

O'REILLY®

Broadview®  
www.broadview.com.cn



# React Native Cookbook (中文版)

React Native Cookbook

将Web开发体验带入原生平台

[美] Jonathan Lebensold 著

刘清 文一山 译

 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>

O'REILLY®

# React Native Cookbook

(中文版)

React Native Cookbook

[美] Jonathan Lebensold 著

刘清 文一山 译



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

在多个平台开发应用往往是非常麻烦的，但是使用 React Native，你就能构建出跨平台的移动应用，而且其外观和行为与用 Java 或者 Swift 编写的原生应用并无二致。本书将帮助你理解 React Native 的生态，教你应对一些常见的挑战，比如处理应用的状态、处理设备 I/O，还介绍了部署过程和一些缩短交付时间的技巧，最后讲解了如何编写可维护的代码。

如果你熟悉 JavaScript，而且打算开发跨平台的应用程序，并且选择的是 React 技术栈，那么这本书就是为你准备的。

©2018 by Paradem Consulting.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Publishing House of Electronics Industry, 2018. Authorized translation of the English edition, 2018 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书简体中文版专有出版权由 O'Reilly Media, Inc. 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字：01-2018-5010

### 图书在版编目 (CIP) 数据

React Native Cookbook: 中文版 / (美) 乔纳森·莱本索德 (Jonathan Lebensold) 著; 刘清, 文一山译. —北京: 电子工业出版社, 2018.9

书名原文: React Native Cookbook

ISBN 978-7-121-34897-6

I. ①R… II. ①乔… ②刘… ③文… III. ①JAVA语言—程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第188096号

责任编辑: 许 艳

封面设计: Karen Montgomery 张 健

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

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: 12 字数: 233千字

版 次: 2018年9月第1版

印 次: 2018年9月第1次印刷

定 价: 59.00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。

---

# O'Reilly Media, Inc. 介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始, O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来, 而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者, O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”; 创建第一个商业网站 (GNN); 组织了影响深远的开放源代码峰会, 以至于开源软件运动以此命名; 创立了 Make 杂志, 从而成为 DIY 革命的主要先锋; 公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖, 共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择, O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务或者面授课程, 每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

## 业界评论

“O'Reilly Radar 博客有口皆碑。”

——Wired

“O'Reilly 凭借一系列 (真希望当初我也想到了) 非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人, 他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了: ‘如果你在路上遇到岔路口, 走小路 (岔路)。’ 回顾过去, Tim 似乎每一次都选择了小路, 而且有几次都是转瞬即逝的机会, 尽管大路也不错。”

——Linux Journal

# 前言

当我的第一个 React Native 应用登陆应用商店时，我就知道这项技术非同一般。那时，我已经花了几天的时间来熟悉 iOS 生态，发现自己被 Xcode、Objective-C 和适用于 iOS 的库弄得晕头转向。在只有 Web 开发经验的情况下，我最后一次突击 iOS 开发已经差不多是 8 年前的事情了，我被吓着了。我知道我不是唯一的一个。

React Native 在一夜之间改变了这一切。我在其中发现了熟悉的设计理念和 Web 应用开发的影子。更妙的是，用它开发出来的应用不会成为次等品——我无法忍受僵硬的动画和笨拙的滚动。React Native 是一个开源工具集，将原生应用程序的开发体验带给世界各地无数的 JavaScript 开发人员。

## 谁应该读这本书

你已经非常熟悉编程和 JavaScript。本书假定你处于原生开发的技术选型对比阶段。你可能正在自己的车库中开发下一个伟大的社交媒体平台，或准备将笨重的企业系统变成轻便的移动应用。如果你试图开发跨平台的原生应用程序，需要将其快速推向市场，并选择了 React，那么本书就是为你准备的。本书的所有内容都源于笔者在构建原生应用程序时的经验。

## 为什么写这本书

网上有非常多有关 React Native 的资料：在 StackOverflow 的论坛上和在 React Native 框架在 GitHub 上的 issue 中有大量的文档，能够解决大多数编程中的细节问题。本书

试图讲得更深入一点：如何组织一个项目？如何设计需要获取用户摄像头权限的用户体验？这些都是一些需要思考的常见问题，而且并不一定有一个解决方案。这本书是一本 cookbook（食谱），能提供一个很好的起点。它能激发你，引导你想出自己的解决方案，或者在你陷入困境时为你带来一些启发！

