EXAMTOPICS

-Expert Verified, Online, **Free**.



CERTIFICATION TEST

-CertificationTest.net - Cheap & Quality Resources With Best Support

Question #1 *Topic 1*

What requirement prevents using Anypoint MQ as the messaging broker for a Mule application?

A. When the payload sent through the message broker must be encrypted

B. When the messaging broker must be deployed on-premises

C. When the payload sent through the message broker must use XML format

D. When the messaging broker must support point-to-point messaging

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #2 *Topic 1*

A company is modernizing its legacy systems to accelerate access to applications and data while supporting the adoption of new technologies. The key to achieving this business goal is unlocking the company's systems and data, including a set of existing services hosted on-premises that can be accessed by authorized external clients. The ГГ staff is mainly experienced only with supporting its legacy systems.

Considering the current aggressive backlog and the project delivery requirements, the company wants to take a strategic approach in the first phase of its transformation projects by quickly deploying APIs in Mule runtimes that are able to scale, connect to on-premises systems, and migrate as needed.

Following MuleSoft best practices, what MuleSoft runtime deployment option best meets the company's goals to begin its digital transformation journey?

A. Customer-hosted self-provisioned runtimes

B. CloudHub runtimes

C. Runtime Fabric on VMs/bare metal

D. Customer-hosted runtimes provisioned by a MuleSoft services partner

Suggested Answer: *A*

Community vote distribution

B (100%)

beast456 Most Recent 2 months, 2 weeks ago

Selected Answer: B

Zero infra ownership, fastest time to first API, elastic scaling, on-prem connectivity via VPC/VPN, portable later.

upvoted 1 times

Question #3 *Topic 1*

An airline is architecting an API-led connectivity project to integrate its flight data into an online aggregation website. The connections must allow for secured communication, high performance, and asynchronous message exchange.

What are suitable interface technologies for the connectors in this integration assuming that MuleSoft fully supports those technologies and that Anypoint Connectors exist for these interfaces?

A. SOAP over HTTPS HOP over TLS gRPC over HTTPS

B. AsyncAPI over SQS AMQP with RabbitMQ JSON/REST over HTTPS

C. XML over ActiveMQ XML over FTP XML/REST over HTTPS

D. CSV over FTP YAML over TLS JSON over HTTPS

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #4 *Topic 1*

A Mule solution implements an Experience API and System API as Mule applications that process messages in this order:

1. The Experience API receives a request Anypoint MQ message (REQU) with a payload containing a variable-length list of request objects.

2. The Experience API uses a For Each scope to split the list into individual objects and sends each object as a REQU_OBJECT message to an Anypoint MQ queue named REQUEST

3. The System API listens on the REQUEST queue, processes each message independently of all other messages, and sends a response message to a response queue named RESPONSE_OBJECT. The System API processing times can vary and are occasionally fairly long.

4. The Experience API listens on the RESPONSE_OBJECT queue and receives, transforms, and accumulates each response.

5. After receiving all the responses corresponding to the original REQU message, the Experience API creates and publishes a response Anypoint MQ message (RESP) to an Anypoint MQ queue named RESPONSE with a payload containing the list of responses returned by the System API, organized in the same order as the request objects originally sent in the REQU message.

Assume successful response messages are returned by Experience API to the RESPONSE queue for all REQU messages.

What is required so the Experience API can ensure the length and order of the list of objects in RESP and REQU match, while at the same time maximizing message throughput?

A. After the For Each scope, add a second For Each scope configured with a persistent object store to collect response messages in the order in which they arrive, and then send the RESP message using this list of responses.

B. Use a persistent object store to keep track of the list length and all object indexes in the REQU message, both in the For Each scope and in all communication involving the System API. Set the maxConcurrency parameter for the receiving flow in the Experience API to a value greater than 1.

C. Use a Scatter-Gather within the For Each scope to ensure response message order Configure the Scatter-Gather with a persistent object store.

D. Perform all communication involving the System API synchronously from within the For Each scope so that responses in RESPONSE_OBJECT are in the exact same order as request objects in the RESP message. Within the For Each scope, use an Async scope configured with a persistent object store.

