

# SIWIFI本地接口文档

## 历史版本

| 版本号 | 作者  | 修改日期       | 修改说明                                  |
|-----|-----|------------|---------------------------------------|
| V10 | 常玉成 | 2015-06-29 | 新建                                    |
| V10 | 沈城  | 2016-06-05 | 增加get version 接口                      |
| V10 | 郭喜俊 | 2017-07-03 | 增加检测router wan 状态接口                   |
| V11 | 郭喜俊 | 2017-07-27 | 增加侦测wan type接口                        |
| V11 | 夏勤  | 2018-02-26 | 统一规范                                  |
| V12 | 郭喜俊 | 2018-03-20 | 添加p2p、内部使用、internet默认权限相关接口           |
| V12 | 夏勤  | 2018-04-04 | 添加wifi高级设置，局域网设置接口                    |
| V12 | 沈城  | 2018-04-04 | 添加 定时开关机 根据流量限制设备访问 设备上网行为 关闭硬件恢复出厂设置 |

- [协议中的概念](#)
- [接口](#)
  - [1 路由器管理](#)
    - [1.1 路由器绑定](#)
    - [1.2 路由器解绑](#)
    - [1.3 管理员操作](#)
    - [1.4 路由器绑定普通用户](#)
    - [1.5 普通用户获取访问权限](#)
  - [2. 管理连接设备](#)
    - [2.1 获取设备列表](#)

- 2.2 设置设备相关选项
- 2.3 获取WIFI设备连接提醒
- 2.4 设置WIFI设备连接提醒
- 2.5 获取新设备默认上网权限
- 2.6 设置新设备默认上网权限
- 2.7 设置设备上网行为控制
- 2.8 获取设备上网行为控制设置
- 2.9 设置设备上网流量信息
- 2.10 获取设备上网流量信息
- 3. 路由器信息
  - 3.1 获取路由器token-密码方式
  - 3.2 获取路由器token-远程验证方式
  - 3.3 获取路由器基本信息
  - 3.4 设置路由密码
- 4. WIFI设置
  - 4.1 获取wifi信息列表
  - 4.2 WIFI基本设置
  - 4.3 WIFI高级设置
- 5. 网络设置
  - 5.1 自动检测上网类型
  - 5.2 设置当前外网连接方式
  - 5.3 设置局域网配置
  - 5.4 获取QOS信息
  - 5.5 QOS设置
  - 5.6 网络检测功能
  - 5.7 检测router wan 状态
  - 5.8 侦测wan type
- 6. 系统设置
  - 6.1 系统操作
  - 6.2 路由器初始化信息查询
  - 6.3 OTA升级检测
  - 6.4 OTA升级
  - 6.5 在线日志上传
  - 6.6 路由器本地接口版本查询
  - 6.7 路由器定时开关机
  - 6.8 获取路由器定时开关机设置
  - 6.9 路由器关闭硬件恢复出厂设置
  - 6.10 获取路由器关闭硬件恢复出厂设置
- 7. 家长管控
  - 7.1 设备限速
  - 7.2 设置白名单

- 7.3 获取黑名单
- 7.4 使能黑白名单
- 8. 内部使用接口
  - 8.1 有线设备扫描
  - 8.2 ARP事件处理
  - 8.3 初始化信息
- 9. P2P
  - 9.1 创建新p2p会话
  - 9.2 删除p2p会话
- 10. 其他操作
  - 10.1 测试
- 11. 错误码

## 协议中的概念

### 1. 通讯接口SDK版本

指正式发布的通讯sdk文档版本，app以正式发布的文档为基础完成通讯。

### 2. 协议版本

通讯接口协议版本指当前协议文档中针对每个功能所定义的通讯接口的当前版本号,供app的开发人员参考。后续的开发过程中服务的通讯接口发生变化，会更新对应功能的版本号，同时服务端会向下兼容所有历史版本接口。

### 3. app请求的协议版本

app请求协议版本指当前app开发所参考的协议文档中定义的具体功能的协议版本，如果请求的版本当前路由器或者服务不支持，会返回not support。

### 4. 交互方式

app与服务器、app与路由器按照HTTP的请求方式进行通讯，全部以http post的形式交互。

### 5. 接口

接口即为请求的url,是服务端定义的相应功能的实现接口。需要注意的是完整的url需要根据当前请求是本地请求还是远程操作，具体组织url的方式后续会给出。

### 6. 参数

所有的参数以字符串的形式提供。

http post形式的操作接口将参数放在http请求的包体里面。

必选参数指必须提交的参数，否则服务端会返回错误。

所有的请求中协议版本参数是必选的。

非必选的参数依据客户端的需求实际调整，相应服务端的结果也会有调整。

### 7. 返回

所有服务端返回的数据以json格式封装。

所有的返回值都会有code字段，code为0代表请求成功执行。code不为0时返回值中会带有msg信息，表示服务端返回的错误或者提示信息。

## 接口

### 1 路由器管理

#### 1.1 路由器绑定

功能：app通过局域网绑定路由器,用户为管理员

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfssystem/bind

说明：该接口适用于未绑定的路由器，绑定的同时路由器会触发建表的动作。

e.g.:

```
body:
  {
    "version"(必选)://代表当前app请求的协议版本
    "userobjectid"(可选)://代表当前需要绑定的user的objectid，作为唯一标示
    "deviceid"(可选)://代表当前需要绑定的user的deviceid，作为唯一标示
  }
return:
  {
    "code": 0, //返回码
    "routerobjectid": "df78361fh2", //返回路由器的bmob objectid
    "msg": "error message" //返回码不为0时会有错误信息返回
  }
```

