Salesforce.Mule-Dev-301.v2026-03-31.q29

Exam Code:	Mule-Dev-301
Exam Name:	Salesforce Certified MuleSoft Developer II
Certification Provider:	Salesforce
Free Question Number:	29
Version:	v2026-03-31
# of views:	109
# of Questions views:	337
https://www.freecram.net/torrent/Salesforce.Mule-Dev-301.v2026-03-31.q29.html	

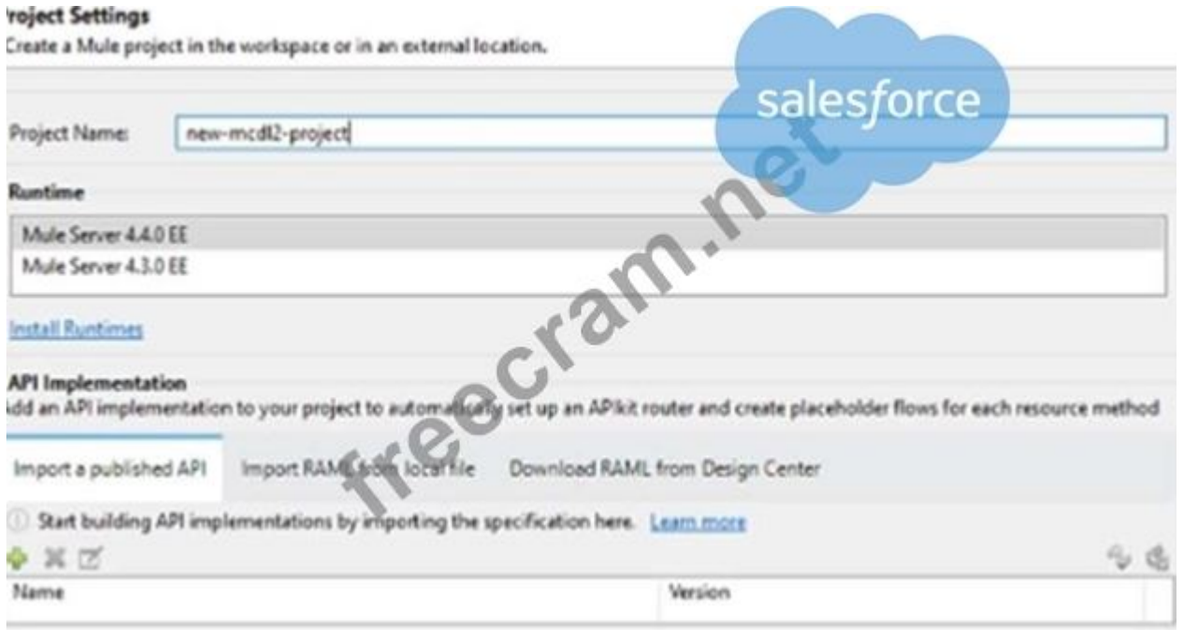
NEW QUESTION: 1
The Center for Enablement team published a common application as a reusable module to the central Nexus repository.
How can the common application be included in all API implementations?
A. Download the common application from Naxus and copy it to the src/main/resources folder in the 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: (SHOW ANSWER)
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

NEW QUESTION: 2
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?
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: (SHOW ANSWER)
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

NEW QUESTION: 3
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
Answer: (SHOW ANSWER)
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

NEW QUESTION: 4
Which plugin or dependency is required to unit test modules created with XML SDK?
A. XMLUnit
B. Junit
C. MUnit Extensions Maven plugin
D. MUnit Maven plugin
Answer: (SHOW ANSWER)
To unit test modules created with XML SDK, the developer needs to use the MUnit Extensions Maven plugin. This plugin allows testing XML SDK modules using MUnit by adding a dependency to the module under test and using a custom processor tag to invoke it. Reference: https://docs.mulesoft.com/mule-sdk/1.1/xml-sdk#testing

NEW QUESTION: 5
Refer to the exhibit.



salesforce
freecram.net
When creating a new project, which API implementation allows for selecting the correct API version and scaffolding the flows from the API specification?
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: (SHOW ANSWER)
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

NEW QUESTION: 6
An API has been built to enable scheduling email provider. The front-end system does very little data entry validation, and problems have started to appear in the email that go to patients. A validate-customer" flow is added validate the data.
What is he expected behavior of the 'validate-customer" flow?

