# 后台接口文档

一 长连接部分

数据交换格式全部分为json形式字符串

服务器地址：

tcp长连接 47.106.114.236:9510

Websocket ws://47.106.114.236:9511

两者选一进行连接，对客户端来说是等效的。 (pc和安卓需要选同一种)

长连接命令字：

(一)管理端发送

1. 开机登陆操作

req: {“act”:”ma\_login”,”user\_name”:xxx,”pwd”:”xxxx”,”udid”:xxx}

res: {“cmd”:”ma\_login”,”status”:xxx,”msg”:xxx}

1. 长连接心跳包（10秒一次）

req: {“act”:”ma\_heartbeat”,”user\_id”:xxx}

res: {“cmd”:” ma\_heartbeat”,”status”:xxx,”msg”:xxx}

1. 请求某个设备开启推流

req: {“act”:”ma\_open\_rtsp”,”device\_id”:xxx}

res: {“cmd”:”ma\_open\_rtsp”,”device\_id”:xxx,”status”:xxx,”msg”:xxx}

1. 请求关闭某个设备的推流

req: {“act”:”ma\_stop\_rtsp”,”device\_id”:xxx}

res: {“cmd”:”ma\_stop\_rtsp”,”device\_id”:xxx,”status”:xxx,”msg”:xxx}

5) 请求和某个设备进行视频通话

Req: {“act”:”ma\_open\_video\_call”,”device\_id”:xxx,”room\_id”:xxx}

Res:{“cmd”:”ma\_open\_video\_call”,”status”:xxx,”msg”:”xxxx”}

1. 群视频聊天

Req:{"act":"ma\_open\_group\_video\_call","room\_id":"xxx","device\_ids":["xxx1","xxx2"]}

Res:{"cmd":"ma\_open\_group\_video\_call","device\_ids":["adaa-dadna-daa","xxx2"],"status":true,"msg":"请求发送成功！"}

1. 群音频聊天

Req:{"act":"ma\_open\_group\_audio\_call","room\_id":"xxx","device\_ids":["xxx1","xxx2"]}

Res:{"cmd":"ma\_open\_group\_audio\_call","device\_ids":["adaa-dadna-daa","xxx2"],"status":true,"msg":"请求发送成功！"}

(二)管理端接收

1) 用户登陆或下线状态

Res: {“cmd”:”server\_push\_user\_status”,”data”:{“user\_id”:xxx,”action”:xxxx,”status”:xxx}}

Res: {“cmd”:”server\_push\_sos”,”data”:{“device\_id”:xxx,”xxx”:xxx,...}}

1. 帽子端发送

长连接断开后30秒重连。

1. 帽子登陆

req: {“act”:”ca\_login”,”device\_id”:xxx}

Res: {“cmd”:”ca\_login”,”status”:true,”msg”:xxx,”data”:{}}

1. 帽子上报位置（当作心跳包）

Req: {“act”:”ca\_report\_location”,”user\_id”,”x\_point”:xxx,”y\_point”}

Res: {“cmd”:”ca\_report\_location”,”status”:true,”msg”:xxx}

1. SOS

Req: {“act”:”ca\_sos”,”device\_id”:xxx,”x\_point”:xxx,”y\_point”:xxx}

Res: {“cmd”:”ca\_sos”,”status”:true,”msg”:xxx}

1. 上传wifi列表到服务器

wifi\_enabled 表示当前是否开启wifi

Req: {“act”:”ca\_upload\_wifi\_list”,device\_id:xxx,,”wifi\_enabled":1,wifi\_list:[

{“spot”,”a”, //wifi名

”status”,0, //连接状态 0未连接，1已连接

”pwd”,”xxxx”, //是否需要密码：0否，1要

“intensity”:xxx //信号强度

},

]}

Res: {“cmd”:”ca\_upload\_wifi\_list”,”status”:true,”msg”:xxx}

1. 上报wifi连接结果到服务器

Req: {“act”:”ca\_report\_wifi\_connect\_status”,”device\_id”:xxx,”spot”:xxx,”status”:xxx,”msg”:xxx}

