API 文档

国威HB1930/HB1960

开发者文档

Table of Contents

1 API验证 5

1.1 登陆 5

1.1.1 Endpoint 5

1.1.2 请求参数 5

1.1.3 响应参数 5

1.1.4 实际示例 6

1.2 API 心跳包 7

1.2.1 Endpoint 7

1.2.2 请求参数 7

1.2.3 实际示例 7

1.3 退出登录 8

1.3.1 Endpoint 8

1.3.2 请求参数 8

1.3.3 实际示例 8

2 API接口示例 9

2.1 查询 PBX 信息 9

2.1.1 Endpoint 9

2.1.2 请求参数 9

2.1.3 响应参数 9

2.1.4 实际示例 9

2.2 分机 10

2.2.1 查询分机列表 10

2.2.2 查询分机设置 13

2.2.3 添加分机 18

2.2.4 编辑分机设置 22

2.2.5 删除分机 25

2.3 中继 26

2.3.1 查询中继列表 26

2.3.2 查询 SIP 中继 28

2.3.3 添加 SIP 中继 30

2.3.4 编辑 SIP 中继 32

2.3.5 删除 SIP 中继 34

2.4 呼入路由 35

2.4.1 查询呼入路由 35

2.4.2 添加呼入路由 37

2.4.3 Endpoint 39

2.4.4 删除呼入路由 40

2.5 呼出路由 41

2.5.1 查询呼出路由 41

2.5.2 添加呼出路由 43

2.5.3 编辑呼出路由 45

2.5.4 删除呼出路由 46

2.6 会议室 47

2.6.1 查询会议室详情 47

2.6.2 查询正在进行的会议 49

2.6.3 修改会议室 51

2.6.4 邀请会议室成员 52

2.6.5 踢除会议室成员 53

2.6.6 静音会议室成员 55

2.6.7 取消静音会议室成员 56

2.6.8 新建会议室 57

2.6.9 删除会议室 58

2.7 IVR 59

2.7.1 查询 IVR 设置 59

2.7.2 编辑 IVR 设置 61

2.8 队列 62

2.8.1 查询队列的设置 62

2.8.2 编辑队列设置 66

2.8.3 查询队列状态 69

2.8.4 签入队列 71

2.8.5 签出队列 72

2.8.6 暂停坐席服务 73

2.8.7 恢复坐席服务 74

2.9 广播组 75

2.9.1 查询广播组列表 75

2.9.2 查询广播组详情 76

2.9.3 添加广播组 78

2.9.4 编辑广播组 80

2.9.5 删除广播组 82

2.10 语音通话 83

2.10.1 查询来电 83

2.10.2 查询去电 85

2.10.3 分机互拨 86

2.10.4 拨打外线号码 87

2.10.5 挂断分机（强拆） 89

2.10.6 Endpoint 90

2.10.7 挂断去电（强拆） 91

2.10.8 通话保持 91

2.10.9 恢复通话 92

2.10.10 通话静音 93

2.10.11 取消静音 94

2.10.12 普通监听 95

2.10.13 密语监听 96

2.10.14 强插监听 98

2.10.15 通话转接 99

2.10.16 自动拨号（队列） 100

2.10.17 自动拨号（响铃组） 101

2.10.18 双向外呼 103

2.11 播放语音 104

2.11.1 给分机播放语音 104

2.11.2 给外线播放语音 105

2.12 语音留言 106

2.12.1 查询语音留言 106

2.12.2 删除语音留言 108

2.12.3 下载语音留言 108

2.13 下载录音文件 110

2.13.1 Endpoint 110

2.13.2 请求参数 110

2.13.3 响应参数 110

2.13.4 实际示例 110

2.13.5 全局录音下载链接 111

2.14 下载通话记录 111

2.14.1 Endpoint 111

2.14.2 请求参数 112

2.14.3 响应参数 112

2.14.4 实际示例 112

2.14.5 通话记录下载链接 113

2.15 酒店应用 113

2.15.1 添加闹铃 113

2.15.2 查询闹铃 115

2.15.3 修改闹铃 117

2.15.4 删除闹铃 118

2.15.5 酒店退房 120

3 API报告示例 122

3.1 系统启动报告 122

3.1.1 报告参数 122

3.1.2 报告示例 122

3.2 分机状态变更报告 122

3.2.1 分机状态 122

3.2.2 报告参数 122

3.2.3 报告示例 123

3.3 ‘响铃&回铃’报告 123

3.3.1 报告参数-回铃事件 123

3.3.2 报告参数-响铃事件 124

3.3.3 报告示例 124

3.4 ‘通话应答’报告 126

3.4.1 报告参数 126

3.4.2 报告示例 127

3.5 ’通话被应答’报告 127

3.5.1 报告参数 128

3.5.2 报告示例 128

3.6 ‘通话结束’报告 129

3.6.1 报告参数 129

3.6.2 报告示例 129

3.7 ‘呼叫转移’报告 131

3.7.1 报告参数 131

3.7.2 报告示例 132

3.8 ‘呼叫失败’报告 133

3.8.1 报告参数 133

3.8.2 报告示例 134

3.9 ‘通话记录’报告 134

3.9.1 报告参数 134

3.9.2 报告示例 134

# API验证

## 登陆

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。
* UC的HTTP API的端口号是9011。

### Endpoint

|  |
| --- |
| POST /api/v0.0.1/login?type={type} |

### 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： 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?type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.101.37  {  "username": "xiaoshao",  "password": "e10adc3949ba59abbe56e057f20f883e",  "port": "8260",  "version": "0.0.1",  "url": "172.16.6.150:8260",  "urltag": 1  } |

|  |
| --- |
| POST /api/v1.1.0/login?type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.101.37  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <username>api</username>  <password>e10adc3949ba59abbe56e057f20f883e</password>  <port>8260</port>  <version>0.0.1</version>  <url>172.16.6.150:8260</url>  <urltag>1</urltag>  </xml> |

**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": "813b621cfe8eecf445a2ce1f4a079ffe"  } |

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

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/v0.0.1/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172.16.6.150，端口号变更为8280，URL 变更为172.16.6.150:8280/report。

|  |
| --- |
| POST /api/v0.0.1/heartbeat?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1 HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "ipaddr": "172.16.6.150",  "port": "8280",  "url": "172.16.6.150:8280",  "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/v0.0.1/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/v0.0.1/logout?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1 HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

**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/v0.0.1/deviceinfo/query?token={token}&type={type} |

### 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。 |
| hardwarever | String | 硬件版本号。 |
| 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&type=json HTTP/1.1 HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

|  |
| --- |
| POST /api/v1.1.0/deviceinfo/query?token=277ac400357b509b4a587ff2157f7ad5&type=xml HTTP/1.1 HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

**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": "UC501",  "sn": "a09805021987",  "hardwarever": "3.1",  "firmwarever": "4.1.1",  "systemtime": "2020-05-19 04:06:07",  "uptime": "2020-Mar-23-08:33:22",  "extensionstatus": "21/300"  }  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <deviceinfo>  <devicename>UC501</devicename>  <sn>a09805021987</sn>  <hardwarever>3.1</hardwarever>  <firmwarever>4.1.1</firmwarever>  <systemtime>2020-05-22 07:24:00</systemtime>  <uptime>2020-Mar-23-08:33:22</uptime>  <extensionstatus>21/300</extensionstatus>  </deviceinfo>  </xml> |

## 分机

### 查询分机列表

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/extensionlist/query?token={token}&type={type} |

#### 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。 * Up:  正在通话 * Busy：忙线。 * Hold：通话保持。 * Malfunction：故障。 * Idle：空闲。 * Fxsnoport |
| type | String | 分机类型。 |
| port | String | 分机端口，当分机为模拟分机时则显示该项。 |
| username | String | 用户名。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/extensionlist/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1 HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

|  |
| --- |
| POST /api/v0.0.1/extensionlist/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1 HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

**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": "100",  "status": "Unavailable",  "type": "pjsip",  "username": "100",  "agentid": ""  },  {  "extnumber": "101",  "status": "Registered",  "type": "pjsip",  "username": "xiaoshaokun",  "agentid": ""  },  {  "extnumber": "102",  "status": "Registered",  "type": "pjsip",  "username": "102",  "agentid": ""  }  ]  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <extlist>  <item id="0">  <extnumber>100</extnumber>  <status>Unavailable</status>  <type>pjsip</type>  <username>100</username>  <agentid></agentid>  </item>  <item id="1">  <extnumber>101</extnumber>  <status>Unavailable</status>  <type>pjsip</type>  <username>xiaoshaokun</username>  <agentid></agentid>  </item>  <item id="2">  <extnumber>102</extnumber>  <status>Unavailable</status>  <type>pjsip</type>  <username>102</username>  <agentid></agentid>  </item>  <item id="3">  <extnumber>103</extnumber>  <status>Unavailable</status>  <type>pjsip</type>  <username>103</username>  <agentid></agentid>  </item>    </extlist>  </xml> |

### 查询分机设置

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/extension/query?token={token}&type={type} |

#### 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。 * Fxsnoport |
| type | String | 分机类型。 |
| port | String | FXS 分机端口号。 |
| callerid | String | 来电显示号码。 |
| 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。 |
| moblie | 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。 |
| 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。 |
| enablemobile | String | 移动分机功能。   * on：开启。 * off：关闭。 |
| ringsimultaneous | String | 移动分机与分机同时响铃。   * on：开启。 * off：关闭。 |
| mobileprefix | String | 移动分机呼出前缀。 |
| allowbeingmonitored | String | 允许被监听。   * 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。 |
| 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。 |

#### 实际示例

**请求示例**

查询分机101。

|  |
| --- |
| POST /api/v1.1.0/extension/query?token=277ac400357b509b4a587ff2157f7ad5&type=json HTTP/1.1 HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "101"  } |

|  |
| --- |
| POST /api/v1.1.0/extension/query?token=277ac400357b509b4a587ff2157f7ad5&type=xml HTTP/1.1 HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="utf-8"?>  <xml>  <extid>101</extid>  </xml> |

查询所有分机。

|  |
| --- |
| POST /api/v0.0.1/extension/query?token=277ac400357b509b4a587ff2157f7ad5 HTTP/1.1 HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

**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": "101",  "username": "xiaoshaokun",  "type": "pjsip",  "noanswerforward": "off",  "busyforward": "off",  "status": "Unavailable",  "callerid": "858585",  "userpassword": "12345678qW@",  "registerpassword": "12345133333323123qW@",  "email": "897257036@qq.com",  "moblie": "18088861612",  "maxregistrations": "3",  "hasvoicemail": "on",  "vmsecret": "666666",  "enablevmtoemail": "on",  "language": "cn",  "allowbeingmonitored": "off",  "monitormode": "disable",  "maxduration": "10800",  "dnd": "off"  }  ]  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <extinfos>  <item id="0">  <extnumber>101</extnumber>  <username>xiaoshaokun</username>  <type>pjsip</type>  <noanswerforward>off</noanswerforward>  <busyforward>off</busyforward>  <status>Unavailable</status>  <callerid>888888</callerid>  <registerpassword>12345133333323123qW@</registerpassword>  <email>897257036@qq.com</email>  <moblie>18088861612</moblie>  <maxregistrations>3</maxregistrations>  <hasvoicemail>on</hasvoicemail>  <vmsecret>666666</vmsecret>  <userpassword>12345678qW@</userpassword>  <enablevmtoemail>on</enablevmtoemail>  <language>cn</language>  <allowbeingmonitored>off</allowbeingmonitored>  <monitormode>disable</monitormode>  <maxduration>10800</maxduration>  <dnd>off</dnd>  </item>  </extinfos>  </xml> |

### 添加分机

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/extension/add?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 是否必需 | 类型 | 描述 |
| number | 否 | Int | 分机号。 |
| username | 否 | String | 用户名。 |
| callerid | 否 | String | 分机来电显示名称。 |
| 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。 |
| 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。 |
| 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。 |
| busyforward | 否 | String | 忙时转移。   * on：开启。 * off：关闭。 |
| btransferto | 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。 |
| btransferext | 否 | String | 目的地为分机时的分机号。 |
| btransferprefix | 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。 |
| btransfernum | 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。 |
| enablemobile | 否 | String | 移动分机功能。   * on：开启 。 * off：关闭。 |
| ringsimultaneous | 否 | String | 移动分机和分机同时响铃。   * on：开启 。 * off：关闭。 |
| mobileprefix | 否 | String | 移动分机呼出前缀。 |
| allowbeingmonitored | 否 | String | 允许被监听。   * on：开启。 * off：关闭。 |
| monitormode | 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。 |
| ringtimeout | 否 | String | 响铃超时。   * 15 * 30 * 60 * 120 * 300 |
| maxduration | 否 | String | 最大通话时长。   * 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。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

添加分机2002。

|  |
| --- |
| POST /api/v0.0.1/extension/add?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"2002",  "username":"xiaoshao\_api2",  "callerid":"",  "registerpassword":"12345133333323123qW@",  "maxregistrations":"3",  "loginpassword":"12345678qW@",  "email":"897257036@qq.com",  "mobile":"18088861612",  "hasvoicemail":"on",  "vmsecret":"888888",  "enablevmtoemail":"on",  "noanswerforward":"off",  "ntransferto":"Extension",  "ntransferext":"6000",  "ntransferprefix":"",  "busyforward":"off",  "btransferto":"Extension",  "btransferext":"110",  "btransferprefix":"",  "btransfernum":"",  "enablemobile":"off",  "ringsimultaneous":"off",  "mobileprefix":"",  "allowbeingmonitored":"off",  "monitormode":"Disable",  "ringtimeout":"15",  "maxduration":"Follow System",  "dnd":"off"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 编辑分机设置

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/extension/update?token={token}&type={type} |

#### 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。 |
| 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。 |
| 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。 |
| 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。 |
| busyforward | 否 | String | 忙时转移。   * on：开启。 * off：关闭。 |
| btransferto | 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。 |
| btransferext | 否 | String | 目的地为分机时的分机号。 |
| btransferprefix | 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。 |
| btransfernum | 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。 |
| enablemobile | 否 | String | 移动分机功能。   * on：开启 。 * off：关闭。 |
| ringsimultaneous | 否 | String | 移动分机和分机同时响铃。   * on：开启 。 * off：关闭。 |
| mobileprefix | 否 | String | 移动分机呼出前缀。 |
| allowbeingmonitored | 否 | String | 允许被监听。   * on：开启。 * off：关闭。 |
| monitormode | 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。 |
| ringtimeout | 否 | String | 响铃超时。   * 15 * 30 * 60 * 120 * 300 |
| maxduration | 否 | String | 最大通话时长。   * 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。 |

#### 实际示例

**请求示例**

将分机101的用户名修改为Tony，其他信息不变。

|  |
| --- |
| POST /api/v0.0.1/extension/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "101",  "username": "Tony"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <extid>101</extid>  <username>xiaoshao</username>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 删除分机

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/extension/delete?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 分机号码。  **取值范围**：   * 单个分机号码：删除指定分机。   例如：“number”:“2002”   * 多个分机号码：删除多个分机。多个分机号码用半角逗号（,）隔开。   例如：“number”:“2002,1001” |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

删除分机2002。

|  |
| --- |
| POST /api/v0.0.1/extension/delete?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "number": "2002"  } |

#### 响应示例

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

## 中继

### 查询中继列表

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/trunklist/query?token={token}&type={type} |

#### 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。   * IAX：IAX 协议类型的 VoIP 中继。 * SIP：SIP 协议类型的 VoIP 中继。 * FXO：FXO 模拟中继。 |
| status | String | SIP/IAX 中继状态。   * Registering：注册中 * Failure：注册失败 * Registered：已注册 * Disable：禁用 * Unknown：未知 |
| port | String | 中继端口。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/trunklist/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

|  |
| --- |
| POST /api/v0.0.1/trunklist/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  {  "status": "Success",  "trunkname": [  {  "trunkname": "to-108-16",  "status": "Registered",  "type": "SIP"  },  {  "trunkname": "to-18-16-2",  "status": "Registered",  "type": "SIP"  }  ]  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <trunkname>  <item id="0">  <trunkname>to-18-16-2</trunkname>  <status>Registered</status>  <type>SIP</type>  </item>  </trunkname>  </xml> |

### 查询 SIP 中继

#### Endpoint

|  |
| --- |
| POST /api/v1.1.0/siptrunk/query?token={token}&type={type} |

#### 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。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/siptrunk/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "trunkname": "all"  } |

|  |
| --- |
| POST /api/v0.0.1/siptrunk/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <trunkname>all</trunkname>  </xml> |

**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": "1",  "trunkname": "to-108-16",  "tech": "pjsip",  "trunktype": "register",  "host": "172.16.108.16",  "port": "5060",  "domain": "172.16.108.16",  "username": "6100",  "authname": "6100",  "fromuser": "",  "password": "@123456wW"  },  {  "id": "2",  "trunkname": "to-18-16-2",  "tech": "pjsip",  "trunktype": "peer",  "host": "172.16.108.16",  "port": "5060",  "domain": "172.16.108.16"  }  ]  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <trunks>  <item id="0">  <id>2</id>  <trunkname>to-18-16-2</trunkname>  <tech>pjsip</tech>  <trunktype>peer</trunktype>  <host>172.16.108.16</host>  <port>5060</port>  <domain>172.16.108.16</domain>  </item>  </trunks>  </xml> |

### 添加 SIP 中继

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/siptrunk/add?token={token}&type={type} |

#### 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。 |
| 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。 |
| 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。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/siptrunk/add?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "trunkname": "to-108-16-test",  "trunktype": "register",  "host": "172.16.108.16",  "port": "5060",  "username":"6100",  "authname":"6100",  "password":"Rqzswb55"  } |

|  |
| --- |
| POST /api/v0.0.1/siptrunk/add?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <trunkname>to-108-16-test</trunkname>  <trunktype>register</trunktype>  <host>172.16.108.16</host>  <port>5060</port>  <domain>172.16.108.16</domain>  <username>6100</username>  <authname>6100</authname>  <password>Rqzswb55</password>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 编辑 SIP 中继

