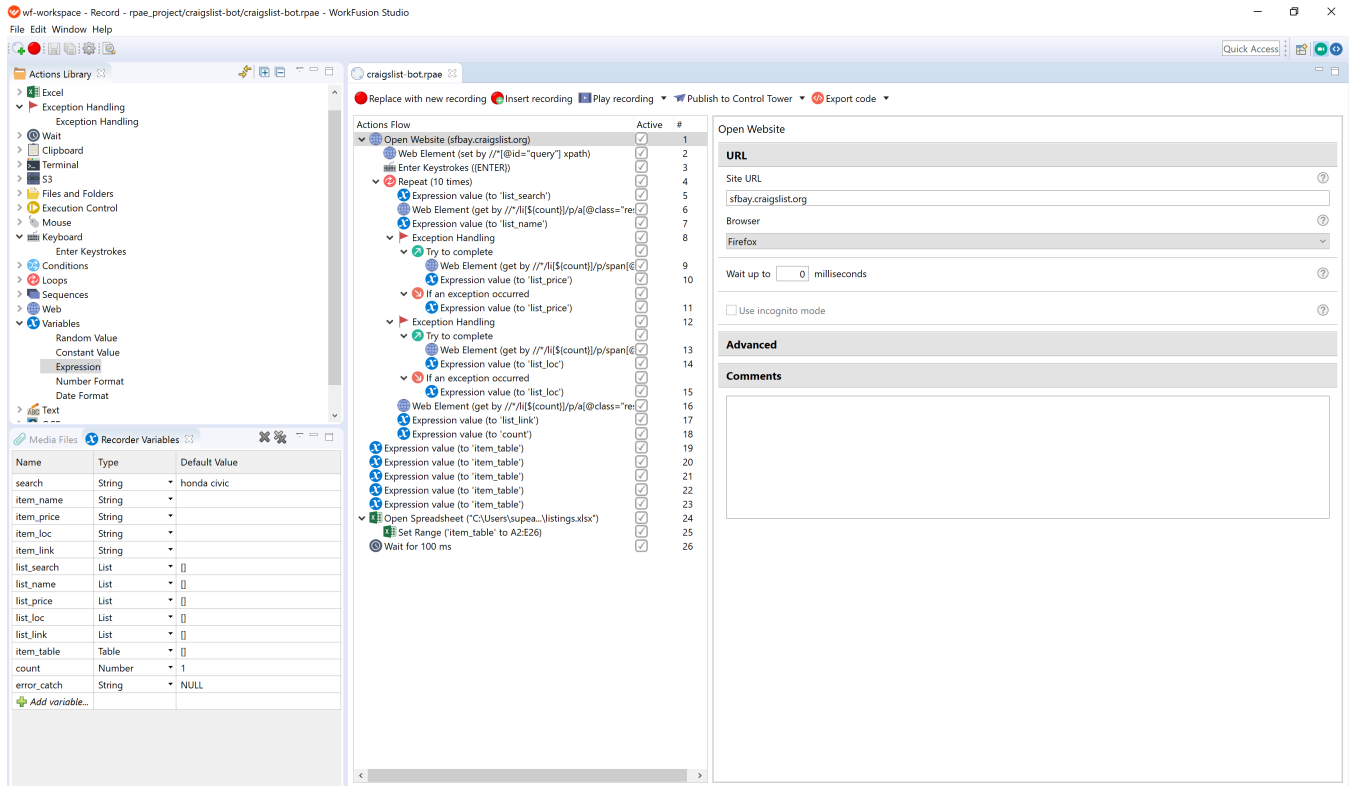


Craigslist Bot

- [Bot Overview](#)
- [Quick Links](#)
- [Accessing Bot](#)
- [Describing Workflow](#)
 - [Configuring Variables](#)
 - [Specifying Search Parameters](#)
 - [Extracting Search Results](#)
 - [Entering Results into Excel File](#)
- [Playing Script](#)

Bot Overview

The Craigslist Bot conducts a search on sfbay.craigslist.org and returns the top results to a spreadsheet.



