

Kofax Kapow 10.3 Training and Certification

## Module 8 – Input Variables/More Loops

Using a variable for input  
Repeat/Next Loops

**Kofax  
Kapow™**



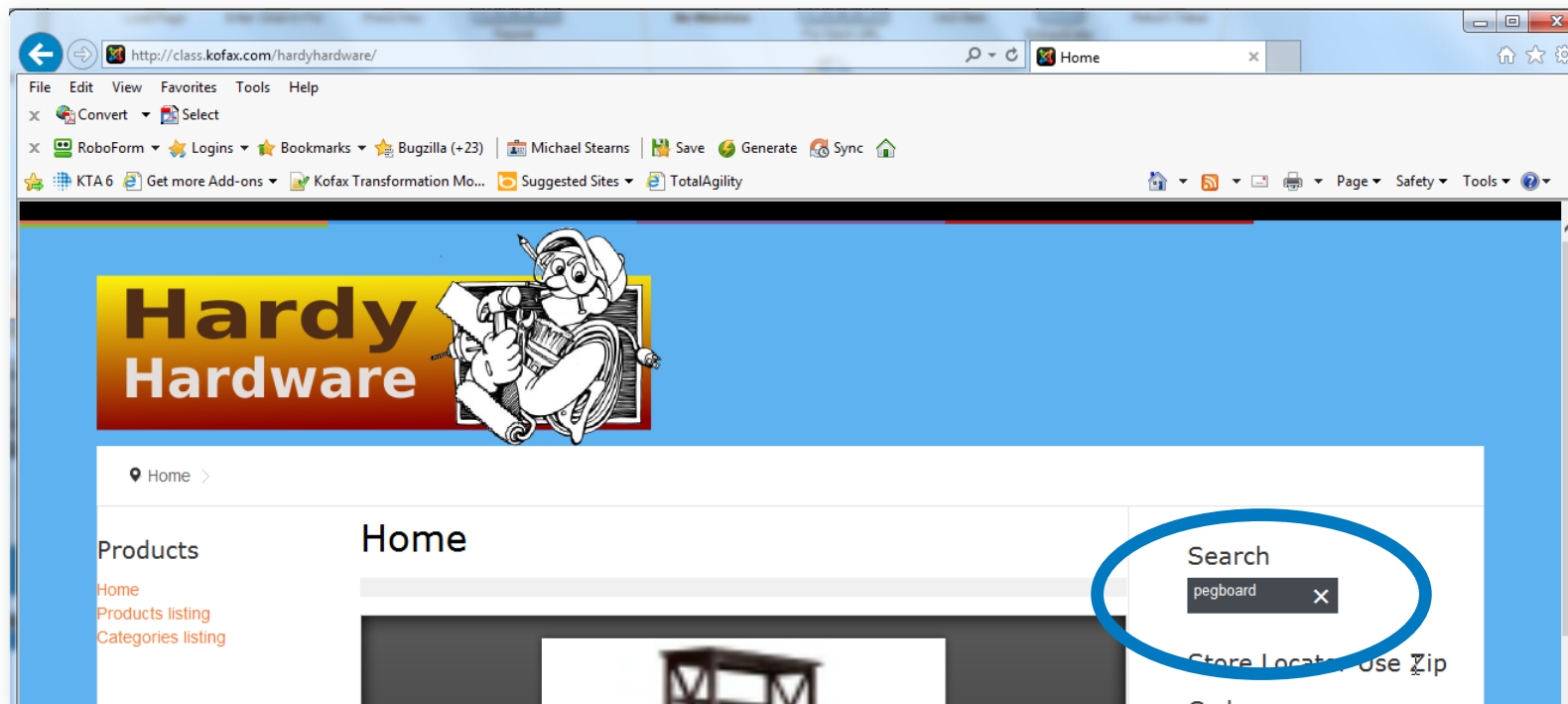
# Kofax Kapow Module Overview



- ◆ 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

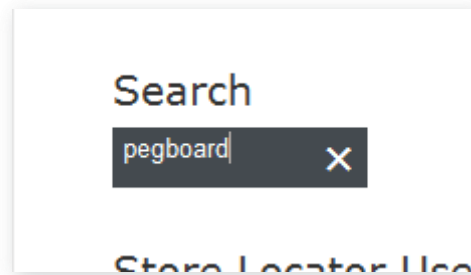
# What We Want to Do

- ◆ 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.

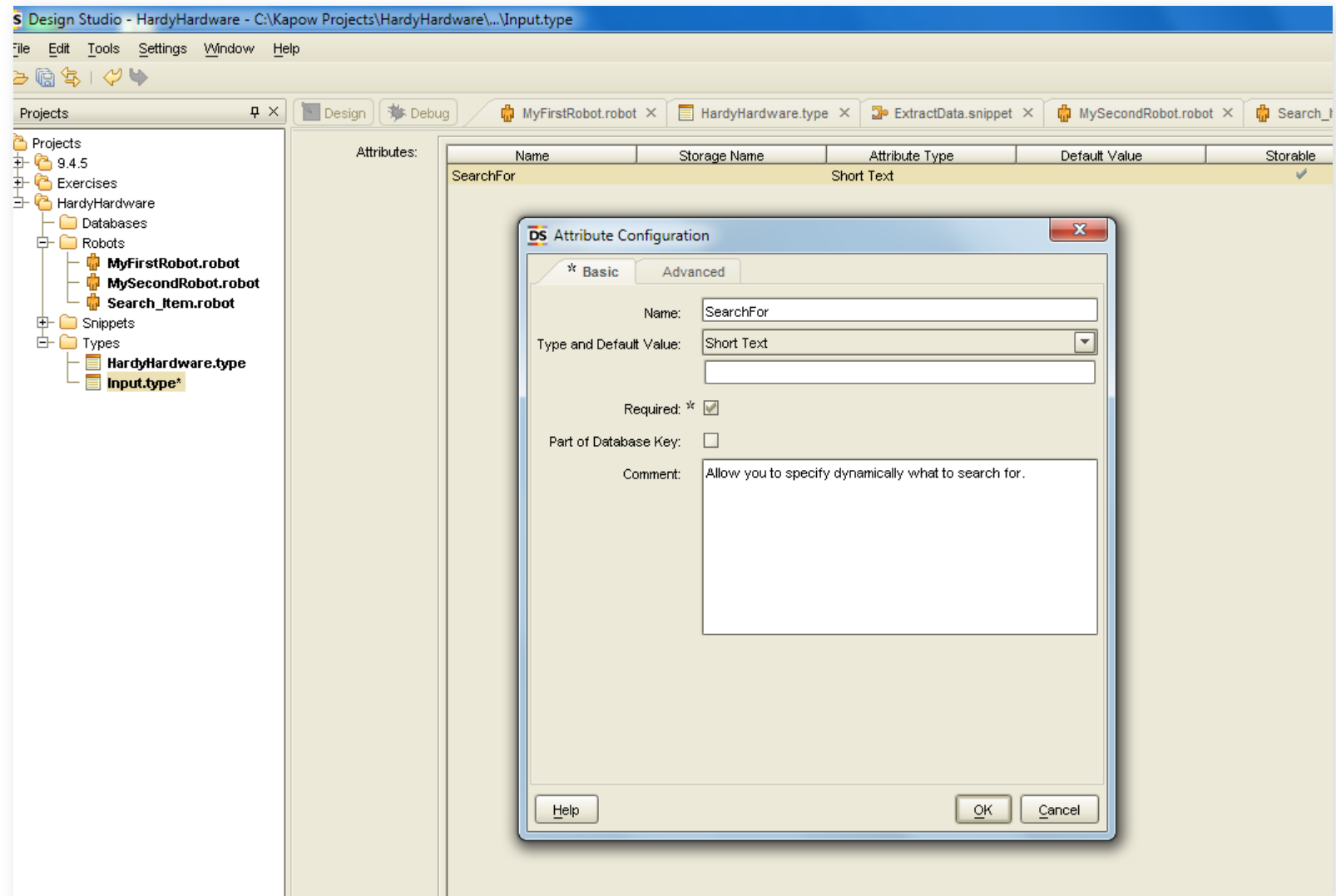


## A Little Bit about Variables

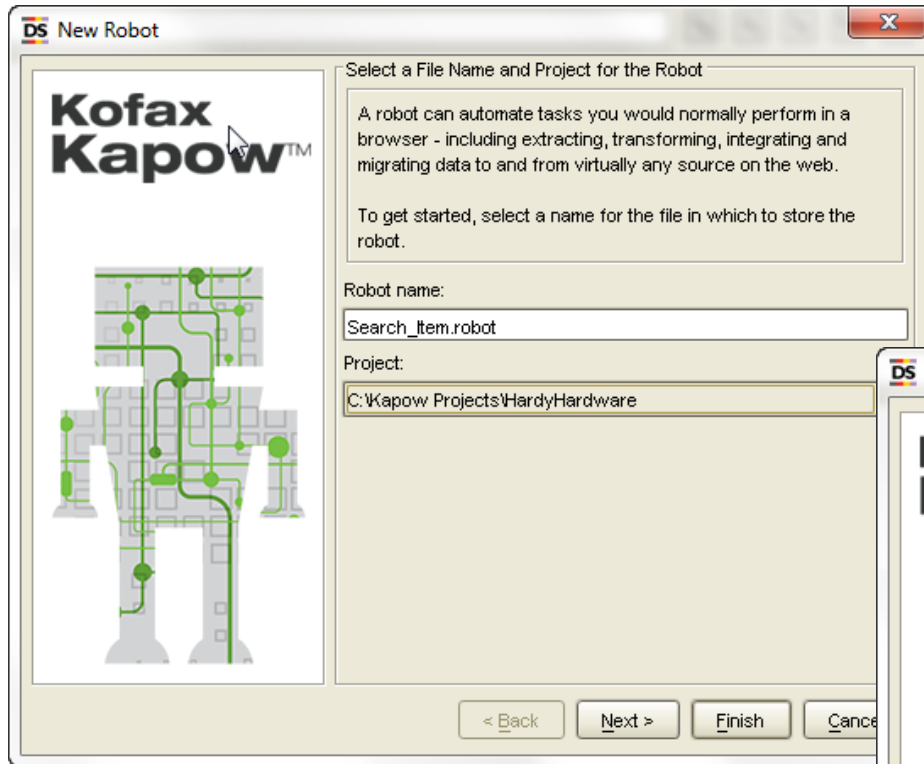
- ◆ Every variable can be associated with a default initial value that it retains unless the robot explicitly reassigns it, which it often will as values are extracted and manipulated during the execution. Most robots output the values of variables, e.g. by returning them to the caller or inserting them 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."