## 简单聊聊如今的 JavaScript

似乎每周 JavaScript 都会用一个新的名字、一套新的语言功能以及新的转译器和编译器重塑自己！无论你将其称为 ECMAScript、ES6、ES6+，还是从 TypeScript、CoffeeScript、NativeScript、Flow、Elm 或 Reason 中进行转译，最终的输出都会在 JavaScript 虚拟机上运行。疲于追踪 JavaScript 的这些变化，要如何才能专注于稳定的技术呢？

我的回答是没有什么标准答案。所有这些工具都只是为了让软件开发人员更有效率。如果一切顺利，那么你应该高枕无忧，因为你知道在下一拨技术被淘汰之前，开源社区会帮助你看出端倪。就本书而言，我会交叉使用术语 JavaScript 和 ES6。在最后一节中，我会逼着你拥抱变化，让你学习如何用基于 OCaml 的函数式编程语言 Reason，重写之前用 ES6 编写的组件！

## 内容介绍

本书分为 6 章。

- 第 1 章讨论 JavaScript 工具以及它们如何与 React Native 配合使用。
- 第 2 章探讨 React Native 生态：了解可用的组件资源，以及如何将其引入你的项目。
- 第 3 章和第 4 章介绍大多数应用程序开发中的一些常见挑战：处理应用程序的状态、处理设备 I/O 以及处理设计资源。
- 第 5 章介绍部署过程和一些缩短交付时间的技巧。
- 第 6 章解决编写可维护的代码的问题：使代码可重用、可移植、可读，添加了一些能在用户使用产品前捕获程序错误的工具。

像对待所有的 cookbook 类图书一样，你最好读一遍书中的例子，看看它们是如何与你正在努力完成的工作联系起来的。如果你已经熟悉 React Native 或者熟悉 Node、NPM

和 Yarn，我建议你跳过第 2 章。如果你编写过原生应用程序，那么第 1 章可能值得一读。

## 在线资源

React Native 依赖的工具大致可以分为三类：JavaScript 工具、Apple SDK 和 Android SDK。React Native 将你的所有代码打包到一个 JavaScript bundle 中，然后在原生平台上运行（例如 Android 或 iOS）。根据《React Native 入门指南》（*React Native: Getting Started Guide*），确保你正确安装了这些原生开发平台。

如果你对 React 没有什么使用经验，*React Overview* (<http://bit.ly/2nIA4vr>) 应该能帮助你大致了解。我建议你在学习本书之前查阅这些相关的参考资料。下面这份清单列出了一组技术，它们都是很多 React Native 开发人员需要了解的知识，其中包括 JavaScript / ES6、NPM、React、React Native 和 Redux。

- 权威指南：*React Native: Getting Started Guide* (<http://bit.ly/2GWQDfo>)
- 快速了解支撑 JavaScript pipeline 的转译器的能力：*Learn ES2015* (<http://bit.ly/2nJb1Z8>)
- Node 包管理器（NPM）的介绍：*What is npm?* (<http://bit.ly/2EalFyf>)
- 关于 React 的通俗易懂的讲解：*React Overview* (<http://bit.ly/2nIA4vr>)
- 社区收集的 React Native 相关资源目录：*Awesome React Native* (<http://bit.ly/2EoLga3>)
- 由 Redux 的创立者讲解状态管理的免费视频教程：*Getting Started with Redux* (<http://bit.ly/2ENagFH>)
- React Native 软件包目录：*Opinionated Catalog of Open Source React Native Packages* (<https://native.directory/>)
- NPM 提供的 React Native 软件包列表：*An Open Catalog of React Native Libraries* (<http://bit.ly/2E8wuRF>)

## 排版约定

本书使用了下列排版约定。

中文楷体或英文斜体 (*Italic*)

表示新的术语、URL、邮件地址、文件名称和文件扩展名。

### 等宽字体 (Constant width)

表示程序片段，也可以用在正文中表示变量名或函数名等程序元素、数据库、环境变量、语句和关键字等。

### 等宽字体加粗 (Constant width bold)

表示应该由用户输入的命令或其他文本。

### 等宽字体加斜 (Constant width italic)

表示应当被用户自定义的值或上下文决定的值所替换的文本。



该图标表示提示或建议。



该图标表示一般注解。



该图标表示警告或警示。

## O'Reilly Safari



Safari

Safari (以前的 Safari Books Online) 是企业、政府、教育者和个人的会员制培训及参考平台。

订阅者可以从一个完全可搜索的数据库中获得来自 250 多家出版商提供的成千上万的书



籍、培训视频、互动教程，这些出版商包括 O'Reilly Media、Harvard Business Review、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Adobe、Focal Press、Cisco Press、John Wiley&Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones&Bartlett 和 Course Technology。

