PROGRAMMER TO PROGRAMMER™

ASP.NET MVC 1.0 Website Programming

Problem-Design-Solution

ASP.NET MVC 网站编程案例精解

Nick Berardi
(美) Al Katawazi 著
Marco Bellinaso
颜炯 陈钢 译

WILEY

wrox™

清华大学出版社

ASP.NET MVC 网站编程
案例精解

Nick Berardi
(美) Al Katawazi 著
Marco Bellinaso

清华大学出版社
北 京

Nick Berardi, Al Katawazi, Marco Bellinaso
ASP.NET MVC 1.0 Website Programming: Problem – Design – Solution
EISBN: 978-0-470-41095-0

图书在版编目(CIP)数据

ASP.NET MVC 网站编程案例精解/(美)贝拉尔迪(Berardi, N.) 等著；颜炯，陈钢 译. —北京：清华大学出版社，2010.6
书名原文：ASP.NET MVC 1.0 Website Programming: Problem－Design－Solution
ISBN 978-7-302-22523-2

Ⅰ. A… Ⅱ. ①贝… ②颜… ③陈… Ⅲ. 主页制作－程序设计 Ⅳ. TP393.092

中国版本图书馆 CIP 数据核字(2010)第 068409 号

责任编辑：王　军　于　平
装帧设计：孔祥丰
责任校对：成凤进
责任印制：何　芊
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 刷 者：北京密云胶印厂
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：32.25　**字　数**：785 千字
版　　次：2010 年 6 月第 1 版　　**印　　次**：2010 年 6 月第1次印刷
印　　数：1～4000
定　　价：68.00 元

产品编号：032197-01

前　言

亲爱的读者，非常感谢您选择了这本书，也欢迎您开始学习最新版本的《ASP.NET MVC网站编程案例精解》，本书已完全升级到 ASP.NET MVC 1.0！写作本书的最初想法是在ASP.NET 1.0 诞生的同一年(即 2001 年)萌发的，最初的目的就是讲解如何实现一个真正的网站。本书第 1 版于 2002 年出版，很幸运，这本书获得了成功。因此，我们希望在 ASP.NET 2.0 发布之后能够将本书进行更新和升级。本书的第 2 版在 2006 年问世，受到了开发人员的欢迎，再次获得了成功。我们认为，本书之所以能够取得成功，是因为市场上绝大部分关于 ASP.NET 的书籍都是参考手册性质的，这些书籍面面俱到地介绍了这个架构中的各个控件，包括这些控件的所有方法和属性，但是这些书籍所提供的示例却常常过于简单，往往只是介绍了控件的某一项功能而已。这些参考手册性质的书籍没能说明应该如何把ASP.NET 的各种功能和控件集成在一起来构成一个功能丰富的网站，而这恰恰是读者在实际工作中最为需要的内容。真正的网站开发与简单的小例子是完全不同的，所以本书的内容真正地帮助开发人员解决日常工作中所遇到的问题。

本书的最新版本是完全重新编写的，尽可能地使用了 ASP.NET MVC 1.0 的所有功能，并在以下方面进行了改进：书中所开发的项目更加完善(例如，更新了电子商务模块和一个Web 2.0 风格的论坛模块)，也更为专业化(例如，整个网站都使用了 CSS 等当前的主流设计技术)；同时，本书在每一章内容中都尽可能详细地介绍了 ASP.NET MVC 1.0 的相关背景知识，即使是完全没有这方面经验的读者也能够顺利地阅读和理解本书的内容，在第 1 版中未能实现这一点。

首先，本书的目的是描述、设计和实现一个读者很有可能在工作中遇到的网站，同时利用各种机会介绍和解释 ASP.NET MVC 1.0 架构所提供的各种激动人心的新功能。本书不会为了简化我们的开发工作而避重就轻；相反，本书尽可能地把读者在开发网站过程中可能遇到的各种问题都解释清楚，同时提供一种或多种解决方案。