Res:{“cmd”:”ca\_report\_wifi\_connect\_status”,”status”:”xxx”,”msg”:xxx}

1. 帽子端接收
2. 开启推流

Res: {“cmd”:”server\_push\_open\_rtsp”}

1. 关闭推流

Res: {“cmd”:”server\_push\_stop\_rtsp”}

1. 服务器请求上传wifi列表

Res: {“cmd”,”server\_pull\_wifi\_list”}

1. 进行wifi连接

Res:{"cmd":"server\_push\_set\_wifi","data":{"act":"app\_set\_wifi","wifi\_enabled":1,"spot":"wifi1","pwd":"123456"}}

pwd可选，无密码的时候没有该字段

1. 进入视频聊天房间

Res:{“cmd”:”server\_push\_open\_video\_call”,”room\_id”:xxx}

1. 进入音频聊天房间

Res:{"cmd":"server\_push\_open\_audio\_calll","room\_id":"xxx"}

1. 接收创建房间邀请他人帮助

Res:{"cmd":"server\_push\_app\_ask\_for\_help","user\_ids":["1000","1002"]}

1. 用户app端发送
2. 用户登陆

req: {”act“:”app\_login“,”user\_name”:xxx,”pwd”:”xxxx”,”udid”:xxx}

res: {“cmd”:“app\_login”,“status”:xxx,“msg”:xxx}

1. 请求wifi列表

req: {“act”:“app\_get\_wifi\_list”}

res: {”cmd“:“app\_get\_wifi\_list”,“status”:xxx,“msg”:“xxx”}

1. 选择wifi进行连接

req: {“act”:“app\_connect\_wifi”,”spot”:”xxx”,”pwd”:xxx}

pwd如果wifi不需要密码，则去掉该字段

res: {”cmd“:“app\_connect\_wifi”,“status”:xxx,“msg”:“xxx”}

4)长连接心跳包（10秒一次）

req: {“act”:”app\_heartbeat”}

res: {“cmd”:” app\_heartbeat”,”status”:xxx,”msg”:xxx}

1. app对求助进行回应

// type: 0表示拒绝；1表示同意

Req: {"act":"app\_reply\_for\_help","type":0,"user\_id":"1001"}

Res: {"cmd":"app\_reply\_for\_help","status":true,"msg":"发送成功！"}

1. 用户app端接收
2. 接收wifi列表

Res: {“cmd”,”server\_push\_wif\_list”,”wifi\_list”:[{

”spot”:”xxxx”, //信号名

”status”,0, //连接状态 0未连接，1已连接

”pwd”,”xxxx”, //是否需要密码：0否，1要

”intensity”:xxx //信号强度

}]

1. 接收wifi连接状态

Res：{"cmd":"server\_push\_set\_connect\_status","data":

