

天正高科 API 接口文档

北京天正高科智能科技有限公司

修订记录

时间	版本	备注
2018-04-21	V1.0	初版
2018-05-30	V1.1	添加雨感传感器关联、视频、数据统计
2018-08-30	V1.2	文档功能简化
2018-12-24	V.1.3	删除简化部分接口
2019-08-21	V1.4	更新设备指令
2020-01-13	V1.5	添加 LED 接口
2020-06-10	V1.6	更新设备指令
2020-07-22	V1.7	添加数据统计接口

目录

1. 用户登录.....	3
2. 获取网关列表.....	4
3. 查询网关状态.....	5
4. 获取所有设备列表.....	5
5. 获取用户的设备分类.....	6
6. 依据类别 ID 获取设备列表.....	7
7. 获取设备预警消息列表.....	7
8. 获取用户信息.....	8
9. 获取某一设备的历史记录（数据统计）.....	9
10. 获取某个设备的信息.....	10
11. 用户退出.....	11
12. WebScoket 连接、发送消息与消息推送.....	11
13. 控制指令和设备指令说明.....	13
14. 设备数据及单位.....	17
15. 获取 LED 显示屏列表.....	19
16. 获取 LED 一个字符分区的设备列表.....	20
17. 字符分区添加设备.....	21
18. 字符分区删除设备.....	21
19. 控制 LED 显示屏开关.....	22
20. 管理平台--六条数据统计接口.....	22

天正高科 API 接口说明

地址: <http://www.tscsmart.com>

1. 用户登录

用户登陆: /API/user/login

请求方式: GET

参数说明:

名称	类型	是否必须	描述
name	String	是	用户名
pwd	String	是	密码

返回参数说明:

名称	类型	描述
返回参数		失败: {"statu":0,"id":null,"msg":"登陆失败!!!","list":[]} 成功: {"statu":1,"id":36,"msg":"de2e53a7-2621-47bf-aa03-7d8290f392ef","list":[]}
statu	int	登录状态
id	int	uid,须保存
msg	String	tokey,须保存下来,一天会更新一次

注意: 登陆后, 所有请求的接口必须带上下边两个参数; 请求方式都为"GET".

名称	类型	描述
uid	int	/user/login 返回的用户 ID
tokey	String	/user/login 返回的 msg

http 请求示例:

```
$.ajax({
  url:'http://www.tscsmart.com/API/App/****',
  type:'GET',      //GET\POST
  async:true,     //或 false,是否异步
  data:{uid:uid,tokey:tokey},
  timeout:5000,   //超时时间
  dataType:'jsonp', //返回的数据格式: json/xml/html/script/jsonp/text
  jsonp: 'callback',
  jsonpCallback:"success_jsonp",
  success:function(data){
    //返回数据 进行处理
  }
})
```

```

    },
    error:function(xhr,textStatus){
        console.log('错误');
        console.log(xhr);
        console.log(textStatus);
    }
});

```

2. 获取网关列表

接口名: /API/App/gateway

接口说明: 获取用户下的网关列表

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg

返回结果示例:

```

[{"gmac":"01-00-00-00-00-00-00-00-04",
  "gname":"网关 1#",
  "remarks":"备注",

  "uid":36},

{"gmac":"01-00-00-00-00-00-00-00-05",
  "gname":"网关 2#",
  "remarks":"备注",

  "uid":36},

.....]

```

返回参数说明:

名称	类型	描述
gmac	String	网关 mac 地址
gname	String	网关名称
remarks	String	网关备注
uid	int	网关 mac 地址

3. 查询网关状态

接口名: /API/control/Online

接口说明: 查询网关状态, 判断是否正常

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg
gmac	String	是	网关 mac 地址

返回结果示例:

网关断开: {"statu":0,"id":null,"msg":"网关 00-00-00-00-00-00-00-00-04 断线!",
"list":[]}

连接正常: {"statu":1,"id":null,"msg":"网关 00-00-00-00-00-00-00-00-04 正常!",
"list":[]}

返回参数说明:

名称	类型	描述
statu	String	网关状态
msg	String	网关状态信息

4. 获取所有设备列表

接口名: /API/App/checkEquipment

接口说明: 获取用户下所有设备列表

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg

返回结果示例:

```
[{"emac":"01-00-00-00-00-00-00-00-04",
  "ename":"1#电磁阀",
  "eqmsg":"0000",
  "gmac":"00-00-00-00-00-00-00-00-05",
  "eqtypename":"无线电磁阀",
  "eid":6},
{"emac":"01-00-00-00-00-00-00-00-05",
  "ename":"1#湿度",
```

```

    "eqmsg":"0.2356",
    "gmac":"00-00-00-00-00-00-00-00-05",
    "eqtypename":"湿度传感器",
    "eid":3},
    {"emac":"01-00-00-00-00-00-00-00-06",
    "ename":"双控电磁阀",
    "eqmsg":"1000,1101",
    "gmac":"00-00-00-00-00-00-00-00-05",
    "eqtypename":"双控电磁阀",
    "eid":9}
    .....]

```

返回参数说明:

名称	类型	描述
emac	String	设备 mac 地址
ename	String	设备名称
eqmsg	String	设备的开关状态或采集数值
gmac	String	网关 mac 地址
eqtypename	String	设备类型名称
eid	Int	设备类型 ID

5. 获取用户的设备分类

接口名: /API/App/eqtype

接口说明: 获取用户设备的所有类别及类别下的设备数量

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg

返回结果示例:

```

[{"eid":6,"eqtypename":"无线电磁阀","num":3},
 {"eid":3,"eqtypename":"湿度传感器","num":2},
 {"eid":8,"eqtypename":"水泵","num":1},
 .....]

```

返回参数说明:

名称	类型	描述
eid	int	设备类型 ID
eqtypename	String	设备类型名称

num	int	在用户中，设备类型的设备数量
-----	-----	----------------

6. 依据类别 ID 获取设备列表

接口名: /API/App/eqlist

接口说明: 获取设备的所有类别

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg
eid	int	是	设备类型 ID

返回结果示例:

```
[{"emac": "01-00-00-00-00-00-00-00-04",
  "ename": "1#电磁阀",
  "eqmsg": "0000",
  "gmac": "00-00-00-00-00-00-00-00-05",
  "eqtypename": "无线电磁阀",
  "eid": 6},
 {"emac": "01-00-00-00-00-00-00-00-05",
  "ename": "2#电磁阀",
  "eqmsg": "0001",
  "gmac": "00-00-00-00-00-00-00-00-05",
  "eqtypename": "无线电磁阀",
  "eid": 6},
  ....]
```

返回参数说明:

名称	类型	描述
emac	String	设备 mac 地址
ename	String	设备名称
eqmsg	String	设备的开关状态或采集数值
gmac	String	网关 mac 地址
eqtypename	String	设备类型名称
eid	Int	设备类型 ID

7. 获取设备预警消息列表

接口名: /API/App/notice

接口说明: 获取设备预警消息列表

请求方式：GET

参数说明：

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg

返回结果示例：

```
[{"emac": "01-00-00-00-00-00-00-00-04",
  "ename": "1#电磁阀",
  "msg": "00",
  "create_date_format": "2018-03-23 12:23:23",
  "htype": 2},
 {"emac": "01-00-00-00-00-00-00-00-05",
  "ename": "2#电磁阀",
  "msg": "00",
  "create_date_format": "2018-03-23 12:23:23",
  "htype": 1},
 .....]
```

返回参数说明：

名称	类型	描述
emac	String	设备 mac 地址
ename	String	设备名称
msg	String	预警消息内容
create_date_format	String	预警消息创建时间
htype	int	预警消息状态：0-发现异常，1-正在处理，2-处理完成

8. 获取用户信息

接口名：/API/App/userInfo

接口说明：获取区域下的设备列表

请求方式：GET

参数说明：

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg

返回结果示例：

```
{
  "uid":12,
  "name":"test",
  "pwd":"123456",
  "alias":"测试用户",
  "phone":"13762876749",
  "address":"北京市海淀区新东源大厦",
  "company":"天正高科"}

```

返回参数说明:

名称	类型	描述
uid	int	用户 ID
name	String	用户名
pwd	String	密码
alias	String	别名
phone	String	手机号
address	String	地址
company	String	项目名称

9. 获取某一设备的历史记录 (数据统计)

接口名: /API/App/eqls

接口说明: 获取用户下某一个设备的历史数据记录

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg
bigdate	date	是	查询开始时间
enDate	date	是	查询结束时间
emac	String	是	设备 Mac

返回结果示例:

```
[{"create_date":1527665036,
  "val":"0001"},
 {"create_date":1527666333,
  "val":"0000"},
 {"create_date":1527668673,
  "val":"0001"},
 .....]
```

返回参数说明:

名称	类型	描述
create_date	date	数据创建时间 (毫秒)
val	String	设备数值

10. 获取某个设备的信息

接口名: /API/App/eqinfo

接口说明: 获取用户下某一个设备的信息

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg
emac	String	是	设备 Mac

返回结果示例:

```
{
  "emac": "01-00-00-00-00-00-00-00-04",
  "ename": "1#电磁阀",
  "eqmsg": "0000",
  "gmac": "00-00-00-00-00-00-00-00-05",
  "eqtypename": "无线电磁阀",
  "eid": 6,
  "sort": 11,
  "collection": null
}
```

返回参数说明:

名称	类型	描述
emac	String	设备 mac 地址
ename	String	设备名称
eqmsg	String	设备的开关状态或采集数值
gmac	String	网关 mac 地址
eqtypename	String	设备类型名称
eid	Int	设备类型 ID
sort	Int	设备排序号
collection	Int	传感器采集间隔 (S)

11. 用户退出

用户登陆: /API/user/quit

接口说明: 用户退出

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg

返回结果示例:

成功: 1

失败: null

12. WebScoket 连接、发送消息与消息推送

WebScoket 链接: /API/echo/uid(用户 ID)/tokey

接口说明: 连接 webscoket 服务 (接收推送消息)

注: 1.重连机制 (重新登录 (/API/user/login) => 得到 (id ,tokey) => 重新发送 websocket 连接) 2.断开查询做心跳检验

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg

发送消息接口: /API/control/offon

接口说明: 向服务器发送控制设备指令

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg
EMac	String	是	设备 mac
GMac	String	是	网关 mac
Instructions	String	是	控制指令
Msg	String	是	发送消息指令 (设备指令)

返回结果示例:

```
{"emac": "01-00-00-00-00-00-00-04", "msg": "0001", "dictate": "59"}
```

或

```
{"quit": "有其他人上线, 你将无法收到设备消息"}
```

返回参数说明（消息推送）：

名称	类型	描述
emac	String	设备 mac 地址
msg	String	接收设备的开关状态或采集数值
dictate	String	接收指令
quit	String	将用户踢下线提示

webscoket 前端连接示例:

```

var ws;
var lockReconnect = false; //避免重复连接
var wsUrl = 'ws://www.tscsmart.com/API/echo/uid(用户 id)/tokey(tokey 值)';
//创建连接
function createWebSocket(url) {
    try {
        ws = new WebSocket(url);
        initEventHandle();
    } catch(e) {
        reconnect(url);
    }
}
function initEventHandle() {
    ws.onclose = function() {
        console.log('websocket 已经关闭' + new Date());
        reconnect(wsUrl);
    };
    ws.onerror = function() {
        console.log('错误: ' + event.data);
        reconnect(wsUrl);
    };
    ws.onopen = function() {
        console.log('websocket 已经连接' + new Date());
        //心跳检测重置
        heartCheck.reset().start();
    };
    ws.onmessage = function(event) {
        //如果获取到消息，心跳检测重置，拿到任何消息都说明当前连接是正常的
        heartCheck.reset().start();
        onMessage(event);
    }
}
//重连
function reconnect(url) {
    if(lockReconnect) return;
    lockReconnect = true;

