

智能设施云平台

开发指南

文档版本 01
发布日期 2025-08-19



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，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。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目 录

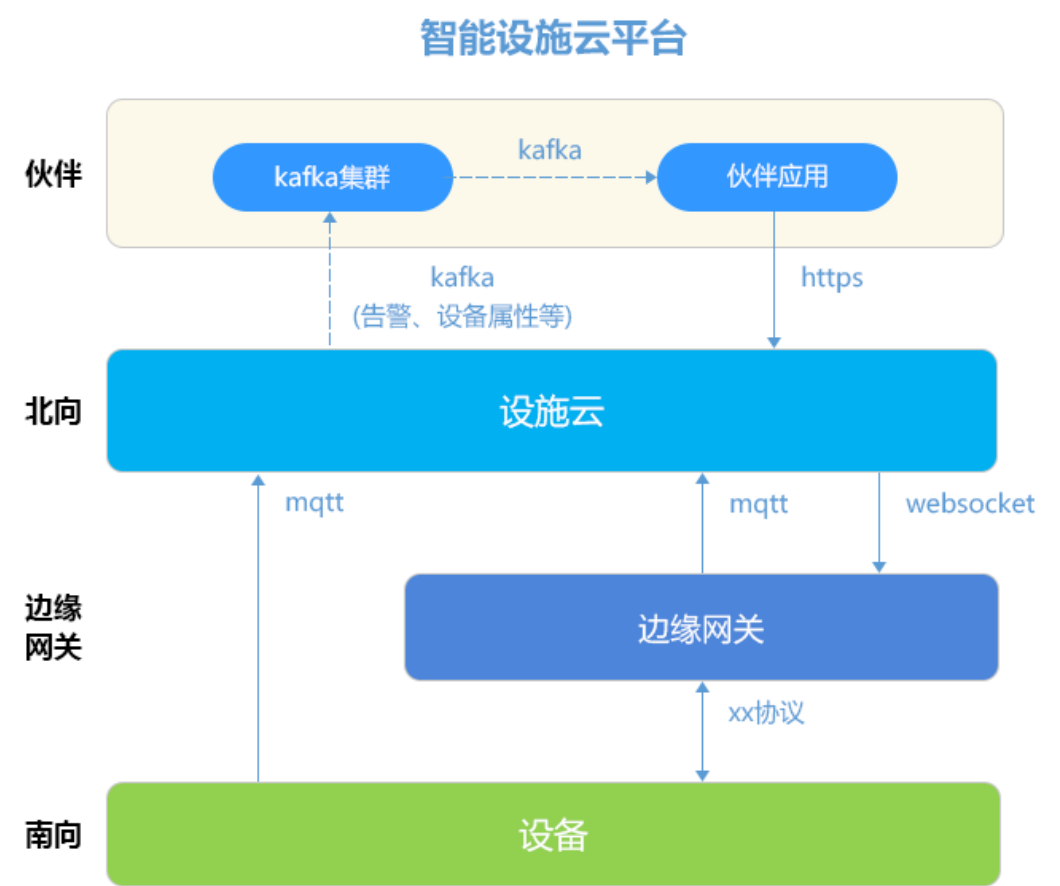
1 开发前必读.....	1
2 北向应用集成.....	3
2.1 北向 API 对接.....	3
2.2 集成组态页面.....	7
2.2.1 集成系统或空间组态页面步骤.....	7
2.2.1.1 集成方案.....	7
2.2.1.2 获取组态地址.....	8
2.2.1.3 获取会话 Ticket.....	10
2.2.1.4 组态页面通过单点登录嵌入.....	10
2.2.2 组态事件开发.....	11
2.2.2.1 组态页发送事件.....	11
2.2.2.1.1 对外发送事件.....	11
2.2.2.1.2 设备类事件.....	15
2.2.2.1.3 空间/系统类事件.....	20
2.2.2.1.4 执行跳转事件.....	20
2.2.2.2 组态页监听事件.....	23
2.2.2.2.1 空间/系统组态类事件.....	23
2.2.2.2.2 退出登录事件.....	24
2.3 数据推送.....	24
2.3.1 推送方案.....	24
2.3.2 推送步骤.....	25
2.3.3 伙伴订阅推送数据.....	27
3 南向设备接入.....	29
3.1 整体接入架构.....	29
3.2 MQTT/MQTTS 接入方案.....	29
3.2.1 设备侧 MQTT/MQTTS 接口参考.....	30
3.2.1.1 使用前必读.....	30
3.2.1.2 通信方式概述.....	31
3.2.1.3 Topic 定义.....	32
3.2.1.4 设备连接鉴权.....	34
3.2.1.5 设备命令.....	36
3.2.1.5.1 平台命令下发.....	36

3.2.1.6 设备消息.....	38
3.2.1.6.1 设备消息上报.....	38
3.2.1.6.2 平台消息下发.....	39
3.2.1.7 设备属性.....	41
3.2.1.7.1 设备属性上报.....	41
3.2.1.7.2 网关批量设备属性上报.....	42
3.2.1.7.3 平台设置设备属性.....	44
3.2.1.7.4 平台查询设备属性.....	46
3.2.1.7.5 设备侧获取平台的设备影子数据.....	48
3.2.1.8 网关与子设备管理.....	50
3.2.1.8.1 平台通知网关子设备新增.....	50
3.2.1.8.2 平台通知网关子设备删除.....	53
3.2.1.8.3 网关同步子设备列表.....	55
3.2.1.8.4 网关更新子设备状态.....	56
3.2.1.8.5 网关更新子设备状态响应.....	58
3.2.1.8.6 网关新增子设备请求.....	61
3.2.1.8.7 网关新增子设备请求响应.....	64
3.2.1.8.8 网关删除子设备请求.....	67
3.2.1.8.9 网关删除子设备请求响应.....	69
3.2.2 基于 IoTDA 的接入方式.....	71
3.2.2.1 接入方案.....	72
3.2.2.2 接入方式.....	73
3.2.3 基于 IoTEdge 的接入方式.....	73
3.2.3.1 接入方案.....	74
3.2.3.2 接入方式.....	75
3.3 数采插件接入方案.....	75
3.3.1 接入方案.....	76
3.3.2 接入案例.....	77
3.4 模拟设备测试.....	86
3.4.1 使用 mqttfx 模拟物联网设备.....	86
3.4.2 使用 mqttfx 模拟机电设备.....	92

1

开发前必读

基于华为自身业务需求打造设施及能源管理综合解决方案，涉及使用智能设施云平台服务开发的部分在架构上涉及部分为：



开发操作	开发说明
北向应用开发	主要为智能设施云平台应用侧的集成对接开发，包括API接口调用、组态页面集成和业务数据推送。

开发操作	开发说明
南向设备开发	主要为设备与智能设施云平台的接入，包括设备接入、业务数据上报和对平台下发控制命令的处理。

业务概览

智能设施云平台采用公有云SaaS方式部署，面向不同租户提供统一的应用功能。通过边缘网关和控制器实现各租户园区机电设备的统一上云。

- 北向应用集成：以智能设施云平台为底座进行快速构建，为设备设施统一接入、统一服务、统一运维提供支撑，通过API总线、数据总线、SDK/工具等提供集成开放能力。
- 南向设备接入：通过MQTT/MQTTS和数采插件接入方案完成南向设备接入，并可以在没有真实设备的情况下，通过mqttfx模拟工具来模拟设备的在线和点位上报等功能来模拟设备测试。

注意

- 1、伙伴严禁使用测试环境去对称设施云的现网环境的API和KAFKA，伙伴测试环境只能对接设施云的POC测试环境，伙伴只能使用生产环境对应设施云生产环境。
- 2、伙伴对接设施云API接口的账号要自己创建一个机机账号，而不要和前端页面登录的人机账号共用。

2 北向应用集成

2.1 北向 API 对接

智能设施云平台以aPaaS模式构筑设施核心能力，对客户应用提供API、数据开放能力，实现客户应用新业务快速上线。您可以查看《API参考》文档，了解如何使用API对智能设施云平台核心数据进行相关操作，如创建、删除、查看等。

调用说明

智能设施云平台应用侧（又名“北向”）提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用。

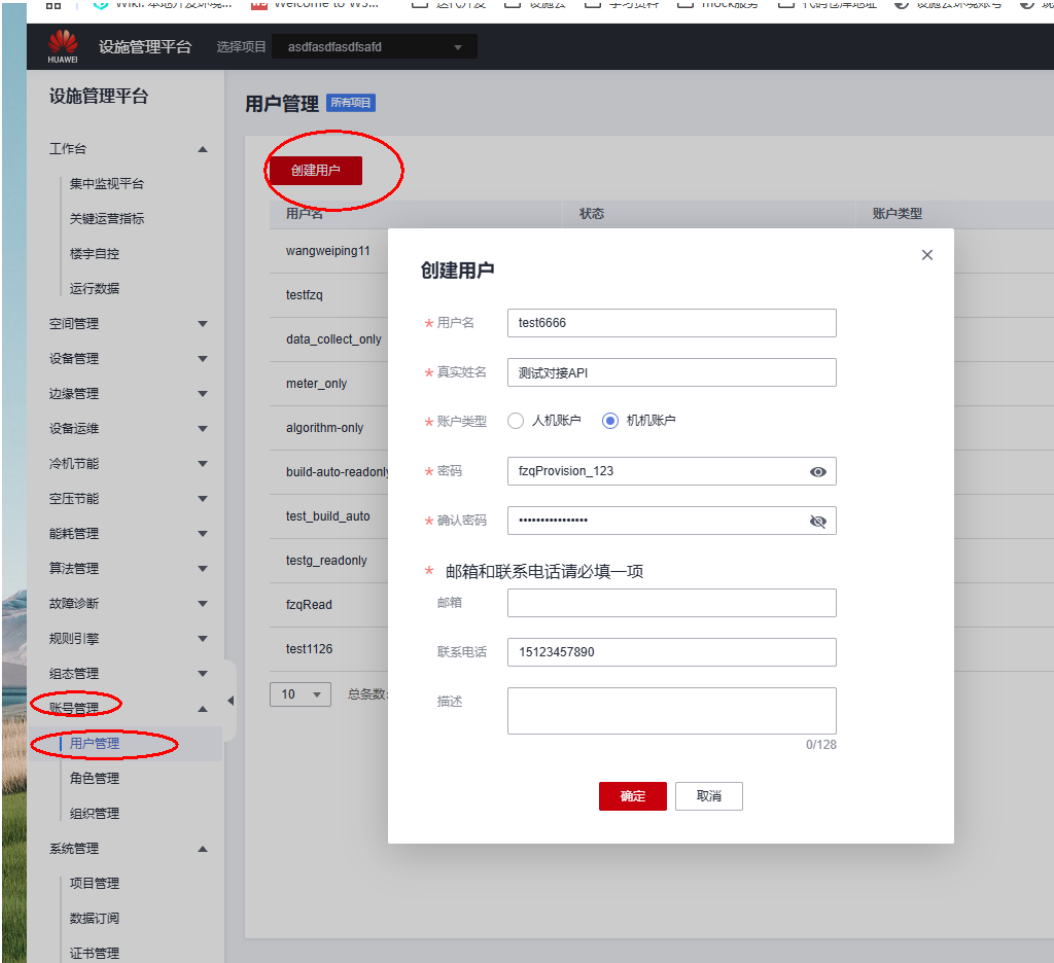
约束与限制

- 您能创建的资源的数量与配额有关系。
- 更详细的限制请参见具体API的说明。

基本概念

- 用户
在设施aPaaS中创建的用户，是设施aPaaS平台的使用人员，具有身份凭证（密码）。
通常在调用API的鉴权过程中，您需要用到用户名和密码信息。
拿到对应项目的设施云前端用户和密码后登录到设施云前端页面后，参考下面步骤创建一个机机账号用于对接API接口：
1、找到账号管理菜单下的用户管理，单击“创建用户”，账号类型要选择机机账户，邮箱和联系电话填写一个。

图 2-1 创建机机账号



2、单击“确定”成功后会弹出下面提示，继续单击“确认”。

图 2-2 确认页面



3、接着会跳转到编辑用户页面，为该用户的角色选择“te_admin”后单击“确定”，这样一个机机账号就创建完成了，后面调用生成token接口请求体里面name字段值对应本次机机账号的用户名，password字段值对应本次机机账号的密码。

图 2-3 绑定角色

The screenshot shows a modal window titled "编辑用户" (Edit User). It contains the following fields and controls:

- 用户名 (Username): test6666
- ★ 真实姓名 (Real Name): 测试对接API
- ★ 账户类型 (Account Type): 机机账户
- 状态 (Status): ☒ 启用 ☐ 停用
- ★ 角色 (Role): te_admin (with a dropdown arrow)
- ★ 邮箱和联系电话请必填一项 (Email and Contact Number are required):
 - 邮箱 (Email): [empty field]
 - 联系电话 (Contact Number): 15123457890
 - 描述 (Description): [empty field, with 0/128 character count]

At the bottom, there are two buttons: "确定" (Confirm) in red and "取消" (Cancel) in white.

- 实例

客户开通设施aPaaS后，对应开通一个实例。设施aPaaS核心资源（包括用户）存储都在实例下，不同实例的资源相互独立，不支持跨实例类型的资源。在API中不同实例对应的实例ID不同，登录设施云前端成功后，可以鼠标放在右上角的用户后出现查看实例信息单击，就可以获取到当前用户的实例ID，**实例ID对应的就是API接口中请求头的X-Instance-Id这个字段**（通常不同的交付局点对应这个实例ID不同）。

图 2-4 查看实例信息



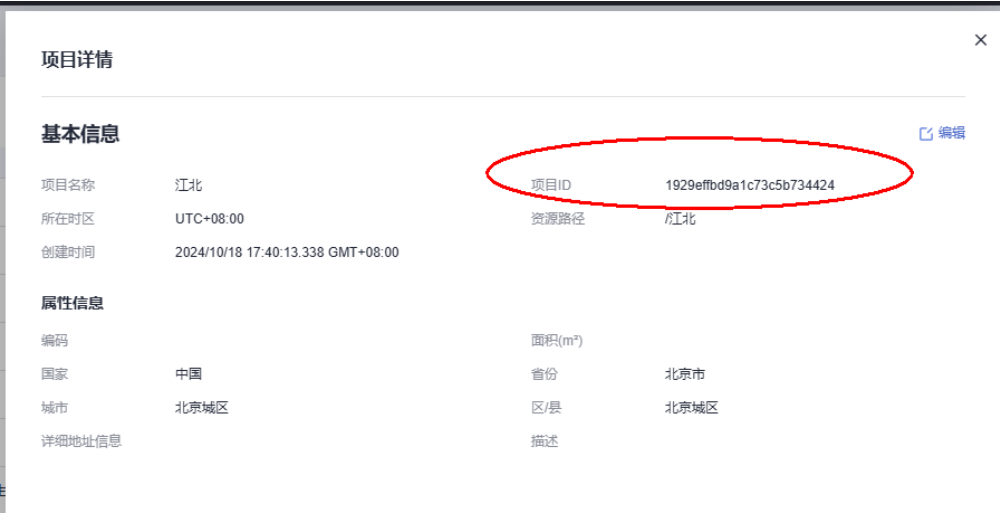
- 项目
设施云针对不同的项目做了二级资源隔离（比如在不同项目下创建的设备只能在该项目中查看对应项目的设备）。在设施云前端登录成功后，可以在“系统管理-》项目管理”菜单下查看当前用户的所有项目，如下图所示：

图 2-5 查看项目列表



单击对应项目详情就可以查看对应项目的项目ID，项目ID对应的就是API接口中请求头的X-Root-Zone-Id这个字段。也可以通过查看API文档中项目管理下的查询项目列表接口获取对应项目ID。

图 2-6 查看项目 ID



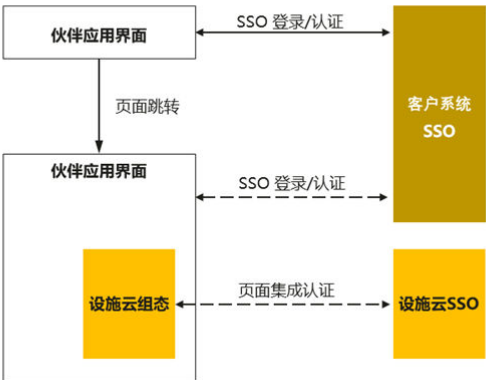
2.2 集成组态页面

2.2.1 集成系统或空间组态页面步骤

2.2.1.1 集成方案

组态页面集成方案

图 2-7 组态页面集成
组态页面集成



场景：
1. 伙伴页面集成设施云组态页面
2. 设施云组态页面支持临时会话

获取 token

参考API指导文档的Token管理的- 》生成Token章节，对应URL为post v1/iotfm/auth/tokens，那么生成Token的地址就为：对应环境API地址 + /v1/iotfm/auth/tokens（比如北京四现网环境为 https://facility-api.cn-north-4.myhuaweicloud.com/v1/iotfm/auth/tokens）

对应请求头里面的X-Instance-Id使用上面前端获取的实例ID。

请求体格式如下，其中name字段填写设施云前端登录用户对应的用户名，password字段填写该用户的密码。

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "xxx",
          "password": "xxxxx"
        }
      }
    }
  }
}
```

发送成功后响应头的X-Subject-Token字段对应的值就是接口鉴权需要的Token，也就是接口文档中头部的X-Auth-Token字段。**Token的有效时间为一天，建议获取该Token后做缓存，并定时刷新以避免过期（如每小时刷新一次）。**

Body	Cookies	Headers (14)	Test Results
KEY	VALUE		
Date ①	Thu, 20 Feb 2025 02:53:34 GMT		
Content-Type ①	application/json		
Transfer-Encoding ①	chunked		
Connection ①	keep-alive		
Content-Security-Policy ①	default-src 'self'		
X-Request-Id ①	6a0a8cdaee4d42b0b479c35e52c9b128-1740020014576-iotf-system-mgr-5bc		
X-Frame-Options ①	SAMEORIGIN		
X-Download-Options ①	noopen		
Strict-Transport-Securit ①	max-age=31536000;includeSubDomains		
Cache-Control ①	no-cache, no-store, max-age=0, must-revalidate		
X-XSS-Protection ①	1; mode=block		
X-Content-Type-Options ①	nosniff		
X-Subject-Token ①	JCNJb1RTdGFnZTEyQ=eyJhbGciOiJSUzUxMiJ9.eyJ0b2t1b1l6eyJ0b2t1b1l6eyJk		
Server ①	elb		

2.2.1.2 获取组态地址

操作步骤

步骤1 获取系统组态地址

参考API指导文档的设备系统管理-》查询设备系统列表章节，对应URL为GET /v1/iotfm/ba/device-systems，那么查询设备系统的地址就为：对应环境API地址 + /v1/

iotfm/ba/device-systems（比如北京四现网环境为 <https://facility-api.cn-north-4.myhuaweicloud.com/v1/iotfm/ba/device-systems>）。

根据文档填写请求中的实例ID和Token以及相关请求参数查询出结果，响应结果中 **ba_config_url** 字段返回的结果对应的就是系统组态的地址。

参数	参数类型	描述
selected_product_properties	Array of ProductPropertiesData objects	系统选择的产品及对应属性列表配置信息，当display_mode为LIST类型时该字段必填。JSON格式，整体是一个数组对象，数组中每个元素是一个对象，每个对象表示系统中的某个产品需要展示的属性列表，一个产品下最多选择10个服务，每个服务最多选择10个属性展示
ba_config_url	String	组态图页面地址,以设施aPaaS Console域名开头，长度不超过255。当display_mode为DIAGRAM类型时该字段必填。
sort_num	Integer	排序，系统在同一设备类型同一展示形式不同系统的展示顺序
remark	String	备注，描述信息。
create_time	String	创建时间，格式为："yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"。
update_time	String	最后一次修改时间，格式为："yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"。

步骤2 获取空间组态地址

参考API指导文档的组态配置管理-》查询空间平面组态章节，对应URL为GET /v1/iotfm/ba/zone-configurations/match/{zone_id}，那么查询空间平面组态的地址就为：对应环境API地址 + /v1/iotfm/ba/zone-configurations/match/{zone_id}（比如北京四现网环境为 https://facility-api.cn-north-4.myhuaweicloud.com/v1/iotfm/ba/zone-configurations/match/%7Bzone_id%7D）。

根据文档填写请求中的实例ID、项目ID和Token以及相关请求参数zone_id（对应空间）查询出结果，响应结果中 **ba_config_url** 字段返回的结果对应的就是空间组态的地址。

参数	参数类型	描述
ba_config_url	String	组态图页面地址,以设施aPaaS Console域名开头，长度不超过255。当display_mode为DIAGRAM类型时该字段必填。

步骤3 获取设备组态地址

参考API指导文档的组态配置管理-》查询单个设备组态章节，对应URL为GET /v1/iotfm/ba/device-configurations/match/{device_id}，那么查询空间平面组态的地址就为：对应环境API地址 + /v1/iotfm/ba/device-configurations/match/{device_id}（比如北京四现网环境为 [/v1/iotfm/ba/device-configurations/match/{device_id}](#)）。

根据文档填写请求中的实例ID、项目ID和Token以及相关请求参数device_id（对应设备ID）查询出结果，响应结果中ba_config_url字段返回的结果对应的就是设备组态的地址。

zone_id	String	参数说明：区域ID。 取值范围：长度不超过24，只允许字母、数字的组合。
zone_name	String	参数说明：区域名称。 取值范围：长度不超过64。
ba_config_url	String	组态图页面地址,以设施aPaaS Console域名开头，长度不超过255。当display_mode为DIAGRAM类型时该字段必填。

----结束

2.2.1.3 获取会话 Ticket

参考API指导文档的SSO管理- 》创建session-ticket章节，对应URL为POST /v1/iotfm/sso/session-tickets，那么获取tickets的地址就为：对应环境API地址 + /v1/iotfm/sso/session-tickets（比如北京四现网环境为 https://facility-api.cn-north-4.myhuaweicloud.com/v1/iotfm/sso/session-tickets）。

根据文档填写请求中的实例ID和Token，响应结果中session_ticket就是需要获取的会话Ticket，expire_time是该会话Ticket过期时间。

表 2-2 响应 Body 参数

参数	参数类型	描述
session_ticket	String	参数说明：会话Ticket。 取值范围：长度1-1024。
expire_time	String	参数说明：失效时间，格式为yyyy-MM-dd'T'HH:mm:ss.SSS'Z'。 取值范围：长度固定为24。

 注意

当前这个会话Ticket过期时间很快，不要缓存使用，每次跳转组态页面前先调这个接口获取最新的Ticket。

2.2.1.4 组态页面通过单点登录嵌入

参考API指导文档的SSO管理- 》基于session-ticket 单点登录章节，对应URL为GET /v1/iotfm/sso/sign-in，那么获取单独登录的地址就为：对应环境前端地址 + /v1/iotfm/sso/session-tickets（比如北京四现网环境为 https://facility-console.cn-north-4.huaweicloud.com/v1/iotfm/sso/sign-in）。

伙伴在前端页面嵌入的页面就可以调用上面单独登录的地址，请求参数中session-ticket就为上面获取的会话Ticket, target填写上面获取的组态地址（**ba_config_url**字段对应的值，针对设备组态嵌入时，**需要注意的是target要用设备组态的地址+&deviceNumber={device_id}才能正常显示设备组态中设备点位数据**），这样就能将对应的系统组态或空间组态或者设备组态显示在自己的页面了。

GET /v1/iotfm/sso/sign-in

表 2-3 Query 参数

参数	是否必选	参数类型	描述
session-ticket	是	String	单点登录鉴权票据
target	是	String	单点登录鉴权成功后重定向的地址

⚠ 注意

- 1、如果请求的Cookie中携带设施云会话信息（IOTFM_SESSION），且会话有效、ticket信息一致，那么会直接重定向到target地址。
- 2、对于集成设施云的平台，设施云属于第三方网站，且当前方案设施云通过iframe的方式集成。当cookie的samesite属性为lax时会被浏览器拦截，只有配置为samesite=none;secure;时，cookie才能被第三方携带。
- 3、嵌入的组态页面如果要支持点位设置等事件需要参考[组态事件开发](#)资料进行开发相关事件来实现。

2.2.2 组态事件开发

2.2.2.1 组态页发送事件

2.2.2.1.1 对外发送事件

组态事件开放，用于外部页面适配组态弹窗业务联动。

组态发布页面一般是通过iframe的方式嵌入外部页面的，其中组态弹窗分为“查询属性”和“下发指令”弹窗两类，其中需要“查询属性”弹窗时会向window发送类型为：'attrDetail'的消息；需要“下发指令”弹窗则会发送类型为'deviceControl'或者'propertyControl'的消息。因此外部页面只需在页面中监听以下两种类型的消息则可以判断需要如何显示弹窗，代码示例如下：

