



ASP.NET MVC 5 框架揭秘

◎蒋金楠 著

ASP.NET MVC 5

框架揭秘

©蒋金楠 著

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

内 容 简 介

本书以一个模拟 ASP.NET MVC 内部运行机制的“迷你版 MVC 框架”作为开篇，其目的在于将 ASP.NET MVC 真实架构的“全景”勾勒出来。接下来本书以请求消息在 ASP.NET MVC 框架内部的流向为主线将相关的知识点串联起来，力求将“黑盒式”的消息处理管道清晰透明地展示在读者面前。相信精读本书的读者一定能够将 ASP.NET MVC 从接收请求到响应回复的整个流程了然于胸，对包括路由、Controller 的激活、Model 元数据的解析、Action 方法的选择与执行、参数的绑定与验证、过滤器的执行及 View 的呈现等相关机制具有深刻的理解。

本书以实例演示的方式介绍了很多与 ASP.NET MVC 相关的最佳实践，同时还提供了一系列实用性的扩展，相信它们一定能够解决你在真实开发过程中遇到的很多问题。本书最后一章提供的案例不仅用于演示实践中的 ASP.NET MVC，很多架构设计方面的东西也包含其中。除此之外，本书在很多章节还从设计的角度对 ASP.NET MVC 的架构进行了深入分析，所以从某种意义上讲本书可以当成一本架构设计的书来读。

虽然与市面上任何一本相关的书相比，本书走得更远，并更加近距离地触及 ASP.NET MVC 框架的内核，但是就其内容本身来讲却没有涉及太多“高深莫测”的知识点，所以阅读本书不存在太高的门槛。如果你觉得自己对 ASP.NET MVC 所知甚少，可以利用此书来系统地学习 ASP.NET MVC；如果你觉得自己对 ASP.NET MVC 足够精通，一定能够在此书中找到相应的“盲点”。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

ASP.NET MVC 5 框架揭秘 / 蒋金楠著. —北京：电子工业出版社，2014.8

ISBN 978-7-121-23781-2

I. ①A… II. ①蒋… 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（2014）第 150027 号

策划编辑：张春雨

责任编辑：徐津平

特约编辑：赵树刚

印 刷：北京市京科印刷有限公司

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：787×980 1/16 印张：41 字数：1050 千字

版 次：2014 年 8 月第 1 版

印 次：2014 年 8 月第 1 次印刷

定 价：99.00 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。

前言

ASP.NET MVC 是一个建立在 ASP.NET 平台上基于 MVC 模式的 Web 开发框架，它提供了一种与 Web Form 完全不同的开发方式。ASP.NET Web Form 借鉴了 Windows Form 基于控件和事件注册的编程模式，使 Web 应用的开发变得简单而快捷，但是它却使开发人员与 Web 的本质渐行渐远。ASP.NET MVC 是一种回归，它使开发人员可以真正地面向 Web 进行编程，我们面对的不再是拖拽到 Web 页面的控件，而是整个 HTTP 请求和响应的流程。

这不是一本传统意义上的入门书籍

在《ASP.NET MVC 4 框架揭秘》出版之后，读者通过多种渠道将他们对本书的看法、建议和意见反馈给笔者。有一些读者觉得本书对他们来说太“深”了，因为他们希望买到的是一本单纯的入门书籍，所以我不得不再次强调——“这不是一本传统意义上的入门书籍”。如果你之前根本没有接触过 ASP.NET MVC，并且希望有一本书能让你在一两个星期内掌握 ASP.NET MVC 的基本编程模式，那么笔者觉得选择本书并不是明智的。

笔者个人认为掌握 ASP.NET MVC 具有 3 个层次。了解基本的编程模式，掌握 Controller 和 View 的定义方式，知道路由如何注册及验证规则如何定义，此为第一层次。第二层次要求我们对 ASP.NET MVC 框架本身从请求接收到响应回复的整个流程有一个清晰的认识，这包括请求如何被路由、目标 Controller 如何被激活、Model 元数据如何被解析、Action 方法如何被执行、View 如何呈现等。ASP.NET MVC 本身是一个极具可扩展的开发框架，合理利用其扩展性可以解决很多开发中的实际问题，而掌握 ASP.NET MVC 的最高层次就是凭着对框架本身的运行机制的了解准确地找到相应的扩展点，并创建相应的扩展来解决我们遇到的问题。

