

高品質なSalesforce-MuleSoft-Associate認定資格試験一回合格-効果的なSalesforce-MuleSoft-Associateトレーニングサンプル



Xhs1991が提供した商品の品質が高く、頼られているサイトでございます。購入前にネットで部分なSalesforce-MuleSoft-Associate問題集を無料でダウンロードしてあとで弊社の商品を判断してください。Xhs1991は君のSalesforce-MuleSoft-Associate試験に100%の合格率を保証いたします。迷ってないください。

なぜ弊社は試験に失敗したら全額で返金することを承諾していますか。弊社のSalesforceのSalesforce-MuleSoft-Associateソフトを通してほとんどの人が試験に合格したのは我々の自信のある原因です。SalesforceのSalesforce-MuleSoft-Associate試験は、ITに関する仕事に就職している人々にとって、重要な能力への証明ですが、難しいです。だから、弊社の専門家たちは尽力してSalesforceのSalesforce-MuleSoft-Associate試験のための資料を研究します。あなたに提供するソフトはその中の一部です。

>> Salesforce-MuleSoft-Associate認定資格試験 <<

Salesforce-MuleSoft-Associateトレーニングサンプル、Salesforce-MuleSoft-Associate日本語的中対策

Xhs1991は我々が研究したトレーニング資料を無料で更新します。それはあなたがいつでも最新のSalesforce-MuleSoft-Associate試験トレーニング資料をもらえるということです。Salesforce-MuleSoft-Associate認定試験の目標が変更されれば、Xhs1991が提供した勉強資料も変化に追従して内容を変えます。Xhs1991は各受験生のニーズを知っていて、あなたがSalesforce-MuleSoft-Associate認定試験に受かることに有効なヘルプを差し上げます。あなたが首尾よく試験に合格するように、我々は最も有利な価格と最高のクオリティーを提供して差し上げます。

Salesforce Salesforce-MuleSoft-Associate 認定試験の出題範囲:

トピック	出題範囲
トピック 1	API管理におけるAnypoint Platformのコンポーネントと利点を説明する：この試験セクションは統合アーキテクト向けに設計されており、MuleSoftのAPI管理アプローチに焦点を当てています。ユニバーサルAPI管理を含む、ライフサイクル全体のAPI開発を促進するAnypoint Platformの主要コンポーネントの概要を説明します。この内容では、プラットフォームがAPI主導の接続性をどのようにサポートしているかに焦点を当て、従来のAPI管理アプローチと比較し、スケーラブルで管理しやすいエンタープライズAPIの提供における優位性を強調します。

トピック 2	一般的な統合の問題、ユースケース、そして技術的ソリューションを認識する：このセクションでは、統合アーキテクトのスキルを評価し、統合シナリオの認識と適切なテクノロジーの選択に焦点を当てます。エンタープライズシステムの種類を区別し、従来の統合アプローチと最新の統合アプローチを比較します。受験者は、複雑なビジネス上の問題をコアとなるユースケースに分解し、それらをサポートする適切なテクノロジーを特定することが求められます。テクノロジーの分類とビジネスシナリオにおけるその適用に関する確かな理解に加え、統合ソリューションをシステムコンポーネントに分解する方法に関する知識も試されます。
トピック 3	システム統合におけるAnypoint Platformのコンポーネントと利点について説明します。このセクションでは、プラットフォームスペシャリストの知識ベースを対象とし、MuleSoftのAnypoint Platformがエンタープライズ統合をどのようにサポートするかを検証します。コアプラットフォームコンポーネントを特定し、システム接続におけるそれらの機能を理解する必要があります。受験者は、プロトコルベースとアプリケーションベースの両方の様々なAnypointコネクタを認識し、様々なホスティング環境におけるランタイムプレーンとコントロールプレーンの利点を説明する必要があります。また、統合およびDevOpsの専門家が使用する開発ツールと言語に焦点を当て、統合デリバリーを加速するAnypoint Exchange内の再利用可能なコンポーネントについても説明します。
トピック 4	統合開発における一般的な技術的複雑さとパターンを説明する：このセクションでは、プラットフォームスペシャリストの専門知識をテストし、統合開発に見られる様々な技術的パターンと複雑さを探求します。バッチ、ストリーム、マルチキャストなどのインタラクションパターン、およびオーケストレーションやコレオグラフィなどの統合構成スタイルの比較検討が含まれます。このセクションでは、設計重視のAPI開発、可観測性の実践、ログ管理の活用に重点を置いています。また、マイクロサービスとモノリシック展開の比較、ハイブリッドおよびクラウドインフラストラクチャ、APIゲートウェイとサービスメッシュの役割といったアーキテクチャの概念についても紹介します。
トピック 5	統合プロジェクトの役割、責任、ライフサイクルを理解する：このセクションでは、統合アーキテクトのスキルを評価し、MuleSoft 統合プロジェクトにおける基本的な責任について学習します。統合プロジェクトが失敗する理由を探り、IT デリバリーギャップとそれを埋めるための MuleSoft のフレームワークを紹介します。特に、プロデューサーとコンシューマーの両方をサポートする API 主導のデリバリーモデルの重要性を強調します。また、MuleSoft の製品中心のアプローチにおける一般的なデリバリー方法論、DevOps のベストプラクティス、ライフサイクルの各段階（設計、実装、管理）についても概説します。さらに、このようなプロジェクトに一般的に関与するチームメンバーの役割と責務を定義します。

Salesforce Certified MuleSoft Associate 認定 Salesforce-MuleSoft-Associate 試験問題 (Q39-Q44):

質問 #39

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'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?

