Validate your MuleSoft-Integration-Associate Exam Preparation with MuleSoft-Integration-Associate Practice Test (Online & Offline) [Q15-Q35

Validate your MuleSoft-Integration-Associate Exam Preparation with MuleSoft-Integration-Associate Practice Test (Online & Offline) [Q15-Q35]

Validate your MuleSoft-Integration-Associate Exam Preparation with MuleSoft-Integration-Associate Practice Test (Online & Offline) Get all the Information About Salesforce MuleSoft-Integration-Associate Exam 2025 Practice Test Questions
QUESTION 15

According to the National Institute of Standards and Technology (NIST) which cloud computing deployment model describes a composition of two or more distinct clouds that support data and application portability?
* Public cloud
* Private cloud
* Community cloud
* Hybrid cloud

According to the National Institute of Standards and Technology (NIST), a hybrid cloud is a cloud computing deployment model that consists of a combination of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability.
Here&#8217;s a detailed explanation:

* Hybrid Cloud:

* Definition: Combines on-premises infrastructure (private cloud) with public cloud services, allowing data and applications to be shared between them.

* Portability: Ensures seamless data and application movement between the private and public clouds.

* Integration: Uses technology such as VPNs, APIs, or hybrid cloud management tools to integrate the environments.

* Benefits:

* Flexibility: Offers greater flexibility in deploying workloads where they are most appropriate.

* Scalability: Provides scalability by leveraging public cloud resources while maintaining control over critical applications in a private cloud.

* Cost Efficiency: Optimizes costs by utilizing public cloud resources for less sensitive workloads while keeping sensitive data in a private cloud.

References

* NIST Cloud Computing:NIST Definition of Cloud Computing

* Hybrid Cloud: What is Hybrid Cloud?

QUESTION 16

According to MuleSoft a synchronous invocation of a RESTful API using HTTP to gel an individual customer record from a single system is an example of which system integration interaction pattern?
* Request-Reply
* Batch
* Multicast
* One-way
In system integration, different interaction patterns are used depending on the communication requirements between systems. For a synchronous invocation of a RESTful API using HTTP to get an individual customer record from a single system, the Request-Reply pattern is used. Here&#8217;s a detailed explanation:

* Request-Reply Pattern:

* Definition: This pattern involves a client sending a request to a server and waiting for a reply. The communication is synchronous, meaning the client waits for the server to process the request and send back the response.

* Typical Use Case: It is used when immediate feedback is required from the server, such as retrieving a specific customer record.

* RESTful API and HTTP:

* Synchronous Communication: HTTP is inherently synchronous, making it suitable for Request-Reply interactions where the client expects an immediate response.

* Data Retrieval: Commonly used for GET requests in RESTful APIs to retrieve data from a server.

* Example:

* Scenario: A client application requests customer details by making a GET request to a RESTful API endpoint. The server processes the request and returns the customer record.

References

* MuleSoft Documentation: Integration Patterns

* REST API Design: Request-Reply Pattern

QUESTION 17

An IT integration team followed an API-led connectivity approach to implement an order-fulfillment business process It created an order processing API that coordinates stateful interactions with a variety of microservices that validate, create and fulfill new product orders.

Which interaction composition pattern did the integration architect who designed this order processing API use?
* Multicasting
* Orchestration
* Streaming
* Aggregation
In an API-led connectivity approach, different APIs are layered to provide modular and reusable services. For an order processing API that coordinates stateful interactions with various microservices, the integration architect used the orchestration interaction composition pattern. Here&#8217;s a step-by-step explanation:

* Understanding Orchestration:

* Definition: Orchestration involves coordinating multiple services to achieve a complex business workflow. Unlike choreography, which relies on each service knowing its part, orchestration uses a central controller to manage the interactions.

* Role of the Orchestrator: The orchestrator manages the execution sequence, handles the state, and ensures all the necessary steps are completed successfully.

* Order Processing API:

* API-Led Connectivity: An order processing API, following API-led connectivity, sits in the Process layer, handling complex business processes and logic.

* Stateful Interactions: Orchestration is particularly suitable for stateful interactions where the process needs to remember the state between steps, such as validating an order, creating it, and fulfilling it.

* Implementation Steps:

* Microservices Interaction: The order processing API interacts with various microservices:

* Validation Microservice: Checks the validity of the order details.

* Creation Microservice: Creates the order in the system.

* Fulfillment Microservice: Manages the order fulfillment process.

