
科大讯飞股份有限公司

iFLYTEK CO.,LTD

自动更新

Android SDK 接入指南

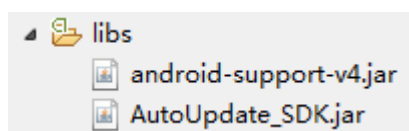
1. 概述

尊敬的开发者朋友，本文档将帮助您以最快速最简洁的方式集成使用自动更新组件。

2. 集成说明

2.1 导入 SDK

1. 下载最新版 SDK 的 zip 包，将其中的 sdk 文件夹中的 android-support-v4.jar 和 AutoUpdate_SDK.jar 复制到本地工程 libs 子目录下。



2. Eclipse 用户右键点击工程根目录，选择 Properties -> Java Build Path -> Libraries，然后点击 Add External JARs... 选择上述说明的两个 jar 文件，点击 OK。

注意：

Eclipse ADT 17 以上版本用户，不要通过上述步骤手动添加 jar 包引用。请在工程目录下建一个文件夹 **libs**，把 jar 包直接拷贝到这个文件夹下，再在 Eclipse 里面刷新一下工程就好了。详情请参考 [Dealing with dependencies in Android projects](#)。

2.2 配置 AndroidManifest 文件

打开 AndroidManifest.xml 文件，将 SDK 所需的权限配置到 **<manifest>** 标签下，将 APPKEY 及渠道配置到 **<application>** 标签下：

权限	用途
INTERNET	允许程序联网和发送统计数据的权限
ACCESS_NETWORK_STATE	允许应用检测网络连接状态
WRITE_EXTERNAL_STORAGE	允许应用获取写 sd 卡权限
meta-data	用途
IFLYTEK_APPKEY	用来唯一标识您的应用，请将以下示例代码中 your_appid 替换为您在讯飞开放平台 (http://www.xfyun.cn) 申请的应用 appid。
IFLYTEK_CHANNEL	用来标注应用推广渠道，区分用户来源，请将以下示例代码中 your_channel 替换为您自定义的渠道名称。格式：32 个字符以内，支持中文、英文、数字，请勿使用其他符号。

代码示例：

```
<manifest.....>
<application.....>
<activity ...../>
<!-- 请注意添加单引号-->
<meta-data android:name="IFLYTEK_APPKEY" android:value="'your_appid'"/>
<meta-data android:name="IFLYTEK_CHANNEL" android:value="your_channel"/>
</application>
<uses-permission android:name="android.permission.INTERNET">
    </uses-permission>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE">
    </uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE">
    </uses-permission>
</manifest>
```

2.3 添加下载 Service 及更新提示界面

添加下载所需的服务和提示弹出框。

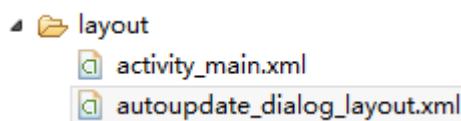
```
<manifest.....>
<application .....>
<activity android:name ="com.iflytek.autoupdate.IFlytekUpdateDialog"/>
<service android:name="com.iflytek.autoupdate.IFlytekDownloadService"/>
</application>
</manifest>
```

2.4 自定义样式

自动更新 SDK 提供两种更新提醒方式：1）对话框提醒更新；2）通知栏提醒更新。

使用第一种提醒方式时，需将 AutoUpdate_Demo 的 autoupdate_dialog_layout.xml 文件导入到您的 layout 文件夹中，如下所示。通过 autoupdate_dialog_layout.xml，您可以修改对话框的样式。

注意：不能修改控件的 id。（2.0.3 版本之后优化了界面布局，请注意替换）



2.5 使用更新接口

主要应用场景：最常见的自动更新模式，当用户进入应用首页后，如果处于 Wi-Fi 环境则检测更新，如果有更新，弹出对话框提示有新版本，用户点选更新开始下载更新。

在应用程序入口方法中调用，代码示例如下：

```
//初始化自动更新对象
IFlytekUpdate updManager = IFlytekUpdate.getInstance(mContext);
//开启调试模式，默认不开启
updManager.setDebugMode(true);
//开启wifi环境下检测更新，仅对自动更新有效，强制更新则生效
updManager.setParameter(UpdateConstants.EXTRA_WIFIONLY, "true");
//设置更新提示类型，默认为通知栏提示
updManager.setParameter(UpdateConstants.EXTRA_STYLE, UpdateConstants.UPDATE_UI_DIALOG);
// 启动自动更新
updManager.autoUpdate(MainActivity.this, updateListener);

//自动更新回调方法，详情参考demo
IFlytekUpdateListener updateListener = new IFlytekUpdateListener() {
    @Override
    public void onResult(int errorcode, UpdateInfo result) {}
};
```

考虑到用户流量的限制，本自动更新 SDK 提供两种更新请求接口，应用于不同场景：

1. 自动更新：该场景适用于后台自动检测更新。此场景支持仅在 Wi-Fi 场景下才会请求更新信息，在移动网路下调用是不会进行更新请求；调用请见 *IFlytekUpdate*.autoUpdate(*this*, *listener*);
2. 强制更新：该场景为用户自动请求自动更新信息。此场景无网络条件限制，即 Wi-Fi 环境和移动网络环境均能请求更新；调用请见 *IFlytekUpdate*.forceUpdate(*this*, *listener*);

注：无论是自动更新还是强制更新，如果开发者在启动更新时主动传入 *IFlytekUpdateListener* 对象，请求到的更新信息会通过回调接口进行返回。开发者获取到更新信息后可以选择继续调用本 SDK 模块或者进行自己的逻辑处理。如果开发者在启动更新时未传 *IFlytekUpdateListener* 对象，sdk 会自行完成更新流程，即正常进行更新提示和用户点击下载流程，且该流程不会返回更新信息。

2.6 上传最新 APK

如果开发者已经有了最新的 APK 版本，只要上传到讯飞开放平台(<http://www.xfyun.cn>)，同时客户端版本的版本号小于上传的最新版本，客户端在启动时就会有更新提示。

上传位置：

- 1) 登录讯飞开放平台，在顶部标签处选择“我的语音云” -> “我的应用”；



- 2) 在“我的应用”界面中，点击“创建新应用”，创建您的 APP。如果您已经创建过应用，则跳过该步骤，执行 3)；
- 3) APP 创建成功后，选择“立即开通服务”，在弹出框中选择“版本自动更新”，上传最新 APK。如果您已经创建过应用，则选择“开通更多服务”，执行上传操作。



2.7 集成检测

sdk 中默认开启了集成检测功能，在调用任意更新接口后，我们将替您自动检查上述集成过程中 2.2、2.3、2.4 三个步骤是否被正确完成。如果正确完成不会出现任何提示，否则会打印出以下错误 log。

log 的含义如下：

"Please set IFLYTEK_APPKEY!": 请检查上述步骤 2.2 中的 IFLYTEK_APPKEY 是否正确添加。

"Please add Permission in AndroidManifest!": 请检查上述步骤 2.2 中 permission 是否正确添加。

"Please add Activity in AndroidManifest!": 请检查上述步骤 2.3 中的 Activity 是否正确添加。

"Please add Service in AndroidManifest!": 请检查上述步骤 2.3 中的 Service 是否正确添加。