云平台 3.0 二次开发接口说明书

接口版本	V3.1.2
作者	
创建日期	2021.06.28
更新日期	2024.10.18

公开

文档变更记录

序号	变更说明	作者	日期
1	创建		2021.06.28
2	设备因子信息加入单次报警最大发送条数(语音、短信)		2021.11.10
3	修改实时数据继电器状态，格式修改为 json 字符串		2021.11.17
4	修改设备信息-报警开关参数，增加复归通知		2022.05.20
5	实时数据中增加因子名称字段		2022.05.27
6	新增查询继电器操作记录的接口		2022.06.13
7	修改了实时数据及历史数据的报警级别		2022.06.24
8	增加报警数据查询接口		2023.08.10
9	增加 MQTT 方式获取实时数据		2024.10.18

公开

目录

一、统一说明 1
二、接口详细 1
1、身份验证 1
1.1 根据用户名和密码获取 token 1
2、查询分组列表 3
2.1 查询分组列表 3
2.2 查询设备列表 4
2.3 根据设备地址查询设备信息 7
2.4 根据设备地址获取设备继电器列表 10
2.5 继电器操作 11
3、数据操作 13
3.1 查询实时数据(使用 MQTT 方式见附录) 13
3.2 根据设备地址查询实时数据 16
3.3 获取历史数据列表 18
3.4 删除历史数据 21
3.5 查询继电器操作记录 22
3.6 获取报警数据列表 23
三、附录 26
返回代码 26
使用 MQTT 获取实时数据 26

一、统一说明

1、通讯方式
数据接口采用 http 协议，通讯内容采用 JSON 数据格式。
接口返回数据格式如下：
{"code":1001, "message":"信息内容", "data":object}

ResultData 字段说明

字段	类型	说明
code	Integer	代码(参考返回代码说明表)
message	String	返回操作结果描述
data	Json	Json 数据，具体格式参考各个接口

文档中所有接口的返回数据说明表，如无特殊说明均为 ResultData 数据格式，返回的数据均存放于 data 字段中

2、接口调用注意事项

调用接口（除测试、登录接口外）需要将 token 写入 http 请求头中。请求头字段名固定为 authorization，字段值为获取 token 接口获取到的 token 值。

接口中如无特殊说明，采用 post 方法的接口中参数为 json 格式，放入请求 body 中；采用 get 方法的接口中参数为查询参数，放入请求地址后

3、接口地址

环境云平台　http://www.0531yun.com/

二、接口详细

1、身份验证

1.1 根据用户名和密码获取 token

1.1.1 接口描述

接口编号	1.1
接口名称	根据用户名和密码获取 token
功能描述	获取登录所用的 token
接口地址	/api/getToken

请求方式	GET
返回数据格式	JSON
备注	

表 1.1.1.1

1.1.2 请求参数说明

1.1.2.1 请求参数

字段	类型	说明
loginName	String	登录名
password	String	登录密码

表 1.1.1.2

1.1.3 返回数据说明

字段	类型	说明
expiration	Long	过期时间(秒) – 默认有效期两个小时
token	String	token 令牌

表 1.1.1.3

1.1.4 返回格式示例


```
{
    "code": 1000,
    "data": {
        "expiration": 1691724475,
        "token":
"eyJhbGciOiJIUzI1NiJ9.eyJsb2dpbk5hbWUiOiJjeXkiLCJtb2RlbGUiOiJBUEkiLCJleHAiOjE2OTE3MjQ0NzUsInVzZXJJZCI6IjcONDJkOGYOMGNkZDY4MDBjNzcOYjUONzUyYTM1ODcxIiwiaWFOIjoxNjkxNzE3Mjc1LCJqdGkiOiJOb2tlbklkInO.EObv4NqPNqGRBYOWMnOibgwut8YoTIAV6FKCo7WCN7U"
    },
    "message": "获取成功"
}
```

