

API 开发手册

千量级 IPPBX

版本: 1.1.0

日期: 2025年10月11日



目录

API 接口概览.....	1
API 快速入门指南.....	6
启用 API 功能.....	6
为 REST API 资源启用 CORS.....	7
删除被拉黑的 API IP.....	8
通信方式.....	9
请求结构.....	11
响应消息.....	12
API 报告推送.....	13
API Demo.....	13
API 验证.....	15
登录.....	15
API 心跳包.....	17
退出登录.....	19
API 接口示例.....	21
查询 PBX 信息.....	21
分机.....	22
查询分机列表.....	22
查询分机设置.....	23
编辑分机设置.....	28
中继.....	32
查询中继列表.....	32
查询 SIP 中继.....	33
添加 SIP 中继.....	36
编辑 SIP 中继.....	38
删除 SIP 中继.....	40
呼入路由.....	41

查询呼入路由.....	41
添加呼入路由.....	43
编辑呼入路由.....	45
删除呼入路由.....	46
呼出路由.....	47
查询呼出路由.....	47
添加呼出路由.....	49
编辑呼出路由.....	50
删除呼出路由.....	51
IVR.....	52
查询 IVR 设置.....	53
编辑 IVR 设置.....	55
队列.....	56
查询队列的设置.....	56
编辑队列设置.....	59
查询队列状态.....	62
签入队列.....	65
签出队列.....	66
暂停坐席服务.....	66
恢复坐席服务.....	67
广播组.....	68
查询广播组列表.....	68
查询广播组.....	69
添加广播组.....	71
编辑广播组.....	73
删除广播组.....	75
发起广播.....	76
挂断广播音乐.....	79
语音通话.....	79
查询来电.....	79
查询去电.....	81
分机互拨.....	83
拨打外线号码.....	84

挂断分机（强拆）	85
挂断来电（强拆）	86
挂断去电（强拆）	87
通话保持.....	88
恢复通话.....	89
通话静音.....	89
取消静音.....	90
普通监听.....	91
密语监听.....	92
强插监听.....	94
通话转接.....	95
外线来电转接.....	96
外线去电转接.....	97
通过 IVR 拨打分机.....	98
通过IVR 拨打外线.....	100
自动拨号（队列）	101
自动拨号（响铃组）	102
双向外呼.....	103
来电接听控制.....	105
播放语音.....	107
给分机播放语音.....	107
给外线播放语音.....	108
语音留言.....	110
查询语音留言.....	110
删除语音留言.....	112
下载语音留言.....	113
下载录音文件.....	115
下载通话记录.....	117
酒店叫醒服务.....	119
添加闹铃.....	120
查询闹铃.....	121
更改闹铃设置.....	124
删除闹铃.....	125

API 报告示例.....	127
系统启动报告.....	127
配置变更报告.....	127
分机状态变更报告.....	128
‘响铃&回铃’ 报告.....	130
‘通话应答’ 报告.....	135
‘通话被应答’ 报告.....	137
‘通话结束’ 报告.....	139
‘呼叫转移’ 报告.....	142
‘呼叫失败’ 报告.....	144
‘按键信息’ 报告.....	146
‘通话记录’ 报告.....	150
‘来电呼叫请求’ 报告.....	153
‘来电呼入’ 报告.....	154
‘播放语音结束’ 报告.....	155
‘满意度调查结果’ 事件.....	158
错误代码.....	160

API 接口概览

千量级 IPPBX 提供 二次开发接口（API）。第三方服务器与 PBX 通过 API 对接，你可以在第三方服务器上查询和编辑 PBX 的设置、实现多种呼叫功能。PBX 还会主动发送系统报告和通话报告到第三方服务器。

PBX 版本要求

Yeastar S1000：固件版本80.4.0.41或更高版本

系统接口

功能	Endpoint
查询 PBX 信息	/api/v1.1.0/deviceinfo/query?token={token}

分机

功能	Endpoint
查询分机列表	/api/v1.1.0/extensionlist/query?token={token}
查询分机设置	/api/v1.1.0/extension/query?token={token}
编辑分机设置	/api/v1.1.0/extension/update?token={token}

中继

功能	Endpoint
查询中继列表	/api/v1.1.0/trunklist/query?token={token}
查询 SIP 中继	/api/v1.1.0/siptrunk/query?token={token}
添加 SIP 中继	/api/v1.1.0/siptrunk/add?token={token}
编辑 SIP 中继	/api/v1.1.0/siptrunk/update?token={token}
删除 SIP 中继	/api/v1.1.0/siptrunk/delete?token=#token#

呼入路由

功能	Endpoint
查询呼入路由	/api/v1.1.0/inroute/query?token={token}
添加呼入路由	/api/v1.1.0/inroute/add?token={token}

功能	Endpoint
编辑呼入路由	/api/v1.1.0/inroute/update?token={token}
删除呼入路由	/api/v1.1.0/inroute/delete?token={token}

呼出路由

功能	Endpoint
查询呼出路由	/api/v1.1.0/outroute/query?token={token}
添加呼出路由	/api/v1.1.0/outroute/add?token={token}
编辑呼出路由	/api/v1.1.0/outroute/update?token={token}
删除呼出路由	/api/v1.1.0/outroute/delete?token={token}

IVR

功能	Endpoint
查询 IVR 设置	/api/v1.1.0/ivr/query?token={token}
编辑 IVR 设置	/api/v1.1.0/ivr/update?token={token}

队列

功能	Endpoint
查询队列设置	/api/v1.1.0/queue/query?token={token}
编辑队列设置	/api/v1.1.0/queue/update?token={token}
查询队列状态	/api/v1.1.0/queuestatus/query?token={token}
签入队列	/api/v1.1.0/queue/add_dynamicagent?token={token}
签出队列	/api/v1.1.0/queue/del_dynamicagent?token={token}
暂停坐席服务	/api/v1.1.0/queue/pause_agent?token={token}
恢复坐席服务	/api/v1.1.0/queue/unpause_agent?token={token}

广播组

功能	Endpoint
查询广播组列表	/api/v1.1.0/paginggrouplist/query?token={token}
查询广播组	/api/v1.1.0/paginggroup/query?token={token}
添加广播组	/api/v1.1.0/paginggroup/add?token={token}

功能	Endpoint
编辑广播组	/api/v1.1.0/paginggroup/update?token={token}
删除广播组	/api/v1.1.0/paginggroup/delete?token={token}
发起广播	/api/v1.1.0/extension/dial_number?token={token}
挂断广播音乐	/api/v1.1.0/paginggroup/hangup_music?token={token}

呼叫功能

功能	Endpoint
查询来电	/api/v1.1.0/inbound/query?token={token}
查询去电	/api/v1.1.0/outbound/query?token={token}
分机互拨	/api/v1.1.0/extension/dial_extension?token={token}
拨打外线号码	/api/v1.1.0/extension/dial_outbound?token={token}
挂断分机 (强拆)	/api/v1.1.0/extension/hangup?token={token}
挂断来电 (强拆)	/api/v1.1.0/inbound/hangup?token={token}
挂断去电 (强拆)	/api/v1.1.0/outbound/hangup?token={token}
通话保持	/api/v1.1.0/extension/hold?token={token}
恢复通话	/api/v1.1.0/extension/unhold?token={token}
通话静音	/api/v1.1.0/extension/mute?token={token}
取消静音	/api/v1.1.0/extension/unmute?token={token}
普通监听	/api/v1.1.0/extension/listen?token={token}
密语监听	/api/v1.1.0/extension/whisper?token={token}
强插监听	/api/v1.1.0/extension/barge?token={token}
通话转接	/api/v1.1.0/calltransfer?token={token}
外线来电转接	/api/v1.1.0/inbound/transfer_number?token={token}
外线去电转接	/api/v1.1.0/outbound/transfer_number?token={token}
通过 IVR 拨打分机	/api/v1.1.0/ivr/dial_extension?token={token}
通过 IVR 拨打外线	/api/v1.1.0/ivr/dial_outbound?token={token}
自动拨号 (队列)	/api/v1.1.0/queue/dial_outbound?token={token}
自动拨号 (响铃组)	/api/v1.1.0/ringgroup/dial_outbound?token={token}
双向外呼	/api/v1.1.0/outbound/dial_outbound?token={token}
接听来电	/api/v1.1.0/inbound/accept?token={token}

功能	Endpoint
拒接来电	/api/v1.1.0/inbound/refuse?token={token}

播放语音

功能	Endpoint
给分机播放语音	/api/v1.1.0/extension/playprompt?token={token}
给外线播放语音	/api/v1.1.0/outbound/playprompt?token={token}

语音留言

功能	Endpoint
查询语音留言	/api/v1.1.0/voicemail/query?token={token}
删除语音留言	/api/v1.1.0/voicemail/delete?token={token}
下载语音留言	/api/v1.1.0/voicemail/get_random?token={token}

通话记录和录音

功能	Endpoint
下载录音文件	/api/v1.1.0/recording/get_random?token={token}
下载通话记录	/api/v1.1.0/cdr/get_random?token={token}

酒店相关

功能	Endpoint
添加闹钟	/api/v1.1.0/wakeupcall/create?token={token}
查询闹钟	/api/v1.1.0/wakeupcall/query?token={token}
更改闹钟设置	/api/v1.1.0/wakeupcall/update?token={token}
删除闹钟	/api/v1.1.0/wakeupcall/delete?token={token}

API 报告

- [系统启动报告](#)
- [配置变更报告](#)
- [分机状态变更报告](#)
- [‘响铃&回铃’报告](#)

- [‘通话应答’ 报告](#)
- [‘通话被应答’ 报告](#)
- [‘通话结束’ 报告](#)
- [‘呼叫转移’ 报告](#)
- [‘呼叫失败’ 报告](#)
- [‘按键信息’ 报告](#)
- [‘通话记录’ 报告](#)
- [‘来电呼叫请求’ 报告](#)
- [‘来电呼入’ 报告](#)
- [‘播放语音结束’ 报告](#)
- [‘满意度调查结果’ 事件](#)

API 快速入门指南

使用千量级 IPPBX API 之前，你需要先启用 PBX 的 API 功能、了解千量级 IPPBX API 的通信方式、API 请求结构、响应消息和 API 报告。

启用 API 功能

在连接第三方应用服务器与 PBX 之前，你要先启用 PBX 的 API 接口，并做好相应的配置。

1. 登录 PBX 网页，进入 **配置 > PBX > 常规设置 > 二次开发接口**。
2. 启用 API 功能。

- a. 勾选 **启用**。
- b. 设置 **用户名** 和 **密码**，点击 **保存** 和 **应用**。



注：

此处的用户名和密码是第三方应用通过 API 连接到 IPPBX 时所要用到的用户名及密码。

常规设置	特征码	语音信箱	SIP	IAX	二次开发接口
☒ 启用 ⓘ					
用户名 ⓘ:		call-center			
密码 ⓘ:		G5gVz84s0skW		保存	

3. 选择是否监控分机的状态。

当该分机状态有变更时（如：分机状态从忙变为空闲），API 会主动向第三方应用发送分机状态变更的事件报告。

分机	中继	分机, 名称	
分机号	分机名	状态监控	☒
1000	Nancy		☒
1001	Willy		☒
1002	hxp		☒
2000	Eric		☒

4. 选择是否监控中继状态和控制来电。

分机	中继	名称	
中继名	来电接听控制	呼入目的地控制	状态监控
SIP Trunk1	☐	☐	☒
SIP Trunk2	☐	☐	☒

- **来电接听控制**：勾选来电接听控制之后，当该条中继有来电时先交由第三方应用处理，第三方应用可选择接受或者拒绝该来电。

- **接受**：如果第三方应用选择接受该来电，则该来电直接进入对应呼入路由目的地。

- **拒绝**：如果第三方应用选择拒绝该来电，则该来电将被直接挂断，而不会到达 PBX，PBX 上也不会有该来电的通话记录。

PBX 将留有10秒给第三方应用处理的时间，超时则默认第三方应用接受该来电，则该来电直接进入对应呼入路由的目的地。

- **呼入目的地控制**：勾选呼入目的地控制后，当该条中继有来电时将不会直接进入对应呼入路由的目的地，而是有10秒的时间等待第三方应用指定该通通话的目的地，如果10秒超时第三方应用未指定该通话目的地，则默认进入对应呼入路由所设置的目的地。
- **状态监控**：勾选中继的状态监控后，当该中继状态有变更时（如：中继状态从注册上变为注册失败），API 会主动向第三方应用发送中继状态变更的事件报告。

为 REST API 资源启用 CORS

跨源资源共享 (CORS) 是一项浏览器安全特征，浏览器限制脚本内发起跨源 HTTP 请求。如果你无法访问 API 并收到错误响应提示 CORS 跨域错误，那么你需要启用 CORS。本文介绍如何为简单请求启用 CORS。

示例

我们以 `Login` 接口为例介绍如何在 JavaScript 中允许跨域请求。

```
//指定请求 URL 地址
const url = "http://192.168.5.150:8088/api/v1.1.0/login";

//设置请求头中的 Content-type 字段值
const myHeaders = new Headers();
myHeaders.append("Content-Type",
  "application/x-www-form-urlencoded"); //此处指定的值仅限于
//"application/x-www-form-urlencoded"、“multipart/form-data” 或
//"text/plain”。此类请求会被认为是一个简单请求，从而避免预检请求的发送，简化跨域请求的过程。

//设置 body 参数
const raw = JSON.stringify({
  "username": "api",
  "password": "2d7257a528679d01a19c70e3fa773620",
  "port": "8260",
});

//设置请求选项
const requestOptions = {
  method: "POST",
  mode: "cors", //表示允许跨域请求。
  headers: myHeaders,
  body: raw,
  credentials: "include", //无论跨域还是同源都会带cookie。
};

//本例中，使用 fetch() 方法传入指定参数向指定 URL 发起 HTTP 请求。
fetch(http://192.168.5.150:8088/api/v1.1.0/login, requestOptions)
  .then((response) => response.text())
  .then((result) => console.log(result))
  .catch((error) => console.error(error));
```

删除被拉黑的 API IP

如果某个 IP 在短时间内，多次尝试登陆 API 接口失败，PBX 会将该 IP 拉入禁止名单，并将该 IP 类型标记为“API-IP”。如果该 IP 为受信任的 IP，你可以将这个 IP 从禁止名单列表中删除。

1. 进入 **PBX > 系统 > 安全中心 > IP自动防御 > IP禁止名单**，查看被拉黑的 IP。

防火墙规则	IP自动防御	服务	证书	数据库授权			
自动防御规则	IP禁止名单						
删除							
☐	类型	被攻击时间	▼	协议	被攻击端口	源IP地址	删除
☐	API-IP	2018-10-19 14:37:44		TCP	8088	192.168.7.24(api,IP is block...	

2. 选择 IP 地址，点击

通信方式

千量级 IPPBX 和第三方应用服务器的通信是相互的，双方互为服务端和客户端。

通信协议和端口

请求和响应

千量级 IPPBX 支持使用 HTTPS 协议和 HTTP 协议进行 API 通信。



注：

为加强通信安全，我们建议你使用 HTTPS 协议，并上传专业证书机构制作的 HTTPS 证书到 PBX。

PBX 默认使用 HTTPS 网页访问协议，端口 8088，且启用 **80端口重定向**，你可以选择 HTTPS（端口 8088）或 HTTP（端口 80）协议进行 API 通信。

网页访问协议 ⓘ:	HTTPS
端口 ⓘ:	8088
☒ 从80端口重定向 ⓘ	

外网 HTTP 通信

在中国，如果你要通过外网连接第三方服务器和 PBX API，80 端口无法使用，需要备案信息。你可以更改 PBX 设置，使用其他端口。

登录 PBX 网页，进入 **配置 > 系统 > 安全中心 > 服务**，选择 **网络访问协议** 为 **HTTP**，将默认的 80 端口改为其他端口。

网页访问协议 ⓘ:	HTTP
端口 ⓘ:	8282

API 报告推送

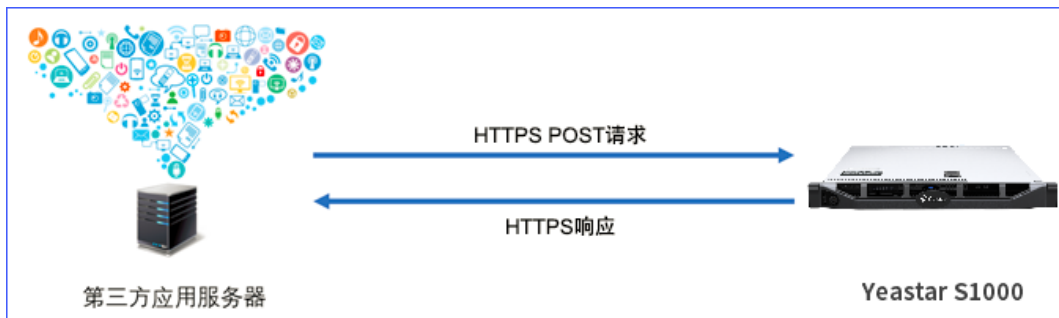
PBX 主动向第三方服务器 推送 API 报告使用 HTTP 协议。

通信方式

千量级 IPPBX 支持两种 API 通信方式。

1. 应用服务器向 PBX 发送请求

应用服务器为客户端，PBX 为服务端。应用场景：应用服务器向 PBX 发起呼叫、查询。



2. PBX 主动向服务器推送消息

前提条件：应用服务器需通过 PBX API 的验证，和 PBX API 建立连接。

应用服务器作为服务端，PBX 为客户端。



注：

第三方服务器接收到报告后，必须回复 200 OK 给 PBX，否则 PBX 重复发送报告。



请求结构

本文介绍千量级 IPPBX 的 API 请求结构，包括通信协议、请求方式和请求地址说明。

所有的 API 请求方式都是 POST。

下面为 [签入队列](#) 的请求示例。

```

POST /api/v1.1.0/queue/add_dynamicagent?token=#token# HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "queueid": "6700",
  "extid": "1000",
  "password": "6700123"
}
  
