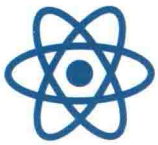
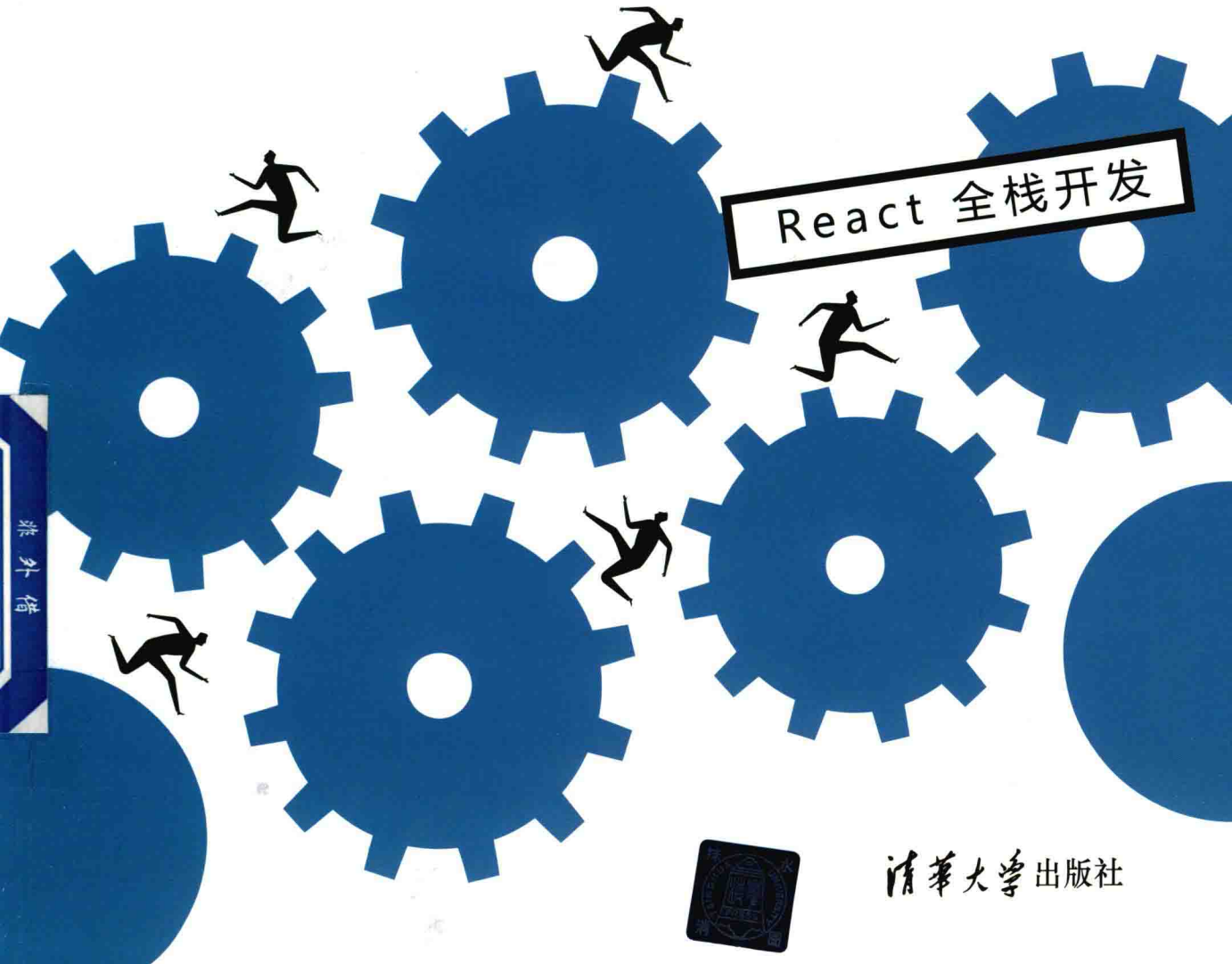


React 16
React Router
Redux
MobX

React 进阶之路

·徐 超 编著·



清華大學出版社

React 进阶之路

React 进阶之路

·徐 超 编著·

清華大學出版社
北京

内 容 简 介

本书详细介绍了 React 技术栈涉及的主要技术。本书分为基础篇、进阶篇和实战篇三部分。基础篇主要介绍 React 的基本用法，包括 React 16 的新特性；进阶篇深入讲解组件 state、虚拟 DOM、高阶组件等 React 中的重要概念，同时对初学者容易困惑的知识点做了介绍；实战篇介绍 React Router、Redux 和 MobX 3 个 React 技术栈的重要成员，并通过实战项目讲解这些技术如何和 React 结合使用。

本书示例丰富、注重实战，适用于从零开始学习 React 的初学者，或者已经有一些 React 使用经验，但希望更加全面、深入理解 React 技术栈的开发人员。阅读本书，需要先掌握基础的前端开发知识。

图书在版编目（CIP）数据

React 进阶之路/徐超编著. —北京：清华大学出版社，2018
ISBN 978-7-302-49801-8

Ⅰ. ①R… Ⅱ. ①徐… 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 037116 号

责任编辑：王金柱
封面设计：王　翔
责任校对：闫秀华
责任印制：李红英

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：北京密云胶印厂
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张**：15.75　　**字　　数**：403 千字
版　　次：2018 年 4 月第 1 版　　**印　　次**：2018 年 4 月第 1 次印刷
印　　数：1~3500
定　　价：69.00 元

产品编号：077579-01

推荐序

小时候，老师问大家长大的理想是什么。我记得曾自豪地说——工程师。后来，真的走进了计算机领域，成为一名软件工程师。在学校里学的都是基础课，记忆犹新的有计算机原理、操作系统、编译原理、数据结构和算法等，感觉是终身受益，就像练武功，都要练好弓、马、仆、虚、歇5种基本步型一样。

那时并没有前端的说法。人机界面开始主要以程序员使用为主，通过黑洞洞的 Terminal 来编程，程序员还乐此不疲。后来出现了第一波突破——各种图形界面，PC 变得亲民。而以 iPhone 带领的移动终端的第二波革新让用户能够通过触摸、视觉和声音真正自然地与设备交互。将来必然拥有超越触摸、视觉和声音识别的技术，属于传感器和物联网的时代。这种技术使用传感器和人工智能识别身体运动、温度变化和其他环境要素，并据此做出回应，使得设备看起来可以读懂内心的想法一样。在不久的将来，一个传感器阵列能够提供高度的情境感知，并且协同工作，收集和处理关于周边环境的信息，通过人工智能预测需求并做出完全个性化的安排。前端工程师的使命也随着人机交互的显著进步而不断拓展。

时光回到刚工作时的 2000 年，正值互联网的发展初期，作为一名软件工程师，解决问题就是关键，对于前后端编程都需要熟悉。当时，前端编程的核心技能有 HTML、CSS、JavaScript， 对于习惯逻辑思维的工程师，学起来并不算难。随着互联网的发展，特别是 2010 年后，移动设备成为主流，前端工程师角色被行业认可，并且越来越重要，涵盖多终端的视觉和交互的实现，面对的是软件工程的一个持久的挑战——人机交互。首先，人机交互是软件产品里变化最频繁的部分，同时是非常关键的一环。其次，兼容各种浏览器、Web 的标准，以及适配多种终端，都是很大的挑战。另外，前端领域的技术发展也越来越快，各种新的思想、设计模式、工具和平台不断出现，怎样快速学习、在不同场景下做出恰当的选择是成为一位优秀前端工程师必备的素质。许多人机交互问题有非常巧妙的思路和精彩的解决办法。不得不说，前端工程师在工程师群体里属于非常有创造力、想象力的一群人。

前端领域各种新技术、新思想不断涌现，AngularJS、React、Vue.js、Node.js、ES 6、ES 7、CoffeeScript、TypeScript，令人眼花缭乱。对于许多开发者，估计还没学明白一样技术，就发现其已被另一些新的技术取代而“过时”了。但是，如果退一步来看，前端的基本功仍然是 HTML、CSS、JavaScript，还有算法、数据结构、编译原理。这一点，有点像《笑傲江湖》里，令狐冲一旦领悟了独孤九剑，永远能够无招胜有招。

除了具备扎实的基本功之外，一个优秀的前端工程师必须要有自己擅长的领域，并且钻研得足够深入，只有花时间学习成体系的知识才能从中总结出规律并形成方法论，从而最大化学习的价

值。同时要有广泛的视野，不能局限于前端本身，因为有很多东西只有站在前端之外才能看得更清晰、更透彻。例如，React 集成了许多后端的优秀理念，包括采用声明范式轻松描述应用、通过抽象 DOM 来达到高效的编程。围绕 React 还出现了许多工具和框架，形成了 React 生态。React 逐渐从最早的 UI 引擎变成了前后通吃的 Web App 解决方案，衍生出来的 React Native 又实现了用 Web App 的方式去写 Native App。这样，同一组人写一次 UI 就能运行在浏览器、移动终端和服务器上。

作为智能物联网先锋的远景智能，一直崇尚工程师文化和工匠精神，非常强调基本功、专业深度以及跨界创新。前端团队徐超写的《React 进阶之路》，内容由浅入深，再结合实战，很像我读大一时的 Java 101 课程的教材，对于需要学习 React 的读者是一部非常好的参考书。读这本书，最好的方法是领悟其精髓，掌握软件设计之路，灵活使用以解决问题。工程师不能因为太细的学科限制了自己的思维，也不能像大公司一个工作一个螺丝钉，在很窄的领域里重复劳动。工程师天生是发现问题、解决问题、优化问题的。达·芬奇、特斯拉之所以是完美的工程师，因为他们会掌握各种学科，融合并创新，在解决问题的同时开创先河。未来的信息化世界就是要不断地聪明学习，融合各种学科，通过实践解决问题，奇思妙想地创造技术的进步。

计算机的不同语言、不同技术和算法就好比一堆便宜或者昂贵的工具（如锥子和刨子），其实这些都不重要，因为大家都忽略了，做出漂亮器具的是那个工匠，而不是工具。脑子里的经验积累、天赋、执着与认真的态度、不停尝试、追求完美的态度，加起来才能创造好的作品与产品。计算机语言就像赛车场上的跑车，换了车队和跑车，舒马赫还是 F1 车神，观众还是会为其欢呼雀跃，正因为车神掌握了与跑车和赛道的沟通之道！

远景智能技术副总裁、前阿里巴巴集团淘宝 CTO

余海峰

前　言

当今，前端应用需要解决的业务场景正变得越来越复杂，这也直接推动了前端技术的迅速发展，各种框架和类库日新月异、层出不穷。面对众多的框架和类库，前端开发者可能感到眼花缭乱，但换一个角度来看，这未尝不是一种百家争鸣的现象。不同框架和类库的设计思想和设计理念各有千秋，解决的问题也有所不同，这些多元化和差异化不断推动前端技术的发展，同时也是前端技术领域的一份思想瑰宝。

React 作为当今众多新技术的一个代表，由 Facebook 开源，致力于解决复杂视图层的开发问题，它提出一种全新的 UI 组件的开发理念，降低了视图层的开发复杂度，提高了视图层的开发效率，让页面开发变得简单、高效、可控。此外，React 不仅是单一的类库，更是一个技术栈生态，可以和生态中的 Redux、MobX 等其他技术结合使用，构建可扩展、易维护、高性能的大型 Web 应用。

本书内容

本书涵盖 React 技术栈中的主要技术，内容由浅到深。本书内容分为基础篇、进阶篇和实战篇，每一篇内容又分成若干章节来介绍。

基础篇，介绍了 React 的基本概念，包括 React 的开发环境和开发工具、React 的基本用法和 React 16 的新特性。每个知识点都有配套的项目示例。

进阶篇，深入介绍了 React 的几个重要概念，如组件 state、虚拟 DOM、高阶组件等，此外，还针对初学者使用 React 时容易产生困惑的知识点做了专门讲解，如组件与服务器通信、组件之间通信、组件的 ref 属性等。

实战篇，介绍了 React 技术栈中最重要的三个技术：React Router、Redux 和 MobX，每一个技术都配有详细的项目实战示例。

本书章节的难度逐步递增，各章节的知识存在依赖关系，所以读者需按照章节顺序阅读本书，不要随意跳跃章节，尤其是在阅读实战篇时，务必保证已经掌握了基础篇和进阶篇的内容，否则，阅读实战篇可能会有些吃力。

本书特点

本书的特点是内容全、知识新、实战性强。

内容全：本书不仅详细介绍了 React 的使用，还详细介绍了 React 技术栈中最常用的其他相关技术：React Router、Redux 和 MobX。

知识新：本书介绍的知识点都是基于各个框架、类库当前的最新版本，尤其是涵盖 React

16 的新特性和 React Router 4 的介绍。对于新版本已经不再支持或建议废弃的特性，本书不会再介绍，确保读者所学知识的时效性。

实战性强：本书配有大量示例代码，保证读者学以致用。实战篇使用的简易 BBS 项目示例接近真实项目场景，但又有所简化，让读者既可以真正理解和领会相关技术在真实项目中的使用方式，又不会因为示例项目过于复杂而影响学习。

本书目标读者

本书面向希望从零开始学习 React 的初学者，或者已经有一些 React 使用经验，希望更加全面、深入理解 React 技术栈的开发人员。

示例代码

本书的示例代码下载地址为 https://github.com/xuchaobei/react-book。如果读者发现代码或者书中的错误，可以直接在该代码仓库提交 issue。

本书中默认的开发环境是 Node.js v8.4.0，书中介绍到的几个主要库的版本分别为 React 16.1.1、React Router 4.2.2、Redux 3.7.2 及 MobX 3.3.1。

致谢

本书的完成离不开在各个方面给过我支持和帮助的人，请允许我在这里向他们表示感谢。

首先，感谢公司的领导余海峰（Colin）和贺鸣（Sky）对我写书的支持。Colin 在百忙之中还抽出时间为本书作序。

其次，感谢我的同事王博、陈小梦、吴福城、詹敏和朱雅琴，他们给本书提出了很多宝贵的意见。

还要感谢我的老婆，2017 年，她的新书《时间的格局：让每一分钟为未来增值》出版，这也让我产生了写书的念头，同时她的写书经验也给了我很多帮助。

最后，感谢清华大学出版社的王金柱老师，正是缘于他的主动联系，才让我写书的念头变成了行动。他认真、负责的工作态度也保证了本书的顺利问世。

联系作者

欢迎各位读者通过我的微信订阅号：老干部的大前端（ID：Broad_FE）和我进行沟通交流，订阅号还提供了更多的大前端学习资源。读者可以扫描右方二维码关注订阅号。

徐　超

2018 年 1 月 1 日于上海

目　录

第 1 篇　基础篇——React，一种革命性的 UI 开发理念

第 2 篇 进阶篇——用好 React，你必须要知道的那些事

第 3 篇 实战篇——在大型 Web 应用中使用 React

第1篇　基础篇

React，一种革命性的UI开发理念

第1章

初识 React

当今，Web 应用的业务场景正在变得越来越复杂，几乎所有应用都在尝试或者已经在 Web 上使用，同时，用户对 Web 应用的体验要求也越来越高，这一切都给前端开发人员开发前端界面带来了巨大的挑战。而 Facebook 开源的 React 技术创造性地提出了一种全新的 UI 开发理念，让 UI 开发变得简单、高效、可控。React 自开源以来，迅速风靡整个前端世界，推动前端开发有了革命性的进步。

1.1 React 简介

前端 UI 的本质问题是如何将来源于服务器端的动态数据和用户的交互行为高效地反映到复杂的用户界面上。React 另辟蹊径，通过引入虚拟 DOM、状态、单向数据流等设计理念，形成以组件为核心，用组件搭建 UI 的开发模式，理顺了 UI 的开发过程，完美地将数据、组件状态和 UI 映射到一起，极大地提高了开发大型 Web 应用的效率。

React 的特点可以归结为以下 4 点：

（1）声明式的视图层。使用 React 再也不需要担心数据、状态和视图层交错纵横在一起了。React 的视图层是声明式的，基于视图状态声明视图形式。但 React 的视图层又不同于一般的 HTML 模板，它采用的是 JavaScript（JSX）语法来声明视图层，因此可以在视图层中随意使用各种状态数据。

（2）简单的更新流程。React 声明式的视图定义方式有助于简化视图层的更新流程。你只需要定义 UI 状态，React 便会负责把它渲染成最终的 UI。当状态数据发生变化时，React 也会根据最新的状态渲染出最新的 UI。从状态到 UI 这一单向数据流让 React 组件的更新流程清晰简洁。

（3）灵活的渲染实现。React 并不是把视图直接渲染成最终的终端界面，而是先把它们渲染成虚拟 DOM。虚拟 DOM 只是普通的 JavaScript 对象，你可以结合其他依赖库把这个对象渲染成不同终端上的 UI。例如，使用 react-dom 在浏览器上渲染，使用 Node 在服务器端渲染，使用 React Native 在手机上渲染。本书主要以 React 在浏览器上的渲染为例介绍 React 的使用，但你依然可以很容易地把本书的知识应用到 React 在其他终端的渲染上。

（4）高效的 DOM 操作。我们已经知道虚拟 DOM 是普通的 JavaScript 对象，正是有了虚拟 DOM 这一隔离层，我们再也不需要直接操作又笨又慢的真实 DOM 了。想象一下，操作一个 JavaScript 对象比直接操作一个真实 DOM 在效率上有多么巨大的提升。而且，基于 React 优异的差异比较算法，React 可以尽量减少虚拟 DOM 到真实 DOM 的渲染次数，以及每次渲染需要改变的真实 DOM 节点数。

虽然 React 有这么多强大的特性，但它并不是一个 MVC 框架。从 MVC 的分层来看，React 相对于 V 这一层，它关注的是如何根据状态创建可复用的 UI 组件，如何根据组件创建可组合的 UI。当应用很复杂时，React 依然需要结合其他库（如 Redux、MobX 等）使用才能发挥最大作用。

1.2　ES 6 语法简介

ES 6 是 JavaScript 语言的新一代标准，加入了很多新的功能和语法。React 的项目一般都是用 ES 6 语法来写的，这也是 Facebook 官方推荐的方式。为保证本书知识体系的完整性，本节我们会对开发 React 应用经常用到的 ES 6 语法做简要介绍。

1. let、const

let 和 const 是 ES 6 中新增的两个关键字，用来声明变量，let 和 const 都是块级作用域。let 声明的变量只在 let 命令所在的代码块内有效。const 声明一个只读的常量，一旦声明，常量的值就不能改变。例如：

```
// let 示例
{
  var a = 1;
  let b = 2;
}
a          // 1
b          // ReferenceError: b is not defined.

//const 示例
const c = 3;
c = 4;      //TypeError: Assignment to constant variable.
```

2. 箭头函数

ES 6 允许使用“箭头”（=>）定义函数。这种方式创建的函数不需要 function 关键字，并且

还可以省略 return 关键字。同时，箭头函数内的 this 指向函数定义时所在的上下文对象，而不是函数执行时的上下文对象。例如：

```
var f = a => a + 1;
//等价于
var f = function(a){
  return a + 1;
}

function foo(){
  this.bar = 1;
  this.f = (a) => a + this.bar;
}
//等价于
function foo(){
  this.bar = 1;
  this.f = (function(a){
    return a + this.bar
  }).bind(this);
}
```

如果箭头函数的参数多于 1 个或者不需要参数，就需要使用一个圆括号代表参数部分。例如：

```
var f = () => 1;
var f = (a, b) => a + b;
```

如果函数体内包含的语句多于一条，就需要使用大括号将函数体括起来，使用 return 语句返回。例如：

```
var f = (x, y) => {
  x++;
  y--;
  return x + y;
}
```

3. 模板字符串

模板字符串是增强版的字符串，用反引号（`）标识字符串。除了可以当作普通字符串使用外，它还可以用来定义多行字符串，以及在字符串中嵌入变量，功能很强大。例如：

```
//普通字符串
`React is wonderful !`

//多行字符串
'JS is wonderful !
React is wonderful!'
```