Develop Web Services and Microservices with Java LVC

JAVA

DURATION	MODULES	COURSE CODE
3 Days	**9 Lectures**	**—**

Course Overview

Develop Web Services and Microservices with Java LVC

What You Will Learn

Develop Web Services and Microservice Applications with Java

- Course Goals
- Audience
- Course Practices

Introduction to Web Services

- Objectives
- What is a Service?
- Web Services Types
- Business and System Integration Challenges
- Building a Portfolio of Services
- Point-to-Point Service Interactions
- Service Oriented Architecture
- SOAP WebServices
- SOAP Message
- Web Service Interaction Patterns
- REST Service Conventions and Resources
- REST Communication Model
- Designing Services
- Java Service Implementation Options
- Java Service Deployment Platforms
- Microservices
- Microservices in Context
- Virtualization and Deployment Containers

- Service Versioning
- Summary
- Practices

Handle XML Using JAXB API

- Objectives
- Agenda
- eXtensible Markup Language
- Document Object Model (DOM)
- What is a Well-formed XML Document?
- XML Namespaces
- Agenda
- Parsing and Validating XML Documents
- Use JAXP API to Parse XML Documents
- Agenda
- Accessing XML Document Content Using XPath Expressions
- Transform XML Data
- XSL Transformation
- XQuery Processing
- Agenda
- DTD Validation
- Create and Reference XML Schemas
- Components of an XML Schema
- Built-in XML Schema Data Types
- Describe a Simple Element
- Describe a Complex Element
- Describe Attributes and Attribute Groups
- Agenda
- JAXB API
- JAXB Automation
- JAXB Annotations: Mapping Elements and Attributes
- JAXB Annotations: Mapping Values and Restrictions
- Converting Values with XMLAdapter
- Marshall and Unmarshall XML with JAXB
- Summary
- Practices

Create SOAP Services Using JAX-WS API

- Objectives
- Agenda
- JAX-WS Implementation Options
- Automating JAX-WS Development
- Agenda
- WSDL Structure
- WSDLs, Schemas and XML Namespaces

- WSDL Messages, PortTypes, and Operations
- WSDL Bindings and Services
- SOAP Message Format Style and Use
- RPC/Literal Message Format
- Document/Literal/Wrapped Message Format
- Document/Literal/Bare Message Format
- Agenda
- Implementing JAX-WS Services
- Mapping Server Endpoint Implementation Class
- Mapping Service Endpoint Interface
- Mapping Service Operations
- Mapping Parameters, Return Types and Exceptions
- Specify SOAP Service Binding Mode
- Specify Transport Protocol Bindings
- Mapping Service Provider
- Handle Requests Using SOAP Messages
- Produce Responses Using SOAP Messages
- Produce and Consume SOAP Attachments
- Using WebServiceContext and MessageContext Objects
- Summary
- Practices

Invoke SOAP Services Using JAX-WS API

- Objectives
- JAX-WS Client Implementation Options
- Generate Proxy Client
- Generated Proxy Client
- Invoke a Service Using a Proxy Client
- Create a Dispatch Client
- Produce a SOAP Request Message with a Dispatch Client
- Process a SOAP Response Message with a Dispatch Client
- Summary
- Practices

Handle JSON Using JSON-P and JSON-B APIs

- Objectives
- What is JSON?
- JSON-P API
- Parse JSON Data Using Object Model API
- Produce JSON Data Using Object Model API
- Parse JSON Data Using Streaming API
- Generate JSON Data Using Streaming API
- JSON-B API
- JSON-B Configuration
- JSON-B Annotations

- Defining and Validating JSON Structures
- Summary
- Practices

Build REST Services Using JAX-RS API

- Objectives
- REST Service Conventions and Resources
- REST Services Example
- HTTP Requests
- HTTP Responses
- Mapping REST Application to URL
- Register REST Resources
- Define REST Resources
- Using HTTP Methods in REST Communications
- Mapping REST Resource Operations
- Defining Parameters
- Validating Values
- Producing Errors
- Automatic Marshalling and Unmarshalling Messages
- Produce Messages Using Response Object
- Define and Document REST Service Interfaces
- Oracle Apiary
- Summary
- Practices

Invoke REST Services

- Objectives
- REST Client Responsibilities
- Simple HTTP Client
- Asynchronous HTTP Client Capabilities
- JAX-RS Client
- Asynchronous JAX-RS Client Capabilities I
- Asynchronous JAX-RS Client Capabilities II
- Reactive JAX-RS Client I
- Reactive JAX-RS Client II
- Microprofile JAX-RS Client
- AJAX JavaScript Client
- Modern JavaScript Client
- Summary
- Practices
- 9.a WebServices Security and Policies
- Objectives 9.a-2
- Web Services Non-Functional Requirements 9.a-3
- Configure Security Infrastructure 9.a-4
- Java EE Security API 9.a-5

- Java EE/MP Authentication and Authorization Flow 9.a-6
- Configure Identity Store 9.a-7
- Configure Authentication Mechanism 9.a-8
- Configure Role-Based Authorization 9.a-9
- Programmatic Security Enforcement 9.a-10
- WebService Policy Enforcement 9.a-11
- WS-Security 9.a-12
- Attaching WebLogic Policies 9.a-13
- Summary 9.a-14
- Practices 9.a-15
- 9.b Secure REST Services using OAuth
- Objectives 9.b-2
- Cross-Application Security Concerns 9.b-3
- OAuth 2.0 Participants 9.b-4
- OAuth 2.0 Access Control 9.b-6
- Security Tokens 9.b-7
- Obtaining Access Tokens Using OAuth 2.0 9.b-8
- Register Client Application with OAuth Authorization Server 9.b-10
- Register Resource Application with OAuth Authorization Server 9.b-11
- Grant Client Application Access to Resource Scopes 9.b-12
- Use Authorisation Token to Access a Resource 9.b-13
- Summary 9.b-14

Service Architecture

- Objectives
- Service Integration Patterns
- Agenda
- Simple Frontend Integration
- Frontend Intermediary
- Backend Intermediary
- Hybrid Integration
- Agenda
- Monolith vs Microservice
- Monolith Characteristics
- Microservice Characteristics
- Microservices Availability and Data Integrity
- "Microlith" Architecture
- Summary