

Web开发经典丛书

使用React、JSX、Redux和GraphQL开发Web Apps

# 快速上手 React 编程

John Sonmez  
作序推荐本书

[美] 阿扎·马尔丹 (Azat Mardan) 著  
郭美青 郭松 唐金州 译

 MANNING

 清华大学出版社

Web 开发经典丛书

# 快速上手 React 编程

[美] 阿扎·马尔丹 (Azat Mardan) 著  
郭美青 郭 松 唐金州 译

清华大学出版社

北 京

Azat Mardan  
React Quickly

EISBN: 978-1-61729-334-4

Original English language edition published by Manning Publications, 178 South Hill Drive, Westampton, NJ 08060 USA. Copyright © 2017 by Manning Publications. Simplified Chinese-language edition copyright © 2018 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Manning 出版公司授权清华大学出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。  
版权所有，侵权必究。

北京市版权局著作权合同登记号 图字：01-2017-7950

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。  
版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

#### 图书在版编目(CIP)数据

快速上手 React 编程 / (美)阿扎·马尔丹(Azat Mardan) 著；郭美青，郭松，唐金州 译. —北京：清华大学出版社，2018

(Web 开发经典丛书)

书名原文：React Quickly

ISBN 978-7-302-50247-0

I. ①O… ②快… II. ①阿… ②郭… ③郭… ④唐… 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(2018)第 103232 号

责任编辑：王 军 李维杰

装帧设计：思创景点

责任校对：孔祥峰

责任印制：刘海龙

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969，[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈：010-62772015，[zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者：清华大学印刷厂

经 销：全国新华书店

开 本：185mm×260mm 印 张：28.75 字 数：700 千字

版 次：2018 年 6 月第 1 版 印 次：2018 年 6 月第 1 次印刷

印 数：1~3500

定 价：88.00 元

---

产品编号：075962-01

# 本书赞誉

“对于任何想要获得 React 入门指导，并了解其周边生态系统(包括工具、概念和库)的人来说，本书提供了一种一站式服务。跟随 Azat 的演练，完成给定的项目，你将很快理解 React、Redux、GraphQL、Webpack 和 Jest，以及如何让它们运行起来。”

——Peter Cooper, *JavaScript Weekly* 杂志编辑

“本书通过 React 向读者传授在构建现代 Web 应用的过程中最具价值和值得关注的概念，包括 GraphQL、Webpack 和服务器端渲染。阅读之后，你应该有足够的信心能够使用 React 创建一个生产级的 Web 应用。”

——Stan Bershadskiy, *React Native Cookbook* 一书的作者

“Azat 是编程领域最具权威的声音之一。本书深入研究了 React 的基础和架构，并远远超越了基础。对于任何开发者来说，它都是必读的！”

——Erik Hanchett, *Ember.js Cookbook* 一书的作者

“这本书很简单。它使用非常基础的语言让你逐步了解每一个概念。”

——Israel Morales, SavvyCard 前端开发者和 Web 设计师

“简单的语言配以简单的逻辑示例，让你起步并快速运行程序，本书涵盖了任何 React 新手在开始编写应用时都会遇到的主要问题。作者的幽默感会让你一直忙到最后。感谢 Azat 花时间与我们分享他的 React 之旅。”

——Suhas Deshpande, Capital One 软件工程师

“本书是加速学习 React 的好资源，非常透彻和中肯。我将用它作为开发下一个应用的参考。”

——Nathan Bailey, SpringboardAuto.com 全栈开发者

“Azat 非常擅长他所做的事情——教人如何写代码。本书包含基础知识和实例，能让你快速开始使用 React。”

——Shu Liu, IT 顾问

“自 2013 年被 Facebook 开源以来，React 已经迅速成为一个被广泛采用的 JS 库，并且

是 GitHub 上最著名的项目之一。在他的新书《快速上手 React 编程》中，Azat Mardan 以他典型的简明风格，把你需要了解的关于 React 生态系统的一切都编排好了，以便快速构建高性能的 SPA 应用。仅仅关于 React 状态和同构 JavaScript 的章，就值得你购买和阅读。”

—Prakash Sarma, New Star Online

“本书将通过传达清晰的基础知识来让你慢慢接纳 React，它将让你构建应用并彻底接受使用 React 带来的好处。”

