Professional Java™ Development
with the Spring Framework

Rod Johnson
Juergen Hoeller
Alef Arendsen
Thomas Risberg
Colin Sampaleanu
Acknowledgments
Introduction

Chapter 1: Introducing the Spring Framework

Why Spring?
Problems with the Traditional Approach to J2EE
Lightweight Frameworks
Enter Spring

Spring’s Values

Spring in Context
Technologies
Techniques
Relationship to Other Frameworks

Architecting Applications with Spring
The Big Picture
Persistence and Integration
Business Service Objects
Presentation

The Future
Release Schedule
The Evolution of Java and J2EE
Technology Currents
Standards and Open Source

The Spring Project and Community
History
Module Summary
Supported Environments

Summary

Chapter 2: The Bean Factory and Application Context

Inversion of Control and Dependency Injection
Different Forms of Dependency Injection
Deciding Between Setter Injection and Constructor Injection

The Container
The Bean Factory
The Application Context
Starting the Container
Contents

Using Beans from the Factory
XML Bean Configuration
The Basic Bean Definition
Specifying Bean Dependencies
Managing the Bean Lifecycle
Abstracting Access to Services and Resources
Reusing Bean Definitions
Using Post-Processors to Handle Customized Beans and Containers

Summary

Chapter 3: Advanced Container Concepts

Abstractions for Low-Level Resources
The Application Context as a ResourceLoader
The Application Context as a Message Source

Application Events

Managing the Container
Resource Location Paths in ApplicationContext Constructors
Declarative Usage of Application Contexts
Splitting Up Container Definitions into Multiple Files
Strategies for Handling Components
Singletons for Accessing the Container

Some Convenience Factory Beans
PropertyPathFactoryBean
FieldRetrievingFactoryBean
MethodInvokingFactoryBean

Property Editors Provided by Spring

Strategies for Testing
Unit Tests
Tests That Use the Spring Container

Alternatives to XML
Definitions from Properties Files
Programmatic Bean Definitions
Other Formats

References
Summary

Chapter 4: Spring and AOP

Goals
Assumptions
Example
Spring's AOP Framework
  The Interceptor Chain
  Pros and Cons
  Advice
  Pointcuts
  Advisors
  Integration with the Spring IoC Container
  Examining and Manipulating Proxy State at Runtime
  Programmatic Proxy Creation
Using Advanced Features of Spring's AOP Framework
  TargetSources
  Terminating the Interceptor Chain Early
  Using Introduction
  Exposing the Current Proxy
  Exposing the Current MethodInvocation
  Understanding Proxy Types
  Debugging and Testing
  Miscellaneous
Integration with Other AOP Frameworks
  Goals
  AspectJ Integration
  AspectWerkz
References
Summary

Chapter 5: DAO Support and JDBC Framework
The Data Access Object Pattern
Introducing Spring's JDBC Framework
  Motivation: Issues with Direct Use of JDBC
  How Spring Can Help
  A Simple Example

Building the Data Access Layer for the Sample Application
  Data Model for the Sample Application
  DataSource
  Exception Translation
Operations with JdbcTemplate
  Use of Callback Methods
  JdbcTemplate Convenience Methods
  Basic Queries Using the JdbcTemplate
  Basic Updating with JdbcTemplate
  Advanced Use of JdbcTemplate
  Support for RowSet
Contents

Using the RDBMS Operation Classes
  SqlQuery and MappingSqlQuery
  Inserts and Updates with SqlUpdate
  Updating a ResultSet Using UpdatableSqlQuery
  Generating Primary Keys
  Retrieving Database-Generated Keys
  Calling a Stored Procedure

Advanced Concepts
  Running Spring JDBC in an Application Server
  Using Custom Exception Translations
  Reading and Writing LOB Data
  Batch Updates
  Advanced Use of Stored Procedures

Additional Considerations
  Performance
  When to Use JDBC Versus O/R Mapping
  JDBC Versions and J2EE Versions

Summary

Chapter 6: Transaction and Resource Management

Background
  What Is a Transaction?
  The ACID Properties
  Concurrency Control

Transactions and J2EE
  Local Transactions
  Global/Distributed Transactions
  Transaction Propagation
  Transaction Demarcation

