



# 安卓 SDK 使用文档

Release Date: 2016-07-19

## 目录

1 产品概述.....	3
1.1 简介.....	3
1.2 概念解释.....	3
1.3 功能介绍.....	3
1.4 应用场景.....	4
2 使用说明.....	4
2.1 适用环境.....	4
2.2 支持版本.....	4
2.3 兼容性.....	4
2.4 集成指南.....	5
2.4.1 添加 TRClient 到工程.....	5
2.4.2 权限声明.....	5
2.4.3 Progurad 配置.....	7
2.5 语音识别.....	7
2.5.1 实例化 VoiceRecognizerManager.....	7
2.5.2 设定 ASR 状态监听.....	8
2.5.3 调用语音识别方法.....	8
2.5.4 获取识别结果.....	8
2.6 语义理解.....	9
2.6.1 SDKInit 类.....	9
2.6.2 实例化 TuringApiManager 类.....	10
2.6.3 设置监听.....	11
2.6.4 发出请求.....	11
2.7 语音合成（TTS）.....	13
2.7.1 实例化 TTSManger 类.....	13
2.7.2 设置监听.....	13
2.7.3 开始语音合成.....	14
2.7.4 处理合成完成.....	14

---

3 常见问题.....	14
-------------	----

# 1 产品概述

## 1.1 简介

图灵机器人 SDK Android 版(以下简称 TRClient)是运行在 Android 平台的语音识别和语义理解一体化解决方案,以“JAR 包+动态链接库(TRC\_SDK\_1.0\libs\armeabi 下的 so 文件)”的形式发布。基于该方案,开发者可以轻松构建功能完备、交互性强的语音识别和语义理解应用程序,而且如果只使用语义理解引擎也是可以的。

## 1.2 概念解释

(1) **语音识别**: Automatic Speech Recognition (ASR), 也称自动语音识别,其目标是将人类的语音中的词汇内容转换相应的文本。

(2) **自然语言理解**: Natural Language Understanding (NLU), 俗称人机对话,是人工智能的分支学科。本学科通过电子计算机模拟人的语言交际过程,从而使计算机能理解和运用人类社会的自然语言,实现人机之间的自然语言通信,进而代替人的部分脑力劳动,包括查询资料、解答问题、摘录文献、汇编资料以及一切有关自然语言信息的加工处理。