Search parameters are configurable. The bot execution takes around 55 seconds for 10 items.

 Constructing this script from scratch requires knowledge of XPath. See [XPath Guide](#) for more details.

Quick Links

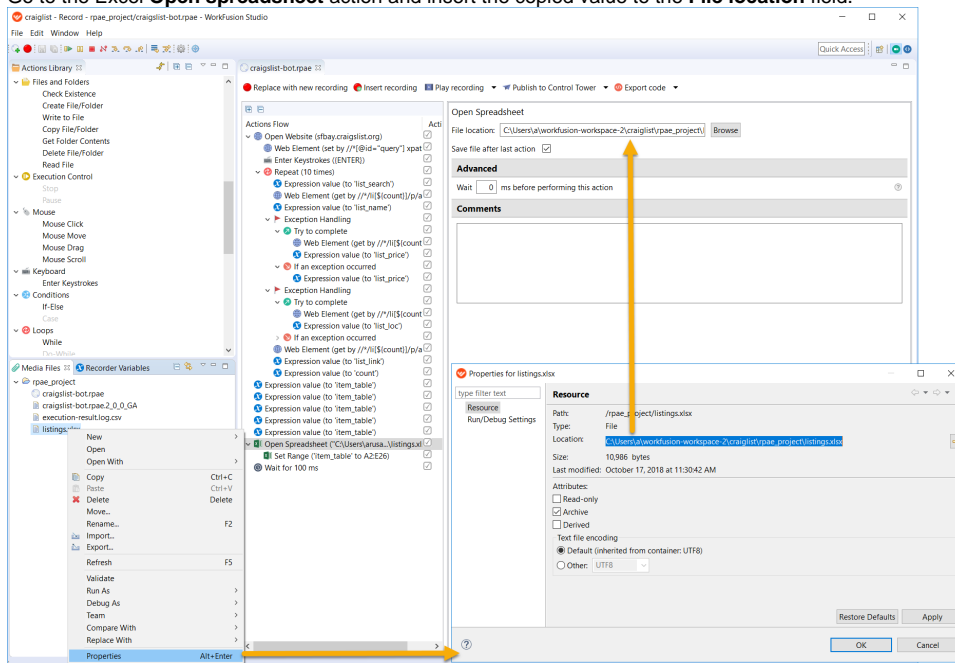
Video: [Craigslist Bot Recording.mp4](#)

Download Link: [craigslist-bot.zip](#)

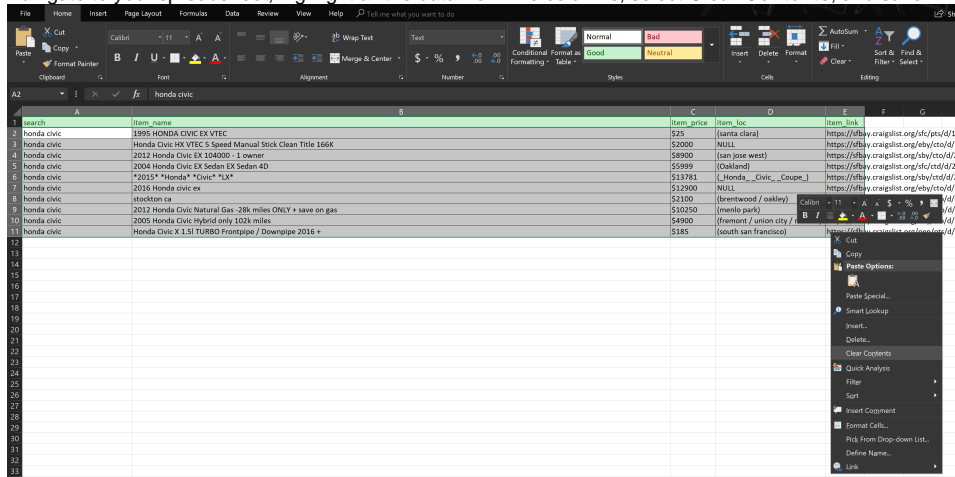
Accessing Bot

1. [Download](#) and extract the craigslist-bot folder to the rpae_project folder in RPA Express workspace.
2. To configure the Excel file path, right-click listings.xlsx on the **Media Files** tab, select **Properties > Resource > Location**, and copy the resource location.