2、查询分组列表

2.1 查询分组列表

2.1.1 接口描述

接口编号	2.1
接口名称	查询分组列表
功能描述	查询分组列表
接口地址	/api/device/getGroupList
请求方式	GET
返回数据格式	JSON
备注	

表 2.1.1.1

2.1.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	String	token

表 2.1.1.2

请求参数

字段	类型	说明

表 2.1.1.3

2.1.3 返回数据说明

字段	类型	说明
code	int	返回状态码
groupId	String	分组 id
groupName	String	分组名
parentId	String	上级组名
message	String	异常信息

表 2.1.1.4

公开

2.1.4 返回格式示例


```
{
    "code": 1000,
    "message": "获取成功",
    "data": [

        {
            "groupId": "097885e5b167402487831d73022cdf88",
            "parentId": "92b7f08645c7426f945515062b3f9a24",
            "groupName": "气象站检测"
        },
        {
            "groupId": "0e8a13f613d34f8fbc9a74d7a272d022",
            "parentId": "92b7f08645c7426f945515062b3f9a24",
            "groupName": "led 屏幕测试"
        }
    ]
}
```


2.2 查询设备列表

2.2.1 接口描述

接口编号	2.2
接口名称	查询设备列表
功能描述	查询设备列表
接口地址	/api/device/getDeviceList
请求方式	GET
返回数据格式	JSON
备注	

表 2.2.1.1

2.2.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	string	token

表 2.2.1.2

请求参数

字段	类型	说明
groupId	String	组编号，为空则查询所有设备

表 2.2.1.3

2.2.3 返回数据说明

字段	类型	说明
alarmRecord	Integer	是否保存报警数据　　0:不保存 1:保存
alarmSwitch	Integer	报警开关　1: 离线报警 2: 超限报警 4：复归通知 3: 离线报警+超限报警 5：离线报警+复归通知 6：超限报警+复归通知 7：离线报警+超限报警+复归通知
deviceAddr	Integer	设备地址码
deviceName	String	设备名称
groupId	String	分组 id
lat	Double	纬度
Lng	Double	经度
offlineinterval	Integer	离线判断间隔
savedatainterval	Short	保存数据间隔
sort	Integer	排序权重
deviceCode	String	设备编码
useMarkLocation	Boolean	是否标记经纬度　false:否 true:是
factors		详细字段参考表 2.2.1.5

表 2.2.1.4

factors 成员属性表		
字段	类型	说明
factorId	String	因子 id
deviceAddr	Integer	设备地址
nodeId	Integer	节点 id
registerId	Integer	寄存器 id
factorName	String	因子名称
factorIcon	String	因子图标
coefficient	Float	系数
offset	Float	偏差
alarmDelay	Integer	报警延时(秒)
alarmRate	Integer	报警频率（分钟）

backToNormalDelay	Integer	复归延时（秒）
digits	Integer	小数位数
unit	String	单位
enabled	Boolean	是否启用 true:启用 false:关闭
sort	Integer	排序权重(数字越大权重越高）
maxVoiceAlarmTimes	Integer	单次报警语音最大发送条数
maxSmsAlarmTimes	Integer	单次报警短信最大发送条数

表 2.2.1.5

2.2.4 返回格式示例


```
{
    "code": 1000,
    "message": "查询成功",
    "data": [{
        "deviceAddr": 10000000,
        "groupId": "96850381083a4cbfa4559de20cd57105",
        "deviceName": "10000000 真实设备",
        "offlineinterval": 3,
        "savedatainterval": 1,
        "alarmSwitch": 3,
        "alarmRecord": 1,
        "lng": 126.7609083505059,
        "lat": 39.27723009628903,
        "useMarkLocation": false,
        "sort": 1001,
        "factors": [
            {
                "factorId": "10000000_10_1",
                "deviceAddr": 10000000,
                "nodeId": 10,
                "registerId": 1,
                "factorName": "PM10",
                "factorIcon": "0.png",
                "coefficient": 1,
                "offset": 0,
                "alarmDelay": 0,
                "alarmRate": 5,
                "backToNormalDelay":3,
                "digits": 1,
                "unit": "ug/m3",
```

