



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

第三版：Android 60 道

Android 系统的架构

1、Android 系统架构之应用程序

Android 会同一系列核心应用程序包一起发布，该应用程序包包括 email 客户端，SMS 短消息程序，日历，地图，浏览器，联系人管理程序等。所有的应用程序都是使用 JAVA 语言编写的。

2、Android 系统架构之应用程序框架

开发人员可以完全访问核心应用程序所使用的 API 框架（android.jar）。该应用程序的架构设计简化了组件的重用；任何一个应用程序都可以发布它的功能块并且任何其它的应用程序都可以使用其所发布的功能块。

3、Android 系统架构之系统运行库

Android 包含一些 C/C++ 库，这些库能被 Android 系统中不同的组件使用。它们通过 Android 应用程序框架为开发者提供服务。

4、Android 系统架构之 Linux 内核

Android 的核心系统服务依赖于 Linux 2.6 内核，如安全性，内存管理，进程管理，网络协议栈和驱动模型。Linux 内核也同时作为硬件和软件栈之间的抽象层。

activity 的生命周期

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

![92_1.png][92_1.png]

Activity 生命周期方法主要有 onCreate()、onStart()、onResume()、onPause()、onStop()、onDestroy()和 onRestart()等 7 个方法。

- 1、启动一个 A Activity，分别执行 onCreate()、onStart()、onResume()方法。
- 2、从 A Activity 打开 B Activity 分别执行 A onPause()、B onCreate()、B onStart()、B onResume()、A onStop()方法。
- 3、关闭 B Activity，分别执行 B onPause()、A onRestart()、A onStart()、A onResume()、B onStop()、B onDestroy()方法。
- 4、横竖屏切换 A Activity，清单文件中不设置 android:configChanges 属性时，先销毁 onPause()、onStop()、onDestroy()再重新创建 onCreate()、onStart()、onResume()方法，设置 orientation|screenSize（一定要同时出现）属性值时，不走生命周期方法，只会执行 onConfigurationChanged()方法。
- 5、Activity 之间的切换可以看出 onPause()、onStop()这两个方法比较特殊，切换的时候 onPause()方法不要加入太多耗时操作否则会影响体验。

Fragment 的生命周期

Fragment 的生命周期

![92_2.png][92_2.png]

Fragment 与 Activity 生命周期对比

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

![92_3.png][92_3.png]

Fragment 的生命周期方法主要有 `onAttach()`、`onCreate()`、`onCreateView()`、`onActivityCreated()`、`onstart()`、`onResume()`、`onPause()`、`onStop()`、`onDestroyView()`、`onDestroy()`、`onDetach()`等 11 个方法。

- 1、 切换到该 Fragment，分别执行 `onAttach()`、`onCreate()`、`onCreateView()`、`onActivityCreated()`、`onstart()`、`onResume()`方法。
- 2、 锁屏，分别执行 `onPause()`、`onStop()`方法。
- 3、 亮屏，分别执行 `onstart()`、`onResume()`方法。
- 4、 覆盖，切换到其他 Fragment，分别执行 `onPause()`、`onStop()`、`onDestroyView()`方法。
- 5、 从其他 Fragment 回到之前 Fragment，分别执行 `onCreateView()`、`onActivityCreated()`、`onstart()`、`onResume()`方法。

Service 生命周期

在 Service 的生命周期里，常用的有：

4 个手动调用的方法

| | |
|------------------------------|------|
| <code>startService()</code> | 启动服务 |
| <code>stopService()</code> | 关闭服务 |
| <code>bindService()</code> | 绑定服务 |
| <code>unbindService()</code> | 解绑服务 |

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

5 个内部自动调用的方法

| | |
|------------------|------|
| onCreat() | 创建服务 |
| onStartCommand() | 开始服务 |
| onDestroy() | 销毁服务 |
| onBind() | 绑定服务 |
| onUnbind() | 解绑服务 |

