

O'REILLY®

TURING

图灵程序设计丛书



# React Native

## 开发指南

Learning React Native

从基础开始逐步深入，利用React Native成功  
部署可100%代码复用的跨平台应用。

[美] Bonnie Eisenman 著  
黄为伟 译



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

# React Native开发指南

Learning React Native  
Building Native Mobile Apps with JavaScript

[美] Bonnie Eisenman 著

黄为伟 译

O'REILLY®

*Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo*

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社  
北京

## 图书在版编目 (CIP) 数据

React Native开发指南 / (美) 艾森曼  
(Bonnie Eisenman) 著 ; 黄为伟译. -- 北京 : 人民邮  
电出版社, 2016.6  
(图灵程序设计丛书)  
ISBN 978-7-115-42526-3

I. ①R… II. ①艾… ②黄… III. ①移动终端—应用  
程序—程序设计—指南 IV. ①TN929.53-62

中国版本图书馆CIP数据核字(2016)第118178号

## 内 容 提 要

本书通过丰富的示例和详细的讲解,介绍了React Native这款JavaScript框架。在React Native中利用现有的JavaScript和React知识,就可以开发和部署功能完备的、真正原生的移动应用,并同时支持iOS与Android平台。除了框架本身的概念讲解之外,本书还讨论了如何使用第三方库,以及如何编写自己的Java或Objective-C的React Native扩展。

本书适合前端工程师或Web开发者,以及希望开发跨平台移动应用的其他开发人员。

- 
- ◆ 著 [美] Bonnie Eisenman  
译 黄为伟  
责任编辑 朱 巍  
执行编辑 温 雪  
责任印制 彭志环
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
三河市中晟雅豪印务有限公司印刷
  - ◆ 开本: 800×1000 1/16  
印张: 14.75  
字数: 349千字 2016年6月第1版  
印数: 1-3 500册 2016年6月河北第1次印刷  
著作权合同登记号 图字: 01-2016-3664号

---

定价: 59.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第 8052 号

---

# 版权声明

© 2016 by Bonnie Eisenman.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2016. Authorized translation of the English edition, 2016 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.

英文原版由 O'Reilly Media, Inc. 出版，2016。

简体中文版由人民邮电出版社出版，2016。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

# 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，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

---

# 前言

本书将介绍 React Native，一款由 Facebook 公司出品的用来构建移动应用的 JavaScript 框架。在 React Native 中利用现有的 JavaScript 和 React 知识，就可以开发和部署功能齐全的、真正原生的移动应用，并同时支持 iOS 与 Android 平台。采用 JavaScript 作为开发语言并不意味着需要退而求其次，相反，React Native 在不牺牲原生样式和体验的前提下，相比传统移动开发仍然有很多优势。

我们将从基础开始学习，然后逐步深入，最终部署一款 100% 代码复用的成熟的移动应用到 iOS 应用商店和 Google Play 商店。除了框架本身的概念讲解之外，我们还将讨论如何使用第三方库，以及如何编写自己的 Java 或 Objective-C 的 React Native 扩展。

如果你想从前端工程师或 Web 开发者的视角接触移动应用开发，那么本书就是为你量身定做的。React Native 是一款令人惊奇的框架，愿你怀着和我一样喜悦的心情来探索它。

## 预备知识

本书总体上不是介绍 React 的，我们假设你对 React 已经有一些了解。如果你从未接触过 React，我建议你在正式开始学习移动开发之前先阅读一两篇相关的教程，尤其应该熟悉 React 的属性 (props) 和状态 (state)、组件的生命周期以及如何创建 React 组件等知识。

同时，我们也会使用一些 ES6 和 JSX 的语法。如果你对这些还不太熟悉也没有关系，我们将在第 2 章讲解 JSX，在附录 A 中介绍 ES6 的语法。这些语法本质上与你习惯的 JavaScript 代码是一一对一的解析关系。