本书不是一本 ASP.NET MVC 入门书籍，而是一本让处于第一层次的读者快速进入第二和第三层次的书。如果你具有进入更高层次的需求，并且有充裕的时间来阅读，笔者坚信你一定能够通过此书获得其他同类书籍难以提供的东西。

不少人觉得本书过于“深入”，但笔者个人宁愿说本书“走得更远”。虽然本书比其他同类书籍更加近距离地触摸到了 ASP.NET MVC 框架的内核，但就其内容本身来说，笔者认为本书并没有什

么“深不可测”的东西。尽管本书不是一本纯粹意义上的入门书籍，但是笔者绝不会根据读者个人的技术水平将其排除在本书的读者范围之外。如果读者具有基本的.NET 方面的知识储备，并且从事过简单的 Web 开发，阅读本书不会有太大的障碍。

这是一本讲述 ASP.NET MVC 框架本质的书

很多.NET 开发人员都在抱怨微软开发技术过快的更新频率让他们无所适从，其实他们看到的只是单纯的版本升级而已，一些本质的东西一直是“稳定”的。微软推出.NET 战略已经十多年了，CLR 却只有 4 个版本而已。最新版本的 ASP.NET 虽然表面上已经看不到太多最初的影子，但是整个请求处理的管道一直未曾改变。对于一项开发技术，只要我们了解了它最根本性的一些东西，不但不应该惧怕其高频率的版本更替，而应该热烈拥抱它。本书力求将关于 ASP.NET MVC 框架最根本的东西带给大家，而不是罗列一些简单的编程技巧。

这本书与《ASP .NET MVC 4 框架揭秘》有何不同

本书针对最新版本的 ASP.NET MVC ，所以 ASP.NET MVC 5 提供的新特性和对 ASP.NET MVC 4 现有特性的改变自然体现在本书之中。通过微软“官方”渠道发布的信息我们知道，“特性路由”和“认证过滤器”是 ASP.NET MVC 5 提供的两个新特性，我们在本书中会通过单独的章节（第 13 章“特性路由”）来对特性路由作详细讲解，针对认证过滤器的介绍则放到新增加的一章（第 12 章“过滤器”）中。

除了这些通过官方渠道了解到的改变之外，其实微软在 ASP.NET MVC 5 框架内部作了很多的改进，这些东西只有当我们深入去分析其源码时才有可能发现。这些“不为人知”的内容也包含在本书之中，如果你手中正好有一本《ASP.NET MVC 4 框架揭秘》，在阅读的时候对照一下就知道了。

本书除了反映 ASP.NET MVC 5 本身的改变之外，还对前版一些遗漏掉的知识点进行了补充。为了让读者更好地理解本书的内容并尽可能地迎合大多数人的思维方式，我们在写作过程中对整本书的结构作了调整，同时对一些表达方式作了相应的改变。

和大部分 ASP.NET MVC 的书籍一样，《ASP.NET MVC 4 框架揭秘》利用一个单独的章节来介绍 Web API，这种做法实际上给很多读者造成了一种误解，让他们觉得 ASP.NET Web API 仅仅是

ASP.NET MVC 的附属产品。实际上两者不但在运行上使用不同的消息处理管道，并且这两个管道在设计上也是截然不同的，所以本书中找不到任何关于 ASP.NET Web API 的内容。如果你希望深入地学习 ASP.NET Web API，可以考虑本书的姊妹篇《ASP.NET Web API 2 框架揭秘》。

这是一本实用的书吗？

可能有人觉得这本剖析 ASP.NET MVC 框架运行原理的书没有什么“实际”的意义，因为我们每天的日常工作就是编程，知道了 ASP.NET MVC 从请求接收到响应回复之间整个处理流程并不会对我们的工作造成实质性的改变，这是一种极端错误的想法。学习一种软件开发技术就像是练一门功夫，不仅要苦练攻敌招式，还得研习内功心法。编程模式是攻敌招式，框架背后的设计原理是内功心法，没有内功心法支撑的招式只能是花拳绣腿，而具有极高内功修为的高手却能“无招胜有招”。

除此之外，由于我们编写的程序最终是在 ASP.NET MVC 框架上运行的，程序的高效性决定了它是否能够最大限度地“迎合”框架的运行机制，所以了解 ASP.NET MVC 框架的运行原理有利于我们写出高质量的程序。

