

PassTest

Bessere Qualität , bessere Dienstleistungen!



Q&A

<http://www.passtest.de>

Einjährige kostenlose Aktualisierung

Exam : MCIA-Level 1 MAINTENANCE

**Title : MuleSoft Certified
Integration Architect - Level
1 MAINTENANCE**

Version : DEMO

1.A leading bank implementing new mule API.

The purpose of API to fetch the customer account balances from the backend application and display them on the online platform the online banking platform. The online banking platform will send an array of accounts to Mule API get the account balances.

As a part of the processing the Mule API needs to insert the data into the database for auditing purposes and this process should not have any performance related implications on the account balance retrieval flow

How should this requirement be implemented to achieve better throughput?

- A. Implement the Async scope fetch the data from the backend application and to insert records in the Audit database
- B. Implement a for each scope to fetch the data from the back-end application and to insert records into the Audit database
- C. Implement a try-catch scope to fetch the data from the back-end application and use the Async scope to insert records into the Audit database
- D. Implement parallel for each scope to fetch the data from the backend application and use Async scope to insert the records into the Audit database

Answer: D

2.An organization is designing Mule application which connects to a legacy backend. It has been reported that backend services are not highly available and experience downtime quite often. As an integration architect which of the below approach you would propose to achieve high reliability goals?

- A. Alerts can be configured in Mule runtime so that backend team can be communicated when services are down
- B. Until Successful scope can be implemented while calling backend API's
- C. On Error Continue scope to be used to call in case of error again
- D. Create a batch job with all requests being sent to backend using that job as per the availability of backend API's

Answer: B

Explanation:

Correct answer is Untill Successful scope can be implemented while calling backend API's

The Until Successful scope repeatedly triggers the scope's components (including flow references) until they all succeed or until a maximum number of retries is exceeded The scope provides option to control the max number of retries and the interval between retries The scope can execute any sequence of processors that may fail for whatever reason and may succeed upon retry

3.Which Salesforce API is invoked to deploy, retrieve, create or delete customization information such as custom object definitions using a Mule Salesforce connector in a Mule application?

- A. Metadata API
- B. REST API
- C. SOAP API
- D. Bulk API

Answer: B

4.How are the API implementation, API client, and API consumer combined to invoke and process an API?

- A. The API consumer creates an API implementation, which receives API invocations from an API such that they are processed for an API client
- B. The API consumer creates an API client which sends API invocations to an API such that they are processed by an API implementation
- C. An API client creates an API consumer, which receives API invocation from an API such that they are processed for an API implementation
- D. The API client creates an API consumer which sends API invocations to an API such that they are processed by API implementation

Answer: C

Explanation:

The API consumer creates an API client which sends API invocations to an API such that they are processed by an API implementation

This is based on below definitions API client

- An application component
 - that accesses a service
 - by invoking an API of that service - by definition of the term API over HTTP API consumer
 - A business role, which is often assigned to an individual
 - that develops API clients, i.e., performs the activities necessary for enabling an API client to invoke APIs
- API implementation
- An application component
 - that implements the functionality

5.A system API EmployeeSAPI is used to fetch employee's data from an underlying SQL database.

The architect must design a caching strategy to query the database only when there is an update to the employees stable or else return a cached response in order to minimize the number of redundant transactions being handled by the database.

What must the architect do to achieve the caching objective?

- A. Use an On Table Row on employees table and call invalidate cache Use an object store caching strategy and expiration interval to empty
- B. Use a Scheduler with a fixed frequency every hour triggering an invalidate cache flow Use an object store caching strategy and expiration interval to empty
- C. Use a Scheduler with a fixed frequency every hour triggering an invalidate cache flow Use an object store caching strategy and set expiration interval to 1-hour
- D. Use an on table rule on employees table call invalidate cache and said new employees data to cache Use an object store caching strategy and set expiration interval to 1-hour

Answer: A