

Web 开发经典丛书

Angular 高级编程

(第 4 版)

[英] 亚当·弗里曼(Adam Freeman) 著
陈磊 译

清华大学出版社

北 京

北京市版权局著作权合同登记号 图字：01-2020-5562
Pro Angular 9: Build Powerful and Dynamic Web Apps
Adam Freeman
EISBN: 978-1-4842-5997-9

Original English language edition published by Adam Freeman. Copyright © 2020 by Adam Freeman.
Simplified Chinese-Language edition copyright © 2021 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。

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。举报：010-62782989, beiqinquan@tup.tsinghua.edu.cn

图书在版编目(CIP)数据

Angular高级编程：第4版 / (英) 亚当·弗里曼(Adam Freeman) 著；陈磊译. —北京：清华大学出版社，2021.1

(Web 开发经典丛书)

书名原文：Pro Angular 9: Build Powerful and Dynamic Web Apps

ISBN 978-7-302-56957-2

I. ①A… II. ①亚… ②陈… 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(2020)第 228228 号

责任编辑：王 军
装帧设计：孔祥峰
责任校对：成凤进
责任印制：沈 露

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：三河市龙大印装有限公司

经 销：全国新华书店

开 本：170mm×240mm 印 张：43 字 数：1185 千字

版 次：2021 年 1 月第 1 版 印 次：2021 年 1 月第 1 次印刷

定 价：158.00 元

产品编号：090157-01

译者序

Web 应用程序的功能越来越丰富、用户体验越来越好，随之而来的，是前端开发人员变得越来越受欢迎。

经验丰富的前端开发人员知道，合适的前端框架能够帮助自己提高工作效率，编写出高性能且容易使用的 Web 应用程序。在前端开发的世界中，Angular 是一个非常出色的框架，目前由 Google 提供支持。

关于 Angular，有一个小故事。Angular 的前身是 AngularJS，而 AngularJS 的前身称作 GetAngular，那是一个由 Miško Hevery 和 Adam Abrons 在工作之余开发的项目。在项目开发期间，Hevery 已经就职于 Google，并负责 Google Feedback 项目。经历了 6 个月的开发后，该项目的代码超过 17 000 行，这让 Hevery 感到非常沮丧。他想到了 GetAngular，认为这个框架能够为 Google Feedback 项目提供很大的帮助。于是，他找到自己的经理，请求用 AngularJS 重写项目，并打赌说自己能够在两周内重写整个项目。尽管他输掉了赌局(因为他最终用了 3 周而不是 2 周)，但项目的代码量从近两万行锐减到 1500 行。也正因如此，Google 才对 GetAngular 产生了浓厚的兴趣，并最终将其纳入自己的产品体系，更名为 AngularJS。

从这个小故事可以看出，Angular 能够显著降低 Web 应用程序的代码量，进而提高开发人员的生产效率、加快项目的开发进度。这种优势源于 Angular 提供的多种功能，例如依赖注入、双向数据绑定、采用指令等。而且，Angular 是使用 TypeScript 构建的，这是 JavaScript 的超集，提供了对强类型的支持，使得前端开发工作变得更加简单。不仅如此，Angular 的更新相当迅速，每个新的主版本都带来不小的改进，使这个框架依然保持着旺盛的生命力。

综上所述，对于有志于转向前端开发人员，Angular 是一个上佳的选择，而本书则可作为学习 Angular 的指路明灯，帮助读者从基础功能到高级功能，了解 Angular 开发的方方面面。

本书分为三部分。第 I 部分讲解 Angular 开发的基础知识，对 HTML、CSS、JavaScript 和 TypeScript 也进行了介绍，确保对前端开发一无所知的读者，也能够快速上手。本部分最后还开发了一个简单的 Web 应用程序，帮助读者巩固基础知识。第 II 部分详细讲解 Angular，内容涵盖项目结构、开发工具、数据绑定、各种指令、事件和表单、管道、服务、服务提供程序、模块等。这部分是大全式讲解，对 Angular 的主要功能都进行了介绍。第 III 部分更偏向较高级的功能，介绍了 Reactive Extensions、HTTP 请求、路由、动画和单元测试。

