

微信图像 SDK(Android 版)上手指南

使用说明

本文档主要介绍微信图像 SDK（Android 版）使用方法，利用 SDK 可以准确搜索到用户上传的图像。

本文属于入门级文档，旨在帮助开发者快速学习 Android SDK 的使用并应用到自身开发工作。具体 API 可查询《Android 开发手册》。

获取应用授权码

请到“开发者应用登记页面”进行登记，登记并选择移动应用进行设置后，将该应用提交审核，只有审核通过的应用才能进行开发。

要获取应用授权码，可以在 Eclipse 中直接查看，Windows -> Preferences -> Android -> Build，界面如图 1，其中的 MD5 fingerprint 就是需要的应用签名值：

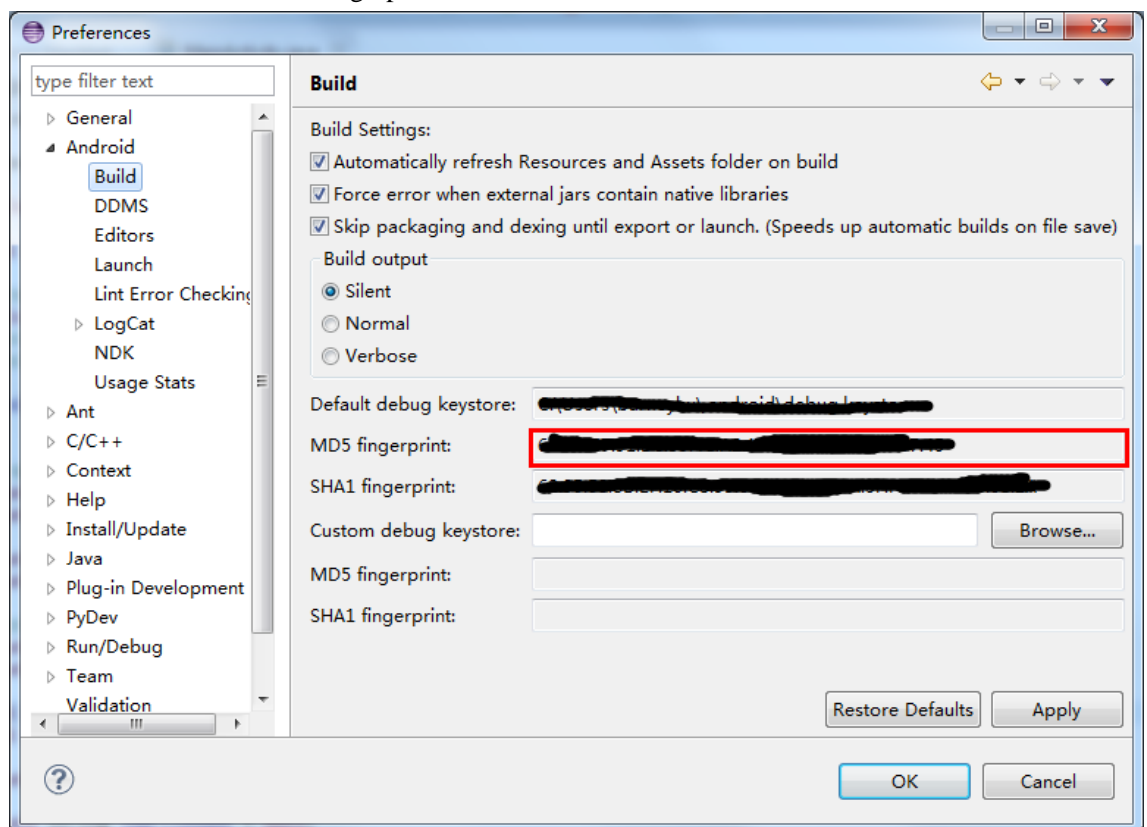


图 1

注册完毕，我们会在 7 个工作日内完成审核工作。

下载 Android SDK

进入“资源中心”下载 Android SDK，压缩包中包括 Demo+SDK+开发文档。其中的 Demo 使用 SDK 的各功能 API；SDK 包括.so 和.jar 文件。

上传图片

每个 AppID 只能检索到自身上传的图片，图片上传位置：资源中心 -> 智能识别图片库，待图片审核通过，即可对此图片进行识图操作。

Demo 介绍

为了更好的理解微信图像 SDK 的使用，下面将通过一个简单的实例来讲解一下 SDK 各个关键 API 接口的使用。

1. 开发工具

Android 开发工具有很多，开发者可以根据自身的喜好来选择。在讲解本示例的时候，我们将使用 Eclipse 来一步步分析。

2. 工程配置

首先新建一个示例工程，按以下步骤进行配置。

i. 引入.so 文件：

如下图，在 libs\armeabi 目录下粘贴 libWXPrNet.so 文件。

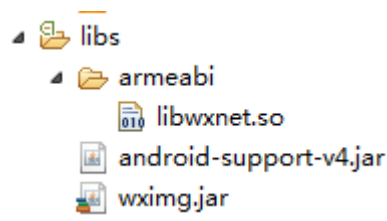


图 2

ii. 引入 jar 包：

在工程属性->Java Build Path->Libraries 中选择“Add External JARs”，选定 wximg.jar，确定后返回，完成后的效果如图 2 所示。

3. AndroidManifest.xml 设置

i. 添加必要的权限支持：

```

<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.READ_PHONE_STATE"></uses-permission>
<uses-permission android:name="android.permission.RECORD_AUDIO"></uses-permission>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>

```

4. 图像识别 Demo（包括通用识别和语法识别）

关于界面的构建此处不进行介绍，下面着重介绍图像识别 SDK 各 API 的使用：

i. 初始化

```

ImgSearcher.shareInstance().setListener(this);
if (ImgSearcher.shareInstance().init(this, screKey) != 0) {
    //初始化失败
}

```

注：此处的 screKey 就是在官网上申请的应用授权码。

ii. 开始识别

```

ImgSearcher.shareInstance().start(img);

```

注：img 是图像的 byte[] 型内容。

开始后的各种阶段状态将由 public void onGetState(ImgSearcherState state) 回调获取。

iii. 取消识别

```

ImgSearcher.shareInstance().cancel();

```

此接口的如果返回非 0 值，则表示 cancel 过程结束，否则其状态将通过 public void onGetState(ImgSearcherState state) 回调获取。

iv. 实现 ImgListener，并重写其函数

直接在 Activity 类中实现 ImgListener，重写 ImgListener 各函数如下所示：

```

@Override
public void onError(int arg0) {
    // TODO Auto-generated method stub
}

```

```

@Override
public void onGetResult(ImgResult result) {
    // TODO Auto-generated method stub
    String mResUrl;
    String mResMD5;
    String mResPicDesc;
    if (result != null) {
        if (1 == result.ret && result.res != null) {
            int resSize = result.res.size();
            for (int i = 0; i < resSize; ++i) {
                ImgResult.Result res = (ImgResult.Result) result.res.get(i);
                if (res != null) {
                    mResMD5 = res.md5;
                    mResUrl = res.url;
                    mResPicDesc = res.picDesc;
                }
            }
        }
    }
}

@Override
public void onGetState(ImgSearcherState state) {
    // TODO Auto-generated method stub
}

```

v. 释放系统资源

```
ImgSearcher.shareInstance().destroy();
```

注：调用此函数， 需要确保识别过程或者 cancel 过程已经结束， 否则会出现问题。

vi. Demo 截图

编译运行，图 3、图 4 分别是开始界面和识别结果界面：



图 3



图 4

更多内容请参考《Android 开发手册》和 WXImgSDKDemo。