In **Katalon Studio**, there are several browser-specific and window-specific keywords that help you interact with web browsers and windows during automation testing.

Browser Keywords:

These keywords allow you to interact with the browser itself.

1. **Open Browser**
    - Opens a new browser instance.
    - Example: WebUI.openBrowser(https://katalon-demo-cura.herokuapp.com/’)
2. **Close Browser**
    - Closes the current browser instance.
    - Example: WebUI.closeBrowser()
3. **Navigate To Url**
    - Navigates the current browser to a specified URL.
    - Example: WebUI.navigateToUrl(‘https://katalon-demo-cura.herokuapp.com/’)
4. **Maximize Browser Window**
    - Maximizes the browser window to full screen.
    - Example: WebUI.maximizeWindow()
5. **Minimize Browser Window**
    - Minimizes the browser window.
    - Example: WebUI.minimizeWindow()
6. **Set Browser Size**
    - Sets the browser's window size (width and height).
    - Example: WebUI.setBrowserSize(1024, 768)
7. **Get Browser Title**
    - Retrieves the title of the current browser window.
    - Example: String title = WebUI.getWindowTitle()
8. **Get Browser Url**
    - Retrieves the current URL in the browser.
    - Example: String url = WebUI.getUrl()
9. **Wait For Browser Loading**
    - Waits for the browser to fully load.
    - Example: WebUI.waitForBrowserToLoad(10) (waits for up to 10 seconds)

Window Keywords:

These keywords allow you to interact with browser windows or tabs, especially in multi-window scenarios.

1. **Switch To Window**
    - Switches to a specific window or tab by its title.
    - Example: WebUI.switchToWindowTitle('Google Chrome')
2. **Switch To Window Index**
    - Switches to a specific window based on the window index.
    - Example: WebUI.switchToWindowIndex(1) (switches to the second window/tab)
3. **Close Window**
    - Closes the current window or tab.
    - Example: WebUI.closeWindowIndex(1) (closes the second window/tab)
4. **Get Window Title**
    - Retrieves the title of the current window.
    - Example: String windowTitle = WebUI.getWindowTitle()
5. **Get Window Handle**
    - Retrieves the handle (identifier) of the current window.
    - Example: String handle = WebUI.getWindowHandle()
6. **Get All Window Handles**
    - Retrieves handles of all open windows.
    - Example: List<String> handles = WebUI.getAllWindowHandles()
7. **Wait For Window**
    - Waits for a specific window to appear, specified by the window index or title.
    - Example: WebUI.waitForWindowTitle('New Tab', 10) (waits for a window with the title "New Tab" to appear for 10 seconds)

Checkbox-related keywords:

These keywords allow you to check, uncheck, verify, and validate the state of checkboxes during automation testing.

Here are the most commonly used **checkbox-related keywords** in Katalon Studio:

1. checkCheckbox

This keyword checks (selects) a checkbox.

Example:

WebUI.checkCheckbox(findTestObject('Object Repository/checkbox_Agree'))

2. uncheckCheckbox

This keyword unchecks (deselects) a checkbox.

Example:

WebUI.uncheckCheckbox(findTestObject('Object Repository/checkbox_Agree'))

3. verifyCheckboxChecked

This keyword verifies that a checkbox is checked (selected).

Example:

WebUI.verifyCheckboxChecked(findTestObject('Object Repository/checkbox_Example'))

4. verifyCheckboxNotChecked

This keyword verifies that a checkbox is not checked (deselected).

Example:

WebUI.verifyCheckboxChecked(findTestObject('Object Repository/checkbox_Agree'))

5. setCheckbox

This keyword checks or unchecks a checkbox depending on the value of the check parameter (true to check, false to uncheck).

Example:

- WebUI.setCheckbox(findTestObject('Object Repository/checkbox_ Agree '), true) // To check
- WebUI.setCheckbox(findTestObject('Object Repository/checkbox_ Agree '), false) // To uncheck

Notes:

- **Test Object**: You will need to specify a valid test object (like a checkbox) from your Object Repository using findTestObject().
- **Verifying Checkbox States**: verifyCheckboxChecked and verifyCheckboxNotChecked are typically used in assertions to confirm that the checkbox is in the expected state.

In Katalon Studio, Combo Boxes (also known as dropdowns or select elements) using a variety of keywords. These keywords allow you to select options, verify selected options, and handle different interactions with dropdown lists.

