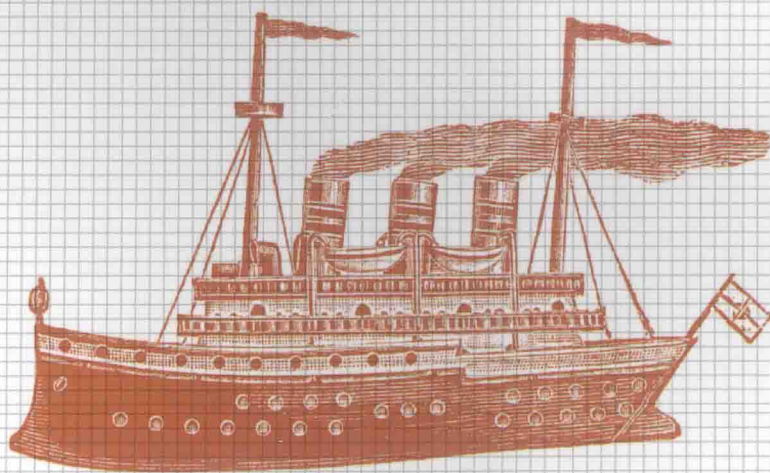


React Native

用JavaScript开发 移动应用

The React Native Book



[越南]Truong Hoang Dung 著

奇舞团 译



中国工信出版集团

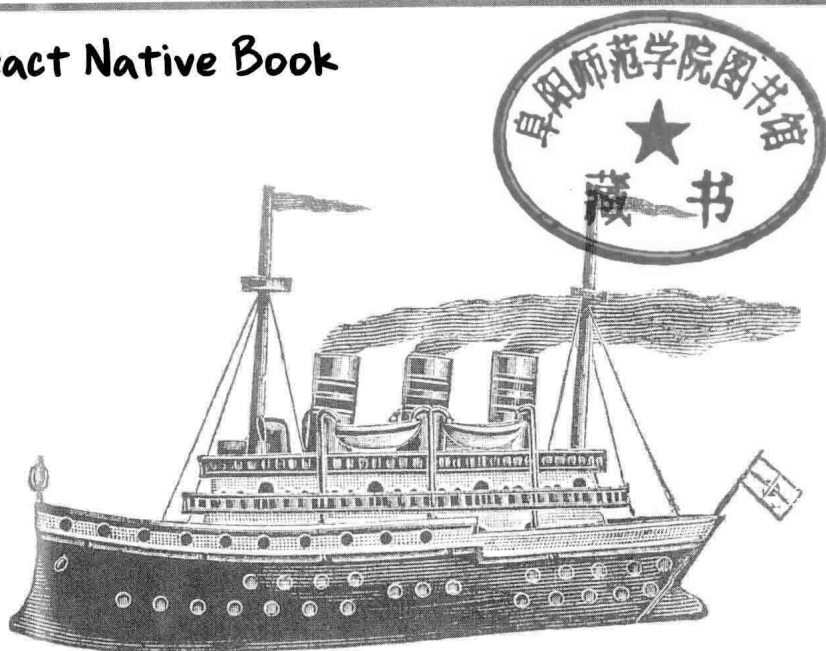


电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

React Native

用JavaScript开发 移动应用

The React Native Book



【越南】Truong Hoang Dung 著
奇舞团 译

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

内容简介

React Native 是当前移动端开发中的优秀解决方案。本书围绕着如何将一个完整 App 提交到 App Store, 讲解了使用 React Native 开发 iOS 应用所涉及的方方面面。首先介绍了 Flexbox 布局, 教大家从零开始搭建一个初始应用, 以此阐明 React Native 的基础运行机理; 然后介绍了 Flux 的设计思想, 怎么理解和使用 Promise、fetch 等新 API, 以及数据库 SQLite 存储方面的知识, 以便让你对一个完整的 App 形成感性认识; 最后讲解了怎样测试 React Native 组件, 并将完整的 App 发布到 App Store 中。

如果你对开发 Web 端的原生移动应用感兴趣, 本书就是一本不容错过的以实例代码为引导的入门书籍。

The React Native Book

Copyright © 2015 Truong Hoang Dung

Chinese translation Copyright © 2015 by Publishing House of Electronics Industry.

