

SELENIUM WEB AUTOMATION

Browser Setup

- **Using System Property method**

```
System.setProperty("webdriver.chrome.driver", "path/to/chromedriver"); WebDriver driver = new ChromeDriver();
```

- **Using WebDriverManager method**

```
WebDriverManager.chromedriver().setup(); WebDriver driver = new ChromeDriver();
```

Basic Browser Actions

```
driver.get(url); // Open URL
```

```
driver.navigate().to("https://another.com"); // Navigate to another URL
```

```
driver.navigate().back(); // Go back
```

```
driver.navigate().forward(); // Go forward
```

```
driver.navigate().refresh(); // Refresh the page
```

```
driver.getTitle(); // Get the title
```

```
driver.getCurrentUrl(); // Get the current URL
```

```
driver.getPageSource(); // Get the page source
```

Locators

```
WebElement element = driver.findElement(By.id("id"));
```

```
WebElement element = driver.findElement(By.name("name"));
```

```
WebElement element = driver.findElement(By.className("class"));
```

```
WebElement element = driver.findElement(By.tagName("tag"));
```

```
WebElement element = driver.findElement(By.linkText("Exact Link Text"));
```

```
WebElement element = driver.findElement(By.partialLinkText("Partial Link"));
```

```
WebElement element = driver.findElement(By.cssSelector("css-selector"));
```

```
WebElement element = driver.findElement(By.xpath("xpath"));
```

// Find multiple elements

```
List elements = driver.findElements(By.tagName("tag"));
```

Web Element Actions

// Click an element

```
WebElement button = driver.findElement(By.id("buttonId")); button.click();
```

// Enter text

```
WebElement input = driver.findElement(By.id("inputId")); input.sendKeys("text");
```

// Clear input field

```
input.clear();
```

// Get text

```
String text = element.getText();
```

// Get attribute

```
String attribute = element.getAttribute("attributeName");
```

// Handling dropdown

```
WebElement dropdown = driver.findElement(By.id("dropdownId"));
```

```
Select select = new Select(dropdown);
```

// Select by visible text

```
select.selectByVisibleText("Option Text");
```

// Select by value

```
select.selectByValue("value");
```

// Select by index

```
select.selectByIndex(0);
```

// Get all options

```
List<WebElement> options = select.getOptions();
```

Handling Alerts

// Switch to alert

```
Alert alert = driver.switchTo().alert();
```

```
alert.accept(); // Accept alert
```

```
alert.dismiss(); // Dismiss alert
```

```
String alertText = alert.getText(); // Get alert text
```

```
alert.sendKeys("text"); // Send input to alert
```

Waits

- **Implicit wait**

```
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

- **Explicit wait**

```
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
```

```
WebElement e = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("id")));
```

Handling Frames

// Switch to frame by index

```
driver.switchTo().frame(0);
```

// Switch to frame by name or ID

```
driver.switchTo().frame("frameName");
```

// Switch to frame by WebElement

```
WebElement frameElement = driver.findElement(By.tagName("iframe"));
```

```
driver.switchTo().frame(frameElement);
```

// Switch back to default content

```
driver.switchTo().defaultContent();
```

Taking Screenshots

```
File screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);
```

```
FileUtils.copyFile(screenshot, new File("/screenshot.png"));
```

Assertions

// Assert conditions

```
Assert.assertEquals(actual, expected);
```

```
Assert.assertTrue(condition);
```

```
Assert.assertFalse(condition);
```

```
Assert.assertNotNull(object);
```

Handling windows

```
JavascriptExecutor js = (JavascriptExecutor) driver; // Execute JavaScript
```

```
js.executeScript("window.scrollTo(0,500)"); // Scroll down
```

```
js.executeScript("arguments[0].click();", element); // Click an element
```