
智慧路灯

API 技术文档

版本修订

修改记录累积了每次文档更新的说明。

编号	文档版本	修订人	发布日期	修改说明
1	v1.0.0	方纪正	2021.09.16	初始版本
2	v1.0.1	方纪正	2022.04.03	双灯控制定时参数修改
3	v1.0.2	方纪正	2022.06.07	灯控调光接口添加灯源 2 操作
4	V1.0.3	方纪正	2022.12.20	增加灯控器能耗查询
5	V1.0.4	杨晓玉	2023.2.17	增加下发节目

1 目录

1 目录	2
2 前言	9
3 鉴权	9
3.1 鉴权.....	9
3.1.1 令牌接口（鉴权）.....	9
3.1.1.1 接口描述.....	10
3.1.1.2 接口地址.....	10
3.1.1.3 获取方式.....	10
3.1.1.4 请求参数.....	11
3.1.1.5 响应信息.....	11
3.1.1.6 消息示例.....	11
4 照明	12
4.1 路灯设备列表.....	12
接口地址.....	12
获取方式 POST.....	12
请求参数.....	12
响应信息.....	12
消息示例.....	13
4.2 设备状态.....	14
接口地址.....	14
获取方式 POST.....	14
请求参数.....	14
响应信息.....	14
消息示例.....	15
4.3 添加路灯设备.....	16

接口地址.....	16
获取方式 POST.....	16
请求参数.....	16
响应信息.....	16
消息示例.....	17
4.4 删除设备.....	17
接口地址.....	17
获取方式 POST.....	17
请求参数.....	17
响应信息.....	17
消息示例.....	18
4.5 调节亮度.....	18
接口地址.....	18
获取方式 POST.....	18
请求参数.....	18
响应信息.....	19
消息示例.....	19
4.6 读取定时参数.....	19
接口地址.....	19
获取方式 POST.....	19
请求参数.....	19
响应信息.....	20
消息示例.....	21
4.7 下发定时参数.....	22
接口地址.....	22
获取方式 POST.....	22
请求参数.....	22
响应信息.....	22
消息示例.....	23
5 音响.....	25

5.1 设备列表.....	25
接口地址.....	25
获取方式 POST.....	25
请求参数.....	25
响应信息.....	26
消息示例.....	26
5.2 添加音响.....	27
接口地址.....	27
获取方式 POST.....	27
请求参数.....	27
响应信息.....	27
消息示例.....	28
5.3 音响状态.....	28
接口地址.....	28
获取方式 POST.....	28
请求参数.....	28
响应信息.....	28
消息示例.....	29
5.4 删除音响.....	30
接口地址.....	30
获取方式 POST.....	30
请求参数.....	30
响应信息.....	30
消息示例.....	30
5.5 调节音量.....	31
接口地址.....	31
获取方式 POST.....	31
请求参数.....	31
响应信息.....	31
消息示例.....	31

5.6 媒体库列表.....	32
接口地址.....	32
获取方式 POST.....	32
请求参数.....	32
响应信息.....	32
消息示例.....	33
5.7 添加媒体.....	33
接口地址.....	33
获取方式 POST.....	33
请求参数.....	33
响应信息.....	34
消息示例.....	34
5.8 删除媒体.....	34
接口地址.....	34
获取方式 POST.....	34
请求参数.....	34
响应信息.....	35
消息示例.....	35
5.9 播放/停止音乐.....	35
接口地址.....	35
获取方式 POST.....	36
请求参数.....	36
响应信息.....	36
消息示例.....	36
6 环境监测.....	37
6.1 设备列表.....	37
接口地址.....	37
获取方式 POST.....	37
请求参数.....	37
响应信息.....	37

消息示例.....	38
6.2 环境监测状态	40
接口地址.....	40
获取方式 POST	40
请求参数.....	40
响应信息.....	40
消息示例.....	41
6.3 添加环境监测设备	42
接口地址.....	42
获取方式 POST	43
请求参数.....	43
响应信息.....	43
消息示例.....	43
6.4 移除环境监测	44
接口地址.....	44
获取方式 POST	44
请求参数.....	44
响应信息.....	44
消息示例.....	44
6.5 更新环境监测设备	45
接口地址.....	45
获取方式 POST	45
请求参数.....	45
响应信息.....	45
消息示例.....	45
7 屏幕.....	46
7.1 设备列表.....	46
接口地址.....	46
获取方式 POST	46
请求参数.....	46

