl://iot-bs.cn-north-4.myhuaweicloud.com:8883
2022-12-13 21:01:00 INFO MqttConnection:104 - Mqtt client connected, address is 7/sys/bootstrap/down, msg = {"address":"iot-bs.cn-north-4.myhuaweicloud.com:8883"}
2022-12-13 21:01:00 INFO DefaultBootstrapActionListener:30 - bootstrap success, the address is 7/sys/bootstrap/up, msg =
2022-12-13 21:01:00 INFO MqttConnection:254 - publish message topic is $oc/devices/7
2022-12-13 21:01:00 INFO BootstrapClient:176 - create BootstrapClient, the deviceId is 8
2022-12-13 21:01:00 INFO MqttConnection:84 - messageArrived topic = $oc/devices/8
2022-12-13 21:01:00 INFO MqttConnection:183 - bootstrap ok, the address is 8/sys/bootstrap/down, msg = {"address":"iot-bs.cn-north-4.myhuaweicloud.com:8883"}
2022-12-13 21:01:00 INFO BootstrapClient:183 - bootstrap ok, the address is 8/sys/bootstrap/down, msg = {"address":"iot-bs.cn-north-4.myhuaweicloud.com:8883"}
2022-12-13 21:01:00 INFO MqttConnection:193 - try to connect to ssl://iot-bs.cn-north-4.myhuaweicloud.com:8883
2022-12-13 21:01:00 INFO MqttConnection:214 - connect success, the uri is ssl://iot-bs.cn-north-4.myhuaweicloud.com:8883
2022-12-13 21:01:00 INFO MqttConnection:104 - Mqtt client connected, address is 8/sys/bootstrap/down, msg = {"address":"iot-bs.cn-north-4.myhuaweicloud.com:8883"}
2022-12-13 21:01:00 INFO DefaultBootstrapActionListener:30 - bootstrap success, the address is 8/sys/bootstrap/up, msg =
2022-12-13 21:01:00 INFO DefaultBootstrapActionListener:30 - bootstrap success, the address is 8/sys/bootstrap/up, msg =
```

- 用户可以在设备接入控制台看到已发放的设备数目。

图 2-190 查看已发放的设备数目列表

The screenshot shows the 'Device Distribution' page. On the left is a navigation sidebar with categories like General, Product, Device, Rules, Storage Management, Monitoring, Resource Space, IoTDA Examples, Device Distribution, Product Documentation, API Search and Testing, and Forum Help. The 'Device Distribution' section is selected and highlighted with a red box. The main area displays a table of devices with columns: Status, Device Name, Device Identifier, Device ID, Resource Space, Product, Node Type, and Operations. A dropdown menu for 'All Status' is open above the table. The table shows several entries, all in the '未激活' (Not Activated) status. The bottom of the table shows a total of 138,007 devices and a page navigation bar.

3. 用户可以通过右上角的“所有状态”来过滤“在线”的设备，可以看到在线设备数一直在增长。

图 2-191 设备在线状态

This screenshot shows the same 'Device Distribution' page as the previous one, but with a different filter applied. The 'All Status' dropdown is now set to 'Online'. The table below shows the same device list, but only devices in the '在线' (Online) status are visible. The count at the bottom of the table is 8,321, which is significantly lower than the total in the previous screenshot, demonstrating the filtering effect.

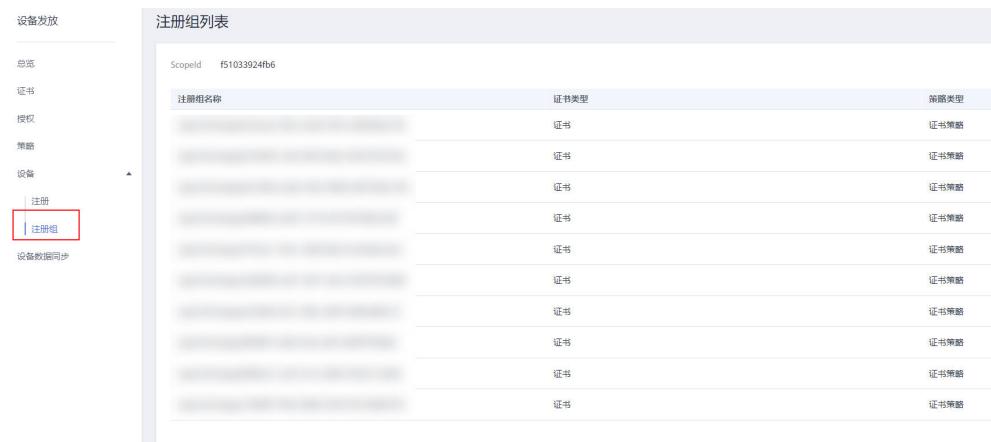
4. 用户可以回到**设备发放控制台**，看到SDK默认创建好的证书策略。

图 2-192 设备发放证书策略



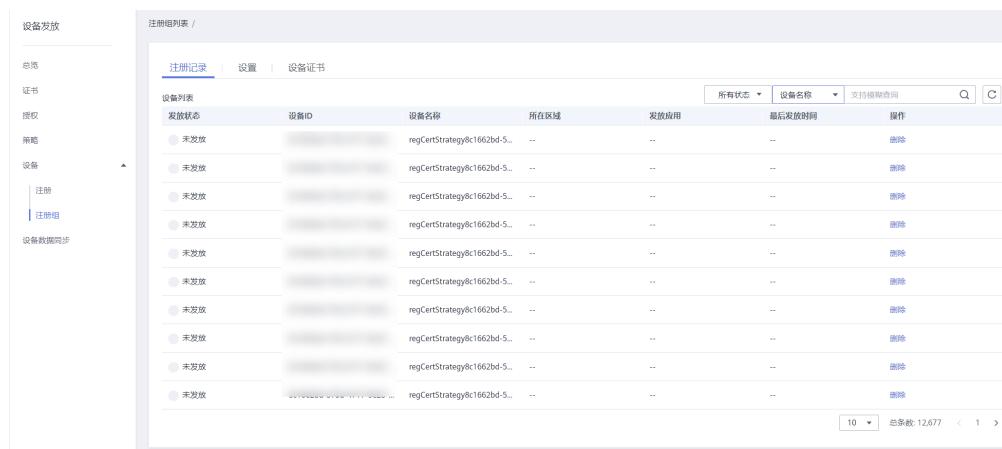
5. 用户可以看到SDK默认创建的注册组。

图 2-193 注册组列表



6. 单击上图中的“注册组”名称，可以看到注册组里设备发放状态。

图 2-194 设备发放状态



## 2.18 使用自定义 Topic 进行通信

### 场景说明

自定义Topic功能用于MQTT协议设备接入物联网平台的场景，支持[设备消息上报](#)、[平台消息下发](#)的Topic进行自定义。根据Topic不同，应用侧可实现不同的业务逻辑处理，也适用于应用侧和设备侧之间，设备无法按照产品模型中定义的属性上报或命令下发的场景。

本示例中，应用服务器收到设备上报Topic为室内温度的数据，判断是否开启或关闭室内空调。

### 前提条件

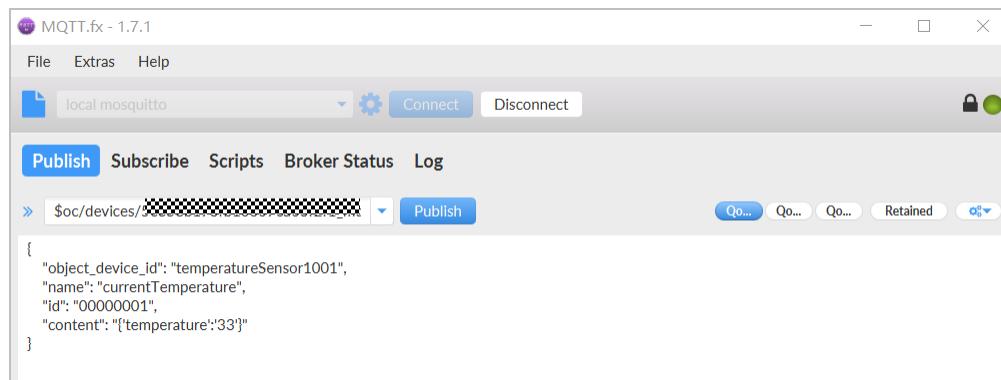
- 已经注册华为帐号并开通华为云，具体可参考[华为账号注册](#)。
- 已开通设备接入服务。
- 已在[管理控制台](#)创建MQTT协议类型的产品、开发产品模型、新增设备。创建产品、开发产品模型和新增设备的具体操作细节，请参考[创建产品](#)、[开发产品模型](#)、[注册单个设备](#)或[批量注册设备](#)。
- 已建立设备和平台之间的连接。具体操作细节，请参考[连接鉴权](#)。

### 新增自定义 Topic

具体操作细节请参考[新增自定义Topic](#)。

### MQTT 设备消息上报

- 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 选择“设备 > 所有设备”，找到对应的设备，单击“查看”进入设备详情页。
- 选择“消息跟踪”页签，单击“启动消息跟踪”跟踪时长根据您的需求自定义。
- 参考[MQTT设备快速接入](#)，以MQTT.fx模拟器为例，模拟设备上报自定义Topic消息。



#### 说明

使用IoT Device SDK或原生MQTT协议接入的设备，需要在设备程序中，设置设备上报的自定义Topic名称。

- 在消息跟踪页面，可以查看设备上报的自定义Topic消息。

图 2-195 查看消息跟踪

The screenshot shows the 'Device Management / Device Details' interface. The top navigation bar includes tabs for '设备信息', '云端运行日志', '云端下发', '设备影子', '消息跟踪' (highlighted with a red border), '设备监控', '子设备', '标签', and '群组'. On the left, a sidebar lists categories like '总览', '产品', '设备', '所有设备' (highlighted with a red border), '群组', '策略', '软固件升级', '设备CA证书', '设备代理', '自注册模板', '规则', '监控运维', '资源空间', and 'IoTDA实例'. The main content area has sections for '执行情况' (Execution Status) and '修改跟踪配置' (Modify Tracking Configuration). A search bar allows filtering by '业务详情' (Business Details). Below is a table with columns: '业务类型' (Business Type), '业务步骤' (Business Step), '业务详情' (Business Details), '记录时间' (Record Time), '消息状态' (Message Status), and '操作' (Operation). One row in the table is highlighted with a red box, showing a detailed log entry:

```
IoTDA has received the properties reported from the device data {"services": [{"service_id": "basicData", "properties": {"A1": 0, "REG20003": 267, "REG20004": 439, "REG20130": 322, "REG20015": 657}, "event_time": "201512121212122"}]}, app_id: 25f6c3f757747a8ab26706d6699229, device_id: test111, topic: $oc/devices/65af02e5b6dc2cd690a7138_12345678/sys/properties/report, request_id: fdc19f02-0989-4b47-b624-8c774d0af727.
```

6. 应用侧通过数据转发的方式，获取MQTT设备上报的自定义Topic消息。更多数据转发方式，请参考[数据转发](#)。您也可以参考最佳实践案例[数据转发到OBS长期存储](#)。

## MQTT 设备消息下发

本示例通过使用postman，下发开启室内空调指令。

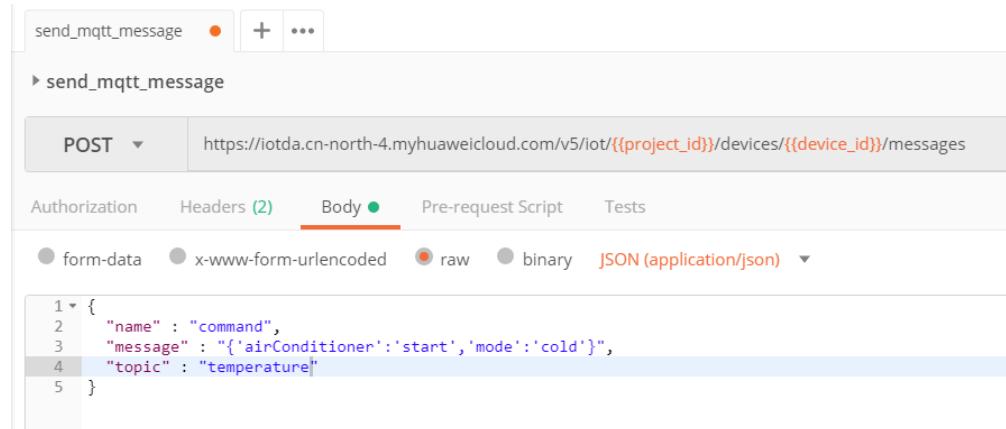
1. 使用MQTT.fx模拟器，订阅自定义Topic。

The screenshot shows the MQTT.fx 1.7.1 application window. The top menu bar includes 'File', 'Extras', and 'Help'. The connection status shows 'local mosquito' connected. The main interface has tabs for 'Publish', 'Subscribe' (highlighted with a red border), 'Scripts', 'Broker Status', and 'Log'. In the 'Subscribe' tab, the topic '\$oc/devices/[REDACTED]' is entered in the input field, and the 'Subscribe' button is highlighted with a red border. Below the input field, there are buttons for 'Dump Messages', 'Mute', and 'Unsubscribe'. The 'Topics Collector' section shows a single topic '\$oc/devices/[REDACTED]/user/temperature'. In the 'Log' section, a message is shown with a red box around it:

```
{"name": "command", "id": "[REDACTED]", "content": "{\"airConditioner\": 'start', 'mode': 'cold'}"}
```

## 说明

- 确保创建自定义Topic时，设备操作权限包含“订阅”功能，详细操作请参考[自定义Topic](#)。
  - 使用IoT Device SDK或者原生MQTT协议接入的设备，需要在设备程序中，设置设备订阅的自定义Topic名称。
2. 使用postman工具模拟应用侧调用[下发设备消息](#)接口，下发开启室内空调的命令。



send\_mqtt\_message

send\_mqtt\_message

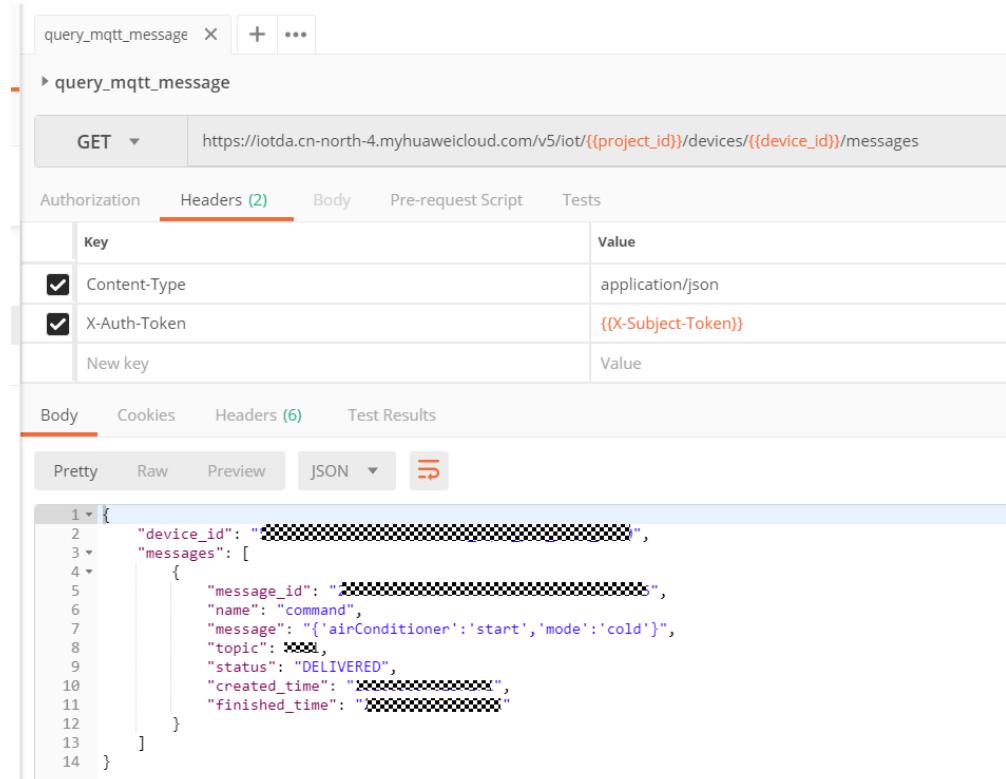
POST https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/{{project\_id}}/devices/{{device\_id}}/messages

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2 "name": "command",
3 "message": "{ 'airConditioner': 'start', 'mode': 'cold' }",
4 "topic": "temperature"
5 }
```

3. 调用[查询设备消息](#)接口，查看下发命令是否成功。若命令下发成功，室内空调将被开启。



query\_mqtt\_message

query\_mqtt\_message

GET https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/{{project\_id}}/devices/{{device\_id}}/messages

Authorization Headers (2) Body Pre-request Script Tests

Content-Type application/json

X-Auth-Token {{X-Subject-Token}}

Body Cookies Headers (6) Test Results

Pretty Raw Preview JSON

```
1 {
2 "device_id": "██",
3 "messages": [
4 {
5 "message_id": "██",
6 "name": "command",
7 "message": "{ 'airConditioner': 'start', 'mode': 'cold' }",
8 "topic": "████████",
9 "status": "DELIVERED",
10 "created_time": "██████████",
11 "finished_time": "██████████"
12 }
13]
14 }
```

## 2.19 基于 Jmeter 工具进行 MQTT 性能测试

### 场景说明

随着物联网技术的发展，全球物联网设备连接规模与日俱增。大量设备接入和设备管理对网络带宽、通信协议以及平台服务架构都带来了很大挑战。IoT架构选型时，对平台进行性能测试显得尤为重要。本文介绍如何使用JMeter工具对平台的MQTT协议接入能力进行性能压测。

本文描述的测试方案如下：

测试场景：

- 模拟1万个设备同时在线，验证平台长连接的稳定性。
- 模拟设备每秒发起100次消息上报，验证平台消息处理能力。

测试环境：

- 压测目标平台：华为云设备接入服务低频单元S1（1万设备在线、200TPS消息上下行）。
- 压测执行机：Jmeter执行机1台，规格如下：

表 2-11 压测执行机

| 实例类型    | 规格名称        | vCPUs  | 内存   |
|---------|-------------|--------|------|
| 通用计算型s6 | s6.xlarge.2 | 4vCPUs | 8GiB |

#### 说明

单台Jmeter执行机最多能模拟5万设备在线，如果需要模拟更高规格的在线设备数可以使用[华为云的CPTS服务](#)，部署多台Jmeter执行机，以满足规格测试的要求。

### 前提条件

- 已注册华为帐号并开通华为云。未注册可点击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上点击[实名认证](#)完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，选择“购买实例”，购买“低频单元S1（1万设备在线、200TPS消息上下行）”。

### 前期准备

- Jmeter执行机安装Java运行环境。访问[Java官网](#)，下载并参照说明安装Java运行环境。
- [下载](#)并安装JMeter（JMeter 应使用5.1.1及以上版本）。
- [下载](#)mqtt-jmeter 插件包，并将插件放置于JMeter安装目录的“lib/ext”目录下。

## 业务流程

使用Jmeter工具对平台进行MQTT协议进行性能压测的具体流程如下：

- 步骤1 **创建产品**。创建一个MQTT协议的产品。
- 步骤2 **批量注册设备**。采用批量导入的方式注册1万个用于性能测试的设备。
- 步骤3 **导入测试计划**。导入已经创建的用于IoT性能测试的测试计划。
- 步骤4 **发起压测**。根据业务规格发起对平台的性能压测。
- 步骤5 **查看压测结果**。在IoT平台上根据监控指标检查压测结果是否符合预期。

----结束

## 创建产品

- 步骤1 登录**管理控制台**，单击左侧导航栏“产品”，单击页面左侧的“**创建产品**”。
- 步骤2 根据页面提示填写参数，然后单击“**确定**”。

表 2-12 参数说明

| 基本信息   |                                                                                |
|--------|--------------------------------------------------------------------------------|
| 所属资源空间 | 平台自动将新创建的产品归属在默认资源空间下。如需归属在其他资源空间下，下拉选择所属的资源空间。如无对应的资源空间，请先创建 <b>资源空间</b> 。    |
| 产品名称   | 自定义。支持字母、数字、下划线（_）、连字符（-）的字符组合。                                                |
| 协议类型   | 选择MQTT。                                                                        |
| 数据格式   | 选择JSON。                                                                        |
| 所属行业   | 请根据产品模型的所属行业填写。                                                                |
| 设备类型   | 使用平台预置的产品模型，会自动关联设备类型，不需要再输入设备类型。                                              |
| 高级配置   |                                                                                |
| 产品ID   | 定制ProductID，用于唯一标识一个产品。如果携带此参数，平台将产品ID设置为该参数值；如果不携带此参数，产品ID在物联网平台创建产品后由平台分配获得。 |

----结束

## 批量注册设备

- 步骤1 访问**设备接入服务**，单击“管理控制台”进入设备接入控制台。
- 步骤2 在左侧导航栏选择“设备 > 所有设备”，进入“**批量注册**”页签，单击“**批量注册**”。
- 步骤3 下载并填写“**批量注册设备文件模板**”，可以参照如下表进行参数填写。**样例参考**。

表 2-13 参数说明

| 参数名         | 参数说明                                         |
|-------------|----------------------------------------------|
| node_id     | 设备标识，性能测试时可以按递增的方式进行填写，如 10001, 10002, 10003 |
| product_id  | 产品ID， <a href="#">创建产品</a> 时生成的产品ID。         |
| app_id      | 资源空间，获取方式参考 <a href="#">获取资源空间</a>           |
| device_name | 设备名称，可以同node_id一致。                           |
| secret      | 设备密钥，性能测试时可以固定写死一个密钥。                        |

**步骤4** 在“批量注册设备窗口”点击“添加文件”上传写好的批量注册文件模板，单击“确定”创建任务。

#### 步骤5

图 2-196 设备-批量注册设备



**步骤6** 批量注册执行成功后，保存好设备ID和密钥。

----结束

## 导入测试计划

**步骤1** [下载](#)用于JMeter测试的测试计划。

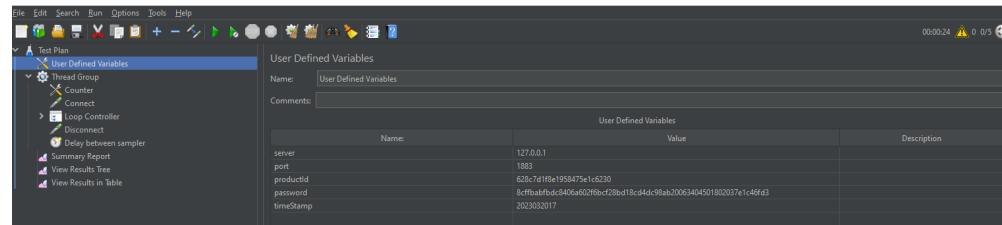
**步骤2** 打开JMeter，选择“Open”导入下载的测试计划。

**步骤3** 在 JMeter左侧文件目录中单击“User Defined Variables”，在“User Defined Variables”对话框中配置以下信息。

表 2-14 参数说明

| 参数名       | 参数说明                                                                                                                   |
|-----------|------------------------------------------------------------------------------------------------------------------------|
| server    | MQTT协议服务端地址。获取方式参考 <a href="#">获取接入地址</a> 。                                                                            |
| port      | MQTT协议端口，填写为8883。                                                                                                      |
| productId | 产品ID， <a href="#">创建产品</a> 时生成的产品ID。                                                                                   |
| password  | MQTT连接密码。为使用“HMACSHA256”算法以时间戳为密钥对secret进行加密后的值。secret为 <a href="#">批量注册设备</a> 时填写的密钥。可以通过 <a href="#">工具</a> 获取加密后的值。 |
| timeStamp | 为对password进行加密的时间戳。时间格式为YYYYMMDDHH。                                                                                    |

图 2-197 参考样例

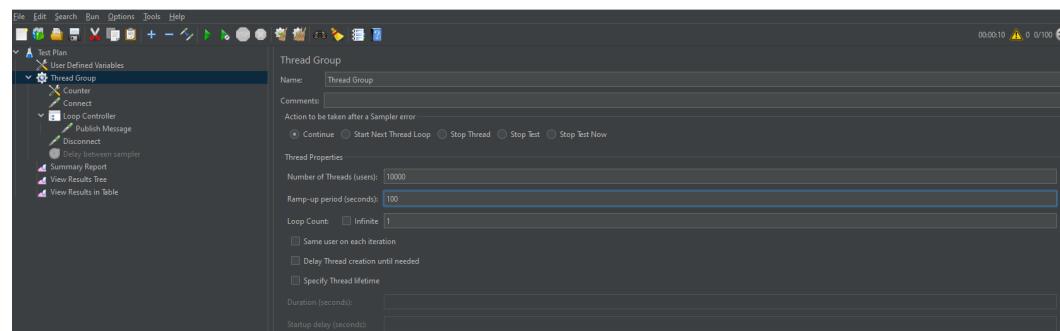


----结束

## 发起压测

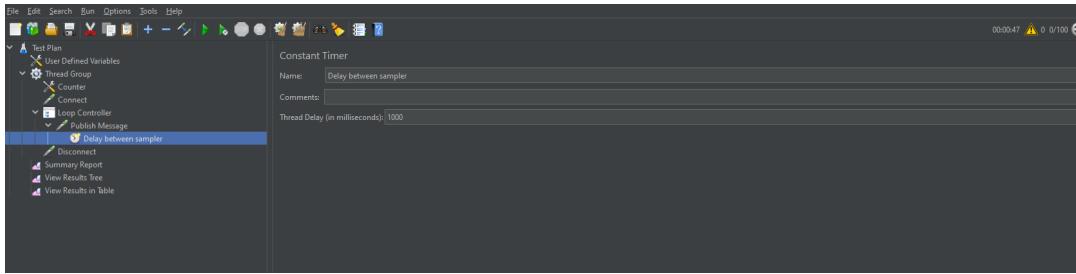
**步骤1** 在 JMeter 左侧文件目录中单击 “Thread Group”，配置 “Number of Threads” 为 10000（一个线程对应一个设备在线，配置为10000表示设备与平台间保持1万个设备在线）。

图 2-198 配置设备



**步骤2** 在 JMeter 左侧文件目录 “Thread Group” 目录下单击 “Loop Controller” -> “Publish Message ” -> "Delay between sampler"，配置 “Thread Delay(in milliseconds)” 为100000（每设备每100秒发布一次消息）。

图 2-199 配置设备



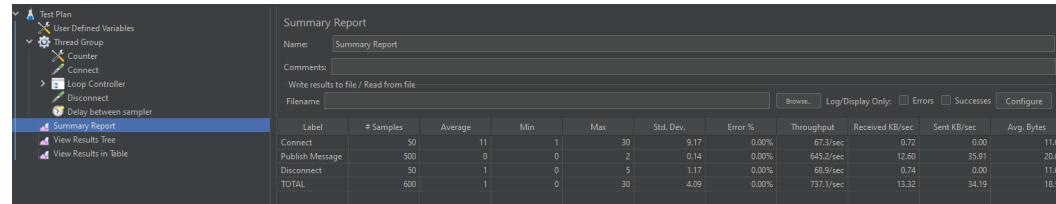
步骤3 在Jmeter工具栏图表中点击“开始”按钮，启动性能测试。

图 2-200 性能测试



步骤4 在 JMeter 左侧文件目录中单击“Summary Report”，可以看到“Connect”、“Publish Message”的吞吐量。可以通过调整“Number of Threads”和“Thread Delay(in milliseconds)”来调整Jmeter发起的吞吐量。

图 2-201 性能测试



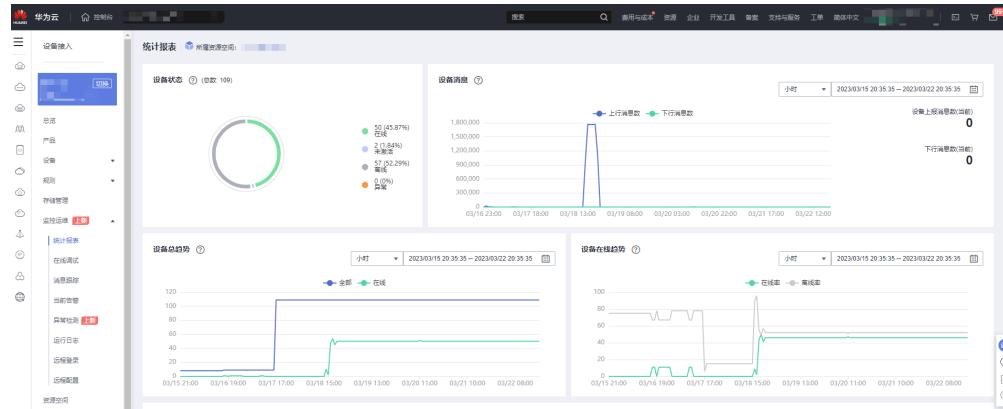
步骤5 JMeter测试计划调试正常后，可以将测试计划导入到[华为云的CodeArts PerfTest服务](#)，进行分布式部署以满足更高规格的性能测试。

----结束

## 查看压测结果

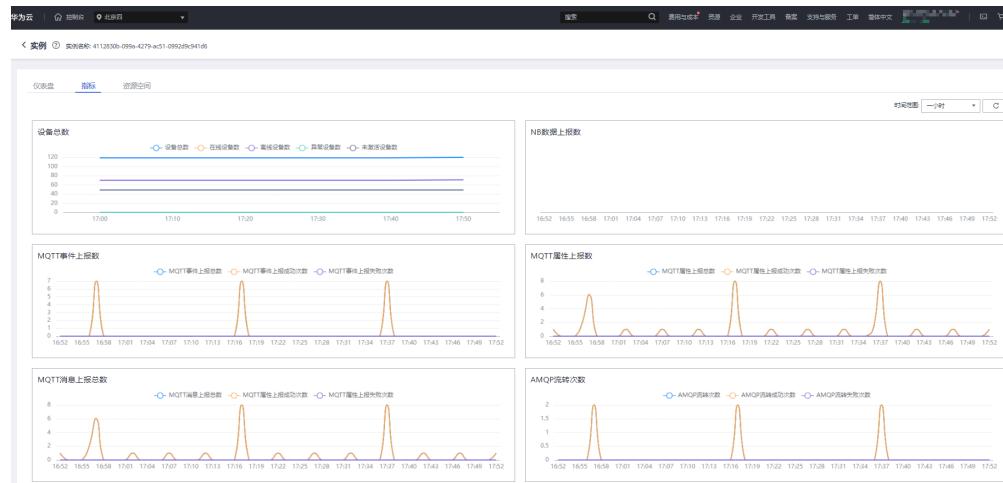
1. 登录[管理控制台](#)，在左侧导航栏“监控运维”->“统计报表”，查看平台的统计指标。

图 2-202 查看统计报表



2. 更多报表，可登录[应用运维管理](#)控制台，选择监控 > 云服务监控 > 物联网>IoTDA 查看。

图 2-203 查看指标



# 3 数据转发

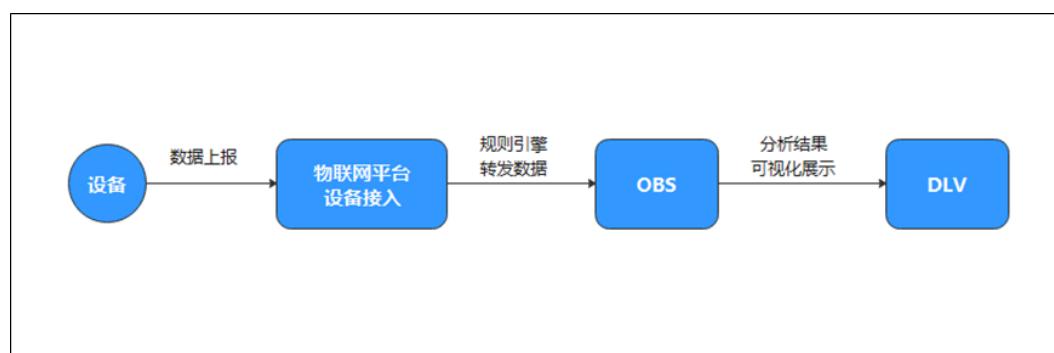
## 3.1 零代码搭建物联网监控大屏

### 场景说明

物联网解决方案中，作为数据主体的“物”可能数量会非常大，产生的数据已经无法通过传统的数据处理服务进行处理。如何分析与利用这庞大的物联网设备数据对物联网企业来说又是一个新的挑战。

