

Module Title : AD421- Camel Integration and Development with Red Hat Fuse on OpenShift

Duration : 5 days

Overview

Camel Integration and Development with Red Hat Fuse on OpenShift (AD421) is a hands-on, lab-based course that gives Java™ developers and architects an understanding of Apache Camel and the enhancements and tools Red Hat offers in support of Camel development.

This course is based on Red Hat® Enterprise Linux® 7.5, Red Hat® OpenShift® Container Platform 3.9, Red Hat® Fuse 7.1, and Apache Camel 2.21.

Camel and Red Hat Fuse enable developers to create complex integrations in a simple and maintainable format. Camel development is organized around:

- Routes that define a sequence or flow of processing.
- Processors that transform, interpret, and modify messages within a Camel route.
- Components that enable the creation of endpoints that interact with the outside world for acquiring and transmitting data.

This course can assist in preparing for the [Red Hat Certified Specialist in Camel Development exam \(EX421\)](#).

Course content summary

- Create basic routes and integration solutions with Camel.
- Combine messages through aggregation.
- Develop tests with Camel test kit and mock components.
- Develop REST services with the REST DSL and Swagger.
- Deploy a route on Karaf, EAP, and Spring Boot.
- Increase route throughput with parallel processing.
- Deploy microservices with Red Hat Fuse on OpenShift.

Audience

- Java developers who need to learn how to use Camel, implement enterprise integration patterns (EIPs), and develop integration applications with Camel

- Java architects who need to understand how Apache Camel can be used in software architectures requiring integration between applications

Prerequisites

- Experience in developing and deploying Java EE 7 applications
- Experience with relational databases, JDBC, and SQL
- Experience with Java development tools such as Maven and Eclipse

Course Outline

Introduce Red Hat Fuse and Camel

Describe how Red Hat Fuse and Camel are used to integrate applications.

Create routes

Develop simple Camel routes.

Transform data

Convert messages between data formats using implicit and explicit transformation.

Create tests for routes and error handling with Camel

Develop reliable routes by developing route tests and handling errors.

Route with Java beans

Create dynamic routes in Camel using Java beans.

Implement REST services

Enable REST support on Camel with Java REST APIs.

Deploy Camel routes

Package and deploy Camel applications for deployment with Red Hat Fuse.

Implement transactions

Provide data integrity in route processing by implementing transactions.

Implement parallel processing

Improve route processing throughput using Camel parallel processing mechanisms.

Create microservices with Red Hat Fuse

Create microservices from Camel routes.

Deploy microservices with Red Hat Fuse on OpenShift

Deploy microservices based on Camel routes to an OpenShift cluster using Red Hat Fuse on OpenShift.

Perform comprehensive review

Demonstrate use of the knowledge gained in this course by coding Camel routes.

Recommended next exam or course

[Red Hat Certified Specialist in Camel Development exam \(EX421\)](#)

Outcomes

Impact on the organization

This course is intended to develop the skills needed to produce more resilient, mission-critical integration solutions with Red Hat Fuse and Camel, resulting in less development time spent maintaining and developing integration solutions.

Impact on the individual

As a result of taking this course, you will have a strong understanding of fundamental Camel concepts, commonly used Camel components and their configurations, and Camel deployment options. You will also develop a thorough grasp of enterprise integration patterns to solve integration problems.

You will be able to demonstrate these skills:

- Develop Camel routes to integrate systems such as JMS, FTP, Databases, and REST services.
- Filter and transform messages to create integration routes that are highly reliable.
- Develop tests and use mock components to thoroughly test routes.
- Create reliable routes by implementing transactional routes that prevent data loss.
- Increase message throughput by implementing parallel processing in routes.
- Deploy routes with Red Hat OpenShift Container Platform