Free Questions for ACD101 by braindumpscollection

Shared by Watts on 24-05-2024

For More Free Questions and Preparation Resources

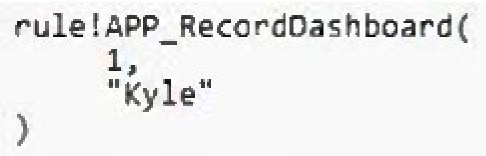
Check the Links on Last Page

Question 1

Question Type: **MultipleChoice**

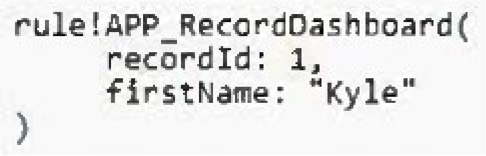
Which code snippet calls the interface APP_RecordDashboard while following best practices for passing in values for "recordId" and "firstName"?

A)



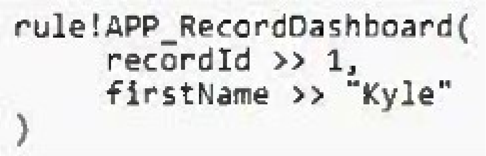
```
rule!APP_RecordDashboard(
      1,
      "Kyle"
)
```

B)



```
rule!APP_RecordDashboard(
      recordId: 1,
      firstName: "Kyle"
)
```

C)



```
rule!APP_RecordDashboard(
      recordId >> 1,
      firstName >> "Kyle"
)
```

Options:

A- Option A

B- Option B

C- Option C

Answer:

B

Explanation:

The best practice in Appian for passing values into an interface is to use named parameters, which is demonstrated by Option B. Named parameters make the code more readable and maintainable by clearly specifying which parameter each value is being passed to. In this case, the recordId and firstName parameters are clearly being assigned the values 1 and 'Kyle' respectively. Reference:

Appian Documentation: Passing Parameters to Interfaces

Question 2

Question Type: **MultipleChoice**

You are developing an expression rule. You need to find information on employing an Appian function that you have not used before.

For more information on the Appian function, what should you do first?

Options:

A- Look up the function in the Appian Documentation.

B- Search the Appian Knowledge Base Articles.

C- Post a question on Appian Community.

Answer:

A

Explanation:

When you need information on using a specific Appian function that you have not used before, the first step should be to consult the Appian Documentation. The documentation provides comprehensive details on each function, including syntax, parameters, usage examples, and best practices, which is essential for understanding how to correctly employ the function in an expression. Reference: Appian Documentation - Functions

Question 3

Question Type: MultipleChoice

A user needs to navigate from a record summary to an external URL.

Which interface component can be used to support this goal?

Options:

A- Button

B- Record Link

C- Card Layout with a link

Answer:

A

Explanation:

In Appian, to navigate from a record summary to an external URL, you can use a Button component configured with a 'Link' action. This approach allows you to define a URL that the button will navigate to when clicked. The Button component offers flexibility in terms of design and functionality, making it suitable for such navigation purposes within Appian interfaces. You can specify the URL directly in the Button's properties, allowing for dynamic link generation based on record data if needed. Reference:

Appian Documentation: Designing Interfaces (This section provides comprehensive details on using various interface components, including buttons, to achieve specific user interaction goals within Appian.)

Appian Documentation: Buttons (This page specifically focuses on the Button component, detailing its properties, usage, and how to configure it for different actions, including navigating to an external URL.)

Question 4

Question Type: MultipleChoice

You are creating a form used to order a pizz

a. You use a radio button component for the selection.

The pizza selection labels include a list of toppings. You do not want the selection labels to be truncated.

Which layout should you choose?

Options:

A- Compact

B- Grid

C- Stacked

Answer:

C

Explanation:

For a pizza ordering form where you do not want the radio button selection labels to be truncated, the Stacked layout is the most appropriate. This layout will list the options vertically, giving each one adequate space and preventing truncation, which is particularly useful when the labels include longer text, such as a list of toppings. Reference: Appian Documentation - Interface Components

Question 5

Question Type: MultipleChoice

How can you prevent users from accessing Tempo?

Options:

A- Remove the users from the Tempo Users system group.

B- Change the default User Start Page.

C- Ensure the user is in the Application Users group, which by default does not have access to Tempo.

Answer:

A

Explanation:

To prevent users from accessing Tempo, you should remove them from the Tempo Users system group. This group controls access to the Tempo interface in Appian. By removing users from this group, they will no longer have the necessary permissions to access Tempo features and content. Reference: Appian Documentation - System Groups and User Access

Question 6

Question Type: **MultipleChoice**

When applying a default filter to a record type, what is a true statement for excluded data?

Options:

A- The data does not show up in the record list. End-users are unable to clear the filter condition to view the data.

B- The data does not show up in the record list, but can be accessed by end-users using a direct link to a record view.

C- The data does not show up in the record list, but end-users can clear the filter condition to view the data.

Answer:

C

Explanation:

When a default filter is applied to a record type in Appian, the data excluded by that filter does not appear in the default record list. However, end-users with appropriate permissions can modify or clear the filter conditions to view the excluded data. This behavior allows for a flexible and user-driven data exploration experience within Appian record lists. Reference: Appian Documentation - Record Lists and Filters

Question 7

Question Type: **MultipleChoice**

You are creating a new interface object to display a pie chart.

The data for the chart is stored in a local variable in the parent interface object which references your child interface.

In terms of performance, what is the most efficient method to access the data required for the pie chart?

Options:

A- Create a rule input on the child interface and pass the local variable data from the parent interface.

B- Reference the local variable directly from the child Interface using a process report.

C- Query the data separately in the child interface to avoid passing it from the parent interface.

Answer:

A

Explanation:

The most efficient method to access data for a pie chart in a child interface is to create a rule input on the child interface and pass the local variable data from the parent interface. This method avoids redundant data queries and takes advantage of Appian's pass-by-reference mechanism, which does not duplicate data in memory when passing it to the child interface. Reference: Appian Documentation - Interface Design