



API 接口协议调用说明

最新版本 4.1



目录

1. 协议调用说明	6
1.1 调用验证	6
2. 协议调用详细说明	7
2.1 组建/编辑用户	7
2.2 调用用户列表	7
2.3 用户登录	7
2.4 删除用户	8
2.5 调用角色	8
2.6 调用版本信息	8
2.7 告知服务器更新终端信息	9
2.8 调用终端信息	9
2.9 过滤终端信息	11
2.10 终端控制	11
2.11 添加终端	11
2.12 设置终端文本扩展信息	12
2.13 上传终端图片扩展信息	12
2.14 编辑终端	13
2.15 删除终端	13
2.16 新增任务计划	13
2.17 修改任务计划	15
2.18 删除任务计划	15
2.19 手动开始任务	16
2.20 手动停止任务	16
2.21 添加分区	16
2.22 调用分区信息	16
2.23 编辑分区	17
2.24 删除分区	17
2.25 查询录音日志	17
2.26 调用终端扩展信息	18
2.27 快速执行任务	18
2.28 控制终端短路输出	20
2.29 查询系统日志	21
2.30 查询计划任务日志	22
2.31 调用系统配置	22
2.32 保存系统参数	23
2.33 查询定时任务执行结果	23
2.34 修改服务器时间	24
2.35 终端广播控制	24
2.36 查询 HTTP SDK 接口列表	24
2.37 添加/修改/删除 HTTP SDK 接口	24
2.38 查询设备 IO 报警状态	25
2.39 启用/禁用计划任务	25



2.40	通话控制	25
2.41	查询实时状态	26
2.42	处理实时状态	26
2.43	截取录音录像文件	27
2.44	重启系统	27
2.45	振铃组配置	27
2.46	发送手机短信	28
2.47	发送语音通知	28
2.48	添加离线任务	28
2.49	删除离线任务	29
2.50	查询终端离线任务(发送查询命令)	29
2.51	查询终端离线任务(查询分页结果)	29
2.52	调用自检任务列表	29
2.53	新增自检任务	30
2.54	修改自检任务	30
2.55	删除自检任务	30
2.56	查询终端自检记录	31
2.57	手动运行终端自检	31
2.58	查询终端自检记录排名	31
2.59	调用通讯录	31
2.60	模板保存	32
2.61	调用无线语音网关模板	32
2.62	调用无线语音网关模板详情	32
2.63	调用无线语音网关语音通知	33
2.64	调用无线语音网关语音系统设置	33
2.65	调用无线语音网关模板删除命令	33
2.66	调用无线语音网关重新呼叫命令	33
2.67	调用无线语音网关自动录音开关命令	34
2.68	调用无线语音网关设置录音开关	34
2.69	终端升级	34
2.70	调用报警数据	34
3.	上传消息	35



修改日志

序号	版本	修改内容	修改日期	修改人
1	V1.0	第一次发行	2018-1-16	平台部
2	V1.1	补充终端管理和任务管理接口	2018-1-19	平台部
3	V1.2	增加录音查询接口	2018-1-26	平台部
4	V1.3	增加扩展信息接口 补充推送信息 增加快速执行任务接口	2018-1-30	平台部
5	V1.4	修改推送任务的扩展信息读取执行任务上传的扩展信息	2018-2-1	平台部
6	V1.5	并增加 postman 截图参考； 扩展信息增加 Develop 数据判断；	2018-2-23	平台部
7	V1.6	更新部分细节和错误，增加推送消息接收判断	2018-3-12	平台部
8	V1.7	增加音量控制和短路输出	2018-6-1	平台部
9	V1.8	增加用户和日志等相关接口	2018-6-10	平台部
10	V1.9	录音增加按终端 ID 进行筛选 补充任务方案相关接口	2018-6-23	平台部
11	V2.0	推送数据增加任务编号 增加登录接口	2018-7-18	平台部
12	V2.1	补充任务相关接口 增加系统配置接口	2018-8-22	平台部
13	V2.2	增加设置时间接口 增加批量删除接口	2018-9-13	平台部
14	V2.3	报警类型增加电源报警	2018-9-26	平台部
15	V2.4	增补部分接口	2018-10-5	平台部
16	V2.5	增加终端控制的接听功能 推送消息增加是否为发起方	2018-10-10	平台部
17	V2.6	增加电话接口 补全部分接口	2019-1-6	平台部
18	V2.7	补充部分接口	2019-2-26	平台部
19	V2.8	增加通话控制和单位的相关接口	2019-3-27	平台部
20	V2.9	增加实时状态相关接口	2019-4-5	平台部
21	V3.0	增加录音录像截取的接口	2019-6-23	平台部
22	V3.1	增加 HTTP 安全验证	2019-8-25	平台部
23	V3.2	补全部分接口 增加振铃组相关接口	2019-8-29	平台部
24	V3.3	增加快速执行任务，文件播放控制的接口	2019-10-29	平台部
25	V3.4	设计短信及语音通知相关接口 快速执行任务增加查询实时状态的参数	2020-6-7	平台部
26	V3.5	完善相关接口的参数说明	2020-7-3	平台部
27	V3.6	增加离线任务的接口	2020-9-9	平台部



28	V3.7	修改部分接口说明	2020-10-23	平台部
29	V3.8	增加自检任务的接口 增加会见接口	2020-11-21	平台部
30	V3.9	增加关键字报警的端口号	2021-5-23	平台部
31	V4.0	增加终端升级接口	2021-8-28	平台部
32	V4.1	修正推送消息接口文档	2022-1-3	平台部



1. 协议调用说明

a. 接口均使用 HTTP 协议，接口调用采用基本的 POST 方法，Content-Type 为“application/x-www-form-urlencoded”格式，传递的参数均为 Developdata[xxx]格式，返回的数据结果均为 DEVELOP 格式，如下图所示（建议使用 PostMan 工具来进行调试）：

b. 生成的代码格式如下所示：

```
POST /php/getsetzonedata.php HTTP/1.1
Host: 192.168.0.104
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
```

```
Developdata%5BpageIndex%5D=0&Developdata%5BpageCount%5D=30
```

c. HTML/js/Ajax 调用方式如下所示：

```
function loaderZoneData(pIndex, pCount) {
    var DevelopData = {
        "pageIndex" : pIndex,
        "pageCount" : pCount
    };
    $.ajax({
        type: "POST",
        url: "192.168.1.13/php/getsetzonedata.php",
        dataType: "Develop",
        data: { "Developdata" : DevelopData },
        success: function(resData) {
            // 处理数据
        },
        error: function() {
        }
    });
}
```

1.1 调用验证

为保证接口调用的安全性，系统支持调用验证，可在综合设置配置开启：

当开启 HTTP 安全验证后，调用步骤如下：

- 调用登录接口，获取 token 字段，请参考 [2.48 用户登录](#)；
- 其他全部接口，调用时需增加 token 参数： Developdata[token]：登录返回密钥
- 若 token 未发送或者 token 错误/过期，接口调用完成后，服务器会返回如下信息：

字段名	说明
res	是否调用接口成功(1:成功 0:失败)
errcode	-1: token为空 -2: token不存在 -3: token过期



2. 协议调用详细说明

2.1 组建/编辑用户

路径: /php/saveuserinfo.php

调用参数: Developdata[user]: 用户名

Developdata[pass]: 密码

Developdata[display]: 显示名称

Developdata[role]: 角色 (参考 2.30 返回 role 值)

Developdata[rights]: 终端权限 (以, 分割), *表示所有终端

Developdata[servers]: 服务器权限 (以, 分割), *表示所有服务器

Developdata[bindterminals]: 绑定终端 ID, 用于视频联动弹窗

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

2.2 调用用户列表

路径: /php/getuserdata.php

调用参数: Developdata[pageIndex]: 页码序号 (从 0 开始)

