

- ◆ 基于**Angular 7.0**版本
- ◆ 共有**74个**基于知识点的实例和**6个**综合性实例
- ◆ 免费提供实例的**素材**和**源文件**
- ◆ 案例多数来源于真正的**商业项目**，具有高度的参考价值

Broadview[®]
www.broadview.com.cn



Angular

企业级应用开发实战

柳伟卫 著



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>



Angular

企业级应用开发实战

柳伟卫 著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书分为 5 篇。第 1 篇介绍 Angular 的基础概念,使读者对 Angular 有一个初步的印象。第 2 篇介绍 TypeScript 基础知识,并带领读者快速实现一个“用户管理”应用。第 3 篇介绍 Angular 的常用知识点,包括组件、模板、指令、管道、动画和表单等。第 4 篇介绍 Angular 的高级知识点,包括使用 Observable 与 RxJS 实现响应式编程、Angular 模块、依赖注入、通过 HTTP 来访问网络资源、路由与导航、测试、实现国际化、编译与部署 Angular 应用程序、页面静态化和启用缓存。第 5 篇带领读者从零开始实现一个完整的企业级新闻头条客户端应用,使读者具备 Angular 企业级应用开发的完整能力。

本书实例丰富,共有 74 个基于知识点的实例和 6 个综合性实例,将理论讲解最终落实到代码实现上。而且,这些实例会伴随着图书内容的推进,不断趋近于工程化的项目风格,具有很高的应用价值和参考性。

本书由浅及深、层层推进、结构清晰、实例丰富、通俗易懂、实用性强,适合 Angular 的初学者和进阶读者作为自学教程,也适合培训学校作为培训教材,还适合大、中专院校的相关专业作为教学参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Angular 企业级应用开发实战 / 柳伟卫著. —北京: 电子工业出版社, 2019.6

ISBN 978-7-121-36757-1

I. ①A… II. ①柳… III. ①超文本标记语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2019)第 111925 号

责任编辑: 吴宏伟 特约编辑: 田学清

印 刷: 三河市良远印务有限公司

装 订: 三河市良远印务有限公司

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: 28.75 字数: 736 千字

版 次: 2019 年 6 月第 1 版

印 次: 2019 年 6 月第 1 次印刷

定 价: 99.00 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 zlt@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。

前言

写作背景

前端组件化开发是目前主流的开发方式，不管是 Angular、React 还是 Vue.js 都如此。相比较而言，Angular 不管是其开发功能，还是编程思想，在所有前端框架中都是首屈一指的，特别适合企业级应用的开发。

笔者从 2014 年开始关注 Angular 的技术发展，并用 Angular 1.x 开发实际的企业级应用。那时，Angular 还只是实现数据双向绑定的 JS 小框架（所以被叫作 AngularJS）。如今，Angular 已经是 7.0 版本，形成了从开发到部署、从 PC 端到移动端完整流程全覆盖的开放平台。因此，当 Angular 发布 7.0 版本时，笔者便迫不及待地想把这项技术介绍给读者，希望读者能够掌握 Angular 的使用，以简化企业级应用的开发。

内容介绍

本书分为 5 篇。

- “第 1 篇 准备”介绍 Angular 的基础概念，并带领读者快速创建一个 Angular 应用，使读者对 Angular 有一个初步的印象。
- “第 2 篇 入门”介绍 TypeScript 基础，并带领读者快速实现一个简易版本的“用户管理”应用。通过对这部分内容的学习，读者可以了解到 Angular 常用的知识点。
- “第 3 篇 进阶”深入讲解了 Angular 的常用知识点。
- “第 4 篇 高阶”深入讲解了 Angular 的高级知识点。
- “第 5 篇 商业实战”带领读者从零开始实现一个完整的企业级新闻头条客户端应用，使读者具备 Angular 企业级应用开发的完整能力。

本书特点

1. 可与笔者在线上交流

本书提供线上交流网址：

<https://github.com/waylau/angular-enterprise-application-development-samples/issues>

读者有任何技术上的问题，都可以向笔者提问。

2. 提供了 75 个基于知识点的实例和 6 个综合性实例

本书提供了 75 个基于 Angular 知识点的实例，将理论讲解最终落实到代码实现上来。在掌握了 Angular 基础之后，另外提供了 6 个综合性实例。这些实例从零开始，最终实现了一个完整的企业级应用，内容具有很高的应用价值和参考性。

3. 免费提供素材

本书提供了书中实例用到的所有素材。读者可以采用这些素材，完全再现书中的案例效果，如图 1 所示。



图 1 本书案例用到的素材

4. 免费提供书中实例的源文件

本书提供了书中涉及的所有实例的源文件，如图 2 所示。读者可以一边阅读本书，一边参照源文件动手练习，这样不仅提高了学习的效率，而且可以对书中的内容有更加直观的认识，从而逐渐培养自己的编程能力。

5. 覆盖的知识面广

本书几乎囊括了 Angular 7.0 版本中所涉及的全部内容，技术前瞻，案例丰富。不管是 Angular 的初学者，还是编程高手，都能从本书中获益。本书可作为读者案头的工具书，随手翻阅。

ch02/hello-world	update version:
ch03/typescript-demos	update version:
ch04/user-management	update version:
ch05	add * ch05/dynamic-component: 动态加载组件的
ch07	update to v 7.1.0
ch08/pipes	update to v 7.1.0
ch09/animations	update to v 7.1.0
ch10	update to v 7.1.0
ch12/customer-app	update to v 7.1.0
ch13/dependency-injection	update to v 7.1.0
ch14/http-client	update to v 7.1.0
ch15/router	update to v 7.1.0
ch16/testing	完成测试用例
ch17	初始化应用
ch19/universal	add * ch20/service-worker: 使用ServiceWorker的例子
ch20/service-worker	add * ch20/service-worker: 使用ServiceWorker的例子
ch21/news-headlines	add * ch21/news-headlines: News Headlines应用骨架
ch22/news-headlines	update
ch23/news-headlines	update
ch24/news-headlines	update

案例的源文件
读者可以快速
再现结果

图2 本书案例源文件

6. 采用短段、短句，便于流畅阅读

本书采用结构化的层次，并采用短小的段落和语句，让读者读来有顺水行舟的轻快感。

7. 案例的商业性、应用性强

本书提供的案例多数来源于真正的商业项目，具有高度的参考价值。有些代码甚至可以直接移植到自己的项目中，进行重复使用，使从“学”到“用”这个过程变得更加直接。

源代码

本书提供的素材和源代码可从以下网址下载：

<https://github.com/waylau/angular-enterprise-application-development-samples>

勘误和交流

本书如有勘误，会在以下网址发布：

<https://github.com/waylau/angular-enterprise-application-development-samples/issues>

由于笔者能力有限、时间仓促，书中难免有错漏之处，欢迎读者通过以下方式与笔者联系。

- 博客: <https://waylau.com>
- 邮箱: waylau521@gmail.com
- 微博: <http://weibo.com/waylau521>
- GitHub: <https://github.com/waylau>

致谢

感谢电子工业出版社的吴宏伟编辑，在本书写作过程中他审阅了大量稿件，给予了很多指导和帮助。感谢工作在幕后的电子工业出版社评审团队对本书在校对、排版、审核、封面设计、错误改正等方面所给予的帮助，使本书得以顺利出版发行。

