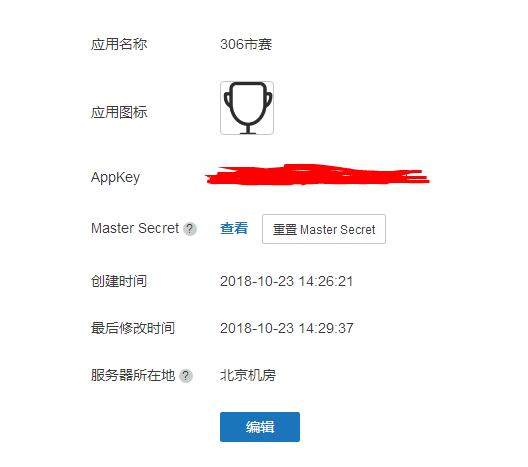
## 极光推送

1. 开发平台：Android studio3.2
2. 去官网注册极光账号：<https://www.jiguang.cn/accounts/platform>
3. 进入你的后台管理界面，新建一个应用，这个Appkey很重要，工程里要用到



1. 导入极光推送的SDK

|  |
| --- |
| 这里有两种方法  1.Jcenter自动集成  2.手动集成(sdk下载地址： <https://docs.jiguang.cn/jpush/resources/> 教程：<https://docs.jiguang.cn/jpush/client/Android/android_guide/> )  因为我的开发平台版本是3.2所以我用了第一种  第一步：用jcenter自动集成的话，要保证build.gradle(Project)里面有jcenter支持 |

1. 在 module 的 gradle(App) 中添加依赖和 AndroidManifest 的替换变量。

|  |
| --- |
| android {  defaultConfig {  applicationId "com.xxx.xxx" //JPush 上注册的包名.  ndk {  //选择要添加的对应 cpu 类型的 .so 库。  abiFilters 'armeabi', 'armeabi-v7a', 'arm64-v8a'  // 还可以添加 'x86', 'x86\_64', 'mips', 'mips64'  }  manifestPlaceholders = [  JPUSH\_PKGNAME : applicationId,  JPUSH\_APPKEY : "你的 Appkey ", //JPush 上注册的包名对应的 Appkey.  JPUSH\_CHANNEL : "developer-default", //暂时填写默认值即可.  ]  }  }  depndencies {  compile 'cn.jiguang.sdk:jpush:3.1.6' // 此处以JPush 3.1.6 版本为例。  compile 'cn.jiguang.sdk:jcore:1.2.5' // 此处以JCore 1.2.5 版本为例。  } |

