

Web开发经典丛书

Pro Angular, Second Edition

Angular 5高级编程

(第2版)

学习如何编程驾驭现代浏览器

[美] Adam Freeman 著
杨明军 颜 炯 等译

Apress®



清华大学出版社

Web 开发经典丛书

Angular 5 高级编程

(第 2 版)

[美] Adam Freeman 著

杨明军 颜炯 等译



清华大学出版社

北京

Pro Angular, Second Edition

Adam Freeman

EISBN: 978-1-4842-2306-2

Original English language edition published by Apress Media. Copyright © 2017 by Apress Media.

Simplified Chinese-Language edition copyright © 2018 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：01-2017-5757

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 5高级编程：第2版 / (美)亚当·弗里曼(Adam Freeman) 著；杨明军 等译. —北京：清华大学出版社，2018 (2018.7重印)

(Web 开发经典丛书)

书名原文：Pro Angular, Second Edition

ISBN 978-7-302-49117-0

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

中国版本图书馆 CIP 数据核字(2017)第 315412 号

责任编辑：王 军 李维杰

装帧设计：孔祥峰

责任校对：曹 阳

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：三河市君旺印务有限公司

经 销：全国新华书店

开 本：185mm×260mm 印 张：47.5 字 数：1126 千字

版 次：2018 年 1 月第 1 版 印 次：2018 年 7 月第 2 次印刷

定 价：128.00 元

产品编号：076390-01

译者序

在欣欣向荣的 Web 前端领域，新的框架层出不穷，如此多的选择给开发者带来极大的痛苦：学习难度如何？开发效率如何？运行性能如何？可维护性高吗？还有开发社区、发展前景等因素需要考虑。评估任何一个框架不仅要看其优势，还要看它给开发者挖的坑有多深。Angular 框架自最初的 AngularJS，迭代到最新的 Angular 5，其不变的最显著特征就是其整合性。作为一个一体化框架，Angular 经过精心的前期设计，涵盖了开发中的各个层面，层与层之间都经过精心调适。开发者借助这个“开箱即用”的框架，就可以完成大部分的前端开发工作，而不需要费时费力去组合、评估其他前端技术。这可以有效降低开发者和开发团队的决策成本，有利于项目的快速起步。

Angular 5 是构建动态 JavaScript 应用程序的领先框架，而想讲清楚这个包罗万象的一站式框架的各项强大功能是一项很大的挑战。Adam Freeman 撰著的《Angular 5 高级编程(第 2 版)》详细描述如何在项目中使用 Angular，从最基本的构造块开始，到最终构建最先进和复杂的功能，深入讲解开发者所需要掌握的所有知识。本书讲解 Angular 框架的面面俱到，同时也不放过任何有用的细节，特别是对于一些最重要的 Angular 功能，本书都给出了常见问题以及解决办法。

即便是一位希望从事 Web 前端开发工作的新手，考虑到 Angular 框架陡峭的学习曲线，也不要担心，本书由易到难的循序渐进式风格可以极大地降低学习 Angular 框架的门槛，降低学习痛苦指数。而对于经验丰富的前端老兵，尽管具有 Angular 框架使用经验，本书也值得推荐，因为本书内容已经升级到 Angular 5，能助你快速将项目升级到 Angular 框架的最新版，充分利用最新框架的强大功能。

本书由杨明军、颜炯翻译。此外，参与本书翻译的还有肖国尊、胡季红、李辉、马蓉、李新军、易民全、姚建军、鲍春雷、甘信生、郝雪松、凌栋、王发云、王继云、赵建军、朱宝庆、朱钱。Be Flying 工作室负责人肖国尊负责本书翻译质量和进度的控制与管理。敬请广大读者提供反馈意见，我们的邮箱是 215565222@qq.com。我们会仔细阅读读者发来的每一封邮件，尽快回应读者的问题。



作者简介

Adam Freeman 是一位经验丰富的 IT 专业人士，曾在一系列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。

Adam Freeman 是一位经验丰富的 IT 专业人士，曾在一系列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。他拥有多项认证，包括 Microsoft Certified Application Developer、微软认证应用程序开发人员、IBM Professional Certified Professional。此外，他还拥有多项认证，包括 IBM Professional Certified Professional。多年来，他为多家企业撰写了技术文档，并出版了多本关于 IT 管理的书籍。