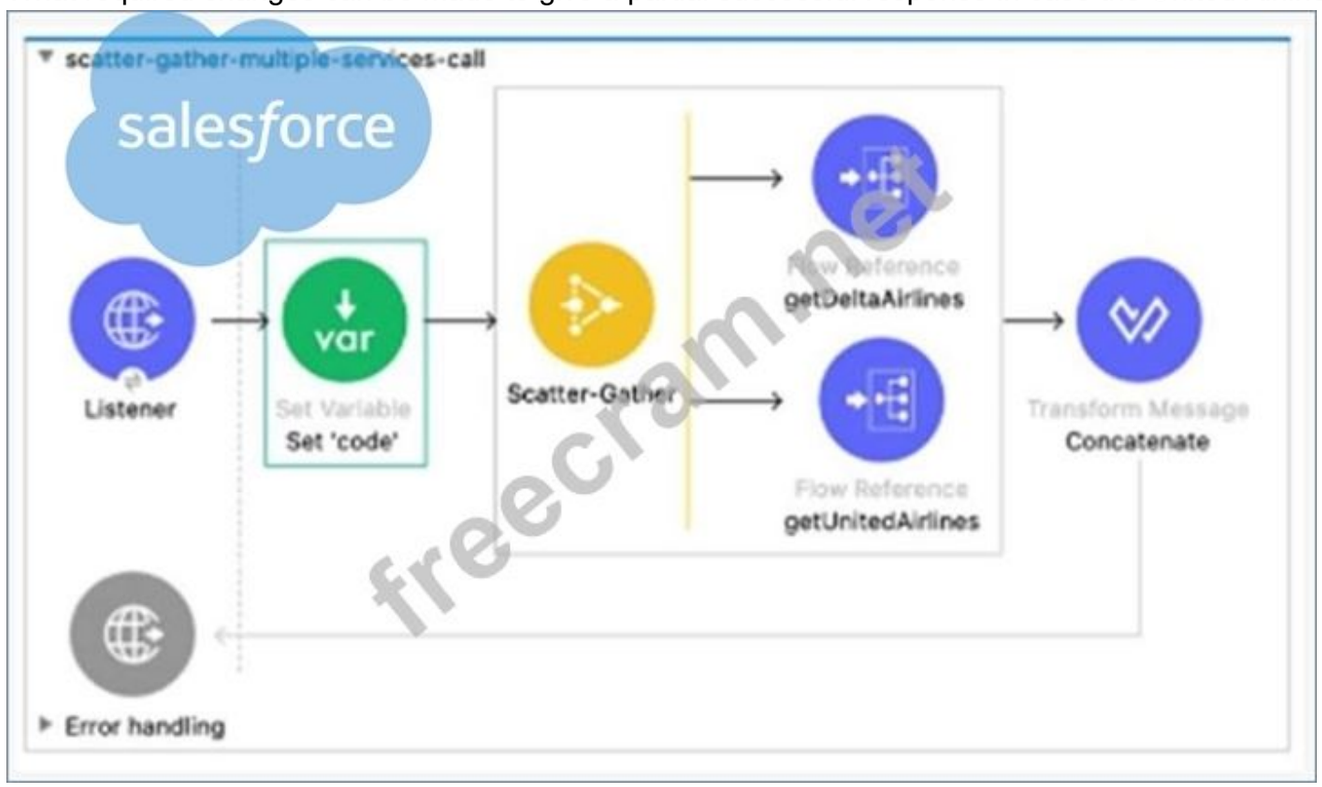
```
<flow name="validate-customer">
  <validation:all>
    <validation:is-email email="#[payload.customer.emailAddress]" message="invalid email address">
      <error-mapping sourceType="VALIDATION:INVALID_EMAIL" targetType="SCHEDULE:INVALID_EMAIL_ADDRESS"/>
    </validation:is-email>
    <validation:matches-regex value="#[payload.schedule.appointmentDate]"
      regex="^\d{4}-\d{2}-\d{2}$" message="Invalid appointment date">
      <error-mapping sourceType="VALIDATION:MISMATCH" targetType="SCHEDULE:INVALID_APPOINTMENT_DATE"/>
    </validation:matches-regex>
    <validation:is-not-null value="#[payload.customer.name]" message="Invalid customer name">
      <error-mapping sourceType="VALIDATION:NULL" targetType="SCHEDULE:INVALID_CUSTOMER_NAME"/>
    </validation:is-not-null>
  </validation:all>
</flow>
```

freecram.net
salesforce

A. If only the email address Is invalid a VALIDATION.INVALID_EMAIL error is raised
B. If the email address is invalid, processing continues to see if the appointment data and customer name are also invalid
C. If the appointment date and customer name are invalid, a SCHEDULE:INVALID_APPOINTMENT_DATE error is raised
D. If all of the values are invalid the last validation error is raised:SCHEDULE:INVALID_CUSTOMER_NAME
Answer: A (LEAVE A REPLY)
The validate-customer flow uses an until-successful scope to validate each field of the customer data. The until-successful scope executes its processors until they succeed or exhausts the maximum number of retries. If any processor fails, it raises an error and stops executing the remaining processors. Therefore, if only the email address is invalid, a VALIDATION.INVALID_EMAIL error is raised and the validation of appointment date and customer name is skipped. Reference: https://docs.mulesoft.com/mule-runtime/4.3/until-successful-scope

NEW QUESTION: 7
What is the MuleSoft recommended method to encrypt sensitive property data?
A. The encryption key and sensitive data should be different for each environment
B. The encryption key should be identical for all environments
C. The encryption key should be identical for all environments and the sensitive data should be different for each environment
D. The encryption key should be different for each environment and the sensitive data should be the same for all environments
Answer: A (LEAVE A REPLY)
The MuleSoft recommended method to encrypt sensitive property data is to use the Secure Properties Tool that comes with Anypoint Studio. This tool allows encrypting properties files with a secret key and then decrypting them at runtime using the same key. The encryption key and sensitive data should be different for each environment to ensure security and avoid accidental exposure of sensitive data. Reference: https://docs.mulesoft.com/mule-runtime/4.3/secure-configuration-properties

