Course duration

• 5 days

Course Benefits

- Understand the advantages of the REST architecture for web services.
- Use JAX-RS to develop simple RESTful services.
- Control dispatching to service methods based on URL patterns and HTTP methods.
- Bind request values to method parameters when expressed as HTTP query parameters, form values, headers, cookies, and more.
- Manage XML and JSON content using XML Schema and JAXB -- or without JAXB using leading JAX-RS providers and Reflection-driven entity providers such as MOXy and Jackson.
- Handle error conditions by producing appropriate HTTP responses.
- Use JSR-303 validation for request parameters, headers, and entities.
- Use Java generics to implement REST API patterns for various domain classes.
- Take advantage of lifecycle and context services available to JAX-RS services.
- Organize request-handling methods into sub-resource classes to make REST APIs extensible and maintainable.
- Implement REST clients using the JAX-RS standard API.
- Build filters and interceptors to adapt service endpoint behavior.
- Develop unit tests for JAX-RS services that cover both method code and JAX-RS annotations, using the Jersey test framework.
- Be aware of security concerns for RESTful services and secure services appropriately.

Course Outline

- 1. Overview of REST and JAX-RS
 - 1. The REST Vision
 - 2. Use of HTTP
 - 3. Use of URIs
 - 4. Use of Content Types
 - 5. CRUD Operations and Business Operations
 - 6. HATEOAS and the Richardson Maturity Model
 - 7. JAX-RS
 - 8. Applications, Resources, and Providers
- 2. Configuration and Lifecycle
 - 1. The JAX-RS Application
 - 2. XML Configuration
 - 3. Annotation-Driven Configuration
 - 4. Applications

- 5. Root Resource Classes
- 6. Per-Request vs. Singleton Lifecycle
- 7. Providers
- 3. Handling Requests
 - 1. The Application Path
 - 2. The @Path Annotation
 - 3. The HTTP Method Annotations
 - 4. Sub-Resource Locators
 - 5. Annotation Inheritance and overriding
 - 6. @XXXParam Annotations
 - 7. The @DefaultValue Annotation
 - 8. Parameter Types
 - 9. Parameter Converters
- 4. Producing Responses
 - 1. Supported Return Types
 - 2. The Response Class
 - 3. Response Entities
 - 4. Binary Content
 - 5. Delivering a File
- 5. Entity Translation
 - 1. Entity Parameter and Return Type
 - 2. Entity Providers
 - 3. @Consumes and @Produces Annotations
 - 4. Built-In Entity Providers
 - 5. Custom Entity Providers
- 6. Working with XML and JSON
 - 1. The JAXB Entity Provider
 - 2. Driving XML Representations from Schema
 - 3. Driving JSON Representations with JAXB
 - 4. JSON Without JAXB: Jackson, MOXy, etc.
 - 5. CRUD Patterns
 - 6. Error Handling
 - 7. Sub-Resources
- 7. Dependency Injection
 - 1. The @Context Annotation
 - 2. Injectable Types
 - 3. The Application Subclass
 - 4. Servlet Configuration and Context
 - 5. Impact of Lifecycle Policies
 - 6. Context Providers
 - 7. Using CDI
- 8. Validation and Error Handling
 - 1. Using Response
 - 2. Throwing WebApplicationException
 - 3. Exception Mapping Providers
 - 4. Selection of Exception Mappers
 - 5. Java EE Bean Validation

- 6. Constraint Annotations
- 7. Support for JSR-303
- 8. Annotating Method Parameters
- 9. Annotating Entity Classes
- 10. Error Reporting
- 9. Generic Services
 - 1. Generic Entities
 - 2. Generic Entity Providers
 - 3. ParameterizedType
 - 4. Reflection-Driven Entity Providers
 - 5. Annotation Inheritance
 - 6. CRUD Patterns, II
 - 7. Serialization, Recursion, and Scope
 - 8. Dynamic Sub-Resources
- 10. Working with Databases
 - 1. Persistence Services
 - 2. The Java Persistence API
 - 3. JPA Support for JSR-303
 - 4. Handling IDs and Keys
 - 5. Cascading
 - 6. Caching
 - 7. Error Handling
 - 8. Hypermedia Challenges
- 11. Sub-Resources
 - 1. Significance of Sub-Resources
 - 2. Exposing Sub-Objects
 - 3. Exposing Collections
 - 4. Multiple Paths to Resources
 - 5. Exposing Actions
 - 6. Using Sub-Resource Locators
 - 7. Collection vs. Instance Services
- 12. The Client API
 - 1. The Builder Pattern
 - 2. Client
 - 3. WebTarget
 - 4. Invocation
 - 5. Basic Usage
 - 6. Managing Content Types and Entities
 - 7. Error Handling
 - 8. Registering Providers
 - 9. The Service Locator Pattern
 - 10. Generic Clients
- 13. Filters and Interceptors
 - 1. The Filter Interfaces
 - 2. Processing Pattern
 - 3. The Request and Response Context Interfaces
 - 4. Aborting a Request

- 5. The Interceptor Interfaces
- 6. Adaptive Streams
- 7. Filters on the Client Side
- 8. Interceptor Strategy for Hypermedia
- 14. Testing
 - 1. Testing JAX-RS Services
 - 2. Unit Testing and Integration Testing
 - 3. Mocking the Container
 - 4. The Jersey Test Framework
 - 5. Test Configuration
 - 6. Mocking Dependencies
 - 7. Testing JAX-RS Clients
 - 8. Mocking Services
- 15. Security
 - 1. Concerns for RESTful Services
 - 2. Authentication and Authorization
 - 3. HTTP BASIC and DIGEST
 - 4. HTTPS
 - 5. Programmatic Security
 - 6. SQL Injection
 - 7. Cross-Site Request Forgery
 - 8. Message-Level Security
 - 9. HMACs

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Java class:

- Strong Java programming skills are essential
- Experience with other Java EE standards, especially servlets and JSP, will be very helpful in class, but is not strictly required.