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/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。 |
| 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。 |
| 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。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/siptrunk/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "id":"3",  "trunkname": "to-108-16",  "trunktype": "register",  "host": "172.16.108.100",  "port": "5060",  "username":"6100",  "authname":"6100",  "password":"@123456wW"  } |

|  |
| --- |
| POST /api/v0.0.1/siptrunk/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <id>3</id>  <trunkname>to-108-16</trunkname>  <trunktype>register</trunktype>  <host>172.16.108.100</host>  <port>5060</port>  <domain>172.16.108.100</domain>  <username>6100</username>  <authname>6100</authname>  <password>@123456wW</password>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 删除 SIP 中继

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/siptrunk/delete?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 是否必需 | 类型 | 描述 |
| trunkname | 是 | String | SIP 中继的名称。   * **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/v0.0.1/siptrunk/delete?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "trunkname": "to-108-16"  } |

|  |
| --- |
| POST /api/v0.0.1/siptrunk/delete?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <trunkname>to-108-16</trunkname>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

## 呼入路由

### 查询呼入路由

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/inroute/query?token={token}&type={type} |

#### 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。 |
| desttype | String | 目的地类型。   * hangup：挂断 * extension：分机 * voicemail：语音邮箱 * ivr：IVR * ringgroup：响铃组 * queue：队列 * conference：会议室 * disa：DISA * callback：回拨 |
| dest | String | 具体目的地。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/inroute/query?token=277ac400357b509b4a587ff2157f7ad5&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "name": "all"  } |

|  |
| --- |
| POST /api/v0.0.1/inroute/query?token=277ac400357b509b4a587ff2157f7ad5&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <name>all</name>  </xml> |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <inroutes>  <item id="0">  <id></id>  <name>test</name>  <did></did>  <caller\_id></caller\_id>  <desttype>extension</desttype>  <dest>102</dest>  </item>  </inroutes>  </xml> |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  {  "status": "Success",  "inroutes": [  {  "id": "",  "name": "test",  "did": "",  "caller\_id": "",  "desttype": "extension",  "dest": "102"  }  ]  } |

### 添加呼入路由

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/inroute/add?token={token}&type={type} |

#### 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。 |
| desttype | 是 | String | 目的地类型。   * hangup：挂断 * extension：分机 * voicemail：语音邮箱 * ivr：IVR * ringgroup：响铃组 * queue：队列 * conference：会议室 * disa：DISA * callback：回拨 |
| dest | 是 | String | 具体目的地。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/inroute/add?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "name":"test2",  "did":"",  "caller\_id":"",  "desttype":"extension",  "dest":"110"  } |

|  |
| --- |
| POST /api/v0.0.1/inroute/add?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <name>test2</name>  <did></did>  <caller\_id></caller\_id>  <desttype>extension</desttype>  <dest>110</dest>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### Endpoint

|  |
| --- |
| POST /api/v0.0.1/inroute/update?token={token}&type={type} |

#### 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。 |
| desttype | 是 | String | 目的地类型。   * hangup：挂断 * extension：分机 * voicemail：语音邮箱 * ivr：IVR * ringgroup：响铃组 * queue：队列 * conference：会议室 * disa：DISA * callback：回拨 |
| dest | 是 | String | 具体目的地。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/inroute/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "name":"test2",  "did":"",  "caller\_id":"",  "desttype":"extension",  "dest":"5000"  } |

|  |
| --- |
| POST /api/v0.0.1/inroute/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <name>test2</name>  <desttype>extension</desttype>  <dest>5000</dest>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 删除呼入路由

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/inroute/delete?token={token}&type={type} |

#### 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”   * **删除所有呼入路由**：设置 name 为 all。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/inroute/delete?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "name":"test2"  } |

|  |
| --- |
| POST /api/v0.0.1/inroute/delete?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <name>test2</name>  </xml> |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

## 呼出路由

### 查询呼出路由

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/outroute/query?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 是否必需 | 类型 | 描述 |
| name | 是 | String | 呼出路由的名称。   * **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 | 前缀删除成功的匹配。拨打的号码与此以及随后的列进行比较。在匹配之后，在将前缀发送到中继之前，将该前缀从所拨号码中移除。 |
| prepend | String | 数字预先匹配成功。如果拨打的号码与后续列指定的模式相匹配，则在发送到中继之前，这将被前置。 |
| matchpattern | String | 所拨号码将与前缀+此匹配模式进行比较。一旦匹配，所拨号码的匹配模式部分将被发送到中继。 |
| callerid | String | 如果提供了主叫方号码，被叫号码将只匹配前缀+匹配模式，如果发送的主叫方ID与此匹配。当分机拨打外部电话时，来电显示将是他们的分机号码，而不是他们的出局CID。以上特殊匹配序列可用于与其它号码匹配类似的来电号码匹配。 |
| trunks | String | 中继名称。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/outroute/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "name":"all"  } |

|  |
| --- |
| POST /api/v0.0.1/outroute/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <name>all</name>  </xml> |

**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": "19",  "name": "to-pbx",  "prepend": "",  "prefix": "16",  "matchpattern": "X.",  "callerid": "",  "trunks": "to-18-16-2"  }  ]  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <outroutes>  <item id="0">  <id>19</id>  <name>to-pbx</name>  <prepend></prepend>  <prefix>16</prefix>  <matchpattern>X.</matchpattern>  <callerid></callerid>  <trunks>to-18-16-2</trunks>  </item>  </outroutes>  </xml> |

### 添加呼出路由

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/outroute/add?token={token}&type={type} |

#### 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 | 否 | String | 前缀删除成功的匹配。拨打的号码与此以及随后的列进行比较。在匹配之后，在将前缀发送到中继之前，将该前缀从所拨号码中移除。 |
| matchpattern | 是 | String | 所拨号码将与前缀+此匹配模式进行比较。一旦匹配，所拨号码的匹配模式部分将被发送到中继。 |
| prepend | 否 | String | 数字预先匹配成功。如果拨打的号码与后续列指定的模式相匹配，则在发送到中继之前，这将被前置。 |
| callerid | 否 | String | 如果提供了主叫方号码，被叫号码将只匹配前缀+匹配模式，如果发送的主叫方ID与此匹配。当分机拨打外部电话时，来电显示将是他们的分机号码，而不是他们的出局CID。以上特殊匹配序列可用于与其它号码匹配类似的来电号码匹配。 |
| trunks | 是 | String | 中继名称。  多个中继之间用 , 隔开，例如 “trunks”:“trunk1,trunk2” |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/outroute/add?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "name":"to16",  "matchpattern":"X.",  "prefix":"16",  "trunks":"to-18-16-2"  } |

|  |
| --- |
| POST /api/v0.0.1/outroute/add?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <name>to16</name>  <matchpattern>X.</matchpattern>  <prefix>16</prefix>  <trunks>to-18-16-2</trunks>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 编辑呼出路由

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/outroute/update?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 是否必需 | 类型 | 描述 |
| id | 是 | Int | 呼出路由的标识码。 |
| name | 是 | String | 呼出路由的名称。 |
| prefix | 否 | String | 前缀删除成功的匹配。拨打的号码与此以及随后的列进行比较。在匹配之后，在将前缀发送到中继之前，将该前缀从所拨号码中移除。 |
| matchpattern | 是 | String | 所拨号码将与前缀+此匹配模式进行比较。一旦匹配，所拨号码的匹配模式部分将被发送到中继。 |
| prepend | 否 | String | 数字预先匹配成功。如果拨打的号码与后续列指定的模式相匹配，则在发送到中继之前，这将被前置。 |
| callerid | 否 | String | 如果提供了主叫方号码，被叫号码将只匹配前缀+匹配模式，如果发送的主叫方ID与此匹配。当分机拨打外部电话时，来电显示将是他们的分机号码，而不是他们的出局CID。以上特殊匹配序列可用于与其它号码匹配类似的来电号码匹配。 |
| trunks | 是 | String | 中继名称。  多个中继之间用 , 隔开，例如 “trunks”:“trunk1,trunk2” |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/outroute/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "id":"20",  "name":"to20",  "matchpattern":"X.",  "prefix":"20",  "trunks":"to-18-16-2"  } |

|  |
| --- |
| POST /api/v0.0.1/outroute/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <id>21</id>  <name>to20</name>  <matchpattern>X.</matchpattern>  <prefix>20</prefix>  <trunks>to-18-16-2</trunks>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 删除呼出路由

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/outroute/delete?token={token}&type={type} |

#### 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/v0.0.1/outroute/delete?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "name":"to16"  } |

|  |
| --- |
| POST /api/v0.0.1/outroute/delete?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <name>to20</name>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

## 会议室

### 查询会议室详情

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/conference/query?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 固定会议室号码。  **取值范围**：   * 单个会议室号码：查询指定会议室。   例如：“number”:“6500”   * 多个会议室号码：查询多个固定会议室。多个会议室号码用半角逗号（,）隔开。   例如：“number”:“6500,7500,8000”   * all：查询所有会议室。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |
| conferences | Array [<ConferenceInfo>](https://help.yeastar.com/zh-cn/s-series-developer-v2/api-v2/conference-query.html#conference-query__conferencesInfo) | 会议室信息列表。 |

ConferenceInfo

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| number | String | 会议室号码。 |
| name | String | 会议室名称。 |
| participantepassword | String | 与会者密码。 |
| moderatorpassword | String | 主持人密码。 |
| waitformoderator | String | 是否等候主持人进入后才开始会议。   * yes：是。 * no：否。 |

#### 示例

**请求示例**

查询固定会议室6400的配置详情。

|  |
| --- |
| POST /api/v0.0.1/conference/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number": "all"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success",  "conferences": [  {  "number": "6500",  "name": "6500",  "participantepassword": "111111",  "moderatorpassword": "222222",  "waitformoderator": "no"  },  {  "number": "7500",  "name": "7500",  "participantepassword": "",  "moderatorpassword": "",  "waitformoderator": "no"  },  {  "number": "8000",  "name": "testapi1",  "participantepassword": "444444",  "moderatorpassword": "555555",  "waitformoderator": "no"  }  ]  } |

### 查询正在进行的会议

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/conference/query\_in\_session?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 会话中的会议室号码。  **取值范围**：   * 单个会议室号码：查询指定的会话中的会议室。   例如：“number”:“8000”   * 多个会议室号码：查询多个会话中的会议室。多个会议室号码用半角逗号（,）隔开。   例如：“number”:“6500,8000” |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |
| conference | Array <ConferenceInfo> | 会议室信息列表。 |

ConferenceInfo

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| callid | String | 每个通话的唯一 ID。 |
| conferencenumber | Integer | 会议室号码。 |
| conferencename | String | 会议室名称。 |
| members | Array <MemberType> | 与会成员信息。 |

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| from | Integer | 主叫号码。 |
| to | String | 被叫号码。 |
| trunkname | String | 通话使用的中继名称。 |
| channelid | String | 通话中，每个成员的通话通道 ID。 |
| memberstatus | String | 会议室成员的状态。   * Up：该成员接受邀请，进入会议室。 * HOLD：该成员已进入会议室，等待主持人进入会议室。 |

#### 示例

**请求示例**

查询会话中的会议室8000的详细信息。

|  |
| --- |
| POST /api/v0.0.1/conference/query\_in\_session?token=813b621cfe8eecf445a2ce1f4a079ffe  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number": "8000"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success",  "conference": [  {  "conferencenumber": "8000",  "member": [  {  "number": "111",  "channelid": "PJSIP/111-00000003",  "memberstatus": "Up"  },  {  "number": "114",  "channelid": "PJSIP/114-00000004",  "memberstatus": "Up"  }  ]  }  ]  } |

### 修改会议室

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/conference/update?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 固定会议室号码。 |
| name | 否 | String | 固定会议室名称。注： 不允许输入以下字符：&'<>`|$]&quot;。 |
| participantepassword | 否 | String | 与会者密码。注： 只允许输入数字。 |
| moderatorpassword | 否 | String | 主持人密码。注： 只允许输入数字。 |
| waitformoderator | 否 | String | 是否等候主持人进入后才开始会议。  **取值范围**：   * yes：是。 * no：否。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

修改会议室8000。

|  |
| --- |
| POST /api/v0.0.1/conference/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 192.168.5.150    {  "number":"8000",  "participantepassword":"444444",  "moderatorpassword":"555555",  "waitformoderator":"yes"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 邀请会议室成员

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/conference/add\_member?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 会议室号码。表示邀请成员进入哪个会议室。 |
| invitemember | 是 | Integer | 要邀请的成员的号码。可邀请内部分机用户或外部用户。  **取值范围**：   * 单个号码：邀请单个成员。   例如：“invitemember”:“2023”   * 多个号码：邀请多个成员。多个号码用半角逗号（,）隔开。   例如：“invitemember”:“2023,2024” |
| retries | 否 | Integer | 重试邀请次数,默认不重试 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

邀请内部分机111加入会议室8000。

|  |
| --- |
| POST /api/v0.0.1/conference/add\_member?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"8000",  "invitemember":"111"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

### 踢除会议室成员

#### 请求地址

|  |
| --- |
| {base\_url}/{api\_path}/conference/kick\_member?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 会议室号码。表示要踢除哪个会议室的成员。 |
| membernumber | 是 | Integer | 成员号码。**取值范围**：   * 单个号码：踢除指定成员。   例如：“membernumber”:“111”   * 多个号码：踢除多个成员。多个号码用半角逗号（,）隔开。   例如：“membernumber”:“111,112”   * 固件4.3.4后增加使用"all"关键字直接剔除会议室所有成员 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**请求从会议室8000踢除分机111。

|  |
| --- |
| POST /api/v2.0.0/conference/kick\_member?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"8000",  "membernumber":"111"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 静音会议室成员

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/conference/mute\_member?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 会议室号码。 |
| membernumber | 是 | Integer | 成员号码。  **取值范围**：   * 单个号码：静音指定成员。   例如：“membernumber”:“111”   * 多个号码：静音多个成员。多个号码用半角逗号（,）隔开。   例如：“membernumber”:“111,112” |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

静音会议室8000的成员-分机111。

|  |
| --- |
| POST /api/v0.0.1/conference/mute\_member?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"8000",  "membernumber":"111"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 取消静音会议室成员

#### 请求地址

|  |
| --- |
| {base\_url}/{api\_path}/conference/unmute\_member?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 会议室号码。 |
| membernumber | 是 | Integer | 成员号码。  **取值范围**：   * 单个号码：取消静音指定成员。   例如：“membernumber”:“111”   * 多个号码：取消静音多个成员。多个号码用半角逗号（,）隔开。   例如：“membernumber”:“111,112” |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**取消静音会议室8000的成员-分机111。

|  |
| --- |
| POST /api/v0.0.1/conference/unmute\_member?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 192.168.5.150    {  "number":"8000",  "membernumber":"111"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 新建会议室

#### 固件4.3.4后增加

#### 请求地址

|  |
| --- |
| POST /api/{{version}}/conference/insert?token={{openvox\_token}} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 会议室号码。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |
|  |  |  |

#### 示例

**请求示例**

新增会议室20210713113

|  |
| --- |
| POST /api/v0.0.1/conference/insert?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "number":"20210713113"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

### 删除会议室

#### 固件4.3.4后增加

#### 请求地址

|  |
| --- |
| POST /api/{{version}}/conference/delete?token={{openvox\_token}} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 会议室号码。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |
|  |  |  |

#### 示例

**请求示例**

删除会议室20210713113

|  |
| --- |
| POST /api/v0.0.1/conference/delete?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "number":"20210713113"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

## IVR

### 查询 IVR 设置

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/ivr/query?token={token}&type={type} |

#### 请求参数

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

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| 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。 |

#### 实际示例

**请求示例**

查询 IVR 名称为test\_ivr。

|  |
| --- |
| POST /api/v0.0.1/ivr/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "name": "test\_ivr"  } |

|  |
| --- |
| POST /api/v0.0.1/ivr/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <name>test\_ivr</name>  </xml> |

查询所有IVR。

|  |
| --- |
| POST /api/v0.0.1/ivr/query?token=813b621cfe8eecf445a2ce1f4a079ffe 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": [  {  "ivrid": "4",  "promptrepeat": "3",  "responsetimeout": "5",  "ivrname": "test\_ivr",  "prompt": "haruhi3"  }  ]  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <ivr>  <item id="0">  <ivrid>4</ivrid>  <promptrepeat>3</promptrepeat>  <responsetimeout>5</responsetimeout>  <ivrname>test\_ivr</ivrname>  <prompt>haruhi3</prompt>  </item>  </ivr>  </xml> |

### 编辑 IVR 设置

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/ivr/update?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 是否必需 | 类型 | 描述 |
| ivrid | 是 | String | 要配置的 IVR 唯一标识（IVR 号码）。 |
| ivrname | 否 | String | IVR 名称。限制：   * 不允许输入! $ ( ) / # ; , [ ] " = < > & ' ` ^ % @ { } | 和空格。 * 最大31位。 * 不能为空。 |
| promptrepeat | 否 | String | 提示音播放次数  可选择的值：1、2、3、4、5 |
| responsetimeout | 否 | Int | 响应超时时间  可选择的值：1、2、3、4、5、6、7、8、9、10 |

#### 实际示例

**请求示例**修改 IVR 4 的 IVR 号码为test\_ivr，promptrepeat为3，responsetimeout为5。

|  |
| --- |
| POST /api/v1.1.0/ivr/update?token=1e3b3ebb6a974cb42ed31de5413df52d&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 192.168.5.150  {  "ivrid": "4",  "ivrname": "test\_ivr",  "promptrepeat":"3",  "responsetimeout":"5"  } |

|  |
| --- |
| POST /api/v1.1.0/ivr/update?token=1e3b3ebb6a974cb42ed31de5413df52d&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 192.168.5.150  <?xml version="1.0" encoding="utf-8"?>  <xml>  <ivrid>4</ivrid>  <ivrname>test\_ivr</ivrname>  <promptrepeat>3</promptrepeat>  <responsetimeout>5</responsetimeout>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

## 队列

### 查询队列的设置

#### Endpoint

|  |
| --- |
| POST /api/v1.1.0/queue/query?token={token}&type={type} |

#### 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 |
| 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。 |
| 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。 |
| announcepos | String | 公告当前位置。   * on：启用。 * off：关闭。 |
| announcefreq | String | 用户公告频率。 |
| announceholdtime | Int | 公告等待时间。 |

