

# API 与酒店管理软件对接方案

## 方案背景：

1. 酒店对接需要的接口说明
2. PBX 要做的配置

## 具体方案：

如果您第一次使用朗视 API 接口，请先查看文档“如何启用 API 进行指令交互以及接收通话记录等 PBX 事件”

### 一、PBX 配置

#### 1. 开启 API 接口

- 1). 登录 PBX 网页，进入 配置→PBX→常规设置→二次开发接口。
- 2). 勾选 启用。
- 3). 设置用户名和密码，点击 保存 和 应用。



The screenshot shows the '配置' (Configuration) page for a PBX system. The '二次开发接口' (Secondary Development Interface) tab is selected. Under the '高级设置' (Advanced Settings) section, the '启用' (Enable) checkbox is checked. The '用户名' (Username) field is set to 'api' and the '密码' (Password) field is set to 'Soshiwub123'. There are '保存' (Save) and '应用' (Apply) buttons. Below the fields, there are tabs for '分机' (Extension) and '中继' (Trunk), with a '分机名称' (Extension Name) field.

**注：** 此处的用户名和密码是第三方应用通过 API 连接到 IPPBX 时所要用的用户名及密码（注意登录时，密码要 md5 加密）。

4). 选择是否监控分机的状态。当该分机状态有变更时（如：分机状态从忙变为空闲），API 会主动向第三方应用发送分机状态变更的事件报告。对接酒管，可以不需要状态监控



## 2. 配置呼出路由权限

International                      对应国际长途                      拨号模式为（注意 00 后面小数点） 00.

National                              对应国内长途                      拨号模式为（注意字母后面小数点） 0Z.

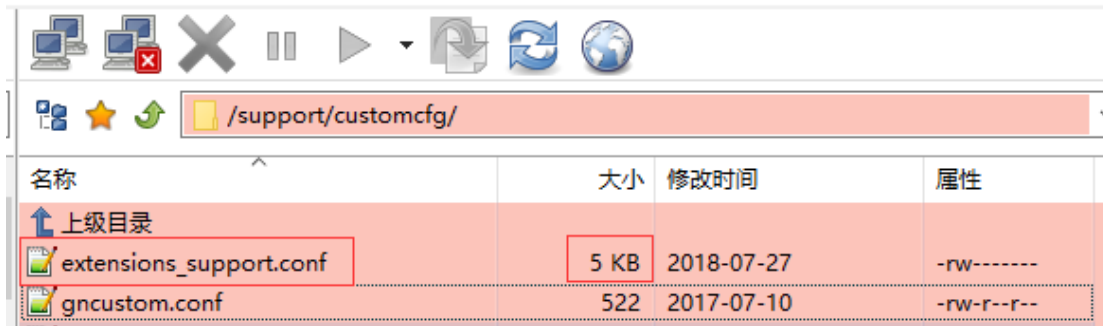
Local                                      对应本地市话                      拨号模式为（注意字母后面小数点） Z.



## 3. 定义清洁状态脚本和迷你吧

1) 可以直接使用附件 extensions\_support.conf 上传到 FTP 目录 /support/customcfg/ ，先删除原有的 extensions\_support.conf，再上传新的文件：

替换完成后大小如下图，然后再到 WEB 界面点下保存应用，即可使脚本生效。



2) 或者您也可以直接到后台编辑

/ysdisk/support//customcfg/extensions\_support.conf

在[Local\_Default\_CallingRules]

加入

```
include = clean
```

```
include = hotelba
```

```
[clean]
```

```
exten = 777, 1, NoOp(777)
```

```
same = n, Goto(clean-action, play, 1)
```

```
[clean-action]
```

```
exten = play, 1, Background(roomstatus)
```

```
same = n, WaitExten(4)
```

```
same = n, Goto(h, 1)
```

```
exten = *00, 1, Playback(room)
```

```
exten = *00, n, SayDigits(${CALLERID(num)}) //
```

```
exten = *00, n, UserEvent(DtmfEndString, Info:777${EXTEN})
```

```
exten = *00, n, Playback(uncleaned) //, 设置为未清洁, 再见,,
```

```
exten = *00, n, Hangup()
```

```
exten = *01, 1, Playback(room)
```

```
exten = *01, n, SayDigits(${CALLERID(num)})
```

```
exten = *01, n, UserEvent(DtmfEndString, Info:777${EXTEN})
exten = *01, n, Playback(cleaned) //, 设置为已清洁, 再见,,,
exten = *01, n, Hangup()
exten = h, 1, Hangup()
exten = i, 1, Goto(always-Hangup, h, 1)
```

