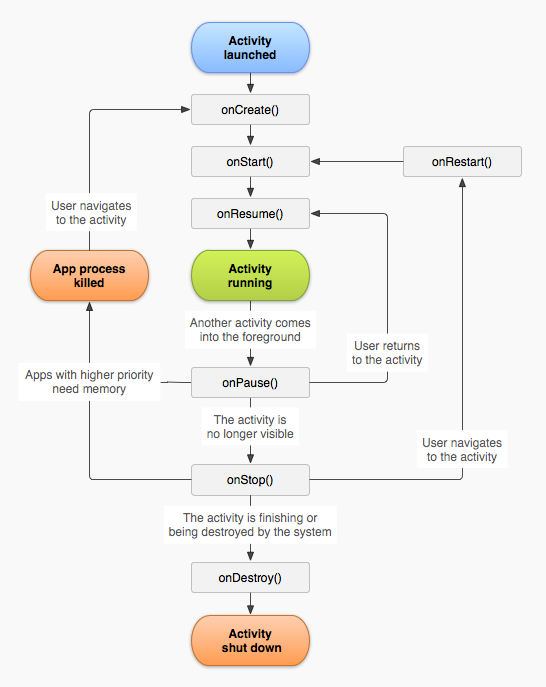
**Android面试题整理**

1. Activity
2. 生命周期



1. 切换屏幕

不设置Activity的android:configChanges时，切屏会重新调用各个生命周期

默认首先销毁当前activity,然后重新加载

设置Activity的android:configChanges="orientation|keyboardHidden"时，切屏不会重新调用各个生命周期，只会执行onConfigurationChanged方法

1. 定义成对话框模式

第一种方法：<activity android:name=".activity2" android:theme="@android:style/Theme.Dialog"></activity>

第二种方法：

<style name="dialog" parent="@android:style/Theme.Dialog">

<item name="android:windowFrame">@null</item>

<item name="android:windowIsFloating">true</item>

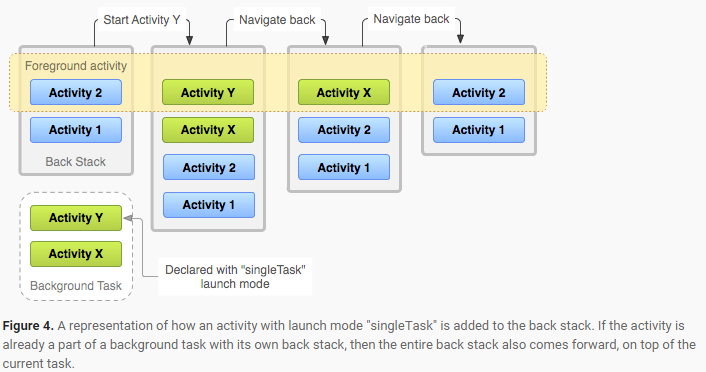
<item name="android:windowIsTranslucent">true</item>

