

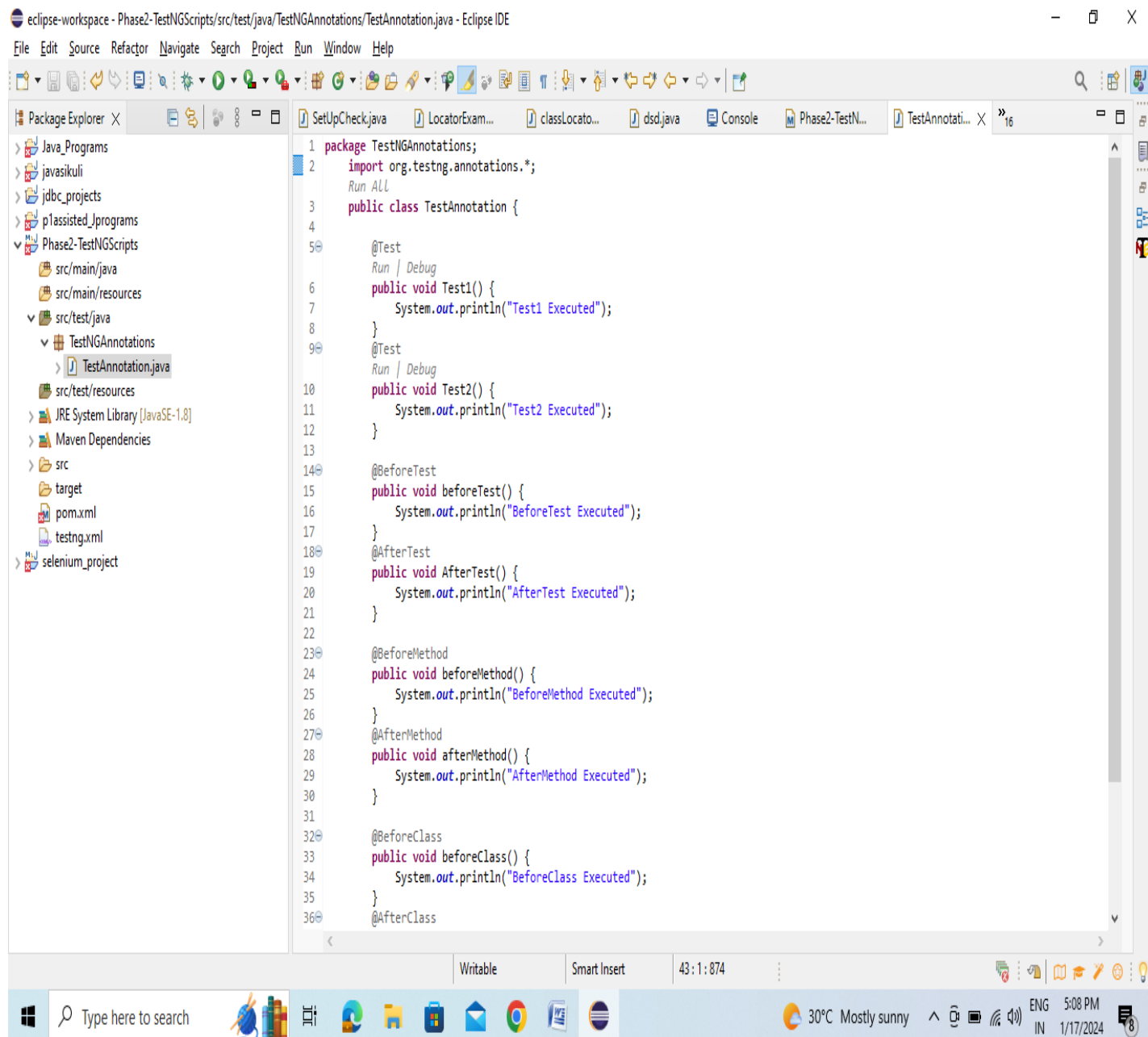
PHASE 2

EMP ID: 2587325

ASSISTED PRACTICE PROJECTS

NAME: HIRANMAI

1. Perform All Test Annotations



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure, including the 'TestNGAnnotations' package. The main editor window shows the code for 'TestAnnotation.java'. The code includes package declarations, imports, and several test methods and annotations.

```
1 package TestNGAnnotations;
2 import org.testng.annotations.*;
3
4 public class TestAnnotation {
5
6     @Test
7     public void Test1() {
8         System.out.println("Test1 Executed");
9     }
10
11     @Test
12     public void Test2() {
13         System.out.println("Test2 Executed");
14     }
15
16     @BeforeTest
17     public void beforeTest() {
18         System.out.println("BeforeTest Executed");
19     }
20
21     @AfterTest
22     public void AfterTest() {
23         System.out.println("AfterTest Executed");
24     }
25
26     @BeforeMethod
27     public void beforeMethod() {
28         System.out.println("BeforeMethod Executed");
29     }
30
31     @AfterMethod
32     public void afterMethod() {
33         System.out.println("AfterMethod Executed");
34     }
35
36     @BeforeClass
37     public void beforeClass() {
38         System.out.println("BeforeClass Executed");
39     }
40
41     @AfterClass
```

Package Explorer X

- Java_Programs
- javasikuli
- jdbc_projects
- plassisted_programs
- Phase2-TestNGScripts
 - src/main/java
 - src/main/resources
 - src/test/java
 - TestNGAnnotations
 - TestAnnotation.java
 - src/test/resources
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - pom.xml
 - testng.xml
 - selenium_project

SetUpCheck.java LocatorExam... classLocato...

<terminated> TestAnnotation [TestNG] C:\Users\GVH\p2\pool\plugins\org.a

[RemoteTestNG] detected TestNG version 7.7.1

SLF4J: No SLF4J providers were found.