其次，我们根据问题、设计和解决方案把每一章分成若干节，然后根据模型、视图和控制器(MVC)把每节划分成若干个小节。之所以如此划分，是为了抛弃传统的解释 ASP.NET 的方法，转而使用从数据库到用户界面(database-to-user-interface)的思路来解释项目的开发，这样读者在学习一项内容的过程中就能够获得所有的相关知识，然后翻到下一页再重复这个过程。在学习 ASP.NET 的过程中，传统的方法并不适合于 MVC，我们觉得使用MVC 的思路真正理解 ASP.NET MVC 是非常重要的，而不能用传统的 ASP.NET Web Forms 的思路来理解 ASP.NET MVC。我们希望本书所介绍的 MVC 方法能够让读者更加完整地理解 MVC 的概念，从而在开发自己的 MVC 应用程序时能够按代码的功能(或关注点)把代码分成逻辑块，并从中学习到知识的内涵。这样，读者才能深刻地理解 MVC 的基本原理，而不会错误地使用 Web Forms 的思路开发 MVC 应用程序。

本书最终开发出来的网站采用了基于 CSS 的页面布局，具有完整的成员资格系统

(membership system)，还提供了能够发表和同步网站文章并支持投票和邮件列表的内容管理系统(Content Management System)。网站支持 Web 2.0 风格的论坛，还支持能够通过 PayPal 实时处理信用卡的网上商店，并提供了本地化功能(网站所实现具体功能的详细列表请参见本书第 1 章)。我们希望读者能够乐于阅读本书，并希望帮助读者在新项目开发工作中提高开发效率，使得开发的网站具有更好的健壮性、可扩展性，并具有更好的整体架构。

本书内容简介

本书实际上是一个大型的项目案例研究，这个项目案例从各项基础工作开始，通过采用各种设计方案和解决方案，一步步地添加了各种新的功能，最终实现了整个项目的完整功能。与其他 Wrox 系列书籍相比，“问题-设计-解决方案”系列书籍的独特之处就在于书籍的结构以及针对项目从始至终的完整描述。本书将会带领读者经历一个完整的 ASP.NET MVC 1.0 网站的开发历程，这个网站不仅提供了目前比较时髦的功能，还提供了电子商务网站的功能，而这正是当前大部分用户需要的功能。这些功能具体包括：

- 账户注册
- 按类别组织的新闻和事件
- 投票
- 新闻组
- 论坛
- 具有购物车和订单管理功能的网上商店
- 本地化

从网站管理人员的角度来看，本书还涵盖了以下功能和问题：

- 完善的在线后台管理功能，能够通过直观的用户界面管理各种数据
- 网站部署

在实现这些功能的过程中，我们将会学习 ASP.NET MVC 1.0 所引入的各种新功能和新概念，具体包括：

- 模型(Model)的概念
- 视图(View)
- 视图引擎(View Engine)的作用
- 视图母版页(View Master Page)、视图页面(View Page)和视图用户控件(View User Control)的使用方式
- 利用视图数据(View Data)和临时数据(Temp Data)在控制器操作和视图之间传送数据
- HTML 和 AJAX 扩展方法的使用
- 如何在 MVC 中使用 jQuery
- 控制器(Controller)
- 控制器工厂(Controller Factory)的作用
- 利用路由(Route)创建 REST 风格的 URL
- 控制器操作(Controller Action)

- 利用操作筛选器、操作结果和操作选择器来加强应用程序

我们还将使用.NET 3.5提供的以下新功能，这些新功能都可以通过使用C# 3.0进行编程：

- LINQ
- LINQ-to-SQL
- 扩展方法
- 匿名方法

此外，读者还将学到如何把这些新功能和新概念与标准的ASP.NET 2.0功能集成在一起，这些标准的ASP.NET 2.0功能包括：

- 母版页(master page)
- 成员资格(membership)和用户配置(profile)模块

本书不仅介绍了ASP.NET MVC 1.0的各种新功能，还演示了如何将这些新功能集成在一起以构成一个功能完整的网站。书中将会解释和探讨所有的设计思路(包括数据库设计、模型、视图和控制器设计以及网站的体系结构)；本书最后一部分内容将介绍多种在稳定、可伸缩、可扩展的架构基础上进行Web开发的实践技巧。