(3) **TRClient**: TRClient 是一个封装了语音采集、处理、网络收发、语义理解等功能的语音识别和语义解析整体解决方案。

(4) **应用程序**: 在开发中使用了 TRClient, 具有语音识别功能的产品线产品。

## 1.3 功能介绍

(1) **语音识别**: 将语音识别成相应的文本。

(2) **语义理解**：将文本识别成领域相关的语义结果。

(3) **语音合成**：将文本转化成语音读出。

## 1.4 应用场景

(1) 产品需要流畅自然的中文聊天能力。我们提供精准的语义分析，可正确识别用户意图；支持多种上下文结构，满足连续对话和多重对话的需要。

(2) 产品需要自定义语义库。支持可自定义的 NLP 智能知识库系统，满足用户特殊语义处理需求。

(3) 产品需要广泛的知识库。支持上百个使用场景功能，以及丰富的百科知识。

# 2 使用说明

## 2.1 适用环境

Android2.2 及以上版本

## 2.2 支持版本

Android2.2 及以上版本

## 2.3 兼容性

(1) **语音识别**：Android2.2 版本以上均可支持。

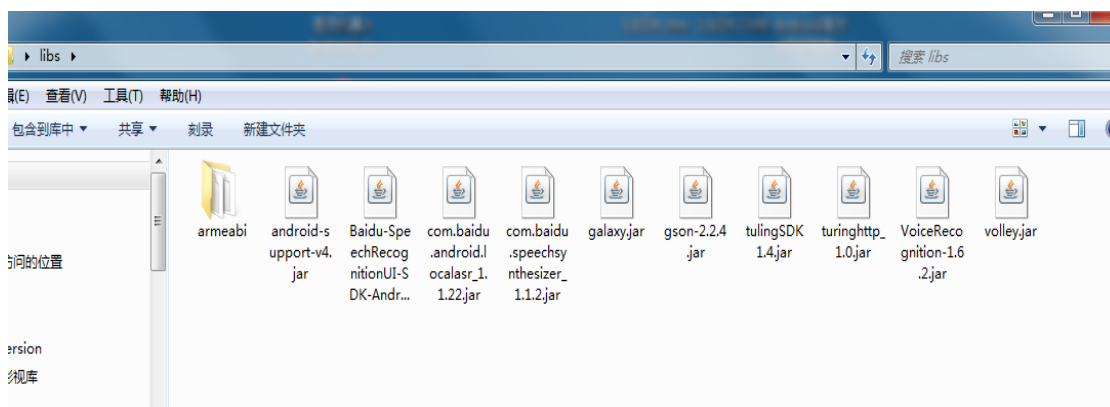
(2) **语义理解**：支持所有 JAVA 平台。

## 2.4 集成指南

### 2.4.1 添加 TRClient 到工程

#### 1. 添加 libs 到工程

开发者需要将 Demo 包中的 libs 目录整体 Copy 到工程目录，Libs 目录包括了语音识别合成所需的 so 库以及 jar 包



### 2.4.2 权限声明

名称	用途
android.permission.RECORD_AUDIO	允许使用麦克风录音
android.permission.INTERNET	允许联网，发送语音数据至服务器，获得识别结果
android.permission.ACCESS_NETWORK_STATE	允许获取当前网络状态，优化录音参数及网络参数
android.permission.READ_PHONE_STATE	允许获取用户手机的电话状态信息
android.permission.MODIFY_AUDIO_SETTINGS	允许蓝牙录音

android.permission.BROADCAST_STICKY	某些手机启动 SCO 音频连接需要此权限
android.permission.BLUETOOTH	允许蓝牙录音检测耳机状态
android.permission.WRITE_SETTINGS	允许修改和读取系统配置信息
android.permission.WRITE_EXTERNAL_STORAGE	允许向存储设备中写入

需要在 AndroidManifest.xml 文件，增加以上七个权限：

```

<uses-permission android:name="android.permission.RECORD_AUDIO"/>

<uses-permission

android:name="android.permission.ACCESS_NETWORK_STATE">

<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission

android:name="android.permission.READ_PHONE_STATE"/>

<uses-permission

android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<uses-permission android:name="android.permission.READ_CONTACTS"/>

<uses-permission

android:name="android.permission.ACCESS_WIFI_STATE"/>

```

如果需要使用蓝牙设备作为输入源，需要额外在 AndroidManifest.xml 文件添加下列权限：

```

<uses-permission

android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>

<uses-permission    android:name="android.permission.BROADCAST_STICKY

