

Spring Recipes

A Problem-Solution Approach
Third Edition



Marten Deinum

Josh Long

Gary Mak

Daniel Rubio

Apress®

Spring Recipes: A Problem-Solution Approach

Copyright © 2014 by Marten Deinum, Josh Long, Gary Mak and Daniel Rubio

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN-13 (pbk): 978-1-4302-5908-4

ISBN-13 (electronic): 978-1-4302-5909-1

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr

Lead Editor: Steve Anglin

Development Editor: Matthew Moodie

Technical Reviewer: Felipe Gutierrez

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Louise Corrigan, Jonathan Gennick,

Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper,

Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Steve Weiss

Coordinating Editor: Anamika Panchoo

Copy Editor: Lori Cavanaugh

Composer: SPi Global

Indexer: SPi Global

Artist: SPi Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science+Business Media Finance Inc. (SSBM Finance Inc.). SSBM Finance Inc. is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales-eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this text is available to readers at www.apress.com/9781430259084. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

*For my wife Djoke, my daughter Geeske and my, at the moment, unborn child.
I love you.*

—Marten Deinum

Contents at a Glance

About the Authors	xxxiii
About the Technical Reviewer	xxxv
Acknowledgments	xxxvii
Introduction	xxxix
■ Chapter 1: Spring Development Tools	1
■ Chapter 2: Spring Core Tasks	47
■ Chapter 3: Spring Annotation Driven Core Tasks	135
■ Chapter 4: Spring @MVC	217
■ Chapter 5: Spring REST	275
■ Chapter 6: Spring Social	303
■ Chapter 7: Spring Security	331
■ Chapter 8: Spring Mobile	385
■ Chapter 9: Spring with Other Web Frameworks	401
■ Chapter 10: Data Access	419
■ Chapter 11: Spring Transaction Management	475
■ Chapter 12: Spring Batch	511
■ Chapter 13: NoSQL and BigData	549
■ Chapter 14: Spring Java Enterprise Services and Remoting Technologies	591
■ Chapter 15: Spring Messaging	659

■ Chapter 16: Spring Integration	691
■ Chapter 17: Spring Testing	723
■ Chapter 18: Grails	757
Index	799

Contents

About the Authors	xxxiii
About the Technical Reviewer	xxxv
Acknowledgments	xxxvii
Introduction	xxxix
■ Chapter 1: Spring Development Tools	1
1-1. Build a Spring application with the Spring Tool Suite.....	1
Problem	1
Solution	1
How It Works.....	2
1-2. Build a Spring application with the Eclipse IDE	12
Problem	12
Solution.....	12
How It Works.....	12
1-3. Build a Spring application with the IntelliJ IDE.....	26
Problem	26
Solution.....	26
How It Works.....	26
1-4. Build a Spring application with the Maven command line interface	42
Problem	42
Solution	42
How It Works	42

1-5. Build a Spring application with the Gradle command line interface.....	43
Problem	43
Solution	43
How It Works	43
1-6. Build a Spring application with the Gradle Wrapper	45
Problem	45
Solution.....	45
How It Works.....	45
Summary	46
■ Chapter 2: Spring Core Tasks	47
2-1. Manage and Configure POJOs with the Spring IoC Container	48
Problem	48
Solution.....	48
How It Works.....	48
2-2. Create POJOs by Invoking a Constructor	53
Problem	53
Solution.....	53
How It Works.....	53
2-3. Use POJO References, Auto-Wiring, and Imports to Interact with Other POJOs	59
Problem	59
Solution.....	59
How It Works.....	60
2-4. Configure POJOs with Java Collection Attributes	68
Problem	68
Solution.....	68
How It Works.....	68
2-5. Set a POJOs Scope	77
Problem	77
Solution.....	77
How It Works.....	78

2-6. Use Data from External Resources (Text Files, XML Files, Properties Files, or Image Files) 80

 Problem 80

 Solution..... 80

 How It Works..... 80

2-7. Resolve I18N Text Messages for Different Locales in Properties Files 84

 Problem 84

 Solution..... 84

 How It Works..... 84

2-8. Customize POJO Initialization and Destruction 87

 Problem 87

 Solution..... 87

 How It Works 87

2-9. Create Post Processors to Validate and Modify POJOs 90

 Problem 90

 Solution..... 90

 How It Works..... 91

2-10. Create POJOs with a Factory (Static Method, Instance Method, Spring’s FactoryBean) 92

