



# Android高级进阶

顾浩鑫 / 著

Android高级研发工程师50个必备技能点



中国工信出版集团



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

# Android高级进阶

顾浩鑫 / 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书是Android的进阶学习指南，主要为Android初中级开发者进阶所需的知识，高级开发者也可以从本书中发现很多共鸣点。本书从8个方面对50个知识点进行分类讲解，包括基础篇、系统架构篇、经验总结篇、新技术篇、性能优化篇、移动安全篇、工具篇、测试篇。熟练掌握这些知识点后，应该就能够应付实际项目开发中的绝大部分问题了。

本书的主要目的在于给读者一个完整的Android中高级开发者知识图谱。笔者希望通过本书的系统讲解，能够帮助读者在面试和工作中收获自己满意的成绩。

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高级进阶 / 顾浩鑫著. —北京：电子工业出版社，2016.10  
ISBN 978-7-121-29845-5

I. ①A… II. ①顾… 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（2016）第208289号

策划编辑：董 英

责任编辑：徐津平

印 刷：北京天宇星印刷厂

装 订：北京天宇星印刷厂

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：33.5 字数：643千字

版 次：2016年10月第1版

印 次：2016年11月第2次印刷

印 数：3001~5000册 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zlbs@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。

Android 应用开发发展到今天，已经成为一个非常成熟的技术方向，市面上涌现了大量关于 Android 开发的图书，总的来说可以分为以下几类。

- **入门实战类**：这类图书是市面上占比最多的，也是中初级开发者比较青睐的一类图书，基本上介绍的都是Android的基础知识，例如界面开发、数据存储、网络通信、数据库操作、传感器使用等，最后附上一两个简单实战项目的介绍。建议读者在入门的时候买一本评价还不错的这类图书学习一遍，然后结合实际项目进行知识巩固即可，之后这类书基本上扮演的是工具书的角色，就是在忘记某个知识点的时候到书中查一查重新温习一遍。
- **源码分析类**：这类图书又可以分为Framework层源码分析类和Native层源码分析类两种，这类图书主要针对从事Android系统开发的读者。当然，从事应用开发的读者如果对Android Framework等底层的实现原理有所了解的话那也是大有裨益的，而且相对而言更有优势。
- **安全和逆向分析类**：这类图书主要介绍的是Android平台的软件安全、逆向分析及加解密技术等，主要涉及Android软件的静态分析、动态调试、破解及反破解等技术。这个方向的知识既有Android应用开发者所需要的，也有Android系统开发者所需要的，市场上也有专门的Android系统安全相关的职位。
- **系统移植和驱动类**：这类图书主要介绍的是Android内核、移植和驱动开发的整个底层嵌入式开发知识，这一类知识点其实并不能算作移动端开发，本质上属于传统的嵌入式开发领域，毕竟Android底层就是一个Linux系统。

以上便是目前笔者所看到的 Android 图书市场所覆盖的内容，可以说是大而全。但事实上在笔者看来，还有一类图书是目前没有出现的，那就是体现一线互联网公司工程实践中涉及的重要知识点，笔者将之归类为第 5 类：工程实践类。希望本书能够弥补这个空白，并期待能够看到更多此类图书的出现。

## 本书内容

本书从结构上分为 8 篇，共 50 个知识点。

- **基础篇**：这是占比最大的一个篇章，虽然名为基础篇，但你应该理解为是高级进阶里面的基础知识，而不是Android开发入门的基础知识。本篇主要包含Android View、动画、Support Library、Gradle、NDK、ANR、异步技术、注解、数据序列化和Hybrid等知识点。
- **系统架构篇**：本篇侧重介绍架构和项目整体的相关知识，主要包括UI架构、事件总线、编码规范和技术堆栈等知识点。
- **经验总结篇**：本篇侧重介绍Android工程实践中得出的经验，主要包括64K方法数限制、插件框架、推送原理、APP瘦身、Crash统计等知识点。
- **新技术篇**：本篇侧重介绍新近出现的技术点，当然可能本书出版时已经没那么新了，但并不妨碍其重要性，主要包括函数式编程思想简介、依赖注入、Kotlin、React Native、在线热修复、AOP和Facebook Buck等知识点。
- **性能优化篇**：本篇侧重介绍目前Android平台常见的性能优化相关知识，主要包括代码优化、图片优化、电量优化、布局优化和网络优化等知识点。
- **移动安全篇**：本篇侧重介绍很多应用中经常会忽略的安全知识，主要包括混淆、反编译、密钥隐藏、加固和如何编写安全的代码等知识点。
- **工具篇**：本篇介绍了Android开发中会用到的几个工具，主要包括Android Studio、Stetho、LeakCanary和Redex等知识点。
- **测试篇**：本篇介绍了测试相关的知识，这部分是很多开发人员经常容易忽略的内容，本质上属于测试领域，但开发人员需要有所了解，主要包括Android单元测试、UI自动化测试、静态代码分析和自动化构建等工具。

