

# **Pfizer Eclinical Test Trails**

*Release 0.0.1*

**ITT Digital**

Jul 01, 2021



<b>Python Module Index</b>	<b>19</b>
<b>Index</b>	<b>21</b>



**class** browser.**BrowserActions** ( *driver* )

A class for all BrowserActions that can be performed on a webpage or webdriver. It contains all the methods which are associated with browsers

**Class** *browseractions*

**close\_current\_tab** ( )

To close the current opened tab

**Returns** None

**close\_tab** ( *name: str* )

Closing the tab/window by passing the name of window/tab

**Parameters** **name** – name\_of\_window/tab (String)

**Returns** None

**close\_tab\_at\_index** ( *index: int* )

Close the tab/window at a given index position

**Parameters** **index** – index\_of\_window/tab (int)

**Returns** None

**get\_current\_tab\_name** ( ) → str

Returns the current tab's name through windows handle

**Returns** tab name (String)

**get\_current\_url** ( ) → str

Retrieves the current url of the webpage

**Returns** url (String)

**get\_number\_of\_tabs** ( ) → int

Returns the number of tabs currently opened

**Returns** no of tabs (String)

**get\_page\_source** ( ) → str

Returns the complete page source o the current page.

**Returns** page source (String)

**get\_tab\_list** ( ) → List[str]

Returns the currently opened tabs in list

**Returns** list of currently opened tabs (List)

**get\_title ( )** → str

Returns the current browser/tab title in string

Returns title (String)

**go\_to\_protected\_url ( full\_url: str, username: str, password: str )**

Opening the protected url page.

**Parameters**

- **full\_url** – url\_of\_the\_page (String)
- **username** – username\_to\_login (String)
- **password** – password\_to\_login (String)

**Returns** None

**go\_to\_url ( url: str )**

Navigate to the given URL

**Parameters** url – String

**Returns** None

**navigate\_back ( )**

Navigated to previous page

**Returns** None

**refresh ( )**

Refresh the current page

**Returns** None

**resize\_to ( width, height )**

Resize the window to the given width and height

**Parameters**

- **width** – integer
- **height** – integer

**Returns** None

**switch\_to\_tab ( tab\_name: str )**

Switch to a tab using the name of the window

**Param** tab\_name: name\_of\_window/tab (String)

**Returns** None

**switch\_to\_tab\_index ( index: int )**

Switch to tab using index of that tab

**Param** index : index\_of\_the\_tab (int)

**Returns** None

**steps\_custom.check\_external\_site\_loading ( site\_url )**

Fetching the external site URL and comparing the status code to be less than 300

**Parameters** site\_url – external\_site\_url

**Returns** None

**steps\_custom.click\_on\_element** ( *selenium\_generics: page\_objects.selenium\_generics.SeleniumGenerics, locators: utils.locators.Locators, locator\_path* )

To set the text into a given field

**Parameters** `locator_path` – locator\_of\_the\_element\_from\_locators.json

**Returns** None

`steps_custom.click_on_shadow_element` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, locators: utils.locators.Locators, element\_path* )

To perform a click on the shadow element on a webpage

**Parameters** `element_path` – locator/xpath of the element

**Returns** None

`steps_custom.compare_search_result` ( *selenium,* *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, text, locators: utils.locators.Locators, locator\_path1,*  
*locator\_path2, locator\_path3* )

Searching and comparing the results of given texts

**Parameters**

- `text` – text\_to\_search
- `locator_path1` – search\_box\_location
- `locator_path2` – find\_key\_location
- `locator_path3` – result\_label\_location

**Returns** None

`steps_custom.element_is_displayed` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, locators: utils.locators.Locators, locator\_path* )

To validate an element is displayed on the given path or not

**Parameters** `locator_path` – locator\_of\_the\_element\_from\_locators.json

**Returns** None

`steps_custom.element_is_hidden` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, locators: utils.locators.Locators, locator\_path* )

To validate the element is hidden or not

**Parameters** `locator_path` – locator\_of\_the\_hidden\_element\_from\_locators.json

**Returns** None

`steps_custom.element_text` ( *selenium\_generics: page\_objects.selenium\_generics.SeleniumGenerics,*  
*locators: utils.locators.Locators, locator\_path, expected\_text* )

To validate the element text with expected text

**Parameters**

- `locator_path` – locator\_of\_the\_element\_from\_locators.json
- `expected_text` – text\_to\_compare

