

一、简介

1、IVR 简介

IVR(Interactive Voice Response)即互动式语音应答，是一种功能强大的电话自动服务系统。它用预先录制或TTS文本转语音技术合成的语音进行自动应答的系统，提供一种为客户进行菜单导航的功能。您只须用电话即可进入服务中心，可以根据操作提示收听相关产品信息，也可以根据用户输入的内容播放有关的信息。

云通讯平台IVR有两种接入方式：

- (1)外呼方式：通过调用云通讯平台提供的rest IVR外呼接口；
- (2)呼入方式：通过落地电话呼入云通讯平台提供的IVR接入号或者通过云通讯平台提供的VOIP呼入serviceid。

2、集成流程

- (1)注册成为云通讯平台用户。
- (2)登陆云通讯平台，进入管理控制台，创建自己的应用。
- (3)下载IVR demo布置到服务器后进行测试体验。

3、前期准备

- (1)注册成为云通讯的用户，详见 [新手指引](#)
- (2)登陆云通讯平台，进入管理控制台，创建自己的应用，IVR只能用创建应用进行测试，demo应用不支持IVR测试。在创建IVR类应用的时候，需填写应用名称和选择应用对应的行业，并且勾选"启用IVR"后在 "IVR通知消息URL" 中填写服务器的回调地址，如果需要用的文本转语音的业务就需要勾选"启用TTS"，不必选择"启用应用回调地址"、"服务器白名单"等选项。

IVR 通知消息 URL 填写注意：

1).URL 中如果没有加上端口则是默认端口 80；

2).云通讯平台只支持 http 的消息发送，如果填写 https 则无法回调；

The screenshot shows the '创建应用' (Create Application) page in the Yuntongxun console. The left sidebar has a blue header with the '应用' (Applications) menu item highlighted. Below it, the '创建应用' (Create Application) button is also highlighted. The main content area is titled '创建应用' and contains the following fields and options:

- * 应用名称** (Application Name): A text input field with the value 'myivr'. A note indicates it should not exceed 20 characters.
- 启用回调地址** (Enable Callback Address): A checkbox labeled '勾选启用' (Select to Enable).
- 启用IVR** (Enable IVR): A checkbox labeled '勾选启用 (收费功能)' (Select to Enable (Paid Feature)).
- IVR通知消息URL** (IVR Notification Message URL): A text input field with the value 'http://www.yuntongxun.com/ivr'. A note indicates that after enabling, the application will generate a 100 yuan/month function fee, starting from the go-live date, and multiple applications will be charged cumulatively.
- 坐席管理功能开关** (Seat Management Function Switch): A checkbox labeled '勾选启用 (坐席通知消息)' (Select to Enable (Seat Notification Message)).
- 启用TTS** (Enable TTS): A checkbox labeled '勾选启用 (收费功能)' (Select to Enable (Paid Feature)).
- 服务器白名单** (Server Whitelist): A checkbox labeled '勾选启用' (Select to Enable).
- * 应用行业** (Application Industry): A dropdown menu with the value '电子商务' (E-commerce).

At the bottom, there are two buttons: '确定' (Confirm) and '取消' (Cancel).

(3)应用创建成功后，在"应用列表"里面可以看见自己刚刚创建的应用，可以对该应用进行编辑和删除操作。当该应用集成完云通讯IVR能力后准备正式对外发布时，需要点击"上线"申请，此时应用才可以在云通讯的生产环境使用，否则只能在沙盒环境使用。



(4)如果需要上传IVR放音文件，在"应用列表"里面可以看见自己刚刚创建的应用，然后点创建的应用名称，在进入的页面点“语音库管理”后再点“上传新语音”即可，上传成功后就等待云通讯的审核人员进行审核了，可以在当前页面中的“语音库”中进行查看审核的状态，审核通过后才能在云通讯平台进行播放，注意单条语音最大为5M。



