Pegasystems

PEGACPSA23V1 Exam

Certified Pega System Architect
Questions & Answers
Demo

Version: 5.1

Question: 1

A door manufacturer offers a finite list of colors on all its doors. As part of the order, customers can select the color of the door. Select the data page definition configuration settings to source a color drop-down list to minimize memory usage.
Object Type » ACME-Products-Data-Color or ACME-Products-Work-Color or ACME-Products-Work-ColorFeedback
Edit Mode » Readonly Or Editable Or Savable Scope » Thread Or Requestor Or Node

A. Page, ACME-Products-Data-Color,ReadOnly,Node
B. List, ACME-Products-Data-Color,ReadOnly,Thread
C. Page, ACME-Products-Data-Color,ReadOnly,Requestor
D. List, ACME-Products-Work-Color,ReadOnly,Node
E. List, ACME-Products-Work-ColorFeedback,ReadOnly,Node
F. List, ACME-Products-Data-Color,ReadOnly,Node

Answer: F

Explanation:

In Pega, when configuring data pages to source UI controls such as drop-down lists, it's crucial to select the configuration that optimally uses memory and efficiently serves the data to the users. For this scenario, the correct answer is Option F: List, ACME-Products-Data-Color, ReadOnly, Node.
List vs. Page: Choosing 'List' is appropriate because the requirement is to populate a drop-down list with multiple entries (colors in this case). 'Page' would be used for single record data, which is not the case here.
Object Type: 'ACME-Products-Data-Color' is chosen because the data represents a list of color options provided by the manufacturer, which is typically static reference data, fitting the 'Data' class layer rather than the 'Work' layer which is used for case management or the 'Work-ColorFeedback' which would be used for storing feedback or interactions specific to colors.
Edit Mode: 'ReadOnly' is the correct mode since users are selecting a color from a pre-defined list and are not expected to edit these values directly.
Scope: 'Node' is selected to minimize memory usage. By setting the scope to 'Node', the data page is loaded once per node (server instance) and shared across all requestors (users) on that node, which is memory efficient for data that doesn't change often and is applicable across all users, such as a list of colors.
This configuration ensures that the data page is loaded efficiently in a read-only list format at the node level, reducing the overall memory footprint while making the color selections available across the application.
Reference: Pega Academy's Data Pages module, which covers data page types, scopes, and best practices for optimizing application performance and memory usage.

Question: 2

A Declare expression evaluates a circumstanced decision tree. The decision tree evaluates a property set by a data transform. What two steps do you perform to ensure that the decision tree is configured properly? (Choose 2)

A. Test the declare expression to verify that the rule is configured correctly
B. Configure a test page with data to satisfy the circumstancing condition
C. Specify the value of the circumstancing property when prompted in the run rule dialog
D. Test the data transform to verity the result is correct

Answer: A, B

Explanation:

When configuring a Declare Expression that evaluates a circumstanced decision tree, ensuring accuracy and expected behavior is crucial. The steps necessary involve:
A . Test the declare expression: This ensures that the Declare Expression, which automatically sets the value of a property based on changes in other properties, works as expected. Testing confirms the logic encapsulated in the decision tree is correctly evaluated and applied by the Declare Expression.
B . Configure a test page with data to satisfy the circumstancing condition: This involves setting up a scenario that meets the specific circumstances under which the decision tree's logic is evaluated. Circumstancing allows different versions of a rule to be applied based on specific conditions. By creating a test page that mimics these conditions, you ensure that the circumstanced version of the decision tree is invoked, allowing for accurate testing.
Options C and D, while important in broader testing contexts, are less directly involved in the specific task of ensuring the correct configuration of a circumstanced decision tree via a Declare Expression.
Reference: Pega Academy's modules on Declare Expressions and Decision Trees, which detail the process for configuring, circumstancing, and testing rules to ensure they meet application logic requirements.

Question: 3

A bookseller maintains a database of more than 10,000 book titles. You have been asked to configure a form that a allow users to select a book by title.
How do you configure the form?

A. Add a drop-down control arid source the data from the property value used in the control.
B. Add a drop-down control and Source the data using a data page.
C. Add an autocomplete control and Source the data using a data page.
D. Add an autocomplete control and source the data from the clipboard work page.

