Kofax Kapow 10.3 Training and Certification

Module 8 – Input Variables/More Loops

Kofax Kapow™

Using a variable for input
Repeat/Next Loops

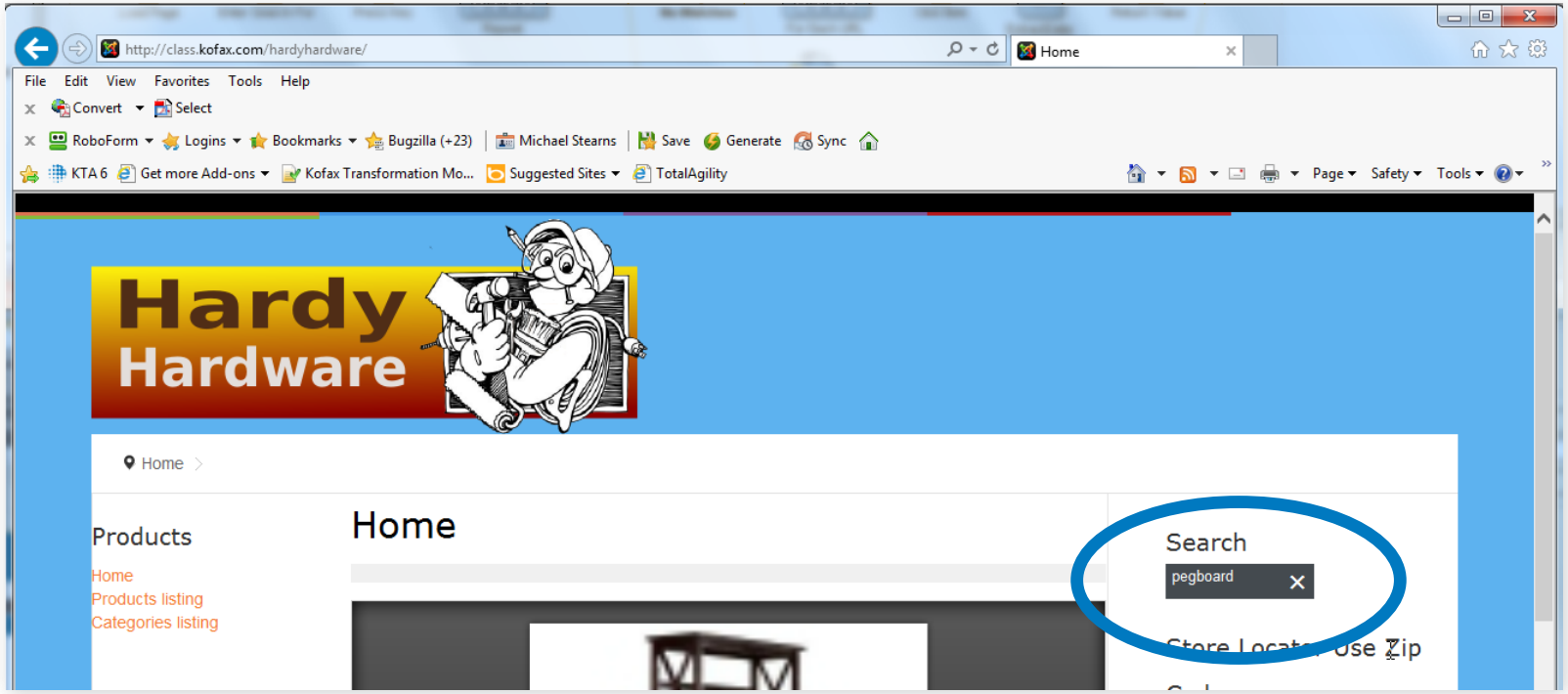
KOFAX

Kofax Kapow Module Overview

- What we Want to Do
- Creating a New Complex Type for Input
- Creating a New Robot
- Creating an Input Variable
- Entering Text as a Variable Value
- Looping through Lines and Pages
- Branches
- Extracting Values
- Try Steps
- Returning and Logging Error Text
- Returning Values

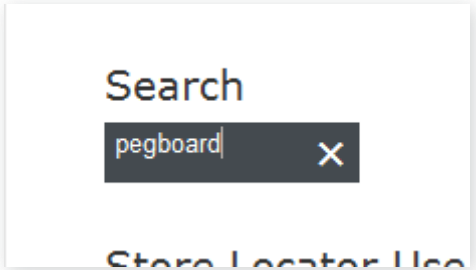
What We Want to Do

- We want to extract data for all items matching a search for a particular item on the HardyHardware web site.
- If no matching items are found, we want our Robot to terminate, but return a descriptive message to the log file.



How Does a Human Do It?

- To accomplish this, an operator would have to manually type a value in the search box and then press the Enter key.
- This value will change depending on what you want to search for.
- Creating an Input Variable will accomplish what we want to do.

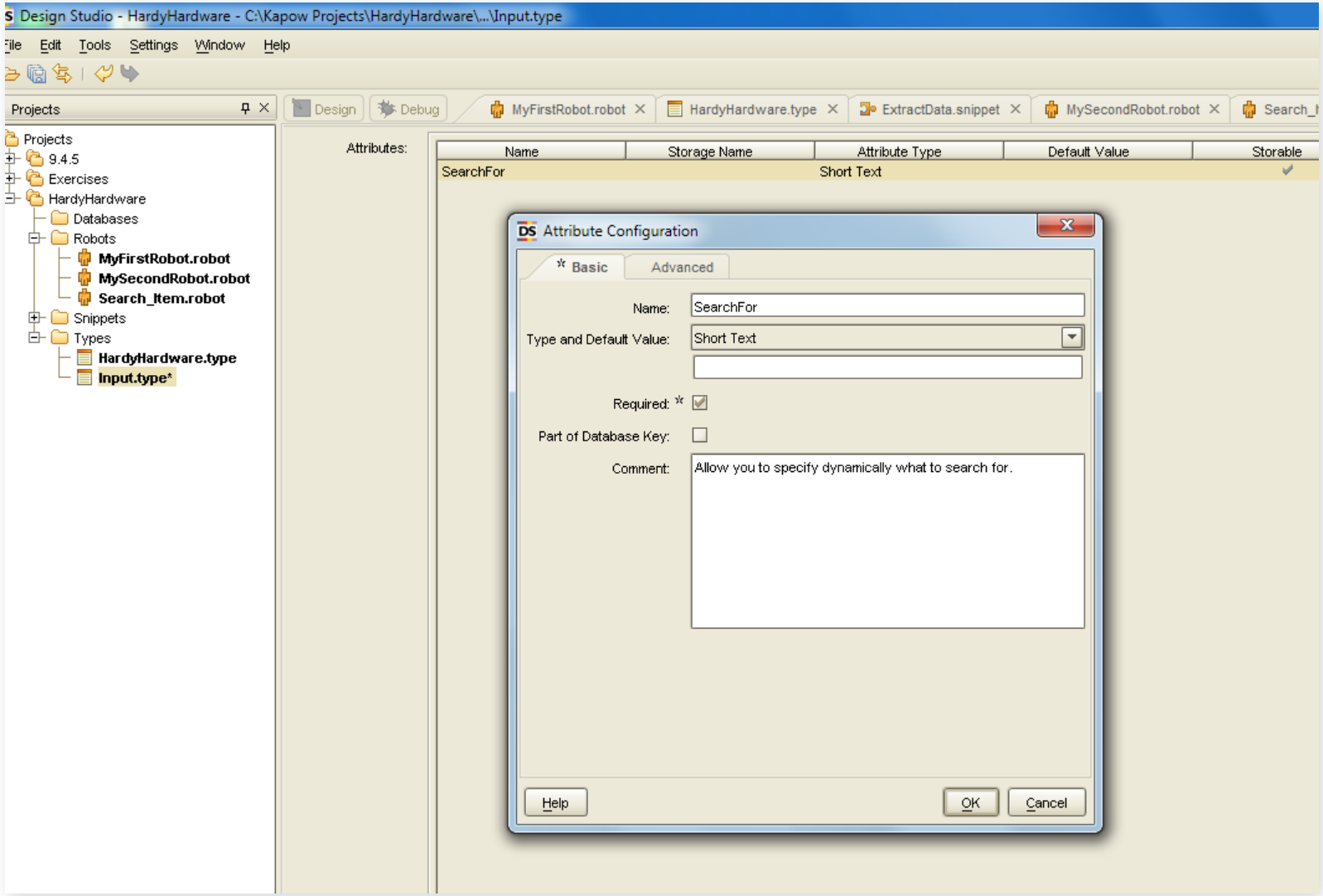


A Little Bit about Variables

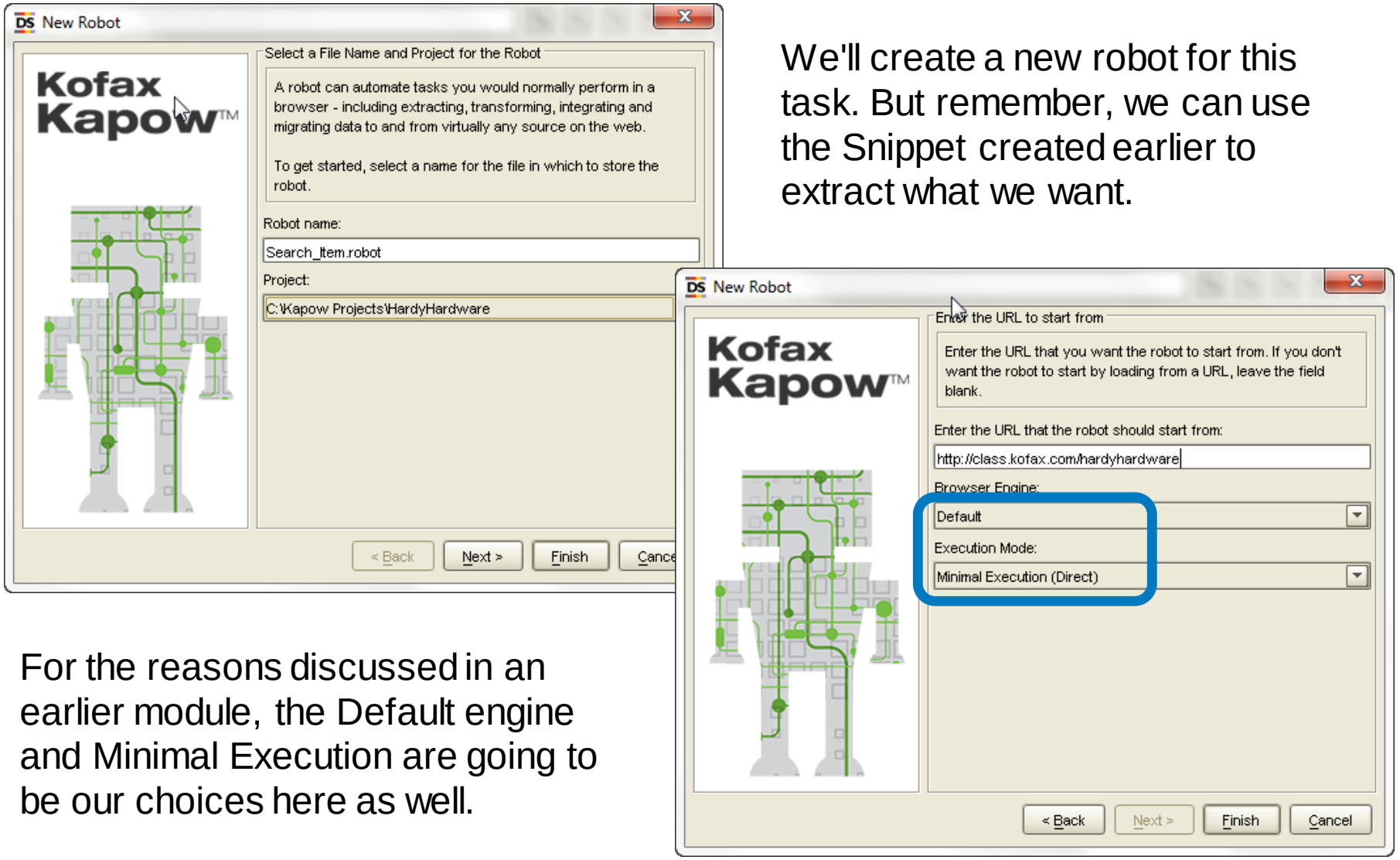
- Every variable can be associated with a default initial value that it retains unless the robot explicitly reassigns it, which it often will as values are extracted and manipulated during the execution. Most robots output the values of variables, e.g. by returning them to the caller or inserting them in a database. Robots might also take **input values** which are then assigned to specific variables that have been marked as receiving their values from input. These are simply called **input variables**.
- An important difference between variables of complex and simple types is that variables of simple types cannot be used as input variables, and their values cannot be output. However, they are useful, for instance, for extracting temporary data or as global counters. Generally, simple variables should be viewed as temporary variables, internal to the robot. They are not associated with Types.