Suggested Answer: *B*

Community vote distribution



beast456 Most Recent 2 months, 2 weeks ago

Selected Answer: B

in Anypoint MQ / JMS request-reply designs, ordering across a queue boundary is never an emergent property of the transport. It must be reconstructed from explicitly propagated correlation metadata (index, total, correlation ID) held in shared persistent state. Any answer that implies the broker or the arrival sequence preserves order is wrong.

upvoted 1 times

Question #5 *Topic 1*

An insurance company is designing a hybrid, load-balanced, single-cluster runtime production environment. Due to performance Service Level Agreement (SLA) goals, it is looking into running the Mule applications in an active-active multi-node cluster configuration.

What should be considered when running its Mule applications in this type of environment?

A. Although the cluster environment is fully installed, configured, and running, it will not process any requests until an outage condition is detected by the primary node in the cluster

B. A Mule application deployed to multiple nodes runs in isolation from the other nodes in the cluster

C. All event sources, regardless of type, can be configured as the target source by the primary node in the cluster

D. An external load balancer is required to distribute incoming TCP/HTTP requests throughout the cluster nodes

Suggested Answer: *D*

Community vote distribution

D (100%)

beast456 Most Recent 2 months, 2 weeks ago

Selected Answer: D

load balancing is never provided by Mule clustering itself. CloudHub 1.0 → Shared LB or DLB; CloudHub 2.0 → ingress provided by the platform; RTF → ingress controller; customer-hosted cluster → you supply an external LB. Clustering provides HA, shared state, and failover only.

upvoted 1 times

Question #6 *Topic 1*

A marketing organization is designing a Mule application to process campaign data. The Mule application will periodically check for a file in an SFTP location and process the records in the file. The size of the file can vary from 10MB to 5GB. Due to the limited availability of vCores, the Mule application is deployed to a single CloudHub worker/replica configured with vCore size 0.2.

The application must transform and send different formats of this file to three different downstream SFTP locations.

What is the most direct way to configure the SFTP operations or event sources to process the large files to support these deployment requirements?

A. Use a file-stored repeatable stream

B. Use a file-stored nonrepeatable stream

C. Use an in-memory repeatable stream

D. Use an in-memory nonrepeatable stream

Suggested Answer: *A*

Community vote distribution

A (100%)

beast456 Most Recent 2 months, 2 weeks ago

Selected Answer: A

on Mule streaming questions, resolve the axes separately. Number of consumers → repeatable vs. non-repeatable. Payload size vs. available heap → in-memory vs. file-stored. The correct option is always the intersection, and the trap is always the option that satisfies whichever axis the stem mentions first.

upvoted 1 times

Question #7 *Topic 1*

A new Mule application has been deployed through Runtime Manager to CloudHub 1.0 using a CI/CD pipeline with sensitive properties set as cleartext. The Runtime Manager Administrator opened a high priority incident ticket about this violation of their security requirements indicating these sensitive properties values must not be stored or visible in Runtime Manager but should be changeable in Runtime Manager by Administrators with proper permissions.

How can the Mule application be deployed while safely hiding the sensitive properties?

A. Add encrypted versions of the sensitive properties as global configuration properties in the Mule application

B. Add a new wrapper.java.additional.XX parameter for each sensitive property in the wrapper.conf file used by the CI/CD pipeline scripts

C. Create a variable for each sensitive property and declare them as hidden in the CI/CD pipeline scripts

D. Add an ArrayList of all the sensitive properties' names in the mule-artifact.json file of the application

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #8 *Topic 1*

An organization has built a large monolithic application over the years and is currently looking to transition to a microservices architecture. During this transition phase, it has built some System APIs as Mule applications that provide access to different pieces of functionality of the monolithic application. The System APIs are deployed to Mule runtimes in a customer-hosted EC2 instance in AWS.

The System APIs communicate directly with one database used by the monolithic application. Currently, a Database connector and configuration for that connector must be built into every project. There is a plan to partition this database into various smaller data stores in the future.

