




Form Actions

This list will have tasks that are commonly used for form-related actions. These tasks include the following:

- [Select](#)
- [Deselect](#)
- [Check](#)
- [Uncheck](#)
- [Clear](#)
- [Upload](#)

Select


This task is used to automate the action of selecting an option from a dropdown field that has a predefined set of options. For example, if you want to select your preferred payment method from a dropdown menu during online shopping.

 **Note:** The Select and Check tasks are two distinct actions that serve different purposes when interacting with web elements. The Select task focuses on setting a value to a dropdown field, while the Check task interacts with the checkbox. [Learn more](#).

Syntax

```
select(<locator>, <text>, <select_type>);
```

Parameter	Data Type	Description
<locator>	STRING	Address to locate the dropdown containing the options you want to choose from.

		 Note: You can inspect from the webpage to find the locators of the elements or record and store the elements. Learn more about Elements .
<text>	STRING	The text or attribute value to be selected based on the select type
<select_type>	STRING	<p>The type of select used to interact with dropdowns based on their index, displayed text, or attribute value.</p> <ul style="list-style-type: none"> • select.INDEX : This indicates the specific index of an option within the dropdown. It allows you to choose an option based on its position within the dropdown, starting from zero for the first option. • select.TEXT : This selects an option from a dropdown based on its visible text content. It allows you to choose an option by specifying the exact text that is displayed to the user. • select.VALUE : This selects an option within the dropdown based on its underlying value attribute. It allows you to choose an option by specifying the value associated with the option, which may differ from the visible text content. For example, the user sees the shipping country in the options as 'India' but the value attribute corresponding to this visible text will be 'IN'.

Example: A user is testing the intractability of the checkout functionality of an ecommerce website. Here, the user tests if a customer can select their preferred country for shipping within the dropdown provided in the website. The following script selects the country "US" from the dropdown.


```
1. select(ui.zwatch.checkout-dropdown,"US",select.VALUE);
```

where:

ui.zwatch.checkout-dropdown	The stored locator for the dropdown element that lists the available shipping countries on the website.
"US"	The text associated with one of the options within the dropdown, in this case, the shipping country.
select.VALUE	The select type chosen as the attribute value associated with the option was used inside as the parameter for the task.


Deselect

This task is used to automate the action of deselecting an option from a dropdown field that has a predefined set of options. For example, the user on a shopping website removes an item by deselecting one from their shopping cart.

 **Note:** The Deselect and Uncheck tasks are two distinct actions that serve different purposes when interacting with web elements. The Deselect task focuses on deselecting the currently selected value from a dropdown field, while the Uncheck task interacts with the checkbox. [Learn more](#).

Syntax

deselect(<locator>,<text>,<select_type>);

Parameter	Data Type	Description
<locator>	STRING	Address to locate the dropdown containing the options you want to choose from.  Note: You can inspect from the webpage to find the locators of the elements or record and store the elements. Learn more about Elements .
<text>	STRING	The text or attribute value to be deselected based on the select type
<select_type>	STRING	The type of select used to interact with dropdowns based on their index, displayed text, or attribute value. <ul style="list-style-type: none">• select.INDEX : This indicates the specific index of an option within the dropdown. It allows you to choose an option based on its position within the dropdown, starting from zero for the first option.• select.TEXT : This selects an option from a dropdown based on its visible text content. It allows you to choose an option by specifying the exact text that is displayed to the user.

		<ul style="list-style-type: none"> • select.VALUE : This selects an option within the dropdown based on its underlying value attribute. It allows you to choose an option by specifying the value associated with the option, which may differ from the visible text content. For example, the user sees the shipping country in the options as 'India' but the value attribute corresponding to this visible text will be 'IN'.
--	--	--

Example: A user is testing the intractability of the checkout functionality of an ecommerce website. Here, the user wants to test if they can deselect the preferred choice. The following script deselects the country "US" from a dropdown.


```
1. deselect(ui.zwatch.checkout-dropdown,"US",select.VALUE);
```

where:

ui.zwatch.checkout-dropdown	The stored locator for the dropdown element that lists the available shipping countries on the website.
"US"	The text associated with one of the options within the dropdown, in this case, the shipping country.
select.VALUE	The select type chosen as the attribute value associated with the option was used inside as the parameter for the task.


