



Web Automation Actions

Web automation in Zoho RPA streamlines and automates various web tasks, empowering you to interact with web elements, extract data, and perform actions without manual intervention. This module simplifies complex web interactions and enhances workflow efficiency.

Supported Web Browsers

Supported browsers [for automation execution]: Mozilla Firefox, Google Chrome, Microsoft Edge, and Ulaa. Click [here](#) to download the extension.

Web browser extension [for Web Recorder]: Google Chrome, Microsoft Edge, and Ulaa. [Learn more](#) about Web browser recorder.

 To configure your RPA Web Automation actions, you must install the web browser extension. [Learn more](#).

Available Actions

1. Open URL

Navigates to a specific URL in a web browser.

Example: Use this action to navigate to a website or a web application as the starting point for your workflow.

Open URL: https://www.zylker.com
Navigates the web browser to a given URL.

Choose browser
Chrome

Execute in the background?
 Yes No

URL *
https://www.zylker.com

Open in
 Same tab New tab

[Delay settings](#) ▾

INSERT VARIABLE
Click fields to map to your action

Search variable here

System Variables
Variables holding predefined contextual information

- date Current date (e.g. 2020-12-22)
- date time Current datetime (e.g. 2021-05-30T23:30:30+05:30)

Current date (e.g. 2020-12-22)

CANCEL DONE

Configuration

Choose browser: Select the web browser you want to use to run the RPA flow.

Execute in the background: Choose whether to display the browser in the foreground or run the automation in the background.

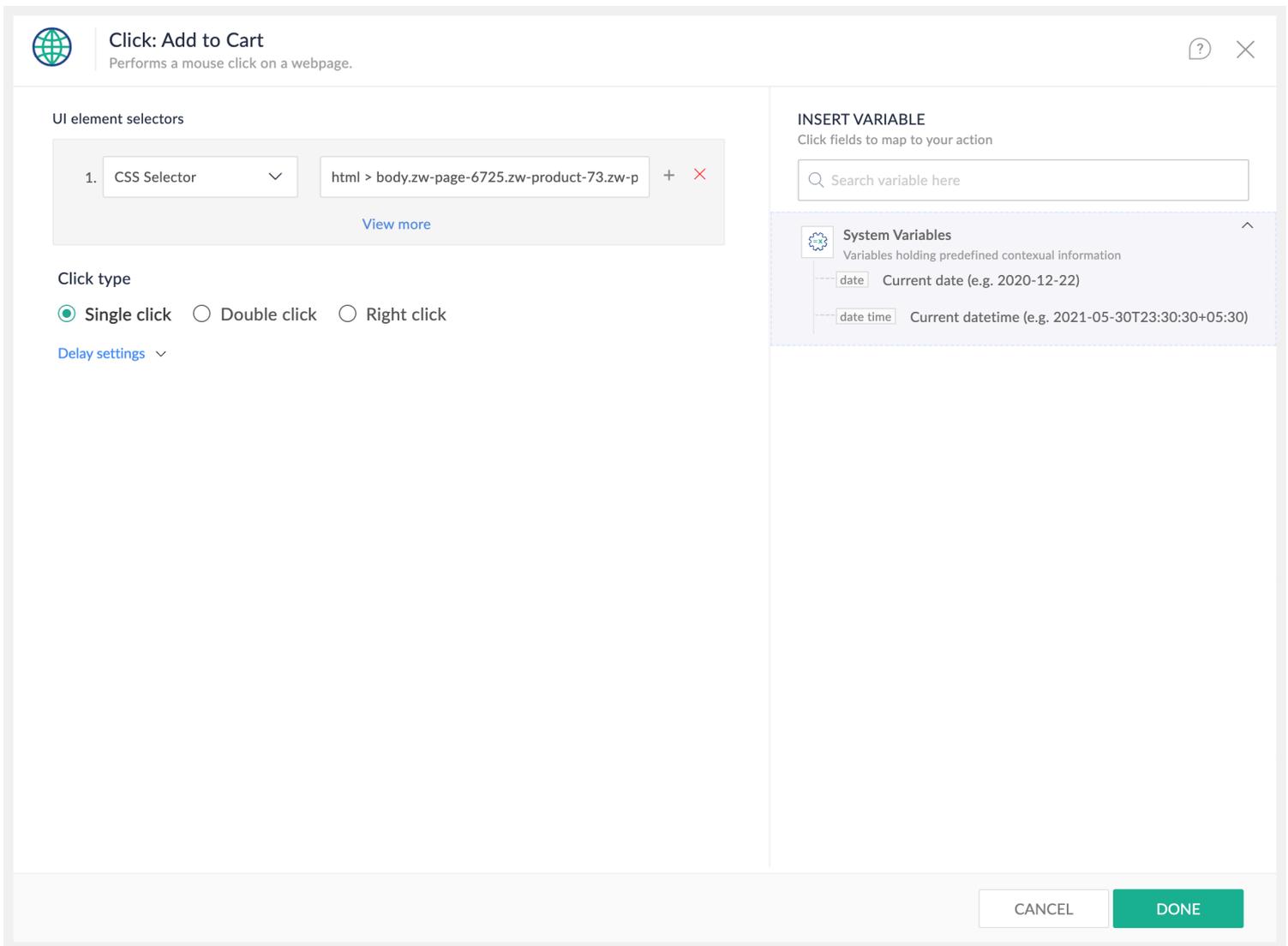
URL: Enter the website URL you want to navigate to.

Open in: Specify how you want to open the new URL. Choose whether to open it in a new tab or the same tab.

2. Click

Simulates a mouse click on a web element (e.g. buttons, links).

Example: When automating an ecommerce site, the bot can click the "Add to Cart" button to add items to the shopping cart.



Configuration

UI element selectors: UI element selectors allows to identify and interact with specific elements on a web page.

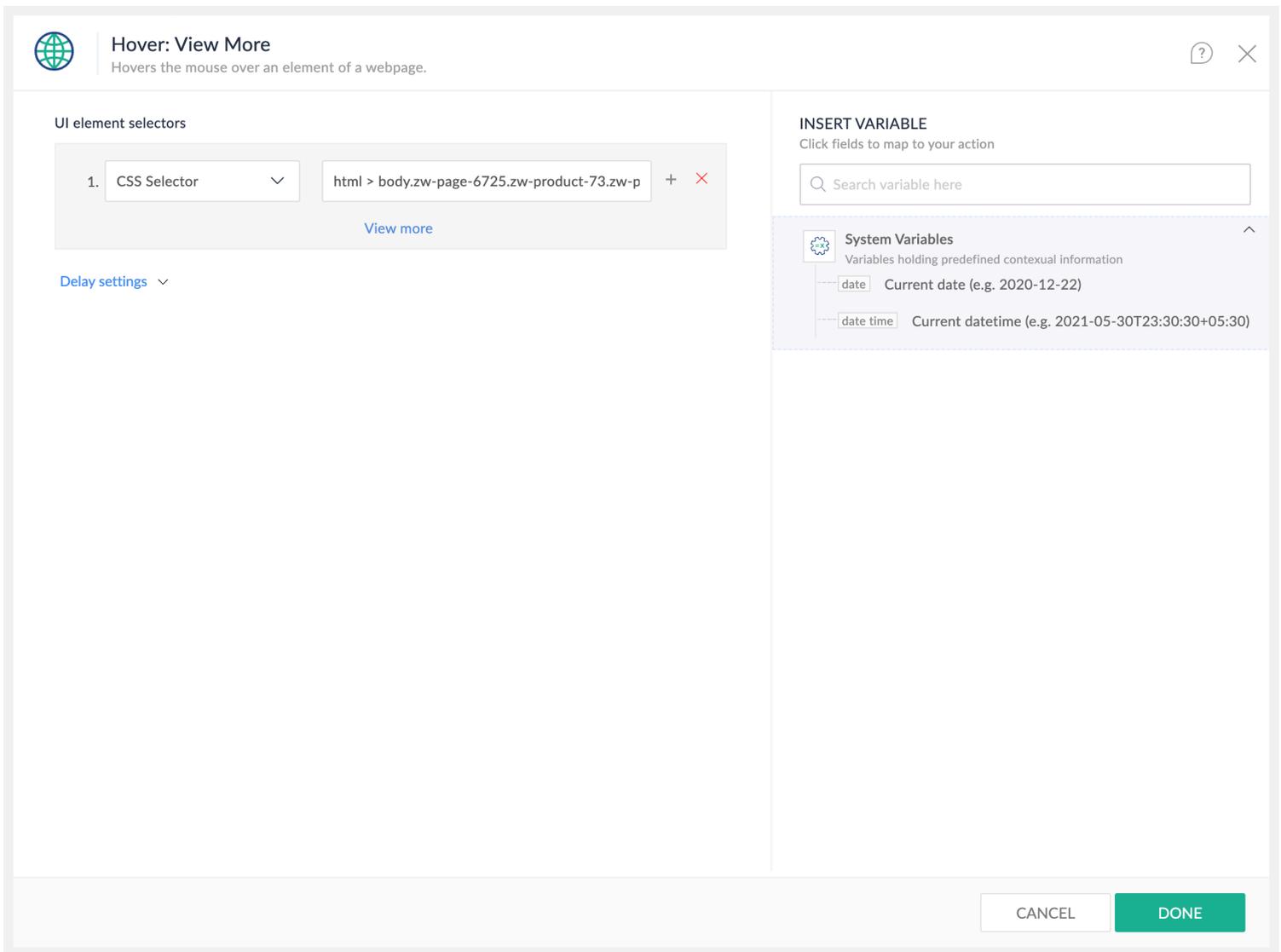
- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

