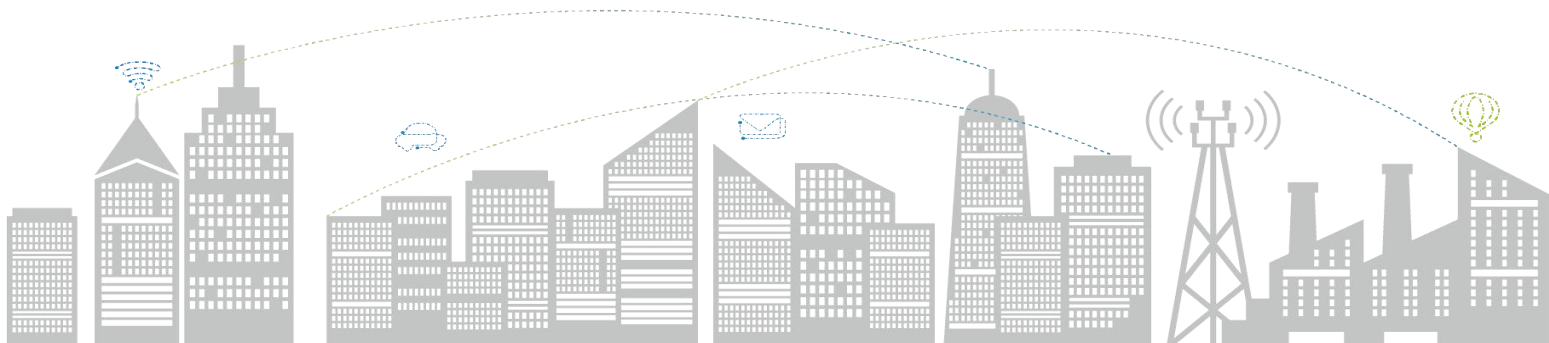


InfiCombo 能力开放 API 接口规范 V2.0



InfiCombo 能力开放 API 接口规范 V2.0

版本	日期	作者	审核者	备注
V2.0	2018-6-4	中兴克拉		初稿
V2.0.1	2018-10-4	中兴克拉		增加最新业务数据查询接口
V2.0.2	2018-11-16	中兴克拉		增加设备运行状态查询接口
V2.0.3	2018-11-21	中兴克拉		增加项目设备类型查询接口
V2.0.4	2018-12-12	中兴克拉		增加告警查询接口
V2.0.5	2019-2-20	中兴克拉		增加接口 URL 访问样例说明
V2.0.6	2019-5-10	中兴克拉		增加告警操作接口

目录

- 1 总则.....4
 - 1.1 字符集编码约定.....4
 - 1.2 设备标识 xEUI 约定.....4
 - 1.3 接口形式.....4
 - 1.4 接口路径组成.....4
 - 1.5 接口分类.....5
- 2 安全管理接口.....5
 - 2.1 认证管理.....5
 - 2.1.1 用户认证接口.....5
- 3 业务管理接口.....6
 - 3.1 设备类型查询.....6
 - 3.1.1 设备类型查询.....6
 - 3.1.2 实配类型查询.....7
 - 3.1.3 设备列表查询.....8
 - 3.2 业务模型查询.....10
 - 3.2.1 数据模型查询.....10
 - 3.2.2 指令模型查询.....11
 - 3.2.3 告警模型查询.....13
 - 3.3 业务数据查询.....15
 - 3.3.1 业务数据查询.....15
 - 3.3.2 业务数据批量查询.....17
 - 3.3.3 最新业务数据查询.....19
 - 3.3.4 设备运行状态查询.....21
 - 3.3.5 设备当前告警查询.....22
 - 3.3.6 最新告警列表查询.....24
 - 3.3.7 设备历史告警查询.....24
 - 3.3.8 历史告警批量查询.....25

3.4 控制指令下发.....	26
3.4.1 设备控制指令下发.....	26
3.4.2 设备组播指令下发.....	28
3.5 业务数据上报.....	29
3.5.1 上报接口注册.....	29
3.5.2 上报接口注销.....	31
3.5.3 上报接口注册查询.....	32
3.6 告警数据上报.....	33
3.6.1 上报接口注册.....	33
3.6.2 上报接口注销.....	36
3.6.3 上报接口注册查询.....	36
3.7 设备状态上报.....	37
3.7.1 上报接口注册.....	37
3.7.2 上报接口注销.....	39
3.7.3 上报接口注册查询.....	39
3.8 告警维护接口.....	40
3.8.1 告警确认.....	40
3.8.2 告警反确认.....	41
3.8.3 告警手工清除.....	42

1 总则

1.1 字符集编码约定

如未特别说明，统一按 **UTF-8** 编码处理；

1.2 设备标识 xEUI 约定

包括 AppEUI、GwEUI、MoteEUI、JoinEUI，统一使用 16 进制数字字符串，小写字母，长度固定为 16 个字符。

举例：004a770065010001.

对于 GSP 平台产品，传感设备的编号是在 GSP 的 DEVEUI 增加 n@前缀，举例：

举例：1@004a770065010001

1.3 接口形式

1、Restful 接口（http+json）。

2、**POST** 请求的 contenttype 约定：

- ❖ 默认是：application/x-www-form-urlencoded;charset=UTF-8，请求参数默认放 header 中。URL 样例：
http://xxxxip:port/openapi/v2/sm/login?username=xxxx&pswd=xxxxx
- ❖ 特别地：个别接口需要 application/json;charset=UTF-8，相关接口方法会有专门说明，此时请求参数放到 body 中。

1.4 接口路径组成

完整的 OpenAPI 路径包括以下几个部分：

1. HTTP 协议(支持 http 和 https)；
2. IP 地址、端口：需要根据中兴克拉提供的 inficombo 系统 ip 和端口信息来设置；
3. 接口名称，固定为 openapi；

4. 接口版本号(v1, v2, v3);

举例: <https://47.96.153.2:8093/openapi/v2/devtype/devtypes>。

1.5 接口分类

1. 用户认证接口;
2. 设备类型查询
3. 业务模型查询;
4. 业务数据查询;
5. 控制指令下发;
6. 业务数据上报;
7. 告警数据上报;
8. 告警操作接口;
9. 设备状态上报;

2 安全管理接口

2.1 认证管理

2.1.1 用户认证接口

【接口功能】: 用户登录认证, 获取访问令牌 token。

【URL 举例】: <http://xxxxxx:8093/openapi/v2/sm/login?username=xxx&pswd=xxxx>

【接口定义】:

接口参数项	参数取值	备注
URL	/openapi/v2/sm/login	用户登录认证, 获取访问令牌 token。
Para	{ "username": "admin", "pswd": "admin" }	输入真实的登陆用户名和密码

Method	POST	
响应	{ "result":0, "remark":"remark info", "data": json }	result: 错误码 0 表示成功; 1 表示失败; remark: 错误的详细信息 data: 返回数据 执行成功时返回数据 执行失败时返回 null

用户登陆成功响应 data 对应的内容（json 格式）：（样例）

```
{  
  "token": "B7700650300b9212A7700650300b927878980009675634265781221231321"  
}
```

Data 的属性定义参见下表：

字 段 名	类 型	长 度	约 束	解 释
token	varchar	500	必选	访问令牌，有效期 12 个小时。 令牌失效后需要重新申请令牌，才能进行正常访问。

3 业务管理接口

3.1 设备类型查询

3.1.1 设备类型查询

【接口功能】：查询应用平台当前支持的全部设备类型列表。

【URL 举例】：http://xxxxxx:8093/openapi/v2/dev/devtypes?token=xxxxxxx

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/dev/devtypes	查询当前平台支持的所有设备类型列表。
Para	{ "token": "111111111111" }	1) token 是用户登陆成功时分配的，必选。 2) 这里 token 取值是举例，下同。
Method	GET	

响应	<pre>{ "result":0, "remark":"remark info", "data": json }</pre>	result: 错误码 0-表示成功; 1-表示失败; 2- token 过期或无效; remark: 错误的详细信息 data: 返回数据 执行成功时返回数据 执行失败时返回 null
----	---	---

设备类型查询成功时，响应中 **data** 字段对应的内容（json 格式）：（样例）

```
[
  {
    "devtype": "claa_th20",
    "devtypedesc": "克拉温湿度监测设备",
    "vendor": "XXXX",
    "appclass": "智慧社区"
  },
  ...
]
```

Data 中包含的各字段定义表：

字段名	类型	长度	约束	解释
devtype	varchar	200	必选	设备类型标识
devtypedesc	varchar	200	必选	设备类型名称
vendor	varchar	100	必选	设备类型归属厂家
appclass	varchar	100	必选	设备类型对应的应用归类

3.1.2 实配类型查询

【接口功能】：查询项目当前实际配置的设备类型列表。

【URL 举例】：<http://xxxxxx:8093/openapi/v2/dev/projectdevtypes?token=xxxxxx>

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/dev/projectdevtypes	查询项目配置的设备类型列表。
Para	<pre>{ "token": "111111111111" }</pre>	1) token 是用户登陆成功时分配的，必选。 2) 这里 token 取值是举例，下

		同。
Method	GET	
响应	{ "result":0, "remark":"remark info", "data": json }	result: 错误码 0-表示成功; 1-表示失败; 2- token 过期或无效; remark: 错误的详细信息 data: 返回数据 执行成功时返回数据 执行失败时返回 null

设备类型查询成功时，响应中 data 字段对应的内容（json 格式）：（样例）

```
[  
  {  
    "devtype": "claa_th20",  
    "devtypedesc": "克拉温湿度监测设备",  
    "vendor": "XXXX",  
    "appclass": "智慧社区"  
  },  
  ...  
]
```

Data 中包含的各字段定义表：

字 段 名	类 型	长 度	约 束	解 释
devtype	varchar	200	必选	设备类型标识
devtypedesc	varchar	200	必选	设备类型名称
vendor	varchar	100	必选	设备类型归属厂家
appclass	varchar	100	必选	设备类型对应的应用归类

注：该接口推荐在项目的各类设备实际配置好之后查询。

3.1.3 设备列表查询

【接口功能】：查询用户有权访问的设备列表。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/dev/devs?token=xxxxx&devtype=xxxxx&pageindex=1&batch=100

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/dev/devs	查询用户有权访问的设备列表
Para	{	1) token 是用户登陆成功时

	<pre> "token": "111111111111", "devtype": "claa_level", "pageindex": 1, "batch": 100 } </pre>	<p>分配的，必选。</p> <p>2) pageindex 和 batch 用作分页查询，默认值分别为 1 和 100，表示查询第 1 页最多 100 条记录，必选。Batch 最大不能超过 200。</p> <p>3) devtype: 设备类型 ID，可选。</p>
Method	GET	
响应结果	<pre> { "result": 0, "remark": "remark info", "data": json } </pre>	<p>result: 错误码 0 表示成功; 1 表示失败; 2 token 过期或无效; remark: 错误的详细信息 data: 返回数据: 执行成功时返回数据; 执行失败时返回 null;</p>

查询成功时，响应中 data 对应的内容（json 格式）：（样例）

```

{
  "page": {"pageindex": 1, "batch": 100, "hasmorepage": "false"},
  "devs": [
    {
      "deveui": "004a070001111156",
      "devtype": "claa_level",
      "memo": "xxxxxxxx",
      "project": "xxxxxx-projectid"
    },
    ...
  ]
}

```

[注] 以上 hasmorepage=true|false，true 表示后续还有数据，false 表示没有数据。

查询结果各属性字段定义如下表：

字段名	类型	长度	约束	解释
deveui	varchar	50	必选	终端电子识别号
devtype	varchar	200	必选	终端设备类型
memo	varchar	500	必选	终端备注
project	varchar	200	必选	终端归属项目

3.2 业务模型查询

3.2.1 数据模型查询

【接口功能】：查询指定设备类型的业务测量数据模型定义。

【URL 举例】：

<http://xxxxxx:8093/openapi/v2/model/datamodel?token=xxxxxx&devtype=xxxxxx>

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/model/datamodel	查询指定设备类型的业务数据模型定义。
Para	{ "token": "111111111111", "devtype": "haitong_dc" }	1) token 是用户登陆成功时分配的，必选。 2) devtype 是设备类型。
Method	GET	
响应结果	{ "result": 0, "remark": "remark info", "data": json }	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期; remark: 错误的详细信息; data: 返回数据; 执行成功时返回数据; 执行失败时返回 null;

查询成功时，响应中 **data** 对应的内容（json 格式）：（样例）

```
{  
  "devtype": "haitong",  
  "sfields": [  
    {  
      "sfield": "barometer",  
      "sfielddescp": "field zh name xx",  
      "datatype": "float",  
      "sunit": "ppa",  
      "remark": "field description"  
    },  
    {  
      "sfield": "humidity",  
      "sfielddescp": "field zh name xx",  
      "datatype": "float",  
      "sunit": "ppa",  
      "remark": "field description"  
    }  
  ]  
}
```

```

        "datatype": "float",
        "sunit": "%",
        "remark": "field description"
    },
    {
        "sfield": "temp",
        "sfielddescp": "field zh name xx",
        "datatype": "int",
        "sunit": "℃",
        "remark": "field description"
    }
]
}

```

以上 data 中各属性定义如下：

字 段 名	类 型	长 度	约 束	解 释
devtype	varchar	200	必选	设备类型
sfields	Json 数组	N	必选	一个设备类型包括若干个数据模型字段定义
sfield	varchar	100	必选	数据模型字段名
sfielddescp	varchar	100	可选	数据模型字段描述
datatype	varchar	20	必选	字段对应的数据类型： Byte, short, int, long, float, double, string.
sunit	varchar	20	可选	指标值单位
remark	varchar	200	可选	字段含义描述。

3.2.2 指令模型查询

【接口功能】：指定设备类型的控制指令模型查询（即控制指令的模板）。

【URL 举例】：

<http://xxxxxx:8093/openapi/v2/model/controlmodel?token=xxxx&devtype=xxxx>

【接口定义】：

接口参数项	参数取值	备注
-------	------	----

URL	/openapi/v2/model/controlmodel	指定设备类型的控制指令模型查询（即控制指令的模板）。
Para	{ "token": "111111111111", "devtype": "haitong_dc" }	1) token 是用户登陆成功时分配的，必选。 2) devtype 是设备类型。
Method	GET	
响应结果	{ "result":0, "remark": "remark info", "data": json }	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期或无效; remark: 错误的详细信息 data: 返回数据: 执行成功时返回数据; 执行失败时返回 null;

查询成功时，响应中 data 对应的内容（json 格式）：（样例）

```
{
  "devtype": "haitong_ac",
  "commands": [
    {
      "cmd": "AirControlerCmd",
      "cmddescp": "cmd function description",
      "confirm": 0,
      "cmdtype": 0,
      "paras": [
        {
          "sno": 0,
          "paraname": "p1",
          "datatype": "int",
          "remark": "开关控制参数，1-开，0-关"
        },
        {
          "sno": 1,
          "paraname": "p2",
          "datatype": "int",
          "remark": "风速控制参数，1~10 级"
        }
      ]
    },
    ... ..
  ]
}
```

```
}
```

以上 data 中各属性定义如下:

字段名	类型	长度	约束	解释
devtype	varchar	200	必选	设备类型
commands	Json 数组	N	必选	控制指令定义数组
cmd	varchar	200	必选	控制指令对应的控制字, 如空调控制字
cmddescp	Varchar	200	可选	控制指令功能描述
confirm	byte	4	必选	命令是否需要确认: 0--否; 1--是;
cmdtype	byte	4	必选	命令类型: 0--普通命令; 1--组播命令;
paras	Json 数组	N	必选	控制指令对应的控制参数, 用于控制指令下发时设置控制参数值。
sno	Int	2	必选	控制参数序号, 从 0 开始。
paraname	varchar	50	必选	控制参数名, 如 P1, P2, ...
datatype	varchar	20	必选	控制参数值类型: byte, short, int, long, float, double, string.
remark	varchar	200	可选	对指令控制字及控制参数的描述, 也可参见具体设备厂家的产品说明。具体控制指令下发时, 不需要携带 remark 字段。

3.2.3 告警模型查询

【接口功能】: 查询指定设备类型的告警码模型定义。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/model/alarmmodel?token=xxxx&devtype=xxxx

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/model/alarmmodel	查询指定设备类型的告警码模型定义。
Para	{ "token": "111111111111", "devtype": "xxxxx" }	1) token 是用户登陆成功时分配的，必选。 2) devtype 是设备类型。
Method	GET	
响应结果	{ "result": 0, "remark": "remark info", "data": json }	result: 错误码 0--表示成功; 1--表示失败; 2 --token 过期; remark: 错误的详细信息; data: 返回数据; 执行成功时返回数据; 执行失败时返回 null;

查询成功时，响应中 data 对应的内容（json 格式）：（样例）

```
{
  "devtype": "xxxxxxxx",
  "alarmcodes": [
    {
      "alarmcode": "xxxxxx",
      "devtype": "xxxxxxxx",
      "codetype": 1,
      "codegroup": 1,
      "remark": "alarm code description"
    },
    {
      "alarmcode": "xxxxxx",
      "devtype": "xxxxxxxx",
      "codetype": 1,
      "codegroup": 1,
      "remark": "alarm code description"
    }
  ]
}
```

以上 data 中各属性定义如下：

字段名	类型	长度	约束	解释
devtype	varchar	200	必选	设备类型
alarmcodes	Json 数组	N	必选	一个设备类型包括若干个告警码定义
alarmcode	varchar	100	必选	告警码
devtype	varchar	100	必选	归属设备类型
codetype	int	10	必选	0—通知码； 1—设备告警码； 2—门限告警码； 3—其他告警码；
codegroup	int	10	必选	告警码分组，相同值表示相同分组，比如该值为 1 的各告警码属于设备告警码分组。
remark	varchar	200	必选	告警码含义描述。

3.3 业务数据查询

3.3.1 业务数据查询

【接口功能】：单个设备的业务数据查询。

【URL 举例】：

<http://xxxxxx:8093/openapi/v2/data/devdata?token=xxxx&deveui=xxxxx&begintime=2018-02-01+00:00:00&endtime=2018-02-02+23:59:59>

【接口定义】：

接口参数项	参数取值	备注
URL	openapi/v2/data/devdata	单个设备的业务数据查询
Para	{ "token": "111111111111" , "deveui": "004a070001111156", "begintime": "2018-02-01 00:00:00",	1) token 是用户登陆成功时分配的，必选。 2) deveui 是设备标识，必选； 3) begintime: 查询开始时间

	<code>"endtime": "2018-02-02 23:59:59"</code> <code>}</code>	戳，yyyy-MM-dd HH:mm:ss 格式，时区为国内时区（GTC+08:00）， 必选 ； 4) endtime : 查询结束时间戳，yyyy-MM-dd HH:mm:ss 格式， 必选 ； 注：起止时间最大间隔为 3 天，返回数据记录条数最多 5000。
Method	GET	
响应结果	<code>{</code> <code> "result":0,</code> <code> "remark":"remark info",</code> <code> "data": json</code> <code>}</code>	result : 错误码 0 表示成功； 1 表示失败； 2 token 过期 remark : 错误的详细信息 data : 返回数据，按对应数据业务模型组织的 json 对象，执行成功时返回数据，执行失败时返回 null。

查询成功时，响应中 **data** 对应的内容（json 格式）：（样例）

```
{
  "devtype": "haitong",
  "datarows":
  [
    {
      "collecttime": "2018-02-01 02:00:00",
      "deveui": "004a070001111156",
      "devname": "xxxxxxxxxx",
      "tempature": 23.9,
      "humdity": 3.8,
      "pressure": 2
    },
    ...
  ]
}
```

响应结果 data 中各属性定义：

字段名	类型	长度	约束	解释
devtype	varchar	200	必选	业务数据对应的设备类型 ID，方便解析。
datarows	数组 []	N	必选	业务数据记录数组名。

				数组中每条记录对应一个 json 对象，json 对象属性和数据模型中的 subtype 模型的定义一致。
collecttime	采集时间	100	必选	yyyy-MM-dd HH:mm:ss 格式
deveui	设备标识	50	必选	
devname	设备名称	100	可选	
xxxx	数据模型字段		可选	和模型定义保持一致

3.3.2 业务数据批量查询

【接口功能】：批量设备的业务数据查询。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/data/batchdevdata?token=xxxx&deveuis=xxxxx&begintime=2018-02-01+00:00:00&endtime=2018-02-02+23:59:59&pageindex=1

【接口定义】：

接口参数项	参数取值	备注
URL	openapi/v2/data/batchdevdata	批量设备的业务数据查询。
Para	{ "token": "111111111111" , "deveuis": "deveui1_deveui2_xxx", "begintime": "2018-02-01 00:00:00", "endtime": "2018-02-01 23:59:59", "pageindex": 1 }	1) token 是用户登陆成功时分配的，必选。 2) deveuis 是设备标识，必选，多个 deveui 使用下划线“_”连接，最多 100 个； 注：要求 deveuis 中设置的所有设备对应的设备类型都相同，另外归属项目也相同（即设备列表查询结果中 project 相同）。 3) begintime: 查询开始时间戳，yyyy-MM-dd HH:mm:ss 格式，时区为国内时区（GTC+08: 00），必选； 4) endtime: 查询结束时间戳，yyyy-MM-dd HH:mm:ss 格式，必选；

		5) pageindex: 分页标记, 初次查询时默认填写 1, 如果还有后续页, 第二次查询时, 其值+1, 必选。分页返回时每页最大 100 条数据。 注: 起止时间段不能超过 24 小时。
Method	GET	
响应结果	{ "result":0, "remark":"remark info", "data": json }	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期; remark: 错误的详细信息 data: 返回数据, 按对应数据业务模型组织的 json 对象, 执行成功时返回数据, 执行失败时返回 null。

查询成功时, 响应中 data 对应的内容 (json 格式): (样例)

```
{
  "devtype": "haitong",
  "hasmorepage": true|false,   [注]: true--表示还有后续页; false--无后续页。
  "datarows":
  [
    {
      "collecttime": "2018-02-01 02:00:00",
      "deveui": "004a070001111156",
      "devname": "xxxxxxxxxx",
      "tempature": 23.9,
      "humdity": 3.8,
      "pressure": 2
    },
    ...
  ]
}
```

响应结果 data 中各属性定义:

字 段 名	类 型	长 度	约 束	解 释
devtype	varchar	200	必选	业务数据对应的设备类型 ID, 方便解析。
datarows	数组 []	N	必选	业务数据记录数组名。

				数组中每条记录对应一个 json 对象，json 对象属性和数据模型中的定义一致。
collecttime	采集时间	100	必选	yyyy-MM-dd HH:mm:ss 格式
deveui	设备标识	50	必选	
devname	设备名称	100	可选	
xxxx	数据模型字段		可选	和模型定义保持一致

3.3.3 最新业务数据查询

【接口功能】：查询指定设备的最新业务数据。

【URL 举例】：

<http://xxxxxx:8093/openapi/v2/data/latestdevdata?token=xxxx&deveuis=xxxxx>

【接口定义】：

接口参数项	参数取值	备注
URL	openapi/v2/data/latestdevdata	单个设备的最新 1 条业务数据查询。
Para	{ "token": "111111111111" , "deveuis": "deveui1_deveui2_xxx" }	1) token 是用户登陆成功时分配的，必选。 2) deveuis 是设备标识，必选，多个 deveui 时使用英文下划线“_”分隔，最多不超过 100 个； 注：要求 deveuis 中设置的所有设备对应的设备类型都相同，另外归属项目也相同（即设备列表查询结果中 project 相同）。
Method	GET	
响应结果	{ "result":0, "remark": "remark info", "data": json }	result: 错误码 0 表示成功； 1 表示失败； 2 token 过期 remark: 错误的详细信息 data: 返回数据，按对应数据

		业务模型组织的 json 对象，执行成功时返回数据，执行失败时返回 null。
--	--	---

查询成功时，响应中 data 对应的内容（json 格式）：（样例）

```
{
  "devtype": "claa_door",
  "datarows":
  [
    {
      "collecttime": "2018-02-01 02:00:00",
      "deveui": "004a070001111156",
      "devname": "xxxxxxxxxx",
      "xxxxxx": 23.9,
      "yyyyyy": 3.8,
      "zzzzzz": 2,
      .....
    },
    ...
  ]
}
```

响应结果 data 中各属性定义：

字 段 名	类 型	长 度	约 束	解 释
devtype	varchar	200	必选	业务数据对应的设备类型 ID，方便解析。
datarows	数组 []	N	必选	业务数据记录数组名。 数组中每条记录对应一个 json 对象，json 对象属性和数据模型中的 subtype 模型的定义一致。
collecttime	采集时间	100	必选	yyyy-MM-dd HH:mm:ss 格式
deveui	设备标识	50	必选	
devname	设备名称	100	可选	
xxxx	数据模型字段			和模型定义保持一致

3.3.4 设备运行状态查询

【接口功能】：查询指定设备的最新状态（在线/离线）数据。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/data/devstatedata?token=xxxx&deveuis=xxxxxx

【接口定义】：

接口参数项	参数取值	备注
URL	openapi/v2/data/devstatedata	查询指定设备的最新状态（在线/离线）数据。
Para	{ "token": "111111111111" , "deveuis": "deveui1_deveui2_xxx" }	1) token 是用户登陆成功时分配的，必选。 2) deveuis 是设备标识，必选，多个 deveui 时使用英文下划线 "_" 分隔，最多不超过 100 个。
Method	GET	
响应结果	{ "result":0, "remark":"remark info", "data": json }	result: 错误码 0 表示成功； 1 表示失败； 2 token 过期 remark: 错误的详细信息 data: 返回数据，对应设备的状态 json 数组对象，执行成功时返回数据，执行失败时返回 null。

查询成功时，响应中 data 对应的内容（json 数组）：（样例）

```
[
  {
    "deveui": "4a77021106000d",
    "state": 0
  },
  ...
]
```

响应结果 data 中数组对象的属性定义如下：

字段名	类型	长度	约束	解释
deveui	varchar	50	必选	设备标识 devEUI
state	Int		必选	0--在线；

				1--离线； 2--异常； 3--未知；
--	--	--	--	----------------------------

注：对于查询条件中无效的 deveui，返回结果中将不会包含对应的数据记录。

3.3.5 设备当前告警查询

【接口功能】：查询指定设备的最新告警数据。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/alarm/devcuralarm?token=xxxx&deveuis=xxxxxx

【接口定义】：

接口参数项	参数取值	备注
URL	openapi/v2/alarm/devcuralarm	查询指定设备的最新告警。
Para	{ "token": "111111111111" , "deveuis": "deveui1_deveui2_xxx" }	1) token 是用户登陆成功时分配的，必选。 2) deveuis 是设备标识，必选，多个 deveui 时使用英文下划线"_"分隔，最多不超过 100 个。
Method	GET	
响应结果	{ "result":0, "remark": "remark info", "data": json }	result: 错误码 0 表示成功； 1 表示失败； 2 token 过期 remark: 错误的详细信息 data: 返回数据，对应设备的状态 json 数组对象，执行成功时返回数据，执行失败时返回 null。