—Allan Von Schenkel, FoundHuman 技术与战略副总裁

“本书以易于理解的方式涵盖了 React 的所有重要方面。和 Azat 的所有作品一样，本书清晰而简洁，并且包含快速提高生产力所需的内容。如果你有兴趣将 React 加入自己的技能集，我建议从这里开始。”

—Bruno Watt, hypermedia.tech 咨询架构师

“这是一本关于使用 React 进行全栈 Web 开发的非常全面的书，不仅涵盖 React 本身，还包含周边的生态系统。我总是对 React 服务端渲染感到困惑，Azat 的这本书终于帮我理解了它。如果你是 React 新手并想要真正掌握它，看这本书就足够了。”

—Richard Kho, Capital One 软件工程师

# 译者序

在本书翻译之初，React 社区刚刚经历了两个重量级的事件。一件是前后持续一年有余的 Facebook 附加专利条款事件，后因“React 能不能继续商用”的问题在国内被炒得沸沸扬扬，但随着事件持续发酵，最终迎来了让开发者皆大欢喜的局面；另一件是 React 团队发布了最新的基于 Fiber 架构的 v16.0.0 正式版，带来了众多优秀特性和性能提升。一方面我们看到了 React 无与伦比的业界影响力，甚至连 Facebook 都不惜以得罪开源社区的方式，希望从 React 身上获得商业利益保护的能力；另一方面我们又看到一支孜孜不倦、勇于探索、不断进取的行业先锋团队，当大部分人还在纠结该使用 React、Vue 还是 Angular 2 的时候，React 团队再一次用 Fiber 架构突破浏览器自身的局限，为 React 的发展带来更大的想象空间。React 无疑是当下最热门、最具前景的前端技术之一。

当下，我们正处在“JavaScript 文艺复兴时代”。各种标准、思想、框架、类库百花齐放，各种工具、概念、术语层出不穷。条条大路通罗马，要实现一个前端应用，解决方案除了当年那个永恒的 jQuery 以外，可谓数不胜数，涉及的专有名词更是目不暇接。这是件令人兴奋的事情，因为这意味着前端生态在不断壮大，JavaScript 的能力也在日益增强。但对于前端从业者而言，却是不小的挑战。

依稀记得刚接触 React 时，面对扑面而来的新鲜术语的那种困惑，就好像看见散落一地的珍珠，却不知该如何捡起串成美丽的项链一样。Azat 无疑非常擅长这项工作，他把自己在一线教授 React 的课程、经验、资源以及学员的反馈重新梳理、优化、整合，按照由浅入深、由表及里的编排方式，不仅关注核心，也注重生态，同时配上丰富的实战案例和架构思想解读，为我们带来了这本学习 React 不容错过的《快速上手 React 编程》。相信这本书一定会为读者带来醍醐灌顶般的阅读体验，因为该书不仅仅教你如何使用 React，还告诉你 React 技术体系中每一项设计的原因和思考。希望读者能从中受益，享受学习和使用 React 带来的快乐，这也是译者莫大的荣幸。

本书由郭美青、郭松、唐金州合译完成。特别感谢梁宵的引荐，让我们得以共同合作参与本书的翻译，在整个翻译过程中，他高效的组织、协调能力和审校的专业让我们受益匪浅。感谢我们的家人，你们的支持和鼓励让我们感到无比的温暖。感谢清华大学出版社李阳老师的悉心指导以及在翻译过程中给予的极大理解和帮助。

我们在本书翻译过程中力求做到行文流畅、风格一致，希望能给读者带来愉悦的阅读体验，并尽力修正一些原书中的小错误，但鉴于自身水平有限，疏漏在所难免，敬请广大读者批评指正。

最后，希望本书能帮助业界同仁系统地掌握 React 并使用 React 打造出高性能、更易于维护的 Web 产品。

译者

# 序 言

认真地讲，我一直希望 JavaScript 可以凋零、磨灭。

这并不是说我完全不喜欢 JavaScript——多年以来，它已经改进了不少；而是因为我对复杂的极度厌恶——以至于我将我的博客和公司起名为 Simple Programmer。我的口号一直是：“让复杂变得简单。”

让复杂变得简单并不容易。它需要一套特殊技能。你必须能够理解复杂，并把它理解得很好，这样你才能够从中提炼出核心——因为任何事情在核心上都是简单的。这正是 Azat 在本书中所做的。