Create a New Type

- So…we're creating a new Type called "Input" with one Attribute called SearchFor. It is marked "Required."



Create a New Robot

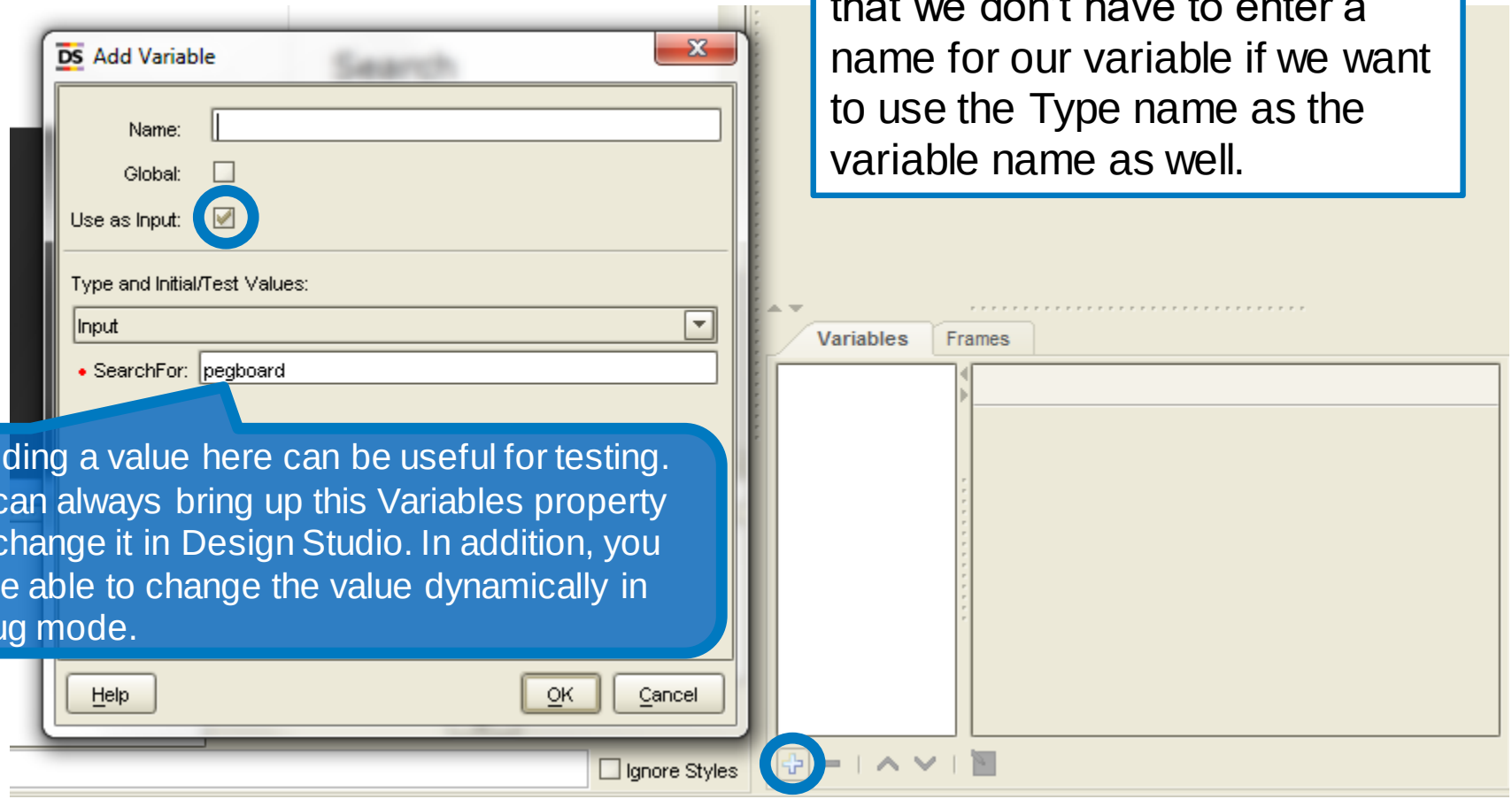


We'll create a new robot for this task. But remember, we can use the Snippet created earlier to extract what we want.

For the reasons discussed in an earlier module, the Default engine and Minimal Execution are going to be our choices here as well.

Add Variables

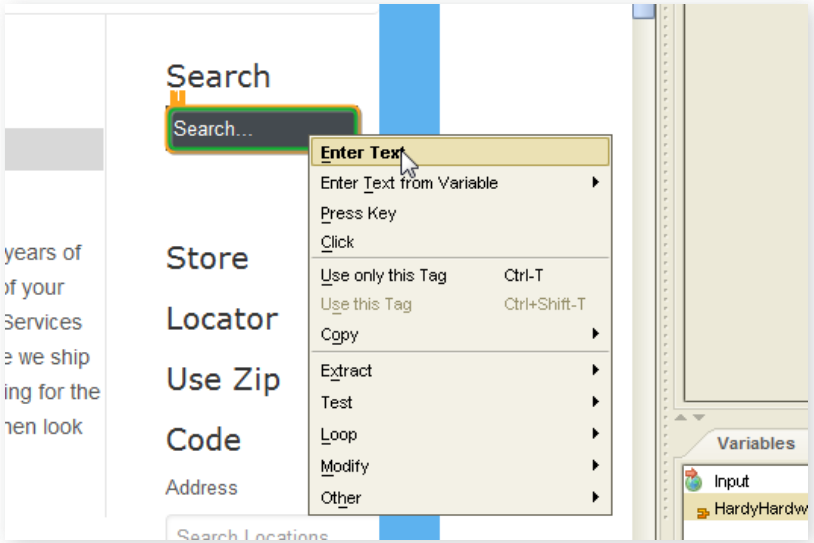
We'll add our Input Type…*as an input variable.* We do this by checking "Use as Input.“ Note that we don’t have to enter a name for our variable if we want to use the Type name as the variable name as well.

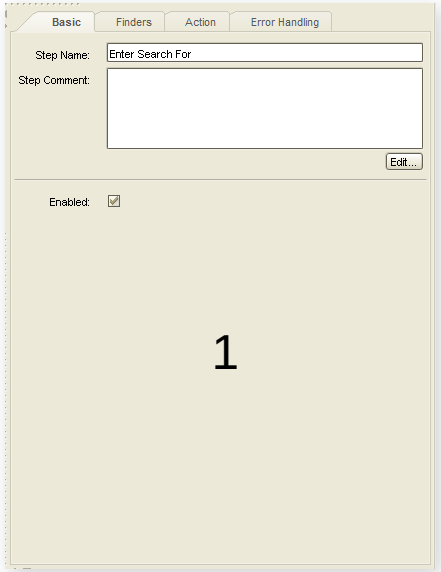


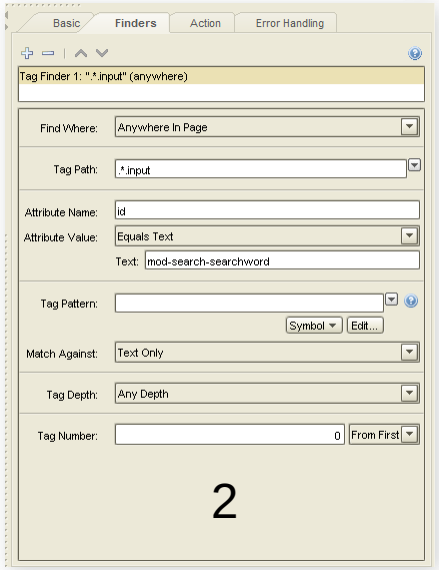
Providing a value here can be useful for testing. You can always bring up this Variables property and change it in Design Studio. In addition, you will be able to change the value dynamically in Debug mode.

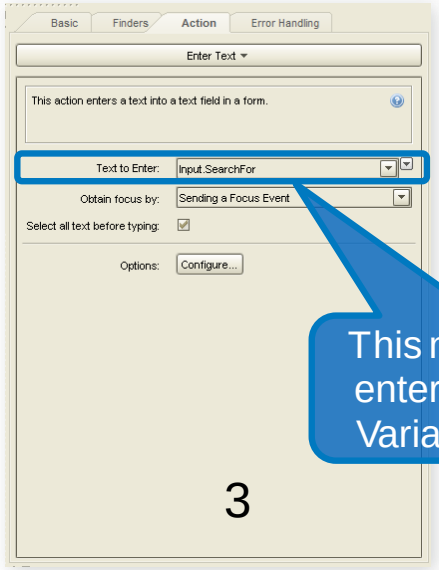
After the Load Page Step, add an Enter Text Action Step

- The Enter Text Step replicates the input a human operator would provide to do a lookup of specific items

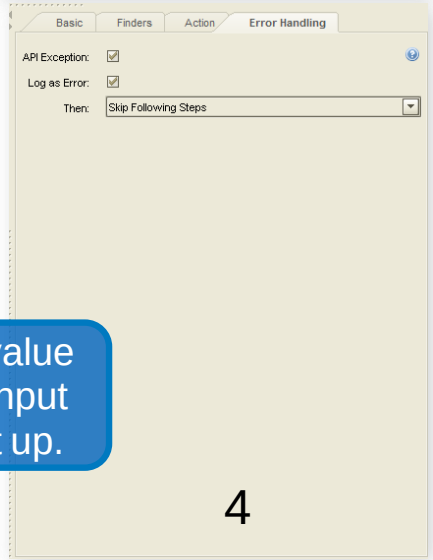






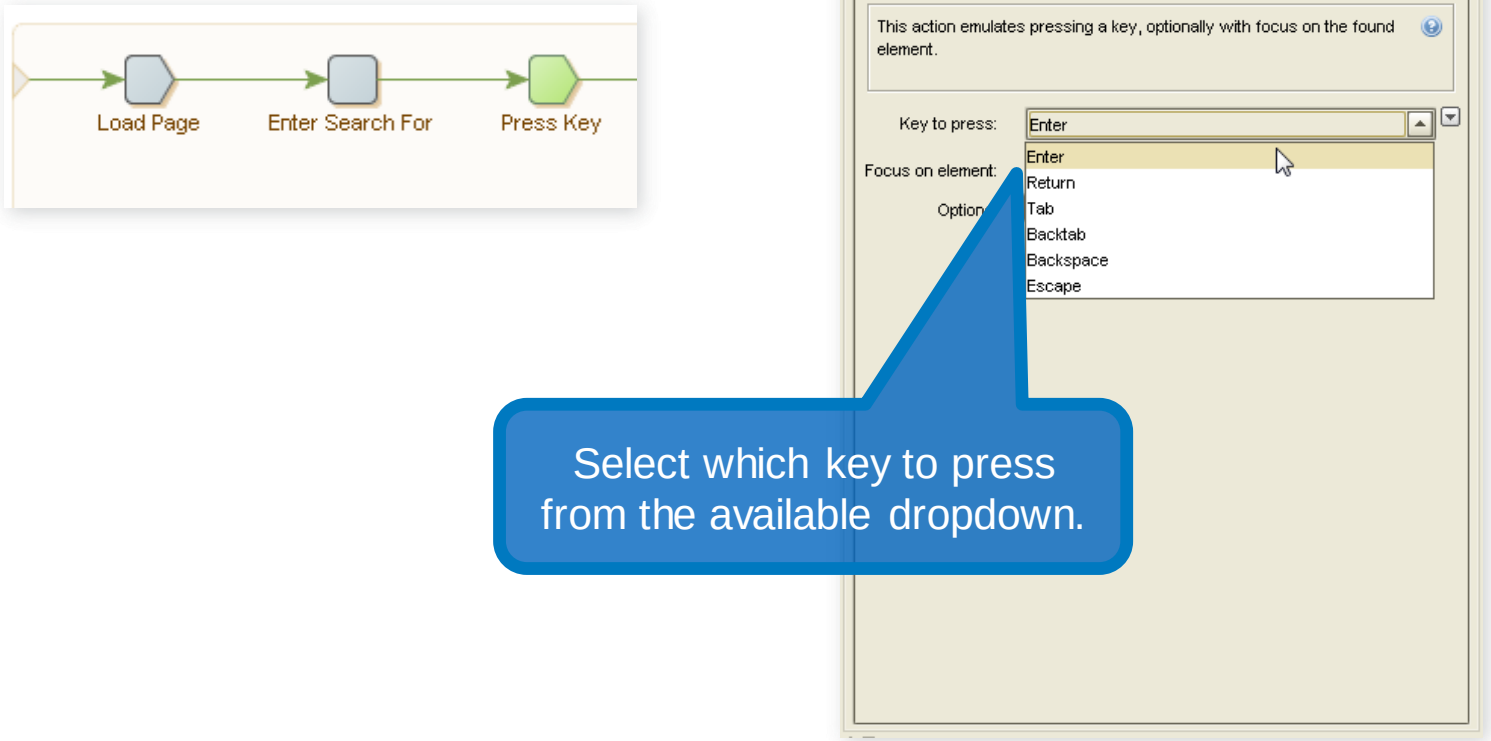


This maps the value entered to the Input Variable we set up.



