

contents

<i>preface</i>	<i>xxvii</i>
<i>about this book</i>	<i>xxix</i>
<i>about the author</i>	<i>xxxi</i>
<i>about the cover illustration</i>	<i>xxxii</i>

1 Putting ASP.NET Core in context 1

1.1 Understanding the application frameworks 3

*Understanding the MVC Framework 3 ■ Understanding
Razor Pages 4 ■ Understanding Blazor 4 ■ Understanding
the utility frameworks 5 ■ Understanding the ASP.NET Core
platform 5*

1.2 Understanding this book 6

*What software do I need to follow the examples? 6 ■ What
platform do I need to follow the examples? 6 ■ What if I have
problems following the examples? 6 ■ What if I find an error in
the book? 7 ■ What does this book cover? 7 ■ What doesn't this
book cover? 8 ■ How do I contact the author? 8 ■ What if I
really enjoyed this book? 9 ■ What if this book has made me angry
and I want to complain? 9*

Summary 9

Getting started 13

- 2.1 Choosing a code editor 13
 - Installing Visual Studio Code 14* ■ *Installing Visual Studio 16*
- 2.2 Creating an ASP.NET Core project 19
 - Opening the project using Visual Studio 20* ■ *Opening the project with Visual Studio Code 21*
- 2.3 Running the ASP.NET Core application 22
 - Understanding endpoints 23* ■ *Understanding routes 25*
 - Understanding HTML rendering 25* ■ *Putting the pieces together 30*
- Summary 30

Your first ASP.NET Core application 31

- 3.1 Setting the scene 31
- 3.2 Creating the project 32
 - Preparing the project 32* ■ *Adding a data model 34*
 - Creating a second action and view 35* ■ *Linking action methods 37* ■ *Building the form 38* ■ *Receiving form data 39* ■ *Adding the thanks view 42* ■ *Displaying responses 44* ■ *Adding validation 46* ■ *Styling the content 51*
- Summary 57

Using the development tools 58

- 4.1 Creating ASP.NET Core projects 59
 - Creating a project using the command line 59*
- 4.2 Adding code and content to projects 62
 - Understanding item scaffolding 64*
- 4.3 Building and running projects 64
 - Using the hot reload feature 66*
- 4.4 Managing packages 68
 - Managing NuGet packages 68* ■ *Managing tool packages 69*
 - Managing client-side packages 69*
- 4.5 Debugging projects 71
- Summary 72

Essential C# features 73

- 5.1 Preparing for this chapter 74
 - Opening the project 75* ■ *Enabling the MVC Framework 75*
 - Creating the application components 75* ■ *Selecting the HTTP port 77* ■ *Running the example application 78*
- 5.2 Understanding top-level statements 78
- 5.3 Understanding global using statements 79
 - Understanding implicit using statements 80*
- 5.4 Understanding null state analysis 81
 - Ensuring fields and properties are assigned values 82*
 - Providing a default value for non-nullable types 84*
 - Using nullable types 84* ■ *Checking for null values 86*
 - Overriding null state analysis 89* ■ *Disabling null state analysis warnings 89*
- 5.5 Using string interpolation 90
- 5.6 Using object and collection initializers 90
 - Using an index initializer 92*
- 5.7 Using target-typed new expressions 93
- 5.8 Pattern Matching 94
 - Pattern matching in switch statements 94*
- 5.9 Using extension methods 95
 - Applying extension methods to an interface 97*
 - Creating filtering extension methods 99*
- 5.10 Using lambda expressions 100
 - Defining functions 102* ■ *Using lambda expression methods and properties 105*
- 5.11 Using type inference and anonymous types 107
 - Using anonymous types 107*
- 5.12 Using default implementations in interfaces 109
- 5.13 Using asynchronous methods 111
 - Working with tasks directly 111* ■ *Applying the async and await keywords 112* ■ *Using an asynchronous enumerable 114*
- 5.14 Getting names 117
- Summary 118

