

covers Spring 4

Spring IN ACTION

FOURTH EDITION

Craig Walls



 MANNING

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 SPRING1

- 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 SPRING.....1

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 WEB129

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