现在，我承认 Azat 提供了一点帮助。你能看出，我个人喜欢 React 的原因之一就是它很简单。它被设计得很简单，用于处理越来越复杂的 JavaScript 框架，并通过回归基础(即原始的 JavaScript)来降低复杂性(至少在大多数情况下是这样的)。React 的确有一种可以编译成 JavaScript 的 JSX 语言，但是我会让 Azat 介绍给你)。

重点是，尽管我喜欢 Angular、Backbone，以及一些其他的 JavaScript 框架，因为它们能够帮助 Web 开发者更容易地创建异步 Web 应用和单页面应用，但这同时也增加了大量的复杂性。使用模板并理解这些框架的语法和微妙之处能提高生产率，但它们把复杂性从后端转移到了前端。React 一开始就摒弃了模板，而是提供了一种使用 JavaScript 将基于组件的架构应用于 UI 的方式。我喜欢这样。它很简单。但即使最简单的事情也很难解释——或者更糟糕的是，被缺乏这种能力的老师变复杂了。

这正是 Azat 的强项。他知道如何教学，如何简化。他通过解释 React 开始本书，对比了你可能已经知道的 Angular。即使你不了解 Angular，他对 React 的解释也能快速帮助你理解其基本原理和用途。然后，Azat 很快示范了如何创建一个基本的 React 应用，这样你就能了解并亲自实践了。在那之后，他会通过任何人都很容易掌握的真实示例，带你去了解你需要知道的 20%，以便完成你将在 React 中做的剩余 80%。最后——也是我最喜欢的部分——里面包含了丰富的例子和项目。学习的最佳方式绝对是实践，而 Azat 会使用 React 带你去创建 6 个非凡的项目。

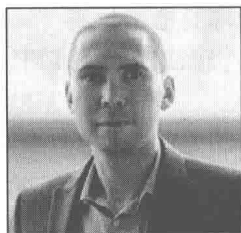
为了保持我简单的风格，让我最后说一句：《快速上手 React 编程》提供了我所知道的学习 React 的最佳方法。

John Sonmez

*Soft Skills* 一书的作者(<http://amzn.to/2hFHXAu>)

Simple Programmer 创始人(<https://simpleprogrammer.com>)

# 作者简介



我已经出版了超过 14 本书和 17 门在线课程(<https://node.university>), 它们中的大多数存储在云端, 涉及 React、JavaScript 和 Node.js。在关注 Node 之前, 我也使用过其他语言编程(Java、C、Perl、PHP、Ruby), 几乎从高中开始编程(十多年前), 并且绝对超过了规定的一万小时。

现在, 我是美国十大银行之一的技术研究员, 它也是一家财富 500 强公司: Capital One Financial Corporation, 位于美丽的旧金山。在那之前, 我曾为小型初创公司、大型企业, 甚至美国联邦政府工作过, 编写桌面、Web 和移动应用, 从事教学、开发协调和项目管理。

我不想占用你太多的时间介绍自己, 你可以在我的博客(<http://webapplog.com/about>)和社交媒体([www.linkedin.com/in/azatm](http://www.linkedin.com/in/azatm))上了解我的更多信息。相反, 我想把关于这本书的经历写下来。

2011 年, 我搬到了阳光明媚的加州, 开始通过企业加速器创业(如果你对此好奇, 它就是 500 家初创公司), 开始使用现代的 JavaScript。我学会了使用 Backbone 为公司创建一些应用, 并对此印象深刻。该框架极大地改进了我前几年构建的单页面应用的代码组织。它有路由和模型。这很棒!

我在 DocuSign(去 Google 搜索一下 e-signatures, 它的市场占有率达到 70%)担任软件工程团队领导时, 又有机会看到了 Backbone 和同构 JavaScript 的惊人能力。我们重构了一个有 7 年历史的 ASP.NET Web 应用, 每一个小版本的发布都需要四周的时间, 而使用时髦的 Backbone-Node-CoffeeScript-Express 应用不仅具有非常好的用户体验, 而且发布版本只需要一到两周的时间。设计团队在可用性方面做得很好。毋庸置疑, 有大量具有不同程度交互的 UI 视图。

最终的应用是同构的, 甚至在此之前还不存在该术语。我们使用服务器上的 Backbone 模型从 API 获取数据并将其缓存。我们在浏览器和服务器上使用相同的 Jade 模板。

这是一个有趣的项目, 它让我更加相信使用一种语言横跨整个技术栈的力量。精通 C# 和前端 JavaScript(大部分是 jQuery)的老款应用开发者, 如果花点时间突击一下(一个发布周期, 通常是一到两周), 就会爱上结构清晰的 CoffeeScript、Backbone 的组织结构以及 Node 的速度(开发和运行速度)。