```
window.addEventListener("message", (event) => {
  if (event.data.type === 'deviceControl' || event.data.type === 'propertyControl') {
    this.sendCommand(event.data)
  }
  if (event.data.type === 'attrDetail') {
    this.sendDetail(event.data)
  }
})
```

操作步骤

步骤1 监听“查询属性”弹窗的消息体时，外部页面可根据如下信息查询对应设备的属性信息：

type：消息类型（attrDetail为查询属性）

property_path：服务ID/属性编码

device_id：设备ID

'http://localhost:8080'：母页面域名，是由左邻页面来监听该事件

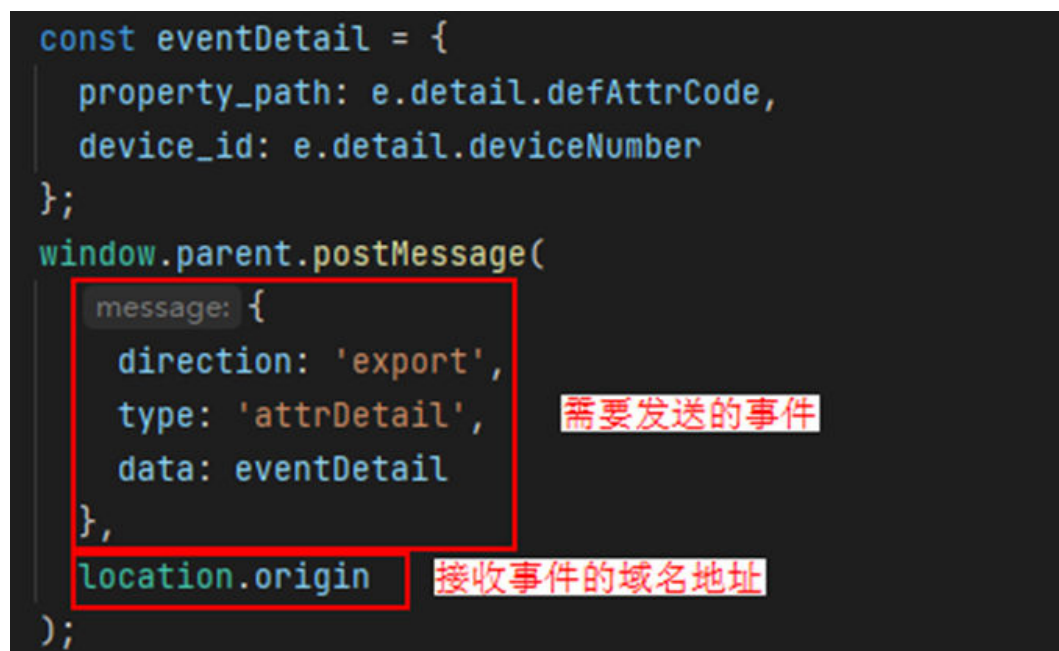
1、父页面监听message，写法固定

```
1 // 监听单个事件
2 window.addEventListener('message', function (msg) {
3     console.log(msg)
4 })
5
6 // 监听多个事件，可以使用if来判断 msg里边的参数进行区分
```

2、子页面发送 message 消息，并附带参数

```
1 // window.parent 是 iframe 子页面获取父页面的 window 对象
2
3 // 后面的 * 号就是处理跨域问题的，任何域名都不会出现跨域问题
4 window.parent.postMessage("需要传递的参数", '*')
5
6 // 也可以指定传递域名地址，这个域名不会出现跨域问题，写父页面(接收)域名地址
7 window.parent.postMessage("需要传递的参数", 'http://0.0.0.0:8080')
8 // 传递的参数可以是数组，对象，字符串等
```

组态发送事件代码示例截图：



```
const eventDetail = {
  property_path: e.detail.defAttrCode,
  device_id: e.detail.deviceNumber
};
window.parent.postMessage(
  {
    message: {
      direction: 'export',
      type: 'attrDetail',
      data: eventDetail
    },
    location.origin
  },
  'http://0.0.0.0:8080'
);
```

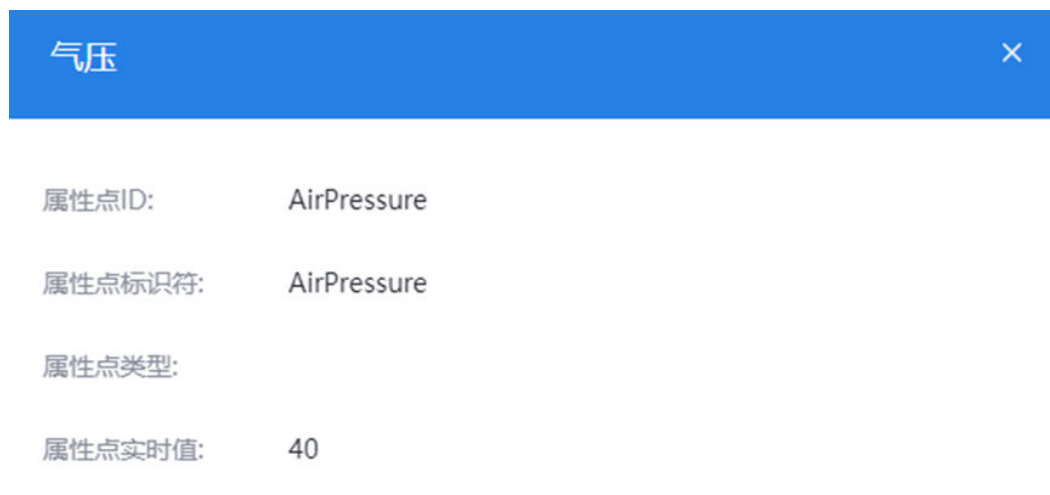
需要发送的事件

接收事件的域名地址

当type类型为attrDetail时，说明查看的是具体设备的指定属性

```
{
  direction:"export",
  type: "attrDetail",
  data: {
    property_path: "EnviromentService/AirPressure",
    device_id: "wwp_test_elec"
  }
},
"http://localhost:8080"
```

如单击组态页面中"文本增强组件"的文字内容时，此时为查询设备属性的点位信息，组态预览页的默认效果如下（发布页该弹框需由外部系统完成开发）：



步骤2 监听“下发指令”弹窗的消息体时，外部页面可根据如下信息给对应的设备属性下发指令：

type：消息类型（deviceControl为下发指令）

property_path：服务ID/属性编码

device_id：设备ID

isDeviceIcon：是否单击设备图标

'http://localhost:8080'：母页面域名，是由左邻页面来监听该事件

1. 当type类型为deviceControl时，说明控制的只是具体设备而没有指定具体属性，所以没有property_path字段。

```
{
  direction:"export",
  type: "deviceControl",
  data: {
    device_id: "wwp_test_elec"
  }
},
"http://localhost:8080"
```

如单击组态页面中的"设备组件"时，由于所选定的是具体设备没有指定具体的属性，此时在弹窗中可以自行选择设备属性的点位，组态预览页的默认效果如下（发布页该弹框需由外部系统完成开发）：

The dialog box is titled "wwp_test_elec" and includes the following controls:

- 选择点位**: A dropdown menu with the placeholder text "请选择".
- 控制**: Two radio buttons, "强制" (selected) and "解除强制".
- 当前值**: A horizontal range slider.
- 设定值**: A toggle switch currently set to "关" (Off), with "开" (On) as the alternative.
- 说明**: A text input field with the placeholder "请输入说明" and a character count "0/255".

Buttons at the bottom: "取消" (Cancel) and "确定" (Confirm).

2. 当type类型为propertyControl时，说明控制的是具体设备的指定属性，所以有property_path字段。

```
{
  direction:"export",
  type: "propertyControl",
  data: {
    property_path: "EnviromentService/AirPressure",
    device_id: "wwp_test_elec"
  }
},
"http://localhost:8080"
```

如单击组态页面中"文本增强组件"的齿轮图标时，由于已经选定了具体设备的指定属性，此时在弹窗中无法再自行选择设备属性的点位，组态预览页的默认效果如下（发布页该弹框需由外部系统完成开发）：

新风温度

控制

强制

解除强制

当前值

60.00℃

设定值

请输入设定值

℃

说明

请输入说明

0/255

取消

确定

----结束

2.2.2.1.2 设备类事件

组态页支持用户单击组态中查询详情图标或设置图标时，向window发送 "propertyDetail"事件和"deviceControl"/"propertyControl"事件，可用于母页面联动组态，展示相应的“查询属性”弹窗和”下发指令”弹窗。

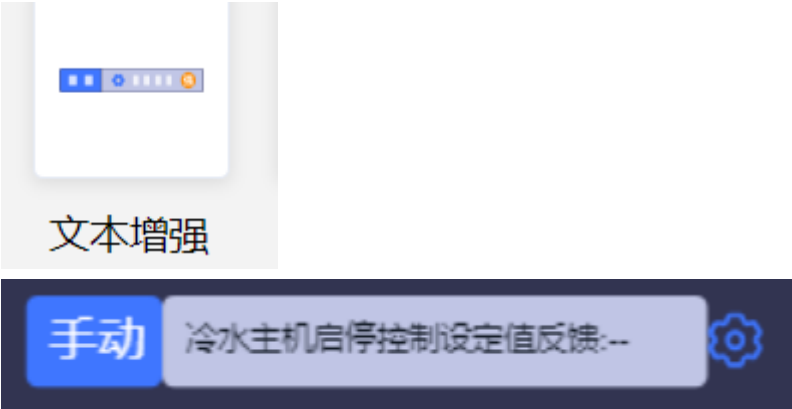
组态发布页面一般是通过iframe的方式嵌入外部母页面的，其中组态弹窗分为“查询属性”和”下发指令”弹窗两类。

操作步骤

步骤1 设备属性明细事件。

若单击组态页查询详情类图标时，组态页会向window发送类型为: 'propertyDetail'的消息，（母页面）打开“查询属性”弹窗；

- **关联组件：**
 1. 文本增强组件



2. 表单组件



- 事件格式:

```
{
  direction:"export",
  type: "propertyDetail",
  data: {
    property_path: ${service_id}/${property_code},
    device_id: ${device_id}
  }
}
```

- 参数说明:

type: 消息类型 (propertyDetail为属性详情)

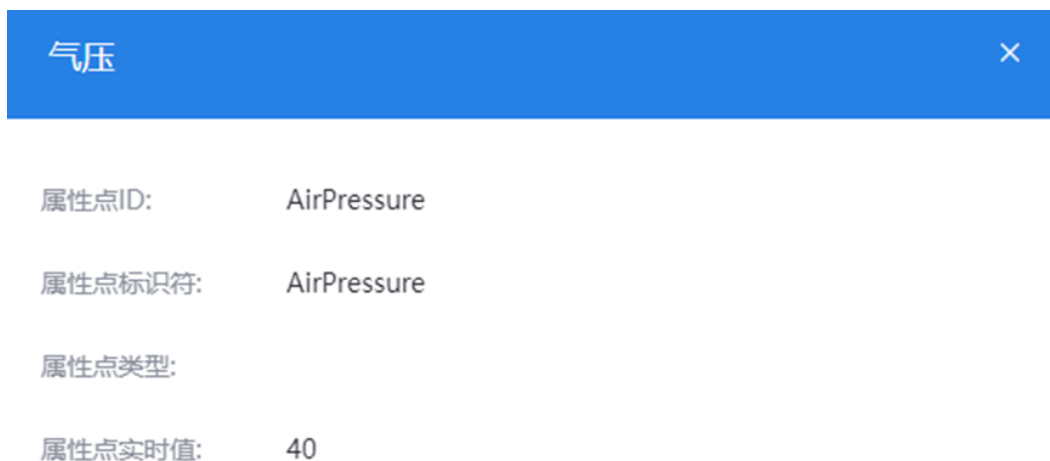
property_path: 服务ID/属性编码

device_id: 设备ID

当事件type类型为propertyDetail时, 说明查看的是具体设备的指定属性,

- 处理样例:

当用户单击组态页面中"文本增强组件"的文字内容时, 组态页将展示该设备属性详情点位弹框, 组态预览时默认效果如下 (组态发布后, 该弹框需由外部嵌入组态发布页的母页面完成开发):



母页面监听该事件的代码示例如下:

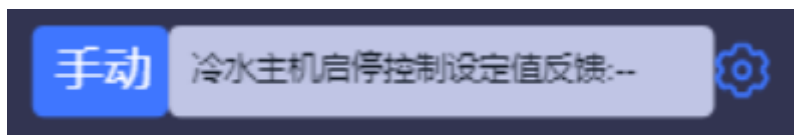
```
window.addEventListener("message", (event) => {  
  if (event.data.type === 'propertyDetail') {  
    this.sendDetail(event.data)  
  }  
})
```

步骤2 设备控制事件。

若单击组态页某设置图标时，组态页会向window发送类型为'deviceControl'或者'propertyControl'的消息，（母页面）打开 ” 下发指令” 弹窗。

- **关联组件：**

1. 文本增强组件



2. 表单组件



3. 风扇组件



4. 设备图标组件



- 事件格式:

```
{
  direction:"export",
  type: "deviceControl",
  data: {
    device_id: ${device_id}
  }
}
```

或:

```
{
  direction:"export",
  type: "propertyControl",
  data: {
    property_path: ${service_id}/${property_code},
    device_id: ${device_id}
  }
}
```

- 参数说明:

type: 消息类型 (*deviceControl*为向设备下发指令,*propertyControl*为控制xx属性)

property_path: 服务ID/属性编码

device_id: 设备ID

isDeviceIcon: 是否单击设备图标

- 处理样例:

(1) 当*type*类型为*deviceControl*时, 说明控制的只是具体设备而没有指定具体属性, 所以没有*property_path*字段

如用户单击组态页面中的“设备组件”时, 由于所选定的是具体设备没有指定具体的属性, 此时在弹窗中可以自行选择设备属性的点位, 组态预览页的默认效果如下 (发布页该弹框需由外部母页面完成开发):

wwp_test_elec

选择点位

请选择

控制

☒ 强制 ☐ 解除强制

当前值

设定值

☒ 关 ☐ 开

说明

请输入说明

0/255

取消

确定

(2) 当type类型为propertyControl时，说明控制的是具体设备的指定属性，所以有property_path字段

如单击组态页面中“文本增强组件”的齿轮图标时，由于已经选定了具体设备的指定属性，此时在弹窗中无法再自行选择设备属性的点位，组态预览页的默认效果如下（发布页该弹框需由外部外部母页面完成开发）。

新风温度

控制

☒ 强制 ☐ 解除强制

当前值

60.00℃

设定值

请输入设定值

℃

说明

请输入说明

0/255

取消

确定

外部母页面，可在页面中监听以下两种类型的消息则可以判断需要如何显示弹窗，代码示例可参考下图：

```
window.addEventListener("message", (event) => {
  if (event.data.type === 'deviceControl' || event.data.type === 'propertyControl') {
    this.sendCommand(event.data)
  }
})
```

```
}  
})
```

----结束

2.2.2.1.3 空间/系统类事件

组态页（空间组态图）支持完成组态渲染后，向windows发送组态页面元数据事件，用于外部母页面监听后，同步展示组态图对应的页面元素信息。

- 事件格式：

```
{  
  "direction": "export",  
  "type": "baPageMetadata",  
  "data": {  
    "products": [  
      {  
        "product_id": ${product_id},  
        "product_name": ${product_name}  
      }  
    ]  
  }  
}
```

- 参数说明：

type: 消息类型（baPageMetadata为组态页元数据）

products: 产品列表

product_id: 产品ID

product_name: 产品名称

- 处理样例：

母页面监听后，可调用相关服务接口，查询出对应空间组态图上，所有产品列表及其归属的系统列表，代码示例如下：

```
window.addEventListener("message", (event) => {  
  if (event.data.type === 'baPageMetadata') {  
    this.productList = event.data.data.products;  
  }  
});
```

2.2.2.1.4 执行跳转事件

为方便母页面区分跳转的联动方式，在现有事件的基础上新增"deviceInfo"事件，该事件涵盖所有组态组件。该事件主要用于单击进行页面跳转并支持进行配置，做到通过设置不同执行方式实现。

操作步骤

步骤1 事件传输：母页面接收监听到window事件后决定执行方式。

事件动作编辑

事件动作选择

事件结果编辑

打开链接

显示/隐藏

控制设备

设置组件行为

设置组件属性

执行方式

事件传输

取消

确认

步骤2 页面跳转：自定义URL地址或选择现有组态页面进行跳转。

事件动作编辑

事件动作选择

事件结果编辑

打开链接

显示/隐藏

控制设备

设置组件行为

设置组件属性

执行方式

页面跳转

跳转地址

URL地址

选择页面

请输入URL

打开在

请选择

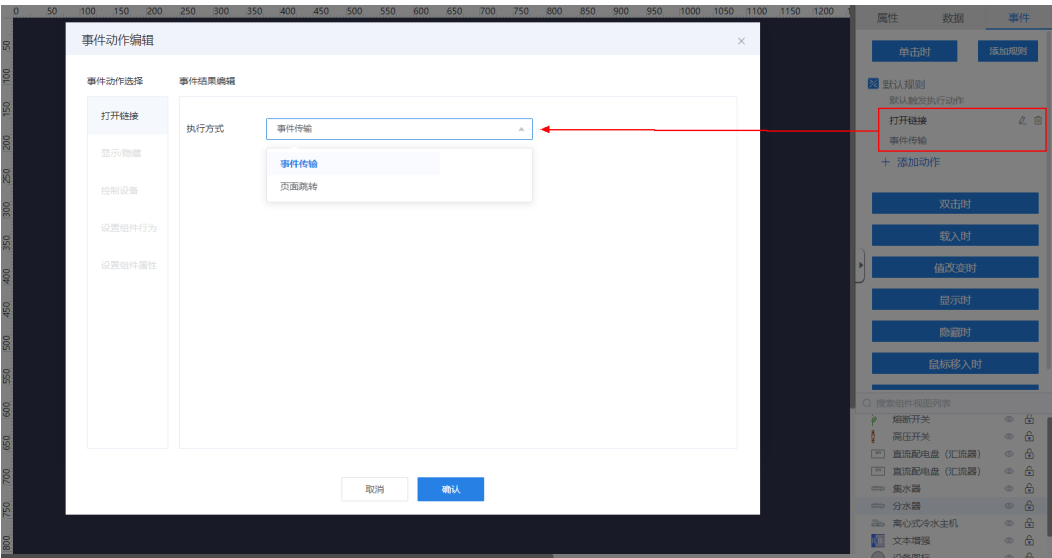
取消

确认

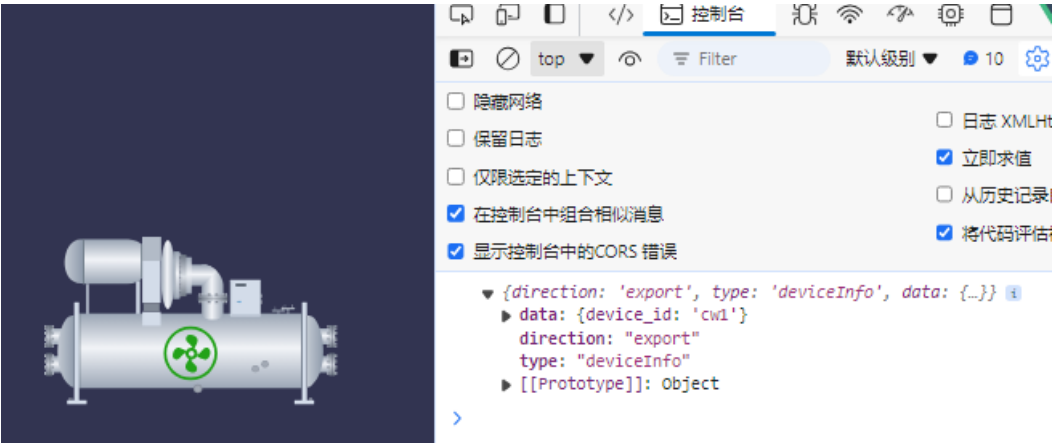
- **事件触发条件：**

选择需要跳转的组件后，在页面右侧选择事件 > 单击或双击时(不支持其他操作类型) > 添加动作。

在事件动作编辑弹窗的打开连接页签中，选择事件传输的执行方式，如下图所示：



此时在发布页面中通过鼠标单击事件单击组件时会发送对应的Window事件：



● 事件格式：

```
{
  direction:"export",
  type: "deviceInfo",
  data: {
    device_id: ${device_id}
  }
}
```

type：消息类型（*deviceInfo*为设备详情）

device_id：设备ID

当事件*type*类型为*deviceInfo*时，说明查看的是具体设备详情，

● 处理样例：

母页面监听该事件的代码示例如下：

```
window.addEventListener("message", (event) => {
  if (event.data.type === 'deviceInfo') {
    this.sendDetail(event.data)
  }
})
```

- **向前兼容:**

1、用户之前未在当前组件的打开连接页签配置跳转地址时，添加动作时打开连接页签默认选择的执行方式为：事件传输；

2、用户之前已在当前组件的打开连接页签配置跳转地址时，编辑动作时则会自动选择执行方式为：页面跳转。

----结束

2.2.2.2 组态页监听事件

2.2.2.2.1 空间/系统组态类事件

组态监听母页面发送的组态显示/隐藏事件，联动组态页中设备图标进行显示/隐藏

- **关联组件:**

组态工具系统组件库中所有组件

- **事件格式:**

```
{
  "direction": "import",
  "type": "baPageVisible",
  "data": {
    "visibleProductIds": [
      "${product_id1}",
      "${product_id2}",
      "allProducts"
    ],
    "invisibleProductIds": [
      "${product_id3}",
      "${product_id4}",
      "allProducts"
    ]
  }
}
```

- **参数说明:**

type: 消息类型（*baPageVisible*为组态显示/隐藏）

visibleProductIds: 需要显示的产品ID列表

invisibleProductIds: 需要隐藏的产品ID列表

若产品id传“allProducts”，则表示组态图中所有产品ID

若某productId既在可见产品ID列表中，又在不可见产品ID列表中，组态图将对该产品对应组件进行显示，而不进行隐藏。

- **处理样例:**

```
this.visible = [];
this.invisible = [];
this.productList.forEach(item => {
  if (item.visible) {
    this.visible.push(item.product_id);
  } else {
    this.invisible.push(item.product_id);
  }
});
if (!this.visible.length) {
  this.invisible = ['allProducts'];
}
```

```
}
if (!this.invisible.length) {
  this.visible = ['allProducts'];
}
let iframeEl = document.getElementById("BaPage");
iframeEl.contentWindow.postMessage(
  {
    type: 'bastudioVisible',
    data: {
      visibleProductIds: this.visible,
      invisibleProductIds: this.invisible
    }
  },
  'http://localhost:8082'
);
```

2.2.2.2.2 退出登录事件

组态监听母页面发送的退出登录事件，执行退出智能设施云平台

- 关联组件

不涉及

- 事件格式：

```
{
  "direction": "import",
  "type": "bastudioLogout"
}
```

- 参数说明：

type: 消息类型（*bastudioLogout*为退出登录）

- 处理样例

假设在母页面中将嵌入的组态iframe页面id定义为BaPage，该母页面可通过调用*postMessage*的方法给组态页面发送消息，发送该事件代码示例，参考如下：

