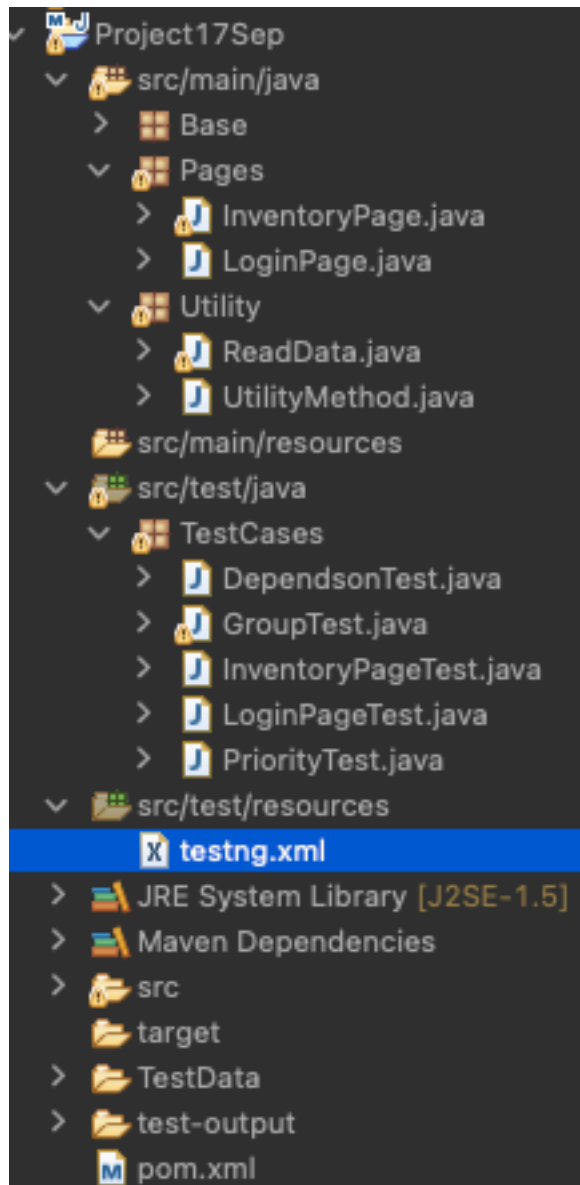


## Project Architecture



## Base Package

### TestBase Class

```
package Base;
```

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

```
import io.github.bonigarcia.wdm.WebDriverManager;
```

```
public class TestBase {
```

```

        public static WebDriver driver; //Global and Public

        public void initialization()
        {
            WebDriverManager.chromedriver().setup();
            driver = new ChromeDriver();
            driver.manage().window().maximize();
            driver.get(ReadData.readPropertyFile("url"));
//            driver.get("https://www.saucedemo.com/");    }
        }

```

## Pages Package

### LoginPage class

```

package Pages;

import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

import Base.TestBase;
import Utility.ReadData;

public class LoginPage extends TestBase {

    //Object Repository
    @FindBy(xpath = "//div[@class='login_logo']") private WebElement loginLogo;
    @FindBy(xpath = "//div[@class='bot_column']") private WebElement botLogo;
    @FindBy(xpath = "//input[@id='user-name']") private WebElement
usernameTextbox;
    @FindBy(xpath = "//input[@id='password']") private WebElement
passwordTextbox;
    @FindBy(xpath = "//input[@id='login-button']") private WebElement loginBtn;
    @FindBy(xpath = "//span[@class='title']") private WebElement productLabel;

//    driver.findElement(By.xpath(""));

    //Constructor to initialize the elements of page
    public LoginPage()
    {
        PageFactory.initElements(driver, this);
    }

    public String verifyTitle() throws Exception
    {
        Thread.sleep(2000);
        return driver.getTitle();
    }
}

```

```

        public String loginToApp() throws Exception
        {
            usernameTextbox.sendKeys(ReadData.readPropertyFile("username"));
//        usernameTextbox.sendKeys(ReadData.readExcelFile(0, 0));
            passwordTextbox.sendKeys(ReadData.readPropertyFile("password"));
            loginBtn.click();
            return driver.getCurrentUrl();
        }

        public String verifyLabelofInventory() throws Exception
        {
            loginToApp();
            return productLabel.getText();
        }
    }

//static method - use only static
//non static method - No restriction

