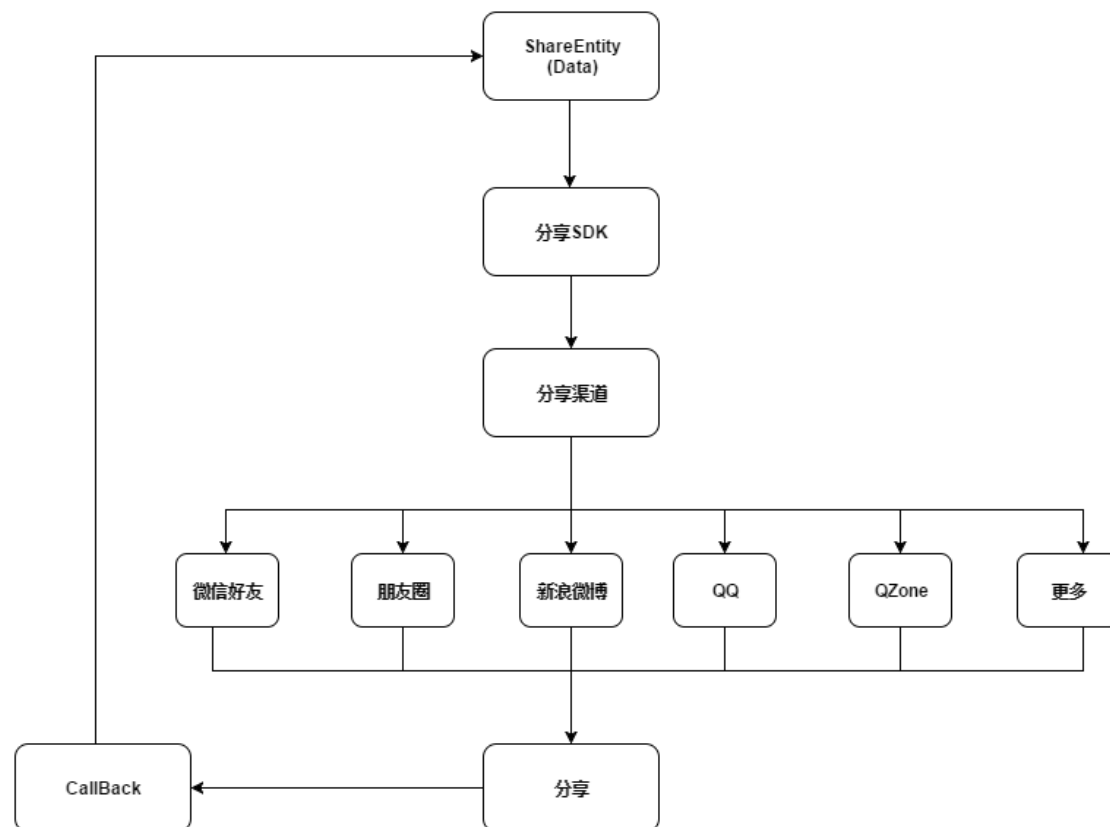


分享 SDK 使用说明文档

一、分享流程

分享 SDK 的工作流程其实很简单，即：



二、设计思路

鉴于微博分享回调必须依附于 Activity，腾讯 QQ、QZone 也是一样。因此分享中，用一个 Activity 作为分享渠道的转发，通过 onActivityResult（）回调结果。

三、分享配置

1、混淆相关，在 proguard-rules 里面加上：

```
-ignorewarnings
-keep class com.tencent.mm.sdk.** {
    *,
}
}
```

2、AndroidManifest.xml 相关配置：

```
<meta-data
    android:name="tencent_qq_appid"
    android:value="@string/tencent_qq_appid"/>
<meta-data
    android:name="weixin_key"
    android:value="@string/weixin_key"/>
<!-- 微信分享注册 -->
<activity
    android:name=".wxapi.WXEntryActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:exported="true"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />

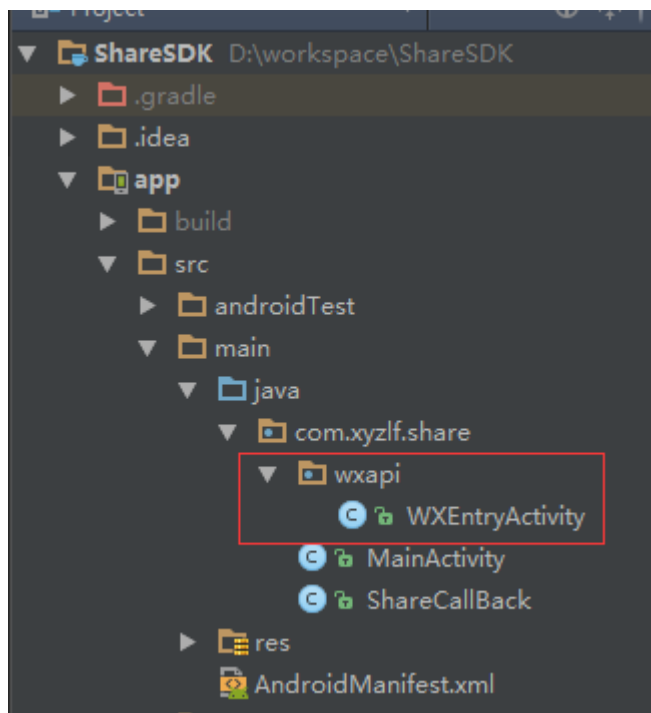
<!-- QQ SDK 需要注册 Activity -->
<activity
    android:name="com.tencent.connect.common.AssistActivity"
    android:configChanges="orientation|keyboardHidden"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.tencent.taauth.AuthActivity"
    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="tencent222222" />
        <!-- 100380359 100381104 222222 -->
    </intent-filter>
```

```
</activity>
<!-- QQ SDK 需要注册 Activity -->
```