- 1、 手动调用 startService()启动服务，自动调用内部方法：onCreate()、onStartCommand()，如果一个 Service 被 startService()多次启动，那么 onCreate()也只会调用一次。
- 2、 手动调用 stopService()关闭服务，自动调用内部方法：onDestory()，如果一个 Service 被启动且被绑定，如果在没有解绑的前提下使用 stopService()关闭服务是无法停止服务的。
- 3、 手动调用 bindService()后，自动调用内部方法：onCreate()、onBind()。
- 4、 手动调用 unbindService()后，自动调用内部方法：onUnbind()、onDestory()。
- 5、 startService()和 stopService()只能开启和关闭 Service，无法操作 Service，调用者退出后 Service 仍然存在；bindService()和 unbindService()可以操作 Service，调用者退出后，Service 随着调用者销毁。

Android 中，帧动画

帧动画是最容易实现的一种动画，这种动画更多的依赖于完善的 UI 资源，他的原理就是将一张张单独的图片连贯的进行播放，从而在视觉上产生一种动画的效果；有点类似于某些软件制作 gif 动画的方式。在有些代码中，我们还会看到 android:

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

oneshot="false"，这个 oneshot 的含义就是动画执行一次 (true) 还是循环执行多次。

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list
xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:drawable="@drawable/a_0"
    android:duration="100" />
  <item
    android:drawable="@drawable/a_1"
    android:duration="100" />
  <item
    android:drawable="@drawable/a_2"
    android:duration="100" />
</animation-list>
```

补间动画

补间动画又可以分为四种形式，分别是 alpha (淡入淡出)，translate (位移)，scale (缩放大小)，rotate (旋转)。

补间动画的实现，一般会采用 xml 文件的形式；代码会更容易书写和阅读，同时也更容易复用。Interpolator 主要作用是可以控制动画的变化速率，就是动画进行的快慢节奏。pivot 决定了当前动画执行的参考位置

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@[package:]anim/interpolator_resource"
    android:shareInterpolator=["true" | "false"] >
  <alpha
```

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

```
        android:fromAlpha="float"
        android:toAlpha="float" />
    <scale
        android:fromXScale="float"
        android:toXScale="float"
        android:fromYScale="float"
        android:toYScale="float"
        android:pivotX="float"
        android:pivotY="float" />
    <translate
        android:fromXDelta="float"
        android:toXDelta="float"
        android:fromYDelta="float"
        android:toYDelta="float" />
    <rotate
        android:fromDegrees="float"
        android:toDegrees="float"
        android:pivotX="float"
        android:pivotY="float" />
    <set>
        ...
    </set>
</set>
```

属性动画

属性动画，顾名思义它是对于对象属性的动画。因此，所有补间动画的内容，都可以通过属性动画实现。属性动画的运行机制是通过不断地对值进行操作来实现的，而初始值和结束值之间的动画过渡就是由 ValueAnimator 这个类来负责计算的。它的内部使用一种时间循环的机制来计算值与值之间的动画过渡，我们只需

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

要将初始值和结束值提供给 ValueAnimator，并且告诉它动画所需运行的时长，那么 ValueAnimator 就会自动帮我们完成从初始值平滑地过渡到结束值这样的效果。除此之外，ValueAnimator 还负责管理动画的播放次数、播放模式、以及对动画设置监听器等。

Android 中 4 大组件

- 1、Activity: Activity 是 Android 程序与用户交互的窗口，是 Android 构造块中最基本的一种，它需要为保持各界面的状态，做很多持久化的事情，妥善管理生命周期以及一些跳转逻辑。
- 2、Broadcast Receiver: 接受一种或者多种 Intent 作触发事件，接受相关消息，做一些简单处理，转换成一条 Notification，统一了 Android 的事件广播模型。
- 3、Content Provider: 是 Android 提供的第三方应用数据的访问方案，可以派生 Content Provider 类，对外提供数据，可以像数据库一样进行选择排序，屏蔽内部数据的存储细节，向外提供统一的接口模型，大大简化上层应用，对数据的整合提供了更方便的途径。
- 4、service: 后台服务于 Activity，封装有一个完整的功能逻辑实现，接受上层指令，完成相关的事务，定义好需要接受的 Intent 提供同步和异步的接口。

