



ASP.NET MVC5

网站开发之美

demo 小朱 陈传兴 王育民 陈仕杰 编著

BOOKASK.COM

清华大学出版社
北京

本书版权登记号：图字：01-2015-1303

本书为碁峰资讯股份有限公司授权出版发行的中文简体字版本。

内 容 简 介

ASP.NET MVC 是微软 Web 开发平台中重要的一块拼图，其架构特性更适合用来开发大型的 Web 应用程序，且 ASP.NET MVC 的开发方式也越来越受到重视，因此学习 MVC 已是刻不容缓。

本书以 ASP.NET MVC 5 为主，涉及 MVC 5.2 RTM、Web API 2.2 RTM 及 Visual Studio 2013 Update 2 等最新内容，应用范围可涵盖 MVC 2~5。全书从快速学会 Model 的概念入手，讲述 LINQ 以及 Entity Framework 开发技术；深度认识 Controller 以及如何灵活运用、使用 Routing 技术，让网站在搜寻引擎中拥有高排名；View+Bootstrap 应用开发；丰富实用的网站安全性技术；详解 Web API、ASP.NET Identity 技术；完整的 CMS 范例分析；充分运用 MVC 的技术与特性；抢先预览 ASP.NET vNext 及 MVC 6 等；是一本内容丰富且扎实的教材与技术参考书。

本书作者在 ASP.NET MVC 开发上均有丰富的实战经验，通过本书可以让读者快速进入 ASP.NET MVC 的开发领域，掌握实用的网站开发技术；对于需要学习 MVC 应用开发的初学者或是已经具备 Web Forms 开发经验的软件工程师来说，都是难得的经典之作。

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 5 网站开发之美 / demo 等编著. —北京：清华大学出版社，2015

ISBN 978-7-302-41338-7

I. ①A… II. ①d… 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 206204 号

责任编辑：夏非彼

封面设计：王 翔

责任校对：闫秀华

责任印制：王静怡

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：清华大学印刷厂

装 订 者：三河市新茂装订有限公司

经 销：全国新华书店

开 本：190mm×260mm

印 张：47.5

字 数：1216 千字

版 次：2015 年 9 月第 1 版

印 次：2015 年 9 月第 1 次印刷

印 数：1~3500

定 价：128.00 元

产品编号：063196-01

推荐（一）

迈向现代化网站技术的世界 — ASP.NET 开发者的路更宽广！

Web 开发技术在这几年，为了顺应移动设备及云服务的应用趋势，有很大的变化和发展，而 MVC 这样的开发框架，在大型及关键的应用开发上也愈来愈普及。然而开发一个完整且现代化的 Web 应用，所需要的技术也变得更加多元化，常让刚进入 Web 开发领域的新手无所适从。ASP.NET 及 Visual Studio 2013 为了适应新时代的软件开发趋势及需求，提供了一个完善的开发环境并且不断地强化，让开发者很容易上手以进行开发工作。

例如 ASP.NET Web API 这项技术，打造了一个共享性很高的服务，以供不同平台及项目开发来使用，是开发人员所需学习的实用技术。另外，像 Bootstrap 这个 CSS/JavaScript 框架，在 ASP.NET MVC 或 Web Forms 的项目模板都已内置其中了，让开发者可以很容易地开发一个响应式网页，给手机、平板电脑及各种设备的用户提供了最好的操作体验；此外，许多对外的网络服务可使用各种社交账号登录，通过 ASP.NET Identity 这个令人赞赏的技术，开发身份验证及管理的程序现在就更方便了！

这本由多位微软 MVP 所执笔的 ASP.NET MVC 书籍，融合了这些开发现代化网站所需的技术，难得集结了各个专业领域有实践经验的专家，由浅入深的章节设计，很适合有系统性地学习。书中这些现代化网站的技术也专门写成独立的章节，即使是使用 Web Forms 技术的开发者也可以参考，还有像是程序的日志处理，都是很实用的网站开发技巧。本书最后提到了软件测试以及网站安全性的概念及方法，例如如何设计一个安全且维护性高的网站架构、如何测试以确保程序的质量，这些实际运用中会碰到的问题及方法，可运用在你的工作上，更增加了本书的价值与实用性。

台湾微软 开发工具暨平台推广处
资深产品营销经理

吴典璋 Dann Wu

BOOKASK.COM

推荐（二）

拥抱 ASP.NET MVC，网站开发再升级

ASP.NET Web Forms 概念的出现掀起一波网页开发革命，对 HTML、JavaScript 一无所知的开发者也能写出 AJAX 网页，简直是件奇迹。然而，魔法般的 Web Forms Server Event，背后需付出性能的代价；过于便捷自由的开发模式，则易让初学者不知不觉在前后端穿插商业逻辑与 SQL 查询，导致逻辑散乱却又紧密关联，维护与测试难度大增。对照 Java Spring Framework、Ruby on Rail、CakePHP 等架构，运用 MVC 设计模式已经成为网站开发主流；常陷于逻辑混杂泥沼的 ASP.NET Web Forms 网站，有时会给人欠缺严谨性，难以构建中大型系统的非专业印象。

2009 年微软推出 ASP.NET MVC 1.0，自此 .NET 开发者有了新选择。根据 MVC 架构，开发人员能按直觉切割出 HTML/JavaScript 等 UI 端逻辑——View、数据逻辑——Model 以及流程衔接逻辑——Controller，无形中实践 MVC 最重要的关注点分离（SoC）精神。而 ASP.NET MVC 在设计时大量应用依赖注入（DI）、AOP（ActionFilter 即为经典范例）等设计模式，处处预留扩展与改写弹性，使架构易于调整，足以面对各种艰巨挑战。而网站能结合单元测试，相比 Web Forms 也是一大突破。转眼间，ASP.NET MVC 已开发到第五版，架构转向 OWIN 开放标准并摆脱对 IIS 的依赖，开发者可自由切换网站平台、身份认证或是处理管道的任一程序包，可塑性几乎没有限制。

个人参与 ASP/ASP.NET 网站开发已十多年，乍见 ASP.NET MVC 便心动不已，深入了解后更为其架构之美所吸引，之后即在新项目身体力行。但对现有的 Web Forms 开发人员而言，MVC 概念差异甚大，转换需要一番学习及引导。个人曾在博客发表 ASP.NET MVC 猪走路系列文章，便是希望引导更多人投身到 ASP.NET MVC，但网上发文及博客文章毕竟较为零散，系统化整理必备知识的书籍仍是学习新技术最有效率的途径。

