ASR-MRCP 接口文档

此文档适用于使用 freeswitch v1.8.7 作为 mrcp 客户端，调用灵伴 mrcp 服务端的 asr 接口，完成语音识别。

IP 白名单说明：

- IP 白名单为全局白名单，填写后，能力开放平台内所有接口的调用，都需要从白名单中的 IP 进行发起。
- 若须调用 ASR-MRCP 接口，则 IP 白名单为必填。
- IP 白名单的添加、编辑在能力开放平台首页-企业信息中进行（paas.lingban.cn）。

1 配置 freeswitch 中的 mod_unimrcp 模块

安装 mod_unimrcp 模块

```
cd /项目源码地址/freeswitch

vim modules.conf
```

取消掉 asr_tts/mod_unimrcp 的注释

```
asr_tts/mod_unimrcp
```

安装 mod_unimrcp 模块

```
make mod_unimrcp-install
```

编辑/usr/local/freeswitch/conf/autoload_configs/modules.conf.xml，添加或者去掉注释 mod_unimrcp，让模块启动默认加载

```
vim /usr/local/freeswitch/conf/autoload_configs/modules.conf.xml

<load module="mod_unimrcp">
```

2 设置 profile 文件与 conf 文件

在 mrcp_profiles 目录下新建 lingban-asr-Normal-mrcp-v2.xml 配置文件：

```
vim /usr/local/freeswitch/conf/mrcp_profiles/lingban-asr-Normal-mrcp-v2.xml
```

然后输入以下内容：

```
<include>
    <!-- 后面使用该配置文件，均使用 name 作为唯一标识，而不是文件名 -->
        <profile name="lingban-asr-Normal" version="2">
        <!-- 灵伴 MRCP 服务器地址和端口号 -->
        <param name="server-ip" value="120.92.17.158"/>
        <param name="server-port" value="6060"/>
        <!-- 语音识别模型标识，不同的模型使用不同的 profile，支持 Normal、Bank 等 -->
        <param name="server-username" value="Normal"/>
        <param name="resource-location" value=""/>
        <!-- FreeSWITCH IP、端口以及 SIP 传输方式 -->
        <param name="client-ip" value="192.168.16.4" />
        <!-- 每个 profile 的 client 端口号唯一 -->
        <param name="client-port" value="5091"/>
        <param name="sip-transport" value="udp"/>
        <param name="speechsynth" value="speechsynthesizer"/>
        <param name="speechrecog" value="speechrecognizer"/>
        <!--param name="rtp-ext-ip" value="auto"/-->
        <param name="rtp-ip" value="192.168.16.4"/>
```

```
        <!-- 端口范围与/usr/local/freeswitch/conf/autoload_configs/sw
itch.conf.xml 中的 rtp-start-port rtp-end-port 不能重叠 -->

        <param name="rtp-port-min" value="4000"/>

        <param name="rtp-port-max" value="5000"/>

        <param name="codecs" value="PCMU PCMA L16/96/8000"/>

        <!-- Add any default MRCP params for SPEAK requests here -->

        <synthparams>

        </synthparams>

        <!-- Add any default MRCP params for RECOGNIZE requests here
 -->

        <recogparams>

        <!--param name="start-input-timers" value="false"/-->

        </recogparams>

   </profile>

</include>
```

接下来修改 unimrcp 默认使用的 ASR 驱动，vim/usr/local/freeswitch/conf/aut
oload_configs/unimrcp.conf.xml：default-asr-profile 为新创建的 lingba
n-asr：

```
<!-- UniMRCP profile to use for ASR -->

<param name="default-asr-profile" value="lingban-asr-Normal"/>
```

