

Eclipse集成步骤

第一步： 在个推开发者平台创建应用

登录[个推开发者平台](#)，登记应用，并获取到相应的APPKEY、APPID、APP SECRET信息。参见 [登记应用](#)

注：老版本升级2.9+

- 1.删除原来所有旧的布局文件， 加上新的getui_notification.xml
- 2.替换旧的GetuiSDKxxx.jar并删除GetuiExt.xxx.jar和所有对应目录下的libgetuiext.so
- 3.修改AndroidManifest.xml中个推广播和服务的配置， 修改后见下方第七步

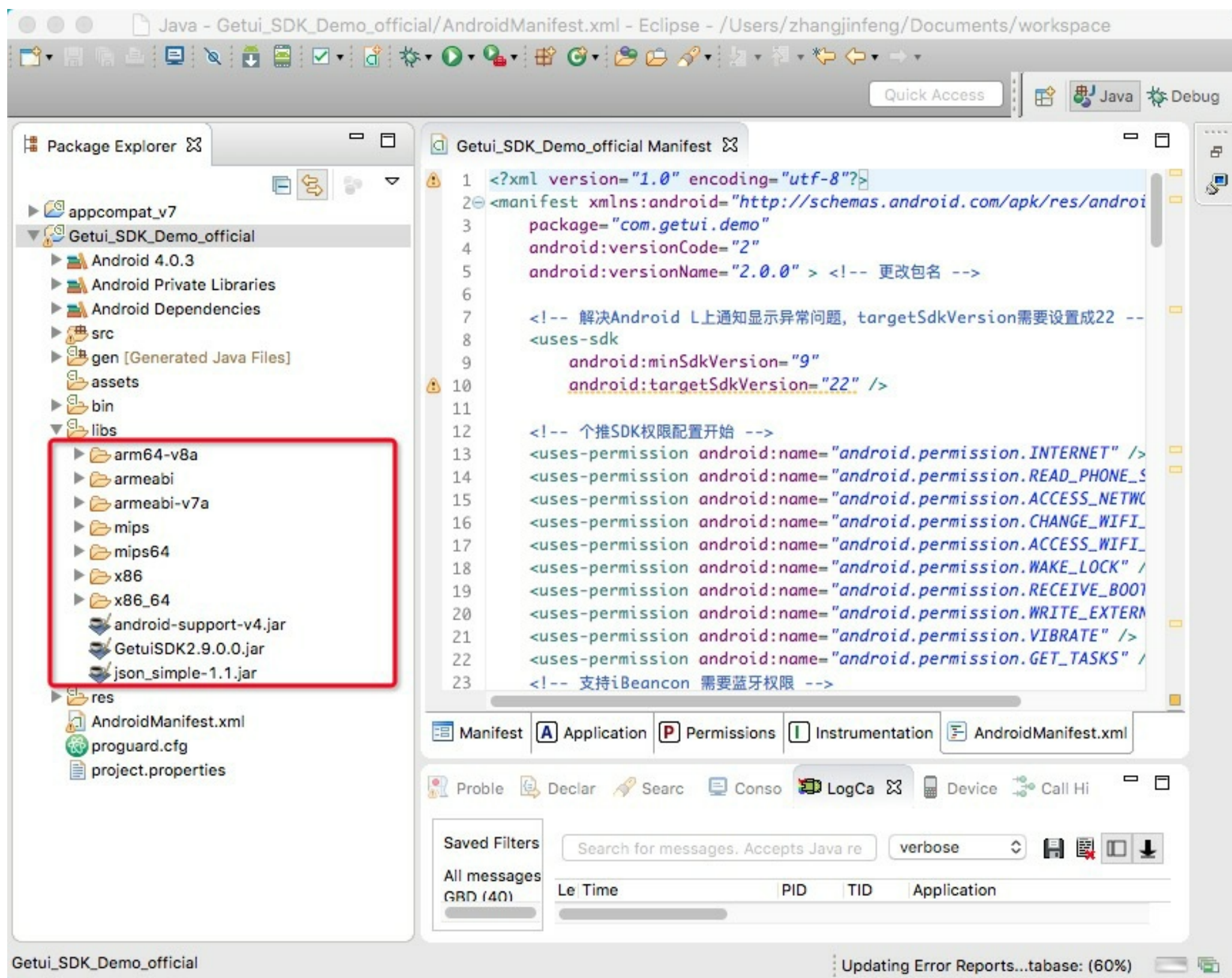
第二步： 导入个推SDK LIB

第一步： 导入资源

将SDK资料包中“GETUI_ANDROID_SDK\资源文件”目录下的GetuiSdk-xxx.jar、so文件夹子文件复制到工程根目录下的libs文件夹中。

注： 导入需要的 cpu 架构的 so 库即可

若没有libs目录， 则选中工程右键“New”->Folder->命名为“libs”。导入后工程目录结构如下：



第二步：添加引用

右键单击工程，选择Build Path中的Configure Build Path...，选中Libraries，并通过Add Jars...导入工程libs目录下的GetuiSdk-xxx.jar文件（最新的ADT工具会自动导入libs目录下的jar包，故该步骤可以省略）

如需使用最新的展开式样式和浮动通知，需导入android-support-v4.jar包

第三步：初始化SDK

在您应用程序主Activity里导入PushManager如下所示：

```
import com.igexin.sdk.PushManager;
```

然后在您应用程序启动初始化阶段，初始化SDK：

```
PushManager.getInstance().initialize(this.getApplicationContext());
```

