

Instruction Manual of Selenium



Author Name And Detail

Safinul ibrat Sakib	19-40297-1
Tahmid hassan Ridoy	19-40307-1
Lynette Yang	Author name
Author name	Author name
Author name	Author name

13/07/2022

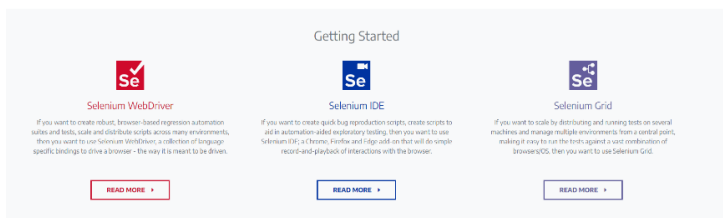
American International University

Faculty : ABHIJIT BHOWMIK

Table of Contents

Types Of Selenium.....	1
How to install Selenium IDE Flow Chart.....	2
How to run test in Selenium IDE.....	4
How to use Selenium Web driver Flow Chart	Error! Bookmark not defined.
How to run test using Selenium Web driver	Error! Bookmark not defined.

Types Of Selenium



Selenium WebDriver

If you want to create robust, browser-based regression automation suites and tests, scale and distribute scripts across many environments, then you want to use Selenium WebDriver, a collection of language specific bindings to drive a browser - the way it is meant to be driven.

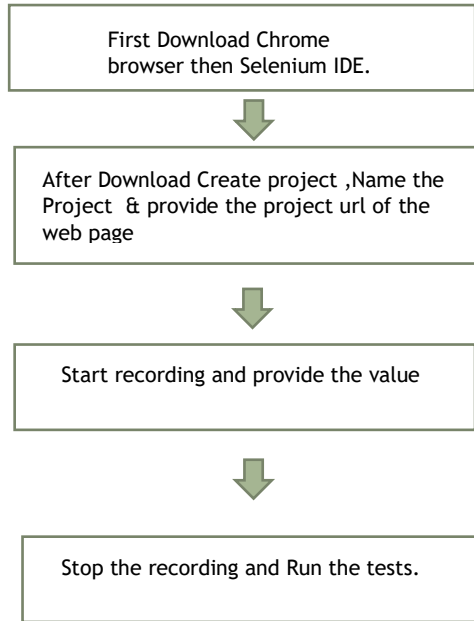
Selenium IDE

If you want to create quick bug reproduction scripts, create scripts to aid in automation-aided exploratory testing, then you want to use Selenium IDE; a Chrome, Firefox and Edge add-on that will do simple record-and-playback of interactions with the browser.

Selenium Grid

If you want to scale by distributing and running tests on several machines and manage multiple environments from a central point, making it easy to run the tests against a vast combination of browsers/OS, then you want to use Selenium Grid.

Selenium IDE Flow Chart



Downloading Process

The screenshot shows the Selenium website's 'Getting Started' section. It features three main options: Selenium WebDriver, Selenium IDE, and Selenium Grid. Selenium IDE is highlighted with a 'Click on it' arrow pointing to its 'READ MORE' button. Below this, the Selenium IDE page is shown, featuring a 'Click on it' arrow pointing to the 'LATEST ZIP' download button. The page also highlights features like 'Web Ready', 'Easy Debugging', and 'Cross-browser Execution'. At the bottom, the Selenium IDE extension is shown in the Chrome Web Store, with a 'Click on it' arrow pointing to the 'Add to Chrome' button. The final part of the screenshot shows a Windows activation dialog box with the text 'Activate Windows Go to Settings to activate Windows'.

Getting Started

Selenium WebDriver

If you want to create robust, browser based regression automation suites and tests, scale and distribute scripts across many environments, then you want to use Selenium WebDriver, a collection of language specific bindings to drive a browser – the way it is meant to be driven.

Selenium IDE

If you want to create quick bug reproduction scripts, create scripts to aid in automation aided exploratory testing, then you want to use Selenium IDE, a Chrome, Firefox and Edge add-on that will do simple record and playback of interactions with the browser.

Selenium Grid

If you want to scale by distributing and running tests on several machines and manage multiple environments from a central point, making it easy to run the tests against a vast combination of browsers/OS, then you want to use Selenium Grid.

Selenium IDE

Open source record and playback test automation for the web

Click on it

Web Ready

Simple, turnkey solution to quickly author, initiate and execute tests. Works out of the box for any web app.

Easy Debugging

Enjoy native test debugging with rich IDE features like setting breakpoints and pausing on exceptions.

