

Course 418:
**Implementing a Service Oriented Architecture with
XML Web Services in Java
(4 days)**

This in-depth, advanced workshop will explain the need for a service-oriented architecture (SOA), best practices for integrating across enterprise systems using a SOA and provide concrete details on implementing and orchestrating web services with the Java API for XML Web Services (JAX-WS).

JAX-WS is now the standard way of implementing web services in Java (superseding JAX-RPC, Apache Axis, etc.) and provides means of creating and accessing web services that are interoperable with the .NET Windows Communication Foundation (WCF).

This course also provides a comprehensive introduction to XML and how to parse, create, transform and bind to XML from Java programs.

Who can benefit?

This is an intermediate level Java programming course, designed for architects and developers who wish to implement integration architectures using XML web services in Java. The student should be an experienced JEE / Java programmer, with practical development experience.

Prerequisites:

Attendees should have proficiency in Java equivalent to the level of ROI's course **430: *Essential Java***.

Learning Objectives

Students will leave this course armed with the skills to:

1. Design a Service Oriented Architecture with XML Web Services
2. Describe the structure of an XML document using schema
3. Interact with XML documents from Java programs using SAX, DOM, JAXB and TRaX
4. Monitor SOAP messages between web services and clients
5. Write and deploy WSDL for WSDL-first web services
6. Create WSDL-first and code-first web services
7. Create synchronous and asynchronous web service clients
8. Interpose custom SOAP handlers on client-side and server-side
9. Orchestrate web services
10. Enable message optimization, reliability and security with WS-* standards

Course Customization Option:

This course can be customized to meet your needs. Please contact your ROI Representative to schedule a call with the course author. In most cases, ROI will customize the class at no charge!



Course Outline

Chapter 1: Introduction to SOA and Web Services

- What is SOA?
- Why SOA?
- SOA Implementation Options
- SOA in the Real World
- Quick introduction to Eclipse, Ant and Tomcat
- **Try It Now: Exploring the case study**

Chapter 2: Designing a SOA with XML Web Services

- Designing a SOA Integration Architecture
- Implementing a SOA with Web Services
- What SOA does not provide
- Business Process Modeling with SOA
- **Exercise 1: Designing Integration Layer**

Chapter 3: XML in the Enterprise

- XML: syntax and grammar
- XML Namespaces
- XML Schema
- Simple Types, Complex Types
- Interoperability limitations
- Binary data
- Facets, enumeration, union
- **Exercise 2: Writing XML payload to conform to a schema**
- XPath
- XSLT
- **Try It Now: Invoke XSLT from command-line**

Chapter 4: Java and XML

- Java APIs for XML Parsing
- SAX, DOM, and JAXB
- **Try It Now: SAX**
- **Try It Now: DOM**
- Binding XML to objects
- JAXB limitations
- **Exercise 3: JAXB Exercise: Creating JavaBeans from permanent exhibit document**
- Java Transformation API
- **Try It Now: TRaX**

Chapter 5: Web Service Ingredients (SOAP and WSDL)

- Role of SOAP in Web Services
- Structure of a SOAP message
- Why a Header?
- Types of SOAP messages
- **Exercise 4: Monitoring SOAP interactions**
- Role of WSDL in Web Services
- Interface
- Implementation
- Deploying a WSDL
- **Exercise 5: Write and deploy WSDL for permanent gallery**



Chapter 6: Creating a Web Service from WSDL

- Why choose WSIT/JAX-WS?
- Importing a WSDL and generating a web service endpoint
- Best practices in dealing with generated code
- Customizing the generated code
- Structure of a web service WAR
- **Exercise 6: Create permanent gallery web application**
- SOAPHandler and LogicalHandler

Chapter 7: Building Web Service Clients

- Generating client code from WSDL
- Customizing the created client code
- Synchronous, polling and asynchronous services
- **Exercise 7: Obtaining list of permanent exhibits**
- Asynchronous clients
- Java Executor framework
- **Try It Now: Building an asynchronous client to greeting service**
- SOAPHandler and SOAPMessage
- LogicalHandler
- Changing SOAP messages using DOM, TRaX and JAXB

Chapter 8: Code-first Web Services

- Service annotations
- Generating a web service tie
- **Exercise 8: Exposing list of traveling exhibits**

Chapter 9: Orchestrating Web Services

- Orchestrating web services in Java
- **Exercise 9: Orchestrating exhibits list access in Java**
- Orchestrating web services with BPEL

Chapter 10: Security, Reliability and Addressing

- Message optimization (MTOM and XOP)
- Reliability, addressing
- Security
- Enabling WS-* features
- **Try It Now: Enabling WS-* features on web service**

Chapter 11: XML-level Clients and Services (OPTIONAL)

- Need for XML-level clients and services
- Provider API
- **Exercise 10: Building a Provider Service**
- Dispatch API
- **Exercise 11: Building a Dispatch Client**
- RESTful services
- Try It Now: HTTP Provider Service and Dispatch Client
- Limitations of JAX-WS and Servlet API for RESTful services
- Building a RESTful service with JAX-RS
- Try It Now: JAX-RS

Appendix A: Java Annotations (Using and defining annotations)

Appendix B: Cryptography (Java cryptography APIs, tools, managing keys and certificates)