Testing ASP.NET Core applications 119**6.1 Preparing for this chapter 120**

Opening the project 121 ■ *Selecting the HTTP port 121*
Enabling the MVC Framework 122 ■ *Creating the application components 122* ■ *Running the example application 124*

6.2 Creating a unit test project 124**6.3 Writing and running unit tests 125**

Running tests with the Visual Studio Test Explorer 127
Running tests with Visual Studio Code 128 ■ *Running tests from the command line 128* ■ *Correcting the unit test 129*
Isolating components for unit testing 130 ■ *Using a mocking package 135* ■ *Creating a mock object 135*

Summary 137

SportsStore: A real application 138**7.1 Creating the projects 139**

Creating the unit test project 140 ■ *Opening the projects 140*
Configuring the HTTP port 140 ■ *Creating the application project folders 141* ■ *Preparing the services and the request pipeline 142* ■ *Configuring the Razor view engine 143*
Creating the controller and view 144 ■ *Starting the data model 144* ■ *Checking and running the application 145*

7.2 Adding data to the application 145

Installing the Entity Framework Core packages 146 ■ *Defining the connection string 146* ■ *Creating the database context class 147* ■ *Configuring Entity Framework Core 148*
 ■ *Creating a repository 148* ■ *Creating the database migration 151* ■ *Creating seed data 151*

7.3 Displaying a list of products 154

Preparing the controller 154 ■ *Updating the view 156*
Running the application 157

7.4 Adding pagination 157

Displaying page links 159 ■ *Improving the URLs 167*

7.5 Styling the content 169

Installing the Bootstrap package 169 ■ *Applying Bootstrap styles 169* ■ *Creating a partial view 172*

Summary 174

8 SportsStore: Navigation and cart 175**8.1 Adding navigation controls 175**

Filtering the product list 176 ■ *Refining the URL scheme 180*
Building a category navigation menu 184 ■ *Correcting the page count 191*

8.2 Building the shopping cart 194

Configuring Razor Pages 194 ■ *Creating a Razor Page 196*
Creating the Add to Cart buttons 197 ■ *Enabling sessions 199*
Implementing the cart feature 200

Summary 210

9 SportsStore: Completing the cart 211**9.1 Refining the cart model with a service 211**

Creating a storage-aware cart class 212 ■ *Registering the service 214* ■ *Simplifying the cart Razor Page 215*

9.2 Completing the cart functionality 217

Removing items from the cart 217 ■ *Adding the cart summary widget 220*

9.3 Submitting orders 223

Creating the model class 223 ■ *Adding the checkout process 224* ■ *Creating the controller and view 225*
Implementing order processing 227 ■ *Completing the order controller 230* ■ *Displaying validation errors 234*
Displaying a summary page 236

Summary 237

10 SportsStore: Administration 238**10.1 Preparing Blazor Server 239**

Creating the imports file 240 ■ *Creating the startup Razor Page 240* ■ *Creating the routing and layout components 241*
Creating the Razor Components 242 ■ *Checking the Blazor setup 242*

10.2 Managing orders 243

Enhancing the model 244 ■ *Displaying orders to the administrator 245*

10.3 Adding catalog management 249

Expanding the repository 249 ■ *Applying validation attributes to the data model 250* ■ *Creating the list component 251*

Creating the detail component 253 ■ *Creating the editor component* 254 ■ *Deleting products* 257

Summary 259

SportsStore: Security and deployment 260

11.1 Creating the Identity database 261

Installing the Identity package for Entity Framework Core 261
Creating the context class 261 ■ *Defining the connection string* 262 ■ *Configuring the application* 262 ■ *Creating and applying the database migration* 264 ■ *Defining the seed data* 264

11.2 Adding a conventional administration feature 267

11.3 Applying a basic authorization policy 268

11.4 Creating the account controller and views 269

11.5 Testing the security policy 273

11.6 Preparing ASP.NET Core for deployment 274

Configuring error handling 274 ■ *Creating the production configuration settings* 276 ■ *Creating the Docker image* 277
Running the containerized application 280

Summary 281

283

Understanding the ASP.NET Core platform 285

12.1 Preparing for this chapter 286

Running the example application 288

12.2 Understanding the ASP.NET Core platform 288

Understanding middleware and the request pipeline 288
Understanding services 289

12.3 Understanding the ASP.NET Core project 289

Understanding the entry point 291 ■ *Understanding the project file* 292

12.4 Creating custom middleware 293

Defining middleware using a class 297 ■ *Understanding the return pipeline path* 299 ■ *Short-Circuiting the request pipeline* 300 ■ *Creating pipeline branches* 302
Creating terminal middleware 304

12.5 Configuring middleware 306

Using the options pattern with class-based middleware 308

Summary 310

13

Using URL routing 311

13.1 Preparing for this chapter 312

Understanding URL routing 315 ■ *Adding the routing middleware and defining an endpoint* 316 ■ *Simplifying the pipeline configuration* 319 ■ *Understanding URL patterns* 321 ■ *Using segment variables in URL patterns* 322
Generating URLs from routes 326

13.2 Managing URL matching 330

Matching multiple values from a single URL segment 330
Using default values for segment variables 331 ■ *Using optional segments in a URL Pattern* 333 ■ *Using a catchall segment variable* 334 ■ *Constraining segment matching* 336
Defining fallback routes 340

13.3 Advanced routing features 341

Creating custom constraints 341 ■ *Avoiding ambiguous route exceptions* 342 ■ *Accessing the endpoint in a middleware component* 344

Summary 347

14

Using dependency injection 348

14.1 Preparing for this chapter 350

Creating a middleware component and an endpoint 350
Configuring the request pipeline 351

14.2 Understanding service location and tight coupling 352

Understanding the service location problem 353
Understanding the tightly coupled components problem 355

14.3 Using dependency injection 357

Using a Service with a Constructor Dependency 359
Getting services from the HttpContext object 360

14.4 Using Service Lifecycles 364

Creating transient services 365 ■ *Avoiding the transient service reuse pitfall* 366 ■ *Using scoped services* 369

14.5 Other dependency injection features 374

Creating dependency chains 374 ■ *Accessing services in the Program.cs file* 376 ■ *Using service factory functions* 377
Creating services with multiple implementations 379
Using unbound types in services 381

Summary 383

5 Using the platform features, part 1 384

15.1 Preparing for this chapter 385

15.2 Using the configuration service 387

Understanding the environment configuration file 387
Accessing configuration settings 388 ■ *Using the configuration data in the Program.cs file* 390 ■ *Using configuration data with the options pattern* 390 ■ *Understanding the launch settings file* 392 ■ *Using the environment service* 397 ■ *Storing user secrets* 398

15.3 Using the logging service 401

Generating logging messages 401 ■ *Logging messages with attributes* 405 ■ *Configuring minimum logging levels* 407
Logging HTTP requests and responses 409

15.4 Using static content and client-side packages 411

Adding the static content middleware 412 ■ *Using client-side packages* 415

Summary 417

16 Using the platform features, part 2 418

16.1 Preparing for this chapter 419

16.2 Using cookies 420

Enabling cookie consent checking 422 ■ *Managing cookie consent* 424

16.3 Using sessions 426

Configuring the session service and middleware 426
Using session data 429

16.4 Working with HTTPS connections 431

Enabling HTTPS connections 431 ■ *Detecting HTTPS requests* 433 ■ *Enforcing HTTPS requests* 434
Enabling HTTP strict transport security 435

16.5 Using rate limits 437

16.6 Handling exceptions and errors 441

Returning an HTML error response 442 ■ *Enriching status code responses* 444

16.7 Filtering requests using the host header 446

Summary 449

17 Working with data 450

17.1 Preparing for this chapter 452

17.2 Caching data 452

Caching data values 454 ■ *Using a shared and persistent data cache* 457

17.3 Caching responses 461

17.4 Caching output 464

Defining a custom cache policy 467

17.5 Using Entity Framework Core 469

Installing Entity Framework Core 469 ■ *Creating the data model* 470 ■ *Configuring the database service* 471
Creating and applying the database migration 473 ■ *Seeding the database* 473 ■ *Using data in an endpoint* 476

Summary 479

PART 3..... 48**18****Creating the example project 483**

18.1 Creating the project 484

18.2 Adding a data model 485

Adding NuGet packages to the project 485 ■ *Creating the data model* 485 ■ *Preparing the seed data* 487 ■ *Configuring EF Core services and middleware* 488 ■ *Creating and applying the migration* 489

18.3 Adding the CSS framework 490

18.4 Configuring the request pipeline 490

18.5 Running the example application 491

19**Creating RESTful web services 492**

19.1 Preparing for this chapter 493

- 19.2 Understanding RESTful web services 494
 Understanding request URLs and methods 494
 Understanding JSON 495
- 19.3 Creating a web service using the minimal API 496
- 19.4 Creating a web service using a controller 498
 Enabling the MVC Framework 499 ■ *Creating a controller* 500
- 19.5 Improving the web service 510
 Using asynchronous actions 511 ■ *Preventing over-binding* 512 ■ *Using action results* 514 ■ *Validating data* 519 ■ *Applying the API controller attribute* 521
 Omitting Null properties 523 ■ *Applying a rate limit* 526
- Summary 529

Advanced web service features 530

- 20.1 Preparing for this chapter 531
 Dropping the database 532 ■ *Running the example application* 532
- 20.2 Dealing with related data 533
 Breaking circular references in related data 535
- 20.3 Supporting the HTTP PATCH method 536
 Understanding JSON Patch 536 ■ *Installing and configuring the JSON Patch package* 537 ■ *Defining the action method* 538
- 20.4 Understanding content formatting 540
 Understanding the default content policy 540 ■ *Understanding content negotiation* 542 ■ *Specifying an action result format* 547 ■ *Requesting a format in the URL* 548
 Restricting the formats received by an action method 549
 Caching output 551
- 20.5 Documenting and exploring web services 553
 Resolving action conflicts 553 ■ *Installing and configuring the Swashbuckle package* 554 ■ *Fine-Tuning the API description* 556
- Summary 560

- 21 **Using controllers with views, part I 561**
- 21.1 Preparing for this chapter 562
 Dropping the database 563 ■ *Running the example application* 563
- 21.2 Getting started with views 564
 Configuring the application 564 ■ *Creating an HTML controller* 565 ■ *Creating a Razor View* 568 ■ *Selecting a View by name* 571
- 21.3 Working with Razor Views 575
 Setting the view model type 578 ■ *Understanding the view model type pitfall* 584
- 21.4 Understanding the Razor syntax 587
 Understanding directives 587 ■ *Understanding content expressions* 588 ■ *Setting element content* 589 ■ *Setting attribute values* 590 ■ *Using conditional expressions* 591
 Enumerating sequences 594 ■ *Using Razor code blocks* 596
- Summary 598

- 22 **Using controllers with views, part II 599**
- 22.1 Preparing for this chapter 600
 Dropping the database 601 ■ *Running the example application* 601
- 22.2 Using the view bag 602
- 22.3 Using temp data 604
- 22.4 Working with layouts 607
 Configuring layouts using the view bag 609 ■ *Using a view start file* 610 ■ *Overriding the default layout* 612 ■ *Using layout sections* 615
- 22.5 Using partial views 622
 Enabling partial views 622 ■ *Creating a partial view* 622
 Applying a partial view 623
- 22.6 Understanding content-encoding 625
 Understanding HTML encoding 626 ■ *Understanding JSON encoding* 628
- Summary 629

