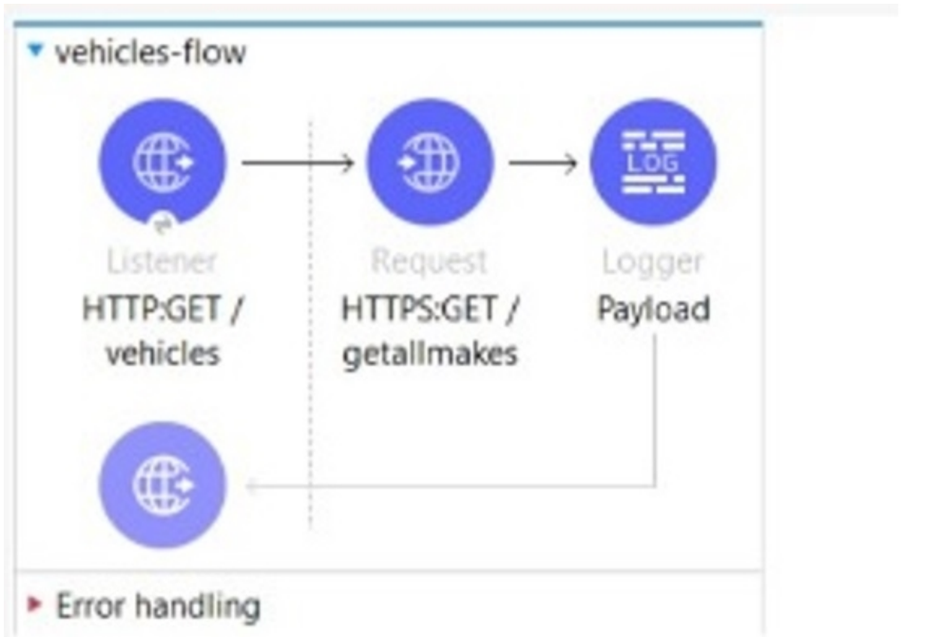
Number: Certified MuleSoft Developer II
Passing Score: 800
Time Limit: 120
File Version: 3.2

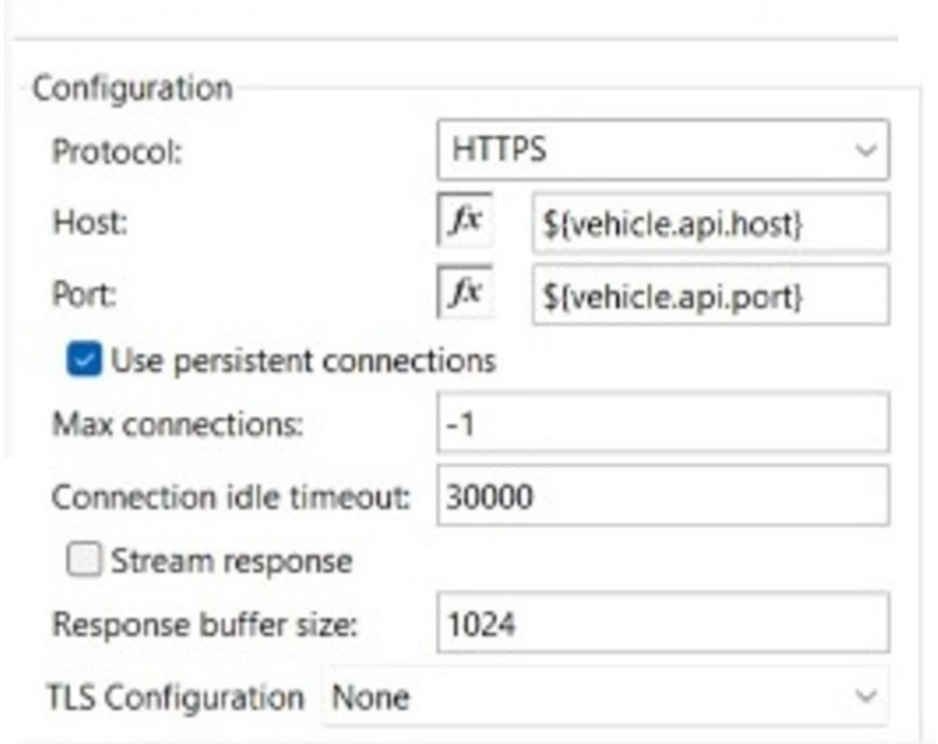
Exam Code: Certified MuleSoft Developer II

