

微博 Android 平台 SDK 文档

编号：WEIBO_ANDROID_SDK

版本：WEIBO_ANDROID_SDK V2.4.0

修订记录：

| 时间 | 文档版本 | 修订人 | 备注 |
|------------|-------|-----|---|
| 2012/7/20 | 2.0.0 | | 初稿 |
| 2012/8/02 | 2.0.0 | | |
| 2013/4/17 | 2.1.0 | | 新增分享微博 |
| 2013/9/12 | 2.3.0 | | 新增登入登出按钮、好友邀请接口 |
| 2013/11/11 | 2.4.0 | | 重大版本变更： <ul style="list-style-type: none">➤ 授权、分享等代码的重构➤ 代码、文档等规范化➤ 开源 OpenAPI 部分代码 |

目录

| | |
|---------------------------------|----|
| 微博 Android 平台 SDK 文档..... | 1 |
| 一、 概述..... | 3 |
| 名词解释..... | 3 |
| 二、 功能列表..... | 3 |
| 1. 认证授权..... | 3 |
| 2. 微博分享..... | 3 |
| 3. 登录/注销按钮..... | 4 |
| 4. 开放接口..... | 4 |
| 三、 运行示例代码..... | 4 |
| 1. 替换默认的 debug.keystore..... | 4 |
| 2. 导入工程..... | 5 |
| 3. 编译运行..... | 5 |
| 四、 微博 SDK 及 DEMO 工程目录结构及分析..... | 5 |
| 1. 闭源部分结构分析..... | 6 |
| 2. 开源部分结构分析..... | 6 |
| 3. Demo 部分结构分析..... | 7 |
| 五、 集成前准备..... | 7 |
| 1 申请应用程序的 APP_KEY..... | 7 |
| 2 注册应用程序的包名和签名..... | 7 |
| 3 选择您要集成的方式..... | 9 |
| 4 在您的应用中添加 SDK 所需要的权限..... | 10 |
| 六、 示例分析：认证授权..... | 11 |
| 1 替换成自己应用的 APP_KEY 等参数..... | 11 |
| 2 创建微博授权类对象..... | 12 |
| 3 实现 WeiboAuthListener 接口..... | 12 |
| 4 调用方法，认证授权..... | 13 |
| 七、 示例分析：分享微博..... | 14 |
| 1. 认识 IWeiboShareAPI 接口..... | 14 |
| 2. 从第三方应用唤起微博客户端进行分享..... | 15 |
| 3. 从微博客户端唤起第三方应用进行分享..... | 19 |
| 八、 示例分析：登录/注销按钮..... | 22 |
| 1. 一键登录按钮..... | 22 |
| 2. 登录/注销按钮..... | 23 |
| 九、 示例分析：开放接口..... | 25 |
| 1. 注销接口..... | 25 |
| 2. 好友邀请接口..... | 27 |

一、概述

微博 SDK 为开发者提供了 OAuth2.0 授权认证，并集成 SSO 登录功能，使第三方应用无需了解复杂的验证机制即可进行授权登录操作，并提供微博分享功能，第三方应用可直接通过微博客户端分享微博。

本文档对使用 SDK 时所用的一些接口、参数等进行说明，并分析简单的示例分析，帮助第三方开发者快速集成应用。

名词解释

| 名词 | 注解 |
|------------------------|--|
| AppKey | 分配给每个第三方应用的 App Key。用于鉴权身份，显示来源等功能。 |
| RedirectURI | 第三方应用授权回调页面。 授权回调页对移动客户端应用来说对用户是不可见的，所以定义为哪种形式都将不影响，但是没有定义将无法使用 SDK 认证登录。建议使用默认回调页 https://api.weibo.com/oauth2/default.html 可以在新浪微博开放平台->我的应用->应用信息->高级应用->授权设置->应用回调页中找到。 |
| Scope | Scope 是 OAuth2.0 新版授权页提供的一个功能，通过 scope，平台将开放更多的微博核心功能给开发者，同时也加强用户隐私保护，提升了用户体验，用户在新 OAuth2.0 授权页中有权利选择赋予应用的功能。 |
| AccessToken | 表示用户身份的 Token，用于微博 OpenAPI 的调用。 |
| Expire in | Token 过期时间，用于判断登录是否过期。 |
| OAuth2.0 Web 授权 | 通过 WebView 进行授权，并返回 Token 信息。 |
| SSO 授权 | 通过唤起微博客户端进行授权，并返回 Token 信息。 |

二、功能列表

1. 认证授权

目前微博 SDK 为开发者提供 OAuth2.0 Web 授权认证，并集成 SSO 登录功能。使用带 SSO 功能的 SDK 进行登录，只需调用登录接口，并完成回调方法对接收登录结果即可。

注意：SSO 需要微博客户端才能够进行。

2. 微博分享

通过微博 SDK，第三方应用能够分享文字、图片、视频、音乐等内容，目前分享有两种方式：

- 通过第三方应用唤起微博客户端进行分享（该分享方式为第三方客户端通常的使用方式）
- 通过微博客户端唤起第三方应用进行分享（该分享方式需要合作接入，详情请查看：<http://t.cn/aex4JF>）

注意：目前 SDK 只支持通过微博官方客户端进行分享。

3. 登录/注销按钮

微博 SDK 目前提供了两类登录按钮：一种是一键登陆按钮，一种是登陆/注销按钮。

两者实际上都是调用了 SSO 登录接口。另外，不再单独提供注销按钮，用户可以调用已封装好的 OpenAPI 接口直接注销。

4. 开放接口

微博 SDK 目前提供了一个简单的 OpenAPI 接口调用框架，并封装了一些简单的开放接口，以供大家参考：

- 注销接口：授权回收接口，帮助开发者主动取消用户的授权。
- 邀请接口：提供好友邀请接口，支持登录用户向自己的微博互粉好友发送私信邀请、礼物。

注：该模块目前还需要进一步设计和完善，后续将会提供一个更加合理好用的 OpenAPI 框架，并封装更多常用的接口（如利用 OpenAPI 获取用户信息，分享微博等），敬请期待。

三、 运行示例代码

为了方便第三方应用更快的集成微博 SDK，更清晰的了解目前微博 SDK 所提供的功能，我们在 GitHub（https://github.com/mobileresearch/weibo_android_sdk）上提供了一个简单的示例工程以及对应的 APK 安装包。

