

Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™



Professional **ASP.NET MVC 2**

Jon Galloway, Phil Haack, Scott Hanselman, Scott Guthrie, Rob Conery

PROFESSIONAL ASP.NET MVC 2

FOREWORD	xxv
INTRODUCTION	xxvii
CHAPTER 1 NerdDinner	1
CHAPTER 2 Model-View-Controller and ASP.NET	167
CHAPTER 3 ASP.NET > ASP.NET MVC	179
CHAPTER 4 Routes and URLs	203
CHAPTER 5 Controllers	243
CHAPTER 6 Views	279
CHAPTER 7 AJAX	311
CHAPTER 8 Filters	341
CHAPTER 9 Securing Your Application	363
CHAPTER 10 Test Driven Development with ASP.NET MVC	395
CHAPTER 11 Testable Design Patterns	413
CHAPTER 12 Best of Both Worlds: Web Forms and MVC Together	441
CHAPTER 13 What's New in ASP.NET MVC 2	471
INDEX	489

PROFESSIONAL
ASP.NET MVC 2

PROFESSIONAL
ASP.NET MVC 2

Jon Galloway
Phil Haack
Scott Hanselman
Scott Guthrie
Rob Conery



WILEY

Wiley Publishing, Inc.

Professional ASP.NET MVC 2

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-64318-1

ISBN: 978-0-470-90807-5 (ebk)

ISBN: 978-0-470-90861-7 (ebk)

ISBN: 978-0-470-90862-4 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2010926831

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

To my wife Rachel, my daughters Rosemary, Esther, and Ellie, and the rest of my family for their patience and support during the whole “move, change jobs, and write a book in the same year” thing.

—JON GALLOWAY

My wife, Akumi, deserves to have her name on the cover as much as I do for all her support made this possible. And thanks to Cody for his infectious happiness.

—PHIL HAACK

ABOUT THE AUTHORS



JON GALLOWAY works at Microsoft as an ASP.NET Community Program Manager. Jon previously worked at Vertigo Software, where he led teams building Microsoft conference websites and worked on some high profile Silverlight projects. Prior to that, he worked in a wide range of web development shops, from scrappy startups to Fortune 500 financial companies, and even spent some time as a submarine lieutenant. He's an ASP.NET and Silverlight Insider, former ASP.NET MVP, and regular contributor to several open source .NET projects. He's part of the Herding Code podcast (<http://herdingcode.com>), blogs at <http://weblogs.asp.net/jgalloway>, and twitters as @jongalloway. He lives in San Diego with his wife, three daughters, and a bunch of avocado trees.



PHIL HAACK is a Senior Program Manager with the ASP.NET team working on the ASP.NET MVC project. Prior to joining Microsoft, Phil worked as a product manager for a code search engine, a dev manager for an online gaming company, and as a senior architect for a popular Spanish language television network, among other crazy pursuits. As a code junkie, Phil Haack loves to craft software. Not only does he enjoy writing software, he enjoys writing about software and software management on his blog, <http://haacked.com/>. In his spare time, Phil contributes to various open source projects and is the founder of the Subtext blog engine project, which is undergoing a re-write, using ASP.NET MVC, of course.



SCOTT HANSELMAN works for Microsoft as a Principal Program Manager Lead in Server and Tools Online, aiming to spread the good word about developing software, most often on the Microsoft stack. Before this, he worked in eFinance for 6+ years and before that he was a Principal Consultant at Microsoft Partner for nearly 7 years. He was also involved in a few things like the MVP and RD programs and will speak about computers (and other passions) whenever someone will listen to him. He blogs at <http://www.hanselman.com> and podcasts at <http://www.hanselminutes.com> and contributes to sites like <http://www.asp.net>, <http://www.windowsclient.net>, and <http://www.silverlight.net>. You can also find him on twitter at @shanselman, far too often.



SCOTT GUTHRIE is corporate vice president of Microsoft's .NET Developer Division, where he runs the development teams responsible for delivering Microsoft Visual Studio developer tools and Microsoft .NET Framework technologies for building client and Web applications. A founding member of the .NET project, Guthrie has played a key role in the design and development of Visual Studio and the .NET Framework since 1999. Guthrie is also responsible for Microsoft's Web server platform and development tools teams. He has also more recently driven the development of Silverlight — a cross browser, cross platform plug-in for delivering next generation media experiences and rich internet applications for the Web. Today, Guthrie directly manages the development teams that build the Common Language Runtime

(CLR), ASP.NET, Silverlight, Windows Presentation Foundation (WPF), IIS, Commerce Server and the Visual Studio Tools for Web, Client and Silverlight development. Guthrie graduated with a degree in computer science from Duke University.