迷你吧:

```
[hotelba]
exten = 778, 1, NoOp(778)
same = n, Background(product_id)
same = n, Read(digito,,9,,4) // 9 位商品编号
same = n, Gotoif($[${LEN(${digito})}>0]?check, 1)
same = n, Background(product_id)
same = n, Read(digito,,9,,4)
same = n, Gotoif($[${LEN(${digito})}>0]?check, 1)
same = n, Background(product_id)
same = n, Read(digito,,9,,4)
same = n, Gotoif($[${LEN(${digito})}>0]?check, 1)
same = n, Goto(h, 1) //三次没有输入商品编号, 则挂断
exten = check, 1, Macro(productlist, ${digito}, 1)
same = n, Goto(778, 2)
exten = h, 1, Hangup()
```

```
[macro-productlist]
exten = s, 1, Set(product_id=${ARG1})
same = n, Background(product_idc) //您输入的商品编码是
same = n, SayDigits(${product_id})
```

```
same = n, Background(product_num)
same = n, Read(digito,,2,,4) //2 位商品个数, 超
时 4 秒
same = n, Gotoif($[${LEN(${digito})}>0]?num, 1)
same = n, Background(product_num)
same = n, Read(digito,,2,,4)
same = n, Gotoif($[${LEN(${digito})}>0]?num, 1)
same = n, Background(product_num) //两次没有输入数量就取消返
回
exten = num, 1, Set(product_num=${digito})
same = n, Background(product_num) // 您输入的商品个数是, ,
same = n, SayDigits(${product_num}) //
same = n, Goto(confirm, 1)
exten = confirm, 1, Background(confirm)
same = n, Read(digito,,1,,4)
same = n, Gotoif($[${digito}=1]?yes, 1) //按 1 确认
same = n, Gotoif($[${digito}=*]?778, 1) //按*返回
same = n, Gotoif($[${digito}=9]?h, 1) //按 9 挂掉
same = n, Background(vm-goodbye)
exten = yes, 1, UserEvent(DtmfEndString, Info:
778*${product_id}*${product_num}) //发出 DTMF 串
exten = yes, 2, Background(success) //输入成功
exten = h, 1, hangup()
```

#### 4. 上传房态变更和迷你吧相关语音

语音格式要求 8K 16Bit 单声道 wav 格式, 可以使用合成语音,  
语音制作完成后(附件已包含以下语音, 解压后), 通过 FTP 上传到  
/ysapps/pbxcenter/var/lib/asterisk/sounds/zh/

文件名称	语音内容	所属模块
cleaned.wav	设置为已清洁, 再见	房态变更
uncleaned.wav	设置为未清洁, 再见	房态变更
roomstatus.wav	请输入房间清洁状态	房态变更
room.wav	房间	房态变更
confirm.wav	确认请按 1, 取消并返回请按*	迷你吧
playnum.wav	您输入的编号是	迷你吧
pro_name.wav	请输入商品编号和个数	迷你吧
product_id.wav	请输入商品编码	迷你吧
product_idc.wav	您输入的商品编码是	迷你吧
product_num.wav	请输入商品个数	迷你吧
product_numc.wav	您输入的商品个数是	迷你吧
success.wav	成功输入	迷你吧

## 二、对接说明:

1. 如何开启 API 以及通话记录等基本对接说明, 请查阅“如何启用 API 进行指令交互以及接收通话记录等 PBX 事件”文档

### 2. 话单输出 (计费用):

当一通通话结束后, PBX 向应用服务器实时发送通话详细记录 (CDR)。 报告示例

电话呼出话单, 标识 "type": "Outbound":

外部用手机 1277378812 呼入分机 1005, 推送内容如下:

