

Java RESTful Services Essentials (TT7305)

Modality: Virtual Classroom

Duration: 2 Days

About this course:

Geared for experienced developers, Java RESTful Service Essentials is a two day, lab-intensive services training course that introduces developers to the core concepts, principles, and Java implementations for RESTful services.

Today's development environments are increasingly dominated by sophisticated tooling that makes the implementation of RESTful services less arduous. The proper design of these services is far more complex and demanding.

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

Course Objective:

Java RESTful Service Essentials focuses on providing an understanding of the fundamental principles and technologies that are used in building these services. This understanding is critical to being able to diagnose, troubleshoot, tune, and perform other lifecycle activities.

Working within in an engaging, hands-on learning environment, guided by our expert team, attendees will learn to:

- Understand and apply the basic concepts of REST
- Understand and intelligently discuss the similarities and differences between RESTful services and SOAP-based services
- Appreciate the security concerns associated with RESTful services
- Design, develop, and deploy real-world RESTful Services
 - Effectively define and design endpoints
 - Work with query parameters
 - Determine the best format to use for exchanging data
 - Understand the principles associated with HTTP methods and how to stay congruent to them
 - Extend the semantics of the RESTful service beyond the HTTP methods
- Implement RESTful services using JAX-RS to
 - Handle various HTTP methods
 - Use different approaches for parameters
 - Deal with content negotiation
 - Work with different data formats including XML and JSON
 - Handle exceptions
- Implement RESTful Java clients
 - Develop JAX-RS Java clients using the JAX-RS 2.0 Client API
 - Manage different Web targets

- Deal with content negotiation on the client
 - Work with different data formats including XML and JSON
 - Properly process server responses
- Implement JAX-RS Filters and Interceptors
 - Intercept and manipulate service requests and responses
 - Dynamically register interceptor to resources
 - Understand the concept of NameBinding
- Discover asynchronous JAX-RS processing
 - Implement an asynchronous JAX-RS service endpoint
 - Register response listeners

Audience:

This is an intermediate level training course, designed for experienced Java developers and architects who need to identify, design, and implement web services. We will explore and apply the terminology, the specification, the processes and technologies specific to web services.

Prerequisite:

Students should have 1-2 years of working knowledge with Servlets and Spring, and should be familiar with XML and JSON

Course Outline:

Module 1: Working with REST

Lesson: Overview of REST

- REpresentational State Transfer
- REST Characteristics
- REST Elements
- REST Architectural Principles
- REST and HTTP
- REST/HTTP: Representation-Oriented
- REST Design Principles
- Tutorial: Working with Eclipse Neon (JEE Version) and Tomcat 8.5

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

Module 2: JAX-RS

Lesson: Introduction to JAX-RS

- Understand some of the features of the JAX-RS framework
- Be familiar with process for implementing RESTful services
- 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

- Invoke a JAX-RS service using the client API
- Understand the purpose of the WebTarget object
- Dynamically resolve path template values
- Understand some of the return codes
- Filter client request and server responses
- Exercise: JAX-RS Clients

Lesson: JAX-RS Filters and Interceptors

- Understand the purpose of JAX-RS filters
- Develop interceptors
- Define global filters and interceptors
- Dynamically add interceptors and filters
- Understand NameBinding
- Exercise: JAX-RS Filters and Interceptors

Lesson: Asynchronous JAX-RS

- Make asynchronous requests
- Develop resources to process request asynchronously
- Exercise: Asynchronous JAX-RS

Module 3: Security and RESTful Services (Optional)

Lesson: Securing Untrusted Input

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

Lesson: Defending RESTful Services

- How Attackers See REST
- Factors that Increase Attack Surface
- Bridging and its Potential Problems
- Dangerous Developer Assumptions
- Three Basic Tenets for Safe REST