本书的组织结构

本书完整地构建了一个项目。除前两章外，本书每一章内容将介绍这个大项目中的一个独立模块，每章内容分为3节：

- **问题**：确定本章要解决的问题：本章将要做些什么？本章需要为网站添加什么功能？为什么要添加这些功能？存在哪些限制？还要考虑什么因素？
- **设计**：在明确了需要解决的问题之后，接下来我们必须知道需要使用哪些功能来解决这个问题。这样就能够大体知道问题的解决方案，或在解决该问题的过程中将会涉及哪些技术。
- **解决方案**：在确定了设计思路之后，我们已经知道应该如何解决一开始所提出的问题。本节将提供解决该问题所需的所有代码和其他各种材料。与本书所提供的完整解决方案一样，每一章都会为其所针对的问题提供一个完整的解决方案。读者在这里能够获得实际动手练习的机会，并亲自创建这些代码。

第5章到第11章中的**设计**和**解决方案**两节将会进一步细分为多个小节：

- 模型
- 控制器
- 视图

在这几章的**设计**和**解决方案**两节中，每个小节的内容都能够帮助读者进一步巩固MVC的概念。

希望读者能够从头到尾按顺序阅读本书，这样读者就能够从一无所有开始，最终完成并部署一个随时可以联机运行的网站。当然，由于本书模块化的结构特点，每一章内容都独立地实现了一个功能模块，所以如果有需要的话，读者完全可以将本书中实现的具体模块单独拿出来，应用到其他网站中。

本书面向的读者

首先要说明本书并不适合没有任何经验的新程序员，也不适合从未接触过 ASP.NET 和.NET Framework 的程序员。本书讲述的是如何从基础开始编写一个真正的网站，所以不可能事无巨细地介绍各种技术细节，只能把重点放在设计和实际解决方案的实现上。如果读者希望顺利地阅读本书，那么应该具备一些 ASP.NET 2.0 方面的基础知识，当然，只要具备基础知识就可以。我们并不要求读者接触过 ASP.NET MVC，因为每一章内容都会介绍与此有关的概念和功能，这样就可以为具体实现和解决方案提供足够丰富的背景信息。如果读者希望进一步学习涉及某个功能的全部知识内容，那么可以参考 MSDN 官方文档，也可以参考其他参考手册类型的书籍，例如 Wrox 出版的《ASP.NET MVC 1.0 高级编程》一书。

阅读本书过程中所需要的软件环境

如果读者打算在自己的计算机上构建本书所介绍的项目，或者运行下载的代码，那么就需要以下软件环境的支持：

- Windows XP、Windows Server 2003、Windows Vista、Windows 7 或 Windows Server 2008 操作系统。
- 任何一个 C#语言版本的 Visual Studio 2008，包括可以免费获得的 Visual Web Developer 2008 Express。本书中的图片来自 Visual Studio 2008 Professional。

  `http://www.microsoft.com/express/vwd/`

- 任何一个版本的 SQL Server 2008，包括可以免费获得的 SQL Server 2008 Express。本书中的项目使用的是 SQL Server 2008 Express，但是实际上可以使用各个版本的 SQL Server 2008。

  `http://www.microsoft.com/express/sql/download/`

- 任何一个版本的 SQL Server Management Studio 2008，包括与 SQL Server 2008 Express with Tools 版本一同发行的 Basic 版本，这个版本可以免费获得。本书中的图片来自 SQL Server Management Studio 2008。

  `http://www.microsoft.com/express/sql/download/`

配套网站

在本书的配套网站中，读者可以找到最新的代码及有关问题的跟踪进展，还可以就本书中的 TheBeerHouse 应用程序展开讨论。本书配套网站的网址是：

`http://thebeerhouse.codeplex.com`

读者可以以访客的身份浏览我们的维基、源代码和各种讨论，也可以参与到 TheBeerHouse 社区中与其他成员交流。我们时刻欢迎新成员的到来，也鼓励那些在实际工作中使用 TheBeerHouse 应用程序的成员与社区中的其他成员分享他们的知识。

除了前面提到的配套网站，本书还提供了 TheBeerHouse 的演示网站。这个演示网站的网址是：

```
http://www.TheBeerHouseExample.com
```

这个网站与本书所构建的网站是完全一样的，在使用本书提供的代码时，读者可以将这个网站作为参考。

源代码

读者在阅读本书提供的代码时，既可以亲自输入所有代码，也可以使用随书提供的代码文件。本书所有代码均可以从 http://www.wrox.com/或 www.tupwk.com.cn/downpage 网站下载。进入该网站后，读者可以根据本书的书名查找本书(既可以使用搜索框，也可以使用书名列表进行查找)，然后单击本书详细内容页面上提供的 Download Code 链接，就可以下载本书提供的所有代码。

注意：

*由于许多书籍名称与本书类似，读者也可以通过 ISBN 进行查找，本书的 ISBN 为：*978-0-470-41095-0。

另外，读者可以从前面提到的 CodePlex 网站下载本书或其他 Wrox 书籍的代码，也可以从 Wrox 的代码下载页面 http://www.wrox.com/dynamic/books/download.aspx 和 http://www.tupwk.com.cn/downpage 下载本书或其他 Wrox 书籍的代码。

源代码下载成功后，读者用任一解压工具将其解压即可。

勘误表

为了避免本书文字和代码中存在错误，我们已经竭尽全力。然而，世界上并不存在完美无缺的事物，所以本书可能仍然存在错误。如果读者在我们编写的某本书籍中发现了诸如拼写错误或代码缺陷等问题，那么请告诉我们，我们对此表示感谢。利用勘误表反馈错误信息，可以为其他读者节省大量时间，同时，我们也能够受益于读者的帮助，这样有助于我们编写出质量更高的专业著作。

如果读者需要参考本书的勘误表，请在网站 http://www.wrox.com 中用搜索框或书名列表查找本书书名。然后，在本书的详细内容页面上，单击 Book Errata 链接。在随后显示的页面中，读者可以看到与本书相关的所有勘误信息，这些信息是由读者提交、并由 Wrox 的编辑们加上的。通过访问 http://www.wrox.com/misc-pages/booklist.shtml，读者还可以看到 Wrox 出版的所有书籍的勘误表。

如果读者没有在 Book Errata 页面上找到自己发现的错误，那么请转到页面 http://www.wrox.com/contact/techsupport.shtml，针对您所发现的每一项错误填写表格，并将表格发给我们，我们将对表格内容进行认真审查，如果确实是我们书中的错误，我们将在该书的 Book Errata 页面上标明该错误信息，并在该书的后续版本中改正。

关于 p2p.wrox.com 网站

如果读者希望能够与作者进行讨论，或希望能够参与到读者的共同讨论中，那么请加入 p2p.wrox.com 论坛。该论坛是一个基于 Web 的系统，读者可以在论坛发表与 Wrox 出版的书籍及相关技术的信息，并与其他读者和技术用户进行讨论。论坛提供了订阅功能，可以将与读者所选定主题相关的新帖子定期发送到读者的电子邮箱。Wrox 的作者、编辑、业界专家，以及其他读者都会参与论坛中的讨论。

读者可以在 http://p2p.wrox.com 参与多个论坛的讨论，这些论坛不仅能够帮助读者更好地理解本书，还有助于读者更好地开发应用程序。如果读者希望加入论坛，那么请按照以下步骤执行：

1. 进入 http://p2p.wrox.com 页面，单击 Register 链接。
2. 阅读使用条款，然后单击 Agree。
3. 填写必要的信息及可选信息，然后单击 Submit。
4. 随后读者会收到一封电子邮件，邮件中说明了如何验证账户并完成整个加入过程。

读者无须加入 P2P 论坛即可阅读论坛消息，但如果需要发表主题或发表回复，那么必须加入论坛。

成功加入论坛后，读者就可以发表新主题了。此时，读者还可以回复其他用户发表的主题。读者在任何时间都可以阅读论坛信息，如果需要论坛将新的信息发送到自己的电子邮箱，那么可以单击论坛列表中论坛名称旁的 Subscribe to this Forum 图标完成这项功能设置。

