

智慧家庭/主机接入服务器系统 - 接口设计

Revision:

2019.4.19 v0.1 scott

- 创建

2019.4.22 v0.2 scott

- 保持api接口与smartbox主机设备api一致

2019/6/1 v0.3.1 scott

- 增加 1.5 , 1.6 , 1.7
- 所有RETS接口增加 authcode 参数 ;
- 在Tcp通道上增加 订阅设备消息接口

2019/6/2 v0.3.2 scott

- 1.增加tcp 取消订阅, 心跳 消息
- 2.增加获取设备列表接口 (1.7)
3. 增加 平台app授权码请求接口(1.8)

目录

1. REST主机接口

[1.1 主机状态查询](#)

[1.2 室内主机参数设置](#)

[1.3 端设备状态值查询](#)

[1.4 端设备参数设置\(控制](#)

[1.5 获取设备profile信息](#)

[1.6 获取推送服务器地址](#)

[1.8 获取设备列表](#)

[1.8 App登录请求授权码](#)

2. 设备TCP接口

[2.1 设备消息订阅](#)

[2.2 取消订阅](#)

[2.3 心跳消息](#)

[2.4 设备消息推送](#)

[附录: 1. 智能设备Profile定义规格](#)

名词定义

- `SmartBox` 智能主机，安装在业主家庭，用于控制业务房间的各种智能化传感设备，例如：空调、新风、地热、灯光等。
- `SmartServer` 云端接入smartbox的通信服务器
- `McuController` 智能主机设备上的外围传感器控制模块，与ARM系统通过UART通信。

说明

系统特点：

错误码

```
#define Error_NoError 0
#define Error_UnknownError 1    // 未定义
#define Error_SystemFault 1001  // 系统错误
#define Error_TokenInvalid 1002 // 令牌错误
#define Error_AccessDenied 1003 // 访问受限
#define Error_PermissionDenied 1004 // 权限受限
#define Error_ParameterInvalid 1005 // 参数无效
#define Error_PasswordError 1006    // 密码错误
#define Error_UserNotExist 1007     // 用户不存在
#define Error_ObjectHasExist 1008   // 对象已存在
#define Error_ObjectNotExist 1009   // 对象不存在
#define Error_ResExpired 1010       // 资源过期
#define Error_ReachLimit 1011       // 达到上限

#define Error_DeviceServerNotFound 2001 //接入服务器未配置
#define Error_DeviceNotOnline 2002 //设备未在线
```

Http 返回消息格式

```

{
    status: 0 ,    // 0 : 正常 ,   1 :异常
    errcode: 0,    // 错误码
    errmsg: '' ,   // 错误信息
    result: {}     // 返回的数据对象, dict/array/object/简单数据类型(string,int,float)
}

```

TCP 消息封包

采用tcp进行消息的传输控制，连续的消息包之间以分隔符 `SP='\0'` 进行分隔。

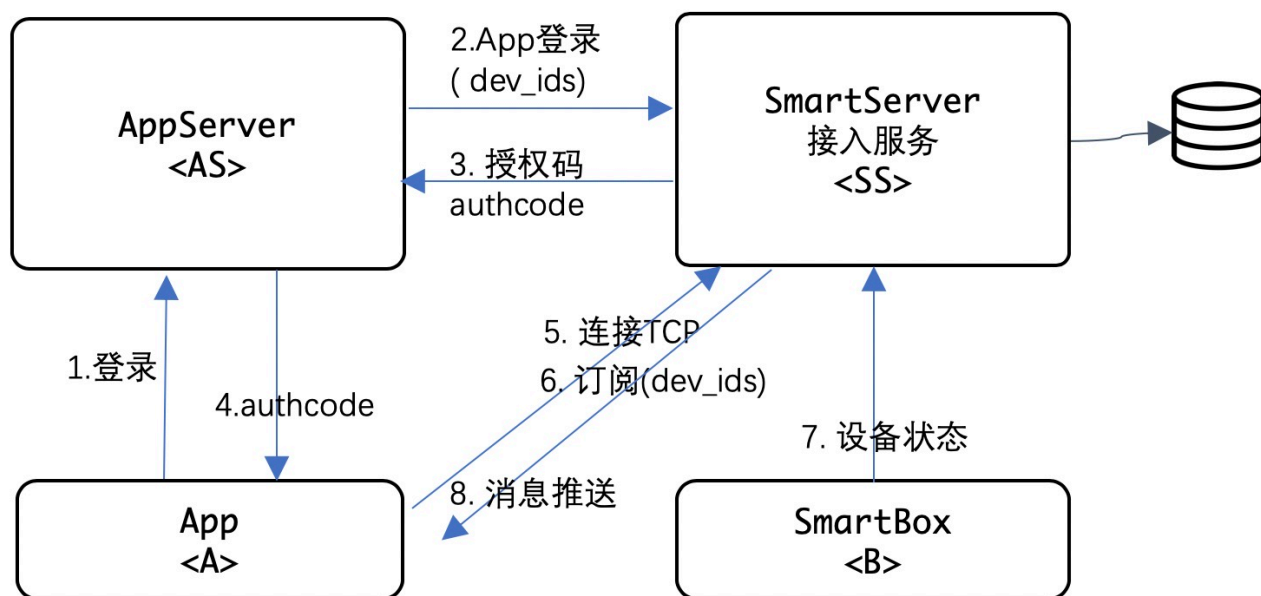
Pack1	SP	Pack2	SP	Pack n
{..}	\0	{..}	\0	{..}

```

{ "id":    "10001","name": "call_in","values": {} }
\0
{ "id":    "10002","name": "call_keep","values": {} }
\0
{ "id":    "10003","name": "call_end","values": {} }