SLF4J: Defaulting to no-operation (NOP) logger implementation

SLF4J: See <https://www.slf4j.org/codes.html#noProviders> for

BeforeTest Executed

BeforeClass Executed

BeforeMethod Executed

Test1 Executed

AfterMethod Executed

BeforeMethod Executed

Test2 Executed

AfterMethod Executed

AfterClass Executed

AfterTest Executed

PASSED: TestNGAnnotations.TestAnnotation.Test1

PASSED: TestNGAnnotations.TestAnnotation.Test2

=====

Default test

Tests run: 2, Failures: 0, Skips: 0

=====

Default suite

Total tests run: 2, Passes: 2, Failures: 0, Skips: 0

=====

Outline Results of running class TestAnnotation X

Search: Passed: 2 Failed: 0 Skipped: 0 tests: 1/1 Methods: 2 (863 ms)

All Tests Failed Tests Summary

Default suite (2/0/0/0) (0.022 s) Failure Exception

Default test (0.022 s)

TestNGAnnotations.TestAnnotation

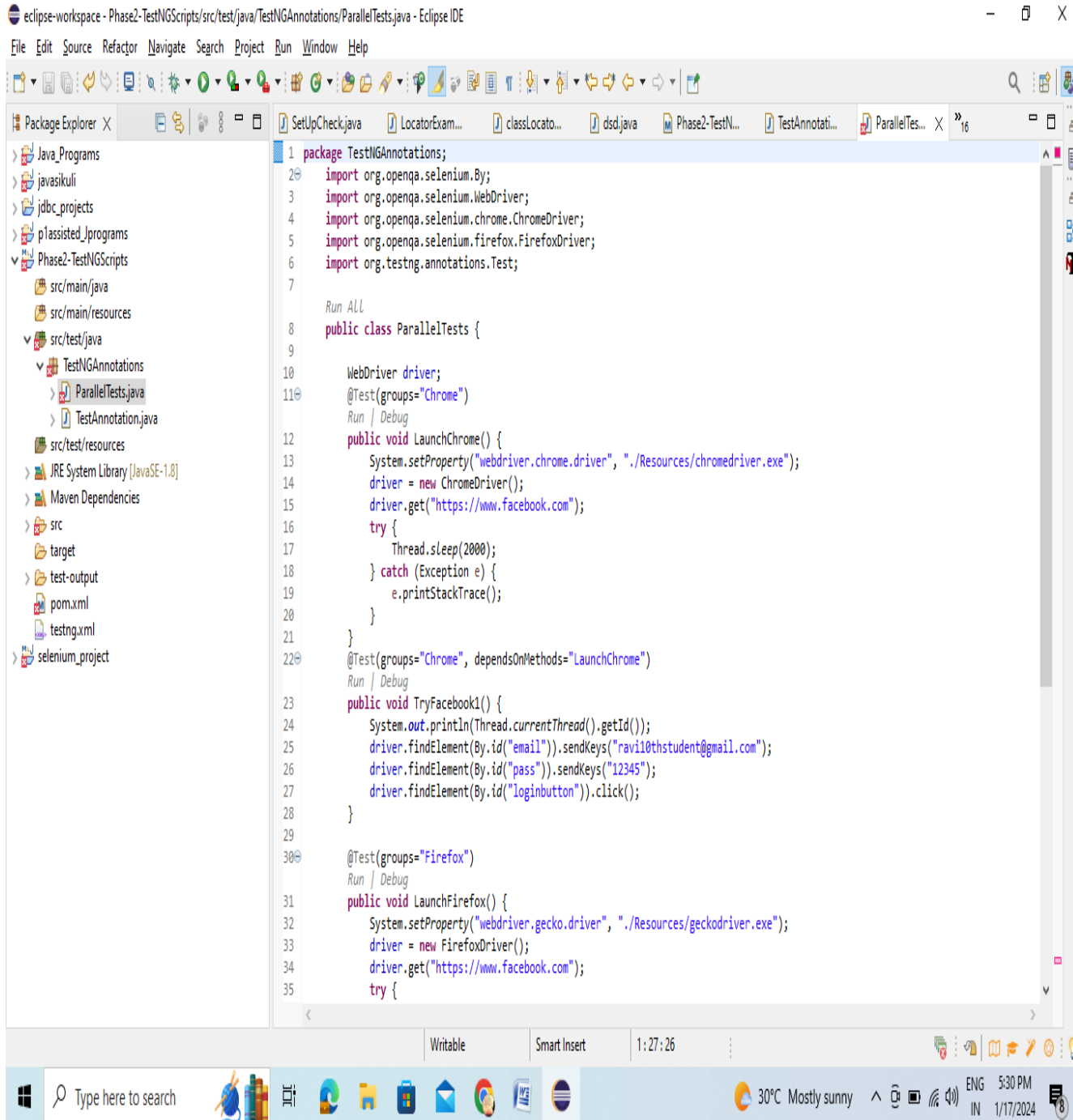
Test1 (0.012 s)

Test2 (0.01 s)

