



Praise for the Third Edition of *Spring in Action*

Continues to be the de-facto reference guide to Spring. Offers clear explanations of concepts with very good examples in an easy-to-read format.

—Dan Dobrin, CIBC

An indispensable guide to the large landscape of Spring.

—Mykel Alvis, Automaton Online

The one book you need on your desk when working with Spring.

—Josh Devins, Nokia

Covers both the fundamentals and the breadth of Spring.

—Chad Davis, Blackdog Software, Inc.

Using Spring is not difficult—but with this book it becomes much easier.

—Alberto Lagna, Biznology

One of my favorite technology books. Great content delivered by a great teacher.

—Robert Hanson, Author of Manning's *GWT in Action*

The right dose of humor with a load of technical wisdom is the perfect mix for learning Spring.

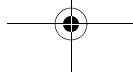
—Valentin Crettaz, Goomzee

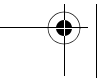
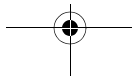
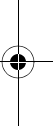
Tremendous focus—and fun to read.

—Doug Warren, Java Web Services

Craig's witty examples make complex concepts easy to understand.

—Dan Alford







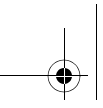
Spring in Action

FOURTH EDITION

CRAIG WALLS



MANNING
SHELTER ISLAND



For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity.

For more information, please contact


Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

©2015 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

- © Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

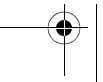
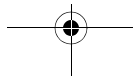
 Manning Publications Co.
20 Baldwin Road
Shelter Island, NY 11964

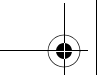
Development editor: Cynthia Kane
Copyeditor: Andy Carroll
Proofreader: Alyson Brener
Typesetter: Dottie Marsico
Cover designer: Marija Tudor

ISBN 9781617291203

Printed in the United States of America

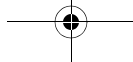
1 2 3 4 5 6 7 8 9 10 – EBM – 19 18 17 16 15 14

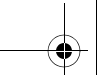




brief contents

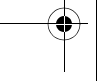
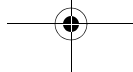
PART 1	CORE SPRING	1
1	▪ Springing into action	3
2	▪ Wiring beans	32
3	▪ Advanced wiring	64
4	▪ Aspect-oriented Spring	97
PART 2	SPRING ON THE WEB.....	129
5	▪ Building Spring web applications	131
6	▪ Rendering web views	164
7	▪ Advanced Spring MVC	194
8	▪ Working with Spring Web Flow	219
9	▪ Securing web applications	244
PART 3	SPRING IN THE BACK END	279
10	▪ Hitting the database with Spring and JDBC	281
11	▪ Persisting data with object-relational mapping	305
12	▪ Working with NoSQL databases	327
13	▪ Caching data	362
14	▪ Securing methods	379

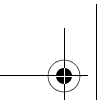




PART 4 INTEGRATING SPRING.....391

- 15 ■ Working with remote services 393
- 16 ■ Creating REST APIs with Spring MVC 416
- 17 ■ Messaging in Spring 452
- 18 ■ Messaging with WebSocket and STOMP 485
- 19 ■ Sending email with Spring 511
- 20 ■ Managing Spring beans with JMX 523
- 21 ■ Simplifying Spring development with Spring Boot 540





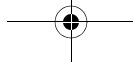
contents

preface xvii
acknowledgments xix
about this book xxi

PART 1 CORE SPRING1

1 *Springing into action* 3

- 1.1 Simplifying Java development 4
 - Unleashing the power of POJOs* 5 ■ *Injecting dependencies* 5
 - Applying aspects* 11 ■ *Eliminating boilerplate code with templates* 16
- 1.2 Containing your beans 18
 - Working with an application context* 19 ■ *A bean's life* 20
- 1.3 Surveying the Spring landscape 21
 - Spring modules* 22 ■ *The Spring portfolio* 24
- 1.4 What's new in Spring 27
 - What was new in Spring 3.1?* 27 ■ *What was new in Spring 3.2?* 28 ■ *What's new in Spring 4.0?* 29
- 1.5 Summary 30