{"spot":"wifi1","status":false,"msg":"密码错误！"}}

1. 收到邀请

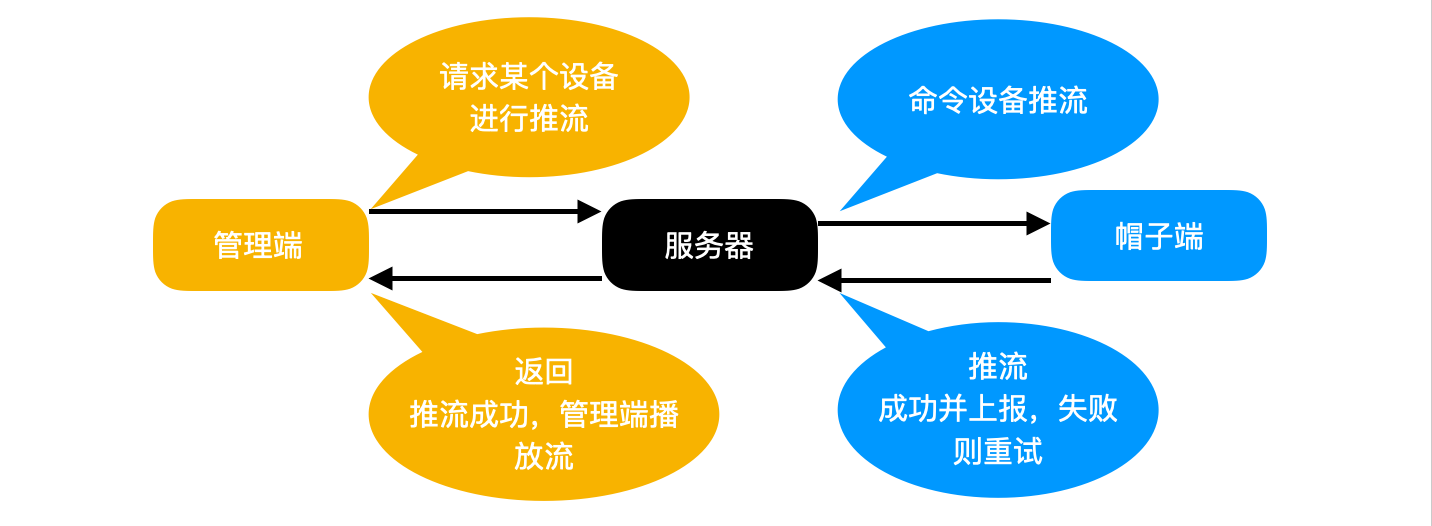
Res:{"cmd":"server\_push\_add\_room\_to\_help","room\_id":"xxx","ca\_info":{"user\_id":"1002","device\_id":"12345-12345-12345","mobile":"134644234567","pwd":"123456","c\_time":"1542424205","app\_last\_login\_time":"0","ca\_last\_login\_time":"1542623786","user\_name":"runde\_01","real\_name":"runde\_01","user\_img":"","department":"group2","role":"施工"}}

4)接收邀请的回复

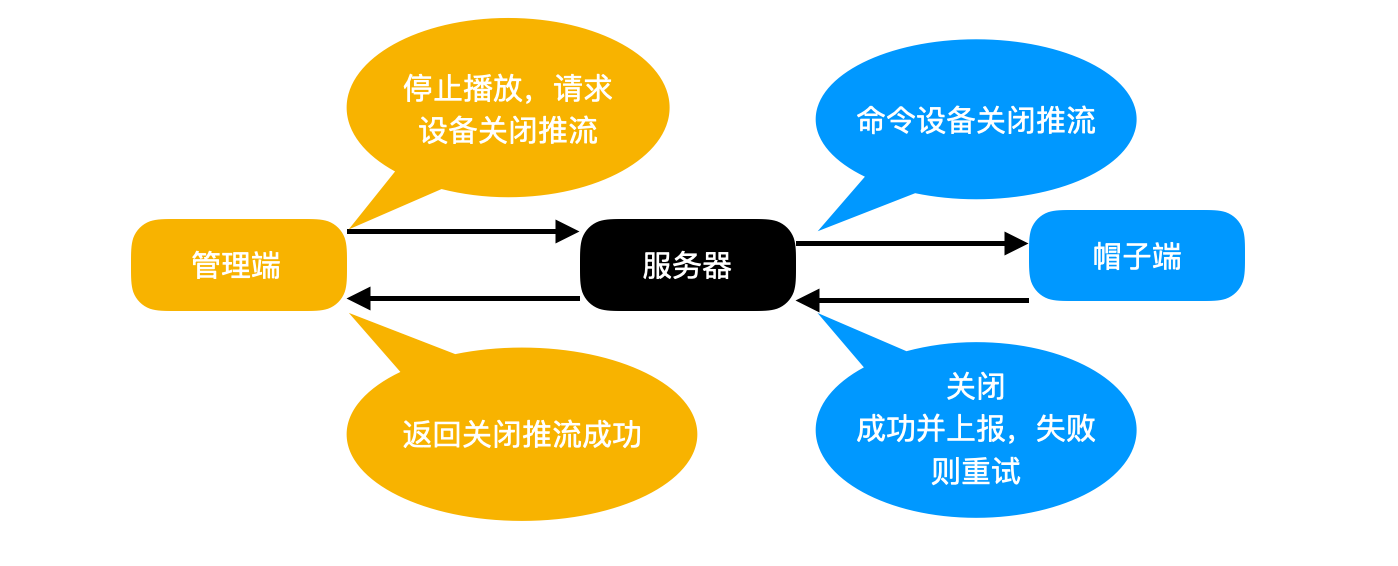
Res:{"cmd":"server\_push\_help\_reply","type":0,"user\_info":{"user\_id":"1000","device\_id":"255533366988887","mobile":"1234567890","pwd":"123456","c\_time":"123456789","app\_last\_login\_time":"123456789","ca\_last\_login\_time":"1542614644","user\_name":"test1","real\_name":"bruce","user\_img":"123.png","department":null,"role":null}}

长连接流程图

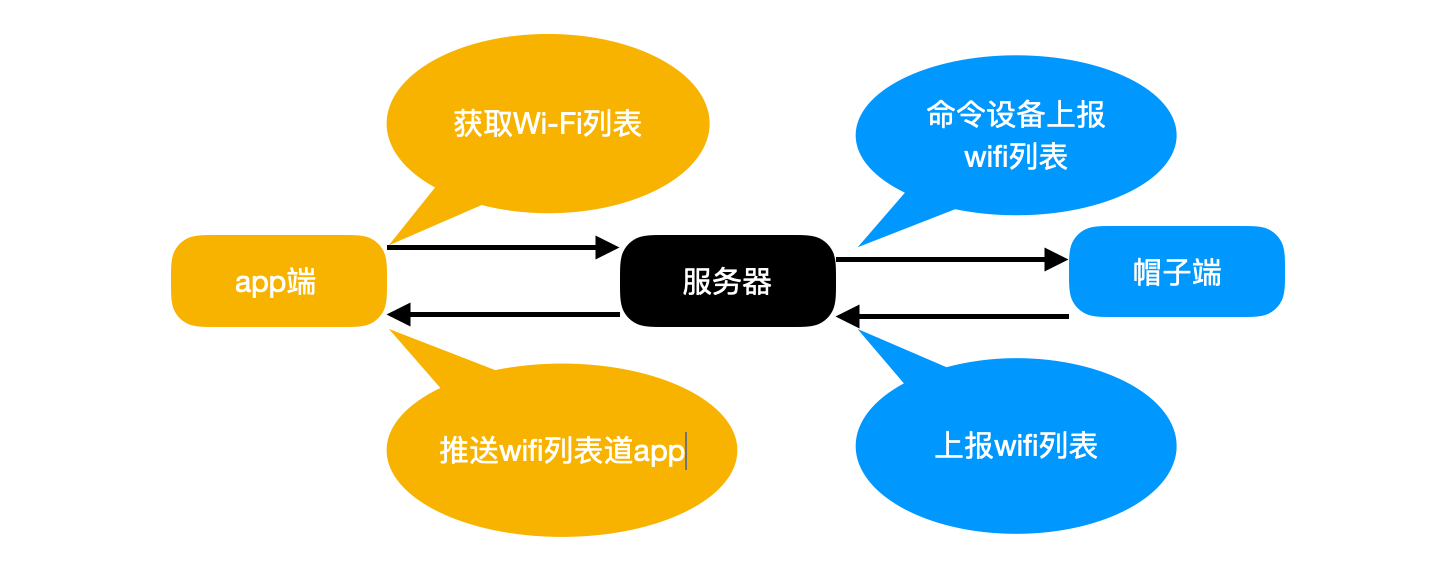
1. 推流流程图



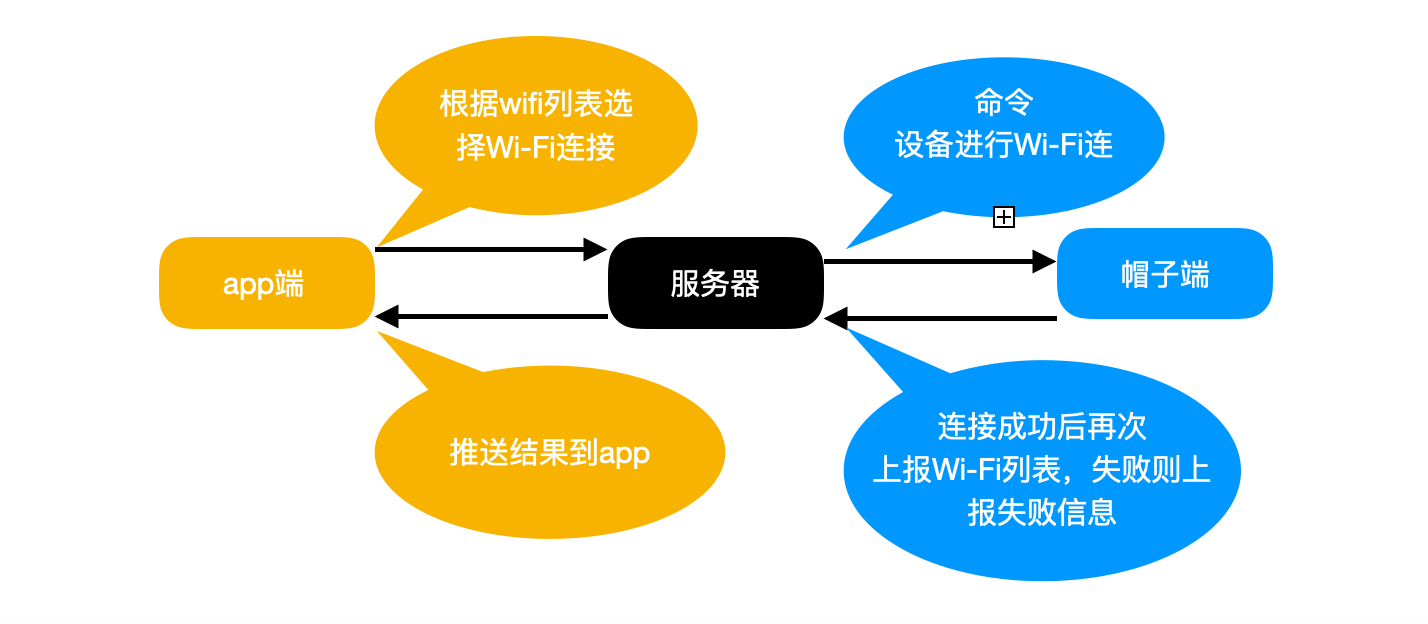
1. 关闭推流流程图



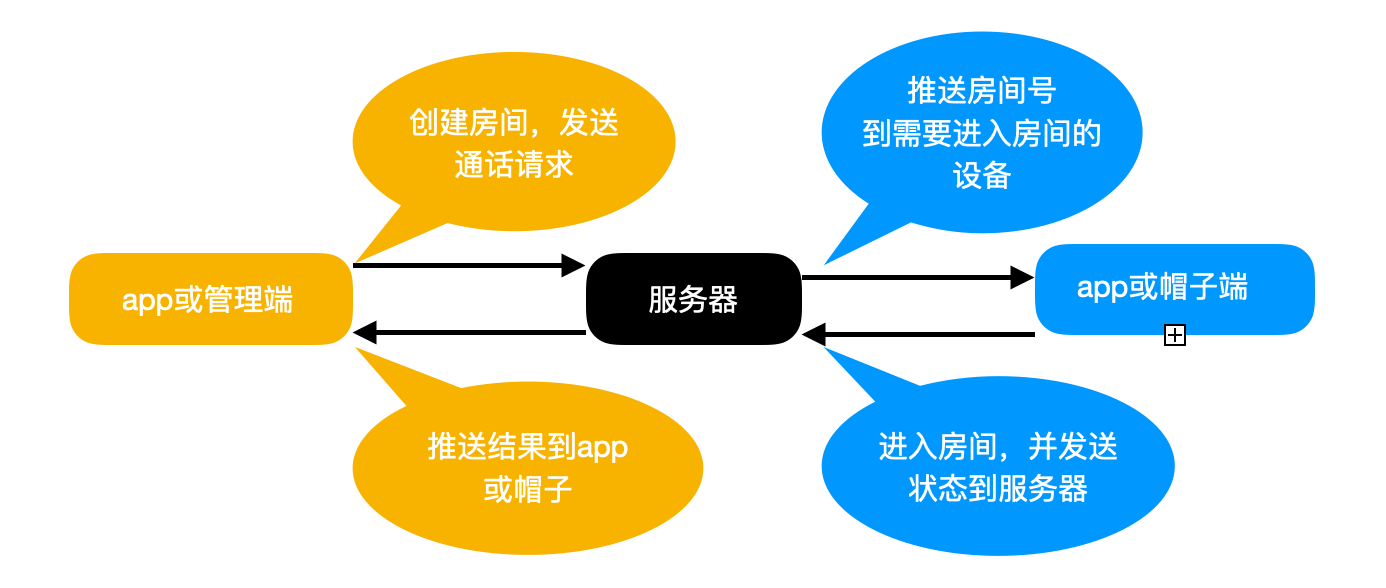
1. app获取帽子wifi列表流程图



1. app操作wifi连接流程图



1. 音视频通话及群聊



6 app控制帽子进行视频通话

二 推拉流部分

采用RTSP协议进行推拉流，只需播放并保存视频，支持本地回放。

详细参考

三 音视频通话部分

详细参照环信。

四 管理端API部分

服务器地址 http://aqm.runde.pro/api/

1. 任务管理
2. 添加任务地址

POST : index.php?ctl=task&act=add\_address

Body : admin\_id:xxx 管理员id

udid=xxx mac地址的md5值，小写

address=xxx 地址

x\_point=xxx

y\_point=xxx

Result : {“status”:true,”msg”:”添加成功”}

1. 获取任务地址列表(不分页)

POST : index.php?ctl=task&act=get\_address\_list

Body : admin\_id:xxx

udid=xxx

Result : {“status”:true,”msg”:”获取成功”,”data”;[{

“i\_id”:xxx

“address”:xxx

}]}

1. 删除任务地址

POST : index.php?ctl=task&act=del\_address

Body : admin\_id:xxx 管理员id

udid=xxx

i\_id:xxx 地址id

Result : {“status”:true,”msg”:”删除成功”}

