

Weibo Android SDK 使用手册

1. 创建新浪微博开放平台应用

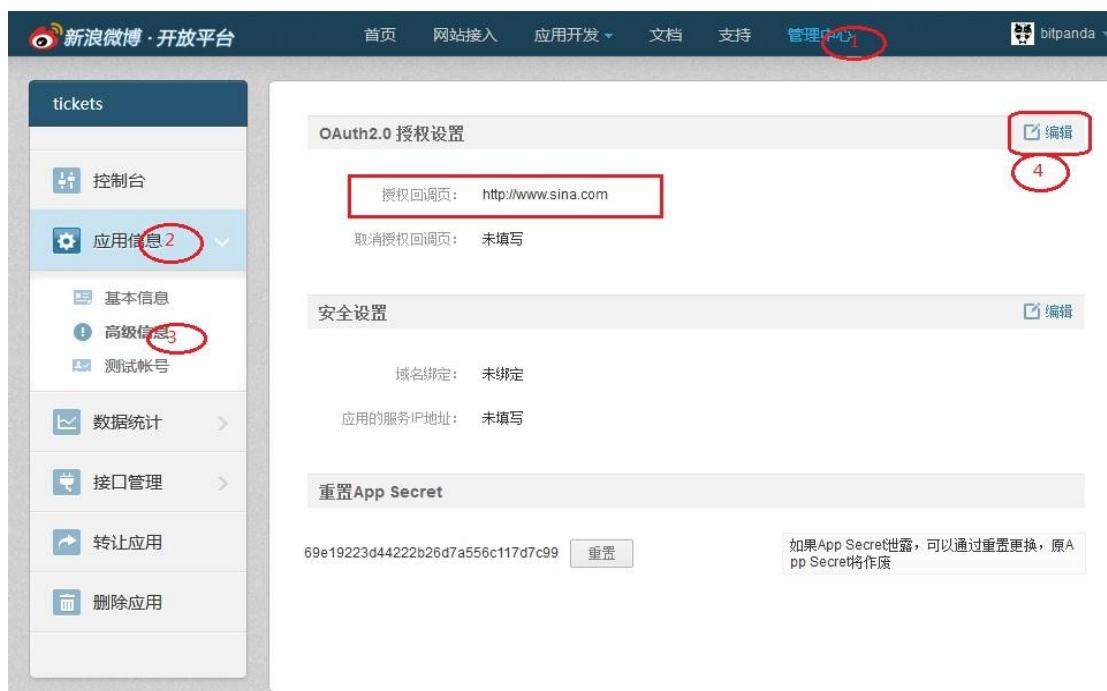
1) 获取 app_key, app_secret

第三方开发者须到新浪微博开放平台 <http://open.weibo.com/> 注册并创建第三方应用。进而获取应用专属 app_key 及 app_secret。但应用登记完成后还需要提交审核，只有审核通过的应用才能正式发布使用。App Key 及 App Secret 查看方式：第三方应用主页→应用信息→基本信息，应用基本信息一栏。



2) 配置授权回调页

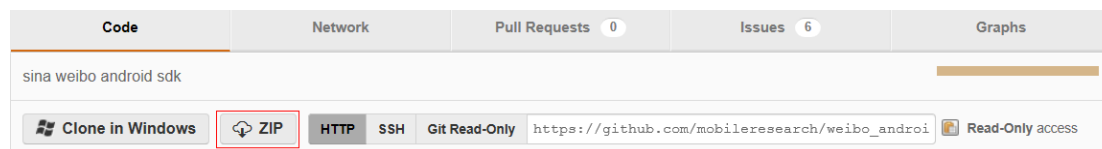
授权回调页是为 OAuth2.0 认证机制中的登陆认证地址，用户登录完成后最后会跳转此地址。须在应用中配置回调跳转地址才能使用 SDK 完成用户登录。授权回调页查看及设置地址：第三方应用主页→应用信息→高级信息→OAuth2.0 授权设置。如下图所示：



注：在使用 SDK 时，配置授权毁掉也是必不可少的，若没有配置，则登陆完成后无法检测到授权地址，就无法获取授权的 Token 等信息。此地址并非必须配置成能访问地址，保证格式正确即可。

2. 下载 Android 终端开发工具包

在新浪微博开放平台文档一页有 SDK 下载一项，找到 IOS SDK 下载链接 https://github.com/mobileresearch/weibo_android_sdk/downloads，打开后可打包下载。



上图“ZIP”按钮，下载后解压缩可得到开发工具包。开发工具主要包含：

Weibo.sdk.android.api (封装 Weibo.api 接口)

Weibo.sdk.android.sso.demo (使用 sso 一个 Demo，包含调用 weibo.api 接口)

Weibo.sdk.android.sso (基于 weibo.sdk.android 扩展的 sso 源码)