```
let iframeEl = document.getElementById("BaPage");
iframeEl.contentWindow.postMessage(
  {
    direction: "import",
    type: "bastudioLogout"
  },
  "http://localhost:8082"
);
```

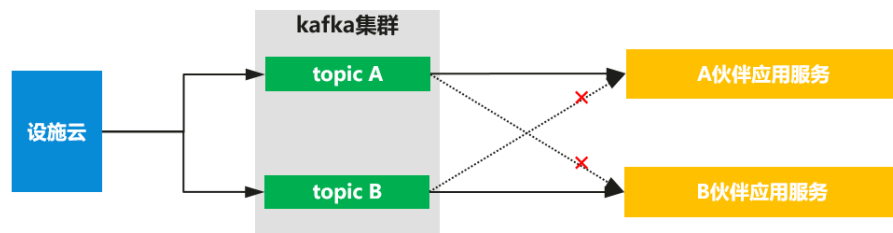
其中：'http://localhost:8082'：iframe子页面域名，是由组态页面来监听该事件（[解决iframe父子页面域名不一样时出现的跨域问题](#)）。

2.3 数据推送

2.3.1 推送方案

数据推送主要用于将设施云实时数据（如告警、设备状态、设备属性等）通过配置推送规格推送到KAFKA服务端，对应伙伴应用去监听对应topic来获取这些实时数据。

图 2-8 推送方案



- 设施云在云上购买了北向KAFKA服务，用于伙伴订阅推送的数据。
- 通过不同topic进行不同项目数据逻辑隔离。
- 通过Kafka ACL进行Topic权限管理。

设施云Kafka用户。

- 用户名：root
- 权限：发布

伙伴Kafka用户

- 用户名：sumg_zuolin
- 用户名规则：{项目名}_{伙伴名}
- 权限：订阅指定topic消息

Topic定义

格式定义：

- >由租户名、消息类型、应用名组成，中间用点分隔。
- >项目名：各项目名称，如musi，可以取项目实例名称。
- >消息类型：分三种类型event-up|meta-data|properties-report。
- >应用名：伙伴名称，如zuolin。
- >字符要求：小写，支持字母数字，中划线，点。

2.3.2 推送步骤

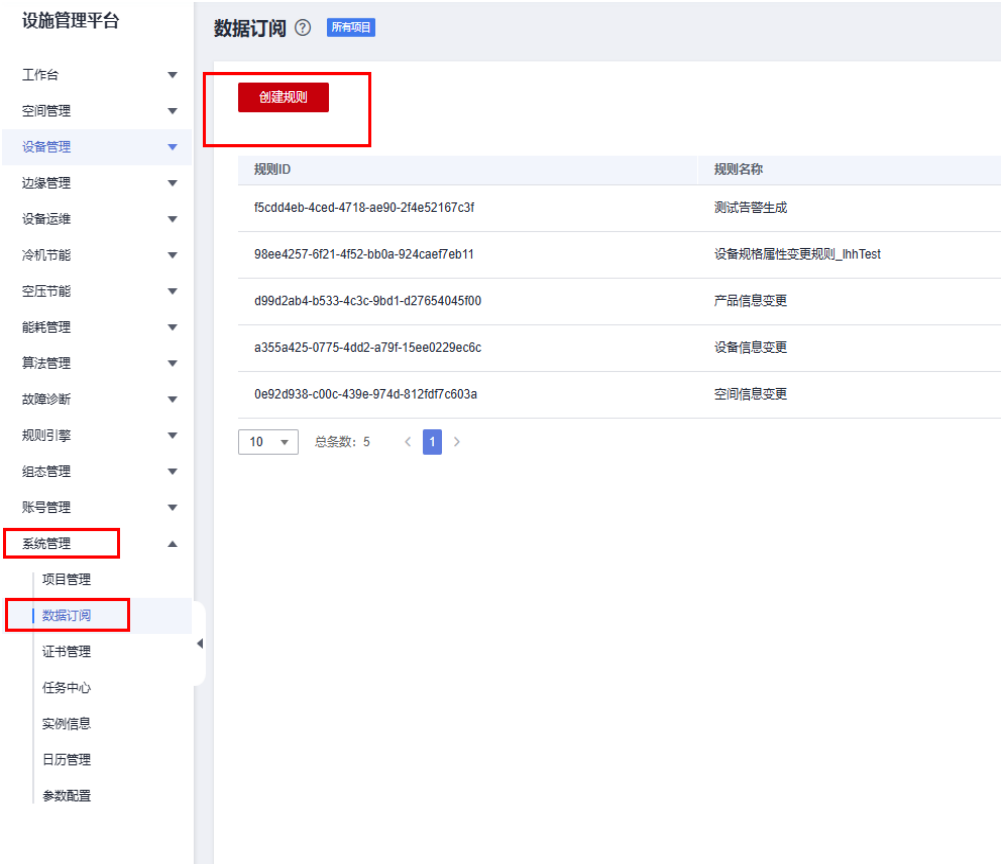
前置条件

如果伙伴应用需要订阅设施云的推送数据（比如告警、设备状态属性等），需要自己搭建或者去华为云上订购一个KAFKA服务，提供对应的topic、连接的用户名、密码和提供北向KAFKA服务端连接地址（调测环境topic可以自己定义）。

配置推送规则

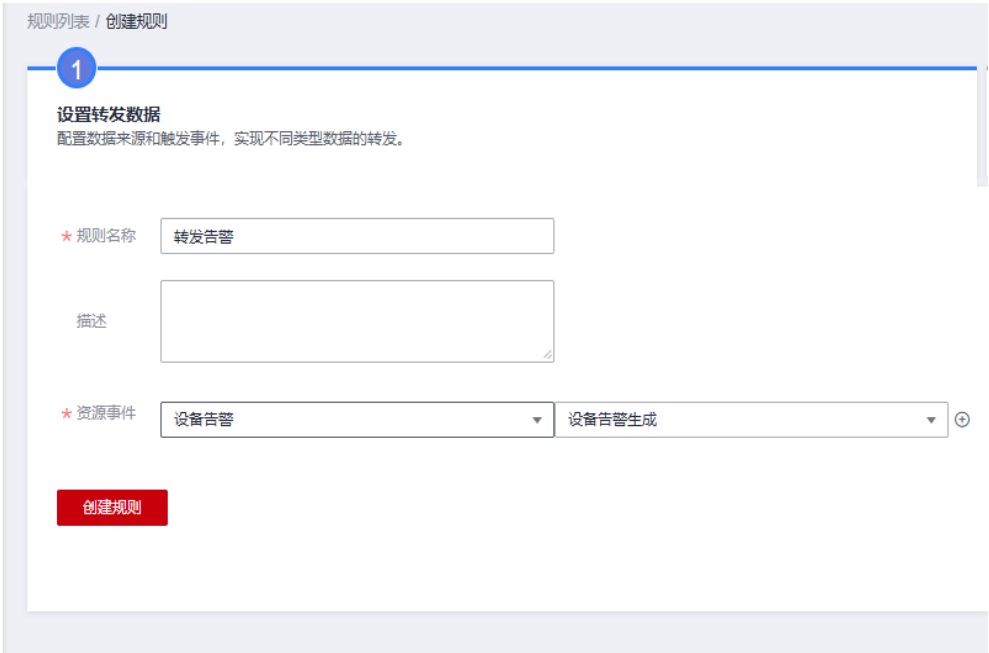
- 步骤1** 访问智能设施云平台管理控制台，选择“系统管理 > 数据订阅”菜单，单击“创建规则”，如下图所示：

图 2-9 创建推送规则



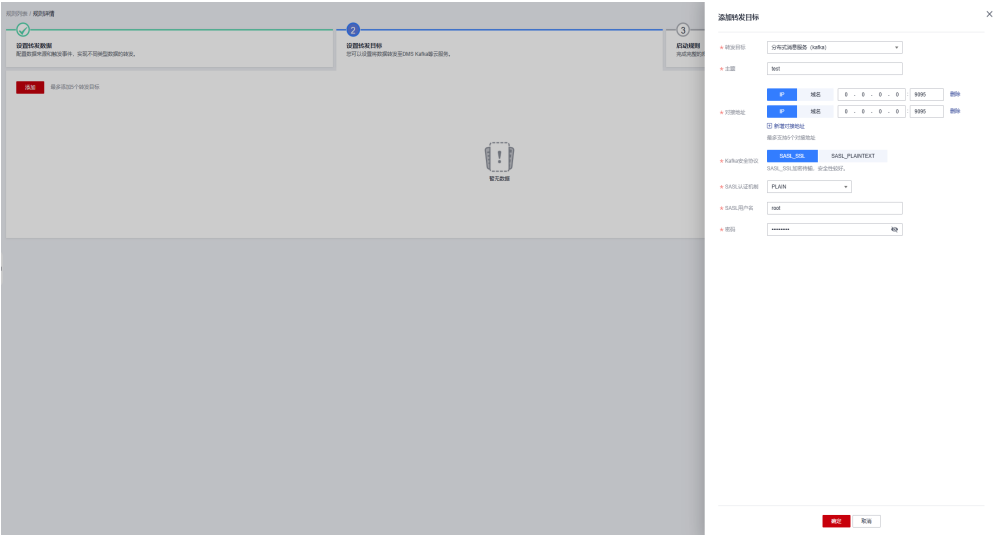
步骤2 填写规则名称，根据自己的需求填写对应资源事件，如果是推送告警选择设备告警，推送设备点位就选择设备属性，推送设备状态就选择设备状态。如下图所示：

图 2-10 配置推送规则



步骤3 单击“创建规则”后弹出设置转发目标框后单击“确认”后在转发页面单击“添加”按钮，配置需要转发到的KAFKA服务端信息（对应信息就是设施云北向KAFKA的topic、IP地址、用户名和密码）。**注意生产环境该用户需要使用root（具有发布消息权限），如果是调测环境可以使用调测环境的partner-test用户。**如下图所示：

图 2-11 kafka 配置



步骤4 如果填写的信息没有问题，单击确定后就可以添加成功了。添加成功后启动规则后就可以把对应的数据推送到设施云提供的北向KAFKA服务端对应的Topic。接下来伙伴应用去监听这个Topic去做处理就行。

----结束

2.3.3 伙伴订阅推送数据

针对设施云推送的数据，伙伴应用通过监听对应topic来订阅推送数据。

订阅步骤

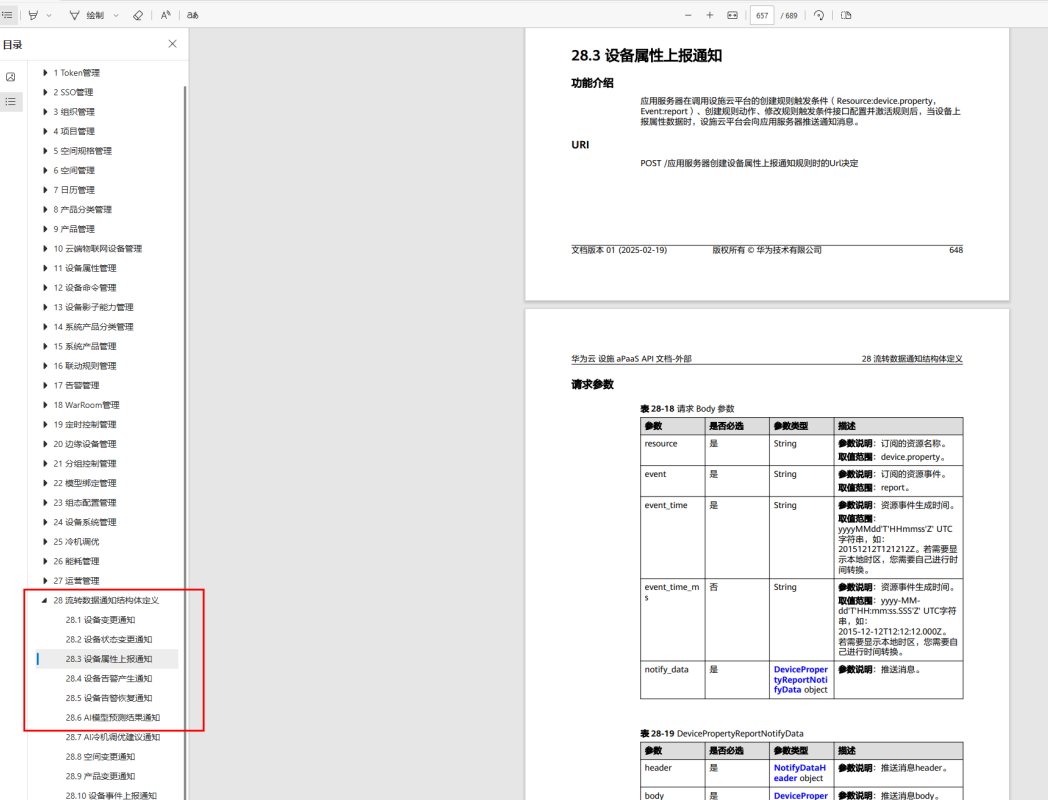
步骤1 订阅前先获取到KAFKA的服务端IP地址、用户名、密码以及需要订阅的Topic。具体监听代码指导可以参考华为云KAFKA的产品资料。链接如下：

<https://support.huaweicloud.com/devg-kafka/kafka-spring.html>

步骤2 对应资料中涉及的client.jks证书文件请联系管理员获取。

步骤3 监听到的资源消息格式和各个字段的说明可以查看智能设施云平台 API文档的“流转数据通知结构体定义”章节，如下图所示：

图 2-12 推送格式



注意

订阅推送数据的代码耗时不能太长，否则会导致消费组消息积压，如果监听到消息后需要复杂处理逻辑，建议采用异步方式处理。

----结束

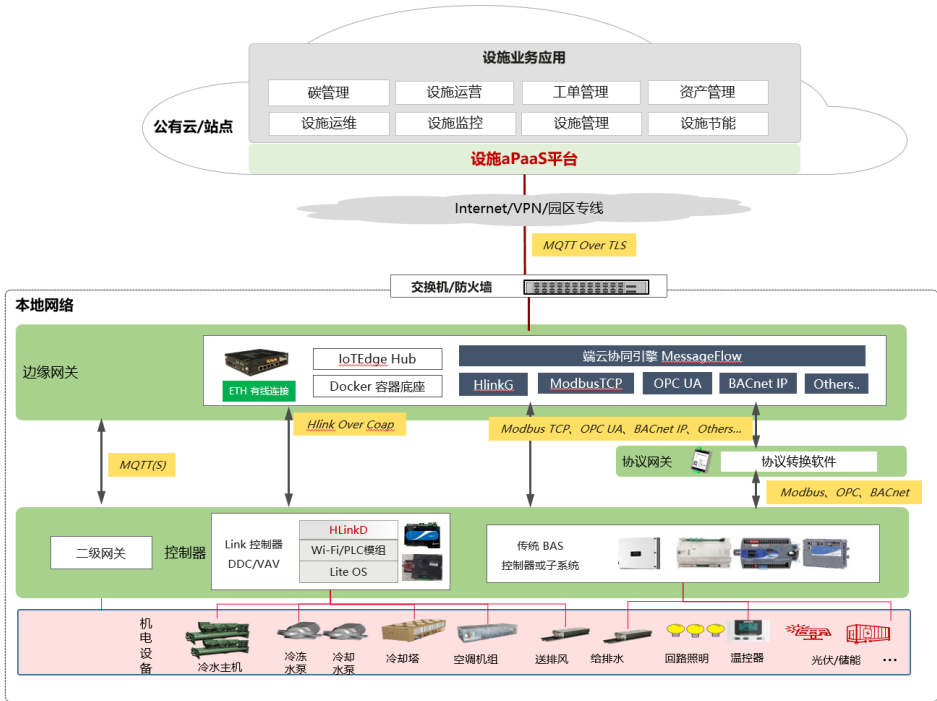
3 南向设备接入

3.1 整体接入架构

设施云整体接入架构图包含如下三个部分：

- 公有云/站点
- Internet/VPN/园区专线
- 本地网络

图 3-1 接入架构图



3.2 MQTT/MQTTS 接入方案

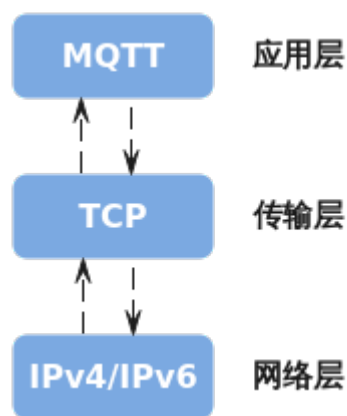
3.2.1 设备侧 MQTT/MQTTS 接口参考

3.2.1.1 使用前必读

概述

MQTT是一种底层传输协议，一般用于网络通信，它提供有序有的、可靠的、双向字节流传输，支持设备到云端和云端到设备之间的消息传递，是基于发布/订阅模式的“轻量级”通信协议。MQTT最大优点在于，可以以极少的代码和有限的带宽，为连接远程设备提供实时可靠的消息服务。作为一种低开销、低带宽占用的即时通讯协议，使其在物联网、小型设备、移动应用等方面有较广泛的应用。

图 3-2 MQTT



数据包组成

MQTT消息由固定报头（Fixed header）、可变报头（Variable header）和有效载荷（Payload）三部分组成。在MQTT规范中有说明，建议使用[设备侧SDK](#)进行接入。

其中固定报头（Fixed header）和可变报头（Variable header）格式的填写请参考MQTT标准规范。有效载荷（Payload）的格式由应用定义（须使用UTF-8编码格式），即设备和物联网平台之间的定义。

- 固定报头：每个 MQTT 控制报文都包含一个固定报头，用于决定控制报文类型、在PUBLISH中还用于决定最大服务质量（QoS），如：建链、订阅、发布。
- 可变报头：某些 MQTT 控制报文包含一个可变报头部分。它在固定报头和负载之间。可变报头的内容根据报文类型的不同而不同。包含可变报头的报文标识符（Packet Identifier），用于区分同一链路的不同数据包。
- 有效载荷：某些 MQTT 控制报文在报文的最后部分包含一个有效载荷，对于PUBLISH来说有效载荷就是应用消息（由用户自己定义）。

📖 说明

MQTT的语法和接口细节，请以MQTT标准规范为准。

主要控制报文类型

常见MQTT消息类型主要有CONNECT、SUBSCRIBE、PUBLISH、PINGREQ。

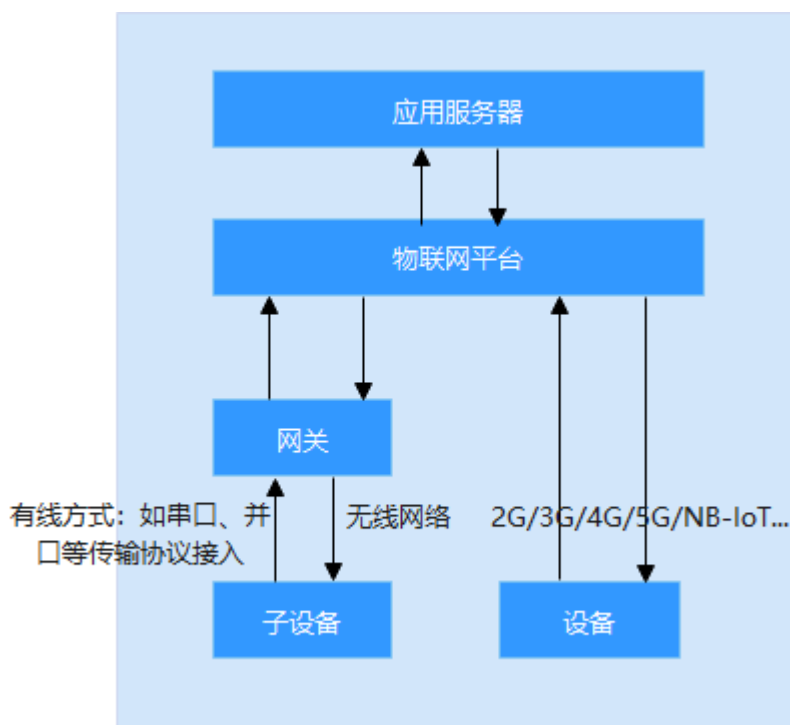
- CONNECT: 指客户端请求和服务端连接。有效载荷（Payload）的主要参数，参考[设备连接鉴权](#)填写。
- SUBSCRIBE: 订阅包括主题筛选器（Topic Filter）和最大服务质量（QoS），一个会话可以进行多个订阅。向平台订阅可以参考：[Topic定义](#)。
- PUBLISH: 是指从客户端向服务端或者服务端向客户端传输一个应用消息。
 - 可变报头（Variable header）中的主要参数“Topic name”，指服务端或客户端的发布主题。详细请参考[Topic定义](#)。
 - 有效载荷（Payload）中的主要参数为完整的数据上报和命令下发的消息内容，一般是一个JSON或字符串对象。
- PINGREQ: 指客户端发给服务端的心跳请求。主要用于告知服务端客户端还活着、确认网络连接没有断开。约定的发送周期由协商的Keep Alive决定。

使用限制

- 上行Topic是指设备向平台发送请求，或上报数据，或回复响应。
- 下行Topic是指平台向设备下发指令，或回复响应。
- 设备与平台建立连接后，需要订阅下行Topic(华为云平台中具有隐式订阅功能，若通信质量为Qos0，平台的[系统topic](#)无需订阅。)，否则无法收到平台下发的指令或回复的响应。应用侧接口的调用，需要设备侧的配合，例如应用侧下发命令，设备侧需要先订阅“平台命令下发”的下行Topic，否则设备无法收到平台命令，应用下发命令的接口也会报超时。

3.2.1.2 通信方式概述

设备接入物联网平台，通过物联网平台进行通信。设备、服务器和物联网平台的通信流程示意图如下。



设备发送数据到物联网平台

设备接入物联网平台后，便可与物联网平台进行通信。设备可通过以下方式发送数据到物联网平台：

- **设备消息上报**：设备无法按照产品模型中定义的属性格式进行数据上报时，将设备的自定义数据通过设备消息上报接口上报给平台，平台将设备上报的消息转发给应用服务器或华为云其他云服务上进行存储和处理。
- **设备属性上报**：用于设备按产品模型中定义的格式将属性数据上报给平台。
- **网关批量属性上报**：用于网关设备将多个设备的属性数据一次性上报给平台。

应用服务器下发指令给设备

设备接入物联网平台后，应用服务器可通过以下方式发送指令到设备。

- **平台消息下发**：用于平台下发自定义格式的数据给设备。
- **平台设置设备属性**：设备的产品模型中定义了平台可向设备设置的属性，应用服务器可通过属性设置的方式修改指定设备的属性值。
- **平台查询设备属性**：应用服务器通过属性查询的方式，实时查询指定设备的属性数据。
- **平台命令下发**：应用服务器按产品模型中定义的命令格式下发控制命令给设备。

3.2.1.3 Topic 定义

设备使用MQTT协议接入平台时，平台和设备通过Topic进行通信。平台预置的topic列表如下：

Topic分类	Topic	Publisher(发布者)	Subscriber(订阅者)	用途
设备消息相关Topic	\$oc/devices/{device_id}/sys/messages/up	设备	平台	设备消息上报
	\$oc/devices/{device_id}/sys/messages/down	平台	设备	平台下发消息
设备命令相关Topic	\$oc/devices/{device_id}/sys/commands/request_id={request_id}	平台	设备	平台下发命令
	\$oc/devices/{device_id}/sys/commands/response/request_id={request_id}	设备	平台	设备返回命令响应
设备属性相关Topic	\$oc/devices/{device_id}/sys/properties/report	设备	平台	设备上报属性
	\$oc/devices/{device_id}/sys/gateway/sub_devices/properties/report	设备	平台	网关批量上报属性

Topic分类	Topic	Publisher(发布者)	Subscriber(订阅者)	用途
	\$oc/devices/{device_id}/sys/properties/set/request_id={request_id}	平台	设备	平台设置设备属性
	\$oc/devices/{device_id}/sys/properties/set/response/request_id={request_id}	设备	平台	属性设置的响应结果
	\$oc/devices/{device_id}/sys/properties/get/request_id={request_id}	平台	设备	平台查询设备属性
	\$oc/devices/{device_id}/sys/properties/get/response/request_id={request_id}	设备	平台	属性查询响应结果
	\$oc/devices/{device_id}/sys/shadow/get/request_id={request_id}	设备	平台	设备侧主动获取平台的设备影子数据
	\$oc/devices/{device_id}/sys/shadow/get/response/request_id={request_id}	平台	设备	设备侧主动获取平台设备影子数据的响应
设备事件相关Topic	\$oc/devices/{device_id}/sys/events/up	设备	平台	设备事件上报与平台事件下发，可用于：设备网关管理、软固件升级、文件上传/下载、设备时间同步、设备信息上报、设备日志收集、远程配置
	\$oc/devices/{device_id}/sys/events/down	平台	设备	

📖 说明

- {device_id}用于标识Topic路由的目标设备，设备侧订阅该topic或往topic推送消息时，该值需要替换为设备与平台建立MQTT连接时使用的设备ID参数值。
- {request_id}用于唯一标识这次请求。设备侧发起的消息带该参数时，需要保证设备侧该参数的唯一性，可以用递增的数字或者UUID来实现。当设备收到平台下发的topic中包含request_id时，设备侧响应的request_id需要跟平台下发的保持一致。
- 设备侧订阅带{request_id}参数的topic时，可以使用通配#。如设备侧订阅命令下发请求的topic \$oc/devices/{device_id}/sys/commands/request_id={request_id}时，可以用\$oc/devices/{device_id}/sys/commands/#。
- 平台采用了隐式订阅的功能，对于下行的系统topic，设备无需订阅，平台默认该设备订阅了qos为0的系统topic。如果需要qos为1的下行系统topic，需要设备自行订阅。
- 除了device_id、request_id其他均为系统字段。

3.2.1.4 设备连接鉴权

接口说明

IoT平台设备侧支持MQTT协议的connect消息接口，鉴权通过后建立设备与平台间的MQTT连接。

参数说明

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。

- 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。

📖 说明

访问[参数生成工具](#)，填写注册设备后生成的设备ID（device_id）和密钥（secret），生成设备连接鉴权所需的参数（ClientId、Username、Password）。

参数	是否 必选	类型	描述
ClientId	是	String	参数解释： 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。 ● 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： - “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。 取值范围： 长度不超过256。
UserName	是	String	参数解释： UserName在此处即为设备ID（device_id）。 取值范围： 长度不超过256。
Password	是	String	参数解释： Password是使用了“HMACSHA256”算法，以secret为内容，时间戳（格式为：YYYYMMDDHH）为密钥，进行加密后获取的值（secret为注册设备时平台返回的secret）。Password = hmacsha256("secret", "时间戳")。 当设备认证类型使用密钥认证接入（SECRET）需填写“Password”，X.509证书认证接入（CERTIFICATES）不需填写“Password”。 “HMACSHA256”即使用SHA-256生成哈希值的HMAC算法，生成的哈希值通用一个长度为64位的十六进制字符串来表示。例如，设备密码为：12345678；时间戳为：2025041401；那么，HMACSHA256后的值为： “c75150e6cb841417396819e4d2ee4358a416344a03a083e3a8567074ddec820a”。 取值范围： 长度不超过256。

原生 MQTT 协议接入建链返回码

原生MQTT协议设备和平台建链时，常见返回码如下：

返回码	返回码描述	原因
0x00	连接成功。	连接成功。
0x01	请求拒绝，协议版本错误。	服务器不支持客户端请求MQTT协议版本。
0x02	请求拒绝，无效的客户端标识符。	clientId不符合格式要求或者心跳时间间隔不满足平台要求。
0x03	请求拒绝，服务器不可用。	平台服务不可用。
0x04	请求拒绝，用户名或密码错误。	用户名或密码错误。
0x05	请求拒绝，没有授权。	客户端没有权限连接。

3.2.1.5 设备命令

3.2.1.5.1 平台命令下发

功能介绍

用于平台向设备下发设备控制命令。平台下发命令后，需要设备及时将命令的执行结果返回给平台，如果设备没响应，平台会认为命令执行超时。命令下发和消息下发的区别，请查看[消息通信说明](#)。

说明

低配置且资源受限或者对网络流量有要求的设备，不适合直接构造JSON数据与物联网平台通信时，可将原始二进制数据透传到物联网平台。通过[开发编解码插件](#)实现二进制数据到平台定义JSON格式的转换。

Topic

下行：\$oc/devices/{device_id}/sys/commands/request_id={request_id}

上行：\$oc/devices/{device_id}/sys/commands/response/request_id={request_id}

说明

- {request_id}用于唯一标识这次请求。设备侧收到下行请求的topic带该参数时，上行响应的topic需要将该参数值返回给平台。
- 应用通过平台[下发设备命令](#)时，平台会生成唯一ID（command_id）用于标识该命令，并返回给应用。同时该唯一标识会通过设备命令下行Topic中的request_id携带给设备。
- 设备无法提前感知该request_id，在订阅该Topic时请使用通配符“#”来替代“request_id={request_id}”即为：\$oc/devices/{device_id}/sys/commands/#。

下行请求参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： - 平台下发时，若为直连设备，不携带该参数。 - 平台下发时，若为网关子设备，该参数为Topic中设备的子设备Id。
service_id	可选	String	参数解释： 设备的服务ID，在设备关联的 产品模型 中定义。
command_name	可选	String	参数解释： 设备命令名称，在设备关联的 产品模型 中定义。
paras	必选	Object	参数解释： 设备命令的执行参数，具体字段在设备关联的 产品模型 中定义。

上行响应参数说明

命令应答的json格式，具体字段在设备关联的产品模型中定义。

字段名	必选/可选	类型	参数描述
result_code	可选	Integer	参数解释： 标识命令的执行结果，0表示成功，其他表示设备执行结果为失败。不带默认为0。
response_name	可选	String	参数解释： 命令的响应名称。
paras	可选	Object	参数解释： 命令的响应参数，具体字段在设备关联的 产品模型 中定义。

下行请求示例

```
Topic: $oc/devices/{device_id}/sys/commands/request_id={request_id}
数据格式:
{
  "object_device_id": "{object_device_id}",
  "command_name": "ON_OFF",
  "service_id": "WaterMeter",
  "paras": {
    "value": "1"
  }
}
```

```
}  
}
```

上行响应示例

```
Topic: $oc/devices/{device_id}/sys/commands/response/request_id={request_id}  
数据格式:  
{  
  "result_code": 0,  
  "response_name": "COMMAND_RESPONSE",  
  "paras": {  
    "result": "success"  
  }  
}
```

3.2.1.6 设备消息

3.2.1.6.1 设备消息上报

功能描述

是指设备无法按照产品模型中定义的属性格式进行数据上报时，可调用此接口将设备的自定义数据格式上报给平台，平台对该消息不进行解析，该消息可以转发给应用服务器或华为云其他云服务上进行存储和处理。

消息上报和属性上报的区别，请查看[消息通信说明](#)。

说明

低配置且资源受限或者对网络流量有要求的设备，不适合直接构造JSON数据与物联网平台通信时，可将原始二进制数据透传到物联网平台。通过[开发编解码插件](#)实现二进制数据到平台定义JSON格式的转换。

Topic

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

说明

- 消息上报时除了使用系统预置的消息上报topic外，设备也可以使用非平台声明的自定义topic。如\$oc/devices/{device_id}/user/{自定义}。
- 数据上报的过程中，可以通过在topic后面携带?request_id来指定request_id。比如说消息上报: \$oc/devices/{device_id}/sys/messages/up?request_id={request_id}。若不指定，平台会自动生成request_id，用于标识此次请求。

参数说明

消息上报对数据内容不做固定的要求，当使用系统格式进行下发时，参数说明如下：

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： - 网关设备上报时，子设备进行上报需填写该参数。object_device_id为Topic中设备的子设备Id，否则将请求失败。 - 直连设备上报时，object_device_id需与Topic中的device_id一致。 - 若为空则默认该参数与Topic中的device_id一致。
name	可选	String	参数解释： 消息名称，可不填写，做说明用。
id	可选	String	参数解释： 消息的唯一标识，用于区分与查找消息。如不填写系统会自动生成一个消息ID，消息ID不可重复。
content	必选	Object、String	参数解释： 消息内容，可以为base64编码格式。

 说明

设备消息上报时平台不会校验消息格式，这里的参数说明和示例为给的消息样例，设备可以根据需要进行自定义数据格式的上报。

示例

假设备上报数据内容为"hello!"，请求的样例参考如下：

- 仅发送消息内容：
Topic: \$oc/devices/{device_id}/sys/messages/up
数据格式：
hello!
- 以系统格式进行上报：
Topic: \$oc/devices/{device_id}/sys/messages/up
数据格式：
{
 "object_device_id": "{object_device_id}",
 "name": null,
 "id": "aca6a906-c74c-4302-a2ce-b17ba2ce630c",
 "content": "hello!"
}

3.2.1.6.2 平台消息下发

功能介绍

设备无法按照产品模型中定义的格式进行指令下发时，可使用此接口下发自定义格式的数据给设备。在此之上也可以使用平台封装的标准格式，即在应用侧API的[下发设备](#)

消息中payload_format填写为“standard”，或者在设备详情页的[云端下发-消息下发](#)中下发时选择“按系统格式”。

消息下发和命令下发的区别，请查看[消息通信说明](#)。

 说明

低配置且资源受限或者对网络流量有要求的设备，不适合直接构造JSON数据与物联网平台通信时，可将原始二进制数据透传到物联网平台。通过[开发编解码插件](#)实现二进制数据到平台定义JSON格式的转换。

Topic

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

参数说明

表 3-1 系统格式下发消息字段表

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： - 平台下发时，若为直连设备，不携带该参数。 - 平台下发时，若为网关子设备，该参数为Topic中设备的子设备Id。
name	可选	String	参数解释： 消息名称，可不填写，做说明用。
id	可选	String	参数解释： 消息的唯一标识，用于区分与查找消息。如不填写系统会自动生成一个消息ID，消息ID不可重复。
content	必选	String	参数解释： 消息内容，可以为base64编码格式。

示例

系统格式：

```
Topic: $oc/devices/{device_id}/sys/messages/down
数据格式：
{
  "object_device_id": "{object_device_id}",
  "name": "name",
  "id": "id",
  "content": "hello"
}
```

自定义格式：

Topic: \$oc/devices/{device_id}/sys/messages/down
数据格式：
arbitrary content

3.2.1.7 设备属性

3.2.1.7.1 设备属性上报

功能介绍

用于设备按产品模型中定义的格式将属性数据上报给平台。属性上报和消息上报的区别，请查看[消息通信说明](#)。

说明

低配置且资源受限或者对网络流量有要求的设备，不适合直接构造JSON数据与物联网平台通信时，可将原始二进制数据透传到物联网平台。通过[开发编解码插件](#)实现二进制数据到平台定义JSON格式的转换。

Topic

上行: \$oc/devices/{device_id}/sys/properties/report

说明

数据上报的过程中，可以通过在topic后面携带?request_id来指定request_id。比如说消息上报: \$oc/devices/{device_id}/sys/properties/report?request_id={request_id}。若不指定，平台会自动生成request_id，用于标识此次请求。

参数说明

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

ServiceProperty结构定义：

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 设备的服务ID，由创建的 产品模型 确定。
properties	必选	Object	参数解释： 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，可以设置多个字段。

字段名	必选/可选	类型	参数描述
event_time	可选	String	参数解释： 设备采集数据UTC时间（格式可选：秒级别：yyyyMMdd'T'HHmmss'Z'，毫秒级别：yyyy-MM-dd'T'HH:mm:ss.SSS'Z'），如： 20161219T114920Z或者 2020-08-12T12:12:12.333Z。 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。

示例

```
Topic: $oc/devices/{device_id}/sys/properties/report
数据格式：
{
  "services": [{
    "service_id": "Temperature",
    "properties": {
      "value": 57,
      "value2": 60
    },
    "event_time": "20151212T121212Z"
  },
  {
    "service_id": "Battery",
    "properties": {
      "level": 80,
      "level2": 90
    },
    "event_time": "20151212T121212Z"
  }
]
```

3.2.1.7.2 网关批量设备属性上报

功能介绍

用于批量设备上报属性数据给平台。网关设备可以用此接口同时上报多个子设备的属性数据。网关与子设备关系，请查看[网关与子设备](#)。
属性上报和消息上报的区别，请查看[消息通信说明](#)。

 说明

网关批量设备属性上报，单批次最多可上报100个子设备的属性数据。如果子设备数量超过100个，建议分不同批次进行上报。

Topic

上行：\$oc/devices/{device_id}/sys/gateway/sub_devices/properties/report

 说明

Topic中device_id为网关设备的设备id。

参数说明

字段名	必选/可选	类型	参数描述
devices	必选	List<DeviceService>	参数解释： 设备数据，是一个字段名为devices的数组，用于存放不同子设备的属性上报。

DeviceService定义表：

字段名	必选/可选	类型	参数描述
device_id	必选	String	参数解释： 此处device_id为网关设备所属的子设备id，用于标记不同子设备的属性上报。
services	必选	List<ServiceProperty>	参数解释： 设备服务数据列表。

ServiceProperty定义表：

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 设备的服务ID，由创建的 产品模型 确定。
properties	必选	Object	参数解释： 设备服务的属性列表，具体字段在产品模型里定义，可以设置多个字段。以<属性名:值>的形式存储属性消息。
event_time	可选	String	参数解释： 设备采集数据UTC时间（格式可选：秒级别：yyyyMMdd'T'HHmmss'Z'，毫秒级别：yyyy-MM-dd'T'HH:mm:ss.SSS'Z'），如：20161219T114920Z或者2020-08-12T12:12:12.333Z。 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。

示例

Topic: \$oc/devices/{device_id}/sys/gateway/sub_devices/properties/report
数据格式:

```
{
  "devices":[
    {
      "device_id":"bf40f0c4-4022-41c6-a201-c5133122054a",
      "services":[
        {
          "service_id":"analog",
          "properties":{
            "PhV_phsA":"1",
            "PhV_phsB":"2"
          },
          "event_time":"20190606T121212Z"
        }
      ]
    },
    {
      "device_id":"42aa08ea-84c1-4025-a7b2-c1f6efe547c2",
      "services":[
        {
          "service_id":"analog",
          "properties":{
            "PhV_phsA":"3",
            "PhV_phsB":"5"
          },
          "event_time":"20190606T121212Z"
        },
        {
          "service_id":"parameter",
          "properties":{
            "Load":"6",
            "ImbA_strVal":"8"
          },
          "event_time":"20190606T121212Z"
        }
      ]
    }
  ]
}
```

3.2.1.7.3 平台设置设备属性

功能介绍

用于平台设置设备属性。设备的产品模型中定义了平台可向设备设置的属性，平台可调用此[修改设备属性](#)接口设置指定设备的属性数据。设备收到属性设置请求后，需要将执行结果返回给平台，如果设备没回响应平台会认为属性设置请求执行超时。

📖 说明

低配置且资源受限或者对网络流量有要求的设备，不适合直接构造JSON数据与物联网平台通信时，可将原始二进制数据透传到物联网平台。通过[开发编解码插件](#)实现二进制数据到平台定义JSON格式的转换。

Topic

下行: \$oc/devices/{device_id}/sys/properties/set/request_id={request_id}

上行: \$oc/devices/{device_id}/sys/properties/set/response/request_id={request_id}

 说明

- 应用[修改设备属性](#)时，平台会生成唯一ID用于标识该请求。同时该唯一标识会通过平台设置设备属性下行Topic中的request_id携带给设备。
- 设备无法提前感知该request_id，在订阅该Topic时请使用通配符“#”来替代“request_id={request_id}”即为：\$oc/devices/{device_id}/sys/properties/set/#。

下行请求参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： - 平台下发时，若为直连设备，不携带该参数。 - 平台下发时，若为网关子设备，该参数为Topic中设备的子设备ID。
services	必选	List<ServiceProperty>	设备服务数据列表。

ServiceProperty结构定义：

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 设备的服务ID，由创建的 产品模型 确定。
properties	必选	Object	参数解释： 设备服务的属性列表，具体字段在产品模型里定义，可以设置多个字段。

上行响应参数说明

字段名	必选/可选	类型	参数描述
result_code	可选	Integer	参数解释： 命令的执行结果，0表示成功，其他表示失败。不带默认认为成功。
result_desc	可选	String	参数解释： 属性设置的响应描述。

下行请求示例

```
Topic: $oc/devices/{device_id}/sys/properties/set/request_id={request_id}
数据格式:
{
  "object_device_id": "{object_device_id}",
  "services": [{
    "service_id": "Temperature",
    "properties": {
      "value": 57,
      "value2": 60
    }
  },
  {
    "service_id": "Battery",
    "properties": {
      "level": 80,
      "level2": 90
    }
  }
]
```

上行响应示例

```
Topic: $oc/devices/{device_id}/sys/properties/set/response/request_id={request_id}
数据格式:
{
  "result_code": 0,
  "result_desc": "success"
}
```

3.2.1.7.4 平台查询设备属性

功能介绍

用于平台向设备查询属性信息。平台可调用[查询设备属性](#)接口查询设备的属性数据。设备收到属性查询请求后，需要将设备的属性数据返回给平台，如果设备没响应平台会认为属性查询请求执行超时。

说明

低配置且资源受限或者对网络流量有要求的设备，不适合直接构造JSON数据与物联网平台通信时，可将原始二进制数据透传到物联网平台。通过[开发编解码插件](#)实现二进制数据到平台定义JSON格式的转换。

Topic

下行: \$oc/devices/{device_id}/sys/properties/get/request_id={request_id}

上行: \$oc/devices/{device_id}/sys/properties/get/response/request_id={request_id}

说明

- {request_id}用于唯一标识这次请求。设备侧收到下行请求的topic带该参数时，上行响应的topic需要将该参数值返回给平台。
- 应用[查询设备属性](#)时，平台会生成唯一ID用于标识该请求。同时该唯一标识会通过平台查询设备属性下行Topic中的request_id携带给设备。
- 设备无法提前感知该request_id，在订阅该Topic时请使用通配符“#”来替代“request_id={request_id}”即为：\$oc/devices/{device_id}/sys/properties/get/#。

下行请求参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： - 平台下发时，若为直连设备，不携带该参数。 - 平台下发时，若为网关子设备，该参数为Topic中设备的子设备ID。
service_id	可选	String	参数解释： 设备的服务ID，由创建的 产品模型 确定。

上行响应参数说明

字段名	必选/可选	类型	参数描述
services	可选	List<ServiceProperty>	参数解释： 设备服务数据列表。

ServiceProperty结构定义：

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 设备的服务ID，由创建的 产品模型 确定。
properties	必选	Object	参数解释： 设备服务的属性列表，具体字段在设备关联的产品模型里定义，可以设置多个字段。
event_time	可选	String	参数解释： 设备采集数据UTC时间（格式：yyyyMMdd'T'HHmmss'Z'），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。

下行请求示例

```
Topic: $oc/devices/{device_id}/sys/properties/get/request_id={request_id}
数据格式：
{
  "object_device_id": "{object_device_id}"
}
```

```
"service_id": "Temperature"
}
```

上行响应示例

Topic: \$oc/devices/{device_id}/sys/properties/get/response/request_id={request_id}

数据格式:

```
{
  "services": [
    {
      "service_id": "Temperature",
      "properties": {
        "PhV_phsA": "1",
        "PhV_phsB": "2"
      },
      "event_time": "20190606T121212Z"
    }
  ]
}
```

3.2.1.7.5 设备侧获取平台的设备影子数据

功能介绍

用于设备向平台获取设备影子数据。用户可以通过应用服务器或物联网控制台配置设备影子预期数据，设备上线时订阅该topic，可以获得到平台设备影子数据，以此来同步设备属性期望值，从而完成设备属性值的修改。

简单交互逻辑介绍如下：

- ① 应用调用“配置设备影子预期数据”接口或在控制台配置设备影子数据；
- ② 设备侧（已完成关联Topic订阅）主动上报请求，获取平台的设备影子数据；
- ③ 平台响应设备请求，返回设备影子数据；
- ④ 设备侧解析属性期望值，并完成设备属性值修改。

Topic

上行: \$oc/devices/{device_id}/sys/shadow/get/request_id={request_id}

下行: \$oc/devices/{device_id}/sys/shadow/get/response/request_id={request_id}

📖 说明

- {request_id}用于唯一标识这次请求。设备侧发起的消息带该参数时，需要保证设备侧该参数的唯一性，可以用递增的数字或者UUID来实现。
- 设备侧上行请求参数中携带唯一标识，该唯一标识会在下行topic中request_id携带给设备以标记唯一请求链。
- 订阅下行Topic时建议使用通配符“#”来替代“request_id={request_id}”即为：\$oc/devices/{device_id}/sys/shadow/get/response/#。

上行请求参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： 需要获取设备影子的目标设备ID。
service_id	可选	String	参数解释： 需要获取设备影子的设备服务ID，不带的话查询所有服务ID的设备影子数据。

下行响应参数说明

字段名	必选/可选	类型	参数描述
object_device_id	必选	String	参数解释： 设备影子的目标设备ID。
shadow	可选	List<ShadowData>	参数解释： 服务影子数据。

ShadowData结构定义：

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 设备的服务ID，由创建的 产品模型 确定。
desired	可选	PropertiesData	参数解释： 设备影子desired区的属性列表。
reported	可选	PropertiesData	参数解释： 设备影子reported区的属性列表。
version	可选	Integer	参数解释： 设备影子版本信息。

PropertiesData结构定义：

字段名	必选/可选	类型	参数描述
properties	必选	Object	参数解释： 设备服务的属性列表，具体字段在设备关联的产品模型里定义，可以设置多个字段。
event_time	可选	String	参数解释： 设备属性数据的UTC时间（格式：yyyyMMdd'T'HHmmss'Z'），如：20161219T114920Z。

上行请求示例

```
Topic: $oc/devices/{device_id}/sys/shadow/get/request_id={request_id}
数据格式：
{
  "object_device_id": "40fe3542-f4cc-4b6a-98c3-61a49ba1acd4",
  "service_id": "WaterMeter"
}
```

下行响应示例

```
Topic: $oc/devices/{device_id}/sys/shadow/get/response/request_id={request_id}
数据格式：
{
  "object_device_id": "40fe3542-f4cc-4b6a-98c3-61a49ba1acd4",
  "shadow": [
    {
      "service_id": "WaterMeter",
      "desired": {
        "properties": {
          "temperature": "60"
        },
        "event_time": "20151212T121212Z"
      },
      "reported": {
        "properties": {
          "temperature": "60"
        },
        "event_time": "20151212T121212Z"
      },
      "version": 1
    }
  ]
}
```

3.2.1.8 网关与子设备管理

3.2.1.8.1 平台通知网关子设备新增

功能介绍

平台将该网关新增的子设备列表信息通知给网关设备。网关与子设备关系，请查看[网关与子设备](#)。

Topic

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

参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释: 该参数与Topic中的device_id一致。
services	可选	List<ServiceEvent>	参数解释: 事件服务列表。

ServiceEvent定义表

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释: 系统字段。 取值范围: 固定为: "\$sub_device_manager"。
event_type	必选	String	参数解释: 系统字段。 取值范围: 固定为: "add_sub_device_notify"。
event_time	可选	String	参数解释: 事件时间。UTC时间, 格式: yyyyMMdd'T'HHmmss'Z'。
paras	必选	Object	参数解释: 事件参数JSON对象。

paras参数列表

字段名	必选/可选	类型	参数描述
devices	必选	List<DeviceInfo>	参数解释: 设备列表。
version	必选	Long	参数解释: 子设备信息版本。网关设备可以保存此信息用于同步子设备列表时携带此参数。

DeviceInfo定义表

字段名	必选/可选	类型	参数描述
parent_device_id	必选	String	参数解释： 父节点设备ID。
node_id	必选	String	参数解释： 设备标识码。
device_id	必选	String	参数解释： 设备ID。
name	可选	String	参数解释： 设备名称。
description	可选	String	参数解释： 设备描述。
manufacturer_id	可选	String	参数解释： 厂商ID。
model	可选	String	参数解释： 设备型号。
product_id	可选	String	参数解释： 产品ID。
fw_version	可选	String	参数解释： 固件版本。
sw_version	可选	String	参数解释： 软件版本。
status	可选	String	参数解释： 设备在线状态。 取值范围： ONLINE：设备在线。 OFFLINE：设备离线。 INACTIVE：设备未激活。
extension_info	可选	Object	参数解释： 设备扩展信息。用户自定义的扩展信息。

示例

```
Topic: $oc/devices/{device_id}/sys/events/down
数据格式:
{
  "object_device_id": "{object_device_id}",
  "services": [{
    "service_id": "$sub_device_manager",
    "event_type": "add_sub_device_notify",
    "event_time": "20151212T121212Z",
    "paras": {
      "devices": [{
        "parent_device_id": "c6b39067b0325db34663d3ef421a42f6_12345678",
        "node_id": "subdevice11",
        "device_id": "2bb4ddba-fb56-4566-8577-063ad2f5a6cc",
        "name": "subDevice11",
        "description": null,
        "manufacturer_id": "of0",
        "model": "twx2",
        "product_id": "c6b39067b0325db34663d3ef421a42f6",
        "fw_version": null,
        "sw_version": null,
        "status": "ONLINE"
      }],
      "version": 1
    }
  ]
}
```

3.2.1.8.2 平台通知网关子设备删除

功能介绍

平台将该网关删除的子设备信息通知给网关设备。网关与子设备关系，请查看[网关与子设备](#)。

Topic

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

参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释: 该参数与Topic中的device_id一致。
services	可选	List<ServiceEvent>	参数解释: 事件服务列表。

ServiceEvent定义表

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 系统字段。 取值范围： 固定为："\$sub_device_manager"。
event_type	必选	String	参数解释： 系统字段。 取值范围： 固定为： "delete_sub_device_notify"。
event_time	可选	String	参数解释： 事件时间。UTC时间，格式： yyyyMMdd'T'HHmmss'Z'。
paras	必选	Object	参数解释： 事件参数JSON对象。

paras参数列表

字段名	必选/可选	类型	参数描述
devices	必选	List<DeviceInfo>	参数解释： 设备信息列表。
version	必选	Long	参数解释： 子设备信息版本。

DeviceInfo定义表

字段名	必选/可选	类型	参数描述
parent_device_id	必选	String	参数解释： 父节点设备ID。
node_id	可选	String	参数解释： 设备标识。
device_id	必选	String	参数解释： 设备ID。

示例

```
Topic: $oc/devices/{device_id}/sys/events/down
数据格式:
{
  "object_device_id": "{object_device_id}",
  "services": [{
    "service_id": "$sub_device_manager",
    "event_type": "delete_sub_device_notify",
    "event_time": "20151212T121212Z",
    "paras": {
      "devices": [{
        "parent_device_id": "c6b39067b0325db34663d3ef421a42f6_12345678",
        "node_id": "subdevice11",
        "device_id": "2bb4ddba-fb56-4566-8577-063ad2f5a6cc"
      }],
      "version": 1
    }
  ]
}
```

3.2.1.8.3 网关同步子设备列表

功能介绍

网关设备从平台同步子设备列表。网关设备不在线时，平台无法将子设备新增和删除的信息及时通知到网关设备。网关设备离线再上线时，可以通过此接口从平台同步这段时间内新增或者删除的子设备信息。新增的子设备信息会通过[平台通知网关子设备新增](#)接口通知网关设备，删除的子设备信息会通过[平台通知网关子设备删除](#)通知网关设备。网关与子设备关系，请查看[网关与子设备](#)。

Topic

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

参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： 该参数目前暂未实装使用，无需填写。
services	可选	List<EventService>	参数解释： 事件服务列表。

EventService定义表

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 系统字段。 取值范围： 固定为："\$sub_device_manager"。

字段名	必选/可选	类型	参数描述
event_type	必选	String	参数解释： 系统字段。 取值范围： 固定为： "sub_device_sync_request"。
event_time	可选	String	参数解释： 事件时间。UTC时间，格式： yyyyMMdd'T'HHmmss'Z'。 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。
paras	必选	Object	参数解释： 事件参数JSON对象。

paras参数列表

字段名	必选/可选	类型	参数描述
version	可选	Long	参数解释： 子设备信息版本。网关设备收到的最近一次的子设备新增或删除通知时的子设备版本信息。平台会根据此版本信息将此版本后新增或者删除的子设备信息通知给网关设备。

示例

```
Topic: $oc/devices/{device_id}/sys/events/up
数据格式:
{
  "object_device_id": "{object_device_id}",
  "services": [{
    "service_id": "$sub_device_manager",
    "event_type": "sub_device_sync_request",
    "event_time": "20151212T121212Z",
    "paras": {"version": 1}
  }]
}
```

3.2.1.8.4 网关更新子设备状态

功能介绍

网关更新子设备状态，更新结果通过网关更新子设备状态响应接口通知网关设备。网关与子设备关系，请查看[网关与子设备](#)。

- 接口tps限制为直连设备并发建链的1.5倍。即为单个实例每秒最大建立请求连接数量为100*购买单元数量 *1.5。
- 该接口1次调用的tps为1*上报子设备的数量。

Topic

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

参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释: 该参数目前暂未实装使用，无需填写。
services	可选	List<EventService>	参数解释: 事件服务列表。

EventService定义表

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释: 系统字段，固定为： "\$sub_device_manager"。
event_type	必选	String	参数解释: 系统字段，固定为： "sub_device_update_status"。
event_time	可选	String	参数解释: 事件时间。UTC时间，格式： yyyyMMdd'T'HHmmss'Z'。 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。
event_id	可选	String	参数解释: 事件请求Id，唯一标识这次事件请求。如果不携带该参数，该参数由物联网平台自动生成，生成规则为数字、字母、中划线组成的36位随机字符串。
paras	必选	Object	参数解释: 事件参数JSON对象。

paras参数列表

字段名	必选/可选	类型	参数描述
device_statuses	必选	List<DeviceStatus>	参数解释： 设备状态列表，列表大小1~100。

DeviceStatus定义表:

字段名	必选/可选	类型	参数描述
device_id	必选	String	参数解释： 子设备ID。
status	必选	String	参数解释： 子设备状态。 取值范围： ● OFFLINE：设备离线。 ● ONLINE：设备上线。

示例

```
Topic: $oc/devices/{device_id}/sys/events/up
数据格式:
{
  "object_device_id": "{object_device_id}",
  "services": [{
    "service_id": "$sub_device_manager",
    "event_type": "sub_device_update_status",
    "event_time": "20151212T1212Z",
    "paras": {
      "device_statuses": [{
        "device_id": "bf40f0c4-4022-41c6-a201-c5133122054a",
        "status": "ONLINE"
      },
      {
        "device_id": "4459c0f7-10bb-4718-9b07-7a82c2d508a5",
        "status": "ONLINE"
      }
    ]
  }
}]
}
```

3.2.1.8.5 网关更新子设备状态响应

功能介绍

平台将该网关更新子设备状态结果以列表返回，平台收到[网关更新子设备状态](#)请求后会在30秒内通过此接口返回响应消息。网关与子设备关系，请查看[网关与子设备](#)。

Topic

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

参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释: 该参数与Topic中的device_id一致。
services	可选	List<ServiceEvent>	参数解释: 事件服务列表。

ServiceEvent定义表

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释: 系统字段。 取值范围: 固定为: "\$sub_device_manager"。
event_type	必选	String	参数解释: 系统字段。 取值范围: 固定为: "sub_device_update_status_response"。
event_time	可选	String	参数解释: 事件时间。UTC时间, 格式: yyyyMMdd'T'HHmmss'Z'。
event_id	必选	String	参数解释: 事件请求Id, 通过该参数关联对应的事件请求。
paras	必选	Object	参数解释: 事件参数JSON对象。

paras参数列表

字段名	必选/可选	类型	参数描述
successful_devices	必选	List<DeviceStatus>	参数解释: 成功更新的子设备列表详情。

字段名	必选/可选	类型	参数描述
failed_devices	必选	List<Reason>	参数解释： 更新子设备状态失败的原因。

DeviceStatus定义表

字段名	必选/可选	类型	参数描述
device_id	必选	String	参数解释： 设备ID。
status	可选	String	参数解释： 设备更新状态结果。 取值范围： ● ONLINE：设备在线。 ● OFFLINE：设备离线。

Reason定义表

字段名	必选/可选	类型	参数描述
device_id	必选	String	参数解释： 对应请求中指定的设备的device_id。
error_code	必选	String	参数解释： 更新失败错误原因码。
error_msg	必选	String	参数解释： 更新失败原因描述。

示例

```
Topic: $oc/devices/{device_id}/sys/events/down
数据格式:
{
  "object_device_id": "{object_device_id}",
  "services": [
    {
      "service_id": "$sub_device_manager",
      "event_type": "sub_device_update_status_response",
      "event_time": "20151212T121212Z",
      "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
      "paras": {
        "successful_devices": [
```

```
{
  "device_id": "c6b39067b0325db34663d3ef421a42f6_subdevice11",
  "status": "ONLINE"
},
"failed_devices": [
  {
    "device_id": "c6b39067b0325db34663d3ef421a42f6_subdevice11",
    "error_code": "XXX",
    "error_msg": "XXXX"
  }
]
}
]
```

3.2.1.8.6 网关新增子设备请求

功能介绍

网关主动新增其下接入的子设备，在平台上完成开户。网关与子设备关系，请查看[网关与子设备](#)。

Topic

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

参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： 该参数目前暂未实装使用，无需填写。
services	必选	List<ServiceEvent>	参数解释： 事件服务列表。

ServiceEvent定义表

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 系统字段。 取值范围： 固定为："\$sub_device_manager"。

字段名	必选/可选	类型	参数描述
event_type	必选	String	参数解释： 系统字段。 取值范围： 固定为： "add_sub_device_request"。
event_time	可选	String	参数解释： 事件时间。UTC时间，格式： yyyyMMdd'T'HHmmss'Z'。 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。
event_id	可选	String	参数解释： 事件请求Id，唯一标识这次事件请求。如果不携带该参数，该参数由物联网平台自动生成，生成规则为数字、字母、中划线组成的36位随机字符串。
paras	必选	Object	参数解释： 事件参数JSON对象。

paras参数列表

字段名	必选/可选	类型	参数描述
devices	必选	List<DeviceInfo>	参数解释： 待新增的子设备信息列表，单次增加最大不超过50个设备。

DeviceInfo定义表

字段名	必选/可选	类型	参数描述
parent_device_id	可选	String	参数解释： 父节点设备ID，默认为对应网关设备id。
node_id	必选	String	参数解释： 设备标识。

字段名	必选/可选	类型	参数描述
device_id	可选	String	参数解释： 设备ID，用于唯一标识一个设备。如果携带该参数，平台将设备ID设置为该参数值；如果不携带该参数，设备ID由物联网平台分配获得，生成规则为"product_id" + "_" + "node_id"拼接而成。
name	可选	String	参数解释： 设备名称。
description	可选	String	参数解释： 设备描述。
product_id	必选	String	参数解释： 设备关联的产品ID，用于唯一标识一个产品模型，在控制台导入产品模型后由平台分配获得。
extension_info	可选	Object	参数解释： 设备扩展信息。用户可以自定义任何想要的扩展信息。字段值大小上限为1K。

示例

```
Topic: $oc/devices/{device_id}/sys/events/up
数据格式:
{
  "object_device_id": "{object_device_id}",
  "services": [
    {
      "service_id": "$sub_device_manager",
      "event_type": "add_sub_device_request",
      "event_time": "20151212T121212Z",
      "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
      "paras": {
        "devices": [
          {
            "name": "subdevice11",
            "node_id": "subdevice11",
            "product_id": "c6b39067b0325db34663d3ef421a42f6",
            "description": "subdevice11"
          },
          {
            "name": "subdevice12",
            "node_id": "subdevice12",
            "product_id": "c6b39067b0325db34663d3ef421a42f6",
            "description": "subdevice12"
          }
        ]
      }
    }
  ]
}
```

3.2.1.8.7 网关新增子设备请求响应

功能介绍

平台将该网关新增的子设备列表信息通知给网关设备，平台收到网关新增子设备请求后会在30秒内通过此接口返回响应消息。网关与子设备关系，请查看[网关与子设备](#)。

Topic

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

参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： 该参数与Topic中的device_id一致。
services	可选	List<ServiceEvent>	参数解释： 事件服务列表。

ServiceEvent定义表

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 系统字段。 取值范围： 固定为："\$sub_device_manager"。
event_type	必选	String	参数解释： 系统字段。 取值范围： 固定为： "add_sub_device_response"。
event_time	可选	String	参数解释： 事件时间。UTC时间，格式： yyyyMMdd'T'HHmmss'Z'。
event_id	必选	String	参数解释： 事件请求Id，通过该参数关联对应的事件请求。
paras	必选	Object	参数解释： 事件参数JSON对象。

paras参数列表

字段名	必选/可选	类型	参数描述
successful_devices	必选	List<DeviceInfo>	参数解释： 成功新增的子设备列表详情。
failed_devices	必选	List<Reason>	参数解释： 新增子设备失败的原因。

DeviceInfo定义表

字段名	必选/可选	类型	参数描述
parent_device_id	必选	String	参数解释： 父节点设备ID。
node_id	必选	String	参数解释： 设备标识。
device_id	必选	String	参数解释： 设备ID。
name	可选	String	参数解释： 设备名称。
description	可选	String	参数解释： 设备描述。
manufacturer_id	可选	String	参数解释： 厂商ID。
model	可选	String	参数解释： 设备型号。
product_id	可选	String	参数解释： 产品ID。
fw_version	可选	String	参数解释： 固件版本。
sw_version	可选	String	参数解释： 软件版本。

字段名	必选/可选	类型	参数描述
status	可选	String	参数解释： 设备在线状态。 取值范围： • ONLINE：设备在线。 • OFFLINE：设备离线。 • INACTIVE：设备未激活。
extension_info	可选	Object	参数解释： 设备扩展信息。用户可以自定义任何想要的扩展信息。

Reason定义表

字段名	必选/可选	类型	参数描述
node_id	必选	String	参数解释： 对应请求中指定的设备的node_id。
product_id	必选	String	参数解释： 对应请求中指定的设备的product_id。
error_code	必选	String	参数解释： 新增失败错误原因码。
error_msg	必选	String	参数解释： 新增失败原因描述。

示例

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

数据格式：

```
{
  "object_device_id": "{object_device_id}",
  "services": [
    {
      "service_id": "$sub_device_manager",
      "event_type": "add_sub_device_response",
      "event_time": "20151212T121212Z",
      "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
      "paras": {
        "successful_devices": [
          {
            "device_id": "c6b39067b0325db34663d3ef421a42f6_subdevice11",
            "name": "subdevice11",
            "node_id": "subdevice11",
            "product_id": "c6b39067b0325db34663d3ef421a42f6",

```

```
        "description": "subdevice11",
        "manufacturer_id": "ofo",
        "model": "twx2",
        "fw_version": null,
        "sw_version": null,
        "status": "ONLINE",
        "extension_info": null,
        "parent_device_id": null
    }
],
"failed_devices": [
    {
        "node_id": "subdevice12",
        "product_id": "c6b39067b0325db34663d3ef421a42f6",
        "error_code": "XXX",
        "error_msg": "XXXX"
    }
]
}
}
```

3.2.1.8.8 网关删除子设备请求

功能介绍

网关主动删除其下接入的子设备，在平台上完成销户。网关与子设备关系，请查看[网关与子设备](#)。

Topic

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

参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： 该参数目前暂未实装使用，无需填写。
services	可选	List<ServiceEvent>	参数解释： 事件服务列表。

ServiceEvent定义表

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 系统字段。 取值范围： 固定为："\$sub_device_manager"。

字段名	必选/可选	类型	参数描述
event_type	必选	String	参数解释： 系统字段。 取值范围： 固定为： "delete_sub_device_request"。
event_time	可选	String	参数解释： 事件时间。UTC时间，格式： yyyyMMdd'T'HHmmss'Z'。 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。
event_id	可选	String	参数解释： 事件请求Id，唯一标识这次事件请求。如果不携带该参数，该参数由物联网平台自动生成，生成规则为数字、字母、中划线组成的36位随机字符串。
paras	必选	Object	参数解释： 事件参数JSON对象。

paras参数列表

字段名	必选/可选	类型	参数描述
devices	必选	List<String>	参数解释： 待删除的子设备（设备id）列表，单次删除最大不超过50个设备。

示例

```
Topic: $oc/devices/{device_id}/sys/events/up
数据格式:
{
  "object_device_id": "{object_device_id}",
  "services": [
    {
      "service_id": "$sub_device_manager",
      "event_type": "delete_sub_device_request",
      "event_time": "20151212T121212Z",
      "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
      "paras": {
        "devices": [
          "c6b39067b0325db34663d3ef421a42f6_subdevice11",
          "c6b39067b0325db34663d3ef421a42f6_subdevice12"
        ]
      }
    }
  ]
}
```

```
]
}
```

3.2.1.8.9 网关删除子设备请求响应

功能介绍

平台将该网关删除的子设备列表信息通知给网关设备。平台收到网关删除子设备请求后会在30秒内通过此接口返回响应消息。网关与子设备关系，请查看[网关与子设备](#)。

Topic

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

参数说明

字段名	必选/可选	类型	参数描述
object_device_id	可选	String	参数解释： 该参数与Topic中的device_id一致。
services	可选	List<ServiceEvent>	参数解释： 事件服务列表。

ServiceEvent定义表

字段名	必选/可选	类型	参数描述
service_id	必选	String	参数解释： 系统字段。 取值范围： 固定为："\$sub_device_manager"。
event_type	必选	String	参数解释： 系统字段。 取值范围： 固定为："delete_sub_device_response"。
event_time	可选	String	参数解释： 事件时间。UTC时间，格式：yyyyMMdd'T'HHmmss'Z'。
event_id	必选	String	参数解释： 事件请求Id，通过该参数关联对应的事件请求。

字段名	必选/可选	类型	参数描述
paras	必选	Object	参数解释： 事件参数JSON对象。

paras参数列表

字段名	必选/可选	类型	参数描述
successful_devices	必选	List<String>	参数解释： 成功删除的子设备（设备id）列表。
failed_devices	必选	List<Reason>	参数解释： 子设备删除失败的原因。

Reason定义表

字段名	必选/可选	类型	参数描述
device_id	必选	String	参数解释： 对应请求中指定的设备的device_id。
error_code	必选	String	参数解释： 删除失败错误原因码。
error_msg	必选	String	参数解释： 删除失败原因描述。

示例

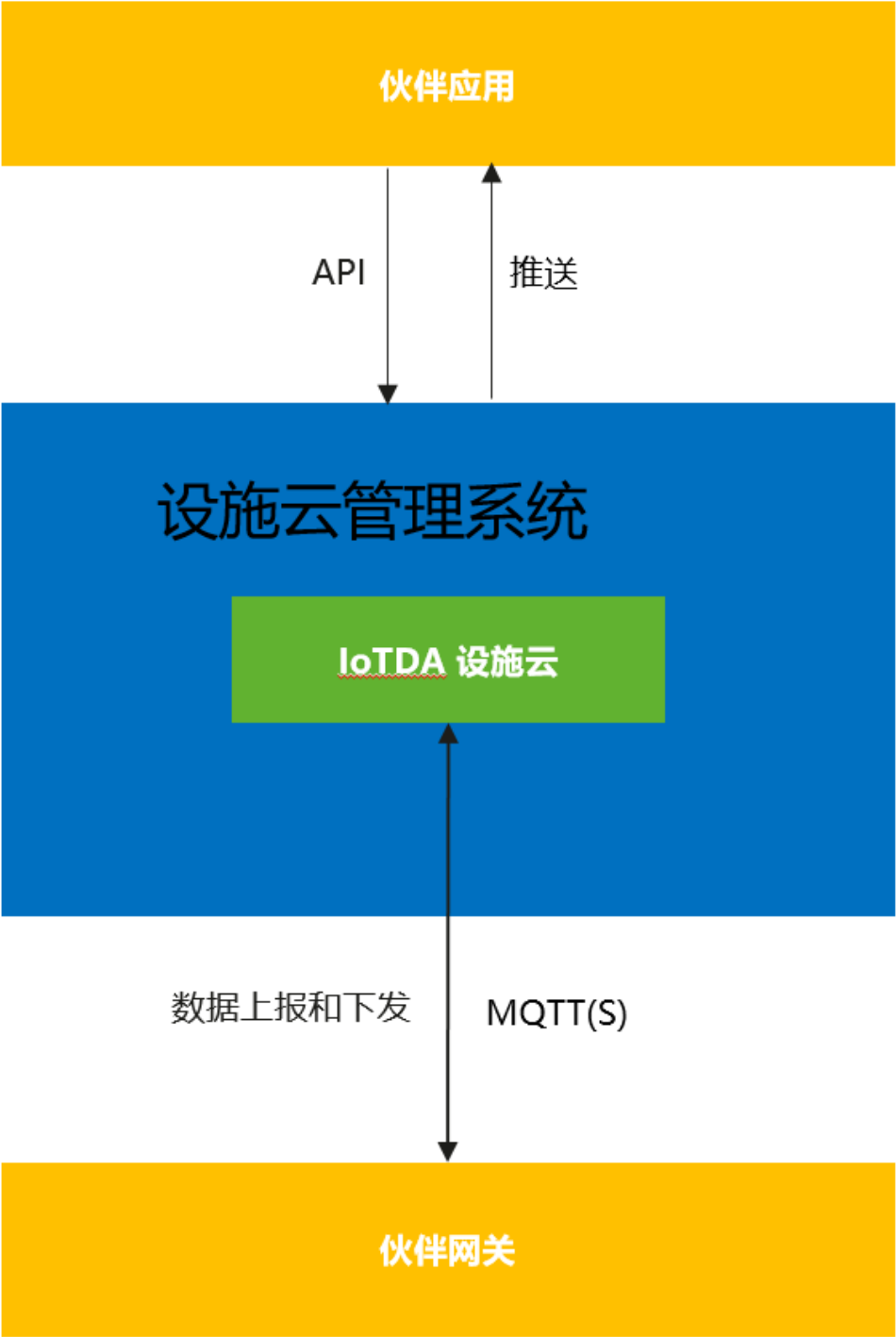
```
Topic: $oc/devices/{device_id}/sys/events/down
数据格式:
{
  "object_device_id": "{object_device_id}",
  "services": [
    {
      "service_id": "$sub_device_manager",
      "event_type": "delete_sub_device_response",
      "event_time": "20151212T121212Z",
      "event_id": "40cc9ab1-3579-488c-95c6-c18941c99eb4",
      "paras": {
        "successful_devices": [
          "c6b39067b0325db34663d3ef421a42f6_subdevice11"
        ],
        "failed_devices": [
          {
            "device_id": "c6b39067b0325db34663d3ef421a42f6_subdevice12",
            "error_code": "XXX",
            "error_msg": "XXXX"
          }
        ]
      }
    }
  ]
}
```

```
    }  
  ]  
}  
}  
]  
}
```

3.2.2 基于 IoTDA 的接入方式

3.2.2.1 接入方案

图 3-3 IoTDA 接入方案



方案说明

- 1. 设备注册需要在设施云页面进行注册物联网类型的设备作为伙伴网关设备，设施云注册后会同步设备到IoTDA平台。

2. 设施云后台有一个IoTDA实例，伙伴网关只需通过MQTT的相关接口对接到这个IoTDA实例上，对应的设备数据就能转发到设施云。
3. 针对伙伴网关接入的那些子设备，需要在这个伙伴网关设备下注册对应子设备来承载。
4. 该对接方案使用的直连设备接入，只能进行简单的点位控制，不支持设备批量控制、设备定时控制和部署控制逻辑等复杂的控制。

3.2.2.2 接入方式

伙伴网关南向对接连接信息：

对接MQTT服务端地址为IoTDA南向设备接入地址去设施云前端页面“系统管理”》“实例信息”菜单获取，对应就是设备接入的MQTT协议地址，MQTT端口为8883，MQTT端口为1883，连接身份使用伙伴网关设备。针对伙伴网关下接入的设备，对应的就是网关下注册的子设备。

伙伴网关南向对接必须支持的接口：

[网关下子设备状态上报](#)

[网关下更新子设备状态响应](#)

[网关上报子设备属性](#)

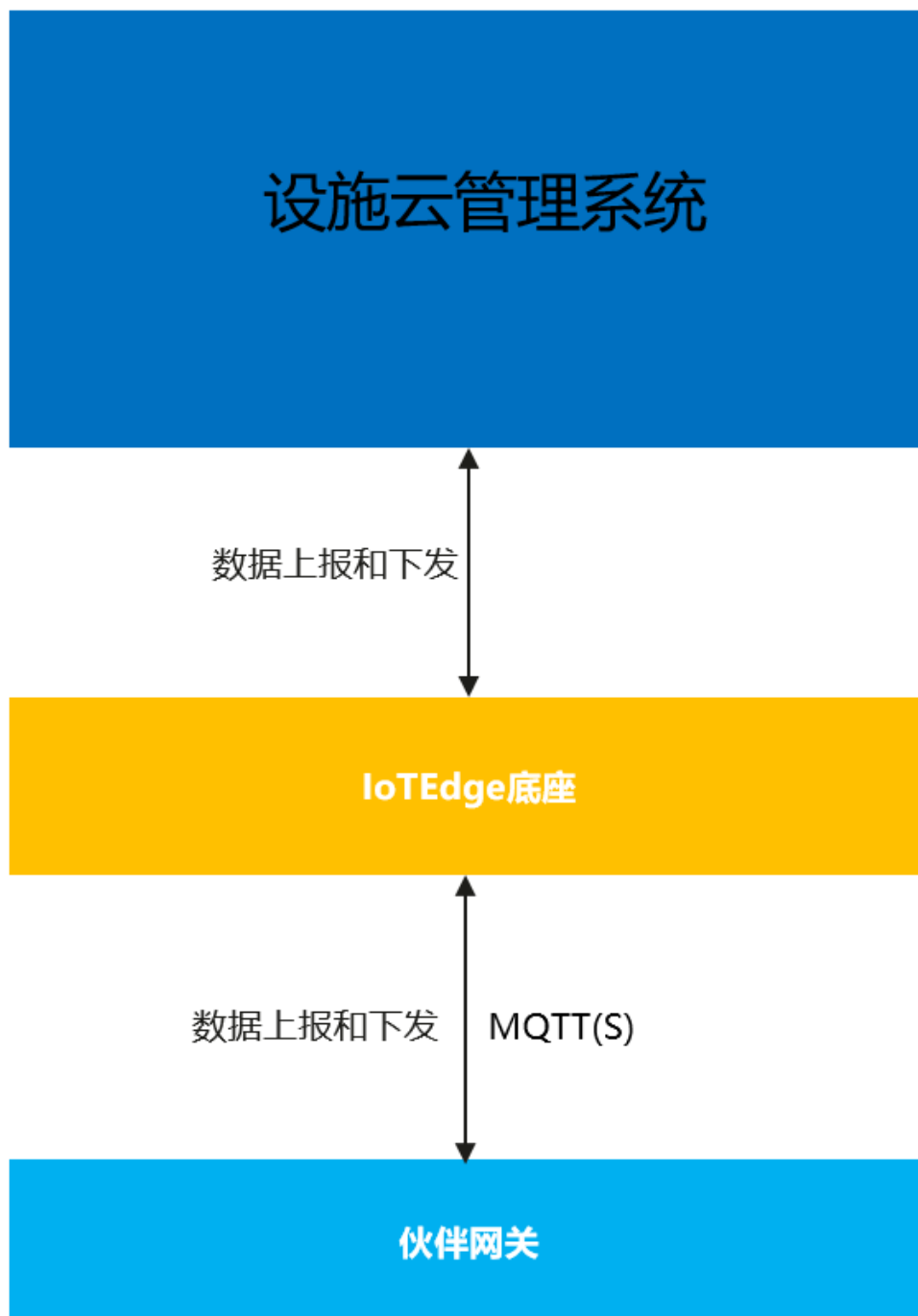
[设施云控制点位下发](#)

[设施云查询设备属性](#)

3.2.3 基于 IoTEdge 的接入方式

3.2.3.1 接入方案

图 3-4 IoTEdge 接入方案



方案说明

1. 伙伴网关如果无法安装IoTEdge网关节点，就需要购买微能网关来安装IoTEdge底座。
2. 设备注册先注册一个**网关**类型的设备来安装IoTEdge底座，接着需要注册**机电设备**类型的设备作为伙伴网关设备，设施云注册后会同步设备到IoTEdge平台。

3. 伙伴网关只需通过MQTT的相关接口对接到这个IoTEdge底座，对应的设备数据就能转发到设施云。
4. 针对伙伴网关接入的那些子设备，需要在这个伙伴网关设备下注册对应子设备来承载。
5. 该对接方案可以支持点位控制，设备批量控制、设备定时控制和部署控制逻辑等复杂的控制。

3.2.3.2 接入方式

伙伴网关南向对接连接信息：

对接MQTT服务端地址为安装网关的节点服务器公网IP地址，MQTTs端口为7883，MQTT端口为7882，连接身份使用伙伴网关设备。针对伙伴网关下接入的设备，对应的就是网关下注册的子设备。

伙伴网关南向对接必须支持的接口：

[网关下子设备状态上报](#)

[网关下更新子设备状态响应](#)

[网关上报子设备属性](#)

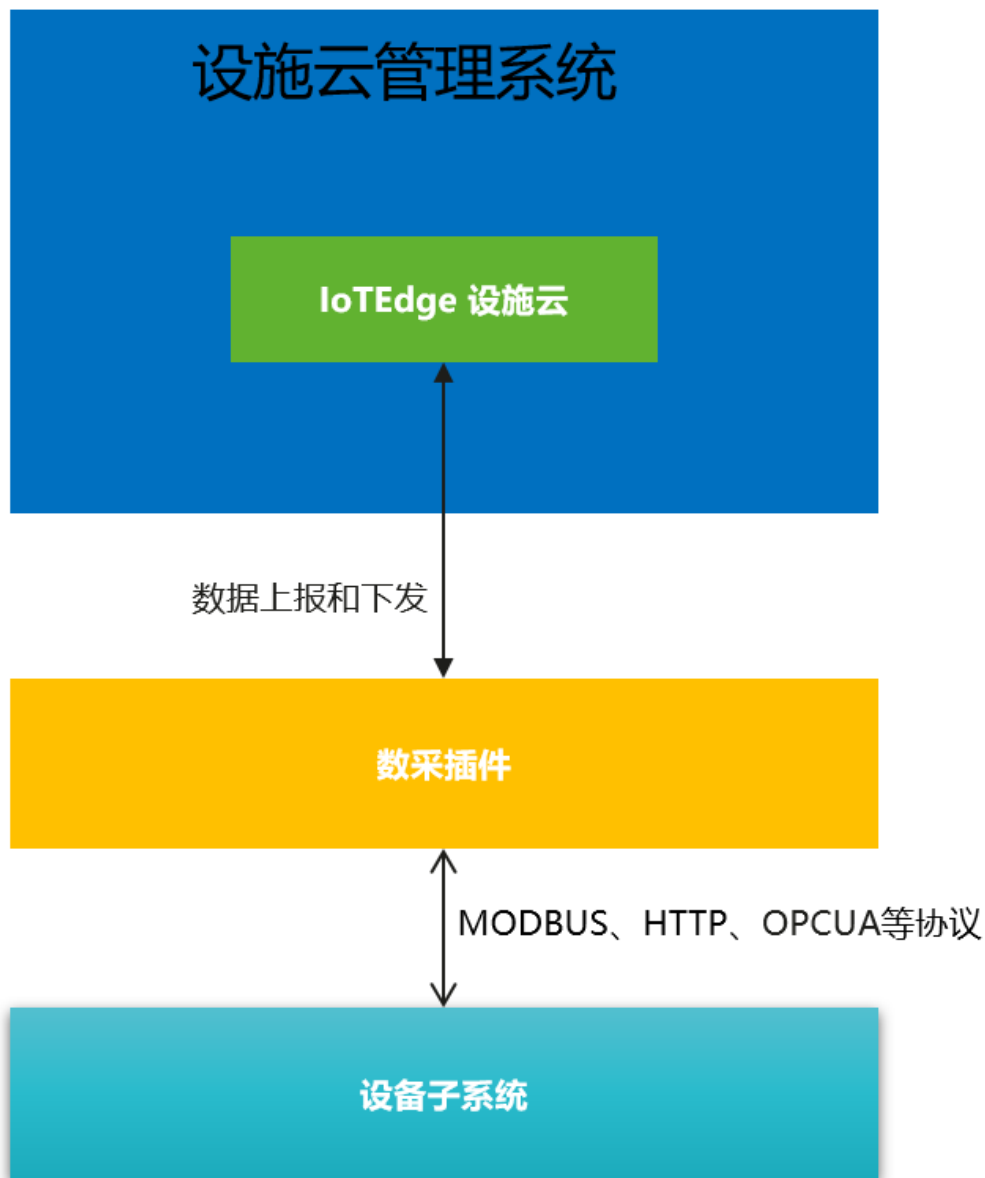
[设施云控制点位下发](#)

[设施云查询设备属性](#)

3.3 数采插件接入方案

3.3.1 接入方案

图 3-5 数采插件接入方案



方案说明

1. 伙伴网关如果无法安装IoTEdge网关节点，就需要购买微能网关来安装IoTEdge底座。
2. 设备注册先注册一个**网关**类型的设备来安装IoTEdge底座，接着需要注册**机电设备**类型的设备作为接入设备，设施云注册后会同步设备到IoTEdge平台。
3. 伙伴可以根据平台提供的SDK来开发数采插件来实现不同协议对接设备数据的采集和下发，具体插件开发可以参考https://support.huaweicloud.com/devg-iotedge/iotedge_devg_0008.html，推荐使用 **集成ModuleSDK**进行OT数采。

4. 目前设施云只提供了MODBUS协议的数采插件，可以支持通过MODBUS协议来对接设备。
5. 该对接方案可以支持点位控制，设备批量控制、设备定时控制和部署控制逻辑等复杂的控制。

3.3.2 接入案例

当前以设施云自研的Modbus数采插件为例来配置设备点位通过数采方式接入，如果是伙伴自己开发的数采插件，流程也是类似，只是数采模板和对应应用的镜像地址这些容器配置会不同。

添加数采模板

1. 保存以下模板保存到本地，文件名为Modbus-Tcp.json

数采模板内容：