```

交互流程



App登录AS系统，AS从SS服务器获取本次App登录的设备访问授权码(authcode)。随后App建立到SS

的Tcp连接，发送设备订阅请求之后，等待接收设备smartbox上行的设备状态消息。

- A - 移动app程序
- AS - app的服务端程序（张磊提供）
- SS - 设备接入服务器
- B - 终端智能主机

1. REST 设备接口

1.1 主机状态查询

名称:

getDeviceStatus(id)

描述

查询设备的当前运行状态信息，包括当前所有连接主机的家庭智能设备的状态值。

Request

```
URL: /api/smartbox/status
Method: GET
Headers:
Character Encoding: utf-8
Content-Type: x-www-form-urlencoded
Query Parameters:
  - authcode  授权码
  - id        设备编号
```

Response

Headers:

Character Encoding: utf-8

Content-Type: application/json

Data:

- status 状态码 0 : succ; others : error
- errcode 错误码
- errmsg 错误信息
- result
 - id 设备编号
 - time 当前时间
 - host_ver 软件版本号
 - mcu_ver 软件版本号
 - boot_time 启动时间
 - ips 当前设备ip地址 多个ip以 , 分隔
 - loginserver_url 登录服务器地址
 - alarm_enable 启用报警
 - watchdog_enable 启用看门狗
 - sensors (array)
 - sensor_type 端设备类型
 - sensor_id 端设备编号
 - p(1..n) 端设备当前运行参数值

Examples

```
{
  status:0,
  result:{
    id: 'FB000278000000000001',
    time: 1500092322,
    host_ver: 1.0.1,
    mcu_ver: 1.0.0 ,
    boot_time: 150000765,
    ips: '192.168.0.101,11.0.0.11 ',
    loginserver_url : 'svr1.smarthome.com',
    alarm_enable : 1
    watchdog_enable : 1
    sensors : [
      { sensor_type: 5 ,  sensor_id: 1, s:'on' , p: 1 },
      { sensor_type: 9 ,  sensor_id: 1, x:'on' , y: 1 },
      ....
    ]
  }
}
```

Remarks

1.2 主机参数设置

名称:

setSmartBoxParams()

描述

设置主机运行参数

Request

```
URL: /api/smartbox/params
Medthod: POST
Headers:
Character Encoding: utf-8
Content-Type: x-www-form-urlencoded
Query Parameters:
Body:
  - authcode 授权码
  - id 设备编号
  - name 参数名称
  - value 参数值
  - module 设备模块编号 0 : arm , 1:mcu

** 支持参数(name) **
watchdog_enable 是否启用看门狗 0 :close , 1: open
alarm_enable 是否启用报警功能 0: close , 1: open
reboot 设备重启等待时间 0: 即刻 , n : 推迟秒数
save 设置参数保存 0:不保存 (默认) , 1:保存
```

Response

```
Headers:
Character Encoding: utf-8
Content-Type: application/json
Data:
  - status 状态码 0 : succ; others : error
  - errcode 错误码
  - errmsg 错误信息
  - result
```

Remarks

1.3 端设备状态值查询

名称:

getSensorStatusValue()