Click type: Lets you choose the type of click you want to perform, whether a single click, double click, or right click.

3. Hover

Hovers the mouse over an element of a webpage to view tooltips or reveal hidden or dropdown menus.

Example: Hovering over navigational menu items on a webpage usually displays tooltips that explain what the menu options do.



Configuration

UI element selectors: UI element selectors allows to identify and interact with specific elements on a webpage.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

4. Set text

Writes text to the text attribute of a specified UI element on a webpage. It can be used to input text into a text field or input box.

Example: Fill in the name field on a Contact Us form on a website.

Configuration

UI element selectors: UI element selectors allow you to identify and interact with specific elements on a web page.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

Input text: The text to be entered into the input field.

When the input field already has text: If there is existing text in the input field, determine how to handle it. We can do the following:

- **Replace text:** Replace the existing text with the new text.
- **Append text:** Add the given text to the existing text.

5. Get text

Extracts text content from a specified UI element on a webpage.

Example: On a weather forecasting website, we can use Get Text to extract and display the current temperature.

Configuration

UI element selectors: UI element selectors allows to identify and interact with specific elements on a web page.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

6. Extract table data

Extracts data from an HTML table on a webpage.

Example: Extract weather forecast data like temperature, humidity, and wind speed from a weather website, or gather population statistics for various countries from a demographic data website's table.

Configuration

UI element selectors: UI element selectors allow you to identify and interact with specific elements on a web page.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.

- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

Next button selectors: These are specific UI element selectors for pager elements, next buttons, or navigation elements used for paginated tables. This helps navigate through tables with large datasets.

Number of data rows to extract: If the table contains a large number of rows (e.g., 1000), specify the maximum number of rows to extract.

Data columns to exclude: Specify the column numbers (in the HTML table) that you want to exclude from data extraction. Separate multiple column numbers with commas (e.g., 1, 2) or specify a range (e.g., 1-5)

 To use this action on a webpage, hover an element in an HTML table, then click the gear icon (settings) on the recorder and change the action to Extract Table Data.

7. Set checkbox

Selects or clears checkboxes or toggle switches on a webpage.

Example: On websites or web applications, users can often customize their notification preferences for different types of alerts, such as push notifications, email notifications, and in-app notifications. These preferences are typically managed through checkboxes or toggle switches. You can use this action to automate the task.

Set checkbox: Same as Shipping Address
Selects or clears checkboxes or toggle switches on a webpage.