本书中文简体版专有出版权由 Truong Hoang Dung 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。

图书在版编目 (CIP) 数据

React Native: 用 JavaScript 开发移动应用 / (越) 张皇容著; 奇舞团译. — 北京: 电子工业出版社, 2015.10

书名原文: The React Native Book

ISBN 978-7-121-27170-0

I. ① R…II. ① 张… ② 奇…III. ① JAVA 语言—程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2015) 第 224196 号

策划编辑: 张春雨

责任编辑: 张 玲

文字编辑: 王中英

印 刷: 北京中新伟业印刷有限公司

装 订: 北京中新伟业印刷有限公司

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: 787×980 1/16

印张: 16 字数: 340 千字

版 次: 2015 年 10 月第 1 版

印 次: 2015 年 10 月第 1 次印刷

定 价: 65.00 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。

质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

译者序

从接触前端开发到现在已 7 年有余，参与了各种类型的前端应用的开发，在这期间有幸见证了前端领域乃至整个软件领域的蓬勃发展，也见证了互联网行业的飞速发展和不断创新。现在的前端开发者再也不是当年所被定位的“切图仔”，前端领域早已渗透到了软件工程的方方面面。

这几年前端技术的发展可谓百花齐放，不断地颠覆思想，用应接不暇来形容也不为过。从个人写工具函数库的时代开始，过渡到各大框架百家争艳，如 Prototype、YUI、jQuery 等；再到近几年，YUI 宣布停止更新，jQuery 奠定了在库阶段的统治地位，也标志着一个时代的结束。

随着前端业务发展愈加庞大，以及移动端业务的飞速发展，即便在单页面场景的需求下，前端工程的复杂度也提高了很多，所以业务分层和架构设计就应运而生了。在这种场景下，Backbone 一类的 MV* 框架诞生了，至此库的时代开始向框架的时代转移。MVC 迅速过渡到了以 AngularJS 等为代表的 MVVM 框架，双向数据绑定无疑是一个高效的解决方案。

这期间各大公司都在不断地创造、颠覆，然后再创造，再去颠覆自己的思想和历史。Facebook 作为前端领域的引领者，在这期间开发了 React，React 以其颠覆式的思想迅速进入了大家的视野，它重新定义了前端的架构模式。基于虚拟 DOM 的抽象，无论是内建的 DOM、Native 还是 Canvas，都是一种基础 React 模式抽象的实现，这个抽象所提供的想象空间是无限的。

React 的设计让大家开始重新对 UI 设计进行思考，而且已经不仅仅是面向浏览器了，而是面向整个前端。紧接着 Facebook 在 F8 大会上又开源了 React Native，它的设计理念是：既拥有原生 Native 的交互体验，又能够保留 React 自有的开发效率。这个理念对于目前 Hybrid 模式红火的年代而言，无疑是一个重磅炸弹。使用灵活的 HTML 和 CSS 布局，使用 React 语法构建组件，然后同时运行在 iOS 和 Android 平台上，“Learn once, write anywhere”，这不正是前端工程师们梦寐以求的模式吗？

所以国内各大公司也都纷纷投入了 React 的怀抱，不断思考和尝试所有的可能性，我们团队也很早在这方面进行了尝试实践。巧合的是，此时奇舞团接到了出版社的翻译邀约，说有一本英文的 React Native 的书籍，希望我们团队能够接下来。当时的考虑是，React 本身还处于一个初步试水的阶段，所以翻译此类书籍还是存在一些风险的：

