

# 关于onNewIntent你应该知道的

原创

且听风吟9527

2018-12-04 14:43:25

19790

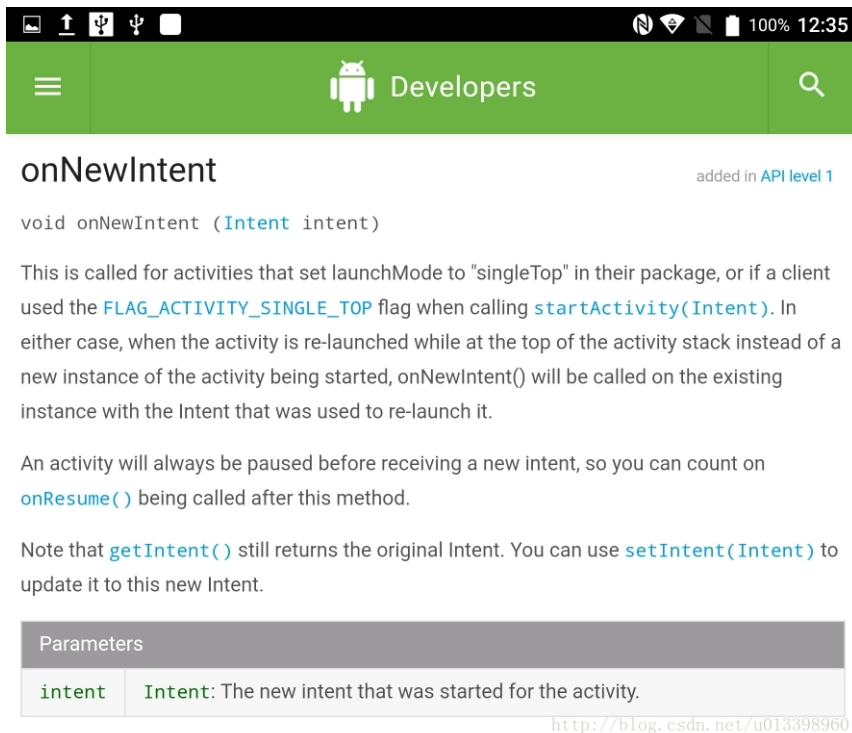
收藏 39

版权

文章标签: onNewIntent

启动模式

## 一、API描述如下



大概意思是当Activity被设以singleTop模式启动，当需要再次响应此Activity启动需求时，**会复用栈顶的已有Activity**，还会调用onNewIntent方法。并且，再接受新发送来的intent(onNewIntent方法) 之前，**一定会先执行onPause方法**。

## 二、onNewIntent与启动模式

前提:ActivityA已经启动过,处于当前应用的Activity任务栈中;

当ActivityA的LaunchMode为Standard时:

由于每次启动ActivityA都是启动新的实例，和原来启动的没关系，所以不会调用原来ActivityA的onNewIntent方法

当ActivityA的LaunchMode为SingleTop时:

如果ActivityA在栈顶,且现在要再启动ActivityA，这时会调用onNewIntent()方法，生命周期顺序为:

onCreate--->onStart--->onResume---onPause--->onNewIntent--->onResume

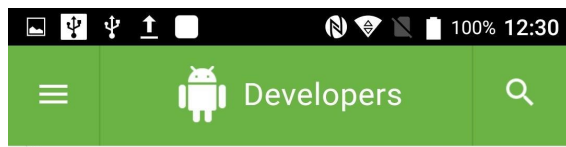
当ActivityA的LaunchMode为SingleInstance,SingleTask:

如果ActivityA已经在任务栈中，再次启动ActivityA，那么此时会调用onNewIntent()方法，生命周期调用顺序为:

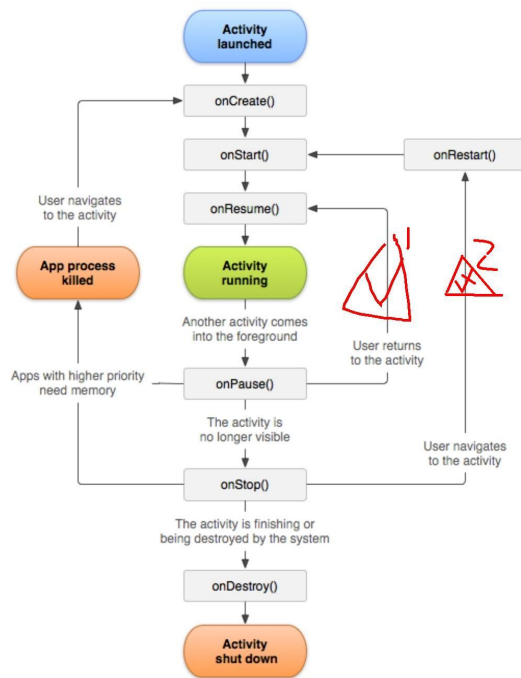
onPause--->跳转其它页面--->onCreate--->onStart--->onResume--->onPause--->跳转A--->onNewIntent--->onRestart--->onStart--->onResume

因此: onNewIntent在情况1下调用，在情况2不调用

更准确的说法是，只对SingleTop(且位于栈顶)，SingleTask和SingleInstance(且已经在任务栈中存在实例)的情况下，再次启动它们时才会调用，即只对startActivity有效，对仅仅从后台切换到前台而不再次启动的情形，不会触发onNewIntent



in.



There are three key loops you may be interested in



### 三、注意事项

通过前面我们知道，如果一个Activity已经启动过，并且存在当前应用的Activity任务栈中，启动模式为singleTask，singleInstance或singleTop(此时已在任务栈顶端)，那么在此启动或回到这个Activity的时候，不会创建新的实例，也就是不会执行onCreate方法，而是执行onNewIntent方法，如下所示：

```
1  protected void onNewIntent(Intent intent) {
2      super.onNewIntent(intent);
3
4      setIntent(intent); // must store the new intent unless getIntent() will return the old one
5  }
```

我们需要知道，在内存吃紧的情况下，系统可能会kill掉后台运行的 Activity，如果不巧要启动的那个activity实例被系统kill了，那么系统就会调用 onCreate 方法，而不调用 onNewIntent 方法。这里有个解决方法就是在 onCreate 和 onNewIntent 方法中调用同一个处理数据的方法，如下所示：

```
1  public void onCreate(Bundle savedInstanceState) {
2      super.onCreate(savedInstanceState);
3      setContentView(R.layout.main);
4      getNewIntent();
5  }
6
7  protected void onNewIntent(Intent intent) {
8      super.onNewIntent(intent);
9
10     setIntent(intent); // must store the new intent unless getIntent() will return the old one
11     getNewIntent();
12
13     private void getNewIntent(){
14         Intent intent = getIntent(); // use the data received here
15     }
```

注意onNewIntent()中的陷阱：

有时候，我们在多次启动同一个栈唯一模式下的activity时，在onNewIntent()里面的getIntent()得到的intent感觉都是第一次的那个数据。对，这里就是这个陷阱。因为它就是会返回第一个intent的数据。就是这么坑。

原因就是我们没有在onNewIntent()里面设置setIntent()，将最新的intent设置给这个activity实例。

```
1  @Override
2  protected void onNewIntent(Intent intent) {
3      super.onNewIntent(intent);
4      setIntent(intent); //设置新的intent
5
6      int data = getIntent().getIntExtra("tanksu", 0); //此时的到的数据就是正确的了
7  }
```

在这里，如果你没有调用setIntent()方法的话，则getIntent()获取的intent都只会是最初那个intent（Note that getIntent() still returns the original Intent. You can use setIntent(Intent) to update it to this new Intent.）。