An Example of Spring's Transaction Support

Introducing Spring's Transaction Abstraction
  Overview of Choices for Transaction Control
  Transaction Definition
  Transaction Status
  Transaction Demarcation Strategies
  Transaction Management Strategies

DataSource Declarations
  Local Unpooled
  Local Pooled
  JNDI
  Choosing Between Local and JNDI DataSource

Summary

XIV
Chapter 7: Object/Relational Mapping

Background
- Basic O/R Mapping
- Object Query Languages
- Transparent Persistence
- When to Choose O/R Mapping

O/R Mapping Support in Spring
- Data Access Objects
- Transaction Management

iBATIS SQL Maps
- Mapping File
- DAO Implementation
- Setup in a Spring Context
- Transaction Management
- iBATIS Summary

Hibernate
- Mapping File
- DAO Implementation
- Setup in a Spring Context
- Transaction Management
- Open Session in View
- BLOB/CLOB Handling
- Hibernate: Summary

JDO
- Persistent Object Lifecycle
- DAO Implementation
- Setup in a Spring Context
- Transaction Management
- PersistenceManager Lifecycle
- Open PersistenceManager in View
- JDO Dialect
- JDO: Summary

Other O/R Mapping Tools
- Apache OJB
- TopLink
- Cayenne
- JSR-220 Persistence

Summary

Chapter 8: Lightweight Remoting

Concepts and Scope
Contents

Common Configuration Style

Hessian and Burlap
   Accessing a Service
   Exporting a Service

HTTP Invoker
   Accessing a Service
   Exporting a Service
   Customization Options

RMI
   Accessing a Service
   Stub Lookup Strategies
   Exporting a Service
   Customization Options
   RMI-IIOP

Web Services via JAX-RPC
   Accessing a Service
   Exporting a Service
   Custom Type Mappings

Summary

Chapter 9: Supporting Services

JMS
   Introduction
   Motivation for Spring's JMS Support
   JMS Access via a Template
   Exception Handling
   ConnectionFactory Management
   Message Converters
   Destination Management
   Transaction Management
   JmsGatewaySupport
   The Future

Scheduling with Spring
   Timers Versus Quartz
   Timers
   Quartz

Sending Email with Spring
   Getting Started
   Reusing an Existing Mail Session
   Mail Sending Using COS
   Generic Mail Manager
Scripting
  - Consistent Model
  - Other Scripting Languages

Summary

Chapter 10: Acegi Security System for Spring

Enterprise Application Security Choices
  - Typical Requirements
  - Acegi Security in a Nutshell
  - Java Authentication and Authorization Service
  - Servlet Specification

Acegi Security Fundamentals
  - Authentication
  - Storing the Authentication Object
  - Authorization
  - Domain Object Instance Security

Example Code
  - Introducing the Sample
  - Security Unaware Implementation
  - Security Approach
  - Authentication
  - Authorization

Summary

Chapter 11: Spring and EJB

Deciding on the Need for EJBs

Accessing EJBs
  - The Boilerplate Approach
  - The Spring Approach

Implementing EJBs with Spring
  - Stateless Session Beans
  - Stateful Session Beans
  - Message Driven Beans
  - A Digression into XDoclet

Singleton Container Access, Good or Evil?
  - ContextSingletonBeanFactoryLocator and SingletonBeanFactoyLocator
  - A Shared Context as the Parent of a Web-App Application Context
  - Using a Shared Context from EJBs
Contents

Testing Concerns
Implement Business Functionality in POJO Delegate
Use a Mock EJB Container
Integration Testing Inside the AppServer

Summary

Chapter 12: Web MVC Framework

A Simple Example
General Architecture
Web MVC Concepts
Generic Web MVC Using a Dispatcher and Controllers
Requirements for a Good Web MVC Framework
Spring's Web MVC Equivalents

Infrastructural Components
The DispatcherServlet
The WebApplicationContext

Workflow Involved with Handling Requests
Common Layout of a Spring MVC Web Application

HandlerMappings
BeanNameUrlHandlerMapping
SimpleUrlHandlerMapping
CommonsPathMapUrlHandlerMapping
More Than One HandlerMapping

HandlerExecutionChain and Interceptors
WebContentInterceptor
UserRoleAuthorizationInterceptor
Other Handler Interceptors Provided by Spring MVC