<item name="android:windowNoTitle">true</item>

<itemname="android:windowBackground">@android:color/transparent</item>

</style>

1. 数据存储
2. 一般是在onSaveInstanceState中保存activity的数据，然后在Oncreate或者[onRestoreInstanceState(Bundle)](http://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle))中获取保存的数据，使用这种方法保存数据的前提是这个Activity必须有机会能够被恢复，假如A启动了B，然后从B返回到A，就可以在A中采用这个方法保存A的运行状态，但是B就不行，因为B没有机会重新加载，因为再次在A中打开B都是一个新的Task。这个方法在onStop()方法之前调用，但是不能确定是在onPause()之前或者之后调用。
3. 在onPause()中保存Activity的状态和其他数据是google官方推荐的方式，因为onPause()之后是onResume(),或者onStop(),如果是onStop()，那么a的存储方式就可行，但是如果是onResume()，那么onSaveInstanceState就不会执行，导致数据丢失。
4. 综上所述，a适合保存视图的状态，b适合保存持久的数据，包括保存关键的数据到数据库等，因为在a中不能保证onSaveInstanceState每次都运行。
5. 退出多个Activity
6. 使用一个容器记住每一个打开的activity，并将其假如到容器中，在每一个activity的onDestory()方法中移除这个activity的实例，在最后退出应用的时候遍历容器并将每一个activity结束掉。
7. 使用广播，在每一个打开的activity中注册一个广播，在程序要退出的时候发送结束的广播，此时每一个打开的activity会自行关闭。
8. 加载模式(LaunchMode)
9. Standard ,默认模式，在一个任务里面为每一个activity创建一个实例，一个activity能够被实例化多次，每一个实例可以属于多个任务，一个任务可以有多个实例。
10. Singletop，单顶模式([FLAG\_ACTIVITY\_SINGLE\_TOP](http://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_SINGLE_TOP))，即A B C D，此时要添加D，但是在原来的task中D的加载模式为单顶模式，那么原Task中的D就会执行onNewIntent()方法，不会重新创建一个D的实例，而是变成A B C D，但是如果原Task中D的加载模式是默认模式的话就会变成A B C D D。
11. SingleTask,单任务模式([FLAG\_ACTIVITY\_NEW\_TASK](http://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_NEW_TASK))，例如activity A打开浏览器，之后退出浏览器，此时activity B打开浏览器，这个浏览器在A中打开时相当于SingleTask模式，在B中打开时相当于返回之前已经打开的实例，一个典型的应用关系图如下：



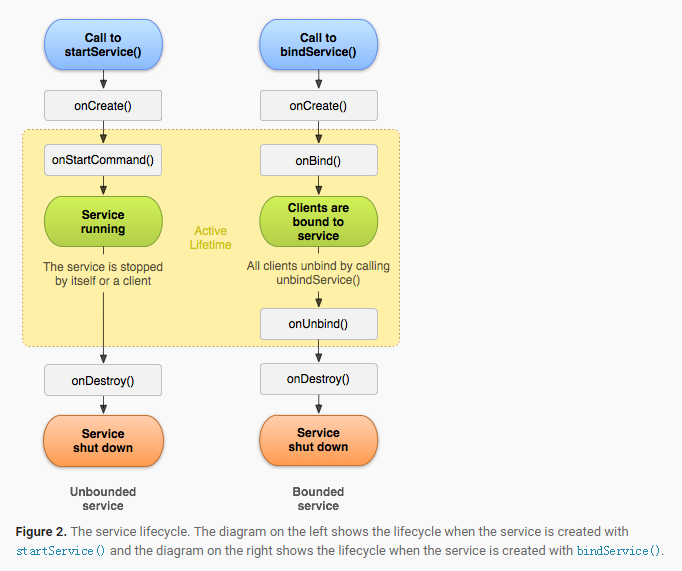
1. singleInstance,单例模式，即在该task中只有一个Activity的实例，任何从该task启动的activity都在另外一个task中。
2. [FLAG\_ACTIVITY\_CLEAR\_TOP](http://developer.android.com/reference/android/content/Intent.html#FLAG_ACTIVITY_CLEAR_TOP),清除该activity顶层的activity，意思是清除在这个实例上面的所有实例，然后使用这个模式的activity进入resume()的状态。

<http://developer.android.com/guide/components/tasks-and-back-stack.html>

<http://developer.android.com/guide/components/activities.html#SavingActivityState>

<http://stackoverflow.com/questions/151777/saving-activity-state-in-android>

<http://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)>

1. Service
2. 生命周期
3. 启动方式
4. 在activity中startService,如果该Service是第一次创建，那么所经历的生命周期是；onCreate() onStartCommand() onDestory()

如果该Service已经启动，再次启动时的生命周期是：

onStartCommand() onDestory()

1. bindService,绑定Service，如果是第一次绑定这个Service，那么其生命周期是：

onCreate() onBind() onUnbind onDestory

如果该Service已经启动，再次绑定时的生命周期是：

onBind() onUnbind onDestory

1. 在Manifest中注册Service

<manifest ... >  
  ...  
  <application ... >  
      <service android:name=".YourService" />  
      ...  
  </application>  
</manifest>

<http://developer.android.com/reference/android/app/IntentService.html>

<http://developer.android.com/guide/components/services.html>