Salesforce Integration Architect Practice Test (Sample)

Study Guide

BY EXAMZIFY

Everything you need from our exam experts!

Questions

1. **Which integration process types are time-based?**
   - **A. Synchronous, Asynchronous, Batch**
   - **B. Real-time, Delayed, Synchronous**
   - **C. Synchronous, Immediate, Bulk**
   - **D. Batch, Restful, Dynamic**

2. **What is the maximum number of callouts that can be made from a single Apex transaction?**
   - **A. 50 callouts**
   - **B. 75 callouts**
   - **C. 100 callouts**
   - **D. 150 callouts**

3. **What is a benefit of using the Force.com Bulk API?**
   - **A. It guarantees real-time data updates**
   - **B. It improves stability and monitoring of data loads**
   - **C. It requires manual initiation for data processing**
   - **D. It is only used for small data sets**

4. **What is a limitation of using Apex web service methods compared to Salesforce APIs?**
   - **A. They cannot be configured**
   - **B. They can encapsulate too much logic**
   - **C. They do not support complex queries**
   - **D. They perform slower for bulk data operations**

5. **Which URI should be used for the Batching REST resource?**
   - **A. /services/data/v37.0/composite/batch**
   - **B. /services/data/v37.0/batch**
   - **C. /services/data/v37.0/composite/allbatch**
   - **D. /services/data/v37.0/composite/updatebatch**

6. Which annotation is used to expose an Apex class as a REST web service?

A. @HttpGet

B. @RestResource

C. @HttpPost

D. @ServiceEndpoint

7. In an Apex REST class, what keyword is used to indicate that the class is globally accessible?

A. public

B. private

C. internal

D. global

8. What solutions are appropriate for displaying billing history through a service?

A. Visualforce page with a Batch Callout

B. Custom Apex Trigger

C. Visualforce page initiating an Apex HTTP callout

D. Process Builder with Data Loader

9. What status code signifies an "Internal Server Error"?

A. 200

B. 404

C. 500

D. 403

10. What should you do if you want to mock a callout in a test?

A. Ignore the callout in the test

B. Use real API calls to validate

C. Set up appropriate callout mocks

D. Implement the callout in the test

Answers

1. A
2. C
3. B
4. D
5. A
6. B
7. D
8. C
9. C
10. C

Explanations

1. Which integration process types are time-based?

A. Synchronous, Asynchronous, Batch

B. Real-time, Delayed, Synchronous

C. Synchronous, Immediate, Bulk

D. Batch, Restful, Dynamic

The integration process types that are considered time-based include both asynchronous and batch processes. Asynchronous processes allow for operations that do not require an immediate response; they can execute at a later time, making them time-based in nature. This is particularly useful for scenarios where immediate user interaction is not required, allowing for better resource management and reduced wait times. Batch processing is also inherently time-based because it involves the collection and processing of data in groups over a specified period. This can be beneficial for handling large volumes of data or executing operations during off-peak hours, thus optimizing performance and resource allocation. While synchronous processes do involve immediate execution and responses, they do not align as time-based since they require real-time interaction rather than scheduled or delayed processing. Therefore, it is primarily the asynchronous and batch methods that fit the time-based criteria, validating the inclusion of these processes in the integration types listed in the correct answer.

2. What is the maximum number of callouts that can be made from a single Apex transaction?

A. 50 callouts

B. 75 callouts

C. 100 callouts

D. 150 callouts

The limit for the maximum number of callouts that can be made from a single Apex transaction is 100. This limit is imposed to ensure efficient resource usage and maintain system performance, as each callout to an external service consumes resources and time. The ability to make up to 100 callouts per transaction allows developers to design robust integrations with external services while still adhering to Salesforce's governor limits, which are in place to prevent a single process from monopolizing shared resources. Having a clear understanding of this limit is crucial for developers when designing solutions that integrate Salesforce with other applications or services. Exceeding the callout limit would result in runtime exceptions, which could disrupt business processes. Therefore, strategies such as batching requests or optimizing integration points may be necessary to stay within this limit.

