



Camel Development with Red Hat JBoss Fuse

JB421; 5 Days, Instructor-led

Course Description

Camel development is organized around:

- Routes that define a sequence or flow of processing through one or more processors and endpoints
- Processors that massage, interpret, and modify messages within a Camel flow
- Components that enable creating endpoints that interact with the outside world for acquiring and transmitting messages

Attendees will learn the skills required to develop, implement, test, and deploy enterprise integration patterns (EIP)-based applications using Camel. This course can also help you in your preparation for the Red Hat Certificate of Expertise in Camel Development exam (EX421).

Content summary

- Camel route building and execution
- Developing integration solutions with Camel
- Transforming data format in Camel
- Using Camel and Spring beans to implement on route processing
- Components for interaction with outside world (both default and custom)
- Testing Camel routes using JUnit and test components
- Implementing error handling during route execution
- Enterprise integration via Fuse Fabric
- Threading aspects of routes for increased performance
- Developing custom Camel components

Audience

- Java developers and architects who need to learn how to use Camel, implement EIPs, and use Camel Fabric to publish custom components
- People who attended the previous version of this course (prior to August 2014) and who want to prepare for the updated Red Hat Certificate of Expertise in Camel Development exam (EX421)

Prerequisites

- Understanding of Java
- Java development tools (i.e. Maven and Eclipse)
- Basic knowledge of the Spring Framework
- Basic knowledge of and experience with Apache Maven

Course Outline

Introduction to Camel

- Understand when to use Camel, the main components in Camel's architecture, and the EIP implemented by Camel

Developing Camel applications using Maven and Fuse IDE

- Create Camel projects using Maven

Transforming data format in Camel

- Transform data formats in Camel during route processing

Using beans

- Work with Spring facilities to invoke Camel applications

Key Camel components

- Customize the default endpoint available in Camel to support special use cases

Testing Camel applications

- Create test cases using Camel and Spring facilities

Managing errors

- Manage errors generated during route processing in Camel to support retries and alternative route processing

Threading and transactions

- Improve performance in Camel applications using multi-threaded routes

Integrating Camel and Fabric

- Use Camel Fabric to manage large deployment use cases

Developing custom Camel components

- Create Camel components according to customers' needs

Camel review

- Review what you've learned about Camel