PASSEXAM 問題集

更に上のクオリティ　更に上のサービス

EXAM

1年で無料進級することに提供する

http://www.passexam.jp

Exam : **MCIA-Level 1**

Title : MuleSoft Certified Integration Architect - Level 1

Version : DEMO

1.An external web UI application currently accepts occasional HTTP requests from client web browsers to change (insert, update, or delete) inventory pricing information in an inventory system's database. Each inventory pricing change must be transformed and then synchronized with multiple customer experience systems in near real-time (in under 10 seconds). New customer experience systems are expected to be added in the future.
The database is used heavily and limits the number of SELECT queries that can be made to the database to 10 requests per hour per user.
What is the most scalable, idiomatic (used for its intended purpose), decoupled. reusable, and maintainable integration mechanism available to synchronize each inventory pricing change with the various customer experience systems in near real-time?
A. Write a Mule application with a Database On Table Row event source configured for the inventory pricing database, with the watermark attribute set to an appropriate database column
In the same now, use a Scatter-Gather to call each customer experience system's REST API with transformed inventory-pricing records
B. Add a trigger to the inventory-pricing database table so that for each change to the inventory pricing database, a stored procedure is called that makes a REST call to a Mule application
Write the Mule application to publish each Mule event as a message to an Anypoint MQ exchange
Write other Mule applications to subscribe to the Anypoint MQ exchange, transform each received message, and then update the Mule application's corresponding customer experience system(s)
C. Replace the external web UI application with a Mule application to accept HTTP requests from client web browsers
In the same Mule application, use a Batch Job scope to test if the database request will succeed, aggregate pricing changes within a short time window, and then update both the inventory pricing database and each customer experience system using a Parallel For Each scope
D. Write a Mule application with a Database On Table Row event source configured for the inventory pricing database, with the ID attribute set to an appropriate database column
In the same flow, use a Batch Job scope to publish transformed Inventory-pricing records to an Anypoint MQ queue
Write other Mule applications to subscribe to the Anypoint MQ queue, transform each received message, and then update the Mule application's corresponding customer experience system(s)
Answer: B

2.An ABC Farms project team is planning to build a new API that is required to work with data from different domains across the organization.
The organization has a policy that all project teams should leverage existing investments by reusing existing APIs and related resources and documentation that other project teams have already developed and deployed.
To support reuse, where on Anypoint Platform should the project team go to discover and read existing APIs, discover related resources and documentation, and interact with mocked versions of those APIs?
A. Design Center
B. API Manager
C. Runtime Manager
D. Anypoint Exchange
Answer: D

Explanation:
The mocking service is a feature of Anypoint Platform and runs continuously. You can run the mocking service from the text editor, the visual editor, and from Anypoint Exchange. You can simulate calls to the API in API Designer before publishing the API specification to Exchange or in Exchange after publishing the API specification.
Reference: https://docs.mulesoft.com/design-center/design-mocking-service

3.What Mule application can have API policies applied by Anypoint Platform to the endpoint exposed by that Mule application?
A. A Mule application that accepts requests over HTTP/1x
B. A Mule application that accepts JSON requests over TCP but is NOT required to provide a response.
C. A Mule application that accepts JSON requests over WebSocket
D. A Mule application that accepts gRPC requests over HTTP/2
Answer: A
Explanation:
* HTTP/1.1 keeps all requests and responses in plain text format.
* HTTP/2 uses the binary framing layer to encapsulate all messages in binary format, while still maintaining HTTP semantics, such as verbs, methods, and headers. It came into use in 2015, and offers several methods to decrease latency, especially when dealing with mobile platforms and server-intensive graphics and videos
* Currently, Mule application can have API policies only for Mule application that accepts requests over HTTP/1x

4.An integration Mute application consumes and processes a list of rows from a CSV file. Each row must be read from the CSV file, validated, and the row data sent to a JMS queue, in the exact order as in the CSV file.
If any processing step for a row falls, then a log entry must be written for that row, but processing of other rows must not be affected.
What combination of Mute components is most idiomatic (used according to their intended purpose) when Implementing the above requirements?
A. Scatter-Gather component On Error Continue scope
B. VM connector first Successful scope On Error Propagate scope
C. For Each scope On Error Continue scope
D. Async scope On Error Propagate scope
Answer: C
Explanation:
* On Error Propagate halts execution and sends error to the client. In this scenario it's mentioned that "processing of other rows must not be affected" so Option B and C are ruled out.
* Scatter gather is used to club multiple responses together before processing. In this scenario, we need sequential processing. So option A is out of choice.
* Correct answer is For Each scope & On Error Continue scope Below requirement can be fulfilled in the below way
1) Using For Each scope, which will send each row from csv file sequentially. each row needs to be sent sequentially as requirement is to send the message in exactly the same way as it is mentioned in the csv

file
2) Also other part of requirement is if any processing step for a row fails then it should log an error but should not affect other record processing. This can be achieved using On error Continue scope on these set of activities. so that error will not halt the processing. Also logger needs to be added in error handling section so that it can be logged.
* Attaching diagram for reference. Here it's try scope, but similar would be the case with For Each loop.

