BDD with Page Factory:

**package** Com.Proj.Cucumber.BaseUtil;

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

**public** **class** Base

{

**public** WebDriver driver; //null

}

package Com.Prog.Cucumber.Hooks;

import org.openqa.selenium.chrome.ChromeDriver;

import Com.Proj.Cucumber.definations.Base;

import io.cucumber.java.After;

import io.cucumber.java.AfterAll;

import io.cucumber.java.Before;

import io.cucumber.java.BeforeAll;

public class Hooks

{

int x; //=0

private static Base base;//null

public Hooks()

{ }

public Hooks(Base base)

{

this.base= base;

}

@Before

public void initBrowser()

{

System.out.println("opening the Browser : Chrome");

base.driver = new ChromeDriver();

}

@After

public void tearBrowser()

{

System.out.println("Closing the Browser : chrome");

//base.driver.quit();

}

@BeforeAll

public static void before\_or\_after\_all1()

{

System.out.println(" @@BeforeAll exected...");

}

@AfterAll

public static void before\_or\_after\_all2()

{

System.out.println(" @@@AfterAll exected...");

}

// @BeforeStep

// public void doSomethingBeforeStep(Scenario @s2){

// }

//

// @AfterStep

// public void doSomethingAfterStep(Scenario scenario){

// }

}

Feature: Store login

This feature deals with the login functionality.

#https://demo.guru99.com/test/newtours/

Scenario Outline: Login with correct username and correct password

Given I navigate to login page

When I insert "<userName>" and "<passWord>"

And I clicked signin button

Then Its navigated to the home page

Examples:

| userName | passWord |

| mercury | mercury |

| admin2 | admin22 |

| admin3 | admin33 |

**package** Com.Prog.Cucumber.Pages;

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

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.support.FindBy;

**import** org.openqa.selenium.support.How;

**import** org.openqa.selenium.support.PageFactory;

**public** **class** HomePage

{

WebDriver driver;

**public** HomePage(WebDriver driver)

{

PageFactory.*initElements*(driver, **this**);

}

// locators

// action method

//By unm= By.name("userName");

@FindBy(name="userName") WebElement username;

@FindBy(how=How.***NAME*** , using="password")WebElement password;

@FindBy(how=How.***NAME*** , using="submit")WebElement submit\_button;

//Driver initialization

**public** **void** setUserName(String uid)

{

username.sendKeys(uid);

}

**public** **void** setPassword(String pass)

{

password.sendKeys(pass);

}

**public** **void** clickSubmitButton()

{

submit\_button.click();

}

**public** **void** login\_to\_FlightBooking(String uid, String pass)

{

username.sendKeys(uid);

password.sendKeys(pass);

submit\_button.click();

//BrowserFactory.wait(1000,username.sendKeys(uid));

}

}

**package** Com.Proj.Cucumber.definations;

**import** org.openqa.selenium.By;

**import** org.testng.Assert;

**import** Com.Prog.Cucumber.Pages.HomePage;

**import** io.cucumber.java.en.And;

**import** io.cucumber.java.en.Given;

**import** io.cucumber.java.en.Then;

**import** io.cucumber.java.en.When;

**public** **class** LoginPageClass

{

**private** **static** Base *base*;

**public** LoginPageClass(Base base)

{

**this**.*base* = base;

}

@Given("^I navigate to login page$")

**public** **void** navigateToLoginPage(){

System.***out***.println("the Driver name is :"+ *base*.driver);

System.***out***.println("Navigated to the Login Page");

*base*.driver.get("https://demo.guru99.com/test/newtours/");

}

@Then("^Its navigated to the home page")

**public** **void** navigateToHomePage(){

System.***out***.println("Navigated to the Home Page");

Assert.*assertTrue*(*base*.driver.findElement(By.*linkText*("SIGN-OFF")).isEnabled());

System.***out***.println("Navigated to home page...");

}

@And("^I clicked signin button")

**public** **void** clickSigninButton(){

HomePage home= **new** HomePage(*base*.driver);

home.clickSubmitButton();

}

@When("I insert {string} and {string}")

**public** **void** enterUsernamePassword(String username , String password)

{

HomePage home= **new** HomePage(*base*.driver);

home.setUserName(username);

home.setPassword(password);

}

}

package Com.Proj.Cucumber.TestRunner;

import org.testng.annotations.Test;

import io.cucumber.testng.AbstractTestNGCucumberTests;

import io.cucumber.testng.CucumberOptions;

//@CucumberOptions(

// dryRun=false,

// features= {".\\src\\test\\java\\Com\\Proj\\Cucumber\\FeaturePkg\\login.feature",

// ".\\src\\test\\java\\Com\\Proj\\Cucumber\\FeaturePkg\\login2.feature"},

// glue={"Com\\Proj\\Cucumber\\definations","Com\\Prog\\Cucumber\\Hooks"},

// plugin = {"pretty", "html:target/cucumber-report.html"}

//

//

// )

@CucumberOptions(

dryRun=false,

features= {".\\src\\test\\java\\Com\\Proj\\Cucumber\\FeaturePkg\\login2.feature"},

glue={"Com\\Proj\\Cucumber\\definations",

"Com\\Prog\\Cucumber\\Hooks"},

plugin = {"pretty", "html:target/cucumber-report.html"

}

)

public class Runner extends AbstractTestNGCucumberTests

{

}

<?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">

<classes>

<class name="Com.Proj.Cucumber.TestRunner.Runner"></class>

</classes>

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

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

Page object model:

-------------

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-picocontainer -->

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-picocontainer</artifactId>

<version>7.23.0</version>

</dependency>

===============

<build>

<pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

<plugins>

<!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.4.0</version>

</plugin>

<!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.3.1</version>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.13.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.3.0</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>RunAll.xml</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

<plugin>

<artifactId>maven-jar-plugin</artifactId>

<version>3.4.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>3.1.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>3.1.2</version>

</plugin>

<!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->

<plugin>

<artifactId>maven-site-plugin</artifactId>

<version>3.12.1</version>

</plugin>

<plugin>

<artifactId>maven-project-info-reports-plugin</artifactId>

<version>3.6.1</version>

</plugin>

</plugins>

</pluginManagement>

</build>

============

Exection on maven from commond prompt:

1. install maven on machine

2. set the maven path in environmnet varible

3. Extecute the project

find the project location (POM.xlm is visible)

mvn clean

mvn install