What is the MuleSoft-recommended best practice to share the connector and configuration information among the APIs?

A. Build a separate Mule domain project for each API, and configure each of them to use a file on a shared file store to load the configuration information dynamically

B. Create an API proxy for each System API and share the Database connector configuration with all the API proxies via an automated policy

C. Build another System API that connects to the database, and refactor all the other APIs to make requests through the new System API to access the database

D. Build a Mule domain project, add the Database connector and configuration to it, and reference this one domain project from each System API

Suggested Answer: *D*

Community vote distribution

D (100%)

beast456 Most Recent 2 months, 2 weeks ago

Selected Answer: D

A Mule domain project is purpose-built for exactly this: sharing connector configurations and other resources across multiple Mule applications deployed to the same runtime

upvoted 1 times

Question #9 *Topic 1*

An architect is designing a Mule application to meet the following two requirements:

1. The application must process files asynchronously and reliably from an FTPS server to a back-end database using VM intermediary queues for load-balancing Mule events.

2. The application must process a medium rate of records from a source to a target system using a Batch Job scope.

To make the Mule application more reliable, the Mule application will be deployed to two CloudHub 1.0 workers.

Following MuleSoft-recommended best practices, how should the Mule application deployment typically be configured in Runtime Manger to best support the performance and reliability goals of both the Batch Job scope and the file processing VM queues?

A. In the Runtime Manager Properties tab, enable persistent VM queues for the FTPS connector

B. Check the Non-persistent VM queues checkbox in the application deployment configuration

C. In the Runtime Manager Properties tab, disable persistent VM queues for Batch Job scopes

D. Check the Persistent VM queues checkbox in the application deployment configuration

Suggested Answer: *D*

Community vote distribution

D (100%)

beast456 Most Recent 2 months, 2 weeks ago

Selected Answer: D

when a CloudHub 1.0 stem mentions VM queues and multiple workers and reliability or load-balancing, persistent VM queues are required — there is no alternative mechanism. Then check the volume descriptor to confirm the latency penalty is acceptable; that descriptor is the exam's tiebreaker, not decoration.

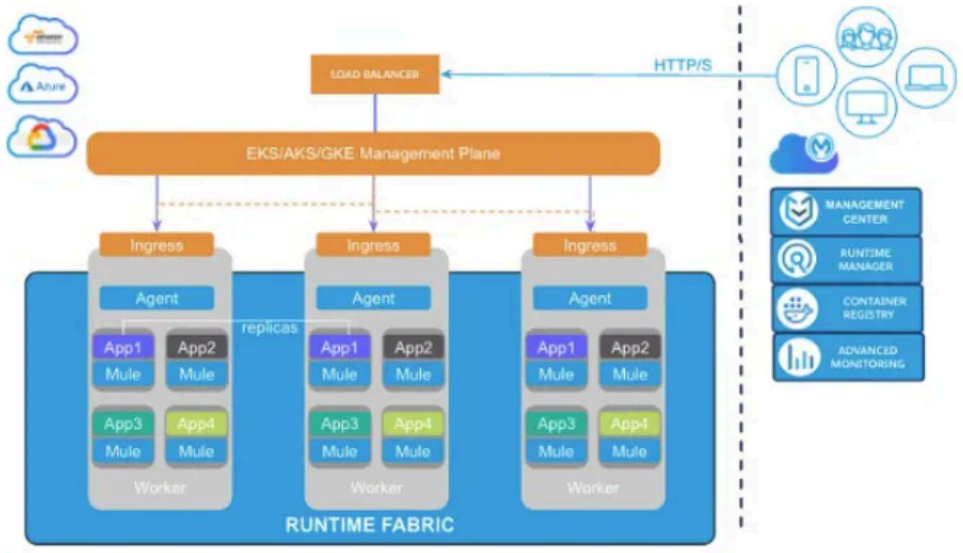
upvoted 1 times

Question #10 *Topic 1*

Refer to the exhibit.

A customer is running Mule applications on Runtime Fabric for Self-Managed Kubernetes (RTF-BYOK8) in a multi-cloud environment.