1. 当前版本更新比较快，所以 API 很容易就会过时。
2. 由于新技术还没有沉淀，尚待在更多工程领域中大量使用，所以在短时间内很难有有深度的书籍出现。

看完书的内容，发现这并不是面向中高级开发者的书籍，而是一本帮助对 React Native 感兴趣的开发者快速入门的书籍，而且作者也从侧面介绍了 React Native 开发所涉及的方方面面。整个书籍编排是以博客文章的形式展现，每个章节都是一个独立的知识点，相互耦合度不高，比如 Promise、Flux 等技术都是可以独立学习思考的。作者用大量的代码和实例来讲解 React Native 开发的整个流程，对于入门者来说，可以照着实例逐个知识点去学习，然后顺着脉络不断深化知识点；也可以单独查看任一感兴趣的章节，对现有知识做补充和巩固。再考虑到 React Native 这个颠覆性的技术应该被国内的同行们尽快了解和掌握，所以我们小组就把这本小书的翻译工作接了下来，希望能够为前端社区贡献一点绵薄之力。

React Native 仅仅是一个开始，以后会有更多颠覆性的思想出现，来不断改革，不断精进。前端领域也会得到空前的发展，朝着越来越重要越来越不可替代的方向去进化，这无疑是我们前端开发者最大的福音。

本书翻译工作由月影领衔的奇舞团翻译小组承担，由梁超负责整体审校，由柏盼、赵岩、马磊、张颖昌、郭瑞负责具体翻译。英文原书网址为：<https://www.gitbook.com/book/checkraiser/beginning-react-native/details>。

前言

本书是一本介绍 React Native 框架的实用快速入门指南。你并不需要成为一名 Objective-C 的开发高手，当你阅读完这本书之后，我相信你就能够在一天时间内把自己的想法变为实际的应用，发布到 Apple Store 上去。

本书中你将学到的内容

- 学习如何搭建和调试一个 React Native 应用。
- 学习如何设计构建一个 iOS 应用。
- 学习如何布局 iOS 界面。
- 学习如何使用基本的 React Native/iOS 控制器和模块。
- 学习如何使用 Node.js 搭建你自己的 API 服务器。
- 学习如何在你的应用中使用 SQLite 数据库。
- 学习如何在应用中集成第三方库。
- 学习如何测试组件。
- 学习 ReactJS 的设计模式。

如何阅读本书

如果你本身就从事 Web 开发工作，那么这本书特别符合你的需求。书中的内容能够帮你通过 React Native 技术将你熟悉的 Web 开发理念直接应用到 iOS 开发中。

本书不会让你通过参考一些网站来学习枯燥的概念点，而会带你从零开始学习这些内容。在我一开始写这本书的时候，只是计划写一个电子书来说明开发应用的步骤，并且认为这样已经足够了。但是在写作的过程中，我的想法发生了改变。我发现在学习完 React/React Native

的理念后，自己就可以随处使用它们了，所以对学习者来说，在掌握所有的 React Native 技术开发自己的 iOS 应用之前，真正学习并理解所有的概念是非常有必要的。

有一个好消息是，Android 版本的 React Native 也将会发布。那时候，你在本书中学到的所有东西都可以自然而然地应用到 Android 应用开发中了。

本书的组织结构

- 每章的开头都会介绍一个概念，帮你了解这个概念在 Web 开发中是如何体现的。然后我们会将这个概念过渡到 React Native 开发中，让你知道这个概念在 React Native 中如何具体体现，这个概念哪些部分做了转变，哪些没有变化，或者哪些部分被舍弃了。
- 在本书的最后，将帮你运用所有学到的知识来开发一个真实的应用。

目录

第 1 章 安装配置新项目	1
创建新的 React Native 应用程序	1
创建步骤	1
应用程序分析	2
在已有的 iOS 应用中集成 React Native	3
首先建立一个简单的 iOS 应用	3
安装 React Native	3
创建 React Native 应用	5
将 React Native 集成到 iOS 应用中去	6
链接库和模块	10
在设备上运行 React Native 应用	11
练习	14
第 2 章 Flexbox 布局介绍.....	17
Web 开发中的 Flexbox 布局	17
Flexbox 布局究竟是什么	18
Flexbox 教程	19
栅格系统	19
圣杯布局	23
带有可变宽度搜索框的流式导航条	26
垂直居中	31
在 React Native 中使用 Flexbox	34