继在台湾地区出版的《ASP.NET MVC 4 开发美学》之后，由微软 MVP 小朱领军，与 demo、Dino、Bruce 等长期致力 ASP.NET MVC 推广的 twMVC 社群专家，以及专注测试领域的 91，联手为 ASP.NET MVC 再添一本中文参考书。对照自身的 ASP.NET MVC 开发经验，全书完整涵盖使用 Visual Studio 开发 ASP.NET MVC 所需的知识，从 MVC 基本概念、开发环境操作技巧、设计细节到注意事项都有详实阐述。除了 ASP.NET MVC 本身，对于新一代 .NET 语言的重要特色如 LINQ、Lambda、Entity Framework、异步调用等亦有所着墨，以方便开发者活用新武器。至于信息安全议题，在书中亦未缺席。借机在此呼吁，如果你开发网站却不知道 XSS 及 SQL Injection，建议搞懂之前一行程序都别写，以免祸国殃民。最后，除了善用设计模式强化架构，实现自动测试也是专业开发重要的一环。自动测试不仅能确保软件质量，更是进行程序重构与系统优化不可或缺的基石，是提高开发质量的必备技能，建议读者在转换 MVC 时一并补齐。

BOOKASK.COM

从近年微软技术大会中 ASP.NET MVC 议题所占的比重来看,不难感受 ASP.NET MVC 将是未来 ASP.NET 开发的主流。或许读者们手上的 Web Forms 系统仍需继续维护运转十几年(特指还活得好好的企业内部 ASP 程序),但一定要密切掌握 ASP.NET MVC 的开发方向,提前学习做好准备,期盼大家早日加入 ASP.NET MVC 的行列,领略其架构之美。

技术博客“黑暗线程”站长
李明儒 Jeffrey



序（一）

这是我人生中编写的第二本书，与写第一本书时的心情很不同，这一年除了持续在“运营”的 twMVC 社群，我又创立了 SkillTree 的教育组织来推广网页相关技术，并且离开了原本上班族的生活，自己出来创业。多重压力的袭击下脑中经常冒出：“别写了！这年头写书又赚不了什么钱。”这样的念头，中途也实际停笔过，但经常在企业内部培训、公开课程甚至拜访客户时都会被追问到新书的出版时间，问到都心虚了，于是又重新开始动笔，好在小朱的紧迫盯人战术非常有效，原稿竟然不可思议的如期交付出版社，不过在原稿交出的当天晚上可能是长期紧绷的压力得到了释放，我就立即感冒了，并且躺了几天。

一本书籍完成的过程需要许多人的配合与协助，不免俗的要感谢这次愿意和我再次跳下来的作者小朱、Bruce、91 以及初次被推到火坑里的 Dino，再感谢一直在旁边说我写的很差的女王大人黄米奥，当然也非常感谢各位看到此序的读者，中文书籍环境困难，感谢各位还愿意抱一部砖头书回家，书籍上的疑虑或开发上的问题欢迎到 twMVC 与 SkillTree 的活动组织来找我们，期待这本书可以让各位读者收获满满。

Demo

<http://demoshop.tw>

2014 年 8 月 1 日



BOOKASK.COM

序（二）

自《ASP.NET MVC 4 网站开发美学》（台湾地区出版）一书（以下简称《MVC 4 美学》）出版一年半之际，ASP.NET MVC 5 很快就在 2013 下半年挟着 Visual Studio 2013 之势问市，搭配着 Entity Framework 6 以及 Web API 2.0 等热门技术快速席卷了 Microsoft 开发平台的社群。台湾当然也不会排除在这股浪潮之外，因此在《MVC 4 美学》出版半年后，2013 年的 9 月中，两位主要作者 demo 以及 Bruce 被我抓去重庆南路的 KFC（肯德基快餐厅），商讨《MVC 4 美学》下一版的编写事宜，并确立大纲与写作方向。因应 MVC 5 的改变，以及市场对 RWD 的变化，我们决定邀请对 MVC 开发有丰富实战经验的 Dino 加入我们的团队，并再次邀请在软件测试领域学有专精的 91 编写测试这个章节，并将测试一章完全交给 91 负责，作者群至此确立。不过我们还是花了半年的时间编写，毕竟大家都不是全职作者，原本的饭碗还是要顾及的☺。

在《MVC 4 美学》上市的半年，我们也陆续接到了很多读者的反馈，其中有些读者对 Model（《MVC 4 美学》第 2 章）的部分不是那么满意，所以 Model 部分由我重新改写，份量由单章扩编为三个章节，充分介绍 Model 所需要的概念、LINQ 技术以及重要的 ORM 框架 Entity Framework，期望给读者一个全新的视角与感受。

这本书能在重要的时机点问世，首先要感谢的是被我紧迫盯人的作者群，虽然我本身也参与写作，身兼这本书 PM 角色的我更重视原稿的质量与交稿期，所以作者群在截稿前的几个星期都被我百般叮咛，直到稿件交出为止，辛苦你们了☺。接着要感谢编辑 Tony，对我们这些非专职作者没有太过于要求时间，无形中也降低了我们不少压力。然后要感谢台湾微软开发工具暨平台推广处的 Dann 经理以及 .NET 社群中无人不知无人不晓的黑暗线程愿意作序推荐。最后要感谢的是在本书写作期间给我帮助的朋友们。

朱明中（小朱）

studyazure.com

2014 年 8 月@高雄

序（三）

很高兴那么快的时间内，马上又和小朱、Demo、Dino、91 有合作的机会。在现今微软网页开发技术快速更新的时期，合作是最能代表 M-V-C 关注点分离（SOC: Separation of concerns）的精神与体现，每位作者都展现独门功夫来带领读者一一破解 M-V-C 每一关卡。

在 Controller 一章我尝试大量使用 ASP.NET MVC 5 源代码来进行说明，对于不习惯阅读源代码的读者，笔者在阅读上建议可分三个层次：一、知；二、懂；三、通。首次阅读，可以先把源代码放一边，先把 Controller API 的使用及轮廓了解清楚。二次阅读，配合源代码慢慢咀嚼，以清楚 Controller API 的运行机制。三次的实践练习，以求融会贯通。

老实说，江湖一点绝，而 MVC 的源代码就是绝窍所在，当你懂得查询 MVC 源代码，不会碰到以往黑箱（封装）操作困扰，取而代之的是明明白白的工作原理，而这些基础可以让读者在未来开发 MVC 应用程序时，能清楚了解所写的每一行程序代码所代表的意思。而我希望带领读者跨出这一步。