Android 中常用布局

常用的布局：

FrameLayout(帧布局):所有东西依次都放在左上角，会重叠

LinearLayout(线性布局):按照水平和垂直进行数据展示

RelativeLayout(相对布局):以某一个元素为参照物，来定位的布局方式

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

不常用的布局：

TableLayout(表格布局): 每一个 TableLayout 里面有表格行 TableRow, TableRow 里面可以具体定义每一个元素 (Android TV 上使用)

AbsoluteLayout(绝对布局):用 X,Y 坐标来指定元素的位置, 元素多就不适用。
(机顶盒上使用)

新增布局：

PercentRelativeLayout (百分比相对布局) 可以通过百分比控制控件的大小。
PercentFrameLayout (百分比帧布局) 可以通过百分比控制控件的大小。

消息推送的方式

1、 使用极光和友盟推送。

2、 使用 XMPP 协议 (Openfire + Spark + Smack)

简介：基于 XML 协议的通讯协议，前身是 Jabber，目前已由 IETF 国际标准化组织完成了标准化工作。

优点：协议成熟、强大、可扩展性强、目前主要应用于许多聊天系统中，且已有开源的 Java 版的开发实例 androidpn。

缺点：协议较复杂、冗余（基于 XML）、费流量、费电，部署硬件成本高。

3、 使用 MQTT 协议 (更多信息见: [mqtt.org/]) [mqtt.org]

简介：轻量级的、基于代理的“发布/订阅”模式的消息传输协议。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

优点：协议简洁、小巧、可扩展性强、省流量、省电，目前已经应用到企业领域（参考：[\[mqtt.org/software\]](http://mqtt.org/software)），且...[\[mqtt.org_software\]](http://mqtt.org_software)

缺点：不够成熟、实现较复杂、服务端组件 rsmb 不开源，部署硬件成本较高。

4、使用 HTTP 轮循方式

简介：定时向 HTTP 服务端接口（Web Service API）获取最新消息。

优点：实现简单、可控性强，部署硬件成本低。

缺点：实时性差。

android 的数据存储

1、使用 SharedPreferences 存储数据；它是 Android 提供的用来存储一些简单配置信息的一种机制，采用了 XML 格式将数据存储到设备中。只能在同一个包内使用，不能在不同的包之间使用。

2、文件存储数据；文件存储方式是一种较常用的方法，在 Android 中读取/写入文件的方法，与 Java 中实现 I/O 的程序是完全一样的，提供了 `openFileInput()` 和 `openFileOutput()` 方法来读取设备上的文件。

3、SQLite 数据库存储数据；SQLite 是 Android 所带的一个标准的数据库，它支持 SQL 语句，它是一个轻量级的嵌入式数据库。

4、使用 ContentProvider 存储数据；主要用于应用程序之间进行数据交换，从而能够让其他的应用保存或读取此 Content Provider 的各种数据类型。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

5、网络存储数据；通过网络上提供给我们的存储空间来上传(存储)和下载(获取)我们存储在网络空间中的数据信息。

Activity 启动模式

介绍 Android 启动模式之前，先介绍两个概念 task 和 taskAffinity

task:

翻译过来就是“任务”，是一组相互有关联的 activity 集合，可以理解为 Activity 是在 task 里面活动的。task 存在于一个称为 back stack 的数据结构中，也就是说，task 是以栈的形式去管理 activity 的，所以也可以称为“任务栈”。

taskAffinity:

官方文档解释是：“The task that the activity has an affinity for.”，可以翻译为 activity 相关或者亲和的任务，这个参数标识了一个 Activity 所需要的任务栈的名字。默认情况下，所有 Activity 所需的任务栈的名字为应用的包名。taskAffinity 属性主要和 singleTask 启动模式或者 allowTaskReparenting 属性配对使用。

