



MCPA-LEVEL-1-MAINTENANCE^{Q&As}

MuleSoft Certified Platform Architect - Level 1 MAINTENANCE

Pass Mulesoft MCPA-LEVEL-1-MAINTENANCE Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.geekcert.com/mcpa-level-1-maintenance.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Mulesoft
Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

Version 3.0.1 of a REST API implementation represents time values in PST time using ISO 8601 hh:mm:ss format. The API implementation needs to be changed to instead represent time values in CEST time using ISO 8601 hh:mm:ss format. When following the semver.org semantic versioning specification, what version should be assigned to the updated API implementation?

- A. 3.0.2
- B. 4.0.0
- C. 3.1.0
- D. 3.0.1

Correct Answer: B

4.0.0

As per semver.org semantic versioning specification:

Given a version number MAJOR.MINOR.PATCH, increment the:

-MAJOR version when you make incompatible API changes.

-

MINOR version when you add functionality in a backwards compatible manner.

-

PATCH version when you make backwards compatible bug fixes. As per the scenario given in the question, the API implementation is completely changing its behavior. Although the format of the time is still being maintained as hh:mm:ss

and there is no change in schema w.r.t format, the API will start functioning different after this change as the times are going to come completely different. Example: Before the change, say, time is going as 09:00:00 representing the PST.

Now on, after the change, the same time will go as 18:00:00 as Central European Summer Time is 9 hours ahead of Pacific Time.

>> This may lead to some uncertain behavior on API clients depending on how they are handling the times in the API response. All the API clients need to be informed that the API functionality is going to change and will return in CEST format. So, this considered as a MAJOR change and the version of API for this new change would be 4.0.0

QUESTION 2

Once an API Implementation is ready and the API is registered on API Manager, who should request the access to the



API on Anypoint Exchange?

- A. None
- B. Both
- C. API Client
- D. API Consumer

Correct Answer: D

API Consumer ***** >> API clients are piece of code or programs that use the client credentials of API consumer but does not directly interact with Anypoint Exchange to get the access >> API consumer is the one who should get registered and request access to API and then API client needs to use those client credentials to hit the APIs So, API consumer is the one who needs to request access on the API from Anypoint Exchange

QUESTION 3

A company requires Mule applications deployed to CloudHub to be isolated between non- production and production environments. This is so Mule applications deployed to non- production environments can only access backend systems running in their customer- hosted non-production environment, and so Mule applications deployed to production environments can only access backend systems running in their customer-hosted production environment. How does MuleSoft recommend modifying Mule applications, configuring environments, or changing infrastructure to support this type of per- environment isolation between Mule applications and backend systems?

- A. Modify properties of Mule applications deployed to the production Anypoint Platform environments to prevent access from non-production Mule applications
- B. Configure firewall rules in the infrastructure inside each customer-hosted environment so that only IP addresses from the corresponding Anypoint Platform environments are allowed to communicate with corresponding backend systems
- C. Create non-production and production environments in different Anypoint Platform business groups
- D. Create separate Anypoint VPCs for non-production and production environments, then configure connections to the backend systems in the corresponding customer-hosted environments

Correct Answer: D

Create separate Anypoint VPCs for non-production and production environments, then configure connections to the backend systems in the corresponding customer-hosted environments. ***** >> Creating different Business Groups does NOT make any difference w.r.t accessing the non-prod and prod customer-hosted environments. Still they will be accessing from both Business Groups unless process network restrictions are put in place. >> We need to modify or couple the Mule Application Implementations with the environment. In fact, we should never implements application coupled with environments by binding them in the properties. Only basic things like endpoint URL etc should be bundled in properties but not environment level access restrictions. >> IP addresses on CloudHub are dynamic until unless a special static addresses are assigned. So it is not possible to setup firewall rules in customer-hosted infrastructure. More over, even if static IP addresses are assigned, there could be 100s of applications running on cloudhub and setting up rules for all of them would be a hectic task, non-maintainable and definitely got a good practice. >> The best practice recommended by Mulesoft (In fact any cloud provider), is to have your Anypoint VPCs seperated for Prod and Non-Prod and perform the VPC peering or VPN tunneling for these Anypoint VPCs to



respective Prod and Non-Prod customer-hosted environment networks. : <https://docs.mulesoft.com/runtime-manager/virtual-private-cloud>

QUESTION 4

A system API has a guaranteed SLA of 100 ms per request. The system API is deployed to a primary environment as well as to a disaster recovery (DR) environment, with different DNS names in each environment. An upstream process API invokes the system API and the main goal of this process API is to respond to client requests in the least possible time. In what order should the system APIs be invoked, and what changes should be made in order to speed up the response time for requests from the process API?

- A. In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment, and ONLY use the first response
- B. In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment using a scatter-gather configured with a timeout, and then merge the responses
- C. Invoke the system API deployed to the primary environment, and if it fails, invoke the system API deployed to the DR environment
- D. Invoke ONLY the system API deployed to the primary environment, and add timeout and retry logic to avoid intermittent failures

Correct Answer: A

In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment, and ONLY use the first response.

>> The API requirement in the given scenario is to respond in least possible time. >> The option that is suggesting to first try the API in primary environment and then fallback to API in DR environment would result in successful response but

NOT in least possible time. So, this is NOT a right choice of implementation for given requirement. >> Another option that is suggesting to ONLY invoke API in primary environment and to add timeout and retries may also result in successful

response upon retries but NOT in least possible time. So, this is also NOT a right choice of implementation for given requirement.

>> One more option that is suggesting to invoke API in primary environment and API in DR environment in parallel using Scatter-Gather would result in wrong API response as it would return merged results and moreover, Scatter-Gather

does things in parallel which is true but still completes its scope only on finishing all routes inside it. So again, NOT a right choice of implementation for given requirement

The Correct choice is to invoke the API in primary environment and the API in DR environment parallely, and using ONLY the first response received from one of them.



QUESTION 5

A REST API is being designed to implement a Mule application.

What standard interface definition language can be used to define REST APIs?

- A. Web Service Definition Language(WSDL)
- B. OpenAPI Specification (OAS)
- C. YAML
- D. AsyncAPI Specification

Correct Answer: B

[MCPA-
LEVEL-1-MAINTENANCE
PDF Dumps](#)

[MCPA-
LEVEL-1-MAINTENANCE
Study Guide](#)

[MCPA-
LEVEL-1-MAINTENANCE
Exam Questions](#)