

微博 Android SDK使用指南

[申请你的App Key](#)[注册应用程序的包名和签名](#)[集成SDK](#)[参数配置](#)[初始化SDK](#)[使用微博授权](#)[分享内容到微博](#)

申请你的App Key

请到微博开放平台上 [我的应用](#) 页面，通过开发者身份认证审核，完善应用信息，将该应用提交审核，只有审核通过的应用才能进行开发。

注册应用程序的包名和签名

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授权。



The screenshot shows a registration form for Android package names and signatures. It includes two radio buttons for selection: '当前页面录入 (当小于3套包名签名信息时)' (selected) and '导入含多套签名包名的CSV文件 (当大于3套, 最多不超过500套包名签名信息时)'. Below are three sets of input fields for 'Android包名' and 'Android签名', each with a link to '通过下载使用平台提供的签名工具获取'. At the bottom is a field for 'Android下载地址'.

注：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。

集成SDK

Android Studio环境下：

在Project的build.gradle文件中添加依赖配置：

```
allprojects {
    repositories {
        maven { url 'https://dl.bintray.com/thelasterstar/maven/' }
    }
}
```

在Module的build.gradle文件中添加依赖和属性配置：

```
android {
    defaultConfig {
        ndk {
            // 设置支持的SO库架构
            abiFilters 'armeabi' //, 'armeabi-v7a', 'arm64-v8a'
        }
    }
}

dependencies {
    compile 'com.sina.weibo.sdk:core:9.12.0:openDefaultRelease'
}
```

参数配置

- 在AndroidManifest.xml中添加权限：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL"
```

注：如果您的App需要上传到 google play store，您需要将 READ_PHONE_STATE 权限屏蔽掉或者移除，否则可能会被下架。

- 请避免混淆微博SDK，在Proguard混淆文件中增加以下配置：

```
-keep public class com.sina.weibo.sdk.**{*;} 
```

初始化SDK

- 初始化代码如下：

```
private static final String APP_KEY = "在微博开发平台为应用申请的AppKey";
private static final String REDIRECT_URL = "在微博开放平台设置的授权回调地址";
private static final String SCOPE = "在微博开放平台为应用申请的高级权限";
private IWBAPI mWBAPI;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //init sdk
    initSdk();
}

private void initSdk() {
    AuthInfo authInfo = new AuthInfo(this, APP_KEY, REDIRECT_URL, SCOPE);
    mWBAPI = WBAPIFactory.createWBAPI(this);
    mWBAPI.registerApp(this, authInfo);
}
```

使用微博授权

- 微博授权示例代码如下：

```
//授权操作
private void startAuth() {
    //auth
    mWBAPI.authorize(new WbAuthListener() {
        @Override
        public void onComplete(OAuth2AccessToken token) {
            Toast.makeText(MainActivity.this, "微博授权成功", To
        }

        @Override
        public void onError(UiError error) {
            Toast.makeText(MainActivity.this, "微博授权出错", To
        }

        @Override
        public void onCancel() {
            Toast.makeText(MainActivity.this, "微博授权取消", To
        }
    });
}
```

```
//通过微博客户端授权操作
private void startClientAuth() {
    mWBAPI.authorizeClient(new WbAuthListener() {
        @Override
        public void onComplete(OAuth2AccessToken token) {
            Toast.makeText(MainActivity.this, "微博授权成功", To
        }

        @Override
        public void onError(UiError error) {
            Toast.makeText(MainActivity.this, "微博授权出错", To
        }

        @Override
        public void onCancel() {
```

```

        Toast.makeText(MainActivity.this, "微博授权取消", To
    }
    });
}

```

```

//通过网页（H5）授权操作
private void startWebAuth() {
    mWBAPI.authorizeWeb(new WbAuthListener() {
        @Override
        public void onComplete(0auth2AccessToken token) {
            Toast.makeText(MainActivity.this, "微博授权成功", To
        }

        @Override
        public void onError(UiError error) {
            Toast.makeText(MainActivity.this, "微博授权出错:" +
        }

        @Override
        public void onCancel() {
            Toast.makeText(MainActivity.this, "微博授权取消", To
        }
    });
}

```

- 重写Activity的onActivityResult方法，其代码如下：