4 种启动模式

1、standard:

标准模式，也是系统默认的启动模式。假如 activity A 启动了 activity B，activity B 则会运行在 activity A 所在的任务栈中。而且每次启动一个 Activity，都会重新创建新的实例，不管这个实例在任务中是否已经存在。非 Activity 类型的 context（如 ApplicationContext）启动 standard 模式的 Activity 时会报错。非 Activity 类型的 context 并没有所谓的任务栈，由于上面第 1 点的原因所以系统会报错。此解决办法就是为待启动 Activity 指定

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

FLAG_ACTIVITY_NEW_TASK 标记位，这样启动的时候系统就会为它创建一个新的任务栈。这个时候待启动 Activity 其实是以 singleTask 模式启动的。

2、singleTop:

栈顶复用模式。假如 activity A 启动了 activity B，就会判断 A 所在的任务栈栈顶是否是 B 的实例。如果是，则不创建新的 activity B 实例而是直接引用这个栈顶实例，同时 onNewIntent 方法会被回调，通过该方法的参数可以取得当前请求的信息；如果不是，则创建新的 activity B 实例。

3、singleTask:

栈内复用模式。在第一次启动这个 Activity 时，系统便会创建一个新的任务，并且初始化 Activity 的实例，放在新任务的底部。不过需要满足一定条件的。那就是需要设置 taskAffinity 属性。前面也说过了，taskAffinity 属性是和 singleTask 模式搭配使用的。

![92_4.png][92_4.png]

1、singleInstance：单实例模式。这个是 singleTask 模式的加强版，它除了具有 singleTask 模式的所有特性外，它还有一点独特的特性，那就是此模式的 Activity 只能单独地位于一个任务栈，不与其他 Activity 共存于同一个任务栈。

广播注册

首先写一个类要继承 BroadcastReceiver

第一种：在清单文件中声明，添加

```
<receive android:name=".BroadcastReceiverDemo">
```

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

```
<intent-filter>
<action android:name="android.provider.Telephony.SMS_RECEIVED">
</intent-filter>
</receiver>
```

第二种：使用代码进行注册如：

```
IntentFilter filter = new
IntentFilter("android.provider.Telephony.SMS_RECEIVED");
BroadcastReceiverDemo receiver = new BroadcastReceiver();
registerReceiver(receiver, filter);
```

两种注册类型的区别是：

- 第一种是常驻型广播，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。
- 第二种不是常驻广播，也就是说广播跟随程序的生命周期。

Android 中的 ANR

ANR 的全称 application not responding 应用程序未响应。

在 android 中 Activity 的最长执行时间是 5 秒。

BroadcastReceiver 的最长执行时间则是 10 秒。

Service 的最长执行时间则是 20 秒。

超出执行时间就会产生 ANR。注意：ANR 是系统抛出的异常，程序是捕捉不了这个异常的。

解决方法：

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

1、运行在主线程里的任何方法都尽可能少做事情。特别是，Activity 应该在它的关键生命周期方法（如 onCreate()和 onResume()）里尽可能少的去做创建操作。（可以采用重新开启子线程的方式，然后使用 Handler+Message 的方式做一些操作，比如更新主线程中的 ui 等）

2、应用程序应该避免在 BroadcastReceiver 里做耗时的操作或计算。但不再是在子线程里做这些任务（因为 BroadcastReceiver 的生命周期短），替代的是，如果响应 Intent 广播需要执行一个耗时的动作的话，应用程序应该启动一个 Service。

ListView 优化

1、convertView 重用,利用好 convertView 来重用 View,切忌每次 getView() 都新建。ListView 的核心原理就是重用 View, 如果重用 view 不改变宽高, 重用 View 可以减少重新分配缓存造成的内存频繁分配/回收;

