

# 智能家居系统项目接口规范

项目名称 Project name	密级 Confidentiality level
	内部公开
文档版本 Document version	发布日期 Release date

---

## 目录

1	http 消息(www-form).....	7
1.1	http 消息封装.....	7
1.2	网络权限.....	9
2	用户接口.....	9
2.1	登录接口.....	9
2.1.1	消息地址 .....	9
2.1.2	消息类型 .....	9
2.1.3	请求消息 .....	10
2.1.4	响应消息 .....	10
2.1.5	实现 .....	11
2.2	获取用户信息.....	11
2.2.1	消息地址 .....	11
2.2.2	消息类型 .....	12
2.2.3	请求消息 .....	12
2.2.4	响应消息 .....	12
2.3	更新用户.....	13
2.3.1	消息地址 .....	13
2.3.2	消息类型 .....	13
2.3.3	请求消息 .....	13
2.3.4	响应消息 .....	14
2.4	获取验证码-邮箱 .....	14
2.4.1	消息地址 .....	14
2.4.2	消息类型 .....	14
2.4.3	请求消息 .....	14
2.4.4	响应消息 .....	15

---

2.5	注册用户-邮箱 .....	15
2.5.1	消息地址 .....	15
2.5.2	消息类型 .....	15
2.5.3	请求消息 .....	15
2.5.4	响应消息 .....	16
2.6	忘记密码-邮箱 .....	16
2.6.1	消息地址 .....	16
2.6.2	消息类型 .....	16
2.6.3	请求消息 .....	16
2.6.4	响应消息 .....	17
3	智能家居接口.....	17
3.1	用户登录.....	17
3.1.1	消息地址 .....	17
3.1.2	消息类型 .....	17
3.1.3	请求消息 .....	17
3.1.4	响应消息 .....	18
3.2	聊天室-推送消息(webSocket) .....	18
3.2.1	消息地址 .....	18
3.2.2	请求消息 .....	18
3.2.3	响应消息 .....	19
3.3	聊天室-发送消息 .....	19
3.3.1	消息地址 .....	19
3.3.2	消息类型 .....	19
3.3.3	请求消息 .....	19
3.3.4	响应消息 .....	20
3.4	房间信息-推送消息(webSocket) .....	20
3.4.1	消息地址 .....	20

---

3.4.2	请求消息 .....	20
3.4.3	响应消息 .....	21
3.5	房间信息-更新 .....	21
3.5.1	消息地址 .....	21
3.5.2	消息类型 .....	21
3.5.3	请求消息 .....	21
3.5.4	响应消息 .....	22
3.5.5	测试用例 .....	22
3.6	房间信息-查询列表 .....	23
3.6.1	消息地址 .....	23
3.6.2	消息类型 .....	23
3.6.3	请求消息 .....	23
3.6.4	响应消息 .....	23
3.6.5	测试用例 .....	25
3.7	传感器-推送消息(webSocket) .....	25
3.7.1	消息地址 .....	25
3.7.2	请求消息 .....	25
3.7.3	响应消息 .....	26
3.8	传感器-更新 .....	26
3.8.1	消息地址 .....	26
3.8.2	消息类型 .....	26
3.8.3	请求消息 .....	26
3.8.4	响应消息 .....	27
3.8.5	测试用例 .....	28
3.9	查询房间里-传感器列表 .....	29
3.9.1	消息地址 .....	29
3.9.2	消息类型 .....	29

---

3.9.3	请求消息 .....	29
3.9.4	响应消息 .....	29
3.10	查询传感器信息.....	34
3.10.1	消息地址.....	34
3.10.2	消息类型.....	34
3.10.3	请求消息.....	34
3.10.4	响应消息.....	34
3.11	MQTT 设备信息.....	35
3.11.1	设备信息 1.....	35
3.11.2	设备信息 2.....	36
3.11.3	服务-智能风扇 .....	37
3.11.4	服务-PM2.5 检测器.....	39
3.11.5	服务-LED 灯 .....	40
3.12	AMQP 消息推送(webSocket) .....	41
3.12.1	消息地址.....	41
3.12.2	请求消息.....	41
3.12.3	响应消息-智能风扇 .....	42
3.12.4	响应消息-PM2.5 检测器.....	43
3.12.5	响应消息-LED 灯 .....	43
3.13	华为云鉴权.....	44
3.13.1	消息地址.....	44
3.13.2	消息类型.....	44
3.13.3	消息方法.....	44
3.13.4	请求消息.....	44
3.13.5	响应消息.....	45
3.14	查询设备.....	45
3.14.1	消息地址.....	45

---

3.14.2	消息类型.....	46
3.14.3	消息方法.....	46
3.14.4	请求消息.....	46
3.14.5	响应消息.....	47
3.15	查询设备属性-同步消息 .....	47
3.15.1	消息地址.....	48
3.15.2	消息类型.....	48
3.15.3	消息方法.....	48
3.15.4	请求消息.....	48
3.15.5	响应消息.....	49
3.15.6	平台查询设备属性.....	49
3.15.7	属性查询响应结果.....	50

# 1 http 消息(www-form)

## 1.1 http 消息封装

```
import http from '@ohos.net.http';
const BASE_ADDR= "http://1.92.113.192:8090"

class ExtraDataEntity {
  name: string
  value: string|number|boolean

  constructor(name: string,value: string) {
    this.name = name;
    this.value = value;
  }
}

class AjaxEntity {
  url: string
  extraData: Array<ExtraDataEntity>
  success:(data:Object)=>void
  fail:(data:Object)=>void

  constructor(url: string,extraData:
Array<ExtraDataEntity>, success: (data:Object)=>void, fail: (data:Object)=>void) {
    this.url = url;
    this.extraData = extraData;
    this.success = success;
    this.fail = fail;
  }
}

export default class EduUtils {

  static sendHttpMsg(value:AjaxEntity) {
    let httpRequest = http.createHttp();
```

```
httpRequest.on('headersReceive', (header) => {
    console.info('header: ' + JSON.stringify(header));
});

httpRequest.request(
    // 填写 http 请求的 url 地址，可以带参数也可以不带参数。URL 地址需要开发者自定义。请求的
    // 参数可以在 extraData 中指定
    BASE_ADDR + value.url,
    {
        method: http.RequestMethod.POST, // 可选，默认为 http.RequestMethod.GET
        // 开发者根据自身业务需要添加 header 字段
        header: {
            'Content-Type': 'application/x-www-form-urlencoded'
        },
        // 当使用 POST 请求时此字段用于传递内容
        extraData: EduUtils.changeWWWParam(value.extraData),
    }, (err, data) => {
        if (!err) {
            if(data.result) {
                try{
                    var rst:string = data.result.toString();
                    value.success(JSON.parse(rst))
                }catch(error) {
                    console.info("error:" + JSON.stringify(error));
                }
            }else{
                value.success(null)
            }
        } else {
            console.info("send msg fail:" + JSON.stringify(err) + JSON.stringify(data))
            value.fail(err)
        }
    }
);
} //end sendHttpMsg

static changeWWWParam(dataArray:Array<ExtraDataEntity>){
    var rst:string = '';
    dataArray.forEach((item)=>{
```





```
rst+=item.name + "=" + item.value + "&"
})
if(rst !== ''){
    rst = rst.substring(0,rst.length - 1)
}
console.info("changeExtraDataWWW:" + rst)
return rst
}
}
```