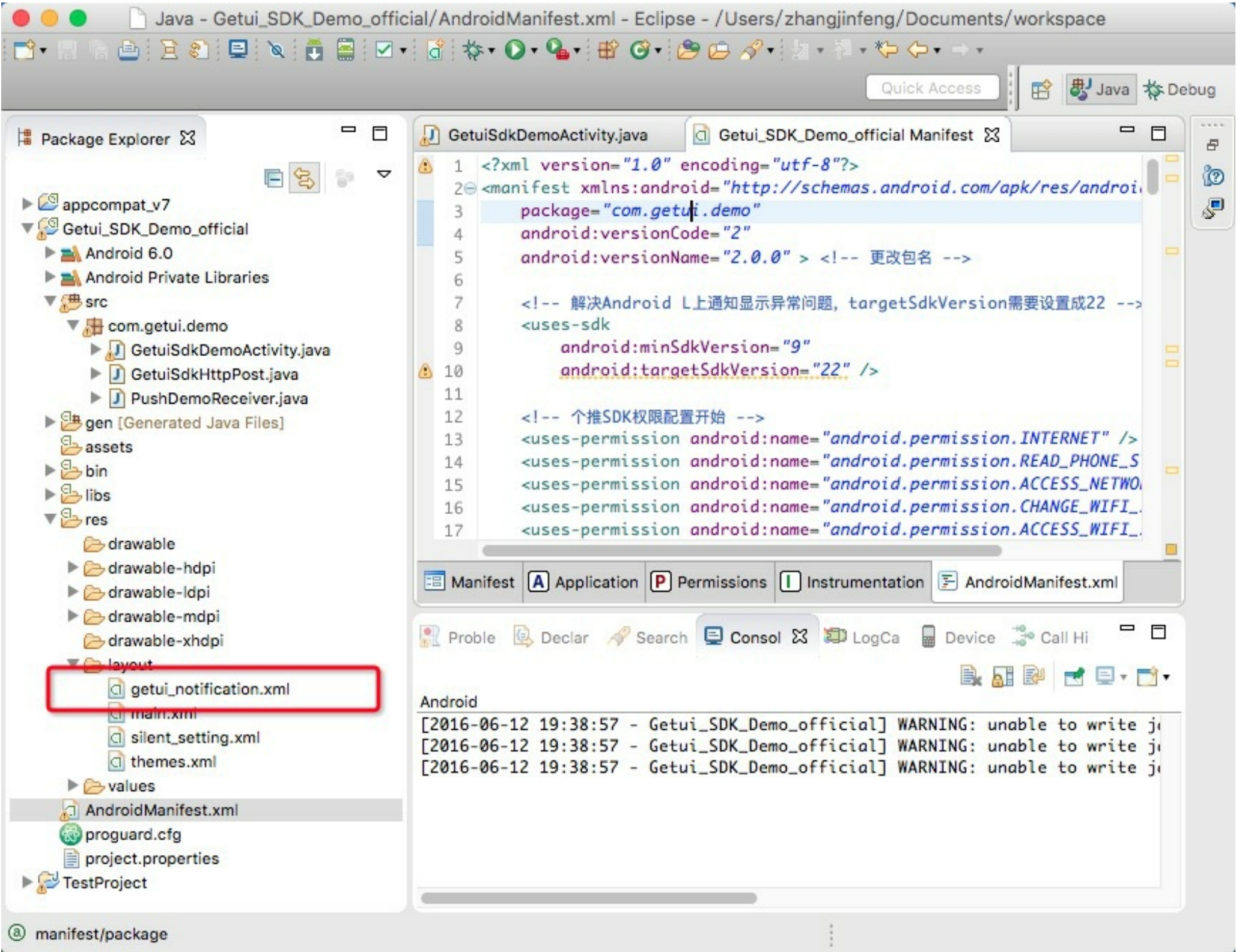
建议应用程序每次启动时都调用一次该初始化接口。

注：该方法必须在Activity或服务类内调用，一般情况下，可以在Activity的onCreate()方法中调用。由于应

用每启动一个新的进程，就会调用一次Application的onCreate()方法，而个推SDK是一个独立的进程，因此如果在Application的onCreate()中调用intialize接口，会导致SDK初始化在一个应用中多次调用，所以不建议在Application继承类中调用个推SDK初始化接口。

第四步：导入布局文件

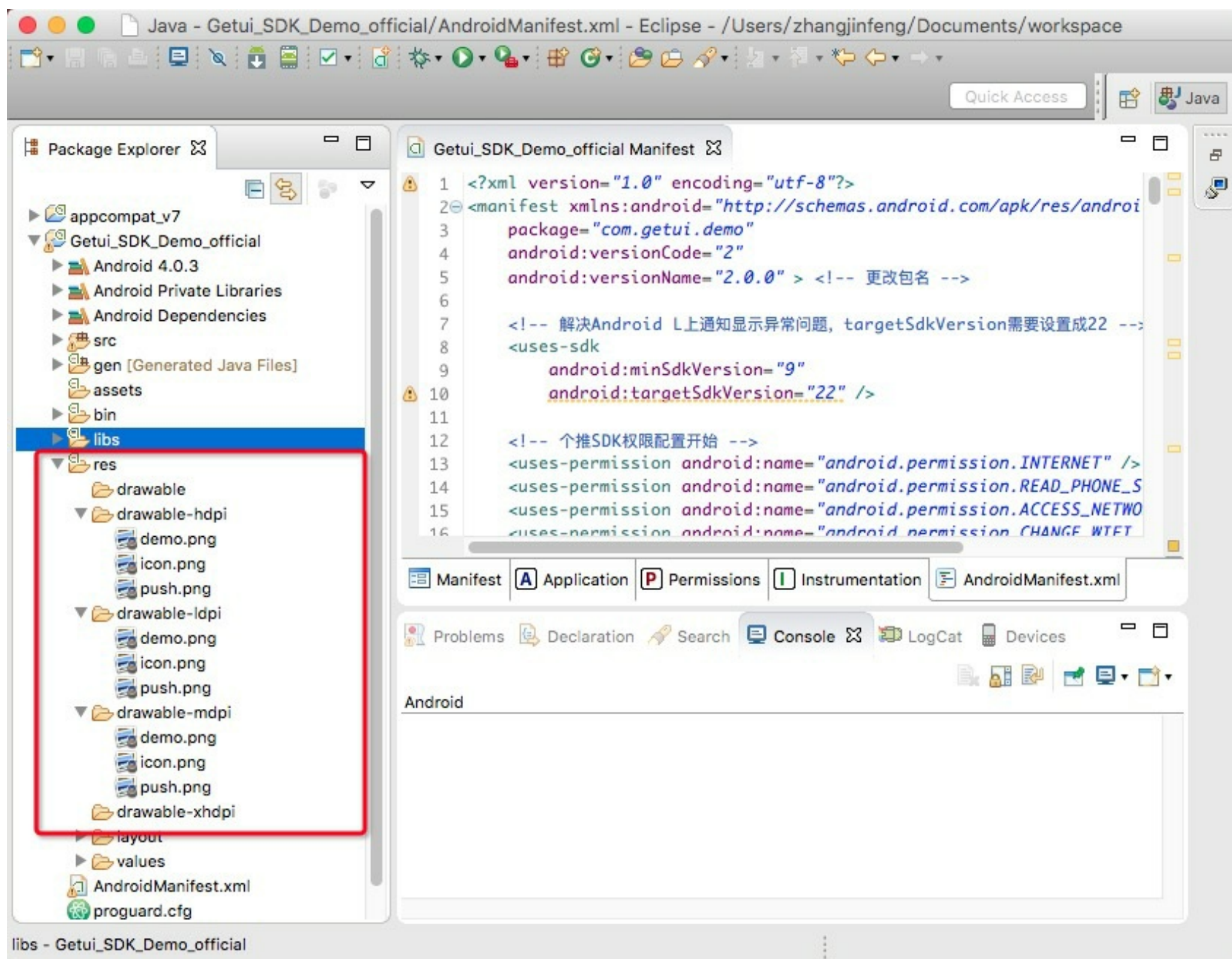
将 “...\GETUI_ANDROID_SDK\资源文件\layout”下的xml布局文件复制到工程res/layout/文件夹中。



