

Quiz 2026 Unparalleled Salesforce Mule-Arch-201: Latest Study Salesforce Certified MuleSoft Platform Architect Questions



2026 Latest ActualTorrent Mule-Arch-201 PDF Dumps and Mule-Arch-201 Exam Engine Free Share:
<https://drive.google.com/open?id=1RfqDihwS8RGpIUYWgx93EVSIObtetG53>

At the same time, Mule-Arch-201 study material also has a timekeeping function that allows you to be cautious and keep your own speed while you are practicing, so as to avoid the situation that you can't finish all the questions during the exam. With Mule-Arch-201 Learning Materials, you only need to spend half your money to get several times better service than others. And you can get the Mule-Arch-201 certification with little effort and money.

Overall, Mule-Arch-201 is committed to helping candidates achieve success in the Salesforce Mule-Arch-201 exam. Their goal is to save students time and money, and they guarantee that candidates who use their product will pass the Mule-Arch-201 Exam on their first try. With the right study material and support team, passing the exam at the first attempt is an achievable goal.

>> Latest Study Mule-Arch-201 Questions <<

Key Features Of Desktop Salesforce Mule-Arch-201 Practice Exam Software

All Mule-Arch-201 online tests begin somewhere, and that is what the Mule-Arch-201 training course will do for you: create a foundation to build on. Study guides are essentially a detailed Mule-Arch-201 tutorial and are great introductions to new Mule-Arch-201 training courses as you advance. The content is always relevant, and compound again to make you pass your Mule-Arch-201 Exams on the first attempt. You will frequently find these Mule-Arch-201 PDF files downloadable and can then archive or print them for extra reading or studying on-the-go.

Salesforce Certified MuleSoft Platform Architect Sample Questions (Q67-Q72):

NEW QUESTION # 67

A client has several applications running on the Salesforce service cloud. The business requirement for integration is to get daily data changes from Account and Case Objects. Data needs to be moved to the client's private cloud AWS DynamoDB instance as a single JSON and the business foresees only wanting five attributes from the Account object, which has 219 attributes (some custom) and eight attributes from the Case Object.

What design should be used to support the API/ Application data model?

- A. Create separate entities for Account and Case Objects by mimicking all the attributes in SAPI, which are combined by the PAPI and filtered to provide JSON output containing 13 attributes.
- B. Start implementing an Enterprise Data Model by defining enterprise Account and Case Objects and implement SAPI and DynamoDB tables based on the Enterprise Data Model.
- C. Request client's AWS project team to replicate all the attributes and create Account and Case JSON table in DynamoDB. Then create separate entities for Account and Case Objects by mimicking all the attributes in SAPI to transfer JSON data to

DynamoDB for respective Objects

- **D. Create separate entities for Account with five attributes and Case with eight attributes in SAPI, which are combined by the PAPI to provide JSON output containing 13 attributes.**

Answer: D

Explanation:

Understanding the Requirements:

The business needs to transfer daily data changes from the Salesforce Account and Case objects to AWS DynamoDB in a single JSON format.

Only a subset of attributes (5 from Account and 8 from Case) is required, so it is not necessary to include all 219 attributes of the Account object.

Design Approach:

A System API (SAPI) should be created for each Salesforce object (Account and Case), exposing only the required fields (5 attributes for Account and 8 for Case).

A Process API (PAPI) can be used to aggregate and transform the data from these SAPIs, combining the 13 selected attributes from Account and Case into a single JSON structure for DynamoDB.

Evaluating the Options:

Option A: Mimicking all attributes in the SAPI is inefficient and unnecessary, as only 13 attributes are required.

Option B: Replicating all attributes in DynamoDB is excessive and would result in higher storage and processing costs, which is unnecessary given the requirement for only a subset of attributes.

Option C: Implementing an Enterprise Data Model could be useful in broader data management but is not required here, as the focus is on a lightweight integration.

Option D (Correct Answer): Creating separate entities in SAPI for Account and Case with only the required attributes and using the PAPI to aggregate them into a single JSON is the most efficient and meets the requirements effectively.

