

## Building XML Web Services with Java: Hands-On - 4 Days

### *Course 577 Overview*

- You Will Learn How To**
- Develop, deploy and monitor Web services and Web service clients with JAX-WS
  - Implement a Service-Oriented Architecture (SOA) using Web services
  - Create and deploy WSDL-first and code-first Web services
  - Build synchronous and asynchronous Web service clients in Java
  - Deliver RESTful Web services for server-side AJAX
  - Secure Web services programmatically and declaratively
- Course Benefits** Web services revolutionize the way businesses interact by enabling interoperability between applications on different hardware and software platforms. The Java APIs for XML Web Services (JAX-WS) deliver a set of powerful tools to develop a Service-Oriented Architecture (SOA). This hands-on course provides the skills to design and build Web services using Java. You develop services and clients using the latest standards-based technologies. You also deploy secure Web services that integrate proven security strategies.
- Who Should Attend** Programmers, architects, managers and those interested in integrating applications over the Web. Course 471, "Java Programming Comprehensive Introduction," or equivalent knowledge is assumed.
- Hands-On Training** Exercises provide practical experience building Web services with Java and include:
- Writing a code-first Web service
  - Binding XML complex types to Java beans
  - Writing and deploying a WSDL
  - Creating a contract-first Web service from WSDL
  - Building asynchronous Web service clients
  - Controlling inventory from a Web browser
  - Authenticating and authorizing access to Web service

## Building XML Web Services with Java: Hands-On - 4 Days

### Course 577 Outline

#### Web Services Overview

##### Interoperable applications with Service-Oriented Architecture (SOA)

- Designing an SOA integration architecture
- Evaluating alternatives to SOA

##### Implementing SOA with Web services

- Core technologies: HTTP, XML, SOAP, WSDL
- What SOA does not provide

#### XML Processing in Java

##### XML essentials

- XML syntax and namespaces
- Describing XML with schema

##### Interacting with XML from Java

- Marshaling and unmarshaling with JAXB
- Customizing XML to Java bindings

#### Defining SOAP Messages with WSDL

##### Structure of SOAP messages

- Role of SOAP in Web services
- Operations, messages and faults

##### Anatomy of a WSDL document

- Defining the interfaces of a Web service
- Specifying implementation
- Deploying WSDL

#### Generating WSDL-first Web Services

##### Architecting a Web Service

- Designing a service endpoint
- Specifying protocol of message interchange
- Preserving flexibility and extensibility

##### Importing a WSDL document

- Building interoperable applications by conforming to Web Services Interoperability (WSI) standards
- Incorporating Web service proxies and adapters
- Implementing a Web service end point using JAX-WS

##### Customizing JAX-WS Web services

- Deploying a Web service WAR file
- Intercepting traffic between Web services and clients
- Optimizing message transmission

#### Exposing Plain Old Java Objects (POJOs) as Web Services

##### Implementing code-first Web services

- Choosing between WSDL-first and code-first Web services
- Generating portable artifacts using JAX-WS
- Preserving maintainability with proxies and adapters

##### Designing reliable and scalable services

- Creating highly parallel Web services
- Bulletproofing multithreaded Web services

##### Improving generated WSDL

- Annotating Java services
- Deploying end points

#### Implementing Web Service Clients in Java

##### Generating client code from WSDL

- Accessing Web services through their WSDL
- Creating client source files from WSDL
- Customizing generated source files with JAX-WS

##### Synchronous, polling and asynchronous services

- Designing and creating one-way services and clients
- Writing multithreaded clients
- Interception and modifying SOAP messages

#### Providing Server-Side AJAX with RESTful Web Services for Interactivity

##### Stateless processing of XML requests

- Building RESTful Web services using JAX-WS
- Implementing a Provider
- Providing client-side interactivity

##### Lightweight clients

- Invoking Web Services with the Dispatch API
- Processing received XML messages

#### Securing Web Services

##### Authenticating and authorizing clients

- Limiting access to Web services and methods
- Providing authentication information to Web services

##### Message-level security

- Transport security vs. end-to-end security
- Turning on WS-Security