Question 1

Question Type: **MultipleChoice**

Which of the following sequence is correct?

Options:

A- API Client implementes logic to call an API >> API Consumer requests access to API >> API Implementation routes the request to >> API

B- API Consumer requests access to API >> API Client implementes logic to call an API >> API routes the request to >> API Implementation

C- API Consumer implementes logic to call an API >> API Client requests access to API >> API Implementation routes the request to >> API

D- API Client implementes logic to call an API >> API Consumer requests access to API >> API routes the request to >> API Implementation

Answer:

B

Explanation:

Correct Answer: API Consumer requests access to API >> API Client implementes logic to

****

>> API consumer does not implement any logic to invoke APIs. It is just a role. So, the option stating 'API Consumer implementes logic to call an API' is INVALID.

>> API Implementation does not route any requests. It is a final piece of logic where functionality of target systems is exposed. So, the requests should be routed to the API implementation by some other entity. So, the options stating 'API Implementation routes the request to >> API' is INVALID

>> The statements in one of the options are correct but sequence is wrong. The sequence is given as 'API Client implementes logic to call an API >> API Consumer requests access to API >> API routes the request to >> API Implementation'. Here, the statements in the options are VALID but sequence is WRONG.

>> Right option and sequence is the one where API consumer first requests access to API on Anypoint Exchange and obtains client credentials. API client then writes logic to call an API by using the access client credentials requested by API consumer and the requests will be routed to API implementation via the API which is managed by API Manager.

Question 2

Which of the below, when used together, makes the IT Operational Model effective?

Options:

A- Create reusable assets, Do marketing on the created assets across organization, Arrange time to time LOB reviews to ensure assets are being consumed or not

B- Create reusable assets, Make them discoverable so that LOB teams can self-serve and browse the APIs, Get active feedback and usage metrics
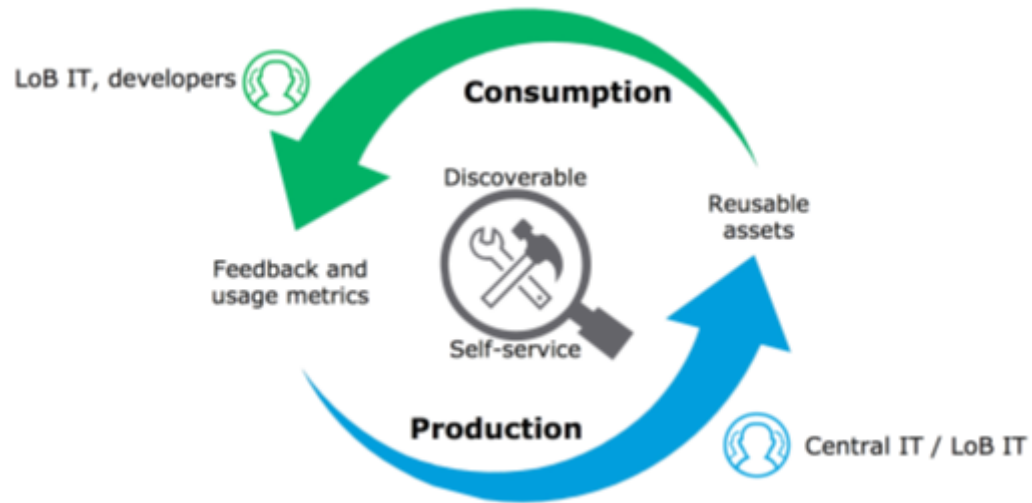
C- Create resuable assets, make them discoverable so that LOB teams can self-serve and browse the APIs

Answer:

C

Explanation:

Correct Answer:Create reusable assets, Make them discoverable so that LOB teams can

****

Question 3

A set of tests must be performed prior to deploying API implementations to a staging environment. Due to data security and access restrictions, untested APIs cannot be granted access to the backend systems, so instead mocked data must be used for these tests. The amount of available mocked data and its contents is sufficient to entirely test the API implementations with no active connections to the backend systems. What type of tests should be used to incorporate this mocked data?

Options:

A- Integration tests

B- Performance tests

C- Functional tests (Blackbox)

D- Unit tests (Whitebox)

Answer:

D

Explanation:

Correct Answer: Unit tests (Whitebox)

****

As per general IT testing practice and MuleSoft recommended practice, Integration and Performance tests should be done on full end to end setup for right evaluation. Which means all end systems should be connected while doing the tests. So, these options are OUT and we are left with Unit Tests and Functional Tests.

As per attached reference documentation from MuleSoft:

Unit Tests - are limited to the code that can be realistically exercised without the need to run it inside Mule itself. So good candidates are Small pieces of modular code, Sub Flows, Custom transformers, Custom components, Custom expression evaluators etc.