* Coordination: The API orchestrates these steps, ensuring each one completes successfully before moving to the next, handling exceptions, and maintaining the state of the process.

References

* MuleSoft Documentation: Orchestration Pattern

* API-led Connectivity: MuleSoft API-led Connectivity

QUESTION 18

Which productivity advantage does Anypoint Platform have to both implement and manage an API?
* Automatic API specification generation
* Automatic API governance
* Automatic API proxy generation
* Automatic API semantic versioning
Anypoint Platform, MuleSoft&#8217;s unified platform for API design and integration, offers several productivity advantages for both implementing and managing APIs. Among these features, automatic API proxy generation is particularly beneficial. Here&#8217;s a step-by-step explanation:

* API Implementation:

* Design Center: In the Design Center, users can create API specifications using RAML or OAS.

This environment provides tools to design and document APIs effectively.

* Exchange: After defining the API, it can be published to Anypoint Exchange where it can be shared and discovered by others within the organization.

* Automatic API Proxy Generation:

* When an API is published to Exchange, Anypoint Platform allows for the automatic creation of an API proxy. An API proxy acts as a facade for your backend API, providing a layer of abstraction and security.

* Advantages:

* Security: Protects backend services by exposing only necessary endpoints and handling authentication, authorization, and rate limiting.

* Traffic Management: Helps in managing traffic through throttling and caching.

* Monitoring: Facilitates monitoring and logging to track API usage and performance.

* This automation saves time and reduces the complexity of manual proxy setup, allowing developers to focus on core business logic.

* API Management:

* API Manager: Provides a dashboard to manage API policies, versions, and SLA tiers. Users can apply security policies, monitor

traffic, and analyze API usage.

* Monitoring: Integrated with Anypoint Monitoring, users get insights into API performance and health, enabling proactive management.

References

* MuleSoft Documentation: API Proxies

* MuleSoft Anypoint Platform Overview: Anypoint Platform

QUESTION 19

An organization&#8217;s IT learn follows an API-led connectivity approach and must use Anypomt Platform to implement a System API that securely accesses customer data The organization uses Salesforce as the system of record for all customer data and its most important objective is to reduce the overall development time to release the System API The team&#8217;s integration architect has identified four different approaches to access the customer data from within the implementation of the System API by using different Anypoint Connectors that all meet the technical requirements of the project Which approach should the team choose to meet the organization&#8217;s objective to reduce the time to develop and release the System API?
* Use the Anypoint Connector for Salesforce to connect to the Salesforce APIs to directly access the customer data
* Use the Anypoint Connector for HTTP to connect to the Salesforce APIs to directly access the customer data
* Use the Anypoint Connector for Database to connect to a MySQL database to access a copy of the customer data
* Use the Anypoint Connector for FTP to download a file containing a recent near-real time extract of the customer data

In an API-led connectivity approach, using the most efficient method to access system data can significantly reduce development time. Here&#8217;s why using the Anypoint Connector for Salesforce is the best approach:

* Direct Access:

* Salesforce APIs: The Anypoint Connector for Salesforce provides direct access to Salesforce APIs, allowing the System API to retrieve customer data efficiently and securely.

* No Middleware: Directly accessing Salesforce eliminates the need for intermediary steps, reducing complexity and potential points of failure.

* Reduced Development Time:

* Out-of-the-Box Functionality: The connector offers pre-built operations and functionalities tailored for Salesforce, speeding up development.

* Configuration Over Coding: Using the connector involves minimal configuration compared to coding custom integration logic, leading to faster implementation.

* Security:

* Built-in Security: The connector handles authentication and authorization, ensuring secure data access in line with Salesforce security protocols.

* Alternative Approaches:

* HTTP Connector: While functional, it requires more custom handling for Salesforce API interactions and error management.

* Database Connector: Accessing a database copy of Salesforce data may involve data synchronization challenges and does not provide real-time data.

* FTP Connector: Using FTP for data extracts is less efficient and introduces delays in accessing up-to-date information.

References

* MuleSoft Documentation: Salesforce Connector

* API-led Connectivity: MuleSoft API-led Connectivity

QUESTION 20

An organization is choosing between API-led connectivity and other integration approaches According to MuleSoft which business benefit is associated with an API-led connectivity approach using Anypoint Platform?
* Higher outcome repeatability through centralized development
* Greater project predictability through tight coupling of systems
* Improved security through adoption of monolithic architectures
* Increased developer productivity through self-service of API assets
API-led connectivity is an approach that emphasizes the reuse of APIs to enhance agility and productivity.