另外，Visual Studio 2013 内建的基架系统，已经能近乎无脑的方式生成所有异步程序代码，因为害怕被开发者滥用，在 Async Programming 一章我也花费大量篇幅进行异步程序开发基础知识的讨论，而并非只是简单讨论 async 与 await 关键词，这些基础知识是希望当读者在使用异步技术碰到问题时，能有思考与排查的能力。当然，这离完全理解异步程序的本质还有一段不小的距离。

其次是下一版的 MVC 6 已开始进行开发了，但读者也不用害怕，MVC 6=MVC+Web API+Web Pages 三位一体的框架，也就是说，MVC 6 注重在集成，并非破坏式更新。学好 MVC 5 对于未来升级至 MVC 6 有直接的帮助。

最后非常感谢老婆家华与两位宝贝女儿（千乐与千爱），写作时间得来不易，每分每秒都是牺牲家庭与亲子时间，利用假日与肝火上升的时间来写作，而你们永远体贴与包容，让我能心无挂碍全心写作，没有你们的爱，我不可能顺利交稿。

陈传兴（Bruce）

<http://kkbruce.tw>

2014 年 8 月于新竹

序（四）

在偶然的机会下，有幸与多位业界知名专家进行本书的合著计划，内心着实兴奋却也惶恐着，虽然以 ASP.NET MVC 闯荡项目领域有一段时日了，但以著作的方式分享自己所知所学还真是头一遭，过程屡屡经过修正，一直揣摩着初学者所需要的知识，同时也试着为已在使用 ASP.NET MVC 的开发人员提供一部分提升自己能力的议题。

在此书中我负责 View、Bootstrap 以及应用程序部署的部分，个人觉得 View 在应用程序开发完成进入维护后，是最为容易变动的部分，维护与开发 View 需要的知识虽然不像后端那样艰深、广泛，但绝对需要相当的技巧来协助工作的进展与组织化，希望章节中分享的技巧与概念，能对开发人员有所帮助。

回首 MVC 开发之路，感谢过去曾经一起工作过的同事们，多元的团队成员总是能激发出更多的开发想法，身在优秀团队中的学习机会绝对比一人单打独斗要强上百倍，有相互切磋开发思维的绝佳机会，虽然辛苦但却是一段非常值得珍惜的回忆。

最感谢的莫过于在这段期间，内人与正值青春期的儿子 byte，还有基隆家中的老爸老妈，他们对我自私的在工作以外还投入书籍撰写工作毫无怨言，并且总是非常支持，赶稿的日子里也减少了假日回到老家基隆团聚的机会，在这里由衷地感谢家人们的支持。

王育民（Dino Wang）

<http://www.hexdigits.com>

2014 年 8 月于新北市



BOOKASK.COM

序（五）

很高兴这么快再次获得好伙伴们：小朱、Demo、Dino 与 Bruce 的邀请，能有这样的机会再次与大家合著 ASP.NET MVC 一书，心里倍感荣幸。与上一本《ASP.NET MVC 4 网站开发美学》出版的时间虽然只间隔一年，但这一年 ASP.NET MVC 加入了相当多设计更美妙、更具弹性的元素，而在这一年之间我对 ASP.NET MVC 及其测试开发相关的知识与经验，也有着大幅度地成长与更多体会。因此，这一次光是关于实际运用中如何进行自动测试的章节就高达六十页左右。相信读者们拿到这本书时，第一个感觉一定就是：“怎么这么厚！”。因为不只是我，每位作者都竭尽全力地想要在书中带给读者一个不同的体验，不再只是从无到有去了解 ASP.NET MVC，而是加入了更多实战的经验与设计技巧。相信对读者来说，这本书不只是用来窥探 ASP.NET MVC 的面貌，而是一本够广也够深的 ASP.NET MVC 开发指南。

感谢老天让我的第一个女儿思彤在酝酿这本新作的过程中诞生，更感谢我的老婆玫君非常用心地照顾女儿跟我这个大小孩，让我能放心专心地跟大家一起贡献自己的知识，我真的很爱你们！也很感谢伙伴们在这个过程中的帮助与打气，让我觉得信息技术这条路上并不孤单，能有这样的机会再次跟你们并肩作战，真的很棒！

另外也要感谢两位导师，一位是 Ruddy 老师，总能在我迷惘时指引我学习与人生的方向，总能激起我的热情与动力，总能解答我的疑惑。另一位是好友黑暗线程，因为黑大 2007 年的一篇 IEUnit 文章，让我迈出了自动测试的第一步。没有这两位，现在就不会有这个对测试开发这么有热情的 91。

陈仕杰（91）

2014 年 8 月于台北南港

前言

从事 Web 开发工作十多年来，从来没有像开发 ASP.NET MVC 应用程序这样的自在过，精巧的架构设计执行在强大的 .NET Framework 中，你能想象到的功能，就算 ASP.NET MVC 不直接内建其中，你也能找到优雅的扩展点，为 Web Application 注入你的需求与想象。这个框架中存在各种各样的美感，呵～当然，这些可是“程序宅人”专用的浪漫啊。

光是吹捧 ASP.NET MVC 可是不行的，毕竟得要拿点成绩出来瞧瞧才有说服力。我是 Dino，经过 Bruce 推荐以及本书其他作者们的同意，就斗胆拿个作品出来现丑，分享一下个人用 ASP.NET MVC 的开发成果——“华人健康网”。简单介绍一下华人健康网，它是一个专业的健康信息提供网站，目前与台湾 Yahoo! 合作，并以联名的模式经营，每天自访自产提供最新健康新知给广大网友，每天的 Page Views（综合浏览量）约 30~60 万，过去一年总访问量超过 150,000,000（1.5 亿）。以 Google Analytics 来看，Active Users（活跃的用户数）可达 4,800 人，由 2 到 3 台网站服务器提供服务¹，访问量目前仍在成长中。这个产品由我个人（后来成为 hexdigits.com）以项目的模式承接开发与持续维护。

华人健康网在技术细节上运用了 ASP.NET MVC（历经版本 3、4、5.x）、Entity Framework Code-First（历经版本 4.1、5.0、6.x）、.NET Framework（历经版本 4、4.5、4.5.1）等微软技术。除了 MVC 本身，其他比较容易引起关注的是使用了 Code-First 技术，而且是在 Entity Framework 4.1 时代就开始使用了。Code-First 以 POCO（简单的 CLR 对象，Plain-old CLR Object）类直接对应数据库的方法，再加上 Metadata 的诠释，打开类文件 schema 信息便一目了然，省去了不少繁琐步骤。采用 Code-based Migration 更是以 C# 语法保留了数据库的变更记录，很精彩，你一定不要错过。Code-First 可说是非常有利于小团队开发的方法²，很可惜的这项精彩技术似乎一直没有被专门推广以及广泛使用，至少周边接触到的朋友、项目乃至顾问项目皆是如此。据了解，不熟悉、掌握度低为主要因素。但笔者直到目前为止仍然以 Code-First 进行维护与持续开发，成立公司以来承接的项目亦全为 Code-First 的成功案例 ©。至于 Code-First 的进一步细节，在 Model 一章中小朱老师有精彩的介绍，在 ASP.NET Identity 一章中 demo 也会有实战分享。

华人健康网起初并不是具有目前规模的网站，在 2011 年我结束一个知名 EC 项目工作后，休息了好一阵子才承接，单枪匹马历时约三个月完成开发，2012 跨年（迎接新年的元旦）前上线，因为有万分的把握，上线不到一天我就合家跑到香港跨年（过元旦）去了。网站前两季的平均表现大约是每天 500 个不重复用户，1,000 多个造访页次，多的时候大概多个三、四倍左右，运行于一台光华买来的简单服务器上，“命很硬的”，就这样一直撑到内容充足了，访客依赖度高了，于是合

¹ 使用 Microsoft Azure 的 Cloud Service，并且启用 Auto Scaling 机制，不必全时开启最大的可用服务器数目。

² 大型团队的 Code First 经验还无法分享，不过据说若有专责 DBA 就不大容易推动 Code-First，若有机会笔者倒想合作看看。

作机会就来临了。

与 Yahoo!这样知名公司的合作，是彼此极好的机会，但是也的确带来极大的精神压力，因为这预期会有一个突破千倍甚至万倍的成长，然后我和业主作了一个一辈子都不会后悔的选择³，让它上公有云“Microsoft Azure”。综合评估下来，像华人健康网这样的新创公司，选用公有云服务是兼顾成本以及质量的最佳选择。十分感谢评估期间小朱老师在一次 twMVC 的每月固定活动课程场合中，不吝抽空提供了经验咨询。

不过你知道吗？华人健康网在面临万倍成长之际，原有的 ASP.NET MVC 应用程序架构可是一行程序代码都不需要调整！是胆子大，赌一把吗？当然不是的，是遵循优秀的 MVC 开发架构让我的应用程序很健康，易于扩展更省下不少修改功夫，加上 ASP.NET MVC 的程序本身运行性能就优异，后来经过了长时间的网站持续成长，才开始进一步的调校。上线前夕只有和公有云上不太一样的存储思维，为先前设计好的 interface 增加了 storage provider 以及 cache provider 的具体实现，并通过 IoC 的技巧更换掉 providers⁴，差不多就完工了。虽然在评估建立机房、联系台湾知名 IDC ☺ 或是选择云技术期间虚度了两个月，最终决定采用公用云，实际上动手调整只花了不到一个礼拜的时间，更实在的说，想透彻后大概只花了三天。当然，如果你是团队协同工作，不可能有几乎为零的沟通成本，应该要自行斟酌时间。

2012 年 10 月 18 日，联名的“Yahoo!奇摩健康 x 华人健康网”改版上线，相隔一周，我刚好又安排合家度假去了。哈，后来这个上线后度假的小惯例多次在公开的 Azure 分享场合中，让微软的 John 给点名了。哈～必须得说，这是因为使用了 ASP.NET MVC 开发应用软件就是这样的悠然自得，运行在 Microsoft Azure 服务上就是让人这样的安心、放心。

³ 感谢华人健康网总经理 Jeff 有支持开发者决策的勇气，他事后也在朋友圈中多次确认因为这项决定，和其他创业者相比，成本低得惊人。

⁴ Storage/Cache Provider Interface 的定义以及 IoC 的使用，归功于过去开发经验的累积，所以提前做了设计上的保留，团队若没有架构上的经验应该多加阅读，或寻求可咨询的顾问服务。

导 读

ASP.NET MVC 5 是一个很有威力、很具扩展性、很艺术的一种 Framework。很有威力是指它具有能让 Web 程序员开发出大型 Web 应用程序的基础建设；很具扩展性是指 MVC 内的每个主要功能都能由程序员自己定制化，MVC 本身也是个 Open Source 项目，程序员可随时查看并修改它的源代码，做出一个自己的 MVC Framework；很艺术是指程序员能运用 MVC 开发出各种不同的应用，而且加上 MVC 的定制化能力，让应用程序的开发不再是一个很固定的模式，而是能由程序员（或团队）自己决定。本书的目的，就是要给想要用 MVC 来开发应用程序的程序员一个方向，指引程序员在不同的关注点（concerns）中，应用手边的资源开发出现代化的 Web 应用程序（Modern Web）。

本书定位

本书的读者群定位在：

- 没有编写过 Web 应用程序，但有 C# 或 VB 程序设计语言的基础，想运用 MVC 开发 Web 应用程序的学生或工程师。
- 编写过一阵子 Web Forms 应用程序，不想再为控件、ViewState 或是固定的 Page Lifecycle 所苦，想跳槽到 MVC 领域的 Web 程序员。
- 编写过 ASP、PHP、JSP 等平台应用程序，想认识并利用 MVC 开发 Web 应用程序，以移植自身能力到 MVC 的 Web 程序员。
- 想彻底认识 ASP.NET MVC 的系统分析师、数据库管理师、系统架构师与软件开发经理。

本书会要求读者先有 C# 或 VB 的基础，至少要懂得如何编译一个 C# 应用程序，并知道基本的 Visual Studio 操作，书中虽然会有一些 step-by-step，但不会涉及完整的 Visual Studio 开发环境的操作，所以没用过 Visual Studio 的读者，可先下载安装 Visual Studio Express for Web 来玩一玩，体验一下这个地表最强开发工具的简易版 ☺，也熟悉一下方案与项目的结构。

开发环境与工具

工欲善其事，必先利其器。

本书使用的开发环境是 Visual Studio 2013，并以 Ultimate 版本为编写环境，不过大部分的功能都能在 Visual Studio Express for Web 上使用，这点读者倒是不用太担心，而本书第 15 章介绍到 Microsoft Azure 时，会要求安装 Microsoft Azure SDK 2.3 及其工具，读者可到 Azure 的下载网站（<http://mvcbook.net/006c>）下载 Visual Studio 2013 版本的 Azure SDK 与工具，安装十分方便。

本书内容

本书共有 16 个章节，各章节介绍如下：

- 第 1 章 MVC 概述

简单介绍 MVC 这个 Pattern（模板），以及 ASP.NET MVC 的总览，包含开发环境、NuGet、IIS Express 等开发时期会用到的工具，同时还会介绍 Web Forms 与 MVC 的差异。