Based on this configuration, how do Agents and Runtime Manager communicate, and what is exchanged between them?



A. Agents communicate with Runtime Manager to provide updates on cluster status, download images, and update Mule applications (automatically when available) from the container registry

B. Agents communicate with Runtime Manager to provide updates on cluster status, download images, and update Mule applications (when available and chosen to do so by customer) from the container registry

C. Runtime Manager initiates traffic to Agents to check the status of clusters and auto-deploys new images when updates to Mule applications are available

D. Runtime Manager communicates with Agents to provide updates on cluster status, download images, and update Mule applications (when available and chosen to do so by customer) from the container registry

Suggested Answer: *B*

Agents report cluster status and retrieve updates; customers choose when available application updates are applied.

Community vote distribution

B (100%)

beast456 Most Recent 2 months, 2 weeks ago

Selected Answer: B

all RTF control-plane traffic is outbound from the cluster. Agent → Anypoint control plane for status, deployment instructions, and licensing; cluster → container registry for image pulls. Nothing in the Anypoint control plane ever reaches into the customer's cluster. Any option describing Runtime Manager as initiating, pushing, or reaching into the cluster is wrong on sight.

upvoted 1 times

Question #11 *Topic 1*

A Mule application is designed to call the Google Maps API to perform a distance computation, and is deployed to CloudHub.

At a minimum, what (if anything) should be configured in the TLS context of the HTTP Request configuration to meet these requirements?

A. Download the Google public certificate from a browser, generate a JKS file from it, and add it in TrustStore as part of the TLS context

B. The configuration is built-in and nothing extra is required for the TLS context

C. Request a private key from Google, create a PKCS#12 file with it, and add it in KeyStore as part of the TLS context

D. Download the Google public certificate from a browser, generate a JKS file from it, and add it in KeyStore as part of the TLS context

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #12 *Topic 1*

A bank is implementing a REST API in a Mule application to receive an array of accounts from an online banking platform user interface (UI), retrieve account balances for those accounts from a backend Finance system, and then return the account balances so they can be displayed in the online banking platform UI. As part of the processing, the MuleSoft API also needs to insert the

retrieved account data into an Audit Database for auditing purposes, the auditing process should not add latency to the account balance retrieval response back to the online banking platform UI.

The retrieveBalances flow in the Mule application is designed to use an operation in a connector to the Finance system (the Finance operation) that can only look up one account record at a time, and a operation from a different connector to the Audit system (the Audit operation) that can only insert one account record at a time.

To best meet the performance-related requirements, what scope or scopes should be used and how should they be used to incorporate the Finance operation and Audit operation into the retrieveBalances flow?

A. Wrap both connector operations in a Async scope.

B. Wrap both connector operations in a For-Each scope.

C. Wrap the Finance operation in a Until-Successful scope.

Wrap the Audit operation in a Try-Catch scope.

D. Wrap the Finance operation in a Parallel For-Each scope.

Wrap the Audit operation in a Async scope.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #13 *Topic 1*

A manufacturing company wants to share inventory updates with dealers D1 and D2 asynchronously and concurrently via Queues Q1 and Q2. Dealer D1 must consume the messages from the Queue Q1, and Dealer D2 must consume the messages from the Queue Q2.

Dealer D1 has implemented a retry mechanism to reprocess the transaction in case of any errors while processing the inventory updates. Dealer D2 has not implemented any retry mechanism.

How should the dealers acknowledge the message to avoid message loss and minimize impact on the current implementation?

A. Dealer D1 can use AUTO acknowledgement and Dealer D2 must use MANUAL acknowledgment and acknowledge the message after successful processing

B. Dealer D1 and Dealer D2 must use AUTO acknowledgement and acknowledge the message after successful processing

C. Dealer D1 can use AUTO acknowledgement and Dealer D2 can use IMMEDIATE acknowledgment and acknowledge the message after successful processing

D. Dealer D2 must use AUTO acknowledgement and Dealer D1 can use MANUAL acknowledgment and acknowledge the message after successful processing

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #14 *Topic 1*

An organization is in the process of building automated deployments using a CI/CD process. As a part of automated deployments, it wants to apply policies to API instances.