"enabled": true,
"sort": 0,
"maxVoiceAlarmTimes": 3,
"maxSmsAlarmTimes": 3
}
],
"deviceCode": null
}]
}

2.3 根据设备地址查询设备信息

2.3.1 接口描述

接口编号	2.3
接口名称	根据设备地址查询设备信息
功能描述	查询设备信息
接口地址	/api/device/getDevice
请求方式	GET
返回数据格式	JSON
备注	

表 2.3.1.1

2.3.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	string	token

表 2.3.1.2

请求参数

字段	类型	说明
deviceAddr	Integer	设备地址

表 2.3.1.3

2.3.3 返回数据说明

字段	类型	说明
code	int	状态码
alarmRecord	Integer	是否保存报警数据　0:不保存 1:保存
alarmSwitch	Integer	报警开关　1: 离线报警 2: 超限报警 4：复归通知 3：离线报警+超限报警 5：离线报警+复归通知 6：超限报警+复归通知 7：离线报警+超限报警+复归通知
deviceAddr	Integer	设备地址
deviceName	String	设备名称
groupId	String	分组 id
lat	Double	纬度
lng	Double	经度
offlineinterval	Short	离线判断间隔(分钟)
savedatainterval	Short	保存数据间隔(分钟)
sort	Integer	排序权重(数字越大权重越高)
deviceCode	String	设备编码(保留字段用于扫描二维码)
useMarkLocation	Boolean	是否标记经纬度　false:否 true:是
factors		详细字段参考表 2.3.1.5

表 2.3.1.4

factors 成员属性表		
字段	类型	说明
factorId	String	因子 id
deviceAddr	Integer	设备地址
nodeId	Integer	节点 id
registerId	Integer	寄存器 id
factorName	String	因子名称
factorIcon	String	因子图标
coefficient	float	系数
offset	float	偏差
alarmDelay	Integer	报警延时(秒)
alarmRate	Integer	报警频率（分钟）
backToNormalDelay	Integer	复归延时（秒）
digits	Integer	小数位数
unit	String	单位
enabled	Boolean	是否启用　true: 启用

		false:关闭
sort	Integer	排序权重(数字越大权重越高)
maxVoiceAlarmTimes	Integer	单次报警语音最大发送条数
maxSmsAlarmTimes	Integer	单次报警短信最大发送条数

表 2.3.1.5

2.3.4 返回格式示例


```
{
    "code": 1000,
    "message": "查询成功",
    "data": {
        "deviceAddr": 10000000,
        "groupId": "96850381083a4cbfa4559de20cd57105",
        "deviceName": "10000000 真实设备",
        "offlineinterval": 3,
        "savedatainterval": 1,
        "alarmSwitch": 3,
        "alarmRecord": 1,
        "lng": 126.7609083505059,
        "lat": 39.27723009628903,
        "useMarkLocation": false,
        "sort": 1001,
        "factors": [
            {
                "factorId": "10000000_10_1",
                "deviceAddr": 10000000,
                "nodeId": 10,
                "registerId": 1,
                "factorName": "PM10",
                "factorIcon": "0.png",
                "coefficient": 1,
                "offset": 0,
                "alarmDelay": 0,
                "alarmRate": 5,
                "backToNormalDelay":3,
                "digits": 1,
                "unit": "ug/m3",
                "enabled": true,
                "sort": 0,
                "maxVoiceAlarmTimes": 3,
                "maxSmsAlarmTimes": 3
```

```
            }
        ],
        "deviceCode": null
    }
}
```


2.4 根据设备地址获取设备继电器列表

2.4.1 接口描述

接口编号	2.4
接口名称	根据设备地址获取设备继电器列表
功能描述	根据设备地址获取设备继电器列表
接口地址	/api/device/getRelayList
请求方式	GET
返回数据格式	JSON
备注	

表 2.4.1.1

2.4.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	string	token

表 2.4.1.2

请求参数

字段	类型	说明
deviceAddr	Integer	设备地址

表 2.4.1.3

2.4.3 返回数据说明

字段	类型	说明
deviceAddr	Integer	设备地址
deviceName	String	设备名称
enabled	Boolean	是否启用　true:启用 false:不启用

relayName	String	继电器名称
relayNo	Integer	继电器编号
relayStatus	Integer	继电器状态(暂时不用)

表 2.4.1.4

2.4.4 返回格式示例


```
{
    "code": 1000,
    "data": [
        {
            "deviceAddr": 10000020,
            "deviceName": "设备 1",
            "enabled": true,
            "relayName": "继电器 1",
            "relayNo": 1,
            "relayStatus": 0
        }
    ],
    "message": "获取成功"
}
```


2.5 继电器操作

2.5.1 接口描述

接口编号	2.5
接口名称	继电器操作
功能描述	继电器操作
接口地址	/api/device/setRelay
请求方式	POST
返回数据格式	JSON
备注	

表 2.5.1.1

2.5.2 请求参数说明

请求头

header	必选	类型	说明

authorization	是	string	token

表 2.5.1.2

请求参数

字段	类型	说明
deviceAddr	Integer	设备地址
opt	Integer	设备操作　0：闭合　1：断开
relayNo	Integer	继电器编号

表 2.5.1.3

2.5.3 返回数据说明

字段	类型	说明
code	int	状态码
data	Boolean	是否成功
message	String	异常信息

表 2.5.1.4

2.5.4 返回格式示例

```
{
    "code": 1000,
    "data": true,
    "message": "操作成功"
}
```

公开

3、数据操作

3.1 查询实时数据(使用 MQTT 方式见附录)

3.1.1 接口描述

接口编号	3.1
接口名称	查询实时数据
功能描述	查询实时数据
接口地址	/api/data/getRealTimeData
请求方式	GET
返回数据格式	JSON
备注	

表 3.1.1.1

3.1.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	string	token

表 3.1.2.1

请求参数

字段	类型	说明
groupId	String	组编号，为空则查询所有设备

表 3.1.2.2

3.1.3 返回数据说明

字段	类型	说明
deviceAddr	Integer	设备地址码
deviceName	String	设备名称
deviceStatus	String	设备状态 normal:正常 alarming:报警 preAlarming:预警 offline:离线
lat	double	纬度

公开

lng	double	经度
relayStatus	String	继电器状态 Json 字符串， 内含两个属性： relayNo 继电器编号 relayStatus 0:断开 1:闭合
systemCode	String	系统编码
timeStamp	long	时间戳
dataItem		见表 3.1.3.2

表 3.1.3.1

dataItem 成员属性表		
nodeId	int	节点 id
registerItem		见表 3.1.3.3

表 3.1.3.2

registerItem 成员属性表		
registerId	int	寄存器 id
registerName	String	因子名称
data	String	数值
value	Double	实际值
alarmLevel	int	报警级别 0: 正常 1: 报警(超报警下限) 2: 预警(超预警下限) 3: 预警(超预警上限) 4：报警(超报警上限) -2: 遥调(开关量)报警
alarmColor	String	报警颜色
alarmInfo	String	报警消息
unit	String	单位

表 3.1.3.3

3.1.4 返回格式示例


```
{
    "code": 1000,
    "message": "获取成功",
    "data": [
        {
            "systemCode": "iot",
            "deviceAddr": 10000000,
```

