**Framework from Scratch:**

Step1: Initial setup

Step2: Execution

Step3: Management

**Application under test:**

URL: <https://www.saucedemo.com/>

Username: standard\_user

Password: secret\_sauce

**Step1: Initial Setup:**

**1.Create Maven Project in Eclipse**

**Project name: TDDFramework**

**Maven:** Maven is an automation building tool used primarily for Java Projects. Whenever we create a Maven Project, we are creating a local repository, which is taking all the external jar files from the Maven remote Repository. So, if we need any jar file, we need to call it in the local repository. You don’t need to download it in your local system. Pom.xml file is automatically created when we create maven project. We can specify what jar file we need in the pom.xml file in the form of dependencies.

With Maven Project we can use command line interface.

When using java project, we need to download all the jars and then we have to configure selenium and all other jar files needed for the framework to the java project. Here the project will be depended on our System and it results in wastage of hard disk.

**2.Dependencies initially required**

1. Selenium - Selenium Java » 4.14.1

2. TestNG- TestNG » 7.7.1

3. Apache Poi- Apache POI Common » 5.2.3

4. Apache poi-ooxml

5. Apache poi-scratchpad

6. Apache poi excelant

7. Extent Reports

8. Common IOS

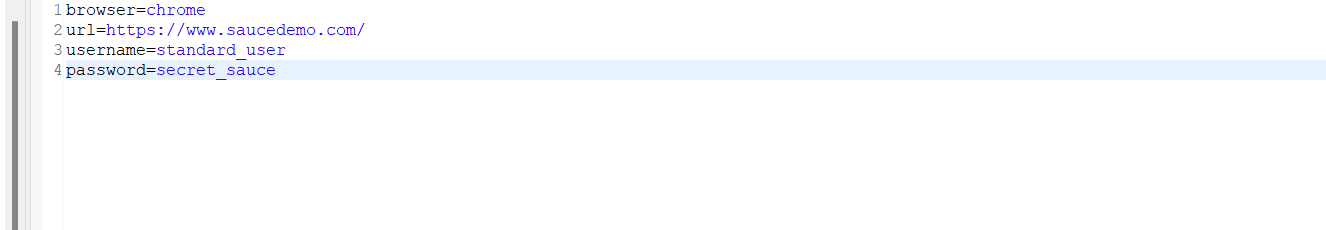
9. Log4j

**3.Create Base Package in Src/test/java**

* Package name: **basePackage**
* Class name: **BaseHRMClass**
* **All the common commands are kept in the base Package**
* **All the data which are subjected to change should be kept in Configuration files (**browser, URL, username, password)
* If we keep these data in the base class, we have to keep on changing the values, that will affect the base class.
* If base class is affected, your complete framework will fail. (Pom layer and Test Layer will fail)

**4.Create another package under src/test/java package**

* Package name: **environment variables**
* Create file under environment variables: **Config.properties**
* In Config.properties add data which are subjected to change like browser, URL, username and password.
* Base class has to read data from Config.properties file
* Properties class is a class, whose methods we can use to read config. properties file.



**5.Add Chromedriver.exe and geckodriver.exe to the project**

[**https://github.com/mozilla/geckodriver/releases**](https://github.com/mozilla/geckodriver/releases)

**6. Create another package in src/test java**

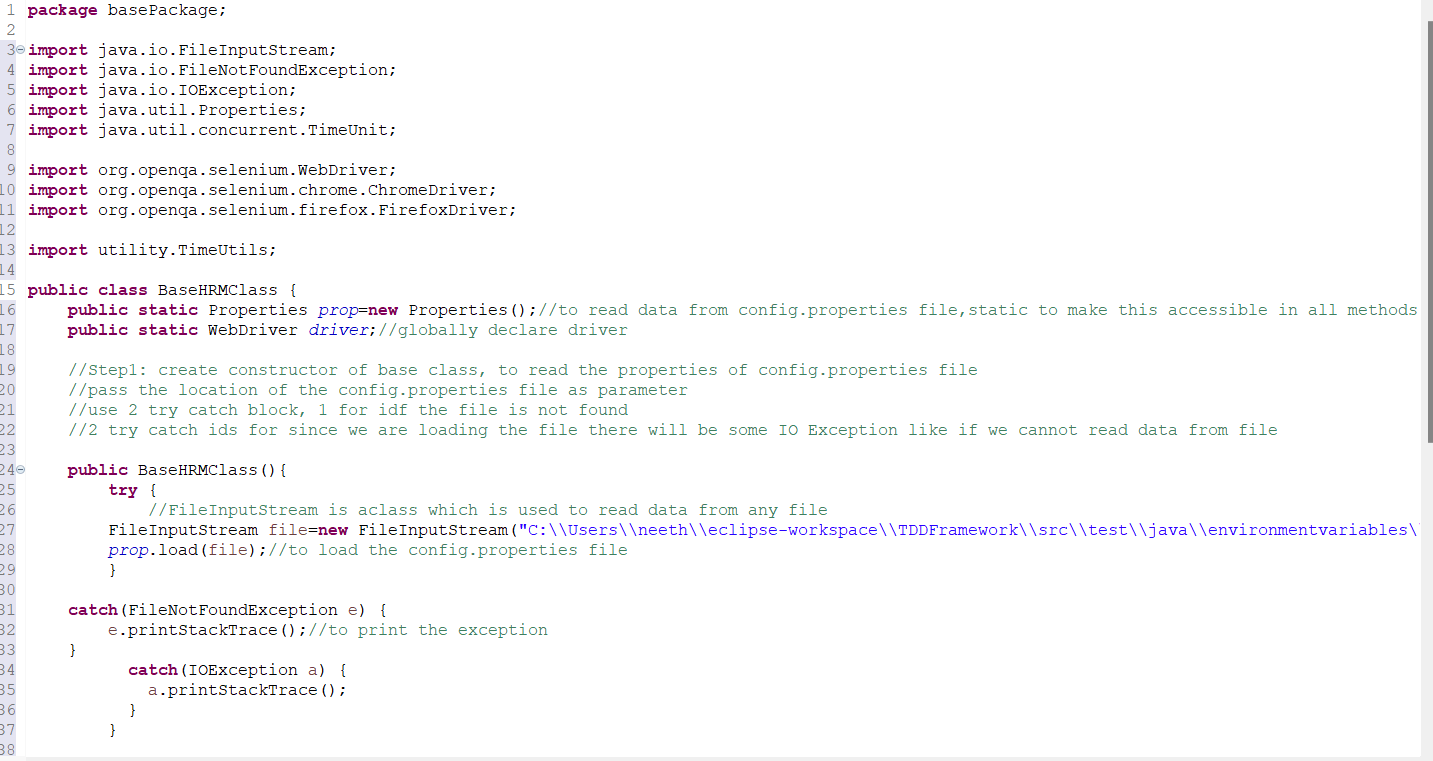
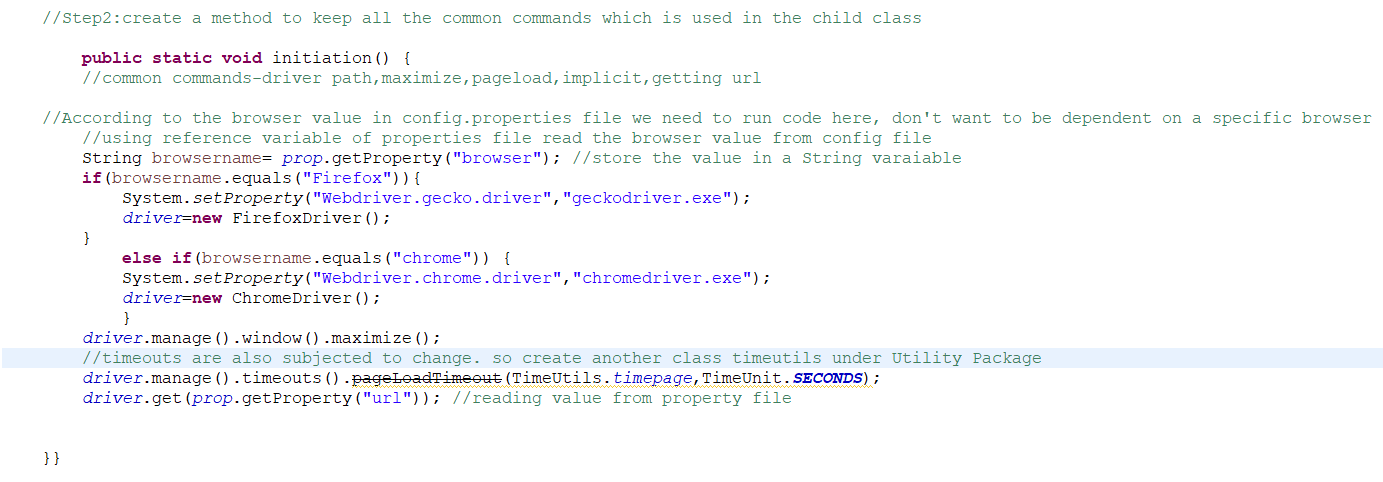
Package name: **Utility**