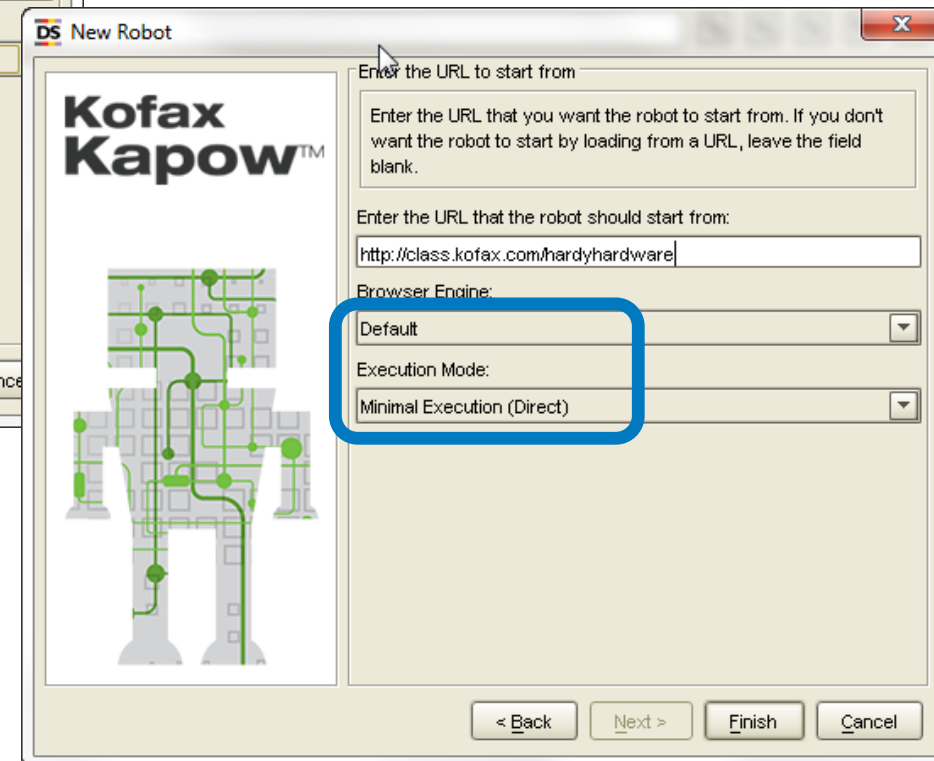


# Create a New Robot



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

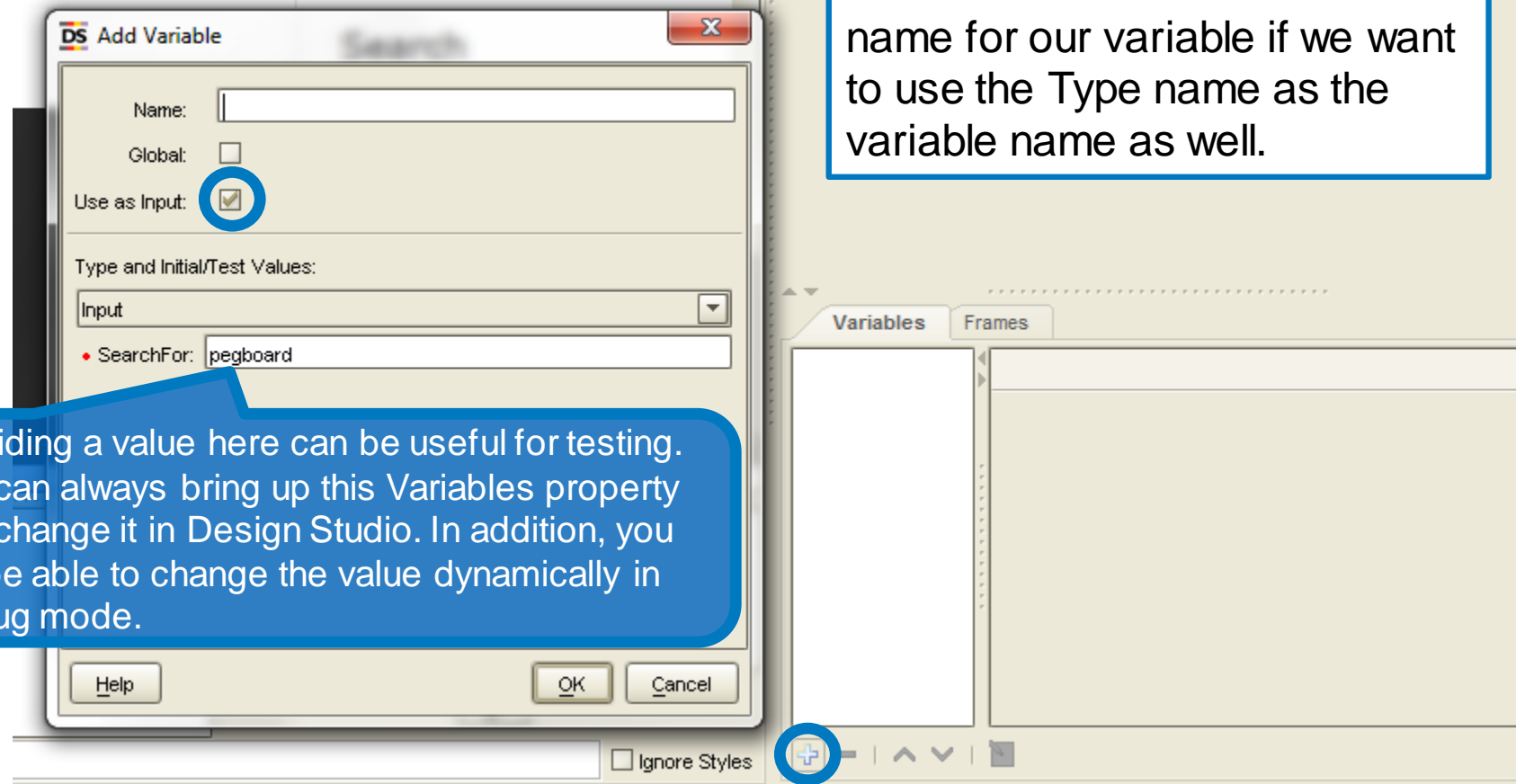
For the reasons discussed in an earlier module, the Default engine and Minimal Execution are going to be our choices here as well.



# 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

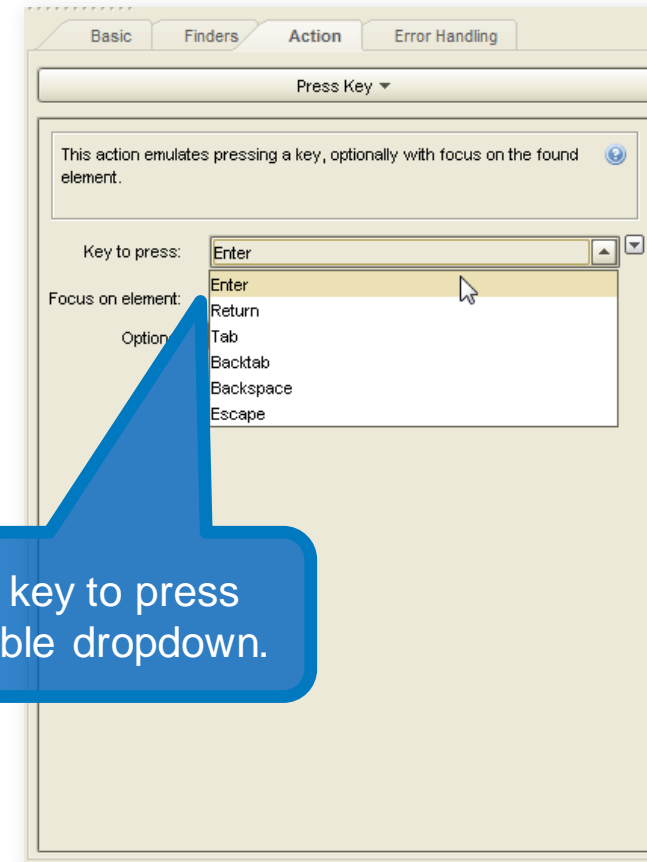
- ◆ The Enter Text Step replicates the input a human operator would provide to do a lookup of specific items

The sequence of screenshots illustrates the setup of an 'Enter Text' action step:

- Screenshot 1:** A context menu is open over a search input field. The 'Enter Text' option is highlighted. The menu also includes options like 'Enter Text from Variable', 'Press Key', 'Click', and various keyboard shortcuts.
- Screenshot 2:** The 'Tag Finder' configuration window. It shows the 'Tag Path' set to 'input' and the 'Text' field containing 'mod-search-searchword'. The 'Find Where' is set to 'Anywhere In Page'.
- Screenshot 3:** The 'Enter Text' action configuration window. The 'Text to Enter' field is set to 'Input.SearchFor'. A blue callout bubble points to this field with the text: 'This maps the value entered to the Input Variable we set up.' The 'Obtain focus by' is set to 'Sending a Focus Event' and 'Select all text before typing' is checked.
- Screenshot 4:** The 'Error Handling' tab of the configuration window. It shows 'API Exception' and 'Log as Error' both checked, and the 'Then' action set to 'Skip Following Steps'.

# 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.



Select which key to press from the available dropdown.

# 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 are many results on multiple lines, and on multiple pages.
- ◆ Note there is a [Next] button at the bottom of the page.

The screenshot shows a web browser window with the URL `class.kofax.com/hardhardware/index.php/component/search/?searchword=pipes&searchphrase=all&Itemid=101`. The search results are displayed as a list of products, each with an orange hyperlink. The products listed are:

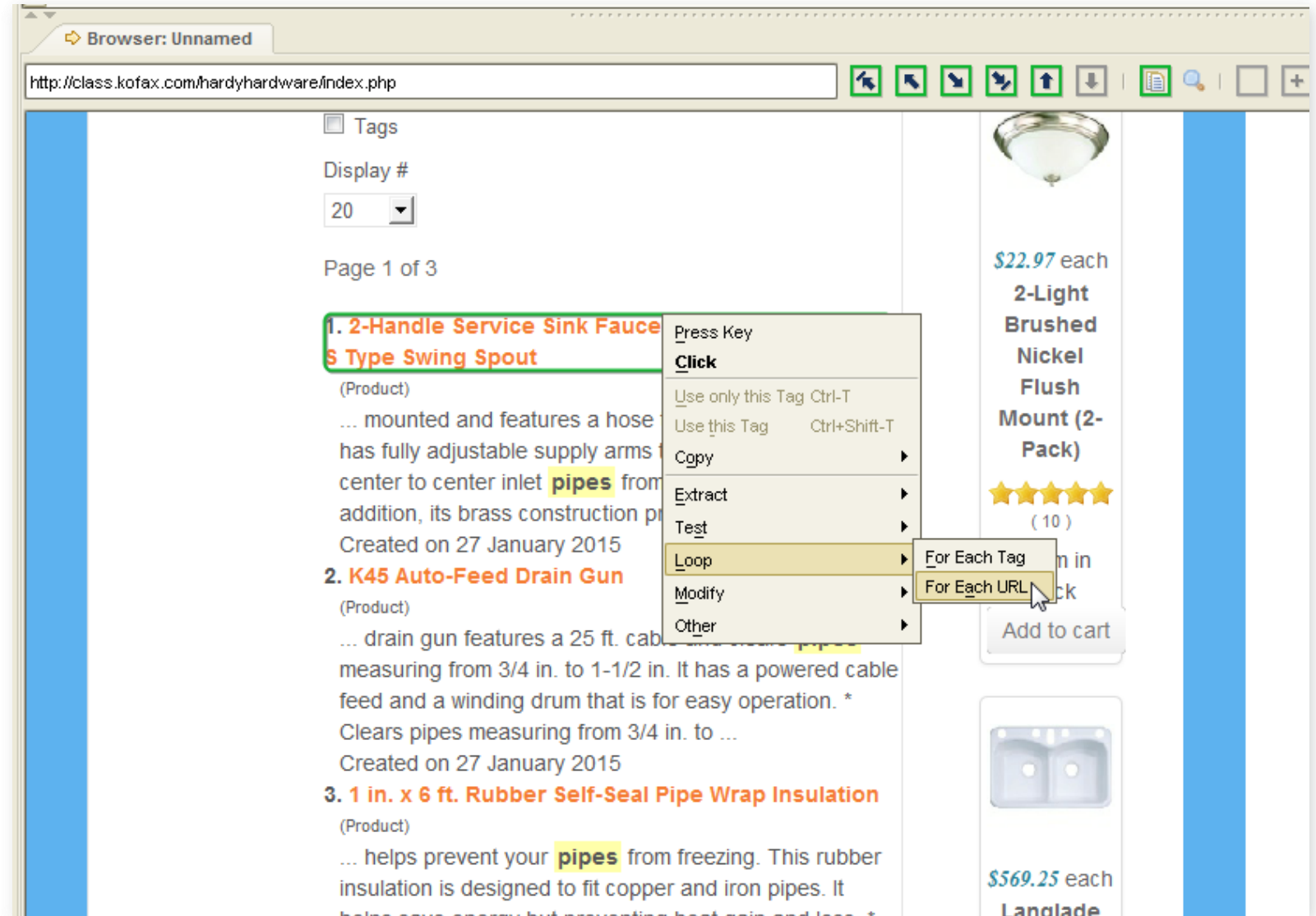
- 16. 2 in. x 3 ft. Fiberglass Self-Sealing Pre-Slit Pipe Cover (Product) ... Self-Sealing Pre-Slit Pipe Cover features a pre-slit tube with a self-sealing joint. Designed for use where foam pipe insulation is not appropriate, this pipe cover fits 2 in. pipe. Use for pipes exiting ... Created on 27 January 2015
- 17. 1 gal. 10-Minute Hair Clog Remover (Case of 4) (Product) SKU 875521027 Aisle NA Bay NA Description ZEP 1 gal. 10-Minute Hair Clog Remover (Case of 4) Product Overview ZEP 1 Gallon Hair Clog Remover Gel works fast to unclog your pipes of hair and ... Created on 27 January 2015
- 18. Quick Release 1-1/8 in. Mini Tube Cutter (Product) ... in. pipes Specifications Type Ratcheting Manufacturer Warranty Guaranteed Forever Assembled Height (in.) 5.94 in Maximum pipe or tube diameter (in.) 1.2 Minimum ... Created on 27 January 2015
- 19. 1 in. x 6 ft. Foam Pipe Insulation (Product) SKU 865170786 Aisle 12 Bay 012 Description Everbilt 1 in. x 6 ft. Foam Pipe Insulation Product Overview For an economical and energy efficient way to help prevent your pipes from freezing ... Created on 27 January 2015
- 20. 42 oz. Drain Max Gel Clog Remover (Case of 8) (Product) ... cuts through free-standing water and dissolves clog sources to re-open your drain and pipes \* Safe for all types of pipe \* Garbage disposal safe \* About 15 minute set time Easy-pour spout Specifications ... Created on 27 January 2015

At the bottom of the page, there are pagination controls: `START`, `PREV`, `1`, `2`, `3`, `NEXT`, and `END`. The `2` and `3` are highlighted in orange. A blue speech bubble points to the `NEXT` button, stating: "There are multiple pages. We will want to loop through those as well."

Another blue speech bubble points to the orange hyperlinks in the product titles, stating: "Each item in orange here is a hyperlink to a new URL with the product details we want. We will want to loop through those."

# 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

Notice clicking on the forward arrow and advancing the counter to "2" goes to the second link below.

Workflow Diagram:

- Load Page
- Enter Text
- Press Key
- Each URL (Counter: 2)

Web Page Content:

re/index.php

▼ vweblinks

Tags

Display #

20

Page 1 of 3

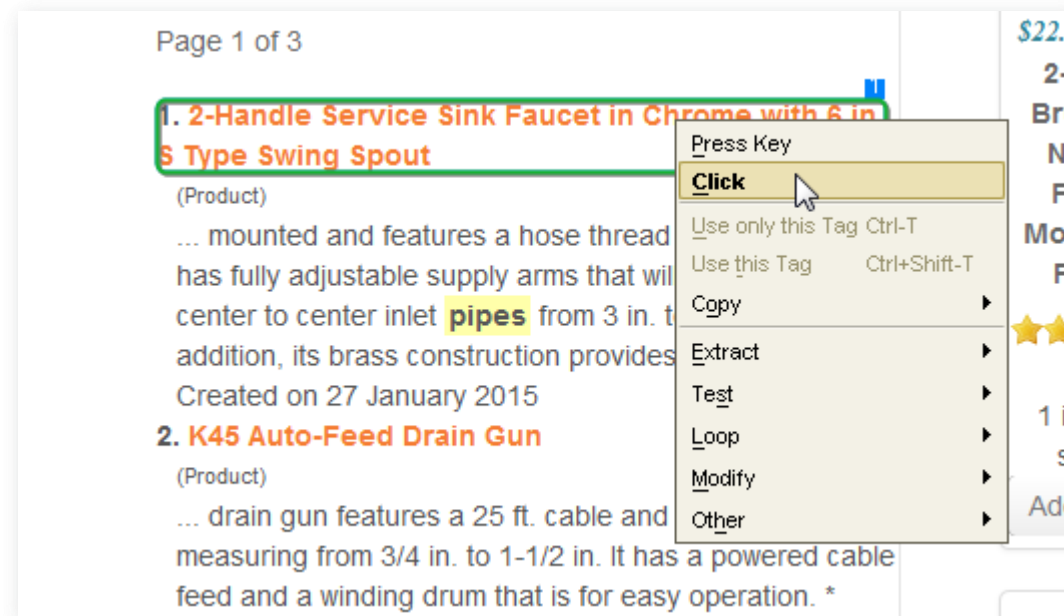
1. **2-Handle Service Sink Faucet in Chrome with 6 in. S Type Swing Spout**  
(Product)  
... mounted and features a hose thread end. The fitting has fully adjustable supply arms that will accommodate center to center inlet pipes from 3 in. to 8-3/8 in. In addition, its brass construction provides ...  
Created on 27 January 2015

2. **K45 Auto-Feed Drain Gun**  
(Product)  
... drain gun features a 25 ft. cable and clears pipes measuring from 3/4 in. to 1-1/2 in. It has a powered cable.

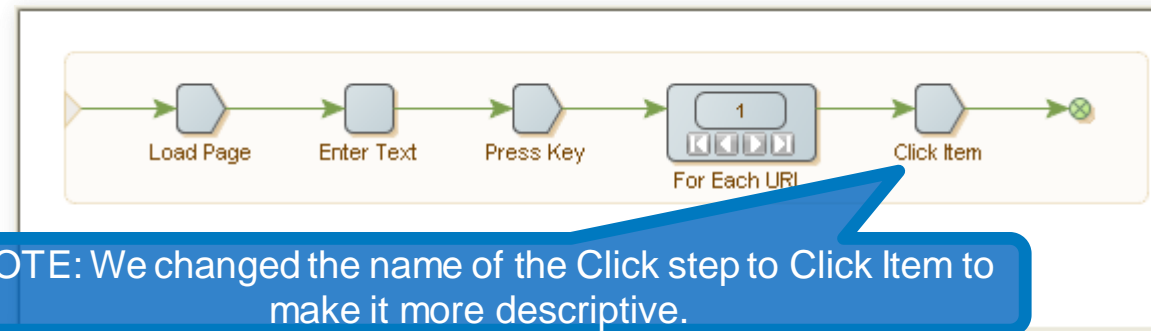
Product Details (Right):

- \$22.97 each
- 2-Light Brushed Nickel Flush Mount (2-Pack)
- ★★★★★ (10)
- 1 item in stock
- Add to cart

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

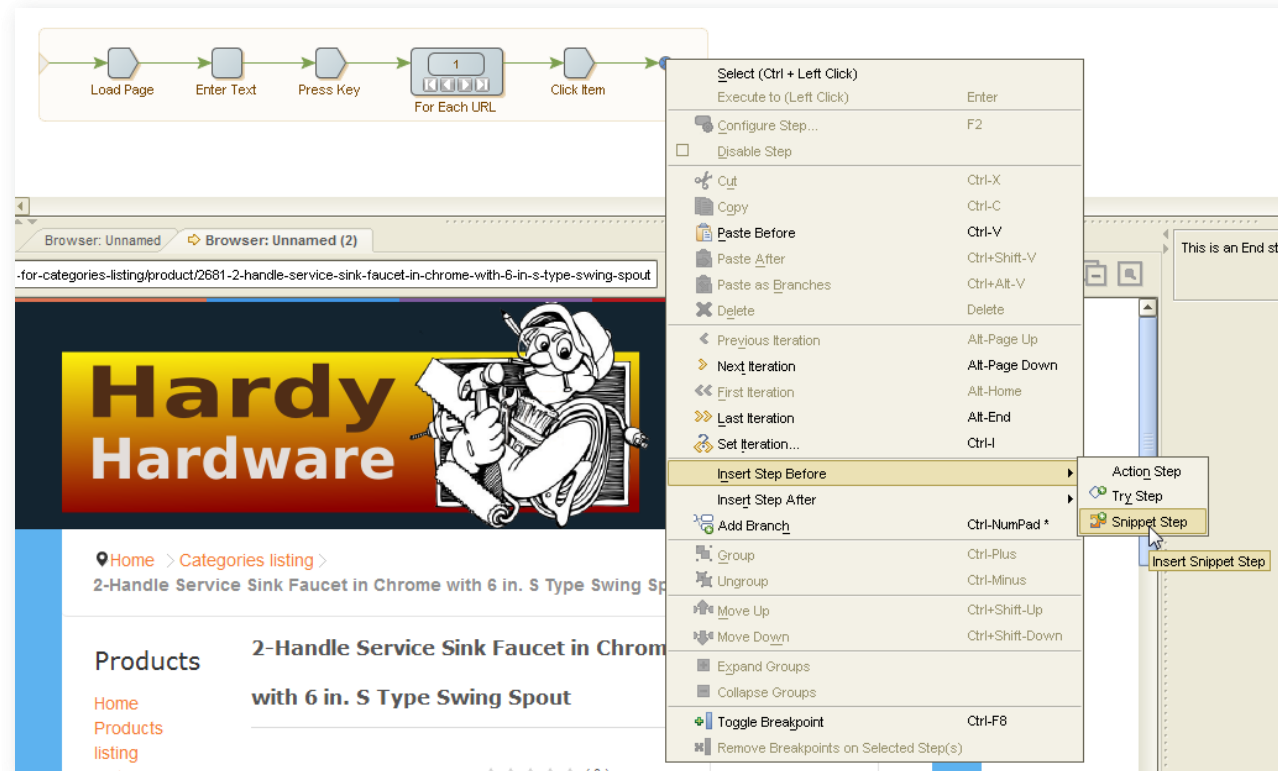


And our robot looks like this so far...