 Problem 92

 Solution..... 93

 How It Works..... 93

2-11. Use Spring Environments and Profiles to Load Different Sets of POJOs 97

 Problem 97

 Solution..... 97

 How It Works..... 97

2-12. Aspect Orientated Programming..... 99

 Problem 99

 Solution..... 99

 How It Works..... 99

2-13. Declare POJOs from Static Fields or Object Properties	104
Problem	104
Solution.....	104
How It Works.....	104
2-14. Making POJOs Aware of Spring's IoC Container Resources.....	107
Problem	107
Solution.....	107
How It Works.....	108
2-15. Communicate Application Events Between POJOs	109
Problem	109
Solution.....	109
How It Works.....	110
2-16. Use Property Editors in Spring.....	112
Problem	112
Solution.....	112
How It Works.....	112
2-17. Inherit POJO Configurations.....	116
Problem	116
Solution.....	116
How It Works.....	116
2-18. Implement POJOs with Scripting Languages.....	120
Problem	120
Solution.....	120
How It Works.....	120
2-19. Inject Spring POJOs into Scripts	124
Problem	124
Solution.....	124
How It Works.....	124

2-20. Refresh POJOs from Scripts 127

 Problem 127

 Solution..... 127

 How It Works..... 127

2-21. Define Script Sources as Inline Code and not in External Files 128

 Problem 128

 Solution..... 128

 How It Works..... 128

2-22. Use the Spring Expression Language to Configure POJOs 129

 Problem 129

 Solution..... 129

 How It Works..... 130

Summary 133

■ Chapter 3: Spring Annotation Driven Core Tasks..... 135

3-1. Using Java Config to configure POJOs 136

 Problem 136

 Solution..... 136

 How It Works..... 136

3-2. Create POJOs by Invoking a Constructor 142

 Problem 142

 Solution..... 142

 How It Works..... 143

3-3. Use POJO References and Auto-Wiring to Interact with other POJOs..... 145

 Problem 145

 Solution..... 145

 How It Works..... 146

3-4. Auto-wire POJOs the @Resource and @Inject annotation 151

 Problem 152

 Solution..... 152

 How It Works 152

3-5. Set a POJOs Scope with the @Scope annotation	154
Problem	154
Solution.....	154
How It Works.....	155
3-6. Use data from External Resources (Text files, XML files, properties files, or image files)	157
Problem	157
Solution.....	157
How It Works.....	157
3-7. Resolve I18N Text Messages for different locales in properties files	161
Problem	161
Solution.....	162
How It Works.....	162
3-8. Customize POJO Initialization and Destruction with annotations	164
Problem	164
Solution.....	164
How It Works.....	164
3-9. Create Post Processors to validate and modify POJOs	168
Problem	168
Solution.....	168
How It Works.....	168
3-10. Create POJOs with a factory (Static method, Instance method, Spring's FactoryBean) ..	171
Problem	171
Solution.....	171
How It Works.....	171
3-11. Use Spring Environments and Profiles to load different sets of POJOs	175
Problem	175
Solution.....	175
How It Works.....	175

3-12. Aspect Orientated programming with Annotations 178
 Problem 178
 Solution..... 178
 How It Works..... 178

3-13. Accessing the Join Point Information 186
 Problem 186
 Solution..... 186
 How It Works..... 186

3-14. Specifying Aspect Precedence with the @Order annotation 187
 Problem 187
 Solution..... 187
 How It Works..... 187

3-15. Reuse Aspect Pointcut Definitions 189
 Problem 189
 Solution..... 189
 How It Works..... 189

3-16. Writing AspectJ Pointcut Expressions..... 191
 Problem 191
 Solution..... 191
 How It Works..... 192

3-17. AOP introductions for POJOs..... 196
 Problem 196
 Solution..... 196
 How It Works..... 196

3-18. Introduce states to your POJOs with AOP 198
 Problem 198
 Solution..... 198
 How It Works..... 199