放音文件的格式样本如下：位速 128kbps，音频采样大小16位，频道 1(单声道)，音频采样级别 8 kHz，音频格式 PCM，这样能保证放音的清晰度。

4、核心概念

(1)生产环境：应用上线后的正式生产环境，对应的接入域名为：

<https://app.cloopen.com:8883>

(2)沙盒环境：应用开发测试时使用的调试环境，对应的接入域名为：

<https://sandboxapp.cloopen.com:8883>

(3)开发者主账号：每个开发者在云通讯平台对应一个主账号

(a)ACCOUNT SID：主账号id。

(b)AUTH TOKEN：主账号验证token。

(4)AppID：应用标识，每个创建的应用都对应唯一的id标识。

(5)IVR接入号：IVR呼入的号码，当应用上线后勾选“启用IVR”后由云通讯平台进行分配，一个接入号只能绑定一个应用，但是一个应用可以绑定多个接入号。需要在测试环境调试IVR呼入可以使用云通讯平台提供IVR测试接入号01082167088进行调试。

(6)serviceid：应用服务id，创建应用生成的绑定唯一id，用户VOIP的IVR接入号，等同于IVR接入号的作用，只能是绑定应用下面的子账号使用。暂时没有在官方网站提供，需要联系技术支持人员进行提供。

说明：每一个注册开发者对应一个主账号。

每一个主账号下可以创建多个App。

5、开发准备

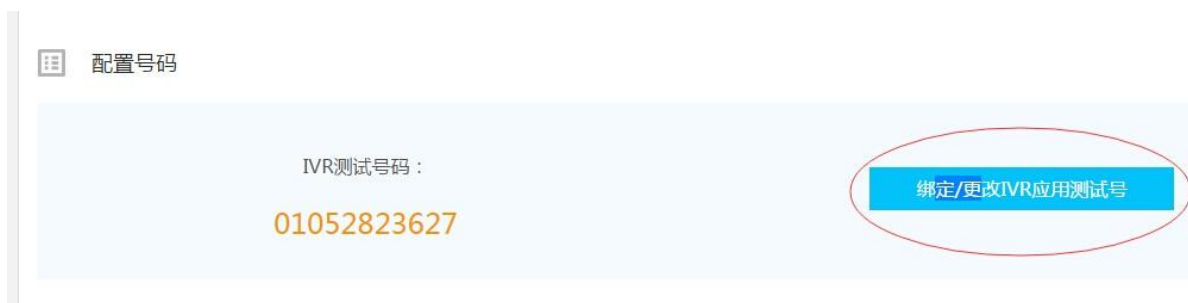
- (1) 解压缩有confIVR和confREST两个工程和confjoin.wav、confjoinfail.wav、confmenu.wav、createconf.wav、timeout.wav、timeoutbye.wav六个放音文件，以及一个IVR Demo开发文档；
- (2) confIVR 目录下面 src 文件夹为工程代码，WebRoot 文件夹下面是做服务端布置到自己的服务器环境中，如果布置在根目录下面，在创建应用的时候“IVR 通知消息 URL”请填写 http://ip:port/，如果是在根目录的目录下面，例如根目录的 confIVR 目录下面，“IVR 通知消息 URL”请填写 http://ip:port/confIVR/，如果需要用域名就把 ip 更换为域名；
- (3) 把解压的放音文件通过云通讯的官网上传进行审核；
- (4) confREST 工程需要做为客户端布置到服务器上，通过 CreateConf 进行创建会议，通过 InviteJoinConf邀请用户加入会议，通过 ConfMute 进行会议静音，通过 ConfUnMute 进行取消会议静音，通过 IvxDial 发起 IVR 外呼；
- (5)确定 IVR 使用的是外呼方式还是呼入方式：
 - a．外呼方式请使用 IvxDial 是做为客户端用来发起 rest 请求消息；
 - b．呼入方式则不需要 IvxDial，通过呼叫 IVR 测试号码 01082167088 即可，因为所有的用户都是用 01082167088 的 IVR 接入号进行调试，所以需要在创建的应用下面进行绑定主叫号码，绑定完成后呼入就会路由到对应的应用，否则呼入只能是云通讯平台的体验。需要注意的是，多个应用下面不能绑定相同的主叫号码，否则只有一个有效。绑定主叫的号码步聚：
 - 1).在“应用”列表中“号码管理”的“测试号码”页面中填写测试的号码，然后进行验证通过；



2).在“应用”列表中“应用管理”的“应用列表”页面中点击需要绑定主叫号码的应用名称；



3).在应用名称页面中找到“配置号码”后点击“IVR 测试号码”后面的“绑定/更改 IVR 应用测试号”，在弹出的页面中勾选需要绑定的主叫号码，完成后点击确定就完成配置。