# 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.


The screenshot displays a web automation workflow at the top and a product page for a faucet below. The workflow includes steps: Load Page, Enter Text, Press Key, For Each URL (highlighted with a blue box and labeled 'Loop'), Click Item, ExtractData, and Return Value. A blue arrow indicates the loop structure. The product page shows details for a '2-Handle Service Sink Faucet in Chrome with 6 in. S Type Swing Spout', including its price (\$224.46), stock status (1 item in stock), and a description. A 'Variables' panel on the right shows the 'HardyHardware' variable populated with product data from the 'ExtractData' step.

Loop

Variables

Input

HardyHardware (Snippet: ExtractDat Type: HardyHardware)

- Name: 2-Handle Service Sink Faucet in Chrome with 6 in. S Type Swing Spout
- Description: Chicago Faucets 2-Handle Service Sink Faucet in Chrome with 6 in. S Type Swing Spout
- Overview: This Chicago Faucets Sink fitting is wall mounted and features a hose thread end. The fitting has fully adjustable supply arms that will accommodate center to center inlet pipes from 3 in. to 6 in.
- SKU: 604158303
- Aisle: NA
- Bay: NA
- Picture: 
- Price: 224.46

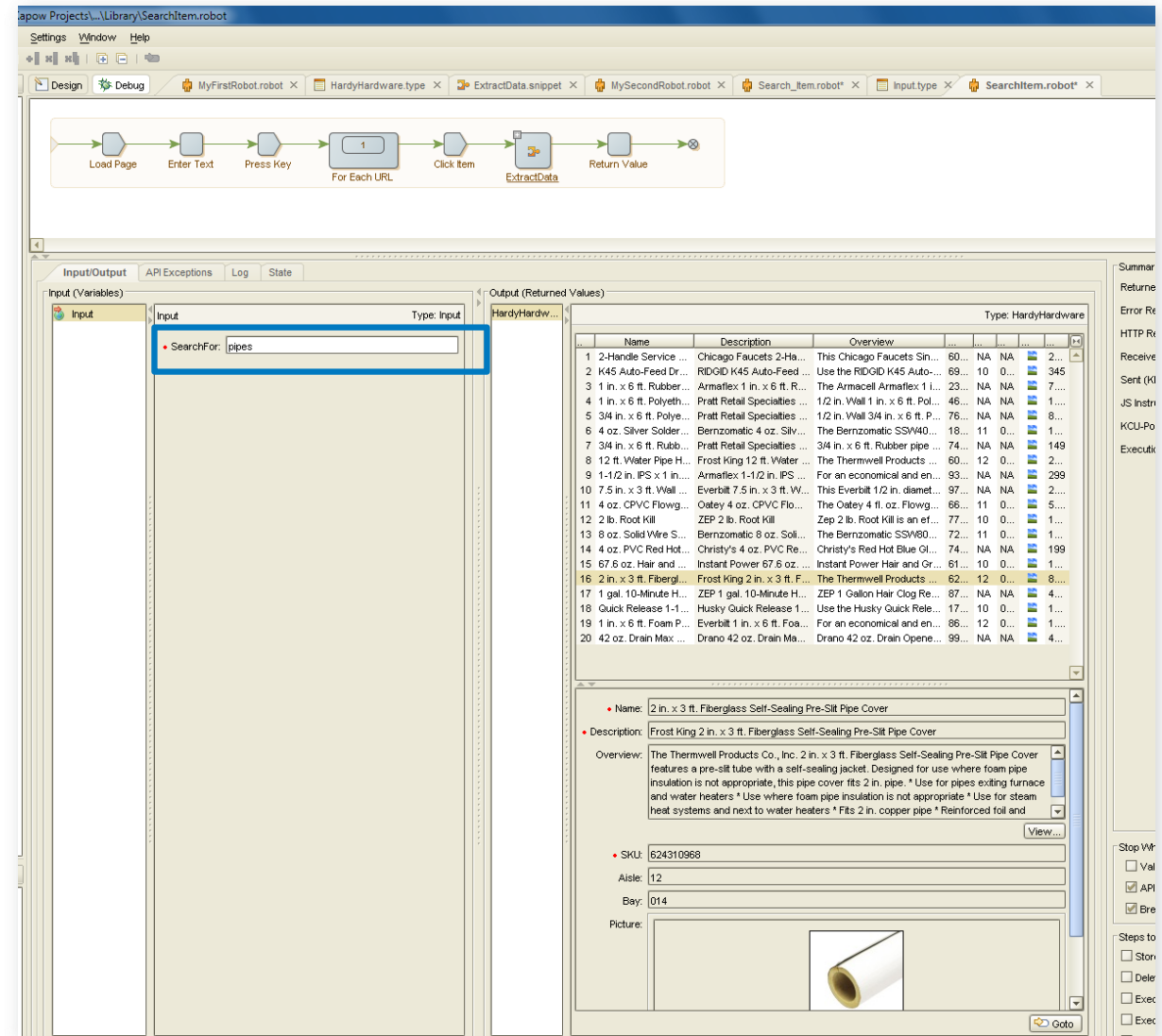
Loops will always go through one iteration, start to finish, and then to the second, repeat the steps, and so on.

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



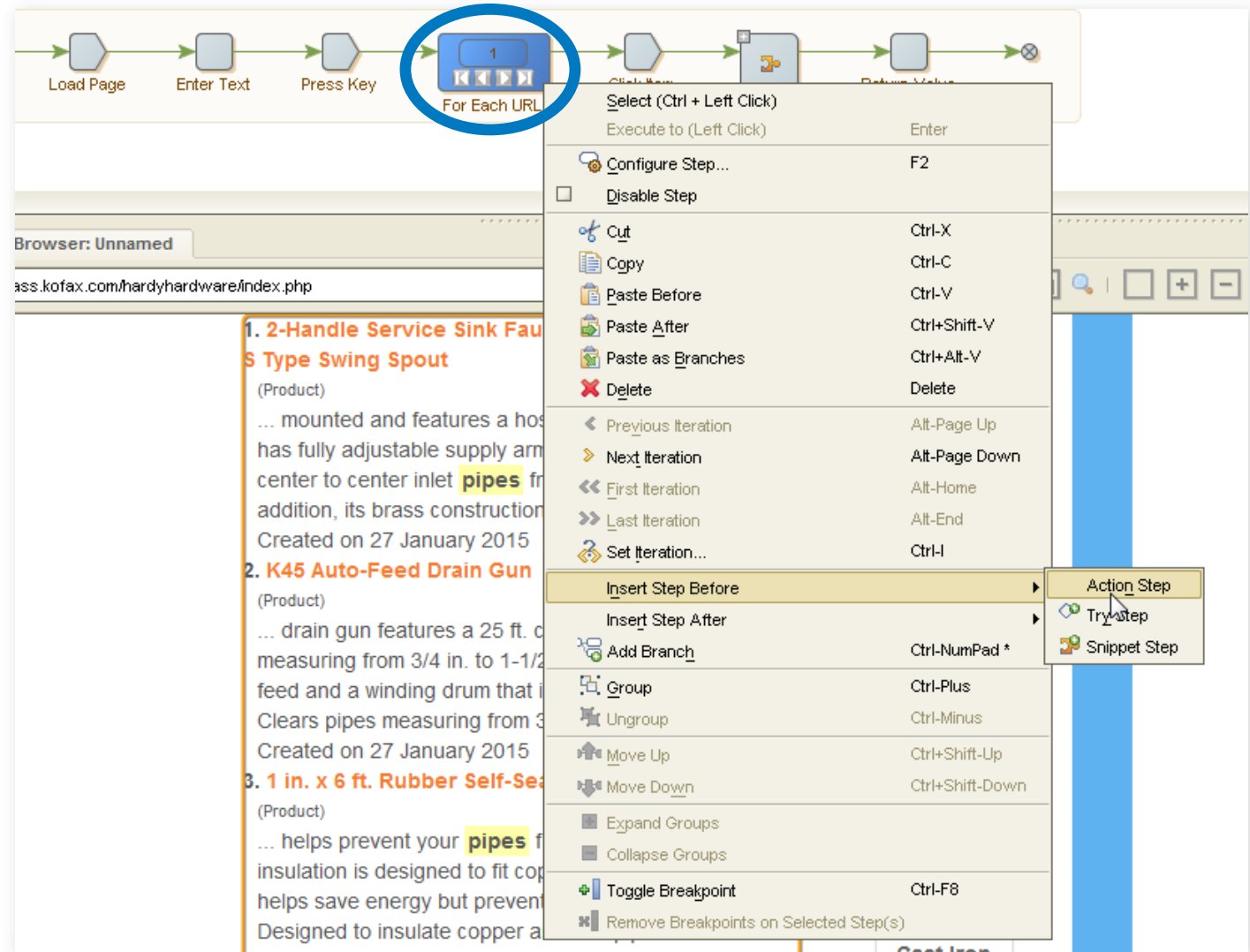
# And Test in Debug Mode

- ◆ 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.