华为云物联网平台提供规则引擎能力，支持将设备上报的数据转发至华为云其他云服务，可实现将海量数据通过对象存储服务（OBS）存储，再由数据可视化服务（DLV）读取数据呈现为可视化报表，实现数据的一站式采集、处理和分析。

图 3-1 场景说明



在本示例中，我们实现下述场景：

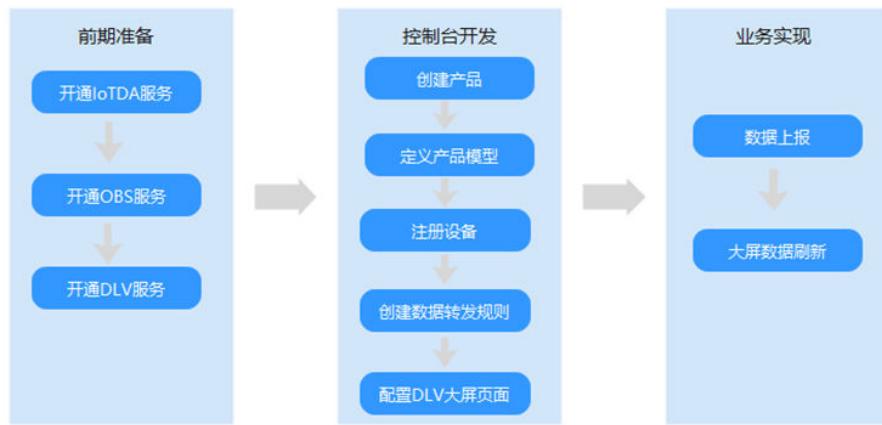
设备上报数据到物联网平台，物联网平台将数据以CSV格式存储到对象存储服务（OBS），数据可视化服务(DLV)从OBS文件读取数据并展现为多个维度的报表。

### 整体流程

1. 创建MQTT协议产品、开发产品模型、并注册设备。
2. 创建数据转发规则，将数据以CSV格式持久化至OBS服务。
3. 配置数据可视化服务（DLV）展示OBS中物联网数据。

4. 使用设备模拟器上报数据，验证DLV大屏数据实时刷新。

图 3-2 整体业务流程图



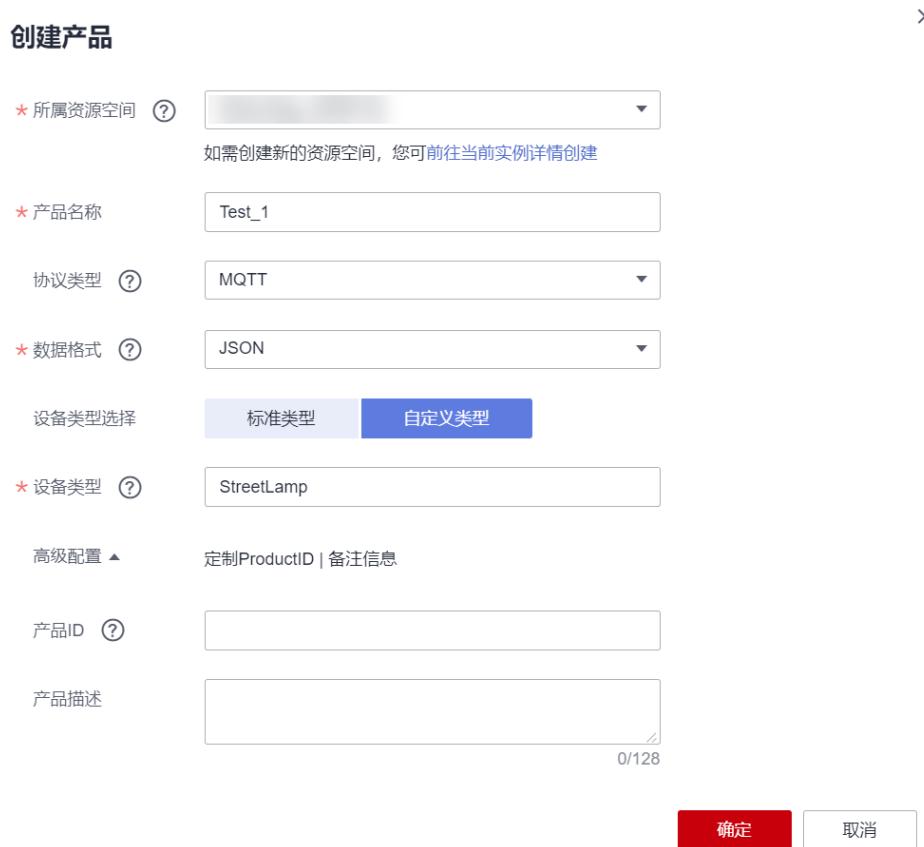
## 前提条件

- 已注册华为云官方帐号。未注册可单击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击[实名认证](#)完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“立即使用”后开通该服务。
- 已开通OBS服务。未开通则访问[OBS服务](#)，单击“立即使用”后开通该服务。
- 已开通数据可视化DLV服务。未开通则访问[数据可视化DLV服务](#)，单击“立即使用”后开通该服务。

## 创建产品和设备

- 步骤1 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 步骤2 选择左侧导航栏的“产品”，单击右上角下拉框，选择新建产品所属的资源空间。
- 步骤3 单击左侧的“创建产品”，创建一个基于MQTT协议的产品，填写参数后，单击“确认”。

图 3-3 创建产品-MQTT



**步骤4** 为上一步创建的产品导入产品模型，请参考[导入产品模型](#)。

**步骤5** 选择第3步创建的产品，在该产品下注册设备、请参考[注册单个设备](#)。

#### □□ 说明

本文中使用的产品模型和设备仅为示例，您可以使用自己的产品模型和设备进行操作。

----结束

## 数据转发规则配置

**步骤1** 选择左侧导航栏的“规则>数据转发”，单击左上角的“创建规则”。

**步骤2** 参考下表参数说明，填写规则内容。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的规则，填写完成后单击“创建规则”。

图 3-4 新建属性上报流转规则-数据转发至 OBS

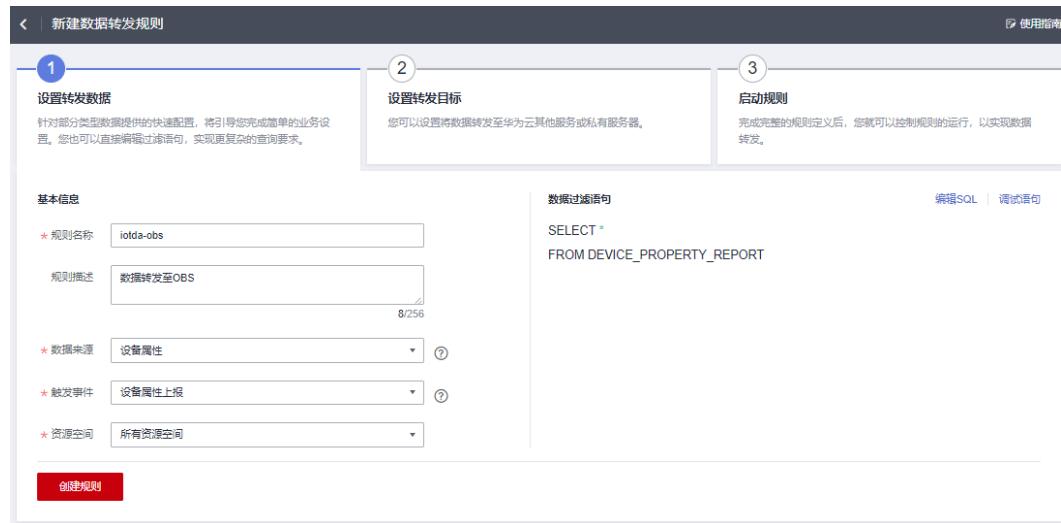
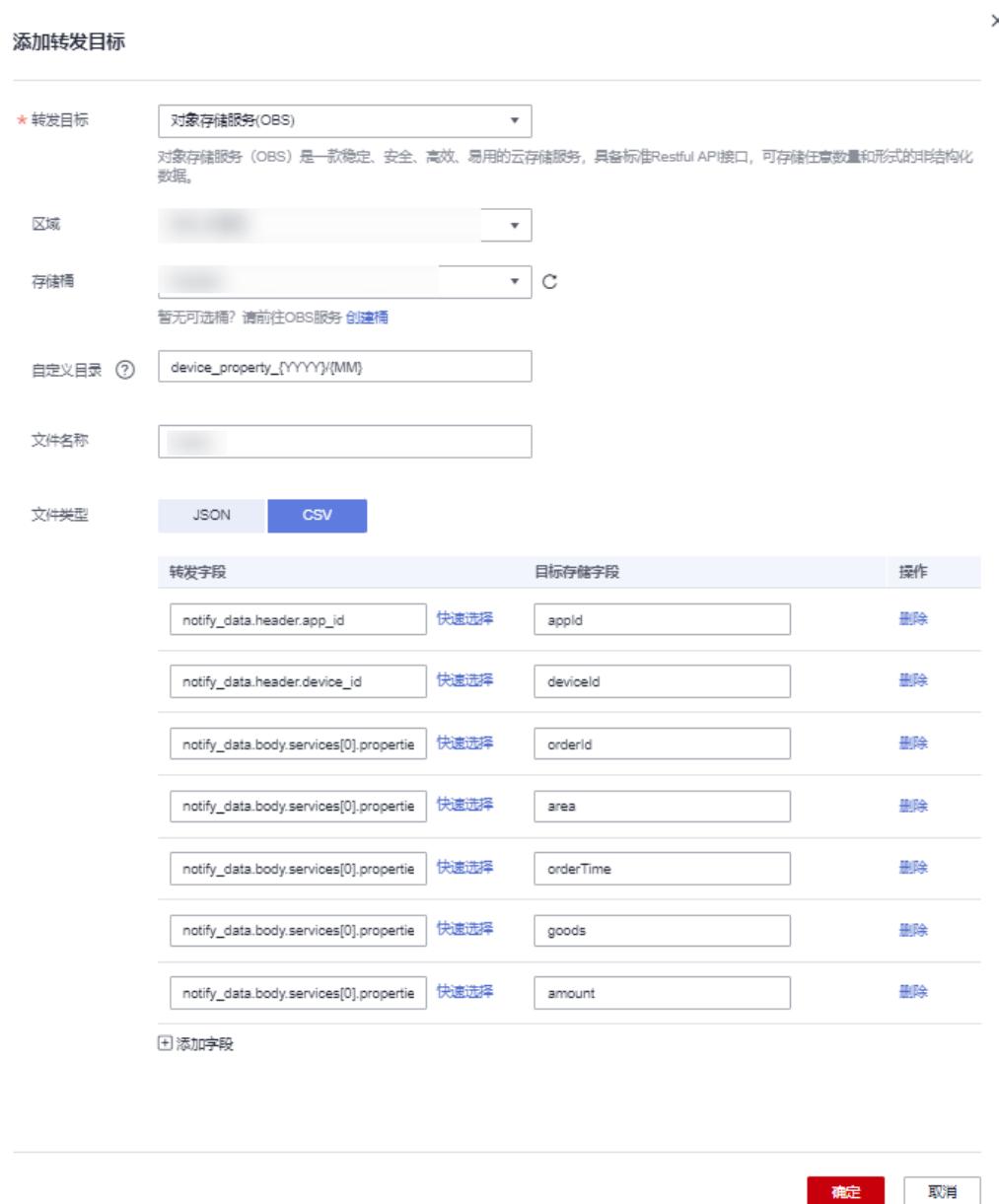


表 3-1 参数说明

| 参数名  | 参数说明                  |
|------|-----------------------|
| 规则名称 | 自定义，如iotda-obs。       |
| 规则描述 | 自定义，如数据转发至OBS服务。      |
| 数据来源 | 选择“设备属性”。             |
| 触发事件 | 自动匹配“设备属性上报”。         |
| 资源空间 | 和上一步创建的产品所在的资源空间保持一致。 |

步骤3 单击“设置转发目标”页签，单击“添加”，设置转发目标。

图 3-5 新建转发目标-转发至 OBS-CVS 格式



参考下表参数说明，填写转发目标。填写完成后单击“确定”。

表 3-2 参数说明

| 参数名   | 参数说明                                         |
|-------|----------------------------------------------|
| 转发目标  | 选择“对象存储服务（OBS）”                              |
| 区域    | 选择“对象存储服务”区域。若未授权访问此区域的服务，请根据界面提示，配置云服务访问授权。 |
| 存储桶   | 选择桶。若没有，请前往OBS服务 <a href="#">创建桶</a> 。       |
| 自定义目录 | 自定义存储数据目录                                    |

| 参数名            | 参数说明                   |
|----------------|------------------------|
| 文件名称           | 自定义存储数据文件名             |
| 文件类型           | 选择“CSV”                |
| 转发字段----目标存储字段 | OBS服务中生成文件和流转数据的对应关系列表 |

本实践中用到的转发字段和转发目标如下：

表 3-3 转发字段和转发目标

| 转发字段                                              | 转发目标      |
|---------------------------------------------------|-----------|
| notify_data.header.app_id                         | appId     |
| notify_data.header.device_id                      | DeviceId  |
| notify_data.body.services[0].properties.orderID   | orderID   |
| notify_data.body.services[0].properties.area      | area      |
| notify_data.body.services[0].properties.orderTime | orderTime |
| notify_data.body.services[0].properties.goods     | goods     |
| notify_data.body.services[0].properties.amount    | amount    |

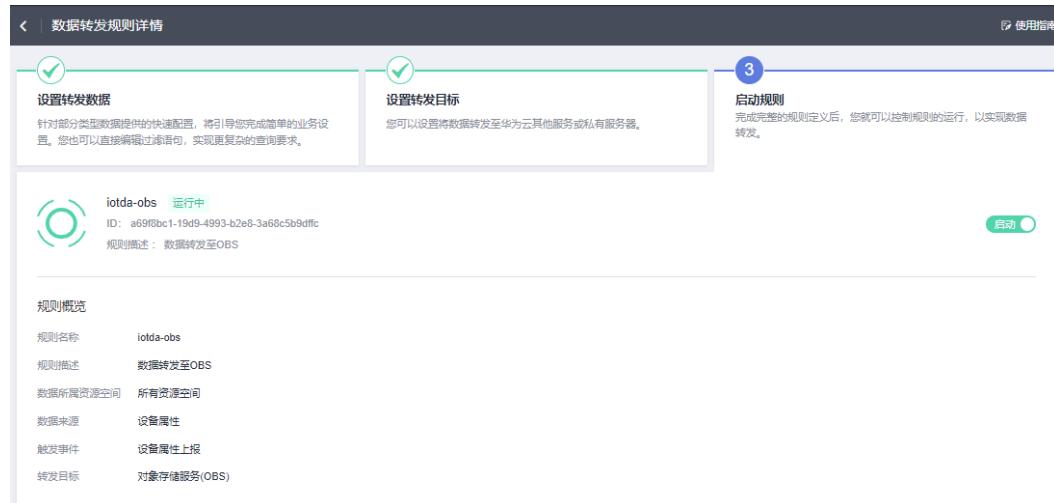
示例：如果设备上报的属性如下，需要把“orderID”保存到CSV文件，需要设置转发字段为“notify\_data.body.services[0].properties.orderID”，转发目标为“orderID”。

```
{
 "resource": "device.property",
 "event": "report",
 "event_time": "20220812T121212Z",
 "notify_data": {
 "header": {
 "device_id": "d4922d8a-6c8e-4396-852c-164aefa6638f",
 "product_id": "ABC123456789",
 "app_id": "d4922d8a-6c8e-4396-852c-164aefa6638f"
 },
 "body": {
 "services": [{
 "service_id": "order",
 "properties": {
 "orderID": "0001"
 },
 "event_time": "20220812T121212Z"
 }]
 }
 }
}
```

{  
}

**步骤4** 单击“启动规则”，激活配置好的数据转发规则。

**图 3-6 启动规则-转发至 OBS**



----结束

## 数据上报及验证数据是否转发成功

**步骤1** 使用MQTT模拟器连接到平台（模拟器使用请参考：[使用MQTT.fx调测](#)）。

**步骤2** 使用模拟器进行属性上报(参考：[设备属性上报](#)）。

上报内容如下：

```
{
 "services": [
 {
 "service_id": "order",
 "properties": {
 "orderID": "0000001",
 "area": "南山区",
 "orderTime": "20220815T121215Z",
 "goods": "商品01",
 "amount": 30.5
 },
 "event_time": "20220815T121215Z"
 }
]
}
```

**步骤3** 前往[OBS控制台](#)，查看转发目标设置的文件是否生成。

图 3-7 查看 OBS 桶内文件



----结束

## 配置数据可视化服务

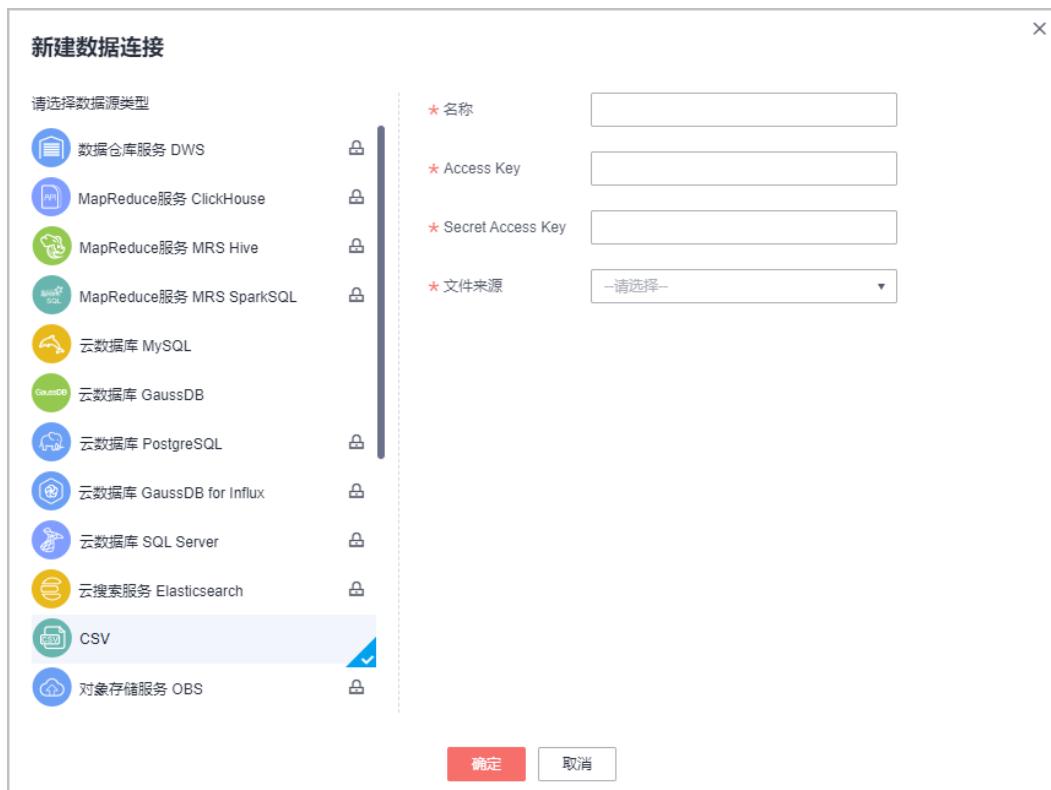
配置数据可视化服务，新建数据报表视图。

- 步骤1 登录华为云官方网站，访问[数据可视化服务](#)。
- 步骤2 单击“进入控制台”。
- 步骤3 访问DLV控制台“我的数据”页面，单击“新建数据连接”，新建CSV文件数据连接（详情请参考：[新建CSV文件数据连接](#)）。

### 说明

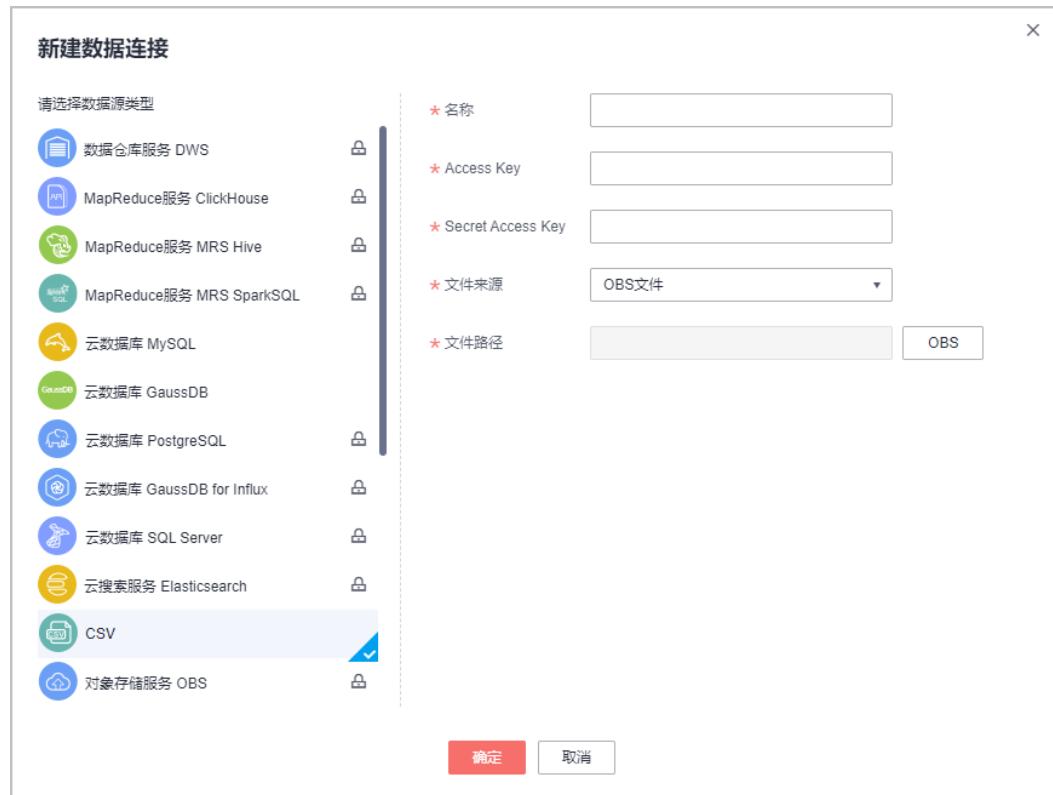
若您未开通DLV服务，可单击“体验试用”获取30天的基础版免费试用。

图 3-8 新建数据连接



**步骤4** 新建CSV文件数据连接中文件来源选择OBS文件，文件路径选择创建规则时指定的转发目标。

图 3-9 选择文件来源



**步骤5** 访问DLV控制台“我的大屏”页面，新建大屏添加并编辑组件实现下图效果(参考：[创建大屏、编辑大屏](#))。也可以通过附件导入大屏（具体操作步骤请参考：[导入导出大屏，自动售货机监控](#)）。

图 3-10 大屏效果展示



**步骤6** 选择上一步创建或导入的大屏进入编辑页面，分别设置每个组件的数据源。

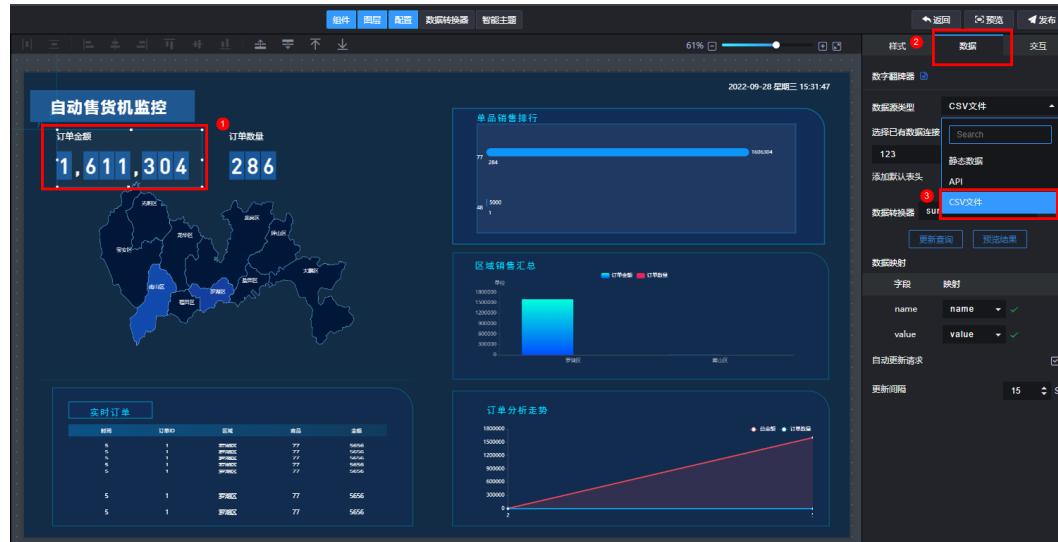
**⚠ 注意**

通过附件导入大屏时，各个组件的数据转换器已经设置完成，只需要修改数据源为自己的CSV文件，即可完成大屏开发。

如果需要手动开发组件，下面以“订单金额”为例，演示如何将CSV文件中数据展示到DLV。

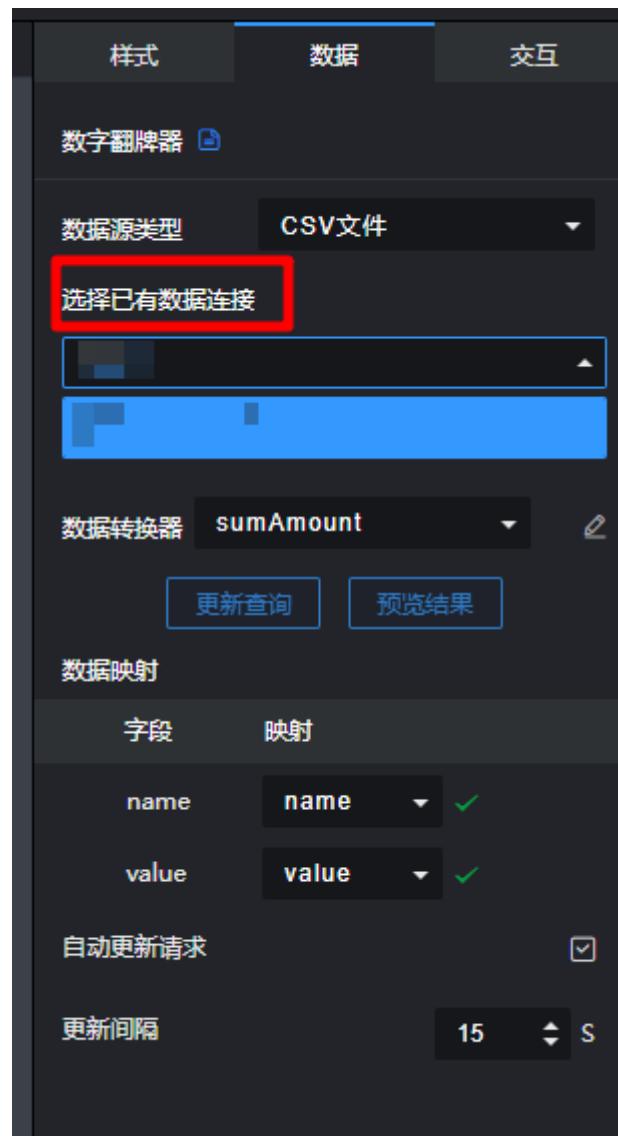
**步骤7** 选中“订单金额”对应的数字翻牌组件，在右侧属性窗口，切换到“数据”页签，选择数据源类型为“CSV”。

**图 3-11** 添加数据源



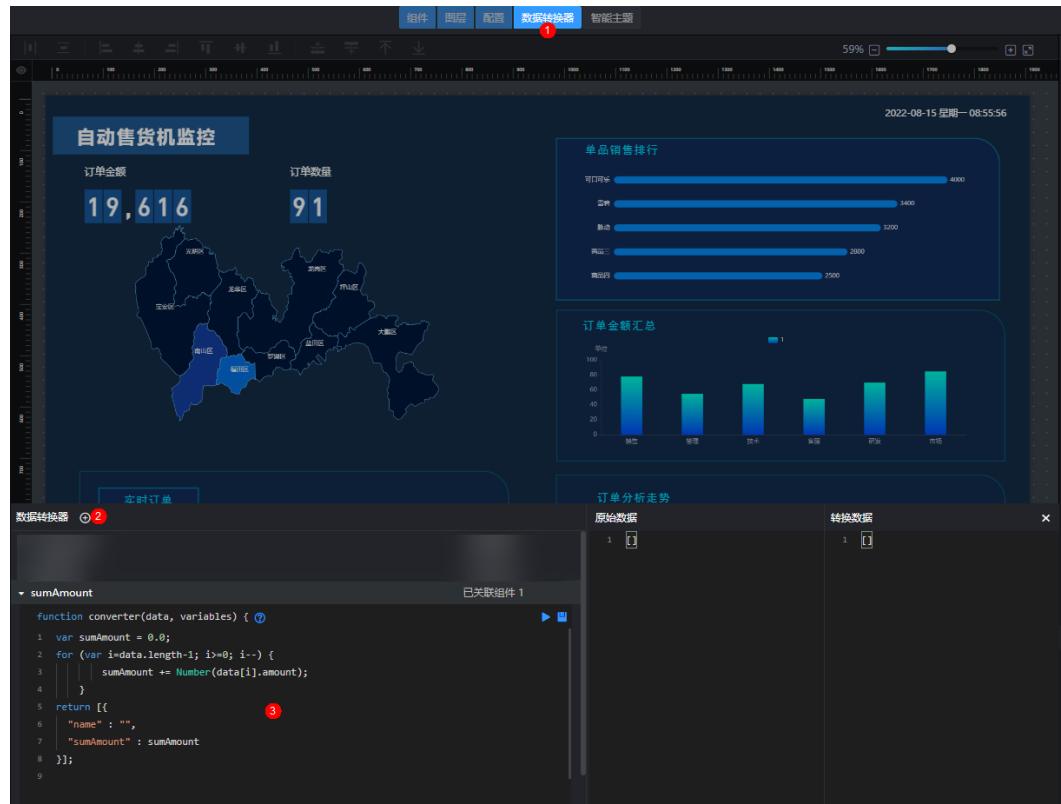
**步骤8** 选择**步骤4**创建的数据连接。

图 3-12 选择已有数据连接



**步骤9** 切换到“数据转换器”页签，新建一个名称为“sumAmount”的数据转换器，编写转换器代码(参考：[使用数据转换器](#) )。

图 3-13 添加数据转换器



转换器源码：

```
var sumAmount = 0.0;
for(var i = data.length-1; i >=0; i--){
 sumAmount +=Number(data[i].amount);
}
return[{
"name":"",
"value": sumAmount
}];
```

**步骤10** 选择组件，更改数据转换器为“sumAmount”，点击“更新查询”，就可以看到数据刷新到组件。（这样“订单金额”组件就完成开发，按照这个方法即可完成其他组件的开发。）

图 3-14 查询更新数据



**步骤11** 大屏制作完成后点击右上角的“预览”或“发布”按钮即可完成大屏的预览和发布。

----结束

## 模拟数据上报及结果验证

**步骤1** 使用MQTT模拟器连接到平台（模拟器使用请参考：[使用MQTT.fx调测](#)）。

**步骤2** 使用模拟器进行属性上报(参考：[设备属性上报](#))。

上报内容如下：

```
{
 "services": [
 {
 "service_id": "order",
 "properties": {
 "orderId": "0000002",
 "area": "罗湖区",
 "orderTime": "20220815T121215Z",
 "goods": "商品02",
 "amount": 30.5
 },
 "event_time": "20220815T121215Z"
 }
]
}
```