感谢我的父母、妻子 Funny 和两个女儿。由于撰写本书，我牺牲了很多陪伴家人的时间。谢谢他们对我的理解和支持。

感谢关心和支持我的朋友、读者、网友。

柳伟卫

2019 年 1 月 7 日于深圳

目 录

第 1 篇 准备

第 1 章 Angular 概述	2	第 2 章 实例 1: 快速创建第一个 Angular 应用	8
1.1 Angular 产生的背景	2	2.1 开发环境准备	8
1.1.1 Angular 与 jQuery 的不同	2	2.1.1 安装 Node.js 和 npm	8
1.1.2 Angular 与 React、Vue.js 优势对比	3	2.1.2 设置 npm 镜像	9
1.1.3 Angular、React、Vue.js 三者怎么选	5	2.1.3 选择合适的 IDE	10
1.2 学习 Angular 需要什么基础	5	2.1.4 安装 Angular CLI	10
1.2.1 具备面向对象思维	5	2.2 创建第一个应用——hello-world	11
1.2.2 熟悉常用的前端开发工具	5	2.2.1 使用 Angular CLI 初始化应用	11
1.2.3 具有一定的前端开发经验	6	2.2.2 运行 Angular 应用	12
1.3 如何使用本书	6	2.3 探索 Angular	13
1.3.1 零基础的读者	6	2.3.1 src 文件夹	13
1.3.2 有后端开发经验的读者	7	2.3.2 根目录	14
1.3.3 有前端开发经验的读者	7	2.3.3 npm 包	16
1.3.4 有 Angular 开发经验的读者	7	2.4 配置 TypeScript 库	18

第 2 篇 入门

第 3 章 TypeScript 基础	21	3.1.4 TypeScript 代码的编译及运行	22
3.1 了解 TypeScript	21	3.2 变量与常量	23
3.1.1 TypeScript 与 JavaScript、ECMAScript 的关系	21	3.2.1 var、let、const 三者的作用域	23
3.1.2 TypeScript 与 Angular 的关系	22	3.2.2 变量与常量的区别	23
3.1.3 使用 TypeScript 的优势	22	3.2.3 什么是变量提升	24
		3.3 TypeScript 的数据类型	25
		3.3.1 基本类型	25

3.3.2	对象类型	30	第 4 章	实例 12: 创建一个“用户管理”应用	55
3.3.3	任意类型	30	4.1	创建应用	55
3.3.4	实例 2: 演示任意类型的使用	31	4.1.1	新建“用户管理”应用	56
3.3.5	联合类型	32	4.1.2	了解 Angular 组件	56
3.3.6	交集类型	32	4.2	修改 AppComponent 组件	56
3.4	强大的面向对象体系	33	4.2.1	修改 app.component.ts 文件	57
3.4.1	类	33	4.2.2	修改 app.component.html 文件	57
3.4.2	接口	37	4.2.3	添加应用级别的样式	57
3.4.3	实例 3: 演示接口的使用	37	4.3	实现用户编辑器	58
3.4.4	泛型	38	4.3.1	创建用户列表组件	58
3.4.5	实例 4: 演示泛型的使用	38	4.3.2	添加 user 属性	60
3.4.6	枚举	39	4.3.3	显示用户	60
3.5	TypeScript 的命名空间	40	4.3.4	显示 UsersComponent 视图	60
3.5.1	声明命名空间	40	4.3.5	创建 User 类	61
3.5.2	实例 5: 声明命名空间	41	4.3.6	编辑用户信息	62
3.5.3	命名空间体	41	4.3.7	添加 FormsModule 模块	63
3.5.4	导入别名声明	42	4.3.8	声明组件	65
3.5.5	实例 6: 导入别名声明	42	4.4	展示用户列表	65
3.5.6	导出声明	43	4.4.1	添加用户列表数据	65
3.5.7	合并声明	43	4.4.2	显示用户列表	66
3.5.8	实例 7: 合并声明	43	4.4.3	用 *ngFor 列出用户	66
3.6	TypeScript 的模块	44	4.4.4	添加样式	67
3.6.1	了解模块	44	4.4.5	添加事件	68
3.6.2	实例 8: 导入声明	45	4.4.6	设置选中的样式	70
3.6.3	实例 9: 导入 Require 声明	45	4.5	多组件化开发	71
3.6.4	实例 10: 导出声明	46	4.5.1	创建 UserDetailComponent 组件	72
3.6.5	实例 11: 导出分配	46	4.5.2	编辑 user-detail.component.html 文件	72
3.6.6	了解 CommonJS 模块	46	4.5.3	编辑 user-detail.component.ts 文件	73
3.6.7	了解 AMD 模式	47	4.5.4	编辑 users.component.html 文件	73
3.7	装饰器	48	4.6	使用服务	74
3.7.1	定义装饰器	49			
3.7.2	了解装饰器的执行时机	49			
3.7.3	认识 4 类装饰器	51			

