**/Android面试之---讲一下你对Activity的了解(面试必问的).**

2010-02-28 21:11:25

标签：[Android](http://blog.51cto.com/tag-Android.html) [面试](http://blog.51cto.com/tag-%E9%9D%A2%E8%AF%95.html) [Activity](http://blog.51cto.com/tag-Activity.html) [移动开发](http://blog.51cto.com/tag-%E7%A7%BB%E5%8A%A8%E5%BC%80%E5%8F%91.html) [休闲](http://blog.51cto.com/tag-%E4%BC%91%E9%97%B2.html)

原创作品，允许转载，转载时请务必以超链接形式标明文章 [原始出处](http://weizhulin.blog.51cto.com/1556324/311495) 、作者信息和本声明。否则将追究法律责任。<http://weizhulin.blog.51cto.com/1556324/311495>

这篇文章会涉及到以下几个内容

一 Activity的生命周期  
二 让Activity变成一个窗口：Activity属性设定  
三 你后台的Activity被系统 回收怎么办：onSaveInstanceState  
四 调用与被调用：我们的通信使者 - Intent

一 Activity的生命周期

   和其他手机 平台 的应用 程序 一样，Android的应用程序 的生命周期是被统一掌控 的，也  
   就是说我们写的应用程序命运掌握在别人（系统）的手里，我们不能改变它，只能学习 并  
   适应它。

简单地说一下为什么是这样：我们手机在运行 一个应用程序的时候，有可能打进来电话  
   发进来短信 ，或者没有电了，这时候程序都会被中断，优先去服务电话的基本功能 ，另  
   外系统也不允许你占用太多资源 ，至少要保证电话功能吧,所以资源不足的时候也就有可    
   能被干掉。

   言归正传，Activity的基本生命周期如下代码 所示：

Java 代码

1. **public**
2. **class** MyActivity **extends** Activity {
3. **protected**
4. **void** onCreate(Bundle savedInstanceState);
6. **protected**
7. **void** onStart();
9. **protected**
10. **void** onResume();
12. **protected**
13. **void** onPause();
15. **protected**
16. **void** onStop();
18. **protected**
19. **void** onDestroy();
20. }
22. **public** **class** MyActivity **extends** Activity {     **protected** **void** onCreate(Bundle savedInstanceState);     **protected** **void** onStart();     **protected** **void** onResume();     **protected** **void** onPause();     **protected** **void** onStop();     **protected** **void** onDestroy();   }

   你自己写的Activity会按需要 重载这些方法，onCreate是免不了的，在一个Activity正常启动的过程中，他们被调用的顺序是 onCreate -> onStart -> onResume, 在Activity被干掉的时候顺序是onPause -> onStop -> onDestroy ，这样就是一个完整的生命周期，但是有人问了 ，程序正运行着呢来电话了，这个程序咋办？中止了呗，如果中止的时候新出的一个Activity是全屏的那么：onPause->onStop ，恢复的时候onStart->onResume ，如果打断  这个应用程序的是一个Theme为Translucent 或者Dialog 的Activity那么只是onPause ,恢复 的时候onResume 。

   详细介绍一下这几个方法中系统在做什么以及我们应该做什么：

   onCreate:   在这里创建界面 ，做一些数据 的初始化工作

   onStart:    到这一步变成用户可见不可交互 的

   onResume:   变成和用户可交互 的，（在activity 栈系统通过栈的方式管理这些个         
                      Activity的最上面，运行完弹出栈，则回到上一个Activity)

   onPause:     到这一步是可见但不可交互 的，系统会停止动画 等消耗CPU 的事情  
                    从上文的描述已经知道，应该在这里保存你的一些数据,因为这个时候  
                    你的程序的优先级降低，有可能被系统收回。在这里保存的数据，应该在

                    onResume里读出来，注意：这个方法里做的事情时间要短，因为下一  
                    个activity不会等到这个方法完成才启动

   onstop:     变得不可见 ，被下一个activity覆盖了

   onDestroy: 这是activity被干掉前最后一个被调用方法了，可能是外面类调用finish方  
                     法或者是系统为了节省空间将它暂时性的干掉，可以用isFinishing()来判  
                     断它，如果你有一个Progress Dialog在线程中转动，请在onDestroy里   
                     把他cancel掉，不然等线程结束的时候，调用Dialog的cancel方法会抛  
                     异常的。

onPause，onstop， onDestroy，三种状态 下 activity都有可能被系统干掉   
为了保证程序的正确性，你要在onPause()里写上持久层操作的代码，将用户编辑的内容都保存到存储介质上（一般都是数据库 ）。实际工作中因为生命周期的变化而带来的问题也很多，比如你的应用程序起了新的线程在跑，这时候中断了，你还要去维护那个线程，是暂停还是杀掉还是数据回滚，是吧？因为Activity可能被杀掉，所以线程中使用的变量和一些界面元素就千万要注意了，一般我都是采用Android的消息机制 [Handler,Message]来处理多线程和界面交互的问题。这个我后面会讲一些，最近因为这些东西头已经很大了，等我理清思绪再跟大家分享。

二 让Activity变成一个窗口：Activity属性设定   
      
   讲点轻松的吧,可能有人希望做出来的应用程序是一个漂浮在手机主界面的东西，那么很  
简单你只需要设置 一下Activity的主题就可以了在AndroidManifest.xml 中定义 Activity的  
地方一句话：

Xml代码

1. android :theme="@android:style/Theme.Dialog"
3. android:theme="@android:style/Theme.Dialog"

这就使你的应用程序变成对话框的形式弹出来了，或者

Xml代码

1. android:theme="@android:style/Theme.Translucent"
3. android:theme="@android:style/Theme.Translucent"