```

```

//没连接上会一直重连, 设置延迟避免请求过多
setTimeout(function() {
    createWebSocket(url);
    lockReconnect = false;
}, 2000);
}
//心跳检测
var heartCheck = {
    timeout: 60000 * 3, //3 分钟
    timeoutObj: null,
    serverTimeoutObj: null,
    reset: function() {
        clearTimeout(this.timeoutObj);
        clearTimeout(this.serverTimeoutObj);
        return this;
    },
    start: function() {
        var self = this;
        this.timeoutObj = setTimeout(function() {
            //这里发送一个心跳, 后端收到后, 返回一个心跳消息,
            //onmessage 拿到返回的心跳就说明连接正常
            ws.send("HeartBeat");
            self.serverTimeoutObj = setTimeout(function() { //如果超过一定时间还没重
置, 说明后端主动断开了
                ws.close(); //如果 onclose 会执行 reconnect, 我们执行 ws.close()就行了.
            }, self.timeout)
        }, this.timeout)
    }
}
//接收消息
function onMessage(event) {
    //消息信息处理在这里编写
    if(event.data == "") {
        return false;
    }
    var json = JSON.parse(event.data);
}

```

具体控制指令和消息指令, 请看下设备指令说明

13. 控制指令和设备指令说明

13.1 控制指令

Instructions (控制指令) Msg (发送消息指令即设备指令)

dictate (接收指令) msg (接收设备消息指令)

一、设备控制指令和接收指令

设备类型	控制指令	发送设备指令	接收指令	接收设备指令
无线电磁阀	58	0001 或 0000	59	0001 或 0000
双控电磁阀	58	1001、1000、1100、1101	59	1001、1000、1100、1101
水泵	58	0001 或 0000	59	0001 或 0000
双控水泵	58	1001、1000、1100、1101	59	1001、1000、1100、1101
无线电动阀	58	00 + 03/05/01/00	60 (先)	195a (看注释)
			59 (后)	00 + 03/05/01/00
双控电动阀	58	10/11 + 03/05/01/00	60 (先)	1931 (看注释)
			59 (后)	10/11 + 03/05/01/00
三通电动阀	58	2001 (左开) /2004 (右开) /2000 (关)	60 (先)	114a (看注释)
			59 (后)	2001/2004/2000
压力反馈电磁阀	58	0001/0000	59	0001/0000
			76	压力值 (主动推送)
采集器设备			59	采集值

发送设备指令和接收设备指令相同

注释:

无线电动阀、双控电动阀接收指令 ‘60’ (为电动阀操作时长) :

前 2 位 表示设备开启所需时长, 单位秒 (十六进制)

第 3 位 表示开启度数 3 (30°) /5 (60°) /1 (90°) /0 (关闭)

第 4 位 表示设备区分 a (无线电动阀) /0 (双控电动阀 1 路) /1 (双控电动阀 2 路)

三通电动阀接收指令 ‘60’ (为三通电动阀操作时长) :

前 2 位 表示设备开启所需时长, 单位秒 (十六进制)

第 3 位 表示开启阀门 1 (左阀) /4 (右阀) /0 (关闭)

第 4 位 表示设备区分 a (三通电动阀)

二、部分功能指令

其他功能	控制指令	设备指令	接收指令	接收设备指令
设备组网	17	4G 多通道 (见注释)	18	无
删除组网	19	无	20	无
查看设备信息	0	无	1(设备)、5(网关)	请看注释
水肥机	36	请看设备指令介绍	37	请看设备指令介绍

注释:

设备组网:

4G 多通道添加设备指令----00/01+类型 ID 即 eid (均为十六进制)

00-----控制类设备 01-----采集类设备

设备信息:

(1) 设备返回:

① 4G 单通道网关下: 设备 mac、网关 mac、频点、型号、固件版本

② 4G 多通道网关下: 设备 mac、网关 mac、频点、信号强度、网关通道、电量强度、固件版本、设备温度

(2) 网关返回:

- ① 4G 单通道网关: 网关 mac、频点、型号、固件版本 1、固件版本 2
- ② 4G 多通道网关: 网关 mac、频点 1、频点 2、频点 3、频点 4、型号、固件版本

13.2 设备指令

1. 无线电磁阀、水泵指令

通道 指令

00 00/01 关闭/开启

2. 双控电磁阀、双控水泵指令

第一通道:

通道 指令

10 00/01 关闭/开启 (第一路)