2、ViewHolder 优化,使用 ViewHolder 的原因是 findViewById 方法耗时较大, 如果控件个数过多, 会严重影响性能, 而使用 ViewHolder 主要是为了可以省去这个时间。通过 setTag, getTag 直接获取 View。

3、减少 Item-View 的布局层级, 这是所有 layout 都必须遵循的, 布局层级过深会直接导致 View 的测量与绘制浪费大量的时间。

4、adapter 中的 getView 方法尽量少使用逻辑

5、图片加载采用三级缓存, 避免每次都要重新加载。

6、尝试开启硬件加速来使 ListView 的滑动更加流畅。

7、使用 RecyclerView 代替。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

Android 数字签名

- 1、 所有的应用程序都必须有数字证书，Android 系统不会安装一个没有数字证书的应用程序
- 2、 Android 程序包使用的数字证书可以是自签名的，不需要一个权威的数字证书机构签名认证
- 3、 如果要正式发布一个 Android ，必须使用一个合适的私钥生成的数字证书来给程序签名。
- 4、 数字证书都是有有效期的，Android 只是在应用程序安装的时候才会检查证书的有效期。如果程序已经安装在系统中，即使证书过期也不会影响程序的正常功能。

Android root 机制

root 指的是你有权限可以再系统上对所有档案有“读”“写”“执行”的权力。root 机器不是真正能让你的应用程序具有 root 权限。它原理就跟 linux 下的像 sudo 这样的命令。在系统的 bin 目录下放个 su 程序并属主是 root 并有 suid 权限。则通过 su 执行的命令都具有 Android root 权限。当然使用临时用户权限想把 su 拷贝到/system/bin 目录并改属性并不是一件容易的事情。这里用到 2 个工具跟 2 个命令。把 busybox 拷贝到你有权访问的目录然后给他赋予 4755 权限，你就可以用它做很多事了。

View

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

显示视图，内置画布，提供图形绘制函数、触屏事件、按键事件函数等，必须在 UI 主线程内更新画面，速度较慢

SurfaceView

基于 view 视图进行拓展的视图类，更适合 2D 游戏的开发，是 view 的子类，类似使用双缓冲机制，在新的线程中更新画面所以刷新界面速度比 view 快

GLSurfaceView

基于 SurfaceView 视图再次进行拓展的视图类，专用于 3D 游戏开发的视图，是 surfaceView 的子类，openGL 专用

AsyncTask

AsyncTask 的三个泛型参数说明

- 1、 第一个参数：传入 doInBackground() 方法的参数类型
- 2、 第二个参数：传入 onProgressUpdate() 方法的参数类型
- 3、 第三个参数：传入 onPostExecute() 方法的参数类型，也是 doInBackground() 方法返回的类型

运行在主线程的方法：

```
onPostExecute ()  
onPreExecute()
```

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

onProgressUpdate(Progress...)

运行在子线程的方法：

`doInBackground ()`

控制 AsyncTask 停止的方法：

`cancel(boolean mayInterruptIfRunning)`

AsyncTask 的执行分为四个步骤

1、 继承 AsyncTask。

2、 实现 AsyncTask 中定义的下面一个或几个方法 `onPreExecute()`、`doInBackground(Params...)`、`onProgressUpdate(Progress...)`、`onPostExecute(Result)`。

3、 调用 `execute` 方法必须在 UI thread 中调用。

4、 该 task 只能被执行一次，否则多次调用时将会出现异常，取消任务可调用 `cancel`。

Android i18n

i18n 叫做国际化。android 对 i18n 和 L10n 提供了非常好的支持。软件在 `res/values` 以及其他带有语言修饰符的文件夹。如：`values-zh` 这些文件夹中提供语言，样式，尺寸 xml 资源。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

NDK

- 1、 NDK 是一系列工具集合，NDK 提供了一系列的工具，帮助开发者迅速的开发 C/C++ 的动态库，并能自动将 so 和 Java 应用打成 apk 包。
- 2、 NDK 集成了交叉编译器，并提供了相应的 mk 文件和隔离 cpu、平台等的差异，开发人员只需要简单的修改 mk 文件就可以创建出 so 文件。