Class name: **TimeUtils**

Page Load Timeout is subjected to change. So, add the timeout value in a variable and call it in the Base class.



**BaseHRMClass:**



**7.Create POM class all the pages of the application**

Package: pompackage

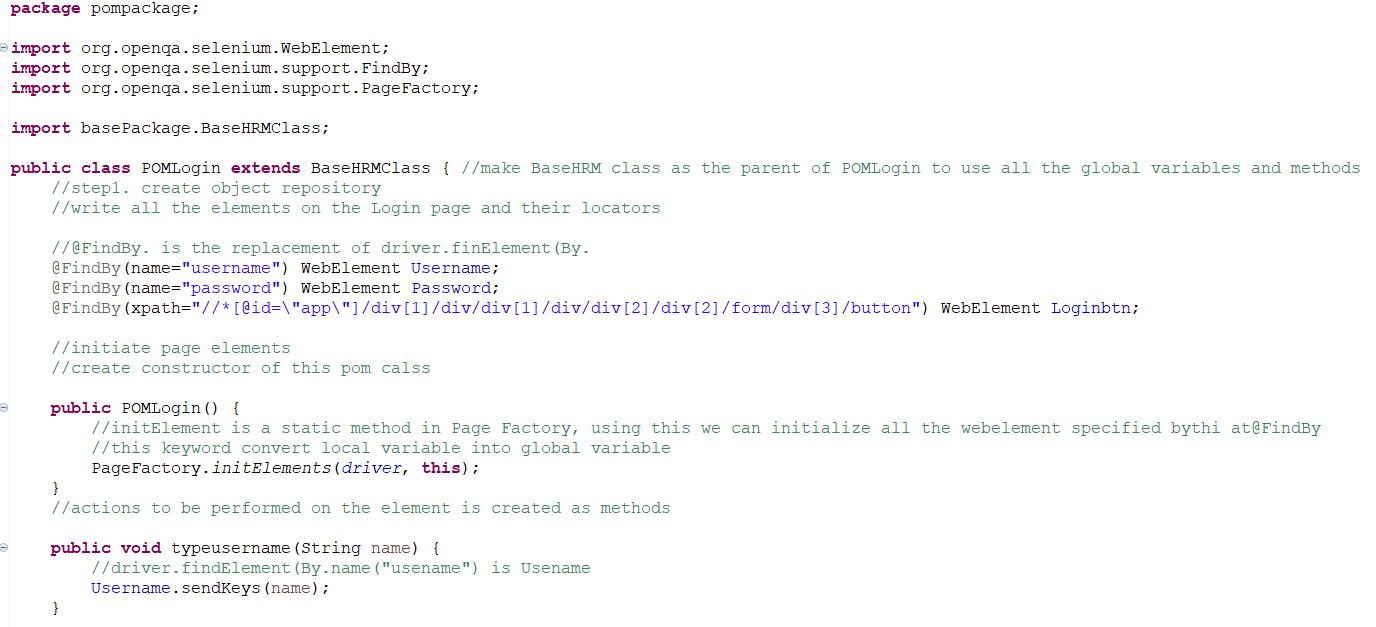
Class: POMLogin-for login Page

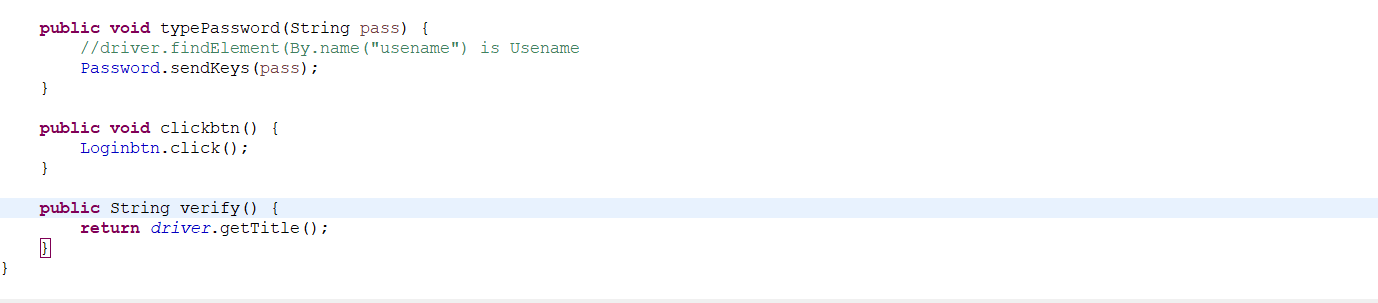
All the elements of the login page is stored in POM class

