



Professional ASP.NET MVC 5

# ASP.NET MVC 5

## 高级编程 (第5版)

Azure Web团队首席社区架构师Scott Hanselman作序推荐

[美] Jon Galloway Brad Wilson 著  
K. Scott Allen David Matson 译  
孙远帅



清华大学出版社

.NET 开发经典名著

# ASP.NET MVC 5 高级编程

(第 5 版)

[美] Jon Galloway  
Brad Wilson 著  
K. Scott Allen  
David Matson  
孙远帅 译

清华大学出版社

北 京

Jon Galloway, Brad Wilson, K. Scott Allen, David Matson  
Professional ASP.NET MVC 5  
EISBN: 978-1-118-79475-3  
Copyright © 2014 by John Wiley & Sons, Inc., Indianapolis, Indiana  
All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 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: 01-2014-7614

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签,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

ASP.NET MVC 5 高级编程(第 5 版)/(美)加洛韦(Galloway, J.) 等著;孙远帅 译. —北京:清华大学出版社, 2015

(.NET 开发经典名著)

书名原文: Professional ASP.NET MVC 5

ISBN 978-7-302-39062-6

I. ①A… II. ①加… ②孙… 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(2015)第 011268 号

责任编辑:王 军 于 平

装帧设计:孔祥峰

责任校对:成凤进

责任印制:杨 艳

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:30 字 数:768 千字

版 次:2015 年 2 月第 1 版 印 次:2015 年 2 月第 1 次印刷

印 数:1~4000

定 价:59.80 元

产品编号:058142-01

# 译者序

作为微软.NET Framework的重要组成部分，ASP.NET自2002年发布以来，迅速成为服务器端应用程序的热门开发工具，数以万计使用ASP.NET开发的网站如雨后春笋般地出现在网络上，越来越多的IT从业者使用ASP.NET技术开发网站。

随同ASP.NET一起问世的Web Forms框架，由于其性能稳定、高度抽象、易于上手等特点得到了广大IT从业者的青睐。随着使用的深入，在大型工程项目中，Web Forms框架暴露出了很多缺点，Web Forms根据抽象模型生成HTML标记，由于不能完全控制HTML标记，导致HTML标记冗余，布局混乱。此外，ViewState中存储的数据可能会远超所需，导致HTML页面加载过慢，最糟糕的一点，Web Forms不能有效地分离业务逻辑和表现层逻辑。在分离应用程序内部关注点方面，MVC是一种强大而简洁的方式，尤其在Web应用程序中。因此，为了有效地解决这些问题，微软ASP.NET团队提出了一个有效的解决方案，就是在ASP.NET中引入MVC框架，这样ASP.NET MVC架构就应运而生了。

ASP.NET MVC架构降低了程序间的耦合性，测试驱动开发，代码重用性好，使用路由机制解析URL，并支持自定义路由机制和视图引擎。此外，ASP.NET MVC基于CLR和成熟的MVC架构构建，不支持ViewState，这就意味着它没有自动状态管理机制，从而降低了页面传递的数据，提高了程序性能。同时也不支持服务器端控件，这样可以实现完全控制HTML，可以很容易地与第三方JavaScript库集成开发。

自2009年3月第一版发布以来，在过去的5年时间里，ASP.NET MVC已经发布了5个主要版本，期间还有很多临时版本。升级速度如此之快，这与开发人员使用率是分不开的，市场需求和反馈带动了技术的步步升级，甚至革新。相对于第4版，本书新增了第12章“应用AngularJS构建单页面应用程序”和附录部分。其中第12章结合一个示例Web应用程序介绍了第三方JavaScript库AngularJS，如何构建Web API，以及如何结合AngularJS构建Web应用程序。附录A主要介绍了ASP.NET MVC 5.1的常用功能。例如，Enum支持原理，如何使用自定义约束执行特性路由，以及Bootstrap和JavaScript增强的使用方法等。

书籍给出了方法，检验方法是否有效的途径就是实践。行胜于言，从书中学习的方法，一定要在实践中应用检验。这一点在计算机学科中尤为重要。如果只是通过读书学习技术架构，而没有在工程应用中实践，这样不足以对技术架构有刻骨铭心的理解，不会发现ASP.NET MVC架构的精妙，当然也不会发现其中的不足，不能为ASP.NET MVC的下一次升级完善作贡献。

在这里要感谢清华大学出版社的编辑们，他们对本书的翻译校正投入了巨大的心血。他们对工作的热情和一丝不苟的工作态度深深地感染着我，我有幸能够翻译《ASP.NET MVC 3高级编程(第3版)》、《ASP.NET MVC 4高级编程(第4版)》和《ASP.NET MVC 5高级编程(第5版)》，并顺利交稿，与他们的鼓励和帮助是分不开的。厦门大学的邹权教授对我翻译《ASP.NET

MVC 4 高级编程(第4版)》也提供了很多帮助，并参与了部分翻译工作。最后，感谢对我工作学习一直全力支持的父母和妹妹，还有所有帮助支持我的人。

本书全部章节由孙远帅翻译，参与本次翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松、蒋晓冬、邱培强、洪妍、李亮辉、高娟妮、曹小震、陈笑。在此一并表示感谢！

对于本书的翻译工作，我力求做到“信、达、雅”，限于本人水平有限，翻译不足之处，还请不吝赐教，不胜感激！

译 者

# 序

我很高兴能为本书写序，本书介绍了最新发布的ASP.NET MVC版本，并由一支优秀的作者团队编写。本书的作者都是我的好友，他们都是非常优秀的技术专家。

Jon Galloway是专注于Azure和ASP.NET的技术传道者。处在这个角色，使他有接触成千上万的或者新接触、或者十分熟悉ASP.NET MVC的开发人员。他负责编写了MVC Music Store教程，该教程帮助成千上万的开发人员编写了他们的第一个ASP.NET MVC应用程序。他与各种ASP.NET社区的互动使得他拥有很强的洞察力，知道开发人员如何开始、学习和掌握ASP.NET MVC。

Brad Wilson不仅是我最爱的怀疑论者，而且他在Microsoft任职期间，也帮助构建了几个版本的ASP.NET MVC。从动态数据到数据注解，再到测试等，没有作为程序员的Brad干不了的事。他从事过许多开源项目(如XUnit .NET)，并继续推动Microsoft公司的内部和外部人员走向光明。

Phil Haack是ASP.NET MVC项目经理，他从一开始就参与该项目。因为他有植根于社区和开源的背景，所以我一直认为他不仅是一名优秀的技术人员，而且还是我的一位亲密的朋友。在Microsoft工作期间，Phil还从事新的.NET包管理器NuGet的开发。

David Matson在本书这一版中加入了作者团队。他是Microsoft的一名高级开发人员，带来了关于ASP.NET MVC和Web API的新特性的大量详尽信息，因为他帮助开发了这些技术。David为本书这一版带来了深刻的技术知识和指导。

最后也是相当重要的是，K. Scott Allen增强了团队的力量，不仅仅是因为他明智地决定使用他听起来更加智能的中间名，而且也因为他带来了一名世界级著名培训师的经验 and 智慧。Scott Allen是Pluralsight技术团队中的一员，曾经在财富50强公司从事网站和创业咨询方面的工作。他善良、体贴、值得尊重，重要的是他非常透彻地了解自己。

随着ASP.NET Web开发平台的发展，这些伙计团结在一起共同把《ASP.NET MVC 5高级编程(第5版)》一书推升到了一个新的高度。该平台目前正在由全球数百万的开发人员使用。一个充满朝气的社区支持该平台的在线版和离线版，线上论坛([www.asp.net](http://www.asp.net))平均每天都有成千上万的问答。

ASP.NET和ASP.NET MVC 5的应用面很广，像新闻网站、网上零售商店以及我们最喜欢的社交网站。除此之外，或许我们当地的运动队、读书俱乐部或博客使用的也是ASP.NET MVC 5。

当ASP.NET MVC刚刚被引入时，它打破了很多领域。尽管使用的是旧模式，但是这些模式对于现有的ASP.NET社区来说都是新的；它在生产率和控制、功能和灵活性之间求得了微妙平衡。今天，对我来说，ASP.NET MVC 5代表了选择——语言的选择、框架的选择、开源库的选择、模式的选择。一切都是可插拔的。ASP.NET MVC 5是我们对环境绝对控制的缩影——如果喜欢，就使用；如果不喜欢，就改变。我们可以按照自己想要的方式进行单元测试。

试，创建自己想要的组件，使用自己选择的JavaScript框架。

ASP.NET MVC 5中最令人振奋的更新可能就是引入了One ASP.NET的概念。在这个版本中，我们很容易开发混合应用程序，并在ASP.NET MVC和Web Forms之间共享代码。ASP.NET MVC运行在公共的ASP.NET核心组件之上，例如ASP.NET Identity、ASP.NET Scaffolding和Visual Studio New Project体验。这意味着我们可以跨平台运用我们的ASP.NET技能，不管这些平台是ASP.NET MVC、Web Forms、Web Pages、Web API还是SignalR。这些更新设计了扩展点，用于与其他框架(如NancyFX和ServiceStack)共享代码和库。

我建议到[www.asp.net/mvc](http://www.asp.net/mvc)上下载最新的内容，以及最新的示例、视频和教程。

我们希望本书讲解的内容能够使你精通ASP.NET MVC 5历程中的下一步。

——Scott Hanselman  
Microsoft Azure Web团队首席社区架构师

# 前 言

对于一名ASP.NET开发人员来说，这是一个伟大的时刻！

无论是对于已经拥有ASP.NET多年开发经验的开发人员，还是对于刚刚入门的初学者，现在都是深入学习ASP.NET MVC的绝佳时机。ASP.NET MVC从一开始就有很多乐趣，但最近两个版本添加了许多特性，使整个开发过程变得非常愉悦。

ASP.NET MVC 3带来了像Razor视图引擎这样的新特性，与NuGet包管理系统和jQuery内置整合来简化Ajax开发。ASP.NET MVC 5继续这一趋势，添加了更新的可视化设计、移动Web支持、使用ASP.NET Web API的HTTP服务、内置支持OAuth与流行网站的整合等。这样我们就可以快速地开始使用全功能Web应用程序。

这也不是简单地利用拖放功能提高短期生产率。这一切都建立在一个基于模式的Web框架上，当需要时，这个框架可帮助我们控制应用程序的每个方面。

加入我们会踏上有趣翔实的ASP.NET MVC 5之旅！

## 本书读者对象

本书由浅入深地介绍ASP.NET MVC，是一本优秀的ASP.NET MVC教程。

如果刚刚接触ASP.NET MVC，本书首先会帮助学习MVC概念，然后演示如何在应用代码示例中应用这些概念。本书作者已经指导成千上万名开发人员开始学习ASP.NET MVC，指导怎样安排结构思路，以便快速创建，入门开发。

我们知道许多读者都熟悉ASP.NET Web Forms，在一些上下文中，我们介绍它们之间的异同来帮助理解它们之间的关系。事实上，ASP.NET MVC 5不是ASP.NET Web Forms的替代品。许多Web开发人员也使用其他Web框架，比如Ruby on Rails、Node.js、Django，一些PHP框架等，这些框架都适用于MVC(模型-视图-控制器，Model-View-Controller)应用模式。如果你属于这类开发人员，或者只是好奇，本书就适合你。

我们也付出了很大努力，确保本书能够为拥有ASP.NET MVC经验的开发人员提供一些帮助。在本书的各个章节，我们介绍了组件设计原理，以及如何最好地使用它们。我们添加了新的内容，包括大大扩展了介绍路由的一章，以介绍ASP.NET MVC 5中新增的特性路由功能。我们还利用从NuGet Gallery开发团队那里直接得到的知识，更新了最后一章的NuGet Gallery案例分析，解释了NuGet开发团队如何构建和开发真实世界中高容量的ASP.NET MVC网站。另外，K. Scott Allen还新撰写了一章，解释了如何使用AngularJS构建单页面应用程序。



## 本书组织结构

本书分为两大部分，每部分由几个章节构成。前6章主要介绍了MVC模式，以及ASP.NET MVC是如何实现MVC模式的。

第1章“入门”帮助你开始进行ASP.NET MVC 5开发。首先介绍了ASP.NET MVC的概念，然后解释ASP.NET MVC 5如何顺应以前的发布版本。最后，在确保正确安装软件之后，帮助你开始创建你的第一个ASP.NET MVC 5应用程序。

第2章“控制器”讲解控制器和操作的基础内容。你开始编写一些基本的“hello world”示例，然后创建从URL中提取信息并在屏幕上显示应用程序。

第3章“视图”介绍如何从控制器操作中使用视图模板控制输出的可视化表示。此外，还会全面地介绍Razor视图引擎，其中包括帮助组织和维护的语法和特征。

第4章“模型”帮助你学习如何使用模型在控制器和视图之间传递信息，以及如何使用Entity Framework的Code First开发集成数据库和模型。

第5章“表单和HTML辅助方法”深入介绍编辑情形，解释ASP.NET MVC处理表单的方式。你将从本章中学习到如何使用HTML辅助方法精简视图。

第6章“数据注解和验证”介绍如何使用特性定义模型显示、编辑和验证的规则。

接下来的10章以前面的内容为基础，介绍了一些更加高级的概念和应用程序。

第7章“成员资格、授权和安全性”介绍如何确保ASP.NET MVC应用程序安全，并指出常见的安全陷阱以及避开这些陷阱的方法。此外，你还会学习到如何利用ASP.NET MVC应用程序中的ASP.NET成员资格和授权特性来控制访问权限。另外还将学到新增的ASP.NET Identity系统的重要信息。

第8章“Ajax”介绍ASP.NET MVC应用程序中的Ajax程序，并特别强调jQuery和jQuery插件。本章中，你将会学习到如何使用ASP.NET MVC的Ajax辅助方法，以及如何高效地应用jQuery验证系统。

第9章“路由”深入介绍用来管理如何将URL映射到控制器操作的路由机制。本章介绍了传统路由和新增的特性路由，展示了如何结合使用这两种路由，并解释了两种路由的适用场合。

第10章“NuGet”介绍NuGet包管理系统。通过本章内容，你将学习到如何将NuGet关联到ASP.NET MVC，如何安装NuGet以及如何使用NuGet来安装、更新和创建新包。

第11章“ASP.NET Web API”展示如何使用ASP.NET Web API创建HTTP服务。

第12章“应用AngularJS构建单页面应用程序”介绍如何将ASP.NET MVC技能和Web API技能与流行的AngularJS库结合起来使用，创建出单页面应用程序。另外还提供了一个有趣的“*At The Movies*”示例应用程序。

第13章“依赖注入”介绍依赖注入以及如何在应用程序中利用依赖注入。

第14章“单元测试”教你如何在ASP.NET应用程序中使用测试驱动开发，并提供编写高

效测试的一些有益忠告。

第15章“扩展ASP.NET MVC”深入讲解ASP.NET MVC中的扩展点,并展示如何扩展MVC框架来满足你的具体需求。

第16章“高级主题”介绍一些高级主题,这些主题在阅读本书前15章之前讲解可能会使你感到吃力。本章涵盖Razor、基架系统、路由机制、模板和控制器的一些复杂应用。

第17章“ASP.NET MVC实战:构建NuGet.org网站”结合学习的每个知识点来进行NuGet Gallery网站(<http://nuget.org>)案例研究。在这里,你会学习到,当使用ASP.NET MVC构建高性能网站时,高级ASP.NET工程师处理测试、成员资格、部署和数据迁移的方法。

#### 经验丰富的读者请注意:

本书前6章的节奏有点慢。这些章节介绍了ASP.NET MVC中的一些基本概念,并假定读者没有多少相关经验。如果读者已经熟悉了MVC,可以快速浏览前几章。从第7章开始,讲解速度将会加快。

## 使用本书的条件

为使用ASP.NET MVC 5,你可能需要安装Visual Studio。可以使用Microsoft Visual Studio Express 2013的Web版或Visual Studio 2013的任何付费版本(如Visual Studio 2013 Professional)。Visual Studio 2013中包含了ASP.NET MVC 5。可以从以下网址下载Visual Studio和Visual Studio Express:

- Visual Studio: [www.microsoft.com/vstudio](http://www.microsoft.com/vstudio)
- Visual Studio Express: [www.microsoft.com/express/](http://www.microsoft.com/express/)

也可以在Visual Studio 2012中使用ASP.NET MVC 5。ASP.NET MVC 5包含在Visual Studio 2012的ASP.NET和Web Tools更新中,下载地址如下:

- ASP.NET and Web Tools 2013.2 for Visual Studio 2012: <http://www.microsoft.com/en-us/download/41532>

第1章详细介绍了软件需求,并演示了如何在开发机和服务器上安装。

## 源代码

整本书中,你会注意到,当建议你安装NuGet包以尝试一些样例代码时,我们会放置如下标识:

```
Install-Package SomePackageName
```

NuGet是Outercurve Foundation为.NET和Visual Studio而编写的包管理器,后来被Microsoft公司整合到了ASP.NET MVC中。

我们不必再在Wrox网站上搜索源代码示例的压缩文件了,因为我们可以通过使用NuGet轻松地把这些文件添加到ASP.NET MVC应用程序中。我们认为自此尝试样例将不再痛苦,而

变得更容易、更方便。第10章将详细介绍NuGet系统。

如果你想下载NuGet包，以便在以后不能上网时使用，这些包也可以从www.wrox.com下载。登录该网站之后，只需要使用Search框或标题列表中的一个找到书的标题，单击本书详细页面上的Download Code链接，即可下载本书涉及的所有源代码。另外，也可从<http://www.tupwk.com.cn/downpage>下载本书的源代码。



由于许多图书的标题都很类似，所以按ISBN搜索是最简单的；本书英文版的ISBN是978-1-118-79475-3。

在下载了代码后，只需要用自己喜欢的解压缩软件对它们进行解压缩即可。另外，也可以进入<http://www.wrox.com/dynamic/books/download.aspx>上的Wrox代码下载页面，查看本书和其他Wrox图书的源代码。

## 勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果你在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给wkservice@vip.163.com发电子邮件，我们就会检查你的信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书的勘误表，可以登录<http://www.wrox.com>，通过Search框或书名列表查找本书，然后在本书的详细页面上，单击Errata链接。在这个页面上可以查看到Wrox编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是[www.wrox.com/misc-pages/booklist.shtml](http://www.wrox.com/misc-pages/booklist.shtml)。

## p2p.wrox.com

要与作者和同行讨论，请加入p2p.wrox.com上的P2P论坛。这个论坛是一个基于Web的系统，便于你张贴与Wrox图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给你传送感兴趣的论题。Wrox作者、编辑和其他业界专家和读者都会到这个论坛上探讨问题。

在<http://p2p.wrox.com>上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入p2p.wrox.com，单击Register链接。
- (2) 阅读使用协议，并单击Agree按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击Submit按钮。
- (4) 你会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。



不加入P2P也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在Web上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的Subscribe to this Forum图标。

关于使用Wrox P2P的更多信息，可阅读P2P FAQ，了解论坛软件的工作情况以及P2P和Wrox图书的许多常见问题。要阅读FAQ，可以在任意P2P页面上单击FAQ链接。

# 目 录

第 1 章 入门 .....	1	1.4.2 约定优于配置 .....	21
1.1 ASP.NET MVC 简介 .....	1	1.4.3 约定简化通信 .....	22
1.1.1 ASP.NET MVC 如何 适应 ASP.NET .....	1	1.5 小结 .....	22
1.1.2 MVC 模式简介 .....	2	第 2 章 控制器 .....	23
1.1.3 MVC 在 Web 框架中的 应用 .....	2	2.1 控制器的角色 .....	23
1.1.4 ASP.NET MVC 5 的 发展历程 .....	3	2.2 示例应用程序: MVC Music Store .....	25
1.1.5 ASP.NET MVC 4 概述 .....	6	2.3 控制器基础 .....	28
1.1.6 开源发布 .....	9	2.3.1 简单示例: Home Controller .....	28
1.2 ASP.NET MVC 5 概述 .....	9	2.3.2 创建第一个控制器 .....	30
1.2.1 One ASP.NET .....	10	2.3.3 控制器操作中的参数 .....	33
1.2.2 新的 Web 项目体验 .....	10	2.4 小结 .....	35
1.2.3 ASP.NET Identity .....	10	第 3 章 视图 .....	36
1.2.4 Bootstrap 模板 .....	11	3.1 视图的作用 .....	37
1.2.5 特性路由 .....	13	3.2 视图的基础知识 .....	37
1.2.6 ASP.NET 基架 .....	13	3.3 理解视图约定 .....	40
1.2.7 身份验证过滤器 .....	13	3.4 强类型视图 .....	41
1.2.8 过滤器重写 .....	13	3.4.1 ViewBag 的不足 .....	41
1.3 安装 MVC 5 和创建 应用程序 .....	13	3.4.2 理解 ViewBag、ViewData 和 ViewDataDictionary .....	43
1.3.1 ASP.NET MVC 5 的 软件需求 .....	13	3.5 视图模型 .....	44
1.3.2 安装 ASP.NET MVC 5 .....	14	3.6 添加视图 .....	45
1.3.3 创建 ASP.NET MVC 5 应用程序 .....	14	3.7 Razor 视图引擎 .....	47
1.3.4 New ASP.NET Project 对话框 .....	15	3.7.1 Razor 的概念 .....	48
1.4 ASP.NET MVC 应用程序的 结构 .....	19	3.7.2 代码表达式 .....	49
1.4.1 ASP.NET MVC 和约定 .....	21	3.7.3 HTML 编码 .....	50
		3.7.4 代码块 .....	52
		3.7.5 Razor 语法示例 .....	52
		3.7.6 布局 .....	54
		3.7.7 ViewStart .....	56
		3.8 指定部分视图 .....	57

3.9	小结	58	5.4.1	Html.ActionLink 和 Html.RouteLink	104
<b>第 4 章</b>	<b>模型</b>	<b>59</b>	5.4.2	URL 辅助方法	105
4.1	为 MVC Music Store 建模	60	5.4.3	Html.Partial 和 Html.RenderPartial	106
4.2	为商店管理器构造基架	63	5.4.4	Html.Action 和 Html.RenderAction	107
4.2.1	基架的含义	63	5.5	小结	108
4.2.2	基架和实体框架	64	<b>第 6 章</b>	<b>数据注解和验证</b>	<b>109</b>
4.2.3	执行基架模板	66	6.1	为验证注解订单	110
4.2.4	执行基架代码	71	6.1.1	验证注解的使用	112
4.3	编辑专辑	75	6.1.2	自定义错误提示 消息及其本地化	116
4.3.1	创建编辑专辑的资源	76	6.1.3	注解的后台原理	117
4.3.2	响应编辑时的 POST 请求	79	6.1.4	控制器操作和 验证错误	118
4.4	模型绑定	80	6.2	自定义验证逻辑	120
4.4.1	DefaultModelBinder	81	6.2.1	自定义注解	120
4.4.2	显式模型绑定	82	6.2.2	IValidatableObject	123
4.5	小结	84	6.3	显示和编辑注解	124
<b>第 5 章</b>	<b>表单和 HTML 辅助方法</b>	<b>85</b>	6.3.1	Display	124
5.1	表单的使用	85	6.3.2	ScaffoldColumn	125
5.1.1	action 和 method 特性	86	6.3.3	DisplayFormat	125
5.1.2	GET 方法还是 POST 方法	86	6.3.4	ReadOnly	125
5.2	HTML 辅助方法	89	6.3.5	DataType	126
5.2.1	自动编码	90	6.3.6	UIHint	126
5.2.2	辅助方法的使用	91	6.3.7	HiddenInput	126
5.2.3	HTML 辅助方法的 工作原理	91	6.4	小结	126
5.2.4	设置专辑编辑表单	92	<b>第 7 章</b>	<b>成员资格、授权和安全性</b>	<b>127</b>
5.2.5	添加输入元素	93	7.1	安全性: 无趣、但极其重要	127
5.2.6	辅助方法、模型和 视图数据	98	7.2	使用 Authorize 特性登录	129
5.2.7	强类型辅助方法	100	7.2.1	保护控制器操作	129
5.2.8	辅助方法和模型元数据	101	7.2.2	Authorize 特性在表单身份 验证和 AccountController 控制器中的用法	133
5.2.9	模板辅助方法	101	7.2.3	Windows Authentication	134
5.2.10	辅助方法和 ModelState	102	7.3	要求角色成员使用 Authorize 特性	137
5.3	其他输入辅助方法	103			
5.3.1	Html.Hidden	103			
5.3.2	Html.Password	103			
5.3.3	Html.RadioButton	103			
5.3.4	Html.CheckBox	104			
5.4	渲染辅助方法	104			

7.4	扩展用户身份	138	8.4.1	jQuery UI	196
7.4.1	存储额外的用户资料数据	139	8.4.2	使用 jQuery UI 实现自动完成部件	197
7.4.2	持久化控制	139	8.4.3	JSON 和客户端模板	200
7.4.3	管理用户和角色	139	8.4.4	Bootstrap 插件	204
7.5	通过 OAuth 和 OpenID 的外部登录	139	8.5	提高 Ajax 性能	206
7.5.1	注册外部登录提供器	141	8.5.1	使用内容分发网络	206
7.5.2	配置 OpenID 提供器	142	8.5.2	脚本优化	206
7.5.3	配置 OAuth 提供器	144	8.5.3	捆绑和微小	207
7.5.4	外部登录的安全性	144	8.6	小结	208
7.6	Web 应用程序中的安全向量	145	第 9 章	路由	209
7.6.1	威胁: 跨站脚本	146	9.1	统一资源定位符——URL	209
7.6.2	威胁: 跨站请求伪造	154	9.2	路由概述	210
7.6.3	威胁: cookie 盗窃	158	9.2.1	对比路由和 URL 重写	211
7.6.4	威胁: 重复提交	160	9.2.2	路由方法	211
7.6.5	威胁: 开放重定向	162	9.2.3	定义特性路由	212
7.7	适当的错误报告和堆栈跟踪	166	9.2.4	定义传统路由	221
7.7.1	使用配置转换	167	9.2.5	选择特性路由还是传统路由	229
7.7.2	在生产环境中使用 Retail 部署配置	168	9.2.6	路由命名	229
7.7.3	使用专门的错误日志系统	168	9.2.7	MVC 区域	231
7.8	安全回顾和有用资源	168	9.2.8	catch-all 参数	233
7.9	小结	169	9.2.9	段中的多个路由参数	233
第 8 章	Ajax	171	9.2.10	StopRoutingHandler 和 IgnoreRoute	234
8.1	jQuery	172	9.2.11	路由的调试	235
8.1.1	jQuery 的特性	172	9.3	揭秘路由如何生成 URL	236
8.1.2	非侵入式 JavaScript	175	9.3.1	URL 生成的高层次概述	237
8.1.3	jQuery 的用法	176	9.3.2	URL 生成详解	237
8.2	Ajax 辅助方法	181	9.3.3	外界路由值	238
8.2.1	在项目中添加非侵入式 Ajax 脚本	182	9.3.4	Route 类生成 URL 的若干示例	240
8.2.2	Ajax 的 ActionLink 方法	183	9.4	揭秘路由如何绑定到操作	241
8.2.3	HTML 5 特性	185	9.4.1	高层次请求的路由管道	241
8.2.4	Ajax 表单	186	9.4.2	路由数据	242
8.3	客户端验证	188	9.5	自定义路由约束	242
8.3.1	jQuery 验证	188	9.6	Web Forms 和路由机制	243
8.3.2	自定义验证	191	9.7	小结	244
8.4	辅助方法之外	196			

<b>第 10 章 NuGet</b> .....	245	11.4.1 Web 托管 Web API 的 配置 .....	279
10.1 NuGet 概述 .....	245	11.4.2 自托管 Web API 的配置 .....	280
10.2 以包的形式添加库 .....	246	11.5 向 Web API 添加路由 .....	282
10.2.1 查找包 .....	247	11.6 绑定参数 .....	283
10.2.2 安装包 .....	248	11.7 过滤请求 .....	285
10.2.3 更新包 .....	251	11.8 启用依赖注入 .....	286
10.2.4 包恢复 .....	251	11.9 探索 API 编程 .....	286
10.2.5 包管理器控制台的 用法 .....	252	11.10 跟踪应用程序 .....	287
10.3 创建包 .....	254	11.11 Web API 示例: ProductsController .....	287
10.3.1 打包项目 .....	255	11.12 小结 .....	289
10.3.2 打包文件夹 .....	256	<b>第 12 章 应用 AngularJS 构建 单页面应用程序</b> .....	290
10.3.3 配置文件和源代码 转换 .....	256	12.1 理解和安装 AngularJS .....	291
10.3.4 NuSpec 文件 .....	257	12.1.1 AngularJS 简介 .....	291
10.3.5 元数据 .....	258	12.1.2 本章目标 .....	291
10.3.6 依赖库 .....	259	12.1.3 入门 .....	291
10.3.7 指定要包含的文件 .....	260	12.1.4 向网站中添加 AngularJS .....	293
10.3.8 工具 .....	260	12.1.5 数据库设置 .....	294
10.3.9 框架和轮廓定位 .....	263	12.2 创建 Web API .....	296
10.3.10 预发布包 .....	264	12.3 创建应用程序和模块 .....	297
10.4 发布包 .....	264	12.3.1 创建控制器、模型和 视图 .....	298
10.4.1 发布到 NuGet.org .....	265	12.3.2 服务 .....	300
10.4.2 使用 NuGet.exe .....	267	12.3.3 路由 .....	303
10.4.3 包浏览器的用法 .....	268	12.3.4 详细视图 .....	305
10.5 小结 .....	269	12.3.5 自定义电影服务 .....	306
<b>第 11 章 ASP.NET Web API</b> .....	271	12.3.6 删除电影 .....	308
11.1 定义 ASP.NET Web API .....	272	12.3.7 编辑和创建电影 .....	310
11.2 Web API 入门 .....	272	12.4 小结 .....	315
11.3 编写 API 控制器 .....	273	<b>第 13 章 依赖注入</b> .....	316
11.3.1 检查示例 ValuesController .....	273	13.1 软件设计模式 .....	316
11.3.2 异步设计: IHttpController .....	274	13.1.1 设计模式——控制 反转模式 .....	317
11.3.3 传入的操作参数 .....	277	13.1.2 设计模式——服务 定位器 .....	318
11.3.4 操作返回值、错误 和异步 .....	277		
11.4 配置 Web API .....	279		



13.1.3	设计模式——依赖注入	322	15.2	视图扩展	364
13.2	MVC 中的依赖解析	325	15.2.1	自定义视图引擎	364
13.2.1	MVC 中的单一注册服务	326	15.2.2	编写 HTML 辅助方法	366
13.2.2	MVC 中的复合注册服务	326	15.2.3	编写 Razor 辅助方法	367
13.2.3	MVC 中的任意对象	328	15.3	控制器扩展	368
13.3	Web API 中的依赖解析	330	15.3.1	操作选择	368
13.3.1	Web API 中的单一注册服务	330	15.3.2	操作过滤器	370
13.3.2	Web API 中的复合注册服务	331	15.3.3	提供自定义结果	378
13.3.3	Web API 中的任意对象	332	15.4	小结	379
13.3.4	对比 MVC 和 Web API 中的依赖解析器	332	第 16 章	高级主题	381
13.4	小结	333	16.1	移动支持	381
第 14 章	单元测试	334	16.1.1	适应性呈现	382
14.1	单元测试和测试驱动开发的意义	334	16.1.2	显示模式	386
14.1.1	单元测试的定义	335	16.2	高级 Razor	388
14.1.2	测试驱动开发的定义	336	16.2.1	模板化的 Razor 委托	388
14.2	创建单元测试项目	339	16.2.2	视图编译	389
14.2.1	检查默认单元测试	339	16.3	高级视图引擎	390
14.2.2	只测试自己编写的代码	340	16.3.1	视图引擎配置	391
14.3	单元测试用于 ASP.NET MVC 和 ASP.NET Web API 应用程序的技巧和窍门	341	16.3.2	查找视图	392
14.3.1	控制器测试	341	16.3.3	视图本身	393
14.3.2	路由测试	346	16.3.4	备用视图引擎	393
14.3.3	验证测试	349	16.3.5	新视图引擎还是新 ActionResult	395
14.4	小结	352	16.4	高级基架	395
第 15 章	扩展 ASP.NET MVC	353	16.4.1	ASP.NET 基架简介	395
15.1	模型扩展	354	16.4.2	自定义基架模板	395
15.1.1	把请求数据转换为模型	354	16.4.3	自定义基架器	398
15.1.2	用元数据描述模型	359	16.5	高级路由	398
15.1.3	验证模型	361	16.5.1	RouteMagic	398
			16.5.2	可编辑路由	399
			16.6	高级模板	402
			16.6.1	默认模板	403
			16.6.2	自定义模板	406
			16.7	高级控制器	408
			16.7.1	定义控制器： IController 接口	408
			16.7.2	ControllerBase 抽象基类	409