启动一个程序，可以主界面点击图标进入，也可以从一个程序中跳转过去，二者有什么区别？

通过主界面进入，就是设置默认启动的 activity。在 manifest.xml 文件的 activity 标签中，写以下代码

```
<intent-filter>
<intent android:name="android.intent.action.MAIN" >
<intent android:name="android.intent.category.LAUNCHER" >
</intent-filter>
```

从另一个组件跳转到目标 activity，需要通过 intent 进行跳转。具体

```
Intent intent=new Intent(this,activity.class),startActivity(intent)
```

内存溢出和内存泄漏有什么区别？何时会产生内存泄漏？

内存溢出：当程序运行时所需的内存大于程序允许的最高内存，这时会出现内存溢出；

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

内存泄漏：在一些比较消耗资源的操作中，如果操作中内存一直未被释放，就会出现内存泄漏。比如未关闭 io,cursor。

sim 卡的 EF 文件有何作用

sim 卡就是电话卡，sim 卡内有自己的操作系统，用来与手机通讯的。Ef 文件用来存储数据的。

Activity 的状态有几种？

运行

暂停

停止

让 Activity 变成一个窗口

设置 activity 的 style 属性=" @android:style/Theme.Dialog"

android:gravity 与 android:layout_gravity 的区别

gravity：表示组件内元素的对齐方式

layout_gravity：相对于父类容器，该视图组件的对齐方式

如何退出 Activity

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

结束当前 activity

```
Finish()
killProgress()
System.exit(0)
```

关闭应用程序时，结束所有的 activity

可以创建一个 List 集合，每新建一个 activity，将该 activity 的实例放进 list 中，程序结束时，从集合中取出循环取出 activity 实例，调用 finish() 方法结束。

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 onPause 方法中调用 onSaveInstanceState()

Android 中的长度单位详解

Px: 像素

Sp 与 dp 也是长度单位，但是与屏幕的单位密度无关。

activity, service, intent 之间的关系

这三个都是 android 应用频率非常的组件。Activity 与 service 是四大核心组件。Activity 用来加载布局，显示窗口界面，service 运行后台，没有界面显示，intent 是 activity 与 service 的通信使者。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

activity 之间传递参数，除了 intent，广播接收器，contentProvider 之外，还有那些方法？

File: 文件存储，推荐使用 sharedPreferecnces 静态变量。

Adapter 是什么？你所接触过的 adapter 有那些？

是适配器，用来为列表提供数据适配的。经常使用的 adapter 有 baseadapter, arrayAdapter, SimpleAdapter, cursorAdapter, SpinnerAdapter 等

Fragment 与 activity 如何传值和交互？

Fragment 对象有一个 getActivity 的方法，通过该方法与 activity 交互使用 fragmentManager.findFragmentByXX 可以获取 fragment 对象，在 activity 中直接操作 fragment 对象

如果 Listview 中的数据源发生改变，如何更新 listview 中的数据

使用 adapter 的 notifyDataSetChanged 方法

广播接受者的生命周期？

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

广播接收者的生命周期非常短。当执行 onReceive 方法之后，广播就会销毁。在广播接收者不能进行耗时较长的操作。

在广播接收者不要创建子线程。广播接收者完成操作后，所在进程会变成空进程，很容易被系统回收。

ContentProvider 与 sqlite 有什么不一样的？

ContentProvider 会对外隐藏内部实现，只需要关注访问 contentProvider 的 uri 即可，contentProvider 应用在不同应用间共享。

Sqlite 操作本应用程序的数据库。

ContentProvider 可以对本地文件进行增删改查操作。

如何保存 activity 的状态？

默认情况下 activity 的状态系统会自动保存，有些时候需要我们手动调用保存。

当 activity 处于 onPause, onStop 之后，activity 处于未活动状态，但是 activity 对象却仍然存在。当内存不足，onPause, onStop 之后的 activity 可能会被系统摧毁。

