



# 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

Refer to the exhibit.

```
<flow name="implementation" >
  <raise-error doc:name="Raise error" type="APP:CUSTOM_ERROR"/>
</flow>

<munit:test name="start-up-test" description="Test that Mule app starts up" expectedErrorType="APP:CUSTOM_ERROR">
  <munit:execution>
    <flow-ref doc:name="implementation" name="implementation"/>
  </munit:execution>
  <munit:validation >
    <munit-tools:assert-that doc:name="Assert that" expression="#[true]" is="#[MunitTools::equalTo(false)]"/>
  </munit:validation>
</munit:test>
```

The flow name is `implementation` with code for the MUnit test case.

When the MUnit test case is executed, what is the expected result?

- A. The test case fails with an assertion error
- B. The test throws an error and does not start
- C. The test case fails with an unexpected error type
- D. The test case passes

Correct Answer: A

Based on the code snippet and MUnit test case below, when the MUnit test case is executed, the expected result is that the test case fails with an assertion error. This is because the assert-equals processor compares two values for equality, and fails if they are not equal. In this case, the expected value is `Hello World`, but the actual value returned by the implementation flow is `Hello Mule`. Therefore, the assertion fails and an error is thrown. References:<https://docs.mulesoft.com/munit/2.3/assert-equals-processor>

## QUESTION 2

An API has been built to enable scheduling email provider. The front-end system does very little data entry validation, and problems have started to appear in the email that go to patients. A `validate-customer` flow is added to validate the data.

What is the expected behavior of the `validate-customer` flow?



```
<flow name="validate-customer">
  <validation:all>
    <validation:is-email email="#{payload.customer.emailAddress}" message="invalid email address">
      <error-mapping sourceType="VALIDATION:INVALID_EMAIL" targetType="SCHEDULE:INVALID_EMAIL_ADDRESS"/>
    </validation:is-email>
    <validation:matches-regex value="#{payload.schedule.appointmentDate}"
      regex="^\d{4}-\d{2}-\d{2}$" message="Invalid appointment date">
      <error-mapping sourceType="VALIDATION:MISMATCH" targetType="SCHEDULE:INVALID_APPOINTMENT_DATE"/>
    </validation:matches-regex>
    <validation:is-not-null value="#{payload.customer.name}" message="Invalid customer name">
      <error-mapping sourceType="VALIDATION:NULL" targetType="SCHEDULE:INVALID_CUSTOMER_NAME"/>
    </validation:is-not-null>
  </validation:all>
</flow>
```

- A. If only the email address is invalid a VALIDATION.INVALID\_EMAIL error is raised
- B. If the email address is invalid, processing continues to see if the appointment data and customer name are also invalid
- C. If the appointment date and customer name are invalid, a SCHEDULE:INVALID\_APPOINTMENT\_DATE error is raised
- D. If all of the values are invalid the last validation error is raised: SCHEDULE:INVALID\_CUSTOMER\_NAME