#### 实际示例

**请求示例**

查询队列9999，9998。

|  |
| --- |
| POST /api/v0.0.1/queue/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "queueid": "9999,9998"  } |

|  |
| --- |
| POST /api/v0.0.1/queue/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <queueid>9999,9998</queueid>  </xml> |

查询所有队列。

|  |
| --- |
| POST /api/v0.0.1/queue/query?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

**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": "9998",  "queuename": "queue\_xiaoshao",  "password": "666666",  "agents": "101",  "announcefreq": "300",  "announceholdtime": "off",  "announcepos": "off",  "joinempty": "on",  "leavewhenempty": "off",  "retry": "10",  "ringstrategy": "Ring All",  "agenttimeout": "20",  "wrapuptime": "10",  "dynamicagents": "6000",  "agentannounce": "",  "musiconhold": "Inherit",  "maxwaittime": "600"  },  {  "queuenumber": "9999",  "queuename": "queue\_test",  "password": "123456",  "agents": "101,103,104",  "announcefreq": "0",  "announceholdtime": "off",  "announcepos": "off",  "joinempty": "on",  "leavewhenempty": "off",  "retry": "5",  "ringstrategy": "Ring All",  "agenttimeout": "15",  "wrapuptime": "60",  "dynamicagents": "",  "agentannounce": "",  "musiconhold": "Inherit",  "maxwaittime": "5"  }  ]  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <queues>  <item id="0">  <queuenumber>9998</queuenumber>  <queuename>queue\_xiaoshao</queuename>  <password>666666</password>  <agents>101</agents>  <announcefreq>300</announcefreq>  <announceholdtime>off</announceholdtime>  <announcepos>off</announcepos>  <joinempty>on</joinempty>  <leavewhenempty>off</leavewhenempty>  <retry>10</retry>  <ringstrategy>Ring All</ringstrategy>  <agenttimeout>20</agenttimeout>  <wrapuptime>10</wrapuptime>  <dynamicagents>6000</dynamicagents>  <agentannounce></agentannounce>  <musiconhold>Inherit</musiconhold>  <maxwaittime>600</maxwaittime>  </item>  <item id="1">  <queuenumber>9999</queuenumber>  <queuename>queue\_test</queuename>  <password>123456</password>  <agents>101,103,104</agents>  <announcefreq>0</announcefreq>  <announceholdtime>off</announceholdtime>  <announcepos>off</announcepos>  <joinempty>on</joinempty>  <leavewhenempty>off</leavewhenempty>  <retry>5</retry>  <ringstrategy>Ring All</ringstrategy>  <agenttimeout>15</agenttimeout>  <wrapuptime>60</wrapuptime>  <dynamicagents></dynamicagents>  <agentannounce></agentannounce>  <musiconhold>Inherit</musiconhold>  <maxwaittime>5</maxwaittime>  </item>  </queues>  </xml> |

### 编辑队列设置

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/queue/update?token={token}&type={type} |

#### 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。 |
| 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 |
| 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 |
| 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。 |
| announcefreq | 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。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/queue/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "queueid":"9998",  "queuename":"queue\_xiaoshao",  "password":"666666",  "ringstrategy":"Ring All",  "agents":"101",  "dynamicagents":"6000",  "agenttimeout":"20",  "wrapuptime":"10",  "retry":"10",  "maxwaittime":"600",  "joinempty":"on",  "leavewhenempty":"off",  "announcepos":"off",  "announcefreq":"300",  "announceholdtime":"off"  } |

|  |
| --- |
| POST /api/v0.0.1/queue/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <queueid>9998</queueid>  <queuename>queue\_xiaoshao</queuename>  <password>666666</password>  <ringstrategy>Ring All</ringstrategy>  <agents>101</agents>  <dynamicagents>6000</dynamicagents>  <agenttimeout>20</agenttimeout>  <wrapuptime>10</wrapuptime>  <retry>10</retry>  <maxwaittime>600</maxwaittime>  <joinempty>on</joinempty>  <leavewhenempty>off</leavewhenempty>  <announcepos>off</announcepos>  <announcefreq>300</announcefreq>  <announceholdtime>off</announceholdtime>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 查询队列状态

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/queuestatus/query?token={token}&type={type} |

#### 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/v0.0.1/queuestatus/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "queueid":"9998"  } |

|  |
| --- |
| POST /api/v0.0.1/queuestatus/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <queueid>9998</queueid>  </xml> |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  {  "status": "Success",  "queue": {  "queuenumber": "9998",  "queuestatus": {  "members": [  {  "agent": "102",  "agentstatus": "1"  },  {  "agent": "101",  "agentstatus": "6"  },  {  "agent": "6000",  "agentstatus": "1"  }  ],  "callercount": 0  }  }  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  <queue>  <queuenumber>9998</queuenumber>  <queuestatus>  <members>  <item id="0">  <agent>102</agent>  <agentstatus>1</agentstatus>  </item>  <item id="1">  <agent>101</agent>  <agentstatus>5</agentstatus>  </item>  <item id="2">  <agent>6000</agent>  <agentstatus>1</agentstatus>  </item>  </members>  <callercount>0</callercount>  </queuestatus>  </queue>  </xml> |

### 签入队列

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/queue/add\_dynamicagent?token={token}&type={type} |

#### 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6000签入队列9998，队列的密码为666666。

|  |
| --- |
| POST /api/v0.0.1/queue/add\_dynamicagent?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "queueid": "9998",  "extid": "6000",  "password": "666666"  } |

|  |
| --- |
| POST /api/v0.0.1/queue/add\_dynamicagent?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <queueid>9998</queueid>  <extid>6000</extid>  <password>666666</password>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 签出队列

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/queue/del\_dynamicagent?token={token}&type={type} |

#### 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6000 签出队列9998；队列的密码为666666。

|  |
| --- |
| POST /api/v1.1.0/queue/del\_dynamicagent?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "queueid":"9998",  "extid":"6000",  "password":"666666"  } |

|  |
| --- |
| POST /api/v1.1.0/queue/del\_dynamicagent?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <queueid>9998</queueid>  <extid>6000</extid>  <password>666666</password>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 暂停坐席服务

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/queue/pause\_agent?token={token}&type={type} |

#### 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/v0.0.1/queue/pause\_agent?token=1e3b3ebb6a974cb42ed31de5413df52d&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "queueid": "9998",  "extid": "6000"  } |

|  |
| --- |
| POST /api/v0.0.1/queue/pause\_agent?token=1e3b3ebb6a974cb42ed31de5413df52d&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <queueid>9998</queueid>  <extid>6000</extid>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 恢复坐席服务

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/queue/unpause\_agent?token={token}&type={type} |

#### 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/v0.0.1/queue/unpause\_agent?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "queueid": "9998",  "extid": "6000"  } |

|  |
| --- |
| POST /api/v0.0.1/queue/unpause\_agent?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <queueid>9998</queueid>  <extid>6000</extid>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

## 广播组

### 查询广播组列表

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/paginggroup/lists?token={token}&type={type} |

#### 请求参数

无参数，直接发送查询广播组列表的请求即可。

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |
| paginggrouplist | Array <PaginggroupInfo> | 广播组信息列表。 |

PaginggroupInfo

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| number | String | 广播组号码。 |
| name | String | 广播组名称。 |

#### 示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/paginggrouplist/lists?token=277ac400357b509b4a587ff2157f7ad5  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Server: Boa/0.94.14rc21  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success",  "paginggrouplist": [  {  "number": "8891",  "name": "test\_api\_paging\_4"  }  ]  } |

### 查询广播组详情

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/paginggroup/query?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 广播组号码。  **取值范围**：   * 单个广播组号码：查询单个广播组。   例如：“number”:“8891”   * 多个广播组号码：查询多个广播组。多个广播组号码用半角逗号（,）隔开。   例如：“number”:“8891,8892”   * all：查询所有广播组。   例如：“number”:“all” |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |
| paginggroup | Array <PaginggroupInfo> | 广播组信息列表。 |

PaginggroupInfo支持两种广播组：单向传呼、双向对讲。

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| number | String | 广播组号码。 |
| name | String | 广播组名称。 |
| type | 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。 |
| member | String | 广播组成员，即分机号码。 |

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| number | String | 广播组号码。 |
| name | String | 广播组名称。 |
| type | String | 广播组类型。  intercom：双向对讲。广播组内的所有分机都可以进行通话并且都能听见彼此的声音。 |
| member | String | 广播组成员，即分机号码。 |

#### 示例

**请求示例**

查询广播组6300的详细配置。

|  |
| --- |
| POST /api/v0.0.1/paginggroup/query?token=277ac400357b509b4a587ff2157f7ad5&type=json  Content-Type:application/json; charset=utf-8  Host: 192.168.5.150    {  "number":"8891"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Server: Boa/0.94.14rc21  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success",  "paginggroup": [  {  "number": "8891",  "name": "test\_api\_paging\_4",  "type": "paging",  "member": "101,106,114"  }  ]  } |

### 添加广播组

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/paginggroup/add?token={token}&type={type} |