## 1.2 网络权限

```
"requestPermissions": [
  {
    "name": "ohos.permission.INTERNET"
  }
],
```

# 2 用户接口

## 2.1 登录接口

### 2.1.1 消息地址

/api/user/login

### 2.1.2 消息类型

'Content-Type': 'application/x-www-form-urlencoded'



### 2.1.3 请求消息

消息类型	字段	名称	备注
请求接口	userName	用户名	
	password	密码	

userName=admin&password=123456

### 2.1.4 响应消息

消息类型	字段	名称	备注
响应接口	success	是否成功	
	msg	消息	
	description	描述	
	email	邮箱	
	mobile	手机号码	
	niceName	昵称	
	userName	用户名	
	userId	用户编码	

```
{  
  "msg": "验证成功",  
  "success": true,  
  "obj": {  
    "email": "admin@iss.com",  
    "mobile": "13461106251",  
  }  
}
```



```
"niceName": "管理员",  
"userName": "admin",  
"userId": "999"  
},  
"description": "验证成功"  
}
```

### 2.1.5 实现

```
sendHttpMsg() {  
  var that = this  
  EduUtils.sendHttpMsg(  
    {  
      url: '/api/user/login',  
      extraData: [  
        {name: "userName", value: this.userName},  
        {name: "password", value: this.password}  
      ],  
      success: (data: LoginMsg) => {  
        console.info("login data:" + JSON.stringify(data))  
        promptAction.showToast({message: data.msg})  
        if (data.success) {  
          that.userId = data.obj.userId  
          router.replaceUrl({url: "pages/Index"})  
        }  
      },  
      fail: (data) => {  
        promptAction.showToast({message: "登录失败"})  
      }  
    }  
  )  
}
```

## 2.2 获取用户信息

### 2.2.1 消息地址

```
/api/user/queryUserInfoById
```



### 2.2.2 消息类型

'Content-Type': 'application/x-www-form-urlencoded'

### 2.2.3 请求消息

消息类型	字段	名称	备注
请求接口	userId	用户编码	

userId=999

### 2.2.4 响应消息

消息类型	字段	名称	备注
响应接口	userId	用户编码	
	email	邮箱	
	mobile	手机号码	
	nickName	昵称	
	userName	用户名	
	photo	头像地址	
	dvcId	设备 ID	

```
{
  "userId": "999",
  "userName": "admin",
  "email": "admin@iss.com",
```

```
"mobile": "13461106251",  
"niceName": "管理员",  
"photo": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/headPhoto/admin/headPhoto.jpg"  
}
```

## 2.3 更新用户

### 2.3.1 消息地址

```
/api/user/updateUserInfo
```

### 2.3.2 消息类型

```
'Content-Type': 'application/x-www-form-urlencoded'
```

### 2.3.3 请求消息

消息类型	字段	名称	备注
请求接口	userId	用户编码	必选
	userName	用户名	可选
	email	邮箱	可选
	mobile	手机号	可选
	niceName	昵称	可选
	photo	用户头像地址	可选
	dvcUuid	设备 Id	可选

```
userId=999
```

### 2.3.4 响应消息

消息类型	字段	名称	备注
响应接口	success	是否成功	
	msg	消息	
	description	描述	

## 2.4 获取验证码-邮箱

### 2.4.1 消息地址

/api/user/getVerifyCode

### 2.4.2 消息类型

'Content-Type': 'application/x-www-form-urlencoded'

### 2.4.3 请求消息

消息类型	字段	名称	备注
请求接口	邮箱	email	必选

email=xinfenhua1@163.com



#### 2.4.4 响应消息

消息类型	字段	名称	备注
响应接口	success	是否成功	
	msg	消息	
	description	描述	
	code	验证码	

### 2.5 注册用户-邮箱

#### 2.5.1 消息地址

/api/user/registerUserByEmail

#### 2.5.2 消息类型

'Content-Type': 'application/x-www-form-urlencoded'

#### 2.5.3 请求消息

消息类型	字段	名称	备注
请求接口	邮箱	email	必选
	密码	password	必选

email=xinfenhua1@163.com&password=234567

## 2.5.4 响应消息

消息类型	字段	名称	备注
响应接口	success	是否成功	
	msg	消息	
	description	描述	

## 2.6 忘记密码-邮箱

### 2.6.1 消息地址

/api/user/forgetPasswdByEmail

### 2.6.2 消息类型

'Content-Type': 'application/x-www-form-urlencoded'

### 2.6.3 请求消息

消息类型	字段	名称	备注
请求接口	邮箱	email	必选
	密码	password	必选

email=xinfenhua1@163.com&password=234567



### 2.6.4 响应消息

消息类型	字段	名称	备注
响应接口	success	是否成功	
	msg	消息	
	description	描述	

## 3 智能家居接口

### 3.1 用户登录

#### 3.1.1 消息地址

/api/user/loginJson

#### 3.1.2 消息类型

'Content-Type': 'application/json'

#### 3.1.3 请求消息

消息类型	字段	名称	备注
请求接口	userName	用户名	
	password	密码	

```
{
  'userName': 'zhaowei',
  'password': '123456'
}
```

### 3.1.4 响应消息

消息类型	字段	名称	备注
响应接口	success	是否成功	
	msg	消息	
	description	描述	
	token	Token 码	

```
{
  'success': true,
  'msg': '验证成功',
  'description': '验证成功',
  'token': '07387508-32c3-11ef-ab56-fa163e0d65b3',
}
```