4.6.1 服务的好处	74	4.7.7 支持路由的 UserDetailComponent 组件	97
4.6.2 创建 UserService 服务	75	4.8 使用 HTTP	100
4.6.3 提供 UserService 服务	76	4.8.1 启用 HTTP 服务	101
4.6.4 修改 UsersComponent 组件	77	4.8.2 模拟数据服务器	102
4.6.5 使用 Observable 数据	78	4.8.3 通过 HTTP 获取用户数据	105
4.6.6 显示消息	79	4.8.4 修改、添加、删除、搜索用户	108
4.7 使用路由	83	4.9 “用户管理”应用技术点总结	119
4.7.1 路由的用处	83	4.9.1 Angular 架构概览	119
4.7.2 创建 AppRoutingModuleModule	84	4.9.2 模块	120
4.7.3 添加路由出口	86	4.9.3 组件	121
4.7.4 添加路由链接	87	4.9.4 模板、指令和数据绑定	121
4.7.5 添加仪表盘视图	89	4.9.5 服务与依赖注入	121
4.7.6 导航用户详情	93	4.9.6 路由	122

第 3 篇 进阶

第 5 章 组件	124	5.3 组件交互方式	133
5.1 数据展示	124	5.3.1 实例 15: 通过 @Input 把数据从父组件传到子组件	133
5.1.1 实例 13: 数据展示的例子	124	5.3.2 实例 16: 通过 setter 监听输入属性值的变化	135
5.1.2 使用插值表达式显示组件属性	126	5.3.3 实例 17: 通过 ngOnChanges() 方法监听输入属性值的变化	136
5.1.3 组件关联模板的两种方式	126	5.3.4 实例 18: 父组件监听子组件的事件	138
5.1.4 在模板中使用指令	127	5.3.5 实例 19: 父组件与子组件通过本地变量交互	140
5.2 生命周期	127	5.3.6 实例 20: 父组件调用 @ViewChild() 方法获取子组件的值	141
5.2.1 生命周期钩子	127	5.3.7 实例 21: 父组件和子组件通过服务来通信	143
5.2.2 实例 14: 生命周期钩子的例子	127	5.4 样式	146
5.2.3 生命周期钩子的调用顺序	129	5.4.1 实例 22: 使用组件样式的例子	146
5.2.4 了解 OnInit 钩子	130		
5.2.5 了解 OnDestroy 钩子	131		
5.2.6 了解 OnChanges 钩子	132		
5.2.7 了解 DoCheck 钩子	132		
5.2.8 了解 AfterView 钩子	133		
5.2.9 了解 AfterContent 钩子	133		

