

# Contents

<b>Acknowledgments</b>	<b>ix</b>
<b>Introduction</b>	<b>xxiii</b>
<b>Chapter 1: Introducing the Spring Framework</b>	<b>1</b>
<b>Why Spring?</b>	<b>1</b>
Problems with the Traditional Approach to J2EE	1
Lightweight Frameworks	3
Enter Spring	5
<b>Spring's Values</b>	<b>6</b>
<b>Spring in Context</b>	<b>8</b>
Technologies	8
Techniques	18
Relationship to Other Frameworks	19
<b>Architecting Applications with Spring</b>	<b>21</b>
The Big Picture	21
Persistence and Integration	23
Business Service Objects	27
Presentation	27
<b>The Future</b>	<b>29</b>
Release Schedule	29
The Evolution of Java and J2EE	29
Technology Currents	31
Standards and Open Source	31
<b>The Spring Project and Community</b>	<b>32</b>
History	32
Module Summary	33
Supported Environments	37
<b>Summary</b>	<b>37</b>
<b>Chapter 2: The Bean Factory and Application Context</b>	<b>39</b>
<b>Inversion of Control and Dependency Injection</b>	<b>40</b>
Different Forms of Dependency Injection	43
Deciding Between Setter Injection and Constructor Injection	45
<b>The Container</b>	<b>47</b>
The Bean Factory	47
The Application Context	48
Starting the Container	50

# Contents

---

Using Beans from the Factory	52
XML Bean Configuration	53
The Basic Bean Definition	53
Specifying Bean Dependencies	56
Managing the Bean Lifecycle	64
Abstracting Access to Services and Resources	67
Reusing Bean Definitions	72
Using Post-Processors to Handle Customized Beans and Containers	75
<b>Summary</b>	<b>81</b>
<b>Chapter 3: Advanced Container Concepts</b>	<b>83</b>
<b>Abstractions for Low-Level Resources</b>	<b>84</b>
The Application Context as a ResourceLoader	84
The Application Context as a Message Source	87
<b>Application Events</b>	<b>89</b>
<b>Managing the Container</b>	<b>91</b>
Resource Location Paths in ApplicationContext Constructors	92
Declarative Usage of Application Contexts	93
Splitting Up Container Definitions into Multiple Files	95
Strategies for Handling Components	97
Singletons for Accessing the Container	99
<b>Some Convenience Factory Beans</b>	<b>99</b>
PropertyPathFactoryBean	99
FieldRetrievingFactoryBean	101
MethodInvokingFactoryBean	101
<b>Property Editors Provided by Spring</b>	<b>102</b>
<b>Strategies for Testing</b>	<b>103</b>
Unit Tests	103
Tests That Use the Spring Container	106
<b>Alternatives to XML</b>	<b>108</b>
Definitions from Properties Files	108
Programmatic Bean Definitions	110
Other Formats	110
<b>References</b>	<b>110</b>
<b>Summary</b>	<b>111</b>
<b>Chapter 4: Spring and AOP</b>	<b>113</b>
<b>Goals</b>	<b>113</b>
<b>Assumptions</b>	<b>114</b>
<b>Example</b>	<b>114</b>

---

<b>Spring's AOP Framework</b>	<b>118</b>
The Interceptor Chain	118
Pros and Cons	118
Advice	119
Pointcuts	125
Advisors	130
Integration with the Spring IoC Container	132
Examining and Manipulating Proxy State at Runtime	145
Programmatic Proxy Creation	146
<b>Using Advanced Features of Spring's AOP Framework</b>	<b>147</b>
TargetSources	147
Terminating the Interceptor Chain Early	153
Using Introduction	153
Exposing the Current Proxy	156
Exposing the Current MethodInvocation	157
Understanding Proxy Types	158
Debugging and Testing	159
Miscellaneous	161
<b>Integration with Other AOP Frameworks</b>	<b>163</b>
Goals	164
AspectJ Integration	164
AspectWerkz	169
<b>References</b>	<b>169</b>
<b>Summary</b>	<b>170</b>
<b>Chapter 5: DAO Support and JDBC Framework</b>	<b>171</b>
<b>The Data Access Object Pattern</b>	<b>172</b>
<b>Introducing Spring's JDBC Framework</b>	<b>173</b>
Motivation: Issues with Direct Use of JDBC	173
How Spring Can Help	175
A Simple Example	176
<b>Building the Data Access Layer for the Sample Application</b>	<b>177</b>
Data Model for the Sample Application	177
DataSource	178
Exception Translation	180
<b>Operations with JdbcTemplate</b>	<b>183</b>
Use of Callback Methods	183
JdbcTemplate Convenience Methods	184
Basic Queries Using the JdbcTemplate	185
Basic Updating with JdbcTemplate	186
Advanced Use of JdbcTemplate	187
Support for RowSet	188