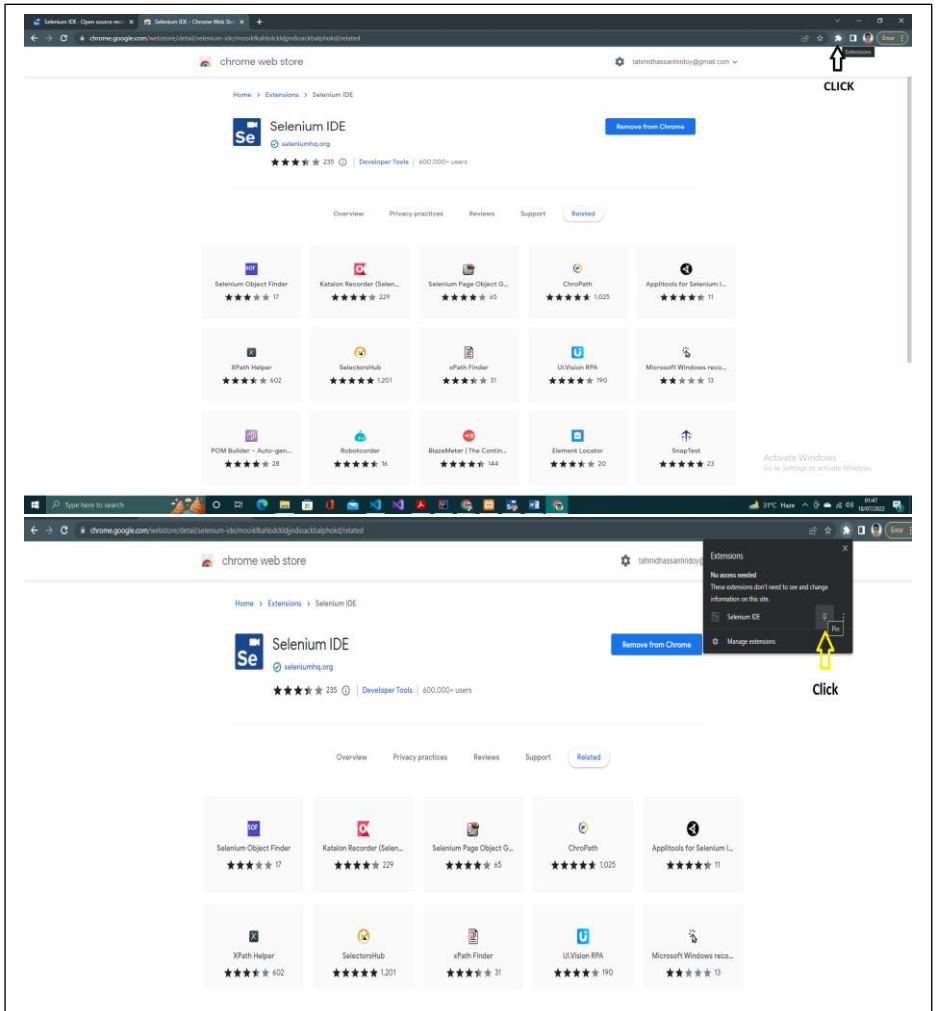
Cross-browser Execution

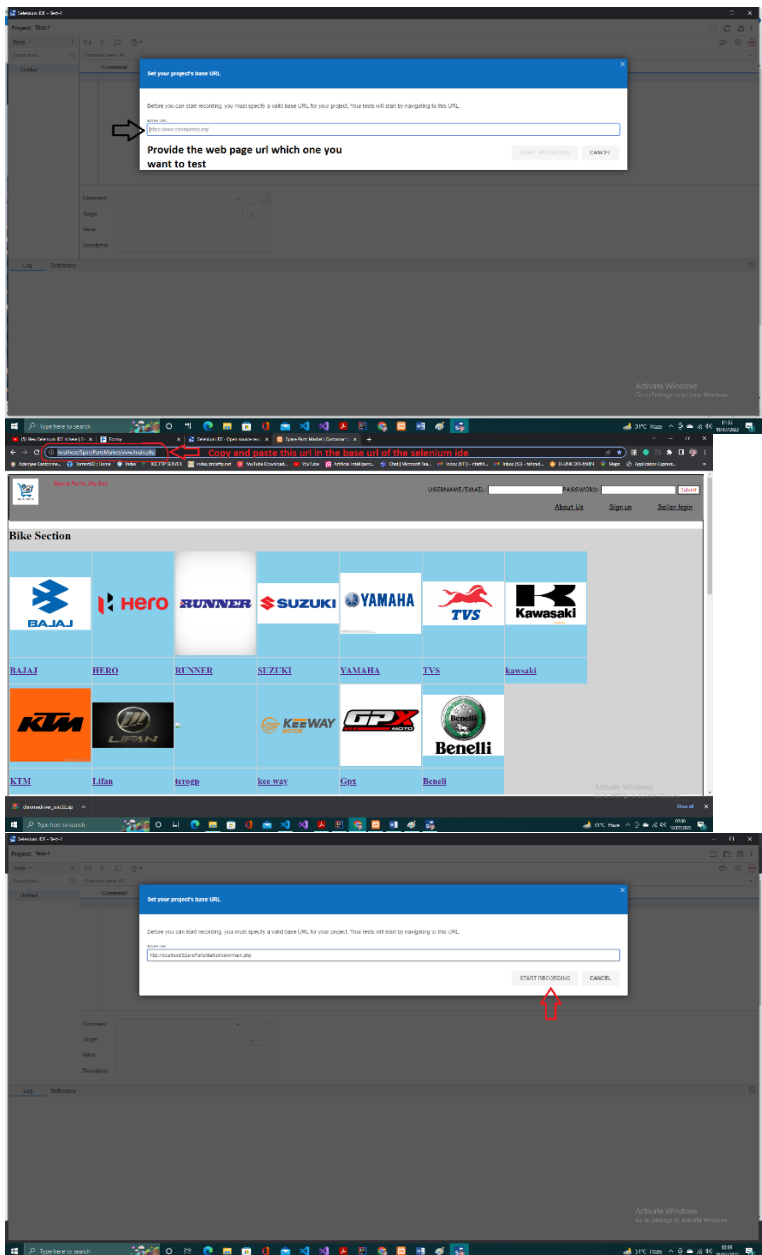
Run your tests on any browser/OS combination in parallel using the Command line runner for Selenium IDE.

