Document Generated: 08/31/2024

Learning Style: Virtual Classroom

Provider: Java

Difficulty: Intermediate

Course Duration: 4 Days

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

About this course:

This course is designed for experienced Java developers new to JEE, who want to learn service development and essential dynamic web and skills. This course is presented by the cooperation of various leading Java EE / JEE s professionals and authors. In this course candidates will learn techniques to create and design web components, incorporating hands-on labs and all the essential concepts that will need to build working server-side apps very quickly. This course offers main JEE expertise and ability that can be utilized as the base for building a production-quality web app to a baseline.

Java Servlet is used for building web apps. Java Servlet program runs on a web server, It can create dynamic content and respond to the user demand. It can enable flexible creation of dynamic content.

Web services and Service-Oriented Architecture (SOA) represent revolutionary in distributed computing. The ideas are not entirely new, however, the acceptance of fundamental standards, such as WSDL, SOAP, XML, HTTP, and numerous WS-policies, have set the stage for SOA and XML Web services. Too much concern with services is security. Several areas of security are linked in this course, enable candidates to observe the full array of problems as well as solutions. These incorporate authorization and authentication assertions, digital signatures, encryption, and acknowledge app security problems like Injection attacks and Cross-Site Scripting.

The Java Developer can make an average salary of $90,992 per annum.

Course Objective:

After completion of this course, students have the expertise and skills need to design, maintain, build, and secure web applications. In this program, you will confront with usual web app design issues and provide the ability you will be required to resolve them, like JEE design patterns. Students will also explore a variety of web technologies and JEE like CDI, Java Naming and Directory Interface, JavaServer Pages, JavaServer Faces and learn how to utilize them. You will also study the abilities of Servlets, their benefits, session management, and servlet architecture. This course also provides you the knowledge of how to utilize custom tags, application models, deployment, managing resources, techniques to develop a capable and robust web app by utilizing servlets and other components.

This course aimed to provide you knowledge and expertise of the basic technologies utilize in web services. This knowledge is crucial to enable you to troubleshoot, diagnose, tune, and execute other lifecycle activities.

This course covered the following areas.

· JSF (JavaServer Faces)

· JSTL, EL, JSP (JavaServer Pages)

· Servlets

· JEE Application Architecture

After completion of this course students can:

· Understand and read a Web Services Description Language document

· Analyze, operate and respond to a SOAP message

· Discuss and understand Web Services and the main technologies involved

· Explain the concepts behind Representation State Transfer and deployment a REST-based web service

· Build, design, and implement real-world JEE Web Services

· Deploy handlers to insert cross-cutting solutions for auditing, logging, security, and other requirements

· Write Java components

Audience:

This training course is intended for experienced Java developers, new to JEE, who want to enhance their skills and expertise in JEE web development.

Prerequisite:

Candidates are required to have extensive working knowledge in developing fundamental 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

Credly Badge:

Display your Completion Badge And Get The Recognition You Deserve.

Add a completion and readiness badge to your Linkedin profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

Find Out More or See List Of Badges