

covers Spring 3.0

Spring IN ACTION

THIRD EDITION

Craig Walls

 MANNING

www.it-ebooks.info



Praise for *Spring in Action*

This is an excellent book. It is very well written. Examples are very concise and easy to follow.

—Sunil Parikh, DZone

5 out of 5 stars ... a great instructive book.

—Nicola Pedot, Java User Group Trento

You will learn how to use Spring to write simpler, easier-to-maintain code so that you can focus on what really matters—your critical business needs.

—Springframework.org

Encyclopedic and eminently readable. Five stars all around!

—JavaLobby.org

Superbly organized and fluently written.

—Internet Bookwatch

Easy to read ... with just enough humor mixed in.

—Books-On-Line

A rare book.

—Computing Reviews

Best overall introduction to Spring.

—Taruvai Subramaniam, Amazon reader

“Really pushes Spring into Action.”

—Patrick Steger, Zühlke Engineering

“Tremendous focus and fun to read ... zooms in on things developers need to know.”

—Doug Warren, Java Web Services

Spring in Action

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

©2011 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 editors: Sebastian Stirling
Copyeditor: Benjamin Berg
Proofreader: Katie Tennant
Typesetter: Dottie Marsico
Cover designer: Marija Tudor

ISBN 9781935182351
Printed in the United States of America
1 2 3 4 5 6 7 8 9 10 – MAL – 16 15 14 13 12 11

brief contents

PART 1	CORE SPRING	1
1	■ Springing into action	3
2	■ Wiring beans	30
3	■ Minimizing XML configuration in Spring	64
4	■ Aspect-oriented Spring	84
PART 2	SPRING APPLICATION ESSENTIALS	111
5	■ Hitting the database	113
6	■ Managing transactions	146
7	■ Building web applications with Spring MVC	164
8	■ Working with Spring Web Flow	199
9	■ Securing Spring	224
PART 3	INTEGRATING SPRING	253
10	■ Working with remote services	255
11	■ Giving Spring some REST	277
12	■ Messaging in Spring	310
13	■ Managing Spring beans with JMX	333
14	■ Odds and ends	350

contents

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

PART 1 CORE SPRING.....1

1 *Springing into action* 3

- 1.1 Simplifying Java development 4
 - Unleashing the power of POJOs* 5 ■ *Injecting dependencies* 6
 - Applying aspects* 10 ■ *Eliminating boilerplate code with templates* 15
- 1.2 Containing your beans 17
 - Working with an application context* 18 ■ *A bean's life* 19
- 1.3 Surveying the Spring landscape 20
 - Spring modules* 20 ■ *The Spring portfolio* 23
- 1.4 What's new in Spring 27
 - What's new in Spring 2.5?* 27 ■ *What's new in Spring 3.0?* 28
 - What's new in the Spring portfolio?* 28
- 1.5 Summary 29

2 *Wiring beans* 30

2.1 Declaring beans 31

Setting up Spring configuration 32 ■ *Declaring a simple bean* 33 ■ *Injecting through constructors* 34 ■ *Bean scoping* 38 ■ *Initializing and destroying beans* 39

2.2 Injecting into bean properties 41

Injecting simple values 42 ■ *Referencing other beans* 43
Wiring properties with Spring's `p` namespace 46 ■ *Wiring collections* 47 ■ *Wiring nothing (null)* 52

2.3 Wiring with expressions 52

Expressing SpEL fundamentals 53 ■ *Performing operations on SpEL values* 56 ■ *Sifting through collections in SpEL* 59

2.4 Summary 63

3 *Minimizing XML configuration in Spring* 64

3.1 Automatically wiring bean properties 65

The four kinds of autowiring 65 ■ *Default autowiring* 68
Mixing auto with explicit wiring 69

3.2 Wiring with annotations 70

Using `@Autowired` 71 ■ *Applying standards-based autowiring with `@Inject`* 74 ■ *Using expressions with annotation injection* 76

3.3 Automatically discovering beans 77

Annotating beans for autodiscovery 78
Filtering component-scans 79

3.4 Using Spring's Java-based configuration 80

Setting up for Java-based configuration 80 ■ *Defining a configuration class* 81 ■ *Declaring a simple bean* 81
Injecting with Spring's Java-based configuration 82