本书假设你使用 OS X 操作系统进行开发。开发 iOS 应用必须使用 OS X 操作系统。使用 Linux 和 Windows 开发 Android 应用的支持工作仍在进行中，你可以在官网 (<http://facebook.github.io/react-native/docs/linux-windows-support.html>) 阅读更多关于 Linux 与 Windows 支持的相关内容。

# 排版约定

本书使用了下列排版约定。

- 楷体  
表示新术语。
- 等宽字体 (*Constant width*)  
表示程序片段，以及正文中出现的变量、函数名、数据库、数据类型、环境变量、语句和关键字等。
- 加粗等宽字体 (*Constant width bold*)  
表示应该由用户输入的命令或其他文本。
- 等宽斜体 (*Constant width italic*)  
表示应该由用户输入的值或根据上下文确定的值替换的文本。



该图标表示提示或建议。



该图标表示一般注记。



该图标表示警告或警示。

## 使用代码示例

补充材料（代码示例、练习等）可以从 <https://github.com/bonniee/learning-react-native> 下载。

本书是要帮你完成工作的。一般来说，如果本书提供了示例代码，你可以把它用在你的程序或文档中。除非你使用了很大一部分代码，否则无需联系我们获得许可。比如，用本书的几个代码片段写一个程序就无需获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用本书中的示例代码回答问题无需获得许可，将书中大量的代码放到你的产

品文档中则需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明一般包括书名、作者、出版社和 ISBN。比如：“*Learning React Native* by Bonnie Eisenman (O’Reilly). Copyright 2016 Bonnie Eisenman, 978-1-491-92900-1”。

如果你觉得自己对示例代码的用法超出了上述许可的范围，欢迎你通过 [permissions@oreilly.com](mailto:permissions@oreilly.com) 与我们联系。

## Safari® Books Online



Safari Books Online (<http://www.safaribooksonline.com>) 是应运而生的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。技术专家、软件开发人员、Web 设计师、商务人士和创意专家等，在开展调研、解决问题、学习和认证培训时，都将 Safari Books Online 视作获取资料的首选渠道。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O’Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、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、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

## 联系我们

请把对本书的评价和问题发给出版社。

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

O’Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例



代码以及其他信息。本书的网站地址是：

<http://shop.oreilly.com/product/0636920041511.do>

对于本书的评论和技术性问题，请发送电子邮件到：[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

<http://www.oreilly.com>

我们在 Facebook 的地址如下：<http://facebook.com/oreilly>

请关注我们的 Twitter 动态：<http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下：<http://www.youtube.com/oreillymedia>

## 资源

单枪匹马会让学习过程变得困难；虽然事实并不一定如此，但你不一定要这样做。这里有一些资源，也许在阅读过程中会给你带来一些帮助。

- 本书中所有的代码示例都在 GitHub 代码仓库中 (<https://github.com/bonniec/learning-react-native>)，如果在学习过程中遇到困难或者需要代码的上下文，不妨看看这里。
- 加入 LearningReactNative.com (<http://learningreactnative.com>) 的邮件列表获取后续的文章、建议和实用的资源。
- 官方文档 (<https://facebook.github.io/react-native/>) 中有大量优秀的参考资料。

此外，React Native 社区也是实用的资源：

- Brent Vatne 的 React Native newsletter (<http://brentvatne.ca/react-native-newsletter/>)
- Stack Overflow 上的 react-native 标签分类 (<http://stackoverflow.com/questions/tagged/react-native>)
- Freenode 上的 #reactnative 小组 (<irc://chat.freenode.net/reactnative>)

## 致谢

写成本书离不开大家的帮助和支持。首先要感谢我的编辑 Meg Foley 以及 O'Reilly 团队中的其他成员。同时也要感谢技术审稿人 David Bieber、Jason Brown、Erica Portnoy 和 Jonathan Stark，他们花费了大量时间审阅本书并提出了深有见地的反馈意见。感谢 React Native 团队，如果没有他们杰出的产品，自然也不会有本书的诞生。感谢 Zachary Elliot 对 Zebreto 应用和 Android 开发提供的帮助。