POST / HTTP/1.1

User-Agent: WebAPI

Host: 192.168.5.122: 8260

Content-Type: application/json Accept: application/json Content-Length: 256

```
{
  "action": "NewCdr", "callid": "1547392256.333",
  "timestart": "2019-01-13 23:10:56",
  "callfrom": "1277378812",
  "callto": "1005",
  "srctrunkname": "SPS-6-36", "callduration": "11",
  "talkduration": "4", "status": "ANSWERED", "type": "Inbound",
  "didnumber": "1005",
  "recording": "20180315201242-1521173562.46-203-201-Internal.wav",
  "sn": "369362089179"
}
```

### 3. 叫醒（闹钟）

#### 1) 添加闹铃

通过本接口可以给酒店房间的分机设置闹铃。

注：

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

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

#### 请求地址

https://{pbx ip}:{https port}/api/{api version}/wakeupcall/create?token={token}

注：请求地址中的 {api version}，必须在 API 版本号前加小写字母 “v”，

如：v1.1.0

**请求参数**

名称	是否必需	类型	描述
extid	是	Int	分机号。
wakeup	是	Object	对象参数。
time	是	String	闹铃时间。
type	是	String	闹铃频率 <ul style="list-style-type: none"> <li>• onetime: 一次</li> <li>• everyday: 每天</li> <li>• custom: 自定义</li> </ul> 注：设置为 onetime 时，当闹铃完成设置的重复次数之后，该条闹铃将被自动删除。
weekdays	否	String	类型为自定义时的选项。 可选值：0, 1, 2, 3, 4, 5, 6 1-6 表示周一到周六，0: 为周日。

名称	是否必需	类型	描述
repeats	是	String	闹铃的重复次数 可选值：1, 2, 3
Repeatinterval	否	String	重复时间，设置好重复次数和重复时间后，话机响铃一次后，间隔所设置的重复时间后会再次响铃。 单位：分钟。



Prompt	否	String	闹铃提示音  此提示音必须为自定义提示音，不加字段 则用系统默认的 macroform-cold_day 提示 音。
--------	---	--------	--

#### 响应参数

名称	是否必需	类型	描述
status	是	String	请求状态： <ul style="list-style-type: none"> <li>• Success: 请求成功。</li> <li>• Failed: 请求失败。</li> </ul>
请求失败			
errno	否	String	错误代码。

#### 实际示例

给分机 1000 设置一个闹铃。

```

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

## 2) 查询闹铃

查询特定分机已添加的闹铃信息。

请求地址

```

https://{pbx ip}:{https port}/api/{api
version}/wakeupcall/query?token={token}
    
```

注：请求地址中的 {api version}， 必须在 API 版本号前加小写字母“v”，  
如：v1.1.0

请求参数：

名称	是否必需	类型	描述
extid	是	Int	分机号

响应参数

名称	是否必需	类型	描述
status	是	String	请求状态： <ul style="list-style-type: none"> <li>Success: 请求成功。</li> <li>Failed: 请求失败。</li> </ul>
请求成功			
extid	否	Int	分机号。
wakeup	否	Object	对象参数。
wakeupid	否	Int	闹铃的唯一标识。
time	否	String	闹铃时间。
type	否	String	闹铃频率 <ul style="list-style-type: none"> <li>onetime: 一次</li> <li>everyday: 每天</li> <li>custom: 自定义</li> </ul> 注：设置为 onetime 时，当闹铃完所设置的重复次数之后，该条闹铃将被自动删除。
weekdays	否	String	类型为自定义时的选项。 可选值：0, 1, 2, 3, 4, 5, 6 1-6 表示周一到周六，0: 为周

			日。
repeats	否	String	闹铃的重复次数 可选值：1，2，3
Repeatinterval	否	String	重复时间，设置好重复次数和重复时间后，话机响铃一次后，间隔所设置的重复时间后会再次响铃。 单位：分钟。
Prompt	否	String	闹铃提示音 此提示音必须为自定义提示音，不加字段则用系统默认的macroform-cold_day 提示音。
请求失败			
errno	否	String	错误代码。

请求查询分机 1000 中已添加的闹铃信息

```
{"extid": "1000"}
```

查询成功:

