

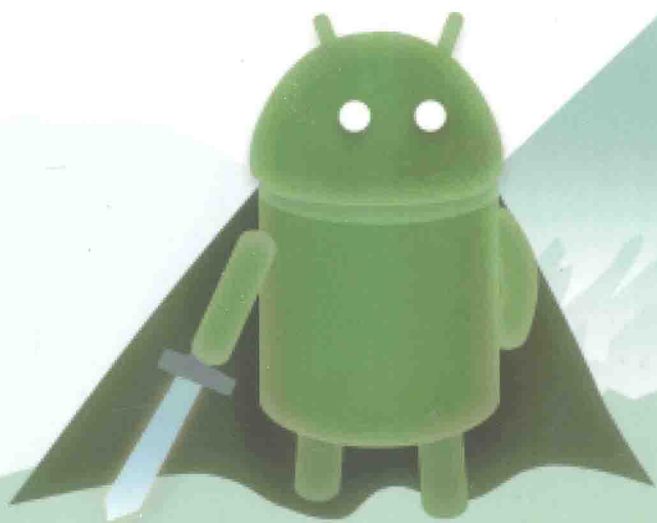
本书基于Android 7.0和Android Studio，对Android开发进阶要点进行深入讲解，为工程师的进阶之路带来指引和光明。

Broadview[®]
www.broadview.com.cn

Android

进阶之光

刘望舒 著



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

进阶之光



刘望舒 著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是一本 Android 进阶类书籍，书中各知识点由浅入深、环环相扣，最终这些知识点形成了一个体系结构。本书共分为 11 章。第 1 章介绍 Android 5.0 到 Android 7.0 的新特性。第 2 章介绍 Material Design。第 3 章介绍 View 体系，包括 View 的事件分发、工作流程、自定义 View 等知识点。第 4 章介绍多线程的知识。第 5 章介绍网络编程与网络框架的知识。第 6 章介绍常用的设计模式。第 7 章介绍事件总线。第 8 到第 10 章介绍架构设计所需要的知识点。第 11 章简单介绍 Android 系统框架与 MediaPlayer 框架。

本书详细并深入讲解 Android 开发者必备的和前沿的知识，适合有一定开发基础的开发者阅读，这有助于他们提高技术水平；同时，本书系统的知识体系结构也令高级开发者从中获益良多。

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 进阶之光 / 刘望舒著. —北京：电子工业出版社，2017.7
ISBN 978-7-121-31530-5

I. ①A… II. ①刘… 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 (2017) 第 108511 号

策划编辑：付 睿

责任编辑：李云静

印 刷：北京京科印刷有限公司

装 订：北京京科印刷有限公司

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：787×1092 1/16

印张：30.75

字数：764 千字

版 次：2017 年 7 月第 1 版

印 次：2017 年 7 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。

前言

为什么写这本书

从 2008 年 Android 系统发布以来，Android 已经发展了 9 年。在此期间，Android 开发也相当火热。这时，大量人员涌入 Android 开发职场，并导致 Android 开发人才市场相对饱和。如此一来，很多 Android 开发者会发现工作越来越难找，企业对开发者的要求也越来越高，企业需求最多的不再是初中级别的 Android 工程师，而是 Android 高级工程师。但是，Android 高级工程师有限。有些人在从事了几年开发工作后，对很多技术的理解却仍停留在会用的阶段。他们对于原理不求甚解，这导致他们进入技术瓶颈期并长期无法得到提高。很多开发者为了突破技术瓶颈，看了大量的网络视频和博客。尽管如此，他们仍旧无法突破自身的技术瓶颈。其主要原因是，他们没有将学到的知识点形成体系化。因此，这就需要有一本成体系的进阶书来帮助这些开发人员成为 Android 高级工程师并突破自身的技术瓶颈。纵观市面上 Android 开发相关的书籍，其中大部分书籍是入门级别类图书，还有一部分系统源码、逆向分析和系统移植类图书，而关于应用开发进阶的书籍则少得可怜。本书正是一本成体系的应用开发进阶图书，书中所要传达的不仅仅是知识，其同时还会告诉读者以下几点。