若想获得更多资讯，请访问 <http://oreilly.com/safari>。

## 联系我们

请将对本书的评价和发现的问题通过如下地址通知出版社。

美国：

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)  
奥莱利技术咨询(北京)有限公司

我们提供了本书网页，上面列出了勘误表、示例和其他信息。请通过 <http://bit.ly/reactNativeCookbook> 访问该页。

要给出本书意见或者询问技术问题，请发送邮件到 [bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)。

更多有关书籍、课程、会议和新闻的信息，请见网站 <http://www.oreilly.com>。

在 Facebook 找到我们：<http://facebook.com/oreilly>。

在 Twitter 上关注我们：<http://twitter.com/oreillymedia>。

在 YouTube 上观看：<http://www.youtube.com/oreillymedia>。

## 致谢

感谢本书所有的审校者：Spencer Carli、Matt Hamil 和 Arnar Þór Sveinsson。他们对快速

变化的生态系统的见解以及技术上的反馈使我对本书内容更有信心。我对本书的内容承担全部责任，但如果没有他们的建议，这本书的可读性就会大打折扣。

如果没有 Paradem 团队，特别是如果没有我的联合创始人 Kevin Pratt，我就无法专注于写作，这本书就不可能诞生。我也很感谢 Ezra Hopkins、Scott Luetke 和 Abdullah Norozi，他们协助我通读全书章节。Bivee 公司的 Scott Schaffter 和 Jay Perry 为我提供了写作的灵感，一直鼓励我。感谢 O'Reilly Media，特别是 Nan Barber，他给了我反馈并指导出版流程。

感谢 Facebook 与世界分享它的 React Native 技术。我也很感谢 Reactiflux Discord 聊天室中的人们提供的技术支持，特别是对 Reason 的技术支持。我非常感谢所有向开源社区贡献其知识和源代码的个人和组织。

最后，我要感谢我的爱妻 Tara，她鼓励我前行，并给我不少深思熟虑的见解。

---

## 读者服务

轻松注册成为博文视点社区用户 ([www.broadview.com.cn](http://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/34897>



# 关于作者

Jonathan Lebensold 在摆弄各种带状电缆搭建乐高积木中度过了自己的童年。12 岁时他在 IRC 上给人们提供技术支持——这是他第一次指导别人。后来，Jonathan 为世界五百强企业、非营利机构和初创公司开发大型信息系统。在了解软件设计模式、测试驱动开发和函数式开发后，他对编程的热情一下子爆发了。Jonathan 与人合开了一家软件咨询公司 Cofound Paradem，这也使他有可能会参与全球范围的软件与产品设计研讨会，最近主要是在欧洲和东非。他使用 Ruby、React 和 React Native 帮助团队构建可伸缩、可维护的架构解决方案。你可以通过推特联系他 (<https://twitter.com/jonlebensold>)，不过他有可能正在厨房里专心做苹果派。

# 关于封面

本书封面上的动物是一种北方的苍鹰 (Accipiter Gentilis)，它是一种遍布于欧亚大陆和北美的猛禽。因为这种鸟飞行速度极快，而且喜欢追随猎物进入茂密的植被，几百年来一直是猎鹰运动中非常受欢迎的鸟类。“苍鹰 (goshawk)”这个名字来源于盎格鲁撒克逊语中的“鹅鹰 (goose hawk)”，尽管它们习惯捕食兔子、水禽、松鸡和雉鸡。

苍鹰在鹰族中属于中等体型的成员，但拥有相当大的喙和爪，在狩猎时能为其带来优于其他猛禽的优势。它们的翅膀短而宽，尾巴长，这个特性让它们在森林栖息地中来去自由。不同地理区域的苍鹰，羽毛颜色有所不同，但一般来说，成年苍鹰的眼睛为橙色或红色，背部和翅膀呈蓝灰色或棕灰色，腹部有淡灰色和深色条纹。幼年苍鹰的羽毛为棕色，眼睛为黄色。

雌性苍鹰比雄性体型大得多。交配的夫妻们常常会多年使用同一个巢穴，或至少留在其附近筑巢。雄性通常建造巢穴，而雌性在附近监督，有时也可能会帮助巩固老巢。巢穴的结构可能长 31~47 英寸，宽 20~28 英寸。苍鹰是领土型的鸟类，它们和异性伴侣一起捍卫自己的狩猎场和筑巢地，对抗其他猛禽。