```

- POST 表示请求方式。
- /api/v1.1.0/queue/add_dynamicagent?token=#token# 表示 [签入队列](#) 的请求终端路径（Endpoint）。
- Content-Type:application/json; charset=utf-8 表示请求内容格式。
- Host 表示千量级 IPPBX 的 IP 地址。
- queueid、extid、password 表示具体请求参数。

请求地址

API 请求地址格式：

```
{base_url}{endpoint}
```

• Base URL

根据 PBX 的网页访问设置，填写 API 请求的 Base URL。

如果你使用 HTTPS 协议进行通信，Base URL 的格式如下：

```
https://{pbx_ip}:{https_port}
```

如果你使用 HTTP 协议进行通信，Base URL 的格式如下：

```
http://{pbx_ip}:{http_port}
```

• Endpoint

Endpoint（终点），也称作“路径”，表示 API 请求的具体地址。

点击[API 接口概览](#)，查看千量级 IPPBX 支持的所有 API endpoint。

响应消息

千量级 IPPBX 返回的响应消息为 JSON 格式。

不管是正常响应消息或异常响应消息，HTTP 状态码为 200 OK。

正常响应示例

接口调用成功后会返回“Success”状态，有时候还会返回相应的参数。我们称这样的响应为正常响应。HTTP 状态码为 200 OK。

```
HTTP/1.1 200 OK
{
  "status": "Success"
}
```

异常响应示例

接口调用出错后，会返回“Failed”状态和错误码。我们称这样的响应为异常响应。HTTP 状态码为 200 OK。

```
HTTP/1.1 200 OK
{
  "status": "Failed",
  "errno": "10003"
}
```

API 报告推送

第三方服务器通过 API 接口与千量级 IPPBX 对接后，PBX 会主动发送 API 报告给第三方服务器。

第三方服务器通过 API 接口 [登录](#) PBX，需要带 `port` 参数。`port` 表示 API 报告监听端口。PBX 将 API 报告发送到该端口。



注：

- 第三方服务器接收到报告后，必须回复 200 OK 给 PBX，否则 PBX 重复发送报告。
- 如果第三方服务器与 PBX 不在同一个网络，你需要将监听端口（默认 TCP 8260）映射到外网。
- 如果你需要在第三方服务器接收分机和中继相关的事件报告，你需要在 PBX API 配置界面，启用分机和中继的 **状态监控**。

API 报告格式

千量级 IPPBX 通过 POST 方式向第三方服务器发送 API 报告，报告格式为 JSON。

下面为 PBX 系统启动的 API 报告示例。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json
{
  "action": "BootUp",
  "sn": "369351034049"
}
```

API Demo

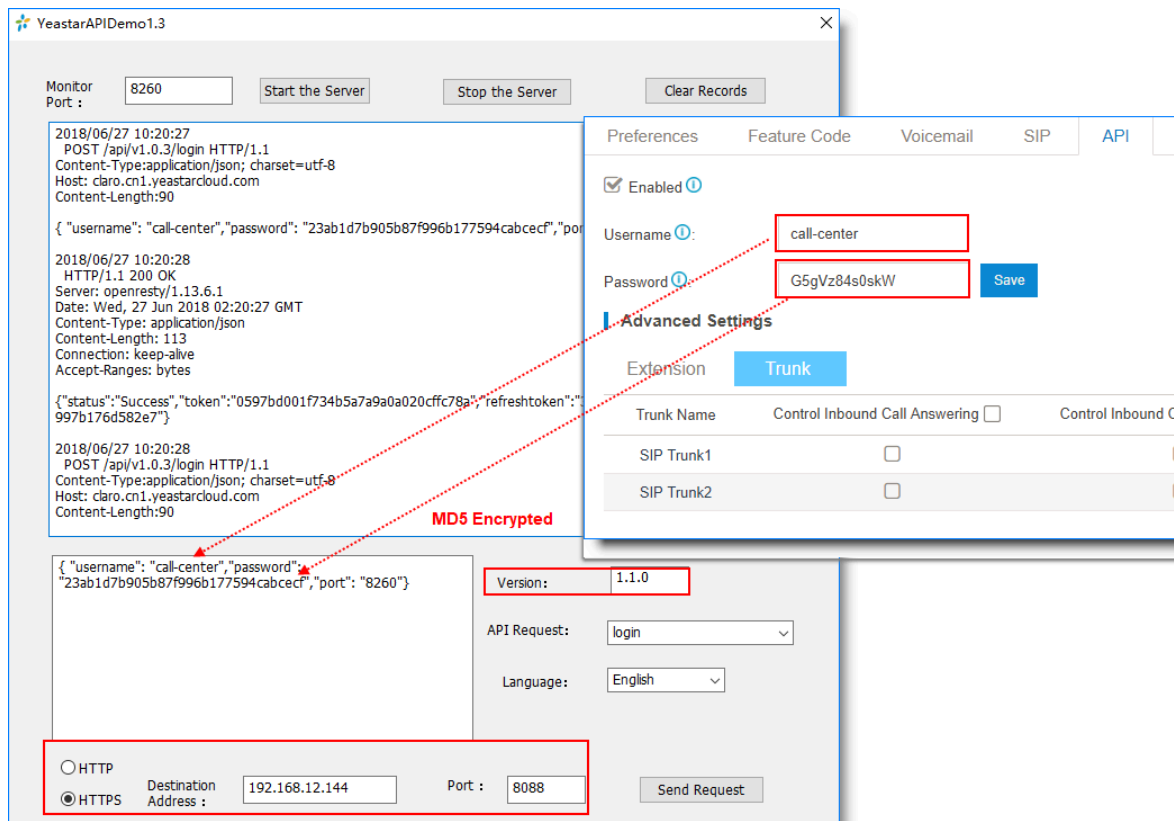
Yeastar提供 API demo 软件，你可以通过 API demo 软件测试 API 接口。

下载 API Demo

[点击下载 API demo](#)

如何使用 API Demo

1. 解压下载下来的 API Demo 压缩文件。
2. 双击 API Demo exe 文件。
3. 根据你的 PBX 的 API 设置，填写 API 登录信息。



- **Version:** 填写 1.1.0
 - **API Request:** 选择 login。
 - 选择通信协议：HTTPS。
 - **Destination Address:** 填写 PBX 的 IP 地址。
 - **Port:** 默认端口为 8088。
4. 在输入框内，根据 PBX 的 API 设置，更改 API 登录的 username 和 password。
 5. 点击 **Send Request**。
 6. API 登录成功后，你可以选择 **API Request** 类型，在左边的输入框内更改相应的请求信息，点击 **Send Request** 测试 API。

API 验证

登录

千量级 IPPBX API 采用用户名和密码的方式验证，只有用户名和密码验证通过的应用服务器，API 才会处理其发送的请求。



注：

- 你需要先将 API 的密码通过 MD5 加密，获取32位小写密码。使用加密过的密码进行 API 验证。
- 如果密码连续错误五次，则该第三方应用的 IP 将会被锁10分钟。

Endpoint

POST /api/v1.1.0/login

请求参数

名称	是否必需	类型	描述
username	是	String	在 PBX 二次开发接口界面设置的用户名。
password	是	String	将 PBX 二次开发接口界面设置的密码通过 MD5 加密。 填写加密后的32位小写密码。
port	是	String	端口号，此端口号为第三方应用用于监听 API 发送的事件报告的端口号。
url	否	String	第三方应用服务器获取 API 报告的 URL。 注： 如果不带此参数，则 PBX 默认向第三方应用服务器的 IP 地址发送 API 报告。
version	否	String	发送 API 事件报告格式的版本。 注：

名称	是否必需	类型	描述
			 设置"version":"1.0.2"可以修复 API 事件报告的语句不符合 JSON 格式的问题。
urltag	否	String	指定 URL 的类型： • 0：定义url的值为相对路径。PBX 会将相对路径的 url 与第三方服务器 IP 地址结合起来，发送 API 报告到具体的路径。 例如：第三方服务器 IP 地址为110.22.2.3；监听端口为8260；url设置为 report；则 PBX 发送 API 报告到110.22.2.3:8260/report。 • 1：定义url 的值为绝对路径。 例如：第三方服务器 IP 地址为110.22.2.3；url 设置为110.22.2.3:8260/report；则 PBX 发送 API 报告到110.22.2.3:8260/report。  注： 使用绝对路径，可以避免复杂网络环境下，API 报告无法发送成功的问题。

响应参数

名称	类型	描述
token	String	调用接口凭证，所有的 API 请求都需用到该 token。



注：

- API token 是第三方应用成功连接 PBX 后生成的。所有的 API 请求都需要带上 API 验证时所返回的token。
- token 有效时长为30分钟。

30分钟内如果 API 和第三方应用没有任何交互（如：第三方应用发送 API 请求），则该 token 将被 IPPBX 系统清除。可通过[心跳包](#)接口延长 token 的有效时长，每发送一次心跳包可使 token 的有效时长延长为30分钟。

实际示例

请求示例

```
POST /api/v1.1.0/login HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "username": "api",
  "password": "2d7257a528679d01a19c70e3fa773620",
  "port": "8260",
  "version": "1.0.2",
  "url": "110.22.2.3:8260/report",
  "urltag": "1"
}
```

响应示例

第三方应用服务器登录成功。

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "token": "48a7d7481a5355aa4fb5dc285edeb40e"
}
```

第三方应用服务器登录失败。

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Failed",
  "errno": "30003"
}
```

API 心跳包

当连接 PBX 的第三方应用的 IP 地址、监听端口或者 URL 有变更的时候，可通过 API 的心跳包接口对 IP、Port 和 URL 进行更新。

当第三方应用和 API 没有交互的时候，可利用此接口定期发送请求从而更新 token 的有效时长，避免 token 在 30分钟超时后被清除，造成事件不会上报以及请求失败。



注：



发送一次心跳包请求可使 token 的有效时长延长为30分钟。

Endpoint

POST /api/v1.1.0/heartbeat?token={token}

请求参数

名称	是否必需	类型	描述
ipaddr	是	String	第三方应用变更后的 IP 地址。
port	是	String	第三方应用变更后的监听端口。
url	否	String	第三方应用服务器获取 API 报告的 URL。  注： 如果不带此参数，则 PBX 默认向第三方应用服务器的 IP 地址发送 API 报告。
urltag	否	String	指定 URL 的类型： • 0：定义url 的值为相对路径。PBX 会将相对路径的url与第三方服务器 IP 地址结合起来，发送 API 报告到具体的路径。 例如：第三方服务器 IP 地址为192.168.5.122；监听端口为8260；url设置为report；则 PBX 发送 API 报告到192.168.5.122:8260/report。 • 1：定义url的值为绝对路径。 例如：第三方服务器 IP 地址为192.168.5.122；url 设置为192.168.5.122:8260/report；则 PBX 发送 API 报告到192.168.5.122:8260/report。  注： 使用绝对路径，可以避免复杂网络环境下，API 报告无法发送成功的问题。

实际示例

请求示例

第三方应用发送 API 心跳包对IP、Port 和 URL 进行更新。IP 地址变更为192.168.5.122，端口号变更为8280，URL 变更为192.168.5.122:8280/report。

```
POST /api/v1.1.0/heartbeat?token=277ac400357b509b4a587ff2157f7ad5 HTTP/1.1
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "url": "192.168.5.122:8260/report",urltag: "1"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

退出登录



注：

请求示例中的 token 为API 登录时返回的 token。

Endpoint

```
POST /api/v1.1.0/logout?token={token}
```

请求参数

无参数，直接发送退出登录的请求即可。

实际示例

请求示例

```
POST /api/v1.1.0/logout?token=277ac400357b509b4a587ff2157f7ad5 HTTP/1.1
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
```

响应示例

```
HTTP/1.1 200 OK
```

```
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

API 接口示例

查询 PBX 信息

Endpoint

```
POST /api/v1.1.0/deviceinfo/query?token={token}
```

请求参数

无参数，直接发送查询设备信息的请求即可。

响应参数

名称	类型	描述
devicename	String	产品设备名称。
sn	String	产品序列号。
firmwarever	String	固件版本号。
systemtime	String	系统时间。
uptime	String	启动时间。
extensionstatus	String	当前分机数/总分机数。

实际示例

请求示例

```
POST /api/v1.1.0/deviceinfo/query?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1 HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "deviceinfo": {
```

```
{  "devicename": "Yeastar S100",
  "sn": "369364835218",
  "hardwarever": "V1.30 0000-0000",
  "firmwarever": "30.5.0.19",
  "systemtime": "2017-05-24 23:25:20 UTC-8",
  "uptime": "03:25:04",
  "extensionstatus": "7/100"
}
```

分机

查询分机列表

查询 PBX 上分机列表的基本信息，如：分机名、分机号、分机状态、类型等。

Endpoint

```
POST /api/v1.1.0/extensionlist/query?token={token}
```

请求参数

无参数，直接发送查询分机列表的请求即可。

响应参数

名称	类型	描述
extlist	Object	分机对象。
extnumber	Int	分机号。
status	String	分机当前状态： • Unavailable：不可使用。 • Registered：已注册。 • Ringing：响铃。 • Busy：忙线。 • Hold：通话保持。 • Malfunction：故障。 • Idle：空闲。
type	String	分机类型。

名称	类型	描述
username	String	用户名。
agentid	String	报工号时要播报的号码。此参数默认为空，表示播报分机号。

实际示例

请求示例

```
POST /api/v1.1.0/extensionlist/query?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1 HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "extlist": [
    {
      "extnumber": "1000",
      "status": "Registered",
      "type": "SIP",
      "username": "Jayson",
      "agentid": ""
    },
    {
      "extnumber": "1001",
      "status": "Unavailable",
      "type": "SIP",
      "username": "Erwin Co",
      "agentid": ""
    }
  ]
}
```

查询分机设置

查询单个或者多个分机的详细信息，如：高级配置信息等。当发送查询多个分机请求的时候，请求参数需用逗号隔开。

Endpoint

POST /api/v1.1.0/extension/query?token=***{token}***

请求参数

名称	是否必需	类型	描述
extid	否	Int	分机号码。 • 查询单个分机：设置 extid 为分机号码。 例如： "extid": "1000" • 查询多个分机：设置 extid 为分机号码，多个分机之间用 , 隔开。 例如： "extid": "1000,1001" • 查询所有分机：不带 extid 请求参数。

响应参数

名称	类型	描述
extinfos	Object	对象参数。
extnumber	Int	分机号。
username	String	用户名。
status	String	分机当前状态： • Unavailable：不可使用。 • Registered：已注册。 • Ringing：响铃。 • Busy：忙线。 • Hold：通话保持。 • Malfunction：故障。 • Idle：空闲。
type	String	分机类型。
callerid	String	来电显示号码。
registername	String	注册名称。
registerpassword	String	注册密码。
maxregistrations	Int	同时注册数。
loginpassword	String	用户密码。
email	String	邮件地址。

名称	类型	描述
mobile	String	电话号码。
language	String	提示音语言。
hasvoicemail	String	启用语音邮箱。 • on: 开启。 • off: 关闭。
vmsecret	Int	语音邮箱密码。
enablevmtoemail	String	发送语音邮件到邮箱。 • on: 开启。 • off: 关闭。
alwaysforward	String	总是转移功能。 • on: 开启。 • off: 关闭。
atransferto atransferext atransferprefix atransfernum	String	总是转移目的地，如果选择总是转移到分机，则需设置要转移到的分机号；如果总是转移到外线号码，则需设置要转移到的外线号码以及呼叫规则。
noanswerforward	String	无应答转移功能。 • on: 开启。 • off: 关闭。
ntransferto ntransferext ntransferprefix ntransfernum	String	无应答转移目的地，如果选择总是转移到分机，则需设置要转移到的分机号；如果总是转移到外线号码，则需设置要转移到的外线号码以及呼叫规则。
busyforward	String	忙时转移功能。 • on: 开启。 • off: 关闭。
btransferto btransferext btransferprefix btransfernum	String	忙时转移目的地，如果选择总是转移到分机，则需设置要转移到的分机号；如果总是转移到外线号码，则需设置要转移到的外线号码以及呼叫规则。
allowbeingmonitored	String	允许被监听。

名称	类型	描述
		• on: 开启。 • off: 关闭。
monitormode	String	监听模式。
ringtimeout	String	响铃超时。
maxduration	String	最大通话时长。
dnd	String	免打扰功能。 • on: 开启。 • off: 关闭。
callrestriction	String	外呼限制。 • on: 开启。 • off: 关闭。
agentid	String	报工号时要播报的号码。此参数默认为空，表示播报分机号。
inbound	Object	来电，呼入的外线通话。
inboundid	String	来电的编号，依据该参数对来电进行转接、查询、挂断等操作。
from	String	主叫号码。
to	String	被叫号码。
trunk	String	呼入时通过的中继名。
outbound	String	去电，呼出到外线的通话。
outboundid	Object	去电的编号，依据该参数对来电进行转接、查询、挂断等操作。
from	String	主叫号码。
to	String	被叫号码。
trunk	String	呼出时通过的中继名。
ext	Object	内部分机互拨的通话方。
extid	Int	和查询分机正在通话的分机号。
unselectoute	String	分机不可用的呼出路由。
selectoute	String	分机可用的呼出路由。

实际示例

请求示例

查询分机1000。

```

POST /api/v1.1.0/extension/query?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1 HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000"
}

```

查询所有分机。

```

POST /api/v1.1.0/extension/query?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1 HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150

```

响应示例

```

HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "extinfos": [
    {
      "extnumber": "1000",
      "username": "Jayson",
      "status": "Registered",
      "type": "SIP",
      "callerid": "1000",
      "registername": "1000",
      "registerpassword": "xMn5W4Gs",
      "maxregistrations": "1",
      "loginpassword": "5e12ba8dd1083a7f946e457cf3c18779",
      "email": "",
      "moblie": "",
      "language": "System Default",
      "hasvoicemail": "on",
      "enablevmtoemail": "off",
      "vmsecret": "1000",
      "alwaysforward": "off",
      "noanswerforward": "on",
      "ntransferto": "Voicemail",
      "ntransferprefix": "",
      "busyforward": "on",
      "btransferto": "Voicemail",
      "btransferprefix": ""
    }
  ]
}

```

