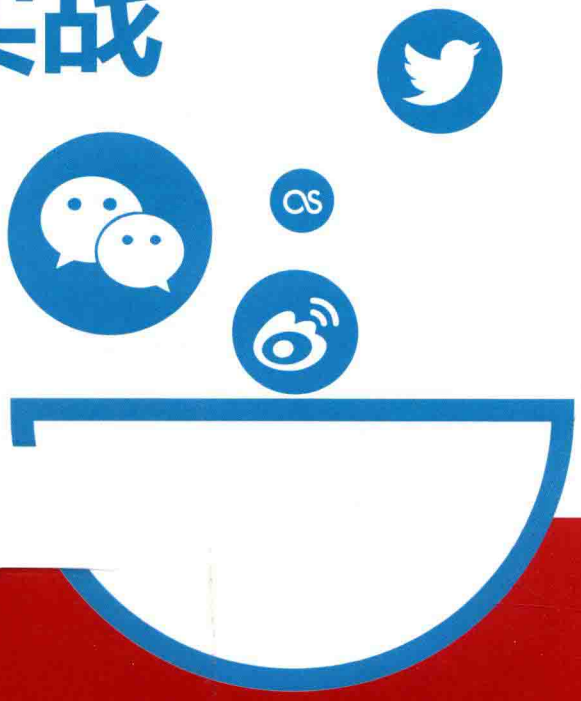
HZ BOOKS
华章IT

详解React Native应用从创建、开发到发布的全过程，展示各组件和API的用法

实战为王，通过典型项目案例，让读者快速掌握React Native应用开发

React Native
移动开发实战

袁林◎编著

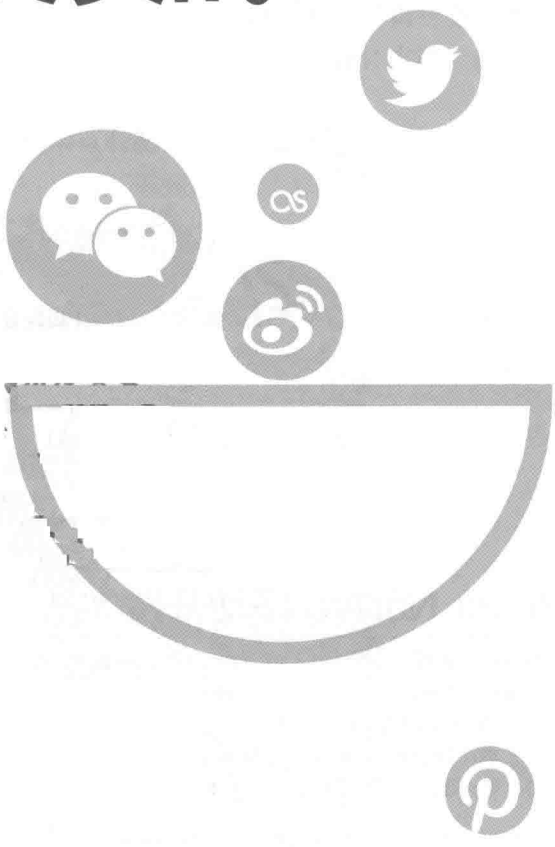


- 书中所有内容都配合详细的实例和源代码进行讲解
- 全面涵盖React Native组件、API、布局、第三方组件及原生接口开发等内容
- 详解React Native的开发工具、命令行工具及各种调试工具的使用
- 详细讲解一个电商App项目案例的开发过程，提高读者的实战开发水平
- 涉及软件开发流程、应用架构设计、代码重构，以及原生平台与跨平台开发等

机械工业出版社
China Machine Press

React Native
移动开发实战

袁林◎编著



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

React Native 移动开发实战/袁林编著. —北京：机械工业出版社，2017.6

ISBN 978-7-111-57179-7

Ⅰ. ①R… Ⅱ. ①袁… 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 146546 号

React Native 移动开发实战

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：欧振旭　　责任校对：姚志娟

印　　刷：中国电影出版社印刷厂　　版　　次：2017 年 7 月第 1 版第 1 次印刷

开　　本：185mm×260mm　1/16　　印　　张：20

书　　号：ISBN 978-7-111-57179-7　　定　　价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88379426　88361066　　投稿热线：（010）88379604

购书热线：（010）68326294　88379649　68995259　　读者信箱：hzit@hzbook.com

前言

随着手机和移动互联网技术的日益成熟，移动应用的领域也从如何开发，发展到如何更高效、更低成本地开发。传统的原生平台（iOS、Android）开发技术虽然比较成熟，但由于开发效率和成本的限制，已经越来越无法满足移动互联网应用的开发需求。

