

Developing RESTful Services with REST/JAX-RS

Course Summary

Description

This course provides in-depth coverage of REST principles and building RESTful services using the JAX-RS 2.0 API. It includes a brief overview of SOAP-based services and WSDL, to provide a contrast to RESTful services.

To really understand REST, we need to remind (teach) students about the full capabilities of HTTP, since RESTful service design is centered on the innate principles and capabilities of the HTTP protocol. Stated differently, RESTful services should use HTTP the way it was originally intended, not just as GET/POST plumbing for request/response data. You will emerge from the course with a full understanding of the HTTP protocol and its vision of the "original web," and how these ideas are implemented in RESTful applications.

We'll show you how to handle the most common types of request input, culminating in the transmission of HTTP entities in both XML and JSON format, in line with the original intention of HTTP. You'll also learn how to create appropriate responses given a variety of inputs and conditions, according to the guidelines and principles in the HTTP standard.

On the client side, we look at Ajax-JavaScript clients, as well as standalone Java clients using the JAX-RS 2.0 Client API.

Your daily personal use of the web is increasingly RESTful, and this course will show you how to expose your backend business systems using REST and the innate API and capabilities of HTTP. We also take cues from the masters such as Google, examining some of their public services and using them as a backdrop in building our own.

The course is very hands-on with frequent labs, in which the student can see these ideas implemented in software.

Attendees must be experienced in Java and preferably familiar with XML, Java web applications, and other Java EE standards. Familiarity with Ajax, JavaScript, and JSON is also beneficial, but not required.

All labs are done with the Eclipse IDE Java EE version and the WildFly Application Server, and the lab instructions include detailed directions for using them. Other IDEs and servers available on request.

Objectives

After taking this course, students will be able to:

- Understand the purpose and role of web services in general, and how they are architected to expose business systems and processes over the web
- Understand the concepts and principles of REST and HTTP applications
- Expanded knowledge of HTTP, including its full set of methods and their intended uses, important headers, response codes, and content types

Developing RESTful Services with REST/JAX-RS

Course Summary

- Understand REST APIs, including resource identifiers and the URI namespace, resources and subresources, and WADL
- Understand configuration, deployment, and the runtime environment, including per-request and singleton objects, options for dependency injection, etc.
- Understand how HTTP requests get dispatched to service methods
- Understand content negotiation and the importance of Accept and Content-Type headers, and how they impact method dispatching
- Bind request inputs to method parameters, including path parameters, query parameters, and headers
- Exchange business data by communicating in HTTP entities in both XML and JSON format
- Handle errors using Java exceptions and appropriate HTTP response codes
- Learn how to integrate JAX-RS services with other Java EE technologies like servlets, EJB, and CDI, and how JAX-RS fits into the larger Java EE landscape
- Write browser clients using Ajax-JavaScript
- Write Java clients using the JAX-RS 2.0 Client API, including standalone clients and server-side components invoking remote services
- Learn how to secure RESTful resources

Topics

- Web Services Overview
- Introduction to REST
- Introduction to JAX-RS
- Resources and Requests
- HTTP Entities
- Responses
- Java Client API
- Integration with Java EE
- Security

Audience

This course is designed for those wanting to learn REST principles and build RESTful services using the JAX-RS 2.0 API.

Prerequisites

Before taking this course, attendees must be experienced in Java and preferably familiar with XML, Java web applications, and other Java EE standards. Familiarity with Ajax, JavaScript, and JSON is also beneficial, but not required.