就变成半透明的，[友情提示-.-]类似的这种activity的属性可以在android.R.styleable 类的AndroidManifestActivity 方法中看到，AndroidManifest.xml中所有元素的属性的介绍都可以参考这个类android.R.styleable

上面说的是属性名称，具体有什么值是在android.R.style中 可以看到，比如这个"@android:style/Theme.Dialog" 就对应于android.R.style.Theme\_Dialog ,（'\_'换成'.' <--注意：这个是文章内容不是笑脸）就可以用在描述文件 中了,找找类定义和描述文件中的对应关系就都明白了。

三 你后台的Activity被系统回收怎么办：onSaveInstanceState   
     
   当你的程序中某一个Activity A 在运行时中，主动或被动地运行另一个新的Activity B   
这个时候A会执行

Java代码

1. **public**
2. **void** onSaveInstanceState(Bundle outState) {
3. **super**.onSaveInstanceState(outState);
4. outState.putLong("id", 1234567890);
5. }
7. **public** **void** onSaveInstanceState(Bundle outState) {    **super**.onSaveInstanceState(outState);    outState.putLong("id", 1234567890);}

B 完成以后又会来找A, 这个时候就有两种情况，一种是A被回收，一种是没有被回收，被回  
收的A就要重新调用onCreate()方法，不同于直接启动的是这回onCreate()里是带上参数  
savedInstanceState，没被收回的就还是onResume就好了。

savedInstanceState是一个Bundle对象，你基本上可以把他理解为系统帮你维护的一个Map对象。在onCreate()里你可能会用到它，如果正常启动onCreate就不会有它，所以用的时候要判断一下是否为空。

Java代码

if(savedInstanceState != null){     
     long id = savedInstanceState.getLong("id");     
}

if(savedInstanceState != null){     long id = savedInstanceState.getLong("id");}

就像官方的Notepad教程 里的情况，你正在编辑某一个note，突然被中断，那么就把这个note的id记住，再起来的时候就可以根据这个id去把那个note取出来，程序就完整一些。这也是看你的应用需不需要保存什么，比如你的界面就是读取一个列表，那就不需要特殊记住什么，哦， 没准你需要记住滚动条的位置...

四 调用与被调用：我们的通信使者Intent

要说Intent了，Intent就是这个这个意图 ，应用程序间Intent进行交流，打个电话啦，来个  
电话啦都会发Intent, 这个是Android架构的松耦合的精髓部分，大大提高了组件的复用性，比如你要在你的应用程序中点击按钮，给某人打电话，很简单啊，看下代码先：

Java代码

Intent intent = new Intent();     
intent.setAction(Intent.ACTION\_CALL);     
intent.setData(Uri.parse("tel:" + number));     
startActivity(intent);

Intent intent = new Intent(); intent.setAction(Intent.ACTION\_CALL); intent.setData(Uri.parse("tel:" + number)); startActivity(intent);

扔出这样一个意图，系统看到了你的意图就唤醒了电话拨号程序，打出来电话。什么读联系人，发短信啊，邮件啊，统统只需要扔出intent就好了，这个部分设计 地确实很好啊。

那Intent通过什么来告诉系统需要谁来接受他呢？  
通常使用Intent有两种方法，第一种是直接说明需要哪一个类来接收代码如下:

Java代码

Intent intent = new Intent(this, MyActivity.class);     
intent.getExtras().putString("id", "1");     
tartActivity(intent);

Intent intent = new Intent(this, MyActivity.class);intent.getExtras().putString("id", "1");tartActivity(intent);

第一种方式很明显，直接指定了MyActivity为接受者,并且传了一些数据给MyActivity，在MyActivity里可以用getIntent()来的到这个intent和数据。

第二种就需要先看一下AndroidMenifest中的intentfilter的配置了

Xml代码

<intent-filter>   
    <action  
android:name="android.intent.action.VIEW"  
/>   
    <action  
android:value="android.intent.action.EDIT"  
/>   
    <action  
android:value="android.intent.action.PICK"  
/>   
    <category  
android:name="android.intent.category.DEFAULT"  
/>   
    <data  
android:mimeType="vnd.android.cursor.dir/vnd.google.note"  
/>   
</intent-filter>

<intent-filter>    <action android:name="android.intent.action.VIEW" />    <action android:value="android.intent.action.EDIT" />    <action android:value="android.intent.action.PICK" />    <category android:name="android.intent.category.DEFAULT" />    <data android:mimeType="vnd.android.cursor.dir/vnd.google.note" /></intent-filter> 这里面配置用到了action, data, category这些东西，那么聪明的你一定想到intent里也会有这些东西，然后一匹配不就找到接收者了吗？

action其实就是一个意图的字符串名称。  
上面这段intent-filter的配置文件说明了这个Activity可以接受不同的Action，当然相应的程序逻辑也不一样咯,提一下那个 mimeType,他是在ContentProvider里定义的，你要是自己实现一个ContentProvider就知道了，必须指定 mimeType才能让数据被别人使用。

不知道原理说明白没，总结一句，就是你调用别的界面不是直接new那个界面，而是通过扔出一个intent，让系统帮你去调用那个界面，这样就多么松藕合啊，而且符合了生命周期被系统管理的原则。

想知道category都有啥，Android为你预先定制好的action都有啥等等，请亲自访问官方链接Intent

ps:想知道怎么调用系统应用程序的同学，可以仔细看一下你的logcat，每次运行一个程序的时候是不是有一些信息比如:  
Starting activity: Intent { action=android.intent.action.MAINcategories={android.intent.category.LAUNCHER} flags=0x10200000comp={com.android.camera/com.android.camera.GalleryPicker} }  
再对照一下Intent的一些set方法，就知道怎么调用咯，希望你喜欢：）