- 通过 adb install 命令直接安装 WeiboSDKDemo.apk
- 在 Eclipse 中导入并运行 WeiboSDKDemo 工程

我们简要的描述一下如何运行该 Demo 工程：

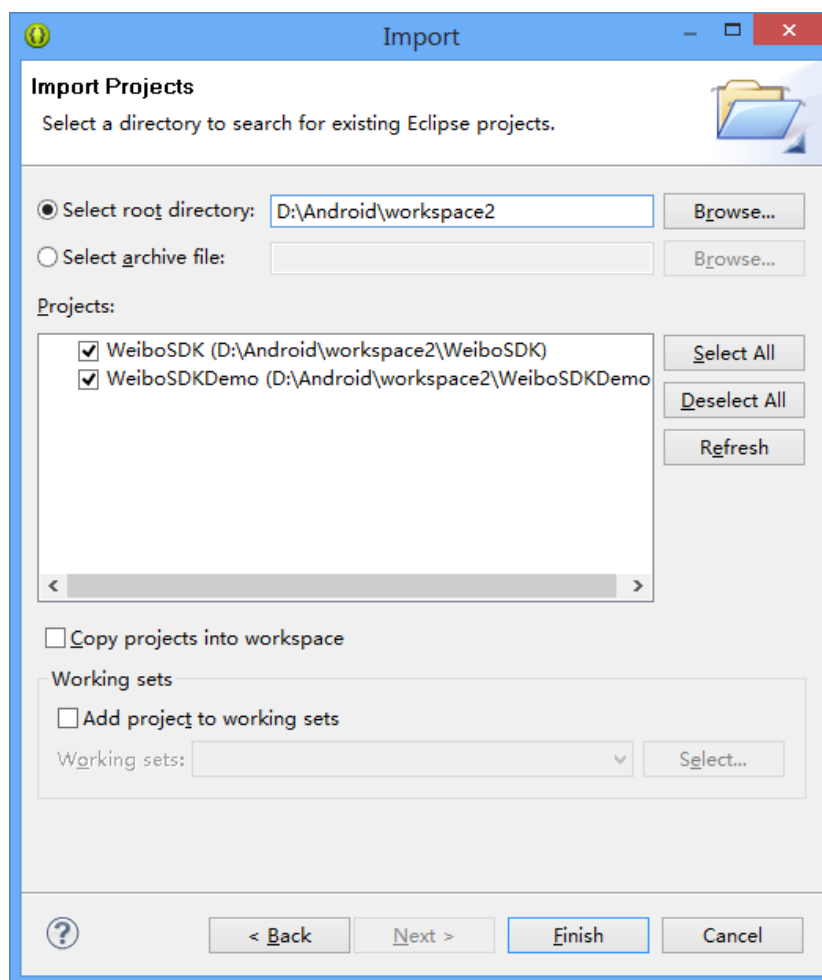
1. 替换默认的 debug.keystore

在 C:\Users\XXXXXX\android 目录下，把 Android 默认的 debug.keystore 替换成官方在 GitHub 上提供的 debug.keystore。

请注意：这一步是必须的，如果没有替换，demo 程序在运行时将无法正确的授权成功。用户在替换前，最好先备份一下原始的 debug.keystore。另外，该 debug.keystore 是新浪官方的，除了编译运行官方 DEMO 外，请不要直接使用它，出于安全的考虑，您应该为自己的应用提供一份 keystore。

2. 导入工程

在 Eclipse 中，点击 File→Import→Existing Projects into Workspace，输入正确的路径，导入工程，如下图：



3. 编译运行

在编译前，请先 Clean 一下工程，然后右键点击工程→Run As→Android Application，运行即可。默认的编译 target 是 Android 2.3.3，第三方可以自行修改成合适的版本。

另外，目前整个工程全采用中文注释，为了防止乱码滋生，请修改文本编码方式为 UTF-8。
Eclipse→Window→General→Workspace→Text file encoding。

四、 微博 SDK 及 DEMO 工程目录结构及分析

微博 SDK 目前以是**部分开源**的形式提供给第三方开发者的，简单来说，可以分为以下三部分：

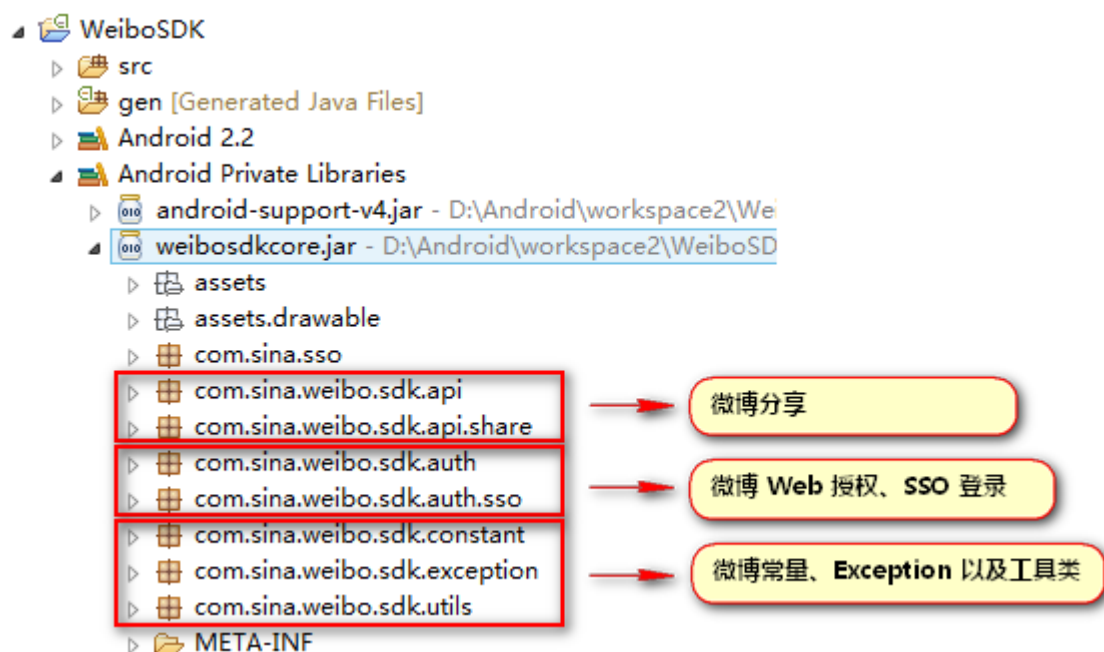
- **闭源部分**：[weibosdkcore.jar](#)，该 JAR 包集成了微博授权、SSO 登录以及分享等核心功能。
- **开源部分**：[WeiboSDK 工程 \(Library \)](#)，该工程引用了 [weibosdkcore.jar](#)，这里面主要是对 OpenAPI 进行

