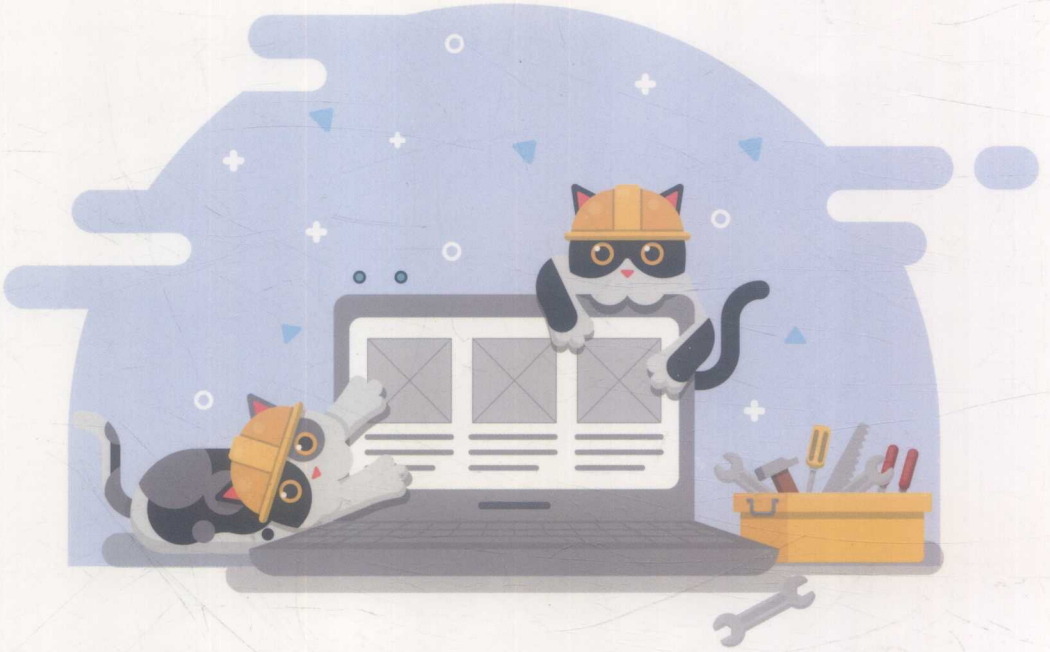
本书从实际工程问题入手，深入剖析源码！
从实践中逐步得出解决方案，是Android工程师
从量变到质变的随身锦囊！

Broadview®
www.broadview.com.cn



Android 工程化最佳实践

金凯 / 著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Android
工程化最佳实践

金凯 / 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书从工程实践角度详细阐述了 Android 的知识内容，全书分为基础知识和工程优化两部分。在工程优化部分专门增加了常用的 App 编译提速和瘦身的内容，对于大型分层项目的测试技巧也有所涉及。

本书涵盖 Android 开发的实际业务知识，涉及 Dialog、Intent、Fragment 等源码的核心细节分析，并扩展了一部分框架设计的内容，章节最后总结了开箱即用的开源库方案，实现从理论到实际的完整论述。最后还给出了抓包工具的使用技巧，帮助读者能方便地寻找到适合自己的工具集。

本书适合中、高级 Android 程序员阅读，也可以作为初级程序员进阶学习的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Android 工程化最佳实践 / 金凯著. —北京：电子工业出版社，2019.3
ISBN 978-7-121-35924-8

Ⅰ. ①A… Ⅱ. ①金… Ⅲ. ①移动终端－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字（2019）第 011494 号

策划编辑：陈晓猛
责任编辑：宋亚东
印　　刷：山东华立印务有限公司
装　　订：山东华立印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×980　1/16　印张：25　字数：623 千字
版　　次：2019 年 3 月第 1 版
印　　次：2019 年 3 月第 1 次印刷
定　　价：89.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

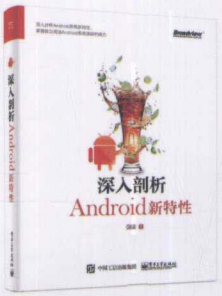
本书咨询联系方式：（010）51260888-819，faq@phei.com.cn。

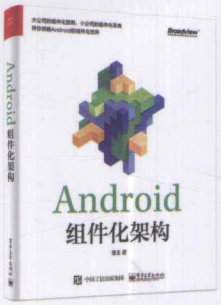
作者简介

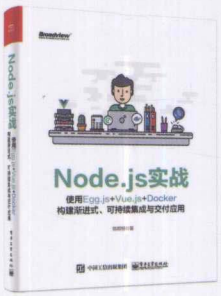
金 凯

爱奇艺高级工程师。致力于摆脱程序思维，寻找各种简易的手段来方便程序的开发工作。目前的研究方向为Android业务基础框架和AI相关方面。

/好书分享/







拒绝堆砌臃肿，支持纯正原创

欢迎投稿：chenxm@phei.com.cn

此为试读，需要完整PDF请访问：www.ertongbook.com

前　言

开发者的焦虑

当今，Android 和 iOS 手机无论在系统还是外形上都趋于雷同，App 功能的差异也是微乎其微，移动端发展趋于缓慢已成为开发者的共识。移动开发者面对小程序、React Native、Flutter 和 Weex 跨平台方案都显得焦躁不安，对于自己未来需要发展和生根的方向或多或少地会感到迷茫。

