

SPRING IN ACTION

Craig Walls
Ryan Breidenbach

 MANNING



Spring in Action

Spring in Action

CRAIG WALLS
RYAN BREIDENBACH



MANNING

Greenwich
(74° w. long.)

For online information and ordering of this and other Manning books, please go to 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.

209 Bruce Park Avenue
Greenwich, CT 06830

Fax: (203) 661-9018

email: orders@manning.com

©2005 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 they publish printed on acid-free paper, and we exert our best efforts to that end.



Manning Publications Co.
209 Bruce Park Avenue
Greenwich, CT 06830

Copyeditor: Liz Welch
Typesetter: Denis Dalinnik
Cover designer: Leslie Haimes

ISBN 1-932394-35-4

Printed in the United States of America

1 2 3 4 5 6 7 8 9 10 – VH G – 09 08 07 06 05

Licensed to Lars Helge ??verland <larshelge@gmail.com>

Maisy Grace, see you soon
—C.W.

For my brother, Lee
—R.B.

brief contents

PART 1 SPRING ESSENTIALS 1

- 1 ■ A Spring jump start 3
- 2 ■ Wiring beans 42
- 3 ■ Creating aspects 91

PART 2 SPRING IN THE BUSINESS LAYER 131

- 4 ■ Hitting the database 133
- 5 ■ Managing transactions 173
- 6 ■ Remoting 207
- 7 ■ Accessing enterprise services 240

PART 3 SPRING IN THE WEB LAYER 267

- 8 ■ Building the web layer 269
- 9 ■ View layer alternatives 319
- 10 ■ Working with other web frameworks 346
- 11 ■ Securing Spring applications 367

contents

preface xvii
acknowledgments xx
about this book xxiii

PART 1 SPRING ESSENTIALS 1

- 1** ***A Spring jump start*** 3
- 1.1 Why Spring? 5
 - A day in the life of a J2EE developer* 5 ■ *Spring's pledge* 6
 - 1.2 What is Spring? 8
 - Spring modules* 9
 - 1.3 Spring jump start 12
 - 1.4 Understanding inversion of control 15
 - Injecting dependencies* 16 ■ *IoC in action* 16
 - IoC in enterprise applications* 23
 - 1.5 Applying aspect-oriented programming 25
 - Introducing AOP* 25 ■ *AOP in action* 27 ■ *AOP in the enterprise* 30

1.6 Spring alternatives 33

Comparing Spring to EJB 33 ▪ *Considering other lightweight containers 36* ▪ *Web frameworks 38* ▪ *Persistence frameworks 40*

1.7 Summary 40

2 **Wiring beans 42**

2.1 Containing your beans 44

Introducing the BeanFactory 44 ▪ *Working with an application context 46* ▪ *A bean's life 47*

2.2 Basic wiring 50

Wiring with XML 54 ▪ *Adding a bean 55* ▪ *Injecting dependencies via setter methods 58* ▪ *Injecting dependencies via constructor 65*

2.3 Autowiring 69

Handling ambiguities of autowiring 71 ▪ *Mixing auto and explicit wiring 72* ▪ *Autowiring by default 72* ▪ *To autowire or not to autowire 72*

2.4 Working with Spring's special beans 73

Postprocessing beans 74 ▪ *Postprocessing the bean factory 76* ▪ *Externalizing the configuration 78* ▪ *Customizing property editors 80* ▪ *Resolving text messages 83* ▪ *Listening for events 85* ▪ *Publishing events 86* ▪ *Making beans aware 87*

2.5 Summary 90

3 **Creating aspects 91**

3.1 Introducing AOP 92

Defining AOP terminology 93 ▪ *Spring's AOP implementation 95*

3.2 Creating advice 97

Before advice 99 ▪ *After advice 101* ▪ *Around advice 102* ▪ *Throws advice 104* ▪ *Introduction advice 105*

3.3 Defining pointcuts 105

Defining a pointcut in Spring 105 ▪ *Understanding advisors 107* ▪ *Using Spring's static pointcuts 107* ▪ *Using dynamic pointcuts 111* ▪ *Pointcut operations 113*