2 Wiring beans 32

- 2.1 Exploring Spring's configuration options 33
- 2.2 Automatically wiring beans 34
 - Creating discoverable beans* 34
 - *Naming a component-scanned bean* 38
 - *Setting a base package for component scanning* 38
 - Annotating beans to be automatically wired* 39
 - *Verifying automatic configuration* 41
- 2.3 Wiring beans with Java 43
 - Creating a configuration class* 43
 - *Declaring a simple bean* 44
 - *Injecting with JavaConfig* 45
- 2.4 Wiring beans with XML 46
 - Creating an XML configuration specification* 47
 - *Declaring a simple <bean>* 48
 - *Initializing a bean with constructor injection* 49
 - *Setting properties* 54
- 2.5 Importing and mixing configurations 59
 - Referencing XML configuration in JavaConfig* 59
 - Referencing JavaConfig in XML configuration* 61
- 2.6 Summary 63

3 Advanced wiring 64

- 3.1 Environments and profiles 64
 - Configuring profile beans* 66
 - *Activating profiles* 70
- 3.2 Conditional beans 72
- 3.3 Addressing ambiguity in autowiring 75
 - Designating a primary bean* 76
 - *Qualifying autowired beans* 77
- 3.4 Scoping beans 81
 - Working with request and session scope* 82
 - *Declaring scoped proxies in XML* 84
- 3.5 Runtime value injection 84
 - Injecting external values* 85
 - *Wiring with the Spring Expression Language* 89
- 3.6 Summary 95

4 Aspect-oriented Spring 97

- 4.1 What is aspect-oriented programming? 98
 - Defining AOP terminology* 99
 - *Spring's AOP support* 101

- 4.2 Selecting join points with pointcuts 103
 - Writing pointcuts* 104
 - *Selecting beans in pointcuts* 106
- 4.3 Creating annotated aspects 106
 - Defining an aspect* 106
 - *Creating around advice* 110
 - Handling parameters in advice* 112
 - *Annotating introductions* 115
- 4.4 Declaring aspects in XML 117
 - Declaring before and after advice* 118
 - *Declaring around advice* 121
 - *Passing parameters to advice* 122
 - Introducing new functionality with aspects* 124
- 4.5 Injecting AspectJ aspects 125
- 4.6 Summary 127

PART 2 SPRING ON THE WEB 129

5 *Building Spring web applications* 131

- 5.1 Getting started with Spring MVC 132
 - Following the life of a request* 132
 - *Setting up Spring MVC* 134
 - *Introducing the Spitttr application* 138
- 5.2 Writing a simple controller 139
 - Testing the controller* 140
 - *Defining class-level request handling* 142
 - *Passing model data to the view* 143
- 5.3 Accepting request input 148
 - Taking query parameters* 149
 - *Taking input via path parameters* 151
- 5.4 Processing forms 154
 - Writing a form-handling controller* 156
 - *Validating forms* 159
- 5.5 Summary 162

6 *Rendering web views* 164

- 6.1 Understanding view resolution 164
- 6.2 Creating JSP views 167
 - Configuring a JSP-ready view resolver* 167
 - *Using Spring's JSP libraries* 169
- 6.3 Defining a layout with Apache Tiles views 182
 - Configuring a Tiles view resolver* 182

- 6.4 Working with Thymeleaf 187
 - Configuring a Thymeleaf view resolver* 187
 - *Defining Thymeleaf templates* 189
- 6.5 Summary 193

7 **Advanced Spring MVC 194**

- 7.1 Alternate Spring MVC configuration 195
 - Customizing DispatcherServlet configuration* 195
 - Adding additional servlets and filters* 196
 - *Declaring DispatcherServlet in web.xml* 197
- 7.2 Processing multipart form data 200
 - Configuring a multipart resolver* 201
 - *Handling multipart requests* 205
- 7.3 Handling exceptions 208
 - Mapping exceptions to HTTP status codes* 209
 - *Writing exception-handling methods* 211
- 7.4 Advising controllers 212
- 7.5 Carrying data across redirect requests 213
 - Redirecting with URL templates* 214
 - *Working with flash attributes* 215
- 7.6 Summary 217

