

Spring in Practice

WILLIE WHEELER
with JOSHUA WHITE



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 261
Shelter Island, NY 11964
Email: orders@manning.com

©2013 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
PO Box 261
Shelter Island, NY 11964

Development editor: Cynthia Kane
Technical editor: Doug Warren
Copyeditor: Tiffany Taylor
Proofreader: Elizabeth Martin
Typesetter: Gordan Salinovic
Cover designer: Marija Tudor

ISBN 9781935182054

Printed in the United States of America

1 2 3 4 5 6 7 8 9 10 – MAL – 18 17 16 15 14 13

brief contents

- 1 ■ Introducing Spring: the dependency injection container 1
- 2 ■ Data persistence, ORM, and transactions 33
- 3 ■ Building web applications with Spring Web MVC 65
- 4 ■ Basic web forms 105
- 5 ■ Enhancing Spring MVC applications with Web Flow 134
- 6 ■ Authenticating users 173
- 7 ■ Authorizing user requests 209
- 8 ■ Communicating with users and customers 244
- 9 ■ Creating a rich-text comment engine 277
- 10 ■ Integration testing 306
- 11 ■ Building a configuration management database 338
- 12 ■ Building an article-delivery engine 392
- 13 ■ Enterprise integration 422
- 14 ■ Creating a Spring-based “site-up” framework 467

contents

preface xiii
acknowledgments xv
about Spring xvii
about this book xix
about the cover illustration xxiii

1 *Introducing Spring: the dependency injection container* 1

- 1.1 What is Spring, and why use it? 2
 - The major pieces of the framework* 2 ▪ *Why use it?* 4
- 1.2 Flexible configuration via dependency injection 5
 - Configuring dependencies the old way* 5 ▪ *Dependency injection* 6 ▪ *Inversion of control* 8
- 1.3 A simple bean configuration example 9
 - Creating the account domain object* 9 ▪ *Creating the account DAO interface and implementation* 10 ▪ *Configuring CsvAccountDao with Spring* 12 ▪ *Creating the account service that finds delinquent accounts* 13 ▪ *Wiring up the AccountService to CsvAccountDao* 14
- 1.4 Wiring beans using XML 16
 - An overview of the beans namespace* 16 ▪ *Bean scopes* 20
 - The p namespace* 23 ▪ *The c namespace* 24

- 1.5 Autowiring and component scanning using annotations 25
 - @Autowired* 26
 - *Stereotype annotations* 28
 - *Component scanning* 29
 - *XML vs. annotations: which is better?* 31
- 1.6 Summary 32

2 *Data persistence, ORM, and transactions* 33

- 2.1 Data access using JDBC 35
- 2.2 Looking up a DataSource with JNDI 39
- 2.3 Object-relational mapping and transactions via Hibernate 42
- 2.4 Creating a data access layer 52
- 2.5 Working with JPA (optional) 57
- 2.6 Spring Data JPA overview (optional) 61
- 2.7 Summary 63

3 *Building web applications with Spring Web MVC* 65

- 3.1 Spring Web MVC background 66
 - A review of the model-view-controller (MVC) pattern* 66
 - *What is Spring Web MVC?* 67
 - *An architectural overview of Spring Web MVC* 68
- 3.2 Creating your first Spring Web MVC application 69
 - Configuring the application* 69
 - *A simple domain object* 70
 - Writing a basic controller* 71
 - *Implementing the master and details views* 73
- 3.3 Serving and processing forms 74
 - Using domain objects as form beans* 74
 - *Adding a controller* 76
 - Adding a form JSP and a “thanks” JSP* 78
 - *Updating the application context* 79
 - *Adding redirect-after-post behavior* 79
 - *Adding form-binding whitelisting* 80
 - *Adding form validation* 81
- 3.4 Configuring Spring Web MVC: web.xml 81
- 3.5 Configuring Spring Web MVC: the application context 83
 - Configuring HandlerMappings* 84
 - *Configuring HandlerAdapters* 89
 - Configuring HandlerExceptionResolvers* 89
 - *Configuring ViewResolvers* 89
 - *Configuring a RequestToViewNameTranslator* 92
 - Configuring other resolvers* 93
- 3.6 Spring Mobile technology preview 93
 - A brief anatomy of an HTTP request* 94
 - *Detecting a mobile device with Spring Mobile* 95
 - *Configuring Spring Mobile* 96
 - *Handling site preferences* 98
 - *Using JavaScript frameworks for enhanced look and feel* 100
 - *Switching to a separate mobile site* 102