最后，非常感谢我亲爱的朋友们，在本书写作过程中给了我无限的包容、莫大的精神支持和悉心的指导，并在需要的时候陪我消遣。谢谢你们！

# 目录

前言	xi
第 1 章 初识 React Native	1
1.1 React Native 的优点	2
1.1.1 开发者体验	2
1.1.2 代码复用与知识共享	3
1.2 风险和缺点	4
1.3 小结	4
第 2 章 React Native 工作原理	5
2.1 React Native 是如何工作的	5
2.2 渲染周期	7
2.3 在 React Native 中创建组件	7
2.3.1 编写视图	7
2.3.2 使用 JSX	9
2.3.3 原生组件的样式	10
2.4 宿主平台接口	11
2.5 小结	12
第 3 章 构建你的第一个应用	13
3.1 搭建环境	13
3.1.1 安装 React Native	14
3.1.2 iOS 依赖	14
3.1.3 Android 依赖	14
3.2 创建一个新的应用	17

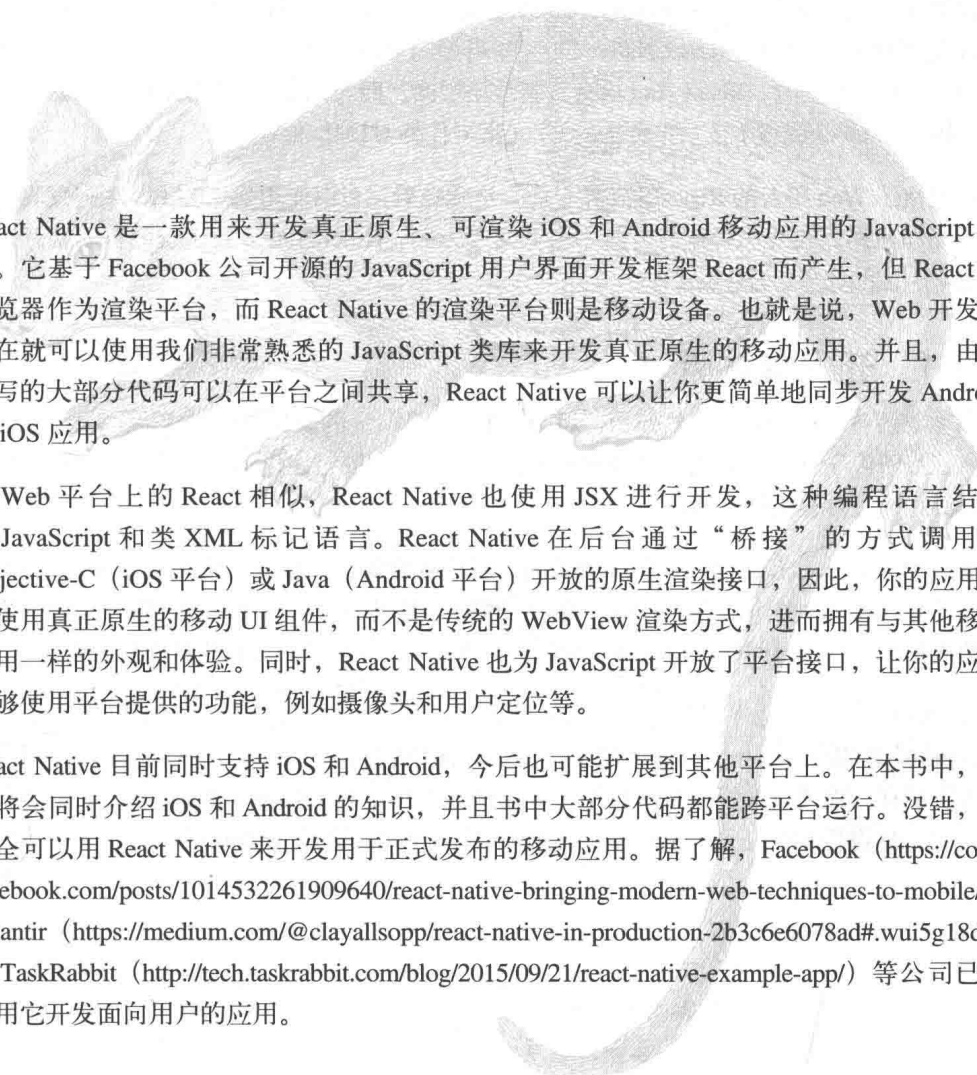
3.2.1	在 iOS 平台运行 React Native 应用 .....	18
3.2.2	部署到 iOS 设备 .....	20
3.2.3	在 Android 平台运行 React Native 应用 .....	23
3.2.4	小结：创建并运行项目 .....	24
3.3	探索示例代码 .....	24
3.3.1	添加组件到视图中 .....	24
3.3.2	React Native 中的模块导入 .....	25
3.3.3	FirstProject 组件 .....	26
3.4	开发天气应用 .....	27
3.4.1	处理用户输入 .....	28
3.4.2	展现数据 .....	30
3.4.3	添加背景图片 .....	34
3.4.4	从 Web 获取数据 .....	36
3.4.5	整合 .....	37
3.5	小结 .....	40
<b>第 4 章</b>	<b>移动应用组件</b> .....	<b>42</b>
4.1	类比 HTML 元素与原生组件 .....	42
4.1.1	文本组件 .....	43
4.1.2	图片组件 .....	45
4.2	处理触摸和手势 .....	46
4.2.1	使用 TouchableHighlight .....	47
4.2.2	GestureResponder 系统 .....	49
4.2.3	PanResponder .....	52
4.3	使用结构化组件 .....	58
4.3.1	使用 ListView .....	58
4.3.2	使用 Navigator .....	66
4.3.3	其他结构化组件 .....	68
4.4	平台特定组件 .....	69
4.4.1	iOS 或 Android 特定组件 .....	69
4.4.2	平台特定版本的组件 .....	70
4.4.3	何时使用平台特定组件 .....	74
4.5	小结 .....	74
<b>第 5 章</b>	<b>样式</b> .....	<b>75</b>
5.1	声明和操作样式 .....	75
5.1.1	内联样式 .....	76
5.1.2	对象样式 .....	76
5.1.3	使用 Stylesheet.Create .....	77
5.1.4	样式拼接 .....	77

