

微软（中国）有限公司 开发工具及平台事业部 高级产品市场经理 胡德民 真心推荐

一本看得懂、用得上的开发指南



ASP.NET MVC 4 开发指南

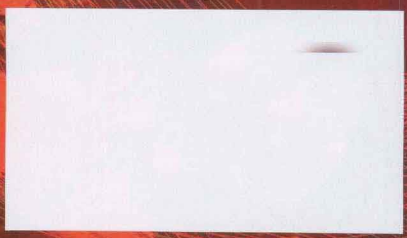


黄保翕 编著

- 着重加强观念，落实关注点
- 强化网站安全性
- Entity Framework代码先行 (Code First)开发技术
- 生命周期、路由、模型、控制器、视图完整解析
- 通过完整的电子商务演示进行教学，立即展开实际项目
- 多种ASP.NET MVC开发技巧
- Visual Studio 2012安装部署

清华大学出版社

013052919



ASP.NET MVC 4 开发指南

黄保翁 编著



清华大学出版社



内 容 简 介

本书是由台湾知名博客站长与畅销书作者“Will 保哥”所撰写的 ASP.NET MVC 4 开发指南，融入作者多年实务开发工作之精髓，从基础观念、技术讲解、开发实例、开发技巧到安装部署，都将以深入浅出的例子带领读者理解 ASP.NET MVC 框架的精妙之处，让网站生成工作更加安全、快速，也更容易维护！

本书共分 11 章，第 1 章讲解通用的 MVC 特性，第 2 章讲述正确的开发观念，第 3 章以一个简单的例子带领读者快速上手 ASP.NET MVC，第 4~8 章分别介绍了 ASP.NET MVC 最重要的核心观念与技术解析，其中包括路由与生命周期、模型、控制器、Razor 视图引擎技术、本地技术等，第 9 章则以更高级的方式完成一套完整的电子商务网站开发实例，第 10 章分享笔者多年累积的各种开发技巧，第 11 章详述如何部署 ASP.NET MVC 到正式环境。

本书适合有 ASP.NET 窗体开发经验的开发人员阅读，书中的演示能清楚带领读者快速了解 ASP.NET MVC 的开发细节与观念，相信能为读者带来全新的视野，轻松排查网站开发过程中衍生的各种难题。

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。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

ASP.NET MVC 4 开发指南 / 黄保翕 编著. -- 北京：清华大学出版社，2013.7

ISBN 978-7-302-32429-4

I. ①A… II. ①黄… III. ①网页制作工具—程序设计—指南 IV. ①TP393.092-62

中国版本图书馆 CIP 数据核字(2013)第 105251 号

责任编辑：栾大成
装帧设计：杨玉芳
责任校对：徐俊伟
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

质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 刷 者：北京鑫丰华彩印有限公司

装 订 者：三河市溧源装订厂

经 销：全国新华书店

开 本：188mm×260mm 印 张：29 插 页：1 字 数：623 千字

版 次：2013 年 7 月第 1 版 印 次：2013 年 7 月第 1 次印刷

印 数：1~3500

定 价：59.00 元

作者序

大家好，我是 Will 保哥，当前任职于台湾多奇数位创意有限公司，担任技术总监一职，负责公司技术方向的决策。大约从 4 年多前我开始注意到，ASP.NET MVC 这门技术在国外渐渐发展起来，起初不觉得有什么特别的地方，但慢慢地看到了一些国外讨论的开发观念与设计样式，好像真的能够解决当时 ASP.NET Web Form 开发上的困扰点，例如，内建的控件不好用、控件套版不易、抽象的 ASP.NET Web Form 事件模型常常让开发人员搞不清楚状况、太大的 ViewState 等，这些都是我们日常开发工作会遇到

