

Broadview®
www.broadview.com.cn

dp 悦知文化
Delight Press

ASP.NET MVC 2新版抢先曝光与全面剖析



ASP.NET MVC 2 开发实战



黄保翕 (Will保哥) 著

陈酉玫 李杨 改编



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

ASP.NET MVC 2

开发实战



黄保翕 (Will保哥) 著

陈西玫 李杨 改编

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书以循序渐进的方式揭开ASP.NET MVC的神秘面纱,帮助开发人员将现有的ASP.NET Web Forms项目转移到ASP.NET MVC架构,引导读者建立正确的开发观念,并搭配Visual Studio 2010开发工具,带领读者体验ASP.NET MVC的强大魅力。

本书主要分成3篇。基础观念篇介绍了开发人员在学习ASP.NET MVC之前需要了解的技术和观念,以及MVC架构的基础知识。技术讲解篇针对ASP.NET MVC 2的各项开发技术进行解析,以帮助开发人员解决实际问题。开发实践篇通过一套完整的留言板系统案例详述ASP.NET MVC 2的开发技巧与流程,以及ASP.NET MVC的安装部署和单元测试方法。

本书适合对MVC架构感兴趣的ASP.NET开发人员阅读。

本书为精诚资讯股份有限公司授权电子工业出版社于中国大陆(台港澳除外)地区之中文简体版本。本著作物之专有出版权为精诚资讯股份有限公司所有。该专有出版权受法律保护,任何人不得侵害之。

版权贸易合同登记号 图字:01-2010-8233

图书在版编目(CIP)数据

ASP.NET MVC 2 开发实战 / 黄保翥著; 陈酉玫, 李杨改编. —北京: 电子工业出版社, 2011.3
ISBN 978-7-121-12788-5

I. ①A… II. ①黄… ②陈… ③李… III. ①主页制作—程序设计 IV. ①TP393.092

中国版本图书馆 CIP 数据核字 (2011) 第 008907 号

责任编辑: 高洪霞

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 22 字数: 518 千字

印 次: 2011 年 3 月第 1 次印刷

印 数: 4000 册 定价: 49.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

自序

笔者目前任职于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。大约在两年多前，我开始注意到，ASP.NET MVC 技术在国外渐渐发展起来。起初，我并不觉得它有什么特别的地方，但慢慢看到一些国外对 ASP.NET MVC 的开发观念与设计方法的讨论——好像它真的能够解决当时 ASP.NET Web Forms 开发上的难点，例如内置的控件不好用、控件套版不易、抽象的 ASP.NET Web Forms 事件模型经常让新手搞不清楚状况、过大的 ViewState 等，而这些都是我们在日常开发工作中会遇到的问题。

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 Forms 项目不会改动之外，我毅然决然地带领公司的所有开发人员投入 ASP.NET MVC 的怀抱，并宣布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，也是一个我至今未曾后悔的决定。对我来说，学习一门全新技术的动机很重要。当你因为接触一门新技术而必须要放弃许多以往的开发经验时，如果没有明确的动机并发现转变的价值，是很难下定决心的。因此，要进入一个全新的技术领域，首先要感受技术的价值，然后是进一步研究技术的细节。

本书的写作历时一年多的时间，在这段时间里，ASP.NET MVC 的版本也从 ASP.NET MVC 1.0 升级到了 ASP.NET MVC 2.0。由于 ASP.NET MVC 的观念与架构十分优秀、严谨，因此，技术升级的过程非常顺利，而且开发观念也不需要重新建立。对开发人员来说，即使升级至 ASP.NET MVC 2.0，在学习上也没什么负担。即便是未来的 ASP.NET MVC 3.0，微软的 ASP.NET MVC 开发团队也能保证它会与 ASP.NET MVC 1.0 兼容，早期编写的程序完全可以顺利转移。新版本的 ASP.NET MVC 只会继续在同一套架构上增强功能，并强化对 Visual Studio 2010 的支持。因此，笔者认为，ASP.NET MVC 在未来非常具有可发展性。

