Free Questions for Salesforce-MuleSoft-Developer-II by actualtestdumps

Shared by Gillespie on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

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?

Options:

A- Object Store v2 REST API

B- CloudHub Connector

C- Object Store Connector

D- CloudHub REST API

Answer:

A

Explanation:

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. Reference: https://docs.mulesoft.com/object-store/osv2-apis

Question 2

Question Type: **MultipleChoice**

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?

Options:

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

Answer:

A

Explanation:

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 3

Question Type: **MultipleChoice**

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

Options:

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 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. Reference: https://docs.mulesoft.com/api-manager/2.x/api-auto-discovery-new-concept#flowref

Question 4

Question Type: MultipleChoice

A Flight Management System publishes gate change notification events whenever a flight's arrival gate changes. Other systems, including Baggage Handler System. Inflight Catering System and Passenger Notifications System, must each asynchronously receive the same gate change notification to process the event according.

Which configuration is required in Anypoint MQ to achieve this publish/subscribe model?

Options:

A- Publish each client subscribe directly to the exchange.
Have each client subscribe directly to the queue.

B- Publish the gate change notification to an Anypoint MC queue
Have each client subscribe directly to the queue

C- Publish the gate change notification to an Anypoint MQ queue.
Create different anypoint MQ exchange meant for each of the other subscribing systems
Bind the queue with each of the exchanges

D- Publish the gate change notification to an Anypoint MQ exchanhe.
Create different Anypoint MQ queues meant for each of the other subscribing systems.
Bind the exchange with each of the queues.

Answer:

D

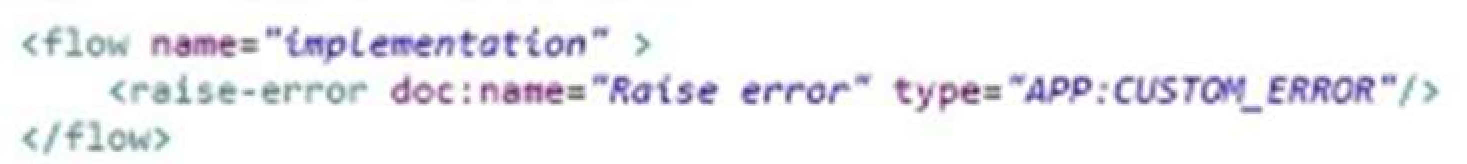
Explanation:

To achieve a publish/subscribe model using Anypoint MQ, where each system receives the same gate change notification event, the developer should publish the gate change notification to an Anypoint MQ exchange, create different Anypoint MQ queues meant for each of the other subscribing systems, and bind the exchange with each of the queues. An exchange is a message routing agent that can send messages to different queues based on predefined criteria. By binding an exchange with multiple queues, each queue receives a copy of every message sent to that exchange. Therefore, each system can subscribe to its own queue and receive every gate change notification event. Reference: https://docs.mulesoft.com/anypoint-mq/3.x/anypoint-mq-exchanges

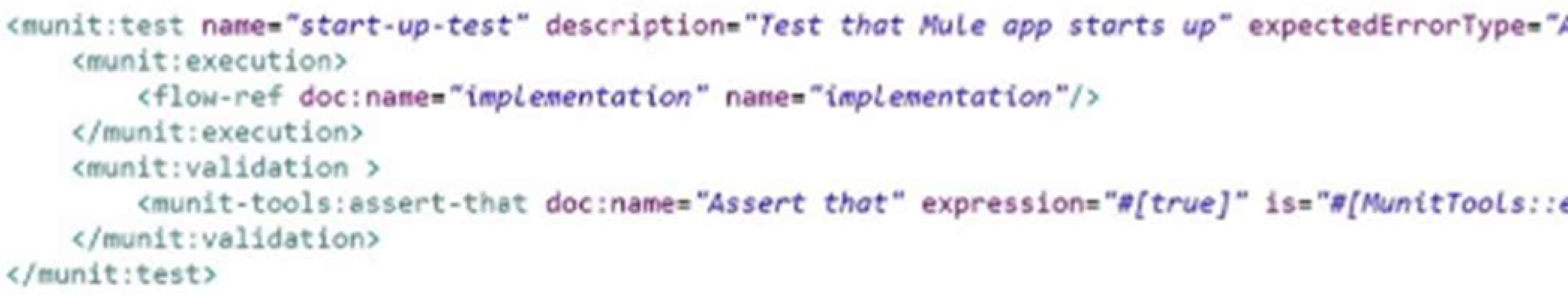
Question 5

Question Type: **MultipleChoice**

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="A
    <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::e
    </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?