3-19. Load-Time Weaving AspectJ Aspects in Spring	201
Problem	201
Solution.....	201
How It Works.....	201
3-20. Configuring AspectJ Aspects in Spring	205
Problem	205
Solution.....	205
How It Works.....	205
3-21. Inject POJOs into Domain Objects with AOP	207
Problem	207
Solution.....	207
How It Works.....	207
3-22. Concurrency with Spring and TaskExecutors.....	209
Problem	209
Solution.....	209
How It Works.....	209
Summary.....	215
■ Chapter 4: Spring @MVC	217
4-1. Developing a Simple Web Application with Spring MVC	217
Problem	217
Solution.....	217
How It Works.....	219
4-2. Mapping requests with @RequestMapping	232
Problem	232
Solution.....	233
How It Works.....	233
4-3. Intercepting Requests with Handler Interceptors	236
Problem	236
Solution.....	236
How It Works.....	236

4-4. Resolving User Locales	239
Problem	239
Solution.....	239
How It Works.....	240
4-5. Externalizing Locale-Sensitive Text Messages	242
Problem	242
Solution.....	242
How It Works.....	242
4-6. Resolving Views by Names	243
Problem	243
Solution.....	243
How It Works.....	244
4-7. Views and Content Negotiation	247
Problem	247
Solution.....	247
How It Works.....	247
4-8. Mapping Exceptions to Views	249
Problem	249
Solution.....	249
How It Works.....	249
Mappings exceptions using @ExceptionHandler.....	251
4-9. Handling Forms with Controllers	252
Problem	252
Solution.....	252
How It Works.....	252
4-10. Bean validation with Annotations (JSR-303)	265
Problem	265
Solution.....	265
How It Works.....	266

4-11. Creating Excel and PDF Views	267
Problem	267
Solution.....	267
How It Works.....	268
Summary.....	273
■ Chapter 5: Spring REST	275
5-1. Publishing XML with REST Services	275
Problem	275
Solution.....	275
How It Works.....	276
5-2. Publishing JSON with REST services.....	284
Problem	284
Solution.....	284
How It Works.....	284
5-3. Accessing a REST Service with Spring	289
Problem	289
Solution.....	289
How It Works.....	289
5-4. Publishing RSS and Atom feeds	293
Problem	293
Solution.....	293
How It Works.....	293
Summary.....	302
■ Chapter 6: Spring Social.....	303
6-1. Setting Up Spring Social	303
Problem	303
Solution.....	303
How It Works.....	303

6-2. Connecting to Twitter	305
Problem	305
Solution.....	305
How It Works.....	305
6-3. Connecting to Facebook	310
Problem	310
Solution.....	310
How It Works.....	310
6-4. Showing Service Provider Connection Status.....	313
Problem	313
Solution.....	313
How It Works.....	313
6-5. Using the Twitter API.....	318
Problem	318
Solution.....	318
How It Works.....	318
6-6. Using a Persistent UsersConnectionRepository	319
Problem	319
Solution.....	319
How It Works.....	320
6-7. Integrating Spring Social and Spring Security	321
Problem	321
Solution.....	321
How It Works.....	322
Summary.....	330
■ Chapter 7: Spring Security	331
7-1. Securing URL Access	331
Problem	331
Solution.....	331
How It Works.....	332

7-2. Logging In to Web Applications	344
Problem	344
Solution.....	344
How It Works.....	345
7-3. Authenticating Users	351
Problem	351
Solution.....	351
How It Works.....	352
7-4. Making Access Control Decisions	365
Problem	365
Solution.....	365
How It Works.....	366
7-5. Securing Method Invocations	371
Problem	371
Solution.....	371
How It Works.....	371
7-6. Handling Security in Views	374
Problem	374
Solution.....	374
How It Works.....	374
7-7. Handling Domain Object Security	376
Problem	376
Solution.....	376
How It Works.....	376
Summary	384
■ Chapter 8: Spring Mobile	385
Recipe 8-1. Device detection without Spring Mobile	385
Problem	385
Solution.....	385
How It Works.....	385