3.5 Summary 83

4 *Aspect-oriented Spring* 84

4.1 What's aspect-oriented programming? 85

Defining AOP terminology 86 ■ *Spring's AOP support* 88

4.2 Selecting join points with pointcuts 91

Writing pointcuts 92 ■ *Using Spring's `bean()` designator* 93

- 4.3 Declaring aspects in XML 93
 - Declaring before and after advice* 95 ■ *Declaring around advice* 97 ■ *Passing parameters to advice* 98 ■ *Introducing new functionality with aspects* 100
- 4.4 Annotating aspects 102
 - Annotating around advice* 104 ■ *Passing arguments to annotated advice* 105 ■ *Annotating introductions* 106
- 4.5 Injecting AspectJ aspects 107
- 4.6 Summary 110

PART 2 SPRING APPLICATION ESSENTIALS.....111

5 *Hitting the database* 113

- 5.1 Learning Spring's data access philosophy 114
 - Getting to know Spring's data access exception hierarchy* 115
 - Templating data access* 117 ■ *Using DAO support classes* 119
- 5.2 Configuring a data source 121
 - Using JNDI data sources* 121 ■ *Using a pooled data source* 122 ■ *JDBC driver-based data source* 123
- 5.3 Using JDBC with Spring 124
 - Tackling runaway JDBC code* 124 ■ *Working with JDBC templates* 127
- 5.4 Integrating Hibernate with Spring 132
 - A Hibernate overview* 134 ■ *Declaring a Hibernate session factory* 134 ■ *Building Spring-free Hibernate* 137
- 5.5 Spring and the Java Persistence API 138
 - Configuring an entity manager factory* 139 ■ *Writing a JPA-based DAO* 143
- 5.6 Summary 144

6 *Managing transactions* 146

- 6.1 Understanding transactions 147
 - Explaining transactions in only four words* 148
 - Understanding Spring's transaction management support* 149

- 6.2 Choosing a transaction manager 150
 - JDBC transactions 151 ■ Hibernate transactions 151 ■ Java Persistence API transactions 152 ■ Java transaction API transactions 153*
- 6.3 Programming transactions in Spring 153
- 6.4 Declaring transactions 155
 - Defining transaction attributes 156 ■ Declaring transactions in XML 160 ■ Defining annotation-driven transactions 162*
- 6.5 Summary 163
- 7 Building web applications with Spring MVC 164**
 - 7.1 Getting started with Spring MVC 165
 - Following a request through Spring MVC 165 ■ Setting up Spring MVC 167*
 - 7.2 Writing a basic controller 169
 - Configuring an annotation-driven Spring MVC 170 ■ Defining the home page controller 170 ■ Resolving views 173 ■ Defining the home page view 177 ■ Rounding out the Spring application context 179*
 - 7.3 Handling controller input 181
 - Writing a controller that processes input 181 ■ Rendering the view 183*
 - 7.4 Processing forms 185
 - Displaying the registration form 185 ■ Processing form input 187 ■ Validating input 189*
 - 7.5 Handling file uploads 193
 - Adding a file upload field to the form 193 ■ Receiving uploaded files 194 ■ Configuring Spring for file uploads 197*
 - 7.6 Summary 197

- 8 Working with Spring Web Flow 199**
 - 8.1 Installing Spring Web Flow 200
 - Configuring Web Flow in Spring 200*
 - 8.2 The components of a flow 203
 - States 203 ■ Transitions 206 ■ Flow data 207*
 - 8.3 Putting it all together: the pizza flow 209
 - Defining the base flow 209 ■ Collecting customer information 213 ■ Building an order 218 ■ Taking payment 221*

8.4 Securing web flows 222

8.5 Summary 223

9 *Securing Spring* 224

9.1 Introducing Spring Security 225

Getting started with Spring Security 226 ■ *Using the Spring Security configuration namespace* 226

9.2 Securing web requests 227

Proxying servlet filters 228 ■ *Configuring minimal web security* 228 ■ *Intercepting requests* 232