# Contents

---

<b>Using the RDBMS Operation Classes</b>	<b>189</b>
SqlQuery and MappingSqlQuery	190
Inserts and Updates with SqlUpdate	192
Updating a ResultSet Using UpdatableSqlQuery	193
Generating Primary Keys	194
Retrieving Database-Generated Keys	195
Calling a Stored Procedure	196
<b>Advanced Concepts</b>	<b>199</b>
Running Spring JDBC in an Application Server	199
Using Custom Exception Translations	201
Reading and Writing LOB Data	203
Batch Updates	208
Advanced Use of Stored Procedures	210
<b>Additional Considerations</b>	<b>214</b>
Performance	214
When to Use JDBC Versus O/R Mapping	215
JDBC Versions and J2EE Versions	215
<b>Summary</b>	<b>215</b>
<b>Chapter 6: Transaction and Resource Management</b>	<b>217</b>
<b>Background</b>	<b>217</b>
What Is a Transaction?	217
The ACID Properties	218
Concurrency Control	220
<b>Transactions and J2EE</b>	<b>220</b>
Local Transactions	221
Global/Distributed Transactions	221
Transaction Propagation	221
Transaction Demarcation	221
<b>An Example of Spring's Transaction Support</b>	<b>222</b>
<b>Introducing Spring's Transaction Abstraction</b>	<b>224</b>
Overview of Choices for Transaction Control	225
Transaction Definition	226
Transaction Status	229
Transaction Demarcation Strategies	229
Transaction Management Strategies	242
<b>DataSource Declarations</b>	<b>251</b>
Local Unpooled	251
Local Pooled	252
JNDI	253
Choosing Between Local and JNDI DataSource	253
<b>Summary</b>	<b>254</b>

---

<b>Chapter 7: Object/Relational Mapping</b>	<b>255</b>
<b>Background</b>	<b>256</b>
Basic O/R Mapping	256
Object Query Languages	257
Transparent Persistence	258
When to Choose O/R Mapping	258
<b>O/R Mapping Support in Spring</b>	<b>259</b>
Data Access Objects	259
Transaction Management	260
<b>iBATIS SQL Maps</b>	<b>261</b>
Mapping File	261
DAO Implementation	263
Setup in a Spring Context	265
Transaction Management	266
iBATIS Summary	267
<b>Hibernate</b>	<b>268</b>
Mapping File	269
DAO Implementation	270
Setup in a Spring Context	273
Transaction Management	276
Open Session in View	282
BLOB/CLOB Handling	285
Hibernate: Summary	287
<b>JDO</b>	<b>288</b>
Persistent Object Lifecycle	289
DAO Implementation	289
Setup in a Spring Context	291
Transaction Management	293
PersistenceManager Lifecycle	294
Open PersistenceManager in View	295
JDO Dialect	297
JDO: Summary	298
<b>Other O/R Mapping Tools</b>	<b>299</b>
Apache OJB	299
TopLink	300
Cayenne	302
JSR-220 Persistence	302
<b>Summary</b>	<b>302</b>
<b>Chapter 8: Lightweight Remoting</b>	<b>305</b>
<b>Concepts and Scope</b>	<b>306</b>

---

# Contents

---

<b>Common Configuration Style</b>	<b>307</b>
<b>Hessian and Burlap</b>	<b>309</b>
Accessing a Service	310
Exporting a Service	312
<b>HTTP Invoker</b>	<b>313</b>
Accessing a Service	314
Exporting a Service	315
Customization Options	316
<b>RMI</b>	<b>317</b>
Accessing a Service	318
Stub Lookup Strategies	320
Exporting a Service	322
Customization Options	323
RMI-IIOP	323
<b>Web Services via JAX-RPC</b>	<b>324</b>
Accessing a Service	326
Exporting a Service	328
Custom Type Mappings	330
<b>Summary</b>	<b>332</b>
<b>Chapter 9: Supporting Services</b>	<b>335</b>
<b>JMS</b>	<b>335</b>
Introduction	336
Motivation for Spring's JMS Support	337
JMS Access via a Template	338
Exception Handling	340
ConnectionFactory Management	340
Message Converters	340
Destination Management	342
Transaction Management	343
JmsGatewaySupport	343
The Future	345
<b>Scheduling with Spring</b>	<b>345</b>
Timers Versus Quartz	346
Timers	346
Quartz	348
<b>Sending Email with Spring</b>	<b>354</b>
Getting Started	354
Reusing an Existing Mail Session	355
Mail Sending Using COS	356
Generic Mail Manager	356