如果读者需要获得更多与 Wrox P2P 相关的信息，请阅读 P2P FAQs，这样可以获得大量与 P2P 和 Wrox 出版的书籍相关的具体信息。阅读 FAQs 时，请单击 P2P 页面上的 FAQs 链接。

目　录

第 1 章

TheBeerHouse 项目简介

本章简要介绍本书将要开发的项目。本章中将解释这个示例网站中所涉及的各种概念，这也是本书将要讲述的主题，但是读者在阅读本书的过程中要记住，这个网站是一个通用的、数据驱动的、基于内容的网站，只要对这个网站稍作修改，就能满足实际工作中对各种网站的需求。尽管我们还要用到很多 ASP.NET 的原有功能，但是本书的重点是介绍 ASP.NET MVC 在开发一个真正的大型网站过程中使用的各种强大的新功能。

本书的每一章都遵循“问题-设计-解决方案”的思路进行讲述。在“问题”一节中，将介绍这一章所设计模块的业务需求；在“设计”一节中将开发满足这些需求的路线图；在“解决方案”一节中则要使用代码来实现设计。本书与传统的计算机书籍有些不同，因为本书所关注的并不是讲解基本概念，而是演示如何应用各种知识满足实际的业务需求。这本书并不适合作为 ASP.NET 新手的入门教程；但是如果用户已经对 Web 开发和 ASP.NET(任何版本的 ASP.NET 都可以)有了初步的了解，并且已经准备将这些知识应用到实际开发中，或者希望学习 ASP.NET MVC 提供的新功能，那么就请开始本书的学习旅程吧！

1.1 问题

多年前笔者就读于宾夕法尼亚州立大学(The Pennsylvania State University)，该学校所在城市里超过一半的人口都是学生。因为年轻人如此之多，所以城市里自然就出现了许多大小酒吧以及各种能让年轻人和朋友一起度过夜晚和周末的公共场所。音乐会、聚会、走秀和其他的娱乐活动也很常见。但是各个酒吧之间激烈的竞争使得每个酒吧都需要找到自己与众不同的特色来吸引潜在的客户。营销在其中扮演了重要角色，各个酒吧的老板都想让自己的酒吧在这个地区脱颖而出。例如，TheBeerHouse 酒吧老板一直以来都是使用印刷出来的传统海报为其酒吧(TheBeerHouse 的名称纯属虚构)做宣传，但是现在老板想尝试一下新媒体，为此首先就必须拥有一个自己的网站。他觉得这应该会产生良好的效果，因为只要客户熟悉了这个网站，他们就能从网站上获知新的促销信息和活动，可能还会通过自己的电子邮箱订阅网站的新闻邮件。客户还可以浏览以前举办活动的照片，给照片评分，与网站的其他访问者分享消息，访问者之间也可以建立虚拟的关系并将其延续到酒吧中，最终可以进行面对面的交流！这个想法颇为诱人，特别是酒吧的目标客户都是经常使用计

算机并上网查找各种信息的年轻人，因此与传统的餐厅相比，这种项目更适合充满娱乐气氛的酒吧。当然，即使是传统的餐厅也可以考虑建设一个这种类型的网站。

1.2 设计

本书每一章都将在“设计”一节中对本章所面对的问题进行探讨，从而设计出一种解决方案。所谓的设计，通常就是写下一个业务需求列表和需要实现的功能，并为数据存储设计好必要的数据库对象，以及用于获取、操作和向用户展示数据的类结构。在项目的开始阶段，首先要考虑客户的需求，以及如何才能满足他们的需求，如果时间和预算允许，那么还可以为客户提供一些额外的功能。在“问题”一节中已经说过，这个项目的客户是一个酒吧老板，他希望通过网站向客户提供活动信息，记录以往活动的情况，以此来推广和宣传自己的酒吧。可以采用很多方法来扩展这个初始的想法，向这个网站中添加更多有趣的内容，为酒吧的潜在客户提供便利，也给酒吧老板带来利润。首先我们要写下一个基于内容的网站(这个概念现在相当时髦)所需的功能清单，还要列出这些功能如此有用的一些原因：

