Integrating Spring 2.5 with iBatis

Course Summary

Description

This leading-edge course provides added coverage of Spring's Aspect-Oriented Programming and the use of Annotations**.** This course thoroughly explores iBATIS and how to apply iBATIS to persistence requirements in Java and J2EE applications. Students then learn how to integrate iBATIS as the persistence layer for Spring applications. The Spring framework is an application framework that provides a lightweight container that supports the creation of simple-to-complex components in a non-invasive fashion. Spring's flexibility and transparency is congruent and supportive of incremental development and testing. The framework's structure supports the layering of functionality such as persistence, transactions, view-oriented frameworks, and enterprise systems and capabilities. Spring's Aspect-Oriented Programming (AOP) framework enables developers to declaratively apply common features and capabilities across data types in a transparent fashion. Spring makes J2EE development easier. Spring simplifies commons tasks and encourages good design based on programming to interfaces. Spring makes your application easier to configure and reduces the need for many J2EE design patterns. Spring puts the OO design back into your J2EE application, and it integrates nicely with JSF.

Objectives

At the end of this course, students will be able to:

- Explain the issues associated with complex frameworks such as J2EE and how Spring addresses those issues
- Understand the relationships between Spring and J2EE, AOP, IOC, JDBC, iBATIS, JSF, Struts, JMS, and EJBs.
- Discuss the challenges to adopting Spring
- Write applications that take advantage of the Spring container and the declarative nature of assembling simple components into applications.
- Understand and work with various options for integrating persistence into a Spring application.
- Work with Spring's support for transactions
- Understand and work with various options for integrating view-oriented frameworks for web applications into Spring. Frameworks that are examined include Spring MVC, JSF, and Struts.

Topics

- Introduction to the Spring 2.5 Framework
- Aspect-Oriented Programming
- Data Access
- Spring Views
- Spring Security Framework
- Spring and iBATIS

Audience

This course is geared for experienced Java developers who need to understand what the Spring Framework is in terms of today's systems and architectures, and how to use Spring in conjunction with other technologies and frameworks.

Prerequisites

Students should have practical basic Java development experience.

Duration

Five days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Integrating Spring 2.5 with iBatis
Course Outline

I. Introduction to the Spring 2.5 Framework
- A. Understanding the value of Spring
- B. Inversion of Control
- C. Dependency Injection
- D. Spring Overview
- E. Spring Application Architectures
- F. Spring Container
- G. Managing the Container
- H. Access to Services and Resources
- I. Application Contexts
- J. Beans as Components
- K. Beans and Factories
- L. XML Bean Configuration
- M. Bean Definition and Dependencies
- N. Bean Lifecycle
- O. Customization Options
- P. Post-Processors
- Q. Property Editors

II. Aspect-Oriented Programming
- A. AOP Benefits and Concepts
- B. Aspect Defined
- C. Decoupling Through Aspects
- D. Code Generation Styles
- E. Cross-Cutting Concerns
- F. Spring's AOP Framework
- G. Advice and Weaving
- H. Proxies: Cost/Benefit
- I. Types of Advice
- J. Interceptor Chain
- K. Joinpoints and PointCuts
- L. Advisors
- M. Working with Proxies
- N. Annotations and AOP
- O. Aspects, Advice, and Pointcuts Using Annotation
- P. Introductions
- Q. Introductions and Annotations

III. Data Access
- A. Data Access Pattern
- B. Overview of Persistence Layer and Transactions
- C. Transaction Overview
- D. Spring Transactions
- E. Defining Spring Transactions
- F. Working with Demarcation
- G. Managing Spring Transactions
- H. Spring JDBC
- I. Spring JDBC Architecture
- J. Working with JDBC Template
- K. Database Operations
- L. Handling JDBC Exceptions

IV. Spring Views
- A. Spring/Web Framework Architecture
- B. Spring MVC
- C. Spring MVC Architecture
- D. Spring MVC Components
- E. Spring MVC Flow
- F. Dispatcher and Controllers
- G. Handlers and Mapping
- H. Interceptors
- I. Spring and Struts (optional)
- J. Spring/Struts Architecture
- K. Integrating Struts into Spring
- L. Spring and JSF (optional)
- M. Spring/JSF Architecture
- N. Integrating JSF into Spring

V. Spring Security Framework
- A. Understand basic security concepts
- B. Secure Web Pages
- C. Use Multiple Roles – Secure pages based on roles
- D. Allow login and logout
- E. Create custom login pages
- F. Use JSP Tags to conditionally allow content based on roles
- G. Securing Services based on roles
- H. Securing Services based on identity

VI. Spring and iBATIS
- A. iBATIS Overview
- B. ORM Mapping Issues
- C. iBATIS Architecture
- D. Benefits and Goals of iBATIS
- E. Ibatis SQLMapper
- F. Mapping Classes to Databases
- G. Configuring iBATIS
- H. Mapping Configuration
- I. Working with SQLMapper framework
- J. Handling Transactions with iBATIS
- K. Simplifying iBATIS Configuration with Spring
- L. Configuring iBATIS in Spring
- M. Spring's SqlMapClientFactoryBean
- N. Using iBATIS from Spring

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically