Free Questions for ACD101

Shared by Good on 04-10-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Review the following expression rule:

```
wherecontains(
    ri!name,
    index(ri!directory, "name")
)
```

ri!name is defined as "Maria".

ri!directory is defined as the following:

```
{
  a!map(
    name: "Maria",
    employeeNumber: 100
  ),
  a!map(
    name: "Steven",
    employeeNumber: 101
  )
}
```

What is the expected output?

Options:

A- Maria

B- 0

C- 1

Answer:

C

Explanation:

Given that ri!name is defined as 'Maria' and ri!directory contains two a!map() structures, one of which includes the name 'Maria,' the expression wherecontains(ri!name, index(ri!directory, 'name')) will evaluate as follows: The index() function will return a list of values from ri!directory for the key 'name,' which will be {'Maria', 'Steven'}. The wherecontains() function will then check where 'Maria' is found within this list. Since 'Maria' is the first element, the function will return a list of indices where 'Maria' is found, in this case, {1}. Appian lists are 1-indexed, so the first position is represented by 1, not 0. Reference: Appian Expression Language Documentation - Functions

Question 2

Question Type: **MultipleChoice**

Which step can be critical in passing information from a form back to a process model?

Options:

A- Configure the Data Management tab.

B- Configure the activity class parameters of a Write to Data Store Entity node, a

C- Configure inputs on the Data tab of a User Input Task.

Answer:

C

Explanation:

The critical step in passing information from a form back to a process model is to configure inputs on the Data tab of a User Input Task. When you create a User Input Task, it includes a form for users to interact with. The data entered into this form can be mapped to process variables via the Data tab configuration. This ensures that the information collected in the form is available to the process for further use. Reference: Appian Documentation - User Input Tasks

Question 3

Your customer wants to change the name of a field of an existing Custom Data Type (CDT) to match a renamed database field.

The CDT is backed by a database entity, whose data store has the Automatically Update Database Schema option disabled. The old column name was BIRTHDATE and the new column name is DATE_OF_BIRTH.

How should you proceed?

Options:

A- Download the CDT as XSD, make the appropriate changes, and re-upload the XSD. Verify and publish the data store.

B- Rename the field in the record type in Appian to automatically update the CDT field.

C- Rename the field in the CDT in Appian. Verify and publish the data store.

Answer:

C

Explanation:

When a field name in an existing Custom Data Type (CDT) needs to be changed to match a renamed database field, and the Automatically Update Database Schema option is disabled, the correct approach is to rename the field in the CDT within Appian. After renaming the field in the CDT to match the new database column name (from BIRTHDATE to DATE_OF_BIRTH in this case), you

should verify the changes and publish the data store to reflect the updates. This approach ensures that the Appian data model remains in sync with the underlying database schema. Reference: Appian Documentation - Data Types and Data Stores

Question 4

Question Type: MultipleChoice

You created and published a new process model.

The process model has a start form with two synchronous subprocesses with 40 and 66 nodes each. All nodes are chained from the start node through the subprocesses to the end node. After the tasks and subprocesses, there is a second User Input Task in which the user can confirm the entries and add a comment.

When testing as a normal Acme business user, you see that the confirmation screen is not shown to you.

What might be the reason for this behavior?

Options:

A- The maximum number of activity chained nodes is exceeded and breaks.

B- The second User Input Task is assigned to the process initiator.

C- The second User Input Task is assigned to the Acme business user group.

Answer:

A

Explanation:

In Appian, there is a limitation on the number of activities that can be chained in a process, known as the 'chaining limit.' If a process model exceeds this limit, which includes synchronous subprocesses and their nodes, the process may break or not behave as expected. In this scenario, with two large subprocesses chained from start to end, the maximum number of activity chained nodes could be exceeded, resulting in the confirmation screen not being shown. Adjusting the process model to reduce chaining or using asynchronous patterns where possible can help mitigate this issue. Reference: Appian Documentation - Process Model Best Practices

Question 5

Question Type: MultipleChoice

Where can an Appian Developer connect with and share their expertise with other Appian Developers?