我在 Web 开发领域十几年的经验向我揭示了好的、坏的以及丑陋的(大部分是丑陋的)前端开发。然而柳暗花明, 因为自从切换到 React, 我就越来越喜欢它了。

如果你想收到更新、新闻和小贴士, 可以在网上通过以下方式联系我, 订阅、加好友、



悄悄关注均可：

- Twitter—[https://twitter.com/azat\\_co](https://twitter.com/azat_co)
- 网站—<http://azat.co>
- LinkedIn—<http://linkedin.com/in/azatm>
- 专业博客—<http://webapplog.com>
- 出版物—<http://webapplog.com/books>

关于面对面研讨会和课程，请访问 <http://NodeProgram.com> 或 <https://Node.University>，  
或通过 <https://webapplog.com/azat> 给我发消息。

# 致 谢

我要感谢互联网、全世界以及人类的聪明才智，让我们相信心灵感应是可能的。不必开口，就可以通过 Twitter、Facebook 和 Instagram 等社交媒体与全球数百万人分享自己的想法。

我对我的老师心存感激，无论是在中学还是在大学，不经意间，我从书本和耳濡目染的学习中领悟到他们的智慧。

史蒂芬·金说过：“写作是人类做的事情，编辑是上帝做的事情。”因此，我要无数次感谢本书的编辑，更要感谢那些读者，他们在本书中不得不面对不可避免的打字错误和 bug。这是我的第 14 本书，我知道无论怎样都会产生这些问题。

感谢 Manning 出版社的工作人员，他们使本书付梓成为可能：发行人 Marjan Bace 以及所有编辑制作团队，包括 Janet Vail、Kevin Sullivan、Tiffany Taylor、Katie Tennant、Gordan Salinovic、Dan Maharry，还有很多其他的幕后工作者。

我不知如何表达对 Ivan Martinovic 领导的超凡技术审校人团队的感谢：James Anaipakos、Dane Balia、Art Bergquist、Joel Goldfinger、Peter Hampton、Luis Matthew Heck、Ruben J. Leon、Gerald Mack、Kamal Raj 和 Lucas Tettamanti。他们的贡献包括捕获术语中的技术失误、错误、拼写错误，并提供专题建议。每个人都经历了审校过程，论坛主题中的每一条反馈塑造了本书的手稿。

在技术方面，特别感谢 Anto Aravinth，他是本书的技术编辑；还要感谢 German Frigerio，他是本书的技术校对。他们正是我所期望的最好的技术编辑。

非常感谢 John Sonmez 为本书作序，他闻名于 Pluralsight、Manning 出版社以及 SimpleProgrammer.com。感谢 Peter Cooper、Erik Hanchett 和 Stan Bershadskiy 所做的审校工作，你们赋予本书额外的可信度。没有听说过 John、Peter、Erik 或 Stan 的读者应该关注他们在软件工程领域所做的工作。

最后，感谢所有 MEAP 读者的反馈。基于你们的评论，我修订了本书，虽然使本书出版延迟了一年，但却造就了当下关于 React 的一本好书。

# 前 言

那是 2008 年，银行纷纷倒闭。我在联邦存款保险公司(Federal Deposit Insurance Corporation, FDIC)工作，主要任务是偿还倒闭、失败和破产的银行储户。我承认，就职业安全感而言，我的工作等同于在雷曼兄弟上班或在泰坦尼克号上卖票。但是，当我所在的部门还没有开始最终的预算削减时，我还有机会开发一款名为 EDIE 的应用。这款应用由于一个简单的原因而变得非常流行：人们急于想知道他们的存款中有多少由美国联邦政府提供保险，而 EDIE 能够估算这一数额。

但是存在一个问题：人们不喜欢把他们的私人账户告诉政府。为了保护他们的隐私，这款应用完全在前端完成：JavaScript、HTML 和 CSS，没有任何后端技术。如此，FDIC 就不会收集任何个人财产信息。

这款应用经历了数十次迭代，留下了一团混乱的意大利面条式代码。开发人员来去匆匆，没有留下任何文档，也没有任何符合逻辑的简单算法。这就像没有地图就要去乘坐纽约市地铁一样。有很多函数可以调用其他函数、奇怪的数据结构以及更多的函数。在现代术语中，这款应用采用纯用户界面(User Interface, UI)，因为没有后端。

我多么希望那时能有 React。React 带来了愉悦。它是一种全新的思考方式，也是一种全新的开发方式。将核心功能放在一处，而不是将其分解为 HTML 和 JS，这种简单是一种解放。它重新点燃了我对前端开发的热情。

React 是开发 UI 组件的一种新方法。它是新一代的表现层库。与模型和路由库一起配合，React 可以在 Web 和移动技术栈中取代 Angular、Backbone 或 Ember。这就是我写这本书的原因。我从来不喜欢 Angular：它太复杂和大一统了。模板引擎是特定领域的，甚至不再是 JavaScript；它是另一种语言。我使用过 Backbone，喜欢它的简单和 DIY 方式。Backbone 是成熟的，它更像是你自己框架的基石，而不是一个全面的、大一统的框架。Backbone 的问题是模型与视图之间不断增加的交互复杂性：多个视图更新多种模型，有的会更新其他视图，有的会触发模型上的事件。

举办 React 在线课程(<http://mng.bz/XgkO>)，以及参加各种会议和活动的个人经验已经显示，开发者渴望一种更好的开发 UI 的方法。大多数商业价值存在于 UI，而后端负责提供商品。在海湾地区，也是我居住和工作的地方，大多数软件工程师的职位空缺是前端或(一个时髦的头衔)通才/全栈开发者。只有少数像谷歌、亚马逊和 Capital One 这样的大公司，对数据科学家和后端工程师的需求依然强劲。

想要确保职业安全感，或找到一份好工作，最好的方法就是成为一名多面手。要做到这一点，最快的方法是使用同构的、可扩展的、对开发者友好的库，就像前端的 React 搭配后端的 Node.js，以防止需要处理服务器端代码。

对于移动开发者来说,HTML5 是两三年以前的一个肮脏词汇。Facebook 放弃了 HTML5 应用,转而支持更高效的原生(native)实现。但是这种不利观点正在迅速改变。通过 React Native,可以渲染移动应用:可以保留 UI 组件,但在不同的环境下调整它们,这是另一种支持学习 React 的观点。

编程可以是创造性的。不要陷入乏味的任务、复杂的事情以及伪关注点分离。砍掉所有不必要的垃圾,通过 React 提供的简化的模块之美、基于组件的 UI 释放你的创造力。为同构的 JavaScript 引入一些 Node,你将达到禅定的境界。

祝你阅读愉快,并让我知道你对这本书的评价,请在 Amazon.com(<http://amzn.to/2gPxy9Q>)上留言。

# 学习建议

本书旨在解决前端开发者的烦恼，使他们的生活更有意义、更加快乐，并通过介绍 React 帮助他们赚到更多的钱——这些会以一种快速的方式进行。本书是十几个人工作一年半的成果。至少，本书会开拓你的思维，让你了解一些不寻常的概念，如 JSX、单向数据流和声明式编程。

## 路线图

本书分为两部分：“React 核心”（第 1 至第 11 章）和“React 生态”（第 12 至第 20 章）。每一章都会为描述性的文字配以恰当的代码示例和图表。每一章也都有一段可选的介绍视频，帮助你决定是否需要阅读该章。各章是以独立的方式编写的，这意味着即便不按顺序阅读，也不会带来麻烦——尽管建议按顺序阅读。每章的结尾是一个小测验，以强化你对该章内容的记忆，最后还有小结。

每一部分都以一系列更大的项目结尾，这些项目将给你带来更多的 React 经验，通过前几章介绍的概念和知识构建应用，巩固你对知识的理解。这些项目以可选的截屏视频作为补充来强化你的学习，向你展示动态的过程，如创建文件和安装依赖（Web 开发包含很多组成部分！）。这些项目是书中不可或缺的一部分——不要跳过它们。建议你自己键入每一行代码，并避免复制和粘贴。研究表明：打字和写作能提高学习效率。

本书以 5 个附录为结尾，它们提供了补充材料。在开始阅读之前，请连同目录一起把它们看一遍。

本书的网站是 [www.manning.com/books/react-quickly](http://www.manning.com/books/react-quickly) 和 <http://reactquickly.co>。如果需要最新的信息，很可能会在那里找到。