8 **Working with Spring Web Flow 219**

- 8.1 Configuring Web Flow in Spring 220
 - Wiring a flow executor* 220
 - *Configuring a flow registry* 221
 - *Handling flow requests* 222
- 8.2 The components of a flow 222
 - States* 223
 - *Transitions* 226
 - *Flow data* 227
- 8.3 Putting it all together: the pizza flow 229
 - Defining the base flow* 229
 - *Collecting customer information* 232
 - *Building an order* 238
 - Taking payment* 240
- 8.4 Securing web flows 242
- 8.5 Summary 242

9 **Securing web applications 244**

- 9.1 Getting started with Spring Security 245
 - Understanding Spring Security modules* 246
 - *Filtering web requests* 246
 - *Writing a simple security configuration* 248

- 9.2 Selecting user details services 250
 - Working with an in-memory user store* 251
 - *Authenticating against database tables* 252
 - *Applying LDAP-backed authentication* 255
 - *Configuring a custom user service* 259
- 9.3 Intercepting requests 260
 - Securing with Spring Expressions* 263
 - *Enforcing channel security* 264
 - *Preventing cross-site request forgery* 265
- 9.4 Authenticating users 267
 - Adding a custom login page* 268
 - *Enabling HTTP Basic authentication* 269
 - *Enabling remember-me functionality* 270
 - Logging out* 270
- 9.5 Securing the view 271
 - Using Spring Security's JSP tag library* 272
 - *Working with Thymeleaf's Spring Security dialect* 275
- 9.6 Summary 277

PART 3 SPRING IN THE BACK END.....279

10 *Hitting the database with Spring and JDBC* 281

- 10.1 Learning Spring's data-access philosophy 282
 - Getting to know Spring's data-access exception hierarchy* 283
 - Templating data access* 286
- 10.2 Configuring a data source 288
 - Using JNDI data sources* 288
 - *Using a pooled data source* 289
 - Using JDBC driver-based data sources* 291
 - *Using an embedded data source* 292
 - *Using profiles to select a data source* 293
- 10.3 Using JDBC with Spring 295
 - Tackling runaway JDBC code* 296
 - *Working with JDBC templates* 299
- 10.4 Summary 304

11 *Persisting data with object-relational mapping* 305

- 11.1 Integrating Hibernate with Spring 307
 - Declaring a Hibernate session factory* 307
 - *Building Spring-free Hibernate* 309
- 11.2 Spring and the Java Persistence API 311
 - Configuring an entity manager factory* 311
 - *Writing a JPA-based repository* 316

- 11.3 Automatic JPA repositories with Spring Data 318
 - Defining query methods* 320
 - *Declaring custom queries* 323
 - Mixing in custom functionality* 324
- 11.4 Summary 326

12 Working with NoSQL databases 327

- 12.1 Persisting documents with MongoDB 328
 - Enabling MongoDB* 329
 - *Annotating model types for MongoDB persistence* 332
 - *Accessing MongoDB with MongoTemplate* 335
 - *Writing a MongoDB repository* 337
- 12.2 Working with graph data in Neo4j 341
 - Configuring Spring Data Neo4j* 342
 - *Annotating graph entities* 344
 - *Working with Neo4jTemplate* 348
 - Creating automatic Neo4j repositories* 349
- 12.3 Working with key-value data in Redis 354
 - Connecting to Redis* 354
 - *Working with RedisTemplate* 355
 - Setting key and value serializers* 359
- 12.4 Summary 360

13 Caching data 362

- 13.1 Enabling cache support 363
 - Configuring a cache manager* 364
- 13.2 Annotating methods for caching 368
 - Populating the cache* 369
 - *Removing cache entries* 373
- 13.3 Declaring caching in XML 374
- 13.4 Summary 378

14 Securing methods 379

- 14.1 Securing methods with annotations 380
 - Restricting method access with @Secured* 380
 - *Using JSR-250's @RolesAllowed with Spring Security* 382
- 14.2 Using expressions for method-level security 383
 - Expressing method access rules* 383
 - *Filtering method inputs and outputs* 385
- 14.3 Summary 390