**步骤3** 打开数据可视化服务查看数据是否刷新。

图 3-15 结果查看

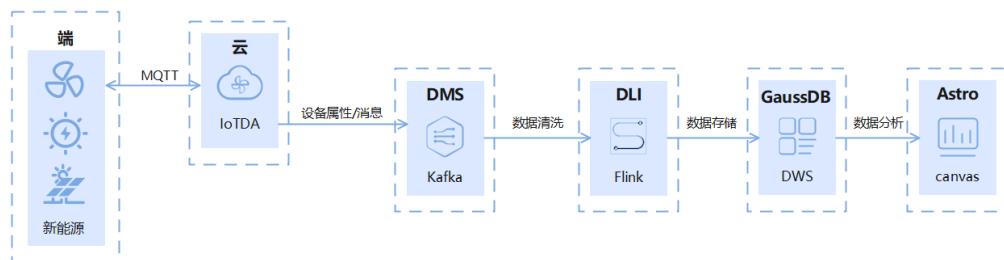


## 3.2 IoTDA 结合 DLI+DWS+Astro 实现大数据分析

### 场景说明

在物联网解决方案中，设备产生庞大的数据，使用传统的Mysql数据库已经无法达到要求，为了降低存储成本，提高数据查询效率，华为云物联网平台可以通过规则引擎，将数据转发到华为云其他云服务，例如可以将海量数据经过数据湖探索服务（DLI）进行数据清洗，再由云数据仓库（DWS）进行存储，再由Astro大屏应用读取数据进行可视化展示。实现数据一站式存储、处理和分析。

图 3-16 场景说明



在本示例中，我们实现以下场景：

新能源设备上报发电量、地区等数据到物联网平台，物联网平台将数据转发至kafka，由DLI进行大数据清洗将数据存储至DWS，再由Astro大屏应用读取数据进行可视化展示实现各个地区发电量的统计分析。

### 整体流程

- 1、创建MQTT协议产品，并创建设备。

- 2、创建流转规则，将数据流转至kafka。
- 3、配置DLI以及DWS，将kafka中的数据进行清洗并转发至DWS。
- 4、配置Astro大屏应用，读取DWS数据进行展示。

## 前提条件

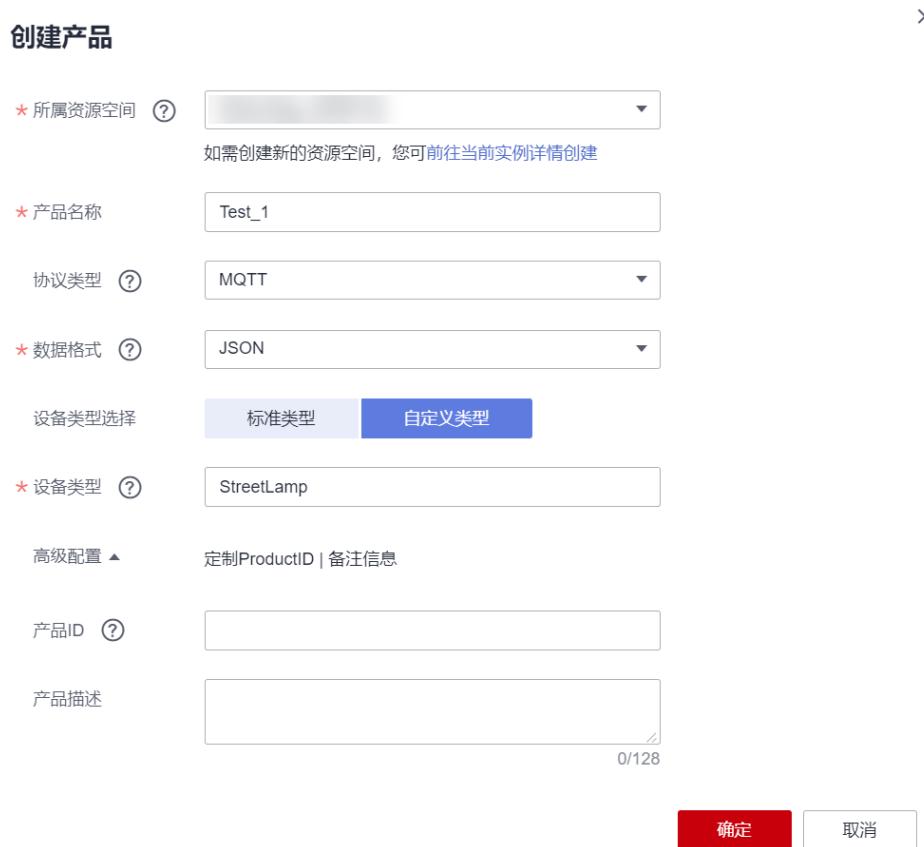
- 已注册华为官方帐号。未注册可参考[注册华为账户](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击[实名认证](#)完成认证，否则会影响后续云服务的开通。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“免费试用”或单击“价格计算器”购买并开通该服务。
- 已开通Kafka服务。未开通则访问[Kafka服务](#)，单击“立即购买”后开通该服务。
- 已开通数据湖探索服务。未开通则访问[数据湖探索服务](#)，单击“立即购买”后开通该服务。
- 已开通云数据仓库GaussDB(DWS)服务。未开通则访问[云数据仓库 GaussDB\(DWS\)服务](#)，单击“购买”后开通该服务。
- 已开通Astro大屏服务，未开通则访问[Astro大屏应用](#)，单击“立即使用”后开通该服务。

## 创建产品和设备

**步骤1** 访问[设备接入服务](#)，单击“管理控制台”进入设备接入控制台。

**步骤2** 选择左侧导航栏的“产品”，单击“创建产品”，创建一个基于MQTT协议的产品，填写参数后，单击“确认”。

图 3-17 创建产品-MQTT



**步骤3** 导入产品模型，请参考[导入产品模型](#)。

**步骤4** 在该产品下注册设备，请参考[注册单个设备](#)。

#### □□ 说明

本文中使用的产品模型和设备仅为示例，您可以使用自己的产品模型和设备进行操作。

----结束

## 数据转发规则配置

**步骤1** 选择左侧导航栏的“规则>数据转发”，单击“创建规则”。

**步骤2** 参考下表参数说明，填写规则内容。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的规则，填写完成后单击“创建规则”。

图 3-18 新建消息上报流转规则-数据转发至 Kafka

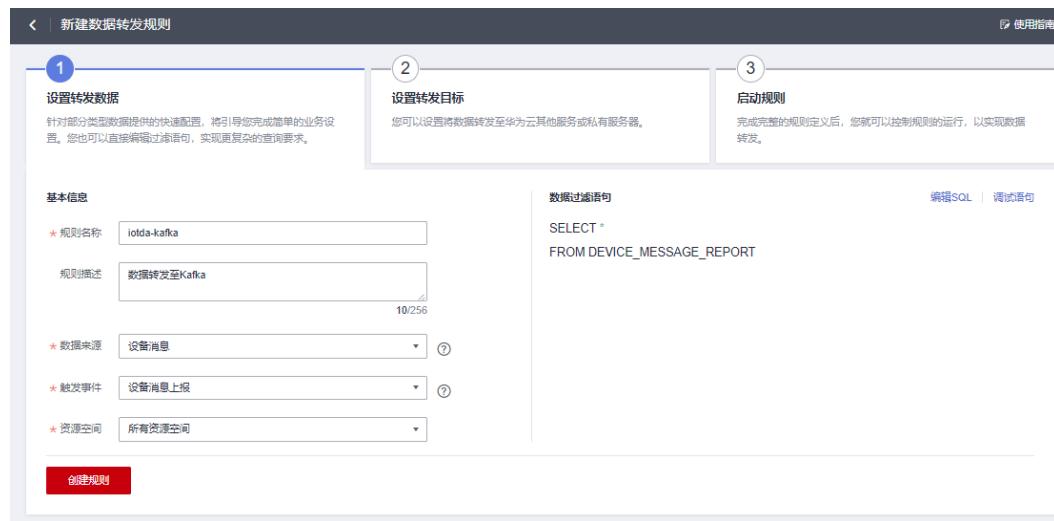


表 3-4 参数说明

| 参数名    | 参数说明                    |
|--------|-------------------------|
| 规则名称   | 自定义，如iotda-kafka。       |
| 规则描述   | 自定义，如数据转发至Kafka服务。      |
| 数据来源   | 选择“设备消息”。               |
| 触发事件   | 自动匹配“设备消息上报”。           |
| 资源空间   | 和上一步创建的产品所在的资源空间保持一致。   |
| 数据过滤语句 | 通过编写SQL来解析和处理上报的JSON数据。 |

## 说明

该示例使用如下数据过滤语句进行转发：

```
notify_data.body.content.id as id, notify_data.body.content.device_id as device_id,
notify_data.body.content.report_time as report_time, notify_data.body.content.province as province,
notify_data.body.content.city as city, notify_data.body.content.daily_power_generation as
daily_power_generation, notify_data.body.content.total_power_generation as total_power_generation,
notify_data.body.content.total_power as total_power, notify_data.body.content.running_status as
running_status
```

**步骤3** 单击“设置转发目标”页签，单击“添加”，设置转发目标。

图 3-19 新建转发目标-转发至 Kafka

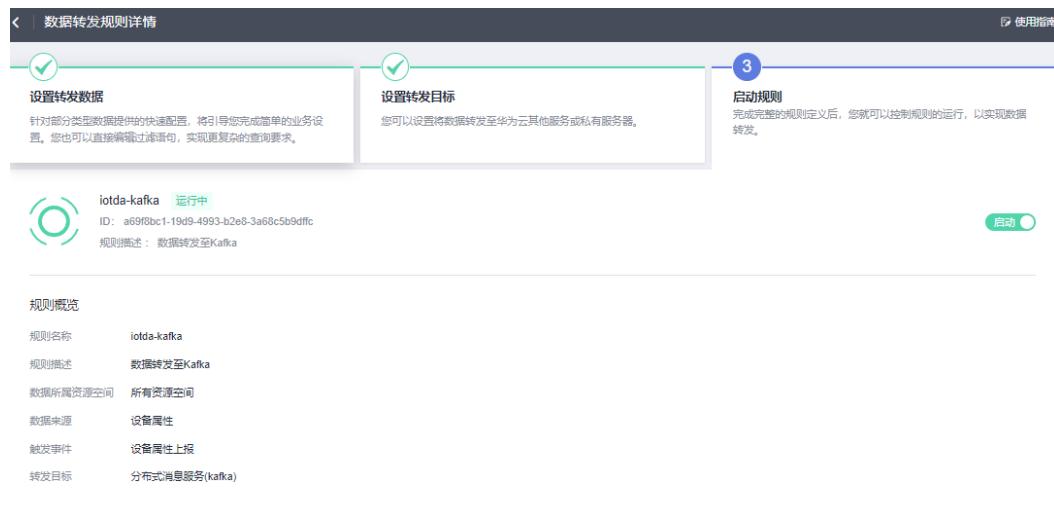
参考下表参数说明，填写转发目标。填写完成后单击“确定”。

表 3-5 参数说明

| 参数名       | 参数说明                                                                       |
|-----------|----------------------------------------------------------------------------|
| 转发目标      | 选择“分布式消息服务（Kafka）”                                                         |
| 区域        | 选择“分布式消息服务”区域。                                                             |
| 对接地址      | 选择kafka服务的对接地址。                                                            |
| 主题        | 填写接收推送消息的kafka topic。                                                      |
| SASL认证    | 若开启SASL认证，请填写您在 <a href="#">购买Kafka实例</a> 中所选安全协议、SASL认证机制以及所填的SASL用户名和密码。 |
| Kafka安全协议 | 填写您在 <a href="#">购买Kafka实例</a> 中所启用的kafka安全协议。                             |
| SASL认证机制  | 填写您在 <a href="#">购买Kafka实例</a> 中所开启的SASL认证机制。                              |
| SASL用户名   | 填写您在 <a href="#">购买Kafka实例</a> 中输入的SASL用户名。                                |
| 密码        | 填写您在 <a href="#">购买Kafka实例</a> 中输入的密码。                                     |

步骤4 单击“启动规则”，激活配置好的数据转发规则。

图 3-20 启动规则-转发至 Kafka



----结束

## 数据上报及验证数据是否转发成功

**步骤1** 使用MQTT模拟器连接到平台（模拟器使用请参考：[使用MQTT.fx调测](#)）。

**步骤2** 使用模拟器进行消息上报(参考：[设备消息上报](#))。上报内容如下：

```
{
 "id": "2",
 "device_id": "6459f49b97bacf7a56439f97_321135434423",
 "report_time": "2023-10-19 19:39:42",
 "province": "guangzhou",
 "city": "shengzhen",
 "daily_power_generation": 32,
 "total_power_generation": 108,
 "total_power": 1023,
 "running_status": "ONLINE"
}
```

**步骤3** 前往[Kafka控制台](#)，查看是否收到转发的消息。

图 3-21 查看 Kafka 消息



----结束

## 配置 DWS 数据仓库服务

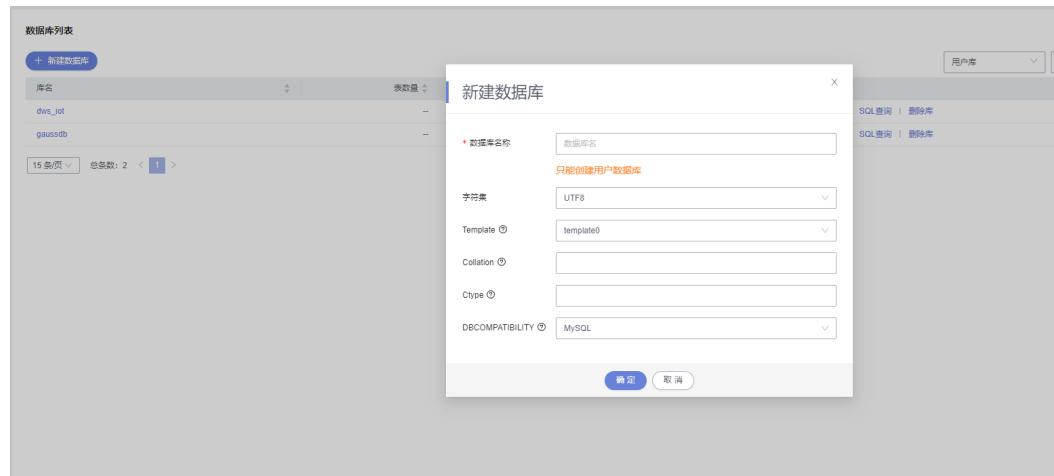
**步骤1** 登录华为云官方网站，访问[数据仓库服务](#)。

**步骤2** 单击“进入控制台”。

**步骤3** 访问数据仓库服务控制台“集群管理”页面，选择购买的集群，若无集群，则参考[购买集群](#)，购买集群后，点击登录集群。

**步骤4** 登录集群后，点击新建数据库，创建数据库。

图 3-22 创建数据库



**步骤5** 点击SQL查询，进入SQL查询页面，执行以下SQL语句创建数据表。

```
CREATE TABLE IF NOT EXISTS power_test_dws (
 "id" BIGINT,
 "report_time" TIMESTAMP,
 "province" varchar(255),
 "city" varchar(255),
 "device_id" varchar(255) NOT NULL,
 "daily_power_generation" numeric(10,2),
 "total_power_generation" numeric(10,2),
 "total_power" numeric(10,1),
 "running_status" varchar(10),
 CONSTRAINT "device_unique" UNIQUE ("device_id", "report_time")
) WITH (ORIENTATION = COLUMN, PERIOD='15 days', storage_policy = 'LMT:180') PARTITION BY RANGE
(report_time);
```

----结束

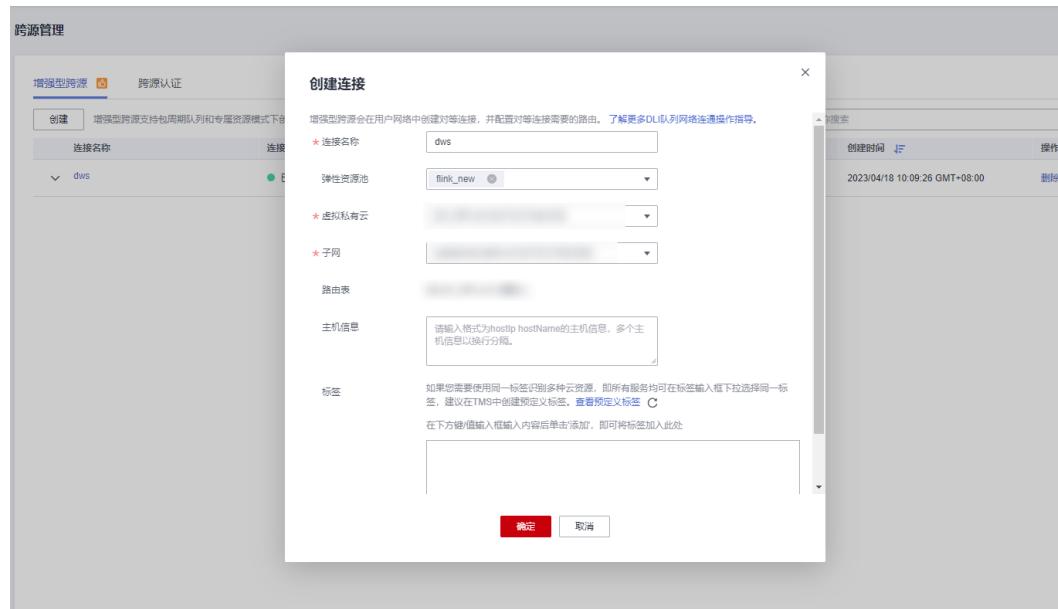
## 配置 DLI 数据湖探索服务

**步骤1** 登录华为云官方网站，访问[数据湖探索服务](#)。

**步骤2** 单击“进入控制台”。

**步骤3** 在跨源管理中创建增强型跨源，用于打通Flink队列与DMS Kafka和DWS之间的网络。弹性资源池选择购买的DLI队列，虚拟私有云选择Kafka和DWS所属的虚拟私有云以及子网。

图 3-23 跨源管理



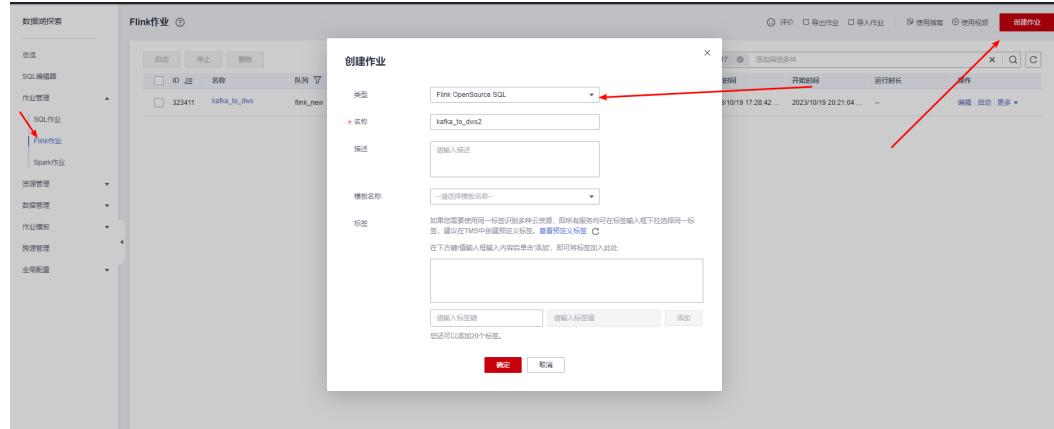
**步骤4** 创建完跨源管理后，可在“资源管理 > 队列管理 > 更多 > 测试地址连通性”中进行 Kafka 和 DWS 的连通性测试。

图 3-24 连通性测试

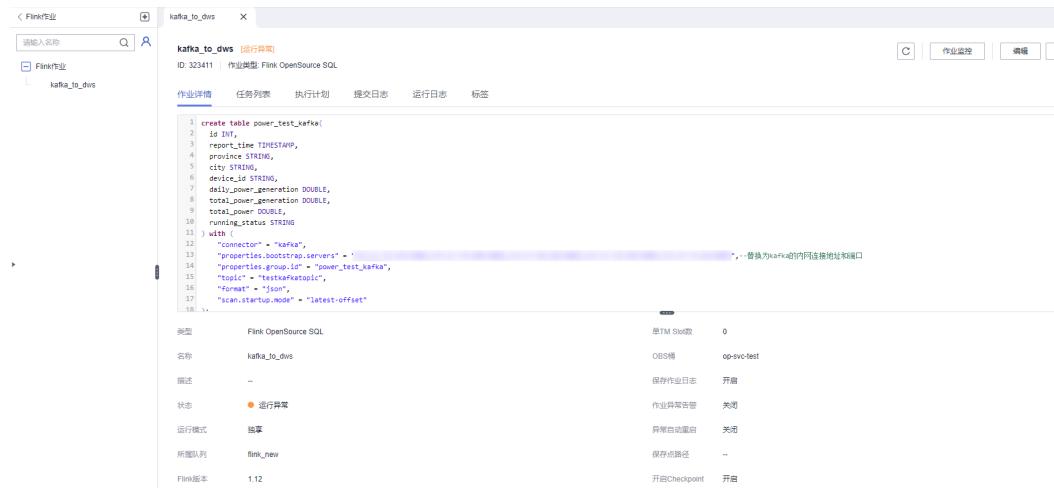


图 3-25 dws 联通测试



**步骤5 选择“作业管理>Flink作业”中创建作业****图 3-26 创建 Flink 作业**

**步骤6 单击“编辑”，在作业详情中编写SQL语句，详情请参考[从Kafka读取数据写入到DWS](#)。**

**图 3-27 配置 Flink 作业**

**步骤7 配置完Flink作业后，启动作业，完成Flink配置。**

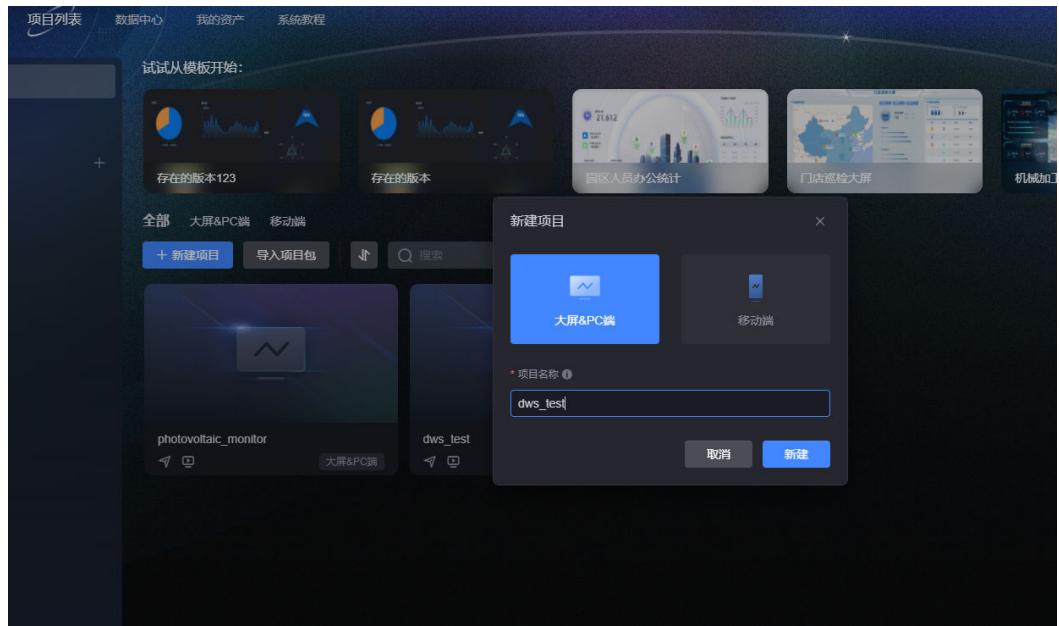
----结束

## 配置 Astro 大屏服务

**步骤1 登录华为云官方网站，访问[Astro大屏服务](#)。**

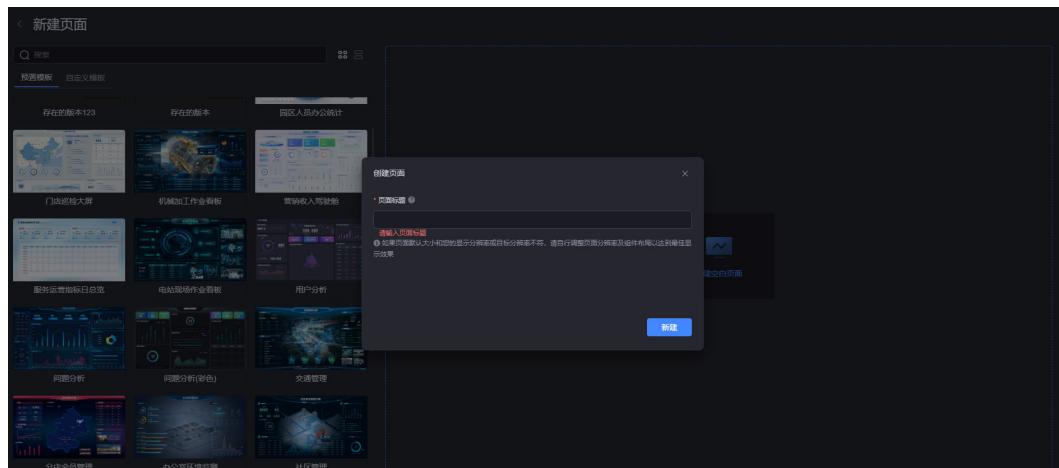
**步骤2 选择进入首页，新建项目。**

图 3-28 新建项目



步骤3 进入新建项目后，点击“新建页面”。点击“新建空白页”完成页面创建。

图 3-29 新建空白页



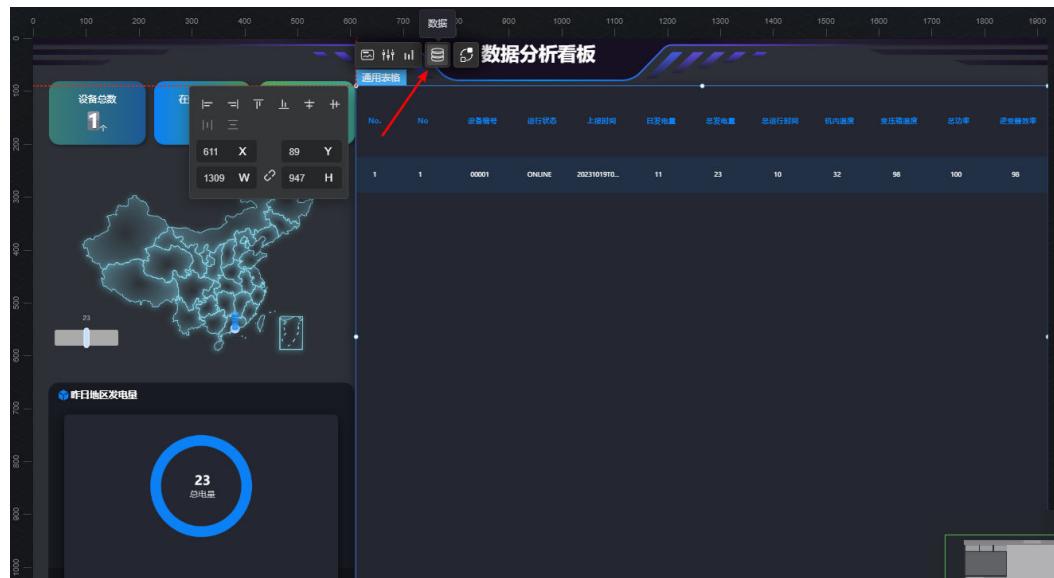
步骤4 可通过拖拉的方式，将组件中的元素拉到页面，从而完成页面设计。详情请参考[开发页面](#)。

图 3-30 最终页面展示



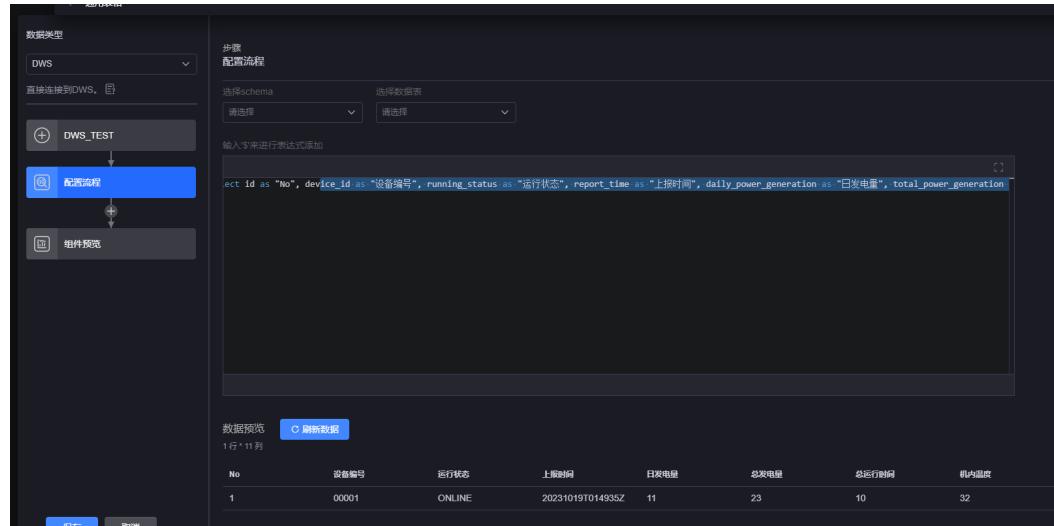
步骤5 设置数据源，选中一个视图组件，如图点击数据按钮。

图 3-31 设置数据源



步骤6 数据类型选择DWS，并[配置DWS数据源](#)，配置流程中通过sql语句查询DWS数据，并在组件预览中将查询出来的数据与组件的配置做映射。详情参考[数据集管理](#)。

图 3-32 配置数据



步骤7 参考上述步骤完成所有组件的配置。

----结束

## 模拟数据上报及结果验证

步骤1 使用MQTT模拟器连接到平台（模拟器使用请参考：[使用MQTT.fx调测](#)）。

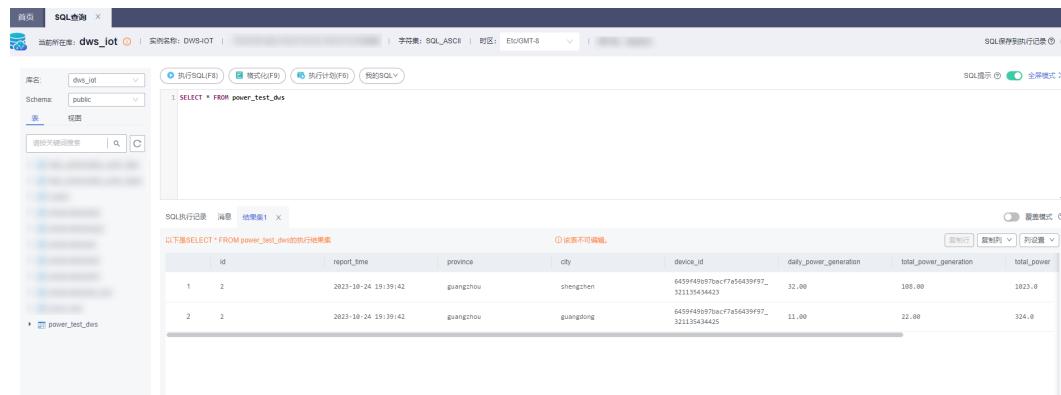
步骤2 使用模拟器进行消息上报，详情请参考：[设备消息上报](#)。

上报内容如下：

```
{
 "id": "2",
 "device_id" : "6459f49b97bacf7a56439f97_321135434425",
 "report_time": "2023-10-24 19:39:42",
 "province": "guangzhou",
 "city": "guangdong",
 "daily_power_generation": 11,
 "total_power_generation": 22,
 "total_power": 324,
 "running_status": "OFFLINE"
}
```