响应信息.....	46
消息示例.....	47
7.2 屏幕状态.....	48
接口地址.....	48
获取方式 POST.....	48
请求参数.....	48
响应信息.....	48
消息示例.....	49
7.3 添加屏幕.....	49
接口地址.....	49
获取方式 POST.....	50
请求参数.....	50
响应信息.....	50
消息示例.....	50
7.4 屏幕开关.....	51
接口地址.....	51
获取方式 POST.....	51
请求参数.....	51
响应信息.....	51
消息示例.....	51
7.5 屏幕开关状态.....	52
接口地址.....	52
获取方式 POST.....	52
请求参数.....	52
响应信息.....	52
消息示例.....	52
7.6 设置亮度.....	53
接口地址.....	53
获取方式 POST.....	53
请求参数.....	53

响应信息.....	53
消息示例.....	53
7.7 获取亮度.....	54
接口地址.....	54
获取方式 POST.....	54
请求参数.....	54
响应信息.....	54
消息示例.....	54
7.8 设置音量.....	55
接口地址.....	55
获取方式 POST.....	55
请求参数.....	55
响应信息.....	55
消息示例.....	55
7.9 获取音量.....	56
接口地址.....	56
获取方式 POST.....	56
请求参数.....	56
响应信息.....	56
消息示例.....	56
8 错误码.....	57
8.1 错误编码.....	59
8.1.1 描述.....	59
8.1.2 字典表.....	59

2 前言

- 1、本规范详细介绍每个设备类型对接的接口的功能、输入参数、输出参数、和消息样例等信息。
- 2、接口使用 UTF-8 作为编码。
- 3、除了应用安全(鉴权)接口，其他接口有对应 token 鉴权。
- 4、请求头里加: `Content-Type: application/x-www-form-urlencoded`

3 鉴权

3.1 鉴权

3.1.1 令牌接口（鉴权）

- 1、在 headers 中增加 Basic auth 的认证，填入(appkey, Md5(appsecret+时间戳)+时间戳) (时间戳必须一致)
 - 2、认证方式为 HTTP Basic Authorization，认证信息为 appkey: Md5(appsecret+时间戳)+时间戳，详细调用请参考各个语言的 HTTP Basic Authorization
 - 3、数据接口规范 MD5 小写 32 位，示例：
内容: HelloWorld
摘要: 68e109f0f40ca72a15e05cc22786f8e6
 - 4、名词说明
-

名称	说明
appkey	开发者标识,系统生成
appsecret	开发者密钥,系统生成
timestamp	示例: 当前时间 2021-09-14 10:35:06 时间戳: 1631586906000 固定 13 位长度, 时间戳与服务端时间误差在 5 分钟内有效
认证示例	开发者标识 (appkey) : 7f66a6ca25ff4d18aa8e0f53e287d4b6 开发者密钥 (appsecret) : 2784a65293f341f19f0f5bb1eb777df0 时间戳 (timestamp) : 1631586906000 认证信息 (Authorization) : Basic Base64(7f66a6ca25ff4d18aa8e0f53e287d4b6:Md5(2784a65293f341f19f0f5bb1eb777df01631586906000) 1631586906000)

3.1.1.1 接口描述

实现第三方系统在访问开放 API 之前的认证。

鉴权接口是调用其他 API 的前提, 其他接口调用都需要在 request header 中携带参数 token。

3.1.1.2 接口地址

http://localhost:post/api/ getAccessToken

3.1.1.3 获取方式

为避免乱码问题, 建议使用 POST 方式发起请求。

3.1.1.4 请求参数

参数名称	参数类型	数据类型	必填	说明
Authorization	Head	string	Y	认证信息

3.1.1.5 响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功, 其他: 失败。
msg	string	Y	响应描述:返回对应的描述信息。
data	String	Y	鉴权 token (有效期 15 天)

3.1.1.6 消息示例

```
Method: GET
request:
http://server:port/api/getAccessToken
Header:
"Authorization: Basic MTIzNDU2NDc4OTc5NTY0NjIzMTMyMWprZ2hrZ2NoY2c6dmhqa2dndW1"
response:
Status Code: 200 OK
Body:
{
  "code": 200,
  "msg": success,
  "data": "631359a684db4a02b4f7ba3fde8cb8c9"
}
```

4 照明

4.1 路灯设备列表

接口地址

/api/light/moduleList

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值
total_active_power	有功总功率 (用电量) KWH
latitude	维度
voltage	电压 V
is_lighting	是否亮灯 (0 否 1 是)

update_time	更新时间
current	电流 A
brightness	亮度
serial	序列号
reactive_power	功率因素
name	设备名称
active_power	有功功率
imei	设备 imei
id	设备 id
is_online	是否在线
signal_strength	信号强度
longitude	经度
timestamp	时间戳

消息示例

```
Method: POST
request:
/api/light/moduleList
Header:
token: { token }
Request:
response:
Status Code: 200 OK
Body:
```

```
{
  "msg": "success",
  "code": 10000,
  "data": [
    {
      "total_active_power": 0.0,
      "latitude": "24.5561603867",
      "voltage": 0.0,
```

```
    "is_lighting": null,
    "update_time": null,
    "current": 0.0,
    "brightness": 0,
    "serial": "123456",
    "reactive_power": 0.0,
    "name": "123456",
    "active_power": 0.0,
    "imei": "123456",
    "id": 1,
    "is_online": 0,
    "signal_strength": 0,
    "longitude": "118.1330152193",
    "timestamp": null
  },
]
}
```

4.2 设备状态

接口地址

/api/light/lightState

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
imei	Body	String	Y	设备 imei

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。

msg	响应描述:返回对应的描述信息。
data	接口返回值
total_active_power	有功总功率 (用电量) KWH
voltage	电压 V
is_lighting	是否亮灯 (0 否 1 是)
current	电流 A
brightness	灯源 1 亮度
reactive_power	功率因素
name	设备名称
active_power	有功功率
imei	设备 imei
is_online	是否在线
signal_strength	信号强度
timestamp	时间戳
brightness2	灯源 2 亮度

消息示例

```
Method: POST
request:
/api/light/lightState
Header:
token: { token }
Request:
{imei:12345}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 1000,
```

```
    "data": {
      "is_lighting": null,
      "current": 0.0,
      "brightness": 0,
      "total_active_power": 0.0,
      "reactive_power": 0.0,
      "name": "123456",
      "active_power": 0.0,
      "imei": "12345",
      "is_online": 0,
      "signal_strength": 0,
      "timestamp": null,
      "voltage": 0.0
    }
  }
}
```

4.3 添加路灯设备

接口地址

/api/light/addLight

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
serial	Body	String	Y	设备序列号, 设备外部二维码
name	Body	String	Y	设备名称

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。

msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/light/lightState
Header:
token: { token }
Request:
{ serial:12345
  name:123456
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": ""
}
```

4.4 删除设备

接口地址

/api/light/delLight

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
imei	Body	String	Y	设备 imei

响应信息

参数名称	说明
------	----

code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/light/lightState
Header:
token: { token }
Request:
{ imei:12345
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": ""
}
```

4.5 调节亮度

接口地址

/api/light/adjustLight

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
imei	Body	String	Y	设备 imei
outPower	Body	int	Y	灯源 1: -1 为不操作; 0-10 为关灯; 11-100 为调光
outPower2	Body	int	N	灯源 2: 同灯源 1

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/light/lightState
Header:
token: { token }
Request:
{ imei:12345
  outPower:100
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": ""
}
```

4.6 读取定时参数

接口地址

/api/light/getParamLight

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token

imei	Body	String	Y	设备 imei
------	------	--------	---	---------

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值
stategy_content	<pre>[{ "brightness2": 100,//灯头 2 亮度 "brightness1": 100,//灯头 2 亮度 "time": "12:01",时刻 "state2": 0,//灯头 2 状态 "state1": 1//灯头 1 状态 }, { "brightness2": 100, "brightness1": 100, "time": "12:09", "state2": 0, "state1": 0 }</pre>

]
timestamp	更新时间

消息示例