```

"/>

<uses-permission android:name="android.permission.BLUETOOTH "/>

### 2.4.3 Proguard 配置

如果应用配置了代码混淆，需要在 Proguard 配置文件增加以下参数：

-keep class com.baidu.android.\*\*{\*;}

-keep class com.baidu.voicerecognition.android.\*\*{\*;}

-keep class com.turing.androidsdk.\*\*{\*;}

注：在提供的 demo 中有详细的实现实例，可供参考

## 2.5 语音识别

语音识别部分主要包括以下四个步骤

### 2.5.1 实例化 VoiceRecognizerManager

实例化 VoiceRecognizerManager，由于目前我们支持百度 ASR 方式，不再支持讯飞

ASR 方式，暂时使用这个版本

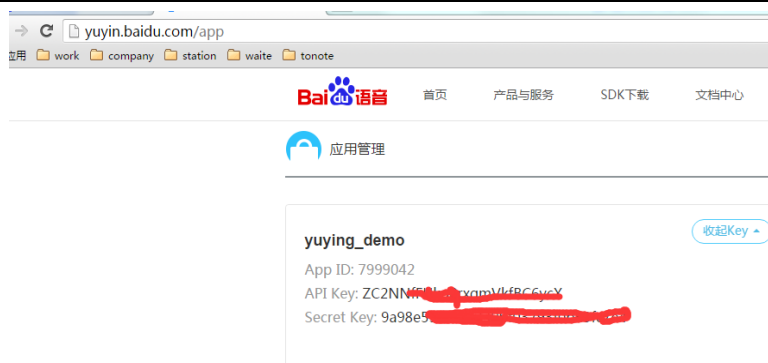
使用百度 ASR，请使用构造方法：

```
VoiceRecognizerManager(Context context,String bdAPI_KEY,String
bdSECRET_KEY)
```

参数: bdAPI\_KEY 和 bdSECRET\_KEY

这里的 bdAPI\_KEY 和 bdSECRET\_KEY，需要自己到百度官方去申请，位置如图所示：





注：需要去百度开放平台注册申请，百度语音开放平台的网址为：

<http://yuyin.baidu.com/>

## 2.5.2 设定 ASR 状态监听

设定 ASR 状态监听

示例：`VoiceRecognizerManager.setVoiceRecognizeListener(listener);`

这个 listener 就是需要实现接口 `VoiceRecognizeListener`，在不同的识别状态下回调其不同的方法。

## 2.5.3 调用语音识别方法

调用语音识别方法

示例：`VoiceRecognizerManager.startRecognize();`

这个方法就是开始进行识别

## 2.5.4 获取识别结果

获取识别结果 在步骤 2 里 listener 的回调方法 `onRecognizeResult` 中获取识别结果。

```
public void onRecognizeResult(String result)
```

```
{
```

```
        handleRecognizeResult(result);  
  
    }
```

这里的 result 就是返回的结果

## 2.6 语义理解

语义理解部分主要包括以下四个步骤

### 2.6.1 SDKInit 类

SDKInit 类初始化

调用 SDKInit.init

```
public static void init(SDKInitBuilder builder,InitListener initListener)
```

参数：

( 1 ) SDKInitBuilder 是封装了初始化的各种参数，这个类的变量说明

Context 为上下文

Secret 为官网上机器人详情页中自动生成的一个 secret，开启加密即可，如下图

turingKey 为该机器人账号的 apikey，可在“机器人详情”页获取，如下图

uniqueId 为自己添加的一个标示符，如邮箱、手机号等等

## 接入

### API接入

API地址:

APIKEY:

secret:  ☒ ON

[下载使用文档](#) 

### SDK接入

Android SDK :

[下载使用文档](#) 

IOS SDK :

[下载使用文档](#) 

(2) 参数 InitListener 是一个初始化后回调方法的接口

onComplete()是成功后，回调的方法

onFail(java.lang.String error)是失败后，回调的方法

## 2.6.2 实例化 TuringApiManager 类

实例化 TuringApiManager 类

这里一定要在上一步 SDKInit 初始化成功后，再初始化 TuringApiManager，否则很多功能将无法使用

推荐的代码：

```
onComplete()

{

    TuringApiManager m = new TuringApiManager (this);

}
```

### 2.6.3 设置监听

添加监听：

```
public void setHttpListener(HttpConnectionListener httpConnectionListener)
```

参数：httpConnectionListener 用于监听联网请求结果的回调

代码形式如下：

```
TuringApiManager m = new TuringApiManager (this);
```

```
m.setHttpListener(httpConnectionListener);
```

其中 httpConnectionListener 就是实现接口 HttpConnectionListener

```
@Override
```

```
public void onSuccess(RequestResult result)
```

```
{
```

```
    //成功后回调
```

```
}
```

```
@Override
```

```
public void onError(ErrorMessage errorMessage)
```

```
{
```

```
    //失败后回调
```

```
    Log.d(TAG, errorMessage.getMessage());
```

```
}
```

### 2.6.4 发出请求

发出请求

```
public void requestTuringAPI(String requestInfo)
```

参数：requestInfo 为传递的文本

比如 requestInfo = “你好” ,那么在上一步的 **public void** onSuccess(RequestResult result) 中 result 可以得到一串 json 字符串，其中 json 字符串根据不同的类型会有不同的格式.