Element locators value is stored in a variable

All the action to be performed on elements are made as methods

**POMLogin:**



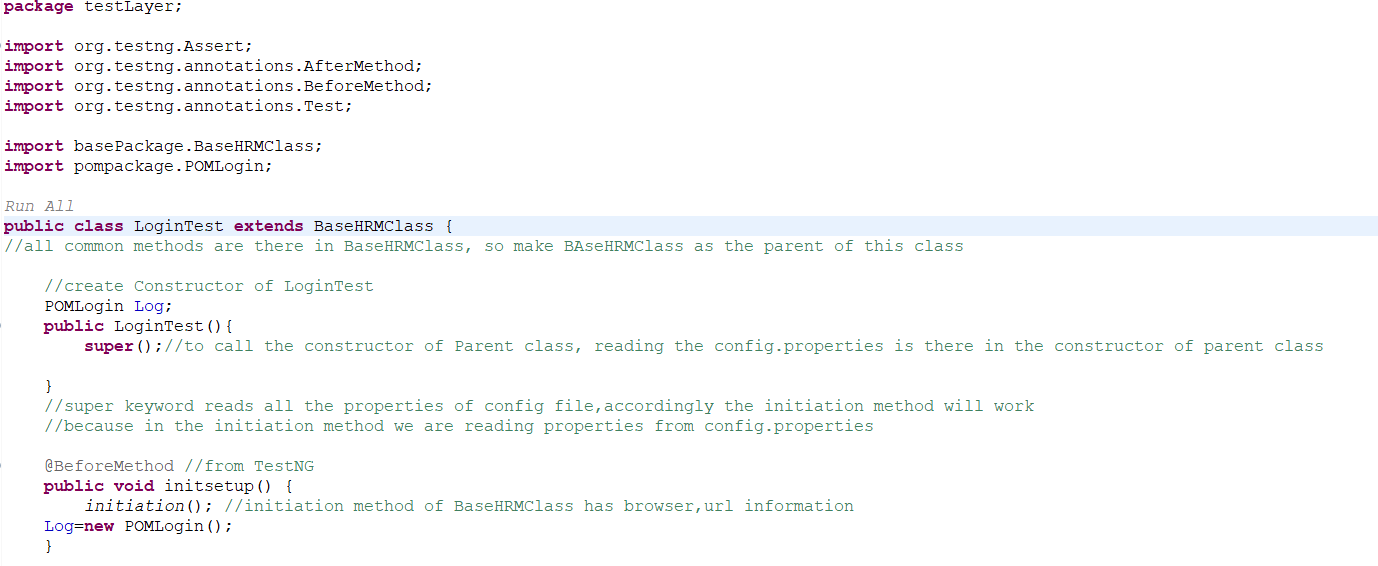
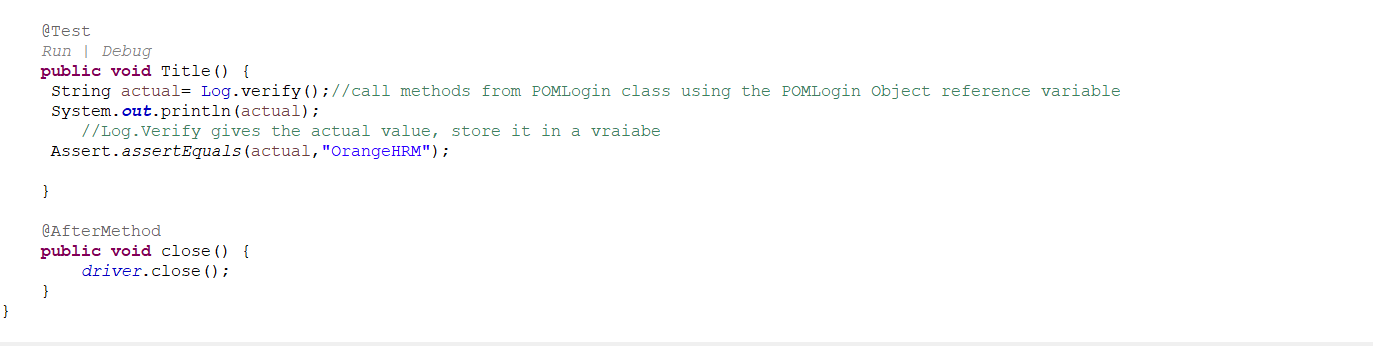


8. **Create package for Test Layer**

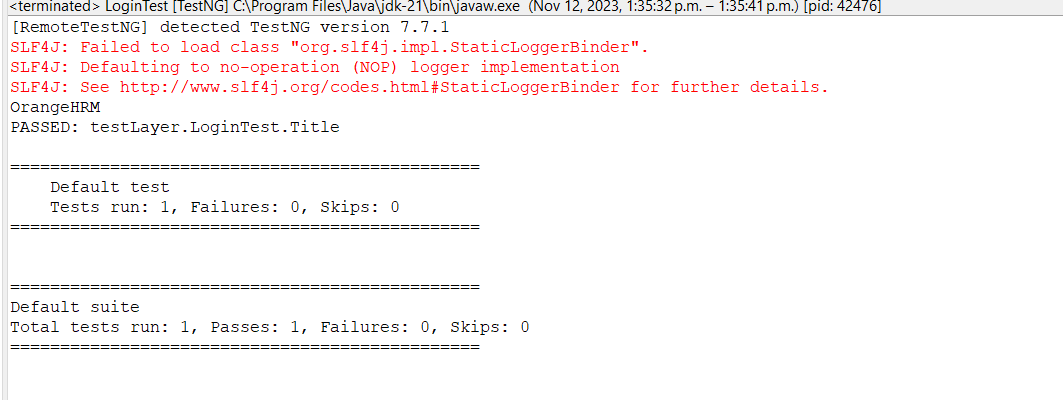
Package name: test Layer

Class name: LoginTest

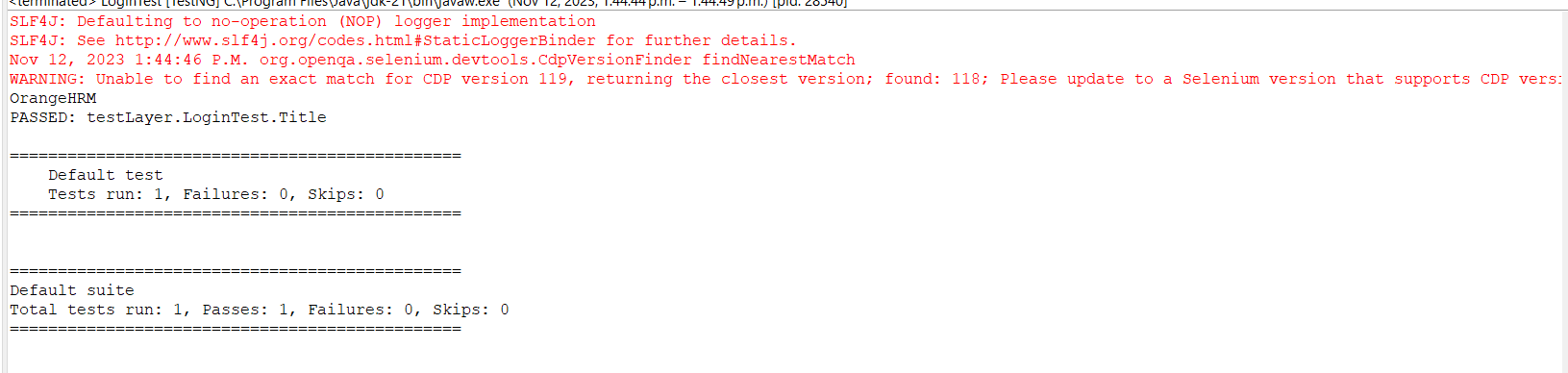
**LoginTest**



**Output in Firefox browser**

****

**Output in Chrome Browser**

****

**How to make Framework Data Driven Framework:**

To pass multiple set of values to the test case.

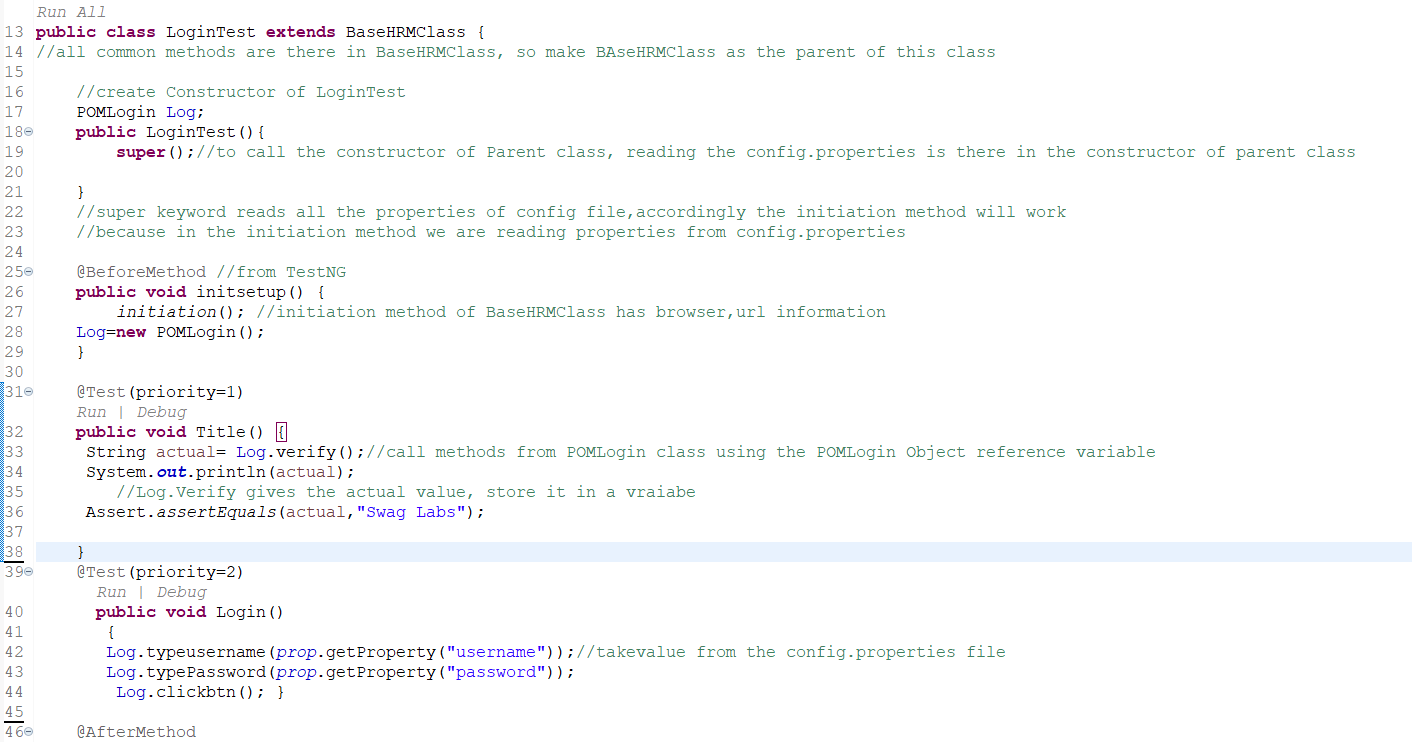
We can pass data through Data Providers and also through external excel sheets.