很多读者都问笔者为何有那么多时间和精力去深入学习多个不同的开发框架（比如 WCF、ASP.NET MVC 和 ASP.NET Web API 等），其实这和从事的工作有关。多年以来，笔者在公司所作的工作就是设计、开发和维护一套内部的开发框架，这个框架的根本目的在于让我们可以采用一种类似于流水线的项目开发方式，不仅可以提高开发效率，还能提高项目本身的质量并降低对最终开发人员的技能要求，实现的途径就是让最终的开发人员只关心具体业务功能的实现，所有非业务功能都由框架本身来完成。

为了完成这个使命，我们针对 .NET 平台上的开发技术作了很多扩展。为了能够精准地定位采用的扩展点，我们不得不对开发技术本身有一个深刻的认识，所以深入学习 .NET 平台主流开发技术是笔者的本质工作之一。为了将这些实用的扩展服务于大众，笔者将这些扩展放到了相关的书籍之中。笔者自然不会将我们做的东西直接共享出来（因为这是违法的），所以书中涉及的这些扩展都经过刻意简化，因为笔者希望的不是让读者直接使用这些扩展，而是根据其体现出来的原理来设计你们需要的扩展。

ASP.NET MVC 之外的一些东西

在笔者的周围存着这样的一些人，他们以刚毕业一两年的毕业生为主。他们大都工作勤奋、聪明好学，手中经常捧着 GoF 的《设计模式》，总是希望在将书中的设计模式应用到具体项目之中，或者希望通过项目的实践来印证他们在书本上的设计模式，但是理论和实践之间的距离总让他们感到困惑。

要从真实的项目或者产品中学习“实用”的软件架构设计知识，先得确定目标项目或者产品中采用的架构思想和设计模式是正确的，而我们参与的很多项目其实被“架构”得一塌糊涂。对于像 ASP.NET 这样的产品，其基础架构能够在很长一段时间内保持不变，本身就证明了应用在上面的架构设计的正确性，它们不正是我们学习架构设计最好的素材吗？本书对 ASP.NET MVC 框架的运行机制进行了深入剖析，实际上是将 ASP.NET MVC 的整个设计展示在读者面前，读者也许可以将本书作为一本“架构设计”的书来读。

很多读者向笔者咨询针对一种新的开发技术，如何才能更加有效地掌握其“精髓”。这是一个关于学习方法的问题，笔者个人采用的学习方法不可能适用于所有的人。尽管如此，本书也或多或少地体现了笔者个人采用的学习方法，所以在论述某个知识点的时候，不但会告诉读者“是什么”和“为什么”，还会告诉读者“如何证明是这样”。换句话说，笔者不仅仅将某个论点抛给你，还会为你展现整个论证的过程。

本书的写作特点

和其他同类书籍作者总是从“静态编程”作为切入点不同，笔者以“动态执行”的视角来审视 ASP.NET MVC 框架。本书以框架本身处理请求的流程为主线，力求将 ASP.NET MVC 框架从接收请求到回复响应的整个流程完整而清晰地展现在读者面前。在本书的第 1 章中，我们设计了一个“迷你版”的 ASP.NET MVC 框架，其目的在于让读者能够对 ASP.NET MVC 的执行管道有一个整体的认识，至于组成这个管道的每个环节，则通过后续的章节对其架构设计进行详细介绍，对其执行原理进行深入剖析。

本书具有一个与其他中文原创或者翻译书籍截然不同的特点，那就是几乎所有的术语都采用英文，比如 Controller、Action 和 Model 在本书中并没有翻译成中文“控制器”、“操作”和“模型”。

因为笔者认为很多术语其实很难找到一个语义完全等同的中文词组或短语与之对应，对于习惯了英文作为“开发语言”的读者来说，强行翻译其实是不必要的。

除此之外，这不是一本纯理论的书，而是一本“实证型”的书，在书中提供了一百多个可供单独下载的实例演示。这些实例在本书中具有不同的作用，有的是为了探测和证明对应的论点，有的是为了演示某种实用的编程技巧。

关于作者

蒋金楠（网名 Artech），《WCF 全面解析（上、下册）》、《ASP.NET MVC 4 框架揭秘》、《ASP.NET Web API2 框架揭秘》等多部畅销 IT 图书作者，现就职于一知名软件公司担任高级软件顾问。拥有一个访问量超过 200 万的个人博客（<http://www.cnblogs.com/artech>），2012 年度 51CTO IT 博客大赛十佳得主。2007—2013 年被连续 7 次评为微软 MVP，同时也是少数几个跨多领域（Solutions Architect、Connected System、Microsoft Integration 和 ASP.NET/IIS）的 MVP 之一。