Type here to search

30°C Mostly sunny 5:08 PM 1/17/2024

2. Group Test Cases and Parallel Test Execution

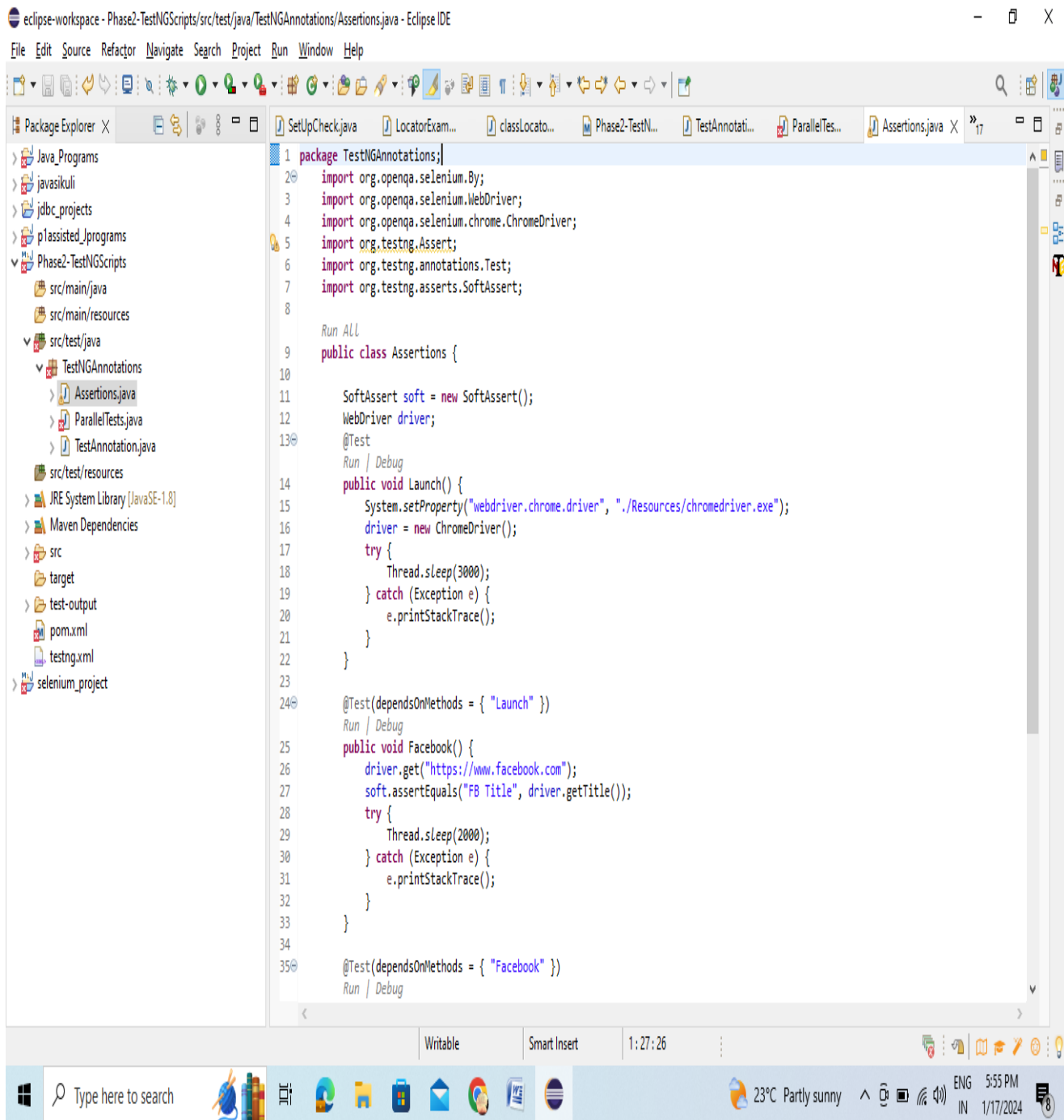


The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure, with the file `ParallelTests.java` selected under the `TestNGAnnotations` package. The main editor window shows the following Java code:

```
1 package TestNGAnnotations;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.chrome.ChromeDriver;
5 import org.openqa.selenium.firefox.FirefoxDriver;
6 import org.testng.annotations.Test;
7
8 Run All
9 public class ParallelTests {
10
11     WebDriver driver;
12     @Test(groups="Chrome")
13     Run / Debug
14     public void LaunchChrome() {
15         System.setProperty("webdriver.chrome.driver", "./Resources/chromedriver.exe");
16         driver = new ChromeDriver();
17         driver.get("https://www.facebook.com");
18         try {
19             Thread.sleep(2000);
20         } catch (Exception e) {
21             e.printStackTrace();
22         }
23     }
24     @Test(groups="Chrome", dependsOnMethods="LaunchChrome")
25     Run / Debug
26     public void TryFacebook1() {
27         System.out.println(Thread.currentThread().getId());
28         driver.findElement(By.id("email")).sendKeys("ravi10thstudent@gmail.com");
29         driver.findElement(By.id("pass")).sendKeys("12345");
30         driver.findElement(By.id("loginbutton")).click();
31     }
32     @Test(groups="Firefox")
33     Run / Debug
34     public void LaunchFirefox() {
35         System.setProperty("webdriver.gecko.driver", "./Resources/geckodriver.exe");
36         driver = new FirefoxDriver();
37         driver.get("https://www.facebook.com");
38         try {
```

The bottom status bar shows the system clock at 1:27:26, temperature at 30°C, and the date 1/17/2024.

3. Evaluating Test Cases



The screenshot displays the Eclipse IDE interface. The title bar indicates the workspace is 'eclipse-workspace - Phase2-TestNGScripts/src/test/java/TestNGAnnotations/Assertions.java - Eclipse IDE'. The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations, running, and debugging. The Package Explorer on the left shows a project structure with folders like 'Java_Programs', 'jdbc_projects', 'p1assisted_lprograms', and 'Phase2-TestNGScripts'. The 'Phase2-TestNGScripts' folder is expanded, showing subfolders 'src/main/java', 'src/main/resources', 'src/test/java', and 'src/test/resources'. The 'TestNGAnnotations' folder is also expanded, showing 'Assertions.java', 'ParallelTests.java', and 'TestAnnotation.java'. The main editor window displays the code for 'Assertions.java'. The code includes package declarations, imports for Selenium and TestNG, and two test methods: 'Launch()' and 'Facebook()'. The 'Launch()' method sets the WebDriver to Chrome and sleeps for 3000ms. The 'Facebook()' method navigates to 'https://www.facebook.com', asserts the title is 'FB Title', and sleeps for 2000ms. The status bar at the bottom shows 'Writable', 'Smart Insert', and the time '1:27:26'. The Windows taskbar at the very bottom shows the search bar, task view, and several application icons, along with system information like '23°C Partly sunny' and the date '1/17/2024'.

```
1 package TestNGAnnotations;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.chrome.ChromeDriver;
5 import org.testng.Assert;
6 import org.testng.annotations.Test;
7 import org.testng.asserts.SoftAssert;
8
9 Run ALL
10 public class Assertions {
11     SoftAssert soft = new SoftAssert();
12     WebDriver driver;
13     @Test
14     Run | Debug
15     public void Launch() {
16         System.setProperty("webdriver.chrome.driver", "./Resources/chromedriver.exe");
17         driver = new ChromeDriver();
18         try {
19             Thread.sleep(3000);
20         } catch (Exception e) {
21             e.printStackTrace();
22         }
23     }
24     @Test(dependsOnMethods = { "Launch" })
25     Run | Debug
26     public void Facebook() {
27         driver.get("https://www.facebook.com");
28         soft.assertEquals("FB Title", driver.getTitle());
29         try {
30             Thread.sleep(2000);
31         } catch (Exception e) {
32             e.printStackTrace();
33         }
34     }
35     @Test(dependsOnMethods = { "Facebook" })
36     Run | Debug
```