```
{
  "tpl_id": "iotfm_general_modbustcp_V3",
  "name": "智能设施云平台自研ModbusTcp数采模板V3",
  "description": "智能设施云平台自研ModbusTcp数采模板V3",
  "datasource_meta": {
    "config_tabs": [
      {
        "key": "connection_info",
        "name": "连接信息",
        "description": "连接信息",
        "type": "object",
        "quantity": 10,
        "config_items": [
          {
            "key": "connection_ext_info",
            "name": "连接扩展信息",
            "description": "连接扩展信息，协议应用上线后模板增删字段需重建数采连接，新增扩展信息容纳后续新增的字段。",
            "data_type": "string",
            "required": false,
            "crypted": false,
            "max_length": 65535,
            "example": "{\"xxx\": \"xxx\"}"
          }
        ]
      },
      {
        "key": "array_info",
        "name": "连接信息",
        "description": "连接信息",
        "type": "array",
        "quantity": 10,
        "config_items": [
          {
            "key": "endpoint",
            "name": "服务端点信息",
            "description": "服务端点信息，为json字符串数组格式",
            "data_type": "string",
            "required": true,
            "crypted": false,
            "max_length": 128,
            "example": "{\"endpoint_id\": \"1\", \"endpoint_url\": \"127.0.0.1:123\"}"
          }
        ]
      }
    ]
  },
  "default_values": {
    "drivename": "ModbusTcp"
  }
}
```

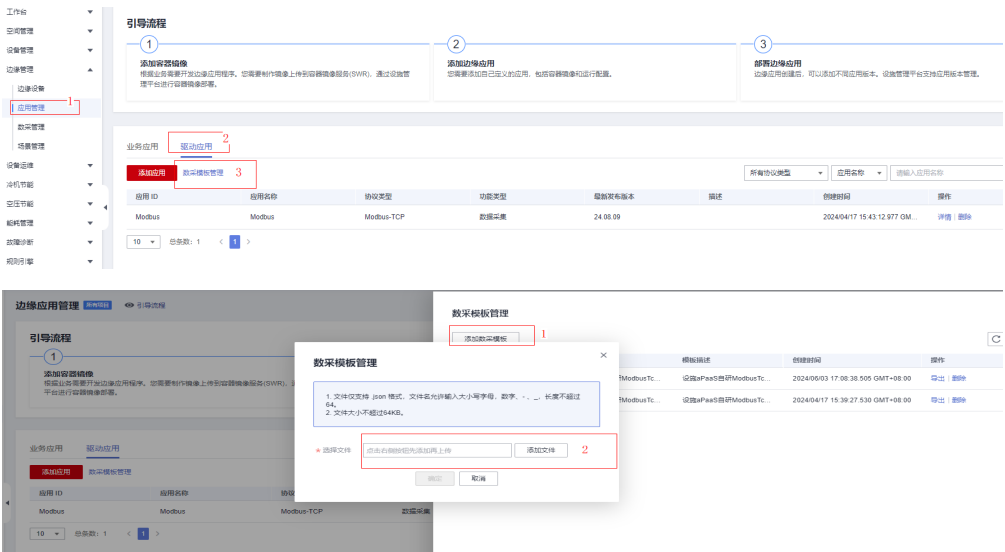
```
},
"point_meta": {
  "data_type_enums": [
    "bool",
    "short",
    "ushort",
    "int",
    "uint",
    "long",
    "ulong",
    "float",
    "double"
  ],
  "config_items": [
    {
      "key": "endpoint_id",
      "name": "终端节点ID",
      "description": "Modbus终端节点ID，来源于数采连接配置，如配置失败，则将被丢弃。",
      "data_type": "int",
      "required": true,
      "crypted": false,
      "max_length": 256,
      "example": 1
    },
    {
      "key": "slave_id",
      "name": "从站号",
      "description": "modbus传感器从站号",
      "data_type": "int",
      "required": true,
      "crypted": false,
      "max_length": 256,
      "example": 1
    },
    {
      "key": "register_type",
      "name": "寄存器类型",
      "description": "寄存器类型",
      "data_type": "string",
      "required": true,
      "crypted": false,
      "max_length": 256,
      "enum_value": "Coil_Status|Input_Status|Holding_Registers|Input_Registers",
      "example": "Coil_Status"
    },
    {
      "key": "register_address",
      "name": "起始地址",
      "description": "寄存器起始地址",
      "data_type": "string",
      "required": true,
      "crypted": false,
      "max_length": 256,
      "example": "0x0"
    },
    {
      "key": "cycle",
      "name": "采集周期",
      "description": "点位采集周期",
      "data_type": "int",
      "required": true,
      "crypted": false,
      "max_length": 65535,
      "example": 10000
    },
    {
      "key": "byte_swap",
      "name": "寄存器高低位交换",
      "description": "寄存器高低位交换",

```

```
    "data_type": "bool",
    "required": false,
    "crypted": false,
    "max_length": 256,
    "default": false,
    "example": true
  },
  {
    "key": "register_swap",
    "name": "寄存器顺序交换",
    "description": "寄存器顺序交换",
    "data_type": "bool",
    "required": false,
    "crypted": false,
    "max_length": 256,
    "default": false,
    "example": true
  },
  {
    "key": "point_ext_info",
    "name": "点位扩展信息",
    "description": "点位扩展信息，协议应用上线后模板增删字段需全量调整点位，新增扩展信息容
    纳后续新增的点位字段。",
    "data_type": "string",
    "required": false,
    "crypted": false,
    "max_length": 65535,
    "example": "{\"field_a\": \"valueA\", \"field_b\": 123}"
  },
  {
    "crypted": false,
    "data_type": "string",
    "description": "该物理点位枚举值和物模型点位枚举值的映射关系",
    "example": "物理点位枚举值1:物模型点位枚举值1|物理点位枚举值2:物模型点位枚举值2",
    "key": "enum_map",
    "max_length": 256,
    "name": "枚举值映射表",
    "required": false
  }
}
```

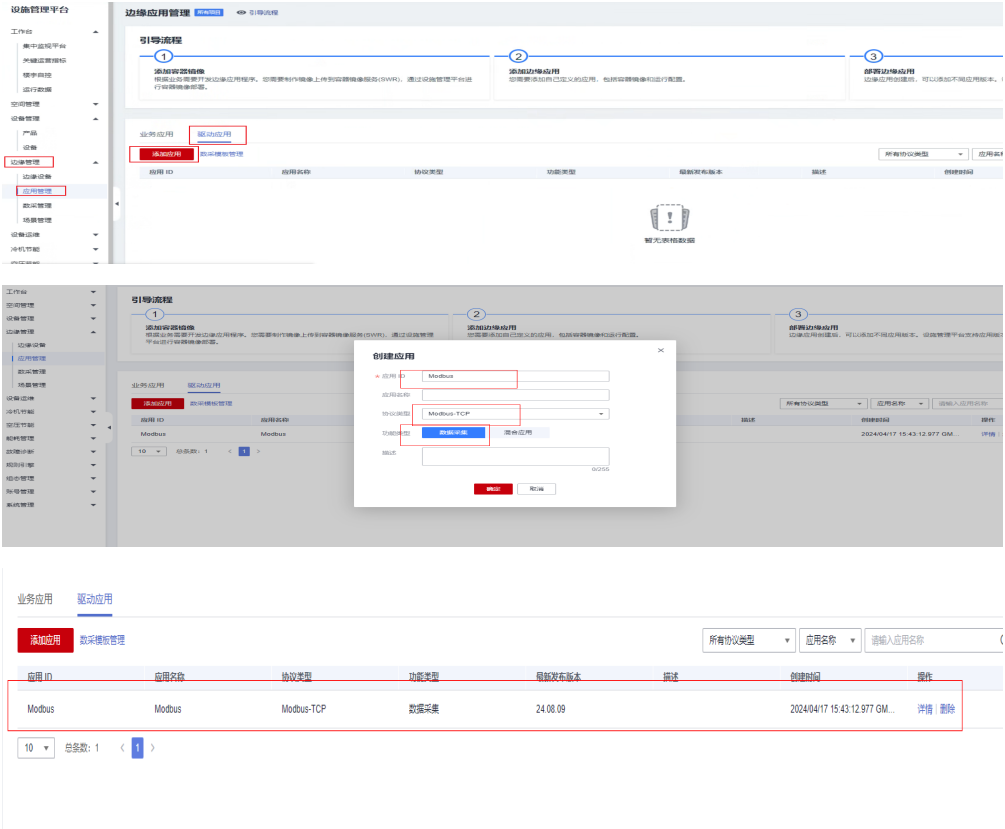
纳后续新增的点位字段。",

2. 导入上面文件到数采模板





添加数采应用



添加应用版本

须知：该步骤将会为上述业务应用添加一个可用版本

【边缘管理】->【应用管理】->【驱动应用】->【操作--详情】->【添加版本】





1. 基本配置，容器镜像地址可以使用：`swr.cn-north-4.myhuaweicloud.com/huawei-iot-facility/iotfm-edge-app-plugins-modbus-tcp:2025.0113.1944.37`



2. 配置镜像地址



3. 配置容器规格



4. 高级配置



高级配置

运行命令 选项配置 环境变量 数据存储 外挂设备 健康检查

① 勾选选项，开启特权模式，容器将具有其宿主主机几乎所有能力，包括一些内核特性和所有设备访问。

是否开启 ☐

高级配置

运行命令 选项配置 环境变量 数据存储 外挂设备 健康检查

① 环境变量会明文展示所输入信息，请不要输入敏感信息，如涉及敏感信息，请先加密，防止信息泄露

容器运行环境中设置的一个变量，可以在应用部署后修改，为应用提供极大的灵活性，设置容器运行环境中的系统环境变量。

添加环境变量

变量名称	变量值	操作
------	-----	----

数据存储

高级配置

运行命令 选项配置 环境变量 数据存储 外挂设备 健康检查

① 支持挂载本地卷到容器中，以实现数据文件的持久化存储

添加卷

本地卷名称	类型	主机目录	容器目录	权限	操作
config	CONFIG	/config	/opt/iot/iotfm-edge-app-pl	读写	删除
logs	LOG	/tmp/iotfm-edge-app-plug	/opt/iot/iotfm-edge-app-pl	读写	删除

本地卷名称	类型	主机目录	容器目录	权限
config	CONFIG	/config	/opt/iot/iotfm-edge-app-plugins-modbus-tcp/config	读写
logs	LOG	/tmp/iotfm-edge-app-plugins-modbus-tcp/logs	/opt/iot/iotfm-edge-app-plugins-modbus-tcp/logs	读写

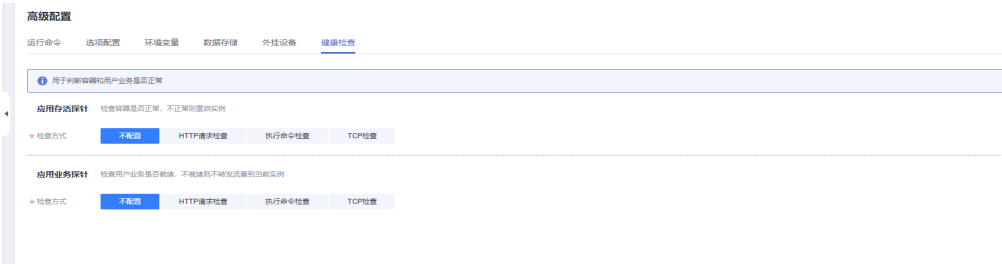
高级配置

运行命令 选项配置 环境变量 数据存储 外挂设备 健康检查

① 支持Linux外挂设备的挂载到容器中，如常见的串口、GPU、NPU设备

添加设备

主机目录	容器目录	权限	操作
------	------	----	----



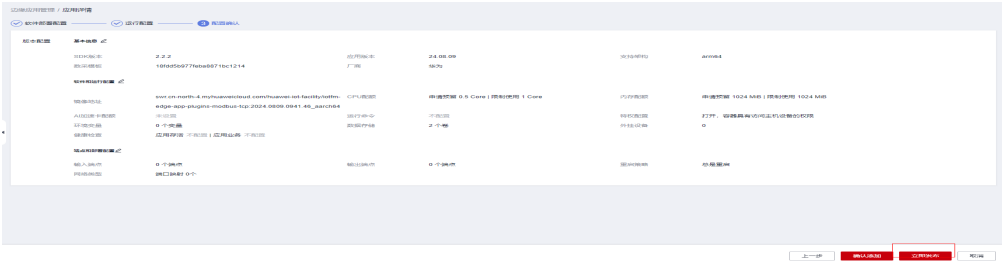
5. 端点无需配置



6. 部署配置
重启策略：总是重启
网络类型：端口映射



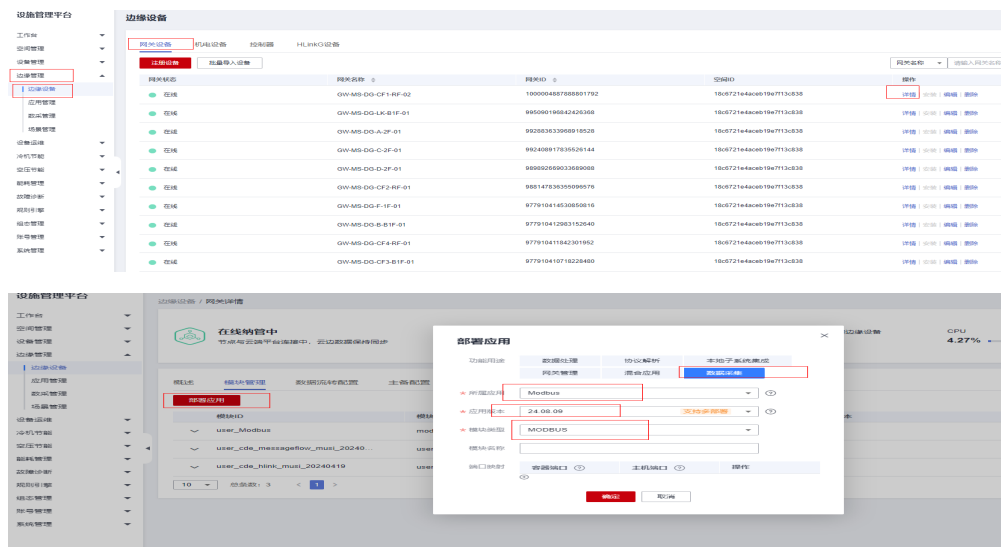
7. 确认信息，立即发布



插件部署

部署Modbus应用模块

1. 【边缘管理】->【边缘设备】->【网关设备】->【操作】->【详情】->【模块管理】->【部署应用】



2. 单击确定即开始进行Modbus插件安装，等待安装完成，状态为运行中

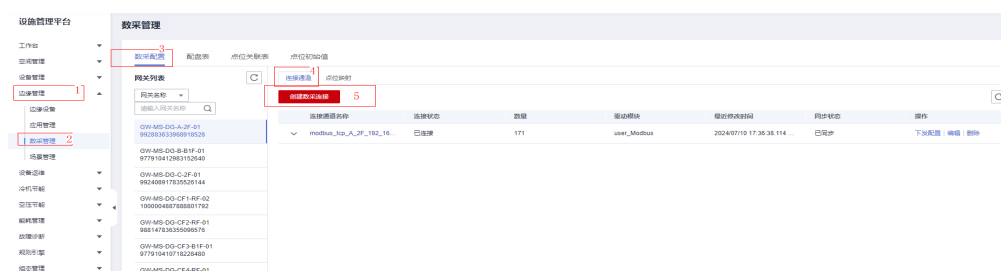


数采配置

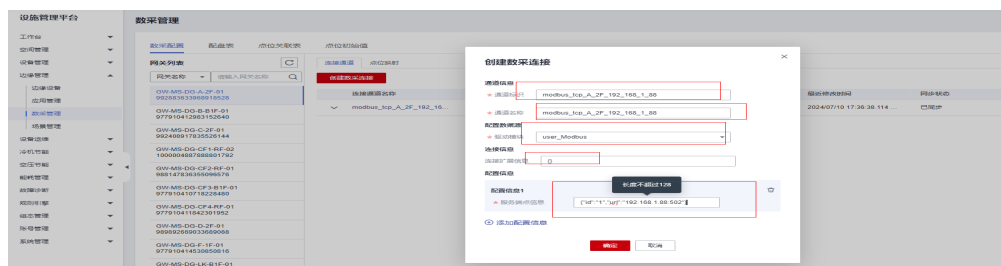
- ## 1. 创建数采连接

【边缘管理】->【数采管理】->【数采配置】->【连接通道】->【创建数采连接】

选择指定网关创建数采连接:



创建数采连接



服务短点信息配置：按照指定格式即可

如: {"id":"1","url":"192.168.1.88:502"}

2. 导入点位

根据点位表模板，梳理完成点位表信息内容进行导入

common_config				collection_config			
point_id	name	device_id	property	data_type	endpoint_id	slave_id	register_type
PositiveActiveEnergy_EM-M	正向有功总电能	EM-MS-DG-A-2F-01	InfoService/PositiveActiveEnergy	float	1	1	Input_Registers
Va_EM-MS-DG-A-2F-01	A相电压	EM-MS-DG-A-2F-01	InfoService/Va	float	1	1	Input_Registers
Vb_EM-MS-DG-A-2F-01	B相电压	EM-MS-DG-A-2F-01	InfoService/Vb	float	1	1	Input_Registers
Vc_EM-MS-DG-A-2F-01	C相电压	EM-MS-DG-A-2F-01	InfoService/Vc	float	1	1	Input_Registers
Ia_EM-MS-DG-A-2F-01	A相电流	EM-MS-DG-A-2F-01	InfoService/Ia	float	1	1	Input_Registers
Ib_EM-MS-DG-A-2F-01	B相电流	EM-MS-DG-A-2F-01	InfoService/Ib	float	1	1	Input_Registers
Ic_EM-MS-DG-A-2F-01	C相电流	EM-MS-DG-A-2F-01	InfoService/Ic	float	1	1	Input_Registers
ActivePowerSum_EM-MS-DG-A	瞬时总有功功率	EM-MS-DG-A-2F-01	InfoService/ActivePowerSum	float	1	1	Input_Registers
PFSum_EM-MS-DG-A-2F-01	总功率因数	EM-MS-DG-A-2F-01	InfoService/PFSum	float	1	1	Input_Registers
PositiveActiveEnergy_EM-M	正向有功总电能	EM-MS-DG-A-3F-01	InfoService/PositiveActiveEnergy	float	1	1	Input_Registers
Va_EM-MS-DG-A-3F-01	A相电压	EM-MS-DG-A-3F-01	InfoService/Va	float	1	1	Input_Registers
Vb_EM-MS-DG-A-3F-01	B相电压	EM-MS-DG-A-3F-01	InfoService/Vb	float	1	1	Input_Registers
Vc_EM-MS-DG-A-3F-01	C相电压	EM-MS-DG-A-3F-01	InfoService/Vc	float	1	1	Input_Registers
Ia_EM-MS-DG-A-3F-01	A相电流	EM-MS-DG-A-3F-01	InfoService/Ia	float	1	1	Input_Registers
Ib_EM-MS-DG-A-3F-01	B相电流	EM-MS-DG-A-3F-01	InfoService/Ib	float	1	1	Input_Registers
Ic_EM-MS-DG-A-3F-01	C相电流	EM-MS-DG-A-3F-01	InfoService/Ic	float	1	1	Input_Registers
ActivePowerSum_EM-MS-DG-A	瞬时总有功功率	EM-MS-DG-A-3F-01	InfoService/ActivePowerSum	float	1	1	Input_Registers
PFSum_EM-MS-DG-A-3F-01	总功率因数	EM-MS-DG-A-3F-01	InfoService/PFSum	float	1	1	Input_Registers

3. 下发配置，当单击下发配置后，连接状态为已连接，同步状态为已同步，即可完成数据采集

网关名称	连接源名称	连接状态	数据	驱动模块	最近修改时间	同步状态	操作
GW-MS-DG-A-2F-01 9903803633988918528	modbus_tcp_a_2f_1a...	已连接	171	user_Modbus	2024/07/10 17:36:38.114...	已同步	下发配置 编辑 删除

点位表

点位名称	点位名称	设备ID	设备名称	设备属性	更新时间	操作
Vc_EM-MS-DG-A-10F-01	C相电压	EM-MS-DG-A-10F-01	A栋设备10楼电压	InfoService/Vc	2024/07/10 17:35:556	编辑 删除

查看采集结果

1. 查看机电设备详情

【边缘管理】->【机电设备】

找到对应数采的机电设备，单击设备详情：

图 3-6 机电设备详情

设备ID	设备名称	设备属性	设备ID	设备名称	设备属性	设备ID	设备名称	设备属性
9903803633988918528	modbus_tcp_a_2f_1a...	已连接	9903803633988918528	modbus_tcp_a_2f_1a...	已连接	9903803633988918528	modbus_tcp_a_2f_1a...	已连接

2. 查看机电设备点位，跳转到机电设备详情页面，单击属性管理，就可以查看到当前这个机电设备点位上报值和上报时间，如果数采配置的点位都能看到有值，就说明该设备点位采集成功了，如下图所示：

图 3-7 机电设备属性管理

属性ID	属性名称	属性单位	当前值	单位	最近上报时间	操作
InfoService	FastAlarm	报警管理	无报警		2025/05/06 14:36:43.000 GMT+08:00	详情 编辑 删除
InfoService	WaterReturnTemp	冷却水回管温度采集设备	27	℃	2025/05/06 14:36:43.000 GMT+08:00	详情 编辑 删除
InfoService	RunState	运行状态	停止		2025/05/06 14:36:43.000 GMT+08:00	详情 编辑 删除
InfoService	SupplyAirHumTempDiff	供应空气湿度差	25.880001	℃	2025/05/06 14:36:43.000 GMT+08:00	详情 编辑 删除
InfoService	WaterReturnTemp	冷却水回管温度	0	℃	2025/05/06 14:36:43.000 GMT+08:00	详情 编辑 删除
InfoService	WaterReturnTemp	冷却水回管温度采集设备	0	℃	2025/05/06 14:36:43.000 GMT+08:00	详情 编辑 删除
InfoService	WaterSupplyTemp	冷却水供水温度	26.880001	℃	2025/05/06 14:36:43.000 GMT+08:00	详情 编辑 删除

3.4 模拟设备测试

模拟设备测试主要用于没有真实设备情况下，通过mqttfx模拟工具来模拟设备的在线和点位上报等功能。

3.4.1 使用 mqttfx 模拟物联网设备

操作步骤

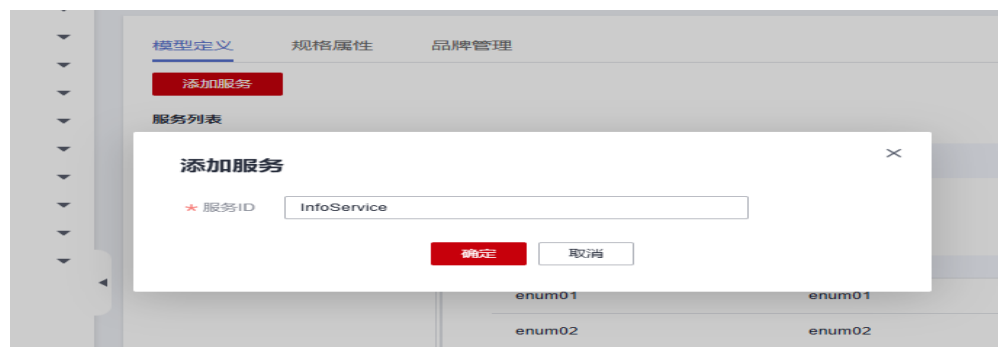
- 步骤1** 访问智能设施云平台管理控制台，选择“设备管理 > 产品”，在对应产品分类下创建一个产品，注意产品ID要唯一，协议类型必须为MQTT。

图 3-8 创建产品



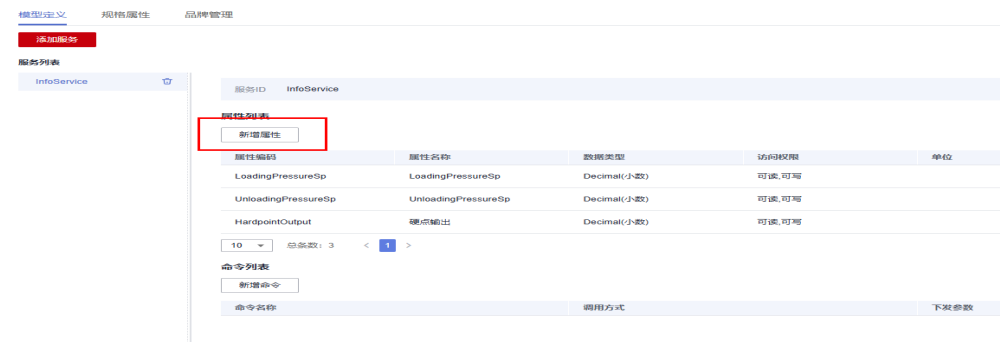
- 步骤2** 创建产品成功后单击详情页面，在该页面下添加服务ID，一般默认固定填写为InfoService。

图 3-9 添加服务



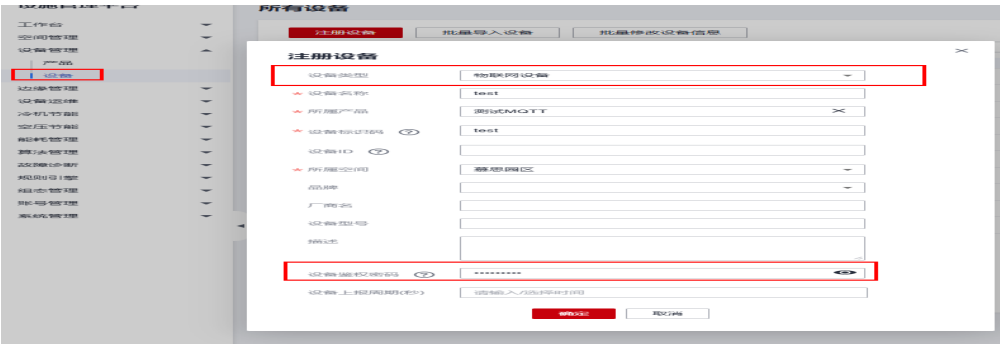
- 步骤3** 添加服务成功后可以为该服务添加对应的设备属性，具体属性名称和属性类型根据实际的设备属性来，操作如下：

图 3-10 添加属性



步骤4 接着在“设备”菜单注册对应设备，设备所属产品要选择上面创建的产品，设备类型为物联网设备，设备鉴权密码填写一个好记住的密码（后面连接mqtt需要用到），操作如下：

图 3-11 注册物联网设备



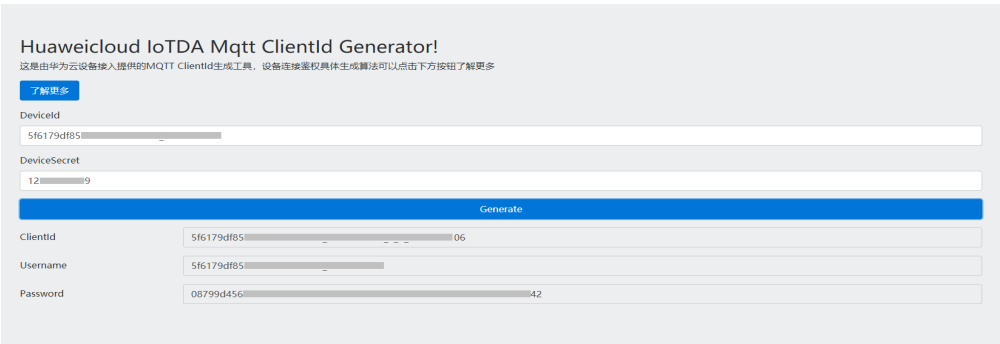
步骤5 注册成功后会弹出关闭保持框，单击确认后就会下载一个DEVICES-KEY.txt的文件，该文件记录好了该设备的secret值（对应的就是设备鉴权密码）。同时去设备详情里面查看设备id并记录下来。

步骤6 进入下面url下载安装MQTT.fx工具后进行安装：

<https://iotda-document.obs.cn-north-4.myhuaweicloud.com/mqttfx-1.7.1-windows-x64.exe>

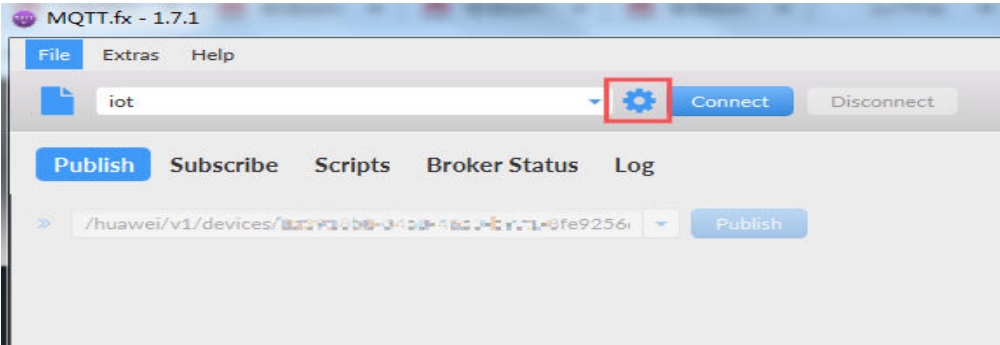
步骤7 访问<https://iot-tool.obs-website.cn-north-4.myhuaweicloud.com>这个地址，填写步骤4的设备ID（DeviceId）和设备的设备鉴权密码，密码签名类型为密钥认证，密码签名类型为不校验时间戳，生成连接信息（ClientId、Username、Password）。

图 3-12 clientID 生成工具



步骤8 打开MQTT.fx软件，单击设置图标。

图 3-13 设置按钮



步骤9 参考下表配置鉴权参数，然后单击“Apply”。

图 3-14 mqtt 连接配置

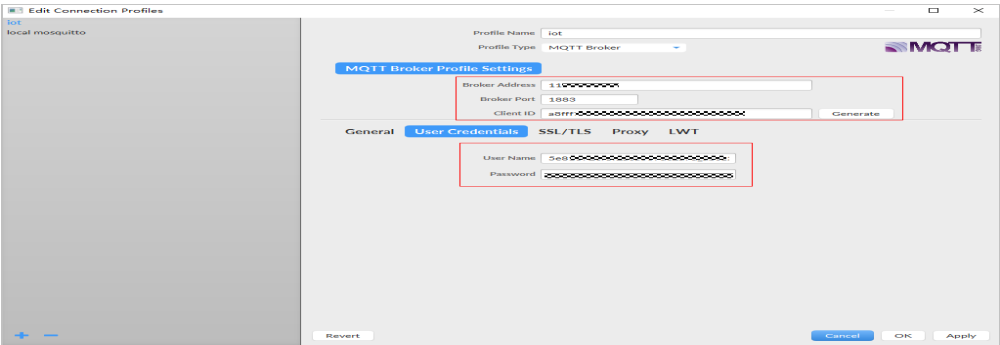


表 3-2 mqtt 连接参数

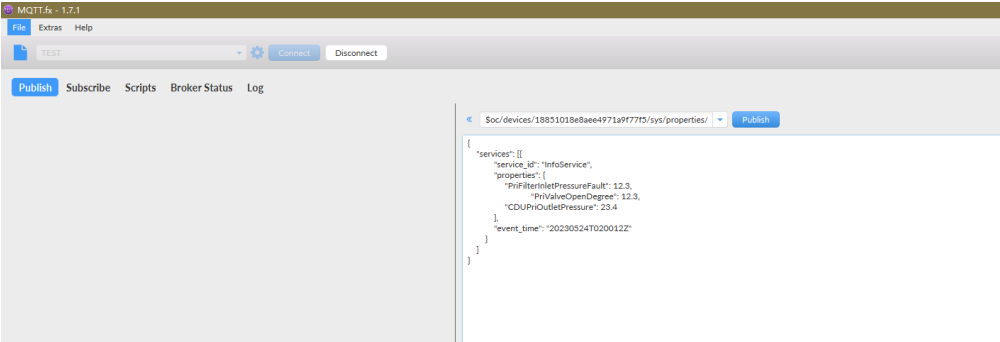
参数名称	说明
Broker Address	设备对接地址，此地址为域名信息。北京四伙伴POC测试环境可以使用 e6aeb02b19.st1.iotda-device.cn-north-4.myhuaweicloud.com
Broker Port	默认使用1883
CliendID	设备cliendID，请参考步骤7中获取。
User Name	即设备ID，请参考步骤7中获取。
Password	加密后的设备密钥，请参考步骤7中获取。

步骤10 单击“Connect”，设备鉴权成功后，在设施云平台“设备”界面可以看到对应物联网设备处于在线状态就说明模拟设备连接成功了。

步骤11 连接成功后就可以通过发布消息到对应设备的url就能上报该设备的点位数据，对应url如下：

“\$oc/devices/{device_id}/sys/properties/report”为例，如“\$oc/devices/5e4e2e92ac-164aefa8fouquan1/sys/properties/report”。

图 3-15 上报点位



步骤12 填写上报的数据格式如下：