Press Key Action Step

- Next, an operator would press the [Enter] key to execute the search. We can do the same thing by adding a Press Key Action Step.

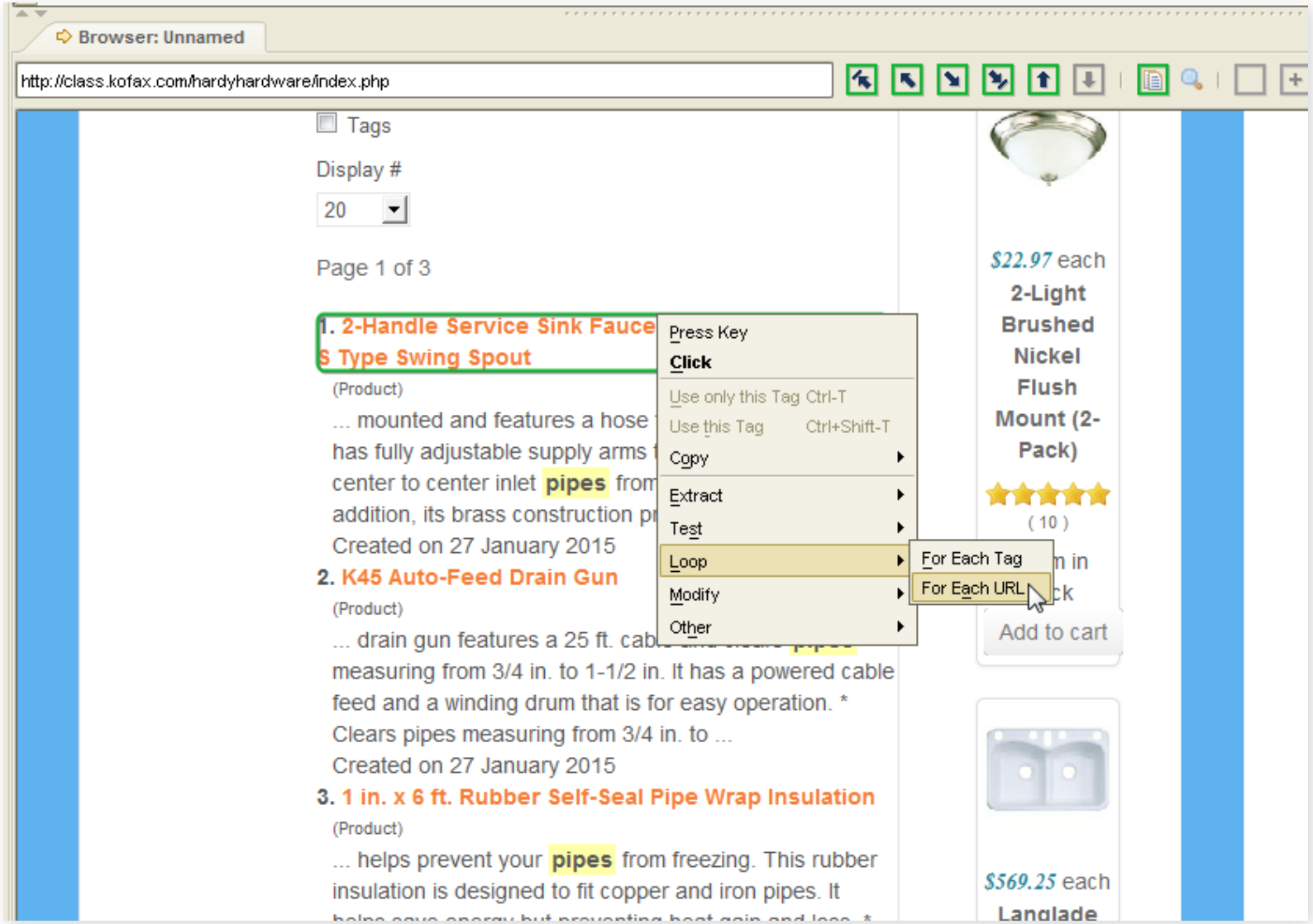


So where are we? What's next?

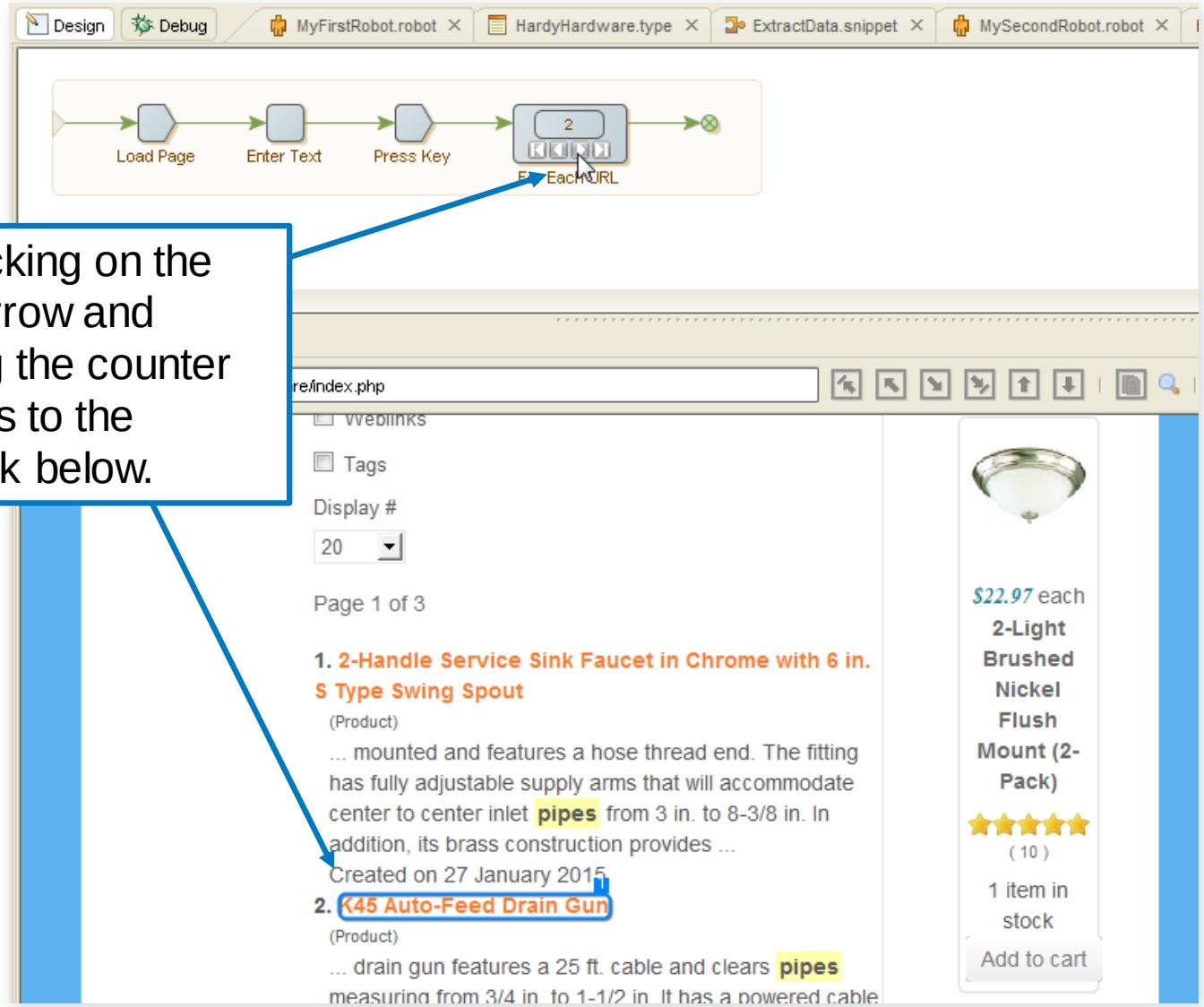
- So far, we've entered a URL, entered a search and pressed [Enter]. This is an example of what shows up when we search for "Pipes."
- There a many results on multiple lines, and on multiple pages.
- Note there is a [Next] button at the bottom of the page.

Let's Set up a Loop through the Items on One Page First

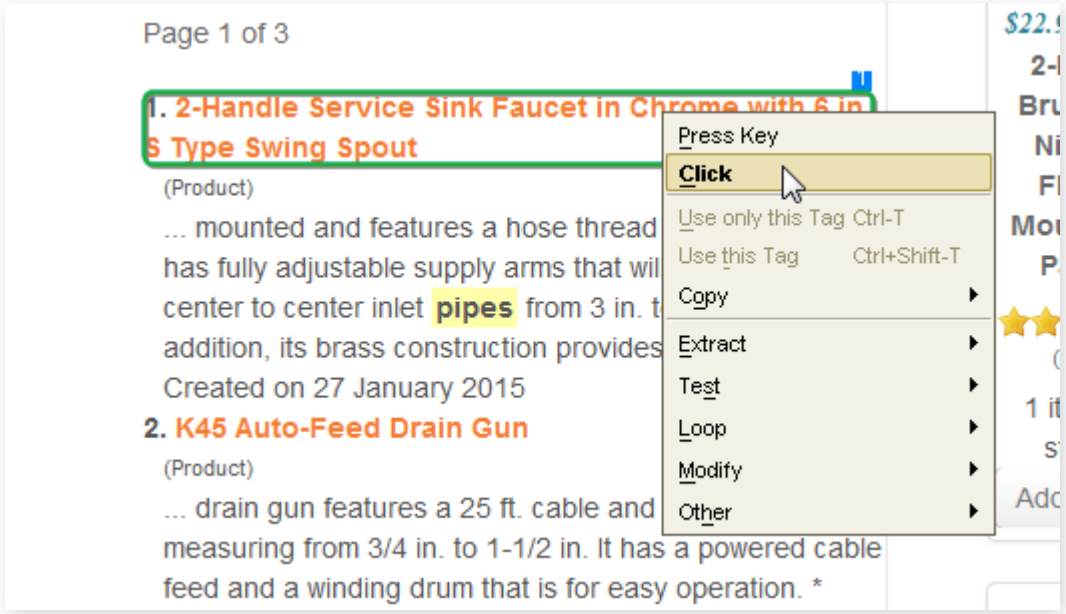
- You've already accomplished this in an earlier lab, so this should be easy.