从市场的角度来说，公司缺少的并非基础的开发人员，更希望可以招聘到有大量实践经验的开发者。要进阶成为高级开发者，更需要阅读有深度、偏重实践的书籍。因此，我从 2015 年 8 月开始在 GitHub 上创建了“Android-Best-Practices”项目。这个项目的目的是将经实际验证的经验分享出来，力求达到“学完即用”的效果。在收到出版社的邀请后，我将相关的文章进行重写，同时新增大量内容，目的是**让更多的开发者可以在短时间内得到更多的实际经验**。

面对当前移动开发的生存环境，开发者的无力感会越发严重。作为移动应用开发者，应该了解与本职工作相关的知识，同时多涉猎其他邻域。如果你想快速巩固移动开发的知识后再投身到大前端中去，那么这本书很适合你。**当阅读完本书的每一章后，你收获的是已经经过时间检验的知识，需要做的仅仅是去实践它、理解它。**

写作风格

本书不会像其他书籍一样提供完整的代码下载，也不会花大量的篇幅分析源码。官方每隔几个月就会对源码的实现进行各种修改，而源码的设计思路则较为稳定。“授人以鱼不如授人以渔”，**本书会将分析源码的过程提炼出来，主要阐述设计思路和关键的代码，甚至会为了可读性删除没必要的代码逻辑。**这样做的目的是希望读者既可以快速理解源码的思路，又不会陷入代码的困境中。

本书内容

本书分为基础知识和工程优化两部分。基础知识部分讲述基础知识，读者可以找到合适的第三方库来提高业务开发速度；工程优化部分讲述工程实践方面的内容，仔细阅读此部分，可以提升自身内在的知识广度。

基础知识

在实际工程中，Android 本身的 API 大多不会被直接使用，需要做二次封装和扩展。基础知识部分将基础的知识提升到一个实践的层面，不再讲述如何使用它们，而是讲如何通过封装这些 API 解决实际中的各种问题。

第 1 章 Java 的反射是必知必会的内容，我们需要知道：

- 什么时候该用反射，Android 中的反射和 JDK 中的反射有何不同？
- 反射低效的原因是什么，如何让其变得高效。反射的缓存该如何实现？
- 为什么 Android P 不让反射被@hide 标记的 API，其影响是什么？
- 做一个好的反射封装，应注意什么问题，这种封装的意义是什么？

第 2 章 Log 是最简单的代码了，这里要讲的是日志的高级知识：

- 打印日志的时机是什么，什么日志应该留在 Release 版本中，这么做是否安全？
- 如何区分不同项目组的日志，如何避免日志间的相互干扰？
- 如何优化和封装日志，为什么说日志的分发功能是必须具备的？
- 微信是如何用 MMAP 提升日志的写盘成功率的，Xlog 的设计目的是什么？

第 3 章 Intent 是非常简单的对象，着重来讲：

- 如何封装和扩展 Intent，为什么它只能存储基本对象，Intent 的本质是什么？
- 如何在使用 Intent 时避免管理 Intent 的 Key，动态下发 Intent 的思路是什么？
- Intent 的最大数据存放量是多少，该如何传递一个大对象？
- Intent 传递 map 对象时会有什么隐患，linkedHashMap 可以直接存入 Intent 吗？

第 4 章 SharedPreferences 作为基本的存储对象，需要了解如下知识点：

- 用 SharedPreferences 存储 Java 对象有什么问题，如何清理其缓存？
- 为什么拆分 SharedPreferences 后可以有效地减少内存的消耗？
- SharedPreferences 的线程、进程安全吗，它是如何引起 ANR 的？
- 如果将 SharedPreferences 中的数据放云端，该如何设计这种云同步的功能？

第 5 章 Fragment 是开发者很容易出错的类，这里需要思考：

- Fragment 和 Activity 的生命周期是依次执行的吗？动态添加时的生命周期会如何执行？
- Square 不建议用 Fragment，但不用 Fragment 的话，该如何模块化地管理 UI 呢？
- Google 公司提出的 Lifecycle-Components 和 Fragment 有关系吗？使用的是什么原理？
- Fragment 的嵌套和回退栈是一个难点，Jetpack 中的 Navigation 是怎么简化它的？

工程优化

工具是开发者的武器，拥有一个好的武器可以战无不胜。在工程优化篇中，会提到如何利用工具和技巧帮助开发者优化项目。这部分的知识点较为零碎，但学起来并不难，很适合读者用碎片时间进行学习。