5.2	组织和继承	79
5.2.1	导出样式对象	79
5.2.2	样式作为属性传递	80
5.2.3	复用和共享样式	81
5.3	定位和设计布局	81
5.3.1	使用 flexbox 布局	82
5.3.2	使用绝对定位	86
5.3.3	学以致用	86
5.4	小结	91
<b>第 6 章 平台接口</b>		<b>92</b>
6.1	使用定位接口	93
6.1.1	获取用户地理位置	93
6.1.2	处理权限问题	94
6.1.3	在 iOS 模拟器上测试定位	95
6.1.4	监听用户位置	96
6.1.5	限制	96
6.1.6	改进天气应用	96
6.2	使用用户图片与摄像头	99
6.2.1	相机模块	99
6.2.2	通过 getPhotoParams 获取图片	101
6.2.3	从相机渲染一张图片	101
6.2.4	展示照片列表	103
6.2.5	上传图片至服务器	107
6.3	AsyncStore 持久化数据存储	108
6.4	智能天气应用	109
6.4.1	WeatherProject 组件	111
6.4.2	Forecast 组件	114
6.4.3	Button 组件	115
6.4.4	LocationButton 组件	116
6.4.5	PhotoBackdrop 组件	117
6.5	小结	119
<b>第 7 章 模块</b>		<b>120</b>
7.1	使用 npm 安装 JavaScript 类库	120
7.2	iOS 原生模块	121
7.2.1	导入第三方组件	122
7.2.2	使用视频组件	125
7.2.3	剖析 Objective-C 原生模块	125
7.2.4	RCTVideo 的实现	128

