

资深架构师十多年经验结晶

包含React Native 框架底层原理，以及与iOS、Android混合开发案例，
精选了大量实例代码，方便读者实战演练



邱鹏源 编著

introduction to React Native: A Hands-On Approach

React Native

精解与实战



机械工业出版社
China Machine Press



内 容 简 介

本书由资深架构师撰写，包含React Native框架底层原理，以及与iOS、Android混合开发案例，精选了大量实例代码，方便读者快速学习。主要内容分为两大部分，第Ⅰ部分“入门”包括第1~9章，介绍React Native框架的基本原理与使用方法；第Ⅱ部分“进阶”包括第10~15章，介绍React Native框架的高阶开发与App部署相关知识。附录部分剖析了React Native的源码，可帮助读者研究React Native底层本质，还分享了一些React Native框架学习的相关资源。

本书适合各类移动App开发人员，深入学习React Native框架可以使你开发一套React Native源码同时部署到iOS平台与Android平台。

本书的配套源代码位于<https://github.com/ParryQiu/ReactNative-Book-Demo>。





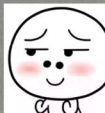
仅供非商业用途或交流学习使用



华章IT
HZBOOKS | Information Technology



仅供非商业用途或交流学习使用





introduction to React Native: A Hands-On Approach

React Native 精解与实战

邱鹏源 编著



机械工业出版社
China Machine Press



图书在版编目 (CIP) 数据

React Native 精解与实战 / 邱鹏源编著. —北京: 机械工业出版社, 2018.6
(实战)

ISBN 978-7-111-60385-6

I. R… II. 邱… III. 移动终端—应用程序—程序设计 IV. TN929.53

中国版本图书馆 CIP 数据核字 (2018) 第 146531 号

React Native 精解与实战

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: 陈佳媛

责任校对: 李秋荣

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2018 年 8 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 22

书 号: ISBN 978-7-111-60385-6

定 价: 79.00 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

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东



Preface 前言

从 2015 年 React Native 框架发布开始，我就关注 React Native 框架的发展，在得知 React Native 框架将同时支持 iOS 平台与 Android 平台的部署后，我们就开始着手将之前的项目从混合开发的方式慢慢转移到使用 React Native 框架开发的方式上来。当时国内的 React Native 方面的资料非常少，很多难题的解决都需要查阅大量的国外文档，正是在这个摸索的过程中，加深了对 React Native 框架的理解。

后来，我们的很多 Web 项目都在使用 React 框架，React 正是 React Native 最底层的技术框架，同时也深深体会到只有理解底层架构，才能对于很多表象的难题快速定位并找到解决方案。

本书基于我多年写的技术博客以及使用 React Native 框架的实战经验，我认为底层的原理永远比一些组件的使用方法更重要，所以本书介绍了大量框架底层原理。通过阅读本书你会发现掌握框架后，组件和 API 的使用将变得非常简单，希望你在学习时能体会到这种触类旁通的感觉。

本书主要内容

本书分为两大部分，第 I 部分“入门”包括第 1~9 章，介绍 React Native 框架的基本原理与使用；第 II 部分“进阶”包括第 10~15 章，介绍 React Native 框架的高阶开发与 App 部署相关知识。

第 1 章介绍 React 与 React Native 框架产生的背景与原理，以及开发优势。

第 2 章介绍 Node.js 框架，并实战演示了 React Native 开发环境的安装与配置。



IV

第 3 章介绍 React Native 框架的构成、工作原理、组件间通信以及生命周期，包括代码实战演示。

第 4 章介绍 React Native 页面布局开发使用的 CSS Flex，帮助读者掌握好 React Native 框架中元素布局的基本方法。

第 5 章介绍 React Native 框架下 iOS 平台与 Android 平台环境配置与代码调试的方法，并对 React Native 框架的调试工具以及借助 Chrome 进行远程调试的方法进行了实战讲解。

第 6 章介绍 React Native 框架中常用的组件，如 View、TabBar、NavigatorIOS、Image、Text、TextInput、WebView、ScrollView 等。同时介绍了 iOS 平台与 Android 平台的适配以及更适合的第三方组件。

第 7 章介绍 React Native 框架重点 API 的使用，包括提示框、App 运行状态、异步存储、相机与相册、地理位置信息、设备网络信息等 API，希望读者熟练掌握这些基础 API，进而能举一反三。

第 8 章介绍 React Native 框架下的网络请求以及列表数据的绑定，这是开发 App 需要使用的技术重点。

第 9 章介绍 React Native 开发生态下一些常用的第三方组件，通过代码实战的方式进行讲解，并介绍了如何快速地找到自己的项目需要使用的第三方组件。

第 10 章结合 iOS 与 Android 平台深入讲解了 React Native 框架的底层运行原理，并分别介绍了两个平台的部署与测试方法。

第 11 章介绍 React Native 框架下 iOS 平台的混合开发方法，通过混合开发，你可以在 React Native 框架下访问任何 iOS 原生平台。

第 12 章介绍 React Native 框架下 Android 平台的混合开发方法，以及案例分析。

第 13 章详细讲解 React Native 框架下 iOS 平台与 Android 平台的消息推送原理，并介绍两个平台的消息推送实战。

第 14 章介绍项目最终打包前 App 的图标与启动图的设置，并介绍了如何通过第三方工具快速生成这些相关资源。

第 15 章介绍 React Native 性能调优的方法与技巧，以便在 App 上架前测试以及后期 App 遇到性能问题时可以快速定位到问题所在。

本书附录简单剖析了 React Native 的源码，希望能帮助你深入研究 React Native 框架的源码，以便能探究其底层本质。同时也分享了一些学习 React Native 框架的相关资源。



本书的读者对象

- 各类移动 App 开发人员，学习 React Native 框架可以使你开发一套 React Native 源码同时部署到 iOS 平台与 Android 平台。
- 想进入移动 App 开发领域的初学者，React Native 框架比 Android 和 iOS 两个原生平台的技术门槛低很多，只需掌握 HTML、CSS、JavaScript 相关知识就可以动手开发跨平台的移动 App。
- 已经在使用 React Native 框架开发移动 App 的开发人员，书中讲解了 React Native 框架的底层原理，以及与 iOS 平台、Android 平台的高阶混合开发部分，完全用代码进行讲解，学习起来更加直观。

本书配套源代码

本书的配套源代码都可以在 <https://github.com/ParryQiu/ReactNative-Book-Demo> 下载。

后续如果遇到 React Native 框架的大升级，我同样会在此代码库中相应地更新实战演示的代码。

致谢

首先感谢多年来共事过的同事们、领导们，多年来容忍我在一些技术问题上吹毛求疵，给了我很多好的建议和思路启发。还要感谢吴怡编辑，从约稿到审稿，都体现了她的认真态度和专业知识，她能一针见血地指出问题所在，促使我更加认真、专业地去编写此书。