#### 请求参数

根据不同类型的广播组，需要发送不同的请求参数。

* 单向传呼
* 双向对讲

单向传呼

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 广播组号码。 |
| name | 是 | String | 广播组名称。 |
| type | 是 | String | 广播组类型。  **取值**：  paging：单向传呼。广播组内的分机只能听到发起者的声音。 |
| member | 是 | String | 广播组成员。  **取值范围**：选择分机为广播组成员。多个分机用半角逗号（,）隔开。  例如：“member”:“1000,1001,1002”。 |

双向对讲

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 广播组号码。 |
| name | 是 | String | 广播组名称。 |
| type | 是 | String | 广播组类型。  **取值**：  intercom：双向对讲。广播组内的所有分机都可以进行通话并且都能听见彼此的声音。 |
| member | 是 | String | 广播组成员。  **取值范围**：选择分机为广播组成员。多个分机用半角逗号（,）隔开。  例如：“member”:“1000,1001,1002”。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

添加广播组8891，广播组类型为 paging。

|  |
| --- |
| POST /api/v0.0.1/paginggroup/add?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"8891",  "name":"test\_api\_paging\_4",  "type":"paging",  "member":"106,114,101"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Server: Boa/0.94.14rc21  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 编辑广播组

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/paginggroup/update?token={token} |

#### 请求参数

根据不同类型的广播组，需要发送不同的请求参数。

* 单向传呼
* 双向对讲

单向传呼

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 广播组号码。 |
| name | 否 | String | 广播组名称。 |
| type | 是 | String | 广播组类型。  **取值**：  paging：单向传呼。广播组内的分机只能听到发起者的声音。 |
| member | 否 | String | 广播组成员。  **取值**：选择分机为广播组成员。多个分机用半角逗号（,）隔开。  例如：“member”:“1000,1001,1002”。 |

双向对讲

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 广播组号码。 |
| name | 否 | String | 广播组名称。 |
| type | 是 | String | 广播组类型。  **取值**：  intercom：双向对讲。广播组内的所有分机都可以进行通话并且都能听见彼此的声音。 |
| member | 否 | String | 广播组成员。  **取值**：选择分机为广播组成员。多个分机用半角逗号（,）隔开。  例如：“member”:“1000,1001,1002”。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/paginggroup/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"8891",  "name":"test\_api\_paging\_10",  "type":"intercom",  "member":"106,108"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Server: Boa/0.94.14rc21  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 删除广播组

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/paginggroup/delete?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 广播组号码。  **取值范围**：   * 单个广播组号码：删除指定广播组。   例如：“number”:“8891”   * 多个广播组号码：删除多个广播组。多个广播组号码用半角逗号（,）隔开。   例如：“number”:“8891,8892”   * all：删除所有广播组。   例如：“number”:“all” |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

删除广播组8891。

|  |
| --- |
| POST /api/v0.0.1/paginggroup/delete?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"8891"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Server: Boa/0.94.14rc21  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

## 语音通话

### 查询来电

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/inbound/query?token={token}&type={type} |

#### 请求参数

* 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 | 是 | 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。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/inbound/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "inboundid":"1589879501.901"  } |

|  |
| --- |
| POST /api/v0.0.1/inbound/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <inboundid>1589879501.901</inboundid>  </xml> |

**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": "1589881290.963",  "from": "103",  "to": "101",  "trunk": "to-18-16-2",  "callee": {  "extid": "101"  },  "status": "Talking"  }  ]  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  <inbound>  <inboundid>1589881290.963</inboundid>  <from>103</from>  <to>101</to>  <trunk>to-18-16-2</trunk>  <callee>  <extid>101</extid>  </callee>  <status>Talking</status>  </inbound>  </xml> |

### 查询去电

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/outbound/query?token={token}&type={type} |

#### 请求参数

* 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。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。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/outbound/query?token=1e3b3ebb6a974cb42ed31de5413df52d&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 192.168.5.150  {  "outboundid": "1495705009.316"  } |

|  |
| --- |
| POST /api/v0.0.1/outbound/query?token=1e3b3ebb6a974cb42ed31de5413df52d&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 192.168.5.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <outboundid>1495705009.316</outboundid>  </xml> |

**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": "1589881535.982",  "from": "66666",  "to": "103",  "trunk": "to-18-16-2",  "status": "Talking"  }  ]  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  <outbound>  <outboundid>1589881535.982</outboundid>  <from>66666</from>  <to>103</to>  <trunk>to-18-16-2</trunk>  <status>Talking</status>  </outbound>  </xml> |

### 分机互拨

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/extension/dial\_extension?token={token}&type={type} |

#### 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 |

#### 实际示例

**请求示例**分机 101呼叫分机 102。

|  |
| --- |
| POST /api/v0.0.1/extension/dial\_extension?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "caller": "101",  "callee": "102",  "autoanswer": "no"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/dial\_extension?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <caller>101</caller>  <callee>102</callee>  <autoanswer>no</autoanswer>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 拨打外线号码

#### 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/v0.0.1/extension/dial\_outbound?token={token}&type={type} |

#### 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）   * yes：是。 * no：否。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/extension/dial\_outbound?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid":"101",  "outto":"16103",  "autoanswer":"no"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/dial\_outbound?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <extid>101</extid>  <outto>16103</outto>  <autoanswer>no</autoanswer>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 挂断分机（强拆）

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/extension/hangup?token={token}&type={type} |

#### 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/v0.0.1/extension/hangup?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "101"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/hangup?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <extid>101</extid>  </xml> |

**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"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### Endpoint

|  |
| --- |
| POST /api/v0.0.1/inbound/hangup?token={token}&type={type} |

#### 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/v0.0.1/inbound/hangup?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "inboundid":"1589938168.1111"  } |

|  |
| --- |
| POST /api/v0.0.1/inbound/hangup?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <inboundid>1589938168.1111</inboundid>  </xml> |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 挂断去电（强拆）

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/outbound/hangup?token={token}&type={type} |

#### 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/v0.0.1/outbound/hangup?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "outboundid":"1589938378.1130"  } |

|  |
| --- |
| POST /api/v0.0.1/outbound/hangup?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <outboundid>1589938378.1130</outboundid>  </xml> |

**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

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/extension/hold?token={token}&type={type} |

#### 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/v0.0.1/extension/hold?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "101"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/hold?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <extid>101</extid>  </xml> |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <extid>101</extid>  </xml> |

### 恢复通话

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/extension/unhold?token={token}&type={type} |

#### 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/v0.0.1/extension/unhold?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "101"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/unhold?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <extid>101</extid>  </xml> |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 通话静音

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/extension/mute?token={token}&type={type} |

#### 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101的通话静音。

|  |
| --- |
| POST /api/v0.0.1/extension/mute?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "101"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/mute?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <extid>101</extid>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 取消静音

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/extension/unmute?token={token}&type={type} |

#### 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/v0.0.1/extension/unmute?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "101"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/unmute?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <extid>101</extid>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 普通监听

#### 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/v0.0.1/extension/listen?token={token}&type={type} |

#### 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101 监听分机106的通话。

|  |
| --- |
| POST /api/v0.0.1/extension/listen?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "listener": "101",  "listenedext": "106"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/listen?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <listener>101</listener>  <listenedext>106</listenedext>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 密语监听

#### 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/v0.0.1/extension/whisper?token={token}&type={type} |

#### 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=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "whisperer": "101",  "whisperedext": "106"  } |

|  |
| --- |
| POST /api/v1.1.0/extension/whisper?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <whisperer>101</whisperer>  <whisperedext>106</whisperedext>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 强插监听

#### 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/v0.0.1/extension/barge?token={token}&type={type} |

#### 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/v0.0.1/extension/barge?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "bargein": "101",  "bargedext": "106"  } |

|  |
| --- |
| POST /api/v0.0.1/extension/barge?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <bargein>101</bargein>  <bargedext>106</bargedext>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 通话转接

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/calltransfer?token={token}&type={type} |

#### 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。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/calltransfer?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "callid": "1589940297.1207",  "transferor":"101",  "transferto":"16103"  } |

|  |
| --- |
| POST /api/v0.0.1/calltransfer?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <callid>1589940297.1207</callid>  <transferor>101</transferor>  <transferto>16103</transferto>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 自动拨号（队列）

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/queue/dial\_outbound?token={token}&type={type} |

#### 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。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/queue/dial\_outbound?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "queueid": "9998",  "outto": "16103"  } |

|  |
| --- |
| POST /api/v0.0.1/queue/dial\_outbound?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <queueid>9998</queueid>  <outto>16103</outto>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 自动拨号（响铃组）

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/ringgroup/dial\_outbound?token={token}&type={type} |

#### 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。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/ringgroup/dial\_outbound?token=813b621cfe8eecf445a2ce1f4a079ffe\*type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 192.168.5.150  {  "ringgroupid": "600",  "outto": "16103"  } |

|  |
| --- |
| POST /api/v0.0.1/ringgroup/dial\_outbound?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 192.168.5.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <ringgroupid>600</ringgroupid>  <outto>16103</outto>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

