



Free Questions for UiPath-ADPv1 by certsdeals

Shared by Soto on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which of the following statements accurately describes the primary components of a State Machine in a workflow?

Options:

- A-** A State Machine is composed of States, Triggers, and Loops.
- B-** A State Machine is made up of States, Transitions, Entry Actions, and Exit Actions.
- C-** A State Machine comprises three main components: Sequences, Flowcharts, and Input/Output Arguments.
- D-** A State Machine consists of four primary components: States, Triggers, Transitions, and Actions.

Answer:

D

Explanation:

A State Machine is a type of automation that uses a finite number of states in its execution. It can go into a state when it is triggered by an activity, and it exits that state when another activity is triggered. A State Machine consists of four primary components:

* **States:** These are the nodes that represent the possible situations or conditions of the automation. Each state can have one or more activities inside it that define the behavior of the automation in that state. There are two types of states: State and Final State. A State Machine must have one initial state and one or more final states.

* **Triggers:** These are the events or conditions that cause the automation to enter a state. A trigger can be an activity, a variable, or an argument that is executed or evaluated before entering a state. For example, a Click activity can be a trigger for entering a state that performs some actions after clicking a button.

* **Transitions:** These are the arrows or branches that connect the states and define the flow of the automation. A transition can have a condition that determines whether the automation can move from one state to another. For example, a Boolean expression can be a condition for a transition that checks if a certain condition is true or false before moving to the next state.

* **Actions:** These are the activities or sequences that are executed when the automation moves from one state to another. An action can be performed before or after the transition, depending on the type of the transition. For example, a Log Message activity can be an action that logs a message when the automation exits a state.

You can find out more about state machines on the [Studio - State Machines - UiPath Documentation Portal](#) or in the official [Workflow Foundation](#) documentation.

Question 2

Question Type: MultipleChoice

What is the role of connections in the UiPath Integration Service?

Options:

- A-** Connections establish tasks and exchanges between users and external applications using the authentication process of the UI automation provider.
- B-** Connections establish tasks and exchanges between users and external applications using the authentication process of the API provider.
- C-** Connections establish tasks and exchanges based on a connector's compatibility with the external application.
- D-** Connections establish tasks and exchanges between users and external applications using the server-side triggers.

Answer:

B

Explanation:

The UiPath Integration Service is a feature of Automation Cloud that allows you to connect to various external applications and systems using predefined connectors. A connector is a component that provides the logic and interface for interacting with a specific application or system. A connection is an instance of a connector that is configured with the credentials and parameters required to access the application or system.

The role of connections in the UiPath Integration Service is to establish tasks and exchanges between users and external applications using the authentication process of the API provider. An API provider is a service that exposes an application programming interface (API) for accessing and manipulating data and functionality of an application or system. For example, Microsoft Outlook 365, Salesforce, and Google Sheets are some of the API providers that have connectors in the UiPath Integration Service.

To create a connection, you need to select a connector and provide the authentication details for the API provider. The authentication process may vary depending on the type and security level of the API provider. For example, some API providers may use OAuth 2.0, which is a standard protocol for authorizing access to resources on behalf of a user. Other API providers may use API keys, which are unique identifiers that are used to authenticate requests to the API.

Once you create a connection, you can use it to perform various tasks and exchanges with the external application or system. For example, you can use the Microsoft Outlook 365 connector to send and receive emails, create and update calendar events, and manage contacts. You can use the Salesforce connector to create and update records, execute queries, and invoke actions. You can use the Google Sheets connector to read and write data, format cells, and manage worksheets.

You can use the connections in your automation workflows in UiPath Studio, UiPath Apps, UiPath Assistant, and UiPath Test Suite. You can also manage and share your connections in UiPath Orchestrator, using the folder structure and permission system.

Question 3

Question Type: MultipleChoice

When creating a new test case, which option restricts dynamic data update in UiPath Orchestrator once the test case is published?










New Test Case

Test Data

Test Data

Configure a data source that can be imported through arguments and used when running the Test Case.

Source

-  None
-  None
-  Existing Data
-  File
-  Data Service
-  Test Data Queue
-  Auto Generate

Options:

- A- Test Data Queue
- B- Data Service with Filter
- C- File
- D- Data Service

Answer:

C

Explanation:

When creating a new test case in UiPath Studio, you can configure a data source that can be imported through arguments and used when running the test case. You can choose from different options for the data source, such as File, Data Service, Test Data Queue, and Auto Generate. Each option has different implications for how the test data is updated in UiPath Orchestrator once the test case is published.

The option that restricts dynamic data update in UiPath Orchestrator once the test case is published is File. This option allows you to use Excel and CSV files as data sources for your test cases. When you import a file as a data source, a JSON file is created in the Test Data folder of your project, which contains the test data values. When you publish the test case, the JSON file is also published and stored in Orchestrator. The test data values are not updated dynamically in Orchestrator, unless you manually update the test data by choosing the Update Test Data option in Studio or Test Explorer. This option is useful when you want to use static or predefined test data that does not change frequently.

The other options do not restrict dynamic data update in UiPath Orchestrator once the test case is published. Option A, Test Data Queue, allows you to use a queue in Orchestrator as a data source for your test cases. You need to configure a JSON schema and add a test data queue in Orchestrator before using this option. The test data values are updated dynamically in Orchestrator, as the queue items are processed by the test case. This option is useful when you want to use queue items as test data and track their status and progress. Option B, Data Service with Filter, allows you to use Data Service entities as data sources for your test cases. Data Service is a feature of Automation Cloud that lets you create and manage data entities and relationships. You need to select an entity and apply a filter using the Query Builder before using this option. The test data values are updated dynamically in Orchestrator, as the entity records are modified by the test case or other processes. This option is useful when you want to use data entities as test data and perform CRUD operations on them. Option D, Auto Generate, allows you to use test data that is automatically generated by UiPath Studio to cover as many paths as possible during execution. You need to select a test data generation strategy and a number of test data sets before using this option. The test data values are updated dynamically in Orchestrator, as the test case is executed with different data sets. This option is useful when you want to use random or diverse test data that can increase the test coverage and quality.

Question 4

Question Type: MultipleChoice

A developer has defined a variable named "CurrentDate" of type Date Time. Which of the following expressions can be used to show what hour was 12 hours prior to the date

from the variable?

Options:

- A- `DateTime.AddHours(CurrentDate, -12).Hour`
- B- `CurrentDate.AddHours(-12).Hour`
- C- `CurrentDate.SubtractHours(12).Hour`
- D- `DateTime.SubtractHours(CurrentDate, 12).Hour`

Answer:

B

Explanation:

In UiPath Studio, you can use `DateTime` and `TimeSpan` variables to store and manipulate date and time values. A `DateTime` variable represents a specific point in time, such as the current date and time, or a date in the past or future. A `TimeSpan` variable represents a time interval, such as one hour, two days, or three weeks.

To show what hour was 12 hours prior to the date from the variable, you can use the `AddHours` method on the `DateTime` variable and pass a negative value. For example, if you have a `DateTime` variable called `CurrentDate` that stores a date and time value, you can write:

```
CurrentDate.AddHours(-12).Hour
```

This expression returns an integer value that is the hour component of the `DateTime` value that is 12 hours before the `CurrentDate` value. For example, if `CurrentDate` is `2024-02-08 10:51:49`, then the expression returns `22`.

The other options are not correct expressions for showing what hour was 12 hours prior to the date from the variable. Option A is invalid, because the DateTime class does not have an AddHours method. Option C is incorrect, because the DateTime variable does not have a SubtractHours method. Option D is incorrect, because the DateTime class does not have a SubtractHours method.

Question 5

Question Type: MultipleChoice

What status needs to be selected in the Set Transaction Status activity in UiPath Studio so that the respective Queue Item will be retried?

Options:

- A- Retried
- B- Failed
- C- Abandoned
- D- Postponed

Answer:

B

Explanation:

The Set Transaction Status activity in UiPath Studio is used to set the status of an Orchestrator queue item to Failed or Successful. This activity is usually placed at the end of the Process Transaction state in the REFramework template, after the Try Catch block. The status of the queue item determines whether it will be retried or not.

To retry a queue item, the status needs to be set to Failed. This means that the transaction was not completed successfully, and it needs to be processed again. The number of retries for a queue item depends on the MaxRetryNumber parameter in the Config file of the REFramework. By default, it is set to 2, which means that each queue item can be retried twice before it is marked as permanently failed.

