ACD301자격증참고서, ACD301인기자격증덤프문제

참고: Itexamdump에서 Google Drive로 공유하는 무료 2026 Appian ACD301 시험 문제집이 있습니다: https://drive.google.com/open?id=1mYEkkNiTatVbP7n4_0uL854II3q7Tqw8

Itexamdump 의 Appian인증 ACD301덤프는Appian인증 ACD301시험에 도전장을 던진 분들이 신뢰할수 있는 든든한 길잡이 입니다. Appian인증 ACD301시험대비 덤프뿐만아니라 다른 IT인증시험에 대비한 덤프자료도 적중율이 끝내줍니다. Appian인증 ACD301시험이나 다른 IT인증자격증시험이나Itexamdump제품을 사용해보세요.투자한 덤프 비용보다 훨씬 큰 이득을 보실수 있을것입니다.

우리Itexamdump 에서 제공하는 학습가이드에는 IT전문가들이 만들어낸 시험대비 자료들과Appian ACD301인증시험의 완벽한 문제와 답들입니다. 그리고 우리Itexamdump에서는 IT업계에서의 높은 신뢰감으로 여러분들한테 100% 보장을 드립니다. 우리에 믿음을 드리기 위하여Appian ACD301관련자료의 일부분 문제와 답 등 샘플을 무료로 다운받아 체험해볼수 있게 제공합니다.

>> ACD301자격증참고서 <<

ACD301인기자격증 덤프문제 & ACD301최신버전 덤프데모문제

인재가 넘치는 IT업계에서 자기의 자리를 지켜나가려면 학력보다 능력이 더욱 중요합니다.고객님의 능력을 증명해주는 수단은 국제적으로 승인받은 IT인증자격증이 아니겠습니까? Appian인증 ACD301시험이 어렵다고 하여 두려워 하지 마세요. IT자격증을 취득하려는 분들의 곁에는Itexamdump가 있습니다. Itexamdump의Appian인증 ACD301 시험준비를 하시고 시험패스하여 자격증을 취득하세요. 국제승인 자격증이라 고객님의 경쟁율을 업그레이드 시켜드립니다.

최신 Lead Developer ACD301 무료샘플문제 (Q10-Q15):

질문 # 10
You need to design a complex Appian integration to call a RESTful API. The RESTful API will be used to update a case in a customer's legacy system.
What are three prerequisites for designing the integration?

- A. Understand whether this integration will be used in an interface or in a process model.
- B. Understand the content of the expected body, including each field type and their limits.
- C. Understand the different error codes managed by the API and the process of error handling in Appian.
- D. Define the HTTP method that the integration will use.
- E. Understand the business rules to be applied to ensure the business logic of the data.

정답： B,C,D

설명:
Comprehensive and Detailed In-Depth Explanation:
As an Appian Lead Developer, designing a complex integration to a RESTful API for updating a case in a legacy system requires a structured approach to ensure reliability, performance, and alignment with business needs. The integration involves sending a JSON