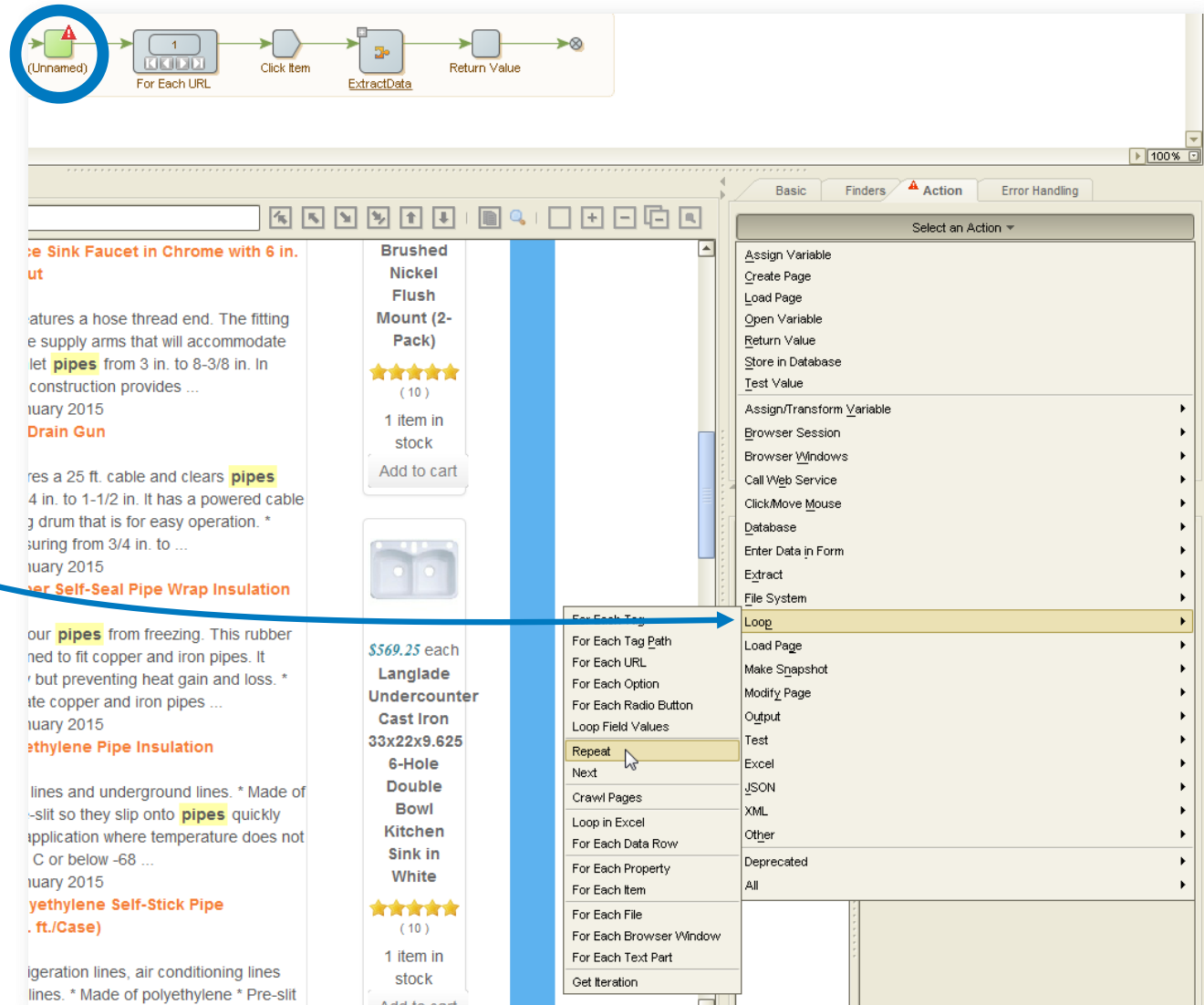
# Loop through Pages

- ◆ 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

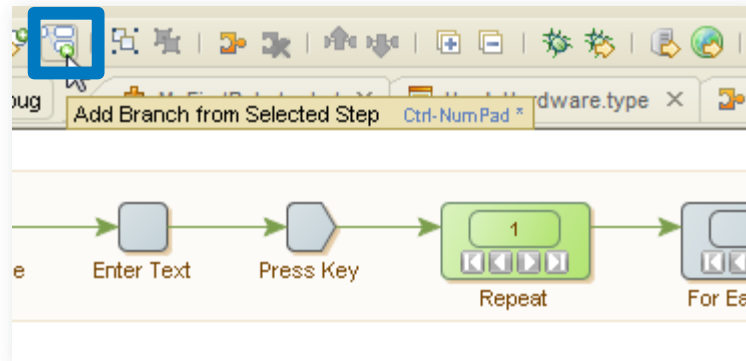
- ◆ So we'll select the new unnamed step and for the action, select "Loop" and then "Repeat."



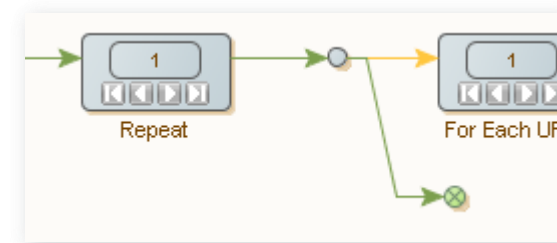
# Repeat Step Always Goes with a Next Step

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

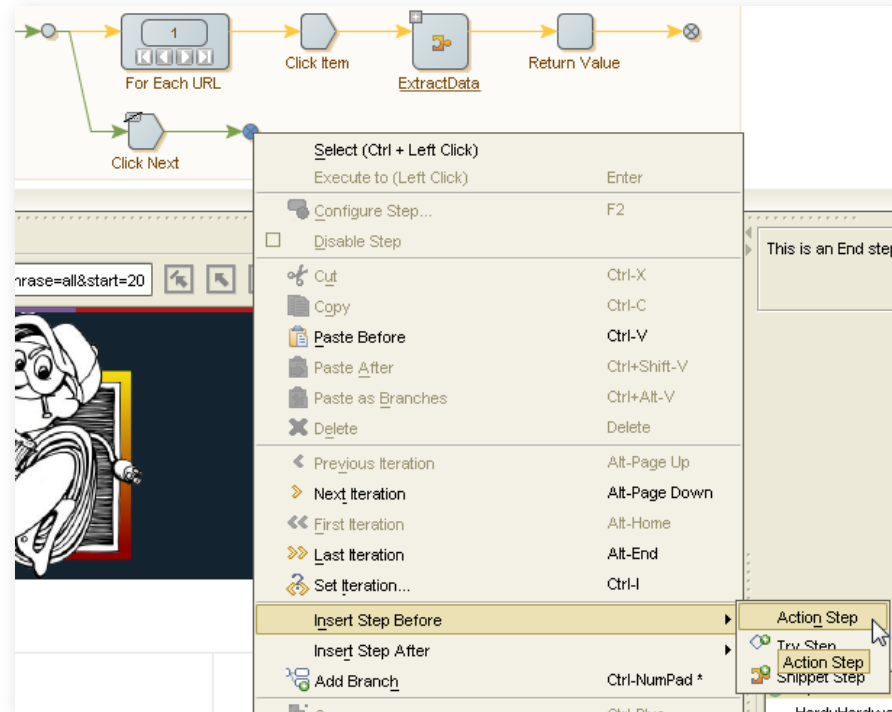
The screenshot displays the Kofax software interface. At the top, a workflow diagram shows a sequence of steps: Load Page, Enter Text, Press Key, Repeat (containing a 'For Each URL' loop), Click Item, ExtractData, and Return Value. A green arrow points from the 'Click Item' step to a browser window below. The browser window shows a product page for '20. 42 oz. Drain Max Gel Clog Remover (Case of 8)' with pagination controls at the bottom: 'START', 'PREV', '1', '2', '3', 'NEXT', 'END'. The 'NEXT' button is highlighted with a green box. A context menu is open over the 'NEXT' button, with the 'Click' option selected. To the right, the 'Error Handling' tab of the 'Set properties of step' dialog is shown, with 'Break Loop' selected in the 'Then:' list. A blue box highlights the 'Error Handling' tab. A text box on the right contains the following text:

Set properties of step.

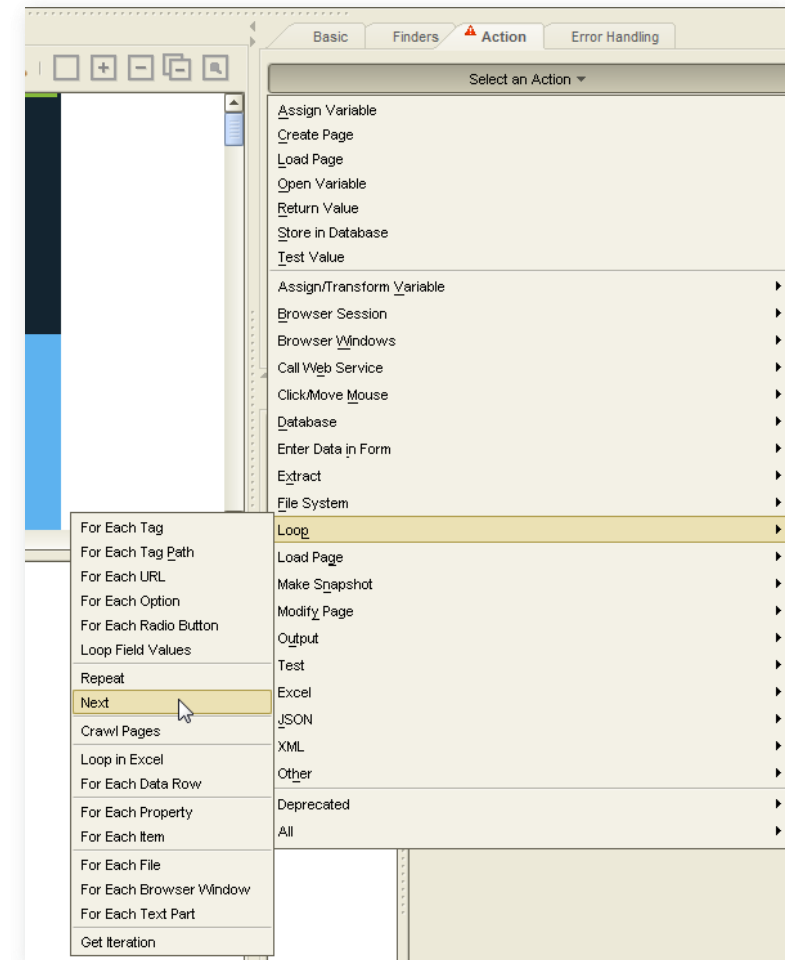
IMPORTANT: What if the Next button isn't available (as on the last page)? We don't want to report an error. We want to break the loop at that point.