Weibo.sdk.android (OAuth2.0 封装源码)

Android 平台微博 SDK 说明文档 (建议下载，方便开发)

Weibo.sdk.android.sso.jar (必须下载，该 jar 封装了 weibo.api 和 sso 相关方法)

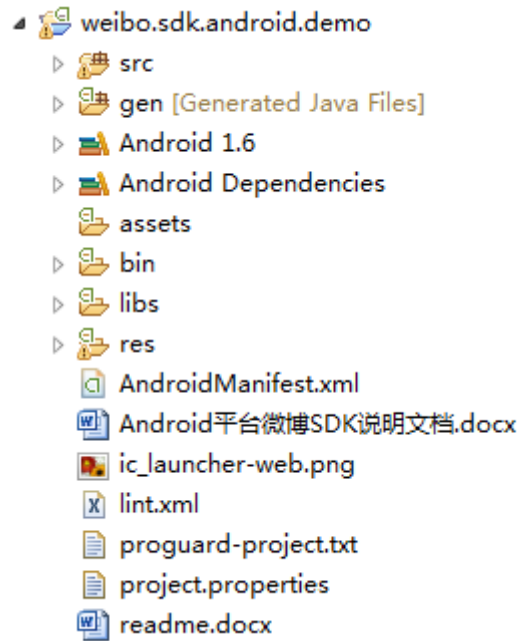
开发工具下载地址：

3. 搭建开发环境

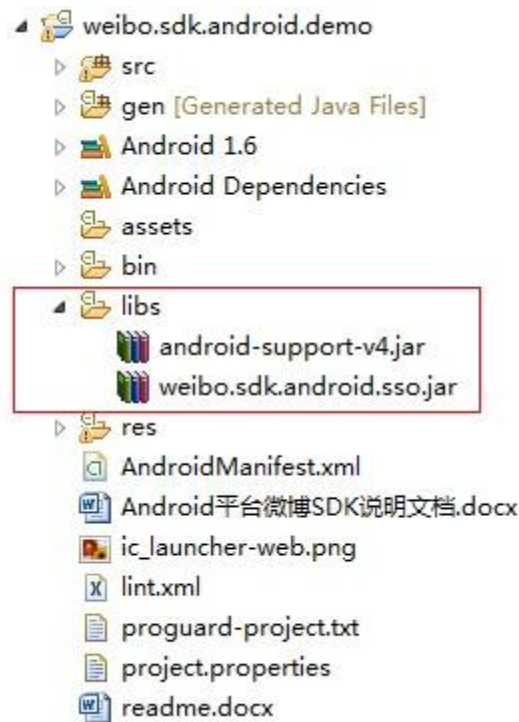
环境需求：JDK+SDK+IDE (Eclipse) + Weibo.sdk.android.sso.jar

此处基于 Weibo.sdk.android.sso.demo 为例。可以类比新建项目。

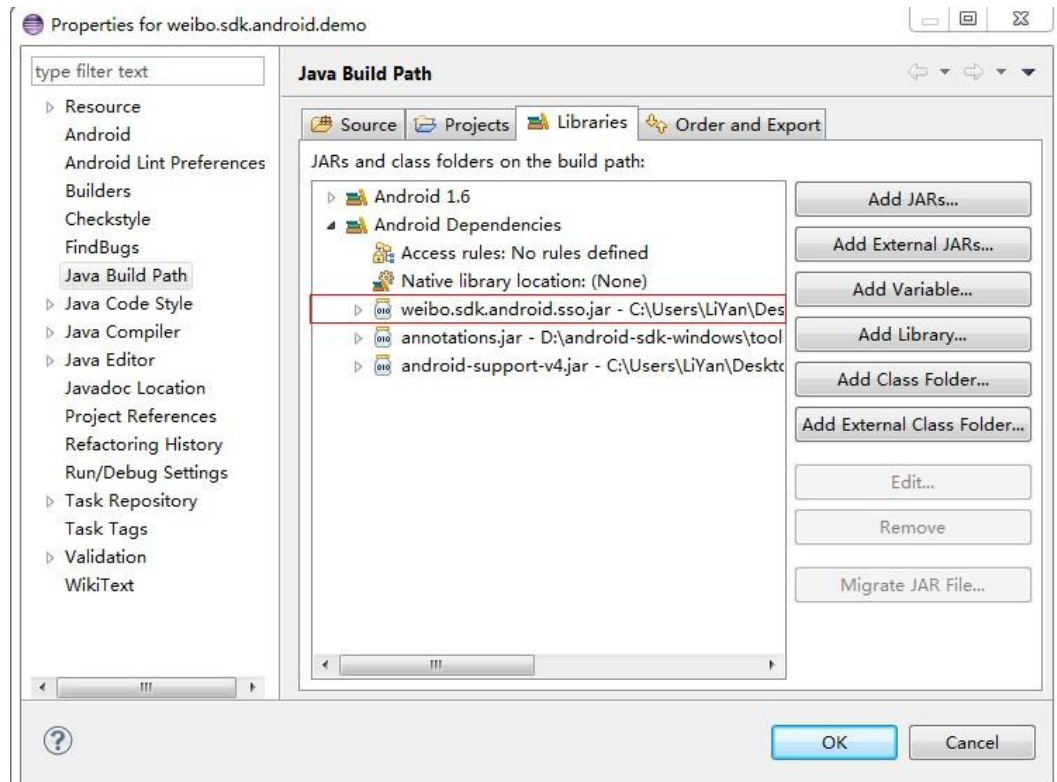
- 1) 在 Eclipse 中右键选择 Import，选择 Existing Projects into Workspace，从中选择 weibo.sdk.android.sso.demo 项目所在位置。导入结果如下图所示：