**Returns** None

`steps_custom.generate_pdf` ( )

To generate a pdf with all the screenshot taken

**Returns** None

`steps_custom.generate_pdfs` ( )

To generate a pdf with the current screenshot taken

**Returns** None

`steps_custom.header_menu_button_style` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, data\_table* )

To validate the css style of a shadow element against given styles

**Parameters** `data_table` – styles\_as\_list

**Returns** None

`steps_custom.hover_over_element` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, locators: utils.locators.Locators, locator\_path* )

To Perform an hover on an element on webpage:

**Parameters** `locator_path` – locator/xpath of the element

**Returns** None

`steps_custom.hover_over_shadow_element` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, locators: utils.locators.Locators, element\_path* )

To Perform an hover on an shadow element on webpage:

**Parameters** `element_path` – path of the shadow element

**Returns** None

`steps_custom.navigate_to_external_page` ( *browser: actions.browser.BrowserActions, url* )

Navigate to a external page url

**Parameters** • **browser** – value will be passed from BrowserActions class  
• **url** – url\_which\_you\_wants\_to\_navigate

**Returns** None

`steps_custom.nextgen_simple_card_style` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, component\_locator, data\_table* )

To set the style of an element

**Parameters** • **component\_locator** – locator\_path\_of\_the\_component  
• **data\_table** – styles\_as\_list

**Returns** None

`steps_custom.page_loaded` ( *selenium\_generics: page\_objects.selenium\_generics.SeleniumGenerics,*  
*page\_url* )

To validate the given page url is loaded or not

**Parameters** `page_url` – url\_value

**Returns** None

`steps_custom.page_scroll_position` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, scroll\_position* )

Scrolling to the particular position of the webpage

**Parameters** `scroll_position` – value\_of\_scroll\_position

**Returns** None

`steps_custom.page_visual_is_valid` ( *selenium,* *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, data\_table* )

To validate the full page screenshot against the base/given image

**Parameters** `data_table` – elements\_to\_hide\_values

**Returns** None

`steps_custom.sitemap_urls_loading` ( *log* )

Sitemap urls are fetching and validating the status code

**Returns** None



`steps_custom.take_element_screenshot_with_actions ( selenium, selenium_generics, base_url, data_table )`

To take screen of elements with desired actions

**Parameters**

- **base\_url** – url
- **data\_table** – list\_of\_actions

**Returns** None

`steps_custom.take_full_page_screenshot_from_sitemap_with_actions ( selenium, selenium_generics, base_url, data_table )`

To take a full page screenshot with desired actions

**Parameters**

- **base\_url** – url\_of\_the\_page
- **data\_table** – list\_of\_actions

**Returns** None

`steps_custom.urls_from_sitemap ( site_url, log )`

To fetch all the URL's frm the external site

**Parameters** **site\_url** – external\_url\_value

**Returns** None

`steps_given.check_cookie_content ( selenium, name, value )`

Validating the cookie contains the given value

**Parameters**

- **name** – name\_of\_cookie
- **value** – value\_of\_cookie

**Returns** None

`steps_given.check_cookie_content_is_not ( selenium, name, value )`

Validating the cookie doesn't contains the given value

**Parameters**

- **name** – name\_of\_cookie
- **value** – value\_of\_cookie

**Returns** None

`steps_given.check_cookie_does_not_exist ( selenium, name )`

Validating the given cookie is not present

**Parameters** **name** – name\_of\_the\_cookie

**Returns** None

`steps_given.check_cookie_exists ( selenium, name )`

Validating the given cookie is present

**Parameters** **name** – name\_of\_the\_cookie

**Returns** None

`steps_given.check_css_property_is ( attribute, selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, value )`

Validating the CSS attribute value of an element is equal to

**Parameters**

- **locator\_path** – locator\_of\_the\_element
- **value** – value\_of\_attribute\_to\_validate

**Returns**    None

`steps_given.check_css_property_is_not ( attribute, selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, value )`

Validating the CSS attribute value of an element is not equal to

**Parameters**    • **locator\_path** – locator\_of\_the\_element  
                  • **value** – value\_of\_attribute\_to\_validate

**Returns**    None

`steps_given.check_modal ( )`

Validating an alertbox/confirmbox/prompt is opened

**Returns** None

`steps_given.check_modal_not_present ( )`

Validating an alertbox/confirmbox/prompt is not opened

**Returns** None

`steps_given.check_property_is ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, value, attribute )`