## 3.2 聊天室-推送消息(webSocket)

### 3.2.1 消息地址

```
/api/chatSocketServer/{userId}
```

### 3.2.2 请求消息

消息类型	字段	名称	备注
------	----	----	----



请求接口	userId	消息 Id	
------	--------	-------	--

### 3.2.3 响应消息

消息类型	字段	名称	备注
响应接口	Hello world	建立成功	

## 3.3 聊天室-发送消息

### 3.3.1 消息地址

/api/chat/room/sendMsg

### 3.3.2 消息类型

'Content-Type': 'application/x-www-form-urlencoded'

### 3.3.3 请求消息

消息类型	字段	名称	备注
请求接口	id	用户编码	
	message	消息内容	

id=10&message=sendMessage



### 3.3.4 响应消息

消息类型	字段	名称	备注
响应接口	success	是否成功	
	msg	消息	
	description	描述	

```
{  
  "msg": "发送消息成功。消息： sendMessage",  
  "success": true,  
  "description": "发送消息成功。消息： sendMessage"  
}
```

## 3.4 房间信息-推送消息(webSocket)

### 3.4.1 消息地址

/api/roomSocketServer/{roomId}

### 3.4.2 请求消息

消息类型	字段	名称	备注
请求接口	roomId	房间 Id	

### 3.4.3 响应消息

消息类型	字段	名称	备注
响应接口	Hello world	建立成功	

## 3.5 房间信息-更新

### 3.5.1 消息地址

/api/hall/updateRoomInfo

### 3.5.2 消息类型

'Content-Type': 'application/x-www-form-urlencoded'

### 3.5.3 请求消息

消息类型	字段	名称	备注
请求接口	roomId	房间编码	
	dvcUuid	设备编码	
	roomName	房间名称	
	sensorNum	传感器数量	
	temperature	温度	
	humidity	湿度	



	brightness	亮度	
	roomIcon	房间图标	
	roomSort	排序	

```
roomId=rr1&dvcUuid=aa1&roomName=室外&sensorNum=4
&temperature=30&humidity=49&brightness=51&roomSort=1
&roomIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/room/home1.png
```

### 3.5.4 响应消息

消息类型	字段	名称	备注
响应接口	success	是否成功	
	msg	消息	增加房间成功 更新房间成功
	description	描述	

```
{
  "msg": "更新房间成功",
  "success": true,
  "description": "更新房间成功"
}
```

### 3.5.5 测试用例

```
http://1.92.113.192:8090/api/hall/updateRoomInfo?roomId=rr1&dvcUuid=aa1
&roomName=室外&sensorNum=4&temperature=30&humidity=49&brightness=51&roomSort=1
&roomIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/room/home1.png
```

```
http://1.92.113.192:8090/api/hall/updateRoomInfo?roomId=rr2&dvcUuid=aa1
&roomName=主卧&sensorNum=5&temperature=31&humidity=50&brightness=52&roomSort=2
&roomIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/room/home2.png
```

```
http://1.92.113.192:8090/api/hall/updateRoomInfo?roomId=rr3&dvcUuid=aa1
&roomName=厨房&sensorNum=6&temperature=32&humidity=51&brightness=53&roomSort=3
&roomIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/room/home3.png
```

## 3.6 房间信息-查询列表

### 3.6.1 消息地址

```
/api/hall/queryRoomListByDvcId
```

### 3.6.2 消息类型

```
'Content-Type': 'application/x-www-form-urlencoded'
```

### 3.6.3 请求消息

消息类型	字段	名称	备注
请求接口	dvcUuid	设备编码	

```
dvcUuid=aa1
```

### 3.6.4 响应消息

消息类型	字段	名称	备注
------	----	----	----



响应接口	roomList	房间列表	[
	roomId	房间编码	
	dvcUuid	设备编码	
	roomName	房间名称	
	sensorNum	传感器数量	
	temperature	温度	
	humidity	湿度	
	brightness	亮度	
	roomIcon	房间图标	
			]

```
[{
  "brightness": 51,
  "dvcUuid": "aa1",
  "humidity": 49,
  "roomIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/room/home1.png",
  "roomId": "rr1",
  "roomName": "室外",
  "sensorNum": 4,
  "temperature": 30
},
{
  "brightness": 52,
  "dvcUuid": "aa1",
  "humidity": 50,
  "roomIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/room/home2.png",
  "roomId": "rr2",
  "roomName": "主卧",
  "sensorNum": 5,
  "temperature": 31
}]
```



```

},
{
  "brightness": 53,
  "dvcUuid": "aa1",
  "humidity": 51,
  "roomIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/room/home3.png",
  "roomId": "rr3",
  "roomName": "厨房",
  "sensorNum": 6,
  "temperature": 32
}
  
```

### 3.6.5 测试用例

```
http://1.92.113.192:8090/api/hall/queryRoomListByDvcId?dvcUuid=aa1
```

## 3.7 传感器-推送消息(webSocket)

### 3.7.1 消息地址

```
/api/sensorSocketServer/{sensorId}
```

### 3.7.2 请求消息

消息类型	字段	名称	备注
请求接口	sensorId	传感器 Id	



### 3.7.3 响应消息

消息类型	字段	名称	备注
响应接口	Hello world	建立成功	

## 3.8 传感器-更新

### 3.8.1 消息地址

/api/hall/updateSensorInfo

### 3.8.2 消息类型

'Content-Type': 'application/x-www-form-urlencoded'

### 3.8.3 请求消息

消息类型	字段	名称	备注
请求接口	sensorId	传感器编码	
	roomId	房间编码	
	sensorName	传感器名称	
	sensorMessage1	消息 1	
	sensorMessage2	消息 2	
	sensorValue1	值 1	

	sensorValue2	值 2	
	sensorType	类型	0:未知类型 1:空气净化器 2:智能 led 灯 3:智能门锁 4:电水壶
	sensorIcon	传感器图标	
	sensorSort	排序	

```

sensorId=ss1&roomId=rr1&sensorName=空气净化器&sensorMessage1=温度
&sensorMessage2=湿度&sensorValue1=30℃
&sensorValue2=49%25&sensorType=1&sensorSort=1
&sensorIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u316.png
    
```

### 3.8.4 响应消息

消息类型	字段	名称	备注
响应接口	success	是否成功	
	msg	消息	增加传感器成功 更新传感器成功
	description	描述	

```

{
  "msg": "更新传感器成功",
  "success": true,
  "description": "更新传感器成功"
}
    
```

