

讯飞语音+ 新手指南

Release1.0

2013 年 5 月 14 日

版权声明

版权所有 © 2013，安徽科大讯飞信息科技股份有限公司，保留所有权利。

商标说明

安徽科大讯飞信息科技股份有限公司的产品是安徽科大讯飞信息科技股份有限公司专有。在提及其他公司及其产品时将使用各自公司所拥有的商标，这种使用的目的仅限于引用。本文档可能涉及安徽科大讯飞信息科技股份有限公司的专利（或正在申请的专利）、商标、版权或其他知识产权，除非得到安徽科大讯飞信息科技股份有限公司的明确书面许可协议，本文档不授予使用这些专利（或正在申请的专利）、商标、版权或其他知识产权的任何许可协议。

不作保证声明

安徽科大讯飞信息科技股份有限公司不对此文档中的任何内容作任何明示或暗示的陈述或保证，而且不对特定目的的适销性及适用性或者任何间接、特殊或连带的损失承担任何责任。本手册内容若有变动，恕不另行通知。本手册例子中所用的公司、人名和数据若非特别声明，均属虚构。未得到安徽科大讯飞信息科技股份有限公司明确的书面许可，不得为任何目的、以任何形式或手段（电子的或机械的）复制或传播手册的任何部分。

保密声明

本文档（包括任何附件）包含的信息是保密信息。接收人了解其获得的本文档是保密的，除用于规定的目的外不得用于任何目的，也不得将本文档泄露给任何第三方。本软件产品受最终用户许可协议（EULA）中所述条款和条件的约束，该协议位于产品文档和/或软件产品的联机文档中，使用本产品，表明您已阅读并接受了EULA 的条款。

版权所有© 安徽科大讯飞信息科技股份有限公司
Copyrights © Anhui USTC iFLYTEK CO., LTD.

1 概述

“讯飞语音+”是科大讯飞公司向开发者提供的语音开发包。本文讲解“讯飞语音+”SDK 集成使用方法，默认读者已经掌握 Android 应用程序开发知识。

注：1、目前的版本支持语音合成、语音识别、语音理解，声纹识别和唤醒将在后续版本开放，敬请关注。

2、如果需要混淆代码，为了保证 sdk 的正常使用，需要在 `proguard.cfg` 加上下面三行配置：

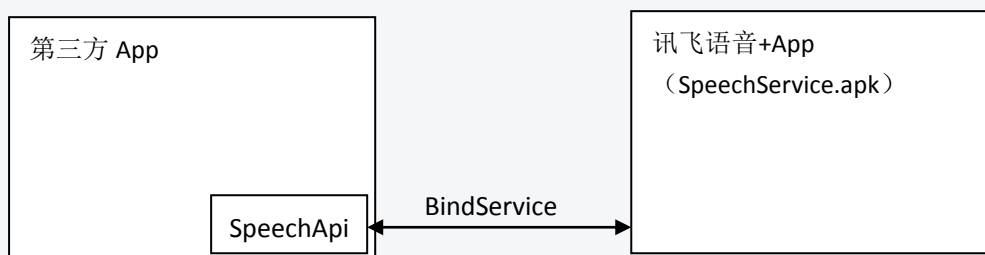
```
-dontwarn com.iflytek.speech.**  
-keepattributes Signature  
-keep class com.iflytek.speech.**{*;} 
```