#### 1.2 路由器解绑

功能：app通过局域网解除绑定路由器

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfssystem/unbind

说明：解除绑定成功返回code == 0

e.g.:

```

body:
{
  "version"(可选)://代表当前app请求的协议版本
  "userobjectid"(可选)://代表当前需要绑定的user的objectid, 作为唯一标示
  "deviceid"(可选)://代表当前需要绑定的user的deviceid, 作为唯一标示
}
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 1.3 管理员操作

功能：管理员操作

方式：APP + 路由器/外网服务器以http post的形式提交操作

接口：/api/sfsystem/manager

说明：该操作接口同时支持远程/本地操作userid和phonenummer二者必选其一

e.g.:

```

body:
{
  "version"(可选)://代表当前app请求的协议版本
  "action"(可选):
    0表示邀请管理员
    1表示删除管理员
    2表示接受邀请
    3表示拒绝邀请
    4 表示退出分享
    5 表示取消分享
    6 表示添加管理员(微信分享使用)
    7 表示删除管理员(微信分享使用)
  "userid"(可选)://代表目标用户的objectid, 作为唯一标示
    邀请/删除为对方的user objectid
    接受/拒绝/退出则为自身的user objectid
    其中action=1/2/时userid必选不能为空
  "phonenummer"(可选)://表示被操作的管理员的注册phone
  "username"(可选)://表示被操作的管理员的user name
  "tag"(可选)://填写收到服务端发送的邀请消息时收到的tag, 详细见RouterMessage中管理员消息
    action=2 或者action =3时必选
  "managerid"(可选):"903182909672783873"//参数为分享者的用户Id
    在action 为 6或7时必选
}
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 1.4 路由器绑定普通用户

功能: app通过局域网绑定路由器,用户为普通用户

方式: APP + 路由器以http post的形式提交操作

接口: /api/sfsystem/adduser

说明: 该接口适用于未绑定的路由器, 绑定的同时路由器会触发建表的动作。

e.g.:

```
body:
{
  "version"(必选)://代表当前app请求的协议版本
  "userobjectid"(可选)://代表当前需要绑定的user的objectid, 作为唯一标示
  "deviceid"(可选)://代表当前需要绑定的user的deviceid, 作为唯一标示
  "rootmessage"(可选)://代表当前绑定操作是否需要通知管理员
}
return:
{
  "code": 0, //返回码
  "routerobjectid": "df78361fh2", //返回路由器的bmob objectid
  "msg": "error message" //返回码不为0时会有错误信息返回
}
```

## 1.5 普通用户获取访问权限

功能: app通过wifi密码获取路由器普通用户权限

方式: APP + 路由器以http post的形式提交操作

接口: /api/sfsystem/getaccess

说明: 该接口用于普通用户获取访问权限

e.g.:

```
body:
{
  "version"(必选)://代表当前app请求的协议版本
  "data"(必选)://代表加密过后的pattern和wifi密码数据
}
return:
{
  "code": 0, //返回码
  "access": "*****", //返回路由器的访问权限pattern
  "msg": "error message" //返回码不为0时会有错误信息返回
}
```

## 2. 管理连接设备

### 2.1 获取设备列表

功能: 获取路由器上所有连接设备信息, 包括在线、离线

方式: APP + 路由器以http post的形式提交操作

接口: /api/sfsystem/device\_list

说明：支持版本 V9

e.g.:

```
body:
{
  "version"(必选)://代表当前app请求的协议版本
  "type"(必选)://代表了查询的类型
    0 获取所有连接设备信息，包括在线和离线的
    1 获取所有在线设备信息
    2 仅获取在线设备数量
    3 获取单个设备的信息
    type参数默认值为0
  "mac"(可选)://查询单个设备时有效，即(type 3)时按照mac地址查询
}
//when type 0 or 1
return:
{
  "list": [
    {
      "hostname": "android-f77e95d4560d37af", //设备实际接入名称
      "nickname": "Zhanggong", // 修改备注后名称
      "mac": "CC:2A:60:D0:62:EE", //设备mac地址
      "online": 0, //是否在线
      "ip": "191.167.30.237", //设备ip地址，离线则返回0.0.0.0
      "port": 0, //设备连接类型，1为有线,0为无线
      "dev": "wlan0", //连接路由器网络端口名称
      "count": //MB 为单位， 返回剩余可用流量 -1 没有限制
      "restrictenable": //是否使能 访问类型限制
      "usageenable": //是否使能访问流量限制

      "authority": { //设备权限相关信息
        "internet": 0, //互联网访问模式 0禁止 1允许 2限时模式
        "lan": 0, //是否允许接入路由器
        "listfunc" : 0 //0 no use 1 使用白名单 2 使能黑名单
        "speedlimit" : 0 //是否开启了限速功能
        "notify": -1, //上线离线通知
        "speedlvl": -1, //网速优先级别
        "disk": -1, //访问全盘数据()
        "limitup": -1 //网速上行限制，单位KBytes/s
        "limitdown": -1 //网速下行限制，单位KBytes/s
      },

      "timelist": [
        {
          "enable": [1/0], //时间段是否禁止访问
          "starttime": "xx:xx",
          "endtime": "xx:xx"
          "week": "1,7"
          "config": 1
        },
        ... //多个时间段
      ],

      "speed": { //设备网络速度相关信息
```

```

        "maxdownloadspeed": "76395", //最大下载速度
        "uploadtotal": "1130816", //上传总流量Bytes
        "upspeed": "29", //实时上传速度,单位Bytes/s
        "downspeed": "-1", //实时下载速度.单位Bytes/s
        "online": "19069", //本次连接时间
        "maxuploadspeed": "30289", //最大上传速度
        "downloadtotal": "1612667" //下载总流量Bytes
    }
},
...
], // list would be empty when get fail
"code": 0, //返回码,code为0代表获取数据成功
"msg": "OK" //表示接口执行正常 or return error string
}