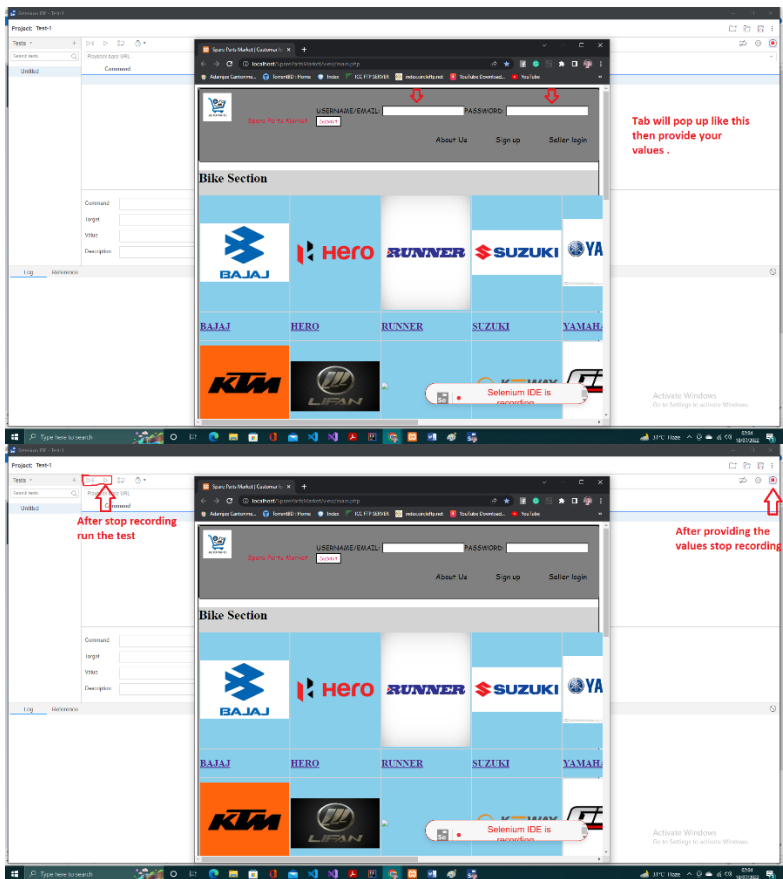
Click on it

Activate Windows
Go to Settings to activate Windows

How to run test in Selenium IDE