Exam Name: Certified MuleSoft Developer II

Exam A

QUESTION 1
The flow is invoicing a target API. The API's protocol is HTTPS. The TLS configuration in the HTTP Request Configuration global element is set to None. A web client submits a request to http:localhost:8081/vehicles.





If the certificate of the target API is signed by a certificate authority (CA), what is true about the HTTP Request operation when the flow executes?

A. The HTTP Request operation will succeed if the CA'S certificate is present in the JRE's default keystore
B. The HTTP Request operation will succeed if the CA's certificate is present in the JRE's default truststore.
C. The HTTP Request operation will always succeed regardless of the CA
D. The HTTP Request operation will always fail regardless of the CA

Correct Answer: B
Section:
Explanation:
The HTTP Request operation will use the default truststore of the JRE to validate the certificate of the target API. If the CA's certificate is present in the truststore, the operation will succeed. Otherwise, it will fail with a handshake exception.
Reference: https://docs.mulesoft.com/mule-runtime/4.3/tls-configuration#tls-default

QUESTION 2
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

Correct Answer: C
Section:
Explanation:
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

QUESTION 3
A healthcare portal needs to validate the token that it sends to a Mule API. The developer plans to implement a custom policy using the HTTP Policy Transform Extension to match the token received in the header from the heathcare portal.
Which files does the developer need to create in order to package the custom policy?

A. Deployable ZIP file, YAML configuration file
B. JSON properties file, YAML configuration file
C. JSON properties file, XML template file
D. XML template file, YAML configuration file

Correct Answer: D
Section:
Explanation:
To package a custom policy using the HTTP Policy Transform Extension, the developer needs to create an XML template file and a YAML configuration file. The XML template file defines the policy logic using Mule components and placeholders for user-defined properties. The YAML configuration file defines the metadata of the policy, such as its name, description, category, parameters, and dependencies.
Reference: https://docs.mulesoft.com/api-manager/2.x/http-policy-transform#packaging-the-policy

QUESTION 4
Refer to the exhibit.
What action must be performed to log all the errors raised by the VM Connector?

A. Add <AsyncLOgger name='orgroute.extensions vm' level=ERROR'I> inside the Logger tag
B. Add <AsyncLOgger name='orgroute.extensions vm' level=ERROR'/> inside the Appenders tag
C. Configure <Logger level-'ERROR'/> inside the VM Connector configuration
D. Nothing, as error-level events are automatically logged

Correct Answer: B
Section:
Explanation:
To log all the errors raised by the VM Connector, the developer needs to add an async logger with the name 'org.mule.extension.vm' and the level 'ERROR' inside the appenders tag of the log4j2.xml file. This will enable logging all error-level events generated by the VM Connector to the console appender.
Reference: https://docs.mulesoft.com/mule-runtime/4.3/logging-in-mule#configuring-custom-logging-settings

QUESTION 5
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

Correct Answer: B
Section:
Explanation:
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