2 Speech API 概述

应用程序若使用“讯飞语音+”提供的语音能力，需要将“讯飞语音+”开发包中 `SpeechApi.jar`（见 SDK 目录）集成到应用程序中去。

Speech API 接口的使用方法，请参考《Speech API 帮助手册》。

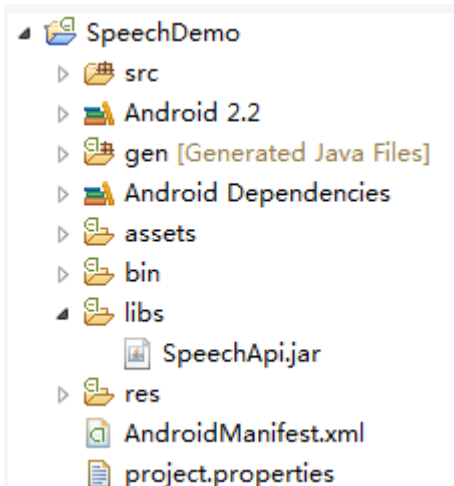
第三方应用程序集成使用“讯飞语音+”开发包逻辑关系图：



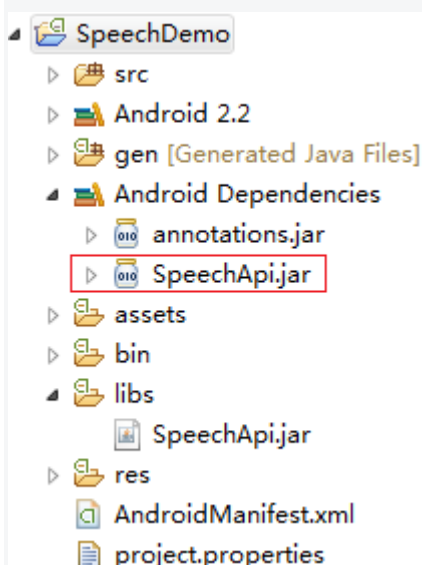
3 搭建开发环境

[1] 在 Eclipse 中建立你的 Android 工程。

[2] 将开发工具包中 `lib` 目录下的 `SpeechApi.jar` 复制到新建工程的 `libs` 目录中（如下图所示）。



[3] 右键单击工程，选择 Build Path 中的 Configure Build Path...，选中 Libraries 这个 tab，并通过 Add Jars... 导入工程 libs 目录下的 SpeechApi.jar 文件。（如下图所示）。



[4] 在你需要使用讯飞语音+服务的文件中导入相应的类。

```
import com.iflytek.speech.SpeechRecognizer;
```

4 在代码中使用开发工具包

[1] 语音识别

使用示例如下图所示。

```
// 检测是否安装了讯飞语音服务  
if (SpeechUtility.getUtility(this).queryAvailableEngines())
```

```
== null || SpeechUtility.getUtility(this).
    queryAvailableEngines().length<= 0)
{
    // 下载安装或者本地安装，请参照demo代码
}

// 设置申请的应用的appid
SpeechUtility.getUtility(this).setAppid("应用appid");

// 初始化识别对象
SpeechRecognizer mIat = new SpeechRecognizer(this, mInitListener);

// 转写会话
mIat.setParameter(SpeechConstant.PARAMS, "asr_ptt=1");
mIat.startListening(mRecognizerListener);

// 转写回话停止
mIat.stopListening(mRecognizerListener);

// 取消
mIat.cancel(mRecognizerListener);

// 转写回调
RecognizerListener mRecognizerListener = new RecognizerListener.Stub()
{
    @Override
    public void onVolumeChanged(int v) throws RemoteException {
        // 录音音量回调
    }

    @Override
    public void onResult(final RecognizerResult result, boolean isLast) throws
        RemoteException {
        // 结果回调
    }

    @Override
    public void onError(int errorCode) throws RemoteException {
        // 错误回调
    }

    @Override
    public void onEndOfSpeech() throws RemoteException {
```

```

        // 录音结束回调
    }

    @Override
    public void onBeginOfSpeech() throws RemoteException {
        // 录音启动回调
    }
};

```

※也可以调用我们定义的语音输入窗口，如下：

1) 通过以下两个 action 可以调用窗口

```
com.iflytek.speech.action.voiceinput
```

```
com.iflytek.speech.action.voiceisearch
```

2) 可以设置弹出窗口的两个按钮的名字

```
intent.putExtra("title_done", "确定");
```

```
intent.putExtra("title_cancel", "取消");
```

```

// 检测是否安装了讯飞语音服务
Intent intent = new Intent();
// 指定action名字
intent.setAction("com.iflytek.speech.action.voiceinput");
intent.putExtra(SpeechConstant.PARAMS, "asr_ptt=0");
intent.putExtra(SpeechConstant.VAD_EOS, "1000");
// 设置弹出框的两个按钮名称
intent.putExtra("title_done", "确定");
intent.putExtra("title_cancel", "取消");
startActivityForResult(intent, REQUEST_CODE_SEARCH);

protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(requestCode == REQUEST_CODE_SEARCH && resultCode == RESULT_OK)
    {
        // 取得识别的字符串
        ArrayList<String> results =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
        String res = results.get(0);
        EditText editor = ((EditText)findViewById(R.id.iat_text));
    }
}

```

```
String text = editor.getText().toString()+res;
editor.setText(text);
}
super.onActivityResult(requestCode, resultCode, data);
}
```