了简单的封装，第三方可以参考使用流程，模仿并添加自己需要的接口，利用 OpenAPI 接口获取用户信息，分享微博等，另外，还提供了一键登录/注销功能。

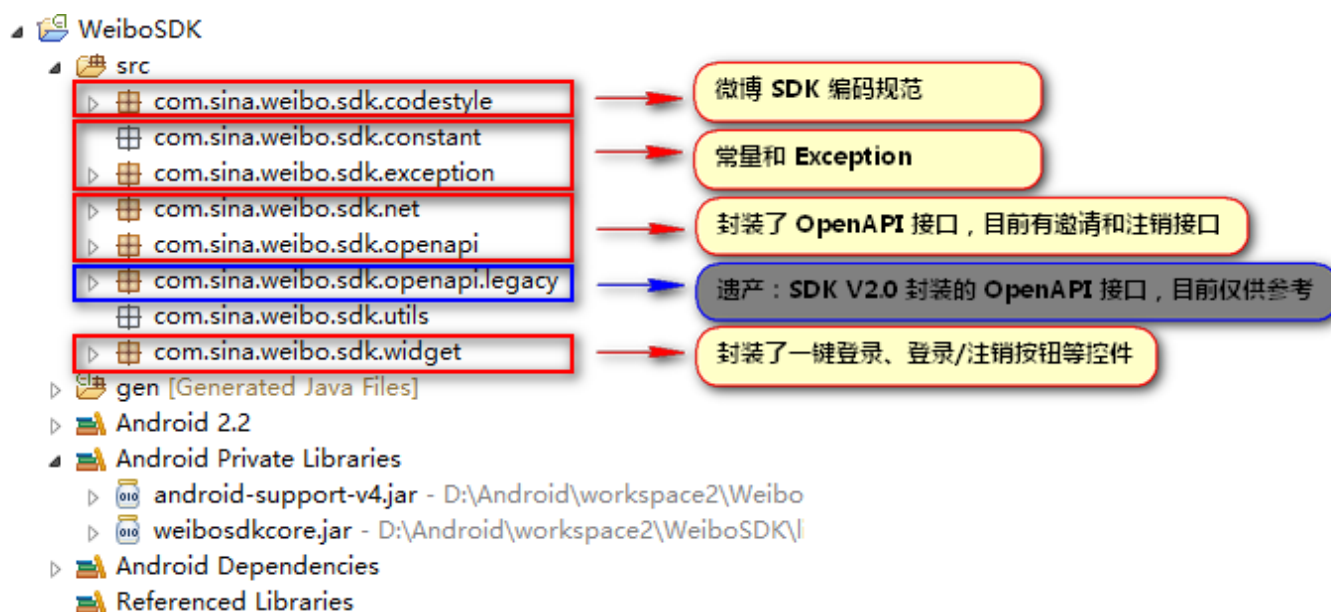
- **Demo 部分：**[WeiboSDKDemo 工程](#)，该工程引用了 [WeiboSDK 工程](#)，提供了目前微博所支持的所有功能的示例代码。

请注意：第三方在使用时，如果只想要授权和分享功能，可直接使用 [weibosdkcore.jar](#)；如果想使用其它功能，可直接导入 [WeiboSDK 工程](#)。如何导入 [WeiboSDK 工程](#)请详见：[集成步骤及示例分析](#)

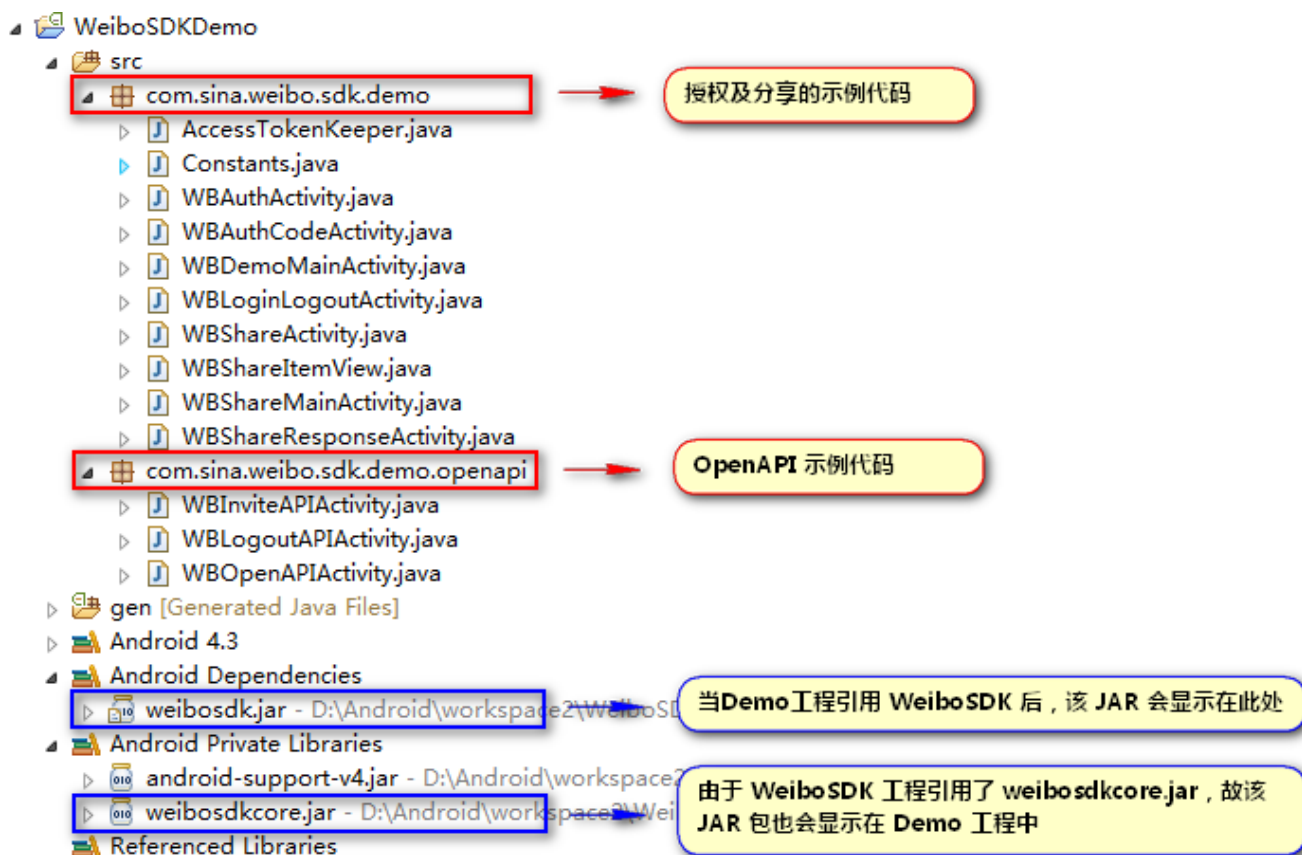
1. 闭源部分结构分析



2. 开源部分结构分析



3. Demo 部分结构分析



五、集成前准备

1 申请应用程序的 APP_KEY

在这一步中,您需要在微博开放平台上,对您的应用进行注册,并获取 APP_KEY,添加应用的授权回调页(Redirect URI)。详情请仔细阅读: [移动客户端接入](http://t.cn/aex4JF) (<http://t.cn/aex4JF>)

2 注册应用程序的包名和签名

您需要在微博开放平台上注册应用程序的包名和签名后,才能正确进行授权。

请注意: 包名和签名未注册,或者签名注册不正确,都会导致无法授权。

- 应用程序包名: 指 AndroidManifest.xml 文件中, package 标签所代表的内容。
- 应用程序签名: 该签名是通过官方提供的签名工具生成的 MD5 值

下图即为注册应用程序的包名和签名的页面,可以在新浪微博开放平台→我的应用→应用信息→应用基本信息处找到, 点击编辑按钮即可注册。

应用基本信息

应用类型： 普通应用 - 客户端

应用名称： 我的安卓客户端应用

该名称也用于来源显示，不超过10个汉字或20个字母

应用平台： 手机

[查看移动客户端接入指南](#)☐ iPhone ☒ Android ☐ BlackBerry☐ Windows Phone ☐ Symbian☐ WebOS ☐ Other

