上海顺舟智能科技股份有限公司

**顺舟智能APP**

**功能开发接口文档**

|  |  |  |  |
| --- | --- | --- | --- |
| **编制** | **王保来** | **日期** | **2018-12-12** |
| **审核** |  | **日期** |  |

# 目的

本文档目的是帮助使用顺舟api的客户在开发APP过程中所遇到的接口问题。帮助客户更快的对接顺舟云平台。(本文档适用对接家居设备的客户)

# 范围

本文档适用“顺舟智能”APP开发和需要使用顺舟云平台自主开发app的客户。

# 基础服务

此文档提供顺舟家居app对接顺舟云平台的对接说明，包含接口的调用方式，参数类型，数据返回格式。

BASE\_URL="http://192.168.4.207:8080/CloudPlatform/app/";

VERSION\_NUM = "/v1\_0/";

# 消息主体规范

|  |  |  |
| --- | --- | --- |
| 参数名 | 说明 | 举例 |
| OperatorID | 调用方唯一标识 |  |
| Data | 各接口具体参数信息 | 以下接口的请求参数需进行CBC加密之后放在此Data参数中 |
| TimeStamp | 时间戳 | 接口请求时的时间戳信息，参  照GB/T7408，格式为  yyyyMMddHHmmss |
| Seq | 自增序列 | 4位自增序列取自时间戳，同  一秒内按序列自增长，新秒重  计，如0001 |
| Sig | 参数签名 |  |

说明: 1、[CBC加密在附件中贴出](#_CBC加密方法)

2、[Sig签名方法](#_Sig签名方法)

# 接口说明

## 查询Token信息接口

### 1接口描述

通过查询Token

|  |  |  |  |
| --- | --- | --- | --- |
| path | 版本 | 描述 | 请求方式 |
| query\_token | 无 |  | Post |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| OperatorID | String | 是 | 757915966 |
| OperatorSecret | String | 是 | 91a27d564842a1bb |

### 3返回参数

## 登录

### 1接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| user/v1\_0/user\_login | v1\_0 | post |  |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| UserName | String | 是 | 用户名 |
| Password | string | 是 | 密码 |

### 3请求示例

### 4返回参数

## 验证码

### 1、接口描述

调用此接口之前需要先查询Token，token获取成功之后才能使用此接口，接口的功能用于用户手机或者邮箱注册时，发送验证码功能；同时适用于密码重置时发送验证码

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| user/v1\_0/send\_verification\_code | v1\_0 | post | 发送验证码接口 |

### 2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| VerType | Int | 是 | 1表示注册发送验证码  2密码重置 |
| Type | Int | 是 | 1:手机号  2表示邮箱注册 |
| Tel | String | Type为1时必填 | 手机号码 |
| Email | String | Type为2时必填 | 邮箱 |

### 3、请求示例

{"Email":"752134268@qq.com","VerType":1,"Type":2}

### 4、返回参数

{"SuccStat":2}

SuccStat:0表示发送成功

1表示发送验证码太频繁

2表示账号已存在

3表示账号不存在(重置密码的时候需要)

## 校验 验证码

### 1、接口描述

用于校验验证码是否正确

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| user/v1\_0/validate\_verification\_code | v1\_0 | post | 发送验证码接口 |

### 2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| VerType | Int | 是 | 1表示注册发送验证码  2密码重置 |
| Code | String | 是 | 收到的验证码 |
| Type | Int | 是 | 1:手机号  2表示邮箱注册 |
| Tel | String | Type为1时必填 | 手机号码 |
| Email | String | Type为2时必填 | 邮箱 |

### 3、请求示例

{"Email":"752134268@qq.com","VerType":1,"Code":"575547","Type":2}

### 4、返回参数

{"SuccStat":0,"Result":true}

SuccStat:0表示发送成功

1表示发送验证码太频繁

2表示账号已存在

3表示账号不存在(重置密码的时候需要)

## 重置密码

### 1、接口描述

用于找回或者修改密码

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| user/v1\_0/app\_user\_password\_reset | v1\_0 | post | 重置密码 |

### 2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| Password | String | 是 | 新密码 |
| Type | Int | 是 | 1:手机号  2表示邮箱注册 |
| Tel | String | Type为1时必填 | 手机号码 |
| Email | String | Type为2时必填 | 邮箱 |

### 3、请求示例

{

“Type”:1,

“Tel”:”13698815777”,

“Password”:”xxx”

