



**Free Questions for [Salesforce-MuleSoft-Developer-II](#) by
[go4braindumps](#)**

Shared by [Meadows](#) on [22-07-2024](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

A healthcare customer wants to use hospital system data, which includes code that was developed using legacy tools and methods. The customer has created reusable Java libraries in order to read the data from the system.

What is the most effective way to develop an API retrieve the data from the hospital system?

Options:

- A- Refer to JAR files in the code
- B- Include the libraries writes deploying the code into the runtime
- C- Create the Java code in your project and invoice the data from the code
- D- Install libraries in a local repository and refer to it in the pm.xml file

Answer:

D

Explanation:

To develop an API that retrieves data from a hospital system using reusable Java libraries, the developer should install libraries in a local repository and refer to it in the pom.xml file. This way, the developer can use Maven to manage dependencies and invoke Java code from Mule applications using Java Module operations. Reference: <https://docs.mulesoft.com/mule-runtime/4.3/java-module-reference#add-the-java-module-to-your-project> <https://docs.mulesoft.com/mule-runtime/4.3/java-module-reference#invoke-java-code>

Question 2

Question Type: MultipleChoice

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?

Options:

- 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. Reference: <https://docs.mulesoft.com/mule-runtime/4.3/cryptography-pgp>

Question 3

Question Type: MultipleChoice

A Mule application deployed to multiple Cloudhub 2.0 replicas needs to temporarily persist large files over 10MB between flow executions, and routinely needs to query whether the file data exists on separate executions.

How can this be achieved?

Options:

- A- Store the contents of the file on separate storage, and store the key and location of the file Object using Object Store v2
- B- Use an in-memory Object Store
- C- Store the key and full contents of the file in an Object Store
- D- Store the key and full contents of the file, caching the filename and location between requests

Answer:

A

Explanation:

To temporarily persist large files over 10MB between flow executions, and routinely query whether the file data exists on separate executions, the developer should store the contents of the file on separate storage, and store the key and location of the file object using Object Store v2. This way, the developer can avoid storing large files in memory or exceeding the size limit of Object Store v2 (10MB per object). The developer can also use Object Store v2 operations to query, retrieve, or delete the file object by its key. Reference: <https://docs.mulesoft.com/object-store/osv2-faq#can-i-store-files-in-object-store-v2>

Question 4

Question Type: MultipleChoice

A company has been using CI/CD. Its developers use Maven to handle build and deployment activities.

What is the correct sequence of activities that takes place during the Maven build and deployment?

Options:

- A- Initialize, validate, compute, test, package, verify, install, deploy
- B- Validate, initialize, compile, package, test, install, verify, verify, deploy
- C- Validate, initialize, compile, test package, verify, install, deploy
- D- Validation, initialize, compile, test, package, install verify, deploy

Answer:

C

Explanation:

The correct sequence of activities that takes place during the Maven build and deployment is validate, initialize, compile, test package, verify, install, deploy. These are Maven lifecycle phases that define a sequence of goals to execute during a build process. Each phase represents a stage in the build lifecycle and can have zero or more goals bound to it. Reference: <https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>

Question 5

Question Type: MultipleChoice

Refer to the exhibit.

```
traits:
  client-id-required:
    headers:
      client_id:
        type: string
      client_secret:
        type: string

protocols:
- HTTPS
- HTTP

types:
  Customers: !include datatypes/Customers-request.raml

/customers:
  is: [client-id-required]
  put:
    body:
      application/json:
        type: Customers
        example: !include examples/Customers-request.json
    responses:
      200:
        description: Successfull
        body:
          application/json:
            example: !include examples/Customers-response.json
      400:
        description: Bad request
        body:
          application/json:
            example: !include examples/postErrorCode403.json
      404:
        description: Resource not found
        body:
          application/json:
            example:
              !include examples/postErrorCode404.json
      500:
        description: Internal server error
        body:
          application/json:
            example:
              !include examples/postErrorCode500.json
      501:
        description: Not implemented
        body:
          application/json:
            example:
              !include examples/postErrorCode501.json
```


A developer generates the base scaffolding for an API in Anypoint Studio.