注：如需使用最新的展开式样式和浮动通知，需导入getui_notification.xml布局文件

第五步：导入通知栏图标

为了修改通知栏提示图标，请在res/drawable-hdpi/、res/drawable-mdpi/、res/drawable-ldpi/等各分辨率资源目录下，放置相应尺寸的push.png图片。



该图标将会作为通知图标展示在通知栏顶部，如下所示：



第六步：混淆打包配置

如果您需要使用proguard进行混淆打包，请在proguard.cfg添加如下代码：

```
-dontwarn com.igexin.**  
-keep class com.igexin.**{*;}  

```

确保个推SDK相关的类不被混淆。

注：如果在proguard.cfg中已经添加-dontwarn 无需再添加 -dontwarn com.igexin.**

第七步： 在AndroidManifest.xml 里添加相关声明

第一步： 添加服务声明

在Application标签内加入如下服务声明：

```
<!-- 个推SDK配置开始 -->
<!-- 配置的第三方参数属性 -->
<meta-data
    android:name="PUSH_APPID"
    android:value="你的APPID" /> <!-- 替换为第三方应用的APPID -->
<meta-data
    android:name="PUSH_APPKEY"
    android:value="你的APPKEY" /> <!-- 替换为第三方应用的APPKEY -->
<meta-data
    android:name="PUSH_APPSECRET"
    android:value="你的APPSECRET" /> <!-- 替换为第三方应用的APPSECRET -->
<!-- 配置SDK核心服务 -->
<service
    android:name="com.igexin.sdk.PushService"
    android:exported="true"
    android:label="NotificationCenter"
    android:process=":pushservice" >
    <intent-filter>
        <action android:name="com.igexin.sdk.action.service.message"/>
    </intent-filter>
</service>

<service
    android:name="com.igexin.sdk.PushServiceUser"
    android:exported="true"
    android:label="NotificationCenterUser">
    <intent-filter>
        <action android:name="com.igexin.sdk.action.user.message"/>
    </intent-filter>
</service>

<receiver android:name="com.igexin.sdk.PushReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
        <action android:name="android.intent.action.USER_PRESENT" />
        <action android:name="com.igexin.sdk.action.refreshls" />
        <!-- 以下三项为可选的action声明，可大大提高service存活率和消息到达速度 -->
        <action android:name="android.intent.action.MEDIA_MOUNTED" />
        <action android:name="android.intent.action.ACTION_POWER_CONNECTED" />
        <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED" />
    </intent-filter>
```



```

</receiver>
<receiver
    android:name="com.igexin.sdk.PushManagerReceiver"
    android:exported="false" >
    <intent-filter>
        <action android:name="com.igexin.sdk.action.pushmanager" />
    </intent-filter>
</receiver>
<activity
    android:name="com.igexin.sdk.PushActivity"
    android:excludeFromRecents="true"
    android:exported="false"
    android:process=":pushservice"
    android:taskAffinity="com.igexin.sdk.PushActivityTask"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.igexin.sdk.GActivity"
    android:excludeFromRecents="true"
    android:exported="true"
    android:process=":pushservice"
    android:taskAffinity="com.igexin.sdk.PushActivityTask"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<service
    android:name="com.igexin.download.DownloadService"
    android:process=":pushservice" />
<receiver android:name="com.igexin.download.DownloadReceiver" >
    <intent-filter>
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    </intent-filter>
</receiver>
<provider
    android:name="com.igexin.download.DownloadProvider"
    android:authorities="downloads.你的包名"
    android:exported="true"
    android:process=":pushservice" /> <!-- 替换为第三方应用的包名 -->
<!-- 个推SDK配置结束 -->

