Web开发经典丛书

Redux in Action

Redux 实战

[美] 马克•加罗(Marc Garreau)
威尔•福罗(Will Faurot) 著
黄金胜 王冬阳 熊建刚 译

MANNING

清华大学出版社

Web 开发经典丛书

Redux 实战

[美] 马克·加罗(Marc Garreau)
威尔·福罗(Will Faurot) 著
黄金胜 王冬阳 熊建刚 译

清华大学出版社
北 京

Marc Garreau, Will Faurot
Redux in Action
EISBN: 978-1-61729-497-6

图书在版编目(CIP)数据

Redux 实战 /(美)马克·加罗(Marc Garreau)，(美)威尔·福罗(Will Faurot) 著；黄金胜，王冬阳，熊建刚 译. —北京：清华大学出版社，2019
(Web 开发经典丛书)
书名原文：Redux in Action
ISBN 978-7-302-53033-6

Ⅰ. ①R… Ⅱ. ①马… ②威… ③黄… ④王… ⑤熊… Ⅲ. ①JAVA 语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 094462 号

责任编辑：王　军
封面设计：孔祥峰
版式设计：思创景点
责任校对：牛艳敏
责任印制：沈　露

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
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市吉祥印务有限公司
经　　销：全国新华书店
开　　本：170mm×240mm　　印　　张：18　　字　　数：363 千字
版　　次：2019 年 7 月第 1 版　　印　　次：2019 年 7 月第 1 次印刷
定　　价：68.00 元

产品编号：081804-01

译 者 序

2015 年年中，我开始系统学习 React 与 Redux 技术栈。当时，前端技术领域的新技术层出不穷，React、Redux 等技术方兴未艾。

2012 年，我开始从后端开发转向前端开发工作，有幸见证了前端领域的发展、迭代与兴起。最令我感慨的是前端技术日新月异的发展速度，相信很多开发人员也有同感。因为我在工作中，学习过、也用到过很多前端技术。因此，亲身经历了很多前端技术的兴衰与淘汰。

由于前端技术的快速发展，前端开发人员也需要不断学习，而一些新技术刚出现时，学习材料仅有官方文档和一些博客文章，成体系又贴近实际开发工作的书籍资料相对缺乏。相信很多前端开发人员都有这样的经历：起初，踌躇满志、快速地学习了新技术的文档材料，但在公司实际的商业系统开发中，应用起来又困难重重，进展缓慢。

在最初学习 React 与 Redux 技术栈时，我也经历了一些困难和工作压力。彼时，我在国内某一线互联网公司工作。当时组内的前端技术栈以一套内部实现的 MVC 框架及相关衍生工具为主，框架已完成多年，经过几批不同维护者的迭代变更，已经变得陈旧、臃肿且难以学习和维护。所以，我最初学习 React 与 Redux 技术栈的目的之一，就是要在组内形成新技术的储备，并在新项目中试用积累。

然而，由于项目时间紧张且业务相对复杂，整个前端方向的推进遇到了很多困难。其中最大的困难，在于如何合理抽象相关业务，将它们与 Redux 很好结合起来。幸运的是，最终相关项目工作顺利完成。但在这个项目中经历的压力，也引发了我的一些思考：与传统的 MVC 型前端框架相比，Redux 和 React 技术栈在设计理念和实现上都有很大不同！只有深刻理解其精髓，才能不受经验束缚进而合理运用。

时至今日，使用 Redux 和 React 技术栈进行开发已经有几年时间了。期间不断学习思考，也不断在工作中实践，对 Redux 和 React 技术的理解自然也在加深。毫无疑问，Redux 这种前端数据(状态)管理技术的出现与流行，对前端开发领域具有重要意义，标志着前端开发更精细化，也更专业化。

回顾自己的学习历程，有经验也有教训。可以分享的经验是，新技术的学习要和实际的开发实践结合进行，尽可能做到学以致用，在实际中感受并验证所学，这样学

习效果和效用才能最大化。个人的重要教训是，在开始学习尝试一项新技术时，最好能找到合适的学习资料，相对系统全面地学习了解一遍相关的知识点，中间遇到暂时不能理解的内容也没有关系，知道后可以略过继续。这样做的好处是，能够快速建立对新技术知识的整体结构。了解全局的知识结构后，后续的学习实践很多都是对局部知识点认知的加深与强化。对于 Redux 初学者或者有一定经验的开发者，本书就是很好的学习材料。

近两年工作变得异常繁忙，但接到翻译本书的邀请后仍欣然接受。初衷之一也是希望能梳理自己的技术知识点，弥补自己之前学习认知的不足。参与本书翻译的还有我的两位同事：王冬阳和熊建刚。从工作实践角度讲，我们都参与了非常多的基于 React 和 Redux 技术栈的项目开发，有丰富的一线开发经验。但图书翻译这个领域，之前的相关经验有限，对此我们也诚惶诚恐，虽已努力追求更好的翻译效果，但纰漏在所难免，希望读者海涵，具体问题也可以联系沟通，共同讨论学习。

本书能顺利翻译并出版涉及很多人的认真工作，特别感谢清华大学出版社以及相关编辑老师！感谢对我们翻译工作的耐心支持，以及细心的译稿校对工作！希望通过本书，读者朋友能对 Redux 有更好的认知，在技术上取得更大进步！

黄金胜

序　言

Redux 是一款神奇的小工具。如你所见，它并没有涵盖太多的内容。在你喝完一杯咖啡之前，就可以熟悉它的每个方法。

Redux 不仅维护良好，而且还是成品。你可能经常听说 Redux 没有路线图，没有项目经理或看板。提交仍然被添加到 GitHub 存储库，但这些提交通常是对文档或官方示例的改进。

这怎么可能呢？你可能会发现将 Redux 视为架构模式很有帮助。你从 npm 安装的软件包是该模式的一个实现，它为你提供足够多的功能，使你的应用程序能够实现。

关键在于你能用这几个方法完成多少。事实上，Redux 模式可以完全解开一个 JavaScript 应用程序，留下更可预测、更直观和更具性能优势的内容。由于采用相同的模式，开发者工具还可以提供对应用程序状态和数据流前所未有的深入洞察。

但是有什么收获呢？所有软件选择都需要权衡，Redux 也不例外。成本是巨大的灵活性。这可能听起来像另一个优势，但它带来了有趣的挑战。Redux 模式并不由库或任何其他工具严格执行，并且不能希望这个小的软件包可以教育或指导开发者有效地使用此模式。

最后，开发人员可以找到自己的方式。这就解释了为什么 GitHub 存储库中的文档行数远远超过实现代码的行数。与官方文档一样出色的是，开发人员通常会从网络内外的分散资源中收集上下文和最佳实践：博客文章、书籍、推文、视频和在线课程等。

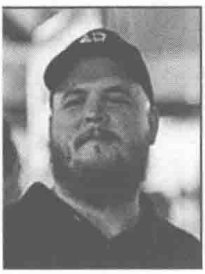
Redux 所允许的灵活性还带来了丰富的附加组件生态系统：选择器、增强器和中间件等。你将很难找到使用完全相同工具集的两个 Redux 应用程序。虽然每个项目都可以根据自己的独特需求定制工具，但对于新引入的开发人员来说，这可能是产生困惑的根源。Redux 新手经常发现，当他们不仅要掌握 Redux，而且还要考虑补充包的复杂性时，他们会沿着一条具有挑战性的学习曲线前进。这是我们想要撰写本书的主要原因：将我们的个人经验和知识从几十个不同的来源提炼成一个整洁、易用的包。

我们相信本书的真正价值在于能够指导你一步步地体验丰富的 Redux 生态系统。

这并不是对所有补充工具的全面看法。相反，我们选择了一些最常见的附加组件，这些附加组件可能会在其他资源中看到并且足够强大，足以应对任何客户端项目。请带上本书，快乐地阅读！我们很高兴你选择与我们共度美好时光。

作者简介

Marc Garreau 是以太坊(Ethereum)基金会 Mist 核心团队的一名开发人员，他长期思考和研究 Mist 浏览器中的应用状态管理。在这之前，他在 Cognizant 和 Quick Left 咨询公司使用 Redux 设计和开发应用程序。他撰写了许多流行的 Redux 博客文章，并在丹佛地区的几个 JavaScript 会议上发表过演讲。