PART 4 INTEGRATING SPRING.....391**15 Working with remote services 393**

- 15.1 An overview of Spring remoting 394
- 15.2 Working with RMI 396
 - Exporting an RMI service* 397
 - Wiring an RMI service* 399
- 15.3 Exposing remote services with Hessian and Burlap 402
 - Exposing bean functionality with Hessian/Burlap* 402
 - Accessing Hessian/Burlap services* 405
- 15.4 Using Spring's HttpInvoker 407
 - Exposing beans as HTTP services* 407
 - Accessing services via HTTP* 408
- 15.5 Publishing and consuming web services 410
 - Creating Spring-enabled JAX-WS endpoints* 410
 - Proxying JAX-WS services on the client side* 413
- 15.6 Summary 415

16 Creating REST APIs with Spring MVC 416

- 16.1 Getting REST 417
 - The fundamentals of REST* 417
 - How Spring supports REST* 418
- 16.2 Creating your first REST endpoint 419
 - Negotiating resource representation* 421
 - Working with HTTP message converters* 426
- 16.3 Serving more than resources 432
 - Communicating errors to the client* 432
 - Setting headers in the response* 436
- 16.4 Consuming REST resources 439
 - Exploring RestTemplate's operations* 440
 - GETting resources* 441
 - Retrieving resources* 442
 - Extracting response metadata* 443
 - PUTting resources* 444
 - DELETEing resources* 445
 - POSTing resource data* 446
 - Receiving object responses from POST requests* 446
 - Receiving a resource location after a POST request* 448
 - Exchanging resources* 448
- 16.5 Summary 450

17 *Messaging in Spring* 452

- 17.1 A brief introduction to asynchronous messaging 453
 - Sending messages* 454
 - *Assessing the benefits of asynchronous messaging* 456
- 17.2 Sending messages with JMS 458
 - Setting up a message broker in Spring* 458
 - *Using Spring's JMS template* 460
 - *Creating message-driven POJOs* 469
 - *Using message-based RPC* 472
- 17.3 Messaging with AMQP 474
 - A brief introduction to AMQP* 475
 - *Configuring Spring for AMQP messaging* 477
 - *Sending messages with RabbitTemplate* 479
 - *Receiving AMQP messages* 482
- 17.4 Summary 484

18 *Messaging with WebSocket and STOMP* 485

- 18.1 Working with Spring's low-level WebSocket API 486
- 18.2 Coping with a lack of WebSocket support 491
- 18.3 Working with STOMP messaging 493
 - Enabling STOMP messaging* 495
 - *Handling STOMP messages from the client* 498
 - *Sending messages to the client* 501
- 18.4 Working with user-targeted messages 505
 - Working with user messages in a controller* 505
 - Sending messages to a specific user* 507
- 18.5 Handling message exceptions 508
- 18.6 Summary 509

19 *Sending email with Spring* 511

- 19.1 Configuring Spring to send email 512
 - Configuring a mail sender* 512
 - *Wiring and using the mail sender* 514
- 19.2 Constructing rich email messages 515
 - Adding attachments* 515
 - *Sending email with rich content* 516
- 19.3 Generating email with templates 517
 - Constructing email messages with Velocity* 518
 - *Using Thymeleaf to create email messages* 520
- 19.4 Summary 522

20 *Managing Spring beans with JMX* 523

20.1 Exporting Spring beans as MBeans 524

Exposing methods by name 527 ■ *Using interfaces to define MBean operations and attributes* 529 ■ *Working with annotation-driven MBeans* 530 ■ *Handling MBean collisions* 532

20.2 Remoting MBeans 533

Exposing remote MBeans 533 ■ *Accessing remote MBeans* 534
Proxying MBeans 536

20.3 Handling notifications 537

Listening for notifications 538

20.4 Summary 539

21 *Simplifying Spring development with Spring Boot* 540

21.1 Introducing Spring Boot 541

Adding starter dependencies 541 ■ *Autoconfiguration* 546
The Spring Boot CLI 546 ■ *The Actuator* 547

21.2 Building an application with Spring Boot 547

Handling requests 550 ■ *Creating the view* 552
Adding static artifacts 554 ■ *Persisting the data* 555
Try it out 557

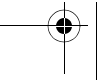
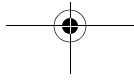
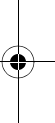
21.3 Going Groovy with the Spring Boot CLI 560

Writing a Groovy controller 560 ■ *Persisting with a Groovy repository* 563 ■ *Running the Spring Boot CLI* 564

21.4 Gaining application insight with the Actuator 565

21.5 Summary 568

index 570





preface

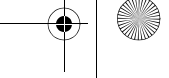
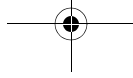
The best keeps getting better. More than a dozen years ago, Spring entered the Java development scene with the ambitious goal of simplifying enterprise Java development. It challenged the heavyweight programming models of the time with a simpler and lighter programming model based on plain old Java objects.