- * Android包名： com
- * Android签名： 702465el
Debug签名：
通过下载使用平台提供的[签名工具](#)获取
用户根据需要填写，可以不填。
- Android下载地址： http://

如何使用签名工具获取您应用的签名？

☞ 签名工具地址：

https://github.com/mobileresearch/weibo_android_sdk/blob/master/app_signatures.apk

☞ 使用方式：首先要安装您想要签名的应用，然后再安装该工具，安装完后，输入您的应用程序的包名，点击生成按钮，即可获得 MD5 签名，如下图所示。

请注意：要签名的第三方应用程序**必须安装**在该设备上才能够生成对应的 MD5 签名

3 选择您要集成的方式

在集成微博 SDK 前，您有两种可选的方式来集成微博 SDK：

- 直接导入 weibosdkcore.jar：适用于只需要微博授权及分享的项目
- 引用 WeiboSDK 工程（Library）：适用于微博授权、分享，以及需要登陆按钮、调用 OpenAPI 的项目

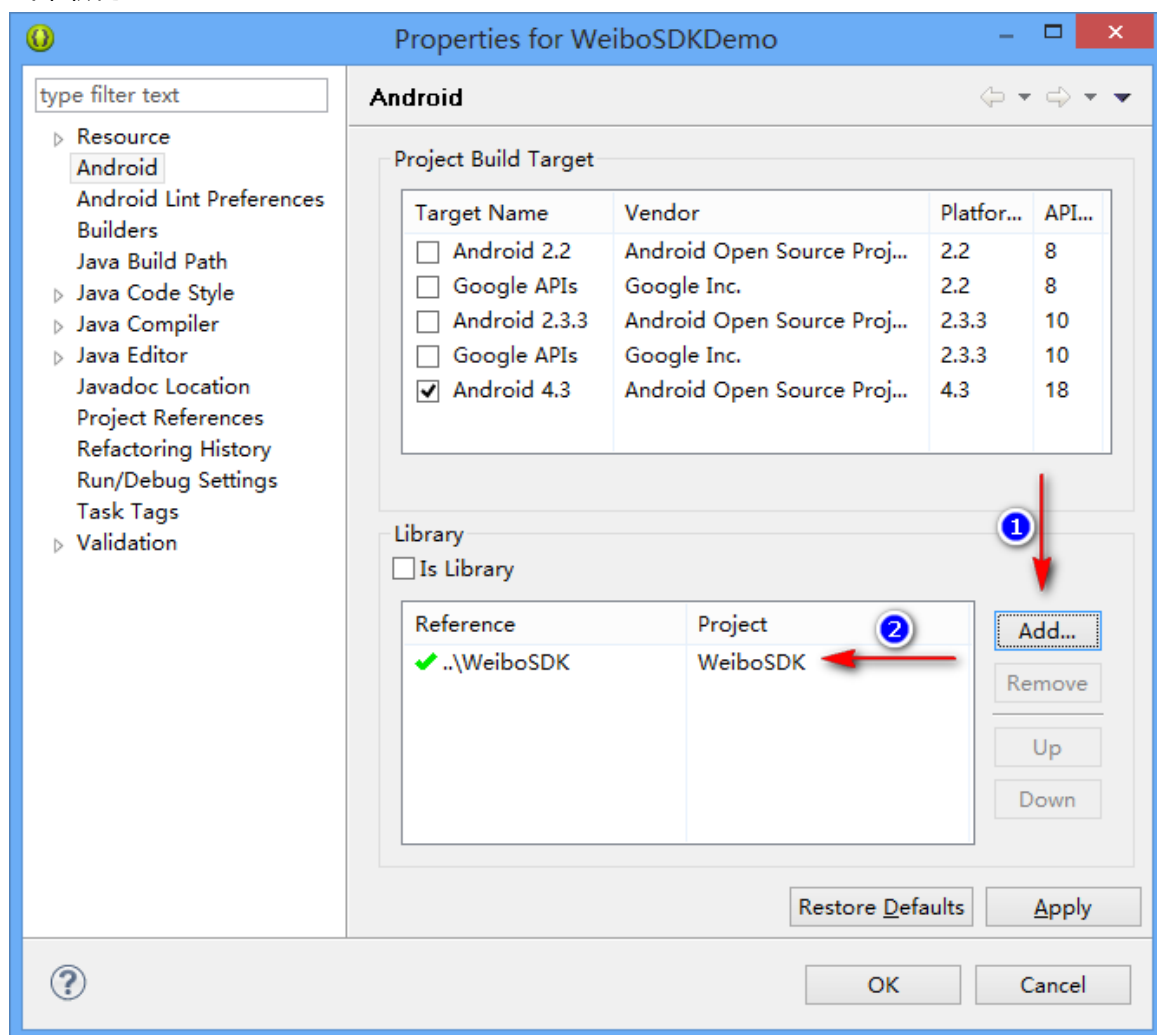
我们简单描述一下两种导入方式：

1. 直接导入 weibosdkcore.jar

- ☞ 将官方的提供 SDK JAR 包（weibosdkcore.jar）放到工程的 libs 目录下
- ☞ 添加 JAR 包：工程→右键→properties→java build path→libraries→add external jar

2. 引用 WeiboSDK 工程（Library）

- ☞ 将 WeiboSDK 工程整个目录拷贝到和你自己的工程相同的目录下
- ☞ 在你自己的工程中，添加 WeiboSDK 工程的引用：工程→右键→properties→Android→Add→选择工程，具体如下图所示：



4 在您的应用中添加 SDK 所需要的权限

打开 AndroidManifest.xml 文件，将 SDK 需要的权限添加到该文件中：

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

六、 示例分析：认证授权

微博 SDK 目前提供两种授权方式，一种是通过 WebView 进行 Web 授权，另外一种是通过微博客户端进行 SSO 授权。以下部分详细描述这两种授权方式。

注：对于想直接通过认证 Code 获取 Token 的开发者，可以详见 WBAuthCodeActivity 类。

1 替换成自己应用的 APP_KEY 等参数

鉴于目前有很多第三方开发**直接拷贝并使用** Demo 中的 Constants 类，因此，有必要说明，第三方开发者需要将 Constants 类中的各种参数替换成自己应用的参数，请仔细阅读代码注释。

```
public interface Constants {
    /** 当前 DEMO 应用的 APP_KEY，第三方应用应该使用自己的 APP_KEY 替换该 APP_KEY */
    public static final String APP_KEY      = "2045436852";

    /**
     * 当前 DEMO 应用的回调页，第三方应用可以使用自己的回调页。
     *
     * <p>
     * 注：关于授权回调页对移动客户端应用来说对用户是不可见的，所以定义为何种形式都将不影响，
     * 但是没有定义将无法使用 SDK 认证登录。
     * 建议使用默认回调页：https://api.weibo.com/oauth2/default.html
     * </p>
     */
    public static final String REDIRECT_URL = "http://www.sina.com";

    /**
     * Scope 是 OAuth2.0 授权机制中 authorize 接口的一个参数。通过 Scope，平台将开放更多的
     * 微博核心功能给开发者，同时也加强用户隐私保护，提升了用户体验，用户在新 OAuth2.0 授权
     * 页中有权利选择赋予应用的功能。
     * 我们通过新浪微博开放平台-->管理中心-->我的应用-->接口管理处，能看到我们目前已有哪些接
     * 口的使用权限，高级权限需要进行申请。
     * 目前 Scope 支持传入多个 Scope 权限，用逗号分隔。
     *
     * 有关哪些 OpenAPI 需要权限申请，请查看：http://open.weibo.com/wiki/%E5%BE%AE%E5%8D%9AAPI
     * 关于 Scope 概念及注意事项，请查看：http://open.weibo.com/wiki/Scope
     */
    public static final String SCOPE =
        "email,direct_messages_read,direct_messages_write,"
        + "friendships_groups_read,friendships_groups_write,statuses_to_me_read,"
        + "follow_app_official_microblog," + "invitation_write";
}
```

