

## **Web Services Development Using Eclipse**

**Duration: 5 days**

This course will allow you to gain complete mastery of Web Services development in the open source platform. This five day course will teach you all you need to know about XML schemas, WSDL and SOAP. This course will also teach you various Java specifications that cover Web Services development (JSR 101 and 109).

### **Topics**

- Interoperability issues
- Web Services Description Language (WSDL)
- JAX/RPC
- Design of XML's and schemas
- JSR 109
- Simple Object Access Protocol (SOAP)
- Universal Description, Discovery and Integration (UDDI)
- WS-Security

### **Practical work**

This course explores Web Services Development and the move within organizations to adopt a Service Oriented Architecture (SOA). This course will equip developers and programmers with a working knowledge of all the components of Web Services, using a hands-on experience and applying some of the most popular J2EE API'. The Students will also gain a broad awareness of the implications of SOA and how to best support the adoption of java and SOA through Web Services Platform.

This course covers and intensive hands on exercises; these exercises take audience through all major aspects of the design and development of Web Services.

### **What you will learn**

After completing this course, the attendees should be able to:

- You will master designing XML schema and WSDL.
- Understand the difference between the two SOAP styles (Remote Procedure Calls 'RPC' and document literal)
- Designing and Implement a Web Service using the IDE.
- Implementing various Web Services client using standard Java specifications.
- Ability to register a service in a public or private UDDI registry.
- How to make your web services secure.
- Learn the interoperability issues with other platform, especially with the .NET platform.)

### **Audience**

Developers who will like to learn about how to build a Web Services based solution. In this many advanced topics will be covered that will help them build a solution appropriate for a large business..

### **Prerequisites**

Java programming. Beginner level knowledge of J2EE and XML.

### **Course Content**

#### **1. Overview of Eclipse**

- Overview, The Eclipse Platform, Eclipse Java IDE
- Views, Perspective and Editor Areas
- Basic Operations with Eclipse Views and Perspectives
- The Java Perspective, The Debug Perspective
- Navigator View, Build and Validation
- Templates and Code Completion, Searching, WTP
- The Tomcat Servlet and JSP Web Container
- The JBoss Application Server

## 2. J2EE Overview

- Objectives
- The Goals of Enterprise Applications
- What is Java? , What is J2EE?
- Role of Application Server, The J2EE Specifications
- J2EE Components, What is a Servlet?
- What is a JSP?, What is an EJB?, Model-View-Controller Architecture
- MVC an Example, Web Client, Applet, Application Client
- J2EE Vendor Specifications, Containers, J2EE Blueprint

## 3. Introduction to Web Services

- Objectives
- Service Oriented Architecture (SOA)
- Basic SOA, SOA Runtime Implementation
- SOA, Web Services, Typical Development Workflow
- Advantages of Web Services, Web Services Business Models
- Case Study: Internal System Integration
- Case Study: Business Process Externalization
- SOAP, UDDI, Web Services Description Language (WSDL)
- Web Services Invocation Framework (WSIF)
- WSIF and WSDL, Web Services Interoperability (WS-I)
- WS-I Deliverables, Web Services Support in Eclipse/WTP

## 4. Development using XML

- Objectives
- Enforcing Validity: DTDs, Presentation Style
- Sections of an XML Document, XML Elements
- Nesting and Hierarchy of XML Elements
- XML Overview, Data and Document Structure
- An Employee Document, Tags, First XML Document
- Tag Attributes, Naming Rules, Namespaces
- Using Namespaces, Java API for XML
- The XML Example, Example SAX Handler
- Markup Languages, What is XML?, Why XML?
- An Example of XML Document
- Well-formed vs. Valid XML Document
- Example: Begin Parsing, Once Again With Namespace
- Using DOM to Parse, With Namespace Enabled
- Build DOM Document, Save DOM Document in a File

## 5. XML Schema Basics

- Objectives
- What is XML Schema?, Creating a Schema File
- Defining Abstract Data Types, Adding Restrictions
- Defining a Simple Element, Defining a Complex Element
- Defining Element Attributes, Referring to an Element From Another Element
- Referring to a Schema from a XML Document