Add Apache jar dependencies to the pom.xml

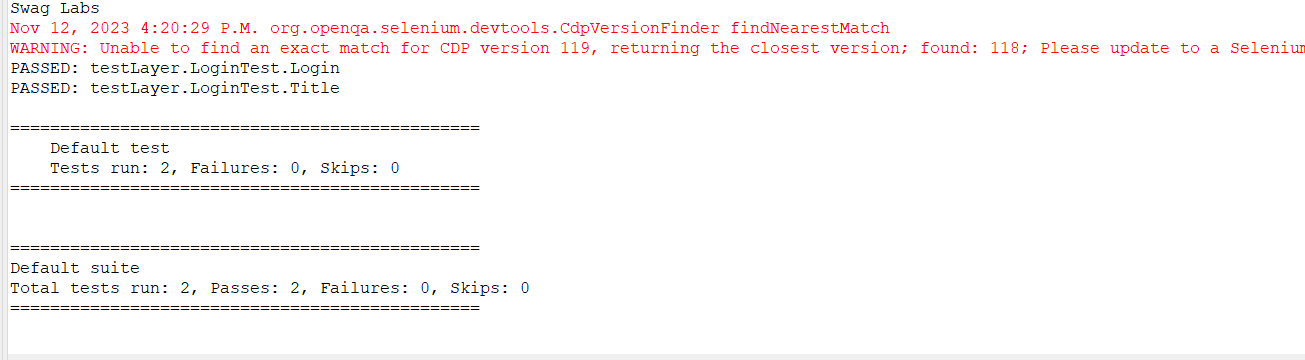
Add Apache poi, Apache poi-ooxml, Apache poi-scratchpad, Apache poi-excelant

**1. Write test case to pass multiple set of data**

In Login Test, create testcases and take value for username and password from config. properties file



Output:

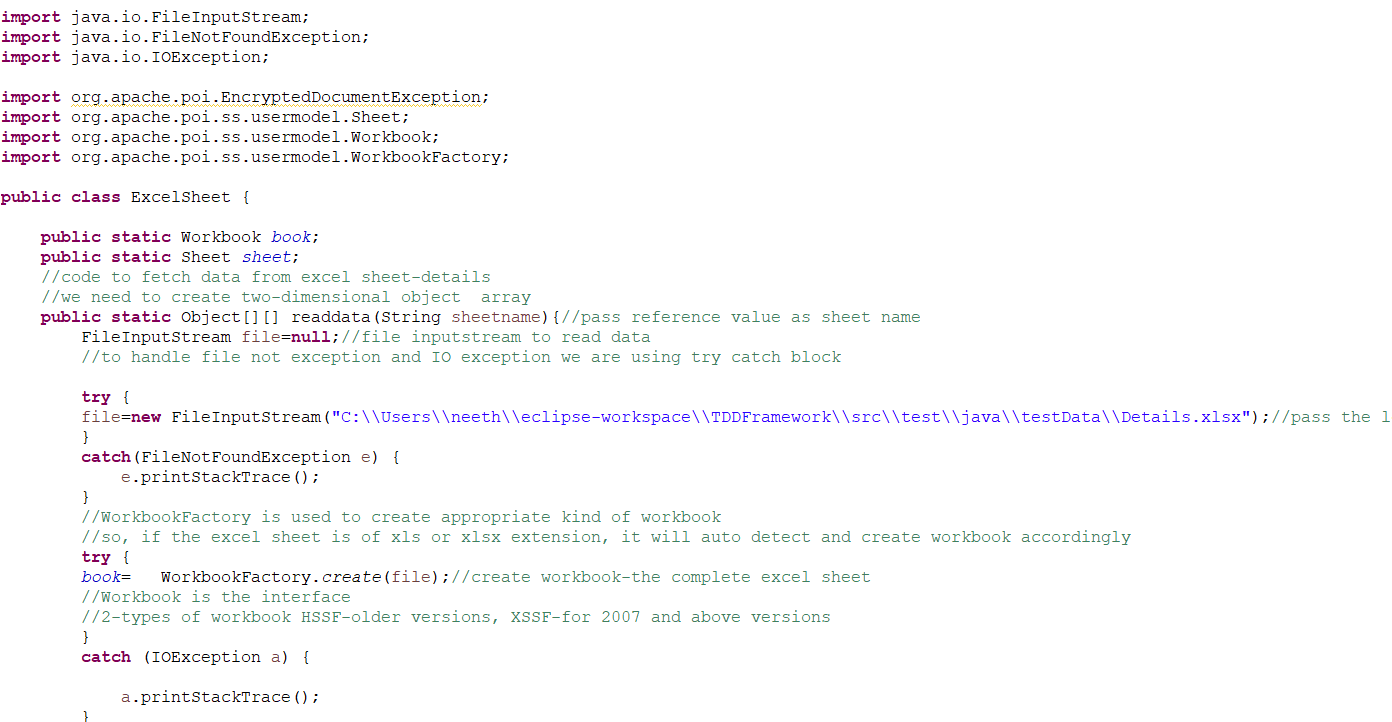


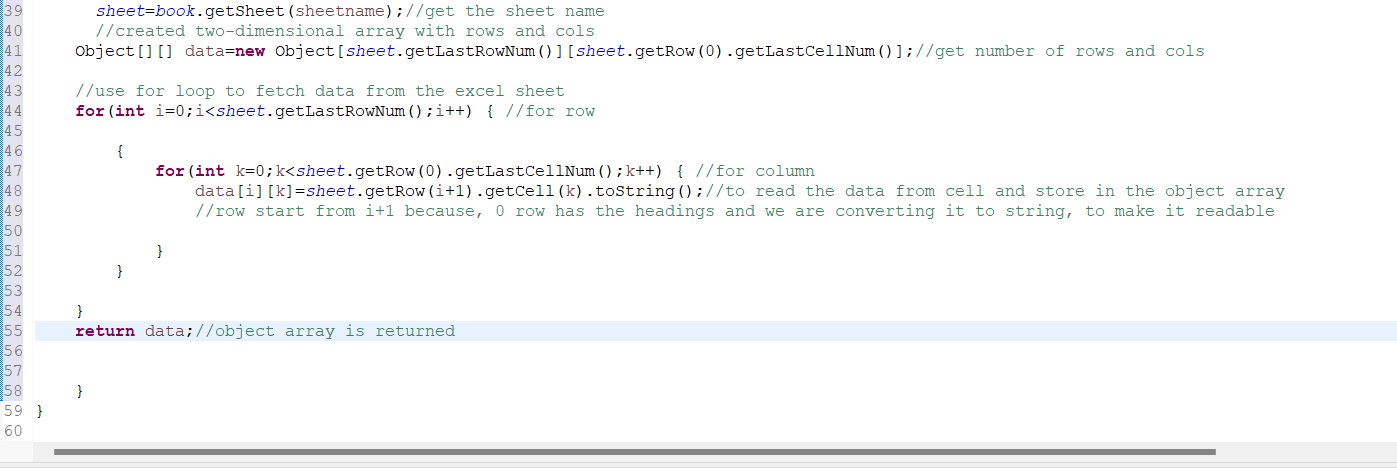
Create a Package**: testData**

Create a class**: ExcelSheet**

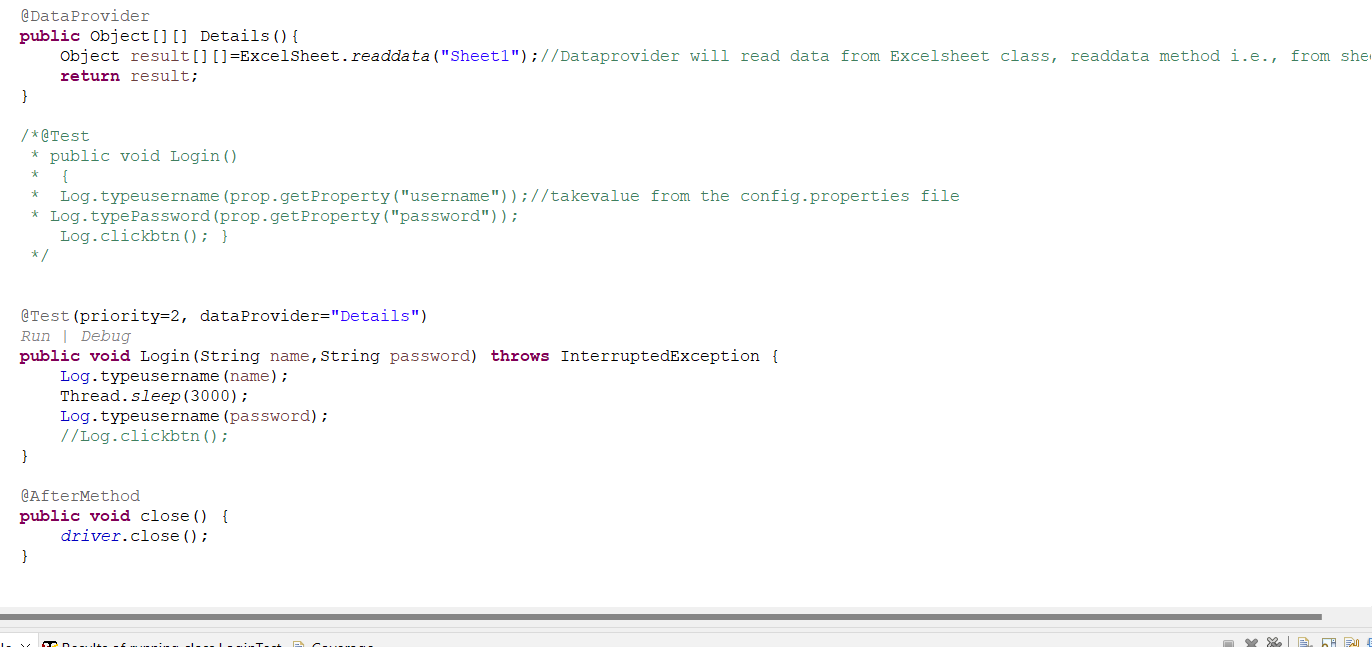
* Go the location of the testData and create a excel sheet and name it as details
* Pass multiple set of values for login.

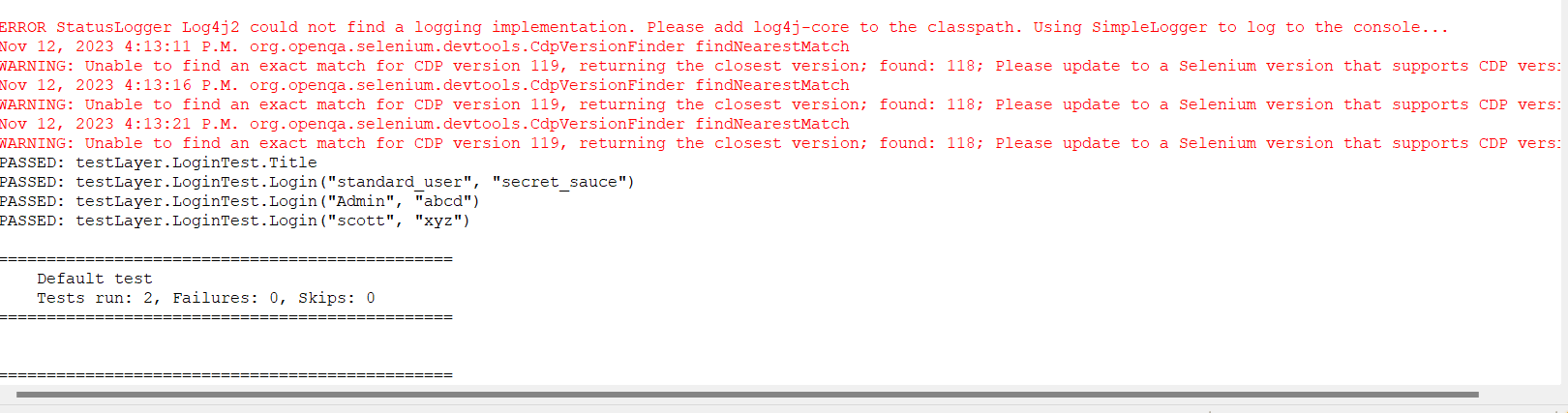
In the ExcelSheet class, write code to fetch the data from the Excel sheet as a 2-dimensional Object array.

**ExcelSheet.java**

****

**2. In the LoginTest.java, we create @DataProvider**

****

**Output:**

**Screenshot of Execution:**

**Create a package: screenshots**

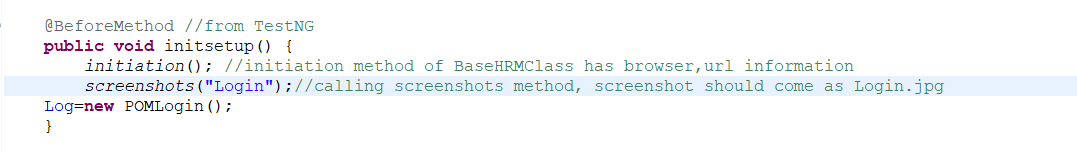
**Create a Folder inside the package: Screenshots**

* In the base class, we need to write a method to take screen shot
* Add Apache Common IO dependency to Pom.xml