Options:

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

Answer:

A

Explanation:

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. Reference: https://docs.mulesoft.com/munit/2.3/assert-equals-processor

Question 6

Question Type: MultipleChoice

The Center for Enablement team published a common application as a reusable module to the central Nexus repository.

How can the common application be included in all API implementations?

Options:

A- Download the common application from Naxus and copy it to the src/main/resources folder in the API

B- Copy the common application's source XML file and out it in a new flow file in the src/main/mule folder

C- Add a Maven dependency in the PCM file with multiple-plugin as <classifier>

D- Add a Maven dependency in the POM file with jar as <classifier>

Answer:

D

Explanation:

To include a common application as a reusable module in all API implementations, the developer should add a Maven dependency in the POM file with jar as <classifier>. This way, the developer can reuse Mule code from another application by packaging it as a JAR file and adding it as a dependency in the POM file of the API implementation. The classifier element specifies that it is a JAR file. Reference: https://docs.mulesoft.com/mule-runtime/4.3/mmp-concept#add-a-maven-dependency-to-the-pom-file

Question 7

Question Type: MultipleChoice

An order processing system is composed of multiple Mule application responsible for warehouse, sales and shipping. Each application communication using Anypoint MQ. Each message must be correlated against the original order ID for observability and tracing.

How should a developer propagate the order ID as the correlation ID across each message?

Options:

A- Use the underlying HTTP request of Anypoint MQ to set the 'X-CORRELATION_ID' header to the order ID

B- Set a custom Anypoint MQ user property to propagate the order ID and set the correlation ID in the receiving applications.

C- Use the default correlation ID, Anypoint MQ will sutomatically propagate it.

D- Wrap all Anypoint MQ Publish operations within a With CorrelationID scope from the Tracing module, setting the correlation ID to the order ID

Answer:

D

Explanation:

To propagate the order ID as the correlation ID across each message using Anypoint MQ, the developer should wrap all Anypoint MQ Publish operations within a With CorrelationID scope from the Tracing module, setting the correlation ID to the order ID. The With CorrelationID scope allows setting a custom correlation ID for any event that occurs within it. The Tracing module also enables distributed tracing across different Mule applications and services using Anypoint Monitoring. Reference:

https://docs.mulesoft.com/tracing-module/1.0/tracing-module-reference#with-correlation-id-scope https://docs.mulesoft.com/tracing-module/1.0/tracing-module-concepts

Question 8

Question Type: MultipleChoice

A company deploys 10 public APIs to CloudHub. Each API has its individual health endpoint defined. The platform operation team wants to configure API Functional Monitoring to monitor the health of the APIs periodically while minimizing operational overhead and cost.

How should API Functional Monitoring be configured?

Options:

A- From one public location with each API in its own schedule

B- From one private location with all 10 APIs in a single schedule

C- From one public location with all 10 APIs in a single schedule

D- From 10 public locations with each API in its own schedule

Answer:

C

Explanation:

To configure API Functional Monitoring to monitor the health of 10 public APIs periodically while minimizing operational overhead and cost, the developer should use one public location with all 10 APIs in a single schedule. A public location is a worker that runs in a CloudHub shared environment, which is cheaper and easier to maintain than a private location. A single schedule allows running all 10 APIs tests at the same time and frequency, which reduces complexity and resource consumption. Reference: https://docs.mulesoft.com/functional-monitoring/fm-create-monitor#create-a-monitor

Question 9

Question Type: MultipleChoice

A Mule implementation uses a HTTP Request within an Unit Successful scope to connect to an API.

How should a permanent error response like HTTP:UNAUTHORIZED be handle inside Until Successful to reduce latency?