在编写的过程中，虽然对于每一个知识点我都查阅了大量的相关文档，但是本书涉及海量的知识点，难免会有疏漏，恳请各位读者斧正。

邱鹏源

2018 年 4 月



目 录 Contents

前言

第 I 部分 入门

第 1 章 React 与 React Native

简介 2

1.1 React 简介 2

1.2 React Native 简介 9

1.3 React Native 前置知识点 11

第 2 章 Node.js 简介与开发环境

配置 13

2.1 Node.js 与 npm 简介 13

2.2 React Native 开发环境配置 15

2.2.1 安装 Node.js 16

2.2.2 安装 React Native 18

2.2.3 代码编辑器以及推荐插件 22

第 3 章 React Native 工作原理

与生命周期 25

3.1 React Native 框架及工作原理 25

3.1.1 React Native 与原生平台

通信 27

3.1.2 组件间通信 27

3.2 React Native 中的生命周期 31

3.3 本章小结 35

第 4 章 React Native 页面布局 36

4.1 CSS 3 简介 36

4.2 Flex 弹性盒模型 37

4.3 Flex 属性详解与实例 39

4.3.1 justify-content 属性 40

4.3.2 align-items 属性 42

4.3.3 align-self 属性 45

4.3.4 flex-direction 属性 48

4.3.5 flex-basis 属性 51

4.3.6 flex-wrap 属性 52

4.3.7 align-content 属性 54

4.3.8 flex-grow 属性 58

4.3.9 flex-shrink 属性 60

4.3.10 order 属性 63

4.3.11 flex-flow 属性 64

4.3.12 flex 属性 64



4.4	React Native 中的 Flex 属性	65	6.5	Image 组件	111
4.5	本章小结	65	6.5.1	Image 组件介绍	111
第 5 章 React Native 开发调试技巧与工具			6.5.2	Image 组件实例	112
5.1	配置 iOS 开发环境	66	6.6	Text 组件	114
5.2	配置 Android 开发环境	70	6.6.1	Text 组件介绍	114
5.3	常用调试属性的说明	73	6.6.2	Text 组件基本使用	116
5.4	Chrome 中远程调试代码	77	6.6.3	Text 组件嵌套	117
5.5	React Developer Tools 工具 安装与应用	80	6.6.4	Text 组件样式统一	119
5.6	本章小结	82	6.7	TextInput 组件	122
第 6 章 React Native 组件详解			6.7.1	TextInput 组件介绍	122
6.1	React Native 组件简介	83	6.7.2	TextInput 组件实例	124
6.2	视图组件	86	6.8	触摸处理类组件	130
6.2.1	View 组件介绍	86	6.8.1	TouchableHighlight 组件 介绍	130
6.2.2	View 组件实例	87	6.8.2	TouchableHighlight 组件 实例	131
6.3	底部导航 TabBar 组件	92	6.8.3	TouchableNativeFeedback 组件介绍	132
6.3.1	TabBar 组件介绍	92	6.8.4	TouchableNativeFeedback 组件实例	133
6.3.2	iOS 平台下 TabBarIOS 组件 实例	94	6.8.5	TouchableOpacity 组件介绍	133
6.3.3	Android 平台下 TabBar 组件 实例	99	6.8.6	TouchableOpacity 组件实例	134
6.4	iOS 与 Android 的页面跳转	102	6.8.7	TouchableWithoutFeedback 组件介绍	134
6.4.1	NavigatorIOS 组件介绍	102	6.9	Web View 组件	135
6.4.2	NavigatorIOS 组件实例	104	6.9.1	WebView 组件介绍	135
6.4.3	react-native-navigation 组件 介绍	108	6.9.2	WebView 组件实例	137
6.4.4	react-native-navigation 组件 实例	110	6.10	ScrollView 组件	139
			6.10.1	ScrollView 组件介绍	139
			6.10.2	ScrollView 组件实例	140
			6.11	本章小结	142

第7章 React Native API 详解 ··· 143

- 7.1 React Native API 简介 ··· 143
- 7.2 提示框 ··· 145
 - 7.2.1 Alert 介绍 ··· 145
 - 7.2.2 Alert 实例 ··· 145
- 7.3 App 运行状态 ··· 150
 - 7.3.1 AppState 介绍 ··· 150
 - 7.3.2 AppState 实例 ··· 150
- 7.4 异步存储 ··· 152
 - 7.4.1 AsyncStorage 介绍 ··· 152
 - 7.4.2 AsyncStorage 实例 ··· 154
 - 7.4.3 登录状态处理 ··· 159
- 7.5 相机与相册 API ··· 161
 - 7.5.1 CameraRoll 介绍 ··· 161
 - 7.5.2 相册 / 相机组件实例 ··· 161
- 7.6 地理位置信息 ··· 168
 - 7.6.1 Geolocation 介绍 ··· 168
 - 7.6.2 Geolocation 实例 ··· 169
- 7.7 设备网络信息 ··· 175
 - 7.7.1 NetInfo 介绍 ··· 175
 - 7.7.2 NetInfo 实例 ··· 175
- 7.8 本章小结 ··· 178

第8章 React Native 网络请求 详解 ··· 179

- 8.1 RESTful API 简介 ··· 179
- 8.2 React Native 中的网络请求 ··· 180
- 8.3 ListView 组件 ··· 184
- 8.4 React Native 网络请求与列表
绑定方案 ··· 186