```

## Inventory Page

```

package Pages;

import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.Select;

import Base.TestBase;
import Utility.UtilityMethod;

public class InventoryPage extends TestBase {

    @FindBy(xpath = "//div[@class='peek']") private WebElement peekLogo;
    @FindBy(xpath = "//img[@alt='Swag Bot Footer']") private WebElement
swagBot;
    @FindBy(id = "add-to-cart-sauce-labs-backpack") private WebElement backpack;
    @FindBy(id = "add-to-cart-sauce-labs-bike-light") private WebElement
bikeLight;
    @FindBy(id = "add-to-cart-sauce-labs-bolt-t-shirt") private WebElement
boltTshirt;
    @FindBy(id = "add-to-cart-sauce-labs-fleece-jacket") private WebElement
fleeceJacket;
    @FindBy(xpath = "//select[@class='product_sort_container']") private
WebElement sortProductDropdown;
    @FindBy(xpath = "//span[@class='shopping_cart_badge']") private WebElement
productCount;

```

```

public InventoryPage()
{
    PageFactory.initElements(driver, this);
}

public boolean verifyPeekLogo()
{
    return peekLogo.isDisplayed();
}

public boolean verifySwagBotLogo()
{
    return swagBot.isDisplayed();
}

public String addProduct() throws Exception
{
    UtilityMethod.selectClass(sortProductDropdown, "Name (Z to A)");
    Thread.sleep(3000);
    backpack.click();
    bikeLight.click();
    fleesJacket.click();
    Thread.sleep(3000);
    productCount.click();
    return productCount.getText();
}
}

```

## Utility Package

### ReadData Class

```

package Utility;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.Properties;

import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class ReadData {

    public static String readPropertyFile(String value) throws Exception
    {

```

```

        Properties p = new Properties();
        FileInputStream file = new FileInputStream("/Users/Zenith/eclipse-
workspace/Project17Sep/TestData/config.properties");
        p.load(file);
        return p.getProperty(value);
    }

    public static String readExcelFile(int row,int col) throws Exception
    {
        FileInputStream file = new FileInputStream("/Users/Zenith/eclipse-
workspace/Project17Sep/TestData/Data.xlsx");
        Sheet excel = WorkbookFactory.create(file).getSheet("Sheet1");
        String value = excel.getRow(row).getCell(col).getStringCellValue();
        return value;
    }
}

```

### UtilityMethod Class

```

package Utility;

import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.Select;

public class UtilityMethod {

    public static void selectClass(WebElement ele,String option)
    {
        Select s = new Select(ele);
        s.selectByVisibleText(option);
    }

    public void captureScreenshot()
    {

    }

}

```

## Testcases Package

### LoginPageTest Class

```

package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

```

```

import Base.TestBase;
import Pages.LoginPage;

public class LoginPageTest extends TestBase {

    LoginPage login;// = new LoginPage();

    @BeforeMethod
    public void setup() throws Exception
    {
        initialization();
        login = new LoginPage();
    }

    @Test (enabled = true)
    public void verifyTitleTest() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        Assert.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }

    @Test (enabled = true)
    public void verifyUrlTest1() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        Assert.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }

    @Test
    public void loginToAppTest() throws Exception
    {
        String expRes = "https://www.saucedemo.com/inventory.html";
        String actRes = login.loginToApp();
        Assert.assertEquals(expRes, actRes);
        Reporter.log("URL of web application :- " + actRes);
    }

    @Test (enabled = true)
    public void verifyLabelofInventoryTest() throws Exception
    {
        String expRes = "PRODUCTS";
        String actRes = login.verifyLabelofInventory();
        Assert.assertEquals(expRes, actRes);
        Reporter.log("Label Title :- " + actRes);
    }
}

```

```

    }

    @AfterMethod
    public void closeBrowser()
    {
        driver.close();
    }
}