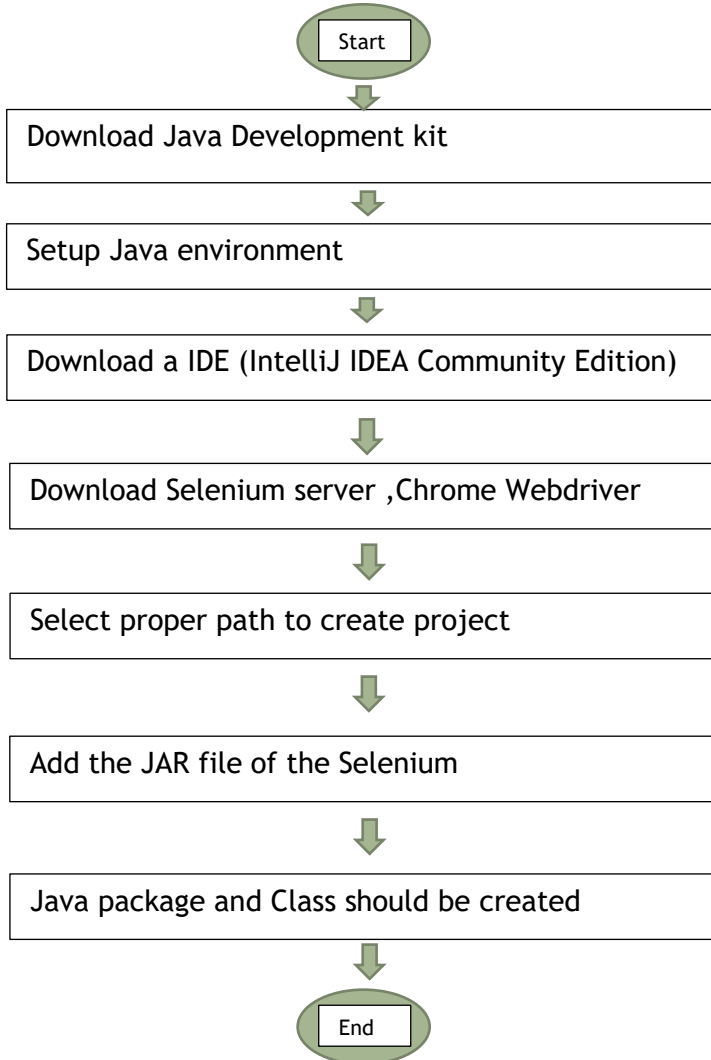




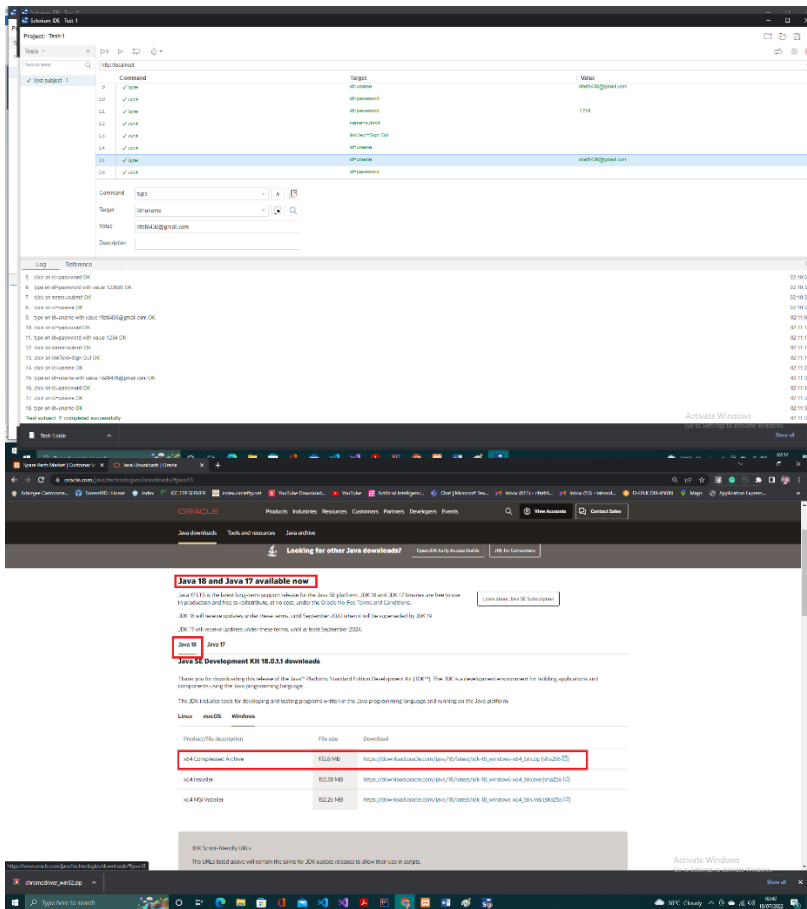




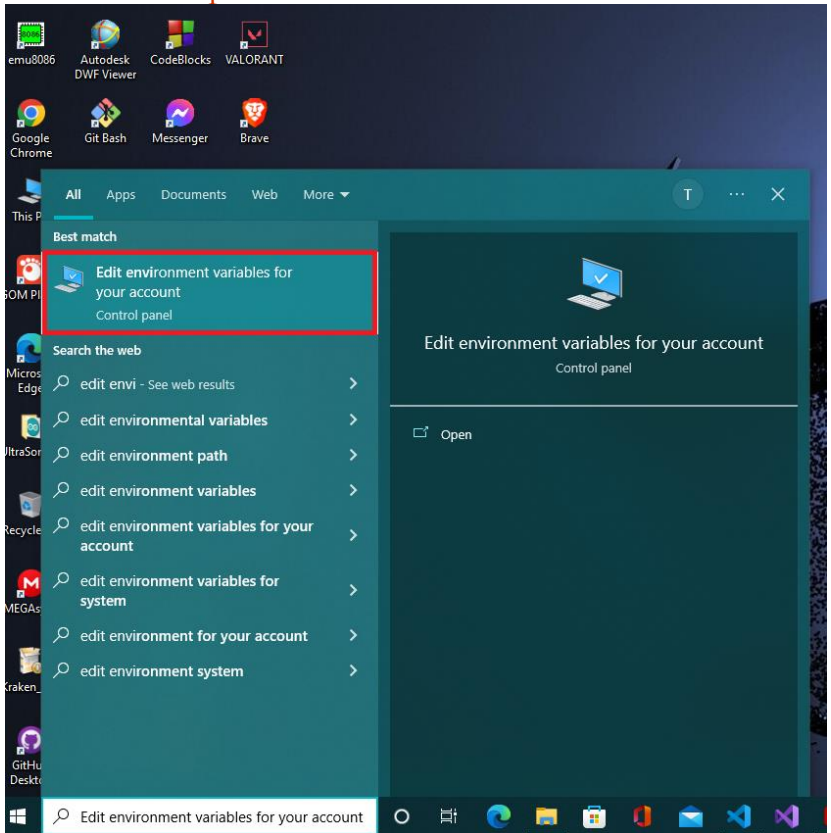
How to Use Selenium Web driver Flow chart