Using Razor Pages 630**23.1 Preparing for this chapter 632***Running the example application 632***23.2 Understanding Razor Pages 633***Configuring Razor Pages 633 ■ Creating a Razor Page 634***23.3 Understanding Razor Pages routing 639***Specifying a routing pattern in a Razor Page 640**Adding routes for a Razor Page 642***23.4 Understanding the Page model class 644***Using a code-behind class file 644 ■ Understanding action results in Razor Pages 647 ■ Handling multiple HTTP methods 651 ■ Selecting a handler method 653***23.5 Understanding the Razor Page view 656***Creating a layout for Razor Pages 656 ■ Using partial views in Razor Pages 658 ■ Creating Razor Pages without page models 659***Summary 661****Using view components 662****24.1 Preparing for this chapter 663***Dropping the database 666 ■ Running the example application 666***24.2 Understanding view components 667****24.3 Creating and using a view component 667***Applying a view component 668***24.4 Understanding view component results 671***Returning a partial view 672 ■ Returning HTML fragments 675***24.5 Getting context data 678***Providing context from the parent view using arguments 679**Creating asynchronous view components 683***24.6 Creating view components classes 685***Creating a hybrid controller class 688***Summary 690****25 Using tag helpers 691****25.1 Preparing for this chapter 692***Dropping the database 694 ■ Running the example application 694***25.2 Creating a tag helper 695***Defining the tag helper class 695 ■ Registering tag helpers 698*
*Using a tag helper 698 ■ Narrowing the scope of a tag helper 700 ■ Widening the scope of a tag helper 701***25.3 Advanced tag helper features 703***Creating shorthand elements 703 ■ Creating elements programmatically 706 ■ Prepending and appending content and elements 707 ■ Getting view context data 710 ■ Working with model expressions 712 ■ Coordinating between tag helpers 718 ■ Suppressing the output element 720***25.4 Using tag helper components 721***Creating a tag helper component 721 ■ Expanding tag helper component element selection 723***Summary 725****26 Using the built-in tag helpers 726****26.1 Preparing for this chapter 727***Adding an image file 729 ■ Installing a client-side package 729 ■ Dropping the database 730 ■ Running the example application 730***26.2 Enabling the built-in tag helpers 730****26.3 Transforming anchor elements 731***Using anchor elements for Razor Pages 733***26.4 Using the JavaScript and CSS tag helpers 735***Managing JavaScript files 735 ■ Managing CSS stylesheets 743***26.5 Working with image elements 746****26.6 Using the data cache 748***Setting cache expiry 750***26.7 Using the hosting environment tag helper 753****Summary 754**

Using the forms tag helpers 755

- 27.1 Preparing for this chapter 756
 - Dropping the database 758* ■ *Running the example application 758*
- 27.2 Understanding the form handling pattern 758
 - Creating a controller to handle forms 759* ■ *Creating a Razor Page to handle forms 761*
- 27.3 Using tag helpers to improve HTML forms 764
 - Working with form elements 764* ■ *Transforming form buttons 766*
- 27.4 Working with input elements 767
 - Transforming the input element type attribute 769* ■ *Formatting input element values 771* ■ *Displaying values from related data in input elements 775*
- 27.5 Working with label elements 779
- 27.6 Working with select and option elements 781
 - Populating a select element 783*
- 27.7 Working with text areas 785
- 27.8 Using the anti-forgery feature 787
 - Enabling the anti-forgery feature in a controller 789* ■ *Enabling the anti-forgery feature in a Razor Page 790* ■ *Using anti-forgery tokens with JavaScript clients 790*

Using model binding 794

- 28.1 Preparing for this chapter 796
 - Dropping the database 797* ■ *Running the example application 797*
- 28.2 Understanding model binding 797
- 28.3 Binding simple data types 799
 - Binding simple data types in Razor Pages 800* ■ *Understanding default binding values 802*
- 28.4 Binding complex types 805
 - Binding to a property 807* ■ *Binding nested complex types 809* ■ *Selectively binding properties 813*
- 28.5 Binding to arrays and collections 816
 - Binding to arrays 816* ■ *Binding to simple collections 820*

