ACD300$^{Q\&As}$

Appian Certified Lead Developer

Pass Appian ACD300 Exam with 100% Guarantee

QUESTION 1

You are designing a process that is anticipated to be executed multiple times a day. This process retrieves data from an external system and then calls various utility processes as needed. The mam process will not use the results of the utility processes, and there are no user forms anywhere.

Which design choice should be used to start the utility processes and minimize the load on the execution engines?

A. Use the Start Process Smart Service to start the utility processes.

B. Start the utility processes via a subprocess synchronously.

C. Use Process Messaging lo star! the utility process.

D. Start the utility processes via a subprocess asynchronously

Correct Answer: C

To design a process that is anticipated to be executed multiple times a day, that retrieves data from an external system and then calls various utility processes as needed, you should use Process Messaging to start the utility process and minimize the load on the execution engines. Process Messaging is a feature that allows you to send and receive messages between processes in Appian. By using Process Messaging, you can start the utility process asynchronously, which means that the main process does not have to wait for the utility process to finish before continuing. This way, you can improve the performance and scalability of your process design, and reduce the load on the execution engines. The other options are not as effective. Option A, using the Start Process Smart Service to start the utility processes, would also start the utility process asynchronously, but it would require more configuration and maintenance than Process Messaging. Option B, starting the utility processes via a subprocess synchronously, would start the utility process as a part of the main process flow, which means that the main process would have to wait for the utility process to finish before continuing. This would reduce the performance and scalability of your process design, and increase the load on the execution engines. Option D, starting the utility processes via a subprocess asynchronously, would also start the utility process as a part of the main process flow,but it would not wait for the utility process to finish before continuing. However, this option would still create more overhead than Process Messaging, as it would create more instances of processes in Appian.

QUESTION 2

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. Define the HTTP method that the integration will use.

B. Understand the content of the expected body. Deluding each field type and their limits

C. Understand whether this integration will be used in an interface or in a process model

D. Understand the different error codes managed by the API and the process of error handing m Appall

E. Understand the business rules to be applied to ensure the business logic of the data

Correct Answer: ABD

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.

QUESTION 3

You are required to create an integration from your Appian cloud instance to an application hosted within a customers self-managed environment.

The customers IT team has provided you with a REST API endpoint to test with; httpsV/lnternal networkVapi/api /ping

Which recommendation should you make to progress this integration?

A. Expose the API as a SOAP-basedweb service.

B. Deploy the API / service into Appian Cloud

C. Add Appian Cloud\\'s IP address ranges lo the customer network\\'s allowed IP listing

D. Set up a VPN tunnel

Correct Answer: D

To create an integration from your Appian cloud instance to an application hosted within a customer\\'s self-managed environment, you need to ensure that there is a secure and reliable connection between the two systems. One way to do this is to set up a VPN tunnel, which is a virtual private network that encrypts and transports data over the internet. A VPN tunnel allows you to access the customer\\'s internal network and API endpoint from your Appian cloud instance, without exposing them to the public internet. This way, you can ensure the security and privacy of the data that is exchanged between the two systems. Verified References: [Appian Cloud VPN], [Appian Integration Guide]

QUESTION 4

You are just starting with a new team that has been working together on an application for months. They ask you toreview some of their views thathave been degrading inperformance. The viewsare highly complex with hundreds of lines of SOL

What is the first step in troubleshooting the degradation?

A. Go through the entire database structure to obtain on overview, ensure you understand the business needs, andthen normalize the tables to optimizeperformance.

B. Run an explain statement on the views, identify criticalareas of improvement that can be remediated and without business knowledge

C. Go through all of the tables one by one to identify which of the grouped by. ordered by.or joined keys are currently indexed

D. Browse through the tables, note any tables that contain a large volume of null values, and work with your team to plan for table restructure.

Correct Answer: B

The first step in troubleshooting the degradation of the views is to run an explain statement on the views, identify critical areas of improvement that can be remediated without business knowledge. An explain statement is a tool that shows how a database executes a query or a view, and provides information about the cost, plan, and steps involved in the execution. By running an explain statement on the views, you can identify any inefficiencies or bottlenecks that are causing the degradation, such as missing indices, full table scans, nested loops, or hash joins. You can then apply some basic optimization techniques that do not require business knowledge, such as creating indices, limiting the number of columns or rows returned, using joins instead of subqueries, or using materialized views. Verified References: Appian Documentation, section "Query Optimization".

QUESTION 5

You are presented with the following application requirement:

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. Use Bullous as Cards pattern on the home page lo prominently display application choices.

B. Implement an Activity History pattern to track an organizations activity measures.

C. implement a drilldown report pattern to show detailed information about report data.

D. Include a breadcrumbs pattern on applicable inert aces to show the organizational hierarchy

Correct Answer: C

To meet the application requirement of allowing users to view summary and detailed information about report data, you should implement a drilldown report pattern to show detailed information about report data. A drilldown report pattern is a user interface component that displays data in a hierarchical structure, and allows users to expand or collapse different levels of data. For example, if the user is viewing a sales report by region, the drilldown report pattern could show something like "North America > USA > California > Los Angeles". The user can click on any level of data to see more or less details. This way, the user can see both summary and detailed information about report data, and explore different aspects of the data. The other options are not as effective. Option A, using Tiles as Cards pattern on the home page to prominently display application choices, would provide a way for users to access different parts of the application from the home page, but it would not show summary or detailed information about report data. Option B, implementing an Activity History pattern to track an organization\\'s activity measures, would provide a way for users to see the recent actions performed by themselves or others in the application, but it would not show summary or detailed information about report data. Option D, including a breadcrumbs pattern on applicable interfaces to show the organizational hierarchy, would provide a way for users to see where they are in the application, and easily go back to any previous level by clicking on the corresponding link, but it would not show summary or detailed information about report data.