

## Next Level Web Services & REST for Java / JEE (TT7380)

**Modality:** Virtual Classroom

**Duration:** 4 Days

### **About this course:**

Mastering Java Services is a five-day hands-on JEE / Java EE training course geared for experienced Java developers new to JEE, who need to get up and running with essential dynamic web and services development and skills. Created in collaboration with several leading JEE / Java EE authors and industry experts, this comprehensive course teaches students how to design and program web components, including all the important concepts and hands on labs that will have you building working server-side applications in no time flat. This course provides core JEE knowledge and skills that can be used as the foundation for developing production-quality web applications to a basic level.

Servlets are a key server-side Java technology for building web applications. Servlets are programs that run on a web server; they can respond to client requests and create dynamic content. Servlets allow flexible generation of dynamic content. Additional technologies allow one to separate static from dynamic content while harnessing the power of servlets.

SOA and Web Services represent a groundbreaking evolution in distributed computing. The concepts are not altogether new, but the application of them, and the unanimous acceptance of core standards like HTTP, XML, SOAP, WSDL, and the various WS-\* policies, has paved the way for XML Web Services and service-oriented architectures. An overreaching concern related to services is security. Various aspects of security are woven into this course, allowing students to see the entire spectrum of issues as well as solutions. These include encryption, digital signatures, authentication and authorization assertions, and recognized application security issues such as Cross-Site Scripting and Injection attacks.

The average salary of a Java Developer is **\$90,992** per year.

### **Course Objective:**

Students who attend **Mastering Java Services** will leave the course armed with the required skills to design and build scalable, secure, maintainable web applications - leveraging our extensive experience in the delivery of scalable enterprise applications with complex web interfaces based on JEE technologies.

Throughout this training, you will be confronted with common web application design problems and given the tools you will need to solve them, such as JEE design patterns. You will also be exposed to a range of JEE and web technologies such as Servlets, JSPs, JSF, JNDI, CDI and advice on when and how to use them. Students will also learn about the capabilities of servlets, their advantages, servlet architecture, and session management. Developers will also learn about managing resources, deployment, and application models, how to use custom tags, and how to build robust and capable web applications using servlets and other components.

Today's development environments are increasingly dominated by sophisticated tooling that makes the initial development of web services less arduous. The production-level implementation, deployment, and maintenance of web services are far more complex and demanding.

This course focuses on providing an understanding of the fundamental technologies used in web services. This understanding is critical to being able to diagnose, troubleshoot, tune, and perform other lifecycle activities.

Course topics covered include the following areas.

- JEE Application Architecture
- Servlets
- Java Server Pages (JSP), EL, and JSTL
- JavaServer Faces (JSF)

Working in a dynamic, lab-intensive hands-on coding environment students will learn to:

- Understand and intelligently discuss Web Services and the core technologies involved
- Design, develop, and deploy real-world JEE Web ServicesExpose existing Java components as XML Web Services
- Write Java components that access remote Web Services hosted by a third party
- Read and understand a WSDL document
- Parse, process, and respond to a SOAP message
- Understand the concepts behind REST and implement a REST-based web service
- Implement handlers to inject cross-cutting solutions for security, logging, auditing, and other needs

## **Audience:**

This is an intermediate level Java EE (JEE) developer course, designed for experienced Java developers, new to JEE, that need to further extend their skills in JEE web development.

## **Prerequisite:**

Attendees should have an extensive working knowledge in developing basic Java applications.

## **Course Outline:**

### **Module 1: Path to Useful Web Services**

#### **Lesson: Services Via the Web**

- Architectural Style: Common Framework
- Loose Coupling: Spectrum of Options
- Software Agents: Services
- Interacting: Orchestrated
- SOA Reference Architecture

- Service Layers
- Governance and Compliance

### **Lesson: Web Services Overview**

- Web Services Architecturally
- Spec and Standard Evolution
- Web Services Interoperability Organization
- .NET Platform & .NET Web Services
- Java and Web Services
- Exercise: Web Services in Action

### **Lesson: Web Services, Java, and JEE**

- XML Signature
- XML Encryption
- JAXP, JAXB, and JAX-WS
- JEE and Web Services
- Web Services Stacks at a Glance

### **Lesson: Web Services Quickstart**

- “Typical” Web Services Stack
- How Stack is Used on the Service-Side
- How Stack is Used on the Client-Side
- Debugging Web Services
- Exercise: Implementing a Web Service
- Exercise: Debugging Web Services

## **Module 2: Foundation for Web Services**

### **Lesson: XML, Namespaces and Schemas**

- XML Separates Structure, Content and Format
- XML Namespaces
- Namespaces Best Practices
- W3C XML Schemas
- Exercise: Namespaces and Schemas

### **Lesson: XML in Java - JAXP and JAXB**

- JAXP: Java API for XML Processing
- Security Concerns Relative to Parsing
- JAXB: Binding XML to Java
- Exercise: Working With JAXB