Validating the attribute value of an element is equal to

**Parameters**    • **locator\_path** – locator\_of\_the\_element  
                  • **value** – value\_of\_attribute\_to\_validate  
                  • **attribute** – name\_of\_the\_attribute

**Returns**    None

`steps_given.check_property_is_not ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, value, attribute )`

Validating the attribute value o an element is not equal to

**Parameters**    • **locator\_path** – locator\_of\_the\_element  
                  • **value** – value\_of\_attribute\_to\_validate  
                  • **attribute** – name\_of\_the\_attribute

**Returns**    None

`steps_given.close_all_but_first_tab ( selenium )`

Validating only one tab/window is opened

**Returns** None

`steps_given.contains_any_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path )`

Validating the button/element contains some text

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns**    None

`steps_given.contains_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, text )`

Validating the button/element contains the given text

**Parameters**

- **locator\_path** – locator\_of\_the\_element
- **text** – text\_to\_check

**Returns** None

`steps_given.does_not_contain_any_text` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, locators: utils.locators.Locators, locator\_path* )

Validating the button/element doesn't contain any text

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_given.does_not_contain_text` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, locators: utils.locators.Locators, locator\_path, text* )

Validating the button/element not contains the given text

**Parameters**

- **locator\_path** – locator\_of\_the\_element
- **text** – text\_to\_check

**Returns** None

`steps_given.element_displayed` ( *selenium, locators: utils.locators.Locators, locator\_path* )

Performs a custom wait for element to be visible

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_given.element_enabled` ( *selenium, locators: utils.locators.Locators, locator\_path* )

Validating the element is enabled after waiting for some seconds

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_given.element_equals_text` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics, locators: utils.locators.Locators, locator\_path, text* )

Validating the button/element text is as given

**Parameters**

- **locator\_path** – locator\_of\_the\_element
- **text** – expected\_text\_of\_the\_element

**Returns** None

`steps_given.element_exists` ( *selenium, locators: utils.locators.Locators, locator\_path* )

Validating an element is present on the page

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_given.element_not_displayed` ( *selenium, locators: utils.locators.Locators, locator\_path* )

Performs a wait and checking the element is not visible

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_given.element_not_enabled` ( *selenium, locators: utils.locators.Locators, locator\_path* )

Validating the element is not enabled after waiting for some seconds

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns**     None

`steps_given.element_not_equals_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, text )`

Validating the button/element text is not as given

**Parameters**     • **locator\_path** – locator\_of\_the\_element  
                     • **text** – text\_of\_the\_element

**Returns**     None

`steps_given.element_not_exists ( selenium, locators: utils.locators.Locators, locator_path )`

Validating an element is not present on the page

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns**     None

`steps_given.element_not_selected ( selenium, locators: utils.locators.Locators, locator_path )`

Validating the element is not selected

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns**     None

`steps_given.element_selected ( selenium, locators: utils.locators.Locators, locator_path )`

Validating the element is selected

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns**     None

`steps_given.given_page_url_is ( selenium, url )`

Validating the current URL is given URL

**Parameters** **url** – url\_to\_validate

**Returns**     None

`steps_given.given_page_url_is_not ( selenium, url )`

Validating the current URL is not given URL

**Parameters** **url** – url\_to\_validate

**Returns**     None

`steps_given.open_external_webpage ( browser: actions.browser.BrowserActions, site_url )`

Performing a navigation to an external url/page

**Parameters** **site\_url** – url\_of\_the\_page

**Returns**     None

`steps_given.open_webpage ( browser: actions.browser.BrowserActions, base_url, page_url )`

Performing a navigation to given url

**Parameters** **page\_url** – url\_of\_the\_page

**Returns**     None

`steps_given.page_title ( selenium, title )`

Validating the title is not as given

**Parameters** **title** – title\_of\_page

**Returns**     None

`steps_given.window_size ( selenium, width, height )`

Setting up the browser resolution with given width and height

**Parameters**

- **width** – width\_resolution
- **height** – height\_resolution

**Returns**    None

`steps_then.check_contains_any_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path )`

Validating the button/element text is containing some text

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns**    None

`steps_then.check_contains_no_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path )`

Validating the button/element text is not containing any text

:param locator\_path:locator\_of\_the\_element :return: None

`steps_then.check_contains_same_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path1, locator_path2 )`