}

### 4、返回参数

{"SuccStat":0 }

## 注册

### 1、接口描述

手机号或者邮箱注册

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| user/v1\_0/app\_user\_register | v1\_0 | post | 注册 |

### 2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| FullName | String | 是 | 昵称 |
| Type | Int | 是 | 1:手机号  2表示邮箱注册 |
| Password | String | 是 | 密码 |
| Tel | String | Type为1时必填 | 手机号码 |
| Email | String | Type为2时必填 | 邮箱 |

### 3、请求示例

**{**

**“Type”:1,**

**“Tel”:”13698815777”,**

**“Password”:”xx”,**

**“FullName”:”xx”**

**}**

### 4、返回参数

**{"SuccStat":0 }**

**SuccStat:0表示发送成功**

**1表示账号已存在**

## 创建分组

### 1接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| group/v1\_0/create | v1\_0 | post |  |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| code | Int | 是 | 2001 |
| platform | String | 是 | 平台类型，家居传2 |
| addr | String | 是 | 网关的地址 |
| gname | String | 是 | 分组的名称 |
| device | jsonArray | 是 | 传要加入分组的设备id和设备端口号例如: {"ep":1,"id":"010100124b000fe220dd"} |

### 3请求示例

{

"device": [{

"ep": 1,

"id": "010100124b000fe220dd"

}, {

"ep": 2,

"id": "010100124b000fe220dd"

}, {

"ep": 3,

"id": "010100124b000fe220dd"

}, {

"ep": 1,

"id": "010100124b000e0bdaf7"

}],

"gname": "测试分组",

"platform": 2,

"addr": "2c:6a:6f:00:51:9b",

"code": 2001

}

### 4返回参数

{

"serial": 2,

"gname": "测试分组",

"device": [{

"ep": 1,

"id": "010100124b000fe220dd"

}, {

"ep": 2,

"id": "010100124b000fe220dd"

}, {

"ep": 3,

"id": "010100124b000fe220dd"

}, {

"ep": 1,

"id": "010100124b000e0bdaf7"

}],

"cmdsource": 2,

"gid": 2,

"result": 0,

"code": 2001

}

## 获取分组

### 1接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| group/v1\_0/read | v1\_0 | post | 家居和路灯获取分组公用此接口 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| skip | Int | 是 | 从0开始，计算方式: *第n页，每页3个，skip应该是limit（totalNumber-1），limit就是3，但是你需要先判断这个n超过总页数没，超过了skip就是0\*/* |
| platform | String | 是 | 平台类型，家居传2 |
| limit | String | 是 | 每页展示的数量 |
| name | String | 是 | 当不为空时表示搜索分组 |

### 3请求示例

{"skip":0,"limit":14,"platform":2,"name":""}

### 4返回参数

{

"total": 2,

"rows": [{

"modifyUser": "5a72e3efb7384e2e48e333a0",

"gid": 1,

"modifyDate": "Wed Nov 07 18:28:25 CST 2018",

"addUser": "5a72e3efb7384e2e48e333a0",

"id": "5be2be490e374d388014527b",

"gname": "测试分组",

"addDate": "Wed Nov 07 18:28:25 CST 2018",

"device": [{

"ep": 1,

"id": "010100124b000fe220dd"

}, {

"ep": 2,

"id": "010100124b000fe220dd"

}, {

"ep": 3,

"id": "010100124b000fe220dd"

}, {

"ep": 1,

"id": "010100124b000e0bdaf7"

}],

"mac": "2c:6a:6f:00:51:9b"

}, {

"modifyUser": "5a72e3efb7384e2e48e333a0",

"gid": 2,

"modifyDate": "Wed Nov 07 19:01:23 CST 2018",

"addUser": "5a72e3efb7384e2e48e333a0",

"id": "5be2c6030e374d388014527c",

"gname": "测试分组",

"addDate": "Wed Nov 07 19:01:23 CST 2018",

"device": [{

"ep": 1,

"id": "010100124b000fe220dd"

}, {

"ep": 2,

"id": "010100124b000fe220dd"

}, {

"ep": 3,

"id": "010100124b000fe220dd"

}, {

"ep": 1,

"id": "010100124b000e0bdaf7"

}],

"mac": "2c:6a:6f:00:51:9b"

}]