---

<b>Scripting</b>	<b>361</b>
Consistent Model	361
Other Scripting Languages	365
<b>Summary</b>	<b>365</b>
<b>Chapter 10: Acegi Security System for Spring</b>	<b>367</b>
<hr/>	
<b>Enterprise Application Security Choices</b>	<b>367</b>
Typical Requirements	367
Acegi Security in a Nutshell	368
Java Authentication and Authorization Service	372
Servlet Specification	375
<b>Acegi Security Fundamentals</b>	<b>377</b>
Authentication	377
Storing the Authentication Object	381
Authorization	383
Domain Object Instance Security	386
<b>Example Code</b>	<b>389</b>
Introducing the Sample	389
Security Unaware Implementation	389
Security Approach	391
Authentication	391
Authorization	392
<b>Summary</b>	<b>395</b>
<b>Chapter 11: Spring and EJB</b>	<b>397</b>
<hr/>	
<b>Deciding on the Need for EJBs</b>	<b>397</b>
<b>Accessing EJBs</b>	<b>399</b>
The Boilerplate Approach	399
The Spring Approach	400
<b>Implementing EJBs with Spring</b>	<b>407</b>
Stateless Session Beans	407
Stateful Session Beans	410
Message Driven Beans	411
A Digression into XDoclet	412
<b>Singleton Container Access, Good or Evil?</b>	<b>413</b>
ContextSingletonBeanFactoryLocator and SingletonBeanFactoryLocator	414
A Shared Context as the Parent of a Web-App Application Context	416
Using a Shared Context from EJBs	419

# Contents

---

<b>Testing Concerns</b>	<b>420</b>
Implement Business Functionality in POJO Delegate	420
Use a Mock EJB Container	420
Integration Testing Inside the AppServer	421
<b>Summary</b>	<b>422</b>
<b>Chapter 12: Web MVC Framework</b>	<b>423</b>
<b>A Simple Example</b>	<b>423</b>
<b>General Architecture</b>	<b>425</b>
Web MVC Concepts	426
Generic Web MVC Using a Dispatcher and Controllers	427
Requirements for a Good Web MVC Framework	428
Spring's Web MVC Equivalents	429
<b>Infrastructural Components</b>	<b>430</b>
The DispatcherServlet	431
The WebApplicationContext	434
<b>Workflow Involved with Handling Requests</b>	<b>436</b>
<b>Common Layout of a Spring MVC Web Application</b>	<b>440</b>
<b>HandlerMappings</b>	<b>441</b>
BeanNameUrlHandlerMapping	442
SimpleUrlHandlerMapping	443
CommonsPathMapUrlHandlerMapping	444
More Than One HandlerMapping	445
<b>HandlerExecutionChain and Interceptors</b>	<b>446</b>
WebContentInterceptor	447
UserRoleAuthorizationInterceptor	448
Other Handler Interceptors Provided by Spring MVC	449
<b>Handlers and Their Adapters</b>	<b>449</b>
<b>ModelAndView and ViewResolvers</b>	<b>450</b>
UrlBasedViewResolvers	450
BeanNameViewResolver and XmlViewResolver	451
ResourceBundleViewResolver	451
Chaining View Resolvers	452
<b>Locale Changing and Resolution</b>	<b>454</b>
<b>HandlerExceptionResolvers</b>	<b>456</b>
<b>Controllers</b>	<b>458</b>
WebContentGenerator	458
AbstractController	459
UrlFilenameViewController	459
ParameterizableViewController	460
MultiActionController	460

---

<b>Data Binding</b>	<b>461</b>
Convenient Functionality When Working with Data Binding	463
<b>Real-World Examples of Using Controllers</b>	<b>464</b>
Viewing a List of Performances Using the AbstractController	464
Making a Reservation	466
<b>Wizard Functionality</b>	<b>474</b>
Basic Configuration	474
Template Methods	475
Flow of a Wizard	476
Page Changes, Numbering, and Other Actions	476
<b>Extending Spring's Handler Infrastructure</b>	<b>478</b>
<b>Uploading Files</b>	<b>478</b>
Configuring the Multipart Resolver	479
Creating a Form to Upload a File	479
Handling Uploaded Data	479
<b>Testing Your Controllers</b>	<b>481</b>
Testing Without the Application Context	481
More Extensive Testing Using Mock Objects	483
Testing Using the Complete Application Context	484
Other Ways to Test a Web Application	485
<b>Summary</b>	<b>485</b>
<b>Chapter 13: Web View Technologies</b>	<b>487</b>
<b>An Example</b>	<b>488</b>
General Configuration	488
JavaServer Pages	489
FreeMarker	489
Generating PDFs Using iText	490
<b>Considerations When Choosing a Technology</b>	<b>491</b>
<b>View Objects and Models</b>	<b>491</b>
<b>Features Offered by the AbstractView</b>	<b>493</b>
Issuing a New Request Using the RedirectView	494
Using View Prefixes to Issue Forwards or Redirects	495
<b>JavaServer Pages</b>	<b>496</b>
Configuring Your Application to Use JSP	497
Creating Forms Using Custom Tags	497
Using Tag Files to Create Reusable Elements	504
<b>Velocity and FreeMarker</b>	<b>505</b>
Configuring the View Resolver	505
Using the Form Simplification Macros	507