Here&#8217;s a detailed explanation of the associated business benefits:

* Self-Service of API Assets:

* Definition: API-led connectivity enables developers to discover, access, and use APIs through a centralized platform like Anypoint Exchange, promoting self-service.

* Productivity: Developers can quickly find and integrate existing APIs, reducing the time and effort required to build new functionalities from scratch.

* Business Benefits:

* Reusability: Encourages the reuse of APIs across projects, leading to faster development cycles and reduced duplication of efforts.

* Agility: Enhances the ability to respond to changing business needs by providing a flexible and modular integration framework.

* Scalability: Facilitates the scaling of integration solutions as business requirements grow.

References

* API-led Connectivity: MuleSoft API-led Connectivity

* Business Benefits: Why API-led Connectivity?

QUESTION 21

What are two reasons why a typical MuleSoft customer favors a MuleSoft-hosted AnypointPlatform runtime plane over a customer-hosted runtime for its Mule application deployments? (Choose two.)

* Reduced time-to-market for the first application
* Reduced application latency
* Reduced IT operations effort
* increased application isolation
* Increased application throughput

Choosing a MuleSoft-hosted Anypoint Platform runtime plane offers several advantages, particularly in terms of deployment efficiency and operational management. Here&#8217;s a detailed explanation of the selected reasons:

* Reduced Time-to-Market for the First Application:

* Pre-Configured Environment: MuleSoft-hosted Anypoint Platform provides a ready-to-use environment, which accelerates the deployment process.

* Ease of Use: Developers can quickly set up and deploy applications without the need for extensive infrastructure setup and configuration.

* Reduced IT Operations Effort:

* Managed Services: MuleSoft handles the infrastructure management, including updates, scaling, and maintenance, reducing the operational burden on the IT team.

* Focus on Development: IT teams can focus on developing and optimizing applications rather than managing runtime environments.

References

* MuleSoft Documentation: Anypoint Platform Deployment Models

* Benefits of MuleSoft-Hosted Runtime: CloudHub Advantages

QUESTION 22

According to MuleSoftwhich principle Is common to both Service Oriented Architecture (SOA) and API-Jed connectivity approaches*?

* Service interdependence
* Service statefulness
* Service reusability
* Service centralization

Both Service-Oriented Architecture (SOA) and API-led connectivity emphasize the principle of service reusability. Here&#8217;s a detailed explanation:

* Service Reusability:

* Definition: Service reusability is the principle where services are designed to be reusable across different applications and use cases.

* SOA: In SOA, services are modular components that can be reused in various business processes, reducing redundancy and promoting efficient service composition.

* API-led Connectivity: This approach also stresses creating reusable APIs (System APIs, Process APIs, Experience APIs) that can

be leveraged across multiple projects and applications.

* Benefits:

* Efficiency: Reduces development time and effort by reusing existing services.

* Consistency: Ensures consistency in business logic and data access across different applications.

* Scalability: Facilitates scaling by using standardized and reusable services/APIs.

References

* MuleSoft Documentation: SOA vs. API-led Connectivity

* Service Reusability: Principles of Service Reusability

QUESTION 23

In preparation for a digital transformation initiative an organization is reviewing related IT integration projects that failed for various reasons According to MuleSoft&#8217;s surveys of global IT leaders, what is a common cause of IT project failure that this organization may likely discover in its assessment?
* Lack of alignment around business outcomes
* Reliance on an Integration-Platform-as-a-Service (iPaaS)
* Following an Agile delivery methodology
* Spending too much time on enablement
One common cause of IT project failure identified by MuleSoft&#8217;s surveys of global IT leaders is the lack of alignment around business outcomes. Here&#8217;s a detailed explanation:

* Lack of Alignment:

* Definition: This occurs when IT projects are not clearly linked to the organization&#8217;s strategic goals and business objectives.

* Impact: Misalignment can lead to projects that do not deliver the intended business value, resulting in wasted resources and failed initiatives.

* Common Causes:

* Poor Communication: Lack of effective communication between business stakeholders and IT teams can lead to misunderstandings and misaligned priorities.

* Undefined Objectives: Projects without clearly defined business outcomes and success metrics struggle to demonstrate value and justify investments.

* Solution:

* Business-IT Collaboration: Foster strong collaboration between business and IT to ensure projects are aligned with strategic goals.