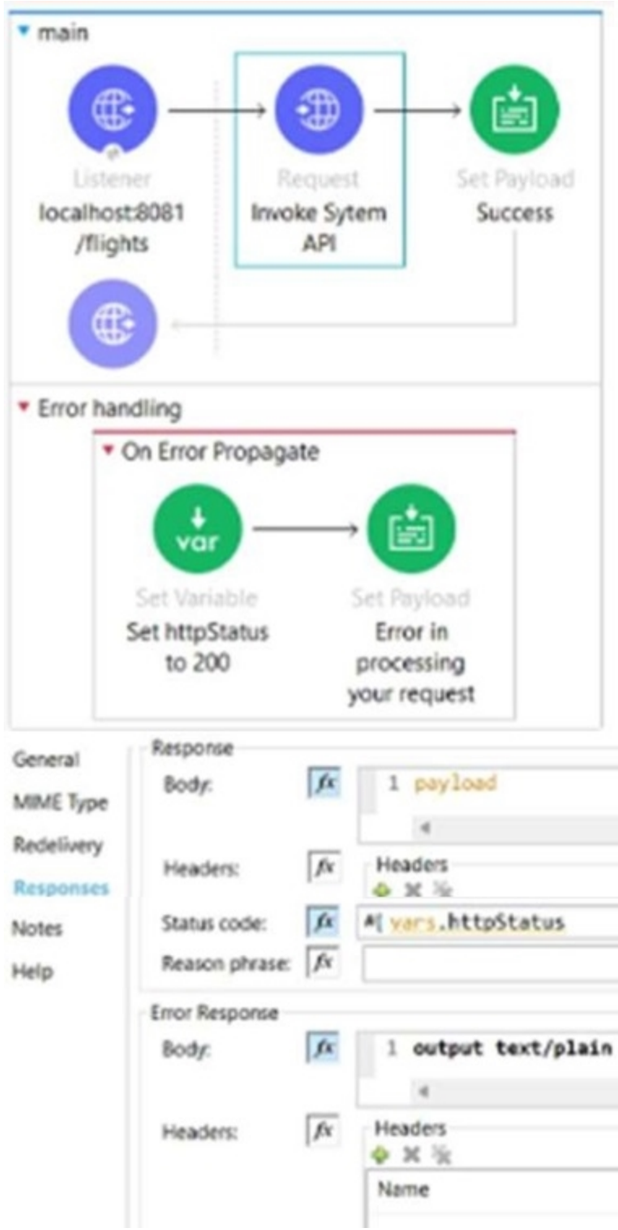
QUESTION 6
A system API that communicates to an underlying MySQL database is deploying to CloudHub. The DevOps team requires a readiness endpoint to monitor all system APIs.
Which strategy should be used to implement this endpoint?

A. Create a dedicated endpoint that responds with the API status and reachability of the underlying systems

B. Create a dedicated endpoint that responds with the API status and health of the server
C. Use an existing resource endpoint of the API
D. Create a dedicated endpoint that responds with the API status only

Correct Answer: A
Section:
Explanation:
To implement a readiness endpoint to monitor all system APIs, the developer should create a dedicated endpoint that responds with the API status and reachability of the underlying systems. This way, the DevOps team can check if the system API is ready to receive requests and if it can communicate with its backend systems without errors.
Reference: https://docs.mulesoft.com/mule-runtime/4.3/deployment-strategies#readiness-probes

QUESTION 7
The HTTP Request operation raises an HTTP CONNECTIVITY error.
Which HTTP status code and body are returned to the web client?



A. HTTP Status Code:200. Body 'Error in processing your request
B. HTTP Status Code:500. Body 'The HTTP CONNECTIVITY Error description
C. HTTP Status Code:500. Body 'Error in processing your request
D. HTTP Status Code:500. Body 'Error in processing your request

Correct Answer: C
Section:
Explanation:
When the HTTP Request operation raises an HTTP CONNECTIVITY error, it triggers an on-error-continue handler that sets a payload with 'Error in processing your request'. Since no status code is explicitly set in this handler, it defaults to 500 (INTERNAL SERVER ERROR). Therefore, the web client receives an HTTP response with status code 500 and body 'Error in processing your request'.

Reference: https://docs.mulesoft.com/mule-runtime/4.3/error-handling#on-error-continue

QUESTION 8
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}

Correct Answer: B
Section:
Explanation:
secure::tiskeystore,keyPassowrdShortExplanationofCorrectAnswerOnly: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

QUESTION 9
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

Correct Answer: B
Section:
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.
Reference: https://docs.mulesoft.com/munit/2.3/munit-test-flow

QUESTION 10
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

Correct Answer: B
Section:
Explanation:
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