### 双向外呼

#### 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

* PBX 至少有两条空闲可用的中继。

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/outbound/dial\_outbound?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 是否必需 | 类型 | 描述 |
| caller | 是 | String | 呼出的外线号码。 |
| outto | 是 | String | 呼出的外线号码。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/outbound/dial\_outbound?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "caller":"16101",  "outto":"16103"  } |

|  |
| --- |
| POST /api/v0.0.1/outbound/dial\_outbound?token=813b621cfe8eecf445a2ce1f4a079ffe&type=xml HTTP/1.1  Content-Type:application/xml; charset=utf-8  Host: 172.16.6.150  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <caller>16101</caller>  <outto>16103</outto>  </xml> |

**响应示例**

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

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="UTF-8" ?>  <xml>  <status>Success</status>  </xml> |

## 播放语音

### 给分机播放语音

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/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 | 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 System Recordings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/v0.0.1/extension/playprompt?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid":"101",  "prompt":"haruhi3",  "autoanswer":"no"  } |

播放多个语音文件。

|  |
| --- |
| POST /api/v0.0.1/extension/playprompt?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid":"101",  "prompt":"haruhi3+xiaoshao5",  "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"  } |

### 给外线播放语音

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/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 System Recordings页面。   多个语音文件之间用+ 连接。例如：“prompt:music1+music2” |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/outbound/playprompt?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "outto": "16103",  "prompt": "haruhi3+xiaoshao5"  } |

**响应示例**

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

## 语音留言

### 查询语音留言

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/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：“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

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。 |
| messagefrom | String | 语音留言是哪个号码发送的。 |
| 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102的语音留言。

|  |
| --- |
| POST /api/v0.0.1/voicemail/query?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "102"  } |

查询所有分机的语音留言。

|  |
| --- |
| POST /api/v0.0.1/voicemail/query?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150 |

**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": "102",  "voicemail": [  {  "voicemailfile": "msg0000.wav",  "messagefrom": "6000",  "voicemaildate": "2020-04-07 10:56:40",  "voicemaillength": "4"  },  {  "voicemailfile": "msg0001.wav",  "messagefrom": "7001",  "voicemaildate": "2020-04-07 10:57:31",  "voicemaillength": "3"  },  {  "voicemailfile": "msg0002.wav",  "messagefrom": "7001",  "voicemaildate": "2020-04-07 11:00:08",  "voicemaillength": "2"  }  ]  }  ]  } |

### 删除语音留言

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/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。 |

#### 实际示例

**请求示例**

|  |
| --- |
| POST /api/v0.0.1/voicemail/delete?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "102",  "voicemailfile": "msg0000.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"  } |

### 下载语音留言

#### Endpoint

|  |
| --- |
| POST /api/v0.0.1/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。 |
| 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。 |
| 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=1e3b3ebb6a974cb42ed31de5413df52d HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 192.168.5.150  {  "extid": "1004",  "voicemailfile": "msg0000.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",  "extid": "1004",  "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：

http://{pbx ip}:{https port}/api/v0.0.1/voicemail/download?extid=｛extid｝&voicemailstatus=｛voicemailstatus｝&token=｛token｝&voicemailfile={voicemailfile}&random=｛random｝

下面是语音留言下载链接示例：

<http://172.16.101.37:9011/api/v1.1.0/voicemail/download?extid=102&token=1733765a8f15fef25c90075c394e6b45&voicemailfile=msg0001.wav&random=385d9c3cdf0fd31b90887d1b5815151b>

## 下载录音文件

**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/v0.0.1/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/v0.0.1/recording/get\_random?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "recording":"exten-101-106-20200505-042135-1588641695.1772.wav",  "allowedip":"172.16.6.150"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  {  "status": "Success",  "recording": "exten-101-106-20200505-042135-1588641695.1772.wav",  "random": "fc19203cddc1d7846c2236f9eafe826b"  } |

### 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**：

http://{pbx ip}:{https port}/api/v1.1.0/recording/download?recording={recording}&random={random}&token={token}

**录音下载链接示例**：

<http://172.16.101.37:9011/api/v1.1.0/recording/download?recording=exten-101-106-20200505-042135-1588641695.1772.wav&random=1c574a38b96a0a94fe631208ce30ae53&token=1733765a8f15fef25c90075c394e6b45>

## 下载通话记录

**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。

### Endpoint

|  |
| --- |
| POST /api/v0.0.1/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。   * 所有分机：设置 extid 为 all，表示下载所有分机的通话记录。 * 单个分机：设置 extid 为分机号码。 * 多个分机：设置 extid 为分机号码，多个分机号码之间用 , 隔开。例如：“extid”: “1000,1001,1002” |
| starttime | 是 | String | 查询通话记录的起始时间。  时间格式：yyyy-mm-dd hh:mm:ss |
| endtime | 是 | String | 查询通话记录的结束时间。  时间格式：yyyy-mm-dd hh:mm:ss |
| allowedip | 否 | String | 设置允许下载通话记录的设备 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 | String | 分机。 |
| starttime | String | 查询通话记录的起始时间。 |
| endtime | String | 查询通话记录的结束时间。 |
| random | 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/v0.0.1/cdr/get\_random?token=813b621cfe8eecf445a2ce1f4a079ffe HTTP/1.1  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150  {  "extid": "all",  "starttime": "2020-05-18 00:00:00",  "endtime": "2020-05-20 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": "2020-05-01 00:00:00",  "endtime": "2020-05-05 23:59:59",  "random": "9ec0ef42229c68406363d21e8a82ca31"  } |

### 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**：

http://{pbx ip}:{https port}/api/v0.0.1/cdr/download?extid={extid}&starttime={starttime}&endtime={endtime}&token={token}&random={random}

**通话记录下载链接示例**：

http://172.16.101.37:9011/api/v0.0.1/cdr/download?extid=all&starttime=2020-05-01 00:00:00&endtime=2020-05-05 23:59:59&token=1733765a8f15fef25c90075c394e6b45&random=6bbae3a53045c07c53b3222132987156

## 酒店应用

### 添加闹铃

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/wakeupcall/create?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 分机号码。  **取值范围**：   * 单个分机号码：为指定分机添加闹铃。   例如：“number”:“110”   * 多个分机号码：为多个分机添加同一个闹铃。多个分机号码用半角逗号（,）隔开。   例如：“number”:“110,111” |
| time | 是 | String | 闹铃时间。 |
| type | 是 | String | 闹铃类型。  **取值范围**：   * onetime：一次。   当闹铃完成设置的重复次数之后，该条闹铃将被自动删除。   * custom：自定义。   type 为 custom 时，weekdays 不为空。 |
| weekdays | 否 | Integer | 闹铃适用的星期。  **取值范围**：0~6。其中，0表示周日，1-6表示周一到周六。注：  type 为 custom 时，此项不为空。 |
| prompt | 否 | String | 闹铃提示音。  此提示音必须为自定义提示音。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

为分机106添加闹铃。

|  |
| --- |
| POST /api/v0.0.1/wakeupcall/create?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"106",  "type":"custom",  "time":"15:55",  "weekdays":"0,1,2,3,5",  "prompt":"custom/test"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 查询闹铃

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/wakeupcall/query?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 分机号码。  **取值范围**：   * 单个分机号码：查询指定分机的闹铃。   例如：“number”:“114”   * 多个分机号码：查询多个分机的闹铃。多个分机号码用半角逗号（,）隔开。   例如：“number”:“110,114” |

#### 响应参数

注： 如果查询的分机没有闹铃信息，PBX 只响应请求成功或请求失败的参数。

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |
| wakeups | Array <WakeupInfo> | 闹铃列表。 |

WakeupInfo

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| number | String | 分机号码。 |
| wakeup | Array <ExtensionWakeup> | 分机闹铃信息列表。 |

ExtensionWakeup

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| wakeupid | Integer | 闹铃的唯一 ID。 |
| time | String | 闹铃时间。 |
| type | String | 闹铃类型。   * onetime：一次。 * custom：自定义。 |
| weekdays | Integer | type 为 custom 时，闹铃适用的星期。   * 0 * 1 * 2 * 3 * 4 * 5 * 6   其中，0表示周日，1-6表示周一到周六。 |
| prompt | String | 闹铃提示音。 |

#### 示例

**请求示例**

查询分机114的闹铃信息。

|  |
| --- |
| POST /api/v0.0.1/wakeupcall/query?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"114"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success",  "wakeups": [  {  "number": "114",  "wakeup": [  {  "wakeupid": "10",  "type": "custom",  "weekdays": "0,1,2,3,5",  "time": "15:55",  "prompt": "custom/test"  }  ]  }  ]  } |

### 修改闹铃

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/wakeupcall/update?token={token} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 分机号码。 |
| wakeupid | 是 | Integer | 闹铃的唯一 ID。 |
| time | 是 | String | 闹铃时间。 |
| type | 是 | String | 闹铃类型。  **取值范围**：   * onetime：一次。   当闹铃完成设置的重复次数之后，该条闹铃将被自动删除。   * custom：自定义。   type 为 custom 时，weekdays 不为空。 |
| weekdays | 否 | Integer | 闹铃适用的星期。  **取值范围**：0~6。其中，0表示周日，1-6表示周一到周六。注：  type 为 custom 时，此项不为空。 |
| prompt | 否 | String | 闹铃提示音。  此提示音必须为自定义提示音。 |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**更改分机106的闹铃设置。

|  |
| --- |
| POST /api/v0.0.1/wakeupcall/update?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "wakeupid":"9",  "number":"106",  "type":"custom",  "time":"15:55",  "weekdays":"0,1",  "prompt":"custom/test12311"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 删除闹铃