当通过返回退出 activity 时，activity 状态并不会保存。

保存 activity 状态需要重写 onSaveInstanceState() 方法，在执行 onPause, onStop 之前调用 onSaveInstanceState 方法，onSaveInstanceState 需要一个 Bundle 类型的参数，我们可以将数据保存到 bundle 中，通过实参传递给 onSaveInstanceState 方法。

Activity 被销毁后，重新启动时，在 onCreate 方法中，接受保存的 bundle 参数，并将之前的数据取出。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

Android 中 activity, context, application 有什么不同。

Content 与 application 都继承与 contextWrapper, contextWrapper 继承于 Context 类。

Context: 表示当前上下文对象, 保存的是上下文中的参数和变量, 它可以让更加方便访问到一些资源。

Context 通常与 activity 的生命周期是一样的, application 表示整个应用程序的对象。

对于一些生命周期较长的, 不要使用 context, 可以使用 application。

在 activity 中, 尽量使用静态内部类, 不要使用内部类。内部里作为外部类的成员存在, 不是独立于 activity, 如果内存中还有内存继续引用到 context, activity 如果被销毁, context 还不会结束。

Service 是否在 main thread 中执行, service 里面是否能执行耗时的操作?

默认情况 service 在 main thread 中执行, 当 service 在主线程中运行, 那在 service 中不要进行一些比较耗时的操作, 比如说网络连接, 文件拷贝等。

Service 和 Activity 在同一个线程吗

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

默认情况下 service 与 activity 在同一个线程，都在 main Thread，或者 ui 线程中。

如果在清单文件中指定 service 的 process 属性，那么 service 就在另一个进程中运行。

Service 里面可以弹吐司么

可以。

在 service 的生命周期方法 onStartCommand() 可不可以执行网络操作？如何在 service 中执行网络操作？

可以的，就在 onStartCommand 方法内执行。

说说 ContentProvider、ContentResolver、

ContentObserver 之间的关系

ContentProvider：内容提供者，对外提供数据的操作，

contentProvider.notifyChanged(uir)：可以更新数据

contentResolver：内容解析者，解析 ContentProvider 返回的数据

ContentObServer: 内容监听者，监听数据的改变，

contentResolver.registerContentObServer()

请介绍下 ContentProvider 是如何实现数据共享的

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

ContentProvider 是一个对外提供数据的接口，首先需要实现 ContentProvider 这个接口，然后重写 query, insert, getType, delete, update 方法，最后在清单文件定义 contentProvider 的访问 uri

Intent 传递数据时，可以传递哪些类型数据？

基本数据类型以及对应的数组类型

可以传递 bundle 类型，但是 bundle 类型的数据需要实现 Serializable 或者 parcelable 接口

Serializable 和 Parcelable 的区别？

如果存储在内存中，推荐使用 parcelable，使用 serializable 在序列化的时候会产生大量的临时变量，会引起频繁的 GC

如果存储在硬盘上，推荐使用 Serializable，虽然 serializable 效率较低

Serializable 的实现：只需要实现 Serializable 接口，就会自动生成一个序列化 id

Parcelable 的实现：需要实现 Parcelable 接口，还需要 Parcelable.CREATOR 变量

请描述一下 Intent 和 IntentFilter

Intent 是组件的通讯使者，可以在组件间传递消息和数据。

IntentFilter 是 intent 的筛选器，可以对 intent 的 action, data, category, uri 这些属性进行筛选，确定符合的目标组件。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

什么是 IntentService? 有何优点?