## 读者对象

本书的读者对象如下。

- Android应用研发工程师
- 计算机相关专业的学生

## 致谢

感谢董英编辑找到我并建议我出一本关于 Android 开发相关的图书，不然也不会有这本书的存在；感谢妻子恋恋对我的写作和生活的鼓励和陪伴；感谢父母和老师对我的培养；感谢我曾经就职的公司：华为、百度、平安科技，以及现在所在的平安金融科技；感谢华为 PTN、华为推送、百度文库、百度导航、百度打车、Hyperion、任意门、银行一账通等项目的兄弟姐妹们，是过往的这些公司和团队磨砺了我。

## 勘误与互动

读者如果发现本书文字、代码和图片等信息存在错误或者纰漏，欢迎反馈给我。对书中内容或者 Android 应用开发中有什么疑问，也可以与我互动，届时将在微信公众号定期发布本书的勘误信息，并解答大家的疑问。我的相关信息如下。

- 微信公众号：ASCE1885
- 微博：[http://weibo.com/asce885?is\\_all=1](http://weibo.com/asce885?is_all=1)
- GitHub：<https://github.com/ASCE1885>

# 目录

## 第1篇 基础篇

第1章 Android触摸事件传递机制	2
1.1 触摸事件的类型	2
1.2 事件传递的三个阶段	3
1.3 View的事件传递机制	4
1.4 ViewGroup的事件传递机制	10
第2章 Android View的绘制流程	16
2.1 绘制的整体流程	17
2.2 MeasureSpec	17
2.3 Measure	19
2.4 Layout	22
2.5 Draw	22
第3章 Android 动画机制	25
3.1 逐帧动画 (Frame Animation)	25
3.1.1 XML 资源文件方式	25
3.1.2 代码方式	26
3.2 补间动画 (Tween Animation)	27
3.2.1 插值器 Interpolator	27
3.2.2 AlphaAnimation	29
3.2.3 ScaleAnimation	30
3.2.4 TranslateAnimation	31
3.2.5 RotateAnimation	32
3.2.6 自定义补间动画	34
3.3 属性动画 (Property Animation)	34
3.3.1 Evaluator	35
3.3.2 AnimatorSet	36
3.3.3 ValueAnimator	36
3.3.4 ObjectAnimator	38
3.4 过渡动画 (Transition Animation)	40
第4章 Support Annotation Library 使用详解	46
4.1 Nullness 注解	47

4.2	资源类型注解 .....	48
4.3	类型定义注解 .....	50
4.4	线程注解 .....	52
4.5	RGB 颜色值注解 .....	52
4.6	值范围注解 .....	53
4.7	权限注解 .....	53
4.8	重写函数注解 .....	54
4.9	返回值注解 .....	55
4.10	@VisibleForTesting .....	55
4.11	@Keep .....	55
<b>第5章 Percent Support Library使用详解 .....</b>		<b>57</b>
<b>第6章 Design Support Library使用详解 .....</b>		<b>62</b>
6.1	Snackbar .....	62
6.2	TextInputLayout .....	63
6.3	TabLayout .....	64
6.4	NavigationView .....	65
6.4.1	导航菜单 .....	66
6.4.2	导航头部 .....	67
6.5	FloatingActionButton .....	70
6.5.1	使用浮动操作按钮 .....	70
6.5.2	其他选项 .....	71
6.5.3	点击事件 .....	71
6.6	CoordinatorLayout .....	72
6.7	CollapsingToolbarLayout .....	73
6.8	BottomSheetBehavior .....	75
<b>第7章 Android Studio中的NDK开发 .....</b>		<b>77</b>
7.1	ABI的基本概念 .....	77
7.2	引入预编译的二进制 C/C++ 函数库 .....	79
7.3	直接从 C/C++ 源码编译 .....	79
7.3.1	配置 ndk.dir 变量 .....	79
7.3.2	在 Gradle 中配置 NDK 模块 .....	79
7.3.3	添加 C/C++ 文件到指定的目录 .....	81
7.4	使用 .so 文件的注意事项 .....	81
7.4.1	使用高平台版本编译的 .so 文件运行在低版本的设备上 .....	81
7.4.2	混合使用不同的 C++ 运行时编译的 .so 文件 .....	82
7.4.3	没有为每个支持的 CPU 架构提供对应的 .so 文件 .....	82
7.4.4	将 .so 文件放在错误的地方 .....	82
7.4.5	只提供 armeabi 架构的 .so 文件而忽略其他 ABI 的 .....	83