7.3	Android 原生模块	130
7.3.1	安装第三方组件	130
7.3.2	剖析 Java 原生模块	134
7.3.3	LinearGradient 的 Android 实现	137
7.4	跨平台原生模块	139
7.5	小结	141
<b>第 8 章</b>	<b>调试与开发者工具</b>	<b>142</b>
8.1	JavaScript 调试实践和解释	142
8.1.1	激活开发者选项	142
8.1.2	使用 console.log 调试	143
8.1.3	使用 JavaScript 调试器	145
8.1.4	使用 React 开发者工具	146
8.2	React Native 调试工具	147
8.2.1	使用审查元素功能	147
8.2.2	宕机红屏	148
8.3	JavaScript 之外的调试方法	152
8.3.1	常见的开发环境问题	153
8.3.2	常见的 Xcode 问题	153
8.3.3	常见的 Android 问题	154
8.3.4	React Native 包管理器	155
8.3.5	部署至 iOS 设备的问题	156
8.3.6	模拟器行为	157
8.4	测试代码	158
8.4.1	使用 Flow 进行类型检查	158
8.4.2	使用 Jest 进行测试	158
8.5	当你陷入困境	160
8.6	小结	160
<b>第 9 章</b>	<b>学以致用</b>	<b>161</b>
9.1	闪卡应用	161
9.1.1	项目结构	163
9.1.2	组件层次结构	164
9.2	模型与数据存储	168
9.2.1	数据流架构: Reflux 与 Flux	170
9.2.2	在 Zebreto 中使用 Reflux	173
9.2.3	AsyncStorage 与 Reflux Store 的持久化	175
9.3	使用 Navigator	177
9.4	探索第三方依赖	180
9.5	响应式设计 with 字体尺寸	180

9.6 小结及任务	183
<b>第 10 章 部署至 iOS 应用商店</b>	<b>184</b>
10.1 准备 Xcode 工程	184
10.1.1 选择支持的设备和目标 iOS 版本	185
10.1.2 启动界面图像	186
10.1.3 添加应用图标	188
10.1.4 设置 Bundle 名称	190
10.1.5 更新 AppDelegate.m	190
10.1.6 为发布设置 Schema	191
10.2 上传应用	192
10.2.1 完成协议文书	192
10.2.2 创建归档	193
10.2.3 在 iTunes Connect 上创建应用	196
10.3 使用 TestFlight 进行 Beta 测试	199
10.4 提交应用审核	200
10.5 小结	201
<b>第 11 章 部署 Android 应用</b>	<b>203</b>
11.1 设置应用图标	203
11.2 生成 release 版本的 APK	205
11.3 通过邮件或链接发布	207
11.4 提交应用至 Play 商店	207
11.4.1 通过 Play Store 进行 Beta 测试	209
11.4.2 Play 商店列表	210
11.4.3 商店列表所需的资源	211
11.4.4 发布应用	212
11.5 小结	214
总结	215
附录 A ES6 语法	216
附录 B 命令与快速入门指南	219
作者简介	221
关于封面	221

# 初识 React Native



React Native 是一款用来开发真正原生、可渲染 iOS 和 Android 移动应用的 JavaScript 框架。它基于 Facebook 公司开源的 JavaScript 用户界面开发框架 React 而产生，但 React 将浏览器作为渲染平台，而 React Native 的渲染平台则是移动设备。也就是说，Web 开发者现在就可以使用我们非常熟悉的 JavaScript 类库来开发真正原生的移动应用。并且，由于编写的大部分代码可以在平台之间共享，React Native 可以让你更简单地同步开发 Android 和 iOS 应用。