Will Faurot 是 Instacart 的一名全栈开发人员，他在 Instacart 从事多个面向用户产品的开发。他酷爱所有关于前端的内容，擅长用 React 和 Redux 构建复杂的用户界面。在过去的生活中，他教过网球，还录制过复古和蓝草音乐。如果你在海湾区域的一个安静夜晚仔细聆听，可能会听到他弹奏班卓琴的声音。

致　谢

撰写一本书是一项艰巨的任务。有很多人对这个过程至关重要，无论是直接的还是间接的，将他们全部署名可能需要占用本书太多篇幅。本书之所以得以顺利出版，是因为站在巨人的肩膀上。

强大的社区是所有成功软件的基础。Redux 社区非常强大，我们感谢所有在博客中分享自己技能的人，在 GitHub 上帮助 Redux 用户处理问题的人，以及在全球众多 Redux 用户经常访问的在线平台上回答问题的人。

首先最重要的是，如果没有 Redux 的创作者 Dan Abramov 和 Andrew Clark，本书是不可能出版的。除了花时间研究和实现 Redux 外，他们在过去几年里花了无数小时来支持开发人员。我们还要感谢 Redux 的现有维护者 Mark Erikson 和 Tim Dorr。除了定期维护之外，例如，回答问题和合并代码，他们还自愿在几个不同的平台上花时间。这些人一起为本书做了大量的研究，没有他们就没有本书。无论是权衡最佳实践、编写文档，还是向好奇的开发人员提供反馈，他们的价值不可估量。我们感谢他们。

感谢 Manning 的整个团队，包括我们所有的编辑，感谢他们的指导和支持。特别要感谢 Ryan Burrows 提供的宝贵反馈，帮助我们改进了本书的代码，以及 Mark Erikson 花时间整理了一篇精彩的前言。我们还想感谢那些花时间阅读和评论本书的评论家：Alex Chittock、Clay Harris、Fabrizio Cucci、Ferit Topcu、Ian Lovell、Jeremy Lange、John Hooks、Jorge Ezequiel Bo、Jose San Leandro、Joyce Echessa、Matej Strasek、Matthew Heck、Maura Wilder、Michael Stevens、Pardo David、Rebecca Peltz、Ryan Burrows、Ryan Huber、Thomas Overby Hansen 和 Vasile Boris。感谢大家。

特别感谢我们的 MEAP 读者和论坛参与者，他们的反馈和鼓励对本书的顺利出版至关重要。

—Marc Garreau 和 Will Faurot

首先感谢我的妻子 Becky，她做出了很大牺牲：和写书的人一起生活。我保证我可能不会再写书了。感谢我的家人真实地反映了我内心的兴奋，即使我写的书是关于小毛虫的。感谢我的朋友鼓励我，帮助我战胜冒名顶替症，并给我提供健康的娱乐方式。更要感谢 Jeff Casimir、JorgeTéllez、Steve Kinney、Rachel Warbelow、Josh Cheek

和 Horace Williams，他们为 Will 和我在这个行业打开大门。感谢 Ingrid Alongi 和 Chris McAvoy 帮助我在职业生涯里塑造情感技术领导力。

—Marc Garreau

首先感谢我的父母，没有你们给予的指导、热情和鼓励，我就不会成功。你们帮助我意识到这样的事情是可能的。你们教我如何相信自己。谢谢。

感谢我的家人，感谢你们的支持与关爱。

感谢 Instacart 的每个人，你们通过提供反馈或讨论想法为我提供帮助。特别要感谢 Dominic Cocchiarella 和 Jon Hsieh。

最后，感谢 Lovisa Svallingson、Alan Smith、Allison Larson、Gray Gilmore、Tan Doan、Hilary Denton、Andrew Watkins、Krista Nelson 和 Regan Kuchan，你们在整个写作过程中提供了非常宝贵的反馈和鼓励。你们是我遇见的最好的朋友和软件知己。

—Will Faurot

前　言

自 2015 年年中发布以来，Redux 引起了 JavaScript 世界的关注。从它作为会议演示的概念验证和“只是另一个 Flux 实现”标签的简单开端，已经发展成为 React 应用程序中使用最广泛的状态管理解决方案。它也被 Angular、Ember 和 Vue 社区采用，并启发了许多模仿品和衍生产品。