QUESTION 11
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>
```

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

Correct Answer: A
Section:
Explanation:
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

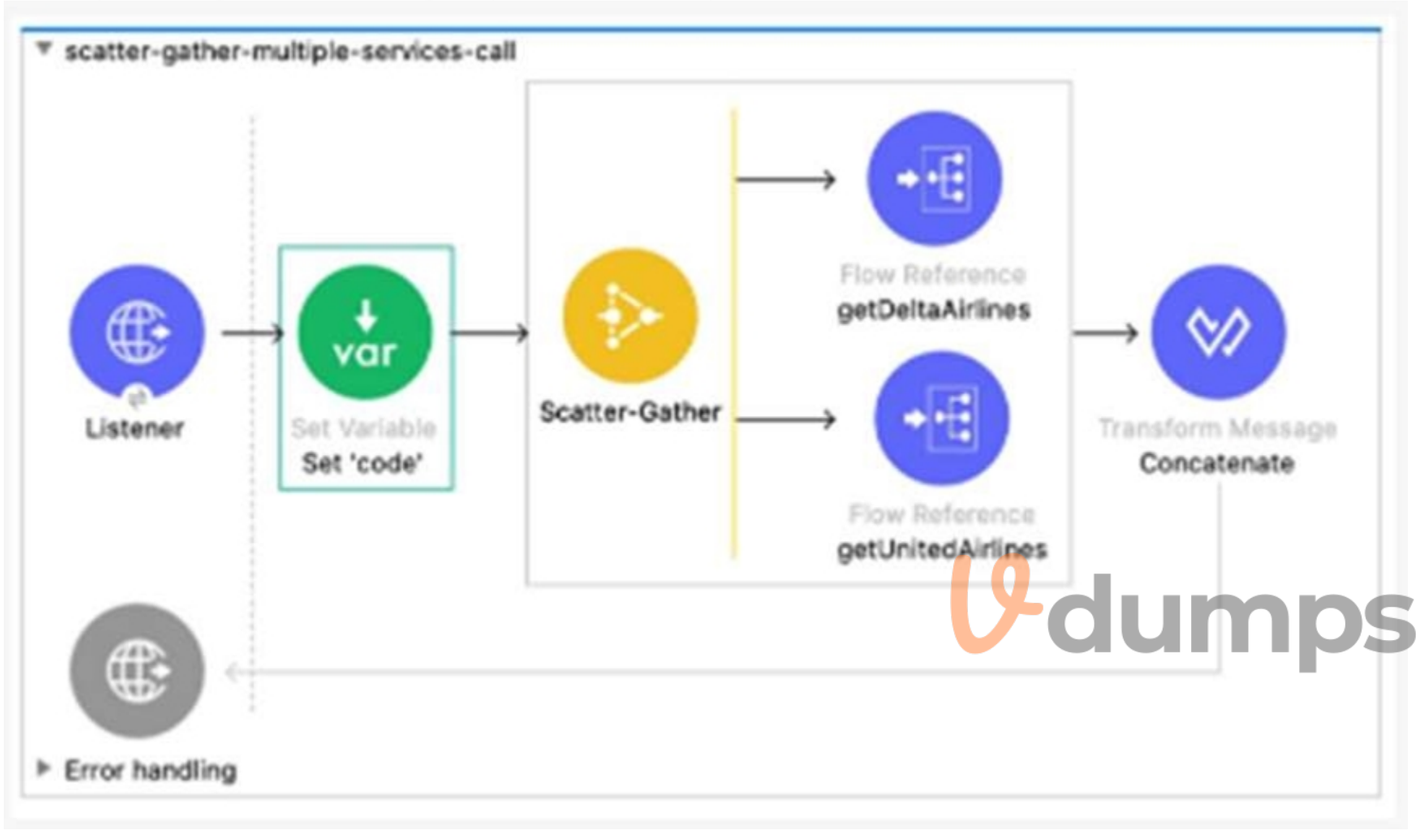
QUESTION 12
Multiple individual Mute application need to use the Mule Maven plugin to deploy to CloudHub.
The plugin configuration should .. reused where necessary and anything project, specific should be property-based.
Where should the Mule Maven details be configured?

A. A parent pom.xml
B. Settings, xml
C. Pom, xml
D. A Bill of Materials (BOM) parent pm

Correct Answer: A
Section:
Explanation:
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

QUESTION 13
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?