```
Method: POST
request:
/api/light/getParamLight
Header:
token: { token }
Request:
{ imei:12345
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": {
    strategy_content: "[
      {
        "brightness2": 100,//灯头 2 亮度
        "brightness1": 100,//灯头 1 亮度
        "time": "12:01",时刻
        "state2": 0,//灯头 2 状态
        "state1": 1//灯头 1 状态
      },
      {
        "brightness2": 100,
        "brightness1": 100,
        "time": "12:09",
        "state2": 0,
        "state1": 0
      }
    ]",
    timestamp: 1605007728015
  }
}
```

4.7 下发定时参数

接口地址

/api/light/setParamLight

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
imeis	Body	String	Y	设备 imei
timeBrightness	Body	String	Y	注意: 字符串类型 [{ "brightness2": 100,//灯头 2 亮度 "brightness1": 100,//灯头 2 亮度 "time": "12:01",时刻 "state2": 0,//灯头 2 状态 "state1": 1//灯头 1 状态 }, { "brightness2": 100, "brightness1": 100, "time": "12:09", "state2": 0, "state1": 0 }]

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。

data	接口返回值
strategy_content	<pre>[{ "brightness2": 100,//灯头 2 亮度 "brightness1": 100,//灯头 2 亮度 "time": "12:01",时刻 "state2": 0,//灯头 2 状态 "state1": 1//灯头 1 状态 }, { "brightness2": 100, "brightness1": 100, "time": "12:09", "state2": 0, "state1": 0 }]</pre>
timestamp	更新时间

消息示例

```
Method: POST
request:
/api/light/setParamLight
Header:
token: { token }
Request:
{
```

```
imei:12345
timeBrightness: [{"brightness2": 100,"brightness1": 100,"time": "12:01","state2":
0, "state1": 1},{ "brightness2": 100,"brightness1": 100,"time": "12:09","state2":
0,"state1": 0}]
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": {
    strategy_content: [{"brightness2": 100,"brightness1": 100,"time":
"12:01","state2": 0, "state1": 1},{ "brightness2": 100,"brightness1": 100,"time":
"12:09","state2": 0,"state1": 0}],
    timestamp: 1605007728015
  }
}
```

4.8 能耗查询

接口地址

/api/light/power

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Header	String	Y	鉴权 Token
imei	Body	String	Y	设备 imei
startDate	Body	String	Y	开始日期, 例如: '2022-12-18'
stopDate	Body	String	Y	结束日期, 例如: '2022-12-19'

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。

msg	响应描述:返回对应的描述信息。
data	接口返回值
today_power	当日能耗, kwh
total_power	累计能耗, kwh
date	日期

5 音响

5.1 设备列表

接口地址

/api/audio/audioList

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值
volume	音量
task	任务状态
name	设备名称
time	时间
producer	设备型号
state	是否在线
tid	设备序列号 (audioid)

消息示例

```
Method: POST
request:
/api/audio/audioList
Header:
token: { token }
Request:
{
}
response:
Status Code: 200 OK
Body:
```

```
{
  "msg": "success",
  "code": 10000,
  "data": [
    {
      "volume": null,
      "task": null,
      "user_id": null,
```

```
        "name": null,
        "producer": 2,
        "id": 1,
        "time": null,
        "state": null,
        "tid": "10052"
    }
]
}
```

5.2 添加音响

接口地址

/api/audio/addAudio

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
audiold	Body	String	Y	设备序列号
name	Body	String	Y	设备名称

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/audio/addAudio
Header:
token: { token }
Request:
{
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data":""
}
```

5.3 音响状态

接口地址

/api/audio/audioState

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
audiold	Body	String	Y	音柱序列号

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。

data	接口返回值
volume	音量
task	任务状态
name	设备名称
time	时间
producer	设备型号
state	是否在线
tid	设备序列号 (audioid)

消息示例

```
Method: POST
request:
/api/audio/audioState
Header:
token: { token }
Request:
{
audioId:10052
}
response:
Status Code: 200 OK
Body:
```

```
{
  "msg": "success",
  "code": 10000,
  "data":
  {
    "volume": null,
    "task": null,
    "user_id": null,
    "name": null,
    "producer": 2,
    "id": 1,
    "time": null,
    "state": null,
    "tid": "10052"
  }
}
```

}

5.4 删除音响

接口地址

/api/audio/delAudio

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
audiold	Body	String	Y	音柱序列号

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/audio/delAudio
Header:
token: { token }
Request:
{
}
response:
Status Code: 200 OK
Body:
{
```

```
"msg": "success",
"code": 10000,
"data":
  ""
}
```

5.5 调节音量

接口地址

/api/audio/setTerVolume

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
audiold	Body	String	Y	音柱序列号
bcoutv	Body	String	Y	音量

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/audio/setTerVolume
Header:
token: { token }
Request:
```

```
{
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data":
    ""
}
```