Conclusion:

Option D is the best choice as it provides a lightweight, efficient design that meets the requirements by transferring only the necessary attributes and minimizing resource use.

Refer to MuleSoft's best practices for API-led connectivity and data modeling to structure SAPIs and PAPIs efficiently.

NEW QUESTION # 68

An organization has built an application network following the API-led connectivity approach recommended by MuleSoft. To protect the application network against attacks from malicious external API clients, the organization plans to apply JSON Threat Protection policies.

To which API-led connectivity layer should the JSON Threat Protection policies most commonly be applied?

- A. System layer
- **B. Experience layer**
- C. All layers
- D. Process layer

Answer: B

Explanation:

Understanding JSON Threat Protection Policies:

JSON Threat Protection policies are used to protect APIs from attacks that exploit JSON payloads, such as oversized payloads, deeply nested objects, and excessive array elements. This helps prevent Denial of Service (DoS) attacks and other malicious payload-related threats.

These policies are typically applied to safeguard APIs that are directly exposed to external clients, where the risk of receiving malicious payloads is highest.

API-led Connectivity Layers:

Experience Layer: This layer is designed to expose APIs to end-users or external API clients, often acting as the interface that interacts with users or applications.

Process Layer: This layer is used for orchestration and aggregation of data from various System APIs, typically operating within a trusted environment and not directly exposed to external clients.

System Layer: This layer provides access to backend systems and databases, often within the organization's secure environment and not directly accessible to external clients.

Evaluating the Options:

Option A (All layers): While JSON Threat Protection can technically be applied to all layers, it is most commonly applied at the Experience layer, where APIs are exposed to external traffic and are more vulnerable to attacks.

Option B (System layer): The System layer is generally not exposed to external clients directly, so JSON Threat Protection is less critical here.

Option C (Process layer): Similar to the System layer, the Process layer is typically internal and not exposed directly to external clients, so JSON Threat Protection is less commonly applied.

Option D (Correct Answer): The Experience layer is the correct answer because it is the layer that directly interacts with external clients, making it the primary target for malicious payloads. Applying JSON Threat Protection here effectively protects the application network from external threats.

Conclusion:

Option D is the correct answer, as the Experience layer is the most common layer for applying JSON Threat Protection policies to protect against external attacks.

For further reference, consult MuleSoft's documentation on API security policies and best practices for securing APIs at the Experience layer.

NEW QUESTION # 69

A Mule 4 API has been deployed to CloudHub and a Basic Authentication - Simple policy has been applied to all API methods and resources. However, the API is still accessible by clients without using authentication.

How is this possible?

- A. No... for client applications have been created of this API
- B. The APE Router component is pointing to the incorrect Exchange version of the APT
- C. The Autodiscovery element is not present, in the deployed Mule application
- D. One of the application's CloudHub workers restarted

Answer: C

Explanation:

When a Basic Authentication policy is applied to an API on CloudHub but clients can still access the API without authentication, the likely cause is a missing Autodiscovery element. Here's how this affects API security:

Autodiscovery in MuleSoft:

The Autodiscovery element is essential for linking an API implementation deployed in CloudHub with its API instance defined in API Manager. This connection allows the policies applied in API Manager, such as Basic Authentication, to be enforced on the deployed API.

Why Option B is Correct:

Without Autodiscovery, the deployed application does not "know" about the policies configured in API Manager, resulting in unrestricted access. Adding Autodiscovery enables the API to enforce the policies correctly.

of Incorrect Options:

Option A (incorrect Exchange version) would not cause bypassing of security policies.

Option C (missing client applications) does not impact authentication policy enforcement.

Option D (worker restart) is irrelevant to policy enforcement.

Reference

Refer to MuleSoft documentation on Autodiscovery configuration and linking API Manager policies for additional information on setting up secure API policies.

NEW QUESTION # 70

A company is building an application network using MuleSoft's recommendations for various API layers.

What is the main (default) role of a process API in an application network?

