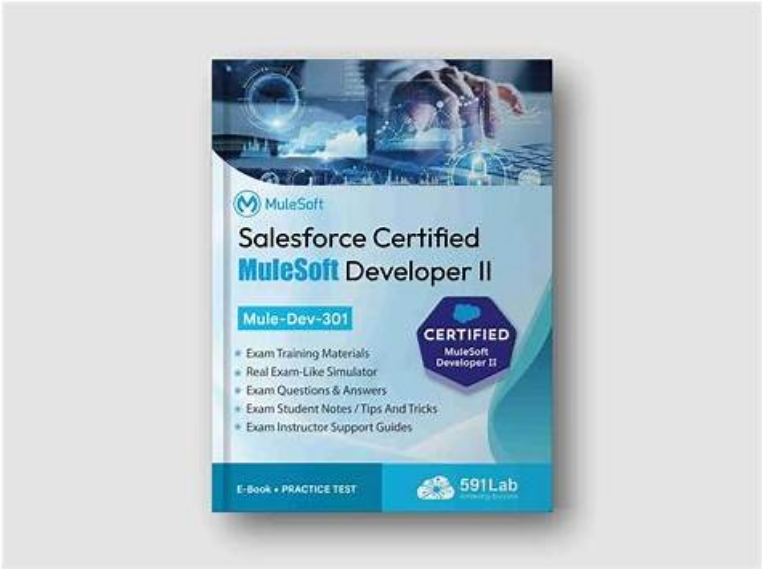
Mule-Dev-301 Training Pdf, Latest Mule-Dev-301 Mock Exam



BTW, DOWNLOAD part of Actualtests4sure Mule-Dev-301 dumps from Cloud Storage: https://drive.google.com/open?id=1je2t2T2AHEWme1qLlSFXWrTNMEedn1Jx

If you are preparing for the Salesforce Mule-Dev-301 exam dumps our Mule-Dev-301 Questions help you to get high scores in your Salesforce Mule-Dev-301 exam. Test your knowledge of the Salesforce Mule-Dev-301 Exam Dumps with Actualtests4sure Salesforce Mule-Dev-301 practice questions. The software is designed to help with Salesforce Mule-Dev-301 exam dumps preparation.

Salesforce Mule-Dev-301 Exam Syllabus Topics:

Section	Objectives
DataWeave and Data Transformation	- Mapping, filtering, and payload manipulation - DataWeave scripting and transformations
Security and Governance	- Authentication and authorization mechanisms - API policies and security enforcement
Error Handling and Troubleshooting	- Debugging and logging techniques - Exception handling strategies in Mule applications
Integration and API-led Connectivity	- API-led connectivity principles - System, Process, and Experience APIs design
Anypoint Platform Development	- Anypoint Studio usage and project structure - Mule application development
Deployment and Testing	- Unit testing and integration testing approaches - Deployment strategies across environments

>> Mule-Dev-301 Training Pdf <<

Pass Guaranteed Quiz 2026 Salesforce Mule-Dev-301 – High-quality Training Pdf

The majority of people encounter the issue of finding extraordinary Salesforce Mule-Dev-301 exam dumps that can help them prepare for the actual Salesforce Certified MuleSoft Developer II exam. They strive to locate authentic and up-to-date Salesforce Mule-Dev-301 Practice Questions for the Salesforce Mule-Dev-301 exam, which is a tough ask.

Salesforce Certified MuleSoft Developer II Sample Questions (Q48-Q53):

NEW QUESTION # 48
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.

```
<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>
```

- B.

```
<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>

<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>
```

- C.

```
<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>
```

- D.
- E.

```
<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>
```

Answer: B

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

NEW QUESTION # 49
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. Validation, initialize, compile, test, package, install verify, deploy
- B. Validate, initialize, compile, test package, verify, install, deploy
- C. Validate, initialize, compile, package, test, install, verify, verify, deploy
- D. Initialize, validate, compute, test, package, verify, install, deploy

Answer: B

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

NEW QUESTION # 50
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 refresh token
- B. Token URL, authorization URL, client ID, client secret local callback URL
- C. External callback URL, access token URL, local authorization URL, authorization URL, client ID, client secret
- D. External callback URL, access token URL, client ID response access token

Answer: C

Explanation:
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 # 51
A heathcare 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?

- A. Install libraries in a local repository and refer to it in the pm.xml file
- 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. Refer to JAR files in the code

Answer: A

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

NEW QUESTION # 52
Which pattern should be used to invoke multiple HTTP APIs in parallel and roll back failed requests in sequence?

- A. Scatter-Gather as central Saga orchestrator for all API request with compensating actions for failing routes
- B. A Parallel for Each scope with each HTTP request wrapped in a Try scope
- C. A database as a transactional outbox and an Until Successful router to retry any requests
- D. VM queues as a reliability pattern with error handlers to roll back any requests

Answer: A

Explanation:
To invoke multiple HTTP APIs in parallel and roll back failed requests in sequence, the developer should use a Scatter-Gather router as a central Saga orchestrator for all API requests with compensating actions for failing routes. A Scatter-Gather router executes multiple routes concurrently and aggregates the results. A Saga orchestrator coordinates a series of actions across different services and handles failures by executing compensating actions. Therefore, using a Scatter-Gather router as a Saga orchestrator allows invoking multiple HTTP APIs in parallel and rolling back any failed requests in sequence. Reference: https://docs.mulesoft.com/mule-runtime/4.3/scatter-gather-concept https://docs.mulesoft.com/mule-runtime/4.3/saga

NEW QUESTION # 53
.....

The test software used in our products is a perfect match for Windows' Mule-Dev-301 learning material, which enables you to enjoy the best learning style on your computer. Our Mule-Dev-301 study materials also use the latest science and technology to meet the new requirements of authoritative research material network learning. Unlike the traditional way of learning, the great benefit of our Mule-Dev-301 Study Materials are that when the user finishes the exercise, he can get feedback in the fastest time.

myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.slideshare.net, www.stes.tyc.edu.tw, www.impactio.com, mayagriffiths253.blogspot.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, Disposable vapes

P.S. Free & New Mule-Dev-301 dumps are available on Google Drive shared by Actualtests4sure: https://drive.google.com/open?id=1je2t2T2AHEWme1qLlSFXWrTNMEedn1Jx