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

Which type of cache invalidation does the Cache scope support without having to write any additional code?

A. Write-through invalidation

B. White-behind invalidation

C. Time to live

D. Notification-based invalidation

Correct Answer: C

The Cache scope supports time to live (TTL) as a cache invalidation strategy without having to write any additional code. TTL specifies how long the cached response is valid before it expires and needs to be refreshed. The Cache scope also supports custom invalidation strategies using MEL or DataWeave expressions.

References:https://docs.mulesoft.com/mule-runtime/4.3/cache-scope#cache_invalidation

QUESTION 2

A Mule application uses API autodiscovery to access and enforce policies for a RESTful implementation.

A. Northing because flowRef is an optional attribute which can be passed runtime

B. The name of the flow that has APIkit Console to receive all incoming RESTful operation requests.

C. Anyof the APIkit generate implement flows

D. The name of the flow that has HTTP listener to receive all incoming RESTful operation requests

Correct Answer: D

To use API autodiscovery to access and enforce policies for a RESTful implementation, flowRef must be set to the name of the flow that has HTTP listener to receive all incoming RESTful operation requests. This way, API autodiscovery can identify the API implementation and associate it with the corresponding API specification and policies in API Manager. The flow that has HTTP listener is usually the main flow that contains the APIKit Router.
References:https:// docs.mulesoft.com/api-manager/2.x/api-auto-discovery-new-concept#flowref

QUESTION 3

When implementing a synchronous API where the event source is an HTTP Listener, a developer needs to return the same correlation ID back to the caller in the HTTP response header.

How can this be achieved?

A. Enable the auto-generate CorrelationID option when scaffolding the flow

B. Enable the CorrelationID checkbox in the HTTP Listener configuration

C. Configure a custom correlation policy

D. NO action is needed as the correlation ID is returned to the caller in the response header by default

Correct Answer: D

When implementing a synchronous API where the event source is an HTTP Listener, Mule automatically propagates some message attributes between flows via outbound and inbound properties. One of these attributes is correlation ID,

which is returned to the caller in the response header by default as MULE_CORRELATION_ID.

References: https://docs.mulesoft.com/mule-runtime/4.3/about-mule-message#message-attributes

QUESTION 4

A mule application exposes and API for creating payments. An Operations team wants to ensure that the Payment API is up and running at all times in production.

Which approach should be used to test that the payment API is working in production?

A. Create a health check endpoint that listens on a separate port and uses a separate HTTP Listener configuration from the API

B. Configure the application to send health data to an external system

C. Create a health check endpoint that reuses the same port number and HTTP Listener configuration as the API itself

D. Monitor the Payment API directly sending real customer payment data

Correct Answer: A

To test that the payment API is working in production, the developer should create a health check endpoint that listens on a separate port and uses a separate HTTP Listener configuration from the API. This way, the developer can isolate the health check endpoint from the API traffic and avoid affecting the performance or availability of the API. The health check endpoint should return a simple response that indicates the status of the API, such as OK or ERROR.
References:https://docs.mulesoft.com/api-functional-monitoring/afm-create-monitor#create-a-monitor

QUESTION 5

A Mule application contain two policies Policy A and Policy A has order1, and Policy B has order 2. Policy A Policy B, and a flow are defined by he configuration below.

```
<http-policy:proxy name="policy-A">
  <http-policy:source>
    <A1/>
    <http-policy:execute-next/>
    <A2/>
  </http-policy:source>
</http-policy:proxy>

<http-policy:proxy name="policy-B">
  <http-policy:source>
    <B1/>
    <http-policy:execute-next/>
    <B2/>
  </http-policy:source>
</http-policy:proxy>

<flow name="flow">
  <http:listener/>
  <F1/>
</flow>
```

When a HTTP request arrives at the Mule application\\'s endpoint, what will be the execution order?

A. A1, B1, F1, B2, A2

B. B1, A1, F1, A2, B2

C. F1, A1, B1, B2, A2

D. F1, B1, A1, A2, B2

Correct Answer: A

Based on the configuration below, when a HTTP request arrives at the Mule application\\'s endpoint, the execution order will be A1, B1, F1, B2, A2. This is because policies are executed before and after the API implementation flow according to their order attribute. Policy A has order 1, which means it is executed first before Policy B, which has order 2. The flow is executedafter both policies are executed before the flow. Then, Policy B is executed after the flow before Policy A is executed after the flow. References:https://docs.mulesoft.com/api-manager/2.x/policies-policy-order