步骤3 前往[DWS控制台](#)，查看是否收到转发的消息。

图 3-33 查看 DWS 消息



步骤4 前往[Astro大屏服务](#)查看监控视图。

图 3-34 监控大屏数据



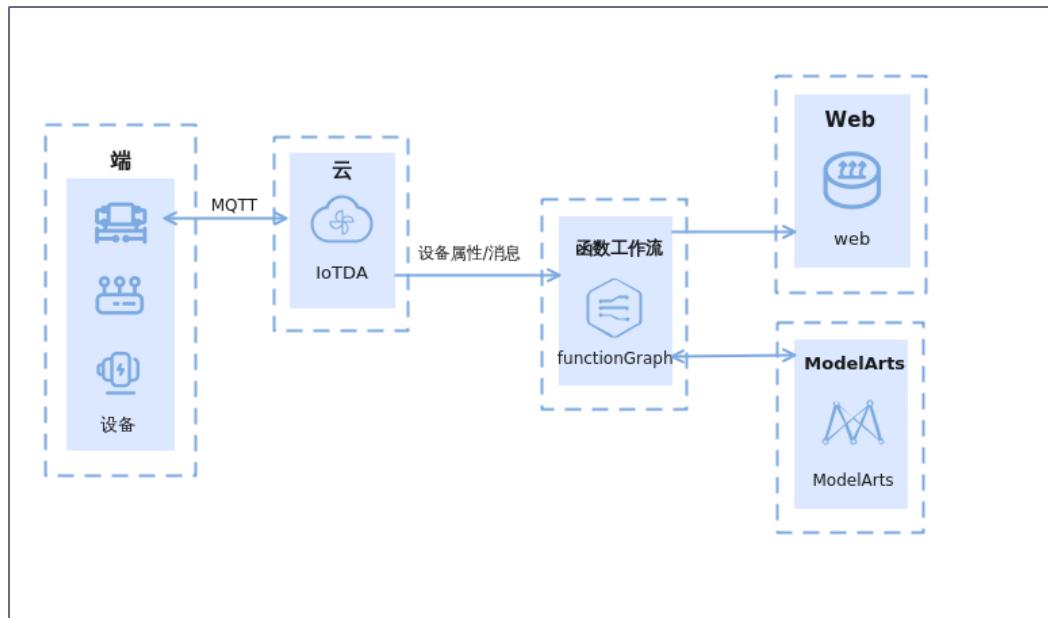
----结束

### 3.3 IoTDA 结合 ModelArts 实现预测分析

#### 场景说明

在物联网解决方案中，针对庞大的数据进行自动学习时，需要对海量数据进行标注、训练，按照传统的方式进行标注、训练不仅耗时耗力，而且对资源消耗也是非常庞大的。华为云物联网平台可以通过规则引擎，将数据转发到华为云其他云服务，可实现将海量数据通过函数工作流（FunctionGraph）进行处理，再将数据流入AI开发平台（ModelArts）进行AI分析，并将分析结果统一转发至HTTP服务器中。

图 3-35 场景说明



在本示例中，我们实现以下场景：

设备上报银行客户特征信息，物联网平台将数据转发至FunctionGraph，由FunctionGraph转发至ModelArts进行AI分析，最终将分析的结果转发至HTTP服务器中。

## 整体流程

1. 创建并发布ModelArts模型。
2. 创建FunctionGraph函数。
3. 构建一个HTTP服务器。
4. 创建MQTT协议产品，并创建设备。
5. 创建流转规则，将数据流转至FunctionGraph。
6. 查看HTTP服务器是否收到AI分析后的消息。

## 前提条件

- 已注册华为官方帐号。未注册可参考[注册华为账户](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击[实名认证](#)完成认证，否则会影响后续云服务的开通。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“免费试用”或单击“价格计算器”购买并开通该服务。
- 已开通FunctionGraph服务。未开通则访问[FunctionGraph服务](#)，单击“立即使用”后开通该服务。
- 已开通ModelArts服务。未开通则访问[AI开发平台](#)，单击“控制台”后进入该服务。
- 自建一个HTTP服务器，并提供POST接口用来接收推送的数据（本示例默认已经提供好相应的服务器与接口，不再展示如何搭建HTTP服务器指导）。

## 配置 ModelArts 模型

- 步骤1** 下载**ModelArts-Lab工程**，在\ModelArts-Lab-master\official\_examples\Using\_ModelArts\_to\_Create\_a\_Bank\_Marketing\_Application\data”目录下获取训练数据文件“train.csv”。该训练数据主要展示银行中的一种常见业务：根据客户特征（年龄、工作类型、婚姻状况、文化程度、是否有房贷和是否有个人贷款），预测客户是否愿意办理定期存款业务。
- 步骤2** 可将训练数据存放在OBS中，供创建数据集使用。进入OBS控制台，选则一个桶，然后单击“上传对象”。若没有桶，可以单击右上角“创建桶”创建一个新的桶。

图 3-36 上传训练数据



**步骤3** 登录华为云官方网站，访问**AI开发平台**，单击“控制台”，进入ModelArts服务。

**步骤4** 选择左侧导航栏“自动学习>前往新版>创建项目”，进入创建预测分析界面。

图 3-37 预测分析

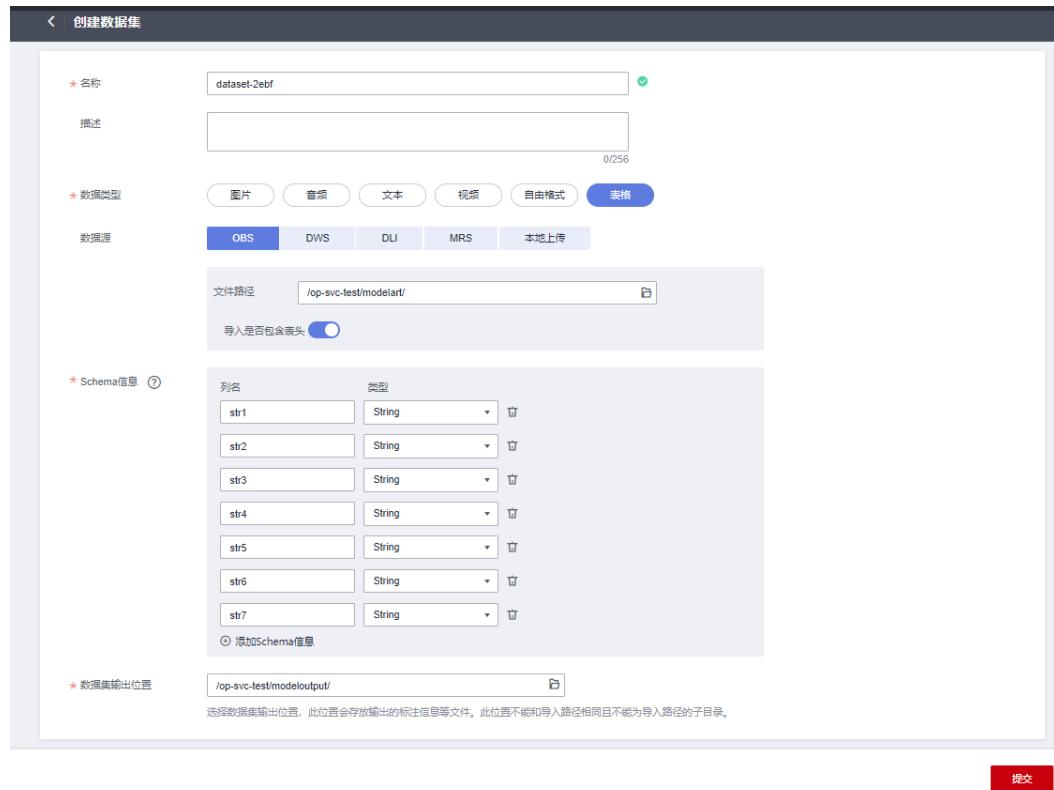


**步骤5** 选择数据集、标签列（数据中预测结果的列，本示例中为str7），若没有数据集，可以单击“创建数据集”进行创建。

图 3-38 创建预测分析

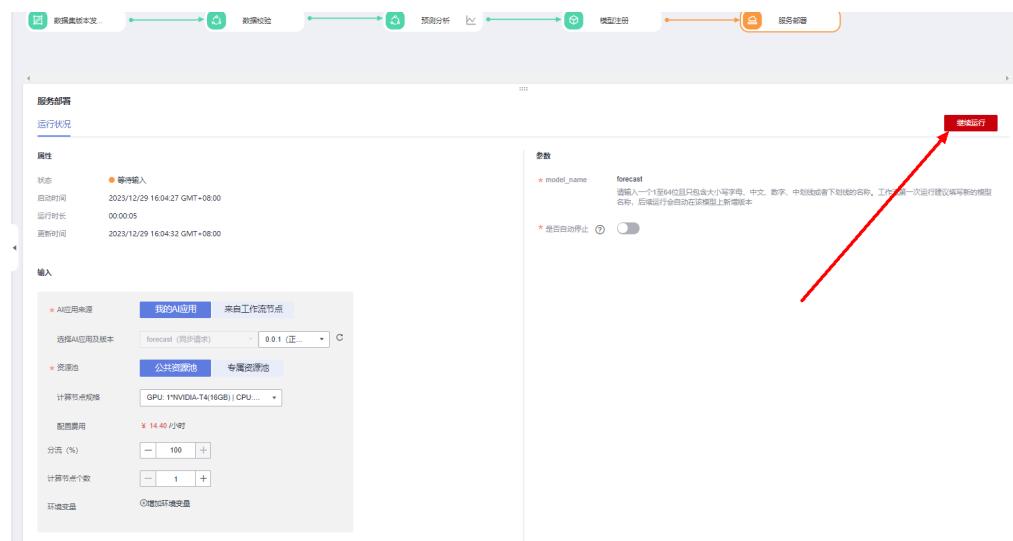


图 3-39 创建数据集



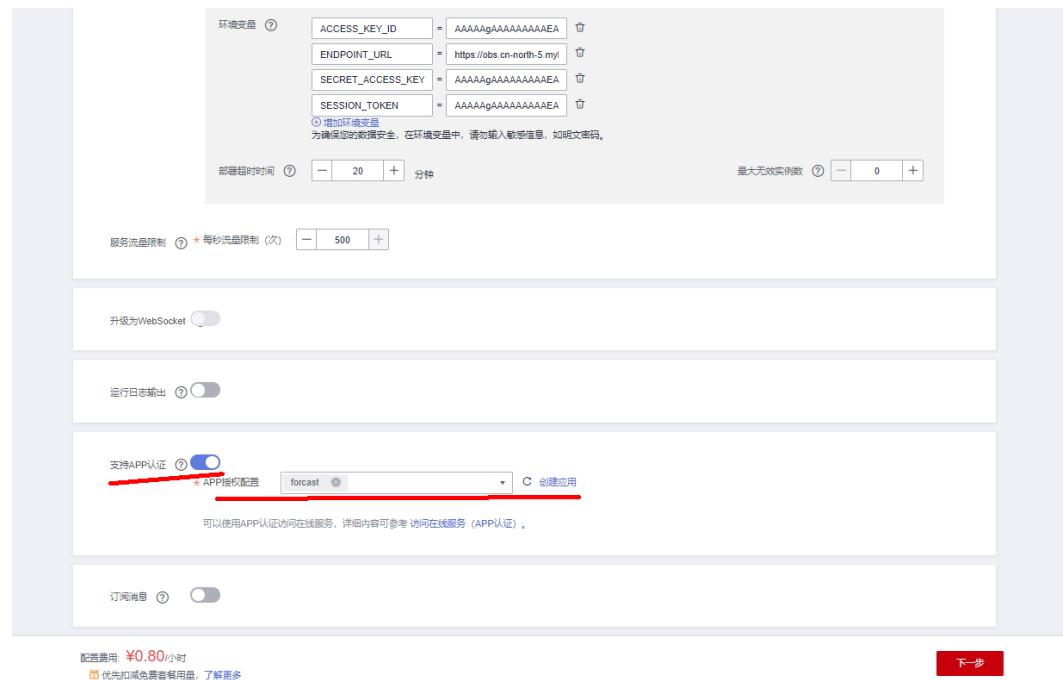
步骤6 当执行到服务部署时，选择资源池、AI应用及版本，单击“继续运行”。

图 3-40 服务部署



步骤7 等部署完成之后，选择左侧导航栏“部署上线 > 在线服务”，进入在线服务页面中选择部署的服务，单击“修改”，进入修改服务页面，打开APP认证进行授权配置，完成后单击“下一步”并提交。

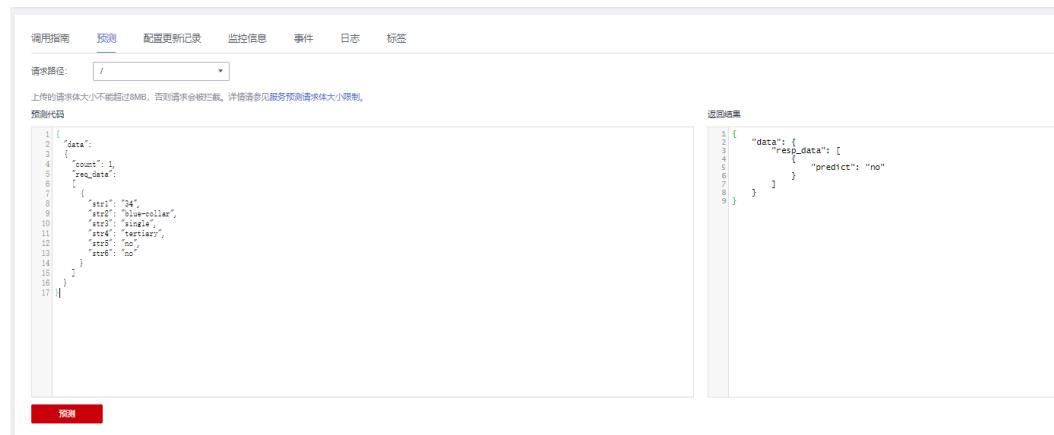
图 3-41 授权



**步骤8** 单击“部署上线>在线服务”，点击进入已部署的服务，选择“预测”，复制以下数据到预测代码中后，单击“预测”后可查看返回结果，结果中的predict为no则表示用户不会办理存款。

```
{
 "data":
 {
 "count": 1,
 "req_data":
 [
 {
 "str1": "34",
 "str2": "blue-collar",
 "str3": "single",
 "str4": "tertiary",
 "str5": "no",
 "str6": "no"
 }
]
 }
}
```

图 3-42 预测



**步骤9** 更多详细关于Modelarts的说明可以参考[ModelArts相关文档](#)。

----结束

## 配置 FunctionGraph 函数

**步骤1** 参考[数据转发至FunctionGraph函数工作流](#)进行函数工作流配置。本示例中由于需要使用ModelArts相关配置参数，可按照如下方式，在代码中添加配置项并访问ModelArts预测接口，body体结构参考**步骤8**。

```
//2.获取ModelArts预测链接. 用来拼装请求URL
String forecastServerAddress = context.getUserData(FORECAST_SERVER_ADDRESS);
log.log("forecastServerAddress: " + forecastServerAddress);
//3.获取ModelArts中的AK/APP_KEY
String ak = context.getUserData(ACCESS_KEY);
//4.获取ModelArts中的SK/APP_SECRET
String sk = context.getUserData(ACCESS_SECRET);

Request request = new Request();
request.setUrl(forecastServerAddress);
request.setMethod(HttpMethodName.POST.name());
request.setAppKey(ak);
request.setAppSecret(sk);
request.addHeader(HttpHeaders.CONTENT_TYPE, ContentType.APPLICATION_JSON.toString());
request.setBody(body);
Signer signer = new Signer();
signer.sign(request);

Map<String, String> headers = request.getHeaders();
HttpPost httpPost = new HttpPost(url);
headers.forEach(httpPost::setHeader);
httpPost.setEntity(new StringEntity(body, ContentType.APPLICATION_JSON));
CloseableHttpResponse response = null;
try {
 response = httpClient.execute(httpPost);
 if (response.getStatusLine().getStatusCode() == HttpStatus.SC_OK) {
 String content = EntityUtils.toString(response.getEntity(), StandardCharsets.UTF_8);
 IoTDAModelArtsDemo.log.log("response content is: " + content);
 return content;
 }
 String errContent = EntityUtils.toString(response.getEntity(), StandardCharsets.UTF_8);
 IoTDAModelArtsDemo.log.log("response err content is: " + errContent);
 return errContent;
} finally {
 if (response != null) {
 response.close();
 }
}
```

步骤2 在函数工作流中，单击“设置>环境变量”，环境变量信息如下。

表 3-6 环境变量说明

| 环境变量                    | 说明                                                                 |
|-------------------------|--------------------------------------------------------------------|
| FORECAST_SERVER_ADDRESS | 访问ModelArts服务，单击“部署上线>在线服务”，进入部署服务后单击“调用指南”，选择支持App认证方式的API接口公网地址。 |
| ACCESS_KEY              | 同上，选择AppKey。                                                       |
| ACCESS_SECRET           | 同上，选择AppSecret。                                                    |
| NA_MOCK_SERVER_ADDRESS  | 将预测结果推送至服务器的地址（本示例不再提供HTTP服务器搭建指导）。                                |

图 3-43 设置环境变量

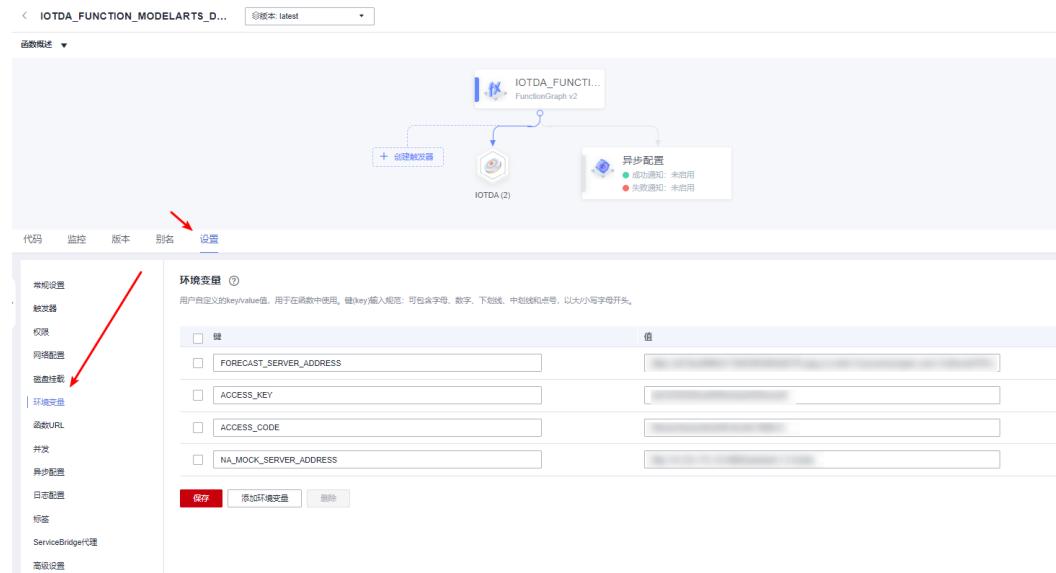
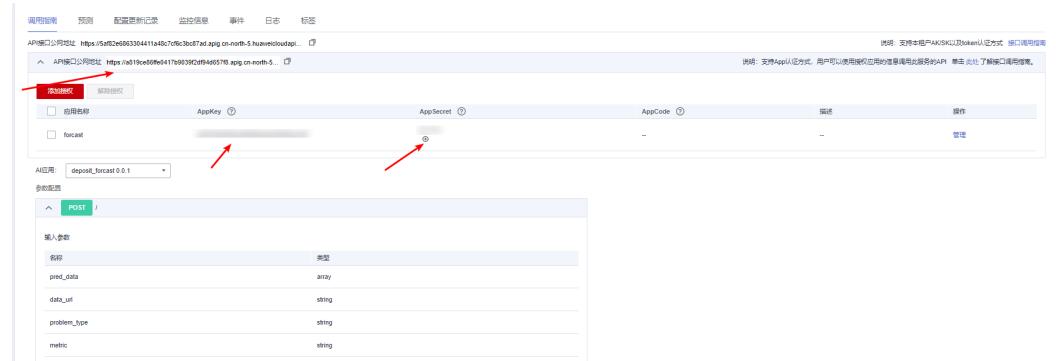


图 3-44 查看预测接口信息

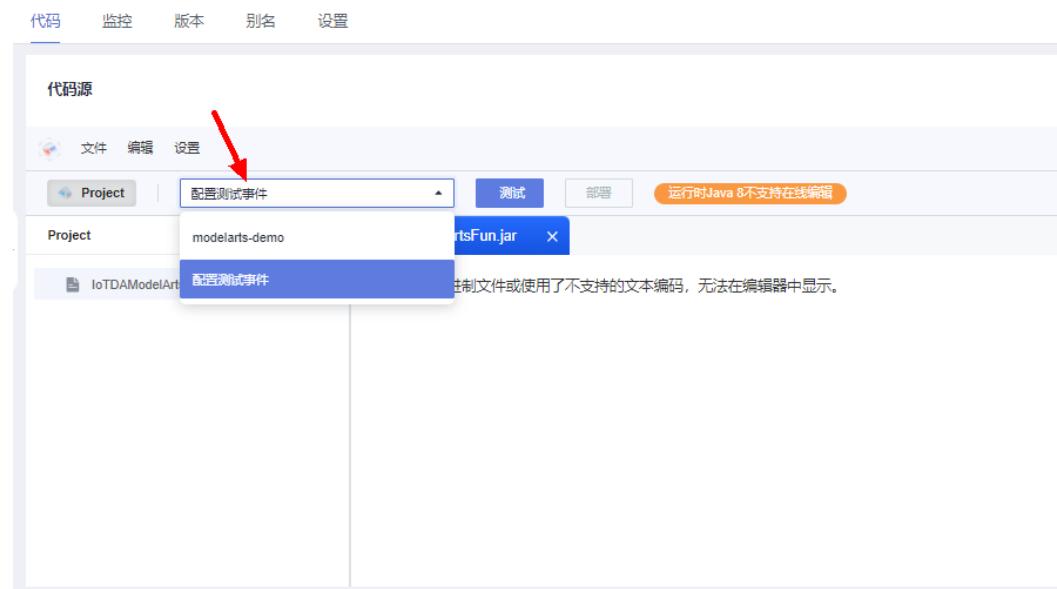


步骤3 单击“代码>配置测试事件>创建新的测试事件>空白模板”。内容示例如下：

```
{
 "resource": "device.message",
```

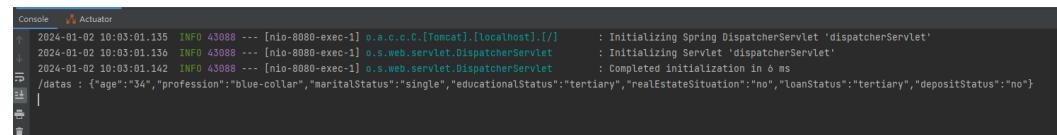
```
 "event": "report",
 "event_time": "20231227T082702Z",
 "event_time_ms": "2023-12-27T08:27:02.944Z",
 "request_id": "1d041aa3-29b8-43d3-aae3-3905de130537",
 "notify_data": {
 "header": {
 "app_id": "dc12bf47e95c4723a00f4a007073fc7e",
 "device_id": "658bdb475d3bc3372c99feb9_12345484121",
 "node_id": "12345484121",
 "product_id": "658bdb475d3bc3372c99feb9",
 "gateway_id": "658bdb475d3bc3372c99feb9_12345484121"
 },
 "body": {
 "topic": "$soc/devices/658bdb475d3bc3372c99feb9_12345484121/sys/messages/up",
 "content": {
 "age": "34",
 "profession": "blue-collar",
 "maritalStatus": "single",
 "educationalStatus": "tertiary",
 "realEstateSituation": "no",
 "loanStatus": "tertiary"
 }
 }
 }
 }
```

图 3-45 配置测试事件



**步骤4** 配置完测试事件后，单击“测试”，执行结果返回success（以实际函数返回结果为准），则表示成功。配置的HTTP服务器则能收到对应的预测结果。

图 3-46 预测结果



----结束

## 创建产品和设备

**步骤1** 访问[设备接入服务](#)，单击“管理控制台”进入设备接入控制台。

**步骤2** 选择左侧导航栏的“产品”，单击“创建产品”，创建一个基于MQTT协议的产品，填写参数后，单击“确定”。

图 3-47 创建产品-MQTT



**步骤3** 导入产品模型，请参考[导入产品模型](#)。

**步骤4** 在该产品下注册设备，请参考[注册单个设备](#)。

### 说明

本文中使用的产品模型和设备仅为示例，您可以使用自己的产品模型和设备进行操作。

----结束

## 数据转发规则配置

**步骤1** 选择左侧导航栏的“规则>数据转发”，单击“创建规则”。

**步骤2** 参考下表参数说明，填写规则内容。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的规则，填写完成后单击“创建规则”。

图 3-48 新建消息上报流转规则-数据转发至 FunctionGraph

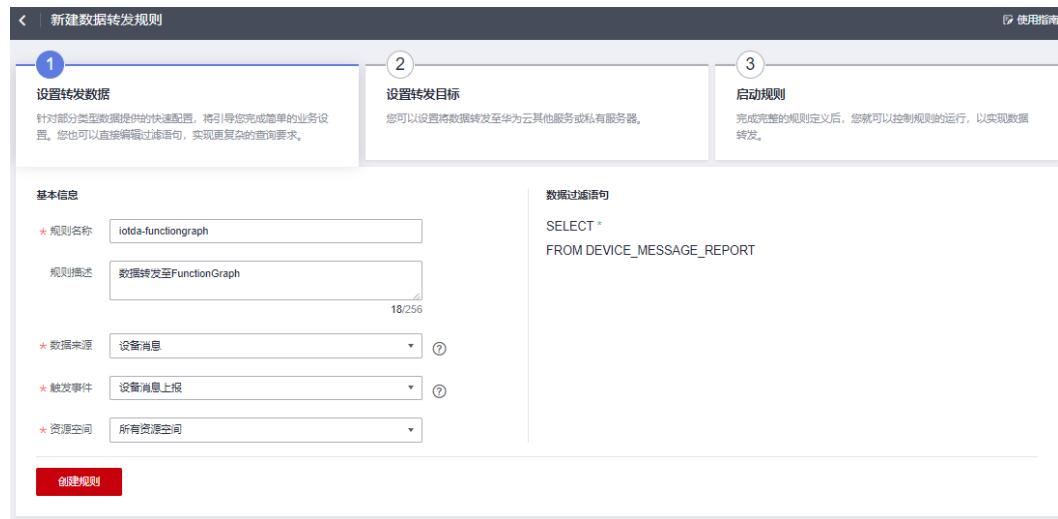


表 3-7 参数说明

| 参数名    | 参数说明                       |
|--------|----------------------------|
| 规则名称   | 自定义，如iotda-functiongraph。  |
| 规则描述   | 自定义，如数据转发至FunctionGraph服务。 |
| 数据来源   | 选择“设备消息”。                  |
| 触发事件   | 自动匹配“设备消息上报”。              |
| 资源空间   | 和上一步创建的产品所在的资源空间保持一致。      |
| 数据过滤语句 | 通过编写SQL来解析和处理上报的JSON数据。    |

步骤3 单击“设置转发目标”页签，单击“添加”，设置转发目标。

图 3-49 新建转发目标-转发至 FunctionGraph



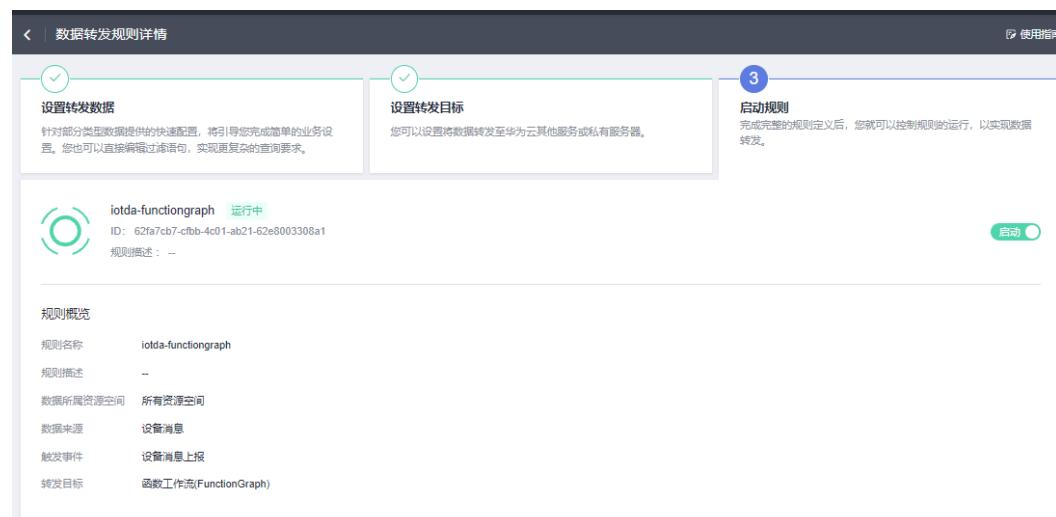
参考下表参数说明，填写转发目标。填写完成后单击“确定”。

表 3-8 参数说明

| 参数名  | 参数说明                     |
|------|--------------------------|
| 转发目标 | 选择“函数工作流（FunctionGraph）” |
| 区域   | 选择“函数工作流”区域。             |
| 目标函数 | 选择已配置的函数工作流。             |

**步骤4** 单击“启动”，激活配置好的数据转发规则。

图 3-50 启动规则-消息上报-转发至 FunctionGraph



----结束

## 模拟数据上报及结果验证

**步骤1** 使用MQTT模拟器连接到平台（模拟器使用请参考：[使用MQTT.fx调测](#)）。

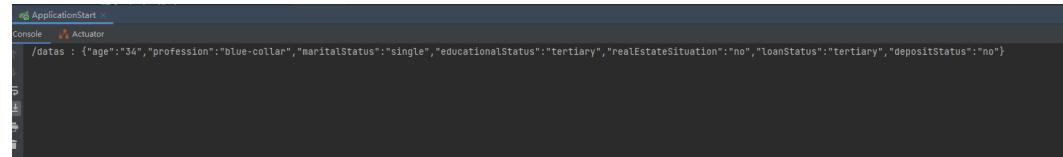
**步骤2** 使用模拟器进行消息上报，详情请参考：[设备消息上报](#)。

上报内容如下：

```
{
 "age": "34",
 "profession": "blue-collar",
 "maritalStatus": "single",
 "educationalStatus": "tertiary",
 "realEstateSituation": "no",
 "loanStatus": "tertiary"
}
```

**步骤3** 查看HTTP服务器是否收到预测结果。

图 3-51 查看消息



----结束

# 4 设备联动

## 4.1 设备触发告警并邮件或短信通知

### 场景说明

很多物联网设备24小时运行，设备管理者并不需要知道设备每一刻的状态，只需要设备在上报某些特定状态时能通知到管理者。

设备接入服务提供规则引擎功能可以满足该诉求，您可以通过设置规则简单实现当设备上报的数据满足某个条件时，物联网平台触发指定动作进行通知。

本示例为设备上报的电池电量小于20%时，物联网平台会上报告警并发送邮件或短信通知给指定的手机号码。

### 配置消息通知服务

在消息通知服务中创建主题并添加订阅，供设备接入服务调用以发送邮件或短信。

1. 登录华为云官方网站，访问[消息通知服务](#)。
2. 单击“立即使用”进入消息通知服务控制台。若您未开通消息通知服务，请先开通服务。
3. 进入“主题管理 > 主题”页面，单击“创建主题”。
4. 输入主题名称，如“Test\_1”，单击“确定”。

图 4-1 创建主题-SMN



5. 进入“主题管理 > 订阅”页面，单击“添加订阅”。
6. 填写订阅信息，填写完成后单击“确定”。

图 4-2 添加订阅-SMN



表 4-1 添加订阅参数填写说明

| 参数名称 | 说明                                                                                                                                          |
|------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 主题名称 | 选择步骤4创建的主题。                                                                                                                                 |
| 协议   | <ul style="list-style-type: none"><li>若需要发送邮件通知，选择“邮件”。</li><li>若需要发送短信通知，选择“短信”。</li></ul>                                                 |
| 订阅终端 | <ul style="list-style-type: none"><li>若协议为“邮件”，填写接收通知邮件的邮箱地址；</li><li>若协议为“短信”，输入接收短信通知的手机号码。</li></ul> <p>需要添加多个终端时，每个终端地址占1行，最多填写10行。</p> |

## 配置设备接入服务

在设备接入服务中创建产品模型、注册设备并设置设备联动规则，实现当设备上报特定数据时触发告警并发送邮件或短信。

1. 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
2. 选择左侧导航栏的“产品”。
3. 单击左侧的“创建产品”，创建一个基于MQTT协议的产品，填写参数后，单击“确认”。  
**注：**本文中使用的产品模型和设备仅为示例，您可以使用自己的产品模型和设备进行操作。

表 4-2 创建产品参数示例

| 基本信息 |                  |
|------|------------------|
| 产品名称 | 自定义，如MQTT_Device |
| 协议类型 | 选择“MQTT”         |
| 数据格式 | 选择“JSON”         |
| 所属行业 | 根据实际情况进行填写。      |
| 设备类型 |                  |

4. 单击[文件](#)，获取产品模型文件样例。
5. 找到新增的产品，单击“产品”进入产品详情界面。
6. 在模型定义页面，单击“上传模型文件”，在弹出的页面中加载产品模型文件，然后单击“确认”。

图 4-3 产品-上传模型文件



7. 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。填写完成后单击“确定”，请注意保存注册成功返回的“设备ID”和“设备密钥”。

图 4-4 单设备注册-MQTT

The screenshot shows the 'Single Device Registration' interface for MQTT devices. It includes fields for:

- \* 所属资源空间 (Resource Space): A dropdown menu showing a blurred option.
- \* 所属产品 (Product): A dropdown menu set to 'Test\_1'. A note below it states: 'MQTT类型的设备已默认订阅平台预置topic, 查看已订阅topic列表' (MQTT type devices have default subscribed to platform-predefined topics, view the subscribed topic list).
- \* 设备标识码 (Device Identifier): A text input field containing 'Test\_1'.
- 设备名称 (Device Name): A text input field containing a blurred value.
- 设备ID (Device ID): A text input field containing a blurred value.
- 设备描述 (Device Description): A text area with a character limit of 0/2,048. The text area is empty.
- 设备认证类型 (Device Authentication Type): A radio button group with two options: '密钥' (Key) (selected) and 'X.509证书' (X.509 Certificate).
- 密钥 (Key): A text input field for entering the key.
- 确认密钥 (Confirm Key): A text input field for confirming the key.