```

        "ringsimultaneous": "off",
        "mobileprefix": "",
        "enablemobile": "off",
        "allowbeingmonitored": "off",
        "monitormode": "Disabled",
        "ringtimeout": "30",
        "maxduration": "Follow System",
        "dnd": "off",
        "callrestriction": "off",
        "agentid": ""
    }
}

```

编辑分机设置

更改分机的配置，如：分机号、分机名、同时注册数、邮箱、移动分机等。

Endpoint

POST /api/v1.1.0/extension/update?token={token}

请求参数

名称	是否必需	类型	描述
extid	是	Int	要修改配置的分机号。
extnumber	否	Int	修改后的分机号。
username	否	String	用户名。
callerid	否	String	分机来电显示名称。
registername	否	String	注册名称。
registerpassword	否	String	注册密码。
maxregistrations	否	Int	同时注册数。
loginpassword	否	String	用户密码。
email	否	String	邮件地址。
mobile	否	String	电话号码。
hasvoicemail	否	String	语音邮箱功能。 • on: 开启。 • off: 关闭。

名称	是否必需	类型	描述
vmsecret	否	Int	语音邮箱密码。
enablevmtoemail	否	String	是否发送语音邮件到邮箱。 • on: 开启。 • off: 关闭。
alwaysforward	否	String	总是转移。 • on: 开启。 • off: 关闭。
atransferto	否	String	总是转移目的地。 • Voicemail: 语音信箱。 • Extension: 分机。 • Mobile Number: 用户手机。 • Custom Number: 自定义号码。
atransferext	否	String	目的地为分机时的分机号。
atransferprefix	否	String	目的地为自定义号码或用户手机时的呼出前缀。  限制: 数字, 最大7位。
atransfernum	否	String	目的地为自定义号码或用户手机时的呼出号码。  限制: 数字, 最大15位, 转移目的地 Custom Number 不为空。
noanswerforward	否	String	无应答转移。 • on: 开启。 • off: 关闭。
ntransferto	否	String	无应答转移目的地。 • Voicemail: 语音信箱。 • Extension: 分机。 • Mobile Number: 用户手机。 • Custom Number: 自定义号码。
ntransferext	否	String	目的地为分机时的分机号。
ntransferprefix	否	String	目的地为自定义号码或用户手机时的呼出前缀。
ntransfernum	否	String	目的地为自定义号码或用户手机时的号码。

名称	是否必需	类型	描述
<code>busyforward</code>	否	String	忙时转移。 • on: 开启。 • off: 关闭。
<code>btransferto</code>	否	String	忙时转移目的地。 • Voicemail: 语音信箱。 • Extension: 分机。 • Mobile Number: 用户手机。 • Custom Number: 自定义号码。
<code>btransferext</code>	否	String	目的地为分机时的分机号。
<code>btransferprefix</code>	否	String	目的地为自定义号码或用户手机时的呼出前缀。  限制: 数字, 最大7位。
<code>btransfenum</code>	否	String	目的地为自定义号码或用户手机时的呼出号码。  限制: 数字, 最大15位, 转移目的地Custom Number不为空。
<code>allowbeingmonitored</code>	否	String	允许被监听。 • on: 开启。 • off: 关闭。
<code>monitormode</code>	否	String	监听模式。 • Disabled: 禁用。 • Extensive: 通用模式。 • Listen: 普通监听。 • Whisper: 密语监听。 • Barge-in: 强插监听。
<code>ringtimeout</code>	否	String	响铃超时。 • 15 • 30 • 60 • 120 • 300
<code>maxduration</code>	否	String	最大通话时长。

名称	是否必需	类型	描述
			• Follow System：系统默认值。 • Unlimited：无限制。 • 60 • 300 • 600 • 900 • 1800 • 3600 • 6000
dnd	否	String	免打扰功能。 • on：开启。 • off：关闭。
callrestriction	否	String	外呼限制。 • on：开启。 • off：关闭。
agentid	否	String	报工号时要播报的号码。此参数默认为空，表示播报分机号。
selectoutroute	否	String	选择分机的呼出路由。从 unselectoutroute 中选择一个呼出路由。selectoutroute 最多支持4096个字符。

实际示例

请求示例

将分机1002的用户名修改为Amy，其他信息不变。

```
POST /api/v1.1.0/extension/update?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1002",
  "username": "Amy"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
```

```
    "status": "Success"
}
```

中继

查询中继列表

查询所有中继的基本信息，如：中继名、中继状态、中继类型等。

Endpoint

```
POST /api/v1.1.0/trunklist/query?token={token}
```

请求参数

无参数，直接发送查询中继列表请求即可。

响应参数

名称	类型	描述
trunklist	Object	对象参数。
trunkname	Int	中继名称。
type	String	中继类型 • SIP
status	String	中继当前状态。 • Registering：注册中 • Failure：注册失败 • Registered：已注册 • Disable：禁用 • Unknown：未知

实际示例

请求示例

```
POST /api/v1.1.0/trunklist/query?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
```

Host: 192.168.5.150

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "trunklist": [
    {
      "trunkname": "voip123",
      "status": "failure",
      "type": "SIP"
    },
    {
      "trunkname": "siptrunk8",
      "status": "failure",
      "type": "SIP"
    }
  ]
}
```

查询 SIP 中继

查询单个 SIP 中继、多个 SIP 中继 或所有 SIP 中继的基本设置。

Endpoint

POST /api/v1.1.0/siptrunk/query?token={token}

请求参数

名称	是否必需	类型	描述
trunkname	是	String	SIP 中继的名称。 • 查询单个 SIP 中继: 设置 trunkname 为 SIP 中继的名称。 例如: "trunkname": "SIP" • 查询多个 SIP 中继: 设置 trunkname 为 SIP 中继的名称, 多个 SIP 中继之间用 , 隔开。 例如: "trunkname": "SIP1, SIP2" • 查询所有 SIP 中继: 设置 trunkname 为 all。

响应参数

名称	类型	描述
trunks	Object	对象参数。
id	Int	中继的标识码。
trunkname	String	中继名称。
trunktype	String	中继类型。 • register: 注册中继。 • peer: 点对点中继。 • account: 账号中继。
注册中继		
host	String	SIP 运营商的域名或 IP 地址。
port	String	中继的 SIP 端口。
domain	String	SIP 运营商的域名或 IP 地址。
username	String	SIP 运营商提供的 SIP 账号。
authname	String	SIP 运营商提供的 SIP 账号。
fromuser	String	From 头域。
		 注： 如果 SIP 运营商不支持，则放空。
password	String	SIP 账号的密码。
点对点中继		
host	String	SIP 运营商的域名或 IP 地址。
port	String	中继的 SIP 端口。
domain	String	SIP 运营商的域名或 IP 地址。
账号中继		
username	String	账号中继的用户名。
authname	String	账号中继的认证名。
password	String	账号中继的密码。
DOD 号码		
extensionsdod	String	分机绑定的 DOD 号码。 显示格式为: <code>#dod_number#-#dod_name#-{extension_number}</code> 例如: "extensionsdod": "5503301-yeastar-1000"

名称	类型	描述
extensionsgroupdod	String	分机组绑定的 DOD 号码。 显示格式 为: <i>#dod_number#-#dod_name#-#extension_group_name#</i> 例如: "extensionsdod":"5503301-yeastar-sales"

实际示例

请求示例

```
POST /api/v1.1.0/siptrunk/query?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "trunkname": "SIP"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "trunks": [
    {
      "id": "8",
      "trunkname": "SIP",
      "trunktype": "register",
      "host": "192.168.11.158",
      "port": "5060",
      "domain": "192.168.11.158",
      "username": "8001",
      "authname": "8001",
      "fromuser": "8001",
      "password": "E7P0?LywXnXa_B-n",
      "extensionsdod": "",
      "extensionsgroupdod": ""
    }
  ]
}
```

添加 SIP 中继

添加并设置 SIP 中继。

Endpoint

```
POST /api/v1.1.0/siptrunk/add?token={token}
```

请求参数

名称	是否必需	类型	描述
trunkname	是	String	中继名称。
trunktype	是	String	中继类型。 • register：注册中继。 • peer：点对点中继。 • account：账号中继。
注册中继			
host	是	String	SIP 运营商的域名或 IP 地址。
port	是	String	中继的 SIP 端口。
domain	是	String	SIP 运营商的域名或 IP 地址。
username	是	String	SIP 运营商提供的 SIP 账号。
authname	是	String	SIP 运营商提供的 SIP 账号。
fromuser	否	String	From 头域。  注： 如果 SIP 运营商不支持，则放空。
password	是	String	SIP 账号的密码。
点对点中继			
host	是	String	SIP 运营商的域名或 IP 地址。
port	是	String	中继的 SIP 端口。
domain	是	String	SIP 运营商的域名或 IP 地址。
账号中继			
username	是	String	账号中继的用户名。
authname	是	String	账号中继的认证名。
password	是	String	账号中继的密码。

名称	是否必需	类型	描述
DOD 号码			
extensionsdod	否	String	分机绑定的 DOD 号码。 - 格式: <code>#dod_number#-#dod_name#-{extension_number}</code> - 例 如: <code>"extensionsdod": "5503301-yeastar-1000"</code> - 多个 DOD 号码之间用 <code>,</code> 隔开。 - 例 如: <code>"extensionsdod": "5503301-yeastar-1000, 5503302-yeastar-1001"</code>
extensionsgroupdod	否	String	分机组绑定的 DOD 号码。 - 格式: <code>#dod_number#-#dod_name#-{extension_group_name}</code> - 例 如: <code>"extensionsdod": "5503301-yeastar-sales"</code> - 多个 DOD 号码之间用 <code>,</code> 隔开。 - 例 如: <code>"extensionsdod": "5503301-yeastar-sales, 5503302-yeastar-support"</code>

实际示例

请求示例

```
POST /api/v1.1.0/siptrunk/add?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "trunkname": "siptrunk1",
  "trunktype": "register",
  "host": "192.168.12.161",
  "port": "5060",
  "domain": "192.168.12.161",
  "username": "1001",
  "authname": "1001",
```

```
    "fromuser": "",
    "password": "Pajsjx9183",
    "extensionsdod": "",
    "extensionsgroupdod": ""
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
    "status": "Success"
}
```

编辑 SIP 中继

更改 SIP 中继的基本设置和 DOD 号码设置。

Endpoint

```
POST /api/v1.1.0/siptrunk/update?token={token}
```

请求参数

名称	是否必需	类型	描述
id	是	Int	中继的标识码。
trunkname	是	String	中继名称。
trunktype	是	String	中继类型。 • register：注册中继。 • peer：点对点中继。 • account：账号中继。
注册中继			
host	是	String	SIP 运营商的域名或 IP 地址。
port	是	String	中继的 SIP 端口。
domain	是	String	SIP 运营商的域名或 IP 地址。
username	是	String	SIP 运营商提供的 SIP 账号。
authname	是	String	SIP 运营商提供的 SIP 账号。
fromuser	否	String	From 头域。

名称	是否必需	类型	描述
			 注： 如果 SIP 运营商不支持，则放空。
password	是	String	SIP 账号的密码。
点对点中继			
host	是	String	SIP 运营商的域名或 IP 地址。
port	是	String	中继的 SIP 端口。
domain	是	String	SIP 运营商的域名或 IP 地址。
账号中继			
username	是	String	账号中继的用户名。
authname	是	String	账号中继的认证名。
password	是	String	账号中继的密码。
DOD 号码			
extensionsdod	否	String	分机绑定的 DOD 号码。 • 格式： 式： <code>#dod_number#-#dod_name#-{extension_number}</code> 例 如： <code>"extensionsdod": "5503301-yeastar-1000"</code> • 多个 DOD 号码之间用 <code>,</code> 隔开。 例 如： <code>"extensionsdod": "5503301-yeastar-1000, 5503302-yeastar-1001"</code>
extensionsgroupdod	否	String	分机组绑定的 DOD 号码。 • 格式： 式： <code>#dod_number#-#dod_name#-{extension_group_name}</code> 例 如： <code>"extensionsdod": "5503301-yeastar-sales"</code> • 多个 DOD 号码之间用 <code>,</code> 隔开。 例 如： <code>"extensionsdod": "5503301-yeastar-sales, 5503302-yeastar-support"</code>

实际示例

请求示例

```
POST /api/v1.1.0/siptrunk/update?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "id": "8",
  "trunkname": "siptrunk1",
  "trunktype": "register",
  "host": "192.168.12.161",
  "port": "5060",
  "domain": "192.168.12.161",
  "username": "1001",
  "authname": "1001",
  "fromuser": "",
  "password": "Pajsjx9183",
  "extensionsdod": "",
  "extensionsgroupdod": ""
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

删除 SIP 中继

删除单个 SIP 中继、多个 SIP 中继或所有 SIP 中继。

Endpoint

```
POST /api/v1.1.0/siptrunk/delete?token=#token#
```

请求参数

名称	是否必需	类型	描述
trunkname	是	String	SIP 中继的名称。

名称	是否必需	类型	描述
			• 删除单个 SIP 中继: 设置 trunkname 为 SIP 中继名称。 例如: "trunkname": "SIP" • 删除多个 SIP 中继: 设置 trunkname 为 SIP 中继名称, 多个 SIP 中继之间用 , 隔开。 例如: "trunkname": "SIP,siptrunk" • 删除所有 SIP 中继: 设置 trunkname 为 all。

实际示例

请求示例

```
POST /api/v1.1.0/siptrunk/delete?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "trunkname": "Test,Test1"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

呼入路由

查询呼入路由

查询单个呼入路由、多个呼入路由或所有呼入路由的设置。

Endpoint

```
POST /api/v1.1.0/inroute/query?token={token}
```

请求参数

名称	是否必需	类型	描述
name	是	String	呼入路由的名称。 • 查询单个呼入路由: 设置 name 为呼入路由的名称。 例如: "name": "route1" • 查询多个呼入路由: 设置 name 为呼入路由的名称, 多个呼入路由之间用 , 隔开。 例如: "name": "route1, route2" • 查询所有呼入路由: 设置 name 为 all。

响应参数

名称	类型	描述
inroutes	Object	对象参数。
id	Int	呼入路由的标识码。
name	String	呼入路由的名称。
did	String	DID 匹配模式。
caller_id	String	来电匹配模式。
trunks	String	中继。
desttype	String	目的地类型。 • hangup: 挂断 • extension: 分机 • extrange: 分机范围 • voicemail: 语音邮箱 • ivr: IVR • ringgroup: 响铃组 • queue: 队列 • conference: 会议室 • disa: DISA • callback: 回拨 • outroute: 呼出路由 • faxtoemail: 传真到邮箱
dest	String	具体目的地。

实际示例

请求示例

```
POST /api/v1.1.0/inroute/query?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "name": "Routein"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "inroutes": [
    {
      "id": "4",
      "name": "Routein",
      "did": "",
      "caller_id": "",
      "trunks": "siptrunk",
      "desttype": "ivr",
      "dest": "6500"
    }
  ]
}
```

添加呼入路由

添加并设置呼入路由。

Endpoint

```
POST /api/v1.1.0/inroute/add?token={token}
```

请求参数

名称	是否必需	类型	描述
name	是	String	呼入路由的名称。

名称	是否必需	类型	描述
did	否	String	DID 匹配模式。
caller_id	否	String	来电匹配模式。
trunks	是	String	中继名称。 多个中继之间用, 隔开, 例如: "trunks": "trunk1, trunk2"
desttype	是	String	目的地类型。 • hangup: 挂断 • extension: 分机 • exrange: 分机范围 • voicemail: 语音邮箱 • ivr: IVR • ringgroup: 响铃组 • queue: 队列 • conference: 会议室 • disa: DISA • callback: 回拨 • outroute: 呼出路由 • faxtoemail: 传真到邮箱
dest	是	String	具体目的地。

实际示例

请求示例

```
POST /api/v1.1.0/inroute/add?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150

{
  "name": "sipin",
  "did": "",
  "caller_id": "",
  "trunks": "siptrunk",
  "desttype": "extension",
  "dest": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
```

```
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

编辑呼入路由

更改呼入路由的设置。

Endpoint

```
POST /api/v1.1.0/inroute/update?token={token}
```

请求参数

名称	是否必需	类型	描述
name	是	String	呼入路由的名称。
did	否	String	DID 匹配模式。
caller_id	否	String	来电匹配模式。
trunks	是	String	中继名称。 多个中继之间用, 隔开, 例如: "trunks":"trunk1,trunk2"
desttype	是	String	目的地类型。 • hangup: 挂断 • extension: 分机 • extrange: 分机范围 • voicemail: 语音邮箱 • ivr: IVR • ringgroup: 响铃组 • queue: 队列 • conference: 会议室 • disa: DISA • callback: 回拨 • outroute: 呼出路由 • faxtoemail: 传真到邮箱
dest	是	String	具体目的地。

实际示例

请求示例

```

POST /api/v1.1.0/inroute/update?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "id": "1",
  "name": "Routein",
  "did": "10.",
  "caller_id": "",
  "trunks": "siptrunk",
  "desttype": "ivr",
  "dest": "6500"
}

```

响应示例

```

HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}

```

删除呼入路由

删除单个呼入路由、多个呼入路由或所有呼入路由。

Endpoint

```
POST /api/v1.1.0/inroute/delete?token={token}
```

请求参数

名称	是否必需	类型	描述
name	是	String	呼入路由的名称。 • 删除单个呼入路由：设置 name 为呼入路由的名称。 例如： "name": "route1" • 删除多个呼入路由：设置 name 为呼入路由的名称，多个呼入路由之间用 , 隔开。 例如： "name": "route1,route2"

名称	是否必需	类型	描述
			• 删除所有呼入路由：设置 name 为 all。

实际示例

请求示例

```
POST /api/v1.1.0/inroute/delete?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "name": "route1,route2"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

呼出路由

查询呼出路由

查询单个呼出路由、多个呼出路由或所有呼出路由的设置。

Endpoint

```
POST /api/v1.1.0/outroute/query?token={token}
```

请求参数

名称	是否必需	类型	描述
name	是	String	呼出路由的名称。

名称	是否必需	类型	描述
			• 查询单个呼出路由: 设置 name 为呼出路由的名称。 例如: "name": "route1" • 查询多个呼出路由: 设置 name 为呼出路由的名称, 多个呼出路由之间用 , 隔开。 例如: "name": "route1,route2" • 查询所有呼出路由: 设置 name 为all。

