

OpenAPI 接口文档

1. 概述

物联网 OpenAPI 是一套高度标准化的 MQTT 接口规范，致力于构建物联网平台与第三方应用系统间高效、安全、可靠的数据交换通道。该规范充分利用 MQTT 协议的轻量级、低延迟特性，为设备全生命周期管理提供实时数据获取、精准状态监控及毫秒级事件通知等核心能力，全面支撑企业级设备管理、大数据分析与业务系统无缝集成，加速智能化物联网应用的构建与落地。

物联网 OpenAPI 创新性地采用“查询+推送”双模架构设计：通过主动查询满足按需数据获取，借助实时推送确保关键事件的即时触达，二者协同工作，既保证了系统资源的高效利用，又实现了关键业务场景下零延迟的事件响应机制，为物联网应用提供坚实的通信基础。

1.1 基本信息

- 协议：MQTT
- 数据格式：JSON
- 字符编码：UTF-8

1.2 说明

本文档是 MQTT 协议规范，HTTP 接口规范请参阅 HTTP 协议文档。

在接口设计上，HTTP 协议因其直观的请求-响应模式，通常展现出更高的实用性和易用性，尤其适用于平台对外开放接口场景，调用过程简单直观。

相比之下，MQTT 协议的交互模式要求客户端先发布请求消息，再异步监听平台响应，这种双向通信机制虽灵活但实现复杂度较高。

因此，MQTT 更适合以下特定场景：

企业级防火墙穿透环境，在严格网络隔离的企业环境中，MQTT 可有效规避多端口开放策略限制，实现跨安全域的安全通信。

如果觉得 MQTT 协议的交互模式过于繁琐，可以 HTTP 与 MQTT 混合使用。

查询接口 → 使用 HTTP 配置管理 → 使用 HTTP

推送功能 → 使用 MQTT（也可以 HTTP，MQTT 主要应用于企业防火墙的特殊环境）

非企业防火墙限制等特殊情况，还是建议使用 HTTP 的方式

[HTTP 文档请联系管理员](#)

混合模式既满足了特殊环境的限制，又简化了第三方系统集成复杂度，为企业提供灵活、高效的物联网连接方案。

2. 主题规范

2.1 认证方式

MQTT 无需认证。

请求时携带生成的 AppKey 即可。具体使用详见下面各个接口。

2.2 主题规范

2.2.1 请求主题

格式：/openapi/data/request/{appKey}

示例：/openapi/data/request/ABC123456

第三方向此主题请求消息，平台会收到请求

2.2.2 响应主题

格式：/openapi/data/response/{appKey}

示例：/openapi/data/response/ABC123456

服务端向此主题发布响应消息，第三方需订阅此主题接收响应

2.2.3 定时推送主题

格式：/openapi/data/scheduled/{appKey}

示例：/openapi/data/scheduled/ABC123456

服务端定时推送数据（设备状态、实时数据等）时使用此主题

2.2.4 订阅主题说明

请求主题为第三方向平台调用，无需订阅。

但是响应主题和定时推送主题，是平台向第三方发布消息。第三方需要订阅这 2 个主题才能接收到消息。

所以第三方需要订阅 2 个主题

/openapi/data/response/{appKey} 响应主题

/openapi/data/scheduled/{appKey} 定时推送主题

不能使用通配符进行订阅，使用通配符订阅会提示无法订阅

2.3 消息格式

2.3.1 请求消息格式

```
{  
  "messageId": "uuid-1234-5678-abcd", // 建议每次生成 UUID  
  "appKey": "ABC123456",  
  "method": "queryRegions",  
  "data": {  
    // 请求参数，根据不同方法而定  
  }  
}
```

参数	数据类型	是否必填	描述
messageId	String	是	消息唯一标识，用于关联请求和响应。 响应时会携带这个消息 ID 建议使用 UUID
appKey	String	是	appKey
method	String	是	调用的方法名

参数	数据类型	是否必填	描述
data	Object	否	请求参数，根据不同方法而定

2.3.2 响应消息格式

成功响应

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryRegions",
  "data": {
    "code": 200,
    "message": "success",
    "data": [
      // 业务数据
    ]
  }
}
```

错误响应

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryRegions",
  "data": {
    "code": 500,
    "message": "AppKey 不存在",
    "data": null
  }
}
```