- 3.7 Related technologies 103
 - Spring Web Flow* 103 ▪ *Spring JavaScript* 103 ▪ *Spring Faces* 103 ▪ *Spring Security* 104 ▪ *RESTful web services* 104
- 3.8 Summary 104

4 *Basic web forms* 105

- 4.1 Displaying a web form 105
- 4.2 Externalizing strings in the view 112
- 4.3 Validating form data 115
- 4.4 Saving form data 124
- 4.5 Summary 133

5 *Enhancing Spring MVC applications with Web Flow* 134

- 5.1 Is Spring Web Flow right for you? 135
- 5.2 An overview of Spring Web Flow 135
 - Defining a flow* 136 ▪ *The five types of states* 136 ▪ *Transitions between states* 140 ▪ *Flow data* 141
- 5.3 The Spring Soccer Club demo application 143
- 5.4 Using action classes 156
- 5.5 Form data binding 160
- 5.6 Form validation 162
- 5.7 Flow and state inheritance 165
- 5.8 Securing web flows 167
- 5.9 Summary 172

6 *Authenticating users* 173

- 6.1 Implementing login and logout with remember-me authentication 173
- 6.2 Customizing the login page 182
- 6.3 Implementing an always-resident login form 185
- 6.4 Sourcing user data from a database 188
- 6.5 Customizing the user database schema 191
- 6.6 Using a custom user service and user principal 193
- 6.7 Secure user passwords in the database 201
- 6.8 Auto-authenticating the user after a successful registration 206
- 6.9 Summary 208

- 7 *Authorizing user requests* 209**
 - 7.1 Authorizing Java methods using authentication levels, roles, and permissions 211
 - 7.2 Authorizing JSP views using authentication levels, roles, and permissions 217
 - 7.3 Authorizing web resources using authentication levels, roles, and permissions 220
 - 7.4 Authorizing method invocations based on ACLs 223
 - 7.5 Displaying web navigation and content based on ACLs 241
 - 7.6 Summary 243

- 8 *Communicating with users and customers* 244**
 - 8.1 Create a web-based Contact Us form 245
 - 8.2 Autogenerate an email response and email notification 252
 - 8.3 Speeding up autogenerated emails 258
 - 8.4 Allowing users to subscribe to a mailing list 262
 - 8.5 Publishing a news feed 272
 - 8.6 Summary 276

- 9 *Creating a rich-text comment engine* 277**
 - 9.1 Creating a basic user comment engine 278
 - 9.2 Integrating the comment engine with an article-delivery service 287
 - 9.3 Adding rich-text support to the comment engine 295
 - 9.4 Testing the HTML filter 302
 - 9.5 Summary 305

- 10 *Integration testing* 306**
 - 10.1 Configuring Maven for integration testing 307
 - 10.2 Writing transactional happy-path integration tests 313
 - 10.3 Verifying that code under test throws an exception 325
 - 10.4 Creating integration tests that verify performance 328
 - 10.5 Ignoring a test 331
 - 10.6 Running integration tests against an embedded database 332
 - 10.7 Summary 337

- 11** *Building a configuration management database* 338
 - 11.1 Creating a simple configuration item 342
 - 11.2 Creating related configuration items 347
 - 11.3 Adding a RESTful web service 359
 - 11.4 Updating the CMDB after successful builds 369
 - 11.5 Sourcing public GitHub data 374
 - 11.6 Sourcing private GitHub data 379
 - 11.7 Encrypting access tokens for production use 388
 - 11.8 Summary 391

- 12** *Building an article-delivery engine* 392
 - 12.1 Storing articles in a content repository 394
 - 12.2 Creating a web-based article-delivery engine 405
 - 12.3 Storing articles in a document repository 416
 - 12.4 Summary 420

- 13** *Enterprise integration* 422
 - 13.1 A whirlwind tour of Spring Integration 423
 - 13.2 Integrating applications via a shared database 425
 - 13.3 Decoupling applications with RESTful web services 430
 - 13.4 Implementing a message bus using RabbitMQ and Spring Integration 438
 - 13.5 Sourcing tickets from an IMAP store 456
 - 13.6 Send confirmation messages over SMTP 462
 - 13.7 Summary 465

- 14** *Creating a Spring-based “site-up” framework* 467
 - 14.1 Circuit-breaker overview 468
 - 14.2 Creating a circuit-breaker template and callback 469
 - 14.3 Exposing the circuit breaker as a JMX MBean 482
 - 14.4 Supporting AOP-based configuration 487
 - 14.5 Supporting custom namespaces 494
 - 14.6 Supporting annotation-based configuration 502
 - 14.7 Summary 512

<i>appendix</i>	<i>Working with the sample code</i>	513
	<i>index</i>	519