To validate the text of two web element in a webpage is same

**Parameters**

- **locator\_path1** – locator\_of\_the\_element1
- **locator\_path2** – locator\_of\_the\_element2

**Returns**    None

`steps_then.check_contains_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, text )`

Validating the button/element text is equal to the given text

**Parameters**

- **locator\_path** – locator\_of\_the\_element
- **text** – text\_to\_compare

**Returns**    None

`steps_then.check_cookie_delete ( selenium, name )`

Validating the cookie is deleted

**Parameters** **name** – name\_of\_the\_cookie

**Returns**    None

`steps_then.check_cookie_presence ( selenium, name, value )`

Validating a particulare is present

**Parameters**

- **name** – name\_of\_the\_cookie
- **value** – value\_of\_cookie

**Returns**    None

`steps_then.check_cookie_update ( selenium, name, value )`

Validating the newly added value to a cookie is updated

**Parameters**

- **name** – name\_of\_cookie

- **value** – value\_of\_the\_cookie

**Returns** None

`steps_then.check_cookies_presence ( selenium )`

Validating the cookies are present in the page

**Returns** None

`steps_then.check_css_property_is ( attribute, selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, value )`

Validating the CSS attribute value of an element is equal to

- Parameters**
- **locator\_path** – locator\_of\_the\_element
  - **value** – value\_of\_attribute\_to\_validate

**Returns** None

`steps_then.check_css_property_is_not ( attribute, selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, value )`

Validating the CSS attribute value of an element is not equal to

- Parameters**
- **locator\_path** – locator\_of\_the\_element
  - **value** – value\_of\_attribute\_to\_validate

**Returns** None

`steps_then.check_does_not_contain_same_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path1, locator_path2 )`

To validate the text of two web element in a webpage is not same

- Parameters**
- **locator\_path1** – locator\_of\_the\_element1
  - **locator\_path2** – locator\_of\_the\_element2

**Returns** None

`steps_then.check_element_enabled ( selenium, locators: utils.locators.Locators, locator_path )`

Validating the element is enabled

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_then.check_element_equals_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, text )`

Validating the text of a button/element with given text is equal after removing the leading and trailing whitespace

- Parameters**
- **locator\_path** – locator\_of\_the\_element
  - **text** – text\_to\_compare

**Returns** None

`steps_then.check_element_exists ( selenium, locators: utils.locators.Locators, locator_path, occurrence_count )`

To validate the element occurrences count in a webpage is equal

- Parameters**
- **locator\_path** – locator\_of\_the\_element

- **occurrence\_count** – occurrence\_count

**Returns** None

`steps_then.check_element_not_enabled ( selenium, locators: utils.locators.Locators, locator_path )`

Validating the element is not enabled

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_then.check_element_not_equals_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, text )`

Validating the text of a button/element with given text is not equal after removing the leading and trailing whitespace

**Parameters** • **locator\_path** – locator\_of\_the\_element

- **text** – text\_to\_compare

**Returns** None

`steps_then.check_element_not_existing ( selenium, locators: utils.locators.Locators, locator_path )`

To validate the element not exists in a webpage DOM

**Parameters** **locator\_path** –

**Returns** None

`steps_then.check_element_not_exists ( selenium, locators: utils.locators.Locators, locator_path, occurrence_count )`

To validate the element occurrences count in a webpage is not equal

**Parameters** • **locator\_path** – locator\_of\_the\_element

- **occurrence\_count** – occurrence\_count

**Returns** None

`steps_then.check_element_not_selected ( selenium, locators: utils.locators.Locators, locator_path )`

Validating the element is not selected

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_then.check_element_selected ( selenium, locators: utils.locators.Locators, locator_path )`

Validate the element is selected

:param locator\_path:locator\_of\_the\_element :return: None

`steps_then.check_is_existing ( selenium, locators: utils.locators.Locators, locator_path )`

To validate the element exists in a webpage DOM

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_then.check_is_opened_in_new_window ( selenium, url )`

Validating the given URL is opened in a new tab/window

**Parameters** **url** – url\_to\_open

**Returns** None

`steps_then.check_modal ( )`

Validating an alertbox/confirmbox/prompt box is opened

**Returns** None

`steps_then.check_modal_not_present ( )`

Validating an alertbox/confirmbox/prompt box is not opened

**Returns** None

`steps_then.check_new_window ( selenium )`

Validating a new window/tab is opened

**Returns** None

`steps_then.check_no_new_window ( selenium )`

Validating a new window/tab is not opened

**Returns** None

`steps_then.check_not_contains_text ( selenium_generics: page_objects.selenium_generics.SeleniumGenerics, locators: utils.locators.Locators, locator_path, text )`