1. 要关注 Android 新技术；
2. Java 基础和设计模式很重要；
3. 学习框架要深入其原理；
4. 要学习架构设计；
5. 要了解和学习系统源码。

本书内容

本书共分为 11 章，各章内容如下。

- 第 1 章介绍 Android 5.0、Android 6.0、Android 7.0 的新特性，包括 Android 5.0 的 RecyclerView、Android 6.0 的运行时权限机制和 Android 7.0 的多窗口模式等知识点。
- 第 2 章介绍 Material Design 以及 Design Support Library 常用的控件，并给出实例将 Design Support Library 中的常用控件结合在一起使用。
- 第 3 章介绍 View 相关的进阶知识，包括 View 的滑动、View 的事件分发和 View 的工作流程。最后结合以上知识点来介绍自定义 View。
- 第 4 章介绍多线程编程，本章不仅包括基础的线程知识，还会介绍线程同步和线程池等进阶知识点，最后结合这些知识点来分析 Android 7.0 的 AsyncTask 的源码。
- 第 5 章介绍网络编程的基础知识以及常用的网络框架：Volley、OkHttp 和 Retrofit 的使用方法和原理分析。
- 第 6 章将设计模式进行分类，并介绍每个分类中常用的设计模式。
- 第 7 章介绍事件总线 EventBus 和 otto 的使用方法和原理。
- 第 8 章介绍函数响应式编程 RxJava 的使用方法，包括 RxJava 的基本使用、操作符、使用场景和源码分析等知识点。
- 第 9 章介绍注解的知识点和依赖注入框架 ButterKnife 和 Dagger2 的使用方法以及原理。
- 第 10 章介绍 Android 应用架构设计，包括 MVP 框架以及 MVP 结合 RxJava 和 Dagger2，还有 MVVM 框架相关的 Data Binding 支持库。
- 第 11 章主要是指引导读者进行 Android 系统源码阅读并带其入门，介绍 Android 系统框架、源码目录和阅读源码工具，并以分析 MediaPlayer 框架的源码作为示例。

本书特色

本书主要有以下特点。

- 本书整体结构由浅入深，从最简单的第 1 章到难一些的第 11 章，其难度是逐步加深的。
- 本书为了分析一些框架的原理，会介绍一些知识点做铺垫，比如为了更好地介绍依赖注入框架，需要首先了解注解相关的知识点。再比如要分析 AsyncTask 的源码，则需要了解线程池和阻塞队列等知识点。
- 本书的知识点环环相扣，比如要介绍 MVP 框架的设计，就需要先学习 Retrofit、RxJava 和 Dagger2 的相关知识点。
- 本书对于很多知识点都有很深入的讲解。其中，对于常用的框架，比如 OkHttp、Retrofit、EventBus 和 RxJava 等不只是讲解了如何使用，而且更加深入地介绍了其原理。
- 本书是目前市场上详细介绍有关 Android 新特性、Material Design、网络框架、事件总线、RxJava、依赖注入框架和应用架构设计的难得一见的图书。

读者对象

本书的章节设计是由浅入深的，适合 Android 初、中、高级工程师阅读。本书的定位是学习 Android 的第二本书，其阅读前提是要有一定的 Android 基础。

致谢

感谢本书的策划编辑付睿，她在 CSDN 博客中发现了，并积极推动本书的出版进度，这才使得本书得以及时出版。感谢本书的责任编辑李云静，她审稿时很细致，这使得书中的一些错误能被提早发现并改正。感谢我的父母在写书过程中对我的不断鼓励，这样我才得以全力以赴地投入编写工作。感谢所有关注我的朋友们，你们的鼓励和认可为我写博客以及写书带来了不可或缺的动力。

勘误与互动

本人虽已竭尽全力，但书中难免会有错误，欢迎大家向我反馈，我也会在独立博客和 CSDN 博客中定期发布本书的勘误信息。

本书互动地址

独立博客：<http://liuwangshu.cn>

CSDN 博客：<http://blog.csdn.net/itachi85>

GitHub：<https://github.com/henrymorgen>

微信公众号：刘望舒

QQ 交流群：499174415

源码下载

<https://github.com/henrymorgen/android-advanced-light>

<http://www.broadview.com.cn/31530>

读者服务

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。

- **下载资源**：本书如提供示例代码及资源文件，均可在[下载资源处](#)下载。
- **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/31530>



目录

第 1 章 Android 新特性	1
1.1 Android 5.0 新特性	1
1.1.1 Android 5.0 主要新特性概述	1
1.1.2 替换 ListView 和 GridView 的 RecyclerView	3
1.1.3 卡片 CardView	13
1.1.4 3 种 Notification	18
1.1.5 Toolbar 与 Palette	23
1.2 Android 6.0 新特性	29
1.2.1 Android 6.0 主要新特性概述	29
1.2.2 运行时权限机制	30
1.3 Android 7.0 新特性	43
1.3.1 Android 7.0 主要新特性概述	43
1.3.2 多窗口模式	44
1.4 本章小结	47
第 2 章 Material Design	48
2.1 Material Design 概述	48
2.1.1 核心思想	48
2.1.2 材质与空间	49
2.1.3 动画	49
2.1.4 样式	50
2.1.5 图标	51
2.1.6 图像	51

2.1.7	组件	51
2.2	Design Support Library 常用控件详解	54
2.2.1	Snackbar 的使用	54
2.2.2	用 TextInputLayout 实现登录界面	55
2.2.3	FloatingActionButton 的使用	60
2.2.4	用 TabLayout 实现类似网易选项卡的动态滑动效果	61
2.2.5	用 NavigationView 实现抽屉菜单界面	68
2.2.6	用 CoordinatorLayout 实现 Toolbar 隐藏和折叠	74
2.3	本章小结	86
第 3 章	View 体系与自定义 View	87
3.1	View 与 ViewGroup	87
3.2	坐标系	89
3.2.1	Android 坐标系	89
3.2.2	View 坐标系	90
3.3	View 的滑动	91
3.3.1	layout()方法	92
3.3.2	offsetLeftAndRight()与 offsetTopAndBottom()	94
3.3.3	LayoutParams (改变布局参数)	95
3.3.4	动画	95
3.3.5	scrollTo 与 scollBy	96
3.3.6	Scroller	98
3.4	属性动画	99
3.5	解析 Scroller	105
3.6	View 的事件分发机制	108
3.6.1	源码解析 Activity 的构成	108
3.6.2	源码解析 View 的事件分发机制	112
3.7	View 的工作流程	119
3.7.1	View 的工作流程入口	119
3.7.2	理解 MeasureSpec	122
3.7.3	View 的 measure 流程	126
3.7.4	View 的 layout 流程	132
3.7.5	View 的 draw 流程	135
3.8	自定义 View	139
3.8.1	继承系统控件的自定义 View	140
3.8.2	继承 View 的自定义 View	141
3.8.3	自定义组合控件	147

3.8.4 自定义 ViewGroup	152
3.9 本章小结	164
第 4 章 多线程编程	165
4.1 线程基础	165
4.1.1 进程与线程	165
4.1.2 线程的状态	167
4.1.3 创建线程	168
4.1.4 理解中断	170
4.1.5 安全地终止线程	171
4.2 同步	173
4.2.1 重入锁与条件对象	173
4.2.2 同步方法	175
4.2.3 同步代码块	176
4.2.4 volatile	177
4.3 阻塞队列	183
4.3.1 阻塞队列简介	183
4.3.2 Java 中的阻塞队列	184
4.3.3 阻塞队列的实现原理	186
4.3.4 阻塞队列的使用场景	188
4.4 线程池	190
4.4.1 ThreadPoolExecutor	190
4.4.2 线程池的处理流程和原理	192
4.4.3 线程池的种类	193
4.5 AsyncTask 的原理	197
4.6 本章小结	203
第 5 章 网络编程与网络框架	204
5.1 网络分层	204
5.2 TCP 的三次握手与四次挥手	205
5.3 HTTP 协议原理	207
5.3.1 HTTP 简介	207
5.3.2 HTTP 请求报文	208
5.3.3 HTTP 响应报文	209
5.3.4 HTTP 的消息报头	210
5.3.5 抓包应用举例	211
5.4 HttpClient 与 HttpURLConnection	212

