Updated on 19/05/2025

Sign up

Apache Camel training: mastering integration routes

2 days (14 hours)

Presentation

Our Apache Camel training course will enable you to design and deploy robust, modular integration flows, in a spirit of agility and modernity. You'll learn how to create robust integration flows, orchestrate exchanges between services and automate data processing in distributed environments.

Our training program covers all the essential basics, right through to advanced functionalities. You'll be able to design complex integration routes, transform and route messages in real time, and easily connect applications.

At the end of this course, you'll be able to model integration scenarios using EIPs, handle errors with finesse, and deploy your flows on Kubernetes using Camel K, for smooth, scalable cloud-native integration.

As with all our training courses, this one is delivered with the latest version of Apache Camel 4.8.

Objectives

- Install and configure Apache Camel
- Understand EIP architecture and principles
- Master the design, routing and transformation of data flows between applications
- Integrate Apache Camel with third-party systems: Kafka, REST APIs, databases or JMS
- Deploy and execute integration routes in a cloud-native environment with Camel K

Target audience

- Developers

- DevOps engineers
- Technical architects

Prerequisites

- Notions of integration architecture
- Basic knowledge of Java

Apache Camel training program

Introduction to Apache Camel

- Presentation of Apache Camel
- What is an EIP
- Apache alternatives
- The Camel route concept
- Using the Java DSL to write a readable route
- Starting a Camel project locally and running a first route

Mastering the main EIPs

- Content-based routing of messages
- Splitter and Aggregator
- Enriching a message with external data
- Multicast and Recipient List
- Monitoring and error handling with WireTap and Dead Letter Channel

Connections with external systems

- Creating and consuming a REST endpoint with Camel
- Read and write from a database with jdbc or jpa
- Manage JSON, XML and CSV formats with Jackson and XStream
- Produce JMS messages with ActiveMQ or RabbitMQ
- Using Apache Kafka

Testing, logs, transformation and robustness

- Use CamelTestSupport to test routes
- Trace and log exchanges with
    - log
    - trace
    - debugging
- Redelivery, DLQ
- Transforming with XSLT

Introduction to Camel K and Kubernetes

- Key Camel K concepts: serverless and rapid integration
- Installing and configuring Kamel CLI
- Deploying a Java or Yaml route on Kubernetes
- Using ConfigMaps to inject parameters

Further information: Camel Quarkus

- Introducing Camel Quarkus and its relevance to microservices
- Creating a Quarkus project with Camel extensions
- Startup time reduction and native execution with GraalVM
- Deploying Camel microservices on Kubernetes or OpenShift
- Camel standalone, Camel K and Camel Quarkus

Monitoring and security

- Enabling Camel metrics with Prometheus / Micrometer
- Integration with OpenTelemetry and visualization in Grafana
- Protecting endpoints
  - SSL
  - authentication
  - headers
- Best practices for performance, modularity and resilience
- Structuring a professional integration project

Further information

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced IT technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the forthcoming course, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical training: 60% hands-on, 40% theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Certification

A certificate will be awarded to each trainee who has completed the entire course.