

Latest Integration-Architect Exam Fee | Valid Integration-Architect Test Book



2026 Latest PrepAwayTest Integration-Architect PDF Dumps and Integration-Architect Exam Engine Free Share:
<https://drive.google.com/open?id=1VKRKe9nlUkgKwlaLX6Yhc4hX0mjYzhKY>

We have security and safety guarantee, which mean that you cannot be afraid of virus intrusion and information leakage since we have data protection acts, even though you end up studying Integration-Architect test guide of our company, we will absolutely delete your personal information and never against ethic code to sell your message to the third parties. Our Integration-Architect Exam Questions will spare no effort to perfect after-sales services. Thirdly countless demonstration and customer feedback suggest that our Salesforce Certified Integration Architect study question can help them get the certification as soon as possible, thus becoming the elite, getting a promotion and a raise and so forth.

The Salesforce Certified Integration Architect certification exam covers a wide range of topics related to Salesforce integration, including data integration, application integration, and identity integration. Integration-Architect exam also covers integration security, governance, and design patterns. Candidates are tested on their ability to design and implement integrations that meet business requirements while adhering to industry best practices and security standards.

To become a Salesforce Certified Integration Architect, you must pass the Integration-Architect Exam. Integration-Architect exam measures your ability to design and implement complex integration solutions using various tools and technologies. Integration-Architect exam consists of multiple-choice questions, and you must score at least 65% to pass the exam. Integration-Architect exam is designed to test your knowledge and skills in various areas, including integration architecture, data integration, security, and performance.

>> Latest Integration-Architect Exam Fee <<

Latest Integration-Architect Exam Fee Free PDF | Valid Valid Integration-Architect Test Book: Salesforce Certified Integration Architect

Our Integration-Architect test braindumps are carefully developed by experts in various fields, and the quality is trustworthy. What's more, after you purchase our products, we will update our Integration-Architect exam questions according to the new changes and then send them to you in time to ensure the comprehensiveness of learning materials. We also have data to prove that 99% of those who use our Integration-Architect Latest Exam torrent to prepare for the exam can successfully pass the exam and get Integration-Architect certification. As long as you decide to choose our Integration-Architect exam questions, you will have an opportunity to prove your abilities, so you can own more opportunities to embrace a better life.

Salesforce Integration-Architect (Salesforce Certified Integration Architect) Exam is designed to test the knowledge and skills required to design and implement effective and efficient integrations between different systems and applications using the Salesforce platform. Salesforce Certified Integration Architect certification is aimed at professionals who have expertise in integrating systems with Salesforce and have a deep understanding of Salesforce's integration capabilities and tools.

Salesforce Certified Integration Architect Sample Questions (Q28-Q33):

NEW QUESTION # 28

A company's cloud-based single page application consolidates data local to the application with data from on premise and 3rd party systems. The diagram below typifies the application's combined use of synchronous and asynchronous calls.

The company wants to use the average response time of its application's user interface as a basis for certain alerts. For this purpose, the following occurs:

1. Log every call's start and finish date and time to a central analytics data store.
2. Compute response time uniformly as the difference between the start and finish date and time - A to H in the diagram.

Which computation represents the end-to-end response time from the user's perspective?

- A. Sum of A and H
- B. Sum of A, G, and H
- C. Sum of A to H
- D. Sum of A to F

Answer: A

NEW QUESTION # 29

Northern Trail Outfitters needs a synchronous callout from Salesforce to an Order Management System (OMS) when an opportunity is "Closed/Won" with products attached. What should an integration architect do to satisfy these requirements?

- A. Write a trigger that invokes an Apex proxy class to make a REST callout to the OMS.
- B. Develop a batch Apex job that aggregates closed opportunities and makes a REST callout to the OMS hourly.
- C. Build a Lightning component that makes a synchronous Apex REST callout to the OMS when a button is clicked.

Answer: C

Explanation:

To satisfy a requirement for a synchronous callout triggered by a user action, the architect should use a UI-driven approach, such as a Lightning component and a button.

In Salesforce, triggers (Option B) are primarily used for asynchronous logic in integration contexts. Because a trigger executes as part of the database save operation, making a synchronous callout directly from a trigger is prohibited as it would block the database transaction until the external system responds, leading to performance degradation and "uncommitted work pending" errors. If a trigger must initiate an integration, it must do so asynchronously (using @future or Queueable Apex), which violates the requirement for a synchronous call.

