

首页 > 程序开发 > 移动开发 > Android > 正文

Android activity四大启动模式详解

2015-03-040 个评论来源：路在脚下收藏我要投稿

Activity中四大启动模式

在AndroidManifest.xml中，有一个默认的activity 在里面可以设置activity启动模式，android:launchMode=" "，该属性用于配置Activity的加载模式，该属性支持4中属性 每不同的模式出现不同的效果，下面详解启动模式。

standard:标准模式, 默认加载模式

singleTop: Task顶单例模式

singleTask:Task内单例模式

singleInstance:全局单单例模式

1 Activity为什么要用指定模式？？

首先介绍下Android对Activity的管理：Android采用Task来管理多个Activity，当我们启动一个Activity时，系统就会创建一个Task，然后启动这个Activity的入口。

Android并未给Task提供API, 只能通过调用Activity的getTaskId() 方法获取它所在的Task的ID, 我们可以把Task理解为Activity 栈，Task以栈来管理Activity。

2. 四大启动模式的理解

standard加载模式：

每次通过这种模式启动Activity时,Android总会为启动的Activity创建一个新的实例,并将该Activity 添加到当前Task栈中，这种模式不会创建新的Task, 只是将新Activity添加到原有的Task 中

standard加载模式示意图

第三次启动

第二次启动

第一次启动

创建第三个Activity实例

创建第二个Activity实例

创建第一个Activity实例

Activity栈

singleTop模式

文章 推荐

- win7激活工具
- win10激活工具
- win7激活
- office2010激活密钥
- windows7激活密钥
- office2010激活工具
- 小马激活工具
- win10激活工具

点击排行

- Android：IPC之AIDL的学习和总结
- Android 网络
- Android之内容提供者Content Provide
- Android基础——JSON数据的全方位解析
- Android SDK上手指南：Activity与生命
- Android应用框架之PackageManagerServ
- Android官方开发文档Training系列课程
- Android实战教程—第三十八话《自定义

新闻排行榜天

- Win10小马 原版镜像激活工具 永
- KMS通用激活工具v2016.05.
- Microsoft Toolkit (
- C++ QT库开发
- Android开发过程中遇到的问题集
- Windows10激活 Win10激
- Android图像常用压缩技术
- cocos2d
- win7永久激活码免费分享
- SQL Server基于正则表达式的

如果在任务的栈顶正好存在该Activity的实例，就重用该实例，否则就会创建新的实例并放入栈顶(即使栈中已经存在该Activity实例，只要不在栈顶，都会创建实例)。

singleTask模式

被启动的Activity在同一个Task内只有一个Activity实例，具体分为如下三种情况：

<1>. 如果启动的目标Activity不存在Task栈中，系统将会创建一个目标Activity实例，并将它加入到Task栈顶

<2>. 如果启动的目标Activity已存在Task栈顶，此时模式和singleTop模式相同

<3>. 若果启动的目标Activity已存在但没有位于Task栈顶，系统将会把该目标Activity上面的所有Activity移除Task栈，使该Activity置于Task栈顶

singleInstance模式

这种加载模式下，无论从哪个Task中启动目标Activity，只会创建一个目标Activity实例，并会使用一个全新的Task栈来装载该Activity实例。具体可分为两种情况：

<1>. 如果创建的目标Activity不存在，系统先会创建一个全新的Task，接着创建一个Activity实例，然后将该目标Activity加入到新的Task栈顶

<2>. 如果创建的目标Activity已经存在，无论在哪个Task栈中，系统将会把Activity所在的栈置于前台。

注意：采用singleInstance加载模式的Activity总是位于Task栈顶，并且Activity所在的Task栈只包含该Activity。

3. 四大启动模式示例

standard

其中standard是系统默认的启动模式。

下面通过实例来演示standard的运行机制：

```
1 private Button btn_mode;
2 @Override
3 public void onCreate(Bundle savedInstanceState) {
4     super.onCreate(savedInstanceState);
5     setContentView(R.layout.activity_main);
6 }
7
```

```
8 text_show = (TextView) this.findViewById(R.id.text_show);
9
10
11 text_show.setText(this.toString());
12
13
14 btn_mode = (Button) this.findViewById(R.id.btn_mode);
15
16
17 }
18
19
20 //按钮单击事件
21 public void LaunchStandard(View v){
22     startActivity(new Intent(this,MainActivity.class));
23
24
25     text_show.setText(this.toString());
```

初始化界面如下：



当点击按钮时，会创建新的Activity，通过TextView@后16进制数的显示即可看出，点击两次分别界面如下：



此时，我们分析栈内部的运行机制：（依次从栈顶向上）



因此，这种Standard模式是每次都会创建新的Activity对象，当点击返回按钮时，他会将栈顶(当前Activity)消灭，然后跳 到下一层，例如如果现在Activity是44ed8c50，那么当我们点击返回时Activity会变为44f28a48，不过此时在这个 Activity中再次点击按钮创建对象时，它会另外创建新的Activity对象，这种模式可能大多数情况下不是我们需要的，因为对系统性能的消费过 大。

singleTop

从上面的解释中即可知道，在每次使用新的Activity时会自动检测栈顶的当前Activity是否需要引用的Activity，如果是则直接引用此Activity，而不会创建新的Activity。