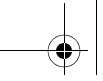
Now, several years and many releases later, we see that Spring has had a tremendous impact on enterprise application development. It has become a de facto standard framework for countless Java projects and has had an impact on the evolution of some of the specifications and frameworks that it originally set out to replace. It'd be hard to deny that the current Enterprise JavaBeans (EJB) specification may have turned out very differently had Spring not challenged earlier versions of the EJB spec.

But Spring itself continues to evolve and improve upon itself, always seeking to make the difficult development tasks simpler and empower Java developers with innovative features. Where Spring had first set out to challenge the status quo, Spring now has leapt ahead and is paving trails in Java application development.

Therefore, it's time for an updated edition of this book to expose the current state of Spring. There's so much that has happened in the past few years since the previous edition of this book; it'd be impossible to cover everything in a single edition. Nevertheless, I still tried to pack this fourth edition of *Spring in Action* with as much as I could. Here are just a few of the exciting new things that have been added in this edition:

- An emphasis on Java-based Spring configuration with Java configuration options available for almost every area of Spring development
- Conditional configuration and profiles that make runtime decisions regarding what Spring configuration should be used or ignored

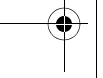
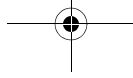




- Several enhancements and improvements to Spring MVC, especially with regard to creating REST services
- Using Thymeleaf with Spring web applications as an alternative to JSP
- Enabling Spring Security with Java-based configuration
- Using Spring Data to automatically generate repository implementations at runtime for JPA, MongoDB, and Neo4j
- Spring's new declarative caching support
- Asynchronous web messaging with WebSocket and STOMP
- Spring Boot, a game-changing new approach to working with Spring

If you're a seasoned Spring veteran, you'll find that these new elements will become valuable additions to your Spring toolkit. On the other hand, if you're new to Spring, you've picked a good time to learn Spring, and this book will help you get started.

This is, indeed, an exciting time to be working with Spring. It's been a blast to develop with Spring and write about it during the past 12 years. I can't wait to see what Spring does next!





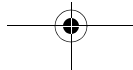
acknowledgments

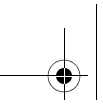
Before this book goes to press, before it is bound, before it is boxed, before it is shipped, and before you get your hands on it, there are many other hands that have touched it along the way. Even if you have an eBook copy that didn't go through that process, there were numerous hands on the bits and bytes that you downloaded—hands that edited it, reviewed it, typeset it, and proofread it. If it weren't for all of those hands, this book wouldn't exist.

First, a big thank you to everyone at Manning for working hard, for their patience when the writing wasn't moving as fast as it should have, and for prodding me along to get it done: Marjan Bace, Michael Stephens, Cynthia Kane, Andy Carroll, Benjamin Berg, Alyson Brener, Dottie Marisco, Mary Piergies, Janet Vail, and many others behind the scenes.

Getting feedback early and often is just as critical when writing a book as it is when developing software. While the pages of this book were still in a very rough form, there were several great reviewers who took the time to read the drafts and provide feedback that helped shape the final product. Thanks to the following: Bob Casazza, Chaoho Hsieh, Christophe Martini, Gregor Zurowski, James Wright, Jeelani Basha, Jens Richter, Jonathan Thoms, Josh Hart, Karen Christenson, Mario Arias, Michael Roberts, Paul Balogh, and Ricardo da Silva Lima. And special thanks to John Ryan for his thorough technical review of the manuscript shortly before it went into production.

Of course, I want to thank my beautiful wife for enduring yet another writing project and for her encouragement along the way. I love you more than you could possibly ever know.





