POM

package BaseLayer;

import java.time.Duration;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class BaseClass {

public static WebDriver driver;

public static void initialization() {

driver = new ChromeDriver();

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(30));

driver.manage().window().maximize();

driver.get("https://www.facebook.com/reg");

}

}

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

package PageLayer;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import BaseLayer.BaseClass;

public class RegisterPage extends BaseClass {

// create Object Repistory usin Page Object model without Page Factory

private By firstName = By.name("firstname");

private By lastname = By.name("lastname");

private By username = By.name("reg\_email\_\_");

private By password = By.name("reg\_passwd\_\_");

// initialize the object repository using constructor

public RegisterPage(WebDriver driver) {

super.driver = driver;

}

// write associated method without entering the test data for each and every

// object Repository

public void registerFunctionality(String Fname,String LName, String Uname, String Pass)

{

driver.findElement(firstName).sendKeys(Fname);

driver.findElement(lastname).sendKeys(LName);

driver.findElement(username).sendKeys(Uname);

driver.findElement(password).sendKeys(Pass);

}

}

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

package TestLayer;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import BaseLayer.BaseClass;

import PageLayer.RegisterPage;

public class RegisterPageTest extends BaseClass {

@BeforeTest

public void setUp() {

BaseClass.initialization();

}

@Test

public void validateRegisterFunctionality() {

RegisterPage registerPage = new RegisterPage(driver);

registerPage.registerFunctionality("Pooja", "Mane", "pooja@gmail.com", "Pooja@123");

}

@AfterTest

public void tearDown() {

// driver.quit();

}

}=============================================================

What is POM? VVVVVIMMPPPPPPPPPPPPPPPPPP

- POM means Page Object Model.

- Page Object Model is an element or object design pattern to create the Object Repository (OR).

Element/object means : text box, radio button, button, checkbox, drop down, link, images, physical text etc..

- We create object Repository for centralized maintenance of Object or element.

- We maintain all Object Repository in Page Layer package in src/main/java.

- Under the Page Object Model we create container classes for each and every Web Page.

- just example in application we have different pages as like Register Page, Login Page, Home Page, Customer creation Page,

Employee Information page etc.

- As per the application wise we create container classes in Page Layer package as like RegisterPage class, LoginPage class, CustomerCreationPage class, EmployeeInformationPage class etc.

- there are 2 different ways we can create Object Repository in Page Layer package VVVVVVIMMPPPP

1) Page Object Model with Page Factory

2) Page Object Model without Page Factory.

- There is No Object repository facility in selenium Webdriver, Selenium WebDriver is an programming interface.

- under the Page Object Model for each and every Web Page in application there should be corresponding Page class.

- In this Page Layer Package we maintain 3 different things VVVVVVIMMPPPP

1) we create private object repository for web element

2) we initialize the Object Repository using constructor

3) we create associated methods for each and every object repository without entering the test data.

- if there is no Object Repository concept then it difficult to maintain the test cases or test scripts.

- in current company we use """Page Object Model with Page Factory""""

Adv of Page Object Model

1) Reusability of the Web Element as well associated methods.

2) It is efficient and scalable

3) centralized maintenance of object Repository

4) we can reduce the project size.

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

2) Page Object Model without Page Factory.

package BaseLayer;

public class BaseClass

{

//declare the WebDriver global static

public static void initialization()

{

//connect to actual browser

//up casting

//implicit wait

//Page load time out

//maximize the browser

//open a url

}

}

package PageLayer;

public class LoginPage extends BaseClass

{

//Page Object Model (POM) without Page Factory

private By username = By.name("username");

private By password = By.name("password");

private By loginButton = By.xpath("//button[text()=' Login ']");

//initialize the object Repository using constructor

public LoginPage(WebDriver driver)

{

//parent driver and child driver are same.

super.driver =driver;

}

//create associated method for each and every object repository without entering the test data

public void loginFunctionality(String Uname, String Pass)

{

driver.findElement(username).sendKeys(Uname);

driver.findElement(password).sendKeys(Pass);

driver.findElement(loginButton).click();

}

}

package TestLayer;

public class LoginPageTest extends BaseClass

{

// Pre condition annotation using Pre Condition annotation

//create the test cases using @Test annotation

//call associated methods from respective Page Layer classes by creating object of class and at the time of calling associated method

entering the test data.

