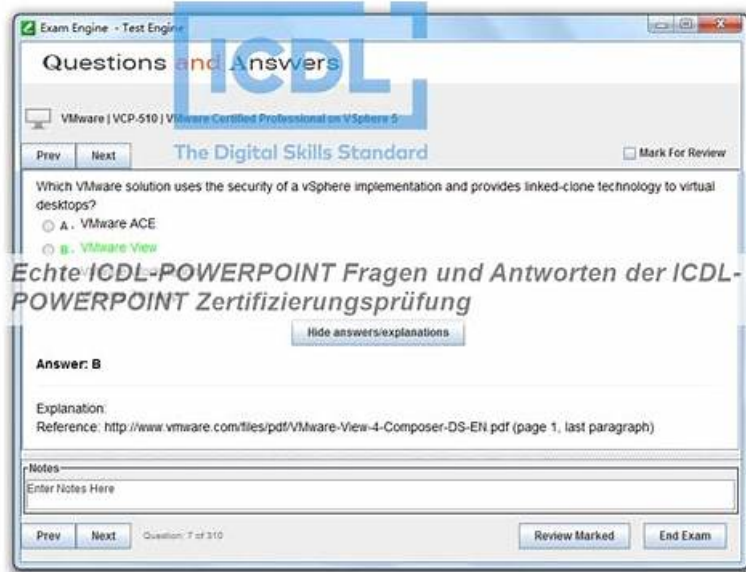


# Echte Mule-101 Fragen und Antworten der Mule-101 Zertifizierungsprüfung



Übrigens, Sie können die vollständige Version der ZertFragen Mule-101 Prüfungsfragen aus dem Cloud-Speicher herunterladen: <https://drive.google.com/open?id=1r9Ebr0vMCddbCAJiLEv7shqaPPE3xAK>

Mit den Schulungsunterlagen zur Salesforce Mule-101 Zertifizierungsprüfung von ZertFragen können Sie die neuesten Fragen und Antworten zur Salesforce Mule-101 Zertifizierungsprüfung bekommen und somit die Salesforce Mule-101 Zertifizierungsprüfung erfolgreich einmalig bestehen. Die Salesforce Mule-101 Zertifizierungsprüfung ist nützlich für Ihre Berufskarriere. Die Schulungsunterlagen zur Salesforce Mule-101 Zertifizierungsprüfung von ZertFragen garantieren, dass Sie die Fragen sowie deren Konzept verstehen können.

Wir versprechen, dass Sie die Prüfung zum ersten Mal mit unseren Schulungsunterlagen zur Salesforce Mule-101 Zertifizierungsprüfung bestehen können. Sonst erstatten wir Ihnen die gesamte Summe zurück.

>> **Mule-101 Deutsch Prüfungsfragen** <<

## Mule-101 Übungstest: Salesforce Certified MuleSoft Integration Foundations & Mule-101 Braindumps Prüfung

ZertFragen ist eine Website, die Ihnen zum Erfolg führt. ZertFragen bietet Ihnen die ausführlichen Schulungsmaterialien zur Salesforce Mule-101 (Salesforce Certified MuleSoft Integration Foundations) Zertifizierungsprüfung, mit deren Hilfe Sie in kurzer Zeit das relevante Wissen zur Prüfung auswendiglernen und die Prüfung einmalig bestehen können.

### Salesforce Mule-101 Prüfungsplan:

Thema	Einzelheiten
Thema 1	<ul style="list-style-type: none"> <li>• Explain the common technical complexities and patterns in integration development: This domain explores interaction patterns, composition patterns, API specifications, observability approaches, and deployment</li> <li>• application architecture comparisons.</li> </ul>
Thema 2	<ul style="list-style-type: none"> <li>• Recognize and interpret essential integration concepts and terminology: This domain focuses on foundational concepts including cloud service models, infrastructure types, networking protocols, data formats, security principles, and API classifications.</li> </ul>

Thema 3	<ul style="list-style-type: none"> <li>Identify the roles, responsibilities, and lifecycle of a integration project: This domain covers integration project lifecycles, common failure points, MuleSoft's API-led delivery model, DevOps practices, and team roles within integration projects.</li> </ul>
---------	--

## Salesforce Certified MuleSoft Integration Foundations Mule-101 Prüfungsfragen mit Lösungen (Q34-Q39):

### 34. Frage

A platform architect includes both an API gateway and a service mesh in the architecture of a distributed application for communication management.

- A. Between application services and the firewall
- B. Between the application and external API clients
- C. Between services within the application
- D. Between the application and external API implementations

**Antwort: C**

Begründung:

Service Mesh vs. API Gateway:

API Gateway: Typically manages North-South traffic (traffic entering the application network from external clients).

Service Mesh: Is designed to manage East-West traffic (traffic flowing between microservices within the application network or cluster).

Anypoint Service Mesh: It manages, secures, and observes communication between services within the application (microservices), ensuring zero-trust security and policy enforcement inside the Kubernetes cluster.

### 35. Frage

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

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

**Antwort: A**

Begründung:

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.

### 36. Frage

A MuleSoft developer must implement an API as a Mule application, run the application locally, and execute unit tests against the running application.

- A. API Designer
- B. Anypoint Studio

- C. Anypoint CLI
- D. API Manager

**Antwort: B**

Begründung:

Anypoint Studio: This is the desktop Integrated Development Environment (IDE) for MuleSoft.

Capabilities:

Implement: It provides the graphical interface to drag-and-drop connectors and configure flows.

Run Locally: It includes an embedded Mule Runtime engine, allowing developers to run and debug apps on their own machines.

Execute Unit Tests: It has MUnit fully integrated, allowing developers to run tests and see coverage reports directly in the IDE.

Why others are incorrect:

API Designer: Web-based tool for designing specs (RAML), not implementing logic or running local runtimes.

Anypoint CLI: Command-line tool for platform operations, not development/testing.

### 37. Frage

According to MuleSoft's API development best practices, which type of API development approach starts with writing and approving an API contract?

- A. Agile
- **B. Design-first**
- C. Catalyst
- D. Implement-first

**Antwort: B**

Begründung:

Design-First: This approach dictates that the API Contract (the Specification, e.g., RAML/OAS) must be written, reviewed, and approved before any implementation code is written<sup>1</sup>.

The Contract: The "Contract" serves as the agreement between the API provider and the consumer.

Why others are incorrect:

Implement-first: You write the code (Mule flows) first, and the contract is generated from the code (or ignored).

Catalyst: Is a broader delivery methodology, not specifically the "Contract-first" technical approach.

### 38. Frage

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.

- **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

**Antwort: A**

Begründung:

Design-First Approach: This methodology prioritizes creating the API contract (RAML/OAS) before writing any code<sup>19</sup>.

The Benefit: By defining the specification first and publishing it to Exchange (often with a Mocking Service), API consumers (frontend developers or other teams) can test and provide feedback on the design immediately. <sup>20</sup>This ensures the API meets business needs before the expensive work of backend implementation begins, preventing the failure described in the scenario<sup>21,22,23</sup> Why others are incorrect:

Publishing fully implemented API (C): This is a "Code-First" approach (build first, share later).

Global Policies (B): Relates to governance, not the design methodology.

### 39. Frage

.....