我最喜欢引用的一句话是，“Redux 是一个通用框架，它提供了足够结构化和足够灵活性的平衡。因此，它为开发人员提供了一个平台，可以让他们为自己的用例构建自定义状态管理，同时能够重用图形化调试器或中间件之类的东西。”[1]的确，Redux 提供了一组基本的工具以供使用，并概述了组织应用程序更新逻辑的一般模式，最终由你来决定如何围绕 Redux 构建应用程序。你可以设计应用程序的文件结构，编写 reducer 逻辑，连接组件，并确定要在 Redux 上使用多少抽象。

Redux 的学习曲线有时会很陡峭。对于来自面向对象语言的大多数开发人员来说，函数式编程和不可变性是不熟悉的概念。编写另一个 TodoMVC 示例并没有真正展示 Redux 的好处，也不能解决构建“真实”应用程序的问题。但最终的收益是值得的。能够清楚地追踪应用程序中的数据流并了解特定状态变更的位置/时间/原因/方式是非常有价值的，并且良好的 Redux 使用方式最终会让代码在长期内更易于维护和可预测。

我大部分时间都是 Redux 维护人员，通过回答问题、改进文档和撰写教程博客来帮助人们学习 Redux。在这个过程中，我看过数百种不同的 Redux 教程。有鉴于此，我非常乐意推荐将《Redux 实战》一书作为学习 Redux 的最佳资源之一。

通过《Redux 实战》一书，Marc Garreau 和 Will Faurot 写了我希望自己写的 Redux 书籍。它非常全面、实用，并且可以很好地为开发现实世界中的 Redux 应用程序讲授许多关键的主题。我特别欣赏本书所涵盖的领域，并不总是有明确的答案，如构建项目，通过列出利弊，让读者知道这是一个他们可能不得不自己决定的领域。

在当今快速发展的编程世界中，没有一本书可以包罗有关工具的所有知识。但是，

1　来自 Facebook 工程师 Joseph Savona(https://github.com/reactjs/redux/issues/775#issuecomment-257923575)。

《Redux 实战》将为你打下坚实的基础和对 Redux 基础知识的理解——各部分如何结合，如何将这些知识用于实际应用程序，以及在何处寻找更多信息。我很高兴看到本书出版，并期待你加入 Redux 社区！

Mark Erikon
Redux 核心维护者

关 于 本 书

Redux 是一个状态管理库，旨在简化复杂用户界面的构建。它通常与 React 一同使用，在本书中亦如此，但它也在其他前端库(如 Angular)中日趋流行。

在 2015 年，React 生态系统亟待 Redux 的到来。在 Redux 之前，Flux 架构模式是一次令人兴奋的突破，世界各地的 React 开发人员都在尝试实现。数十个库受到空前的关注和使用。最后，兴奋变得疲惫。React 应用程序中的状态管理方式让人眼花缭乱。

Redux 发布后立即开始飞速发展，并很快成为最受推荐的受 Flux 启发的库。它使用单一数据源，着眼于不可变性，以及令人惊叹的开发体验，都证明 Redux 是一个更简单、更优雅、更直观的解决方案，解决了现有 Flux 库所面临的大多数问题。对于复杂应用程序中的状态管理，虽然仍有多个选择，但是对于那些喜欢类似 Flux 模式的开发人员来说，Redux 已经成为默认选择。

本书将带你了解 Redux 的基本原理，然后再继续探讨强大的开发者工具。我们将一起逐步开发一个任务管理应用程序，在该应用程序中，将探索真实世界中 Redux 的使用，并关注最佳实践。最后，将回到测试策略和构建应用程序的各种约定。

本书读者对象

读者应该熟悉 JavaScript(包括 ES2015)，并且至少掌握一些 React 的基础知识。不过，我们也了解到，许多开发人员都是在接触 React 的同时投身于 Redux。我们尽了最大努力来适应这类读者，我们相信他们通过一点额外的努力就可以阅读本书。虽然如此，我们的建议是在阅读这本书之前牢固地掌握 React 的基础知识。如果没有任何 React 开发经验，请考虑阅读《React 实践》和《快速上手 React 编程》。

本书的组织结构：路线图

本书共包含 12 章和 1 个附录。

第 1 章介绍 Redux 的诞生环境及其产生原因。你将了解 Redux 是什么，它的用途，

以及何时不应该使用它。该章总结了几种类似 Redux 的状态管理方法。

第 2 章直接开始开发第一个 React 和 Redux 应用程序。新功能的开发过程简单而快捷，如同旋风。你将从更高的角度了解其中每个角色：action、reducer 和 store 等。

第 3 章介绍强大的 Redux 开发者工具。Redux 开发者工具是 Redux 的最大亮点之一，本章将解释原因。

第 4 章最终将副作用引入从第 2 章开始的示例中。你将设置本地服务器，并在 Redux 模式中处理 API 请求。

第 5 章深入介绍一个更高级的特性：中间件。你将了解中间件在堆栈中的位置、它的功能以及如何构建自定义中间件。

第 6 章探讨用于处理更复杂的副作用的高级模式。你将学习如何使用 ES6 generator 函数，然后将学习如何使用 saga 管理长期运行的流程。

第 7 章重点介绍 Redux store 和视图之间的连接。你将了解选择器函数如何工作，然后使用 reselect 库实现一个健壮的解决方案。

第 8 章讨论一个常见问题——在 Redux store 中如何最好地构造数据。你将回顾到目前为止本书使用过的策略，然后探索另一种方法：范式化。

第 9 章回头讨论关于测试的所有内容。你将了解到一些流行的测试工具，如 jest 和 Enzyme，以及用于测试 Redux action、reducer 和 selector 等的策略。

第 10 章介绍如何保持应用程序精简和高效，涵盖了性能分析工具、React 最佳实践，以及用于提高性能的特定 Redux 策略。

第 11 章介绍组织 Redux 应用程序代码的几种策略。Redux 不介意内容的位置，所以你将学习已经成熟的常用惯例。

第 12 章提醒你，Redux 不仅仅可以管理 React Web 应用程序的状态。你将快速了解 Redux 可以在移动端、桌面端和其他 Web 应用程序环境中扮演的角色。

附录提供环境设置和工具安装的说明。

关于代码

大多数代码示例都是针对本书的示例应用程序Parsnip 的。这些示例以代码清单的形式呈现，其中多数都添加了注释，以保证代码清晰并说明某些代码选择背后的原因。正文中出现的代码以等宽字体显示，如 `like this`。

本书示例的源代码可从出版商网站 https://www.manning.com/books/redux-in-action 或 https://github.com/wfro/parsnip 下载。

一键安装脚本可用于 Mac OS X、Linux 和 Windows，它们与本书其他源代码放在一起。有关入门说明，请参阅附录。

软件需求

大多数代码示例，尤其是与示例应用程序相关的示例，都需要 Web 浏览器。我们推荐使用 Chrome，它完美支持 React 和 Redux 开发者工具。

我们用 create-react-app 构建示例应用程序，这不是严格要求，但强烈推荐。这是构建现代 React 开发环境最愉悦的方式。

本书使用以下 Create React App 和 Redux 版本：

- Redux：3.7.2。
- Create React App：1.0.17。

图书论坛

购买本书的读者能够免费访问 Manning Publications 运营的私人 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。要访问论坛，请访问 https://forums.manning.com/forums/redux-in-action。你还可以在 https://forums.manning.com/forums/about 上了解有关 Manning 论坛和行为规范的更多信息。

其他线上资源

Redux 社区在几个不同的平台上非常活跃。我们建议使用以下所有资源来了解更多信息，帮助巩固概念，并提出问题：

- Reactiflux 是讨论 React 和 Redux 的主要聊天室，位于 https://www.reactiveFlux.com/。
- Redux 文档中的术语表。尽管 Redux 有种种优点，但它确实需要相当数量的术语。特别是对于初学者而言，返回去查阅术语表是非常有用的。有关详细信息，请参阅 https://github.com/reactjs/redux/blob/master/docs/ Glossary.md。
- Redux 官方文档位于 https://redux.js.org/。
- Mark Erikson 的“实用 Redux”系列博客。对于中高级 Redux 用户来说，这是更好的选择，因为它提供了更多对Redux使用的深入探索。有关详细信息，请参阅 http://blog.isquaredsoftware.com/2016/10/practical-redux-part-0-introduction/。

目　录