- Go to the Excel **Open spreadsheet** action and insert the copied value to the **File Location** field.



- Navigate to your spreadsheet, highlight all the data from five columns, select **Clear Contents**, and save.



Describing Workflow

Configuring Variables

Before you start building your script, you should define all the needed Recorder Variables.

Name	Type	Default Value
search	String	honda civic
item_name	String	
item_price	String	
item_loc	String	
item_link	String	
list_search	List	[]
list_name	List	[]
list_price	List	[]
list_loc	List	[]
list_link	List	[]
item_table	Table	[]
count	Number	1
error_catch	String	NULL
+ Add variabl...		

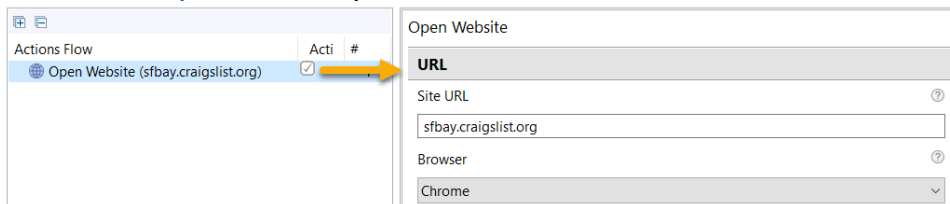
- Set up the following String variables:
 - search – the default search value, you can search for 'apple watch', 'macbook pro', 'honda civic', etc.
 - item_name – item name
 - item_price – item price
 - item_loc – item location
 - item_link – item link
 - error_catch – the default value is NULL
- Set up the following List variables:
 - list_search
 - list_name
 - list_price
 - list_loc
 - list_link
- Add a Table variable (item_table) and a Number variable (count).

Specifying Search Parameters

Actions to be used:

- Web
- Keyboard

- Go to **Actions Library** and add **Web > Open Website** to Actions Flow.



- Enter sfbay.craigslist.org into the **Site URL** field.
 - Specify your browser in **Browser**.
- Add **Web > Web Element** to identify a search bar for entering a search request. Drag the action inside the **Open Website** group.
 - Select **Mode > Set value**.
 - Select the `search` variable for **Input**.

c. Specify the element's XPath – `//*[@id="query"]`

Web Element

Mode

Get value Set value ?

Input

Use value from variable ?

search ▼

Options

XPath of the element ?

`//*[@id="query"]`

3. Add **Keyboard > Enter Keystrokes**. To set up the key, press **Change** within **Key combination** and press Enter.

Enter keystrokes

Key combination Type text Text from variable

Input

Key to press

`{{ENTER}}` Change

Advanced

Extracting Search Results

i Actions to be used:

- [Loops](#)
- [Variables](#)
- [Web](#)
- [Exception Handling](#)

1. Go to Actions Library and add **Loops > Repeat**. Enter 10 (times) for **Repeat the nested actions**.

Actions Flow

	Acti	#
Open Website (sfbay.craigslist.org)	<input checked="" type="checkbox"/>	1
Web Element (set by <code>//*[@id="query"]</code> xpath)	<input checked="" type="checkbox"/>	2
Enter Keystrokes ({{ENTER}})	<input checked="" type="checkbox"/>	3
Repeat (10 times)	<input checked="" type="checkbox"/>	4
Expression value (to 'list_search')	<input checked="" type="checkbox"/>	5

Expression value

Variable

Select variable:

list_search ▼

Expression

Choose operation for selected list ?

Push (append)

Get size

Value to append: ?

search ▼

Advanced

i To instruct your Bot which search results to extract, add expression values.