- 第 2 章 与数据的对话：Model 与 ADO.NET

主要介绍 Model 的概念，ADO.NET 的基本功能，以及用 ADO.NET 简单开发出一个简单的 Model 层，并强调 Model 并不是一定要用 Entity Framework 才算数。

- 第 3 章 LINQ：驱动数据的查询能力

本章会介绍 LINQ 这个在 .NET Framework 3.5 就有的语言功能，不会使用 LINQ 的话，日后会无法阅读很多新技术所提供的程序代码（无论是源代码或是范例程序），所以 LINQ 的基础知识将在本章夯实。

- 第 4 章 Entity Framework

本章介绍 Entity Framework，这个由微软亲手打造的 ORM Framework，包含 Database First, Model First 与 Code First，以及处理关联和继承时的方法，最后还会说明 Database Migration 的方法。

- 第 5 章 Routing

ASP.NET Routing 是 MVC（以及 Web Forms 的 Friendly URL）的核心功能，通过 Routing 处理，才能让 URL 对应到正确的 Controller 与 Action，所以如何善用 Routing 是个重要的议题。本章也会介绍大型应用程序会用到的 Area，运用 Area 切割并模块化网站。

- 第 6 章 Controller

本章会是这本书份量最多的一章，Controller 作为处理与传递 Model 给 View 的中间人，重要性当然不在话下，Model Binding、Metadata、Action、ActionResult、Filters 等重要 Controller 内用到的技术，在这一章都不放过，完整交代清楚，所以读者要花较多时间在这个章节中。

- 第 7 章 Async Programming

都升级到 .NET 4.5 了，还没搞懂 async 和 await 这两个运算符在做什么吗？本章将会给读者充分的异步程序设计的基本概念，以及各种异步程序设计方法的说明。

- 第 8 章 View：摇曳生姿的美人

View 在 MVC 应用程序中扮演了为应用程序擦脂抹粉的角色，所以也是应用程序中十分重要的一部分，本章说明了 View 的操作方式，以及搭配 View 的重要指令工具：Razor，Razor 配合各种各样的 HTML Helpers 与 URL Helpers，能让编写前端的工作事半功倍。

- 第 9 章 Bootstrap

本章为本书特色之一，Bootstrap 是一个能轻易让网站变美的前端 CSS/JavaScript Framework，本书将会对它做一个介绍，以及它如何与 MVC 搭配使用。

- 第 10 章 诊断与日志处理

本章将说明 MVC 应用程序的诊断（Diagnostics）功能，包含内建的 Trace、搭配 ELMAH 处理记录的传递与提示，以及轻量化的 NLog 工具。

- 第 11 章 网站安全之道

本章与第 12 章为 Web 应用程序安全之章，本章会以 XSS、CSRF、加密技术以及密码处理原则为主，诠释一个 ASP.NET MVC 应用程序的网站安全需要关注的地方。

- 第 12 章 身份验证与授权

承接第 11 章的内容，本章介绍的是身份验证与授权，包含常见的 Session、Cookie 以及 Forms Authentication 方法，同时会更进一步的介绍 ASP.NET Identity 这个新一代的认证与授权服务，它能集成数据库、OAuth 等不同来源的验证要求，并支持自定义密码复杂度与双因素认证等高级功能。

- 第 13 章 ASP.NET Web API 2 概述

本章会介绍 Web API 2，这个在 MVC 应用程序中提供 RESTful API 的开发平台，包含 HTTP 动词与 Action 的对应、Web API Scaffolding、Entity Framework 循环引用以及 JSON 等内容。

- 第 14 章 自动测试完整攻略

本章亦为本书特色之一，由概念、工具的操作与使用、个案实现等一应俱全，完整的诠释了在开发过程中所需的单元测试与集成测试的概念，对于想要学习 Web 测试的读者而言，本章是最佳的献礼。

- 第 15 章 网站部署

本章将介绍如何部署 MVC 应用程序到 IIS 或是 Microsoft Azure 环境，包含 Website 及 Cloud Service 两种不同的服务。

- 第 16 章 CMS 范例实战

本章将以实例方式串连本章前面所有的章节，实现一个 CMS 范例应用程序。

范例程序

读者可在本书专用的 GitHub 网站：

(<https://github.com/MVCAppDesignAndDevelop/MVC5Book>)

找到完整的范例程序代码，进入 GitHub 网站后，使用 Download ZIP 就能下载所有范例程序，若是对 git 指令⁵ 熟悉的读者，也可以使用 git clone 指令来复制所有的范例程序。

<https://github.com/MVCAppDesignAndDevelop/MVC5Book>

本书应用程序若有使用到数据库，大部分都是以 Northwind 数据库为主，若有不同会在该章节中特别说明，同时我们使用的是 SQL Express LocalDb，若想用 SQL Express 或是 SQL Server 作为数据源时，请记得修改连接字符串，若使用的是 Entity Framework 时，则也要修改 Entity Framework 本身的 Provider 设置，这部分请参考第 4 章的内容。

读者服务

本书的读者服务由本书作者群共同负责，由 twMVC 社群作为主要窗口，读者若有本书内容与

⁵ 若使用 Windows 又没有 git 指令，请安装 Git for Windows: <http://mvcbook.net/006d>。

范例程序的任何问题，可以直接联络 twMVC，或是利用各作者的博客与各作者取得联系。

demo: <http://demo.tc>

小朱: <http://www.dotblogs.com.tw/regionbbs/>

Dino: <http://dinowang.blogspot.tw/>

Bruce: <http://blog.kkbruce.net/>

91: <http://www.dotblogs.com.tw/hatelove/>

本书配套源代码下载地址（注意数字与字母大小写）：<http://pan.baidu.com/s/1mg5XVPA>，若下载有问题，请发送电子邮件至 booksaga@126.com，邮件标题为“求代码，ASP.NET MVC 5 网站开发之美”。



BOOKASK.COM

特别介绍



twMVC 社群位于台北，成立两年多以来已经举办过超过 15 场的研讨会以及无数场的【每周四固定聚会】，这两年来十分感谢一路支持的朋友与伙伴，我们深知群体学习的好处与效果，于是我们还坚持在这里！每周四固定聚会不限主题轻松聊技术，最新活动消息欢迎关注我们的 FB 粉丝团以及官方网站，期待您就是我们的下一位访客。