1. ok配置完成
2. 实现接收极光推送的消息，和处理消息

|  |  |  |
| --- | --- | --- |
| 1. 首先要在自己的Application类里面初始化极光推送   JPushInterface.*setDebugMode*(**true**); JPushInterface.*init*(**this**);   1. 自定义接收消息的Receiver类继承BroadcastReceiver  |  | | --- | | **public class** MyReceiver **extends** BroadcastReceiver {  **private static final** String ***TAG*** = **"JIGUANG-Example"**;  **private** SharedPreferences **pref**;  **private** SharedPreferences.Editor **editor**; Handler **handler**=**new** Handler(){  @Override  **public void** handleMessage(Message msg) {  **super**.handleMessage(msg);  **if**(msg.**what**==0x30){  **editor**.putString(**"getMessage"**,**"1"**);  **editor**.commit();  Intent intent = **new** Intent();  *//设置广播的名字（设置Action）* intent.setAction(**"myBroadCast"**);  *//携带数据* intent.putExtra(**"data"**,**"hello"**);  *//发送广播（无序广播）* ECApplication.*getmContext*().sendBroadcast(intent);  }**else if**(msg.**what**==0x31){  **editor**.putString(**"getMessage"**,**"0"**);  **editor**.commit();  }  } };  @Override  **public void** onReceive(Context context, Intent intent) {  **pref**=context.getSharedPreferences(**"jackie"**,Context.***MODE\_PRIVATE***);  **editor**=**pref**.edit();  **try** {  Bundle bundle = intent.getExtras();  Log.*d*(**"zjc"**, **"[MyReceiver] onReceive - "** + intent.getAction() + **", extras: "** + *printBundle*(bundle));  **if** (JPushInterface.***ACTION\_REGISTRATION\_ID***.equals(intent.getAction())) {  String regId = bundle.getString(JPushInterface.***EXTRA\_REGISTRATION\_ID***);  Log.*d*(**"zjc"**, **"[MyReceiver] 接收Registration Id : "** + regId);  *//send the Registration Id to your server...* } **else if** (JPushInterface.***ACTION\_MESSAGE\_RECEIVED***.equals(intent.getAction())) {  Log.*d*(**"zjc"**, **"[MyReceiver] 接收到推送下来的自定义消息: "** + bundle.getString(JPushInterface.***EXTRA\_MESSAGE***));  processCustomMessage(context, bundle);   } **else if** (JPushInterface.***ACTION\_NOTIFICATION\_RECEIVED***.equals(intent.getAction())) {   **handler**.sendEmptyMessage(0x30);  Log.*d*(**"zjc"**, **"[MyReceiver] 接收到推送下来的通知"**);  **int** notifactionId = bundle.getInt(JPushInterface.***EXTRA\_NOTIFICATION\_ID***);  Log.*d*(**"zjc"**, **"[MyReceiver] 接收到推送下来的通知的ID: "** + notifactionId);   } **else if** (JPushInterface.***ACTION\_NOTIFICATION\_OPENED***.equals(intent.getAction())) {  **handler**.sendEmptyMessage(0x31);  Log.*d*(**"zjc"**, **"[MyReceiver] 用户点击打开了通知"**);  *//打开自定义的Activity* Intent i = **new** Intent(context, tongzhiActivity.**class**);  i.putExtras(bundle);  *//i.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);* i.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK*** | Intent.***FLAG\_ACTIVITY\_CLEAR\_TOP*** );  context.startActivity(i);   } **else if** (JPushInterface.***ACTION\_RICHPUSH\_CALLBACK***.equals(intent.getAction())) {  Log.*d*(**"zjc"**, **"[MyReceiver] 用户收到到RICH PUSH CALLBACK: "** + bundle.getString(JPushInterface.***EXTRA\_EXTRA***));  *//在这里根据 JPushInterface.EXTRA\_EXTRA 的内容处理代码，比如打开新的Activity， 打开一个网页等..* } **else if**(JPushInterface.***ACTION\_CONNECTION\_CHANGE***.equals(intent.getAction())) {  **boolean** connected = intent.getBooleanExtra(JPushInterface.***EXTRA\_CONNECTION\_CHANGE***, **false**);  Log.*w*(**"zjc"**, **"[MyReceiver]"** + intent.getAction() +**" connected state change to "**+connected);  } **else** {  Log.*d*(**"zjc"**, **"[MyReceiver] Unhandled intent - "** + intent.getAction());  }  } **catch** (Exception e){   }   }   *// 打印所有的 intent extra 数据* **private static** String printBundle(Bundle bundle) {  StringBuilder sb = **new** StringBuilder();  **for** (String key : bundle.keySet()) {  **if** (key.equals(JPushInterface.***EXTRA\_NOTIFICATION\_ID***)) {  sb.append(**"\nkey:"** + key + **", value:"** + bundle.getInt(key));  }**else if**(key.equals(JPushInterface.***EXTRA\_CONNECTION\_CHANGE***)){  sb.append(**"\nkey:"** + key + **", value:"** + bundle.getBoolean(key));  } **else if** (key.equals(JPushInterface.***EXTRA\_EXTRA***)) {  **if** (TextUtils.*isEmpty*(bundle.getString(JPushInterface.***EXTRA\_EXTRA***))) {  Log.*i*(***TAG***, **"This message has no Extra data"**);  **continue**;  }  **try** {  JSONObject json = **new** JSONObject(bundle.getString(JPushInterface.***EXTRA\_EXTRA***));  Iterator<String> it = json.keys();   **while** (it.hasNext()) {  String myKey = it.next();  sb.append(**"\nkey:"** + key + **", value: ["** +  myKey + **" - "** +json.optString(myKey) + **"]"**);  }  } **catch** (JSONException e) {  Log.*e*(***TAG***, **"Get message extra JSON error!"**);  }   } **else** {  sb.append(**"\nkey:"** + key + **", value:"** + bundle.get(key));  }  }  **return** sb.toString();  }    *//send msg to MainActivity* **private void** processCustomMessage(Context context, Bundle bundle) {  **if** (MainActivity.*isForeground*) {  String message = bundle.getString(JPushInterface.***EXTRA\_MESSAGE***);  String extras = bundle.getString(JPushInterface.***EXTRA\_EXTRA***);  Intent msgIntent = **new** Intent(MainActivity.***MESSAGE\_RECEIVED\_ACTION***);  msgIntent.putExtra(MainActivity.***KEY\_MESSAGE***, message);  **if** (!ExampleUtil.*isEmpty*(extras)) {  **try** {  JSONObject extraJson = **new** JSONObject(extras);  **if** (extraJson.length() > 0) {  msgIntent.putExtra(MainActivity.***KEY\_EXTRAS***, extras);  }  } **catch** (JSONException e) {   }   }  LocalBroadcastManager.*getInstance*(context).sendBroadcast(msgIntent);  }  } } |  1. 然后在AndroidManifest.xml里面配置这个Receiver类  |  | | --- | | <**receiver  android:name=".utils.MyReceiver"  android:enabled="true"  android:exported="false"**>  <**intent-filter**>  <**action android:name="cn.jpush.android.intent.REGISTRATION"** /> *<!-- Required 用户注册SDK的intent -->* <**action android:name="cn.jpush.android.intent.MESSAGE\_RECEIVED"** /> *<!-- Required 用户接收SDK消息的intent -->* <**action android:name="cn.jpush.android.intent.NOTIFICATION\_RECEIVED"** /> *<!-- Required 用户接收SDK通知栏信息的intent -->* <**action android:name="cn.jpush.android.intent.NOTIFICATION\_OPENED"** /> *<!-- Required 用户打开自定义通知栏的intent -->* <**action android:name="cn.jpush.android.intent.CONNECTION"** /> *<!-- 接收网络变化 连接/断开 since 1.6.3 -->* <**category android:name="com.example.a11699.graduatemanager"** />  </**intent-filter**> </**receiver**> | |

1. 上面写了收到推送的点击事件跳转到tongzhiActivity.**class**

在ApplicationContext.xml声明这个类

<**activity android:name=".activity.tongzhiActivity"**>  
 <**intent-filter**>  
 <**action android:name="jpush.testAction"** />  
  
 <**category android:name="jpush.testCategory"** />  
 </**intent-filter**>  
</**activity**>

1. **给tongzhiActivity填充代码**

|  |
| --- |
| **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***tongggg***);  **textView**=findViewById(R.id.***te***);  Intent intent = getIntent();  **if** (**null** != intent) {  Bundle bundle = getIntent().getExtras();  **title** = **null**;  **content** = **null**;  **if**(bundle!=**null**){  **title** = bundle.getString(JPushInterface.***EXTRA\_NOTIFICATION\_TITLE***);  **content** = bundle.getString(JPushInterface.***EXTRA\_ALERT***);  }  **if**(**content**==**null**){  **content**=**""**;  }  //以上及时获取广播里的数据  **textView**.setText(**content**);  }  AlertDia |