1. WebDriver Installation and Integration in Eclipse.

→Create new maven project

Pom.xml

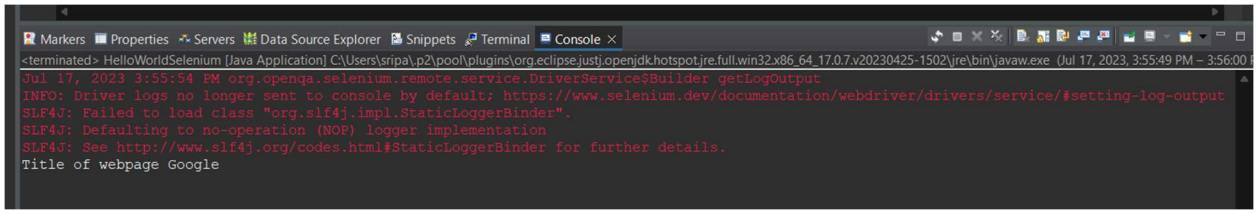
```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>hello-selenium</groupId>
  <artifactId>hello-selenium</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
        <!--
https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.10.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
chrome-driver -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-chrome-driver</artifactId>
    <version>4.10.0</version>
</dependency>
  </dependencies>
</project>
```

HelloWorldSelenium.java

```
package com.ecommerce.test;

import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class HelloWorldSelenium {

      public static void main(String[] args) throws InterruptedException {

            // Basic configuration
            WebDriver driver = new ChromeDriver();
```

```
            String baseUrl = "https://www.google.com";

            driver.get(baseUrl);

            System.out.println("Title of webpage "+driver.getTitle());

            ///
            Thread.sleep(1000);// 1 ms
            driver.close();
        }

}
```

Output



2. Locating Web Page Elements.

Pom.xml

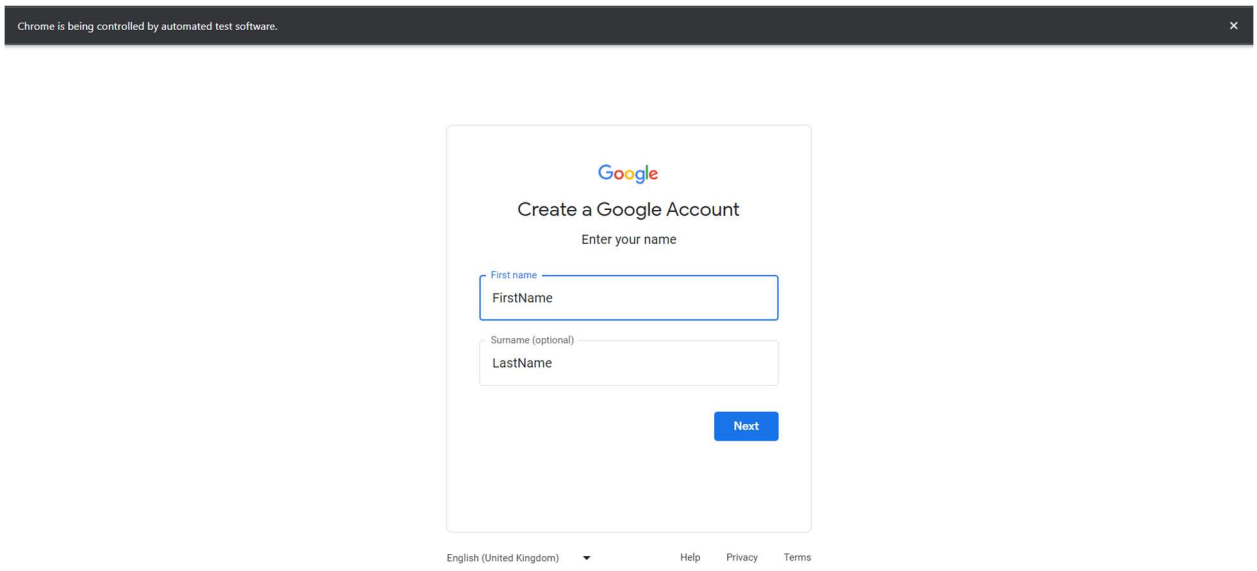
```
3.  <project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
4.    <modelVersion>4.0.0</modelVersion>
5.    <groupId>hello-selenium</groupId>
6.    <artifactId>hello-selenium</artifactId>
7.    <version>0.0.1-SNAPSHOT</version>
8.    <dependencies>
9.      <!--
    https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java
    -->
10. <dependency>
11.     <groupId>org.seleniumhq.selenium</groupId>
12.     <artifactId>selenium-java</artifactId>
13.     <version>4.10.0</version>
14. </dependency>
15. <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
    chrome-driver -->
16. <dependency>
17.     <groupId>org.seleniumhq.selenium</groupId>
18.     <artifactId>selenium-chrome-driver</artifactId>
19.     <version>4.10.0</version>
20. </dependency>
21.   </dependencies>
22. </project>
23.
```

LocateElementsDemo.java

```
package com.ecommerce.test;
import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;
//import org.openqa.selenium.firefox.FirefoxDriver;

public class LocateElementsDemo {

        public static void main(String[] args) throws InterruptedException {
                // Basic configuration
                                WebDriver driver = new ChromeDriver();

                                String baseUrl =
"https://accounts.google.com/signup/v2/createaccount?flowEntry=SignUp";

                                driver.get(baseUrl);
```

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

                        //System.out.println("firstname text feild details "+firstNameTF);

                        firstNameTF.sendKeys("FirstName");
                        WebElement lastNameTF = driver.findElement(By.name("lastName"));

                        lastNameTF.sendKeys("LastName");
                        //Locate the button using class
                        WebElement nextButton = driver.findElement(By.className("VfPpkd-
vQzf8d"));

                        nextButton.click();

                        Thread.sleep(5000);
                        driver.close();

        }

}
```

OUTPUT



3.Locating Elements through CSS and XPath.

Pom.xml

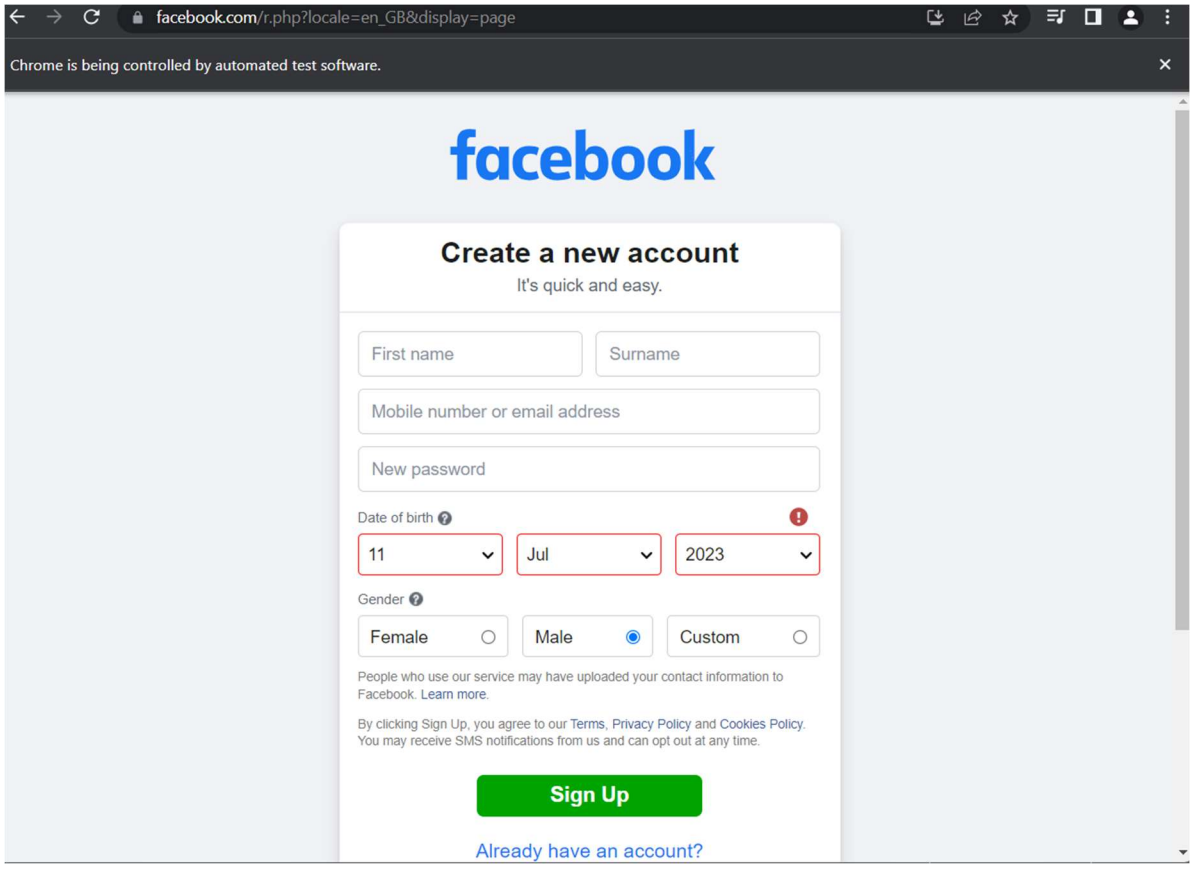
```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>hello-selenium</groupId>
  <artifactId>hello-selenium</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
        <!--
https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.10.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
chrome-driver -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-chrome-driver</artifactId>
    <version>4.10.0</version>
</dependency>
  </dependencies>
</project>
```

LocateElement.java

```
package com.ecommerce.test;

import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
//import org.openqa.selenium.firefox.FirefoxDriver;

public class LocateElementsDemo {

      public static void main(String[] args) throws InterruptedException {
            // Basic configuration
                        WebDriver driver = new ChromeDriver();

                        // Demo for locating TextFeild, Button, Select
                        // using id, name and className, xpath
                        facebookAccCreation(driver);
      }

      public static void facebookAccCreation(WebDriver driver) throws
InterruptedException {
```

```
            String baseUrl =
"https://www.facebook.com/r.php?locale=en_GB&display=page";

            driver.get(baseUrl);

            //xpath
            String fullXPathOfDayDropDown =

      "/html/body/div[1]/div[1]/div[1]/div[2]/div/div[2]/div/div/div[1]/form/
div[1]/div[5]/div[2]/span/span/select[1]";
            WebElement dayDropDown =
driver.findElement(By.xpath(fullXPathOfDayDropDown));

            Select select = new Select(dayDropDown);
            select.selectByVisibleText("11");

            //css selector
            WebElement gender= driver.findElement(By.cssSelector("span > span
> input[type='radio'][value='2']"));
            gender.click();
//          Thread.sleep(15000);
//    driver.close();

      }

}
```

OUTPUT



4. Handling Various Web Elements.

test.html

```
<html>
<head>
<title>Hello</title>
<script>
	function myDblClickFunction() {
		document.getElementById("demo-messages").textContent = "You
double clicked me!";
		document.getElementById("demo-messages").style.color = "red";
	}

	function mySglClickFunction() {
		document.getElementById("demo-messages").textContent = "You
single clicked me!";
		document.getElementById("demo-messages").style.color = "blue";
	}
</script>
</head>
<body bgcolor="#ffeeee">
	DEMO-MESSAGES:
	<span id="demo-messages">GETTING STARTED</span>

	<p id="xyz">Hello world!</p>


	Alerts The simplest of these is referred to as an alert, which shows a
	custom message, and a single button which dismisses the alert, labelled
	in most browsers as OK. It can also be dismissed in most browsers by
	pressing the close button, but this will always do the same thing as
	the OK button.
	<a href="javascript:alert('hello alerts!')">See an example alert</a>

	<br>
	<br>
	<div id="modal">
		<iframe id="myframe" name="myframe"
			src="https://seleniumhq.github.io" width=1200 height=600>
		</iframe>
	</div>

	<br>
	<br>

	<button id="dblButton" ondblclick="myDblClickFunction()"
		onclick="mySglClickFunction()">Double-click me</button>

	<h3>0 Some h3 text</h3>
	<form>
		<input type="text" id="admin"> <br> <input type="text"
			id="admin123"> <br> <input type="text" id="adminxyz">
		<br> <input type="text" id="adm"> <br>
```

```
	</form>
	<h3>1 Some h3 text</h3>
	<h3>2 Some h3 text</h3>
	<h3>3 Some h3 text</h3>

	<select multiple aria-label="Month" name="birthday_month" id="month"
		title="Month" class="_9407 _5dba _9hk6 _8esg">
		<option
			value="1">Jan</option>
		<option value="2">Feb</option>
		<option value="3">Mar</option>
		<option value="4">Apr</option>
		<option value="5">May</option>
		<option value="6">Jun</option>
		<option value="7" selected="1">Jul</option>
		<option value="8">Aug</option>
		<option value="9">Sep</option>
		<option value="10">Oct</option>
		<option value="11">Nov</option>
		<option value="12">Dec</option>
		</select>


</body>
</html>
```

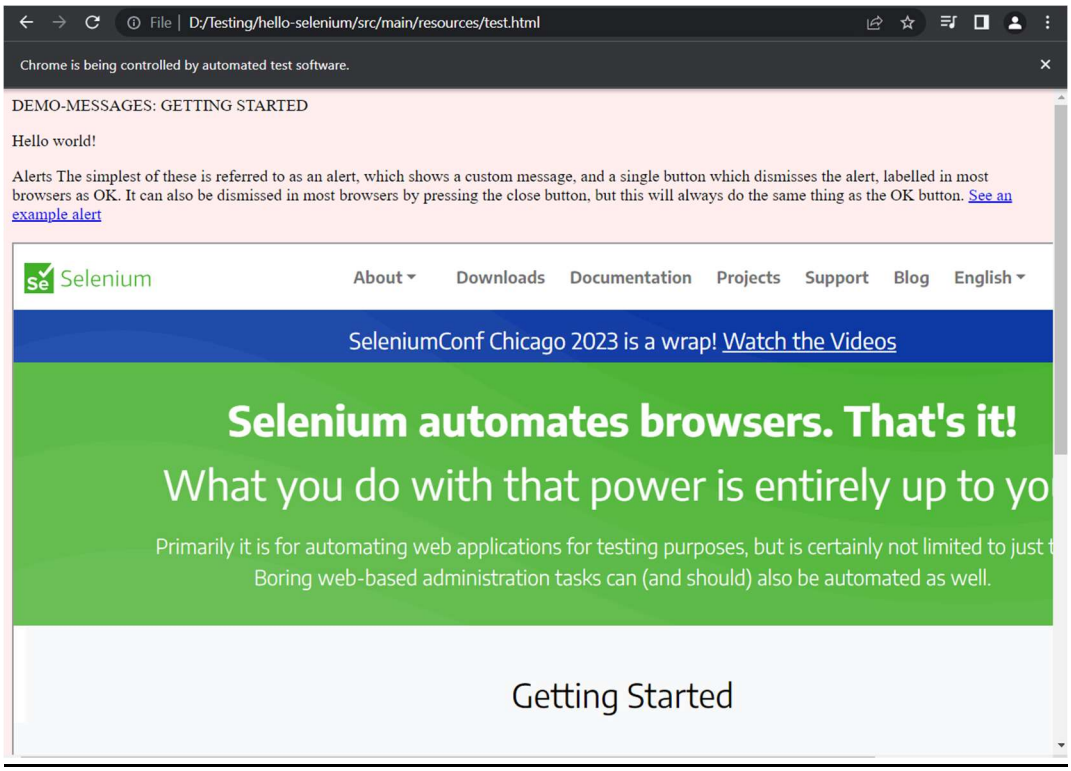
AdvancedLocatingElements

```
package com.ecommerce.test;

import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class AdvancedLocatingElements {

	public static void main(String[] args) throws InterruptedException {
		// Basic configuration
				WebDriver driver = new ChromeDriver();

				//demoAdvancedXPathCSSSelector(driver);

				demoTableDetails(driver);

	}

	static void demoAdvancedXPathCSSSelector(WebDriver driver) {
```

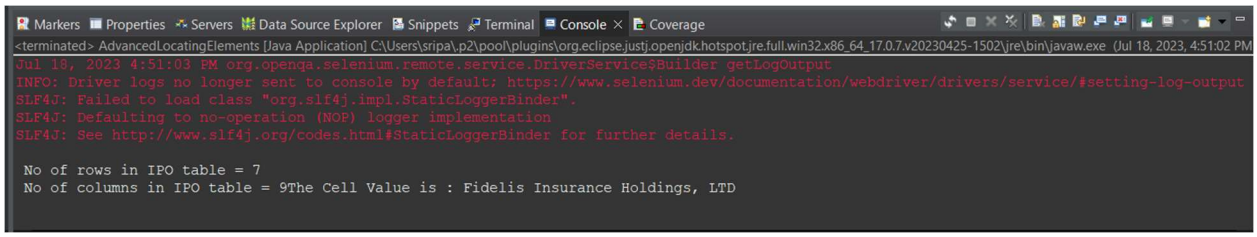
```
            String baseUrl = "File:///D:\\Testing\\hello-
selenium\\src\\main\\resources\\test.html";

            //String baseUrl =
"https://github.com/awtraining1/sl/blob/main/PHASE5/afternoon/hello-
selenium/src/main/resources/test.html";

            driver.get(baseUrl);


            List<WebElement> inputAdminElements =
driver.findElements(By.xpath("//input[contains(@id, 'admin')]"));

            System.out.println("inputAdminElements has " +
inputAdminElements.size());//3

            // Same as above but using CSS selectors
            List<WebElement> inputAdminElementsUsingCSSSelectors =
driver.findElements(By.cssSelector("input[id*='admin']"));
            System.out.println("inputAdminElementsUsingCSSSelectors has " +
inputAdminElementsUsingCSSSelectors.size());//3

            List<WebElement> inputAdminElementsUsingCSSSelectors2 =
driver.findElements(By.cssSelector("input[id$='xyz']"));
            System.out.println("input  whose ids end with xyz " +
inputAdminElementsUsingCSSSelectors2.size()); //2

            WebElement h4SecondChildInsideDiv =
driver.findElement(By.cssSelector("div[id='eCommerce'] h4:nth-child(2)"));
            System.out.println("h4SecondChildInsideDiv text is " +
h4SecondChildInsideDiv.getText()); // h4 second

            // Locate and Process multi select list
            WebElement selectMonthElement =
driver.findElement(By.id("month"));
            Select selectMonth =  new Select(selectMonthElement);

            System.out.printf("\n selectMonth.isMultiple() = %s",
selectMonth.isMultiple());


            selectMonth.selectByIndex(0);
            selectMonth.selectByIndex(3);
            selectMonth.selectByIndex(6);

            List<WebElement>  allMonthsSelected =
selectMonth.getAllSelectedOptions();
            for(WebElement monthOption: allMonthsSelected) {
                  System.out.printf("\n option selected = %s",
monthOption.getText());
            }


      }
```

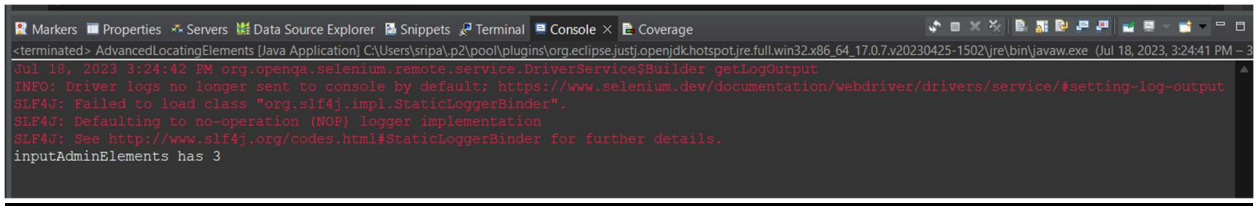
```
	// Locate and Process Table (no of rows, no of columns, particular cell
data, etc)
	static void demoTableDetails(WebDriver driver) {
		String baseUrl = "https://www.nyse.com/ipo-center/recent-ipo";
		driver.get(baseUrl);

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

		String tableXPath =
"/html/body/div[1]/div[4]/div[2]/div[3]/div[1]/div[4]/table";
		String tableXPathRow =
"/html/body/div[1]/div[4]/div[2]/div[3]/div[1]/div[4]/table/tbody/tr";

		List<WebElement> rowList =
driver.findElements(By.xpath(tableXPathRow));

		System.out.printf("\n No of rows in IPO table = %s",
rowList.size());

		// Column count
			String tableHeadingXPath =
"/html/body/div[1]/div[4]/div[2]/div[3]/div[1]/div[4]/table/thead/tr/th";
			List<WebElement> colList =
driver.findElements(By.xpath(tableHeadingXPath));

			System.out.printf("\n No of columns in IPO table =
%s", colList.size());

			// Finding cell value at 4th row and 3rd column
			WebElement cellAddress =
driver.findElement(By.xpath(tableXPath + "/tbody/tr[4]/td[3]"));
			String value = cellAddress.getText();
			System.out.println("The Cell Value is : " + value);

			//Change the cell contents using JS
			// JS Executor
			JavascriptExecutor js = (JavascriptExecutor) driver;
			String cssSelectScriptForCell3_3 = "var e
=document.querySelector(\"body > div:nth-child(1) > div:nth-child(4) >
div:nth-child(2) > div:nth-child(3) > div:nth-child(1) > div:nth-child(9) >
table > tbody > tr:nth-child(3) > td:nth-child(3) > strong\");
e.textContent='Bharat Gas'";
			js.executeScript(cssSelectScriptForCell3_3);

	}


}
```

OUTPUT



Date	Date	Issuer	Industry	Bookrunner(S)	day	[%]
6/28/2023	12/22/2021	**Savers Value Village, Inc**	Consumer Services	JPM, JEFFER, GS, UBS, RBAIRD, CIBCINC, GUGGEN, PIPERSAND	27.3	31.3
7/13/2023	6/22/2023	**Apogee Therapeutics, Inc.**	Healthcare	JEFFER, TDSEC, STFL, GUGGEN	24.9	22.4
6/28/2023	3/31/2023	**Bharat Gas**	Energy	GS, JPM, BRCLY, BAML, RAYMND, RBC, STFL, TFC , PRLLAWNBG	-1.9	1.8
6/28/2023	4/14/2023	**Fidelis Insurance Holdings, LTD**	Financials	JPM, BRCLY, JEFFER, BMOBMON, CITI, STFL, UBS	-7.9	-2.2
7/13/2023	6/23/2023	**Sagimet Biosciences Inc.**	Healthcare	GS, TDSEC, PIPERSAND	-0.3	-0.3


Markers Properties Servers Data Source Explorer Snippets Terminal Console Coverage
<terminated> AdvancedLocatingElements [Java Application] C:\Users\sripa\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (Jul 18, 2023, 4:51:02 PM
Jul 18, 2023 4:51:03 PM org.openqa.selenium.remote.service.DriverService$Builder getLogOutput
INFO: Driver logs no longer sent to console by default; https://www.selenium.dev/documentation/webdriver/drivers/service/#setting-log-output
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
No of rows in IPO table = 7
No of columns in IPO table = 9The Cell Value is : Fidelis Insurance Holdings, LTD


Markers Properties Servers Data Source Explorer Snippets Terminal Console Coverage
<terminated> AdvancedLocatingElements [Java Application] C:\Users\sripa\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (Jul 18, 2023, 3:24:41 PM – 3
Jul 18, 2023 3:24:42 PM org.openqa.selenium.remote.service.DriverService$Builder getLogOutput
INFO: Driver logs no longer sent to console by default; https://www.selenium.dev/documentation/webdriver/drivers/service/#setting-log-output
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
inputAdminElements has 3

5. Working with External Elements.

AdvancedLocatingElements.java

```
package com.ecommerce.test;

import java.time.Duration;
import java.util.List;
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;

public class AdvancedLocatingElements {

	public static void main(String[] args) throws InterruptedException {
		// Basic configuration
				WebDriver driver = new ChromeDriver();

				//demoAdvancedXPathCSSSelector(driver);

				demoExternalElements(driver);

	}

	static void demoExternalElements(WebDriver driver) throws
InterruptedException {

		String baseUrl = "File:///D:\\Testing\\hello-
selenium\\src\\main\\resources\\test.html";
		driver.get(baseUrl);

		//driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);
		Thread.sleep(10000);

		driver.findElement(By.linkText("See an example alert")).click();

		// alert will appear now, may be in 10 secs
		WebDriverWait wait = new WebDriverWait(driver,
Duration.ofSeconds(10));

		// Wait for the alert to be displayed
		wait.until(ExpectedConditions.alertIsPresent());

		// Store the alert in a variable
		Alert alert = driver.switchTo().alert();
System.out.printf("\n alert text is %s \n", alert.getText());
```

```
            Thread.sleep(10000);

            alert.accept();
      }

      static void demoAdvancedXPathCSSSelector(WebDriver driver) {

            String baseUrl = "File:///D:\\Testing\\hello-
selenium\\src\\main\\resources\\test.html";

            //String baseUrl =
"https://github.com/awtraining1/sl/blob/main/PHASE5/afternoon/hello-
selenium/src/main/resources/test.html";

            driver.get(baseUrl);


            List<WebElement> inputAdminElements =
driver.findElements(By.xpath("//input[contains(@id, 'admin')]"));

            System.out.println("inputAdminElements has " +
inputAdminElements.size());//3

            // Same as above but using CSS selectors
            List<WebElement> inputAdminElementsUsingCSSSelectors =
driver.findElements(By.cssSelector("input[id*='admin']"));
            System.out.println("inputAdminElementsUsingCSSSelectors has " +
inputAdminElementsUsingCSSSelectors.size());//3

            List<WebElement> inputAdminElementsUsingCSSSelectors2 =
driver.findElements(By.cssSelector("input[id$='xyz']"));
            System.out.println("input  whose ids end with xyz " +
inputAdminElementsUsingCSSSelectors2.size()); //2

            WebElement h4SecondChildInsideDiv =
driver.findElement(By.cssSelector("div[id='eCommerce'] h4:nth-child(2)"));
            System.out.println("h4SecondChildInsideDiv text is " +
h4SecondChildInsideDiv.getText()); // h4 second

            // Locate and Process multi select list
            WebElement selectMonthElement =
driver.findElement(By.id("month"));
            Select selectMonth =  new Select(selectMonthElement);

            System.out.printf("\n selectMonth.isMultiple() = %s",
selectMonth.isMultiple());


            selectMonth.selectByIndex(0);
            selectMonth.selectByIndex(3);
            selectMonth.selectByIndex(6);

            List<WebElement>  allMonthsSelected =
selectMonth.getAllSelectedOptions();
            for(WebElement monthOption: allMonthsSelected) {
```

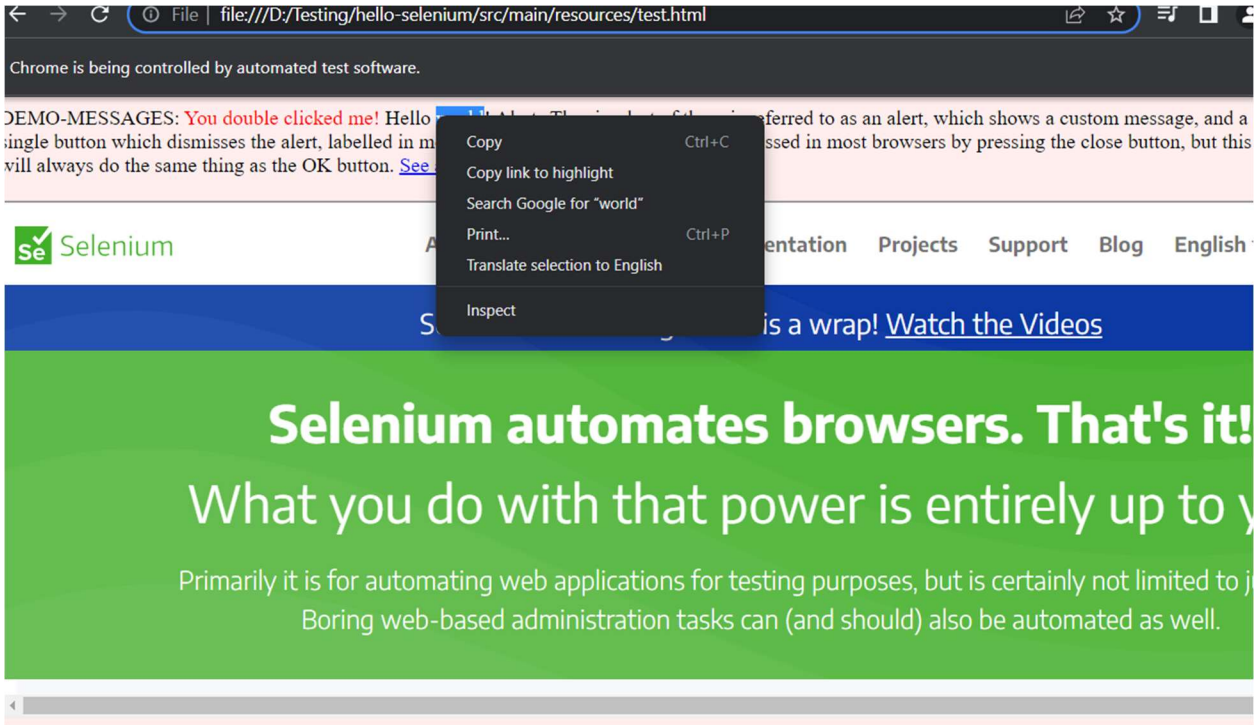
```
                  System.out.printf("\n option selected = %s",
monthOption.getText());
            }


      }

      // Locate and Process Table (no of rows, no of columns, particular cell
data, etc)
      static void demoTableDetails(WebDriver driver) {
            String baseUrl = "https://www.nyse.com/ipo-center/recent-ipo";
            driver.get(baseUrl);

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

            String tableXPath =
"/html/body/div[1]/div[4]/div[2]/div[3]/div[1]/div[4]/table";
            String tableXPathRow =
"/html/body/div[1]/div[4]/div[2]/div[3]/div[1]/div[4]/table/tbody/tr";

            List<WebElement> rowList =
driver.findElements(By.xpath(tableXPathRow));

            System.out.printf("\n No of rows in IPO table = %s",
rowList.size());

            // Column count
                        String tableHeadingXPath =
"/html/body/div[1]/div[4]/div[2]/div[3]/div[1]/div[4]/table/thead/tr/th";
                        List<WebElement> colList =
driver.findElements(By.xpath(tableHeadingXPath));

                        System.out.printf("\n No of columns in IPO table =
%s", colList.size());

                        // Finding cell value at 4th row and 3rd column
                        WebElement cellAddress =
driver.findElement(By.xpath(tableXPath + "/tbody/tr[4]/td[3]"));
                        String value = cellAddress.getText();
                        System.out.println("The Cell Value is : " + value);

                        //Change the cell contents using JS
                        // JS Executor
                        JavascriptExecutor js = (JavascriptExecutor) driver;
                        String cssSelectScriptForCell3_3 = "var e
=document.querySelector(\"body > div:nth-child(1) > div:nth-child(4) >
div:nth-child(2) > div:nth-child(3) > div:nth-child(1) > div:nth-child(9) >
table > tbody > tr:nth-child(3) > td:nth-child(3) > strong\");
e.textContent='Bharat Gas'";
                        js.executeScript(cssSelectScriptForCell3_3);

      }


}
```

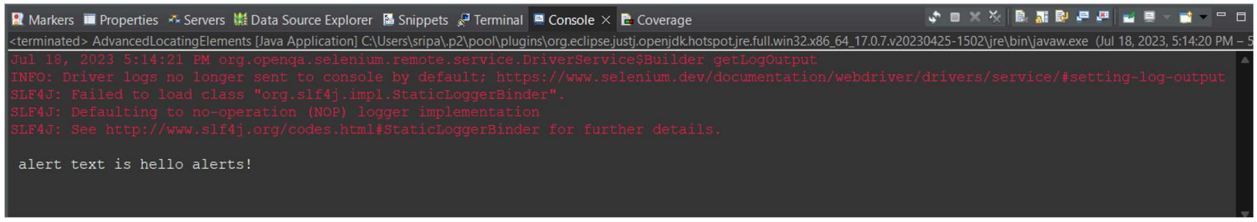
DemoMouseActions.java

```
package com.ecommerce.test;

import java.io.IOException;

import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Action;
import org.openqa.selenium.interactions.Actions;

public class DemoMouseActions {

	public static void main(String[] args) throws InterruptedException,
IOException {
		// Basic configuration
		WebDriver driver = new ChromeDriver();

		demoDoubleClick(driver);

	}

	static void demoDoubleClick(WebDriver driver) throws
InterruptedException {

		String baseUrl = "File:///D:\\Testing\\hello-
selenium\\src\\main\\resources\\test.html";
		driver.get(baseUrl);

		// driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);
		Thread.sleep(5000);

		WebElement button = driver.findElement(By.id("dblButton"));


		Actions actions = new Actions(driver);
		actions.doubleClick(button).perform();

		Thread.sleep(7000);

		//We can execute  many actions together.
		Thread.sleep(5000);
		WebElement helloPara = driver.findElement(By.id("xyz"));
		Actions multiActions = new Actions(driver);

//		Action action = multiActions
//		.doubleClick(helloPara)
//		.contextClick(helloPara)
//		.pause(5000)
//		.click(helloPara)
//		.build();
//
```

```
//            action.perform();

            //multi action
            Actions multiActions2 = new Actions(driver);
            Action action2 = multiActions2
            .moveToElement(button)
            .click(button)
            .pause(1000)
            .moveToElement(helloPara)
            .pause(2000)
            .doubleClick(helloPara)
            .release()
            .build();

            action2.perform();

            // scroll demo
            Thread.sleep(2000);
            Actions scrollPageActions = new Actions(driver);
            scrollPageActions
            .sendKeys(Keys.PAGE_DOWN)
            .sendKeys(Keys.PAGE_DOWN)
            .pause(5000)
            .sendKeys(Keys.PAGE_UP)
            .build()
            .perform();

      }


}
```

OUTPUT: -


File | file:///D:/Testing/hello-selenium/src/main/resources/test.html
Chrome is being controlled by automated test software.
DEMO-MESSAGES: You double clicked me! Hello
ferred to as an alert, which shows a custom message, and a
single button which dismisses the alert, labelled in m
ssed in most browsers by pressing the close button, but this
will always do the same thing as the OK button. See
Copy Ctrl+C
Copy link to highlight
Search Google for "world"
Print... Ctrl+P
Translate selection to English
Inspect
Selenium
entation
Projects
Support
Blog
English
is a wrap! Watch the Videos
Selenium automates browsers. That's it!
What you do with that power is entirely up to y
Primarily it is for automating web applications for testing purposes, but is certainly not limited to j
Boring web-based administration tasks can (and should) also be automated as well.


Markers
Properties
Servers
Data Source Explorer
Snippets
Terminal
Console
Coverage
<terminated> AdvancedLocatingElements [Java Application] C:\Users\sripa\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (Jul 18, 2023, 5:14:20 PM – 5
Jul 18, 2023 5:14:21 PM org.openqa.selenium.remote.service.DriverService$Builder getLogOutput
INFO: Driver logs no longer sent to console by default; https://www.selenium.dev/documentation/webdriver/drivers/service/#setting-log-output
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
alert text is hello alerts!

6. Screenshots

Test.html

```
<html>
<head>
<title>Hello</title>
<script>
	function myDblClickFunction() {
		document.getElementById("demo-messages").textContent = "You
double clicked me!";
		document.getElementById("demo-messages").style.color = "red";
	}

	function mySglClickFunction() {
		document.getElementById("demo-messages").textContent = "You
single clicked me!";
		document.getElementById("demo-messages").style.color = "blue";
	}
</script>
</head>
<body bgcolor="#ffeeee">
	DEMO-MESSAGES:
	<span id="demo-messages">GETTING STARTED</span>

	<span id="xyz">Hello world!</span>


	Alerts The simplest of these is referred to as an alert, which shows a
	custom message, and a single button which dismisses the alert, labelled
	in most browsers as OK. It can also be dismissed in most browsers by
	pressing the close button, but this will always do the same thing as
	the OK button.
	<a href="javascript:alert('hello alerts!')">See an example alert</a>

	<br>
	<br>
	<div id="modal">
		<iframe id="myframe" name="myframe"
			src="https://seleniumhq.github.io" width=1200 height=400>
		</iframe>
	</div>

	<br>
	<br>

	<button id="dblButton" ondblclick="myDblClickFunction()"
		onclick="mySglClickFunction()">Double-click me or Single click
me</button>

	<h3>0 Some h3 text</h3>
	<form>
		<input type="text" id="admin"> <br>
		 <input type="text" id="admin123"> <br>
		 <input type="text" id="adminxyz">
```

```
            <br> <input type="text" id="adm"> <br>

      </form>
      <h3>1 Some h3 text</h3>
      <h3>2 Some h3 text</h3>
      <h3>3 Some h3 text</h3>

      <div id='eCommerce'>
            <h4>h4 first</h4>
            <h4>h4 second</h4>
      </div>

      <select multiple aria-label="Month" name="birthday_month" id="month"
            title="Month" class="_9407 _5dba _9hk6 _8esg" size="12">
            <option
                  value="1">Jan</option>
            <option value="2">Feb</option>
            <option value="3">Mar</option>
            <option value="4">Apr</option>
            <option value="5">May</option>
            <option value="6">Jun</option>
            <option value="7" selected="1">Jul</option>
            <option value="8">Aug</option>
            <option value="9">Sep</option>
            <option value="10">Oct</option>
            <option value="11">Nov</option>
            <option value="12">Dec</option>
            </select>

      <input type="text" id="WWWWxyz">

</body>
</html>
```

Pom.xml

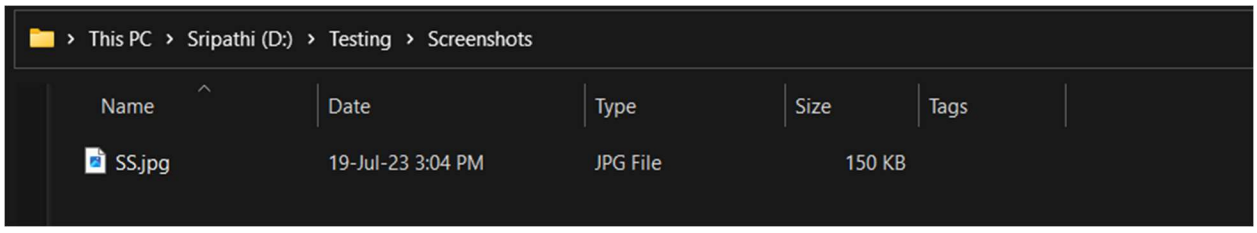
```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>hello-selenium</groupId>
  <artifactId>hello-selenium</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
        <!--
https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.10.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
chrome-driver -->
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-chrome-driver</artifactId>
```

```
    <version>4.10.0</version>
</dependency>
  </dependencies>
</project>
```

ScreenshotDemo.java

```
package com.ecommerce.test;

import java.io.File;
import java.io.IOException;

import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import com.google.common.io.Files;

public class ScreenshotDemo {

	public static void main(String[] args) throws Exception {
		// Basic configuration
		WebDriver driver = new ChromeDriver();

		demoTakingScreenshot(driver);
	}

	static void demoTakingScreenshot(WebDriver driver) throws IOException,
InterruptedException {

		String baseUrl = "File:///D:\\Testing\\hello-
selenium\\src\\main\\resources\\test.html";
		driver.get(baseUrl);

		Thread.sleep(2000);

		TakesScreenshot scrShot = ((TakesScreenshot) driver);

		File imgTmpFile = scrShot.getScreenshotAs(OutputType.FILE);

		File destinationFile = new File("D:/Testing/Screenshots/SS.jpg");

		Files.copy(imgTmpFile, destinationFile);
	}

}
```

OUTPUT



7. Handling File Uploads.

- Needed to download AutoIt
- Choose file upload button in https://demoqa.com/automation-practice-form
- Then open AutoIt v3 in options choose freeze
- Then add below code to sciTE file

```
Local $hWnd=WinWait("[CLASS:#32770]","",10)

ControlFocus($hWnd,"","Edit1")

Sleep(2000)

ControlSetText($hWnd, "", "Edit1", "D:\Testing\Screenshots\SS.jpg")

Sleep(2000)
ControlFocus($hWnd,"","Button1")

ControlClick($hWnd, "","Button1");
```

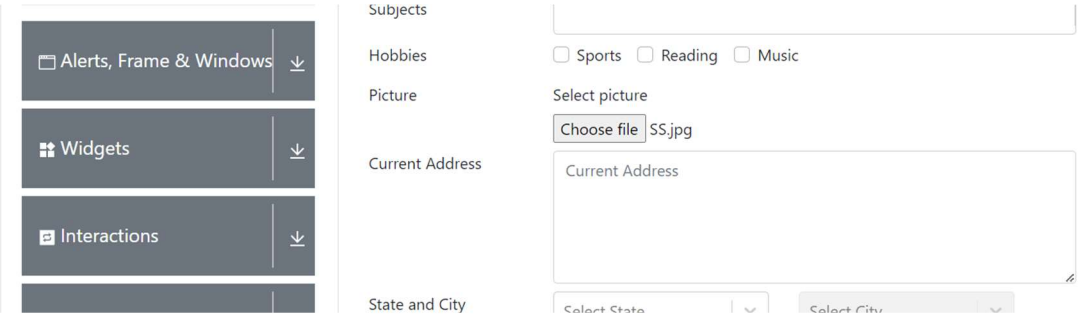
- And then convert this into exe file using compile script to exe

FileUploadUsingAutoITDemo.java

```
package com.ecommerce.test;

import java.io.IOException;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class FileUploadUsingAutoITDemo {

	public static void main(String[] args) throws Exception {
		// Basic configuration
		WebDriver driver = new ChromeDriver();

		demoFileUpload(driver);

	}

	static void demoFileUpload(WebDriver driver) throws
InterruptedException, IOException {

		// String baseUrl =
```

```
            // "File:///F:\\Users\\HomeWk\\git\\sl\\PHASE5\\afternoon\\hello-
selenium\\src\\main\\resources\\test.html";

            String baseUrl = "https://demoqa.com/automation-practice-form";
            driver.get(baseUrl);

            Thread.sleep(2000);

            WebElement uploadFileElememt =
driver.findElement(By.id("uploadPicture"));
            //uploadFileElememt.click();

            System.out.println("uploadFileElememt details " +
uploadFileElememt);

            Thread.sleep(10000);

            Runtime.getRuntime().exec("D:/Testing/Phase5Auto.exe");

      }

}
```

OUTPUT



8. Perform All Test Annotations.

FBRegistration.java

```
package com.ecommerce.test;

import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterSuite;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

public class FBRegistration {
	public WebDriver driver;
	public String baseUrl = "File:///D:\\Testing\\hello-
selenium\\src\\main\\resources\\test.html";

	@Test()
	public void searchGoogle() {
		System.out.printf("Inside %s and thread-id is %s
\n","searchGoogle", Thread.currentThread().getId()  );

		driver.get("http://www.google.com");

		System.out.println("Title of google page is " +
driver.getTitle());
	}

	@Test()
	public void fbAccount() {
		System.out.printf("Inside %s and thread-id is %s \n","fbAccount",
Thread.currentThread().getId()  );

		String baseUrl =
"https://www.facebook.com/r.php?locale=en_GB&display=page";

		driver.get(baseUrl);

		String cssDay = "#day";
		WebElement cssDaySelect =
driver.findElement(By.cssSelector(cssDay));

		Select daySelect = new Select(cssDaySelect);
```

```
            daySelect.selectByVisibleText("11");

            WebElement cssGenderRadio =
driver.findElement(By.cssSelector("span > span >
input[type='radio'][value='2']"));
            cssGenderRadio.click();

            System.out.println("Gender is enabled = " +
cssGenderRadio.isSelected());
      }

      @Test()
      public void googleAccount() {
            System.out.printf("Inside %s and thread-id is %s
\n","googleAccount", Thread.currentThread().getId()  );

            String baseUrl =
"https://accounts.google.com/signup/v2?biz=true&flowEntry=SignUp";
            driver.get(baseUrl);

            // Let's locate the first name text field by its id.
            WebElement firstNameTF = driver.findElement(By.id("firstName"));
            firstNameTF.sendKeys("Myname");

            // Let's locate the surname name text field by its name.
            WebElement surNameTF = driver.findElement(By.name("lastName"));
            surNameTF.sendKeys("latName");

            // The Next button using its class
            WebElement nextButton = driver.findElement(By.className("VfPpkd-
vQzf8d"));
            System.out.println("Text on the button is " +
nextButton.getText());
      }


      @Test()
      public void f1() {
            System.out.printf("Inside %s and thread-id is %s \n","f1",
Thread.currentThread().getId()  );
      }
      @Test()
      public void f2() {
            System.out.printf("Inside %s and thread-id is %s \n","f2",
Thread.currentThread().getId()  );
      }

      @Test()
      public void testLocalPage() {
            System.out.printf("Inside %s and thread-id is %s
\n","testLocalPage", Thread.currentThread().getId()  );

            driver.get(baseUrl);

            WebElement selectMonthElement =
driver.findElement(By.id("month"));
            Select selectMonth = new Select(selectMonthElement);
```

```
            System.out.println("selectMonth is muti select " +
selectMonth.isMultiple());

            selectMonth.selectByIndex(0);
            selectMonth.selectByIndex(6);

            List<WebElement> optionsList =
selectMonth.getAllSelectedOptions();

            for (WebElement option : optionsList)
                  System.out.println("selectMonth selected options are " +
option.getText());
      }

      @BeforeMethod
      public void beforeMethod() {
            System.out.printf("Inside %s and thread-id is %s
\n","beforeMethod", Thread.currentThread().getId()  );
      }

      @AfterMethod
      public void afterMethod() {
            System.out.printf("Inside %s and thread-id is %s
\n","afterMethod", Thread.currentThread().getId()  );
      }

      @DataProvider
      public Object[][] dp() {
            return new Object[][] { new Object[] { 1, "a" }, new Object[] {
2, "b" }, };
      }

      @BeforeClass
      public void beforeClass() {
            System.out.printf("Inside %s and thread-id is %s
\n","beforeClass", Thread.currentThread().getId()  );
      }

      @AfterClass
      public void afterClass() {
            System.out.printf("Inside %s and thread-id is %s \n
","afterClass", Thread.currentThread().getId()  );
      }

      @BeforeTest
      public void beforeTest() {
            System.out.printf("Inside %s and thread-id is %s
\n","beforeTest", Thread.currentThread().getId()  );
      }

      @AfterTest
      public void afterTest() {
            System.out.printf("Inside %s and thread-id is %s \n","afterTest",
Thread.currentThread().getId()  );
      }
```

```
	@BeforeSuite
	public void beforeSuite() {
		System.out.printf("Inside %s and thread-id is %s
\n","beforeSuite", Thread.currentThread().getId()   );

	//System.setProperty("webdriver.chrome.driver","C:\\Users\\HomeWk\\Down
loads\\chromedriver_win32\\chromedriver.exe");

		driver = new ChromeDriver();
	}

	@AfterSuite
	public void afterSuite() {
		System.out.printf("Inside %s and thread-id is %s
\n","afterSuite", Thread.currentThread().getId()   );
		driver.quit();
	}

}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>hello-selenium-testng</groupId>
  <artifactId>hello-selenium-testng</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
		<!--

	https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
java -->
		<dependency>
			<groupId>org.seleniumhq.selenium</groupId>
			<artifactId>selenium-java</artifactId>
			<version>4.10.0</version>
		</dependency>

		<!--

	https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
chrome-driver -->
		<dependency>
			<groupId>org.seleniumhq.selenium</groupId>
			<artifactId>selenium-chrome-driver</artifactId>
			<version>4.10.0</version>
		</dependency>

		<!--
```

```
      https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
firefox-driver -->
            <dependency>
                  <groupId>org.seleniumhq.selenium</groupId>
                  <artifactId>selenium-firefox-driver</artifactId>
                  <version>4.10.0</version>
            </dependency>
            <dependency>
                  <groupId>org.testng</groupId>
                  <artifactId>testng</artifactId>
                  <version>7.8.0</version>
                  <scope>compile</scope>
            </dependency>

      </dependencies>
      <build>
            <plugins>
                  <plugin>
                        <groupId>org.apache.maven.plugins</groupId>
                        <artifactId>maven-compiler-plugin</artifactId>
                        <configuration>
                              <source>1.8</source>
                              <target>1.8</target>
                        </configuration>
                  </plugin>
                  <plugin>
                        <groupId>org.apache.maven.plugins</groupId>
                        <artifactId>maven-surefire-plugin</artifactId>
                        <configuration>
                              <suiteXmlFiles>
                                    <suiteXmlFile>testng.xml</suiteXmlFile>
                              </suiteXmlFiles>
                        </configuration>
                  </plugin>
            </plugins>
      </build>
</project>
```

Testng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test FaceBook">
    <classes>
      <class name="com.ecommerce.test.FBRegistration"/>
    </classes>
  </test> <!-- Test -->


</suite> <!-- Suite -->
```

OUTPUT

```
[RemoteTestNG] detected TestNG version 7.8.0
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Inside beforeSuite and thread-id is 1
Jul 20, 2023 2:27:58 PM org.openqa.selenium.remote.service.DriverService$Builder getLogOutput
INFO: Driver logs no longer sent to console by default; https://www.selenium.dev/documentation/webdri
Inside beforeTest and thread-id is 1
Inside beforeClass and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside f1 and thread-id is 1
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside f2 and thread-id is 1
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside fbAccount and thread-id is 1
Gender is enabled = true
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside googleAccount and thread-id is 1
Text on the button is Next
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside searchGoogle and thread-id is 1
Title of google page is Google
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside testLocalPage and thread-id is 1
selectMonth is muti select true
selectMonth selected options are Jan
selectMonth selected options are Jul
Inside afterMethod and thread-id is 1
Inside afterClass and thread-id is 1
 Inside afterTest and thread-id is 1
PASSED: com.ecommerce.test.FBRegistration.fbAccount
PASSED: com.ecommerce.test.FBRegistration.f1
PASSED: com.ecommerce.test.FBRegistration.googleAccount
PASSED: com.ecommerce.test.FBRegistration.f2
PASSED: com.ecommerce.test.FBRegistration.searchGoogle
PASSED: com.ecommerce.test.FBRegistration.testLocalPage
```

9. Group Test Cases and Parallel Test Execution.

FBResgistration.java

```
package com.ecommerce.test;

import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterSuite;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

public class FBRegistration {
	public WebDriver driver;
	public String baseUrl = "File:///D:\\Testing\\hello-
selenium\\src\\main\\resources\\test.html";

	@Test(groups= {"Google"})
	public void searchGoogle() {
		System.out.printf("Inside %s and thread-id is %s
\n","searchGoogle", Thread.currentThread().getId()  );

		driver.get("http://www.google.com");

		System.out.println("Title of google page is " +
driver.getTitle());
	}

	@Test(groups= {"Account Creation"})
	public void fbAccount() {
		System.out.printf("Inside %s and thread-id is %s \n","fbAccount",
Thread.currentThread().getId()  );

		String baseUrl =
"https://www.facebook.com/r.php?locale=en_GB&display=page";

		driver.get(baseUrl);

		String cssDay = "#day";
		WebElement cssDaySelect =
driver.findElement(By.cssSelector(cssDay));

		Select daySelect = new Select(cssDaySelect);
```

```
            daySelect.selectByVisibleText("11");

            WebElement cssGenderRadio =
driver.findElement(By.cssSelector("span > span >
input[type='radio'][value='2']"));
            cssGenderRadio.click();

            System.out.println("Gender is enabled = " +
cssGenderRadio.isSelected());
      }

      @Test(groups= {"Account Creation", "Google"})
      public void googleAccount() {
            System.out.printf("Inside %s and thread-id is %s
\n","googleAccount", Thread.currentThread().getId()  );

            String baseUrl =
"https://accounts.google.com/signup/v2?biz=true&flowEntry=SignUp";
            driver.get(baseUrl);

            // Let's locate the first name text field by its id.
            WebElement firstNameTF = driver.findElement(By.id("firstName"));
            firstNameTF.sendKeys("Myname");

            // Let's locate the surname name text field by its name.
            WebElement surNameTF = driver.findElement(By.name("lastName"));
            surNameTF.sendKeys("latName");

            // The Next button using its class
            WebElement nextButton = driver.findElement(By.className("VfPpkd-
vQzf8d"));
            System.out.println("Text on the button is " +
nextButton.getText());
      }


      @Test(groups= {"Dummy"}, dependsOnGroups = {"Account Creation"})
      public void f1() {
            System.out.printf("Inside %s and thread-id is %s \n","f1",
Thread.currentThread().getId()  );
      }
      @Test()
      public void f2() {
            System.out.printf("Inside %s and thread-id is %s \n","f2",
Thread.currentThread().getId()  );
      }

      @Test()
      public void testLocalPage() {
            System.out.printf("Inside %s and thread-id is %s
\n","testLocalPage", Thread.currentThread().getId()  );

            driver.get(baseUrl);

            WebElement selectMonthElement =
driver.findElement(By.id("month"));
            Select selectMonth = new Select(selectMonthElement);
```

```
            System.out.println("selectMonth is muti select " +
selectMonth.isMultiple());

            selectMonth.selectByIndex(0);
            selectMonth.selectByIndex(6);

            List<WebElement> optionsList =
selectMonth.getAllSelectedOptions();

            for (WebElement option : optionsList)
                  System.out.println("selectMonth selected options are " +
option.getText());
      }

      @BeforeMethod
      public void beforeMethod() {
            System.out.printf("Inside %s and thread-id is %s
\n","beforeMethod", Thread.currentThread().getId()  );
      }

      @AfterMethod
      public void afterMethod() {
            System.out.printf("Inside %s and thread-id is %s
\n","afterMethod", Thread.currentThread().getId()  );
      }

      @DataProvider
      public Object[][] dp() {
            return new Object[][] { new Object[] { 1, "a" }, new Object[] {
2, "b" }, };
      }

      @BeforeClass
      public void beforeClass() {
            System.out.printf("Inside %s and thread-id is %s
\n","beforeClass", Thread.currentThread().getId()  );
      }

      @AfterClass
      public void afterClass() {
            System.out.printf("Inside %s and thread-id is %s \n
","afterClass", Thread.currentThread().getId()  );
      }

      @BeforeTest
      public void beforeTest() {
            System.out.printf("Inside %s and thread-id is %s
\n","beforeTest", Thread.currentThread().getId()  );
      }

      @AfterTest
      public void afterTest() {
            System.out.printf("Inside %s and thread-id is %s \n","afterTest",
Thread.currentThread().getId()  );
      }
```

```
	@BeforeSuite
	public void beforeSuite() {
		System.out.printf("Inside %s and thread-id is %s
\n","beforeSuite", Thread.currentThread().getId()   );

	//System.setProperty("webdriver.chrome.driver","C:\\Users\\HomeWk\\Down
loads\\chromedriver_win32\\chromedriver.exe");

		driver = new ChromeDriver();
	}

	@AfterSuite
	public void afterSuite() {
		System.out.printf("Inside %s and thread-id is %s
\n","afterSuite", Thread.currentThread().getId()   );
		driver.quit();
	}

}
```

Testng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test FaceBook" parallel="methods">
    <classes>
      <class name="com.ecommerce.test.FBRegistration"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

OUTPUT

```
[RemoteTestNG] detected TestNG version 7.8.0
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Inside beforeSuite and thread-id is 1
Jul 20, 2023 2:44:32 PM org.openqa.selenium.remote.service.DriverService$Builder getLogOutput
INFO: Driver logs no longer sent to console by default; https://www.selenium.dev/documentation/webdriver/drivers/service/#setting-log
Inside beforeTest and thread-id is 1
Inside beforeClass and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside f2 and thread-id is 1
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside fbAccount and thread-id is 1
Gender is enabled = true
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside googleAccount and thread-id is 1
Text on the button is Next
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside searchGoogle and thread-id is 1
Title of google page is Google
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside testLocalPage and thread-id is 1
selectMonth is muti select true
selectMonth selected options are Jan
selectMonth selected options are Jul
Inside afterMethod and thread-id is 1
Inside beforeMethod and thread-id is 1
Inside f1 and thread-id is 1
Inside afterMethod and thread-id is 1
Inside afterClass and thread-id is 1
 Inside afterTest and thread-id is 1
PASSED: com.ecommerce.test.FBRegistration.f1
PASSED: com.ecommerce.test.FBRegistration.f2
PASSED: com.ecommerce.test.FBRegistration.googleAccount
PASSED: com.ecommerce.test.FBRegistration.searchGoogle
PASSED: com.ecommerce.test.FBRegistration.testLocalPage
PASSED: com.ecommerce.test.FBRegistration.fbAccount
```

10. Evaluating Test Cases.

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>hello-selenium-testng</groupId>
  <artifactId>hello-selenium-testng</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
            <!--

      https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
java -->
            <dependency>
                  <groupId>org.seleniumhq.selenium</groupId>
                  <artifactId>selenium-java</artifactId>
                  <version>4.10.0</version>
            </dependency>

            <!--

      https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
chrome-driver -->
            <dependency>
                  <groupId>org.seleniumhq.selenium</groupId>
                  <artifactId>selenium-chrome-driver</artifactId>
                  <version>4.10.0</version>
            </dependency>

            <!--

      https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-
firefox-driver -->
            <dependency>
                  <groupId>org.seleniumhq.selenium</groupId>
                  <artifactId>selenium-firefox-driver</artifactId>
                  <version>4.10.0</version>
            </dependency>
            <dependency>
                  <groupId>org.testng</groupId>
                  <artifactId>testng</artifactId>
                  <version>7.8.0</version>
                  <scope>compile</scope>
            </dependency>

            <dependency>
                  <groupId>org.uncommons</groupId>
                  <artifactId>reportng</artifactId>
                  <version>1.1.4</version>
            </dependency>
```

```
            <dependency>
                  <groupId>com.google.inject</groupId>
                  <artifactId>guice</artifactId>
                  <version>4.2.2</version>
            </dependency>

      </dependencies>
      <build>
            <plugins>
                  <plugin>
                        <groupId>org.apache.maven.plugins</groupId>
                        <artifactId>maven-compiler-plugin</artifactId>
                        <configuration>
                              <source>1.8</source>
                              <target>1.8</target>
                        </configuration>
                  </plugin>
                  <plugin>
                        <groupId>org.apache.maven.plugins</groupId>
                        <artifactId>maven-surefire-plugin</artifactId>
                        <configuration>
                              <suiteXmlFiles>
                                    <suiteXmlFile>testng.xml</suiteXmlFile>
                              </suiteXmlFiles>
                              <properties>
                                    <property>
                                          <name>usedefaultlisteners</name>
                                          <value>false</value>
                                    </property>
                                    <property>
                                          <name>listener</name>

      <value>org.uncommons.reportng.HTMLReporter,

      org.uncommons.reportng.JUnitXMLReporter
                                          </value>
                                    </property>
                              </properties>
                        </configuration>
                  </plugin>
            </plugins>
      </build>
</project>
```

Testing.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <!-- <test thread-count="5" name="Test FaceBook" parallel="methods"> -->
   <test thread-count="5" name="Test FaceBook" parallel="methods">
    <classes>
      <class name="com.ecommerce.test.FBRegistration"/>
    </classes>
```

```
  </test> <!-- Test -->

  <!-- Test
  <test thread-count="5" name="Test Google">
    <classes>
      <class name="com.ecommerce.test.GoogleRegistration"/>
    </classes>
  </test>
  -->

</suite> <!-- Suite -->
```

FBRegistraton.java

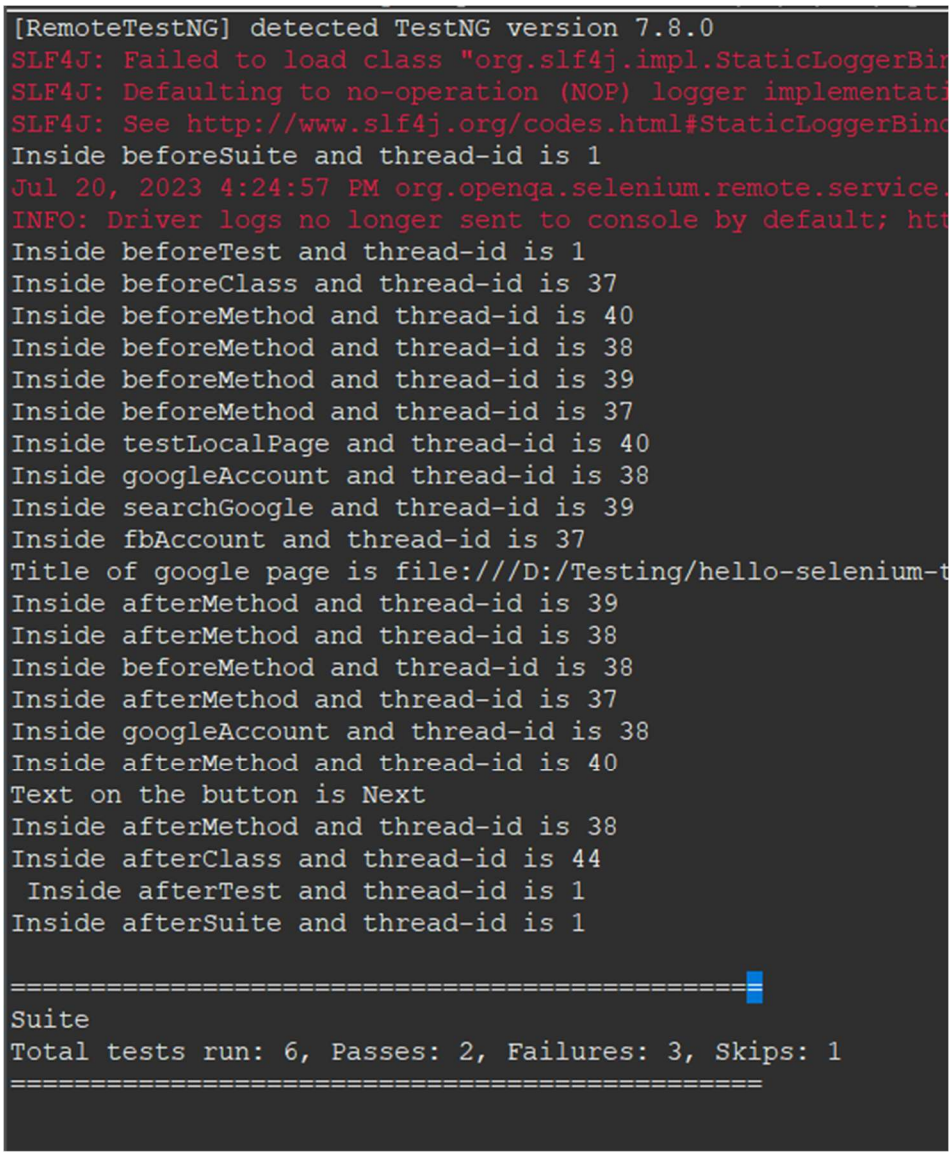
```
package com.ecommerce.test;

import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterSuite;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

@Test(groups = "Class1")
public class FBRegistration {
      public WebDriver driver;
      public String baseUrl = "File:///D:\\Testing\\hello-selenium-
testng\\src\\main\\resources\\test.html";

      @Test(groups = { "Google" })
      public void searchGoogle() {
            System.out.printf("Inside %s and thread-id is %s \n",
"searchGoogle", Thread.currentThread().getId());

            driver.get("http://www.google.com");

            System.out.println("Title of google page is " +
driver.getTitle());
      }

      @Test(groups = { "Account Creation" })
      public void fbAccount() {
```

```
            System.out.printf("Inside %s and thread-id is %s \n",
"fbAccount", Thread.currentThread().getId());

            String baseUrl =
"https://www.facebook.com/r.php?locale=en_GB&display=page";

            driver.get(baseUrl);

            String cssDay = "#day";
            WebElement cssDaySelect =
driver.findElement(By.cssSelector(cssDay));

            Select daySelect = new Select(cssDaySelect);
            daySelect.selectByVisibleText("11");

            WebElement cssGenderRadio =
driver.findElement(By.cssSelector("span > span >
input[type='radio'][value='2']"));
            cssGenderRadio.click();

            System.out.println("Gender is enabled = " +
cssGenderRadio.isSelected());
      }

      @Test(groups = { "Account Creation", "Google" }, dataProvider =
"googleNewUserData")
      public void googleAccount(String fName, String lName) {
            System.out.printf("Inside %s and thread-id is %s \n",
"googleAccount", Thread.currentThread().getId());

            String baseUrl =
"https://accounts.google.com/signup/v2?biz=true&flowEntry=SignUp";
            driver.get(baseUrl);

            // Let's locate the first name text field by its id.
            WebElement firstNameTF = driver.findElement(By.id("firstName"));
            firstNameTF.sendKeys(fName) ;//("Myname");

            // Let's locate the surname name text field by its name.
            WebElement surNameTF = driver.findElement(By.name("lastName"));
            surNameTF.sendKeys(lName);//("latName");

            // The Next button using its class
            WebElement nextButton = driver.findElement(By.className("VfPpkd-
vQzf8d"));
            System.out.println("Text on the button is " +
nextButton.getText());
      }

      @Test(groups = { "Dummy" }, dependsOnGroups = { "Account Creation" })
      public void f1() {
            System.out.printf("Inside %s and thread-id is %s \n", "f1",
Thread.currentThread().getId());
      }

      @Test(enabled = false)
      public void f2() {
```

```
            System.out.printf("Inside %s and thread-id is %s \n", "f2",
Thread.currentThread().getId());
      }

      @Test()
      public void testLocalPage() {
            System.out.printf("Inside %s and thread-id is %s \n",
"testLocalPage", Thread.currentThread().getId());

            driver.get(baseUrl);

            WebElement selectMonthElement =
driver.findElement(By.id("month"));
            Select selectMonth = new Select(selectMonthElement);

            System.out.println("selectMonth is muti select " +
selectMonth.isMultiple());

            selectMonth.selectByIndex(0);
            selectMonth.selectByIndex(6);

            List<WebElement> optionsList =
selectMonth.getAllSelectedOptions();

            for (WebElement option : optionsList)
                  System.out.println("selectMonth selected options are " +
option.getText());
      }

      @BeforeMethod
      public void beforeMethod() {
            System.out.printf("Inside %s and thread-id is %s \n",
"beforeMethod", Thread.currentThread().getId());
      }

      @AfterMethod
      public void afterMethod() {
            System.out.printf("Inside %s and thread-id is %s \n",
"afterMethod", Thread.currentThread().getId());
      }

      @DataProvider(name = "googleNewUserData")
      public Object[][] googleNewUserData() {
            return new Object[][] {

                        { "fname1", "lname1" },
                        { "fname2__2", "lname2__2" }
                  };
      }

      @BeforeClass
      public void beforeClass() {
            System.out.printf("Inside %s and thread-id is %s \n",
"beforeClass", Thread.currentThread().getId());
      }

      @AfterClass
```

```
	public void afterClass() {
		System.out.printf("Inside %s and thread-id is %s \n ",
"afterClass", Thread.currentThread().getId());
	}

	@BeforeTest
	public void beforeTest() {
		System.out.printf("Inside %s and thread-id is %s \n",
"beforeTest", Thread.currentThread().getId());
	}

	@AfterTest
	public void afterTest() {
		System.out.printf("Inside %s and thread-id is %s \n",
"afterTest", Thread.currentThread().getId());
	}

	@BeforeSuite
	public void beforeSuite() {
		System.out.printf("Inside %s and thread-id is %s \n",
"beforeSuite", Thread.currentThread().getId());
		//
System.setProperty("webdriver.chrome.driver","C:\\Users\\HomeWk\\Downloads\\c
hromedriver_win32\\chromedriver.exe");

		driver = new ChromeDriver();
	}

	@AfterSuite
	public void afterSuite() {
		System.out.printf("Inside %s and thread-id is %s \n",
"afterSuite", Thread.currentThread().getId());
		driver.quit();
	}

}
```

OUTPUT



```
[RemoteTestNG] detected TestNG version 7.8.0
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBi
SLF4J: Defaulting to no-operation (NOP) logger implementat
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBin
Inside beforeSuite and thread-id is 1
Jul 20, 2023 4:24:57 PM org.openqa.selenium.remote.service.
INFO: Driver logs no longer sent to console by default; htt
Inside beforeTest and thread-id is 1
Inside beforeClass and thread-id is 37
Inside beforeMethod and thread-id is 40
Inside beforeMethod and thread-id is 38
Inside beforeMethod and thread-id is 39
Inside beforeMethod and thread-id is 37
Inside testLocalPage and thread-id is 40
Inside googleAccount and thread-id is 38
Inside searchGoogle and thread-id is 39
Inside fbAccount and thread-id is 37
Title of google page is file:///D:/Testing/hello-selenium-t
Inside afterMethod and thread-id is 39
Inside afterMethod and thread-id is 38
Inside beforeMethod and thread-id is 38
Inside afterMethod and thread-id is 37
Inside googleAccount and thread-id is 38
Inside afterMethod and thread-id is 40
Text on the button is Next
Inside afterMethod and thread-id is 38
Inside afterClass and thread-id is 44
 Inside afterTest and thread-id is 1
Inside afterSuite and thread-id is 1

===============================================
Suite
Total tests run: 6, Passes: 2, Failures: 3, Skips: 1
===============================================
```

11.Selenium Integration with Jenkins.