2 创建微博授权类对象

在 WBAuthActivity 中，创建微博授权类对象：

```
// 创建微博实例
mWeiboAuth = new WeiboAuth(this, Constants.APP_KEY,
                             Constants.REDIRECT_URL, Constants.SCOPE);
```

3 实现 WeiboAuthListener 接口

授权成功后，SDK 会将 access_token、expires_in、uid 等通过 Bundle 的形式返还给应用，在 onComplete 函数中，我们可以获取该信息。具体如何保存和 Token 信息由开发者自行处理。

注意：当您注册的应用程序签名不正确时，SDK 会将 Code 返还给用户，请确保应用的签名是正确的。

```
/**
 * 微博认证授权回调类。
 * 1.SSO 授权时，需要在 {@link #onActivityResult} 中调用 {@link SsoHandler#authorizeCallBack}
 * 后，该回调才会被执行。
 * 2. 非 SSO 授权时，当授权结束后，该回调就会被执行。
 * 当授权成功后，请保存该 access_token、expires_in、uid 等信息到 SharedPreferences 中。
 */
class AuthListener implements WeiboAuthListener {

    @Override
    public void onComplete(Bundle values) {
        // 从 Bundle 中解析 Token
        mAccessToken = OAuth2AccessToken.parseAccessToken(values);
        if (mAccessToken.isSessionValid()) {
            // 显示 Token
            updateTokenView(false);
            // 保存 Token 到 SharedPreferences
            AccessTokenKeeper.writeAccessToken(WBAuthActivity.this, mAccessToken);
            .....
        } else {
            // 当您注册的应用程序签名不正确时，就会收到 Code，请确保签名正确
            String code = values.getString("code", "");
            .....
        }
    }
    .....
}
```

4 调用方法，认证授权

- OAuth2.0 Web 授权：直接调用 WeiboAuth#authorize 方法即可。授权成功后即可获得 Token。

```
// 通过应用签名信息获取 Token
@Override
public void onClick(View v) {
    mWeiboAuth.authorize(new AuthListener());
    // 或者使用: mWeiboAuth.authorize(new AuthListener(), Weibo.OBTAIN_AUTH_TOKEN);
}
```

- SSO 登录授权：对于 SSO 授权，我们还需要以下步骤才能正确授权。

- 创建 SsoHandler 对象
- 重写 Activity#onActivityResult 方法，调用 SsoHandler# authorizeCallBack
- 调用 SsoHandler# authorize 方法

代码如下：

```
// 通过单点登录 (SSO) 获取 Token
@Override
public void onClick(View v) {
    mSsoHandler = new SsoHandler(WBAuthActivity.this, mWeiboAuth);
    mSsoHandler.authorize(new AuthListener());
}
```

```
/**
 * 当 SSO 授权 Activity 退出时，该函数被调用。
 *
 * @see {@link Activity#onActivityResult}
 */
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // SSO 授权回调
    // 重要：发起 SSO 登陆的 Activity 必须重写 onActivityResult
    if (mSsoHandler != null) {
        mSsoHandler.authorizeCallBack(requestCode, resultCode, data);
    }
}
```

七、 示例分析：分享微博

目前，通过微博 SDK，第三方应用能够分享文字、图片、视频、音乐等内容，目前分享有两种方式。

- 通过第三方应用唤起微博客户端进行分享（该分享方式为第三方客户端通常的使用方式）
- 通过微博客户端唤起第三方应用进行分享（该分享方式需要合作接入，详情请查看：<http://t.cn/aex4JF>）

接下来，我们将——描述微博分享的流程。

1. 认识 IWeiboShareAPI 接口

该接口是微博分享的主要接口类，其内部定义了两种分享方式的接口行为以及处理函数。

接口的详细定义请参考：**WeiboSDK_API-V2.4.0.CHM**

```
/** 检查用户是否安装了微博客户端程序 */
public boolean isWeiboAppInstalled();

/** 检查微博是否支持此 SDK。*/
public boolean isWeiboAppSupportAPI();

/** 获取当前微博客户端程序支持的 SDK 的最高版本号。*/
public int getWeiboAppSupportAPI();

/** 注册第三方应用到微博客户端中，注册成功后该应用将显示在微博的应用列表中（需要合作接入）*/
public boolean registerApp();

/** 处理微博客户端分享后的响应数据。*/
public boolean handleWeiboResponse(Intent intent, IWeiboHandler.Response handler);

/** 处理微博客户端发送过来的请求。*/
public boolean handleWeiboRequest(Intent intent, IWeiboHandler.Request handler);

/**
 * 发送微博分享请求给微博客户端程序。
 * 默认情况下，如果未安装微博客户端，在分享时，会弹出对话框询问用户是否下载微博客户端。
 */
public boolean sendRequest(BaseRequest request);

/** 收到微博客户端程序的请求后，发送对应的应答内容给微博客户端程序。*/
public boolean sendResponse(BaseResponse response);

/** 注册微博客户端下载的监听器。*/
public void registerWeiboDownloadListener(IWeiboDownloadListener listener);

/** 检查微博客户端环境是否正常。*/
public boolean checkEnvironment(boolean bShowDownloadDialog)
    throws WeiboShareException;
```

2. 从第三方应用唤起微博客户端进行分享

在进行微博分享前，我们需要在 AndroidManifest.xml 中，在需要接收消息的 Activity (唤起微博主程序的类) 里声明对应的 Action : ACTION_SDK_REQ_ACTIVITY，如下所示：

```
<activity
    android:name="com.sina.weibo.sdk.demo.WBShareActivity"
    android:configChanges="keyboardHidden|orientation"
    android:screenOrientation="portrait" >
    <intent-filter>
        <action android:name="com.sina.weibo.sdk.action.ACTION_SDK_REQ_ACTIVITY" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

然后在对应的 Activity 类，我们需要按以下步骤来实现分享：

1. 在 onCreate 函数创建微博分享接口实例

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    .....
    // 创建微博分享接口实例
    mWeiboShareAPI = WeiboShareSDK.createWeiboAPI(this, Constants.APP_KEY);
    // 如果未安装微博客户端，设置下载微博对应的回调
    if (!mWeiboShareAPI.isWeiboAppInstalled()) {
        mWeiboShareAPI.registerWeiboDownloadListener(new IWeiboDownloadListener() {
            @Override
            public void onCancel() {
                Toast.makeText(WBShareActivity.this,
                    R.string.weibosdk_demo_cancel_download_weibo,
                    Toast.LENGTH_SHORT).show();
            }
        });
    }
    // 当 Activity 被重新初始化时（该 Activity 处于后台时，可能会由于内存不足被杀掉了），
    // 需要调用 {@link IWeiboShareAPI#handleWeiboResponse} 来接收微博客户端返回的数据。
    // 执行成功，返回 true，并调用 {@link IWeiboHandler.Response#onResponse};
    // 失败返回 false，不调用上述回调
    if (savedInstanceState != null) {
        mWeiboShareAPI.handleWeiboResponse(getIntent(), this);
    }
}
```