ROB CONERY is an expert in ASP.NET. He is the Creator of SubSonic and was the Chief Architect of the Commerce Starter Kit (a free, Open Source eCommerce platform for .NET). He lives in Kauai, HI with his wife and two daughters (Maddy and Ruby).

ABOUT THE TECHNICAL EDITORS

BRAD WILSON has been a professional software developer for nearly 20 years, which includes more than 10 years of web development. He joined Microsoft in 2005, joined the ASP.NET team in 2008, and has worked on the ASP.NET MVC project since 2009.

EILON LIPTON joined the ASP.NET team as a developer at Microsoft in 2002. On this team, he has worked on areas ranging from data source controls to localization to the UpdatePanel control. He now works on the ASP.NET MVC Framework as a senior development lead. Eilon is also a frequent speaker on a variety of ASP.NET-related topics at conferences worldwide. He graduated from Boston University with a dual degree in Math and Computer Science. In his spare time Eilon spends time in his garage workshop building what he considers to be well-designed furniture. If you know anyone who needs a coffee table that's three feet tall and has a slight slope to it, send him an e-mail.

CREDITS

ACQUISITIONS EDITOR

Paul Reese

PROJECT EDITOR

Maureen Spears

TECHNICAL EDITORS

Brad Wilson

Eilon Lipton

PRODUCTION EDITOR

Daniel Scribner

COPY EDITOR

Cate Caffrey

EDITORIAL DIRECTOR

Robyn B. Siesky

EDITORIAL MANAGER

Mary Beth Wakefield

ASSOCIATE DIRECTOR OF MARKETING

David Mayhew

PRODUCTION MANAGER

Tim Tate

VICE PRESIDENT AND EXECUTIVE**GROUP PUBLISHER**

Richard Swadley

VICE PRESIDENT AND EXECUTIVE PUBLISHER

Barry Pruett

ASSOCIATE PUBLISHER

Jim Minatel

PROJECT COORDINATOR, COVER

Lynsey Stanford

COMPOSITOR

Jeff Lytle, Happenstance Type-O-Rama

PROOFREADER

Publication Services, Inc.

INDEXER

Robert Swanson

COVER DESIGNER

Michael E. Trent

COVER IMAGE

© David Madison / Photographer's
Choice RF / Getty Images

ACKNOWLEDGMENTS

THANKS TO FAMILY and friends who graciously acted as if “Jon without sleep” is someone you’d want to spend time with. Thanks to Scott Hanselman for believing in me. Thanks to Scott Stanfield and all my friends at Vertigo for just being awesome. Thanks to the whole ASP.NET team for making work fun since 2002, and especially to Brad Wilson and Phil Haack for answering tons of random questions. Thanks to Warren G. Harding for normalcy. Thanks to Philippians 4:4-9 for continually reminding me which way is up.

—JON GALLOWAY

THANKS GO TO my lovely wife Akumi for her support which went above and beyond all expectations and made this possible. I’d like to also give a shout out to my son Cody for his sage advice, delivered only as a two year old can deliver it. I’m sure he’ll be embarrassed ten years from now that I used such an anachronism (“shout out”) in my acknowledgment to him. Thanks go to my daughter Mia as her smile lights up the room like unicorns.

—PHIL HAACK

CONTENTS

FOREWORD	xxx
INTRODUCTION	xxvii
CHAPTER 1: NERDDINNER	1
<hr/>	
File ⇄ New Project	6
Examining the NerdDinner Directory Structure	8
Running the NerdDinner Application	9
Testing the NerdDinner Application	12
Creating the Database	14
Creating a New SQL Server Express Database	14
Creating Tables within Our Database	15
Setting up a Foreign Key Relationship between Tables	17
Adding Data to Our Tables	20
Building the Model	21
Entity Framework	21
Adding Entity Framework Classes to Our Project	21
Creating Data Model Classes with Entity Framework	25
NerdDinnerEntities Class	27
Creating a DinnerRepository Class	28
Retrieving, Updating, Inserting, and Deleting Using the DinnerRepository Class	31
Integrating Validation and Business Rule Logic with Model Classes	33
Controllers and Views	37
Adding a DinnersController Controller	37
Adding Index and Details Action Methods to the DinnersController Class	38
Understanding ASP.NET MVC Routing	40
Using the DinnerRepository from Our DinnersController	41
Using Views with Our Controller	42
Implementing the “NotFound” View Template	44
Implementing the “Details” View Template	46
Implementing the “Index” View Template	51
Convention-Based Naming and the \Views Directory Structure	56
Create, Update, Delete Form Scenarios	57
URLs Handled by DinnersController	57
Implementing the HTTP-GET Edit Action Method	58
Html.BeginForm and Html.TextBoxFor Html Helper Methods	62
Implementing the HTTP-POST Edit Action Method	64