- 官方网站: <http://mvc.tw>
- 粉丝团: <http://fb.me/twMVC>



SkillTree 的创立是因为 demo 觉得研讨会时间太短无法完整交代整个流程与细节，而另外成立的教育组织，与一般补习班不同，我们都是拥有丰富经验的业界讲师，我们不把课程时间浪费在述说历史与沿革，我们并不是教您考取证照，而是教您如何上场杀敌，拳拳到肉的课程内容才是您花钱想要听到的，而这也刚好是我们擅长的。

- 官方网站: <http://skilltree.my>
- 粉丝团: <http://fb.me/skilltree.my>



为学习而生的技术社群，由一群爱好技术的台北、中、南朋友们，希望凭借社群的力量，给有兴趣讲课的朋友一个舞台，给感兴趣听课的朋友一个座位，给那偏远地区的朋友一个机会；我们持续地在台湾各地区办活动，希望凭借各个地方的聚会，认识更多爱好技术的朋友，并且互相分享与交流不同领域的技术；让在校的学生，在职的人士，能看到更多更广更有趣的东西。

- 官方网站: <http://study4.tw>
- 粉丝团: <http://fb.me/Study4.tw>

BOOKASK.COM

目 录

第 1 章 MVC 概述	1
1.1 MVC 架构概述	1
1.1.1 MVC 是什么	1
1.1.2 关注点分离	2
1.1.3 约定优于配置	2
1.1.4 MVC 的优点与特长	2
1.1.5 MVC 的成长趋势	3
1.2 ASP.NET MVC 概述	4
1.2.1 与传统 ASP.NET Web Forms 的差异	4
1.2.2 Model 的责任	5
1.2.3 View 的范围	6
1.2.4 Controller 要做哪些事	7
1.3 ASP.NET MVC 5 开发环境	8
1.3.1 One ASP.NET	8
1.3.2 Visual Studio 2013 Web Templates	9
1.3.3 Scaffold (基架)	12
1.3.4 Visual Studio 2013 新增功能介绍	12
1.4 NuGet 的使用	20
1.4.1 安装程序包	20
1.4.2 还原程序包	22
1.4.3 更新程序包	26
1.4.4 程序包管理器控制台 (Package Manager Console)	26
1.5 IIS Express	29
1.5.1 如何在 Visual Studio 中使用 IIS Express	30
1.5.2 如何使用命令行启用 IIS Express	31
1.5.3 让 IIS Express 也支持外部连接	34
1.5.4 整理过多的网站信息	35

1.6	ASP.NET Web Forms 转换至 MVC 开发.....	36
1.6.1	从 Web Forms 转到 MVC 后开发概念的改变.....	36
1.6.2	数据源的改变.....	36
1.7	创建第一个 MVC 因特网应用程序.....	41
1.7.1	创建 MVC 项目.....	41
1.7.2	创建数据库（使用 Local DB）.....	45
1.7.3	创建 MVC 中的 Model.....	48
1.7.4	创建 MVC 中的 View 与 Controller.....	52
1.7.5	更新为最新 MVC 版本.....	55
1.7.6	MVC 的生命周期.....	56
1.8	结术语.....	58
	参考资料.....	58
第 2 章	与数据的对话：Model 与 ADO.NET.....	60
2.1	Model 的概念.....	60
2.1.1	Model 的类型.....	61
2.1.2	Model 的设计.....	61
2.2	ADO.NET.....	63
2.2.1	面向连接的操作.....	63
2.2.2	面向无连接的操作.....	64
2.3	泛型概念.....	65
2.3.1	泛型基础.....	65
2.3.2	设计泛型类.....	66
2.4	Model 的实现.....	70
2.4.1	Model 的设计.....	70
2.4.2	Model 数据读写的设计.....	71
2.5	SQL Injection.....	82
2.6	结术语.....	83
第 3 章	LINQ：驱动数据的查询功能.....	84
3.1	LINQ 概念.....	84
3.1.1	LINQ vs. 循环处理.....	85
3.1.2	LINQ 的类型.....	88
3.2	LINQ 的基础.....	89

3.2.1	扩展方法	89
3.2.2	匿名类型与对象初始化器	91
3.2.3	类型推论	93
3.2.4	yield 指令与延迟查询	94
3.2.5	Fluent Interface	96
3.3	Lambda 表达式	97
3.3.1	Lambda 表达式的由来	97
3.3.2	Lambda 表达式的格式	98
3.3.3	通用委派与 Lambda 表达式	100
3.4	LINQ 语句	100
3.5	LINQ 函数	101
3.5.1	查询结果过滤: Where()	102
3.5.2	选取数据: Select()、SelectMany()	102
3.5.3	群组数据: GroupBy()、ToLookup()	105
3.5.4	联接数据: Join()与 GroupJoin()	108
3.5.5	数据排序: OrderBy()与 ThenBy()	109
3.5.6	获取集合	112
3.5.7	划分并获取集合	113
3.5.8	集合的运算	114
3.5.9	访问元素	116
3.5.10	聚合与汇总	118
3.6	远程查询: IQueryable<T>与 IEnumerable<T>	120
3.7	Expression	121
3.8	LINQ 与 ADO.NET: DataSet/DataTable 的使用	124
3.9	结术语	126
第 4 章	EF	127
4.1	ORM 概念	127
4.2	EF 概述	129
4.3	EF 架构	133
4.3.1	数据模型映射	133
4.3.2	Entity SQL Service	135
4.3.3	Object Services	135
4.3.4	LINQ to Entities	137

4.4	Database First.....	138
4.5	Model First	143
4.6	Code First	150
4.6.1	生成模型——空白 Code First 的做法	150
4.6.2	生成模型——由数据库生成	155
4.6.3	Code First Modeling APIs.....	156
4.6.4	实现关联性	159
4.6.5	实现继承	163
4.6.6	存储过程和函数映射	170
4.7	通过 DbContext 访问 EF 数据模型.....	172
4.7.1	DbSet<T>	172
4.7.2	以对象为主的查询.....	173
4.7.3	以 SQL 为主的查询	174
4.7.4	数据的变动	176
4.7.5	验证.....	177
4.7.6	更新的考虑	179
4.8	更新数据库: Database Migration	180
4.8.1	数据库初始化器	180
4.8.2	数据库迁移-自动迁移	181
4.8.3	数据库迁移-手动迁移	186
4.8.4	数据库迁移-程序迁移	188
4.9	结术语	189
	参考文献	189
第 5 章	路由 (Routing)	190
5.1	URL Rewriting 与 Route 基本概念.....	190
5.2	URL Routing 基本应用	191
5.3	加入自定义的 URL Routing	193
5.4	URL Routing 如何对比	194
5.5	为 URL Routing 加上限制条件	195
5.6	URL Routing 高级应用	197
5.6.1	如何从程序建立一组网址	198
5.6.2	直接获取 RouteData 内的值	198
5.7	如何调试网站	199