NEW QUESTION: 8
Refer to the exhibit.
What required changes can be made to give a partial successful response in case the United Airlines API returns with a timeout?



freecram.net

A. Add a Scatter-gather component inside a Try scope.
Set the payload to a default value 'Error' inside the error handler using the On Error Propagate scope.
B. Add Flow Reference components inside a Try scope.
Set the payload to a default value" insider the error handler using the ON Error Continue scope
C. Add Flow Reference components inside a Try scope
Set the payload to a default value " inside the error handler using the On Error Propagate scope
D. Add a Scatter-Gather component inside a Try scope.
Set the payload to a default value 'Error" inside the error handler using the On Error Continue scope.
Answer: (SHOW ANSWER)
To give a partial successful response in case the United Airlines API returns with a timeout, the developer should add a Scatter-Gather component inside a Try scope, and set the payload to a default value 'Error' inside the error handler using the On Error Continue scope. A Scatter-Gather component allows sending multiple requests concurrently and aggregating the responses into an array. A Try scope allows handling errors that occur within it using an error handler. An On Error Continue scope allows continuing the flow execution after handling an error. Therefore, by using these components, the developer can send requests to both APIs in parallel, handle any timeout errors from United Airlines API, and return a partial response with a default value for that API. Reference: https://docs.mulesoft.com/mule-runtime/4.3/scatter-gather-concept https://docs.mulesoft.com/mule-runtime/4.3/try-scope-concept https://docs.mulesoft.com/mule-runtime/4.3/on-error-continue-concept

NEW QUESTION: 9
Which properties are mandatory on the HTTP Connector configuration in order to use the OAuth 2.0 Authorization Code grant type for authentication?
A. External callback URL, access token URL, client ID response access token
B. Token URL, authorization URL, client ID, client secret local callback URL
C. External callback URL, access token URL, client ID, response refresh token

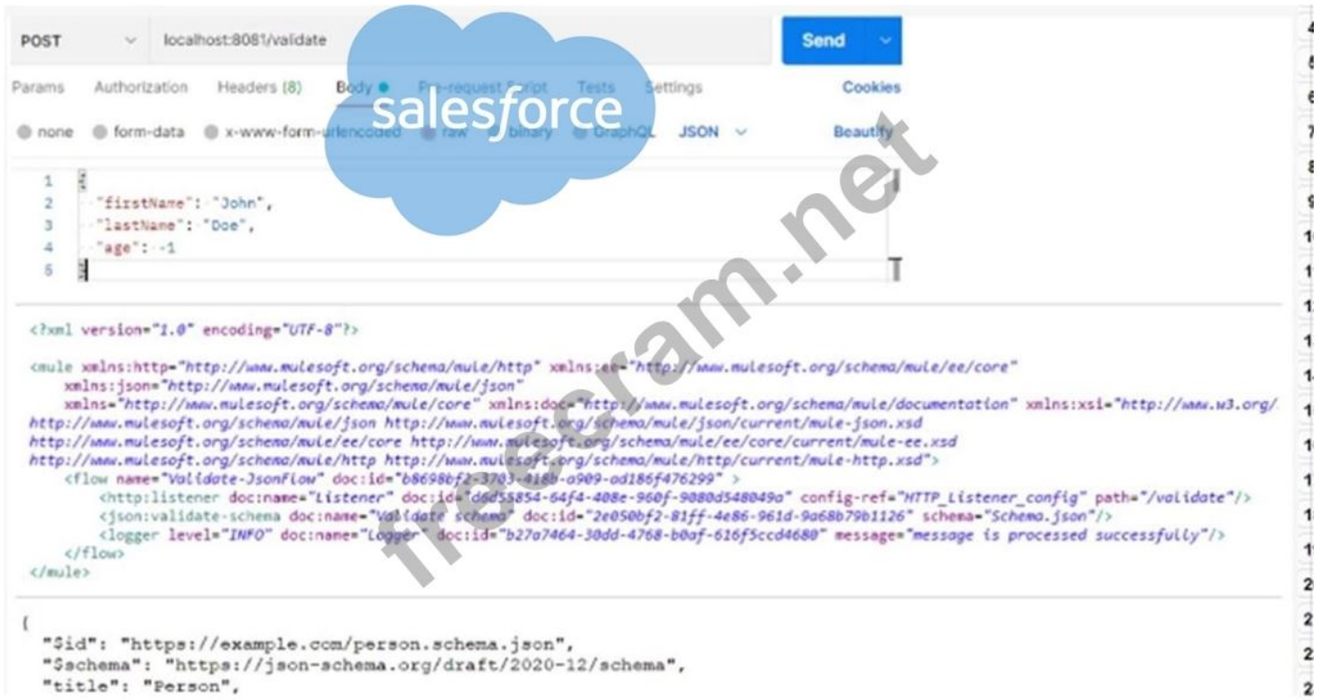
D. External callback URL, access token URL, local authorization URL, authorization URL, client ID, client secret

Answer: (SHOW ANSWER)

To use the OAuth 2.0 Authorization Code grant type for authentication, the HTTP Connector configuration requires the following properties: token URL, authorization URL, client ID, client secret, and local callback URL. The token URL is the endpoint of the authorization server that provides access tokens. The authorization URL is the endpoint of the authorization server that initiates the user consent flow. The client ID and client secret are the credentials of the Mule application registered with the authorization server. The local callback URL is the endpoint of the Mule application that receives the authorization code from the authorization server. Reference: https://docs.mulesoft.com/http-connector/1.6/http-authentication#oauth-2-0

NEW QUESTION: 10

Refer to the exhibit.



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?

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: (SHOW ANSWER)

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

NEW QUESTION: 11

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
Answer: (SHOW ANSWER)
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

NEW QUESTION: 12
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?
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: (SHOW ANSWER)
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

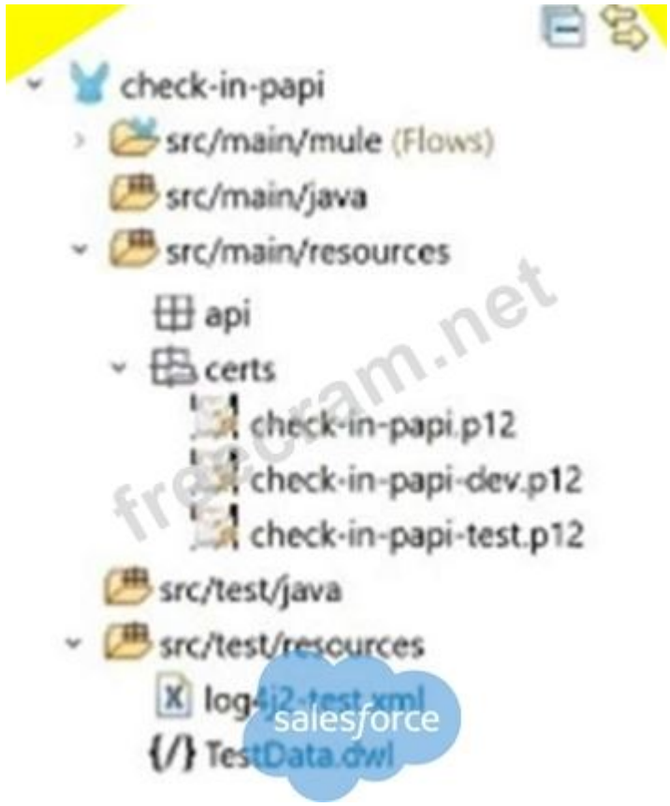
NEW QUESTION: 13
Refer to the exhibit.