Recipe 8-2. Device detection with Spring Mobile	390
Problem	390
Solution.....	390
How It Works.....	390
8-3. Using Site preferences.....	392
Problem	392
Solution.....	392
How It Works.....	392
8-4. Using the Device Information to Render Views.....	394
Problem	394
Solution.....	394
How It Works.....	394
8-5. Site Switching.....	398
Problem	398
Solution.....	398
How It Works.....	398
Summary.....	400
■ Chapter 9: Spring with Other Web Frameworks	401
9-1. Accessing Spring in Generic Web Applications.....	401
Problem	401
Solution.....	401
How It Works.....	401
9-2. Using Spring in Your Servlets and Filters.....	406
Problem	406
Solution.....	407
How It Works.....	407
9-3. Integrating Spring with JSF	412
Problem	412
Solution.....	412
How It Works.....	412
Summary.....	417

Chapter 10: Data Access	419
Problems with Direct JDBC	420
Setting Up the Application Database	420
Understanding the Data Access Object Design Pattern	421
Implementing the DAO with JDBC	422
Configuring a Data Source in Spring	424
Running the DAO	425
Taking It a Step Further	426
10-1. Using a JDBC Template to Update a Database.....	426
Problem	426
Solution.....	426
How It Works.....	427
10-2. Using a JDBC Template to Query a Database.....	431
Problem	431
Solution.....	431
How It Works.....	432
10-3. Simplifying JDBC Template Creation.....	436
Problem	436
Solution.....	436
How It Works.....	437
10-4. Using Named Parameters in a JDBC Template	439
Problem	439
Solution.....	439
How It Works.....	439
10-5. Handling Exceptions in the Spring JDBC Framework	441
Problem	441
Solution.....	441
How It Works.....	442
10-6. Problems with Using ORM Frameworks Directly	446
Problem	446
Solution.....	446

How It Works.....	446
Persisting Objects Using the Hibernate API with Hibernate XML Mappings	448
Persisting Objects Using the Hibernate API with JPA Annotations	451
Persisting Objects Using JPA with Hibernate as the Engine.....	452
10-7. Configuring ORM Resource Factories in Spring.....	456
Problem	456
Solution.....	456
How It Works.....	456
10-8. Persisting Objects with Spring’s ORM Templates	463
Problem	463
Solution.....	463
How It Works.....	464
10-9. Persisting Objects with Hibernate’s Contextual Sessions.....	466
Problem	466
Solution.....	467
How It Works.....	467
10-10. Persisting Objects with JPA’s Context Injection.....	469
Problem	469
Solution.....	469
How It Works.....	470
10-11. Simplify JPA with Spring Data JPA	472
Problem	472
Solution.....	472
How It Works.....	473
Summary.....	474
■ Chapter 11: Spring Transaction Management	475
11-1. Problems with Transaction Management.....	476
Managing Transactions with JDBC Commit and Rollback.....	481

11-2. Choosing a Transaction Manager Implementation.....	482
Problem	482
Solution.....	482
How It Works.....	483
11-3. Managing Transactions Programmatically with the Transaction Manager API	484
Problem	484
Solution.....	484
How It Works.....	484
11-4. Managing Transactions Programmatically with a Transaction Template	486
Problem	486
Solution.....	486
How It Works.....	486
11-5. Managing Transactions Declaratively with Transaction Advices.....	489
Problem	489
Solution.....	489
How It Works.....	490
11-6. Managing Transactions Declaratively with the @Transactional Annotation.....	492
Problem	492
Solution.....	492
How It Works.....	492
11-7. Setting the Propagation Transaction Attribute	494
Problem	494
Solution.....	494
How It Works.....	495
11-8. Setting the Isolation Transaction Attribute	499
Problem	499
Solution.....	499
How It Works.....	500