5.4.2	样式的作用域	147	6.5	属性绑定	174
5.4.3	特殊的样式选择器	148	6.5.1	单向输入	174
5.4.4	把样式加载进组件中的几种方式	149	6.5.2	绑定目标	174
5.4.5	控制视图的封装模式	150	6.5.3	一次性字符串初始化	174
5.4.6	了解 Emulated 封装模式	151	6.5.4	选择“插值表达式”还是 “属性绑定”	174
5.5	动态加载组件	152	6.6	attribute、class 和 style 绑定	175
5.5.1	实例 23：动态加载组件的例子	152	6.6.1	attribute 绑定	175
5.5.2	使用指令	157	6.6.2	class 绑定	175
5.5.3	加载及解析组件	158	6.6.3	style 绑定	176
5.5.4	运行效果	159	6.7	事件绑定	176
5.6	自定义元素	160	6.7.1	目标事件	176
5.6.1	使用自定义元素	160	6.7.2	\$event 和事件处理语句	177
5.6.2	自定义元素的工作原理	162	6.7.3	使用 EventEmitter 类自定义事件	177
5.6.3	把组件转换成自定义元素	162	6.8	模板引用变量	178
5.6.4	实例 24：使用自定义元素的例子	163	6.9	输入和输出属性	179
第 6 章	模板	168	第 7 章	指令	180
6.1	模板概述	168	7.1	指令类型	180
6.1.1	模板的语法	168	7.2	属性型指令	180
6.1.2	模板中的 HTML	168	7.2.1	了解 NgClass、NgStyle、NgModel 指令	180
6.2	模板表达式	169	7.2.2	实例 25：创建并使用属性型指令	181
6.2.1	模板表达式上下文	169	7.2.3	实例 26：响应用户引发的事件	183
6.2.2	编写模板表达式的最佳实践	170	7.2.4	实例 27：使用 @Input 数据绑定向 指令传递值	184
6.2.3	管道操作符	170	7.2.5	实例 28：绑定多个属性	186
6.2.4	安全导航操作符和空属性路径	171	7.3	结构型指令	187
6.2.5	非空断言操作符	171	7.3.1	了解 NgIf 指令	187
6.3	模板语句	171	7.3.2	了解 NgSwitch 指令	187
6.3.1	模板语句上下文	172	7.3.3	了解 NgFor 指令	188
6.3.2	编写模板语句的最佳实践	172	7.3.4	了解 <ng-template> 标签	189
6.4	数据绑定	173	7.3.5	了解 <ng-container> 标签	189
6.4.1	从数据源到视图	173			
6.4.2	从视图到数据源	173			
6.4.3	双向绑定	173			

7.3.6 实例 29: 自定义结构型指令	190	9.7.2 实例 37: 基于关键帧的多阶段 动画	207
第 8 章 管道	193	9.8 并行动画组	209
8.1 使用管道	193	9.9 动画回调	210
8.1.1 实例 30: 使用 DatePipe 管道	193	第 10 章 表单	212
8.1.2 实例 31: 使用 UpperCasePipe 管道	194	10.1 模板驱动	212
8.1.3 实例 32: 使用 LowerCasePipe 管道	194	10.1.1 实例 38: 创建表单	213
8.1.4 实例 33: 使用 CurrencyPipe 管道	194	10.1.2 实例 39: 绑定数据	217
8.1.5 实例 34: 使用 PercentPipe 管道	195	10.1.3 实例 40: 提交表单	219
8.2 对管道进行参数化	195	10.2 用户输入	222
8.3 链式管道	196	10.2.1 实例 41: 绑定用户输入	223
8.4 自定义管道	196	10.2.2 实例 42: 通过 \$event 对象取得 用户输入	224
第 9 章 动画	198	10.2.3 实例 43: 从一个模板引用变量中 获得用户输入	225
9.1 实例 35: 一个动画的例子	198	10.2.4 实例 44: Enter 键事件	225
9.1.1 初始化应用	198	10.2.5 实例 45: 失去焦点事件	226
9.1.2 准备工作	199	10.3 表单验证	227
9.1.3 创建组件	199	10.3.1 模板驱动验证	227
9.1.4 运行效果	200	10.3.2 响应式表单验证	228
9.2 状态与转场	200	10.3.3 自定义验证器	231
9.2.1 *状态	201	10.3.4 自定义样式	233
9.2.2 void 状态	202	10.3.5 跨字段交叉验证	234
9.3 进场与离场	202	10.4 响应式表单	236
9.4 Animatable 属性与单位	203	10.4.1 实例 46: 响应式表单的例子	236
9.5 自动属性值计算	204	10.4.2 管理控件的值	239
9.6 动画时间线	205	10.4.3 实例 47: 表单控件分组的例子	240
9.6.1 持续时间	205	10.4.4 实例 48: 嵌套的表单组的例子	243
9.6.2 延迟	205	10.4.5 使用 patchValue() 方法更新部分 模型	246
9.6.3 缓动函数	205	10.4.6 使用 FormBuilder	246
9.6.4 实例 36: 动画时间线的例子	205	10.4.7 使用 FormBuilder 时的表单验证	248
9.7 基于关键帧的多阶段动画	207		
9.7.1 什么是关键帧和偏移量	207		

