ProTech "Charting the Course ...

protechtraining.com

... to Your Success!"

Developing Java Web Services v6.0

Course Summary

Description

This course is a comprehensive look at the state of the art in developing interoperable web services on the Java EE 6 platform. Students learn the key standards for both SOAP-based and RESTful services, and the Java architectures that have evolved to build interoperable services and clients. We begin with an introductory module that covers both SOAP-based and RESTful services, and therefore both JAX-WS and JAX-RS. We then discuss JAXB, as this highly useful XML API is integral to both. The remainder of the course is split between two larger modules, each of which falls on one side of the fence: either SOAP services with JAX-WS, or RESTful services with JAX-RS. We cover SOAP, WSDL, and both WSDL-driven and Java-driven development paths for JAX-WS, as well as client-side development. Then students work with the Jersey implementation of JAX-RS to create RESTful services, from simple single-value interactions to more sophisticated services that manage CRUD (create/retrieve/update/delete) operations on more complex data types, using JAXB to marshal and unmarshal data over the wire.

Objectives

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

- Be able to describe the interoperable web services architecture, including the roles of SOAP and WSDL in component-based services and XML and HTTP in the REST architecture.
- Understand the importance of the WS-I Basic Profile for interoperable web services.
- Build JAX-WS services and clients that take full advantage of the automated data binding of JAXB.
- Build WSDL-to-Java and Java-to-WSDL services, with equal facility.
- Use JAX-RS to develop simple RESTful services.
- Control dispatching to service methods based on URL patterns and HTTP methods.
- Bind request values to method parameters when expressed as HTTP query parameters, form values, headers, cookies, and more.
- Manage XML content using XML Schema and JAXB.
- Incorporate XML entities into service input and output.
- Take advantage of lifecycle and context services available to JAX-RS services as Java EE components.

Topics

- Overview of Web Services
- Web Services for Java EE
- The Java API for XML Binding
- The Simple Object Access Protocol
- Web Services Description Language
- The Java API for XML-Based Web Services
- WSDL-to-Java Development
- Client-Side Development
- Java-to-WSDL Development
- The Java API for RESTful Services
- Dispatching Requests to Methods
- Parameter and Return Types
- Entities and Complex Content
- Context and Lifecycle

Audience

This course is designed for the experienced Java Programmer who wants to gain a comprehensive look at the state of the art in developing interoperable web services on the Java EE 6 platform.

Prerequisites

Strong Java programming skills are essential. A Java programming course is excellent preparation. Students must be able to read XML documents and to write well-formed XML by hand -- consider an introductory XML course. Knowledge of XML Schema will be helpful, too, but is not a strict prerequisite – consider an XML Schema course. Experience with other Java EE standards, especially servlets and JSP, will be very helpful in class, but is not strictly required.

Duration

Five days

ProTech "Charting the Course ...

protechtraining.com

... to Your Success!"

Developing Java Web Services v6.0

Course Outline

I. Overview of Web Services
- A. Why Web Services?
- B. Service-Oriented Architecture
- C. HTTP and XML
- D. SOAP
- E. WSDL
- F. The SOAP Vision
- G. The REST Vision
- H. UDDI
- I. The WS-I Basic Profile
- J. Security

II. Web Services for Java EE
- A. Hosting Web Services: Scenarios
- B. Web Services for Java EE
- C. JAX-WS and JAXB
- D. Web-Services Metadata
- E. WSDL-to-Java and Java-to-WSDL Paths
- F. Provider and Dispatch APIs
- G. SAAJ and JAXP
- H. JAX-RS for RESTful Services
- I. JAXR

III. The Java API for XML Binding
- A. The Need for Data Binding
- B. XML Schema
- C. Two Paths
- D. JAXB Compilation
- E. Mapping Schema Types to Java
- F. Java-to-XML Mapping Using Annotations
- G. Marshaling and Unmarshaling
- H. Working with JAXB Object Models

IV. The Simple Object Access Protocol
- A. Messaging Model
- B. Namespaces
- C. SOAP over HTTP
- D. The SOAP Envelope
- E. The Message Header
- F. The Message Body
- G. SOAP Faults
- H. Attachments

V. Web Services Description Language
- A. Web Services as Component-Based Software
- B. The Need for an IDL
- C. Web Services Description Language
- D. WSDL Information Model
- E. The Abstract Model -- Service Semantics
- F. Message Description
- G. Messaging Styles
- H. The Concrete Model -- Ports, Services, Locations
- I. Extending WSDL -- Bindings
- J. Service Description

VI. The Java API for XML-Based Web Services
- A. Two Paths
- B. How It Works: Build Time and Runtime
- C. The Service Endpoint Interface
- D. Working from WSDL
- E. Working from Java
- F. RPC and Document Styles
- G. One-Way Messaging
- H. Binary Protocols

VII. WSDL-to-Java Development
- A. The @WebService Annotation
- B. Generated Code
- C. Scope of Code Generation
- D. Parameter Order
- E. More JAXB: Mapping Collections
- F. More JAXB: Mapping Enumerations
- G. Applying JAXB Customizations

VIII. Client-Side Development
- A. Stubs and Proxies
- B. Generated Code
- C. Locating a Service
- D. Invoking a Service
- E. The @WebServiceRef Annotation

Developing Java Web Services v6.0

Course Outline (cont'd)

IX. Java-to-WSDL Development
- A. Generating the WSDL and Schema
- B. The @WebMethod, @XmlParam, and Related Annotations
- C. More JAXB: Mapping Inheritance
- D. Controlling the XML Model
- E. Controlling the WSDL Description
- F. JAXB Customizations with @XmlJavaTypeAdapter

X. The Java API for RESTful Services
- A. Applications
- B. Resources
- C. Sub-Resources
- D. Providers
- E. Scanning and @ApplicationPath

XI. Dispatching Requests to Methods
- A. The Application Path
- B. The @Path Annotation
- C. The HTTP Method Annotations
- D. Sub-Resource Locators
- E. Annotation Inheritance and overriding

XII. Parameter and Return Types
- A. Simple Parameter Types
- B. @Consumes and @Produces Annotations
- C. @XXXParam Annotations
- D. The @DefaultValue Annotation
- E. Return Types
- F. Binary Content
- G. Delivering a File

XIII. Entities and Complex Content
- A. Entity Providers
- B. Built-In Entity Providers
- C. Working with XML
- D. Driving XML Representations from Schema

XIV. Context and Lifecycle
- A. Reading Web Resources
- B. Finding Java EE Components
- C. Finding Databases
- D. Security Contexts

XV. Appendix A. Course Tools and Utilities

XVI. Appendix B. Handy Guide to Web-Services Acronyms