Validating the button/element text is not equal to the given text

**Parameters**

- **locator\_path** – locator\_of\_the\_element
- **text** – text\_to\_compare

**Returns** None

`steps_then.check_number_of_tabs ( selenium, count: int )`

Validating the count of currently opened tabs/windows

**Parameters** **count** – count\_to\_validate

**Returns** None

`steps_then.check_page_url ( selenium, url )`

Validating the current page URL is given URL

**Parameters** **url** – url\_to\_compare

**Returns** None

`steps_then.check_page_url_contains ( selenium, url )`

Validating the current page URL contains the given URL path

**Parameters** **url** – url\_path\_to\_compare

**Returns** None

`steps_then.check_page_url_is_not ( selenium, url )`

Validating the current page URL is not given URL

**Parameters** **url** – url\_to\_compare

**Returns** None

`steps_then.check_page_url_not_contains ( selenium, url )`

Validating the current page URL not contains the given URL path

**Parameters** **url** – url\_path\_to\_compare

**Returns** None



`steps_then.check_property_is` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics,* *locators:* *utils.locators.Locators,* *locator\_path,* *value,*  
*attribute* )

Validating the attribute value of an element is equal to

**Parameters**

- **locator\_path** – locator\_of\_the\_element
- **value** – value\_of\_attribute\_to\_validate
- **attribute** – name\_of\_the\_attribute

**Returns** None

`steps_then.check_property_is_not` ( *selenium\_generics:*  
*page\_objects.selenium\_generics.SeleniumGenerics,* *locators:* *utils.locators.Locators,* *locator\_path,* *value,*  
*attribute* )

Validating the attribute value o an element is not equal to

**Parameters**

- **locator\_path** – locator\_of\_the\_element
- **value** – value\_of\_attribute\_to\_validate
- **attribute** – name\_of\_the\_attribute

**Returns** None

`steps_then.check_title_contains` ( *selenium,* *title* )

Validating the title of the page contains the given title/string

**Parameters** **title** – partial\_string/title\_to\_compare

**Returns** None

`steps_then.check_title_is` ( *selenium,* *title* )

Validating the title of the page is expected one

**Parameters** **title** – tile\_of\_the\_page\_to\_compare

**Returns** None

`steps_then.check_title_is_not` ( *selenium,* *title* )

Validating the title of the page is not expected one

**Parameters** **title** – tile\_of\_the\_page\_to\_compare

**Returns** None

`steps_then.check_title_not_contains` ( *selenium,* *title* )

To Validate the title of the page not contains the given title/string

**Parameters** **title** – partial\_string/title\_to\_compare

**Returns** None

`steps_then.check_within_viewport` ( *selenium,* *locators:* *utils.locators.Locators,* *locator\_path* )

To validate the element is not within the viewport

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_then.element_displayed` ( *selenium,* *locators:* *utils.locators.Locators,* *locator\_path* )

To validate the element is displayed

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns** None

`steps_then.element_not_displayed ( selenium, locators: utils.locators.Locators, locator_path )`

To validate the element is not displayed

**Parameters** `locator_path` – locator\_of\_the\_element

**Returns** None

`steps_then.wait_for_displayed ( selenium, locators: utils.locators.Locators, locator_path )`

To validate the element becomes visible after some custom wait time

**Parameters** `locator_path` – locator\_of\_the\_element

**Returns** None

`steps_then.wait_for_not_displayed ( selenium, locators: utils.locators.Locators, locator_path )`

To validate the element becomes invisible after some custom wait time

:param `locator_path`:locator\_of\_the\_element :return: None

`steps_when.accept_modal ( selenium )`

Accepting the prompt/alertbox/confirmbox popup

**Returns** None

`steps_when.add_element_value ( selenium, text, locators: utils.locators.Locators, locator_path )`

Performing a text append in an element

**Parameters** • `text` – text\_to\_append

• `locator_path` – locator\_of\_the\_element

**Returns** None

`steps_when.check_cookie_content ( selenium, name )`

Deleting a cookie

**Parameters** `name` – name\_of\_the\_cookie

**Returns** None

`steps_when.check_modal ( selenium, text )`

Entering the given text into an alertbox/confirmbox/prompt popup

**Parameters** `text` – text\_to\_enter

**Returns** None

`steps_when.click_element ( selenium, locators: utils.locators.Locators, locator_path )`