1. 获取任务类容列表(不分页)

POST : index.php?ctl=task&act=get\_task\_item\_list

Body : admin\_id:xxx

udid=xxx

Result : {“status”:true,”msg”:”获取成功”,”data”;[{

“i\_id”:xxx

“contents”:xxx

}]}

1. 删除任务类容

POST : index.php?ctl=task&act=del\_task\_item

Body : admin\_id:xxx 管理员id

udid=xxx

i\_id:xxx 任务id

Result : {“status”:true,”msg”:”删除成功”}

1. 添加任务类容

POST : index.php?ctl=task&act=add\_task\_item

Body : admin\_id=xxx 管理员id

udid=xxx

task=xxx 任务类容

Result : {“status”:true,”msg”:”添加成功”}

1. 发布任务

POST : index.php?ctl=task&act=publish\_task

Content-Type: application/json(body为json字符串)

Body : admin\_id : xxx 管理员id

udid=xxx

main\_user\_id : xxx 执行人id

e\_time : xxxx 截止时间（unix时间戳）

b\_time : xxx 开始时间（unix时间戳）

location : 任务地址

x\_point :

y\_point :

aux\_user : [aux\_user\_id,] 辅助人员id

task\_detail : [{“i\_id”:”contents”},{“”:””}]当自定义时 i\_id 为A，B，C依次递加

