SALESFORCE-MULESOFT-DEVELOPER-II$^{Q\&As}$

Salesforce Certified MuleSoft Developer 2 (SP24)

Pass Salesforce SALESFORCE-MULESOFT-DEVELOPER-II Exam with 100% Guarantee

SATISFACTION GUARANTEED
100%
SATISFACTION GUARANTEED

QUESTION 1

A Mule application deployed to a standardalone Mule runtime uses VM queues to publish messages to be consumed asynchronously by another flow. In the case of a system failure, what will happen to in-flight messages in the VM queues that have been consumed?

A. For nay type of queue, the message will be processed after the system comes online

B. For persistent queues, the message will be processed after the system comes online

C. For transient queues, the message will be processed after the system comes online

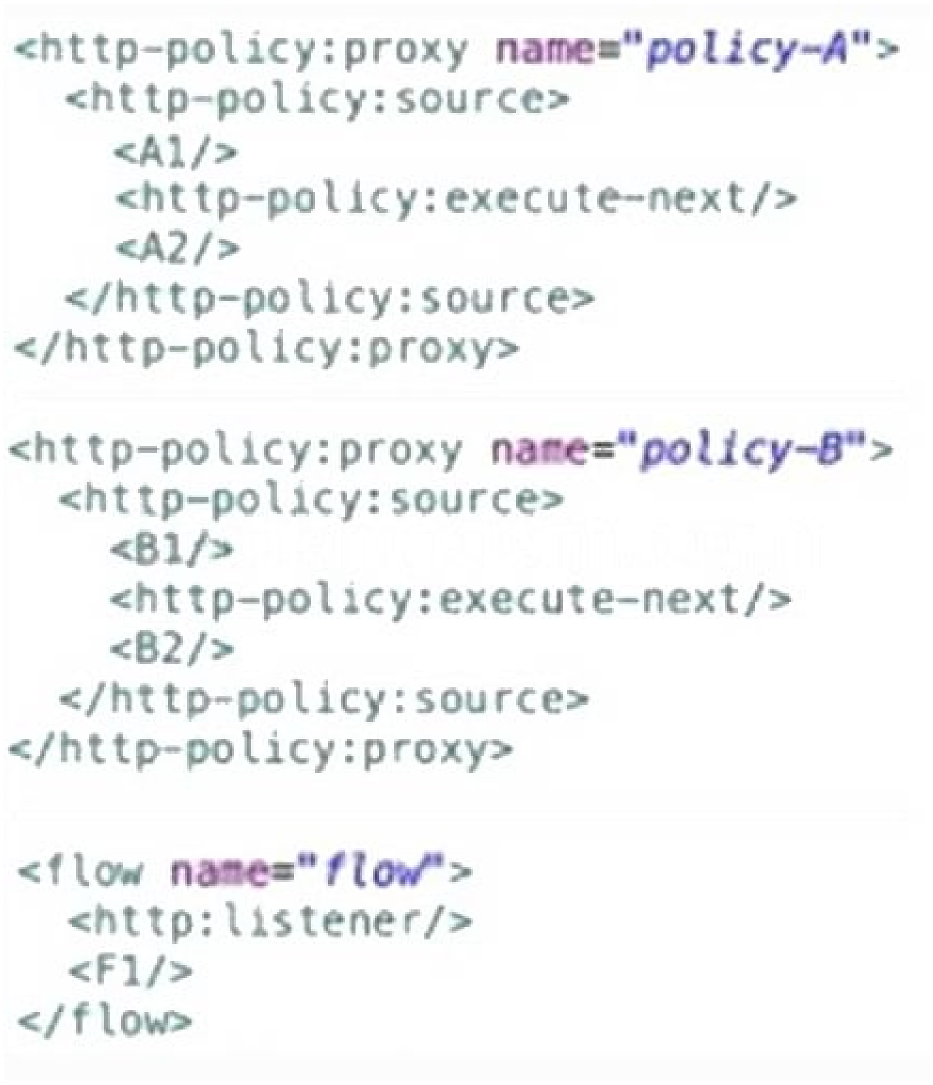
D. For any type of queue, the message will be lost

Correct Answer: B

In case of a system failure, in-flight messages in persistent VM queues that have been consumed will be processed after the system comes online. This is because persistent VM queues store messages on disk and guarantee delivery even if there is a system crash or restart. Therefore, any in-flight messages that have been consumed but not processed will be recovered from disk and processed when the system is back online. https://docs.mulesoft.com/mule-runtime.3/vmconnector#persistent-queues

QUESTION 2

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 executed after both policies are executed before the flow. Then, Policy B is executed after the flow before Policy A is executed after the flow. https://docs.mulesoft.com/api- manager.x/policies-policy-order

QUESTION 3

Which pattern can a web API use to notify its client of state changes as soon as they occur?

A. HTTP Webhock

B. Shared database trigger

C. Schedule Event Publisher

D. ETL data load

Correct Answer: A

A web API can use HTTP Webhook to notify its client of state changes as soon as they occur. A webhook is an HTTP callback that allows an API to send real-time notifications to another system or application when an event happens. The client registers a URL with the API where it wants to receive notifications, and then the API sends an HTTP request to that URL with information about the event. https://docs.mulesoft.com/connectors/webhook/webhook-connector

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.

Reference:

https://docs.mulesoft.com/api-functional-monitoring/afm-create-monitor#create-a-monitor

QUESTION 5

A Mule application need to invoice an API hosted by an external system to initiate a process. The external API takes anywhere between one minute and 24 hours to compute its process. Which implementation should be used to get response data from the external API after it completes processing?

A. Use an HTTP Connector to invoke the API and wait for a response

B. Use a Scheduler to check for a response every minute

C. Use an HTTP Connector inside Async scope to invoice the API and wait for a response

D. Expose an HTTP callback API in Mule and register it with the external system

Correct Answer: D

To get response data from the external API after it completes processing, the developer should expose an HTTP callback API in Mule and register it with the external system. This way, the external API can invoke the callback API with the

response data when it is ready, instead of making the Mule application wait for a long time or poll for a response repeatedly.

Reference:

https://docs.mulesoft.com/mule-runtime.3/http-listener-ref#callback

---

QUESTION 6

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

Correct Answer: C

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. Reference: https://docs.mulesoft.com/mule-runtime.3/cryptography-pgp

---

QUESTION 7

An API has been developed and deployed to CloudHub Among the policies applied to this API is an allowlist of IP addresses. A developer wants to run a test in Anypoint Studio and does not want any policies applied because their workstation is not included in the allowlist. What must the developer do in order to run this test locally without the policies applied?

A. Create a properties file specifically for local development and set the API instance ID to a value that is not used in API Manager

B. Pass in the runtime parameter "-Danpow.platform.gatekeeper=disabled\\'\\'

C. Deactivate the API in API Manager so the Autodiscovery element will not find the application when it runs in Studio

D. Run the test as-s, with no changes because the Studio runtime will not attempt to connect to API Manager

Correct Answer: B

To run a test locally without the policies applied, the developer should create a properties file specifically for local development and set the API instance ID to a value that is not used in API Manager. This way, the developer can use different configuration properties for different environments and avoid triggering API autodiscovery when running tests locally. API autodiscovery is a mechanism that associates an API implementation with its corresponding API specification and policies in API Manager based on its API instance ID. By setting this ID to a value that does not exist in API Manager, the developer can prevent API autodiscovery from finding and applying any policies to the local test. https://docs.mulesoft.com/api-manager.x/api-auto-discovery-newconcept#configuring-api-autodiscovery https://docs.mulesoft.com/mule-runtime.3/configuringproperties

QUESTION 8

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.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. https:// docs.mulesoft.com/tracing-module.0/tracingmodule-reference#set-logging-variables

QUESTION 9

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 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

Correct Answer: C

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. https:// docs.mulesoft.com/api-designer/design-modifyraml-specs#baseuri https://docs.mulesoft.com/api-manager.x/api-versioning

QUESTION 10

Mule application A is deployed to CloudHub and is using Object Store v2. Mute application B is also deployed to CloudHub. Which approach can Mule application B use to remove values from Mule application A\\'S Object Store?

A. Object Store v2 REST API

B. CloudHub Connector

C. Object Store Connector

D. CloudHub REST API

Correct Answer: A

To remove values from Mule application A\\'s Object Store v2, Mule application B can use Object Store v2 REST API. This API allows performing operations on Object Store v2 resources using HTTP methods, such as GET, POST, PUT, and DELETE. Mule application B can use the DELETE method to remove values from Mule application A\\'s Object Store v2 by specifying the object store ID and the key of the value to delete. https:// docs.mulesoft.com/object-store/osv2-apis