At the bottom are 'Confirm' and 'Cancel' buttons.

表 4-3 注册设备参数填写说明

| 参数名称 | 说明            |
|------|---------------|
| 所属产品 | 选择在步骤3中创建的产品。 |

| 参数名称   | 说明                                                               |
|--------|------------------------------------------------------------------|
| 设备标识码  | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                             |
| 设备认证类型 | 选择“密钥”。                                                          |
| 密钥     | 设备密钥，可自定义，不填写物联网平台会自动生成。                                         |

8. 选择左侧导航栏的“规则>设备联动”，单击右上角的“创建规则”按钮。（创建规则前需选择所属资源空间。）
9. 参考下表参数说明，填写规则内容。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的规则，填写完成后单击“创建规则”。

图 4-5 新建联动规则-BatteryProperty

The screenshot shows the 'Create Rule' configuration page. It includes:

- 基本信息**: Fields for '所属资源空间' (selected: 'Test\_1'), '规则名称' (selected: 'Test\_1'), '激活规则' (checked), '规则描述' (empty), and '生效时间' (selected: '一直生效').
- 触发条件**: A complex query builder for triggering rules based on device status (e.g., '设备属性' 'Test\_1' '指派设备' 'streetlight' '重新连接' 'Battery' 'batteryLevel' < 20).
- 执行动作**: Action selection for '上报告警' (selected) and '发送通知' (selected: 'Test\_1' to 'Test\_1').

表 4-4 创建联动规则参数填写说明

| 参数名称 | 说明                                          |
|------|---------------------------------------------|
| 规则名称 | 自定义，例如“Battery_Low_Mail”或“Battery_Low_SMS”。 |
| 激活规则 | 勾选“激活规则”                                    |
| 生效时间 | 选择“一直生效”。                                   |
| 描述   | 自定义对该规则的描述，如“设备上报电量低于20%时上报告警并发送通知”。        |

| 参数名称 | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 触发条件 | <ol style="list-style-type: none"><li>单击“添加条件”；</li><li>选择“设备属性触发”；</li><li>选择<b>3</b>中添加的产品，选择“指定设备”，再选择<b>7</b>中添加的设备；</li><li>“服务类型”选择“Battery”，“属性”选择“batteryLevel”，“操作”选择“&lt;”，“值”填写“20”，单击“触发机制”，“触发策略”选择“重复抑制”，“数据时效”填写“3600”，然后单击“确认”。</li></ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 执行动作 | <p>首先添加告警：</p> <ol style="list-style-type: none"><li>单击“添加动作”；</li><li>动作类型选择“上报告警”；</li><li>“告警等级”选择“次要”，“告警隔离级别”选择“设备”，“告警名称”输入“设备电量低”，“告警内容”输入“设备电量低于20%，请及时确认并更换电池。”，然后单击“确认”。</li></ol> <p>再添加通知：</p> <ol style="list-style-type: none"><li>单击“添加动作”。</li><li>动作类型选择“发送通知”。</li><li>选择您开通的SMN服务区域，如“cn-north-4”。首次创建对接到SMN服务的规则时，平台会根据对接的云服务和区域弹出对应的云服务访问授权窗口，请根据界面提示，配置云服务访问授权。（可登录SMN服务控制台后在左上角查看）。</li><li>主题名称选择<b>配置消息通知服务</b>时创建的主题：<ul style="list-style-type: none"><li>若主题对应的订阅协议是邮件，消息标题填写为邮件标题，如 “[Huawei IoT Platform] Low Battery Warning”；消息内容填写“You have a device with less than 20% charge, please log in to the Huawei IoT Platform for details.”</li><li>若主题对应的订阅协议是短信，消息标题无需填写，消息内容填写 “[Huawei IoT Platform] You have a device with less than 20% charge, please log in to the Huawei IoT Platform for details.”</li></ul></li></ol> |

## 验证操作

- 您可以使用配置设备接入服务时注册的真实设备接入平台，上报“batteryLevel”小于20的数据。
- 您也可以使用模拟器模拟设备上报“batteryLevel”小于20的数据，操作方法请参考[在线开发MQTT协议的智慧路灯](#)。
- 您还可以使用虚拟设备进行在线调试，上报“batteryLevel”小于20的数据。

期望结果：

- 在左侧导航栏选择“监控运维 > 当前告警”，单击“前往AOM”，跳转至应用运维管理服务，有一条名称为“设备电量低”的一般告警。
- 若订阅了邮件通知，接受订阅的邮箱会收到一封通知设备电量低的邮件。
- 若订阅了短信通知，接受订阅的手机会收到一条通知设备电量低的短信。

## 4.2 温度过高时自动关闭设备

### 场景说明

物联网平台支持设备数据上报和设备命令下发，但若要将两者联动起来，一般需要由应用服务器实现对应逻辑。

设备接入服务提供规则引擎功能，平台上简单几步操作即可实现数据上报特定数据时平台自动下发指定命令，减少应用服务器开发工作量。

本示例为设备自带的温度传感器上报的温度大于80度时，平台自动下发命令关闭设备。

### 配置设备接入服务

在设备接入服务中创建产品模型、注册设备并设置设备联动规则，实现当设备上报特定数据时平台自动下发命令。

- 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
  - 选择左侧导航栏的“产品”。
- 注：**本文中使用的产品模型和设备仅为示例，您可以使用自己的产品模型和设备进行操作。
- 单击右上角的“创建产品”，创建一个基于MQTT协议的产品，填写参数后，单击“确认”。

表 4-5 创建产品参数示例

| 基本信息 |                  |
|------|------------------|
| 产品名称 | 自定义，如MQTT_Device |
| 协议类型 | 选择“MQTT”         |
| 数据格式 | 选择“JSON”         |
| 厂商名称 | 自定义              |
| 所属行业 | 根据实际情况进行填写。      |
| 设备类型 |                  |

- 单击[Profile\\_tempSensor.zip](#)，获取产品模型文件样例。
- 在模型定义页面，单击“上传模型文件”，在弹出的页面中加载产品模型文件，然后单击“确认”。

图 4-6 产品-上传模型文件



6. 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。

图 4-7 单设备注册-MQTT

The screenshot shows the 'Single Device Registration' (MQTT) configuration page. It includes fields for: '所属资源空间' (Selected: [redacted]), '所属产品' (Selected: Test\_1), 'MQTT类型的设备已默认订阅平台预置topic, 查看已订阅topic列表' (MQTT device has default subscribed to platform-predefined topic, View subscribed topic list), '设备标识码' (Selected: Test\_1), '设备名称' ([redacted]), '设备ID' ([redacted]), '设备描述' (0/2,048 characters), '设备认证类型' (Selected: 密钥), '密钥' (Input field), '确认密钥' (Input field), and a note that both fields must be identical. At the bottom are '确定' (Confirm) and '取消' (Cancel) buttons.

表 4-6 注册设备参数填写说明

| 参数名称  | 说明                                                               |
|-------|------------------------------------------------------------------|
| 所属产品  | 选择在步骤3中创建的产品。                                                    |
| 设备标识码 | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |

| 参数名称   | 说明                       |
|--------|--------------------------|
| 设备名称   | 自定义。                     |
| 设备认证类型 | 选择“密钥”。                  |
| 密钥     | 设备密钥，可自定义，不填写物联网平台会自动生成。 |

填写完成后单击“确定”，请注意保存注册成功返回的“设备ID”和“设备密钥”。

7. 选择左侧导航栏的“规则>设备联动”，单击右上角的“创建规则”按钮。（创建规则前需选择所属资源空间。）
8. 参考下表参数说明，填写规则内容。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的规则，填写完成后单击“创建规则”。

图 4-8 新建联动规则-TemperatureProperty

表 4-7 创建联动规则参数填写说明

| 参数名称 | 说明                           |
|------|------------------------------|
| 规则名称 | 创建的规则名称，例如“Overheated”。      |
| 立即触发 | 勾选“激活规则”                     |
| 生效时间 | 选择“一直生效”。                    |
| 描述   | 对该规则的描述，如“设备温度高于80度时自动关闭设备”。 |

| 参数名称 | 说明                                                                                                                                                                           |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 触发条件 | 1. 单击“添加条件”；<br>2. 选择“设备属性触发”；<br>3. 选择步骤3中添加的产品，选择指定设备，再选择步骤6中添加的设备；<br>4. 服务选择“tempSensor”，属性选择“temperature”，操作选择“>”，值填写“80”，单击“触发机制”，“触发策略”选择“重复抑制”，数据时效填写“300”，然后单击“确定”。 |
| 执行动作 | 1. 单击“添加动作”；<br>2. “动作类型”选择“下发命令”，下发设备选择步骤6创建的设备；<br>3. 服务选择“deviceSwitch”，命令选择“ON/OFF”，单击“参数配置”，“power”的值选择“OFF”，然后单击“确定”。                                                  |

## 验证操作

- 您可以使用配置设备接入服务时注册的真实设备接入平台，上报“temperature”大于80的数据。
- 您也可以使用模拟器模拟设备订阅Topic：“\$oc/devices/{device\_id}/sys/properties/report”（其中{device\_id}需要替换成实际设备ID）后，上报“temperature”大于80的数据，操作方法请参考[在线开发MQTT协议的智慧路灯](#)。
- 您还可以使用虚拟设备进行在线调试，上报“temperature”大于80的数据。

期望结果：

- 若您使用真实设备上报数据，设备会收到一条“power”取值为“OFF”的“ON/OFF”命令。
- 若您使用模拟器上报数据，可在“Subscribe”页签查看到平台下发的“power”取值为“OFF”的“ON/OFF”命令。

## 4.3 煤气浓度过高自动打开推窗器

### 场景说明

煤气安全关系到千家万户的生命财产安全。本示例基于华为物联网平台提供的设备接入服务，通过触发设备联动规则，自动打开推窗器。具体场景是煤气监测器将煤气浓度数值上报到物联网平台，当煤气浓度超标时，会触发设置好的设备联动规则，物联网平台会下发打开窗户的命令到无线推窗器，无线推窗器收到命令后，自动把窗户打开。

### 创建煤气监测产品

步骤1 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。

**步骤2** 选择左侧导航栏的“产品”。

**步骤3** 单击左侧的“创建产品”，创建煤气浓度监测产品，填写参数后，单击“确定”。

**表 4-8** 创建产品参数示例

| 基本信息 |                 |
|------|-----------------|
| 产品名称 | 自定义，如gasdevice。 |
| 协议类型 | 选择“MQTT”        |
| 数据格式 | 选择“JSON”        |
| 所属行业 | 自定义             |
| 设备类型 |                 |

**步骤4** 在模型定义页面，单击“自定义模型”，参考下表，完成产品模型配置。gaslevel服务用于监控煤气浓度，windowswitch服务用于执行开关窗命令。

**表 4-9** 煤气浓度监测产品模型参数示例

| 服务ID         | 参数类型 | 添加参数说明内容                                                 |
|--------------|------|----------------------------------------------------------|
| gaslevel     | 属性   | 属性名称：gaslevel<br>数据类型：int<br>访问权限：可读<br>取值范围：0~100       |
| windowswitch | 命令   | 命令名称：switch<br>下发参数名称：switch<br>数据类型：enum<br>枚举值：on, off |

----结束

## 注册设备

**步骤1** 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。注册煤气浓度监测设备，并记录设备ID和密钥。

图 4-9 单设备注册-gasdevice

单设备注册

所属资源空间 (gasdevice)

所属产品 (gasdevice)  
MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

设备标识码 (gasdevice)

设备ID (641d1a65c570f41c39fd44de\_gasdevice)

设备名称 (gasdevice)

设备描述 (0/2,048)

设备认证类型 (密钥 X.509证书) 密钥

密钥 (输入框)

确认密钥 (输入框)

确定 取消

表 4-10 注册设备参数填写说明

| 参数名称   | 说明                                                                   |
|--------|----------------------------------------------------------------------|
| 所属产品   | 选择在创建煤气监测产品步骤3中创建的产品。                                                |
| 设备标识码  | 即node_id, 填写为设备的IMEI、MAC地址或Serial No; 若没有真实设备, 填写自定义字符串, 由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                                 |
| 设备认证类型 | 选择“密钥”。                                                              |
| 密钥     | 设备密钥, 可自定义, 不填写物联网平台会自动生成。                                           |

**步骤2** 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。注册推窗器设备，并记录设备ID和密钥。

图 4-10 单设备注册-windowswitch

单设备注册

\* 所属资源空间 ( )

\* 所属产品 (gasdevice)  
MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

\* 设备标识码 (windowswitch)

设备ID (641d1a65c570f41c39fd44de\_windowswitch)

设备名称 (windowswitch)

设备描述 (0/2,048)

设备认证类型 ( )

密钥 (selected) X.509证书

密钥 ( )

确认密钥 ( )

确定 取消

表 4-11 注册设备参数填写说明

| 参数名称   | 说明                                                                   |
|--------|----------------------------------------------------------------------|
| 所属产品   | 选择在步骤3中创建的产品。                                                        |
| 设备标识码  | 即node_id, 填写为设备的IMEI、MAC地址或Serial No; 若没有真实设备, 填写自定义字符串, 由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                                 |
| 设备认证类型 | 选择“密钥”。                                                              |
| 密钥     | 设备密钥, 可自定义, 不填写物联网平台会自动生成。                                           |

----结束

## 配置设备联动规则

**步骤1** 选择左侧导航栏的“规则>设备联动”，单击右上角的“创建规则”按钮。（创建规则前需选择所属资源空间。）

**步骤2** 参考下表参数说明，填写规则内容。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的规则，填写完成后单击“创建规则”。

图 4-11 新建联动规则-GasLevel

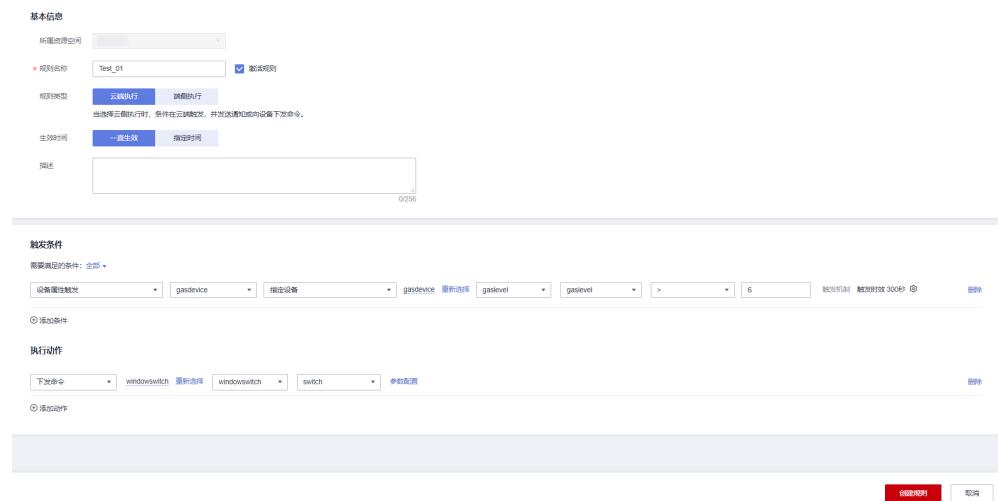


表 4-12 创建联动规则参数填写说明

| 参数名称 | 说明                                                                                                                                                                                                                                     |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 规则名称 | 创建的规则名称，例如“windowswitch”。                                                                                                                                                                                                              |
| 立即触发 | 勾选“激活规则”                                                                                                                                                                                                                               |
| 生效时间 | 选择“一直生效”。                                                                                                                                                                                                                              |
| 描述   | 对该规则的描述，如“煤气浓度高于6时自动打开推窗设备”。                                                                                                                                                                                                           |
| 触发条件 | <ol style="list-style-type: none"><li>单击“添加条件”；</li><li>选择“设备属性触发”；</li><li>选择3中添加的产品，选择“指定设备”后选择1中添加的设备；</li><li>“服务类型”选择“gaslevel”，“属性名字”选择“gaslevel”，“操作”选择“&gt;”，“值”填写“6”，单击“触发机制”，“触发策略”选择“重复抑制”，数据时效”填写“300”，然后单击“确定”。</li></ol> |
| 执行动作 | <ol style="list-style-type: none"><li>单击“添加动作”；</li><li>“动作类型”选择“下发命令”，下发设备选择步骤2创建的设备；</li><li>“服务类型”选择“windowswitch”，“命令名字”选择“on_off”，单击“参数配置”，“switch”的值选择“on”，然后单击“确定”。</li></ol>                                                   |

----结束

## 验证操作

### 方法一：

您可以使用MQTT.fx模拟设备验证。

1. 使用MQTT.fx分别模拟煤气探测器设备和推窗器，并连接到物联网平台。操作方法请参考[在线开发MQTT协议的智慧路灯](#)。
2. 打开模拟推窗器的MQTT.fx，订阅平台下发的命令。
  - a. 选择“Subscribe”页签。
  - b. 输入订阅平台下发命令的Topic:=\$oc/devices/{device\_id}/sys/commands/#（其中与步骤2获取的设备ID保持一致。）。
  - c. 单击“Subscribe”按钮，完成平台命令下发订阅。

图 4-12 创建 MQTT 订阅



3. 切换到模拟煤气探测的MQTT.fx，上报属性。
  - a. 选择“Publish”页签。
  - b. 输入属性上报的Topic：“\$oc/devices/{device\_id}/sys/properties/report（其中{device\_id}与步骤1获取的设备ID保持一致。）。
  - c. 上报1条“gaslevel”大于6的属性。

样例：

```
{
 "services": [
 {"service_id": "gaslevel",
 "properties": {
 "gaslevel": 45
 }
 }
]
}
```

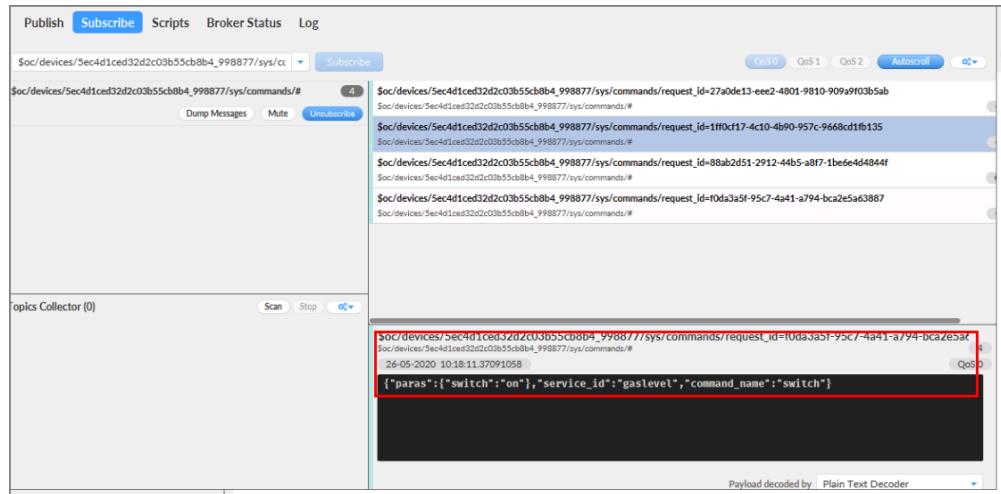
- d. 单击“Publish”按钮上报属性值。

图 4-13 MQTT 属性上报



4. 切换到模拟推窗器的MQTT.fx，选择“Subscribe”页签。可以看到收到平台下发的“switch”取值为“on”的命令。

图 4-14 查看下发命令



### 方法二：

您可以使用配置设备接入服务时注册的真实设备接入平台，上报“gaslevel”大于6的数据。设备会收到一条“switch”取值为“on”的命令，自动推开窗户。

## 4.4 监测设备状态变化情况并发送通知

### 场景说明

在物联网场景下，部分设备具备重要的应用场景，比如物联网网关等，设备管理者需要感知这些设备的上下线情况。

设备接入服务提供规则引擎功能可以满足该诉求，您可以通过简单的操作实现当设备状态满足某个条件时，物联网平台触发指定动作进行通知。

### 示例说明：

某企业的网关产品下有一批网关设备，单个网关设备下挂载了约400个子设备，用户需要实时关注这批网关设备的状态，确保子设备正常上报数据，同时由于网关设备和物联网平台通过4G网络建立连接，存在由于网络抖动导致频繁告警的问题，因此用户认为短暂的离线后上线属于正常场景，不希望感知这种场景。

通过以下示例可以实现监控该网关产品下所有网关设备在离线持续时间达到5分钟后通过物联网平台上报告警，在设备上线持续时间达到1分钟后恢复该告警，同时发送邮件或短信通知给指定的手机号码。

### 对接流程

1. **配置设备接入服务**：创建物联网产品和设备，创建联动规则，实现设备上下线满足特定条件后发送告警到应用运维管理服务。
2. **配置消息通知服务**：创建短信或邮件订阅。
3. **配置应用运维管理服务**：创建告警规则，处理由设备接入服务上报上来的告警，并通过消息通知服务发送短信或邮件通知。

## 配置设备接入服务

在设备接入服务中创建产品模型、注册设备并设置设备联动规则，实现当设备离线达到5分钟后上报告警到应用运维管理平台，设备上线1分钟后恢复该告警。

### 说明

本文中使用的产品模型和设备仅为示例，您可以使用自己的产品模型和设备进行操作。

- 步骤1** 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 步骤2** 选择左侧导航栏的“产品”，单击页面左侧的“创建产品”，创建一个基于MQTT协议的产品，填写参数后，单击“确认”，完成产品的创建。

**表 4-13 创建产品参数示例**

| 基本信息 |                   |
|------|-------------------|
| 产品名称 | 自定义，如MQTT_Device。 |
| 协议类型 | 选择“MQTT”。         |
| 数据格式 | 选择“JSON”。         |
| 所属行业 | 根据实际情况进行填写。       |
| 设备类型 |                   |

- 步骤3** 单击[文件](#)，获取产品模型文件样例。
- 步骤4** 找到新增的产品，单击“产品”进入产品详情界面。
- 步骤5** 在产品详情“模型定义”页面，单击“上传模型文件”，在弹出的页面中加载产品模型文件，然后单击“确定”。

**图 4-15 产品-上传模型文件**



- 步骤6** 选择左侧导航栏的“设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。填写完成后单击“确定”，请注意保存注册成功返回的“设备ID”和“设备密钥”。

图 4-16 单设备注册-gateway

单设备注册

所属资源空间 (必填)

所属产品 (必填) MQTT\_Device  
MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

设备标识码 (必填) gateway\_1

设备ID (必填) 6417cb7cc570f41c39fcc27b\_gateway\_1

设备名称 gateway\_1

设备描述   
0/2,048

设备认证类型 (必填) 密钥 X.509证书

密钥

确认密钥

确定 取消

表 4-14 注册设备参数填写说明

| 参数名称   | 说明                                                               |
|--------|------------------------------------------------------------------|
| 所属产品   | 选择在步骤4中创建的产品。                                                    |
| 设备标识码  | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                             |
| 设备认证类型 | 选择“密钥”。                                                          |
| 密钥     | 设备密钥，可自定义，不填写物联网平台会自动生成。                                         |

**步骤7** 选择左侧导航栏的“规则>设备联动”，单击右上角的“创建规则”按钮。（创建规则前需选择所属资源空间）

**步骤8** 参考下表参数说明，填写规则内容，创建网关离线上报告警规则。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的告警规则，填写完成后单击“创建规则”。

表 4-15 创建联动规则参数填写说明

| 参数名称 | 步骤说明                                                                                                                          |
|------|-------------------------------------------------------------------------------------------------------------------------------|
| 规则名称 | 自定义，例如网关离线规则“Gateway_Offline_Monitor”。                                                                                        |
| 激活规则 | 勾选“激活规则”。                                                                                                                     |
| 生效时间 | 选择“一直生效”。                                                                                                                     |
| 描述   | 对该规则的描述，如“网关设备离线持续时长达到5分钟上报告警。”。                                                                                              |
| 触发条件 | 1. 单击“添加条件”；<br>2. 选择“设备状态触发”；<br>3. 选择步骤4中添加的产品“MQTT_Device”，选择“全部设备”，触发状态选择“下线”；<br>4. “状态持续时长”选择“5分钟”。                     |
| 执行动作 | 1. 单击“添加动作”；<br>2. 动作类型选择“上报告警”；<br>3. “告警等级”选择“重要”，告警隔离级别选择“设备”维度隔离，“告警名称”输入“网关状态变更”，“告警内容”输入“网关设备离线，请及时确认并安排人员进行维护”并单击“确定”。 |

图 4-17 新建联动规则-GatewayOffline

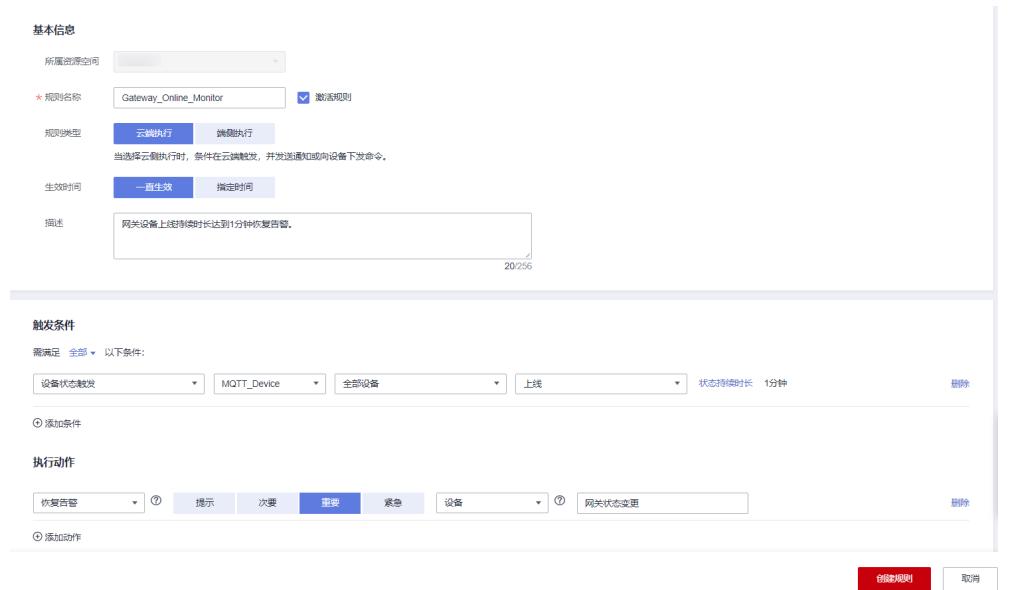
The screenshot shows the 'Create Rule' interface for 'GatewayOffline'. It is a multi-step configuration form:

- 基本信息 (Basic Information):** Includes fields for '所属资源空间' (Resource Space), '规则名称' (Rule Name) set to 'Gateway\_Offline\_Monitor', '激活规则' (Enable Rule) checked, '规则类型' (Rule Type) set to '云端执行' (Cloud Execution), '生效时间' (Effective Time) set to '一直生效' (Always Effective), and a '描述' (Description) field containing '网关设备离线持续时长达到5分钟上报告警'.
- 触发条件 (Trigger Conditions):** Shows a condition setup where '设备状态触发' (Device Status Trigger) is selected. Under '需满足 全部 以下条件:' (Must meet all of the following conditions), there are dropdowns for 'MQTT\_Device' and '全部设备' (All Devices), and a '下线' (Offline) status. Other options like '状态持续时长' (Duration) and '5分钟' (5 minutes) are also visible.
- 执行动作 (Execution Actions):** Shows an action setup where '上报告警' (Report Alarm) is selected. Under '告警等级' (Alarm Level), '重要' (Important) is selected. Other settings include '设备' (Device), '网关状态变更' (Gateway Status Change), and a '编辑告警内容' (Edit Alarm Content) button.

At the bottom right of the interface are '创建规则' (Create Rule) and '取消' (Cancel) buttons.