2. 在分享前，调用 IWeiboShareAPI#registerApp 将应用注册到微博客户端

```
// 检查微博客户端环境是否正常，如果未安装微博，弹出对话框询问用户下载微博客户端
if (mWeiboShareAPI.checkEnvironment(true)) {

    // 注册第三方应用 到微博客户端中，注册成功后该应用将显示在微博的应用列表中。
    // 但该附件栏集成分享权限需要合作申请，详情请查看 Demo 提示
    mWeiboShareAPI.registerApp();
}
```

3. 创建要分享的内容

```
/**
 * 创建文本消息对象。
 *
 * @return 文本消息对象。
 */
private TextObject getTextObj() {
    TextObject textObject = new TextObject();
    textObject.text = getSharedText();
    return textObject;
}
```

4. 通过 IWeiboShareAPI#sendRequest 唤起微博客户端发博器进行分享

```
private void sendMultiMessage(boolean hasText, boolean hasImage, boolean hasWebpage,
    boolean hasMusic, boolean hasVideo, boolean hasVoice) {
    // 1. 初始化微博的分享消息
    WeiboMultiMessage weiboMessage = new WeiboMultiMessage();
    if (hasText) {
        weiboMessage.textObject = getTextObj();
    }
    .....
    // 2. 初始化从第三方到微博的消息请求
    SendMultiMessageToWeiboRequest request = new SendMultiMessageToWeiboRequest();
    // 用transaction唯一标识一个请求
    request.transaction = String.valueOf(System.currentTimeMillis());
    request.multiMessage = weiboMessage;

    // 3. 发送请求消息到微博，唤起微博分享界面
    mWeiboShareAPI.sendRequest(request);
}
```


5. 实现 IWeiboHandler#Response 接口，接收分享后微博返回的数据

```

protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);

    // 从当前应用唤起微博并进行分享后，返回到当前应用时，需要在此处调用该函数
    // 来接收微博客户端返回的数据；执行成功，返回 true，并调用
    // {@link IWeiboHandler.Response#onResponse}；失败返回 false，不调用上述回调
    mWeiboShareAPI.handleWeiboResponse(intent, this);
}

/**
 * 接收微客户端博请求的数据。
 * 当微博客户端唤起当前应用并进行分享时，该方法被调用。
 *
 * @param baseRequest 微博请求数据对象
 * @see {@link IWeiboShareAPI#handleWeiboRequest}
 */
@Override
public void onResponse(BaseResponse baseResp) {
    switch (baseResp.errCode) {
        case WBConstants.ErrorCode.ERR_OK: .....
            break;
        case WBConstants.ErrorCode.ERR_CANCEL: .....
            break;
        case WBConstants.ErrorCode.ERR_FAIL: .....
            break;
    }
}

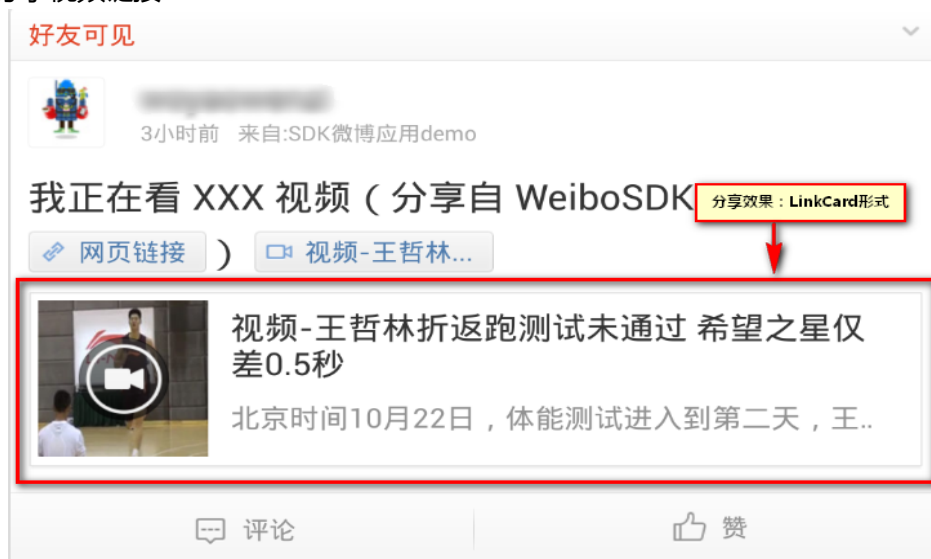
```

微博分享成功后，我们可以打开微博客户端查看效果，如下图所示：

☞ 分享新闻链接：



分享视频链接：



请注意：

1. 微博 LinkCard 效果

上图分享视频所示效果是 LinkCard 效果，分享普通链接是没有这种效果的。如果你想要你分享出去的内容展示成 LinkCard 的样子，其前提条件是你分享的链接支持 LinkCard。即 *LinkCard 的分享开关，是基于分享出来的链接的域名*。打个比方，开启 v.youku.com 的 LinkCard 开关，从而发布器中带这个链接的，都能呈现为 LinkCard，该功能并不是针对于某个特定的应用，而是针对于特殊的网内容的。如果某个网站需要进行 LinkCard 商务合作，请联系 BD。详情请阅读：移动客户端接入（<http://t.cn/aex4JF>）

2. 附件栏集成分享

关于第三方应用注册到微博客户端效果，如下图所示：

如果优质的第三方应用也想被显示在此处，也需要进行商务合作，默认情况下是不被显示在此处的。详情请阅读：移动客户端接入（<http://t.cn/aex4JF>）



3. 从微博客户端唤起第三方应用进行分享

通过微博客户端进行分享，即上面所提到的**附件栏集成分享**，如果您没有进行过商务合作，是无法进行该分享的。通过微博唤起第三方应用进行分享前，我们同样需要在 AndroidManifest.xml 中，在需要接收消息的 Activity（被微博唤起的程序的类）里声明对应的 Action：ACTION_SDK_RESP_ACTIVITY，如下所示：

```
<activity
    android:name=" com.sina.weibo.sdk.demo.WBShareResponseActivity"
    android:configChanges="keyboardHidden|orientation"
    android:screenOrientation="portrait" >
    <intent-filter>
        <action android:name="com.sina.weibo.sdk.action. ACTION_SDK_RESP_ACTIVITY" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

然后在对应的 Activity 类，我们需要按以下几步来实现分享（该分享过程和第三方应用唤起微博客户端进行分享的过程，其原理基本相差不大，实现代码也基本类似，请仔细区分一下）：

1. 微博客户端唤起第三方应用（该过程属于微博客户端行为，无示例代码）
2. 在 onCreate 函数创建微博分享接口实例