11-9. Setting the Rollback Transaction Attribute..... 506

 Problem 506

 Solution..... 506

 How It Works..... 507

11-10. Setting the Timeout and Read-Only Transaction Attributes 507

 Problem 507

 Solution..... 508

 How It Works..... 508

11-11. Managing Transactions with Load-Time Weaving..... 509

 Problem 509

 Solution..... 509

 How It Works..... 509

Summary 510

■ **Chapter 12: Spring Batch 511**

 Runtime Metadata Model 512

 12-1. Setting Up Spring Batch’s Infrastructure 513

 Problem 513

 Solution..... 513

 How It Works..... 513

 12-2. Reading and Writing..... 518

 Problem 518

 Solution 518

 How It Works..... 518

 12-3. Writing a Custom ItemWriter and ItemReader 525

 Problem 525

 Solution..... 525

 How It Works..... 525

 12-4. Processing Input Before Writing 527

 Problem 527

 Solution..... 527

 How It Works..... 527

12-5. Better Living through Transactions	530
Problem	530
Solution.....	530
How It Works.....	530
12-6. Retrying	533
Problem	533
Solution.....	533
How It Works.....	533
12-7. Controlling Step Execution.....	536
Problem	536
Solution.....	536
How It Works.....	536
12-8. Launching a Job	541
Problem	541
Solution.....	541
How It Works.....	541
12-9. Parameterizing a Job	546
Problem	546
Solution.....	546
How It Works.....	546
Summary.....	548
■ Chapter 13: NoSQL and BigData	549
13-1. Using MongoDB	549
Problem	549
Solution.....	549
How It Works.....	549

13-2. Using Redis	561
Downloading and Starting Redis	561
13-3. Using Hadoop.....	568
Problem	568
Solution.....	568
How It Works.....	568
13-4. Using Neo4j.....	574
Problem	574
Solution.....	574
How It Works.....	574
Summary.....	590
■ Chapter 14: Spring Java Enterprise Services and Remoting Technologies	591
14-1. Register Spring POJOs as JMX MBeans	591
Problem	591
Solution.....	592
How It Works.....	592
14-2. Publish and Listen to JMX Notifications	606
Problem	606
Solution.....	606
How It Works.....	606
14-3. Access Remote JMX MBeans in Spring	608
Problem	608
Solution.....	608
How It Works.....	608
14-4. Send E-mail with Spring's E-mail Support	612
Problem	612
Solution.....	612
How It Works.....	613

14-5. Schedule tasks with Spring’s Quartz Support	620
Problem	620
Solution.....	620
How It Works.....	620
14-6. Schedule tasks with Spring’s Scheduling.....	625
Problem	625
Solution.....	625
How It Works.....	625
14-7. Expose and Invoke Services through RMI.....	627
Problem	627
Solution.....	628
How It Works.....	628
14-8. Expose and Invoke Services through HTTP.....	632
Problem	632
Solution.....	632
How It Works.....	632
14-9. Expose and invoke SOAP Web Services with JAX-WS	635
Problem	635
Solution.....	636
How It Works.....	636
14-10. Introduction to contract first SOAP Web Services.....	641
Problem	641
Solution.....	641
How It Works.....	641
14-11. Expose and invoke SOAP Web Services with Spring-WS.....	646
Problem	646
Solution.....	646
How It Works.....	646

14-12. Develop SOAP Web Services with Spring-WS and XML Marshalling	653
Problem	653
Solution.....	653
How It Works.....	654
Summary.....	658
■ Chapter 15: Spring Messaging	659
15-1. Send and Receive JMS Messages with Spring.....	659
Problem	659
Solution.....	660
How It Works.....	660
15-2. Convert JMS Messages	673
Problem	673
Solution.....	673
How It Works.....	673
15-3. Manage JMS Transactions	675
Problem	675
Approach	675
Solution.....	676
15-4. Create Message-Driven POJOs in Spring.....	677
Problem	677
Solution.....	678
How It Works.....	678
15-5. Cache and pool JMS connections.....	685
Problem	685
Solution.....	685
How It Works.....	685
15-6. Send and Receive AMQP Messages with Spring	686
Problem	686
Solution.....	686
How It Works.....	686
Summary.....	690

Chapter 16: Spring Integration	691
16-1. Integrating One System with Another Using EAI	692
Problem	692
Solution.....	692
How It Works.....	692
16-2. Integrating Two Systems Using JMS.....	694
Problem	694
Solution.....	694
How It Works.....	695
16-3. Interrogating Spring Integration Messages for Context Information	698
Problem	698
Solution.....	698
How It Works.....	698
16-4. Integrating Two Systems Using a File System	701
Problem	701
Solution.....	702
How It Works.....	702
16-5. Transforming a Message from One Type to Another	704
Problem	704
Solution.....	704
How It Works.....	704
16-6. Error Handling Using Spring Integration	707
Problem	707
Solution.....	707
How It Works.....	707
16-7. Forking Integration Control: Splitters and Aggregators.....	710
Problem	710
Solution	710
How It Works.....	710

