

xposed模块入门开发

 h080294 (/u/22207f3e84de) [+ 关注](#)
0.1 2018.02.02 12:26* 字数 581 阅读 4766 评论 1 喜欢 1
(/u/22207f3e84de)

具体的配置环境和手机环境就不再赘述了。

电脑端：

Android studio

Android sdk

XposedBridgeApi jar包

手机端：

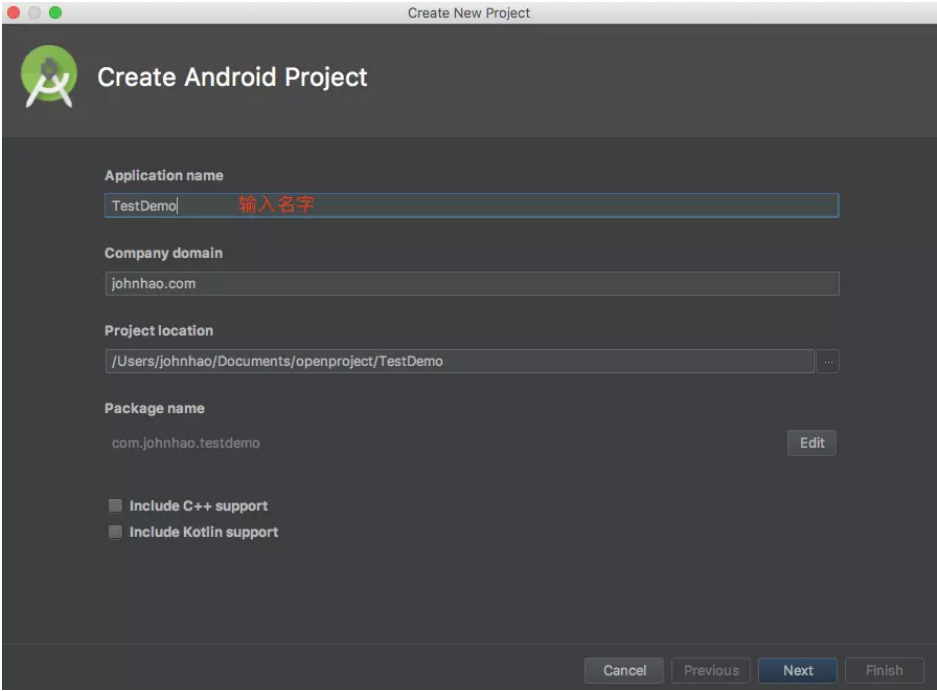
root

安装了xposed installer

已安装并激活xposed框架

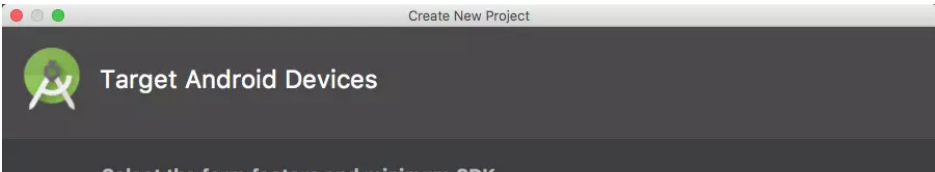
创建Android工程

