Exercise 7a – Build Template & Link Values

Objective

By the end of this exercise, you will be able to build a Template from a more advanced TestCase and link it to a more complicated TestSheet.

Why is this important?

The Log in Process TestCase we have used until now is just a simple TestCase for you to get used to the concept of TestCaseDesign and Templates. In real-life project scenarios, you are more likely to work with a more complicated TestCase, such as the Discount Code TestCase.

Project perspective

You are to test a new function in the DemoWebShop: the discount codes. Most online web stores offer special deals and promotions for marketing purposes. One of the most common forms of promotion is to offer discount codes. You need to test that the valid discount codes give the appropriate discounts and invalid codes entered do not give any discount. In order to do so, the first steps are to build a Template out of the existing TestCase and then link the TestSheet to that Template.

Instructions

1	Within the Template folder, create a TestCase folder named "**7a Build Template**". Copy the TestCase: "**Discount Code**" from the "**Automation Specialist 1 TestCases**" folder and paste it into the new folder.	☐
2.	Convert the TestCase "**Discount Code**" into a Template.	☐
3.	Resolve the References for all folders except "**Precondition_Reference**" and "**Order Product_Reference**".	☐
4.	Add a new folder after "**Order Product_Reference**", named: **Verify Coupon Code**.	☐
5.	Add the Reusable TestStepBlock "**Empty Shopping Cart, Log out and Close Browser**" to the Template. Resolve the Reference and rename it to "**Empty Shopping Cart**".	☐
6.	Adapt the folder structure to match the following: • Precondition_Reference • Order Product_Reference • Verify Coupon Code • Checkout Process • Start Checkout • Verification of Prices • Confirmation • Verification of Success • Empty Shopping Cart • Postcondition	☐
7.	Check that the structure of the TestCase matches the one outlined below. Change the TestSteps in the TestStep folders: "Verify Coupon Code" and "Empty Shopping Cart" as shown.	☐

TestStep Sub-Folder	TestStep
TestStep Folder Precondition_Reference	
	Business Parameters
TestStep Folder **Order Product_Reference**	
	Navigate to Apparel and Shoes
	Navigate to Blue Jeans
	Order the BlueJeans
TestStep Folder **Verify Coupon Code**	
	Navigate to Shopping Cart (moved from TestStep folder "Start Checkout")
	"Shopping Cart Procedures" Rename "Shopping Cart Procedures" to "Enter Code" (moved from TestStep folder "Start Checkout", then delete the "Start Checkout" folder.)
TestStep Folder **Checkout Process**	
Checkout Process	Billing address - Continue
Shipping (sub-folder of Checkout Process)	Shipping address - Continue
Shipping (sub-folder of Checkout Process)	Shipping method - Choose Ground
Checkout Process	Payment method - Choose PM
Checkout Process	Payment Information - Credit card
TestStep Folder **Verification of Prices**	
sub-folder ***Recovery Scenarios***	
	Verification of Prices
TestStep Folder **Confirmation**	
	Confirm the Order
TestStep Folder **Verification of Success**	
	Verify the Order Success
TestStep Folder **Empty Shopping Cart**	
	Navigate to Cart
	WHILE Statement
	Delete the "**Log out**" and "**Close the Application**" TestSteps
TestStep Folder **Postcondition**	
	Log out
	Close the Application

Exercise 7b – Link Values

Objective

By the end of this exercise, you will be able to link the Values from the TestSheet to their respective TestStepValues.

Why is this important?

Without the Values from the TestSheet, the Template contains no test data.

Project perspective

There are different discount codes that give different types of discounts. One code applies only to the goods, another code applies to the shipping methods. Additionally, there is a code that applies a discount to the total price. There are also different methods of shipping, each has its own cost. When we verify the prices, we will need to keep the shipping cost and the discount in mind.

First, we calculate the base price the customer needs to pay. Then we calculate the shipping cost and the discount. Finally, we add the values together for a total price.