payload (implied by the context) and handling responses, so the focus is on technical and functional prerequisites. Let's evaluate each option:
A . Define the HTTP method that the integration will use:
This is a primary prerequisite. RESTful APIs use HTTP methods (e.g., POST, PUT, GET) to define the operation-here, updating a case likely requires PUT or POST. Appian's Connected System and Integration objects require specifying the method to configure the HTTP request correctly. Understanding the API's method ensures the integration aligns with its design, making this essential for design. Appian's documentation emphasizes choosing the correct HTTP method as a foundational step.
B . Understand the content of the expected body, including each field type and their limits:
This is also critical. The JSON payload for updating a case includes fields (e.g., text, dates, numbers), and the API expects a specific structure with field types (e.g., string, integer) and limits (e.g., max length, size constraints). In Appian, the Integration object requires a dictionary or CDT to construct the body, and mismatches (e.g., wrong types, exceeding limits) cause errors (e.g., 400 Bad Request). Appian's best practices mandate understanding the API schema to ensure data compatibility, making this a key prerequisite.
C . Understand whether this integration will be used in an interface or in a process model:
While knowing the context (interface vs. process model) is useful for design (e.g., synchronous vs. asynchronous calls), it's not a prerequisite for the integration itself-it's a usage consideration. Appian supports integrations in both contexts, and the integration's design (e.g., HTTP method, body) remains the same. This is secondary to technical API details, so it's not among the top three prerequisites.
D . Understand the different error codes managed by the API and the process of error handling in Appian:
This is essential. RESTful APIs return HTTP status codes (e.g., 200 OK, 400 Bad Request, 500 Internal Server Error), and the customer's API likely documents these for failure scenarios (e.g., invalid data, server issues). Appian's Integration objects can handle errors via error mappings or process models, and understanding these codes ensures robust error handling (e.g., retry logic, user notifications). Appian's documentation stresses error handling as a core design element for reliable integrations, making this a primary prerequisite.
E . Understand the business rules to be applied to ensure the business logic of the data:
While business rules (e.g., validating case data before sending) are important for the overall application, they aren't a prerequisite for designing the integration itself-they're part of the application logic (e.g., process model or interface). The integration focuses on technical interaction with the API, not business validation, which can be handled separately in Appian. This is a secondary concern, not a core design requirement for the integration.
Conclusion: The three prerequisites are A (define the HTTP method), B (understand the body content and limits), and D (understand error codes and handling). These ensure the integration is technically sound, compatible with the API, and resilient to errors-critical for a complex RESTful API integration in Appian.
Reference:
Appian Documentation: "Designing REST Integrations" (HTTP Methods, Request Body, Error Handling).
Appian Lead Developer Certification: Integration Module (Prerequisites for Complex Integrations).
Appian Best Practices: "Building Reliable API Integrations" (Payload and Error Management).
To design a complex Appian integration to call a RESTful API, you need to have some prerequisites, such as:
Define the HTTP method that the integration will use. The HTTP method is the action that the integration will perform on the API, such as GET, POST, PUT, PATCH, or DELETE. The HTTP method determines how the data will be sent and received by the API, and what kind of response will be expected.
Understand the content of the expected body, including each field type and their limits. The body is the data that the integration will send to the API, or receive from the API, depending on the HTTP method. The body can be in different formats, such as JSON, XML, or form data. You need to understand how to structure the body according to the API specification, and what kind of data types and values are allowed for each field.
Understand the different error codes managed by the API and the process of error handling in Appian. The error codes are the status codes that indicate whether the API request was successful or not, and what kind of problem occurred if not. The error codes can range from 200 (OK) to 500 (Internal Server Error), and each code has a different meaning and implication. You need to understand how to handle different error codes in Appian, and how to display meaningful messages to the user or log them for debugging purposes.
The other two options are not prerequisites for designing the integration, but rather considerations for implementing it.
Understand whether this integration will be used in an interface or in a process model. This is not a prerequisite, but rather a decision that you need to make based on your application requirements and design. You can use an integration either in an interface or in a process model, depending on where you need to call the API and how you want to handle the response. For example, if you need to update a case in real-time based on user input, you may want to use an integration in an interface. If you need to update a case periodically based on a schedule or an event, you may want to use an integration in a process model.
Understand the business rules to be applied to ensure the business logic of the data. This is not a prerequisite, but rather a part of your application logic that you need to implement after designing the integration. You need to apply business rules to validate, transform, or enrich the data that you send or receive from the API, according to your business requirements and logic. For example, you may need to check if the case status is valid before updating it in the legacy system, or you may need to add some additional information to the case data before displaying it in Appian.

질문 # 11
For each requirement, match the most appropriate approach to creating or utilizing plug-ins Each approach will be used once.
Note: To change your responses, you may deselect your response by clicking the blank space at the top of the selection list.

정답：

설명：

질문 # 12
Your team has deployed an application to Production with an underperforming view. Unexpectedly, the production data is ten times that of what was tested, and you must remediate the issue. What is the best option you can take to mitigate their performance concerns?

- A. Introduce a data management policy to reduce the volume of data.
- B. Bypass Appian's query rule by calling the database directly with a SQL statement.
- C. Create a materialized view or table.
- D. Create a table which is loaded every hour with the latest data.

정답：C

설명：
Comprehensive and Detailed In-Depth Explanation:
As an Appian Lead Developer, addressing performance issues in production requires balancing Appian's best practices, scalability, and maintainability. The scenario involves an underperforming view due to a significant increase in data volume (ten times the tested amount), necessitating a solution that optimizes performance while adhering to Appian's architecture. Let's evaluate each option:
A . Bypass Appian's query rule by calling the database directly with a SQL statement:
This approach involves circumventing Appian's query rules (e.g., a!queryEntity) and directly executing SQL against the database. While this might offer a quick performance boost by avoiding Appian's abstraction layer, it violates Appian's core design principles. Appian Lead Developer documentation explicitly discourages direct database calls, as they bypass security (e.g., Appian's row-level security), auditing, and portability features. This introduces maintenance risks, dependencies on database-specific logic, and potential production instability-making it an unsustainable and non-recommended solution.
B . Create a table which is loaded every hour with the latest data:
This suggests implementing a staging table updated hourly (e.g., via an Appian process model or ETL process). While this could reduce query load by pre-aggregating data, it introduces latency (data is only fresh hourly), which may not meet real-time requirements typical in Appian applications (e.g., a customer-facing view). Additionally, maintaining an hourly refresh process adds complexity and overhead (e.g., scheduling, monitoring). Appian's documentation favors more efficient, real-time solutions over periodic refreshes unless explicitly required, making this less optimal for immediate performance remediation.
C . Create a materialized view or table:
This is the best choice. A materialized view (or table, depending on the database) pre-computes and stores query results, significantly improving retrieval performance for large datasets. In Appian, you can integrate a materialized view with a Data Store Entity, allowing a!queryEntity to fetch data efficiently without changing application logic. Appian Lead Developer training emphasizes leveraging database optimizations like materialized views to handle large data volumes, as they reduce query execution time while keeping data consistent with the source (via periodic or triggered refreshes, depending on the database). This aligns with Appian's performance optimization guidelines and addresses the tenfold data increase effectively.
D . Introduce a data management policy to reduce the volume of data:
This involves archiving or purging data to shrink the dataset (e.g., moving old records to an archive table). While a long-term data management policy is a good practice (and supported by Appian's Data Fabric principles), it doesn't immediately remediate the performance issue. Reducing data volume requires business approval, policy design, and implementation-delaying resolution. Appian documentation recommends combining such strategies with technical fixes (like C), but as a standalone solution, it's insufficient for urgent production concerns.
Conclusion: Creating a materialized view or table (C) is the best option. It directly mitigates performance by optimizing data retrieval, integrates seamlessly with Appian's Data Store, and scales for large datasets-all while adhering to Appian's recommended practices. The view can be refreshed as needed (e.g., via database triggers or schedules), balancing performance and data freshness. This approach requires collaboration with a DBA to implement but ensures a robust, Appian-supported solution.
Reference:
Appian Documentation: "Performance Best Practices" (Optimizing Data Queries with Materialized Views).
Appian Lead Developer Certification: Application Performance Module (Database Optimization Techniques).
Appian Best Practices: "Working with Large Data Volumes in Appian" (Data Store and Query Performance).

질문 # 13
You are selling up a new cloud environment. The customer already has a system of record for Its employees and doesn't want to re-create them in Appian. so you are going to Implement LDAP authentication.
What are the next steps to configure LDAP authentication?
To answer, move the appropriate steps from the Option list to the Answer List area, and arrange them in the correct order. You may or may not use all the steps.