"deviceName": "10000000 真实设备",
"lat": 0,
"lng": 0,
"deviceStatus": "normal",
"relayStatus":
"[{"relayNo":1,"relayStatus":0},{"relayNo":2,"relayStatus":1},{"relayNo":3,"relayStatus":1},{"relayNo":4,"relayStatus":1},{"relayNo":5,"relayStatus":1},{"relayNo":6,"relayStatus":1},{"relayNo":7,"relayStatus":1},{"relayNo":8,"relayStatus":1}]",
"dataItem": [
{
"nodeId": 1,
"registerItem": [
{
"registerId": 1,
"registerName": "温度",
"data": "225",
"value": 225,
"alarmLevel": 0,
"alarmColor": "ff0000",
"alarmInfo": "",
"unit": "级"
},
{
"registerId": 2,
"registerName": "湿度",
"data": "47.2",
"value": 47.20000076293945,
"alarmLevel": 0,
"alarmColor": "ff0000",
"alarmInfo": "",
"unit": "m/s"
}
]
}
],
"timeStamp": 1637130231422
}
]
}

3.2 根据设备地址查询实时数据

3.2.1 接口描述

接口编号	3.2
接口名称	根据设备地址查询实时数据
功能描述	根据设备地址查询实时数据
接口地址	/api/data/getRealTimeDataByDeviceAddr
请求方式	GET
返回数据格式	JSON
备注	

表 3.2.1.1

3.2.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	string	token

表 3.2.2.1

请求参数

字段	类型	说明
deviceAddrs	String	设备地址，多个用英文,分隔

表 3.2.2.2

3.2.3 返回数据说明

字段	类型	说明
deviceAddr	Integer	设备地址码
deviceName	String	设备名称
deviceStatus	String	设备状态 normal:正常 alarming:报警 preAlarming:预警 offline:离线
lat	double	纬度
lng	double	经度
relayStatus	String	继电器状态 Json 字符串， 内含两个属性：

		relayNo 继电器编号 relayStatus 0:断开 1:闭合
systemCode	String	系统编码
timeStamp	long	时间戳
dataItem		见表 3.2.3.2

表 3.2.3.1

dataItem 成员属性表		
字段	类型	说明
nodeId	int	节点 id
registerItem		见表 3.2.3.3

表 3.2.3.2

registerItem 成员属性表		
字段	类型	说明
registerId	int	寄存器 id
registerName	String	因子名称
data	String	数值
value	Double	实际值
alarmLevel	int	报警级别 0: 正常 1: 报警(超报警下限) 2: 预警(超预警下限) 3: 预警(超预警上限) 4：报警(超报警上限) -2: 遥调(开关量)报警
alarmColor	String	报警颜色
alarmInfo	String	报警消息
unit	String	单位

表 3.2.3.3

3.2.4 返回格式示例


```
{
    "code": 1000,
    "message": "获取成功",
    "data": [
        {
            "systemCode": "iot",
            "deviceAddr": 10000000,
            "deviceName": "10000000 真实设备",
            "lat": 0,
            "lng": 0,
            "deviceStatus": "normal",
```

"relayStatus": "[{"relayNo":1,"relayStatus":0},{"relayNo":2,"relayStatus":1},{"relayNo":3,"relayStatus":1},{"relayNo":4,"relayStatus":1},{"relayNo":5,"relayStatus":1},{"relayNo":6,"relayStatus":1},{"relayNo":7,"relayStatus":1},{"relayNo":8,"relayStatus":1}]",
"dataItem": [
{
"nodeId": 1,
"registerItem": [
{
"registerId": 1,
"registerName": "温度",
"data": "225",
"value": 225,
"alarmLevel": 0,
"alarmColor": "ff0000",
"alarmInfo": "",
"unit": "级"
},
{
"registerId": 2,
"registerName": "湿度",
"data": "47.2",
"value": 47.20000076293945,
"alarmLevel": 0,
"alarmColor": "ff0000",
"alarmInfo": "",
"unit": "m/s"
}
]
}
],
"timeStamp": 1637130231422
}
]
}