Binding to dictionaries 820 ■ *Binding to collections of complex types 822*

- 28.6 Specifying a model binding source 824
 - Selecting a binding source for a property 827* ■ *Using headers for model binding 827* ■ *Using request bodies as binding sources 829*
- 28.7 Manual model binding 830

29 Using model validation 833

- 29.1 Preparing for this chapter 835
 - Dropping the database 836* ■ *Running the example application 836*
- 29.2 Understanding the need for model validation 837
- 29.3 Validating data 837
 - Displaying validation messages 841* ■ *Understanding the implicit validation checks 843* ■ *Performing explicit validation 844* ■ *Configuring the default validation error messages 847* ■ *Displaying property-level validation messages 850* ■ *Displaying model-level messages 851*
- 29.4 Explicitly validating data in a Razor Page 854
- 29.5 Specifying validation rules using metadata 857
 - Creating a custom property validation attribute 861*
 - Creating a custom model validation attribute 862*
- 29.6 Performing client-side validation 866
- 29.7 Performing remote validation 869
 - Performing remote validation in Razor Pages 872*

30 Using filters 875

- 30.1 Preparing for this chapter 876
 - Enabling HTTPS Connections 878* ■ *Dropping the database 879* ■ *Running the example application 880*
- 30.2 Using filters 880
- 30.3 Understanding filters 885
- 30.4 Creating custom filters 887
 - Understanding authorization filters 887* ■ *Understanding resource filters 890* ■ *Understanding action filters 894*

Understanding page filters 899 ■ *Understanding result filters* 903 ■ *Understanding exception filters* 908 ■ *Creating an exception filter* 909

30.5 Managing the filter lifecycle 911

Creating filter factories 913 ■ *Using dependency injection scopes to manage filter lifecycles* 915

30.6 Creating global filters 916

30.7 Understanding and changing filter order 918

Creating form applications 924

31.1 Preparing for this chapter 924

Dropping the database 928 ■ *Running the example application* 928

31.2 Creating an MVC forms application 928

Preparing the view model and the view 928 ■ *Reading data* 931 ■ *Creating data* 933 ■ *Editing data* 936 ■ *Deleting data* 939

31.3 Creating a Razor Pages forms application 940

Creating common functionality 942 ■ *Defining pages for the CRUD operations* 945

31.4 Creating new related data objects 948

Providing the related data in the same request 948 ■ *Breaking out to create new data* 952

Creating the example project 961

32.1 Creating the project 962

Adding NuGet packages to the project 963

32.2 Adding a data model 963

Preparing the seed data 965 ■ *Configuring Entity Framework Core* 966 ■ *Creating and applying the migration* 967

32.3 Adding the Bootstrap CSS framework 967

32.4 Configuring the services and middleware 968

32.5 Creating a controller and view 969

32.6 Creating a Razor Page 971

32.7 Running the example application 973

33

Using Blazor Server, part 1 974

33.1 Preparing for this chapter 976

33.2 Understanding Blazor Server 976

Understanding the Blazor Server advantages 978

Understanding the Blazor Server disadvantages 978

Choosing between Blazor Server and Angular/React/Vue.js 978

33.3 Getting started with Blazor 978

Configuring ASP.NET Core for Blazor Server 979

Creating a Razor Component 981

33.4 Understanding the basic Razor Component features 986

Understanding Blazor events and data bindings 986

Working with data bindings 994

33.5 Using class files to define components 1000

Using a code-behind class 1000 ■ *Defining a Razor Component class* 1002

34

Using Blazor Server, part 2 1004

34.1 Preparing for this chapter 1005

34.2 Combining components 1006

Configuring components with attributes 1008 ■ *Creating custom events and bindings* 1012

34.3 Displaying child content in a component 1017

Creating template components 1019 ■ *Using generic type parameters in template components* 1022 ■ *Cascading parameters* 1028

34.4 Handling errors 1031

Handling connection errors 1031 ■ *Handling uncaught application errors* 1034 ■ *Using error boundaries* 1036

