



# Selenium WebDriver Breakdown with examples



## In Java



LinkedIn: Japneet Sachdeva

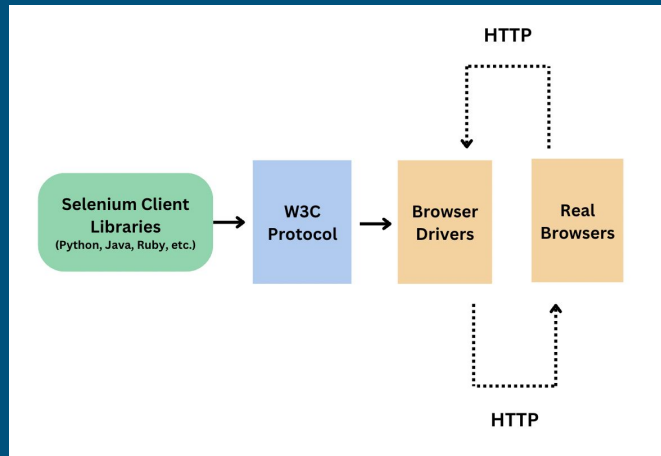
# Selenium WebDriver Architecture

The architecture of Selenium 4 is similar to Selenium 3, however it uses W3C protocol instead of JSON wire protocol for communication between Client Libraries and Browser Drivers.

1. W3C stands for the World Wide Web Consortium
2. Selenium adheres to the standards and specifications laid by the W3C for web automation
3. Compliance helps in making the communication easy and direct between the client libraries and the browser drivers

Read more here: [Link](#)

Image credits: BrowserStack



# Opening and Closing Browsers

---

```
WebDriver driver = new ChromeDriver(); // Open Chrome  
driver.get("https://www.example.com"); // Navigate to URL  
driver.quit(); // Close all browser windows  
driver.close(); // Close the current browser window
```

LinkedIn: Japneet Sachdeva

# Element Interaction

---

```
driver.findElement(By.id("username")).sendKeys("testuser"); // Enter text
```

```
driver.findElement(By.name("submit")).click(); // Click a button
```

```
driver.findElement(By.linkText("Home")).click(); // Click a link
```

```
driver.findElement(By.xpath("//div[@class='message']")).getText(); // Get text
```

LinkedIn: Japneet Sachdeva

# Navigation

---

```
driver.navigate().to("https://www.newsite.com"); // Navigate to a new URL
```

```
driver.navigate().back(); // Go back to the previous page
```

```
driver.navigate().forward(); // Go forward to the next page
```

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

LinkedIn: Japneet Sachdeva

# Window and Frame Handling

---

`driver.switchTo().window("windowName");` // Switch to a specific window

`driver.switchTo().frame("frameId");` // Switch to a frame

`driver.switchTo().defaultContent();` // Switch back to the main content

LinkedIn: Japneet Sachdeva

# Browser Management

---

```
driver.manage().window().maximize(); // Maximize the browser window
```

```
driver.manage().window().fullscreen(); // Set full-screen mode
```

```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS); // Set  
implicit wait
```

LinkedIn: Japneet Sachdeva

# Mouse and Keyboard Actions

---

```
Actions actions = new Actions(driver);  
actions.moveToElement(element).click().build().perform(); // Move to and click  
actions.doubleClick(element).perform(); // Double-click  
actions.dragAndDrop(source, target).perform(); // Drag and drop  
actions.sendKeys(element, "text").perform(); // Send keystrokes
```

LinkedIn: Japneet Sachdeva



# Assertions

---

```
Assert.assertTrue(element.isDisplayed()); // Assert element visibility
```

```
Assert.assertEquals(driver.getTitle(), "Expected Title"); // Assert title
```

By default these are hard assertions, means if these fail then tests will be exited or stopped

LinkedIn: Japneet Sachdeva

# Soft Assertions

---

```
SoftAssert softassert = new SoftAssert();  
  
//Soft assert applied to verify title  
  
softassert.assertEquals(ActualTitle, ExpectedTitle);  
  
softassert.assertAll();
```

By default these assertions, do not stop if these assertions fail & are reported in your test reports

LinkedIn: Japneet Sachdeva

# Explicit Wait

---

```
WebDriverWait wait = new WebDriverWait(driver, 10);  
wait.until(ExpectedConditions.elementToBeClickable(element)); // Explicit wait
```

These are conditional waits & can be applied to satisfy a particular condition, then continue test execution if condition met or failed if not met in mentioned amount of time

LinkedIn: Japneet Sachdeva

# Implicit Wait

---

```
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

Once the command is run, Implicit Wait remains for the entire duration for which the browser is open. It's default setting is 0, and the specific wait time needs to be set by the following protocol.

Implicit wait increases test script execution time

LinkedIn: Japneet Sachdeva

# Fluent Wait -

looks for a web element repeatedly at regular intervals until timeout happens or until the object is found

---

```
FluentWait wait = new FluentWait(driver);  
  
//Specify the timeout of the wait  
wait.withTimeout(5000, TimeUnit.MILLISECONDS);  
  
//Specify polling time  
wait.pollingEvery(250, TimeUnit.MILLISECONDS);  
  
//Specify what exceptions to ignore  
wait.ignoring(NoSuchElementException.class)  
  
//This is how we specify the condition to wait on.  
wait.until(ExpectedConditions.alertIsPresent());
```

LinkedIn: Japneet Sachdeva

# Alerts

---

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

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

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

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

LinkedIn: Japneet Sachdeva

# Getting Page Information

---

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

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

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

LinkedIn: Japneet Sachdeva

# Element Information

---

`element.isDisplayed();` // Check if an element is visible

`element.isEnabled();` // Check if an element is enabled

`element.isSelected();` // Check if an element is selected

`element.getAttribute("attribute_name");` // Get an element's attribute value

`element.getTagName();` // Get an element's tag name

LinkedIn: Japneet Sachdeva



# Menu Handling

---

```
Select select = new Select(element); // Create a Select object for dropdowns
```

```
select.selectByIndex(1); // Select an option by index
```

```
select.selectByValue("option2"); // Select an option by value
```

```
select.selectByVisibleText("Option 3"); // Select an option by visible text
```

LinkedIn: Japneet Sachdeva

# Keyboard Actions

---

```
actions.keyDown(Keys.CONTROL).sendKeys("a").keyUp(Keys.CONTROL).perform();  
// Ctrl+A
```

```
actions.keyDown(Keys.SHIFT).sendKeys("hello").keyUp(Keys.SHIFT).perform(); //  
Type "HELLO"
```

```
actions.sendKeys(Keys.ENTER).perform(); // Press Enter
```

LinkedIn: Japneet Sachdeva

# Mouse Actions

---

`actions.contextClick(element).perform(); // Right-click`

`actions.clickAndHold(element).perform(); // Click and hold`

`actions.release(element).perform(); // Release the mouse button`

LinkedIn: Japneet Sachdeva

# Taking Screenshots

---

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

LinkedIn: Japneet Sachdeva

# Switching to Iframes

---

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

LinkedIn: Japneet Sachdeva

# Scrolling

---

```
JavascriptExecutor js = (JavascriptExecutor)driver;  
js.executeScript("window.scrollTo(0, 1000)");
```

```
Actions actions = new Actions(driver);
```

```
// Scroll down a specific distance:
```

```
actions.moveToElement(element).clickAndHold().moveByOffset(0,  
500).release().build().perform();
```

LinkedIn: Japneet Sachdeva

# Cookie Management

---

Get all cookies: `Set<Cookie> cookies = driver.manage().getCookies();`

Get a specific cookie: `Cookie cookie = driver.manage().getCookieNamed("cookie_name");`

Add a cookie: `driver.manage().addCookie(new Cookie("cookie_name", "cookie_value"));`

Delete a cookie: `driver.manage().deleteCookieNamed("cookie_name");`

Delete all cookies: `driver.manage().deleteAllCookies();`

LinkedIn: Japneet Sachdeva

# Alert Handling (Extended)

---

Dismiss an alert: `alert.dismiss();`

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



# Page Source Interactions

---

Find elements in page source: `List<WebElement> elements = driver.findElements(By.xpath("//*"));`

Execute JavaScript code: `JavascriptExecutor js = (JavascriptExecutor)driver;`  
`js.executeScript("document.getElementById('myElement').click();");`

LinkedIn: Japneet Sachdeva

# Advanced Navigation

---

Refresh the page with hard reload: `driver.navigate().refresh(true);`

Navigate to a page relative to the current URL:

`driver.navigate().to("new_page.html");`

LinkedIn: Japneet Sachdeva

# Logging and Reporting

---

Enable logging: `System.setProperty("webdriver.chrome.logfile", "chromedriver.log");`

Create custom test reports: Utilize third-party libraries like ExtentReports or Allure or TestNG reports

LinkedIn: Japneet Sachdeva

# Headless Mode

---

Run tests without a visible browser: `ChromeOptions options = new ChromeOptions();`

`options.addArguments("--headless");`

`WebDriver driver = new ChromeDriver(options);`

LinkedIn: Japneet Sachdeva

# Setting browser capabilities

---

```
ChromeOptions options = new ChromeOptions();  
options.addArguments("--start-maximized");  
WebDriver driver = new ChromeDriver(options);
```

LinkedIn: Japneet Sachdeva

# Disabling images

---

```
options.addArguments("--disable-images");
```

LinkedIn: Japneet Sachdeva

# Enabling experimental features

---

```
options.setExperimentalOption("useAutomationExtension", false);
```

LinkedIn: Japneet Sachdeva

# Uploading Files

---

Using `sendKeys()` for file uploads:

```
WebElement uploadElement = driver.findElement(By.id("upload"));  
uploadElement.sendKeys("C:\\path\\to\\file.txt");
```

LinkedIn: Japneet Sachdeva



# Handling Shadow DOM Elements

---

Using `executeScript()` to interact with shadow DOM:

```
JavascriptExecutor js = (JavascriptExecutor)driver;
```

```
WebElement shadowRoot = (WebElement)js.executeScript("return  
document.querySelector('shadow-host').shadowRoot");
```

```
WebElement element =  
shadowRoot.findElement(By.id("element-inside-shadow-dom"));
```

LinkedIn: Japneet Sachdeva

# Cross-Browser Testing:

---

Using different WebDriver implementations for various browsers:

```
WebDriver driver = new FirefoxDriver();
```

```
WebDriver driver = new EdgeDriver();
```

LinkedIn: Japneet Sachdeva

# Different Browser Options

---

```
ChromeOptions options = new ChromeOptions();  
options.setCapability("browserName", "chrome");  
options.setCapability("version", "104.0");  
options.setCapability("platformName", "Windows 10");  
options.addArguments("--start-maximized");  
options.setCapability("acceptInsecureCerts", true);
```

```
WebDriver driver = new ChromeDriver(options);
```

LinkedIn: Japneet Sachdeva

# Refer to Selenium Documentation for insights

---

[Selenium Documentation Link](#)

Prepare for Test Automation Interviews for SDETs: [Link](#)

Complete Test Automation Frameworks, Design Patterns Implementation and Understanding for SDETs: [Link](#)

LinkedIn: Japneet Sachdeva