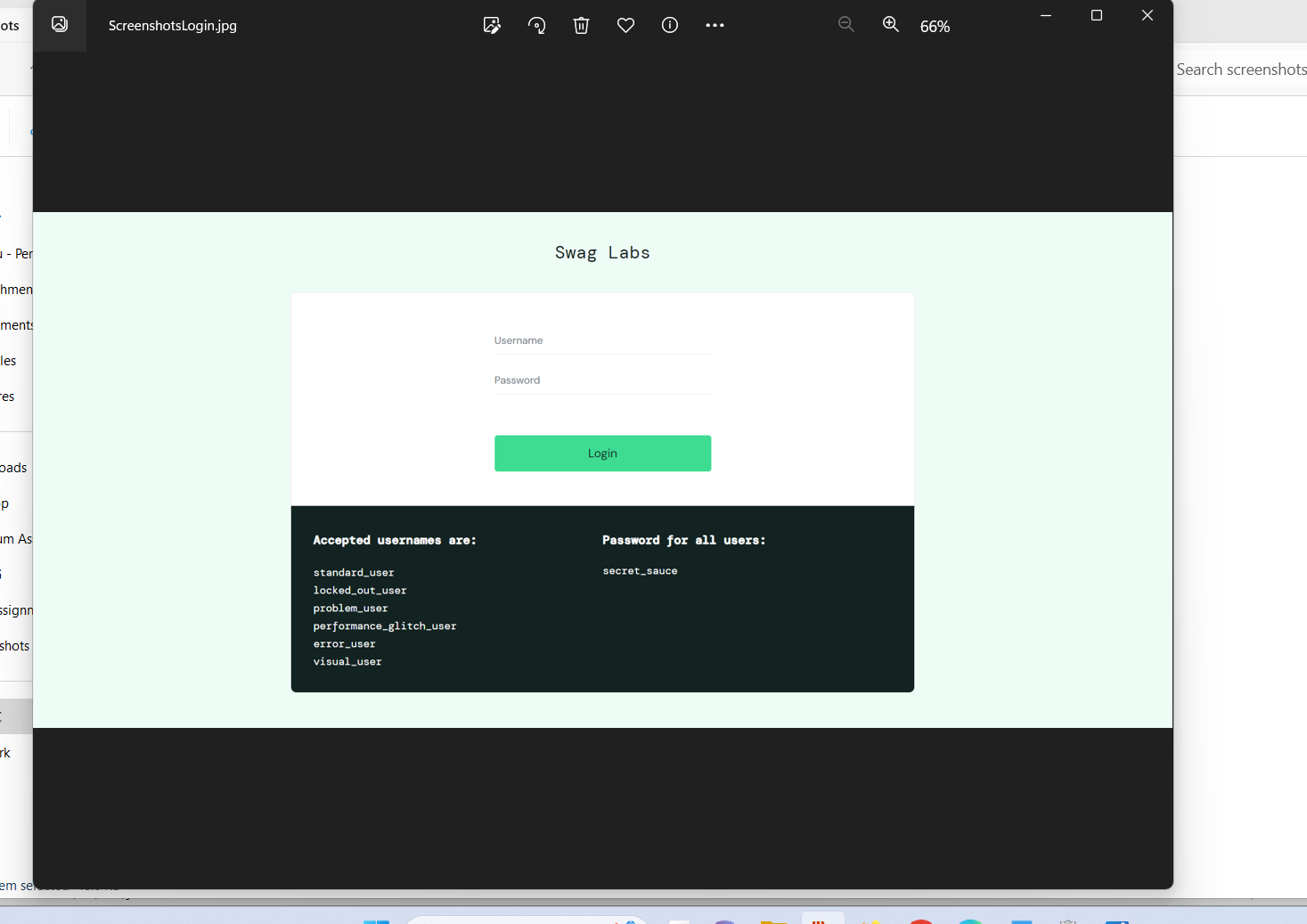
**BaseHRMClass**



**In the LoginTest use the screenshot method**

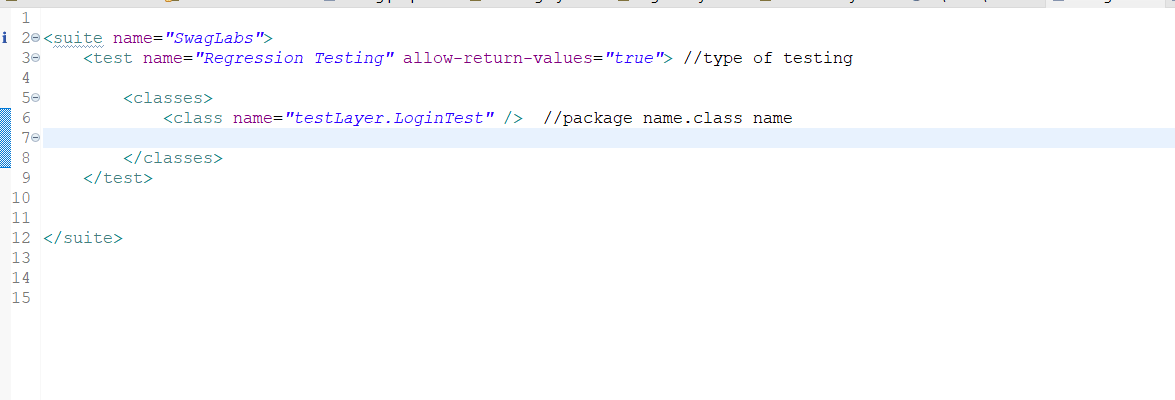
****

**Output:**

****

**Batch Execution**

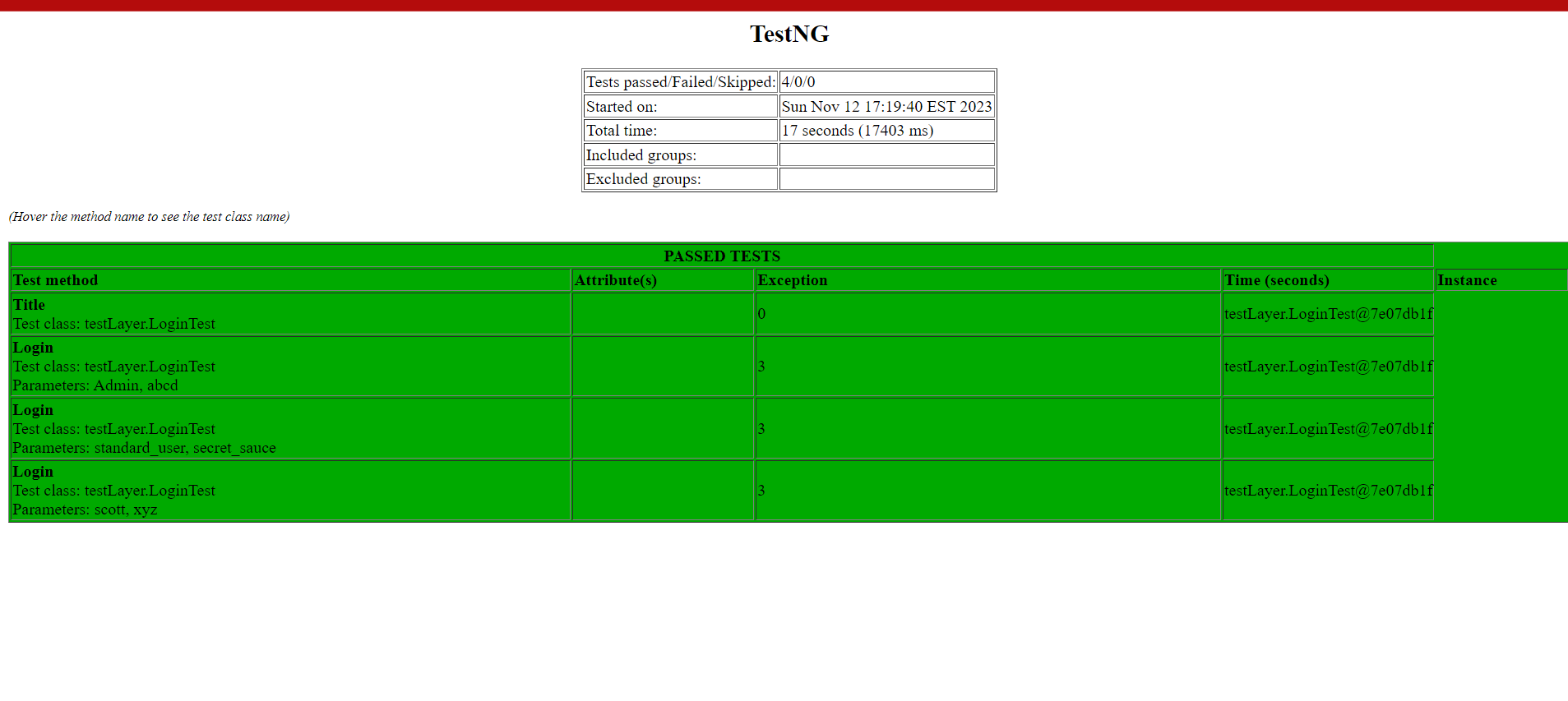
**Create a file-Testng.xml and Run the file as TestNG.Suite**

****

**Output:**

In the test-Output folder, Open Swag Labs🡪Under Regression Testing open Testng.Html file in web

Browser.