第 3 章 用 React Native 开发一个应用	37
在 React Native 中创建 “Hello World” 程序	37
创建项目	37
使用 Xcode	38
使用模拟器	39
项目结构	39
React 基础	41
render 函数	42
样式	43
注册组件	44
修改组件	45
搜索框和数据检索	48
增加搜索框	48
方法代理	49
检索数据	50
显示结果	53
组件状态	53
创建 ListView 组件	54
创建每个数据单元	55
给列表中每个结果项设置样式	56
最终代码	59
第 4 章 在 React Native 中使用导航	64
第 5 章 动画和滑动菜单	69
相关的代码	69
它是如何运行的	74
SlideMenu 的使用	74
扩展一些缺少的功能	77
第 6 章 用 React Native 绘制 Canvas	78
创建 Canvas 组件	78
使用 Canvas 组件	79

使用 ReactART library	81
在网页中使用 ReactART	81
在 React Native 中使用 ReactART	84
结论	91
第 7 章 使用 React Native 播放音频	92
第 8 章 你的第一个自定义视图	98
尝试创建自定义视图	99
第 9 章 Flux 介绍	105
使用 ReactJS 的弊端	105
从 React 到 Flux	106
Flux 代码	111
练习	119
第 10 章 处理复杂的应用程序状态	120
使用 Flux 所面临的问题和解决方案	120
初识 Baobab	121
把状态树并入 ReactJS	123
将单个游标连接到组件	123
将多个游标连接到组件	124
如何改变 Baobab 树的状态	125
使用 Baobab 树构建应用程序状态	127
使用 Baobab 创建稍微复杂的例子	128
ReactJS 的级联渲染	130
ReactJS 的重复渲染	131
第 11 章 使用 Node 来实现服务端 API	133
安装 Express	133
Request 和 Response 对象	135
理解中间件	137
从 URL 中获取参数	140

处理相关参数	141
路由实例	141
重构路由	142
使用 JSON	144
JSON 结构	144
在 JavaScript 中使用 JSON	146
iOS 中的 HTTP	147
Web 服务	149
NSURL、NSURLRequest、NSURLSession 和 NSURLSessionTask	150
构造 URL 和发送请求	150
使用 NSURLSession	151
JSON 数据	153
解析 JSON 数据	153
主线程	154
更深入的学习：请求主体	155
第 12 章 在 React Native 中使用文件上传.....	157
创建服务器	157
注解	159
进行文件验证	160
总结	161
React Native 文件上传的客户端	161
第 13 章 理解 JavaScript Promise	170
事件机制并不是万能的	171
Promise 术语	172
JavaScript 中的 Promise	173
浏览器支持和 Polyfill	174
与其他库的兼容性	174
让复杂的异步代码变得更加简单	175
将 Promise 用于 XMLHttpRequest	176
链式调用	177
数值处理	177
异步操作队列	178

错误处理	179
JavaScript 异常和 Promise	181
错误处理实战	182
并行和串行——鱼和熊掌兼得	183
创建序列	184
Promise 和 Generator	190
Promise API 参考	192
静态方法	192
构造器	193
实例方法	193
第 14 章 fetch 简介	194
XMLHttpRequest 快要过时了	194
fetch 基础使用	194
使用 XMLHttpRequest 实现	194
使用 fetch 实现	195
响应中的元数据	196
响应类型	196
Promise 方法链	197
使用 fetch 请求发送凭证信息	198
问答环节	198
第 15 章 在 iOS 中使用 SQLite	200
创建并打开数据库	201
绑定变量	202
开发 SQLite 模块	203
第 16 章 集成 Google Admob	209
创建项目	209
将 SDK 添加到你的 Xcode 项目	209
第 17 章 React Native 组件国际化	214
我要做什么	214
工作机制	214