5.6 媒体库列表

接口地址

/api/audio/getMediaData

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值
name	媒体名称
durations	文件持续时间
time	文件大小

type	文件类型
------	------

消息示例

```
Method: POST
request:
/api/audio/getMediaData
Header:
token: { token }
Request:
{
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": [{ "name": "世间美好与你环环相扣.mp3",
    "durations": "00:03:12",
    "id": "1",
    "time": "7.33M",
    "type": "mp3"
  }]
}
```

5.7 添加媒体

接口地址

/api/audio/addMediaData

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
file	Body	File	Y	文件

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/audio/getMediaData
Header:
token: { token }
Request:
{
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data" :
}
}
```

5.8 删除媒体

接口地址

/api/audio/delMediaData

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token

fileName	Body	String	Y	文件名称
----------	------	--------	---	------

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/audio/delMediaData
Header:
token: { token }
Request:
{
fileName:
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data" :
}
}
```

5.9 播放/停止音乐

接口地址

/api/audio/exeRealPlayFile

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
id	Body	String	Y	设备 id, 多个以"<"分隔
state	Body	String	Y	start 开始, stop 结束
file	Body	String	Y	文件名
guid	Body	String	N	停止时使用

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/audio/exeRealPlayFile
Header:
token: { token }
Request:
{
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": "93f04e82-f697-4780-81c0-7a07ce284e3b" 供停止时使用
}
```

6 环境监测

6.1 设备列表

接口地址

/api/environ/environList

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值
date	更新时间
beam	光照
temp	温度
rainfall	雨量

O3	
UV_index	紫外指数
CO2	
wind_direction	风向
pressure	大气压强
CO	
NO2	
pm2	PM2.5
wind_direction_num	风向角度
pm1	PM10
radiation	辐射
serial	环境监测序列号
SO2	
noise	噪声
name	名称
humidity	湿度
wind_speed	风速
is_online	是否在线
sn	网关二维码

消息示例

```
Method: POST
request:
/api/environ/environList
```

```
Header:
token: { token }
Request:
{
}
response:
Status Code: 200 OK
Body:
```

```
{
  "msg": "success",
  "code": 10000,
  "data": [
    {
      "date": null,
      "beam": null,
      "temp": null,
      "rainfall": null,
      "O3": null,
      "UV_index": null,
      "CO2": null,
      "wind_direction": null,
      "pressure": null,
      "CO": null,
      "NO2": null,
      "pm2": null,
      "wind_direction_num": null,
      "pm1": null,
      "radiation": null,
      "serial": "00000001",
      "SO2": null,
      "noise": null,
      "name": "123456",
      "humidity": null,
      "wind_speed": null,
      "is_online": 0,
      "sn": "dsafasfxczvdr113ewrq"
    }
  ]
}
```

6.2 环境监测状态

接口地址

/api/environ/environState

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
serial	Body	String	Y	环境监测序列号

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值
date	更新时间
beam	光照
temp	温度
rainfall	雨量
O3	
UV_index	紫外指数
CO2	

wind_direction	风向
pressure	大气压强
CO	
NO2	
pm2	PM2.5
wind_direction_num	风向角度
pm1	PM10
radiation	辐射
serial	环境监测序列号
SO2	
noise	噪声
name	名称
humidity	湿度
wind_speed	风速
is_online	是否在线
sn	网关二维码

消息示例

```
Method: POST
request:
/api/environ/environState
Header:
token: { token }
Request:
{
serial:
}
response:
Status Code: 200 OK
Body:
```

```
{
  "msg": "success",
  "code": 10000,
  "data": [
    {
      "date": null,
      "beam": null,
      "temp": null,
      "rainfall": null,
      "O3": null,
      "UV_index": null,
      "CO2": null,
      "wind_direction": null,
      "pressure": null,
      "CO": null,
      "NO2": null,
      "pm2": null,
      "wind_direction_num": null,
      "pm1": null,
      "radiation": null,
      "serial": "00000001",
      "SO2": null,
      "noise": null,
      "name": "123456",
      "humidity": null,
      "wind_speed": null,
      "is_online": 0,
      "sn": "dsafasfxczvdr113ewrq"
    }
  ]
}
```

6.3 添加环境监测设备

接口地址

/api/environ/addEnviron

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
sn	Body	String	Y	网关二维码
name	Body	String	Y	名称

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/environ/addEnviron
Header:
token: { token }
Request:
{
sn:
name:
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data":
}
```