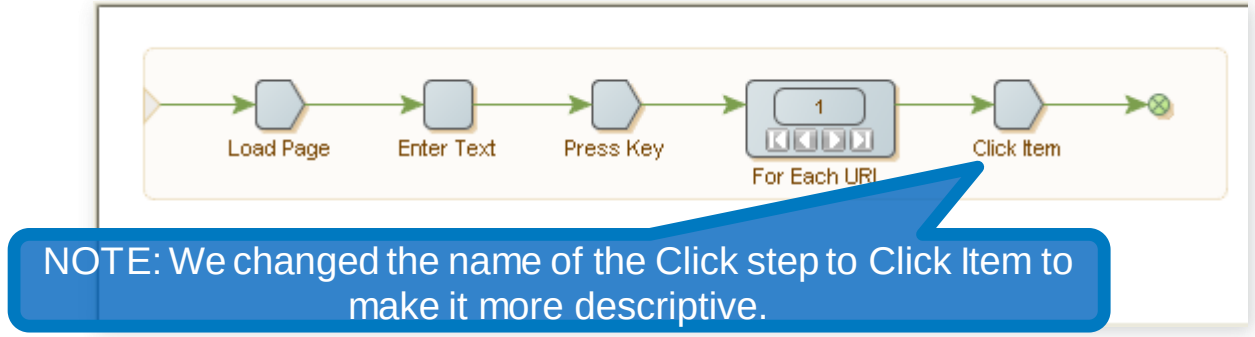
And Test Our Loop



Then We Need to Click on Our Item; Add a Click Step

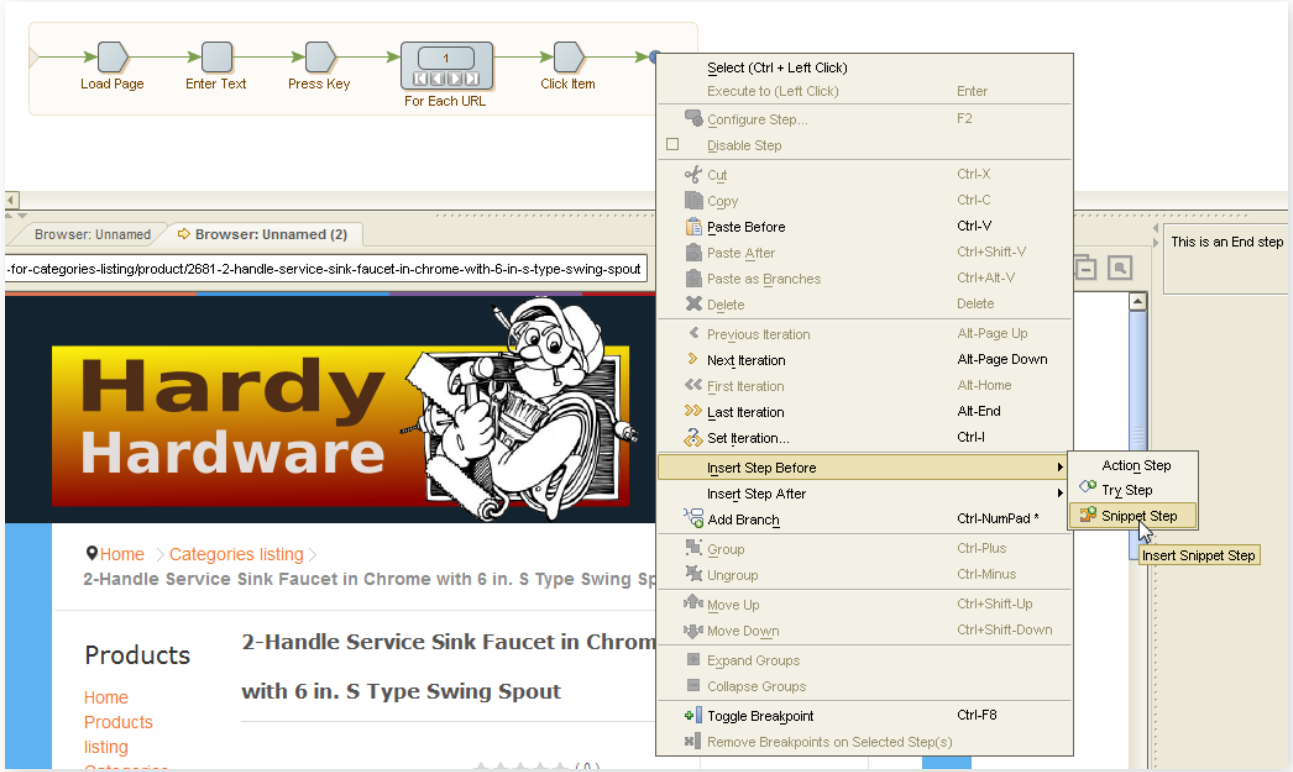


And our robot looks like this so far…



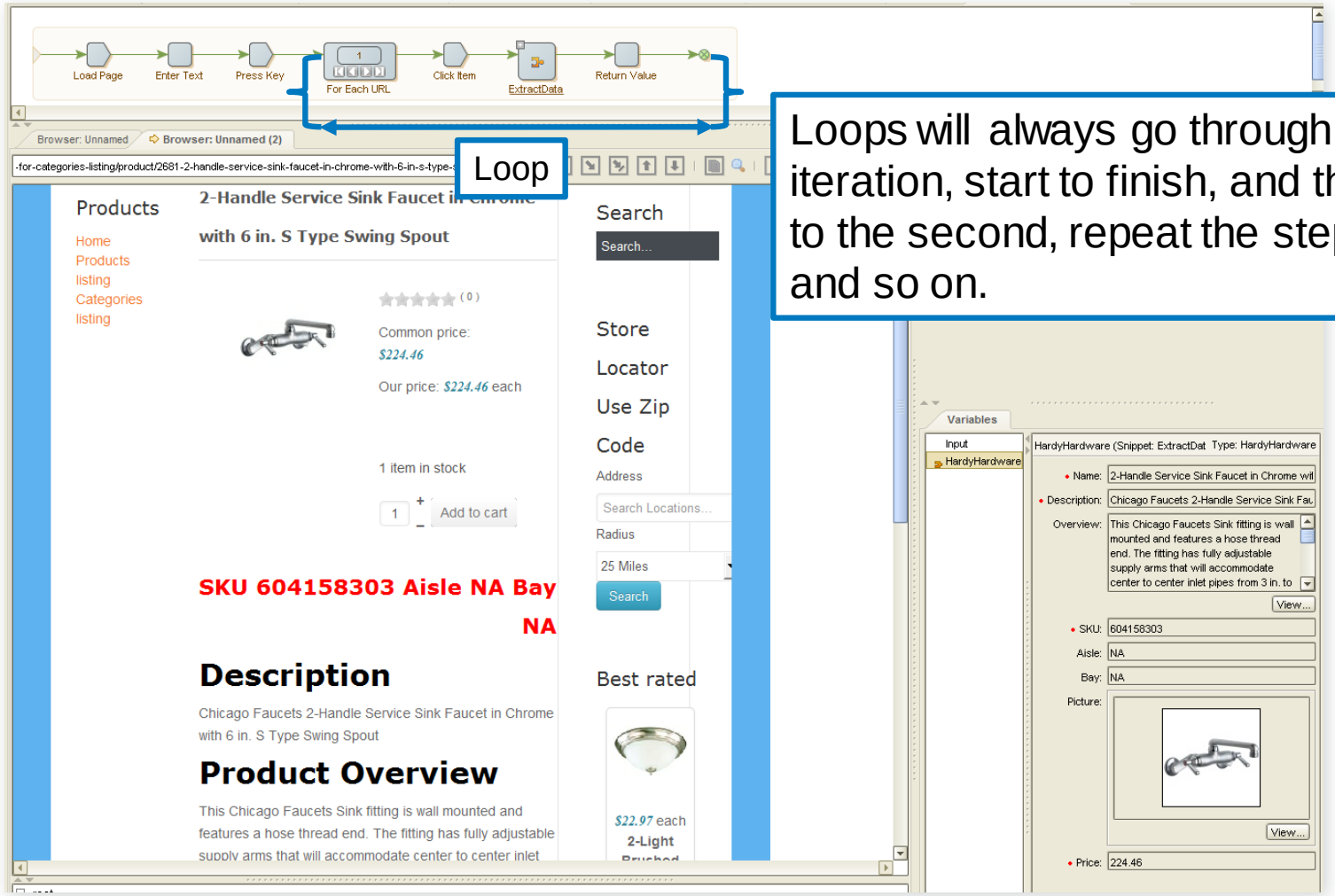
Then Add our Extraction Snippet

- The extraction will be the same as in our previous Robot.
- Remember, we've already created a Snippet for that.
- Let's add the Snippet to perform extraction.



And Return Values

- And for testing purposes, let's add a Return Value step.

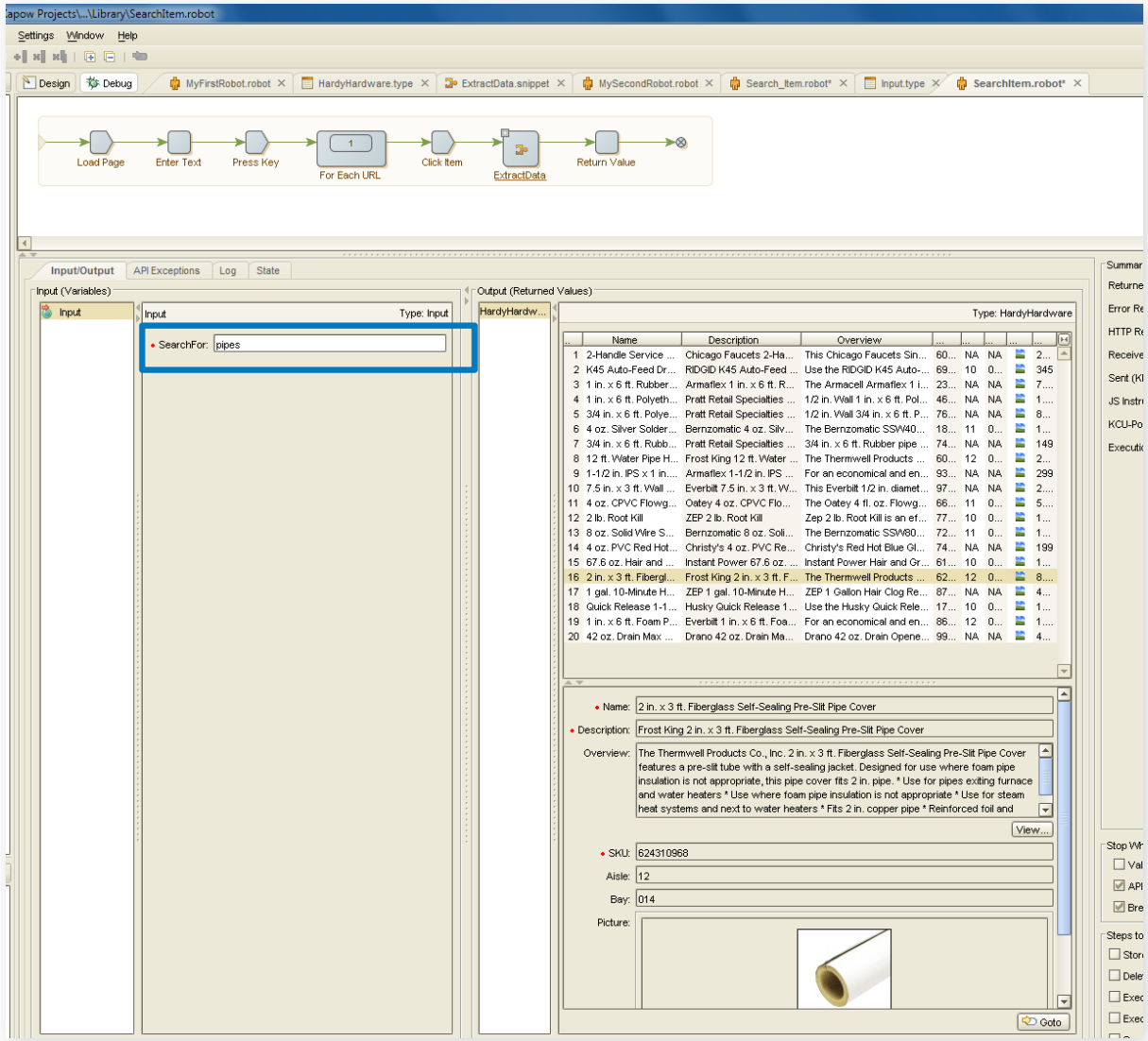


Loops will always go through one iteration, start to finish, and then to the second, repeat the steps, and so on.

Notice that the HardyHardware variable has been automatically added because it was included with the Snippet

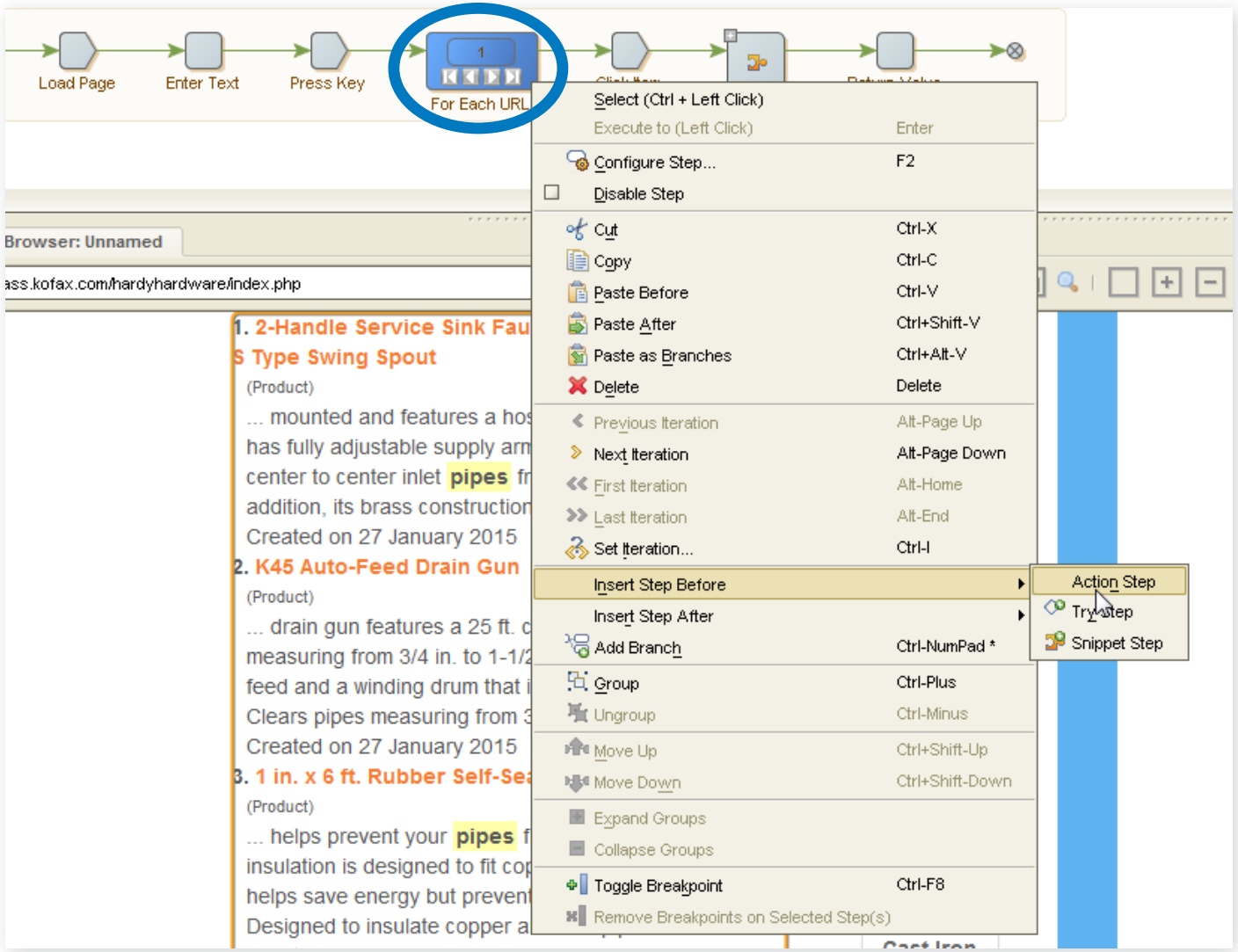
And Test in Debug Mode

- Note that in Debug mode, if you've set up an input variable as we have in this Robot, you can dynamically supply a value to test against.