所以，具有简单、迅速、跨平台的优势，而且基于 Web 开发语言和布局技术的 React Native 得以迅速流行，并一举夺得跨平台开发技术的头筹。

目前市场上大多数 React Native 书籍主要以翻译和讲解官方文档为主，并未从开发实际应用出发，通过典型案例来指导读者提高开发水平。本书以实战为主旨，通过完整的电商类 App 项目实例，来介绍 React Native 中常用的组件、API、布局、第三方组件和原生接口，让读者全面、深入、透彻地理解 React Native 主流的开发和设计方法，提升实际开发水平和项目实战能力。

本书涉及的概念较多，下面给出一个技术点云图，希望读者有所了解。

本书的进阶顺序，也给出如下一个图，便于读者了解。

本书特色

1．每一步都有详细的源码和实例参考

为了便于读者理解本书内容，提高学习效率，本书的所有内容都有详细的源码和实例参考。对于这些源码和实例，作者均亲自编写和验证，杜绝复制、粘贴代码以敷衍读者的不负责任行为。本书源码可以在 https://coding.net/u/learnreactnative/p/learnreactnative-sourcecode/git 里下载。

2．内容涵盖React Native开发的各个方面

本书涵盖 React Native 组件、API、布局、第三方组件以及原生接口等 React Native 应用开发的各个方面，尽量保证不出现知识“死角”。凡是涉及的一些技术（如原生、瀑布流、耦合性和 JSON），也给出了概念或原理的解释。

3．结合工具助力更高效的React Native开发

在本书“实战”讲解的过程中，详细介绍了 React Native 开发工具 Nuclide 的使用、React Native 命令行工具的用法及各种调试工具（包括布局、断点及实时加载等）的使用，不仅教读者如何开发，还教读者如何更高效地开发。

4．项目案例典型，实战性强，有较高的应用价值

本书以开发一个电商类应用为例，涵盖了 React Native 应用开发中会用到的所有重点知识，设计和源码做到拿来可用，方便应用开发者随时查阅和参考。

5．收获的不仅仅是React Native平台和编码

对于一些学习能力较强的读者，完全可以在 React Native 开发文档的帮助下快速学习和掌握 React Native。而本书希望读者在掌握平台和编码之外，还能够了解实际应用开发过程中涉及的软件开发流程、应用架构设计、代码重构技巧，以及原生平台与其他跨平台开发的相关知识，让读者融会贯通地理解应用开发技术。

本书内容及知识体系

第1篇　React Native入门和基础（第1～2章）

本篇介绍了跨平台开发的主流方案和 React Native 基础知识，主要包括开发环境搭建、

React Native 命令行工具和 React Native 布局调试。

第2篇　React Native应用开发实战（第3～7章）

本篇介绍了 React Native 实际应用开发中常用的技术，主要包括基本组件、使用第三方组件、搭建基于 Node.js 的服务器为应用绑定真实数据、fetch API、AsyncStorage/SQLite/Realm 数据库存储、更多 React Native 组件和 API 的用法、原生平台接口开发等。

第3篇　React Native混合编程（第8~10章）

本篇主要总结和回顾了前 7 章所开发的电商类应用的技术和架构，主要包括应用的文件结构、Flexbox 的整体布局、应用的逻辑结构、应用的通信过程及进一步改进的地方和思路，其中就包括了 redux 开发框架。

第4篇　App的发布和更新（第11～12章）

本篇主要介绍了 React Native 应用打包和发布的全过程，配以详细的截图说明，并且对 React Native 应用发布后的热更新实现和方案 CodePush 做了详细的示例说明。

适合阅读本书的读者

- React Native 学习人员；
- iOS 平台应用开发工程师；
- Android 平台应用开发工程师；
- Web 前端开发工程师；
- Node.js 服务端开发工程师；
- 计算机相关专业的学生；
- 专业培训机构的学员；
- 软件开发项目经理。

本书作者

本书由袁林主笔编写。其他参与编写的人员有高旭、贺庆端、黎林、李伟浩、刘成智、刘利、刘源、谭建利、吴贵文、吴娟、夏秀英、肖太平、郑星。

致谢

感谢本书的编辑，让我有机会和本书结缘。感谢我的伙伴们：邵长磊、刘冬冬、袁方、袁满、翟绍虎、洪敏、郭晨光及张砚，与我一起探讨新技术并和 React Native 结缘。感谢我的妻子韩丽、女儿可可及我的父母，写作占用了我很多陪伴家人的时间和精力，正是有了家人的支持，才得以坚持下去。