5.4.1	HttpClient.....	212
5.4.2	URLConnection.....	216
5.5	解析 Volley.....	218
5.5.1	Volley 基本用法.....	218
5.5.2	源码解析 Volley.....	223
5.6	解析 OkHttp.....	231
5.6.1	OkHttp 基本用法.....	232
5.6.2	源码解析 OkHttp.....	240
5.7	解析 Retrofit.....	255
5.7.1	Retrofit 基本用法.....	255
5.7.2	源码解析 Retrofit.....	261
5.8	本章小结.....	270
第 6 章 设计模式.....		271
6.1	设计模式六大原则.....	271
6.2	设计模式分类.....	273
6.3	创建型设计模式.....	273
6.3.1	单例模式.....	274
6.3.2	简单工厂模式.....	277
6.3.3	工厂方法模式.....	279
6.3.4	建造者模式.....	281
6.4	结构型设计模式.....	284
6.4.1	代理模式.....	285
6.4.2	装饰模式.....	288
6.4.3	外观模式.....	291
6.4.4	享元模式.....	295
6.5	行为型设计模式.....	298
6.5.1	策略模式.....	298
6.5.2	模板方法模式.....	301
6.5.3	观察者模式.....	304
6.6	本章小结.....	307
第 7 章 事件总线.....		308
7.1	解析 EventBus.....	308
7.1.1	使用 EventBus.....	308
7.1.2	源码解析 EventBus.....	314
7.2	解析 otto.....	324

7.2.1 使用 otto.....	324
7.2.2 源码解析 otto.....	327
第 8 章 函数响应式编程.....	333
8.1 RxJava 基本用法.....	333
8.1.1 RxJava 概述.....	333
8.1.2 RxJava 基本实现.....	334
8.1.3 RxJava 的不完整定义回调.....	336
8.2 RxJava 的 Subject.....	338
8.3 RxJava 操作符入门.....	339
8.3.1 创建操作符.....	339
8.3.2 变换操作符.....	340
8.3.3 过滤操作符.....	344
8.3.4 组合操作符.....	349
8.3.5 辅助操作符.....	352
8.3.6 错误处理操作符.....	355
8.3.7 条件操作符和布尔操作符.....	357
8.3.8 转换操作符.....	360
8.4 RxJava 的线程控制.....	362
8.5 RxJava 的使用场景.....	362
8.5.1 RxJava 结合 OkHttp 访问网络.....	362
8.5.2 RxJava 结合 Retrofit 访问网络.....	364
8.5.3 用 RxJava 实现 RxBus.....	368
8.6 RxJava 源码解析.....	370
8.6.1 RxJava 的订阅过程.....	371
8.6.2 RxJava 的变换过程.....	372
8.6.3 RxJava 的线程切换过程.....	376
8.7 本章小结.....	381
第 9 章 注解与依赖注入框架.....	382
9.1 注解.....	382
9.1.1 注解分类.....	382
9.1.2 定义注解.....	384
9.1.3 注解处理器.....	385
9.2 依赖注入的原理.....	392
9.2.1 控制反转与依赖注入.....	392
9.2.2 依赖注入的实现方式.....	393

