

Selecting The Valid Mule-Arch-201 Exam Sample Means that You Have Passed Salesforce Certified MuleSoft Platform Architect



It Contains a pool of real Salesforce Mule-Arch-201 exam questions. This Salesforce Certified MuleSoft Platform Architect (Mule-Arch-201) practice test is compatible with every windows-based system. One downloaded does not require an active internet connection to operate. You can self-evaluate your mistakes after each Mule-Arch-201 Practice Exam attempt and work on the weak points that require more attention.

A lot of things can't be tried before buying or the product trail will charge a certain fee, but our Mule-Arch-201 exam questions are very different, you can try it free before you buy it. It's like buying clothes, you only know if it is right for you when you try it on. In the same way, in order to really think about our customers, we offer a free trial version of our Mule-Arch-201 study prep for you, so everyone has the opportunity to experience a free trial version of our Mule-Arch-201 learning materials.

>> Valid Mule-Arch-201 Exam Sample <<

Reliable Salesforce Mule-Arch-201 Exam Book, Mule-Arch-201 Test Labs

If you take a little snack, you will find that young people are now different. They made higher demands on themselves. This is a

change in one's own mentality and it is also a requirement of the times! Whether you want it or not, you must start working hard! And our Mule-Arch-201 exam materials may slightly reduce your stress. With our Mule-Arch-201 study braidumps for 20 to 30 hours, we can proudly claim that you can pass the exam easily just as a piece of cake. And as long as you try our Mule-Arch-201 practice questions, you will love it!

Salesforce Certified MuleSoft Platform Architect Sample Questions (Q60-Q65):

NEW QUESTION # 60

Which statement is true about identity management and client management on Anypoint Platform?

- A. If an external identity provider is configured, the SAML 2.0 bearer tokens issued by the identity provider cannot be used for invocations of the Anypoint Platform web APIs
- B. If an external client provider is configured, it must be configured at the Anypoint Platform organization level and cannot be assigned to individual business groups and environments
- C. Anypoint Platform supports configuring one external identity provider
- D. Both client management and identity management require an identity provider

Answer: C

Explanation:

Anypoint Platform allows organizations to integrate one external identity provider (IdP) for identity and access management (IAM), supporting SSO and centralized user authentication.

Identity Provider Limit:

Anypoint Platform supports configuring a single IdP for the organization, which can be used to authenticate all users across business groups and environments within that Anypoint organization.

of Correct Answer (C):

Configuring one IdP ensures centralized and secure identity management, aligned with MuleSoft's architecture.

of Incorrect Options:

Option A is incorrect because SAML 2.0 bearer tokens from external IdPs can indeed be used for invoking Anypoint Platform APIs.

Option B is incorrect as client providers can be assigned to specific business groups and environments.

Option D is incorrect since only identity management strictly requires an IdP; client management does not.

Reference

For further details on identity management options, consult MuleSoft documentation on Anypoint Platform's IAM capabilities.

NEW QUESTION # 61

Say, there is a legacy CRM system called CRM-Z which is offering below functions:

1. Customer creation
2. Amend details of an existing customer
3. Retrieve details of a customer
4. Suspend a customer

- A. Implement a system API named customerManagement which has all the functionalities wrapped in it as various operations/resources
- B. Implement different system APIs named createCustomerInCRMZ, amendCustomerInCRMZ, retrieveCustomerFromCRMZ and suspendCustomerInCRMZ as they are modular and has separation of concerns
- C. Implement different system APIs named createCustomer, amendCustomer, retrieveCustomer and suspendCustomer as they are modular and has separation of concerns

Answer: C

Explanation:

Correct Answer: Implement different system APIs named createCustomer, amendCustomer, retrieveCustomer and suspendCustomer as they are modular and has separation of concerns

>> It is quite normal to have a single API and different Verb + Resource combinations. However, this fits well for an Experience API or a Process API but not a best architecture style for System APIs. So, option with just one customerManagement API is not the best choice here.

>> The option with APIs in createCustomerInCRMZ format is next close choice w.r.t modularization and less maintenance but the

naming of APIs is directly coupled with the legacy system. A better foreseen approach would be to name your APIs by abstracting the backend system names as it allows seamless replacement/migration of any backend system anytime. So, this is not the correct choice too.

>> createCustomer, amendCustomer, retrieveCustomer and suspendCustomer is the right approach and is the best fit compared to other options as they are both modular and same time got the names decoupled from backend system and it has covered all requirements a System API needs.

NEW QUESTION # 62

A company stores financial transaction data in two legacy systems. For each legacy system, a separate, dedicated System API (SAPI) exposes data for that legacy system. A Process API (PAPI) merges the data retrieved from all of the System APIs into a common format. Several API clients call the PAPI through its public domain name.

The company now wants to expose a subset of financial data to a newly developed mobile application that uses a different Bounded Context Data Model. The company wants to follow MuleSoft's best practices for building out an effective application network.

Following MuleSoft's best practices, how can the company expose financial data needed by the mobile application in a way that minimizes the impact on the currently running API clients, API implementations, and support asset reuse?