Check

This task performs the action of checking a checkbox on a webpage. For example, using the check task, the user accepts the terms and conditions on web forms.


Note: The Select and Check tasks are two distinct actions that serve different purposes when interacting with web elements. The Select task focuses on setting a value to a dropdown field, while the Check task interacts with the checkbox. [Learn more.](#)

Syntax

check(<locator>);

Parameter	Data Type	Description
<locator>	STRING	Address to locate the checkbox on the webpage. <div>  Note: You can inspect from the webpage to find the locators of the elements or record and store the elements. Learn more about Elements. </div>

Example: You are automating an e-commerce website for a clothing store. The following script ensures shoppers can apply a discount by checking a checkbox during checkout.


1. `check(ui.zwatch.GMT.discount);`

where:

ui.zwatch.GMT.discount	The stored locator for the discount checkbox element.
------------------------	---

Uncheck


The task performs the action of unchecking a checkbox on a webpage. For example, to uncheck the "keep me signed in" checkbox in a login website.

 **Note:** The Deselect and Uncheck tasks are two distinct actions that serve different purposes when interacting with web elements. The Deselect task focuses on deselecting the currently selected value from a dropdown field, while the Uncheck task interacts with the checkbox. [Learn more](#).

Syntax

`uncheck(<locator>);`

Parameter	Data Type	Description
<locator>	STRING	Address to locate the checkbox on the webpage.

		 Note: You can inspect from the webpage to find the locators of the elements or record and store the elements. Learn more about Elements .
--	--	--

Example: The user wants to test if they can uncheck a charity checkbox provided on the webpage. The following script unchecks the charity box.

```
1. uncheck(ui.zwatch.GMT.discount);
```

where:


ui.zwatch.GMT.discount	The stored locator for the discount checkbox element.
------------------------	---

Clear

This task is used to clear the current value of an input element on a webpage or web application. For example, you can clear the input in an email input field of a login form.

Syntax

```
clear(<locator>);
```

Parameter	Data Type	Description
<locator>	STRING	Address to locate the element you want to clear. <div> Note: You can inspect from the webpage to find the locators of the elements or record and store the elements. Learn more about Elements.</div>

Example: The user is testing a login input form for a website. The following script clears the input from the email input field of the login.

1. `openURL("https://www.zwatch.com" , "same window");` // Navigates to the website. [Understand about openURL task.](#)
2. `setValue("#login_id",$Email);` // Inputs the email in the field login ID. [Learn more about Set Value task.](#)
3. `clear(#login_id);`

where:



"https://www.zwatch.com"	URL of the website that needs to be opened.
"same window"	This represents that the URL needs to be opened in the same window.
"#login_id"	The locator of the element "login ID" of the login form. Here, the '#' is used to refer to the ID of the element.
\$Email	The variable where the email value is stored. <Learn more about <i>Variables</i> >

Upload

The task performs the action of uploading a file to a webpage.

Syntax

`upload(<locator>, files.<filename>);`

Parameter	Data Type	Description
<locator>	STRING	<p>Address to locate the input element you want to upload the file.</p> <div> Note: You can inspect from the webpage to find the locators of the elements or record and store the elements. Learn more about Elements.</div>
<filename>	OBJECT	<p>Name of the file to be uploaded.</p> <div> Note: The files are accessed from Files inside Settings for each project.</div>

--	--	--

Example: You are automating an e-commerce website that allows sellers to add product images to their listings. The following script uploads a product image on the website.

```
1. upload(ui.blue_bay_gmt.h,files.Blue_Bay_GMT);
```

where:

ui.blue_bay_gmt.h	The stored locator for the input element the file has be uploaded.
files.Blue_Bay_GMT	Name of the file to be uploaded.