返回博客列表

原 一些常见的Android面试基础题做下总结，看看你能做出多少道

gavin_jin

发布时间: 2012/02/27 18:27 阅读: 6498 收藏: 21 点赞: 0 评论: 1

1. Intent的几种有关Activity启动的方式有哪些，你了解每个含义吗?

这里Android123提示大家，Intent的一些标记有FLAG_ACTIVITY_BROUGHT_TO_FRONT 将activity带动最前面FLAG_ACTIVITY_CLEAR_TOP清除顶部FLAG_ACTIVITY_CLEAR_WHEN_TASK_RESET重要任务时清除

FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS排除最近的

FLAG_ACTIVITY_MULTIPLE_TASK多任务器的

FLAG_ACTIVITY_NEW_TASK 新任务启动。

每种含义大家看SDK文档和具体跑下这样你的记忆会更深刻些。

2. Activity和Task的启动模式有哪些? 每种含义是什么?

有关在AndroidManifest.xml中的android:launchMode定义，主要有standard、singleTop、singleTask和singleInstance，同时对于android:taskAffinity这些问题大家也要了解，Android开发网在以前的文章中讲过，不过很多开发者仍然不是很清楚，这些基础问题我们以后仍然会再次总结。

关于Activity的启动模式，参考： http://blog.csdn.net/feng88724/archive/2011/05/11/6412638.aspx

3. 通过Intent传递一些二进制数据的方法有哪些?

1). 使用Serializable接口实现序列化，这是Java常用的方法。

2). 实现Parcelable接口，这里Android的部分类比如Bitmap类就已经实现了，同时Parcelable在Android AIDL中交换数据也很常见的。

4. 能说下Android应用的入口点吗?

在Sun的Java体系中入口点和标准c语言一样是main()，而每个Android程序都包含着一个Application实例，一个Application实例中有多个Activity、Service、ContentProvider或Broadcast Receiver。因为大部分的应用都包含Activity所以，说很多网友认为是Activity的onCreate，但是你没有发现你的工程中有多个Activity吗? 你可能没有见过没有Activity的Android应用吧

其实在android.app.Application这个包的onCreate才是真正的Android入口点，只不过大多数开发者无需重写该类，他的继承关系如下图:

```
java.lang.Object
  ↳ android.content.Context
    ↳ android.content.ContextWrapper
      ↳ android.app.Application
```

android.app.Application类包含了4个公开的方法

```
void  onConfigurationChanged(Configuration newConfig)
void  onCreate()  //这里才是真正的入口点。
void  onLowMemory()
void  onTerminate()
```

所以希望大家，**记住真正的Android入口点是application的main**，你可以看下androidmanifest.xml的包含关系就清楚了，并不是每个应用都必须有Activity的

5. Android都有哪些XML解析器，都熟练掌握吗?

这里XmlPull、SAX和DOM相信做过Web开发的都已经滚瓜烂熟了。

6. SQLite支持事务吗? 添加删除如何提高性能?

SQLite作为轻量级的数据库，比MySQL还小，但支持SQL语句查询，提高性能可以考虑通过原始经过优化的SQL查询语句方式处理。

7. Android Service和Binder、AIDL你都熟练吗?

作为Android重要的后台服务，这些每个Android开发者都应该掌握，这也算是和Java SE最大的不同了，具体的实例大家可以查看Android音乐播放器的源代码Music.git中的，这里不再赘述。

http://my.oschina.net/u/226973/blog/42108

8. 你用过哪款Android手机，有哪些优点和不足，相对于iPhone或Symbian又有哪些优缺点?

把这个作为面试题也是考察下，可以大概了解到它对Android的了解程度，多移动开发的认识。

1. 请描述下Activity的生命周期。

创建 oncreate - 启动onstart – 恢复 onResume – 暂停 onPause – 结束 onEnd – 销毁onDestroy

2. 如果后台的Activity由于某原因被系统回收了，如何在被系统回收之前保存当前状态？

在”暂停 onPause” 状态将数据保存。

3. 如何将一个Activity设置成窗口的样式。

设置Theme。

4. 如何退出Activity？如何安全退出已调用多个Activity的Application？

5. 请介绍下Android中常用的五种布局。

线性布局LinearLayout， 相对布局 RelativeLayout

表单布局 TableLayout，
绝对布局AbsLayout（已淘汰）

帧布局FrameLayout

6. 请介绍下Android的数据存储方式。

sharedPreference,文件,数据库SQlite,网络存储

7. 请介绍下ContentProvider是如何实现数据共享的。

8. 如何启用Service，如何停用Service。

启动：
Context.startService()
and
Context.bindService().

关闭： Context.stopService().

Service.stopSelf()
or
Service.stopSelfResult()

9. 注册广播有几种方式，这些方式有何优缺点？请谈谈Android引入广播机制的用意。

Android广播机制（两种注册方法）

在android下，要想接受广播信息，那么这个广播接收器就得我们自己来实现了，我们可以继承BroadcastReceiver，就可以有一个广播接受器了。有个接受器还不够，我们还得重写BroadcastReceiver里面的onReceiver方法，当来广播的时候我们要干什么，这就要我们自己来实现，不过我们可以搞一个信息防火墙。具体的代码：

```
public class SmsBroadCastReceiver extends BroadcastReceiver

{

   @Override

   public void onReceive(Context context, Intent intent)

   {

      Bundle bundle = intent.getExtras();

      Object[] object = (Object[])bundle.get("pdus");

      SmsMessage sms[]=new SmsMessage[object.length];

      for(int i=0;i<object.length;i++)

      {

         sms[0] = SmsMessage.createFromPdu((byte[])object[i]);

         Toast.makeText(context, "来自"+sms[i].getDisplayOriginatingAddress()+" 的消息是："+sms[i].getDisplayMessageBody(), Toast.LENGTH_SHORT).show();

      }

      //终止广播，在这里我们可以稍微处理，根据用户输入的号码可以实现短信防火墙。

      abortBroadcast();

   }

}
```

当实现了广播接收器，还要设置广播接收器接收广播信息的类型，这里是信息：android.provider.Telephony.SMS_RECEIVED

我们就可以把广播接收器注册到系统里面，可以让系统知道我们有个广播接收器。这里有两种，一种是代码动态注册：

```
//生成广播处理

smsBroadCastReceiver = new SmsBroadCastReceiver();

//实例化过滤器并设置要过滤的广播

IntentFilter intentFilter = new IntentFilter("android.provider.Telephony.SMS_RECEIVED");

//注册广播

BroadCastReceiverActivity.this.registerReceiver(smsBroadCastReceiver, intentFilter);
```

一种是在AndroidManifest.xml中配置广播

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="spl.broadCastReceiver"

    android:versionCode="1"

    android:versionName="1.0">

   <application android:icon="@drawable/icon" android:label="@string/app_name">

      <activity android:name=".BroadCastReceiverActivity"

                android:label="@string/app_name">

         <intent-filter>

            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />

         </intent-filter>

      </activity>

      <!--广播注册-->

      <receiver android:name=".SmsBroadCastReceiver">

         <intent-filter android:priority="20">
```

```
            <intent-filter android:priority= 20 >
                <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
            </intent-filter>
        </receiver>
    </application>
    <uses-sdk android:minSdkVersion="7" />
    <!-- 权限申请 -->
    <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
</manifest>
```

两种注册类型的区别是：

1)第一种不是常驻型广播，也就是说广播跟随程序的生命周期。

2)第二种是常驻型，也就是说当应用程序关闭后，如果有信息广播来，程序也会被系统调用自动运行。

10. 请解释下在单线程模型中Message、Handler、Message Queue、Looper之间的关系。

1. Android进程

当一个程序第一次启动的时候，Android会启动一个LINUX进程和一个主线程。默 认的情况下，所有该程序的组件都将在该进程和线程中运行。

同 时，Android会为每个应用程序分配一个单独的LINUX用户。Android会尽量保留一个正在运行进程，只在内存资源出现不足时，Android 会尝试停止一些进程从而释放足够的资源给其他新的进程使用， 也能保证用户正在访问的当前进程有足够的资源去及时地响应用户的事件。Android会根据进程中运行的组件类别以及组件的状态来判断该进程的重要 性，Android会首先停止那些不重要的进程。

按照重要性从高到低一共有五个级别：

前台进程
前台进程是用户当前正在使用的进程。只有一些前台进程可以在任何时候都存在。他们是最后一个被结束的，当内存低到根本连他们都不能运行的时候。一般来说， 在这种情况下，设备会进行内存调度，中止一些前台进程来保持对用户交互的响应。
可见进程
可见进程不包含前台的组件但是会在屏幕上显示一个可见的进程是的重要程度很高，除非前台进程需要获取它的资源，不然不会被中止。
服务进程
运 行着一个通过startService() 方法启动的service，这个service不属于上面提到的2种更高重要性的。service所在的进程虽然对用户不是直接可见的，但是他们执行了用 户非常关注的任务（比如播放mp3，从网络下载数据）。只要前台进程和可见进程有足够的内存，系统不会回收他们。
后台进程
运 行着一个对用户不可见的activity（调用过 onStop() 方法).这些进程对用户体验没有直接的影响，可以在服务进程、可见进程、前台进 程需要内存的时候回收。通常，系统中会有很多不可见进程在运行，他们被保存在LRU (least recently used) 列表中，以便内存不足的时候被第一时间回收。如果一个activity正 确的执行了它的生命周期，关闭这个进程对于用户体验没有太大的影响。
空进程
未运行任何程序组件。运行这些进程的唯一原因是作为一个缓存，缩短下次程序需要重新使用的启动时间。系统经常中止这些进程，这样可以调节程序缓存和系统缓 存的平衡。
Android 对进程的重要性评级的时候，选取它最高的级别。另外，当被另外的一个进程依赖的时候，某个进程的级别可能会增高。一个为其他进程服务的进程永远不会比被服 务的进程重要级低。因为服务进程比后台activity进程重要级高，因此一个要进行耗时工作的activity最好启动一个service来做这个工 作，而不是开启一个子进程——特别是这个操作需要的时间比activity存在的时间还要长的时候。例如，在后台播放音乐，向网上上传摄像头拍到的图片， 使用service可以使进程最少获取到“服务进程”级别的重要级，而不用考虑activity目前是什么状态。broadcast receivers做费时的工作的时候，也应该启用一个服务而不是开一个线程。
2. 单线程模型

当一个程序第一次启动时，Android会同时启动一个对应的主线程（Main Thread），主线程主要负责处理与UI相关的事件，如用户的按键事件，用户接触屏幕的事件以及屏幕绘图事件，并把相关的事件分发到对应的组件进行处 理。所以主线程通常又被叫做UI线程。在开发Android应用时必须遵守单线程模型的原则： Android UI操作并不是线程安全的并且这些操作必须在UI线程中执行。
2.1 子线程更新UI
Android的UI是单线程(Single-threaded)的。为了避免拖住GUI，一些较费时的对象应该交给独立的线程去执行。如果幕后的线程来 执行UI对象，Android就会发出错误讯息
CalledFromWrongThreadException。以后遇到这样的异常抛出时就要知道怎么回 事了！
2.2 Message Queue

在单线程模型下，为了解决类似的问题，Android设计了一个Message Queue(消息队列)， 线程间可以通过该Message Queue并结合Handler和Looper组件进行信息交换。下面将对它们进行分别介绍：
1. Message

Message消息，理解为线程间交流的信息，处理数据后台线程需要更新UI，则发送Message内含一些数据给UI线程。
2. Handler