8.5 本章小结 ··· 192

第9章 常用 React Native 开源 组件详解 ··· 193

- 9.1 React Native 热门资源列表 ··· 194
- 9.2 React Native 接入微博、微信、
QQ 登录 ··· 196
- 9.3 更加美观的组件库 ··· 200
- 9.4 React Native 图表 ··· 202
- 9.5 react-native-gifted-listview ··· 205
- 9.6 react-native-vector-icons ··· 207
- 9.7 本章小结 ··· 210

第II部分 进阶

第10章 React Native 运行原理 与部署调试 ··· 214

- 10.1 React Native 运行原理 ··· 214
- 10.2 iOS 平台部署与调试 ··· 220
- 10.3 Android 平台部署与调试 ··· 225
- 10.4 Android 模拟器简介 ··· 230
- 10.5 本章小结 ··· 233

第11章 iOS 平台与 React Native 混合开发 ··· 234

- 11.1 iOS 平台混合开发简介 ··· 234
- 11.2 iOS 平台混合开发原理详解 ··· 235
 - 11.2.1 iOS 原生代码实现 ··· 235
 - 11.2.2 iOS 项目编译设置 ··· 237
 - 11.2.3 React Native 中调用混合
开发代码 ··· 239

11.2.4	iOS 平台混合开发特性 详解	241	13.2	Android 平台消息推送机制	282
11.3	iOS 平台混合开发实例	249	13.3	React Native 极光推送实战	284
11.3.1	iOS 原生代码实现	249	13.3.1	极光推送平台设置	284
11.3.2	React Native 调用混合 开发代码	250	13.3.2	React Native 插件安装与 配置	286
11.4	本章小结	252	13.3.3	理解标签、别名、 Registration ID 概念	289
第 12 章 Android 平台与 React Native 混合开发			13.3.4	React Native 极光推送 API 与代码调用	290
12.1	Android 平台混合开发简介	253	13.3.5	服务器端进行消息推送 请求	298
12.2	Android 平台混合开发原理 详解	254	13.4	本章小结	300
12.2.1	Android 原生代码实现	254	第 14 章 iOS、Android 平台发布 与热更新		
12.2.2	Android 原生模块注册	257	14.1	App 图标与启动图	302
12.2.3	Android 包定义	258	14.2	快速生成所有平台 App 图标 与启动图的方法	308
12.2.4	React Native 中调用混合 开发代码	260	14.3	iOS 项目打包并上架 App Store	310
12.2.5	Android 平台混合开发特性 详解	263	14.4	Android 平台打包与上架	312
12.3	Android 平台混合开发实例	270	14.5	React Native 热更新	314
12.3.1	Android 原生代码实现	270	14.6	本章小结	320
12.3.2	Android 包定义	271	第 15 章 React Native 性能调优 方法与技巧		
12.3.3	Android 原生模块注册	272	15.1	性能调优基准参数	321
12.3.4	React Native 调用混合开发 代码	273	15.2	常见造成 App 性能低下的 原因	323
12.4	本章小结	276	15.3	查找性能问题以及调优方法	326
第 13 章 React Native 消息推送					
13.1	iOS 平台消息推送机制	278			

15.4 性能优化方法与组件	328
15.4.1 性能优化原则	329
15.4.2 使用特定平台组件	329
15.4.3 高性能第三方组件	330

15.4.4 资源优化	330
-------------------	-----

15.5 本章小结	331
-----------------	-----

附录 React Native 源码学习方法 及其他资源	332
---------------------------------------	-----



第 I 部分 *Part 1*

入 门

- 第 1 章 React 与 React Native 简介
- 第 2 章 Node.js 简介与开发环境配置
- 第 3 章 React Native 工作原理与生命周期
- 第 4 章 React Native 页面布局
- 第 5 章 React Native 开发调试技巧与工具
- 第 6 章 React Native 组件详解
- 第 7 章 React Native API 详解
- 第 8 章 React Native 网络请求详解
- 第 9 章 常用 React Native 开源组件详解

React 与 React Native 简介

这一章我们将对 React 与 React Native 的基本概念进行介绍。首先，详细介绍 React 产生的背景及 React 的框架，然后简单介绍 React 的底层实现原理，最后介绍 React Native 的基本概念。

你将对 React 与 React Native 框架的发展、框架之间的关系有一个基本的了解，具体的技术细节在后续的章节将有更加详细的讲解与实战解读。

1.1 React 简介

React 框架最早孵化于 Facebook 内部，Jordan Walke 是框架的创始人。React 作为内部使用的框架，在 2011 年的时候用于 Facebook 的新闻流（newsfeed），并于 2012 年用在了 Instagram 项目上。在 2013 年 5 月美国的 JSConf 大会上，Facebook 宣布 React 框架项目开源。

图 1-1 为 GitHub 上 React 的开源项目截图，地址为：<https://github.com/facebook/react/>。

React 框架产生的缘由是在当时的技术背景下，前端 MVC（Model-View-Controller）框架性能不能满足 Facebook 项目的性能需求以及扩展需求，所以 Jordan Walke 索性就自己着手开始写 React 框架，这种精神值得学习。

在当时 Facebook 内部极其复杂的项目中，面临的一个问题是，在 MVC 架构

的项目中当 Model 和 View 有数据流动时，可能会出现双向的数据流动，那么项目的调试以及维护将变得异常复杂。

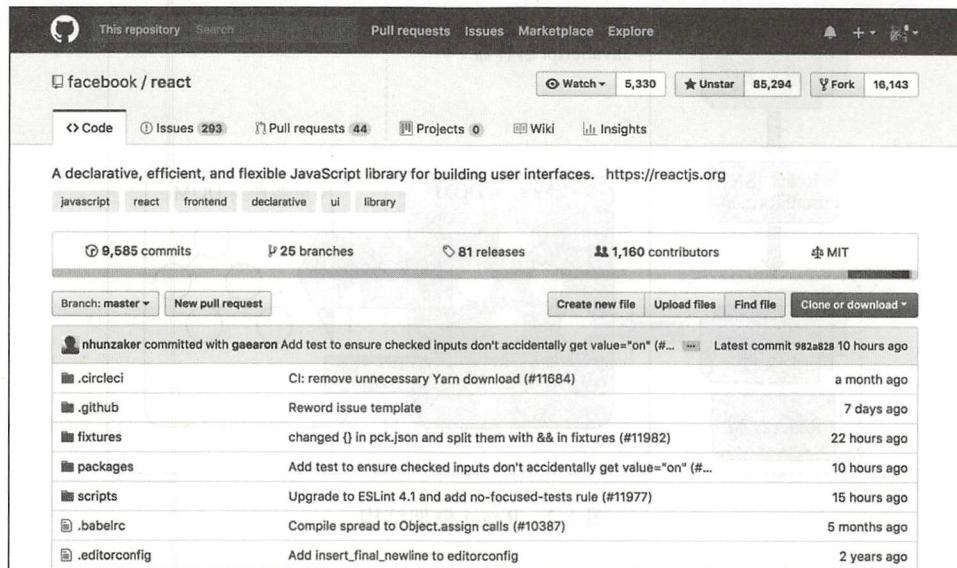


图 1-1 GitHub 上的 React 项目

React 官方也说自己不是一个 MVC 框架 (<https://reactjs.org/blog/2013/06/05/why-react.html>)，或者说 React 只专注于 MVC 框架设计模式中的 View 层面的实现。

为了大大减少传统前端直接操作 DOM 的昂贵花费，React 使用 Virtual DOM (虚拟 DOM) 进行 DOM 的更新。

图 1-2 为 React 框架的基本结构，清晰明了地描述出了 React 底层与前端浏览器的沟通机制。

React 的组件是用户界面的最小元素，与外界的所有交互都通过 state 和 props 进行传递。通过这样的组件封装设计，使用声明式的编程方式，使得 React 的逻辑足够简化，并可以通过模块化开发逐步构建出项目的整体 UI。

React 框架中还有一个重要的概念是单向数据流，所有的数据流从父节点传递到子节点。假设父节点数据通过 props 传递到子节点，如果相对父节点（或者说相对顶层）传递的 props 值改变了，那么其所有的子节点（默认在没有使用