参数	数据类型	描述
----	------	----

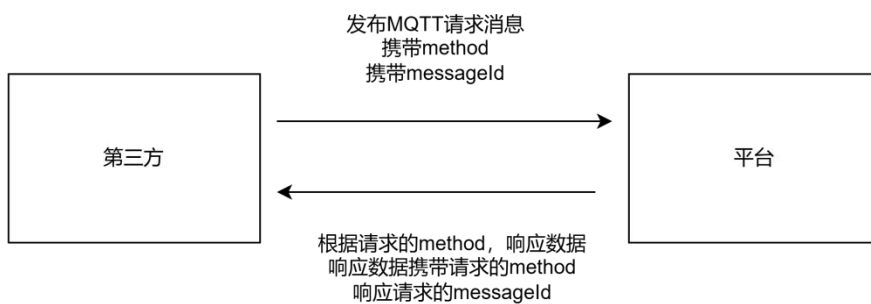
参数	数据类型	描述
messageId	String	与请求消息的 messageId 对应
appKey	String	appKey
method	String	调用的方法名
data	Object	请求参数，根据不同方法而定
data.code	Integer	响应码：200-成功，500-失败
data.message	String	响应消息
data.data	Object	业务数据，成功时返回实际数据，失败时为 null

3. 数据接口

数据接口是第三方向平台调用。

第三方向平台发送 MQTT 消息，平台给第三方响应 MQTT 接口数据。

详情 2.2 2.3



3.1 查询区域列表

该接口返回所有区域

请求参数：无

方法名 (method) : queryRegions

MQTT 请求主题 /openapi/data/request/{appKey}

```
{  
  "messageId": "uuid-1234-5678-abcd",  
  "appKey": "ABC123456",  
  "method": "queryRegions",  
  "data": { }  
}
```

MQTT 响应主题 /openapi/data/response/{appKey}

```
{  
  "messageId": "uuid-1234-5678-abcd",  
  "appKey": "ABC123456",  
  "method": "queryRegions",  
  "data": {  
    "code": 200,  
    "message": "success",  
    "data": [  
      {  
        "regionId": 1,  
        "regionName": "一级区域",  
        "parentId": 0,  
        "parentName": null,  
        "remark": "备注"  
      }  
    ],  
    "timestamp": 1732694400000  
  }  
}
```

响应参数	数据类型	示例	描述
regionId	Long	719943688798277	区域 ID

响应参数	数据类型	示例	描述
regionName	String	一级区域	区域名称
parentId	Long	719943688798270	父区域 ID
parentName	String	顶级区域	父区域名称
remark	String	备注	备注

3.2 查询设备列表

方法名 (method) : queryDeviceList

MQTT 请求主题 /openapi/data/request/{appKey}

请求体参数:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryDeviceList",
  "data": {
    "regionIds": [1, 2], // 可选, 为空则返回所有设备
    "deviceIds": [1, 2], // 可选, 设备 id
    "businessIds": [1, 2], // 可选, 设备行业 id, 例如 IMEI, mac 地址等设备的唯一参数
    "page": 1, // 必填, 分页参数
    "size": 10 // 必填, 分页参数
  }
}
```

根据查询参数, 查询设备列表。

单页最多 1000 个设备。如果设备不多的情况, 可以一次性查询全部。

请求参数	数据类型	示例	必填	描述
regionIds	整型 List	[1, 2]	否	区域 ID 集合, 可以单个
deviceIds	整型 List	[1, 2]	否	设备 ID 集合, 可以单个
businessIds	整型 List	[1, 2]	否	行业 ID 集合, 可以单个

响应示例:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryDeviceList",
  "data": {
    "code": 200,
    "message": "success",
    "data": [
      {
        "deviceId": 123,
        "businessId": "IMEI123456",
        "deviceName": "设备 001",
        "productId": 1,
        "productName": "4G 终端",
        "regionId": 1,
        "regionName": "一级区域",
        "onlineStatus": 1,
        "deviceStatus": 0,
        "activeTime": "2025-01-01 10:00:00",
        "lastOnlineTime": "2025-11-27 16:00:00",
        "remark": "备注"
      }
    ],
    "timestamp": 1732694400000
  }
}
```

响应参数	数据类型	示例	描述
deviceId	Long	719943688798277	区域 ID
businessId	String	一级区域	行业 ID
deviceName	String	719943688798270	设备名称
productId	Long	719943688798271	产品 ID
productName	String	产品名称	产品名称
regionId	Long	719943688795245	区域 ID
regionName	String	区域名称	区域名称