4. Reports Using Extent Reports

The screenshot displays the Eclipse IDE interface with the following components:

- Package Explorer:** Shows the project structure with folders like `Java_Programs`, `javasikuli`, `jdbc_projects`, `plassisted_Jprograms`, and `Phase2-TestNGScripts`. The `TestNGAnnotations` package is expanded, showing `Assertions.java`, `ExtentReportsClass.java`, `ParallelTests.java`, and `TestAnnotation.java`.
- Code Editor:** Displays the `ExtentReportsClass.java` file with the following code:

```
1 package TestNGAnnotations;
2
3 import java.io.File;
4 import org.testng.Assert;
5 import org.testng.ITestResult;
6 import org.testng.SkipException;
7 import org.testng.annotations.AfterMethod;
8 import org.testng.annotations.AfterTest;
9 import org.testng.annotations.BeforeTest;
10 import org.testng.annotations.Test;
11
12 import com.relevantcodes.extentreports.ExtentReports;
13 import com.relevantcodes.extentreports.ExtentTest;
14 import com.relevantcodes.extentreports.LogStatus;
15
16 Run All
17 public class ExtentReportsClass{
18     Extentreports extent;
19     ExtentTest logger;
20     @BeforeTest
21     public void startReport(){
22         //ExtentReports(String filePath,Boolean replaceExisting)
23         //filepath - path of the file, in .htm or .html format - path where your report needs to generate.
24         //replaceExisting - Setting to overwrite (TRUE) the existing file or append to it
25         //True (default): the file will be replaced with brand new markup, and all existing data will be lost. Use this option to create
26         //False: existing data will remain, new tests will be appended to the existing report. If the supplied path does not exist, a new
27         extent = new ExtentReports(System.getProperty("user.dir") + "/test-output/STMExtentReport.html", true);
28         //extent.addSystemInfo("Environment","Environment Name")
29         extent
30             .addSystemInfo("Host Name", "SoftwareTesting")
31             .addSystemInfo("Environment", "Automation Testing")
32             .addSystemInfo("User Name", "TestEngineer");
33         //loading the external xml file (i.e., extent-config.xml) that was placed under the base directory
34         //You could find the xml file below. Create xml file in your project and copy paste the code mentioned below
35         extent.loadConfig(new File(System.getProperty("user.dir")+"\\extent-config.xml"));
36     }
37     @Test
38     Run | Debug
39     public void passTest(){
```
- Bottom Bar:** Shows the status bar with the message "Extentreports cannot be resolved to a type", the "Writable" and "Smart Insert" modes, the time "17:12:532", and the system tray with the date "1/17/2024".

5. Exporting Reports in Excel .

eclipse-workspace - Phase2-TestNGScripts/src/test/java/TestNGAnnotations/TEST_01.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

- Java_Programs
- javasikuli
- jdbc_projects
- plassisted_programs
- Phase2-TestNGScripts
 - src/main/java
 - src/main/resources
 - src/test/java
 - TestNGAnnotations
 - Assertions.java
 - ExtendsReportsClass.java
 - ParallelTests.java
 - TEST_01.java
 - TestAnnotation.java
 - src/test/resources
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - test-output
 - pom.xml
 - testng.xml
 - selenium_project

```
1 package TestNGAnnotations;
2 {
3
4 import org.opengs.selenium.By;
5 import org.opengs.selenium.WebDriver;
6 import org.opengs.selenium.chrome.ChromeDriver;
7 import org.testng.annotations.AfterMethod;
8 import org.testng.annotations.BeforeMethod;
9 import org.testng.annotations.Test;
10
11 import base.Baseclass;
12
13 Run ALL
14 public class TEST_01 extends Baseclass {
15
16 @Test
17 Run | Debug
18 public void t_001()
19 {
20 driver.findElement(By.xpath("//*[@contains(text(),'Categories')][1]")).click();
21 driver.findElement(By.xpath("//*[@contains(text(),'Central')][1]")).click();
22 System.out.println("Test_01 executed successfully");
23 }
24 }
25
26
27
28 • Write the test script in the Test_02 class.
29
30 package Testcases;
31
32 import org.opengs.selenium.By;
33 import org.testng.annotations.Test;
34
35 import base.Baseclass;
36
37 Run ALL
```

Writable Smart Insert 98 : 25 : 2239

27°C Partly cloudy 6:09 PM 1/17/2024

7.XML Parsers

There are mainly three types of XML parsers:

- 1) SAX
- 2) DOM
- 3) Pull parser

1: SAX

SAX stands for 'Simple API for XML'. It does not create any internal structure. Clients do not know what methods to call. They just override the methods of the API and place his own code inside the method. It is an event-based parser, it works as an event handler in Java.

- Advantages
 - Since it reads each unit of XML, it creates an event so that the calling program can use it.
 - SAX uses what it likes to, by ignoring the bits which it doesn't care about.
 - It is memory efficient.
 - It's very fast and works for huge documents.
- Disadvantages
 - The main disadvantage of SAX is that the Calling program must keep track of everything it might ever need.
 - Since its Event-based, its API is less Intuitive.

2: DOM

DOM stands for 'Document Object Model'. A DOM Parser creates an internal structure in memory which is a DOM document object and the client applications get information of the original XML document by invoking methods on this document object. DOM Parser has a tree-based structure.