}

## 删除分组

### 1接口描述

通过此接口用来删除分组

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| group/v1\_0/delete | v1\_0 | post | 删除分组接口 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| gid | Int | 是 | 分组gid |
| uid | String | 是 | 分组的uid |
| addr | String | 是 | 网关的mac地址 |
| platform | String | 是 | 平台类型，家居传2 |

### 3请求示例

{"gid":1,"uid":"5be2be490e374d388014527b","platform":2,"addr":"2c:6a:6f:00:51:9b"}

### 4返回参数

{"gid":1,"serial":1,"cmdsource":2,"result":0,"code":2004}

## 修改分组

### 1接口描述

通过此接口用来修改分组信息

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| group/v1\_0/update | v1\_0 | post | 修改分组接口 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| gid | Int | 是 | 分组gid |
| uid | String | 是 | 分组的uid |
| addr | String | 是 | 网关的mac地址 |
| platform | String | 是 | 平台类型，家居传2 |
| gid | Int | 是 | 要修改的分组gid |
| gname | String | 是 | 分组名称 |
| uid | String | 是 | 数据库主键,分组列表中对应的id字段 |
| device | JsonArray | 是 | 只需要传要修改的设备，status = 0 是添加设备; status = 1删除设备 |

### 3请求示例

{

"device": [{

"ep": 1,

"id": "010100124b000e0bdaf7",

"status": 1

}, {

"ep": 2,

"id": "010100124b000e0bdaf7",

"status": 0

}],

"uid": "5be2c6030e374d388014527c",

"gname": "测试分组",

"gid": 2,

"platform": 2,

"addr": "2c:6a:6f:00:51:9b"

}

### 4返回参数

{

"code": 2002,

"serial": 10,

"result": 201,

"response": "gateway server timeout"

}

## 获取常用设备列表

### 1接口描述

通过此接口用来修改分组信息

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| /commons/v1\_0/read | v1\_0 | post | 获取设备列表 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| limit | Int | 是 | 每页请求的数量 |
| platform | Int | 是 | 平台类型，家居传2 |
| skip | Int | 是 | 网关的mac地址 |
| type | Int | 是 | 平台类型，家居传2 |
| country | String | 否 | 语言类型，英文环境下传US,其它默认为CN |
| name | String | 是 | 要搜索的关键字 |

### 3请求示例

{

"limit": 14,

"platform": 2,

"skip": 0,

"type": -1,

"country": "CN",

"name": ""

}

### 4返回参数

其中type 参考”[设备类型说明](#_设备类型说明)”

[{

"modifyUser": "5a72e3efb7384e2e48e333a0",

"endpoints": [{

"dsp": "Sz0001 ",

"uid": "5be9638e26b06747f6f551c6",

"port\_id": 2,

"fac": "Shuncom",

"gateway\_id": "5bea9cf7af51078c221ce293",

"refresh\_time": 1542355830000,

"on": "1",

"eplist": "002001"

}, {

"dsp": "Sz0001 ",

"uid": "5be9638e26b06747f6f551c7",

"port\_id": 1,

"fac": "Shuncom",

"swid": "HierSWh2\_BP\_3.25",

"gateway\_id": "5bea9cf7af51078c221ce293",

"refresh\_time": 1542355830000,

"on": "1",

"eplist": "002001"

}],

"modifyDate": 1542285135040,

"addUser": "5a72e3efb7384e2e48e333a0",

"name": "0521",

"online": 1,

"addr": "010100124b000e4f0521",

"type": 140,

"addDate": 1542022030772,

"mac": "2c:6a:6f:00:5f:a7",

"dblControlZigbeeLamp": false

}, {

"modifyUser": "5a72e3efb7384e2e48e333a0",

"endpoints": [{

"dsp": "SZSL",

"reacte": "0",

"apte": "0",

"gateway\_id": "5bea9cf7af51078c221ce293",

"lightrbt": "0",

"refresh\_time": 1542279404000,

"uid": "5be258f926b0671e69f052c9",

"lightrt": "0",

"lightf": "4993",

"actp": "0",

"reactp": "17",

"volt": "23026",

"mtemp": "15692",

"bri": "0",

"port\_id": 1,

"curr": "7",

"acte": "0",

"eplist": "001",

"on": "0"

}],

"modifyDate": 1541841625896,

"addUser": "5a72e3efb7384e2e48e333a0",

"name": "12",

"online": 0,

"addr": "02020000000001005012",

"type": 231,

"addDate": 1541560569751,

"mac": "2c:6a:6f:00:5f:a7",

"dblControlZigbeeLamp": false

}]