源代码可以在 Manning 出版社的网站 ([www.manning.com/books/react-quickly](http://www.manning.com/books/react-quickly)) 和 GitHub (<https://github.com/azat-co/react-quickly>) 上找到。查看“源代码”部分可获得更多详情。读者也可扫描本书封底的二维码来下载源代码和每章的介绍视频。书中已展示完整的代码清单——这比跳转到 GitHub 或代码编辑器去查看文件方便得多。

## 本书适合的读者(必读！)

本书适合 Web 和移动开发者以及具备两到三年工作经验的软件工程师阅读，他们想开始学习并使用 React，用于 Web 或移动开发。基本上，本书适合那些对开发者工具的快捷

键了然于心的人(比如 Mac 上的 `Cmd+Opt+J` 或 `Cmd+Opt+I`)。本书的目标读者是知晓并熟悉以下概念的人:

- 单页面应用(Single Page Application, SPA)
- RESTful 服务和 API 架构
- JavaScript, 尤其包括闭包、作用域, 以及字符串和数组方法
- HTML、HTML5, 以及它们的元素和属性
- CSS 及其样式, 以及 JavaScript 选择器

有 jQuery、Angular、Ember、Backbone 或其他 MVC 类框架的使用经验是一个加分项, 因为你可以用 React 的方式来对比它们。但这是没有必要的, 在某种程度上甚至可能是有害的, 因为你需要忘掉某些模式。React 并不完全是 MVC。

你将使用命令行工具, 如果害怕它们, 那么这是对抗命令行/终端/命令提示符恐惧的最佳时机。一般来说, CLI 比它们的可视化(GUI)版本更加强大和通用(比如 Git 的命令行版及桌面版, 后者把我搞糊涂了)。

熟悉 Node.js 会让你比从未听说过 Node.js、npm、Browserify、CommonJS、Gulp 或 Express.js 的人学得更快。我写过几本关于 Node.js 的书, 对于那些想要重温它的人, 最流行的就是 *Practical Node.js*(<http://practicalnodebook.com>)。或者, 你也可以去网上寻找免费的 NodeSchool 课程(<http://nodeschool.io>)(免费并不总是意味着更糟)。

## 本书未包含的内容(也请必读! )

本书并不是一本全面的 Web 或移动开发指南。本书假设你已经了解这些知识。如果你想学习基本的编程概念或 JavaScript 基础知识, 有很多关于这些主题的好书。我想到的有: Kyle Simpson 的 *You Don't Know Js*(<https://github.com/getify/You-Dont-Know-JS>)、*Secrets of the JavaScript Ninja, Second Edition*([www.manning.com/books/secrets-of-the-javascript-ninja-second-edition](http://www.manning.com/books/secrets-of-the-javascript-ninja-second-edition)), 以及 Marijn Haverbeke 的 *Eloquent JavaScript*(<http://eloquentjavascript.net>)。所以, 我没必要在本书中重复已有的内容。

## 如何使用本书

首先, 你应该阅读这本书。这不是开玩笑。大多数人把书买回来, 但从不去读。阅读这本书吧, 并逐章去完成那些项目。

每一章都涉及一个或一系列主题, 它们互为基础。因此, 建议你从头到尾阅读这本书, 然后将个别章作为参考。但正如之前所说, 也可以不按顺序阅读个别章, 因为各章的项目是独立的。

有许多外部资源的链接。它们中的大多数是可选的, 并提供了关于该主题的额外详情。因此, 建议在电脑旁阅读本书, 以便在提到它们时可以打开链接。

有时你会看到带有奇怪缩进的代码，比如：

```
document.getElementById('end-of-time').play()  
}
```

这表示正在注解一大块代码并将其分解成块。这一块属于一个从位置 0 开始的更大的代码清单，这一小块代码不能自己运行。

其他时候，代码块没有缩进。这种情况下，可以假设代码片段是完整的：

```
ReactDOM.render(  
<Content />,  
  document.getElementById('content')  
)
```

如果看到一个美元符号(\$)，这表明是一条终端/命令提示符命令。例如：

```
$ npm install -g babel@5.8.34
```

在使用本书的时候，最重要的是要知道并记住：你必须乐在其中。如果不好玩，那就不是 JavaScript！

## 源代码

书中的所有代码可从 [www.manning.com/books/react-quickly](http://www.manning.com/books/react-quickly) 和 <https://github.com/azat-co/react-quickly> 下载，也可通过扫描封底二维码获得。请遵循文件夹命名约定 chNN，此处 NN 是以 0 开始的各章编号(例如，ch02 文件夹代表里面是第 2 章的代码)。GitHub 仓库中的源代码会不断更新，包括补丁、bug 修复，甚至可能是新的版本和风格(ES2020?)。访问网址 <http://reactquickly.co/demos>，可以得到各章的一些示例。

