

Mule-101日本語版参考資料 & Mule-101最新知識



すべての顧客の誠実な要件を考慮して、Mule-101テスト問題は「品質第一とクライアント最高」の原則に沿って持続し、高品質の製品を豊富に備えた候補者に約束します。試験での99%の合格率、購入前の無料試用版など、Mule-101トレーニング資料の多数の利点がよく知られています。お客様の観点から、当社のMule-101テスト問題では、すべての候補者の要求が最優先事項となっています。最適なMule-101模擬テストに対するお客様の信頼とフィードバックを大切にしています。

Salesforce Mule-101 認定試験の出題範囲：

トピック	出題範囲
トピック 1	一般的な統合の問題、ユースケース、および技術的ソリューションを認識する:このドメインでは、統合シナリオを調査し、従来のアプローチと最新のアプローチを比較し、ビジネスの問題に適した統合テクノロジーの選択をガイドします。
トピック 2	API 管理のための Anypoint Platform のコンポーネントと利点について説明します。このドメインでは、Anypoint Platform の API 管理機能、ライフサイクル開発、および API 主導の接続の利点に焦点を当てます。
トピック 3	統合プロジェクトの役割、責任、ライフサイクルを特定します。このドメインでは、統合プロジェクトのライフサイクル、一般的な障害ポイント、MuleSoft の API 主導の配信モデル、DevOps プラクティス、統合プロジェクト内のチームの役割について説明します。
トピック 4	重要な統合の概念と用語を認識し、解釈する:このドメインでは、クラウドサービス モデル、インフラストラクチャの種類、ネットワークプロトコル、データ形式、セキュリティの原則、API 分類などの基本的な概念に焦点を当てます。
トピック 5	- 統合開発における一般的な技術的な複雑さとパターンについて説明します。このドメインでは、インタラクション パターン、構成パターン、API 仕様、可観測性アプローチ、およびデプロイメント - アプリケーション アーキテクチャの比較について説明します。

Mule-101最新知識 & Mule-101日本語認定対策

日常から離れて理想的な生活を求めるには、職場で高い得点を獲得し、試合に勝つために余分なスキルを習得する必要があります。同時に、社会的競争は現代の科学、技術、ビジネスの発展を刺激し、Mule-101試験に対する社会の認識に革命をもたらし、人々の生活の質に影響を与えます。Mule-101試験問題は、あなたの夢をかなえるのに役立ちます。さらに、Mule-101ガイドトレントに関する詳細情報を提供する当社のWebサイトにアクセスできます。

Salesforce Certified MuleSoft Integration Foundations 認定 Mule-101 試験問題 (Q33-Q38):

質問 # 33

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.

- A. Adopt a new approach that decouples core IT projects from the innovation that happens within each line of business
- B. Hire more IT developers, architects, and project managers to increase IT delivery
- C. Modify IT governance and security controls so that line of business developers can have direct access to the organization's systems of record
- D. Switch from a design-first to a code-first approach for IT development

正解: A

解説:

The IT Delivery Gap: This question addresses the "IT Delivery Gap"-the widening gap between business demands and IT's capacity to deliver.

The Solution (New Operating Model): MuleSoft recommends shifting from a centralized factory model to an Asset-Based consumption model.

Decoupling (Option A): By using API-led Connectivity:

Central IT builds reusable System APIs (unlocking core assets/systems of record) and governs them.

Lines of Business (LOB) consume these assets to build their own Process and Experience APIs (Innovation)2.

Why others fail:

(B) Hiring doesn't scale linearly with exponential demand.

(C) Removing governance creates security risks ("Shadow IT").

(D) Code-first ignores reuse and leads to technical debt.

質問 # 34

An organization is choosing between API-led connectivity and other integration approaches.

- A. Higher outcome repeatability through centralized development
- B. Improved security through adoption of monolithic architectures
- C. Increased developer productivity through self-service of API assets
- D. Greater project predictability through tight coupling of systems

正解: C

解説:

The Value Proposition: A primary goal of API-led connectivity is to close the IT delivery gap. It achieves this by turning APIs into reusable Assets published to Exchange. 8 Self-Service: When assets are discoverable, other developers (e.g., Line of Business developers) can reuse them without waiting for central IT to build everything from scratch. This "Self-Service" model significantly increases overall developer productivity.

Why others are incorrect:

Tight Coupling (B): API-led promotes loose coupling. Tight coupling makes systems brittle and hard to change (Point-to-Point).

Centralized Development (C): Creates a bottleneck. API-led enables federated development.

Monolithic (D): API-led breaks monoliths into composable services (Microservices/APIs).

質問 # 35

A DevOps team has adequate observability of individual system behavior and performance, but it struggles to track the entire lifecycle of each request across different microservices. Which additional observability approach should this team consider adopting?

- **A. Tracing**
- B. Analytics
- C. Data mining
- D. Metrics

正解: A

解説:

The Challenge: In a microservices architecture, a single user request might traverse dozens of different services. If an error occurs or latency is high, looking at the logs of just one service isn't enough.

Distributed Tracing: This is the specific technology used to track a request as it hops between services.

How it works: It assigns a unique Trace ID (Correlation ID) to the request at the entry point. This ID is passed to every downstream service. Tracing tools (like Anypoint Monitoring's Telemetry or Jaeger) verify the full path, showing exactly how long the request spent in each hop.

質問 # 36

Which productivity advantage does Anypoint Platform have to both implement and manage an API?

- **A. Automatic API proxy generation**
- B. Automatic API semantic versioning
- C. Automatic API specification generation
- D. Automatic API governance

正解: A

解説:

Automatic API Proxy Generation: When managing an API in API Manager, Anypoint Platform allows you to automatically generate and deploy an API Proxy application to CloudHub.

Functionality: This proxy sits in front of your backend implementation (or a non-Mule API) and enforces policies (like rate limiting or security) without requiring you to write code for the proxy manually. This significantly speeds up the process of securing and managing APIs compared to building custom gateway solutions.

質問 # 37

According to MuleSoft, which system integration term describes the method, format, and protocol used for communication between two systems? 4

- A. Interaction
- B. Message
- C. Component
- **D. Interface**

正解: D

解説:

Comprehensive and Detailed Explanation:

The Interface: In system integration and MuleSoft terminology, the Interface is the contract that defines how two systems communicate. It specifies:

Protocol: How data is transmitted (e.g., HTTP, FTP, AMQP).

Format: The structure of the data (e.g., JSON, XML, CSV).

Method: The specific action (e.g., GET, POST).

Why other options are incorrect:

Message: This refers to the actual data payload (the content) being sent, not the rules of communication.

Component: This usually refers to a specific building block within the Mule flow (like a Logger or a Database connector) or a software module, not the communication definition itself.

Interaction: This describes the act of communicating, not the definition of the standard used.

• • • • •

Mule-101最新知識: <https://www.xhs1991.com/Mule-101.html>

- [illegible]