的问题。

直到 2009 年初，ASP.NET MVC 1.0 正式版上市，除了还在创建与维护中的 ASP.NET Web Form 项目不会改动之外，我毅然决然地带领公司所有开发人员转向 ASP.NET MVC 的怀抱，并宣告未来所有新项目皆以 ASP.NET MVC 技术为主。这是一个重大的决定，也是一个至今从未后悔的决定。对我来说，学习一门全新的技术时，动机”很重要，而且当你接触一门新技术必须要放弃许多以往的开发经验时，如果没有明确的动机与察觉转变的价值，很难让人改变，因此要进入一个全新技术的领域，首要任务就是先感受技术的价值，然后再进一步深入研究技术的细节。

这本书是《ASP.NET MVC 2 开发实战》这本书的火力加强版，两年多前写人生第一本书时，从决定要写，到写完历时一年多的时间，并且当时 ASP.NET MVC 版本也从 ASP.NET MVC 1.0 升级到 ASP.NET MVC 2.0，由于 ASP.NET MVC 的观念与架构十分优秀与严谨，因此技术升级的过程非常顺利，也因为 ASP.NET MVC 的开发观念不需要重新创建，所以当时在学习 ASP.NET MVC 2.0 的时候也没什么太多负担。

时至今日，ASP.NET MVC 又升级了两版，来到当前最新的 ASP.NET MVC 4.0 版，如同我两年前的预测一样，新的 ASP.NET MVC 3.0 与 ASP.NET MVC 4.0 与早期 ASP.NET MVC 1.0 与 ASP.NET MVC 2.0 的兼容性都非常好，其开发观念也都完全一致，对于早期投入 ASP.NET MVC 研究的人来说，并不会带来多大学习压力，只要另外学习新版带来的新功能与新特性即可。

新版的 ASP.NET MVC 3.0 与 ASP.NET MVC 4.0 也如大家所预期的，微软持续不断地在 ASP.NET MVC 架构上增强功能，并且不断地强化 Visual Studio 工具的支持，例如，当前最新版的 Visual Studio 2012 的 Page Inspector 功能，就能帮助 ASP.NET MVC 开发

人员在不同的 View 之间快速切换，开发除错的效率大幅提升，因此，笔者依然认为 ASP.NET MVC 在未来还是非常具有发展前景。

这些年来，我们公司已经全面转型使用 ASP.NET MVC 开发各种大大小小的项目，现在创建的网站不但更加安全、开发速度更快、项目创建完成的时间更短，以及最重要的是网站变得更容易维护，质量也更高了，这都要归功于 ASP.NET MVC 的技术架构所赐，因为 ASP.NET MVC 非常强调“关注点分离”的概念，而且从开发架构上来看就很容易让开发人员做到“关注点分离”，这个优势是 ASP.NET Web Form 完全无法匹敌的。有趣的是，当我问公司里的工程师：“你以后还想要写 ASP.NET WebForm 吗？”他给我的回答竟然是：“保哥，我回不去了。”

笔者整理这几年累积的 ASP.NET MVC 开发经验与教学经验，希望本书的出版能带给所有 ASP.NET 开发人员另一个全新且优秀的选择，就让我们一起拥抱 ASP.NET MVC 的世界吧！

黄保翁 (Will 保哥)

多奇数位创意有限公司：<http://www.miniasp.com/>

我的博客：<http://blog.miniasp.com/>

The Will Will Web-记载着 Will 在网络世界的学习心得与技术分享

Will 保哥的新浪微博：<http://www.weibo.com/cnwill>

本书源码下载：

<https://dl.dropboxusercontent.com/u/1011445/MVC4SampleCode.zip>

前言

ASP.NET MVC 问世已久，几年前或许有人会担心 ASP.NET MVC 框架是否能用在实务的项目上，也担心用在新项目上是否真的能改善开发效率与质量，但笔者这几年下来，已经累积数十个网站项目改用 ASP.NET MVC 框架来开发，不但运作得十分顺畅，而且也能让完全不懂 ASP.NET MVC 的新手迅速理解核心观念与开发架构，相信只要读者拥有正确的观念、学习关键的核心技术，很快就能活用在工作与项目上。

本书主要分成三大部分，将以循序渐进的方式，带领读者揭开 ASP.NET MVC 4 的神秘面纱，引导读者创建正确的观念，以及实际体验 ASP.NET MVC 4 的强大魅力。

TIPS

建议章节阅读顺序：

- 1 → 2 → 3 想按部就班学习 ASP.NET MVC 的人
- 1 → 3 → 2 想先感受程序，再创建观念的人
- 3 → 1 → 2 给懒得阅读文字，只想看程序代码的人

第 1 篇：基础观念篇

第 1 章：在学习 ASP.NET MVC 之前

本章将介绍 ASP.NET MVC 的基础知识，帮助你了解 ASP.NET MVC 的轮廓。由于着重于观念，如果觉得看不懂，可先直接跳过，待后续的章节读完之后，再回头阅读或许更能帮助你创建正确的观念。

第 2 章：创建正确的开发观念

主要介绍使用 ASP.NET MVC 进行网站开发时应有的正确观念。强大的工具若没有正确的观念支持，就像是给你一台马力强又省油的手排车，而你不知道离合器如何使用是一样的，也许你试了一段时间后，觉得车子还是开不快时，就提前放弃了一部好车。本章最后也整理了初学者对 ASP.NET MVC 经常会问的问题，并逐一进行回答。

第 3 章：新手上路初体验

将介绍 Visual Studio 2012 开发环境与操作技巧，并利用 Visual Studio 2012 创建你的第一个 ASP.NET MVC 专案，让读者亲身体验 ASP.NET MVC 在进行实务开发时的完整过程，相信在逐步教学的带领之下能让现有的 ASP.NET Web Form 开发人员迅速了解与 ASP.NET MVC 的开发差异，进而快速上手。

第 2 篇：技术讲解篇

第 4 章：Routing 与 ASP.NET MVC 生命周期

要想掌握 ASP.NET MVC，最重要的是要了解网址路由(Routing)与运行生命周期的重要观念。网址路由在 ASP.NET MVC 有两个目的，第一个是比对通过浏览器传来的 HTTP 要求，并映射到适当的 Controller 与 Action 进行处理；另一个目的则是决定 ASP.NET MVC 应该输出什么样的网址响应给浏览器。虽然跳过本章仍然可以成功创建 ASP.NET MVC 网站，但了解它能帮助我们理解 ASP.NET MVC 运行时的先后顺序，进而减少犯错的机会。

第 5 章：Model 相关技术

在 ASP.NET MVC 开发的过程中，通常 Model(模型)是整个项目首要开发的部分，所有需要数据访问的地方都需仰赖 Model 提供服务。本章将从最基本的 ORM 观念讲起，介绍 Visual Studio 2012 内建的 SQL Server 2012 Express LocalDB 数据库，以及学习如何使用 Entity Framework Code First 快速创建数据模型，并利用 Code First 数据库迁移功能简化数据库操作的复杂度。最后还会介绍如何手动创建检视数据模型(ViewModel)，并通过部分类别的扩充达到基本的字段验证。

第 6 章：剖析 Controller 相关技术

ASP.NET MVC 的核心就是 Controller (控制器)，负责处理浏览器传来的所有要求，并决定响应的属性，但 Controller 并不负责应如何显示属性，仅响应特定形态的属性给 ASP.NET MVC 框架，而 View 才是决定响应属性的重要角色。本章也将会应用第 2 章所提及的“关注点分离”、“以习惯替换配置”、“不要重复你自己”等观念，让你迈入 ASP.NET MVC 的殿堂，从而发现 ASP.NET MVC 的核心之美。