## 获取设备详情

### 1接口描述

通过此接口用来修改分组信息

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| polymerize/v1\_0/query\_polymerize\_by\_addr | v1\_0 | post | 根据地址查询设备的详细信息其中包含设备的最新状态 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| type | Int | 是 | 设备类型参考[”设备类型说明”](#_设备类型说明) |
| addr | Int | 是 | 设备的addr |

### 3请求示例

{

"type": 140,

"addr": "010100124b000e4f0521"

}

### 4返回参数

[{

"modifyUser": "5a72e3efb7384e2e48e333a0",

"endpoints": [{

"dsp": "Sz0001 ",

"uid": "5be9638e26b06747f6f551c6",

"port\_id": 2,

"fac": "Shuncom",

"gateway\_id": "5bea9cf7af51078c221ce293",

"refresh\_time": 1542358355000,

"on": "1",

"eplist": "002001"

}, {

"dsp": "Sz0001 ",

"uid": "5be9638e26b06747f6f551c7",

"port\_id": 1,

"fac": "Shuncom",

"swid": "HierSWh2\_BP\_3.25",

"gateway\_id": "5bea9cf7af51078c221ce293",

"refresh\_time": 1542358355000,

"on": "1",

"eplist": "002001"

}],

"modifyDate": 1542285135040,

"addUser": "5a72e3efb7384e2e48e333a0",

"name": "0521",

"online": 1,

"addr": "010100124b000e4f0521",

"type": 140,

"addDate": 1542022030772,

"mac": "2c:6a:6f:00:5f:a7",

"dblControlZigbeeLamp": false

}]

## 解绑子设备

### 1接口描述

通过此接口用来修改分组信息

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| commons/v1\_0/ delete | v1\_0 | post | 解绑设备 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| type | Int | 是 | 设备类型参考[”设备类型说明”](#_设备类型说明) |
| uid | JSONArray | 是 | 设备端口的uid |
| property | JSONObject | 是 | 参照property数据 |
| platform | int | 是 | 2 |

**property**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| addr | Int | 是 | 设备所在网关的mac地址(例如*2c:6a:6f:00:50:e0*) |
| id | JSONArray | 是 | 设备的addr(例如*010300124b0008f1e921*  ) |

通过此接口用来修改分组信息

### 3请求示例

{

"type": 140,

"addr": "010100124b000e4f0521"

}

### 4返回参数

[{

"modifyUser": "5a72e3efb7384e2e48e333a0",

"endpoints": [{

"dsp": "Sz0001 ",

"uid": "5be9638e26b06747f6f551c6",

"port\_id": 2,

"fac": "Shuncom",

"gateway\_id": "5bea9cf7af51078c221ce293",

"refresh\_time": 1542358355000,

"on": "1",

"eplist": "002001"

}, {

"dsp": "Sz0001 ",

"uid": "5be9638e26b06747f6f551c7",

"port\_id": 1,

"fac": "Shuncom",

"swid": "HierSWh2\_BP\_3.25",

"gateway\_id": "5bea9cf7af51078c221ce293",

"refresh\_time": 1542358355000,

"on": "1",

"eplist": "002001"

}],

"modifyDate": 1542285135040,

"addUser": "5a72e3efb7384e2e48e333a0",

"name": "0521",

"online": 1,

"addr": "010100124b000e4f0521",

"type": 140,

"addDate": 1542022030772,

"mac": "2c:6a:6f:00:5f:a7",

"dblControlZigbeeLamp": false

}]

## 解绑网关设备

### 1接口描述

通过此接口用来修改分组信息

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| commons/v1\_0/ delete | v1\_0 | post | 解绑设备 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| type | Int | 是 | 设备类型参考”[网关类型说明](#_网关类型说明)” |
| uid | string | 是 | 网关的uid |
| platform | int | 是 | 2 |

### 3请求示例

{

"type": 140,

"uid": "010100124b000e4f0521"

“platform”:2

}

### 4返回参数

[{

"modifyUser": "5a72e3efb7384e2e48e333a0",

"endpoints": [{

"dsp": "Sz0001 ",

"uid": "5be9638e26b06747f6f551c6",

"port\_id": 2,

"fac": "Shuncom",

"gateway\_id": "5bea9cf7af51078c221ce293",

"refresh\_time": 1542358355000,

"on": "1",

"eplist": "002001"