9.3 依赖注入框架.....	395
9.3.1 为何使用依赖注入框架.....	395
9.3.2 解析 ButterKnife.....	395
9.3.3 解析 Dagger2.....	405
9.4 本章小结.....	421
第 10 章 应用架构设计.....	422
10.1 MVC 模式.....	422
10.2 MVP 模式.....	423
10.2.1 应用 MVP 模式.....	424
10.2.2 MVP 结合 RxJava 和 Dagger2.....	431
10.3 MVVM 模式.....	438
10.3.1 解析 Data Binding.....	439
10.3.2 应用 Data Binding.....	457
10.4 本章小结.....	459
第 11 章 系统架构与 MediaPlayer 框架.....	460
11.1 Android 系统架构.....	460
11.2 Android 系统源码目录.....	463
11.2.1 整体结构.....	463
11.2.2 应用层部分.....	464
11.2.3 应用框架层部分.....	465
11.2.4 C/C++程序库部分.....	465
11.3 Source Insights 使用.....	466
11.4 MediaPlayer 框架.....	467
11.4.1 Java Framework 层的 MediaPlayer 分析.....	467
11.4.2 JNI 层的 MediaPlayer 分析.....	469
11.4.3 Native 层的 MediaPlayer 分析.....	471
11.5 本章小结.....	478
后记.....	479

第 1 章

Android 新特性

作为本书的第 1 章，肯定是全书比较简单的内容，而本书面向中高级的开发者，所以基础知识讲得较少。在本章笔者会介绍 Android 5.0、Android 6.0 以及 Android 7.0 的新特性，并会详细介绍 Android 5.0 和 Android 6.0 的新特性。虽然 Android 7.0 已经发布了，但是目前 Android 5.0 和 Android 6.0 的技术仍旧没有普及。笔者希望通过本章的学习，读者能够了解各版本有什么新的功能、带给用户怎样的新体验，且能够掌握 Android 的新特性并尝试运用到项目中。

1.1 Android 5.0 新特性

Android 5.0 Lollipop 是 Google 于 2014 年 10 月 15 日发布的 Android 操作系统。北京时间 2014 年 6 月 26 日，Google I/O 2014 开发者大会在旧金山正式召开，发布了 Android 5.0 的开发者预览版。下面我们先来看看 Android 5.0 给我们带来了什么。

1.1.1 Android 5.0 主要新特性概述

作为一个 Android 开发者，我们需要了解最近的 Android 版本带来了什么特性，这样更有利于开发。谷歌（Google）在 Android 5.0 中带给我们很多惊喜。

1. 全新的 Material Design 设计风格

Material Design 是一种大胆的平面化创新（见图 1-1）。换句话说，谷歌希望能够让 Material Design 给用户带来纸张化的体验。这种新的视觉语言，在基本元素的处理上，借鉴了传统的印刷设计，以及字体版式、网格系统、空间、比例、配色和图像使用等这些基础的平面设计规范。另外，Material Design 还推崇实体隐喻理念，利用实体的表面与边缘的质感打造出视觉线索，

让用户感受到真实性。熟悉的触感让用户可以快速理解并认知。在设计中可以在符合物理规律的基础上灵活地运用物质，打造出不同的使用体验。为了吸引用户的注意力，Material Design 还带来了有意义而且更合理的动态效果，以及维持整个系统的连续性体验。需要注意的是 Material Design 虽然是在 Android 5.0 时被提出来的，但是它也是在不断更新的，所以关于 Material Design 的内容会在第 2 章专门介绍。



图 1-1 Material Design

2. 支持多种设备

Android 系统的身影早已出现在多种设备中，比如：智能手机、平板电脑、笔记本电脑、智能电视、汽车、智能手表甚至是各种家用电子产品等。

3. 全新的通知中心设计

谷歌在 Android 5.0 中加入了全新风格的通知系统。改进后的通知系统会优先显示对用户来说比较重要的信息，而将不太紧急的内容隐藏起来。用户只需要向下滑动就可以查看全部的通知内容，如图 1-2 所示。

4. 支持 64 位 ART 虚拟机

Android 5.0 内部的性能上也提升了不少，它放弃了之前一直使用的 Dalvik 虚拟机，改用了 ART 虚拟机，实现了真正的跨平台编译，在 ARM、X86、MIPS 等无处不在。

5. Overview

多任务视窗现在有了一个新的名字，Overview。在界面中，每一个 App 都是一张独立的卡片，拥有立体式的层叠效果，用户可以设定“最近应用程序”，通过滑动来快速切换 App，如图 1-3 所示。