```

//when type 3

return:

```

{
    "list":
    {
        "hostname": "android-f77e95d4560d37af", //设备实际接入名称
        "nickname": "Zhanggong", // 修改备注后名称
        "mac": "CC:2A:60:D0:62:EE", //设备mac地址
        "online": 0, //是否在线
        "ip": "191.167.30.237", //设备ip地址, 离线则返回0.0.0.0
        "port": 0, //设备连接类型, 1为有线, 0为无线
        "dev": "wl0", //连接路由器网络端口名称
        "count": //MB 为单位, 返回剩余可用流量 -1 没有限制

        "authority": { //设备权限相关信息
            "internet": 0, //是否运行互联网访问
            "lan": 0, //是否允许接入路由器
            "listfunc" : 0 //0 no use 1 使用白名单 2 使能黑名单
            "speedlimit" : 0 //是否开启了限速功能
            "notify": -1, //上线离线通知
            "speedlvl": -1, //网速优先级别
            "disk": -1, //访问全盘数据()
            "limitup": -1 //网速上行限制, 单位KBytes/s
            "limitdown": -1 //网速下行限制, 单位KBytes/s
        },

        "timelist": [
            {
                "enable": [1/0], //时间段是否禁止访问
                "starttime": "xx:xx",
                "endtime": "xx:xx",
                "config": 1
            },
            ... //多个时间段
        ],

        "speed": { //设备网络速度相关信息
            "maxdownloadspeed": "76395", //最大下载速度
            "uploadtotal": "1130816", //上传总流量Bytes
            "upspeed": "29", //实时上传速度,单位Bytes/s

```



```
        "downspeed": "-1", //实时下载速度.单位Bytes/s
        "online": "19069", //本次连接时间
        "maxuploadspeed": "30289", //最大上传速度
        "downloadtoal": "1612667" //下载总流量Bytes
    }
}
"code": 0, //返回码,code为0代表获取数据成功
"msg": "OK" //表示接口执行正常 or return error string
}

//when type 2
return:
{
    "online": 1
}
```

## 2.2 设置设备相关选项

功能：设置连接设备的各类信息

方式：**APP + 路由器**以http post的形式提交操作

接口：**/api/sfsystem/setdevice**

说明：支持版本 V12,所有非必选参数均可单独设置

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "mac"(必选)://代表要设置的设备mac地址，以此为索引
  "internet"://代表是否运行互联网访问，取值1表示可以访问，否则不能访问
  "lan"://代表是否允许接入lan口，取值1表示允许访问，否则不能访问
  "notify"://代表是否接收上线离线通知，取值1表示接收，否则不接收
  "disk"://代表能否访问路由器硬盘，取值1表示可以访问，否则不能访问
  "nickname"://代表当前设备的别名
  "timelist": [ // set timelist, only available when internet is 1
    {
      "enable": [0/1], //时间段是否禁止访问
      "starttime": "xx:xx",
      "endtime": "xx:xx",
      "week": "1,2,3,4,5,6,7",
      "config":1 // config 为1 代表储存配置使用
    },
    {
      "enable": 1, //时间段是否禁止访问
      "starttime": "xx:xx",
      "endtime": "xx:xx",
      "week": "1,2,3,4,5,6,7",
      "config":0 // config为0代表配置程序使用
    },
    ... //多个时间段
  ]
}
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 2.3 获取WIFI设备连接提醒

功能：获取wifi设备连接提醒

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/get\_wifi\_filter

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
}
return:
{
  "enable": 0, //代表wifi陌生设备的连接提醒使能开关,1表示开启, 0表示关闭
  "mode": 0, //代表当前推送的模式,0表示自动模式 1表示晚间不会推送
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 2.4 设置WIFI设备连接提醒

功能：陌生设备连接提醒，设置开关和推送时间

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/set\_wifi\_filter

e.g.:

```
body:
  {
    "version"(必选)://代表当前app请求的协议版本
    "enable"(必选)://代表wifi陌生设备的连接提醒使能开关，1表示开启，0表示关闭
    "mode"://代表当前推送的模式，目前两种模式
      //0表示自动模式 1表示晚间不会推送
  }
return:
  {
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
  }
```

## 2.5 获取新设备默认上网权限

功能：获取新设备默认上网权限

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/getdefault

e.g.:

```
body:
  {
    "version"(必选)://代表当前app请求的协议版本
  }
return:
  {
    "internet"://1表示允许上网，0表示禁止上网
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
  }
```

## 2.6 设置新设备默认上网权限

功能：设置新设备默认上网权限

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/setdefault

e.g.:

```
body:
  {
    "version"(必选)://代表当前app请求的协议版本
    "internet"(必选)://1表示允许上网，0表示禁止上网
  }
return:
  {
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
  }
```

## 2.7 设置设备上网行为控制

功能：设置设备默认上网行为，限制社交/视频/游戏

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/setdevicerestrict

e.g.:

```
body:
  {
    "version"(必选)://代表当前app请求的协议版本
    "mac"(必选)://设备mac地址
    "social"(必选)://0表示允许，1表示禁止
    "video"(必选)://0表示允许，1表示禁止
    "game"(必选)://0表示允许，1表示禁止
    "restrictenable"(必选): //是否启动此功能
  }
return:
  {
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
  }
```

