Plat-Arch-204認定資格試験問題集、Plat-Arch-204復習教材

我々はPlat-Arch-204問題集の英語版と日本語版を開発しています。英語版と日本語版の内容が同じですが、言葉だけ違います。Plat-Arch-204問題集に英語試験と日本語試験を準備する受験者たちは気楽に試験に合格することができます。それに、我々のSalesforceのPlat-Arch-204日本語版問題集を購入するなら、英語版をおまけにさし上げます。

Salesforce Plat-Arch-204 認定試験の出題範囲：

トピック	出題範囲
トピック 1	• Design Integration Solutions: This domain centers on selecting integration patterns, designing complete solutions with appropriate components, understanding trade-offs and limitations, choosing correct Salesforce APIs, and determining required standards and security mechanisms.
トピック 2	• Evaluate Business Needs: This domain addresses gathering functional and non-functional requirements, classifying data by sensitivity, identifying CRM success factors, and understanding how business growth and regulations impact integration choices.
トピック 3	• Translate Needs to Integration Requirements: This domain involves converting business needs into technical specifications by documenting systems and patterns, evaluating constraints, defining security requirements, and determining performance needs like volumes, response times, and latency.
トピック 4	• Evaluate the Current System Landscape: This domain covers analyzing existing technical environments to understand current systems, their standards, protocols, limitations, and boundaries, while identifying constraints and authentication • authorization requirements.

Plat-Arch-204復習教材、Plat-Arch-204日本語認定対策

当社Salesforceは、Plat-Arch-204試験問題を編集するために、この分野で多くの主要な専門家を採用しています。当社のチームベースの作業システムは、次世代の最高のPlat-Arch-204試験トレントがIt-Passports最終的に形を成し遂げる精神と手を携える人材を最大限に引き出すように設計されています。 当社は、優れたアフターサービスを提供し、ガイドの急流に革新をもたらした実績があります。 当社の専門家がお客様に世界クラスのPlat-Arch-204のSalesforce Certified Platform Integration Architectガイドトレントを作成できるため、お客様の成功が保証されます。 Plat-Arch-204試験に合格する必要があります。

Salesforce Certified Platform Integration Architect 認定 Plat-Arch-204 試験問題 (Q41-Q46):

質問 # 41
Northern Trail Outfitters (NTO) wants to improve the quality of callouts from Salesforce to its REST APIs. For this purpose, NTO will require all API Clients/consumers to adhere to REST API Markup Language (RAML) specifications that include the field-level definition of every API request and response Payload. The RAML specs serve as interface contracts that Apex REST API Clients can rely on. Which design specification should the integration architect include in the integration architecture to ensure that Apex REST API Clients' unit tests confirm Adherence to the RAML specs?

- A. Require the Apex REST API Clients to implement the HttpCalloutMock
- B. Call the HttpCalloutMock Implementation from the Apex REST API Clients.
- C. Implement HttpCalloutMock to return responses per RAML specification.

正解：C

解説：
In a contract-first integration strategy using RAML (RESTful API Modeling Language), the specification defines the exact structure of requests and responses. Because Salesforce unit tests cannot perform actual network callouts, the platform requires d1evelopers to use the HttpCalloutMock interface to simulate responses.
To ensure that the integration code strictly adheres to the established RAML contract, the integration architect must mandate that the HttpCalloutMock implementation returns responses that mirror the RAML specification. This means the mock must include all required fields, correct data types, and the expected HTTP status codes (e.g., 200 OK, 201 Created) as defined in the contract. By doing this, the unit tests verify that the Apex client code can successfully parse and process the specific JSON or XML payloads defined in the RAML spec.
Option A and B are technically imprecise. The Apex client does not "implement" the mock; rather, the test class provides a separate mock implementation to the runtime via Test.setMock(). The value of the integration architecture lies in the content of that mock. If the mock is designed to return contract-compliant data, then any change to the RAML that breaks the Apex code's ability to process it will be caught immediately during the testing phase. This "Mock-as-a-Contract" approach provides a safety net, ensuring that Salesforce remains compatible with external services even as those services evolve, provided the RAML is kept up to date.

質問 # 42
Northern Trail Outfitters (NTO) is planning to create a native employee-facing mobile app with the look and feel of Salesforce Lighting Experience. The mobile app needs to integrate with NTO's Salesforce org. Which Salesforce API should be used to implement this integration?