<b>第8章 Gradle 必知必会</b> .....	85
8.1 共享变量的定义 .....	85
8.2 通用配置 .....	87
8.3 aar 函数库的引用 .....	88
8.4 签名和混淆的配置 .....	90
<b>第9章 通过Gradle打包发布函数库到JCenter和Maven Central</b> .....	92
9.1 Maven Central 和 JCenter .....	92
9.1.1 Maven Central .....	93
9.1.2 JCenter .....	93
9.2 Android Studio 获取函数库的原理 .....	94
9.3 上传函数库到 JCenter .....	96
9.3.1 步骤一：在 Bintray 网站上注册一个账号 .....	96
9.3.2 步骤二：创建一个 Sonatype 账号 .....	96
9.3.3 步骤三：在 Bintray 网站使能自动签名 .....	97
9.3.4 步骤四：生成 POM 相关的信息 .....	100
9.3.5 步骤五：上传函数库到 Bintray .....	104
9.3.6 步骤六：发布 Bintray 用户仓库到 JCenter .....	107
9.3.7 步骤七：同步函数库到 Maven Central .....	109
<b>第10章 Builder模式详解</b> .....	110
10.1 经典的 Builder 模式 .....	110
10.2 Builder 模式的变种 .....	113
10.3 变种 Builder 模式的自动化生成 .....	119
10.4 开源函数库的例子 .....	122
<b>第11章 注解在 Android 中的应用</b> .....	124
11.1 注解的定义 .....	124
11.2 标准注解 .....	125
11.2.1 编译相关注解 .....	125
11.2.2 资源相关注解 .....	125
11.2.3 元注解 .....	125
11.3 运行时注解 .....	127
11.4 编译时注解 .....	127
11.4.1 定义注解处理器 .....	127
11.4.2 注册注解处理器 .....	131
11.4.3 android-apt 插件 .....	132
<b>第12章 ANR产生的原因及其定位分析</b> .....	134
12.1 ANR 产生的原因 .....	135
12.2 典型的 ANR 问题场景 .....	135

12.3 ANR 的定位和分析 .....	136
12.3.1 Logcat 日志信息 .....	136
12.3.2 traces.txt 日志信息 .....	138
12.4 ANR的避免和检测 .....	141
12.4.1 StrictMode .....	141
12.4.2 BlockCanary .....	142
<b>第13章 Android异步处理技术 .....</b>	<b>144</b>
13.1 Thread .....	144
13.2 HandlerThread .....	146
13.3 AsyncQueryHandler .....	149
13.4 IntentService .....	150
13.5 Executor Framework .....	153
13.6 AsyncTask .....	155
13.7 Loader .....	156
13.8 总结 .....	159
<b>第14章 Android数据序列化方案研究 .....</b>	<b>160</b>
14.1 Serializable .....	160
14.2 Parcelable .....	166
14.3 SQLiteDatabase .....	169
14.4 SharedPreferences .....	170
14.5 JSON .....	171
14.6 Protocol Buffers及Nano-Proto-Buffers .....	171
14.7 FlatBuffers .....	171
<b>第15章 Android WebView Java 和 JavaScript 交互详解 .....</b>	<b>173</b>
15.1 Java 调用 JavaScript .....	173
15.2 JavaScript 调用 Java .....	174

