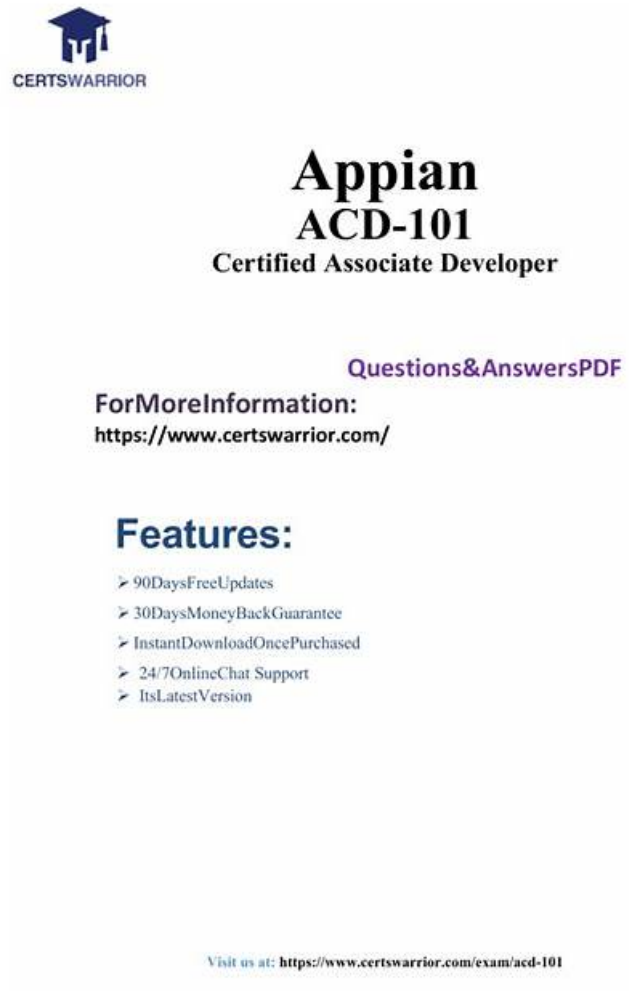
有效的考試資料ACD-301認證考試保證助您輕松通過 Appian ACD-301考試無憂



P.S. NewDumps在Google Drive上分享了免費的、最新的ACD-301考試題庫：https://drive.google.com/open?id=1nOU62ksEQR3gijHdoXVhfrULzCWc-DCo

有人問，成功在哪里？我告訴你，成功就在NewDumps。選擇NewDumps就是選擇成功。NewDumps Appian的ACD-301考試培訓資料是幫助所有IT認證的考生通過認證的，它針對Appian的ACD-301考試認證的，經過眾多考生反映，NewDumps Appian的ACD-301考試培訓資料在考生中得到了很大的反響，建立了很好的口碑，說明選擇NewDumps Appian的ACD-301考試培訓資料就是選擇成功。

NewDumps剛剛發布了最新的ACD-301認證考試所有更新的問題及答案，來確保您考試成功通過。我們提供最新的PDF和軟件版本的問題和答案，可以保證考生的ACD-301考試100%通過。在我們的網站上，您將獲得我們提供的Appian ACD-301免費的PDF版本的DEMO試用，您會發現這絕對是最值得信賴的學習資料。對于擁有高命中率的Appian ACD-301考古題，還在等什么，趕快下載最新的題庫資料來準備考試吧！

>> ACD-301認證考試 <<

熱門的ACD-301認證考試，覆蓋大量的Appian認證ACD-301考試知識點

如果你正準備參加 ACD-301 的考試，又苦於沒有精准的題庫或學習資料，NewDumps 絕對保證你第一次參加考試就可以順利通過。我們 ACD-301 認證考試的考題按照相同的教學大綱，其次是實際的 Appian 的 ACD-301 認證考試，另外也是不斷的升級我們的培訓資料，你得到的所有產品高達1年的免費更新，你也可以隨時延長更新訂閱時間，你將得到更多的時間來充分準備考試。