第 7 章：View 数据呈现相关技术

View 负责数据的呈现，所有呈现数据的逻辑都会由 View 来控制管理，不过，View 开发应该是整个 ASP.NET MVC 项目最费时间的，因为与显示逻辑相关的技术五花八门，本章将会介绍众多 ASP.NET MVC 内建的 View 开发技术，以迅速解决各种复杂的开发情境。此外，从 ASP.NET MVC 3 开始新增的 Razor 语法，是一种有别于常规 ASP.NET Web Form 的全新撰写风格，在本章也会详加阐述所有细节，让读者在最短的时间内学会这

个崭新、优异的 Razor 语法。

第 8 章：Area 区域相关技术

本章将介绍如何利用 ASP.NET MVC 的 Area(区域)机制，协助你架构较为大型的项目，让独立性高的功能独立成一个 ASP.NET MVC 子网站，以降低网站与网站之间的耦合性，也可以通过 Area 的切割让多人同时开发同一个项目时，减少互相冲突的机会。

第 3 篇：开发实战篇

第 9 章：高级实战：电子商务网站开发

本章将通过第 2 篇所学到的知识，以一套完整的电子商务网站为蓝图，详述 ASP.NET MVC 4 开发流程与分享许多实务开发技巧，从数据模型规划、控制器架构规划、创建检视页面、添加数据库功能与购物车功能，最后再介绍如何强化现有 ASP.NET MVC 项目与信息分页技巧，相信在融会贯通之后，即可有效运用于其他更复杂的项目上。

第 10 章：ASP.NET MVC 开发技巧

本章将整理一些在实务上经常使用的开发技巧。包括强化网站安全性、多国语言支持、使用 Visual Studio 程序代码模板快速开发、如何在 ASP.NET MVC 与 ASP.NET Web Form 之间传递数据、如何对 ASP.NET MVC 4 源代码进行调试、使用 Visual Studio 程序代码模板快速开发等。虽然善用工具能有效提升开发效率，但还是要记得，拥有正确的观念与扎实的技术，才是开发效率提升的不二法门。

第 11 章：安装部署

部署网站往往是一件麻烦事，因为在安装部署的过程中，经常有许多步骤要运行，对于许多不太熟悉 IIS/SQL 的新手来说，部署网站变成一件非常困难且危险的事。Visual Studio 2012 在 ASP.NET 网站部署方面提升了不少能力，有助于让你将现有网站快速且简便地发布到远程的 IIS 服务器上，而免除了许多繁杂的设置程序。此外还整理了几个部署 ASP.NET MVC 的常见问题，当遇到问题时可供读者进行参考。