- Advantages

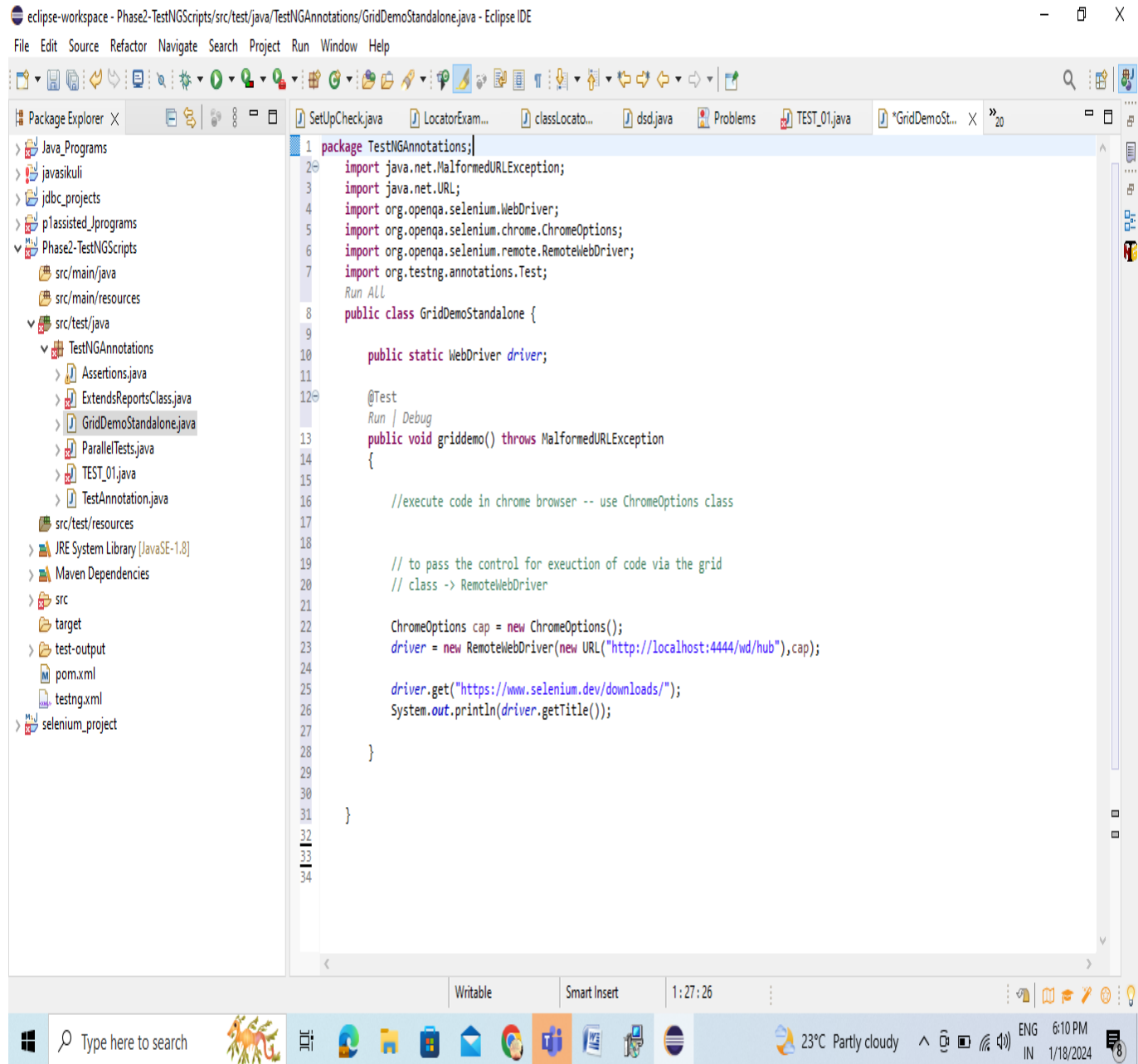
- It supports both Read and Write operations.
- It is preferred when there is random access to widely separated parts of the documents required.
- It builds the entire XML document representation in memory and then hands the calling program the whole chunk of memory.
- Disadvantages
 - It consumes more memory since the whole XML document will be loaded into the memory.

3: Pull Parser

Pull parser waits for the application to come calling. That is, they ask for the next available event, and the application basically loops until it runs out of XML.

- Advantages
 - It is designed to be used with large data sources.
 - Pull parser chooses to skip the events (whole section of the document) which it is not interested in.

8. Grid Configuration Using JSON



The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure, with the file `GridDemoStandalone.java` selected under the `TestNGAnnotations` package. The main editor window shows the following Java code:

```
1 package TestNGAnnotations;
2 import java.net.MalformedURLException;
3 import java.net.URL;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeOptions;
6 import org.openqa.selenium.remote.RemoteWebDriver;
7 import org.testng.annotations.Test;
8 Run ALL
9 public class GridDemoStandalone {
10     public static WebDriver driver;
11
12     @Test
13     Run | Debug
14     public void griddemo() throws MalformedURLException
15     {
16         //execute code in chrome browser -- use ChromeOptions class
17
18         // to pass the control for exeuction of code via the grid
19         // class -> RemoteWebDriver
20
21         ChromeOptions cap = new ChromeOptions();
22         driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"),cap);
23
24         driver.get("https://www.selenium.dev/downloads/");
25         System.out.println(driver.getTitle());
26
27     }
28
29 }
30
31 }
32
33
34
```

The bottom status bar shows the time as 1:27:26. The Windows taskbar at the very bottom displays the search bar, taskbar icons, and system tray information including temperature (23°C), weather (Partly cloudy), and date/time (6:10 PM, 1/18/2024).