## **Module 3: Binding – SOAP/REST**

## **Lesson: SOAP Overview**

- Anatomy of a SOAP Message
- SOAP and HTTP
- SOAP Messaging
- Remote Procedure Calls
- SOAP With Attachments
- The SOAP Envelope
- SOAP Data Model
- Exercise: SOAP in Action

## **Lesson: REST Overview**

- REpresentational State Transfer
- REST Characteristics
- REST Elements
- REST Architectural Principles
- REST and HTTP
- REST/HTTP: Representation-Oriented
- REST Design Principles
- Exercise: Working With REST

## **Module 4: Description and Discovery**

### **Lesson: WSDL**

- Describing Web Services
- WSDL 2.0/1.1 in Practice
- WSDL Namespaces
- WSDL Anatomy

### **Lesson: Discovery**

- Issues With Broadly Scoped Discovery
- UDDI Registries
- Tools That Support Discovery
- Exercise: Description and Discovery in Action

## **Module 5: Web Services in Java – JAX-WS**

### **Lesson: JAX-WS Overview**

- JAX-WS Architecture
- JAX-WS Features

- Web Service Annotations
- JAX-WS Programming Model
- JAX-WS Handlers

### **Lesson: Working with JAX-WS**

- JAX-WS Development Process
- Bottom-up Building of a Web Service
- Top-Down Building of a Web Service
- Types of JAX-WS Clients
- Exercise: Cost Service
- Exercise: Cost Service Clients
- Exercise: Membership Registration Service
- Exercise: Membership Registration Client
- Exercise: Modifying the Registration Service

### **Lesson: Handlers**

- JAX-WS and Handlers
- Handler Life Cycle
- Configuring Handlers
- Understanding SAAJ
- Connections
- Exercise: Working with Handlers
- Exercise: Checking Compliance with Handlers (optional)

### **Lesson: Working with Attachments**

- SOAP With Attachments
- Sending Binary Data
- Optimized Serialization
- WS-I Attachment Profile
- swaRef
- Enabling MTOM in JAX-WS
- Attached and In-line
- JAX-WS and swaRef

## **Module 6: JAX-RS**

### **Lesson: Designing RESTful Services**

- Effectively Designing RESTful Services
- Best Practices for Endpoint Definition
- Using Query Parameters
- Working with HTTP GET and DELETE
- Working with HTTP PUT
- Working with HTTP POST

- Best Practices for HTTP Methods
- Handling Additional Operations

### **Lesson: Introduction to JAX-RS**

- Understand some of the features of the JAX-RS framework
- Be familiar with process for implementing RESTful services
- Be able to develop and deploy a simple REST service
- How JAXB supports XML interoperability
- How to use JAXB with JAX-RS
- Exercise: Introduction to JAX-RS

### **Lesson: @Path: URI Matching**

- JAX-RS mechanisms for mapping URIs to resource
- How to use @Path expressions to map URIs
- How to use @Path expressions to extract values from URIs
- Exercise: URI Matching JAX-RS

### **Lesson: JAX-RS Content Negotiation**

- JAX-RS built-in content handler
- How to inject content handlers into service
- Basics of working with JSO
- Capabilities for HTTP content negotiation
- JAX-RS mechanisms for supporting content negotiation
- Exercise: Content Negotiation

### **Lesson: JAX-RS Request and Response**

- How information about the request can be injected into the resource
- The different injection targets of the resource
- How to build a 'complex' response
- How to respond to a HTTP POST
- The JAX-RS exception hierarchy
- How exceptions can be mapped to response codes
- Exercise: The Reservation Service

### **Lesson: JAX-RS Client API**

- Be able to invoke a JAX-RS service using the client API
- Understand the purpose of the WebTarget object
- Be able to dynamically resolve path template values
- Understand some of the return codes
- Be able to filter client request and server responses
- Exercise: JAX-RS Clients

### **Lesson: JAX-RS Filters and Interceptors**

- Understand the purpose of JAX-RS filters
- Be able to develop interceptors
- Be able to define global filters and interceptors
- Dynamically add interceptors and filters
- Understand NameBinding
- Exercise: JAX-RS Filters and Interceptors

### **Lesson: Asynchronous JAX-RS**

- Be able to make asynchronous requests
- Develop resources to process request asynchronously
- Exercise: Asynchronous JAX-RS

## **Module 7: Security – WS-Security and Defenses**

### **Lesson: XML Signature and Encryption**

- XML Challenges
- XML Signature
- XML Signature Usage
- XML Encryption
- XML Encryption Usage

### **Lesson: WS-Security**

- Transport-Level Security
- Message-Level Security
- Web Services Security Roadmap
- WS-Security Enables Interoperability
- Networking Devices Usage

### **Lesson: Securing Untrusted Input**

- Input Data Attacks
- Protecting a Web Service
- Tenacious D
- Responding to Error State