```
{
  "services": [
    {
      "event_time": "20240914T081438Z",
      "properties": {
        "ElecQuality": 10.1
      },
      "service_id": "InfoService"
    }
  ]
}
```

表 3-3 表 1 设备服务数据列表

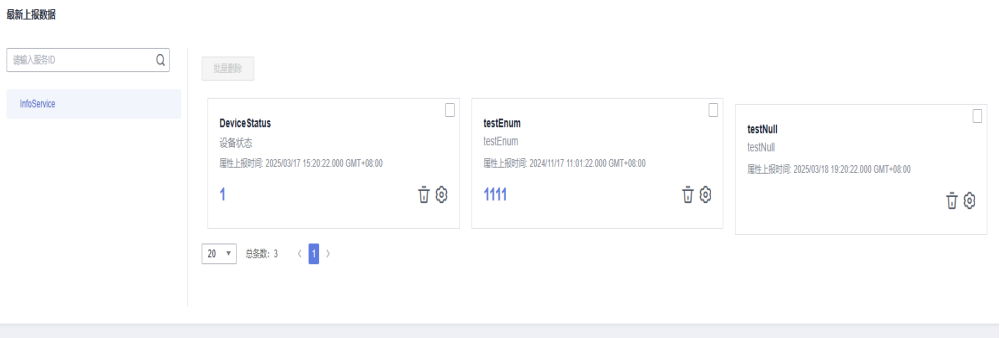
字段名	必选/可选	类型	参数描述
services	必选	List<ServiceProperty>	设备服务数据列表（具体结构参考下表 ServiceProperty定义表）

表 3-4 表 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。固定填写为InfoService.
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：yyyyMMddTHH:mm:ssZ），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。

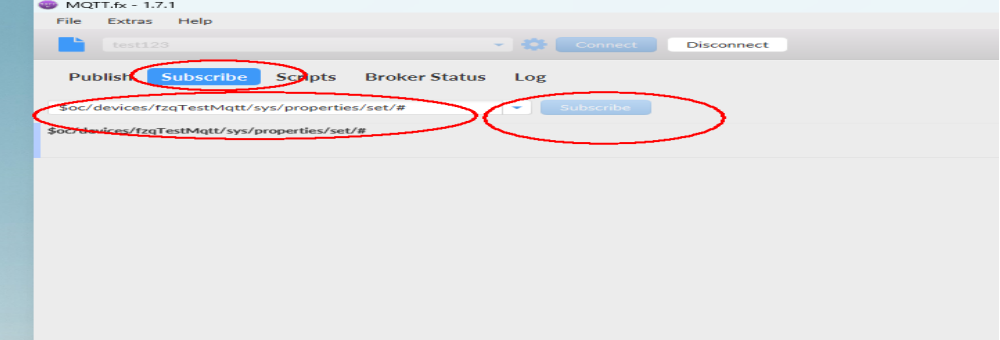
步骤13 单击“Publish”，可以在设施云设备详情界面上查看设备是否成功上报数据。

图 3-16 物联网设备点位



步骤14 如果要模拟设置点位，在mqtt.fx模拟工具打开订阅功能，订阅的topic为\$oc/devices/{device_id}/sys/properties/set/#，如下图所示：

图 3-17 订阅点位设置



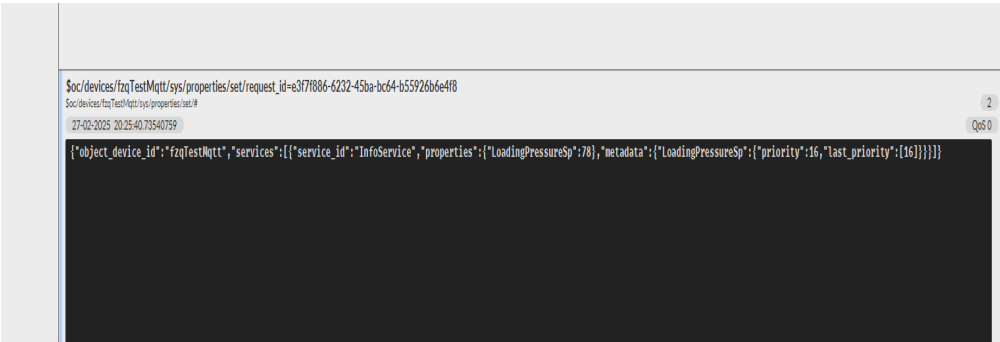
步骤15 接着在机电设备详情页面对应的属性管理页面，单击对应点位设置按钮来设置这个设备的点位进行下发：

图 3-18 设定点位值



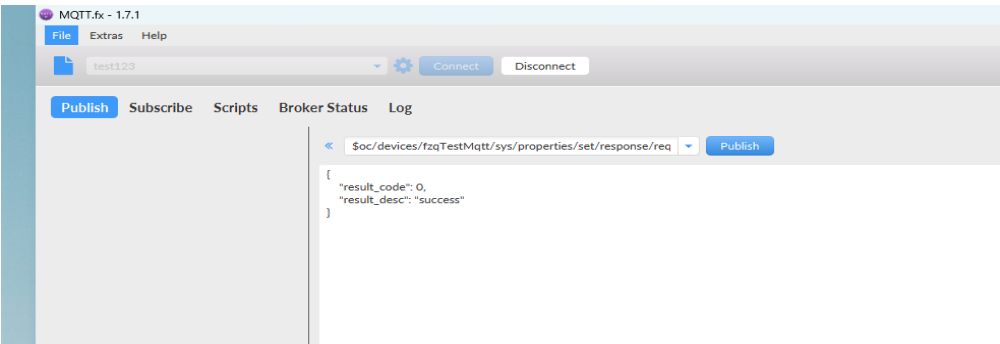
步骤16 设置成功后就可以监听到一个消息了，如下所示，可以看到一个后面有个request_id参数，获取这个参数值后，接着发布一个消息来响应设置点位成功。

图 3-19 监听点位设置信息



步骤17 发布的topic为\$oc/devices/{device_id}/sys/properties/set/response/request_id={request_id}，速度要快，不然就容易响应超时导致设置失败了。

图 3-20 响应点位设置



步骤18 如果需要进行其他IOTDA和设备的交互，IOTDA提供的基础topic可参照如下表格（仅适用于MQTT协议设备）。

表 3-5 topic 定义表

Topic	发布者	订阅者	用途
\$oc/devices/{device_id}/sys/messages/up	设备	平台	设备消息上报
\$oc/devices/{device_id}/sys/messages/down	平台	设备	平台下发消息给设备
\$oc/devices/{device_id}/sys/commands/request_id={request_id}	平台	设备	平台下发命令给设备
\$oc/devices/{device_id}/sys/commands/response/request_id={request_id}	设备	平台	设备返回命令响应
\$oc/devices/{device_id}/sys/properties/report	设备	平台	设备上报属性数据
\$oc/devices/{device_id}/sys/gateway/sub_devices/properties/report	设备	平台	网关批量上报属性数据
\$oc/devices/{device_id}/sys/properties/set/request_id={request_id}	平台	设备	平台设置设备属性

\$oc/devices/{device_id}/sys/properties/set/response/request_id={request_id}	设备	平台	属性设置的响应结果
\$oc/devices/{device_id}/sys/properties/get/request_id={request_id}	平台	设备	平台查询设备属性
\$oc/devices/{device_id}/sys/properties/get/response/request_id={request_id}	设备	平台	属性查询响应结果，这个结果不会对设备属性和影子产生影响
\$oc/devices/{device_id}/sys/shadow/get/request_id={request_id}	设备	平台	设备侧主动获取平台的设备影子数据
\$oc/devices/{device_id}/sys/shadow/get/response/request_id={request_id}	平台	设备	设备侧主动获取平台设备影子数据的响应
\$oc/devices/{device_id}/sys/events/up	设备	平台	设备事件上报
\$oc/devices/{device_id}/sys/events/down	平台	设备	平台事件下发

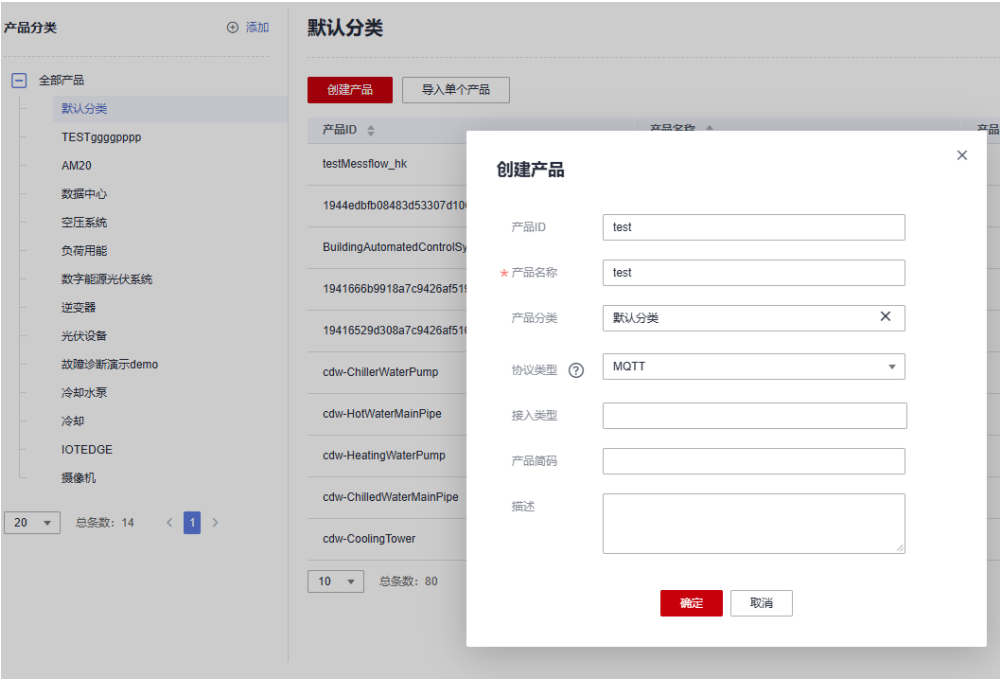
----结束

3.4.2 使用 mqttfx 模拟机电设备

操作步骤

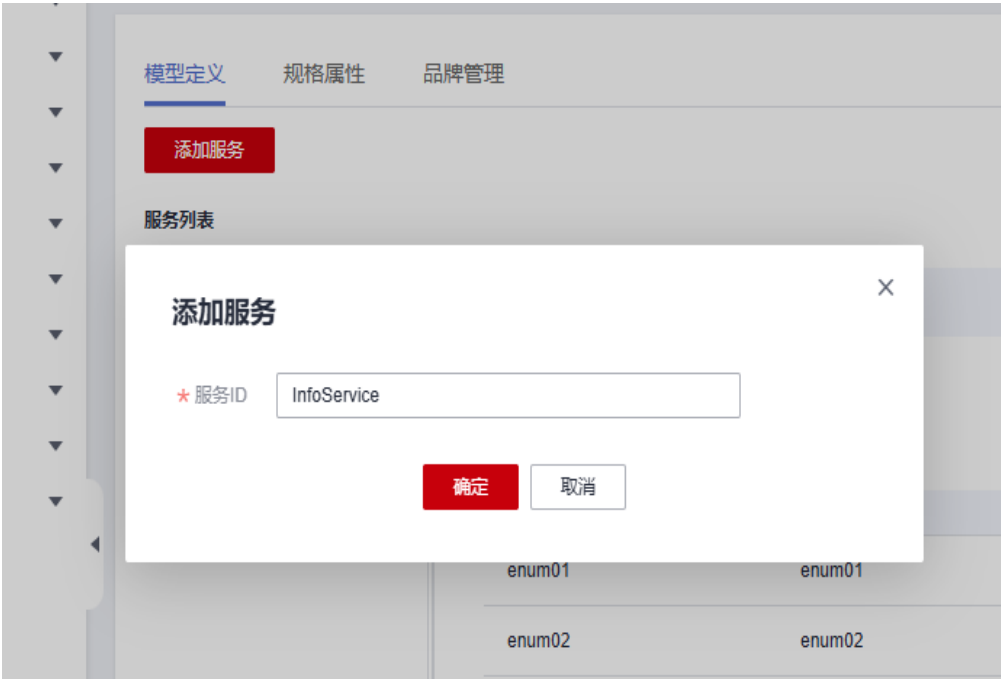
步骤1 访问智能设施云平台管理控制台，选择“设备管理 > 产品”，在对应产品分类下创建一个产品，注意产品ID要唯一，协议类型必须为MQTT。

图 3-21 创建产品



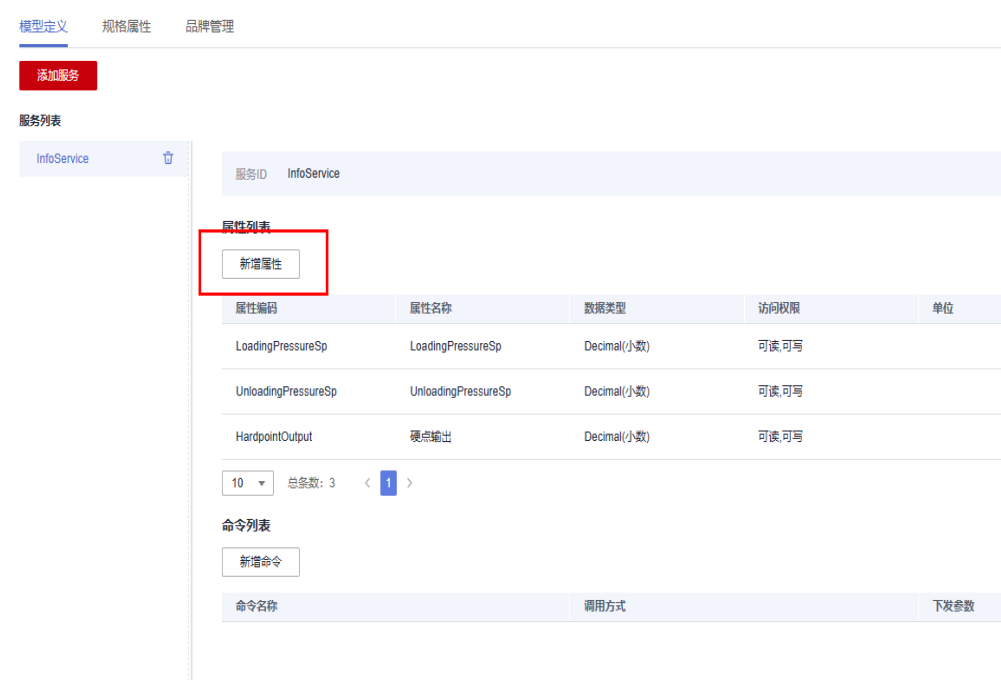
步骤2 创建产品成功后单击详情页面，在该页面下添加服务ID，一般默认固定填写为InfoService。

图 3-22 添加服务



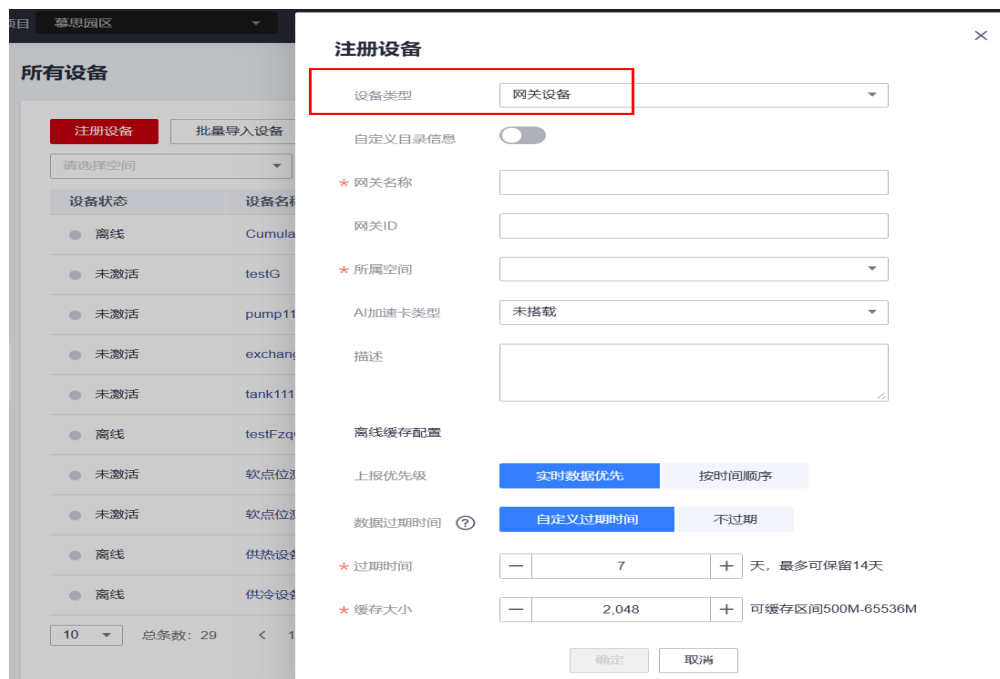
步骤3 添加服务成功后可以为该服务添加对应的设备属性，具体属性名称和属性类型根据自己的设备属性来，操作如下：

图 3-23 添加属性



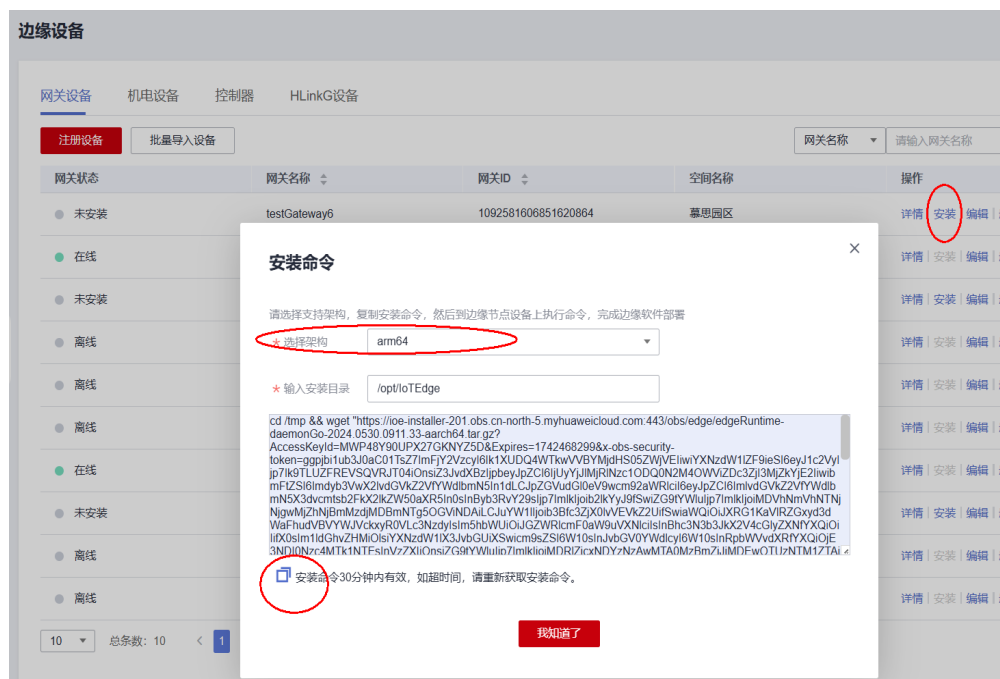
步骤4 接着在“设备”菜单注册对应一个网关设备，设备类型为网关设备，填写网关名称，其它使用默认值即可，操作如下：

图 3-24 注册网关设备



步骤5 准备一台连接公网的虚拟机，并且该虚拟机上安装好docker，接着点“边缘管理》边缘设备”菜单找到步骤4注册好网关后，单击后面的安装，选择对应架构就会生成安装命令，在虚拟机上复制对应安装命令，使用root执行复制的命令来安装网关，安装成功后该网关设备将变成在线状态，同时记录好虚拟机的IP地址（后面mqtt连接需要用到），对应页面如下：

图 3-25 安装网关



步骤6 在“设备”菜单注册对应设备，设备所属产品要选择上面创建的产品，设备类型为机电设备，边缘网关设备选择上面安装好的网关设备，设备鉴权密码填写一个好记住的密码（后面连接mqtt需要用到），操作如下：

图 3-26 注册机电设备

注册设备

设备类型 机电设备

★ 设备名称 test

★ 所属产品 测试MQTT

★ 设备标识码 ? test

★ 所属空间 慕思园区

品牌 品牌变动将导致设备属性变动，请谨慎修改。

厂商名

厂商编码

设备型号

描述

设备鉴权密码 ?

★ 边缘网关设备 ? 威能网关

关联模块 ?

前往创建

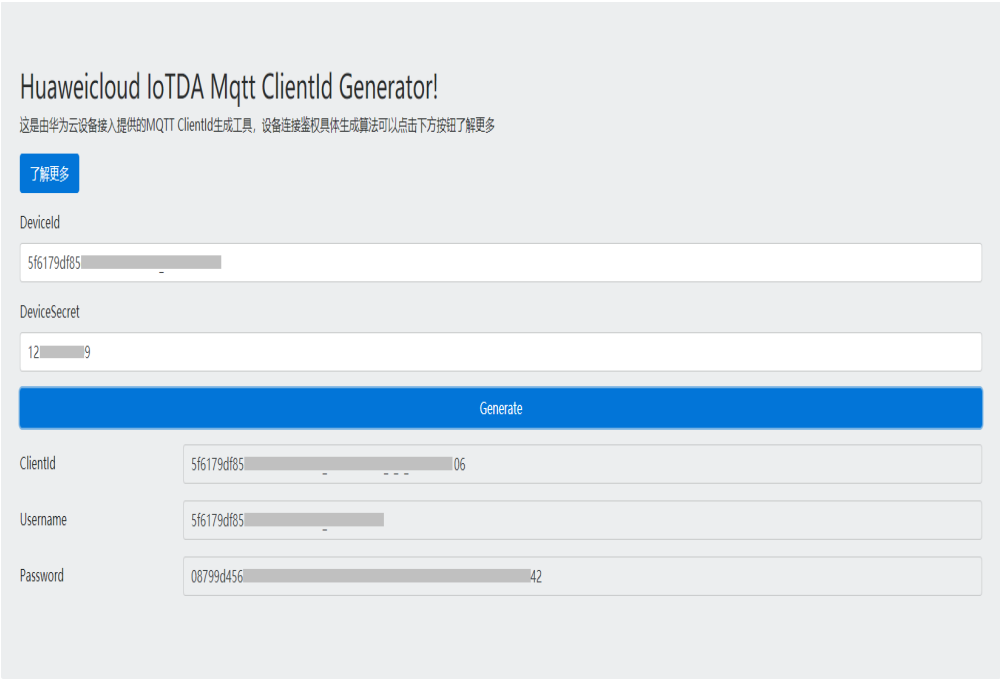
步骤7 注册成功后会弹出关闭保持框，单击确认后就会下载一个DEVICES-KEY.txt的文件，该文件记录好了该设备的secret值（对应的就是设备鉴权密码）。同时去设备详情里面查看设备id并记录下来。

步骤8 进入下面url下载安装MQTT.fx工具后进行安装：

<https://iotda-document.obs.cn-north-4.myhuaweicloud.com/mqttx-1.7.1-windows-x64.exe>

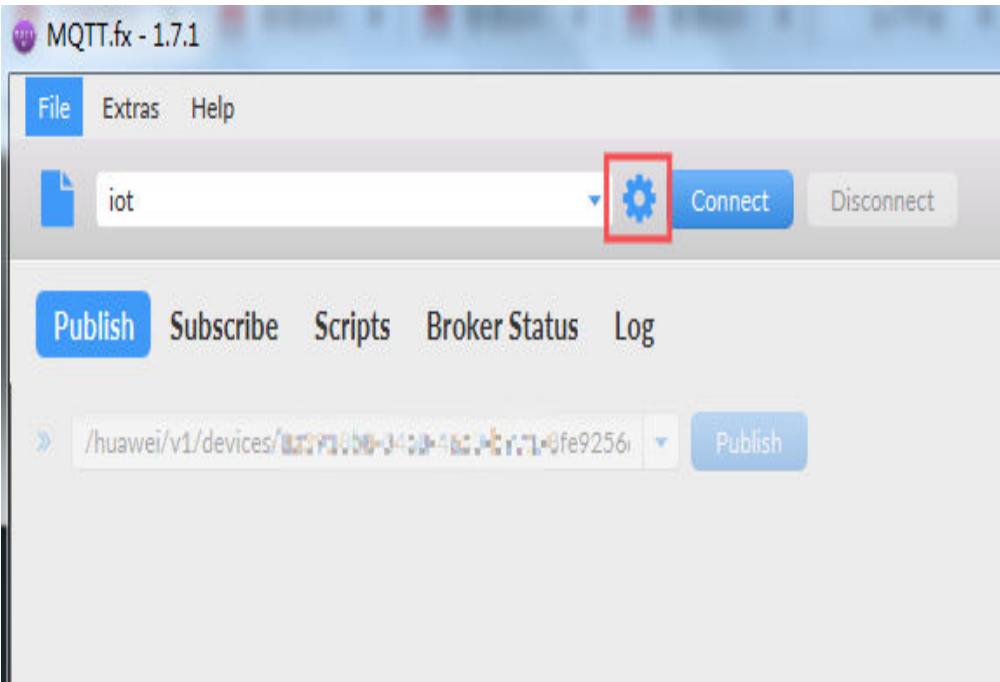
步骤9 访问<https://iot-tool.obs-website.cn-north-4.myhuaweicloud.com>这个地址，填写步骤4的设备ID（DeviceId）和设备的设备鉴权密码，密码签名类型为密钥认证，密码签名类型为不校验时间戳，生成连接信息（ClientId、Username、Password）。

图 3-27 clientID 生成工具



步骤10 打开MQTT.fx软件，单击设置图标。

图 3-28 设置按钮



步骤11 参考下表配置鉴权参数，然后单击“Apply”。

图 3-29 mqtt 连接配置

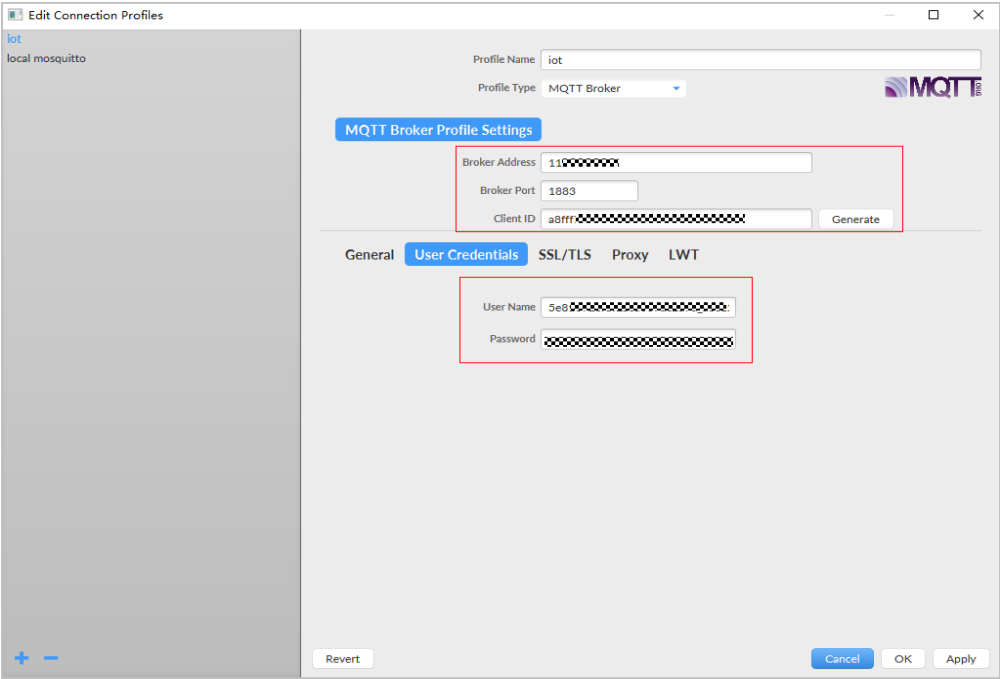
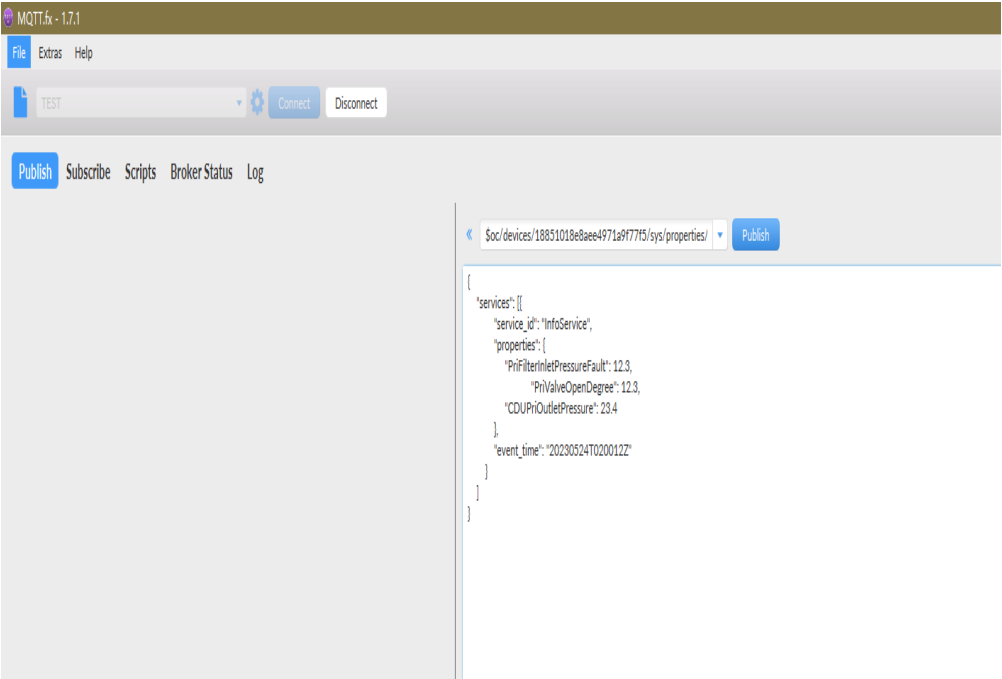


表 3-6 mqtt 连接参数

参数名称	说明
Broker Address	设备对接地址，此地址为安装网关虚拟机对外的IP地址。
Broker Port	默认使用7882
CliendID	设备cliendID，请参考步骤9中获取。
User Name	即设备ID，请参考步骤9中获取。
Password	加密后的9备密钥，请参考步骤7中获取。

- 步骤12 单击“Connect”，设备鉴权成功后，在设施云平台“设备”界面可以看到对应物联网设备处于在线状态就说明模拟设备连接成功了。
- 步骤13 连接成功后就可以通过发布消息到对应设备的url就能上报该设备的点位数据，对应url如下：
“\$oc/devices/{device_id}/sys/properties/report”为例，如“\$oc/devices/5e4e2e92ac-164aefa8fouquan1/sys/properties/report”。

图 3-30 上报点位



步骤14 填写上报的数据格式如下：