- **富有吸引力的用户界面**。外观非常重要，因为这是用户第一眼所看到的事物，而且是在用户体验到网站的功能和服务之前所看到的事物。但是用户界面(UI)并不仅仅是图形。网站的信息必须有序地组织在一起，能够方便用户访问。网站必须方便使用，提供良好甚至是优秀的用户体验。也就是说，要让用户方便地浏览网站中的信息并与网站交互。此外还要注意网站的跨浏览器兼容性，也就是要确保网站在不同的平台和不同的浏览器上都具有一致的外观和行为。对于本书所开发的这类网站来说，一定要注意这一点。一般来说，在公司内部使用的企业内部网站中，我们可以预计到访问者可能使用的浏览器；而对于本书所开发的这类网站来说，我们根本无法预计到访问者会使用什么样的浏览器。
- **个性化的用户体验**。成功的内容网站必定是受到用户的欢迎。忠实的用户会定期地访问网站，为网站提供内容，并积极地参与投票和活动，只有这样的忠实用户才能保证网站的持续发展。要创建一个充满活力的社区，首先必须要帮助用户拥有自己的身份，并通过用户身份来区分不同的用户。所以，网站的身份验证和授权基础设施首先要提供注册功能。通过这个基础设施，我们还能控制用户对网站不同区域的访问权限。
- **动态内容**。网站的内容必须及时更新才能确保网站的活力和吸引力。如果网站的内容总是一成不变，访问者肯定会对网站失去兴趣。酒吧的网站必须及时地更新各种活动、聚会和音乐会的信息才能吸引到用户。如果网站上连最近聚会拍摄的照片都没有，那还有什么意思呢？为了使网站的内容经常得到更新，网站需要具备某种便于编辑以更新动态内容的机制。此外，考虑到负责网站内容更新的编辑人员通常都不会是专门的技术人员，所以有必要尽可能地简化网站管理页面，使得非技术人员也能方便地更新网站内容。
- **网站与用户的沟通**。当网站的内容更新后，网站管理员必须能够通过某些渠道来通知用户。并不是所有的用户每天都浏览网站，所以网站管理员必须主动地把网

站最近的更新通知给客户。如果客户在注册的时候填写了电子邮件地址，就可以把网站最近的更新通过邮件列表发送给他们。当然，还有其他发送更新信息的方法，例如把新闻放到 RSS 源中，客户可以在自己的 RSS 阅读器中订阅新闻，这样客户无须访问网站就能自动获得最新内容的通知。

- **用户与网站之间的交流**。这样的网站也是从客户那里获得各种反馈的好渠道：酒吧最近什么商品销售最好？什么牌子的啤酒最受欢迎？喜欢在酒吧一边与朋友喝酒一边听现场演奏的音乐吗？还是不喜欢这些噪声？构建用户与网站沟通的渠道很重要，如果能获得足够数量的反馈，就能据此做出战略性的决策，进而改进酒吧的业务。
- **用户与用户之间的交流**。如果说用户和网站之间的沟通渠道非常重要，那么用户之间的交流就更为重要，因为只有这样才能构建一个拥有忠实用户的社区。这些忠实用户会经常访问网站，参与聊天，讨论网站上的新闻，相互交流最新的活动信息等。这样会为网站带来更多的流量，用户的这种归属感对网站的短期效益和长期发展都是有好处的。
- **网上商店**。如果实体酒吧拥有良好的客户基础，酒吧老板可能会决定把这种客户基础扩展到网上商店。实际上，酒吧已经为啤酒爱好者提供了诸如眼镜、T 恤、钥匙链之类的产品目录。如果网站的访问量很可观，那么通过网站宣传这些商品，客户就可以在无须亲自访问酒吧的情况下订购这些商品。当客户看到喜欢的商品时，还可以在网站上为他喜欢的商品进行评分，并与其他客户分享他对该商品的评价。在线商店必须能够方便非技术人员进行管理，因为负责添加和编辑商品、管理订单的人员很有可能就是酒吧老板，所以这个模块应该提供简单直观的用户界面，尽可能地自动完成各种操作并引导客户完成下订单的过程。
- **本地化内容**。前面已经提到，酒吧的顾客常常来自不同的国家，酒吧老板自然希望自己网站的访问者也会来自不同的国家。因此，网站的某些部分(甚至是整个网站)都应该翻译成多国语言，使绝大部分用户都能理解网站的内容。在本地化的过程中，不仅仅是网站中的文本，日期、数字之类的信息也应该按用户首选的区域设置进行显示，这样用户才不会对即将到来的活动或聚会的预告产生误解。