정답：

설명：
Explanation:
* Navigate to the Admin console > Authentication > LDAP. This is the first step, as it allows you to access the settings and options for LDAP authentication in Appian.
* Work with the customer LDAP point of contact to obtain the LDAP authentication xsd. Import the xsd file in the Admin console. This is the second step, as it allows you to define the schema and structure of the LDAP data that will be used for authentication in Appian. You will need to work with the customer LDAP point of contact to obtain the xsd file that matches their LDAP server configuration and data model. You will then need to import the xsd file in the Admin console using the Import Schema button.
* Enable LDAP and enter the LDAP parameters, such as the URL of the LDAP server and plaintext credentials. This is the third step, as it allows you to enable and configure the LDAP authentication in Appian. You will need to check the Enable LDAP checkbox and enter the required parameters, such as the URL of the LDAP server, the plaintext credentials for connecting to the LDAP server, and the base DN for searching for users in the LDAP server.
* Test the LDAP integration and see if it succeeds. This is the fourth and final step, as it allows you to verify and validate that the LDAP authentication is working properly in Appian. You will need to use the Test Connection button to test if Appian can connect to the LDAP server successfully.
You will also need to use the Test User Lookup button to test if Appian can find and authenticate a user from the LDAP server using their username and password.
Configuring LDAP authentication in Appian Cloud allows the platform to leverage an existing employee system of record (e.g., Active Directory) for user authentication, avoiding manual user creation. Theprocess involves a series of steps within the Appian Administration Console, guided by Appian's Security and Authentication documentation. The steps must be executed in a logical order to ensure proper setup and validation.
* Navigate to the Admin Console > Authentication > LDAP:The first step is to access the LDAP configuration section in the Appian Administration Console. This is the entry point for enabling and configuring LDAP authentication, where administrators can define the integration settings. Appian requires this initial navigation to begin the setup process.
* Work with the customer LDAP point-of-contact to obtain the LDAP authentication xsd. Import the xsd file in the Admin Console:The next step involves gathering the LDAP schema definition (xsd file) from the customer's LDAP system (e.g., via their point-of-contact). This file defines the structure of the LDAP directory (e.g., user attributes). Importing it into the Admin Console allows Appian to map these attributes to its user model, a critical step before enabling authentication, as outlined in Appian's LDAP Integration Guide.
* Enable LDAP and enter the appropriate LDAP parameters, such as the URL of the LDAP server and plaintext credentials:After importing the schema, enable LDAP and configure the connection details. This includes specifying the LDAP server URL (e.g., ldap://ldap.example.com) and plaintext credentials (or a secure alternative like LDAPS with certificates). These parameters establish the connection to the customer's LDAP system, a prerequisite for testing, as per Appian's security best practices.
* Test the LDAP integration and save if it succeeds:The final step is to test the configuration to ensure Appian can authenticate against the LDAP server. The Admin Console provides a test option to verify connectivity and user synchronization. If successful, saving the configuration applies the settings, completing the setup. Appian recommends this validation step to avoid misconfigurations, aligning with the iterative testing approach in the documentation.
Unused Option:
* Enter two parameters: the URL of the LDAP server and plaintext credentials:This step is redundant and not used. The equivalent action is covered under "Enable LDAP and enter the appropriate LDAP parameters," which is more comprehensive and includes enablingthe feature.
Including both would be duplicative, and Appian's interface consolidates parameter entry with enabling.
Ordering Rationale:
* The sequence follows a logical workflow: navigation to the configuration area, schema import for structure, parameter setup for connectivity, and testing/saving for validation. This aligns with Appian's step-by-step LDAP setup process, ensuring each step builds on the previous one without requiring backtracking.
* The unused option reflects the question's allowance for not using all steps, indicating flexibility in the process.
References:Appian Documentation - Security and Authentication Guide, Appian Administration Console - LDAP Configuration, Appian Lead Developer Training - Integration Setup.

질문 # 14
Users must be able to navigate throughout the application while maintaining complete visibility in the application structure and easily navigate to previous locations. Which Appian Interface Pattern would you recommend?

- A. Include a Breadcrumbs pattern on applicable interfaces to show the organizational hierarchy.
- B. Use Billboards as Cards pattern on the homepage to prominently display application choices.
- C. Implement an Activity History pattern to track an organization's activity measures.
- D. Implement a Drilldown Report pattern to show detailed information about report data.

정답：A

설명:
Comprehensive and Detailed In-Depth Explanation:The requirement emphasizes navigation with complete visibility of the application structure and the ability to return to previous locations easily. TheBreadcrumbs patternis specifically designed to meet this need. According to Appian's design best practices, the Breadcrumbs pattern provides a visual trail of the user's navigation path, showing the hierarchy of pages or sections within the application. This allows users to understand their current location relative to the overall structure and quickly navigate back to previous levels by clicking on the breadcrumb links.
* Option A (Billboards as Cards):This pattern is useful for presenting high-level options or choices on a homepage in a visually appealing way. However, it does not address navigation visibility or the ability to return to previous locations, making it irrelevant to the requirement.
* Option B (Activity History):This pattern tracks and displays a log of activities or actions within the application, typically for auditing or monitoring purposes. It does not enhance navigation or provide visibility into the application structure.
* Option C (Drilldown Report):This pattern allows users to explore detailed data within reports by drilling into specific records. While it supports navigation within data, it is not designed for general application navigation or maintaining structural visibility.
* Option D (Breadcrumbs):This is the correct choice as it directly aligns with the requirement. Per Appian's Interface Patterns documentation, Breadcrumbs improve usability by showing ahierarchical path (e.g., Home > Section > Subsection) and enabling backtracking, fulfilling both visibility and navigation needs.
References:Appian Design Guide - Interface Patterns (Breadcrumbs section), Appian Lead Developer Training - User Experience Design Principles.

질문 # 15
.....

Appian ACD301 덤프로 많은 분들께서 Appian ACD301시험을 패스하여 자격증을 취득하게 도와드렸지만 저희는 자만하지않고 항상 초심을 잊지않고 더욱더 퍼펙트한Appian ACD301덤프를 만들기 위해 모든 심여를 기울일것을 약속드립니다.