响应参数	数据类型	示例	描述
onlineStatus	Integer	0	在线状态, 0-在线, 1-离线
alarmStatus	Integer	0	告警状态, 0-正常, 1-告警
activeTime	String	2025-01-01 14:00:00	激活时间, 设备激活时间
offlineTime	String	2025-01-01 14:00:00	离线时间,如果设备在线, 为空串
remark	String	备注	备注

3.3 查询单个设备信息

方法名 (method) : queryDevice

MQTT 请求主题 /openapi/data/request/{appKey}

请求参数:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryDevice",
  "data": {
    "deviceId": 1 //必填, 设备 ID
  }
}
```

响应示例:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryDevice",
  "data": {
    "code": 200,
    "message": "success",
    "data": {
      "deviceId": 123,
      "businessId": "IMEI123456",
      "deviceName": "设备 001",
      "productId": 1,
      "productName": "4G 终端",
      "regionId": 1,
      "regionName": "一级区域",
    }
  }
}
```

```

        "onlineStatus": 1,
        "deviceStatus": 0,
        "activeTime": "2025-01-01 10:00:00",
        "lastOnlineTime": "2025-11-27 16:00:00",
        "remark": "备注"
    },
    "timestamp": 1732694400000
}
}

```

3.4 查询设备实时数据

方法名 (method) : queryDeviceListRealtime

MQTT 请求主题 /openapi/data/request/{appKey}

请求体参数:

```

{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryDeviceListRealtime",
  "data": {
    "regionIds": [1, 2], // 可选, 为空则返回所有设备
    "deviceIds": [1, 2], // 可选, 设备 id
    "businessIds": [1, 2] // 可选, 设备行业 id, 例如 IMEI, mac 地址等设备的唯一参数
    "page": 1, // 必填, 分页参数
    "size": 10 // 必填, 分页参数
  }
}

```

根据查询参数, 返回单个或者多个设备的实时数据。

单页最多 1000 个设备的实时数据。如果设备不多的情况, 可以一次性查询全部。

请求参数	数据类型	示例	必填	描述
regionIds	整型 List	[1, 2]	否	区域 ID 集合, 可以单个
deviceIds	整型 List	[1, 2]	否	设备 ID 集合, 可以单个
businessIds	整型 List	[1, 2]	否	行业 ID 集合, 可以单个

响应示例:

```

{
  "messageId": "uuid-1234-5678-abcd",

```

```

"appKey": "ABC123456",
"method": "queryDeviceListRealtime",
"data": {
  "code": 200,
  "message": "success",
  "data": {
    "deviceId": 123,
    "deviceName": "设备 001",
    "businessId": "IMEI123456",
    "regionId": 1,
    "regionName": "一级区域",
    "onlineStatus": 1,
    "metrics": [
      {
        "meteId": "1555",
        "meteName": "温度",
        "value": "25.5",
        "sValue": "25.5",
        "unit": "°C",
        "time": "2024-11-27 16:00:00"
      },
      {
        "meteId": "12125",
        "meteName": "湿度",
        "value": "60",
        "sValue": "60",
        "unit": "%",
        "time": "2024-11-27 16:00:00"
      }
    ],
    "timestamp": 1732694400000
  },
  "timestamp": 1732694400000
}
}

```

响应参数	数据类型	示例	描述
deviceId	Long	719943688798277	区域 ID
businessId	String	一级区域	行业 ID
deviceName	String	719943688798270	设备名称

响应参数	数据类型	示例	描述
productId	Long	719943688798271	产品 ID
productName	String	产品名称	产品名称
regionId	Long	719943688795245	区域 ID
regionName	String	区域名称	区域名称
onlineStatus	Integer	0	在线状态，0-在线，1-离线
metrics	List		
metelId	String	1855521	量 id
meteName	String	温度	量名称
value	String	26.5	实时值
unit	String	℃	单位
time	String	2025-01-01 14:00:00	时间
sValue	String	翻译之后的实时值	如果是枚举类型，就是翻译之后的实时值。 如果是字符串，和 value 等值

3.5 查询单个设备实时数据

方法名 (method) : queryDeviceRealtime

MQTT 请求主题 /openapi/data/request/{appKey}

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryDeviceRealtime",
  "data": {
    "deviceId": 1 // 必填
  }
}
```

deviceId: 设备 ID

响应示例:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryDeviceRealtime",
  "data": {
    "code": 200,
    "message": "success",
    "data": {
      "deviceId": 123,
      "deviceName": "设备 001",
      "businessId": "IMEI123456",
      "regionId": 1,
      "regionName": "一级区域",
      "onlineStatus": 1,
      "metrics": [
        {
          "meteId": "1555",
          "meteName": "温度",
          "value": "25.5",
          "sValue": "25.5",
          "unit": "°C",
          "time": "2024-11-27 16:00:00"
        },
        {
          "meteId": "12125",
          "meteName": "湿度",
          "value": "60",
          "unit": "%",
          "time": "2024-11-27 16:00:00"
        }
      ]
    },
    "timestamp": 1732694400000
  }
}
```