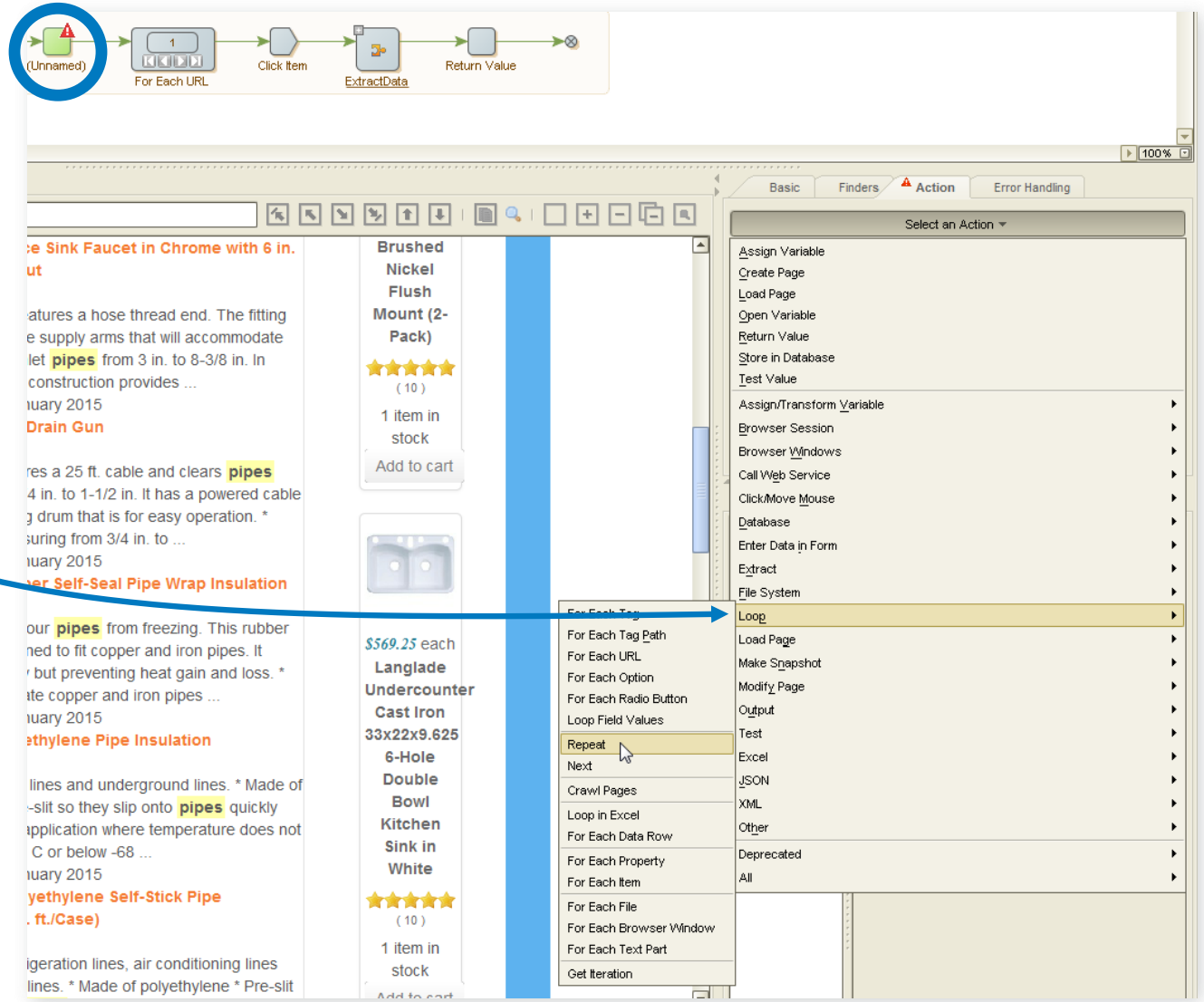
Loop through Pages

- Now we want to add an action step that loops through every page of our returned results.
- That can be accomplished with a Repeat/Next loop.
- Since we will want to go to the next page BEFORE we loop through each URL, we'll insert the Repeat/Next loop there.



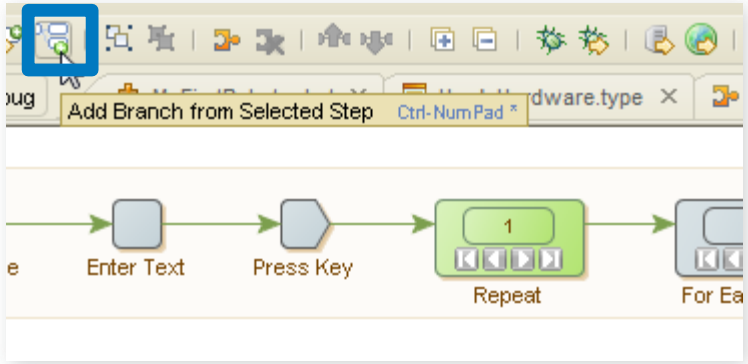
Set Action for New Unnamed Step

- So we'll select the new unnamed step and for the action, select "Loop" and then "Repeat."

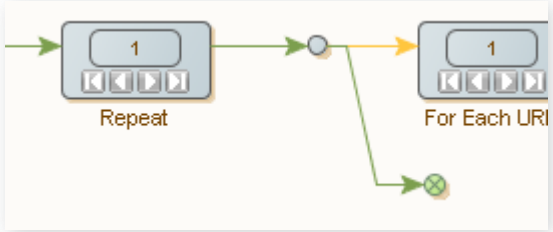


Repeat Step Always Goes with a Next Step

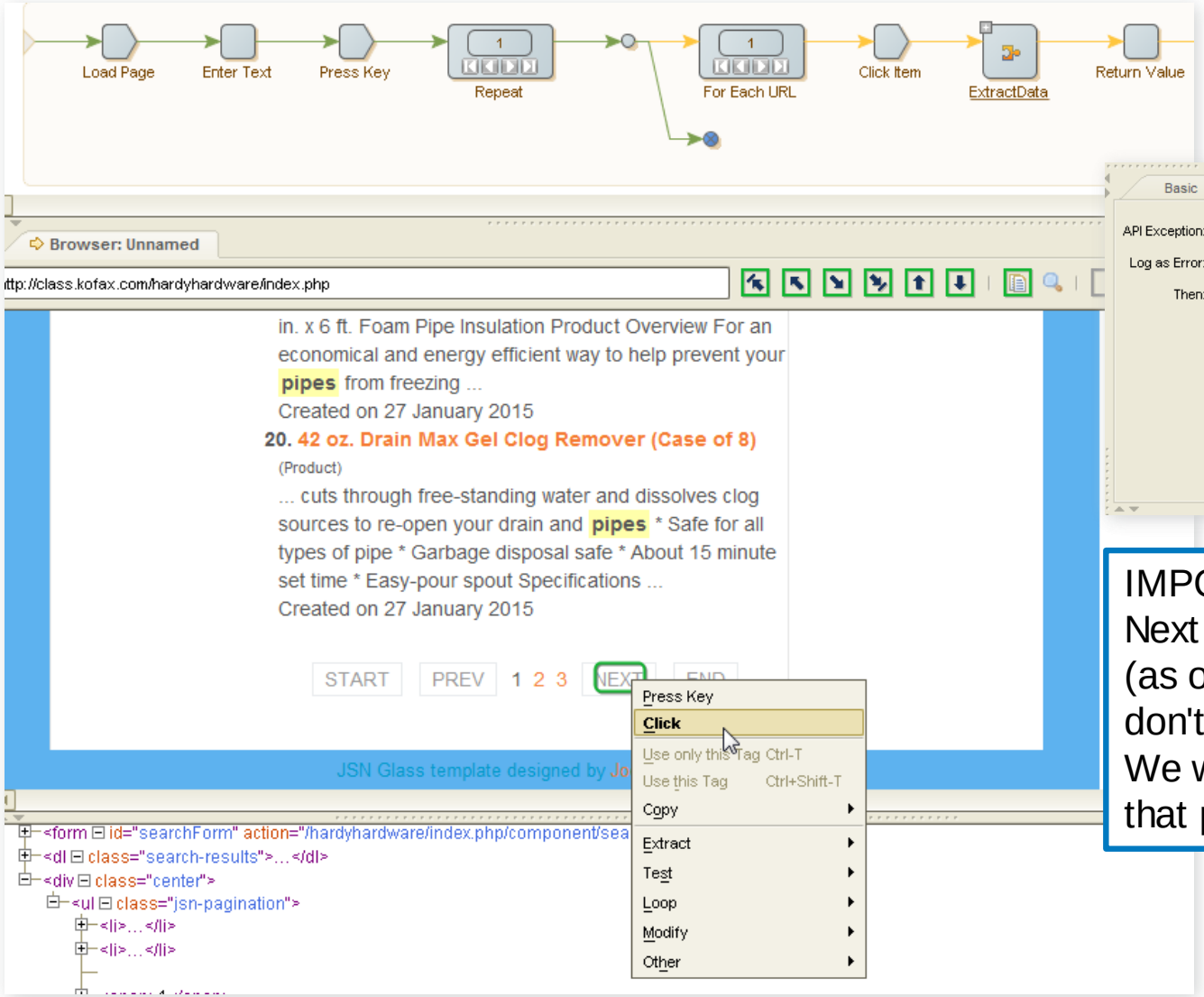
Now we're going to add a Next Step, BUT…
We want the Next step to execute after each iteration of extraction which will increment the iteration by one…and start the loop over. To do that, we must put the Next step on a branch. This will be more fully explained in a minute.



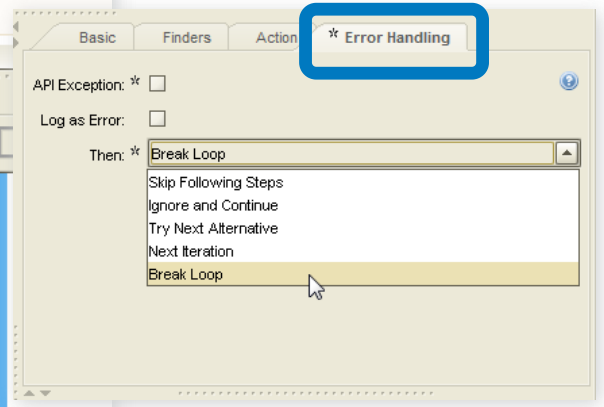
The branch has been added. Now we can add a Click step (to click on the Next button at the bottom of the page), followed by the required Next step to return to the loop.