总而言之，TheBeerHouse 网站将是一个完整而时髦的基于内容的网站，它提供了动态的文章和新闻、用于帮助用户与网站进行交流的投票系统、用于用户之间交流的论坛、用于通知网站成员当前网站新内容的邮件列表和 RSS 源，还提供了销售各种产品的网上商店，以及个性化的主页和本地化的内容。尽管这个网站是为一个虚构的酒吧而创建的，但是读者将会发现，这些需求实际上就是现在网络上大多数基于内容和基于商务的网站的主要功能，所以有可能是读者不久之后就要开发的网站的主要功能，甚至有可能就是读者正在开发的网站的主要功能。

1.3　解决方案

每一章的“解决方案”一节中将会给出具体的指令和代码，实现前两节中所规划和

设计的需求与功能。但是第 1 章是例外情况，在此将对后面每一章的主要内容做一个比较详细而精确的描述，这样读者就能对最终开发出来的网站建立一个良好的概念。

在第 2 章中，读者将会学到 MVC 的基本概念，还将学习 Microsoft 的 MVC 实现：ASP.NET MVC。在开始使用 ASP.NET MVC 编写代码之前，读者将了解到 ASP.NET MVC 应用程序的结构，模型、控制器、操作、筛选器、路由和视图等术语的含义，以及这些内容之间的关系，从而基本理解该架构。

第 3 章将会通过母版页构建整个网站的设计和所有网页共享的图形和布局。利用 CSS，可以方便地维护和定制这个灵活的设计。

第 4 章将介绍构建一个灵活的、易配置的、可检测的网站的基础知识。本章首先使用.NET 3.5 提供的新型语言集成查询(Language INtegrated Query，LINQ)设计一个模型，也就是数据访问层(Data Access Layer，DAL)。在该模型的基础上再构建控制器，也就是业务逻辑层。业务逻辑层负责必要的逻辑验证、事务管理、事件记录和必要的缓存。本章最后介绍了作为网站表示层的视图，即用户界面，该视图用于构建复杂的、功能丰富的、数据驱动的页面。

第 5 章将把 ASP.NET 的成员资格基础设施(membership infrastructure)集成到网站中，创建用户注册表单和用于用户验证和授权的支持逻辑。还可以使用 Profile 模块，以声明方式定义自动保存到持久性存储介质中的用户级特性。这不同于常见的传统会话状态变量，会话状态变量只有在用户访问网站时才是有效的。我们将构建一个完整的管理控制台，管理员在这个管理控制台中可以看到成员列表，屏蔽在网站上有不良行为的用户，还可以查看和编辑每个用户的配置文件。

第 6 章将会建立一个内容管理系统，利用该系统模块，管理员能够在直观的用户界面中全面管理网站的文章、新闻和博客帖子，即使非技术人员也能够完成这些操作。除了常见的功能之外，该模块还会与 ASP.NET 内置的成员资格系统集成在一起，从而能够确保模块的安全性，还能够跟踪发表文章的用户，并且能够提供聚合服务(syndication service)，通过聚合服务发布某一分类内容或是全部分类内容的最新新闻的聚合源(syndication feed)，还支持评分和评论。该模块的功能很强大，编辑人员可以预先准备好格式丰富的内容，然后按照预定的时间自动发布或撤消这些内容，这样，编辑人员只需要付出很少的精力和时间就能方便地更新网站内容。在这一章的末尾，读者将会深入地理解如何创建 MVC 应用程序，如何将各个部分整合在一起以构成一个功能丰富的网站。

