

Web开发经典丛书

Angular高级编程

(第3版)

源代码为Angular 7，书中范例兼容Angular 7

[美] 亚当·弗雷曼(Adam Freeman) 著
· 睢 丹 译

apress®



清华大学出版社

Web 开发经典丛书

Angular 高级编程

(第 3 版)

[美] 亚当·弗雷曼(Adam Freeman) 著
睢 丹 译

清华大学出版社

北 京

Pro Angular 6, Third Edition

By Adam Freeman

EISBN: 978-1-4842-3648-2

Original English language edition published by Apress Media. Copyright © 2018 by Apress Media. Simplified Chinese-Language edition copyright © 2019 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司授权清华大学出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2019-2479

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Angular 高级编程(第3版)/(美)亚当·弗雷曼(Adam Freeman)著;睢丹译. —北京:清华大学出版社,2019
(Web 开发经典丛书)

书名原文: Pro Angular 6, Third Edition

ISBN 978-7-302-52917-0

I. ①A… II. ①亚… ②睢… III. ①超文本标记语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 083540 号

责任编辑:王 军 韩宏志

装帧设计:孔祥峰

责任校对:成凤进

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 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 190mm×260mm 印 张: 34 字 数: 1251 千字

版 次: 2019 年 6 月第 1 版 印 次: 2019 年 6 月第 1 次印刷

定 价: 138.00 元

产品编号: 082587-01

译者序

在 Web 开发领域，一般使用 HTML 作为前端页面元素的声明式语言，使用 CSS 技术作为展示样式的描述语言，JavaScript 作为业务处理交互的命令式语言。而构建非常复杂的 Web 应用时，纯粹而有限的 HTML 就显得非常不足，JavaScript 本身也会随着项目代码量的膨胀而变得难以维护和管理，研发工期和成本也会随之难以控制。此时，一般使用类库(如 jQuery、Dojo 等)或框架(如 Backbone、Ember 和 ExtJS 等)来提升开发效率，缩短项目的工期和降低成本，方便后续的维护和管理。

而 AngularJS 不仅是一个理念先进的前端开发框架，更是一种端对端的解决方案，已经被用于 Google 的多款产品中。AngularJS 遵从架构设计中的 MVC 模式，提倡展现、数据和逻辑处理组件的松耦合(类似 Flex 和 WPF)。AngularJS 通过指令技术对传统 HTML 实现了自然扩展，通过编译技术实现了数据模型与展现视图的双向自动同步，消除了前端开发中繁杂的 DOM 操作。最后通过模块化设计解决了 JS 代码管理维护和按需加载的问题，提高了广大前端程序员以及后端程序员的生产效率。而且这种解耦本身，也对前端的自动化测试技术提供了良好的支持。

AngularJS 诞生于 2009 年，由 Misko Hevery 和 Adam Abrons 创建，后为 Google 所收购。AngularJS 有许多特性，最为核心的是：MVVM、模块化、自动化双向数据绑定、语义化标签、依赖注入等。AngularJS 是专门为应用程序设计的 HTML。

Angular 2.0 是 2014 年 10 月 22 日在 ng-Europe 会议上发布的，该版本完全重写了 AngularJS。为避免开发人员的困惑，开发团队宣布每个框架都应该使用单独的术语，其中“AngularJS”指的是版本 1。x 版本和没有“JS”的“Angular”指的是版本 2 及更高版本。

Angular 6 于 2018 年 5 月 4 日发布。这是一个重要版本，它对底层框架的关注较少，考虑更多的是工具链，以及如何使 Angular 更易于在未来提高效率，如 ng update、ng add、Angular Elements、Angular Material + CDK Components、Angular Material Starter Components、CLI Workspaces、Library Support、Tree Shakable Providers、Animations Performance Improvements 和 RxJS v6。

Angular 7 让 Angular 变得更出色，为顺应现代趋势增加了比特币挖掘、虚拟滚动、拖放等功能。

本书分为三个部分，每部分涵盖一组相关的主题。本书第 I 部分回顾了一些关键技术的基本知识，包括 HTML、CSS 和 TypeScript，并展示如何构建第一个 Angular 应用程序。第 II 部分介绍 Angular 为构建应用程序提供的构造块，依次描述所有这些构造块。深入描述 Angular 包含的很多内置功能，以及丰富的 Angular 自定义选项。第 III 部分介绍如何使用高级功能来创建更复杂、可扩展的应用程序。演示如何在 Angular 应用程序中进行异步 HTTP 请求，如何使用 URL 路由在应用程序中导航，以及在应用程序的状态发生变化时如何使 HTML 元素具有动画效果。