响应参数

名称	类型	描述
outroutes	Object	对象参数。
id	Int	呼出路由的标识码。
name	String	呼出路由的名称。
prefix	String	呼出模式。
strip	String	删除前缀位数。
prepend	String	前置号码。
trunks	String	中继名称。
extensions	String	分机号码。
extensionsgroup	String	分机组。

实际示例

请求示例

```
POST /api/v1.1.0/outroute/query?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "name": "route1"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
```

```
{
  "status": "Success",
  "outroutes": [
    {
      "id": "5",
      "name": "route1",
      "prefix": "6.",
      "strip": "1",
      "prepend": "",
      "trunks": "SIP1",
      "extensions": "1000,1001,1002",
      "extensionsgroup": ""
    }
  ]
}
```

添加呼出路由

添加并设置呼出路由。

Endpoint

```
POST /api/v1.1.0/outroute/add?token={token}
```

请求参数

名称	是否必需	类型	描述
name	是	String	呼出路由的名称。
prefix	是	String	呼出模式。
strip	否	String	删除前缀位数。
prepend	否	String	前置号码。
trunks	是	String	中继名称。 多个中继之间用 , 隔开，例如 "trunks": "trunk1, trunk2"
extensions	是	String	允许使用该呼出路由的分机。 多个分机之间用 , 隔开，例如 "extensions": "1001, 1002"
extensionsgroup	否	String	允许使用该呼出路由的分机组。 多个分机组之间用 , 隔开，例如 "extensionsgroup": "sales, support"

实际示例

请求示例

```
POST /api/v1.1.0/outroute/add?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "name": "outroute1",
  "prefix": "0",
  "strip": "1",
  "prepend": "",
  "trunks": "siptrunk",
  "extensions": "1001",
  "extensionsgroup": "sales,support"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

编辑呼出路由

编辑呼出路由的设置。

Endpoint

```
POST /api/v1.1.0/outroute/update?token={token}
```

请求参数

名称	是否必需	类型	描述
name	是	String	呼出路由的名称。
prefix	是	String	呼出模式。
strip	否	String	删除前缀位数。
prepend	否	String	前置号码。

名称	是否必需	类型	描述
trunks	是	String	中继名称。 多个中继之间用, 隔开, 例如 "trunks":"trunk1,trunk2"
extensions	是	String	允许使用该呼出路由的分机。 多个分机之间用, 隔开, 例如 "extensions":"1001,1002"
extensionsgroup	否	String	允许使用该呼出路由的分机组。 多个分机组之间用, 隔开, 例如 "extensionsgroup":"sales,support"

实际示例

请求示例

```
POST /api/v1.1.0/outroute/update?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "id": "8",
  "name": "outroute2",
  "prefix": "0",
  "strip": "1",
  "prepend": "",
  "trunks": "siptrunk",
  "extensions": "1003",
  "extensionsgroup": ""
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

删除呼出路由

删除单个呼出路由、多个呼出路由或所有呼出路由。

Endpoint

```
POST /api/v1.1.0/outroute/delete?token={token}
```

请求参数

名称	是否必需	类型	描述
name	是	String	呼出路由的名称。 • 删除单个呼出路由：设置 name 为呼出路由的名称。 例如： "name": "route1" • 删除多个呼出路由：设置 name 为呼出路由的名称，多个呼出路由之间用 , 隔开。 例如： "name": "route1,route2" • 删除所有呼出路由：设置 name 为 all。

实际示例

请求示例

```
POST /api/v1.1.0/outroute/delete?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "name": "route1,route2"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

IVR

查询 IVR 设置

查询单个或多个 IVR 的详细信息，如：IVR 号码、名称、按键事件、响应超时时间。

Endpoint

POST /api/v1.1.0/ivr/query?token={token}

请求参数

名称	是否必需	类型	描述
ivrid	否	String	IVR 号码。 • 查询单个 IVR ：设置 ivrid 为 IVR 号码。 例如： "ivrid":"6500" • 查询多个 IVR ：设置 ivrid 为 IVR 号码，多个 IVR 之间用 , 隔开。 例如： "ivrid":"6500,6501,6502" • 查询所有 IVR ：不带 ivrid 请求参数。

响应参数

名称	类型	描述
ivrnumber	String	IVR 号码。
ivrname	String	IVR 名称。
prompt	String	IVR 提示音。
promptrepeat	String	提示音播放次数。
responsetimeout	Int	响应超时时间。
digittimeout	Int	按键超时时间。
dialext	String	允许从分机呼出。 • on：允许。 • off：不允许。
dialoutboundroutes	String	允许从呼出路由拨出。 • on：允许。 • off：不允许。
selectedrouters	String	所选择的路由。
dialtocheckvoicemail	String	允许查阅语音留言。

名称	类型	描述
		• on: 允许。 • off: 不允许。

实际示例

请求示例

查询 IVR 6500。

```
POST /api/v1.1.0/ivr/query?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "ivrid": "6500"
}
```

查询所有IVR。

```
POST /api/v1.1.0/ivr/query?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "ivr": [
    {
      "ivrnumber": "6500",
      "ivrname": "6500",
      "prompt": "[Default]",
      "promptrepeat": "3",
      "responsetimeout": "3",
      "digittimeout": "3",
      "dialext": "on",
      "dialoutboundroutes": "off",
      "dialtocheckvoicemail": "off"
    }
  ]
}
```

编辑 IVR 设置

修改 IVR 的配置，如：修改 IVR 号码、名称等。

Endpoint

POST /api/v1.1.0/ivr/update?token={token}

请求参数

名称	是否必需	类型	描述
ivr <code>id</code>	是	String	要配置的 IVR 唯一标识（IVR 号码）。
ivr <code>number</code>	否	String	修改后的 IVR 号码。
ivr <code>name</code>	否	String	IVR 名称。 限制： • 不允许输入！\$（）/ # ； ， [] " = < > & ' ` ^ % @ { } 和空格。 • 最大31位。 • 不能为空。
prompt <code>repeat</code>	否	String	提示音播放次数 可选择的值：1、2、3、4、5
respon <code>settimeout</code>	否	Int	响应超时时间 可选择的值：1、2、3、4、5、6、7、8、9、10
digitt <code>timeout</code>	否	Int	按键超时时间 可选择的值：1、2、3、4、5、6、7、8、9、10
diale <code>xt</code>	否	String	允许从分机呼出 • on：允许。 • off：不允许。
dialto <code>checkvoicemail</code>	否	String	允许查阅语音留言 • on：允许。 • off：不允许。

实际示例

请求示例

修改 IVR 6500 的 IVR 号码为6501。

```
POST /api/v1.1.0/ivr/update?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
    "ivrid": "6500",
    "ivrnumber": "6501"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
    "status": "Success"
}
```

队列

查询队列的设置

查询单个或多个队列的详细信息，如：队列号码、队列名称、静态坐席、动态坐席等。

Endpoint

```
POST /api/v1.1.0/queue/query?token={token}
```

请求参数

名称	是否必需	类型	描述
queueid	否	String	队列号码。 • 查询单个队列 : 设置 queueid 为 IVR 号码。 例如: "queueid": "6700" • 查询多个队列 : 设置 queueid 为 IVR 号码, 多个 IVR 之间用 , 隔开。 例如: "queueid": "6700,6701,6702" • 查询所有队列 : 不带 queueid 请求参数。

响应参数

名称	类型	描述
queues	Object	对象参数。
queuenumber	Int	队列号码。
queuename	String	队列号码。
password	String	加入动态坐席的密码。
ringstrategy	String	响铃策略。 • Ring All • Least Recent • Fewest Calls • Random • Rrmemory • Linear
failoveraction	String	呼入失败目的地。
agents	String	固定坐席。
dynamicagents	String	动态坐席。
agenttimeout	String	坐席响铃时间。
agentannounce	String	坐席应答提示音。
wrapuptime	Int	休息时间。
ringinuse	String	使用中振铃。 • on: 启用。 • off: 关闭。
retry	Int	重试间隔时间。
musiconhold	String	等待音乐。
maxwaittime	Int	最大等待时间。
joinempty	String	无座席时允许呼入。 • on: 启用。 • off: 关闭。
leavewhenempty	String	无座席时结束等待。 • on: 启用。 • off: 关闭。
joinannounce	String	进入队列提示音。

名称	类型	描述
announcepos	String	公告当前位置。 • on: 启用。 • off: 关闭。
announcfreq	String	用户公告频率。
announceltdtime	Int	公告等待时间。
userannounce	String	系统公告提示音。
userannouncfreq	Int	系统公告频率。
breakoutkey	String	按键 DTMF。
breakoutaction	String	按键目标。
breakoutdest	String	按键目标的最终目的地。
idannouncement	String	报工号提示音文件名。没有设置则默认不播报。此字段默认为 none。  注： 查询队列时此参数为必须返回参数。
satisfactionsurvey	String	满意度调查需要播放的提示音文件名。没有设置则默认不播报。此字段默认为 none。

实际示例

请求示例

查询队列6701。

```
POST /api/v1.1.0/queue/query?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "queueid": "6701"
}
```

查询所有队列。

```
POST /api/v1.1.0/queue/query?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "queues": [
    {
      "queuenumber": "6700",
      "password": "6700123",
      "queue name": "6700",
      "ringstrategy": "Ring All",
      "failoveraction": "Hang up",
      "agents": "103,102,",
      "dynamicagents": "Local/4001@only-dialextension-q6700",
      "agenttimeout": "10", "agentannounce": "[None]",
      "wrapuptime": "10",
      "ringinuse": "on",
      "retry": "10",
      "musiconhold": "[None]",
      "maxwaittime": "30",
      "joinempty": "on",
      "leavewhenempty": "on",
      "joinannounce": "[None]",
      "announcepos": "off",
      "announcfreq": "15",
      "announceholdtime": "on",
      "userannounce": "[None]",
      "userannouncfreq": "60",
      "breakoutkey": "0",
      "breakoutaction": "Hang up",
      "satisfactionsurvey": "None",
      "idannouncement": "Default"
    }
  ]
}
```

编辑队列设置

修改队列的配置，如：修改队列号码、名称等。

Endpoint

POST /api/v1.1.0/queue/update?token={token}

请求参数

名称	是否必需	类型	描述
queueid	是	Int	要修改的队列号码。
queuenumber	否	Int	修改后的队列号码。
queuename	否	String	队列名称。
password	否	String	加入动态坐席的密码。
ringstrategy	否	String	响铃策略。 • Ring All: 全部响铃 • Least Recent: 最近最少被叫响铃 • Fewest Calls: 最少接通响铃 • Random: 随机响铃 • Rrmemory: 顺序响铃 • Linear: 线性响铃
failoveraction	否	String	呼入失败目的地。 • Hangup: 挂机 • Extension: 分机 • Voicemail: 语音信箱 • IVR: IVR (自动话务员) • Ring Group: 响铃组 • Queue: 队列 • Conference: 会议室 • Fax to Email: 传真到邮件 • Dial by name: 通过名字呼叫
failoverdest	否	String	呼入失败目的地。
agents	否	String	固定座席。
dynamicagents	否	String	动态坐席。
agenttimeout	否	String	坐席响铃时间 可选值: 10、20、30、40、50
wrapuptime	否	Int	休息时间 可选值: 10、20、30、40、50
ringinuse	否	String	使用中振铃

名称	是否必需	类型	描述
			• on: 开启。 • off: 关闭。
retry	否	Int	重试间隔时间 可选值: 10、20、30、40、50
maxwaittime	否	Int	最大等待时间 可选值: 300、600、900、1200、1800
joinempty	否	String	无座席时允许呼入 • on: 开启。 • off: 关闭。
leavewhenempty	否	String	无座席时结束等待 • on: 开启。 • off: 关闭。
announcepos	否	String	公告当前位置 • on: 开启。 • off: 关闭。
announcerefq	否	String	用户公告频率 可选 值: 0、15、30、45、60、120、180、240、300、600、1200
announceholdtime	否	Int	公告等待时间 • on: 开启。 • off: 关闭。
userannouncerefq	否	Int	系统公告频率 可选 值: 0、15、30、45、60、120、180、240、300、600、1200
breakoutkey	否	String	按键 DTMF 可选值: None、0、1、2、3、4、5、6、7、8、9、*、#
breakoutaction	否	String	按键目标 • Hang up: 挂机 • Extension: 分机 • Voicemail: 语音信箱 • IVR: IVR (自动话务员)

名称	是否必需	类型	描述
			• Ring Group: 响铃组 • Queue: 队列 • Conference: 会议室 • Fax to Email: 传真到邮件 • Dial by Name: 通过名字呼叫
breakoutdest	否	String	按键目标的最终目的地。
idannouncement	否	String	报工号提示音文件名。没有设置则默认不播报。此字段默认为 none。
satisfactionsurvey	否	String	满意度调查提示音。没有设置则默认不播报。此参数默认为 None。

实际示例

请求示例

修改队列6701的号码为6702。

```
POST /api/v1.1.0/queue/update?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "queueid": "6701",
  "queuenumber": "6702"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

查询队列状态

查询队列坐席的状态。

Endpoint

```
POST /api/v1.1.0/queuestatus/query?token={token}
```

请求参数

名称	是否必需	类型	描述
queueid	是	Int	队列号码。

响应参数

名称	类型	描述
queues	Object	队列。
queuenumber	Int	队列号码。
queuestatus	Object	队列状态。
callercount	Int	当前队列等候人数。
members	Object	队列成员。
agent	Int	坐席号码。
agentstatus	Int	坐席状态。 • 0-Unknown：未知 • 1-Not In Use：坐席处于空闲状态 • 2-In Use：坐席处于通话中 • 3-Busy：未知 • 4-Invalid：未知 • 5-Unavailable：坐席分机未注册上 • 6-Ringing：坐席分机正在响铃 • 7-In Use Ringing：坐席通话中收到新来电 • 8-On Hold：坐席保持当前通话 • 9-Paused：坐席被暂停服务，不接收来电业务 • 10-Warp-up time：坐席处于休息状态，不接收来电业务

实际示例

请求示例

```
POST /api/v1.1.0/queuestatus/query?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
    "queueid": "6700"
}
```

响应示例

```

HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "queues": [
    {
      "queuenumber": "6700",
      "queuestatus": [
        {
          "callercount": "0",
          "members": [
            {
              "agent": "8034",
              "agentstatus": "1"
            },
            {
              "agent": "8033",
              "agentstatus": "1"
            },
            {
              "agent": "5007",
              "agentstatus": "1"
            },
            {
              "agent": "1004",
              "agentstatus": "9"
            },
            {
              "agent": "6000",
              "agentstatus": "5"
            },
            {
              "agent": "6036",
              "agentstatus": "5"
            },
            {
              "agent": "6001",
              "agentstatus": "5"
            }
          ]
        }
      ]
    }
  ]
}

```

```
    ]
}
```

签入队列

通过此接口，可以将指定动态坐席签入指定的队列。

Endpoint

```
POST /api/v1.1.0/queue/add_dynamicagent?token={token}
```

请求参数

名称	是否必需	类型	描述
queueid	是	Int	请求时的队列号码。
extid	是	Int	动态坐席的号码。
password	否	Int	队列的密码。

实际示例

请求示例

将坐席1000签入队列6700，队列的密码为6700123。

```
POST /api/v1.1.0/
queue/add_dynamicagent?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "queueid": "6700",
  "extid": "1000",
  "password": "6700123"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

签出队列

通过此接口，可以将指定动态坐席签出指定队列。

Endpoint

```
POST /api/v1.1.0/queue/del_dynamicagent?token={token}
```

请求参数

名称	是否必需	类型	描述
queueid	是	Int	请求时的队列号码。
extid	是	Int	动态坐席的号码。
password	是	Int	队列的密码。

实际示例

请求示例

请求将坐席1000 签出队列6700；队列的密码为6700123。

```
POST /api/v1.1.0/
queue/del_dynamicagent?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "queueid": "6700",
  "extid": "1000",
  "password": "6700123"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

暂停坐席服务

暂停某个坐席的服务。坐席被暂停之后，将不会接听到呼入到该队列的来电。

Endpoint

```
POST /api/v1.1.0/queue/pause_agent?token={token}
```

请求参数

名称	是否必需	类型	描述
queueid	是	Int	请求时的队列号码。
extid	是	Int	坐席的号码。

实际示例

请求示例

```
POST /api/v1.1.0/queue/pause_agent?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "queueid": "6700",
  "extid": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

恢复坐席服务

恢复某个坐席的服务。坐席被恢复之后，可以接听到呼入到该队列的来电。

Endpoint

```
POST /api/v1.1.0/queue/unpause_agent?token={token}
```

请求参数

名称	是否必需	类型	描述
queueid	是	Int	队列号码。
extid	是	Int	坐席号码。

实际示例

请求示例

请求恢复坐席1000的服务。

```
POST /api/v1.1.0/queue/unpause_agent?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "queueid": "6700",
  "extid": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

广播组

查询广播组列表

查询已创建的广播组。

Endpoint

```
POST /api/v1.1.0/paginggroupplist/query?token={token}
```

请求参数

无参数，直接发送查询广播组请求即可。

响应参数

名称	类型	描述
paginggrouplist	Object	对象参数。
id	Int	广播组的标识码。
number	String	广播组号码。
name	String	广播组名称。

实际示例

请求示例

```
POST /api/v1.1.0/  
paginggrouplist/query?token=277ac400357b509b4a587ff2157f7ad5 HTTP/1.1  
Content-Type:application/json; charset=utf-8  
Host: 192.168.5.150
```

响应示例

```
HTTP/1.1 200 OK  
Access-control-allow-origin: *  
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  
{  
  "status": "Success",  
  "paginggrouplist": [  
    {  
      "id": "1",  
      "number": "6300",  
      "name": "6300"  
    }  
  ]  
}
```