安装步骤	214
在项目中使用	218
第 18 章 测试 React Native 组件	221
安装配置测试环境	221
编写测试程序	226
附录 A React.js 快速介绍	228
React 的特殊语法：JSX	228
开始和运行	229
构建 React 的基石：组件	229
在 JavaScript 代码中出现了 HTML 片段	230
组件和状态	231
附录 B Objective-C Primer	233
类的基本语法	233
方法	233
属性	234
实现	235
规范	237
附录 C webpack 入门	238
为什么要使用 webpack	238
webpack 也适合 Browserify 用户	238
启动 webpack	239
loader 的使用	239
样式表和图片	240
功能开关	241
多个入口点 (entrypoint)	242
优化共用代码	243
异步加载	243

第 1 章

安装配置新项目

创建新的 React Native 应用程序

创建步骤

在 React Native 的主页上，你能看到以下的内容。

配置环境要求

1. OS X: 目前只有 OS X 中能够安装使用 iOS 的开发环境，并且 Xcode 也只能运行在 Mac 上。
2. 最新版的 Xcode 开发工具：在 Mac 的 App Store 上下载就可以。
3. 推荐使用 Homebrew 来安装 node、watchman 和 flow。
4. 使用命令行 `brew install node` 来安装 node。
5. 使用命令行 `brew install --HEAD watchman` 来安装 watchman。我们推荐安装 watchman，否则你可能会需要一个 node 文件来监控开发中的 bug。
6. 使用命令行 `brew install flow`，如果你要使用 flow 的话。

快速开始

1. 使用命令行 `npm install -g react-native-cli`。
2. 使用命令行 `react-native init AwesomeProject`。
3. 使用命令行 `cd AwesomeProject`。

在新创建的文件夹 AwesomeProject 中：

1. 使用 Xcode 打开 AwesomeProject.xcodeproj 文件并运行项目。
2. 用你常用的文本编辑器打开 index.ios.js 文件，选中几行修改一下代码。
3. 在 Xcode 中打开 iOS 模拟器，按下组合键 Cmd+R（两次）重新加载应用，来观察修改的效果。

恭喜你！你刚刚已经成功地修改并运行了你的第一个 React Native 应用。

应用程序分析

我们来仔细研究一下新创建的应用程序。

首先来看一下 AwesomeProject/ 下都包含着什么内容：

1. 文件夹 AwesomeProject.xcodeproj
2. 文件夹 iOS
3. 文件 index.ios.js
4. 文件夹 node_modules/
5. 文件 package.json

为了让你能有一个全面的了解，这里会详细地说明一下运行 react-native 命令初始化 AwesomeProject 项目时都发生了什么事情。

- 新建了一个名为 AwesomeProject 的文件夹。
- 在这个文件夹中建立 package.json 文件。
- `npm install --save react-native` 命令开始运行，这个命令安装了 react-native 及其依赖，这些依赖放在 AwesomeProject/node_modules 文件夹中。在 AwesomeProject/package.json 中声明 react-native，作为你的项目的依赖。
- 全局安装 react-native CLI（react-native 命令行）工具，并且将控制权交给本地的 CLI 工具，就是 AwesomeProject/node_modules/react-native/local-cli/cli.js 文件。
- 接下来执行 AwesomeProject/node_modules/react-native/init.sh 文件。这个文件是用来生成标准代码的脚本，比如 index.ios.js、iOS 文件夹中的 Objective-C 代码，以及 AwesomeProject.xcodeproj 文件夹中的 Xcode 项目配置。

我们已经看到，React Native 是基于 JavaScript 的。这些 JavaScript 代码在 index.ios.js 中组成一个真实的应用。

package.json 文件对于已经使用过 Node.js 的开发者来说并不陌生，它定义了我们项目的一些元数据。更重要的是，它声明了 react-native 作为我们项目的依赖。

node_modules 文件夹是 npm install 命令进行项目初始化时生成的。它包含 react-native 源代码、其他的 npm 依赖、大量的 JavaScript 和 Objective-C 代码。