## 第2篇 系统架构篇

<b>第16章 MVP模式及其在Android中的实践 .....</b>	<b>180</b>
16.1 MVP 的基本概念 .....	180
16.2 MVP 与 MVC 的区别 .....	181
16.3 MVP 的开源实现 .....	182
16.3.1 Android-Architecture .....	182
16.3.2 TODO-MVP .....	182
16.3.3 TODO-MVP-Loaders .....	183
16.3.4 TODO-MVP-Clean .....	183
16.3.5 TODO-Databinding .....	184

16.3.6 其他开源参考实现 .....	184
16.4 MVP 的好处 .....	185
16.5 MVP 存在的问题 .....	185
<b>第17章 MVVM模式及Android DataBinding实战 .....</b>	<b>186</b>
17.1 Data Binding 表达式 .....	187
17.2 数据对象 .....	188
17.3 数据绑定 .....	188
17.4 事件绑定 .....	189
<b>第18章 观察者模式的拓展：事件总线 .....</b>	<b>191</b>
18.1 为何要使用 .....	191
18.2 原理 .....	192
18.3 开源实现 .....	193
18.3.1 EventBus .....	193
18.3.2 otto .....	194
18.4 与观察者模式及 Android 广播的区别 .....	196
<b>第19章 书写简洁规范的代码 .....</b>	<b>197</b>
19.1 Java 编码规范 .....	197
19.1.1 源代码文件的定义 .....	197
19.1.2 源代码文件的结构 .....	197
19.1.3 遵循的格式 .....	198
19.1.4 命名约定 .....	200
19.1.5 Javadoc .....	200
19.2 Android 命名规范 .....	200
19.2.1 布局文件的命名 .....	200
19.2.2 资源文件的命名 .....	201
19.2.3 类的命名 .....	201
19.3 CheckStyle 的使用 .....	202
<b>第20章 基于开源项目搭建属于自己的技术堆栈 .....</b>	<b>203</b>
20.1 APP 的整体架构 .....	203
20.2 技术选型的考量点 .....	205
20.3 日志记录能力 .....	205
20.4 JSON 解析能力 .....	207
20.4.1 gson .....	207
20.4.2 jackson .....	207
20.4.3 Fastjson .....	208
20.4.4 LoganSquare .....	208

20.5	数据库操作能力 .....	210
20.5.1	ActiveAndroid .....	210
20.5.2	ormlite .....	211
20.5.3	greenDAO .....	211
20.5.4	Realm .....	212
20.6	网络通信能力 .....	213
20.6.1	android-async-http .....	213
20.6.2	OkHttp .....	215
20.6.3	Volley .....	216
20.6.4	Retrofit .....	217
20.7	图片缓存和显示能力 .....	217
20.7.1	BitmapFun .....	218
20.7.2	Picasso .....	218
20.7.3	Glide .....	218
20.7.4	Fresco .....	219
20.7.5	Android-Universal-Image-Loader .....	219

### 第3篇 经验总结篇

第21章	64K方法数限制原理与解决方案 .....	222
21.1	64K 限制的原因 .....	222
21.2	使用 MultiDex 解决 64K 限制的问题 .....	223
21.2.1	Android 5.0 之前的版本 .....	223
21.2.2	Android 5.0 及之后的版本 .....	223
21.3	如何避免出现 64K 限制 .....	223
21.4	配置 MultiDex .....	224
21.5	MultiDex Support Library 的局限性 .....	226
21.6	在开发阶段优化 MultiDex 的构建 .....	227
第22章	Android 插件框架机制研究与实践 .....	230
22.1	基本概念 .....	231
22.1.1	宿主和插件 .....	231
22.1.2	ClassLoader 机制 .....	231
22.2	开源框架 .....	231
22.2.1	android-pluginmgr .....	232
22.2.2	dynamic-load-apk .....	232
22.2.3	DynamicAPK .....	232
22.2.4	DroidPlugin .....	233
22.2.5	Small .....	234

<b>第23章 推送机制实现原理详解</b>	<b>235</b>
23.1 推送的开源实现方案	236
23.1.1 基于XMPP协议	236
23.1.2 基于MQTT协议	236
23.2 推送的第三方平台	236
23.3 自己实现推送功能	237
23.3.1 长连接的建立(TCPConnectThread)	237
23.3.2 数据的发送(TCPSendThread)	237
23.3.3 数据的接收(TCPReceiveThread)	238
23.3.4 心跳包的实现(TCPHeartBeatThread)	240
<b>第24章 APP瘦身经验总结</b>	<b>241</b>
24.1 APP为什么变胖了	241
24.2 从APK文件的结构说起	242
24.3 优化图片资源占用的空间	245
24.3.1 无损压缩[ImageOptim]	246
24.3.2 有损压缩[ImageAlpha]	246
24.3.3 有损压缩[TinyPNG]	246
24.3.4 PNG/JPEG转换为WebP	246
24.3.5 尽量使用NinePatch格式的PNG图	247
24.4 使用Lint删除无用资源	248
24.5 利用Android Gradle配置	248
24.5.1 minifyEnable	248
24.5.2 shrinkResources	249
24.5.3 resConfigs	249
24.5.4 ndk.abiFilters	250
24.6 重构和优化代码	250
24.7 资源混淆	251
24.8 插件化	251
<b>第25章 Android Crash日志收集原理与实践</b>	<b>252</b>
25.1 Java层Crash捕获机制	253
25.1.1 基本原理	253
25.1.2 线程信息	254
25.1.3 SharedPreferences信息	255
25.1.4 系统设置	257
25.1.5 Logcat中的日志记录	261
25.1.6 自定义Log文件中的内容	264
25.1.7 MemInfo信息	266

25.2	Native 层 Crash 捕获机制 .....	267
25.3	Crash 的上报 .....	269

## 第4篇 新技术篇

第26章	函数式编程思想及其在Android中的应用 .....	272
26.1	代码的简化 .....	274
26.2	Operators 简介 .....	275
第27章	依赖注入及其在Android中的应用 .....	277
27.1	基本概念 .....	277
27.1.1	构造函数注入 .....	278
27.1.2	Setter 函数注入 .....	279
27.1.3	接口注入 .....	279
27.2	为何需要框架 .....	280
27.3	开源框架的选择 .....	280
27.3.1	ButterKnife .....	280
27.3.2	RoboGuice .....	282
27.3.3	Dagger .....	285
27.3.4	Dagger2 .....	288
27.3.5	框架的对比 .....	289
第28章	Android世界的Swift: Kotlin在Android中的应用 .....	290
28.1	选择 Kotlin 的原因 .....	290
28.2	Kotlin 的安装和配置 .....	291
28.3	Kotlin 语言的特性 .....	292
28.3.1	可表达性 .....	292
28.3.2	空类型安全 .....	294
28.3.3	扩展函数 .....	295
28.4	Kotlin 的 Gradle 配置 .....	296
28.5	将 Java 类转换成 Kotlin 类 .....	299
28.6	相关资料 .....	302
第29章	React Native For Android入门指南 .....	304
29.1	环境配置 .....	304
29.1.1	Homebrew .....	304
29.1.2	nvm .....	305
29.1.3	Node.js .....	305
29.1.4	watchman .....	306
29.1.5	flow .....	306

29.2	Android 开发环境的要求 .....	306
29.3	React Native 工程配置 .....	307
29.3.1	安装react-native .....	307
29.3.2	生成工程 .....	307
29.4	Android Studio 工程概览 .....	308
29.5	React Native 依赖库修改为本地 .....	314
29.5.1	下载 react-native.aar .....	314
29.5.2	react-native.aar 的文件内容 .....	315
29.5.3	Gradle 本地依赖 .....	316
29.5.4	将 node_modules 上传到 svn/git .....	318
29.6	React Native 学习建议 .....	319
<b>第30章</b>	<b>Android在线热修复方案研究 .....</b>	<b>320</b>
30.1	在线热修复的基本流程 .....	320
30.2	Dexposed .....	321
30.2.1	如何集成 .....	322
30.2.2	基本用法 .....	323
30.2.3	在线热修复 .....	325
30.2.4	平台的限制 .....	328
30.3	AndFix .....	329
30.3.1	如何集成 .....	329
30.3.2	补丁包生成工具 .....	331
30.3.3	平台的限制 .....	332
30.4	Nuwa .....	332
30.4.1	基本原理 .....	332
30.4.2	如何集成 .....	333
30.4.3	补丁生成工具 .....	334
30.4.4	平台的限制 .....	334
30.5	总结 .....	334
<b>第31章</b>	<b>面向切面编程及其在Android中的应用 .....</b>	<b>335</b>
31.1	AOP 的基本概念 .....	335
31.2	代码织入的时机 .....	336
31.3	基于 AspectJ 实现 Android 平台的 AOP .....	337
31.3.1	Hugo 的用法简介 .....	337
31.3.2	Hugo 的实现原理 .....	339
31.4	其他 AOP 开源框架 .....	344
<b>第32章</b>	<b>基于Facebook Buck改造Android构建系统 .....</b>	<b>345</b>
32.1	Buck环境配置 .....	346
32.1.1	Homebrew 方式 .....	346