目录

第 1 部分 基础观念篇

第 1 章 在学习 ASP.NET MVC

之前 3

1.1 何谓 MVC 3

1.1.1 何谓 Model 4

1.1.2 何谓 View 5

1.1.3 何谓 Controller 6

1.2 初探 MVC 架构 7

1.2.1 彼此的关联性 7

1.2.2 Controller 与 View 的关联性 7

1.2.3 View 与 Model 的关联性 8

1.2.4 Model 与 Controller 的关联性 9

1.3 彼此的独立性 9

1.3.1 Controller 与 View 之间 的独立性 9

1.3.2 View 与 Model 之间 的独立性 9

1.3.3 Model 与 Controller 之间 的独立性 10

1.4 为什么要 ASP.NET MVC 10

1.4.1 关注点分离与可维护性 10

1.4.2 开放特性与社群支持 10

1.4.3 开发工具与效率 13

1.4.4 易于测试的架构 13

1.4.5 易于分工的架构 14

1.5 总结 14

第 2 章 创建正确的开发观念 15

2.1 关注点分离 15

2.2 以习惯替换配置 16

2.2.1 Controller 17

2.2.2 View 18

2.2.3 Model 18

2.3 开发 ASP.NET MVC 项目时 的建议 19

2.4 ASP.NET MVC 常见问题 23

2.5 总结 31

第 3 章 新手上路初体验 33

3.1 认识 Visual Studio 2012 开发工具 33

3.2 介绍 NuGet 套件管理员 40

3.2.1 遭遇问题 40

3.2.2 使用方法 41

3.2.3 开启程序包管理器 控制台(Package Manager Console) 43

3.2.4 启用 NuGet 套件还原 45

3.3 创建第一个 ASP.NET MVC 专案 47

3.3.1 利用 ASP.NET MVC 4 项目模板创建项目 48

3.3.2 创建数据模型 57

3.3.3 创建控制器、动作与 检视 59

3.3.4 测试当前创建好的留言板 网页 62

3.3.5 查看数据库属性 66

3.3.6 了解自动生成的程序 代码 67

3.3.7 调整前台让用户留言 的版面 85

- 3.4 学习 MVC 的注意事项 91
 - 3.4.1 了解不同的项目类型 91
 - 3.4.2 初学者常犯的的错误 92
 - 3.4.3 小心使用 Request 与 Response 对象 93
- 3.4.4 不要在检视中撰写过多的程序逻辑 93
- 3.5 总结 93

第 2 篇 技术讲解篇

第 4 章 Routing 与 ASP.NET MVC

- 生命周期 99
- 4.1 Routing——网址路由 99
 - 4.1.1 比对通过浏览器传来的 HTTP 要求 99
 - 4.1.2 响应适当的网址给浏览器 100
 - 4.1.3 默认网址路由属性解说 101
- 4.2 HTTP 要求的 URL 如何对应网址路由 104
 - 4.2.1 网址路由演示 104
 - 4.2.2 替网址路由加上路由值的条件约束 106
- 4.3 网址路由如何在 ASP.NET MVC 中生成网址 107
- 4.4 ASP.NET MVC 的运行生命周期 113
 - 4.4.1 网址路由比对 113
 - 4.4.2 运行 Controller 与 Action 115
 - 4.4.3 运行 View 并回传结果 120
- 4.5 总结 120

第 5 章 Model 相关技术 121

- 5.1 关于 Model 的责任 121
- 5.2 开发 Model 的基本观念 122
 - 5.2.1 何谓 ORM 122
 - 5.2.2 数据库开发模式 124
- 5.3 LocalDB 介绍 125
 - 5.3.1 LocalDB 的运作方式 125
 - 5.3.2 如何连接 LocalDB 实例 128
 - 5.3.3 管理 LocalDB 自动实例 130

- 5.3.4 管理 LocalDB 具名实例 132
- 5.4 使用 Code First 创建数据模型 134
 - 5.4.1 创建数据模型 134
 - 5.4.2 创建数据上下文类 148
 - 5.4.3 设计模型之间的关联性 149
 - 5.4.4 启用延迟装入特性 154
- 5.5 使用 Code First 数据库迁移 155
 - 5.5.1 EF Code First 如何记录版本 157
 - 5.5.2 启用数据库迁移 159
 - 5.5.3 运行数据库迁移 163
 - 5.5.4 自定义数据库迁移规则 167
 - 5.5.5 自动数据库迁移 170
 - 5.5.6 如何避免数据库被自动创建或自动迁移 171
- 5.6 使用 ViewModel 数据检视模型 172
- 5.7 扩充数据模型 172
 - 5.7.1 定义数据模型的 Metadata 173
 - 5.7.2 自定义 Metadata 验证属性 178
 - 5.7.3 ASP.NET MVC 3 新增的验证属性 179
 - 5.7.4 Entity Framework 新增的验证属性 179
 - 5.7.5 .NET 4.5 新增的验证属性 180
- 5.8 总结 180