第二通道:

通道 指令

11 00/01 关闭/开启 (第二路)

3. 无线电动阀指令

通道 指令

00 00/03/05/01 关闭/开启 30° /开启 60° /开启 90°

4. 双控电动阀指令

第一通道:

通道 指令

10 00/03/05/01 关闭/开启 30° /开启 60° /开启 90° (第一路)

第二通道:

通道 指令

11 00/03/05/01 关闭/开启 30° /开启 60° /开启 90° (第二路)

5. 三通电动阀指令

通道 指令

20 00/01/04 关闭/开启左阀/开启右阀

6. 压力反馈电磁阀

通道 指令

00 00/01 关闭/开启

电磁阀详细指令说明:

命令编码 (16 进制 (10 进制))	通信类型	命令说明	返回命令 (16 进制 (10 进制))
-----------------------------	------	------	----------------------------

<p>3A (58)</p>	<p>下行</p>	<p>控制单路电磁阀开关或采集器采集的指令，如果该设备硬件是控制器则控制器电磁阀开关，如果该设备硬件是采集器则执行采集或校准，</p> <p>00 00: 控制单路电磁阀关闭，不适用于采集器；</p> <p>00 01: 控制单路电磁阀开启，或者使采集器执行采集；湿度采集值小数点后保留 4 位小数，温度采集值小数点后保留 2 位小数；</p> <p>10 00: 控制双路电磁阀第一路关闭；</p> <p>10 01: 控制双路电磁阀第一路打开；</p> <p>11 00: 控制双路电磁阀第二路关闭；</p> <p>11 01: 控制双路电磁阀第二路打开；</p> <p>FF 00: 控制双路电磁阀两路全部关闭；</p> <p>FF 01: 控制双路电磁阀两路全部打开；</p> <p>控制电动阀的开启或关闭，开启包括开 30 度、开 60 度、开 90 度（全开）三种状态，不同角度控制水流量的大小；</p> <p>00 00: 控制单路电动阀关闭；</p> <p>00 01: 控制单路电动阀打开；</p> <p>00 03: 控制单路电动阀打开 30 度；</p> <p>00 05: 控制单路电动阀打开 60 度；</p> <p>10 00: 控制双路电动阀第一路关闭；</p> <p>10 01: 控制双路电动阀第一路打开全开状态；</p> <p>10 03: 控制双路电动阀第一路打开 30 度；</p> <p>10 05: 控制双路电动阀第一路打开 60 度；</p> <p>11 00: 控制双路电动阀第二路关闭；</p> <p>11 01: 控制双路电动阀第二路打开全开状态；</p> <p>11 03: 控制双路电动阀第二路打开 30 度；</p> <p>11 05: 控制双路电动阀第二路打开 60 度；</p> <p>FF 00: 控制双路电动阀两路全部关闭；</p> <p>FF 01: 控制双路电动阀两路全部打开全开状态；</p> <p>FF 03: 控制双路电动阀两路全部打开 30 度；</p> <p>FF 05: 控制双路电动阀两路全部打开 60 度；</p>	<p>3B (59)</p>
<p>3B (59)</p>	<p>上行</p>	<p>终端设备设置电磁阀或电动阀或采集完成后的响应；</p> <p>电磁阀或电动阀的返回指令内容与发送指令内容一致；</p>	<p>3A (58)</p>
<p>3C (60)</p>	<p>上行</p>	<p>适用于节点为电动阀的情况。电动阀操作时间长，收到 3A 指令后第一次返回 3C (60)，操作完成后第二次返回 3B (59)，</p> <p>前 2 位 表示设备开启所需时长，单位秒（十六进制）</p> <p>第 3 位 表示开启度数 3 (30°) /5 (60°) /1 (90°) /0 (关闭)</p> <p>第 4 位 表示设备区分 a (无线电动阀) /0 (双控电动阀 1 路) /1 (双控电动阀 2 路)</p> <p>如本次操作是双控电动阀一路，开启 30 度，需要时间为 50s，</p>	<p>3A (58)</p>