3.3 获取历史数据列表

3.3.1 接口描述

接口编号	3.3

接口名称	获取历史数据列表
功能描述	获取历史数据列表
接口地址	/api/data/historyList
请求方式	GET
返回数据格式	JSON
备注	查询时间精确到分钟 startTime <= 待查询历史数据 < endTime

表 3.3.1.1

3.3.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	string	token

表 3.3.2.1

请求参数

字段	类型	说明
deviceAddr	Integer	设备地址编号
nodeId	Integer	节点编号（如果查询所有节点的数据，则 nodeId 填写-1）
startTime	String	开始时间(YYYY-MM-dd HH:mm:ss)
endTime	String	结束时间(YYYY-MM-dd HH:mm:ss)

表 3.3.2.2

3.3.3 返回数据说明

字段	类型	说明
lng	Double	经度
lat	Double	纬度
deviceAddr	Integer	设备地址
recordTime	long	记录时间(UTC 时间 毫秒数)
nodeId	int	节点 id
recordTimeStr	String	记录时间
recordId	String	历史数据 id
data		见表 3.3.3.2

表 3.3.3.1

data 成员属性表		
字段	类型	说明
alarmLevel	int	报警级别 0: 正常

		1：报警(超报警下限) 2：预警(超预警下限) 3：预警(超预警上限) 4：报警(超报警上限) -2：遥调(开关量)报警
registerId	int	寄存器 id
registerName	String	寄存器名称
text	String	数值
value	Double	数值

表 3.3.3.2

3.3.4 返回格式示例


```
{
    "code": 1000,
    "message": "获取成功",
    "data": [
        {
            "deviceAddr": 21063295,
            "nodeId": 1,
            "data": [
                {
                    "registerId": 1,
                    "registerName": "压力",
                    "value": 49521.0,
                    "text": "49521",
                    "alarmLevel": 0
                }
            ],
            "lat": 0.0,
            "lng": 0.0,
            "recordTime": 1691721068241,
            "recordId": "64d59d6c731cc035b41e44ee",
            "recordTimeStr": "2023-08-11 10:31:08"
        }
    ]
}
```

3.4 删除历史数据

3.4.1 接口描述

接口编号	3.4
接口名称	删除历史数据
功能描述	删除历史数据
接口地址	/api/data/delHistory
请求方式	POST
返回数据格式	JSON
备注	

表 3.4.1.1

3.4.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	string	token

表 3.4.2.1

请求参数

字段	类型	说明
id	String	历史数据 id

表 3.4.2.2

3.4.3 返回数据说明

字段	类型	说明
code	int	状态码
data	Boolean	是否成功
Message	String	异常信息

表 3.4.3.1

3.4.4 返回格式示例

```
{
    "code": 1000,
```

"data": true,
"message": "删除成功"
}

3.5 查询继电器操作记录

3.5.1 接口描述

接口编号	3.5
接口名称	查询继电器操作记录
功能描述	查询继电器操作记录
接口地址	/api/data/getRelayOptRecord
请求方式	GET
返回数据格式	JSON
备注	

表 3.5.1.1

3.5.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	string	token

表 3.5.2.1

请求参数

字段	类型	说明
deviceAddr	Integer	设备地址
beginTime	long	开始时间(时间戳)
endTime	long	结束时间(时间戳)

表 3.5.2.2

3.5.3 返回数据说明

字段	类型	说明
code	int	状态码
data	Boolean	是否成功
Message	String	异常信息

表 3.5.3.1

3.5.4 返回格式示例


```
{
    "code": 1000,
    "message": "获取成功",
    "data": [
        {
            "recordId": "62a04ef355e110067b29c9f1",
            "deviceAdd": 30044592,
            "relayNo": 1,
            "relayName": "继电器 1",
            "createTime": 1654673139904,
            "opt": 1,
            "optUserId": "55b86a1e954aeefad6c4c07d327c8153",
            "optLoginName": "jnrstest"
        },
        {
            "recordId": "62a04ef055e110067b29c9f0",
            "deviceAdd": 30044592,
            "relayNo": 1,
            "relayName": "继电器 1",
            "createTime": 1654673136459,
            "opt": 0,
            "optUserId": "55b86a1e954aeefad6c4c07d327c8153",
            "optLoginName": "jnrstest"
        }
    ]
}
```


