Lab 8-1

Lab – Looping through Search Item Results

In this lab, you will extract data for all items matching a search for a particular item on the HardyHardware web site. If no matching items are found, you want your Robot to terminate gracefully, but return a descriptive message to the log file.

1. Before you develop your Robot, go to the HardyHardware web site using your browser and note how you, as a human user accomplishes a search. Note that you have to manually type a value in the search box and then press the Enter key. The value you enter, of course changes depending on what you want to search for.
2. *To perform this kind of function using a Robot requires creating an input Variable. Remember that variables of simple types cannot be used as input variables (their values cannot be output either). So what you need to create is a new Type from which you will create a new Complex Variable for your new Robot.*

   In Design Studio, create a new type under the HardyHardware project named "Input.type."
3. Add one new Attribute to your Input Type named "SearchFor" and set its Type and Default Value to "Short Text. Make it "Required" but not part of the database key.
4. Create a new Robot named "Search_Item.robot". Set the starting URL to http://class.kofax.com/hardyhardware. Leave the Browser engine set to "Default."
5. Add a new variable to the robot using the "Input" Type.
   a. Check the "Use as Input" checkbox.
   b. Enter the word "pipes" in the "SearchFor" box. Click on [OK].
6. The HardyHardware home page should be already displayed in the browser panel. As a human user, you would enter a product you want to search for. To add that action as a step in your Robot, click on the "Search" box on the web page. Then right mouse-click on the highlighted search box and select "Enter Text" from the resulting context menu.
   a. On the **Basic** tab, for the Step Name, enter "Enter Search For."
   b. Examine the **Finders** tab. It should already be populated for you.
   c. On the **Action** tab, "Enter Text" should already be selected from the action drop down list. In the "Text to Enter" drop down, select the "Variable" option button and then "Input.SearchFor" as the variable to populate the value.
   d. On the **Error Handling tab**, API Exception and Log as Error should both be checked. "Skip Following Steps" should be selected from the dropdown.
7. *After entering a search term, a human operator would press the [Enter] key to execute the search. You can do the same thing by adding a "Press Key" Action Step.* Make sure you are on the end step in the robot view and that the Search box on the web page has been selected. Right mouse-click on the search box and select "Press Key" from the context menu. Accept the default "Enter" on the resulting dialog.

8. If you have the end step selected (the search has been executed), you'll notice that there are multiple items returned. You want to select the first item, return details from which you'll perform your extraction, and then loop through all items on the page. BUT…every once in a while, depending on the search term (*pipe* instead of *pipes*, for example) the first couple of items in the returned list might be a *category* of items rather than only *individual* items. In these cases, the extraction steps you're going to set up will fail because the html tag is different for categories. To remedy that problem:
   a. In your browser panel, scroll down to the "Search Only:" section of the web page and select the "Products" checkbox. Once it's selected (green box), right mouse-click on it and select "Set checkbox" from the context menu.
   b. A "Set Checkbox" dialog box will pop up. Set Checkbox to "Checked" and click [OK].
   c. Scroll back up to the top of the web page and re-execute the search by selecting the search icon (magnifying class), right mouse-clicking on it and selecting "Click" from the context menu.

9. Now, let's set up our loop for extraction:
   a. Left mouse-click on the first item presented (2-Handle Service Sink Faucet…) and then right mouse-click on the green box that is presented around that item.
   b. From the resulting context menu, select "Loop" and then "For Each URL."
   c. Test your loop by clicking on the forward arrow in the "For Each URL" loop step just created. Make sure the blue box advances as expected, item by item. After testing, reset the number in the loop step to 1.

10. Then you'll need to add a "Click" Action step between your loop step and the end step. With the end step selected, click on the first item in your browser panel (2-Handle Service Sink Faucet…), right mouse-click on the green box shown around the item and from the context menu, select "Click."

11. Change the name of the action step you just created to "Click item."

12. Now you're going to add extraction to your robot. But you've already created a Snippet in a previous Robot to extract what you want. Let's reuse it here. The extraction will be exactly the same as in our previous Robot.
   a. Right mouse-click on the end step in the robot view. Select "Insert Step Before" from the context menu. Then select "Snippet Step.
   b. Select the "ExtractData.snippet" file you created in the last lab.

13. And for testing purposes, let's add a Return Value step.
   a. Right mouse-click on the end step. Select "Insert Step Before."
   b. Select "Return Value.
   c. On the Basic tab, select the HardyHardware variable.

14. Click on the "Debug" mode tab and run a test in debug mode.
   a. NOTE 1: You can change the name of the input variable to return different results..

b. NOTE 2: Your Robot only loops through the 20 items displayed on the first page…so far.

15. Leave Debug mode and go back to Design mode. Notice that there are 3 pages of pipes, so now you need to set up another kind of loop to return ALL of the results. Let's set up a loop to loop through all pages. Begin by selecting the step BEFORE WHICH you want the loop to occur. That step is the "For Each URL" step.

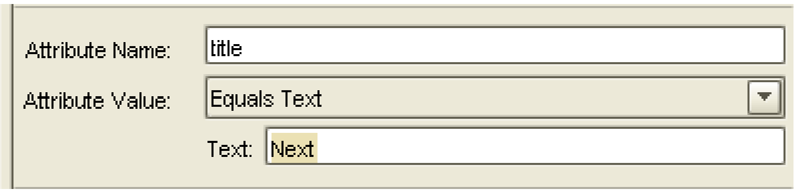
16. Right mouse click on it and select "Insert Step Before" and then "Action Step."

17. Select your new unnamed action step and click on the "Action" tab in its properties. Click on the "Select an Action" dropdown and then Select "Loop" and "Repeat."

18. Remember, a Repeat Step Always Goes with a Next Step. You want the Next step to execute after each iteration of extraction which will increment the iteration by one…and start the loop over. To do that, you must put the Next step on a branch. With your new "Repeat" loop selected in the robot view, click on the [Add Branch from Selected Step] button on the main toolbar.

19. You want to configure the Robot to click on the [Next] button at the bottom of the web page displayed in the browser view. Make sure the end step on the branch is selected and click on the [Next] button on the web page. Right mouse click on the green box surrounding "Next" and select "Click" from the context menu.

20. Go to the "Finders" tab of your Click Next step. Manually enter the following on the tab:



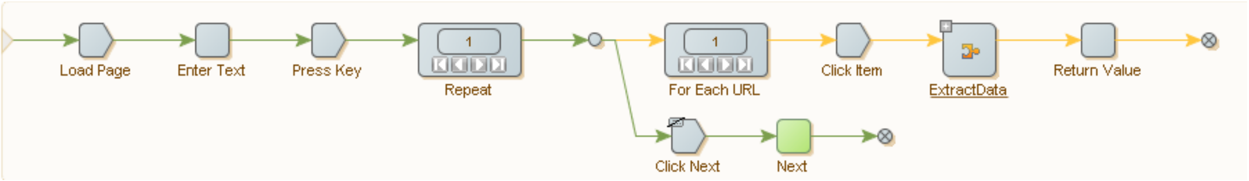
Attribute Name: title
Attribute Value: Equals Text
Text: Next

This comes from examining the HTML code for the click next step. The Next button always has the "title" attribute and "Next" as the text of the button.

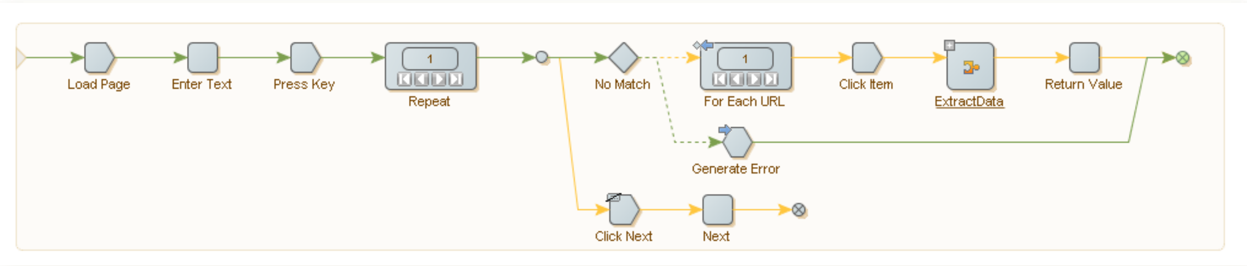
21. But what if "Next" doesn't show up on the page because your search is only one page long or because you've reached the last page for an item? An error would be returned. You need to tell your Robot what to do. Go to the properties of your new click step and select the **Error Handling** tab. Uncheck "API Exception" and "Log as Error." And from the "Then" dropdown, select "Break Loop."

22. Now you need a "Next" step to repeat the loop. Right mouse-click on the end step in the branch you just created. Select "Insert Step Before" and "Action Step."

23. Go to the **Action** tab of the new step and from the "Select an Action" dropdown select "Loop" and then "Next." Now your robot looks like this:



24. Test your Robot in Debug mode. NOTE: You don't have to wait until all items are returned. There are a lot of pipes. Just make sure that you are returning new items after item number 20 (that's the number on the first page). You may stop the robot with the [Stop] button on the toolbar.
25. There's one more consideration to address: What Happens if there are no items returned for a search? An API exception would be generated. You want to log a more user-friendly, descriptive message instead – and stop the robot from running – gracefully. To address this, you'll add a "Try" step.
    a. Right mouse-click on the "For Each URL" loop step.
    b. Select "Insert Step Before" and then "Try Step" from the context menu..
    c. On the Basic tab in the Try Step's properties name the step, "No Match."
26. You need to tell the robot what to do if there is no match for the item. What you want it to do is to go back to the branch created by the try step, return a message without throwing an error, and terminate.
    a. To do that, select the "For Each URL" step and select it's **Error Handling** tab. Uncheck "API Exception" and "Log as Error."
    b. And select "Try Next Alternative" from the "Then" dropdown. This will send the Robot back to the Try step where it takes the next alternative branch and then it will end.
27. But you also want to log a descriptive message that the item searched for wasn't found. Click on the end step of the Try step branch. Right mouse-click and select "Insert Step Before." From the new step's **Action** tab and dropdown, select "Other" and then "Generate Error." Enter "No matching items found" in the Error Message box.
28. Then go to the **Error Handling** tab, uncheck "API Exception" but check "Log as Error. Then "Ignore and Continue."
29. Finally, drag the end step after "Generate Error" to the end step on the line above. Your robot should now look like this:



30. Test in Debug mode. Try entering a new search term as input.
31. Save your Robot.