# umeng集成社会化分享-说明文档

author：leobert.lan

注：使用集成的弊端比较多，尤其是版本问题，所以Android端会走单独的接入，不走集成。

以下内容作为半成品调研结果保留

## Pan\_android需求

支持QQ、微信（及微信朋友圈）、新浪微博、腾讯微博第三方分享。使用Umeng集成。

## 申请接入权限

使用UMeng相关功能需要Umeng接入权限，网盘移动端再使用Umeng统计时已经申请，相关信息如下：

* 开发者账号：

zhwwang@landhightech.com

* 密码：

LCloud160118

* Android 唯一Appkey为：

569c7b7767e58e0928000a28

## 下载并导入SDK

* sdk下载链接：http://dev.umeng.com/social/android/sdk-download
* 添加资源文件，对此我只能说：小心操作，避免覆盖
* 添加代码，相应的jar包导入工程。

具体可以参考demo

## AndroidManifest配置

### 其中增加一个activity

（这个如果和分享的界面有关，可能自定义时需要额外处理）

|  |
| --- |
| <activity  android:name="com.umeng.socialize.editorpage.ShareActivity"  android:theme="@style/Theme.UMDefault"  android:excludeFromRecents="true"/> |

### 设置友盟appid

（从未添加的请注意节点位置，使用umeng只需要配置一次即可，即使是不同功能，key是统一的）

|  |
| --- |
| <meta-data  android:name="UMENG\_APPKEY"  android:value="4eaee02c527015373b000003" >  </meta-data> |

### 增加权限

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

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

<uses-permission android:name="android.permission.READ\_LOGS" />

<uses-permission android:name="android.permission.CALL\_PHONE" />

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />

<uses-permission android:name="android.permission.GET\_TASKS" />

<uses-permission android:name="android.permission.SET\_DEBUG\_APP" />

<uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW" />

<uses-permission android:name="android.permission.GET\_ACCOUNTS" />

<uses-permission android:name="android.permission.USE\_CREDENTIALS" />

<uses-permission android:name="android.permission.MANAGE\_ACCOUNTS" />

### 针对不平台所需要添加的数据

我们需要添加以下平台，如需要增加，移步umeng sdk文档并下载额外的sdk

#### 微信平台

|  |
| --- |
| <activityandroid:name="com.umeng.soexample.wxapi.WXEntryActivity"android:configChanges="keyboardHidden|orientation|screenSize"android:exported="true"android:screenOrientation="portrait"android:theme="@android:style/Theme.Translucent.NoTitleBar" /> |

#### 新浪微博

|  |
| --- |
| <activityandroid:name=".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><activityandroid:name="com.sina.weibo.sdk.component.WeiboSdkBrowser"android:configChanges="keyboardHidden|orientation"android:windowSoftInputMode="adjustResize"android:exported="false" ></activity><serviceandroid:name="com.sina.weibo.sdk.net.DownloadService"android:exported="false"></service> |

#### QQ、Qzone

|  |
| --- |
| <activityandroid:name="com.tencent.tauth.AuthActivity"android:launchMode="singleTask"android:noHistory="true" ><intent-filter><action android:name="android.intent.action.VIEW" /><category android:name="android.intent.category.DEFAULT" /><category android:name="android.intent.category.BROWSABLE" /><data android:scheme="tencent100424468" /></intent-filter></activity><activityandroid:name="com.tencent.connect.common.AssistActivity"android:screenOrientation="portrait"android:theme="@android:style/Theme.Translucent.NoTitleBar" /> |

