

Selenium Class 35 - POM - Page Object Model in Selenium

- Page Object Model is an Object / Element design pattern to create Repositories for web elements
(Ex: Link, Edit box, Button etc...)
- We create Object Repositories for centralized maintenance of elements.
- Under this model container classes for web elements are created that behave as object repositories
- **We can create POM in two ways,**
 - 1) POM without Page factory
 - 2) POM with Page factory
- No Object Repository facility in Selenium WebDriver, It is a programming interface, In UFT/QTP we have object Repository facility (Local and Shared Object Repositories)
- Under this POM, for each web page there should be corresponding Page Class...
- This Page class will find the web elements of the web page and also contains customized methods which perform operations on those web elements.
- If no Object Repository / Page Object Model then it is difficult to maintain elements in Test Cases...

Advantages of Page Object Model

- Reusability of Elements, reduces code size
- Centralized maintenance of Elements

Implement Page Object Model

- In this approach all pages/elements of the application and user actions on those web elements are maintained as methods inside class file/s, by importing the page class/s we can create test cases

1) Create a Page Class for Login Page in Admin Interface of gcrShop Application

```
public class LoginPage {  
    WebDriver driver;  
  
    //Create Elements using Element Locators  
    By User = By.name("username");  
    By Password = By.name("password");  
    By LoginButton = By.id("tdb1");  
  
    //this keyword is invoke an instance current class constructor  
    //A Constructor in Java is a block of code similar to a method  
    //that is called an instance of Object is created.  
  
    public LoginPage(WebDriver driver){
```

```
this.driver = driver;
}
//Create User Actions / Customized Commands
public void typeUsername(String Uname){
driver.findElement(User).sendKeys(Uname);
}

public void captureUsername(){
driver.findElement(User).getAttribute("value");
}

public void clearUsername(){
driver.findElement(User).clear();
}

public void typePassword(String Pwd){
driver.findElement>Password).sendKeys(Pwd);
}
public void clickLoginButton(){
driver.findElement(LoginButton).click();
}
}
```

2) Test Case Class for Admin Login Functionality

```
public class TestCase {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", "E:/chromedriver.exe");
        WebDriver abc = new ChromeDriver();
        abc.manage().window().maximize();

        LoginPage login = new LoginPage(abc);

        //Create Login Test Case
        abc.get("http://www.gcrit.com/build3/admin/");
        login.typeUsername("admin");
        login.typePassword("admin@123");
        login.clickLoginButton();
        Thread.sleep(2000);

        String url = abc.getCurrentUrl();

        if (url.contains("http://www.gcrit.com/build3/admin/index.php")){
            System.out.println("Admin Login is Successful - Passed");
        }
        else
        {
            System.out.println("Admin Login is Unsuccessful - Failed");
        }
    }
}
```

```
abc.close();  
}  
}
```

2a) Create a Page Class / Object Repository (using some elements)

(Username, Password, Login Button, Error message, Online Catalog Link etc...)

2b) Create Test Cases

- i) Admin Login
- ii) Verify "Error Message"
- iii) Verify "Redirect" Functionality

.....

WebDriver Command/s -
To Identify a Web Element,
To perform operation on the Web Element

Customized Command -
Element identification and Operation on the Element

.....

2a) Create Page Class

```
public class LoginPage {  
    WebDriver abcd;  
  
    //Create Web Elements using Element Locators  
    By User = By.name("username");
```

```
By Pwd = By.name("password");
By LoginButton = By.id("tdb1");
By ErrorMessage = By.className("messageStackError");
By CatalogLink = By.linkText("Online Catalog");

public LoginPage (WebDriver abcd2){
this.abcd = abcd2;
}

//Create Customized Commands
public void typeUsername(String Uname){
abcd.findElement(User).sendKeys(Uname);
}
public void typePassword(String Password){
abcd.findElement(Pwd).sendKeys(Password);
}
public void clickLoginButton(){
abcd.findElement(LoginButton).click();
}
public String captureErrorMessage(){
String message = abcd.findElement(ErrorMessage).getText();
return message;
}
public void clickLink(){
abcd.findElement(CatalogLink).click();
}
}
```

2b) Create Test Case 1 Class (Admin Login)

2b) Create Test Case 2 Class (Error Message)

```
System.setProperty("webdriver.chrome.driver", "F:/chromedriver.exe");
```

```
WebDriver Susheela= new ChromeDriver();
```

```
Susheela.manage().window().maximize();
```

```
LoginPage Nathiya = new LoginPage(Susheela);
```

```
Susheela.get("http://www.gcrit.com/build3/admin/");
```

```
Nathiya.typeUsername("abcdef");
```

```
Nathiya.typePassword("xyz123");
```

```
Nathiya.clickLoginButton();
```

```
String error= Nathiya.captureErrorMessage();
```

```
if (error.contains("Error: Invalid administrator login attempt.)){
```

```
System.out.println("Login Unsuccessful and Showing Correct Error Message -  
Passed");
```

```
}
```

```
else
```

```
{
```

```
System.out.println("Login Unsuccessful and Not Showing Correct Error  
Message - Passed");
```

```
}
```

```
Susheela.close();
```

2b) Create Test Case 3 Class (Redirect Functionality)

```
System.setProperty("webdriver.chrome.driver", "F:/chromedriver.exe");  
WebDriver Virat= new ChromeDriver();  
Virat.manage().window().maximize();
```

```
LoginPage redirect = new LoginPage(Virat);  
Virat.get("http://www.gcrit.com/build3/admin/");  
redirect.typeUsername("admin");  
redirect.typePassword("admin@123");  
redirect.clickLoginButton();  
Thread.sleep(3000);  
redirect.clickLink();  
String url=Virat.getCurrentUrl();
```

```
if (url.contains("http://www.gcrit.com/build3/")){  
System.out.println("Page is Redirecting from Admin to User Interface -  
Passed");  
}  
else {  
System.out.println("Page is Not Redirecting from Admin to User Interface -  
Failed");  
}  
Virat.close();  
}  
}
```

.....