

## Camel Development with Red Hat JBoss Fuse

### Course Summary

#### Description

This course covers the following:

- Routes that define an order or flow of processing by using one or more processors and endpoints
- Processors that send, interpret, and customize messages within a Camel flow.
- Components that create endpoints, which interact with the outer world for receiving and sending messages.

#### Topics

- Camel route building and performing
- Creating integration solutions by using Camel
- Converting data format in Camel
- Using Camel and Spring beans to affect route processing
- Means of interaction with the outer world (default and custom)
- Testing Camel routes with the help of JUnit and test components
- Implementing debugging while executing a route
- Enterprise integration through Fuse Fabric
- Threading aspects of routes while increased performance
- Developing custom components of Camel

#### Audience

This course is designed for Java developers and architects who wish to learn how to use Camel, implement EIPs, and use Camel Fabric for publishing custom components.

#### Prerequisites

Before taking this course, students should have the followings skills and experience:

- Understanding the principles of Java
- Knowledge of Java developments tools (Maven and Eclipse)
- Basic knowledge of Spring Framework
- Basic knowledge and experience of Apache Maven

#### Duration

Two days

## Camel Development with Red Hat JBoss Fuse

### Course Outline

- I. Introduction to Camel**
  - A. Understanding when to use Camel, the principal components of its architecture, and the EIP implemented by it.
  - B. Concepts and terminology fundamental to Camel
- II. Developing Camel applications with Maven and Fuse IDE**
  - A. Creating Camel projects with Maven
  - B. Camel Maven archetypes
  - C. Camel configuration
- III. Converting data format in Camel**
  - A. Data format
  - B. Transforming data formats in Camel while processing a route
- IV. Using beans**
  - A. Bean component
  - B. Bean integration
  - C. Working with Spring tools to activate Camel applications
- V. Key components of Camel**
  - A. Components and Component configuration
  - B. Endpoints
  - C. Customizing a default endpoint available in Camel for supporting particular use cases
- VI. Testing Camel applications**
  - A. Testing mechanisms
  - B. Creating test cases by using the tools of Camel and Spring
- VII. Managing errors**
  - A. Error handling
  - B. Error handler
  - C. Handling errors occurred while processing a route in Camel to support retries and alternative route processing
- VIII. Threading and transactions**
  - A. Threading model
  - B. Improving performance of Camel applications by using multi-threaded routes
  - C. Transactional client
- IX. Integrating Camel and Fabric**
  - A. Using Camel Fabric to manage a massive deployment of use cases
- X. Developing custom components of Camel**
  - A. Writing components
  - B. Creating Camel components according to customers' needs
- XI. Camel review**
  - A. Reviewing what you have learned about Camel