第 6 章 Controller 相关技术	181
6.1 关于 Controller 的责任	181
6.2 Controller 的类别与方法	181
6.3 Controller 的运行过程	182
6.3.1 找不到 Action 时的 处理方式	183
6.3.2 动作名称选定器	185
6.4 动作方法选定器	186
6.4.1 NonAction 属性	186
6.4.2 HTTP 动词限定属性	187
6.5 ActionResult 解说	188
6.5.1 ViewResult	189
6.5.2 PartialViewResult	193
6.5.3 EmptyResult	193
6.5.4 ContentResult	193
6.5.5 FileResult	194
6.5.6 JavaScriptResult	198
6.5.7 JsonResult	199
6.5.8 RedirectResult	201
6.5.9 RedirectToRoute	202
6.5.10 HttpStatusCodeResult	204
6.5.11 HttpNotFoundResult	205
6.5.12 HttpUnauthorizedResult	205
6.6 ViewData、ViewBag 与 TempData 概述	206
6.6.1 ViewData	206
6.6.2 ViewData.Model	207
6.6.3 ViewBag	209
6.6.4 TempData	210
6.7 模型绑定	212
6.7.1 简单模型绑定	212
6.7.2 使用 FormCollection 取得 窗体信息	214
6.7.3 复杂模型绑定	214
6.7.4 多个复杂模型绑定	217
6.7.5 判断模型绑定的验证 结果	220

6.7.6 模型绑定验证失败的错误 详细信息	222
6.7.7 清空模型绑定状态	223
6.7.8 使用 Bind 属性限制可被 更新的数据模型属性	224
6.7.9 使用 UpdateModel 与 TryUpdateModel	226
6.8 动作过滤器	228
6.8.1 授权过滤器属性	230
6.8.2 动作过滤器属性	234
6.8.3 结果过滤器属性	238
6.8.4 例外过滤器属性	239
6.8.5 自定义动作过滤器属性	242
6.9 总结	244

第 7 章 View 数据呈现相关 技术

7.1 关于 View 的责任	245
7.2 了解 Razor 语法	245
7.2.1 Razor 基本语法	246
7.2.2 Razor 与 HTML 混合 输出	247
7.2.3 Razor 与 HTML 混合输出 陷阱与技巧	251
7.2.4 Razor 与 ASPX 语法比较	255
7.2.5 Razor 的主版页面框架	258
7.2.6 @helper 辅助方法	262
7.2.7 @functions 自定义函数	266
7.2.8 @model 引用参考资料 类型	267
7.2.9 @using 引用命名空间	267
7.3 View 如何从 Action 取得 数据	268
7.3.1 使用弱型别模型取得 数据	269
7.3.2 使用强型别模型取得 数据	271
7.4 HTML 辅助方法	271

7.4.1 使用 HTML 辅助方法输出超链接	272
7.4.2 使用 HTML 辅助方法输出表单	274
7.4.3 使用 HTML 辅助方法载入分部视图	285
7.4.4 使用检视模板输出内容	290
7.4.5 自定义 HTML 辅助方法	304
7.5 Url 辅助方法	312
7.6 Ajax 辅助方法	313
7.6.1 使用 Ajax 超链接功能	315
7.6.2 使用 Ajax 表单功能	316
7.6.3 了解 AjaxOptions 型别	317
7.7 总结	318

第 8 章 Area 区域相关技术	319
8.1 何时会需要使用 Area 切割网站	319
8.2 如何在现有项目中新增区域	321
8.3 如何设置区域的网址路由	321
8.4 默认路由与区域路由的优先级	323
8.5 就算使用区域, 控制器的名称仍然会冲突	324
8.6 如何指定默认网站与区域网站的链接	325
8.7 总结	325

第 3 篇 开发实战篇

第 9 章 高级实战: 电子商务网站开发	329
9.1 需求分析	329
9.2 数据模型规划	331
9.2.1 商品类别	332
9.2.2 商品信息	333
9.2.3 会员信息	334
9.2.4 购物车项目	336
9.2.5 订单主文件	337
9.2.6 订单明细	338
9.2.7 回顾数据模型定义	340
9.3 控制器架构规划	340
9.3.1 商品浏览	341
9.3.2 会员功能	342
9.3.3 购物车功能	345
9.3.4 订单结账功能	347
9.3.5 回顾控制器架构规划	348
9.4 创建视图页面	348
9.4.1 商品浏览	349
9.4.2 会员功能	356