步骤9 创建网关上线恢复告警规则。

图 4-18 新建联动规则-GatewayOnline



### 须知

- 告警名称、级别和维度组合起来共同标识一条AOM（应用运维管理）告警，恢复的告警和上报的告警需要保持同一告警属性，否则恢复告警失败。
- 设备状态监控有流控限制，监控设备量过多会触发流控导致设备离线告警无法上报的情况，详细请参考[使用限制](#)。

### ----结束

## 配置消息通知服务

在消息通知服务中创建主题并添加订阅，供应用运维管理服务调用以发送邮件或短信。

- 步骤1 登录华为云官方网站，访问[消息通知服务](#)。
- 步骤2 单击“立即使用”进入消息通知服务控制台。若您未开通消息通知服务，请先开通服务。
- 步骤3 进入“主题管理 > 主题”页面，单击“创建主题”。
- 步骤4 输入主题名称，如“Test\_1”，单击“确定”。

图 4-19 创建主题-SMN



**步骤5** 进入“主题管理 > 订阅”页面，单击“添加订阅”。

**步骤6** 填写订阅信息。填写完成后单击“确定”。

图 4-20 添加订阅-SMN



表 4-16 添加订阅参数填写说明

| 参数名称 | 说明                                                                                                                                          |
|------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 主题名称 | 选择 <b>步骤4</b> 创建的主题。                                                                                                                        |
| 协议   | <ul style="list-style-type: none"><li>若需要发送邮件通知，选择“邮件”。</li><li>若需要发送短信通知，选择“短信”。</li></ul>                                                 |
| 订阅终端 | <ul style="list-style-type: none"><li>若协议为“邮件”，填写接收通知邮件的邮箱地址；</li><li>若协议为“短信”，输入接收短信通知的手机号码。</li></ul> <p>需要添加多个终端时，每个终端地址占1行，最多填写10行。</p> |

----结束

## 配置应用运维管理服务

在应用运维管理服务中创建告警规则和告警行动规则，当设备接入服务满足条件上报告警后，对上报告警进行处理并发送邮件或短信。

- 步骤1** 登录华为云官方网站，访问[应用运维管理服务](#)。
- 步骤2** 单击“立即使用”进入应用运维管理服务控制台。若您未开通应用运维管理服务，请先开通服务。
- 步骤3** 进入“告警 > 告警行动规则”页面，单击“创建告警行动规则”。
- 步骤4** 输入告警行动规则名称，如“Test\_1”，主题选择[配置消息通知服务](#)中创建的主题“Test\_1”，单击“确定”。

图 4-21 创建行动规则-AOM

The screenshot shows the 'Create Alert Action Rule' dialog box. The fields filled in are:

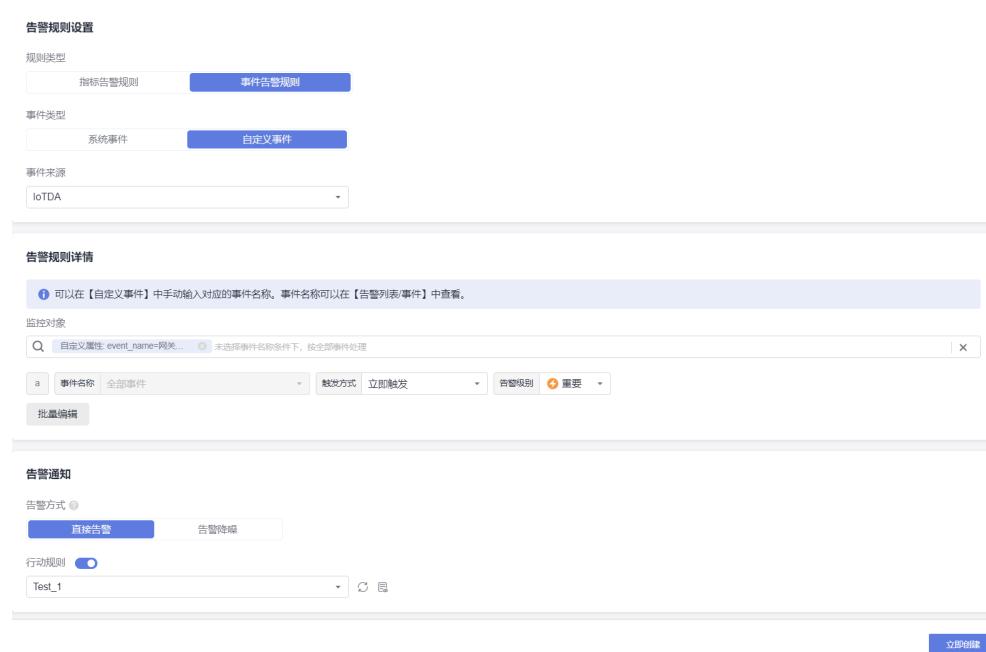
- Action Rule Name: Test\_1
- Enterprise Project: default
- Description: - ↴
- Action Rule Type: 指标或事件 (Metrics or Events)
- Action Method: 通知 (Notification)
- Theme: Test\_1
- Message Template: aom.built-in.template.zh

At the bottom right are '确定' (Confirm) and '取消' (Cancel) buttons.

- 步骤5** 进入“告警 > 告警规则”页面，单击“添加告警”。
- 步骤6** 输入规则名称，如“Gateway\_Status\_Change\_Alarm\_Rule”，选择“事件告警规则 > 自定义事件”，告警来源选择“IoTDA”，监控对象选择自定义属性“event\_name=

网关状态变更”（“=”后面内容为告警名称），触发策略选择“立即触发”，告警方式选择直接告警，行动规则选择步骤4中创建的行动规则，单击右下角“立即创建”。

图 4-22 创建告警规则-AOM



----结束

## 验证操作

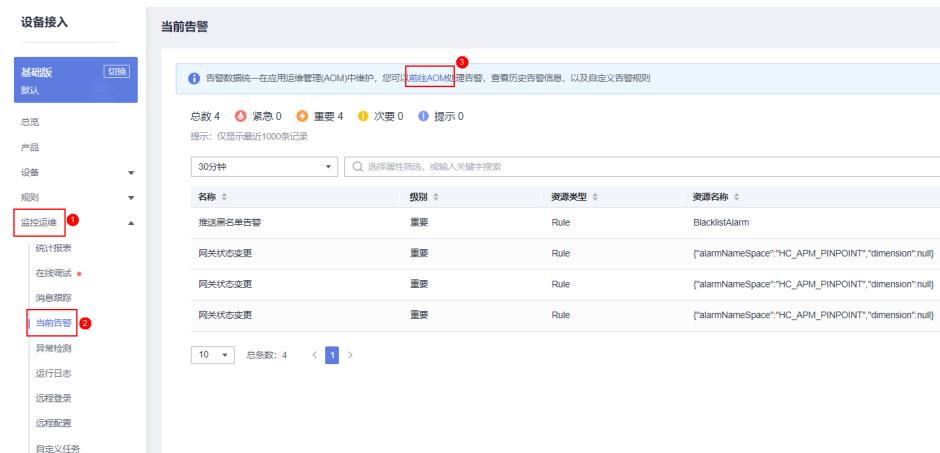
- 您可以使用配置设备接入服务时注册的真实设备接入平台。
- 您也可以使用模拟器模拟设备上下线，操作方法请参考[在线开发MQTT协议的智慧路灯](#)。

期望结果：

- 设备离线持续时长达到5分钟后：

- 在左侧导航栏选择“监控运维 > 当前告警”，单击“前往Aom”，跳转至应用运维管理服务，有一条名称为“网关状态变更”的重要告警。

图 4-23 当前告警-前往 Aom



- 若在应用运维管理服务配置了告警规则和邮件通知告警行动规则，接受订阅的邮箱会收到一封通知网关离线的邮件。
  - 若在应用运维管理服务配置了告警规则和短信通知告警行动规则，接受订阅的手机会收到一条通知网关离线的短信。
2. 设备上线持续时长达到1分钟后：
- 该设备的“网关状态变更”重要告警恢复，可在历史告警中查看。
  - 若在应用运维管理服务配置了告警规则和邮件通知告警行动规则，接受订阅的邮箱会收到一封网关离线恢复的邮件。
  - 若在应用运维管理服务配置了告警规则和短信通知告警行动规则，接受订阅的手机会收到一条网关离线恢复的短信。

# 5 设备发放

## 5.1 设备厂商多实例接入方案

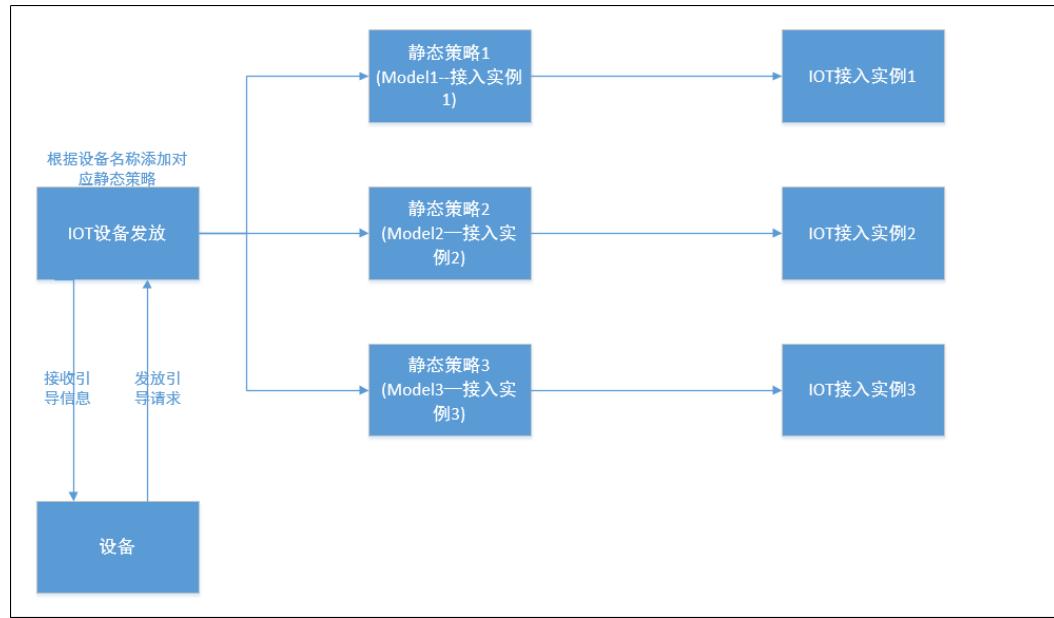
### 场景说明

随着厂商IoT设备的不断增加，或者厂商设备本来就面向多个销售中心（不同的销售中心有着自己的IoT接入实例），面对这样的场景，厂商在设备出厂的时候不知道设备最终需要连接到哪个IoT接入实例，设备出厂后不希望再对设备进行二次烧录。为此，华为IoT解决方案推出设备发放服务，设备出厂统一烧录设备发放的地址，通过在设备发放服务预置不同的发放策略，设备上电后自动发放到对应的IoT接入实例，并把对应的IoT接入实例地址下发给设备，设备就可以和目的实例实现通信。

### 前提条件

- 已注册华为云官方帐号[设备发放服务](#)。未注册可单击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击[实名认证](#)完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“立即使用”后开通该服务。开通设备接入服务，进入到[设备发放服务](#)页面，单击“授权开通”对服务进行授权。
- 对应的设备接入服务已经上传了产品。

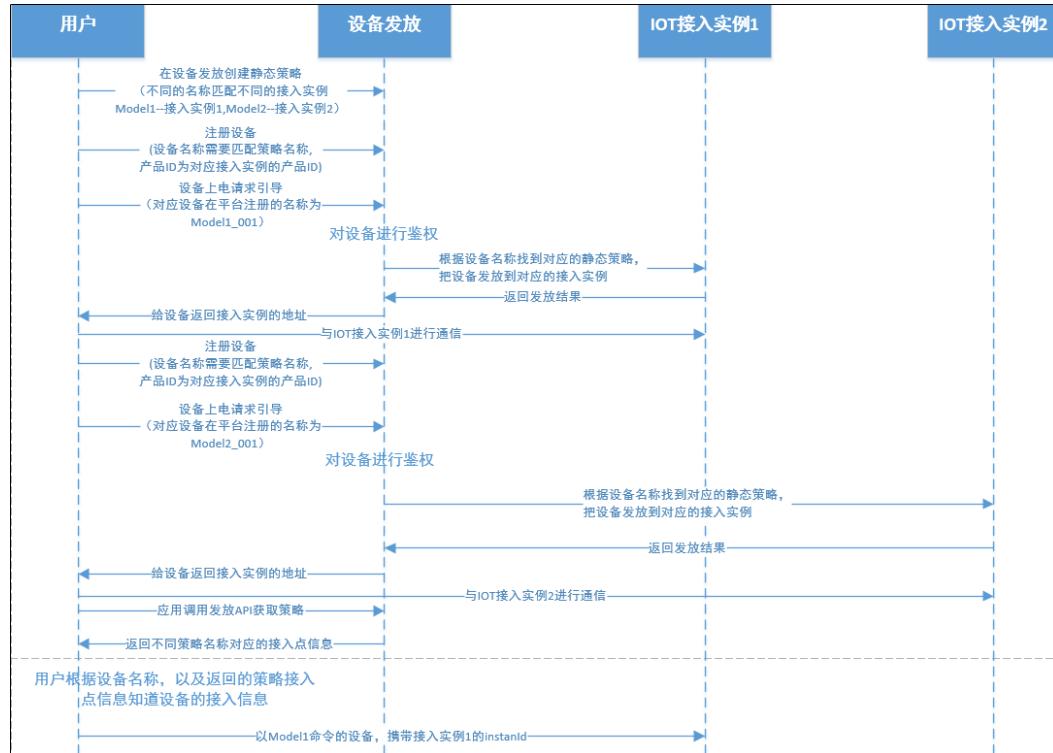
## 整体方案



基于MQTT协议的上行请求引导和下行接受引导信息的业务定义如下：

| 业务场景     | 通信Topic                                     | 报文Payload                      |
|----------|---------------------------------------------|--------------------------------|
| 设备请求引导消息 | \$oc/devices/{device_id}/sys/bootstrap/up   | /                              |
| 服务接收引导信息 | \$oc/devices/{device_id}/sys/bootstrap/down | { "address": "10.0.0.1:8883" } |

## 业务流程



1. 创建策略。其中以静态策略为例。
2. 注册设备。注册MQTT协议的设备，体验发放业务。可以注册单个，也可以批量注册设备。
3. 设备请求引导。
4. 设备接收信息，解析出IoT接入实例地址，与IoT接入实例1进行通信。
5. 动态扩容设备接入实例，创建对应实例的资源空间，在对应资源空间下上传产品。
6. 注册设备。选择对应实例的产品创建设备。
7. 新增策略。添加新实例的静态策略，以不同的关键字命名静态策略。
8. 设备请求引导。
9. 设备接收信息，解析出IoT接入实例地址，与IoT接入实例2进行通信。
10. 应用主动**设备发放控制台**与设备侧通信。

## 创建策略

有两种使用场景：**设备发放控制台**

1. 用户预先开通了多个实例，通过设备的不同销售数据，根据对应的策略，设备上电自动发放到对应的实例。
2. 用户预先只开通一个实例，预先创建对应的策略把设备发放到当前实例。当对应实例达到上限，动态开通新的接入实例，用户删除当前策略，添加新的策略，把新上电的设备发放到新开通的实例。

本示例讲述的是第二种使用场景。

登录**设备发放控制台**，进入到策略页面的静态策略页面，单击添加实例，根据关键字发放到指定的IoTDA实例。

图 5-1 创建静态策略

| 关键词                 | 规则类型 | 授权范围  | 目标区域                              | 目标应用 | 操作 |
|---------------------|------|-------|-----------------------------------|------|----|
| devicename_SNWtctyH | 4条   | 乌兰三-1 | DefaultApp_CNN_1st_00424729_01_st | 删除   |    |
| testm               | 4条   | 乌兰三-1 | DefaultApp_CNN_1st_00424729_01_st | 删除   |    |

图 5-2 上传 CA 证书

| 证书状态 | 证书名称             | 证书使用                           | 证书颁发者                                                        | 使用状态 | 有效期                           | 操作         |
|------|------------------|--------------------------------|--------------------------------------------------------------|------|-------------------------------|------------|
| 自签证书 | ca20200915095729 | serverCommonName20200915095729 | CN=serverCommonName20200915095729,OU=MIT sub group2020091... | 可用   | 2020/09/12 09:57:30 GMT+08:00 | 删除 复制 新建证书 |
| 已签证书 | ca20200915093225 | serverCommonName20200915093225 | CN=serverCommonName20200915093225,OU=MIT sub group2020091... | 可用   | 2020/09/12 09:32:25 GMT+08:00 | 删除 复制 新建证书 |

## 注册设备

本次以注册单个MQTT密钥设备为示例，登录**设备发放控制台**，进入到“设备”页面，单击“注册”，选择对应接入点的产品进行设备注册，设备名称为“Model1\_001”，填写设备的密钥，发放策略选择“静态策略”，创建设备。

### 说明

实际使用中，用户可以使用批量注册功能注册大量设备。下载批量注册模板，按照模板说明正确填写后上传批量注册文件以批量注册设备。建议每个设备名称的以策略的关键字开头（比如策略关键字为Model1，设备名称为Model1\_XXX001）。

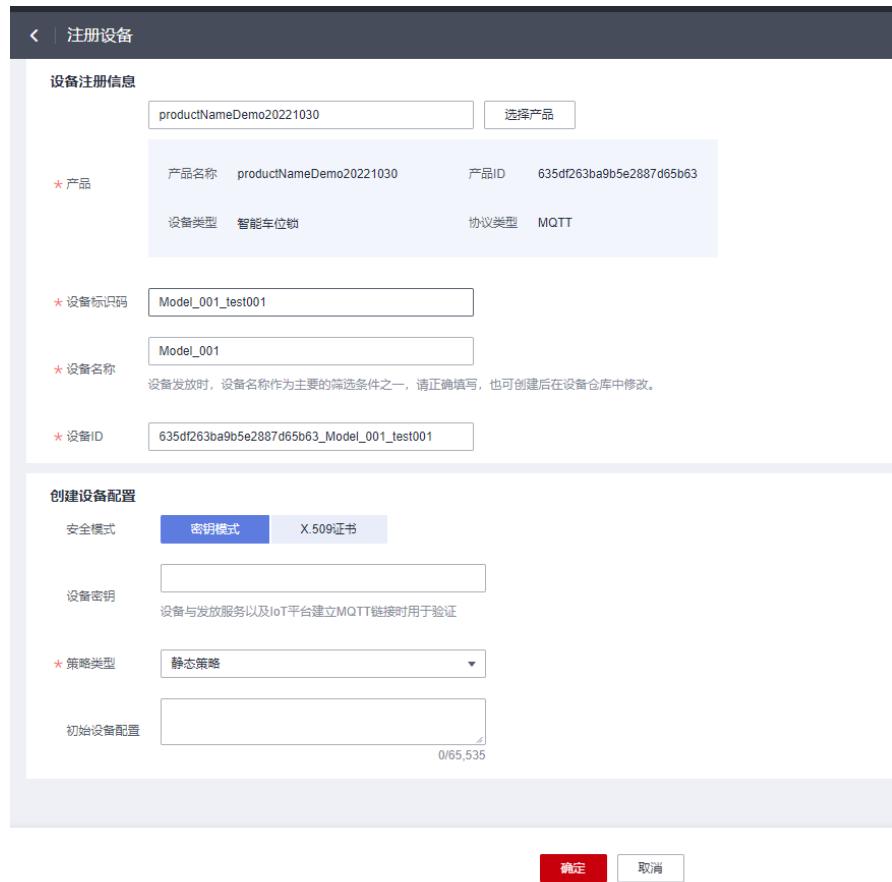
图 5-3 设备列表

The screenshot shows a device management interface with a sidebar on the left containing navigation links like '设备接入' (Device Access), '产品' (Product), '设备' (Device), etc. The main area is titled '设备列表' (Device List) and displays a table of registered devices. The table columns include: 状态 (Status), 设备名称 (Device Name), 设备ID (Device ID), 协议类型 (Protocol Type), 服务类型 (Service Type), 证书名称 (Certificate Name), 所属产品 (Product), 所在区域 (Region), and 发放应用 (Deployment Application). The table shows several entries, each with a color-coded status indicator (e.g., green for successful, red for failed) and a detailed view or edit icon.

图 5-4 注册设备产品列表

This screenshot shows a modal dialog box titled '选择产品' (Select Product) over a larger registration form. The dialog lists various products available for selection. The products listed are: productNameDemo02, testProduct, EchoTest\_DPS\_Device, testMpApp202110, testProduct, EchoTest\_DPS\_API, testMpApp202110, testProduct, test0103, and wirelessLoraServer. Each entry includes the product name, product ID, device type, and protocol type (e.g., MOTT, LWM2M, CoAP). The 'productNameDemo02' entry is currently selected.

图 5-5 创建密钥模式静态策略设备



## 设备引导请求

### 终端节点

| 区域名称   | 区域         | 终端节点 (Endpoint)                     | 端口   | 协议    |
|--------|------------|-------------------------------------|------|-------|
| 华北-北京四 | cn-north-4 | iot-bs.cn-north-4.myhuaweicloud.com | 8883 | MQTTS |

下载并修改华为SDK示例代码进行设备引导（这里以[java sdk代码](#)为示例）。

### 说明

用IDEA/Eclipse打开SDK代码工程，修改iot-device-demo目录下的DEMO示例“BootstrapSample”中的参数，其中deviceId和secret替换为[注册设备](#)中生成的设备ID和密钥即可，bootstrapUri为上述[终端节点](#)。

```
package com.huaweicloud.sdk.iot.device.demo;

import com.huaweicloud.sdk.iot.device.bootstrap.BootstrapClient;

/**
 * 演示设备启动时，通过引导服务获取真实的服务器地址
 */
public class BootstrapSample {

 public static void main(String[] args) {
 String deviceId = "60136b0682401c03e0b1ccf5_aaa_device2";
 String secret = "73ce5656f5d4a9919532";
 String bootstrapUri = "ssl://100.95.158.64:8883";

 // 创建引导客户端，发起引导
 BootstrapClient bootstrapClient = new BootstrapClient(bootstrapUri, deviceId, secret);
 DefaultBootstrapActionListener defaultBootstrapActionListener = new DefaultBootstrapActionListener(deviceId,
 secret, bootstrapClient);
 bootstrapClient.bootstrap(defaultBootstrapActionListener);

 }
}
```

运行DEMO程序，看到如下日志，代表设备发放成功，并且已经收到设备发放下发的设备接入地址。如果程序运行没报错，在对应的设备接入平台可以看到设备，并已在上线。

```
2021-01-29 14:46:47 INFO BootstrapClient:50 - create BootstrapClient: 60136b0682401c03e0b1ccf5_aaa_device2
2021-01-29 14:46:48 INFO MqttConnection:167 - try to connect to ssl://100.95.158.64:8883
2021-01-29 14:46:48 INFO MqttConnection:169 - connect success ssl://100.95.158.64:8883
2021-01-29 14:46:48 INFO MqttConnection:185 - Mqtt client connected. address :ssl://100.95.158.64:8883
2021-01-29 14:46:48 INFO MqttConnection:240 - publish message topic = $oc/devices/$oc/60136b0682401c03e0b1ccf5_aaa_device2/sys/bootstrap/up, msg =
2021-01-29 14:46:48 INFO DefaultBootstrapActionListener:30 - bootstrap success:$oc/devices/$oc/60136b0682401c03e0b1ccf5_aaa_device2/sys/bootstrap/down
2021-01-29 14:46:49 INFO DefaultBootstrapActionListener:30 - bootstrap success:null
2021-01-29 14:46:49 INFO MqttConnection:105 - messageArrived topic = $oc/devices/$oc/60136b0682401c03e0b1ccf5_aaa_device2/sys/bootstrap/down, msg = {"address":"100.95.174.182:1883","initConfig":null}
2021-01-29 14:46:49 INFO BootstrapClient:102 - bootstrap ok address=100.95.174.182:1883
2021-01-29 14:46:49 INFO DefaultBootstrapActionListener:30 - bootstrap success:100.95.174.182:1883
```

收到设备发放下发的设备接入地址后，需要关闭设备侧的设备发放的连接，用新的URL地址与设备接入通信，进行相关业务。

```
public class DefaultBootstrapActionListener implements ActionListener {

 private static final Logger log = LogManager.getLogger(DefaultBootstrapActionListener.class);

 private String deviceId;
 private String secret;
 private BootstrapClient bootstrapClient;

 public DefaultBootstrapActionListener(String deviceId, String secret, BootstrapClient bootstrapClient) {
 this.deviceId = deviceId;
 this.secret = secret;
 this.bootstrapClient = bootstrapClient;
 }

 @Override
 public void onSuccess(Object context) {
 String address = (String) context;
 log.info("Bootstrap success:" + address);

 // 引导成功后关闭客户端
 bootstrapClient.close();
 IoTDevice device = new IoTDevice(serverUri: "ssl://" + address, deviceId, secret);
 if (device.init() != 0) {
 return;
 }

 // 上报消息
 device.getClient().reportDeviceMessage(new DeviceMessage("hello"), listener: null);
 }

 @Override
 public void onFailure(Object context, Throwable var2) {
 log.error("bootstrap WFH failed: {}", var2.getMessage());
 }
}
```

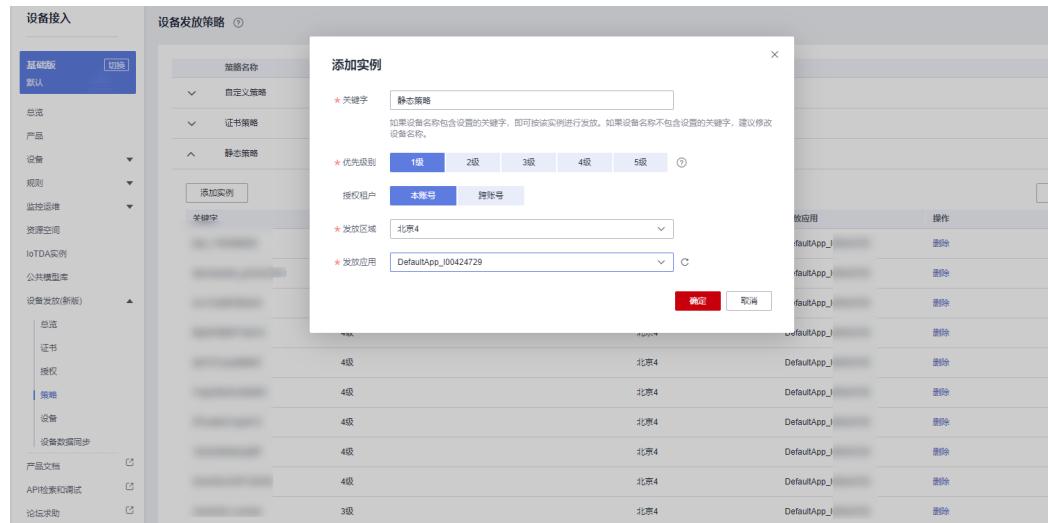
## 新增 IoT 接入实例

用户新增IoT接入实例，创建对应实例的资源空间，在对应资源空间下上传产品。

## 新增策略

当新的实例扩容后，添加新实例的静态策略，通过不同的关键字命名关联新的实例。

图 5-6 创建静态策略详情



然后按照上面的操作以新策略的名称注册设备，请求引导即可。

## 应用主动与设备通信

多实例场景下，当应用侧需要主动与设备侧通信时，比如主动下发命令，需要应用侧知道设备是在哪个实例上，应用要规划好每批命令设备发放哪个实例，在设备发放通过查询静态策略，找到对应设备名称的设备的实例ID，调用设备接入北向API下发命令需要指定实例ID。

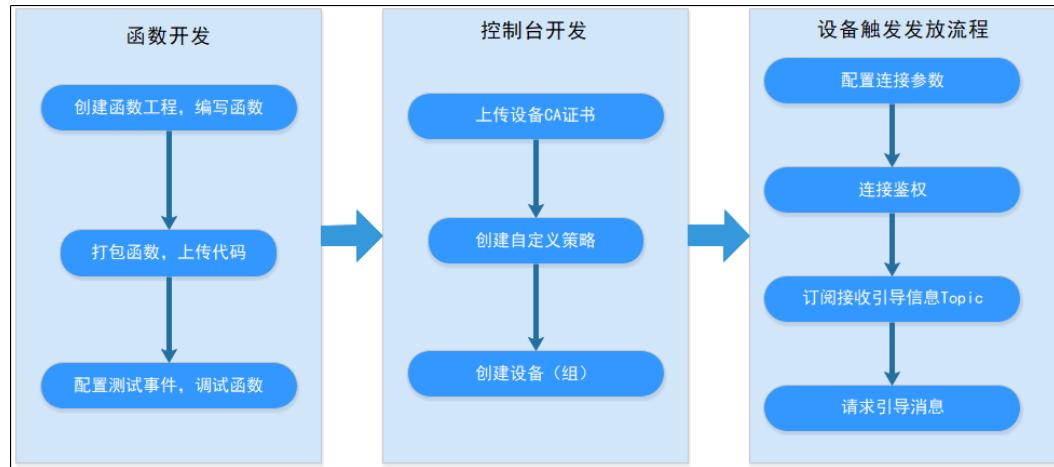
## 5.2 结合函数服务通过自定义策略发放证书认证的设备

### 场景说明

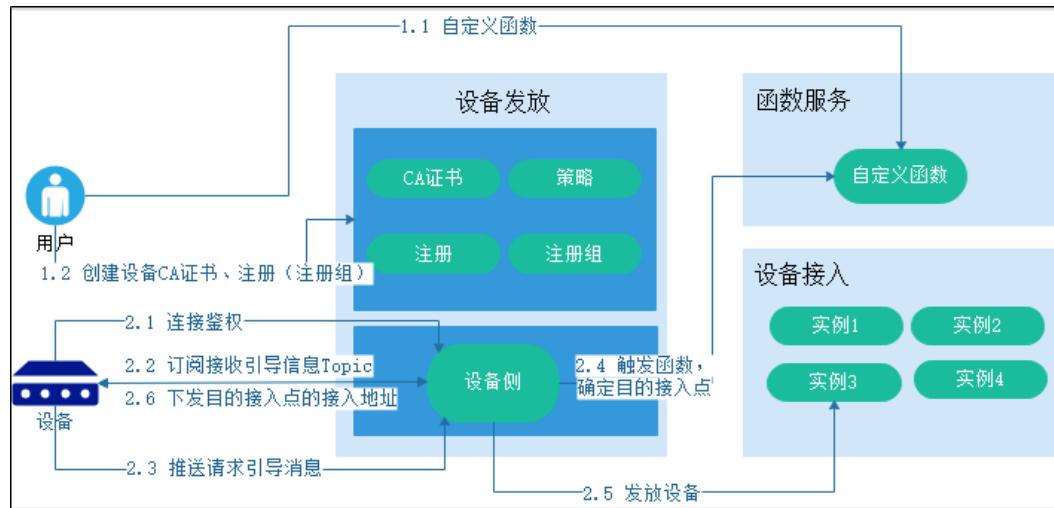
本文以MQTT.fx的设备模拟器替代真实的设备，结合[函数工作流服务（FunctionGraph）](#)，带您快速体验结合函数服务使用设备发放服务，通过自定义策略将设备发放到指定的物联网平台（设备接入实例）上。

### 整体流程

使用证书认证的设备采用自定义策略发放设备操作流程如下图所示。



通过自定义策略发放设备功能流程图如下图所示。

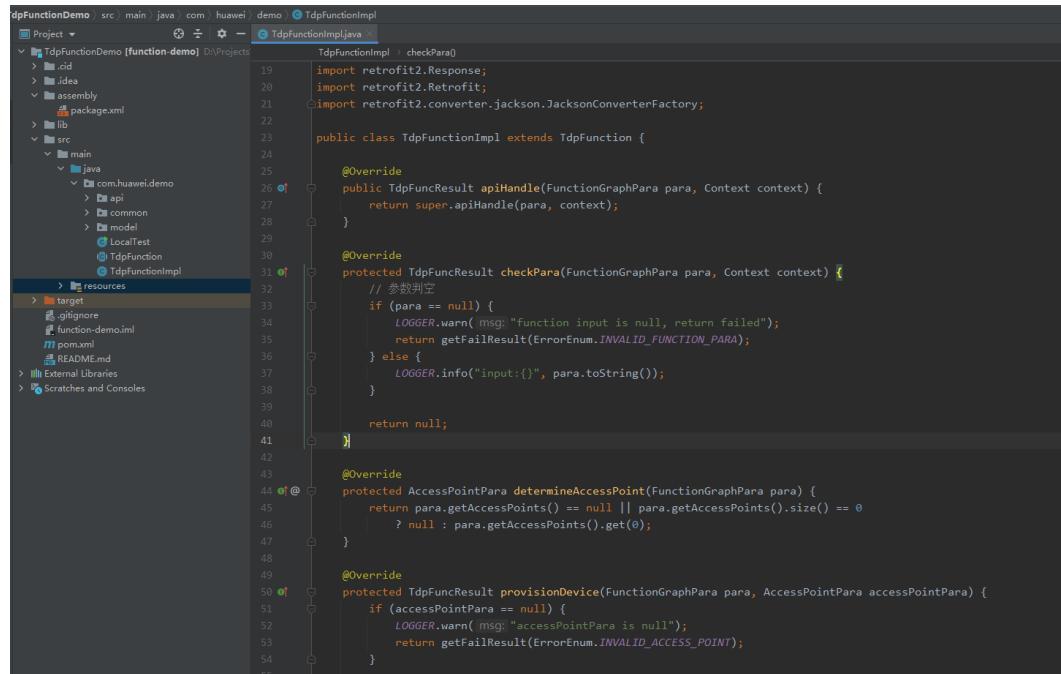


## 函数开发

自定义策略的函数代码编写指南参见[创建自定义策略函数](#)。

此处以Java语言为例，如何编写Java语言的函数，参见[函数工作流 FunctionGraph > 开发指南> 如何开发函数> Java函数开发指南](#)。

**步骤1** 创建函数工程，编写函数。



```
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.jackson.JacksonConverterFactory;

public class TdpFunctionImpl extends TdpFunction {

 @Override
 public TdpFuncResult apiHandle(FunctionGraphPara para, Context context) {
 return super.apiHandle(para, context);
 }

 @Override
 protected TdpFuncResult checkPara(FunctionGraphPara para, Context context) {
 // 参数判断
 if (para == null) {
 LOGGER.warn("msg: " + para);
 return getFailResult(ErrorEnum.INVALID_FUNCTION PARA);
 } else {
 LOGGER.info("input:{}", para);
 }
 return null;
 }

 @Override
 protected AccessPointPara determineAccessPoint(FunctionGraphPara para) {
 return para.getAccessPoints() == null || para.getAccessPoints().size() == 0
 ? null : para.getAccessPoints().get(0);
 }

 @Override
 protected TdpFuncResult provisionDevice(FunctionGraphPara para, AccessPointPara accessPointPara) {
 if (accessPointPara == null) {
 LOGGER.warn("msg: " + accessPointPara);
 return getFailResult(ErrorEnum.INVALID_ACCESS_POINT);
 }
 }
}
```

