

Contents

Introduction	xvii
Chapter 1: NerdDinner	1
File ⇨ New Project	5
Examining the NerdDinner Directory Structure	7
Running the NerdDinner Application	9
Testing the NerdDinner Application	12
Creating the Database	13
Creating a New SQL Server Express Database	14
Creating Tables within Our Database	15
Setting Up a Foreign Key Relationship Between Tables	18
Adding Data to Our Tables	20
Building the Model	20
LINQ to SQL	21
Adding LINQ to SQL Classes to Our Project	21
Creating Data Model Classes with LINQ to SQL	22
NerdDinnerDataContext Class	25
Creating a DinnerRepository Class	26
Retrieving, Updating, Inserting, and Deleting Using the DinnerRepository Class	28
Integrating Validation and Business Rule Logic with Model Classes	30
Controllers and Views	34
Adding a DinnersController Controller	35
Adding Index and Details Action Methods to the DinnersController Class	36
Understanding ASP.NET MVC Routing	37
Using the DinnerRepository from Our DinnersController	39
Using Views with Our Controller	40
Implementing the “NotFound” View Template	42
Implementing the “Details” View Template	44
Implementing the “Index” View Template	49
Convention-Based Naming and the \Views Directory Structure	54
Create, Update, Delete Form Scenarios	56
URLs Handled by DinnersController	56
Implementing the HTTP-GET Edit Action Method	57
Html.BeginForm and Html.TextBox Html Helper Methods	61
Implementing the HTTP-POST Edit Action Method	62

Contents

Handling Edit Errors	66
Understanding ModelState and the Validation HTML Helper Methods	68
Using a AddRuleViolations Helper Method	70
Complete Edit Action Method Implementations	71
Implementing the HTTP-GET Create Action Method	72
Implementing the HTTP-POST Create Action Method	74
Implementing the HTTP-GET Delete Action Method	78
Implementing the HTTP-POST Delete Action Method	80
Model Binding Security	82
CRUD Wrap-Up	83
ViewData and ViewModel	86
Passing Data from Controllers to View Templates	86
Using the ViewData Dictionary	87
Using a ViewModel Pattern	89
Custom-Shaped ViewModel Classes	92
Partials and Master Pages	92
Revisiting Our Edit and Create View Templates	92
Using Partial View Templates	93
Using Partial View Templates to Clarify Code	97
Master Pages	98
Paging Support	101
Index() Action Method Recap	102
Understanding IQueryable<T>	102
Adding a “page” Value to the URL	103
Adding Page Navigation UI	106
Authentication and Authorization	110
Understanding Authentication and Authorization	110
Forms Authentication and the AccountController	111
Authorizing the /Dinners/Create URL Using the [Authorize] Filter	114
Using the User.Identity.Name Property When Creating Dinners	116
Using the User.Identity.Name Property When Editing Dinners	116
Showing/Hiding Edit and Delete Links	118
AJAX Enabling RSVPs Accepts	119
Indicating Whether the User Is RSVP'ed	120
Implementing the Register Action Method	122
Calling the Register Action Method Using AJAX	123
Adding a jQuery Animation	125
Cleanup — Refactor out a RSVP Partial View	127
Integrating an AJAX Map	127
Creating a Map Partial View	127
Creating a Map.js Utility Library	129
Integrating the Map with Create and Edit Forms	131

Contents

Integrating the Map with the Details View	135
Implementing Location Search in Our Database and Repository	136
Implementing a JSON-Based AJAX Search Action Method	140
Calling the JSON-Based AJAX Method Using jQuery	141
Unit Testing	145
Why Unit Test?	145
NerdDinner.Tests Project	146
Creating Unit Tests for Our Dinner Model Class	147
Running Tests	149
Creating DinnersController Unit Tests	150
Dependency Injection	152
Extracting an IDinnerRepository Interface	152
Updating DinnersController to Support Constructor Injection	154
Creating the FakeDinnerRepository Class	154
Using the FakeDinnerRepository with Unit Tests	157
Creating Edit Action Unit Tests	159
Mocking the User.Identity.Name Property	160
Testing UpdateModel() Scenarios	162
Testing Wrap-Up	163
NerdDinner Wrap-Up	164
Chapter 2: Model-View-Controller and ASP.NET	165
What Is Model-View-Controller?	165
MVC on the Web Today	167
Ruby on Rails	168
Django and Python	169
Spring, Struts, and Java	169
Zend Framework and PHP	170
MonoRail	170
ASP.NET MVC: The New Kid on the Block	170
Serving Methods, Not Files	171
Is This Web Forms 4.0?	171
Why Not Web Forms?	172
Cost/Benefit of Web Forms	172
Should You Fear ASP.NET MVC?	173
Summary	174
Chapter 3: ASP.NET > ASP.NET MVC	175
Abstraction: What Web Forms Does Well	175
A Basic Web Forms Application	176
The Importance of Events	180



Contents

The Leak: Where Web Forms Doesn't Exactly Fit	181
ViewState	183
Controlling Your Angle Brackets	183
Client IDs	183
Testing	183
Back to Basics: ASP.NET MVC Believes . . .	184
Orchestration versus Composing	184
Separation of Concerns: What It Means	184
Approaches to Maintainability	185
Caring About Testability	186
Common Reactions to ASP.NET MVC	187
This Looks Like Classic ASP from 1999!	187
Who Moved My "<asp:Cheese runat='server'>"	187
Yet Another Web Framework	188
Why "(ASP.NET > ASP.NET MVC) == True"	188
Convention over Configuration	190
Your First, er, Third, Request	194
The Request Lifecycle	196
Summary	196

