

Pro
ASP.NET
MVC 4

Apress®

精通 ASP.NET MVC 4

[美] Adam Freeman 著
李萍 徐燕萍 林逸 译

- 来自.NET专家的权威指南
- 在微软平台上构建最可维护、与标准兼容且运行最佳的Web应用

人民邮电出版社
POSTS & TELECOM PRESS

Pro
ASP.NET
MVC 4

精通 ASP.NET MVC 4

[美] Adam Freeman 著
李萍 徐燕萍 林逸 译

人民邮电出版社
北京

图书在版编目(CIP)数据

精通ASP.NET MVC 4 / (美) 弗瑞曼 (Freeman, A.)
著; 李萍, 徐燕萍, 林逸译. — 北京: 人民邮电出版社, 2014.10
ISBN 978-7-115-36097-7

I. ①精… II. ①弗… ②李… ③徐… ④林… III.
①网页制作工具—程序设计 IV. ①TP393.092

中国版本图书馆CIP数据核字(2014)第153722号

版 权 声 明

Pro ASP.NET MVC 4 (4th ed.)

By Adam Freeman, ISBN: 978-1-4302-4236-9

Original English language edition published by Apress Media.

Copyright © 2012 by Apress Media

Simplified Chinese-language edition copyright © 2014 by Post & Telecom Press

All rights reserved.

本书中文简体字版由 Apress L.P. 授权人民邮电出版社独家出版。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。

版权所有, 侵权必究。

◆ 著 [美] Adam Freeman

译 李 萍 徐燕萍 林 逸

责任编辑 张 涛

责任印制 彭志环 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京昌平百善印刷厂印刷

◆ 开本: 800×1000 1/16

印张: 38.25

字数: 991千字

2014年10月第1版

印数: 1-2500册

2014年10月北京第1次印刷

著作权合同登记号 图字 01-2013-6315 号

定价: 99.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

内 容 提 要

ASP.NET MVC 4 框架是微软 ASP.NET Web 平台的最新进展。该框架提供了一种与 ASP.NET 的所有优势整合在一起的高生产率编程模型，该模型提倡更整洁的代码架构、测试驱动开发以及强大的可扩展性。本书没有简单地解释或孤立地讨论“模型—视图—控制器”架构的核心概念，而是进行实战演示。通过一个拓展的教程，本书给出了创建一个可实际运行的电子商务 Web 应用程序，将 ASP.NET MVC 与最新的 C# 语言特性以及单元测试最佳实践结合在一起。通过获得这种有益的实践体验，读者可以发现 MVC 的长处和弱点，将最新学到的理论付诸于实践。

本书适合程序员、Web 开发者学习，也适合作为大中专院校相关专业的师生用书和培训学校的教材。

前言

ASP.NET MVC 是微软的一个 Web 开发框架，它整合了“模型—视图—控制器 (MVC)”架构的高效与整洁、敏捷开发最新的思想与技术，以及当前 ASP.NET 平台的精华部分。ASP.NET MVC 可以完全替代传统的 ASP.NET Web Form，除了一些微不足道的 Web 小项目之外，在各种 Web 开发项目中它都具有明显的优势。ASP.NET MVC 4 框架是微软 ASP.NET Web 平台的最新进展。它提供了一种与 ASP.NET 的所有优势整合在一起的高生产率编程模型，该模型提倡更整洁的代码架构、测试驱动开发以及强大的可扩展性。

这本书并未简单地解释或孤立地讨论“模型—视图—控制器”架构的核心概念，而是进行实战演示。你将通过一个拓展的教程，创建一个可实际运行的电子商务 Web 应用程序，它将 ASP.NET MVC 与最新的 C# 语言特性以及单元测试最佳实践结合在一起。通过获得这种有益的实践体验，你将发现 MVC 的长处和弱点，并将你最新学到的理论付诸于实践。

ASP.NET MVC 4 比以前版本含有许多显著提升，本书将向你演示运用这些新特性的最好方式。请阅读这本《精通 ASP.NET MVC 4》，并开始使用这种提倡更整洁的代码架构、测试驱动开发，以及强大可扩展性的高生产率编程模型吧！

本书的作者 Adam Freeman 目睹了 ASP.NET MVC 自第一版以来的成长。Adam 十五年前就开始设计并建立 Web 应用程序，曾负责过一些世界级最大型的、最引人注目的项目。

本书第 1 章帮助读者理解 ASP.NET MVC 的基本思想，包括 ASP.NET MVC 4 的新特性，并实际体验该框架的初步使用。第 2 章创建了一个新的 MVC 项目，并用它构造了一个简单的 MVC 数据录入应用程序，让你概览了 MVC 框架的体系结构和方法。第 3 章介绍了 MVC 架构模式，并将其与其他几种常见模式进行了比较。讨论了域模型的重要性，并创建了一个简单的例子。也介绍了依赖性注入，它让我们能够消除组件耦合，以强迫应用程序各部件之间严格分离。第 4 章首先概述了一个高效的 MVC 程序员需要了解的关键 C# 语言特性。这些特性被结合到了 LINQ 之中，本书将用 LINQ 查询数据。也向读者展示了新的 `async` 和 `await` 关键字，它们使异步方法更易于使用。第 5 章给出了 Razor 视图引擎以及如何用它生成 HTML 的。我们向你展示了如何引用控制器通过视图模型对象和视图包传递而来的数据，也向你演示了根据你所使用的的数据，如何用 Razor 表达式来剪辑发送给用户的响应。第 6 章讲解了我们所喜欢的可用于有效从事 MVC 开发的 3 个基本工具——Ninject、Visual Studio 内建的单元测试支持，以及 Moq。对这 3 种工具还有许多其他选择，开源的或是商业的。第 7 章建立了 SportsStore 应用程序最核心的基础结构，它带有一个产品存储库，其背后有 SQL Server 和 Entity Framework 的支持。还有一个控制器 ProductController，它可以产生分页的产品列表。并且，本章已经建立了 DI 和一个清晰友好的 URL 方案。第 8 章开始对 SportsStore 应用程序中面向顾客的部分进行了充实，让用户能够按分类进行浏览，并在适当的地方放置了一些基本的构建块，以便对购物车添加物品。第 9 章完成了 SportsStore 面向客户部分的所有主要部件，已经有了一个能够通过分类和页面进行浏览的产品分类、一个灵活的购物车和一个简单的结算过程。分离良好的体系结构，意味着我们可以很容易地修改应用程序任何片段的行为，而不必担心会引起其他问题或矛盾。第 10 章介绍了管理能力，并向你展示了如何实现 CRUD 操作，让管理员能够创建、读取、更新和删除存

储库中的产品。第 11 章以及前面几章演示了如何运用 ASP.NET MVC 框架来创建一个真实的电子商务应用程序。这个扩展的示例介绍了 MVC 框架的许多关键特性：控制器、动作方法、路由、视图、模型绑定、元数据、验证、布局、认证等。同时你也看到了如何使用与 MVC 相关的一些关键技术。这些技术包括实体框架（Entity Framework）、Ninject、Moq，以及 Visual Studio 对单元测试的支持。第 12 章展示了 Visual Studio MVC 项目的结构，以及各个部分如何构成一个整体。也接触了 MVC 框架的一个最重要的特征：约定。第 13 章深度讲解了路由系统。可以学习如何对输入 URL 进行匹配和处理，如何通过改变路由匹配 URL 片段的方式、使用默认值，以及可选片段等办法，对路由进行定制。我们也向你展示了约束路由，以缩小对请求进行匹配的范围，以及如何对静态内容的请求进行路由。第 14 章展示了 MVC 框架路由系统的高级特性，并展示了如何生成输出链接和 URL，以及如何定制路由系统。沿着这条路线，我们介绍了区域的概念，并考察了如何创建有用且有意义的 URL 方案。第 15 章控制器是 MVC 设计模式中的关键构建块之一。在本章中，学习了如何通过实现 `ApiController` 接口创建“原始的”控制器，以及通过从 `Controller` 类派生的办法创建更方便的控制器。也学习了动作方法在 MVC 框架控制器中所起的作用，而且它们是易于单元测试的。我们向你展示了由一个动作方法接收输入，并生成输出的不同方法，并且向你演示了使这一过程更为简单而灵活的各种 `ActionResult`。第 16 章讲解了如何将交叉关注的逻辑封装为过滤器。我们演示了各种可用的不同过滤器，以及如何实现它们。你也看到了如何将过滤器运用于控制器和动作方法，以及如何将它们用作全局过滤器。过滤器是在对请求进行处理时，对动作逻辑进行扩展的手段，却不需要把这种逻辑包含在动作方法中。第 17 章讲解了 MVC 框架如何创建控制器和调用方法。我们对关键接口的内建实现进行了考察和定制，也创建了自定义版本，以演示它们的工作机制。你学习了如何用动作方法选择器去区分动作方法，也看到了一些特殊种类的控制器，它们可用于提高你应用程序的请求处理能力。第 18 章讲解了 MVC 的视图系统，以及 Razor 视图引擎的细节，讲解了如何创建自定义视图引擎、如何定制默认 Razor 引擎的行为以及将动态内容插入到视图之中的不同技术。第 19 章介绍了辅助器方法的概念，可以将它们运用于视图，以生成可重用的内容块。从如何创建自定义的内联和外部辅助器开始，接着演示了可以用来创建 HTML 的 `form`、`input` 和 `select` 元素的辅助器。第 20 章讲解了可以通过模板视图辅助器的方法访问的模型模板系统，建立自己所需要的元数据，以及创建自定义模板，并在视图模型数据的显示和编辑方面，为你提供充分的灵活性。第 21 章讲解了 MVC 框架的渐进式 Ajax 特性，它让我们能够以一种简单而优雅的方式来利用 jQuery 库的功能优势，而不必对视图添加很多代码。第 22 章介绍了模型绑定过程的工作方式，演示了默认模型绑定器的操作，以及可以对该过程进行定制的不同方式。许多 MVC 框架的应用程序都只需要默认模型绑定器，它很好地协调处理了辅助器方法所生成的 HTML。但是，在一些更高级的应用程序中，使用自定义绑定器，以更为有效或更为特殊的方式来使用模型对象，是很有用的。第 23 章讲解了可用来执行模型验证的各种技术，这些技术保证了用户提供的数据与我们施加于数据模型的约束相一致。第 24 章展示了捆绑包和显示模式，它们在开发 Web 应用程序的客户端部分时是有用的。第 25 章展示了如何使用 Web API 特性，以创建 REST 化的 Web 服务，以便让客户端能够使用。第 26 章讲解了如何准备用于部署的应用程序，以及如何创建一个简单的 Windows Azure 服务，并将一个应用程序部署其上。部署应用程序有很多不同的方式，也有很多不同的目标平台，但本章所展示的过程可能是你期望的一个典型，即使你并未使用 Azure。

这些就是我们要教给你的 MVC 框架的全部。通过创建一个简单的应用程序，然后带领你在框架的不同组件中进行了一次综合旅行，并展示了如何对它们进行配置、定制或者完全取代。我们希望你的每个 MVC 项目都能成功，愿你能像我们愉快地写作本书一样，也能愉快地阅读，这是我们唯一的希望。

本书适合程序员、Web 开发者学习，也适合作为大中专院校相关专业师生的用书和培训学校的教材。源程序下载网址为：www.apress.com/source-code。编辑联系邮箱为：zhangtao@ptpress.com.cn。

关于作者



Adam Freeman 是一位经验丰富的 IT 专家，曾在一系列公司担任高级职务。最后的职务是一家全球性银行的首席技术官（CTO）和首席运营官（COO）。现已退休，目前正把他的时间用于写作和运动。

关于技术评阅人

Fabio Claudio Ferracchiati: 微软技术的高级顾问和高级分析师及开发者。他任职于 Brain Force (www.brainforce.com) 的意大利分支机构 (www.brainforce.it)。Fabio 是一位微软认证的 .NET 解决方案开发专家 (MCSD — Microsoft Certified Solution Developer for .NET)、微软认证的 .NET 应用程序开发专家 (MCAD — Microsoft Certified Application Developer for .NET)、微软认证专家 (MCP — Microsoft Certified Professional)，也是一位多产的作家和技术评阅人。在过去的 10 多年中，他已为意大利和许多国际杂志撰写了大量文章，并参与写作了 10 多部各类有关计算机主题的著作。

鸣谢

我要感谢 Apress 的每一位员工，是他们的艰辛工作才使本书得以出版。特别地，要感谢 Ewan Buckingham 启动并编辑了这一修订本，Christine Ricketts 跟踪了所有事务。还要感谢我们的技术审稿人 Fabio 先生，他的努力使本书在原来的基础上得到了很大提升。

Adam Freeman

目 录

第 1 部分 ASP.NET MVC 导论

第 1 章 伟大的思想	2
1.1 Web 开发简史	2
1.1.1 传统的 ASP.NET Web Form	3
1.1.2 ASP.NET Web Form 的问题	3
1.2 当今的 Web 开发	4
1.2.1 Web 标准与 REST	4
1.2.2 敏捷与测试驱动开发	5
1.2.3 Ruby on Rails	6
1.2.4 Sinatra	6
1.2.5 Node.js	6
1.3 ASP.NET MVC 的关键优点	7
1.3.1 MVC 的体系结构	7
1.3.2 可扩展性	8
1.3.3 在 HTML 及 HTTP 上的严密控制	8
1.3.4 可测试性	8
1.3.5 强大的路由系统	9
1.3.6 建立在 ASP.NET 平台的精华之上	9
1.3.7 现代 API	10
1.3.8 ASP.NET MVC 是开源的	10
1.4 谁该使用 ASP.NET MVC	10
1.4.1 与 ASP.NET Web Form 的比较	10
1.4.2 从 Web Form 迁移到 MVC	11
1.4.3 与 Ruby on Rails 的比较	11
1.4.4 与 MonoRail 的比较	11
1.5 ASP.NET MVC 4 的新特性	12
1.6 小结	12
第 2 章 第一个 MVC 应用程序	13
2.1 准备工作站	13

2.2	创建 ASP.NET MVC 新项目	13
2.2.1	添加第一个控制器	15
2.2.2	理解路由	17
2.3	渲染 Web 页面	18
2.3.1	创建并渲染视图	18
2.3.2	添加动态输出	20
2.4	创建一个简单的数据录入应用程序	21
2.4.1	设置场景	21
2.4.2	设计一个数据模型	22
2.4.3	链接动作方法	23
2.4.4	建立表单	26
2.4.5	处理表单	28
2.4.6	添加验证	31
2.4.7	完成示例	35
2.5	小结	36
第 3 章	MVC 模式	37
3.1	MVC 简史	37
3.2	理解 MVC 模式	37
3.2.1	理解域模型	38
3.2.2	MVC 的 ASP.NET 实现	38
3.2.3	MVC 与其他模式比较	39
3.3	运用域驱动开发 (DDD)	42
3.3.1	模拟一个例子域	42
3.3.2	通用语言	43
3.3.3	聚合与简化	44
3.3.4	定义存储库	45
3.4	建立松耦合组件	46
3.4.1	使用依赖性注入	47
3.4.2	一个 MVC 专用的 DI 示例	48
3.4.3	使用依赖性注入容器	49
3.5	自动测试初步	50
3.5.1	理解单元测试	50
3.5.2	理解集成测试	57
3.6	小结	57
第 4 章	基本语言特性	58
4.1	创建示例项目	58

4.2	使用自动实现的属性	59
4.3	使用对象与集合的初始化器	62
4.4	使用扩展方法	64
4.4.1	对接口运用扩展方法	66
4.4.2	创建过滤扩展方法	68
4.5	使用 Lambda 表达式	69
4.6	使用自动类型接口	72
4.7	使用匿名类型	73
4.8	执行语言集成查询	74
4.9	使用 Async 方法	79
4.10	小结	81
第 5 章	使用 Razor	82
5.1	创建示例项目	82
5.1.1	定义模型	82
5.1.2	定义控制器	83
5.1.3	创建视图	83
5.2	使用模型对象	85
5.3	使用布局	86
5.3.1	创建布局	86
5.3.2	运用布局	88
5.3.3	使用视图起始文件	88
5.3.4	演示共享布局	89
5.4	使用 Razor 表达式	92
5.4.1	插入数据值	92
5.4.2	设置标签属性的值	94
5.4.3	使用条件语句	96
5.4.4	枚举数组和集合	98
5.4.5	处理命名空间	100
5.5	小结	101
第 6 章	MVC 基本工具	102
6.1	创建示例项目	102
6.1.1	创建模型类	103
6.1.2	添加控制器	104
6.1.3	添加视图	104
6.2	使用 Ninject	105

6.2.1	理解问题	105
6.2.2	将 Ninject 添加到 Visual Studio 项目	107
6.2.3	Ninject 初步	107
6.2.4	建立 MVC 依赖性注入	109
6.2.5	指定属性与构造器参数值	113
6.2.6	使用条件绑定	115
6.3	Visual Studio 的单元测试	116
6.3.1	创建单元测试项目	117
6.3.2	创建单元测试	118
6.3.3	运行单元测试 (并失败)	121
6.3.4	实现特性	121
6.3.5	测试并修正代码	122
6.4	使用 Moq	123
6.4.1	理解问题	124
6.4.2	将 Moq 添加到 Visual Studio 项目	125
6.4.3	对单元测试添加模仿对象	126
6.4.4	创建更复杂的模仿对象	129
6.5	小结	131
第 7 章	SportsStore——一个真实的应用程序	132
7.1	开始	133
7.1.1	创建 Visual Studio 解决方案和项目	133
7.1.2	添加引用	134
7.1.3	设置 DI 容器	135
7.1.4	运行应用程序	136
7.2	从域模型开始	137
7.2.1	创建一个抽象的存储库	138
7.2.2	创建模仿存储库	138
7.3	显示产品列表	139
7.3.1	添加控制器	139
7.3.2	添加视图	141
7.3.3	设置默认路由	142
7.3.4	运行应用程序	142
7.4	准备数据库	143
7.4.1	创建数据库	143
7.4.2	定义数据库方案	144
7.4.3	向数据库添加数据	146

7.4.4	创建实体框架上下文	146
7.4.5	创建 Product 存储库	147
7.5	添加分页	149
7.5.1	显示页面链接	151
7.5.2	改进 URL	158
7.6	设置内容样式	159
7.6.1	定义布局中的公用内容	159
7.6.2	添加 CSS 样式	160
7.6.3	创建分部视图	161
7.7	小结	163
第 8 章	SportsStore: 导航	164
8.1	添加导航控件	164
8.1.1	过滤产品列表	164
8.1.2	调整 URL 方案	167
8.1.3	建立分类导航菜单	169
8.1.4	修正页面计数	176
8.2	建立购物车	178
8.2.1	定义购物车实体	178
8.2.2	添加“加入购物车”按钮	182
8.2.3	实现购物车控制器	183
8.2.4	显示购物车内容	185
8.3	小结	187
第 9 章	SportsStore: 完成购物车	188
9.1	使用模型绑定	188
9.2	完成购物车	192
9.2.1	删除购物车物品	192
9.2.2	添加购物车摘要	193
9.3	递交订单	196
9.3.1	扩充域模型	196
9.3.2	添加结算过程	196
9.3.3	实现订单处理器	199
9.3.4	注册(接口)实现	201
9.3.5	完成购物车控制器	202
9.3.6	显示验证错误	205
9.3.7	显示致谢页面	206

9.4 小结	207
第 10 章 SportsStore: 管理	208
10.1 添加分类管理	208
10.1.1 创建 CRUD 控制器	208
10.1.2 创建新的布局	210
10.1.3 实现 List 视图	212
10.1.4 编辑产品	214
10.1.5 创建新产品	226
10.1.6 删除产品	227
10.2 小结	229
第 11 章 SportsStore: 安全性与收尾工作	230
11.1 实现管理控制器的安全	230
11.1.1 运用过滤器进行授权	231
11.1.2 创建认证提供者	232
11.1.3 创建 Account 控制器	234
11.1.4 创建视图	235
11.2 图像上传	238
11.2.1 扩展数据库	238
11.2.2 增强域模型	239
11.2.3 创建上传用户界面的元素	239
11.2.4 将图像保存到数据库	240
11.2.5 实现 GetImage 动作方法	241
11.2.6 显示产品图像	243
11.3 小结	244

第 2 部分 ASP.NET MVC 4 细节

第 12 章 MVC 项目综述	246
12.1 使用 Visual Studio 的 MVC 项目	246
12.2 调试 MVC 应用程序	250
12.2.1 创建项目	250
12.2.2 运行 Visual Studio 调试器	252
12.2.3 引发 Visual Studio 调试器中断	254
12.2.4 使用编辑并继续	257
12.3 小结	260

第 13 章 URL 路由	261
13.1 创建示例项目	261
13.2 URL 模式	263
13.3 创建并注册一条简单路由	265
13.4 定义默认值	270
13.5 使用静态 URL 片段	272
13.6 定义自定义片段变量	276
13.6.1 用自定义变量作为动作方法参数	278
13.6.2 定义可选 URL 片段	279
13.6.3 定义可变长路由	281
13.6.4 按命名空间区分控制器优先顺序	283
13.7 约束路由	286
13.7.1 用正则表达式约束路由	286
13.7.2 将一条路由约束到一组指定的值	287
13.7.3 使用 HTTP 方法约束路由	287
13.7.4 定义自定义约束	288
13.8 对磁盘文件的请求进行路由	290
13.8.1 配置应用程序服务器	291
13.8.2 为磁盘文件定义路由	292
13.9 绕过路由系统	292
13.10 小结	293
第 14 章 高级路由特性	294
14.1 准备示例项目	294
14.2 在视图中生成输出 URL	294
14.2.1 用路由系统生成输出 URL	295
14.2.2 以其他控制器为目标	298
14.2.3 传递额外的值	299
14.2.4 指定 HTML 标签属性	300
14.2.5 生成链接中的全限定 URL	301
14.2.6 生成 URL (而不是链接)	302
14.2.7 在动作方法中生成输出 URL	302
14.2.8 根据指定路由生成 URL	303
14.3 定制路由系统	304
14.3.1 创建自定义的 RouteBase 实现	304
14.3.2 创建自定义路由处理程序	308

14.4 使用区域	309
14.4.1 创建一个区域	310
14.4.2 填充区域	311
14.4.3 解析不明确的控制器问题	313
14.4.4 生成对区域动作的链接	314
14.5 URL 方案最佳做法	315
14.5.1 使 URL 整洁和人性化	315
14.5.2 GET 和 POST: 选用正确的一个	316
14.6 小结	316
第 15 章 控制器和动作	317
15.1 控制器介绍	317
15.1.1 准备示例项目	317
15.1.2 用 Icontroller 创建一个控制器	317
15.1.3 创建派生于 Controller 类的控制器	318
15.2 接收输入	320
15.2.1 通过上下文对象获取数据	320
15.2.2 使用动作方法参数	321
15.3 产生输出	323
15.3.1 理解动作结果	325
15.3.2 通过渲染视图返回 HTML	328
15.3.3 将数据从动作方法传递给视图	331
15.3.4 执行重定向	334
15.3.5 返回错误及 HTTP 代码	339
15.4 小结	340
第 16 章 过滤器	341
16.1 使用过滤器	341
16.1.1 过滤器的四种基本类型	342
16.1.2 将过滤器运用于控制器和动作方法	343
16.2 创建示例项目	344
16.3 使用授权过滤器	344
16.3.1 运用自定义授权过滤器	346
16.3.2 使用内建的授权过滤器	347
16.4 使用异常过滤器	348
16.4.1 创建异常过滤器	348
16.4.2 运用异常过滤器	350