笔者曾经想过，我们公司会不会是台湾地区唯一一个只使用 ASP.NET MVC 技术开发网站项目的公司呢？不过，这并不是重点，重点是我们现在创建的网站不但更加安全、开发速度更快、项目生成和完成的时间更短，最重要的是，网站变得易于维护，质量也更高了，这都要归功于 ASP.NET MVC 的技术架构。因为 ASP.NET MVC 非常强调“分解关注点”的观念，而且从开发架构上来看，也很容易让开发人员做到“分解关注点”，这个优势是 ASP.NET Web Forms 完全无法匹敌的。

本书集结了笔者在学习与开发 ASP.NET MVC 时的经验与心得，希望能带给所有 .NET

开发人员一个全新且优秀的选择。让我们一起拥抱 ASP.NET MVC 的世界吧！

能够顺利出版这本书，我要感谢很多人。首先，最感谢的是我的 ASP.NET MVC 开发团队，在公司里，他们写的 Code 比我写的字还多出很多很多，本书涉及的很多开发技巧与学习心得都来自于他们。除此之外，Demo、Thomsen、Danny、Bibby 协助我撰写了部分章节，星寂、Jerry 帮我完成了许多范例程序。我还要感谢帮我校稿与验证范例程序的朋友们，如果没有他们的协助，这本书恐怕很难完成。

同时，也非常感谢悦知文化的编辑 Vicky，在我写作的过程中提供了许多的协助并给予我包容，因为我从来就没有按照“写作说明”写作文章，即便要出书，也还是觉得写博客的心情与角度比较符合我的风格。

这是我写的第一本书，如果读者对书中的内容或观念有任何不理解的地方，欢迎到我的博客、Facebook 粉丝专页或噗浪（Plurk）上与我交流、互动。

黄保翕（Will 保哥）

<http://blog.miniasp.com/>

导 读

在开始学习 ASP.NET MVC 之前，我想许多读者心中应该都有疑惑，尤其是正宗的 ASP.NET Web Forms 开发者，更可能会对 ASP.NET MVC 技术给予异样的眼光甚至误解。

本书分为 3 篇，以循序渐进的方式揭开 ASP.NET MVC 的神秘面纱，引导读者建立正确的观念，实际体验 ASP.NET MVC 的强大魅力。

TIP

建议阅读顺序：

- 第 1 篇→第 2 篇→第 3 篇：想按部就班学习 ASP.NET MVC 的人。
- 第 1 篇→第 3 篇→第 2 篇：想先感觉程序，再建立观念的人。
- 第 3 篇→第 1 篇→第 2 篇：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 的概况。由于本章内容着重于观念的讲解，如果觉得看不懂，可先跳过，待将读完后续的章节之后再回头阅读，或许更能帮助你建立正确的观念。

第 2 章 建立正确的开发观念

本章主要说明使用 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动挡汽车，而你却不知道离合器应该如何使用一样——也许当你尝试了一段时间，觉得车子还是开不快时，就会提前放弃一部好车。本章的最后也整理了一些初学者在学习 ASP.NET MVC 时经常会问的问题，并逐一进行回答。

第 3 章 新手上路初体验

本章介绍如何利用 Visual Studio 2010 创建一个 ASP.NET MVC 项目，让读者亲身体会 ASP.NET MVC 实际开发的完整过程。相信在逐步教学的带领之下，能让现有的 ASP.NET Web Forms 开发人员了解其与 ASP.NET MVC 开发的差异，进而快速上手。