## 初始化配置

各个平台的配置，建议放在全局Application或者程序入口

|  |
| --- |
| PlatformConfig.setWeixin("wx967daebe835fbeac", "5bb696d9ccd75a38c8a0bfe0675559b3");  //微信 appid appsecret  PlatformConfig.setSinaWeibo("3921700954","04b48b094faeb16683c32669824ebdad");  //新浪微博 appkey appsecret  PlatformConfig.setQQZone("100424468", "c7394704798a158208a74ab60104f0ba");  // QQ和Qzone appid appkey  PlatformConfig.setAlipay("2015111700822536");  //支付宝 appid  PlatformConfig.setYixin("yxc0614e80c9304c11b0391514d09f13bf");  //易信 appkey  PlatformConfig.setRenren("201874","28401c0964f04a72a14c812d6132fcef","3bf66e42db1e4fa9829b955cc300b737");  //人人 appid appkey appsecret  PlatformConfig.setTwitter("3aIN7fuF685MuZ7jtXkQxalyi", "MK6FEYG63eWcpDFgRYw4w9puJhzDl0tyuqWjZ3M7XJuuG7mMbO");  //Twitter appid appkey  PlatformConfig.setPinterest("1439206");  //Pinterest appid  PlatformConfig.setLaiwang("laiwangd497e70d4", "d497e70d4c3e4efeab1381476bac4c5e");  //来往 appid appkey |

* 说明：有些平台的分享时通过调起本地客户端的形式完成的，因此，并不需要appid，但是诸如需要使用第三方公司api的平台，新浪，腾讯等，是需要去申请appkey的。其实新浪微博，人人网可以把appid等配置在代码中，也可在友盟后台中设置，豆瓣只能在友盟后台设置。其它需要配置appid的平台，如qq，微信，易信,twitter等都需要在代码中设置。还有一部分平台需要在mainfest中配置，前面已经提到过。

5. 第三方登录

目前友盟社会化组件支持的第三方登录平台为：新浪微博、腾讯微博、QQ、QQ空间、微信、人人网、豆瓣、Facebook、Twitter。

5.1 授权

首先获取UMShareAPI

mShareAPI = UMShareAPI.get(this);

选取需要授权的平台，并进行授权，其中umAuthLisrener是回调监听器，需要开发者根据需求重新定义

SHARE\_MEDIA platform = SHARE\_MEDIA.SINA;

mShareAPI.doOauthVerify(**this**, platform, umAuthListener)

**private** UMAuthListener umAuthListener = **new** UMAuthListener() {

@Override

**public** **void** **onComplete**(SHARE\_MEDIA platform, **int** action, Map<String, String> data) {

Toast.makeText( getApplicationContext(), "Authorize succeed", Toast.LENGTH\_SHORT).show();

}

@Override

**public** **void** **onError**(SHARE\_MEDIA platform, **int** action, Throwable t) {

Toast.makeText( getApplicationContext(), "Authorize fail", Toast.LENGTH\_SHORT).show();

}

@Override

**public** **void** **onCancel**(SHARE\_MEDIA platform, **int** action) {

Toast.makeText( getApplicationContext(), "Authorize cancel", Toast.LENGTH\_SHORT).show();

}

};

对于删除授权使用的接口是

mShareAPI.deleteOauth(AuthActivity.this, platform, umdelAuthListener);

不难看出与授权的参数是一样的。

注意要重写

onActivityResult()

@Override

**protected** **void** **onActivityResult**(**int** requestCode, **int** resultCode, Intent data) {

**super**.onActivityResult(requestCode, resultCode, data);

mShareAPI.onActivityResult(requestCode, resultCode, data);

}

**5.1.1 获取客户端安装信息**

使用如下接口

mShareAPI.isInstall(this, SHARE\_MEDIA.WEIXIN)

5.2 获取用户信息和好友列表

已授权的平台，可以获取用户信息（新浪微博可以获取用户好友列表） 实现的方法与授权和解除授权类似：

mShareAPI = UMShareAPI.get(this);

初始化UMShareAPI，然后进行用户信息获取：

mShareAPI.getPlatformInfo(UserinfoActivity.this, platform, umAuthListener);

对于新浪微博好友列表的获取使用的接口是：

mShareAPI.getFriend(UserinfoActivity.this, SHARE\_MEDIA.SINA, umGetfriendListener);

注意要重写

onActivityResult()

@Override

**protected** **void** **onActivityResult**(**int** requestCode, **int** resultCode, Intent data) {

**super**.onActivityResult(requestCode, resultCode, data);

mShareAPI.onActivityResult(requestCode, resultCode, data);