最后还要感谢读者，本书的价值因你们而存在。

编者

目录

第 2 篇 React Native 应用开发实战

第 3 篇 React Native 混合编程

第 4 篇 App 的发布和更新

第 1 篇
React Native 入门和基础

第 1 章　为什么要学习 React Native

无论读者是移动平台开发者，还是 Web 前端开发者，想必对现在“大红大紫”的 React Native 都有所耳闻。那么，除了“乘着 Facebook 这棵大树好乘凉”的优势之外，React Native 到底是何方“神圣”，有什么令大家“趋之若鹜”的优点呢？下面带着这样的好奇，来随本书一探究竟吧！

本章主要内容有：

- React Native 与 React.js 的对比。
- 为什么说 React Native 是跨平台的。
- React Native 应用的结构。
- React Native 的特点。
- React Native 的环境搭建。
- 创建第一个 React Native 应用。

1.1　看透 React Native

React Native（http://facebook.github.io/react-native/）第一次进入公众的视野是在 2015 年 1 月的 React.js Conf（http://conf.reactjs.org/）上，随后，同年 5 月份，Facebook 在 F8 Conference（https://www.fbf8.com/）上正式宣布：React Native 项目（如图 1.1 所示）在 Github 开源。结果一天之内，就收获了 5000 多颗星，受关注程度可见一斑！

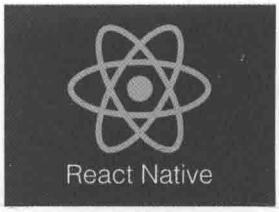


图 1.1　React Native Logo

小知识：React.js Conf 是指 Facebook 的 React 开发者大会，F8 Conference 是指 Facebook 的开发者大会，Github 是全球最大的软件项目托管平台，也被戏称为“人类的代码仓库”。

1.1.1　React Native 与 React.js

想必读者在还没弄清 React Native 之前，又发现了一个“新朋友”React.js（下文简称 React），那到底什么是 React 呢？它和 React Native 又是什么关系呢？

先来看看 Facebook 官方（https://code.facebook.com/projects/176988925806765/react/）对 React 的定义：

```
React is a JavaScript library for building user interfaces
```

从上述官方的定义可以知道：React 是一个用于前端 UI 开发的 JavaScript 库。和其他类似的前端框架相比，例如，老牌的 Backbone（http://backbonejs.org/）、Google 推出的 Angular（https://angularjs.org/）和以轻量级著称的 Vue.js（http://cn.vuejs.org/），React 最大的不同是提出了 Virtual DOM（即虚拟 DOM）的设计，可以大大提升页面渲染的效率。

小知识：移动平台开发很好理解，即移动平台上（例如 Apple 的 iOS 平台，Google 的 Android 平台）的软件开发，开发语言和技术主要有 Objective C、Swift 及 Java 等。而前端开发是相对于后端（又称服务器端）开发而言的，前端主要负责开发通过浏览器和用户交互的部分，开发语言和技术主要有 HTML、CSS 及 JavaScript 等。

但是，Facebook 不仅仅满足于 React 对前端开发技术的革新，又将 React 的设计移植到原生开发中，从而诞生了 React + Native 结合的产物，即 React Native。

虽然，React Native 刚开始只支持 iOS App 开发，但是从 2015 年 9 月起，React Native 也支持 Android App 开发，而且随着微软、三星等“IT 大佬”的加入，React Native 还将支持更多的移动平台，例如，Sansung 的 Tizen 平台（https://www.tizen.org/）、Microsoft 的 Window Phone（http://microsoft.github.io/code-push/articles/ReactNativeWindows.html）。

所以，简单来说：

- React Native 和 React 使用了相同的开发语言 JavaScript 和相同的设计理念 React。
- React Native 和 React 运行的环境和平台却是完全不同的，React Native 是基于移动平台（如 iOS、Android 等），而 React 是基于浏览器。

提示：国内网络环境下访问 React Native 官网（http://facebook.github.io/react-native/）可能较慢，读者可以访问国内的中文资源网站，例如 React Native 中文网（http://reactnative.cn/），或者自行搜索加快 React Native 官网访问速度的办法。

1.1.2　React Native 的跨平台

简单了解了 React Native 的由来之后，读者或许会有这样的疑问，开发移动平台 App 使用原生开发平台和语言就好了，为什么要出现使用 React Native 来开发移动平台 App 的技术呢？换句话说，React Native 到底可以解决什么问题呢？