## 2.8 获取设备上网行为控制设置

功能：获取设置设备默认上网行为，限制社交/视频/游戏设置

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/getdevicerestrict

e.g.:

```

body:
    {
        "version"(必选)://代表当前app请求的协议版本
        "mac"(必选)://设备mac地址
    }
return:
    {
        "social"(必选)://0表示允许, 1表示禁止
        "video"(必选)://0表示允许, 1表示禁止
        "game"(必选)://0表示允许, 1表示禁止
        "restrictenable"(必选): //是否启动此功能

        "code": 0, //返回码
        "msg": "error message" //返回码不为0时会有错误信息返回
    }

```

## 2.9 设置设备上网流量信息

功能：设置设备上网流量，可以已一次性或者周期方式配置流量, 需要支持多条设置

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/setdevicedatausage

e.g.:

```

body:
    {
        "version"(必选)://代表当前app请求的协议版本
        "mac"(必选)://设备mac地址
        setlist:[ {
            "count"(必选)://以MB为单位 设备可以使用流量, 可以是赋值
            "type"(必选)://1表示一次性流量, 2表示周期流量 0 表示关闭此功能
            "week"(可选):"1,7":// 周期
        },
        ...
    ]
        "change"(必选):// 以MB为单位 设备可以使用流量, 可以是赋值
        "usageenable"://是否使能
    }
return:
    {
        "code": 0, //返回码
        "msg": "error message" //返回码不为0时会有错误信息返回
    }

```

## 2.10 获取设备上网流量信息

功能：返回设备流量配置

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/getdevicedatausage

e.g.:

```

body:
  {
    "version"(必选)://代表当前app请求的协议版本
    "mac"(必选)://以MB为单位 设备可以使用流量
  }
return:
  {
    "code": 0,//返回码
    "setlist":[ {
      "count"(必选)://以MB为单位 设备可以使用流量,可以是赋值
      "type"(必选)://1表示一次性流量, 2表示周期流量 0 表示关闭此功能
      "week"(可选):"1,7":// 周期
    },
    ...
  ]
  }
  "credit":// 当天剩余流量
  "usageenable"://是否使能
  "msg": "error message"//返回码不为0时会有错误信息返回
}

```

## 3. 路由器信息

### 3.1 获取路由器token-密码方式

功能：获取路由器session token信息，作为后续交互中的参数传递

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/get\_stok\_local

说明：该接口的使用前提是知道后台管理密码

e.g.:

```

body:
  {
    "version"(必选)://代表当前app请求的协议版本
    "password"(必选)://后台管理密码
  }
return:
  {
    "code": 0,//返回码
    "msg": "error message"//返回码不为0时会有错误信息返回
    "stok": "7b358734a72a0873b4ce4317e6de3e1f"//路由器的stok
  }

```

### 3.2 获取路由器token-远程验证方式

功能：获取路由器session token信息，作为后续交互中的参数传递

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/get\_stok\_remote

说明：用户登录远程服务器后，会返回sessiontoken用于路由器一端的验证，如果传递sessiontoken

的话，路由器会去服务端验证传递的sessionToken是否有效，userid用于唯一标示一个用户

e.g.:

```
body:
  {
    "version"(必选)://代表当前app请求的协议版本
    "userid"(必选)://远程账户的Objectid
    "sessiontoken"(必选)://登录远程服务器后返回的sessiontoken
  }
return:
  {
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
    "stok": "7b358734a72a0873b4ce4317e6de3e1f" //路由器的stok
  }
```

### 3.3 获取路由器基本信息

功能：查询路由器状态

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/main\_status

说明：用户登录远程服务器后，会返回sessiontoken用于路由器一端的验证，如果传递sessiontoken的话，路由器会去服务端验证传递的sessionToken是否有效，userid用于唯一标示一个用户

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "querycpu"://代表是否查询cpu状态, 取值1表示查询, 否则无效
  "querymem"://代表是否查询memory状态, 取值1表示查询, 否则无效
}
return:
{
  "status": 0,//路由器状态
    //0 正常运行
    //1 升级中
    //2 重启中
    //3 重置中
  "upspeed": 7924,//路由器上传速率, 单位Bytes/s
  "downspeed": 7924,//路由器下载速率, 单位Bytes/s
  "devicecount": 1,//当前连接设备数目
  "useable space": 1025140,//路由器可用存储空间, 单位Bytes
  "downloadingcount": 1,//当前下载任务个数
  "cpuload": 5,//cpu负载, 以1%为单位
  "memoryload": 25,//memory负载, 以1%为单位
  "code": 0,//返回码
  "msg": "error message"//返回码不为0时会有错误信息返回
}

```

### 3.4 设置路由密码

功能: 设置路由器管理密码

方式: APP + 路由器以http post的形式提交操作

接口: /api/sfsystem/setpasswd

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "oldpwd"(必选)://加密后的旧密码
  "newpwd"(必选)://加密后的新密码
}
return:
{
  "code": 0,//返回码
  "msg": "error message"//返回码不为0时会有错误信息返回
}

```

## 4. WIFI设置

### 4.1 获取wifi信息列表

功能: 获取所有wifi信息, 包括2.4G、5G、访客wifi

方式: APP + 路由器以http post的形式提交操作



接口: /api/sfsystem/wifi\_detail

说明: 返回结果以json数组的形式提供, 访客wifi暂时不支持

e.g.:

```
body:
  {
    "version"(必选): //代表当前app请求的协议版本
  }
return:
  {
    "info": [
      {
        "ifname": "wl1", //端口设备名称
        "ssid": "xiaomi-zhongji", //ssid名称
        "enable": 1, //是否打开
        "encryption": "mixed-psk", //加密方式
        "signal": 1, //信号强度
        "password": "1928374655", //无线密码, 加密
        "channel": 3, //无线信道
        "band": "2.4G", //无线带宽, 2.4G或者5G
        "mac": "00:16:88:43:8B:DF" //wifi的mac地址
      }
    ],
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
  }
```