10.4.8 实例 49: 使用 FormArray 管理动态控件的例子	249	10.5.1 动态表单的优势	251
10.5 动态表单	251	10.5.2 实例 50: 动态表单的例子	252

第 4 篇 高阶

第 11 章 使用 Observable 与 RxJS 实现响应式编程	260	第 12 章 Angular 模块	281
11.1 了解 Observable 机制	260	12.1 模块概述	281
11.1.1 了解基本概念	260	12.1.1 什么是模块化	281
11.1.2 定义观察者	261	12.1.2 认识基本模块	282
11.1.3 执行订阅	262	12.1.3 认识特性模块	283
11.1.4 创建 Observable 对象	263	12.2 引导启动	283
11.1.5 实现多播	264	12.2.1 了解 declarations 数组	284
11.1.6 处理错误	267	12.2.2 了解 imports 数组	285
11.2 了解 RxJS 技术	268	12.2.3 了解 providers 数组	285
11.2.1 创建 Observable 对象的函数	268	12.2.4 了解 bootstrap 数组	285
11.2.2 了解操作符	269	12.3 常用模块	285
11.2.3 处理错误	271	12.3.1 常用模块概述	285
11.3 了解 Angular 中的 Observable	272	12.3.2 BrowserModule 和 CommonModule	286
11.3.1 EventEmitter	273	12.4 特性模块分类	286
11.3.2 HTTP	273	12.4.1 领域特性模块	286
11.3.3 AsyncPipe	274	12.4.2 带路由的特性模块	287
11.3.4 Router	274	12.4.3 路由模块	287
11.3.5 响应式表单	275	12.4.4 服务特性模块	288
11.4 对比 Observable 对象和 Promise	276	12.4.5 可视部件特性模块	288
11.4.1 创建与订阅	276	12.5 入口组件	288
11.4.2 串联	277	12.5.1 引导用的入口组件	288
11.4.3 可取消	277	12.5.2 路由用的入口组件	289
11.4.4 错误处理	277	12.5.3 entryComponents 数组	289
11.4.5 总结对比结果	277	12.5.4 entryComponents 数组和编译器	290
11.5 对比 Observable 对象和事件 API	278	12.6 服务提供商	290
11.6 对比 Observable 对象和数组	279	12.6.1 创建和提供服务	290

12.6.2	服务提供商的作用域	291	13.2.11	可选依赖	315
12.6.3	providedIn 与 NgModule	291	13.3	多级依赖注入器	315
12.6.4	使用惰性加载模块限制服务提供商 的作用域	291	13.3.1	注入器树	316
12.6.5	使用组件限定服务提供商的 作用域	292	13.3.2	注入器冒泡	316
12.6.6	选择模块中的服务还是组件中的 服务	292	13.3.3	在不同层级再次提供同一个 服务	316
12.7	单例服务	293	13.3.4	组件注入器	316
12.7.1	提供单例服务	293	第 14 章	通过 HTTP 访问网络资源	318
12.7.2	了解 forRoot()	293	14.1	初识 HttpClient	318
12.7.3	如何防止重复导入 CoreModule	294	14.2	认识空气质量数据资源	318
12.8	共享模块	295	14.3	实例 54: 使用 HttpClient	320
12.8.1	实例 51: 共享模块的例子	295	14.3.1	导入 HttpClient	320
12.8.2	使用来自其他模块的组件和 服务	296	14.3.2	编写空气质量组件	321
12.9	惰性加载的特性模块	296	14.3.3	编写空气质量服务	321
12.9.1	实例 52: 惰性加载特性模块	296	14.3.4	注入服务到组件中	322
12.9.2	了解 forRoot() 与 forChild()	300	14.3.5	返回带类型检查的响应	324
第 13 章	依赖注入	301	14.3.6	读取完整的响应体	325
13.1	初识依赖注入	301	14.4	实例 55: 错误处理	327
13.2	实例 53: 在 Angular 中实现依赖注入	302	14.4.1	获取错误详情	327
13.2.1	初始应用	303	14.4.2	重试	328
13.2.2	创建服务	305	14.5	实例 56: 请求非 JSON 格式的数据	329
13.2.3	理解注入器	305	14.6	实例 57: 发送数据到服务器	330
13.2.4	理解服务提供商	308	14.6.1	添加请求头	330
13.2.5	注入服务	312	14.6.2	发起 POST 请求	330
13.2.6	单例服务	313	14.6.3	发起 DELETE 请求	331
13.2.7	组件的子注入器	313	14.6.4	发起 PUT 请求	331
13.2.8	测试组件	314	14.7	高级用法	331
13.2.9	服务依赖服务	314	14.7.1	实例 58: 配置请求	331
13.2.10	依赖注入令牌	315	14.7.2	实例 59: 请求去抖	332
			14.7.3	实例 60: 拦截请求和响应	333
			14.7.4	实例 61: 监听进度	334
			14.8	XSRF 防护	335

14.8.1	了解 XSRF 防护	335	16.2.2	配置 Jasmine 和 Karma	355
14.8.2	实例 62: 实现 XSRF 防护	335	16.2.3	启用代码覆盖率报告	355
14.9	实例 63: 测试 HTTP 请求	336	16.2.4	了解测试工具 API	356
14.9.1	采用 Mock 方式	336	16.3	理解组件测试	357
14.9.2	设置环境	336	16.3.1	实例 67: 单独测试组件类	357
14.9.3	期望并回复请求	337	16.3.2	测试组件 DOM	360
14.9.4	测试对错误的预期	337	16.4	实例 68: 测试服务	361
第 15 章	路由与导航	339	16.4.1	测试普通服务	362
15.1	路由器的作用	339	16.4.2	测试带有依赖的服务	362
15.2	基础知识	339	16.4.3	测试 HTTP 服务	363
15.2.1	设置<base>标签关联应用目录	339	16.5	实例 69: 测试属性型指令	364
15.2.2	从路由库中导入路由器	340	16.6	实例 70: 测试管道	366
15.3	实例 64: 配置路由器	340	16.7	调试测试程序	367
15.3.1	配置路由器	340	第 17 章	实现国际化	369
15.3.2	输出导航生命周期中的事件	342	17.1	国际化概述	369
15.4	实例 65: 设置路由出口	342	17.2	实例 71: 通过 i18n 工具来实现国际化	369
15.5	理解路由器链接	343	17.2.1	使用 i18n 属性	370
15.5.1	路由器状态	343	17.2.2	在 i18n 属性中添加描述	370
15.5.2	激活的路由	343	17.2.3	在 i18n 属性中添加意图	370
15.6	路由事件	344	17.2.4	创建翻译源文件	371
15.7	重定向 URL	345	17.2.5	了解翻译源文件	372
15.8	实例 66: 一个路由器的例子	346	17.2.6	创建翻译文件	372
15.8.1	创建应用及组件	346	17.2.7	增加应用配置	373
15.8.2	修改组件的模板	346	17.2.8	运行应用	374
15.8.3	导入并设置路由器	346	17.3	实例 72: 通过 ngx-translate 插件实现国际化	375
15.8.4	添加路由出口	347	17.3.1	添加 ngx-translate 库	375
15.8.5	美化界面	348	17.3.2	导入并配置 ngx-translate 模块	375
15.8.6	定义通配符路由	350	17.3.3	注入 TranslateService 服务	376
第 16 章	测试	352	17.3.4	编写翻译文件	377
16.1	测试准备	352	17.3.5	使用 ngx-translate	378
16.2	认识测试工具	353	17.3.6	运行应用	379
16.2.1	认识 Jasmine 和 Karma	354			

第 18 章 编译与部署 Angular 应用程序	
程序	380
18.1 编译 Angular 应用程序	380
18.1.1 编译概述	380
18.1.2 编译分类	380
18.1.3 为什么需要 AOT 编译	381
18.2 了解 Angular 模板编译器选项	381
18.3 理解 Angular 元数据与 AOT	385
18.3.1 理解元数据的限制	385
18.3.2 理解 AOT 的工作原理	385
18.4 部署 Angular 应用程序	386
18.4.1 最简化的部署方式	386
18.4.2 为生产环境准备的部署方式	387
第 19 章 页面静态化	389
19.1 Universal 概述	389
19.1.1 了解 Universal	389
19.1.2 使用 Universal 的好处	389
19.2 初始化 Universal 示例	390
19.3 实例 73: 使用 Universal	390
19.3.1 安装工具	390
19.3.2 修改根模块 AppModule	391

19.3.3 创建 AppServerModule 模块	391
19.3.4 创建 main 文件	392
19.3.5 创建 AppServerModule 配置文件	392
19.3.6 创建新的构建目标并打包	393
19.3.7 设置服务器环境	393
19.3.8 打包并运行应用	395
第 20 章 启用缓存	397
20.1 了解 Service Worker	397
20.2 实例 74: 使用 Service Worker	398
20.2.1 添加 service-worker 包	398
20.2.2 用 http-server 启动服务器	399
20.2.3 模拟网络故障	400
20.2.4 查看缓存内容	400
20.2.5 修改应用	402
20.2.6 在浏览器中更新应用	402
20.3 了解 Service Worker 的通信机制	403
20.3.1 有可用更新和已激活更新	404
20.3.2 检查更新	404
20.3.3 强制激活更新	405
20.4 配置 Service Worker	405

第 5 篇 商业实战

第 21 章 新闻头条客户端总体设计	409
21.1 应用概述	409
21.2 需求分析	409
21.2.1 首页需求分析	409
21.2.2 新闻详情页面需求分析	410
21.3 架构设计	411
21.3.1 获取访问 API 的密钥	411
21.3.2 了解新闻 API	412
21.4 实例 75: 初始化新闻头条客户端应用	414

21.4.1 修改 app.component.html	414
21.4.2 修改 app.component.ts	414
21.4.3 启动应用	415
第 22 章 实现新闻头条客户端首页	416
22.1 首页需求分析	416
22.2 首页架构设计	416
22.3 实例 76: 实现新闻头条客户端首页	416
22.3.1 添加 Angular Material	417

22.3.2	配置动画	417	23.6.1	监听导航栏单击事件	433
22.3.3	导入组件模块	417	23.6.2	自定义导航切换事件	433
22.3.4	包含一个主题	417	23.6.3	监听导航切换事件	434
22.3.5	创建组件	418	23.6.4	运行应用	435
22.3.6	实现界面原型	418			
22.3.7	查看完整的首页原型效果	422			
第 23 章	实现新闻头条客户端导航栏	424	第 24 章	实现新闻头条客户端新闻详情	
23.1	导航栏概述	424		页面	436
23.2	导航栏需求分析	424	24.1	新闻详情页面概述	436
23.3	导航栏架构设计	424	24.2	新闻详情页面需求分析	436
23.4	实例 77: 实现新闻头条客户端新闻		24.3	新闻详情页面架构设计	437
	分类	425	24.4	实例 80: 实现新闻头条客户端新闻详情	
				页面	437
23.4.1	新建新闻分类服务	425	24.4.1	新建新闻详情组件及服务	437
23.4.2	解析新闻分类数据	425	24.4.2	获取新闻详情数据	437
23.4.3	展示新闻分类	426	24.4.3	展示新闻详情	438
23.5	实例 78: 实现新闻头条客户端新闻		24.4.4	创建首页组件	439
	列表	427	24.4.5	创建路由器	439
23.5.1	新建新闻列表服务	427	24.4.6	修改新闻列表组件	440
23.5.2	解析新闻列表数据	428	24.4.7	接收路由参数	441
23.5.3	展示新闻列表	430	24.4.8	增加“返回”按钮	442
23.6	实例 79: 实现新闻分类与新闻列表		24.4.9	运行应用	442
	组件通信	433	参考文献		444