其实就是一路next，创建工程完成后等待gradle加载完毕。

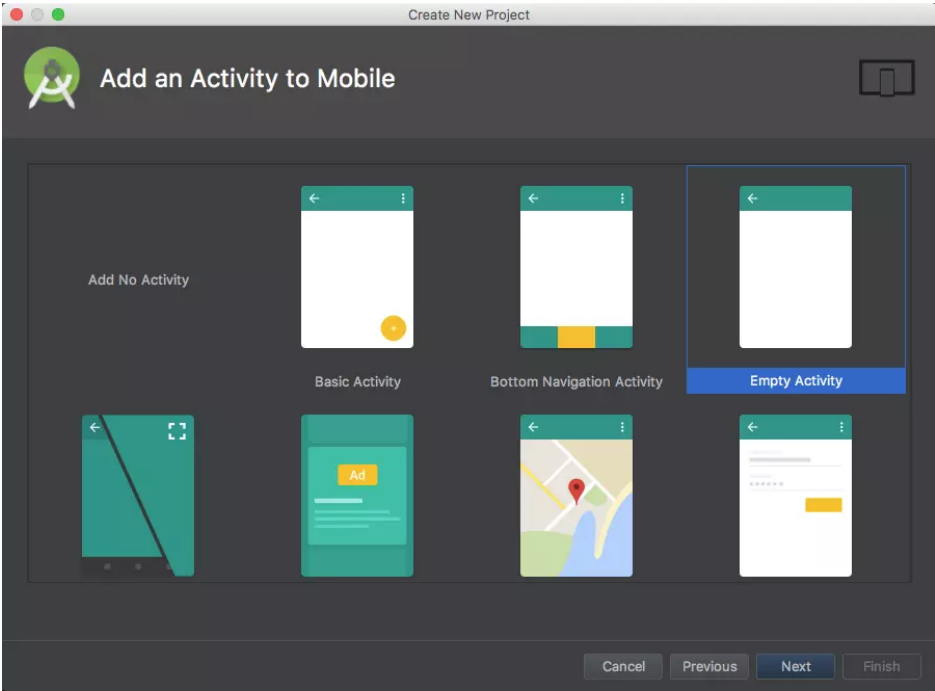


(/apps/redirect?
utm_source=side-
banner-click)

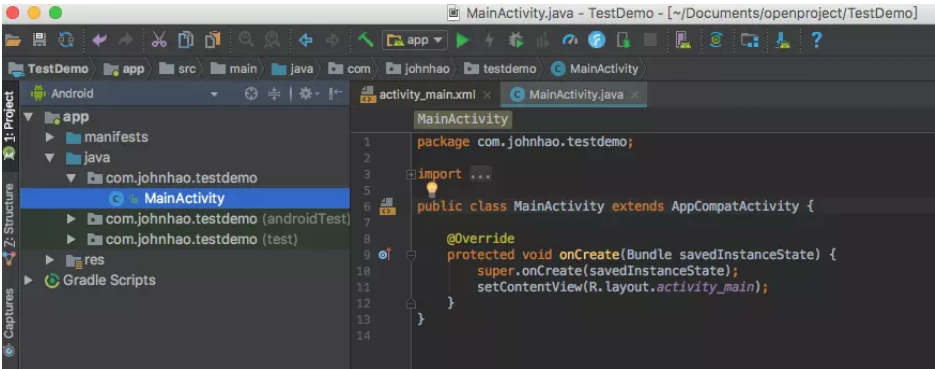
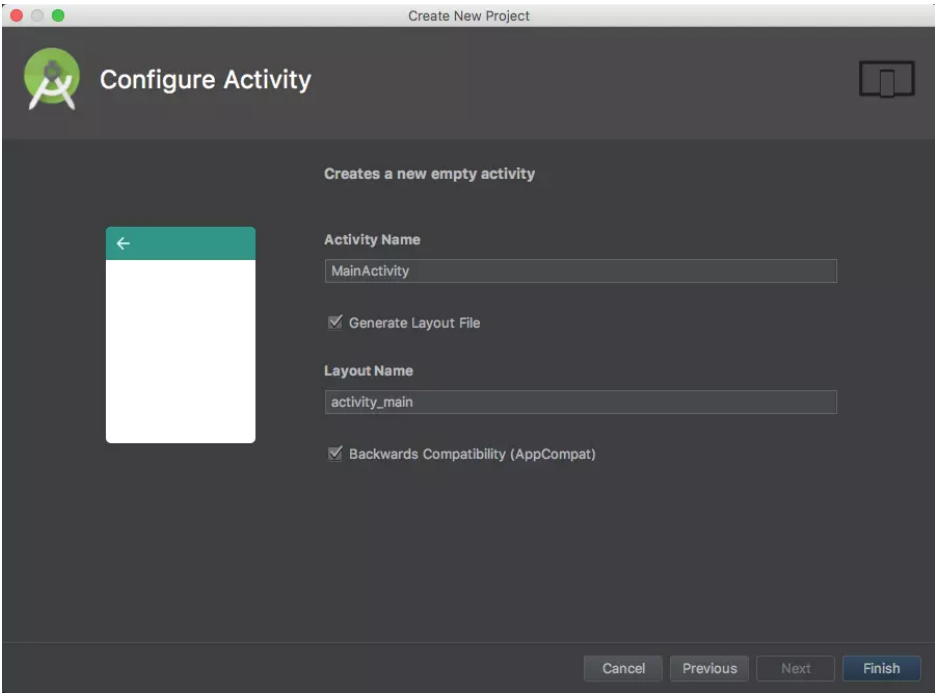
(https://log-
yex.youdao.com/ct?
slot=30edd91dd8637
7b8b-48e6-8a04-
5829f156bffb&yexi=f:
click.youdao.com%2f
7b8b-48e6-8a04-
5829f156bffb&iid=%7