2. Add **Variables > Expression** for the `list_search` variable inside the **Loops** group.

- Select `list_search` for **Variable**.
 - Check **Push (append)**.
 - Select `search` as **Value to append**.
3. Add **Web > Web Element** to get the item name.
 - Select **Mode > Get value**.
 - Specify the element's XPath – `//*[@li[{$count}]]/p/a[@class="result-title hdrlnk"]`
 - Select `item_name` for **Output**.

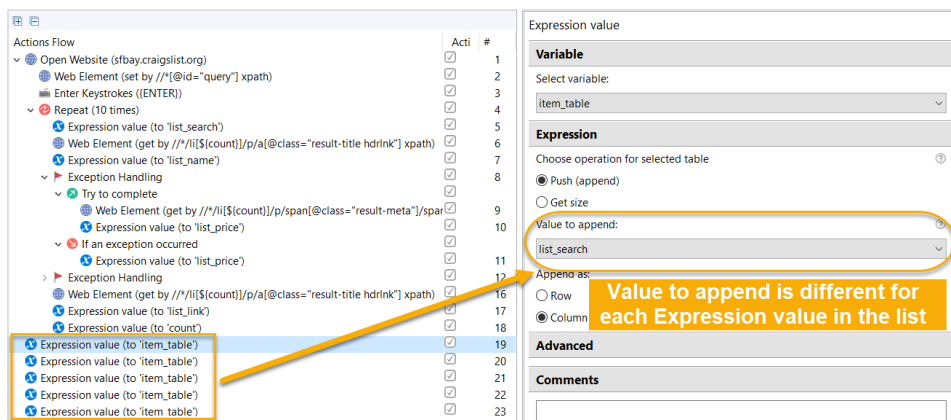
4. Add **Variables > Expression** for the `list_name` variable.
 - Select `list_name` for **Variable**.
 - Check **Push (append)**.
 - Select `item_name` as **Value to append**.
5. Add **Exception Handling** to ensure a safe execution of your workflow.
 - Within **Try to complete**, add **Web > Web element** to get the item price.
 - Select **Mode > Get value**.
 - Specify the element's XPath – `//*[@li[{$count}]]/p/span[@class="result-meta"]/span[@class="result-price"]`
 - Select `item_price` for **Output**.

- Add **Variables > Expression**.
 - Select `list_price` for **Variable**.
 - Check **Push (append)**.
 - Select `item_price` as **Value to append**.
- Within **an exception occurred**, add **Variables > Expression**.
 - Select `list_price` for **Variable**.

- ii. Check **Push (append)**.
 - iii. Select `error_catch` as **Value to append**.
6. Add one more **Exception Handling**.
 - a. Within **Try to complete**, add **Web > Web element** to get the item location.
 - i. Select **Mode > Get value**.
 - ii. Enter `//*[@li[{$count}]/p/span[@class="result-meta"] /span[@class="result-hood"]` in the **XPath** field.
 - iii. Select `item_loc` for **Output**.
 - b. Add **Variables > Expression**.
 - i. Select `list_loc` for **Variable**.
 - ii. Check **Push (append)**.
 - iii. Select `item_loc` as **Value to append**.
 - c. Within **if an exception occurred**, add **Variables > Expression**.
 - i. Select `list_loc` for **Variable**.
 - ii. Check **Push (append)**.
 - iii. Select `error_catch` as **Value to append**.
7. Add **Web > Web element** to get the item link.
 - a. Select **Mode > Get value**.
 - b. Specify the element's XPath - `//*[@li[{$count}]/p/a[@class="result-title hdrlnk"]`
 - c. Check **Use element attribute** and enter `href`.
 - d. Select `item_link` for **Output**.
8. Add **Variables > Expression** for the `list_link` variable.
 - a. Select `list_link` for **Variable**.
 - b. Check **Push (append)**.
 - c. Select `item_link` as **Value to append**.
9. Add **Variables > Expression** for the `count` variable.
 - a. Select `count` for **Variable**.
 - b. Enter `#{count}+1` for **Expression**.
10. To populate the list, add **Variables > Expression** for the `item_table` variable.
 - a. Select `item_table` for **Variable**
 - b. Check **Push (append)**.
 - c. Select **Column** for **Append as**.
 - d. Select a variable for **Value to append**. The following variables to be selected in the field:
 - `list_search`
 - `list_name`
 - `list_price`
 - `list_loc`
 - `list_link`

Attention!