9.3 Securing view-level elements 235

Accessing authentication details 235 ■ *Rendering with authorities* 236

9.4 Authenticating users 238

Configuring an in-memory user repository 239
Authenticating against a database 240 ■ *Authenticating against LDAP* 241 ■ *Enabling remember-me functionality* 245

9.5 Securing methods 246

Securing methods with @Secured 246 ■ *Using JSR-250's @RolesAllowed* 247 ■ *Pre-/Post-invocation security with SpEL* 247 ■ *Declaring method-level security pointcuts* 252

9.6 Summary 252

PART 3 INTEGRATING SPRING.....253

10 *Working with remote services* 255

10.1 An overview of Spring remoting 256

10.2 Working with RMI 258

Exporting an RMI service 259 ■ *Wiring an RMI service* 261

10.3 Exposing remote services with Hessian and Burlap 263

Exposing bean functionality with Hessian/Burlap 264
Accessing Hessian/Burlap services 266

10.4 Using Spring's HttpInvoker 268

Exposing beans as HTTP services 268 ■ *Accessing services via HTTP* 269

10.5 Publishing and consuming web services 270

Creating Spring-enabled JAX-WS endpoints 271

Proxying JAX-WS services on the client side 274

10.6 Summary 276

11 **Giving Spring some REST 277**

11.1 Getting REST 278

The fundamentals of REST 278 ■ *How Spring supports REST* 279

11.2 Writing resource-oriented controllers 279

Dissecting a RESTless controller 280 ■ *Handling RESTful URLs* 281 ■ *Performing the REST verbs* 284

11.3 Representing resources 287

Negotiating resource representation 288 ■ *Working with HTTP message converters* 291

11.4 Writing REST clients 294

Exploring RestTemplate's operations 295 ■ *GETting resources* 296 ■ *PUTting resources* 299 ■ *DELETE-ing resources* 301 ■ *POSTing resource data* 301 ■ *Exchanging resources* 304

11.5 Submitting RESTful forms 306

Rendering hidden method fields in JSP 306 ■ *Unmasking the real request* 307

11.6 Summary 309

12 **Messaging in Spring 310**

12.1 A brief introduction to JMS 311

Architecting JMS 312 ■ *Assessing the benefits of JMS* 314

12.2 Setting up a message broker in Spring 316

Creating a connection factory 316 ■ *Declaring an ActiveMQ message destination* 317

12.3 Using Spring's JMS template 318

Tackling runaway JMS code 318 ■ *Working with JMS templates* 319

12.4 Creating message-driven POJOs 325

Creating a message listener 326 ■ *Configuring message listeners* 327

- 12.5 Using message-based RPC 327
 - Working with Spring message-based RPC* 328 ■ *Asynchronous RPC with Lingo* 330
- 12.6 Summary 332

13 *Managing Spring beans with JMX* 333

- 13.1 Exporting Spring beans as MBeans 334
 - Exposing methods by name* 337 ■ *Using interfaces to define MBean operations and attributes* 339 ■ *Working with annotation-driven MBeans* 340 ■ *Handling MBean collisions* 342
- 13.2 Remoting MBeans 343
 - Exposing remote MBeans* 343 ■ *Accessing remote MBeans* 344
 - Proxying MBeans* 345
- 13.3 Handling notifications 346
 - Listening for notifications* 348
- 13.4 Summary 349

14 *Odds and ends* 350

- 14.1 Externalizing configuration 351
 - Replacing property placeholders* 352 ■ *Overriding properties* 354 ■ *Encrypting external properties* 355
- 14.2 Wiring JNDI objects 357
 - Working with conventional JNDI* 357 ■ *Injecting JNDI objects* 359 ■ *Wiring EJBs in Spring* 362
- 14.3 Sending email 363
 - Configuring a mail sender* 363 ■ *Constructing the email* 365
- 14.4 Scheduling and background tasks 370
 - Declaring scheduled methods* 371 ■ *Declaring asynchronous methods* 373
- 14.5 Summary 374
- 14.6 The end...? 374
 - index* 377