- A. Developing a specification so consumers can test before the implementation is built
- B. Publishing the fully implemented API to Exchange so all developers can reuse the API
- C. Adding global policies to the API so all developers automatically secure the implementation before coding anything
- D. Building MUnit tests so administrators can confirm code coverage percentage during deployment

正解: A

解説:

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's a detailed explanation:

API Specification:

Definition: An API specification is a detailed, formal description of the API'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.

Design-First Approach: Design-First API Development

API Mocking: API Designer Mocking Service

質問 # 40

According to MuleSoft which deployment characteristic applies to a microservices application architecture?

- A. A deployment to enhance one capability requires a redeployment of all capabilities
- **B. Services exist as independent deployment artifacts and can be scaled independently of other services**
- C. All services of an application can be deployed together as single Java WAR file
- D. Core business capabilities are encapsulated in a single deployable application

正解: B

解説:

Microservices architecture is designed to enhance flexibility, scalability, and maintainability by decomposing applications into small, independent services. Here's a detailed explanation:

Independent Deployment:

Definition: Each microservice is developed, deployed, and managed independently. This allows teams to work on different services without interfering with each other.

Scalability: Services can be scaled independently based on demand, improving resource utilization and system resilience.

Benefits:

Flexibility: Enhances the ability to update or scale specific parts of an application without affecting the whole system.

Resilience: Isolates failures to individual services, preventing cascading failures across the entire application.

Technology Diversity: Allows the use of different technologies and languages best suited for each service.

Microservices Architecture: What are Microservices?

Benefits of Microservices: Microservices Characteristics

質問 # 41

Which Anypoint Platform component should a MuleSoft developer use to create an API specification prior to building the API implementation?

- A. API Manager
- B. MUnit
- **C. API Designer**
- D. Runtime Manager

正解: C

解説:

Creating an API specification before building the API implementation is a critical step in API development. MuleSoft's API Designer is the tool designed for this purpose. Here's a detailed explanation:

API Designer:

Purpose: API Designer is a web-based tool within Anypoint Platform that allows developers to design, document, and test APIs.

Features:

Specification Languages: Supports RAML and OAS (OpenAPI Specification) for defining APIs.

Interactive Editing: Provides a graphical and text-based interface to design API specifications interactively.

Mocking Service: Allows developers to create mock services to simulate API behavior before the actual implementation.

Process:

Define API: Use API Designer to create a detailed API specification, including endpoints, methods, request/response schemas, and security schemes.

Documentation: Automatically generate API documentation that can be shared with stakeholders.

Testing: Test the API design using the built-in mocking service to ensure it meets requirements.
MuleSoft Documentation: API Designer
API Design Best Practices: Designing APIs

質問 # 42

An integration team follows MuleSoft's recommended approach to full lifecycle API development. Which activity should this team perform during the API implementation phase?

- A. Validate the API specification
- **B. Use the API specification to build the MuleSoft application**
- C. Use the API specification to monitor the MuleSoft application
- D. Design the API specification

正解: B

解説:

MuleSoft recommends a full lifecycle API development approach which includes several phases such as design, implementation, testing, deployment, and management. During the API implementation phase, the primary activity is to use the API specification to build the MuleSoft application. Here's a detailed explanation:

API Design:

Create API Specification: Initially, an API specification is created using RAML or OAS (OpenAPI Specification) to define the API's structure, endpoints, request/response formats, and security requirements.

API Implementation:

Build Mule Application: Using the API specification as a blueprint, the development team implements the MuleSoft application. This involves creating flows, integrating with backend systems, and ensuring the API functions as specified.

APIKit: MuleSoft provides APIKit, a tool that automatically generates Mule flows based on the API specification, speeding up the development process.

Testing: During implementation, unit tests (using MUnit) and integration tests are created to ensure the API behaves as expected.

Validation and Monitoring:

Validate Against Specification: Throughout the implementation phase, the API is continuously validated against the original specification to ensure compliance.

Deployment and Monitoring: Post-implementation, the API is deployed, and tools like Anypoint Monitoring are used to monitor its performance and usage.

MuleSoft Documentation: Full Lifecycle API Management

APIKit: Building APIs with APIKit

質問 # 43

In which order are the API Client, API Implementation, and API Interface components called in a typical REST request?

- A. API Client > API Implementation > API Interface
- B. API Implementation > API Interface > API Client
- **C. API Client > API Interface > API Implementation**
- D. API Interface > API Client > API Implementation

正解: C

解説:

In a typical REST request, the components are called in a specific order to handle the client's request and provide the response.

Here'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.

REST API Design: RESTful Web Services

• • • • •

Salesforce-MuleSoft-Associate トレーニング サンプル: <https://www.xhs1991.com/Salesforce-MuleSoft-Associate.html>

- [illegible]