本书堪称 Angular 领域的里程碑式著作，涵盖了关于 Angular 的几乎所有内容。本书平实、通俗的讲解，丰富、翔实的示例，递进、严密的组织，可以让新手毫无压力地登堂入室，迅速吸取新一代 Web 应用开发的精髓。对于具有相关经验的用户，本书对 Angular 概念和技术细节的全面剖析，以及引人入胜、切中肯綮的讲解，将帮助读者彻底掌握这个框架，在自己职业技术修炼之路上更进一步。

这里要感谢清华大学出版社的编辑们，他们为本书的翻译投入了巨大热情并付出了很多心血。没有他们的帮助和鼓励，本书不可能顺利付梓。

对于这本经典之作，译者本着“诚惶诚恐”的态度，在翻译过程中力求“信、达、雅”，但是鉴于译者水平有限，错误和失误在所难免，如有任何意见和建议，请不吝指正。

作者简介



Adam Freeman 是一位经验丰富的 IT 专业人士，曾在多家公司担任过高级职位，曾经担任一家全球银行的首席技术官和首席运营官。退休之后，他热衷于写作和长跑。

技术评审员简介

Fabio Claudio Ferracchiati 是微软技术领域的高级顾问、高级分析师和高级开发人员。他在 BluArancio 公司 (www.bluarancio.com) 工作。他拥有 .NET MCSD (Microsoft Certified Solution Developer, 微软认证解决方案开发人员)、.NET MCAD (Microsoft Certified Application Developer, 微软认证应用程序开发人员)、MCP (Microsoft Certified Professional, 微软认证专家) 等多项认证，而且是一位多产的作家和技术评审员。十多年来，他为多家国际杂志撰写了大量文章，与他人合著了十多本关于各类计算机主题的书籍。

目 录

第 I 部分 Angular 基础知识

第 1 章 准备工作	3
1.1 需要了解什么	3
1.2 本书结构	3
1.2.1 第 I 部分: Angular 基础知识	3
1.2.2 第 II 部分: Angular 详解	3
1.2.3 第 III 部分: Angular 高级功能	3
1.3 大量示例	4
1.4 获取示例代码	5
1.5 如何搭建开发环境	5
1.6 联系作者	5
1.7 本章小结	5
第 2 章 第一个 Angular 应用程序	7
2.1 准备开发环境	7
2.1.1 安装 Node.js	7
2.1.2 安装 angular-cli 包	8
2.1.3 安装 Git	8
2.1.4 安装编辑器	8
2.1.5 安装浏览器	9
2.2 创建并准备项目	9
2.2.1 创建项目	9
2.2.2 添加 Bootstrap CSS 包	9
2.2.3 启动开发工具	10
2.2.4 编辑 HTML 文件	10
2.3 向项目中添加 Angular 功能	12
2.3.1 准备 HTML 文件	12
2.3.2 创建数据模型	12
2.3.3 创建模板	14
2.3.4 准备组件	14
2.3.5 将应用程序组合起来	16
2.4 向示例应用程序中添加功能	17
2.4.1 添加待办事项表格	17
2.4.2 创建双向数据绑定	19
2.4.3 添加待办事项	21

