MCIA-LEVEL-1 Q&As

MuleSoft Certified Integration Architect - Level 1

Pass Mulesoft MCIA-LEVEL-1 Exam with 100% Guarantee

QUESTION 1

The implementation of a Process API must change. What is a valid approach that minimizes the impact of this change on API clients?

A. Implement required changes to the Process API implementation so that whenever possible, the Process API\\'s RAML definition remains unchanged

B. Update the RAML definition of the current Process API and notify API client developers by sending them links to the updated RAML definition

C. Postpone changes until API consumers acknowledge they are ready to migrate to a new Process API or API version

D. Implement the Process API changes in a new API implementation, and have the old API implementation return an HTTP status code 301 - Moved Permanently to inform API clients they should be calling the new API implementation

Correct Answer: A

Explanation:

*

Option B shouldn\\'t be used unless extremely needed, if RAML is changed, client needs to accommodate changes. Question is about minimizing impact on Client. So this is not a valid choice.

*

Option C isn\\'t valid as Business can\\'t stop for consumers acknowledgment.

*

Option D again needs Client to accommodate changes and isn\\'t viable option.

*

Best choice is A where RAML definition isn\\'t changed and underlined functionality is changed without any dependency on client and without impacting client.

QUESTION 2

A platform architect includes both an API gateway and a service mesh in the architect of a distributed application for communication management. Which type of communication management does a service mesh typically perform in this architecture?

A. Between application services and the firewall

B. Between the application and external API clients

C. Between services within the application

D. Between the application and external API implementations.

Correct Answer: C

QUESTION 3

An organization is designing a Mule application to periodically poll an SFTP location for new files containing sales order records and then process those sales orders. Each sales order must be processed exactly once.

To support this requirement, the Mule application must identify and filter duplicate sales orders on the basis of a unique ID contained in each sales order record and then only send the new sales orders to the downstream system.

What is the most idiomatic (used for its intended purpose) Anypoint connector, validator, or scope that can be configured in the Mule application to filter duplicate sales orders on the basis of the unique ID field contained in each sales order record?

A. Configure a Cache scope to filter and store each record from the received file by the order ID

B. Configure a Database connector to filter and store each record by the order ID

C. Configure an Idempotent Message Validator component to filter each record by the order ID

D. Configure a watermark In an On New or Updated File event source to filter unique records by the order ID

Correct Answer: C

Reference: https://docs.mulesoft.com/mule-runtime/3.9/idempotent-filter

QUESTION 4

A Mule application currently writes to two separate SQL Server database instances across the internet using a single XA transaction. It is 58. proposed to split this one transaction into two separate non-XA transactions with no other changes to the Mule application.

What non-functional requirement can be expected to be negatively affected when implementing this change?

A. Throughput

B. Consistency

C. Response time

D. Availability

Correct Answer: B

Explanation: Correct answer is Consistency as XA transactions are implemented to achieve this. XA transactions are added in the implementation to achieve goal of ACID properties. In the context of transaction processing, the acronym ACID refers to the four key properties of a transaction: atomicity, consistency, isolation, and durability. Atomicity : All changes to data are performed as if they are a single operation. That is, all the changes are performed, or none of them are. For example, in an application that transfers funds from one account to another, the atomicity property ensures that, if a debit is made successfully from one account, the corresponding credit is made to the other account. Consistency : Data is in a consistent state when a transaction starts and when it ends.For example, in an application that transfers funds from one account to another, the consistency property ensures that the total value of funds in both the accounts is the same at the start and end of each transaction. Isolation : The intermediate state of a transaction is invisible to other transactions. As a result, transactions that run concurrently appear to be serialized. For example, in an application that transfers funds from one account to another, the isolation property ensures that another transaction sees the transferred funds in one account or the other, but not in both, nor in neither. Durability : After a transaction successfully completes, changes to data persist and are not undone, even in the event of a system failure. For example, in an application that transfers funds from one account to another, the durability property ensures that the changes made to each account will not be reversed. MuleSoft reference: https://docs.mulesoft.com/mule-runtime/4.3/xa-transactions

QUESTION 5

A project team uses RAML specifications to document API functional requirements and deliver API definitions. As per the current legal requirement, all designed API definitions to be augmented with an additional non-functional requirement to protect the services from a high rate of requests according to define service level agreements.

Assuming that the project is following Mulesoft API governance and policies, how should the project team convey the necessary non-functional requirement to stakeholders?

A. Create proxies in API manager for the non functional requirement and publish to exchange

B. Add all non functional requirements as comments to RAML specification and publish to exchange

C. Create various SLA\\'s in API manager for the non functional requirement and publish to exchange

D. Update API definitions with the fragment for the appropriate policy and publish to exchange

Correct Answer: D