



Developing Web Services Using Java Technology

Duration: 5 days

Course description:

The Developing Web Services Using Java Technology course provides business component and client developers with the information they need to create, implement, and deploy web services and web service clients using Java technology components and the Java Platform. Students will understand and appreciate web services as a realization of Service Oriented Architecture (SOA), and how to use the Java Platform, Enterprise Edition (Java EE) technology design patterns and best practices when designing web services.

Course topics:

Introduction to Web Services

- Explore the need for web services
- Define a web service and describe the motivation behind developing and using web services in business software
- Describe the characteristics of a web service
- Describe the two major approaches to developing web services
- Describe the advantages of developing web services within a JavaEE container.

JAX-WS-based Web Services

- Understand how to create web services using JAX-WS
- Understand how to deploy web services providers using JavaSE
- Understand how to create and deploy simple web services clients using JavaSE.

SOAP and WSDL

- Understand the basic structure of a SOAP message, and how it is encapsulated by transports
- Understand how WSDL defines a web service, including its message representation and transport mechanism



- Understand the different styles of SOAP messages that a web service can use, and their trade-offs
- Customize a web service to control the style of SOAP message that that web service will use

JAX-WS and JavaEE

- Understand how to deploy POJO web services to a web container
- Understand how to define a web service in terms of an Enterprise Java Bean
- Understand how to deploy an EJB web service to an EJB container
- Describe the benefits associated with implementing a web service as an EJB

Implementing More Complex Web Services Using JAX-WS

- Apply JAXB to pass complex objects to and from a web service
- Understand how to map Java exceptions from a web service endpoint to SOAP faults
- Understand how to inject attributes into JAX-WS web service endpoints
- Describe JAX-WS artifacts that can be injected and how to use them

AX-WS Web Service Clients

- Understand how to create web service clients using JAX-WS
- Understand how to create web service clients using JAX-WS that support asynchronous interactions

Introduction to RESTful Web Services

- Understand what RESTful Web Services are
- Understand the five principles behind RESTful Web Services
- Understand the advantages and disadvantages of a RESTful approach.

JAX-RS-based Web Services

- Understand how the five principles of RESTful web services map to JAX-RS constructs
- Understand how to implement REST web services using JAX-RS
- Understand how to deploy REST web services using Jersey, an implementation of JAX-RS



JAX-RS-based Web Service Clients

- Understand how to create JAX-RS clients using URL and HttpURLConnection.
- Understand how to create JAX-RS clients using the Jersey Client API.

JAX-RS and JavaEE

- Understand how to deploy POJO web services to a web container
- Understand how to define a web service in terms of an Enterprise Java Bean
- Understand how to deploy an EJB web service to an EJB container
- Describe the benefits associated with implementing a web service as an EJB

Implementing More Complex Web Services Using JAX-RS and Jersey

- Understand how to produce and consume custom types.
- Define JAX-RS web services that provide results by linking to other resources.
- Understand how to manage exceptions. • Define JAX-RS web services in terms of resources and sub-resources.
- Understand the different scopes defined by JAX-RS for web services endpoints.

Trade-Offs Associated with the Java Web Services APIs

- Understand the trade-offs involved in the choice to implement a web service using either JAX-WS or JAX-RS technology.

Web Services Design Patterns

- Describe web services-based design patterns
- Describe web services-based deployment patterns

Best Practices and Design Patterns for Use with JAX-WS

- Describe JAX-WS-specific design patterns
- Recognize and apply best practices associated with implementing web services using JAX-WS

Best Practices and Design Patterns for Use with JAX-RS

- Describe JAX-RS-specific design patterns
- Recognize and apply best practices associated with implementing web services using JAX-RS