关键流程代码参考如下：

```
package com.huawei.demo;

import com.huawei.demo.common.logger.DefaultLogger;
import com.huawei.demo.model.AccessPointPara;
import com.huawei.demo.model.FunctionGraphPara;
import com.huawei.demo.model.TdpFuncResult;
import com.huawei.services.runtime.Context;

import java.util.Optional;

/**
 * 实现该抽象类。
 * WARNING: {@link #apiHandle(FunctionGraphPara, Context)} 该方法必须在子类中定义，否则函数无法触发该函数接口！！！
 * <pre>
 * {@code
 * @Override
 * public TdpFuncResult apiHandle(FunctionGraphPara para, Context context) {
 * return super.apiHandle(para, context);
 * }
 * }
 * </pre>
 */
public abstract class TdpFunction {
 protected static final DefaultLogger LOGGER = new DefaultLogger(TdpFunction.class);

 /**
 * 函数定义
 */
 public TdpFuncResult apiHandle(FunctionGraphPara para, Context context) {
 // 获取日志
 DefaultLogger.init(Optional.ofNullable(context)
 .map(Context::getLogger)
 .orElse(null));

 // 校验入参
 TdpFuncResult result = checkPara(para, context);
 if (result != null) {
 return result;
 }
 }
}
```

```
// 确定接入点
AccessPointPara accessPointPara = determineAccessPoint(para);

// 发放设备
result = provisionDevice(para, accessPointPara);

LOGGER.info("result:{}", result);
return result;
}

/**
 * 校验入参
 */
protected abstract TdpFuncResult checkPara(FunctionGraphPara para, Context context);

/**
 * 从备选的接入点中选择合适的接入点
 */
protected abstract AccessPointPara determineAccessPoint(FunctionGraphPara para);

/**
 * 调用设备发放的发放接口发放设备
 */
protected abstract TdpFuncResult provisionDevice(FunctionGraphPara para, AccessPointPara
accessPointPara);
}
```

## 步骤2 打包函数，上传代码。

如不依赖第三方Jar包，可将工程打包成一个单独的Jar包，如下图所示。



如工程依赖第三方Jar包，则需将工程Jar包和所有第三方Jar包打包成一个zip包，如下图所示。



进入函数工作流服务控制台，创建函数。函数执行入口以[包名].[类名].[执行函数名]格式填写。

创建函数

当前配置

|        |                                           |
|--------|-------------------------------------------|
| 函数名称   | functionDemo-20210413                     |
| 委托名称   | 不使用任何委托                                   |
| 运行时语言  | Java 8                                    |
| 函数执行入口 | com.huawei.demo.TdpFunctionImpl.apiHandle |

配置费用

按用量收费

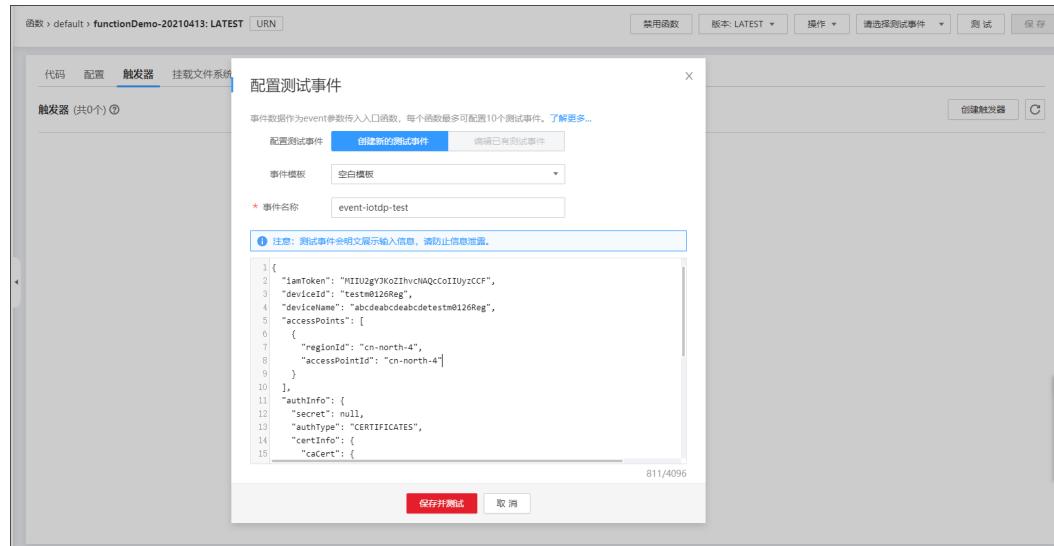
创建免费，使用阶段按用量收费。[了解计费详情](#)

函数运行过程中产生的日志管理费用按需收取。[了解应用运维管理服务计费详情](#)

创建函数 取消

### 步骤3 配置测试事件，调试函数。

创建并配置测试事件，调试函数，如下图所示。



----结束

## 控制台开发

登录[设备发放控制台](#)。

**步骤1** 进入“[证书](#)”界面，上传并验证设备CA证书。

如何上传并验证证书参见[证书](#)。

图 5-7 证书列表

| 证书状态 | 证书名称                          | 证书使用者                              | 证书颁发者                                                            | 过期标志 | 到期时间                          | 操作             |
|------|-------------------------------|------------------------------------|------------------------------------------------------------------|------|-------------------------------|----------------|
| 已验证  | ca020209101749                | iot ca                             | CN=iot ca, OU=iot, O=tiet, L=SZ, ST=GD, C=CN                     | 可用   | 2159/05/03 17:49:41 GMT+08:00 | 更新   刷新   验证证书 |
| 未验证  | AZURE                         | Microsoft RSA TLS CA 02            | CN=Baltimore CyberTrust Root, O=CyberTrust, O=Baltimore, C=IE    | 可用   | 2024/10/08 15:00:00 GMT+08:00 | 更新   刷新   验证证书 |
| 已验证  | caCertDemo                    | deviceCaCommonName02020323174931   | CN=deviceCaCommonName02020323174931, OU=company sub group02...   | 可用   | 2032/03/19 17:46:31 GMT+08:00 | 更新   刷新   验证证书 |
| 已验证  | 0eb1c1f699c4621580e455293efec | serverCommonName0202032024170723   | CN=serverCommonName0202032024170723, OU=MIT sub group02...       | 可用   | 2120/01/30 17:07:23 GMT+08:00 | 更新   刷新   验证证书 |
| 已验证  | server20211212                | ca.iot.huaweicloud.com             | CN=ca.iot.huaweicloud.com, OU=iot, O=Huawei, L=SZ, ST=GD, C=CN   | 已过期  | 2022/12/12 15:36:55 GMT+08:00 | 更新   刷新   验证证书 |
| 已验证  | EchoTest_DPS_API              | serverCommonName020020211112155408 | CN=serverCommonName020020211112155408, OU=MIT sub group02...     | 可用   | 2121/10/18 15:54:08 GMT+08:00 | 更新   刷新   验证证书 |
| 已验证  | EchoTest_DPS_STRATEGY         | serverCommonName020120211112155408 | CN=serverCommonName020120211112155408, OU=MIT sub group02...     | 可用   | 2121/10/18 15:54:08 GMT+08:00 | 更新   刷新   验证证书 |
| 已验证  | ca02010916204622              | serverCommonName02010916204622     | CN=serverCommonName02010916204622, OU=MIT sub group02010916...   | 可用   | 2031/09/13 20:46:22 GMT+08:00 | 更新   刷新   验证证书 |
| 已验证  | test                          | serverCommonName02010813182522     | CN=serverCommonName02010813182522, OU=MIT sub group02010813...   | 可用   | 2031/08/10 18:25:23 GMT+08:00 | 更新   刷新   验证证书 |
| 已验证  | ca20210207194023              | serverCommonName0202010207194023   | CN=serverCommonName0202010207194023, OU=MIT sub group02020102... | 可用   | 2030/10/04 19:40:23 GMT+08:00 | 更新   刷新   验证证书 |

**步骤2** 进入“[策略](#)”界面，添加自定义策略。

如何添加自定义策略实例参见[添加自定义策略实例](#)。

图 5-8 添加自定义策略

The screenshot shows the 'Device Release' section under the 'Custom Strategy' tab. It displays a table of existing strategies, each with a name, description, and operation status. A red box highlights the 'Add Instance' button at the top right of the table area.

步骤3 进入“设备”界面，单击左上角“注册设备”，进行注册设备。

如何注册设备，参见[设备](#)。

图 5-9 注册设备

The screenshot shows the 'Device Registration' list page. It lists various registration attempts, each with a status (e.g., Success, Failed), device information, and operation details. A red box highlights the 'Registration List' tab at the top left of the table area.

图 5-10 创建证书模式自定义策略设备



----结束

## 设备触发放流程

### 步骤1 配置连接参数。

证书认证设备如何使用MQTT.fx接入设备发放请参见[MQTT X509证书认证接入示例](#)。

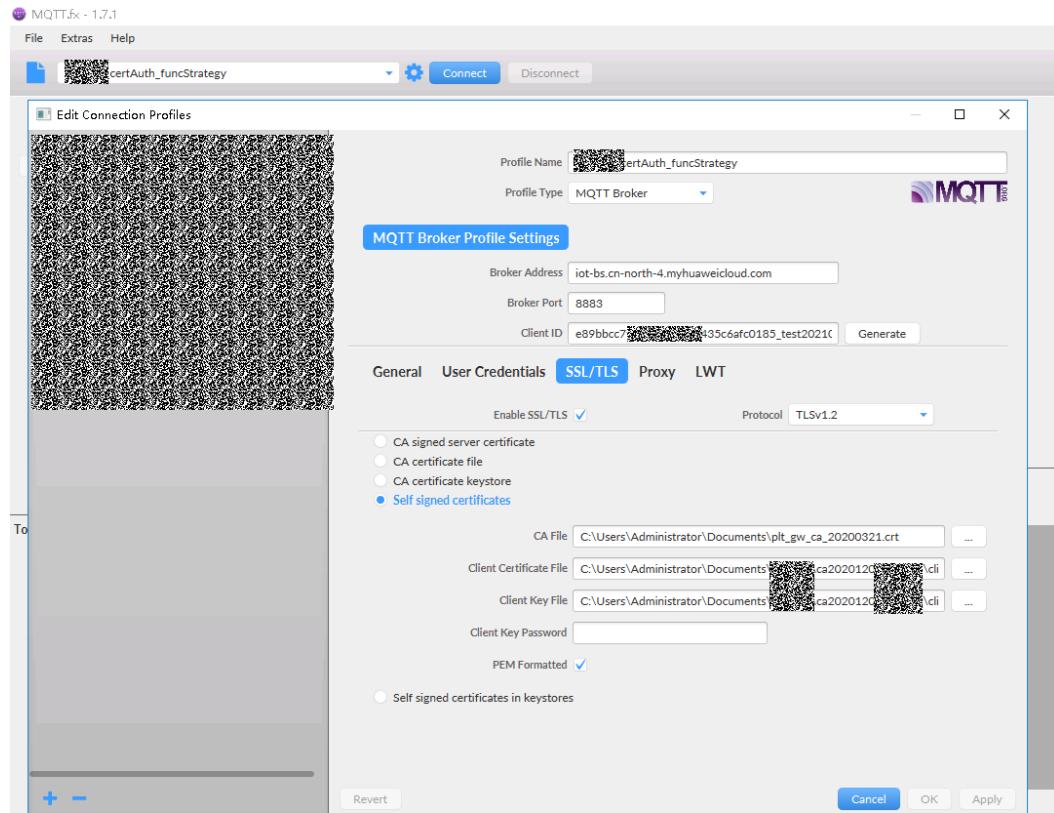
#### MQTT Broker Profile Settings

Broker Address填写设备发放设备侧接入地址，Broker Port填写设备发放设备侧接入端口，Client ID按规则（[设备ID]\_0\_0\_[10位日期时间]）填写，具体规则参见[MQTT CONNECT连接鉴权](#)。

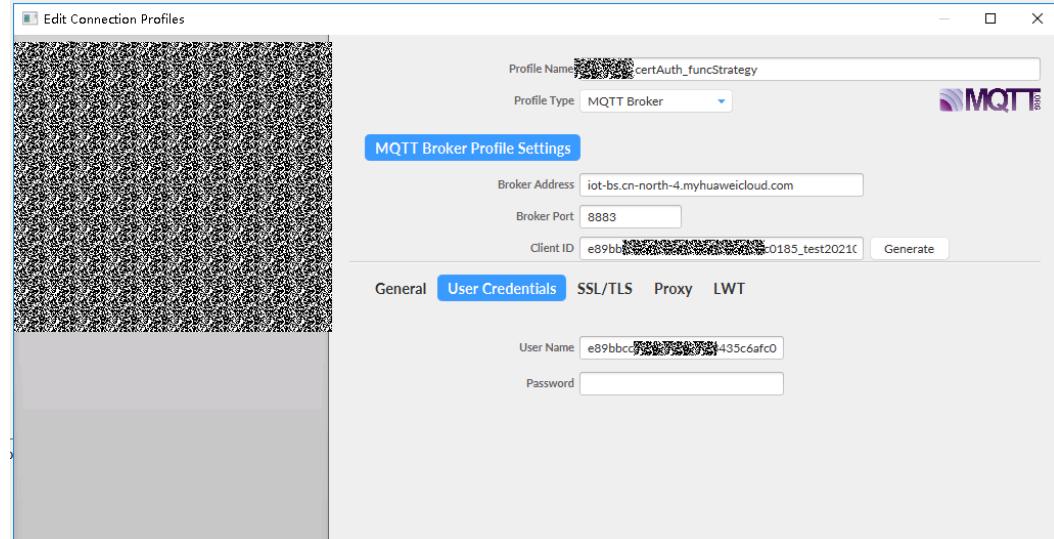
#### SSL/TLS

由于设备使用证书认证方式，“Enable SSL/TLS”勾选开启，选择Self signed certificates，CA File填写用于签发设备发放设备侧接入SSL/TLS通道证书的CA证书路径，Client Certificate File和Client Key File填写使用已上传并验证了的CA证书签发的设备证书和设备证书私钥路径。如证书为PEM格式，请勾选“PEM Formatted”。

具体配置请参见下图：

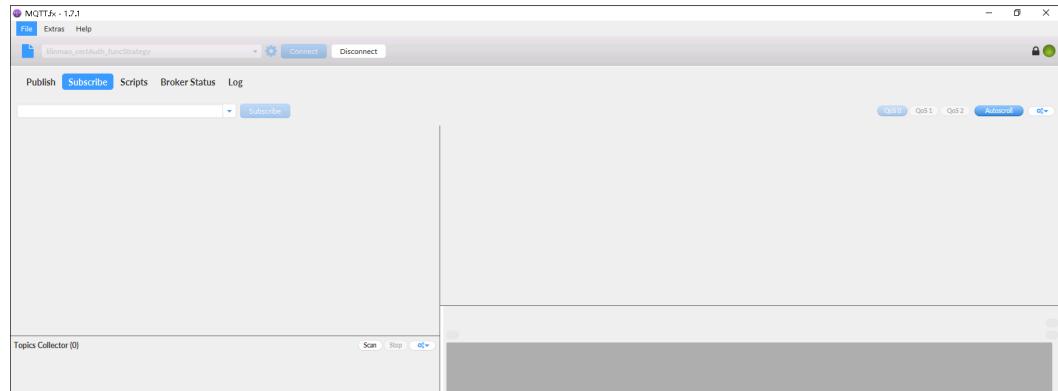


“User Credentials”的“User Name”字段填写设备ID。具体配置参见下图：



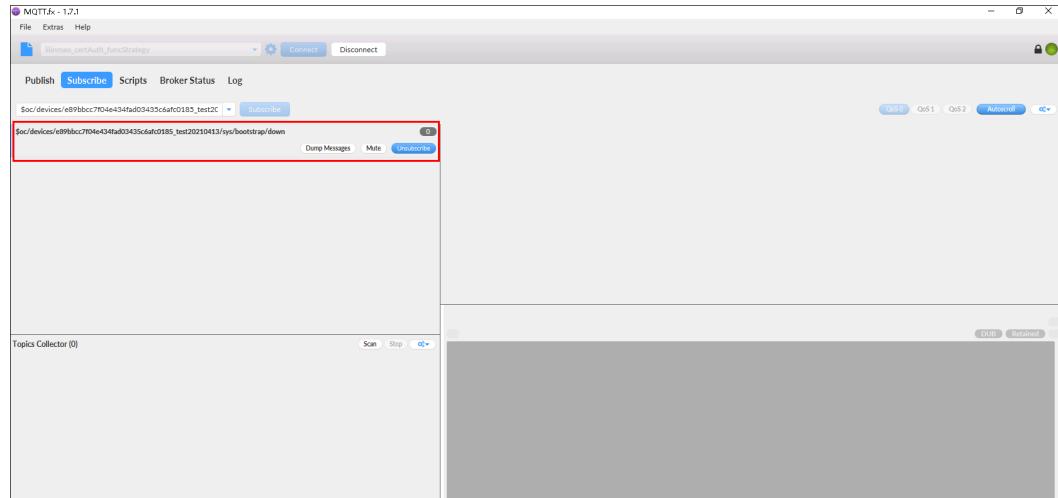
## 步骤2 连接鉴权。

连接参数配置完成后，单击“Connnect”。若右上角圆圈呈现绿色，说明设备已成功上设备发放平台。



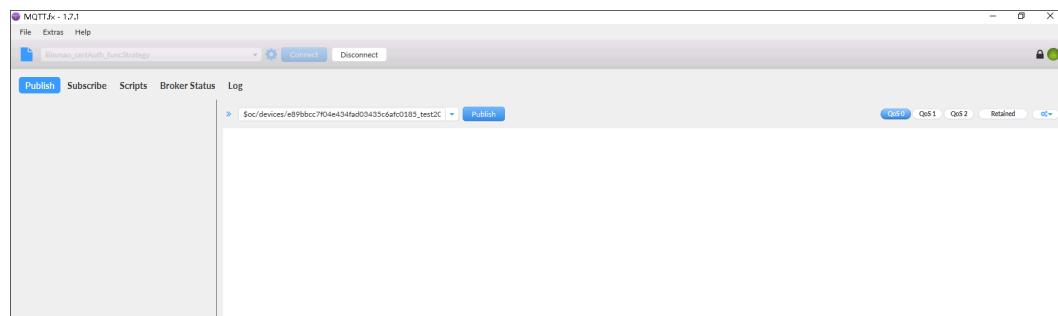
### 步骤3 订阅接收引导消息Topic。

参照**设备接收引导信息**填写用于接收引导地址的Topic，单击“Subscribe”订阅该Topic。若订阅Topic填写框下方的已订阅Topic列表中存在该Topic，则订阅成功。

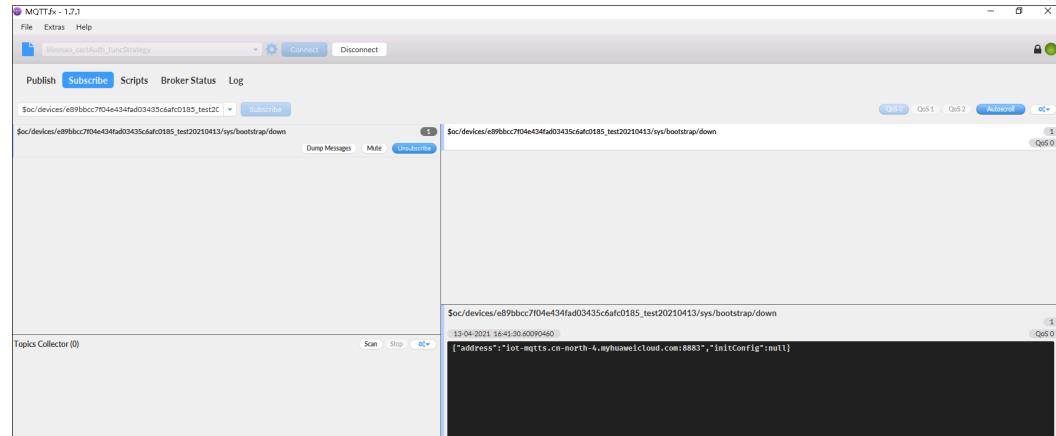


### 步骤4 发送请求引导消息。

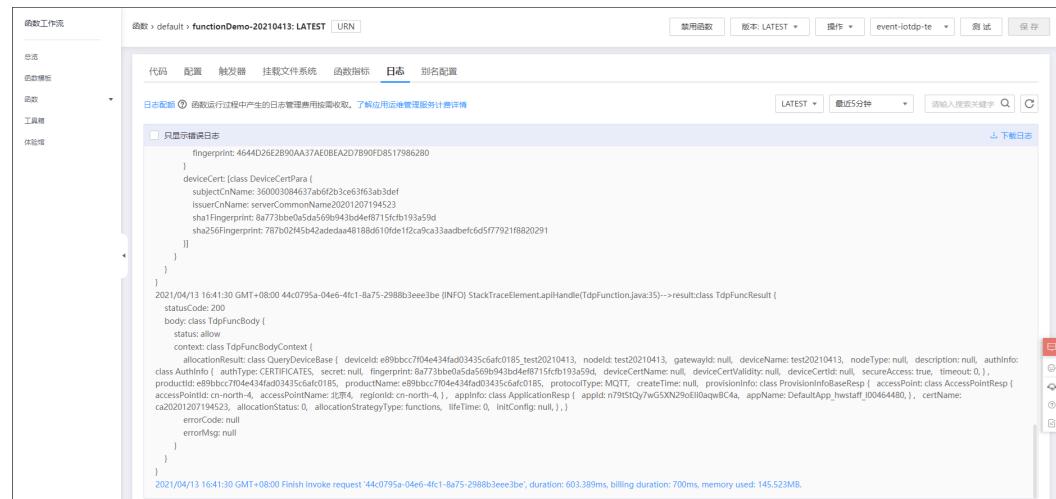
参照**设备请求引导消息**填写发送请求引导消息的Topic，单击“Publish”向该Topic推送消息。



查看订阅的Topic，很快在订阅Topic下，接收到了目的接入点的设备侧接入地址。



进入函数服务控制台，通过查看使用的函数的日志，可看出自定义策略发放设备过程中，该函数被成功触发且正确执行。



进入到设备发放控制台，可查看到一条该设备的发放记录。

图 5-11 设备详情页

| 状态 | 发放策略                         | 发放区域 | 发放应用                        | 失败原因 | 执行时间                          |
|----|------------------------------|------|-----------------------------|------|-------------------------------|
| 成功 | 自定义策略(functionDemo-20210413) | 北京4  | DefaultApp_hwstaff_00464480 | --   | 2021/04/13 16:41:30 GMT+08:00 |

至此，您已成功完成了，使用设备发放和函数服务将证书认证的设备通过自定义策略触发函数将设备发放到指定的设备接入实例中。

----结束

## 5.3 共享实例存量设备同步

### 场景说明

本文介绍如何通过设备发放将共享实例中的存量设备数据同步到专享实例。

相比于共享实例，专享实例运行在用户指定的虚拟私有云内，用户可以使用独立分配的公网和私网IP接入物联网平台。用户购买的资源完全独享，资源安全性和隔离性更高，是企业设备上云的首选，实例详情介绍参考华为云设备接入[IoTDA实例介绍](#)。

若用户已经开通共享实例，设备也在共享实例上运行，但是用户想让设备运行在专享实例上，可以将存量设备同步到用户购买的专享实例(企业版)中管理。通过设备发放，可以很方便地将共享实例中的设备同步到新的专享实例。

### 注意事项

- 使用设备发放服务跨实例同步设备时，如果是指定资源空间和设备同步，同步数据仅包括设备连接新实例认证需要的身份信息，包括产品、证书和设备；不包括原实例中的设备分组、服务端订阅、规则引擎配置和平台日志等；支持同步设备的状态。
- 目前除了支持资源空间和设备的同步，还能指定插件的同步。
- 设备同步成功后，可以通过设备发放获取到新实例的地址，也可以直接更新设备固件，对接新的实例。如果用户的设备不支持更新固件，也没有对接设备发放，那么可以[提交工单](#)说明诉求。
- 设备同步到新实例后，应用侧调用设备接入实例API管理设备时，所有相关API调用的InstanceId需修改为新实例ID。

### 操作步骤

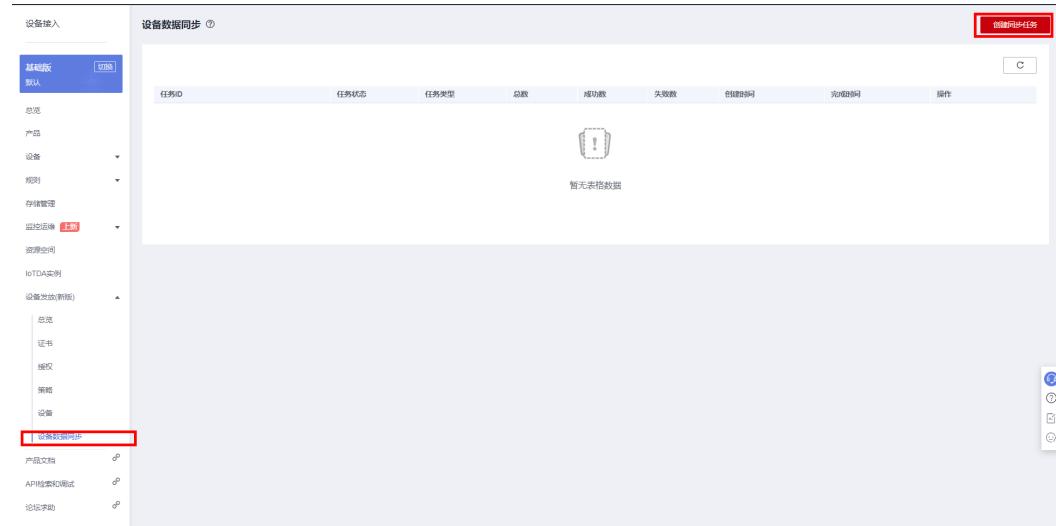
**步骤1** 默认用户已经开通通过共享版设备接入。访问[设备接入服务](#)，单击“立即使用”进入“设备接入控制台”，通过“设备接入”进入到“设备发放服务”页面，单击“授权开通”对服务进行授权。

**步骤2** 开通专享实例，参考设备接入[实例开通管理](#)。

**步骤3** 根据业务进行同步操作，如果设备量大，选择按照资源空间同步。

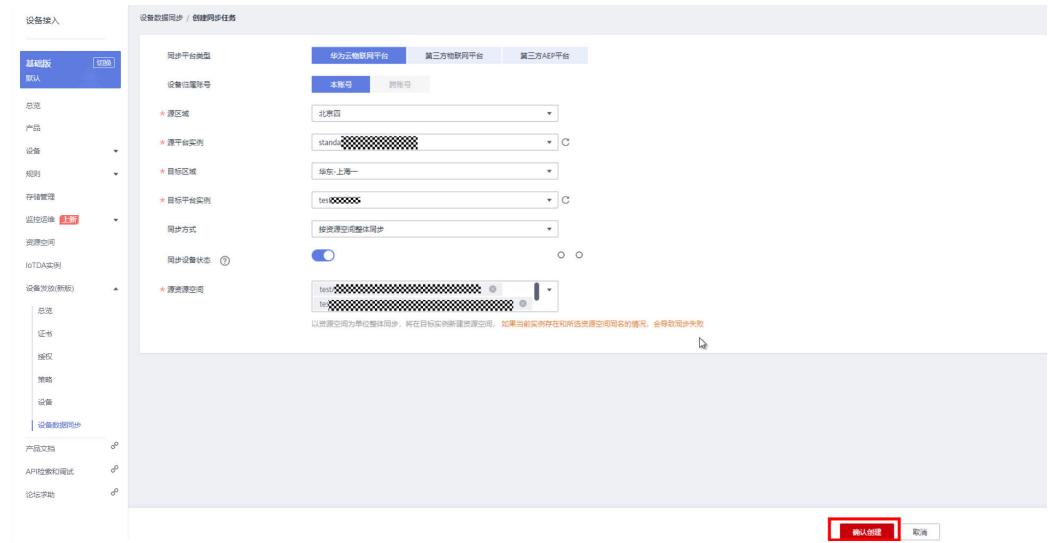
**步骤4** 进入设备发放“设备数据同步”界面，单击右上角“创建同步任务”。

图 5-12 创建同步任务



**步骤5** 同步方式选择“按资源空间整体同步”，将所有必填选项选择完毕后，单击“确认创建”即可，设备发放会自动把对应资源空间的产品、证书和设备数据同步过去。

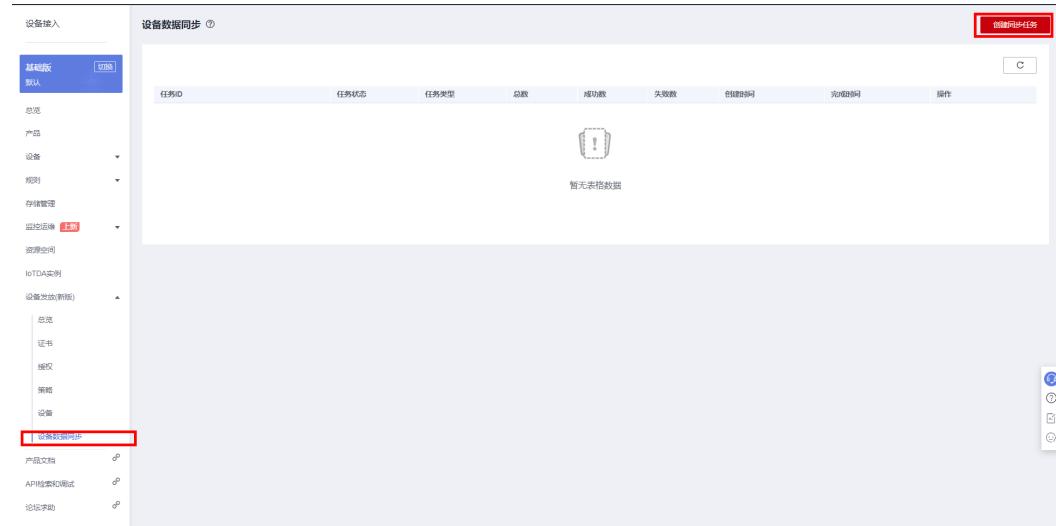
图 5-13 创建内部 region 按资源空间整体迁移任务



**步骤6** 如果设备量小，可以选择“指定设备同步”方式。

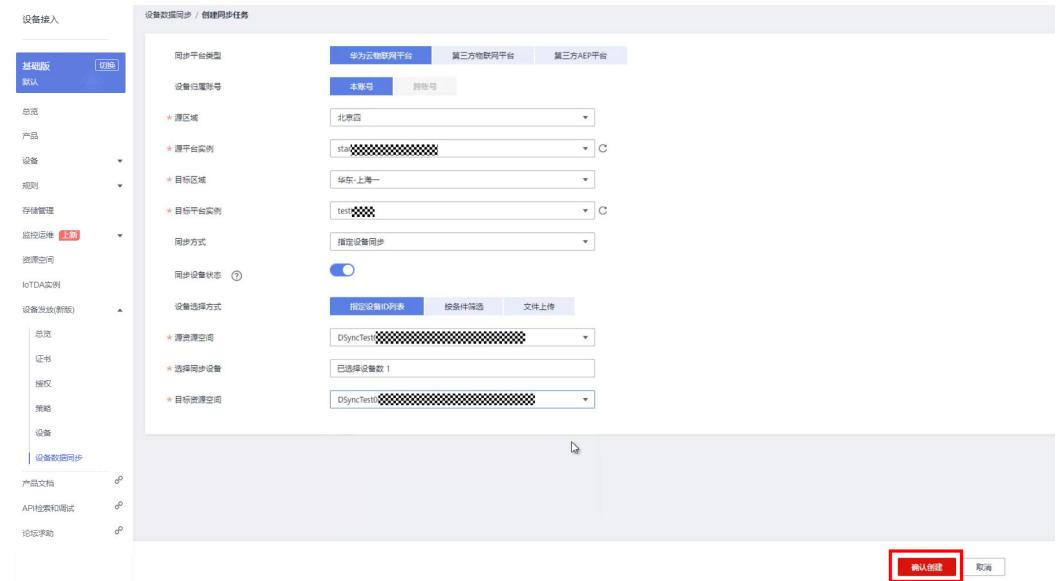
**步骤7** 进入“设备数据同步”界面，单击右上角“创建同步任务”。

图 5-14 创建同步任务



**步骤8** 同步方式选择“指定设备同步”，将所有必填选项选择完毕后，单击“确认创建”。设备发放会自动把对应产品下的设备数据同步过去，同步后的设备状态保持一致。

图 5-15 创建华为云内部 region 按指定设备 id 迁移任务



**步骤9** 单击“任务ID”或者“详情”，可以看到刚创建的任务的详细内容，并且可以使用“高级搜索”对同步设备进行过滤查询操作。

图 5-16 设备迁移任务详情页高级搜索

The screenshot shows the 'Device Migration Task Detail Page'. On the left, there's a sidebar with navigation items like '基础版' (Basic), '高级版' (Advanced), '总览', '产品', '设备', '规则', '存储管理', '同步设备', '资源空间', 'IoTDA 实例', '设备发放记录', and '总览'. The main content area has a title '设备数据同步 - 同步任务详情 (ID: 7e254a00xxxxxx)' and a summary table with columns: '操作' (Operation), '同步设备总数' (Number of synchronized devices), '成功数' (Success count), and '失败数' (Failure count). Below this is a detailed table for '设备' (Devices) with columns: '状态' (Status), '设备名称' (Device name), '设备标识码' (Device identifier), '设备ID' (Device ID), '所属产品ID' (Product ID), '所属资源空间ID' (Resource space ID), '失败原因' (Failure reason), and '同步时间' (Sync time). A red box highlights the '高级搜索' (Advanced search) button at the top right of the table.