As the expression value is added for five variables mentioned above, repeat Step 10 five times in the script, with each **variable** selected for **Value to append**.



Entering Results into Excel File

Actions to be used:

- Excel
- Wait

1. Go to Actions Library and add **Excel > Open spreadsheet**.
 - a. Specify the file path in the **Craigslist Bot#File location** field: `C:\Users\username\workfusion-workspace-2\craigslist\pae_project\listings.xlsx`

- b. To save the document when all the values are inserted, check **Save file after last action**.
2. Add **Excel > Set Range**.
 - a. Select `item_table` for **Input**.
 - b. Go to **Options** and set up the range: **From cell #** – A2; **To cell #** – E26.

Set Range

Input

Select Table variable: ?

item_table ▼

Options

Paste variable value into this range:

From cell #: ?

To cell #: ?

Advanced

Comments

3. Add the **Wait** action and set up **Pause script execution** (for 100 milliseconds).

Playing Script

When you finish building your script, you should have the following order of actions in Actions Flow.

Actions Flow		Acti	#
▼	🌐 Open Website (sfbay.craigslist.org)	<input checked="" type="checkbox"/>	1
	🌐 Web Element (set by //*[@id="query"] xpath)	<input checked="" type="checkbox"/>	2
	⌨ Enter Keystrokes ({ENTER})	<input checked="" type="checkbox"/>	3
▼	🔄 Repeat (10 times)	<input checked="" type="checkbox"/>	4
	🔗 Expression value (to 'list_search')	<input checked="" type="checkbox"/>	5
	🌐 Web Element (get by //*/li[\${count}]/p/a[@class="result-title hdrlnk"] xpath)	<input checked="" type="checkbox"/>	6
	🔗 Expression value (to 'list_name')	<input checked="" type="checkbox"/>	7
▼	🚩 Exception Handling	<input checked="" type="checkbox"/>	8
	▼ 🟢 Try to complete	<input checked="" type="checkbox"/>	
	🌐 Web Element (get by //*/li[\${count}]/p/span[@class="result-meta"]/spar	<input checked="" type="checkbox"/>	9
	🔗 Expression value (to 'list_price')	<input checked="" type="checkbox"/>	10
	▼ 🛑 If an exception occurred	<input checked="" type="checkbox"/>	
	🔗 Expression value (to 'list_price')	<input checked="" type="checkbox"/>	11
>	🚩 Exception Handling	<input checked="" type="checkbox"/>	12
	🌐 Web Element (get by //*/li[\${count}]/p/a[@class="result-title hdrlnk"] xpath)	<input checked="" type="checkbox"/>	16
	🔗 Expression value (to 'list_link')	<input checked="" type="checkbox"/>	17
	🔗 Expression value (to 'count')	<input checked="" type="checkbox"/>	18
	🔗 Expression value (to 'item_table')	<input checked="" type="checkbox"/>	19
	🔗 Expression value (to 'item_table')	<input checked="" type="checkbox"/>	20
	🔗 Expression value (to 'item_table')	<input checked="" type="checkbox"/>	21
	🔗 Expression value (to 'item_table')	<input checked="" type="checkbox"/>	22
	🔗 Expression value (to 'item_table')	<input checked="" type="checkbox"/>	23
▼	📄 Open Spreadsheet ("C:\Users\arusa...\listings.xlsx")	<input checked="" type="checkbox"/>	24
	📄 Set Range ('item_table' to A2:E26)	<input checked="" type="checkbox"/>	25
	⌚ Wait for 100 ms	<input checked="" type="checkbox"/>	26

Click **Play Recording** to view the Craigslist performance.