第 7 章将为这个网站实现一个能够创建和管理多个投票的投票系统。管理员可以通过 Web 浏览器访问管理控制台来管理投票，修改各个页面上所显示的投票，还可以使用一个历史页面查看已归档的投票结果。

第 8 章将进一步丰富这个网站的内容，为这个网站添加一个能够向在配置页面中注册为可接收新闻通讯的成员发送新闻通讯的模块。该模块可以利用后台线程向用户发送电子邮件新闻通讯，而不是使用处理页面请求的主线程，这样就不会带来页面超时的风险，更重要的是编辑人员无须在发送电子邮件的过程中面对一个空白页面无聊地等上几分钟。利用异步 JavaScript 和 XML 编程技术(Asynchronous JavaScript and XML Programming，AJAX)，可以实现页面局部刷新，实时地显示在后台发送新闻通讯反馈的情况。最后，在归档页面上，最终用户还可以看到以前发送的新闻通讯。

第 9 章将会从无到有一步步地开发出一个论坛系统，该论坛系统支持提供了审查选项

的多个子论坛，论坛可以根据不同的排序方法，以自定义分页方式显示话题和回复，还有类似于 Digg、Reddit 和 Stackoverflow 的众包(crowd sourcing)机制以及其他最新的论坛类软件所拥有的典型功能。论坛系统同时还提供了完整的管理功能(删除、编辑、批准、关闭帖子和话题)。

在第 10 章中将会为这个项目添加具有完整功能的电子商务商店，包括完整的商品目录和订单管理系统、持久化的购物车、通过信用卡的集成在线支付功能、产品评分、产品可用库存、带有文本和图像的丰富格式产品描述、可配置的发运方法和订单状态等。我们只需要添加少数几个网页就能够实现这些功能，因为可以利用前面几章提供的内容作为基础，当然还会用到 ASP.NET 内置的成员资格和用户配置系统，以及 ASP.NET MVC 提供的其他新功能。

第 11 章将为网站添加完全本地化为另一种语言的主页，还会按照用户选择的区域设置，使用适当的格式显示日期和数字。因为 ASP.NET 提供了自动资源生成以及隐式和显式的本地化表达式，并且支持强类型的和动态编译的全局资源，所以这些功能都很容易实现。

最后，第 12 章将会介绍 ASP.NET MVC 网站的多种部署方法，包括在本地 IIS 服务器上部署网站、在远程生产网站中部署网站，以及在廉价的共享主机托管服务器中部署网站。新的 ASP.NET 编译模型允许使用简单的 XCOPY 命令来部署所有的内容。本章还将介绍如何把本地的 SQL Server Express 数据库部署到具有完整功能的远程 SQL Server 2008 中，以及如何创建一个安装程序包，利用这个安装程序包可以简化应用程序分发任务，还可以自动化安装任务。

1.4　本章小结

第 1 章介绍了一个雄心勃勃的计划：我们打算开发一个具有丰富功能的基于内容的网站，这个网站将说明如何利用 ASP.NET MVC 的完整功能。本章广泛介绍了本书随后将通篇讨论、设计和实现的项目。在接下来的每一章中，读者将会学习 ASP.NET MVC 的一部分新功能。在本书结尾处，读者将会完成一个真正意义上的全功能网站，该网站是一个时髦的、以内容为中心的网站，并且提供了电子商务功能。而且，本书所开发的网站有可能是读者开发过的功能最为丰富的网站，并且与通常预期的开发工作量相比，只需要付出很少的开发工作量，就能够在很短的时间内完成该网站的开发工作。Microsoft 曾经表示，发布 ASP.NET MVC 的最重要的目标之一就是简化开发人员的工作，更好更灵活地控制从服务器发送给用户浏览器的 HTML 内容。为了实现这个目标，首先就要将我们关注的内容划分为模型、视图和控制器，减少实现常用功能所需的工作量。这样，开发人员就能有更充足的时间关注业务需求，为用户和网站管理员提供更为高级的功能，同时还保持网站的可维护性、可测试性和可伸缩性。本书将会帮助读者来评判 Microsoft 是否实现了这个目标。接下来就让我们踏上本书的学习旅程！