### 3.8.5 测试用例

http://1.92.113.192:8090/api/hall/updateSensorInfo?sensorId=ss1&roomId=rr1&sensorName=空气净化器 &sensorMessage1= 温度 &sensorMessage2= 湿度 &sensorValue1=30 °C &sensorValue2=49%25&sensorType=1&sensorSort=1&sensorIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u316.png

http://1.92.113.192:8090/api/hall/updateSensorInfo?sensorId=ss2&roomId=rr1&sensorName=智能 LED 灯 &sensorMessage1= 状态 &sensorMessage2= 亮度 &sensorValue1=open&sensorValue2=50%25&sensorType=2&sensorSort=2 &sensorIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u317.png

http://1.92.113.192:8090/api/hall/updateSensorInfo?sensorId=ss3&roomId=rr1&sensorName=智能门锁&sensorMessage1=状态&sensorMessage2=功能 &sensorValue1=close&sensorValue2=运动追踪&sensorType=3&sensorSort=3 &sensorIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u445.png

http://1.92.113.192:8090/api/hall/updateSensorInfo?sensorId=ss4&roomId=rr1 &sensorName=电水壶&sensorMessage1=状态&sensorMessage2=温度 &sensorValue1=open&sensorValue2=70 °C &sensorType=4&sensorSort=4 &sensorIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u446.png

http://1.92.113.192:8090/api/hall/updateSensorInfo?sensorId=ss5&roomId=rr1 &sensorName= 智能 摄 像 机 &sensorMessage1= 状 态 &sensorMessage2= 功 能 &sensorValue1=open&sensorValue2=运动追踪&sensorType=0&sensorSort=5 &sensorIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u417.png

http://1.92.113.192:8090/api/hall/updateSensorInfo?sensorId=ss6&roomId=rr1 &sensorName=智能摄像机&sensorMessage1=状态&sensorMessage2=功能 &sensorValue1=close&sensorValue2=运动追踪&sensorType=0&sensorSort=6 &sensorIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u417.png

http://1.92.113.192:8090/api/hall/updateSensorInfo?sensorId=ss7&roomId=rr1 &sensorName=Wifi 放大器&sensorMessage1=状态&sensorMessage2=连接 &sensorValue1=open&sensorValue2=2 个设备&sensorType=0&sensorSort=7 &sensorIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u433.png

```
http://1.92.113.192:8090/api/hall/updateSensorInfo?sensorId=ss8&roomId=rr1
&sensorName=路由器&sensorMessage1=状态&sensorMessage2=连接
&sensorValue1=open&sensorValue2=2 个设备&sensorType=0&sensorSort=8
&sensorIcon=https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u443.png
```

### 3.9 查询房间里-传感器列表

#### 3.9.1 消息地址

```
/api/hall/querySensorListByRoom
```

#### 3.9.2 消息类型

```
'Content-Type': 'application/x-www-form-urlencoded'
```

#### 3.9.3 请求消息

消息类型	字段	名称	备注
请求接口	roomId	房间编码	

```
roomId=rr1
```

#### 3.9.4 响应消息

消息类型	字段	名称	备注
响应接口	roomId	房间编码	
	dvcUuid	设备编码	



	roomName	房间名称	
	sensorNum	传感器数量	
	temperature	温度	
	humidity	湿度	
	brightness	亮度	
	roomIcon	房间图标	
	sensorEntities	传感器列表	[
	sensorId	传感器编码	
	roomId	房间编码	
	sensorName	传感器名称	
	sensorMessage1	消息 1	
	sensorMessage2	消息 2	
	sensorValue1	值 1	
	sensorValue2	值 2	
	sensorType	类型	0:未知类型 1:空气净化器 2:智能 led 灯 3:智能门锁 4:电水壶
	sensorIcon	传感器图标	
	sensorSort	排序	

		End 传感器列表	]
--	--	-----------	---

```
{
  "dvcUuid": "aa1",
  "roomName": "室外",
  "roomId": "rr1",
  "sensorNum": 4,
  "humidity": 49,
  "temperature": 30,
  "brightness": 51,
  "roomIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/room/home1.png",
  "sensorEntities": [{
    "createTime": "2024-04-11 17:55:25",
    "roomId": "rr1",
    "sensorIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u316.png",
    "sensorId": "ss1",
    "sensorMessage1": "温度",
    "sensorMessage2": "湿度",
    "sensorName": "空气净化器",
    "sensorSort": 1,
    "sensorType": 1,
    "sensorValue1": "30℃",
    "sensorValue2": "49%"
  },
  {
    "createTime": "2024-04-12 11:42:57",
    "roomId": "rr1",
    "sensorIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u317.png",
    "sensorId": "ss2",
    "sensorMessage1": "状态",
    "sensorMessage2": "亮度",
    "sensorName": "智能 LED 灯",
    "sensorSort": 2,
    "sensorType": 2,
    "sensorValue1": "open",
    "sensorValue2": "50%"
  },
  {
    "createTime": "2024-04-12 11:45:14",
```



```
"roomId": "rr1",
"sensorIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u445.png",
"sensorId": "ss3",
"sensorMessage1": "状态",
"sensorMessage2": "功能",
"sensorName": "智能门锁",
"sensorSort": 3,
"sensorType": 3,
"sensorValue1": "close",
"sensorValue2": "运动追踪"
},
{
  "createTime": "2024-04-12 11:47:26",
  "roomId": "rr1",
  "sensorIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u446.png",
  "sensorId": "ss4",
  "sensorMessage1": "状态",
  "sensorMessage2": "温度",
  "sensorName": "电水壶",
  "sensorSort": 4,
  "sensorType": 4,
  "sensorValue1": "open",
  "sensorValue2": "70℃"
},
{
  "createTime": "2024-04-12 15:35:45",
  "roomId": "rr1",
  "sensorIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u417.png",
  "sensorId": "ss5",
  "sensorMessage1": "状态",
  "sensorMessage2": "功能",
  "sensorName": "智能摄像机",
  "sensorSort": 5,
  "sensorType": 0,
  "sensorValue1": "open",
  "sensorValue2": "运动追踪"
},
{
  "createTime": "2024-04-12 15:39:09",
  "roomId": "rr1",
```





```
"sensorIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u417.png",
"sensorId": "ss6",
"sensorMessage1": "状态",
"sensorMessage2": "功能",
"sensorName": "智能摄像机",
"sensorSort": 6,
"sensorType": 0,
"sensorValue1": "close",
"sensorValue2": "运动追踪"
},
{
  "createTime": "2024-04-12 15:39:24",
  "roomId": "rr1",
  "sensorIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u433.png",
  "sensorId": "ss7",
  "sensorMessage1": "状态",
  "sensorMessage2": "连接",
  "sensorName": "WiFi 放大器",
  "sensorSort": 7,
  "sensorType": 0,
  "sensorValue1": "open",
  "sensorValue2": "2 个设备"
},
{
  "createTime": "2024-04-12 15:39:31",
  "roomId": "rr1",
  "sensorIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/u443.png",
  "sensorId": "ss8",
  "sensorMessage1": "状态",
  "sensorMessage2": "连接",
  "sensorName": "路由器",
  "sensorSort": 8,
  "sensorType": 0,
  "sensorValue1": "open",
  "sensorValue2": "2 个设备"
}
}]
}
```