### 说明

当同步任务类型为“指定设备同步”时，此类任务下所有同步设备均属于同一应用、同一产品。因此在进行同步设备列表的“高级搜索”时，搜索项“所属产品”和“所属资源空间”均为空。

**步骤10** 在设备同步详情页，可以查看单个设备的信息和设备的发放记录。

图 5-17 设备迁移任务详情页

The screenshot shows the 'Device Migration Task Detail Page'. The left sidebar is identical to the previous screenshot. The main area has a summary table with a green dot indicating '已发放' (Deployed) status, '所在区域' (Region), and '所属资源空间' (Resource space). Below this is a table for '发放记录' (Deployment records) with columns: '状态' (Status), '发放策略' (Deployment strategy), '发放区域' (Deployment region), '失败原因' (Failure reason), and '执行时间' (Execution time). A red box highlights the '-----结束' (-----End) text at the bottom of the page.

-----结束

## 5.4 设备自动注册安全接入示例

### 场景说明

平台基于客户业务安全考虑，只有将设备的基本信息（例如设备ID、鉴权信息）注册到平台后，设备才能使用成功注册后的设备ID和鉴权信息接入物联网平台。当用户注册的设备不断增多时，如何安全且高效地注册设备变得尤为关键。

一种方案是，使用批量注册模板或者循环调用平台应用侧注册设备的API，实现高效地批量注册设备。此方案能够满足客户对批量注册设备的大部分场景需求，但要求客户业务应用侧和客户产线侧相互配合，客户应用侧对设备ID提前进行规划，应用侧完成批量注册后，客户业务产线侧需将应用侧获取到的设备ID逐一烧录到各个设备比较麻烦。

另一种更加高效的方案是，使用设备发放的注册组功能实现设备免注册安全极简地接入物联网平台。

本文基于设备发放注册组功能，使用证书策略，带您快速体验设备免注册安全极简地接入物联网平台。

表 5-1 方案对比

| 设备注册方案 | 基于单个注册                                                                                                                                      | 批量注册                                                                                                                                                             | 免注册安全极简接入                                                                                                       |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| 适用场景   | <ul style="list-style-type: none"><li>设备数量不多；</li><li>对设备身份标识和认证信息可提前规划；</li><li>产线侧与应用侧可紧密配合。</li></ul>                                    | <ul style="list-style-type: none"><li>设备数量多；</li><li>对设备身份标识和认证信息可提前规划。</li></ul>                                                                                | <ul style="list-style-type: none"><li>设备数量较多；</li><li>设备能力强；</li><li>对设备注册的灵活性要求高。</li></ul>                    |
| 过程     | <ol style="list-style-type: none"><li>应用侧通过多次调用单个注册API将设备信息（包括但不限于身份标识和认证信息）逐一注册到平台；</li><li>产线侧与应用侧密切配合，将设备身份标识和与之对应的认证信息逐个写入设备。</li></ol> | <ol style="list-style-type: none"><li>下载并填写批量注册模板文件，定义设备信息（包括但不限于身份标识和认证信息）；</li><li>将填写完成的批量注册模板文件上传至平台，完成批量注册过程；</li><li>产线侧将设备身份标识和与之对应的认证信息逐个写入设备。</li></ol> | <ol style="list-style-type: none"><li>基于设备发放的注册组和发放策略能力；</li><li>具体过程在本文后续章节展开。</li></ol>                       |
| 特点     | <ul style="list-style-type: none"><li>多次调用API；</li><li>整体流程简单。</li></ul>                                                                    | 将单个注册过程模板化。                                                                                                                                                      | <ul style="list-style-type: none"><li>无需对设备身份标识和认证信息提前规划；</li><li>应用侧无需逐一注册设备信息；</li><li>降低产线管理和维护成本。</li></ul> |

## 整体流程

通过注册组实现设备免注册极简接入平台操作流程图如下图所示。

图 5-18 注册流程

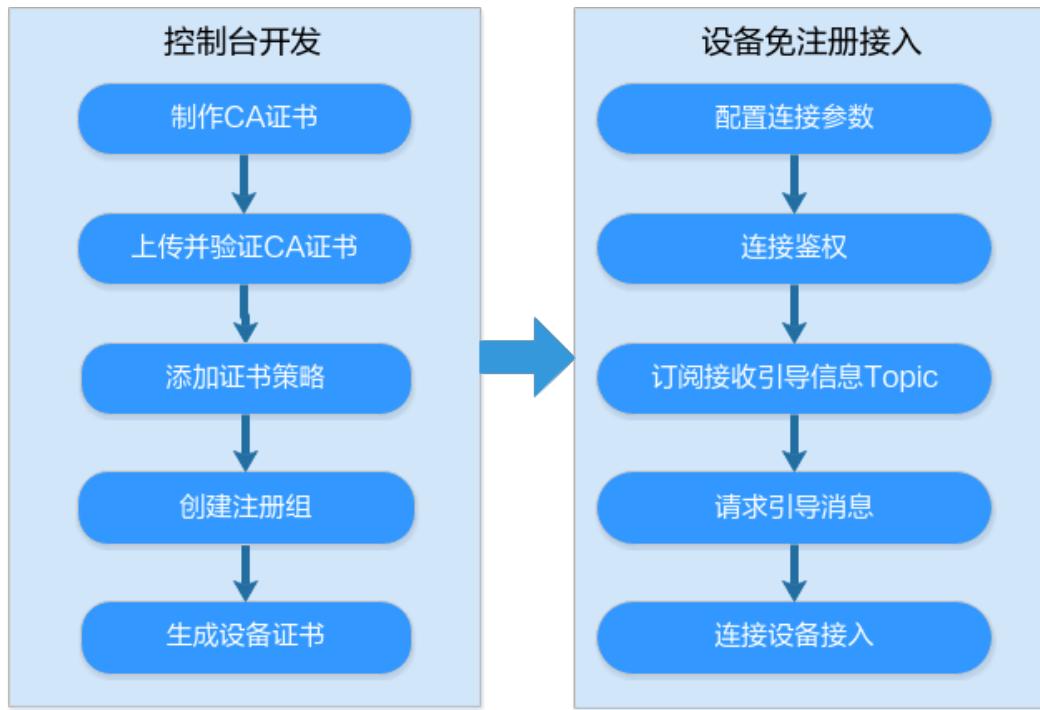
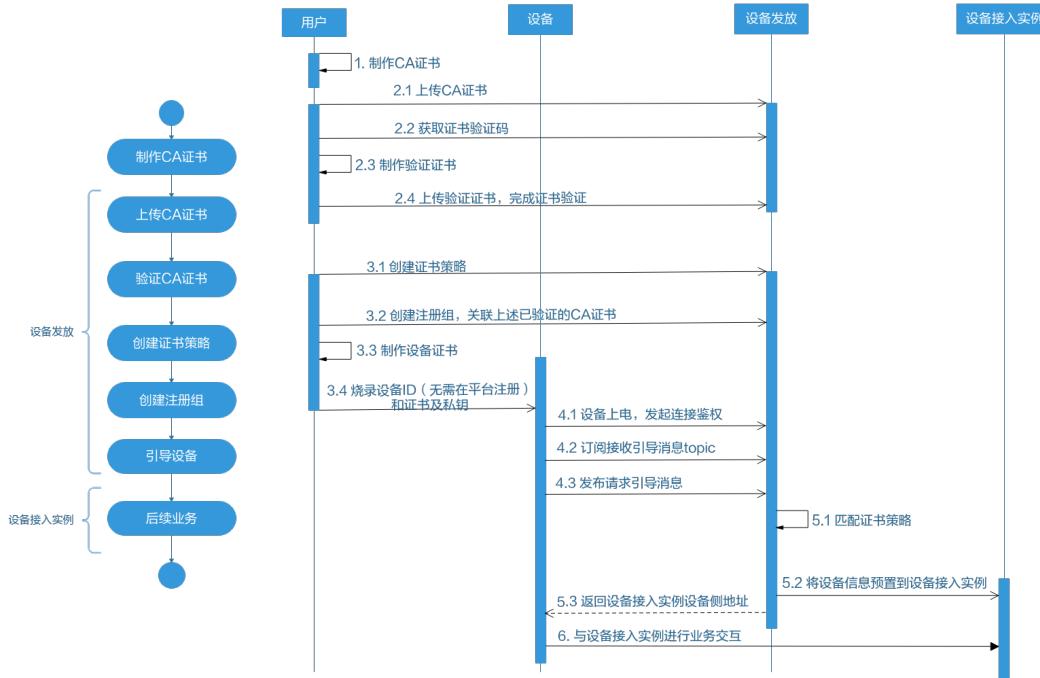


图 5-19 接入平台操作流程



## 制作 CA 证书

本文以Windows环境为例，介绍通过OpenSSL工具制作CA证书和验证证书的方法。

**⚠ 注意**

以下“生成密钥对 (rootCA.key)”和“生成CA证书 (rootCA.crt)”为操作过程中需要使用到的两个文件。

具体步骤：

**步骤1** 在浏览器中访问[这里](#)，下载并进行安装OpenSSL工具，安装完成后配置环境变量。

**步骤2** 在 D:\certificates 文件夹下，以管理员身份运行cmd命令行窗口。

**步骤3** 生成密钥对 (rootCA.key)：

**□ 说明**

生成“密钥对”时输入的密码在生成“证书签名请求文件”、“CA证书”，“验证证书”以及“设备证书”时需要用到，请妥善保存。

```
openssl genrsa -des3 -out rootCA.key 2048
```

**步骤4** 使用密钥对生成证书签名请求文件：

**□ 说明**

生成证书签名请求文件时，要求填写证书唯一标识名称 (Distinguished Name, DN) 信息，参数说明如下[表2](#)所示。

**表 5-2 参数说明**

| 提示                                              | 参数名称                | 取值样例                          |
|-------------------------------------------------|---------------------|-------------------------------|
| Country Name (2 letter code) []:                | 国家/地区               | CN                            |
| State or Province Name (full name) []:          | 省/市                 | GuangDong                     |
| Locality Name (eg, city) []:                    | 城市                  | ShenZhen                      |
| Organization Name (eg, company) []:             | 组织机构 (或公司名)         | Huawei Technologies Co., Ltd. |
| Organizational Unit Name (eg, section) []:      | 机构部门                | Cloud Dept.                   |
| Common Name (eg, fully qualified host name) []: | CA名称 (CN)           | Huawei IoTDP CA               |
| Email Address []:                               | 邮箱地址                | /                             |
| A challenge password []:                        | 证书密码，如您不设置密码，可以直接回车 | /                             |
| An optional company name []:                    | 可选公司名称，如您不设置，可以直接回车 | /                             |

```
openssl req -new -key rootCA.key -out rootCA.csr
```

**步骤5 生成CA证书 ( rootCA.crt ) :**

```
openssl x509 -req -days 50000 -in rootCA.csr -signkey rootCA.key -out rootCA.crt
```

**□ 说明**

“-days”后的参数值指定了该证书的有效天数，此处示例为50000天，您可根据实际业务场景和需要进行调整。

----结束

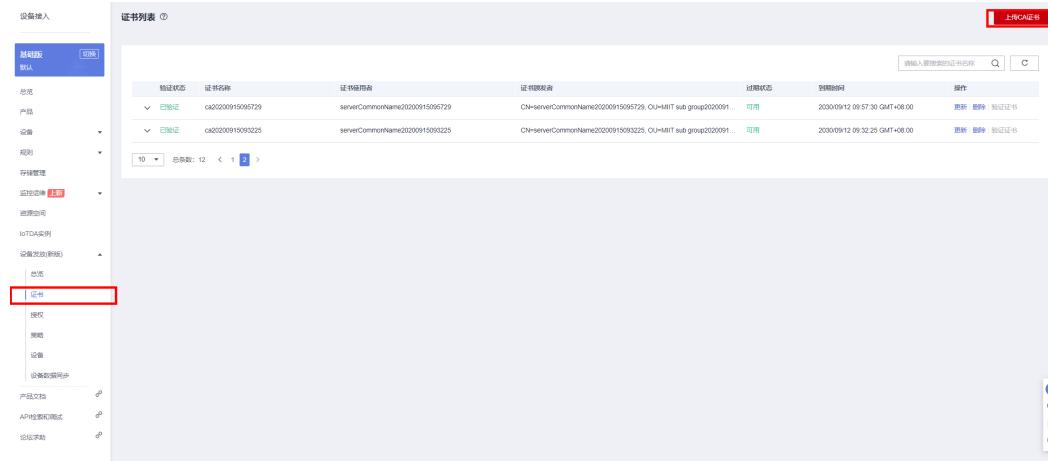
## 上传 CA 证书

### 具体操作步骤

**步骤1 登录设备发放控制台。**

**步骤2 在设备发放控制台，左侧导航窗格中，选择“证书”，单击右上方的“上传CA证书”。**

**图 5-20 上传 CA 证书**



**步骤3 在“上传CA证书”页面，填写“证书名称”，单击“添加文件”，上传此前“[制作CA证书](#)”步骤中生成的“CA证书 (rootCA.crt文件)”，单击“确定”。**

**图 5-21 上传 CA 证书详情页**



----结束

**□ 说明**

上传的CA证书初始状态为“未验证”，需要完成“[验证CA证书](#)”过程，方可正常使用该CA证书。

表 5-3 证书状态表

| CA证书状态 | 说明                                  |
|--------|-------------------------------------|
| 已验证    | 可正常使用。                              |
| 未验证    | 不可正常使用，待验证通过后，方可正常使用。               |
| 已过期    | CA证书已过期，需更新，但不影响平台使用该CA证书验证对应的设备证书。 |
| 即将过期   | CA证书30天内即将过期，需及时更新。                 |

## 验证 CA 证书

对于已上传的CA证书，平台要求用户完成“验证CA证书”过程，以验证用户具备该CA证书的签发能力。

### 操作步骤

- 步骤1 登录**设备发放控制台**。
- 步骤2 在设备发放控制台，左侧导航窗格中，选择“证书”，单击“证书列表”条目的操作栏中的“验证证书”。

图 5-22 上传 CA 证书完成页

- 步骤3 在上传验证证书页面，单击“生成验证码”，单击“复制图标”复制此CA证书的随机验证码。

图 5-23 复制验证码

## 说明

CA证书验证码有效期为一天，请及时使用验证码生成验证证书并完成验证。

验证码的生成为替换机制，即对于一个CA证书，即使此前的验证码未过期，也将被新生成的验证码替换。

### 步骤4 使用OpenSSL工具为验证证书生成密钥对。

```
openssl genrsa -out verificationCert.key 2048
```

### 步骤5 利用此验证码生成证书签名请求文件CSR。

```
openssl req -new -key verificationCert.key -out verificationCert.csr
```

## 说明

CSR文件的Common Name (e.g. server FQDN or YOUR name) 需要填写此验证码。

### 步骤6 使用CA证书、CA证书私钥和上一步骤中生成的CSR文件创建验证证书 ( verificationCert.crt )。

```
openssl x509 -req -in verificationCert.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -out verificationCert.crt -days 36500 -sha256
```

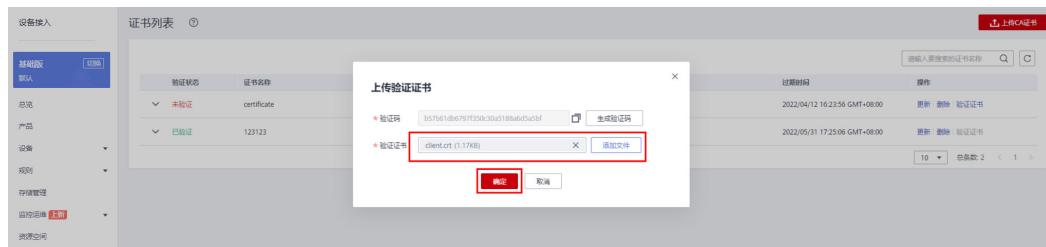
## 说明

生成验证证书用到的“rootCA.crt”和“rootCA.key”这两个文件，为“[制作CA证书](#)”中所生成的两个文件。

“-days”后的参数值指定了该证书的有效天数，此处示例为36500天，您可根据实际业务场景和需要进行调整。

### 步骤7 上传验证证书进行验证。

图 5-24 上传验证证书

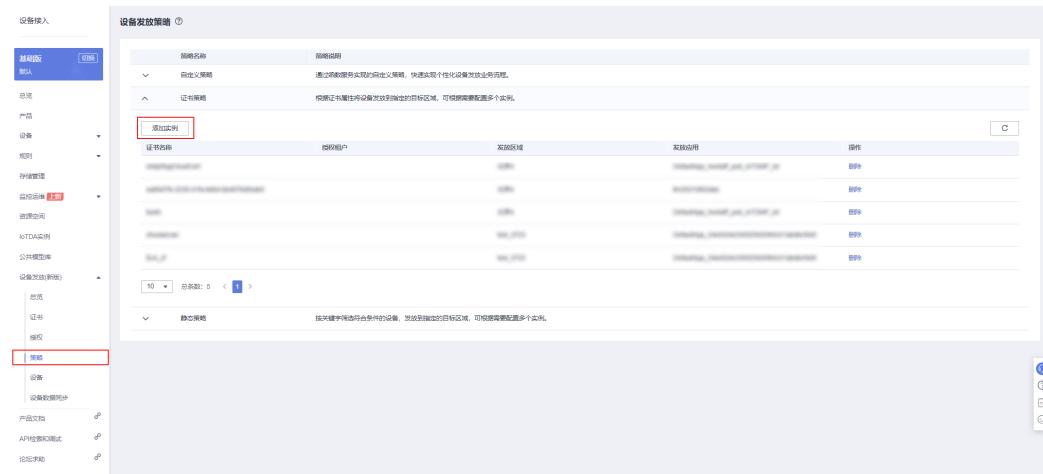


----结束

## 添加证书策略

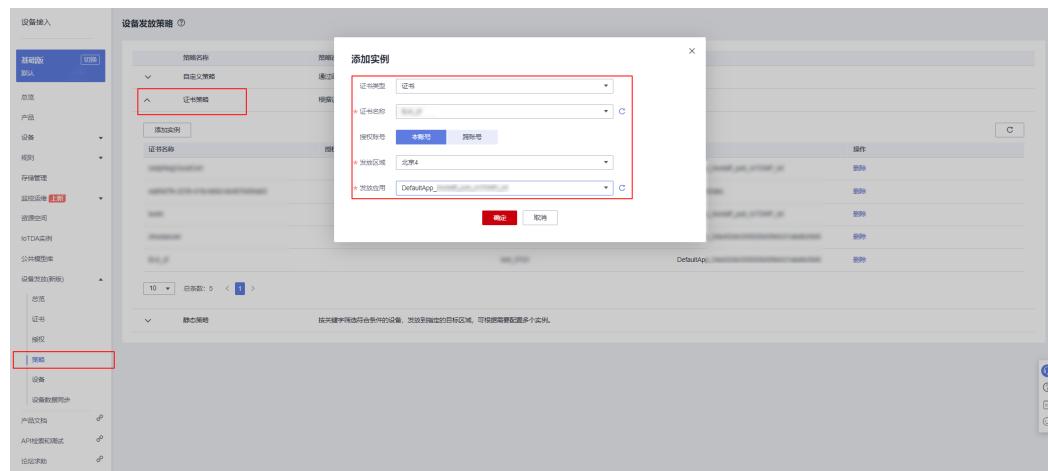
### 步骤1 添加证书策略，发放CA证书到指定的IoTDA，并且由此CA签发的设备证书都会发放到指定的IoTDA。

图 5-25 添加证书策略



步骤2 进入“策略”界面，单击展开“证书策略”，单击“添加实例”。

图 5-26 添加证书策略详情



步骤3 按照下方参数说明填写关键参数信息后，单击“确定”。

表 5-4 参数信息

| 参数名称 | 说明                                                                                                | 示例                                                                                                                                                                |
|------|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 证书名称 | 即所要根据证书属性将设备发放到指定的目标区域，选择对应的证书。                                                                   | 将需要通过证书“certificates”发放的设备发放至华北-北京四的物联网平台。 <ul style="list-style-type: none"><li>需通过证书“certificates”发放的设备：WaterMeter-Beijing0001、WaterMeter-Beijing0002</li></ul> |
| 发放区域 | 发放到指定区域后，设备将接入对应区域的设备接入服务。<br>所选区域未开通设备接入服务时，如果确定添加实例，系统将自动为您开通设备接入服务。不同区域设备接入服务价格不同，收费详情请参考价格说明。 | <ul style="list-style-type: none"><li>证书名称：certificates</li><li>发放区域：华北-北京四</li><li>发放应用：beijing-app1</li></ul>                                                   |
| 发放应用 | 选择对应设备接入服务区域已创建的应用。在物联网平台中，设备由应用统一管理。<br>如果对应设备接入服务区域未创建应用，需要前往对应服务创建应用。                          |                                                                                                                                                                   |

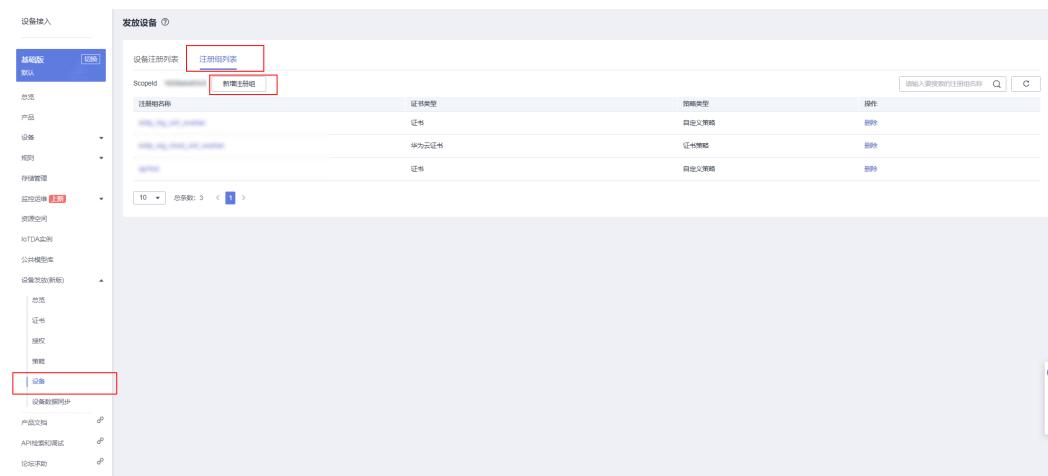
----结束

## 创建注册组

MQTT证书接入的设备，可以在设备发放创建一个注册组，绑定对应的CA证书和自定义策略，可以实现批量设备的自注册，实现设备一键上电即可上云的动作，可在注册组详情中查看该注册组下所有的设备。

**步骤1** 进入“设备-注册组”界面，单击右上角“新增注册组”。

图 5-27 新增注册组



**步骤2** 按照下方参数说明填写关键参数信息后，完成创建。

表 5-5 参数信息

| 参数名称  | 说明                                     |
|-------|----------------------------------------|
| 注册组名称 | 注册组的唯一标识。                              |
| 选择证书  | 用于和注册组绑定，同一个证书只能同时绑定一个注册组，不能同时绑定多个注册组。 |
| 发放策略  | 当前只支持“自定义策略”，同时需要选择所要运行的函数。            |

----结束

## 生成设备证书

**步骤1** 使用OpenSSL工具为设备证书生成密钥对（设备私钥）：

```
openssl genrsa -out deviceCert.key 2048
```

**步骤2** 使用设备密钥对，生成证书签名请求文件：

```
openssl req -new -key deviceCert.key -out deviceCert.csr
```

### 说明

生成证书签名请求文件时，要求填写证书唯一标识名称（Distinguished Name，DN）信息，参数说明如下[表6](#)所示。

表 5-6 参数说明

| 提示                                              | 参数名称                | 取值样例                          |
|-------------------------------------------------|---------------------|-------------------------------|
| Country Name (2 letter code) []:                | 国家/地区               | CN                            |
| State or Province Name (full name) []:          | 省/市                 | GuangDong                     |
| Locality Name (eg, city) []:                    | 城市                  | ShenZhen                      |
| Organization Name (eg, company) []:             | 组织机构（或公司名）          | Huawei Technologies Co., Ltd. |
| Organizational Unit Name (eg, section) []:      | 机构部门                | Cloud Dept.                   |
| Common Name (eg, fully qualified host name) []: | CA名称（CN）            | Huawei IoTDP CA               |
| Email Address []:                               | 邮箱地址                | /                             |
| A challenge password []:                        | 证书密码，如您不设置密码，可以直接回车 | /                             |
| An optional company name []:                    | 可选公司名称，如您不设置，可以直接回车 | /                             |

**步骤3 使用CA证书、CA证书私钥和CSR文件创建设备证书（deviceCert.crt）。**

```
openssl x509 -req -in deviceCert.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -out deviceCert.crt -days 36500 -sha256
```

**□ 说明**

生成设备证书用到的“rootCA.crt”和“rootCA.key”这两个文件，为“制作CA证书”中所生成的两个文件，且需要完成“上传并验证CA证书”。

“-days”后的参数值指定了该证书的有效天数，此示例为36500天，您可根据实际业务场景和需要进行调整。

----结束

## 设备免注册接入

**步骤1 下载并修改华为SDK示例代码进行设备引导（这里以[java sdk代码](#)为例）。****□ 说明**

使用IDEA/Eclipse打开SDK代码工程，修改iot-device-demo目录下的DEMO示例“BootstrapSelfRegSample”中的参数。

1. ScopeId从设备发放的注册组页面中获取；
2. deviceId由客户自主规划且未在平台注册（需注意平台要求deviceId全局唯一）；
3. 设备证书指定为“控制台开发”中生成的设备证书；
4. bootstrapUri为上述[终端节点](#)。

**图 5-28 修改 demo 示例**

```
/**
 * 演示自注册场景（证书方式），设备启动时，通过引导服务获取真实的服务器地址
 */
public class BootstrapSelfRegSample extends BaseBootstrapSample {
 /**
 * ScopeID，与租户相关，请从注册组页面获取
 */
 private static String scopeId = "[Please input your scope id here, example:f67f8df43c4a]";

 /**
 * 设备ID（自注册场景下，设备ID无需提前在设备发放上注册）
 */
 private static String deviceId = "[Please input your device id here, example:702b1038-a174-4a1d-969f-f67f8df43c4a]";

 /**
 * 设备证书信息
 */
 private static String DEVICE_CERT
 = "[Please input your device cert path here, example:D:\\SDK\\cert\\deviceCert.pem]";

 private static String DEVICE_CERT_KEY
 = "[Please input your device cert key path here, example:D:\\SDK\\cert\\deviceCert.key]";

 private static String DEVICE_CERT_KEY_PWD
 = "[Please input your device cert key pwd here, example:yourpwd. If not set, input empty string]";

 public static void main(String[] args) throws Exception {
 // 读取pem格式设备证书
 KeyStore keyStore = CertificateUtil.getKeyStore(DEVICE_CERT, DEVICE_CERT_KEY, DEVICE_CERT_KEY_PWD);

 // 创建引导客户端，发起引导
 BootstrapClient bootstrapClient = new BootstrapClient(BOOTSTRAP_URL, deviceId, keyStore, DEVICE_CERT_KEY_PWD, scopeId, PLATFORM_CA_PROVIDER);
 bootstrapClient.bootstrap(new SimpleBootstrapActionListener(bootstrapClient));
 }
}
```

**步骤2 运行DEMO程序，看到如下日志，代表设备发放成功，并且已经收到设备发放下发的设备接入地址。如果程序运行正常，在对应的设备接入实例可以看到该设备，且该设备已在线。**

图 5-29 日志信息

```
2021-01-29 14:46:47 INFO BootstrapClient:50 - create BootstrapClient: 60136b0682401c03e0b1ccf5_aaa_device2
2021-01-29 14:46:48 INFO MqttConnection:167 - try to connect to ssl://100.95.158.64:8883
2021-01-29 14:46:48 INFO MqttConnection:200 - connect success ssl://100.95.158.64:8883
2021-01-29 14:46:48 INFO MqttConnection:105 - Mqtt client connected. address :ssl://100.95.158.64:8883
2021-01-29 14:46:48 INFO MqttConnection:240 - publish message topic = $oc/devices/60136b0682401c03e0b1ccf5_aaa_device2/sys/bootstrap/up, msg =
2021-01-29 14:46:48 INFO DefaultBootstrapActionListener:30 - bootstrap success:$oc/devices/60136b0682401c03e0b1ccf5_aaa_device2/sys/bootstrap/down
2021-01-29 14:46:49 INFO DefaultBootstrapActionListener:30 - bootstrap success:null
2021-01-29 14:46:49 INFO MqttConnection:85 - messageArrived topic = $oc/devices/60136b0682401c03e0b1ccf5_aaa_device2/sys/bootstrap/down, msg = {"address":"100.95.174.182:1883","initConfig":null}
2021-01-29 14:46:49 INFO BootstrapClient:120 - bootstrap ok address:100.95.174.182:1883
2021-01-29 14:46:49 INFO DefaultBootstrapActionListener:30 - bootstrap success:100.95.174.182:1883
```

**步骤3** 收到设备发放下发的设备接入地址后，需要关闭设备侧的设备发放的连接，用新的URL地址与设备接入通信，进行相关业务。

图 5-30 关闭连接示例代码

```
@Override
public void onSuccess(Object context) {
 // 引导成功，获取到iot平台的地址
 String address = (String) context;
 log.info("bootstrap success, the address is {}", address);

 // 引导成功后关闭客户端
 bootstrapClient.close();

 // 与iot平台建立连接，上报消息
 IoTDevice device = bootstrapClient.getIoTDevice(serverUri: "ssl://" + address);
 if (device == null || device.init() != 0) {
 return;
 }
 device.getClient().reportDeviceMessage(new DeviceMessage("hello"), listener: null);
}
```

## 说明

说明：

1. DeviceID即设备ID，用于唯一标识一个设备。设备需使用未注册的DeviceID进行免注册接入，不同设备的DeviceID不同。
2. 在客户实际业务场景中，推荐一个设备使用一个设备证书，不建议多个设备共用一个设备证书。

----结束