- 2) 在工程中的 libs 目录下，将从开发工具下载地址中下载的 weibo.sdk.android.sso.jar 复制到该目录中（如下图所示，在 weibo.sdk.android.sso.demo 工程中，将 weibo.sdk.android.sso.jar 包复制到 libs 目录下）。



- 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 weibo.sdk.android.sso.jar 文件。



4. 在代码中使用开发工具包

1) 通过 OAuth2.0 授权获得 AccessToken

i. MainActivity.java

通过 Weibo 的构造函数设置 AppKey 和应用回调页

```
mWeibo = Weibo.getInstance(CONSUMER_KEY, REDIRECT_URL);
```

此处 CONSUMER_KEY: 申请应用时分配的唯一应用认证 ID。

此处 REDIRECT_URL: 授权回调页。

ii. MainActivity.java

实现 WeiboDialogListener 接口，授权成功后即可在 onComplete 回调方法中获得 accessToken 信息，此时可以自由保存处理。如下图所示：

```

.....class AuthDialogListener implements WeiboAuthListener {
}
.....@Override
.....public void onComplete(Bundle values) {
.....String token = values.getString("access_token");
.....String expires_in = values.getString("expires_in");
.....MainActivity.accessToken = new OAuth2AccessToken(token, expires_in);
.....if (MainActivity.accessToken.isSessionValid()) {
.....String date = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss")
.....format(new java.util.Date(MainActivity.accessToken
.....getExpiresTime()));
.....mText.setText("认证成功: \r\n access_token: " + token + "\r\n"
.....+ "expires_in: " + expires_in + "\r\n有效期: " + date);
.....try {
.....Class sso = Class
.....forName("com.weibo.sdk.android.api.WeiboAPI"); // 如果支持weiboapi的话, 显示api功能演示入口按钮
.....apiBtn.setVisibility(View.VISIBLE);
.....} catch (ClassNotFoundException e) {
.....// e.printStackTrace();
.....Log.i(TAG, "com.weibo.sdk.android.api.WeiboAPI not found");
.....}
.....cancelBtn.setVisibility(View.VISIBLE);
.....AccessTokenKeeper.keepAccessToken(MainActivity.this,
.....accessToken);
.....Toast.makeText(MainActivity.this, "认证成功", Toast.LENGTH_SHORT)
......show();
.....}
}
}

```

iii. APITypeListActivity.java

通过已经获得 accessToken 调用 WeiboAPI 相关方法，如下图所示：

```

.....StatusesAPI statusApi = new StatusesAPI(MainActivity.accessToken);
.....statusApi.friendsTimeline(0, 0, 50, 1, false, FEATURE.ALL, false,
.....this);

```

2) 通过 SSO 功能进行授权

- i. 使用 SSO 功能需要覆写 onActivityResult() 方法，并调用 AuthorizeCallBack() 方法。如下图所示：

```

.....@Override
.....protected void onActivityResult(int requestCode, int resultCode, Intent data) {
.....super.onActivityResult(requestCode, resultCode, data);
.....
...../**
..... * 下面两个注释掉的代码，仅当sdk支持sso时有效，
..... */
.....if (mSsoHandler != null) {
.....mSsoHandler.authorizeCallBack(requestCode, resultCode, data);
.....}
}
}

```

5. 特殊说明

Weibo.sdk.android.sso.jar 是由 weibo.sdk.android.jar、weibo.sdk.android.api.jar、weibo.sdk.android.sso.jar 三个组成。可以自行打包或查看源码。

注：本文旨在说明第三方应用程序如何使用 Android SDK，只涉及 SDK 的引入和使用方法。有关 SDK 中接口的描述请参见《Android 平台微博 SDK 说明文档》，详尽的使用方法请参考 Demo 中的代码及相关 jar 包源码。