35

Advanced Blazor features 1042

35.1 Preparing for this chapter 1044

35.2 Using component routing 1044

Preparing the Razor Page 1045 ■ *Adding routes to components* 1047 ■ *Navigating between routed*

components 1050 ■ *Receiving routing data* 1053
Defining common content using layouts 1055

35.3 Understanding the component lifecycle methods 1057
Using the lifecycle methods for asynchronous tasks 1060

35.4 Managing component interaction 1062
Using references to child components 1062 ■ *Interacting with components from other code* 1065 ■ *Interacting with components using JavaScript* 1070

Blazor forms and data 1080

36.1 Preparing for this chapter 1081
Dropping the database and running the application 1085

36.2 Using the Blazor form components 1085
Creating custom form components 1088 ■ *Validating form data* 1092 ■ *Handling form events* 1096

36.3 Using Entity Framework Core with Blazor 1098
Understanding the EF Core context scope issue 1098
Understanding the repeated query issue 1103

36.4 Performing CRUD operations 1109
Creating the list component 1110 ■ *Creating the details component* 1111 ■ *Creating the editor component* 1113

36.5 Extending the Blazor form features 1115
Creating a custom validation constraint 1116
Creating a valid-only submit button component 1119

Using Blazor WebAssembly 1122

37.1 Preparing for this chapter 1123
Dropping the database and running the application 1125

37.2 Setting Up Blazor WebAssembly 1126
Creating the shared project 1126 ■ *Creating the Blazor WebAssembly project* 1126 ■ *Preparing the ASP.NET Core project* 1127 ■ *Adding the solution references* 1127
Opening the projects 1127 ■ *Completing the Blazor WebAssembly configuration* 1128 ■ *Testing the placeholder components* 1130

37.3 Creating a Blazor WebAssembly component 1131

Importing the data model namespace 1131 ■ *Creating a component* 1131 ■ *Creating a layout* 1136 ■ *Defining CSS styles* 1137

37.4 Completing the Blazor WebAssembly Form application 1138

Creating the details component 1138 ■ *Creating the editor component* 1139

38

Using ASP.NET Core Identity 1143

38.1 Preparing for this chapter 1144

38.2 Preparing the project for ASP.NET Core Identity 1145
Preparing the ASP.NET Core Identity database 1146
Configuring the application 1147 ■ *Creating and applying the Identity database migration* 1148

38.3 Creating user management tools 1148

Preparing for user management tools 1150 ■ *Enumerating user accounts* 1151 ■ *Creating users* 1152 ■ *Editing users* 1160
Deleting users 1162

38.4 Creating role management tools 1163

Preparing for role management tools 1165 ■ *Enumerating and deleting roles* 1165 ■ *Creating roles* 1167 ■ *Assigning role membership* 1168

39

Applying ASP.NET Core Identity 1172

39.1 Preparing for this chapter 1173

39.2 Authenticating users 1174

Creating the login feature 1175 ■ *Inspecting the ASP.NET Core Identity cookie* 1177 ■ *Creating a Sign-Out page* 1178
Testing the authentication feature 1179 ■ *Enabling the Identity authentication middleware* 1179

39.3 Authorizing access to endpoints 1183

Applying the authorization attribute 1183 ■ *Enabling the authorization middleware* 1184 ■ *Creating the access denied endpoint* 1184 ■ *Creating the seed data* 1185 ■ *Testing the authentication sequence* 1187

39.4 Authorizing access to Blazor applications 1188

Performing authorization in Blazor components 1190
Displaying content to authorized users 1192

39.5 Authenticating and authorizing web services 1194

Building a simple JavaScript client 1196 ■ *Restricting access to the web service* 1198 ■ *Using cookie authentication* 1199
Using bearer token authentication 1202 ■ *Creating tokens* 1203 ■ *Authenticating with tokens* 1206 ■ *Restricting access with tokens* 1208 ■ *Using tokens to request data* 1208

index 1211