MCPA-LEVEL1 Q&As

MuleSoft Certified Platform Architect - Level 1

Pass Mulesoft MCPA-LEVEL1 Exam with 100% Guarantee

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

https://www.geekcert.com/mulesoft-certified-platform-architect-level-1.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

Select the correct Owner-Layer combinations from below options

A. 1. App Developers owns and focuses on Experience Layer APIs

2.

Central IT owns and focuses on Process Layer APIs

3.

LOB IT owns and focuses on System Layer APIs

B. 1. Central IT owns and focuses on Experience Layer APIs

2.

LOB IT owns and focuses on Process Layer APIs

3.

App Developers owns and focuses on System Layer APIs

C. 1. App Developers owns and focuses on Experience Layer APIs

2.

LOB IT owns and focuses on Process Layer APIs

3.

Central IT owns and focuses on System Layer APIs

Correct Answer: C

1.

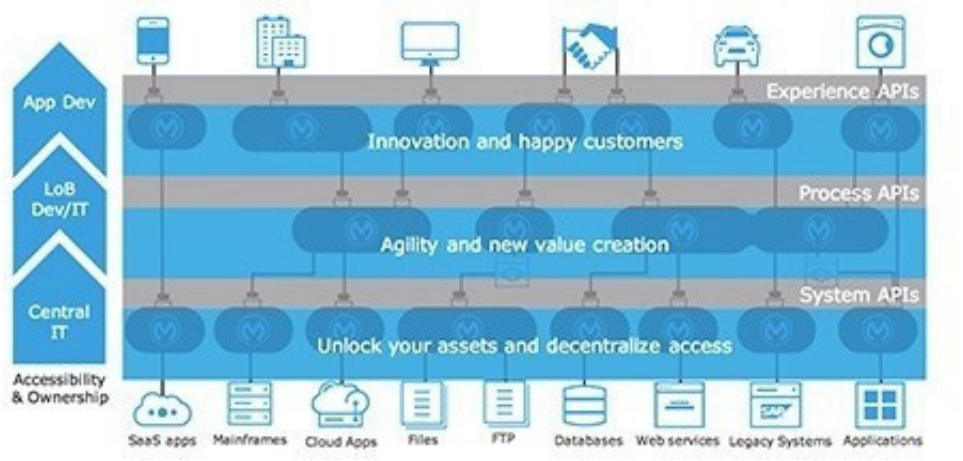
App Developers owns and focuses on Experience Layer APIs

2.

LOB IT owns and focuses on Process Layer APIs

3.

Central IT owns and focuses on System Layer APIs



References: https://blogs.mulesoft.com/biz/api/experience-api-ownership/ https://blogs.mulesoft.com/biz/api/process-api-ownership/ https://blogs.mulesoft.com/biz/api/system-api-ownership/

QUESTION 2

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

****

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 3

Traffic is routed through an API proxy to an API implementation. The API proxy is managed by API Manager and the API implementation is deployed to a CloudHub VPC using Runtime Manager. API policies have been applied to this API. In this deployment scenario, at what point are the API policies enforced on incoming API client requests?

A. At the API proxy

B. At the API implementation

C. At both the API proxy and the API implementation

D. At a MuleSoft-hosted load balancer

Correct Answer: A

At the API proxy

****

>> API Policies can be enforced at two places in Mule platform. >> One - As an Embedded Policy enforcement in the same Mule Runtime where API implementation is running.

>> Two - On an API Proxy sitting in front of the Mule Runtime where API implementation is running.

>> As the deployment scenario in the question has API Proxy involved, the policies will be enforced at the API Proxy.

QUESTION 4

A company uses a hybrid Anypoint Platform deployment model that combines the EU control plane with customer-hosted Mule runtimes. After successfully testing a Mule API implementation in the Staging environment, the Mule API implementation is set with environment-specific properties and must be promoted to the Production environment. What is a way that MuleSoft recommends to configure the Mule API implementation and automate its promotion to the Production environment?

A. Bundle properties files for each environment into the Mule API implementation\\'s deployable archive, then promote the Mule API implementation to the Production environment using Anypoint CLI or the Anypoint Platform REST APIsB.

B. Modify the Mule API implementation\\'s properties in the API Manager Properties tab, then promote the Mule API implementation to the Production environment using API Manager

C. Modify the Mule API implementation\\'s properties in Anypoint Exchange, then promote the Mule API implementation to the Production environment using Runtime Manager

D. Use an API policy to change properties in the Mule API implementation deployed to the Staging environment and another API policy to deploy the Mule API implementation to the Production environment

Correct Answer: A

Bundle properties files for each environment into the Mule API implementation\\'s deployable archive, then promote the Mule API implementation to the Production environment using Anypoint CLI or the Anypoint Platform REST APIs ********** >> Anypoint Exchange is for asset discovery and documentation. It has got no provision to modify the properties of Mule API implementations at all. >> API Manager is for managing API instances, their contracts, policies and SLAs. It has also got no provision to modify the properties of API implementations. >> API policies are to address Non-functional requirements of APIs and has again got no provision to modify the properties of API implementations. So, the right way and recommended way to do this as part of development practice is to bundle properties files for each environment into the Mule API implementation and just point and refer to respective file per environment.

QUESTION 5

What is a key performance indicator (KPI) that measures the success of a typical C4E that is immediately apparent in responses from the Anypoint Platform APIs?

A. The number of production outage incidents reported in the last 24 hours

B. The number of API implementations that have a publicly accessible HTTP endpoint and are being managed by Anypoint Platform

C. The fraction of API implementations deployed manually relative to those deployed using a CI/CD tool

D. The number of API specifications in RAML or OAS format published to Anypoint Exchange

Correct Answer: D

The number of API specifications in RAML or OAS format published to Anypoint Exchange ********** >> The success of C4E always depends on their contribution to the number of reusable assets that they have helped to build and publish to Anypoint Exchange. >> It is NOT due to any factors w.r.t # of outages, Manual vs CI/CD deployments or Publicly accessible HTTP endpoints >> Anypoint Platform APIs helps us to quickly run and get the number of published RAML/OAS assets to Anypoint Exchange. This clearly depicts how successful a C4E team is based on number of returned assets in the response. Reference: https://help.mulesoft.com/s/question/0D52T00004mXSTUSA4/how-should-a- company-measure-c4e-success