		接收命令为: 3230	
24 (36)	下行	<p>水肥一体机指令: 前 4 位 表示设备编号 0001 后 2 位 01 (打开)、00 (关闭)</p> <p>000101: 自动开启 000100: 自动关闭 000201: 注肥泵 000200: 注肥泵关闭 000301: 搅拌泵 000300: 搅拌泵关闭 000401: 搅拌泵 000400: 搅拌泵关闭 000501: 搅拌泵 000500: 搅拌泵关闭</p>	25 (37)
25 (37)	上行	<p>水肥一体机返回指令: 前 4 位 表示设备编号 后 2 位或 3 位表示设备状态 01 (打开) 00 (关闭) 或采集值</p> <p>000101: 自动开启 000100: 自动关闭 000201: 注肥泵开启 000200: 注肥泵关闭 000301: 搅拌泵开启 000300: 搅拌泵关闭 000401: 搅拌泵开启 000400: 搅拌泵关闭 000501: 搅拌泵开启 000500: 搅拌泵关闭 00061a: ph 值 0007107: ec 值</p> <p>注意: ec 和 ph 后两位或三位是它们的采集值, 转换成十进制后需除 10。例如: 107 (十六进制) →263 (十进制) /10=26.3</p>	24 (36)

14. 设备数据及单位

表一: 设备数据

Eid	设备类型	设备数据
1	水表	水表
2	气象站	风速,风向,温度,湿度,气压,雨量,太阳辐射,蒸发量,光照
3	湿度传感器	湿度
4	温湿盐传感器	湿度,温度,盐分
5	温湿度传感器	湿度,温度
6	无线电磁阀	开关状态
7	压力传感器	压力
8	水泵	开关状态
9	双控电磁阀	一路开关状态,二路开关状态

10	四路湿度传感器	湿度,湿度,湿度,湿度
11	四路温湿度传感器	湿度,温度,湿度,温度,湿度,温度,湿度,温度
12	四路温湿盐传感器	湿度,温度,盐分,湿度,温度,盐分,湿度,温度,盐分,湿度,温度,盐分
13	雨感传感器	状态
14	二路湿度传感器	湿度,湿度
15	空气温湿度传感器	湿度,温度
17	双控水泵	一路开关状态,二路开关状态
18	无线电动阀	开关状态
20	水质传感器	温度,盐度,PH,ORP,溶解氧,NH4,浊度
21	双控电动阀	一路开关状态,二路开关状态
24	液位传感器	液位
25	二路温湿度传感器	湿度,温度,湿度,温度
26	pH 传感器	pH
27	二路温湿盐传感器	湿度,温度,盐分,湿度,温度,盐分
28	智能水肥机	
30	三路湿度传感器	湿度,湿度,湿度
31	土壤氮磷钾传感器	氮,磷,钾
32	六路湿度传感器	湿度,湿度,湿度,湿度,湿度,湿度
33	六路温湿度传感器	湿度,温度,湿度,温度,湿度,温度,湿度,温度,湿度,温度,湿度,温度
38	三通电动阀	开关状态
40	压力反馈电磁阀	开关状态,压力
41	倾角传感器	X 轴角度,Y 轴角度

表二：数据单位

数据类型	单位
湿度	%
温度	°C
盐分	uS/cm
水表	m ³
压力	Mpa
液位	m
pH	无
风速	m/s
风向	°
气压	hPa
雨量	mm
太阳辐射	W/m ²
蒸发量	mm
光照	Klux
ORP	mV

NH4	mg/L
浊度	NTU
溶解氧	Mg/L
盐度 (水质)	S/m
氮	mg/kg
磷	mg/kg
钾	mg/kg
倾角	°

15. 获取 LED 显示屏列表

接口名: /API/led/getLedList

接口说明: 获取用户下的 LED 列表

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg

返回结果示例:

```
[{"ledmac": "00-00-00-01",
  "ledname": "LED 显示 1",
  "uid": 5,
  "msg": "0001",
  "regnum": 2,
  "showtime": 6,
  "isshowname": 3,
  "isvalid": 1,
  .....
}, {"ledmac": "00-00-00-02",
  "ledname": "LED 显示 2",
  "uid": 5,
  "msg": "0001",
  "regnum": 2,
  "showtime": 6,
  "isshowname": 3,
  "isvalid": 1,
  .....
},.....]
```

