



React Native

● 视频课程

● 案例素材

● 交流社区

● QQ 讨论组

React Native 企业实战

主 编 肖 睿 王曙光
副主编 南 洋 桑宇鹏 宋 磊



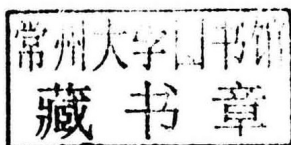
中国水利水电出版社
www.waterpub.com.cn

前端开发工程师系列

React Native 企业实战

主 编 肖 睿 王曙光

副主编 南 洋 桑宇鹏 宋 磊



 中国水利水电出版社
www.waterpub.com.cn

• 北京 •

内 容 提 要

随着前端框架的发展,来自 Facebook 的前端类库 React 因为独特的设计而被开发者所喜爱。React 引入了一些激动人心的新概念(如虚拟 DOM、JSX 等),使得前端开发者更关注应用的 View(视图)部分,并慢慢从 Web 开发领域跨越到客户端领域。React 同时拥有 Native App 的性能和 Hybrid App 的可维护性,并可以应用在多个平台上,因此很多知名 App 中都使用了这项技术。

作者结合自身丰富的开发经验,以实际开发场景为案例,详细讲述了从理论到应用并解决实际问题的过程。本书不仅包括大量 GitHub 资源,更有很多实际开发经验和技巧总结,这也是本书的亮点所在。另外,本书还配有完善的学习资源和支持服务,包括视频教程、案例素材下载、学习交流社区、讨论组等终身学习内容,为读者带来全方位的学习体验。更多技术支持请访问课工场(www.kgc.cn)。

图书在版编目(CIP)数据

React Native企业实战 / 肖睿, 王曙光主编. -- 北
京 : 中国水利水电出版社, 2017. 12
(前端开发工程师系列)
ISBN 978-7-5170-6082-6

I. ①R… II. ①肖… ②王… 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293029号

策划编辑: 石永峰 责任编辑: 周益丹 加工编辑: 张溯源 封面设计: 梁 燕

书 名	前端开发工程师系列 React Native 企业实战 React Native QIYE SHIZHAN
作 者	主 编 肖 睿 王曙光 副主编 南 洋 桑宇鹏 宋 磊
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市铭浩彩色印装有限公司
规 格	184mm×260mm 16 开本 11.75 印张 284 千字
版 次	2017 年 12 月第 1 版 2017 年 12 月第 1 次印刷
印 数	0001—3000 册
定 价	30.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换
版权所有·侵权必究

课工场介绍

课工场是专注互联网教育的生态平台，汇聚了中国和北美数百位来自知名互联网企业的行业大咖，向寻求就业和技术提升的人群提供直播、录播、面授等多模式教学场景，并通过遍布全国的线下服务中心提供成熟的学习服务，形成完善的“互联网+教育”解决方案。同时，课工场也为高校、企业、行业提供教育技术赋能，依托 Transformer 智能教育生态平台，打造智慧校园、企业大学、行业培训的教育场景，提供一站式教育解决方案。

课工场于 2017 年荣膺新浪网“2017 中国品牌影响力教育机构”，腾讯网“2017 年度影响力教育品牌”，网易“2017 年度品牌影响力教育机构”，中国网“中国好教育-2017 年度综合实力突出教育集团”，作者肖睿荣获中国网“中国好教育-2017 年度互联网+教育风云人物”。



扫一扫关注课工场公众号
关注微信，立送204币
可购买收费课程



课工场APP客户端下载
产品/设计/开发/运维/运营
随时随地随心学

课工场岗位课程

- 大数据开发工程师
- 前端开发工程师
- Android 开发工程师
- 新媒体运营师
- 移动端 UI 设计师
- 网页 UI 设计师
- 互联网 UI 设计师
- Python 开发工程师
- 云计算工程师
- VR 游戏设计师
- VR 游戏开发工程师
- VR 商用开发工程师

更多课程请访问 kgc.cn

联系我们

北京课工场教育科技有限公司

网址: kgc.cn

Q Q: 800161516

邮箱: ke@kgc.cn

电话: 010-88550007

地址: 北京市海淀区成府路 207 号
B 座一层

前端开发工程师系列

编委会

主任：肖 睿

副主任：杨 欢

委员：相洪波 谢伟民 张德平 庞国广 潘贞玉

张惠军 段永华 李 娜 孙 苹 董泰森

于学杰 王宗娟 何娅玲 陈 璇 陈 金

课工场：祁春鹏 宗 娜 王博君 曹紫涵 尚永祯

吴宇迪 刁志星 张雪妮 刘景元 赵 娜

原 婧 赵文雅 周翠玉 王玉伟 刘 洋

刘建宁 陈 茵 乔 伟 王丙辰 王芹芹

周士昆 胡杨柳依

前言

随着互联网技术的飞速发展,“互联网+”时代已经悄然到来,这自然催生了互联网行业工种的细分,前端开发工程师这个职业应运而生,各行业、企业对前端设计开发人才的需求也日益增长。与传统网页开发设计人员相比,新“互联网+”时代对前端开发工程师提出了更高的要求,传统网页开发设计人员已无法胜任。在这样的大环境下,这套“前端开发工程师系列”教材应运而生,旨在帮助读者快速成长为符合“互联网+”时代企业需求的优秀的前端开发工程师。

“前端开发工程师系列”教材是由课工场(www.kgc.cn)的教研团队研发的。课工场是北京大学下属企业北京课工场教育科技有限公司推出的互联网教育平台,专注于互联网企业各岗位人才的培养。平台汇聚了数百位来自知名培训机构、高校的顶级名师和互联网企业的行业专家,面向大学生以及需要“充电”的在职人员,针对与互联网相关的产品设计、开发、运维、推广和运营等岗位,提供在线的直播和录播课程,并通过遍及全国的几十家线下服务中心提供现场面授以及多种形式的教学服务,并同步研发出版最新的课程教材。本书由肖睿、王曙光任主编,南洋、桑宇鹏、宋磊任副主编,其中王曙光编写第1章至第3章,南洋编写第4章至第7章,桑宇鹏编写第8章,宋磊编写第9章。

为培养互联网前端设计开发人才,课工场特别推出“前端开发工程师系列”教育产品,提供各种学习资源和支持,包括:

- 现场面授课程
- 在线直播课程
- 录播视频课程
- 案例素材下载
- 学习交流社区
- QQ讨论组(技术、就业、生活)

以上所有资源请访问课工场(www.kgc.cn)。

本套教材特点

- (1) 科学的训练模式。
 - 科学的课程体系。
 - 创新的教学模式。
 - 技能人脉,实现多方位就业。
 - 随需而变,支持终身学习。
- (2) 真实的项目驱动。
 - 覆盖80%的网站效果制作。
 - 几十个实训项目,涵盖电商、金融、教育、旅游、游戏等行业。
- (3) 便捷的学习体验。

- 每章提供二维码扫描，可以直接观看相关视频讲解和案例操作。
- 课工场开辟教材配套版块，提供素材下载、学习社区等丰富的在线学习资源。

读者对象

（1）初学者：本套教材将帮助你快速进入互联网前端开发行业，从零开始逐步成长为专业的前端开发工程师。

（2）初级前端开发者：本套教材将带你进行全面、系统的互联网前端设计开发学习，帮助你梳理全面、科学的技能理论，提供实用的开发技巧和项目经验。

课工场出品（www.kgc.cn）

课程设计说明

课程目标

读者学完本书后，能够顺利搭建 React 开发环境，使用 Flux 管理 React 数据，运用 React Router，并使用 React + Redux 完成 TodoList 项目开发，顺利开启 React Native 开发之旅。

训练技能

- 能够搭建 Node.js 环境，熟练使用 Gulp。
- 能够搭建 React 开发环境。
- 能够使用 React + Redux 开发独立功能。
- 能够使用 React Native 进行开发。

课程设计思路

本课程分为 9 章、5 个阶段来设计学习，即 Node.js 及 Gulp、React 与 Flux、React Router、Server Side Render 与 React Native 开发，具体安排如下：

- 第 1 章至第 3 章：了解前端发展历史、顺利安装 Node.js 环境、使用 Gulp，为 React 学习做好铺垫。
- 第 4 章至第 6 章：搭建 React 环境、使用 Flux 对 React 数据进行管理、使用 React+Redux 完成一个项目实战——TodoList。
- 第 7 章：对 React Router 进行介绍，从原理、主要组件再到实际应用，最后讲述两个案例的实现——Sidebar 和 Modal Gallery。
- 第 8 章：对 Server Side Render 进行学习，掌握 React 应用服务器端渲染的方法和利弊，让任何搜索引擎的爬虫方便地抓取网站内容。
- 第 9 章：带领你走进 React Native 开发之旅，从开发环境搭建、第一个 React Native 程序到 Flexbox 布局，从 JSX 语法的应用到 React Native UI 组件的介绍与应用，一步一步领略 React Native 开发的奥秘与乐趣。

章节导读

- 本章技能目标：学习本章所要掌握的技能，可以作为检验学习效果的标准。
- 本章简介：学习本章内容的原因和对本章内容的概述。
- 内容讲解：对本章涉及的技能内容进行分析并展开讲解。
- 操作案例：对所学内容的实操训练。
- 本章总结：针对本章内容的概括和总结。
- 本章作业：针对本章内容的补充练习，用于加强对技能的理解和运用。

学习资源

- 学习交流社区（课工场）
- 案例素材下载
- 相关视频教程

更多内容详见课工场（www.kgc.cn）。



关于引用作品版权说明

为方便学校课堂教学、促进知识传播、使读者学习优秀作品，本书选用了一些知名网站的相关内容作为知识引入和案例等。为了尊重这些内容所有者的权利，特在此声明：凡书中涉及的版权、著作权、商标权等权益均属于原作品版权人、著作权人、商标权人。

为维护原作品相关权益人的权益，现对本书选用的主要作品的出处给予说明（排名不分先后）。

序号	使用或访问资源
1	http://facebook.github.io/flux/
2	https://nodejs.org/en/
3	阮一峰的博客

由于篇幅有限，以上列表中可能并未全部列出本书所选用的作品。在此，衷心感谢所有原作品的相关版权权益人及所属公司对职业教育的大力支持！

2017 年 8 月

目 录

前言	
课程设计说明	
关于引用作品版权说明	
第1章 前端的发展之路	1
1 前端的各个时代	2
1.1 IE6时代	2
1.2 Web 2.0时代	2
1.3 Node.js时代	3
1.3.1 服务	3
1.3.2 构建	3
1.4 跨端时代	4
2 MV*框架的发展	4
2.1 Angular	4
2.2 Vue.js	4
2.3 React	5
2.4 Polymer	5
2.5 Riot	5
2.6 Backbone.js	6
3 React 包含哪些	6
3.1 虚拟 DOM	6
3.2 JSX	7
3.3 render 函数	7
3.4 组件	8
4 准备学习 React	8
4.1 组件生命周期	8
4.2 ES6	9
4.3 NPM	9
4.4 Babel	10
4.5 Gulp	10
本章总结	11
本章作业	11
第2章 Node.js 环境搭建	13
1 Node.js 介绍	14
1.1 Node.js 安装	14
1.2 模块化	16
1.2.1 CommonJS 规范	16
1.2.2 require 方法	16
1.2.3 综合案例分析	17
1.3 核心模块	17
1.3.1 http	18
1.3.2 fs	18
1.3.3 path	19
1.3.4 querystring	19
1.3.5 crypto	19
1.3.6 util	20
1.4 第三方模块	20
1.4.1 request	21
1.4.2 async	21
1.4.3 commander	21
1.4.4 html-minifier	22
1.4.5 less	22
1.4.6 lru-cache	23
1.4.7 qs	24
1.4.8 rimraf	24
1.4.9 shelljs	24
1.4.10 yargs	24
2 Node.js 调试	25
2.1 GUI 方式——Node Inspector	25
2.2 内置调试——Node debug	26
3 Node.js 命令行工具	26
3.1 可执行文件	26
3.2 命令行支持参数	27
4 NPM	27
4.1 安装第三方工具包	28
4.2 package.json 文件	29

4.3 常用命令	30	1.2 MVP	56
4.4 发布工具包	31	1.3 MVVM	57
4.5 取消发布过的工具包	32	2 React 简介	57
本章总结	32	2.1 什么是 React	58
本章作业	33	2.2 如何安装	58
第3章 Gulp	35	2.3 特性	58
1 Gulp 是什么	36	2.3.1 JSX	60
1.1 Gulp 安装	36	2.3.2 数据传递	61
1.2 gulpfile 文件	37	3 React 组件化	68
1.3 Gulp 命令行	38	3.1 组件生命周期	68
1.3.1 gulp -T	38	3.2 编写一个完整的 Button 组件	69
1.3.2 gulp -h	38	本章总结	71
2 Gulp 常用 API	38	本章作业	71
2.1 gulp.src	39	第5章 Flux	73
2.2 gulp.dest	39	1 安装	74
2.3 gulp.task	40	2 基础知识	74
2.4 gulp.watch	41	本章总结	81
3 Gulp 插件	42	本章作业	81
3.1 编译 less 文件	42	第6章 React + Redux 实战	83
3.2 编译 stylus 文件	43	1 为什么使用 Redux	84
3.3 编译 sass 文件	44	1.1 撤销与重做	85
3.4 压缩 css 文件	44	1.2 Redux DevTools 调试工具	86
3.5 在 css 里面自动添加浏览器前缀	45	2 Redux	87
3.6 压缩 js 文件	46	2.1 Redux 设计思想	87
3.7 合并多个文件	47	2.2 Redux 核心概念与 API	88
3.8 压缩 html 文件	47	2.3 Redux 工作流程	92
3.9 给文件名增加 md5	48	2.4 react-redux 的用法	92
3.10 如何启动本地服务	49	3 实例: TodoList	96
3.11 支持 pug 模板编译	50	4 单元测试	102
3.12 支持 zip 压缩	51	4.1 Action 创建函数 (Action Creators)	103
4 Gulp 优化	51	4.2 Reducers	103
4.1 优化一: 清理文件和文件夹	51	4.3 Components	105
4.2 优化二: 把配置抽离到文件中去	52	4.4 连接组件	106
4.3 优化三: 拆分 Gulp 任务	53	本章总结	107
本章总结	54	本章作业	107
本章作业	54	第7章 React Router	109
第4章 初识 React	55	1 React Router 概述	110
1 MV*模式	56	1.1 路由的基本原理	110
1.1 MVC	56	1.2 React Router 基本用法	114

2	React Router 组件介绍	117	4	Router 的处理	144
2.1	Router 组件	117		本章总结	148
2.2	Route 组件	119		本章作业	148
2.3	Link 组件	121	第 9 章	React Native 开发	151
2.4	其他组件	122	1	React Native 开发入门	152
2.4.1	Switch 组件	122	1.1	搭建 iOS App 的 RN 开发环境	152
2.4.2	Redirect 组件	123	1.2	搭建 Android App 的 RN 开发环境	154
2.4.3	Prompt 组件	123	1.3	Hello World	158
3	案例: React Router 的运用	124	2	布局	158
3.1	Sidebar	124	2.1	React Native 布局 and 标准 Flexbox 的区别	159
3.2	Modal Gallery	126	2.2	Flexbox 三个重要布局属性的用法	159
	本章总结	130	2.3	Flexbox 布局案例	161
	本章作业	130	3	JSX 在 React Native 中的应用	165
第 8 章	服务器端渲染	131	4	React Native UI 组件	166
1	为什么用服务器端渲染	132	4.1	运行 UI 组件示例	167
1.1	利于 SEO	132	4.2	理解基础组件 View	169
1.2	减少首次渲染时间	134		本章总结	173
1.3	前后端代码同构	136		本章作业	173
2	服务器端渲染示例	136			
3	state 的处理	140			

第 1 章

前端的发展之路

本章技能目标

- 了解前端发展趋势
- 了解前端 MV*框架的对比
- 了解 React 的优势
- 为 React 的学习做准备

本章简介

前端开发的高速发展推动了各种框架和工具的出现，前端发展基本经历了如下几个时代：

- IE6 时代：前端大部分的痛点都在兼容低版本的浏览器上。
- Web 2.0 时代：从 jQuery 这种 lib 框架开始转向 MV*分层模式上，移动端开发也日益增多。
- Node.js 时代：围绕前端工程化体系，集成环境的搭建工具、构建工具一个接着一个产生。
- 跨端时代：React Native 等框架出现，可以实现一份代码来跨客户端开发。

本章会通过比较来看一下 React 的优势和需要准备的学习内容。

1 前端的各个时代

这几年，前端的受关注度越来越大，前端框架和相关技术工具等日新月异。我们也从 PC 端浏览器应用慢慢地过渡到移动端浏览器应用，再到客户端混合应用，以及以 React Native（简称 RN）和 Weex 为代表的跨端框架。

下面介绍前端发展经历的几个时代。

1.1 IE6 时代

还记得 IE6 的那些兼容规则吗？现在看，应该还是有很多前端开发者在和它打交道。

在这个时代里面，我们更注重页面构建（HTML + CSS），更多精力花费在低版本浏览器的兼容性上，基本包含：

- 样式 CSS 的兼容性。
- 脚本 JavaScript 的兼容性。

导致兼容性问题的主要影响因素：PC 时代大部分使用环境基本都是内置了 IE 浏览器的 Windows 计算机，由于 IE 浏览器和谷歌、火狐在内核和标准上存在差异，而且对 ECMAScript 的支持力度偏弱，因此导致各浏览器存在兼容性问题。

这个时代的前端还处于特别早期，需要的技能相对较基础，调试和开发工具比较单一，面对更多的是 PC 的浏览器环境。

1.2 Web 2.0 时代

这个时代的前端开始由 PC 转向移动端，也开始出现前端 MV* 分层框架的新模式，更多关注 UI 视图和数据的绑定。

高级浏览器开始覆盖更多的用户使用场景，而不只是 IE 浏览器，比如谷歌的 Chrome、火狐的 Firefox、苹果的 Safari 等。

加上移动端开发工作越来越多，我们开始关注移动浏览器的适配性。

jQuery、Zepto 等兼容性的库开始流行，基本都使用它来磨平不同浏览器之间的差异，这里主要是脚本的兼容性。

慢慢地前端开发者也开始关注用户界面（UI）和数据之间的绑定关系，开始学习后端语言的框架设计，开始考虑分层，引入了 MV* 的概念。

1. MVC

一般分为 Model（模型）、View（视图）和 Controller（控制器）。View 一般通过 Controller 来和 Model 进行联系，不直接联系。而且各个方向都是单向的。

2. MVP

将 Controller 改成了 Presenter，View 一般通过 Presenter 来和 Model 进行联系，不直接联系。

3. MVVM

将 Controller 改成了 ViewModel, View 一般通过 ViewModel 来和 Model 进行联系, 不直接联系。而且 View 的变化会自动更新到 ViewModel, ViewModel 的变化也会自动同步到 View 上显示。

越来越多的前端框架采用不同的分层模式, 它们各有自己独特的优势和应用场景, 而且随着 ECMAScript 的标准化, 更多的开发者开始使用 ES6 来简洁高效地编写代码。前端开发者使用的工具也开始多样化, 并使用工具完成代码合并、压缩等功能。

第 4 章将会重点介绍几种分层模式的区别。

1.3 Node.js 时代

随着 Node.js 的发展, 前端开始进入了一个新时代, 开发者可以编写命令行工具, 搭建本地开发环境, 通过构建将资源进行打包合并来优化前端加载等性能指标。

1.3.1 服务

开发者可以通过 Node.js 的内置核心包 http 来搭建一个本地服务:

- 可以是静态文件服务, 指向一个固定目录, 然后逐级访问。
- 可以是 API 服务, 支持 GET、POST 等常见方法, 返回 json 数据。
- 可以通过如同 request 这个第三方工具包来进行数据转发。
- 可以搭建 Mock 服务, 生成配置化数据。
- 可以和数据存储媒介进行通信。

第 2 章将会重点介绍与服务相关的核心包和第三方工具包。

1.3.2 构建

前端随着 Node.js 的诞生, 开始不断地出现构建工具, 来处理打包压缩合并等工程化问题。在命令行显示 Gulp 的帮助 (help) 内容如图 1.1 所示。本书第 3 章中会重点介绍 Gulp。

```
➔ ~ gulp -h

Usage: gulp [options] tasks

选项:
  --help, -h          Show this help. [boolean]
  --version, -v       Print the global and local gulp versions. [boolean]
  --require           Will require a module before running the gulpfile.
                     This is useful for transpilers but also has other
                     applications. [string]
  --gulpfile          Manually set path of gulpfile. Useful if you have
                     multiple gulpfiles. This will set the CWD to the
                     gulpfile directory as well. [string]
```

图 1.1 Gulp 命令行 help

除了 Gulp, 还有很多类似的构建类工具, 例如:

- Grunt——最早的构建类工具, 曾经和 yo、bower 并驾齐驱。