Handling Edit Errors	67
Understanding ModelState and the Validation HTML Helper Methods	69
Complete Edit Action Method Implementations	71
Implementing the HTTP-GET Create Action Method	71
Implementing the HTTP-POST Create Action Method	74
Implementing the HTTP-GET Delete Action Method	77
Implementing the HTTP-POST Delete Action Method	79
Model Binding Security	81
CRUD Wrap-Up	83
ViewData and ViewModel	85
Passing Data from Controllers to View Templates	85
Using the ViewData Dictionary	86
Using a ViewModel Pattern	88
Custom-Shaped ViewModel Classes	91
Partials and Master Pages	91
Revisiting Our Edit and Create View Templates	92
Using Partial View Templates	93
Using Partial View Templates to Clarify Code	96
Master Pages	97
Paging Support	100
Index Action Method Recap	101
Understanding IQueryable<T>	101
Adding a <i>page</i> Value to the URL	103
Adding Page Navigation UI	105
Authentication and Authorization	109
Understanding Authentication and Authorization	109
Forms Authentication and the AccountController	110
Authorizing the /Dinners/Create URL Using the [Authorize] Filter	113
Using the User.Identity.Name Property When Creating Dinners	114
Using the User.Identity.Name Property When Editing Dinners	115
Showing/Hiding Edit and Delete Links	117
AJAX Enabling RSVPs Accepts	118
Indicating Whether the User Is RSVP'ed	118
Implementing the Register Action Method	121
Calling the Register Action Method Using AJAX	122
Adding a jQuery Animation	124
Cleanup — Refactor Out a RSVP Partial View	126
Integrating an AJAX Map	127
Creating a Map Partial View	127
Creating a Map.js Utility Library	128
Integrating the Map with Create and Edit Forms	130
Integrating the Map with the Details View	134
Implementing Location Search in Our Database and Repository	135

Implementing a JSON-Based AJAX Search Action Method	141
Calling the JSON-Based AJAX Method Using jQuery	142
Unit Testing	146
Why Unit Test?	146
NerdDinner.Tests Project	147
Creating Unit Tests for Our Dinner Model Class	147
Running Tests	150
Creating DinnersController Unit Tests	151
Dependency Injection	153
Extracting an IDinnerRepository Interface	153
Updating DinnersController to Support Constructor Injection	155
Creating the FakeDinnerRepository Class	156
Using the FakeDinnerRepository with Unit Tests	158
Creating Edit Action Unit Tests	160
Mocking the User.Identity.Name Property	161
Testing UpdateModel Scenarios	163
Testing Wrap-Up	164
NerdDinner Wrap-Up	165
CHAPTER 2: MODEL-VIEW-CONTROLLER AND ASP.NET	167
<hr/>	
What Is Model-View-Controller?	167
MVC on the Web Today	169
Ruby on Rails	170
Django and Python	171
Spring, Struts, and Java	171
Zend Framework and PHP	172
MonoRail	172
ASP.NET MVC: The New Kid on the Block	173
Serving Methods, Not Files	174
Is This Web Forms 4.0?	174
Why Not Web Forms?	174
Cost/Benefit of Web Forms	175
Should You Fear ASP.NET MVC?	176
Summary	177
CHAPTER 3: ASP.NET > ASP.NET MVC	179
<hr/>	
Abstraction: What Web Forms Does Well	179
A Basic Web Forms Application	180
The Importance of Events	184
The Leak: Where Web Forms Doesn't Exactly Fit	185
ViewState	187
Controlling Your Angle Brackets	187