Options:

A- Configure retrying until a MULERETRY_EXHAUSTED error is raised or the API responds back with a successful response.

B- In Until Successful configuration, set the retry count to 1 for error type HTTP: UNAUTHORIZED.

C- Put the HTTP Request inside a try scope in Unit Successful.
In the error handler, use On Error Continue to catch permanent errors like HTTP UNAUTHORIZED.

D- Put the HTTP Request inside a try scope in Unit Successful.
In the error handler, use On Error Propagate to catch permanent errors like HTTP UNAUTHORIZED.

Answer:

C

Explanation:

To handle a permanent error response like HTTP:UNAUTHORIZED inside Until Successful, the developer should put the HTTP Request inside a try scope in Unit Successful, and use On Error Continue to catch permanent errors like HTTP UNAUTHORIZED in the error handler. This way, the developer can avoid retrying requests that will always fail due to a permanent error, and reduce latency. On Error Continue allows the flow to continue processing after handling the error. Reference: https://docs.mulesoft.com/mule-runtime/4.3/until-successful-scope https://docs.mulesoft.com/mule-runtime/4.3/on-error-continue-concept

Question 10

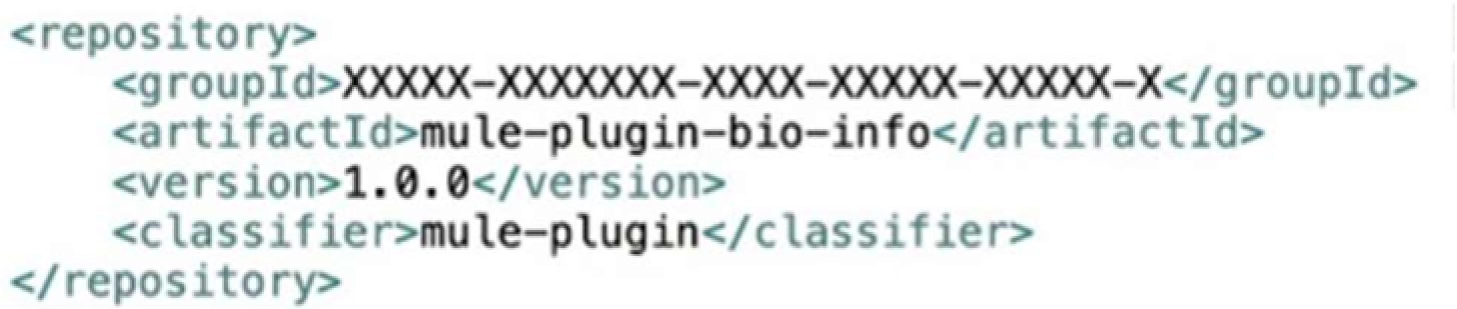
Question Type: MultipleChoice

Refer to the exhibits.

Bioinfo System API is implemented and published to Anypoint Exchange. A developer wants to invoke this API using its REST Connector.

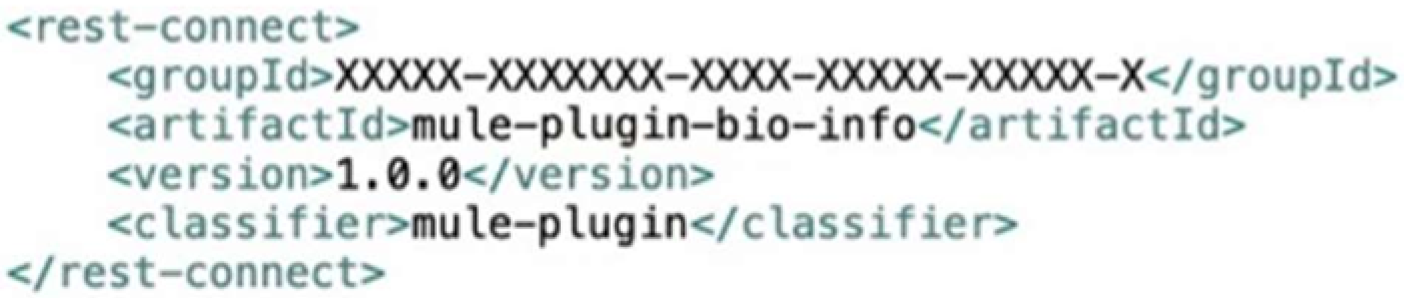
What should be added to the POM?