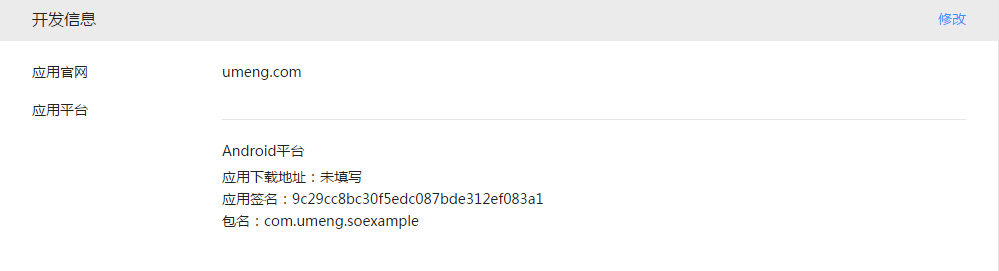
}

5.3 微信登录

实现方法参照上面所说的授权及获取用户信息方法

注意以下几点

**5.3.1 签名设置**

注意打包后的apk文件签名必须与[微信平台](https://open.weixin.qq.com/)设置一致，微信平台签名设置位置如下图 

apk签名可以直接安装[微信签名工具](http://dev.umeng.com/system/resources/W1siZiIsIjIwMTUvMTIvMjEvMTRfMDJfMTZfMzA4X18uYXBrIl1d/%E8%8E%B7%E5%8F%96%E7%AD%BE%E5%90%8D.apk)，获取签名后与上图所示位置比对，如果不一致，则无法进入微信授权确认页面

**5.3.2 回调类的实现**

在AndroidManifest.xml中下注册下面的回调Activity。

<activity

android:name=".wxapi.WXEntryActivity"

android:theme="@android:style/Theme.Translucent.NoTitleBar"

android:configChanges="keyboardHidden|orientation|screenSize"

android:exported="true"

android:screenOrientation="portrait" />

然后将SDK包中platforms/weixin目录下的wxapi文件夹拷贝到您AndroidMainFest.xml文件application标签里配置的包目录下,然后修改WXEntryActivity的完整路径即可

* 以我们的Demo工程为例

social\_sdk\_example的包名为com.umeng.soexample,因此将wxapi文件夹拷贝到com.umeng.soexample下,最终WXEntryActivity的完整路径为com.umeng.soexample.wxapi.WXEntryActivity

**注意WXEntryActivity.java的完整路径必须正确，否则微信登录会无法返回相关用户数据，也无法正常执行回调**

5.4 QQ及Qzone登录

实现方法同样如上，注意下面几点：

* QQ及Qzone登录都必须安装QQ客户端，并且只能跳转QQ客户端实现登录，无法跳转到Qzone客户端
* QQ及Qzone登录需要确保在[腾讯平台](https://open.qq.com/)通过审核或添加测试账号，否则会报错110406
* QQ及Qzone登录必须确保apk签名与腾讯平台上传的apk签名匹配，否则会报错100044，出现此错误请联系腾讯平台人员（QQ：800013811 ）修改apk签名

注意将Android manifest xml里的 腾讯平台APPID替换为自己申请的APPID，注意tencent前缀不要修改，否则无法正常实现QQ及Qzone登录

* 平台授权，是需要安装对应的客户端的。新浪微博，豆瓣网，人人网除外，新浪和人人如果本地安装了对应平台是走本地授权，如果没有安装，需要走网络授权

6. 分享

分享分为两种样式，使用默认分享面板选择页面和自定义分享样式

6.1 默认shareboard：

**final** SHARE\_MEDIA[] displaylist = **new** SHARE\_MEDIA[]

{

SHARE\_MEDIA.WEIXIN, SHARE\_MEDIA.WEIXIN\_CIRCLE,SHARE\_MEDIA.SINA,

SHARE\_MEDIA.QQ, SHARE\_MEDIA.QZONE,SHARE\_MEDIA.DOUBAN

};

**new** ShareAction(**this**).setDisplayList( displaylist )

.withText( "呵呵" )

.withTitle("title")

.withTargetUrl("http://www.baidu.com")

.withMedia( image )

.setListenerList(umShareListener,umShareListener)

.setShareboardclickCallback(shareBoardlistener)

.open();