What tool can the organization use to promote and deploy API Manager policies?

A. Runtime Manager agent

B. Mule Maven plugin

C. MUnit Maven plugin

D. Anypoint CLI

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #15 *Topic 1*

How does a timeout attribute in a configured transaction manager help inform design decisions while using a JMS connector listening for incoming messages in an extended Architecture (XA) transaction managed by the transaction manager?

A. The timeout defines the time that is allowed to pass without the transaction ending explicitly, and after the timeout expires, the transaction rolls back

B. The time allowed to pass between committing the transaction and the completion of the Mule flow, and then after the timeout, flow processing triggers an error

C. After the timeout is exceeded, stale JMS consumer threads are destroyed and new threads are created

D. The timeout specifies the time allowed to pass between receiving JMS messages on the same JMS connection, and then after the timeout, a new JMS connection is established

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #16 *Topic 1*

Following MuleSoft's recommended best practices for API governance and API policies, a project team has used RAML specifications to document and publish functional requirements and detailed design definitions of its APIs. These API specifications have been used by various stakeholders to implement APIs.

Later, the project team requires all API specifications to be augmented with an additional non-functional requirement (NFR) to protect the backend services from a high rate of requests, according to defined service-level agreements (SLAs). The NFR's SLAs are based on a new tiered subscription level "Gold", "Silver", or "Platinum" that must be tied to a new parameter that is being added to the Accounts object in their enterprise data model.

Following MuleSoft's recommended best practices, how should the project team now convey the necessary non-functional requirement to stakeholders?

A. Create and deploy API proxies in API Manager for the NFR, change the baseUrl in each API specification to the corresponding API proxy implementation endpoint, and publish each modified API specification to Exchange

B. Create a shared RAML fragment required to implement the NFR, list each API implementation endpoint in the RAML fragment, and publish the RAML fragment to Exchange

C. Update each API specification with a shared RAML fragment required to implement the NFR and publish the RAML fragment and each modified API specification to Exchange

D. Update each API specification with comments about the NFR's SLAs and publish each modified API specification to Exchange

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #17 *Topic 1*

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

A. sObject Platform Action API

B. User Interface API

C. Metadata API

D. Process Rules API

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #18 *Topic 1*

Why would an Enterprise Architect use a single enterprise-wide canonical data model (COM) when designing an integration solution using Anypoint Platform?

A. To automate AI-enabled API implementation generation based on normalized backend databases from separate vendors

B. To reduce dependencies when integrating multiple systems that use different data formats

C. To remove the need to perform data transformation when processing message payloads in Mule applications

D. To leverage a data abstraction layer that shields existing Mule applications from non-backward compatible changes to the model's data structure

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #19 *Topic 1*

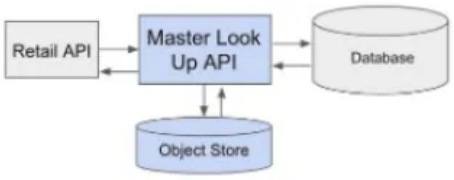
Refer to the exhibit.

A manufacturing company is developing a new set of APIs for its retail business. One of the APIs is a Master Look Up API, which is a System API. The API uses a persistent object-store. This API will be used by almost all other APIs to provide master lookup data.

The Master Look Up API is deployed on two CloudHub workers of 0.1 vCore each because there is a lot of master data to be cached. Most of the master lookup data is stored as a key-value pair. The cache gets refreshed if the key is not found in the cache.

During performance testing, it was determined that the Master Look Up API has a high response time due to the latency of database queries executed to fetch the master lookup data.

What two methods can be used to resolve these performance issues? (Choose two.)



A. Implement locking to synchronize access to the Object Store

B. Upgrade the vCore size from 0.1 vCore to 0.2 vCore

C. Implement the HTTP caching policy for all GET endpoints for the Master Look Up API

D. Implement an HTTP caching policy for all GET endpoints in the Master Look Up API

Suggested Answer: *B, D*
More vCore capacity and HTTP caching reduce repeated latency-heavy database reads.