Answer: C

Explanation:

In scenarios involving a large dataset, such as a database of over 10,000 book titles, the most efficient way to allow users to find and select an item is by using an autocomplete control. This approach enhances user experience by filtering options based on user input, making it easier to navigate through large lists of data.
Why not a drop-down? A drop-down control (Options A and B) would not be practical for such a large number of items because it would require loading all items at once, which can lead to performance issues and a poor user experience.
Why an autocomplete control? The autocomplete control (Option C) is designed for scenarios like this. It allows the user to begin typing the name of a book, and the control suggests matching titles based on the input, narrowing down the options dynamically. This approach is much more user-friendly and efficient for large datasets.
Why sourced from a data page? Sourcing the autocomplete control from a data page (Option C) is the best practice because it allows for the efficient loading and caching of data. Data pages can be configured to refresh at appropriate intervals or under specific conditions, ensuring that the data is up-to-date. Furthermore, using a data page abstracts the data source from the control, making the application easier to maintain and update.
This setup ensures a smooth and efficient user experience, allowing for quick and easy selection from a vast database of book titles

Question: 4

A business architect has developed a new process for a case type. To verify that the UI elements collect the expected result, you want to test the process and the fields.
Which two configurations, when used together, allow you to record a set of interactions and save the test result to verify process functionality? (Choose two.)

A. Create a unit test for the case type.
B. Create a scenario test for the case type.
C. Add explicit assertions on the UI elements.
D. Add explicit assertions on the Scenario testing landing page.

Answer: B, C

Explanation:

To effectively test a new process for a case type and verify that the UI elements collect the expected results, utilizing scenario testing in Pega is highly recommended. This approach allows for recording a set of interactions and saving the test result, which is crucial for verifying process functionality.
B . Create a scenario test for the case type: Scenario testing in Pega allows for the simulation of an end-to-end process, including interactions with UI elements. This tool enables the creation and execution of tests that mimic real-world user actions, ensuring that the entire process functions as intended. By creating a scenario test, you can capture and automate the testing of specific pathways through a case type, making it an essential tool for validation.
C . Add explicit assertions on the UI elements: Assertions are conditions that you expect to be true at a certain point in your test. By adding explicit assertions to the UI elements during scenario testing, you can specify the expected outcomes or states of those elements. This ensures that not only does the process flow correctly, but the UI elements also behave and capture data as intended. Assertions provide a powerful way to validate the functionality and user interface of your application precisely.

Together, these configurations enable a robust testing framework that can simulate user interactions, verify UI behaviors, and ensure the overall functionality of the process and its fields.
Reference: Pega's official documentation on scenario testing and the use of assertions within these tests provides guidance on how to effectively utilize these features for comprehensive testing and verification.

Question: 5

A business architect has developed a new process for a case type. To verify that the UI elements collect the expected results, you want to test the process and the fields. Which two configurations, when used together, allow you to record a set of interactions and save the test results to verify process functionality? (Choose Two)

A. Add explicit assertions on the UI elements
B. Add validations on the UI elements
C. Create a unit test for the case type
D. Create a scenario test for the case type

Answer: A, D

Explanation:

For verifying the functionality of a new process and its associated UI elements, the combination of creating scenario tests and adding explicit assertions is most effective.
A . Add explicit assertions on the UI elements: Assertions are used in testing to verify that a UI element is in the expected state or contains the expected value after certain interactions have occurred. They are crucial for ensuring that the UI behaves as intended, making them an integral part of thorough testing strategies.
D . Create a scenario test for the case type: Scenario tests allow you to record a sequence of interactions within your application to simulate real-world usage. Creating a scenario test for a case type enables the recording and playback of interactions, including navigating through the process and inputting data into fields, to verify that the entire case behaves as expected from start to finish.
Both options, when used together, provide a powerful approach to testing by allowing you to define and verify the expected behavior of UI elements within the context of the complete process flow of a case type.
Reference: Pega Academy materials on scenario testing, which include instructions for creating scenario tests and adding assertions to ensure the accuracy of both the process flow and the UI element interactions.

Thank You For Trying PEGACPSA23V1 PDF Demo

To try our PEGACPSA23V1 Premium Files visit link below:

https://examsland.com/latest-exam-questions/PEGACPSA23V1/

Start Your PEGACPSA23V1 Preparation

Use Coupon EL25 for extra 25% discount on the purchase of Practice Test Software.