ACD301인기자격증 덤프문제: https://www.itexamdump.com/ACD301.html

이는Itexamdump ACD301인기자격증 덤프문제 의 IT전문가가 오랜 시간동안 IT인증시험을 연구한 끝에 시험대비자료로 딱 좋은 덤프를 제작한 결과입니다, 우리는Appian ACD301인증시험관련 모든 자료를 여러분들에서 제공할 것입니다, Itexamdump의 Appian인증 ACD301덤프는 업계에서 널리 알려진 최고품질의Appian인증 ACD301시험대비자료입니다, Itexamdump의Appian인증 ACD301덤프는 시험패스율이 거의 100%에 달하여 많은 사랑을 받아왔습니다, Appian인증 ACD301시험준비자료는 Itexamdump에서 마련하시면 기적같은 효과를 안겨드립니다.

대체 이게 다 얼마야, 하는 생각은 들었지만 별로 부담스럽다거나 하지는 않았다, 상대ACD301가 물을 뿌리면 잽싸게 피해 버린다든가, 이는Itexamdump 의 IT전문가가 오랜 시간동안 IT인증시험을 연구한 끝에 시험대비자료로 딱 좋은 덤프를 제작한 결과입니다.

시험패스에 유효한 ACD301자격증참고서 인증시험정보

우리는Appian ACD301인증시험관련 모든 자료를 여러분들에서 제공할 것입니다, Itexamdump의 Appian인증 ACD301덤프는 업계에서 널리 알려진 최고품질의Appian인증 ACD301시험대비자료입니다.

Itexamdump의Appian인증 ACD301덤프는 시험패스율이 거의 100%에 달하여 많은 사랑을 받아왔습니다, Appian인증 ACD301시험준비자료는 Itexamdump에서 마련하시면 기적같은 효과를 안겨드립니다.

- 최근 인기시험 ACD301자격증참고서 덤프 □ 오픈 웹 사이트{ www.itdumpskr.com }검색➡ ACD301 □□□무료 다운로드ACD301유효한 최신덤프자료
- 완벽한 ACD301자격증참고서 시험덤프공부 □ 무료로 다운로드하려면▷ www.koreadumps.com◁로 이동하여《 ACD301 》를 검색하십시오ACD301인기자격증 시험 덤프자료
- ACD301공부자료 □ ACD301시험패스 인증덤프 □ ACD301시험응시 □ 검색만 하면✔ www.itdumpskr.com □✔□에서▷ ACD301 ◁무료 다운로드ACD301유효한 최신덤프자료
- 100% 유효한 ACD301자격증참고서 시험대비자료 □ ⇛ www.passtip.net ⇚에서 검색만 하면（ ACD301 ）를 무료로 다운로드할 수 있습니다ACD301덤프샘플 다운
- 완벽한 ACD301자격증참고서 시험덤프공부 □ 지금[www.itdumpskr.com]을(를) 열고 무료 다운로드를 위해 □ ACD301 □를 검색하십시오ACD301시험패스 가능한 공부문제
- ACD301자격증참고서 덤프 ----- IT전문가의 노하우로 만들어진 시험자료 □ 지금▸ www.exampassdump.com ◂을(를) 열고 무료 다운로드를 위해▸ ACD301 ◂를 검색하십시오ACD301인증시험 인기 덤프문제
- 100% 유효한 ACD301자격증참고서 시험대비자료 □ ✔ www.itdumpskr.com □✔□을(를) 열고➡ ACD301 □□□를 입력하고 무료 다운로드를 받으십시오ACD301시험패스 가능한 공부문제
- 높은 통과율 ACD301자격증참고서 시험공부 □ 무료 다운로드를 위해 지금➽ www.dumptop.com □에서☀ ACD301 □☀□검색ACD301최고품질 인증시험덤프데모
- ACD301자격증참고서 시험준비에 가장 좋은 기출문제 모은 덤프자료 □ 검색만 하면「 www.itdumpskr.com 」에서➤ ACD301 □무료 다운로드ACD301유효한 최신덤프자료
- 최근 인기시험 ACD301자격증참고서 덤프 □ □ www.dumptop.com □을(를) 열고□ ACD301 □를 검색하여 시험 자료를 무료로 다운로드하십시오ACD301최신버전 덤프자료
- www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, bbs.t-firefly.com, Disposable vapes

그 외, Itexamdump ACD301 시험 문제집 일부가 지금은 무료입니다: https://drive.google.com/open?id=1mYEkkNiTatVbP7n4_0uL854II3q7Tqw8