* title参数对新浪、人人、豆瓣不生效

修改默认shareboard中的分享平台，只需要传入自己需要的平台参数，放入shareaction中，并执行open方法即可。 shareboard点击响应回调方法如下：

**private** ShareBoardlistener shareBoardlistener = **new** ShareBoardlistener() {

@Override

**public** **void** **onclick**(SHARE\_MEDIA share\_media) {

}

};

6.2 自定义分享样式

对于需要自己定义分享UI样式的开发者，可以自行绘制分享模块的整个UI，在对应的分享按钮中实现下面的方法

**new** ShareAction(**this**)

.setPlatform(SHARE\_MEDIA.SINA)

.setCallback(umShareListener)

.withText("hello umeng video")

.withTargetUrl("http://www.baidu.com")

.withMedia(image)

.share();

选取分享的平台，设定分享回调接口和内容，并执行share方法。 回调接口如下：

**new** UMShareListener() {

@Override

**public** **void** **onResult**(SHARE\_MEDIA platform) {

Toast.makeText(ShareActivity.**this**,platform + " 分享成功啦", Toast.LENGTH\_SHORT).show();

}

@Override

**public** **void** **onError**(SHARE\_MEDIA platform, Throwable t) {

Toast.makeText(ShareActivity.**this**,platform + " 分享失败啦", Toast.LENGTH\_SHORT).show();

}

@Override

**public** **void** **onCancel**(SHARE\_MEDIA platform) {

Toast.makeText(ShareActivity.**this**,platform + " 分享取消了", Toast.LENGTH\_SHORT).show();

}

};

值得注意的是，分享也应该重写

onActivityResult()

@Override

**protected** **void** **onActivityResult**(**int** requestCode, **int** resultCode, Intent data) {

**super**.onActivityResult(requestCode, resultCode, data);

UMShareAPI.get( **this** ).onActivityResult( requestCode, resultCode, data);

}

6.3 设置分享内容

目前支持文本、图片（本地及URL）、音频及视频URL的分享

图片(url)

UMImage image = **new** UMImage(ShareActivity.**this**, "http://www.umeng.com/images/pic/social/integrated\_3.png");

图片(本地资源引用)

UMImage image = **new** UMImage(ShareActivity.**this**,

BitmapFactory.decodeResource(getResources(), R.drawable.image));

图片(本地绝对路径)

UMImage image = **new** UMImage(ShareActivity.**this**,

BitmapFactory.decodeFile("/mnt/sdcard/icon.png")));

URL音频及图片

UMusic music = **new** UMusic("http://music.huoxing.com/upload/20130330/1364651263157\_1085.mp3");

music.setTitle("sdasdasd");

music.setThumb(**new** UMImage(ShareActivity.**this**,"http://www.umeng.com/images/pic/social/chart\_1.png"));

url视频

UMVideo video = **new** UMVideo("http://video.sina.com.cn/p/sports/cba/v/2013-10-22/144463050817.html");

6.4 是否开启分享编辑页

Config.OpenEditor = false;

注意只有网页分享时分享编辑页才会有效，故只有豆瓣、人人网有效

6.5 关闭log和toast

Log.LOG ＝ false

Config.IsToastTip = false

6.6 修改dialog

在Android SDK v5.0.2版本中可以使用下面的方法修改默认的progress dialog

ProgressDialog dialog = **new** ProgressDialog(**this**);

dialog.setTitle("11");

dialog.setMessage("222");

Config.dialog = dialog;

6.7 高级功能

对于以上介绍的基本功能已经可以实现一般用户的需求，对于特殊要求，我们进行了特别处理： 针对分享面板，我们进行了如下的处理：

**6.7.1 针对不同平台使用不同监听器，以及分享不同内容**

根据前面的介绍可以知道，使用ShareAction(this).setDisplayList( displaylist)

可以进行分享面板的平台设置。同时使用setContentList(),

可以进行不同分享内容的处理，使用setListenerList()

进行不同回调的处理。

参考下面的代码

**new** ShareAction(**this**).setDisplayList(SHARE\_MEDIA.SINA,SHARE\_MEDIA.QQ,SHARE\_MEDIA.QZONE)

.setContentList(**new** ShareContent(),**new** ShareContent())

.withMedia(image)