A)



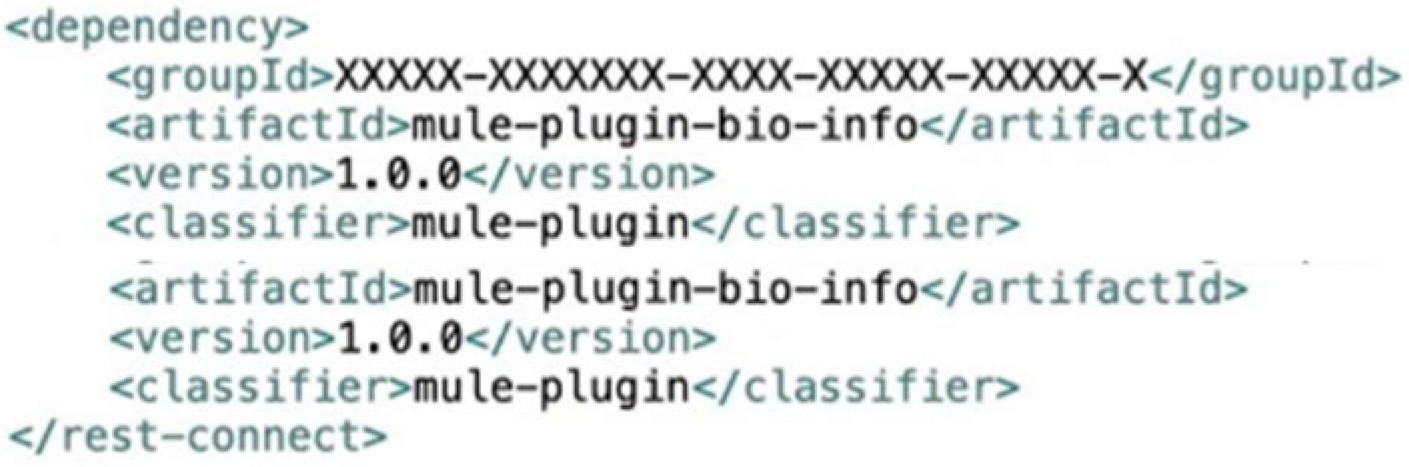
```
<repository>
    <groupId>XXXXX-XXXXXXX-XXXX-XXXXX-XXXXX-X</groupId>
    <artifactId>mule-plugin-bio-info</artifactId>
    <version>1.0.0</version>
    <classifier>mule-plugin</classifier>
</repository>
```

B)



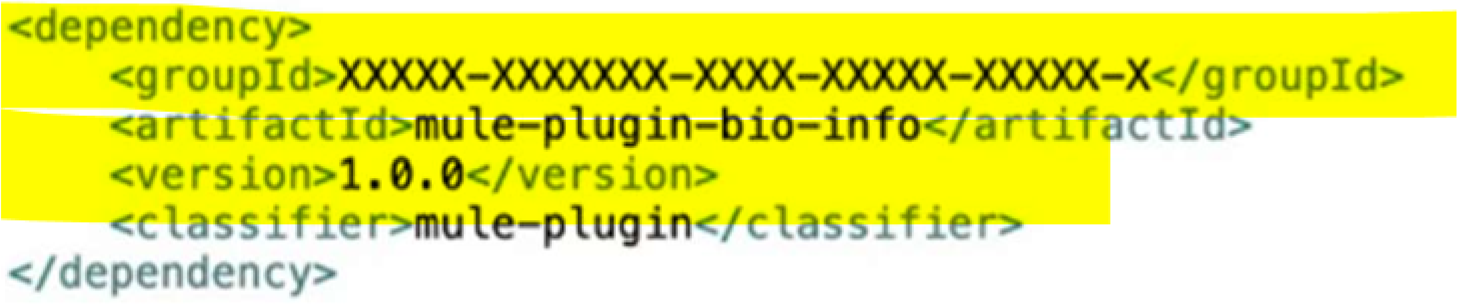
```
<rest-connect>
    <groupId>XXXXX-XXXXXXX-XXXX-XXXXX-XXXXX-X</groupId>
    <artifactId>mule-plugin-bio-info</artifactId>
    <version>1.0.0</version>
    <classifier>mule-plugin</classifier>
</rest-connect>
```