Client IDs	187
Testing	188
Back to Basics: ASP.NET MVC Believes...	188
Orchestration versus Composing	189
Separation of Concerns: What It Means	189
Approaches to Maintainability	190
Caring about Testability	190
Common Reactions to ASP.NET MVC	192
This Looks Like Classic ASP from 1999!	192
Who Moved My <asp:Cheese runat="server">?>	192
Yet Another Web Framework	192
Why “(ASP.NET > ASP.NET MVC) == True”	193
Convention over Configuration	195
Your First, er, Third, Request	198
The Request Life Cycle	200
Summary	201
CHAPTER 4: ROUTES AND URLS	203
<hr/>	
Introduction to Routing	204
Compared to URL Rewriting	205
Defining Routes	205
Named Routes	212
Areas	213
Catch-All Parameter	214
StopRoutingHandler and IgnoreRoute	215
Debugging Routes	216
Under the Hood: How Routes Generate URLs	218
Under the Hood: How Routes Tie Your URL to an Action	226
The High-Level Request Routing Pipeline	226
Route Matching	227
Advanced Routing with Custom Constraints	227
Route Extensibility	228
Using Routing with Web Forms	235
Routing With Web Forms in ASP.NET 4	235
Routing with Web Forms in ASP.NET 3.5	237
Summary	241
CHAPTER 5: CONTROLLERS	243
<hr/>	
History of the Controller	243
Defining the Controller: The IController Interface	245
The ControllerBase Abstract Base Class	247

The Controller Class and Actions	248
Action Methods	248
The ActionResult	252
Action Result Types	254
Action Result Helper Methods	257
Implicit Action Results	258
Action Invoker	260
How an Action Is Mapped to a Method	260
Mapping Parameters	264
Invoking Actions	264
Using Asynchronous Controller Actions	265
Passing Data to Actions: The Model Binders	273
A Word about User Input	276
Summary	278
CHAPTER 6: VIEWS	279
<hr/>	
What a View Does	279
What a View Shouldn't Do	281
Specifying a View	281
Strongly Typed Views	283
ViewModels	284
HTML Helper Methods	285
HtmlHelper Class and Extension Methods	285
Using the HTML Helpers	286
HTML Template Customization	299
The View Engine	299
Configuring a View Engine	300
Finding a View	300
The View Itself	301
Alternative View Engines	302
New View Engine or New ActionResult?	309
Summary	309
CHAPTER 7: AJAX	311
<hr/>	
When Ajax Is Cool	312
When It's Not	313
Ajax Examples	314
Handling Disabled Scripting	315
Using Partial Views for Rendering	319
Some Things You May Not Know about Microsoft ASP.NET Ajax	323
Updating an HTML Element When Submitting a Form	325
The AutoComplete TextBox	327

Implementing AutoComplete with Microsoft ASP.NET Ajax	328
Filtering Data with a Selectbox	331
The Modal Pop-Up with jQuery	332
The Modal Pop-Up Code	333
The Rating Control	335
Summary	339
CHAPTER 8: FILTERS	341
<hr/>	
Filters Included with ASP.NET MVC	341
Authorize	342
RequireHttps	344
OutputCache	345
Exception Filter	347
Custom Filters	349
Writing a Custom Action Filter	354
Writing a Custom Authorization Filter	355
Writing a Custom Exception Filter	357
Filter Ordering	358
Filter Naming	359
Summary	361
CHAPTER 9: SECURING YOUR APPLICATION	363
<hr/>	
This Is a War	365
What Do Hackers Want?	366
How Do They Get It?	367
Weapons	369
Spam	369
Case Study: Profiting from Evil with the Srizbi and Storm Botnets	370
Digital Stealth Ninja Network	372
Threat: Cross-Site Scripting (XSS)	372
Threat Summary	372
Preventing XSS	377
Threat: Cross-Site Request Forgery	381
Threat Summary	382
Preventing CSRF Attacks	384
Threat: Cookie Stealing	386
Threat Summary	386
Preventing Cookie Theft with HttpOnly	387
Threat: Over-Posting	387
Threat Summary	388
Preventing Over-Posting with the Bind Attribute	388
Keeping Your Pants Up: Proper Error Reporting and the Stack Trace	389

