

# 微博Android SDK 4.0文档

全新的微博sdk更新了，如果你是sdk老用户，请在接入先阅读下面的变更逻辑，如果你是微博sdk新用户，请跳过这一步骤直接进入开始接入。

变更逻辑：

1.删除原来分享页面manifest中注册的intent-filter:

```
android:name=".sdk.share.WbShareTransActivity">
<intent-filter>
    <action android:name="com.sina.weibo.sdk.action.ACTION_SDK_REQ_ACTIVITY" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
</activity>
```

2.关闭了sdk原有的http请求入口，如果你使用了sdk内部的网络请求openApi,请切换成你们自己的网络框架进行请求。关键类（HttpManager,AsyncWeiboRunner）

3.添加了第三方客户端信息收集的统一入口，请在你程序的Application中添加：

```
WbSdk.install(Context context,AuthInfo authInfo)
```

4.删除IWeiboShareApi和相关操作，新的分享入口更换为WbShareHandler

5.修改微博授权的回调方式，授权结果更清晰。

如果你完成了上述操作，下来正式进入sdk的接入说明

开始接入：

## 1.1 认证授权

新浪微博Android 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进行分享。

本文档介绍了新浪微博Android SDK的三种授权方式，各种分享并给出简单的示例分析，帮助第三方开发者快速集成应用。

1.2 名词解释

2:集成前的准备

2.1 申请APP\_KEY

	注解
AppKey	分配给每个第三方应用的唯一身份认证，用来区分来源等功能（获取方式： <a href="http://open.weibo.com/development/mobile">http://open.weibo.com/development/mobile</a> ）
Scope	Scope是OAuth2.0新版授权页的一个功能，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	表明用户身份的Token，通过SSO授权后可以获取
Oauth 2.0 Web授权	通过WebView进行授权，返回Token信息，无需安装微博客户端
SSO授权	通过唤起微博客户端进行身份授权，返回Token信息

第三方需要接入微博SDK必须在微博开放平台上对应用进行注册，并获取APP\_KEY,添加应用的授权回调页面（Redirect URI），详情请参考：[微博移动接入平台](#)

2.2注册程序包名和签名

集成SDK前，需要在微博开放平台上注册应用的包名和签名。

注意：包名和签名未注册，或者编译运行时的签名和注册签名不一致都可能导致无法授权(debug运行apk和发布时的apk)。

应用包名获取方式：工程主modules目录下build.gradle中的applicationId的值，或者AndroidManifest.xml中package节点的数据。

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（按照下图所示）

下载签名软件：[下载地址](#)

使用方式：首先要安装你已经签名的应用，然后再安装该工具，安装后，输入应用的包名，点击生成按钮，既可以获取MD5签名，如下图所示。

注意：要签名的应用程序必须安装到该设备上才能生成对应的MD5签名。如果你在开发平台上只填写的是发布版本的签名MD5,请确保你平时debug版本的签名和发布版本签名一致，否则授权不能通过。

\*\*\*\*\*



sdk授权和分享必须保证 appKey、scope、redirectUrl、packageName四个参数和开发这官网填写一致，否则不能授权分享成功

redirect\_rui:在设置\_高级设置里面填写

## 2接入方式

全新的SDK已经上传到中央仓库

这里按照Android studio为例子，在项目根目录的build.gradle中设置中央仓库

```
maven { url "https://dl.bintray.com/thelasterstar/maven/" }
```

在需要引入SDK的module目录的build.gradle中引入sdk-core依赖

```
dependencies {  
    compile 'com.sina.weibo.sdk:core:2.0.1:openDefaultRelease@aar'  
}
```



```

buildscript {
    repositories {
        jcenter()
        mavenCentral()
        maven { url "https://dl.bintray.com/thelasterstar/maven/" }
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.5.0'
        classpath 'com.jfrog.bintray.gradle:gradle-bintray-plugin:1.4'
        classpath 'com.github.dcendents:android-maven-gradle-plugin:1.3'
    }
}

allprojects {
    repositories {
        jcenter()
        mavenCentral()
        maven { url "https://dl.bintray.com/thelasterstar/maven/" }
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}

dependencies {
    compile 'org.jbundle.util.osgi.wrapped:org.jbundle.util.osgi.wrapped.org.apache.http.client:4.1.2'
    compile 'com.sina.weibo.sdk:openDefault:1.0.0:openDefaultRelease@aar'
}

```

点击同步按钮，等待SDK库下载完成。

也可以直接在github上下载微博 SDK demo，参考demo接入方案。

## 示例代码分析

### 1:认证授权

1)首先初始化WbSdk对象(在你的应用的Application或者调用Sdk功能代码前)

WbSdk.install(this,new AuthInfo(this, Constants.APP\_KEY, Constants.REDIRECT\_URL, Constants.SCOPE));

AuthInfo维护了授权需要的基本信息，APP\_KEY(开发平台生成的唯一key)、Redirect URI (授权回调)、SCOPE (需要请求的权限功能，默认参考demo中数据)。

2)初始化SsoHandler对象

mSsoHandler = new SsoHandler(WBAuthActivity.this);

SsoHandler是发起授权的核心类。

2) 调用授权

SDK中有三种模式的授权方案：

- 1: AuthorizeClientSso:只通过微博客户端进行授权
- 2: AuthorizeWeb:通过SDK自带的WebView打开H5页面进行授权
- 3: Authorize:如果安装了微博客户端通过客户端授权，如果没有通过Web方式授权

```
// SSO 授权, 仅客户端
findViewById(R.id.obtain_token_via_sso).setOnClickListener((v) -> {
    mSsoHandler.authorizeClientSso(new SelfWbAuthListener());
});

// SSO 授权, 仅Web
findViewById(R.id.obtain_token_via_web).setOnClickListener((v) -> {
    mSsoHandler.authorizeWeb(new SelfWbAuthListener());
});

// SSO 授权, ALL IN ONE 如果手机安装了微博客户端则使用客户端授权, 没有则进行网页授权
findViewById(R.id.obtain_token_via_signature).setOnClickListener((v) -> {
    mSsoHandler.authorize(new SelfWbAuthListener());
});
```

调用授权登录:

SelfWbAuthListener: 授权结果回调, 实现WbAuthListener接口

```
public interface WbAuthListener {
    void onSuccess(Oauth2AccessToken token);
    void cancel();
    void onFailure(WbConnectErrorMessage errorMessage);
}
```

onSuccess: 授权成功回调, 返回Oauth2AccessToken对象, 封装了登录相关信息

onFailure: 授权失败回调(更多相关失败信息, 请参考文档结尾错误码)

cancel: 授权取消回调。

注意: 要接收到授权的相关数据, 必须在当前页面Activity或者Fragment的onActivityResult方法中添加SSOHandler的调用, 如下所示

```
@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);
    }
}
```

### 3 微博分享

#### 3.1 从第三方应用唤起微博客户端进行分享

##### 3.1.1 准备工作:

1: 在分享当前页面的Activity实现WbShareCallback接口, 效果如下:

```

    */
    public class WbShareActivity extends Activity implements OnClickListener, WbShareCallback {
        public static final String KEY_SHARE_TYPE = "key_share_type";
        public static final int SHARE_CLIENT = 1;
        public static final int SHARE_ALL_IN_ONE = 2;
    }

```

WbShareCallback接口实现了分享结果的各种回调，如图：

```

@Override
public void onWbShareSuccess() {
    Toast.makeText(this, "分享成功", Toast.LENGTH_LONG).show();
}

@Override
public void onWbShareFail() {
    Toast.makeText(this,
        "分享失败" + "Error Message: ",
        Toast.LENGTH_LONG).show();
}

@Override
public void onWbShareCancel() {
    Toast.makeText(this, "取消分享", Toast.LENGTH_LONG).show();
}

```

2：初始化WbShareHandler，并注册应用

```

shareHandler = new WbShareHandler(this);
shareHandler.registerApp();

```

3：分享微博：

```

WeiboMultiMessage weiboMessage = new WeiboMultiMessage();
if (hasText) {
    weiboMessage.textObject = getTextObj();
}
if (hasImage) {
    weiboMessage.imageObject = getImageObj();
}
weiboMessage.mediaObject = getWebpageObj();
shareHandler.sendMessage(weiboMessage, false);

```

4：处理分享回调：重写onNewIntent方法

```

@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    shareHandler.doResultIntent(intent, this);
}

```

## 以上四步少一步都可能会导致分享无法实现

\*\*\*\*\*以下是接入时可能会导致无法分享的一些说明\*\*\*\*\*

由于老版本的微博客户端分享存在一定逻辑缺陷，在没有用户登录的状态下进行第一次分享时可能导致第三程序鉴定失败引起不能正常分享。这里特别说明下，在进行第一次分享的时候，可以先进行SSO授权登录，确保微博客户端处于账户登录状态，然后进行分享。

## 关于LinkCard

LinkCard已经在微博客户端中移除，如果你想从sdk中分享出card样式的微博在信息流中，请和微博商务洽谈合作，实现特有的card样式。

\*：具体想知道如接入weibo sdk请参考weibo sdk 官方github地址中的demo:

[https://github.com/sinaweibosdk/weibo\\_android\\_sdk](https://github.com/sinaweibosdk/weibo_android_sdk)

错误code集合：

code码	解释
C8998	开放平台填写的签名和程序签名不一致，检查下开放平台填写的是发布签名，运行程序为debug签名。
8000	没有安装微博客户端
8001	安装的微博客户端为非官方客户端