致谢

本书得以出版，需要感谢博文视点的编辑，你们的专业水准和责任心是为本书提供的质量保证，多次合作产生的默契让我对下次合作充满期待。此外，最需要感谢的是我的老婆徐妍妍，只有我知道你在本书提交给出版社之前所作的校对工作有多么重要。

本书支持

本书针对最新版本的 ASP.NET MVC，同时涉及太多底层实现的内容，所以大部分内容是找不到任何现成参考资料的，这些内容大都来自于笔者对源码的分析和试验的证明。这些因素加上作者自身能力的限制，都可能造成一些无法避免的错误或者偏差，如果读者在阅读过程中发现了任何问题，希望能够加以反馈。如果读者遇到任何 ASP.NET MVC、ASP.NET Web API 或者 WCF 的问题，欢迎与我通过以下的方式进行交流。

VIII ■ 前 言

- 作者博客：<http://www.cnblogs.com/artech>
- 作者微博：<http://www.weibo.com/artech>
- 电子邮箱：jiangjinnan@gmail.com
- 微信公众账号：Artech1981

本书每一章节都会提供一系列实例演示，读者可以根据编号（比如 S101、S202 等）从下载的源代码压缩包中找到对应的实例。本书的附录部分给出了所有源代码可供下载的所有实例演示的列表和相关描述。

源代码下载地址：<https://onedrive.live.com/redir?resid=5760EBEEB92818D2%21109>

目 录

第 1 章 ASP.NET + MVC	1
1.1 传统 MVC 模式	2
1.1.1 自治视图	2
1.1.2 什么是 MVC 模式	3
1.2 MVC 的变体	4
1.2.1 MVP	5
1.2.2 Model 2	13
1.2.3 ASP.NET MVC 与 Model 2	15
1.3 IIS/ASP.NET 管道	15
1.3.1 IIS 5.x 与 ASP.NET	16
1.3.2 IIS 6.0 与 ASP.NET	17
1.3.3 IIS 7.0 与 ASP.NET	19
1.3.4 ASP.NET 集成	20
1.3.5 ASP.NET 管道	22
1.4 ASP.NET MVC 是如何运行的	28
1.4.1 建立在“迷你版”ASP.NET MVC 上的 Web 应用	28
1.4.2 路由	31
1.4.3 Controller 的激活	37
1.4.4 Action 的执行	41
1.4.5 完整的流程	49
第 2 章 路由	51
2.1 ASP.NET 路由	52
2.1.1 请求 URL 与物理文件的分离	52
2.1.2 实例演示：通过路由实现请求地址与.aspx 页面的映射（S201）	53
2.1.3 Route 与 RouteTable	57

2.1.4	路由注册	65
2.1.5	根据路由规则生成 URL	77
2.2	ASP.NET MVC 路由	79
2.2.1	路由映射	79
2.2.2	路由注册 (S210)	80
2.2.3	缺省 URL 参数	83
2.2.4	基于 Area 的路由映射	85
2.2.5	链接和 URL 的生成	91
2.3	动态 HttpHandler 映射	98
2.3.1	UrlRoutingModule	99
2.3.2	PageRouteHandler 与 MvcRouteHandler	100
2.3.3	ASP.NET 路由系统扩展	101
2.3.4	实例演示：通过自定义 Route 对 ASP.NET 路由系统进行扩展 (S214)	102
第 3 章	Controller 的激活	107
3.1	Controller 激活系统全景展示	108
3.1.1	Controller	108
3.1.2	ControllerFactory	114
3.1.3	ControllerBuilder	115
3.1.4	Controller 的激活与路由	122
3.2	Controller 默认激活机制	125
3.2.1	Controller 类型的解析	125
3.2.2	Controller 类型的缓存	130
3.2.3	Controller 的释放和会话状态行为的控制	131
3.3	IoC 的应用	133
3.3.1	从 Unity 来认识 IoC	133
3.3.2	Controller 与 Model 的解耦	135
3.3.3	基于 IoC 的 ControllerFactory	137
3.3.4	基于 IoC 的 ControllerActivator	143
3.3.5	基于 IoC 的 DependencyResolver	145
第 4 章	Model 元数据的解析	149
4.1	Model 元数据	150