```

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

```

注：这里是设置授权回调，如果不写，将不会收到授权结果。

分享内容到微博

- 实现回调，其代码如下：

```
// 实现WbShareCallback接口，获取分享结果。
private static class ShareCallback implements WbShareCallback
{
    @Override
    public void onComplete() {
        Toast.makeText(ShareActivity.this, "分享成功", Toast.LENGTH_
    }

    @Override
    public void onError(UiError error) {
        Toast.makeText(ShareActivity.this, "分享失败:" + error.error
    }

    @Override
    public void onCancel() {
        Toast.makeText(ShareActivity.this, "分享取消", Toast.LENGTH_
    }
}
```

- 重写Activity的onActivityResult方法设置回调，其代码如下：

```
@Override
protected void onActivityResult(int requestCode, int resultCode
{
    super.onActivityResult(requestCode, resultCode, data);
    if (mWBAPI != null) {
        mWBAPI.doResultIntent(data, new ShareCallback());
    }
}
```

- 微博分享示例代码如下：

```
//通过微博客户端授权操作
private void doWeiboShare() {
    WeiboMultiMessage message = new WeiboMultiMessage();

    TextObject textObject = new TextObject();
    String text = "我正在使用微博客户端发博器分享文字。";

    // 分享文字
    if (mShareText.isChecked()) {
        text = "这里设置您要分享的内容! ";
    }
    textObject.text = text;
    message.textObject = textObject;

    // 分享单图
    if (mShareImage.isChecked()) {
        ImageObject imageObject = new ImageObject();
        Bitmap bitmap = BitmapFactory.decodeResource(getResources()
            imageObject.setImageData(bitmap);
        message.imageObject = imageObject;
    }

    // 分享网页
    if (mShareUrl.isChecked()) {
        WebpageObject webObject = new WebpageObject();
        webObject.identify = UUID.randomUUID().toString();
        webObject.title = "标题";
        webObject.description = "描述";
        Bitmap bitmap = BitmapFactory.decodeResource(getResources()
        ByteArrayOutputStream os = null;
        try {
            os = new ByteArrayOutputStream();
            bitmap.compress(Bitmap.CompressFormat.JPEG, 85, os);
            webObject.thumbData = os.toByteArray();
        }
```

```
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        if (os != null) {
            os.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

webObject.actionUrl = "https://weibo.com";
webObject.defaultText = "分享网页";
message.mediaObject = webObject;
}

// 分享多图
if (mShareMultiImage.isChecked()) {
    MultiImageObject multiImageObject = new MultiImageObject
    List<Uri> list = new ArrayList<Uri>();
    list.add(Uri.fromFile(new File(getExternalFilesDir(null)
    list.add(Uri.fromFile(new File(getExternalFilesDir(null)
    list.add(Uri.fromFile(new File(getExternalFilesDir(null)
    list.add(Uri.fromFile(new File(getExternalFilesDir(null)
    list.add(Uri.fromFile(new File(getExternalFilesDir(null)
    list.add(Uri.fromFile(new File(getExternalFilesDir(null)
    list.add(Uri.fromFile(new File(getExternalFilesDir(null)
    list.add(Uri.fromFile(new File(getExternalFilesDir(null)
    multiImageObject.imageList = list;
    message.multiImageObject = multiImageObject;
}

// 分享视频
if (mShareVideo.isChecked()) {
    VideoSourceObject videoObject = new VideoSourceObject();
    videoObject.videoPath = Uri.fromFile(new File(getExternalFilesDir(null)
    message.videoSourceObject = videoObject;
}
```



```
// mShareClient.isChecked() 表示是否通过微博客户端分享。  
mWBAPI.shareMessage(message, mShareClient.isChecked());  
}
```

分享多图注：

- 设置的是本地图片文件的路径,并且是当前应用可以访问的路径，现在不支持网络路径。
- 接入程序必须有文件读写权限，否则会造成分享失败。
- 使用前通过**WbSdk.isSupportMultipleImage(context)**判断是否支持。

分享视频注：

- 设置的是本地视频文件的路径,并且是当前应用可以访问的路径，现在不支持网络路径。
- 目前支持视频格式为：mp4。

H5分享注：

- H5分享只支持分享文字和单图。
- 图片要求是合规图片。