Duration

Three days

Developing RESTful Services with REST/JAX-RS

Course Outline

- I. Web Services Overview**
 - A. Definition
 - 1. Legacy Systems
 - 2. Benefits of Web Services
 - 3. Architecture
 - B. Standards and Portability
 - 1. XML and Related Standards
 - 2. JSON
 - 3. HTTP
 - C. SOAP-Based Services
 - 1. Overview
 - 2. SOAP Messages, Requests, and Responses
 - 3. WSDL
 - 4. Java APIs and Programming Models
- II. Introduction to REST**
 - A. Overview and Principles
 - 1. REST Characteristics
 - 2. Resources and Operations
 - 3. REST Principles
 - B. Requests and Responses
 - 1. REST APIs
 - 2. URI Templates
 - 3. GET, POST, PUT, DELETE
 - 4. Safe and Idempotent Methods
 - C. Comparison of REST and SOAP
- III. Introduction to JAX-RS**
 - A. APIs and Implementations
 - 1. JAX-RS Overview, Annotations
 - 2. JAX-RS Implementations
 - B. Runtime Environment
 - 1. Application Server, Servlet-Only Container
 - 2. Architectural and Implementation Perspectives
 - 3. Configuring the Application
 - C. Applications, Resources, and Providers
 - 1. JAX-RS Applications
 - 2. Resource Classes and @Path
 - 3. Provider Classes and @Provider
 - 4. Default Lifecycles
 - 5. The Application Class and rest-path
- D. Ajax-JavaScript Clients**
 - 1. Overview
 - 2. Classic vs. Ajax Interactions
 - 3. Working with Ajax-JavaScript
- IV. Resources and Requests**
 - A. Resources and Subresources
 - 1. Root Resource Classes, Resource Methods, Subresource Methods
 - 2. @GET, @POST, @PUT, @DELETE
 - 3. Subresource Locators
 - 4. Naming Conventions and Rules
 - 5. Dispatching Requests to Methods
 - B. Binding Request Data
 - 1. Request Data Injection and Conversion
 - 2. Default Values
 - 3. Fields vs. Method Parameters
 - C. Context-Based Injection
 - 1. Injection via @Context
 - 2. Context-Injectable Types
 - 3. Context Injection from the Web Container
 - 4. Fields vs. Method Parameters
- V. HTTP Entities**
 - A. Complex Content and Entities
 - 1. Working with Complex Content
 - 2. @Consumes and @Produces
 - 3. Content Negotiation
 - 4. Standard Entity Providers
 - B. Working with JSON
 - 1. Returning Data as JSON
 - 2. Working with JSON in JavaScript
 - 3. Processing JSON Responses
 - C. Working with XML
 - 1. JAXB and Mapping to XML
 - 2. Returning Data as XML
 - 3. Working with XML on the Client
 - 4. Customizing Content, Custom Media Types
 - 5. Working with Collections
 - D. XML vs. JSON

Developing RESTful Services with REST/JAX-RS

Course Outline (cont'd)

VI. Responses

- A. Response Class
 - 1. Return Types and HTTP Response Codes
 - 2. Appropriate Responses for HTTP Methods
 - 3. Choosing the Right Response
- B. Error Handling
 - 1. Exception Mappers
 - 2. WebApplicationException
 - 3. Response vs. Thrown Exception
 - 4. Error Responses
- C. Subresource Locators
 - 1. Motivation and Uses
 - 2. Locating the Locator
 - 3. Initializing the Subresource
- D. Binary Content
 - 1. File, InputStream, StreamingOutput
 - 2. Using StreamingOutput

VII. Java Client API

- A. Java Client - Options and Ingredients
- B. Building and Sending the Request
- C. Consuming the Response
- D. Options for the Response Data
- E. Asynchronous Requests

VIII. Integration with Java EE

- A. Integration with EJB
- B. CDI - Contexts and Dependency Injection
 - 1. Activation, Scopes, and JAX-RS Lifecycles
 - 2. Injection in CDI-Enabled JAX-RS Applications
 - 3. Enhanced Java EE Lifecycle

IX. Security

- A. Java EE Security Overview
 - 1. Security Requirements in JAX-RS
 - 2. Declarative, Role-Based Security
 - 3. Security Constraints
 - 4. Annotation-Based Security
- B. Authentication
 - 1. Configuration
 - 2. Authentication Models: Basic, Digest, Client-Cert
- C. Programmatic Security
 - 1. SecurityContext
 - 2. Client Security
- D. HTTPS