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.

Correct Answer: D
Section:
Explanation:
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

QUESTION 14
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
Section:
Explanation:
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.
Reference: https://docs.mulesoft.com/api-manager/2.x/policies-policy-order

QUESTION 15
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

Correct Answer: D
Section:
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 16
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
Section:
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 17
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
Section:
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

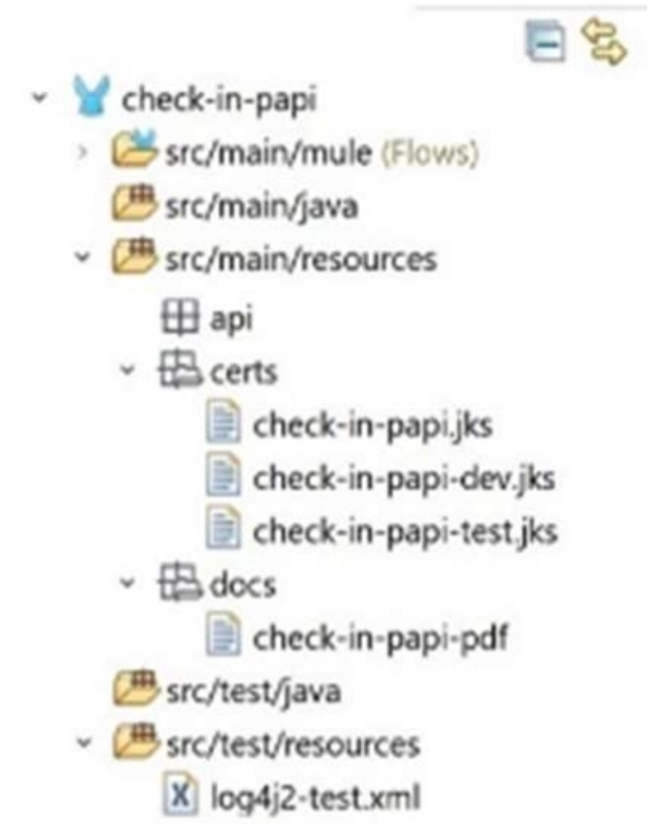
QUESTION 18
Refer to the exhibit.