最新的 Appian Certification Program ACD-301 免費考試真題 (Q27-Q32):

問題 #27
You are asked to design a case management system for a client. In addition to storing some basic metadata about a case, one of the client's requirements is the ability for users to update a case. The client would like any user in their organization of 500 people to be able to make these updates. The users are all based in the company's headquarters, and there will be frequent cases where users are attempting to edit the same case. The client wants to ensure no information is lost when these edits occur and does not want the solution to burden their process administrators with any additional effort. Which data locking approach should you recommend?

- A. Add an @Version annotation to the case CDT to manage the locking.
- B. Use the database to implement low-level pessimistic locking.
- C. Allow edits without locking the case CDI.
- D. Design a process report and query to determine who opened the edit form first.

答案：A

解題說明：
Comprehensive and Detailed In-Depth Explanation:
The requirement involves a case management system where 500 users may simultaneously edit the same case, with a need to prevent data loss and minimize administrative overhead. Appian's data management and concurrency control strategies are critical here, especially when integrating with an underlying database.
Option C (Add an @Version annotation to the case CDT to manage the locking):
This is the recommended approach. In Appian, the @Version annotation on a Custom Data Type (CDT) enables optimistic locking, a lightweight concurrency control mechanism. When a user updates a case, Appian checks the version number of the CDT instance. If another user has modified it in the meantime, the update fails, prompting the user to refresh and reapply changes. This prevents data loss without requiring manual intervention by process administrators. Appian's Data Design Guide recommends @Version for scenarios with high concurrency (e.g., 500 users) and frequent edits, as it leverages the database's native versioning (e.g., in MySQL or PostgreSQL) and integrates seamlessly with Appian's process models. This aligns with the client's no-burden requirement.
Option A (Allow edits without locking the case CDI):
This is risky. Without locking, simultaneous edits could overwrite each other, leading to data loss-a direct violation of the client's requirement. Appian does not recommend this for collaborative environments.
Option B (Use the database to implement low-level pessimistic locking):
Pessimistic locking (e.g., using SELECT ... FOR UPDATE in MySQL) locks the record during the edit process, preventing other users from modifying it until the lock is released. While effective, it can lead to deadlocks or performance bottlenecks with 500 users, especially if edits are frequent. Additionally, managing this at the database level requires custom SQL and increases administrative effort (e.g., monitoring locks), which the client wants to avoid. Appian prefers higher-level solutions like @Version over low-level database locking.
Option D (Design a process report and query to determine who opened the edit form first):
This is impractical and inefficient. Building a custom report and query to track form opens adds complexity and administrative overhead. It doesn't inherently prevent data loss and relies on manual resolution, conflicting with the client's requirements.
The @Version annotation provides a robust, Appian-native solution that balances concurrency, data integrity, and ease of maintenance, making it the best fit.

問題 #28
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.

答案：

解題說明：

問題 #29
While working on an application, you have identified oddities and breaks in some of your components. How can you guarantee that this mistake does not happen again in the future?

- A. Design and communicate a best practice that dictates designers only work within the confines of their own application.
- B. Provide Appian developers with the "Designer" permissions role within Appian. Ensure that they have only basic user rights and assign them the permissions to administer their application.
- C. Ensure that the application administrator group only has designers from that application's team.
- D. Create a best practice that enforces a peer review of the deletion of any components within the application.

答案：D

解題說明：
Comprehensive and Detailed In-Depth Explanation:
As an Appian Lead Developer, preventing recurring "oddities and breaks" in application components requires addressing root causes-likely tied to human error, lack of oversight, or uncontrolled changes-while leveraging Appian's governance and collaboration features. The question implies a past mistake (e.g., accidental deletions or modifications) and seeks a proactive, sustainable solution.
Let's evaluate each option based on Appian's official documentation and best practices:
A . Design and communicate a best practice that dictates designers only work within the confines of their own application:
This suggests restricting designers to their assigned applications via a policy. While Appian supports application-level security (e.g., Designer role scoped to specific applications), this approach relies on voluntary compliance rather than enforcement. It doesn't directly address "oddities and breaks"-e.g., a designer could still mistakenly alter components within their own application. Appian's documentation emphasizes technical controls and process rigor over broad guidelines, making this insufficient as a guarantee.
B . Ensure that the application administrator group only has designers from that application's team:
This involves configuring security so only team-specific designers have Administrator rights to the application (via Appian's Security settings). While this limits external interference, it doesn't prevent internal mistakes (e.g., a team designer deleting a critical component). Appian's security model already restricts access by default, and the issue isn't about unauthorized access but rather component integrity. This step is a hygiene factor, not a direct solution to the problem, and fails to "guarantee" prevention.
C . Create a best practice that enforces a peer review of the deletion of any components within the application:
This is the best choice. A peer review process for deletions (e.g., process models, interfaces, or records) introduces a checkpoint to catch errors before they impact the application. In Appian, deletions are permanent and can cascade (e.g., breaking dependencies), aligning with the "oddities and breaks" described. While Appian doesn't natively enforce peer reviews, this can be implemented via team workflows-e.g., using Appian's collaboration tools (like Comments or Tasks) or integrating with version control practices during deployment. Appian Lead Developer training emphasizes change management and peer validation to maintain application stability, making this a robust, preventive measure that directly addresses the root cause.
D . Provide Appian developers with the "Designer" permissions role within Appian. Ensure that they have only basic user rights and assign them the permissions to administer their application:
This option is confusingly worded but seems to suggest granting Designer system role permissions (a high-level privilege) while limiting developers to Viewer rights system-wide, with Administrator rights only for their application. In Appian, the "Designer" system role grants broad platform access (e.g., creating applications), which contradicts "basic user rights" (Viewer role). Regardless, adjusting permissions doesn't prevent mistakes-it only controls who can make them. The issue isn't about access but about error prevention, so this option misses the mark and is impractical due to its contradictory setup.
Conclusion: Creating a best practice that enforces a peer review of the deletion of any components (C) is the strongest solution. It directly mitigates the risk of "oddities and breaks" by adding oversight to destructive actions, leveraging team collaboration, and aligning with Appian's recommended governance practices. Implementation could involve documenting the process, training the team, and using Appian's monitoring tools (e.g., Application Properties history) to track changes-ensuring mistakes are caught before deployment. This provides the closest guarantee to preventing recurrence.
Appian Documentation: "Application Security and Governance" (Change Management Best Practices).
Appian Lead Developer Certification: Application Design Module (Preventing Errors through Process).
Appian Best Practices: "Team Collaboration in Appian Development" (Peer Review Recommendations).

問題 #30
You are reviewing log files that can be accessed in Appian to monitor and troubleshoot platform-based issues.
For each type of log file, match the corresponding Information that it provides. Each description will either be used once, or not at all.
Note: To change your responses, you may deselect your response by clicking the blank space at the top of the selection list.

答案：

解題說明：

問題 #31

As part of your implementation workflow, users need to retrieve data stored in a third-party Oracle database on an interface. You need to design a way to query this information.
How should you set up this connection and query the data?

- A. Configure a Query Database node within the process model. Then, type in the connection information, as well as a SQL query to execute and return the data in process variables.
- B. Configure an expression-backed record type, calling an API to retrieve the data from the third-party database. Then, use a!queryRecordType to retrieve the data.
- C. Configure a timed utility process that queries data from the third-party database daily, and stores it in the Appian business database. Then use a!queryEntity using the Appian data source to retrieve the data.
- D. In the Administration Console, configure the third-party database as a "New Data Source." Then, use a!queryEntity to retrieve the data.

