

With MvcContrib, NHibernate, and more

ASP.NET MVC IN ACTION

Jeffrey Palermo
Ben Scheirman
Jimmy Bogard

FOREWORD BY PHIL HAACK



 MANNING

contents

foreword xiii
preface xv
acknowledgments xviii
about this book xxi
about the authors xxvi
about the cover illustration xxviii

1 *Getting started with the ASP.NET MVC Framework* 1

- 1.1 Picking apart the default application 3
 - Creating the project* 4 ▪ *Your first routes* 7 ▪ *Running with the starter project* 9
- 1.2 Your first ASP.NET MVC controller from scratch 14
- 1.3 Our first view 16
- 1.4 Ensuring the application is maintainable 18
- 1.5 Testing controller classes 20
- 1.6 Summary 22

2 *The model in depth* 24

- 2.1 Understanding the basics of domain-driven design 25
- 2.2 Domain model for this book 26
 - Key entities and value objects* 26 ▪ *Aggregates* 27 ▪ *Persistence for the domain model* 29

- 2.3 Presentation model 31
 - Presentation model responsibilities* 31 ▪ *Projecting from the domain model* 33
- 2.4 Working with the model 34
 - Crafting the route* 35 ▪ *Crafting the controller action* 35 ▪ *Test-driving the feature* 36 ▪ *Finishing the view* 39
- 2.5 Summary 42

3 *The controller in depth* 44

- 3.1 The controller action 45
- 3.2 Simple controllers do not need a view 47
- 3.3 Testing controllers 50
 - Testing the RedirectController* 50 ▪ *Making dependencies explicit* 52 ▪ *Using test doubles, such as stubs and mocks* 53 ▪ *Elements of a good controller unit test* 55
- 3.4 Simple actions and views 56
- 3.5 Working with form values 57
- 3.6 Processing querystring parameters 58
- 3.7 Binding more complex objects in action parameters 59
- 3.8 Options for passing ViewData 61
- 3.9 Filters 62
- 3.10 Summary 64

4 *The view in depth* 65

- 4.1 How ASP.NET MVC views differ from Web Forms 66
- 4.2 Folder structure and view basics 67
- 4.3 Overview of view basics 69
 - Examining the IViewEngine abstraction* 70 ▪ *Understanding master pages in the ASP.NET MVC Framework* 71 ▪ *Using ViewData to send objects to a view* 73 ▪ *Partial views can help decompose a complex screen* 76
- 4.4 Leveraging the view to create dynamic screens 79
 - Rendering forms with view helpers and data binding* 79 ▪ *Posting HTML forms back to the server* 84 ▪ *Validation and error reporting* 85 ▪ *Extending HtmlHelper* 88
- 4.5 Summary 90

5 Routing 91

- 5.1 What are routes? 92
 - What's that curl command?* 92
 - *Taking back control of the URL with routing* 94
- 5.2 Designing a URL schema 95
 - Make simple, clean URLs* 95
 - *Make hackable URLs* 96
 - *Allow URL parameters to clash* 96
 - *Keep URLs short* 97
 - *Avoid exposing database IDs wherever possible* 97
 - Consider adding unnecessary information* 98
- 5.3 Implementing routes in ASP.NET MVC 99
 - URL schema for an online store* 102
 - *Adding a custom static route* 103
 - *Adding a custom dynamic route* 103
 - *Catch-all routes* 105
- 5.4 Using the routing system to generate URLs 107
- 5.5 Creating routes for Code Camp Server 108
- 5.6 Testing route behavior 111
- 5.7 Using routing with existing ASP.NET projects 115
- 5.8 Summary 117

6 Customizing and extending the ASP.NET MVC Framework 119

- 6.1 Extending URL routing 120
- 6.2 Creating your own ControllerFactory 125
 - The ControllerFactory implementation* 126
 - *Leveraging IoC for your controllers* 130
- 6.3 Extending the controller 134
 - Creating a FormattableController* 135
 - *Working with action filters* 138
- 6.4 Creating a custom view engine 141
- 6.5 Customizing Visual Studio for ASP.NET MVC 145
 - Creating custom T4 templates* 145
 - *Adding a custom test project template to the new project wizard* 148
- 6.6 Summary 150

7 Scaling the architecture for complex sites 152

- 7.1 Taming large controller actions 153
- 7.2 Whipping views into shape 156
 - Using and creating view helpers* 156
 - *Creating partials* 159
 - *Creating components* 163

- 7.3 Using action filters to load common data 165
- 7.4 Organizing controllers into areas 167
 - Capturing the area for a request* 168
 - *Creating a view engine with support for areas* 168
 - *Tying it all together* 171
- 7.5 Summary 172

8 **Leveraging existing ASP.NET features 174**

- 8.1 ASP.NET server controls 175
 - The TextBox* 175
 - *Other common controls* 176
 - *The GridView* 178
 - *Where do I get the good stuff?* 179
- 8.2 State management 179
 - Caching* 179
 - *Session state* 183
 - *Cookies* 184
 - *Request storage* 184
- 8.3 Tracing and debugging 185
 - TraceContext* 186
 - *Health monitoring* 186
- 8.4 Implementing personalization and localization 187
 - Leveraging ASP.NET personalization* 187
 - *Leveraging ASP.NET localization* 188
- 8.5 Implementing ASP.NET site maps 192
- 8.6 Summary 194

9 **AJAX in ASP.NET MVC 195**

- 9.1 Diving into AJAX with an example 196
- 9.2 AJAX with ASP.NET Web Forms 198
- 9.3 AJAX in ASP.NET MVC 200
 - Hijacking Code Camp Server* 201
 - *AJAX with JSON* 207
 - Adding alternate view formats to the controller* 208
 - *Consuming a JSON action from the view* 210
 - *AJAX helpers* 213
- 9.4 Summary 214

10 **Hosting and deployment 216**

- 10.1 Deployment scenarios 217
- 10.2 XCOPY deployment 218
- 10.3 Deploying to IIS 7 219
- 10.4 Deploying to IIS 6 and earlier 223
 - Configuring routes to use the .aspx extension* 224
 - *Configuring routes to use a custom extension* 225
 - *Using wildcard mapping with selective disabling* 226
 - *Using URL rewriting* 229

- 10.5 Automating deployments 232
 - Employing continuous integration* 232
 - *Enabling push-button XCOPY deployments* 233
 - *Managing environment configurations* 234
- 10.6 Summary 237

11 **Exploring MonoRail and Ruby on Rails** 238

- 11.1 MonoRail 239
 - Feature overview* 239
 - *ActiveRecord and Windsor* 248
 - MonoRail and Castle features available in ASP.NET MVC* 255
- 11.2 Ruby on Rails 255
 - Convention over configuration and “the Rails way”* 256
 - *Active Record* 260
 - *ActionPack* 264
- 11.3 Summary 269

12 **Best practices** 270

- 12.1 Controllers 271
 - Layer Supertype* 271
 - *Filters* 272
 - *Smart binders* 274
 - Hardcoded strings* 279
 - *Separated view models* 280
 - Validation* 281
- 12.2 Views 283
 - Strongly typed views* 283
 - *Fighting duplication* 284
 - *Embracing expressions* 286
- 12.3 Routes 289
 - Testing routes* 289
 - *Action naming* 292
- 12.4 Testing 294
 - Controller unit tests* 295
 - *Model binder unit tests* 298
 - *Action filter unit tests* 301
 - *Testing the last mile with UI tests* 303
- 12.5 Summary 311

13 **Recipes** 312

- 13.1 jQuery autocomplete text box 312
- 13.2 Automatic client-side validation 318
- 13.3 Data access with NHibernate 325
 - Functional overview of reference implementation* 326
 - Application architecture overview* 327
 - *Domain model—the application core* 328
 - *NHibernate configuration—infrastructure of the application* 330
 - *UI leverages domain model* 338
 - Pulling it together* 341
 - *Wrapping up data access with NHibernate* 344

- 13.4 Designing views with the Spark view engine 345
 - Installing and configuring Spark* 345
 - *Simple Spark view example* 347
- 13.5 Summary 351
 - index* 353