与 Web 平台上的 React 相似，React Native 也使用 JSX 进行开发，这种编程语言结合了 JavaScript 和类 XML 标记语言。React Native 在后台通过“桥接”的方式调用由 Objective-C (iOS 平台) 或 Java (Android 平台) 开放的原生渲染接口，因此，你的应用将会使用真正原生的移动 UI 组件，而不是传统的 WebView 渲染方式，进而拥有与其他移动应用一样的外观和体验。同时，React Native 也为 JavaScript 开放了平台接口，让你的应用能够使用平台提供的功能，例如摄像头和用户定位等。

React Native 目前同时支持 iOS 和 Android，今后也可能扩展到其他平台上。在本书中，我们将会同时介绍 iOS 和 Android 的知识，并且书中大部分代码都能跨平台运行。没错，你完全可以用 React Native 来开发用于正式发布的移动应用。据了解，Facebook (<https://code.facebook.com/posts/1014532261909640/react-native-bringing-modern-web-techniques-to-mobile/>)、Palantir (<https://medium.com/@clayallsopp/react-native-in-production-2b3c6e6078ad#.wui5g18dx>) 和 TaskRabbit (<http://tech.taskrabbit.com/blog/2015/09/21/react-native-example-app/>) 等公司已在使用它开发面向用户的应用。

## 1.1 React Native的优点

事实上，React Native 调用宿主平台标准渲染接口的方式已经使它从其他现有的跨平台应用开发方案（比如 Cordova 或 Ionic）中脱颖而出。目前通过编写 JavaScript、HTML 和 CSS 的方式进行应用开发的方案大多使用 WebView 进行界面渲染，当然这种方案是可行的，但也带来了一些问题，尤其是性能损耗。同时，这种方案通常无法使用宿主平台的原生 UI 组件，所以这些框架尝试去模仿原生 UI 组件的行为，而模仿的效果通常让人觉得不够真实。为了模仿各种类似动画这样的细节，一般都要付出巨大的努力，然而它们很快又会过时。

相反，React Native 则把你的代码解析成真正原生的 UI 组件，利用了所用平台上现有的视图渲染方式。并且，由于 React 不在 UI 主线程中运行，你的应用可以在不牺牲灵活性的前提下保持高性能。React Native 的生命周期与 React 相同，当属性（props）或状态（state）发生改变时，React Native 会重新渲染视图。而与浏览器上的 React 最大的不同在于，React Native 使用了宿主平台上的 UI 元素来代替 HTML 和 CSS。

对于习惯了 Web 平台的 React 开发者来说，这意味着你可以使用熟悉的工具来开发真正原生的移动应用。在开发者体验与跨平台开发等方面，React Native 较传统的移动端开发来说也有一定的优势。

### 1.1.1 开发者体验

如果你曾经有过移动端的开发经历，将会对 React Native 的易用性感到震惊。React Native 团队已经研发了强大的开发工具并在框架内嵌入了友好的错误提示，因此使用这些强大的工具会让开发体验更加自然。

例如，由于 React Native 使用了 JavaScript，我们查看修改结果时不需要重新编译。相反，按下 Command+R 就可以刷新应用，就和在网页上开发一样。在传统移动端开发中，编译构建应用所花费的时间会积少成多，相比之下 React Native 的快速迭代就像是天赐之福。

React Native 还可以让你更好地利用智能调试工具以及错误报告机制。如果你习惯于使用 Chrome 或者 Safari 的开发工具（图 1-1），那么使用它们进行移动开发一定也会让你十分愉悦。同样，你可以选择喜爱的任何文本编辑器来开发 JavaScript：React Native 不强制你使用 Xcode 进行 iOS 开发，也不强制使用 Android Studio 进行 Android 开发。