- A. REST API
- B. User Interface API
- C. Connect REST API

正解：B

解説：
When building custom mobile or web applications that aim to replicate the look and feel of Salesforce Lightning Experience, the User Interface (UI) API is the architecturally recommended choice.
The UI API is specifically designed to provide the metadata and data needed to build high-fidelity user interfaces. Unlike the standard REST API (Option B), which returns raw record data, the UI API returns both data and metadata in a single response.

This includes information about page layouts, field-level security, picklist values, and localized labels. By using the UI API, the mobile app can dynamically render fields according to the user's permissions and the organization's layout configurations, ensuring that the custom app stays in sync with changes made in Salesforce Setup without requiring code updates in the mobile app.
Connect REST API (Option A) is primarily used for Chatter, Communities (Experience Cloud), and CMS content, and while it is useful for those specific social features, it does not provide the layout and record-level metadata required for a full CRM interface.
The UI API is the same underlying technology that powers the Salesforce mobile app and Lightning Experience itself. Therefore, utilizing this API allows NTO's developers to build a native app that perfectly mimics the Lightning Experience while reducing the amount of custom logic needed to handle complex Salesforce UI requirements.

質問 # 43
Service agents at Northern Trail Outfitters use Salesforce to manage cases and B2C Commerce for ordering.22 Which integration solution should an architect recommend in order for the service agents to see order history from a business-to-consumer (B2C) Commerce system?

- A. Salesforce B2C Commerce to Service Cloud Connector
- B. REST API offered by Commerce Platform
- C. MuleSoft Anypoint Platform

正解：A

解説：
For organizations using both Salesforce Service Cloud and B2C Commerce (formerly Demandware), Salesforce provides a specialized, pre-built integrat25ion known as the Salesforce B2C Commerce to Service Cloud Connector. This connector is part of the Salesforce B2C Solution Architecture and is the recommended choice because it offers "out-of-the-box" cross-cloud functionality.
The connector enables several critical business processes for service agents:
Unified Customer Profile: Synchronizes customer data between the two platforms, ensuring agents have the most current contact information.
Order History Visibility: Allows agents to view real-time order data and status from the Commerce system directly within the Service Console.
Service Actions: Enables agents to perform commerce-related tasks, such as "Order on Behalf Of," without leaving the Salesforce interface.
While an architect could build a custom integration using the Commerce REST API (Option A) or MuleSoft (Option C), these approaches require significant development, testing, and long-term maintenance effort. The B2C Connector reduces time-to-market and leverages Salesforce's own engineering to handle complex synchronization logic and API versioning. Recommending the standard connector aligns with the architectural principle of "clicks before code" and ensures that the integration remains supported by Salesforce as both platforms evolve.

質問 # 44
Salesforce is considered to be the system of record for the customer. UC plans on using middleware to integrate Salesforce with external systems (ERP, ticketing, data lake). UC has a requirement to update the proper external system with record changes in Salesforce and vice versa. Which solution should an integration architect recommend?

- A. Locally cache external Ids at the middleware layer and design business logic to map updates between systems.
- B. Store unique identifiers in an External ID field in Salesforce and use this to update the proper records across systems.
- C. Use Change Data Capture to update downstream systems accordingly when a record changes.

正解：B

解説：
In a multi-system landscape, maintaining data synchronization requires a robust Identity Mapping strategy. The standard Salesforce architectural recommendation is to use External ID fields to store the unique identifiers from each secondary system.
By storing the ERP ID, Ticketing ID, and Data Lake ID as External IDs in Salesforce, the middleware can perform upsert operations without needing to first query Salesforce for its internal ID. This reduces the number of API calls and simplifies the integration logic. Conversely, when Salesforce pushes a change to the ERP, it sends the stored ERP ID, allowing the ERP to instantly identify the correct target record.
Option B (Caching at the middleware) is a high-maintenance "anti-pattern" that introduces a new point of failure if the cache goes out of sync with the actual systems. Option C (Change Data Capture) is a mechanism for notifying systems of changes, but it does not solve the underlying identity mapping problem. Using External IDs creates a stable, searchable, and performant cross-reference that

is the backbone of any successful "hub-and-spoke" integration architecture.

