



## Initialize

```
// Maven: selenium-chrome-driver
import org.openqa.selenium.chrome.ChromeDriver;
WebDriver driver = new ChromeDriver();
// Maven: selenium-firefox-driver
import org.openqa.selenium.firefox.FirefoxDriver;
WebDriver driver = new FirefoxDriver();
// Maven: selenium-edge-driver
import org.openqa.selenium.firefox.EdgeDriver;
WebDriver driver = new EdgeDriver();
// Maven: selenium-ie-driver
import org.openqa.selenium.ie.InternetExplorerDriver;
WebDriver driver = new InternetExplorerDriver();
// Maven: selenium-safari-driver
import org.openqa.selenium.safari.SafariDriver;
WebDriver driver = new SafariDriver();
```

## Locators

```
driver.findElement(By.className("className"));
driver.findElement(By.cssSelector("css"));
driver.findElement(By.id("id"));
driver.findElement(By.linkText("text"));
driver.findElement(By.name("name"));
driver.findElement(By.partialLinkText("pText"));
driver.findElement(By.tagName("input"));
driver.findElement(By.xpath("//*[@id='editor']"));
// Find multiple elements
List<WebElement> anchors = driver.findElements(By.tagName("a"));
// Search for an element inside another
WebElement div = driver.findElement(By.tagName("div"))
    .findElement(By.tagName("a"));
```

## Basic Browser Operations

```
// Navigate to a page
driver.navigate().to("http://google.com");
// Get the title of the page
String title = driver.getTitle();
// Get the current URL
String url = driver.getCurrentUrl();
// Get the current page HTML source
String html = driver.getPageSource();
```

## Basic Elements Operations

```
WebElement element = driver.findElement(By.id("id"));
element.click();
element.sendKeys("someText");
element.clear();
element.submit();
String innerText = element.getText();
boolean isEnabled = element.isEnabled();
boolean isDisplayed = element.isDisplayed();
boolean isSelected = element.isSelected();
WebElement element = driver.findElement(By.id("id"));
Select select = new Select(element);
select.selectByIndex(1);
select.selectByVisibleText("Ford");
select.selectByValue("ford");
select.deselectAll();
select.deselectByIndex(1);
select.deselectByVisibleText("Ford");
select.deselectByValue("ford");
List<WebElement> allSelected = select.getAllSelectedOptions();
boolean isMultipleSelect = select.isMultiple();
```

## Advanced Element Operations

```
// Drag and Drop
WebElement element = driver.FindElement(
    By.xpath("//*[@id='project']/p[1]/div/div[2]"));
Actions move = new Actions(driver);
move.dragAndDropBy(element, 30, 0).build().perform();
// How to check if an element is visible
Assert.assertTrue(driver.findElement(
    By.xpath("//*[@id='tve_editor']/div")).isDisplayed());
// Upload a file
WebElement element = driver.findElement(By.id("RadUpload1file0"));
String filePath = "D:\\WebDriver.Series.Tests\\WebDriver.xml";
element.sendKeys(filePath);
// Scroll focus to control
WebElement link =
    driver.findElement(By.partialLinkText("Previous post"));
String js = String.format("window.scroll(0, %d);",
    link.getLocation().getY());
((JavascriptExecutor)driver).executeScript(js);
link.click();
// Taking an element screenshot
WebElement element =
    driver.findElement(By.xpath("//*[@id='tve_editor']/div"));
File screenshotFile =
    ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
BufferedImage fullImg = ImageIO.read(screenshotFile);
Point point = element.getLocation();
int elementWidth = element.getSize().getWidth();
int elementHeight = element.getSize().getHeight();
BufferedImage eleScreenshot = fullImg.getSubimage(point.getX(),
    point.getY(), elementWidth, elementHeight);
ImageIO.write(eleScreenshot, "png", screenshotFile);
String tempDir = getProperty("java.io.tmpdir");
File destFile = new File(Paths.get(tempDir, fileName +
    ".png").toString());
FileUtils.getFileUtils().copyFile(screenshotFile, destFile);
// Focus on a control
WebElement link =
    driver.findElement(By.partialLinkText("Previous post"));
Actions action = new Actions(driver);
action.moveToElement(link).build().perform();
// Wait for visibility of an element
WebDriverWait wait = new WebDriverWait(driver, 30);
wait.until(ExpectedConditions.visibilityOfAllElementsLocatedBy(
    By.xpath("//*[@id='tve_editor']/div[2]/div[2]/div/div")));
```

