

Android 开发中常见的一些问题专题

2. 混合开发有了解吗？

<https://www.cnblogs.com/kingplus/p/5588339.html>

<https://www.jianshu.com/p/09b00ebf5e15>

3. 知道哪些混合开发的方式？说出它们的优缺点和各自使用场景？（解答：比如：RN，weex，H5，小程序，WPA 等。做 Android 的了解一些前端 js 等还是很有好处的）；

Ract Native (FaceBook)、Weex (阿里)、H5、小程序、WAP、Flutter、APPCan 等；

Ract Native: <https://www.cnblogs.com/zengkefu/p/5978945.html>

Weex:

WAP:

6. 服务器只提供数据接收接口，在多线程或多进程条件下，如何保证数据的有序到达？

7. 动态布局的理解

<https://www.jianshu.com/p/06c9c6685108>

使用 xml 动态加载

直接使用 java 代码加载

8. 怎么去除重复代码？

<http://toughcoder.net/blog/2014/10/25/tricks-to-remove-duplicates/>

<https://blog.csdn.net/hitlion2008/article/details/6737537>

9. 画出 Android 的大体架构图

11. ListView 图片加载错乱的原理和解决方案

ListView item 缓存机制：为了使得性能更优，ListView 会缓存行 item(某行对应的 View)。ListView 通过 adapter 的 getView 函数获得每行的 item。

滑动过程中

1) 如果某行 item 已经滑出屏幕，若该 item 不在缓存内，则 put 进缓存，否则更新缓存；

2) 获取滑入屏幕的行 item 之前会先判断缓存中是否有可用的 item，如果有，做为 convertView 参数传递给 adapter 的 getView。

出现的问题：

1) 行 item 图片显示重复，当前行 item 显示了之前某行 item 的图片。

比如 ListView 滑动到第 2 行会异步加载某个图片，但是加载很慢，加载过程中 listView 已经滑动到了第 14 行，且滑动过程中该图片加载结束，第 2 行已不在屏幕内，根据上面介绍的缓存原理，第 2 行的 view 可能被第 14 行复用，这样我们看到的就是第 14 行显示了本该属于第 2 行的图片，造成显示重复。

2) 行 item 图片显示闪烁

如果第 14 行图片又很快加载结束,所以我们看到第 14 行先显示了第 2 行的图片,立马又显示了自己的图片进行覆盖造成闪烁错乱。

解决方法

通过上面的分析我们知道了出现错乱的原因是异步加载及对象被复用造成的,如果每次 getView 能给对象一个标识,在异步加载完成时比较标识与当前行 item 的标识是否一致,一致则显示,否则不做处理即可。

12. 动态权限适配方案, 权限组的概念

<https://blog.csdn.net/yuguqinglei/article/details/80375702>

13. Android 系统为什么会设计 ContentProvider?

在系统部署一个又一个 Android 应用之后,系统里将会包含多个 Android 应用,有时候需要在不同的应用之间共享数据,比如现在又一个短信接收应用,用户想把数据接收到陌生短信的发信人添加到联系人管理应用中,就需要在不同应用之间共享数据。对于这种需要在不同应用之间共享数据的需求,当然可以让一个应用程序直接去操作另一个应用程序所记录的数据,比如操作它所记录的 SharedPreferences,文件或者是数据库,这种方式太杂乱了:不同的应用程序记录数据的方式差别很大,这种方式不利于应用程序之间进行数据交换。

为了在应用程序之间交换数据,Android 提供了 ContentProvider,ContentProvider 是不同应用程序之间进行数据交换的标准 API,当一个应用程序需要把自己的数据暴露给其他程序使用时,该应用程序就可以通过提供 ContentProvider 来实现;其他应用程序就可以通过 ContentProvider 来操作 ContentProvider 暴露的数据。

ContentProvider 也是 Android 应用的四大组件之一,与 Activity、Service/BroadcastReceiver 的相似,他们需要在 AndroidManifest.xml 文件中进行配置。

一旦某个应用程序通过 ContentProvider 暴露了自己的数据操作接口,那么不管该应用程序是否启动,其他应用程序都可以通过该接口来操作该应用程序的内部数据,包括增加数据、删除数据、修改数据、查询数据等。

<https://blog.csdn.net/yhaolpz/article/details/51304345>

<https://blog.csdn.net/yhaolpz/article/details/51304345>

14. 下拉状态栏是不是影响 activity 的生命周期

不会影响。

15. 如果在 onStop 的时候做了网络请求, onResume 的时候怎么恢复?

恢复的是网络请求暂停后恢复? 还是页面更新?

stop 的时候请求被暂停, onStart 的时候检测重新恢复请求即可

如果是恢复页面请求后的页面数据，分两种，1 activity 被销毁，那么使用 saveInstanceState 存储数据，onRestoreInstanceState() 恢复数据，2，没有被销毁，那就不需要恢复

17. Bitmap 的 recycler()

Bitmap 类的构造方法都是私有的，所以开发者不能直接 new 出一个 Bitmap 对象，只能通过 BitmapFactory 类的各种静态方法来实例化一个 Bitmap。仔细查看 BitmapFactory 的源代码可以看到，生成 Bitmap 对象最终都是通过 JNI 调用方式实现的。所以，加载 Bitmap 到内存里以后，是包含两部分内存区域的。简单的说，一部分是 Java 部分的，一部分是 C 部分的。这个 Bitmap 对象是由 Java 部分分配的，不用的时候系统就会自动回收了，但是那个对应的 C 可用的内存区域，虚拟机是不能直接回收的，这个只能调用底层的功能释放。所以需要调用 recycle() 方法来释放 C 部分的内存。从 Bitmap 类的源代码也可以看到，recycle() 方法里也的确是调用了 JNI 方法了的。

```
public void recycle() {
    if (!mRecycled) {
        if (nativeRecycle(mNativeBitmap)) {
            // return value indicates whether native
            pixel object was actually recycled.
            // false indicates that it is still in use at the native
            level and these
            // objects should not be collected now. They will be
            collected later when the
            // Bitmap itself is collected.
            mBuffer = null;
            mNinePatchChunk = null;
        }
        mRecycled = true;
    }
}
```

<https://www.jianshu.com/p/b5c8e98ff5b0>

19. ViewPager 使用细节，如何设置成每次只初始化当前的 Fragment，其他的不初始化？

Fragment 懒加载

https://blog.csdn.net/e_Inch_Photo/article/details/69054511

20. 点击事件被拦截，但是想传到下面的 View，如何操作？

21. 微信主页面的实现方式

22. 微信上消息小红点的原理

23. CAS 介绍

<https://blog.csdn.net/v123411739/article/details/79561458>

<https://blog.csdn.net/mmoren/article/details/79185862>