Developdata[pageCount]: 每页数量 (如: 请求从 10-20 的数据, 则 pageIndex 为 1, pageCount 为 10)

返回数据:

```
{ "res": "1", "rows": [{"res": "1", "username": "123", "password": "123", "display": "123", "role": "操作员", "rights": "2,3", "bindterminals": ""}], "total": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)
username	用户名
password	密码
display	显示名称
role	角色名
rights	终端权限
bindterminals	绑定话筒终端 ID
total	用户总数

2.3 用户登录

路径: /php/login.php

调用参数: Developdata[username]: 用户名

Developdata[password]: 密码

返回数据:

```
{"res": "1", "username": "admin", "password": "admin", "display": "admin", "role": "超级管理员", "rights": "*", "isadmin": "1", "serverrights": "*", "bindterminals": "1" }
```



字段名	说明
res	1: 登录成功 0: 登陆失败
username	用户名
password	密码
display	显示名称
role	角色
rights	终端权限
isadmin	是否为管理员
serverrights	服务器权限
bindterminals	绑定终端
token	其他接口安全认证所需密钥

2.4 删除用户

路径: /php/deleteuserinfo.php

调用参数: Developdata[user]: 用户名

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

2.5 调用角色

路径: /php/getroledata.php

调用参数: Developdata[pageIndex]: 页码序号 (从 0 开始)

Developdata[pageCount]: 每页数量 (如: 请求从 10-20 的数据, 则 pageIndex 为 1, pageCount 为 10)

返回数据: rolelist:

```
[{role: "广播站"}, {role: "操作员"}, {role: "用户"}, {role: "管理员"}, {role: "超级管理员"}]
```

2.6 调用版本信息

路径: /php/getversiondata.php

调用参数: 无

返回数据:

```
{"sipver": "V3.0.0(20200529)", "sdkver": "v2.0(20180408)_WS", "eslver": "v2.0(20180408)_WS", "messagetarget": "9990001-9990999", "sysver": "V3.0.1_20200716_RELEASE"}
```

字段名	说明
sysver	系统版本
sipver	核心服务器版本
sdkver	应用 SDK 版本
eslver	核心 SDK 版本
messagetarget	内部系统编号



2.7 告知服务器更新终端信息

路径: /php/reloadxml.php

调用参数: Developdata[id]: 终端 ID (为空表示通知服务器更新所有终端)

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

2.8 调用终端信息

路径: /php/getzoneterminaldata.php

调用参数: Developdata[pageIndex]: 页码序号 (从 0 开始)

Developdata[pageCount]: 每页数量 (如: 请求从 10-20 的数据, 则 pageIndex 为 1, pageCount 为 10)

Developdata[groupName]: 分区名称, (*表示所有分区)

Developdata[showType]: 显示类型 (0: 全部; 1: 只显示在线; 2: 只显示离线)

Developdata[user]: 当前登陆用户账号

Developdata[simple]: 显示内容 (1: 简要信息 2: 详细信息)

Developdata[ishowmutiserver]: 是否显示跨服务器信息

返回数据:

```
{ "res": "1", "rows": [{"name": "Terminal1", "id": "1", "ip": "0.0.0.0", "state": "-1", "bcoutv": "7", "bcinv": "9", "talkoutv": "9", "ilenam": "9", "volume": "9,9,9,7", "localip": "0.0.0.0", "task": "0"}], "total": "3" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)
id	终端编号
name	终端名称
ip	终端 IP 地址 (登录地址)
bcoutv	广播输出音量 (0-15)
bcinv	广播输入音量 (0-15)
talkoutv	对讲输出音量 (0-15)
talkinv	对讲输入音量 (0-15)
volume	所有音量 (以逗号分隔)
bcttype	广播类型 (单播 unicast, 组播 multicast)
terminall evel	终端优先级
calltimeo ut	呼叫超时时长
calltarge t1	呼叫终端 (用,分割多个目标 id)



calltarget2	呼叫终端（用,分割多个目标 id）
localip	终端本机 IP 地址
recvbtlevel	广播优先级
recvtalklevel	对讲优先级
recvspylevel	监视优先级
calleelist	对讲成员列表
monitorlist	监事成员列表
bctlist	广播成员列表
rtspurl	视频地址
bctrecvtype	接受广播类型
recvmettinglevel	会议优先级
mettingeelist	参会成员列表
members	终端密码
multiparty	是否允许多方通话 0:1
enablevideo-record	是否允许录音录像 0:1
deviceinfo	终端扩展信息，如型号，品牌等（部分设备支持）
deposit	
state	终端类型（-1:脱机； 0:XC 终端； 1:NAS 终端； 2:SIP 终端； 3:Android 安卓 APP； 4:IOS 苹果 APP； 5: Windows APP； 6:TS 拾音器终端）
task	终端状态（包含两种类型的值） 0: 空闲； X1 X2 X3 X4 X5 X1: (0: 对讲 1: 监听； 2: 广播 3: 会议) X2: (任务目标对应的终端 ID) X3: (0: 接收方 1: 发起方) X4: (对应的任务名称) X5: (对应的播放列表名称，如 test.list) 【若未指定播放列表，则为当前任务播放的文件名称】
panels	终端面板状态
deviceinfo	终端扩展信息，如型号，品牌等（部分设备支持）



o	
nvrurl	Nvr 地址
multitask	是否多任务 0:1
total	终端总数

2.9 过滤终端信息

路径: /php/ getsingleterminaldata. php

调用参数: Developdata[pageIndex]: 页码序号 (从 0 开始)

Developdata[pageCount]: 每页数量 (如: 请求从 10-20 的数据, 则 pageIndex 为 1, pageCount 为 10)

Developdata[showType]: 显示类型 (0: 全部; 1: 只显示在线; 2: 只显示离线)

Developdata[user]: 当前登陆用户账号

Developdata[searchTxt]: 筛选内容 (系统会模糊匹配 ID 和名称符合的终端)

Developdata[onlyId]: ~~是否只筛选 ID~~

Developdata[groupName]: 分区名称, (*表示所有分区)

(若 searchTxt 中包含”, ”号, 则筛选多个终端信息; 如 1,3,8 表示筛选终端 ID 为 1, 3, 8 的终端信息, 当为筛选终端 ID 模式, showType 不再生效)

返回数据: 同 2.3 获取终端信息

2.10 终端控制

路径: /php/exesdkcommand. php

调用参数: Developdata[source]: 源终端

Developdata[target]: 目标终端 (如有多个目标终端, 用” - “分隔, 如 2-3)

Developdata[commandtype]: 命令类型 (1: 对讲 2: 监听 3: 广播 5: 会议)

Developdata[isstop]: 是否停止 (0: 开始 1: 停止 2: 接听【当命令类型为“对讲”时有效】)

Developdata[user]: 当前登陆用户账号

Developdata[extdata]: 第三方传入数据, 可在推送接口中返回

Developdata[hasvideo]: 1 表示视频会议; 0 表示音频会议 (commandtype=5 为会议时有效)

Developdata[isarea]: 1 分区广播; 0 终端广播 (commandtype=3 为广播时有效)

Developdata[targetpanel]: 目标终端面板号 (commandtype=1 或 2 为对讲或监听时有效)

Developdata[phonenummer]: 呼叫电话号码 (commandtype=1 时有效)

返回数据:

{ “res” :” 1”, “extdata” :” ” }

字段名	说明
res	是否调用接口成功(1:成功 0:失败)
extdata	传入的数据原样返回

2.11 添加终端

路径: /php/newterminal. php

调用参数: Developdata[tid]: 设备 id (非空数字)

Developdata[tname]: 终端名称 (非空)



