



# MCD-LEVEL-2<sup>Q&As</sup>

MuleSoft Certified Developer - Level 2 (Mule 4)

## Pass Mulesoft MCD-LEVEL-2 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass4itsure.com/mcd-level-2.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Mulesoft Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers



**QUESTION 1**

A custom policy needs to be developed to intercept all outbound HTTP requests made by Mule applications.

Which XML element must be used to intercept outbound HTTP requests?

- A. It is not possible to intercept outgoing HTTP requests, only inbound requests
- B. http-policy:source
- C. http-policy:operation
- D. http-policy:processor

Correct Answer: D

The http-policy:processor element is used to intercept outbound HTTP requests made by Mule applications. It allows customizing the request before it is sent to the target API and modifying the response after it is received from the target API. References: <https://docs.mulesoft.com/api-manager/2.x/policy-mule4-custom-policy#policy-xml-file>

---

**QUESTION 2**

Which pattern should be used to invoke multiple HTTP APIs in parallel and roll back failed requests in sequence?

- A. A database as a transactional outbox and an Until Successful router to retry any requests
- B. A Parallel for Each scope with each HTTP request wrapped in a Try scope
- C. Scatter-Gather as central Saga orchestrator for all API request with compensating actions for failing routes
- D. VM queues as a reliability pattern with error handlers to roll back any requests

Correct Answer: C

To invoke multiple HTTP APIs in parallel and roll back failed requests in sequence, the developer should use a Scatter-Gather router as a central Saga orchestrator for all API requests with compensating actions for failing routes. A Scatter-Gather router executes multiple routes concurrently and aggregates the results. A Saga orchestrator coordinates a series of actions across different services and handles failures by executing compensating actions. Therefore, using a Scatter-Gather router as a Saga orchestrator allows invoking multiple HTTP APIs in parallel and rolling back any failed requests in sequence. References: <https://docs.mulesoft.com/mule-runtime/4.3/scatter-gather-concept>  
<https://docs.mulesoft.com/mule-runtime/4.3/saga>

---

**QUESTION 3**

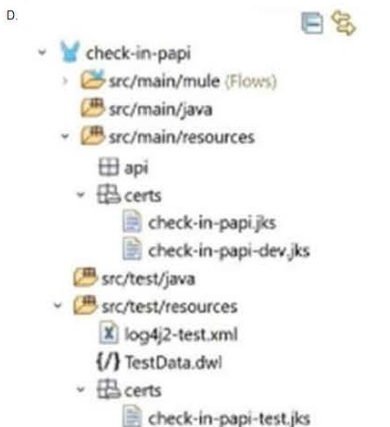
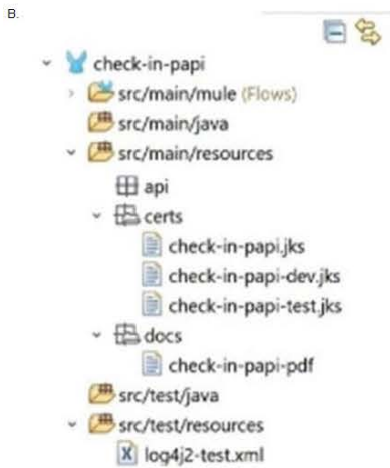
Refer to the exhibit.



```
40 <build>
41   <resources>
42     <resource>
43       <directory>src/main/resources</directory>
44       <filtering>true</filtering>
45     </resource>
46   </resources>
47   <testResources>
48     <testResource>
49       <directory>src/test/resources</directory>
50       <filtering>true</filtering>
51     </testResource>
52     <testResource>
53       <directory>src/test/funmon</directory>
54       <filtering>true</filtering>
55       <targetPath>funmon</targetPath>
56     </testResource>
57   </testResources>
58   <pluginManagement>
59     <plugins>
60       <plugin>
61         <groupId>org.apache.maven.plugins</groupId>
62         <artifactId>maven-resources-plugin</artifactId>
63         <configuration> .
64           <nonFilteredFileExtensions>
65             <nonFilteredFileExtension>p12</nonFilteredFileExtension>
66             <nonFilteredFileExtension>crt</nonFilteredFileExtension>
67             <nonFilteredFileExtension>pem</nonFilteredFileExtension>
68           </nonFilteredFileExtensions>
69         </configuration>
70       </plugin>
```

A Mule application pom.xml configures the Maven Resources plugin to exclude parsing binary files in the project's src/main/resources/certs directory.

Which configuration of this plugin achieves a successful build?





- A. Option A
- B. Option B
- C. Option C
- D. Option D

Correct Answer: C

To configure the Maven Resources plugin to exclude parsing binary files in the project's src/main/resources/certs directory, option C should be used. This option specifies that any files with .cer or .jks extensions under the certs directory should be excluded from filtering. Filtering is a process of replacing placeholders with actual values in resource files during the build process. Binary files should not be filtered because they may become corrupted or unusable.

References: <https://maven.apache.org/plugins/maven-resources-plugin/examples/filter.html>

<https://maven.apache.org/plugins/maven-resources-plugin/examples/include-exclude.html>

---

#### QUESTION 4

Which properties are mandatory on the HTTP Connector configuration in order to use the OAuth 2.0 Authorization Code grant type for authentication?

- A. External callback URL, access token URL, client ID response access token
- B. Token URL, authorization URL, client ID, client secret local callback URL
- C. External callback URL, access token URL, client ID, response refresh token
- D. External callback URL, access token URL, local authorization URL, authorization URL, client ID, client secret

Correct Answer: B

To use the OAuth 2.0 Authorization Code grant type for authentication, the HTTP Connector configuration requires the following properties: token URL, authorization URL, client ID, client secret, and local callback URL. The token URL is the endpoint of the authorization server that provides access tokens. The authorization URL is the endpoint of the authorization server that initiates the user consent flow. The client ID and client secret are the credentials of the Mule application registered with the authorization server. The local callback URL is the endpoint of the Mule application that receives the authorization code from the authorization server. References: <https://docs.mulesoft.com/http-connector/1.6/http-authentication#oauth-2-0>

---

#### QUESTION 5

A company has been using CI/CD. Its developers use Maven to handle build and deployment activities.

What is the correct sequence of activities that takes place during the Maven build and deployment?

- A. Initialize, validate, compute, test, package, verify, install, deploy
- B. Validate, initialize, compile, package, test, install, verify, verify, deploy
- C. Validate, initialize, compile, test package, verify, install, deploy
- D. Validation, initialize, compile, test, package, install verify, deploy



Correct Answer: C

The correct sequence of activities that takes place during the Maven build and deployment is validate, initialize, compile, test package, verify, install, deploy. These are Maven lifecycle phases that define a sequence of goals to execute during a build process. Each phase represents a stage in the build lifecycle and can have zero or more goals bound to it.

References:<https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>

[MCD-LEVEL-2 Study Guide](#)

[MCD-LEVEL-2 Exam  
Questions](#)

[MCD-LEVEL-2 Braindumps](#)