



## 🔧 Setup & Installation

>\_ Add Selenium WebDriver dependency to your Maven project

```
org.seleniumhq.selenium
selenium-java
4.11.0
```

>\_ Add TestNG dependency to your Maven project

```
org.testng
testng
7.9.1
test
```

>\_ Install TestNG plugin for your IDE (Eclipse, IntelliJ)

Search for 'TestNG' in your IDE's plugin marketplace and install.

>\_ Create a simple Selenium test

```
@Test
public void testGoogleSearch() {
    WebDriver driver = new
    ChromeDriver();

    driver.get("https://www.google.com
");
    driver.quit();
}
```

## 🛡️ Actions & Waits

>\_ Click on an element

```
element.click();
```

>\_ Type text into an element

```
element.sendKeys("text to type");
```

>\_ Get text from an element

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

>\_ Explicit wait (for specific condition)

```
WebDriverWait wait = new
WebDriverWait(driver,
Duration.ofSeconds(10));
WebElement element =
wait.until(ExpectedConditions.elem
entToBeClickable(By.id("elementId"
)));
```

>\_ Implicit wait (for all elements)

```
driver.manage().timeouts().implici
tlyWait(Duration.ofSeconds(10));
```

## ⚙️ Page Objects & TestNG

>\_ Create a Page Object class

```
//Page Object Class for Google
Search Page
public class GoogleSearchPage {
    // WebElements declarations
}
```

>\_ Use Page Objects in Test Cases

```
GoogleSearchPage googleSearchPage
= new GoogleSearchPage(driver);
googleSearchPage.search("Selenium"
);
```

>\_ Run tests in parallel using TestNG

```
// TestNG configuration for parallel
execution
// Example: Run tests in parallel
// using TestNG
```

>\_ Use TestNG annotations (@BeforeTest, @AfterTest)

```
@BeforeTest
public void setUp() { ... }
@AfterTest
public void tearDown() { ... }
```

## 📍 Locators & WebDriver Methods

>\_ Find element by ID

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

>\_ Find element by Name

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

>\_ Find element by Class Name

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

>\_ Find element by Xpath

```
WebElement element =
driver.findElement(By.xpath("//ele
ment[@attribute='value']"));
```

>\_ Find element by CSS selector

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

## ✅ Assertions & Verification

>\_ Assert equality of strings

```
Assert.assertEquals(actualText,
expectedText);
```

>\_ Assert true condition

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

>\_ Verify element is displayed

```
Assert.assertTrue(element.isDispla
yed());
```

>\_ Verify element is enabled

```
Assert.assertTrue(element.isEnable
d());
```

>\_ Verify element is selected

```
Assert.assertTrue(element.isSelect
ed());
```

## 📸 Screenshots & Reports

>\_ Take a screenshot

```
TakesScreenshot scrShot =
((TakesScreenshot)driver);
File
SrcFile=scrShot.getScreenshotAs(Out
putType.FILE);
FileUtils.copyFile(SrcFile, new
File("path/screenshot.png"));
```

>\_ Generate TestNG HTML report

Run your tests using TestNG. An HTML report will be generated in the test-output directory.

>\_ Use ExtentReports for customized reporting

```
// Add ExtentReports dependency to
pom.xml and use ExtentReports API
for detailed reporting.
```

>\_ Add Logging using Log4j

```
//Add Log4j dependency to pom.xml
and use Log4j for logging
statements.
```



## Master Selenium WebDriver with Java!

Become a Selenium expert. Learn Selenium WebDriver from basics to advanced frameworks.

- Comprehensive Selenium WebDriver tutorial
- Master various automation frameworks
- Build robust and scalable automation projects
- Hands-on projects and real-world examples
- Detailed reporting and logging techniques

[Enroll now >>](#)

<https://ikalamtech.com/selenium-web-driver-with-java-basics-to-advanced-frameworks>

