Spring 3.0, Rich Interfaces, and the Web

Course Summary

Description

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.

As an enabler for rich web interfaces, the Spring framework represents a significant step forward. If you want to deliver a web application within the Spring framework, you'll find this course essential.

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

- Work with the technologies that form the foundation for Ajax
- Integrate Ajax into Spring to provide a rich, interactive web interface
- Understand how to use Struts and JSF within the Spring framework
- Use Spring's Web Flow as a framework to build interfaces for complex web applications
- Work with HTTPUnit and Spring to facilitate unit testing in the context of the web
- Interoperate with RESTful services from within Spring 3.0
- Understand and work Spring Security to acquire and process authentication credentials as well as enforce authorization on enterprise resources
- Understand how to defend Spring applications from the perspective of application security

Topics

- Review of the Spring 3.0 Framework and Spring MVC
- Spring and Web Flow
- Spring and JSF (Optional)
- Spring and Struts (Optional)
- Web Testing in Spring
- Spring Security Framework
- Spring Security – the Web
- Spring and Application Security
- Ajax Overview/Review
- Spring and Ajax

Audience

This course is designed for developers who need to understand how and when to use Spring applications with the web.

Prerequisites

Students should have practical basic Java development experience as well as an understanding of the Spring framework as well as Struts. They also should have attended Core Spring 3.0 Framework prior to taking this class.

Duration

Three 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!"

Spring 3.0, Rich Interfaces, and the Web

Course Outline

I. Review of the Spring 3.0 Framework and Spring MVC
- A. Spring Application Architectures
- B. Spring Container
  - 1. Managing the Container
  - 2. Application Contexts
- C. Beans as Components
  - 1. Beans and Factories
  - 2. XML Bean Configuration
  - 3. Autowiring
  - 4. Bean Lifecycle
  - 5. Factory Beans
- D. Data Handling
  - 1. Validation
  - 2. Type Conversion
  - 3. Formatting
- E. Spring Expression Language (SPEL)
  - 1. Using SPEL
- F. Spring Data Access Pattern
  - 1. Spring DAO
  - 2. DAO Implementations
- G. Spring/Web Framework Architecture
- H. Spring MVC
  - 1. Spring MVC Architecture
  - 2. Spring MVC Components
  - 3. Spring MVC Flow
  - 4. Dispatcher and Controllers
  - 5. Handlers and Mapping
  - 6. Interceptors

II. Spring and Web Flow
- A. Stateful Web Application Challenges
- B. Flow-Based Web Development
- C. Web Flow Components
- D. Creating a Flow
- E. Controllers and Executors
- F. States and Variables
- G. Triggering Events
- H. Subflows and Substates

III. Spring and JSF (Optional)
- A. Spring/JSF Architecture
- B. Integrating JSF into Spring
- C. Resolving Spring bean references from JSF
- D. Making Spring bean references from JSF

IV. Spring and Struts (Optional)
- A. Spring/Struts Architecture
- B. Integrating Struts into Spring
- C. Using Spring's ActionSupport
- D. Using DI with Struts

V. Web Testing in Spring
- A. Spring and Testing
- B. Spring, DI, and Testing
- C. Spring-Specific Testing
- D. Testing Spring MVC Controllers
- E. Using Mock Objects
- F. Assertions and Testing Methods

VI. Spring and RESTful Services
- A. RESTful Services
- B. Request Mapping to Resources
- C. HTTP Method to Java Method
- D. URI Mapping to Parameters
- E. Resolving Content Types
- F. Integration into MVC
- G. RequestMapping Annotation
- H. PathVariable Annotation
- I. Adjusting to Content Types

VII. Spring Security Framework
- A. Core Security Concepts
- B. Spring Security Framework
- C. Spring Works Well With The Following
- D. Other Spring Features
- E. Other Security That Works Well With Spring

VIII. Spring Security – the Web
- A. Spring Security Transparent To Client
- B. Run-As And After-Invocation Managers
- C. Spring Security - Securing A Web Page
- D. Adding Core Security To A JEE Web Site
- E. The Standard Set Of Filters
- F. Configuring Web.Xml
- G. Spring Beans Security File
  - 1. Spring Security Config File
  - 2. Spring Security Extensions
- H. Http Section
  - 1. Http Auto-Config="True"
  - 2. <Intercept-Url />

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

Spring 3.0, Rich Interfaces, and the Web

Course Outline (cont'd)

I. Pages Without Access Control
J. Using Spring Security
   1. Customizing The Experience
   2. Protecting Different Pages With Different Roles
   3. Adding A Logout Page
   4. Adding A Custom Login Page
K. Using JSP Tags To Conditionally Show Content
L. Securing Java Code
   1. Security Guidance
   2. Using Spring Security With Spring Beans
   3. Security Within The Bean
M. User Based Vs. Role Based Security
   1. User Based Security
   2. Use SecurityContextholder To Get User Details

IX. Spring and Application Security
A. Application Security Overview
B. OWASP Top Ten Vulnerabilities
C. SQL Injection
D. Cross-Site Scripting (XSS)
E. Defending Applications Using Spring
F. Defending Against SQL Injection
G. Defending Against XSS

X. Ajax Overview/Review
A. XMLHttpRequest (XHR) Mechanics
   1. XMLHttpRequest Overview
   2. Working with the XHR Object
   3. XHR Operations and Properties
H. JavaScript Mechanics
   4. JavaScript Overview
I. CSS Mechanics
   1. CSS Overview
J. XML Mechanics
   1. XML Overview
N. JavaScript Applied in Ajax
   1. Handling Requests
O. Ajax Applications
   1. Example Forms
   2. Data Processing

XI. Spring and Ajax
A. Direct Web Remoting (DWR)
   1. Mechanics and Communication
   2. Configuring DWR
B. Spring and DWR
   1. Springs and DWR
   2. Configuring DWR within Spring
   3. Handling Ajax Requests with DWR

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