Performing an click operation on an element

**Parameters** `locator_path` – locator\_of\_the\_element

**Returns** None

`steps_when.close_last_opened_window ( selenium )`

Closing the latest window

**Returns** None

`steps_when.dbl_click_element ( selenium, locators: utils.locators.Locators, locator_path )`

Performing a double click on an element

**Parameters** `locator_path` – locator\_of\_the\_element

**Returns** None

`steps_when.deselect_option_by_value ( selenium, option, locators: utils.locators.Locators, locator_path )`

Deselecting the option from a dropdown using the value

**Parameters**

- **option** – value\_to\_deselect
- **locator\_path** – locator\_of\_the\_element

**Returns**    None

`steps_when.deselect_option_by_visible_text ( selenium, option, locators: utils.locators.Locators, locator_path )`

Deselecting the option from a dropdown using the visible text

**Parameters**

- **option** – text\_to\_deselect
- **locator\_path** – locator\_of\_the\_element

**Returns**    None

`steps_when.deselect_option_index ( selenium, index: int, locators: utils.locators.Locators, locator_path )`

Deselecting the option from a dropdown using the index

**Parameters**

- **index** – index\_to\_deselect
- **locator\_path** – locator\_path

**Returns**    None

`steps_when.dismiss_modal ( selenium )`

Dismissing the prompt/alertbox/confirmbox popup

**Returns** None

`steps_when.drag_and_drop_element ( selenium, locators: utils.locators.Locators, source, target )`

Performing a drag and drop of an element using source and target value

**Parameters**

- **source** – source\_of\_the\_element
- **target** – target\_of\_the\_element

**Returns**    None

`steps_when.fetch_cookies ( selenium )`

Fetching the existing cookies from the site

**Returns** None

`steps_when.js_click_element ( selenium, locators: utils.locators.Locators, locator_path )`

Performing a click on an element using java script

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns**    None

`steps_when.move_to_element_by_offset ( selenium, locators: utils.locators.Locators, locator_path )`

Moving the mouse cursor to the element

**Parameters** **locator\_path** – locator\_of\_the\_element

**Returns**    None

`steps_when.pause_execution ( selenium, time )`

Performing a custom wait for give seconds

**Parameters** `time` – seconds\_to\_wait

**Returns** None

`steps_when.press_button ( selenium, key )`

Performing a key press

**Parameters** `key` – key\_to\_press

**Returns** None

`steps_when.scroll_to_element ( selenium, locators: utils.locators.Locators, locator_path )`

Performing a scroll to an element

**Parameters** `locator_path` – locator\_of\_the\_element

**Returns** None

`steps_when.select_option_by_index ( selenium, index: int, locators: utils.locators.Locators, locator_path )`

Selecting an option from a dropdown using the index

**Parameters**

- `index` – index\_to\_select
- `locator_path` – locator\_of\_element

**Returns** None

`steps_when.select_option_by_value ( selenium, option, locators: utils.locators.Locators, locator_path )`

Selecting an option from a dropdown using the value

**Parameters**

- `option` – value\_to\_select
- `locator_path` – locator\_of\_the\_element

**Returns** None

`steps_when.select_option_by_visible_text ( selenium, option, locators: utils.locators.Locators, locator_path )`

Selecting an option from a dropdown using the visible text

**Parameters**

- `option` – text\_to\_select
- `locator_path` – locator\_of\_element

**Returns** None

`steps_when.set_add_element_value ( selenium, locators: utils.locators.Locators, locator_path )`

Clearing the text from an input text field

**Parameters** `locator_path` – locator\_of\_the\_element

**Returns** None

`steps_when.set_element_value ( selenium, text, locators: utils.locators.Locators, locator_path )`

Performing a text insert in an element

**Parameters**

- `text` – text\_to\_insert
- `locator_path` – locator\_of\_the\_element

**Returns** None

`steps_when.set_locale ( locators: utils.locators.Locators, locale )`

Setting up the locale for an element using the locators

**Parameters** `locale` – locale\_name

**Returns** None

`steps_when.switch_to_last ( selenium )`

Focusing on the latest window

**Returns** None

- `genindex`
- `modindex`
- `search`



## **b**

browser, [XX](#)

## **s**

steps\_custom, [2](#)

steps\_given, [5](#)

steps\_then, [9](#)

steps\_when, [14](#)