## 4.2 WIFI基本设置

功能: 统一提交所有wifi的信息

方式: APP + 路由器以http post的形式提交操作

接口: /api/sfsystem/setwifi

说明: 参数中ParseError: KaTeX parse error: Expected 'EOF', got '填' at position 7: (ssid)填写需要修改参数的wifi热点的...(ssid).enable字段的设置

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "setting"(必选): [ //WIFI设置
  {
    "$(ssid).enable"://其中$(ssid)代表需要修改参数的wifi热点的ssid值代表该ssid代表的wi
    "$(ssid).ssid"://代表该ssid的名称, 取值为字符串
    "$(ssid).encryption"://代表该wifi热点的加密方式, 取值为常见的加密方式{psk+ccmp/psk:
    "$(ssid).signalmode"://代表该wifi热点的信号强度, 三种模式
                        0—省电模式
                        1—标准模式
                        2—穿墙模式
    "$(ssid).password"://代表该wifi热点的访问密码, 以加密方式传输
    "$(ssid).channel"://代表该wifi热点的信道, 2.4G和5G的信道取值范围不一样

  }
  ],
  "disableall"(可选)://打开/关闭所有wifi 取值0表示打开所有wifi, 取值1表示关闭所有wifi
}
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 4.3 WIFI高级设置

功能: 统一提交所有wifi的信息

方式: APP + 路由器以http post的形式提交操作

接口: /api/sfsystem/setwifi\_advanced

说明: 参数中 **ParseError: KaTeX parse error: Expected 'EOF', got '填' at position 7: (ssid)填写需要修改参数的wifi热点的...(ssid).enable**字段的设置

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "setting"(必选): [ //WIFI设置
  {
    "${ssid}.enable"://其中$(ssid)代表需要修改参数的wifi热点的ssid值代表该ssid代表的wi
    "${ssid}.encryption"://代表该wifi热点的加密方式,取值为常见的加密方式{psk+ccmp/psk:
    "${ssid}.channel"://代表该wifi热点的信道,2.4G和5G的信道取值范围不一样
    "${ssid}.signalmode"://代表该wifi热点的信号强度, 三种模式
                        0—省电模式
                        1—标准模式
                        2—穿墙模式
    "${ssid}.bandwidth"://带宽, 三种模式
                        2.4G    0—20MHz      5G      0—20MHz
                        1—40MHz      1—40MHz
                        2—20/40MHz    3—80MHz

    "${ssid}.country"://国家码 list_value:CN,00,AD,AE,AF,AG,AI,AL,AM,AN...
  }
],
}
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 5. 网络设置

### 5.1 自动检测上网类型

功能：获取当前外网连接方式

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/get\_wan\_type

说明：为了保持一定的扩展性，type为不同类型时返回的json结果中部分字段会为空

e.g.:

```
body:
{
  "version"(必选)://代表当前app请求的协议版本
}
return:
{
  "type": 0,//网络连接类型, 0,1,2分别代表了DHCP、PPPOE、STATICIP
  "pppname": "wxy00009001",//PPPOE帐号, type为pppoe时可用
  "ppppwd": "123456",//PPPOE密码, type为pppoe时可用
  "ip": "192.168.1.10",//静态ip地址, type为staticip时可用
  "mask": "255.255.255.0",//代表设置的子网掩码, type为staticip时可用
  "autodns": 0,//代表当前dns的获取方式, 0表示手动, 1表示自动
  "dns1": "192.168.1.1",//代表配置为手动模式的首选dns
  "dns2": "192.168.2.1",//代表配置为手动模式的备选dns
  "gateway": "192.168.12.1",//代表配置为静态ip时的网关, type为staticip时可用
  "code": 0,//返回码
  "msg": "error message",//返回码不为0时会有错误信息返回
}
```

## 5.2 设置当前外网连接方式

功能：设置外网的连接方式

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/set\_wan\_type

说明：为了保持一定的扩展性，设置的参数比实际app的使用方式要多，app可以根据实际要设置的内容动态调整

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "type"(必选)://代表设置的外网连接方式, 0,1,2分别代表了DHCP、PPPOE、STATICIP
  当type==0//DHCP方式, 有如下可选参数
    autodns//代表当前dns的获取方式, 0表示手动, 1表示自动
    dns1//代表配置为手动模式的首选dns
    dns2//代表配置为手动模式的备选dns
  当type==1//PPPOE模式, 有如下可选参数
    pppname//PPPOE帐号
    ppppwd//PPPOE密码
    autodns//代表当前dns的获取方式, 0表示手动, 1表示自动
    dns1//代表配置为手动模式的首选dns
    dns2//代表配置为手动模式的备选dns
  当type==2//STATIC方式, 有如下参数
    address//代表设置的静态ip地址
    mask//代表设置的子网掩码
    gateway//代表设置的网关
    dns1//代表配置的首选dns
    dns2//代表配置的备选dns
}
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 5.3 设置局域网配置

功能：设置外网的连接方式

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/set\_lan

说明：为了保持一定的扩展性，设置的参数比实际app的使用方式要多，app可以根据实际要设置的内容动态调整

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "address"(可选)://代表配置LAN口IP
  "mtu(可选)"://代表设置的mtu
  "dynamic_dhcp"(可选)://1表示启用DHCP Server，0表示禁用DHCP Server
  "dhcpstart"(可选)://0~256 表示分配地址池起始地址
  "dhcpend"(可选)://0~256 表示分配地址池结束地址
  "leasetime(可选)"://12h 表示分配地址有效时间
}
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 5.4 获取QOS信息