O'Reilly 封面上的许多动物都濒临灭绝，它们是世界的重要组成部分，如果希望了解如何提供帮助，请访问 [animals.oreilly.com](http://animals.oreilly.com)。

封面图片来自 Wood's Animate Creation。封面字体是 URW Type-writer 和 Guardian Sans。

# 目录

前言 .....	vii
<b>第 1 章 React Native 工具链 .....</b>	<b>1</b>
1.1 设置开发环境 .....	1
1.2 使用 Babel 写 ES6 .....	5
1.3 组织项目文件 .....	8
1.4 处理灾难性故障 .....	16
<b>第 2 章 熟悉 React Native 生态 .....</b>	<b>21</b>
2.1 不要重复你自己：实现自定义组件 .....	21
2.2 添加一个开源的进度条 .....	25
2.3 共享自定义组件 .....	34
2.4 路由到登录界面 .....	42
2.5 使用 Redux 作为全局状态管理工具 .....	54
<b>第 3 章 样式与设计 .....</b>	<b>73</b>
3.1 书写样式表 .....	73
3.2 使用 flexbox 构建灵活的布局 .....	77
3.3 导入矢量图和图标 .....	80

3.4 循环动画 .....	88
<b>第 4 章 管理硬件平台 .....</b>	<b>91</b>
4.1 请求使用用户的设备硬件 (iOS) .....	91
4.2 获取分页请求 .....	101
4.3 使用 Redux 和本地存储保存应用状态 .....	107
4.4 使用文件系统 .....	109
<b>第 5 章 起飞! 分享你的应用 .....</b>	<b>119</b>
5.1 自动发布你的应用 .....	119
5.2 把你的 iOS 应用分享给 beta 测试用户 .....	125
5.3 配置应用设置 .....	126
<b>第 6 章 让你的应用可维护 .....</b>	<b>137</b>
6.1 使用 PropTypes 保护你的组件 .....	137
6.2 使用 Flow 检查运行时报错 .....	143
6.3 组件自动化测试 .....	150
6.4 使用 ESLint 维护代码风格 .....	157
6.5 使用 Reason 编写你的应用 .....	167

## React Native 工具链

React Native 生态链上有许多软件工具，如转义器（Babel、Metro、Webpack）、包管理器（NPM、Yarn）、Linter、单元测试框架，等等。本章将涵盖 React Native 项目中会涉及的一些语言基础知识以及开发这些项目时的一些必要开源工具。你可能正在用 JavaScript，或者诸如 TypeScript、ES6+ 这些 JavaScript 的超集语言开发 React Native 应用。希望本章能帮助你了解 JavaScript 惊人的演进速度。



Expo

最近 React Native 团队与 Expo (<https://nodejs.org/>) 合作，可以在开发中无本地开发环境下交付 React Native 应用。这是试用 React Native 的好方式，但是你很可能需要在一定程度上与硬件打交道，本地开发环境对你的效率很重要。

### 1.1 设置开发环境

如果你在其他 Web 项目中使用一些类似工具的话，就会发现自己经常在解决开发环境的配置问题。所以，如同木匠一样，在开工之前，你需要了解这些工具的工作原理以及如何检修它们。

React Native 包含了三套编程环境：Node.js、iOS 和 Android。NPM，也就是 Node Package Manager，必须保证能够正常工作。

## 问题

React Native 依赖了很多不同的工具。如何确定它们都被正确配置了呢？我们来依次检查，确认一下。

### Node 和 Watchman

Node.js（通常缩写为 Node）允许你的计算机像浏览器运行 Web 页面中的脚本一样在本地运行 JavaScript。因为 Node.js 直接运行在你的操作系统上，所以 Node 代码可以像 PHP、Python、Perl 和 Ruby 等其他编程语言一样，通过包装 / 绑定 C 语言库来解决同样的编程问题。

Watchman 就是这样一个小工具，监测文件在本地的变动并触发相应的事件。这个工具可以让你在模拟器里运行最新的代码而无须重新编译整个项目，安装起来非常简便 (<http://bit.ly/2FUNzyJ>)。



#### 安装 Node.js

不同的操作系统，安装 Node 的方式也不尽相同。最简单的办法是参考 Node.js 网站 (<https://nodejs.org/>)。如果你使用的是 Mac OS，通常更好的方法是使用 Mac OS 的包管理器 Homebrew 安装 Node.js。

**检查 Node.js 是否安装正确。**你的计算机上可能会安装多个版本的 Node.js。像 Node Version Manager (NVM, 参见 <https://github.com/creationix/nvm>) 之类的版本管理器可以帮你管理 Node 的不同版本，以便不同的开发项目使用不同版本的 Node。