- 3.4 Creating introductions 115
 - Implementing IntroductionInterceptor* 115 ▪ *Creating an IntroductionAdvisor* 119 ▪ *Using introduction advice carefully* 120
- 3.5 Using ProxyFactoryBean 122
- 3.6 Autoproxying 124
 - BeanNameAutoProxyCreator* 124 ▪ *DefaultAdvisorAutoProxyCreator* 126 ▪ *Metadata autoproxying* 128
- 3.7 Summary 128

PART 2 SPRING IN THE BUSINESS LAYER..... 131

4

Hitting the database 133

- 4.1 Learning Spring's DAO philosophy 134
 - Understanding Spring's DataAccessException* 135
 - Working with DataSources* 137 ▪ *Consistent DAO support* 139
- 4.2 Using JDBC with Spring 141
 - The problem with JDBC code* 142 ▪ *Using JdbcTemplate* 144
 - Creating operations as objects* 152 ▪ *Auto-incrementing keys* 155
- 4.3 Introducing Spring's ORM framework support 156
- 4.4 Integrating Hibernate with Spring 157
 - Hibernate overview* 157 ▪ *Managing Hibernate resources* 159 ▪ *Accessing Hibernate through HibernateTemplate* 162 ▪ *Subclassing HibernateDaoSupport* 163
- 4.5 Spring and JDO 164
 - Configuring JDO* 164 ▪ *Accessing data with JdoTemplate* 165
- 4.6 Spring and iBATIS 166
 - Setting up SQL Maps* 167
 - Using SqlMapClientTemplate* 168
- 4.7 Spring and OJB 169
 - Setting up OJB's PersistenceBroker* 169
- 4.8 Summary 171

5	Managing transactions 173
5.1	Understanding transactions 174
	<i>Explaining transactions in only four words 176</i> ■ <i>Understanding Spring's transaction management support 177</i> ■ <i>Introducing Spring's transaction manager 178</i>
5.2	Programming transactions in Spring 181
5.3	Declaring transactions 183
	<i>Understanding transaction attributes 185</i> ■ <i>Declaring a simple transaction policy 189</i>
5.4	Declaring transactions by method name 191
	<i>Using NameMatchTransactionAttributeSource 191</i> <i>Shortcutting name-matched transactions 194</i>
5.5	Declaring transactions with metadata 195
	<i>Sourcing transaction attributes from metadata 196</i> <i>Declaring transactions with Commons Attributes 197</i>
5.6	Trimming down transaction declarations 201
	<i>Inheriting from a parent TransactionProxyFactoryBean 202</i> <i>Autoproxying transactions 203</i>
5.7	Summary 206
6	Remoting 207
6.1	Spring remoting overview 208
6.2	Working with RMI 212
	<i>Wiring RMI services 212</i> ■ <i>Exporting RMI services 214</i>
6.3	Remoting with Hessian and Burlap 218
	<i>Accessing Hessian/Burlap services 219</i> ■ <i>Exposing bean functionality with Hessian/Burlap 220</i>
6.4	Using Http invoker 223
	<i>Accessing services via HTTP 224</i> ■ <i>Exposing beans as HTTP Services 225</i>
6.5	Working with EJBs 226
	<i>Accessing EJBs 227</i> ■ <i>Developing Spring-enabled EJBs 231</i>
6.6	Using JAX-RPC web services 233
	<i>Referencing a web service with JAX-RPC 234</i> ■ <i>Wiring a web service in Spring 236</i>
6.7	Summary 238

- ## 7 **Accessing enterprise services 240**
- 7.1 Retrieving objects from JNDI 241
 - Working with conventional JNDI 241* ▪ *Proxying JNDI objects 243*
 - 7.2 Sending e-mail 244
 - 7.3 Scheduling tasks 248
 - Scheduling with Java's Timer 248* ▪ *Using the Quartz scheduler 250* ▪ *Invoking methods on a schedule 254*
 - 7.4 Sending messages with JMS 256
 - Sending messages with JMS templates 257* ▪ *Consuming messages 261* ▪ *Converting messages 263*
 - 7.5 Summary 266