# Contents

---

<b>Tiles</b>	<b>511</b>
<b>Views Based on XML and XSLT</b>	<b>514</b>
<b>Excel and Other Document-Based Views</b>	<b>515</b>
Generating an Excel File from Our List of Shows	516
Using a Template as a Basis for Your Excel File	517
Other Document-Based Views	517
Using a HandlerInterceptor to Distinguish Between HTML and Excel	517
<b>Implementing Custom Views</b>	<b>519</b>
View and AbstractView	520
Implementing a View Generating XML from a Data Object	520
Considerations When Implementing Custom Views	522
<b>Summary</b>	<b>522</b>
<b>Chapter 14: Integrating with Other Web Frameworks</b>	<b>523</b>
<b>Considerations When Choosing an MVC Framework</b>	<b>523</b>
Comparing the Traditional Web MVC Frameworks	524
<b>Integrating with Spring: Core Concepts</b>	<b>534</b>
<b>WebWork Integration</b>	<b>536</b>
Setting Up the ObjectFactory	536
<b>Struts Integration</b>	<b>537</b>
Using ActionSupport	538
Using DelegationRequestProcessor and DelegationActionProxy	539
Using an Auto Wiring Base Action	542
<b>Tapestry Integration</b>	<b>543</b>
Getting Beans for Tapestry	543
The Page Class	544
The Page Definition	544
The Page Template	545
Tapestry Integration Final Thoughts	547
<b>JavaServer Faces Integration</b>	<b>547</b>
<b>Summary</b>	<b>549</b>
<b>Chapter 15: The Sample Application</b>	<b>551</b>
<b>Choice of Server Technologies</b>	<b>552</b>
<b>Application Layers</b>	<b>552</b>
<b>Persistence Layer</b>	<b>554</b>
Data Model	554
Domain Object Model	556
Object/Relational Mapping	558
DAO Implementation	561

---

Data Access Context	563
<b>Business Service Layer</b>	<b>563</b>
Services	564
Application Context	565
<b>Web Layer</b>	<b>566</b>
Flow Through the Application	566
Configuring the Application via web.xml	568
Web Controllers	569
View Technology	571
<b>Comparison with the J2EE Design and Development Implementation</b>	<b>572</b>
Simpler Technology	572
Database Changes	573
<b>Server Configuration</b>	<b>573</b>
MySQL	573
Tomcat	574
<b>Build and Deployment</b>	<b>574</b>
Creating and Loading the Database Tables	574
Building the Application and Deploying It to the Tomcat Server	575
<b>Summary</b>	<b>575</b>
<b>Chapter 16: Conclusion</b>	<b>577</b>
<hr/>	
<b>Problems Solved by Spring</b>	<b>577</b>
<b>The Spring Solution</b>	<b>578</b>
<b>Guidelines for Spring Development</b>	<b>581</b>
Technology Choices	581
Layer by Layer	583
Structuring Applications	589
Testing Your Applications	592
<b>Related Projects</b>	<b>594</b>
Acegi Security for Spring	594
Other Projects	595
<b>Spring in Non-J2EE Environments</b>	<b>596</b>
<b>To Find Out More</b>	<b>596</b>
Books and Articles	597
Online Resources	597
Sample Applications	597
<b>The Future</b>	<b>598</b>
<b>Appendix A: Requirements for the Sample Application</b>	<b>601</b>
<hr/>	
<b>Overview</b>	<b>601</b>

# Contents

---

<b>User Populations</b>	<b>602</b>
Public Internet Users	602
Box Office Users	602
Administrators	603
<b>Assumptions</b>	<b>603</b>
<b>Scope Limitations</b>	<b>604</b>
<b>Delivery Schedule</b>	<b>604</b>
<b>Internet User Interface</b>	<b>605</b>
Basic Workflow	605
Error Handling	606
Application Screens	606
<b>Non-Functional Requirements</b>	<b>618</b>
<b>Hardware and Software Environment</b>	<b>619</b>
<b>Index</b>	<b>621</b>