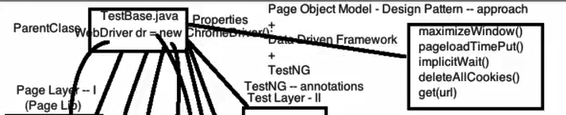
Testbase class: this is the parent class

Here we can initialize the properties , excell data, implicitwait,maximize,initialize webdriver,

Delete all cookies,geturl(),pageload time all should be declared here.



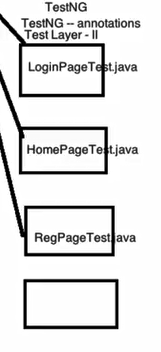
Phase1: here all the pages can be created here with page objects,actionmethods,webelements

Every page should extents testbaseclass. Here test cases are not written.here we can create individual classes for individual pages.



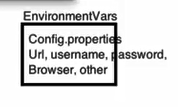
Phase 2:

In this stage test cases should be writen should be extents testbase class. With @test annotation (testNg annotations).

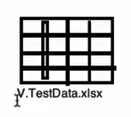


Phase 3:

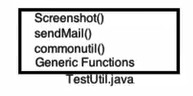
Create config.properties file. Here we can put environmental variables like url.username,password,browser,others



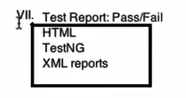
Create testdata in excell file:



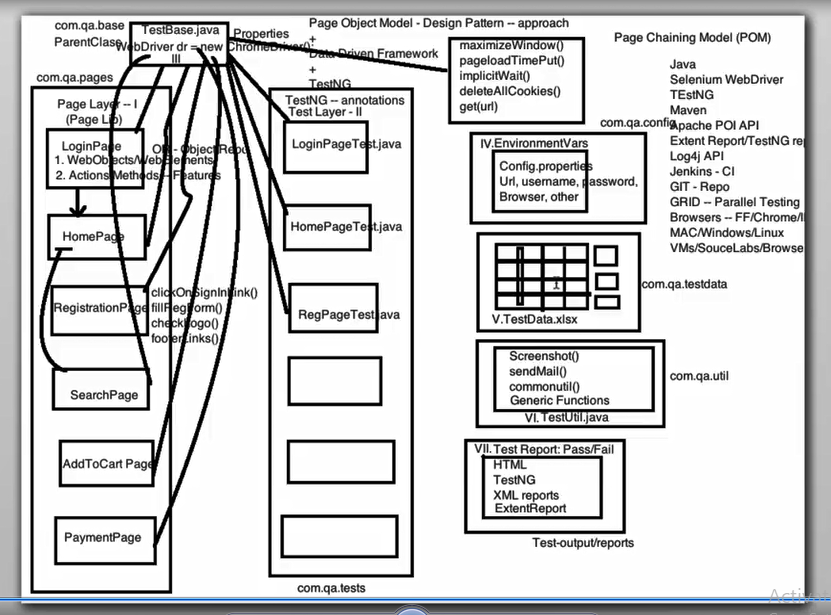
Create Testutilities



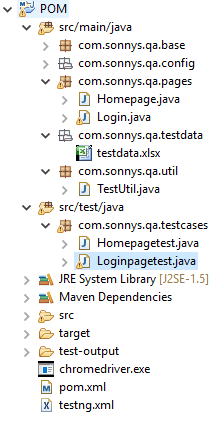
Reports



Overview of pom



Below is the project structure.



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>POMtest</groupId>

<artifactId>POM</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>POM</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.141.59</version>

</dependency>

<dependency>

<groupId>com.beust</groupId>

<artifactId>jcommander</artifactId>

<version>1.27</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.13</version>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.17</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>3.17</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml-schemas</artifactId>

<version>3.17</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-scratchpad</artifactId>

<version>3.17</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>ooxml-schemas</artifactId>

<version>1.4</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>openxml4j</artifactId>

<version>1.0-beta</version>

</dependency>

<dependency>

<groupId>com.relevantcodes</groupId>

<artifactId>extentreports</artifactId>

<version>2.40.2</version>

</dependency>

<!-- <dependency>

<groupId>com.aventstack</groupId>

<artifactId>extentreports</artifactId>

<version>3.0.0</version>

</dependency> -->

</dependencies>

</project>

Config.properties:

url=https://sonnysstg.techwave.net/

username=uatpfazio1@sonnysdirect.com

password=123456

browser=chrome

Testbase class:

**package** com.sonnys.qa.base;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.IOException;

**import** java.util.Properties;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.WebDriver;

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

**import** org.openqa.selenium.support.events.EventFiringWebDriver;

**import** com.sonnys.qa.util.TestUtil;

**public** **class** Testbase {

**public** **static** WebDriver *dr*;

**public** **static** Properties *prop*;

**public** EventFiringWebDriver e\_driver;

**public** Testbase (){

**try**{

*prop*=**new** Properties();

FileInputStream ip=**new** FileInputStream(System.*getProperty*("user.dir")+"\\src\\main\\java\\com\\sonnys\\qa\\config\\config.properties");

*prop*.load(ip);

}**catch**(FileNotFoundException e){

e.printStackTrace();

}**catch**(IOException IOe){

IOe.printStackTrace();

}

}

**public** **void** initialization(){

String browsername=*prop*.getProperty("browser");

**if**(browsername.equals("chrome")){

System.*setProperty*("webdriver.chrome.driver", "E:\\testworkspace\\POM\\chromedriver.exe");

*dr*=**new** ChromeDriver();

}**else** **if**(browsername.equals("firefox")){

System.*setProperty*("webdriver.gecko.driver", "E:\\testworkspace\\POM\\geckodriver.exe");

*dr*=**new** ChromeDriver();

}

*dr*.manage().window().maximize();

*dr*.manage().deleteAllCookies();

*dr*.manage().timeouts().pageLoadTimeout(TestUtil.*PAGE\_LOAD\_TIMEOUT*, TimeUnit.***SECONDS***);

*dr*.manage().timeouts().implicitlyWait(TestUtil.*IMPLICITWAIT*, TimeUnit.***SECONDS***);

*dr*.get(*prop*.getProperty("url"));

}

}

Homepage:

package com.sonnys.qa.pages;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import com.sonnys.qa.base.Testbase;

public class Homepage extends Testbase{

@FindBy(xpath="/html/body/main/div[3]/div/div/ul/li[1]/a")

WebElement signin;

@FindBy(xpath="/html/body/main/div[2]/nav/div/div[1]/div/a/img")

WebElement logo;

@FindBy(xpath="//a[contains(text(),'SHOP')]")

WebElement Shop;

@FindBy(xpath="//a[contains(text(),'PARTS & SUPPLIES')]")

WebElement partandsupply;

@FindBy(xpath="//a[contains(text(),'AIR SUPPLY')]")

WebElement airsupply;

public Homepage(){

PageFactory.initElements(dr, this);

}

public String verifytitle(){

return dr.getTitle();

}

public void verifylogo(){

logo.click();

}

public void clickonAIRSUPPLY(){

Actions a=new Actions(dr);

a.moveToElement(Shop).build().perform();

a.moveToElement(partandsupply);

airsupply.click();

}

public Login clicksignin(){

signin.click();

return new Login();

}

}

Homepage test:

package com.sonnys.qa.testcases;

import org.testng.Assert;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

import com.sonnys.qa.base.Testbase;

import com.sonnys.qa.pages.Homepage;

public class Homepagetest extends Testbase{

Homepage home;

public Homepagetest(){

super();

}

@BeforeMethod

public void setup(){

initialization();

home=new Homepage();

}

@Test(priority=3)

public void verifyHomepageclicksignin(){

String title=home.verifytitle();

Assert.assertEquals(title, "Car Wash Equipment Supplier | Sonny’s The CarWash Factory");

System.out.println("verifyHomepageclicksignin passed");

}

@Test(priority=1)

public void logo(){

home.verifylogo();

System.out.println("logo passed");

}

/\* @Test(priority=2)

public void sigin(){

home.clicksignin();

System.out.println("signin passed");

}\*/

@Test(priority=2)

public void shop(){

home.clickonAIRSUPPLY();

System.out.println("shop passed");

}

@Test(priority=1)

public void signinclick(){

home.clicksignin();

System.out.println("clicksignin passed");

}

@AfterMethod

public void teardown(){

dr.quit();

}

}

Login

package com.sonnys.qa.pages;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.testng.annotations.DataProvider;

import com.sonnys.qa.base.Testbase;

import com.sonnys.qa.util.TestUtil;

public class Login extends Testbase{

@FindBy (xpath="//\*[@id='j\_username']")

WebElement username;

@FindBy(xpath="/html/body/main/div[2]/nav/div/div[1]/div/a/img")

WebElement logo;

@FindBy (xpath="//\*[@id='j\_password']")

WebElement password;

@FindBy (xpath="//\*[@id='loginForm']/button")

WebElement loginbutton;

//initializing the page objects

public Login(){

PageFactory.initElements(dr, this);

}

//actions

public Boolean loginpagetitle(){

return logo.isDisplayed();

}

public String verifylogo(){

return dr.getTitle();

}

public Homepage login(String un,String pwd){

username.sendKeys(un);

password.sendKeys(pwd);

loginbutton.clear();

return new Homepage();

}

}

Loginpag etest:

package com.sonnys.qa.testcases;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

import com.sonnys.qa.base.Testbase;

import com.sonnys.qa.pages.Homepage;

import com.sonnys.qa.pages.Login;

import com.sonnys.qa.util.TestUtil;

public class Loginpagetest extends Testbase{

Login login;

String sheetname="Login";

public Loginpagetest(){

super();

}

@BeforeMethod

public void setup(){

initialization();

login=new Login();

}

@Test

public void logotest(){

login.verifylogo();

System.out.println("logo passed");

}

@DataProvider

public Object[][] getlogintestdata(){

Object data[][]=TestUtil.gettestdata("sheetname");

return data;

}

@Test(priority=2,dataProvider="getlogintestdata")

public void logotest1(String username,String password){

login.login(username,password);

System.out.println("logo passed");

}

}

Testng.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name=*"Suite"*>

<test thread-count=*"5"* name=*"Test"*>

<classes>

<class name=*"com.sonnys.qa.testcases.Homepagetest"*>

<methods>

<include name=*"signinclick"*/>

</methods>

</class>

<class name=*"com.sonnys.qa.testcases.Loginpagetest"*>

<methods>

<include name=*"logotest1"*/>

</methods>

</class>

</classes>

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

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

Testutils:

package com.sonnys.qa.util;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.openxml4j.exceptions.InvalidFormatException;

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

import org.apache.poi.ss.usermodel.Workbook;

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

public class TestUtil {

public static long PAGE\_LOAD\_TIMEOUT=20;;

public static long IMPLICITWAIT=10;

public static String TESTDATA\_SHEET\_PATH=System.getProperty("user.dir")+"\\src\\main\\java\\com\\sonnys\\qa\\testdata\\testdata.xlsx";

static Workbook book;

static Sheet sheet;

public static Object[][]gettestdata(String sheetname){

FileInputStream file=null;

try{

file=new FileInputStream(TESTDATA\_SHEET\_PATH);

}

catch(FileNotFoundException e){

e.printStackTrace();

}

try{

book=WorkbookFactory.create(file);

}catch(IOException e){

e.printStackTrace();

}catch(InvalidFormatException e){

e.printStackTrace();

}

sheet=book.getSheet(sheetname);

Object[][]data=new Object[sheet.getLastRowNum()][sheet.getRow(0).getLastCellNum()];

for(int i=0;i<sheet.getLastRowNum();i++){

for(int k=0;k<sheet.getRow(0).getLastCellNum();k++){

data[i][k]=sheet.getRow(i+1).getCell(k).toString();

}

}

return data;

}

}