**A**

`accept_modal()` (in module `steps_when`), 14  
`add_element_value()` (in module `steps_when`), 14

**B**

`browser`  
    module, 1  
`BrowserActions` (class in `browser`), 1

**C**

`check_contains_any_text()` (in module `steps_then`), 9  
`check_contains_no_text()` (in module `steps_then`), 9  
`check_contains_same_text()` (in module `steps_then`), 9  
`check_contains_text()` (in module `steps_then`), 9  
`check_cookie_content()` (in module `steps_given`), 5  
`check_cookie_content()` (in module `steps_when`), 14  
`check_cookie_content_is_not()` (in module `steps_given`), 5  
`check_cookie_delete()` (in module `steps_then`), 9  
`check_cookie_does_not_exist()` (in module `steps_given`), 5  
`check_cookie_exists()` (in module `steps_given`), 5  
`check_cookie_presence()` (in module `steps_then`), 9  
`check_cookie_update()` (in module `steps_then`), 9  
`check_cookies_presence()` (in module `steps_then`), 10  
`check_css_property_is()` (in module `steps_given`), 5  
`check_css_property_is()` (in module `steps_then`), 10  
`check_css_property_is_not()` (in module `steps_given`), 6

`check_css_property_is_not()` (in module `steps_then`), 10  
`check_does_not_contain_same_text()` (in module `steps_then`), 10  
`check_element_enabled()` (in module `steps_then`), 10  
`check_element_equals_text()` (in module `steps_then`), 10  
`check_element_exists()` (in module `steps_then`), 10  
`check_element_not_enabled()` (in module `steps_then`), 11  
`check_element_not_equals_text()` (in module `steps_then`), 11  
`check_element_not_existing()` (in module `steps_then`), 11  
`check_element_not_exists()` (in module `steps_then`), 11  
`check_element_not_selected()` (in module `steps_then`), 11  
`check_element_selected()` (in module `steps_then`), 11  
`check_external_site_loading()` (in module `steps_custom`), 2  
`check_is_existing()` (in module `steps_then`), 11  
`check_is_opened_in_new_window()` (in module `steps_then`), 11  
`check_modal()` (in module `steps_given`), 6  
`check_modal()` (in module `steps_then`), 12  
`check_modal()` (in module `steps_when`), 14  
`check_modal_not_present()` (in module `steps_given`), 6  
`check_modal_not_present()` (in module `steps_then`), 12  
`check_new_window()` (in module `steps_then`), 12  
`check_no_new_window()` (in module `steps_then`), 12  
`check_not_contains_text()` (in module `steps_then`), 12  
`check_number_of_tabs()` (in module `steps_then`), 12  
`check_page_url()` (in module `steps_then`), 12

check\_page\_url\_contains() (in module steps\_then), 12  
check\_page\_url\_is\_not() (in module steps\_then), 12  
check\_page\_url\_not\_contains() (in module steps\_then), 12  
check\_property\_is() (in module steps\_given), 6  
check\_property\_is() (in module steps\_then), 13  
check\_property\_is\_not() (in module steps\_given), 6  
check\_property\_is\_not() (in module steps\_then), 13  
check\_title\_contains() (in module steps\_then), 13  
check\_title\_is() (in module steps\_then), 13  
check\_title\_is\_not() (in module steps\_then), 13  
check\_title\_not\_contains() (in module steps\_then), 13  
check\_within\_viewport() (in module steps\_then), 13  
click\_element() (in module steps\_when), 14  
click\_on\_element() (in module steps\_custom), 2  
click\_on\_shadow\_element() (in module steps\_custom), 3  
close\_all\_but\_first\_tab() (in module steps\_given), 6  
close\_current\_tab() (browser.BrowserActions method), 1  
close\_last\_opened\_window() (in module steps\_when), 14  
close\_tab() (browser.BrowserActions method), 1  
close\_tab\_at\_index() (browser.BrowserActions method), 1  
compare\_search\_result() (in module steps\_custom), 3  
contains\_any\_text() (in module steps\_given), 6  
contains\_text() (in module steps\_given), 6

## D

dbl\_click\_element() (in module steps\_when), 14  
deselect\_option\_by\_value() (in module steps\_when), 15  
deselect\_option\_by\_visible\_text() (in module steps\_when), 15  
deselect\_option\_index() (in module steps\_when), 15  
dismiss\_modal() (in module steps\_when), 15  
does\_not\_contain\_any\_text() (in module steps\_given), 7  
does\_not\_contain\_text() (in module steps\_given), 7  
drag\_and\_drop\_element() (in module steps\_when), 15