示例一：

```
requestInfo = “你好”
```

返回结果:

```
{
    "code":100000,
    "text":"你好，我是聪明可爱的机器人"
}
```

示例二:

```
requestInfo = “给我一张图片” ;
```

返回结果:

```
{
    "code": 200000,
    "text": "亲，已帮你找到图片",
    "url": "http://m.image.so.com/i?q=%E5%B0%8F%E7%8B%97"
}
```

## 2.7 语音合成（TTS）

语音合成部分主要包括以下四个步骤

### 2.7.1 实例化 TTSTManager 类

TTSTManager 类实例化

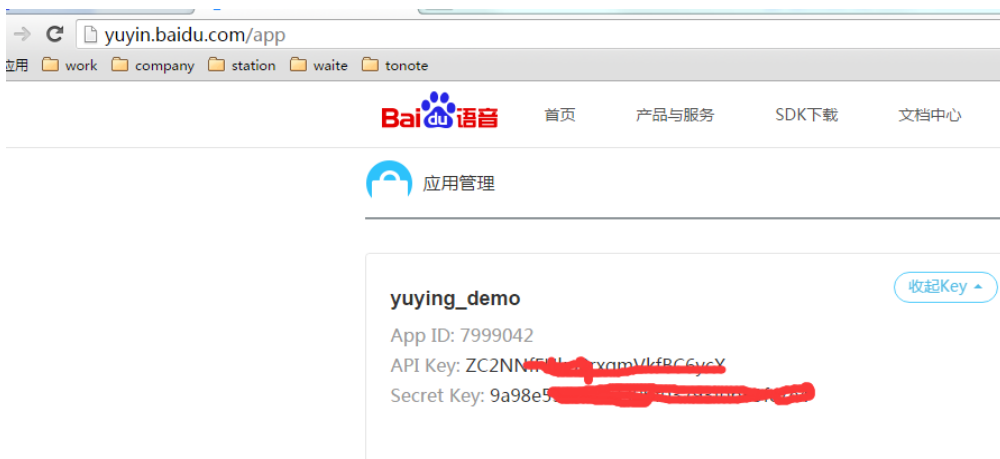
该类为语音合成的管理类，要使用语音合成功能，首先要创建 TTSTManager 的对象实例，由于该 SDK 支持百度 TTS，不再支持讯飞

如果选用百度在线 TTS，其构造函数如下：

```
public TTSTManager(Context context, String bdAPI_KEY, String bdSECRET_KEY)
```

这里的 bdAPI\_KEY 和 bdSECRET\_KEY，需要自己到百度官方去申请

<http://yuyin.baidu.com/>，位置如图所示：



### 2.7.2 设置监听

添加监听：TTSTManager.setTTSTListener(mTTSTListener)

mTTSTListener 需要实现接口 TTSTListener，可以在不同的语音合成状态下回调其不同的方

法.

### 2.7.3 开始语音合成

调用 `TTSManager.startTTS(String ttsContent)` 方法来进行语音合成

参数：`ttsContent` 就是需要被合成的文本，比如 `ttsContent=“你好”`，那么就会读出你好。

### 2.7.4 处理合成完成

语音合成后，就会触发 `onSpeechFinish()`，这样即可在其方法中添加相应的逻辑。

## 3 常见问题

Q：安卓 SDK 在哪里可以下载？

A：登录图灵账号，在除微信机器人、QQ 机器人两个场景外的机器人详情页中就能下载到 SDK，除此之外，图灵机器人还提供 IOS 版本的 SDK。

Q：安卓 SDK ASR 和 TTS 不可用？

A：使用相应的功能，需要去第三方平台申请相应的账号，并且开通语音识别和语音合成的功能。

Q：Demo 无法正常对话？

A：确保在图灵机器人官网申请正确的 `turingApiKey`，并保证正确获取 `userid`，可在 SDK

初始化回调中获取相应的信息。