响应参数	数据类型	示例	描述
deviceId	Long	719943688798277	区域 ID
businessId	String	一级区域	行业 ID
deviceName	String	719943688798270	设备名称

响应参数	数据类型	示例	描述
productId	Long	719943688798271	产品 ID
productName	String	产品名称	产品名称
regionId	Long	719943688795245	区域 ID
regionName	String	区域名称	区域名称
onlineStatus	Integer	0	在线状态，0-在线，1-离线
metrics	List		
metelId	String	1855521	量 id
meteName	String	温度	量名称
value	String	26.5	实时值
unit	String	℃	单位
time	String	2025-01-01 14:00:00	时间
sValue	String	翻译之后的实时值	如果是枚举类型，就是翻译之后的实时值。 如果是字符串，和 value 等值

3.6 获取设备的在线状态

方法名 (method) : queryDeviceOnlineStatusList

MQTT 请求主题 /openapi/data/request/{appKey}

请求体参数:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryDeviceOnlineStatusList",
  "data": {
    "regionIds": [1, 2], // 可选,为空则返回所有设备
    "deviceIds": [1, 2], // 可选,设备 id
    "businessIds": [1, 2], // 可选,设备行业 id, 例如 IMEI,mac 地址等设备的唯一参数
    "page": 1, // 必填, 分页参数
    "size": 10 // 必填, 分页参数
  }
}
```

```
}  
}
```

根据查询参数，返回单个或者多个设备的实时数据。

单页最多 1000 个设备。如果设备不多的情况，可以一次性查询全部。

请求参数	数据类型	示例	必填	描述
regionIds	整型 List	[1, 2]	否	区域 ID 集合，可以单个
deviceIds	整型 List	[1, 2]	否	设备 ID 集合，可以单个
businessIds	整型 List	[1, 2]	否	行业 ID 集合，可以单个

响应示例:

```
{  
  "messageId": "uuid-1234-5678-abcd",  
  "appKey": "ABC123456",  
  "method": "queryDeviceOnlineStatusList",  
  "data": {  
    "code": 200,  
    "message": "success",  
    "data": {  
      "deviceId": 123,  
      "deviceName": "设备 001",  
      "businessId": "IMEI123456",  
      "onlineStatus": 1  
    },  
    "timestamp": 1732694400000  
  }  
}
```

响应参数	数据类型	示例	描述
deviceId	Long	719943688798277	区域 ID
businessId	String	一级区域	行业 ID
deviceName	String	719943688798270	设备名称
onlineStatus	Integer	0	设备状态 0-在线 1-离线

3.7 查询实时告警数据

方法名 (method) : queryRealtimeAlarms

MQTT 请求主题 /openapi/data/request/{appKey}

请求体参数:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryRealtimeAlarms",
  "data":{
    "regionIds": [1, 2], // 可选,为空则返回所有设备
    "deviceIds": [1, 2],// 可选,设备 id
    "businessIds": [1, 2] // 可选,设备行业 id, 例如 IMEI,mac 地址等设备的唯一参数
  }
}
```

根据查询参数, 返回单个或者多个设备的实时告警

这个接口不是分页, 一次查询全部的实时告警

请求参数	数据类型	示例	必填	描述
regionIds	整型 List	[1, 2]	否	区域 ID 集合, 可以单个
deviceIds	整型 List	[1, 2]	否	设备 ID 集合, 可以单个
businessIds	整型 List	[1, 2]	否	行业 ID 集合, 可以单个

响应示例:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryRealtimeAlarms",
  "data": {
    "code": 200,
    "message": "success",
    "data": [
      {
        "alarmId": 456,
        "deviceId": 123,
        "deviceName": "设备 001",
        "businessId": "IMEI123456",
        "regionId": 1,
      }
    ]
  }
}
```

```

        "regionName": "一级区域",
        "meteId": "80015551",
        "meteName": "压力过低",
        "level": 1,
        "value": "155",
        "sValue": "155",
        "alarmTime": "2025-11-2716:00:00"
    }
],
"timestamp": 1732694400000
}
}

```

响应参数	数据类型	示例	描述
alarmId	Long	15454521	告警恢复时，会以这个 Id 为主键
deviceId	Long	719943688798277	区域 ID
businessId	String	一级区域	行业 ID
deviceName	String	719943688798270	设备名称
regionId	Long	719943688795245	区域 ID
regionName	String	区域名称	区域名称
meteld	String	1545451	告警量 ID
meteName	String	温度过低	告警量名称
level	Integer	1	告警等级 1 2 3 4
value	String	22.5	实时值
sValue	String	22.5	如果是枚举类型，就是翻译之后的实时值。 如果是字符串，和 value 等值
alarmTime	String	2025-11-27 16:00:00	告警时间

3.8 查询单个设备历史告警

历史告警较多，所以仅限单个设备查询历史告警

方法名 (method) : queryDeviceHistoryAlarms
MQTT 请求主题 /openapi/data/request/{appKey}

POST /openapi/query/device/alarms/history
请求体参数:

```
{  
  "messageId": "uuid-1234-5678-abcd",  
  "appKey": "ABC123456",  
  "method": "queryDeviceHistoryAlarms",  
  "data": {  
    "deviceId": 1, // 可选, 设备 id  
    "businessId": 1 // 可选, 设备行业 id。但是二者必须其一  
    "page": 1, // 必填, 分页参数  
    "size": 10 // 必填, 分页参数  
  }  
}
```

请求参数	数据类型	示例	必填	描述
deviceId	Long	1	否	设备 ID
businessId	Long	1	否	行业 ID。行业 ID 和设备 ID 必填其中一个

响应示例:

```
{  
  "messageId": "uuid-1234-5678-abcd",  
  "appKey": "ABC123456",  
  "method": "queryDeviceHistoryAlarms",  
  "data": {  
    "code": 200,  
    "message": "success",  
    "data": [  
      {  
        "alarmId": 456,  
        "deviceId": 123,  
        "deviceName": "设备 001",  
        "businessId": "IMEI123456",  
        "regionId": 1,  
        "regionName": "一级区域",  
        "meteId": "80015551",  
      }  
    ]  
  }  
}
```

```

        "meteName": "压力过低",
        "level": 1,
        "value": "155",
        "sValue": "155",
        "alarmTime": "2025-11-2716:00:00",
        "state": "END",
        "closeTime": "2025-11-2718:00:00"
    },
    {
        "alarmId": 456,
        "deviceId": 123,
        "deviceName": "设备 001",
        "businessId": "IMEI123456",
        "regionId": 1,
        "regionName": "一级区域",
        "meteId": "80015551",
        "meteName": "压力过低",
        "level": 1,
        "value": "155",
        "sValue": "155",
        "alarmTime": "2025-11-2716:00:00",
        "state": "BEGIN",
        "closeTime": ""
    }
],
"timestamp": 1732694400000
}
}

```

响应参数	数据类型	示例	描述
alarmId	Long	15454521	告警恢复时，会以这个 Id 为主键
deviceId	Long	719943688798277	区域 ID
businessId	String	一级区域	行业 ID
deviceName	String	719943688798270	设备名称
regionId	Long	719943688795245	区域 ID
regionName	String	区域名称	区域名称

响应参数	数据类型	示例	描述
meteld	String	1545451	告警量 ID
meteName	String	温度过低	告警量名称
level	Integer	1	告警等级 1 2 3 4
value	String	22.5	实时值
sValue	String	22.5	如果是枚举类型，就是翻译之后的实时值。 如果是字符串，和 value 等值
alarmTime	String	2025-11-27 16:00:00	告警时间
state	String	BEGIN	BEGIN 或者 END BEGIN 是告警中 END 是已关闭
closeTime	String	2025-11-27 18:00:00	恢复时间，如果是 END 才会有 closeTime

3.9 查询产品列表

查看产品列表，产品包含物模型

方法名 (method) : queryProductList

MQTT 请求主题 /openapi/data/request/{appKey}

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryProductList",
  "data": { }
}
```

请求参数：无

响应示例：

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
```

```
"method": "queryProductList",
"data": {
  "code": 200,
  "message": "success",
  "timestamp": 1764749697528,
  "data": [
    {
      "productId": 744000000001000,
      "productName": "门磁",
      "onlinePeriod": 1440,
      "thingModel": [
        {
          "meteId": "1",
          "meteName": "门磁状态",
          "unit": "",
          "dataType": 3,
          "meteType": 0,
          "enumValues": [
            {
              "name": "门磁关门",
              "value": "0"
            },
            {
              "name": "门磁打开",
              "value": "1"
            }
          ],
          "level": 0,
          "decimalPoint": 1
        },
        {
          "meteId": "2",
          "meteName": "电池电压",
          "unit": "",
          "dataType": 2,
          "meteType": 0,
          "level": 0,
          "decimalPoint": 1
        }
      ]
    }
  ]
}
```

响应参数	数据类型	示例	描述
productId	Long	15454521	告警恢复时，会以这个 Id 为主键
productName	String	719943688798277	区域 ID
onlinePeriod	String	60	行业 ID
thingModel	List		设备名称
meteld	Long	1545451	量 ID
meteName	String	温度过低	量名称
unit	String	°C	单位
dataType	Integer	1	数据类型 0-字符串 1-整型 2-浮点 3-枚举
meteType	Integer	1	量类型 0-采集量 1-告警量 2-控制量 3-配置量 4-阈值量
level	Integer	1	告警等级 1 2 3 4
decimalPoint	Integer	2	小数点位数 1 2 3 4 1- 4 位
enumValues	List		枚举值定义，详见上面的示例

4.1 查询定时推送频率

查询定时推送的频率，包括实时数据推送频率，设备状态推送频率。

定时推送只会推送实时数据和设备状态。实时数据默认 30 分钟推送一次，设备状态默认 10 分钟推送一次。

告警状态仅在告警变换时才会触发一次。

方法名 (method) : queryScheduleConfig
 MQTT 请求主题 /openapi/data/request/{appKey}
 {
 "messageId": "uuid-1234-5678-abcd",
 "appKey": "ABC123456",

```
"method": "queryScheduleConfig",
"data": { }
}
```

请求参数：无

响应示例：

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "queryScheduleConfig",
  "data": {
    "code": 200,
    "message": "success",
    "data": {
      "datePushPeriod": 30,
      "onlineStatusPushPeriod": 10,
      "dataPushStatus": 0,
      "statusPushStstus": 0
    },
    "timestamp": 1764931447717
  }
}
```

响应参数	数据类型	示例	描述
datePushPeriod	Integer	30	推送频率周期，单位分钟
onlineStatusPushPeriod	Integer	30	推送频率周期，单位分钟
dataPushStatus	Integer	0	0-正常推送 1-停止推送
statusPushStstus	Integer	0	0-正常推送 1-停止推送

4.2 配置定时推送频率

方法名 (method) : configScheduleFrequency
MQTT 请求主题 /openapi/data/request/{appKey}

请求体参数：

```
{
```

```

    "messageId": "uuid-1234-5678-abcd",
    "appKey": "ABC123456",
    "method": "configScheduleFrequency",
    "data": {
      "datePushPeriod": 30,
      "onlineStatusPushPeriod": 10
    }
  }
}

```

请求参数	数据类型	示例	必填	描述
datePushPeriod	Integer	30	否	数据推送频率周期。单位分钟
onlineStatusPushPeriod	Integer	30	否	在线状态推送频率周期。单位分钟

可以只配置一个参数，例如只填 datePushPeriod, onlineStatusPushPeriod。
 onlineStatusPushPeriod: 10 或者 datePushPeriod: 10

响应示例：

```

{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "configScheduleFrequency",
  "data": {
    "code": 200,
    "message": "success",
    "data": null,
    "timestamp": 1732694400000
  }
}

```

4.4 开启/停止推送（全局推送）

这个是全局推送的开启或者关闭。

开启后，所以推送正常推送。默认是推送，配置推送地址之后默认开启推送。

关闭后，所有类型的推送，还有定时的所有推送，都会停止推送。

方法名（method）：setPushStatus

MQTT 请求主题 /openapi/data/request/{appKey}

请求体参数:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "setPushStatus",
  "data": {
    "pushStatus": 0
  }
}
```

请求参数	数据类型	示例	必填	描述
pushStatus	Integer	0	是	0-正常推送 1-停止推送

响应示例:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "setPushStatus",
  "data": {
    "code": 200,
    "message": "success",
    "data": null,
    "timestamp": 1732694400000
  }
}
```

4.5 获取推送状态

获取全局推送和定时推送的开关

方法名 (method): getScheduleStatus

MQTT 请求主题 /openapi/data/request/{appKey}

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "getScheduleStatus",
  "data": {}
}
```

请求参数 无

响应示例:

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "getScheduleStatus",
  "data": {
    "code": 200,
    "message": "success",
    "data": {
      "pushStatus": 0,
      "datePushStatus": 30,
      "onlineStatusPushStatus": 10
    },
    "timestamp": 1732694400000
  }
}
```