描述

查询某一个端设备的当前运行值。端设备以 类型(sensor type)+编号(sensor id)做唯一识别。每个端设备具有多项控制和参数值。

查询直接返回最近一次端设备的运行状态，并下发一次设备的查询，后续设备将通过tcp通道反馈状态消息到app。

Request

```
URL: /api/smartbox/sensor/status
Method: GET
Headers:
Character Encoding: utf-8
Content-Type: x-www-form-urlencoded
Query Parameters:
  - authcode    授权码
  - id          设备编号
  - sensor_type 端设备类型  0 : 表示查询所有的端设备状态
  - sensor_id   端设备编号
```

Response

```
Headers:
Character Encoding: utf-8
Content-Type: application/json
Data:
  - status  状态码  0 : succ; others : error
  - errcode 错误码
  - errmsg  错误信息
  - result
    - device_id    设备编号
    - sensor_type  端设备类型
    - sensor_id    端设备编号
    - p (1..n) 多个端设备的状态值
```

Examples

```
{
  status:0,
  result:{
    sensor_type: 5 ,
    sensor_id: 1,
    s:'on' ,
    p: 1
  }
}
```

Remarks

1.4 端设备参数设置(控制)

名称:

setSensorValue()

描述

下发对端设备的命令控制

Request

```
URL: /api/smartbox/sensor/params
Method: POST
Headers:
Character Encoding: utf-8
Content-Type: x-www-form-urlencoded
Query Parameters:
Body:
  - authcode    授权码
  - id          设备编号
  - sensor_type 端设备类型
  - sensor_id   端设备编号
  - name        参数名称
  - value       参数值
```

Response


```
Headers:
Character Encoding: utf-8
Content-Type: application/json
Data:
  - status    状态码 0 : succ; others : error
  - errcode   错误码
  - errmsg    错误信息
  - result
```

Remarks

1.5 获取设备profile信息

名称:

getProfile()

描述

App第一次连接到智能设备需查询获取设备profile信息来呈现房间智能设备的拓扑

Request

```
URL: /api/smartbox/profile
Method: GET
Headers:
Character Encoding: utf-8
Content-Type: x-www-form-urlencoded
Query Parameters:
  - authcode   授权码
  - id         设备编号
```

Response

Headers:
Character Encoding: utf-8
Content-Type: application/json
Data:
- status 状态码 0 : succ; others : error
- errcode 错误码
- errmsg 错误信息
- result
- 参见 [附录.profile] 定义细节

Remarks

1.6 获取推送服务器地址

名称:

getPushServer ()

描述

app登录设备接入服务器，获取推送服务器地址

Request

URL: /api/smartbox/pushserver
Method: GET
Headers:
Character Encoding: utf-8
Content-Type: x-www-form-urlencoded
Query Parameters:
- authcode 授权码，由app的管理平台获取（测试阶段可以使用固定码）

Response

Headers:
Character Encoding: utf-8
Content-Type: application/json
Data:
- status 状态码 0 : succ; others : error
- errcode 错误码
- errmsg 错误信息
- result
- server_host 状态推送服务器地址
- server_port 状态推送服务器端口

Remarks

APP登录成功之后，主动连接到Server的流推送端口(host/port),发送设备订阅消息，等待服务器发送的设备状态消息。

1.7 获取设备列表

名称:

getDeviceList ()

描述

app查询设备列表。

Request

URL: /api/smartbox/list
Method: GET
Headers:
Character Encoding: utf-8
Content-Type: x-www-form-urlencoded
Query Parameters:
- authcode 授权码，由app的管理平台获取（测试阶段可以使用固定码）

Response

Headers:
Character Encoding: utf-8
Content-Type: application/json
Data:
- status 状态码 0 : succ; others : error
- errcode 错误码
- errmsg 错误信息
- result [array]
- id 设备编号
- name 设备名称

Examples

```
{  status:0 ,
  errcode:0,
  result: [
    {id:"F00991001" , name: "凌云雅阁"},
    {id:"F00991021" , name: "辞别丽宫"}, ...
  ]
}
```

Remarks

1.8 App登录请求授权码

名称:

getAppAuthCode ()

描述

app登录时，appserver 请求 smartserver生成授权码。

Request

URL: /api/smartserver/authcode
Method: POST
Headers:
Character Encoding: utf-8
Content-Type: application/json
Query Parameters:
Body:
- app_id app在平台的应用标识（唯一）
- devices [array]
- id 设备编号
- name 设备名称

