

厚积方能薄发，通往Android封神之路的知识体系，开发移动电商app

原创

k186_5189 于 2021-12-28 17:54:18 发布 36 收藏

分类专栏: [程序员](#) 文章标签: [面试](#) [移动开发](#) [android](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/k186_5189/article/details/122199186

版权



[程序员](#) 专栏收录该内容

29 篇文章 0 订阅

订阅专栏

- 动态规划
- 贪心算法
- 分支限界法

Android基础

- [Application生命周期](#)
- [Android Activity生命周期](#)
- [Android Service、IntentService, Service和组件间通信](#)
- [Activity的onNewIntent](#)
- [Fragment的懒加载实现, 参数传递与保存](#)
- [ContentProvider实例详解](#)
- [BroadcastReceiver使用总结](#)
- [Android消息机制](#)
- [Binder机制, 共享内存实现原理](#)
- [Android 事件分发机制](#)
- [Android 多线程的实现: Thread、HandlerThread、AsyncTask、IntentService、RxJava](#)
-

《Android学习笔记总结+最新移动架构视频+大厂安卓面试真题+项目实战源码讲义》

【docs.qq.com/doc/DSkNLaERkbnFoS0ZF】 完整内容开源分享

ActivityThread工作原理

- 嵌套滑动实现原理
- RecyclerView与ListView(缓存原理, 区别联系, 优缺点)
- View的绘制原理, 自定义View, 自定义ViewGroup
- View、SurfaceView 与 TextureView
- 主线程Looper.loop为什么不会造成死循环
- ViewPager的缓存实现
- requestLayout, invalidate, postInvalidate区别与联系
- AndroidP新特性
- Android两种虚拟机
- ADB常用命令
- Asset目录与res目录的区别
- Android SQLite的使用入门

Android开发高级

引子:

Android高级工程师招聘要求:

- 1.熟悉Android SDK, 熟悉Android UI, 熟悉Android各种调试工具;
- 2.有丰富的Android应用架构能力, 能够独立主导并架构App;
- 3.Mobile Web 开发经验; 具备各种复合技能: 熟悉iOS、H5、Python、.NET等多种开发语言的优先考虑;
- 4.对Android性能优化, 安全, 软件加固, 自动化测试有深刻认识;
- 5.博客, 开源项目

Android技术难点

AIDL、Binder、多进程、View的绘制流程、事件分发、消息队列等。

这类知识对于定位自己为高级Android工程师的人来说是必须掌握的, 同时他也是能鉴别高级和初中级工程师的一块试金石, 其中binder是Android系统进程间通信最重要的手段之一, 现阶段app的发展离不开多进程的运用, 经常会启动例如定位、推送等需要在后台开启的进程来保证主进程的内存运行; 所以合理的使用多进程也是十分必要的; view的绘制是我们自定义控件的理论基础, 只有掌握了view是如何绘制的才能个性化的自定义控件; 事件分发一直是Android开发的难点之一, 也是必须掌握的; 关于handler机制也是android的一块难点, 因为包括AsyncTask、系统启动、IntentService等底层都是通过handler来实现的, 所以掌握后handler机制不仅能提高你的实战开发能力, 更能让你系统的了解整个android系统运作的情况。

Android框架层源码掌握

- Android包管理机制，核心PackageManagerService
- Window管理，核心WindowManagerService
- Android Activity启动和管理，核心ActivityManagerService
- 根Activity工作流程
- Context关联类
- 各种原理，经典第三方库源码系列
- 自定义LayoutManager，RecyclerView中如何自定义LayoutManager
- VLayout实现原理，即如何自定义LayoutManager
- Glide加载原理，缓存方案，LRU算法
- Retrofit的实现与原理
- OKHttp3的使用，网络请求中的Intercept
- EventBus实现原理
- ButterKnife实现原理
- RxJava实现原理
- Dagger依赖注入
- 热修复实现原理，解决方案
- 组件化原理和解决方案

Android进程通信以及多进程开发

Android 多进程和Application关系

经典解决方案：多进程通信解决方案：**Andromeda**

- Android动画机制
- Android绘图原理
- Android页面恢复

Android的页面恢复采用以下两个方法：

- onSaveInstanceState(Bundle outState)
- onRestoreInstanceState(Bundle savedInstanceState)

onSaveInstanceState：当Activity容易被系统销毁时，会触发该方法。具体的说：

- 用户点击Home键
- 用户点击Home键，切换到其他应用程序
- 有电话来了等附加操作

混合开发及Android WebView应用

混合开发涉及到的知识点主要包括：

- APP调用WebView加载url
- 掌握WebView的封装，了解所有的WebSettings配置，掌握WebViewClient、* WebChromeClient
- 掌握WebView和Native双向通信机制，会自己封装双向通信中间件
- 对WebView的封装可参考：GitHub: AgentWeb
- 对通信中间件原理解理解：GitHub: webprogress

Gradle，自动化构建，持续集成相关

Android系统

Android Studio编译过程

其中使用到的编译工具：

aapt、aidl、Java Compiler、dex、zipalign

主要步骤描述：

- 通过aapt打包res资源文件，生成R.java、resources.arsc和res文件（二进制 & 非二进制如res/raw和pic保持原样）
- 处理.aidl文件，生成对应的Java接口文件
- 通过Java Compiler编译R.java、Java接口文件、Java源文件，生成.class文件
- 通过dex命令，将.class文件和第三方库中的.class文件处理生成classes.dex
- 通过apkbuilder工具，将aapt生成的resources.arsc和res文件、assets文件和classes.dex一起打包生成apk
- 通过Jarsigner工具，对上面的apk进行debug或release签名
- 通过zipalign工具，将签名后的apk进行对齐处理。

App启动加载过程

Android虚拟机 Android App运行的沙箱原则

Android架构

在Android源码中最重要的三个类：ActivityManagerService / PackageManagerService / View，推荐大家周末的时候可以去阅读下这部分的源码，阅读源码能提高我们今后设计架构自己代码的能力，同时也能从底层了解整个android系统的运行原理，其他一些比如主线程的消息循环、主线程如何和AMS如何跨进程交互、SystemServer进程中的各种Service的工作方式、AsyncTask的工作原理等。这些知识也是作为一个Android高级开发工程师必须掌握的，不能整天沉溺于ui和四大组件的交互，要站在更高的角度去考虑Android的有些问题。

- MVC模式
- MVP模式
- MVVM模式
- CLEAN模式
- 组件化开发
- 跨平台开发：Flutter、ReactNative（RN未来要黄，了解一下就好）

移动开发外围

服务器开发相关

- SpringBoot技术
- Restful API开发
- 网络协议理解：TCP/IP、HTTP/HTTPS、OSI七层协议



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)