32.1.2	手动构建方式	346
32.1.3	安装 Watchman	348
32.1.4	安装 Android SDK 和 Android NDK	348
32.2	快速创建基于 Buck 构建的 Android 工程	349
32.3	Buck 的基本概念	351
32.3.1	构建规则 (Build Rule)	352
32.3.2	构建目标 (Build Target)	354
32.3.3	构建文件 (Build File)	355
32.3.4	构建目标模式 (Build Target Pattern)	356
32.4	项目改造实战	357
32.4.1	步骤一: 手动下载工程依赖的第三方 Jar包或者aar包	357
32.4.2	步骤二: 将 R.* 常量修改为非 final 的	357
32.4.3	步骤三: 创建 BUCK 文件	358
32.4.4	步骤四: 编译 Buck 的 buck-android-support	363
32.4.5	步骤五: Exopackage 的使用	363
32.5	Buck 的自动化改造	366

## 第5篇 性能优化篇

第33章	代码优化	368
33.1	数据结构的选择	368
33.2	Handler 和内部类的正确用法	370
33.3	正确地使用 Context	373
33.3.1	Context 的种类	374
33.3.2	错误使用 Context 导致的内存泄漏	374
33.3.3	不同 Context 的对比	376
33.4	掌握 Java 的四种引用方式	376
33.5	其他代码微优化	377
33.5.1	避免创建非必要的对象	377
33.5.2	对常量使用 static final 修饰	378
33.5.3	避免内部的 Getters/Setters	378
33.5.4	代码的重构	378
第34章	图片优化	379
34.1	图片的格式	379
34.1.1	JPEG	380
34.1.2	PNG	380
34.1.3	GIF	380
34.1.4	WebP	380
34.2	图片的压缩	380
34.2.1	无损压缩 ImageOptim	381



34.2.2	有损压缩 ImageAlpha .....	381
34.2.3	有损压缩 TinyPNG .....	381
34.2.4	PNG/JPEG 转换为 WebP .....	381
34.2.5	尽量使用 NinePatch 格式的 PNG 图 .....	382
34.3	图片的缓存 .....	382
<b>第35章</b>	<b>电量优化 .....</b>	<b>383</b>
35.1	BroadcastReceiver .....	383
35.2	数据传输 .....	384
35.3	位置服务 .....	384
35.4	AlarmManager .....	386
35.5	WakeLock .....	386
<b>第36章</b>	<b>布局优化 .....</b>	<b>388</b>
36.1	include 标签共享布局 .....	388
36.2	ViewStub 标签实现延迟加载 .....	389
36.3	merge 标签减少布局层次 .....	391
36.4	尽量使用 CompoundDrawable .....	392
36.5	使用 Lint .....	393
<b>第37章</b>	<b>网络优化 .....</b>	<b>395</b>
37.1	避免 DNS 解析 .....	395
37.2	合并网络请求 .....	395
37.3	预先获取数据 .....	396
37.4	避免轮询 .....	396
37.5	优化重连机制 .....	396
37.6	离线缓存 .....	396
37.7	压缩数据大小 .....	396
37.8	不同的网络环境使用不同的超时策略 .....	397
37.9	CDN 的使用 .....	397

## 第6篇 移动安全篇

<b>第38章</b>	<b>Android混淆机制详解 .....</b>	<b>400</b>
38.1	Java 代码的混淆 .....	400
38.1.1	Proguard 的特性 .....	401
38.1.2	Proguard 的使能和配置 .....	401
38.1.3	proguard-rules.pro 文件的编写 .....	404
38.1.4	Proguard 生成的文件 .....	407
38.1.5	Proguard 混淆规则汇总 .....	409
38.2	Native (C/C++) 代码的混淆 .....	409