IntentService 是 Service 的子类, 比普通的 Service 增加了额外的功能。先看 Service 本身存在两个问题:

Service 不会专门启动一条单独的进程, Service 与它所在应用位于同一个进程中;

Service 也不是专门一条新线程, 因此不应该在 Service 中直接处理耗时的任务;

特征

会创建独立的 worker 线程来处理所有的 Intent 请求;

会创建独立的 worker 线程来处理 onHandleIntent()方法实现的代码, 无需处理多线程问题;

所有请求处理完成后, IntentService 会自动停止, 无需调用 stopSelf()方法停止 Service;

为 Service 的 onBind()提供默认实现, 返回 null;

为 Service 的 onStartCommand 提供默认实现, 将请求 Intent 添加到队列中

使用

让 service 类继承 IntentService, 重写 onStartCommand 和 onHandleIntent 实现

Android 引入广播机制的用意

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

从 MVC 的角度考虑(应用程序内) 其实回答这个问题的时候还可以这样问, android 为什么要有那 4 大组件, 现在的移动开发模型基本上也是照搬的 web 那一套 MVC 架构, 只不过稍微做了修改. android 的四大组件本质上就是为了实现移动或者说嵌入式设备上的 MVC 架构, 它们之间有时候是一种相互依存的关系, 有时候又是一种补充关系, 引入广播机制可以方便几大组件的信息和数据交互.

程序间互通消息(例如在自己的应用程序内监听系统来电)

效率上(参考 UDP 的广播协议在局域网的方便性)

设计模式上(反转控制的一种应用, 类似监听者模式)

ListView 如何提高其效率?

当 convertView 为空时, 用 setTag()方法为每个 View 绑定一个存放控件的 ViewHolder 对象. 当 convertView 不为空, 重复利用已经创建的 view 的时候, 使用 getTag()方法获取绑定的 ViewHolder 对象, 这样就避免了 findViewById 对控件的层层查询, 而是快速定位到控件. 复用 convertView, 使用历史的 view, 提升效率 200%

自定义静态类 ViewHolder, 减少 findViewById 的次数. 提升效率 50%

异步加载数据, 分页加载数据.

使用 WeakReference 引用 ImageView 对象

ListView 如何实现分页加载

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

设置 ListView 的滚动监听器: `setOnScrollListener(new OnScrollListener{...})`
在监听器中有两个方法： 滚动状态发生变化的方法(`onScrollStateChanged`)和
listView 被滚动时调用的方法(`onScroll`)

在滚动状态发生改变的方法中，有三种状态：手指按下移动的状态：

`SCROLL_STATE_TOUCH_SCROLL`: 触摸滑动，惯性滚动（滑翔（flgin）状态）：

`SCROLL_STATE_FLING`: 滑翔，静止状态：`SCROLL_STATE_IDLE` // 静止，对不同的状态进行处理：

分批加载数据，只关心静止状态：关心最后一个可见的条目，如果最后一个可见条目就是数据适配器（集合）里的最后一个，此时可加载更多的数据。在每次加载的时候，计算出滚动的数量，当滚动的数量大于等于总数量的时候，可以提示用户无更多数据了。

ListView 可以显示多种类型的条目吗

这个当然可以的，ListView 显示的每个条目都是通过 baseAdapter 的 `getView(int position, View convertView, ViewGroup parent)` 来展示的，理论上我们完全可以让每个条目都是不同类型的 view。

比如：从服务器拿回一个标识为 `id=1`，那么当 `id=1` 的时候，我们就加载类型一的条目，当 `id=2` 的时候，加载类型二的条目。常见布局在资讯类客户端中可以经常看到。

除此之外 adapter 还提供了 `getViewTypeCount()` 和 `getItemViewType(int position)` 两个方法。在 `getView` 方法中我们可以根据不同的 viewType 加载不同的布局文件。

ListView 如何定位到指定位置

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

可以通过 `ListView` 提供的 `lv.setSelection(listView.getPosition())`方法。

磊哥聊编程
号，磊哥聊编程：得最新版，面试题
关注公众号，磊哥聊编程：得最新版，面试题
关注公众号，磊哥聊编程：得最新版，面试题
关注公众号，磊哥聊编程：得最新版，面试题
关注公众号，磊哥聊编程：得最新版，面试题
公众号，磊哥聊编程：得最新版，面试题
磊哥聊编程

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题