```

## InventoryPage Test

```

package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.InventoryPage;
import Pages.LoginPage;

public class InventoryPageTest extends TestBase {

    LoginPage login;
    InventoryPage invent;

    @BeforeMethod
    public void setup() throws Exception
    {
        initialization();
        login = new LoginPage();
        invent = new InventoryPage();
        login.loginToApp();
    }

    @Test(enabled = false)
    public void verifyPeekLogoTest()
    {
        boolean result = invent.verifyPeekLogo();
        Assert.assertEquals(result, true);
    }

    @Test
    public void addProductTest() throws Exception

```

```

    {
        String result = invent.addProduct();
        Assert.assertEquals(result, "3");
        Reporter.log("Total product added = " + result);
    }

    @AfterMethod
    public void closeBrowser()
    {
        driver.close();
    }
}

```

### DependentTest Class

```

package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.LoginPage;

public class DependsonTest extends TestBase {

    LoginPage login;// = new LoginPage();

    @BeforeMethod
    public void setup() throws Exception
    {
        initalization();
        login = new LoginPage();
    }

    @Test
    public void verifyTitleTest() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        Assert.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }

    @Test (dependsOnMethods = "verifyTitleTest")

```



```

    public void loginToAppTest() throws Exception
    {
        String expRes = "https://www.saucedemo.com/inventory.html";
        String actRes = login.loginToApp();
        Assert.assertEquals(expRes, actRes);
        Reporter.log("URL of web application :- " + actRes);
    }

    @AfterMethod
    public void closeBrowser()
    {
        driver.close();
    }
}

```

### PriorityTest Class

```

package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.LoginPage;

public class PriorityTest extends TestBase {

    LoginPage login;// = new LoginPage();

    @BeforeMethod (alwaysRun = true)
    public void setup() throws Exception
    {
        initalization();
        login = new LoginPage();
    }

    @Test (priority = 0, groups = "Sanity")
    public void verifyTitleTest1() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        Assert.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }
}

```

```

    }

    // 0 1 -1 -200 -1000 12 34 = -1000 -200 -1 0 1 12 34

    @Test (priority = -122)
    public void verifyTitleTest2() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        Assert.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }

    @Test (priority = -122)
    public void verifyTitleTest3() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        Assert.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }

    @Test (priority = 2)
    public void verifyTitleTest4() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        Assert.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }

    @Test (priority = 1)
    public void verifyTitleTest5() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        Assert.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }

    @AfterMethod (alwaysRun = true)
    public void closeBrowser()
    {
        driver.close();
    }

}

```

## GroupTest Class

```
package TestCases;

import org.testng.annotations.Test;
import org.testng.AssertJUnit;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.LoginPage;

public class GroupTest extends TestBase {

    LoginPage login;// = new LoginPage();

    @BeforeMethod (alwaysRun = true)
    public void setup() throws Exception
    {
        initalization();
        login = new LoginPage();
    }

    @Test (groups = {"Sanity","Regression"})
    public void Test1() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        AssertJUnit.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }

    @Test (groups = "Smoke")
    public void Test2() throws Exception
    {
        String expTitle = "Swag Labs";
        String actTitle = login.verifyTitle();
        AssertJUnit.assertEquals(expTitle, actTitle);
        Reporter.log("Title of web application :- " + actTitle);
    }
}
```

```

@Test (groups = {"Smoke","Regression"})
public void Test3() throws Exception
{
    String expTitle = "Swag Labs";
    String actTitle = login.verifyTitle();
    AssertJUnit.assertEquals(expTitle, actTitle);
    Reporter.log("Title of web application :- " + actTitle);
}

@Test (groups = "Sanity")
public void Test4() throws Exception
{
    String expTitle = "Swag Labs";
    String actTitle = login.verifyTitle();
    AssertJUnit.assertEquals(expTitle, actTitle);
    Reporter.log("Title of web application :- " + actTitle);
}

@AfterMethod (alwaysRun = true)
public void closeBrowser()
{
    driver.close();
}
}