[2] 语音合成

使用示例如下图所示。

```
// 检测是否安装了讯飞语音服务
if (SpeechUtility.getUtility(this).queryAvailableEngines()
== null || SpeechUtility.getUtility(this).
queryAvailableEngines().length<= 0)
{
    // 下载安装或者本地安装，请参照demo代码
}

// 设置申请的应用的appid
SpeechUtility.getUtility(this).setAppid("应用appid");

// 初始化合成对象
SpeechSynthesizermTts = new SpeechSynthesizer(this, mTtsInitListener);

// 设置引擎类型
mTts.setParameter(SpeechConstant.ENGINE_TYPE, "local");
// 设置发音人
mTts.setParameter(SpeechSynthesizer.VOICE_NAME, "xiaoyan");
// 设置语速
mTts.setParameter(SpeechSynthesizer.SPEED, "50");
// 设置音调
mTts.setParameter(SpeechSynthesizer.PITCH, "50");

// 开始合成
int code = mTts.startSpeaking(text, mTtsListener);
// 停止
mTts.stopSpeaking(mTtsListener);
// 暂停播放
mTts.pauseSpeaking(mTtsListener);
// 恢复播放
mTts.resumeSpeaking(mTtsListener);

// 合成回调监听
```

```
SynthesizerListener mTtsListener = new SynthesizerListener.Stub() {  
    @Override  
    public void onBufferProgress(int progress) throws RemoteException {  
        // 缓冲进度回调  
    }  
  
    @Override  
    public void onCompleted(int code) throws RemoteException {  
        // 结束回调  
    }  
  
    @Override  
    public void onSpeakBegin() throws RemoteException {  
        // 开始播放回调  
    }  
  
    @Override  
    public void onSpeakPaused() throws RemoteException {  
        // 暂停回调  
    }  
  
    @Override  
    public void onSpeakProgress(int progress) throws RemoteException {  
        // 播放进度回调  
    }  
  
    @Override  
    public void onSpeakResumed() throws RemoteException {  
        // 重新播放回调  
    }  
};
```

5 附录

1. 识别结果说明

json 字段	英文全称	类型	说明
sn	sentence	number	第几句
ls	last sentence	boolean	是否最后一句
bg	begin	number	开始

ed	end	number	结束
ws	words	array	词
cw	chinese word	array	中文分词
w	word	string	单字
sc	socre	number	分数

转写结果示例：

```
{
  "sn": 1,
  "ls": true,
  "bg": 0,
  "ed": 0,
  "ws": [
    {
      "bg": 0,
      "cw": [
        {
          "w": "今",
          "sc": 0
        }
      ],
      "bg": 0,
      "cw": [
        {
          "w": "天",
          "sc": 0
        }
      ],
      "bg": 0,
      "cw": [
        {
          "w": "的",
          "sc": 0
        }
      ],
      "bg": 0,
      "cw": [
        {
          "w": "天",
          "sc": 0
        }
      ],
      "bg": 0,
      "cw": [
        {
          "w": "气",
          "sc": 0
        }
      ],
      "bg": 0,
      "cw": [
        {
          "w": "怎么样",
          "sc": 0
        }
      ],
      "bg": 0,
      "cw": [
        {
          "w": "。",
          "sc": 0
        }
      ]
    }
  ]
}
```

多候选结果示例：

```
{
  "sn": 1,
  "ls": false,
  "bg": 0,
  "ed": 0,
  "ws": [
    {
      "bg": 0,
      "cw": [
        {
          "w": "我想听",
          "sc": 0
        }
      ],
      "bg": 0,
      "cw": [
        {
          "w": "拉德斯基进行曲",
          "sc": 0
        },
        {
          "w": "拉得斯进行曲",
          "sc": 0
        }
      ]
    }
  ]
}
```

语法识别结果示例：

```
{
  "sn": 1,
  "ls": true,
  "bg": 0,
  "ed": 0,
  "ws": [
    {
      "bg": 0,
      "cw": [
        {
          "sc": "70",
          "gm": "0",
          "w": "北京到上海"
        },
        {
          "sc": "69",
          "gm": "0",
          "w": "天京到上海"
        },
        {
          "sc": "58",
          "gm": "0",
          "w": "东京到上海"
        }
      ]
    }
  ]
}
```

2. 语义结果说明

请参照《语义理解处理协议 v1 3200. pdf》