Handler处理者，是Message的主要处理者，负[illegible]essage的发送，Message内容的执行处理。后台线程就是通过传进来的 Handler对象引用来sendMessag[illegible]ssage)。而使用Handler，需要implement 该类的handleMessage(Message)
方法，它是处理这些Message的操作内容，例如Update UI。通常需要子类化Handler来实现handleMessage方法。
3. Message Queue

Message Queue消息队列，用来存放通过Handler发布的消息，按照先进先出执行。

每个message queue都会有一个对应的Handler。Handler会向message queue通过两种方法发送消息：sendMessage或post。这两种消息都会插在message queue队尾并按先进先出执行。但通过这两种方法发送的消息执行的方式略有不同：通过sendMessage发送的是一个message对象,会被 Handler的handleMessage()函数处理；而通过post方法发送的是一个runnable对象，则会自己执行。
4. Looper

Looper是每条线程里的Message Queue的管家。Android没有Global的Message Queue，而Android会自动替主线程(UI线程)建立Message Queue，但在子线程里并没有建立Message Queue。所以调用Looper.getMainLooper()得到的主线程的Looper不为NULL，但调用Looper.myLooper() 得到当前线程的Looper就有可能为NULL。

对于子线程使用Looper，API Doc提供了正确的使用方法：

这个Message机制的大概流程：

1. 在Looper.loop()方法运行开始后，循环地按照接收顺序取出Message Queue里面的非NULL的Message。

2. 一开始Message Queue里面的Message都是NULL的。当Handler.sendMessage(Message)到Message Queue，该函数里面设置了那个Message对象的target属性是当前的Handler对象。随后Looper取出了那个Message，则调用 该Message的target指向的Hander的dispatchMessage函数对Message进行处理。

在dispatchMessage方法里，如何处理Message则由用户指定，三个判断，优先级从高到低：

1) Message里面的Callback，一个实现了Runnable接口的对象，其中run函数做处理工作；

2) Handler里面的mCallback指向的一个实现了Callback接口的对象，由其handleMessage进行处理；

3) 处理消息Handler对象对应的类继承并实现了其中handleMessage函数，通过这个实现的handleMessage函数处理消息。

由此可见，我们实现的handleMessage方法是优先级最低的！

3. Handler处理完该Message (update UI) 后，Looper则设置该Message为NULL，以便回收！

在网上有很多文章讲述主线程和其他子线程如何交互，传送信息，最终谁来执行处理信息之类的，个人理解是最简单的方法——判断Handler对象里面的 Looper对象是属于哪条线程的，则由该线程来执行！

1. 当Handler对象的构造函数的参数为空，则为当前所在线程的Looper；
2. Looper.getMainLooper()得到的是主线程的Looper对象，Looper.myLooper()得到的是当前线程的Looper对象。

11. AIDL的全称是什么？如何工作？能处理哪些类型的数据？

12. 请解释下Android程序运行时权限与文件系统权限的区别。(Edited by Sodino)

http://my.oschina.net/u/226973/blog/42238

13. 系统上安装了多种浏览器，能否指定某浏览器访问指定页面？请说明原由。

14. 有一个一维整型数组int[]data保存的是一张宽为width，高为height的图片像素值信息。请写一个算法，将该图片所有的白色不透明(0xffffffff)像素点的透明度调整为50%。

暂时没有找到！

1、什么是ANR 如何避免它？

http://blog.csdn.net/Zengyangtech/archive/2010/11/21/6025671.aspx

ANR：Application Not Responding，五秒
在Android中，活动管理器和窗口管理器这两个系统服务负责监视应用程序的响应。当出现下列情况时，Android就会显示ANR对话框了：
对输入事件（如按键、触摸屏事件）的响应超过5秒
意向接受器（intentReceiver）超过10秒钟仍未执行完毕
Android应用程序完全运行在一个独立的线程中（例如main）。这就意味着，任何在主线程中运行的，需要消耗大量时间的操作都会引发ANR。因为此时，你的应用程序已经没有机会去响应输入事件和意向广播（Intent broadcast）。
因此，任何运行在主线程中的方法，都要尽可能的只做少量的工作。特别是活动生命周期中的重要方法如onCreate()和 onResume()等更应如此。潜在的比较耗时的操作，如访问网络和数据库；或者是开销很大的计算，比如改变位图的大小，需要在一个单独的子线程中完成（或者是使用异步请求，如数据库操作）。但这并不意味着你的主线程需要进入阻塞状态已等待子线程结束 -- 也不需要调用Therad.wait()或者Thread.sleep()方法。取而代之的是，主线程为子线程提供一个句柄（Handler），让子线程在即将结束的时候调用它（xing:可以参看Snake的例子，这种方法与以前我们所接触的有所不同）。使用这种方法涉及你的应用程序，能够保证你的程序对输入保持良好的响应，从而避免因为输入事件超过5秒钟不被处理而产生的ANR。这种实践需要应用到所有显示用户界面的线程，因为他们都面临着同样的超时问题