二、开发指南

1.Demo 的 IVR 流程

- 1.通过 rest 接口创建会议，也可以用 3 和 4 来实现创建会议；
- 2.会议创建成功后通过 rest 接口邀请加入会议；
- 3.使用 IVR 外呼或者呼入方式后提示用户：按 1 创建会议，按 2 加入会议；
- 4.用户按 1 后创建会议，会议创建成功后提示创建的会议 id，播报完成后自动加入会议；
- 5.用户按 2 后提示用户输入 8 位会议 id；
- 6.用户输入超时后提示用户加入失败重新输入 8 位会议 id，用户再输入超时后放音提示挂机；
- 7.用户输入错误会议 id 后加入会议失败提示用户重新输入，只有输入正确后才能加入会议，否则需要用户自己挂机；
- 8.用户输入正确加入会议。

2.Demo 接口说明

(1) CCPRestSDK.java

发送 rest 创建会议接口

```
public static String CreateConf(String action,String maxmember,String passwd,String  
passwderrorurl,String dtmfreporturl,String delreporturl,String confduration,String  
autohangup,String confendprompt,String autorecord,String quiturl,String  
mediaopturl,String autojoin,String joinurl)
```

发送 rest 邀请加入会议接口

```
public static String InviteJoinConf(String confid,String number,String action,String role,String speak,String joinurl)
```

发送 rest 静音接口

```
public static String ConfMute(String callid,String confid,String action)
```

发送 rest 取消静音

```
public static String ConfUnMute(String callid,String confid,String action)
```

发送 rest IVR 外呼接口

```
public static String Dial(String number,String userdata,String record)
```

发送 rest 解散会议接口

```
public static String DismissConf(String action,String confid,String delurl,String autohangup,String quiturl,String recordurl)
```

发送 rest 退出会议接口

```
public static String QuitConf(String callid,String confid,String action)
```

发送 rest 会议放音接口

```
public static String ConfPlay(String confid,String filename,String loop,String action,String playcompleteurl)
```

发送 rest 停止会议放音接口

```
public static String ConfStopPlay(String confid,String action)
```

发送 rest 会议录音接口

```
public static String ConfRecord(String confid,String action,String time,String timeoutrecordurl)
```

发送 rest 停止会议录音接口

```
public static String ConfStopRecord(String confid,String action)
```

发送 rest 调节会议音量接口

```
public static String ConfVolumeAdjust(String confid,String adjusttype,String action)
```

发送 rest 会议成员禁听接口

```
public static String ConfMemberPause(String confid,String callid,String action)
```

发送 rest 取消会议成员禁听接口

```
public static String ConfMemberResume(String confid,String callid,String action,String speak)
```

发送 rest 会议计时闹钟接口

```
public static String ConfAlarmClock(String confid,String time,String action,String timeraction,String clockurl,String promptfile,String autohangup,String quiturl,String recordurl,String delurl)
```

发送 rest 会议状态查询接口

```
public static String QueryConfState(String confid,String action)
```

发送 https 请求消息接口

```
public static String sendPost(String url, String param)
```

(2) Contor.java

a.获取云通讯平台 IVR 通知消息 url 接口

```
String uri = req.getRequestURI();
```

b.获取云通讯平台 IVR 通知消息 url 中参数的接口

```
String appid = req.getParameter("appid");
```

c.IVR 启动业务接口

```
function startservice()
```

响应包体

```
<?xml version='1.0' encoding='UTF-8'?>
  <Response>
    <Get action='firstget' numdigits='1' timeout='30'>
      <Play>confmenu.wav</Play>
    </Get>
    <Play>timeout.wav</Play>
    <Redirect tag='1'>gettimeout</Redirect>
  </Response>
```

说明：主命令为按键命令并且嵌套 confmenu.wav 的放音提示，在用户按键一个就回调

通知到'firstget'的地址，如果用户没有按键就会 20 秒超时后执行副命令，跳转通知

gettimeout。

d.IVR 用户按键超时接口

function gettimeout()

响应包体

```
<?xml version='1.0' encoding='UTF-8'?>
    <Response>
        <Get action='firstget' numdigits='1' timeout='30'>
            <Play>confmenu.wav</Play>
        </Get>
        <Play>timeoutbye.wav</Play>
        <Hangup/>
    </Response>
```