```
<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>pem</nonFilteredFileExtension>
          </nonFilteredFileExtensions>
        </configuration>
      </plugin>
```

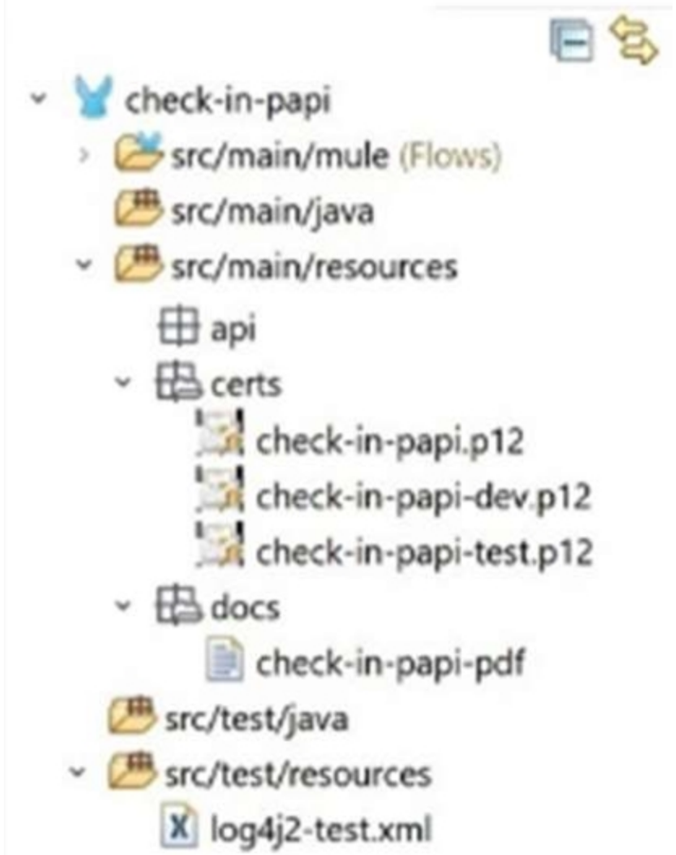
A Mule application pom.xml configures the Maven Resources plugin to exclude parsing binary files in the project's src/main/resources/certs directory.
Which configuration of this plugin achieves a successful build?
A)


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

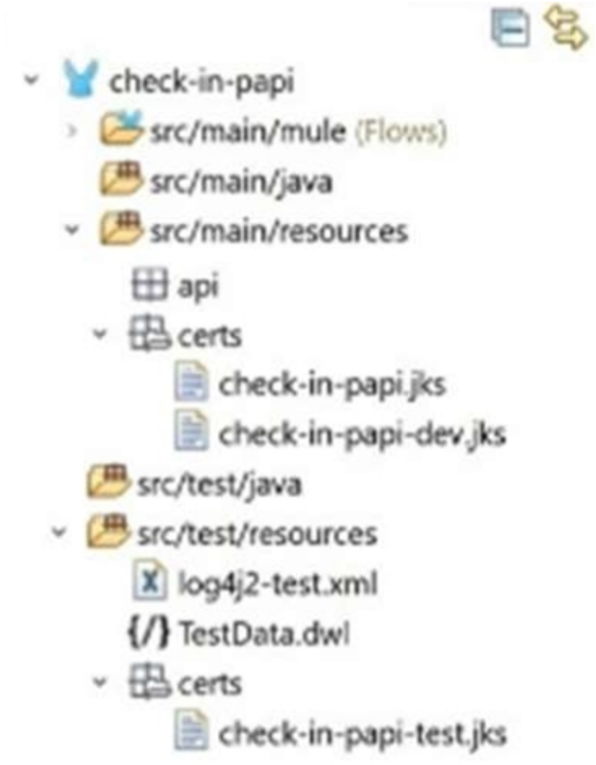
B)


check-in-papi
src/main/mule (Flows)
src/main/java
src/main/resources
api
certs
check-in-papi.jks
check-in-papi-dev.jks
check-in-papi-test.jks
docs
check-in-papi-pdf
src/test/java
src/test/resources
log4j2-test.xml

C)


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
docs
check-in-papi-pdf
src/test/java
src/test/resources
log4j2-test.xml

D)



A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: C
Section:
Explanation:
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

QUESTION 19
When implementing a synchronous API where the event source is an HTTP Listener, a developer needs to return the same correlation ID back to the caller in the HTTP response header.
How can this be achieved?

A. Enable the auto-generate CorrelationID option when scaffolding the flow
B. Enable the CorrelationID checkbox in the HTTP Listener configuration
C. Configure a custom correlation policy
D. NO action is needed as the correlation ID is returned to the caller in the response header by default

Correct Answer: D

Section:
Explanation:
When implementing a synchronous API where the event source is an HTTP Listener, Mule automatically propagates some message attributes between flows via outbound and inbound properties. One of these attributes is correlation ID, which is returned to the caller in the response header by default as MULE_CORRELATION_ID.
Reference: https://docs.mulesoft.com/mule-runtime/4.3/about-mule-message#message-attributes

QUESTION 20
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

Correct Answer: C
Section:
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.
Reference: https://docs.mulesoft.com/mule-runtime/4.3/cache-scope#cache_invalidation

QUESTION 21
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

Correct Answer: A
Section:
Explanation:
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

QUESTION 22
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
Section:
Explanation:

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

QUESTION 23
Refer to the exhibit.
A Mute Object Store is configured with an entry TTL of one second and an expiration interval of 30 seconds.
What is the result of the flow if processing between os'store and os:retrieve takes 10 seconds?