4.1.1	Model 元数据层次化结构	150
4.1.2	Model 元数据的定制	154
4.1.3	IMetadataAware 接口	171
4.2	模板化数据的呈现	176
4.2.1	实例演示：通过模板将布尔值显示为 RadioButton (S409)	176
4.2.2	预定义模板	178
4.2.3	针对数据类型的模板	185
4.2.4	数据类型名称 V.S. 模板名称	189
4.2.5	模板的获取与执行	193
4.2.6	实例演示：通过定制 Model 元数据和自定义模板 实现预定义列表的 呈现 (S412)	197
4.3	Model 元数据的提供机制	205
4.3.1	再谈 ModelMetadata	206
4.3.2	ModelMetadataProvider	210
4.3.3	Model 元数据提供系统的扩展	214
第 5 章	3 个重要的描述对象	217
5.1	ControllerDescriptor	218
5.1.1	ReflectedControllerDescriptor	219
5.1.2	ReflectedAsyncControllerDescriptor	228
5.2	ActionDescriptor	229
5.2.1	AsyncActionDescriptor	230
5.2.2	ReflectedActionDescriptor	231
5.2.3	ReflectedAsyncActionDescriptor	232
5.2.4	TaskAsyncActionDescriptor	233
5.3	ParameterDescriptor	234
第 6 章	Model 的绑定 (一)	236
6.1	源数据的提供	237
6.1.1	NameValueCollectionValueProvider	238
6.1.2	DictionaryValueProvider	246
6.1.3	ValueProviderFactory	254
6.1.4	ValueProviderFactories	255
6.2	ModelBinder 及其提供策略	259

6.2.1	ModelBinder	259
6.2.2	ModelBinderProvider	265
6.2.3	ModelBinders.....	268
6.2.4	CustomModelBinderAttribute	271
6.2.5	针对参数的 ModelBinder 是如何创建的	274
6.3	Model 绑定的实施.....	276
6.3.1	绑定上下文的初始化.....	277
6.3.2	绑定过程中对 ModelState 的设置	279
第 7 章	Model 的绑定 (二)	283
7.1	绑定简单对象	284
7.1.1	利用 ValueProvider 绑定简单对象.....	284
7.1.2	实例演示: 利用 MyDefaultModelBinder 绑定简单类型参数 (S701)	286
7.2	绑定复杂对象	289
7.2.1	复杂对象层次化结构.....	289
7.2.2	递归式绑定.....	291
7.2.3	实例演示: 利用 MyDefaultModelBinder 绑定复杂类型参数 (S702、S703)	294
7.3	绑定集合	297
7.3.1	针对同名数据项的集合绑定.....	297
7.3.2	针对索引的集合绑定.....	302
7.4	绑定字典	311
7.4.1	字典是一个复杂类型的集合.....	311
7.4.2	针对字典类型的 Model 绑定策略	312
7.4.3	实例演示: 利用 MyDefaultModelBinder 绑定字典类型参数 (S707)	316
第 8 章	Model 的验证 (一)	318
8.1	几种参数验证方式	319
8.1.1	ModelError.....	319
8.1.2	验证消息的呈现.....	320
8.1.3	手工验证绑定的参数.....	322
8.1.4	使用 ValidationAttribute 特性.....	327
8.1.5	让数据类型实现 IValidatableObject 接口.....	330
8.1.6	让数据类型实现 IDataErrorInfo 接口.....	332

8.2	ModelValidator 及其提供策略	334
8.2.1	ModelValidator 与 ModelValidatorProvider	334
8.2.2	DataAnnotationsModelValidator	337
8.2.3	ValidatableObjectAdapter	337
8.2.4	DataErrorInfoModelValidator	338
8.2.5	ClientModelValidator	339
8.2.6	CompositeModelValidator	341
8.3	Model 验证的实施	345
8.3.1	Model 绑定过程中的验证	346
8.3.2	实例演示: 模拟 Model 绑定中的验证 (S810)	347
8.3.3	针对“必需”数据成员的验证	351
第 9 章	Model 的验证 (二)	354
9.1	ValidationAttribute 特性	355
9.1.1	数据是如何被验证的	356
9.1.2	几个常用的 ValidationAttribute	358
9.1.3	应用 ValidationAttribute 特性的唯一性	360
9.2	DataAnnotationsModelValidator 及其提供策略	364
9.2.1	“适配”型 DataAnnotationsModelValidator	365
9.2.2	DataAnnotationsModelValidatorProvider	368
9.2.3	将 ValidationAttribute 特性应用到参数上	375
9.2.4	一种 Model 类型, 多种验证规则	382
9.3	客户端验证	389
9.3.1	jQuery 验证	390
9.3.2	基于 jQuery 的 Model 验证	394
9.3.3	自定义验证	398
第 10 章	Action 方法的执行	402
10.1	异步 Action 的定义	403
10.1.1	基于线程池的请求处理机制	403
10.1.2	两种异步 Action 方法的定义	404
10.1.3	AsyncManager	406
10.2	各种同步与异步组件	412
10.2.1	MvcHandler	412