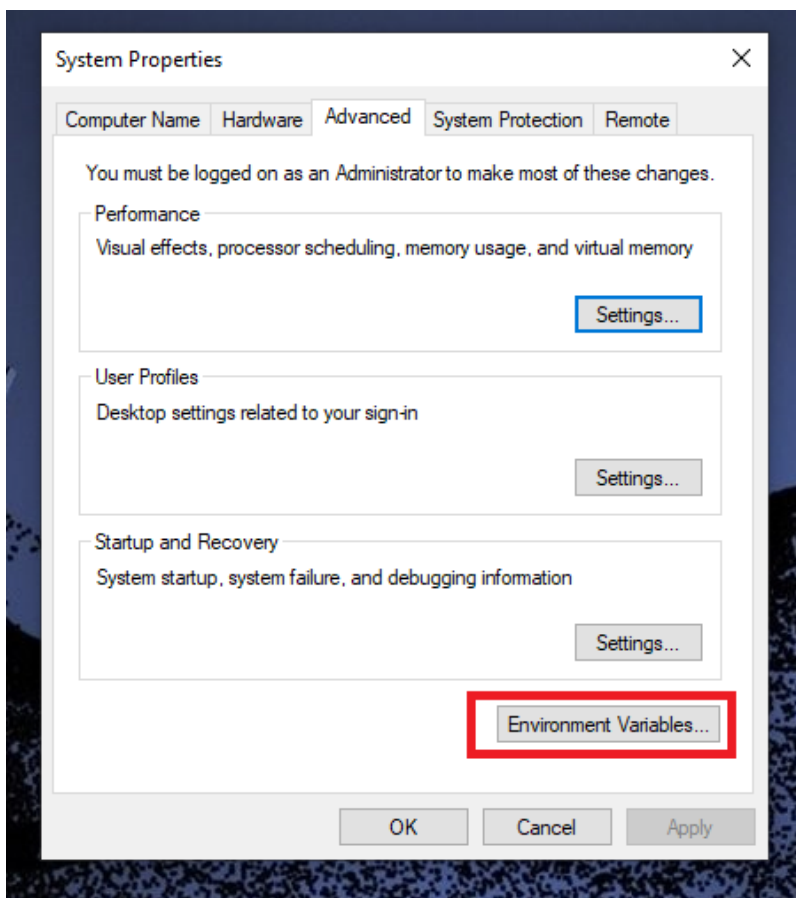


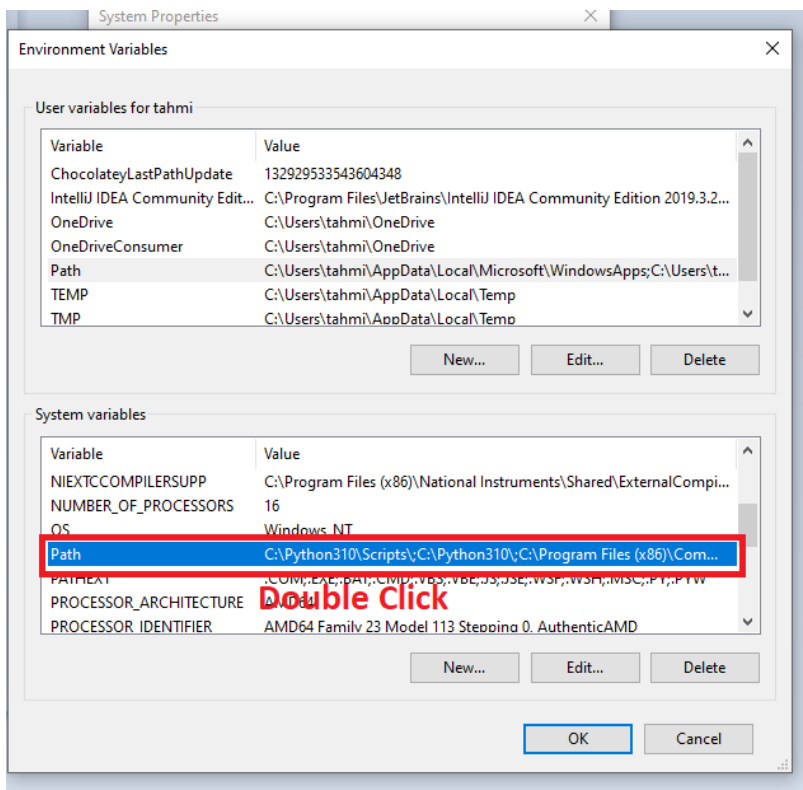
In the beginning JAVA SE Development kit has been downloaded.

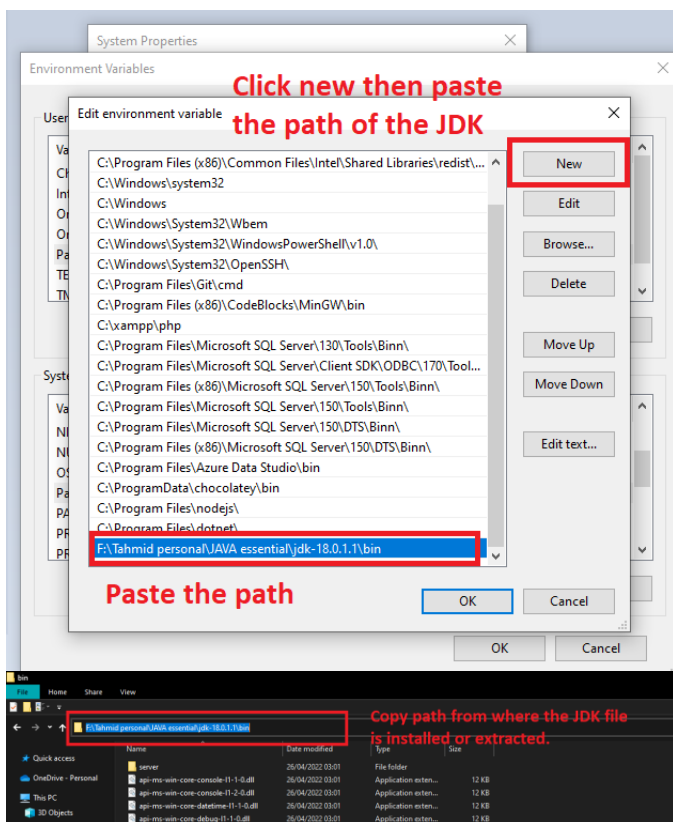


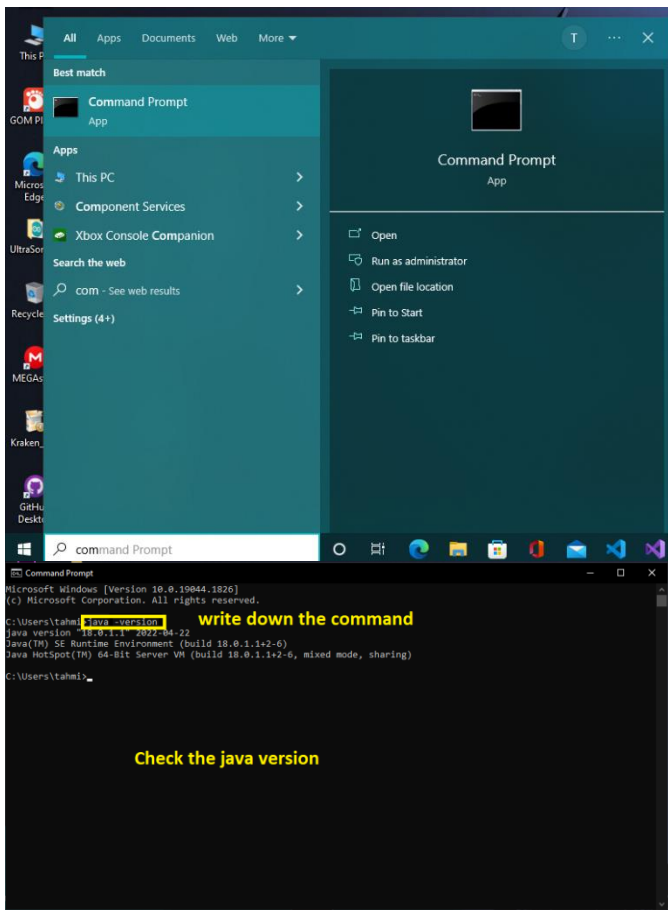
Secondly, Environment Variables for JAVA has been set up












