

# Selenium WebDriver Actions

Press Space for next page →



# Agenda

1. WebDriver Actions in Selenium
2. Double Click: Code Example
3. Right Click: Code Example
4. Jquery - Draggable
5. Jquery - Draggable
6. Drag and Drop(Manual) : Code Example
7. Drag and Drop(Auto) : Code Example
8. Draggable: Code Example
9. Keyboard Actions
10. Keyboard Actions: Code Example
11. Keyboard Actions: Code Snippet
12. Summery
13. Mouse Hover
14. Mouse Hover: Code Example

# WebDriver Actions in Selenium

The Actions class in Selenium WebDriver is a powerful feature used to automate complex user interactions on web applications. This class is part of the `org.openqa.selenium.interactions` package and allows you to perform advanced user interactions like mouse movements, keypresses, drag-and-drop operations, and other interactions that are not possible using the basic WebDriver methods.

## Examples:

- Mouse Actions
- Keyboard Actions
- Drag & Drop actions

Documentation: [Selenium WebDriver Actions](#)

# Double Click: Code Example

```
1  class DoubleClickExample {
2      public static void main(String[] args) {
3          WebDriver driver = new ChromeDriver();
4
5          driver.get("https://api.jquery.com/dblclick/");
6
7          driver.switchTo().frame(driver.findElement(By.xpath("//iframe")));
8
9          Actions actions = new Actions(driver);
10
11         WebElement squareBox = driver.findElement(By.xpath("//span[text()='Double click the block']//parent::body/div"));
12         actions.doubleClick(squareBox).perform();
13
14         Thread.sleep(3000);
15
16         System.out.println("Yeeee I double clicked the element !!! wowwww !!!");
17         driver.quit();
18     }
19 }
```

# Right Click: Code Example

```
1  class RightClickExample {
2      public static void main(String[] args) {
3          WebDriver driver = new ChromeDriver();
4
5          driver.get("https://swisnl.github.io/jquery-contextMenu/demo.html");
6          driver.manage().window().maximize();
7
8          Actions actions = new Actions(driver);
9
10         actions.contextClick(driver.findElement(By.xpath("//span[contains(text(),'right click me')]"))).perform();
11
12         driver.findElement(By.xpath("//li[@class='context-menu-item context-menu-icon context-menu-icon-copy']")).click();
13         System.out.println(driver.switchTo().alert().getText());
14
15         Thread.sleep(3000);
16
17         driver.switchTo().alert().accept();
18
19         Thread.sleep(3000);
20
21         driver.quit();
22     }
23 }
```

# Jquery - Draggable

Functionality: Allows elements to be clicked and dragged using the mouse.

## Common Use Cases:

- Rearranging items on a page.
- Implementing custom sliders.
- Creating drag-and-drop interfaces.

Documentation: [Jquery Draggable](#)

# Jquery - Draggable

Functionality: Designates elements as targets for draggable items.

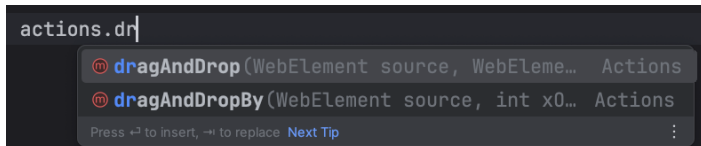
## Common Use Cases:

- Building drag-and-drop file uploaders.
- Creating interactive games.
- Developing intuitive UI components.

## Playground

URL: 1

URL: 2



# Drag and Drop(Manual) : Code Example

```
1  class DragAndDropManualExample {
2      public static void main(String[] args) {
3          WebDriver driver = new ChromeDriver();
4
5          driver.get("https://dhtmlx.com/docs/products/dhtmlxTree/");
6          driver.manage().window().maximize();
7
8          driver.switchTo().frame(driver.findElement(By.xpath("//iframe[@class='js-iframe active']")));
9
10         Actions actions = new Actions(driver);
11
12         WebElement src = driver.findElement(By.xpath("//span[text()='Lawrence Block']"));
13         WebElement dest = driver.findElement(By.xpath("//li[text()='Zend Framework in Action']"));
14
15         actions.clickAndHold(src)
16             .pause(Duration.ofSeconds(2))
17             .moveToElement(dest)
18             .pause(Duration.ofSeconds(2))
19             .release()
20             .build()
21             .perform();
22
23         driver.quit();
24     }
25 }
```