## 6. Web Services Description Language (WSDL)

- Objectives
- Sample WSDL Document Structure
- WSDL SOAP Binding Extensions, soap:binding
- soap:operation, RPC or Document Style?, WSDL API for Java
- < operation >, One-way < operation >, Request-Response < operation >
- WSDL Overview, WSDL Document Tags, WSDL Namespaces
- Modeling Complex Message, < binding > < definitions >, < import >, < types >, < message >, < portType > < service > and < port >, More on < port >
- Solicit-Response < operation >, Notification < operation >
- More on < binding >, < binding > Syntax, SOAP Binding Example
- Modeling Simple Operation, Modeling Complex Operation

## 7. Simple Object Access Protocol (SOAP)

- Objectives
- Message Envelope, Message Envelope - Headers
- Details of the Wrapped Style
- Communication Style, RPC/Encoded Style, RPC/Literal Style
- SOAP HTTP Request Example, SOAP HTTP Response Example
- Header Attributes, SOAP Body, SOAP Fault
- Document/Literal Style, Document/Literal Wrapped Style
- SOAP Overview, SOAP In Protocol Stack, SOAP Components

## 8. Java API's for XML (JAX-RPC), Java Specification Request (JSR 109)

- Objectives
- Java to WSDL/XML Mapping, Mapping Tools
- Mapping XML Types to Java Types Complex Types
- JAX-RPC Supported Types, JAX-RPC Server
- Server Side Artifacts, Generating Server Artifacts
- JAX-RPC and WS-I, JAX-RPC Clients
- JAX-RPC Overview, JAX-RPC Framework
- Java to XML Data Conversion
- Mapping XML Types to Java Types Simple Types
- JAX-RPC Client Model, JAX-RPC Client - Static Stub
- Generating Client Side Code, Client Programming Model
- JAX-RPC Client Dynamic Proxy, Dynamic Invocation Interface
- Main Goals of JAX-RPC, Supported Protocols
- Mapping XML Types to Java Types Arrays
- Mapping Concrete WSDL Types to Java Types

## 9. Web Services for J2EE

- Objectives
- Writing a Client, Handlers, Handlers Example(Server-Side)
- Introduction, JSR-109 – Motivation, When to Use JSR 109?
- Handlers Example(Client-Side)
- EJB Module Service, The Client Programming Model
- Developing a Client, The Service Interface
- JSR-109 Roles, The Server Programming Model
- The Service Endpoint Interface, Web Module Service
- Accessing the Service, The Mapping File
- Implementation Class Details, Lifecycle Callback

## 10. JBoss Support for Web Services

- Objectives
- Service in a Web Container, Example Java Class
- Generated Artifacts, JSR 109 Components
- Using wstools, using wstools Server Side
- Using wstools Sample config.xml
- Web Services Deployment Descriptor
- Example: webservices.xml, The Generated WSDL File
- The WSDL Editor, Service in a EJB Container
- Web Services Deployment Descriptor
- Client Development, Generated Artifacts
- Using wstools sample config.xml, The Service Reference

## 11. Introduction to UDDI

- Objectives
- UDDI Overview, UDDI in Web Services Architecture
- UDDI Registry Structure, UDDI Registry Structure -tModel
- UDDI Interactions, UDDI and Eclipse
- UDDI, Businesses and Services in UDDI
- Static and Dynamic Web Services

## 12. Error Handling

- Objectives
- Introduction
- Business Rule Violation

### **13. Web Services Security**

- Objectives
- The Challenges, Public Key Infrastructure (PKI)
- Overview of Web Services Security
- SOAP Message Security, Message Integrity
- Digital Signature, Certificates
- Message Confidentiality, Authentication, Transport Level Security

### **14. Web Services Interoperability**

- Objectives, Introduction
- Goal, What Comes out of WS-I?, Profiles
- Basic Profile 1.1 Highlights
- Simple SOAP Binding Profile 1.0 Highlights
- Basic Security Profile 1.0, .NET Interoperability

### **15. Introduction to Service Oriented Analysis and Design**

- Objectives
- Introduction to SOAD, Applying OOAD Principles
- Why OOAD is not Enough, Granularity
- The Need for Loose Coupling
- The SOAD Methodology, The SOAD Methodology Steps
- Abstraction, Abstraction in SOAD
- Encapsulation, Encapsulation in SOAD
- Modularity, Modularity in SOAD
- Hierarchy, Hierarchy in SOAD
- Stage 1 - Process Modeling, Stage 2 Service Identification
- Stage 3 Service Design & Implementation
- Stage 4 - Process Implementation

### **16. Best Practices**

- Objectives
- Architecture Best Practices, Data Format Best Practices
- Security Best Practices, Programming Model Best Practices