响应参数	数据类型	示例	描述
pushStatus	Integer	0	0-正常推送 1-停止推送。默认 0
datePushStatus	Integer	0	0-正常推送 1-停止推送。默认 0
onlineStatusPushStatus	Integer	0	0-正常推送 1-停止推送。默认 0

4.6 开启/停止定时推送（定时推送）

这个是定时推送的开启或者关闭的控制。但是优先级小于全局推送。因为全局推送包括数据变化推送和定时推送。

默认是开启定时推送。

如果全局推送开启了，然后停止了实时数据的定时推送，那么还是会进行数据变化推送。当开启实时数据的定时推送时（默认开启），实时数据会进行变化推送和定时推送。

设备状态、实时数据都是同理。

方法名 (method) : setSchedulePushStatus

MQTT 请求主题 /openapi/data/request/{appKey}

```

{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "setSchedulePushStatus",
  "data": {
    "pushStatus": 0
  }
}

```

响应示例:

```

{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "ABC123456",
  "method": "setSchedulePushStatus",
  "data": {
    "datePushStatus": 0,
    "onlineStatusPushStatus": 0
  }
}

```

请求参数	数据类型	示例	必填	描述
datePushStatus	Integer	0	是	0-正常推送 1-停止推送
onlineStatusPushStatus	Integer	0	是	0-正常推送 1-停止推送

5. 数据推送格式

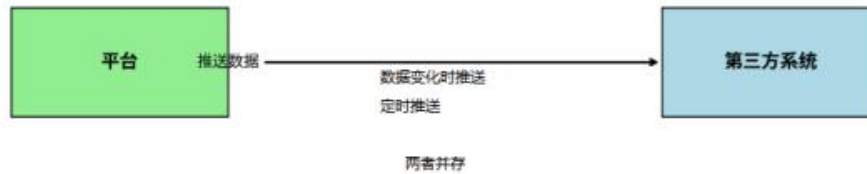
数据推送默认开始推送，包括数据变化推送和定时推送。系统会向主题/openapi/data/scheduled/{appKey}发布消息，第三方平台需要订阅这个主题。

实时数据、告警、在线状态都是向同一个主题推送。但是 pushType 不同，接收方注意区分 pushType 来区分实时数据、告警、在线状态。

dataType 是推送类型，change 是数据发送变化时的上报，scheduled 是定时上报。

MQTT
平台向第三方系统推送数据流程

主题: /openapi/data/scheduled/{appKey}



5.1 实时数据推送格式

订阅推送主题: /openapi/data/scheduled/{appKey}

Content-Type: application/json

pushType 为 realdata

数据样例:

```
{  
  "pushType": "realdata",  
  "dataType": "change",  
  "timestamp": 1732694400000,  
  "data": {  
    "deviceId": 123,  
    "deviceName": "设备 001",  
    "businessId": "IMEI123456",  
    "regionId": 1,  
    "productId": 1,  
    "productName": "温湿度",  
    "regionName": "一级区域",  
    "onlineStatus": 1,  
    "metrics": [  
      {  
        "meteId": "1555",  
        "meteName": "温度",  
        "value": "25.5",  
        "sValue": "25.5",  
        "unit": "°C",
```

```

        "time": "2024-11-2716:00:00"
    },
    {
        "meteId": "12125",
        "meteName": "湿度",
        "value": "60",
        "sValue": "60",
        "unit": "%",
        "time": "2024-11-2716:00:00"
    }
]
}
}
}

```

参数	数据类型	示例	描述
deviceId	Long	719943688798277	区域 ID
businessId	String	一级区域	行业 ID
deviceName	String	719943688798270	设备名称
productId	Long	719943688798271	产品 ID
productName	String	产品名称	产品名称
regionId	Long	719943688795245	区域 ID
regionName	String	区域名称	区域名称
onlineStatus	Integer	0	在线状态，0-在线，1-离线
metrics	List		
meteld	String	1855521	量 id
meteName	String	温度	量名称
value	String	26.5	实时值
unit	String	℃	单位
time	String	2025-01-01 14:00:00	时间