3.6 获取报警数据列表

3.6.1 接口描述

接口编号	3.6

接口名称	获取报警数据列表
功能描述	获取报警数据列表
接口地址	/api/data/alarmRecordList
请求方式	GET
返回数据格式	JSON
备注	查询时间精确到分钟 startTime <= 待查询数据 < endTime

表 3.6.1.1

3.6.2 请求参数说明

请求头

header	必选	类型	说明
authorization	是	string	token

表 3.6.2.1

请求参数

字段	类型	说明
deviceAddr	Integer	设备地址编号
nodeId	Integer	节点编号（如果查询所有节点的数据，则 nodeId 填写-1）
startTime	String	开始时间(YYYY-MM-dd HH:mm:ss)
endTime	String	结束时间(YYYY-MM-dd HH:mm:ss)

表 3.6.2.2

3.6.3 返回数据说明

字段	类型	说明
deviceAddr	Integer	设备地址
nodeId	int	节点 id(离线报警数据 nodeId 为 0)
factorId	String	因子 id
factorName	String	因子名称
alarmLevel	int	报警级别: 1: 报警(超报警下限) 2: 预警(超预警下限) 3: 预警(超预警上限) 4：报警(超报警上限) -1: 离线报警 -2：遥调(开关量)报警
dataValue	Double	报警值
dataText	String	报警显示文本
alarmRange	String	报警限值

lat	Double	纬度
lng	Double	经度
recordTime	long	报警数据记录时间(UTC 时间 毫秒数)
handled	Boolean	处理状态 true:已处理 false:未处理
handleMsg	String	处理意见
handleUser	String	处理人(平台用户登录名)
handleTime	long	处理时间(UTC 时间 毫秒数)
recordId	String	报警数据 id

表 3.6.3.1

3.6.4 返回格式示例


```
{
    "code": 1000,
    "message": "获取成功",
    "data": [
        {
            "deviceAddr": 10000020,
            "nodeId": 0,
            "factorId": null,
            "factorName": null,
            "alarmLevel": -1,
            "dataValue": 0.0,
            "dataText": "离线",
            "alarmRange": null,
            "lat": 0.0,
            "lng": 0.0,
            "recordTime": 1691564241731,
            "handled": true,
            "handleMsg": "3333",
            "handleUser": "yanfaceshi",
            "handleTime": 1691631253000,
            "recordId": "64d338d183d6d2396a7416a0"
        },
        {
            "deviceAddr": 10000020,
            "nodeId": 2,
            "factorId": "10000020_2_1",
            "factorName": "温度",
            "alarmLevel": 4,
            "dataValue": 88.8,
            "dataText": "88.8",
```

"alarmRange": "12.0",
"lat": 39.93862648978329,
"lng": 116.55804142885094,
"recordTime": 1691549909450,
"handled": false,
"handleMsg": "",
"handleUser": "",
"handleTime": 0,
"recordId": "64d300d5731cc035b472c3d1"
}
]
}

三、附录

返回代码

代码	说明
1000	接口执行成功，具体参考各个接口
1001	接口执行失败，具体参考各个接口，原因参考 Message
1002	参数错误
1003	程序内部异常，异常信息参考 Message
1004	鉴权失败

使用 MQTT 获取实时数据

1.1 开发说明 (以 Java 为例)

步骤 1) 添加以下依赖到项目 pom.xml 中

```
<dependency>
        <groupId>org.eclipse.paho</groupId>
        <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
        <version>1.2.5</version>
    </dependency>
```

