

Last updated: Aug 13, 2020

Selenium Java Training - Session 28 - Page Object Model, Page Factory and Autolt

Page Object Model and Page Factory Design Patterns

1. Using Page Object Model and Page Factory design pattern, we can write our scripts in an effective way
 - For each page, we have to create Class and store objects of the page
 - When something changes, we have to update in the respective class, instead of in all test scripts
2. Advantages
 - Achieving lesser maintenance
 - Readability of Scripts
3. Create a Maven Project
 - Create a packages - objectrepository
 - Under objectrepository package, create class for Login page and write the below code:
 - WebDriver driver;
 - Configure Selenium WebDriver for the above statement
 - Create Objects for the required UI elements
 - By email = By.id("input-email");
 - By password = By.id("input-password");
 - By login = By.cssSelector("input[value='Login']");
 - Create a LoginTest class under 'ppack' package
 - Create a login() method - @Test annotated method under this class
 - Configuring TestNG for the above @Test annotation
 - Resolving the error with latest TestNG jar file configuration
 - Write Selenium code inside login() method
 - System.setProperty("webdriver.chrome.driver","D:\\SeleniumTrainingWorksp
ace\\PMDemoProj\\drivers\\chromedriver.exe");
 - WebDriver driver = new ChromeDriver();
 - driver.manage().window().maximize();
 - driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);
 - driver.get("http://tutorialsninja.com/demo/index.php?route=account/login");
 - Create methods to return objects in the LoginPage class of object repository
 - Create object for Login class of objectrepository by passing driver in the constructor
 - Login login = new Login(driver);
 - Create a constructor in the Login class of object repository
 - Access the methods using the object reference of Login class
 - login email() sendKeys("arun.selenium@gmail.com");

```
login.password().sendKeys("Second@123");
login.login().click();
```

Save Copy to Evernote

- Create a MyAccount class under objectrepository package

- Copy all the things from Login class and modify according to needs - [View code](#)
 - Create an object for MyAccount and access the method using object reference
 - MyAccount account = new MyAccount(driver);
 - Assert.assertTrue(account.account().isDisplayed());
 - Execute the Test and see everything is working fine
- #### 4. Page Factory
- Replace By locators in Login class with @FindBy - [View here](#)
 - Add this in the constructor
 - PageFactory.initElements(driver,this);
 - Repeat the same for MyAccount class
 - Execute the Test and see everything is working fine

Autolt

1. Search 'Download Autolt' and download it
2. Install Autolt
3. Create a Project and write the Selenium code to click on the 'Choose File' button
4. