

PROGRAMMER TO PROGRAMMER™



Beginning ASP.NET MVC 1.0

# ASP.NET MVC 1.0

# 入门经典

Simone Chiaretta

(伊朗)

Keyvan Nayyeri

王德才 戴 君

陶小梅

著

译



清华大学出版社

# ASP.NET MVC 1.0

## 入门经典

|       |                  |   |
|-------|------------------|---|
| (意大利) | Simone Chiaretta | 著 |
| (伊朗)  | Keyvan Nayyeri   |   |
|       | 王德才 戴君           | 译 |
|       | 陶小梅              |   |

清华大学出版社

北 京

Simone Chiaretta, Keyvan Nayyeri

Beginning ASP.NET MVC 1.0

EISBN: 978-0-470-43399-7

Copyright © 2009 by Wiley Publishing, 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: 01-2009-6298

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 1.0 入门经典/(意)凯雅雷塔(Chiaretta, S.), (伊)纳耶男(Nayyeri, K.) 著;

王德才, 戴君, 陶小梅 译. —北京: 清华大学出版社, 2011. 3

书名原文: Beginning ASP.NET MVC 1.0

ISBN 978-7-302-24776-0

I. A… II. ①凯… ②纳… ③王… ④戴… ⑤陶… III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2011)第 012846 号

责任编辑: 王 军 韩宏志

装帧设计: 孔祥丰

责任校对: 胡雁翎

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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: 185×260 印 张: 32 字 数: 779 千字

版 次: 2011 年 3 月第 1 版 印 次: 2011 年 3 月第 1 次印刷

印 数: 1~3000

定 价: 68.00 元

---

产品编号: 032027-01

# 序

2007 年的一天，我应邀到 Redmond 大学参加一个会议，在途经 42 号楼的走廊时偶遇分管 .NET Developer Platform 的公司副总裁 Scott Guthrie (他当时任总经理，大家或许更熟悉他的绰号 ScottGu 或 The Gu)。他急匆匆赶去参加一个会议，但约我抽空到他的办公室小聚，因为他要向我展示一些“充满乐趣”的东西。我很了解 Scott，他肯定要透露一些非常重要的技术信息。

当然，那个有趣的东西就是 ASP.NET MVC 的设计雏形。当他开始绘声绘色地介绍这个新框架的约定和代码模式时，我唯一的冲动是“让我参与其中”。虽然当时并非招聘场合，却是我人生中经历的最美妙时刻。此后我加盟 Microsoft 并参与开发这个产品。

ASP.NET MVC 包含哪些激动人心的内容呢？在许多方面，它代表了 Microsoft 在其产品设计方法中的大量前沿技术。我们已经发布了预览版本，而且经常补充一些源代码，以求尽早予以修改。在此期间，来自社区的反馈数量空前，我们相应地大刀阔斧改进了设计。这个项目的一个重要目标就是提高透明度和社区参与程度。

ASP.NET MVC 也是 Microsoft 的首个纳入第三方开源库(即 jQuery)的产品。jQuery 是一个非常优秀的 JavaScript 库，在社区备受青睐，是 Microsoft Ajax 库的绝佳补充。

ASP.NET MVC 源代码自身的许可基于 Ms-PL 许可(已经得到 OSI 的认证)。我十分荣幸地成为一名开源社区成员，并且欣喜地目睹了开源软件结出的累累硕果。

ASP.NET MVC 的项目特点及其代码和工具都令人兴奋不已。ASP.NET MVC 密切关注可靠性(例如分离组件)，以便提供一个极易扩展和测试的框架。我们旨在确保框架是开放的，可供扩展；虽然您也可以酌情修改源代码，但通常不必那样做。

该项目最有趣之处在于生机勃勃的 ASP.NET MVC 社区的涌现。开发人员们极大的热情在感染和鼓励我们不断取得进步。本书的两位作者是 Keyvan 和 Simone 便是社区成员中的佼佼者。本书介绍有关 ASP.NET MVC 软件开发的基本知识，并指引您在此基础上构建 Web 应用程序。在阅读过程中，您将真切感受到两位作者在构建 Web 应用程序方面的激情。让我们在愉悦欢乐的气氛中度过编程时光吧！

Microsoft 公司资深程序经理 Phil Haack

# 前 言

谁也否认不了 Microsoft 在软件行业中的强大实力,并且这个巨头一直不断地扩大其在软件、硬件、网络、游戏以及娱乐等相关领域的市场份额。

Microsoft 关注的一个重要领域是软件开发技术。它试图提供能够扩展其他产品并且能够为商业方案特别是为 Microsoft 平台和操作系统(Windows、Windows Server 以及 Windows Mobile)构建解决方案的开发技术。Web 开发已经是成长迅速并且需求很大的软件开发领域,在过去 10 年是使用最频繁的开发技术之一。

Microsoft 通过推出两个产品对 Web 开发技术的庞大需求做了回应:传统的活动服务器页面(Active Server Pages, 经典 ASP)和 ASP.NET(.NET Framework 的一部分)。后者现已被开发人员广泛使用。ASP.NET 与 .NET Framework 的其他部分高度集成并且保持一致,使许多不同背景的开发人员在短时间内就可以使用它。ASP.NET 的主要优势在于提高了开发速度和简易程度,相对于其他 Web 开发技术,它允许您在更短时间内构建 Web 应用程序。

然而,软件开发是发展最迅猛的科技领域之一。新发明以及新技术的问世要求软件厂商和开发人员在其工作中必须适应最新变化。在过去几年里,由于 Ruby on Rails 取得巨大成功,软件开发领域开始重新重视模型-视图-控制器(Model-View-Controller, MVC)模式,从而迫使 Microsoft 开始考虑占领这个市场。此外,Microsoft 认识到它最初实现的 ASP.NET(现在称为 ASP.NET WebForms)在某些情况下会暴露出一些弱点。

于是,Microsoft 在 2007 年着手开发 ASP.NET WebForms 的替代方案。主要替代方案是一个称为 ASP.NET MVC 的新产品。ASP.NET MVC 在其某些基础类的实现中应用了 ASP.NET 的核心 API,并使用更注重独立性级别的 MVC 模式提供了完全不同的软件开发模型。

ASP.NET MVC 可以与 Microsoft 在软件开发领域的现代发明很好地协作,并且还和当今的 Web 开发工具,例如流行的 jQuery 库(一个 JavaScript 库,该库为 JavaScript 开发人员简化了编码模型,并提供了一些易于开发的极佳功能)很好地集成在一起。

此外,ASP.NET MVC 通过提供对 API 的低级访问以及对应用程序所有元素的完全控制(例如输出 HTML 代码),为开发人员又开启了一扇大门。ASP.NET MVC 还简化了单元测试功能,以帮助您构建更高质量的项目。

## 0.1 本书的哲学

每本书的背后都有一个故事,本书也不例外。在 Microsoft 发布 ASP.NET MVC 预览版之后不久,我们(两位作者)就开始为两个著名.NET 社区撰写一系列实用文章,这些文章在当时成为有关该产品的丰富资源。我们二人也通过.NET 社区、博客以及开源项目而相识,

并认识到可以扩展我们的文章系列，将文章提升到更高层次，即撰写一本有关 ASP.NET MVC 的书籍。当时，这个领域的相关书籍很少，并且我们发现，这些书籍都是针对高级开发人员的。

因此，通过相互协商并与 Wiley 编辑探讨之后，我们决定开始撰写一本 Wrox Press 入门经典系列的书籍。下面列出我们决定将本书确定为入门级书籍的一些主要原因：

- 首先，ASP.NET MVC 是可供 ASP.NET 开发人员从事 Web 开发的一种替代方法，所以尽可能多的独立学习过程是值得的。对于这样的学习过程，入门级书籍是有帮助的。
- 其次，ASP.NET MVC 实现了 MVC 模式，与 ASP.NET WebForms 不同，它高度依赖于一些原理和实践，这些原理和实践对于开发标准 ASP.NET MVC 应用程序是必需的。ASP.NET WebForms 应用了非常经典的拖放和向导开发形式，这种开发形式不要求开发人员深刻了解其原理。遗憾的是，我们发现许多 ASP.NET 开发人员并不了解 ASP.NET MVC 的背景知识，一本依赖于这些概念的入门级书籍是他们的理想读物。
- 最后，我们的文章和发表的博客主要通过示例来提供实用方法，所以对于我们来说，根据前面的指导原则撰写一本入门级书籍是合适的。

因此，我们强烈建议您在阅读本书其余部分之前甚至是购买本书之前，先仔细阅读这个“前言”部分。根据我们在其他书籍中吸取的经验，以及分析一些比较挑剔的读者对其他书籍的书评，我们认识到许多读者选错了书，从而对本来对目标读者大有裨益的书籍产生了不良印象。技术书籍所提供的帮助因人而异，了解书籍的知识层次、结构以及目标读者对于选择最佳读物是有帮助作用的。

## 0.2 使用 C#的原因

本书使用的编程语言主要是 C#，C#是一种.NET 编程语言，在最近几年变得非常流行。为什么我们在本书中使用 C#语言呢？

首先，在过去几年中我们看到，.NET 开发人员能够阅读和理解 Visual Basic 和 C#代码，同时使用两种语言撰写会浪费时间和纸张。

其次，C#日趋流行。.NET 社区开始接受将 C#作为.NET 的首选编程语言，并且大部分代码示例都是使用 C#提供的。

最后，我们相信迁移到 ASP.NET MVC 的大部分开发人员都拥有一些.NET 编程经验，他们能够阅读和理解使用 C#和 Visual Basic 编写的示例代码，所以没必要同时使用两种语言编写代码。

尽管我们选择 C#作为本书的主要编程语言，Visual Basic 开发人员仍然能够从本书的内容受益，而不会有任何问题，因为所有代码示例都非常简单，易于理解，所以 VB 开发人员可以很容易地阅读和理解本书的内容。



## 0.3 读者对象

本书主要面向具有核心 ASP.NET API 基础(不管这些基础知识是来自已有的 ASP.NET WebForms 知识, 还是通过专门学习而获得的), 并且正在寻求掌握 ASP.NET MVC 知识的 Web 开发人员。

本书力求为 ASP.NET MVC 领域的初学者讲解原理、概念和方法, 以帮助他们掌握 ASP.NET MVC 技巧, 从而履行实际工作中的职责。

本着这样一个目标, 我们希望读者在阅读本书之前, 具有一定的特定领域背景信息。前面几项要求是必需的, 但几项要求在某些情况下并不是必需的, 因为本书中已经提供了一些背景信息。针对硬性需求, 如果不具备足够的背景信息, 我们强烈建议在阅读本书之前先学习这些背景知识, 您可以试着阅读其他书籍来完善知识结构。

- 对于本书, 首要并且是最重要的先决条件是具备阅读并理解简单 C#代码示例的能力。不管您是一位 C#开发人员还是一位 VB 开发人员, 您应该能够阅读并理解本书中的示例代码。当然, 这些示例代码具有注释, 但是有些基本内容脚没有给出解释。本书中经常使用 LINQ 表达式以及泛型, 所以需要了解这些知识, 作为您编程知识的一部分。
- 第二项重要的要求是对 ASP.NET WebForms 的开发模型和原则有比较深入的理解(当然, 不是全部内容), 并且应当了解如何构建 ASP.NET WebForms 应用程序。本书是关于如何进行 ASP.NET MVC 开发的, 但是根据经验, 我们相信对于初学者而言, 当使用 MVC 时, ASP.NET WebForms 知识提供的一些信息是非常有帮助的; 这样读者可以比较这两门技术, 并了解一些重要区别。
- 此外, 还需要较好地掌握 ASP.NET 核心 API 背景知识, 不管是来自 WebForms 知识还是通过其他方面的学习掌握了这一背景知识。确切地讲, 需要了解有关高速缓存(caching)、身份验证(authentication)、授权(authorization)、不同的提供程序(成员资格和角色提供程序)以及核心对象(HttpContext、Server、Request 与 Response)。
- 作为 Web 开发的重要方面, 需要了解 HTML 和级联样式表(CSS), 并且为了理解本书中的代码, 还需要知道更特殊的 JavaScript。具备 AJAX、ASP.NET AJAX 以及 JQuery 方面的背景知识是有好处的。
- 在整本书中, 为了演示概念并展示理论, 我们还使用了不同的存储系统, 所以需要在一定程度上熟悉数据存储技术。至少需要了解 SQL Server、ADO.NET 数据库对象、LINQ to SQL 以及 ADO.NET Entity Framework, 对这些内容至少要有个基本了解。当然, 我们试图提供有关这些对您有帮助的技术的一些背景信息, 但是为了在职业生涯中取得成功, 您至少应当相当熟悉某些数据存储技术。
- 最后一项要求是您可能还需要了解 XML 语法以及如何在 .NET 中使用 XML API(特别是 LINQ to XML API)。

## 0.4 本书主要内容

本书涵盖了构建中等规模 ASP.NET MVC 应用程序所需了解的所有入门概念、指导原则、方法以及开发策略。

我们首先介绍了 MVC 模式核心原则的背景知识，然后讨论了这些原则在 ASP.NET MVC 中是如何实现的，并将其与 ASP.NET WebForms 进行了比较。本书主体部分讨论了 ASP.NET MVC 的主要元素，展示如何开发应用程序。我们列举简明扼要的实用例子来阐释概念。还重点描述了 ASP.NET MVC 与 ASP.NET WebForms 的差别，并且指出了它们之间的区别以帮助您进行开发。

最后两章列举两个真实的案例，这是对在本书中探讨的理论的有益补充，可以帮助您学习开发过程，并为开发实际项目做好准备。

本书的重点是概念、原则和简明实用的示例。自然，许多更高级的细节和技术扩展超出了这些核心基础的范围。当对 ASP.NET MVC 更熟练之后，您可能希望阅读清华大学出版社引进并出版的《ASP.NET MVC 1.0 高级编程》一书扩充知识。

## 0.5 本书组织结构

本书共含 19 章和 2 个附录。各章分别包含独立的主题，但在许多情况下，为了阅读后面的新章节并吸收尽量多的知识养分，需要知道之前章节中介绍的信息。大部分章节提供了练习，这些练习可以帮助您专门研究该章讨论的主题，或帮助您熟练掌握其中的部分内容。

第 1~3 章介绍 MVC 模式的一些基础信息，MVC 在 ASP.NET MVC 中的实现方式，以及 ASP.NET、ASP.NET WebForms 与 ASP.NET MVC 各种概念之间的关系。

第 4~11 章讨论 ASP.NET MVC 开发中的主要元素和方法，并介绍 ASP.NET MVC 特有的一些新主题。

第 12~17 章进一步讨论一些特定主题，并在 ASP.NET MVC 中应用一些核心 ASP.NET 功能以及 ASP.NET WebForms 功能，从而可以在 MVC 应用程序中实现一些通用的 Web 开发方案。

第 18~19 章列举两个案例研究，以便帮助您掌握 ASP.NET MVC 领域的更多实践知识，并对本书其他章节进行补充。

附录 A 列出有关 ASP.NET MVC 及相关技术领域的更多读物和电子资料。其中包括一系列图书、社区、博客以及开源项目，我们相信这些内容会对您有所帮助。

附录 B 给出各章练习题的参考答案。我们强烈建议您自己完成练习，然后将自己的方案与附录 B 提供的参考答案进行比较。



## 0.6 使用本书的要求

为了能够使用 ASP.NET MVC 以及本书中的示例，需要满足一些软件和硬件需求。

首先，需要具有能够运行 Windows 操作系统和开发环境的硬件，以及可以正常工作的 Microsoft Windows。可以使用 Windows XP、2003 Server、Vista、2008 Server 或者 Windows 7，本书使用的是 Windows Vista。

显然，需要一个构建 ASP.NET MVC 应用程序所需的开发环境。可以使用 Visual Studio 2008 的任何商业版本，也可以使用免费的 Visual Web Developer Express 2008。注意，还需要安装 Visual Studio 2008 Service Pack 1。在本书中使用的是 Visual Studio 2008 Team Suite Service Pack 1。安装开发环境时会同时安装 .NET Framework 3.5 Service Pack1，.NET Framework 3.5 Service Pack1 是构建 ASP.NET MVC 应用程序的另一项要求。注意，在本书中只有很少一部分内容需要 Service Pack 1，所以即使没有安装 Service Pack 1，仍然可以从本书的大部分内容中受益。

本书中的某些章节用到了 SQL Server 数据库，所以为了使用示例代码，需要安装 SQL Server 2005 或 2008 商业版，也可以安装 SQL Server Express 2005 或 2008。对于 SQL Server 2005，还需要安装 Service Pack 2。

除上述要求外，还需要在计算机上安装最终版本的 ASP.NET MVC Framework 1.0，这会安装所需的程序集并注册到 GAC，并为 Visual Studio IDE 添加相应项目模板。

## 0.7 源代码

在学习本书中的示例时，可以手工输入所有代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/> 或 [www.tupwk.com.cn](http://www.tupwk.com.cn) 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有源代码。

注释：

由于许多图书的标题都很类似，所以根据 ISBN 进行搜索是最简单的，本书英文版的 ISBN 是 978-0-470-43399-7。

在下载了代码后，只需用自己喜欢的解压缩软件进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

## 0.8 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，如果您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。

请通过 [wkservice@vip.163.com](mailto:wkservice@vip.163.com) 给我们发送电子邮件，我们会检查您的反馈信息，如果您的反馈是正确的，我们将在本书的后续版本中采用。

为了在网站上找到本书英文版的勘误表，可以登录到 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book 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)。

## 0.9 P2P.WROX.COM

要与作者和同行讨论，请加入 [p2p.wrox.com](http://p2p.wrox.com) 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 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，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

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](http://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) 您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。

### 注释：

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 模型-视图-控制器模式 .....                                 | 1  |
| 1.1 MVC 的发展历程 .....                                     | 1  |
| 1.2 MVC 的原理 .....                                       | 2  |
| 1.3 MVC 模式的工作流程 .....                                   | 2  |
| 1.3.1 被动视图 .....                                        | 3  |
| 1.3.2 表示模型 .....                                        | 4  |
| 1.4 模型-视图-表示器模式 .....                                   | 5  |
| 1.5 MVC 相对于传统 Web 开发模式的优势 .....                         | 7  |
| 1.5.1 单元测试 .....                                        | 7  |
| 1.5.2 测试驱动开发 .....                                      | 8  |
| 1.5.3 首先设计接口的方式 .....                                   | 8  |
| 1.6 小结 .....                                            | 8  |
| 1.7 练习 .....                                            | 9  |
| 第 2 章 ASP.NET WebForms 与 ASP.NET MVC .....              | 11 |
| 2.1 概述 .....                                            | 11 |
| 2.2 ASP.NET WebForms 与 ASP.NET MVC 的比较 .....            | 12 |
| 2.2.1 ASP.NET WebForms 存在的问题 .....                      | 12 |
| 2.2.2 ASP.NET MVC 解决 ASP.NET WebForms 问题的方式 .....       | 14 |
| 2.3 编程模型 .....                                          | 14 |
| 2.3.1 ASP.NET WebForms 编程模型 .....                       | 15 |
| 2.3.2 ASP.NET MVC 编程模型 .....                            | 15 |
| 2.4 ASP.NET 开发的主要元素 .....                               | 16 |
| 2.5 ASP.NET、ASP.NET WebForms 以及 ASP.NET MVC 之间的关系 ..... | 17 |
| 2.6 ASP.NET MVC 中 MVC 模式的实现 .....                       | 18 |

|                                      |    |
|--------------------------------------|----|
| 2.6.1 ASP.NET MVC 中的模型 .....         | 19 |
| 2.6.2 ASP.NET MVC 中的控制器 .....        | 19 |
| 2.6.3 ASP.NET MVC 中的视图 .....         | 20 |
| 2.6.4 ASP.NET MVC 中的路由 .....         | 21 |
| 2.7 小结 .....                         | 22 |
| 2.8 练习 .....                         | 22 |
| 第 3 章 ASP.NET MVC 入门 .....           | 23 |
| 3.1 安装 ASP.NET MVC .....             | 23 |
| 3.2 创建 ASP.NET MVC 项目 .....          | 24 |
| 3.3 检查解决方案的结构 .....                  | 26 |
| 3.3.1 控制器 .....                      | 26 |
| 3.3.2 视图 .....                       | 27 |
| 3.3.3 模型 .....                       | 28 |
| 3.3.4 ASP.NET MVC 的应用程序类 .....       | 28 |
| 3.3.5 Web 配置文件 .....                 | 29 |
| 3.3.6 内容文件 .....                     | 30 |
| 3.3.7 脚本文件 .....                     | 30 |
| 3.4 测试项目 .....                       | 30 |
| 3.5 一个 ASP.NET MVC 示例 .....          | 32 |
| 3.6 小结 .....                         | 40 |
| 3.7 练习 .....                         | 40 |
| 第 4 章 模型 .....                       | 41 |
| 4.1 Microsoft 为 MVC 中的“M”提供的内容 ..... | 42 |
| 4.2 模型概述 .....                       | 42 |
| 4.3 用于构建模型的不同选项 .....                | 43 |
| 4.4 LINQ to SQL .....                | 43 |
| 4.5 ADO.NET 数据库对象 .....              | 47 |
| 4.6 ADO.NET Entity Framework .....   | 50 |
| 4.7 XML .....                        | 55 |
| 4.8 有关创建数据模型的重要注意事项 .....            | 58 |

|                                       |           |                                       |            |
|---------------------------------------|-----------|---------------------------------------|------------|
| 4.9 第三方组件和工具 .....                    | 58        | 6.7 创建视图的快捷方式 .....                   | 116        |
| 4.10 小结 .....                         | 59        | 6.8 小结 .....                          | 119        |
| 4.11 练习 .....                         | 59        | 6.9 练习 .....                          | 120        |
| <b>第 5 章 控制器 .....</b>                | <b>61</b> | <b>第 7 章 路由 .....</b>                 | <b>121</b> |
| 5.1 控制器的职责 .....                      | 61        | 7.1 没有路由时的状况 .....                    | 121        |
| 5.2 创建控制器 .....                       | 62        | 7.2 重要概念 .....                        | 122        |
| 5.3 定义动作 .....                        | 63        | 7.3 路由剖析 .....                        | 122        |
| 5.4 向视图传递数据 .....                     | 66        | 7.3.1 URL 属性 .....                    | 123        |
| 5.5 从请求检索数据 .....                     | 71        | 7.3.2 Defaults 属性 .....               | 123        |
| 5.5.1 动作参数 .....                      | 71        | 7.3.3 Constraints 属性 .....            | 124        |
| 5.5.2 在动作中绑定数据 .....                  | 75        | 7.3.4 DataTokens 属性 .....             | 127        |
| 5.6 动作结果的类型 .....                     | 76        | 7.3.5 RouteHandler 属性 .....           | 130        |
| 5.6.1 ViewResult .....                | 77        | 7.4 注册路由的方式 .....                     | 131        |
| 5.6.2 PartialViewResult .....         | 77        | 7.5 使用 ASP.NET MVC 更简单地<br>管理路由 ..... | 133        |
| 5.6.3 RedirectResult .....            | 78        | 7.6 调试路由 .....                        | 134        |
| 5.6.4 RedirectToRouteResult .....     | 80        | 7.7 小结 .....                          | 135        |
| 5.6.5 JsonResult .....                | 82        | 7.8 练习 .....                          | 135        |
| 5.6.6 JavaScriptResult .....          | 83        | <b>第 8 章 单元测试概念 .....</b>             | <b>137</b> |
| 5.6.7 ContentResult .....             | 83        | 8.1 单元测试的含义 .....                     | 137        |
| 5.6.8 FileContentResult .....         | 83        | 8.2 单元测试的结构 .....                     | 138        |
| 5.6.9 EmptyResult .....               | 84        | 8.2.1 测试的四个阶段 .....                   | 139        |
| 5.7 使用核心 ASP.NET 对象 .....             | 84        | 8.2.2 固定装置 .....                      | 139        |
| 5.8 小结 .....                          | 85        | 8.2.3 执行顺序 .....                      | 139        |
| 5.9 练习 .....                          | 85        | 8.2.4 用于控制测试的特性 .....                 | 140        |
| <b>第 6 章 视图 .....</b>                 | <b>87</b> | 8.2.5 结果验证 .....                      | 140        |
| 6.1 视图的职责 .....                       | 87        | 8.2.6 代码覆盖率 .....                     | 141        |
| 6.2 视图的结构 .....                       | 87        | 8.3 使用 Visual Studio 编写测试<br>程序 ..... | 142        |
| 6.3 使用 Visual Studio 创建视图 .....       | 89        | 8.3.1 Visual Studio 测试用户界面 .....      | 143        |
| 6.3.1 松散类型视图 .....                    | 89        | 8.3.2 编写第一个测试类 .....                  | 147        |
| 6.3.2 强类型视图 .....                     | 91        | 8.4 测试依赖于外部组件的类 .....                 | 150        |
| 6.4 使用 ASP.NET MVC 方式编写<br>HTML ..... | 94        | 8.4.1 使测试变得更容易的模式 .....               | 151        |
| 6.4.1 生成链接的辅助方法 .....                 | 94        | 8.4.2 使用测试替身 .....                    | 154        |
| 6.4.2 表单辅助方法 .....                    | 97        | 8.4.3 模拟对象 .....                      | 156        |
| 6.4.3 验证辅助方法 .....                    | 110       | 8.5 小结 .....                          | 159        |
| 6.5 其他内容 .....                        | 112       | 8.6 练习 .....                          | 159        |
| 6.6 编写自己的辅助方法 .....                   | 114       |                                       |            |

|                                            |                                             |
|--------------------------------------------|---------------------------------------------|
| <b>第 9 章 测试 ASP.NET MVC 应用程序</b> ..... 161 | <b>第 11 章 动作过滤器</b> .....213                |
| 9.1 为 ASP.NET MVC 创建测试项目..... 161          | 11.1 动作过滤器的含义..... 213                      |
| 9.2 不使用模拟的测试..... 162                      | 11.2 过滤器类型..... 214                         |
| 9.2.1 测试用于呈现内容的动作..... 163                 | 11.3 核心过滤器.....215                          |
| 9.2.2 测试用于重定向到其他动作的动作..... 167             | 11.3.1 HandleError 过滤器.....215              |
| 9.3 System.Web.Abstractions..... 170       | 11.3.2 Authorize 过滤器.....218                |
| 9.4 使用模拟的测试..... 171                       | 11.3.3 ValidateInput 过滤器.....219            |
| 9.5 测试路由..... 173                          | 11.3.4 ValidateAntiForgeryToken 过滤器.....219 |
| 9.6 综合讨论..... 174                          | 11.3.5 OutputCache 过滤器.....219              |
| 9.6.1 分析传统的应用程序..... 174                   | 11.4 构建自定义动作过滤器..... 220                    |
| 9.6.2 为了能够进行测试重构代码..... 175                | 11.5 控制器范围的过滤器..... 225                     |
| 9.6.3 测试应用程序..... 178                      | 11.6 过滤器执行顺序..... 226                       |
| 9.7 使用 TDD 方法的开发..... 185                  | 11.7 使用动作过滤器..... 229                       |
| 9.7.1 需求..... 185                          | 11.7.1 取消执行动作.....229                       |
| 9.7.2 圣诞节的测试..... 185                      | 11.7.2 为视图添加更多数据.....233                    |
| 9.7.3 其他日期的测试..... 187                     | 11.8 小结..... 239                            |
| 9.7.4 新年日期的测试..... 188                     | 11.9 练习..... 240                            |
| 9.7.5 重构代码..... 189                        | <b>第 12 章 AJAX</b> .....241                 |
| 9.8 小结..... 190                            | 12.1 AJAX 的含义..... 242                      |
| 9.9 练习..... 191                            | 12.2 ASP.NET AJAX..... 243                  |
| <b>第 10 章 组件</b> ..... 193                 | 12.3 jQuery..... 245                        |
| 10.1 不使用服务器控件的原因..... 193                  | 12.4 ASP.NET MVC 中的 AJAX..... 246           |
| 10.2 服务器控件..... 194                        | 12.5 ASP.NET MVC 应用程序中 AJAX 开发原则..... 249   |
| 10.2.1 简单的表单控件..... 195                    | 12.6 付诸行动: 在 ASP.NET MVC 中使用 AJAX..... 251  |
| 10.2.2 只进行呈现的组件..... 196                   | 12.7 小结..... 256                            |
| 10.2.3 绑定数据的控件..... 197                    | 12.8 练习..... 256                            |
| 10.3 使用局部视图..... 199                       | <b>第 13 章 部署</b> .....257                   |
| 10.4 编写自定义的 HtmlHelper..... 205            | 13.1 ASP.NET 部署概述..... 257                  |
| 10.5 显示动作结果..... 209                       | 13.2 ASP.NET MVC 驻留选项..... 258              |
| 10.6 使用 JavaScript 编写客户端组件..... 211        | 13.2.1 GAC 安装.....259                       |
| 10.7 何时使用何种方法..... 211                     | 13.2.2 程序集部署.....259                        |
| 10.8 小结..... 212                           | 13.3 驻留先决条件..... 259                        |
| 10.9 练习..... 212                           | 13.4 .NET 安全策略..... 260                     |
|                                            | 13.5 Web 平台安装程序..... 261                    |

|                                                  |            |
|--------------------------------------------------|------------|
| 13.6 发布 Web 应用程序 .....                           | 265        |
| 13.7 小结 .....                                    | 267        |
| <b>第 14 章 利用 ASP.NET WebForms 的功能 .....</b>      | <b>269</b> |
| 14.1 高速缓存 .....                                  | 270        |
| 14.1.1 ASP.NET 中的高速缓存 .....                      | 270        |
| 14.1.2 ASP.NET MVC 中的高速缓存 .....                  | 271        |
| 14.2 验证 .....                                    | 277        |
| 14.2.1 ASP.NET MVC 中的验证 .....                    | 277        |
| 14.2.2 使用模型状态的验证 .....                           | 278        |
| 14.3 母版页 .....                                   | 285        |
| 14.4 其他功能 .....                                  | 287        |
| 14.4.1 站点地图 .....                                | 287        |
| 14.4.2 配置文件系统 .....                              | 287        |
| 14.4.3 运行状况监测 .....                              | 287        |
| 14.4.4 服务器控件 .....                               | 288        |
| 14.5 小结 .....                                    | 288        |
| 14.6 练习 .....                                    | 288        |
| <b>第 15 章 身份验证与授权 .....</b>                      | <b>289</b> |
| 15.1 身份验证和授权的原则 .....                            | 290        |
| 15.2 ASP.NET 中的身份验证和授权 .....                     | 290        |
| 15.2.1 成员资格提供程序 .....                            | 291        |
| 15.2.2 角色提供程序 .....                              | 292        |
| 15.3 ASP.NET MVC 中的身份验证和授权 .....                 | 292        |
| 15.3.1 配置应用程序以使用身份验证和授权 .....                    | 292        |
| 15.3.2 定制成员资格和角色提供程序 .....                       | 297        |
| 15.3.3 使用 Authorize 动作过滤器限制访问资源 .....            | 297        |
| 15.3.4 ASP.NET MVC 动作中的身份验证和授权 .....             | 298        |
| 15.4 小结 .....                                    | 312        |
| 15.5 练习 .....                                    | 313        |
| <b>第 16 章 扩展 ASP.NET MVC .....</b>               | <b>315</b> |
| 16.1 可扩展性简介 .....                                | 315        |
| 16.2 可扩展性区域 .....                                | 316        |
| 16.3 路由处理程序 .....                                | 316        |
| 16.3.1 编写自定义路由处理程序 .....                         | 316        |
| 16.3.2 通用的 WebFormRouteHandler .....             | 319        |
| 16.4 控制器工厂 .....                                 | 320        |
| 16.4.1 控制器工厂剖析 .....                             | 320        |
| 16.4.2 使用 Ninject 定制控制器工厂 .....                  | 322        |
| 16.5 视图引擎 .....                                  | 326        |
| 16.5.1 编写自定义视图引擎 .....                           | 326        |
| 16.5.2 使用自定义视图引擎 .....                           | 331        |
| 16.5.3 替代视图引擎 .....                              | 331        |
| 16.6 次要扩展点 .....                                 | 349        |
| 16.6.1 ActionMethodSelectorAttribute .....       | 349        |
| 16.6.2 TempDataProvider .....                    | 350        |
| 16.6.3 自定义 ActionResult .....                    | 351        |
| 16.7 小结 .....                                    | 353        |
| 16.8 练习 .....                                    | 354        |
| <b>第 17 章 从 ASP.NET WebForms 迁移 .....</b>        | <b>355</b> |
| 17.1 迁移的原因 .....                                 | 355        |
| 17.2 迁移过程概述 .....                                | 356        |
| 17.3 ASP.NET WebForms 与 ASP.NET MVC 的基本相似点 ..... | 357        |
| 17.3.1 身份验证和授权的相似点 .....                         | 358        |
| 17.3.2 提供程序的相似点 .....                            | 358        |
| 17.3.3 HttpModule 和 HttpHandler .....            | 358        |
| 17.4 ASP.NET WebForms 与 ASP.NET MVC 的根本区别 .....  | 358        |
| 17.4.1 ViewState 的区别 .....                       | 358        |
| 17.4.2 服务器控件使用方面的区别 .....                        | 359        |
| 17.4.3 编码模型方面的区别 .....                           | 359        |
| 17.4.4 事件模型方面的区别 .....                           | 359        |
| 17.4.5 URL 使用方面的区别 .....                         | 359        |



|                                          |            |                                         |            |
|------------------------------------------|------------|-----------------------------------------|------------|
| 17.4.6 资源分离方面的区别 .....                   | 360        | 18.4.11 创建 CategoryIndex 视图 .....       | 393        |
| 17.4.7 验证方面的区别 .....                     | 360        | 18.5 WroxBlog 路由 .....                  | 393        |
| 17.4.8 AJAX 支持方面的区别 .....                | 360        | 18.6 检查 .....                           | 395        |
| 17.5 在迁移到 ASP.NET MVC 的<br>过程中替换控件 ..... | 360        | 18.7 小结 .....                           | 397        |
| 17.6 小结 .....                            | 369        | 18.8 练习 .....                           | 398        |
| 17.7 练习 .....                            | 369        |                                         |            |
| <b>第 18 章 案例研究一 .....</b>                | <b>371</b> | <b>第 19 章 案例研究二 .....</b>               | <b>399</b> |
| 18.1 WroxBlog 概述 .....                   | 372        | 19.1 用户详情 .....                         | 399        |
| 18.2 WroxBlog 的模型 .....                  | 373        | 19.2 WroxGallery 外部库 .....              | 400        |
| 18.2.1 WroxBlog 的数据库结构 .....             | 373        | 19.3 使用 TDD 方法 .....                    | 401        |
| 18.2.2 LINQ to SQL 模型 .....              | 374        | 19.4 构建 WroxGallery 模型 .....            | 402        |
| 18.2.3 服务 .....                          | 375        | 19.4.1 创建数据模型 .....                     | 402        |
| 18.3 WroxBlog 控制器 .....                  | 382        | 19.4.2 实现 WroxGallery 服务 .....          | 403        |
| 18.3.1 实现 HomeController .....           | 382        | 19.5 WroxGallery 控制器 .....              | 410        |
| 18.3.2 实现 PostController .....           | 383        | 19.5.1 Home 控制器 .....                   | 410        |
| 18.3.3 实现 CategoryController .....       | 385        | 19.5.2 WroxGallery Account<br>控制器 ..... | 413        |
| 18.4 WroxBlog 视图 .....                   | 385        | 19.5.3 构建 Admin 控制器 .....               | 416        |
| 18.4.1 站点母版页 .....                       | 386        | 19.6 配置 IoCC .....                      | 417        |
| 18.4.2 添加 CategoriesControl<br>控件 .....  | 388        | 19.7 创建 WroxGallery 视图 .....            | 418        |
| 18.4.3 添加 PostControl 控件 .....           | 388        | 19.7.1 创建母版页 .....                      | 419        |
| 18.4.4 添加 PostsControl 控件 .....          | 389        | 19.7.2 添加内容页面 .....                     | 422        |
| 18.4.5 添加 CommentControl<br>控件 .....     | 390        | 19.8 使用 AJAX 和 jQuery .....             | 424        |
| 18.4.6 添加 CommentsControl<br>控件 .....    | 390        | 19.8.1 Lightbox 插件 .....                | 424        |
| 18.4.7 添加 CommentFormControl<br>控件 ..... | 391        | 19.8.2 自动完成插件 .....                     | 424        |
| 18.4.8 创建 Home 视图 .....                  | 392        | 19.9 运行更多测试 .....                       | 425        |
| 18.4.9 添加 About 视图 .....                 | 392        | 19.10 运行中的 WroxGallery .....            | 429        |
| 18.4.10 创建 IndividualPost 视图 .....       | 392        | 19.11 小结 .....                          | 431        |
|                                          |            | 19.12 练习 .....                          | 431        |
|                                          |            | <b>附录 A 参考资料 .....</b>                  | <b>433</b> |
|                                          |            | <b>附录 B 练习参考答案 .....</b>                | <b>435</b> |

# 第 1 章

## 模型-视图-控制器模式

通过学习本书前三章的内容，读者将对 ASP.NET MVC 框架有一个基本了解。

ASP.NET MVC 框架不同于传统的 ASP.NET 编程，它结合了模型-视图-控制器(Model-View-Controller, MVC)模式。因此，在了解该框架的工作原理之前，有必要首先了解其基础模式的工作机理。接下来的章节将介绍 Microsoft 如何在其库中实现该模式，并指导您编写第一个 ASP.NET MVC Web 应用程序。

本章主要内容包括：

- MVC 模式的原理
- MVC 模式的工作方式
- MVC 模式的其他优点

### 1.1 MVC 的发展历程

尽管 MVC 模式近来才开始普及(主要得益于 Ruby on Rails 框架以及 ASP.NET MVC 框架)，但是该模式并非什么新鲜事物。MVC 最早由挪威计算机科学家 Trygve Reenskaug 于 20 世纪 70 年代提出。最初称为 Thing-Model-View-Editor 模式，旨在解决挪威造船业出现的问题。当时需要构建能方便用于当前机构并能随机构的不断发展而改变的信息系统。

当 Trygve 后来与 SmallTalk 团体共同在 Xerox PARC 工作时，将其重新命名为模型-视图-控制器模式。一年后，Xerox PARC 的其他一些研究人员实现了 MVC 的一个版本，并将其融入 Smalltalk-80 类库中，从而使 Smalltalk 成为第一门本身支持 MVC 的编程语言。

后来，其他许多 GUI 框架受到 SmallTalk 实现的启发，其中最值得一提的当属 Cocoa，Cocoa 是 Apple Mac OS X 的主要应用程序编程环境，它是基于其前身 OpenStep 和 NextStep 构建的，而 OpenStep 和 NextStep 都提倡使用 MVC 模式。

从 Web 应用程序的角度看，第一个基于 MVC 的框架是 Apache Struts，该框架用于构建 Java 企业版本的 Web 应用程序。随后，Ruby on Rails 及其他很多基于 MVC 的应用程序框架也相继涌现。

## 1.2 MVC 的原理

在基于 ASP.NET(使用 WebForms 方法)的 Web 应用程序中,有时会将数据库代码、页面设计代码和控制流代码混合在一起。在实际中,除非分离这些元素,否则一些大型应用程序将变得难以维护。

一种用于在软件应用程序中分离这些元素的方法是使用模型-视图-控制器模式。这种范例的基本原则是将应用程序分成三部分,这三部分与三层应用程序中的 3 个常见层有松散的联系。

- 模型(Model)——模型是应用程序的一部分,负责从数据库检索数据,将数据转换为对象,并在此基础上应用特定于域的细节。
- 控制器(Controller)——控制器用于协调各个操作。它负责处理和验证用户输入、调用模型、选择要呈现的视图以及将数据传送至视图以显示给用户。
- 视图(View)——视图是应用程序的前端,用于将操作结果以及检索到的数据显示给用户。

将应用程序分成多个松散联系在一起的组件具有诸多优点,将有助于管理应用程序,使其能随当今 IT 项目的快速变化(这是当今 IT 项目的典型特征)而不断演进。

虽然将其称为 MVC 模式,但是其组件在此处(并将贯穿全书)却是以模型-控制器-视图(Model-Controller-View, MCV)的顺序进行解释的,这或许令您感到迷惑。原因很简单:解释组件工作原理时,最合理的方法是从模型开始,然后是控制器,最后是视图。然而,MCV 不如 MVC 听起来那么顺耳。

在分析 MVC 模式的特有优点之前,下一节将首先介绍基于 MVC 模式的应用程序的工作流程。

## 1.3 MVC 模式的工作流程

正如您所知道的, MVC 模式的核心原理是对应用程序各部分进行严格分离。图 1-1 显示了应用于 Web 应用程序的 MVC 模式。

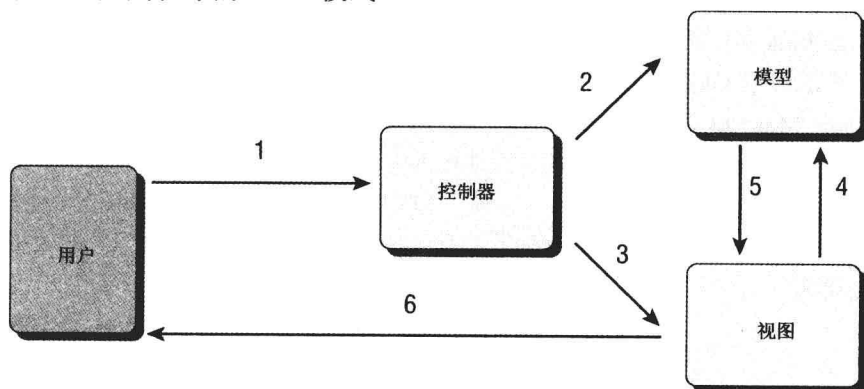


图 1-1