Red Hat Certified Specialist in Cloud-native Integration

Content

1. Create and Maintain Camel Routes

- Develop and manage routes using various methods (Java, XML).

2. Use the Java™ Language

- Implement routes and components in Java.

3. Use the CamelContext XML

- Configure routes and contexts using CamelContext XML files.

4. Exchange Messages

- Understand message structure, including:
  - **Body**: Manipulate message body.
  - **In/Out**: Manage input and output messages.
  - **Headers**: Use headers for message routing and processing.
  - **Attachments**: Handle attachments within messages.

5. Consume and Produce Files

- Work with various file formats:
  - **CSV Format**: Read and write CSV files.
  - **JSON Format**: Process JSON data.
  - **XML Format**: Handle XML files effectively.

6. Use Enterprise Integration Patterns (EIPs)

- Implement common EIPs for message processing:
  - **Content-Based Routing**: Route messages based on content.
  - **Wire Tap**: Duplicate messages for logging or monitoring.
  - **Splitter**: Split messages into multiple parts.
  - **Aggregator**: Combine multiple messages into one.
  - **Recipient List**: Send messages to multiple recipients.

7. Use Camel Components

- Utilize various Camel components for integration:
  - **File2**: Manage file transfers.
  - **FTP**: Interact with FTP servers.
  - **JPA2**: Work with databases using JPA.

- **JMS**: Integrate with JMS messaging systems.
- **Direct**: Use direct endpoint for synchronous communication.
- **Rest**: Interact with RESTful services.

8. Use Camel Test

- Implement testing using:
  - **Mock Endpoints**: Mock endpoints for testing routes.
  - **JUnit 4 Integration**: Integrate with JUnit for unit testing.
  - **Camel Test APIs**: Utilize Camel Test APIs for testing routes.

9. Dynamically Route Messages

- Implement dynamic routing strategies:
  - **Conditional Routes**: Route based on conditions.
  - **Data-Driven Routes**: Route messages based on data content.

10. Handle Exceptions

- Implement exception handling:
  - **Catch and Handle Exceptions**: Manage exceptions in routes.
  - **Use the Dead-Letter Queue**: Route failed messages to a dead-letter queue.