16-8. Conditional Routing with Routers	713
Problem	713
Solution	714
How It Works	714
16-9. Staging Events Using Spring Batch	714
Problem	714
Solution.....	715
How It Works.....	715
16-10. Using Gateways	717
Problem	717
Solution.....	717
How It Works.....	717
Summary.....	722
■ Chapter 17: Spring Testing	723
17-1. Creating Tests with JUnit and TestNG	724
Problem	724
Solution.....	724
How It Works.....	724
17-2. Creating Unit Tests and Integration Tests.....	728
Problem	728
Solution.....	728
How It Works.....	729
17-3. Unit Testing Spring MVC Controllers	737
Problem	737
Solution.....	737
How It Works.....	737
17-4. Managing Application Contexts in Integration Tests	739
Problem	739
Solution.....	739
How It Works.....	740

17-5. Injecting Test Fixtures into Integration Tests.....	743
Problem	743
Solution.....	743
How It Works.....	744
17-6. Managing Transactions in Integration Tests.....	745
Problem	745
Solution.....	745
How It Works.....	746
17-7. Accessing a Database in Integration Tests	750
Problem	750
Solution.....	750
How It Works.....	750
17-8. Using Spring's Common Testing Annotations.....	752
Problem	752
Solution.....	752
How It Works.....	752
17-9. Integration Testing Spring MVC Controllers	753
Problem	753
Solution.....	753
How It Works.....	753
Summary.....	756
■ Chapter 18: Grails.....	757
18-1. Getting and Installing Grails.....	757
Problem	757
Solution.....	757
How It Works.....	757
18-2. Creating a Grails Application.....	758
Problem	758
Solution.....	758
How It Works.....	759

18-3. Grails Plug-Ins.....	763
Problem	763
Solution.....	763
How It Works.....	764
18-4. Developing, Producing, and Testing in Grails Environments	765
Problem	765
Solution.....	765
How It Works.....	765
18-5. Creating an Application's Domain Classes.....	767
Problem	767
Solution.....	767
How It Works.....	768
18-6. Generating CRUD Controllers and Views for an Application's Domain Classes.....	769
Problem	769
Solution.....	770
How It Works.....	770
18-7. Internationalization (I18n) Message Properties	773
Problem	773
Solution.....	773
How It Works.....	773
18-8. Changing Permanent Storage Systems	776
Problem	776
Solution.....	776
How It Works.....	776
18-9. Logging	779
Problem	779
Solution.....	779
How It Works.....	779

18-10. Running Unit and Integration Tests	781
Problem	781
Solution.....	781
How It Works.....	781
18-11. Using Custom Layouts and Templates	787
Problem	787
Solution.....	787
How It Works.....	787
18-12. Using GORM Queries	790
Problem	790
Solution.....	790
How It Works.....	790
18-13. Creating Custom Tags	792
Problem	792
Solution.....	792
How It Works.....	792
18-14. Adding Security.....	794
Problem	794
Solution.....	794
How It Works.....	794
Summary.....	798
Index.....	799

About the Authors



Marten Deinum is a Java/software consultant working for Conspact. He has developed and architected software, primarily in Java, for small and large companies. He is an enthusiastic open source user and longtime fan, user and advocate of the Spring Framework. He has held a number of positions including Software Engineer, Development Lead, Coach, and also as a Java and Spring Trainer. When not working or answering questions on StackOverflow, he can be found in the water training for open water swimming events or under the water diving or guiding other people around.



Josh Long is the Spring developer advocate for SpringSource, an editor for <http://www.InfoQ.com>, and author/co-author of many works (including *Spring Recipes: A Problem-Solution Approach, Second Edition*, published by Apress). Josh has spoken at numerous industry conferences, including Geecon, TheServerSide Java Symposium, SpringOne, OSCON, JavaZone, Devvxx, JAX, and Java2Days. When he is not hacking on Spring Integration and other open-source code (see <http://Git.SpringSource.org> and <http://GitHub.com/JoshLong>), he can be found at the local Java user group, a coffee shop, or the airport. Josh likes solutions that push the boundaries of the technologies that enable them. His interests include scalability, big data, business process management, grid processing, rich Internet applications, mobile computing, and so-called “smart systems”. He blogs at <http://blog.springsource.org> and <http://www.JoshLong.com>, and can be reached at josh@joshlong.com.