.setListenerList(umShareListener,umShareListener)

.open();

**6.7.2 分享面板点击回调**

使用setShareboardclickCallback() 可以进行分享面板不同按钮的点击回调，方便开发者进行特殊处理。

**6.7.3 分享面板增加自定义按钮**

如果需要对分享面板增加自定义的按钮，可以通过 ShareAction(this).addButton(String showword,String Keyword,String icon,String Grayicon,int index) 进行设置其中第一个参数是显示的名字，第二个参数是平台名称， **注意着两个参数需要先声明** ，第三个参数是按钮图标的图片名字，第四个按钮是置灰情况下按钮图标的图片名字， **注意图片直接写路径即可，不要.png等后缀** 最后一个变量是数标（功能保留）。 对于自定义平台的监听需要使用上面提到的分享面板点击回调，如下：

**private** ShareBoardlistener shareBoardlistener = **new** ShareBoardlistener() {

@Override

**public** **void** **onclick**(SnsPlatform snsPlatform,SHARE\_MEDIA share\_media) {

**if** (share\_media==**null**){

**if** (snsPlatform.mKeyword.equals("11")){

Toast.makeText(ShareActivity.**this**,"add button success",Toast.LENGTH\_LONG).show();

}

}

**else** {

**new** ShareAction(ShareActivity.**this**).setPlatform(share\_media).setCallback(umShareListener)

.withText("多平台分享")

.share();

}

}

};

是新增按钮的回调监听是通过判断当获取平台为空时，keyword的值来进行确定的，并在对应的条件中进行处理即可。

**6.7.4 高级功能收集**

由于本次api版本改动较大，对于新版的sdk有特殊需求的用户，可以联系我们的客服，我们将在版本更新中继续添加更多有用的新功能。

7. QQ、Qzone分享集成说明

注意将Android manifest xml里的 腾讯平台APPID替换为自己申请的APPID，注意tencent前缀不要修改，否则会在分享后提示40000或分享取消

8. 微信集成相关说明

微信集成注意以下几点

8.1 微信和微信朋友圈的回调

如果在您的项目中集成了微信或者微信朋友圈，并且您需要准确的分享回调，则需要在AndroidManifest.xml中下注册下面的回调Activity。

<activity

android:name=".wxapi.WXEntryActivity"

android:theme="@android:style/Theme.Translucent.NoTitleBar"

android:configChanges="keyboardHidden|orientation|screenSize"

android:exported="true"

android:screenOrientation="portrait" />

**注意这里不需要修改，否则可能引起微信回调执行不正确，影响微信登录及分享功能**

然后将SDK包中platforms/weixin目录下的wxapi文件夹拷贝到您AndroidMainFest.xml文件application标签里配置的包目录下,然后修改WXEntryActivity的完整路径即可

* 以我们的Demo工程为例

social\_sdk\_example的包名为com.umeng.soexample,因此将wxapi文件夹拷贝到com.umeng.soexample下,最终WXEntryActivity的完整路径为com.umeng.soexample.wxapi.WXEntryActivity