查询成功时，响应中 data 对应的内容（json 数组）：（样例）

```
[  
  {  
    "alarmid": "xxxxxx",           // 告警流水号  
    "devtype": "zte_clss_loc",     // 告警设备类型  
    "deveui": "0047c60739448800", // 告警设备标识  
    "alarmcode": "1-1",           // 告警码
```



```

    "typeflag":1,           // 类型标记: 0: 告警恢复; 1: 告警发生; 2: 通知消息;
    "title":"xxxx",        // 告警标题
    "alarmlevel":1,        // 告警级别
    "alarmtime":"yyyy-MM-dd HH:mm:ss", // 告警发生时间
    "descp":"xxxxxxxxxxxx", // 告警详细描述
    "cleartime":null,       // 告警恢复时间
    "confirmstate":0,       // 告警确认状态: 0-未确认, 1-已确认;
    "confirmtime":null,     // 告警确认时间, 若有值, 为 yyyy-MM-dd HH:mm:ss 格式;
    "confirmer":"xxxx",     // 告警确认人
    "remark":"xxxxxxxx"    // 告警备注信息.
  },
  ...
]

```

响应结果 data 中数组 json 对象的属性定义如下:

字段名	类型	长度	约束	解释
alarmid	Varcahr	200	必选	告警流水号
devtype	Varcahr	100	必选	设备类型
deveui	Varcahr	200	必选	设备标识
alarmcode	Varcahr	100	必选	告警码
typeflag	Byte	4	必选	类型标记: 0: 告警恢复; 1: 告警发生; 2: 通知消息;
title	Varcahr	200	必选	告警标题
alarmlevel	Int	10	必选	告警级别, 固定为 1
alarmtime	varchar	50	必选	告警上报时间戳, yyyy-MM-dd HH:mm:ss 格式字符串。
descp	varchar	512	可选	告警详细描述
cleartime	varchar	50	可选	告警恢复时间, yyyy-MM-dd HH:mm:ss 格式字符串, 可为 null。
confirmstate	byte	4	可选	告警确认状态: 0-未确认, 1-已确认;
confirmtime	varchar	50	可选	告警确认时间, yyyy-MM-dd

				HH:mm:ss 格式字符串，可为 null。
confirmer	varchar	20	可选	告警确认人姓名，可为 null
remark	Varchar	512	可选	告警备注信息

3.3.6 最新告警列表查询

【接口功能】：查询最新的前 N 条告警数据。

【URL 举例】：

<http://xxxxxx:8093/openapi/v2/alarm/getlatestalarms?token=xxxx&maxnum=100>

【接口定义】：

接口参数项	参数取值	备注
URL	openapi/v2/alarm/getlatestalarms	查询最新的前 N 条告警数据。 最新告警是按告警发生时间降序排序，取前 N 条。
Para	{ "token": "111111111111" , "maxnum": 100 }	1) token 是用户登陆成功时分配的，必选。 2) maxnum 是期望查询的最大告警数目，上限值 1000。
Method	GET	
响应结果	{ "result": 0, "remark": "remark info", "data": json }	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期 remark: 错误的详细信息 data: 返回数据，对应设备的状态 json 数组对象，执行成功时返回数据，执行失败时返回 null。

注：查询成功时，响应中 data 对应的内容是告警 json 数组，具体参见 3.3.5 章节定义。

3.3.7 设备历史告警查询

【接口功能】：查询指定设备的历史告警数据。

【URL 举例】：

<http://xxxxxx:8093/openapi/v2/alarm/devhisalarm?token=xxxx&deveui=xxxxx&begintime=2019-02>

-01+00:00:00&endtime=2019-02-02+23:59:59&pageindex=1

【接口定义】：

接口参数项	参数取值	备注
URL	openapi/v2/alarm/devhisalarm	查询指定设备的历史告警数据，一次只能查询一个设备的历史告警。
Para	{ "token": "111111111111", "deveui": "xxxxxxxx", "begintime": "2019-01-01 00:00:00", "endtime": "2019-01-01 23:59:59", "pageindex": 1 }	1) token 是用户登陆成功时分配的，必选。 2) deveui 是设备标识，必选。 3) begintime: 查询开始时间段，必选，字符串，格式：YYYY-MM-DD HH:mm:ss; 4) endtime: 查询开始时间段，字符串，必选，格式：YYYY-MM-DD HH:mm:ss; 5) pageindex: 分页号，正整数，必选。从 1 开始，每页最大 100 条记录，查询结果小于 100 条表示分页查询完毕。 注：查询开始时间小于结束时间，且最大间隔为 30 天，以告警发生时间为参照。
Method	GET	
响应结果	{ "result":0, "remark":"remark info", "data": json }	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期 remark: 错误的详细信息 data: 返回数据，对应设备的状态 json 数组对象，执行成功时返回数据，执行失败时返回 null。

注：查询成功时，响应中 data 对应的内容是告警 json 数组，具体参见 3.3.5 章节定义。

3.3.8 历史告警批量查询

【接口功能】：批量查询历史告警数据。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/alarm/hisalarmlist?token=xxxx&begintime=2019-02-01+00:00:00&

endtime=2019-02-02+23:59:59&pageindex=1

【接口定义】：

接口参数项	参数取值	备注
URL	openapi/v2/alarm/hisalarmlist	批量查询历史告警数据。
Para	{ "token": "111111111111", "begintime": "2019-01-01 00:00:00", "endtime": "2019-01-01 23:59:59", "pageindex": 1 }	1) token 是用户登陆成功时分配的，必选。 2) begintime: 查询开始时间段，必选，字符串，格式：YYYY-MM-DD HH:mm:ss; 3) endtime: 查询开始时间段，字符串，必选，格式：YYYY-MM-DD HH:mm:ss; 4) pageindex: 分页号，正整数，必选。从 1 开始，每页最大 100 条记录，查询结果小于 100 条表示分页查询完毕。 注：查询开始时间小于结束时间，且最大间隔为 3 天，以告警发生时间为参照。
Method	GET	
响应结果	{ "result":0, "remark":"remark info", "data": json }	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期 remark: 错误的详细信息 data: 返回数据，对应设备的状态 json 数组对象，执行成功时返回数据，执行失败时返回 null。

[注：](#)查询成功时，响应中 data 对应的内容是告警 json 数组，具体参见 3.3.5 章节定义。

3.4 控制指令下发

3.4.1 设备控制指令下发

【接口功能】：设备控制指令下发。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/devcontrol/cmd

特别说明：命令参数以 json 对象格式设置到请求的 body 中，具体参见如下接口定义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/devcontrol/cmd	设备控制指令下发
Para	{ "token": "111111111111", "deveui": "004a070001111156", "devtype": "haitong_ac", "confirm": false, "devport": 0, "command": json 字符串 }	<p>1) token 是用户登陆成功时分配的，必选。</p> <p>2) deveui 是命令下发的目标设备标识；</p> <p>3) devtype 下发目标设备的设备类型；</p> <p>4) confirm: 是否为需确认指令，大部分命令是非确认指令，因此默认为 false；</p> <p>5) devport 是设备侧端口，1~200 有效，0 无效；</p> <p>6) command 是待下发命令，json 格式，其格式和对应的控制指令模型定义保持对应一致，详见后文样例。</p> <p>注 1： Para 参数提交采用 Application/json 编码方法，代码示例：</p> <pre>StringEntity entity = new StringEntity(jsonParam.toS tring(),"utf-8"); entity.setContentEncoding("UTF-8"); entity.setContentType("app lication/json"); httpPost.setEntity(entity) ;</pre>
Method	POST	
响应结果	{ "result": 0, "remark": "remark info", "data": {utf-8 字符串} }	<p>result: 错误码 0 表示成功； 1 表示失败； 2 token 过期或无效；</p> <p>remark: 错误的详细信息</p> <p>data: 返回数据</p> <p>执行成功时返回 cmd 下发命令执行的原始结果数据（字符串，utf-8 编码格式），具体格</p>