}, {

"dsp": "Sz0001 ",

"uid": "5be9638e26b06747f6f551c7",

"port\_id": 1,

"fac": "Shuncom",

"swid": "HierSWh2\_BP\_3.25",

"gateway\_id": "5bea9cf7af51078c221ce293",

"refresh\_time": 1542358355000,

"on": "1",

"eplist": "002001"

}],

"modifyDate": 1542285135040,

"addUser": "5a72e3efb7384e2e48e333a0",

"name": "0521",

"online": 1,

"addr": "010100124b000e4f0521",

"type": 140,

"addDate": 1542022030772,

"mac": "2c:6a:6f:00:5f:a7",

"dblControlZigbeeLamp": false

}]

## 获取消息通知按钮的状态

### 1接口描述

通过此接口用来获取消息通知按钮的状态

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| message/v1\_0/msg\_state | v1\_0 | post | 用户获取消息按钮的最新状态 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| 无 |  |  | 此接口不需要参数 |

### 3请求示例

Data 参数传空字符串即可

{"TimeStamp":"20181119140707$-480","Seq":"0001","Data":"","OperatorID":"757915966","Sig":"65F1BE5B2FC204FD3569707BA6F1D56D"}

### 4返回参数

{"state":"1"}

1表示开启，0表示关闭

## 设置消息通知按钮的状态

### 1接口描述

通过此接口用来获取消息通知按钮的状态

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| message/v1\_0/msg\_push | v1\_0 | post | 用户获取消息按钮的最新状态 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| ClientId | String | 是 | 集成个推获取到的clientid，每一个设备对应一个clientid |
| State | String | 是 | 0 关闭；1开启 |

### 3请求示例

{

"ClientId": "12ec84cb438052323619bf9dee6115a4",

"State": "1"

}

### 4返回参数

{

"msg": "success",

"userId": "5a72e3efb7384e2e48e333a0",

"clientId": "12ec84cb438052323619bf9dee6115a4",

"state": "1"

}

1表示开启，0表示关闭

## 房间创建

### 1接口描述

通过此接口用来创建房间

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| room/v1\_0/create | v1\_0 | post | 房间创建 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| Name | String | 是 | 房间名称 |
| Addr | String | 是 | 网关的mac地址 |
| Devs | JSONArray | 是 | 选择的设备列表 |

### 3请求示例

{

"Devs": [{

"type": 105,

"id": "010100124b000fe220dd",

"ep": 3

}, {

"type": 105,

"id": "010100124b000e0bdaf7",

"ep": 1

}],

"Addr": "2c:6a:6f:00:51:9b",

"Name": "主卧"

}

### 4返回参数

{

"devs": [{

"id": "010100124b000fe220dd",

"ep": 3,

"type": 105

}, {

"id": "010100124b000e0bdaf7",

"ep": 1,

"type": 105

}],

"result": 0,

"code": 2333,

"name": "主卧",

"id": "5bf37a74505bd537d8a8c308",

"addr": "2c:6a:6f:00:51:9b"

}

## 房间获取

### 1接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| room/v1\_0/read | v1\_0 | post | 获取房间 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| Name | String | 是 | 房间名 可以为空字符串 空字符串为查全部 |
| Skip | Int | 是 | 第几条开始 |
| Limit | Int | 是 | 每页限制条数 |

### 3请求示例

{"Name":"","Skip":0,"Limit":100}

### 4返回参数

{

"total": 1,

"rows": [{

"devs": [{

"ep": 3,

"id": "010100124b000fe220dd",

"type": 105

}, {

"ep": 1,

"id": "010100124b000e0bdaf7",

"type": 105

}],

"addUser": "5a72e3efb7384e2e48e333a0",

"name": "主卧",

"id": "5bf37a74505bd537d8a8c308",

"addr": "2c:6a:6f:00:51:9b",

"addDate": "Tue Nov 20 11:07:32 CST 2018"

}]

}

## 房间删除

### 1接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| room/v1\_0/delete | v1\_0 | post | 删除房间 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| Id | String | 是 | 房间id |

### 3请求示例

{"Id":"5bf27d5f37220b14d4f64ef9"}

### 4返回参数

{"msg":"success","code":2333,"count":1}

## 策略创建或者编辑

### 1接口描述

通过此接口用来创建策略或者编辑策略 /strategy/v1\_0/read

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| strategy/v1\_1/create | v1\_1 | post | 策略的编辑或者创建 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| platform | String | 是 | 房间名称 |
| property | JsonObject | 是 | 参照下面property表格 |

property参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| uid | string | 创建策略时不需要传此参数，编辑策略是要传 | uid为网关mac地址 |
| body | JsonObject | 是 | 参照body表格 |
| addr | String | 是 | 网关的mac地址 |

body参数说明(此body里面的内容为网关本地创建策略时候的参数)

