

Building Rest And Soap Web Services With Java

Course: 00031

Filter: **Beginner**

Duration: **20 hours**

Category:: **Java**

Price: **3000,00 €**

About Course

This course provides experienced Java programmers with the skills to write new SOAP and REST web services, and access existing services. With this training, you also learn how to secure web services using both Java-specific and language-independent security technologies, and how the Java APIs — JAX-WS and JAX-RS — deliver a set of powerful tools for developing a Service-Oriented Architecture (SOA).

What you'll learn

- Develop, deploy, and monitor SOAP and RESTful web services and web service clients with JAX-WS and JAX-RS
- Implement a Service-Oriented Architecture (SOA) using web services
- Create and deploy WSDL-first and code-first SOAP web services
- Build secure SOAP and REST clients

Pre-requisites

- Knowledge at the level of: Course 471, Java Programming Introduction
- Three months of experience writing Java programs, including an understanding of Java classes and the inheritance model
- You should be able to:
 - o Create and run a complete stand-alone Java application
 - o Use the standard flow control constructs (if/then/else/while/for)
 - o Instantiate Java objects and call instance methods
 - o Create classes that implement interfaces

Curriculum

Module 1: Interoperable applications with SOA

- Designing an SOA integration architecture
- Implementing SOAs with web services

Module 2: Java standard APIs for web services

- Building SOAP-based services with JAX-WS
- Developing RESTful services with JAX-RS

Module 3: XML essentials

- XML namespaces
- Describing XML with schema

Module 4: Structure of SOAP messages

- Role of SOAP in web services
- Operations, messages and faults

Module 5: Anatomy of a WSDL document

- Defining the interfaces of a web service
- Specifying implementation

Module 6: Exposing plain old Java objects (POJOs) as web services

- Applying JAX-WS annotations to POJOs
- Configuring and deploying a web service implementation

Module 7: Implementing SOAP clients in Java

- Generating client-side artifacts from WSDL
- Modifying client code to permit message monitoring

- Authenticating and authorizing clients

Module 8: Augmenting SOAP-based services

- Incorporating best practices for web services
- Implementing policies for security, reliability and message optimization

Module 9: Contract-first service development

- Comparing contract-first and code-first services
- Creating portable Java artifacts

Module 10: Importing a WSDL document

- Building interoperable applications by conforming to Web Services Interoperability (WS-I) standards
- Implementing a web service endpoint using JAX-WS

Module 11: Introduction to REST (Representational State Transfer)

- Describing the REST architectural style
- Comparing SOAP and RESTful web services

Module 12: Developing RESTful web services using JAX-RS

- Adding JAX-RS annotations to a POJO
- Configuring result types using HTTP request headers
- Deploying a JAX-WS service

Module 13: Customizing a RESTful Service Implementation

- Interacting with request URLs
- Mapping URLs to Java classes and methods
- Binding URL components to method arguments

Module 14: Handling JavaScript Object Notation (JSON)

- Applying cases for JSON with RESTful services

- Interpreting a message formatted as JSON

Module 15: Mapping Java to JSON

- Controlling JSON generation with JAXB annotations
- Defining a JSON return type from a Java method

Module 16: Dispatching REST messages using JAX-RS

- Building the client's request
- Handling the service's response code and exceptions

Module 17: Marshalling JavaBeans with JAXB

- Mapping XML with Java API for XML Binding (JAXB)
- Converting Java arguments with JAXB and JAX-RS

Module 18: Processing asynchronous messages

- Implementing server-push with JAX-RS
- Delivering asynchronous client requests

Module 19: Configuring a secure web service

- Configuring HTTP Basic Authentication
- Establishing secure message transmission with SSL/TLS

Module 20: Authenticating and authorizing clients

- Controlling access to web services and methods
- Providing authentication information to web services