* Outcome-Focused Planning: Define clear business outcomes and success criteria at the outset of each project.

References

* MuleSoft Surveys: State of IT Digital Transformation

* Causes of IT Project Failure: Common Reasons for Project Failure

QUESTION 24

According to MuleSoft which system integration term describes the method, format and protocol used for communication between two systems?
* Interaction
* Interface
* Message
* Component
In system integration, the term &#8220;interface&#8221; describes the method, format, and protocol used for communication between two systems. Here&#8217;s a detailed explanation:

* Interface:

* Definition: An interface defines the point of interaction between two systems, specifying how data is exchanged, including the communication method, data format, and protocol.

* Components: Typically includes API endpoints, data formats (e.g., JSON, XML), communication protocols (e.g., HTTP, HTTPS), and authentication mechanisms.

* Importance:

* Standardization: Ensures that different systems can communicate effectively by adhering to predefined standards and protocols.

* Interoperability: Facilitates seamless interaction and data exchange between disparate systems, enhancing overall integration.

* Examples:

* RESTful APIs: Define interfaces using HTTP/HTTPS and data formats like JSON or XML.

* SOAP Web Services: Use XML-based messages and protocols such as HTTP or HTTPS for communication.

References

* MuleSoft Documentation: System Integration Concepts

* Interface Design: API Interface

QUESTION 25

In which order are the API Client API Implementation and API Interface components called m a typical REST request?
* API Client > API Interface > API Implementation
* API Interface > API Client > API Implementation
* API Implementation > API Interface > API Client
* API Client > API Implementation > API Interface

In a typical REST request, the components are called in a specific order to handle the client&#8217;s request and provide the response. Here&#8217;s the order and detailed explanation:

* API Client:

* Initiates Request: The client (e.g., web or mobile application) sends a request to the API endpoint.

* API Interface:

* Gateway/Proxy: This layer is typically managed by an API gateway or proxy, which handles the incoming request, applies security policies, and routes it to the appropriate backend service.

* Responsibilities: Includes request validation, rate limiting, authentication, and authorization.

* API Implementation:

* Backend Service: The actual implementation of the API logic resides here. It processes the request, interacts with the necessary databases or external services, and generates the response.

References

* REST API Design:RESTful Web Services

* API Gateway: What is an API Gateway?

QUESTION 26

An organization is not meeting its growth and innovation objectives because IT cannot deliver projects fast enough to keep up with the pace of change required by the business.

According to MuleSoft&#8217;s IT delivery and operating model which step should the organization take to solve this problem?
* Adopt a new approach that decouples core IT projects from the innovation that happens within each line of business
* Switch from a design-first to a code-first approach for IT development
* Modify IT governance and security controls so that line of business developers can have direct access to theorganization&#8217;s systems of record
* Hire more IT developers, architects, and project managers to increase IT delivery

MuleSoft&#8217;s IT delivery and operating model suggests modernizing IT practices to better support business growth and innovation. Here&#8217;s a detailed explanation:

* Decoupling Core IT Projects:

* Definition: Decoupling involves separating the core IT systems and projects from the innovative and experimental projects conducted by various lines of business.

* Benefits:

* Agility: Enables lines of business to innovate rapidly without being held back by the constraints of core IT systems.

* Focus: Allows core IT to focus on maintaining and enhancing critical systems while business units can experiment and deploy new solutions more quickly.

* Implementation:

* API-led Connectivity: By using an API-led connectivity approach, core IT can expose reusable APIs and services that business units can leverage for their innovation efforts.

* Governance and Security: Ensuring that proper governance and security measures are in place to protect core systems while allowing flexibility for innovation.

* Outcome:

* Faster Delivery: Speeds up the delivery of new features and solutions, aligning with business needs and market demands.

* Enhanced Collaboration: Facilitates better collaboration between IT and business units, driving overall organizational growth.

References

* MuleSoft Whitepaper: API-led Connectivity

* IT Operating Model: Transforming IT Delivery

QUESTION 27

During a planning session with the executive leadership, the development team director presents plans for a new API to expose the data in the company&#8217;s order database. An earlier effort to build an API on top of this data failed, so the director is recommending a design-first approach.