**注意WXEntryActivity.java的完整路径必须正确，否则微信分享回调无法正常执行,微信登录功能也无法正常实现**

其中分享的回调接口SnsPostListener的onComplete方法的第二个参数代表分享的状态码，当值为200时表示分享成功;其余的值则为分享失败。

8.2 如何使用demo中的微信

* Demo中集成的微信无法使用？

原因：Demo中使用的微信AppID绑定的签名信息和你打包时应.所用的签名信息不一致,导致无法拉起微信分享页面， 解决办法有两个，您任选其一：

* 替换Demo中的微信AppID

将Demo中的AppID替换，同时将Demo的包名修改为自己申请微信AppID时设置的包名，同时使用自己的签名文件打包运行Demo

* 使用SDK中的 debug.keystore编译运行Demo工程,debug.keystore密码为android

8.3 微信签名说明

由于微信的签名过程跟验证流程较多，出错的可能性较大，这里将对微信的签名过程跟验证做详细的说明。

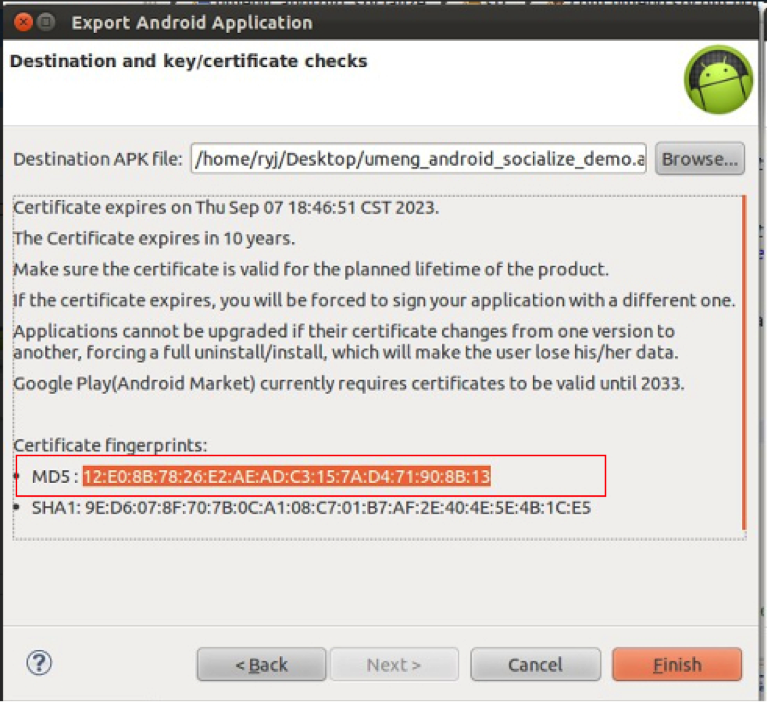
首先必须确保正确设置微信AppID及secret 工程的包名必须同申请应用的包名一致。 在[微信开放平台](https://open.weixin.qq.com/)填写你App的的签名，测试、发布时要保证App的签名跟微信开放平台的签名一致。 开发者的应用需要通过审核。开发者在申请应用必须要通过微信官网的审核才能进行调试，可以使用[签名工具](http://dev.umeng.com/system/resources/W1siZiIsIjIwMTUvMTIvMjEvMTRfMDJfMTZfMzA4X18uYXBrIl1d/%E8%8E%B7%E5%8F%96%E7%AD%BE%E5%90%8D.apk)自行检查比对

**8.3.1 获取签名**

获取应用签名方式有两种

**8.3.1.1 方式一：**

用eclipse导出 带签名 的apk文件，注意此时选择的签名文件是你发布app时的签名文件。在导出的最后一步，eclipse会显示一个Certificate fingerprints(证书指纹)，如下图。此时找到MD5值，将该MD5值复制出来，按照“ 去掉冒号(:)，大写转小写 ”的规则装换，此时字符串就是签名。



**8.3.1.2 方式二：**

导出 带签名 的apk文件，将该apk文件安装至手机中。安装微信的[签名工具](http://dev.umeng.com/system/resources/W1siZiIsIjIwMTUvMTIvMjEvMTRfMDJfMTZfMzA4X18uYXBrIl1d/%E8%8E%B7%E5%8F%96%E7%AD%BE%E5%90%8D.apk),打开应用程序，即可得到签名。

将获取到的签名和下图进行比对



**注意如果修改了微信平台的包名和签名，需要重装微信客户端后才可以测试，否则本地会保留之前的签名缓存**

**8.3.2 请在微信开放平台申请应用**

填写App的相关信息，注意确保包名跟签名填写正确，填写完成提交微信审核。

**8.3.2.1 微信集成测试**

在进行微信集成测试时，请填写你申请App的AppID。测试时需要导出带签名的apk文件测试，否则将不能正常分享。

9. 新浪集成的说明

新浪集成需要注意下面几点

9.1 添加配置文件

在Android manifest xml中添加相关配置， [参考文档](http://dev.umeng.com/social/android/android-update#3)

在你的工程应用包名的相应目录下新增一个WBShareActivity类，该类继承自Activity，预留空类即可

* 以我们的Demo工程为例

social\_sdk\_example的包名为com.umeng.soexample,最终WBShareActivity的完整路径为com.umeng.soexample.WBShareActivity