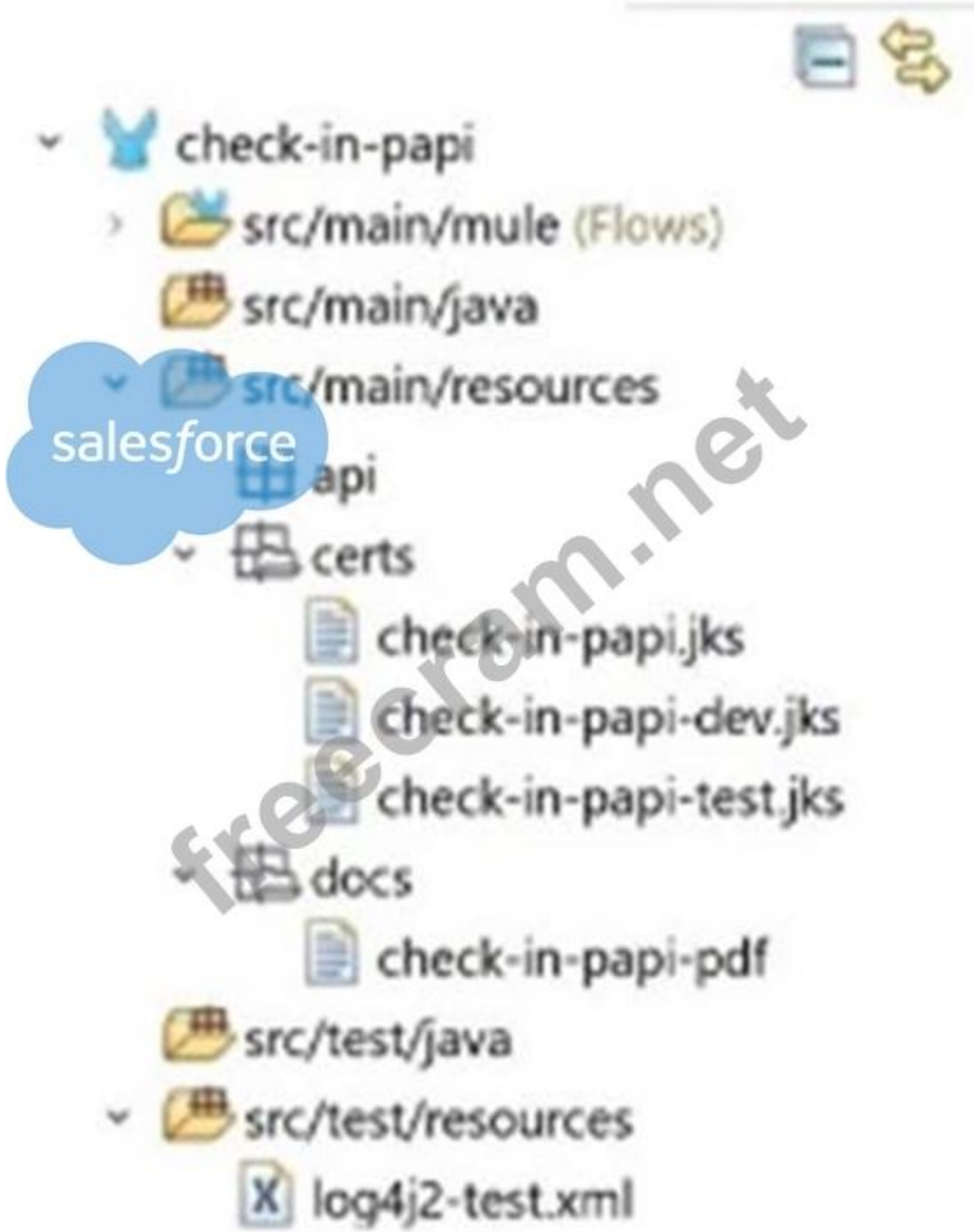
```
<build>
  <resources>
    <resource>
      <directory>src/main/resources</directory>
      <filtering>true</filtering>
    </resource>
  </resources>
  <testResources>
    <testResource>
      <directory>src/test/resources</directory>
      <filtering>true</filtering>
    </testResource>
    <testResource>
      <directory>src/test/funmon</directory>
      <filtering>true</filtering>
      <targetPath>funmon</targetPath>
    </testResource>
  </testResources>
  <pluginManagement>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-resources-plugin</artifactId>
        <configuration>
          <nonFilteredFileExtensions>
            <nonFilteredFileExtension>p12</nonFilteredFileExtension>
            <nonFilteredFileExtension>crt</nonFilteredFileExtension>
            <nonFilteredFileExtension>pe</nonFilteredFileExtension>
          </nonFilteredFileExtensions>
        </configuration>
      </plugin>
```

freecram.net

A Mule application pom.xml configures the Maven Resources plugin to exclude parsing binary files in the project's src/main/resources/certs directory.
Which configuration of this plugin achieves a successful build?

A.