Developdata[tbcoutv]: 终端广播输出音量 (0-15), 默认 9
 Developdata[tbcinv]: 终端广播输入音量 (0-15), 默认 9
 Developdata[ttalkoutv]: 终端对讲输出音量 (0-15), 默认 9
 Developdata[ttalkinv]: 终端对讲输出音量 (0-15), 默认 9
 Developdata[tbcttype]: 广播发送传输方式 (multicast: 组播; unicast: 单播)
 Developdata[bctrecvtype]: 广播接收传输方式 (multicast: 组播; unicast: 单播)
 Developdata[tterminallevel]: 终端级别 (0-15), 默认为 0
 Developdata[tcalltimeout]: 呼叫转移时间 (默认为 30)
 Developdata[tcalltarget1]: 呼叫目标 (已逗号隔开, 默认为,)
 Developdata[tcalltarget2]: 扩展呼叫目标 (已逗号隔开, 默认为,)
 Developdata[tbctlevel]: 终端接收广播优先级 (默认为 4)
 Developdata[ttalklevel]: 终端接收对讲优先级 (默认为 3)
 Developdata[tmonlevel]: 终端接收监听优先级 (默认为 2)
 Developdata[tmeetinglevel]: 终端接收会议优先级 (默认为 1)
 Developdata[tbctlist]: 有权限发送广播终端列表 (*表示全部)
 Developdata[ttalklist]: 有权限发送对讲终端列表 (*表示全部)
 Developdata[tmonlist]: 有权限发送监听终端列表 (*表示全部)
 Developdata[tmeetinglist]: 有权限发送会议终端列表 (*表示全部)
 Developdata[trtspurl]: 绑定外部视频路径 (默认为空)
 Developdata[tpass]: 终端密码 (默认为 1234)
 Developdata[multiparty]: 是否允许多方通话 (0: 禁用 / 1: 启用)
 Developdata[enablevideorecord]: 是否允许对讲录像 (0: 禁用 / 1: 启用, 需 2.0 版本服务器支持)

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否添加终端成功(1:成功 0:失败)

2.12 设置终端文本扩展信息

路径: /php/setextdata.php

调用参数: Developdata[id]: 终端 ID

Developdata[txt]: 扩展文本信息

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

2.13 上传终端图片扩展信息

路径: /php/addextdata.php

(注: 具体格式与 2.20 上传媒体文件方法类似)

JS 调用代码

```

var fileObj = document.getElementById("res_file").files[0]; // js 获取文件对象
if (fileObj)
{

```



```

var url = "../php/addextdata.php";

var form = new FormData();
form.append("upload", fileObj); // 文件对象
form.append("id", curSelTerid); // 终端编号
form.append("index", curExtImgIndex); // 图片序号(1-4)

var xhr = new XMLHttpRequest();
xhr.open("post", url, true);
xhr.onload = uploadComplete;
xhr.onerror = uploadFailed;
xhr.send(form);
}

```

2.14 编辑终端

路径: /php/setterminal.php

调用参数: 同 2.6 添加终端相同

返回数据:

```
{ "res" : "1" }
```

字段名	说明
res	是否修改终端成功(1:成功 0:失败)

2.15 删除终端

路径: /php/delterminal.php

调用参数:

Developdata[tid]: 设备 id (*表示删除所有终端,慎用!)

返回数据:

```
{ "res" : "1" }
```

字段名	说明
res	是否删除终端成功(1:成功 0:失败)

2.16 新增任务计划

路径: /php/addtaskinfo.php

调用参数: Developdata[taskname]: 任务名称

Developdata[isdisable]: 任务是否启用 (0:启用, 1:禁用)

Developdata[level]: 任务优先级 (0-15)

Developdata[creator]: 创建任务用户

Developdata[triggers]: 触发器列表 (参照下方详解)

Developdata[commands]: 操作列表 (参照下方详解)

Developdata[dirname]: 任务所属目录 (为空表示属于根目录)

返回数据:

```
{ "res" : "1" }
```

字段名	说明
-----	----



res	是否添加计划任务成功(1:成功 0: 失败)
taskId	返回新的任务编号
error	res=0 时的相关错误信息

triggers: 触发器列表 (多个触发器之间用“|”分隔, 单个触发器的字段用“*”分隔)

如: |0*2017-10-16*2027-10-16*16-35-30*0*|1*0*1<2*<1|

第一个字段为触发器类型 (0: 定时任务; 1: 事件任务) 【详细字段描述请参照下面表格】

a) 定时任务格式: 0*开始日期*结束日期*开始执行时间*任务周期类型*星期参数

(如: 0*2017-10-16*2027-10-16*16-35-30*0*表示 2017-10-16 至 2027-10-16, 每天 16: 35: 30 执行的定时任务)

(注意: 开始执行时间的传入格式为 16-35-30, 表示 16: 35: 30)

字段	值
触发器类型	0: 定时任务; 1: 事件任务
开始日期	YYYY-MM-DD (如 2017-10-16)
结束日期	YYYY-MM-DD (如 2017-10-16)
开始执行时间	HH-mm-SS (如 16-34-33)
任务周期类型	0: 每天; 1: 每周; 2: 单次
星期参数	以<分隔, (如: 1<2 表示每周只有星期一和星期二执行)

b) 事件任务格式: 1*事件类型*触发终端列表*触发端口

(如: 1*0*1<2*<1 表示终端 1, 2 的端口 1 触发的报警事件任务)

字段	值
触发器类型	0: 定时任务; 1: 事件任务
事件类型	0: 报警; 1: 按键; 2: 刷卡; 3: 登录; 4: 对讲; 5: 监听; 6: 广播
触发终端列表	1<2 表示终端 1, 2
触发端口	范围: 1-500

commands: 操作任务 (多个触发器之间用“|”分隔, 单个触发器的字段用“*”分隔)

如: |3*0*0*0*test1.wav<test2.wav*1<2*0*1|0*2|

第一个字段为操作类型 (0: 延时操作; 1: 对讲操作; 2: 监听操作; 3: 广播操作; 4: 指令操作) 【详细字段描述请参照 2.9B 查询任务信息】

a) 延时操作: 0*持续时间 (单位: 秒)

b) 对讲操作: 1*持续时间*发起终端 ID*接收终端 ID*发起终端面板号*接收终端面板号

c) 监听操作: 2*持续时间*发起终端 ID*接收终端 ID*发起终端面板号*接收终端面板号

d) 广播操作: 3*持续时间*发起终端 ID*数据流类型*文件列表*接收广播终端列表*音量微调*同步强切*音源类型*语音合成文本*是否随机播放

(如 3*0*0*0*test1.wav<test2.wav*1<2*0*0 表示向终端 1 和终端 2 发送 test1.wav 和 test2.wav 文件广播)

(注意: 发起终端 ID 为 0 时, 表示为文件广播; 持续时间为 0 时, 表示持续时间为播放文件总时间: test1.wav 和 test2.wav 时间之和, 持续时间大于 0 时, 不管文件是否播放完成, 都会根据持续时间强行停止; 请在文件名前补齐“//”符号; +文件列表可为播放列表 ID, 如 test.list

音源类型: (0: 文件; 1: 终端; 2: 声卡; 3: 文本)

是否随机播放 (0: 顺序播放 1: 随机播放)

字段	值
----	---



操作类型	0: 延时操作; 1: 对讲操作; 2: 监听操作; 3: 广播操作; 4: 指令操作
持续时间	单位: 秒 持续时间为 0 时, 表示持续时间为播放文件总时间; 持续时间大于 0 时, 不管文件是否播放完成, 都会根据持续时间强行停止
发起终端 ID	3 表示终端 3 (音频类型为 1 时有效)
数据流类型	0: 推流 1: 拉流
文件列表	如 test1.wav<test2.wav
接收广播终端列表	如 1<2 表示终端 1, 2
音量微调	范围: -5 到 5 (默认 0)
同步强切	1: 执行强切 0: 不执行强切
音源类型	0: 文件; 1: 终端; 2: 声卡; 3: 文本
语音合成文本	文本内容 (音频类型为 3 时有效)
是否随机播放	1: 随机播放 0: 顺序播放

e) 指令操作: 4*持续时间*开始指令*结束指令*接收指令终端列表, 如下所示:

4*0*[COMMAND];cmd_type=alarm_out;operation=control;alarm_no=501;alarm_status=1;delay_seconds=0;*2<3 闭合 2, 3 终端输出端口 1 (501 表示端口 1, 502 表示端口 2, 依此类推。。)

4*0*[COMMAND];cmd_type=alarm_out;operation=control;alarm_no=502;alarm_status=0;delay_seconds=0;*1<3 断开 1, 3 终端输出端口 2 (501 表示端口 1, 502 表示端口 2, 依此类推。。)

注: 以上两个列子结束命令皆为空;

2.17 修改任务计划

路径: /php/modifytaskinfo.php

调用参数: Developdata[taskname]: 任务名称

Developdata[isdisable]: 任务是否启用 (0: 启用, 1: 禁用)

Developdata[level]: 任务优先级 (0-15)

Developdata[creator]: 创建任务用户

Developdata[triggers]: 同 2.10 相同字段

Developdata[commands]: 同 2.10 相同字段

Developdata[taskId]: 任务 ID

返回数据:

{ "res": "1" }

字段名	说明
res	是否修改计划任务成功 (1: 成功 0: 失败)

2.18 删除任务计划

路径: /php/exetaskcmd.php

调用参数: Developdata[taskCommand]: 'deltaskinfo'

Developdata[taskId]: 任务 id (*表示删除所有任务)

返回数据:



{ "res": "1" }

字段名	说明
res	是否删除计划任务成功(1:成功 0:失败)

2.19 手动开始任务

路径: /php/exetaskcmd.php

调用参数: Developdata[taskCommand]: 'runtaskinfo'

Developdata[taskId]: 任务 id

返回数据:

{"res": "1", "taskId": "df0eac7d-a6c2-4b30-93a8-a04e8006c7c2", "number": "9990005"}

字段名	说明
res	是否手动开始任务成功(1:成功 0:失败)
taskId	任务 id 原样返回
number	执行计划任务的虚拟 ID 编号, 广播时有效。

2.20 手动停止任务

路径: /php/exetaskcmd.php

调用参数: Developdata[taskCommand]: 'stoptaskinfo'

Developdata[taskId]: 任务 id

返回数据:

{"res": "1", "taskId": "df0eac7d-a6c2-4b30-93a8-a04e8006c7c2"}

字段名	说明
res	是否手动停止任务成功(1:成功 0:失败)
taskId	任务 id 原样返回

2.21 添加分区

路径: /php/zonectl.php

调用参数: Developdata[zonectl]: 固定"add"值表示添加

Developdata[newzonename]: 分区 ID + "_" + 分区名称 (如: 4_Group4)

Developdata[members]: 分区下的终端, 以逗号隔开 (如: ,1,2,3,)

Developdata[order]: 分区排序序号, 序号越大越靠前

返回数据:

{ "res": "1" }

字段名	说明
res	是否添加分区成功(1:成功 0:失败)

2.22 调用分区信息

路径: /php/getsetzonedata.php

调用参数: Developdata[pageIndex]: 页码序号 (从 0 开始)

Developdata[pageCount]: 每页数量 (如: 请求从 10-20 的数据, 则 pageIndex 为 1,

pageCount 为 10)

返回数据:



```
{ "res" : " 1" , " rows" : [ { "id" : " 1" , " name" : " Group1" , " member" : " 1" , " isbusy" : " 1" } , { "id" : " 2" , " name" : " Group2" , " member" : " 2" , " isbusy" : " 0" } , { "id" : " 3" , " name" : " Group3" , " member" : " 3,4,5" , " isbusy" : " 0" } ] , " total" : " 3" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)
id	分区编号
name	分区名称
member	分区下终端编号列表(以逗号分隔)
isbusy	1:忙碌 0:空闲
total	分区总数

2.23 编辑分区

路径: /php/zonemodify.php

调用参数: Developdata[oldname]: 原分区名: 分区 ID + " _" + 分区名称 (如: 4_Group4)
 Developdata[newname]: 新分区名: 分区 ID + " _" + 分区名称 (如: 4_Group4)
 Developdata[members]: 分区下的终端, 以逗号隔开 (如: ,1,2,3,)
 Developdata[order]: 分区排序序号, 序号越大越靠前

返回数据:

```
{ "res" : " 1" }
```

字段名	说明
res	是否修改分区成功(1:成功 0:失败)

2.24 删除分区

路径: /php/zonectl.php

调用参数: Developdata[zonectl]: 固定" del" 表示删除
 Developdata[newzonename]: 分区 ID + " _" + 分区名称 (如: 4_Group4)
 Developdata[members]: 为空

返回数据:

```
{ "res" : " 1" }
```

字段名	说明
res	是否删除分区成功(1:成功 0:失败)

2.25 查询录音日志

路径: /php/getrecorddata.php

调用参数: Developdata[pageIndex]: 页码序号 (从 0 开始)
 Developdata[pageCount]: 每页数量 (如: 请求从 10-20 的数据, 则 pageIndex 为 1, pageCount 为 10)
 Developdata[recordType]: 查询类型 (0: 全部 1: 对讲 2: 监听 3: 广播 4: 会议)
 Developdata[recordStart]: 开始时间 (如: 2018-01-09 00:00:00)
 Developdata[recordEnd]: 结束时间 (如: 2018-01-09 23:59:59)



Developdata[filter]: 终端 id 筛选信息 (如: 1); 如果要对源和目标进行同时筛选, 请用逗号隔开 (如 1,2)

返回数据:

```
{ "res": "1", "rows": [{ "rectype": "3", "source": "AppServer1(9990001)", "target": " - ", "starttime": "2018-01-09 16:29:00", "filesize": "0.43M", "shortname": "broadcast-9990001.2018-01-09-16-29-00.wav" }], "total": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)
rectype	录音文件类型(0:全部 1:对讲 2:监听 3:广播)
source	发起方
target	目标方
starttime	开始时间
filesize	文件大小
shortname	文件名 (注:链接全地址为 http://服务器IP/recordings/ + shortname)

2.26 调用终端扩展信息

路径: /php/getextdata.php

调用参数: Developdata[id]: 终端编号

返回数据:

```
{ "id": "1", "exttxt": "123", "extimg1": "1_1.png", "extimg2": "1_2.png", "extimg3": "1_3.png", "extimg4": "1_4.png" }
```

字段名	说明
id	终端编号
exttxt	扩展文本信息
extimg1	扩展图片信息 1 (实际路径为 http://服务器IP/memberext/ + extimg1)
extimg2	扩展图片信息 2
extimg3	扩展图片信息 3
extimg4	扩展图片信息 4

2.27 快速执行任务

路径: /php/exeRealPlayFile.php

调用参数:

Developdata[rtype]:	startbct(开始广播)	stopbct(停止广播)	getstatus(获取状态)	filecontrol(播放控
---------------------	----------------	---------------	-----------------	-----------------



				制)
Developdata[param 1]:	接收广播终端列表, 以<分隔 (如 1<2<3 表示对 1,2, 3 号终端进行广播)	要停止的广播编号, 为空表示停止所有广播, 请参考下方的返回值说明	要停止的广播编号, 请参考下方的返回值说明	要控制的广播号码, 请参考下方的返回值说明
Developdata[param 2]:	音源类型 (0: 文件; 1: 终端; 2: 声卡; 3: 文本) 【注: 2 声卡需要服务器安装声卡】	要停止的广播对应的 GUID 编号, 请参考下方的返回值说明	要停止的广播对应的 GUID 编号, 请参考下方的返回值说明	控制类型: play: 播放 pause: 暂停 next: 下一首 previous: 上一首
Developdata[param 3]	发起广播终端编号 (音源类型为“1”时有效, 音源类型为其他时请填写“0”)			
Developdata[param 4]	广播文本 (音源类型为“3”时有效, 音源类型为其他时请填写“test”)			
Developdata[param 5]	广播文件列表【音源类型为“0”时有效, 以<分隔 (如 \\1.mp3<\\sub\\2.mp3)】			
Developdata[param 6]	持续时间, 当持续时间大于文件时间时, 将循环播放【默认为 0】 (单位: 秒)			
Developdata[param 7]	广播文本播放次数 (音源类型为“3”时有效)			
Developdata[param 8]	广播文本速度【-10 到 10, 默认为 0, 正常语速】 (音源类型为“3”时有效)			
Developdata[param 9]	广播音源类型 (音源类型为“3”时			



	有效)			
Developdata[param10]	是否随机播放: 1: 随机播放 2: 顺序播放 (音源类 型为“0”时有效)			
Developdata[param11]	播放列表 ID, 如 test.list (默认 为空, 音源类型为 “0”时有效)	注: 播放列表 和文件列表同 时有值时, 以 播放列表优先		
Developdata[param12]	同步强切: 1			
Developdata[param13]	用户自有任务 id			
Developdata[extdata]	第三方传入数据, 可在推送接口中 返回			

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败) 查询状态时: 1表示查询成功 0表示不存在
number	开始广播调用返回的内部编号, 供停止广播时调用 【rtype= startbct 或 rtype= getstatus】
guid	开始广播调用返回的唯一编号, 供停止广播时调用 (程序先判断 guid, 再判断 number) 【rtype= startbct 或 rtype= getstatus】
extdata	extdata 原样返回
targets	接收广播终端列表 【rtype= getstatus】
srctype	音源类型 (0: 文件; 1: 终端; 2: 声卡; 3: 文本) 【rtype= getstatus】
srcid	发起广播终端编号 【rtype= getstatus】

2.28 控制终端短路输出

路径: /php/teriocontrol.php

调用参数: Developdata[terid]: 终端 ID

Developdata[ioport]: 短路输出端口号 (501 表示端口 1, 502 表示端口 2, 以此类推)

Developdata[isopen]: 1 表示闭合; 0 表示断开

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)



2.29 查询系统日志

路径: /php/getlogdata.php

调用参数: Developdata[logType]: 日志类型

(0: 全部; 1: 注册; 2: 对讲; 3: 监听; 4: 广播; 5: 报警; 6: 会议; 238: 操作)

Developdata[logStart]: 开始日期 (如 2018-07-10)

Developdata[logEnd]: 结束日期 (如 2018-07-10)

Developdata[filter]: 筛选内容, 可输入终端 ID 进行筛选

Developdata[pageIndex]: 页码序号 (从 0 开始)

Developdata[pageCount]: 每页数量 (如: 请求从 10-20 的数据, 则 pageIndex 为 1, pageCount 为 10)

返回数据

```
{ "res": "1", "rows": [{"time": "2018-07-10 11:22:33", "logtype": "1", "subtype": "1", "source": "1", "target": "", "reserve": ""}], "total": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)
time	日志记录时间
logtype	日志类型: (参考本方法传入值)
subtype	日志子类型 (0:全部; 1:终端登陆; 2:终端注销; 3:响铃; 4:接听; 5:激活; 6:IO报警; 7:刷卡上传; 8:停止; 9:远程播控; 224:用户登陆; 225:用户注销; 226:软件注册; 227:添加数据; 228:修改数据; 229:删除数据; 230:导出数据)
source	源 (logtype 为 238 操作时, 为操作用户名, 如 admin)
target	目标 (logtype 为 238 操作时, 为操作对象的类型信息, 具体定义如下: 1. 用户 2. 角色 3. 分区 4. 终端 5. 服务器级联 6. SDK 7. 呼叫队列 8. 系统配置 9. 媒体目录 10. 媒体文件 11. 任务方案 12. 任务 13. 备份 14. 扩展模块)



	15. 终端固件 16. 报告 17. 录音)
reserve	备注
total	日志总数

2.30 查询计划任务日志

路径: /php/controllogdata.php

调用参数: Developdata[taskId]: 任务编号

返回数据:

```
[{"time": "2018-07-10 11:22:33", "type": "1"}, {"time": "2018-07-10 11:22:44", "type": "3"}]
```

字段名	说明
time	任务执行时间
type	执行类型 (1: 自动运行; 2: 手动运行; 3: 自动停止; 4: 手动停止)

2.31 调用系统配置

路径: /php/getsysconfig.php

调用参数: 无

返回数据:

```
{"mediamode": "false", "bctmultiip": "235.1.0", "bctmultiport": "34567", "countdefine": "7", "outboundip": "auto-nat", "bcttermfix": "808", "bctareafix": "909", "logdays": "30", "recdays": "5", "limitseconds": "0", "realtasklevel": "14"}
```

字段名	说明
mediamode	对讲数据模式: false: 服务器中转模式 true: 直通模式
bctmultiip	组播服务器 IP
bctmultiport	组播服务器端口
countdefine	终端位数 (3-7)
outboundip	外网映射 IP (默认为自动获取 auto-nat)
bcttermfix	SIP 终端广播前缀
bctareafix	SIP 分区广播前缀
logdays	日志保存天数
recdays	录音保存天数
limitseconds	实时采播限制时间 (秒)
realtasklevel	实时采播优先级
enablerecord	0: 禁用录音 1: 启用录音
recorddir	录音路径



2.32 保存系统参数

路径: /php/setsysconfig.php

调用参数:

Developdata[mediamode]: false: 服务器中转模式 true: 直通模式
 Developdata[terminaldigit]: 终端位数 (3-7)
 Developdata[outboundip]: 外网映射 IP (默认为自动获取 auto-nat)
 Developdata[logdays]: 日志保存天数
 Developdata[recdays]: 录音保存天数
 Developdata[multiip]: 组播服务器 IP
 Developdata[multiport]: 组播服务器端口
 Developdata[zonesuffix]: SIP 分区广播前缀
 Developdata[terminalsuffix]: SIP 终端广播前缀
 Developdata[limitseconds]: 实时采播限制时间 (秒)
 Developdata[realtasklevel]: 实时采播优先级
 Developdata[enablerecord]: 0: 禁用录音 1: 启用录音
 Developdata[recorddir]: 录音路径 (路径需存在)

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

2.33 查询定时任务执行结果

路径: /php/querytaskresult.php

调用参数: Developdata[date]: 查询日期 (如: 2018-09-22)

返回数据:

```
[
  {
    "id": "9616f3e6-d88e-4ccf-a9b5-690539614fb5",
    "name": "上课铃",
    "disable": "0",
    "start": "2018-11-12 07:00:00",
    "durations": "10",
    "result": "0"
  },
  {
    "id": "3fb61cbe-f872-46f8-8d71-4578fc4d21ef",
    "name": "广播操",
    "disable": "0",
    "start": "2018-11-12 09:41:00",
    "durations": "314",
    "result": "0"
  }
]
```



字段名	说明
id	任务编号
name	任务名称
disable	1: 禁用 0: 启用
start	开始时间
durations	持续时间(秒)
result	1: 已执行 0: 未执行

2.34 修改服务器时间

路径: /php/setlocaltime.php

调用参数: Developdata[time]: 时间 (格式: 2018-09-22 10:45:33)

返回数据:

```
{"res": "1"}
```

字段名	说明
res	1: 修改成功; 0: 修改失败

(注: 非特殊定制项目请勿调用此接口)

2.35 终端广播控制

路径: /php/eavesdrop.php

调用参数: Developdata[source]: 发起方ID

Developdata[target]: 接收方ID

Developdata[isadd]: 1: 添加接收方 2: 删除接收方

Developdata[extdata]: 第3方传入数据

返回数据:

```
{"res": "1", "extdata": ""}
```

字段名	说明
res	1: 成功; 0: 失败
extdata	第3方传入数据原样返回

2.36 查询 HTTP SDK 接口列表

路径: /php/getsdkdata.php

调用参数: Developdata[pageIndex]: 页码序号 (从 0 开始)

Developdata[pageCount]: 每页数量 (如: 请求从 10-20 的数据, 则 pageIndex 为 1, pageCount 为 10)

Developdata[ishttp]: 1: 查询 http 类型接口 0: 查询兼容模式接口

返回数据:

```
{"res": "1", "rows": [{"id": "1", "url": "http://192.168.1.101/test.php"}], "total": "1"}
```

2.37 添加/修改/删除 HTTP SDK 接口

路径: /php/updatesdkhttp.php

调用参数: Developdata[url]: URL 地址, 如: http|//192.168.1.101/test.php (冒号用|代替)

Developdata[update]: 1: 添加或修改 0: 删除

Developdata[oldurl]: 原有 URL 地址, 修改时有效, 同 url 格式



返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

注: 当为删除操作时, url 为*表示删除所有 http sdk 列表

2.38 查询设备 IO 报警状态

路径: /php/terioquery.php

调用参数: Developdata[terid]: 终端编号
Developdata[ioport]: IO端口号

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

注: 设备 IO 报警是异步返回的, 此条指令只做通知, IO 报警状态通知统一在推送消息

2.39 启用/禁用计划任务

路径: /php/exetaskenable.php

调用参数: Developdata[isDisable]: 1: 禁用 0: 启用
Developdata[taskId]: 任务编号

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

2.40 通话控制

路径: /php/sessioncontrol.php

调用参数:

Developdata[cmdtype]:	hangup: 通话强拆 hold: 保持通话 unhold: 取消保持 eavesdrop: 监听通话 insert: 强插通话 transfer_caller: 转移主叫 transfer_callee: 转移被叫
Developdata[srcid]:	源终端 ID
Developdata[tarid]:	目标终端 ID
Developdata[ishost]	1: 主叫 0: 被叫
Developdata[rtype]	0: 对讲 1: 监听 2: 广播 3: 会议
Developdata[channelId]	通道编号 (非必选, 在已知通道号的情况下可传入此值)



Developdata[otherId]	外部介入终端编号（监听，强插，转移时有效）
----------------------	-----------------------

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

2.41 查询实时状态

路径: /php/getreportdata.php

调用参数: Developdata[reportType]: 类型

(0: 全部; 1: 注册; 2: 对讲; 3: 监听; 4: 广播; 5: 报警; 6: 会议;)

Developdata[filter]: 筛选信息（可输入终端 ID, 默认为空）

Developdata[pageIndex]: 页码序号（从 0 开始）

Developdata[pageCount]: 每页数量（如：请求从 10-20 的数据，则 pageIndex 为 1, pageCount 为 10）

返回数据

```
{ "res": "1", "rows": [
    { "id": "d1d6f68a", "type": "1", "time": "2019-07-05
    11:47:07", "source": "Terminal7(7)", "target": "", "state": "1", "sourceid": "7", "targetid": "", "i
    shost": "1" } ], "total": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)
id	唯一 ID 值
time	记录时间
type	类型:（参考本方法 reportType 传入值）
state	状态 (0:全部; 1: 终端登陆; 2: 终端注销; 3: 响铃; 4: 接听; 5: 激活; 6: IO 报警; 7: 刷卡上传; 8: 停止; 9: 远程播控;)
source	源名称
sourceid	源 ID
target	目标 (当 state “IO 报警” 时, 此状态下 targetid 值为 “报警类型_IO 端口号_IO 状态”)
targetid	目标 I D
ishost	1: 发起方 0: 接收方
total	总数量

2.42 处理实时状态

路径: /php/removereporrt.php

调用参数: Developdata[channelId]: 唯一 ID（请参考 2.69 返回 ID 值）

Developdata[reportType]: 类型（请参考 2.69 传入值）

返回数据:



{ "res": "1" }

字段名	说明
res	0: 失败 1: 成功

2.43 截取录音录像文件

路径: /php/cutrecord.php

调用参数: Developdata[id]: 发起方终端 ID

Developdata[recordType]: 录音类型 (0: 全部 1: 对讲 2: 监听 3: 广播 4: 会议)

Developdata[recordStart]: 开始截取时间, 格式如 2019-09-23 15:45:00

Developdata[recordEnd]: 停止截取时间, 格式如 2019-09-23 15:45:00

Developdata[newname]: 新文件名, 如 test.wav

返回数据:

{ "res": "1" }

字段名	说明
res	0: 失败 1: 成功

注: 返回成功后, 新文件的地址格式为 <http://服务器 IP/recordings/test.wav>

2.44 重启系统

路径: /php/main.php

调用参数: Developdata[servername]: "all", //默认重启所有服务器

Developdata[currentUser]: "admin" //当前用户

返回数据: { "res": "1" }

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

2.45 振铃组配置

路径: /php/handlinggroup.php

调用参数: Developdata[cmd]: 命令类型

- cmd为“query”时, 查询所有振铃组信息

返回数据:

```
{"rows":[{"id":"221","name":"221","isorder":"0","members":"2,3,"}, {"id":"222","name":"222","isorder":"0","members":"4,5,"}]}
```

字段名	说明
id	振铃组编号
name	振铃组名称
isorder	振铃方式: 0: 同时振铃 1: 顺序振铃
members	成员终端 ID 列表

- cmd为“edit”时, 进行振铃组的添加或修改操作

其他参数:

Developdata[id]: 振铃组编号 (不能与实际终端编号重复)

Developdata[name]: 振铃组名称



Developdata[isorder]: 振铃方式: 0: 同时振铃 1: 顺序振铃

Developdata[members]: 成员终端 ID 列表(以逗号分隔)

返回数据:

```
{"res": "1"}
```

字段名	说明
res	1: 成功; 0: 失败

● cmd 为 "delete" 时, 进行播放列表的删除操作

其他参数:

Developdata[id]: 振铃组编号

返回数据:

```
{"res": "1"}
```

字段名	说明
res	1: 成功; 0: 失败

2.46 发送手机短信

路径: /php/sendsmsmessage.php

调用参数: Developdata[target]: 目标号码 (多个以逗号分隔, 如 13800001111,18922223333)

Developdata[context]: 发送内容

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

注: 此接口为预留接口。

2.47 发送语音通知

路径: /php/sendaudionotify.php

调用参数: Developdata[target]: 目标终端 (多个以逗号分隔)

Developdata[srctype]: 音频源类型 (0: 媒体库文件 1: 文本)

Developdata[context]: srctype 为 0 时, 为媒体库文件路径

srctype 为 1 时, 为文本内容

返回数据:

```
{ "res": "1" }
```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)

注: 此接口为预留接口。

2.48 添加离线任务

路径: /php/addofflinetask.php

调用参数:

Developdata[taskId]: 任务 Id

Developdata[target]: 目标终端 Id (如 1<2<3)

返回数据



```
{"res": "1"}
```

备注：接口功能为 将选择的任务推送到指定的终端

2.49 删除离线任务

路径： /php/delofflinetask.php

调用参数：

Developdata[taskId]: 任务 Id

Developdata[target]: 目标终端 Id (如 1<2<3)