Add Click Step to Branch



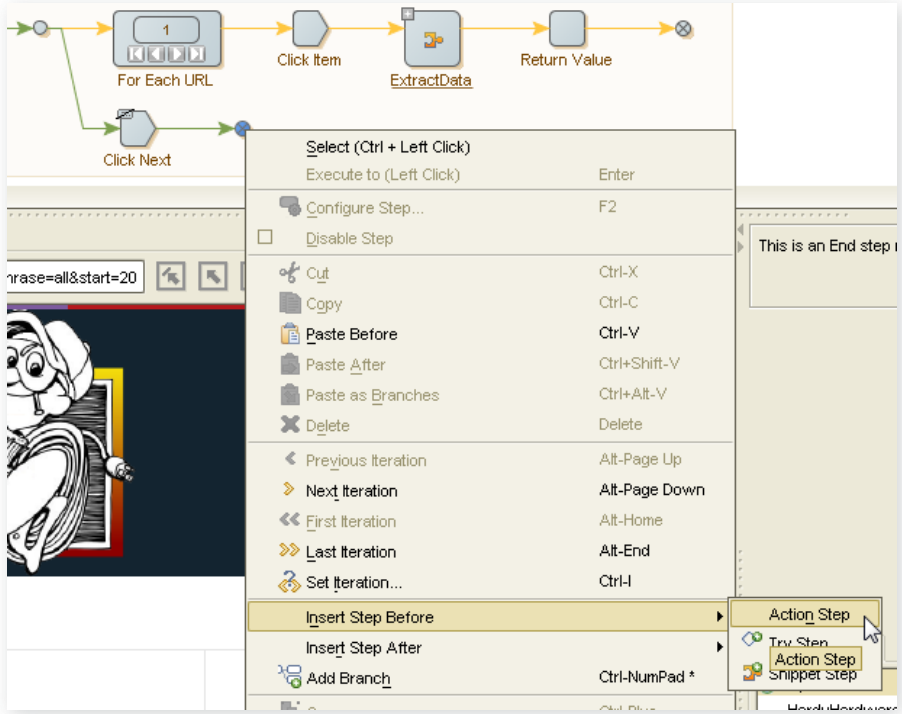
Set properties of step.

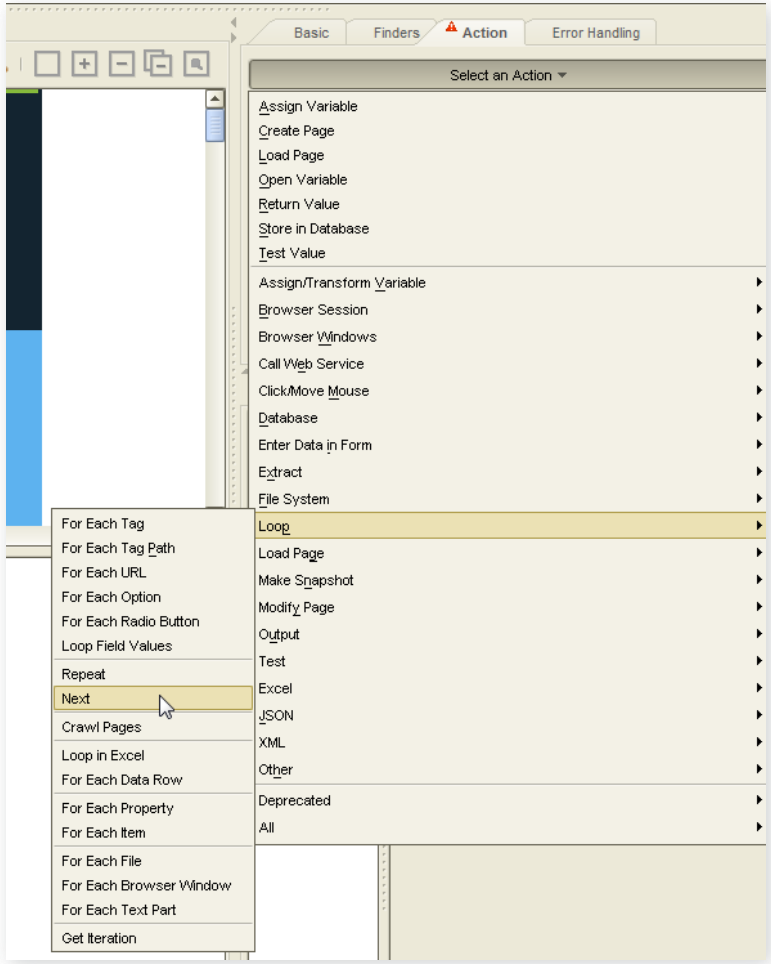


IMPORTANT: What if the Next button isn't available (as on the last page)? We don't want to report an error. We want to break the loop at that point.

Then Add Next Step

Add the Next step as an Action Step before the End Step on the branch.

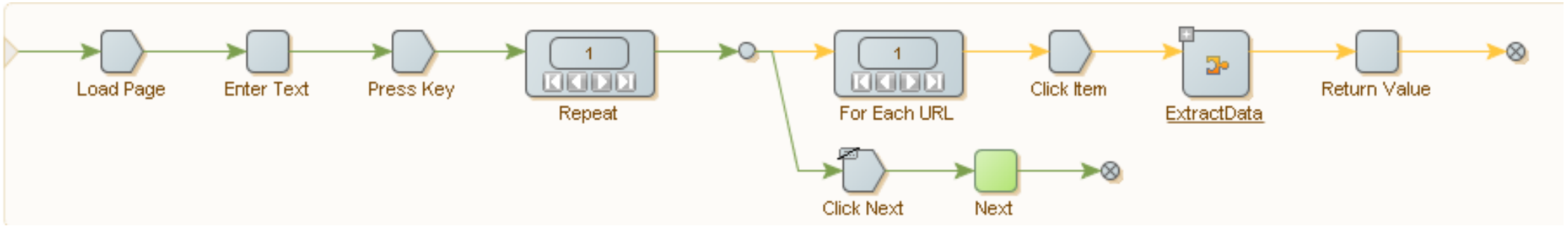




Then choose "Loop" and "Next" as the action you want to perform.

Now Your Robot Looks Like This

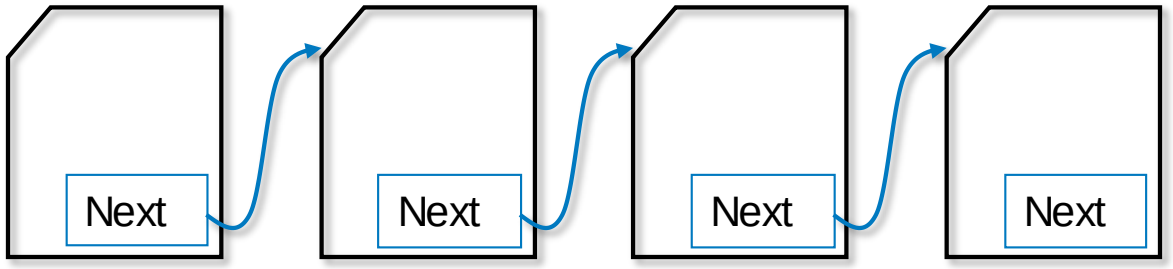
1. The main page is loaded.
2. The search text is entered.
3. The [Enter] key is pressed.
4. The first page is displayed.

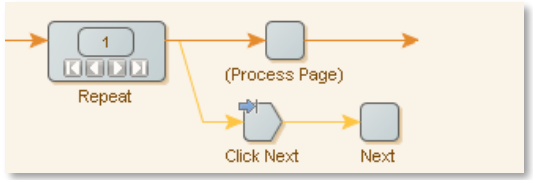


5. The first item is selected and clicked…
6. And the data in the snippet is extracted…
7. And returned. Then the End Step on the first item is encountered.
8. The For Each URL repeats, this time with the second item and so on until the bottom of the page is reached. There are no more items on that page.
9. The Robot returns to the branch point and tries the second alternative which clicks on the Next button. Then it reaches the Next step.
10. The Robot returns to the Repeat step which is now iteration 2 and continues.
11. When there is no Next button, the loop is broken and the Robot is finished.

Repeat Loops

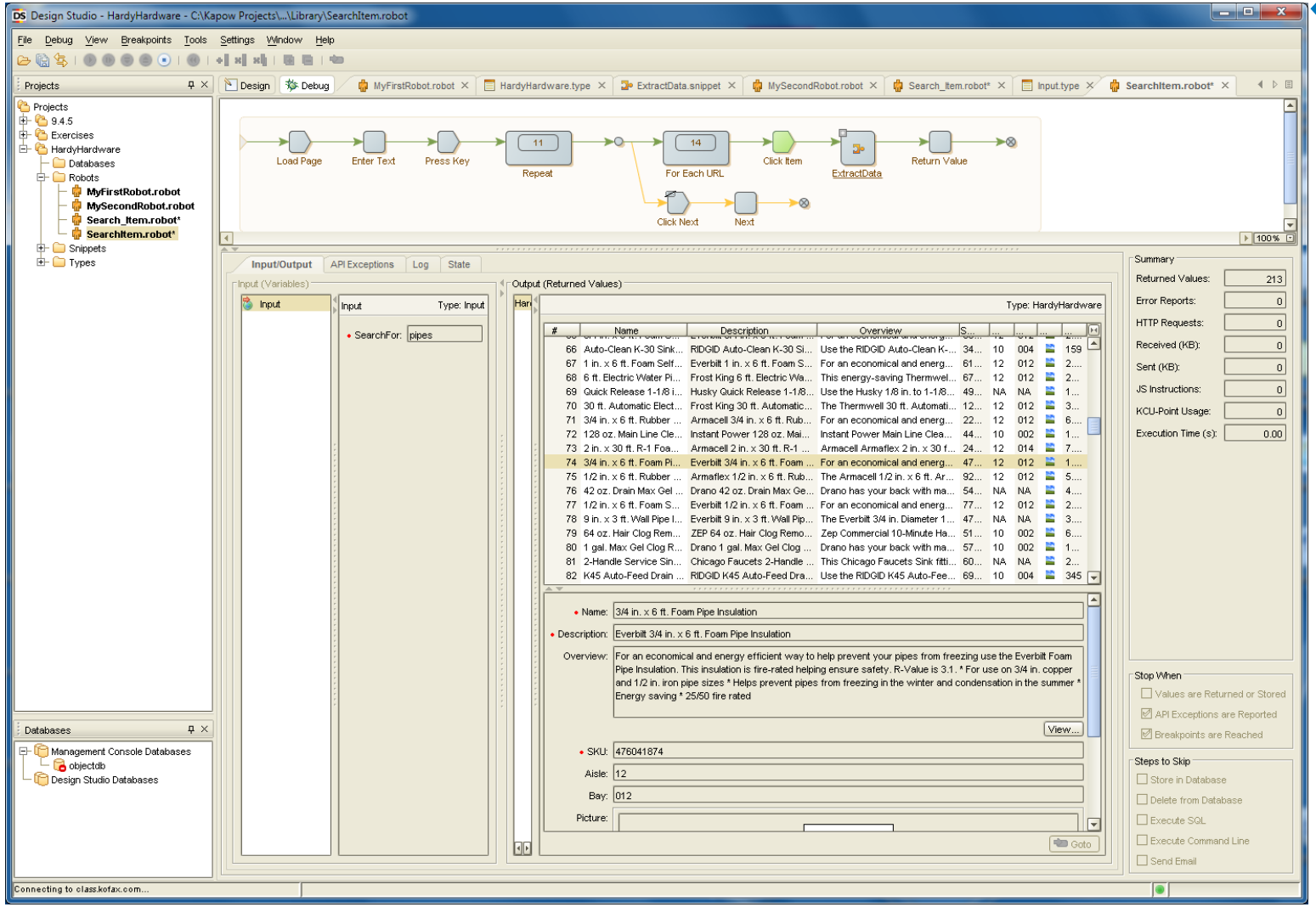
- Repeat/Next Loops...
  - This action creates a repeat loop together with the Next action.
  - The Repeat action marks the start of the repeat loop. In a subsequent step, another iteration of the loop can be requested using a Next action.
  - The windows, pages, etc. at the Next step will be sent back to the Repeat step and will become the output from the Repeat step in the next iteration. If no Next step is executed in a given iteration, that iteration will be the last one, and the repeat loop will end.