Result : {“status”:true,”msg”:”发布成功”}

1. 删除任务

POST : index.php?ctl=task&act=del\_task

Body : admin\_id : xxx

udid=xxx

task\_id:xxx

Result : {“status”:true,”msg”:”删除成功”}

1. 更新任务

POST : index.php?ctl=task&act=update\_task

Content-Type: application/json

Body : admin\_id : xxx 管理员id

udid=xxx

task\_id : xxx 任务id

main\_user\_id : xxx 执行人id

e\_time : xxxx 截止时间（unix时间戳）

b\_time : xxx 开始时间（unix时间戳）

location : 任务地址

x\_point :

y\_point :

aux\_user : [aux\_user\_id,xxx] 辅助人员id

task\_detail : [{“i\_id”:”contents”},{“”:””}]当自定义时 i\_id 为A，B，C依次递加

Result : {“status”:true,”msg”:”修改成功”}

1. 获取任务列表

POST : index.php?ctl=task&act=get\_task\_list

Body : admin\_id :xxx

udid=xxx

p : xxx 分页（1开始）

status : 状态（可选，不传为全部） 0:未开始，1:进行中，2已完成

time1:xxx 开始时间1（可选）unix时间戳

Time2:xxx 开始时间2 （可选）unix时间戳

Result : {“status”:true,”msg”:”获取成功”,”data”:

[

{

“task\_id”:xxx,

“task\_name” : xxx,

“task\_detail”:[{ “i\_id”:”contents”},{}],

“c\_user\_id”:xxx 创建人id

“c\_user\_name”:xxx 创建人姓名

“c\_time”:xxx 创建时间

“b\_time”:xxx 开始时间

“e\_time”:xxx 截止时间

“location”:xxx 位置

“x\_point”:xxx 坐标

“y\_point”:xxx

“main\_user\_id”:xxx 执行人id

“main\_user\_name”:xxx 执行人姓名

“axu\_user”:[ {“aux\_user\_id”,”aux\_user\_name”},{}] 辅助人员

“status”:xxx 状态 0:未开始，1:进行中，2已完成

}

]}

1. 任务统计（获取最近一周每天任务完成情况）

POST : index.php?ctl=task&act=get\_summary

Body : admin\_id=xxx

Result:{“status”:true,”msg”:”获取成功”,”data”:[

{

“date”:xxx

“task\_num”:xxx

“task\_done\_num”:xxx

},

{}...

]}

1. sos统计（获取当月sos报警次数）

POST : index.php?ctl=task&act=get\_sos\_summary

Body:admin\_id=xxx

Result:{“status”:true,

“msg”:”获取成功”，

“total\_sos\_num”:xxx

“data”:[

{

“date”:2018-10-11 日期格式，

“sos\_num”:xxx 报警次数

},

{}

]

}

1. 获取sos列表

POST : index.php?ctl=task&act=get\_sos\_list

Body:admin\_id=xxx

p=xxx

Result:{“status”:true,”msg”:”获取成功”,”data”:[

{

“s\_id”:xxx,

“user\_id”:xxx,

“user\_name”:xxx

“device\_id”:xxx,

“location”:xxx,

“x\_point”:xxx,

“y\_point”:xxx,

“c\_time”:xxx,

“status”:xxx

},

{}...

]}

1. 分组管理

分组只支持一级部分（及某个用户只属于一个部门，该部门下面没有子部分）

1. 创建部门：

POST : index.php?ctl=group&act=create\_group

Body : admin\_id:xxx

udid=xxx

g\_name : xxx 部门名字

Result : {“status”:true,”msg”:”创建成功”}

1. 删除部门：

POST : index.php?ctl=group&act=del\_group

Body : admin\_id:xxx

g\_id : xxx 部门id

Result : {“status”:true,”msg”:”删除成功”}

1. 更新部门信息：

POST : index.php?ctl=group&act=update\_group

Body : admin\_id:xxx

g\_id : xxx 部门id

g\_name : xxx 部门名

Result : {“status”:true,”msg”:”更新成功”}

1. 获取部门列表：

POST:index.php?ctl=group&act=get\_group\_list

Body : admin\_id : xxx

udid=xxx

Result:{

“status”:”true”,

“msg”:”获取成功”

“data”:[

{

g\_id:xxxx 部门id

g\_name:xxx 部门名字

c\_user\_id:xxx 创建人id

c\_user\_name:xxx 创建人姓名

c\_time:xxx 创建时间

}

]

}

1. 添加成员：

POST : index.php?ctl=group&act=add\_user

Body : admin\_id:xxx

udid=xxx

g\_id:xxx

device\_id=xxxx,

mobile=xxx,

pwd=xxx,

user\_name=xxx,

real\_name=xxx,

user\_img=xxx

role=xxxx 职位

Result : {“status”:true,”msg”:”添加成功”}

1. 删除成员：

POST : index.php?ctl=group&act=del\_user

Body : admin\_id:xxx

udid=xxx

user\_id:xxx

Result : {“status”:true,”msg”:”删除成功”}

1. 获取部门成员列表：

POST:index.php?ctl=group&act=get\_group\_member\_list

Body : admin\_id : xxx

udid=xxx

g\_id：xxx

Result:{

“status”:”true”,

“msg”:”获取成功”

