Selenium Testing FrameWork

Table of Content

DESIGN- HYBRID FRAMEWORK

Framework is the well-organized structure of Automation files which enables the coding standards, Maintainability and Reusability of resources.

IMPLEMENTATION

1.Page Objects: For creating individual functions for a module inside a page class

2. Utilities: For creating files which helps reading configured files

3. Test Data: For creating an Excel file where all the test data can be kept and used in Test Cases

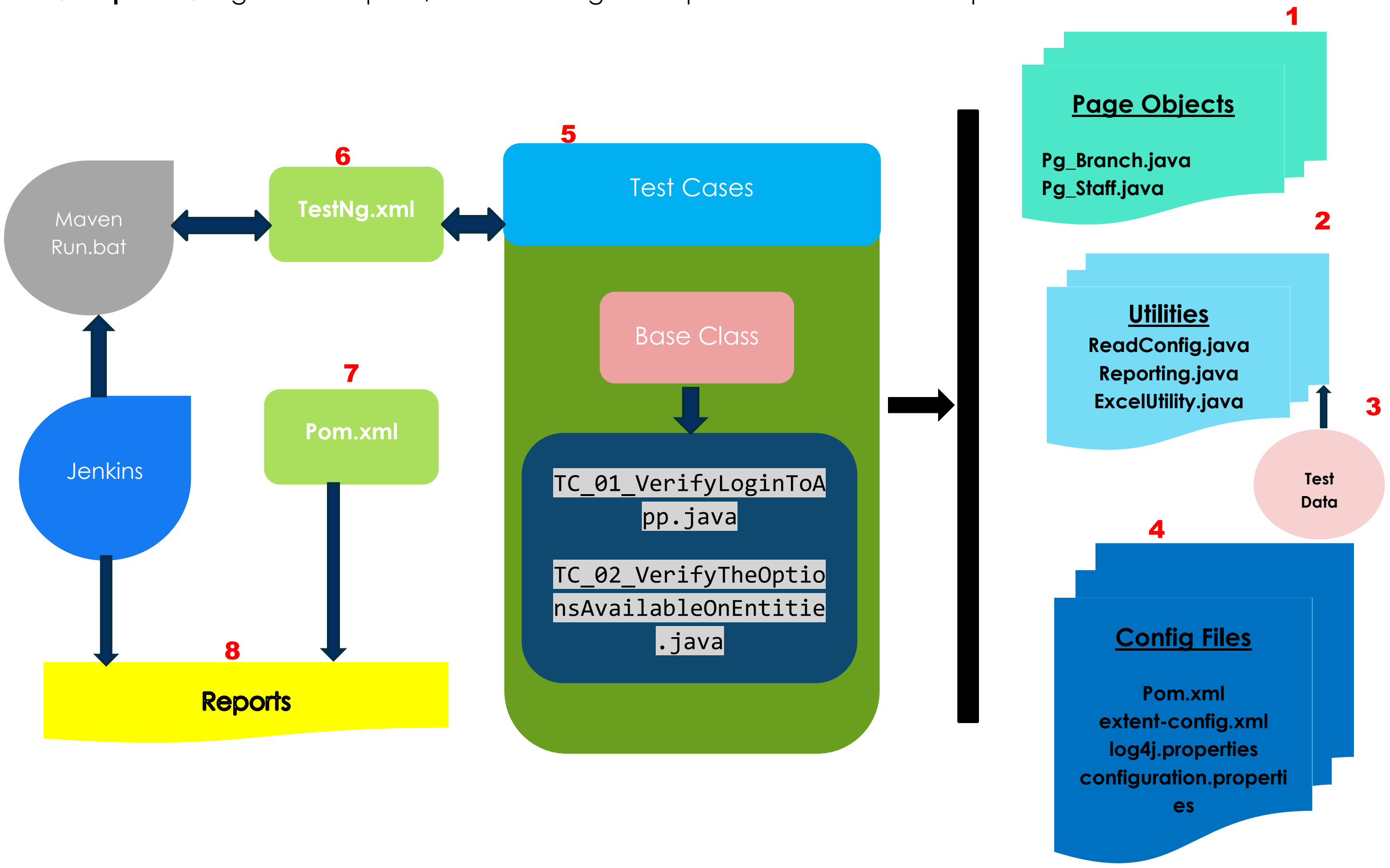
4. Config Files: Files for importing the required dependencies and having properties stored.