## 勘误

书中肯定会有打字错误。是的，虽然编辑们都很负责——他们都是 Manning 出版社的专业人士。但是，我仍对你能找到那个错误表示感激。请不要给我发送 bug 或打字错误。取而代之，你可以在本书的论坛(<https://forums.manning.com/forums/react-quickly>)上报告，或者在 <https://github.com/azat-co/react-quickly/issues> 上创建一个 GitHub issue。通过这种方式，其他人也可以受益于你的发现。

另外，请不要给我发邮件讨论技术问题或勘误。请把它们发在本书的论坛、GitHub 的页面(<https://github.com/azat-co/react-quickly>)或 Stack Overflow 上。其他人也许能比我更快(而且更好)地帮助你。

## 本书论坛

购买《快速上手 React 编程》后，可以免费访问由 Manning 出版社运营的私有 Web 论坛，在这里可以对本书发表评论，提出技术问题，并获得作者和其他用户的帮助。该论坛

的网址是 <https://forums.manning.com/forums/react-quickly>。也可以在 <https://forums.manning.com/forums/about> 上了解到更多关于 Manning 论坛和行为准则的信息。

Manning 出版社对读者的承诺是提供一个场所，在这里各位读者之间，读者与作者之间，可以碰撞出思想的火花。我们不承诺作者现身的次数，他们对该论坛的贡献是自愿的(并且是无酬劳的)。建议你尽量向作者问一些具有挑战性的问题，以激发他的兴趣！



# 目 录

## 第 I 部分 React 基础

第 1 章 初识 React .....	3
1.1 什么是 React .....	4
1.2 React 解决的问题 .....	5
1.3 使用 React 的好处 .....	6
1.3.1 简单性 .....	6
1.3.2 速度和可测试性 .....	11
1.3.3 生态和社区 .....	12
1.4 React 的缺点 .....	13
1.5 React 如何与 Web 应用集成 .....	13
1.5.1 React 类库和渲染目标 .....	14
1.5.2 单页面应用和 React .....	15
1.5.3 React 技术栈 .....	17
1.6 第一个 React 项目： Hello World .....	18
1.7 测验 .....	21
1.8 小结 .....	21
1.9 测验答案 .....	22
第 2 章 React 起步 .....	23
2.1 内嵌元素 .....	23
2.2 创建组件类 .....	26
2.3 属性 .....	29
2.4 测验 .....	34
2.5 小结 .....	34
2.6 测验答案 .....	34
第 3 章 JSX .....	35
3.1 JSX 是什么？它有什么优点 .....	36
3.2 理解 JSX .....	38
3.2.1 使用 JSX 创建元素 .....	39
3.2.2 在组件中使用 JSX .....	40
3.2.3 在 JSX 中输出变量 .....	41
3.2.4 在 JSX 中使用属性 .....	42

3.2.5 创建 React 组件的方法 .....	46
3.2.6 JSX 中的 if/else .....	47
3.2.7 JSX 中的注释 .....	51
3.3 使用 Babel 设置 JSX 转译器 .....	51
3.4 React 和 JSX 陷阱 .....	55
3.4.1 特殊字符 .....	56
3.4.2 data-属性 .....	56
3.4.3 style 属性 .....	57
3.4.4 class 和 for .....	58
3.4.5 布尔类型的属性值 .....	58
3.5 测验 .....	59
3.6 小结 .....	59
3.7 测验答案 .....	59
第 4 章 与状态交互 .....	61
4.1 什么是 React 组件的状态 .....	62
4.2 使用状态 .....	63
4.2.1 访问状态 .....	63
4.2.2 设置初始状态 .....	65
4.2.3 更新状态 .....	67
4.3 状态和属性 .....	70
4.4 无状态组件 .....	71
4.5 有状态组件和无状态组件 .....	73
4.6 测验 .....	77
4.7 小结 .....	77
4.8 测验答案 .....	78
第 5 章 React 组件生命周期 .....	79
5.1 React 组件生命周期事件的 全景视图 .....	80
5.2 事件的分类 .....	80
5.3 实现生命周期事件 .....	82
5.4 执行所有事件 .....	84
5.5 挂载事件 .....	86
5.5.1 componentWillMount() .....	87