# Then Add Next Step

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

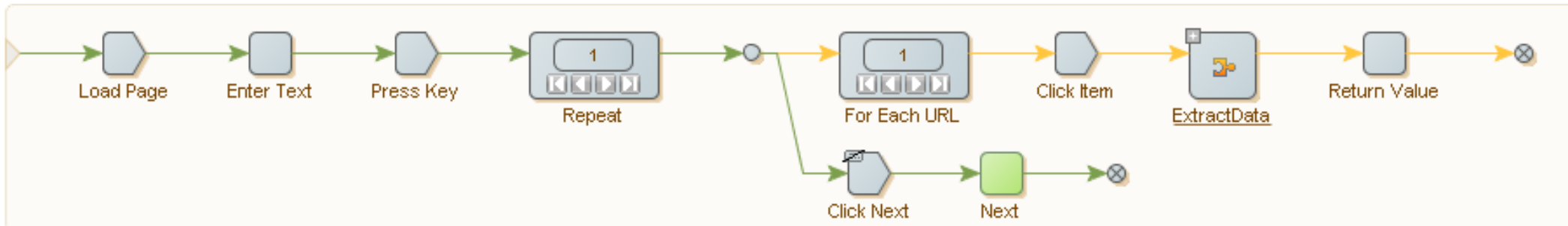


Then choose "Loop" and "Next" as the action you want to perform.



# Now Your Robot Looks Like This

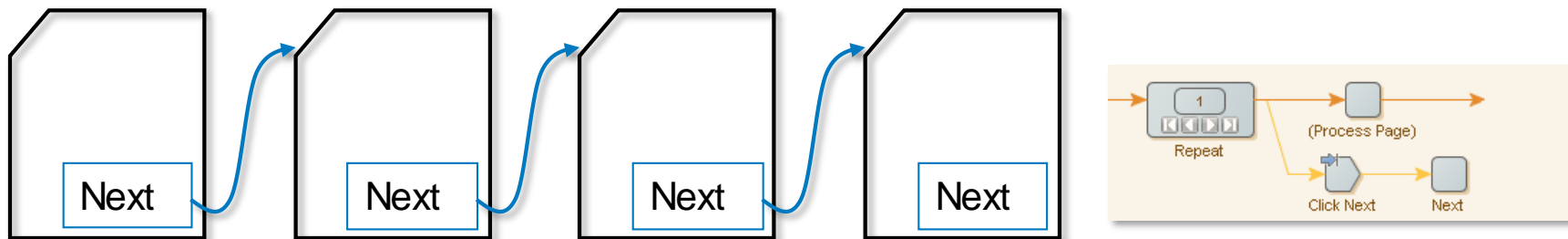
1. The main page is loaded.
2. The search text is entered.
3. The [Enter] key is pressed.
4. The first page is displayed.



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."

The screenshot shows the Design Studio interface for a robot named 'SearchItem.robot'. The main window displays a flowchart with the following steps: Load Page, Enter Text, Press Key, Repeat, For Each URL, Click Item, ExtractData, and Return Value. The 'Input' tab shows 'SearchFor: pipes'. The 'Output' tab shows a table of results from HardyHardware, including items like 'Everbilt 3/4 in. x 6 ft. Foam Pipe Insulation'. The 'Summary' tab on the right shows 'Returned Values: 213' and 'Error Reports: 0'.

#	Name	Description	Overview	S	FE	SV	...
66	Auto-Clean K-30 Sink...	RIDGID Auto-Clean K-30 Si...	Use the RIDGID Auto-Clean K...	34...	10	004	159
67	1 in. x 6 ft. Foam Self...	Everbilt 1 in. x 6 ft. Foam S...	For an economical and energ...	61...	12	012	2...
68	6 ft. Electric Water Pl...	Frost King 6 ft. Electric Wa...	This energy-saving Thermwel...	67...	12	012	2...
69	Quick Release 1-1/8 i...	Husky Quick Release 1-1/8...	Use the Husky 1/8 in. to 1-1/8...	49...	NA	NA	1...
70	30 ft. Automatic Elect...	Frost King 30 ft. Automatic...	The Thermwell 30 ft. Automati...	12...	12	012	3...
71	3/4 in. x 6 ft. Rubber ...	Armaceil 3/4 in. x 6 ft. Rub...	For an economical and energ...	22...	12	012	6...
72	128 oz. Main Line Clea...	Instant Power 128 oz. Mai...	Instant Power Main Line Clea...	44...	10	002	1...
73	2 in. x 30 ft. R-1 Foa...	Armaceil 2 in. x 30 ft. R-1 ...	Armaceil Armaflex 2 in. x 30 f...	24...	12	014	7...
74	3/4 in. x 6 ft. Foam Pi...	Everbilt 3/4 in. x 6 ft. Foam...	For an economical and energ...	47...	12	012	1...
75	1/2 in. x 6 ft. Rubber ...	Armaceil 1/2 in. x 6 ft. Rub...	The Armaceil 1/2 in. x 6 ft. Ar...	92...	12	012	5...
76	42 oz. Drain Max Gel ...	Drano 42 oz. Drain Max Ge...	Drano has your back with ma...	54...	NA	NA	4...
77	1/2 in. x 6 ft. Foam S...	Everbilt 1/2 in. x 6 ft. Foam...	For an economical and energ...	77...	12	012	2...
78	9 in. x 3 ft. Wall Pipe I...	Everbilt 9 in. x 3 ft. Wall Pip...	The Everbilt 3/4 in. Diameter 1...	47...	NA	NA	3...
79	64 oz. Hair Clog Rem...	ZEP 64 oz. Hair Clog Remo...	Zep Commercial 10-Minute Ha...	51...	10	002	6...
80	1 gal. Max Gel Clog R...	Drano 1 gal. Max Gel Clog ...	Drano has your back with ma...	57...	10	002	1...
81	2-Handle Service Sin...	Chicago Faucets 2-Handle...	This Chicago Faucets Sink fitti...	60...	NA	NA	2...
82	K45 Auto-Feed Drain ...	RIDGID K45 Auto-Feed Dra...	Use the RIDGID K45 Auto-Fee...	69...	10	004	345

Summary

Returned Values: 213

Error Reports: 0

HTTP Requests: 0

Received (KB): 0

Sent (KB): 0

JS Instructions: 0

KCU-Point Usage: 0

Execution Time (s): 0.00

Stop When

☐ Values are Returned or Stored

☒ API Exceptions are Reported

☒ Breakpoints are Reached

Steps to Skip

☐ Store in Database

☐ Delete from Database

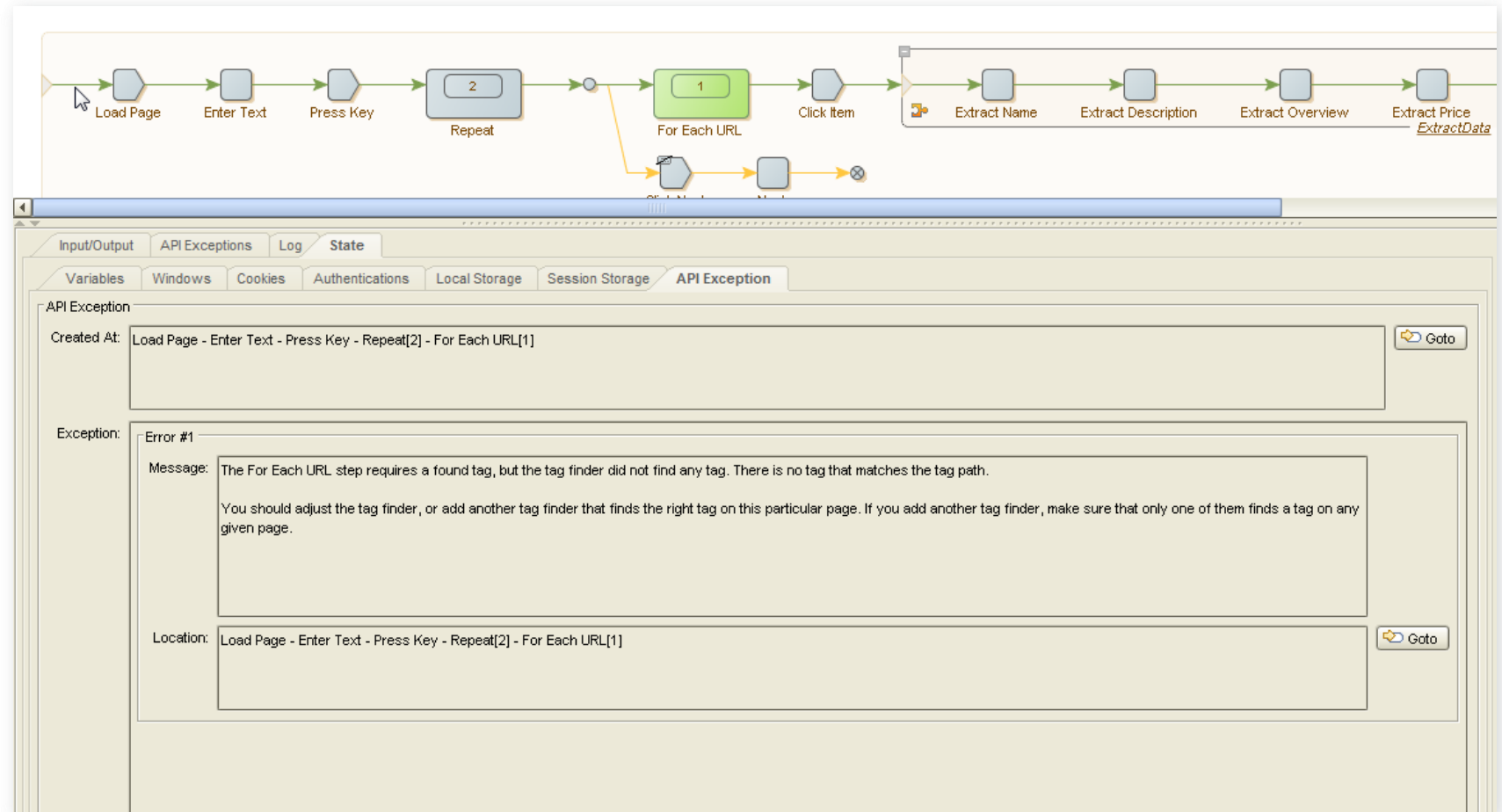
☐ Execute SQL

☐ Execute Command Line

☐ Send Email

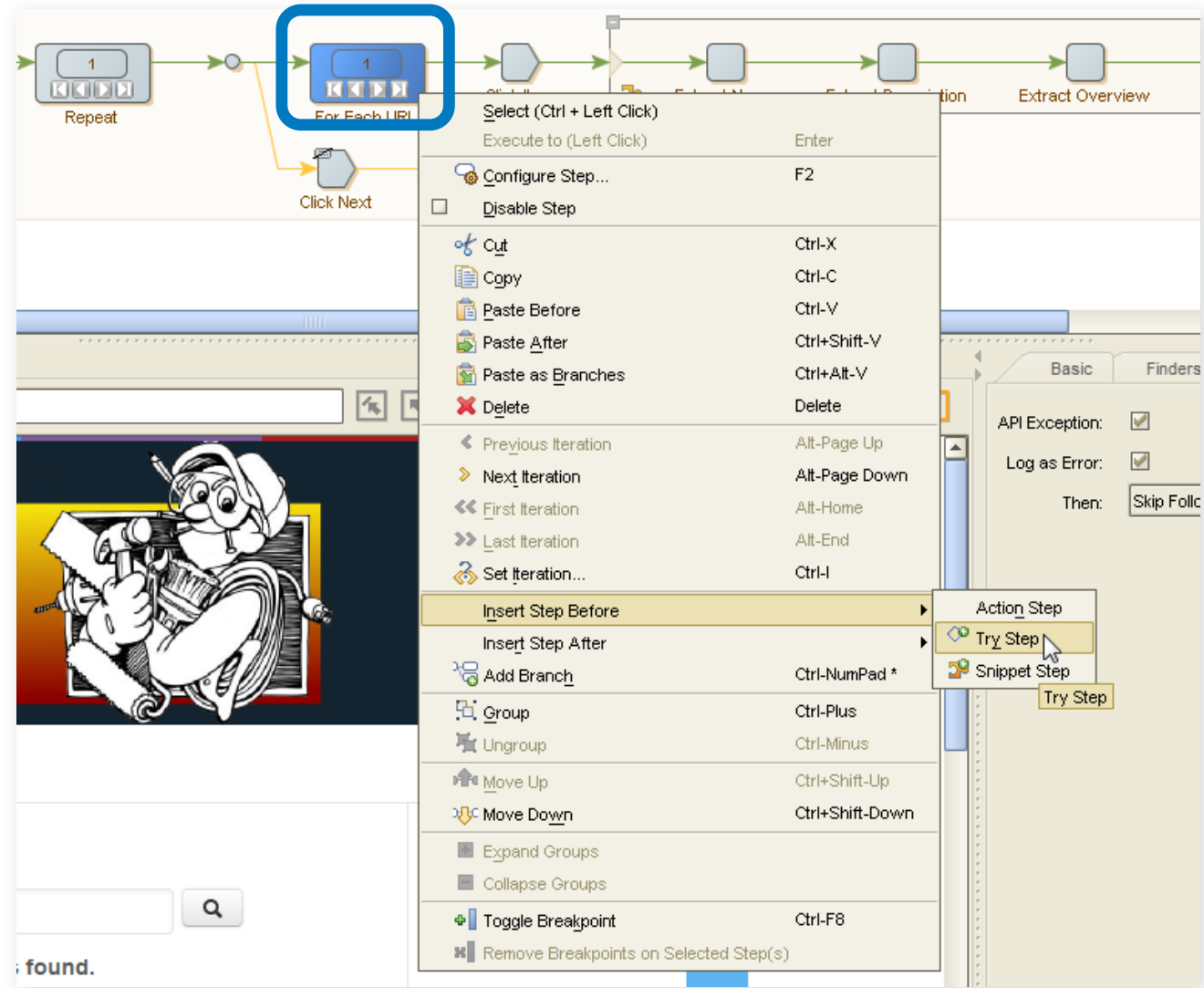
# But What Happens if an Item is Not Found?

- 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.

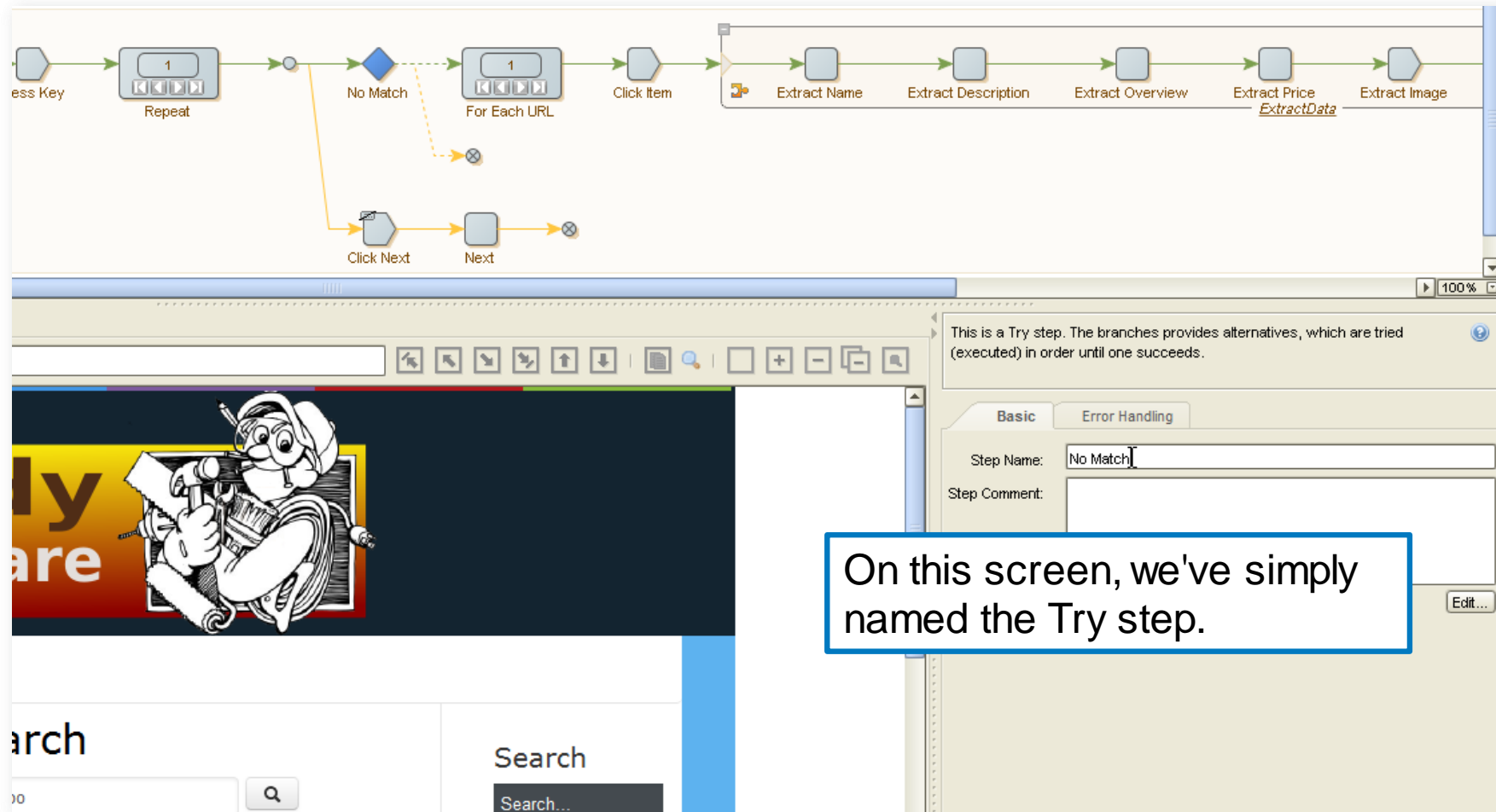


# We'll Add a Try Step

- ◆ 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.