//write assertion for test cases using hard assert

// Post condition annotation using Post condition annotation

@BeforeTest

public void setup()

{

BaseClass.initialization();

}

@Test

public void validateLoginFunctionality()

{

LoginPage loginPage =new LoginPage(driver);

loginPage.loginFunctionality("Admin", "admin123");

}

@AfterTest

public void teardown()

{

driver.quit();

}

}

package BaseLayer;

import java.time.Duration;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class BaseClass {

// declare WebDriver as global static

public static WebDriver driver;

public static void initialization() {

// connect to actual browser

// Up casting

driver = new ChromeDriver();

// implicitlyWait

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

// pageLoadTimeout

driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(30));

// maximize the browser

driver.manage().window().maximize();

// deleteAllCookies

driver.manage().deleteAllCookies();

// open a URL

driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

}

}

package PageLayer;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import BaseLayer.BaseClass;

public class LoginPage extends BaseClass {

// Create Object Repository using Page Object Model without Page Factory concept

private By username = By.name("username");

private By password = By.name("password");

private By loginbutton = By.xpath("//button[text() = ' Login ']");

// Initialize the object repository by creating the constructor

public LoginPage(WebDriver driver) {

super.driver = driver;

}

// create associated method for each and every object repository without

// entering the test data

public void loginFunctionality(String UName, String Pass)

{

driver.findElement(username).sendKeys(UName);

driver.findElement(password).sendKeys(Pass);

driver.findElement(loginbutton).click();

}

}

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

package TestLayer;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import BaseLayer.BaseClass;

import PageLayer.LoginPage;

public class LoginPageTest extends BaseClass {

// Pre condition annotation

@BeforeTest

public void setUp() {

BaseClass.initialization();

}

// create test cases

// call associated methods from page layer package by creating object of Page

// layer classes and at the time of calling the method entering test data.

@Test

public void validateLoginFunctionality()

{

LoginPage loginPage =new LoginPage(driver);

loginPage.loginFunctionality("Admin", "admin123");

}

//post condtion annotation

@AfterTest

public void tearDown()

{

//driver.quit();

}

}

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

2) Page Object Model with Page Factory

- in Page Object Model with Page Factory, we find single element by using @FindBy() annotation by passing locator name and value.

syntax

@FindBy(id="id value")

private WebElement username;

- @FindBy() annotation present inside the selenium webdriver library

- we find multiple elements by using @FindBys() annotation by passing @FindbBy(id="value")

syntax:

@FindBys(@FindBy(id="id value"))

private List<WebElement> listWb;

- we initialize the object repository by using PageFactory.initElements() method and by passing driver, this keyword in constructor

public classname()

{

PageFactory.initElements(driver, this);

}

package BaseLayer;

public class BaseClass

{

//declare Webdriver as global static

public static void initialization()

{

//connect to actual browser

//Up casting

//implicit wait

//page load time out

//maximize

//delete all cookies

//open a url

}

}

package PageLayer;

public class LoginPage extends BaseClass

{

//create private Object Repository with Page factory

@FindBy(name="username")

private WebElement username;

@FindBy(name="password")

private WebElement password;

@FindBy(xpath="//button[text() = ' Login ']")

private WebElement loginButton;

//initialize the Object repository by using PageFactory.initElements(driver, this) in constructor

public LoginPage()

{

PageFactory.initElements(driver, this);

}

//create associated method for each and every object repository without entering the test data

public void loginFunctionality(String UName, String Pass)

{

username.sendKeys(UName);

password.sendKeys(Pass);

loginButton.click();

}

}

package TestLayer;

public class LoginPageTest extends BaseClass

{

// Pre condition

@BeforeTest

public void setup()

{

BaseClass.initialization();

}

//create test cases

//call associated method from Page Layer package and by creating object of Page Layer classes and at time of calling method enter the test data

@Test

public void validateLoginFunctionality()

{

LoginPage loginPage =new LoginPage();

loginPage.loginFunctionality("Admin", "admin123");

}

//post condition annotation

@AfterTest

public void tearDown()

{

driver.quit();

}

}

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

package BaseLayer;

import java.time.Duration;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class BaseClass {

// declare WebDriver global static

public static WebDriver driver;

public static void initilization() {

// connect to actual browser code

// up casting

driver = new ChromeDriver();

// implicitlyWait

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

// pageLoadTimeout

driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(30));

