

Professional Java™ Development with the Spring Framework

Rod Johnson
Juergen Hoeller
Alef Arendsen
Thomas Risberg
Colin Sampaleanu



Wiley Publishing, Inc.

Contents

Acknowledgments	ix
Introduction	xxiii
Chapter 1: Introducing the Spring Framework	1
Why Spring?	1
Problems with the Traditional Approach to J2EE	1
Lightweight Frameworks	3
Enter Spring	5
Spring's Values	6
Spring in Context	8
Technologies	8
Techniques	18
Relationship to Other Frameworks	19
Architecting Applications with Spring	21
The Big Picture	21
Persistence and Integration	23
Business Service Objects	27
Presentation	27
The Future	29
Release Schedule	29
The Evolution of Java and J2EE	29
Technology Currents	31
Standards and Open Source	31
The Spring Project and Community	32
History	32
Module Summary	33
Supported Environments	37
Summary	37
Chapter 2: The Bean Factory and Application Context	39
Inversion of Control and Dependency Injection	40
Different Forms of Dependency Injection	43
Deciding Between Setter Injection and Constructor Injection	45
The Container	47
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
Summary	81
Chapter 3: Advanced Container Concepts	83
Abstractions for Low-Level Resources	84
The Application Context as a ResourceLoader	84
The Application Context as a Message Source	87
Application Events	89
Managing the Container	91
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
Some Convenience Factory Beans	99
PropertyPathFactoryBean	99
FieldRetrievingFactoryBean	101
MethodInvokingFactoryBean	101
Property Editors Provided by Spring	102
Strategies for Testing	103
Unit Tests	103
Tests That Use the Spring Container	106
Alternatives to XML	108
Definitions from Properties Files	108
Programmatic Bean Definitions	110
Other Formats	110
References	110
Summary	111
Chapter 4: Spring and AOP	113
Goals	113
Assumptions	114
Example	114

Spring's AOP Framework	118
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
Using Advanced Features of Spring's AOP Framework	147
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
Integration with Other AOP Frameworks	163
Goals	164
AspectJ Integration	164
AspectWerkz	169
References	169
Summary	170
Chapter 5: DAO Support and JDBC Framework	171
The Data Access Object Pattern	172
Introducing Spring's JDBC Framework	173
Motivation: Issues with Direct Use of JDBC	173
How Spring Can Help	175
A Simple Example	176
Building the Data Access Layer for the Sample Application	177
Data Model for the Sample Application	177
DataSource	178
Exception Translation	180
Operations with JdbcTemplate	183
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

Using the RDBMS Operation Classes	189
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
Advanced Concepts	199
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
Additional Considerations	214
Performance	214
When to Use JDBC Versus O/R Mapping	215
JDBC Versions and J2EE Versions	215
Summary	215
Chapter 6: Transaction and Resource Management	217
Background	217
What Is a Transaction?	217
The ACID Properties	218
Concurrency Control	220
Transactions and J2EE	220
Local Transactions	221
Global/Distributed Transactions	221
Transaction Propagation	221
Transaction Demarcation	221
An Example of Spring's Transaction Support	222
Introducing Spring's Transaction Abstraction	224
Overview of Choices for Transaction Control	225
Transaction Definition	226
Transaction Status	229
Transaction Demarcation Strategies	229
Transaction Management Strategies	242
DataSource Declarations	251
Local Unpooled	251
Local Pooled	252
JNDI	253
Choosing Between Local and JNDI DataSource	253
Summary	254

Chapter 7: Object/Relational Mapping	255
Background	256
Basic O/R Mapping	256
Object Query Languages	257
Transparent Persistence	258
When to Choose O/R Mapping	258
O/R Mapping Support in Spring	259
Data Access Objects	259
Transaction Management	260
iBatis SQL Maps	261
Mapping File	261
DAO Implementation	263
Setup in a Spring Context	265
Transaction Management	266
iBatis Summary	267
Hibernate	268
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
JDO	288
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
Other O/R Mapping Tools	299
Apache OJB	299
TopLink	300
Cayenne	302
JSR-220 Persistence	302
Summary	302
Chapter 8: Lightweight Remoting	305
Concepts and Scope	306

Contents

Common Configuration Style	307
Hessian and Burlap	309
Accessing a Service	310
Exporting a Service	312
HTTP Invoker	313
Accessing a Service	314
Exporting a Service	315
Customization Options	316
RMI	317
Accessing a Service	318
Stub Lookup Strategies	320
Exporting a Service	322
Customization Options	323
RMI-IIOP	323
Web Services via JAX-RPC	324
Accessing a Service	326
Exporting a Service	328
Custom Type Mappings	330
Summary	332
Chapter 9: Supporting Services	335
JMS	335
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
Scheduling with Spring	345
Timers Versus Quartz	346
Timers	346
Quartz	348
Sending Email with Spring	354
Getting Started	354
Reusing an Existing Mail Session	355
Mail Sending Using COS	356
Generic Mail Manager	356

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

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

Data Binding	461
Convenient Functionality When Working with Data Binding	463
Real-World Examples of Using Controllers	464
Viewing a List of Performances Using the AbstractController	464
Making a Reservation	466
Wizard Functionality	474
Basic Configuration	474
Template Methods	475
Flow of a Wizard	476
Page Changes, Numbering, and Other Actions	476
Extending Spring's Handler Infrastructure	478
Uploading Files	478
Configuring the Multipart Resolver	479
Creating a Form to Upload a File	479
Handling Uploaded Data	479
Testing Your Controllers	481
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
Summary	485
Chapter 13: Web View Technologies	487
An Example	488
General Configuration	488
JavaServer Pages	489
FreeMarker	489
Generating PDFs Using iText	490
Considerations When Choosing a Technology	491
View Objects and Models	491
Features Offered by the AbstractView	493
Issuing a New Request Using the RedirectView	494
Using View Prefixes to Issue Forwards or Redirects	495
JavaServer Pages	496
Configuring Your Application to Use JSP	497
Creating Forms Using Custom Tags	497
Using Tag Files to Create Reusable Elements	504
Velocity and FreeMarker	505
Configuring the View Resolver	505
Using the Form Simplification Macros	507

Contents

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

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

Contents

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