WBShareActivity内部实现如下

**package** com.umeng.soexample;

**import** com.umeng.socialize.media.WBShareCallBackActivity;

*/\*\**

*\* Created by wangfei on 15/12/3.*

*\*/*

**public** **class** **WBShareActivity** **extends** **WBShareCallBackActivity**{

}

<service

android:name="com.sina.weibo.sdk.net.DownloadService"

android:exported="false">

</service>

注意这里的DownloadService亮红不影响程序编译运行，不需要处理

9.2 分享内容的设置

* 注意新浪分享的title是不显示的，URL链接只能加在分享文字后显示，并且需要确保withText()不为空
* 当同时传递URL参数和图片时，注意确保图片不能超过32K，否则无法分享，不传递URL参数时图片不受32K限制

10. linkedIn集成分享说明

10.1 添加LinkedIn到分享列表

在集成代码中添加下面的代码

*// 添加LinkedIn平台*

UMLinkedInHandler linkedInHandler = **new** UMLinkedInHandler(getActivity());

linkedInHandler.addToSocialSDK();

10.2 添加相关文件及资源

1.解压已下载的zip文件，将platforms/linkedin/libs/SocialSDK\_linkedin.jar和platforms/linkedin/libs/volley.jar拷贝到你工程libs目录中。如果你已经使用VOLLEY框架就不用再次添加VOLLEY.JAR

2.在libs工程找到SocialSDK\_linkedin.jar文件，鼠标右键-->Add to Build Path。

3.将res中的资源添加到你工程对应的res目录中。

11. 不同平台的分享内容要求

不同分享平台对分享内容的要求不同，可以参照下表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 平台 | title | Text | TargetUrl | 图片 | 音乐 | 视频 |
| 新浪微博 | 不支持 | 新浪编辑页面，限制最多140个字符（1个数字或者英文算半个字符，中文算一个字符） | TatgetUrl只能加在文字中间或后面，当同时传递视频/音频时无效 | 支持JPG\PNG\GIF格式的本地及URL图片，GIF图片无动态效果 | 支持URL音乐,显示音乐播放缩略图需要linkcard权限 | 支持URL视频，显示视频播放缩略图需要linkcard权限 |
| QQ好友 | 支持，最多只显示20个字符 | 最多只显示30个字符 | http链接分享后无法打开targetURL，https链接可以打开，纯图片分享时点击无效 | 支持JPG\PNG\GIF格式的本地及URL图片，URL形式及resource内的GIF图片有动态效果，SD卡中的GIF图片无动态效果 | 支持URL音乐,音乐author无法显示，可以点击音乐播放按钮直接播放 | 支持URL视频，视频会转化为音乐样式 |
| QQ空间 | 支持，最多只显示20个字符 | 最多只显示30个字符 | 不支持www开头的URL链接 | 不支持纯图片分享，支持JPG\PNG\GIF格式的本地及URL图片，URL形式及resource内的GIF图片有动态效果，SD卡中的GIF图片无动态效果 | 支持URL音乐,音乐author无法显示，可以点击音乐播放按钮直接播放 | 支持URL视频，视频会转化为音乐样式 |
| 微信好友 | 支持 | 支持超长字符和特殊字符 | 纯图片分享时targetURL无效 | 支持JPG\PNG\GIF格式的本地及URL图片，图文分享类型时GIF图片无动态效果 | 支持URL音乐,音乐author无法显示，可以点击音乐播放按钮直接播放 | 支持URL视频，视频会转化为音乐样式 |

12. FAQ

常见问题

1.如果出现META-INF重复的问题，请在你APP的BUILD.GRADLE文件中加入 android{ ..... packagingOptions { exclude 'META-INF/maven/org.twitter4j/twitter4j-core/pom.xml' exclude 'META-INF/LICENSE.txt' exclude 'META-INF/maven/org.twitter4j/twitter4j-core/pom.properties' } ...... }

13. 应用信息注册地址列表

下面是Social SDK不同平台应用信息注册网站的地址:

|  |  |
| --- | --- |
| 平台 | 网址 |
|  |  |
| 新浪微博 | http://open.weibo.com |
|  |  |
| QQ空间 | http://open.qq.com/ |
|  |  |
| 微信好友 | http://open.weixin.qq.com |
|  |  |
| FaceBook | https://developers.facebook.com |
|  |  |
| 人人 | http://dev.renren.com |
|  |  |
| 豆瓣 | http://developers.douban.com/ |
|  |  |

14. 混淆

为了保证引用友盟Social SDK jar文件以及腾讯jar文件被混淆，请在proguard.cfg文件中添加以下代码避免被混淆.