### 3.10 查询传感器信息

#### 3.10.1 消息地址

/api/hall/querySensorInfoById

#### 3.10.2 消息类型

'Content-Type': 'application/x-www-form-urlencoded'

#### 3.10.3 请求消息

消息类型	字段	名称	备注
请求接口	sensorId	传感器编码	

sensorId=st1

#### 3.10.4 响应消息

消息类型	字段	名称	备注
响应接口	sensorId	传感器编码	
	roomId	房间编码	
	sensorName	传感器名称	
	sensorMessage1	消息 1	
	sensorMessage2	消息 2	

	sensorValue1	值 1	
	sensorValue2	值 2	
	sensorType	类型	0:未知类型 1:空气净化器 2:智能 led 灯 3:智能门锁 4:电水壶
	sensorIcon	传感器图标	
	sensorSort	排序	

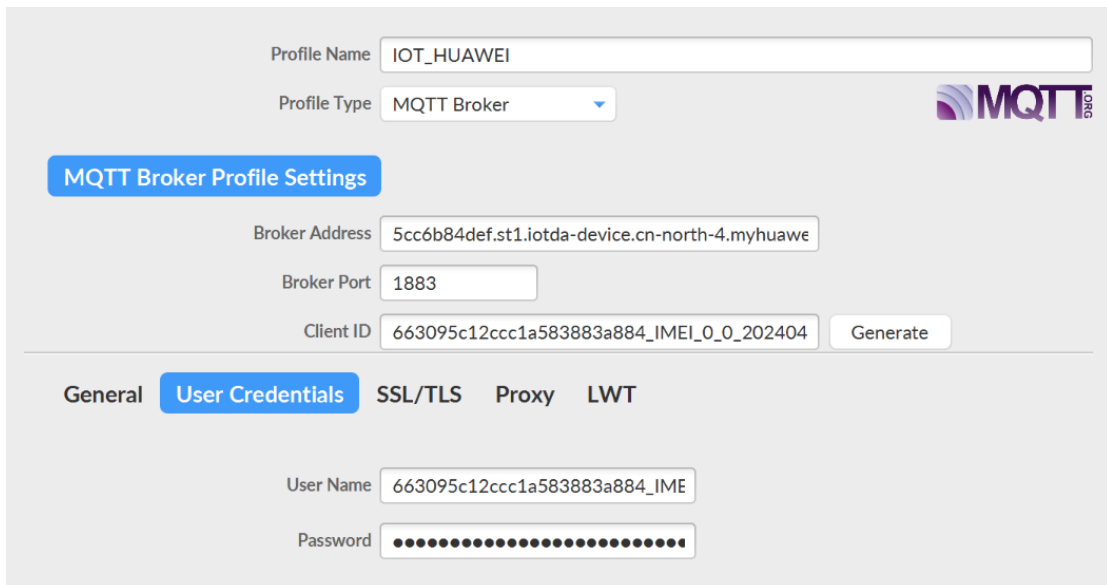
```
{
  "createTime": "2024-06-05 16:23:13",
  "roomId": "aa1",
  "sensorIcon": "https://java-11.obs.cn-north-4.myhuaweicloud.com:443/sensor/purifierV2.png",
  "sensorId": "st1",
  "sensorMessage1": "PM2.5 值",
  "sensorMessage2": "空气质量",
  "sensorName": "空气净化器",
  "sensorSort": 1,
  "sensorType": 1,
  "sensorValue1": "39",
  "sensorValue2": "优"
}
```

### 3.11 MQTT 设备信息

#### 3.11.1 设备信息 1

字段	值
Profile Type	MQTT Broker

Broker Address	5cc6b84def.st1.iotda-device.cn-north-4.myhuaweicloud.com
Broker Port	1883
Client ID	663095c12ccc1a583883a884_IMEI_0_0_2024080101
User Name	663095c12ccc1a583883a884_IMEI
Password	4038e766472d9fab20dc82310809e3407a0c984feebd90278a9fcb6631893e94
Push Address	\$oc/devices/663095c12ccc1a583883a884_IMEI/sys/properties/report

