Web开发经典丛书

Pro React

React开发实战

使用React以组合方式构建复杂的前端应用程序

[美] Cássio de Sousa Antonio 著
杜伟 柴晓伟 涂曙光 译

清华大学出版社

Web 开发经典丛书

React 开发实战

[美] Cássio de Sousa Antonio 著

杜伟 柴晓伟 涂曙光 译

清华大学出版社

北 京

Pro React

By Cássio de Sousa Antonio

EISBN：978-1-4842-1261-5

北京市版权局著作权合同登记号　图字：01-2016-8577

图书在版编目(CIP)数据

React开发实战/(美)卡西奥·德·宗萨·安东尼奥(Cássio de Sousa Antonio) 著；杜伟，柴晓伟，涂曙光 译. —北京：清华大学出版社，2017

(Web 开发经典丛书)

书名原文：Pro React

ISBN 978-7-302-46197-5

Ⅰ. ①R… Ⅱ. ①卡… ②杜… ③柴… ④涂… 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 019984 号

责任编辑：王　军　韩宏志
装帧设计：孔祥峰
责任校对：曹　阳
责任印制：宋　林

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 刷 者：清华大学印刷厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：19.5　　**字　　数**：451 千字
版　　次：2017 年 3 月第 1 版　　**印　　次**：2017 年 3 月第 1 次印刷
印　　数：1~4000
定　　价：58.00 元

产品编号：071425-01

译 者 序

本书的三位译者都有基于Microsoft SharePoint Server平台进行Web应用程序开发的经验。在SharePoint平台(特别是从SharePoint 2007 这个版本开始)上进行Web应用程序开发，主要使用的还是服务器端的开发技术，主要编程语言和框架是C#和ASP.NET。但从那时起，我们就已经不断在自己的Web应用程序中使用越来越多的JavaScript代码，并尝试着开始将(当时还算是比较新潮的)jQuery和其他前端库引入到自己的应用程序中。我们当时就预料到，JavaScript在未来必然将占据越来越重要的地位，对于Web应用程序开发而言，越来越多的服务器端技能都会被前端技能所替代。

但JavaScript的发展之迅猛，仍然超出了我们的意料。时至今日，不但JavaScript已经成为前端工程师的一件利器，由于Node.js的出现，JavaScript还逐渐渗透到服务器端(甚至桌面端)之上。前端工程师们开始尝试基于JavaScript来构建越来越复杂的Web应用程序，这给JavaScript提出了越来越大的挑战，随之而来的，就是异常活跃的JavaScript社区和不断涌现的JavaScript前端框架。这些框架的目标，都是为了以一种更简明直观的方式，将Web应用程序的复杂性进行隔离，然后以一种模块化方式，单独开发和推进整个应用程序中相对独立的部分。

React是众多JavaScript前端框架中的后起之秀。它诞生于Facebook，被Facebook开源之后，便迅速得到众多前端工程师的青睐。React社区也随之蓬勃发展，各种React的插件库不断地由社区成员开发出来，其中的佼佼者(比如本书中所介绍的React Router)也得到了大量关注并被广泛使用。

作者以一种循序渐进、深入浅出的方式，讲述 React 的方方面面。读者在阅读和学习的过程中，应该能很轻松地跟随着作者的节奏，逐个了解 React 的特性，并通过一个贯穿全书的示例应用程序，了解如何规划和构建出一个越来越复杂的 Web 应用程序。书中的示范代码也秉承了简洁易懂、突出重点的风格，能帮助读者迅速了解一个 React 应用程序的代码风格应该是什么样子。

巧合的是，在翻译本书的过程中，Microsoft 在 SharePoint 平台上发布了一个新的开发框架：SharePoint Framework。SharePoint Framework 是一个完全基于前端技术的 SharePoint 平台开发框架，其中所使用的默认 UI 前端库，正是 React。

本书全部章节由杜伟、柴晓伟、涂曙光翻译，参与翻译的还有梁祝权、钟凤华、毛士之、张杉杉、张文旭，在此一并表示感谢！最后，让我们向读者稍微介绍一下自己。杜伟(微博账号：@Erucy)，软件工程师与日本动漫迷，现在一家互联网创业公司担任技

术总监，日常使用 JavaScript 和 Cordova 开发跨平台的 Web 和 Mobile 应用。柴晓伟(微博账号：@WindieChai)，软件工程师，现在一家大型招聘网站担任技术执行总监，日常使用 JavaScript、Node 和 C#开发各种 Web 前后端以及 Mobile 应用。涂曙光(微博账号：@kaneboy)，软件工程师，现在一家金融行业公司担任架构师，日常使用 C#/C++开发证券交易系统，同时还在使用 JavaScript 开发一些 Web 应用程序。我们都是微软最有价值专家，并已合作翻译过多本技术书籍。

Happy Reacting!

译者
作于北京

作者简介

20年前，Cássio de Sousa Antonio使用一台Sinclair Spectrum开启了他的编程生涯，随后在巴西和美国成长为一名软件工程师和技术经理。他参与开发过微软、可口可乐、联合利华和汇丰银行等大公司的项目。他的初创公司于2014年末被收购。Cássio目前担任技术顾问。你可在Twitter上关注他(@cassiozen)。

技术审阅者简介

Jack Franklin是一位讲师、作者和技术作家，他大多数的时间都在编写或讨论JavaScript。他在Pusher公司担任开发者布道师，他还是一名积极的开源贡献者。他是React的狂热粉丝，并在www.javascriptplayground.com上撰写了大量关于JavaScript的文章。你可在Twitter上找到他：@Jack_Franklin。

Tyler Merry 是 Universal Mind 公司的 UX 技术专家，他专注于缩短创意和实现之间的差距。Tyler 通过试验来筛选处理所有问题。他坚信最快且最精准的解决方案就是积极地进行各种试验和非正式的测试。

Tyler 曾就职于可口可乐、索尼、辉瑞、宝洁、福特和 Vail Resorts，在工作中，他认识到了精确度和沟通的价值。他就职于一些早期的初创公司，去帮助它们增加迭代、速度和效率所带来的价值。

除了追赶最新的 Web 和 UX 趋势，Tyler 还把时间花在“少于四个轮子的车辆”上(自行车、摩托车和独轮车)，他也会去学习任何能够吸引他的技能，如编织、摄影或杂耍。

致　谢

我要感谢我的父母：Sergio 和 Dete，是你们给了我自由、独立和爱。

还要特别感谢 Apress 的编辑人员，感谢你们在这个项目中给予我的信任，以及给予我的所有指导和耐心。

前　言

React 是一个用来创建组合式 Web 应用程序的开源库，由 Facebook 维护。自从公开发布后，这个库迅速风靡全球，并且围绕着它产生了一个生机勃勃的社区。

本书将涵盖 React 库的各个细节，并将讨论基于组合式模型来创建 Web 组件接口的最佳实践。React 库本身并不大，所以本书同时涵盖了 React 生态系统中的一些工具和库(例如 React Router 和 Flux 架构)，以便为读者提供创建完整应用程序所需的足够知识。

书中对每个主题的讲解都简洁明了，你将逐一了解到你需要掌握的各个细节，从而学会真正有效地使用它们。本书对 React 中最重要的那些特性的讲解，言简意赅、由浅入深，每个章节中还详细说明实际开发中可能面临的常见问题，并告诉你如何避免它们。

本书概要

第 1 章提供了大量的信息来让你起步，介绍基本的 React 配置，并让你能全面了解在 React 中如何组织用户界面。

第 2 章将深入介绍 JSX(这是 React 用来在 JavaScript 中声明组件标记的 JavaScript 语法扩展)，同时展示如何使用 React 的事件系统，以及 React 的虚拟 DOM 实现。

第 3 章讲述了如何通过使用组件的方式来创建一个完整的应用程序。你将学习 React 应用程序的数据如何在组件间流动，并深入了解组件(包括嵌套组件、公开一个 API、props、state 等知识)。

第 4 章讲述了如何为用户创建丰富的用户体验。你将学习如何在 React 的插件 CSSTransitionGroup 的帮助下实现动画效果，以及使用一个名为 React DnD 的外部库来实现拖放功能。

第 5 章讲述了路由功能。你将学习如何使用 React 社区中被广泛使用的 React Router，来管理 URI 和设置应用程序的访问点。

第 6 章向读者展现了 Flux 架构。你将学习这个架构的细节，它解决了哪些问题，以及如何将它集成到一个 React 应用程序中。

第 7 章讲述了性能调优。在该章，你将学习如何度量应用程序的性能指标。然后你将了解如何优化代码，使应用程序有更好的性能表现。

第 8 章讲述了 React 同构应用程序(或者称 React 通用应用程序，也就是在服务器上渲染 React)。这种技术可以实现更好的性能、搜索引擎优化和优雅降级(如果浏览器禁用

了本地 JavaScript，应用程序也能正常工作)。

最后，第 9 章讲述了测试。你将学习如何使用 React 的 Test Utils 来测试组件。还将学习 Jest，这是一个由 Facebook 创建的、最适合用来测试 React 项目的测试框架。

本书读者对象

本书主要面向使用诸如 jQuery/Backbone/Angular 创建前端应用程序且拥有一些经验的中级水平 JavaScript 开发人员，他们需要了解更好的工具和更多知识，来解决构建复杂前端应用程序过程中所遇到的越来越多的常见问题。

源代码

本书中所包含的示例项目的源代码，位于 Apress 网站上的 Source Code 区域。请访问 www.apress.com，单击 Source Code，然后查找本书的书名(请使用英文书名 *Pro React* 来进行搜索)，就可以找到它们。另外，本书所有的示例代码和一些额外的实用性代码，都可在本书的 GitHub 页面上找到，网址是 pro-react.github.io。此外，可访问 www.tupwk.com.cn/downpage，输入中文书名或中文 ISBN 来下载源代码，也可以扫描本书封底的二维码下载资料。

联系作者

感谢你购买本书。我希望你能享受阅读本书的乐趣，并从中获取有价值的信息。欢迎你个人针对本书内容与源代码给我提供反馈、问题和评论。你可以通过 proreactbook@gmail.com 联系我。

祝你好运！期待着你的 React 应用程序的诞生！

目　录

第 1 章

React 入门

React 是由 Facebook 创建的一个开源项目。它提供了一种在 JavaScript 中构建用户界面的全新方式。自从它公开发布后，这个库迅速风靡全球，并且围绕着它培育了一个生机勃勃的社区。

通过阅读本书，你将掌握在项目中使用 React 所需的方方面面。因为 React 只关注 UI 界面的渲染，而不会对应用程序的其他模块所使用的技术做任何假设，所以本书同时也将介绍能匹配 React 模式的路由(Routing)和应用程序架构。

在本章中，我们将从一个较高的层面讲述一些主题，以便你能尽快开始创建应用程序。这些主题包括：

- React 的完整定义，以及优点概览
- 如何使用 JSX，这是一个在 React 中用来渲染 UI 的 JavaScript 语法扩展
- 如何创建包含属性和状态的 React 组件

1.1 开始学习之前

React 针对的是现代风格的 JavaScript 开发生态系统。为能亲自尝试本书中的代码示例，你需要安装 Node.js 和 npm。此外，还需要熟悉函数式 JavaScript 语法风格和这门语言最新的特性，如箭头函数(arrow functions)和类。

1.1.1 Node.js 和 npm

JavaScript 自诞生之日起就运行在浏览器上，但是通过 Node.js 的开源命令行工具，可以使 JavaScript 运行在你的本地计算机和服务器上。与 npm(Node Package Manager)一道，Node.js 已经成为一项在本地计算机上进行 JavaScript 应用程序开发的极有用工具，它使得开发人员可以创建脚本来运行任务(例如，复制和移动文件，或是启动一个本地开发服务器)，以及自动下载应用程序所依赖的组件。

如果你尚未安装 Node.js，现在就下载它的 Windows、Mac 或者 Linux 版本，将其安装到你的计算机上。下载地址为 https://nodejs.org/。

1.1.2 JavaScript ES6

JavaScript 是一门自诞生起多年一直在不断进化的语言。最近，JavaScript 技术社区已经认同了一组新的语言特性。有一些最新的浏览器已经能够支持这些特性，React 社区也广泛地使用了这些新的特性(例如，箭头函数、类、展开操作符)。React 同时鼓励在 JavaScript 代码中使用函数式编程模式，所以你也需要熟悉在 JavaScript 中函数和上下文是如何工作的，这样你才能了解 map、reduce 和 assign 等方法。如果你对这些细节感觉有些模糊，可参阅 Apress 网站(www.apress.com/)和本书的 GitHub 页面(http://pro-react.github.io/)上有关这些主题的在线附录说明。

1.2 定义 React

为清楚地说明 React 究竟为何物，我将对它做如下定义：

React 是一个使用 JavaScript 和 XML 技术(可选)构建可组合用户界面的引擎。

下面对 React 定义的每个部分再详加说明：

React 是一个引擎：React 的网站将它定义为一个库，但是我觉得使用“引擎”这个词更能体现出 React 的核心优势：用来实现响应式 UI 渲染的方式。React 的方式是将状态(在一个给定的时间点，所有用来定义应用程序的内部数据)从展现给用户的 UI 中分离出来。在 React 中，你可以声明如何将应用程序的状态表现为 DOM 的虚拟元素，然后自动更新 DOM 以反映出状态的变化。

“引擎”这个术语首先被 Justin Deal 用来描述 React，因为他觉得 React 渲染 UI 界面的方式和游戏引擎渲染的工作方式十分相似(https://zapier.com/engineering/react-js-tutorial-guide-gotchas/)。

创建可组合用户界面：减少创建和维护用户界面的复杂性一直是 React 的核心目标。React 拥抱了这样一种理念：将 UI“打散”成易于重用、扩展和维护的组件与自包含的、关注特定用途的构件(building blocks)。

使用 JavaScript 和 XML 技术(可选)：React 是一个可用于浏览器、服务器和移动设备之上的纯 JavaScript 库。如你将在本章中所见，React 有一种可选的语法来让你可以使用 XML 来描述 UI。一开始你可能会对这种语法感到有些陌生，但使用 XML 对于描述用户界面其实有诸多优点，包括：XML 是声明性的，很容易通过 XML 观察元素之间的关系，也很容易使 UI 的整体结构可视化。

1.3 React 的优点

市面上有许多的 JavaScript MVC 框架。Facebook 有什么理由还要创建 React？你有什么理由要使用 React？下面的三节内容将探索 React 的一些优点，从而回答这两个问题。

1.3.1 简单易学的响应式渲染

在 Web 开发的早期，那时还根本没有单页应用程序这个概念，用户每次在页面上进行一次交互(比如单击了一个按钮)，就算新的页面只和原有页面仅有一点不同，服务器都会将一整页新的页面发送回客户端浏览器。从用户的角度看，体验会非常糟糕，不过程序员倒是很容易规划出用户在某一时刻能看到哪些内容。

单页应用程序持续地从服务器获取新数据，并在用户进行交互时变换 DOM 上面的部分内容。在用户界面逐渐变得复杂时，应用程序也要实现越来越复杂的逻辑来检验应用程序的当前状态，并对 DOM 及时进行所需的修改。

许多 JavaScript 框架(特别是那些在 React 出现之前的框架)都使用了数据绑定技术，来处理这种日益增加的复杂性并保持用户界面和应用程序状态的同步，但是数据绑定在可维护性、可扩展性和性能上，都有一些缺陷。

响应式渲染比传统的数据绑定技术要更容易使用一些。它让我们用一种声明方式，来定义组件的外观和行为。当数据发生变化时，React 在概念上会重新渲染整个用户界面。

当然，每次在状态数据发生变化时就真的重新渲染整个用户界面，在性能上是不可接受的，React 使用了一种存在于内存中的轻量级 DOM 表示法，这被称为“虚拟 DOM”。

处理这种内存中的虚拟 DOM 要比处理真正的 DOM 更快、更有效。当因为用户的交互或者数据的获取而导致应用程序的状态发生改变时，React 快速地将 UI 的当前状态与期望的状态进行比较，然后计算出要对真实 DOM 所需进行的最小更改。这种工作方式使得 React 非常快、非常高效。即使在一个移动设备上，React 应用程序也能轻松地达到 60fps 的刷新率。

1.3.2 使用纯 JavaScript 进行面向组件开发

在一个 React 应用程序中，一切都由组件组成，组件就是应用程序中的自包含的、关注特定用途的基础构件。基于组件来开发应用程序使用了一种“分而治之”的途径，来避免在某个地方有太高的复杂性。由于组件可以组合起来，每个组件都可以尽量保持小巧，通过将更小的组件组合在一起，创建复杂的包含更多功能的组件就变得简单了。

不同于使用特定的模板语言或是传统 Web 应用程序 UI 中用到的 HTML 标记，React 组件由普通的 JavaScript 写就。使用普通 JavaScript 代码编写组件有一个很好的理由：模板限定了你可用来构造 UI 界面时能使用到的功能特性。React 则是使用一门功能完整的编程语言来渲染界面，这相对于使用模板具备很大的优势。

另外，通过使组件成为自包含的、并在相关视图逻辑中使用一个统一的标记，React 组件实现了关注点的分离。在Web开发的早期岁月，关注点的分离是通过在不同部分使用不同的语言来强制实现的：内容结构使用HTML，样式使用CSS，逻辑行为使用JavaScript。这种方法在问世之初工作得很好，因为那个时候Web页面还是静态呈现的。但是现在用户界面变得更有交互性、更复杂，显示逻辑和HTML标记不可避免地被绑定在一起；标记、样式和JavaScript之间的分离变成了仅是技术上的分离，而非关注点的分离。

React 假设显示逻辑和 HTML 标记是高度粘合的；它们同时用来实现 UI 的展现，并通过为每个关注点创建离散的、良好封装的、可重用的组件，来鼓励实现关注点的分离。

1.3.3 灵活的文档模型抽象表现

React 内置了自己的一个 UI 轻量级表现模型，以抽象出 UI 底层的文档模型。这种方式最值得一提的优点，就是不论在 Web 页面，还是在原生的 iOS 和 Android 界面上，它都可以使用同样的原则来渲染 HTML。这种抽象表现同时带来了下面这些特性：

- 事件在所有浏览器和设备上都会以一种统一、标准的方式，自动地使用代理，来达到行为的一致性。
- 为实现 SEO(搜索引擎优化)和更好的性能，React 组件可在服务器上被渲染。

1.4 创建你的第一个 React 应用程序

现在你已经知道了组件是 React UI 的基础构件，但是组件到底看起来是什么样子的？怎么样才能创建一个组件呢？简单来说，一个 React 组件就是一个带有 render 方法，并且返回组件 UI 描述的 JavaScript 类，如下所示：

```
class Hello extends React.Component {
  render() {
    return (
      <h1>Hello World</h1>
    )
  }
}
```

你也许已经注意到了JavaScript代码中间的HTML标记。之前曾经提到过，React有一种称为JSX的JavaScript语法扩展，可以让我们在代码中直接书写XML(以生成HTML)。

是否使用 JSX 是可选的，但是它已经成为一种被广泛接受的在 React 组件中定义 UI 的标准方案，由于 JSX 使用了具有丰富表达能力的声明式语法，以及这些内容最终会被转换成普通的 JavaScript 函数调用，所以这意味着 JSX 并没有影响 JavaScript 语言原本的语义。

我们会在下一章更详尽地介绍 JSX，但是现在要提醒你注意的是，React 需要一个额外的“转换”步骤(或者说是翻译步骤)，将 JSX 转换成 JavaScript 代码。

在现代的 JavaScript 开发生态系统中，有许多工具可处理这种需要额外转换步骤的情况。下面花一点时间来讨论如何为 React 项目搭建出一套流畅的开发流程。

1.4.1 React 开发流程

想当年，我们都是将所有 JavaScript 代码都写到一个文件里面，手工下载一两个 JavaScript 库，然后将自己的代码和第三方的库统统塞到一个页面上。当然，现在你仍然