```

注：需要将注释部分参数替换为个推开发者平台上应用所分配到的参数

第二步：添加权限声明

在Application标签外加入个推SDK运行时需要的权限：

```

<!-- 解决Android L上通知显示异常问题，targetSdkVersion需要设置成22 -->
<uses-sdk
    android:minSdkVersion="9"
    android:targetSdkVersion="22" />
<!-- 个推SDK权限配置开始 -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />

```

```

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.GET_TASKS" />
<!-- ibeancon 需要蓝牙权限 -->
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<!-- 支持个推3.0 电子围栏功能 -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<!-- 浮动通知权限 -->
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<!-- 自定义权限 -->
<uses-permission android:name="getui.permission.GetuiService.你的包名" />
<!--替换为第三方应用的包名-->
<permission
    android:name="getui.permission.GetuiService.你的包名"
    android:protectionLevel="normal" >
</permission><!--替换为第三方应用的包名-->
<!-- 个推SDK权限配置结束 -->

```

自定义权限解释：部分手机型号不能正常运行个推SDK，需添加自定义权限进行配置。

第三步：完整示例

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.getui.demo"
    android:versionCode="2"
    android:versionName="2.0.0">
    <!-- 解决Android L上通知显示异常问题，targetSdkVersion需要设置成22 -->
    <uses-sdk
        android:minSdkVersion="9"
        android:targetSdkVersion="22"/>
    <!-- 个推SDK权限配置开始 -->
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.GET_TASKS"/>
    <!-- 支持iBeacon 需要蓝牙权限 -->
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
    <!-- 支持个推3.0 电子围栏功能 -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <!-- 浮动通知权限 -->
    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
    <!-- 自定义权限 -->
    <uses-permission android:name="getui.permission.GetuiService.com.getui.demo"/>
    <permission

```



```
    android:name="getui.permission.GetuiService.com.getui.demo"
    android:protectionLevel="normal">
</permission>
<!-- 个推SDK权限配置结束 -->
<application
    android:icon="@drawable/demo"
    android:label="@string/app_name"
    android:persistent="true">
    <!-- 第三方应用配置 -->
    <activity
        android:name=".GetuiSdkDemoActivity"
        android:label="@string/app_name"
        android:launchMode="singleTop">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <!-- 在上面加入你的你的activity配置 -->
    <!-- 个推SDK配置开始 -->
    <!-- 配置的第三方参数属性 -->
    <meta-data
        android:name="PUSH_APPID"
        android:value="wT1tGnaFC98Kgpfoi2u7g6"/>
    <meta-data
        android:name="PUSH_APPKEY"
        android:value="wT1tGnaFC98Kgpfoi2u7g6"/>
    <meta-data
        android:name="PUSH_APPSECRET"
        android:value="my9R0U9s2Y8vLSfeToj6N5"/>
    <!-- 配置第三方Receiver -->
    <receiver
        android:name="com.getui.demo.PushDemoReceiver"
        android:exported="false">
        <intent-filter>
            <action android:name="com.igexin.sdk.action.wT1tGnaFC98Kgpfoi2u7g6"/>
        </intent-filter>
    </receiver>
    <!-- 配置SDK核心服务 -->
    <service
        android:name="com.igexin.sdk.PushService"
        android:exported="true"
        android:label="NotificationCenter"
        android:process=":pushservice">
        <intent-filter>
            <action android:name="com.igexin.sdk.action.service.message"/>
        </intent-filter>
    </service>

    <service
        android:name="com.igexin.sdk.PushServiceUser"
        android:exported="true"
        android:label="NotificationCenterUser">
        <intent-filter>
            <action android:name="com.igexin.sdk.action.user.message"/>
        </intent-filter>
    </service>
</application>
</manifest>
```