		式由应用数据接收方定义和解析，执行失败时返回 null。
--	--	------------------------------

Command (json 字符串) 举例说明:

```
{ "cmd": "cmdname",      // cmd 属性对应指令模型控制字;
  "paras":               // paras 是固定参数名;
    { "p1name": "v1",
      // p1name 是第一个参数名, 根据模型定义填写, 如 p1;
      "p2name": "v2",
      // p2name 是第二个参数名, 根据模型定义填写, 如 p2;
      ...
    }
}
```

3.4.2 设备组播指令下发

【接口功能】：设备组播指令下发。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/devcontrol/broadcastcmd

特别说明：命令参数以 json 对象格式设置到请求的 body 中，具体参见如下接口定义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/devcontrol/broadcastcmd	设备组播指令下发
Para	{ "token": "111111111111", "appeui": "63c4670001111100", "devtype": "haitong_ac", "devport": 0, "bctimes": 1, "command": json 字符串 }	1) token 是用户登陆成功时分配的，必选。 2) appeui 是命令下发的目标应用标识； 3) devtype 下发目标设备的设备类型； 4) devport 是 lora 终端定义的端口号，1~200 有效，0 无效； 5) bctimes: 广播尝试次数，取值范围：1~5。 6) command 是待下发广播命令，json 格式，其格式和对应的控制指令模型定义保持一致，详见后文样例。 注 1： Para 参数提交采用 Application/json 编码方法，代码示例： StringEntity entity = new

		<code>StringEntity(jsonParam.toString(),"utf-8");</code> <code>entity.setContentType("application/json");</code> <code>httpPost.setEntity(entity);</code> <code>;</code>
Method	POST	
响应结果	<code>{</code> <code> "result":0,</code> <code> "remark":"remark info",</code> <code> "data":{utf-8 字符串}</code> <code>}</code>	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期或无效; remark: 错误的详细信息 data: 返回数据 执行成功时返回 cmd 下发命令执行的原始结果数据（字符串，utf-8 编码格式），具体格式由应用数据接收方定义和解析，执行失败时返回 null 。

Command（json 字符串）举例说明：

```
{ "cmd": "cmdname",      // cmd 属性对应指令模型控制字;
  "paras":               // paras 是固定参数名;
    { "p1name": "v1",
      // p1name 是第一个参数名，根据模型定义填写，如 p1;
      "p2name": "v2",
      // p2name 是第二个参数名，根据模型定义填写，如 p2;
      ...
    }
}
```

3.5 业务数据上报

3.5.1 上报接口注册

【接口功能】：注册数据上报地址，系统会根据当前用户对应的项目，将项目下的指定设备类型的终端数据上报到指定地址。

【URL 举例】：
`http://xxxxxx:8093/openapi/v2/datapush/registre?token=xxxx&devtype=xxxxx&desturl=xxxxx`

注意：desturl 中的特殊字符转义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/datapush/registre	注册数据上报地址，系统会根据当前用户对应的项目，将项目下的指定设备类型的终端数据上报到指定地址。 需要数据接收方实现所注册地址对应的服务。
Para	{ "token": "111111111111" , "devtype": "haitong_ac_v1", "desturl": "http://destip:destport/xxxxxx" }	1) token 是用户登陆成功时分配的，必选。 2) devtype 用于指定设备类型，"all"表示所有设备类型。 3) desturl 是业务数据上报地址。上报数据的 json 对象，详细定义参见下表； 4) desturl 调用方法约定为 http 协议，POST 方法，支持 keep-alive 模式。收到数据后立即回复 200 OK。 注 1: devtype 和 desturl 组合起来唯一确定一条注册项（即二者组合起来是主键）。 注 2: para 参数提交采用 URL 编码方法，例如： <code>httpPost.setEntity(new UriEncodedFormEntity(pairList, "utf-8"));</code>
Method	POST	
响应结果	{ "result":0, "remark": "remark info", "data": json }	result: 错误码 0-表示成功; 1-表示失败; 2-token 过期; remark: 错误的详细信息; data: 为 null, 无内容.

注 1: 建议数据接收方采取异步处理策略，比如将接收到的数据临时放入缓存，然后再异步处理，以保证数据上报的效率。

注 2: 对于注册接口设定的 URL，本 API 会做持久化缓存，第三方模块在接收 URL 不变更的情况下，不需要重复注册。

注 3: 注册成功后，本 API 模块会在设备上上报数据后及时向 URL 对应的地址推送数据，

这个推送过程不受用户注册时所分配令牌是否过期的影响，即推送 URL 持久有效。

注 4：设备产生业务数据时，本 API 模块会向第三方注册的 URL 地址推送数据，数据已 json 格式定义，和前文业务数据查询接口返回的格式一致（通常情况下 datarows 只有一条记录）。举例：

```
{
  "devtype": "haitong",
  "datarows":
  [
    {
      "collecttime": "2018-02-01 02:00:00",
      "deveui": "004a070001111156",
      "temperature": 23.9,
      "humidity": 3.8,
      "pressure": 2
    }
  ]
}
```

推送数据的 json 属性定义如下：

字段名	类型	长度	约束	解释
devtype	varchar	200	必选	业务数据对应的设备类型 ID，方便解析。
datarows	数组 []	N	必选	业务数据记录数组名。 数组中每条记录对应一个 json 对象，json 对象属性和数据模型中的 subtype 模型的定义一致。
collecttime	采集时间	100	必选	yyyy-MM-dd HH:mm:ss 格式
deveui	设备标识	50	必选	
xxxx	数据模型字段		可选	和模型定义保持一致

注：和业务数据查询接口相比，少了 devname 字段。

3.5.2 上报接口注销

【接口功能】：取消数据上报地址注册（表示数据接收方不再需要数据上报）。

【URL 举例】：

<http://xxxxxx:8093/openapi/v2/datapush/unregiste?token=xxxx&devtype=xxxxx&desturl=xxxxx>

注意：desturl 中的特殊字符转义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/datapush/unregiste	取消数据上报地址注册（表示数据接收方不再需要数据上报）。
Para	{ "token": "111111111111" , "devtype": "haitong_ac_v1", "desturl": "http://destip:destport/xxxxxx" }	1) token 是用户登陆成功时分配的，必选。 2) devtype 用于指定设备类型，"all"表示所有设备类型。 3) desturl 是业务数据上报地址。 注 1: devtype 和 desturl 组合起来唯一确定一条注册项（即二者组合起来是主键）。 注 2: para 参数提交采用 URL 编码方法，在请求 url 中携带。
Method	DELETE	
响应结果	{ "result":0, "remark": "remark info", "data": json }	result: 错误码 0 表示成功； 1 表示失败； 2 token 过期； remark: 错误的详细信息 data: 为 null，无内容。