查询广播组

查询单个广播组、多个广播组或所有广播组的设置参数。

Endpoint

POST /api/v1.1.0/paginggroup/query?token={token}

请求参数

名称	是否必需	类型	描述
number	是	String	广播组号码。 • 查询单个广播组：设置 number 为广播组号码。 例如: "number": "6300" • 查询多个广播组：设置 number 为广播组号码，多个广播组号码之间用 , 隔开。 例如: "number": "6300,6301" • 查询所有广播组：设置 number 为 all。

响应参数

名称	类型	描述
paginggroup	Object	对象参数。
id	Int	广播组标识码。
number	String	广播组号码。
name	String	广播组名称。
duplex	String	广播组类型。 • paging: 单向传呼。广播组内的分机只能听到发起者的声音。 • intercom: 双向对讲。广播组内的所有电话分机都可以进行通话并且都能听见彼此的声音。 • multicast: 单向组播。向指定的组播地址发送语音，监听指定组播地址的话机均可以听到该广播。
allowexten	String	广播组成员。如选择分机为广播组成员，则显示分机号码。
allowextengroup	String	广播组成员。如选择分机组为广播组成员，则显示分机组名称。
enablekeyhanup	String	是否开启 按*应答 功能。启用该功能，分机可以按*应答广播。应答后，该分机与发起者进行通话，其他分机的广播终止。 • yes: 开启。 • no: 关闭。
multicastip	String	如果广播组类型为 multicast，则显示该广播组的组播地址。

实际示例

请求示例

```
POST /api/v1.1.0/paginggroup/query?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "number": "6300"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "paginggroup": [
    {
      "id": "1",
      "number": "6300",
      "name": "6300",
      "duplex": "paging",
      "allowexten": "1001",
      "allowextengroup": "",
      "enablekeyhanup": "no",
      "multicastip": ""
    }
  ]
}
```

添加广播组

添加并设置广播组。

Endpoint

```
POST /api/v1.1.0/paginggroup/add?token={token}
```

请求参数

名称	是否必需	类型	描述
number	是	String	广播组号码。
name	是	String	广播组名称。
duplex	是	String	广播组类型。 • paging: 单向传呼。广播组内的分机只能听到发起者的声音。 • intercom: 双向对讲。广播组内的所有电话分机都可以进行通话并且都能听见彼此的声音。 • multicast: 单向组播。向指定的组播地址发送语音, 监听指定组播地址的话机均可以听到该广播。
单向传呼 ("duplex": "paging")			
allowexten	是	String	广播组成员。选择分机为广播组成员。多个分机用 , 隔开。例如: "allowexten": "1000,1001,1002"。
allowextengroup	是	String	广播组成员。选择分机组为广播组成员。多个分机组用 , 隔开。例如: "allowextengroup": "sales,support"。
enablekeyhanup	是	String	是否开启 按*应答 功能。启用该功能, 分机可以按*应答广播。应答后, 该分机与发起者进行通话, 其他分机的广播终止。 • yes: 开启。 • no: 关闭。
双向对讲 ("duplex": "intercom")			
allowextengroup	是	String	广播组成员。选择分机组为广播组成员。多个分机组用 , 隔开。例如: "allowextengroup": "sales,support"。
enablekeyhanup	是	String	是否开启 按*应答 功能。启用该功能, 分机可以按*应答广播。应答后, 该分机与发起者进行通话, 其他分机的广播终止。 • yes: 开启。 • no: 关闭。
allowexten	是	String	广播组成员。选择分机为广播组成员。多个分机用 , 隔开。例如: "allowexten": "1000,1001,1002"。
单向组播 ("duplex": "multicast")			
multicastip	是	String	如果广播组类型为 multicast , 填写发送广播的组播地址。

名称	是否必需	类型	描述
enablekeyhanup	是	String	是否开启 按*应答 功能。启用该功能，分机可以按*应答广播。应答后，该分机与发起者进行通话，其他分机的广播终止。 • yes: 开启。 • no: 关闭。

实际示例

请求示例

```
POST /api/v1.1.0/paginggroup/add?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "number": "6300",
  "name": "6300",
  "duplex": "paging",
  "allowexten": "1000,1001",
  "allowextengroup": "",
  "enablekeyhanup": "no",
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

编辑广播组

更改广播组的设置。

Endpoint

```
POST /api/v1.1.0/paginggroup/update?token={token}
```

请求参数

名称	是否必需	类型	描述
id	是	String	广播组的标识码。通过 查询广播组列表 可以查询广播组的标识码。
number	是	String	广播组号码。
name	是	String	广播组名称。
duplex	是	String	广播组类型。 • paging: 单向传呼。广播组内的分机只能听到发起者的声音。 • intercom: 双向对讲。广播组内的所有电话分机都可以进行通话并且都能听见彼此的声音。 • multicast: 单向组播。向指定的组播地址发送语音，监听指定组播地址的话机均可以听到该广播。
单向传呼 ("duplex": "paging")			
allowexten	是	String	广播组成员。选择分机为广播组成员。多个分机用, 隔开。例如: "allowexten": "1000,1001,1002"。
allowextengroup	是	String	广播组成员。选择分机组为广播组成员。多个分机组用, 隔开。例如: "allowextengroup": "sales,support"。
enablekeyhanup	是	String	是否开启 按*应答 功能。启用该功能，分机可以按*应答广播。应答后，该分机与发起者进行通话，其他分机的广播终止。 • yes: 开启。 • no: 关闭。
双向对讲 ("duplex": "intercom")			
allowextengroup	是	String	广播组成员。选择分机组为广播组成员。多个分机组用, 隔开。例如: "allowextengroup": "sales,support"。
enablekeyhanup	是	String	是否开启 按*应答 功能。启用该功能，分机可以按*应答广播。应答后，该分机与发起者进行通话，其他分机的广播终止。 • yes: 开启。 • no: 关闭。
allowexten	是	String	广播组成员。选择分机为广播组成员。多个分机用, 隔开。例如: "allowexten": "1000,1001,1002"。
单向组播 ("duplex": "multicast")			
multicastip	是	String	如果广播组类型为 multicast，填写发送广播的组播地址。

名称	是否必需	类型	描述
enablekeyhanup	是	String	是否开启 按*应答 功能。启用该功能，分机可以按*应答广播。应答后，该分机与发起者进行通话，其他分机的广播终止。 • yes: 开启。 • no: 关闭。

实际示例

请求示例

```
POST /api/v1.1.0/paginggroup/update?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "id": "8",
  "number": "6300",
  "name": "6300",
  "duplex": "intercom",
  "allowexten": "1000,1001",
  "allowextengroup": "sales",
  "enablekeyhanup": "no",
  "multicastip": ""
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

删除广播组

Endpoint

```
POST /api/v1.1.0/paginggroup/delete?token={token}
```

请求参数

名称	是否必需	类型	描述
number	是	String	广播组号码。 • 删除单个广播组：设置 number 为广播组号码。 例如： "number": "6300" • 删除多个广播组：设置 number 为广播组号码，多个广播组号码之间用 , 隔开。 例如： "number": "6300,6301" • 删除所有广播组：设置 number 为 all。

实际示例

请求示例

```
POST /api/v1.1.0/paginggroup/delete?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
    "number": "6300"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
    "status": "Success"
}
```

发起广播

通过该接口呼叫广播组号码，可以向广播组成员播放实时通知，也可以向广播组成员播放音乐。

实时广播 & 音乐广播

• 实时广播

通过分机向广播组成员播放实时通知。

· 音乐广播

系统直接向广播组成员播放音乐文件。

**注：**
只有广播组类型为 `paging` 或 `intercom` 时，才能使用音乐广播。

Endpoint

```
POST /api/v1.1.0/extension/dial_number?token={token}
```

请求参数

名称	是否必需	类型	描述
实时广播			
caller	是	String	使用哪个分机播放通知。
callee	是	String	广播组号码。
volume	否	String	音量调节。 取值范围：0 ~ 20
音乐广播			
callee	是	String	广播组号码。
volume	否	String	音量调节。 取值范围：0 ~ 20
prompt	是	String	语音文件名称。 注： • 文件名无需加文件后缀。 • 文件名必须包含字母，且不能带 +。 • 必须将语音文件上传到 PBX 的 自定义提示音 页面。 多个语音文件之间用+ 连接。例如：<code>"prompt:music1+music2"</code>
count	否	String	播放次数。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

使用分机1001播放通知：

```
POST /api/v1.1.0/
extension/dial_number?token=277ac400357b509b4a587ff2157f7ad5 HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "caller": "1001",
  "callee": "6300"
}
```

向广播组成员播放语音文件：

```
POST /api/v1.1.0/
extension/dial_number?token=277ac400357b509b4a587ff2157f7ad5 HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "callee": "6300",
  "volume": "",
  "prompt": "music1",
  "count": "3"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1550626703.189"
}
```

挂断广播音乐

如果系统当前正在向广播组成员播放音乐文件，你可以通过该接口挂断广播音乐。

Endpoint

```
POST /api/v1.1.0/paginggroup/hangup_music?token={token}
```

请求参数

名称	是否必需	类型	描述
pageid	是	String	广播组号码。

实际示例

请求示例

```
POST /api/v1.1.0/
paginggroup/hangup_music?token=277ac400357b509b4a587ff2157f7ad5 HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "pageid": "6300"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

语音通话

查询来电

查询当前外部来电的详细信息，如：主叫、被叫、通话状态、通过的中继等。

Endpoint

```
POST /api/v1.1.0/inbound/query?token={token}
```

请求参数

- 不带参数表示查询所有。
- 带参数表示查询单个或多个，多个时需用 `,` 隔开。

名称	是否必需	类型	描述
inboundid	否	Int	外线来电编号。  注： 从 PBX 发送的通话报告中获取 inboundid 的值。

响应参数

名称	类型	描述
inbound	Object	来电，由外线呼入的外部通话。
inboundid	Int	外线来电编号，通过该参数对来电进行转接、查询、挂断等操作。
from	String	主叫号码。
to	String	被叫号码。
callee	Object	来电的通话方，可能为分机、IVR、去电。
trunk	String	呼入时通过的中继名。
status	String	通话状态 • Talking：通话进行中。 • Progress：呼叫处理中。 • Wait：呼叫等待。

实际示例

请求示例

```
POST /api/v1.1.0/inbound/query?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
```

```

    "inboundid": "1495698433.203"
  }

```

响应示例

```

HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "inbound": [
    {
      "inboundid": "1495698433.203",
      "from": "1000",
      "to": "1002",
      "trunk": "SIP-142",
      "status": "Talking",
      "callee": [
        {
          "extid": "1002"
        }
      ]
    }
  ]
}

```

查询去电

查询当前 IPPBX 系统分机通过外线呼出的所有通话的详细信息，如：主叫、被叫、通话状态、通过的中继、目的地等。

Endpoint

```
POST /api/v1.1.0/outbound/query?token={token}
```

请求参数

- 不带参数表示查询所有。
- 带参数表示查询单个或多个，多个时需用 , 隔开。

名称	是否必需	类型	描述
outboundid	否	Int	外线去电编号。

名称	是否必需	类型	描述
			 注： 从 PBX 发送的 通话报告 中获取 outboundid 的值。

响应参数

名称	类型	描述
outbound	Object	去电，为呼出到外线的外部通话。
outboundid	String	外线去电编号，通过该参数对去电进行转接、查询、挂断等操作。
from	String	主叫号码。
to	String	被叫号码。
trunk	String	呼出时通过的中继名称。
status	String	通话状态 • Talking：通话进行中。 • Progress：通话处理中。 • Wait：呼叫等待中。

实际示例

请求示例

```
POST /api/v1.1.0/outbound/query?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "outboundid": "1495705009.316"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "outbound": [
    {
      "outboundid": "1495705009.316",
      "from": "1002",
      "to": "41000",
    }
  ]
}
```

```
        "trunk": "SIP-142",
        "status": "Talking"
    }
  ]
}
```

分机互拨

通过本接口可以让一个分机呼叫另一个分机，从而使两者能够建立通话。

发送分机 A 拨分机 B 的请求后，分机 A 先响铃，摘机后，分机 B 响铃。

如果开启了自动应答，则主叫方直接听到回铃音，被叫方响铃。

Endpoint

```
POST /api/v1.1.0/extension/dial_extension?token={token}
```

请求参数

名称	是否必需	类型	描述
caller	是	String	主叫分机。
callee	是	String	被叫分机。
autoanswer	否	String	是否自动接听（只针对 SIP 线路有效，且需要话机支持，此参数不带则默认为不自动应答） • yes：是。 • no：否。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

分机 1005 呼叫分机 1006。

```
POST /api/v1.1.0/extension/dial_extension?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
```

```
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "caller": "1005",
  "callee": "1006",
  "autoanswer": "no"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1547397927.431"
}
```

拨打外线号码

通过本接口可以让分机向外线发起呼叫，从而使两者能够建立通话。（此接口只适用呼叫外线电话）。

条件

- 需通过呼出路由匹配呼出，所以该分机必须要有使用该呼出路由的权限。
- 发送分机拨打外线请求后，分机先响铃，摘机后，再拨打外线。如果开启了自动应答，则主叫方直接听到回铃音，被叫方响铃。

Endpoint

```
POST /api/v1.1.0/extension/dial_outbound?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	分机号。
outto	是	String	要拨打的外线号码。
autoanswer	否	String	是否自动接听（只针对 SIP 线路有效，且需要话机支持，此参数不带则默认为不自动应答）

名称	是否必需	类型	描述
			• yes: 是。 • no: 否。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

```
POST /api/v1.1.0/
extension/dial_outbound?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1002",
  "outto": "971837373",
  "autoanswer": "no"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1495710387.333"
}
```

挂断分机（强拆）

通过本接口可以挂断指定分机的当前通话，返回挂断成功或失败的结果。

Endpoint

```
POST /api/v1.1.0/extension/hangup?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	要挂断通话的指定分机号。

实际示例

请求示例

```
POST /api/v1.1.0/extension/hangup?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000"
}
```

响应示例

请求成功。

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

挂断来电（强拆）

通过本接口可以指定挂断外线呼入到 PBX 系统的通话。

Endpoint

```
POST /api/v1.1.0/inbound/hangup?token={token}
```

请求参数

名称	是否必需	类型	描述
inboundid	是	String	外线来电编号。

实际示例

请求示例

```
POST /api/v1.1.0/inbound/hangup?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "inboundid": "1495698510.206"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

挂断去电（强拆）

通过本接口可以指定挂断当前 IPPBX 系统分机通过外线呼出的通话。

Endpoint

```
POST /api/v1.1.0/outbound/hangup?token={token}
```

请求参数

名称	是否必需	类型	描述
outboundid	是	String	外线去电编号。

实际示例

请求示例

```
POST /api/v1.1.0/outbound/hangup?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "outboundid": "1495705264.322"
}
```

```
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

通话保持

通过本接口可将指定分机的当前通话保持。如需恢复通话，可使用通话接回接口。

Endpoint

```
POST /api/v1.1.0/extension/hold?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	保持该分机的当前通话。

实际示例

请求示例

```
POST /api/v1.1.0/extension/hold?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

恢复通话

通过本接口可将分机保持的通话重新恢复。

Endpoint

```
POST /api/v1.1.0/extension/unhold?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	恢复被保持的通话。

实际示例

请求示例

```
POST /api/v1.1.0/extension/unhold?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

通话静音

PBX 分机通话时，可利用该接口将分机静音。即分机的通话方不能听到分机的声音，而分机可以听到对方的声音。如需取消静音，请使用取消静音接口。通话结束后，分机也会自动取消静音。

Endpoint

```
POST /api/v1.1.0/extension/mute?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	要静音分机的分机号。

实际示例

请求示例

请求将分机1000的通话静音。

```
POST /api/v1.1.0/extension/mute?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

取消静音

通过被接口可将被静音的分机取消静音。

Endpoint

```
POST /api/v1.1.0/extension/unmute?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	要取消静音的分机号。

实际示例

请求示例

```
POST /api/v1.1.0/extension/unmute?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

普通监听

通过本接口可以实现一个分机监听另一个分机的当前通话。

监听步骤

1. 分机 A 空闲，分机 B 正在通话，（可通过查询分机列表获取分机状态）。
2. 执行分机 A 普通监听分机 B 的命令，即分机 A 可以监听到分机 B 和分机 B 的通话方，但是分机 B 和分机 B 的通话方听不到分机 A 的声音。
3. 执行成功时，分机 A 自动应答后即可监听到分机 B 的通话内容。



注：

有的话机或软电话可能不支持自动应答，则话机或软电话会先响铃，手动接听后开始监听。

普通监听条件

- 监听方的监听方式为普通监听或为通用模式。
- 被监听方必须开启允许被监听，否则不能被监听；一个分机最多只能同时被一个分机监听。

- 只能监听同一台 IPPBX 上的分机。

Endpoint

```
POST /api/v1.1.0/extension/listen?token={token}
```

请求参数

名称	是否必需	类型	描述
listener	是	Int	监听的分机号。
listenedext	是	Int	被监听的分机号。

实际示例

请求示例

分机1005 监听分机1002的通话。

```
POST /api/v1.1.0/extension/listen?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "listener": "1005",
  "listenedext": "1002"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

密语监听

通过本接口可让一个分机密语监听另一个分机的通话，即：监听方和被监听方两个分机之间可以互相通话，但是被监听分机的通话方不能和监听方通话。

监听步骤

1. 分机 A 当前空闲，分机 B 正在通话中（可通过查询分机列表获取分机状态）。
2. 执行分机 A 密语监听分机 B 的命令。
3. 执行成功时，分机 A 自动应答后即可密语监听分机 B 的通话。