```

</service>

<receiver android:name="com.igexin.sdk.PushReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE"/>
        <action android:name="android.intent.action.USER_PRESENT"/>
        <action android:name="com.igexin.sdk.action.refreshls"/>
        <!-- 以下三项为可选的action声明，可大大提高service存活率和消息到达速度 -->
        <action android:name="android.intent.action.MEDIA_MOUNTED"/>
        <action android:name="android.intent.action.ACTION_POWER_CONNECTED"/>
        <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED"/>
    </intent-filter>
</receiver>
<receiver
    android:name="com.igexin.sdk.PushManagerReceiver"
    android:exported="false">
    <intent-filter>
        <action android:name="com.igexin.sdk.action.pushmanager"/>
    </intent-filter>
</receiver>
<activity
    android:name="com.igexin.sdk.PushActivity"
    android:excludeFromRecents="true"
    android:exported="false"
    android:process=":pushservice"
    android:taskAffinity="com.igexin.sdk.PushActivityTask"
    android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<activity
    android:name="com.igexin.sdk.GActivity"
    android:excludeFromRecents="true"
    android:exported="true"
    android:process=":pushservice"
    android:taskAffinity="com.igexin.sdk.PushActivityTask"
    android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<service
    android:name="com.igexin.download.DownloadService"
    android:process=":pushservice"/>
<receiver android:name="com.igexin.download.DownloadReceiver">
    <intent-filter>
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE"/>
    </intent-filter>
</receiver>
<provider
    android:name="com.igexin.download.DownloadProvider"
    android:authorities="downloads.com.getui.demo"
    android:exported="true"
    android:process=":pushservice"/>
<!-- 个推SDK配置结束 -->
</application>
</manifest>


```

第八步：测试

运行您的工程，查看logcat信息，如图所示。在Logcat中搜索“clientid”可以看到“clientid is xxx”，则意味则初始化SDK成功，并获取到相应的cid信息，恭喜你:-D，可以开始进行推送测试了

clientid						
Level	Time	PID	TID	Application	Tag	Text
D	05-04 10:04:06.882	17842	17862	com.getui.demo:pushservice	PushService	clientid is 98f9e6ea5799a3208e14b3cfe18fbcc9

登录[个推开发者平台](#)，点击应用管理， 进入待测试应用的推送界面：



个推 App De...

创建推送

推送通知

透传消息

分组对比测试

数据统计

推送记录

推送数据

推送统计

用户数据

配置管理

应用配置

别名管理

应用标签

故障排查

推送通知 ?

展示内容：☐ 默认 ☐ 安卓 ☐ 图片

* 通知标题：

请输入通知标题

0/20

* 通知内容：

请输入通知内容

0/50

* 展开式通知：

☐ 禁用 ☐ 文本 ☐ 大图

* 目标平台：

☒ Android

* 目标用户：

☐ 全部用户 ☐ 标签用户 ☐ 特定用户 ☐ 用户分组

* 后续动作：

☐ 启动应用 ☐ 打开链接 ☐ 下载应用

* 推送方式：

☐ 按时间

☐ 即时

☐ 定时

是否进离线消息：

☐ 是 ☐ 否

* 有效时长：

2


小时, 该时间段内cid在线过的用户均可收到通知。(0- 72 小时内的正整数)

定时展示：

☐ 否 ☐ 是

高级设置

发送



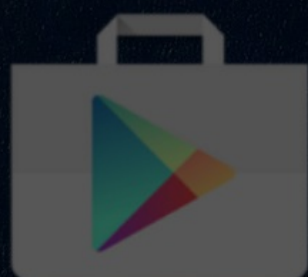
依填写相应的通知标题、通知内容，进行通知推送，具体推送方法见：[创建推送通知](#)

如果手机或模拟器收到通知，如图所示：恭喜您，SDK接入已经成功





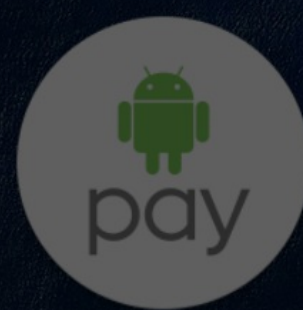
Google



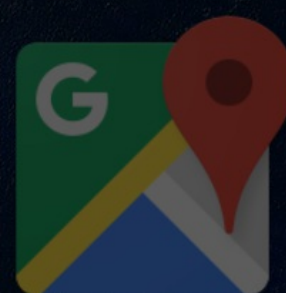
Play 商店



个推演示



Android Pay



地图