3 使用 lua 脚本集成灵伴 ASR-MRCP 接口

在/usr/local/freeswitch/scripts 目录下新增 lingban.lua 脚本：

```
local asr_model_id = argv[1];    --获取语音检测模型名
local _loopcnt = 0;
if (asr_model_id == nil) then
    asr_model_id = "Normal"
end
local calld_id = session:getVariable("uuid");
local action = 0;
local _loopcnt = 0;
session:answer()
session:sleep(200);
function onInput(s, type, obj)
    freeswitch.consoleLog("info", "Get callback with type: " .. typ
e .. "\n");
    if (type == "event") then
        local event = obj:getHeader("Speech-Type");
        freeswitch.consoleLog("info", "Get event: " .. event .. "\n
");
        if (event == "begin-speaking") then
            freeswitch.consoleLog("info", "Get begin-speaking-obj: "
 .. obj:serialize() .. "\n");
        end
```

```
       if (event == "detected-speech") then

          freeswitch.consoleLog("info", "Get detected-speech-obj: "
 .. obj:serialize() .. "\n");

          ret = obj:getHeader("ASR-Completion-Cause");

          if(ret ~= nil and (ret == "0" or ret == "2" or ret == "3
")) then

              action = 1;

          end

          if (obj:getBody()) then

              asr_res = obj:getBody()

              freeswitch.consoleLog("INFO","Get result: \n" .. asr_r
es .. "\n")  --- 输出语音检测结果

              if (string.find(asr_res, "00") ~= nil or string.find(a
sr_res, "001") ~= nil) then

                  action = 1

              end

          end

       end

   end

end

freeswitch.consoleLog("INFO","Enter Lua-script-name: " ..getFileNa
me(debug.getinfo(1).short_src).. "\n")  --- 输出语音检测结果



session:setVariable("fire_asr_events", "true");
```

```
-- Register the input callback

session:setInputCallback("onInput");

session:execute("detect_speech", "unimrcp:".."lingban-asr-"..asr_m
odel_id.." {recognition-timeout=60000,Sensitivity-Level=0.85,Start
-Input-Timers=true,no-input-timeout=10000,call-id=" .. calld_id:gs
ub("%-", "") .."}" ..

 "normal" .. " " .. asr_model_id);

while (session:ready() == true) do

       _loopcnt = _loopcnt + 1;

       if (_loopcnt > 300) then  -- about 1 minute;

              freeswitch.consoleLog("ERR", "session:"..calld_id..
" spend too long time!")

              break;

       end

       if (action == 1) then

       session:execute("detect_speech", "unimrcp:".."lingban-asr-
"..asr_model_id.." {recognition-timeout=60000,Sensitivity-Level=0.
85,Start-Input-Timers=true,no-input-timeout=10000,call-id=" .. cal
ld_id:gsub("%-", "") .."}" .. "normal" .. " " .. asr_model_id);

       end

       action = 0

       session:sleep(200)

end

session:execute("detect_speech", "stop");       --释放资源

session:hangup()
```

在/usr/local/freeswitch/grammar 目录新增 normal.gram 语法文件，可以为空语法文件须满足语音识别语法规范 1.0 标准（简称 SRGS1.0），该语法文件 ASR 引擎在进行识别时可以使用。

```
<?xml version="1.0" encoding="utf-8"?>

<grammar xmlns="http://www.w3.org/2001/06/grammar" xml:lang="en-US
" version="1.0" root="service">

    <rule id="service">

           <one-of>

              <item>

                 <ruleref uri="#voice-guide" />

              </item>

           </one-of>

    </rule>

    <rule id="domain">

           <one-of>

              <item>

                 <ruleref uri="#common" />

              </item>

           </one-of>

    </rule>

</grammar>
```

4 使用 python 脚本监听 DETECTED_SPEECH 事件并打印

```
apt-get install python-dev swig

pip install python-ESL

vim events.py

#!/usr/bin/env python


events.py - subscribe to DETECTED_SPEECH events and print them to s
tdout


import ESL

import logging


logging.basicConfig(filename='esl.log',level=logging.DEBUG)


#/usr/local/freeswitch</span>/conf/autoload_configs/event_socket.
conf.xml 中配置的 listen-ip、listen-port、password


con = ESL.ESLconnection('localhost', '8021', 'ClueCon')


if con.connected():

    con.events('plain', 'DETECTED_SPEECH')
```

```
    while 1:

        e = con.recvEvent()

            if e:

                print e.serialize()

                logging.info(e.serialize())
```

python events.py 启动脚本

5 拨打软电话

在/usr/local/freeswitch/conf/dialplan/default.xml 文件中创建拨号计划：

```
<extension name="unimrcp-asr">

    <condition field="destination_number" expression="^5001$">

        <action application="answer"/>

        <action application="lua" data="lingban.lua Normal"/>

    </condition>

</extension>
```

接下来用我们的软电话注册 freeswitch，然后拨打 5001 即可看到 events.py 打印的识别结果。

6 拨打外网交流

```
```

拨打远程，需要转码和忽略前期声音 originate {ignore_early_media=true,absolute_codec_string=PCMA}sofia/gateway/sip线路/电话 &lua(lingban.lua)