(/apps/redirect?
utm_source=side-
banner-click)



(https://log-
yex.youdao.com/ct?
slot=30edd91dd8637
7b8b-48e6-8a04-
5829f156bffb&yexi=f
lick.youdao.com%2
7b8b-48e6-8a04-
5829f156bffb&iid=%7



导入XposedBridgeApi jar包

导入完成后，修改下app/build.gradle中的依赖声明。将XposedBridgeApi的依赖由implementation改成provided。改完后记得sync一下gradle。

```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
    provided files('lib/XposedBridgeApi-54.jar')
}
```

(/apps/redirect?
utm_source=side-
banner-click)

修改AndroidManifest

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="true"
    android:theme="@style/AppTheme">

    <meta-data
        android:name="xposedmodule"
        android:value="true" />
    <meta-data
        android:name="xposeddescription"
        android:value="这里填写xposed说明" />
    <meta-data
        android:name="xposedminversion"
        android:value="54" />

    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

</application>
```

(https://log-
yex.youdao.com/ct?
slot=30edd91dd8637
7b8b-48e6-8a04-
5829f156bffb&yexi=f
lick.youdao.com%2
7b8b-48e6-8a04-
5829f156bffb&iid=%7

xposedmodule：表示这是一个xposed模块

xposeddescription: 描述该模块的用途，可以引用string.xml中的字符串

xposedminversion：要求支持的Xposed Framework最低版本

模块实现

创建一个或者几个类，并实现IXposedHookLoadPackage,IXposedHookZygoteInit或者其他IXposedMod的子接口。

```
package de.robv.android.xposed.mods.tutorial;

import de.robv.android.xposed.IXposedHookLoadPackage;
import de.robv.android.xposed.XposedBridge;
import de.robv.android.xposed.callbacks.XC_LoadPackage.LoadPackageParam;
import android.util.Log;

public class TestDemo implements IXposedHookLoadPackage {
    public void handleLoadPackage(final LoadPackageParam lpparam) throws Throwable {
        XposedBridge.log("Loaded app: " + lpparam.packageName);
        Log.d("YOUR_TAG", "Loaded app: " + lpparam.packageName )
    }
}
```

XposedBridge.log会将日志输出到logcat，并写入日志文件
也可以用 android.util.Log输出到logcat

开始hook

这里举个例子，实现一下xposed的模块自检，检查自己的模块是否启动。就是在程序启动时，判断模块是否激活了。如果激活了就弹个toast提示模块已开启，如果因为某种未知原因导致激活失败，那么久弹个toast提示模块未开启。

实现也很简单，就是在MainActivity实现一个boolean方法，然后用xposed hook自己。能hook成功，自然代表模块成功启动了，反之亦然。

(/apps/redirect?utm_source=side-banner-click)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    if (!isModuleActive()){
        Toast.makeText(this, "模块未启动", LENGTH_LONG).show();
    }
    else {
        Toast.makeText(this, "模块已启动", LENGTH_LONG).show();
    }
}

private boolean isModuleActive(){
    return false;
}
```