微信分享配置需特别注意:

必须生成一个叫做: `WXEntryActivity` 的 Activity, 完整包名必须是: `{App packageName}.wxapi.WXEntryActivity`, 如下:

```
<!-- 微信分享注册 -->
<activity
    android:name=".wxapi.WXEntryActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:exported="true"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
```



四、使用方式

1、调出分享界面



```
/**
 * 使用统一数据结构
 */
public void showShareDialog() {
    ShareEntity testBean = new ShareEntity("我是标题", "我是内容，描述内容。");
    testBean.setUrl("https://www.baidu.com"); //分享链接
    testBean.setImgUrl("https://www.baidu.com/img/bd_logo1.png");
    ShareUtil.showShareDialog(this, testBean, ShareConstant.REQUEST_CODE);
}

/**
 * 使用数组数据结构
 */
```

```

public void showShareDialog() {
    ShareEntity testBean = new ShareEntity("我是标题", "我是内容, 描述内容。");
    testBean.setUrl("https://www.baidu.com"); //分享链接
    testBean.setImgUrl("https://www.baidu.com/img/bd_logo1.png");

    SparseArray<ShareEntity> sparseArray = new SparseArray<>();
    sparseArray.put(ShareConstant.SHARE_CHANNEL_QQ, testBean);
    sparseArray.put(ShareConstant.SHARE_CHANNEL_QZONE, testBean);
    sparseArray.put(ShareConstant.SHARE_CHANNEL_WEIXIN_CIRCLE, testBean);
    sparseArray.put(ShareConstant.SHARE_CHANNEL_WEIXIN_FRIEND, testBean);
    sparseArray.put(ShareConstant.SHARE_CHANNEL_SINA_WEIBO, testBean);
    sparseArray.put(ShareConstant.SHARE_CHANNEL_SYSTEM, testBean);

    ShareUtil.showShareDialog(this, testBean, ShareConstant.REQUEST_CODE);
}

```

处理分享回调接口, 在调用的 Activity 的 onActivityResult 中处理:

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == ShareConstant.REQUEST_CODE) {
        if (data != null) {
            int channel = data.getIntExtra(ShareConstant.EXTRA_SHARE_CHANNEL, -1);
            int status = data.getIntExtra(ShareConstant.EXTRA_SHARE_STATUS, -1);
            ToastUtil.showToast(this, "callback : channel = " + channel + "; status = " + status, false);
        }
    }
}

```

2、使用分享功能

不需要分享界面, 只是单纯的使用分享功能:

```
/**
 * 使用具体的分享功能
 *
 * QQ 举例
 */
public void startShare(int channel) {

    channel = ShareConstant.SHARE_CHANNEL_QQ;

    ShareEntity testBean = new ShareEntity("我是标题", "我是内容，描述内容。");
    testBean.setUrl("https://www.baidu.com"); //分享链接
    testBean.setImgUrl("https://www.baidu.com/img/bd_logo1.png");
    ShareUtil.startShare(this, channel, testBean, ShareConstant.REQUEST_CODE);
}
```

五、渠道动态配置

在使用方式的基础上，传递渠道值，可动态展示分享渠道，如下：

```
/**
 * 渠道动态可配
 *
 * 只展示微信、微信朋友圈
 */
public void startShare() {

    ShareEntity testBean = new ShareEntity("我是标题", "我是内容，描述内容。");
    testBean.setUrl("https://www.baidu.com"); //分享链接
    testBean.setImgUrl("https://www.baidu.com/img/bd_logo1.png");

    int channel = ShareConstant.SHARE_CHANNEL_WEIXIN_CIRCLE |
ShareConstant.SHARE_CHANNEL_WEIXIN_FRIEND;

    ShareUtil.showShareDialog(this, channel, testBean, ShareConstant.REQUEST_CODE);
}
```

六、分享大图

(1) 分享本地图片


```
ShareEntity testBean = new ShareEntity("我是标题", "我是内容，描述内容。");
```

1、本地 drawable 图片，使用方式：testBean.setDrawableId(R.drawable.icon);

2、本地手机 SD 卡图片，使用方式：testBean.setImgUrl("/sdpath/filepath/icon.png");

3、本地代码截屏 Bitmap，使用方式：

```
Bitmap bitmap = null; // 你自己截屏的 bitmap
```

```
String filePath = ShareUtil.saveBitmapToSDCard(this, bitmap);
```

```
testBean.setImgUrl(filePath);
```

（2）微信分享大图

添加大图属性：

```
ShareEntity testBean = new ShareEntity("", "");
```

```
testBean.setShareBigImg(true);
```

其他设置如上介绍。