第 2 篇 技术讲解篇

第 4 章 网址路由与 ASP.NET MVC 生命周期

想要掌握 ASP.NET MVC，最重要的是要了解网址路由（Routing）与 ASP.NET MVC 生命周期的重要概念。网址路由在 ASP.NET MVC 中有两个用途：一个用途是比对通过浏览器传来的 HTTP 请求，并将其对应到适当的控制器（Controller）与动作（Action）中进行处理；另一个用途是决定 ASP.NET MVC 应该输出什么样的网址来响应浏览器。虽然跳过本章你仍然可以成功地创建 ASP.NET MVC 网站，但阅读本章能帮助你理解 ASP.NET MVC 执行时的顺序，进而减少犯错的机会。

第 5 章 用 Model 创建数据模型

在用 ASP.NET MVC 开发网站的过程中，通常模型（Model）是整个项目中首先要开发的部分，所有需要进行数据访问的地方都仰赖模型提供服务。本章将讲解如何通过 Visual Studio 2010 自动建立数据模型以及手动建立和查看数据模型，并通过部分类的辅助方法实现基本的字段验证，甚至完成商业逻辑验证，最后通过实现库模式（Repository Pattern），让 ASP.NET MVC 项目更易于进行单元测试与测试驱动开发（Test-Driven Development, TDD）。

第 6 章 剖析 Controller 的相关技术

ASP.NET MVC 的核心就是 Controller，它负责处理浏览器传来的所有请求并决定响应的内容。但 Controller 并不负责决定如何显示内容，而是仅将特定类的内容响应给 ASP.NET MVC 框架。视图（View）才是决定响应内容的重要角色。本章也会应用第 2 章提及的“分解关注点”、“以习惯取代配置”、“不要重复你自己”等观念，带你迈入 ASP.NET MVC 的殿堂，发现 ASP.NET MVC 的核心之美。

第 7 章 View 的相关技术

View 负责数据的显示，所有用于数据显示的逻辑都由 View 来管控。不过，View 开发通常是整个 ASP.NET MVC 项目开发中花费时间最长的，因为与显示逻辑相关的技术五花八门，包括 HTML、CSS、JavaScript、DOM、jQuery、JSON、Ajax、Silverlight 等——当然不止这些。但你也不需要把这些技术全部学会，因为在不同的显示情境下会用到不同的技术。本章将说明如何实现“分解关注点”，让你的 ASP.NET MVC 网站更易于维护。

第 8 章 Area 的相关技术

本章将介绍如何利用 ASP.NET MVC 2.0 新建的区域 (Area) 机制构建较为大型的项目, 以及如何将独立性较高的功能切割成一个 ASP.NET MVC 子网站, 以降低网站与网站之间的耦合性。当然, 你也可以通过 Area 的切割, 在多人同时开发同一个项目时降低冲突发生的机会。

第 3 篇 开发实战篇

第 9 章 MvcGuestbook 高级实战

本章将通过第 2 篇讲解的知识强化第 3 章中编写的留言板程序。虽然案例极为简单, 但内容与开发技巧却涵盖 ASP.NET MVC 的许多精华之处。相信在融会贯通之后, 你即可将其有效运用于其他更复杂的项目中。

第 10 章 ASP.NET MVC 开发技巧

本章整理了一些在现实开发中经常使用的开发技巧, 包括多语言支持、隐藏 HTTP 标题中的版本编号、在 ASP.NET MVC 与 ASP.NET Web Forms 之间传递数据, 以及 ASP.NET MVC 的调试技巧和程序代码生成器的入门知识。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巧才是提升开发效率的不二法门。

第 11 章 安装部署

以往在部署网站时, 有许多需要手动设置的步骤, 对很多不熟悉 IIS/SQL 的新手来说, 部署网站成了一件困难的事。Visual Studio 2010 提升了 ASP.NET 网站的部署能力, 并通过 Microsoft Web Deployment Tool (MSDeploy) 简化了许多烦琐的流程。此外, 本章还整理了一些部署 ASP.NET MVC 时的常见问题。

第 12 章 单元测试

ASP.NET MVC 的一大特色在于它有比 ASP.NET Web Forms 更高的可测试性。本章将介绍单元测试的基本概念与实现方法, 并利用 ASP.NET MVC 架构的优势引导你进入测试驱动开发的神秘殿堂。

范例程序及读者互动

本书的范例程序请访问网站 <http://www.broadview.com.cn/12788> 下载。

关于本书内容的问题与建议，欢迎通过以下方式回馈与分享。

- 博客：The Will Will Web (<http://blog.miniasp.com/>)。
- Facebook 粉丝专页：Will 保哥的技术交流中心(<http://www.facebook.com/will.fans>)。
- Plurk：★★★ Will 保哥的噗浪 ★★★ (<http://www.plurk.com/willh>)。

目 录

第 1 篇 基础观念篇

第 1 章 在学习 ASP.NET MVC 之前.....	2
1.1 何谓 MVC.....	3
1.1.1 何谓 Model.....	3
1.1.2 何谓 View.....	4
1.1.3 何谓 Controller.....	5
1.2 MVC 架构初探.....	6
1.2.1 关联性.....	6
1.2.2 独立性.....	7
1.3 为什么要使用 ASP.NET MVC.....	8
1.3.1 分解关注点与可维护性.....	8
1.3.2 开放性与社区支持.....	9
1.3.3 开发工具与开发效率.....	10
1.3.4 易于测试的架构.....	10
1.3.5 易于分工的架构.....	11
总结.....	11
第 2 章 建立正确的开发观念.....	12
2.1 分解关注点.....	13
2.2 以习惯取代配置.....	13
2.3 对开发 ASP.NET MVC 项目的建议.....	14
2.4 ASP.NET MVC 中常见的问题.....	18
总结.....	25
第 3 章 新手上路初体验.....	26
3.1 认识 Visual Studio 2010.....	27

3.1.1	解决方案资源管理器窗口	27
3.1.2	服务器资源管理器窗口	27
3.1.3	工具箱窗口	27
3.1.4	输出窗口	28
3.1.5	错误列表窗口	28
3.1.6	扩充管理器	29
3.1.7	代码段管理器	29
3.2	创建第 1 个 ASP.NET MVC 项目	30
3.2.1	用 ASP.NET MVC 2 项目模板创建项目	30
3.2.2	创建数据模型	38
3.2.3	创建控制器和动作	41
3.2.4	创建视图	42
3.2.5	在视图中创建窗口	44
3.2.6	通过 Action 接收窗口数据	46
3.2.7	实现留言显示功能	49
3.3	学习 MVC 时的注意事项	53
3.3.1	了解不同的项目类型	53
3.3.2	初学者常犯的错误	53
3.3.3	小心使用 Request 对象和 Response 对象	54
3.3.4	不要在视图中编写过多的程序逻辑	54
3.4	改良 ASP.NET MVC 项目	55
3.4.1	使用视图数据模型	55
3.4.2	使用窗口验证功能	58
3.4.3	避免用户重复发送信息	65
	总结	68

第 2 篇 技术讲解篇

第 4 章	网址路由与 ASP.NET MVC 的生命周期	70
4.1	网址路由	71
4.1.1	比对通过浏览器传来的 HTTP 请求	71

4.1.2	将适当的网址返回浏览器	71
4.1.3	默认网址路由	71
4.2	HTTP 请求的 URL 如何对应网址路由	73
4.2.1	网址路由范例	73
4.2.2	为网址路由加上限制条件	75
4.3	网址路由如何在 ASP.NET MVC 中生成网址	76
4.4	ASP.NET MVC 的执行生命周期	81
4.4.1	网址路由比对	81
4.4.2	执行 Controller 和 Action	83
4.4.3	执行 View 并返回结果	86
	总结	87
第 5 章	用 Model 创建数据模型	88
5.1	Model 的任务	89
5.2	创建基础数据模型	89
5.2.1	用 LINQ to SQL 自动创建数据模型	90
5.2.2	用 Entity Framework 自动创建数据模型	93
5.2.3	手动创建数据模型	98
5.3	扩充基础数据模型	99
5.3.1	定义 Model 的 Metadata	100
5.3.2	自定义 Metadata 验证属性	104
5.4	实现库模式	104
	总结	106
第 6 章	剖析 Controller 的相关技术	107
6.1	Controller 的责任	108
6.2	Controller 的类与方法	108
6.3	Controller 的执行过程	109
6.4	动作名称选取器	111
6.5	动作方法选取器	112