我们在刚才的界面中加入一个”启动singletop模式”按钮，当点击时出现我们创建的singletop中，在Activity singletop中有一个按钮，启动singletop模式，表示启动当前Activity，由于我们在清单文件中配置Activity的启动模式为 singleTop，因此此时不会再创建而是利用当前栈顶的singleTop Activity：

```
1 </activity>
2
```

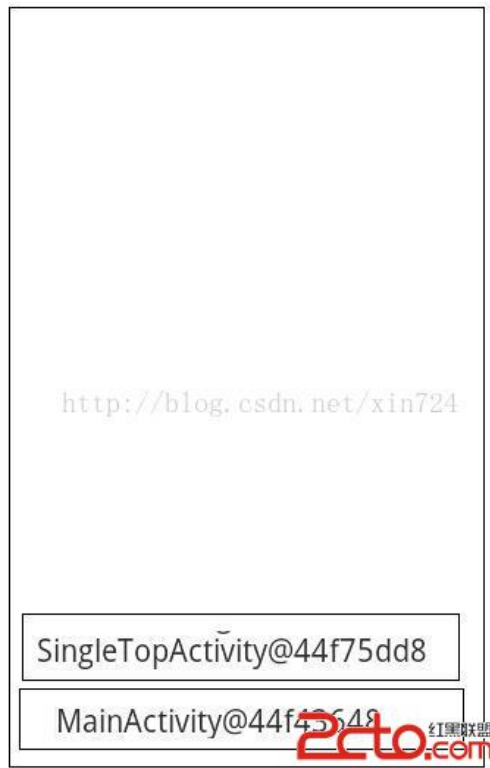
初始化界面如下：



当点击”启动singletop模式”按钮时



我们分析它的运行机制，可知，当程序运行到此时，栈中的数据形式为：



当我们在上面界面中点击”启动singleTop模式”按钮时，由于此Activity设置的启动模式为singleTop，因此它首先会 检测当前栈顶是否为我们请求的Activity对象，经验证成立，因此它不会创建新的Activity，而是引用当前栈顶的Activity。



虽然它不会创建新的Activity对象，不过它每次回调用onNewIntent()方法：

```
1  @Override
2      protected void onNewIntent(Intent intent) {
3          // TODO Auto-generated method stub
4          super.onNewIntent(intent);
5
6
7          Toast.makeText(this, new Date().toString(), 1).show();
8      }
```

我们为此方法编写代码输出当前日期，则在每次点击上面按钮时会输出当前日期。

singleTask

此启动模式和singleTop在名字上即可看出区别，即singleTop每次只检测当前栈顶的Activity是否是我們需要请求创建的，而 singleTask则会检测栈中全部的Activity对象，从上向下，如果检测到是我們所请求的则会消灭此Activity对象上面的对象，直接把检测到的我们需要的Activity置为栈顶。

我们创建一个SingleTaskActivity，此界面中包含一个启动MainActivity和启动SingleTaskActivity按钮。

初始化:



当点击”启动singletop模式“按钮时:



在此界面中点击第二个按钮”启动singleTask模式“按钮，根据定义会检测当前栈中是否有此Activity对象，因此显示的还是当前的Activity，不会重新创建；

再点击”启动Standard模式“按钮，由于MainActivity的启动模式为standard，所以在此会重新创建一个MainActivity对象；



此时栈中的排列顺序数据格式为：



当在上面界面中点击”启动singleTask模式”按钮时，由于检测到当期栈中第二个为我们要创建的Activity，会将最上面的MainActivity消灭，然后将SingleTaskActivity设置为栈顶：



SingleInstance

此启动模式和我们使用的浏览器工作原理类似，我们都知道在多个程序中访问浏览器时，如果当前浏览器没有打开，则打开浏览器，否则会在当前打开的浏览器中访问。此模式会节省大量的系统资源，因为他能保证要请求的Activity对象在当前的栈中只存在一个。

上面即为Android中的四种启动模式，我们在开发Android项目时会经常使用到，巧妙设置Activity的启动模式会节省系统开销和程序运行效率。

点击复制链接 与好友分享!

回本站首页

相关TAG标签

模式

上一篇：Android、web中的图片和语音的加密

下一篇：android调用系统的分享功能实现分享到短信, 邮件和其他

相关文章

- Android Ap开发设计模式第二篇：适配
- Android Ap开发设计模式第四篇：工厂
- Android Ap开发设计模式第六篇：原型
- Android Ap开发设计模式第八篇：抽象
- Activity的四种加载模式
- Android Ap开发设计模式第一篇：迭代
- Android Ap开发设计模式第五篇：单件
- Android Ap开发设计模式第七篇：生成
- Android动画效果Widget（飞行模式）
- Android歌词秀设计思路（6）运用Proxy

热门专题推荐

python

div+css

css教程

html5

html教程

jquery

Android


SDK

php

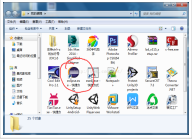
mysql

oracle


图文推荐




【Android】给我一



unity与Android的交互



自定义view实现水波纹



利用ViewPager实现引

登录

来说两句吧...

还没有评论，快来抢沙发吧！

红黑联盟正在使用畅言