功能：获取qos相关信息

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/qos\_info

说明：支持版本 V12，此接口依赖于和国云合作

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
}
return:
{
  "enable": 0, //代表当前qos功能是否开启，1表示开启，0表示关闭
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 5.5 QOS设置

功能：设置QOS的模式

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/qos\_set

说明：支持版本 V12

e.g.:

```
body:
  {
    "version"(必选)://代表当前app请求的协议版本
    "enable"://代表当前qos功能是否开启，1表示开启，0表示关闭
  }
return:
  {
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
  }
```

## 5.6 网络检测功能

功能：检测当前路由器的状态以及一些系统性的风险

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/netdetect

说明：若某项返回值为-1则表明该部分程序执行异常

e.g.:

```

body:
{
  "version"(可选): //代表当前app请求的协议版本
  "nobandwidth"(可选): //若nobandwidth = 1, 则不返回bandwidth{}, 否则计算bandwidth,
}
return:
{
  "wanlink": 1, //网口是否连接, 0 表示断开, 1 表示连接
  "wifi": [ //wifi 密码检测返回状态
    {
      "band" : "2.4G", //wifi频段
      "strong": 0, //wifi 密码强度, 0 表示强度低, 1 表示正常
      "same": 1, //wifi 密码和管理密码相似度, 0 表示正常, 1 表示相同
    },
    {
      "band" : "5G", //wifi频段
      "strong": 0, //wifi 密码强度, 0 表示强度低, 1 表示正常
      "same": 1, //wifi 密码和管理密码相似度, 0 表示正常, 1 表示相同
    },
  ],
  "dns": 1, //代表当前路由器 dns 解析是否正常, 0表示失败 1表示解析正常, 2表示解析超时
  "memoryuse": 40, //memory 使用占比, 以 1%为单位
  "cpuuse": 40, //cpu 使用占比, 以 1%为单位
  "wanspeed": {
    "upsped": 7924 //wan口上行速率, 单位 Bytes/s
    "downspeed": 7924 //wan口 下行速率, 单位 Bytes/s
  },
  "delay": 158 // ping 百度延时, 单位为ms
  "bandwidth": { //检测当前网络带宽
    "upbandwidth": 4096 // 上行带宽, 单位Kbps
    "downbandwidth": 2048 //下行带宽, 单位Kbps
  }
  "ping": {
    "status": 1, //访问外网是否畅通
    "lost": 0, //丢包率, 以 1%为单位
  },
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 5.7 检测router wan 状态

功能: 检测router wan 状态

方式: APP + 路由器以http post的形式提交操作

接口: /api/sfsystem/check\_net

说明: 100 Internet ok

110 link off

120 no ip

130 no gateway

131 gateway is invalid



140 no dns

141 dns is invalid

e.g.:

```
body:
  {
    "version"(必选)://代表当前app请求的协议版本
  }
return:
  {
    "status": 100 ,//状态码
    "code": 0, //返回码
    "msg": "error message"//返回码不为0时会有错误信息返回
  }
```

## 5.8 侦测wan type

功能：侦测wan type

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/detect\_wan\_type

说明：wantype:

pppoe

dhcp

none(建议当返回none时，提示用户检测WAN口网线是否连接正常)

e.g.:

```
body:
  {
    "version"(必选)://代表当前app请求的协议版本
  }
return:
  {
    "wantype": pppoe ,//网络类型
    "code": 0, //返回码
    "msg": "error message"//返回码不为0时会有错误信息返回
  }
```

## 6. 系统设置

### 6.1 系统操作

功能：app通过局域网解除绑定路由器

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/command

说明：

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "cmd"(必选)://代表当前cmd的命令ID 0表示重启 1表示关闭路由器 2表示恢复出厂设置
  "data"://代表当前命令的额外参数，以字符串形式提供，目前没有使用
}
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 6.2 路由器初始化信息查询

功能：查询路由器的初始化状态，包括是否绑定过，是否设置过密码

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/get\_bindinfo

说明：本接口查询无须密码验证或者Session认证，所以节点是在visitor子目录下

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
}
return:
{
  "passwdset": 0 , //密码是否为空，0表示为空，1表示不为空
  "bind": 0, //路由器是否绑定过，0表示未绑定，1表示绑定
  "binderid": "ae782334", //当前绑定者的user objectid
  "routerid": "98344-2", //当期路由器的objectid，如果已经绑定过
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 6.3 OTA升级检测

功能：检测当前路由器固件版本,获取版本信息

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/ota\_check

说明：如果检测ota版本和本地rom版本不一致且otatime比romtime新，则可调用ota升级

e.g.:

```

body:
{
  "version"(必选): //代表当前app请求的协议版本
}
return:
{
  "romversion": "2.5.38" ,//当前rom版本
  "romtime": "2015-06-25 15:37:25",//当前rom发布时间
  "otaversion": "2.5.39",//ota版本
  "otatime": "2015-07-25 15:37:25",//ota版本发布时间
  "size": 50000, //ota文件大小, 以Bytes为单位
  "type": 0, //0表示release稳定版本, 1表示release开发版本
  "log": "fix problem that...", //ota版本升级日志
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 6.4 OTA升级