Here are the most commonly used **Combo Box-related keywords** in Katalon Studio:

1. selectOptionByValue

This keyword selects an option in a combo box by the value attribute of the option element.

Example:

WebUI.selectOptionByValue(findTestObject('Object Repository/comboBox_Country'), 'US', false)

- The second parameter is the value of the option to be selected.
- The third parameter (false or true) indicates whether to trigger the onChange event after selecting.

2. selectOptionByLabel

This keyword selects an option in a combo box by its visible label text.

Example:

WebUI.selectOptionByLabel(findTestObject('Object Repository/comboBox_Country'), 'United States', false)

- The second parameter is the label text of the option to be selected.

3. selectOptionByIndex

This keyword selects an option in a combo box by the index (position) of the option.

Example:

WebUI.selectOptionByIndex(findTestObject('Object Repository/comboBox_Country'), 3)

- The second parameter is the index (starting from 0) of the option to be selected.

4. getSelectedOption

This keyword retrieves the currently selected option's text from the combo box.

Example:

String selectedCountry = WebUI.getSelectedOption(findTestObject('Object Repository/comboBox_Country'))

println("Selected Country: " + selectedCountry)

5. verifySelectedOptionByValue

This keyword verifies if the selected option's value matches the expected value.

Example:

WebUI.verifySelectedOptionByValue(findTestObject('Object Repository/comboBox_Country'), 'US', false)

6. verifySelectedOptionByLabel

This keyword verifies if the selected option's label matches the expected label text.

Example:

WebUI.verifySelectedOptionByLabel(findTestObject('Object Repository/comboBox_Country'), 'United States', false)

7. verifySelectedOptionByIndex

This keyword verifies if the selected option matches the expected index in the combo box.

Example:

WebUI.verifySelectedOptionByIndex(findTestObject('Object Repository/comboBox_Example'), 2)

In **Katalon Studio**, can interact with **Textboxes** (input fields) using various keywords. These keywords allow you to enter text, clear text, and verify text values in input fields during automation testing.

Here are the most commonly used **Textbox-related keywords** in Katalon Studio:

1. setText

This keyword enters (sets) text into a textbox or input field.

Example:

WebUI.setText(findTestObject('Object Repository/textBox_Username'), 'myUsername')

- This will set the text '**myUsername**' in the textbox located at textBox_Username.

2. clearText

This keyword clears the text already present in a textbox or input field.

Example:

WebUI.clearText(findTestObject('Object Repository/textBox_Username'))

- This will clear the current text in the textbox located at textBox_Username.

3. getText

This keyword retrieves the text currently present in a textbox or input field.

Example:

String text = WebUI.getText(findTestObject('Object Repository/textBox_Username'))
println("Textbox Value: " + text)

- This will get the current text value from the textbox and store it in the variable text.

4. verifyText

This keyword verifies that the text in a textbox matches the expected value.

Example:

WebUI.verifyText(findTestObject('Object Repository/textBox_Username'), 'expectedUsername')

- This will verify if the current value in the textbox is equal to expectedUsername.

5. verifyTextNotEqual

Description: This keyword verifies that the text in a textbox is not equal to the expected value.

Example:

WebUI.verifyTextNotEqual(findTestObject('Object Repository/textBox_Username'), 'unexpectedUsername')

In **Katalon Studio**, handling **file uploads** can be accomplished through several keywords designed to interact with file upload dialogs or fields on web pages. The most common method for file uploading is to interact with an `<input type="file">` element, where you can use the **`setText`** keyword to specify the file path.

File Upload Keywords in Katalon Studio:

Here are the key **file upload-related keywords** in Katalon Studio:

1. setText (For File Upload Fields)

This keyword is used to set the file path into a file upload field (an input element with type="file").

Example:

WebUI.setText(findTestObject('Object Repository/fileUpload_Field'), 'C:\\path\\to\\your\\file.txt')

- This will simulate the action of uploading a file by setting the file path into the file input field.
- You must provide the absolute path to the file you want to upload (e.g., C:\\path\\to\\file.txt).

2. `uploadFile`

This keyword uploads a file to a file input field.

Example:

```
WebUI.uploadFile(findTestObject('Object
Repository/fileUpload_Field'), 'C:\\path\\to\\your\\file.txt')
```

- This is a more specialized keyword compared to `setText` and is more convenient for file upload scenarios. It simulates the file selection process in a file upload dialog.