说明：主命令为按键命令并且嵌套 confmenu.wav 的放音提示，在用户按键一个就回调通知到'firstget'的地址，如果用户没有按键就会 20 秒超时后执行副命令，直接挂机。

e.IVR 主菜单用户按键接口

function firstget()

如果用户按 1 的响应包体

```
<?xml version='1.0' encoding='UTF-8'?>
    <Response>
        <CreateConf action='createconfok' maxmember='10' autojoin='false'/>
    </Response>
```

响应创建会议命令，创建会议成功后会通知到 'createconfok'地址

如果用户按 2 的响应包体

```
<?xml version='1.0' encoding='UTF-8'?>
    <Response>
        <Get action='joinconfget' numdigits='8' timeout='30'>
            <Play>confjoin.wav</Play>
        </Get>
        <Play>timeout.wav</Play>
```

```
<Redirect tag='2'>gettimeout</Redirect>
</Response>
```

获取按键命令，接收用户输入的会议 ID，按键超时则跳转到 gettimeout

如果用户按其他键的响应包体

```
<?xml version='1.0' encoding='UTF-8'?>
  <Response>
    <Get action='firstget' numdigits='1' timeout='30'>
      <Play>confmenu.wav</Play>
    </Get>
    <Play>timeout.wav</Play>
    <Redirect tag='1'>gettimeout</Redirect>
  </Response>
```

f. 用户输入会议号后加入会议接口

function joinconfget()

响应包体

```
<?xml version='1.0' encoding='UTF-8'?>
  <Response>
    <JoinConf action='joinconfok' confid='$digits'/>
    <Redirect>joinconffail</Redirect>
  </Response>
```

说明：加入会议命令，加入会议成功会通知到 'joinconfok'

g. 创建会议成功的通知接口

function createconfok()

响应包体

```
<?xml version='1.0' encoding='UTF-8'?>
  <Response>
    <PlayMix type='40'>createconf.wav;$confid</PlayMix>
    <Redirect>autojoinconf</Redirect>
  </Response>
```

说明：播放混合音命令播放会议 ID，后跳转到 autojoinconf

h. 会议号播报完成接口

function autojoinconf()

响应包体

```
<?xml version='1.0' encoding='UTF-8'?>
  <Response>
    <JoinConf action='joinconfok' confid='$confid'/>
    <Redirect>joinconffail</Redirect>
  </Response>
```

说明：加入会议命令

i. 加入会议成功接口

function joinconfok()

响应包体

```
<Response>
  <CmdNone/>
</Response>
```

说明：不需要后续流程，响应空命令

j.IVR 加入会议失败

function joinconffail ()

响应包体

```
<?xml version='1.0' encoding='UTF-8'?>
  <Response>
    <Get action='joinconfget' numdigits='8' timeout='30'>
      <Play>confjoinfail.wav</Play>
    </Get>
    <Play>timeout.wav</Play>
    <Redirect tag='2'>gettimeout</Redirect>
  </Response>
```

说明：主命令为获取按键命令。

K .IVR 结束业务接口

```
function stopservice()
```

响应包体

```
<Response>
```

```
<CmdNone/>
```

```
</Response>
```

说明：主命令为空命令，因为结束业务了不需要返回其他的 IVR 命令。

3. 参考文档

(1) rest 会议管理接口文档

<http://docs.yuntongxun.com/index.php/%E4%BC%9A%E8%AE%AE%E7%AE%A1%E7%90%86>

(2)IVR 基础命令接口文档：

http://docs.yuntongxun.com/index.php/IVR_%E5%9F%BA%E7%A1%80%E5%91%BD%E4%BB%A4

(3)IVR 基础通知消息接口文档

http://docs.yuntongxun.com/index.php/IVR_%E5%9F%BA%E7%A1%80%E9%80%9A%E7%9F%A5%E6%B6%88%E6%81%AF

(4)IVR 会议命令接口文档

http://docs.yuntongxun.com/index.php/IVR_%E4%BC%9A%E8%AE%AE

(5) IVR 会议通知消息接口文档

http://docs.yuntongxun.com/index.php/IVR_%E4%BC%9A%E8%AE%AE%E9%80%9A%E7%9F%A5%E6%B6%88%E6%81%AF