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/wakeupcall/delete?token={token}&type={type} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| wakeupid | 否 | Integer | 闹铃的唯一 ID。不带此参数，表示删除指定分机的所有闹铃。  **取值范围**：   * 单个闹铃 ID：删除指定分机的指定闹铃。   例如：“wakeupid”:“8”   * 多个闹铃 ID：删除指定分机的多个闹铃。多个闹铃 ID 用半角逗号（,）隔开。   例如：“wakeupid”: “8,9”   * all：删除指定分机的所有闹铃。   例如：“wakeupid”: “all” |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

删除分机106的闹铃。

|  |
| --- |
| POST /api/v2.0.0/wakeupcall/delete?token=1e3b3ebb6a974cb42ed31de5413df52d  Content-Type:application/json; charset=utf-8  Host: 192.168.5.150    {  "wakeupid":"106"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

### 酒店退房

#### 请求地址

|  |
| --- |
| POST /api/v0.0.1/hotel/checkout?token={token} |

#### 请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 是否必选 | 类型 | 描述 |
| number | 是 | String | 退房房间的分机号码。  **取值范围**：   * 单个分机号码：为指定分机所在房间退房。   例如：“number”:“111” |

#### 响应参数

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| status | String | 请求结果。   * Success：成功。 * Failed：失败。 |

#### 示例

**请求示例**

为分机111所在的房间退房。

|  |
| --- |
| POST /api/v0.0.1/hotel/checkout?token=813b621cfe8eecf445a2ce1f4a079ffe&type=json  Content-Type:application/json; charset=utf-8  Host: 172.16.6.150    {  "number":"111"  } |

**响应示例**

|  |
| --- |
| HTTP/1.1 200 OK  Server: Boa/0.94.14rc21  Accept-Ranges: bytes  Connection: close  Content-Type: application/json  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE    {  "status": "Success"  } |

|  |
| --- |
| HTTP/1.1 200 OK  Access-control-allow-origin: \*  Access-control-allow-methods: GET, POST, OPTIONS, PUT, DELETE  <?xml version="1.0" encoding="utf-8"?>  <xml>  <status>Success</status>  </xml> |

# API报告示例

## 系统启动报告

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: 172.16.6.150: 8260  Content-Type: application/json  Accept: application/json  {  "action": "BootUp",  "sn": "369351034049"  } |

## 分机状态变更报告

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 (InUse)**：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 InUse 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  Host:172.16.6.150  Content-Length:83  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"ExtensionStatus",  "extension":"101",  "status":"InUse",  "sn":"a09805021987"  } |

## ‘响铃&回铃’报告

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 101呼叫分机 106。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:89  Accept: application/json  Content-Type:application/json  Connection:close  {  "action":"ALERT",  "callid":"1589947952.1670",  "ext":{  "extid":[  "101"  ]  },  "sn":"a09805021987"  }    POST HTTP/1.1  Host:172.16.6.150  Content-Length:94  Accept: application/json  Content-Type:application/json  Connection:close  {  "action":"RING",  "callid":"1589947952.1670",  "ext":{  "extid":["106","101"]  },  "sn":"a09805021987"  } |

分机106 拨打外线号码 16103。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:67  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"ALERT",  "callid":"1589948179.1692",  "ext":{  "extid":"106"  }  }    POST HTTP/1.1  Host:172.16.6.150  Content-Length:155  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"RING",  "callid":"1589948178.1690",  "outbound":{  "from":"106",  "to":"103",  "trunk":"to-18-16-2",  "outboundid":"1589948178.1690"  },  "ext":{  "extid":"106"  }  } |

外线号码103呼叫分机101。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:152  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"ALERT",  "callid":"1589948512.1709",  "inbound":{  "from":"103",  "to":"101",  "trunk":"to-18-16-2",  "inboundid":"1589948512.1707"  },  "sn":"a09805021987"  }    POST HTTP/1.1  Host:172.16.6.150  Content-Length:173  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"RING",  "callid":"1589948512.1709",  "ext":{  "extid":"101"  },  "inbound":{  "from":"103",  "to":"101",  "trunk":"to-18-16-2",  "inboundid":"1589948512.1707"  },  "sn":"a09805021987"  } |

## ‘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101拨打分机106 106接听电话。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:96  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"ANSWER",  "callid":"1589953310.1724",  "ext":{  "extid":[  "106",  "101"  ]  },  "sn":"a09805021987"  } |

外部用户 103呼叫分机 101，分机 101接听来电。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:175  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"ANSWER",  "callid":"1589953417.1743",  "ext":{  "extid":"101"  },  "inbound":{  "from":"103",  "to":"101",  "trunk":"to-18-16-2",  "inboundid":"1589953417.1743"  },  "sn":"a09805021987"  } |

## ’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101呼叫分机106，分机106接听电话。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:96  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"ANSWER",  "callid":"1589953310.1724",  "ext":{  "extid":[  "101",  "106"  ]  },  "sn":"a09805021987"  } |

分机 106 呼叫外部号码 103，外部用户 103接听电话。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:177  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"ANSWER",  "callid":"1589953723.1760",  "ext":{  "extid":"106"  },  "outbound":{  "from":"106",  "to":"103",  "trunk":"to-18-16-2",  "outboundid":"1589953722.1758"  },  "sn":"a09805021987"  } |

## ‘通话结束’报告

### 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101与分机106通话中，分机 106挂断电话。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:93  Accept: application/json  Content-Type:application/json  Connection:close  {  "action":"BYE",  "callid":"1589953914.1775",  "ext":{  "extid":[  "106",  "101"  ]  },  "sn":"a09805021987"  }    POST HTTP/1.1  Host:172.16.6.150  Content-Length:93  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"BYE",  "callid":"1589953914.1777",  "ext":{  "extid":[  "101",  "106"  ]  },  "sn":"a09805021987"  } |

分机 106 与外部用户 103 通话中，外部用户 103 挂断电话。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:174  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"BYE",  "callid":"1589956488.1962",  "outbound":{  "from":"106",  "to":"103",  "trunk":"to-18-16-2",  "outboundid":"1589956488.1962"  },  "ext":{  "extid":"106"  },  "sn":"a09805021987"  }    POST HTTP/1.1  Host:172.16.6.150  Content-Length:174  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"BYE",  "callid":"1589956489.1964",  "ext":{  "extid":"106"  },  "outbound":{  "from":"106",  "to":"103",  "trunk":"to-18-16-2",  "outboundid":"1589956489.1964"  },  "sn":"a09805021987"  } |

## ‘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。

### 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 101 与分机 102 通话中，分机 101 将电话转移到另外一个号码。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:97  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"Tranfer",  "callid":"1589957670.2001",  "ext":{  "extid":[  "102",  "101"  ]  },  "sn":"a09805021987"  } |

外部用户 101 呼入PBX，与分机 102 通话，分机 102 将电话转移到另外一个号码。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:176  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"Tranfer",  "callid":"1589957892.2030",  "ext":{  "extid":"102"  },  "inbound":{  "from":"101",  "to":"102",  "trunk":"to-18-16-2",  "inboundid":"1589957892.2030"  },  "sn":"a09805021987"  } |

## ‘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：   * 480：被叫没有接听来电。 |
| 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  Host:172.16.6.150  Content-Length:113  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"CallFailed",  "reason":480,  "callid":"1589958470.2092",  "ext":{  "extid":[  "102",  "106"  ]  },  "sn":"a09805021987"  } |

## ‘通话记录’报告

### 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。 |
| callto | String | 被叫号码。 |
| callduraction | String | 通话时长。 |
| talkduraction | String | 接听时长。 |
| status | String | 通话状态。 |
| type | String | 通话类型。 |
| recording | String | 全局录音文件名。 |
| sn | String | PBX 的 SN 码。 |

### 报告示例

分机 106 呼叫分机 102。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:211  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"NewCdr",  "callid":"1589958682.2101",  "timestart":"2020-05-20 10:11:22",  "callfrom":"106",  "callto":"102",  "callduraction":"10",  "talkduraction":"5",  "status":"ANSWERED",  "type":"Internal",  "sn":"a09805021987"  } |

分机 106 呼叫外部号码 101。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:213  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"NewCdr",  "callid":"1589958787.2118",  "timestart":"2020-05-20 10:13:07",  "callfrom":"106",  "callto":"16101",  "callduraction":"10",  "talkduraction":"5",  "status":"ANSWERED",  "type":"Outbound",  "sn":"a09805021987"  } |

外部用户 101 呼入分机 102。

|  |
| --- |
| POST HTTP/1.1  Host:172.16.6.150  Content-Length:210  Accept: application/json  Content-Type:application/json  Connection:close    {  "action":"NewCdr",  "callid":"1589959139.2135",  "timestart":"2020-05-20 10:18:59",  "callfrom":"101",  "callto":"102",  "callduraction":"10",  "talkduraction":"4",  "status":"ANSWERED",  "type":"Inbound",  "sn":"a09805021987"  } |

深圳国威电子有限公司

2021-4-22