2.5 本章小结	23
第 3 章 将 Angular 放在上下文中	25
3.1 理解 Angular 的强项	25
3.1.1 往返式应用程序和单页式应用程序	25
3.1.2 Angular 与 jQuery 的比较	26
3.2 比较 Angular、React 和 Vue.js	27
3.3 理解 MVC 模式	27
3.3.1 理解模型	28
3.3.2 理解控制器/组件	29
3.3.3 理解视图/模板	30
3.4 理解 RESTful 服务	30
3.5 常见的设计缺陷	31
3.5.1 将逻辑放错地方	31
3.5.2 数据存储采用的数据格式	32
3.5.3 足够的知识足以制造麻烦	32
3.6 本章小结	32
第 4 章 HTML 和 CSS 入门	33
4.1 准备示例项目	33
4.2 理解 HTML	34
4.2.1 理解空元素	35
4.2.2 理解属性	35
4.2.3 应用无值属性	35
4.2.4 在属性中引用字面量	35
4.2.5 理解元素内容	36
4.2.6 理解文档结构	36
4.3 理解 Bootstrap	37
4.3.1 应用基本的 Bootstrap 类	37
4.3.2 使用 Bootstrap 样式化表格	40
4.3.3 使用 Bootstrap 创建表单	41
4.3.4 使用 Bootstrap 创建网格	42
4.4 本章小结	46
第 5 章 JavaScript 与 TypeScript: 第 1 部分	47
5.1 准备示例项目	48
5.2 使用语句	49

5.3	定义和使用函数	50	7.3.2	创建虚拟数据源	83
5.3.1	定义带参数的函数	51	7.3.3	创建模型存储库	84
5.3.2	定义返回结果的函数	52	7.3.4	创建功能模块	85
5.3.3	将函数用作其他函数的实参	52	7.4	启动商店	85
5.4	使用变量和类型	53	7.4.1	创建 Store 组件和模板	85
5.4.1	使用变量闭包	54	7.4.2	创建商店功能模块	86
5.4.2	使用基本数据类型	54	7.4.3	更新根组件和根模块	87
5.5	使用 JavaScript 操作符	56	7.5	添加商店功能：产品详情	88
5.5.1	使用条件语句	56	7.5.1	显示产品详情	88
5.5.2	相等操作符和恒等操作符	57	7.5.2	添加类别选择	89
5.5.3	显式类型转换	58	7.5.3	添加产品分页功能	90
5.6	处理数组	59	7.5.4	创建自定义指令	92
5.6.1	使用数组字面量	59	7.6	本章小结	95
5.6.2	数组内容的读取和修改	59	第 8 章	SportsStore：订单和结账	97
5.6.3	遍历数组内容	60	8.1	准备示例应用程序	97
5.6.4	spread 操作符	60	8.2	创建购物车	97
5.6.5	使用内置数组方法	61	8.2.1	创建购物车模型	97
5.7	本章小结	62	8.2.2	创建购物车概览组件	98
第 6 章	JavaScript 与 TypeScript：第 2 部分	63	8.2.3	将购物车集成到商店中	100
6.1	准备示例项目	63	8.3	添加 URL 路由	102
6.2	使用对象	63	8.3.1	创建购物车详情和结账组件	102
6.2.1	使用对象字面量	64	8.3.2	创建和应用路由配置	103
6.2.2	将函数用作方法	64	8.3.3	应用程序导航	104
6.2.3	定义类	65	8.3.4	路由守卫	106
6.3	处理 JavaScript 模块	67	8.4	完成购物车详情功能	107
6.4	有用的 TypeScript 特性	70	8.5	处理订单	109
6.4.1	使用类型注解	70	8.5.1	扩展模型	109
6.4.2	使用元组	74	8.5.2	收集订单详情	111
6.4.3	使用可索引类型	74	8.6	使用 RESTful Web 服务	114
6.4.4	使用访问修饰符	74	8.7	本章小结	115
6.5	本章小结	75	第 9 章	SportsStore：管理	117
第 7 章	SportsStore：一个真实的应用程序	77	9.1	准备示例应用程序	117
7.1	准备项目	77	9.1.1	创建模块	117
7.1.1	安装额外的 NPM 软件包	77	9.1.2	配置 URL 路由系统	119
7.1.2	准备 RESTful Web 服务	78	9.1.3	导航到管理 URL	120
7.1.3	准备 HTML 文件	80	9.2	实现身份验证	121
7.1.4	创建文件夹结构	80	9.2.1	理解身份验证系统	121
7.1.5	运行示例应用程序	80	9.2.2	扩展数据源	122
7.1.6	启动 RESTful Web 服务	81	9.2.3	创建身份验证服务	122
7.2	准备 Angular 项目功能	81	9.2.4	启用身份验证	123
7.2.1	更新根组件	81	9.3	扩展数据源和存储库	125
7.2.2	更新根模块	82	9.4	创建管理功能结构	128
7.2.3	检查引导文件	82	9.4.1	创建占位符组件	128
7.3	启动数据模型	83	9.4.2	准备常用内容和功能模块	129
7.3.1	创建模型类	83	9.4.3	实现产品功能	130

9.4.4 实现订单功能	133	12.2 理解单向数据绑定	168
9.5 本章小结	135	12.2.1 理解绑定目标	169
第 10 章 SportsStore: 渐进式功能和部署	137	12.2.2 理解表达式	170
10.1 准备示例应用程序	137	12.2.3 理解括号	171
10.2 添加渐进式特性	137	12.2.4 理解宿主元素	171
10.2.1 安装 PWA 包	137	12.3 使用标准属性和属性绑定	172
10.2.2 缓存数据 URL	137	12.3.1 使用标准属性绑定	172
10.2.3 响应连接的更改	138	12.3.2 使用字符串插入绑定	173
10.3 为部署准备应用程序	140	12.3.3 使用元素属性绑定	174
10.3.1 创建数据文件	140	12.4 设置 CSS 类和样式	174
10.3.2 创建服务器	140	12.4.1 使用类绑定	175
10.3.3 更改存储库类中的 Web 服务 URL	142	12.4.2 使用样式绑定	178
10.4 构建和测试应用程序	142	12.5 更新应用程序的数据	180
10.5 将 SportsStore 应用程序容器化	144	12.6 本章小结	182
10.5.1 安装 Docker	144	第 13 章 使用内置指令	183
10.5.2 准备应用程序	144	13.1 准备示例项目	183
10.5.3 创建 Docker 容器	144	13.2 使用内置指令	185
10.5.4 运行应用程序	145	13.2.1 使用 ngIf 指令	185
10.6 本章小结	146	13.2.2 使用 ngSwitch 指令	187
		13.2.3 使用 ngFor 指令	189
		13.2.4 使用 ngTemplateOutlet 指令	195
		13.3 理解单向数据绑定的限制	197
		13.3.1 使用幂等表达式	197
		13.3.2 理解表达式上下文	199
		13.4 本章小结	201
		第 14 章 使用事件和表单	203
		14.1 准备示例项目	203
		14.1.1 导入表单模块	203
		14.1.2 准备组件和模板	204
		14.2 使用事件绑定	205
		14.2.1 理解动态定义的属性	206
		14.2.2 使用事件数据	208
		14.2.3 使用模板引用变量	209
		14.3 使用双向数据绑定	210
		14.4 处理表单	212
		14.4.1 向示例应用程序添加表单	213
		14.4.2 添加表单数据验证	214
		14.4.3 验证整个表单	221
		14.5 使用基于模型的表单	226
		14.5.1 启用基于模型的表单功能	226
		14.5.2 定义表单模型类	226
		14.5.3 使用模型进行验证	229
		14.5.4 根据模型生成元素	231
		14.6 创建自定义表单验证器	232
		14.7 本章小结	234
第 II 部分 Angular 详解			
第 11 章 创建 Angular 项目	149		
11.1 创建新的 Angular 项目	149		
11.2 了解项目结构	150		
11.2.1 了解 src 文件夹	151		
11.2.2 了解包文件夹	152		
11.3 使用开发工具	154		
11.3.1 了解开发 HTTP 服务器	155		
11.3.2 了解热模型替换	155		
11.3.3 使用 linter	156		
11.4 理解 Angular 应用程序是如何工作的	158		
11.4.1 理解 HTML 文档	158		
11.4.2 理解应用程序引导	158		
11.4.3 理解 Angular 根模块	159		
11.4.4 理解 Angular 组件	160		
11.4.5 理解内容显示	160		
11.5 在 Angular 项目中开始开发	161		
11.5.1 添加 Bootstrap CSS 框架	161		
11.5.2 创建数据模型	161		
11.5.3 创建模板和根组件	164		
11.5.4 配置根 Angular 模块	165		
11.6 本章小结	165		
第 12 章 使用数据绑定	167		
12.1 准备示例项目	167		

第 15 章 创建属性指令.....235

15.1 准备示例项目.....235

15.2 创建简单的属性指令.....237

15.3 在指令中访问应用程序数据.....239

15.3.1 读取宿主元素属性.....239

15.3.2 创建数据绑定输入属性.....241

15.3.3 响应输入属性的变化.....243

15.4 创建自定义事件.....244

15.5 创建宿主元素绑定.....247

15.6 在宿主元素上创建双向绑定.....248

15.7 导出指令用于模板变量.....250

15.8 本章小结.....251

第 16 章 创建结构型指令.....253

16.1 准备示例项目.....253

16.2 创建简单的结构型指令.....254

16.2.1 实现结构型指令类.....255

16.2.2 启用结构型指令.....257

16.2.3 使用结构型指令的简洁语法.....258

16.3 创建迭代结构型指令.....259

16.3.1 提供额外的上下文数据.....261

16.3.2 使用简洁的结构语法.....262

16.3.3 处理属性级数据变更.....263

16.3.4 处理集合级数据变更.....264

16.4 查询宿主元素内容.....271

16.4.1 查询多个子内容.....274

16.4.2 接收查询变更通知.....275

16.5 本章小结.....276

第 17 章 理解组件.....277

17.1 准备示例项目.....278

17.2 使用组件来组织应用程序.....278

17.2.1 创建新组件.....279

17.2.2 定义模板.....282

17.2.3 完成组件的重组.....289

17.3 使用组件样式.....289

17.3.1 定义外部组件样式.....290

17.3.2 使用高级样式特性.....291

17.4 查询模板内容.....296

17.5 本章小结.....298

第 18 章 使用和创建管道.....299

18.1 准备示例项目.....299

18.2 理解管道.....302

18.3 创建一个自定义管道.....303

18.3.1 注册自定义管道.....303

18.3.2 应用自定义管道.....304

18.3.3 组合管道.....305

18.3.4 创建非纯管道.....306

18.4 使用内置管道.....309

18.4.1 格式化数值.....309

18.4.2 格式化货币值.....311

18.4.3 格式化百分比.....313

18.4.4 格式化日期.....314

18.4.5 改变字符串大小写.....317

18.4.6 将数据序列化为 JSON 数据.....317

18.4.7 将数据数组切片.....318

18.5 本章小结.....319

第 19 章 使用服务.....321

19.1 准备示例项目.....321

19.2 理解对象分发问题.....322

19.2.1 问题的提出.....322

19.2.2 利用依赖注入将对象作为服务分发.....326

19.2.3 在其他构造块中声明依赖.....330

19.3 理解测试隔离问题.....335

19.4 完成服务的融入.....338

19.4.1 更新根组件和模板.....338

19.4.2 更新子组件.....339

19.5 本章小结.....340

第 20 章 使用服务提供程序.....341

20.1 准备示例项目.....342

20.2 使用服务提供程序.....343

20.2.1 使用类提供程序.....345

20.2.2 使用值提供程序.....350

20.2.3 使用工厂提供程序.....351

20.2.4 使用已有的服务提供程序.....353

20.3 使用本地提供程序.....354

20.3.1 理解单个服务对象的局限性.....354

20.3.2 在组件中创建本地提供程序.....355

20.3.3 理解服务提供程序的替代方案.....357

20.3.4 控制依赖解析.....360

20.4 本章小结.....361

第 21 章 使用和创建模块.....363

21.1 准备示例项目.....363

21.2 理解根模块.....365

21.2.1 理解 imports 属性.....366

21.2.2 理解 declarations 属性.....366

21.2.3 理解 providers 属性.....367

21.2.4 理解 bootstrap 属性.....367

21.3 创建功能模块.....368

21.3.1 创建模型模块.....369

21.3.2 创建实用工具功能模块	373
21.3.3 用组件创建一个功能模块	377
21.4 本章小结	380

第III部分 Angular 高级功能

第22章 创建示例项目	383
22.1 启动示例项目	383
22.1.1 添加和配置 Bootstrap CSS 包	383
22.1.2 创建项目结构	383
22.2 创建模型模块	384
22.2.1 创建产品数据类型	384
22.2.2 创建数据源和存储库	384
22.2.3 完成模型模块	385
22.3 创建核心模块	385
22.3.1 创建共享状态服务	386
22.3.2 创建表格组件	386
22.3.3 创建表单组件	387
22.4.4 完成核心模块	389
22.4 创建消息模块	389
22.4.1 创建消息模型和服务	389
22.4.2 创建组件和模板	390
22.4.3 完成消息模块	390
22.5 完成项目	391
22.6 本章小结	392
第23章 使用 Reactive Extensions	393
23.1 准备示例项目	394
23.2 理解问题	394
23.3 使用 Reactive Extensions 解决问题	396
23.3.1 理解 Observable	396
23.3.2 理解 Observer	398
23.3.3 理解 Subject	399
23.4 使用 async 管道	400
23.5 扩展应用程序功能模块	402
23.6 更进一步	404
23.6.1 过滤事件	404
23.6.2 转换事件	405
23.6.3 只接收不同的事件	407
23.6.4 获取和忽略事件	409
23.7 本章小结	410
第24章 生成异步 HTTP 请求	411
24.1 准备示例项目	411
24.1.1 配置模型功能模块	412
24.1.2 创建数据文件	412

24.1.3 更新表单组件	413
24.1.4 运行示例项目	413
24.2 理解 RESTful Web 服务	414
24.3 替换静态数据源	414
24.3.1 创建新的数据源服务	414
24.3.2 配置数据源	416
24.3.3 使用 REST 数据源	416
24.3.4 保存和删除数据	417
24.4 加强 HTTP 请求	419
24.5 生成跨域请求	420
24.6 配置请求头	422
24.7 处理错误	424
24.7.1 生成用户可使用的消息	425
24.7.2 处理错误	426
24.8 本章小结	427
第25章 路由与导航: 第1部分	429
25.1 准备示例项目	429
25.2 开始学习路由	431
25.2.1 创建路由配置	431
25.2.2 创建路由组件	433
25.2.3 更新根模块	433
25.2.4 完成配置	433
25.2.5 添加导航链接	434
25.2.6 理解路由的效果	436
25.3 完成路由实现	437
25.3.1 在组件中处理路由变化	438
25.3.2 使用路由参数	439
25.3.3 在代码中导航	444
25.3.4 接收导航事件	445
25.3.5 删除事件绑定和支持代码	446
25.4 本章小结	448
第26章 路由与导航: 第2部分	449
26.1 准备示例项目	449
26.2 使用通配符和重定向	454
26.2.1 在路由中使用通配符	454
26.2.2 在路由中使用重定向	455
26.3 在组件内部导航	456
26.3.1 响应正在发生的路由变化	457
26.3.2 为活动路由设置不同样式的链接	459
26.3.3 修复 All 按钮	461
26.4 创建子路由	462
26.4.1 创建子路由出口	463
26.4.2 从子路由访问参数	464
26.5 本章小结	467

第 27 章 路由与导航: 第 3 部分	469	28.4 理解元素过渡	505
27.1 准备示例项目	469	28.4.1 为内置状态创建过渡	505
27.2 守卫路由	470	28.4.2 控制动画的过渡	506
27.2.1 使用解析器推迟导航	470	28.5 理解动画样式组	510
27.2.2 避免带有守卫的导航	476	28.5.1 在可重用的分组中定义公共样式	510
27.3 动态加载功能模块	484	28.5.2 使用元素变形	511
27.3.1 创建一个简单的功能模块	484	28.5.3 应用 CSS 框架样式	512
27.3.2 动态加载模块	485	28.6 本章小结	514
27.3.3 守卫动态模块	488	第 29 章 Angular 单元测试	515
27.4 指定命名出口	490	29.1 准备示例项目	516
27.4.1 创建附加的出口元素	490	29.2 创建一个简单的单元测试	517
27.4.2 在使用多个出口的情况下导航	491	29.3 使用 Jasmine 完成单元测试	518
27.5 本章小结	493	29.4 测试 Angular 组件	519
第 28 章 使用动画	495	29.4.1 使用 TestBed 类完成工作	519
28.1 准备示例项目	496	29.4.2 测试数据绑定	522
28.1.1 禁用 HTTP 延迟	496	29.4.3 测试带有外部模板的组件	523
28.1.2 简化表格模板和路由配置	496	29.4.4 测试组件事件	525
28.2 开始学习 Angular 动画	498	29.4.5 测试输出属性	526
28.2.1 启用动画模块	498	29.4.6 测试输入属性	528
28.2.2 创建动画	499	29.4.7 测试异步操作	529
28.2.3 应用动画	501	29.5 测试 Angular 指令	531
28.2.4 测试动画效果	503	29.6 本章小结	532
28.3 理解内置的动画状态	504		

第 I 部分



Angular 基础知识

- 第 1 章 准备工作
- 第 2 章 第一个 Angular 应用程序
- 第 3 章 将 Angular 放在上下文中
- 第 4 章 HTML 和 CSS 入门
- 第 5 章 JavaScript 与 TypeScript: 第 1 部分
- 第 6 章 JavaScript 与 TypeScript: 第 2 部分
- 第 7 章 SportsStore: 一个真实的应用程序
- 第 8 章 SportsStore: 订单和结账
- 第 9 章 SportsStore: 管理
- 第 10 章 SportsStore: 渐进式功能和部署

第 1 章



准备工作

Angular 将服务器端开发领域的一些最佳实践用于增强浏览器中的 HTML，为更加简便地构建富应用程序(rich application)打下良好基础。Angular 应用程序围绕模型-视图-控制器(Model-View-Controller, MVC)设计模式构建，该模式的重点在于创建具有如下特点的应用程序：

- **可扩展：**一旦理解 Angular 的基本原理，即便是复杂的 Angular 应用程序，也很容易弄明白其运行方式，而这意味着可以轻易地改进应用程序，为用户创建新的有用功能。
- **可维护：**Angular 应用程序易于调试和修复，这意味着长期维护工作得以简化。
- **可测试：**Angular 对单元测试和端到端测试的支持都非常好，这意味着可以先于用户发现并修复缺陷。
- **兼容标准：**Angular 建立在 Web 浏览器的固有功能上，但是它能实现的功能并未受到这些固有功能的限制，而是能够创建可兼容标准的 Web 应用程序，能够利用最新功能(如 HTML5 API)和流行的工具与框架。

Angular 是由 Google 赞助和维护的开源 JavaScript 库。它已用于一些最大和最复杂的 Web 应用程序。本书将展示在自己的项目中充分利用 Angular 所需要知道的一切。

1.1 需要了解什么

在阅读本书之前，读者应该熟悉 Web 开发的基础知识，了解 HTML 和 CSS 的工作原理，最好熟悉 JavaScript。如果对这些细节不是非常清楚，那么本书将在第 4~6 章中温习一下如何使用 HTML、CSS 和 JavaScript。但是本书不会给出 HTML 元素和 CSS 属性的全面参考。一本关于 Angular 的书不可能涵盖 HTML 的方方面面。

1.2 本书结构

本书分为三个部分，每部分涵盖一组相关的主题。

1.2.1 第 I 部分：Angular 基础知识

本书的第 I 部分为阅读本书其余部分提供了准备信息。它包括本章内容，回顾了一些关键技术的基本知识，包括 HTML、CSS 和 TypeScript(JavaScript 的一个超集，用于 Angular 开发)。这部分还将展示如何构建第一个 Angular 应用程序，并引导完成构建更真实的应用程序(名为 SportsStore)的过程。

1.2.2 第 II 部分：Angular 详解

本书的第 II 部分介绍 Angular 为构建应用程序提供的构造块，依次描述所有这些构造块。Angular 包含很多内置的功能，这部分将深入描述这些功能，此外 Angular 还提供丰富的自定义选项，而这部分也将展示所有这些功能。

1.2.3 第 III 部分：Angular 高级功能

本书的第 III 部分介绍如何使用高级功能来创建更复杂、可扩展的应用程序。这部分演示如何在 Angular 应用程序中进行异步 HTTP 请求，如何使用 URL 路由在应用程序中导航，以及在应用程序的状态发生变化时如何使 HTML 元素具有动画效果。

1.3 大量示例

本书包含大量示例。学习 Angular 的最好方法就是通过示例，本书将尽可能多的示例打包进来。为使本书中的示例尽可能多，本书采用一个简单的约定来避免反复地列出文件的内容。当在某一章中首次使用一个文件时，将列出该文件的完整内容，如代码清单 1-1 所示。代码清单的标题中将包含文件的名称，以及应该在哪个文件夹中创建该文件。更改代码时，将以粗体显示修改过的语句。

代码清单 1-1 完整的示例文档

```
import { NgModule } from "@angular/core";
import { BrowserModule } from "@angular/platform-browser";
import { ProductComponent } from "../component";
import { FormsModule, ReactiveFormsModule } from "@angular/forms";
import { PaAttrDirective } from "../attr.directive";

@NgModule({
  imports: [BrowserModule, FormsModule, ReactiveFormsModule],
  declarations: [ProductComponent, PaAttrDirective],
  bootstrap: [ProductComponent]
})
export class AppModule { }
```

代码清单 1-1 摘自第 15 章。这里暂不关注它的具体作用，只需要注意，这是一个完整清单，显示了文件的所有内容。对同一个文件进行一系列修改，或对大文件进行较小修改时，只展示发生改变的元素，以创建部分代码清单。部分代码清单的特征就是以英文省略号(...)开始和结尾，如代码清单 1-2 所示。

代码清单 1-2 部分代码清单

```
...
<table class="table table-sm table-bordered table-striped">
  <tr><th></th><th>Name</th><th>Category</th><th>Price</th></tr>
  <tr *ngFor="let item of getProducts(); let i = index" pa-attr>
    <td>{{i + 1}}</td>
    <td>{{item.name}}</td>
    <td pa-attr pa-attr-class="bg-warning">{{item.category}}</td>
    <td pa-attr pa-attr-class="bg-info">{{item.price}}</td>
  </tr>
</table>
...
```

代码清单 1-2 也是摘自第 15 章的后续代码清单。可以看到，这里只显示了 table 元素及其内容，并且突出显示了一些语句。希望借此引起读者的注意，关注示例的这一部分，以说明所描述的功能或技术。某些情况下，需要对同一个文件的不同部分进行更改，此时为简洁起见，需要省略一些元素或语句，如代码清单 1-3 所示。

代码清单 1-3 为简洁起见而省略部分语句

```
import { ApplicationRef, Component } from "@angular/core";
import { Model } from "../repository.model";
import { Product } from "../product.model";
import { ProductFormGroup } from "../form.model";

@Component({
  selector: "app",
  templateUrl: "app/template.html"
})
export class ProductComponent {
  model: Model = new Model();
  form: ProductFormGroup = new ProductFormGroup();

  // ...other members omitted for brevity...

  showTable: boolean = true;
}
```

利用这个约定可在本书中放入更多示例，但这也意味着很难定位某项技术。为此，在第 II 部分和第 III 部分中描述 Angular 特性的所有章节都从一个内容摘要表格开始，描述该章包含的技术以及演示如何使用它们的代码清单。

1.4 获取示例代码

可以从 <https://github.com/Apress/pro-angular-6/tree/master/Update%20for%20Angular%207> 下载本书所有章的示例项目。这些示例项目都是针对 Angular 7 更新的代码，可以免费下载，其中包含用来重新创建示例需要的所有支持资源，这样就不必辛苦敲入这些代码。虽然不是一定要下载这些代码，但这是对示例进行实验的最简单方法。

请读者注意，本书对应的英文书籍是 *Pro Angular 6*。在翻译期间，Angular 升级到 Angular 7，因此本书的一部分代码也随之更新。由于 Angular 7 丢弃了 Angular 6 的个别功能，因此中文书籍特别更新了代码清单 14-26、14-27、14-28、15-1、15-2、15-4、15-6、15-21、16-15、16-17、17-14、17-15、17-22、17-24、17-25、19-14、19-27、20-26、20-27、20-30。删除了英文书中的代码清单 15-19。

如果读者想要查看 Angular 6 对应的代码，可访问 <http://github.com/Apress/pro-angular-6>。

读者也可扫描本书封底的二维码获取代码。但请注意，由于代码时常更新，要获取完整详明的代码，请访问上述网址下载。

1.5 如何搭建开发环境

第2章通过创建一个简单的应用程序来介绍 Angular，在介绍过程中，将说明如何建立使用 Angular 的开发环境。

1.6 联系作者

如果在运行本章中示例代码的过程中遇到问题，或者发现书中存在问题，请发电子邮件到 adam@adam-freeman.com，作者将竭尽所能提供帮助。

1.7 本章小结

本章概述了本书的内容和结构。学习 Angular 开发的最佳方法就是通过示例，因此，下一章将直奔主题，展示如何设置开发环境，并使用该环境创建第一个 Angular 应用程序。