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| code | int | 是 | 4001 |
| name | string | 是 | 策略名称 |
| rid | int | 否;编辑策略时必传 | 策略ID |
| state | int | 是 | 策略使能：  0：disable  1：enable |
| ct | string | 是 | 创建时间格式为:2019-02-20T14:34:31 |
| exp | string | 是 | 策略表达式（使用lua语言描述）exp中main(a1,a2,a3,a4)a1、a3分别为cond中条件1、2字段的实际值，a2、a4分别为条件1、2中的设定值。 |
| trig | int | 是 | 策略计划被触发的次数；0表示持续执行。 |
| act | JsonArray | 是 | 要执行的动作,参考act表格参数说明 |
| cond | JsonArray | 是 | 要执行的动作,参考cond表格参数说明 |

cond 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必填 | 描述 |
| idx | int | 是 | 条件序号 |
| type | int | 是 | 条件类型：  1：定时（格林威治时间，只支持逻辑运算符”==”）  2：触发  3：周期轮询 |
| name | string | 是 | 设备的名称 |
| cmd | string | 是 | 设备的可控制的属性 |
| id | string | 是 | 设备的id |
| ep | int | 是 | 设备的端口号 |
| val | string | 是 | 要满足的属性值 |

act参数表格说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必填 | 描述 |
| idx | int | 是 | 动作序号 |
| type | int | 是 | 目标类型：  1：设备ID，160bits  2：group id，16bits  3：scene id，8bits  4：rules |
| cmd | string | 是 | 设备的可控制的属性 |
| id | string | 是 | 设备的id |
| ep | int | 是 | 设备的端口号 |
| val | int | 是 | 要满足的属性值 |
| delay | int | 是 | 动作延迟执行的时间，单位为秒 |

### 3请求示例

{  
    **"platform"**:**2**,  
    **"property"**:{  
        **"body"**:{  
            **"trig"**:**0**,  
            **"exp"**:**"function main(a1,a2) if (a1 == a2) then return true else return false end end"**,  
            **"name"**:**"wbl测试"**,  
            **"state"**:**0**,  
            **"code"**:**4001**,  
            **"act"**:[  
                {  
                    **"val"**:**0**,  
                    **"ep"**:**1**,  
                    **"id"**:**"0302002c6a6f0051b401"**,  
                    **"cmd"**:**"runsta"**,  
                    **"delay"**:**0**,  
                    **"idx"**:**1**,  
                    **"type"**:**1**  
                }  
            ],  
            **"cond"**:[  
                {  
                    **"val"**:**"0"**,  
                    **"ep"**:**1**,  
                    **"id"**:**"0302002c6a6f0051b401"**,  
                    **"cmd"**:**"runsta"**,  
                    **"idx"**:**1**,  
                    **"type"**:**2**,  
                    **"name"**:**"6-1空调"**  
                }  
            ],  
            **"ct"**:**"2019-02-20T14:34:31"**  
        },  
        **"addr"**:**"2c:6a:6f:00:51:b4"**  
    }  
}

### 4返回参数

{  
    **"cmdsource"**:**2**,  
    **"code"**:**4001**,  
    **"trig"**:**0**,  
    **"cond"**:[  
        {  
            **"val"**:**"0"**,  
            **"name"**:**"6-1空调"**,  
            **"ep"**:**1**,  
            **"id"**:**"0302002c6a6f0051b401"**,  
            **"cmd"**:**"runsta"**,  
            **"idx"**:**1**,  
            **"type"**:**2**  
        }  
    ],  
    **"rid"**:**3**,  
    **"result"**:**0**,  
    **"ct"**:**"2019-02-20T14:34:31"**,  
    **"uid"**:**"5c6cf4f7472bfd8e6d749d5c"**,  
    **"act"**:[  
        {  
            **"val"**:**0**,  
            **"delay"**:**0**,  
            **"ep"**:**1**,  
            **"id"**:**"0302002c6a6f0051b401"**,  
            **"cmd"**:**"runsta"**,  
            **"idx"**:**1**,  
            **"type"**:**1**  
        }  
    ],  
    **"serial"**:**15**,  
    **"name"**:**"wbl测试"**,  
    **"state"**:**0**,  
    **"exp"**:**"function main(a1,a2) if (a1 == a2) then return true else return false end end"**  
}