注：

有的话机或软电话可能不支持自动应答，则话机或软电话会先响铃，手动接听后开始监听。

密语监听条件

- 监听方的监听方式必须是密语监听或为通用模式。
- 被监听方必须开启允许被监听。
- 只能监听同一台 PBX 设备上的分机。

Endpoint

```
POST /api/v1.1.0/extension/whisper?token={token}
```

请求参数

名称	是否必需	类型	描述
whisperer	是	Int	密语监听的分机号。
whisperedext	是	Int	被监听的分机号。

实际示例

请求示例

```
POST /api/v1.1.0/extension/whisper?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "whisperer": "1005",
  "whisperedext": "1002"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

强插监听

通过本接口可让一个分机强制插入到另一个分机的通话中去，从而形成三方通话。

监听步骤

1. 分机 A 当前空闲，分机 B 正在通话中（可通过查询分机列表获取分机状态）。
2. 执行分机 A 强插监听分机 B 的命令。
3. 执行成功时，分机 A 自动应答后即可强插到分机 B 的通话中。



注：

有的话机或软电话可能不支持自动应答，则话机或软电话会先响铃，手动接听后开始监听。

强插监听条件

- 监听方监听方式必须是强插监听或为通用模式。
- 被监听方必须开启允许被监听。
- 只能监听同一台 IPPBX 设备上的分机。

Endpoint

```
POST /api/v1.1.0/extension/barge?token={token}
```

请求参数

名称	是否必需	类型	描述
bargein	是	Int	强插监听的分机号。
bargedext	是	Int	被监听的分机号。

实际示例

请求示例

```
POST /api/v1.1.0/extension/barge?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "bargein": "1005",
  "bargedext": "1002"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

通话转接

当 PBX 的分机正在通话时，可使用该接口将通话转移到其他号码，如分机、队列、响铃组、IVR、或外线号码。

Endpoint

```
POST /api/v1.1.0/calltransfer?token={token}
```

请求参数

参数	是否必需	类型	描述
callid	是	String	外线来电编号。
transferor	是	String	指定由通话中的哪一方发起转移，发起转移后，转移方将自动挂断电话。
transferto	是	String	转移到哪个号码。
fromext	否	String	如果转移通话到外部号码，采用哪个分机的呼出权限。

实际示例

请求示例

```
POST /api/v1.1.0/calltransfer?token=277ac400357b509b4a587ff2157f7ad5
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "callid": "1550576451.121",
  "transferor": "1001",
  "transferto": "1003",
  "fromext": ""
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1550576451.121"
}
```

外线来电转接

通过该接口可以将外线的来电转移到其他号码，如分机、队列、响铃组、IVR、或外线号码。



注：
不管来电是在响铃还是正在通话的状态皆可转移到其他号码。

Endpoint

```
POST /api/v1.1.0/inbound/transfer_number?token={token}
```

请求参数

参数	是否必需	类型	描述
inboundid	是	String	外线来电编号。
number	是	String	来电转移到哪个号码。
fromext	否	String	采用哪个分机的呼出权限。

响应参数

参数	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

请求将来电转接给内部分机4001。

```
POST /api/v1.1.0/
inbound/transfer_number?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "inboundid": "1495698591.209",
  "number": "4001",
  "fromext": ""
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1495698591.209"
}
```

外线去电转接

通过本接口可将呼出到外线的电话转接给其他号码，如分机、IVR或外部号码。



注：

不管外线电话是在响铃还是接通的状态皆可转移其他号码。

Endpoint

```
POST /api/v1.1.0/outbound/transfer_number?token={token}
```

请求参数

名称	是否必需	类型	描述
outboundid	是	String	外线去电编号。
number	是	String	转接到哪个号码。
fromext	是	String	使用哪个分机的权限呼出。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

请求将编号为1495698591.209的外线去电转接给外线号码18283773。

```
POST /api/v1.1.0/  
outbound/transfer_number?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1  
Content-Type:application/json; charset=utf-8  
Host: 192.168.5.150  
{  
  "outboundid": "1495698591.209",  
  "number": "18283773",  
  "fromext": "4000"  
}
```

响应示例

```
HTTP/1.1 200 OK  
Access-control-allow-origin: *  
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  
{  
  "status": "Success",  
  "callid": "1495698591.209"  
}
```

通过 IVR 拨打分机

通过 IVR 呼叫分机，向分机播放语音，如播放台风警报等语音。分机可根据 IVR 语音提示，执行相关的按键操作。

Endpoint

```
POST /api/v1.1.0/ivr/dial_extension?token={token}
```

请求参数

名称	是否必需	类型	描述
ivrid	是	Int	IVR 号码。
extid	是	String	要呼叫的分机号。
autoanswer	是	String	是否自动接听(只针对 SIP 分机有效，且需要话机支持，不带此参数则默认为不启用自动应答)。 • yes: 是。 • no: 否。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

```
POST /api/v1.1.0/ivr/dial_extension?token=1e3b3ebbb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "ivrid": "6500",
  "extid": "1002",
  "autoanswer": "no"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1495698843.215"
```

```
}
```

通过IVR 拨打外线

通过 IVR 拨打外线电话，向外部人员播放语音，如播放台风警报等语音。外部人员可根据 IVR 语音提示，执行相关的按键操作。



注：

拨打外线电话，需使用分机的呼出权限呼出。

Endpoint

```
POST /api/v1.1.0/ivr/dial_outbound?token={token}
```

请求参数

名称	是否必需	类型	描述
ivrid	是	Int	IVR 号码。
outto	是	String	呼出的外线号码（匹配呼出规则）。
fromext	是	String	采用哪个分机的呼出权限。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

```
POST /api/v1.1.0/ivr/dial_outbound?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "ivrid": "6500",
  "outto": "18383992993",
  "fromext": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1495698843.215"
}
```

自动拨号（队列）

PBX 会自动拨打外线号码，当外线被接听后，PBX 会将这通电话转接到指定的队列。



注：

- 拨打外线电话，需使用分机的呼出权限呼出。
- 呼叫的外线号码要符合呼出路由的规则。

Endpoint

POST /api/v1.1.0/queue/dial_outbound?token=*{token}*

请求参数

名称	是否必需	类型	描述
queueid	是	Int	队列号码。
outto	是	String	要拨打的外线号码。
fromext	是	String	采用哪个分机的呼出路由权限。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

```
POST /api/v1.1.0/queue/dial_outbound?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "queueid": "6700",
  "outto": "118396210850",
  "fromext": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1495698843.215"
}
```

自动拨号（响铃组）

PBX 会自动拨打外线号码，当外线被接听后，PBX 会将这通电话转接到指定的响铃组。



注：

- 拨打外线电话，需使用分机的呼出权限呼出。
- 呼叫的外线号码要符合呼出路由的规则。

Endpoint

```
POST /api/v1.1.0/ringgroup/dial_outbound?token={token}
```

请求参数

名称	是否必需	类型	描述
ringgroupid	是	Int	响铃组号码。
outto	是	String	要拨打的外线号码。
fromext	是	String	采用哪个分机的呼出权限。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

```
POST /api/v1.1.0/
ringgroup/dial_outbound?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "ringgroupid": "6200",
  "outto": "118396210850",
  "fromext": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1495698843.215"
}
```

双向外呼

通过本接口可让 PBX 依次向外发起两路呼叫，并以 PBX 作为中间点建立连接，从而实现两个外部电话之间建立通话。

操作步骤

1. 拨打主叫号码，主叫摘机后听到回铃音。
 - 拨打被叫号码，被叫响铃。被叫摘机后双方建立通话。

条件

- 拨打外线电话，需使用分机的呼出权限呼出。

- PBX 至少有两条空闲可用的中继。
- 呼叫的外线号码要符合呼出路由的规则。

Endpoint

```
POST /api/v1.1.0/outbound/dial_outbound?token={token}
```

请求参数

名称	是否必需	类型	描述
caller	是	String	呼出的外线号码（匹配呼出规则）。
outto	是	String	呼出的外线号码（匹配呼出规则）。
fromext	是	Int	采用哪个分机的呼出路由权限。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

```
POST /api/v1.1.0/
outbound/dial_outbound?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "caller": "118396210850",
  "outto": "98237374",
  "fromext": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1495698843.215"
}
```

来电接听控制

如果在 PBX 的 API 配置界面，启用了中继的 [来电接听控制](#)，系统将会预留10秒，让第三方服务器控制该中继的来电。

应用服务器可以对外线来电进行如下类型的控制：

- **接听：**使用 [接听来电](#) API 接口接听该来电，进入对应的呼入路由目的地。
- **拒接：**使用 [拒接来电](#) API 接口拒接该来电。该来电将被挂断，且 PBX 上不会有该来电的通话记录。

超时处理机制

如果未在规定时间内（10秒）处理来电，默认接听来电。

接听来电

如果在 PBX 的 API 配置界面，启用了中继的 [来电接听控制](#)，系统将会预留10秒，让第三方服务器控制该外线来电。通过该接口，可接听该外线来电。

Endpoint

```
POST /api/v1.1.0/inbound/accept?token={token}
```

请求参数

名称	是否必需	类型	描述
inboundid	是	String	外线来电编号。

实际示例

请求示例

```
POST /api/v1.1.0/inbound/accept?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "inboundid": "1495703883.314"
}
```

响应示例

```

HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}

```

拒接来电

如果在 PBX 的 API 配置界面，启用了中继的 [来电接听控制](#)，系统将会预留10秒，让第三方服务器控制该外线来电。通过该接口，可拒接该外线来电。

Endpoint

```
POST /api/v1.1.0/inbound/refuse?token={token}
```

请求参数

名称	是否必需	类型	描述
inboundid	是	String	外线来电编号。

实际示例

请求示例

```

POST /api/v1.1.0/inbound/refuse?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "inboundid": "1495703883.314"
}

```

响应示例

```

HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}

```

播放语音

给分机播放语音

通过本接口可给内部分机播放语音，PBX 拨打分机，分机响铃接起后，听到提示音。

Endpoint

```
POST /api/v1.1.0/extension/playprompt?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	分机号。
prompt	是	String	 注： • 文件名无需加文件后缀。 • 文件名必须包含字母，且不能带+。 • 必须将语音文件上传到 PBX 的自定义提示音页面。 多个语音文件之间用+连接。例如: "prompt:music1+music2"
autoanswer	否	String	是否自动接听。 • yes: 是。 • no: 否。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

播放一个语音文件。

```
POST /api/v1.1.0/
extension/playprompt?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000",
  "prompt": "hello111",
  "autoanswer": "no"
}
```

播放多个语音文件。

```
POST /api/v1.1.0/
extension/playprompt?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000",
  "prompt": "1+queue1+queue2",
  "autoanswer": "no"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "callid": "1495697869.171"
}
```

给外线播放语音

通过本接口可给外线号码播放语音，PBX 拨打外线，外线响铃接起后，听到提示音。



限制：

。

Endpoint

```
POST /api/v1.1.0/outbound/playprompt?token={token}
```

请求参数

名称	是否必需	类型	描述
outto	是	String	外线号码。
prompt	是	String	语音文件名称。  注： • 文件名无需加文件后缀。 • 文件名必须包含字母，且不能带+。 • 必须将语音文件上传到 PBX 的自定义提示音页面。 多个语音文件之间用+连接。例如： "prompt:music1+music2"
fromext	否	Int	借用呼出权限的分机号。

响应参数

名称	类型	描述
callid	String	该通通话的 id。

实际示例

请求示例

```
POST /api/v1.1.0/outbound/playprompt?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "outto": "41000",
  "prompt": "hello111",
  "fromext": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
}
```

```
"callid": "1495697869.171"
}
```

语音留言

查询语音留言

查询分机的语音留言。

Endpoint

```
POST /api/v1.1.0/voicemail/query?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	否	Int	要查询哪个分机的语音留言。 • 查询单个分机: 设置 extid 为分机号码。 例如: <code>"extid": "1000"</code> • 查询多个分机: 设置 extid 为分机号码, 多个分机之间用 <code>,</code> 隔开。 例如: <code>"extid": "1000,1001"</code> • 查询所有分机: 不带 extid 请求参数。

响应参数

 **注:**
如果查询的分机没有语音留言文件, PBX 只响应查询失败或者成功的参数。

名称	类型	描述
voicemails	Object	Voicemails 对象参数。
extid	String	查询的分机号码。
voicemail	Object	Voicemail 对象参数。
voicemailfile	String	语音留言文件名。

名称	类型	描述
messagefrom	String	语音留言是哪个号码发送的。
voicemailstatus	String	语音留言的状态 • read: 已读。 • unread: 未读。
voicemaildate	String	收到语音留言的时间。
voicemaillength	String	语音留言的时长。

实际示例

请求示例

查询分机1000的语音留言。

```
POST /api/v1.1.0/voicemail/query?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000"
}
```

查询所有分机的语音留言。

```
POST /api/v1.1.0/voicemail/query?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "voicemails": [
    {
      "extid": "1000",
      "voicemail": [
        {
          "voicemailfile": "msg0000.wav",
          "messagefrom": "2000",
          "voicemailstatus": "read",
```

```
{  "voicemaildate": "2018-05-04 18:00:00",
    "voicemaillength": "42"
}

{
    "voicemailfile": "msg0001.wav",
    "messagefrom": "2000",
    "voicemailstatus": "unread",
    "voicemaildate": "2018-05-04 18:30:00",
    "voicemaillength": "42"
}
```

删除语音留言

删除分机的语音留言。

Endpoint

```
POST /api/v1.1.0/voicemail/delete?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	要删除哪个分机的语音留言文件。
voicemailfile	是	String	语音留言文件名。 • 填写具体语音留言的名称，删除指定的语音留言文件。 • all：删除指定语音留言状态的所有的语音留言文件。
voicemailstatus	是	String	语音留言的状态。 • read：删除已读的语音留言文件。 • unread：删除未读的语音留言文件。 • all：忽略语音留言的状态，删除指定的语音留言文件或者所有的语音留言文件。

实际示例

请求示例

```
POST /api/v1.1.0/voicemail/delete?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000",
  "voicemailfile": "msg0000.wav",
  "voicemailstatus": "read"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

下载语音留言

下载分机的语音留言文件。

Endpoint

```
POST /api/v1.1.0/voicemail/get_random?token={token}
```

请求参数



注：

要下载分机的某个语音留言文件，需要先向 PBX 请求该语音留言文件的随机串。

名称	是否必需	类型	描述
extid	是	Int	要下载哪个分机的语音留言文件。
voicemailfile	是	String	语音留言文件名。
voicemailstatus	是	String	语音留言的状态。 • read：要下载的语音留言已读，从 read 文件夹获取指定的语音留言文件。 • unread：要下载的语音留言未读，从 unread 文件夹获取指定的语音留言文件。

名称	是否必需	类型	描述
allowedip	否	String	设置允许下载语音留言的设备 IP 地址。 • 不填写allowedip：只能通过第三方服务器可以下载文件。 • allowedip 值为空：只能通过第三方服务器可以下载文件。 • allowedip 填写 IP 地址：只能通过该 IP 地址或第三方服务器可以下载文件。  注： • allowedip 只能填写一个 IP 地址。

响应参数

名称	类型	描述
extid	Int	哪个分机的语音留言。
voicemailstatus	String	语音留言的状态。 • read：已读。 • unread：未读。
voicemailfile	String	语音留言文件的名称。
random	String	语音留言文件的随机串。 使用这个随机串组合成下载播放该录音文件请求。

实际示例

请求示例

```
POST /api/v1.1.0/voicemail/get_random?token=1e3b3ebbb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1004",
  "voicemailfile": "msg0000.wav",
  "voicemailstatus": "unread"
}
```

响应示例

```

HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "extid": "1004",
  "voicemailstatus": "unread",
  "voicemailfile": "msg0000.wav",
  "random": "70f477822e6a379d7359b2d926ea95a5"
}

```

语音留言下载链接示例

使用 API 响应的 random 随机串组合成语音留言下载链接。

语音留言下载链接格式：

https://{*pbx ip*}:{*https port*}/api/v1.1.0/voicemail/download?
extid=*#extid#*&voicemailstatus=*#voicemailstatus#*&token=*#token#*&voicemailfile=*{voicemailfile}*&

下面是语音留言下载链接示例：

https://192.168.5.150:8088/api/v1.1.0/voicemail/download?
extid=1005&voicemailstatus=unread&token=72d6640a05811af51e0279461e5272ce&voicemailfile=msg000

下载录音文件

通过本接口可获取 PBX 中的全局录音文件。

下载步骤

1. 通过 [‘通话记录’ 报告](#) 中的 **recording** 参数获取到全局录音文件的名称。
2. 通过获取到的名称从 PBX 获取有关该文件的一个随机串。
3. 通过该随机串组合成录音文件的 [下载链接](#)。



注：

随机串的有效时长为30秒，并且每个随机串使用过一次则失效。

Endpoint

```
POST /api/v1.1.0/recording/get_random?token={token}
```

请求参数

名称	是否必需	类型	描述
recording	是	String	全局录音文件名称。
allowedip	否	String	设置允许下载录音文件的设备 IP 地址。 - 不填写allowedip：只能通过第三方服务器可以下载文件。 - allowedip 值为空：只能通过第三方服务器可以下载文件。 - allowedip 填写 IP 地址：只能通过该 IP 地址或第三方服务器可以下载文件。  注： - allowedip 只能填写一个 IP 地址。

响应参数

名称	类型	描述
recording	String	全局录音文件名称。
random	String	有关全局录音文件的一个随机串。使用这个随机串组合成下载播放该录音文件请求。

实际示例

请求示例

请求录音文件的随机串。

```
POST /api/v1.1.0/
recording/get_random?token=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "recording": "20170901181806-1504261084.7-1001-1003-Internal.wav"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
```

```
{
  "status": "Success",
  "recording": "20170901181806-1504261084.7-1001-1003-Internal.wav",
  "random": "120732c546381fb020f17fba676b0ea0"
}
```

全局录音下载链接

使用随机串（random）组合成录音文件的下载链接。

录音下载链接格式：

https://{pbx_ip}:{https_port}/api/v1.1.0/recording/download?
recording={recording}&random={random}&token={token}

录音下载链接示例：

https://192.168.5.150:8088/api/v1.1.0/recording/download?
recording=20170901181806-1504261084.7-1001-1003-
Internal.wav&random=120732c546381fb020f17fba676b0ea0&token=75c5891b32203d0615f9e3753a7cb77