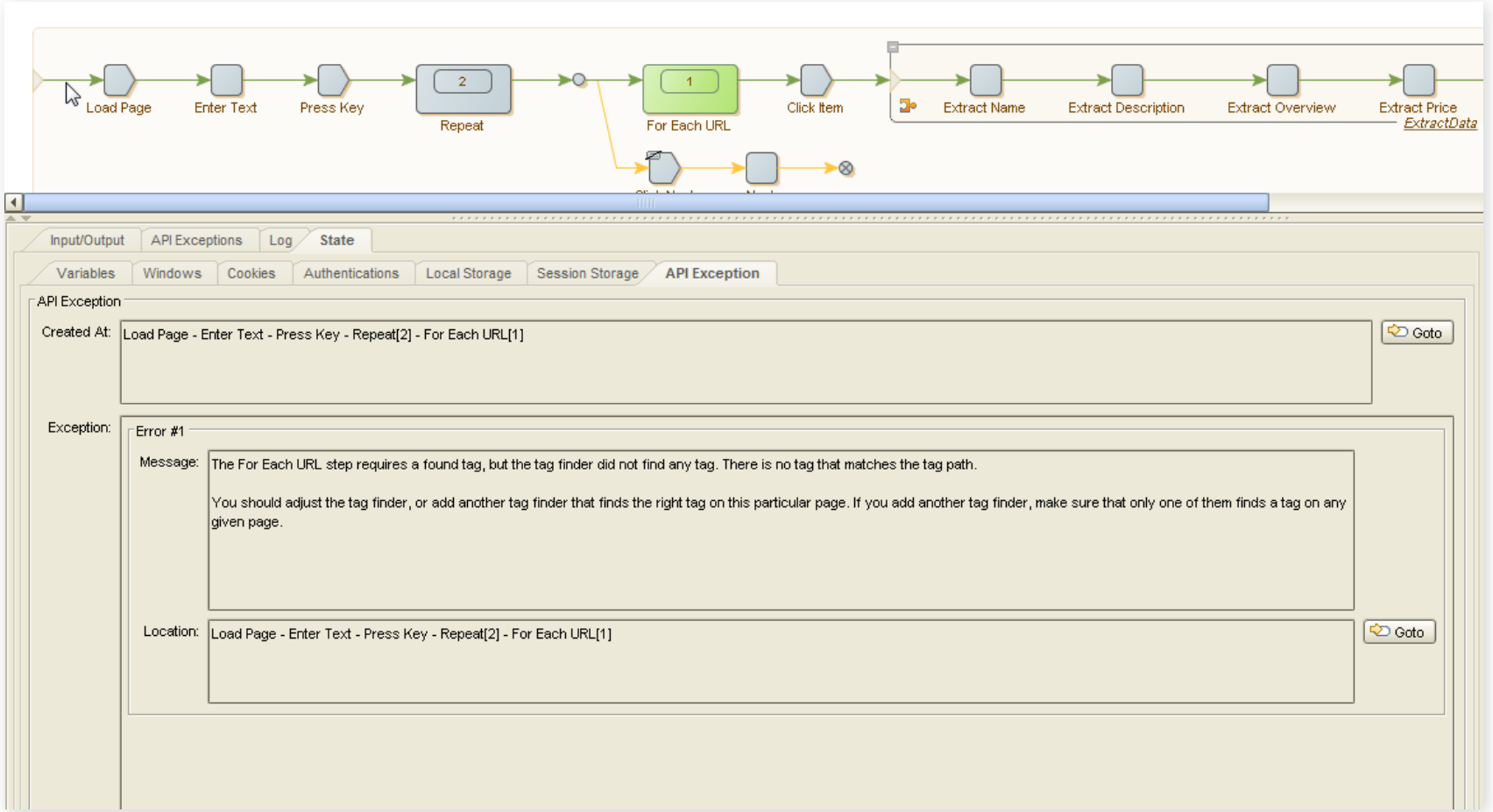
Testing in Debug Mode

- A test in Debug mode reveals our robot runs without errors and returns a lot of items at HardyHardware associated with "pipes."



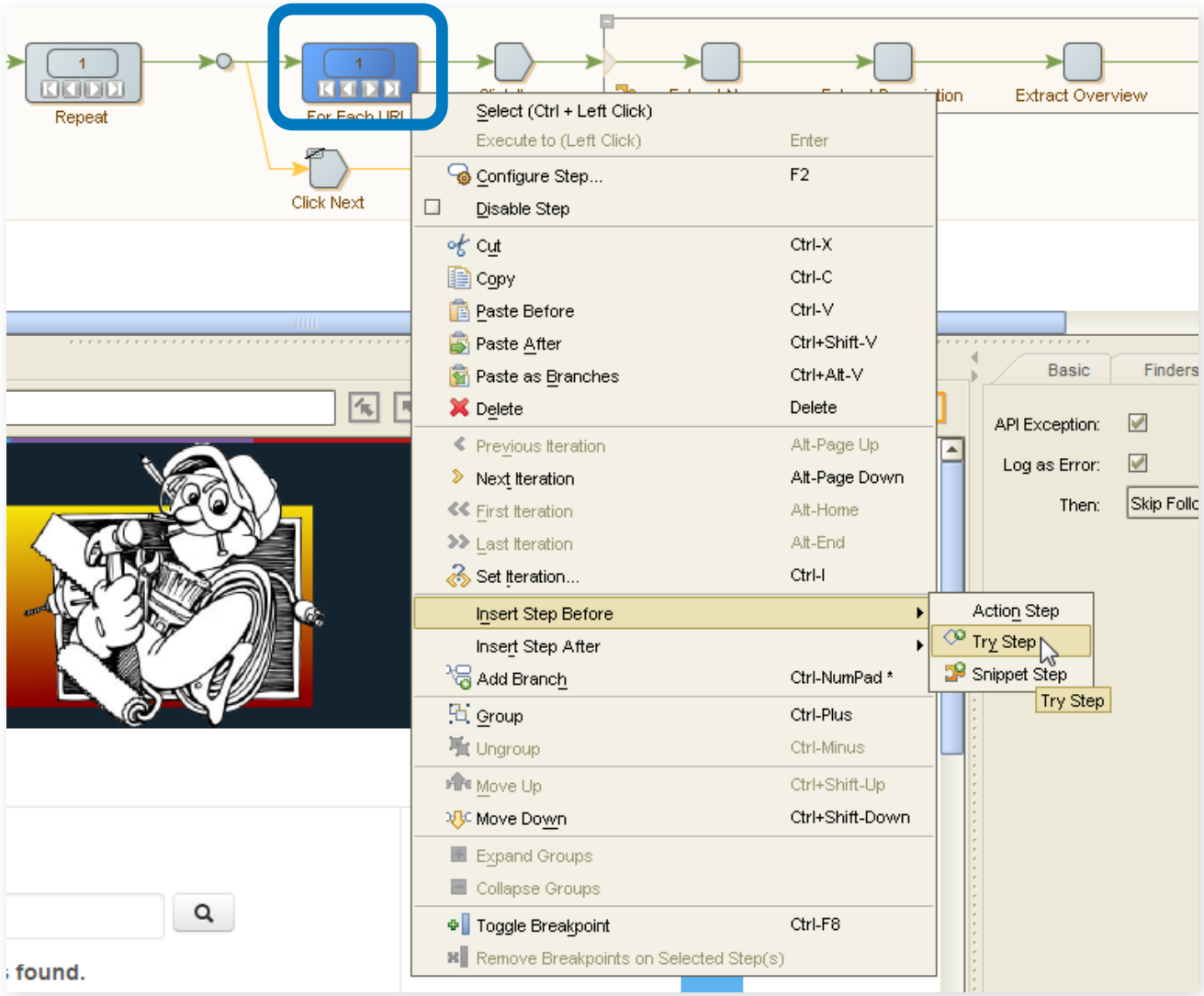
But What Happens if an Item is Not Found?

- Like a search instead for "kangaroos." An API Exception is thrown. Instead, let's say we want to return a more user-friendly, descriptive error and stop the robot from running further.

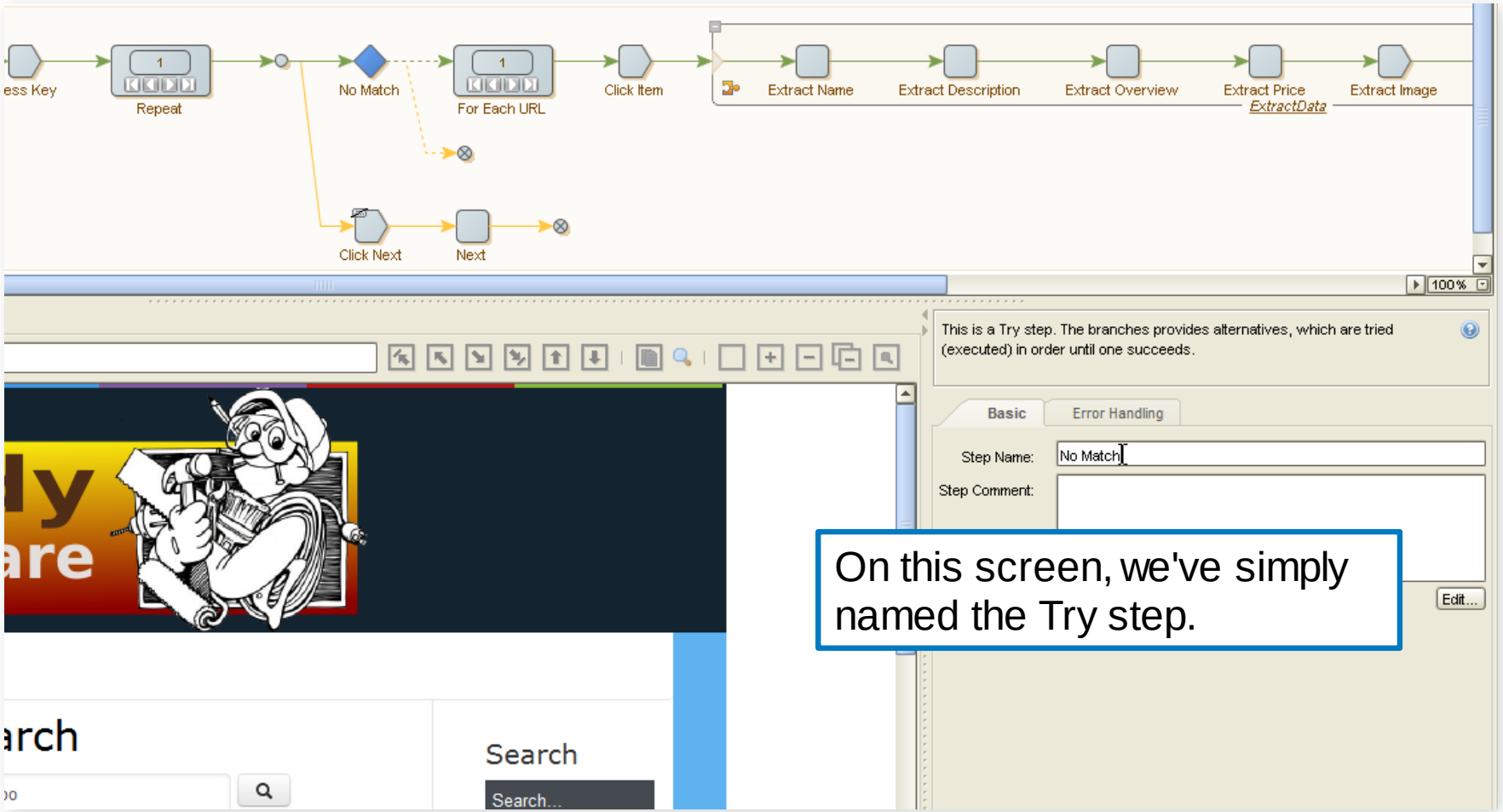


We'll Add a Try Step

- The Try step is used when it is necessary to try several alternative approaches to get a particular thing done.
- The Try step is similar to a branch point because it may have several branches going out from it.
- It differs from a branch point because branches beyond the first one are executed only if a step on the preceding branch activates the error handling option.