## 策略获取

### 1接口描述

通过此接口用来获取已创建的策略列表

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| strategy/v1\_0/read | v1\_0 | post | 获取策略列表 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| skip | Int | 是 | 从0开始，计算方式: *第n页，每页3个，skip应该是limit（totalNumber-1），limit就是3，但是你需要先判断这个n超过总页数没，超过了skip就是0\*/* |
| platform | String | 是 | 平台类型，家居传2 |
| limit | String | 是 | 每页展示的数量 |
| name | String | 是 | 当不为空时表示搜索分组 |

### 3请求示例

{"limit":14,"skip":0,"platform":2,"name":""}

### 4返回参数

{

"total": 29,

"rows": [{

"ct": "2018-09-11 21:58:03",

"act": [{

"val": true,

"uid": "5c35ee32b0037396dede3ff1",

"delay": 0,

"name": "会议室一键6660",

"id": "010100124b00170f6660",

"ep": 1,

"cmd": "on",

"idx": 1,

"type": 1,

"attrtype": 1

}],

"name": "mhouseStart",

"trig": 71,

"id": "5c35ed37fc5e593d247bacf1",

"state": 1,

"exp": "function main(a1,a2) if (a1 == a2) then return true else return false end end",

"rid": 74,

"cond": [{

"val": "true",

"id": "010100124b000e4fcc5d",

"ep": 1,

"cmd": "on",

"idx": 1,

"type": 2

}],

"mac": "a0:9d:c1:b0:2c:bb"

}, {

"ct": "2018-09-11 22:09:56",

"act": [{

"val": false,

"uid": "5c35ee32b0037396dede3ff1",

"delay": 10,

"name": "会议室一键6660",

"id": "010100124b00170f6660",

"ep": 1,

"cmd": "on",

"idx": 1,

"type": 1,

"attrtype": 0

}, {

"val": 1,

"uid": "5c35eeb6b0037396dede3ff3",

"delay": 0,

"name": "会议室二路窗帘3dea",

"id": "010100124b00090a3dea",

"ep": 1,

"cmd": "cts",

"idx": 2,

"type": 1,

"attrtype": 3

}, {

"val": 1,

"uid": "5c35eeb6b0037396dede3ff4",

"delay": 0,

"name": "会议室二路窗帘3dea",

"id": "010100124b00090a3dea",

"ep": 2,

"cmd": "cts",

"idx": 3,

"type": 1,

"attrtype": 3

}],

"name": "mhouseGo",

"trig": 156,

"id": "5c35ed37fc5e593d247bacf2",

"state": 1,

"exp": "function main(a1,a2) if (a1 == a2) then return true else return false end end",

"rid": 63,

"cond": [{

"val": "true",

"id": "010100124b000e4fcc5d",

"ep": 4,

"cmd": "on",

"idx": 1,

"type": 2

}],

"mac": "a0:9d:c1:b0:2c:bb"

}]

}

## 策略删除

### 1接口描述

通过此接口用来删除策略

|  |  |  |  |
| --- | --- | --- | --- |
| Path | 版本 | 请求方式 | 描述 |
| strategy/v1\_1/delete | v1\_1 | post | 删除策略 |

### 2请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| uid | Int | 是 | 从平台获取到的uid |
| platform | int | 是 | 平台类型，家居传2 |
| property | JSONObject | 是 | 参照property参数说明 |

**property参数说明**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| addr | string | 是 | 网关的mac地址 |
| body | JSONObject | 是 | 参照body参数说明 |

**body参数说明**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 必填 | 描述 |
| rid | int | 是 | 策略的rid |
| code | int | 是 | 4003 |

### 3请求示例

{  
    **"uid"**:**"5c6cf4f7472bfd8e6d749d5c"**,  
    **"platform"**:**2**,  
    **"property"**:{  
        **"body"**:{  
            **"rid"**:**3**,  
            **"code"**:**4003**  
        },  
        **"addr"**:**"2c:6a:6f:00:51:b4"**  
    }  
}

### 4返回参数

{  
    **"serial"**:**76**,  
    **"rid"**:**3**,  
    **"result"**:**0**,  
    **"cmdsource"**:**2**,  
    **"code"**:**4003**  
}