```
{
  "services": [
    {
      "event_time": "20240914T081438Z",
      "properties": {
        "ElecQuality": 10.1
      },
      "service_id": "InfoService"
    }
  ]
}
```

表 3-7 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 3-8 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。固定填写为InfoService.
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。

字段名	必选/可选	类型	参数描述
eventTime	可选	String	设备采集数据UTC时间（格式：yyyyMMddTHH:mm:ssZ），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。

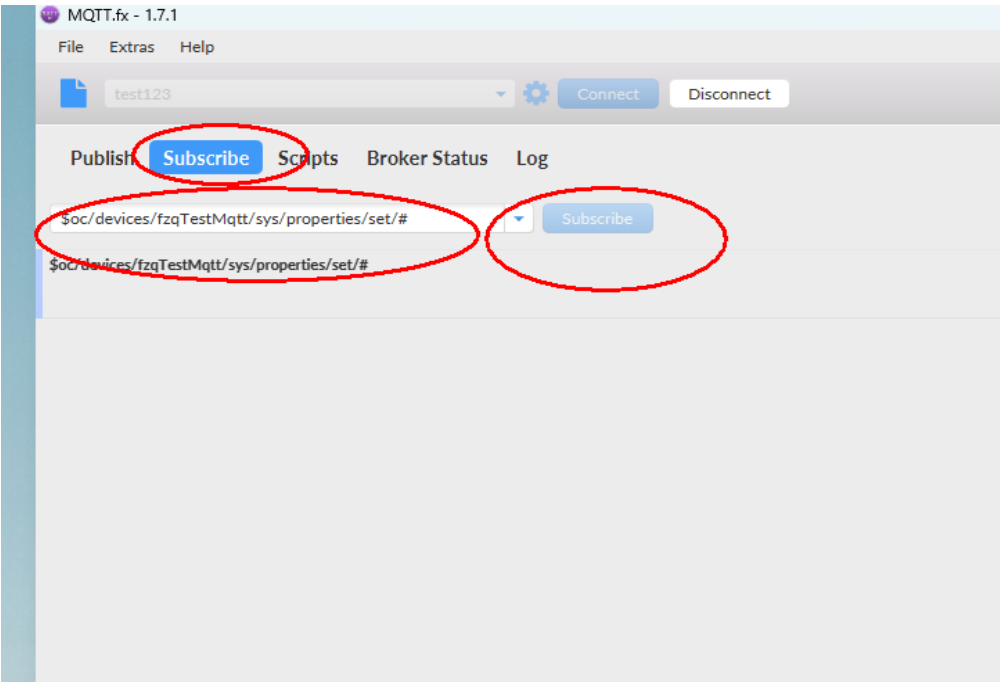
步骤15 单击“Publish”，可以在设施云设备详情界面上查看设备是否成功上报数据。

图 3-31 机电设备点位



步骤16 如果要模拟设置点位，在mqtt.fx模拟工具打开订阅功能，订阅的topic为\$oc/devices/{device_id}/sys/properties/set/#，如下图所示：

图 3-32 订阅点位设置



步骤17 接着在机电设备详情页面对应的属性管理页面，单击对应点位设置按钮来设置这个设备的点位进行下发：

图 3-33 设定点位值



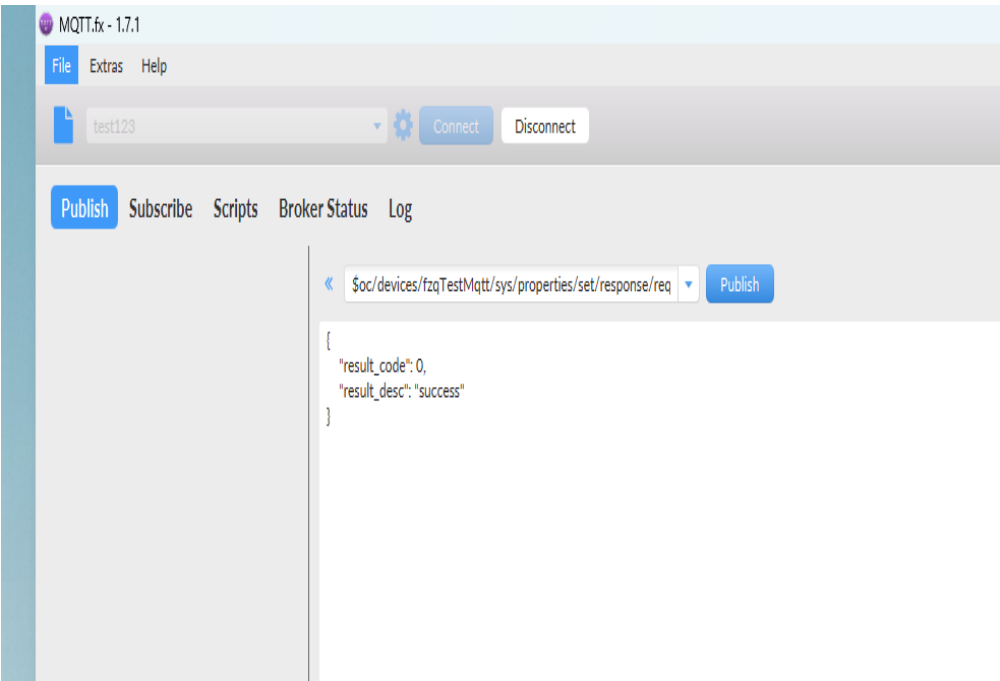
步骤18 设置成功后就可以监听到一个消息了，如下所示，可以看到一个后面有个request_id参数，获取这个参数值后，接着发布一个消息来响应设置点位成功。

图 3-34 监听点位设置信息



步骤19 发布的topic为\$oc/devices/{device_id}/sys/properties/set/response/request_id={request_id}，速度要快，不然就容易响应超时导致设置失败了。

图 3-35 响应点位设置



----结束