## E

element\_displayed() (in module steps\_given), 7  
element\_displayed() (in module steps\_then), 13

element\_enabled() (in module steps\_given), 7  
element\_equals\_text() (in module steps\_given), 7  
element\_exists() (in module steps\_given), 7  
element\_is\_displayed() (in module steps\_custom), 3  
element\_is\_hidden() (in module steps\_custom), 3  
element\_not\_displayed() (in module steps\_given), 7  
element\_not\_displayed() (in module steps\_then), 14  
element\_not\_enabled() (in module steps\_given), 7  
element\_not\_equals\_text() (in module steps\_given), 8  
element\_not\_exists() (in module steps\_given), 8  
element\_not\_selected() (in module steps\_given), 8  
element\_selected() (in module steps\_given), 8  
element\_text() (in module steps\_custom), 3

## F

fetch\_cookies() (in module steps\_when), 15

## G

generate\_pdf() (in module steps\_custom), 3  
generate\_pdfs() (in module steps\_custom), 3  
get\_current\_tab\_name() (browser.BrowserActions method), 1  
get\_current\_url() (browser.BrowserActions method), 1  
get\_number\_of\_tabs() (browser.BrowserActions method), 1  
get\_page\_source() (browser.BrowserActions method), 1  
get\_tab\_list() (browser.BrowserActions method), 1  
get\_title() (browser.BrowserActions method), 2  
given\_page\_url\_is() (in module steps\_given), 8  
given\_page\_url\_is\_not() (in module steps\_given), 8  
go\_to\_protected\_url() (browser.BrowserActions method), 2  
go\_to\_url() (browser.BrowserActions method), 2

## H

header\_menu\_button\_style() (in module steps\_custom), 3  
hover\_over\_element() (in module steps\_custom), 4  
hover\_over\_shadow\_element() (in module steps\_custom), 4

**J**

`js_click_element()` (in module `steps_when`), 15

**M**

module

- `browser`, 1
- `steps_custom`, 2
- `steps_given`, 5
- `steps_then`, 9
- `steps_when`, 14

`move_to_element_by_offset()` (in module `steps_when`), 15

**N**

`navigate_back()` (`browser.BrowserActions` method), 2

`navigate_to_external_page()` (in module `steps_custom`), 4

`nextgen_simple_card_style()` (in module `steps_custom`), 4

**O**

`open_external_webpage()` (in module `steps_given`), 8

`open_webpage()` (in module `steps_given`), 8

**P**

`page_loaded()` (in module `steps_custom`), 4

`page_scroll_position()` (in module `steps_custom`), 4

`page_title()` (in module `steps_given`), 8

`page_visual_is_valid()` (in module `steps_custom`), 4

`pause_execution()` (in module `steps_when`), 16

`press_button()` (in module `steps_when`), 16

**R**

`refresh()` (`browser.BrowserActions` method), 2

`resize_to()` (`browser.BrowserActions` method), 2

**S**

`scroll_to_element()` (in module `steps_when`), 16

`select_option_by_index()` (in module `steps_when`), 16

`select_option_by_value()` (in module `steps_when`), 16

`select_option_by_visible_text()` (in module `steps_when`), 16

`set_add_element_value()` (in module `steps_when`), 16

`set_element_value()` (in module `steps_when`), 16

`set_locale()` (in module `steps_when`), 17

`sitemap_urls_loading()` (in module `steps_cus-`

`tom`), 4

`steps_custom`  
module, 2

`steps_given`  
module, 5

`steps_then`  
module, 9

`steps_when`  
module, 14

`switch_to_last()` (in module `steps_when`), 17

`switch_to_tab()` (`browser.BrowserActions` method), 2

`switch_to_tab_index()` (`browser.BrowserActions` method), 2

**T**

`take_element_screenshot_with_actions()` (in module `steps_custom`), 5

`take_full_page_screenshot_from_sitemap_with_actions()` (in module `steps_custom`), 5

**U**

`urls_from_sitemap()` (in module `steps_custom`), 5

**W**

`wait_for_displayed()` (in module `steps_then`), 14

`wait_for_not_displayed()` (in module `steps_then`), 14

`window_size()` (in module `steps_given`), 9