ACKNOWLEDGMENTS

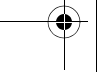
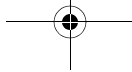


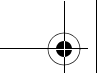
To Maisy and Madi, the most awesome little girls in the world, thank you again for your hugs, laughs, and unusual insights into what should go into the book.

To my colleagues on the Spring team, what can I say? You guys ROCK! I'm humbled and grateful for being a part of the organization that drives Spring forward. I never cease to be amazed at the never-ending awesomeness that you crank out.

And many thanks to everyone I encounter as I travel the country speaking at user groups and No Fluff/Just Stuff conferences.

Finally, thank you to the Phoenicians. You (and Epcot fans) know what you did.





about this book

The Spring Framework was created with a very specific goal in mind—to make developing Java EE applications easier. Along the same lines, *Spring in Action, Fourth Edition* was written to make learning how to use Spring easier. My goal is not to give you a blow-by-blow listing of Spring APIs. Instead, I hope to present the Spring Framework in a way that is most relevant to a Java EE developer by providing practical code examples from real-world experiences. Since Spring is a modular framework, this book was written in the same way. I recognize that not all developers have the same needs. Some may want to learn the Spring Framework from the ground up, while others may want to pick and choose different topics and go at their own pace. That way, the book can act as a tool for learning Spring for the first time as well as a guide and reference for those wanting to dig deeper into specific features.

Spring in Action, Fourth Edition is for all Java developers, but enterprise Java developers will find it particularly useful. While I will guide you along gently through code examples that build in complexity throughout each chapter, the true power of Spring lies in its ability to make enterprise applications easier to develop. Therefore, enterprise developers will most fully appreciate the examples presented in this book. Because a vast portion of Spring is devoted to providing enterprise services, many parallels can be drawn between Spring and EJB.

Roadmap

Spring in Action, Fourth Edition is divided into four parts. The first part introduces you to the essentials of the Spring Framework. Part 2 expands on that by showing how to build web applications with Spring. Part 3 steps behind the front end and shows where

Spring fits in the back end of an application. The final part shows how Spring can be used to integrate with other applications and services.

In part 1, you'll explore the Spring container, dependency injection (DI), and aspect-oriented programming...the essentials of the Spring Framework. This will give you a foundation upon which the rest of the book will build.

- In chapter 1, you'll be given an overview of Spring, including some basic examples of DI and AOP. You'll also get an overview of the greater Spring ecosystem.
- Chapter 2 goes into more detail with DI, showing you various ways that the components in your application (the "beans") can be wired together. This includes wiring with XML, Java, and automatic wiring.
- With the basics of bean wiring down, chapter 3 presents several advanced wiring techniques. You won't need these techniques that often, but when you do need them this chapter will show you how to get the most power out of the Spring container.
- Chapter 4 explores how to use Spring AOP to decouple cross-cutting concerns from the objects that they service. This chapter also sets the stage for later chapters where you'll use AOP to provide declarative services such as transactions, security, and caching.

In part 2 you'll see how to use Spring to build web applications.

- Chapter 5 covers the basics of working with Spring MVC, the foundational web framework in Spring. You'll see how to write controllers to handle web requests and respond with model data.
- Once a controller is finished with its work, the model data must be rendered using a view. Chapter 6 will explore various view technologies that can be used with Spring, including JSP, Apache Tiles, and Thymeleaf.
- Chapter 7 goes beyond the basics of Spring MVC. In this chapter, you'll learn how to customize Spring MVC configuration, handle multipart file uploads, deal with exceptions that may occur in a controller, and pass data between requests with flash attributes.
- Chapter 8 explores Spring Web Flow, an extension to Spring MVC that enables development of conversational web applications. In this chapter, you'll learn how to build web applications that lead the user through a specific, guided flow.
- In chapter 9 you'll learn how to apply security to the web layer of your application using Spring Security.

Part 3 goes behind the front end of an application and looks at how data is processed and persisted.

- Data persistence is first tackled in chapter 10 using Spring's abstraction over JDBC to work with data stored in a relational database.
- Chapter 11 takes on data persistence from another angle, using the Java Persistence API (JPA) to store data in a relational database.