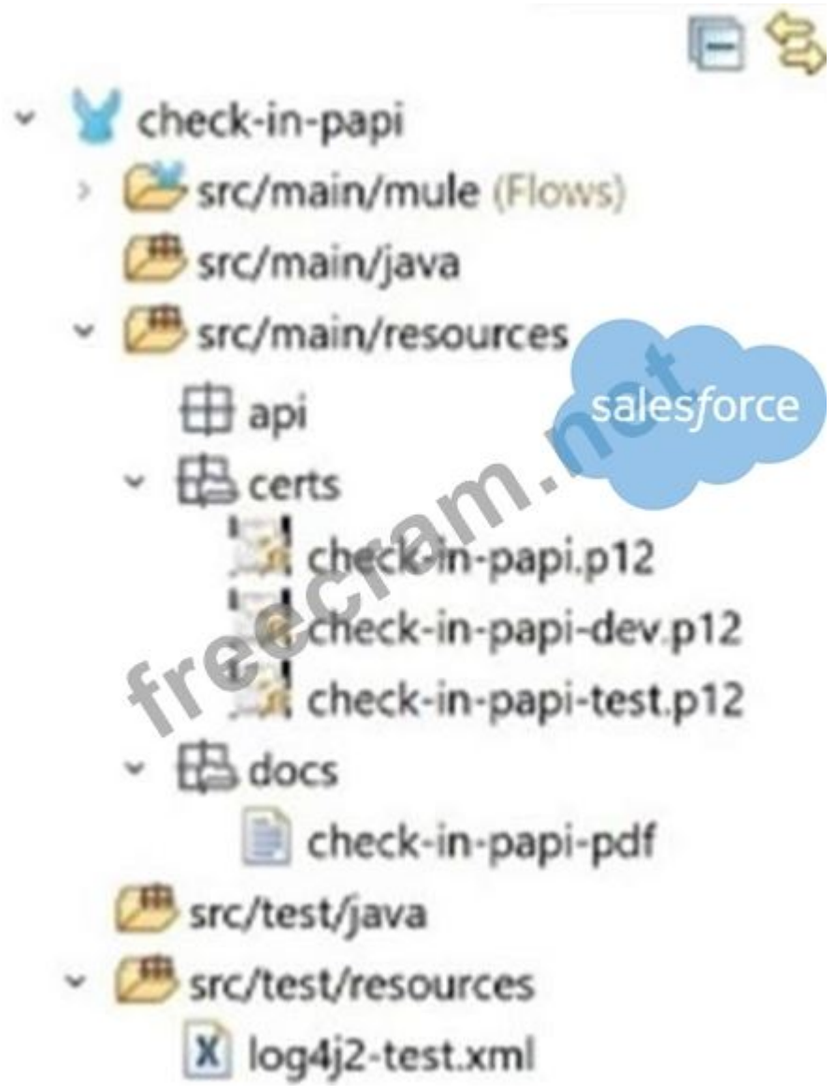

check-in-papi
src/main/mule (Flows)
src/main/java
src/main/resources
api
certs
check-in-papi.p12
check-in-papi-dev.p12
check-in-papi-test.p12
src/test/java
src/test/resources
log4j2-test.xml
TestData.dwl



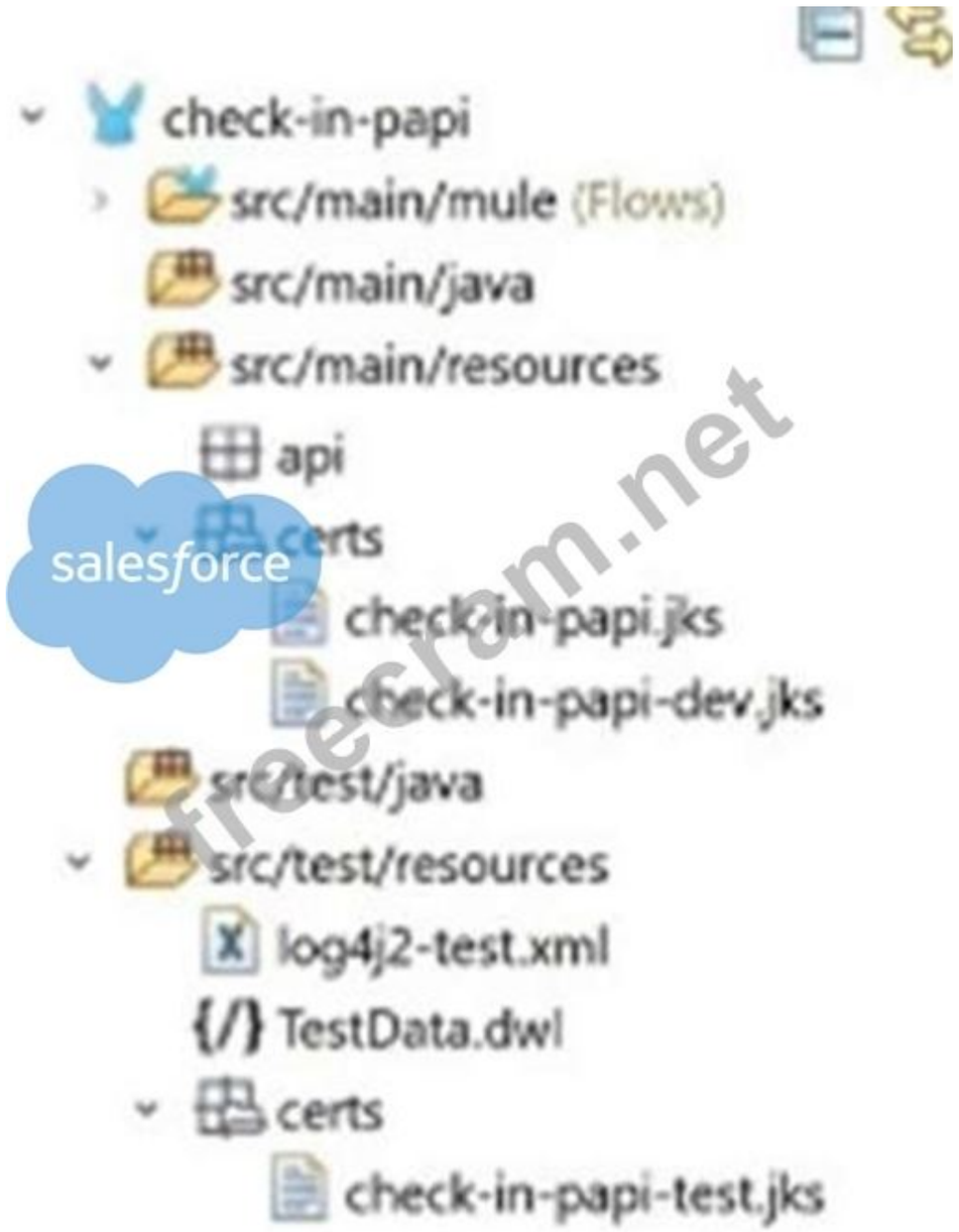
B.

freecram.net

C.



salesforce
freecram.net



D.