```
protected void onCreate(Bundle savedInstanceState) {
    .....
    // 创建微博分享接口实例
    mShareWeiboAPI = WeiboShareSDK.createWeiboAPI(this, Constants.APP_KEY);

    // 处理微博客户端发送过来的请求
    mShareWeiboAPI.handleWeiboRequest(getIntent(), this);
}
```

3. 实现 IWeiboHandler#Request 接口，接收并处理微博客户端发送过来的请求

```

@Override
protected void onNewIntent(Intent intent) {
    .....
    // 处理微博客户端发送过来的请求
    mShareWeiboAPI.handleWeiboRequest(intent, this);
}

/**
 * 接收微客户端博请求的数据。
 * 当微博客户端唤起当前应用并进行分享时，该方法被调用。
 *
 * @param baseRequest 微博请求数据对象
 * @see {@link IWeiboShareAPI#handleWeiboRequest}
 */
@Override
public void onRequest(BaseRequest baseRequest) {
    // 保存从微博客户端唤起第三方应用时，客户端发送过来的请求数据对象
    mBaseRequest = baseRequest;
    .....
}

```

4. 创建要分享的内容

```

/**
 * 创建文本消息对象。
 */
private TextObject getTextObj() {
    TextObject textObject = new TextObject();
    textObject.text = getSharedText();
    return textObject;
}

```

5. 通过 IWeiboShareAPI#sendResponse 发送应答数据给微博客户端

```
private void responseSingleMessage(boolean hasText, boolean hasImage,
    boolean hasWebpage, boolean hasMusic, boolean hasVideo) {

    // 1. 初始化微博的分享消息
    // 用户可以分享文本、图片、网页、音乐、视频中的一种
    WeiboMessage weiboMessage = new WeiboMessage();
    if (hasText) {
        weiboMessage.mediaObject = getTextObj();
    }
    .....

    // 2. 初始化从微博到第三方的消息请求
    ProvideMessageForWeiboResponse response = new ProvideMessageForWeiboResponse();
    response.transaction = mBaseRequest.transaction;
    response.reqPackageName = mBaseRequest.packageName;
    response.message = weiboMessage;

    // 3. 发送响应消息到微博
    mShareWeiboAPI.sendResponse(response);
}
```

八、 示例分析：登录/注销按钮

微博 SDK 目前提供了两类登录按钮：一种是一键登陆按钮，一种是登陆/注销按钮。

两者实际上都是调用了 SSO 登录接口。另外，不再单独提供注销按钮，用户可以调用已封装好的 OpenAPI 接口直接注销。

注意：两种类型的按钮目前都放在 *WeiboSDK* 工程中，第三方可根据需要选择使用何种按钮，可以对按钮进行相关修改，如大小和样式等，也可以将不需要的代码和图片去掉，减小依赖 JAR 包的大小。

1. 一键登录按钮

该按钮提目前提供了以下样式，如下图：



其中，第一种为默认样式。对于第三种样式，目前不提供按下的效果。

使用步骤如下：

1. 在需要集成的 Activity 的布局文件中，添加按钮：

```
<!-- 默认效果，带图片文字的登陆按钮 -->
<com.sina.weibo.sdk.widget.LoginButton
    android:id="@+id/login_button_default"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/com_sina_weibo_sdk_login_button_with_account_text" />
```

用户可根据需要，将 drawable 替换成其它两种：

```
@drawable/com_sina_weibo_sdk_login_button_with_frame_logo
@drawable/com_sina_weibo_sdk_login_button_with_original_logo
```

2. 在对应的 Activity 中，为该控件设置授权认证信息和 listener。

请注意：该 Listener 即为 *WeiboAuthListener*，请参考：[示例分析：授权认证](#)

```
// 创建授权认证信息
AuthInfo authInfo = new AuthInfo(this, Constants.APP_KEY, Constants.REDIRECT_URL,
    Constants.SCOPE);

// 为按钮设置授权认证信息
mLoginBtnDefault = (LoginButton) findViewById(R.id.login_button_default);
mLoginBtnDefault.setWeiboAuthInfo(authInfo, mLoginListener);
```

3. 当用户点击该按钮时，会进行 SSO 登陆，登陆完成后，返回该 Activity，此时，我们需要在 Activity#onActivityResult 中调用 LoginButton#authorizeCallBack 函数，整个登陆才能结束。授权成功后，用户可自行保存自己的 Token。

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (mLoginBtnDefault != null) {
        mLoginBtnDefault.onActivityResult(requestCode, resultCode, data);
    }
}
```

2. 登录/注销按钮

该按钮目前提供了以下样式，蓝色和银白色，如下图：



其中，第一种为默认样式，第三方开发者可自行修改。

使用步骤如下：

1. 在需要集成的 Activity 的布局文件中，添加按钮：

```
<!-- 默认效果，其 style 为: @style/com_sina_weibo_sdk_loginview_default_style -->
<com.sina.weibo.sdk.widget.LoginoutButton
    android:id="@+id/login_out_button_default"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="@style/com_sina_weibo_sdk_loginview_default_style" />
```

用户可根据需要，将为其添加 style 样式，目前有两种选：

```
@style/com_sina_weibo_sdk_loginview_default_style
@style/com_sina_weibo_sdk_loginview_silver_style
```

注意：该 style 文件定义在 *WeiboSDK/res/values/styles.xml* 下，第三方可根据需要修改或重写。

2. 在对应的 Activity 中，为该控件设置授权认证信息和 listener。与一键登录按钮完全一样，不再赘述。
3. 当用户点击该按钮时，会进行 SSO 登陆，其流程与一键登录按钮完全一样，不再赘述。

当登录完后，该按钮会变成注销，方便用户注销。如下图：



值得注意一点的是，如果该按钮已从当前 Activity 移除后（如 Activity 销毁后），下次再进入该 Activity 时，该按钮不会自动变成注销（由于 V2.4 SDK 未保存 Token 信息，以后可能会考虑保存所有 Session 相关信息），需要用户手动调用 `LoginLogoutButton# setLogoutInfo` 设置 Token 信息后，才会变成注销。