- Chapter 12 looks at how Spring works with non-relational databases, such as MongoDB and Neo4j. Regardless of where the data is stored, caching can help improve performance by not hitting the database any more than necessary.
- Chapter 13 introduces you to Spring's support for declarative caching.
- Chapter 14 revisits Spring Security, showing how to use AOP to apply security at the method level.

The final part looks at ways to integrate your Spring applications with other systems.

- Chapter 15 looks at how to create and consume remote services, including RMI, Hessian, Burlap, and SOAP-based services.
- In chapter 16, Spring MVC is revisited to see how to create RESTful services using the same programming model as described previously in chapter 5.
- Chapter 17 explores Spring support for asynchronous messaging. This chapter includes working with Java Message Service (JMS) as well as the Advanced Message Queuing Protocol (AMQP).
- Asynchronous messaging takes a different twist in chapter 18 where you'll see how to use Spring with WebSocket and STOMP for asynchronous communication between the server and a client.
- Chapter 19 looks at how to send emails with Spring.
- Chapter 20 highlights Spring's management support for Java Management Extensions (JMX), enabling you to monitor and modify runtime settings for a Spring application.
- Finally, in chapter 21 you'll be introduced to a game-changing and very new way to work with Spring called Spring Boot. You'll see how Spring Boot can take away much of the boilerplate configuration required in a Spring application, enabling you to focus on the business functionality.

Code conventions and downloads

There are many code examples throughout this book. These examples will always appear in a fixed-width code font like this. Any class name, method name, or XML fragment within the normal text of the book will appear in code font as well.

Many of Spring's classes and packages have exceptionally long (but expressive) names. Because of this, line-continuation markers (`\>`) may be included when necessary.

Not all code examples in this book will be complete. Often I only show a method or two from a class to focus on a particular topic. Complete source code for the applications found throughout the book can be downloaded from the publisher's website at www.manning.com/SpringinActionFourthEdition.

Author Online

Purchase of *Spring in Action, Fourth Edition* includes free access to a private web forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the author and from other users. To

access the forum and subscribe to it, point your web browser to www.manning.com/SpringinActionFourthEdition. This page provides information on how to get on the forum once you are registered, what kind of help is available, and the rules of conduct on the forum.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the author can take place. It is not a commitment to any specific amount of participation on the part of the author, whose contribution to the book's forum remains voluntary (and unpaid). We suggest you try asking the author some challenging questions, lest his interest stray!

The Author Online forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

About the author

Craig Walls is a senior engineer with Pivotal as the project lead for Spring Social and Spring Sync, and is the author of Manning's *Spring in Action* books, now updated in this Fourth Edition. He's a zealous promoter of the Spring Framework, speaking frequently at local user groups and conferences and writing about Spring. When he's not slinging code, Craig spends as much time as he can with his wife, two daughters, two birds, and two dogs.

About the cover illustration

The figure on the cover of *Spring in Action, Fourth Edition*, is "Le Caraco," or an inhabitant of the province of Karak in southwest Jordan. Its capital is the city of Al-Karak, which boasts an ancient hilltop castle with magnificent views of the Dead Sea and surrounding plains. The illustration is taken from a French travel book, *Encyclopédie des Voyages* by J. G. St. Sauveur, published in 1796. Travel for pleasure was a relatively new phenomenon at the time and travel guides such as this one were popular, introducing both the tourist as well as the armchair traveler to the inhabitants of other regions of France and abroad.

The diversity of the drawings in the *Encyclopédie des Voyages* speaks vividly of the distinctiveness and individuality of the world's towns and provinces just two hundred years ago. This was a time when the dress codes of two regions separated by a few dozen miles identified people uniquely as belonging to one or the other. The travel guide brings to life a sense of isolation and distance of that period, and of every other historic period except our own hyperkinetic present.

Dress codes have changed since then and the diversity by region, so rich at the time, has faded away. It is now often hard to tell the inhabitants of one continent from another. Perhaps, trying to view it optimistically, we have traded a cultural and visual diversity for a more varied personal life—or a more varied and interesting intellectual and technical life.

We at Manning celebrate the inventiveness, the initiative, and the fun of the computer business with book covers based on the rich diversity of regional life two centuries ago brought back to life by the pictures from this travel guide.