Answer: (SHOW ANSWER)

To configure the Maven Resources plugin to exclude parsing binary files in the project's src/main/resources/certs directory, option C should be used. This option specifies that any files with .cer or .jks extensions under the certs directory should be excluded from filtering. Filtering is a process of replacing placeholders with actual values in resource files during the build process. Binary files should not be filtered because they may become corrupted or unusable. Reference: https://maven.apache.org/plugins/maven-resources-plugin/examples/filter.html https://maven.apache.org/plugins/maven-resources-plugin/examples/include-exclude.html

NEW QUESTION: 14

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

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: (SHOW ANSWER)

freecram.net

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

NEW QUESTION: 15
Multiple individual Mute application need to use the Mule Maven plugin to deploy to CloudHub.
The plugin configuration should .. reused where necessary and anything project, specific should be property-based.
Where should the Mule Maven details be configured?
A. A parent pom.xml
B. Settings, xml
C. Pom, xml
D. A Bill of Materials (BOM) parent pm
Answer: (SHOW ANSWER)
To reuse Mule Maven plugin configuration across multiple individual Mule applications, the developer should use a parent pom.xml file. A parent pom.xml file defines common configuration for one or more child projects that inherit from it. The developer can specify common properties and dependencies for all child projects in the parent pom.xml file, such as Mule Maven plugin configuration, and then reference them in each child project's pom.xml file using placeholders. Reference: https://docs.mulesoft.com/mule-runtime/4.3/mmp-concept#parent-pom https://maven.apache.org/guides/introduction/introduction-to-the-pom.html#Project_Inheritance

NEW QUESTION: 16
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?
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: (SHOW ANSWER)
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

Valid Mule-Dev-301 Dumps shared by ExamDiscuss.com for Helping Passing Mule-Dev-301 Exam! ExamDiscuss.com now offer the **newest Mule-Dev-301 exam dumps**, the ExamDiscuss.com Mule-Dev-301 exam **questions have been updated** and **answers have been corrected** get the **newest** ExamDiscuss.com Mule-Dev-301 dumps with Test Engine here: https://www.examdiscuss.com/Salesforce/exam/Mule-Dev-301/premium/ (**62** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)

NEW QUESTION: 17
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: (SHOW ANSWER)
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. Reference: https://docs.mulesoft.com/mule-runtime/4.3/cache-scope#cache_invalidation

NEW QUESTION: 18
A custom policy needs to be developed to intercept all cutbound HTTP requests made by Mule applications.
Which XML element must be used to intercept outbound HTTP requests?
A. It is not possible to intercept outgoing HTTP requests, only inbound requests
B. http-policy:source
C. htt-policy:operation
D. http-policy:processor
Answer: (SHOW ANSWER)
The http-policy:processor element is used to intercept outbound HTTP requests made by Mule applications. It allows customizing the request before it is sent to the target API and modifying the response after it is received from the target API. Reference: https://docs.mulesoft.com/api-manager/2.x/policy-mule4-custom-policy#policy-xml-file

NEW QUESTION: 19
An organization uses CloudHub to deploy all of its applications.
How can a common-global-handler flow be configured so that it can be reused across all of the organization's deployed applications?
A. Create a Mule plugin project
Create a common-global-error-handler flow inside the plugin project.
Use this plugin as a dependency in all Mute applications.
Import that configuration file in Mute applications.
B. Create a common-global-error-handler flow in all Mule Applications Refer to it flow-ref wherever needed.
C. Create a Mule Plugin project
Answer: (SHOW ANSWER)
Create a common-global-error-handler flow inside the plugin project.
Use this plugin as a dependency in all Mule applications
D.

Create a Mule daman project.
Create a common-global-error-handler flow inside the domain project.
Use this domain project as a dependency.
Explanation:
To configure a common-global-handler flow that can be reused across all of the organization's deployed applications, the developer should create a Mule Plugin project, create a common-global-error-handler flow inside the plugin project, and use this plugin as a dependency in all Mule applications. This way, the developer can import the common-global-error-handler flow in any application that needs it and avoid duplicating the error handling logic. Reference: https://docs.mulesoft.com/mule-runtime/4.3/error-handling#global-error-handler

NEW QUESTION: 20
Which command is used to convert a JKS keystore to PKCS12?
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: (SHOW ANSWER)
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

NEW QUESTION: 21
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 reuses the same port number and HTTP Listener configuration as the API itself
B. Configure the application to send health data to an external system
C. Create a health check endpoint that listens on a separate port and uses a separate HTTP Listener configuration from the API
D. Monitor the Payment API directly sending real customer payment data
Answer: (SHOW ANSWER)

NEW QUESTION: 22
A Mule application defines as SSL/TLS keystore properly 'tis,keystore.keyPassword" as secure.
How can this property be referenced to access its value within the application?
A. #{secure::tiskeystore,keyPassowrd}
B. ${secure::tiskeystore,keyPassowrd}
C. ${secure::tiskeystore,keyPassowrd}
D. p{secure::tiskeystore,keyPassowrd}
Answer: B (LEAVE A REPLY)
secure::tiskeystore,keyPassowrd**ShortExplanationofCorrectAnswerOnly:Toreferenceasecurepropertyvaluewithintheapplication,thedeveloperneedstousethesyntax${secure::}.
In this case, the property name is tiskeystore,keyPassword, so the correct syntax is ${secure::tiskeystore,keyPassowrd}. Reference: https://docs.mulesoft.com/mule-runtime/4.3/secure-configuration-properties#referencing-secure-properties