第 6 章 Dialog 已经不推荐直接使用了，取而代之的是 dialogFragment，关于它我们需要知道：

- Dialog 和 alertDialog，alertDialog 和 dialogFragment 是什么关系？

- 如何实现一个可以随时展示的全局弹窗，这对于 App 的任务系统有什么好处？
- 为什么说完全可以通过 Style 实现自定义的 Dialog，Dialog 的背景该如何编写？
- alertDialog 引起内存泄露和崩溃的原因是什么，在开发中该如何防御？

第 7 章 Gradle 是常用工具，本章不准备讲它的基础知识，会直接介绍实际的例子：

- 如何自动化生成 versionCode 和 versionName，Gradle 还能做哪些自动化处理？
- 如何强制整个 App 的某个库均使用统一的版本号，如何更好地管理签名文件？
- 如何引入本地的 Maven 仓库，本地依赖的写法是什么，如何制作一个本地库？
- 如何使用 Exclude Lib、Jar、Module 等，对于产生类冲突的两个库该如何解决？

第 8 章 本章除讲述如何提升打 Debug 包的速度外，还介绍了如下内容：

- 为什么小范围的配置优化不如直接升级机器硬件来得快？
- Google 废弃 compile()的原因是什么，新的依赖 API 有哪些，各有什么不同？
- 如何启用 Gradle 缓存，减少每次编译时的网络请求是提速的关键吗？
- Android 有几种签名方案，不同签名方案对应的多渠道打包策略有何不同？

第 9 章 每家公司都希望 APK 越来越小，本章探讨的是切实有效的瘦身策略，主要解惑的点为：

- APK 是由哪些文件组成的？APK 和 Zip 是什么关系，签名相关的文件在哪个目录中？
- 瘦身开始前应该做哪些准备工作，为什么这个看似无关紧要的工作却很重要。
- APK 中的.9 图应该放在哪个文件夹中，到底要不要做屏幕分辨率的适配。
- ABI 是什么意思，从云端下载 so 时需要注意什么问题，具体的可行性如何？

第 10 章 测试用例的编写一直被我们忽视，本章是一个入门级的内容，涵盖了如下的知识点：

- mock 的目的是什么，常见的 mock 框架有哪些，我们一般会如何进行技术选型？
- 测试分为逻辑测试和 UI 测试，哪个才是移动测试的重点？
- 如果想要在 PC 端直接测试 Android 的逻辑，目前最方便的测试框架是哪个？
- 网络层是一个典型的单元测试和集成测试的场景，该如何写好异步代码的测试用例？

第 11 章 本章在前半部分会讲与 Debug 相关的内容，后半部分将介绍一些极其好用的插件，阅读后你将知道：

- 如何在 Debug 时动态打 Log，它和在代码中硬编码的 Log 有什么区别？
- 如何通过 Debug 工具检查内存泄露，怎样清楚地知道当前内存中的对象数目？
- Facebook 的 Sonar 是什么，这个平台可以带来怎样的想象力？
- 号称史上最强的数据库调试插件是什么，它真的值得购买吗，免费的替代品是什么？

第 12 章 使用代理工具是开发人员和测试人员的必会技能，本章会介绍 Whistle 的主要功能：

- 支持用正则制定过滤规则，返回的 JSON 能自动进行格式化。
- 支持为多个环境配置不同的代理，可以灵活地进行环境的切换。
- 可以直接编写 mock 规则，自动化地生成 mock 数据，并且能统一管理 mock 的 values。
- 支持调试 Webview，可以很方便地进行真机上 H5 页面的调试。

致谢

写书本身是一件很严肃的事情，尤其是技术方面的书籍。我是一个讨厌被限制的人，所以一直以来都是偷懒写“网文”，原本不愿意出版图书。下面要感谢一些朋友，没有他们我可能永远都无法迈出这一步，也无法得到一本真正属于自己的作品。

- 感谢博文观点团队的赏识和邀请，如果没有贵团队，我根本就不会有写书的想法和机会。
- 感谢本书的编辑陈晓猛，他的沟通能力和执行力是保障本书快速出版的关键。
- 感谢“不吃鱼同学”，要不是她的支持和鼓励，我就没有勇气开始做这件长达一年的事情。
- 感谢所有在微博关注我的朋友们，你们的支持和批评是鞭策我前进的动力。
- 感谢聚美、网易和美团的一些朋友们，你们给予了我很多的建议和灵感。
- **更要感谢即将开始阅读本书的你，这本书就是为你而写的，你的阅读是我最大的荣耀！**

联系方式

如果你发现了本书的纰漏或是对内容有更好的建议，可以通过下列方式联系到我：
微博：@天之界线 2010；
邮箱：developer-kale@foxmail.com；
QQ 群：390272666；
GitHub：tianzhijiexian。

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误**：您对书中内容的修改意见可在 提交勘误 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方 读者评论 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：http://www.broadview.com.cn/35924

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

此为试读，需要完整PDF请访问：www.ertongbook.com