

Passcert

Higher Quality, better service!



Q&A

[Http://www.passcert.com](http://www.passcert.com)

We offer free update service for one year.

Exam : EX221

**Title : Red Hat Certified Specialist
in Cloud-native Integration**

Version : DEMO

1. Create a Camel route in Java DSL that reads files from /input and writes them to /output.

A. See the Explanation.

Answer: A

Explanation:

1. Create a class extending RouteBuilder.
2. Override the configure() method.
3. Use from("file:/input") and to("file:/output").

```
public class FileRoute extends RouteBuilder {  
    @Override  
    public void configure() {  
        from("file:/input?noop=true")  
        .to("file:/output");  
    }  
}
```

2. Create a Camel route using XML DSL to route messages from a JMS queue to a file directory.

A. See the Explanation.

Answer: A

Explanation:

1. Define CamelContext in camel-context.xml.
2. Use <route> tag to define routing logic.
3. Example:

```
<camelContext xmlns="http://camel.apache.org/schema/spring"> <route>  
    <from uri="jms:queue:myQueue"/>  
    <to uri="file:/var/output"/>  
</route>  
</camelContext>
```

3. In Java DSL, configure a timer route that logs "Heartbeat..." every 10 seconds.

A. See the Explanation.

Answer: A

Explanation:

```
from("timer://heartbeat?period=10000")  
    .log("Heartbeat...");
```

- The timer component fires every 10 seconds (period=10000 ms).
- .log() prints to the console.

4. In XML DSL, route a message through a bean that transforms the content to uppercase.

A. See the Explanation.

Answer: A

Explanation:

1. Define the bean in your Spring XML context:

```
<bean id="uppercaseBean" class="com.example.UppercaseProcessor"/> 2. Define the route:  
<route>
```

```
<from uri="direct:start"/>
<to uri="bean:uppercaseBean"/>
<to uri="log:result"/>
</route>
```

5. Use Java DSL to read a file, transform the content to uppercase, and send it to a log endpoint.

A. See the Explanation.

Answer: A

Explanation:

```
from("file:/input?noop=true")
    .convertBodyTo(String.class)
    .transform().simple("${body.toUpperCase()}")
    .to("log:uppercaseOutput")
```