C)



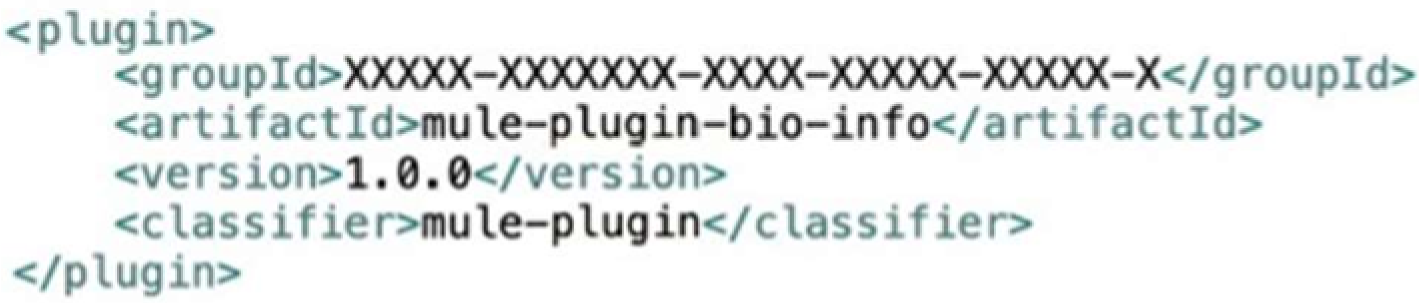
```
<dependency>
    <groupId>XXXXX-XXXXXXX-XXXX-XXXXX-XXXXX-X</groupId>
    <artifactId>mule-plugin-bio-info</artifactId>
    <version>1.0.0</version>
    <classifier>mule-plugin</classifier>
    <artifactId>mule-plugin-bio-info</artifactId>
    <version>1.0.0</version>
    <classifier>mule-plugin</classifier>
</rest-connect>
```

D)



```
<dependency>
    <groupId>XXXXX-XXXXXXX-XXXX-XXXXX-XXXXX-X</groupId>
    <artifactId>mule-plugin-bio-info</artifactId>
    <version>1.0.0</version>
    <classifier>mule-plugin</classifier>
</dependency>
```