## Advanced Browser Operations

```
// Handle JavaScript pop-ups
Alert alert = driver.switchTo().alert();
alert.accept();
alert.dismiss();
// Switch between browser windows or tabs
Set<String> windowHandles = driver.getWindowHandles();
String firstTab = (String>windowHandles.toArray()[1];
String lastTab = (String>windowHandles.toArray()[2];
driver.switchTo().window(lastTab);
// Navigation history
driver.navigate().back();
driver.navigate().refresh();
driver.navigate().forward();
// Maximize window
driver.manage().window().maximize();
// Add a new cookie
Cookie newCookie = new Cookie("customName", "customValue");
driver.manage().addCookie(newCookie);
// Get all cookies
Set<Cookie> cookies = driver.manage().getCookies();
// Delete a cookie by name
driver.manage().deleteCookieNamed("CookieName");
// Delete all cookies
driver.manage().deleteAllCookies();
//Taking a full-screen screenshot
File srceenshotFile =
    ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
String tempDir = getProperty("java.io.tmpdir");
File destFile = new File(Paths.get(tempDir, fileName +
    ".png").toString());
FileUtils.getFileUtils().copyFile(srceenshotFile, destFile);
// Wait until a page is fully loaded via JavaScript
WebDriverWait wait = new WebDriverWait(driver, 30);
wait.until(x -> {
    ((String)((JavascriptExecutor)driver).executeScript(
        "return document.readyState")).equals("complete");
});
// Switch to frames
driver.switchTo().frame(1);
driver.switchTo().frame("frameName");
WebElement element = driver.findElement(By.id("id"));
driver.switchTo().frame(element);
// Switch to the default document
driver.switchTo().defaultContent();
```

## Advanced Browser Configurations

```
// Use a specific Firefox profile
ProfilesIni profile = new ProfilesIni();
FirefoxProfile firefoxProfile = profile.getProfile("ProfileName");
FirefoxOptions firefoxOptions = new FirefoxOptions();
firefoxOptions.setProfile(firefoxProfile);
driver = new FirefoxDriver(firefoxOptions);
// Set a HTTP proxy Firefox
ProfilesIni profile = new ProfilesIni();
FirefoxProfile firefoxProfile = new FirefoxProfile();
firefoxProfile.setPreference("network.proxy.type", 1);
firefoxProfile.setPreference("network.proxy.http", "myproxy.com");
firefoxProfile.setPreference("network.proxy.http_port", 3239);
FirefoxOptions firefoxOptions = new FirefoxOptions();
firefoxOptions.setProfile(firefoxProfile);
driver = new FirefoxDriver(firefoxOptions);
// Set a HTTP proxy Chrome
var proxy = new Proxy();
proxy.setProxyType(Proxy.ProxyType.MANUAL);
proxy.setAutodetect(false);
proxy.setSslProxy("127.0.0.1:3239");
ChromeOptions chromeOptions = new ChromeOptions();
chromeOptions.setProxy(proxy);
driver = new ChromeDriver(chromeOptions);
// Accept all certificates Firefox
FirefoxProfile firefoxProfile = new FirefoxProfile();
firefoxProfile.setAcceptUntrustedCertificates(true);
firefoxProfile.setAssumeUntrustedCertificateIssuer(false);
FirefoxOptions firefoxOptions = new FirefoxOptions();
firefoxOptions.setProfile(firefoxProfile);
driver = new FirefoxDriver(firefoxOptions);
// Accept all certificates Chrome
ChromeOptions chromeOptions = new ChromeOptions();
chromeOptions.addArguments("--ignore-certificate-errors");
driver = new ChromeDriver(chromeOptions);
// Set Chrome options
ChromeOptions chromeOptions = new ChromeOptions();
chromeOptions.addArguments("user-data-dir=C:\\Path\\To\\User Data");
driver = new ChromeDriver(chromeOptions);
// Turn off the JavaScript Firefox
ProfilesIni profile = new ProfilesIni();
FirefoxProfile firefoxProfile = profile.getProfile("ProfileName");
firefoxProfile.setPreference("javascript.enabled", false);
FirefoxOptions firefoxOptions = new FirefoxOptions();
firefoxOptions.setProfile(firefoxProfile);
driver = new FirefoxDriver(firefoxOptions);
// Set the default page load timeout
driver.manage().timeouts().pageLoadTimeout(10, TimeUnit.SECONDS);
// Start Firefox with plugins
FirefoxProfile profile = new FirefoxProfile();
firefoxProfile.addExtension(new
    File("C:\\extensionsLocation\\extension.xpi"));
FirefoxOptions firefoxOptions = new FirefoxOptions();
firefoxOptions.setProfile(firefoxProfile);
driver = new FirefoxDriver(firefoxOptions);
// Start Chrome with an unpacked extension
ChromeOptions chromeOptions = new ChromeOptions();
chromeOptions.addArguments("load-extension=/path/to/extension");
driver = new ChromeDriver(chromeOptions);
// Start Chrome with a packed extension
ChromeOptions chromeOptions = new ChromeOptions();
chromeOptions.addExtensions(new File("local/path/to/extension.crx"));
driver = new ChromeDriver(chromeOptions);
// Change the default files' save location
FirefoxProfile firefoxProfile = new FirefoxProfile();
String downloadFilepath = "c:\\temp";
firefoxProfile.setPreference("browser.download.folderList", 2);
firefoxProfile.setPreference("browser.download.dir", downloadFilepath);
firefoxProfile.setPreference("browser.download.manager.alertOnEXEOpen",
    false);
firefoxProfile.setPreference("browser.helperApps.neverAsk.saveToDisk",
    "application/msword, application/binary, application/ris, text/csv,
    image/png, application/pdf, text/html, text/plain, application/zip,
    application/x-zip, application/x-zip-compressed, application/download,
    application/octet-stream");
FirefoxOptions firefoxOptions = new FirefoxOptions();
firefoxOptions.setProfile(firefoxProfile);
driver = new FirefoxDriver(firefoxOptions);
```