-dontshrink

-dontoptimize

-dontwarn com.google.android.maps.\*\*

-dontwarn android.webkit.WebView

-dontwarn com.umeng.\*\*

-dontwarn com.tencent.weibo.sdk.\*\*

-dontwarn com.facebook.\*\*

-keep **enum** com.facebook.\*\*

-keepattributes Exceptions,InnerClasses,Signature

-keepattributes \*Annotation\*

-keepattributes SourceFile,LineNumberTable

-keep **public** **interface** **com**.**facebook**.\*\*

-**keep** **public** **interface** **com**.**tencent**.\*\*

-**keep** **public** **interface** **com**.**umeng**.**socialize**.\*\*

-**keep** **public** **interface** **com**.**umeng**.**socialize**.**sensor**.\*\*

-**keep** **public** **interface** **com**.**umeng**.**scrshot**.\*\*

-**keep** **public** **class** **com**.**umeng**.**socialize**.\* {\*;}

-keep **public** **class** **javax**.\*\*

-**keep** **public** **class** **android**.**webkit**.\*\*

-**keep** **class** **com**.**facebook**.\*\*

-**keep** **class** **com**.**facebook**.\*\* { \*; }

-keep **class** **com**.**umeng**.**scrshot**.\*\*

-**keep** **public** **class** **com**.**tencent**.\*\* {\*;}

-keep **class** **com**.**umeng**.**socialize**.**sensor**.\*\*

-**keep** **class** **com**.**umeng**.**socialize**.**handler**.\*\*

-**keep** **class** **com**.**umeng**.**socialize**.**handler**.\*

-**keep** **class** **com**.**tencent**.**mm**.**sdk**.**modelmsg**.**WXMediaMessage** {\*;}

-keep **class** **com**.**tencent**.**mm**.**sdk**.**modelmsg**.\*\* **implements** **com**.**tencent**.**mm**.**sdk**.**modelmsg**.**WXMediaMessage**$**IMediaObject** {\*;}

-keep **class** **im**.**yixin**.**sdk**.**api**.**YXMessage** {\*;}

-keep **class** **im**.**yixin**.**sdk**.**api**.\*\* **implements** **im**.**yixin**.**sdk**.**api**.**YXMessage**$**YXMessageData**{\*;}

-dontwarn twitter4j.\*\*

-keep **class** **twitter4j**.\*\* { \*; }

-keep **class** **com**.**tencent**.\*\* {\*;}

-dontwarn com.tencent.\*\*

-keep **public** **class** **com**.**umeng**.**soexample**.**R**$\*{

**public** **static** **final** **int** \*;

}

-keep **public** **class** **com**.**umeng**.**soexample**.**R**$\*{

**public** **static** **final** **int** \*;

}

-keep **class** **com**.**tencent**.**open**.**TDialog**$\*

-**keep** **class** **com**.**tencent**.**open**.**TDialog**$\* {\*;}

-keep **class** **com**.**tencent**.**open**.**PKDialog**

-**keep** **class** **com**.**tencent**.**open**.**PKDialog** {\*;}

-keep **class** **com**.**tencent**.**open**.**PKDialog**$\*

-**keep** **class** **com**.**tencent**.**open**.**PKDialog**$\* {\*;}

-keep **class** **com**.**sina**.\*\* {\*;}

-dontwarn com.sina.\*\*

-keep **class** **com**.**alipay**.**share**.**sdk**.\*\* {

\*;

}

-keepnames **class** \* **implements** **android**.**os**.**Parcelable** {

**public** **static** **final** \*\* CREATOR;

}

-keep **class** **com**.**linkedin**.\*\* { \*; }

-keepattributes Signature