Example Usage for File Upload:

Using setText:

// Set the file path in a file input field
WebUI.setText(findTestObject('Object Repository/fileUpload_Field'), 'C:\\path\\to\\your\\file.txt')

Using uploadFile:

// Upload the file to the file input field
WebUI.uploadFile(findTestObject('Object Repository/fileUpload_Field'), 'C:\\path\\to\\your\\file.txt')

File Path Format:

- Ensure the file path is absolute, especially if running the tests on different machines.
- Use double backslashes \\ to separate directories in the path (e.g., C:\\path\\to\\file.txt).

Verifying File Upload:

While Katalon doesn't have direct keywords to verify the success of a file upload (such as confirming that a file has been uploaded to the server), you can perform indirect checks such as:

- Verifying the appearance of a success message.
- Verifying that the file name appears in a confirmation dialog or within a list.
- Checking if the file exists on the server or file system after the upload.

Example for Verification (indirect check):

```
// Verify that the file name appears in a confirmation field after uploading
String uploadedFileName = WebUI.getText(findTestObject('Object
Repository/fileUpload_SuccessMessage'))
WebUI.verifyMatch(uploadedFileName, 'file.txt', false)
```

In **Katalon Studio**, capturing screenshots during test execution is a useful feature for debugging and reporting purposes. Katalon provides specific keywords to capture screenshots and save them for further analysis.

Here are the **screenshot-related keywords** in Katalon Studio:

1. takeScreenshot

Description: This keyword captures a screenshot of the current page and saves it in the default screenshots folder (<Katalon Studio Workspace>/Reports).

Example:

```
WebUI.takeScreenshot()
```

- This will capture the screenshot of the current screen and save it with a timestamp in the default folder.

2. takeScreenshotAsCheckpoint

This keyword captures a screenshot and stores it as a checkpoint in your test. This can be used for visual regression testing to compare screenshots.

Example:

```
WebUI.takeScreenshotAsCheckpoint('Checkpoint_Example')
```

- This will take a screenshot and save it as Checkpoint_Example in the **Katalon Studio checkpoints** directory.

3. takeScreenshot with File Path

This keyword allows you to capture a screenshot and save it to a specific location with a custom file name

Example:

```
WebUI.takeScreenshot('C:\\path\\to\\your\\custom\\folder\\screenshot.png')
```

- This will save the screenshot in the specified location with the file name screenshot.png.

.

In **Katalon Studio**, you can use specific **scrolling** keywords to automate scrolling actions, such as scrolling down or up on a webpage. These actions are useful for interacting with elements that are not visible in the current viewport (i.e., you may need to scroll to them to interact with them).

Here are the most commonly used **scroll down and scroll up** related keywords in **Katalon Studio**:

1. scrollToElement

This keyword scrolls the page until the specified element is in view.

Example:

```
WebUI.scrollToElement(findTestObject('Object Repository/elementToScrollTo'), 10)
```

- This will scroll the page to the element specified by findTestObject('Object Repository/elementToScrollTo') within **10 seconds**.

2. scrollToPosition

This keyword scrolls the page to a specific position by providing pixel values for the **X (horizontal)** and **Y (vertical)** coordinates.

Example:

```
WebUI.scrollToPosition(0, 500)
```

- This scrolls the page to the **vertical position 500px** (Y-axis). You can also use this to scroll horizontally by setting the **X-coordinate** (for example, scrollToPosition(200, 0) to scroll horizontally).

3. scrollToBottom

This keyword scrolls the page to the bottom.

Example:

```
WebUI.scrollToBottom()
```

- This will scroll to the very bottom of the page, which is useful for testing pages with infinite scrolling or ensuring all elements are loaded.

4. scrollToTop

This keyword scrolls the page to the top.

Example: WebUI.scrollToTop()

- This will scroll the page back to the top, useful when you need to reset the scroll position.

5. sendKeys (Alternative to Scroll)

You can also simulate pressing the **Page Down** or **Arrow Down** keys to scroll down the page.

Example:

```
WebUI.sendKeys(findTestObject('Object Repository/PageObject'),
Keys.chord(Keys.PAGE_DOWN))
```

- This simulates pressing the **Page Down** key to scroll down.

In **Katalon Studio**, you can interact with **alerts, pop-ups, and frames** using several specialized keywords. These keywords allow you to handle browser alerts (like JavaScript alerts), manage frames (like `<iframe>` elements), and switch between them during automated tests.

1. Handling Alerts

Alerts (like **JavaScript Alerts**, **Confirm dialogs**, and **Prompt dialogs**) are common in web applications, and Katalon Studio provides several keywords to interact with them.

i. acceptAlert

Accepts (clicks "OK") on the alert box.
Example:
WebUI.acceptAlert()

- This will accept the alert (clicking "OK").dismissAlert

Dismisses (clicks "Cancel") on the alert box.
Example:
WebUI.dismissAlert()

- This will dismiss the alert (clicking "Cancel").

ii. getAlertText

Retrieves the text displayed in the alert.
Example:

```
String alertText = WebUI.getAlertText()
println("Alert text: " + alertText)
```

- This will get the text of the alert and store it in a variable.

iii. verifyAlertPresent

Verifies if an alert is present on the page.
Example:
WebUI.verifyAlertPresent()

- This will verify if the alert is present on the page and raise an exception if it is not.

iv. verifyAlertNotPresent

Description: Verifies that no alert is present on the page.
Example:
WebUI.verifyAlertNotPresent()

- This will verify that no alert is present on the page.

v. setAlertText

Description: Sends text to a prompt alert (for alerts that ask for user input).
Example:
WebUI.setAlertText('Test input')

- This will set the text in the prompt alert.

2. Handling Frames (iFrames)

Frames (like elements) are commonly used for embedding content within a page. Katalon Studio provides keywords to switch to and from frames so you can interact with elements inside them.

i. switchToFrame

Switches the focus to the specified iframe or frame element.
Example:
WebUI.switchToFrame(findTestObject('Object Repository/frame_Example'), 10)

- This switches to the iframe specified by the test object within **10 seconds**.

ii. switchToDefaultContent

Switches back to the default page (i.e., exits the current frame or iframe and returns to the main content).
Example:
WebUI.switchToDefaultContent()

- This will switch focus back to the main document after interacting with a frame.

iii. switchToWindowIndex

Switches between browser windows by index, useful if frames are embedded in multiple windows.

Example:
WebUI.switchToWindowIndex(1)

- This switches to the second browser window (index starts from 0).

iv. switchToWindowTitle

Description: Switches between browser windows by the window title.
Example:
WebUI.switchToWindowTitle('New Window Title')

- This switches to the window with the title 'New Window Title'.

v. getWindowTitle

Retrieves the title of the current window.
Example:

```
String windowTitle = WebUI.getWindowTitle()
println("Current Window Title: " + windowTitle)
```

In **Katalon Studio**, mouse hover actions are useful for interacting with elements that become visible or change behaviour when hovered over, such as drop-down menus, tooltips, or hover effects.

1. mouseOver

This keyword simulates moving the mouse cursor over a specified element. It is typically used to trigger hover effects like showing hidden menus or tooltips.
Example:
WebUI.mouseOver(findTestObject('Object Repository/hoverElement'))

- This will simulate hovering the mouse over the element defined by the test object hoverElement.
-

2. mouseOverOffset

This keyword allows you to simulate a mouse hover at a specific offset (relative to the element's position). You can specify X and Y offsets to hover at a specific point within the element.
Example:
WebUI.mouseOverOffset(findTestObject('Object Repository/hoverElement'), 10, 20)

- This will move the mouse cursor to the position 10 pixels right and 20 pixels down from the top-left corner of the specified element.

In **Katalon Studio**, you can automate **drag and drop** actions to test user interactions involving moving elements on the page. This is useful for testing features like **reordering items**, **dragging files to upload zones**, or **dragging elements to a different location on the UI**.

Katalon Studio provides several **drag-and-drop** related keywords to perform these actions.

1. dragAndDropToObject

This keyword simulates dragging an element and dropping it onto another element.

Example:

```
WebUI.dragAndDropToObject(findTestObject('Object Repository/dragElement'),
findTestObject('Object Repository/dropTarget'))
```

- This will drag the element specified by dragElement and drop it onto the dropTarget.

2. dragAndDropByOffset

This keyword simulates dragging an element by a specific **offset** (in pixels) from its current position and dropping it at the new position.

Example:

```
WebUI.dragAndDropByOffset(findTestObject('Object Repository/dragElement'), 100, 200)
```

- This will drag the dragElement by **100 pixels horizontally** and **200 pixels vertically** and drop it at the new position.