OpenAPI 接口文档

1. 概述

物联网 OpenAPI 是一套标准化的 HTTP 接口规范，旨在为物联网平台与第三方应用系统之间建立高效、安全、可靠的数据交换通道。该接口体系通过 RESTful 架构设计，实现了设备数据的实时获取、状态监控、事件通知等核心功能，为设备管理、数据分析、业务集成提供基础支撑，旨在为企业快速构建智能化物联应用。
物联网 OpenAPI 采用"查询+推送"双模式架构，确保关键事件的实时触达。

1.1 基本信息

- Base URL: https://cloud.yuwuzl.com/openapi
- 协议: HTTP/HTTPS
- 数据格式: JSON
- 字符编码: UTF-8

2. 认证机制

2.1 认证方式

所有对外 API 请求必须在 URL 中携带以下认证参数：

参数名	类型	必填	说明
appKey	String	是	API 密钥
timestamp	Long	是	当前时间戳（毫秒）
signature	String	是	请求签名

2.2 签名算法

步骤 1: 构建待签名字符串

待签名字符串 = appKey + timestamp + appSecret

步骤 2: 计算 MD5 签名

signature = MD5(待签名字符串).toUpperCase()

示例（Java）：

```
String appKey = "your_app_key";
String appSecret = "your_app_secret";
Long timestamp = System.currentTimeMillis();
String signStr = appKey + ":" + timestamp + ":" + appSecret;
String signature = DigestUtils.md5Hex(signStr).toUpperCase();
```

2.3 时间戳验证

请求时间戳与服务器时间差不能超过 5 分钟

2.4 加密示例

appKey：40593672

appSecret：DET41A2VYN7FPI8BM6OU

timestamp：1764429611747

那么计算方式就是 MD5（40593672:DET41A2VYN7FPI8BM6OU:1764429611747）

最终得到 signature：bfd75be6cb742bc7f10ecbba06167ddf

再 toUpperCase（）转换为大写。

最终 signature=BFD75BE6CB742BC7F10ECBBA06167DDF

2.5 认证示例

URL 示例：

GET
/openapi/query/regions?appKey=&timestamp=1732694400000&signature=5D41402ABC4B2A76B9719D911017C592

完整示例（带业务参数）：

GET
/openapi/query/device/realtime/123?appKey=&timestamp=1732694400000&signature=5D41402ABC4B2A76B9719D911017C592&deviceId=123

注意事项:

所有请求方法（GET/POST/PUT/DELETE）都使用 URL 参数传递认证信息

认证参数（appKey、timestamp、signature）必须放在 URL 中

GET 请求的参数除了认证加密的参数，其他参数在 URL 进行拼接

POST/PUT/DELETE 方式，URL 认证参数不变，请求参数放在 Body 当中。

3. 数据接口

数据接口是第三方向平台调用，所以需要携带上面的认证方式参数，url 当中需要携带?appKey=xxxtamp=xxx&signature=xxx

数据接口可以使用 Api 在线文档进行查看

https://docs.apipost.net/docs/detail/56ce6ea83c51000?locale=zh-cn&target_id=2cca3014be6063

3.1 查询区域列表

返回所有区域

POST /openapi/query/regions

请求体参数:

无

响应示例:

```
{
  "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
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

POST /openapi/query/device/list

请求体参数:

```
{
 "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

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:


```
{
    "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

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
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

GET /openapi/query/device/{deviceId}

路径参数:

`deviceId`: 设备 ID

响应示例: 同"查询设备列表"中的单个设备对象

3.4 查询设备实时数据

POST /openapi/query/device/list/realtime

请求体参数:

```
{
 "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:


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

```
    "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
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		
meteId	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

GET /openapi/query/device/realtime/{deviceId}

路径参数:

deviceId: 设备 ID

响应示例:


```
{
  "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": "℃",
        "time": "2024-11-2716:00:00"
      },
      {
        "meteId": "12125",
        "meteName": "湿度",
        "value": "60",
        "unit": "%",
        "time": "2024-11-2716:00:00"
      }
    ]
  },
  "timestamp": 1732694400000
}
```


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

响应参数	数据类型	示例	描述
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		
meteId	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

POST /openapi/query/device/onlinestatus/list

请求体参数:

```
{
 "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:

```
{
  "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

POST /openapi/query/device/alarm/list

请求体参数:

```
{
 "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:

```
{
  "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
meteId	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，所以单个设备查询历史告警

POST /openapi/query/device/alarms/history

请求体参数:

```
{
 "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:


```
{
  "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
meteId	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

GET/openapi/query/product/list

请求参数: 无

响应示例:


```
{
  "code": 200,
  "message": "success",
  "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
        }
    }
  ],
  "timestamp": 1764749697528
}
```


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
meteId	Long	1545451	量 ID
meteName	String	温度过低	量名称
unit	String	℃	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 查询推送地址

返回推送地址。如果返回空，则为未配置推送

GET /openapi/push/config

请求体参数:

> 无

响应示例:

```
{
  "code": 200,
  "message": "success",
  "data": [
    {
      "httpUrl": 1
    }
  ],
  "timestamp": 1732694400000
}
```

响应参数	数据类型	示例	描述
httpUrl	String	http://182.55.11.18/push/data	第三方推送地址

4.2 配置推送地址

配置推送地址，如果多次配置，那么以最新的为主。
如果不想推送，可以调用下面的停止推送接口。

POST /openapi/push/config/url

请求体参数:

```
{
  "url": "https://y.com/config/push"
}
```

请求参数	数据类型	示例	必填	描述
url	String	https://y.com/config/push	是	第三方平台接收推送地址

响应示例:

```
{ "code": 200, "message": "success", "data": null, "timestamp": 1732694400000 }
```

4.3 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。

GET /openapi/query/schedule/config

请求体参数:

> 无

响应示例:

```
{
    "code": 200,
    "message": "success",
    "data": {
        "dataPushPeriod": 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

4.4 配置定时推送频率

POST /openapi/schedule/config/frequency

请求体参数:

```
{
  "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，例如只填 datePushPeriod，onlineStatusPushPeriod。
{"onlineStatusPushPeriod": 10} 或者 {"datePushPeriod": 10}

响应示例:

```
{ "code": 200, "message": "success", "data": null, "timestamp": 1732694400000 }
```

4.5 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。

POST /openapi/schedule/push/status

请求体参数:

```
{
  "pushStatus": 0
}
```

请求参数	数据类型	示例	必填	描述

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:


```
{ "code": 200, "message": "success", "data": null, "timestamp": 1732694400000 }
```


4.6 获取推送状态

获取全局推送和定时推送的开关

GET /openapi/query/schedule/status

参数 无

响应示例:


```
{
  "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

4.7 开启/停止定时推送（仅定时）

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。

POST /openapi/schedule/fixed/push/status

```
{
  "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

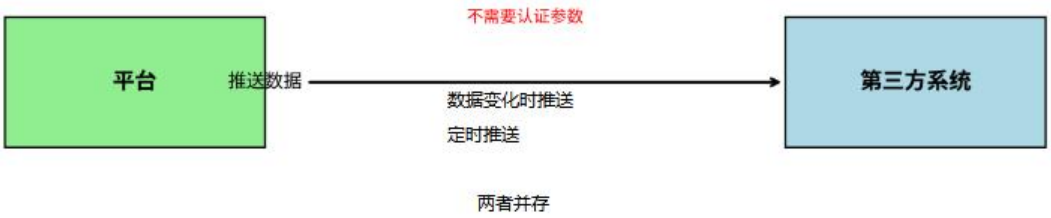
响应示例:

```
{ "code": 200, "message": "success", "data": null, "timestamp": 1732694400000 }
```

5. 数据推送格式

当启用数据推送和配置推送地址之后，系统会向配置的推送地址发送 POST 请求。
实时数据、告警、在线状态都是向同一个地址推送。
但是 pushType 不同，接收方注意区分 pushType。

平台向第三方系统推送数据流程



5.1 实时数据推送格式

请求方法: POST
Content-Type: application/json

pushType 为 realdata

请求体:

```
{
  "pushType": "realdata",
  "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": "℃",
        "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		
meteId	String	1855521	量 id
meteName	String	温度	量名称
value	String	26.5	实时值
unit	String	℃	单位

参数	数据类型	示例	描述
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

5.2 告警数据推送格式

请求方法: POST

Content-Type: application/json

pushType 为 alarm

请求体:

```
{
  "pushType": "alarm",
  "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": "2025-11-2716:00:00",
    "state": "END",
    "closeTime": "2025-11-2718:00:00"
  }
}
```

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
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
meteId	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

请求方法: POST

Content-Type: application/json

pushType 为 onlinestatus

请求体:

```
{
  "pushType": "onlinestatus",
  "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
429	请求过于频繁（超过 100 次/分钟）

错误码	说明
500	服务器内部错误

6.2 认证错误

错误信息	说明
缺少 appKey 参数	URL 中未传递 appKey 参数
缺少 timestamp 参数	URL 中未传递 timestamp 参数
缺少 signature 参数	URL 中未传递 signature 参数
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
签名验证失败	signature 签名不正确
请求已过期	时间戳超过 5 分钟
请求频率超限	超过速率限制

6.3 错误响应格式

```
{ "code": 401, "message": "Invalid signature", "data": null, "timestamp": 1732694400000 }
```

7. 使用流程

7.1 初始化配置

步骤 1: 获取 AppKey 和 AppSecret（请联系管理员）

步骤 2: 配置推送地址

步骤 3: 开启实时推送

步骤 4: 配置定时推送（可选，若不配置安装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

查询设备列表

查询实时数据

....

根据业务调用需要的接口

8. 解释说明

8.1 认证签名

妥善保管 appSecret
每次请求使用新的 timestamp
签名算法简单：MD5(appKey + timestamp + appSecret)
注意 timestamp 有效期为 *5* 分钟

8.2 数据获取

数据变化都会向第三方主动推送（配置推送地址的前提下）
向平台查询数据时注意限流（目前限制频率是 100 次/分钟）

8.3 测试说明

本文档都是 HTTP 接口，推送使用 PostMan 工具进行接口测试，方便测试接口连通性和接口的请求响应数据。

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