9. Running Tests on Selenium Grid on Multiple Browsers

eclipse-workspace - Phase2-TestNGScripts/src/test/java/TestNGAnnotations/GridDemoHubandNode.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

- Java_Programs
- javasikuli
- jdbc_projects
- plassisted_programs
- Phase2-TestNGScripts
 - src/main/java
 - src/main/resources
 - src/test/java
 - TestNGAnnotations
 - Assertions.java
 - ExtendsReportsClass.java
 - GridDemoHubandNode.java
 - GridDemoStandalone.java
 - ParallelTests.java
 - TEST_01.java
 - TestAnnotation.java
 - src/test/resources
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - test-output
 - pom.xml
 - testng.xml
- selenium_project

```
1 package TestNGAnnotations;
2 import java.net.MalformedURLException;
3 import java.net.URL;
4 import org.openqa.selenium.Platform;
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.chrome.ChromeOptions;
7 import org.openqa.selenium.remote.DesiredCapabilities;
8 import org.openqa.selenium.remote.RemoteWebDriver;
9 import org.testng.annotations.Test;
10 Run ALL
11 public class GridDemoHubandNode {
12
13     public static WebDriver driver;
14
15     @Test
16     Run | Debug
17     public void griddemo() throws MalformedURLException
18     {
19         DesiredCapabilities cap = null;
20
21         cap = new DesiredCapabilities();
22
23         cap.setBrowserName("firefox");
24         cap.setPlatform(Platform.WINDOWS);
25
26         driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"),cap);
27
28         driver.get("https://www.selenium.dev/downloads/");
29         System.out.println(driver.getTitle());
30
31     }
32
33 }
34
35
36
```

Writable Smart Insert 32:9:881

Type here to search

23°C Partly cloudy 6:17 PM 1/18/2024

10. Page Object Design Pattern.

The screenshot displays the Eclipse IDE interface with the following components:

- Package Explorer (Left):** Shows the project structure. The 'Phase2-TestNGScripts' project is expanded, revealing the 'src/test/java' directory. Inside, the 'TestNGAnnotations' package is expanded, showing the 'Testbase.java' file.
- Editor (Center):** Displays the code for 'Testbase.java'. The code is as follows:

```
1 package TestNGAnnotations;
2 import java.awt.Desktop;
3 import java.io.File;
4 import java.io.IOException;
5 import java.lang.reflect.Method;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.chrome.ChromeDriver;
8 import org.testng.ITestContext;
9 import org.testng.ITestResult;
10 import org.testng.annotations.AfterMethod;
11 import org.testng.annotations.AfterSuite;
12 import org.testng.annotations.AfterTest;
13 import org.testng.annotations.BeforeSuite;
14 import org.testng.annotations.BeforeTest;
15
16 public class Testbase {
17
18     // add the code to open the browser
19     // add the code to generate extent report
20
21     public static ExtentReports ex;
22     public static WebDriver driver;
23     public static ExtentTest extentTest;
24
25
26     public static void openBrowser()
27     {
28         driver = new ChromeDriver();
29         driver.manage().window().maximize();
30         driver.manage().deleteAllCookies();
31         driver.get("https://tutorialsninja.com/demo/index.php?route=common/home");
32     }
33
34     @BeforeSuite // this will be the first method executed before the suite of test cases run
35     public void InitializeExtentReport() throws IOException
36     {
37         ex = new ExtentReports();
38         ExtentSparkReporter sparkreporter = new ExtentSparkReporter("projectReport.html");
39         ex.attachReporter(sparkreporter);
```
- Bottom Bar:** Shows the status bar with 'Writable', 'Smart Insert', and '38: 96: 1277'.
- Taskbar (Bottom):** Displays the Windows taskbar with the search bar, task view button, and several application icons. The system tray shows the temperature (23°C), weather (Partly cloudy), and the date/time (ENG 6:48 PM IN 1/18/2024).

11. Apache POI Configuration.

eclipse-workspace - Phase2-TestNGScripts/src/test/java/TestNGAnnotations/WriteToExcel.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

- Java_Programs
 - javasikuli
 - jdbc_projects
 - plassisted_programs
 - Phase2-TestNGScripts
 - src/main/java
 - src/main/resources
 - src/test/java
 - TestNGAnnotations
 - Assertions.java
 - ExtendsReportsClass.java
 - GridDemoHUBandNode.java
 - GridDemoStandalone.java
 - ParallelTests.java
 - TEST_01.java
 - TestAnnotation.java
 - Testbase.java
 - WriteToExcel.java
 - src/test/resources
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - test-output
 - pom.xml
 - testng.xml
 - selenium_project

```
1 package TestNGAnnotations;
2 import java.io.File;
3 import java.io.FileOutputStream;
4 import java.io.IOException;
5 import org.apache.poi.xssf.usermodel.XSSFCell;
6 import org.apache.poi.xssf.usermodel.XSSFRow;
7 import org.apache.poi.xssf.usermodel.XSSFSheet;
8 import org.apache.poi.xssf.usermodel.XSSFWorkbook;
9 public class WriteToExcel {
10     public static void main(String[] args) throws IOException {
11         // TODO Auto-generated method stub
12
13         XSSFWorkbook workbook = new XSSFWorkbook();
14
15         XSSFSheet sheet= workbook.createSheet("SheettestData");
16
17         Object [][] data = {{ "Name", "Location"},
18                             { "Sonal", "Gurgaon"},
19                             { "Ravi", "Hyderabd"},
20                             { "Marc", "London"},
21                             { "Peter", "NewYork"} };
22
23         // see how many rows have to created to write the data
24
25         int rows = data.length; //5
26         int col = data[0].length; // 2 columns
27
28         for(int r=0;r<rows;r++)
29         {
30             XSSFRow row = sheet.createRow(r);
31
32             for(int c=0;c<col;c++)
33             {
34                 XSSFCell cell = row.createCell(c);
35                 // get data from the above array and write in to the cell
36
37                 Object cellvalue = data[r][c];
38                 |
39                 // insert the value in the cell
```

Writable Smart Insert 38:17:1155

26°C Partly cloudy 7:04 PM 1/18/2024

12. Reading Data from an Excel Sheet.

eclipse-workspace - Phase2-TestNGScripts/src/test/java/AssistedProjects/ReadExcelData.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