NEW QUESTION: 23
When a client and server are exchanging messages during the mTLS handshake, what is being agreed on during the cipher suite exchange?
A. A protocol
B. The TLS version
C. An encryption algorithm
D. The Public key format
Answer: (SHOW ANSWER)
A cipher suite is a set of cryptographic algorithms that are used to secure the communication between a client and a server. A cipher suite consists of four components: a key exchange algorithm, an authentication algorithm, an encryption algorithm, and a message authentication code (MAC) algorithm. During the cipher suite exchange, the client and the server agree on which encryption algorithm to use for encrypting and decrypting the data. Reference: https://docs.mulesoft.com/mule-runtime/4.3/tls-configuration#cipher-suites

NEW QUESTION: 24
Which configurations are required for HTTP Listener to enable mTLS authentication?
A. Set an appropriate reconnection strategy and use persistent connections for the listener
B. Set an appropriate keystore configuration and use persistent connections for the listener
C. Set an appropriate keystore and truststore configuration for the listener
D. Set an appropriate truststore configuration and reconnection strategy for the listener
Answer: (SHOW ANSWER)
To enable mTLS authentication for HTTP Listener, the developer needs to set an appropriate keystore and truststore configuration for the listener. The keystore contains the certificate and private key of the Mule application that are used to prove its identity to clients. The truststore contains the certificates of trusted clients that are allowed to access the Mule application. Reference: https://docs.mulesoft.com/mule-runtime/4.3/tls-configuration#mutual-authentication

NEW QUESTION: 25
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
Answer: B (LEAVE A REPLY)
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

NEW QUESTION: 26
A Mule API receives a JSON payload and updates the target system with the payload. The developer uses JSON schemas to ensure the data is valid.
How can the data be validation before posting to the target system?
A. Use a DataWeave 2.09 transform operation, and at the log of the DataWeave script, add:
%dw 2.0

Import.json-moduls
B. Using the DataWeave if Else condition test the values of the payload against the examples included in the schema
C. Apply the JSON Schema policy in API Manager and reference the correct schema in the policy configuration
D. Add the JSON module dependency and add the validate-schema operation in the flow, configured to reference the schema
Answer: D (LEAVE A REPLY)
To validate the data before posting to the target system, the developer should add the JSON module dependency and add the validate-schema operation in the flow, configured to reference the schema. The JSON module provides a validate-schema operation that validates a JSON payload against a JSON schema and throws an error if the payload is invalid. Reference: https://docs.mulesoft.com/json-module/1.1/json-validate-schema

NEW QUESTION: 27
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
Answer: D (LEAVE A REPLY)
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/4.3/http-listener-ref#callback

NEW QUESTION: 28
A developer deploys an API to CloudHub and applies an OAuth policy on API Manager. During testing, the API response is slow, so the developer reconfigures the API so that the out-of-the-box HTTP Caching policy is applied first, and the OAuth API policy is applied second.
What will happen when an HTTP request is received?
A. In case of a cache hit, both the OAuth and HTTP Caching policies are evaluated; then the cached response is returned to the caller
B. In case of a cache it, only the HTTP Caching policy is evaluating; then the cached response is returned to the caller
C. In case of a cache miss, only the HTTP Caching policy is evaluated; then the API retrieves the data from the API implementation, and the policy stores the data to be cached in Object Store
D. In case of a cache miss, both the OAuth and HTTP Caching policies are evaluated; then the API retrieves the data from the API implementation, and the policy does not store the data in Object Store
Answer: (SHOW ANSWER)
When an HTTP request is received and the HTTP Caching policy is applied first, it checks if there is a cached response for that request in Object Store. If there is a cache hit, meaning that a valid cached response exists, then only the HTTP Caching policy is evaluated and the cached response is returned to the caller without invoking the OAuth policy or the API implementation. If there is a cache miss, meaning that no valid cached response exists, then both the HTTP Caching policy and the OAuth policy are evaluated before invoking the API implementation. Reference: https://docs.mulesoft.com/api-manager/2.x/http-caching-policy#policy-ordering

NEW QUESTION: 29
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
```

freecram.net

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?

A. HTTP status code:200

B. HTTP status code:403

C. HTTP status code:400

D. HTTP status code:500

Answer: (SHOW ANSWER)

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

Valid Mule-Dev-301 Dumps shared by ExamDiscuss.com for Helping Passing Mule-Dev-301 Exam! ExamDiscuss.com now offer the **newest Mule-Dev-301 exam dumps**, the ExamDiscuss.com Mule-Dev-301 exam **questions have been updated** and **answers have been corrected** get the **newest** ExamDiscuss.com Mule-Dev-301 dumps with Test Engine here: https://www.examdiscuss.com/Salesforce/exam/Mule-Dev-301/premium/ (**62** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)