初始化进程也提供了最基本的 Xcode 项目定义和一些附加的 Objective-C 样板代码，它们允许我们在 Xcode 中打开一个项目并运行一个应用，而不需要任何其他的附加条件。这些本来都应该是手动完成的，但是因为包含了很多流程化的内容，而且这些内容不管我们创建什么类型的应用都是需要的，所以就可以通过 react-native 来简化这些步骤。

如果要将第三方库整合到 React Native 项目中，请在根目录下增加 Podfile 文件。

Podfile 是一个类似于 package.json 的东西，它为 CocoaPods 做依赖管理，声明了 Objective-C 库的依赖。我们将在本书后面讨论更多有关它的细节。

可以看出，我们的 AwesomeProject 项目是一个综合的项目。它是一个 Xcode 项目，也是一个 NPM 项目。它是一个基于 JavaScript 的 React Native 应用，也包含了很多 iOS 胶水代码，用来让我们的 JavaScript 代码运行在 iOS 环境中。

在已有的 iOS 应用中集成 React Native

接下来我们来看一个例子，看看如何在已经存在的 iOS 应用中集成 React Native 的内容。

首先建立一个简单的 iOS 应用

如果你已经有一个原生的 iOS 的应用了，则可以跳过这一步。如果你还没有现成的原生 iOS 应用，可以先通过 Xcode 来建立一个应用，步骤如下。

1. 打开 Xcode，选择 Create a new Xcode project。
2. 选择 Single View Application 建立一个最简单的单页面的应用，然后点击 Next 按钮。
3. 给建立的应用命名（我命名为 simpleApp），并选择要保存的目录，点击 create 就建立了一个最简单的应用，只不过这个应用目前并没有内容和逻辑。

安装 React Native

要在一个已经存在的 App 中集成 React Native 的内容，我们就需要先安装 React Native。不同于之前的安装方式，我们需要通过 CocoaPods 来安装。CocoaPods 是用于 iOS/Mac 开发的一个包管理工具。在下面的三种应用场景下，你可以考虑在项目中使用 CocoaPods。

- 你想要将第三方库（第三方库使用了 CocoaPods）整合到新创建的 React Native 项目中。
- 目前你的项目使用 CocoaPods 来管理依赖和库。
- 你想要将 React Native 框架添加到一个现有的项目中。

如果你还没有安装 CocoaPods，运行下面这个命令来安装：

```
sudo gem install cocoapods
```

注意，你需要提前安装 Ruby 来使用 gem 命令。

根据 React Native 官网最新的说明，它用 io.js 来取代了 node.js。让我们按照官网的指导来进行 io.js 的安装和配置。如果之前安装过 node.js，则先运行命令：

```
brew unlink node
```

这样可以完成 node 的解绑。然后运行：

```
brew install iojs  
brew link iojs --force
```

进行 io.js 的安装和配置。

现在我们已经准备好使用 CocoaPods 了，使用 CocoaPods 来安装 React Native 吧。

1. 进入到刚才那个 iOS 应用的文件目录，建立一个文件，命名为 Podfile。在使用 CocoaPods 安装 React Native 时，Podfile 相当于一个配置文件，告诉安装程序需要下载安装哪些内容。
2. 在 Podfile 中添加如下的文本内容。

```
pod 'React'  
pod 'React/RCTText'
```

这两行的内容告诉 CocoaPods 安装命令需要下载安装的内容为 React 和 React/RCTText（需要安装 React 和 React 中的 Text 组件）。同理，如果需要通过 CocoaPods 下载安装其他的内容，只要在 Podfile 中添加相应的内容就可以。

然后在终端中执行以下命令：

```
pod install
```

至此，你已经成功安装了 React Native。