步骤 2) 设置 MQTT 基本连接参数

```
String broker = "tcp://rtd.mqtt.0531yun.com:1883";
String username = "";    //此处用平台用户名
String password = "";    //此处用平台密码
String clientid = "此处用 uuid";
String topic = "rtd/平台 groupId";    // groupId 获取方式见接口 2.1      主题规则为 rtd/groupId
int qos = 0;    //此处值用 0
```

步骤 3) 设置 MQTT 连接选项

```
MqttConnectOptions options = new MqttConnectOptions();
//设置用户名
options.setUserName(username);
//设置密码
options.setPassword(password.toCharArray());
//设置连接超时时间
options.setConnectionTimeout(60);
//设置心跳间隔
options.setKeepAliveInterval(60);
```

步骤 4) 初始化连接

```
MqttClient client = new MqttClient(broker, clientId, new MemoryPersistence());
```

步骤 5) 设置回调

```
client.setCallback(new MqttCallback() {
    @Override
    public void connectionLost(Throwable cause) {
        System.out.println("connectionLost: " + cause.getMessage());
    }
    @Override
    public void messageArrived(String topic, MqttMessage message) {
        System.out.println("topic: " + topic);
        System.out.println("Qos: " + message.getQos());
        System.out.println("message content: " + new String(message.getPayload()));
    }
    @Override
    public void deliveryComplete(IMqttDeliveryToken token) {
```

```
            System.out.println("deliveryComplete---------" + token.isComplete());
        }
});
```

步骤 6) 建立连接

```
client.connect(options);
```

步骤 7) 设置订阅

```
client.subscribe(topic, qos);
```

步骤 7) 成功建立连接和订阅主题后，即可在步骤四设置的回调中收到数据

1.2 返回类型

mqtt 接收实时数据类型为 utf8 编码的 json 字符串　　字段说明参考表 3.1.3

1.3 返回示例


```
{
  "dataItem": [
    {
      "nodeId": 3,
      "registerItem": [
        {
          "alarmColor": "ff0000",
          "alarmInfo": "",
          "alarmLevel": 0,
          "data": "0.0",
          "registerId": 1,
          "unit": "cm",
          "value": 0
        },
        {
          "alarmColor": "ff0000",
          "alarmInfo": "",
          "alarmLevel": 0,
          "data": "0.0",
          "registerId": 2,
          "unit": "mm",
```

"value": 0
}
]
},
{
"nodeId": 4,
"registerItem": [
{
"alarmColor": "ff0000",
"alarmInfo": "",
"alarmLevel": 0,
"data": "0.0",
"registerId": 1,
"unit": "cm",
"value": 0
},
{
"alarmColor": "ff0000",
"alarmInfo": "",
"alarmLevel": 0,
"data": "0.0",
"registerId": 2,
"unit": "mm",
"value": 0
}
]
}
],
"deviceAddr": 10000000,
"deviceName": "10000000 真实设备 i1",
"deviceStatus": "normal",
"lat": 0,
"lng": 0,
"relayStatus": "[{ \"relayNo\": 1,\"relayStatus\": 0},{ \"relayNo\": 2,\"relayStatus\": 1},{ \"relayNo\": 3,\"relayStatus\": 0},{ \"relayNo\": 4,\"relayStatus\": 0},{ \"relayNo\": 5,\"relayStatus\": 0},{ \"relayNo\": 6,\"relayStatus\": 0},{ \"relayNo\": 7,\"relayStatus\": 0},{ \"relayNo\": 8,\"relayStatus\": 0}]",
"systemCode": "iot",
"timeStamp": 1728891590551
}

1.4 使用说明

先使用二次开发接口获取分组列表，使用 MQTT 订阅某些特定主题，主题格式为: rtd/ + groupId，需要平台账户具有查看相关分组的权限，示例代码见附件 SubscribeDemo.java

SubscribeDemo.java

公开