5. Test Cases: To call page objects in an order to follow the steps in Test Cases

6. TestNg.xml: To run multiple test cases, provide options to run test in multi browsers

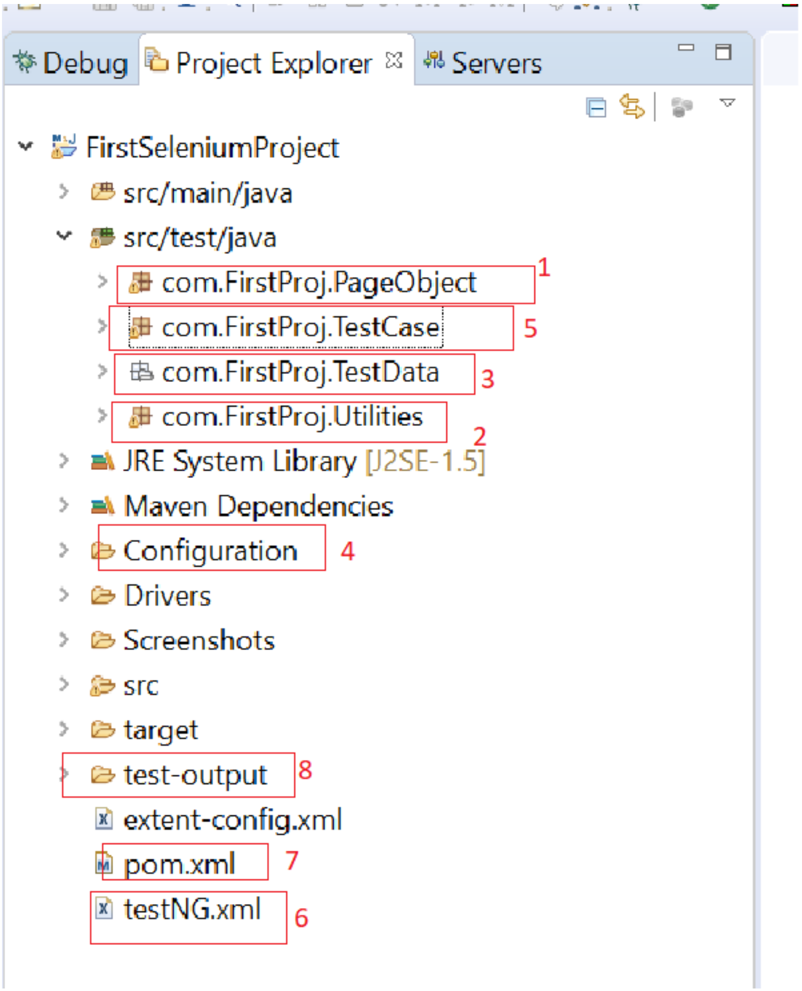
7. pom.xml: Contains all dependencies, which automatically imports relevant jars to function framework

8. Reports: Using Extent Reports, overall testing status print in a form of html reports



Below screenshot from IDE, depicts how we have implemented different part of framework in our

Project: Gurukula



```
FirstSeleniumProject
  src/main/java
  src/test/java
    com.FirstProj.PageObject
    com.FirstProj.TestCase
    com.FirstProj.TestData
    com.FirstProj.Utilities
  JRE System Library [J2SE-1.5]
  Maven Dependencies
  Configuration
  Drivers
  Screenshots
  src
  target
  test-output
  extent-config.xml
  pom.xml
  testNG.xml
```

Step1- Select any IDE(selected Eclipse IDE for writing scripts here), create a Maven Project

Step2 -Create folder structure for Page Object, Test Cases, Utility and Test data inside src/test/java

Step3- Create a page in Page Object folder, which is a simple java class file with constructor as we have to call this page several times in test cases.

Step4- Identify the page element by using locators e.g id, name, class or xpath, create an object for all the relevant elements in WebElement or By objects

Step5- Create methods inside Page class to perform desired actions using locator objects

Step6. Follow the business flow and create methods accordingly

Step7- Create a BaseClass in Test case folder, use TestNg annotations e.g. @BeforeClass, @AfterClass etc.

Step8- Create the Test Case classes inside Test class folder which extends BaseClass

Step9- Create configuration.properties file to read the common data and path

Step10-Create Utility Classes to read the supporting properties and files

Step11-Create Test Data excel to store all related test data's inside Test Data folder

Step12-Create testing.xml file to run the test cases in parallel and to run with multiple browser

FRAMEWORK FEATURES

1. This framework uses an external Test data excel file to provide test data into test cases by using POI and Data provider functionality by TestNG
2. This framework logs the messages in log file to keep a record and logs of running status
3. Using Extent report to display the Pass, Fail and Skip status in tabular format
4. Attaches failure screenshot to the extent reports for better understanding on failed scenarios and defects
5. Using Page Object model to provide good maintainability and reusability

HOW TO USE

Download the framework from git repository https://github.com/swatiGit1430/FirstRepoSwa

Launch the Gurukula Application(As per the steps mentioned in mail)

Run the complete test suite with testing.xml file