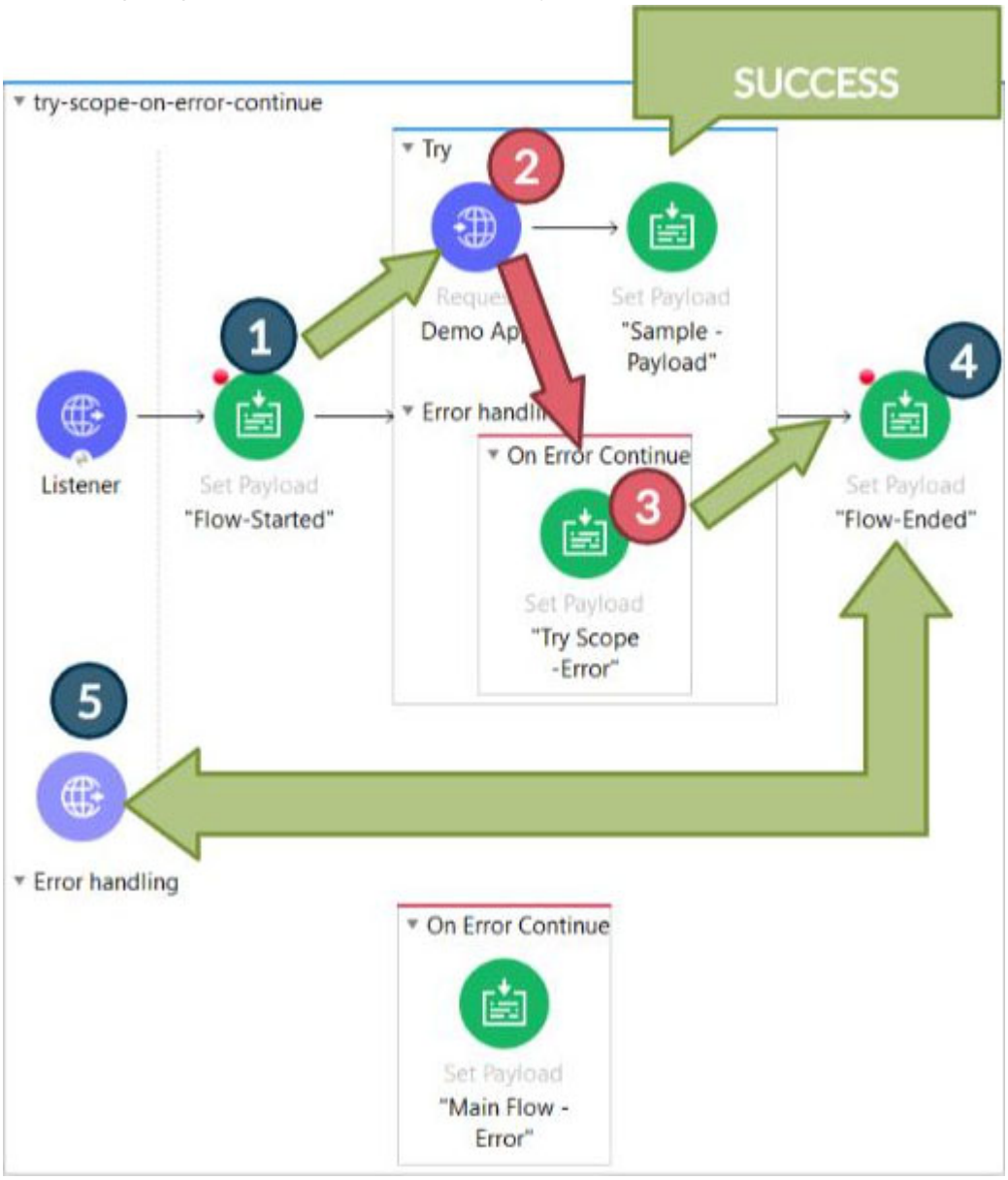

Diagram
Description automatically generated

5.A Mule application is synchronizing customer data between two different database systems.
What is the main benefit of using eXtended Architecture (XA) transactions over local transactions to synchronize these two different database systems?
A. An XA transaction synchronizes the database systems with the least amount of Mule configuration or coding
B. An XA transaction handles the largest number of requests in the shortest time

C. An XA transaction automatically rolls back operations against both database systems if any operation falls
D. An XA transaction writes to both database systems as fast as possible
Answer: B
Explanation:
Reference: https://docs.oracle.com/middleware/1213/wls/PERFM/llrtune.htm#PERFM997