3.5.3 上报接口注册查询

【接口功能】：查询当前用户已经注册的上报接口。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/datapush/regitelist?token=xxxx

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/datapush/regitelist	查询当前用户已经注册的上报接口。
Para	{ "token": "111111111111" , }	1) token 是用户登陆成功时分配的，必选。
Method	GET	
响应	{ "result":0, "remark": "remark info", }	result: 错误码 0-表示成功； 1-表示失败；

	"data": json }	2- 表示 token 无效; remark: 错误的详细信息 data: 返回数据, 执行成功时返回数据; 执行失败时返回 null;
--	---------------------------------	---

上报接口注册查询成功时，响应中 **data** 字段对应的内容（数组，json 格式），样例如下：

```
[
  {
    "devtype": "haitong_ac_v1",
    "desturl": "http://destip:destport/xxxxxx",
    "regtime": "2018-03-02 10:00:00"
  },
  ...
]
```

Data 中包含的各字段定义表：

字 段 名	类 型	长 度	约 束	解 释
devtype	varchar	200	必选	设备类型标识
desturl	varchar	1024	必选	设备业务数据上报 URL
regtime	varchar	50	必选	设备业务数据上报 URL 注册时间， 格式：yyyy-MM-dd HH:mm:ss

3.6 告警数据上报

3.6.1 上报接口注册

【接口功能】：注册告警上报地址，系统会根据当前用户对应的项目，将项目下的对应终端相关的告警上报到指定地址。

【URL 举例】：

<http://xxxxxx:8093/openapi/v2/alarmpush/registre?token=xxxx&desturl=xxxxx>

注意：**desturl** 中的特殊字符转义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/alarmpush/registre	注册告警上报地址，系统会根据当前用户对应的项目，将项目下的对应终端相关的告警上报到指定地址。

		需要告警接收方实现所注册地址对应的服务。
Para	{ "token": "111111111111" , "desturl": "http://destip:destport/xxxxxx" }	1) token 是用户登陆成功时分配的，必选。 2) desturl 是告警上报地址。 3) 上报告警数据是 json 格式字符串，UTF-8 编码格式，具体定义见后文； 4) desturl 调用方法约定为 http 协议，POST 方法，支持 keep-alive 模式。收到数据后立即回复 200 OK。 注 1: para 参数提交采用 URL 编码方法，例如： httpPost.setEntity(new UrlEncodedFormEntity(pairList, "utf-8"));_
Method	POST	
响应结果	{ "result":0, "remark": "remark info", "data": json }	result: 错误码 0-表示成功； 1-表示失败； 2-token 过期； remark: 错误的详细信息； data: 为 null，无内容。

注：建议告警接收方采取异步处理策略，比如将接收到的告警临时放入缓存，然后再异步处理，以保证告警上报的效率。

上报的告警数据 json 格式定义如下：

字段名	类型	长度	约束	解释
alarmid	Varcahr	200	必选	告警流水号
devtype	Varcahr	100	必选	设备类型
deveui	Varcahr	200	必选	设备标识
alarmcode	Varcahr	100	必选	告警码
typeflag	Byte	4	必选	类型标记: 0: 告警恢复； 1: 告警发生；

				2: 通知消息；
title	Varcahr	200	必选	告警标题
alarmlevel	Int	10	必选	告警级别，固定为 1
alarmtime	varchar	50	必选	告警上报时间戳，yyyy-MM-dd HH:mm:ss 格式字符串。
descp	varchar	512	可选	告警详细描述
cleartime	varchar	50	可选	告警恢复时间，yyyy-MM-dd HH:mm:ss 格式字符串，可为 null。
confirmstate	byte	4	可选	告警确认状态：0-未确认，1-已确认；
confirmtime	varchar	50	可选	告警确认时间，yyyy-MM-dd HH:mm:ss 格式字符串，可为 null。
confirmer	varchar	20	可选	告警确认人姓名，可为 null
remark	Varchar	512	可选	告警备注信息

注 1：对于物联网终端而言，一个终端对应于一个告警码在特定时间点最多只会有一条告警。因此，通过 deveui+alarmcode+alarmtype 确定对应设备是否产生了某类告警，还是恢复了某类告警。

注 2：有些终端设备回复正常时，并不上报告警恢复消息，因此业务系统需要考虑相应的处理策略（如人工恢复或定时恢复等）。

注 3：告警 json 举例：

```
{
  "alarmid": "xxxxxx",           // 告警流水号
  "devtype": "zte_clss_loc",     // 告警设备类型
  "deveui": "0047c60739448800", // 告警设备标识
  "alarmcode": "1-1",           // 告警码
  "typeflag": 1,                // 类型标记：0：告警恢复；1：告警发生；2：通知消息；
  "title": "xxxx",              // 告警标题
  "alarmlevel": 1,               // 告警级别
  "alarmtime": "yyyy-MM-dd HH:mm:ss", // 告警发生时间
  "descp": "xxxxxxxxxxxxxx",     // 告警详细描述
  "cleartime": null,              // 告警恢复时间
  "confirmstate": 0,             // 告警确认状态：0-未确认，1-已确认；
  "confirmtime": null,           // 告警确认时间，若有值，为 yyyy-MM-dd HH:mm:ss 格式；
  "confirmer": "xxxx",           // 告警确认人
  "remark": "xxxxxxxxxx"         // 告警备注信息。
}
```

}

3.6.2 上报接口注销

【接口功能】：取消告警上报地址注册（表示告警接收方不再需要告警上报）。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/alarmpush/unregiste?token=xxxx&desturl=xxxxx

注意：desturl 中的特殊字符转义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/alarmpush/unregiste	取消告警上报地址注册（表示数据接收方不再需要数据上报）。
Para	{ "token": "111111111111" , "desturl": "http://destip:destport/xxxxxx" }	1) token 是用户登陆成功时分配的，必选。 2) desturl 是告警上报地址。 注 1: para 参数提交采用 URL 编码方法，在请求 url 中携带。
Method	DELETE	
响应结果	{ "result":0, "remark": "remark info", "data": json }	result: 错误码 0 表示成功； 1 表示失败； 2 token 过期； remark: 错误的详细信息 data: 为 null，无内容。

3.6.3 上报接口注册查询

【接口功能】：查询当前用户已经注册的告警上报接口。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/alarmpush/registelst?token=xxxx

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/alarmpush/registelst	查询当前用户已经注册的告警上报接口。
Para	{ "token": "111111111111" , }	1) token 是用户登陆成功时分配的，必选。
Method	GET	

响应	{ "result":0, "remark":"remark info", "data": json }	result: 错误码 0-表示成功; 1-表示失败; 2-表示 token 无效; remark: 错误的详细信息 data: 返回数据, 执行成功时返回数据; 执行失败时返回 null;
----	--	--

上报接口注册查询成功时，响应中 **data** 字段对应的内容（数组，json 格式），样例如下：

```
[  
  {  
    "desturl": "http://destip:destport/xxxxxx",  
    "regtime": "2018-03-02 10:00:00"  
  },  
  ...  
]
```