Functional Tests - are those that most extensively exercise your application configuration. In these tests, you have the freedom and tools for simulating happy and unhappy paths. You also have the possibility to create stubs for target services and make them success or fail to easily simulate happy and unhappy paths respectively.

As the scenario in the question demands for API implementation to be tested before deployment to Staging and also clearly indicates that there is enough/ sufficient amount of mock data to test the various components of API implementations with no active connections to the backend systems, Unit Tests are the one to be used to incorporate this mocked data.

Question 4

Question Type: **MultipleChoice**

A company wants to move its Mule API implementations into production as quickly as possible. To protect access to all Mule application data and metadata, the company requires that all Mule applications be deployed to the company's customer-hosted infrastructure within the corporate firewall. What combination of runtime plane and control plane options meets these project lifecycle goals?

Options:

A- Manually provisioned customer-hosted runtime plane and customer-hosted control plane

B- MuleSoft-hosted runtime plane and customer-hosted control plane

C- Manually provisioned customer-hosted runtime plane and MuleSoft-hosted control plane

D- iPaaS provisioned customer-hosted runtime plane and MuleSoft-hosted control plane

Answer:

A

Explanation:

Correct Answer: Manually provisioned customer-hosted runtime plane and customer-hosted

****

There are two key factors that are to be taken into consideration from the scenario given in the question.

>> Company requires both data and metadata to be resided within the corporate firewall

>> Company would like to go with customer-hosted infrastructure.

Any deployment model that is to deal with the cloud directly or indirectly (Mulesoft-hosted or Customer's own cloud like Azure, AWS) will have to share atleast the metadata.

Application data can be controlled inside firewall by having Mule Runtimes on customer hosted runtime plane. But if we go with Mulsoft-hosted/ Cloud-based control plane, the control plane required atleast some minimum level of metadata to be sent outside the corporate firewall.

As the customer requirement is pretty clear about the data and metadata both to be within the corporate firewall, even though customer wants to move to production as quickly as possible, unfortunately due to the nature of their security requirements, they have no other option but to go with manually provisioned customer-hosted runtime plane and customer-hosted control plane.

Question 5

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?

Options:

A- 3.0.2

B- 4.0.0

C- 3.1.0

D- 3.0.1

Answer:

B

Explanation:

Correct Answer: 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 6

What is the main change to the IT operating model that MuleSoft recommends to organizations to improve innovation and clock speed?

Options:

A- Drive consumption as much as production of assets; this enables developers to discover and reuse assets from other projects and encourages standardization

B- Expose assets using a Master Data Management (MDM) system; this standardizes projects and enables developers to quickly discover and reuse assets from other projects

C- Implement SOA for reusable APIs to focus on production over consumption; this standardizes on XML and WSDL formats to speed

up decision making

D- Create a lean and agile organization that makes many small decisions everyday; this speeds up decision making and enables each line of business to take ownership of its projects

Answer:
A

Explanation:
Correct Answer: Drive consumption as much as production of assets; this enables developers

****

>> The main motto of the new IT Operating Model that MuleSoft recommends and made popular is to change the way that they are delivered from a production model to a production + consumption model, which is done through an API strategy called API-led connectivity.

>> The assets built should also be discoverable and self-serveable for reusablity across LOBs and organization.

>> MuleSoft's IT operating model does not talk about SDLC model (Agile/ Lean etc) or MDM at all. So, options suggesting these are not valid.

https://blogs.mulesoft.com/biz/connectivity/what-is-a-center-for-enablement-c4e/

https://www.mulesoft.com/resources/api/secret-to-managing-it-projects

Question 7

An Anypoint Platform organization has been configured with an external identity provider (IdP) for identity management and client management. What credentials or token must be provided to Anypoint CLI to execute commands against the Anypoint Platform APIs?

Options:

A- The credentials provided by the IdP for identity management

B- The credentials provided by the IdP for client management

C- An OAuth 2.0 token generated using the credentials provided by the IdP for client management

D- An OAuth 2.0 token generated using the credentials provided by the IdP for identity management

Answer:

A

Explanation:

Correct Answer: The credentials provided by the IdP for identity management

****

>> There is no support for OAuth 2.0 tokens from client/identity providers to authenticate via Anypoint CLI. Only possible tokens are 'bearer tokens' that too only generated using Anypoint Organization/Environment Client Id and Secret from https://anypoint.mulesoft.com/accounts/login. Not the client credentials of client provider. So, OAuth 2.0 is not possible. More over, the token is mainly for API Manager purposes and not associated with a user. You can NOT use it to call most APIs (for example Cloudhub and etc) as per this Mulesoft Knowledge article.

>> The other option allowed by Anypoint CLI is to use client credentials. It is possible to use client credentials of a client provider but requires setting up Connected Apps in client management but such details are not given in the scenario explained in the question.

>> So only option left is to use user credentials from identify provider