下载通话记录

下载指定分机或所有分机的通话记录。

下载步骤

- 1. 获取通话记录文件的随机串。
- 2. 使用随机串，组合成通话记录文件的[下载链接](#)。



注：
随机串的有效时长为30秒，并且每个随机串使用过一次则失效。

Endpoint

```
POST /api/v1.1.0/cdr/get_random?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	String	设置分机。

名称	是否必需	类型	描述
			• 所有分机：设置 <code>extid</code> 为 <code>all</code>，表示下载所有分机的通话记录。 • 单个分机：设置 <code>extid</code> 为分机号码。 • 多个分机：设置 <code>extid</code> 为分机号码，多个分机号码之间用 <code>,</code> 隔开。例如：<code>"extid": "1000,1001,1002"</code>
<code>starttime</code>	是	String	查询通话记录的起始时间。 时间格式： <code>yyyy-mm-dd hh:mm:ss</code>
<code>endtime</code>	是	String	查询通话记录的结束时间。 时间格式： <code>yyyy-mm-dd hh:mm:ss</code>
<code>allowedip</code>	否	String	设置允许下载通话记录的设备 IP 地址。 • 不填写 <code>allowedip</code>：只能通过第三方服务器可以下载文件。 • <code>allowedip</code> 值为空：只能通过第三方服务器可以下载文件。 • <code>allowedip</code> 填写 IP 地址：只能通过该 IP 地址或第三方服务器可以下载文件。  注： • <code>allowedip</code> 只能填写一个 IP 地址。

响应参数

名称	类型	描述
<code>extid</code>	String	分机。
<code>starttime</code>	String	查询通话记录的起始时间。
<code>endtime</code>	String	查询通话记录的结束时间。
<code>random</code>	String	通话记录文件的随机串。使用这个随机串组合成该通话录音文件的下载链接。

实际示例

请求示例

获取所有分机的通话记录的随机码。

```
POST /api/v1.1.0/cdr/get_random?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "all",
  "starttime": "2018-11-07 00:00:00",
  "endtime": "2019-02-18 23:59:59"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "extid": "all",
  "starttime": "2018-11-07 00:00:00",
  "endtime": "2019-02-18 23:59:59",
  "random": "0cbcf8917c41c61c6e9fd3e57ecdd467"
}
```

通话记录下载链接

使用随机串（random）组合成通话记录的下载链接。

通话记录文件的格式为CSV。 [查看通话记录参数说明](#)。

通话记录下载链接格式：

```
https://{pbx_ip}:{https_port}/api/v1.1.0/cdr/download?
extid={extid}&starttime={starttime}&endtime={endtime}&token={token}&random={random}
```

通话记录下载链接示例：

```
https://192.168.5.150:8088/api/v1.1.0/cdr/download?
extid=all&starttime=2018-11-27 00:00:00&endtime=2019-02-18
23:59:59&token=48400f35207bb9c330a0bdaf4a5633e2&random=0cbcf8917c41c61c6e9fd3e57ecdd467
```

酒店叫醒服务

添加闹铃

通过本接口可以给酒店房间的分机设置闹铃。



注：

- 一次只能添加一个闹铃。如果一次添加多个，则只有第一个生效。
- 为多个分机设置相同的闹铃，分机之间使用, 隔开，如：

```
{"extid":"1000,1001","wakeup":  
  [{"time":"00:45","type":"onetime","repeats":"3","repeatinterval":"5"}]}
```

Endpoint

POST /api/v1.1.0/wakeupcall/create?token={token}

请求参数

名称	是否必需	类型	描述
extid	是	Int	分机号。
wakeup	是	Object	对象参数。
time	是	String	闹铃时间。
type	是	String	闹铃频率。 • onetime：一次 • everyday：每天 • custom：自定义 注： 设置为 onetime 时，当闹铃完成设置的重复次数之后，该条闹铃 将被自动删除。
weekdays	否	String	类型为自定义时的选项。 可选值：0、1、2、3、4、5、6 1-6表示周一到周六，0：为周日。
repeats	是	String	闹铃的重复次数 可选值：1、2、3

名称	是否必需	类型	描述
Repeatinterval	否	String	重复时间，设置好重复次数和重复时间后，话机响铃一次后，间隔所设置的重复时间后会再次响铃。 单位：分钟。
Prompt	否	String	闹铃提示音 此提示音必须为自定义提示音，不加字段则用系统默认的macroform-cold_day提示音。

实际示例

请求示例

```
POST /api/v1.1.0/wakeupcall/create?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000",
  "wakeup": [
    {
      "time": "00:45",
      "type": "onetime",
      "repeats": "3",
      "repeatinterval": "5"
    }
  ]
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

查询闹铃

查询特定分机已添加的闹铃信息。

Endpoint

```
POST /api/v1.1.0/wakeupcall/query?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	分机号。

响应参数

名称	类型	描述
extid	Int	分机号。
wakeup	Object	对象参数。
wakeupid	Int	闹铃的唯一标识。
time	String	闹铃时间。
type	String	闹铃频率 • onetime: 一次 • everyday: 每天 • custom: 自定义  注： 设置为 onetime 时，当闹铃完所设置的重复次数之后，该条闹铃将被自动删除。
weekdays	String	类型为自定义时的选项。 可选值：0、1、2、3、4、5、6 1-6表示周一到周六，0：为周日。
repeats	String	闹铃的重复次数 可选值：1、2、3
Repeatinterval	String	重复时间，设置好重复次数和重复时间后，话机响铃一次后，间隔所设置的重复时间后会再次响铃。 单位：分钟。
Prompt	String	闹铃提示音 此提示音必须为自定义提示音，不加字段则用系统默认的 macroform-cold_day提示音。

实际示例

请求示例

```
POST /api/v1.1.0/wakeupcall/query?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success",
  "wakeups": [
    {
      "extid": "1000",
      "wakeup": [
        {
          "wakeupid": "1",
          "time": "00:45",
          "type": "onetime",
          "prompt": "macroform-cold_day",
          "repeats": "3",
          "repeatinterval": "5"
        }
      ]
    }
  ]
}
```

更改闹铃设置

更改特定分机的闹铃设置。

Endpoint

POST /api/v1.1.0/wakeupcall/update?token={token}

请求参数

名称	是否必需	类型	描述
extid	是	Int	分机号。
wakeupid	是	Int	闹铃的唯一标识。
wakeup	是	Object	对象参数。
time	是	String	要修改的时间。
type	是	String	闹铃频率。 • onetime • everyday • custom  注： 设置为 onetime 时，当闹铃完所设置的重复次数之后，该条闹铃将被自动删除。
weekdays	否	String	类型为自定义时的选项。 可选值：0、1、2、3、4、5、6 1-6表示周一到周六，0：为周日。
repeats	否	String	闹铃的重复次数。 可选值：1、2、3
Repeatinterval	否	String	重复时间，设置好重复次数和重复时间后，话机响铃一次后，间隔所设置的重复时间后会再次响铃。 单位：分钟。
Prompt	否	String	闹铃提示音。 此提示音必须为自定义提示音，不加字段则用系统默认的macroform-cold_day 提示音。

实际示例

请求示例

请求更改分机1000的闹铃设置。

```
POST /api/v1.1.0/wakeupcall/update?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000",
  "wakeupid": "1",
  "wakeup": [
    {
      "time": "11:00",
      "type": "onetime",
      "repeats": "3",
      "repeatinterval": "5"
    }
  ]
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

删除闹铃

删除指定分机的闹铃。

Endpoint

```
POST /api/v1.1.0/wakeupcall/delete?token={token}
```

请求参数

名称	是否必需	类型	描述
extid	是	Int	分机号。

名称	是否必需	类型	描述
wakeupid	否	String	闹铃的唯一标识。 • 删除单个闹铃: 设置 wakeupid 为闹铃 id。 例如: <code>"wakeupid": "1"</code> • 删除多个闹铃: 设置 wakeupid 为闹铃 id, 多个闹铃之间用 <code>,</code> 隔开。 例如: <code>"wakeupid": "1,2"</code> • 删除所有闹铃: 不带 wakeupid 请求参数, 表示删除该分机的所有闹铃。

实际示例

请求示例

删除分机1000分闹铃1。

```
POST /api/v1.1.0/wakeupcall/delete?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000",
  "wakeupid": "1"
}
```

删除分机1000的所有闹铃。

```
POST /api/v1.1.0/wakeupcall/delete?token=1e3b3ebb6a974cb42ed31de5413df52d
HTTP/1.1
Content-Type:application/json; charset=utf-8
Host: 192.168.5.150
{
  "extid": "1000"
}
```

响应示例

```
HTTP/1.1 200 OK
Access-control-allow-origin: *
Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE
{
  "status": "Success"
}
```

API 报告示例

系统启动报告

当 PBX 系统启动后，PBX 会向应用服务器推送该报告。



注：

前提条件：应用服务端已连接上 API 服务器，并且完成认证。

报告参数

名称	类型	描述
action	String	启动系统事件。
sn	String	PBX 的 SN 码。

报告示例

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json
{
  "action": "BootUp",
  "sn": "369351034049"
}
```

配置变更报告

当 PBX 的配置发生变化时，PBX 会向应用服务器发送配置变更报告，以便于应用服务器及时更新和同步 PBX 的相关配置。



注：

- 目前只支持发送分机和中继配置变更的报告。



- PBX 只报告哪个分机或哪个中继配置有变更，具体变更信息，你需要登录 PBX 网页查看。

报告参数

名称	类型	描述
action	String	配置变更事件。
type	String	类型。 • extension：分机。 • trunk：中继。
trunkname	String	配置有变更的中继名称。
operation	String	操作。 • add：添加。 • del：删除。 • update：更新。
sn	String	PBX 的 SN 码。

报告示例

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json
{
  "action": "ConfigChange",
  "type": "extension",
  "extid": "1001",
  "operation": "update",
  "sn": "369362089179"
}
```

分机状态变更报告

当分机状态发生变更时，PBX 主动向应用服务器推送变更信息。



注：
你需要提前在 PBX 的**二次开发接口** 配置页面勾选分机的状态监控。

分机状态

- **响铃 (Ringing)**：当分机有来电正在响铃时，PBX 会向应用服务器推送该报告。
- **忙 (Busy)**：当分机摘机时，PBX 会向应用服务器推送该报告。



注：
分机由空闲状态或响铃状态摘机后则分机的状态变为 Busy 状态。

- **空闲 (Idle)**：当分机由忙碌变成空闲时，PBX 会向应用服务器推送该报告。
- **已注册 (Registered)**：当 IP 分机由未注册变为注册上时，或者当 IP 分机的地址变更时，PBX 会向应用服务器推送该报告。
- **未注册 (Unregistered)**：当 IP 分机由注册上变为未注册时，PBX 会向应用服务器推送该报告。

报告参数

名称	类型	描述
action	String	分机状态变更事件。
extension	Int	状态变更的分机号。
status	String	变更后的状态。
sn	String	PBX 的 SN 码。

报告示例

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json
{
  "action": "ExtensionStatus",
  "extension": "1006",
  "status": "Registered",
  "sn": "369362089179"
}
```

‘响铃&回铃’ 报告

当 PBX 的分机拨打电话或收到来电时，PBX 会向第三方服务器发送回铃报告（ALERT）和响铃报告（RING）。



注：

- 必须在 PBX 的 API 配置界面启用分机的 [状态监控](#)，PBX 才会发送响铃报告和回铃报告。
- 同一个呼叫事件中，PBX 同时发送响铃报告和回铃报告。

报告参数-回铃事件

主叫拨打被叫号码后，被叫响铃后，主叫听到回铃音。

名称	类型	描述
action	String	回铃事件：ALERT。
callid	String	该通电话的 id。
call	Object	对象参数。
主叫为分机		
ext	Object	分机对象。
extid	String	主叫号码。
主叫为外线号码		
inbound	Object	外线来电对象。
from	String	主叫号码。
to	String	被叫号码。
trunk	String	通过哪条中继呼入。
inboundid	String	外线来电编号。
sn	String	PBX 的 SN 码。

报告参数-响铃事件

被叫收到来电后，被叫听到响铃音。



注：



响铃报告事件中会显示被叫号码和主叫号码。报告中，先显示被叫号码，再显示主叫号码。

名称	类型	描述
action	String	回铃事件：RING。
callid	String	该通电话的 id。
call	Object	对象参数。
主叫为分机，被叫为分机		
ext	Object	分机对象。
extid	String	分机号码。
主叫为分机，被叫为外线号码，先显示outbound对象，再显示ext对象。		
outbound	Object	外线去电对象。
from	String	主叫号码。
to	String	被叫号码。
trunk	String	通过哪条中继呼出。
outboundid	String	外线去电编码。
主叫为外线号码，被叫为分机，先显示ext对象，再显示inbound对象。		
inbound	Object	外线来电对象。
from	String	主叫号码。
to	String	被叫号码。
trunk	String	通过哪条中继呼入。
inboundid	String	外线来电编号。
sn	String	PBX 的 SN 码。

报告示例

分机 1001 呼叫分机 1002。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json

{
  "action": "ALERT",
  "callid": "1550710849.199",
```

```

    "call": [
      {
        "ext": {
          "extid": "1001"
        }
      }
    ],
    "sn": "369351034049"
  }
}
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json
{
  "action": "RING",
  "callid": "1550710849.199",
  "call": [
    {
      "ext": {
        "extid": "1002"
      }
    },
    {
      "ext": {
        "extid": "1001"
      }
    }
  ],
  "sn": "369351034049"
}

```

分机1001 拨打外线号码 9282883883。

```

POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
  "action": "ALERT",
  "callid": "1547396927.428",
  "call": [

```

```

    {
      "ext": {
        "extid": "1001"
      }
    }
  ],
  "sn": "369362089179"
}

```

POST / HTTP/1.1

User-Agent: WebAPI

Host: 192.168.5.150: 8260

Content-Type: application/json

Accept: application/json

```

{
  "action": "RING",
  "callid": "1547396927.428",
  "call": [
    {
      "outbound": {
        "from": "1001",
        "to": "9282883883",
        "trunk": "SPS-6-36",
        "outboundid": "1547396927.429"
      }
    },
    {
      "ext": {
        "extid": "1001"
      }
    }
  ],
  "sn": "369362089179"
}

```

外线号码123838838呼叫分机1002。

POST / HTTP/1.1

User-Agent: WebAPI

Host: 192.168.5.150: 8260

Content-Type: application/json

Accept: application/json

Content-Length: 98

```
{
  "action": "ALERT",
  "callid": "1550714393.225",
  "call": [
    {
      "inbound": {
        "from": "123838838",
        "to": "1002",
        "trunk": "spstrunk",
        "inboundid": "1550714393.225"
      }
    }
  ],
  "sn": "369351034049"
}
```

POST / HTTP/1.1
 User-Agent: WebAPI
 Host: 192.168.5.150: 8260

Content-Type: application/json
 Accept: application/json

```
{
  "action": "RING",
  "callid": "1550714393.225",
  "call": [
    {
      "ext": {
        "extid": "1002"
      }
    },
    {
      "inbound": {
        "from": "123838838",
        "to": "1002",
        "trunk": "spstrunk",
        "inboundid": "1550714393.225"
      }
    }
  ],
  "sn": "369351034049"
}
```

‘通话应答’ 报告

当分机接听内部来电或外部来电时，PBX 会向应用服务器发送“通话应答”报告。



注：
应答和被应答事件同时一一对应，通话两方，主叫方为被应答，被叫方为应答，两个事件同时存在。报告时，被叫在主叫的前面。

报告参数

名称	类型	描述
action	String	通话应答事件。
callid	String	该通通话的 id。
ext	String	分机对象。
extid	String	分机号。
inboundid outboundid	String	来电编码或去电编码。
outbound inbound	String	对象参数。
from	String	来电的原始主叫号码。
to	String	来电的原始被叫号码。
trunk	String	通过哪条中继呼入。
sn	String	PBX 的 SN 码。

报告示例

分机1005拨打分机1006，1006接听电话。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json

{
  "action": "ANSWER",
  "callid": "1547375031.65",
  "call": [
```

```

    {
      "ext": {
        "extid": "1006"
      }
    },
    {
      "ext": {
        "extid": "1005"
      }
    }
  ],
  "sn": "369362089179"
}

```

外部用户 3834847222 呼叫分机 1005，分机 1005 接听来电。

```

POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
  "action": "ANSWER",
  "callid": "1547375509.72",
  "call": [
    {
      "ext": {
        "extid": "1005"
      }
    },
    {
      "inbound": {
        "from": "3834847222",
        "to": "1005",
        "trunk": "SPS-6-36",
        "inboundid": "1547375509.72"
      }
    }
  ],
  "sn": "369362089179"
}

```

‘通话被应答’ 报告

当分机呼出的通话被应答时，PBX 会向应用服务器发送 “通话被应答” 报告。



注：
应答和被应答事件同时一一对应，通话两方，主叫方为被应答，被叫方为应答，两个事件同时报告，报告时，主叫在被叫的前面。

报告参数

名称	类型	描述
action	String	通话被应答事件。
callid	String	该通通话的 id。
ext	String	分机对象。
extid	String	分机号。
inboundid outboundid	String	来电编码或去电编码。
outbound inbound	String	对象参数。
from	String	来电的原始主叫号码。
to	String	来电的原始被叫号码。
trunk	String	通过哪条中继呼入。
sn	String	PBX 的 SN 码。

报告示例

分机1005呼叫分机1006，分机1006接听电话。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json

{
  "action": "ANSWERED",
  "callid": "1547375031.65",
  "call": [
```

```

    {
      "ext": {
        "extid": "1005"
      }
    },
    {
      "ext": {
        "extid": "1006"
      }
    }
  ],
  "sn": "369362089179"
}

```

分机 1005 呼叫外部号码 172838344，外部用户 172838344 接听电话。

```

POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
  "action": "ANSWERED",
  "callid": "1547375707.75",
  "call": [
    {
      "ext": {
        "extid": "1005"
      }
    },
    {
      "outbound": {
        "from": "1005",
        "to": "172838344",
        "trunk": "SPS-6-36",
        "outboundid": "1547375707.76"
      }
    }
  ],
  "sn": "369362089179"
}

```

‘通话结束’ 报告

当一路通话释放，即通话挂断时，PBX 会向第三方应用服务器发送两个 BYE 报告。在第一个 BYE 报告中，先记录主动挂断电话的用户号码，后记录被挂断电话的用户号码。