Thirdly, for writing JAVA IntelliJ IDEA & Selenium Sever & Chrome Web-Driver has been Downloaded.



Version: 2022.1.8
Build: 21.1963.12
21 June 2022
[Release notes](#)

[System requirements](#)
[Installation instructions](#)
[Other versions](#)
[Third-party software](#)

Download IntelliJ IDEA

[Windows](#) [macOS](#) [Linux](#)

Ultimate
For web and enterprise development
[Download](#) [Free](#)
Free 30-day trial available

Community
For light and desktop development
[Download](#) [Free](#)
Free, built on open source

Download the community

	IntelliJ IDEA Ultimate	IntelliJ IDEA Community Edition
Java, Kotlin, Groovy, Scala	✓	✓
Python, C#, Swift	✓	✓
Qt, QtWebEngine, QtWebKit, QtWebKit2	✓	✓
Unlabeled	✓	✓

Downloads

Below is where you can find the latest releases of all the Selenium components. You can also find a list of previous releases, source code, and additional information for Maven users.

Download the Grid

Selenium Server (Grid)

The Selenium Server is needed in order to run Remote Selenium WebDriver (Grid).

Latest stable version [3.4.0](#)

To use the Selenium Server in a Grid configuration see the [documentation](#)

The Internet Explorer Driver Server

This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver. Please make sure that this is available on your [machine](#) in order for the IE Driver to work as expected.

Download version 4.0.0 for:

- [32-bit Windows IE \(recommended\)](#)
- [64-bit Windows IE](#)
- [ChromeOS](#)

ChromeDriver

Please make sure that your browser version and platform matches the version of the chromedriver, if those are mismatching then your tests will not work.

WebDriver is an open source tool for automated testing of webpages across many browsers. It provides capabilities for navigating to web pages, user inputs, JavaScript execution, and more. Chromedriver is a standalone server that implements the [W3C WebDriver standard](#). Chromedriver is available for Chrome on Android and Chrome on Desktop (Mac, Linux, Windows and ChromeOS).

You can view the current implementation status of the WebDriver standard [here](#).

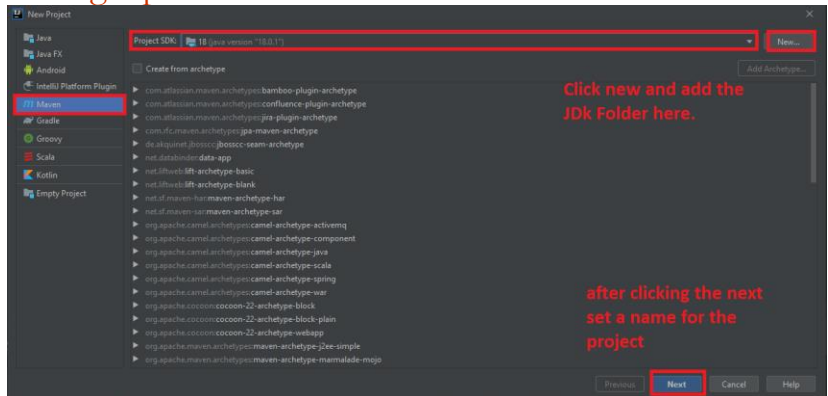
All versions available in Downloads

- Latest beta release ([Chromedriver 104.0.5112.20](#))
- Latest stable release ([Chromedriver 104.0.5090.25](#))

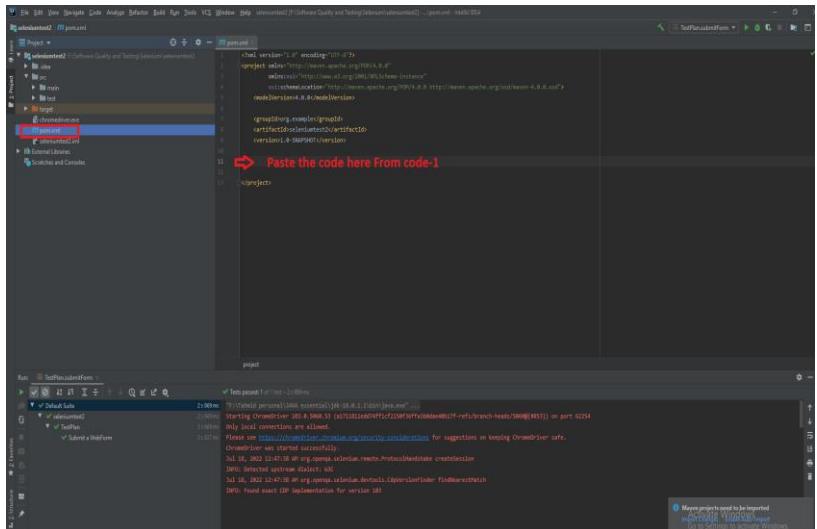
ChromeDriver Documentation

- [Getting started with Chromedriver on Testing](#) (Windows, Mac, Linux)
- [Chromedriver with WebDriver](#)
- [Chromedriver with ChromeOS](#)
- [Chromedriver: for capabilities of Chromedriver](#)
- [Mobile emulation](#)
- [Security considerations](#) with recommendations on keeping Chromedriver safe
- [Chrome Extension installation](#)