Examples

```
{ app_id : 'BB777676QWWQQ',  
  devices : [  
    {id: "F00991001" , name: "凌云雅阁"},  
    {id: "F00991021" , name: "辞别丽宫"}  
  ]  
}
```

Response

Headers:
Character Encoding: utf-8
Content-Type: application/json
Data:
- status 状态码 0 : succ; others : error
- errcode 错误码
- errmsg 错误信息
- result 授权码

Examples

```
{ status:0 ,  
  errcode:0,  
  result:'AASDWWWWW00123'  
}
```

Remarks

2. 设备推送消息接口（TCP）

App调用REST登录接口获得接入App的服务器地址和端口，并与其建立Tcp长连接。App为了获得设备推送消息，需在建立连接后发送订阅消息 `MessageAppSubscribe` 到设备服务器。服务器将推送设备消息 `MessageSensorStatus` 到App。

2.1 设备消息订阅

- 名称: **MessageAppSubscribe**
- 方向: **app -> server**
- 参数:
 - authcode 平台授权码
 - ids [array] 设备编号数组

样例

```
{
  "id" : "",
  "name" : "subscribe",
  "values" : {
    "authcode" : "XQ2231NNNMMMSASWQEWW",
    "ids" : [
      "F01000000801" , "F01000000802" , "F01000000803" , ...
    ]
  }
}
```

2.2 取消订阅

- 名称: **MessageAppUnSubscribe**
- 方向: **app -> server**
- 参数:
 - authcode 平台授权码
 - ids [array] 设备编号数组

样例

```
{
  "id" : "",
  "name" : "subscribe",
  "values" : {
    "authcode" : "XQ2231NNNMMMSASWQEEWW",
    "ids" : [
      "F01000000801" , "F01000000802" , "F01000000803" ,...
    ]
  }
}
```

2.3 心跳消息

- 名称: **MessageAppHeartbeat**
- 方向: **app -> server**
- 参数:
 - authcode 平台授权码

样例

```
{
  "id" : "",
  "name" : "subscribe",
  "values" : {
    "authcode" : "XQ2231NNNMMMSASWQEEWW",
  }
}
```

2.4 设备消息推送

- 名称: **MessageSensorStatus**
- 方向: **** server -> app ****
- 参数:
 - `device_id` 设备编号
 - `sensor_type` 端设备类型
 - `sensor_id` 端设备编号
 - `params` [dict] 参数数组

样例

```
{
  "id" : "",
  "name" : "sensor_status",
  "values" : {
    "params" : {
      "1" : "1",
      "2" : "0"
    },
    "sensor_id" : 1,
    "sensor_type" : 2
  }
}
```

附录

1. 智能设备Profile定义规格

```
{
  "id": "9901",
  "name": "house-template-1",
  "version": "1.0.0",
  "author": "scott",
  "project_id": "9999",
  "garden_id" : "home",
  "manufacture": "fanbei-smart",

  "house": {"name": "我的家"},
  "rooms": {
    "A":{"name": "客厅" },
    "B":{"name": "主卧" },
    "C":{"name": "卫生间" }
  },

  "sensor_defines": [
    {
      "name": "灯",
      "type": 2 ,
      "vendor": "Panasonic",
      "model": "YJ-911",
      "features": {
        "switch": {
          "name": "开关",
          "id": 1,
          "value_type": {"name": "enum", "items": ["on","off"] ,"default":"off"},
          "commands": {
            "on": {"name": "开", "value": 1},

```



```

        "off": {"name": "关", "value": 0}
    },
    "brightness": {
        "name": "亮度",
        "id": 2,
        "value_type": {"name": "num", "start": 1, "end": 10, "step": 1, "default":
1},
        "commands": {
            "max": {"name": "最亮", "value": 10},
            "min": {"name": "最暗", "value": 1},
            "up": {"name": "增亮", "value": "u"},
            "down": {"name": "减暗", "value": "d"}
        }
    },
    "color": {
        "name": "颜色",
        "id": 3,
        "value_type": {"name": "num", "start": 1, "end": 3, "step": 1, "default": 1}
    },
    "commands": {
        "red": {"name": "红", "value": 1},
        "green": {"name": "绿", "value": 2},
        "blue": {"name": "蓝", "value": 3},
        "next": {"name": "下一种", "value": "n"},
        "back": {"name": "上一种", "value": "b"}
    }
}

"sensors": [
    {"type": 2, "id": 1, "room_id": "A", "name": "吊灯"},
    {"type": 2, "id": 2, "room_id": "A", "name": "床头灯"}
]
}

```