Securing Your Controllers, Not Your Routes	390
Using [Authorize] to Lock Down Your Action or Controller	390
Using [NonAction] to Protect Public Methods	391
Summary: It's Up to You	392
CHAPTER 10: TEST DRIVEN DEVELOPMENT WITH ASP.NET MVC	395
<hr/>	
A Brief Introduction to TDD	396
What Does TDD Look Like?	396
Writing Good Unit Tests	399
What Are Some Benefits of Writing Tests?	403
How Do I Get Started?	403
Applying TDD to ASP.NET MVC	404
Testing Routes	404
Testing Controllers	406
Redirecting to Another Action	407
Testing View Helpers	408
Testing Views	412
Summary	412
CHAPTER 11: TESTABLE DESIGN PATTERNS	413
<hr/>	
Why You Should Care About Testability	413
Big Design Up Front (BDUF)	414
Agile Software Development	415
You Want to Write Testable Code	416
Using Tests to Prove You're Done	417
Designing Your Application for Testability	417
Future-Proofing Your Application with Interfaces	418
The Single Responsibility Principle	419
Avoid Using Singletons and Static Methods	419
Testable Data Access	422
Creating the Model	423
The Repository Pattern in Detail	425
Implementing Business Logic with the Service Layer	429
Services Gone Wild	432
Partial Solution: Setting Controller Dependencies Manually	432
Summary	438
CHAPTER 12: BEST OF BOTH WORLDS: WEB FORMS AND MVC TOGETHER	441
<hr/>	
How Is It Possible?	442
Including MVC in Existing Web Forms Applications	442
Step 1: Referencing the Required Libraries	442

Step 2: Creating the Necessary Directories	443
Step 3: Updating the Web.config	444
Adding Web Forms to an Existing ASP.NET MVC Application	448
The Easy Part: Do Nothing	448
When/Home/Isn't/Home/	450
Using System.Web.Routing to Route to Web Forms	450
Sharing Data between Web Forms and MVC	451
Using Route Parameters	451
Using HTTP-POST	452
Using the ASP.NET Session	453
Using Cross-Page Posting	454
Using TempData	455
Migrating from Web Forms to MVC	456
Step 1: Create an Empty ASP.NET MVC Project with a Test Project	457
Step 2: Implement the Structure	457
Step 3: Add Images and Styling	460
Step 4: Set Up Routing and Controllers	461
Step 5: Replace Complex Server Controls	465
Step 6: Upload Files and Work with Images	468
Summary	469
CHAPTER 13: WHAT'S NEW IN ASP.NET MVC 2	471
<hr/>	
Security	471
HTML Encoding Code Blocks	471
JsonResult	472
Productivity	473
Strongly Typed HTML Helpers	474
Templated Helpers	475
Model Validation	478
Client-Side Validation	480
Model Metadata	482
Html.Action and Html.RenderAction	483
Code Snippets in Views	483
Organize Projects with Areas	484
Performance: Asynchronous Controller Actions	486
Miscellaneous	486
Summary	487
INDEX	489

FOREWORD

THE ASP.NET WEB DEVELOPMENT PLATFORM continues to grow and is used by millions of developers worldwide. A vibrant community supports the platform, both online and offline; the online forums at <http://www.asp.net> average 2000 questions and answers a day.

ASP.NET and ASP.NET MVC 2 power news sites, online retail stores and, perhaps, your favorite social networking site. Your local sports team, book club or blog uses ASP.NET MVC 2 as well.

When it was introduced, ASP.NET MVC broke a lot of ground. Although the pattern was old, it was new to much of the existing ASP.NET community; it walked a delicate line between productivity and control, power and flexibility. Today, to me, ASP.NET MVC 2 represents choice — your choice of language, your choice of frameworks, your choice of open source libraries, your choice of patterns. Everything is pluggable. It epitomizes absolute control of my environment — if you like something, use it; if you don't like something, change it. I unit test how I want, create components as I want, and use my choice of JavaScript framework.

ASP.NET MVC 2 brings you model validation, strongly typed HTML helpers, client side validation, areas, new action filters, asynchronous controllers, model and editor templates, scaffolding UI based on the T4 code generation engine, and improved Visual Studio tooling support.

Phil, Eilon, and the ASP.NET MVC team have created version 2 of their amazing framework and given us the source. I encourage you to visit <http://www.asp.net/mvc> for fresh content, new samples, videos and tutorials.

With new chapters, updated prose, and fresh samples from Jon Galloway and Phil Haack, we all hope this book represents the next step for you in your mastery of ASP.NET MVC 2.

Enjoy!

—SCOTT HANSELMAN
*Principal Program Manager Lead
Server and Tools Online-Community
Microsoft Corporation*

INTRODUCTION

WHY DOES THE WORLD NEED yet another web framework?

This is the question that is most likely on your mind—or perhaps it’s what you were thinking when you saw this book sitting on the shelf. We each asked ourselves this many times over the last few years.