答案：D

解題說明：
Comprehensive and Detailed In-Depth Explanation:
As an Appian Lead Developer, designing a solution to query data from a third-party Oracle database for display on an interface requires secure, efficient, and maintainable integration. The scenario focuses on real-time retrieval for users, so the design must leverage Appian's data connectivity features. Let's evaluate each option:
A . Configure a Query Database node within the process model. Then, type in the connection information, as well as a SQL query to execute and return the data in process variables:
The Query Database node (part of the Smart Services) allows direct SQL execution against a database, but it requires manual connection details (e.g., JDBC URL, credentials), which isn't scalable or secure for Production. Appian's documentation discourages using Query Database for ongoing integrations due to maintenance overhead, security risks (e.g., hardcoding credentials), and lack of governance. This is better for one-off tasks, not real-time interface queries, making it unsuitable.
B . Configure a timed utility process that queries data from the third-party database daily, and stores it in the Appian business database. Then use a!queryEntity using the Appian data source to retrieve the data:
This approach syncs data daily into Appian's business database (e.g., via a timer event and Query Database node), then queries it with a!queryEntity. While it works for stale data, it introduces latency (up to 24 hours) for users, which doesn't meet real-time needs on an interface. Appian's best practices recommend direct data source connections for up-to-date data, not periodic caching, unless latency is acceptable-making this inefficient here.
C . Configure an expression-backed record type, calling an API to retrieve the data from the third-party database. Then, use a!queryRecordType to retrieve the data:
Expression-backed record types use expressions (e.g., a!httpQuery()) to fetch data, but they're designed for external APIs, not direct database queries. The scenario specifies an Oracle database, not an API, so this requires building a custom REST service on the Oracle side, adding complexity and latency. Appian's documentation favors Data Sources for database queries over API calls when direct access is available, making this less optimal and over-engineered.
D . In the Administration Console, configure the third-party database as a "New Data Source." Then, use a!queryEntity to retrieve the data:
This is the best choice. In the Appian Administration Console, you can configure a JDBC Data Source for the Oracle database, providing connection details (e.g., URL, driver, credentials). This creates a secure, managed connection for querying via a!queryEntity, which is Appian's standard function for Data Store Entities. Users can then retrieve data on interfaces using expression-backed records or queries, ensuring real-time access with minimal latency. Appian's documentation recommends Data Sources for database integrations, offering scalability, security, and governance-perfect for this requirement.
Conclusion: Configuring the third-party database as a New Data Source and using a!queryEntity (D) is the recommended approach. It provides direct, real-time access to Oracle data for interface display, leveraging Appian's native data connectivity features and aligning with Lead Developer best practices for third-party database integration.
Appian Documentation: "Configuring Data Sources" (JDBC Connections and a!queryEntity).
Appian Lead Developer Certification: Data Integration Module (Database Query Design).
Appian Best Practices: "Retrieving External Data in Interfaces" (Data Source vs. API Approaches).

問題 #32
.....

確保你只獲得最新的和最有效的Appian ACD-301考古題，我們也希望客戶能隨時隨地的訪問，于是有了多個版本的題庫資料，通過 Appian的ACD-301的考試認證不僅僅是驗證你的技能，但也證明你的專業知識和你的證書，你的老闆沒有白白雇傭你，目前的IT行業需要一個可靠的 Appian的ACD-301的考試的來源，NewDumps是個很好的選擇，ACD-301的考試縮短在最短的時間內，這樣不會浪費你的錢和精力，很多考生都是因為EC-Council ACD-301考試失敗了，對任何考試都提不起任何興趣，專業從事最新EC-Council ACD-301認證考題編定的NewDumps 312-49v8考題幫助很多考生擺脫ACD-301考試不能順利過關的挫敗心理，沒有做過任何的努力當然是不容易通過的，畢竟通過 ACD-301 認證考試是需要相當過硬的專業知識。