The other options are not valid statuses for the Set Transaction Status activity. Option A is incorrect, because Retried is not a status, but a property of a queue item that indicates whether it has been retried or not. Option C is incorrect, because Abandoned is a status that is automatically assigned by Orchestrator to a queue item that has been in progress for more than 24 hours, and it cannot be retried. Option D is incorrect, because Postponed is a status that is automatically assigned by Orchestrator to a queue item that has a specific start date in the future, and it cannot be retried until that date is reached.

Question 6

Question Type: MultipleChoice

A developer needs to use the REFramework to automate a business process that involves processing transactions within an Excel table. Each transaction in the table should follow the same steps for processing and queues cannot be used as there is no Orchestrator in the

environment.

Which variable type is best suited for TransactionItem in this scenario?

Options:

- A- UiPath.Core.QueueItem
- B- System.Data.DataRow[]
- C- System.Data.DataTable
- D- System.Data.DataRow

Answer:

D

Explanation:

The REFramework (Robotic Enterprise Framework) is a template that provides a standard structure and best practices for developing UiPath automation projects. It uses the concept of transactions to process items from a data source, such as an Excel table, a queue, or a database. A transaction is a single unit of work that can be executed by the robot. The REFramework has a variable called TransactionItem, which stores the current transaction to be processed.

The variable type of TransactionItem depends on the type of data source that is used for the transactions. If the data source is an Orchestrator queue, then the TransactionItem should be of type UiPath.Core.QueueItem, which represents an item from the queue. If the data source is an Excel table, then the TransactionItem should be of type System.Data.DataRow, which represents a row from the table. A DataRow object contains the values of a single row in a DataTable object, which represents the entire table.

In this scenario, the developer needs to use the REFramework to automate a business process that involves processing transactions within an Excel table. Each transaction in the table should follow the same steps for processing and queues cannot be used as there is no Orchestrator in the environment. Therefore, the best variable type for TransactionItem in this scenario is System.Data.DataRow, as it can store a single row from the Excel table as a transaction.

The other options are not suitable variable types for TransactionItem in this scenario. Option A is incorrect, because UiPath.Core.QueueItem is used for queue items, not Excel rows. Option B is incorrect, because System.Data.DataRow[] is an array of DataRow objects, not a single DataRow object. Option C is incorrect, because System.Data.DataTable is a table of DataRow objects, not a single DataRow object.

Question 7

Question Type: MultipleChoice

In UiPath Studio, how can a specific amount of time be subtracted from the current date using DateTime and TimeSpan variables in the VB.NET?

Options:

- A- Multiply the TimeSpan variable by -1 and then add it to the DateTime variable.
- B- Assign the subtraction result to a DateTime variable with the following syntax Today.Subtract(Span).
- C- Use the Add method on the DateTime variable and pass a negative TimeSpan value.
- D- Call Now.Subtract(Span) directly, without using a DateTime variable.

Answer:

C

Explanation:

In UiPath Studio, you can use DateTime and TimeSpan variables to store and manipulate date and time values. A DateTime variable represents a specific point in time, such as the current date and time, or a date in the past or future. A TimeSpan variable represents a time interval, such as one hour, two days, or three weeks.

To subtract a specific amount of time from the current date using DateTime and TimeSpan variables in VB.NET, you can use the Add method on the DateTime variable and pass a negative TimeSpan value. For example, if you have a DateTime variable called Today that stores the current date and time, and a TimeSpan variable called Span that stores the time interval you want to subtract, you can write:

```
Today.Add(-Span)
```

This expression returns a new `DateTime` value that is the result of subtracting the `Span` value from the `Today` value. For example, if `Today` is `2024-02-08 10:51:49` and `Span` is `1.02:10:04`, then the expression returns `2024-02-07 08:41:45`.

The other options are not correct ways of subtracting a `TimeSpan` value from a `DateTime` value in VB.NET. Option A is invalid, because you cannot multiply a `TimeSpan` value by a number. Option B is incorrect, because the `Subtract` method on the `DateTime` variable returns a `TimeSpan` value, not a `DateTime` value. Option D is incorrect, because the `Now` property is not a method, and it cannot take a `TimeSpan` value as an argument.

To Get Premium Files for UiPath-ADPv1 Visit

<https://www.p2pexams.com/products/ui-path-adpv1>

For More Free Questions Visit

<https://www.p2pexams.com/ui-path/pdf/ui-path-adpv1>