功能：OTA升级操作

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/ota\_upgrade

说明：该接口可通过check参数选择进行升级操作或者查询操作升级操作：分为两个步骤，先从服务器下载镜像、再进行烧录

查询操作：查询升级操作的进度，返回升级所处状态等信息

“code”= 0，表示操作正常

“code”=1500，从bmob获取OTA镜像信息失败，退出升级

“code”=1502，本地版本和OTA版本一样，无需进行升级

“code”=1503，已有一个升级操作在执行中，不执行升级

“status”= 0，代表开始进行升级操作

“status”= 1，表示已有OTA升级操作在进行下载文件中

“status”= 2，表示开始烧写固件

“status”= 3，表示镜像下载错误，已停止升级

“status”= 4，表示查询时未进行升级操作

status = 0在调用升级时返回，其余四中状态会在查询操作中返回，烧录过程期间一切app端发送的命令都会被屏蔽。

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "check"(可选)://若check为1时执行查询操作，若为其他或不选则执行升级操作
  "userid"(远程调用时必选):// 调用者的userid
}
return:
{
  "status": 1 //对应查询操作时得到的升级所处状态
  "downloaded": 56 //下载进度的百分比
  "code": 0, //返回码
  "msg": "error message"//返回码不为0时会有错误信息返回
}

```

## 6.5 在线日志上传

功能：将本地路由器系统日志上传到subcloud服务器

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/upload\_log

说明：该接口会将此时的系统日志和内核日志以及系统状态以文件形式上传到subcloud服务器，并命名为useridsyslog.txt, useriddklog.txt和useridRouterStatus.txt.文件名的userid为变量，代指用户id。若获取不到userid，则为命名为UnbindUser。同时会在bmob上的表RouterLog中建立一个对应的object，包含slogurl, klogurl, statusurl, feedback, routerid, useid, romversion等路由相关信息的保存路径若路由器上未绑定，则routerid值为UnbindRouter.

返回码说明：

“code” = 1504 ,文件上传失败

“code” = 1505, 向bmob的RouterLog表中添加对象信息失败

“code”= 1506, 已有一个上传日志命令在执行中，请等待

e.g.:

```

body:
{
  "version"(必选)://代表当前app请求的协议版本
  "feedback"(可选)://表示用户输入的反馈信息
  "nolog"(可选)://用来选择是否上传日志。若无该参数或该参数为其他值，则上传日志。若为1，！
}
return:
{
  "objectid": "81d8516ccf"//保存在bmbob上表RouterLog中的objectId，供后台查看使用
  "code": 0, //返回码
  "msg": "error message"//返回码不为0时会有错误信息返回
}

```

## 6.6 路由器本地接口版本查询

功能：查询路由器支持的本地接口版本

方式：APP + 路由器以http get的形式提交操作

接口: /api/visitor/get\_version

说明: 本接口查询无须密码验证或者Session认证, 所以节点是在visitor 子目录下

e.g.:

```
return:
{
  "version": V10 ,//返回版本信息
  "code": 0, //返回码
  "msg": "error message"//返回码不为0时会有错误信息返回
}
```

## 6.7 路由器定时开关机

功能: 设置路由器定时开关机时间

方式: APP + 路由器以http post的形式提交操作

接口: /api/sfsystem/routerlivetime

说明: 本接口支持设置定时开关机

e.g.:

```
body:
{
  "version"(必选)://代表当前app请求的协议版本
  "timelist": [
    {
      "starttime": "xx:xx",// 开机时间
      "endtime": "xx:xx", //关机时间
      "week": "1,7"//循环周期
      "enable":0 // 1开启或者0关闭
    }
  ]
}
return:
{
  "code": 0, //返回码
  "msg": "error message"//返回码不为0时会有错误信息返回
}
```

## 6.8 获取路由器定时开关机设置

功能: 获取路由器定时开关机时间设置

方式: APP + 路由器以http post的形式提交操作

接口: /api/sfsystem/getrouterlivetime

说明: 本接口获取定时开关机设置

e.g.:

```

body:
  {
    "version"(必选)://代表当前app请求的协议版本
  }
return:
  {
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
    "timelist": [
      {
        "starttime": "xx:xx", // 开机时间
        "endtime": "xx:xx", // 关机时间
        "week": "1,7" // 循环周期
        "enable": 0 // 1开启或者0关闭
      }
    ]
  }

```

## 6.9 路由器关闭硬件恢复出厂设置

功能： 设置路由器关闭硬件恢复出厂设置

方式： APP + 路由器以http post的形式提交操作

接口： /api/sfsystem/blockrefactory

说明：

e.g.:

```

body:
  {
    "version"(必选)://代表当前app请求的协议版本
    "block"(必选): // 0 不阻止硬件恢复出厂设置 1 阻止硬件恢复出厂设置
  }
return:
  {
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
  }

```

## 6.10 获取路由器关闭硬件恢复出厂设置

功能： 获取路由器关闭硬件恢复出厂设置

方式： APP + 路由器以http post的形式提交操作

接口： /api/sfsystem/getblockrefactory

说明：

e.g.:

```

body:
{
    "version"(必选)://代表当前app请求的协议版本
return:
{
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
    "block": // 0 不阻止硬件恢复出厂设置 1 阻止硬件恢复出厂设置
}
}

```

## 7. 家长管控

### 7.1 设备限速

功能：设置连接设备的限速信息

方式：以http post的形式提交变更

接口：/api/sfsystem/setspeed

说明：支持版本 V12

```

body:
{
    "version"(必选):
    "mac"(必选):
    "enable"(必选) :    //使能限速 0 关闭 1 开启
    "limitup":    //bind with enable 1 代表网速上行限制，单位KBytes/s -1 不限速
    "limitdown": //bind with enable 1 代表网速下行限制，单位KBytes/s -1 不限速
}

return:
{
    "code": errorcode, //返回错误码
    "msg": "error message" //errorcode不为0时返回错误信息
}

```

### 7.2 设置白名单

功能：设置黑白名单

方式：以http post的形式提交修改

接口：/api/sfsystem/urlist\_set

说明：支持版本 V12