- A. Add two new Experience APIs (EAPI-1 and EAPI-2).
Add Mobile PAPI-2 to expose the Intended subset of financial data as requested.
Both PAPIs access the Legacy Systems via SAPI-1 and SAPI-2.
- B. Develop and deploy a new PAPI implementation with data transformation and ... login to support this required endpoints of both mobile and web clients.
Deploy an API Proxy with an endpoint from API Manager that redirect the existing PAPI endpoints to the new PAPI.
- C. Create a new mobile Experience API (EAPI) that exposes that subset of PAPI endpoints.
Add transformation logic to the mobile Experience API implementation to make mobile data compatible with the required PAPIs.
- D. Add two new Experience APIs (EAPI-1 and EAPI-2).
Add Mobile PAPI-2 to expose the Intended subset of financial data as requested.
Both PAPIs access the Legacy Systems via SAPI-1 and SAPI-2.

Answer: A

Explanation:

To achieve the goal of exposing financial data to a new mobile application while following MuleSoft's best practices, the company should follow an API-led connectivity approach. This approach ensures minimal disruption to existing clients, maximizes reusability, and respects the separation of concerns across API layers.

of Solution:

Experience APIs for Client-Specific Requirements:

Create two new Experience APIs (EAPI-1 and EAPI-2) for the mobile application, tailored to meet the specific data and format requirements of the mobile application. These APIs encapsulate the client-specific needs and provide a custom interface without impacting other clients.

Process API Layer for Data Transformation:

By adding Mobile PAPI-2, we allow the mobile application to access the required subset of data, formatted according to the mobile application's requirements. This approach ensures that data transformation and aggregation are handled in the Process layer, maintaining consistency and reusability across different applications.

Reuse of System APIs:

Both the new Mobile PAPI-2 and existing PAPI-1 access data from System APIs (SAPI-1 and SAPI-2), which continue to expose data from each legacy system in a consistent, reusable manner. This avoids duplicating logic and ensures that data access remains centralized and manageable.

Why Option A is Correct:

Option A aligns with MuleSoft's best practices by isolating client-specific requirements in the Experience layer, utilizing Process APIs for data orchestration and transformation, and maintaining reusable System APIs for backend access.

This approach also ensures that the current API clients are not impacted, as new clients (e.g., the mobile app) interact with newly defined Experience APIs without modifying the existing API setup.

of Incorrect Options:

Option B: This option seems similar but lacks clarity on the separation of mobile-specific requirements and does not explicitly mention data transformation, which is essential in this scenario.

Option C: Creating a single mobile Experience API that exposes a subset of PAPI endpoints directly adds unnecessary complexity and may violate the separation of concerns, as transformation logic should not be in the Experience layer.

Option D: Deploying a new PAPI and using an API Proxy to redirect existing endpoints would add unnecessary complexity, disrupt

the current API clients, and increase maintenance efforts.

Reference

For additional guidance, refer to MuleSoft documentation on API-led connectivity best practices and best practices for structuring Experience, Process, and System APIs.

NEW QUESTION # 63

The responses to some HTTP requests can be cached depending on the HTTP verb used in the request. According to the HTTP specification, for what HTTP verbs is this safe to do?

- A. GET, OPTIONS, HEAD
- B. GET, PUT, OPTIONS
- C. GET, HEAD, POST
- D. PUT, POST, DELETE

Answer: A

Explanation:

Correct Answer: GET, OPTIONS, HEAD

<http://restcookbook.com/HTTP%20Methods/idempotency/>

NEW QUESTION # 64

Refer to the exhibit.

A RAML definition has been proposed for a new Promotions Process API, and has been published to Anypoint Exchange. The Marketing Department, who will be an important consumer of the Promotions API, has important requirements and expectations that must be met.

What is the most effective way to use Anypoint Platform features to involve the Marketing Department in this early API design phase?

- A) Ask the Marketing Department to interact with a mocking implementation of the API using the automatically generated API Console
- B) Organize a design workshop with the DBAs of the Marketing Department in which the database schema of the Marketing IT systems is translated into RAML
- C) Use Anypoint Studio to Implement the API as a Mule application, then deploy that API implementation to CloudHub and ask the Marketing Department to interact with it
- D) Export an integration test suite from API designer and have the Marketing Department execute the tests in that suite to ensure they pass

- A. Option B
- B. Option D
- C. Option A
- D. Option C

Answer: C

Explanation:

Correct Answer: Ask the Marketing Department to interact with a mocking implementation of the API using the automatically generated API Console.

As per MuleSoft's IT Operating Model:

>> API consumers need NOT wait until the full API implementation is ready.

>> NO technical test-suites needs to be shared with end users to interact with APIs.

>> Anypoint Platform offers a mocking capability on all the published API specifications to Anypoint Exchange which also will be rich in documentation covering all details of API functionalities and working nature.

>> No needs of arranging days of workshops with end users for feedback.

API consumers can use Anypoint Exchange features on the platform and interact with the API using its mocking feature. The feedback can be shared quickly on the same to incorporate any changes.

NEW QUESTION # 65

.....