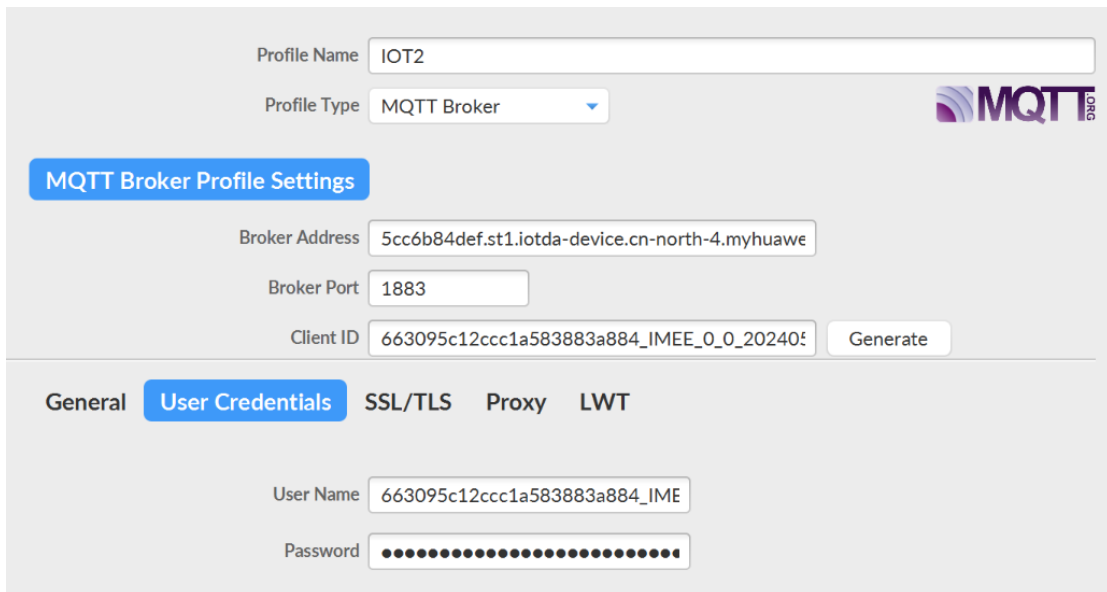


The screenshot shows the 'MQTT Broker Profile Settings' interface. At the top, 'Profile Name' is 'IOT\_HUAWEI' and 'Profile Type' is 'MQTT Broker'. Below this is a blue button labeled 'MQTT Broker Profile Settings'. The main settings section includes 'Broker Address' (5cc6b84def.st1.iotda-device.cn-north-4.myhuawe), 'Broker Port' (1883), and 'Client ID' (663095c12ccc1a583883a884\_IMEI\_0\_0\_202404) with a 'Generate' button. At the bottom, there are tabs for 'General', 'User Credentials', 'SSL/TLS', 'Proxy', and 'LWT'. The 'User Credentials' tab is active, showing 'User Name' (663095c12ccc1a583883a884\_IME) and a masked 'Password' field.

### 3.11.2 设备信息 2

字段	值
Profile Type	MQTT Broker
Broker Address	5cc6b84def.st1.iotda-device.cn-north-4.myhuaweicloud.com
Broker Port	1883

Client ID	663095c12ccc1a583883a884_IMEE_0_0_2024062509
User Name	663095c12ccc1a583883a884_IMEE
Password	c5b89be910801d7e4b7c07c05d7fa0d8336e4b8119dcdcb6b4e06f2f2b6bee6c
Push Address	\$oc/devices/663095c12ccc1a583883a884_IMEE/sys/properties/report



The screenshot shows the MQTT Broker Profile Settings interface. It includes fields for Profile Name (IOT2), Profile Type (MQTT Broker), Broker Address (5cc6b84def.st1.iotda-device.cn-north-4.myhuawe), Broker Port (1883), and Client ID (663095c12ccc1a583883a884\_IMEE\_0\_0\_2024062509). There is a 'Generate' button next to the Client ID field. Below these fields are tabs for General, User Credentials, SSL/TLS, Proxy, and LWT. The User Credentials tab is active, showing fields for User Name (663095c12ccc1a583883a884\_IMEE) and Password (masked with dots).

### 3.11.3 服务-智能风扇

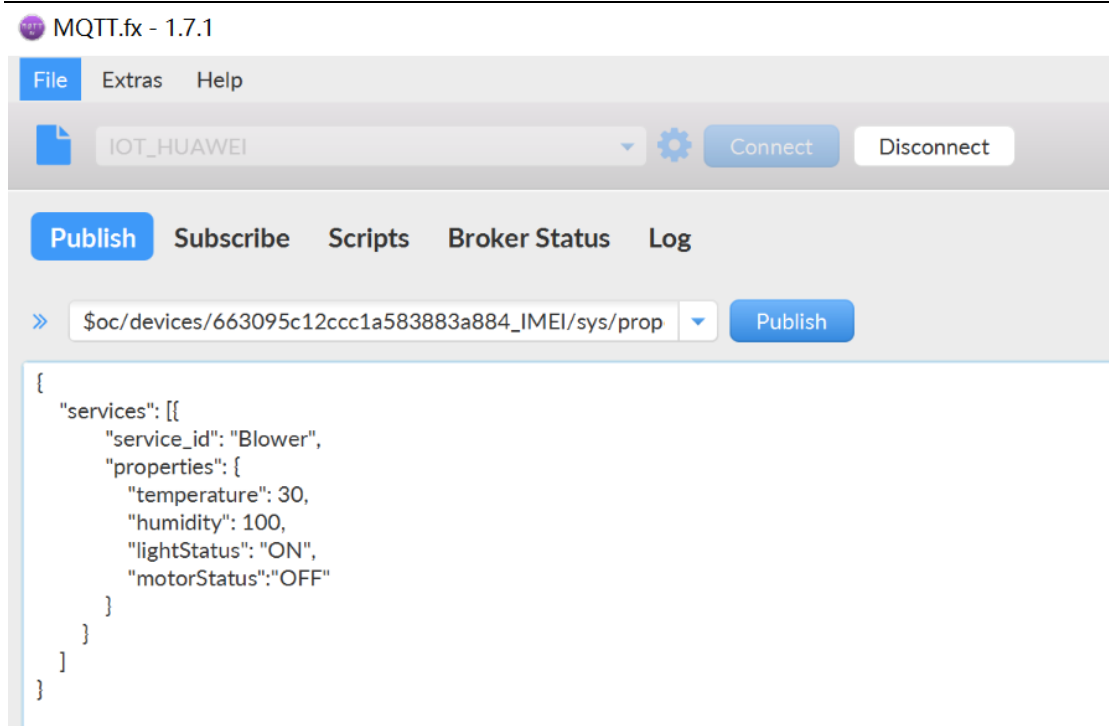
设备上报属性数据: \$oc/devices/ {device\_id}/sys/properties/report

字段	名称	备注
services	业务列表	
service_id	业务编码	Blower
properties	属性列表	
temperature	温度	



humidity	湿度	
lightStatus	灯光状态	ON:开 OFF:关
motorStatus	发动机状态	ON:开 OFF:关

```
{
  "services": [{
    "service_id": "Blower",
    "properties": {
      "temperature": 30,
      "humidity": 100,
      "lightStatus": "ON",
      "motorStatus": "OFF"
    }
  ]
}
```