6.5.1	NonAction 属性	113
6.5.2	HttpGet 属性、HttpPost 属性、HttpDelete 属性和 HttpPut 属性	113
6.6	ActionResult 类	114
6.6.1	ViewResult 类	115
6.6.2	PartialViewResult 类	117
6.6.3	EmptyResult 类	117
6.6.4	ContentResult 类	118
6.6.5	FileResult 类	119
6.6.6	强制下载文件时需注意中文文件名的问题	122
6.6.7	JavaScriptResult 类	124
6.6.8	JsonResult 类	124
6.6.9	RedirectResult 类	126
6.6.10	RedirectToRoute 类	126
6.7	ViewData 与 TempData 概述	128
6.7.1	ViewData	128
6.7.2	TempData	129
6.8	模型绑定	131
6.8.1	简单模型绑定	131
6.8.2	使用 FormCollection 类获取窗口数据	132
6.8.3	复杂模型绑定	132
6.8.4	多个复杂模型的绑定	134
6.8.5	判断模型绑定的结果	136
6.8.6	使用 Bind 属性限制可被更新的 Model 属性	138
6.9	动作过滤器	141
6.9.1	授权过滤器属性	142
6.9.2	动作过滤器属性	145
6.9.3	结果过滤器属性	147
6.9.4	异常过滤器属性	149
6.9.5	自定义动作过滤器属性	150
	总结	152

第 7 章 View 的相关技术	153
7.1 View 的责任	154
7.2 不要用 ASP 的观念来编写 ASP.NET MVC 程序	154
7.3 View 如何从 Action 中获取数据	154
7.3.1 如何正确分配 View	155
7.3.2 使用弱类型模型获取数据	156
7.3.3 使用强类型模型获取数据	157
7.4 HTML 辅助方法	158
7.4.1 如何使用 HTML 辅助方法	158
7.4.2 HTML 辅助方法的应用实例	160
7.4.3 强类型辅助方法	165
7.4.4 Dynamic Data 对 ASP.NET MVC 的支持	166
7.4.5 URL 辅助方法	167
7.4.6 Ajax 辅助方法	169
7.4.7 自定义 HTML 辅助方法	170
7.4.8 TagBuilder 类详解	173
7.5 MasterPage 模板	176
7.6 部分视图	178
7.6.1 什么是部分视图	178
7.6.2 如何创建一个部分视图	179
7.6.3 如何调用部分视图	180
7.6.4 部分视图的高级应用	181
7.6.5 用 RenderAction() 方法加载部分视图	184
总结	185
第 8 章 Area 的相关技术	186
8.1 何时需要使用 Area 切割网站	187
8.2 如何在现有项目中新建 Area	188
8.3 如何设定 Area 的网址路由	189
8.4 默认路由与 Area 路由的优先级	190