Data 中包含的各字段定义表：

字段名	类型	长度	约束	解释
desturl	varchar	1024	必选	设备告警数据上报 URL
regtime	varchar	50	必选	设备告警数据上报 URL 注册时间， 格式：yyyy-MM-dd HH:mm:ss

3.7 设备状态上报

3.7.1 上报接口注册

【接口功能】：注册设备状态上报地址，系统会根据当前用户对应的项目，将项目下的对应终端状态上报到指定地址。

特别说明：设备运行状态（上线、离线）仅在变更时会上报一次。设备正常上线后不会定期上报状态，此时第三方系统可以根据有无数据上报来判断设备是否正常在线（如果有 **metedata** 上报或 **alarm** 上报，则可判定设备在线）。

【URL 举例】：

<http://xxxxxx:8093/openapi/v2/devstatepush/registre?token=xxxx&desturl=xxxxx>

注意：**desturl** 中的特殊字符转义。

【接口定义】：

接口参数项	参数取值	备注
-------	------	----

URL	/openapi/v2/devstatepush/registre	注册设备状态上报地址，系统会根据当前用户对应的项目，将项目下的对应终端状态上报到指定地址。 需要接收方实现所注册地址对应的服务。
Para	{ "token": "111111111111" , "desturl": "http://destip:destport/xxxxxx" }	1) token 是用户登陆成功时分配的，必选。 2) desturl 是状态上报地址。 3) 上报状态数据是 json 格式字符串，UTF-8 编码格式，具体定义见后文； 4) desturl 调用方法约定为 http 协议，POST 方法，支持 keep-alive 模式。收到数据后立即回复 200 OK。 注 1: para 参数提交采用 URL 编码方法，例如： httpPost.setEntity(new UrlEncodedFormEntity(pairList, "utf-8"));_
Method	POST	
响应结果	{ "result":0, "remark":"remark info", "data": json }	result: 错误码 0-表示成功; 1-表示失败; 2-token 过期; remark: 错误的详细信息; data: 为 null，无内容.。

注：建议设备状态接收方采取异步处理策略，比如将接收到的状态临时放入缓存，然后再异步处理，以保证状态上报的效率。

上报的设备状态数据 json 格式定义如下：

字段名	类型	长度	约束	解释
rpttime	varchar	50	必选	状态更新时间戳，yyyy-MM-dd HH:mm:ss 格式字符串。
deveui	varchar	50	必选	设备标识 devEUI
state	Int		必选	0--在线； 1--离线；

				2--异常； 3--未知；
--	--	--	--	------------------

设备状态推送举例：

```
{  
  "rpttime": "2018-03-13 10:00:06",  
  "deveui": "004a77021106000d",  
  "state": 0  
}
```

3.7.2 上报接口注销

【接口功能】：取消设备状态上报地址注册（表示接收方不再需要状态上报）。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/devstatepush/unregiste?token=xxxx&desturl=xxxxx

注意：desturl 中的特殊字符转义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/devstatepush/unregiste	取消状态上报地址注册（表示数据接收方不再需要数据上报）。
Para	{ "token": "111111111111" , "desturl": "http://destip:destport/xxxxxx" }	1) token 是用户登陆成功时分配的，必选。 2) desturl 是状态上报地址。 注 1： para 参数提交采用 URL 编码方法，在请求 url 中携带。
Method	DELETE	
响应结果	{ "result": 0, "remark": "remark info", "data": json }	result: 错误码 0 表示成功； 1 表示失败； 2 token 过期； remark: 错误的详细信息 data: 为 null，无内容。

3.7.3 上报接口注册查询

【接口功能】：查询当前用户已经注册的状态上报接口。

【URL 举例】：

http://xxxxxx:8093/openapi/v2/devstatepush/registelists?token=xxxx

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/devstatepush/registelists	查询当前用户已经注册的设备状态上报接口。
Para	{ "token": "111111111111" , }	1) token 是用户登陆成功时分配的，必选。
Method	GET	
响应	{ "result":0, "remark": "remark info", "data": json }	result: 错误码 0-表示成功; 1-表示失败; 2-表示 token 无效; remark: 错误的详细信息 data: 返回数据, 执行成功时返回数据; 执行失败时返回 null;

上报接口注册查询成功时，响应中 data 字段对应的内容（数组，json 格式），样例如下：

```
[  
  {  
    "desturl": "http://destip:destport/xxxxxx",  
    "regtime": "2018-03-02 10:00:00"  
  },  
  ...  
]
```

Data 中包含的各字段定义表：

字 段 名	类 型	长 度	约 束	解 释
desturl	varchar	1024	必选	设备状态数据上报 URL
regtime	varchar	50	必选	设备状态数据上报 URL 注册时间， 格式：yyyy-MM-dd HH:mm:ss

3.8 告警维护接口

3.8.1 告警确认

【接口功能】：告警确认

【URL 举例】：

http://xxxxxx:8093/openapi/v2/alarm/confirm

特别说明：命令参数以 json 对象格式设置到请求的 body 中，具体参见如下接口定义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/alarm/confirm	确认告警
Para	{ "token": "111111111111", "alarmid": "A598809402045293", "confirmrst": "1", "confirmmsg": "已确认", "confirmer": "admin" }	1) token 是用户登陆成功时分配的，必选。 2) alarmid: 告警流水号，必选。 3) confirmrst: 告警确认结果，可选。 0 未确认 1 告警 2 误报 3 已恢复 4 其它 4) confirmmsg: 确认消息，可选。 5) Confirmer 确认人，可选。 不输入，默认取当前登陆的用户。
Method	POST	
响应结果	{ "result": 0, "remark": "remark info", "data": null }	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期或无效; remark: 错误的详细信息 data: 这里为 null

3.8.2 告警反确认

【接口功能】：告警反确认

【URL 举例】：

http://xxxxxx:8093/openapi/v2/alarm/unconfirm

特别说明：命令参数以 json 对象格式设置到请求的 body 中，具体参见如下接口定义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/alarm/unconfirm	反确认告警
Para	{ "token": "111111111111", "alarmid": "A598809402045293" }	1) token 是用户登陆成功时分配的，必选。 2) alarmid: 告警流水号，必选。

Method	POST	
响应结果	{ "result":0, "remark":"remark info", "data": null }	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期或无效; remark: 错误的详细信息 data: 这里为 null

3.8.3 告警手工清除

【接口功能】：告警手工清除

【URL 举例】：

http://xxxxxx:8093/openapi/v2/alarm/clear

特别说明：命令参数以 json 对象格式设置到请求的 body 中，具体参见如下接口定义。

【接口定义】：

接口参数项	参数取值	备注
URL	/openapi/v2/alarm/clear	清队告警
Para	{ "token": "111111111111", "alarmid": "A598809402045293" }	3) token 是用户登陆成功时分配的，必选。 4) alarmid: 告警流水号，必选。
Method	POST	
响应结果	{ "result":0, "remark":"remark info", "data": null }	result: 错误码 0 表示成功; 1 表示失败; 2 token 过期或无效; remark: 错误的详细信息 data: 这里为 null