5.7.1	使用 Route Debugger 2.0.....	199
5.7.2	使用 Cobisi Routing Assistant.....	200
5.8	Area.....	201
5.8.1	创建第一个 Area.....	202
5.8.2	使用 Area 会遇到的问题.....	204
5.8.3	根据功能模块区分的 Area.....	206
5.9	结术语.....	207
第 6 章	控制器 (Controller).....	208
6.1	了解 Controller 角色.....	209
6.1.1	了解 Controller 工作流程.....	210
6.1.2	ASP.NET MVC 5 验证系统.....	216
6.1.3	创建 Model 层.....	219
6.1.4	基架系统.....	222
6.2	Controller 与 View 的数据传递.....	227
6.2.1	ViewData 属性.....	227
6.2.2	ViewBag 属性.....	229
6.2.3	ViewData、ViewBag 传递 Model.....	231
6.2.4	ViewData.Model 属性.....	234
6.2.5	TempData 属性.....	236
6.2.6	多 Model 与多对象传递.....	241
6.3	Model Binding.....	251
6.3.1	Request 对象.....	251
6.3.2	RouteData.....	251
6.3.3	简单 Model Binding.....	252
6.3.4	复杂的 Model Binding.....	256
6.3.5	JSON Model Binding.....	259
6.3.6	ModelBinder 扩展.....	260
6.4	Metadata 与数据验证.....	263
6.4.1	部分类.....	263
6.4.2	验证属性.....	265
6.4.3	ModelState.....	285
6.5	ActionResult.....	297
6.5.1	EmptyResult.....	300

6.5.2	ContentResult	300
6.5.3	JavaScriptResult	303
6.5.4	JsonResult.....	304
6.5.5	HttpStatusCodeResult	309
6.5.6	RedirectResult 与 RedirectToRouteResult	310
6.5.7	FileResult	317
6.5.8	ViewResult	329
6.5.9	VideoResult	335
6.6	Action Filters	338
6.6.1	Authentication Filter	340
6.6.2	Authorization Filter	347
6.6.3	Action Filter	354
6.6.4	Result Filter	356
6.6.5	Exception Filter	360
6.6.6	Action Filters 接口与扩展.....	363
6.6.7	Filter overrides.....	368
6.7	其他 Controller 技巧.....	371
6.7.1	ActionName 属性	371
6.7.2	NoAction 属性.....	371
6.7.3	支援多国语系	372
6.8	结术语	375
第 7 章	异步程序设计	376
7.1	异步基架生成器	376
7.2	进程与线程	380
7.2.1	进程.....	380
7.2.2	线程.....	381
7.2.3	多线程的启动	385
7.3	同步与异步程序设计	386
7.3.1	同步程序设计	387
7.3.2	异步程序设计	387
7.4	Task 类.....	388
7.4.1	接续工作	389
7.4.2	Task.Delay	390

7.4.3	Parallel 类.....	391
7.5	async 与 await 关键词	392
7.5.1	C# 4.0 与 C# 5.0 异步差异.....	394
7.5.2	同步方法读取网络资源	396
7.5.3	异步方法读取网络资源	398
7.5.4	异步方法执行流程.....	400
7.6	异步与线程池	401
7.6.1	如何选择同步或异步操作	405
7.6.2	执行多个并行操作.....	405
7.6.3	使用 cancellationToken	406
7.6.4	多线程与异步	408
7.7	结术语	409
第 8 章	View: 摇曳生姿的美人	410
8.1	View 概述.....	410
8.2	View 在项目中的位置.....	411
8.2.1	Views 目录	412
8.2.2	Views 下的 Shared 目录.....	413
8.2.3	Views 下的常规目录.....	413
8.3	View 的类型.....	414
8.3.1	View 常规视图	414
8.3.2	Partial View 部分视图	416
8.3.3	Layout 主版页面	417
8.3.4	Template	420
8.4	View 获得数据的方法.....	423
8.4.1	Model	424
8.4.2	ViewData 及 ViewBag	424
8.4.3	TempData	425
8.5	View Engine 概述	425
8.5.1	WebForm View Engine	425
8.5.2	Razor View Engine	426
8.5.3	View Engine 的执行细节	426
8.5.4	View Engine 的扩展性.....	426
8.5.5	Razor View Engine 对于 View 文件的搜索方式.....	427

8.6	Razor 语法.....	428
8.6.1	Razor 语法概述	429
8.6.2	程序代码区块	432
8.6.3	编译指示词	433
8.6.4	排版与辅助功能	434
8.6.5	全站可用的 Razor Helper	438
8.6.6	ASP.NET MVC 4/Razor v2.0 增强部分	440
8.7	Helpers.....	443
8.7.1	UrlHelper.....	443
8.7.2	HtmlHelper.....	445
8.7.3	AjaxHelper	465
8.7.4	扩展 Helper	469
8.8	View 开发技术的高级功能.....	474
8.8.1	前端优化: JavaScript/CSS 的合并与优化	474
8.8.2	View Engine 文件搜索优化	478
8.8.3	Layout 嵌套	481
8.9	结术语	483
第 9 章	Bootstrap	484
9.1	Bootstrap 导览.....	485
9.1.1	Basic Template.....	486
9.1.2	Grid System	486
9.1.3	Component	492
9.1.4	Bootstrap 的 JavaScript 链接库	492
9.2	定制化 Bootstrap.....	494
9.3	ASP.NET MVC 与 Bootstrap 合璧.....	496
9.3.1	Component: Label、Badge、Alert、Progress	496
9.3.2	表单域的集成	499
9.4	结术语	502
第 10 章	诊断及日志处理	503
10.1	预处理器	503
10.1.1	基本规则	503
10.1.2	预处理器	504