A- Appian Learning Paths via Appian Academy

B- Appian Knowledge Base

C- Appian Community discussions

Answer:

C

Explanation:

Appian Community discussions provide a platform for Appian Developers to connect with, share expertise, and learn from each other. The community is a vibrant space where developers can ask questions, share solutions, and discuss best practices related to Appian development. While Appian Learning Paths via Appian Academy and Appian Knowledge Base are valuable resources for learning and troubleshooting, the Community discussions specifically facilitate peer-to-peer interaction and knowledge sharing among developers. Reference: Appian Community Website

Question 6

Question Type: **MultipleChoice**

You are creating a new customer onboarding application. Documents are required from customers for verification and onboarding purposes. You need to store these documents within Appian.

Which two areas in Appian should you configure? (Choose two.)

Options:

A- Knowledge Center

B- Decision Object

C- Folder

D- Feed

Answer:

A, C

Explanation:

In Appian, to store documents such as those required for customer onboarding, you should configure a Knowledge Center and within that, Folders. The Knowledge Center in Appian serves as the top-level container for storing and organizing documents and folders. Folders within a Knowledge Center provide a way to categorize and manage access to documents, making them an essential part of document management in Appian applications. Decision Objects and Feeds are not related to document storage. Reference: Appian Documentation - Knowledge Centers and Folders

Question 7

You have a record type, ABC_Author, backed by a database table.

You need to retrieve the total number of authors without loading all the data.

According to Appian best practices, which code snippet accomplishes this goal in the most efficient way?

A)

```
a!queryRecordType(
    recordType: recordType!ABC Author,
    pagingInfo: a!pagingInfo(
        startIndex: 1,
        batchSize: 1
    ),
    fetchTotalCount: true
)
```

B)

```
a!queryRecordType(
  recordType: recordType!ABC Author,
  fields: {
    a!aggregationFields(
      measures: {
        a!measure(
          field: recordType!ABC Author.fields.id,
          function: "COUNT",
          alias: "count"
        )
      }
    )
  },
  pagingInfo: a!pagingInfo(
    startIndex: 1,
    batchSize: 1,
  )
)
```

C)

```
count(
  a!queryRecordType(
    recordType: recordType!ABC Author,
    fields: {
      recordType!ABC Author.fields.id
    },
    pagingInfo: a!pagingInfo(
      startIndex: 1,
      batchSize: 5000,
    ),
  ).data
)
```

Options:

A- Option A

B- Option B

C- Option C

Answer:

B

Explanation:

According to Appian best practices, the most efficient way to retrieve the total number of authors without loading all the data is to use a query that includes an aggregation function. Option B uses an aggregation field with a COUNT function, which is designed to return the number of rows that match a query without the need to retrieve and load all row data, thus optimizing performance.

Appian Documentation: a!queryRecordType Function

Question 8

Question Type: **Hotspot**

Match each node to the correct description for the node.

Note: Each description will be used once. To change your responses, you may deselect your response by clicking the blank space at the top of the selection list.

Answer Area

Timer Event

Can accept multiple paths: when all paths arrive, all outgoing flows are executed at the same time.
Can be added to a flow or attached to a smart service to trigger an Exception Flow or an Escalation.
Can run automated activities.
Can accept one incoming path and select one outgoing path.

XOR Gateway

Can accept multiple paths: when all paths arrive, all outgoing flows are executed at the same time.
Can be added to a flow or attached to a smart service to trigger an Exception Flow or an Escalation.
Can run automated activities.
Can accept one incoming path and select one outgoing path.

Script Task

Can accept multiple paths: when all paths arrive, all outgoing flows are executed at the same time.
Can be added to a flow or attached to a smart service to trigger an Exception Flow or an Escalation.
Can run automated activities.
Can accept one incoming path and select one outgoing path.

AND Gateway

Can accept multiple paths: when all paths arrive, all outgoing flows are executed at the same time.
Can be added to a flow or attached to a smart service to trigger an Exception Flow or an Escalation.
Can run automated activities.
Can accept one incoming path and select one outgoing path.

Answer:

Explanation:

Appian Documentation: Process Modeler Objects