hook isModuleActive函数

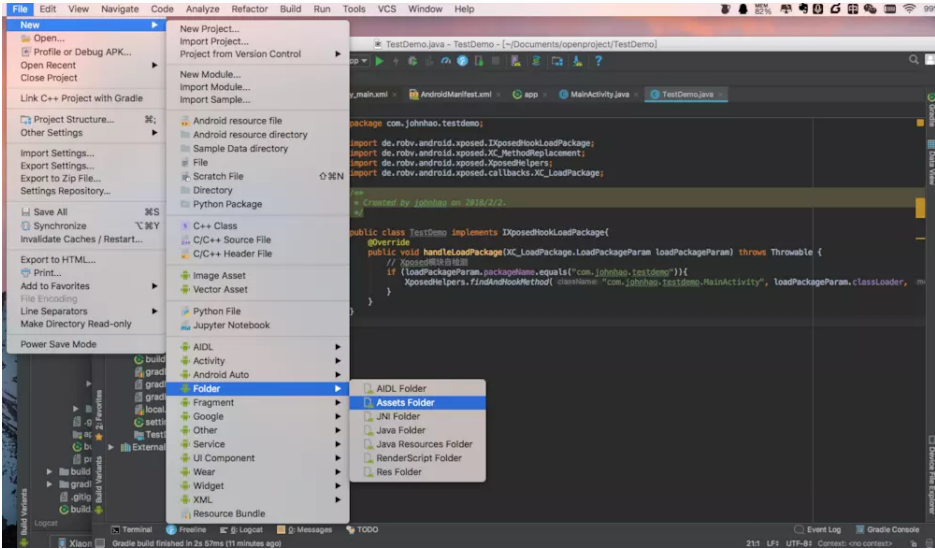
```
public void handleLoadPackage(XC_LoadPackage.LoadPackageParam loadPackageParam) {
    // Xposed模块自检
    if (loadPackageParam.packageName.equals("com.johnhao.testdemo")){
        XposedHelpers.findAndHookMethod("com.johnhao.testdemo.MainActivity", loadPackageParam.classLoader, "isModuleActive", new XC_MethodReplacement() {
            @Override
            protected boolean handleLoadPackage(XC_LoadPackage.LoadPackageParam loadPackageParam) throws Throwable {
                return true;
            }
        });
    }
}
```

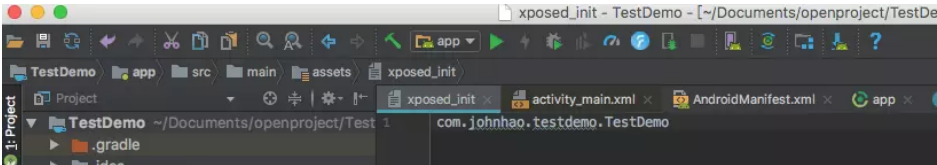
这样就ok了

(https://log-yex.youdao.com/ct?slot=30edd91dd86377b8b-48e6-8a04-5829f156bffb&yexi=click.youdao.com%2f7b8b-48e6-8a04-5829f156bffb&iid=%7

声明xposed入口

我们需要新建一个assets文件夹，并创建文件xposed_init，在里面填上xposed模块的入口。
这里我们声明自己的类“com.johnhao.testdemo.TestDemo”



