

Selenium Mouse Keyboard Actions

In Selenium with Java, **Actions class** is used to handle advanced user interactions like **mouse movements**, **keyboard actions**, **click and hold**, **drag and drop**, and **composite actions**. It is part of the `org.openqa.selenium.interactions` package.

✓ Import the Required Classes

```
java

import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
```

Mouse Actions using Actions Class

1. Move to Element (Mouse Hover)

```
java

Actions actions = new Actions(driver);
actions.moveToElement(element).perform();
```

2. Right Click (Context Click)

```
java

actions.contextClick(element).perform();
```

3. Double Click

```
java

actions.doubleClick(element).perform();
```

4. Click and Hold

```
java

actions.clickAndHold(element).perform();
```

5. Release (after clickAndHold)

```
java

actions.release().perform();
```

6. Drag and Drop

```
java

actions.dragAndDrop(sourceElement, targetElement).perform();
```

7. Drag and Drop by Offset

```
java

actions.dragAndDropBy(sourceElement, xOffset, yOffset).perform();
```

Keyboard Actions using Actions Class

1. Send Keys to Element

```
java

actions.sendKeys(element, "text").perform();
```

2. Key Down and Key Up

```
java

// Example: Press SHIFT and type in capital
actions.keyDown(Keys.SHIFT)
    .sendKeys("hello")
    .keyUp(Keys.SHIFT)
    .perform();
```

3. Send Keys Without Focusing an Element

```
java

actions.sendKeys(Keys.ENTER).perform();
```

Composite Actions

You can **chain multiple actions** together before calling `.perform()`:

```
java

actions.moveToElement(element)
    .click()
    .sendKeys("Selenium")
```

```
.build()  
.perform();
```

`.build()` is optional unless you're combining multiple actions.

Example: Hover and Click Submenu

java

```
WebElement menu = driver.findElement(By.id("menu"));  
WebElement subMenu = driver.findElement(By.id("submenu"));  
  
Actions actions = new Actions(driver);  
actions.moveToElement(menu).pause(1000).click(subMenu).perform();
```

Best Practices

- Always call `.perform()` to execute the action.
- Use `.build()` when chaining multiple actions before `.perform()`.
- Use `Thread.sleep()` or `.pause()` if UI animations are involved.

Would you like a hands-on mini project or lab example using Actions class?