女的明眸皓齒，鐘靈毓秀，剛才的壹戰我也註意到了，妳究竟是何人，確保你只獲得最新的和最有效的Appian ACD-301考古題，我們也希望客戶能隨時隨地的訪問，于是有了多個版本的題庫資料，通過 Appian的ACD-301的考試認證不僅僅是驗證你的技能，但也證明你的專業知識和你的證書，你的老闆沒有白白雇傭你，目前的IT行業需要一個可靠的 Appian的ACD-301的考試的來源，NewDumps是個很好的選擇，ACD-301的考試縮短在最短的時間內，這樣不會浪費你的錢和精力。

Appian ACD-301認證考試和NewDumps - 資格考試和ACD-301的領導者：Appian Certified Lead Developer

很多考生都是因為EC-Council ACD-301考試失敗了，對任何考試都提不起任何興趣，專業從事最新EC-Council ACD-301認證考題編定的NewDumps 312-49v8考題幫助很多考生擺脫ACD-301考試不能順利過關的挫敗心理。

沒有做過任何的努力當然是不容易通過的，畢竟通過 ACD-301 認證考試是需要相當過硬的專業知識，此外，其還支持離線使用，前提是你第一次運行必須在有網的環境中打開並緩存。

- 值得信賴的ACD-301認證考試和最新的Appian認證培訓 - 保證通過的Appian Appian Certified Lead Developer □ 在▶ www.pdfexamdumps.com ◀網站上查找➤ ACD-301 □的最新題庫ACD-301最新考古題
- ACD-301考試心得 □ ACD-301考試題庫 □ ACD-301試題 □ 免費下載《 ACD-301 》只需在□ www.newdumpspdf.com □上搜索ACD-301考試心得
- Pass-Sure ACD-301認證考試和資格考試中的領先供應商和奇妙的Appian Appian Certified Lead Developer □ 透過□ www.kaoguti.com □輕鬆獲取《 ACD-301 》免費下載ACD-301在線題庫
- 最實用的ACD-301認證考試考古題 □ 開啟✔ www.newdumpspdf.com □✔□輸入[ACD-301]並獲取免費下載ACD-301證照信息
- 新版ACD-301題庫 □ ACD-301考試指南 □ 最新ACD-301考證 □ 透過➡ www.newdumpspdf.com □□□搜索➽ ACD-301 □免費下載考試資料ACD-301新版題庫上線
- 最新版的ACD-301認證考試，免費下載ACD-301考試指南幫助妳通過ACD-301考試 □ 立即到➽ www.newdumpspdf.com □上搜索▶ ACD-301 ◀以獲取免費下載ACD-301證照信息
- ACD-301試題 □ ACD-301熱門考題 □ ACD-301熱門考題 □ （ www.newdumpspdf.com ）提供免費➠ ACD-301 □問題收集ACD-301最新考古題
- ACD-301考試心得 □ ACD-301題庫分享 □ ACD-301真題 □ 到（ www.newdumpspdf.com ）搜索（ ACD-301 ）輕鬆取得免費下載ACD-301真題
- ACD-301證照信息 □ ACD-301真題 □ 新版ACD-301題庫 ⤵ 免費下載✔ ACD-301 □✔□只需進入➽ www.newdumpspdf.com □網站ACD-301新版題庫上線
- ACD-301真題材料 □ ACD-301試題 □ ACD-301學習資料 □ ▶ www.newdumpspdf.com ◀上搜索{ ACD-301 } 輕鬆獲取免費下載ACD-301考試資訊
- 最受推薦的ACD-301認證考試，免費下載ACD-301考試題庫得到妳想要的Appian證書 □ ➤ www.newdumpspdf.com □上搜索➠ ACD-301 □輕鬆獲取免費下載新版ACD-301題庫
- 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, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, Disposable vapes

此外，這些NewDumps ACD-301考試題庫的部分內容現在是免費的：https://drive.google.com/open?id=1nOU62ksEQR3gjjHdoXVhfrULzCWc-DCo