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, 而且还是一位多产的作家和技术评审员。十多年来, 他为多家意大利和国际杂志撰写了大量文章, 与人合著了十多本关于各类计算机主题的书籍。

25
27
29
31
31
22
23
34
35
35
6
34
39
40
40
42
44
44
44
44
45
48
50

目 录

第 1 章 准备工作1	
1.1 需要了解什么.....1	
1.2 本书结构.....1	
1.2.1 第 1 部分：准备.....2	
1.2.2 第 2 部分：使用 Angular.....2	
1.2.3 第 3 部分：Angular 的 高级功能.....2	
1.3 大量示例.....2	
1.4 获取示例代码.....4	
1.5 如何搭建开发环境.....4	
1.6 联系作者.....4	
1.7 本章小结.....4	
第 2 章 第一个 Angular 应用程序5	
2.1 准备开发环境.....5	
2.1.1 安装 Node.js.....5	
2.1.2 安装 angular-cli 包.....6	
2.1.3 安装 Git.....6	
2.1.4 安装编辑器.....7	
2.1.5 安装浏览器.....7	
2.2 创建并准备项目.....8	
2.2.1 创建项目.....8	
2.2.2 创建包文件.....8	
2.2.3 启动服务器.....10	
2.2.4 编辑 HTML 文件.....11	
2.3 向项目中添加 Angular 功能.....13	
2.3.1 准备 HTML 文件.....14	
2.3.2 创建数据模型.....14	

2.3.3 创建模板.....17	
2.3.4 创建组件.....17	
2.3.5 将应用程序组合起来.....19	
2.4 向示例应用程序中添加功能.....21	
2.4.1 添加待办事项表格.....21	
2.4.2 创建双向数据绑定.....25	
2.4.3 添加待办事项.....27	
2.5 本章小结.....29	
第 3 章 创建项目31	
3.1 准备创建项目.....31	
3.2 创建项目.....32	
3.2.1 创建项目的 Angular 部分.....33	
3.2.2 创建项目的 ASP.NET Core MVC 部分.....34	
3.3 配置项目.....35	
3.3.1 准备项目(Visual Studio).....35	
3.3.2 创建和编辑配置文件.....36	
3.3.3 启用日志消息.....39	
3.3.4 更新 Bootstrap 包.....39	
3.3.5 移除文件.....40	
3.3.6 更新控制器、布局和视图.....40	
3.4 运行项目.....42	
3.5 理解合并项目和工具.....44	
3.5.1 理解项目结构.....44	
3.5.2 理解工具集成.....44	
3.5.3 理解项目变更系统.....45	
3.5.3 检测 TypeScript 错误.....48	
3.6 本章小结.....50	

第 4 章 HTML 和 CSS 入门	51
4.1 准备示例项目.....	51
4.2 理解 HTML.....	53
4.2.1 理解空元素.....	54
4.2.2 理解属性.....	54
4.2.3 应用无值属性.....	54
4.2.4 在属性中引用字面量.....	55
4.2.5 理解元素内容.....	55
4.2.6 理解文档结构.....	55
4.3 理解 Bootstrap.....	57
4.3.1 应用基本的Bootstrap类.....	57
4.3.2 使用Bootstrap样式化表格.....	61
4.3.3 使用Bootstrap创建表单.....	62
4.3.4 使用Bootstrap创建网格.....	63
4.4 本章小结.....	68
第 5 章 JavaScript 与 TypeScript:	
第 1 部分	69
5.1 准备示例项目.....	70
5.1.1 创建 HTML 和 JavaScript	
文件.....	71
5.1.2 配置 TypeScript 编译器.....	72
5.1.3 运行示例项目.....	72
5.2 理解 script 元素.....	73
5.2.1 使用 JavaScript 模块	
加载器.....	73
5.2.2 理解基本 workflow.....	74
5.3 使用语句.....	75
5.4 定义和使用函数.....	75
5.4.1 定义带参数的函数.....	77
5.4.2 定义返回结果的函数.....	79
5.4.3 将函数用作其他函数的	
实参.....	79
5.5 使用变量和类型.....	80
5.6 使用 JavaScript 操作符.....	84
5.6.1 使用条件语句.....	84
5.6.2 相等操作符和恒等操作符.....	85
5.6.3 显式类型转换.....	86