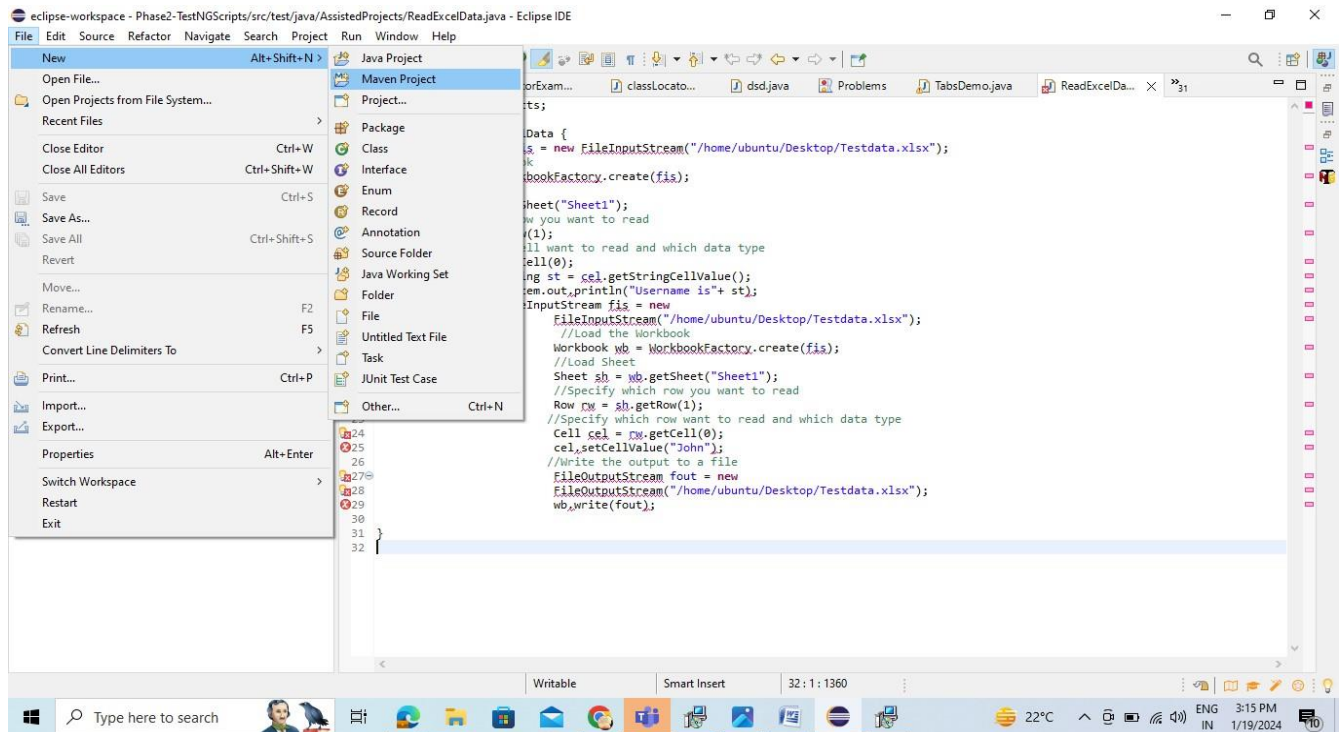
- Java_Programs
 - javasikuli
 - jdbc_projects
 - pllassisted_Jprograms
 - Phase2-TestNGScripts
 - src/main/java
 - src/main/resources
 - src/test/java
 - AssistedProjects
 - ReadExcelData.java
 - LessonEndProject
 - TestNGAnnotations
 - src/test/resources
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - test-output
 - pom.xml
 - testng.xml
 - selenium_project

```
1 package AssistedProjects;
2
3 public class ReadExcelData {
4     FileInputStream fis = new FileInputStream("/home/ubuntu/Desktop/Testdata.xlsx");
5     //Load the Workbook
6     Workbook wb = WorkbookFactory.create(fis);
7     //Load Sheet
8     Sheet sh = wb.getSheet("Sheet1");
9     //Specify which row you want to read
10    Row rw = sh.getRow(1);
11    //Specify which cell want to read and which data type
12    Cell cel = rw.getCell(0);
13    String st = cel.getStringCellValue();
14    System.out.println("Username is"+ st);
15
16    FileInputStream fis = new
17        FileInputStream("/home/ubuntu/Desktop/Testdata.xlsx");
18    //Load the Workbook
19    Workbook wb = WorkbookFactory.create(fis);
20    //Load Sheet
21    Sheet sh = wb.getSheet("Sheet1");
22    //Specify which row you want to read
23    Row rw = sh.getRow(1);
24    //Specify which row want to read and which data type
25    Cell cel = rw.getCell(0);
26    cel.setCellValue("John");
27    //Write the output to a file
28    FileOutputStream fout = new
29        FileOutputStream("/home/ubuntu/Desktop/Testdata.xlsx");
30    wb.write(fout);
31 }
32
```

Writable Smart Insert 32:1:1360

SEN... 3:09 PM 1/19/2024

13. Maven Configuration



New Maven Project

New Maven project

Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

Properties available from archetype:

Name	Value

► **Advanced**

New Maven Project

← → ↺ <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java/3.14.0>

Popular Categories

[Aspect Oriented](#)
[Actor Frameworks](#)
[Application Metrics](#)
[Build Tools](#)
[Bytecode Libraries](#)
[Command Line Parsers](#)
[Cache Implementations](#)
[Cloud Computing](#)
[Code Analyzers](#)
[Collections](#)
[Configuration Libraries](#)
[Core Utilities](#)
[Date and Time Utilities](#)
[Dependency Injection](#)
[Embedded SQL Databases](#)
[HTML Parsers](#)

HomePage	http://www.seleniumhq.org/
Date	(Aug 02, 2018)
Files	pom (3 KB) jar (293 bytes) View All
Repositories	Central Spring Plugins
Used By	1,078 artifacts

Note: There is a new version for this artifact

New Version	4.0.0-alpha-2
--------------------	---------------

[Maven](#) [Gradle](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#) [Buildr](#)

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.14.0</version>
</dependency>
```

☒ Include comment with link to declaration

Google Chrome Version 74.0.3729.108

eclipse-workspace - selenium_project/pom.xml - Eclipse IDE

File Edit Source Navigate Search Project Run Window Help

Package Explorer X

- Java_Programs
- javasikuli
- jdbc_projects
- plassisted_jprograms
- Phase2-TestNGScripts
 - src/main/java
 - src/main/resources
 - src/test/java
 - AssistedProjects
 - ReadExcelData.java
 - LessonEndProject
 - TestNGAnnotations
 - src/test/resources
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - test-output
 - pom.xml
 - testng.xml
- selenium_project
 - src/main/java
 - src/main/resources
 - src/test/java
 - src/test/resources
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - Screenshot
 - src
 - target
 - pom.xml

SetUpCheck.java LocatorExam... classLocato... dsd.java TabsDemo.java ReadExcelDa... selenium_pr... X 32