質問 # 45
A company that is a leading provider of courses and training delivers courses using third-party trainers. The trainer for the company has to be verified by 10 different training accreditation verification agencies before providing training for the company. Each training accreditation agency has its own response time, which means it could take days to confirm a trainer. The company decided to automate the trainer accreditation verification process by integrating it with the agency's web service1s. What is the recommended approach to automa2te this process?3456

- A. Make an Apex callout using @future annotation to make the callout to all different agencies.
- B. Use middleware to handl7e the callout to the 10 different verific8ation services; the middleware will handle the business logic of con9solidating the verification resu10lt from the 10 services. Then, make a call-in to Salesforce and update the verification status to "verified".
- C. Use Salesforce External Service to make the callout; Salesforce External Service should check the verification agencies until the result is verified. Then, update the trainer status to "verified".

正解：B

解説：
In this scenario, the primary architectural challenge is managing high-latency, multi-step orchestration involving 10 disparate external systems. Each agency has a varying response time that can span several days, making a synchronous "Request-Reply" pattern within Salesforce technically impossible due to transaction timeout limits (maximum 120 seconds).
The recommended approach is to leverage Middleware as the orchestration and state-management layer. Middleware (such as an ESB or iPaaS) is specifically designed for Process Choreography. Salesforce initiates a single "Fire and Forget" request to the middleware. The middleware then takes responsibility for:
Sequential or Parallel Callouts: Initiating the requests to all 10 verification agencies.
Callback Management: Handling the asynchronous responses from each agency as they arrive over a period of days.
Aggregation Logic: Consolidating the results and determining when the "Business Process" is complete (e.g., all 10 agencies have approved).
Once the consolidation logic is satisfied, the middleware performs a Remote Call-In to the Salesforce REST API to update the trainer's record. This pattern keeps Salesforce "clean" by moving complex, long-running orchestration logic off-platform, preventing the consumption of excessive Apex CPU time and ensuring that Salesforce only receives a single, final status update.
Option B (External Services) is unsuitable for a multi-day asynchronous process as it is designed for real-time, synchronous Flow actions. Option C (@future) is restricted by the same 120-second timeout and cannot handle the "waiting" state required for days of verification. Using middleware provides the necessary Quality of Service (QoS), durability, and error handling required for such a critical enterprise compliance process.

質問 # 46
.....

- Plat-Arch-204復習テキスト □ Plat-Arch-204テストサンプル問題 □ Plat-Arch-204技術試験 □ ➠ www.xhs1991.com □で▶ Plat-Arch-204 ◀を検索し、無料でダウンロードしてくださいPlat-Arch-204復習攻略問題
- Plat-Arch-204問題と解答 □ Plat-Arch-204復習テキスト □ Plat-Arch-204基礎訓練 □ { www.goshiken.com } に移動し、➤ Plat-Arch-204 □を検索して無料でダウンロードしてくださいPlat-Arch-204日本語版トレーリング
- Plat-Arch-204受験料過去問 □ Plat-Arch-204日本語pdf問題 □ Plat-Arch-204最新日本語版参考書 □ 時間限定無料で使える➻ Plat-Arch-204 □の試験問題は➤ www.passtest.jp □サイトで検索Plat-Arch-204最新日本語版参考書
- Plat-Arch-204日本語pdf問題 □ Plat-Arch-204最新日本語版参考書 □ Plat-Arch-204日本語版トレーリング □ □ 今すぐ▷ www.goshiken.com ◁で⇛ Plat-Arch-204 ⇚を検索し、無料でダウンロードしてくださいPlat-Arch-204復習テキスト
- 最短突破 Plat-Arch-204 合格教本 □ [www.mogiexam.com]を開き、✔ Plat-Arch-204 □✔□を入力して、無料でダウンロードしてくださいPlat-Arch-204受験料過去問
- 最短突破 Plat-Arch-204 合格教本 □ Open Webサイト➤ www.goshiken.com □検索➠ Plat-Arch-204 □無料ダウンロードPlat-Arch-204基礎訓練
- 最短突破 Plat-Arch-204 合格教本 ⌨ 「 www.xhs1991.com 」サイトにて▶ Plat-Arch-204 ◀問題集を無料で使おうPlat-Arch-204技術試験
-