返回数据

```
{"res": "1"}
```

备注：接口功能为 删除终端上的离线任务

2.50 查询终端离线任务(发送查询命令)

路径： /php/sendofflinequery.php

调用参数：

Developdata[target]: 目标终端 Id (只能单个查询)

返回数据

```
{"res": "1"}
```

备注：接口功能为 服务器向终端查询离线任务，终端会通过异步的方式将该终端的离线任务推送到服务器，用于 2.94 接口查询

2.51 查询终端离线任务(查询分页结果)

路径： /php/replyofflinedata.php

调用参数：

Developdata[target]: 目标终端 Id (只能单个查询)

Developdata[pageIndex]: page 页号 (如 1) (注：PageIndex 应该是从 0 开始计数)

Developdata[pageCount]: 分页大小 (如 30)

返回数据

```
{"res": "1", "rows": [{"id": "53a40581-30bf-4c89-870f-b5ea8621eb99", "name": "第一个任务", "description": "第一个描述"}, {"id": "53a40581-30bf-4c89-870f-b5ea8621eba9", "name": "第二个任务", "description": "描述 2"}], "total": "2"}
```

备注：接口功能为 查询终端的离线任务，调用此接口之前必须先调用 2.93 接口，待服务器收到终端返回的任务数据后才能查询，所以建议调用 2.93 接口 3 秒后再调用此接口。

2.52 调用自检任务列表

路径： /php/selfchecktasklist.php

调用参数：

Developdata[user]: 当前用户名

Developdata[pageIndex]: 分页号

Developdata[pagecount]: 每页数量

Developdata[taskname]: 任务名称 (模糊查询)

返回数据：

```
{"res": "1", "rows": [{"selfchecktaskId": "ec5b2e30-1c99-4c39-895c-7f8ab686399f", "taskname": "测
```



试检测

```
","checktypes":"checkvideo,checkaudio","terminalids":"1,2","checkstarttime":"2020/12/6  
12:32:00","createtime":"2020/12/7  
19:05:00","createby":"admin","modifyby":"","modifytime":"","tasktype":"1"}],"total":"0"}
```

2.53 新增自检任务

路径: /php/addselfchecktask.php

调用参数:

Developdata[user]: 当前用户名
Developdata[taskname]: 任务名称
Developdata[tasktype]: 任务类型 (0:单次, 1, 每天, 2, 每周, 3, 每月)
Developdata[checktype]: 检查类型[多个类型(checkvideo:检查视频, checkaudio:

检查音频,

checknetwork: 检查网络, checkkey: 检查按键,
checkktamper: 检查防拆, checkfingerprint: 检查指纹,
checkcard: 检查刷卡)用,分割]

Developdata[terminalids]: 需要检查的终端 id (多个用,分割)

Developdata[checkstarttime]: 开始检查时间

返回数据:

```
{"res":"1"}
```

2.54 修改自检任务

路径: /php/editselfchecktask.php

调用参数:

Developdata[taskid]: 任务 id
Developdata[user]: 当前用户名
Developdata[taskname]: 任务名称
Developdata[tasktype]: 任务类型 (0:单次, 1, 每天, 2, 每周, 3, 每月)
Developdata[checktype]: 检查类型[多个类型(checkvideo:检查视频, checkaudio:

检查音频,

checknetwork: 检查网络, checkkey: 检查按键,
checkktamper: 检查防拆, checkfingerprint: 检查指纹,
checkcard: 检查刷卡)用,分割]

Developdata[terminalids]: 需要检查的终端 id (多个用,分割)

Developdata[checkstarttime]: 开始检查时间

返回数据:

```
{"res":"1"}
```

2.55 删除自检任务

路径: /php/deleteselfchecktask.php

调用参数:

Developdata[taskid]: 任务 id

返回数据:

```
{"res":"1"}
```



2.56 查询终端自检记录

路径: /php/queryselfchecklogbyid.php

调用参数:

Developdata[terminalid]: 终端 id
Developdata[user]: 当前用户名

返回数据:

```
{"res": "1", "rows": [{"id": "1", "selfcheckid": "00000000-0000-0000-0000-000000000000", "terminalid": "1", "terminalname": "Terminal1", "checktype": "checkvideo", "checkresult": "1", "checktime": "2020/12/7 19:03:29"}, {"id": "4", "selfcheckid": "00000000-0000-0000-0000-000000000000", "terminalid": "1", "terminalname": "Terminal1", "checktype": "checkaudio", "checkresult": "1", "checktime": "2020/12/7 19:03:29"}], "total": "0"}
```

2.57 手动运行终端自检

路径: /php/manualsefchecktask.php

调用参数:

Developdata[taskid]: 自检任务 id
Developdata[user]: 当前用户名

返回数据:

```
{"res": "1"}
```

2.58 查询终端自检记录排名

路径: /php/previewselfselfcheck_terminalorder.php

调用参数:

Developdata[user]: 当前用户名
Developdata[starttime]: 开始时间
Developdata[endtime]: 结束时间
Developdata[ordercount]: 排名数量

返回数据:

```
{"res": "1", "rows": [{"terminalid": "1", "terminalname": "Terminal1", "videoerrcount": "3", "audioerrorcount": "5", "networkerrorcount": "5", "lastchecktime": "2020-12-07 19:03:33"}, {"terminalid": "2", "terminalname": "Terminal2", "videoerrcount": "4", "audioerrorcount": "3", "networkerrorcount": "3", "lastchecktime": "2020-12-07 19:03:33"}, {"terminalid": "3", "terminalname": "Terminal3", "videoerrcount": "2", "audioerrorcount": "4", "networkerrorcount": "6", "lastchecktime": "2020-12-07 19:03:33"}]}
```

2.59 调用通讯录

路径: /php/gettelephonenumber.php

调用参数: Developdata[user]: 当前登录用户用户名



返回数据:

```
{ "res" :1, " rows" :[  
    { "name" : " 张三" , " phone" : " 13838389438" },  
    { "name" : " 李四" , " phone" : " 13838389438" }]  
}
```

2.60 模板保存

路径: /php/savetelevoicetemp.php

调用参数:

Developdata[tempname]:	模板名称
Developdata[port]:	端口号
Developdata[enablerepeat]:	允许重听
Developdata[isneedconfirm]:	是否需要回执
Developdata[contenttext]:	语音合成文字
Developdata[playfilename]:	语音文件
Developdata[usernames]:	收件人名称
Developdata[phonenumbers]:	收件人号码
Developdata[user]:	收件人号码

返回数据:

```
{ "res" :1}
```

2.61 调用无线语音网关模板

路径: /php/gettelevoicetemplist.php

调用参数:

Developdata[user]: 当前登录用户用户名

返回数据:

```
{ "res" :1, " rows" :[{"Port": "7", "EnableRepeat": "1", "IsNeedConfirm": "1", "ContentText": "这里是文字内容", "ContentReleaseFileName": "生成的文件名称", "PlayFileName": "语音文件", "Name": "名称列表", "PhoneNumber": "电话列表"}, {"Port": "8", "EnableRepeat": "1", "IsNeedConfirm": "1", "ContentText": "这里是文字内", "ContentReleaseFileName": "生成的文件名称", "PlayFileName": "语音文件", "Name": "名称列表", "PhoneNumber": "电话列表"}]}
```

2.62 调用无线语音网关模板详情

路径: /php/gettelevoicetempdetail.php

调用参数:

Developdata[user]:	用户名
Developdata[televoicetmpname]:	模板名称

返回数据:

```
{"Port": "7", "EnableRepeat": "1", "IsNeedConfirm": "1", "ContentText": "这里是文字内容", "ContentReleaseFileName": "生成的文件名称", "PlayFileName": "语音文件", "Name": "名称列表", "PhoneNumber": "电话列表"}
```