参数	数据类型	示例	描述
sValue	String	翻译之后的实时值	如果是枚举类型，就是翻译之后的实时值。 如果是字符串，和 value 等值

5.2 告警数据推送格式

订阅推送主题: /openapi/data/scheduled/{appKey}

Content-Type: application/json

pushType 为 alarm

数据样例:

```
{
  "pushType": "alarm",
  "dataType": "change",
  "timestamp": 1732694400000,
  "data": {
    "alarmId": 456,
    "deviceId": 123,
    "deviceName": "设备 001",
    "businessId": "IMEI123456",
    "regionId": 1,
    "regionName": "一级区域",
    "meteId": "80015551",
    "meteName": "压力过低",
    "level": 1,
    "value": "155",
    "sValue": "155",
    "alarmTime": "2025-11-27 16:00:00",
    "state": "END",
    "closeTime": "2025-11-27 18:00:00"
  }
}
```

响应参数	数据类型	示例	描述
alarmId	Long	15454521	告警恢复时，会以这个 Id 为主键

响应参数	数据类型	示例	描述
deviceId	Long	719943688798277	区域 ID
businessId	String	一级区域	行业 ID
deviceName	String	719943688798270	设备名称
regionId	Long	719943688795245	区域 ID
regionName	String	区域名称	区域名称
meteld	String	1545451	告警量 ID
meteName	String	温度过低	告警量名称
level	Integer	1	告警等级 1 2 3 4
value	String	22.5	实时值
sValue	String	22.5	如果是枚举类型，就是翻译之后的实时值。 如果是字符串，和 value 等值
alarmTime	String	2025-11-27 16:00:00	告警时间
state	String	BEGIN	BEGIN 或者 END BEGIN 是告警中 END 是已关闭
closeTime	String	2025-11-27 18:00:00	恢复时间，如果是 END 才会有 closeTime

5.3 在线状态推送格式

订阅推送主题: /openapi/data/scheduled/{appKey}

Content-Type: application/json

pushType 为 onlinestatus

数据样例:

```
{
  "pushType": "onlinestatus",
  "dataType": "change",
  "timestamp": 1732694400000,
```

```

"data": {
  "deviceId": 123,
  "deviceName": "设备 001",
  "businessId": "IMEI123456",
  "status": 0
}
}

```

响应参数	数据类型	示例	描述
deviceId	Long	719943688798277	区域 ID
businessId	String	一级区域	行业 ID
deviceName	String	719943688798270	设备名称
status	Integer	0	0-在线 1-离线

6. 错误码

6.1 标准错误码

错误码	说明
200	成功
400	请求参数错误
401	认证失败
403	无权限访问
404	资源不存在
500	服务器内部错误

6.2 认证错误

错误信息	说明
------	----

错误信息	说明
无效的 API Key	AppKey 不存在
API Key 已禁用	AppKey 已被禁用
API Key 已过期	AppKey 已过期

6.3 错误响应格式

```
{
  "messageId": "uuid-1234-5678-abcd",
  "appKey": "UCBYOL",
  "method": "queryRegions",
  "data": {
    "code": 500,
    "message": "AppKey 不匹配",
    "timestamp": 1766470441600
  }
}
```

7. 使用流程

7.1 初始化配置

步骤 1: 获取 AppKey (请联系管理员)

步骤 2: 获取 MQTT 地址、用户、密码。(请联系管理员)

步骤 3: 成功连接 MQTT 之后, 订阅消息响应主题和数据接收主题 (请联系管理员)

步骤 4: 开启实时推送 (默认开启, 无需配置)

步骤 5: 配置定时推送 (默认开启, 无需配置。默认的推送频率进行推送。实时数据为 30 分钟, 设备状态是 10 分钟)

7.2 数据查询 (发布主题)

通过发布主题发送请求

7.3 接收数据推送 (订阅主题)

订阅主题, 接收响应数据

8. 解释说明

8.1 订阅主题

数据推送主题 /openapi/data/scheduled/{appKey}

数据响应主题 /openapi/data/response/{appKey}

8.2 Qos

QoS 1: 保证消息至少送达一次，推荐使用

8.2 数据推送

数据变化都会向第三方主动推送

8.3 数据测试

推荐使用 MQTT 的官方工具 MQTTS 进行测试。方便进行主题、订阅、数据接收等。

8.4 名词说明

区域是树形节点，例如广东省-深圳市-南山区-机房。那么这就是一个 4 层树形节点的区域。

设备会属于一个区域节点，例如机房、基站等，那么设备属于基站下的设备。

区域可以看做是一个设备分组，可以先获取区域节点，再通过区域节点去获取区域下面的设备，也可以一次性获取全部设备

实时数据里面有属性量，例如温湿度设备，设备必定包含 2 个属性量，温度和湿度，那么温度和湿度就属于属性量

产品则是设备的属性量的汇总，例如温湿度设备，除了温度和湿度，还有信号、电量等，这些属性量可以通过获取产品去获取