// maximize

driver.manage().window().maximize();

// deleteAllCookies

driver.manage().deleteAllCookies();

// open a URL

driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

}

}

package PageLayer;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import BaseLayer.BaseClass;

public class LoginPage extends BaseClass {

// create Object Repository using Page Object Model with Page Factory

@FindBy(name = "username")

private WebElement username;

@FindBy(name = "password")

private WebElement password;

@FindBy(xpath = "//button[text()= ' Login ']")

private WebElement loginbutton;

// initialize the Object repository using PageFactory.initElements(driver,this)

// in constructor

public LoginPage() {

PageFactory.initElements(driver, this);

}

// create associated method for each and every object Repository without

// entering the test data.

public void loginFunctionality(String UName, String Pass)

{

username.sendKeys(UName);

password.sendKeys(Pass);

loginbutton.click();

}

}

package TestLayer;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import BaseLayer.BaseClass;

import PageLayer.LoginPage;

public class LoginPageTest extends BaseClass {

// pre condition

@BeforeTest

public void setUp() {

BaseClass.initilization();

}

/// create test cases

// call associated method from Page Layer Package by creating Object of Page

/// layer classes and at the time of calling the associated method enter the

/// test data

@Test

public void validateLoginFunctionality()

{

LoginPage loginPage =new LoginPage();

loginPage.loginFunctionality("Admin", "admin123");

}

//Post condition annotation

@AfterTest

public void tearDown()

{

//driver.quit();

}

}

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

package BaseLayer;

import java.time.Duration;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class BaseClass {

// declare WebDriver global static

public static WebDriver driver;

public static void initilization() {

// connect to actual browser code

// up casting

driver = new ChromeDriver();

// implicitlyWait

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

// pageLoadTimeout

driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(30));

// maximize

driver.manage().window().maximize();

// deleteAllCookies

driver.manage().deleteAllCookies();

// open a URL

driver.get("https://accounts.digilocker.gov.in/signup/smart\_v2/1bad09c305565fabf8122991da6bd537--en");

}

}

package PageLayer;

import java.util.List;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.FindBys;

import org.openqa.selenium.support.PageFactory;

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

import BaseLayer.BaseClass;

public class SignUpPage extends BaseClass {

// create Private Object Repository using Page Factory

@FindBy(id="name")

private WebElement fullName;

@FindBy(id="dd")

private WebElement date;

@FindBy(id="mm")

private WebElement month;

@FindBy(id="yy")

private WebElement year;

@FindBys(@FindBy(xpath="//input[@name='gender']/parent::label"))

private List<WebElement> listGender;

@FindBy(id="mobile")

private WebElement mobile;

@FindBy(id="email")

private WebElement email;

@FindBy(id="profilepin")

private WebElement profilePin;

//initialize the Object using PageFactory.initElements(driver, this) in constructor

public SignUpPage()

{

PageFactory.initElements(driver, this);

}

//create associated method for each and every object repository without entering the test data

public void createANewAccount(String FullName, String Date, String Month, String Year, String Gender, String Mobile, String Email, String Pin)

{

fullName.sendKeys(FullName);

Select selDate= new Select(date);

selDate.selectByVisibleText(Date);

Select selMont= new Select(month);

selMont.selectByVisibleText(Month);

Select selYear =new Select(year);

selYear.selectByVisibleText(Year);

//iterate all value

for(WebElement gen: listGender)

{

String abc = gen.getText();

if(abc.equalsIgnoreCase(Gender))

{

gen.click();

}

}

mobile.sendKeys(Mobile);

email.sendKeys(Email);

profilePin.sendKeys(Pin);

}

}

package TestLayer;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import BaseLayer.BaseClass;

import PageLayer.SignUpPage;

public class SignUpPageTest extends BaseClass {

@BeforeTest

public void setUp() {

BaseClass.initilization();

}

@Test

public void validatecreateANewAccount() {

SignUpPage signUpPage = new SignUpPage();

signUpPage.createANewAccount("Sarthi Patil", "23", "May", "2004", "female", "9090999", "rahul@gmail.com",

"12345");

}

@AfterTest

public void tearDown() {

// driver.quit();

}

}

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

https://www.vsp.com/create-account

https://copyright.gov.in/UserRegistration/frmNewUser.aspx

https://www.indiapost.gov.in/VAS/pages/userregistration.aspx

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