By using a Lightning component, the architect can initiate a synchronous Request-and-Reply pattern. When the sales rep clicks the "Submit to OMS" button, the component invokes an Apex method that makes the REST callout to the OMS in real-time. The user remains on the page while the system waits for the OMS to respond, allowing for immediate feedback-such as an order confirmation number or an error message-to be displayed in the UI. A Batch Apex job (Option C) is inherently asynchronous and delayed, making it unsuitable for a synchronous, real-time fulfillment requirement.

NEW QUESTION # 30

A new Salesforce program has the following high-level abstract requirement: Business processes executed on Salesforce require data updates between some internal systems and Salesforce. Which relevant details should a Salesforce integration architect seek to specifically solve for the integration architecture needs of the program?

- A. Core functional and non-functional requirements for User Experience design, Encryption needs, Community, and License choices
- B. Source and Target system, Directionality, and data volume & transformation complexity, along with any middleware that can be leveraged
- C. Integration skills, SME availability, and Program Governance details

Answer: B

Explanation:

To translate abstract business needs into a functional Integration Architecture, an architect must move beyond "what" the business wants to "how" the data will technically flow. The details in Option B represent the fundamental building blocks of any integration design.

* Source and Target Systems: Determining which systems are involved dictates the available APIs (REST, SOAP, Bulk) and required security protocols (OAuth, Mutual SSL).

* Directionality: Knowing if the update is unidirectional or bidirectional is critical for record mastering, identifying the "System of Record," and preventing infinite data loops.

* Data Volume: This is a primary driver for selecting the Integration Pattern. High volumes (millions of records) require the Bulk API, while low-volume, real-time updates are better suited for Streaming or REST APIs.

* Transformation Complexity: If data must be heavily massaged or merged from multiple sources, it justifies the need for Middleware (ESB/iPaaS).

Option A focuses on user interface and licensing, which are implementation details rather than architectural integration requirements. Option C deals with project management and resource allocation. While these are important for the project's success, they do not help the architect decide between a Request-Reply or Fire-and-Forget pattern. By focusing on systems, data direction, and volume, the architect ensures that the proposed solution is technically viable, scalable, and adheres to Salesforce platform governor limits.

NEW QUESTION # 31

A customer imports data from an external system into Salesforce using Bulk API. These jobs have batch sizes of 2000 and are run in parallel mode. The batch fails frequently with the error "Max CPU time exceeded". A smaller batch size will fix this error.

Which two options should be considered when using a smaller batch size?

Choose 2 answers

- A. Smaller batch size may cause record-locking errors.
- B. Smaller batch size may exceed the concurrent API request limits.
- C. Smaller batch size may increase time required to execute bulk jobs.
- D. Smaller batch size can trigger "Too many concurrent batches" error.

Answer: A,C

NEW QUESTION # 32

A new Salesforce program requires data updates between internal systems and Salesforce. Which relevant detail should an integration architect seek to solve for integration architecture needs?

- A. Timing aspects, real-time/near real-time (synchronous or asynchronous), batch and update frequency
- B. Integration skills, SME availability, and Program Governance details
- C. Core functional and non-functional requirements for User Experience design, Encryption needs, Community and license choices

Answer: A

Explanation:

In the "Discovery" phase of integration architecture, the architect must translate abstract business needs into technical requirements. The most critical variables that define the Integration Pattern are Timing and Volume.

An architect cannot choose between the REST API, Streaming API, Bulk API, or Outbound Messaging without knowing:

* Latency Requirements: Does the business need the update in 200 milliseconds (Synchronous), 2 minutes (Near Real-Time), or 24 hours (Batch)?

* Frequency: Is the data updated every time a user clicks a button, or once at the end of the day?

* Volume: Are we moving 10 records at a time or 10 million?

Option A focuses on UI/UX and licensing, which are project management concerns. Option B focuses on resource allocation and governance. While important for the project, they do not inform the technical design of the data flow.

By specifically seeking out Timing aspects (Synchronous vs. Asynchronous) and Update Frequency, the architect can apply the Salesforce Integration Decision Matrix. For instance, a "Real-time" requirement for small volumes leads to a Request-Reply pattern via Apex Callouts. A "Nightly" requirement for large volumes leads to a Batch Data Synchronization pattern via the Bulk API.

Identifying these "Non-Functional Requirements" (NFRs) early is the only way to ensure the architecture is scalable and stays within platform governor limits.