2.63 调用无线语音网关语音通知

路径: /php/gettelevoicedetails.php

调用参数:

Developdata[notifyno]: 通知编号
Developdata[user]: 当前用户

返回数据:

```
{ "res": "1", "rows": [{"id": "{0}", "notifyno": "{202008281}", "name": "张三", "phonenum": "18620114587", "repeattimes": "1", "sendport": "0", "sendstate": "1", "confirmstate": "1", "confirmtime": "2020-08-28 14:25:33", "isneedconfirm": "1"}]}
```

2.64 调用无线语音网关语音系统设置

路径: /php/savetelevoiceconfig.php

调用参数:

Developdata[confirmkey]: 确认键
Developdata[cancelconfirmkey]: 取消键
Developdata[repeatkey]: 重听键
Developdata[recallduration]: 重呼间隔时间 s
Developdata[recalltimesforconfirm]: 没有确认重复次数
Developdata[recalltimesforfailed]: 失败重呼次数

返回数据:

```
{ "res": "1" }
```

2.65 调用无线语音网关模板删除命令

路径: /php/deletetelvoicetmp.php

调用参数:

Developdata[user]

返回数据:

```
{"res": "1", "lastupdatetime": "2020-09-02 14:22:35"}
```

2.66 调用无线语音网关重新呼叫命令

路径: /php/resenttelvoice.php

调用参数:

Developdata[notifyno]: 通知编号
Developdata[name]: 通知人姓名
Developdata[phonenum]: 通知人电话
Developdata[port]: 端口

返回数据:

```
{ "res": "1" }
```

```
{"res": "0", "errcode": "0"}
```

备注: errorcode: 0, 网关正忙; 1, 传入数据有误; 2, 其他异常



2.67 调用无线语音网关自动录音开关命令

路径: /php/getvoicerecording.php

调用参数:

Developdata[user]: admin 管理员账号

返回数据:

```
{"res": "1", "enablerecord": "0"}
```

2.68 调用无线语音网关设置录音开关

路径: /php/setvoicerecording.php

调用参数:

Developdata[enablerecord]: 1 开关

Developdata[user]: admin 管理员账号

返回数据:

```
{"res": "1"}
```

2.69 终端升级

路径: /php/sendfirmware.php

调用参数:

Developdata["device"]: 终端类型, 上传终端固件时添加

Developdata["target"]: 待升级的目标终端

Developdata["currentUser"]: 当前用户

返回数据:

```
{"res": "1"}
```

备注: 服务器会向终端发送一条MSG消息, 告知终端开始升级

2.70 调用报警数据

路径: /php/getalarmdata.php

调用参数: pageIndex: 页码序号 (从0开始)

pageCount: 每页数量 (如: 请求从10-20的数据, 则pageIndex为1, pageCount 为10)

termid: 需要查询的终端ID编号 如需查询全部传空

isnew: 是否只查询最新的一条数据:1:仅查询最新的数据 0:查询所有

返回数据:

```
{
  "res": "1",
  "rows": [
    {
      "uuid": "8e0e4fdc-e0e4-49af-921a-b84635de3bfb",
      "id": "2",
      "name": "2",
      "time": "2022/3/29 18:49:44",
      "port": "6_502",
      "state": "0"
    }
  ]
}
```



```

    }
  ],
  "total": "1"
}

```

字段名	说明
res	是否调用接口成功(1:成功 0:失败)
uuid	告警消息的唯一 ID 值
id	终端 ID
time	时间
port	报警类型_端口号
state	状态, 0:关闭, 1:开启

3. 上传消息

3.1 在服务器的 SDK 配置页面里增加 Http 推送地址 S:

3.2 推送数据方式为 POST 方法, Content-Type 为 “application/x-www-form-urlencoded” 格式, 推送的数据格式均为 Developdata[xxx] 格式, 具体格式请参照下一节的表格;

若开启了推送确认功能(系统默认不开启), 第三方应用在接收消息后需要按照规定的格式返回 DEVELOP 报文, 格式如下:

```

{
  "res": "1"          // res 为 1 表示接收数据成功, 否则系统会重发数据 3 次
}

```

3.3 推送数据参数详细说明:

Developdata[uuid]: 唯一标识 ID

Developdata[type]: 事件类型 (具体参数值请参照下方表格详细说明)

Developdata[time]: 事件时间

Developdata[sourceid]: 发起方设备 ID

Developdata[targetid]: 目标方设备 ID (多个目标以“-”分隔)

Developdata[state]: 状态 (具体参数值请参照下方表格详细说明)

Developdata[file]: 事件相关录音文件 (对讲/监听/广播类型开始时此字段有值)

注: 对应的 http 全路径为 <http://服务器 IP /recordings/> + Developdata[file]

Developdata[sourcename]: 发起方设备名称

Developdata[targetname]: 目标方设备名称

Developdata[sourceext]: 发起方设备扩展信息

Developdata[targetext]: 目标方设备扩展信息 (为 DEVELOP 数组格式)

Developdata[extdata]: 方法 2.5/2.27 中传入的第三方平台数据

Developdata[taskid]: 当任务为计划任务时, 计划任务的编号

Developdata[playlistid]: 当任务为文件广播时, 为播放列表的 ID (如 test.list) 【若未指定播放列表, 则为当前任务播放的文件名称】

Developdata[ishost]: 是否为发起方 (1: 发起方 0: 接收方)

Developdata[sourcepanel]: 发起方设备面板号

Developdata[targetpanel]: 接收方面板号



Developdata[tasktype]:0: 计划任务, 1: 实时任务

事件类型[type]	状态[state]
1. 设备注册	1: 登录 2: 注销 (targetid 为面板编号)
2. 设备对讲	0: 停止 1: 呼叫 2: 对讲中
3. 设备监听	0: 停止 1: 呼叫 2: 对讲中
4. 设备广播	0: 停止 2: 接收广播中 3: 发起广播中
5. 设备报警	<p>6: IO 口报警 (此状态下 targetid 值为 “报警类型_IO 端口号_IO 状态”) 报警类型: 1: IO 输入报警 (端口号: 1~99 其中端口为 9 时 报警类型为 13: 紧急报警; 为 10 时 报警类型为 14: 移动侦测报警; 为 11 或 12 时 报警类型为 15: 门磁报警) 2: 防拆报警 (端口号: 101~119) 3: 回路检测报警 (端口号: 121~139) 4: 喧哗报警 (端口号: 141~159) 5: 巡更报警 (端口号: 161~179) 6: IO 输出报警 (端口号: 501~599) 7: 功放报警 (端口号: 601~619) 8: 分区报警 (端口号: 621~639) 9: 电源报警 (端口号: 641~659) 10: 短路报警 (端口号: 181~199) 11: 开路报警 (端口号: 201~219) 12: 接地报警 (端口号: 221~239) 16: 关键字报警 (端口号: 241~259) 17: 拾音器报警 (端口号: 260~299) 状态: 1: 闭合 0: 断开</p> <p>7: 设备刷卡 (此状态下 targetid 值为 “卡号”)</p>
1. 设备会议	<p>3: 主持方发起会议 2: 参会方加入会议 0: 主持方/参会方退出会议</p>
2. 任务状态	<p>0: 空闲 1: 自动启动 2: 手动启动 3: 自动停止 4: 手动停止</p>
8. 会见状态	<p>1: 开始会见 2: 取消会见</p>
184. 文件转换成功	<p>0: 格式转换 1: 文件截取 Developdata[sourcename]: 源文件 Developdata[targetname]: 转换后目标文件</p>



185. 接收到外部短信消息
(注:此消息为预留接口)

sondata[sourcenam]: 发送号码
Developdata[targetname]: 短信消息内容