PART 3 SPRING IN THE WEB LAYER..... 267

- ## 8 **Building the web layer 269**
- 8.1 Getting started with Spring MVC 270
 - A day in the life of a request 271* ▪ *Configuring DispatcherServlet 272* ▪ *Spring MVC in a nutshell 275*
 - 8.2 Mapping requests to controllers 279
 - Mapping URLs to bean names 280* ▪ *Using SimpleUrlHandlerMapping 281* ▪ *Using metadata to map controllers 281* ▪ *Working with multiple handler mappings 282*
 - 8.3 Handling requests with controllers 283
 - Writing a simple controller 285* ▪ *Processing commands 287* ▪ *Processing form submissions 289* ▪ *Processing complex forms with wizards 294* ▪ *Handling multiple actions in one controller 301* ▪ *Working with Throwaway controllers 305*
 - 8.4 Resolving views 307
 - Using template views 308* ▪ *Resolving view beans 310* ▪ *Choosing a view resolver 313*
 - 8.5 Using Spring's bind tag 314
 - 8.6 Handling exceptions 317
 - 8.7 Summary 317

9 **View layer alternatives** 319

9.1 Using Velocity templates 321

Defining the Velocity view 321 ▪ *Configuring the Velocity engine* 322 ▪ *Resolving Velocity views* 323 ▪ *Formatting dates and numbers* 324 ▪ *Exposing request and session attributes* 325 ▪ *Binding form fields in Velocity* 326

9.2 Working with FreeMarker 327

Constructing a FreeMarker view 328 ▪ *Configuring the FreeMarker engine* 329 ▪ *Resolving FreeMarker views* 330 ▪ *Binding form fields in FreeMarker* 330

9.3 Designing page layout with Tiles 332

Tile views 332 ▪ *Tile controllers* 335

9.4 Generating non-HTML output 337

Producing Excel spreadsheets 338 ▪ *Generating PDF documents* 340 ▪ *Generating other non-HTML files* 343

9.5 Summary 344

10 **Working with other web frameworks** 346

10.1 Working with Jakarta Struts 347

Registering the Spring plug-in 348 ▪ *Implementing Spring-aware Struts actions* 348 ▪ *Delegating actions* 350

10.2 Working with Tapestry 352

Replacing the Tapestry Engine 353 ▪ *Loading Spring beans into Tapestry pages* 355

10.3 Integrating with JavaServer Faces 357

Resolving variables 357 ▪ *Publishing request handled events* 361

10.4 Integrating with WebWork 362

WebWork 1 363 ▪ *XWork/WebWork2* 364

10.5 Summary 365

11 **Securing Spring applications** 367

11.1 Introducing the Acegi Security System 368

Security interceptors 369 ▪ *Authentication managers* 370 ▪ *Access decisions managers* 370 ▪ *Run-as managers* 370

11.2	Managing authentication	371
	<i>Configuring a provider manager</i>	371
	▪ <i>Authenticating against a database</i>	373
	▪ <i>Authenticating against an LDAP repository</i>	382
	▪ <i>Enabling Single Sign-On with Acegi and Yale CAS</i>	384
11.3	Controlling access	389
	<i>Voting access decisions</i>	389
	▪ <i>Deciding how to vote</i>	390
	<i>Handling voter abstinence</i>	392
11.4	Securing web applications	392
	<i>Proxying Acegi's filters</i>	394
	▪ <i>Enforcing web security</i>	397
	<i>Processing a login</i>	400
	▪ <i>Setting up the security context</i>	406
	<i>Ensuring a secure channel</i>	407
	▪ <i>Using the Acegi tag library</i>	411
11.5	Securing method invocations	412
	<i>Creating a security aspect</i>	412
	▪ <i>Securing methods using metadata</i>	414
11.6	Summary	416
<i>appendix A:</i>	<i>Spring setup</i>	417
A.1	Downloading Spring	418
A.2	Choosing a distribution	418
A.3	Setting up your project	419
A.4	Building with Ant	420
<i>appendix B:</i>	<i>Spring-related projects</i>	422
B.1	AppFuse	423
B.2	Rich Client Project	424
B.3	Spring.NET	424
	<i>index</i>	427