PAGE OBJECT MODEL

- WHAT ARE COMPONENTS ?
- HOW TO HANDLE ?

@rajatt95

SUBSCRIBE

1. PROJECT SETUP

```xml
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <!-- <version>4.10.0</version> -->
  <version>${selenium.java.version}</version>
</dependency>

<!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>webdrivermanager</artifactId>
  <version>${webdrivermanager.version}</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>${testng.version}</version>
  <scope>test</scope>
</dependency>
```

@rajatt95

2. Web Elements - Components

```
/*********************************************************************************************/
// Data Members | Properties
// Web Elements Declaration
/*********************************************************************************************/

// Without Page Factories

// Creating WebElements as private and public methods to access them -
  // This is the example of ENCAPSULATION (OOPS Concept)
    // Encapsulation refers to integrating data (variables) and code (methods) into a single unit.
    // A class's variables are hidden from other classes and can only be accessed by the methods of the class

// Locator for the heading SwagLabs
1 usage
private final By heading_SwagLabs = By.xpath( xpathExpression: "//*[@class='app_logo']");
```

@rajatt95

3. Operations - COMPONENT

```java
/**************************************************************************************/
// Methods | Tasks | Functionality | Behavior
// Operations on the Web Elements
/**************************************************************************************/

/**
 * Returns the WebElement representing the heading SwagLabs in the header component.
 *
 * @return The WebElement representing the heading SwagLabs.
 */
1 usage
public WebElement get_Element_Heading_SwagLabs(){
    return driver.findElement(heading_SwagLabs);
}
```

@rajatt95

4. Declare Web Elements - PAGE

```
/**************************************************************************************/
// Data Members | Properties
// Web Elements Declaration
/**************************************************************************************/

// Without Page Factories

// Creating WebElements as private and public methods to access them -
    // This is the example of ENCAPSULATION (OOPS Concept)
        // Encapsulation refers to integrating data (variables) and code (methods) into a single unit.
        // A class's variables are hidden from other classes and can only be accessed by the methods of the class

1 usage
private final By icon_Twitter = By.xpath( xpathExpression: "//a[normalize-space()='Twitter']");
1 usage
private final By msg_Footer = By.xpath( xpathExpression: "//div[@class='footer_copy']");
```

@rajatt95

5. Operations - PAGE

```java
/*********************************************************************************************/
// Methods | Tasks | Functionality | Behavior
// Operations on the Web Elements
/*********************************************************************************************/

/**
 * Returns the WebElement representing the Twitter icon on the Home page.
 *
 * @return The WebElement representing the Twitter icon.
 */
no usages
public WebElement get_Element_Icon_Twitter() { return driver.findElement(icon_Twitter); }

/**
 * Returns the WebElement representing the footer message on the Home page.
 *
 * @return The WebElement representing the footer message.
 */
1 usage
public WebElement get_Element_Msg_Footer() { return driver.findElement(msg_Footer); }
```

@rajatt95

6. PAGE - Additional Step

```java
public class HomePage {

    3 usages
    WebDriver driver;

    2 usages
    private AppHeader appHeader;

    1 usage
    public AppHeader getAppHeader() {
        return appHeader;
    }

    /**
     * Constructor for the HomePage class.
     * Initializes the WebDriver.
     *
     * @param driver The WebDriver instance to be used for interacting with the web application.
     */
    public HomePage(WebDriver driver){
        // RHS driver - coming from the test class
            // When we are creating the object of this class
                // Constructor of this class gets called
        // LHS driver - defined in this class
        this.driver = driver;
        appHeader = new AppHeader(driver);
    }
```

@rajatt95

7. Create Test and call Objects

```java
@Test
public void test_Login_Functionality_Valid_Credentials(){

    getDriver().get("https://www.saucedemo.com/");

    LoginPage loginPage = new LoginPage(getDriver());
    // This is the example of BUILDER PATTERN
        // After this operation, User remains on the same page

    // This is the example of FLUENT INTERFACE
        // After clicking on the Login button, User navigates to the HomePage
    HomePage homePage = loginPage.
            fill_TextBox_Username("standard_user").
            fill_TextBox_Password("secret_sauce").
            click_Button_Login();

    // Assert.assertEquals() -
        // Boolean values as Actual and Expected
        // String values as Actual and Expected
    // This is the example of POLYMORPHISM (OOPS Concept) - Compile Time (Method Overloading)
        // It allows us to perform a single action in different ways.
    Assert.assertEquals(homePage.getAppHeader().get_Element_Heading_SwagLabs().isDisplayed(), expected: true,
            message: "Assertion for Heading: Swag Labs presence in the Header");

    String msg_FooterText_Actual = homePage.get_Element_Msg_Footer().getText();
    String msg_FooterText_Expected = "© 2023 Sauce Labs. All Rights Reserved. Terms of Service | Privacy Policy";
    Assert.assertEquals(msg_FooterText_Actual, msg_FooterText_Expected,
            message: "Assertion for message present in the Footer");
}
```

@rajatt95

RAJAT VERMA

SUBSCRIBE

SOFTWARE TEST ENGINEER

Gmail: rajatvermaa95@gmail.com

Github Profile: https://github.com/rajatt95

Github Page: https://rajatt95.github.io

LinkedIn: https://www.linkedin.com/in/rajatt95

Topmate: https://topmate.io/rajatt95

Telegram: https://t.me/rajatt95

Instagram: https://www.instagram.com/rajattvermaa95

Youtube: https://www.youtube.com/@rajatt95