# 附件

## CBC加密方法

**public static final** String ***KEY\_ALGORITHM*** = **"AES"**;  
**public static final** String ***CBC\_CIPHER\_ALGORITHM*** = **"AES/CBC/PKCS5Padding"**;  
**public static final** String ***SIG\_ALGORITHM*** = **"HmacMD5"**;  
**public static final** String ***SECRETKEY*** = **"2dd19414e91ac01a"**;  
**public static final** String ***IVVECTOR*** = **"1524b9a8c447e586"**;

**try** {  
 IvParameterSpec ivParameterSpec = **new** IvParameterSpec(***IVVECTOR***.getBytes(**"utf-8"**));  
 Cipher cipher = Cipher.*getInstance*(***CBC\_CIPHER\_ALGORITHM***);  
 cipher.init(Cipher.***ENCRYPT\_MODE***, **new** SecretKeySpec(***SECRETKEY***.getBytes(**"utf-8"**), ***KEY\_ALGORITHM***), ivParameterSpec);**return** Base64.*encodeToString*(cipher.doFinal(sourceText.getBytes(**"utf-8"**)),Base64.***NO\_WRAP***);

} **catch** (NoSuchAlgorithmException | NoSuchPaddingException  
 | InvalidKeyException | InvalidAlgorithmParameterException  
 | IllegalBlockSizeException | BadPaddingException | UnsupportedEncodingException e) {e.printStackTrace();  
 }

## Sig签名方法

**参数说明**

operatorID: "757915966"

data:需传入加密后的数据

timeStamp:时间戳 格式为yyyyMMddHHmmss

seq: "0001"

secretKey: "d66834bd4a978d93"

**public static** String HMACMD5RequestEncode(String operatorID, String data, String timeStamp, String seq, String secretKey) {  
 **try** {  
 SecretKeySpec sks = **new** SecretKeySpec(secretKey.getBytes(**"utf-8"**), ***SIG\_ALGORITHM***);  
 Mac mac = Mac.*getInstance*(***SIG\_ALGORITHM***);  
 mac.init(sks);  
 **return** *byte2HexStr*(mac.doFinal((operatorID + data + timeStamp + seq).getBytes(**"utf-8"**)));  
 } **catch** (NoSuchAlgorithmException | InvalidKeyException | IllegalStateException | UnsupportedEncodingException e) {  
 e.printStackTrace();  
 }  
 **return null**;  
}

# 设备类型说明

|  |  |
| --- | --- |
| type值 | 代表类型 |
| 101 | 门磁传感器 |
| 102 | 插座 |
| 103 | 微波感应 |
| 104 | 春泉空调面板 |
| 105 | 开关面板 |
| 106 | 窗帘面板 |
| 107 | 水浸传感器 |
| 108 | 烟雾传感器 |
| 109 | 声光传感器 |
| 110 | 环境盒子 |
| 111 | 红外线传感器 |
| 112 | 可燃气体传感器 |
| 113 | 温湿度传感器 |
| 114 | 聚盒灯控器 |
| 115 | 锁 |
| 116 | 窗帘电机 |
| 117 | 红外转发 |
| 118 | 情景面板 |
| 119 | 计量插座 |
| 120 | 调光面板 |
| 121 | 家居灯具 |
| 122 | 光照度传感器 |
| 123 | 紧急按钮 |
| 124 | 温湿度传感器\_2 |
| 125 | 氨气检测传感器 |
| 126 | 硫化氢检测传感器 |
| 138 | 橙朴-海尔四键 |
| 139 | 橙朴-海尔三键 |
| 140 | 橙朴-海尔二键 |
| 141 | 橙朴-海尔一键 |
| 166 | 申炜一路窗帘面板 |
| 168 | 调光面板 |
| 169 | 调光面板 |
| 189 | 电表 |
| 231 | 路灯灯具 |

# 网关类型说明

|  |  |
| --- | --- |
| type值 | 代表类型 |
| 0 | 大聚盒网关 |
| 1 | 四代网关 |
| 2 | 网关应用程序 |
| 3 | STM32 |
| 4 | 小聚盒网关 |
| 5 | 家居网关 |
| 6 | 音乐网关(2530) |
| 7 | 音乐网关(EFR32) |
| 8 | 离线语音网关 |
| 9 | 穿墙王网关 |
| 10 | 易百珑网关 |

删除测试