Which HTTP status code is returned while testing using the API Kit console if no values are entered in client-secret?

Options:

- A- HTTP status code:200
- B- HTTP status code:403
- C- HTTP status code:400
- D- HTTP status code:500

Answer:

B

Explanation:

Based on the code snippet and schema.json file below, when testing using the API Kit console if no values are entered in client-secret, HTTP status code 403 (FORBIDDEN) is returned. This is because client-secret is defined as a required header parameter in schema.json file, which means that it must be present in every request. If no values are entered in client-secret, then it is equivalent to omitting this header parameter, which violates the schema and causes APIKit Router to return HTTP status code 403. Reference: <https://docs.mulesoft.com/apikit/4.x/apikit-4-headers>

Question 6

Question Type: MultipleChoice

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?

Options:

- 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

Answer:

B

Explanation:

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. Reference: <https://docs.mulesoft.com/mule-runtime/4.3/vm-connector#persistent-queues>

Question 7

Question Type: MultipleChoice

When registering a client application with an existing API instance or API Group instance, what is required to manually approve or reject request access?

Options:

- A-** To configure the SLA tier for the application and have the role of Organization Administrator, API Manager Environment Administrator, or the Manage Contacts permission
- B-** To configure the SLA tier for the application and have the Exchange Administrator permission

C- To configure the SLA tier for the application

D- To only have Exchange Administrator permission

Answer:

D

Explanation:

To manually approve or reject request access when registering a client application with an existing API instance or API Group instance, it is required to configure the SLA tier for the application and have one of the following roles or permissions: Organization Administrator, API Manager Environment Administrator, or Manage Contracts permission. These roles or permissions allow managing client applications and contracts in API Manager. Reference: <https://docs.mulesoft.com/api-manager/2.x/client-applications#managing-client-applications-and-contracts>

Question 8

Question Type: MultipleChoice

Refer to the exhibit.

POST

localhost:8081/validate

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

JSON

Beautify

```
1 {
2   "firstName": "John",
3   "lastName": "Doe",
4   "age": -1
5 }
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<mule xmlns:http="http://www.mulesoft.org/schema/mule/http" xmlns:ee="http://www.mulesoft.org/schema/mule/ee/core"
  xmlns:json="http://www.mulesoft.org/schema/mule/json"
  xmlns="http://www.mulesoft.org/schema/mule/core" xmlns:doc="http://www.mulesoft.org/schema/mule/documentation" xmlns:xsi="http://www.
http://www.mulesoft.org/schema/mule/json http://www.mulesoft.org/schema/mule/json/current/mule-json.xsd
http://www.mulesoft.org/schema/mule/ee/core http://www.mulesoft.org/schema/mule/ee/core/current/mule-ee.xsd
http://www.mulesoft.org/schema/mule/http http://www.mulesoft.org/schema/mule/http/current/mule-http.xsd">
  <flow name="Validate-JsonFlow" doc:id="b8698bf2-3703-418b-a909-ad186f476299" >
    <http:listener doc:name="Listener" doc:id="d6d55854-64f4-408e-960f-9080d548049a" config-ref="HTTP_Listener_config" path="/validate" />
    <json:validate-schema doc:name="Validate schema" doc:id="2e050bf2-81ff-4e86-961d-9a68b79b1126" schema="Schema.json" />
    <logger level="INFO" doc:name="Logger" doc:id="b27a7464-30dd-4768-b0af-616f5ccd4680" message="message is processed successfully" />
  </flow>
</mule>
```

```
{
  "$id": "https://example.com/person.schema.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "Person",
```

Based on the code snippet, schema,json file, and payload below, what is the outcome of the given code snippet when a request is sent with the payload?

Options:

- A- The Mule flow will execute successfully with status code 200, and the response will be the JSON sent in request
- B- The Mule flow will execute successfully with status code 204
- C- The Mule flow will throw the exception 'JSON:SCHEMA_NOT_HONOURED'
- D- The Mule flow will execute successfully with status code 200m and a response will display the message " Age in years which must equal to or greater than zero."