Correct Answer: A

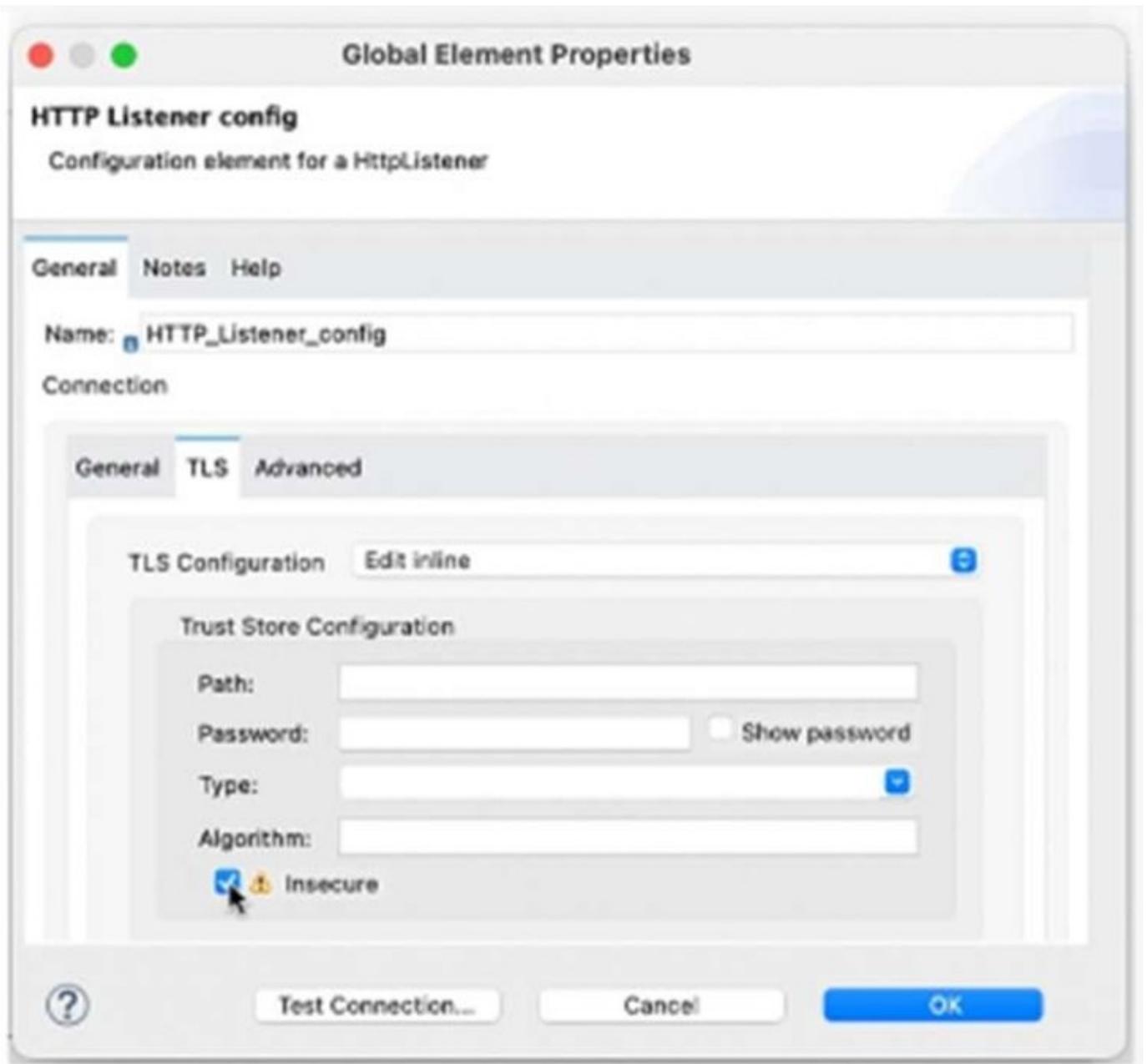
The validate-customer flow uses an until-successful scope to validate each field of the customer data. The until-successful scope executes its processors until they succeed or exhausts the maximum number of retries. If any processor fails, it raises an error and stops executing the remaining processors. Therefore, if only the email address is invalid, a VALIDATION.INVALID\_EMAIL error is raised and the validation of appointment date and customer name is skipped.

References: <https://docs.mulesoft.com/mule-runtime/4.3/until-successful-scope>

---

### QUESTION 3

Refer to the exhibit.



What is the result if `\\Insecure\\` selected as part of the HTTP Listener configuration?

- A. The HTTP Listener will trust any certificate presented by the HTTP client
- B. The HTTP Lister will accept any unauthenticated request
- C. The HTTP listener will only accept HTTP requests
- D. Mutual TLS authentication will be enabled between this HTTP Listener and an HTTP client

Correct Answer: C

Based on the exhibit below, if `\\Insecure\\` is selected as part of the HTTP Listener configuration, the HTTP listener will only accept HTTP requests. This means that no TLS context will be configured for this listener and no encryption or authentication will be applied to incoming requests. The protocol attribute of this listener will be set to HTTP instead of HTTPS. References:<https://docs.mulesoft.com/http-connector/1.6/http-listener-ref#insecure>



#### QUESTION 4

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 5

Which plugin or dependency is required to unit test modules created with XML SDK?

- A. XMLUnit
- B. Junit
- C. MUnit Extensions Maven plugin
- D. MUnit Maven plugin

Correct Answer: C

To unit test modules created with XML SDK, the developer needs to use the MUnit Extensions Maven plugin. This plugin allows testing XML SDK modules using MUnit by adding a dependency to the module under test and using a custom processor tag to invoke it. References:<https://docs.mulesoft.com/mule-sdk/1.1/xml-sdk#testing>

[Latest MCD-LEVEL-2 Dumps](#)

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

[MCD-LEVEL-2 Brindumps](#)