图 1-2 全新的通知中心设计

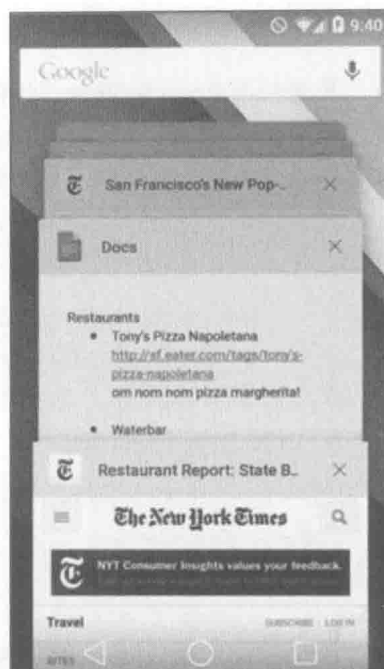


图 1-3 全新的“最近应用程序”

6. 设备识别解锁

现在个人识别解锁已经被普遍使用，比如当特定的智能手表出现在 Android 设备的附近时，就会直接绕过锁屏界面进行操作。而 Android 5.0 也增加了这种针对特定设备识别解锁的模式。换句话说，当设备没有检测到附近有可用的信任设备时，就会启动安全模式以防止未授权访问。

7. Ok Google 语音指令

当手机处于待机状态时，对你的手机轻轻说声“Ok Google”，手机即刻被唤醒，只需说出简单的语言指令，如播放音乐、查询地点、拨打电话和设定闹钟等，一切只需“说说”而已。

8. Face unlock 面部解锁

在 Android 5.0 中，Google 花费大力气优化了面部解锁功能。当用户拿起手机处理锁屏界面上的消息通知时，面部解锁功能便自动被激活。随意浏览几条消息之后，手机已经默默地完成了面部识别。

1.1.2 替换 ListView 和 GridView 的 RecyclerView

有了 ListView、GridView，为什么还需要 RecyclerView 这样的控件呢？从整体上看，RecyclerView 架构提供了一种插拔式的体验，它具有高度的解耦、异常的灵活性和更高的效率，通过设置它提供的不同 LayoutManager、ItemDecoration、ItemAnimator 可实现更加丰富多样的效果。但是 RecyclerView 也有缺点和让人头疼的地方：设置列表的分割线时需要自定义，另外列表的点击事件需要自己去实现。

1. 配置 build.gradle

要想使用 RecyclerView，我们首先要导入 support-v7 包。因为我用的是 Android Studio（本书的所有例子均基于 Android Studio），所以在此需要在 build.gradle 中加入如下代码以自动导入 support-v7 包，记得配置完再重新 Build 一下工程。

```
dependencies {
    ...
    compile 'com.android.support:appcompat-v7:22.2.0'
    compile 'com.android.support:recyclerview-v7:22.1.0'
}
```

2. 使用 RecyclerView

```
RecyclerView mRecyclerView= (RecyclerView) this.findViewById(R.id.id_
recyclerview);
    //设置布局管理器
    mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
    // 设置 item 增加和删除时的动画
    mRecyclerView.setItemAnimator(new DefaultItemAnimator());
    mHomeAdaper=new HomeAdapter(this, mList);
    mRecyclerView.setAdapter(mHomeAdaper);
```

与 ListView 不同的一点就是，需要设置布局管理器用于设置条目的排列样式，可以是垂直排列或者水平排列。这里我们设置 `setLayoutManager(new LinearLayoutManager(this))` 表示条目是线性排列的（默认是垂直排列的）。

```
public LinearLayoutManager(Context context) {
    this(context, VERTICAL, false);
}
```

如果想要设置为水平排列，可以按如下代码所示编写：

```
LinearLayoutManager layoutManager=new LinearLayoutManager(this);
layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
mRecyclerView.setLayoutManager(layoutManager);
```

此外，RecyclerView 比 ListView 的设置要复杂一些，主要是它需要自己去自定义分割线，设置动画和布局管理器，等等。布局文件 `activity_recycler_view.xml` 如下：

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent" >
```