9.4.3 购物车功能	362
9.4.4 订单结账功能	365
9.4.5 撰写主版页面	367
9.4.6 回顾创建视图页面	368
9.5 添加数据库与购物车功能	369
9.5.1 添加信息内容类	369
9.5.2 添加导览属性	370
9.5.3 启用自动数据库迁移	372
9.5.4 商品浏览	372
9.5.5 会员功能	375
9.5.6 购物车功能	377
9.5.7 订单结账功能	379
9.5.8 回顾添加数据库与购物车功能	381
9.6 强化会员功能	382
9.6.1 修正会员注册机制	382
9.6.2 完成会员 E-mail 验证功能	386
9.6.3 修正会员登录机制	387
9.6.4 检查会员注册的账户是否重复	389

9.7 强化现有的 ASP.NET MVC 程序.....	390	10.5.3 如何在专案中自定义代码模板.....	412
9.7.1 抽离多个 Controller 重复的程序代码.....	391	10.5.4 深入 T4 代码产生器技术.....	416
9.7.2 将调试用的程序代码区分不同配置.....	392	10.6 让 Visual Studio 连同 View 一起进行编译.....	417
9.7.3 替产品列表加上分页功能.....	394	10.7 其他 Controller 开发技巧.....	420
9.8 总结.....	398	10.7.1 侦测用户端要求是否为 Ajax.....	420
第 10 章 ASP.NET MVC 开发技巧.....	399	10.7.2 限定 Action 只能通过调用.....	421
10.1 强化网站安全性：避免网站脚本攻击(XSS).....	399	10.8 总结.....	422
10.1.1 使用 Html.Encode 辅助方法.....	399	第 11 章 安装部署.....	423
10.1.2 使用 Url.Encode 辅助方法.....	400	11.1 如何部署到本机的 IIS.....	423
10.1.3 使用 Ajax.JavaScript StringEncode 辅助方法.....	400	11.1.1 安装 IIS 功能.....	423
10.1.4 使用 AntiForgeryToken 辅助方法强化表单安全性.....	401	11.1.2 “Web 一键式发布”功能的使用.....	424
10.2 在 ASP.NET MVC 与 ASP.NET Web Form 之间传递数据.....	403	11.2 如何部署到远程的 IIS.....	430
10.2.1 HTTP GET (QueryString) 或 HTTP POST.....	403	11.2.1 安装 IIS 管理服务.....	431
10.2.2 Session.....	403	11.2.2 启用 IIS 管理服务的远程连接功能.....	432
10.2.3 Cookie.....	404	11.2.3 安装 Web Deploy.....	434
10.3 ASP.NET MVC 的多国语系支持.....	404	11.2.4 启用 Web Deploy 发布.....	436
10.4 从 HTTP 响应标头隐藏 ASP.NET MVC 版本.....	409	11.2.5 设置“Web 一键式发布”.....	438
10.5 使用 Visual Studio 代码模板快速开发.....	410	11.3 如何使用命令提示符手动部署.....	440
10.5.1 如何使用代码模板快速产生 View.....	410	11.3.1 生成部署封装文档.....	441
10.5.2 修改内建的代码模板.....	411	11.3.2 手动安装部署网站.....	443
		11.4 部署 ASP.NET MVC 的常见问题.....	444
		11.4.1 部署到 IIS6 之后看不到网页.....	444
		11.4.2 部署到 IIS6 或 IIS7 之后都无法使用网站.....	447
		11.5 部署 ASP.NET 4.0 的注意事项.....	449

11.5.1	安装注意事项	450	11.5.3	应用程序池不能跨.NET 版本	451
11.5.2	安装正确的.NET Framework 套件	450	11.6	总结	451

第 1 篇 基础观念篇

在开始学习ASP.NET MVC之前，我想许多读者心中应该都有很多疑惑，尤其是正宗的ASP.NET Web Forms开发者，更是对ASP.NET MVC技术不清楚，甚至是误解。在此先将观念清楚，帮助你了解ASP.NET MVC有个清晰的轮廓与观念。