“data”:{

[{

user\_id:xxxx 用户id

user\_name:xxx 用户姓名

mobile:xxx 手机号

nick:xxx 昵称

status：xxx(0,1) 在线或离线

},{}]

}

}

1. 修改用户信息

POST : index.php?ctl=group&act=update\_user

Body : admin\_id:xxx

udid=xxx

User\_id=xxx

g\_id:xxx

device\_id=xxxx,

mobile=xxx,

pwd=xxx,

user\_name=xxx,

real\_name=xxx,

user\_img=xxx

role=xxxx 职位

Result : {“status”:true,”msg”:”修改成功”}

1. 轨迹

1.获取用户轨迹：

POST : index.php?ctl=location&act=get\_user\_path

Body: admin\_id:xxx

user\_id:xxx

result : {“status”:true,”msg”:xxx,”data”:

[l\_id:xxx,

“user\_id”:xxx,

“time”:xxx,

“x\_point”:xxx,

“y\_point”:xxx]}

1. 权限管理
2. 添加管理员

POST:index.php?ctl=admin&act=add\_admin

Body : admin\_id:xxx 当前管理员id

udid=xxx

user\_name:xxx 用户名

mobile:xxx 手机号

department:xxx 部门

pwd:xxx 密码

role:xxx 职位

permission\_list 权限

Result :{“status”:true,”msg”:”添加成功”}

1. 删除管理员

POST : index.php?ctl=admin&act=del\_admin

Body : admin\_id:xxx

udid=xxx

u\_admin\_id:xxx

Result :{“status”:true,”msg”:”删除成功”}

1. 编辑管理员信息（）

POST : index.php?ctl=admin&act=update\_admin

Body : admin\_id:xxx 当前管理员id

udid=xxx

u\_admin\_id :xxx 被修改的管理员账号id

user\_name:xxx 用户名

mobile:xxx 手机号

department:xxx 部门

pwd:xxx 密码

role:xxx 职位

permission\_list 权限列表

Result :{“status”:true,”msg”:”更新成功”}

1. 获取权限列表（）

POST : index.php?ctl=admin&act=get\_permision\_list

Body : admin\_id:xxxx

udid=xxx

Result : {“status”:true,”msg”:”获取成功”,”permission\_list”:[1.2,1.3,2.1,xx]}

1. 用户端API
2. 用户登陆

POST : index.php?Ctl=user&act=do\_login

Body : user\_name=xxx

pwd=xxx

Result :{“status”:true,”msg”:”登陆成功”}

2.获取自己的任务列表

POST : index.php?ctl=user&act=get\_task\_list

Body : user\_id=xxx

Result :{“status”:true,”msg”:”获取成功”,”data”:[

{

"group\_detail": {

"g\_id": "2",

"g\_name": "group22",

"c\_user\_id": "1",

"c\_time": "1540546149"

},

"members\_list": [{

"nick": "test12",

"user\_id": "4",

"device\_id": "1234-dahda-asgah",

"mobile": "123987654",

"pwd": "12345",

"c\_time": "1540548109",

"app\_last\_login\_time": "0",

"ca\_last\_login\_time": "0",

"user\_name": "test12",

"real\_name": "runde",

"user\_img": "",

"department": "group22",

"role": "???",

"g\_name": "group22"

}, {

"nick": "test12",

"user\_id": "5",

"device\_id": "1234-dahda-asgah",

"mobile": "123987654",

"pwd": "12345",

"c\_time": "1540548156",

"app\_last\_login\_time": "0",

"ca\_last\_login\_time": "0",

"user\_name": "test12",

"real\_name": "runde",

"user\_img": "",

"department": "group22",

"role": "技术员",

"g\_name": "group22"

}]

},

},

]}

3.获取群组列表及成员

POST : index.php?ctl=user&act=get\_group\_member\_list

Body : user\_id=xxx

4.更新任务信息

POST : index.php?ctl=user&act=report\_task

Body : user\_id=xxx

task\_id : xxx 任务id

task\_detail : [{“i\_id”:”contents”},{“”:””}]当自定义时 i\_id 为A，B，C依次递加

extra : xxx备注

Result :{“status”:true,”msg”:”更新成功”}