本书内容全面详细，并且通过大量代码示例和屏幕截图来演示概念和功能，理解起来比较轻松。相信读者在阅读完本书后，一定会有丰厚的收获。希望这本书能够为读者成为优秀的前端开发人员助力！

作者简介



Adam Freeman 是一位经验丰富的 IT 专业人士，在多家公司担任过高级职位，还曾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	4
1.2.3 第 III 部分: Angular 高级功能	4
1.3 本书不介绍什么	4
1.4 Angular 开发需要什么软件	4
1.5 如何设置开发环境	4
1.6 学习示例时遇到问题怎么办	4
1.7 如果在书中发现错误怎么办	5
1.8 大量示例	5
1.9 获取示例代码	6
1.10 联系作者	6
1.11 如果你真的很喜欢这本书	7
1.12 如果你不喜欢这本书	7
1.13 本章小结	7
第 2 章 第一个 Angular 应用程序	9
2.1 准备开发环境	9
2.1.1 安装 Node.js	9
2.1.2 安装 angular-cli 包	10
2.1.3 安装编辑器	11
2.1.4 安装浏览器	11
2.2 创建并准备项目	11
2.2.1 创建项目	11
2.2.2 启动开发工具	12
2.3 向项目中添加 Angular 功能	13
2.3.1 创建数据模型	14
2.3.2 向用户显示数据	16
2.3.3 更新组件	17
2.3.4 设置 HTML 元素的样式	20
2.3.5 显示待办事项列表	21
2.3.6 创建双向数据绑定	24
2.3.7 添加待办事项	26
2.3.8 显示完成的待办事项	29
2.4 本章小结	31
第 3 章 将 Angular 放在上下文中	33
3.1 理解 Angular 的强项	33
3.2 比较 Angular、React 和 Vue.js	34
3.3 理解 MVC 模式	35
3.3.1 理解模型	36
3.3.2 理解控制器/组件	37
3.3.3 理解视图/模板	38
3.4 理解 RESTful 服务	38
3.5 常见的设计缺陷	39
3.5.1 将逻辑放错地方	40
3.5.2 数据存储采用的数据格式	40
3.5.3 足够的知识足以制造麻烦	40
3.6 本章小结	40
第 4 章 HTML 和 CSS 入门	43
4.1 准备示例项目	43
4.2 理解 HTML	45
4.2.1 理解空元素	45
4.2.2 理解属性	45
4.2.3 应用无值属性	46
4.2.4 在属性中引用字面量	46
4.2.5 理解元素内容	46
4.2.6 理解文档结构	47
4.3 理解 Bootstrap	48
4.3.1 应用基本的 Bootstrap 类	48
4.3.2 使用 Bootstrap 样式化表格	51

4.3.3	使用 Bootstrap 创建表单	52
4.3.4	使用 Bootstrap 创建网格	53
4.4	本章小结	58

第 5 章 JavaScript 与 TypeScript:

第 1 部分

5.1	准备示例项目	60
5.2	使用语句	61
5.3	定义和使用函数	62
5.3.1	定义带形参的函数	63
5.3.2	定义返回结果的函数	64
5.3.3	将函数用作其他函数的 实参	64
5.4	使用变量和类型	65
5.4.1	使用变量闭包	67
5.4.2	使用基本数据类型	67
5.5	使用 JavaScript 操作符	69
5.5.1	使用条件语句	69
5.5.2	相等操作符和恒等操作符	70
5.5.3	显式类型转换	71
5.6	处理数组	72
5.6.1	使用数组字面量	73
5.6.2	数组内容的读取和修改	73
5.6.3	遍历数组内容	73
5.6.4	spread 操作符	74
5.6.5	使用内置数组方法	75
5.7	本章小结	76

第 6 章 JavaScript 与 TypeScript:

第 2 部分

6.1	准备示例项目	77
6.2	使用对象	78
6.2.1	使用对象字面量	78
6.2.2	将函数用作方法	79
6.2.3	定义类	79
6.3	处理 JavaScript 模块	82
6.4	有用的 TypeScript 特性	86
6.4.1	使用类型注解	86
6.4.2	使用元组	90
6.4.3	使用可索引类型	91
6.4.4	使用访问修饰符	91

6.5	本章小结	92
-----	------	----

第 7 章 SportsStore: 一个真实的应用程序

7.1	准备项目	93
7.1.1	安装额外的 NPM 软件包	94
7.1.2	准备 RESTful Web 服务	95
7.1.3	准备 HTML 文件	97
7.1.4	创建文件夹结构	97
7.1.5	运行示例应用程序	98
7.1.6	启动 RESTful Web 服务	98
7.2	准备 Angular 项目功能	98
7.2.1	更新根组件	98
7.2.2	更新根模块	99
7.2.3	检查引导文件	100
7.3	启动数据模型	100
7.3.1	创建模型类	100
7.3.2	创建虚拟数据源	101
7.3.3	创建模型存储库	102
7.3.4	创建功能模块	103
7.4	启动商店	103
7.4.1	创建 Store 组件和模板	104
7.4.2	创建商店功能模块	105
7.4.3	更新根组件和根模块	106
7.5	添加商店功能: 产品详情	107
7.5.1	显示产品详情	107
7.5.2	添加类别选择	108
7.5.3	添加产品分页功能	110
7.5.4	创建自定义指令	112
7.6	本章小结	116

第 8 章 SportsStore: 订单和结账

8.1	准备示例应用程序	117
8.2	创建购物车	117
8.2.1	创建购物车模型	117
8.2.2	创建购物车概览组件	119
8.2.3	将购物车集成到商店中	120
8.3	添加 URL 路由	123
8.3.1	创建购物车详情和结账 组件	124
8.3.2	创建和应用路由配置	124
8.3.3	应用程序导航	126

8.3.4 路由守卫	128	10.5.2 准备应用程序	174
8.4 完成购物车详情功能	130	10.5.3 创建 Docker 容器	174
8.5 处理订单	133	10.5.4 运行应用程序	175
8.5.1 扩展模型	133	10.6 本章小结	177
8.5.2 收集订单详情	135		
8.6 使用 RESTful Web 服务	138		
8.7 本章小结	140		
第 9 章 SportsStore: 管理	141	第 II 部分 Angular 详解	
9.1 准备示例应用程序	141	第 11 章 理解 Angular 项目和工具 ...	181
9.1.1 创建模块	141	11.1 创建新的 Angular 项目	181
9.1.2 配置 URL 路由系统	144	11.2 了解项目结构	182
9.1.3 导航到管理 URL	145	11.2.1 了解 src 文件夹	183
9.2 实现身份验证	146	11.2.2 了解包文件夹	184
9.2.1 理解身份验证系统	146	11.3 使用开发工具	187
9.2.2 扩展数据源	147	11.3.1 了解开发 HTTP	
9.2.3 创建身份验证服务	148	服务器	187
9.2.4 启用身份验证	149	11.3.2 了解构建过程	188
9.3 扩展数据源和存储库	151	11.3.3 使用 linter	192
9.4 创建管理功能结构	155	11.4 理解 Angular 应用程序是如何	
9.4.1 创建占位符组件	155	工作的	194
9.4.2 准备常用内容和功能		11.4.1 理解 HTML 文档	194
模块	156	11.4.2 理解应用程序引导	195
9.4.3 实现产品功能	158	11.4.3 理解 Angular 根模块	196
9.4.4 实现订单功能	162	11.4.4 理解 Angular 组件	197
9.5 本章小结	164	11.4.5 理解内容显示	197
第 10 章 SportsStore: 渐进式特性和		11.5 了解生产构建过程	199
部署	165	11.5.1 了解预编译	199
10.1 准备示例应用程序	165	11.5.2 了解差异加载	199
10.2 添加渐进式特性	165	11.5.3 运行生产构建	201
10.2.1 安装 PWA 包	165	11.6 在 Angular 项目中开始开发	201
10.2.2 缓存数据 URL	166	11.6.1 创建数据模型	201
10.2.3 响应对连接的更改	167	11.6.2 创建组件和模板	205
10.3 为部署准备应用程序	169	11.6.3 配置根 Angular 模块	206
10.3.1 创建数据文件	169	11.7 本章小结	207
10.3.2 创建服务器	169	第 12 章 使用数据绑定	209
10.3.3 更改存储库类中的		12.1 准备示例项目	210
Web 服务 URL	171	12.2 理解单向数据绑定	210
10.4 构建和测试应用程序	172	12.2.1 理解绑定目标	212
10.5 将 SportsStore 应用程序容器化	174	12.2.2 理解表达式	213
10.5.1 安装 Docker	174	12.2.3 理解括号	214
		12.2.4 理解宿主元素	214
		12.3 使用标准属性和属性绑定	215

12.3.1	使用标准属性绑定	215	14.7	本章小结	292
12.3.2	使用字符串插入绑定	216	第 15 章	创建属性指令	293
12.3.3	使用元素属性绑定	217	15.1	准备示例项目	294
12.4	设置 CSS 类和样式	218	15.2	创建简单的属性指令	296
12.4.1	使用类绑定	218	15.3	在指令中访问应用程序数据	298
12.4.2	使用样式绑定	223	15.3.1	读取宿主元素属性	298
12.5	更新应用程序的数据	226	15.3.2	创建数据绑定输入属性	300
12.6	本章小结	228	15.3.3	响应输入属性的变化	303
第 13 章	使用内置指令	229	15.4	创建自定义事件	305
13.1	准备示例项目	230	15.5	创建宿主元素绑定	308
13.2	使用内置指令	231	15.6	在宿主元素上创建双向绑定	309
13.2.1	使用 ngIf 指令	232	15.7	导出指令用于模板变量	312
13.2.2	使用 ngSwitch 指令	234	15.8	本章小结	315
13.2.3	使用 ngFor 指令	236	第 16 章	创建结构型指令	317
13.2.4	使用 ngTemplateOutlet 指令	244	16.1	准备示例项目	318
13.3	理解单向数据绑定的限制	246	16.2	创建简单的结构型指令	319
13.3.1	使用幂等表达式	246	16.2.1	实现结构型指令类	320
13.3.2	理解表达式上下文	249	16.2.2	启用结构型指令	322
13.4	本章小结	251	16.2.3	使用结构型指令的简洁语法	323
第 14 章	使用事件和表单	253	16.3	创建迭代结构型指令	324
14.1	准备示例项目	254	16.3.1	提供额外的上下文数据	327
14.1.1	导入表单模块	254	16.3.2	使用简洁的结构语法	329
14.1.2	准备组件和模板	254	16.3.3	处理属性级数据变更	330
14.2	使用事件绑定	255	16.3.4	处理集合级数据变更	331
14.2.1	使用事件数据	259	16.4	查询宿主元素内容	340
14.2.2	使用模板引用变量	260	16.4.1	查询多个子内容	344
14.3	使用双向数据绑定	262	16.4.2	接收查询变更通知	346
14.4	处理表单	265	16.5	本章小结	347
14.4.1	向示例应用程序添加表单	265	第 17 章	理解组件	349
14.4.2	添加表单数据验证	267	17.1	准备示例项目	350
14.4.3	验证整个表单	275	17.2	使用组件组织应用程序	350
14.5	使用基于模型的表单	281	17.2.1	创建新组件	351
14.5.1	启用基于模型的表单功能	282	17.2.2	定义模板	354
14.5.2	定义表单模型类	282	17.2.3	完成组件的重组	363
14.5.3	使用模型进行验证	285	17.3	使用组件样式	364
14.5.4	根据模型生成元素	288	17.3.1	定义外部组件样式	365
14.6	创建自定义表单验证器	289			

17.3.2	使用高级样式特性	366
17.4	查询模板内容	372
17.5	本章小结	374
第 18 章	使用和创建管道	375
18.1	准备示例项目	376
18.2	理解管道	378
18.3	创建一个自定义管道	380
18.3.1	注册自定义管道	381
18.3.2	应用自定义管道	382
18.3.3	组合管道	383
18.3.4	创建非纯管道	384
18.4	使用内置管道	387
18.4.1	格式化数值	388
18.4.2	格式化货币值	390
18.4.3	格式化百分比	393
18.4.4	格式化日期	394
18.4.5	改变字符串大小写	397
18.4.6	将数据序列化为 JSON 数据	399
18.4.7	将数据数组切片	400
18.4.8	格式化键/值对	401
18.4.9	选择值	402
18.4.10	复数化值	404
18.5	本章小结	405
第 19 章	使用服务	407
19.1	准备示例项目	408
19.2	理解对象分发问题	409
19.2.1	问题的提出	409
19.2.2	利用依赖注入将对象作为服务分发	414
19.2.3	在其他构造块中声明依赖	419
19.3	理解测试隔离问题	425
19.4	完成服务的融入	428
19.4.1	更新根组件和模板	429
19.4.2	更新子组件	430
19.5	本章小结	431
第 20 章	使用服务提供程序	433
20.1	准备示例项目	434
20.2	使用服务提供程序	435
20.2.1	使用类提供程序	438
20.2.2	使用值提供程序	445
20.2.3	使用工厂提供程序	447
20.2.4	使用已有的服务提供程序	449
20.3	使用本地提供程序	450
20.3.1	理解单个服务对象的局限性	450
20.3.2	在组件中创建本地提供程序	452
20.3.3	理解服务提供程序的替代方案	453
20.3.4	控制依赖解析	458
20.4	本章小结	459
第 21 章	使用和创建模块	461
21.1	准备示例项目	461
21.2	理解根模块	463
21.2.1	理解 imports 属性	465
21.2.2	理解 declarations 属性	465
21.2.3	理解 providers 属性	466
21.2.4	理解 bootstrap 属性	466
21.3	创建功能模块	468
21.3.1	创建模型模块	469
21.3.2	创建实用工具功能模块	473
21.3.3	使用组件创建一个功能模块	478
21.4	本章小结	482
第 III 部分 Angular 高级功能		
第 22 章	创建示例项目	485
22.1	启动示例项目	485
22.1.1	添加和配置 Bootstrap CSS 包	485
22.1.2	创建项目结构	486
22.2	创建模型模块	486
22.2.1	创建产品数据类型	486
22.2.2	创建数据源和存储库	486

22.2.3	完成模型模块	488	24.4	加强 HTTP 请求	531
22.3	创建核心模块	488	24.5	生成跨域请求	533
22.3.1	创建共享状态服务	488	24.6	配置请求头	535
22.3.2	创建表格组件	489	24.7	处理错误	537
22.3.3	创建表单组件	490	24.7.1	生成用户可使用的消息	538
22.3.4	完成核心模块	492	24.7.2	处理错误	540
22.4	创建消息模块	493	24.8	本章小结	541
22.4.1	创建消息模型和服务	493	第 25 章 路由与导航：第 1 部分	543	
22.4.2	创建组件和模板	493	25.1	准备示例项目	544
22.4.3	完成消息模块	494	25.2	开始学习路由	546
22.5	完成项目	495	25.2.1	创建路由配置	546
22.6	本章小结	496	25.2.2	创建路由组件	548
第 23 章 使用 Reactive Extensions	497		25.2.3	更新根模块	548
23.1	准备示例项目	498	25.2.4	完成配置	549
23.2	理解问题	498	25.2.5	添加导航链接	550
23.3	使用 Reactive Extensions 解决问题	501	25.2.6	理解路由的效果	552
23.3.1	理解 Observable	502	25.3	完成路由实现	554
23.3.2	理解 Observer	503	25.3.1	在组件中处理路由变化	554
23.3.3	理解 Subject	505	25.3.2	使用路由参数	556
23.4	使用 async 管道	506	25.3.3	在代码中导航	562
23.5	扩展应用程序功能模块	509	25.3.4	接收导航事件	564
23.6	更进一步	511	25.3.5	删除事件绑定和支持代码	566
23.6.1	过滤事件	511	25.4	本章小结	568
23.6.2	转换事件	513	第 26 章 路由与导航：第 2 部分	569	
23.6.3	只接收不同的事件	515	26.1	准备示例项目	569
23.6.4	获取和忽略事件	517	26.2	使用通配符和重定向	575
23.7	本章小结	519	26.2.1	在路由中使用通配符	575
第 24 章 生成 HTTP 请求	521		26.2.2	在路由中使用重定向	577
24.1	准备示例项目	522	26.3	在组件内部导航	578
24.1.1	配置模型功能模块	522	26.3.1	响应正在发生的路由变化	579
24.1.2	创建数据文件	523	26.3.2	为活动路由设置不同样式的链接	581
24.1.3	更新表单组件	523	26.3.3	修复 All 按钮	584
24.1.4	运行示例项目	524	26.4	创建子路由	585
24.2	理解 RESTful Web 服务	524	26.4.1	创建子路由出口	586
24.3	替换静态数据源	525	26.4.2	从子路由访问参数	588
24.3.1	创建新的数据源服务	525			
24.3.2	配置数据源	527			
24.3.3	使用 REST 数据源	527			
24.3.4	保存和删除数据	529			

26.5	本章小结	592	28.4	理解元素过渡	638
第 27 章	路由与导航：第 3 部分	593	28.4.1	为内置状态创建过渡	638
27.1	准备示例项目	593	28.4.2	控制动画的过渡	639
27.2	守卫路由	595	28.5	理解动画样式组	644
27.2.1	使用解析器推迟导航	595	28.5.1	在可重用的分组中定义公共样式	644
27.2.2	避免带有守卫的导航	601	28.5.2	使用元素变形	646
27.3	动态加载功能模块	613	28.5.3	应用 CSS 框架样式	647
27.3.1	创建一个简单的功能模块	613	28.6	本章小结	649
27.3.2	动态加载模块	614	第 29 章	Angular 单元测试	651
27.3.3	守卫动态模块	618	29.1	准备示例项目	652
27.4	指定命名出口	620	29.2	创建一个简单的单元测试	654
27.4.1	创建附加的出口元素	621	29.3	使用 Jasmine 完成单元测试	655
27.4.2	在使用多个出口的情况下导航	623	29.4	测试 Angular 组件	656
27.5	本章小结	624	29.4.1	使用 TestBed 类完成工作	656
第 28 章	使用动画	625	29.4.2	测试数据绑定	660
28.1	准备示例项目	626	29.4.3	测试带有外部模板的组件	662
28.1.1	禁用 HTTP 延迟	626	29.4.4	测试组件事件	663
28.1.2	简化表格模板和路由配置	627	29.4.5	测试输出属性	665
28.2	开始学习 Angular 动画	629	29.4.6	测试输入属性	667
28.2.1	启用动画模块	629	29.4.7	测试异步操作	669
28.2.2	创建动画	630	29.5	测试 Angular 指令	671
28.2.3	应用动画	633	29.6	本章小结	673
28.2.4	测试动画效果	635			
28.3	理解内置的动画状态	637			

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: 渐进式特性和部署

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 应用程序遵守一种清晰的设计模式，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，即使是复杂的 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 HTML 功能(如 HTML5)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，可在第 5、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

本书分为三部分，每部分介绍一组相关的主题。

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础知识，包括 HTML、CSS 和 TypeScript(JavaScript 的一个超集，用于 Angular 开发)。该

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，而该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。该部分演示如何在 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 本书不介绍什么

本书面向经验丰富，但新接触 Angular 的 Web 开发人员，因此不介绍 Web 应用程序或编程的基础知识，不过本书用几章内容介绍了 HTML、CSS 和 JavaScript 的基础。本书不详细介绍服务器端开发，如果想创建后台服务来支持 Angular 应用程序，则可以参考我撰写的其他图书。

另外，尽管我希望详尽介绍 Angular，但主流开发并不会用到 Angular 提供的每种功能，而且我也要避免让图书太厚。如果没有介绍某种功能，肯定是因为我认为它不够重要，或者已经介绍的某种技术能够实现与其相同的结果。

1.4 Angular 开发需要什么软件

你需要使用代码编辑器和第 2 章介绍的工具。Angular 开发用到的工具可以免费获得，并且可在 Windows、macOS 和 Linux 上使用。

1.5 如何设置开发环境

第 2 章通过创建一个简单的应用程序来介绍 Angular，在此过程中讲解了如何为 Angular 开发设置开发环境。

1.6 学习示例时遇到问题怎么办

首先应该做的是返回到章节开头并重新操作。导致大部分问题的原因是忘记执行某个步骤，或者没有完整遵守代码清单。应该特别留意代码清单中重点强调的地方，因为它们突出说明了需要做出的修改。

之后，检查本书 GitHub 存储库中的勘误表/修订列表。技术图书非常复杂，尽管我和编辑们尽力避免出现错误，但错误是难免的。遇到问题时，可以查看勘误表中的错误列表及修复问题的说明。

如果问题依然存在，则从本书的 GitHub 存储库(<https://github.com/Apress/pro-angular-9>)中下载

对应章节的项目，并将其与自己的项目进行对比。我为每一章创建了代码并放到了 GitHub 存储库中，你的项目中应该有相同的文件，文件中有相同的内容。

1.7 如果在书中发现错误怎么办

你可以通过电子邮件(adam@adam-freeman.com)把错误报告给我，不过建议你在发邮件之前，先检查本书的勘误表/修订列表，因为其他人可能已经报告了相同的错误。在本书的 GitHub 存储库中可找到本书的勘误表/修订列表：<https://github.com/Apress/pro-angular-9>。

我把可能造成读者混淆的错误，尤其是涉及示例代码的问题，添加到 GitHub 存储库中的勘误表/修订列表中，并心怀感激地列出了第一个报告错误的读者。对于不那么严重的错误，通常是示例上下文中的文字错误，我也记录下来，准备在撰写新版本的时候加以订正。

1.8 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 { AppComponent } from "../app.component";
import { StoreModule } from "../store/store.module";

@NgModule({
  imports: [BrowserModule, StoreModule],
  declarations: [AppComponent],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

代码清单 1-1 摘自第 7 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>
...

```

代码段代码清单 1-2 也是摘自后面的一章。可以看到，这里只显示了 `body` 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，关注示例中发生修改的部分，或者说明所描述的功能或技术的部分。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;
}

```

代码段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，描述该章节包含的技术以及演示如何使用它们的代码清单。

1.9 获取示例代码

可以扫描封底二维码下载本书的示例项目。

1.10 联系作者

可以向 adam@adam-freeman.com 发送电子邮件。从我第一次在图书中公开我的电子邮件地址到现在，已经过去几年时间了。当时，我并不完全确定这不是不是一个好主意，不过现在我很高兴自己那么做了。我收到了来自全世界各个行业的读者的电子邮件，并且这些邮件大部分很正面、很有礼貌，读起来让人很愉快。

我努力迅速回复收到的邮件，但是我收到的邮件很多，所以有时候(尤其是我努力完成一本书的时候)会积压一些。我总是尽力帮助在完成书中示例时遇到困难的患者，不过请在联系我之前，先完成本章前面介绍的步骤。

虽然我欢迎读者给我发邮件，但对于某些经常遇到的问题，我的回答始终是否定的。很抱歉，

我不会为你的新公司编写代码，不会帮助你完成大学课堂的作业，不会参与你的开发团队关于产品设计的争论，也不会教你如何编程。

1.11 如果你真的很喜欢这本书

请向 adam@adam-freeman.com 发送电子邮件，告诉我这一点。知道读者喜欢我的图书，总是让我感觉很愉快，我感谢你花时间给我发送那些邮件。撰写图书的过程并不轻松，这种邮件让我有动力来坚持完成一个有时候看起来难以完成的任务。

1.12 如果你不喜欢这本书

你也可以向 adam@adam-freeman.com 发送电子邮件，我仍然会尽力帮助你。请记住，只有你向我解释了问题是什么，以及你希望我为你提供怎样的帮助，我才能够帮助你。你应该理解，有时候，你唯一能够做的是认识到，对于你来说，我不是一个合适的作者，只有你放下这本书，改为选择另外一本书，我们之间的关系才能结束。我会认真思索让你感到沮丧的地方，不过在撰写了 25 年图书后，我已经认识到，并不是每个人都喜欢阅读我撰写的图书。

1.13 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。