UI element selectors

1. Id + X

[View more](#)

Checkbox value

Check Uncheck Toggle

[Delay settings](#) ▾

INSERT VARIABLE
Click fields to map to your action

System Variables
Variables holding predefined contextual information

- date Current date (e.g. 2020-12-22)
- date time Current datetime (e.g. 2021-05-30T23:30:30+05:30)

CANCEL DONE

Configuration

UI element selectors: UI element selectors allow you to identify and interact with specific elements on a web page.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

Checkbox value: The value that should be assigned for the checkbox field, that is check, uncheck or toggle.

8. Get checkbox state

Determines whether a checkbox is checked or unchecked.

Example: Check if users have accepted the terms and conditions on a website form.

Get checkbox state: Agree to Terms
Gets the state of the checkbox or toggle switch, whether on or off.

Variable name *
getCheckboxState_14

UI element selectors

1. Id #ZC_GXXCQK_Agree_to_terms_and_conditions

Delay settings

INSERT VARIABLE
Click fields to map to your action

Search variable here

System Variables
Variables holding predefined contextual information

- date Current date (e.g. 2020-12-22)
- date time Current datetime (e.g. 2021-05-30T23:30:07:00)

Type into: Search Product
Set text

CANCEL DONE

Configuration

UI element selectors: UI element selectors allow you to identify and interact with specific elements on a webpage.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

9. Select item

Selects an option from a dropdown list on a webpage.

Example: On an online travel website, from and to destination places are usually in dropdown lists. We can use the Select Item action to automate the task.

Configuration

UI element selectors: UI element selectors allows to identify and interact with specific elements on a webpage.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

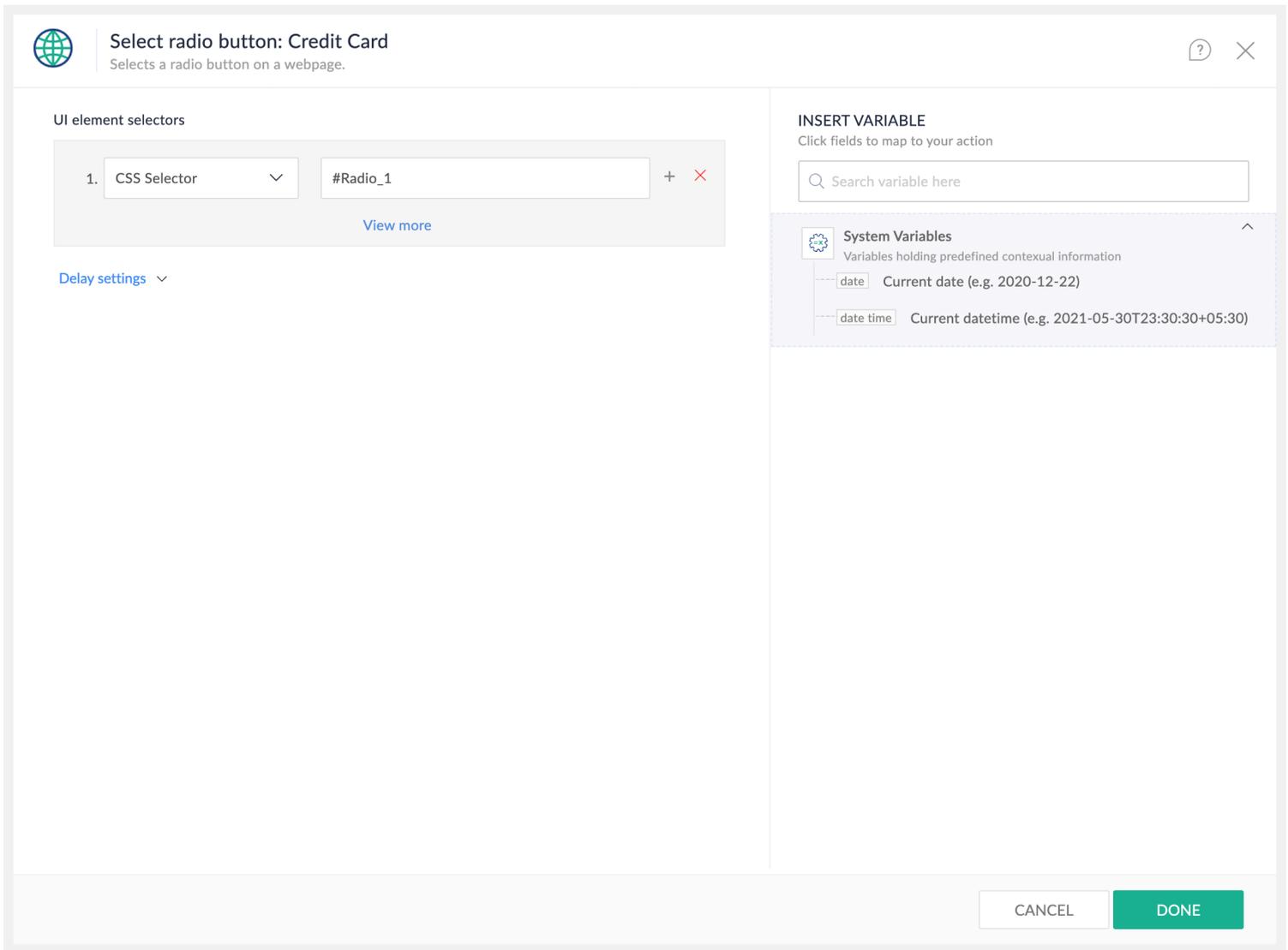
Operation: Choose the action you want to perform on the target element. The available operations are:

- **Select item by value:** Choose a list item based on its value.
 - **Value:** Enter the value of the element you want to select.
- **Select item by index:** Choose a list item based on its position in the list.
 - **Index:** Specify the index of the element you want to select. Indexing starts from 1.
- **Clear all:** Remove all selected items from the list.

10. Select radio button

Selects a radio button on a webpage.

Example: On the checkout page of ecommerce websites, customers select their preferred payment method from a list of available radio buttons. We can use the Select Radio Button action to automate this task.



Configuration

UI element selectors: UI element selectors allows to identify and interact with specific elements on a web page.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

11. Scroll

Allows you to automate the scrolling of web pages, bringing specific content into view or navigating through lengthy webpages.

Example: On an ecommerce website, use the "Scroll" action to scroll through product listings, ensuring that all products are loaded and available for data extraction.



Scroll
Performs a mouse scroll over to a UI element or by pixels.

? ×

Scroll type

UI element selector(s) Scroll by pixels

Scroll direction

Vertical Horizontal

Number of pixels *

To scroll up or left, use negative values

[Delay settings](#) ▾

INSERT VARIABLE
Click fields to map to your action

🔍 Search variable here

System Variables
Variables holding predefined contextual information

- date Current date (e.g. 2020-12-22)
- date time Current datetime (e.g. 2021-05-30T23:30:30+05:30)

CANCEL

DONE

Configuration

Scroll type: Define how you want to perform the scrolling action on the webpage. We support the following options:

- **UI element Selectors:** Scroll to a specific element on the webpage based on UI element selectors.
 - **UI element selectors:** UI element selectors allows to identify and interact with specific elements on a webpage.
 - **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
 - **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.
- **Scroll by pixels:** Scroll the webpage based on a specified number of pixels.
 - **Scroll direction:** Choose whether to scroll vertically or horizontally.
 - **Number of pixels:** The number of pixels that you want to scroll. To scroll up or left, you can use negative values.

12. Send hot keys

Enables the automation of keyboard shortcuts to trigger specific actions or commands within a web application.

Example: You may want to use Ctrl + D to bookmark the webpage you are currently have open.

Send hot keys : Bookmark webpage
Sends keyboard shortcuts to a UI element on a webpage.

UI element selectors

1. CSS Selector + X

[View more](#)

Key modifiers

Ctrl Alt Shift Win

Key

Generated hot key

ctrl+D

[Delay settings](#) v

INSERT VARIABLE
Click fields to map to your action

System Variables
Variables holding predefined contextual information

- Current date (e.g. 2020-12-22)
- Current datetime (e.g. 2021-05-30T23:30:30-07:00)

Configuration

UI element selectors: UI element selectors allows to identify and interact with specific elements on a web page.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

Key modifiers: Choose the key modifiers for your hotkey. Available modifiers include Ctrl, Alt, Shift, and Win.

Key: Select the key to be used along with the key modifiers to form the hotkey.