10.2.2	Controller	413
10.2.3	ActionInvoker	414
10.2.4	ControllerDescriptor	420
10.2.5	ActionDescriptor	423
10.3	目标方法的执行	430
10.3.1	Action 方法并不以“反射”方式执行	430
10.3.2	实例演示：采用针对表达式树执行 Action 方法（S1010）	432
第 11 章	View 的呈现	437
11.1	ActionResult	438
11.1.1	EmptyResult	438
11.1.2	ContentResult	439
11.1.3	FileResult	446
11.1.4	JavaScriptResult	451
11.1.5	JsonResult	455
11.1.6	HttpStatusCodeResult	457
11.1.7	RedirectResult/RedirectToRouteResult	458
11.2	ViewResult 与 ViewEngine	461
11.2.1	View 引擎中的 View	461
11.2.2	ViewEngine	463
11.2.3	ViewResult 的执行	465
11.3	Razor 引擎	474
11.3.1	View 的编译原理	475
11.3.2	WebViewPage 与 WebViewPage<TModel>	480
11.3.3	RazorView	485
11.3.4	RazorViewEngine	495
第 12 章	过滤器	499
12.1	Filter 及其提供机制	500
12.1.1	Filter 与 FilterProvider	500
12.1.2	以特性方式注册过滤器	502
12.1.3	Controller 本身就是过滤器	504
12.1.4	过滤器的全局注册	504

12.1.5	实例演示：验证 Filter 的提供机制和执行顺序 (S1201, S1202, S1203)	506
12.2	AuthenticationFilter	511
12.2.1	AuthenticationFilter 的执行流程	512
12.2.2	实例演示：通过自定义 AuthenticationFilter 实现 Basic 认证 (S1204)	513
12.3	AuthorizationFilter	518
12.3.1	AuthorizeAttribute	518
12.3.2	RequireHttpsAttribute	520
12.3.3	ValidateInputAttribute	520
12.3.4	ValidateAntiForgeryTokenAttribute	523
12.3.5	ChildActionOnlyAttribute	527
12.4	ActionFilter	528
12.4.1	ActionFilter 的执行流程	529
12.4.2	ActionFilter 对 ActionResult 的设置	530
12.4.3	异常处理	532
12.5	ExceptionHandler	534
12.5.1	HandleErrorAttribute	535
12.5.2	实例演示：利用自定义的 ExceptionFilter 集成 Enterprise Library 进行 异常处理 (S1207, S1208, S1209)	537
12.6	ResultFilter 与 OverrideFilter	551
12.6.1	ResultFilter 的执行流程	552
12.6.2	屏蔽外围过滤器	553
第 13 章	特性路由	556
13.1	特性路由注册	557
13.1.1	RouteInfoProvider 特性	557
13.1.2	基本路由映射	558
13.1.3	让路由模板能够尽可能反映资源的层次结构	559
13.1.4	为路由变量设置约束	560
13.1.5	缺省路由变量	561
13.1.6	设置模板前缀	562
13.1.7	设置 Area 名称	563
13.2	约束表达式的解析	564
13.2.1	RangeRouteConstraint	565

13.2.2	InlineConstraintResolver	566
13.2.3	自定义约束.....	570
13.3	Route 的创建.....	574
13.3.1	特性路由注册的 Route 对象	574
13.3.2	Route 的生成机制.....	579
13.3.3	Controller 的激活与 Action 方法的选择	579
第 14 章	案例实践	581
14.1	功能简介	582
14.1.1	商品列表的呈现.....	582
14.1.2	订购商品.....	584
14.1.3	登录与错误页面.....	585
14.2	设计概述	586
14.2.1	Controller-Service-Repository	586
14.2.2	IoC 的应用.....	591
14.2.3	AOP 的应用.....	594
14.2.4	异常处理.....	601
14.3	编程实现	602
14.3.1	数据表的创建.....	603
14.3.2	Repository	604
14.3.3	Service	609
14.3.4	路由注册和布局.....	612
14.3.5	ProductController.....	616
14.3.6	OrderController.....	624
14.3.7	AccountController.....	630
附录 A	实例列表	635