



MCD-LEVEL-2^{Q&As}

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 scatter-gather router is configured with four routes:Route A, B, C and D.

Route C false.

- A. `Error,errorMesage.payload.results [2\\]`
- B. `Payload failures[2\\]`
- C. `Error,errorMessage,payload.failures[2\\]`
- D. `Payload [2\\]`

Correct Answer: C

The result of accessing route C failure is `Error,errorMessage,payload.failures[2\\]`. This is because a scatter-gather router returns an aggregated message that contains an array of results from each route and an array of failures from each route. The failures array contains error objects with information about each failed route execution. To access route C failure, which is the third route (index 2), the developer needs to use `Error.errorMessage.payload.failures[2\\]` expression.

References:<https://docs.mulesoft.com/mule-runtime/4.3/scatter-gather-reference#scatter-gather-output>

QUESTION 2

A Mule application includes a subflow containing a Scatter.Gather scope. Within each log of the Scatter.Gatter. an HTTP connector calls a PUT endpoint to modify records in different upstream system. The subflow is called inside an Unit successful scope to retry if a transitory exception is raised.

A technical spike is being performed to increase reliability of the Mule application.

Which steps should be performed within the Mule flow above the ensure idempotent behavior?

- A. Change the PUT requests inside the Scatter-Gather to POST requests
- B. Ensure an error-handling flow performs corrective actions to roll back all changes if any leg of the Scatter-Gather fails
- C. Remove the Put requests from the Scatter-Getter and perform them sequentially
- D. None, the flow already exhibits idempotent behavior

Correct Answer: B

To ensure idempotent behavior within a Mule flow that contains a subflow with a Scatter-Gather scope, the developer should ensure an error-handling flow performs corrective actions to roll back all changes if any leg of the Scatter-Gather fails. Idempotency means that multiple identical requests have the same effect as a single request. Therefore, if one of the HTTP requests inside the Scatter-Gather fails, the error-handling flow should undo any changes made by other successful requests to ensure consistency and avoid partial updates. References: <https://docs.mulesoft.com/mule-runtime/4.3/scatter-gather-concept><https://docs.mulesoft.com/mule-runtime/4.3/error-handling>

**QUESTION 3**

A Mule application for processing orders must log the order ID for every log message output.

What is a best practice to enrich every log message with the order ID?

- A. Use flow variables within every logger processor to log the order ID
- B. Set a flow variable and edit the log4/2.xml file to output the variable as part of the message pattern
- C. Create a custom XML SDK component to wrap the logger processor and automatically add the order ID within the connector
- D. Use the Tracing module to set logging variables with a Mapped Diagnostic Context

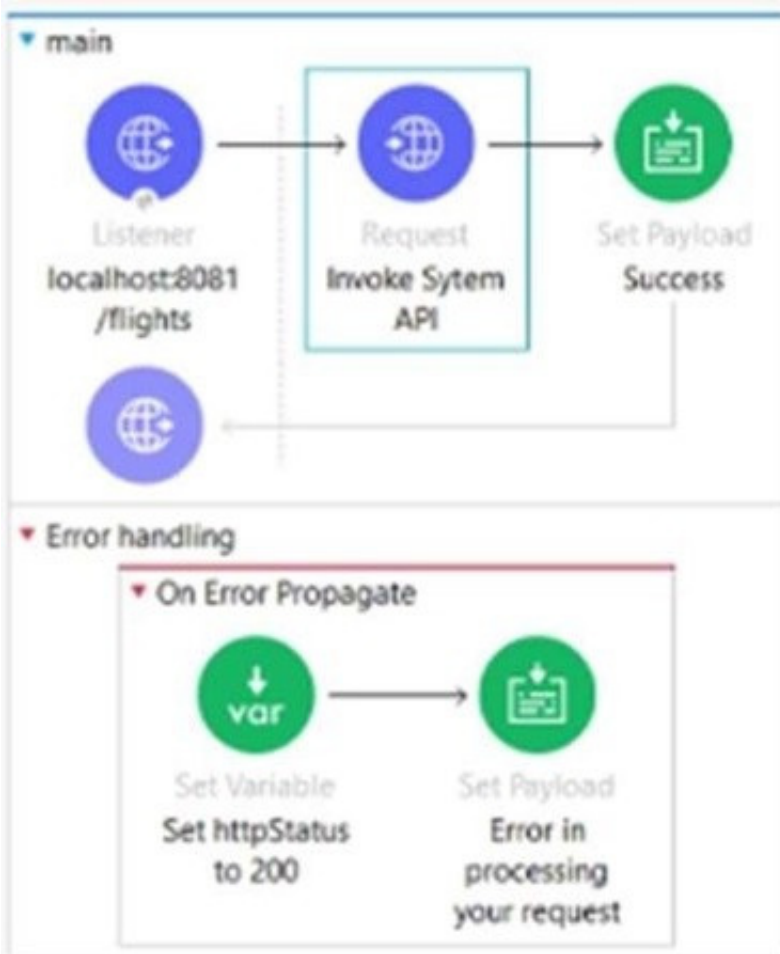
Correct Answer: D

To enrich every log message with the order ID, the developer should use the Tracing module to set logging variables with a Mapped Diagnostic Context (MDC). The Tracing module allows adding custom key-value pairs to log messages using MDC variables. The developer can use Set Logging Variables operation to set the order ID as an MDC variable and then use it in any logger processor within the same thread or event. References:<https://docs.mulesoft.com/tracing-module/1.0/tracing-module-reference#set-logging-variables>

QUESTION 4

The HTTP Request operation raises an HTTP CONNECTIVITY error.

Which HTTP status code and body are returned to the web client?



General

MIME Type

Redelivery

Responses

Notes

Help

Response

Body: `1 payload`

Headers: Headers

Status code: `vars.HttpStatus`

Reason phrase:

Error Response

Body: `1 output text/plain --- error.description`

Headers: Headers

Name

- A. HTTP Status Code:200. Body `Error in processing your request
- B. HTTP Status Code:500. Body `The HTTP CONNECTIVITY Error description
- C. HTTP Status Code:500. Body `Error in processing your request



D. HTTP Status Code:500. Body `Error in processing your request`

Correct Answer: C

When the HTTP Request operation raises an HTTP CONNECTIVITY error, it triggers an on-error-continue handler that sets a payload with `Error in processing your request`. Since no status code is explicitly set in this handler, it defaults to 500 (INTERNAL SERVER ERROR). Therefore, the web client receives an HTTP response with status code 500 and body `Error in processing your request`. References:<https://docs.mulesoft.com/mule-runtime/4.3/error-handling#on-error-continue>

QUESTION 5

A Mule application defines as SSL/TLS keystore properly `tis,keystore.keyPassword` as secure.

How can this property be referenced to access its value within the application?

- A. `#{secure::tis,keystore,keyPassword}`
- B. `#{secure::tis,keystore,keyPassword}`
- C. `#{secure::tis,keystore,keyPassword}`
- D. `p{secure::tis,keystore,keyPassword}`

Correct Answer: B

secure::tis,keystore,keyPassword
Short Explanation of Correct Answer Only: To reference a secure property value within the application, the developer needs to use the syntax `{secure::}`. In this case, the property name is `tis,keystore,keyPassword`, so the correct syntax is `#{secure::tis,keystore,keyPassword}`. References: <https://docs.mulesoft.com/mule-runtime/4.3/secure-configuration-properties#referencing-secure-properties>

[MCD-LEVEL-2 PDF Dumps](#) [MCD-LEVEL-2 VCE Dumps](#)

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