返回参数说明:

名称	类型	描述
ledmac	String	LED 设备 mac 地址

ledname	String	LED 设备名称
uid	int	用户 id
msg	String	LED 设备状态, 0000 关, 0001 开
regnum	Int	LED 字符分区数量
showtime	Int	显示时间 (每一条数据需要显示的时间) 分钟
isshowname	Int	设备显示格式 (1.设备类型+数据, 2.设备名称+数据+类型, 3.设备数据)
isvalid	Int	是否有效

16. 获取 LED 一个字符分区的设备列表

接口名: /API/led/getLedEq

接口说明: 获取 LED 字符分区设备列表

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg
ledmac	String	是	LEDmac
regid	int	是	字符分区编号

返回结果示例:

```
[{"emac": "01-00-00-00-00-00-00-00-04",
  "ename": "湿度 1 号",
  "eqmsg": "0.2314",
  "gmac": "00-00-00-00-00-00-00-00-05",
  "eqtypename": "湿度传感器",
  "eid": 3},
 {"emac": "01-00-00-00-00-00-00-00-05",
  "ename": "1#温湿度",
  "eqmsg": "0.2356, 25.61",
  "gmac": "00-00-00-00-00-00-00-00-05",
  "eqtypename": "温湿度传感器",
  "eid": 5},
 {"emac": "01-00-00-00-00-00-00-00-06",
  "ename": "湿度 2 号",
  "eqmsg": "0.2268",
  "gmac": "00-00-00-00-00-00-00-00-05",
  "eqtypename": "湿度传感器",
  "eid": 3}
.....]
```

返回参数说明:

名称	类型	描述
emac	String	设备 mac 地址
ename	String	设备名称
eqmsg	String	设备采集数值
gmac	String	网关 mac 地址
eqtypename	String	设备类型名称
eid	Int	设备类型 ID

17. 字符分区添加设备

接口名: /API/led/addOneLedEq

接口说明: LED 字符分区添加设备

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg
ledmac	String	是	LEDmac
regid	int	是	字符分区编号
emac	String	是	设备 Mac

返回参数说明:

名称	类型	描述
返回参数		失败: {"statu":0,"msg":"添加失败! "} 成功: {"statu":1,"msg":"添加成功!"}
statu	int	返回状态, 0 失败, 1 成功
msg	String	返回消息

18. 字符分区删除设备

接口名: /API/led/delOneLedEq

接口说明: LED 字符分区删除设备

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg

ledmac	String	是	LEDmac
regid	int	是	字符分区编号
emac	String	是	设备 Mac

返回参数说明:

名称	类型	描述
返回参数		失败: {"statu":0,"msg":"删除失败! "} 成功: {"statu":1,"msg":"删除成功!"}
statu	int	返回状态, 0 失败, 1 成功
msg	String	返回消息

19. 控制 LED 显示屏开关

接口名: /API/led/ledcontrol

接口说明: LED 字符分区删除设备

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID
tokey	String	是	/user/login 返回的 msg
ledmac	String	是	LEDmac
type	String	是	开关状态 0000 关, 0001 开

返回参数说明:

名称	类型	描述
返回参数		失败: {"statu":0,"msg":"发送失败! "} 成功: {"statu":1,"msg":"发送成功!"}
statu	int	返回状态, 0 失败, 1 成功
msg	String	返回消息

20. 管理平台--六条数据统计接口

接口名: /API/App/getTotalData

接口说明: 用户管理平台主页上的---六条数据

请求方式: GET

参数说明:

名称	类型	是否必须	描述
uid	int	是	/user/login 返回的用户 ID

tokey	String	是	/user/login 返回的 msg
-------	--------	---	---------------------

返回参数说明:

名称	类型	描述
eqSum	Int	设备总数
control	Int	设备控制数
collection	Int	设备采集数
yujing	Int	预警总数
onlinesum	Int	在线设备数
unusualSum	Int	异常提醒数
lsnotice	List	近期设备预警消息列表 (采集设备的上下限预警)