### Android的四大组件

Activity, Service, Content Provider, Broadcast Receiver

### Android中常用的布局

Frame Layout, Linear Layout, Absolute Layout, Relative Layout, Table Layout, Constraint Layout

### Android Animation

Tween animation, Frame animation

### activity的启动模式有哪些？是什么含义？‍‍

Standard: Default. The system always creates a new instance of the activity in the target task and routes the intent to it.

SingleTop: If an instance of the activity already exists at the top of the target task, the system routes the intent to that instance through a call to its onNewIntent() method, rather than creating a new instance of the activity. 适合接收通知启动的内容显示页面

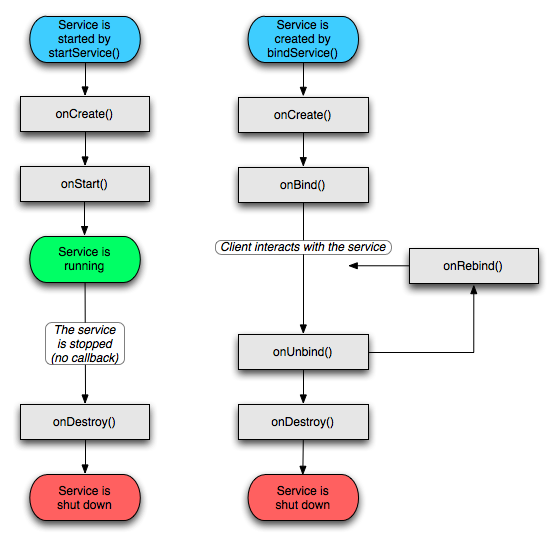
SingleTask: The system creates the activity at the root of a new task and routes the intent to it. However, if an instance of the activity already exists, the system routes the intent to existing instance through a call to its onNewIntent() method, rather than creating a new one. 适合作为程序入口点

SingleInstance: Same as "singleTask", except that the system doesn't launch any other activities into the task holding the instance. The activity is always the single and only member of its task. 闹铃的响铃界面

### android:taskAffinity

The task that the activity has an affinity for. Activities with the same affinity conceptually belong to the same task. The affinity of a task is determined by the affinity of its root activity.

### Service生命周期



Start cannot call service methods. Bind call methods through Binder

bindService(Intent, ServiceConnection, int)

### IntentService

Clients send requests through [startService(Intent)](https://developer.android.com/reference/android/content/Context.html#startService(android.content.Intent)) calls; the service is started as needed, handles each Intent in turn using a worker thread, and stops itself when it runs out of work.

### Activity和Service通信

通过Binder对象

通过[broadcast](http://dict.baidu.com/s?wd=broadcast" \t "_blank)(广播)的形式

### Activity的生命周期

### https://i.stack.imgur.com/fRxIQ.png

### activity在屏幕旋转时的生命周期

android:configChanges="keyboardHidden|orientation|screenSize" : onConfigurationChagned

Without above: Old activity is destroyed and a new one is created

### 介绍不同场景下Activity生命周期的变化过程

* + 启动Activity： onCreate()—>onStart()—>onResume()，Activity进入运行状态。
  + Activity退居后台： 当前Activity转到新的Activity界面或按Home键回到主屏： onPause()—>onStop()，进入停滞状态。
  + Activity返回前台： onRestart()—>onStart()—>onResume()，再次回到运行状态。
  + Activity退居后台，且系统内存不足， 系统会杀死这个后台状态的Activity，若再次回到这个Activity,则会走onCreate()–>onStart()—>onResume()
  + 锁定屏与解锁屏幕 只会调用onPause()，而不会调用onStop方法，开屏后则调用onResume()

### Fragment 状态的保存及恢复

onSaveInstanceState onActivityCreated

### Fragment setRetainInstance(true)

### 注册广播有几种方式

静态注册：AndroidManifest.xml配置文件 无论应用是否在运行，只要有对应广播发送，都会被接收到  
动态注册：Java代码 registerBroadcast unregister 广播跟随程序的生命周期

### 退出多Activity的application的方式

* + 利用ActivityContainer管理所有的Activity引用
  + 使用广播通知BaseActivity结束
  + 直接杀死进程
  + 采用SingleTask的特点，Kill Root activity, 结束应用

### Android Interface Define Language

### Handler消息机制

Looper, Message, Message Queue, Handler

### Handler memory leak

* + Java 中非静态内部类和匿名内部类会持有外部类的引用 Message keeps activity reference
  + Handler 的生命周期比外部类长。

### View的绘制流程

Measure, layout, draw

### Touch事件的传递机制

public Boolean dispatchTouchEvent(MotionEventev); //用来分派event

public boolean onInterceptTouchEvent(MotionEventev);//用来拦截event

public Boolean onTouchEvent(MotionEventev);//用来处理event

### Android 内存泄漏总结

* + 单例造成的内存泄漏
  + 非静态内部类创建静态实例造成的内存泄漏: Static references activity
  + Handler造成的内存泄漏
  + 线程造成的内存泄漏: Activity在销毁之前，任务还未完成
  + 资源未关闭造成的内存泄漏

### Android UI性能优化

* + 合理选择控件容器: LinearLayout易用，效率高，表达能力有限。RelativeLayout复杂，表达能力强，效率稍逊
  + 去掉window的默认背景
  + 去掉其他不必要的背景
  + ClipRect & QuickReject: 通过canvas.clipRect()来帮助系统识别那些可见的区域
  + ViewStub: 高效占位符
  + Merge: 干掉一个view层级
  + 善用draw9patch
  + 慎用Alpha
  + 避免“OverDesign”