```
{"status": "Success", "wakeup": [{"extid": "1000", "wakeup": [{"wakeupid": "1", "time": "11:00", "type": "onetime", "prompt": "macroform-cold_day", "repeats": "3", "repeatinterval": "5"}]}]}
```

### 3) 更改闹铃设置

更改特定分机的闹铃设置。

请求地址

```
https://{pbx ip}:{https port}/api/{api version}/wakeupcall/update?token={token}
```

注：请求地址中的 {api version}，必须在 API 版本号前加小写字母 “v”，如：v1.1.0

## 请求参数

名称	是否必需	类型	描述
extid	是	Int	分机号。
wakeupid	是	Int	闹铃的唯一标识。
wakeup	是	Object	对象参数。
time	是	String	要修改的时间。
type	是	String	闹铃频率 <ul style="list-style-type: none"> <li>• onetime</li> <li>• everyday</li> <li>• custom</li> </ul> 注：设置为 onetime 时，当闹铃完所设置的重复次数之后，该条闹铃将被自动删除。
weekdays	否	String	类型为自定义时的选项。 可选值：0, 1, 2, 3, 4, 5, 6 1-6 表示周一到周六，0：为周日。
repeats	否	String	闹铃的重复次数 可选值：1, 2, 3

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

			则用系统默认的 macroform-cold_day 提示音。
--	--	--	---------------------------------

## 响应参数

名称	是否必需	类型	描述
status	是	String	请求状态： <ul style="list-style-type: none"> <li>• Success: 请求成功。</li> <li>• Failed: 请求失败。</li> </ul>
请求失败			
errno	否	String	错误代码。

## 请求示例

请求更改分机 1000 的闹铃设置。

```
{
  "extid": "1000",
  "wakeupid": "1", "wakeup": [
  {
    "time": "11:00",
    "type": "onetime",
    "repeats": "3",
    "repeatinterval": "5"
  }
  ]
}
```

#### 4) 删除闹铃

删除指定分机的闹铃。

请求地址

```
https://{pbx ip}:{https port}/api/{api
version}/wakeupcall/delete?token={token}
```

注：请求地址中的 {api version}，必须在 API 版本号前加小写字母“v”，如：v1.1.0。

### 请求参数

名称	是否必需	类型	描述
extid	是	Int	分机号。
wakeupid	是	String	闹铃的唯一标识。

### 响应参数

名称	是否必需	类型	描述
status	是	String	请求状态： <ul style="list-style-type: none"><li>Success: 请求成功。</li><li>Failed: 请求失败。</li></ul>
请求失败			
errno	否	String	错误代码。

### 请求示例

注：删除多个闹铃，用 , 隔开。如：{"extid": "1000", "wakeupid": "1,2"}。删除分机 1000 中，标识为 1 的闹铃，不带该参数则删除该分机所有闹铃

```
{  
  "extid": "1000",  
  "wakeupid": "1"  
}
```

#### 4. 房间（分机）名称修改和分机外呼权限控制

两者都是通过更改分机设置来实现的，具体接口指令是 extension/update

##### 1) 首先查询分机列表

查询 PBX 上分机列表的基本信息，如：分机名、分机号、分机状态、类型等。

请求地址 `https://{pbx ip}:https port/api/{api version}/extensionlist/query?token={token}`

注：请求地址中的 {api version}，必须在 API 版本号前加小写字母“v”，如：  
v1.1.0

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	是否必需	类型	描述
status	是	String	请求状态： <ul style="list-style-type: none"> <li>• Success: 请求成功。</li> <li>• Failed: 请求失败。</li> </ul>
请求成功			
extlist	否	Object	分机对象。
extnumber	否	Int	分机号。
status	否	String	分机当前状态： <ul style="list-style-type: none"> <li>• Unavailable: 不可使用。</li> <li>• Registered: 已注册。</li> <li>• Ringing: 响铃。</li> <li>• Busy: 忙线。</li> <li>• Hold: 通话保持。</li> <li>• Malfunction: 故障。</li> <li>• Idle: 空闲。</li> <li>• Fxsnoport</li> </ul>
type	否	String	分机类型。

port	否	String	分机端口，当分机为模拟分机时则显示该项。
username	否	String	用户名。
agentid	否	String	报工号时要播报的号码。此参数默认为空，表示播报分机号。
请求失败			
errno	否	String	错误代码。

实际示例 查询成功