对我来说，学习一门全新的技术时，“动机(Know Why)”很重要。而且当你接触一门必须要放弃许多开发经验的新技术时，如果没有明确的动机与察觉转变的价值，则很难让一个人改变。

本篇将包括以下三个章节。

第1章：在学习ASP.NET MVC之前

本章将介绍ASP.NET MVC的基础知识，帮助你了解ASP.NET MVC的轮廓。由于着重于观念介绍，如果觉得看不懂可先直接跳过，待后续的章节读完之后，再回头阅读本章也许更能帮助你创建观念。

第2章：创建正确观念

本章将介绍利用ASP.NET MVC进行网站开发时应有的正确观念。强大的工具若没有正确的观念支持，就像是给你一台马力强又省油的自排车，而你却不知道离合器是干什么用的一样，也许你试了一段时间觉得车还是开不快时，就提前放弃了一部好车。

第3章：新手上路初体验

本章将介绍如何利用Visual Studio 2012开始一个ASP.NET MVC 4专案，让读者亲身体验ASP.NET MVC在进行实务开发时的完整过程，相信有一点程序功力的人通过这一章的教学能够得到一些启发。

建议章节阅读顺序：

- 1→2→3 (想按部就班学习 ASP.NET MVC 的人)
- 1→3→2 (想先感受程序，再创建观念的人)
- 3→1→2 (给懒得阅读文字，只想看程序代码的人)

第1章 在学习 ASP.NET MVC 之前

本章将介绍ASP.NET MVC的基础知识，帮助你了解ASP.NET MVC的轮廓。由于着重于观念介绍，如果觉得看不懂可先直接跳过，待后续的章节读完之后，再回头阅读本章也许更能帮助你创建正确的观念。

1.1 何谓 MVC

在学习ASP.NET MVC之前，需要先了解“什么是MVC？”。也许这对某些ASP/ASP.NET开发人员来说非常陌生，MVC不是一种程序语言，严格说起来也不算是个技术，而是开发时所使用的一种“架构(框架)”。它就像是一种开发观念，或是一个存在已久的设计样式(Design Pattern)。

开发人员最熟悉的，在软件开发时最常发生的状况就是“变化”。需求会变、技术会变、老板会变、客户也会变、最惨的是PM也常在变。经常改变的需求，对于软件质量与可维护性有很大的杀伤力，但这是现实，也无法跳脱。我们唯一能做的，就是有效降低变化所带来的冲击，而MVC就是其中一种解决方案。

MVC最早是在1979年由Trygve Reenskaug所提出，并且应用于当时火红的Smalltalk程序语言中。之所以会提出MVC的概念，主要的目的就在于简化软件开发的复杂度，以一种概念简单却又权责分明的架构，贯穿整个软件开发流程，通过“商业逻辑层”^①与“数据表现层”的切割，让这两部分的信息切割开来，用以撰写出更模块化、可维护性高的程序代码。

MVC让软件开发的过程大致切割成三个主要单元，分别为：Model(模型)、View(检视)、Controller(控制器)，而这三个单词的缩写便简称为MVC。其定义如图1-1所示。

^① 我们在开发各种应用程序时，通常会依据客户的需求撰写程序逻辑，而大多数应用程序都是应用在“商业环境”里，例如，电子商务或支持企业营运的窗体应用程序，等等，因此各位常听到的“商业逻辑”讲的就是这些应商业环境所撰写的程序逻辑。如果你所开发的应用程序并非商业用途(例如学校单位)，我们也可以通称这些依据需求所开发出来的逻辑为“商业逻辑”。商业逻辑层(Business Logic Layer, BLL)所包含的程序代码通常会包含信息格式定义(ORM)、信息访问程序、窗体的字段格式验证、信息保存的格式验证、数据流验证，等等。