TESTCASE THREE - DISCOUNT CODE

TESTCASE THREE

Exercise 8a - Rescan, Value Range and Module Merge

Objective

By the end of this exercise, you will be able to use XScan to rescan a Module and then to merge two Modules to avoid duplication.

Why is this important?

Modules can be amended; they do not have to be recreated every time you wish to change them. Modules can be reused any number of times and are not specific to the TestStep. Modules are technical representations of the SUT in Tosca, so we should not have multiple Modules for the same controls. Module Merge helps solve this duplication problem.

Instructions

1.	In the **Web Shop**, add items and navigate to the "**Shopping cart**" page; apply a discount code named: • **AutomationDiscount2** Leave the page open as we will be scanning it momentarily.	☐
2.	Navigate to **Webshop>>Check out process>>8a Rescan Shopping Cart** in the Modules section.	☐
3.	Right click and select Rescan the "**Rescan Cart**" Module which is in the **8a Rescan Shopping Cart** Folder.	☐
4.	Ensure the following **controls** are selected, in addition to those which are already in the Module:	☐

Element	Type	Rename to
Update shopping cart	Button	
discountcouponcode	Textbox	Enter discount code
Apply coupon	Button	
Coupon applied message	DIV	Coupon applied message
giftcardcouponcode	Textbox	Gift card coupon code
Shopping cart cost total	Table	Shopping cart costs

5.	Add a value range of possible codes to the discount code in the Module you just rescanned. The range should include the following values: • **AutomationDiscount2** • **PercentageTotal** • **PercentageShipping** • **FlatTotal** Make sure to separate the values by semicolons.	☐
6.	Expand all the Folders in the **Check out process** Module Folder. You will notice that there are currently two **Shopping Cart** Modules (one called "**Shopping Cart**" and one called "**Rescan Cart**" in the **8a Rescan Cart** Module Folder). Use "**Ctrl+Click**" to select both Modules simultaneously.	☐
7.	In the top ribbon, on the "**Modules**" tab, select **Merge Modules**.	☐
8.	The **Target Module** is the one we will keep, and we need to select the **Attributes** we want to bring from the Source Module into the Target Module. Ensure that the **Rescan Cart** Module (the one we just scanned) is listed as the **Target Module**. If it is not, choose the option to **Switch Modules**.	☐

9.	Because of their identical properties, Tosca should automatically link the Attributes which exist in both Modules.	☐
10.	Select the **Target Module** and **Source Module**, then select the Module Merge option **Enable**. When you are ready, select "**Merge**". Tosca will let you know how many Attributes have been updated, which usages have been relinked to the new Module, and deletes the **Source Module**. Select "**Close**". Rename the Module from "**Rescan Cart**" back to "**Shopping Cart**" and save your work.	☐

Hints

- To identify tables, use Highlight on Screen to locate the correct one.
- AutomationDiscount2 will apply a discount of 20% to the total value of the order.

Exercise 8b - Create the TestCase

Objective

By the end of this exercise, you will be able to use the Reusable TestStepBlocks to create References in a TestCase.

Why is this important?

The creation of logical and easy to follow structures within automated TestCases makes maintenance and problem solving easier.

Instructions

1.	On the TestCase Root Folder, create a new TestCaseFolder named "**TestCase 3**". Create a subfolder named "**8b Create the TestCase**".	☐
2.	Within the subfolder, create a new TestCase, name it "**Discount Code**". Add the Test configuration parameter: "**Browser**" with the value "**InternetExplorer**".	☐
3.	Add the following **Reusable TestStepBlocks** from the Library into the new TestCase: • Precondition • Order Product • Checkout Process • Confirmation • Verification of Success • Postcondition	☐
4.	Create two new **TestStepFolders** named: • Start Checkout • Verification of Prices Reorder the TestStepFolders to match the process flow of the **Web Shop**	☐
5.	Add the following Modules into the "**Start Checkout**" Folder: • "**Top Menu**" – rename the TestStep "**Navigate to Shopping Cart**" • "**Shopping Cart**" – rename the TestStep "**Shopping Cart Procedures**"	☐
6.	Add the following Module into the "**Verification of Prices**" Folder: • "**Confirm Order**" – rename the TestStep "**Verification of Prices**"	☐
7.	Enter the following **Values** for the **TestStepValues** as per the table below:	☐

TestStep	TestStepValue	Value	ActionMode
TestStepFolder ***Start Checkout***			
Navigate to Shopping Cart	Shopping cart	X	Input
Shopping Cart Procedures	Enter discount code	AutomationDiscount2	Input
	Apply coupon	X	Input
	Terms of Service	True	Input
	Checkout	X	Input
TestStepFolder ***Verification of Prices***			
Verification of Prices	Cart Total		Select
	Column: Rename to #2		Select
	Cell: Choose Sub-Total	{MATH[{B[PriceBlueJeans]}*25]}	Verify
	Cell: Choose Sub-Total	SubTotal	Buffer
	Cell: Choose Shipping:*	10.00	Verify
	Cell: Choose Discount:	-{MATH[({B[SubTotal]}+10.00)*.2]}	Verify
	Cell: Choose Discount:	Discount	Buffer
	Cell: Total	Use the SubTotal Buffer to verify the total cost adding any shipping / additional fees.	Verify

8. Run the TestCase in the **ScratchBook** ☐

Hints

» Multiply a whole number by 0.20 to calculate a 20% discount.

Exercise 8c - ActionMode WaitOn

Objective

By the end of this exercise, you will be able to create a dynamic wait on a TestStepValue using the ActionMode WaitOn.

Why is this important?

WaitOn instructs Tosca to wait for an element or action to occur before moving on to the next TestStep. This benefits the automation as it can stop possible TestCase failures in situations where the SUT does not react immediately e.g. waiting for a credit card to authorize.

Instructions

1.	Duplicate TestCaseFolder "**8b Create the TestCase**" and name it "**8c ActionMode WaitOn**".	☐
2.	Navigate to the TestStep "**Shopping Cart Procedures**" in the folder "**Start Checkout**".	☐
3.	Set the ActionMode **WaitOn** for the TestStepValue "**Coupon applied message**", instructing Tosca to wait for the message to appear before executing the next TestStep. *Remember: use the blue downward arrow to select: **Visible == True**	☐