```
https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
10<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://m
2  <modelVersion>4.0.0</modelVersion>
3  <groupId>selenium_project</groupId>
4  <artifactId>selenium_project</artifactId>
5  <version>0.0.1-SNAPSHOT</version>
6  <dependencies>
7    <dependency>
8      <groupId>org.seleniumhq.selenium</groupId>
9      <artifactId>selenium-java</artifactId>
10     <version>4.11.0</version>
11   </dependency>
12   <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
13   <dependency>
14     <groupId>io.github.bonigarcia</groupId>
15     <artifactId>webdrivermanager</artifactId>
16     <version>5.4.1</version>
17   </dependency>
18 </dependencies>
19
20
21
22
23</project>
```

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Writable Insert 19:17 [417]

Type here to search 22°C 3:23 PM 1/19/2024

15. Listeners - Context, iTest, iTestResults.

eclipse-workspace - selenium_project/src/test/java/Assisted_project/ListenersTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

Java_Programs

- JRE System Library [JavaSE-17]
- src
- javasikuli
- jdb_projects
- p1assisted_programs
 - JRE System Library [JavaSE-17]
 - src
 - assisted_projects
 - PRACTICE_PROJECT
 - module-info.java
- Phase2-TestNGScripts
- selenium_project
 - src/main/java
 - src/main/resources
 - src/test/java
 - Assisted_project
 - Autolt.java
 - DatePicker.java
 - ListenersTest.java
 - PhaseEndProject.java
 - SeleniumHJDBC.java
 - TestDatabaseWithSelenium.java
 - TestNIMobilePage.java
 - TestSearchPage.java
 - WebElements.java
 - selenium_Scripts
 - src/test/resources
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - Screenshot
 - src
 - target

```
1 package Assisted_project;
2 import org.testng.ITestContext;
3 import org.testng.ITestListener;
4 import org.testng.ITestResult;
5
6 public class ListenersTest implements ITestListener {
7
8     public void onFinish(ITestContext Result) {
9         System.out.println(Result.getName()+"case finished");
10    }
11
12    public void onStart(ITestContext Result) {
13        // TODO Auto-generated method stub
14    }
15
16    public void onTestFailedButWithinSuccessPercentage(ITestResult Result) {
17        // TODO Auto-generated method stub
18    }
19
20    public void onTestFailure(ITestResult Result) {
21        // TODO Auto-generated method stub
22        System.out.println("The name of the testcase failed is :"+Result.getName());
23    }
24
25    public void onTestSkipped(ITestResult Result) {
26        // TODO Auto-generated method stub
27        System.out.println("The name of the testcase Skipped is :"+Result.getName());
28    }
29
30    public void onTestStart(ITestResult Result) {
31        // TODO Auto-generated method stub
32        System.out.println(Result.getName()+" test case started");
33    }
34
35    public void onTestSuccess(ITestResult Result) {
36        // TODO Auto-generated method stub
37    }
38
39 }
```

Writable Smart Insert 1:27:26

Type here to search

21°C ENG IN 3:44 PM 1/19/2024

16. Install Artifactory

Dashboard > Manage Jenkins > Plugins

Updates

44

Download progress

Available plugins

Installed plugins

Advanced settings

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Artifactory

⚠ Downloaded Successfully. Will be activated during the next boot

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running

> Plugins

44

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Artifactory

⚠ Downloaded Successfully. Will be activated during the next boot

Restarting Jenkins

⋮ Running

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→ ☒ Restart Jenkins when installation is complete and no jobs are running

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/Sonal0409/MavenBuild-SL.git

Credentials ?

- none -

+ Add

Advanced

Add Repository

Dashboard > LessonEndProject > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

Pipeline

Definition

Pipeline script

Pipeline script


Pipeline script from SCM

1





try sample Pipeline...

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

**Jenkins**

Search (CTRL+K)

adm

Dashboard > Lessonenproject1 > #1 > Allocate node: Start > Workspace > Workspace

↑ Up

🔍 Status

📄 Console Output

📁 Workspace

Workspace

/ target /

classes/com/geekcap/vmturbo

generated-sources/annotations

generated-test-sources/test-annotations

java-example

maven-archiver

maven-status/maven-compiler-plugin

surefire-reports


test-classes/com/geekcap/vmturbo


java-example.war


19-Jan-2024, 5:09:31 AM 5.13 KB


(all files in zip)


Dashboard > LessonEndprojectArtifactory >


 Status


 Changes


 Workspace


 Build Now

 Configure

 Delete Project

 Artifactory Build Info


 Rename

 **Build History**



trend ▾


🔍 Filter builds...


/

 #3

19-Jan-2024, 5:23 AM



 **LessonEndprojectArtifactory**

 Artifactory Build Info


Permalinks

- Last build (#3), 3 min 27 sec ago
- Last stable build (#3), 3 min 27 sec ago
- Last successful build (#3), 3 min 27 sec ago
- Last failed build (#2), 5 min 0 sec ago
- Last unsuccessful build (#2), 5 min 0 sec ago
- Last completed build (#3), 3 min 27 sec ago

18. Build and Configure CI/CD Pipeline with Selenium WebDriver.

Jenkins 2.414.2 Setup

Service Logon Credentials




Enter service credentials for the service.

Jenkins 2.414.2 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.414.2 to run successfully.

Logon Type:
☒ Run service as LocalSystem (not recommended)
☐ Run service as local or domain user:
Account:
Password:

Jenkins 2.414.2 Setup


Port Selection



Choose a port for the service.

Please choose a port.

Port Number (1-65535):

  Click 'Test Port' button to proceed

It is recommended that you accept the selected default port.

19. Selenium with Jenkins.