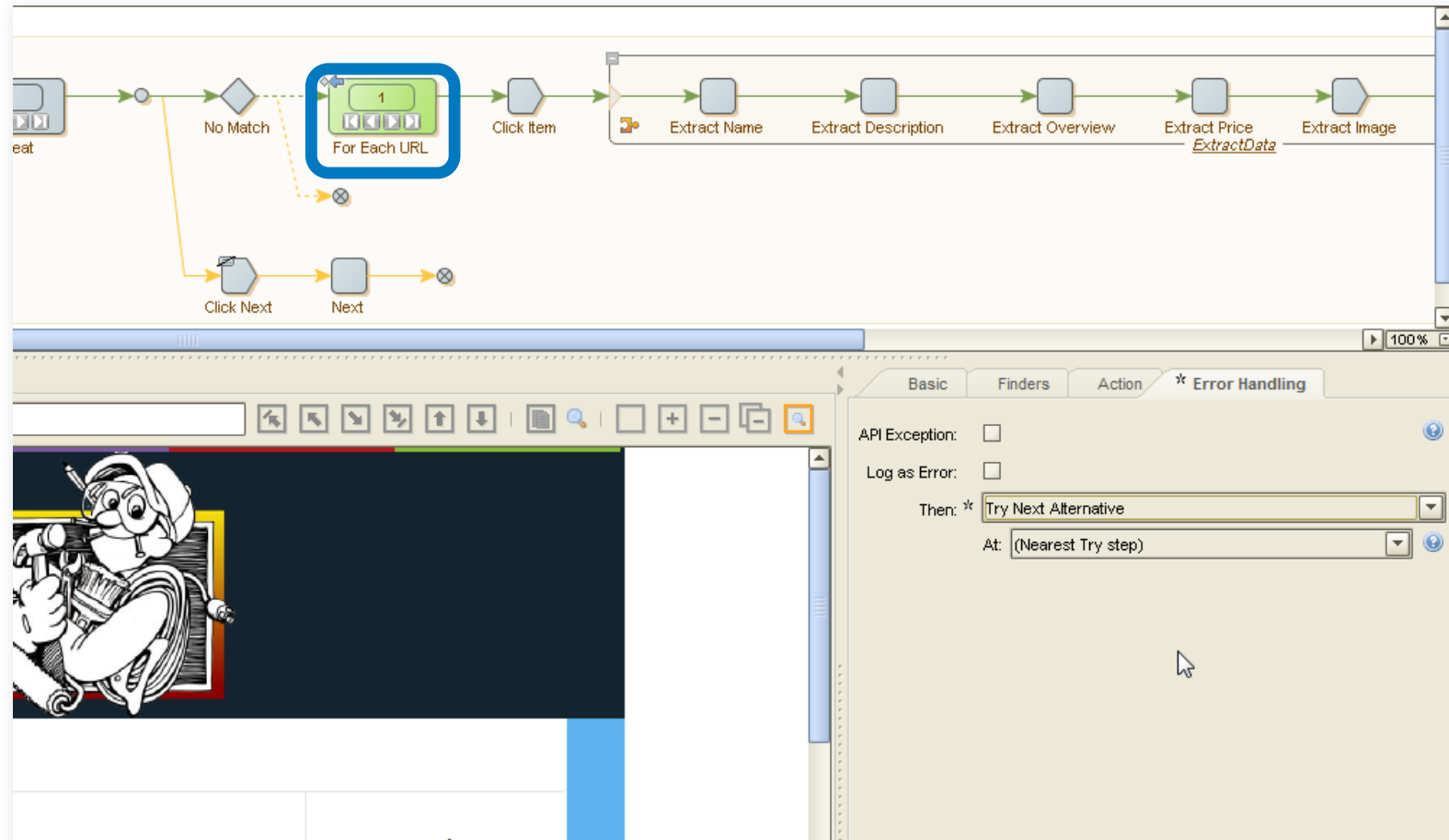


# If the Try Step Fails, We'll Go to the Next Branch



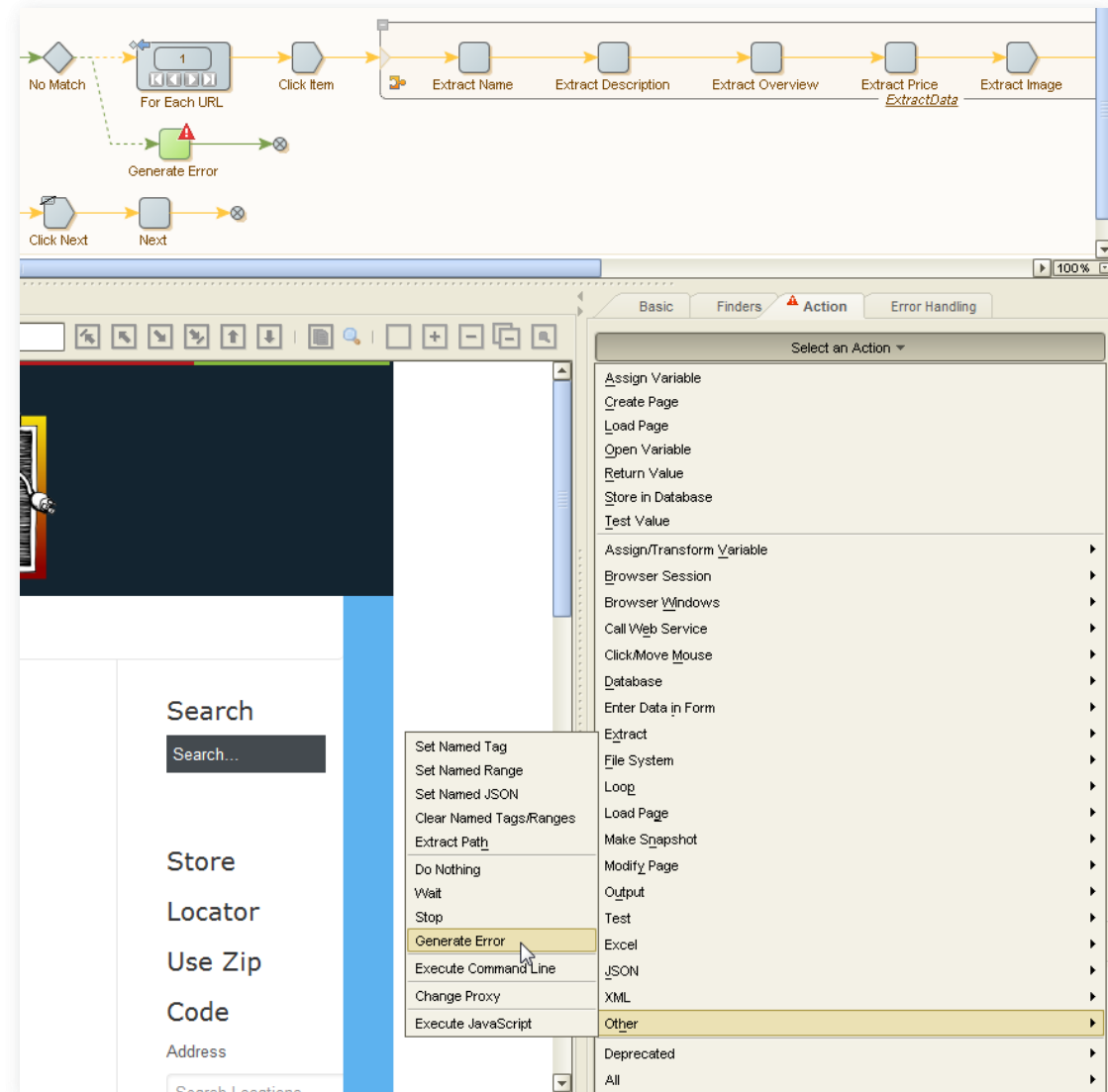
# If We Tell the Robot to Go to the Next Alternative

- 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.

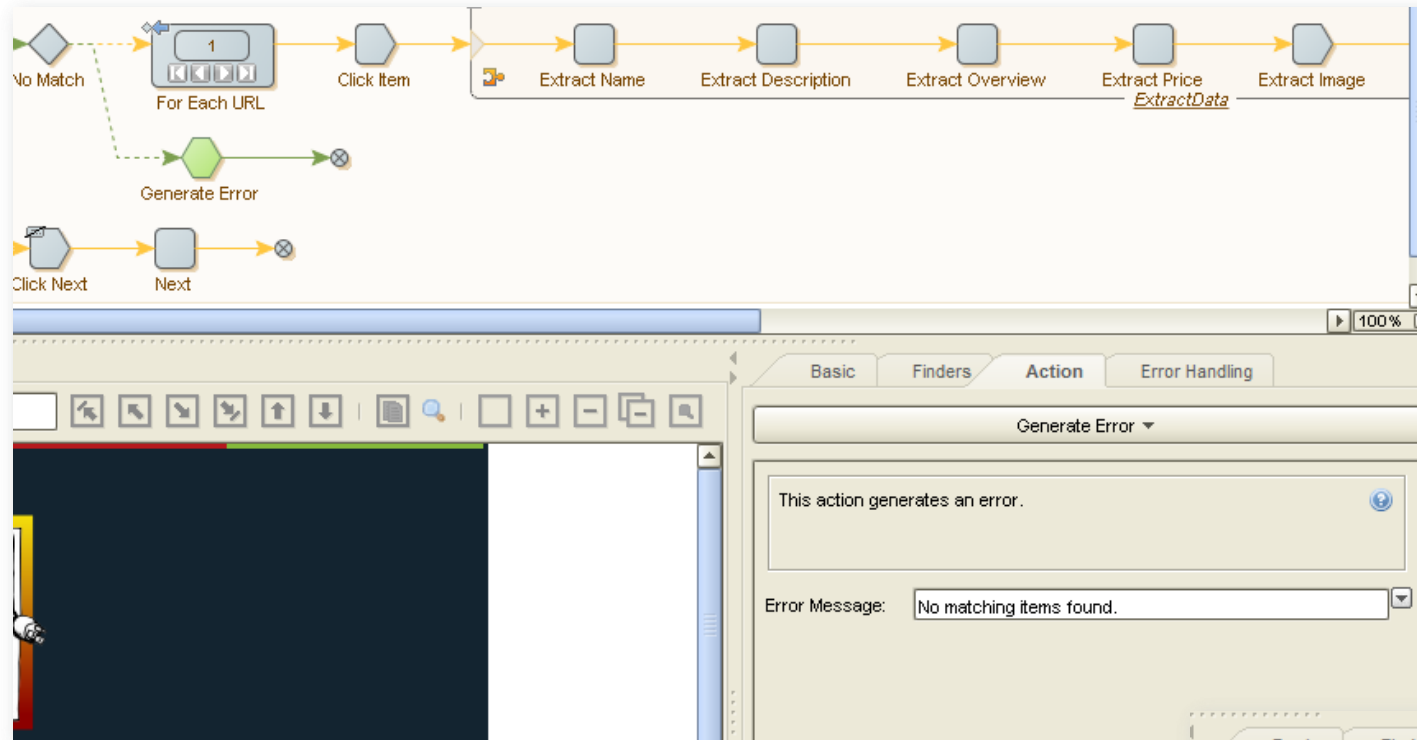


# And Generate an Error that is Logged

- ◆ 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.

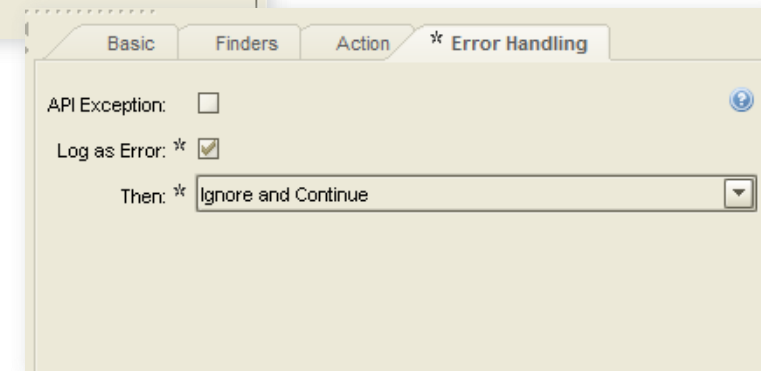


# You Choose the Error Message



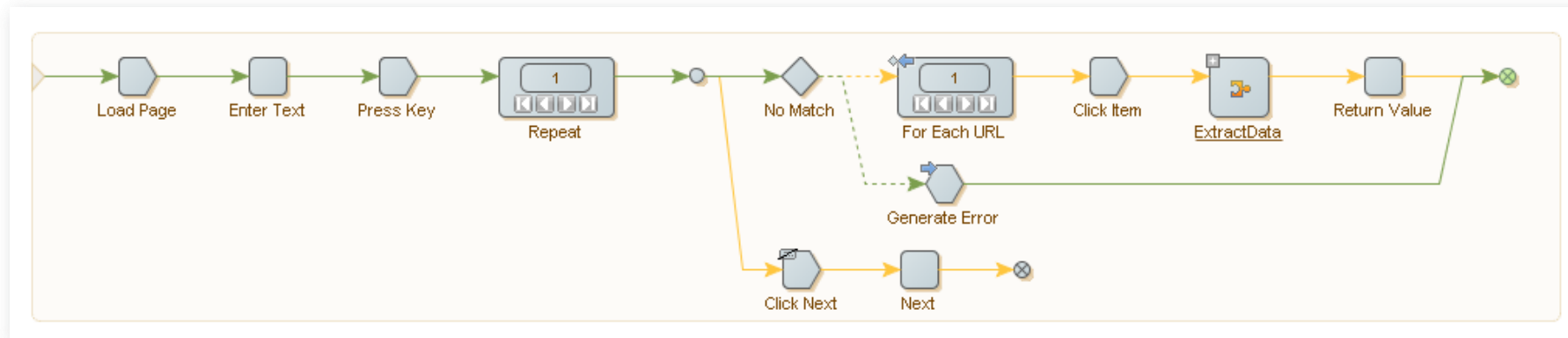
We set the error message to something like: "No matching items found."

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