Which characteristics of a design-first approach will help make this API successful?
* Building MUnit tests so administrators can confirm code coverage percentage during deployment
* Publishing the fully implemented API to Exchange so all developers can reuse the API
* Developing a specification so consumers can test before the implementation is built
* Adding global policies to the API so all developers automatically secure the implementation before coding anything
A design-first approach emphasizes creating the API specification before implementation, ensuring better alignment with consumer needs and reducing the risk of project failure. Here&#8217;s a detailed explanation:

* API Specification:

* Definition: An API specification is a detailed, formal description of the API&#8217;s endpoints, request/response formats, and protocols.

* Consumer Testing: Allows API consumers (developers) to understand, test, and provide feedback on the API design before actual development begins.

* Advantages:

* Early Feedback: Consumers can test the API design using mock services or tools like API Designer and provide feedback, ensuring the API meets their requirements.

* Reduced Rework: Identifies potential issues and design flaws early, reducing costly changes during the implementation phase.

* Documentation: Provides comprehensive documentation that aids in the development and future maintenance of the API.

References

* Design-First Approach: Design-First API Development

* API Mocking: API Designer Mocking Service

QUESTION 28

Which component of AnypointPlatform belongs to the platform control plane&#8221;?
* Runtime Replica
* Anypoint Connectors
* API Manager
* Runtime Fabric

In Anypoint Platform, the control plane is responsible for managing and controlling the various components and services that make up the platform. API Manager is part of the control plane, providing centralized management of APIs. Here&#8217;s a detailed explanation:

* Control Plane:

* Definition: The control plane in Anypoint Platform is responsible for the management, monitoring, and control of APIs, applications, and other platform resources.

* Components: Includes tools for API management, analytics, security, and governance.

* API Manager:

* Purpose: Allows users to manage API policies, monitor API usage, and secure APIs. It provides a centralized interface for managing the entire lifecycle of APIs.

* Features:

* Policy Enforcement: Apply security policies, rate limiting, and other governance rules.

* Analytics and Monitoring: Track API performance, usage statistics, and detect anomalies.

* Access Control: Manage user access and permissions for APIs.

References

* MuleSoft Documentation: API Manager

* Anypoint Platform Overview: Anypoint Platform

QUESTION 29

A developer is examining the responses from a RESTful web service that is compliant with the Hypertext Transfer Protocol (HTTP/1 1) as defined by the Internet Engineering Task Force (IETF).

In this HTTP/1 1-comphanl web service, which class of HTTP response status codes should be specified to indicate when client requests are successfully received, understood and accepted by the web service?
* 2xx
* 3xx
* 5xx
* 4xx

In HTTP/1.1, response status codes are categorized to indicate the result of a client&#8217;s request. Here&#8217;s a detailed explanation of the 2xx class of HTTP response status codes:

* 2xx Success Codes:

* Definition: The 2xx class of status codes indicates that the client&#8217;s request was successfully received, understood, and accepted by the server.

* Common Codes:

* 200 OK: The request has succeeded.

* 201 Created: The request has been fulfilled and resulted in a new resource being created.

* 202 Accepted: The request has been accepted for processing, but the processing is not complete.

* 204 No Content: The server successfully processed the request, but there is no content to

* return.

* Importance:

* Client Acknowledgment: These codes inform the client that their request was processed successfully, enabling appropriate client-side actions.

* RESTful Standards: Adhering to these standards ensures consistent and predictable API behavior.

References

* IETF RFC 7231: HTTP/1.1 Semantics and Content

* HTTP Status Codes:HTTP Status Code Definitions

QUESTION 30

An application load balancer routes requests to a RESTful web API secured by Anypomt Flex Gateway Which protocol is involved in the communication between the load balancer and the Gateway?
* LDAP
* HTTPS
* SFTP
* SMTP

In scenarios where an application load balancer routes requests to a RESTful web API secured by Anypoint Flex Gateway, HTTPS is the protocol used. Here&#8217;s a detailed explanation:

* HTTPS Protocol:

* Definition: HTTPS (HyperText Transfer Protocol Secure) is an extension of HTTP that provides secure communication over a computer network.

* Encryption: It uses SSL/TLS to encrypt the data exchanged between the client and server, ensuring privacy and data integrity.

* Load Balancer to Gateway Communication:

* Secure Communication: The load balancer routes incoming requests to the API Gateway using HTTPS, ensuring that the data is encrypted and secure.

* Standard Practice: HTTPS is the standard protocol for securing API communications, protecting against eavesdropping and man-in-the-middle attacks.

References

* HTTPS Protocol:What is HTTPS?

* API Gateway Security: Anypoint Flex Gateway