### 3.11.4 服务-PM2.5 检测器

设备上报属性数据: \$oc/devices/ {device\_id}/sys/properties/report

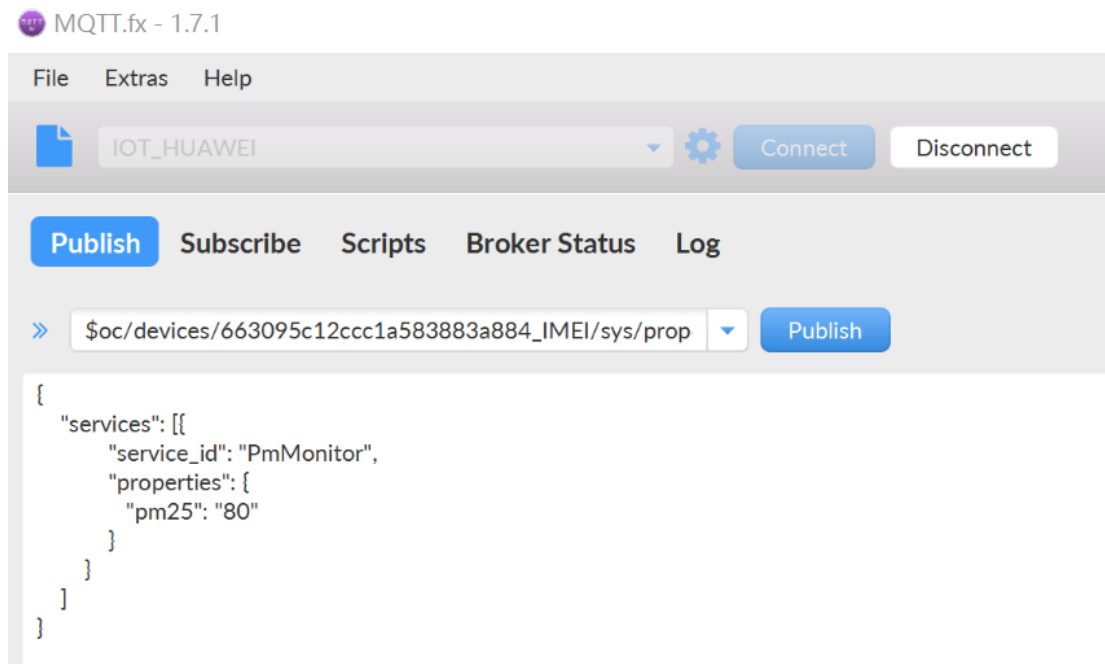
字段	名称	备注
services	业务列表	
service_id	业务编码	PmMonitor
properties	属性列表	
pm25	PM2.5 值	

```
{
  "services": [{
    "service_id": "PmMonitor",
    "properties": {
      "pm25": "80"
    }
  ]
}
```

```

    }
  }
]
}

```



### 3.11.5 服务-LED 灯

设备上报属性数据: \$oc/devices/ {device\_id}/sys/properties/report

字段	名称	备注
services	业务列表	
service_id	业务编码	LedLight
properties	属性列表	
luminance	亮度	

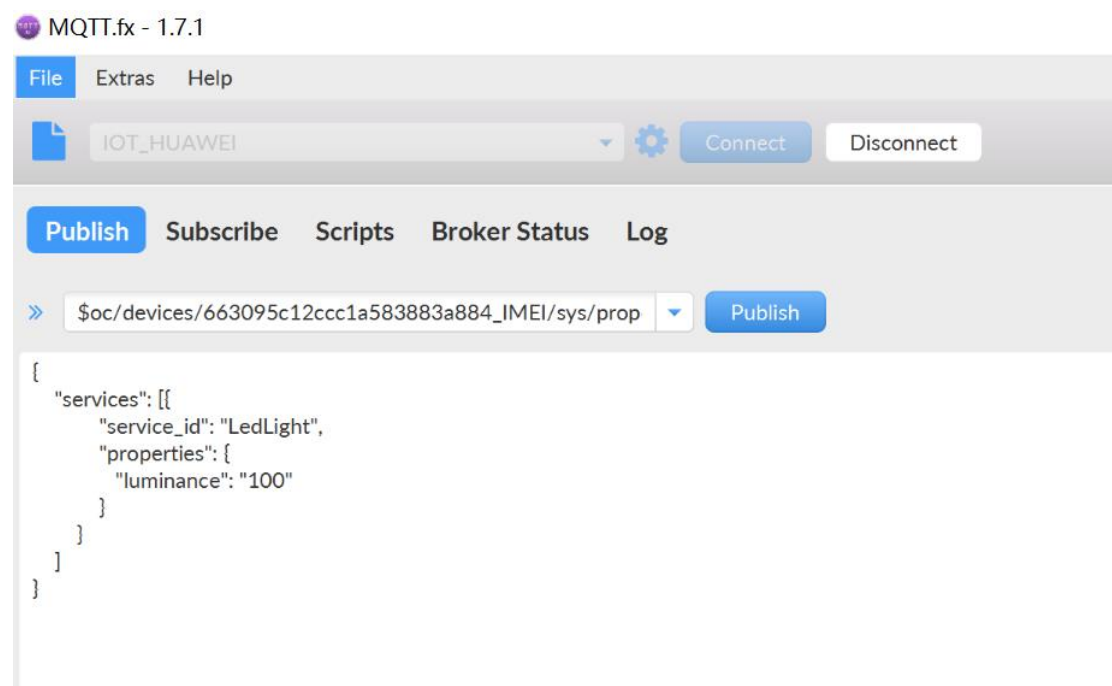
```

{

```



```
{
  "services": [{
    "service_id": "LedLight",
    "properties": {
      "luminance": "100"
    }
  ]
}
```



## 3.12 AMQP 消息推送(webSocket)

### 3.12.1 消息地址

```
/api/ amqpSocketServer/{deviceId}
```

### 3.12.2 请求消息

消息类型	字段	名称	备注
------	----	----	----

请求接口	deviceId	设备编码	
------	----------	------	--

### 3.12.3 响应消息-智能风扇

消息类型	字段	名称	备注
响应接口	deviceId	设备编码	
	serviceId	业务编码	Blower
	temperature	温度	
	humidity	湿度	
	lightStatus	灯光状态	ON:开 OFF:关
	motorStatus	发动机状态	ON:开 OFF:关

```
{
  "deviceId": "663095c12ccc1a583883a884_IMEI",
  "humidity": 91,
  "lightStatus": "OFF",
  "motorStatus": "OFF",
  "serviceId": "Blower",
  "temperature": 31
}
```