Setting Up the IDE



After that Setup code for pom.xml



Code-1 :

```
<properties>

  <maven.compiler.source>18</maven.compiler.source>

  <maven.compiler.target>18</maven.compiler.target>

</properties>

<dependencies>

  <dependency>

    <groupId>org.seleniumhq.selenium</groupId>

    <artifactId>selenium-server</artifactId>

    <version>3.141.59</version>

  </dependency>

  <dependency>

    <groupId>org.testng</groupId>

    <artifactId>testng</artifactId>

    <version>7.1.0</version>

    <scope>test</scope>

  </dependency>

  <dependency>

    <groupId>org.apache.maven.plugins</groupId>

    <artifactId>maven-surefire-plugin</artifactId>

    <version>3.0.0-M4</version>

  </dependency>

  <dependency>

    <groupId>org.testng</groupId>

    <artifactId>testng</artifactId>

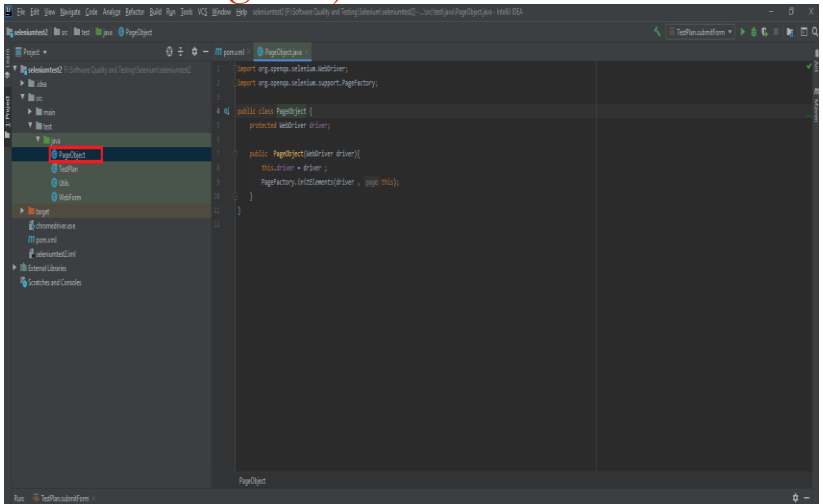
    <version>7.1.0</version>

    <scope>test</scope>

  </dependency>

</dependencies>
```

Code for PageObject :



Code-2:

```
import org.openqa.selenium.WebDriver;

import org.openqa.selenium.support.PageFactory;

public class PageObject {

    protected WebDriver driver;

    public PageObject(WebDriver driver){

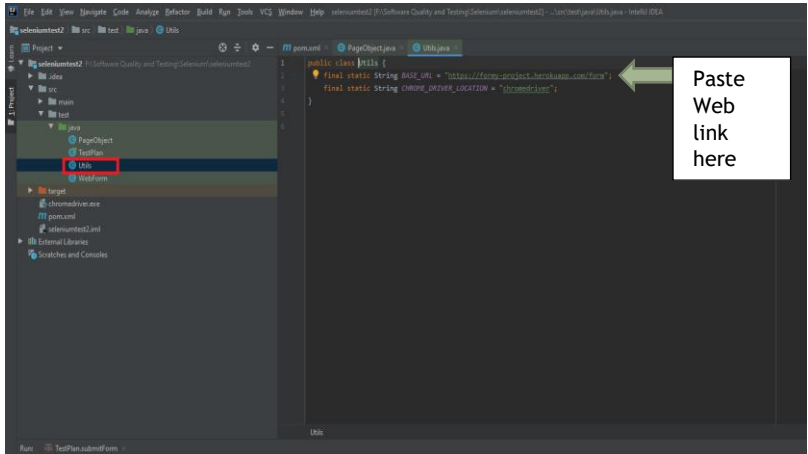
        this.driver = driver ;

        PageFactory.initElements(driver , this);

    }

}
```

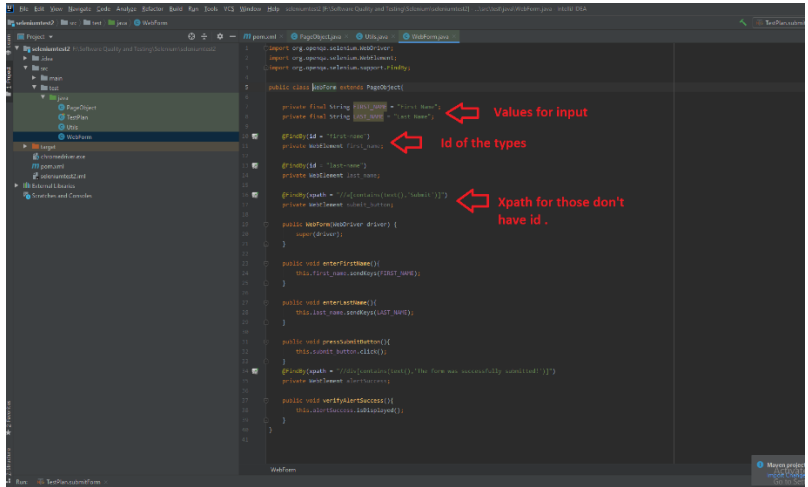
Codes For Utlis :