****

**GIT: Version Control tool**

If we are working in a team, and we are working on one of the project, so If I made some changes, and these changes are in my local system. Other team member will not get these changes, until you share it on the centralized repository.

If you want to share the code in the centralized location and make the code consistent, we use version control tool called GIT.

GIT is easy to collaborate with different tools such as Eclipse IDE. Also, we can know which user made the change and at what time. If you want to undo any change, we can undo the change also.

Git Hub is a website where you can make repository and GIT is the tool with which we create the repository.

Two actions performed by GIT:

1.PUSH

2.PULL

**PUSH** means send your project from local system to your repository

**PULL** means you can take project from repository and save it in local system

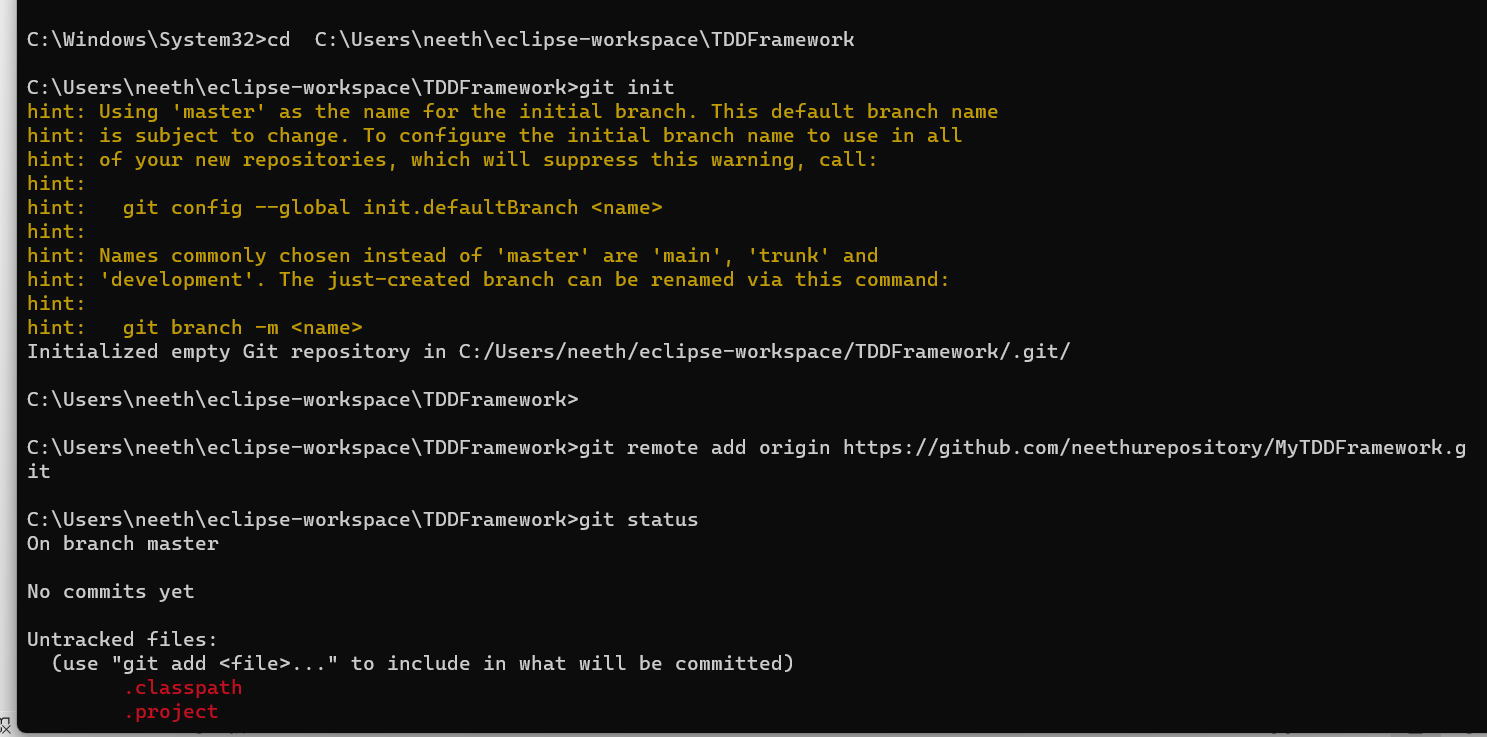
Open Github.com and sign in to GitHub

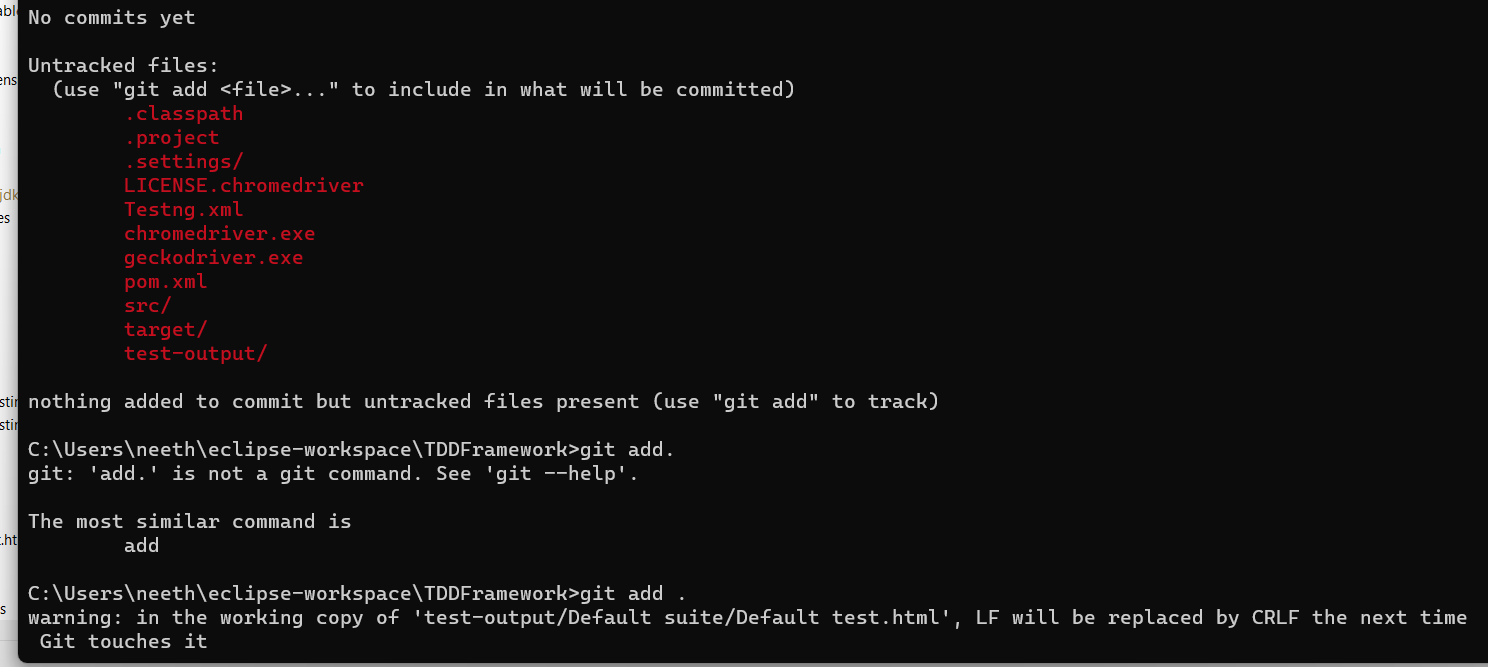
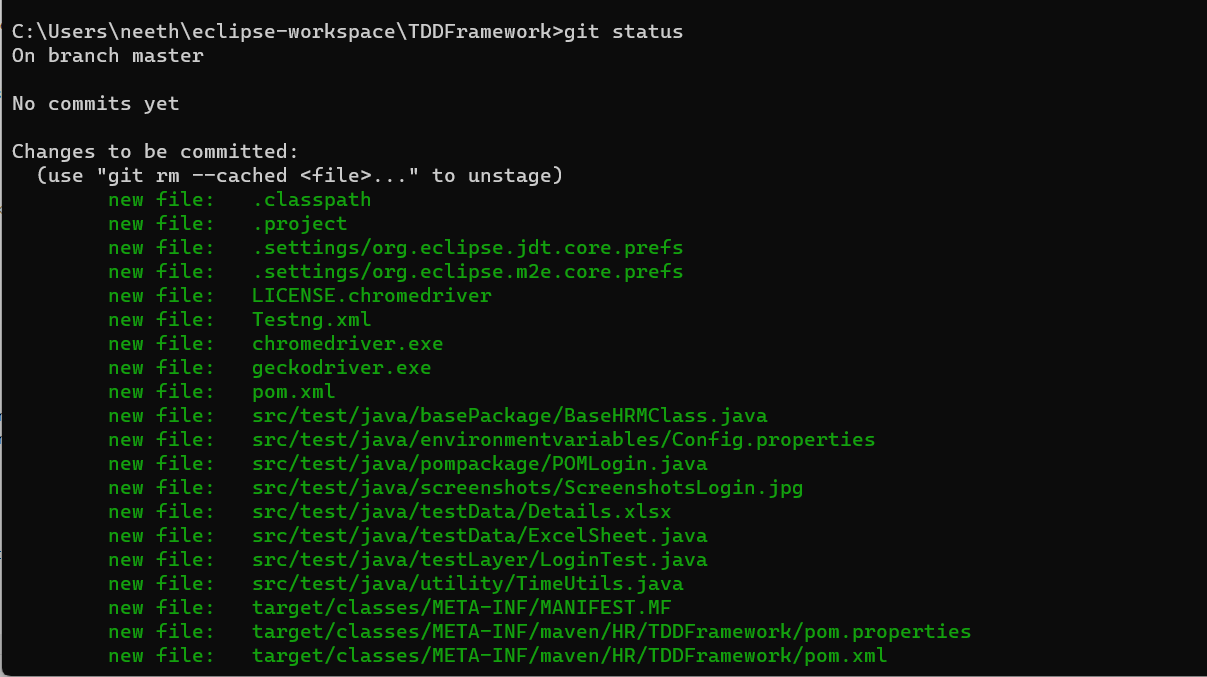
Download Git for Windows

Create a new repository-MyTDDFramework

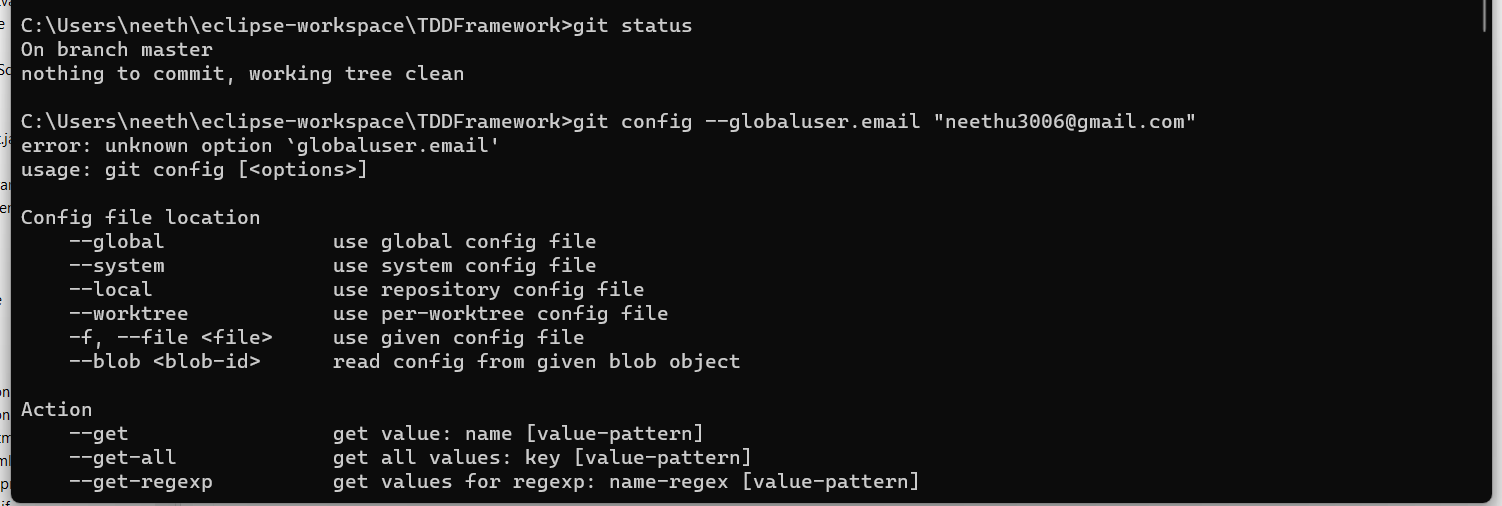
Open Git Cmd🡪

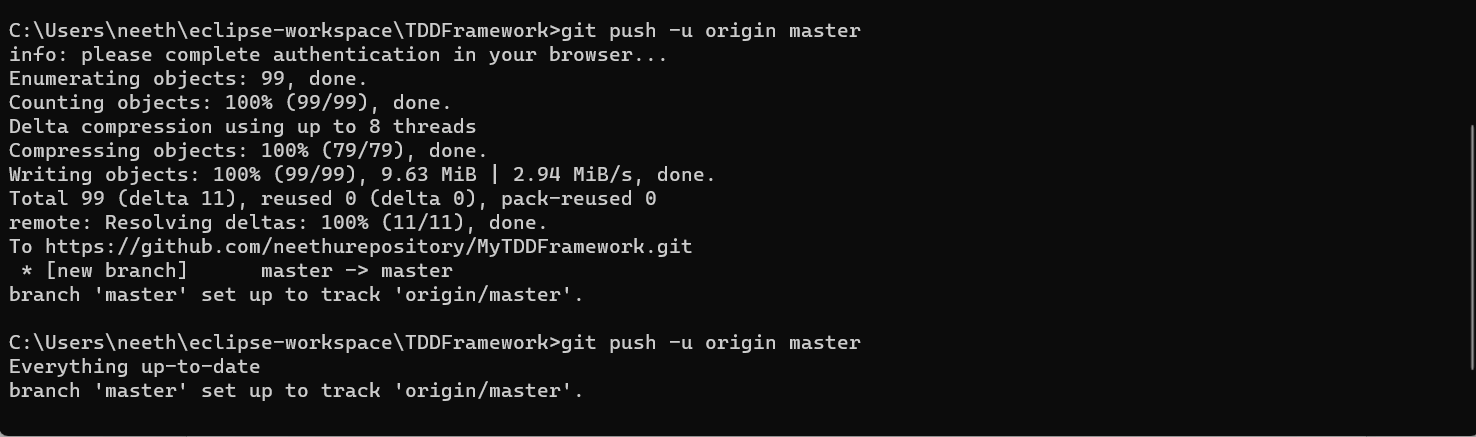
In the command prompt



To commit use command: git commit -m “first commit”





Authenticate the account.

Project is uploaded in the Git Hub