6.4 移除环境监测

接口地址

/api/environ/delEnviron

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
serial	Body	String	Y	环境监测序列号

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/environ/delEnviron
Header:
token: { token }
Request:
{
serial:
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data":
```

}

6.5 更新环境监测设备

接口地址

/api/environ/updateState ---异步

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
serial	Body	String	Y	环境监测序列号

响应信息

参数名称	说明
code	响应码。10000: 成功, 其他: 失败。
msg	响应描述:返回对应的描述信息。
data	接口返回值

消息示例

```
Method: POST
request:
/api/environ/updateState
Header:
token: { token }
Request:
{
serial:
}
response:
Status Code: 200 OK
Body:
{
"msg": "success",
```

```
"code": 10000,  
"data":  
  
}
```

7 屏幕

7.1 设备列表

接口地址

/api/led/getAllLed

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token

响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功, 其他: 失败。
msg	array	Y	响应描述:返回对应的描述信息。
data	Object	Y	接口返回值

totalRow			总数
volume	int		音量
brightness	int		亮度
is_open	Int		屏幕是否开启, 0 关, 1 开
led_id	String		设备序列号
update_time	Long		时间戳
width	Int		宽
name	String		名称
id	Int		Id
is_online	Int		是否在线, 0 否, 1 是
height	Int		高

消息示例

```
Method: POST
request:
/api/led/getAllLed
Header:
token: { token }
Request:
response:
Status Code: 200 OK
Body:
```

```
{
  "msg": "success",
  "code": 10000,
  "data": [
    {
      "volume": 53,
      "update_time": 1623402051000,
      "brightness": 49,
      "is_open": 1,
      "led_id": "y60-a20-12345",
      "width": 128,
      "name": "123456",
```

```
        "id": 3,  
        "is_online": 0,  
        "height": 128  
    },  
    ]  
}
```

7.2 屏幕状态

接口地址

/api/led/ledState

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
id	Body	Int	Y	Id

响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功, 其他: 失败。
msg	String	Y	响应描述:返回对应的描述信息。
data	Object	Y	接口返回值
volume	int		音量
brightness	int		亮度
is_open	Int		屏幕是否开启, 0 关, 1 开

led_id	String		设备序列号
update_time	Long		时间戳
width	Int		宽
name	String		名称
id	Int		Id
is_online	Int		是否在线, 0 否, 1 是
height	Int		高

消息示例

```
Method: POST
request:
/api/led/ledState
Header:
token: { token }
Request:
{
  "id": 3,
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": {
    "volume": 53,
    "update_time": 1623402051000,
    "brightness": 49,
    "is_open": 1,
    "led_id": "y60-a20-12345",
    "width": 128,
    "name": "123456",
    "id": 3,
    "is_online": 1,
    "height": 128
  }
}
```

7.3 添加屏幕

接口地址

/api/led/addLed

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
ledId	Body	String	Y	屏幕序列号
name	Body	String	Y	名称

响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功, 其他: 失败。
msg	String	Y	响应描述:返回对应的描述信息。
data	Object	Y	接口返回值

消息示例

```
Method: POST
request:
/api/led/ledState
Header:
token: { token }
Request:
{
  ledId:y60-a20-13456
  name:789456
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": ""
}
```

7.4 屏幕开关

接口地址

/api/led/setScreenOpen

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
id	Body	Int	Y	屏幕 id
ledOpen	Body	int	Y	0 关 1 开

响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功, 其他: 失败。
msg	String	Y	响应描述:返回对应的描述信息。
data	Object	Y	接口返回值

消息示例

```
Method: POST
request:
/api/led/setScreenOpen
Header:
token: { token }
Request:
{
  id:1
  ledOpen:1
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": ""
```

```
}
```

7.5 屏幕开关状态

接口地址

/api/led/isScreenOpen—该接口为主动查询设备接口（异步），无返回值，查询结果在 7.1 设备列表或 7.2 中体现，同以下查询音量，查询亮度接口

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
id	Body	Int	Y	屏幕 id

响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功，其他: 失败。
msg	String	Y	响应描述:返回对应的描述信息。
data	Object	Y	接口返回值

消息示例

```
Method: POST
request:
/api/led/isScreenOpen
Header:
token: { token }
Request:
{
  id:1
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
```