8.5 就算使用 Area, Controller 的名称还是有冲突的	190
8.6 如何指定默认网站与 Area 网站的连接	191
总结	192

第 3 篇 开发实践篇

第 9 章 MvcGuestbook 高级实战	194
9.1 分析需求与创建网站雏形	195
9.1.1 规划数据库结构	195
9.1.2 创建基础数据模型	196
9.1.3 规划网站功能	196
9.1.4 设计 Controller 与 Action	197
9.1.5 创建基础页面	200
9.2 实现会员机制	201
9.2.1 会员注册机制	201
9.2.2 会员登录/注销功能	212
9.3 实现留言机制	219
9.3.1 编写留言列表	219
9.3.2 新建留言列表页面	221
9.3.3 为留言列表添加分页功能	222
9.3.4 编写会员留言功能	225
9.3.5 编写会员留言视图	228
9.3.6 编写版主回复功能	228
9.3.7 创建版主回复视图	230
9.3.8 编写删除留言功能	231
9.4 实现动作记录机制	232
9.4.1 编写 ActionLogAttribute 动作过滤器属性	232
9.4.2 将属性应用到 Action	233
9.5 强化 Model 的结构	233
9.6 使用控制反转容器	248

9.6.1	ASP.NET MVC 与 Autofac 的集成	250
9.6.2	ASP.NET MVC 与 Autofac 集成的原理	252
9.6.3	让 Autofac 自动解析接口和对应的类	252
9.6.4	修正 Controller 的构造符	255
总结		256
第 10 章	ASP.NET MVC 开发技巧	257
10.1	ASP.NET MVC 的多语系支持	258
10.2	在 HTTP 响应标题中隐藏 ASP.NET MVC 的版本	261
10.3	一起编译 Visual Studio 和 View	262
10.4	MvcHtmlString 类的开发技巧	265
10.5	在 MVC 与 Web Forms 之间传递数据	267
10.5.1	HTTP GET 方法 (QueryString 集合) 和 HTTP POST 方法	267
10.5.2	Session	267
10.6	如何调试 ASP.NET MVC 源代码	268
10.6.1	使用 .NET Framework 源代码逐步调试功能	268
10.6.2	下载 ASP.NET MVC 源代码并添加项目参考	273
10.7	使用 Visual Studio 程序模板进行快速开发	279
10.7.1	如何使用程序模板快速生成 View	279
10.7.2	如何将内置的程序模板添加到项目中	280
10.7.3	如何修改内置的 AddController 程序模板并自动生成程序代码	283
10.7.4	如何新建自定义的 AddView 程序模板并自动生成程序代码	284
10.7.5	深入 T4 程序生成器	287
总结		287
第 11 章	安装部署	288
11.1	如何将网站部署到本地 IIS 中	289
11.2	如何部署远程 IIS	294
11.3	如何用命令提示符手动部署	296
11.4	部署 ASP.NET MVC 时常见的问题	299
11.4.1	无法加载文件或组件	299

11.4.2	部署到 IIS 6 之后看不到网页	300
11.4.3	部署到 IIS 6 或 IIS 7 之后都无法使用网站	302
11.5	部署 ASP.NET 4.0 时的注意事项	304
11.5.1	安装时的注意事项	304
11.5.2	安装正确的 .NET Framework 包	304
11.5.3	应用程序集区不能跨 .NET 版本设置	305
	总结	305
第 12 章	单元测试	306
12.1	关于软件测试	307
12.1.1	何谓单元测试	307
12.1.2	何谓集成测试	308
12.1.3	如何区别单元测试与集成测试	309
12.1.4	何谓测试驱动开发	309
12.2	创建 ASP.NET MVC 的单元测试项目	311
12.2.1	单元测试项目的命名规则	311
12.2.2	新建项目时自动创建单元测试项目	311
12.2.3	通过单元测试向导创建单元测试项目	313
12.2.4	在 Controller 类中创建单元测试项目	315
12.3	如何编写 ASP.NET MVC 单元测试	315
12.3.1	单元测试的 3A 模板	316
12.3.2	编写 Controller 单元测试	316
12.3.3	在 Visual Studio 中执行单元测试程序	319
12.3.4	查看程序代码涵盖范围	320
12.3.5	编写适合进行单元测试的程序代码	323
12.3.6	使用 Stub 假对象完成单元测试	325
12.3.7	使用 Mock 假对象完成单元测试	327
12.3.8	编写 Routing 单元测试	330
	总结	332