Currently there are no comments in this discussion, be the first to comment!

Question #20 *Topic 1*

An organization is using MuleSoft to develop APIs and automatically deploy them to both CloudHub and on-premises targets. Recently, it has decided to include Runtime Fabric deployment targets as well, and Mule infrastructure is set up for this option.

What can now be used to automate deploying Mule applications to all three types of deployment targets?

A. The Object Store v2 REST API

B. The Mule Maven plugin

C. Anypoint Monitoring agent configuration file

D. The Runtime Manager agent configuration file

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #21 *Topic 1*

An organization is successfully using API-led connectivity. However, as the application network grows, all the manually performed tasks to publish, share and discover, register, apply policies to, and deploy an API are becoming repetitive, which is driving the organization to automate this process using efficient CI/CD pipelines.

Considering Anypoint Platform's capabilities, how should the organization automate its API lifecycle?

A. Use Maven with a custom configuration required for the API lifecycle

B. Use Exchange REST APIs for API management and Maven for API deployment

C. Use Anypoint CLI or Anypoint Platform REST APIs with a scripting language such as Groovy

D. Use Runtime Manager REST APIs for API management and Maven for API deployment

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #22 *Topic 1*

A corporation has deployed Mule applications to different customer-hosted Mule runtimes. Mule applications deployed to these Mule runtimes are managed by Anypoint Platform.

What needs to be installed or configured (if anything) to monitor these Mule applications from Anypoint Monitoring, and how is monitoring data from each Mule application sent to Anypoint Monitoring?

A. Install an Anypoint Monitoring agent on each Mule runtime.

Each Anypoint Monitoring agent sends monitoring data from the Mule applications running in its Mule runtime to Anypoint Monitoring.

B. Leave the out-of-the-box Anypoint Monitoring agent unchanged in its default Mule runtime installation.

Each Anypoint Monitoring agent sends monitoring data from the Mule applications running in its Mule runtime to Runtime Manager, then Runtime Manager sends monitoring data to Anypoint Monitoring.

C. Enable monitoring of individual Mule applications from the Runtime Manager application settings.

Runtime Manager sends monitoring data to Anypoint Monitoring for each deployed Mule application.

D. Install a Runtime Manager agent on each Mule runtime.

Each Runtime Manager agent sends monitoring data from the Mule applications running in its Mule runtime to Runtime Manager, then Runtime Manager sends monitoring data to Anypoint Monitoring.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #23 *Topic 1*

A Mule application is required to periodically process a large data set from a back-end database to Salesforce CRM using a Batch Job scope configured properly to process the high rate of records. The Mule application is deployed to two CloudHub 1.0 workers with the persistent VM queues option disabled (unchecked).

What is the consequence if a worker that is processing records from a Batch Job scope crashes during records processing?

A. About half the remaining records will be processed by the second worker and the rest will be lost, resulting in some unprocessed records

B. The entire batch job will be processed from scratch by the second worker, resulting in some duplicate processing

C. Remaining records will be lost, resulting in unprocessed records

D. All the remaining records will be processed by a new replacement worker, resulting in "exactly once" processing of every record

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #24 *Topic 1*

A payment processing company has implemented a Payment Processing API Mule application to process credit card and debit card transactions. Because the Payment Processing API handles highly sensitive information, the payment processing company requires that data must be encrypted both in-transit and at-rest.

To meet these security requirements, consumers of the Payment Processing API must create request message payloads in a JSON format specified by the API, and the message payload values must be encrypted.

How can the Payment Processing API validate requests received from API consumers?

A. The Mule application implementation can use the APIkit module to decrypt and then validate the JSON data

B. The Mule application implementation can use the Validation module to decrypt and then validate the JSON data

C. A Transport Layer Security (TLS) - Inbound policy can be applied in API Manager to decrypt the message payload and the Mule application implementation can then use the JSON Validation module to validate the JSON data

D. The Mule application implementation can use DataWeave to decrypt the message payload and then use the JSON Scheme Validation module to validate the JSON data

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!