```

body:
    {
        "version"(必选):
        "listtype"(必选): [0/1] //代表获取白名单或黑名单, 0 白名单, 1 黑名单
        "mac"(必选):
        "urllist"(必选): [ "url1", "url2", .. ] // input url here    just for save
    }

return:
    {
        "code": 0, //返回码
        "msg": "error message" //返回码不为0时会有错误信息返回
    }

```

## 7.3 获取黑名单

功能: 获取黑白名单

方式: 以http post的形式提交查询

接口: /api/sfssystem/urllist\_get

说明: 支持版本 V12

```

body:
    {
        "version"(必选):
        "listtype"(必选): [0/1] //代表获取白名单或黑名单, 0 白名单, 1 黑名单
        "mac"(必选):
    }

return:
    {
        "urllist": ["url1","url2", ...] //web 列表数组
        "code": 0, //返回码
        "msg": "error message" //返回码不为0时会有错误信息返回
    }

```

## 7.4 使能黑白名单

功能: 设置黑白名单

方式: 以http post的形式提交修改

接口: /api/sfssystem/urllist\_enable

说明: 支持版本 V12



```

body: {
    "version"(必选):
    "listfunc"(必选):// 0 no use 1 使用白名单 2 使能黑名单
    "mac"(必选): // 设备mac地址
    "urllist_en": [ "url1", "url2", .. ] // input url here, for enable
}

return:
{
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 8. 内部使用接口

### 8.1 有线设备扫描

功能：当lan端网口发生网线插拔时调用来扫描设备变化

方式：ssst以http post的形式提交操作

接口：/api/sfsystem/device\_list\_backstage

说明：该协议中无须传递协议的版本参数

e.g.:

```

return:
{
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
}

```

### 8.2 ARP事件处理

功能：当lan端arp表发生变化时更新相应设备状态

方式：ssst以http post的形式提交操作

接口：/api/sfsystem/arp\_check\_dev

说明：该协议中无须传递协议的版本参数

e.g.:

```

body:{
    "ip"(必选):, // 设备IP地址
    "mac"(必选): // 设备mac地址
}
return:
{
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
}

```

## 8.3 初始化信息

功能：获取系统相关信息

方式：**ssst**以**http post**的形式提交操作

接口：**/api/sfsystem/init\_info**

说明：该协议中无须传递协议的版本参数

e.g.:

```
return:
{
    "romtype":, //硬件型号
    "name":, //设备名
    "romversion":, //rom版本
    "sn":, //sn号
    "hardware":, //硬件描述
    "mac":, //eth0 mac 地址
    "routerid":, //设备id
    "storage":, //是否支持外界储存设备
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
}
```

## 9. P2P

### 9.1 创建新p2p会话

功能：创建新p2p会话

方式：**app + http post**的形式提交操作

接口：**/api/sfsystem/new\_oray\_params**

说明：支持版本 V12

e.g.:

```
body:{
    version(必选):
}
return:
{
    "url":, //服务器地址
    "session":, //session id
    "code": 0, //返回码
    "msg": "error message" //返回码不为0时会有错误信息返回
}
```

### 9.2 删除p2p会话

功能：删除p2p会话

方式：**app + http post**的形式提交操作

接口：/api/sfsystem/destroy\_oray\_params

说明：支持版本 V12

e.g.:

```
body:{
  "version"(必选):,
  "session"(必选): //session id
}
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}
```

## 10. 其他操作

### 10.1 测试

功能：作为测试命令

方式：APP + 路由器以http post的形式提交操作

接口：/api/sfsystem/welcome

说明：该协议中无须传递协议的版本参数

e.g.:

```
return:
{
  "code": 0, //返回码
  "msg": "error message" //返回码不为0时会有错误信息返回
}
```

## 11. 错误码

| 错误码(code) | 释义(msg)  |
|-----------|--|
| 0         | OK   |
| 1001      | operation not permit                             |
| 1002      | protocol version not found                       |
| 1003      | protocol version not support                     |
| 1004      | command not support                              |
| 1005      | old_password is incorrect while setting password |
| 1006      | signal mode not support                          |
| 1007      | no ssid exist in router                          |

|      |   |
|------|---|
| 1007 | no ssid exist in router                 |
| 1008 | ssid doesn't match                      |
| 1009 | reset is running                        |
| 1010 | can't get lan speed                     |
| 1011 | session is out of date                  |
| 1012 | router has been bind                    |
| 1013 | empty userid                            |
| 1014 | internal socket error                   |
| 1015 | bind router fail                        |
| 1016 | router has not bind yet                 |
| 1017 | unbind router fail                      |
| 1018 | unbind fail case caller is not binder   |
| 1051 | execute check wan type exception        |
| 1052 | prase wan type result fail              |
| 1053 | execute get wan type exception          |
| 1054 | execute set wan unable to get type      |
| 1055 | execute set wan get a unnormal type     |
| 1056 | execute set wan unable to get dns input |
| 1400 | check manager param fail                |
| 1401 | do manager operation fail               |
| 1500 | otaversion information not downloaded   |
| 1501 | checksum not match the ota_checksum     |
| 1502 | localversion the same with otaversion   |
| 1503 | waiting for ota upgrade                 |
| 1504 | upload log file to bmob failed          |
| 1505 | upload log url info failed              |
| 1506 | waiting for another log upload          |