Gary Mak, founder and chief consultant of Meta-Archit Software Technology Limited, has been a technical architect and application developer on the enterprise Java platform for more than seven years. He is the author of the Apress books *Spring Recipes: A Problem-Solution Approach* and *Pro SpringSource dm Server*. In his career, Gary has developed a number of Java-based software projects, most of which are application frameworks, system infrastructures, and software tools. He enjoys designing and implementing the complex parts of software projects. Gary has a master's degree in computer science. His research interests include object-oriented technology, aspect-oriented technology, design patterns, software reuse, and domain-driven development. Gary specializes in building enterprise applications on technologies including Spring, Hibernate, JPA, JSF, Portlet, AJAX, and OSGi. He has been using the Spring Framework in his projects since Spring version 1.0. Gary has been an instructor of courses on enterprise Java, Spring, Hibernate, Web Services, and agile development. He has written a series of Spring and Hibernate tutorials as course materials, parts of which are open to the public, and they're gaining popularity in the Java community. In his spare time, he enjoys playing tennis and watching tennis competitions.



Daniel Rubio is an independent consultant with over 10 years of experience in enterprise and web-based software. More recently, Daniel is founder and technical lead at MashupSoft.com.

About the Technical Reviewer

Felipe Gutierrez is an active, expert Spring and enterprise Java applications developer. He has used Spring Framework for several years now.

Acknowledgments

The acknowledgements are probably the hardest thing to write. It is impossible to name everyone personally and I probably will forget someone. For that I want to apologize beforehand.

Although this is the second book I wrote I couldn't have done it without the team at Apress; without you this book would have never seen the light of day. A special thanks to Mark Powers and Anamika Panchoo for keeping me focused and on schedule.

I thank **Felipe Gutierrez**, without whose comments and suggestions this book would never have become what it is now.

Thanks to my family and friends for the times they missed me, and to my dive-buddies for all the dives and trips I missed over the last year.

Last but definitely not least I thank my wife, Djoke Deinum, and daughter, Geeske, for their endless support, love, and dedication, despite the long evenings, sacrificed weekends and holidays to finish the book. Without your support I probably would have abandoned the endeavor long ago.

—Marten Deinum

Introduction

The Spring framework is growing. It has always been about choice. Java EE focused on a few technologies, largely to the detriment of alternative, better solutions. When the Spring framework debuted, few would have agreed that Java EE represented the best-in-breed architectures of the day. Spring debuted to great fanfare, because it sought to simplify Java EE. Each release since marks the introduction of new features designed to both simplify and enable solutions.

With version 2.0 and later, the Spring framework started targeting multiple platforms. The framework provided services on top of existing platforms, as always, but was decoupled from the underlying platform wherever possible. Java EE is still a major reference point, but it's not the only target. Additionally, the Spring framework runs on different Cloud environments. With the introduction of Java based configuration and more XML schemas, the Spring framework created powerful configuration options. Frameworks built on top of the Spring framework have emerged supporting application integration, batch processing, messaging, and much more.

This is the 3rd edition of the superb Spring Recipes and it contains mostly updated frameworks, describing the new features and explaining the different configuration (Java and/or XML) options. Additionally, new projects have been added to the Spring ecosystem like the Spring Data family of products.

It was impossible to describe each and every project in the Spring ecosystem, so we had to decide what to keep, what to add, and what to update. This was a hard decision but we think we have included the most important and used content.

Who This Book Is For

This book is for Java developers who want to simplify their architecture and solve problems outside the scope of the Java EE platform. If you are already using Spring in your projects, the more advanced chapters present discussions of newer technologies that you might not know about already. If you are new to the framework, this book will get you started in no time.