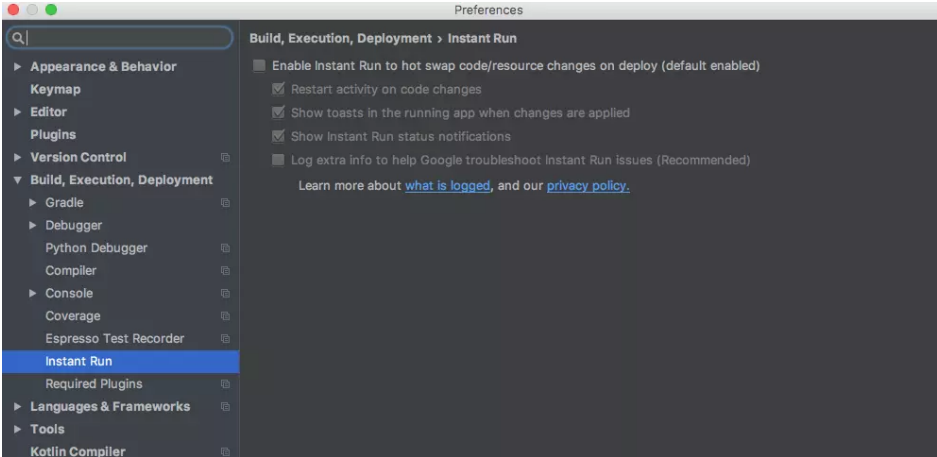


(/apps/redirect?
utm_source=side-
banner-click)

到这里，这个简单的模块就算开发完了。

模块安装

将这个工程，编译，打包，安装到已经支持Xposed的手机中。
需要注意的是，要关闭instant run功能！！

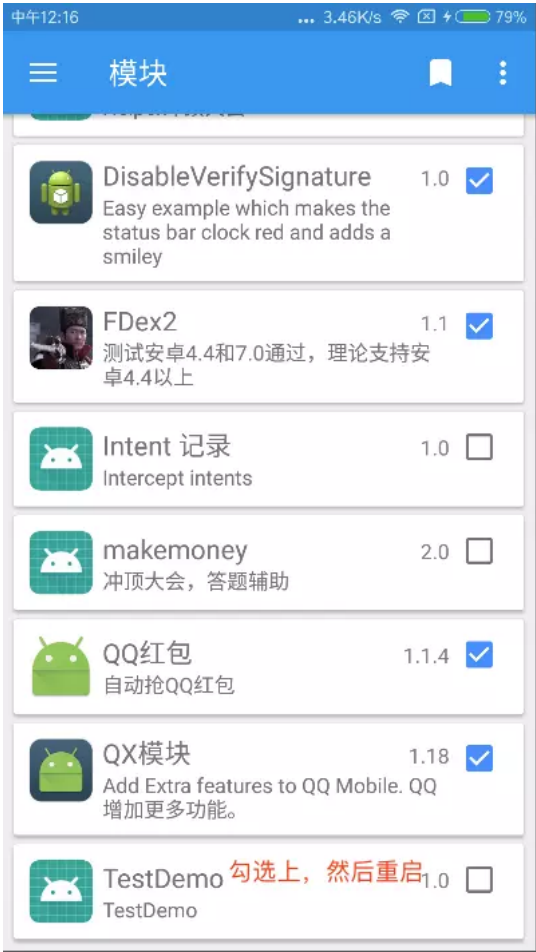


(https://log-
yex.youdao.com/ct?
slot=30edd91dd8637
7b8b-48e6-8a04-
5829f156bffb&yexi=f
click.youdao.com%2l
7b8b-48e6-8a04-
5829f156bffb&iid=%7

下面看下实际效果（为什么gif不能自己动呢，还得点击图片查看）：



勾选激活模块，重启设备



(/apps/redirect?utm_source=side-banner-click)

(https://log-yex.youdao.com/ct?slot=30edd91dd86377b8b-48e6-8a04-5829f156bffb&yexi=click.youdao.com%2f7b8b-48e6-8a04-5829f156bffb&iid=%7

我们再来启动试试（gif怎么弄才能自动循环播放呢）

