

# Android启动模式（LaunchMode）

## 1,Standard

默认启动模式

启动就创建。  
单独服务于一个Intent

撰写邮件的Activity或者发布社交网络状态的Activity都是采用这种Activity的例子。如果你希望Activity单独服务于一个Intent，就可以考虑standard启动模式。

---

## 2,SingleTop

如果在栈顶已经有一个相同类型的Activity实例，Intent不会再创建一个Activity，而是通过onNewIntent()被发送到现有的Activity。

这种启动模式的用例之一就是搜索功能。假设我们创建了一个搜索框，点击搜索的时候将导航到一个显示搜索结果列表的SearchActivity中，为了更好的用户体验，这个搜索框一般也会被放到SearchActivity中，这样用户想要再次搜索就不需要按返回键。

想像一下，如果每次显示搜索结果的时候我们都启动一个新的activity，10次搜索10个activity，那样当我们想返回最初的那个activity的时候需要按10次返回。

---

## 3,SingleTask

singleTask模式的Activity只允许在系统中有一个实例。如果系统中已经有了一个实例，持有这个实例的任务将移动到顶部，同时intent将被通过onNewIntent()发送。如果没有，则会创建一个新的Activity并置放在合适的任务中。

在同一个应用中：

如果系统中还没有singleTask的Activity，会新创建一个，并放在同一任务的栈顶。

但是如果已经存在，singleTask Activity上面的所有Activity将以合适的方式自动销毁，让我们想要显示的Activity处于栈顶。同时Intent也会通过onNewIntent()方法发送到这个singleTask Activity。

官方文档中提到一个问题：

系统会创建一个新的任务，并将这个Activity实例化为新任务的根部（root）。

如果你希望singleTask Activity表现的和文档中描述的一致，你需要为singleTask Activity设置taskAffinity属性。

```
1. <activity
2.     android:name=".SingleTaskActivity"
3.     android:label="singleTask launchMode"
4.     android:launchMode="singleTask"
5.     android:taskAffinity="">
```

和其他应用一起工作的时候：

一旦intent是从另外的应用发送过来，并且系统中没有任何Activity的实例，则会创建一个新的任务，并且新的Activity被作为根Activity创建。

这种模式的应用案例有。邮件客户端的收件箱或者社交网络的时间轴。这些Activity一般不会设计成拥有多个实例，singleTask可以满足。但是在使用这种模式的时候必须要明智，因为有些Activity会在用户不知情的情况下被销毁。

---

## 4,SingleInstance

这个模式非常接近于singleTask，系统中只允许一个Activity的实例存在。区别在于持有这个Activity的任务中只能有一个Activity：即这个单例本身。

总是在新的任务中开启，并且这个新的任务中有且只有这一个实例，也就是说被该实例启动的其他activity会自动运行于另一个任务中。当

再次启动该**activity**的实例时，会重用已存在的任务和实例。并且会调用这个实例的**onNewIntent()**方法，将**Intent**实例传递到该实例中。和**singleTask**相同，同一时刻在系统中只会存在一个这样的**Activity**实例。

---

参考资料:

<http://www.jcodecraeer.com/a/anzhuokaifa/androidkaifa/2015/0520/2897.html>