2、什么情况会导致Force Close ？如何避免？能否捕获导致其的异常？

一般像空指针啊，可以看起logcat，然后对应到程序中 来解决错误

3、Android本身的api并未声明会抛出异常，则其在运行时有无可能抛出runtime异常，你遇到过吗？诺有的话会导致什么问题？如何解决？

会有运行时异常， 运行时异常无需捕捉

4、简要解释一下activity、 intent 、intent filter、service、Broadcast、BroadcaseReceiver

一个activity呈现了一个用户可以操作的可视化用户界面
一个service不包含可见的用户界面，而是在后台无限地运行
可以连接到一个正在运行的服务中，连接后，可以通过服务中暴露出来的借口与其进行通信
一个broadcast receiver是一个接收广播消息并作出回应的component，broadcast receiver没有界面
intent:content provider在接收到ContentResolver的请求时被激活。
activity, service和broadcast receiver是被称为intents的异步消息激活的。
一个intent是一个Intent对象，它保存了消息的内容。对于activity和service来说，它指定了请求的操作名称和待操作数据的URI
Intent对象可以显式的指定一个目标component。如果这样的话，android会找到这个component（基于manifest文件中的声明）并激活它。但如果一个目标不是显式指定的，android必须找到响应intent的最佳component。
它是通过将Intent对象和目标的intent filter相比较来完成这一工作的。一个component的intent filter告诉android该component能处理的intent。intent filter也是在manifest文件中声明的。

5、IntentService有何

其实它也是避免ANR的方法：
IntentService 的好处

* Acitivity的进程，当处理Intent的时候，会产生一个对应的Service
* Android的进程处理器现在会尽可能的不kill掉你
* 非常容易使用

1. android:paddingLeft与android:layout_marginLeft的区别

当按钮分别设置以上两个属性时，得到的效果是不一样的。
android:paddingLeft="30px"
按钮上设置的内容（例如图片）离按钮左边边界30个像素
android:layout_marginLeft="30px"
整个按钮离左边设置的内容30个像素

2. Android 动画有哪几种？描述一下

两种。 Tween动画和Frame动画。 Tween动画主要是透明度、尺寸伸缩、旋转、位移等效果。

Frame动画可以理解成gif， 一帧一帧的显示图片。比较常用的有滚动条效果。

3. 对Intent、Activity、广播、Service等的理解。

4. 哪些情况下会发生ANR？怎么对应 ？ (这个问题也重复了)

Application Not Response。

5. 隐式、显式Intent的区别

· 显式意图

通过名字指明目标组件（这个组件名字字段component name field,
前面提到过,
有一个数值集）。既然组件名称通常不为其他应用程序的开发者所了解，显式意图典型的被用作应用程序的内部消息-例如一个活动启动一个附属服务或姊妹活动。
隐式意图

不命名目标组件（组件名称字段为空）。隐式意图经常用来激活其他应用程序的组件。

这一块参考Intent说明，可以同时看一下Notepad的例子， 帮助理解，个人感觉很重要！

(上面几个问题感觉都不错，各位自己查漏补缺吧！ 有些还没有写答案的， 各位有空可以回答一下~ 我会将你的答案填充上去)

返回键与Home键区别？

back键默认行为是finish处于前台的Activity的即Activity的状态为Destroy状态为止，再次启动该Activity是从onCreate开始的(不会调用onSaveInstanceState方法)。Home键默认是stop前台的Activity即状态为onStop为止而不是Destroy,若再次启动它，会调用onSaveInstanceState方法，保持上次Activity的状态则是从OnRestart开

而不是Destroy,若再次启动它，会调用onSaveInstanceState方法，保持上次Activity的状态则是从OnRestart开始的---->onStart()--->onResume()。

思科面试题
1. Android中如何传递二进制数据？
2. ANR是什么？哪些情况下会发生？开发时如何寻找ANR？
3. 解释AIDL
4. SAX、DOM、PULL解析xml的原理，以及各自优缺点
4. DIP、DPI分别是什么？

5.java线程的sleep(),wait(),notify(),yield()方法的区别？

1.sleep()使线程休眠一段时间，一段时间结束后，线程进入可执行状态，但并不是立即执行，只是在被排程器调用的时候才执行。在休眠期间，并不释放所持有的“锁”；
2.wait()使线程休眠一段时间，若设置参数，时间到时，线程就自动进入可执行状态。若没有，则需要notify()方法去调用。注意：wait()方法和notify()方法都时针对this对象的，调用wait()方法后，会释放加在对象上的“锁”。
3.yield()使线程放弃执行的权利，进入可执行状态，也就意味着线程在yield()方法后，有可能又执行。使用yield()方法，线程并不释放自己锁持有的“锁”。
已有

分类：android　标签：android面试题　字数：5704

gavin_jin　关注

粉丝: 22　博客数: 94　共码了 163523 字

评论（1）

gavin_jin
1楼 2012/03/31 18:16
强烈反对对广告

插入：　表情　开源软件　　发表评论