Indeed there are many frameworks out there today flavored with every buzzword the industry can think of. In short: it’s easy to be skeptical. Yet as we, the authors, delve deeper into the latest and greatest web framework, we’re each starting to realize just how far the industry has come in the last 10 years.

Rob began programming for the web with Classic ASP in 1997 and was giddy with excitement. When .NET came out he remembers running around his office, stopping everyone from working and explaining that the world just tilted on its axis.

We all feel the same way about ASP.NET MVC 2. Not because it’s “something different,” but because it offers developers the ultimate chance to “do it their way.” You don’t like the way the platform renders the View? Change it! Just about every part of the ASP.NET MVC 2 Framework is “swappable” — if the shoes pinch, get different shoes. Don’t like ties? Why not a bow-tie? You’re totally in control.

ASP.NET MVC 2 is a web framework that comes with a bunch of conventions to make your life easier when you follow them, but if you don’t want them, the Framework is quick to step out of your way so you can get your work done in the way you like it done.

This book is going to go into the “out of the box” experience you’ll have with ASP.NET MVC 2, but more importantly we’ll learn practical ways that you can extend ASP.NET MVC 2 with your own magic — then hopefully share that magic with others.

Because of this extensibility and attention to “doing it your way,” we’re happy to embrace Yet Another Web Framework and hope you are willing to come along with us for the ride.

WHO THIS BOOK IS FOR

This book is for web developers who are looking to add more complete testing to their web sites, and who are perhaps ready for “something different.”

In some places, we assume that you’re somewhat familiar with ASP.NET WebForms, at least peripherally. There are a lot of ASP.NET WebForms developers out there who are interested in ASP.NET MVC so there are a number of places in this book where we contrast the two technologies. Even if you’re not already an ASP.NET developer, you might still find these sections interesting for context, as well as for your own edification as ASP.NET MVC 2 may not be the web technology that you’re looking for.

It’s worth noting, yet again, that ASP.NET MVC 2 is not a replacement for ASP.NET Web Forms. Many web developers have been giving a lot of attention to other web frameworks out there (Ruby

on Rails, Django) which have embraced the MVC (Model-View-Controller) application pattern, and if you're one of those developers, or even if you're just curious, this book is for you.

MVC allows for (buzzword alert!) a “greater separation of concerns” between components in your application. We'll go into the ramifications of this later on, but if it had to be said in a quick sentence: *ASP.NET MVC 2 is ASP.NET Unplugged*. ASP.NET MVC 2 is a tinkerer's framework that gives you very fine-grained control over your HTML and Javascript, as well as complete control over the programmatic flow of your application.

There are no declarative server controls in MVC, which some people may like, others may dislike. In the future, the MVC team may add declarative view controls to the mix, but these will be far different from the components that ASP.NET Web Forms developers are used to, in which a control encapsulates both the logic to render the view and the logic for responding to user input, etc... Having all that encapsulated in a single control in the view would violate the “separation of concerns” so central to this framework. The levels of abstraction have been collapsed, with all the doors and windows opened to let the air flow freely.

The final analogy we can throw at you is that ASP.NET MVC 2 is more of a motorcycle, whereas ASP.NET Web Forms might be more like a minivan, complete with airbags and a DVD player in case you have kids and you don't want them to fight while you're driving to the in-laws for Friday dinner. Some people like motorcycles, some people like mini vans. They'll both get you where you need to go, but one isn't technically *better* than the other.

HOW THIS BOOK IS STRUCTURED

This book is divided into two very broad sections, each comprising several chapters.

The first half of the book is concerned with introducing the MVC pattern and how ASP.NET MVC implements that pattern.

Chapter 1 starts off with a description of the Model-View-Controller pattern, explaining the basic concepts of the pattern and providing a bit of its history. The chapter goes on to describe the state of the MVC pattern on the web today as it is implemented by various frameworks, such as ASP.NET MVC 2.

Chapter 2 then covers the ways that ASP.NET MVC is different from ASP.NET Web Forms and how to get ASP.NET MVC 2 up and running.

Chapter 3 explores the structure of a standard MVC application and covers what you get out of the box. It covers some of the conventions and the digs a little under the hood to take a look at the entire request lifecycle for an ASP.NET MVC 2 request.

Chapter 4 digs deep into routing to describe the role that URLs play in your application and how routing figures into that. It also differentiates routing from URL rewriting and covers a bit on extending routing and writing unit tests for routes.