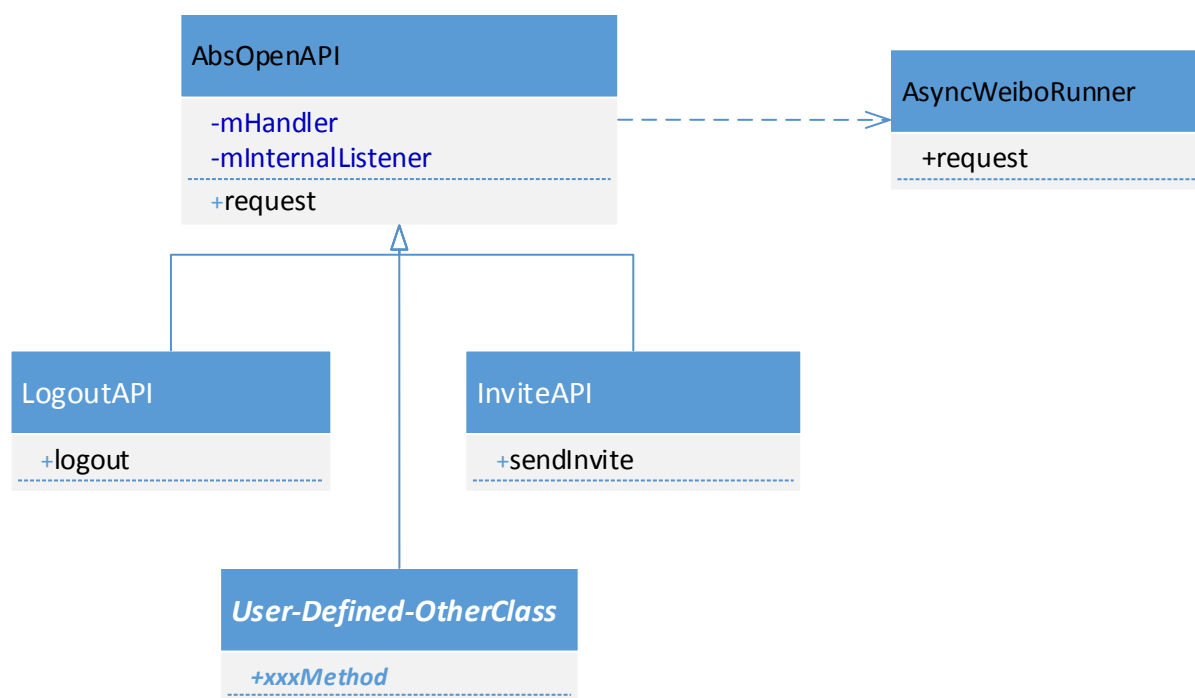
九、 示例分析：开放接口

微博 SDK 目前提供了一个简单的 OpenAPI 接口调用框架，并封装了一些简单的开放接口，以供大家参考：

- 注销接口：授权回收接口，帮助开发者主动取消用户的授权。
- 邀请接口：提供好友邀请接口，支持登录用户向自己的微博互粉好友发送私信邀请、礼物。

注：该模块目前还需要进一步设计和完善，后续将会提供一个更加合理好用的 OpenAPI 框架，并封装更多常用的接口，如利用 OpenAPI 获取用户信息，分享微博等。

目前的 OpenAPI 接口框架简单类图如下：



- 如上图所示，**AbsOpenAPI** 作为一个基类，里面提供了 `Handler` 以及异步请求回调接口 `RequestListener` 的对象。由于异步请求对应的回调是运行在后台线程中的，因此，为了方便第三方能够直接在 UI 线程中使用对应回调，我们设置了一个 `Handler`，将后台线程切换到 UI 线程。
- **LogoutAPI** 类以及 **InviteAPI** 类从 **AbsOpenAPI** 继承过来，用于注销和邀请好友。
- 第三方如果想要扩展接口，可以参照 **LogoutAPI** 或 **InviteAPI** 的实现方式。

注意：更多细节请看源代码。值得说明一点的是，包 `com.sina.weibo.sdk.openapi.legacy` 下所有的类均为 V2.0 版本时期的代码，目前没有经过严格的验证，第三方可以参考实现自己想要的功能，目前仅供参考，新浪微博暂时不提供过多的技术支持，因为后期我们会逐一进行验证，以更好的方式提供给开发者。

1. 注销接口

授权回收接口，可以帮助开发者主动取消用户的授权，即将 Token 回收，回收后，该 Token 不能被再次使用，如果用户想再次使用，必须重新获取 Token。

集成该接口非常简单，其使用方式如下：

1. 实现 RequestListener 接口。

```
/**
 * 注销按钮的监听器，接收注销处理结果。（API请求结果的监听器）
 */
private class LogOutRequestListener implements RequestListener {
    @Override
    public void onComplete(String response) {
        if (!TextUtils.isEmpty(response)) {
            try {
                JSONObject obj = new JSONObject(response);
                String value = obj.getString("result");

                if ("true".equalsIgnoreCase(value)) {
                    AccessTokenKeeper.clear(WBLogoutAPIActivity.this);
                    mTokenView.setText(R.string.weibosdk_demo_logout_success);
                    mAccessToken = null;
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        .....
    }
}
```

2. 创建 LogoutAPI 对象，获取 Token 信息，并调用 LogoutAPI#logout 函数进行注销。

```
// 获取当前已保存过的 Token
mAccessToken = AccessTokenKeeper.readAccessToken(this);

.....

// 注销该Token
public void onClick(View v) {
    if (mAccessToken != null && mAccessToken.isSessionValid()) {
        new LogoutAPI(mAccessToken).logout(mLogoutRequestListener);
    } else {
        mTokenView.setText(R.string.weibosdk_demo_logout_failed_1);
    }
}
```

2. 好友邀请接口

该接口支持登录用户向自己的微博互粉好友发送私信邀请、礼物。

该接口的详细内容可参见：<http://open.weibo.com/wiki/2/messages/invite>

集成该接口步骤也是非常简单，同注销 Token 一样：

1. 实现 RequestListener 接口（同上）
2. 创建邀请接口参数

在初始化 InviteApi 类的实例时，需要设置以下内容：

- InviteAPI.KEY_TEXT (必须)：要回复的私信文本内容。文本大小必须小于 300 个汉字。
- InviteAPI.KEY_URL (可选)：邀请点击后跳转链接。默认为当前应用地址
- InviteAPI.KEY_INVITE_LOGO (可选)：邀请 Card 展示时的图标地址，大小必须为 80px X 80px，仅支持 PNG、JPG 格式。

```
// 创建请求的 JSON 数据对象
JSONObject jsonObject = new JSONObject();
try {
    jsonObject.put(InviteAPI.KEY_TEXT, "这个游戏太好玩了，加入一起玩吧");
    jsonObject.put(InviteAPI.KEY_URL,
        "http://app.sina.com.cn/appdetail.php?appID=770915");
    jsonObject.put(InviteAPI.KEY_INVITE_LOGO,
        "http://hubimage.com2us.com/hubweb/contents/123_499.jpg");
} catch (JSONException e) {
    e.printStackTrace();
}
}
```

3. 创建一个 InviteAPI 类的实例，并调用其 InviteAPI#sendInvite 方法即可，如下图：

```
// 获取 Token
OAuth2AccessToken token = AccessTokenKeeper.readAccessToken(WBInviteAPIActivity.this);
if (token != null && token.isSessionValid()) {
    // 调用 OpenAPI 接口发送邀请
    new InviteAPI(token).sendInvite(uid, jsonObject, mInviteRequestListener);
} else {
    .....
}
}
```

注意事项：

1. 应用引导用户使用微博登录 采用 OAuth2 的 Scope 授权方式获得用户“发送私信邀请给你的微博好友”权限，对应的 scope 权限参数为：invitation_write；
2. 每用户每小时通过每应用可邀请 10 个互粉好友；
3. 必须由用户主动发起。

具体请参考 Scope 权限说明：<http://open.weibo.com/wiki/Scope>

第三方如何如何申请接口权限？如下图：

该页面可以在新浪微博开放平台→管理中心→接口管理→申请权限处找到。



This is the last page