10.2	Trace 与 Debug 类	506
10.2.1	Assert 方法	507
10.2.2	Listeners 集合	508
10.3	艾尔玛 (ELMAH)	509
10.3.1	艾尔玛的作用	510
10.3.2	使用 ELMAH.MVC	514
10.3.3	XML Out 和 Database In	516
10.3.4	网站自动报错功能	518
10.3.5	ELMAH 变形金刚	520
10.3.6	ElmahR = ELMAH + SignalR	522
10.3.7	ELMAH.io	523
10.4	NLog - Advanced .NET Logging	524
10.4.1	调用 N 日志 (NLog)	524
10.4.2	NLog 实战	529
10.5	ELMAH 与 NLog 合体术	540
10.5.1	Log Reporting Dashboard	540
10.5.2	ELMAH.MVC + NLog	540
10.6	结术语	541
第 11 章	网站安全之道	542
11.1	知己知彼	542
11.1.1	XSS 的防范	542
11.1.2	如何有效防范 XSS	544
11.1.3	防止跨网站伪造请求	545
11.1.4	参考资料	548
11.2	加密与解密	549
11.2.1	加密博客	549
11.2.2	窗体验证原则	552
11.2.3	ASP.NET 加密技术	553
11.2.4	哈希密码	556
11.2.5	Crypto 类	557
11.2.6	加密 web.config	557
11.2.7	登录监控	560
11.3	结术语	560

第 12 章 身份验证与授权	561
12.1 使用 Session 做身份验证	562
12.2 使用 Cookie 做身份验证	564
12.3 使用 Forms Authentication 做身份验证	568
12.4 全新的 ASP.NET IDentity 2.0	570
12.4.1 ASP.NET IDentity 2.0	572
12.4.2 自定义验证逻辑	576
12.4.3 电子邮件验证	578
12.4.4 双因素认证	584
12.4.5 将用户加入角色	584
12.4.6 集成 OAuth	587
12.5 扩展会员字段	594
12.6 结术语	598
第 13 章 ASP.NET Web API 2 概述	599
13.1 Web Service	600
13.1.1 Web Service 架构	601
13.1.2 XML Web Service	602
13.1.3 WCF	604
13.1.4 WCF REST	604
13.1.5 小结	605
13.2 ASP.NET Web API 2	605
13.2.1 IIS Express 指令模式	607
13.2.2 Web API 路由	608
13.2.3 REST	609
13.2.4 HTTP 动词+URI	616
13.2.5 小结	616
13.3 Model 与 Web API 2 基架	617
13.4 EF 循环引用	619
13.5 JSON	621
13.5.1 XML 与 JSON	622
13.5.2 了解 JSON 格式	623
13.5.3 使用 JSON 数据	624

13.5.4	JSON-LD	626
13.6	POSTMAN	628
13.7	结术语	630
第 14 章	自动测试完整攻略	631
14.1	测试概念简介	631
14.1.1	测试的目的	631
14.1.2	测试的种类	632
14.1.3	加入测试的时间点	633
14.1.4	测试的效益	634
14.1.5	小结	635
14.2	测试程序设计技巧与工具的介绍	635
14.2.1	单元测试	636
14.2.2	Web 自动测试	654
14.2.3	行为驱动开发 (Behavior-Driven Development, BDD)	662
14.3	实战演练——以重构计算运费为例	668
14.3.1	范例说明	669
14.3.2	建立测试案例 (Feature 与 Scenarios)	672
14.3.3	使用 Selenium IDE 来录制测试案例脚本	673
14.3.4	编写 Selenium 的测试程序 (完成 Steps.cs)	674
14.3.5	帮程序代码说话	677
14.3.6	提取方法	679
14.3.7	职责分离	680
14.3.8	针对各货运商类加入单元测试	682
14.3.9	依赖接口与工厂方法	683
14.3.10	小结	685
14.4	总结	685
第 15 章	网站部署	687
15.1	ASP.NET MVC RTM 的进化史	687
15.2	部署到 IIS	688
15.2.1	Web 部署工具	688
15.2.2	将 ASP.NET MVC 4 应用程序部署到 IIS 5.1 中	691
15.3	部署到 Microsoft Azure 云环境	692

15.3.1 部署到 Microsoft Azure Website	692
15.3.2 部署到 Microsoft Azure Cloud Service	697
15.4 结术语	699
第 16 章 CMS 范例实战	700
16.1 新建项目	700
16.2 创建数据	701
16.3 建立区域	706
16.4 使用 MetaData	708
16.5 使用 NuGet	709
16.6 建立会员机制	716
16.7 分页套用	716
16.8 结术语	719
附录 A ASP.NET MVC 5.1 与 ASP.NET MVC 5.2	720
附录 B ASP.NET MVC 6 预览	725



BOOKASK.COM

第 1 章

MVC 概述

- 1.1 MVC 架构概述
- 1.2 ASP.NET MVC 概述
- 1.3 ASP.NET MVC 5 开发环境
- 1.4 NuGet 的使用
- 1.5 IIS Express
- 1.6 ASP.NET Web Forms 转换至 MVC 开发
- 1.7 创建第一个 MVC 因特网应用程序
- 1.8 结术语

1.1 MVC 架构概述

在开始学习 MVC 之前应该要先了解 MVC，其实 MVC 就是 Model-View-Controller 三个单词的前缀缩写，中文翻译分别为模型、视图、控制器，但是这样的称呼不仅容易造成混淆，平常也很少使用，因此本书往后的章节将继续沿用 Model、View、Controller 的惯用称呼。

1.1.1 MVC 是什么

MVC 既不是程序设计语言也不是框架，而是一种设计模式。MVC 模式最早由 Trygve Reenskaug 在 1974 年提出的，是 Xerox PARC 实验室¹在 20 世纪 80 年代为 SmallTalk²程序设计语言发明的一种软件设计模式，经由将应用程序架构切分为 Model、View、Controller 三个部分的设计，可以让开发者分割出不同方面的应用程序（输入逻辑、商业逻辑、显示逻辑），并且在这些分割出来的应用程序中提供了松散的关联。在 ASP.NET MVC 中这些切割是强制性的，默认就切割了 Model、View、Controller 三部分，使得开发人员不得不将应用程序至少切割为 M-V-C 三层，这里说“至少”的原因是在 MVC 架构下这三层是必需的，但根据项目的需求与实际情况还可以再增加，比如实现

¹ Xerox PARC 实验室，是许多现代计算机技术的诞生地，对于目前信息产业有着很大的影响，如：个人计算机、激光打印机、鼠标、以太网络、图形用户界面。

² SmallTalk 是被公认为历史上第二个面向对象的程序设计语言，第一个则是 1960 年代由挪威奥斯陆的“挪威电子计算中心”所开发出来的 Simula。



本书试读到此结束啦！



[ASP.NET MVC 5网站开发之美](#)

作者：demo, 等编著

出版社：清华大学出版社

通过以下方式阅读更多 [Powered by 书问](#)



● 扫码分享到朋友圈可阅读更多



立即扫码



● 还不过瘾？购买书库畅读卡全本畅读此书！



[查看全部书库](#)



● 购买纸书也可畅读全本哦！

[中国图书网](#)

[蔚蓝网](#)

[云书网](#)