- A. To secure and optimize the data synchronization processing of large data dumps between back-end systems
- B. To automate parts of business processes by coordinating and orchestrating the invocation of other APIs in the application network
- C. To secure, Manage, and process communication with specific types of end-user client applications or devices in the application network
- D. To manage and process the secure direct communication between a back-end system and an end-user client of mobile device in the application network

Answer: B

Explanation:

Role of Process API in API-led Connectivity:

In MuleSoft's API-led connectivity approach, a Process API is used to coordinate, aggregate, and orchestrate data from various System APIs. It is primarily responsible for implementing business logic that spans multiple backend systems or entities, transforming and combining data as needed to support business processes.

Process APIs are not directly exposed to end-user clients; rather, they work between System APIs and Experience APIs, providing business logic and orchestration capabilities.

Evaluating the Options:

Option A: Process APIs are not typically responsible for large data dumps or data synchronization. That function would be handled by a System API or a specialized batch process.

Option B: Managing direct, secure communication between back-end systems and end-user clients is typically the role of Experience APIs rather than Process APIs.

Option C (Correct Answer): Process APIs are designed to coordinate and orchestrate calls to multiple other APIs in the network, which supports the automation of business processes.

Option D: Securing and managing communication with end-user clients is typically the role of Experience APIs, not Process APIs.

Conclusion:

Option C is the correct answer, as the main role of a Process API is to coordinate and orchestrate interactions between other APIs, enabling business processes to function seamlessly across multiple systems.

Refer to MuleSoft's API-led connectivity documentation for further explanation of the roles and responsibilities of Process APIs in an application network.

NEW QUESTION # 71

An eCommerce company is adding a new Product Details feature to their website. A customer will launch the product catalog page, a new Product Details link will appear by product where they can click to retrieve the product detail description. Product detail data is updated with product update releases, once or twice a year. Presently the database response time has been very slow due to high volume.

What action retrieves the product details with the lowest response time, fault tolerant, and consistent data?

- A. Select the product details from a database and put them in Anypoint MQ; the Anypoint MO subscriber will receive the product details and return them within the API response
- B. Select the product details from a database and return them within the API response
- C. Select the product details from a database in a Cache scope and return them within the API response
- **D. Use an object store to store and retrieve the product details originally read from a database and return them within the API response**

Answer: D

Explanation:

Scenario Analysis:

The eCommerce company's Product Details feature requires low response time and consistent data for a feature where data rarely changes (only once or twice a year).

The database response time is slow due to high volume, so querying the database directly on each request would lead to poor performance and higher response times.

Optimal Solution Requirements:

Low Response Time: Data retrieval should be fast and not depend on database performance.

Fault Tolerance and Data Consistency: Cached or stored data should be consistent and resilient in case of database unavailability, as the product details data changes infrequently.

Evaluating the Options:

Option A: Using a Cache scope would temporarily store the product details in memory, which could improve performance but might not be suitable for infrequent updates (only twice a year), as cache expiration policies typically require shorter durations.

Option B: Storing product details in Anypoint MQ and then retrieving it through a subscriber is not suitable for this use case.

Anypoint MQ is better for messaging rather than as a data storage mechanism.

Option C (Correct Answer): Using an object store to store and retrieve product details is ideal. Object stores in MuleSoft are designed for persistent storage of key-value pairs, which allows storing data retrieved from the database initially. This provides quick, consistent access without querying the database on every request, aligning with requirements for low response time, fault tolerance, and data consistency.

Option D: Selecting data directly from the database for each request would not meet the performance requirement due to known slow response times from the database.

Conclusion:

Option C is the best answer, as using an object store allows caching the infrequently updated product details. This approach reduces the dependency on the database, significantly improving response time and ensuring consistent data.

Refer to MuleSoft documentation on Object Store v2 and best practices for data caching to implement this solution effectively.

• • • • •

Training Mule-Arch-201 Solutions: <https://www.actualtorrent.com/Mule-Arch-201-questions-answers.html>

Prominent Features of ActualTorrent Salesforce Mule-Arch-201 Practice Questions

Besides, you have varied choices for there are three versions of our Mule-Arch-201 practice materials.

- [illegible]

P.S. Free 2026 Salesforce Mule-Arch-201 dumps are available on Google Drive shared by ActualTorrent:
<https://drive.google.com/open?id=1RfqDihwS8RGpIUYWgx93EVSIObtetG53>