Chapter 4: Routes and URLs **197**

Introduction to Routing	198
Compared to URL Rewriting	199
Defining Routes	199
Named Routes	206
Catch-All Parameter	206
StopRoutingHandler	208
Under the Hood: How Routes Generate URLs	209
Under the Hood: How Routes Tie Your URL to an Action	216
The High-Level Request Routing Pipeline	217
Route Matching	217
Advanced Routing with Custom Constraints	217
Route Extensibility	218
Using Routing with Web Forms	222
Summary	224

Chapter 5: Controllers **225**

History of the Controller	225
Defining the Controller: The IController Interface	227
The ControllerBase Abstract Base Class	229
The Controller Class and Actions	229
Action Methods	230



Contents

The ActionResult	233
Action Result Types	235
Action Result Helper Methods	238
Implicit Action Results	239
Action Invoker	240
How an Action Is Mapped to a Method	241
Mapping Parameters	243
Invoking Actions	244
Passing Data to Actions: The Model Binders	244
A Word About User Input	248
Summary	249
Chapter 6: Views	251
What a View Does	251
What a View Shouldn't Do	253
Specifying a View	253
Strongly Typed Views	255
HTML Helper Methods	257
HtmlHelper Class and Extension Methods	257
Using the HTML Helpers	258
The View Engine	266
Configuring a View Engine	267
Selecting a View Engine	267
Finding a View	268
The View Itself	269
Alternative View Engines	269
New View Engine or New ActionResult?	275
Summary	275
Chapter 7: AJAX	277
When AJAX Is Cool	278
When It's Not	278
AJAX Examples	280
Handling Disabled Scripting	280
Using Partial Views for Rendering	284
Some Things You May Not Know About Microsoft ASP.NET AJAX	288
Updating an HTML Element When Submitting a Form	290
The Auto-Complete Text Box	292
Implementing Auto-Complete with Microsoft ASP.NET AJAX	292
Filtering Data with a Selectbox	295



Contents

The Modal Popup with jQuery	296
The Modal Popup Code	297
The Rating Control	299
Summary	303
Chapter 8: Filters	305
Filters Included with ASP.NET MVC	305
Authorize	306
OutputCache	308
Exception Filter	310
Custom Filters	311
Writing a Custom Action Filter	316
Writing a Custom Authorization Filter	317
Writing a Custom Exception Filter	319
Filter Ordering	320
Filter Naming	321
Summary	323
Chapter 9: Securing Your Application	325
This Is a War	327
Knowing Your Enemy's Mind	327
Weapons	331
Spam	331
Case Study: Profiting from Evil with the Srizbi and Storm Botnets	332
Digital Stealth Ninja Network	333
Threat: Cross-Site Scripting (XSS)	334
Passive Injection	334
Active Injection	336
Preventing XSS	338
Html.AttributeEncode and Url.Encode	338
Threat: Cross-Site Request Forgery	339
Preventing CSRF Attacks	342
Threat: Cookie Stealing	343
Preventing Cookie Theft with HttpOnly	344
Keeping Your Pants Up: Proper Error Reporting and the Stack Trace	345
Securing Your Controllers, Not Your Routes	345
Using [Authorize] to Lock Down Your Action or Controller	346
Using [NonAction] to Protect Public Methods	346
Whitelist Form Binding	347
Summary: It's Up to You	348



Chapter 10: Test Driven Development with ASP.NET MVC	349
A Brief Introduction to TDD	350
What Does TDD Look Like?	350
Writing Good Unit Tests	353
What Are Some Benefits of Writing Tests?	357
How Do I Get Started?	357
Applying TDD to ASP.NET MVC	357
Testing Routes	358
Testing Controllers	360
Redirecting to Another Action	360
Testing View Helpers	362
Testing Views	364
Summary	365
Chapter 11: Testable Design Patterns	367
Why You Should Care About Testability	368
Big Design Up Front (BDUF)	368
Agile Software Development	369
You Want to Write Testable Code	370
Using Tests to Prove You're Done	371
Designing Your Application for Testability	371
Future-Proofing Your Application with Interfaces	371
The Single Responsibility Principle	373
Avoid Using Singletons and Static Methods	373
Testable Data Access	376
Creating the Model	377
The Repository Pattern in Detail	379
Implementing Business Logic with the Service Layer	383
Services Gone Wild	385
Partial Solution: Setting Controller Dependencies Manually	386
Summary	392
Chapter 12: Best of Both Worlds: Web Forms and MVC Together	393
How Is It Possible?	393
Including MVC in Existing Web Forms Applications	394
Step 1: Referencing the Required Libraries	394
Step 2: Creating the Necessary Directories	395
Step 3: Updating the Web.config	396

Contents

Adding Web Forms to an Existing ASP.NET MVC Application	398
The Easy Part: Do Nothing	399
When/Home/Isn't/Home/	401
Using System.Web.Routing to Route to Web Forms	401
Sharing Data Between Web Forms and MVC	402
Using HTTP POST	402
Using the ASP.NET Session	403
Using Cross-Page Posting	404
Using TempData	406
Migrating from Web Forms to MVC	407
Step 1: Create an Empty ASP.NET MVC Project with a Test Project	407
Step 2: Implement the Structure	408
Step 3: Add Images and Styling	410
Step 4: Setting Up Routing and Controllers	411
Step 5: Replacing Complex Server Controls	415
Step 6: Uploading Files and Working with Images	418
Summary	420
Index	421