报告参数

名称	类型	描述
action	String	通话结束报告。
callid	String	该通通话的 id。
ext	String	分机对象。
extid	String	分机号。
inboundid	String	来电编码。
outboundid	String	去电编码。
outbound	String	对象参数。
inbound	String	对象参数。
from	String	来电的原始主叫号码。
to	String	来电的原始被叫号码。
trunk	String	通过哪条中继呼入。
sn	String	PBX 的 SN 码。

报告示例

分机1006 与分机1005 通话中，分机 1006 挂断电话。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json

{
  "action": "BYE",
  "callid": "1547375817.78",
  "call": [
    {
      "ext": {
        "extid": "1006"
      }
    }
  ],
}
```

```

    {
      "ext": {
        "extid": "1005"
      }
    }
  ],
  "sn": "369362089179"
}

```

POST / HTTP/1.1

User-Agent: WebAPI

Host: 192.168.5.150: 8260

Content-Type: application/json

Accept: application/json

```

{
  "action": "BYE",
  "callid": "1547375817.78",
  "call": [
    {
      "ext": {
        "extid": "1005"
      }
    },
    {
      "ext": {
        "extid": "1006"
      }
    }
  ],
  "sn": "369362089179"
}

```

分机 1005 与外部用户 12564555 通话中，外部用户 12564555 挂断电话。

POST / HTTP/1.1

User-Agent: WebAPI

Host: 192.168.5.150: 8260

Content-Type: application/json

Accept: application/json

```

{
  "action": "BYE",
  "callid": "1547376084.84",

```

```

    "call": [
      {
        "outbound": {
          "from": "1005",
          "to": "12564555 ",
          "trunk": "SPS-6-36",
          "outboundid": "1547376084.85"
        }
      },
      {
        "ext": {
          "extid": "1005"
        }
      }
    ],
    "sn": "369362089179"
  }

```

POST / HTTP/1.1

User-Agent: WebAPI

Host: 192.168.5.150: 8260

Content-Type: application/json

Accept: application/json

```

{
  "action": "BYE",
  "callid": "1547376084.84",
  "call": [
    {
      "ext": {
        "extid": "1005"
      }
    },
    {
      "outbound": {
        "from": "1005",
        "to": "12564555",
        "trunk": "SPS-6-36",
        "outboundid": "1547376084.85"
      }
    }
  ],
  "sn": "369362089179"
}

```

‘呼叫转移’ 报告

当 PBX 分机用户转移通话时，PBX 向应用服务器发送 “呼叫转移” 报告。



注：

- 此转移只代表分机操作的转移，如：
 - 分机拨打特征码*03，*3将通话转接。
 - 分机设置呼叫转移功能。
- API 控制的转移不报告。

报告参数

名称	类型	描述
action	String	呼叫转移事件。
callid	String	该通通话的 id。
ext	String	分机对象。
extid	String	发起转移的分机号。
inboundid	String	来电编码。
outboundid	String	去电编码。
outbound	String	对象参数。
inbound	String	对象参数。
from	String	来电的原始主叫号码。
to	String	来电的原始被叫号码。
trunk	String	通话过程中使用的中继名称。
sn	String	PBX 的 SN 码。

报告示例

分机 1005 与分机 1001 通话中，分机 1005 将电话转移到另外一个号码。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json
```

```
{
  "action": "Transfer",
  "callid": "1547378727.152",
  "call": [
    {
      "ext": {
        "extid": "1005"
      }
    },
    {
      "ext": {
        "extid": "1001"
      }
    }
  ],
  "sn": "369362089179"
}
```

分机 1001 设置呼叫转移到号码 A，分机 1006 呼叫 1001，电话转移到号码 A。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
  "action": "Transfer",
  "callid": "1547379044.160",
  "call": [
    {
      "ext": {
        "extid": "1006"
      }
    }
  ],
  "sn": "369362089179"
}
```

外部用户 182837733 呼入 PBX，与分机 1001 通话，分机 1001 将电话转移到另外一个号码。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
```

```
Content-Type: application/json
Accept: application/json

{
  "action": "Transfer",
  "callid": "1547379694.192",
  "call": [
    {
      "inbound": {
        "from": "182837733",
        "to": "1001",
        "trunk": "SPS-6-36",
        "inboundid": "1547379694.192"
      }
    }
  ],
  "sn": "369362089179"
}
```

‘呼叫失败’ 报告

当发起的呼叫失败时，PBX 会向应用服务器发送“呼叫失败”报告。



注：

只要是被叫方未接通的状况都定义为呼叫失败，转到语音留言算已接通。不管是 API 控制的呼叫还是手动拨号的呼叫，只要呼叫失败都会触发该事件。

报告参数

名称	类型	描述
action	String	呼叫失败事件。
reason	String	呼叫失败原因： • 404 NOT found：被叫号码为空号。 • 480：被叫没有接听来电。 • 484：呼出号码格式错误。 • 486 Busy Here：被叫拒接。 • 487 Request Terminated：被叫响铃后，主叫主动挂断电话。 • 503 Service Unavailable：中继不可用。

名称	类型	描述
		• 603 Declined: 被叫响铃超时。
callid	String	该通通话的 id。
ext	Int	分机对象。
extid	String	分机号。
inboundid outboundid	String	来电编码或去电编码。
outbound inbound	String	对象参数。
from	String	原始主叫号码。
to	String	原始被叫号码。
trunk	String	通过哪条中继呼入。
sn	String	PBX 的 SN 码。

报告示例

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json

{
  "action": "CallFailed",
  "reason": "486 Busy Here",
  "callid": "1536976927.37",
  "outbound": {
    "from": "1002",
    "to": "70871520926",
    "trunk": "Wanatel-1",
    "outboundid": "1536976927.38"
  },
  "ext": {
    "extid": "1002"
  },
  "sn": "369351034049"
}
```

‘按键信息’ 报告

通话过程中，当用户根据提示输入相关的按键（如通过 IVR 输入按键），PBX 向应用服务器推送“按键信息”报告。

报告参数

名称	类型	描述
action	String	按键事件。
callid	String	该通通话的 id。
ext	Int	分机对象。
extid	String	主叫分机号码。
info	String	单个按键信息。
inboundid outboundid	String	来电编码或去电编码。
outbound inbound	String	对象参数。
from	String	原始主叫号码。
to	String	原始被叫号码。
trunk	String	通过的中继名称。
flag	Int	标识 DTMF 按键的顺序。  注： flag 从0开始标识。0表示第一个 DTMF 按键；1表示第二个 DTMF 按键；以此类推。
infos	String	DTMF 按键串。
sn	String	PBX 的 SN 码。

报告示例

分机 1005 拨打 IVR 号码，输入按键 2。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json
```

```
{
  "action": "DTMF",
  "callid": "1547389829.281",
  "call": [
    {
      "ext": {
        "extid": "1005",
      }
    }
  ],
  "info": "2",
  "flag": "0",
  "sn": "369362089179"
}
```

外部号码 192883838 呼入 IVR 6500，拨打分机 1005。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json
{
  "action": "DTMF",
  "callid": "1547391097.305",
  "call": [
    {
      "inbound": {
        "from": "192883838",
        "to": "6500",
        "trunk": "SPS-6-36",
        "inboundid": "1547391097.305",
      }
    }
  ],
  "info": "1",
  "flag": "0",
  "sn": "369362089179"
}
```

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
```

```
Content-Type: application/json
Accept: application/json
{
  "action": "DTMF",
  "callid": "1547391097.305",
  "call": [
    {
      "inbound": {
        "from": "192883838",
        "to": "6500",
        "trunk": "SPS-6-36",
        "inboundid": "1547391097.305",
      }
    }
  ],
  "info": "0",
  "flag": "1",
  "sn": "369362089179"
}
```

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
```

```
Content-Type: application/json
Accept: application/json
{
  "action": "DTMF",
  "callid": "1547391097.305",
  "call": [
    {
      "inbound": {
        "from": "192883838",
        "to": "6500",
        "trunk": "SPS-6-36",
        "inboundid": "1547391097.305",
      }
    }
  ],
  "info": "0",
  "flag": "2",
  "sn": "369362089179"
}
```

```
}
```

```
POST / HTTP/1.1
```

```
User-Agent: WebAPI
```

```
Host: 192.168.5.150: 8260
```

```
Content-Type: application/json
```

```
Accept: application/json
```

```
{
  "action": "DTMF",
  "callid": "1547391097.305",
  "call": [
    {
      "inbound": {
        "from": "192883838",
        "to": "6500",
        "trunk": "SPS-6-36",
        "inboundid": "1547391097.305",

      }
    }
  ],
  "info": "5",
  "flag": "3",
  "sn": "369362089179"
}
```

```
POST / HTTP/1.1
```

```
User-Agent: WebAPI
```

```
Host: 192.168.5.150: 8260
```

```
Content-Type: application/json
```

```
Accept: application/json
```

```
{
  "action": "DTMF",
  "callid": "1547391097.305",
  "call": [
    {
      "inbound": {
        "from": "192883838",
        "to": "6500",
        "trunk": "SPS-6-36",
        "inboundid": "1547391097.305",

      }
    }
  ]
}
```

```
        }
      }
    ],
    "infos": "1005",
    "sn": "369362089179"
  }
}
```

分机 1005 拨打外部号码，进入对方的 IVR，输入按键 1。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
  "action": "DTMF",
  "callid": "1547391490.309",
  "call": [
    {
      "ext": {
        "extid": "1005"
      }
    }
  ],
  "info": "1",
  "flag": "0",
  "sn": "369362089179"
}
```

‘通话记录’ 报告

当通话结束后，PBX 向应用服务器实时发送通话详细记录（CDR）。

报告参数

名称	类型	描述
action	String	通话事件。
timestart	String	开始时间。
callfrom	String	主叫号码。

名称	类型	描述
callto	String	被叫号码。
callduration	String	通话时长。
talkduration	String	接听时长。
srctrunkname	String	源中继名称。
desttrunkname	String	目的中继名称。
status	String	通话状态。
type	String	通话类型。
pincode	Int	密码。
recording	String	全局录音文件名。
didnumber	String	来电用户拨打的号码。
agentringtime	String	队列坐席从响铃到接听的时间。
sn	String	PBX 的 SN 码。

报告示例

分机 1001 呼叫分机 1005。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json

{
  "action": "NewCdr",
  "callid": "1547391782.318",
  "timestart": "2019-01-13 23:03:02",
  "callfrom": "1001",
  "callto": "1005",
  "callduration": "21",
  "talkduration": "10",
  "status": "ANSWERED",
  "type": "Internal",
  "recording": "20180315201242-1521173562.46-203-201-Internal.wav",
  "sn": "369362089179"
}
```

分机 1005 呼叫外部号码 9283737733。

```

POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
  "action": "NewCdr",
  "callid": "1547392142.330",
  "timestart": "2019-01-13 23:09:02",
  "callfrom": "1005",
  "callto": "9283737733",
  "desttrunkname": "SPS-6-36",
  "callduration": "6",
  "talkduration": "3",
  "status": "ANSWERED",
  "type": "Outbound",
  "recording": "20180315201242-1521173562.46-203-201-Internal.wav",

  "sn": "369362089179"
}

```

外部用户 1277378812 呼入分机 1005。

```

POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
  "action": "NewCdr",
  "callid": "1547392256.333",
  "timestart": "2019-01-13 23:10:56",
  "callfrom": "1277378812",
  "callto": "1005",
  "srctrunkname": "SPS-6-36",
  "callduration": "11",
  "talkduration": "4",
  "status": "ANSWERED",
  "type": "Inbound",
  "didnumber": "1005",
  "recording": "20180315201242-1521173562.46-203-201-Internal.wav",
}

```

```
}
  "sn": "369362089179"
}
```

‘来电呼叫请求’ 报告

在开启中继的“来电接听控制”的情况下，当来电通过该中继呼入时，PBX 会向应用服务器推送 INVITE 事件。

报告参数

名称	类型	描述
action	String	来电呼叫请求事件。
callid	String	该通通话的 id。
from	String	原始主叫号码。
to	String	原始被叫号码。
trunk	String	通过的中继名称。
inboundid	String	来电编码。
inbound	String	对象参数。
sn	String	PBX 的 SN 码。

报告示例

外部用户 1929383883 请求呼叫分机 1005。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json

{
  "action": "Invite",
  "callid": "1547389566.275",
  "call": [
    {
      "inbound": {
        "from": "1929383883",
        "to": "1005",
        "trunk": "SPS-6-36",
```

```
        "inboundid": "1547389566.275"
      }
    }
  ],
  "sn": "369362089179"
}
```

‘来电呼入’ 报告

在开启中继的“呼入目的地控制”情况下，当来电通过该中继呼入时，在 PBX 应答该来电后，会向应用服务器推送 INCOMING 事件。

报告参数

名称	类型	描述
action	String	来电呼入事件。
callid	String	该通通话的 id。
inbound	Object	对象参数。
inboundid	String	来电编号。
from	String	原始主叫号码。
to	String	原始被叫号码。
trunk	String	通过的中继名称。
sn	String	PBX 的 SN 码。

报告示例

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json

{
  "action": "Incoming",
  "callid": "1547393303.354",
  "call": [
    {
      "inbound": {
        "from": "182737733",
```

```
        "to": "1005",
        "trunk": "SPS-6-36",
        "inboundid": "1547393303.354"
    }
},
"sn": "369362089179"
}
```

‘播放语音结束’ 报告

PBX 给分机号码或外线号码播放提示音结束后，会发送报告。

报告参数

名称	类型	描述
action	String	播放提示音结束事件。
callid	String	该通通话的 id。
prompt	String	播放的语音文件名称。
ext	Int	分机对象。
extid	String	分机号码。
outbound inbound	String	对象参数。
trunkname	String	使用的中继。（给外线播放语音的报告中使用trunkname，其他类型的报告使用trunk。）
trunk	String	使用的中继。
ivr	String	对象参数。
ivrid	String	IVR 号码。
playtonum	String	播放语音给哪个号码。
sn	String	PBX 的 SN 码。

报告示例

[给分机播放语音文件](#) “record” 结束后的报告。

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
```

```

Content-Type: application/json
Accept: application/json

{
  "action": "PlayPromptEnd",
  "callid": "1547395290.362",
  "ext": {
    "extid": "1001"
  },
  "prompt": "record",
  "sn": "369362089179"
}

```

[给外线播放语音文件](#) “record” 结束后的报告。

```

POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
  "action": "PlayPromptEnd",
  "callid": "1547395463.366",
  "outbound": {
    "trunkname": "SPS-6-36"
  },
  "prompt": "record",
  "sn": "369362089179"
}

```

[通过IVR拨打分机号码](#)或分机拨打 IVR 号码，PBX 播放语音给分机。

```

POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
  "action": "PlayPromptEnd",
  "callid": "1547395676.370",
  "call": [
    {

```

```

        "ext": {
            "extid": "1005"
        }
    },
    "ivr": {
        "ivrid": "6500"
    },
    "playtonum": "1005",
    "sn": "369362089179"
}

```

[通过IVR拨打外线号码](#)或呼出转接到 IVR，PBX 播放语音给给外线号码。

```

POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

{
    "action": "PlayPromptEnd",
    "callid": "1547395803.374",
    "call": [
        {
            "outbound": {
                "trunk": "SPS-6-36"
            }
        }
    ],
    "ivr": {
        "ivrid": "6500"
    },
    "playtonum": "9282837733",
    "sn": "369362089179"
}

```

来电转接到 IVR 或来电呼入到 IVR，播放语音给来电号码。

```

POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260

Content-Type: application/json
Accept: application/json

```

```
{
  "action": "PlayPromptEnd",
  "callid": "1547395941.380",
  "call": [
    {
      "inbound": {
        "trunk": "SPS-6-36"
      }
    }
  ],
  "ivr": {
    "ivrid": "6500"
  },
  "playtonum": "81288833"
  "sn": "369362089179"
}
```

‘满意度调查结果’ 事件

进行满意度调查时，当收到用户的按键信息后，PBX 主动向应用服务器发送满意度调查的结果。

报告参数

名称	类型	描述
action	String	满意度调查事件。
surveyresult	String	满意度调查按键结果。
callid	String	该通通话的 id。
sn	String	PBX 的 SN 码。

报告示例

```
POST / HTTP/1.1
User-Agent: WebAPI
Host: 192.168.5.150: 8260
Content-Type: application/json
Accept: application/json

{
  "action": "satisfaction",
```

```
"surveyresult": "2",  
"callid": "1547396435.415",  
"sn": "369362089179"  
}
```

错误代码

错误代码	描述
10002	不支持 XML 数据封包格式
10003	API 不支持的请求号处理
10004	部分参数为空
10006	分机不存在
10007	通话不存在
10008	分机不处于空闲状态
10009	分机不允许被监听
10010	监听模式不匹配
10011	被叫号不存在
10012	IVR 不存在
10013	队列不存在
10014	响铃组不存在
10015	通话未桥接成功
10016	接受来电/拒绝来电超时
10017	配置分机失败
10018	配置 IVR 失败
10019	配置队列失败
10020	添加闹铃失败或者修改闹铃失败
10021	分机闹铃不存在
10022	分机闹铃已存在
10023	语音文件不存在
10024	分机没有呼出权限
10025	apply 太过频繁
10028	号码不能为空
10031	呼出号码不符合要求
10032	密码错误
10033	分机路由配置失败
20000	只支持 JSON 的格式

错误代码	描述
20001	只支持 POST 的请求
20002	用户登录失败（被锁）
20003	用户登录失败（无效的用户名或密码）
20004	无此类 Token
20005	Token 为空
20006	Token 超时
20007	数据为空
20008	端口错误
20009	请求号为空
20010	请求发送给 apisrv 处理失败
20011	HeartBeat 请求配置失败
20012	录音文件随机串无效
20013	验证码错误
20014	录音文件不存在
20015	allowedip 错误
20016	语音留言状态参数有误
20017	语音留言随机串无效
20018	语音文件不存在
20019	CDR 时间格式错误
20020	CDR 随机串无效
20021	CDR 不存在
30001	JSON 申请空间失败
30003	apisrv 进程未启动