```
<os:object-store name="os" entryTtl="1" entryTtlUnit="SECONDS"
  expirationInterval="30" expirationIntervalUnit="SECONDS"/>

<flow name="main-flow">
  <set-payload value="originalPayload" />
  <os:store objectStore="os" key="#['testKey']">
    <os:value><![CDATA[#["testPayload"]]]></os:value>
  </os:store>
  <os:retrieve objectStore="os" key="#['testKey']">
    <os:default-value>#['nullPayload']</os:default-value>
  </os:retrieve>
</flow>
```

A. nullPayload
B. originalPayload
C. OS:KEY_NOT_FOUND
D. testPayload

Correct Answer: A
Section:
Explanation:
The result of the flow is nullPayload if processing between os:store and os:retrieve takes 10 seconds. This is because the entry TTL of the object store is one second, which means that any stored value expires after one second and is removed from the object store. The expiration interval of 30 seconds only determines how often the object store checks for expired values, but it does not affect the TTL. Therefore, when os:retrieve tries to get the value after 10 seconds, it returns nullPayload because the value has already expired and been removed.
Reference: https://docs.mulesoft.com/object-store/osv2-faq#how-does-the-time-to-live-work

QUESTION 24
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

Correct Answer: C
Section:
Explanation:
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

QUESTION 25
Which statement is true when working with correlation IDS?

A. The HTTP Listener regenerates correlation IDs regardless of the HTTP request
B. The Anypoint MQ Connector automatically propagates correlation IDS
C. The HTTP Listener generates correlation IDS unless a correlation ID is received in the HTTP request
D. The VM Connector does not automatically propagate correction IDs

Correct Answer: C
Section:
Explanation:
When working with correlation IDs, the HTTP Listener generates correlation IDs unless a correlation ID is received in the HTTP request. In that case, it propagates the received correlation ID throughout the flow execution. Correlation IDs are used to track events across different flows or applications.
Reference: https://docs.mulesoft.com/mule-runtime/4.3/about-mule-message#message-attributes

QUESTION 26
Which statement is true about using mutual TLS to secure an application?

A. Mutual TLS requires a hardware security module to be used
B. Mutual TLS authenticates the identity of the server before the identity of the client
C. Mutual TLS ensures only authorized end users are allowed to access an endpoint
D. Mutual TLS increases the encryption strength versus server-side TLS alone

Correct Answer: B
Section:
Explanation:
Mutual TLS (mTLS) is an extension of TLS that requires both parties (client and server) to present their certificates to each other during the handshake process. This way, both parties can verify each other's identity and establish a secure connection. The authentication of the server happens before the authentication of the client, as the server sends its certificate first and then requests the client's certificate.
Reference: https://docs.mulesoft.com/mule-runtime/4.3/tls-configuration#mutual-authentication

QUESTION 27
Which statement is true when using XML SDK for creating custom message processors?

A. Properties are fields defined by an end user of the XML SDK component and serve as a global configuration for the entire Mule project in which they are used
B. An XML SDK provides both inbound and outbound operations
C. Operations can be reused in recursive calls
D. All operations are public

Correct Answer: A
Section:
Explanation:
When using XML SDK for creating custom message processors, all operations are public by default and can be used by any Mule application that imports them. There is no way to make an operation private or protected in XML SDK.
Reference: https://docs.mulesoft.com/mule-sdk/1.1/xml-sdk#operations

QUESTION 28
Refer to the exhibit.
What is the result of the Mule Maven Plugin configuration of the value of property its,keystorePassoword in CloudHub 2.0?

```
<secureProperties>
    <tls.keyStore.password>${tls.keyStore.password}</tls.keyStore.password>
</secureProperties>
```

A. CloudHub encrypts the value
B. The Mule server encrypts the value
C. Anypoint Studio secures the value
D. Runtime Manager masks the value

Correct Answer: D
Section:
Explanation:
The result of the Mule Maven Plugin configuration of the value of property its,keystorePassword in CloudHub 2.0 is that Runtime Manager masks the value. This means that Runtime Manager hides or obscures the value from anyone who views it in Runtime Manager or Anypoint Platform.
Reference: https://docs.mulesoft.com/runtime-manager/runtime-manager-agent-for-mule4#properties-tab

QUESTION 29
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 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.

Correct Answer: C
Section:
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