Generated hot key: The resulting hotkey combination will be displayed in this field for reference.

13. Type into

Automates the process of entering text or data into input fields or forms on a webpage.

Example: Use the Type Into action to simulate typing in your login credentials and password to log in to your online store portal app.



Type into: Search Product

Writes a string to the text attribute of a specified UI element on a webpage.

? ✕

Variable name *

UI element selectors

1. + ✕

[View more](#)

Input text

When the input field already has text

[Delay settings](#) ▾

INSERT VARIABLE

Click fields to map to your action

System Variables

Variables holding predefined contextual information

- Current date (e.g. 2020-12-22)
- Current datetime (e.g. 2021-05-30T23:30:30-07:00)

Configuration

UI element selectors: UI element selectors allows to identify and interact with specific elements on a webpage.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM.

Input text: Enter the text you want to type into the field

Insert keys: Use this option to include special keys like Enter, Tab, and Esc within the input text.

When the input field has text: Lets you choose how to handle existing text in the input field.

- **Replace text:** Replace the existing text with the new text.
- **Append text:** Add the new text to the end of the existing text.

Click before typing: Use this option to ensure the UI element is active before attempting to type.

Delay between keystrokes (in ms): Set the desired time delay (in milliseconds) between each keystroke to simulate a more natural typing pace.

14. Check if element exists

Allows you to verify if a specific piece of text or a UI element exists on a webpage.

Example: On an order checkout page, you may want to check if an "apply coupon" field exists before you try to apply the coupon code.

Check if element exists
Verifies if a specific piece of text or a UI element exists.

Variable name *
checkIfElementExists_21

UI element selectors

1. Id [▼] zh-home-btn [✚]

Check if webpage
 Contains element Contains text
[Delay settings](#) [▼]

INSERT VARIABLE
Click fields to map to your action

Search variable here

System Variables
Variables holding predefined contextual information

- date Current date (e.g. 2020-12-22)
- date time Current datetime (e.g. 2021-05-30T23:30:30-07:00)

Type into: Search Product
Set text

CANCEL DONE

Configuration

UI element selectors: UI element selectors allows to identify and interact with specific elements on a web page.

- **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
- **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM

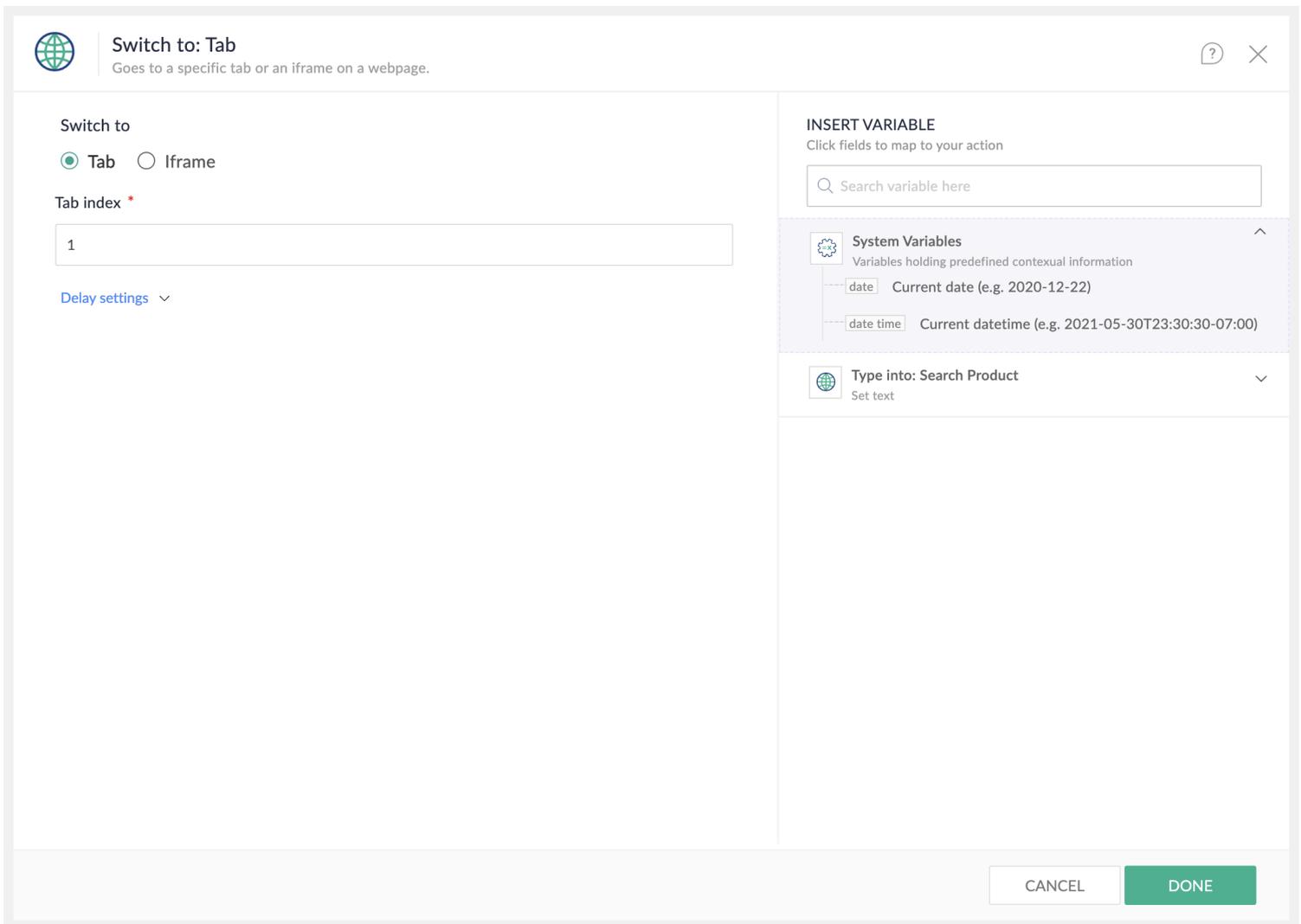
Check if webpage: Check if the webpage contains the element or a text.

Text: Enter the text you want to search for.

15. Switch to

Allows you to switch focus between different open windows or frames within a web application.

Example: Automating flight or hotel bookings for clients by switching between various travel websites to compare prices and availability.



Configuration

Switch to: Choose whether you want to switch to a tab or an iframe on the current active tab.

Tab index: Index of the tab that you want to switch to. Starts with 0.

Select the iframe: You can select the iframe based on the following:

- **Iframe Index:** Order of the iframe on the webpage
- **Name or Id:** Name or ID of the given iframe.
- **UI element selectors:** UI element selectors allows to identify and interact with specific elements on a webpage.
 - **Selector type:** We support various selector types, including XPath, ID, Name, and CSS Selectors.
 - **Selector:** Enter the value of the web component or locator for the element you want to target. This value should correspond to an element in the HTML DOM

16. Refresh webpage

Automates the process of reloading or refreshing a webpage to access updated content.

Example: To ensure that the latest data is extracted in a web-scraping scenario, we may want to periodically refresh the financial news website.

17. Go forward a page

Navigates the web browser forward to the next page in the browsing history.

Example: Navigating to the next page of a paginated list of items, such as a list of products on an e-commerce website

18. Go back a page

Navigates the web browser back to the previous page in the browsing history.

Example: After viewing details of a particular product on an e-commerce website, we can use the "Go back a page" action to return to the list of products.

19. Close browser

Terminates the web browser session, closing all open tabs.

Example: Use the Close Browser action after you complete all of your actions to end your automation process.

Delay Settings

Delay settings allow you to configure a delay for an action. This option ensures that the actions do not fail due to loading or UI elements not present on the screen. Sometimes UI elements take time to display or load on the webpage so a delay may be required.

Delay before action (Time in ms): The amount of time to wait before performing the action.

Delay after action (Time in ms): The amount of time to wait after performing the action.

Wait for element (Time in ms): The amount of time to wait for an element to appear before performing the action. The distinction between wait before action and wait for element is that wait before action will wait for the specified amount of time before beginning the activity. The wait for element is the amount of time spent waiting during the action. For instance, displaying options in a dropdown field may take a few seconds.