类 POSIX 操作系统 (Linux、BSD 或者 Mac OS) 可以通过符号链接 (symlink) 来支持多版本共存。

如果你使用 Homebrew 安装了两个版本的 Node 的话，是很正常的。安装成功以后，你应该看到以下结果 (当然，你本机的用户名和日期等信息可能会稍有不同)：

```
$> which node
/usr/local/bin/node
$> node -v
v8.6.0
```

我使用的是 Node 8.6.0 版本。不过，如果我查询 Homebrew 的目录 (默认是 `/usr/bin/Cellar`) 的话，就会发现这是一个符号链接 (链接到实际的安装目录)：

```
$>ls -l /usr/local/bin/node
lrwxr-xr-x 1 jon admin 29 27 Sep 15:14 /usr/local/bin/node ->
../Cellar/node/8.6.0/bin/node
```

进一步深入发掘，就能找到其他被当前版本所替代的 Node 版本：

```
$>ls -l /usr/local/Cellar/node
total 0
drwxr-xr-x 14 jon admin 476 11 May 14:14 7.10.0
drwxr-xr-x 14 jon admin 476 25 Apr 13:41 7.9.0
drwxr-xr-x 14 jon admin 448 27 Sep 15:14 8.6.0
```

在你的计算机上结果可能会有些不同。不过，重要的是，你一定要安装较新版本的 Node。

## NPM

NPM 包含了两样东西：一个基于命令行的包管理器和一个可以随手获得的开源项目（包）集合。

NPM 上的 `react-native` 包囊括了针对不同平台的 JavaScript ES6 模块。比如，针对 iOS 平台的 `<Text />` 组件的实现位于 `RCTText.m` 中，而其针对 Android 的实现位于 `ReactTextView.java` 中。



### 使用 Yarn

在 Yarn 占领一席之地之前，React Native 是通过 NPM 来设置和构建的。作为比 NPM 更快的替代品，Yarn 同样依赖 NPM 仓库。`yarn.lock` 文件用于确保项目能够正确管理所有的依赖。Yarn 首先检查 `yarn.lock` 文件，然后寻找 `package.json` 文件，以确保从 NPM 到 Yarn 的无缝迁移。

NPM 包可以安装到系统全局中，或者安装在指定项目的 `node_modules` 文件夹下。React Native 最好是全局安装，而项目相关的依赖才应该安装到项目的本地文件夹下。这样做可以保证你在终端的任意位置都能运行 React Native 的命令行工具 `react-native-cli`。如果你需要特定版本的 React Native，可以把它作为项目的本地依赖。

## 检查 NPM 是否安装正确

```
$> which npm
/usr/local/bin/npm
```



终端应该告诉你对应的安装路径。检查 NPM 的版本：

```
$> npm -v  
4.2.0
```

## 安装 React Native 命令行工具

```
$> npm install -g react-native-cli
```

## Xcode (针对 iOS 平台)

Xcode 是用来构建以及运行 Mac OS 和 iOS 平台应用的苹果官方开发环境。构建用 Objective-C 和 Swift 编写的 React Native 组件时必须用到 Xcode (只适用于 Mac OS)。

Xcode 也提供了一系列必要的命令行工具，可以通过命令行来构建代码，以及绑定 Node.js 中的 Mac OS 类库。



### 运行 Xcode Beta

鉴于 iOS 会经常更新，你可能需要在自己的开发机器上安装 Xcode 的 beta 版。安装多个版本的 Xcode 也导致要安装多个版本的 iOS 模拟器。在这种情况下，最好的做法是通过 Xcode 而非命令行来运行模拟器。

## JDK

Android 和 Java 是分不开的，它们就像糖和黄油，一起使用才能使食物更美味。针对 Android 平台的 React Native 也是如此。使用 JavaScript 编写的 React 组件最终还是要运行在 Android 的 Java 虚拟机中。如果你要在本机运行 Android 的话，就必须安装 Java 开发环境 (JDK)。

请从 Oracle 官方网站下载 JDK (最低版本为 8，参见 <http://bit.ly/1X9h0Ea>)。

## Android Studio

Android Studio 是构建和部署 Android 应用的官方开发环境，可以免费下载 (<http://bit.ly/2Dbnmeu>)。安装完成之后，就会多出一个包管理器。不过值得庆幸的是，*React Native Getting Started Guide* 里有详细的步骤教你如何设置 (<http://bit.ly/2GWQDfo>)。