```
"code": 10000,  
"data": ""  
}
```

7.6 设置亮度

接口地址

/api/led/setBrightness

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
id	Body	Int	Y	屏幕 id
brightness	Body	Int	Y	亮度 0-255

响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功, 其他: 失败。
msg	String	Y	响应描述:返回对应的描述信息。
data	Object	Y	接口返回值

消息示例

```
Method: POST  
request:  
/api/led/setBrightness  
Header:  
token: { token }  
Request:  
{  
  id:1  
  brightness:100  
}  
response:
```

```
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": ""
}
```

7.7 获取亮度

接口地址

/api/led/getBrightness

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
id	Body	Int	Y	屏幕 id

响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功, 其他: 失败。
msg	String	Y	响应描述:返回对应的描述信息。
data	Object	Y	接口返回值

消息示例

```
Method: POST
request:
/api/led/getBrightness
Header:
token: { token }
Request:
{
  id:1
}
response:
```

```
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": ""
}
```

7.8 设置音量

接口地址

/api/led/setVolume

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
id	Body	Int	Y	屏幕 id
volume	Body	Int	Y	音量 0-100

响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功, 其他: 失败。
msg	String	Y	响应描述:返回对应的描述信息。
data	Object	Y	接口返回值

消息示例

```
Method: POST
request:
/api/led/setVolume
Header:
token: { token }
Request:
{
```

```
id:1
volume:100
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": ""
}
```

7.9 获取音量

接口地址

/api/led/getVolume

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
id	Body	Int	Y	屏幕 id

响应信息

参数名称	数据类型	必填	说明
code	int	Y	响应码。10000: 成功, 其他: 失败。
msg	String	Y	响应描述:返回对应的描述信息。
data	Object	Y	接口返回值

消息示例

```
Method: POST
request:
/api/led/getVolume
Header:
token: { token }
```

```
Request:
{
  id:1
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": ""
}
```

7.10 节目下发（熙讯）

接口地址

/api/led/onReleaseXiXun

获取方式 POST

请求参数

参数名称	参数类型	数据类型	必填	说明
token	Head	String	Y	鉴权 Token
data	Body	String	Y	下发参数
id	Body	Int	Y	屏幕 id

data 参数示例

```
{
  "type": "commandXixunPlayer", //固定字符串, 不可更改
  "_id": "28dc5b76-51c9-11e9-9b24-4ccc6a95f6dc", //校验命令 id, 可赋随机字符串
  "command": {
    "_type": "PlayXixunTask", //固定字符串, 不可更改
    "id": "28dd5a80-51c9-11e9-9b24-4ccc6a95f6dc", //设备将以该 id 值作为节目唯一 id 保存到设备中,
    可赋随机字符串
    "preDownloadURL": "http://8.136.4.199/upload/", //素材链接, 用于拼接 source 中的 id 构成一个
    完整的链接以便于设备下载素材, 也可以不使用该参数, 在 source 中加入一个 url 字段填入完整的素材下载链接即
    可, 参考下文中的示例部分
    "notificationURL": "http://192.168.2.105:8083/getJSON", //post 函数地址, 用于接收节目下载进
    度
    "task": {
      "_id": "28dd55c6-51c9-11e9-9b24-4ccc6a95f6dc", //校验命令 id, 可赋随机字符串
      "name": "图片节目 test", //总节目名
      "items": [
        {
          "_id": "7111bd0f-edba-4db6-8bd4-f5071d79a265", //校验命令 id, 可赋随机字符串
          "_program": {
            "_id": "28dd2786-51c9-11e9-9b24-4ccc6a95f6dc", //校验命令 id, 可赋随机字符串
```

data	Object	Y	接口返回值
------	--------	---	-------

消息示例

```
Method: POST
request:
/api/led/onReleaseXiXunHeader
token: { token }
Request:
{
  id:1
}
response:
Status Code: 200 OK
Body:
{
  "msg": "success",
  "code": 10000,
  "data": "{
    "_type": "success",
    "_id": "30wztnliduzrq8da0j4axswtzo6hk7",
    "timestamp": 1676601967182
  }"
```

8 错误码

8.1 错误编码

8.1.1 描述

平台统一定义错误编码对应的描述。

8.1.2 字典表

错误编号	错误描述
------	------

10000	成功
10001	请求失败
10002	指令已下发
20004	权限不足
20006	Token 过期
20007	鉴权失败
30001	设备不存在
30002	重复添加
40002	该设备无此功能