Instructions

1. Duplicate the TestCase folder named "**7a Build Template**" and rename it "**7b Link Values**". ☐
2. Link the **TestSheet** "**Calculate Discount**" in folder **7 Calculate Discount** in the TestCaseDesign section to the **TestCase Template** "**Discount Code**". ☐
3. Add the necessary Attributes from the TestSheet "Calculate Discount" to the TestCase. TestSheet Attributes are required in the following TestStepValues: ☐

TestStep	TestStepValue	Path in TestSheet	Value
Business Parameters:	Email	Calculate Discount >> Precondition >> User >> Value - Email	XL Link to "**Value** - **Email**" from the '**Calculate Discount**' TestSheet
	Password	Calculate Discount >> Precondition >> User >> Value - Password	XL Link to "**Value** - **Password**" from the TestSheet
Enter Code	Discount Code	Calculate Discount >> Workflow >> Value - Discount Code	Delete the existing value. XL Link to "**Value** - **Discount Code**" from the TestSheet
	Coupon Applied	Calculate Discount >> Verifications >> Value - Message	Verify that the InnerText of the message has the appropriate value as in the TestSheet (adjust ActionMode to **WaitOn** to wait for the text to appear)

4.. Right click on the TestStep "**Verification of Prices**" and select "**Allow Reorder**". With the "Allow Reorder" function, TestStepValues can be rearranged. ☐

5. For the table Cart total, select the 2nd column. Create the TestStepValues by selecting the appropriate option from the grey <Cell>. Then, reorder them to form the following structure:
   - Sub-Total:
   - Sub-Total:
   - Shipping:*
   - Shipping:*
   - Shipping:*
   - Discount:
   - Discount:
   - Discount:
   - Discount:
   - Total:
   - Total:

   ☐
6. Fill in the following data for the TestStepValues: ☐

TestStepValue	Path in TestSheet	Value
Sub-Total		Use {Math} function to calculate verify the base price of the order without applying shipping cost or discount by multiplying the buffered price of a BlueJeans to the quantity **{MATH[{B[PriceBlueJeans]}*{B[QuantityBlueJeans]}]}**
Sub-Total		Buffer the Sub-Total
Shipping:*		Verify 10.00
Shipping:*	Calculate Discount >> Verifications >> Value - Discount %	Use {Math} function to Verify the percentage discount from the TestSheet for the cost of the Shipping (10.00) **{MATH[(1-{XL[Verifications.Value - Discount %]})*10]}**
Shipping:*		Buffer the shipping costs
Discount:	Calculate Discount >> Verifications >> Value - Discount %	Use {MATH} function to Verify, the sum of the buffered SubTotal and the shipping cost, multiply by the discount percentage from the TestSheet, the percentage discount for the total cost of the order including shipping costs: **[({B[SubTotal]}+{B[Shipping]})*{XL[Verifications.Value - Discount %]}]}**
Discount:	Calculate Discount >> Verifications >> Value - Discount Flat Rate	Verify, from the TestSheet, the flat rate discount: **-{XL[Verifications.Value - Discount Flat Rate]}**
Discount:	Calculate Discount >> Verifications >> Value - Discount %	Verify the percentage discount on the sub-total from the TestSheet **-{MATH[{XL[Verifications.Value - Discount %]}*{B[SubTotal]}]}**
Discount:		Buffer the discount
Total:		Verify the total price (taking into account shipping costs and the discount) **{MATH[{B[SubTotal]}+{B[Shipping]}+{B[Discount]}]}**
Total:		Verify the total (taking into account only shipping costs) **{MATH[{B[SubTotal]}+{B[Shipping]}]}**

Hints

- Previous Values will need to be cleared before you drag and drop new values, as the XL will be added at the end of the current value.
- Be careful with the {MATH[]} function, as some of the TestStepValues (for the Discount values) have a negative sign, as the value of the TestStep should be negative.
- Verifications have Attributes in the TestSheet.
- Depending on the SUT, the DataType for {MATH} may need to be changed to “Numeric”.
- To change Tosca numeric formats, such as the decimal character separator, go to Settings >> TBox >> Number formats.

Exercise 7c – Set Conditions & Instantiation

Objective

By the end of this exercise, you will have added the necessary Conditions to the Template for it to successfully instantiate.