Answer:

C

Explanation:

Based on the code snippet, schema.json file, and payload below, the outcome of the given code snippet when a request is sent with the payload is that the Mule flow will throw the exception 'JSON:SCHEMA_NOT_HONOURED'. This is because the payload does not conform to the schema.json file, which specifies that age must be a number greater than or equal to zero. The payload has age as a string with a negative value, which violates the schema. Therefore, the validate-schema operation throws an error with type 'JSON:SCHEMA_NOT_HONOURED'. Reference: <https://docs.mulesoft.com/json-module/1.1/json-validate-schema>

Question 9

Question Type: MultipleChoice

A company with MuleSoft Titanium develops a Salesforce System API using MuleSoft out-of-the-box Salesforce Connector and deploys the API to CloudHub.

Which steps provide the average number of requests and average response time of the Salesforce Connector?

Options:

A- Access Anypoint Monitoring's built-in dashboard. Select a resource.

Locate the information under the Connectors tab.

B- Access Anypoint Monitoring's built-in dashboard

Select a resource.

Create a custom dashboard to retrieve the information.

C- Access Anypoint Monitoring built-in dashboard.

Select a resource.

Locate the information under Log Manager < Raw Data.

D- Change the API Implementation to capture the information in the log.

Retrieve the information from the log file.

Answer:

A

Explanation:

To get the average number of requests and average response time of the Salesforce Connector, the developer should access Anypoint Monitoring's built-in dashboard, select a resource (such as an application or an API), and locate the information under the Connectors tab. The Connectors tab shows metrics for each connector used by the resource, such as average requests per minute, average response time, and failures. Reference: <https://docs.mulesoft.com/monitoring/built-in-dashboard-reference>

Question 10

Question Type: MultipleChoice

A new Mule project has been created in Anypoint Studio with the default settings.

Which file inside the Mule project must be modified before using Maven to successfully deploy the application?

Options:

- A- Settings.xml
- B- Config.yaml
- C- Pom.xml
- D- Mule.artificact.json

Answer:

C

Explanation:

To use Maven to successfully deploy the application, the developer needs to modify the pom.xml file inside the Mule project. The pom.xml file contains the configuration and dependencies of the Mule application, as well as the deployment information such as the CloudHub region, environment, and worker size. The developer needs to specify these properties in the pom.xml file before using Maven to deploy the application. Reference: <https://docs.mulesoft.com/mule-runtime/4.3/mmp-concept#application-deployment-descriptor>

Question 11

Question Type: MultipleChoice

Which command is used to convert a JKS keystore to PKCS12?

Options:

- A- Keytool-importkeystore --srckeystore keystore p12-srcstoretype PKCS12 --destkeystore keystore.jks --deststoretype JKS
- B- Keytool-importkeystore --srckeystore keystore p12-srcstoretype JKS --destkeystore keystore.p12 --deststoretype PKCS12
- C- Keytool-importkeystore --srckeystore keystore jks-srcstoretype JKS --destkeystore keystore.p13 --deststoretype PKCS12
- D- Keytool-importkeystore --srckeystore keystore jks-srcstoretype PKCS12 --destkeystore keystore.p12 --deststoretype JKS

Answer:

B

Explanation:

To convert a JKS keystore to PKCS12, the developer needs to use the `keytool-importkeystore` command with the following options: `-srckeystore keystore.jks -srcstoretype JKS -destkeystore keystore.p12 -deststoretype PKCS12`. This command imports all entries from a source JKS keystore (`keystore.jks`) into a destination PKCS12 keystore (`keystore.p12`). Reference:

<https://docs.oracle.com/en/java/javase/11/tools/keytool.html#GUID-5990A2E4-78E3-47B7-AE75-6D1826259549>

To Get Premium Files for Salesforce-MuleSoft-Developer-II Visit

<https://www.p2pexams.com/products/salesforce-mulesoft-developer-ii>

For More Free Questions Visit

<https://www.p2pexams.com/salesforce/pdf/salesforce-mulesoft-developer-ii>