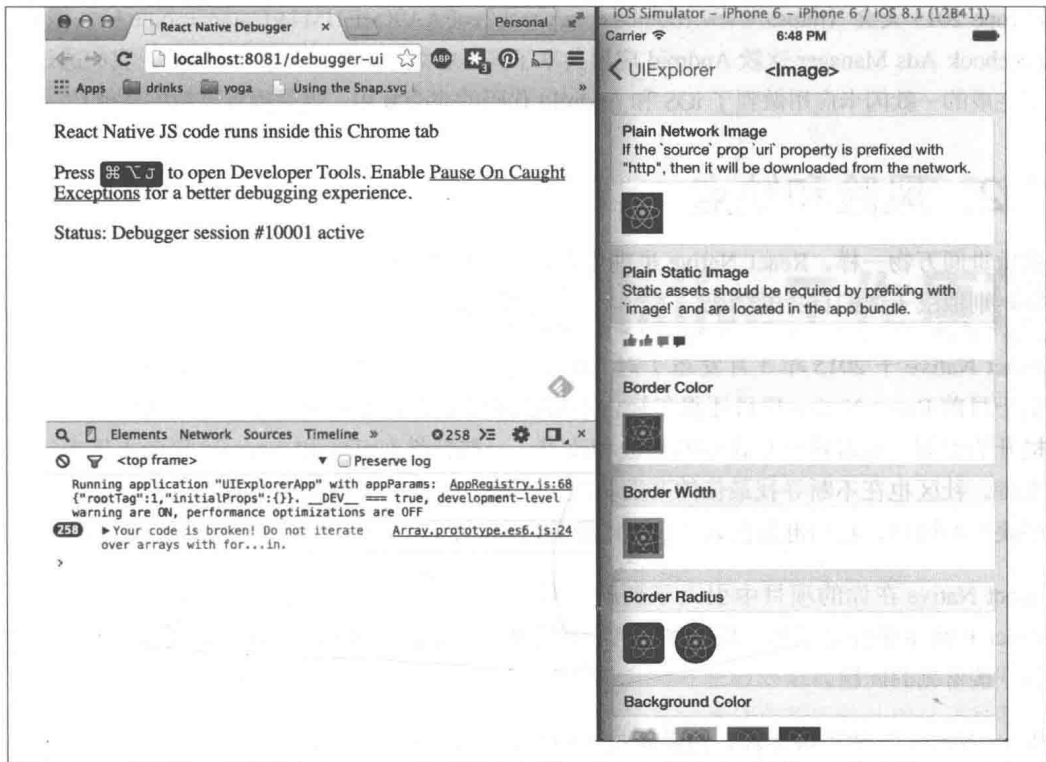


图 1-1: 使用 Chrome 调试器

除了能逐渐改善开发者体验之外，React Native 也极有可能给你的产品发布周期带来一些积极的影响。例如，Apple 公司允许通过网络对基于 JavaScript 开发的功能进行更新，无需额外的审核周期。

所有这些小福利将会节省你和你的伙伴们的时间和精力，让你可以专注于工作中那些更有趣的部分，同时也能提高你的工作效率。

## 1.1.2 代码复用与知识共享

使用 React Native 可以大大减少开发移动应用所需的资源。任何了解如何编写 React 的开发者现在都可以使用相同的技能同时开发 Web 应用、iOS 应用和 Android 应用。React Native 避免了按平台分工的必要，可以让你的团队更加快速地迭代产品，并更加高效地共享知识和资源。

除了知识的共享之外，你的大部分代码也可以被共享。当然，不是你写的所有代码都可以做到跨平台；这取决于你需要在特定的平台上实现什么功能，你可能偶尔也需要涉及 Objective-C 或 Java 的知识（好在这也不是很糟糕，我们将会在第 7 章讲解本地模块的用法）。使用 React Native，在不同平台之间复用代码将会变得出乎意料地简单。例如，React