Developing SOAP Web Services in Java v2.2

Course Summary

Description
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 -- SOAP, WSDL, and the WS-I Basic Profile -- and the Java architecture that has 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. But JAX-WS is central to the course, and we cover both WSDL-driven and Java-driven development paths, as well as message handlers and attachment support. With the new Provider and Dispatch APIs, it's now much easier to integrate SAAJ, JAXB, and JAXP code into services and clients, and we explore these strategies in depth as well.

Objectives
At the end of this course, students will 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.
- Apply advanced techniques and best practices including proper exception handling, care around possible polymorphism, and use of context and lifecycle services.
- Use lower-level SOAP and XML APIs for services and/or clients.
- Customize data binding by specifying specific type mappings or altering method or parameter names.
- Incorporate binary data, such as images, into service and client code.

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
- Exception Handling
- JAX-WS Best Practices
- Metadata
- Provider and Dispatch APIs
- The SOAP with Attachments API for Java
- Message Handlers
- Handling Binary Content
- Appendix A. Course Tools and Utilities
- Appendix B. Handy Guide to Web-Services Acronyms

Prerequisites

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

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!"

Developing SOAP Web Services in Java v2.2

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

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

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

Developing SOAP Web Services in Java v2.2

Course Outline (cont'd)

X. Exception Handling
- A. SOAP Faults vs. Java Exceptions
- B. Mapping Faults from WSDL
- C. Mapping Exceptions from Java
- D. JAX-WS Exception API and Handling
- E. Client Exception Handling

XI. JAX-WS Best Practices
- A. Which Way to Go?
- B. Interoperability Impact
- C. Portability Impact
- D. Polymorphism in Web Services
- E. Web Services as Java EE Components
- F. Lifecycle Annotations
- G. Context Interfaces

XII. Metadata
- A. Abstract and Concrete Model Metadata
- B. Defaults
- C. Annotations
- D. webservices.xml and web.xml
- E. Best Practices and Pitfalls

XIII. Provider and Dispatch APIs
- A. Stepping Down
- B. The @WebServiceProvider Annotation
- C. The Provider<T> Interface
- D. Implementing a Provider
- E. The @ServiceMode Annotation
- F. JAXB Without WSDL
- G. Integrating JAXP
- H. The Dispatch<T> Interface
- I. Building Clients

XIV. The SOAP with Attachments API for Java
- A. The SAAJ Object Model
- B. Parsing a SOAP Message
- C. Reading Message Content
- D. Working with Namespaces
- E. Creating a Message
- F. Setting Message Content
- G. Relationship to the DOM

XV. Message Handlers
- A. Handling SOAP Headers
- B. Handlers and Handler Chains
- C. Configuration by XML or Annotation
- D. MessageContext and SOAPMessageContext
- E. Processing Model and Patterns
- F. Logical and Protocol Handlers
- G. Client-Side Handlers

XVI. Handling Binary Content
- A. The WS-I Attachments Profile
- B. Using base64Binary
- C. MTOM and XOP
- D. JAX-WS Support
- E. Configuration by XML or Annotation
- F. Client-Side Support
- G. SAAJ Support

XVII. Appendix A. Course Tools and Utilities

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

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