# Drag and Drop(Auto) : Code Example

```
1  class DragAndDropAutoExample {
2      public static void main(String[] args) {
3          WebDriver driver = new ChromeDriver();
4
5          driver.get("https://dhtmlx.com/docs/products/dhtmlxTree/");
6          driver.manage().window().maximize();
7
8          driver.switchTo().frame(driver.findElement(By.xpath("//iframe[@class='js-iframe active']")));
9
10         WebElement src = driver.findElement(By.xpath("//span[text()='Lawrence Block']"));
11         WebElement dest = driver.findElement(By.xpath("//li[text()='Zend Framework in Action']"));
12
13         Actions actions = new Actions(driver);
14         actions.dragAndDrop(src, dest).perform();
15
16         driver.quit();
17     }
18 }
```

# Draggable: Code Example

```
1  class DraggableExample {
2      public static void main(String[] args) {
3          WebDriver driver = new ChromeDriver();
4
5          driver.get("https://jqueryui.com/draggable/");
6          driver.manage().window().maximize();
7          driver.switchTo().frame(driver.findElement(By.tagName("iframe")));
8
9          Actions actions = new Actions(driver);
10         WebElement src = driver.findElement(By.xpath("//div[@id='draggable']"));
11
12         actions.dragAndDropBy(src, 394, 16).perform();
13     }
14 }
```

# Keyboard Actions

Keyboard Actions: Enables simulation of keyboard events like key presses and releases.

## Common Keyboard Actions:

- `sendKeys()`: Sends a sequence of keystrokes to an element.
- `keyDown()`: Simulates pressing a key down.
- `keyUp()`: Simulates releasing a key.

# Keyboard Actions: Code Example

```
1  class KeyboardActionsExample {
2      public static void main(String[] args) {
3          WebDriver driver = new ChromeDriver();
4
5          driver.get("http://www.google.com");
6
7          driver.manage().window().maximize();
8
9          Actions actions =new Actions(driver);
10
11         actions.click(driver.findElement(By.xpath("//textarea[@name='q']")))
12             .sendKeys("Nexxvali")
13             .pause(Duration.ofSeconds(2))
14             .sendKeys(Keys.ARROW_DOWN)
15             .pause(Duration.ofSeconds(2))
16             .sendKeys(Keys.ENTER)
17             .build()
18             .perform();
19
20     }
21 }
```

# Keyboard Actions: Code Snippet

## *Selecting all text*

```
1 Actions actions = new Actions(driver);
2 actions.click(inputField)
3     .keyDown(Keys.CONTROL)
4     .sendKeys("a")
5     .keyUp(Keys.CONTROL)
6     .build()
7     .perform();
```

## *Copy-Paste Text*

```
1 Actions actions = new Actions(driver);
2 actions.click(inputField)
3     .keyDown(Keys.CONTROL)
4     .sendKeys("a")
5     .sendKeys("c")
6     .keyUp(Keys.CONTROL)
7     .sendKeys(Keys.TAB)
8     .keyDown(Keys.CONTROL)
9     .sendKeys("v")
10    .keyUp(Keys.CONTROL)
11    .build()
12    .perform();
```

# Summery

- Action Class: Essential for handling complex keyboard interactions.
- Key Methods: `sendKeys()`, `keyDown()`, `keyUp()`.
- Practical Use: Automating tasks like text input, selection, and copy-paste.

Playground: URL

# Mouse Hover

- Mouse Hover: Simulates mouse movement to an element without clicking it.
- Common Use Case: To reveal hidden elements or trigger hover-dependent events (like dropdown menus).
- Action Class
- Purpose: Provides methods to handle complex user gestures.
- Key Method: `moveToElement()` – moves the mouse to the middle of the element.

# Mouse Hover: Code Example

```
1  class MouseHoverExample {
2      public static void main(String[] args) {
3          WebDriver driver = new ChromeDriver();
4
5          driver.get("http://seleniumpractise.blogspot.com/2016/08/how-to-perform-mouse-hover-in-selenium.html");
6
7          driver.manage().window().maximize();
8
9          WebElement hoverButton = driver.findElement(By.xpath("//button [text()='Automation Tools']"));
10
11         Actions act = new Actions(driver);
12         act.moveToElement(hoverButton).perform();
13
14         List<WebElement> links = driver.findElements(By.xpath("//div[@class='dropdown-content']/a"));
15         int total_count = links.size();
16
17         for (int i = 0; i < total_count; i++) {
18
19             WebElement element = links.get(i);
20             String text = element.getAttribute("href");
21             System.out.println("Link: " + text);
22         }
23     }
24 }
```



Thank you 

 [qa-june-2024-automation-with-java-slides](#)