```

### **DemoTest Class (Hard Assertion)**

```

package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.LoginPage;

public class DemoTest extends TestBase {

LoginPage login;// = new LoginPage();

@BeforeMethod
public void setup() throws Exception
{
    initialization();
    login = new LoginPage();
}
}

```

```

@Test (invocationCount = 1, enabled = false)
public void verifyTitleTest() throws Exception
{
    String expTitle = "Swag Labs";
    String actTitle = login.verifyTitle();
    Assert.assertEquals(expTitle, actTitle);
    Reporter.log("Title of web application :- " + actTitle);
}

@Test (timeOut = 2000)
public void verifyTitleTest1() throws Exception
{
    String expTitle = "1Swag Labs";
    String actTitle = login.verifyTitle();
    Assert.assertEquals(expTitle, actTitle, "Test case fail zali ka");
    System.out.println("This is test case of title");
    Reporter.log("Title of web application :- " + actTitle);
    Assert.assertTrue(true);
}

@AfterMethod
public void closeBrowser()
{
    driver.close();
}
}

```

## SoftAssertionTest Class

```

package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;

import Base.TestBase;
import Pages.InventoryPage;
import Pages.LoginPage;

public class SoftAssertion extends TestBase {

    LoginPage login;
    InventoryPage invent;
}

```

```

    @BeforeMethod
    public void setup() throws Exception
    {
        initalization();
        login = new LoginPage();
        invent = new InventoryPage();
        login.loginToApp();
    }

    @Test (enabled = true)
    public void verifyPeekLogoTest()
    {
        System.out.println("Execution started");
        SoftAssert soft = new SoftAssert();
        boolean result = invent.verifyPeekLogo();
        soft.assertEquals(result, false);
//      Assert.assertEquals(result, false);
        System.out.println("Execution ended");
        soft.assertAll();
    }

    @AfterMethod
    public void closeBrowser()
    {
        driver.close();
    }
}

```

## FailedTest Class

```

package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.InventoryPage;
import Pages.LoginPage;

public class FailedTest extends TestBase {

    LoginPage login;
    InventoryPage invent;

    @BeforeMethod
    public void setup() throws Exception

```

```

{
    initalization();
    login = new LoginPage();
    invent = new InventoryPage();
    login.loginToApp();
}

@Test
public void Test1()
{
    boolean result = invent.verifyPeekLogo();
    Assert.assertEquals(result, true);
}

@Test
public void Test2()
{
    boolean result = invent.verifyPeekLogo();
    Assert.assertEquals(result, true);
}

@Test
public void Test3()
{
    boolean result = invent.verifyPeekLogo();
    Assert.assertEquals(result, true);
}

@Test
public void Test4()
{
    boolean result = invent.verifyPeekLogo();
    Assert.assertEquals(result, true);
}

@Test
public void Test5()
{
    boolean result = invent.verifyPeekLogo();
    Assert.assertEquals(result, false);
}

@AfterMethod
public void closeBrowser()
{
    driver.close();
}

```

```
        //5 -- 3F 2P => 3F
    }
```

## Test Resources

### 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">

        <groups>
            <run>
                <include name = "Regression"></include>
                <exclude name = "Smoke"></exclude>
            </run>
        </groups>

        <classes>
            <class name="TestCases.GroupTest"/>
            <class name="TestCases.PriorityTest"/>
        </classes>
    </test> <!-- Test -->
</suite> <!-- Suite -->
```

## TestData Folder

### Pom.xml

```
url = https://www.saucedemo.com/
username = standard_user
password = secret_sauce
browser = chrome
```

### 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>Project17Sep</groupId>
    <artifactId>Project17Sep</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <dependencies>
```



```
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>4.0.0</version>
</dependency>
```

```
<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>7.0.0</version>
  <scope>test</scope>
</dependency>
```

```
<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>webdrivermanager</artifactId>
  <version>5.0.3</version>
</dependency>
```

```
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi</artifactId>
  <version>5.2.3</version>
</dependency>
```

```
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi-ooxml</artifactId>
  <version>5.2.3</version>
</dependency>
```

```
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi-ooxml-schemas</artifactId>
  <version>4.1.2</version>
</dependency>
</dependencies>
```

```
</project>
```