3. What is a benefit of using the Force.com Bulk API?

A. It guarantees real-time data updates

B. It improves stability and monitoring of data loads

C. It requires manual initiation for data processing

D. It is only used for small data sets

Using the Force.com Bulk API offers significant advantages in terms of improving the stability and monitoring of data loads. This API is designed specifically for handling large volumes of data efficiently, allowing developers to submit batches of records for processing. By employing a batch processing approach, the Bulk API can effectively manage data transactions without placing excessive load on the system, which can result in improved stability during high-volume operations. Additionally, the Bulk API provides features for monitoring the status of jobs and batches, allowing users to track the progress of their data uploads and handle failures more effectively. This monitoring capability is critical in ensuring that data loads are completed successfully and provides necessary feedback if any issues arise. In contrast, the other options do not align with the benefits of the Bulk API. It does not guarantee real-time updates, which are generally better suited for the REST or SOAP APIs. The API supports automated processes rather than requiring manual initiation for data processing, and it is explicitly designed to handle large data sets, not just small ones. Thus, option B accurately captures a key benefit of using the Force.com Bulk API.

4. What is a limitation of using Apex web service methods compared to Salesforce APIs?

A. They cannot be configured

B. They can encapsulate too much logic

C. They do not support complex queries

D. They perform slower for bulk data operations

The correct choice highlights a notable aspect of Apex web service methods regarding their performance for bulk data operations. Apex web service methods are indeed designed to handle specific transactions or requests, often operating synchronously and catering to individual or transactional data requests. When it comes to handling large volumes of data, eight distinct challenges arise, primarily centered around the performance and scalability of the Salesforce platform, which is highly optimized for batch processing through Salesforce APIs. Salesforce APIs, particularly the Bulk API, are explicitly designed to process large datasets efficiently. They can handle multiple records in a single request, making them significantly faster and better suited for bulk operations. In contrast, Apex web service methods typically require multiple calls for a large dataset, resulting in increased time and resource consumption. Using Apex web service methods for bulk data can lead to throttling issues and limits on governor limits, which can hinder performance and increase complexity. Consequently, they are less ideal for scenarios requiring bulk data processing, underscoring the statement’s correctness regarding performance limitations.   Understanding this distinction helps architects and developers choose the appropriate integration approach based on the scale and nature of the data operations they need to perform within the Salesforce ecosystem.

5. Which URI should be used for the Batching REST resource?

A. /services/data/v37.0/composite/batch

B. /services/data/v37.0/batch

C. /services/data/v37.0/composite/allbatch

D. /services/data/v37.0/composite/updatebatch

The Batching REST resource in Salesforce is accessed via the URI that includes the term "composite/batch." This structure is designed to handle multiple requests in a single HTTP call, which is essential for efficient data operations. The "composite" endpoint indicates that this resource can handle complex requests that may involve multiple entity types, and the "batch" segment specifies that this particular URI is used for executing batches of requests. Using /services/data/v37.0/composite/batch allows you to send multiple API requests wrapped in a single call, which helps to reduce the number of network calls needed when interacting with Salesforce and can lead to better performance and lower latency when dealing with high volumes of data. The other options do not follow this correct structure or are not designated for batching. For instance, the URI that ends with "/batch" doesn't specify it as a composite operation, which is critical for batching functionality. Thus, the choice /services/data/v37.0/composite/batch is the accurate URI for accessing the Batching REST resource.

6. Which annotation is used to expose an Apex class as a REST web service?

A. @HttpGet

B. @RestResource

C. @HttpPost

D. @ServiceEndpoint

The annotation that is used to expose an Apex class as a REST web service is @RestResource. This annotation allows developers to define a class as a RESTful web service, enabling the class to handle HTTP requests and deliver responses in various formats such as JSON or XML. By utilizing @RestResource, the methods within the class can be mapped to specific HTTP methods such as GET, POST, PATCH, or DELETE, and can be accessed via a designated URL. The choice to use @RestResource is essential for creating REST APIs in Salesforce, as it acts as the entry point for RESTful interactions with the Apex class. This facilitates the integration of Salesforce with other systems, allowing for seamless data access and manipulation. The other options, while related to handling HTTP requests, do not serve the specific purpose of exposing an Apex class as a RESTful service. For instance, @HttpGet and @HttpPost are annotations that indicate methods that respond to GET and POST requests, respectively, but they need to be used within a class that is annotated with @RestResource to function correctly in a REST API context. The @ServiceEndpoint annotation is more relevant to SOAP web services rather than RESTful services.

7. In an Apex REST class, what keyword is used to indicate that the class is globally accessible?

A. public

B. private

C. internal

D. global

In Apex, the keyword used to indicate that a class is globally accessible is "global." This means that the class can be accessed by all other Apex code, regardless of whether it is within the same namespace or a different one. This level of accessibility is particularly important when developing managed packages, as it allows external systems and applications to interact with your Apex code seamlessly. Using the "global" modifier is essential for classes that are intended to be part of a public API, as it ensures that users in other organizations or namespaces can utilize the functionality provided by the class. In the context of Apex REST classes, making them global enables external systems to invoke the REST services exposed by these classes, facilitating integration scenarios. In contrast, other access modifiers such as "public," "private," and "internal" offer different levels of accessibility, but they do not enable inter-namespace accessibility to the same extent as "global." Public classes can be accessed from within the same namespace and would work for shared functionality in the same organization but would not be accessible from other namespaces or organizations.

8. What solutions are appropriate for displaying billing history through a service?

A. Visualforce page with a Batch Callout

B. Custom Apex Trigger

C. Visualforce page initiating an Apex HTTP callout

D. Process Builder with Data Loader

Displaying billing history through a service requires a reliable method for retrieving and presenting data, particularly when dealing with external systems. The correct choice involves using a Visualforce page initiating an Apex HTTP callout.   This solution allows a Visualforce page to dynamically fetch billing information from an external service using an HTTP callout. Apex can execute this callout to retrieve the necessary data, which can then be rendered on the Visualforce page for users to view in real time. This method is effective for integrating external billing systems with Salesforce, ensuring that users have access to the most up-to-date billing history directly within the Salesforce interface. Also, when considering other options, using a Visualforce page with a Batch Callout wouldn't be appropriate because batch processing is generally used for processing large volumes of records rather than for real-time data interactions, which are essential for displaying billing history. A Custom Apex Trigger is designed to run in response to changes in Salesforce records rather than for making external service calls, limiting its use for this scenario. Lastly, using Process Builder with Data Loader wouldn't be suitable for displaying real-time information since Process Builder is primarily focused on automation within Salesforce, and Data Loader is a tool for batch uploading data, not for displaying or retrieving it on demand.   Thus

9. What status code signifies an "Internal Server Error"?

A. 200

B. 404

C. 500

D. 403

The status code that signifies an "Internal Server Error" is 500. This code indicates that the server encountered an unexpected condition that prevented it from fulfilling the request. Essentially, it's a generic error message indicating a problem on the server side that isn't specified by other specific error codes. In web communication, this status code plays a critical role in helping developers and users understand that while the request was valid, something went wrong within the server's processing. This is particularly important for troubleshooting and resolving issues related to server operations, as it often indicates that there might be a bug in the server-side logic or a misconfiguration that needs to be addressed. In contrast, the other status codes represent different outcomes: 200 signifies a successful request; 404 indicates that a resource was not found; and 403 means that access to the requested resource is forbidden. These codes serve distinct purposes within the HTTP protocol, helping to convey the outcome of a web request beyond just successful or unsuccessful execution.

10. What should you do if you want to mock a callout in a test?

A. Ignore the callout in the test

B. Use real API calls to validate

C. Set up appropriate callout mocks

D. Implement the callout in the test

To effectively mock a callout in a test, setting up appropriate callout mocks is essential. This practice allows you to simulate the behavior of an external service without making actual HTTP requests during testing. By creating these mocks, you can define the expected responses and the specific scenarios you want to test, ensuring your code behaves correctly under various conditions without relying on the availability or state of the external service. Utilizing callout mocks also leads to faster test execution and greater reliability, as tests won't fail due to issues with the external API or network connectivity. This approach aligns with best practices in software development, where unit tests should be isolated, fast, and predictable. This way, you can focus on validating the logic of your code rather than the intricacies of external systems.