5.7 处理数组.....	87
5.7.1 使用数组字面量.....	88
5.7.2 数组内容的读取和修改.....	88
5.7.3 遍历数组内容.....	89
5.7.4 使用内置数组方法.....	89
5.8 本章小结.....	91

第 6 章 JavaScript 与 TypeScript:	
第 2 部分	93
6.1 准备示例项目.....	93
6.2 使用对象.....	94
6.2.1 使用对象字面量.....	95
6.2.2 将函数用作方法.....	95
6.2.3 定义类.....	96
6.3 处理 JavaScript 模块.....	99
6.3.1 创建模块.....	99
6.3.2 从 JavaScript 模块导入.....	100
6.4 有用的 TypeScript 特性.....	103
6.4.1 使用类型注解.....	103
6.4.2 使用元组.....	109
6.4.3 使用可索引类型.....	109
6.4.4 使用访问修饰符.....	110
6.5 本章小结.....	111
第 7 章 SportsStore: 一个真实的应用程序	113
7.1 准备项目.....	113
7.1.1 准备项目.....	114
7.1.2 创建文件夹结构.....	114
7.1.3 安装额外的 NPM 软件包.....	114
7.1.4 准备 RESTful Web 服务.....	116
7.1.5 准备 HTML 文件.....	118
7.1.6 运行示例应用程序.....	119
7.1.7 启动 RESTful Web 服务.....	119
7.2 准备 Angular 项目功能.....	120
7.2.1 更新根组件.....	120
7.2.2 更新根模块.....	121
7.2.3 检查引导文件.....	121
7.3 启动数据模型.....	122

7.3.1	创建模型类	122	9.2	实现身份验证	175
7.3.2	创建虚拟数据源	123	9.2.1	理解身份验证系统	175
7.3.3	创建模型存储库	124	9.2.2	扩展数据源	176
7.3.4	创建功能模块	126	9.2.3	创建身份验证服务	177
7.4	启动商店	126	9.2.4	启用身份验证	178
7.4.1	创建 Store 组件和模板	127	9.3	扩展数据源和存储库	181
7.4.2	创建商店功能模块	128	9.4	创建管理功能结构	185
7.4.3	更新根组件和根模块	129	9.4.1	创建占位符组件	185
7.5	添加商店功能: 产品详情	130	9.4.2	准备常用内容和功能 模块	186
7.5.1	显示产品详情	130	9.4.3	实现产品功能	189
7.5.2	添加类别选择	132	9.4.4	实现订单功能	193
7.5.3	添加产品分页功能	133	9.5	本章小结	195
7.5.4	创建自定义指令	137			
7.6	本章小结	140			
第 8 章	SportsStore: 订单和结账	141	第 10 章	SportsStore: 部署	197
8.1	准备示例应用程序	141	10.1	准备部署应用程序	197
8.2	创建购物车	141	10.2	将 SportsStore 应用程序 容器化	197
8.2.1	创建购物车模型	141	10.2.1	安装 Docker	197
8.2.2	创建购物车概览组件	143	10.2.2	准备应用程序	198
8.2.3	将购物车集成到商店中	145	10.2.3	创建 Docker 容器	199
8.3	添加 URL 路由	148	10.2.4	运行应用程序	200
8.3.1	创建购物车详情和结账 组件	148	10.3	本章小结	201
8.3.2	创建和应用路由配置	150			
8.3.3	应用程序导航	151	第 11 章	创建 Angular 项目	203
8.3.4	守卫路由	154	11.1	准备 TypeScript Angular 开发项目	203
8.4	完成购物车详情功能	156	11.1.1	创建项目文件夹结构	204
8.5	处理订单	159	11.1.2	创建和提供 HTML 文档	204
8.5.1	扩展模型	159	11.1.3	准备项目配置	205
8.5.2	收集订单详情	161	11.1.4	添加包	205
8.6	使用 RESTful Web 服务	165	11.1.5	启动监视进程	213
8.7	本章小结	168	11.2	使用 TypeScript 开始 Angular 开发	214
第 9 章	SportsStore: 管理	169	11.2.1	创建数据模型	216
9.1	准备示例应用程序	169	11.2.2	创建模板和根组件	219
9.1.1	创建模块	169	11.2.3	创建 Angular 模块	220
9.1.2	配置 URL 路由系统	172	11.2.4	引导应用程序	221
9.1.3	导航到管理 URL	173			

11.2.5	配置 JavaScript 模块	222
11.2.6	更新 HTML 文档	226
11.2.7	运行应用程序	227
11.3	利用@angular/cli 工具	54
	创建项目	229
11.3.1	创建示例项目	229
11.3.2	创建包文件	230
11.3.3	启动服务器	231
11.4	本章小结	231
第 12 章	使用数据绑定	233
12.1	准备示例项目	234
12.2	理解单向数据绑定	234
12.2.1	理解绑定目标	236
12.2.2	理解表达式	237
12.2.3	理解括号	238
12.2.4	理解宿主元素	239
12.3	使用标准属性和属性绑定	239
12.3.1	使用标准属性绑定	239
12.3.2	使用字符串插入绑定	241
12.3.3	使用元素属性绑定	242
12.4	设置 CSS 类和样式	244
12.4.1	使用类绑定	244
12.4.2	使用样式绑定	248
12.5	更新应用程序的数据	252
12.6	本章小结	254
第 13 章	使用内置指令	255
13.1	准备示例项目	256
13.2	使用内置指令	257
13.2.1	使用 ngIf 指令	258
13.2.2	使用 ngSwitch 指令	260
13.2.3	使用 ngFor 指令	262
13.2.4	使用 ngTemplateOutlet	271
13.3	理解单向数据绑定的限制	273
13.3.1	使用幂等表达式	274
13.3.2	理解表达式上下文	276
13.4	本章小结	279
第 14 章	使用事件和表单	281
14.1	准备示例项目	282
14.1.1	添加表单模块	282
14.1.2	准备组件和模板	283
14.2	使用事件绑定	285
14.2.1	理解动态定义的属性	286
14.2.2	使用事件数据	288
14.2.3	使用模板引用变量	290
14.3	使用双向数据绑定	292
14.4	处理表单	295
14.4.1	向示例应用程序添加	295
	表单	295
14.4.2	添加表单数据验证	297
14.4.3	验证整个表单	307
14.5	使用基于模型的表单	313
14.5.1	启用基于模型的表单	313
	功能	313
14.5.2	定义表单模型类	314
14.5.3	使用模型进行验证	317
14.5.4	根据模型生成元素	321
14.6	创建自定义表单验证器	322
14.7	本章小结	325
第 15 章	创建属性指令	327
15.1	准备示例项目	328
15.2	创建简单的属性指令	330
15.3	在指令中访问应用程序	333
	数据	333
15.3.1	读取宿主元素属性	333
15.3.2	创建数据绑定输入	335
	属性	335
15.3.3	响应输入属性变化	338
15.4	创建自定义事件	340
15.5	创建宿主元素绑定	343
15.6	在宿主元素上创建双向	344
	绑定	344
15.7	导出指令用于模板变量	348

15.8 本章小结 350

第 16 章 创建结构型指令 351

16.1 准备示例项目 352

16.2 创建简单的结构型指令 353

16.2.1 实现结构型指令类 354

16.2.2 启用结构型指令 356

16.2.3 使用结构型指令的
简洁语法 358

16.3 创建迭代结构型指令 359

16.3.1 提供额外的上下文
数据 362

16.3.2 使用简洁的结构语法 363

16.3.3 处理属性级数据变更 364

16.3.4 处理集合级数据变更 365

16.4 查询宿主元素内容 376

16.4.1 查询多个子内容 379

16.4.2 接收查询变更通知 381

16.5 本章小结 382

第 17 章 理解组件 385

17.1 准备示例项目 386

17.2 使用组件来组织应用程序 386

17.2.1 创建新组件 388

17.2.2 定义模板 391

17.2.3 完成组件的重组 400

17.3 使用组件样式 401

17.3.1 定义外部组件样式 402

17.3.2 使用高级样式特性 403

17.4 查询模板内容 410

17.5 本章小结 412

第 18 章 使用和创建管道 413

18.1 准备示例项目 414

18.2 理解管道 418

18.3 创建一个自定义管道 419

18.3.1 注册自定义管道 421

18.3.2 应用自定义管道 421

18.3.3 组合管道 423

18.3.4 创建非纯管道 423

18.4 使用内置管道 427

18.4.1 格式化数值 428

18.4.2 格式化货币值 431

18.4.3 格式化百分比 433

18.4.4 格式化日期 435

18.4.5 改变字符串大小写 438

18.4.6 将数据序列化为 JSON
数据 439

18.4.7 将数据数组切片 440

18.5 小结 442

第 19 章 使用服务 443

19.1 准备示例项目 444

19.2 理解对象分发问题 445

19.2.1 问题的提出 445

19.2.2 利用依赖注入将对象
作为服务分发 450

19.2.3 在其他构造块中声明
依赖 455

19.3 理解测试隔离问题 461

19.4 完成服务的融入 465

19.4.1 更新根组件和模板 465

19.4.2 更新子组件 466

19.5 小结 468

第 20 章 使用服务提供程序 469

20.1 准备示例项目 470

20.2 使用服务提供程序 471

20.2.1 使用类提供程序 474

20.2.2 使用值提供程序 481

20.2.3 使用工厂提供程序 483

20.2.4 使用已有服务提供
程序 486

20.3 使用本地提供程序 487

20.3.1 理解单个服务对象的
限制 488

20.3.2 在一条指令中创建本地
提供程序 489

20.3.3	在组件中创建本地提供程序	491	22.6	完成项目	535
20.3.4	控制依赖解析	495	22.6.1	创建Angular引导程序	536
20.4	小结	498	22.6.2	创建 HTML 文档	536
第 21 章	使用和创建模块	499	22.7	运行示例项目	537
21.1	准备示例项目	499	22.8	小结	538
21.2	理解根模块	501	第 23 章	使用Reactive Extensions	539
21.2.1	理解 imports 属性	504	23.1	准备示例项目	540
21.2.2	理解 declarations 属性	504	23.2	理解问题	540
21.2.3	理解 providers 属性	504	23.3	使用 Reactive Extensions 解决问题	543
21.2.4	理解 bootstrap 属性	504	23.3.1	理解 Observable	544
21.3	创建功能模块	507	23.3.2	理解 Observer	545
21.3.1	创建模型模块	508	23.3.3	理解 Subject	547
21.3.2	创建实用工具功能模块	512	23.4	使用 async 管道	548
21.3.3	用组件创建一个功能模块	518	23.5	扩展应用程序功能模块	551
21.4	小结	521	23.6	基础之上更进一步	553
第 22 章	创建示例项目	523	23.6.1	过滤事件	554
22.1	启动示例项目	523	23.6.2	转换事件	555
22.2	添加和配置包	524	23.6.3	只接收不同的事件	558
22.2.1	配置 TypeScript	525	23.6.4	获取和忽略事件	560
22.2.2	配置 HTTP 开发服务器	525	23.7	小结	561
22.3	创建模型模块	526	第 24 章	生成异步 HTTP 请求	563
22.3.1	创建产品数据类型	526	24.1	准备示例项目	564
22.3.2	创建数据源和存储库	526	24.1.1	配置模型功能模块	565
22.3.3	完成模型模块	528	24.1.2	更新表单组件	566
22.4	创建核心模块	528	24.1.3	运行示例项目	567
22.4.1	创建共享状态服务	528	24.2	理解 RESTful Web 服务	568
22.4.2	创建表格组件	529	24.3	替换静态数据源	569
22.4.3	创建表单组件	531	24.3.1	创建新的数据源服务	569
22.4.4	完成核心模块	533	24.3.2	配置数据源	572
22.5	创建消息模块	533	24.3.3	使用 REST 数据源	572
22.5.1	创建消息模型和服务	533	24.3.4	保存和删除数据	574
22.5.2	创建组件和模板	534	24.4	加强 HTTP 请求	576
22.5.3	完成消息模块	535	24.5	生成跨域请求	578
			24.6	配置请求头	581
			24.7	处理错误	584

24.7.1	生成用户可以使用的消息	585
24.7.2	处理错误	586
24.8	小结	588
第 25 章	路由和导航：第 1 部分	589
25.1	准备示例项目	590
25.2	开始学习路由	593
25.2.1	创建路由配置	594
25.2.2	创建路由组件	595
25.2.3	更新根模块	596
25.2.4	完成配置	596
25.2.5	添加导航链接	597
25.2.6	理解路由的效果	600
25.3	完成路由实现	602
25.3.1	在组件中处理路由变化	602
25.3.2	使用路由参数	605
25.3.3	在代码中导航	611
25.3.4	接收导航事件	614
25.3.5	删除事件绑定和支持代码	615
25.4	小结	618
第 26 章	路由与导航：第 2 部分	619
26.1	准备示例项目	619
26.2	使用通配符和重定向	626
26.2.1	在路由中使用通配符	626
26.2.2	在路由中使用重定向	628
26.3	在组件内部导航	630
26.3.1	响应正在发生的路由变化	631
26.3.2	为活动路由设置不同样式的链接	633
26.3.3	修复 All 按钮	636
26.4	创建子路由	637
26.4.1	创建子路由由出口	638
26.4.2	从子路由访问参数	640
26.5	小结	644

第 27 章	路由与导航：第 3 部分	645
27.1	准备示例项目	645
27.2	守卫路由	647
27.2.1	使用解析器推迟导航	647
27.2.2	避免带有守卫的导航	654
27.3	动态加载功能模块	667
27.3.1	创建一个简单的功能模块	667
27.3.2	动态加载模块	669
27.3.3	守卫动态模块	672
27.4	指定命名出口	675
27.4.1	创建附加的出口元素	676
27.4.2	在使用多个出口的情况下导航	678
27.5	小结	680
第 28 章	使用动画	681
28.1	准备示例项目	682
28.1.1	添加动画 polyfill	682
28.1.2	禁用 HTTP 延迟	685
28.1.3	简化表格模板和路由配置	686
28.2	开始学习 Angular 动画	688
28.2.1	创建动画	688
28.2.2	应用动画	691
28.2.3	测试动画效果	694
28.3	理解内置的动画状态	696
28.4	理解元素过渡	697
28.4.1	为内置状态创建迁移	697
28.4.2	控制动画过渡	699
28.5	理解动画样式分组	704
28.5.1	在可重用分组中定义公共样式	704
28.5.2	使用元素变形	705
28.5.3	应用 CSS 框架样式	707
28.6	理解动画触发器事件	709
28.7	小结	712

第 29 章 Angular 单元测试 713

29.1 准备示例项目 714

29.1.1 添加测试包 715

29.1.2 创建一个简单的单元测试 719

29.1.3 启动工具 719

29.2 使用 Jasmine 完成单元测试 721

29.3 测试 Angular 组件 722

29.3.1 使用 TestBed 类完成工作 722

29.3.2 测试数据绑定 726

29.3.3 测试带有外部模板的组件 728

29.3.4 测试组件事件 730

29.3.5 测试输出属性 732

29.3.6 测试输入属性 734

29.3.7 测试异步操作 736

29.4 测试 Angular 指令 739

29.5 小结 740

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 HTML，为更加简单方便地构建富应用程序(rich application)打下了良好的基础。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，但是能够实现的功能并未受到这些固有功能的限制，从而能够创建可兼容标准的 Web 应用程序，并且能够利用最新功能(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。它已被用于一些最大和最复杂的 Web 应用程序。本书将向你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，你应该熟悉 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 至第 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 的书不可能为涵盖 HTML 的方方面面提供篇幅。如果需要 HTML 和 CSS 的完整参考，那么推荐阅读我的另一本书 *The Definitive Guide to HTML5*，它也是由 Apress 出版社出版的。

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 1 部分：准备

本书第 1 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)。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 2 部分：使用 Angular

本书的第 2 部分将介绍 Angular 为构建应用程序所提供的构造块，依次描述所有这些构造块。Angular 包含了很多内置的功能，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 3 部分：Angular 的高级功能

本书第 3 部分介绍如何使用高级功能来创建更复杂和可扩展的应用程序。这部分介绍 Reactive Extensions 软件包，Angular 提供的大量功能均构建在这个软件包的基础之上，并且有多种 Angular 高级功能直接使用该软件包。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 路由在应用程序中进行导航，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，本书已经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，就像在代码清单 1-1 中所示的那样。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 { }

```

这个代码清单摘自第 15 章。这里暂不关注它的具体作用，只需要注意，在每章中第一次用到的文件将完整列出，类似于代码清单 1-1 所示。对于第二次及以后出现的其他示例，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]="getProducts().length < 6 ? 'bg-success' : 'bg-warning'">
    <td>{{i + 1}}</td>
    <td>{{item.name}}</td>
    <td [pa-attr]="item.category == 'Soccer' ? 'bg-info' : null">
      {{item.category}}
    </td>
    <td [pa-attr]="'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。在这样的部分代码清单中，仅给出本章前面完整代码清单中发生了更改的那部分代码。在某些情况下，需要对同一个文件的不同部分进行更改，此时为了简洁起见，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;
}

```