```
{
  "status": "Success", "extlist": [
    {
      "extnumber": "1000", "status": "Registered", "type": "SIP",
      "username": "Jayson", "agentid": ""
    },
    {
      "extnumber": "1001", "status": "Unavailable", "type": "SIP",
      "username": "Erwin Co", "agentid": ""
    }
  ]
}
```

## 更改分机设置

更改分机的配置，如：分机号、分机名、同时注册数、邮箱、移动分机等。

请求地址

```
https://{pbx ip}:{https port}/api/{api
version}/extension/update?token={token}
```

注：请求地址中的{api version}，必须在 API 版本号前加小写字母“v”，



如：v1.1.0

### 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。
selectoutroute	否	String	选择分机的呼 ft 路由。从 unselectoutroute 中选择一个呼 ft 路由。selectoutroute 最多支持 4096 个字符。

### 响应参数

名称	是否必需	类型	描述
status	是	String	请求状态： <ul style="list-style-type: none"> <li>• Success: 请求成功。</li> <li>• Failed: 请求失败。</li> </ul>
请求失败			
errno	否	String	错误代码。

1) 名称修改，有的酒店需要每个客户入住时，把客户名称填入分机名称，这样当客人用酒店话机呼叫前台时，前台（IP 话机）可以显示客人的名称，方便进行回复。比如接起电话可以，可以说：“陈先生，您好，XXXXX”，可以增加客人的体验度。

### 请求地址

POST https://{pbx ip}:{https port}/api/{api

version}/extension/update?token={token}

#### 请求示例（配置 1002 分机国际、国内、本地呼出权限）

```
{ "extid": "1002", "username": "陈先生" }
```

2) 呼出权限控制，我们前面通过 PBX 配置了，呼出路由控制，International, National, Local 分别是国际长途，国内长途，本地市话，这里我们使用更改分机的配置信息，对单个分机修改路由权限（修改路由可以跟修改分机名称一起提交）。

#### 请求地址

POST https://{pbx ip}:{https port}/api/{api version}/extension/update?token={token}

#### 请求示例（配置 1002 分机国际、国内、本地呼出权限）

```
{ "extid": "1002", "selectoutroute": "International,National,Local" }
```

#### 请求示例（配置 1002 分机没有呼出权限）

```
{ "extid": "1002", "selectoutroute": "" }
```

### 5. 房间清洁状态修改, 话机通过拨打指定号码并操作按键信息, PBX 会把 ‘**按键信息**’ 推送给服务器

话机拨打 777, 听到语音, 拨\*01 设置为已清洁, 按键信息输出:

```
{"action": "DTMF", "callid": "1530768971.265", "ext": {"extid": "1002"}, "infos": "777*01" }
```

话机拨打 777, 听到语音, 拨\*00 设置为未清洁, 按键信息输出:

```
{"action": "DTMF", "callid": "1530768971.265", "ext": {"extid": "1002"}, "infos": "777*00" }
```

### 6. 退房, checkout 指令:

#### 酒店退房

通过本接口为酒店客人退房。

通过此接口退房之后, PBX 将会进行下面操作:

- 清空指定分机的语音信箱。

- 清空指定分机的呼出路由权限。
- 清空指定分机的闹钟信息。
- 重置指定分机的分机名称为分机号码。

请求地址

```
https://{pbx ip}:{https port}/api/{api version}/hotel/checkout?token={token}
```

注：请求地址中的 {api version}，必须在 API 版本号前加小写字母 “v”，如：v1.1.0

### 请求参数

注：为多个客人退房，房间号码之间使用, 隔开。如：

```
{"extid": "1001, 1002"}
```

名称	是否必需	类型	描述
extid	是	String	退房房间的分机号码。

### 响应参数

名称	是否必需	类型	描述
status	是	String	请求状态： <ul style="list-style-type: none"><li>• Success: 请求成功。</li><li>• Failed: 请求失败。</li></ul>

请求示例

```
{"extid": "1001"}
```

为房间分机号码为 4003 的客人退房。

响应示例，退房成功

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

三、附件：