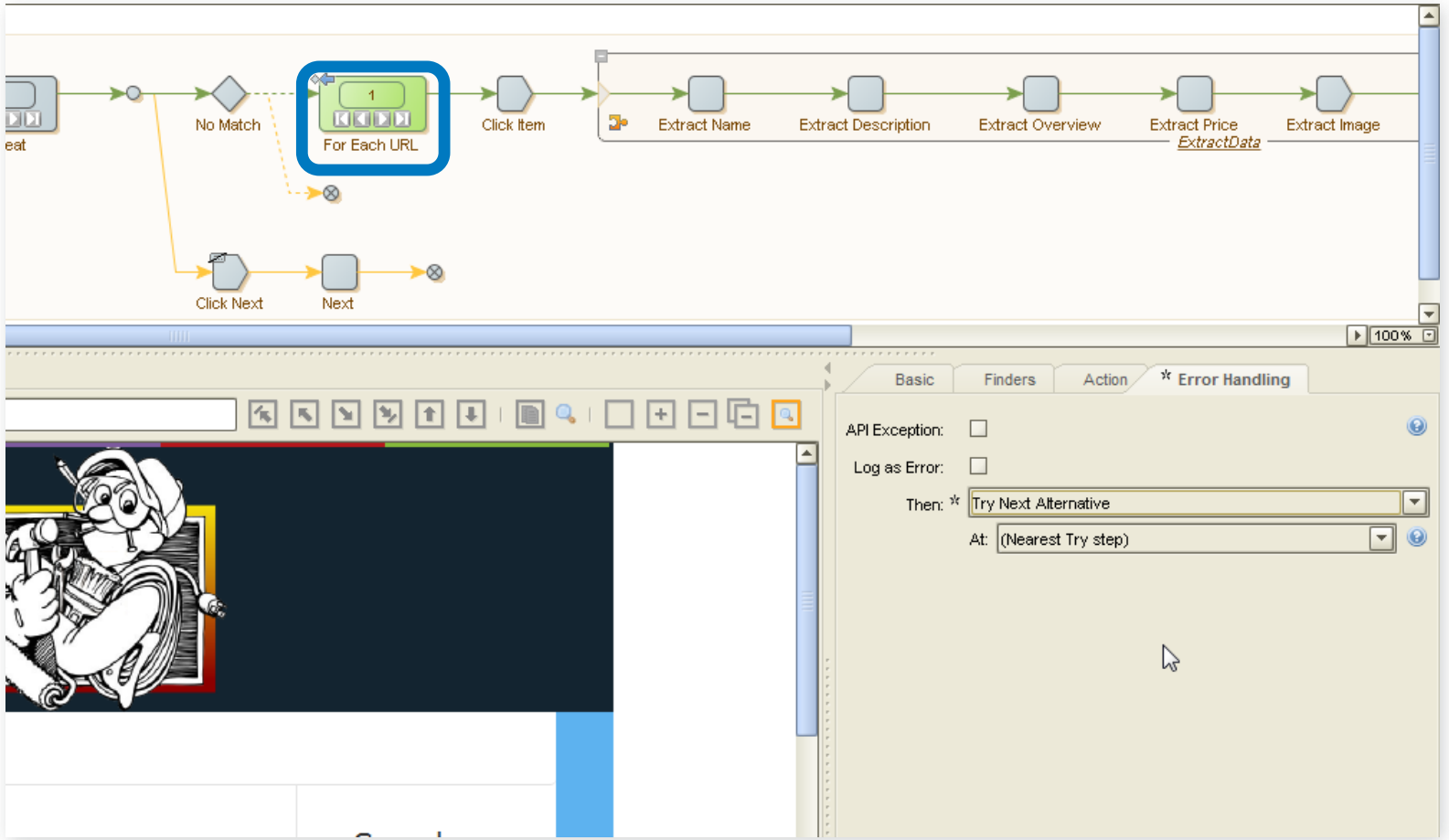


If the Try Step Fails, We'll Go to the Next Branch



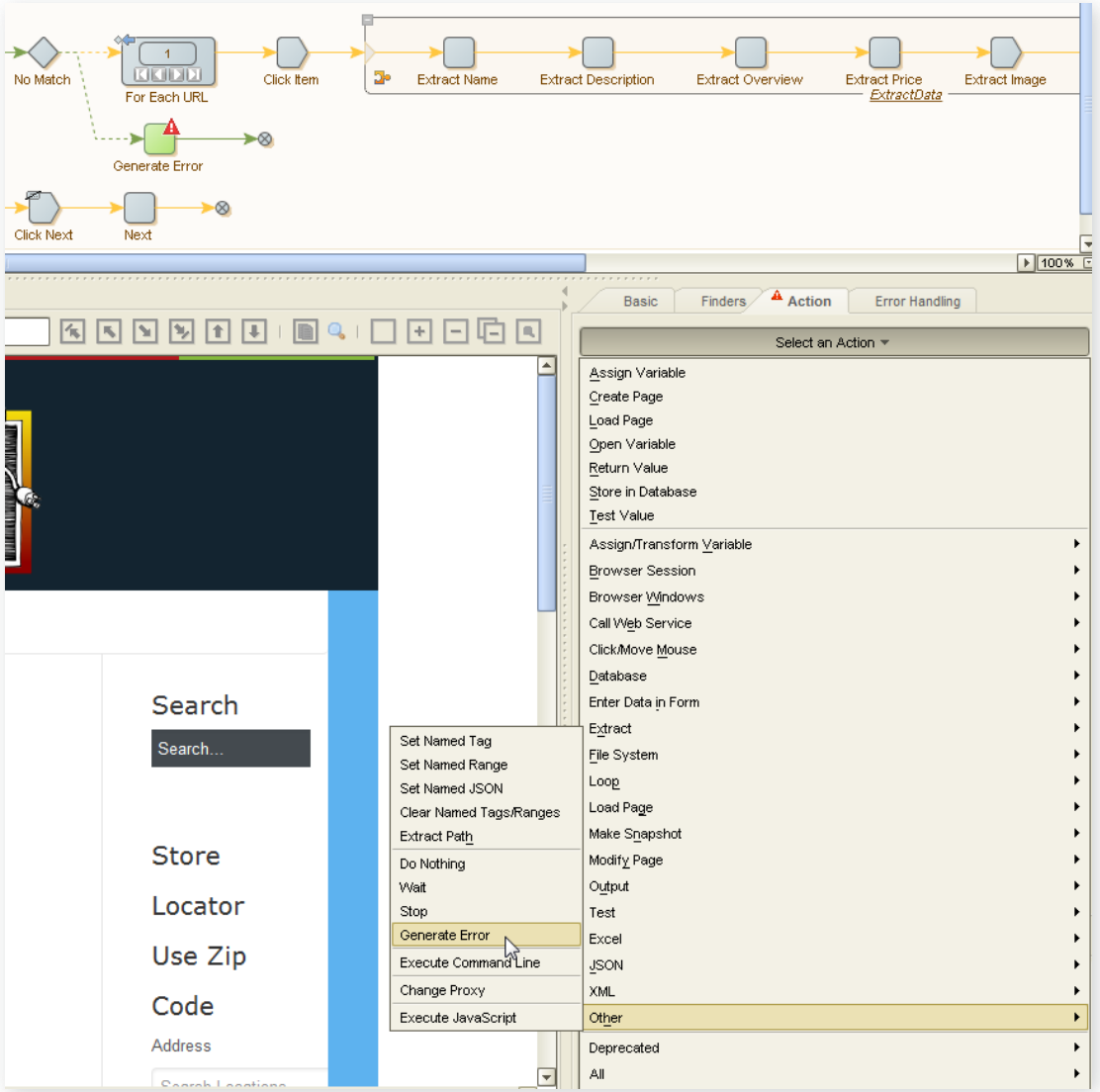
If We Tell the Robot to Go to the Next Alternative

- And then on the For Each URL step, we change the Error Handling to "Try the Next Alternative. This will try the next branch immediately preceding this step.

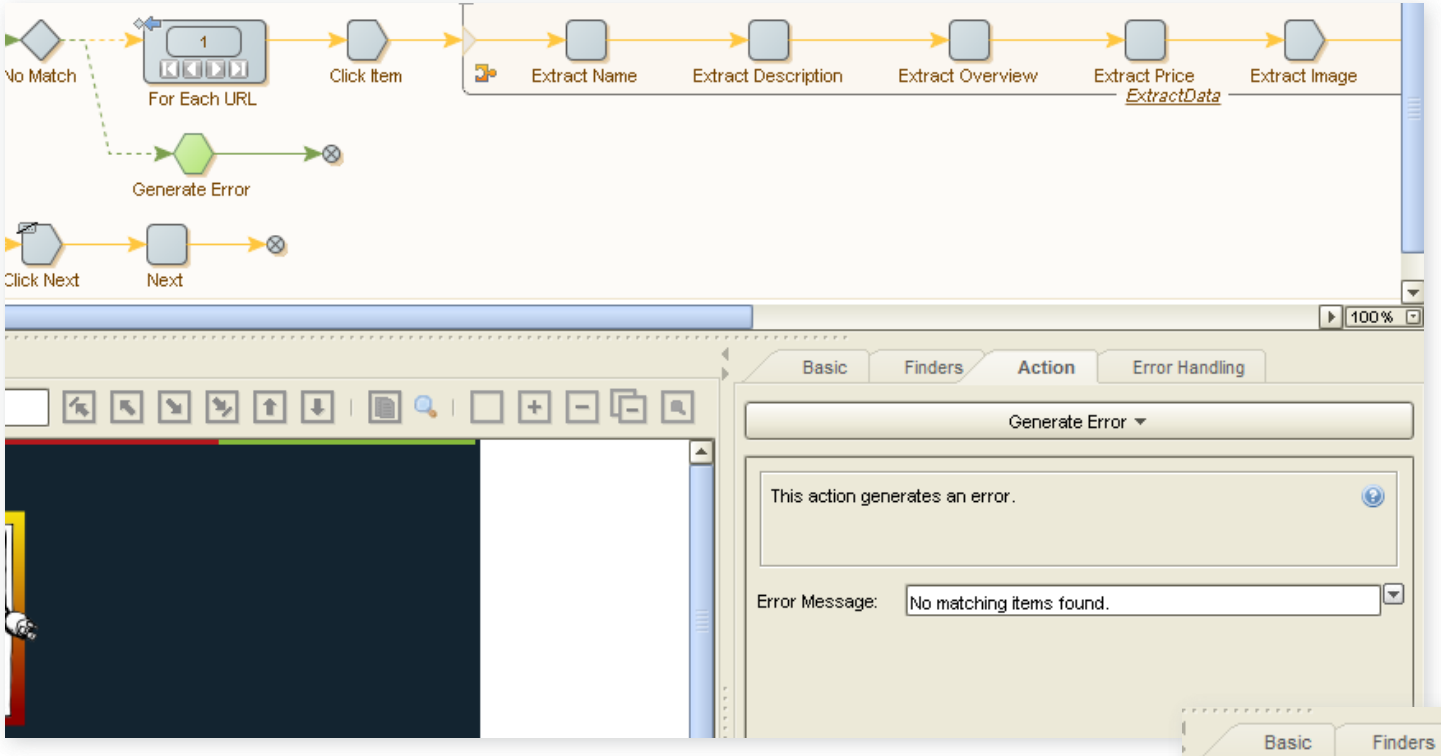


And Generate an Error that is Logged

- On the alternative branch, we'll add a new action step: this one will generate an error.
- So if there is no URL found, the Robot goes to the next alternative and intentionally generates an error that we control.

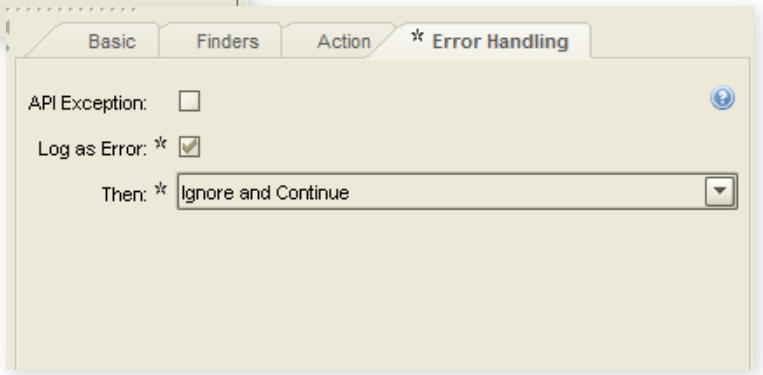


You Choose the Error Message



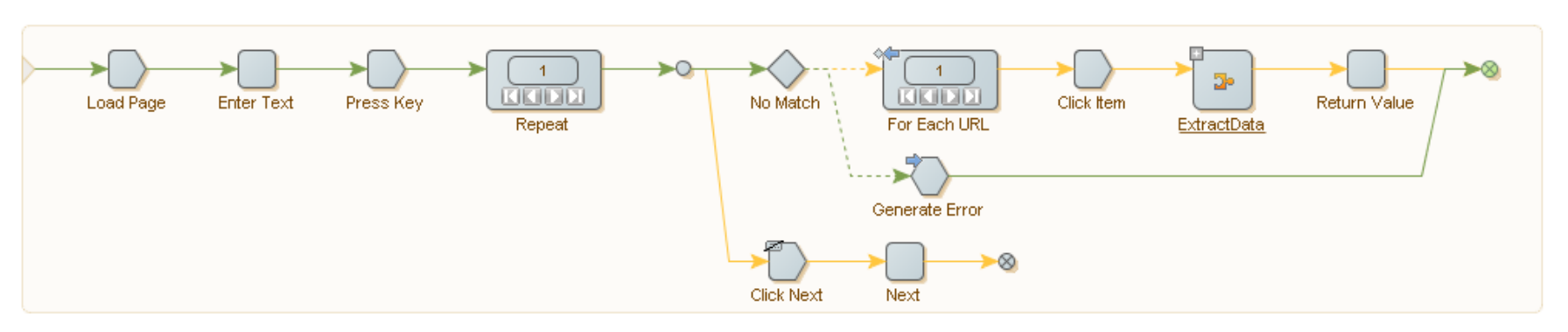
We set the error message to something like: "No matching items found."

And tell our Robot that if there is an error here, simply ignore it and continue. Notice we've also deselected the API Exception and we've told the Robot to Log as Error.



And Go to the End Step

- You can take the end point on this Try step branch and drag it with your mouse to the end step above. So the Robot will log the error and simply go to the end.
- Nice and clean!



Test and Save

- Test one more time in Debug mode.
- Try more than one input variable.
- If test successful, make sure to save your Robot
- We'll be adding to this in the next module.

Demo & Lab

Input Variables
Repeat/Next Loops