E)



```
<plugin>
    <groupId>XXXXX-XXXXXXX-XXXX-XXXXX-XXXXX-X</groupId>
    <artifactId>mule-plugin-bio-info</artifactId>
    <version>1.0.0</version>
    <classifier>mule-plugin</classifier>
</plugin>
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

E- Option E

Answer:

E

Explanation:

To invoke Bioinfo System API using its REST Connector, option E should be added to the pom.xml file. This option adds a dependency for Bioinfo System API REST Connector with its group ID, artifact ID, version, classifier, and type. The classifier specifies that it is a REST Connector (raml-client), and the type specifies that it is a Mule plugin (mule-plugin). Reference: https://docs.mulesoft.com/apikit/4.x/apikit-4-generate-from-rest-api-task#add-the-api-dependency-to-the-pom-file

Question 11

Question Type: **MultipleChoice**

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

Options:

A- HTTP Webhock

B- Shared database trigger

C- Schedule Event Publisher

D- ETL data load

Answer:

A

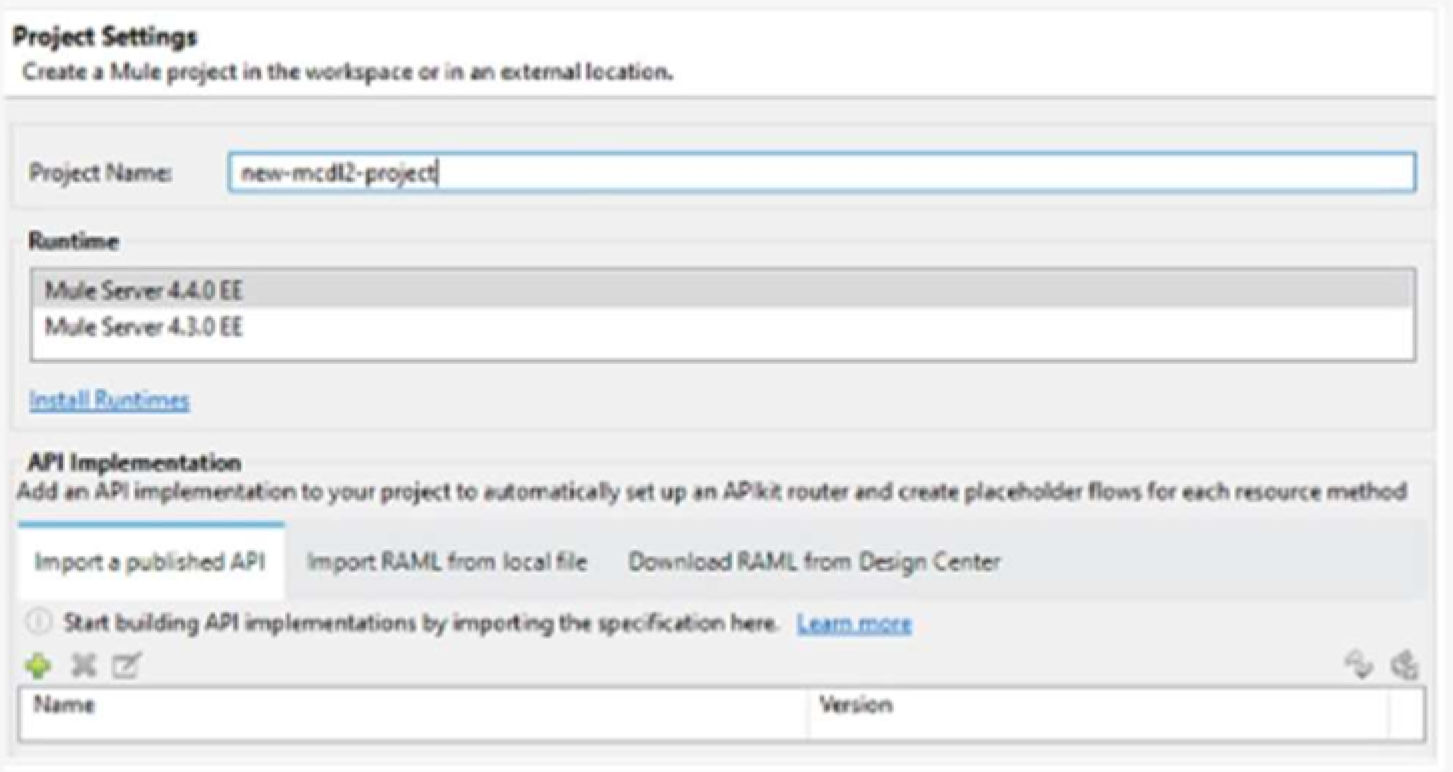
Explanation:

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. Reference: https://docs.mulesoft.com/connectors/webhook/webhook-connector

Question 12

Question Type: MultipleChoice

Refer to the exhibit.



Project Settings

Create a Mule project in the workspace or in an external location.

Project Name: new-mcdl2-project

Runtime

Mule Server 4.4.0 EE
Mule Server 4.3.0 EE

API Implementation

Add an API implementation to your project to automatically set up an APIkit router and create placeholder flows for each resource method

Start building API implementations by importing the specification here. Learn more

Name	Version

When creating a new project, which API implementation allows for selecting the correct API version and scaffolding the flows from the API specification?

Options:

A- Import a published API

B- Generate a local RAML from anypoint Studio

C- Download RAML from Design Center

D- Import RAML from local file

Answer:

C

Explanation:

To create a new project that selects the correct API version and scaffolds the flows from the API specification, the developer should import a published API. This option allows importing an API specification that has been published to Anypoint Exchange or Design Center, and selecting a specific version of that API specification. The developer can also choose to scaffold flows based on that API specification. Reference: https://docs.mulesoft.com/apikit/4.x/apikit-4-new-project-task