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>SeleJenk</groupId>
    <artifactId>SeleJenk</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>2.45.0</version>
        </dependency>
        <dependency>
            <groupId>org.testng</groupId>
            <artifactId>testng</artifactId>
            <version>6.14.2</version>
            <scope>test</scope>
```

```
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-plugin-plugin</artifactId>
                <version>3.6.0</version>
                <configuration>
                    <goalPrefix>plugin</goalPrefix>
                    <outputDirectory>target/dir</outputDirectory>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>
```

NewTest.java

```
package com.example;


import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;
public class NewTest {
```

```
        private WebDriver driver;

        SoftAssert soft=new SoftAssert();

        @Test

        public void testEasy() {

            System.setProperty("webdriver.chrome.driver",
"./Resources/chromedriver.exe");

            driver=new ChromeDriver();

            driver.get("https://www.facebook.com");

            String title = driver.getTitle();

            soft.assertEquals("FB Login",title);

        }

        @BeforeTest

        public void beforeTest() {

            driver = new FirefoxDriver();

        }

        @AfterTest

        public void afterTest() {

            driver.quit();

        }

}
```

Converting the project into TestNG and changing the run configuration

- In the Project Explorer, expand **SeleJenk**
- Right click on **SeleJenk** and choose **TestNG->convert to TestNG**

- **Running the project as Maven test**

- Right click on **SeleJenk**
- Click on **Run AS->Maven Test**

- **Installing Jenkins**

- Jenkins is already installed in your Practice lab.(Refer FSD: Lab Guide - Phase 5)
- Use the following commands to navigate to the above-mentioned directory.

  cd /usr/share

  ls

- **Adding Maven plugins to Jenkins**

- In the Jenkins dashboard, click on **Manage Jenkins**
- Click on **Manage Plugins**
- Select the **Available** tab, then find the **Maven Integration** plugin
- Click **Install** without restart

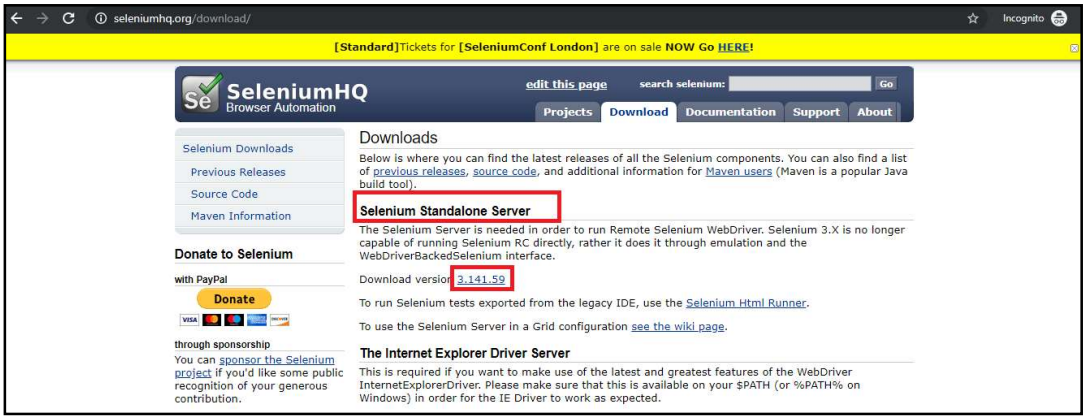
- **Adding the location of pom.xml in Jenkins CI Job**

- Click on **New Item** to create **CI Job**
- Select the **Maven project radio** button and enter **Item Name** as **SeleJenk**
- Click on **Build Environment**
- In **Root POM,** specify the location of pom.xml from your Eclipse workspace
- In **Goals and Options**, type **clean test.** Click on **Save**
- Click on the **SeleJenk** project page and click on the **Build Now** link

12. Set Up Selenium Grid.

Step 1: Setting up Selenium Grid hub

a. Download Selenium standalone Server jar file from https://www.seleniumhq.org/download/ link



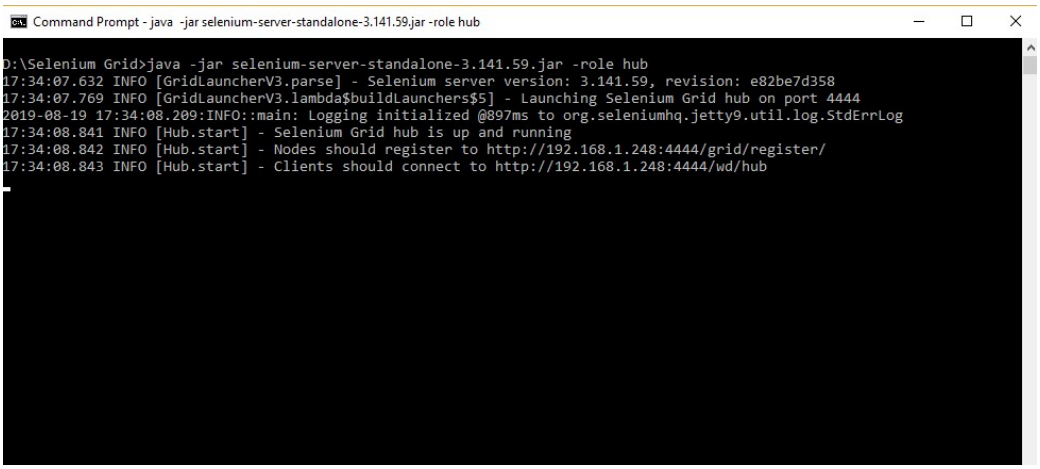
b. Save it in a folder

c. Go to command prompt

d. Navigate to folder structure where you have saved the Selenium standalone Server jar file

e. Type the below command in command prompt

Java -jar selenium-server-standalone-3.141.59.jar -role hub and click on **Enter** button



f. Open the Chrome browser

g. Enter URL as **http://localhost:4444/grid/console** and click on **Enter**

h. Grid console page is loaded as below

Step 2: Setting up the Selenium Grid Nodes

a. Once the Selenium Grid Hub is set up, the next step is to set up Selenium Grid nodes.

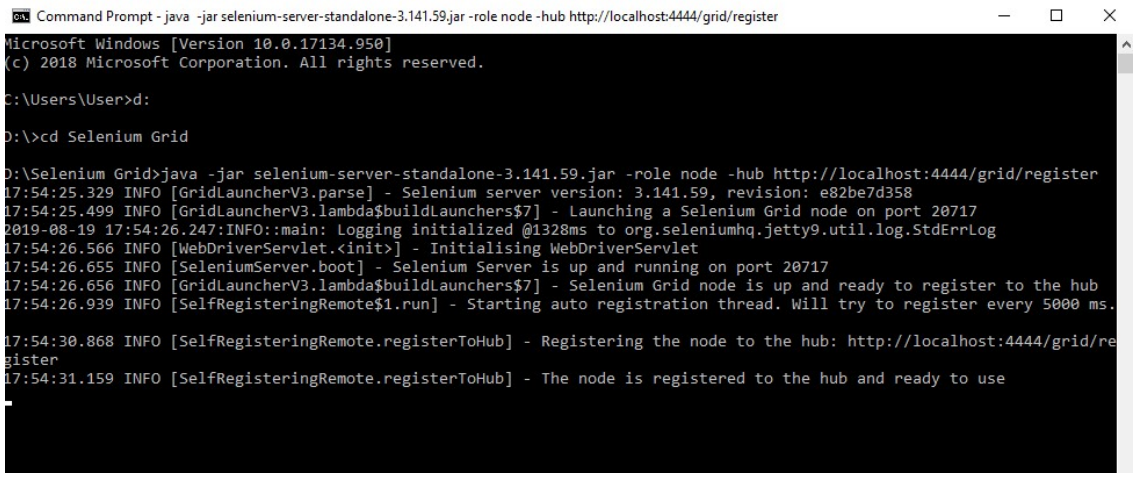
b. Open the new command prompt

c. Navigate to the folder structure where you have saved the Selenium standalone server jar file

d. Type the below command in command prompt

java -jar selenium-server-standalone-3.141.59.jar -role node -hub

http://localhost:4444/grid/register and click on E**nter** button, which looks like

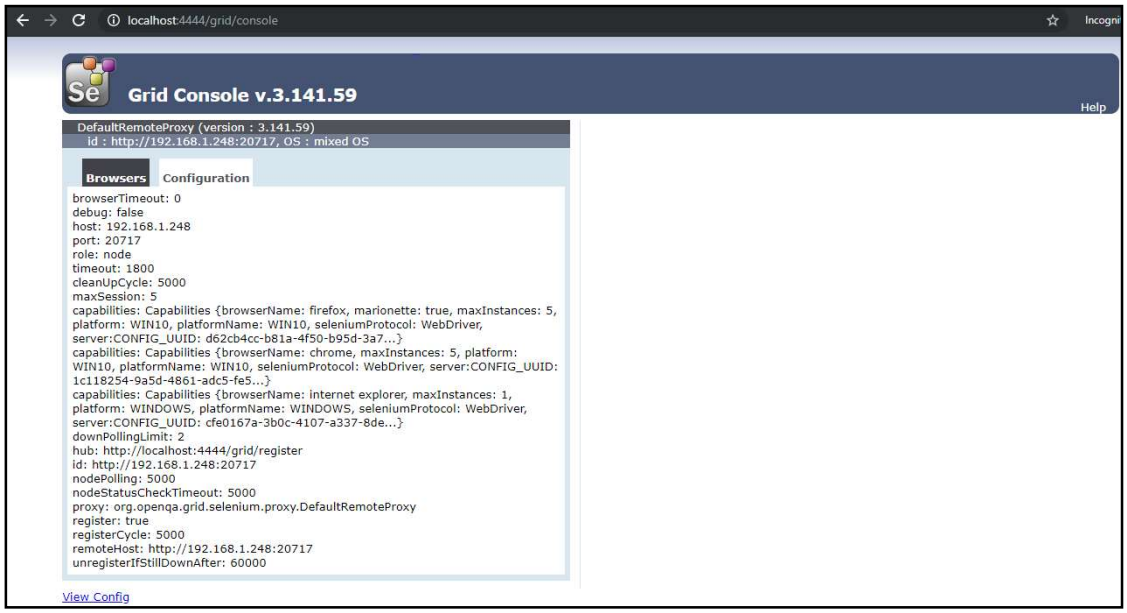


e. Open the browser

f. Enter URL as **http://localhost:4444/grid/console** and click on E**nter**

g. Grid console page is loaded below, which shows **Browsers** by default

h. Click on **Configuration,** which shows Configuration details

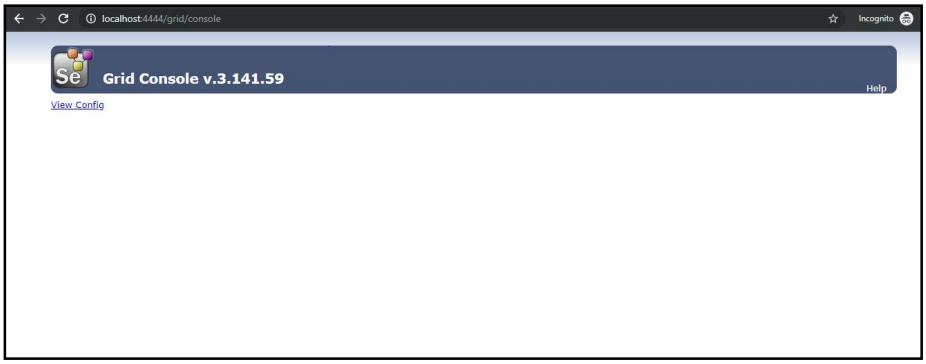


13. Grid Configuration Using JSON.

Step1: Configuring the grid hub using JSON

- Create JSON file for the hub which looks like
  -
```
{
  "port": 4444,
  "newSessionWaitTimeout": -1,
  "servlets" : [],
  "withoutServlets": [],
  "custom": {},
  "capabilityMatcher": "org.openqa.grid.internal.utils.DefaultCapabilityMatcher",
  "throwOnCapabilityNotPresent": true,
  "cleanUpCycle": 5000,
  "role": "hub",
  "debug": false,
  "browserTimeout": 0,
  "timeout": 1800
}
```
- Save it in a folder with a valid name (example: myhub) in which we have saved Selenium standalone Server jar file
- Go to command prompt
- Navigate to folder structure where you have saved the Selenium standalone Server jar file
- Type the below command in command prompt
- **Java -jar selenium-server-standalone-3.141.59.jar -role hub -hubConfig myhub.json** and click on **Enter** button, which looks like
-
```
Command Prompt - java -jar selenium-server-standalone-3.141.59.jar -role hub -hubConfig myhub.json
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>d:

D:\>cd Selenium Grid

D:\Selenium Grid>java -jar selenium-server-standalone-3.141.59.jar -role hub -hubConfig myhub.json
15:44:15.021 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
15:44:15.157 INFO [GridLauncherV3.lambda$buildLaunchers$5] - Launching Selenium Grid hub on port 4444
2019-08-20 15:44:15.581:INFO::main: Logging initialized @929ms to org.seleniumhq.jetty9.util.log.StdErrLog
15:44:16.217 INFO [Hub.start] - Selenium Grid hub is up and running
15:44:16.218 INFO [Hub.start] - Nodes should register to http://192.168.1.248:4444/grid/register/
15:44:16.218 INFO [Hub.start] - Clients should connect to http://192.168.1.248:4444/wd/hub
15:49:28.797 INFO [DefaultGridRegistry.add] - Registered a node http://192.168.1.248:5555
```
- Open the Chrome browser
- Enter URL as ‘http://localhost:4444/grid/console’ and click on enter

- Grid console page is loaded as below



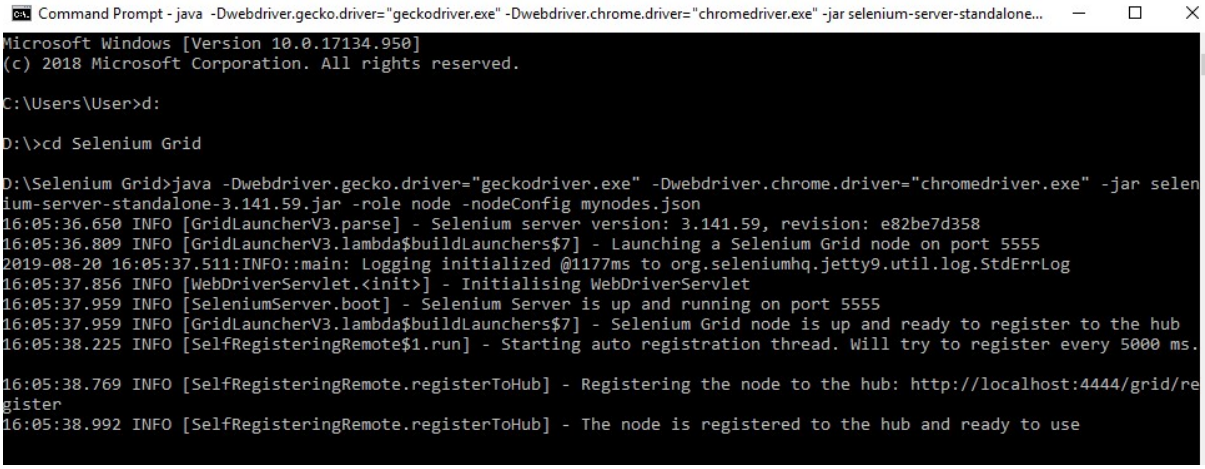
Step 2: Configuring the grid nodes using JSON

- Once the Selenium Grid Hub using JSON is configured, the next step is to configure Selenium Grid nodes using JSON.
- Create JSON file for node, which looks like:

```
{
  "capabilities":
  [
    {
      "browserName": "firefox",
      "maxInstances": 5,
      "seleniumProtocol": "WebDriver"
    },
    {
      "browserName": "chrome",
      "maxInstances": 5,
      "seleniumProtocol": "WebDriver"
    }
  ],
  "proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",
  "maxSession": 5,
  "port": 5555,
  "register": true,
  "registerCycle": 5000,
  "hub": "http://localhost:4444",
  "nodeStatusCheckTimeout": 5000,
  "nodePolling": 5000,
  "role": "node",
  "unregisterIfStillDownAfter": 60000,
  "downPollingLimit": 2,
  "debug": false,
  "servlets" : [],
  "withoutServlets": [],
  "custom": {}
}
```

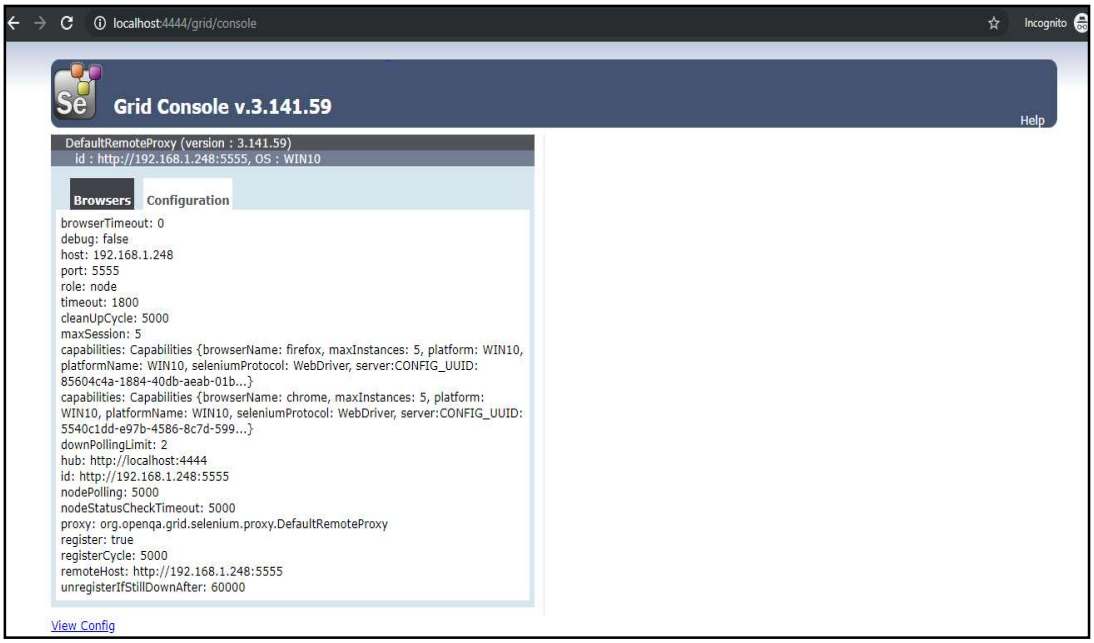
- Save it in a folder with a valid name (example: mynode) in which we have saved Selenium standalone Server jar file
- Open the new command prompt
- Navigate to the folder structure where you have saved the Selenium standalone server jar file
- Type the below command in command prompt
  java -Dwebdriver.gecko.driver="geckodriver.exe" -Dwebdriver.chrome.driver="chromedriver.exe" -jar selenium-server-standalone-3.141.59.jar -role node -nodeConfig mynodes.json

  and click on E**nter** button, which looks like



- Open the browser
- Enter URL as **http://localhost:4444/grid/console** and click on E**nter**
- Grid console page is loaded, which shows **Browsers** by default

- Click on **Configuration** which shows Configuration details



14. Running Tests on Selenium Grid.

Step 1: Running the Tests on Selenium grid

- Open Eclipse
- Click on **Package** and navigate to **New** --> **Class**
- Give a valid Class name (example: GridTest)
- Check the **public static void main** checkbox and click on **finish**, which will create a blank Java class
- Write the desired capabilities in the class, which looks like

```
package testing.sidTesting;

import org.openqa.selenium.Platform;
import org.openqa.selenium.remote.DesiredCapabilities;

public class GridTest {

  public static void main(String[] args) {
    DesiredCapabilities cap = new DesiredCapabilities();
    cap.setBrowserName("chrome");
    cap.setPlatform(Platform.WIN10);
  }

}
```

- Start the selenium grid hub in command prompt using **java -jar selenium-server-standalone-3.141.59.jar -role hub** command
- Start the selenium grid node in Command prompt using **java -Dwebdriver.chrome.driver="chromedriver.exe -jar selenium-server-standalone-3.141.59.jar -role node -hub** http://localhost:4444/grid/register command
- Go to eclipse and add a statement for remoteWebdriver, which has an implementation of WebDriver, should pass the hub port (http://192.168.1.248:4444/wd/hub), and DesiredCapabilities object as parameters

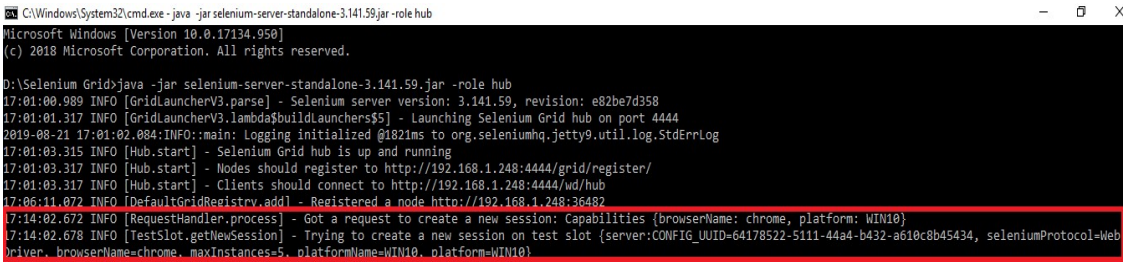
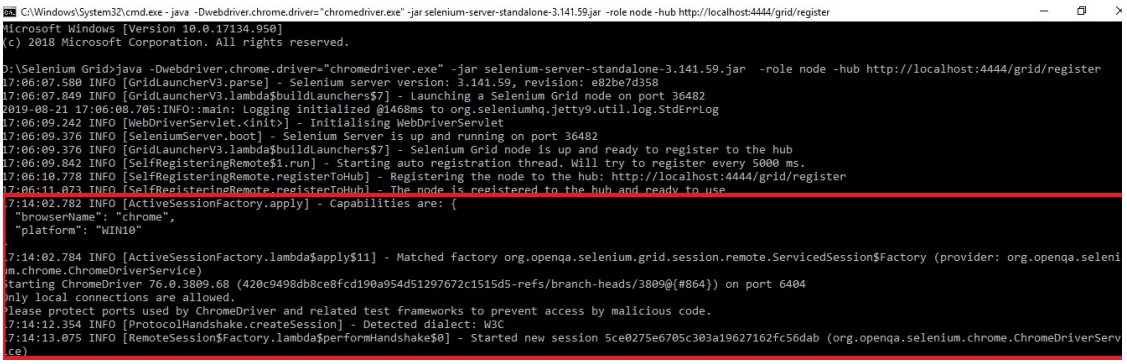
- Write Selenium code to open the browser and navigate to any web page (example: Google page)

```
import java.net.URL;

import org.openqa.selenium.Platform;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;

public class GridTest {

  public static void main(String[] args) throws MalformedURLException {
    DesiredCapabilities cap = new DesiredCapabilities();
    cap.setBrowserName("chrome");
    cap.setPlatform(Platform.WIN10);


    URL url = new URL("http://192.168.1.248:4444/wd/hub");
    WebDriver driver = new RemoteWebDriver(url, cap);

    driver.get("https://www.google.com");
    System.out.println("Google Title: " + driver.getTitle());

    driver.close();
  }

}
```

- Execute the Java program by right-clicking on the program and navigating to **Run As**--> **1 Java Application**
- This is how it looks like in Eclipse console

```
Aug 21, 2019 5:14:13 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Google Title: Google
```

- We can see the capabilities passed through are displayed in both command prompts in server (hub) as well as in clients (node)
- Selenium grid hub in command prompt with desired capabilities looks like



- Selenium grid node in command prompt with desired capabilities looks like