RealValidExam

Try before you buy

Download a free sample of any of our exam questions and answers

- 24/7 customer support, Secure shopping site
- Free One year updates to match real exam scenarios
- If you failed your exam after buying our products we will refund the full amount back to you.

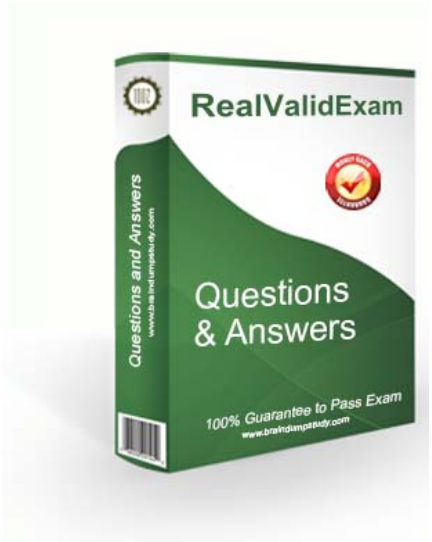
Choose an exam to sample

Select a vendor...

Select an exam...

Your email address

Download Now



QUALITY AND VALUE

RealValidExam Practice Exams are written to the highest standards of technical accuracy, using only certified subject matter experts and published authors for development - no all dumps.

TESTED AND APPROVED

We are committed to the process of vendor and third party approvals. We believe professionals and executives alike deserve the confidence of quality coverage these authorizations provide.

EASY TO PASS

If you prepare for the exams using our RealValidExam testing engine, It is easy to succeed for all certifications in the first attempt. You don't have to deal with all dumps or any free torrent / rapidshare all stuff.

TRY BEFORE BUY

RealValidExam offers free demo of each product. You can check out the interface, question quality and usability of our practice exams before you decide to buy.

http://www.realvalidexam.com

100% real and valid exam dumps can ensure you pass at the first attempt

Exam : **MCD-Level-2**

Title : MuleSoft Certified Developer - Level 2 (Mule 4)

Vendor : MuleSoft

Version : DEMO

NO.1 A developer has created the first version of an API designed for business partners to work commodity prices.
What should developer do to allow more than one major version of the same API to be exposed by the implementation?
A. In Design Center, open the RAML and modify each operation to include the major version number
B. In Anypoint Studio, generate scaffolding from the RAML, and the modify the <http:listerner> in the generated flows to include a parameter to replace the version number
C. In Design Center, open the RAML and modify baseUn to include a variable that indicates the version number
D. In Anypoint Studio, generate scaffolding from the RAML, and then modify the flow names generated by APIKit to include a variable with the major version number
Answer: C
Explanation:
To allow more than one major version of the same API to be exposed by the implementation, the developer should modify the baseUri property in the RAML file to include a variable that indicates the version number.
The baseUri property defines the base URL of the API and can include variables that are replaced with actual values when mocking or deploying the API. By using a variable for the version number, the developer can expose different versions of the API using different base URLs and avoid conflicts or confusion. References:
https://docs.mulesoft.com/api-designer/design-modify-raml-specs#baseurihttps://docs.mulesoft.com/api-manager

NO.2 Two APIs are deployed to a two-node on-prem cluster. Due to a requirements change, the two APIs must communicate to exchange data asynchronously.
A. If the two APIs use the same domain, the VM Connector can be leveraged
B. The VM Connector is used to inter-application communication, so it is not possible to use the VM Connector
C. Instead of using the VM Connector use <flow-ref>directly
D. It is not possible to use the VM Connector since the APIs are running in a cluster mode and each mode has it own set of VM Queues
Answer: A
Explanation:
To communicate asynchronously between two APIs deployed to a two-node on-prem cluster, the developer can use the VM Connector if the two APIs use the same domain. The VM Connector allows passing messages between different Mule applications within a single Mule runtime instance or across different instances using shared memory or persistent storage. If two APIs are deployed under the same domain, they can share resources such as VM queues and communicate asynchronously using VM Connector operations. References:
https://docs.mulesoft.com/mule-runtime/4.3/vm-connectorhttps://docs.mulesoft.com/mule-runtime/4.3/shared-res

NO.3 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. Any of the APIkit generate implement flows
D. The name of the flow that has HTTP listener to receive all incoming RESTful operation requests
Answer: D
Explanation:
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 andassociate 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

NO.4 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
Answer: C
Explanation:
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

NO.5 A developer is working on a project that requires encrypting all data before sending it to a backend application. To accomplish this, the developer will use PGP encryption in the Mule 4 Cryptography module.
What is required to encrypt the data before sending it to the backend application?
A. The application needs to configure HTTPS TLS context information to encrypt the data
B. The application needs to both the private and public keys to encrypt the data
C. The application needs the public key from the backend service to encrypt the data
D. The application needs the private key from the backend service to encrypt the data
Answer: C
Explanation:
To encrypt the data before sending it to the backend application using PGP encryption, the application needs the public key from the backend service. PGP encryption uses a public-key cryptography system, which means that each party has a pair of keys: a public key and a private key. The public key is used to encrypt data, and the private key is used to decrypt data. Therefore, to encrypt data for a specific recipient (the backend service), the application needs to use the recipient's public key. The recipient can then use its own private key to decrypt the data.
References:https://docs.mulesoft.com/mule-runtime/4.3/cryptography-pgp

NO.6 In a Mule project, Flow-1 contains a flow-ref to Flow-2 depends on data from Flow-1 to

execute successfully.
Which action ensures the test suites and test cases written for Flow-1 and Flow-2 will execute successfully?

A. Chain together the test suites and test cases for Flow-1 and Flow-2

B. Use ''Set Event to pass the input that is needed, and keep the test cases for Flow-1 and Flow-2 independent

C. Use ''Before Test Case'' To collect data from Flow-1 test cases before running Flow-2 test cases

D. Use 'After Test Case' to produce the data needed from Flow-1 test cases to pass to Flow-2 test cases

Answer: B

Explanation:
To ensure the test suites and test cases written for Flow-1 and Flow-2 will execute successfully, the developer should use a Set Event processor to pass the input that is needed by Flow-2, and keep the test cases for Flow-1 and Flow-2 independent. This way, the developer can isolate the testing of each flow and avoid coupling them together. References:https://docs.mulesoft.com/munit/2.3/munit-test-flow