### 3.12.4 响应消息-PM2.5 检测器

消息类型	字段	名称	备注
响应接口	deviceId	设备编码	
	serviceId	业务编码	PmMonitor
	pm25	PM2.5 值	

```
{
  "deviceId": "663095c12ccc1a583883a884_IMEI",
  "serviceId": "PmMonitor",
  "pm25": 80
}
```

### 3.12.5 响应消息-LED 灯

消息类型	字段	名称	备注
响应接口	deviceId	设备编码	
	serviceId	业务编码	LedLight
	luminance	亮度	

```
{
  "deviceId": "663095c12ccc1a583883a884_IMEI",
  "serviceId": "LedLight",
  "luminance": 100
}
```

### 3.13 华为云鉴权

#### 3.13.1 消息地址

```
https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
```

#### 3.13.2 消息类型

'Content-Type': 'application/json'

#### 3.13.3 消息方法

http.RequestMethod.POST

#### 3.13.4 请求消息

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "test1",
          "password": "aA123456",
          "domain": {
            "name": "xinfenghua"
          }
        }
      }
    },
    "scope": {
```



```
"project": {  
  "name": "cn-north-4"  
}  
}  
}  
}
```

消息类型	字段	名称	备注
请求接口	password. user. name	IAM 用户名	我的凭证
	password. user. domain.name	账号名	
	scope. project. name	项目列表中-> 项目	

### 3.13.5 响应消息

消息类型	字段	名称	备注
Header 参数	X-Subject-Token	签名后的 Token	小于 32KB

```
this.token = data.header["x-subject-token"]
```

## 3.14 查询设备

### 3.14.1 消息地址

```
/v5/iot/{project_id}/devices/{device_id}
```

标准版

https://5cc6b84def.st1.iotda-app.cn-north-4.myhuaweicloud.com/v5/iot/08f852bf338026152f1bc00c0c05cc50/devices/663095c12ccc1a583883a884\_IMEI

基础版

https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/08f852bf338026152f1bc00c0c05cc50/devices/66bb15dbd23943204ae1a4cb\_IMEE

### 3.14.2 消息类型

'Content-Type': 'application/json'

### 3.14.3 消息方法

http.RequestMethod.GET

### 3.14.4 请求消息

消息类型	字段	名称	备注
路径参数	object_id	项目 ID	我的凭证->项目列表->项目 ID
	device_id	设备 ID	IOTDA->设备->所有设备 设备详情->设备 ID
Header 参数	X-Auth-Token	Tocken 码	鉴权消息-> X-Subject-Token
	Instance-Id	实例 ID	IOTDA->总览 详情->ID

---

```
header: {
  'Content-Type': 'application/json',
  'X-Auth-Token': this.token,
  'Instance-Id': '76a73dd8-7620-49c2-85d2-9d4fa9eced5f'
}
```

### 3.14.5 响应消息

```
{
  "app_id": "jeQDJQZltU8iKgFFoW060F5SGZka",
  "app_name": "testAPP01",
  "device_id": "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "node_id": "ABC123456789",
  "gateway_id": "d4922d8a-6c8e-4396-852c-164aefa6638f",
  "device_name": "dianadevice",
  "node_type": "ENDPOINT",
  "description": "watermeter device",
  "fw_version": "1.1.0",
  "sw_version": "1.1.0",
  "auth_info": {
    "auth_type": "SECRET",
    "secret": "3b935a250c50dc2c6d481d048cefdc3c",
    "fingerprint": "dc0f1016f495157344ac5f1296335cff725ef22f",
    "secure_access": true,
    "timeout": 0
  }
}
```

### 3.15 查询设备属性-同步消息

注意:

在请求消息时, IOTDA 将向设备发送 mqtt 命令: 平台查询设备属性  
在 IOTD 收到设备响应的命令: 属性查询响应结果, 才会响应此接口



### 3.15.1 消息地址

```
/v5/iot/{project_id}/devices/{device_id}/properties?service_id=${service_id }
```

标准版

https://5cc6b84def.st1.iotda-app.cn-north-4.myhuaweicloud.com/v5/iot/08f852bf338026152f1bc00c0c05cc50/devices/663095c12ccc1a583883a884\_IMEI?service\_id=Blower

基础版

https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/08f852bf338026152f1bc00c0c05cc50/devices/66bb15dbd23943204ae1a4cb\_IMEE?service\_id=Blower

### 3.15.2 消息类型

'Content-Type': 'application/json'

### 3.15.3 消息方法

http.RequestMethod.GET

### 3.15.4 请求消息

消息类型	字段	名称	备注
路径参数	object_id	项目 ID	我的凭证->项目列表->项目 ID
	device_id	设备 ID	IOTDA->设备->所有设备 设备详情->设备 ID
Header 参数	X-Auth-Token	Token 码	鉴权消息-> X-Subject-Token





	Instance-Id	实例 ID	IOTDA->总览 详情->ID
Body 参数	service_id	产品 ID	IOTDA->产品 详情-> 服务 ID

```
header: {  
  'Content-Type': 'application/json',  
  'X-Auth-Token': this.token,  
  'Instance-Id': '76a73dd8-7620-49c2-85d2-9d4fa9eced5f'  
}
```

### 3.15.5 响应消息

```
{  
  "request_id" : "b1224afb-e9f0-4916-8220-b6bab568e888",  
  "response" : {  
    "services" : {  
      "serviceId" : "Temperature",  
      "properties" : {  
        "PhV_phsA" : "1",  
        "PhV_phsB" : "2"  
      },  
      "eventTime" : "20190606T121212Z"  
    },  
  },  
}
```

### 3.15.6 平台查询设备属性

\$oc/devices/{device\_id}/sys/properties/get/response/request\_id={request\_id}

消息类型	字段	名称	备注
路径参数	device_id	设备 ID	IOTDA->设备->所有设备 设备详情->设备 ID
	request_id	请求 id	IOTDA 平台自动产生

\$oc/devices/663095c12ccc1a583883a884\_IMEI/sys/properties/get/request\_id=271974e9-1a0d-4257-be76-0e7dea15ad9c

### 3.15.7 属性查询响应结果

\$oc/devices/{device\_id}/sys/properties/get/response/request\_id={request\_id}

消息类型	字段	名称	备注
路径参数	device_id	设备 ID	IOTDA->设备->所有设备 设备详情->设备 ID
	request_id	请求 id	将请求 id 原样返回

\$oc/devices/663095c12ccc1a583883a884\_IMEI/sys/properties/get/response/request\_id=271974e9-1a0d-4257-be76-0e7dea15ad9c