Why is this important?

For a more complicated testing project, more Conditions are required on different levels of the Template in order to fulfill the needs of the project.

Project perspective

After linking the Values, we can see that there are many paths available for testing the discount codes.This depends on what type of shipping an order has, or what discount code was applied. The Template will need Conditions to generate the TestCases that test the situations that occur as a result of choosing certain paths.

If the TestSheet shows that there should be a discount applied on the shipping cost, the Shipping TestStepValue that calculates the discounted shipping cost should be used. Otherwise, the Shipping TestStepValue with the value 10.00 is used. Likewise, each Discount TestStepValue is only used when a certain type of discount code is entered. The total cost is also calculated differently depending on whether there is a discount applied or not.

If the Instance Character is invalid, certain TestSteps should be skipped completely.

Finally, you need to complete the process by instantiating the Template and linking the TemplateInstances to the ExecutionList.

Instructions

1. Duplicate the TestCase folder named “**7b Link Values**” and rename it “**7c Set Conditions**”. ☐

2. To ensure that certain folders will be used only for valid test cases (i.e. those that are not meant to result in an error), add the CharacterInstance Condition **'Instance.Character' != "Invalid**" at the folder level for the following TestStep folders:
   - “Checkout Process”
   - “Verification of prices”
   - “Confirmation”
   - “Verification of Success”

   Note: This must be done manually and cannot be dragged and dropped like the other Conditions we have used previously. ☐

3. Add the CharacterInstance Condition **'Instance.Character' == "Invalid**" for the “**Empty Shopping Cart**” TestSteps Folder to make sure the cart is emptied after the wrong code is entered. ☐

4. Add the necessary **Conditions** to the Template in TestStep folder “**Verification of Prices”>>TestStep “Verification of Prices” >>Cart total** according to the table below: ☐

TestStepValue	Value	Condition
Shipping:*	10.00	‘Workflow.Discount Assignment' != "Discount Assignment Shipping”
Shipping:*	{MATH[(1-{XL[Verifications.Value - Discount %]})*10]}	‘Workflow.Discount Assignment’ == “Discount Assignment Shipping”
Discount:	-{MATH[({B[SubTotal]}+{B[Shipping]})*{XL[Verifications.Value - Discount %]}]}	'Workflow.Discount Type' == "Discount Type Percentage" AND 'Workflow. Discount Assignment' == "Discount Assignment Total"
Discount:	-{XL[Verifications.Value - Discount Flat Rate]}	'Workflow.Discount Type' == "Discount Type Flat""
Discount:	-{MATH[{XL[Verifications.Value - Discount %]}*{B[SubTotal]}]}	'Workflow.Discount Assignment' == “Discount Assignment Sub-Total”

TestStepValue	Value	Condition
Discount:	Discount Buffer	'Workflow.Discount Assignment' != "Discount Assignment Shipping"
Total:	{MATH[{B[SubTotal]}+{B[Shipping]}+{B[Discount]}]}	Workflow.Discount Assignment' != "Discount Assignment Shipping"
Total:	{MATH[{B[SubTotal]}+{B[Shipping]}]}	Workflow.Discount Assignment' == "Discount Assignment Shipping"

5. Use the **Check Template** command to check for errors. ☐
6. **Instantiate** the Template to create the TestCases from the TestSheet. ☐
7. Link the TemplateInstance Folder "**TemplateInstance of Discount Code**" to the Requirement DemoWebShop >> WebShop Frontend >> Shopping cart >> "**Discounts**". ☐
8. Create an ExecutionList in the "**ExecutionLists**" folder and name it "**7 Discount Code**". Link the "**TemplateInstance of Discount Code**" folder to the ExecutionList. ☐
9. Run the ExecutionList. ☐

Hints

- Drag and Drop the necessary Instances onto the Condition fields before adjusting the Conditions.
- Single quotation marks (') are must be used when writing the name of the Attribute in the Condition column when there are spaces in the Attribute name. If there are no spaces in the Attribute name, they may be omitted.