Code-3:

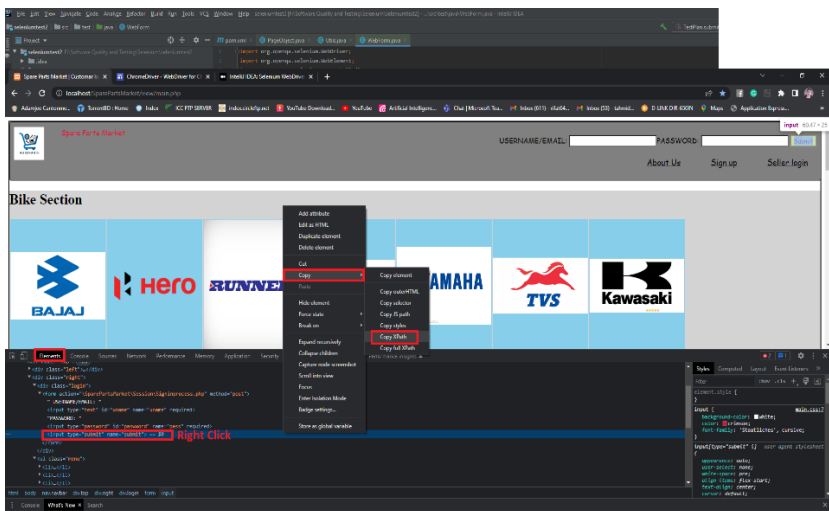
```
public class Utils {  
  
    final static String BASE_URL = "https://formy-project.herokuapp.com/form";  
  
    final static String CHROME_DRIVER_LOCATION =  
    "chromedriver";  
  
}
```


Codes of WebForm :



How to Find XPath :

- 1.Open the web page in the Browser Press F12 from key board.
- 2.Select Element , Search the line of the code that you want the XPath for that.
3. right click
- 4.copy the XPath.



Code-3:

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
```

```
public class WebForm extends PageObject{
```

```
    private final String FIRST_NAME = "First Name";
```

```
    private final String LAST_NAME = "Last Name";
```

```
    @FindBy(id = "first-name")
```

```

private WebElement first_name;

@FindBy(id = "last-name")
private WebElement last_name;

@FindBy(xpath =
"//a[contains(text(),'Submit')]")
private WebElement submit_button;

public WebForm(WebDriver driver) {
    super(driver);
}

public void enterFirstName(){
    this.first_name.sendKeys(FIRST_NAME);
}

public void enterLastName(){
    this.last_name.sendKeys(LAST_NAME);
}

public void pressSubmitButton(){

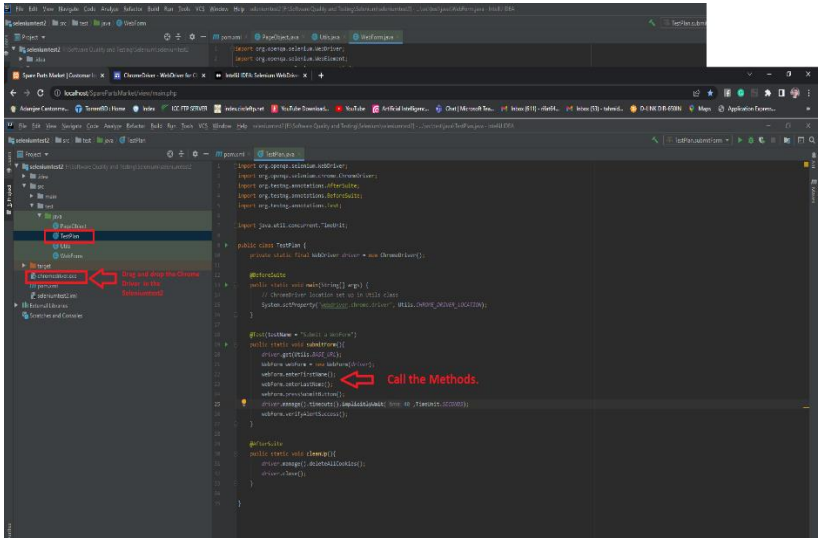
```

```
        this.submit_button.click();
    }

    @FindBy(xpath = "//div[contains(text(),'The
form was successfully submitted!')]")
    private WebElement alertSuccess;

    public void verifyAlertSuccess(){
        this.alertSuccess.isDisplayed();
    }
}
```

Codes For TestPlan :



Code-4:

```
import org.openqa.selenium.WebDriver;

import
org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterSuite;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.Test;

import java.util.concurrent.TimeUnit;

public class TestPlan {
```

```

    private static final WebDriver driver = new
ChromeDriver();

    @BeforeSuite

    public static void main(String[] args) {
        // ChromeDriver location set up in Utils class

System.setProperty("webdriver.chrome.driver",
Utils.CHROME_DRIVER_LOCATION);
    }

    @Test(testName = "Submit a WebForm")
    public static void submitForm(){
        driver.get(Utils.BASE_URL);
        WebForm webForm = new WebForm(driver);
        webForm.enterFirstName();
        webForm.enterLastName();
        webForm.pressSubmitButton();
        driver.manage().timeouts().implicitlyWait(40
, TimeUnit.SECONDS);
        webForm.verifyAlertSuccess();
    }

```

@AfterSuite

```
public static void cleanUp(){
```

```
    driver.manage().deleteAllCookies();
```

```
    driver.close();
```

```
}
```

```
}
```

Final Output :