在进一步讨论之前，笔者觉得有必要明确一下什么是原生应用和跨平台应用。

1．原生应用

所谓的原生应用是指：使用原生开发语言、工具和平台开发的应用。原生应用开发的优势在于拥有较高的平台成熟度，包括平台的稳定性、运行时的性能及完善的生态。

小知识：所谓的“生态”应该算是比较抽象的概念，开发平台的生态圈包含了很多方面，从硬件上芯片和各种电子元器件的生产、供应，到软件上所使用的语言、开发工具及第三方开源库的数量质量，以及人的方面，如开发者的数量、水平等因素。

但是，原生应用开发也不是没有任何缺点，那就是开发成本较高，导致开发效率相对较低。例如，当一个产品需要支持多种类型的移动终端时，就需要熟悉多个原生平台开发的工程师。

2. 跨平台应用

为了解决产品满足多个平台的需求，就有了所谓的跨平台应用开发。根据实现跨平台方案的不同，也就有了以下几种常见的跨平台解决方案。

- 混合应用开发：在移动浏览器中嵌入 HTML 页面来开发移动应用，代表的有 Apache Cordova（http://cordova.apache.org/），以及基于 Apache Cordova 衍生的 Inoic（http://ionicframework.com/）等，如图 1.2 所示。
- 跨平台的语言：例如，基于.NET 和 C#的 Xamarin（https://www.xamarin.com/），以及基于 Ruby 的 RubyMotion（http://www.rubymotion.com/cn/），如图 1.3 所示。

图 1.2　Apache Cordova LOGO

图 1.3　Xamarin LOGO

- React Native：使用的是 Web 开发语言（JavaScript）和环境（Node.js）。除了本书介绍的 React Native 之外，类似的技术方案还有 NativeScript（http://www.telerik.com/nativescript）、Weex（http://weex-project.io/）等，如图 1.4 所示。

图 1.4　Weex LOGO

提示：想要了解关于更多 React Native 的架构和原理，可以参考 1.1.3 节。

1.1.3　解剖 React Native 应用的结构

在了解完这么多关于 React Native 的故事和优势之后，让我们走近 React Native，来进一步了解 React Native 的原理和架构。

React Native 应用的整体结构如图 1.5 所示。

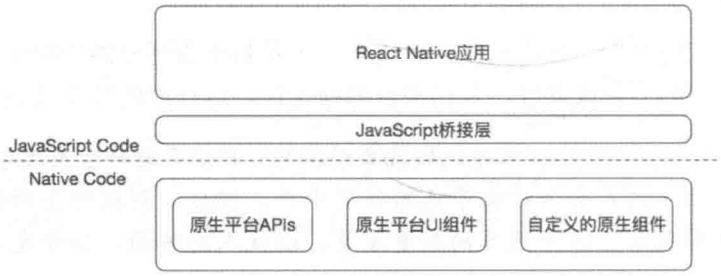


图 1.5　React Native 应用的整体结构

通过之前的介绍和图 1.5 可以看出：React Native 应用开发使用的是与 React 相同的开发语言 JavaScript 和设计思想 React，而底层仍然是基于原生平台的。这样，不同平台的适配就交由 React Native 平台去处理，而开发者只需要专注于 React Native 平台应用开发本身，体现出的优势如下。

- 应用层的开发变得简单、高效和跨平台。
- 应用稳定性、运行时的性能和原生平台接近。
- 在理解 React Native 原理之后，开发者也可以根据实际的产品需求开发自己的 React Native 组件，以复用已有原生平台的大量优秀组件。

1.2　React Native 的特点

那么，作为跨平台应用开发的“新贵”，React Native 相比其他跨平台技术到底有哪些优势呢？

1.2.1　其一：Learn Once, Write Anywhere

这句话是 React Native 官网（http://facebook.github.io/react-native/）对 React Native 的概述，简单明了地概括了 React Native 的最大特点和优点。

只需要学习 React Native 这一种开发方式（包括平台、语言和开发环境等）就可以开发多个不同平台的 App。

这句话简单来说就是 Learn Once, Write Anywhere，这也是 React Native 的宣传广告，如图 1.6 所示。

图 1.6　Learn Once, Write Anythere 宣传广告

小知识：除了 React Native 提出的 Learn Once，Write Anywhere 的口号，Java 语言也提出过类似的口号 Write Once，Run Anywhere，两者看起来类似，但其实是完全不同的。React Native 就像上面介绍的，降低的是学习成本，针对不同平台可能还需要单独开发；而 Java 语言的意思是只需要开发一次，就可以成功运行在不同的平台和设备上。

目前，React Native 对 iOS、Android 平台的支持已经非常好了，在不远的将来，应该还会支持 Windows、Tizen 等更多的移动平台。

而且，React Native 的大多数组件也是可以在多个平台复用的，所以学习了 React Native 开发之后，完全可以胜任多个平台的开发任务且不需要很高的额外学习成本，大大降低了