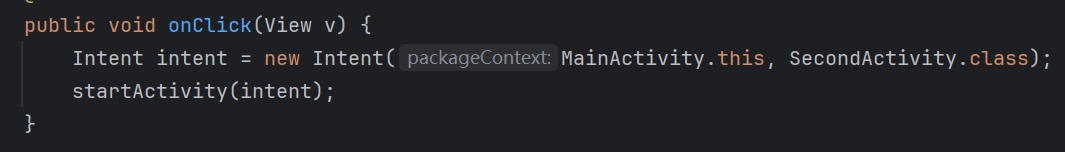
Activity Navigator 应用实现文档

功能说明：

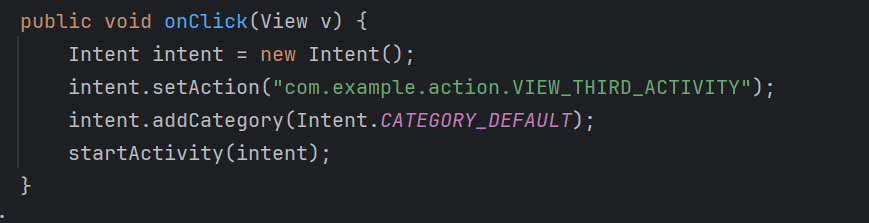
1. 显示跳转的实现
2. 在MainActivity中添加跳转按钮
3. 在MainActivity代码中使用 Intent(Context packageContext, Class<?> cls) 构造函数，直接指定目标 Activity 的类对象，调用 startActivity() 方法执行跳转，在SecondActivity中设置按钮点击监听器，使用finish方法结束当前Activity



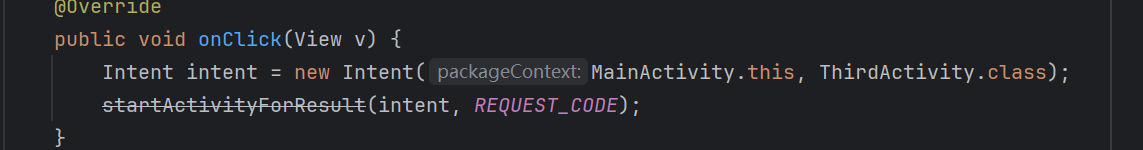
1. 隐式跳转的实现
2. 在AndroidManifest.xml中给ThirdActivity 的 <activity> 标签添加一个 <intent-filter>，设置Action和Category

****

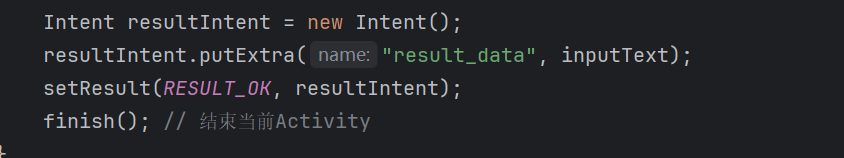
1. 在SecondActivity布局文件中再添加一个按钮，文本为“隐式跳转”，在SecondActivity的Java代码中，为该按钮设置点击监听器，创建隐式Intent，设置Action为自定义字符串，Category为Intent.CATEGORY\_DEFAULT



1. 带返回结果的跳转
2. 在MainActivity的布局文件中添加一个按钮和一个TextView，在MainActivity的Java代码中，为按钮设置点击监听器，使用startActivityForResult方法启动ThirdActivity，并传入请求码

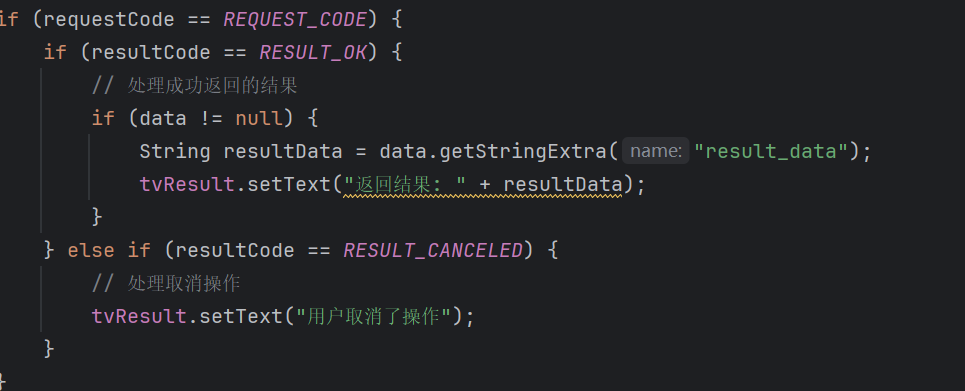
****

1. 在ThirdActivity的Java代码中，为“返回结果”按钮设置点击监听器，获取EditText的文本，创建Intent并放入额外数据，然后设置结果码为RESULT\_OK，并调用finish()。

****

3、为“返回取消”按钮设置点击监听器，设置结果码为RESULT\_CANCELED，并finish()。

在MainActivity中重写onActivityResult方法，根据请求码和结果码处理返回数据，更新TextView

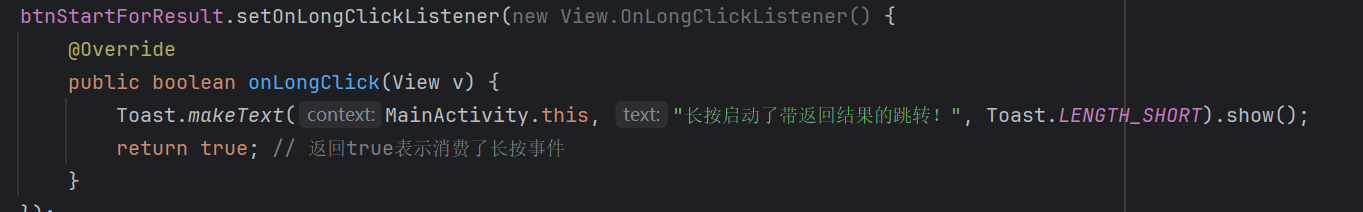
****

1. 长按提示功能

1、在MainActivity的setupClickListeners方法中，找到btnStartForResult按钮，并为其设置OnLongClickListener

2、在onLongClick方法中，使用Toast.makeText创建一个提示信息，内容为“长按启动了带返回结果的跳转！”，显示时长设置为Toast.LENGTH\_SHORT

3、返回true表示已经消费了长按事件，如果返回false，则会先触发长按事件，然后触发点击事件



1. 取消操作

1、在ThirdActivity的布局文件中，添加一个按钮，文本为“返回取消”

2、在ThirdActivity的Java代码中，找到这个按钮并设置点击监听器

3、在点击事件中，调用setResult(RESULT\_CANCELED)，表示用户取消了操作。然后调用finish()方法结束当前Activity

运行截图：



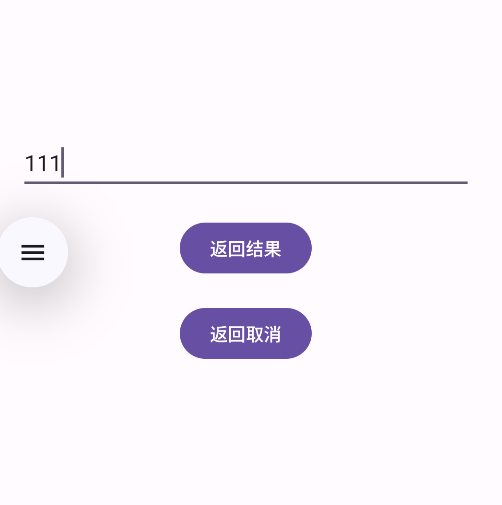
显式跳转：



隐式跳转：



启动带结果的跳转：





返回取消：



长按启动带结果的跳转：

