Web Services, JSF, and Spring

Course Summary

Description

This course introduces J2EE developers to the fundamentals of Service Oriented Architectures (SOA) and the core standards that enable Web Services. It provides them hands-on experience with implementations of the Java XML and Web Service APIs including JAXP, JAXB, SAAJ, JAX-WS/RPC, WSEE, and XWSS. Today's development environments are increasingly dominated by sophisticated tooling that makes the initial development of web services less arduous. This course is focused on providing an understanding of the fundamental technologies used in web services. This understanding is critical to being able to diagnose, troubleshoot, tune, and perform other lifecycle activities.

Helping front-end developers, back-end developers, and architects understand how they can get the most out of JavaServer Faces (JSF), this course explores the official standard for simplifying Java web development, explaining what JSF is, how it works, and how it relates to other frameworks and technologies like Struts, Servlets, JSP, and JSTL. The Spring portion of 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 persistence and Hibernate.

Topics

- Path to Useful Web Services
- Foundation for Web Services
- Binding: SOAP
- Description: WSDL
- Web Services in Java: JAX-WS/RPC
- Web Services in J2EE: WSEE
- Discovery: Finding Web Services
- Security: WS-Security and Defenses
- JSF: Architecture
- JSF: Processing, Components, and JSP 2.0
- JSF: Data Validation, Conversion, and Rendering
- Spring Data Access
- Spring and Hibernate

Audience

This course is designed for experienced J2EE developers.

Prerequisites

Students should have working experience with J2EE components.

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

Web Services, JSF, and Spring

Course Outline

I. Path to Useful Web Services

A. This section provides a high-level overview of SOA and web services
B. It establishes the overall topology of the various standards and specifications from W3C and Java. Various web services stacks and implementation frameworks are described
C. Students interact with various publicly available web services as well as implement, deploy, and interact with their own simple web service
D. This is the first of several opportunities that students are provided with an opportunity to troubleshoot and diagnose working web services

II. Foundation for Web Services

A. This section covers the underlying technologies that make web services possible
B. XML namespaces and schema are reviewed to ensure students understand those critical topics
C. The primary Java XML processing APIs of JAXP and JAXB are covered with students getting some hands-on experience using JAXB to implement Java/XML interoperability

III. Binding: SOAP

A. SOAP is the protocol of choice for binding a service client to a service provider
B. This section provides an in-depth coverage of the SOAP specification
C. Students then interact with several web services to understand how SOAP operates in the real world as well as how errors are handled
D. The Java API for building and consuming SOAP messages (SAAJ) is covered with students using it to construct SOAP messages to send to a working web service

IV. Description: WSDL

A. W3C's WSDL specification has become the option of choice for describing web services. Students are given a complete tour of the vocabulary and structure of WSDL
B. They then take a variety of working examples, dissect them, and compare them the associated web services

V. Web Services in Java: JAX-WS/RPC

A. The central Java API for working with web services is JAX-RPC and its replacement JAX-WS
B. This section covers both APIs, explaining their place in the web services stack as well as the differences between the two
C. Students then take an existing Java POJO and generate web service based on that class. They then take the resulting WSDL and use it to generate a client for their new web service

VI. Web Services in J2EE: WSEE

A. WSEE provides a standardized approach to implementing, packaging, and deploying web services
B. This section shows students how to expose web components and stateless session EJBs as HTTP-based web services in a WSEE compliant fashion
C. Students are also introduced to Handlers, including their use, implementation, and deployment within WSEE.

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

Web Services, JSF, and Spring

Course Outline (cont'd)

VII. Discovery: Finding Web Services

A. There are many options for publishing and discovering web services
B. UDDI is covered in this section, but other options are emerging in the real world
C. This section also covers WS-Discovery and WSIL
D. Students use several WSIL instances to find and interact with working web services

VIII. Security: WS-Security and Defenses

A. Security on a variety of levels is becoming an increasing concern
B. This section covers the various components of WS-Security, including XML Signature, XML Encryption, and the use of tokens
C. Students are shown how these specifications only go so far in protecting web service and enterprise assets
D. They are introduced to a variety of recognized XML and web service vulnerabilities that are not specifically addressed by WS-Security
E. Students have an opportunity to try some of these attacks and to implement some basic defenses

IX. JSF: Architecture

A. In this section, students are introduced to the JavaServer Faces specification, what it addresses and how it compares to other frameworks such as Struts
B. The JSF architecture, including basic components and processing flow, are covered as well. Students get an opportunity to write a basic JSF application

X. JSF: Processing, Components, and JSP 2.0

A. Students are shown request processing and page navigation as well as the context objects, lacking beans, and various JSF tag libraries and user interface components
B. The JSP 2.0 EL is also covered since it is used extensively in JSF

XI. JSF: Data Validation, Conversion, and Rendering

A. In this final JSF section, students are shown how to apply both standard and custom data validators, converters, and renderers
B. Numerous examples are explored and used by students

XII. Spring Data Access

A. After a brief overview of the Spring framework, students are introduced to the Data Access Pattern that is used with Spring
B. An overview of the persistence layer and transactions are also covered.

XIII. Spring and Hibernate

A. In this section, the Spring/Hibernate architecture is covered as well as working with Hibernate DAOs in Spring
B. Students are introduced to the Hibernate template and given an opportunity to work with it.

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