

# **Cloud-native Integration with Red Hat Fuse and Apache Camel (AD221)**

**Modality: Virtual Classroom**

**Duration: 4 Days**

## **About This Course:**

Cloud-native Integration with Red Hat Fuse (AD221) emphasizes learning architectural patterns and implementing integration services based on Apache Camel and OpenShift. Camel and Red Hat Fuse enable developers to create complex integrations in a simple and maintainable format. You will learn how to use the most common integration components in Camel and develop, test, and deploy integration focused applications on OpenShift. This course is based on Red Hat Fuse 7.10 and OpenShift 4.

## **Course Objectives:**

- Deploy Fuse applications on Red Hat OpenShift Container Platform
- Implement REST APIs with the Camel REST DSL
- Implement unit tests, error handling, and mocks for Camel routes
- Implement Enterprise Integration Patterns (EIP) using Camel components
- Integrate Camel applications with a database
- Integrate Camel applications with Apache Kafka
- Integrate Cloud-native services using Camel K
- Consume REST services using the Camel HTTP component

## **Audience:**

This course is designed for Java developers focused on implementing integration solutions in an enterprise.

## **Prerequisites:**

- Experience with Java application development or Red Hat Application Development I: Programming in Java EE (AD183).
- Be proficient in using an IDE such as Visual Studio Code.
- Experience with Maven and version control is recommended, but not required.
- Experience with Red Hat OpenShift or Introduction to OpenShift Applications (DO101) is recommended, but not required.

## **Course Outline:**

Introduction to Red Hat Fuse and Camel

Describe the architecture of Red Hat Fuse and Camel and how they are used to integrate

applications.

Create Camel routes

Implement Camel routes and develop custom processors

Implement enterprise integration patterns

Describe the most commonly used enterprise integration patterns and implement them using Camel components.

Create tests for routes and error handling

Develop reliable routes by creating unit tests and mocks, and by handling errors.

Integrate services using asynchronous messaging

Integrate microservices using Apache Kafka and ActiveMQ (JMS)

Implement transactions

Provide data integrity in route processing by implementing transactions.

Build and consume REST services

Implement and consume REST services with Camel.

Integrate cloud-native services

Deploy cloud-native microservices based on Camel Routes and Camel K components to an OpenShift cluster