Handlers and Their Adapters
ModelAndView and ViewResolvers
UrlBasedViewResolvers
BeanNameViewResolver and XmlViewResolver
ResourceBundleViewResolver
Chaining View Resolvers

Locale Changing and Resolution
HandlerExceptionResolvers

Controllers
WebContentGenerator
AbstractController
UrlFilenameViewController
ParameterizableViewController
MultiActionController
Data Binding
Convenient Functionality When Working with Data Binding

Real-World Examples of Using Controllers
Viewing a List of Performances Using the AbstractController
Making a Reservation

Wizard Functionality
Basic Configuration
Template Methods
Flow of a Wizard
Page Changes, Numbering, and Other Actions

Extending Spring’s Handler Infrastructure

Uploading Files
Configuring the Multipart Resolver
Creating a Form to Upload a File
Handling Uploaded Data

Testing Your Controllers
Testing Without the Application Context
More Extensive Testing Using Mock Objects
Testing Using the Complete Application Context
Other Ways to Test a Web Application

Summary

Chapter 13: Web View Technologies

An Example
General Configuration
JavaServer Pages
FreeMarker
Generating PDFs Using iText

Considerations When Choosing a Technology

View Objects and Models

Features Offered by the AbstractView
Issuing a New Request Using the RedirectView
Using View Prefixes to Issue Forwards or Redirects

JavaServer Pages
Configuring Your Application to Use JSP
Creating Forms Using Custom Tags
Using Tag Files to Create Reusable Elements

Velocity and FreeMarker
Configuring the View Resolver
Using the Form Simplification Macros
Contents

Tiles 5:
Views Based on XML and XSLT 5
Excel and Other Document-Based Views 5
Generating an Excel File from Our List of Shows 5
Using a Template as a Basis for Your Excel File 5
Other Document-Based Views 5
Using a HandlerInterceptor to Distinguish Between HTML and Excel 5
Implementing Custom Views 5
View and AbstractView 5
Implementing a View Generating XML from a Data Object 5
Considerations When Implementing Custom Views 5
Summary 5

Chapter 14: Integrating with Other Web Frameworks 5

Considerations When Choosing an MVC Framework 5
Comparing the Traditional Web MVC Frameworks 7
Integrating with Spring: Core Concepts 5
WebWork Integration 5
Setting Up the ObjectFactory 5
Struts Integration 5
Using ActionSupport 5
Using DelegationRequestProcessor and DelegationActionProxy 5
Using an Auto Wiring Base Action 5
Tapestry Integration 5
Getting Beans for Tapestry 5
The Page Class 5
The Page Definition 5
The Page Template 5
Tapestry Integration Final Thoughts 5
JavaServer Faces Integration 5
Summary 5

Chapter 15: The Sample Application 5

Choice of Server Technologies 5
Application Layers 5
Persistence Layer 5
Data Model 5
Domain Object Model 5
Object/Relational Mapping 5
DAO Implementation 5
Data Access Context

**Business Service Layer**
- Services
- Application Context

**Web Layer**
- Flow Through the Application
- Configuring the Application via web.xml
- Web Controllers
- View Technology

**Comparison with the J2EE Design and Development Implementation**
- Simpler Technology
- Database Changes

**Server Configuration**
- MySQL
- Tomcat

**Build and Deployment**
- Creating and Loading the Database Tables
- Building the Application and Deploying It to the Tomcat Server

**Summary**

---

**Chapter 16: Conclusion**

**Problems Solved by Spring**

**The Spring Solution**

**Guidelines for Spring Development**
- Technology Choices
- Layer by Layer
- Structuring Applications
- Testing Your Applications

**Related Projects**
- Acegi Security for Spring
- Other Projects

**Spring in Non-J2EE Environments**

**To Find Out More**
- Books and Articles
- Online Resources
- Sample Applications

**The Future**

**Smments for the Sample Application**
## Contents

**User Populations**
- Public Internet Users
- Box Office Users
- Administrators

**Assumptions**

**Scope Limitations**

**Delivery Schedule**

**Internet User Interface**
- Basic Workflow
- Error Handling
- Application Screens

**Non-Functional Requirements**

**Hardware and Software Environment**

Index