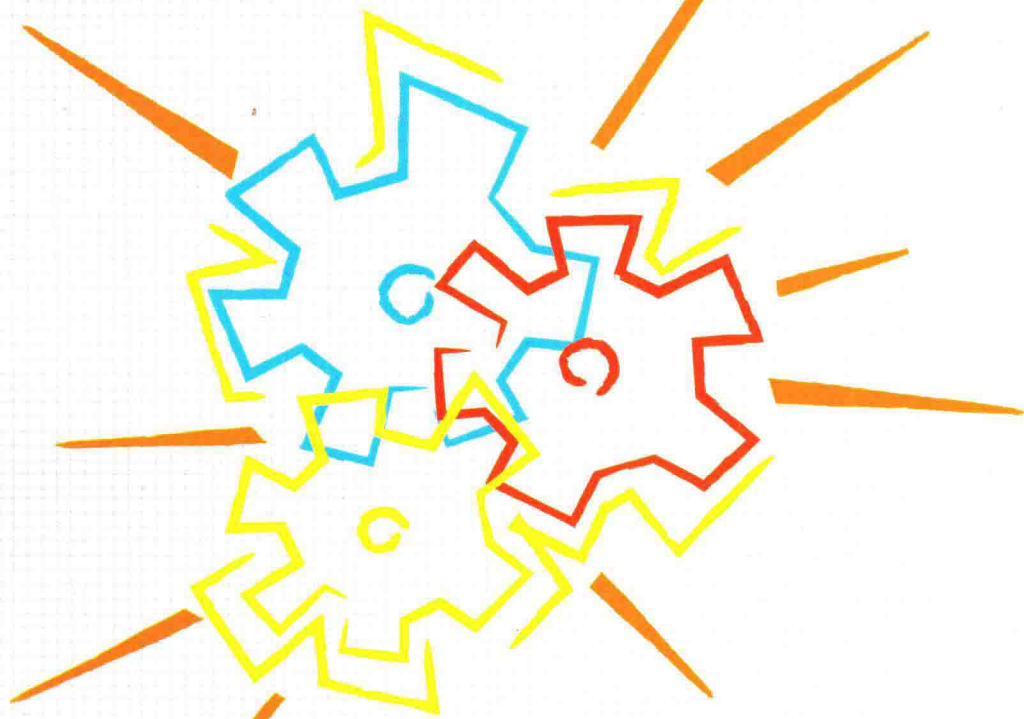


React (第2版)

引领未来的用户界面 开发框架

Developing a React Edge:
The JavaScript Library for User Interfaces, 2e



League of Extraordinary Developers 著
寸志 范洪春 题叶 杨森 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 (第2版)

引领未来的用户界面 开发框架

Developing a React Edge:

The JavaScript Library for User Interfaces, 2e

League of Extraordinary Developers 著

寸志 范洪春 题叶 杨森 译

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

Facebook 的开源框架 React.js, 基于 Virtual DOM 重新定义了用户界面的开发方式, 彻底革新了大家对前端框架的认识, 将 PHP 风格的开发方式迁移到客户端应用开发。其优势在于可以与各种类库、框架搭配使用。本书由多位一线专家精心撰写, 采用全程实例介绍和剖析了 React.js 的方方面面, 适合广大前端开发者、设计人员, 以及所有对未来技术趋势感兴趣者阅读。本书前版作为 React 首作推出之后, React 生态继续蓬勃发展, 技术及社区都在不断推陈出新。本书诸位专家作者适时推出新版, 全面更新示例, 用 ES6 重写代码, 摒弃社区弃用范式, 新增无状态组件、不可变数据、Redux 等热点内容。阅读本书, 不但可以夯实 React 开发基础, 更能全方位紧跟整个 React 生态!

Copyright © 2016 Bleeding Edge Press. All rights reserved. First published in the English language under the title "Developing a React Edge: The JavaScript Library for User Interfaces, 2e" by Bleeding Edge Press, an imprint of Backstop Media.

本书简体中文版专有出版权由 Bleeding Edge Press 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。专有出版权受法律保护。

版权贸易合同登记号 图字: 01-2016-7197

图书在版编目(CIP)数据

React: 引领未来的用户界面开发框架: 第2版 / 卓越开发者联盟 (League of Extraordinary Developers) 著; 寸志等译. —北京: 电子工业出版社, 2016.11

书名原文: Developing a React Edge: The JavaScript Library for User Interfaces, 2e
ISBN 978-7-121-30120-9

I. ① R… II. ①卓… ②寸… III. ①人机界面—程序设计 IV. ① TP311.1

中国版本图书馆 CIP 数据核字 (2016) 第 247610 号

责任编辑: 张春雨

印 刷: 三河市双峰印刷装订有限公司

装 订: 三河市双峰印刷装订有限公司

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: 12.75

字数: 276 千字

版 次: 2015 年 5 月第 1 版

2016 年 11 月第 2 版

印 次: 2016 年 11 月第 1 次印刷

定 价: 69.00 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, 88258888。

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

本书咨询联系方式: (010) 51260888-819, faq@phei.com.cn。

推荐序1

时光回溯。

2011 年我离开 Google 转而加入 Facebook，从事移动互联网（Mobile Web）的核心产品开发工作。

随着时间的推移，工作上逐渐取得了许多有意义的巨大进展，同仁们也都深以此为傲。然而不是所有的事情都进展得特别顺利。其中一个很大的问题与挑战就是因为 HTML5 的技术限制与性能瓶颈，许多产品的开发工作受到了限制。

2012 年 Facebook 公开了一件很多人深有体会却不想说出口的事实，那就是 HTML5 之类的 Web 技术并未成熟到能够担任产品开发工具重任的程度。在很多方面，使用原生代码（native code）开发仍然是必要的选项。

对于很多包括我在内的 Mobile Web 开发者来说，这样的情况是一个让人失望却又不得不接受的事实。

2013 年年初，我离开工作两年多的移动互联网开发部门，转而投入广告部门，从事桌面富客户端（Rich App Client Application）的开发工作。

虽然 Mobile Web 的进展不如预期理想，但此时在 Desktop Web 方面，事情却有了有意思的变化。

当时我参与的新项目主要是要使用一种叫作 React 的新平台技术，将当时广告部门的一个主要产品重构。项目的有趣之处在于，产品的视觉外观与功能将不会也不能有任何变化，但是内部执行的代码将会是以 React 打造的。

由于项目的目标为实际上线且对公司营收有重要影响的产品，所以项目的挑战除了在于应用 React 这门新技术之外，维持产品本身的稳定当然也是不可妥协的目标。

所幸，项目顺利达标，而同仁们也对于 React 这门技术有了更丰富的经验与更强的信心。就连 React 本身也快速吸收众人的回馈，快速演进。

我从事 Web 前端开发工作已经十年，有幸亲身经历众多重大的技术变革与范式转移。我可以负责任也很喜悦地说，作为一门新生技术，React 及其相关工具对于从事 Web 开发的人来说，将会产生巨大且革命性的影响。

虽说 React 初始是为了解决 Facebook 广告部门在产品开发上遇到的很多实际问题，但实际应用的层面却非常广泛。

2015 年 Facebook 也开源了 ReactNative，让 React 能够在 iOS 移动终端执行（对 Android 的平台支持预计为 2015 年年底）。

由于 React 的特殊设计，React 消弥了客户端与服务器端的开发界线。最近的发展则更进一步衍生到 Mobile Native App 与其他非传统 Web（HTML+CSS）的执行环境。

无论你是有多年经验的开发者，或者是刚入门的新人，此时选择 React 都会是一个很好的选择。

React 可以解决很多传统 Web 开发架构碰到的艰难问题，同时由于它是一门新生技术，开发者将更有机会掌握一门强大的开发工具，解决更深入的艰难问题并提升产品开发的质量与境地。

作为一本入门书籍，本书提供基本但足够的范例与介绍，着重在实际的代码与操作应用，可以让读者快速学习 React 的相关知识与技术，并实际打造可执行的程序。

相信对于需要使用 React 开发的人来说，这将会是一本不错的入门参考。

必须要补充的是，目前由于 React 还在 Beta 版本阶段，本书的内容主要是以 v0.12 为主。目前公开的最新版本为 v0.13，书中提到的 API 将会略有差异，细节方面可以在它的官方网站上查询（<https://facebook.github.io/react/blog/2015/02/24/streamlining-react-elements.html>）。

Hedger Wang
Facebook 资深前端工程师

推荐序2

组件化一直是前端领域的圣杯。我至今依旧记得自己初次接触 YUI-Ext 时，被其精致的组件和优雅的设计深深震撼的场景。之后随着 ExtJS 的发布，我在很长时间内都痴迷于探索 ExtJS 深邃的继承层次与架构，并由此进入了前端行业。

ExtJS 作为一个企业级框架，借鉴了 Java 的 Swing 设计，同 Java 一样有着教科书般的学院风派，也同 Swing 一样注定曲高和寡。在快速变化的互联网领域，ExtJS 犹如一条大船行驶在激流中，每一次调头都非常艰难。同时代的不少互联网企业也开源了自己的前端类库，包括 YUI、Closure Library、KISSY、Arale 等，它们同样有着不错的组件设计，但思路和 ExtJS 并无显著不同，只不过更加轻量化。

传统组件化的特点是把组件和原生 DOM 节点的渲染割裂起来，要么如 ExtJS 一样抛弃原生 DOM 节点，要么就在原生 DOM 节点渲染后再渲染自定义组件。现代的组件架构鼓励原生 DOM 节点和自定义组件的统一渲染融合，比如 React 以及未来的 Web Components 规范。

React 最为人称道的是，它是一个专注于组件架构的类库。API 很少，理念也很简单，使用 React 可以非常快速地写出和原生 DOM 标签完美融合的自定义组件（标签），并且能够高效渲染。而想要真正使用好 React，我们必须跳出以往的思维，拥抱 React 的理念和思想，比如状态、虚拟 DOM、组合优于继承、单向数据流等。React 的简单抽象和专注，使得 React 可以更容易与其他各种技术结合。因为 React 的简单抽象，我们

终于可以脱离浏览器中充满敌意的 DOM 环境，从而使得组件也可以运行在服务器端、Native 客户端等各种底层平台。令人惊奇的是，这种抽象泄漏非常少，必要时可以很方便地跳出 React 的抽象而直接操纵 DOM 等底层平台。因为 React 专注于组件架构，所以模块系统可以直接采用 CommonJS，测试框架可以使用 Mocha 或 Jasmine 等，生态圈则可以直接依托 npm，工具可以采用现成的 Browserify 或 webpack，我们不必受制于任何单一技术，这非常符合 Web 的开放本质。

在本书中，作者不仅完整地介绍了 React 本身的方方面面，用通俗的语言和简洁的例子阐述了 React 的开发理念，还介绍了一套基于 React 以及业界其他优秀技术的最佳实践，相信读者在看完本书后能够快速将其中的知识应用于项目开发。React 目前处于快速发展时期，在本书发行后，又增加了不少吸引人的新特性，加大了和 ES6 的进一步整合，从而进一步减少了需要学习的 API，大家在看完本书后可以持续关注 React 社区的最新发展动态。值得注意的是，业界基于 React 的优秀组件与传统组件相比仍然严重不足，这对我们来说是一个很好的机会——有机会可以向业界发出中国的声音。积极学习业界的先进技术，未来我们一定能在前端类库领域创造出让业界称赞的东西。

何一鸣（承玉）

蚂蚁金服技术专家

前 KISSY 核心开发者，现 React 爱好者

推荐序3

React 是一种革命性的 UI 组件开发思路。

在此之前，我们所见到的 JavaScript 框架开发思路几乎是同质的。框架为开发者提供一套组件库，业务开发基于组件库提供的组件来进行就可以了。

而在 UI 组件架构里，有几个特点需要注意：一是越靠近用户端变化越快，用枚举组件的思路在高速迭代快速变化的互联网中开发，将会使 UI 组件库逐渐变得臃肿和难以维护。二是组件开发不再是五六年前那样一穷二白的初始状态，现今行业里组件百花齐放，可选面非常广，即使当下找不到非常匹配的组件，进行自研的成本也不高。三是 UI 组件受具体业务场景的约束。

因此，各大互联网公司在组件上都尽可能地采取自研或统一组织组件库。而组件库在公司级别难抽象，对整体技术的挑战比较大，且收效不确定。于是只能将组件场景定位到更具体的某一类型的业务线再进行抽象。从而让组件库变得轻量、灵活。

开发架构的理想态是“同构”。用相同的内部机制与结构将开发变得透明且测试可控。这在 React 里表现得很明显。它的设计非常大胆，一开始就没有将枚举组件功能作为重点，而是以“同构”的理想架构为起点——将原本的 DOM 操作接管，提出 Virtual DOM、单向数据流，用很少的接口覆盖在组件开发的生命周期上，采取声明式的语法等。实现了一个纯粹的组件“引擎”。

另一方面，React 的思路也可作为连接“异构端”的组件入口。现在，用 React + native 就可以实现 React-native；用 React + canvas 就可以实现一套基于 canvas 的高性能的 Web UI 组件；最近，我还尝试了 React + WebComponents，将两者的优势进行融合。

可见，React 的思路为开发创造了非常大的想象空间。

本书内容围绕示例展开，书中还涵盖了 React 的周边信息，为读者提供了较为全面和丰富的 React 讲解。通过阅读本书，读者能够学会如何将 React 运用到实际开发中去。另外，我建议大家不仅要学习 React 的应用如何实现组件，更重要的是通过书中的实例理解 React 的设计思路。可以预见，React 未来将会影响整个用户端 UI 组件的开发。希望能有更多的人了解 React 的开发思路，大家携手共建 React 的组件生态。

刘平川 (rank)

现美团网架构师，React 爱好者

前百度 FEX 创立者及负责人

前言

React 是什么，为什么要使用它

React 是 Facebook 内部的一个 JavaScript 类库，已于 2013 年开源，可用于创建 web 用户交互界面。它引入了一种新的方式来处理浏览器 DOM。那些需要手动更新 DOM、费力地记录每一个状态的日子一去不复返了——这种老旧的方式既不具备扩展性，又很难加入新的功能，就算可以，也是冒着很大的风险。React 使用很新颖的方式解决了这些问题。你只需声明式地定义各个时间点的用户界面，而无须关心在数据变化时需要更新哪一部分 DOM。在任何时间点，React 都能够以最小的 DOM 修改来更新整个应用程序。

本书内容

React 引入了一些激动人心的新概念，向现有的一些最佳实践发起了挑战。本书将会带领你学习这些概念，帮助你理解它们的优势，创建具备高扩展性的单页面应用（SPA）。

React 把主要的注意力放在了应用的“视图”部分，没有限定与服务端交互和代码组织的方式。在本书中，我们将介绍目前的一些最佳实践及配套工具，帮助你使用 React 构建一个完整的应用。

本书面向的读者

为了更好地掌握本书的内容，你需要有 JavaScript 和 HTML 相关开发经验。倘若你做过 SPA 应用（什么框架不重要，Backbone.js、Angular.js 或者 Ember.js 都可以）那更好，但这不是必需的。

源码和示例

一些来自 Reddit 克隆示例项目的代码片段会贯穿在整本书中。你可以在 <http://git.io/vlcpa> 浏览完整的代码，到 <http://git.io/vlCUI> 可以看到在线的 demo。

编写过程

我们把本书当作一本虚拟的电子书编写，用一到两个月的时间快速迭代。这种方式有助于创建新鲜及时的内容，而传统书籍往往无法覆盖最新的趋势和技术。

这是本书的第 2 版，所有的示例代码都更新到了 React 0.14 版，而且有了一个新的示例项目。

作者

本书由一个团队编写而成，这个团队的成员都是一些经验丰富且专注于 JavaScript 的开发者。

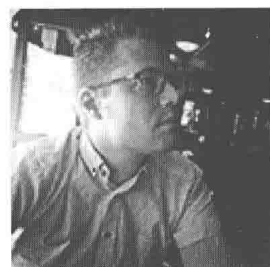
Richard Feldman 是旧金山教育科技公司 NoRedInk 的前端工程师。他是一个函数式编程爱好者，会议发言人，还是 *seamless-immutable* 的作者。*seamless-immutable* 是一个开源类库，可以提供不可变的数据结构，向后兼容普通的 JavaScript 对象和数组。Richard 在 Twitter 和 Github 上都叫 @rtfeldman。



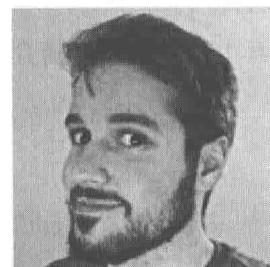
Frankie Bagnardi 是一位高级前端工程师，为多种不同的客户端创造用户体验。在业余时间，他会在 StackOverflow (FakeRainBrigand) 和 IRC (GreenJello) 上回答问题，或者开发一些小项目。可以通过 f.bagnardi@gmail.com 联系他。



Simon Højberg 是一个高级 UI 工程师，在罗德岛普罗维登斯市的 Swipely 公司工作。他是普罗维登斯市线下 JS 见面会的核心组织者，之前还是波士顿创业学院的 JavaScript 讲师。他一直在使用 JavaScript 开发功能性的用户界面，也会开发一些像 cssarrowplease.com 这样的业余项目。Simon 的 Twitter 是 @shojberg。



Jeremiah Hall 现在 OpenGov Inc 任职高级软件工程师 / 架构师，他还是 Aspect Apps 的创始人，该应用使用 React Native 和 JavaScript 来构建一个日志应用的 UI。在 Twitter 可以通过 @jeremiahrhall 找到他。



目录

推荐序1	III
推荐序2	V
推荐序3	VII
前言.....	IX
第 1 章 React 简介.....	1
背景介绍	1
本书概览	3
Component的创建和复合	3
进阶	5
React工具	6
React实践	6
第 2 章 JSX.....	8
什么是JSX	9
使用JSX的好处	10
更加熟悉	10
更加语义化	10
更加直观	11
关注点分离	12

复合组件	12
定义一个自定义组件	12
使用动态值	13
子节点	14
JSX与HTML有何不同	16
属性	16
条件判断	17
非DOM属性	19
事件	21
注释	22
特殊属性	23
样式	23
没有JSX的React	24
创建React元素	24
延伸阅读及参考引用	25
JSX官方规范	25
第 3 章 组件的生命周期	26
生命周期方法	26
实例化	27
存在期	27
销毁&清理期	27
实例化	27
componentWillMount	28
render	28
componentDidMount	29
存在期	30
componentWillReceiveProps	30
shouldComponentUpdate	30

componentWillUpdate	31
componentDidUpdate	31
销毁&清理期	32
componentWillUnmount	32
反模式：把计算后的值赋给state	32
总结	33
第 4 章 数据流	34
Props	35
PropTypes	36
defaultProps	37
State	37
放在state和props的各是哪些部分	40
无状态的函数式组件	40
总结	41
第 5 章 事件处理	42
绑定事件处理器	43
事件和状态	44
根据状态进行渲染	44
更新状态	45
状态没有“更新”！	46
事件对象	47
总结	48
第 6 章 组件的复合	49
扩展HTML	50
组件复合的例子	50
组装HTML	51

追踪状态	53
整合到父组件当中	54
父组件与子组件的关系	56
总结	59
第 7 章 高阶组件和Mixins	60
简单的例子	60
常见使用场景	63
总结	63
第 8 章 DOM操作	64
访问受控的DOM节点	65
在组件内部查找DOM节点	66
整合非React类库	67
侵入式插件	70
总结	72
第 9 章 表单	73
无约束的组件	74
约束组件	76
表单事件	78
Label	79
文本框和 Select	79
复选框和单选框	82
表单元素的name属性	84
多个表单元素与change处理器	86
自定义表单组件	89
Focus	93
可用性	93

把要求传达清楚	94
不断地反馈	94
迅速响应	94
符合用户的预期	95
可访问	95
减少用户的输入	96
总结	97
第 10 章 动画	98
CSS渐变组	99
给渐变class添加样式	99
渐变生命周期	100
使用渐变组的隐患	101
间隔渲染	101
使用requestAnimationFrame实现间隔渲染	102
使用setTimeout实现间隔渲染	104
弹簧动画	105
总结	106
第 11 章 性能优化	107
shouldComponentUpdate	107
键 (key)	109
总结	111
第 12 章 服务端渲染	112
渲染函数	113
React.renderToString	113
React.renderToStaticMarkup	114
用React.renderToString还是用React.renderToStaticMarkup	114
服务端组件生命周期	115