This book assumes that you have some familiarity with Java and an IDE of some sort. While it is possible, and indeed useful, to use Java exclusively with client applications, Java's largest community lives in the enterprise space and that, too, is where you'll see most of these technologies deliver the most benefit. Thus, some familiarity with basic enterprise programming concepts like the Servlet API is assumed.

How This Book Is Structured

Chapter 1, "Spring Development Tools," gives an overview of tools supporting the Spring framework and how to use them.

Chapter 2, "Spring core tasks," gives a general overview of the Spring framework: how to set it up, what it is, and how it's used.

Chapter 3, "Spring Annotation Driven Core Task," reviews, in addition to Chapter 2 more annotation driven concepts that are still key to fully exploiting the container.

Chapter 4, "Spring @MVC," covers web-based application development using the Spring Web MVC framework.

Chapter 5, "Spring REST," provides an introduction to Spring's support for RESTful web services.

Chapter 6, "Spring Social," provides an introduction of Spring Social, which lets you integrate easily with social networks.

Chapter 7, "Spring Security," provides an overview of the Spring Security project, to help you better secure your application.

Chapter 8, "Spring Mobile," provides an introduction of Spring Mobile, which lets you integrate Mobile device detection and usage in your application.

Chapter 9, “Integrating Spring with Other Web Frameworks,” introduces the core web-tier support that Spring provides. This provides a base for all technologies that Spring provides in the web tier.

Chapter 10, “Data Access,” discusses using Spring to talk to data stores using APIs like JDBC, Hibernate, and JPA.

Chapter 11, “Transaction Management in Spring,” introduces the concepts behind Spring’s robust transaction management facilities.

Chapter 12, “Spring Batch,” introduces the Spring Batch framework, which provides a way to model solutions traditionally considered the domain of mainframes.

Chapter 13, “NoSQL and BigData,” an introduction to multiple Spring Data portfolio projects, covering different NoSQL technologies and BigData with Hadoop.

Chapter 14, “Spring Java Enterprise Services and Remoting Technologies,” introduces you to the JMX support, scheduling, e-mail support, and various facilities for RPC, including the Spring Web Services project.

Chapter 15, “Spring Messaging,” discusses using Spring with message-oriented middleware through JMS and RabbitMQ and the simplifying Spring abstractions.

Chapter 16, “Spring Integration,” discusses using the Spring Integration framework to integrate disparate services and data.

Chapter 17, “Spring Testing,” discusses unit testing with the Spring framework.

Chapter 18, “Grails,” discusses the Grails framework, with which you can increase your productivity by using best-of-breed pieces and gluing them together with Groovy code.

Appendix A, “Deployment to the Cloud,” shows how to deploy a Java (Web) application to the cloud using the Pivotal CloudFoundry solution.

Appendix B, “Spring Caching,” introduces the Spring Caching abstraction, including how to configure it and how to transparently add caching to your application.

Conventions

Sometimes, when we want you to pay particular attention to a part within a code example, we will make the font bold. Please note that the bold doesn’t necessarily reflect a code change from the previous version.

In cases when a code line is too long to fit the page’s width, we will break it with a code continuation character. Please note that when you try to type the code, you have to concatenate the line by yourself without any spaces.

Prerequisites

Because the Java programming language is platform independent, you are free to choose any supported operating system. However, some of the examples in this book use platform-specific paths. Translate them as necessary to your operating system’s format before typing the examples.

To make the most of this book, install JDK version 1.7 or higher. You should have a Java IDE installed to make development easier. For this book, the sample code is Gradle based. If you’re running Eclipse and Install the Gradle plug-in, you can open the same code in Eclipse and the CLASSPATH and dependencies will be filled in the by the Gradle metadata.

If you’re using Eclipse, you might prefer SpringSource’s SpringSource Tool Suite (STS), as it comes preloaded with the plug-ins you’ll need to be productive with the Spring framework in Eclipse. If you use IntelliJ IDEA, you need to enable the Gradle (and Groovy) plugins.

Downloading the code

The source code for this book is available from the Apress web site (www.apress.com) in the Source Code / Download section. The source code is organized by chapters, each of which includes one or more independent examples.

Contacting the Authors

We always welcome your questions and feedback regarding the contents of this book. You can contact Marten Deinum at marten@deinum.biz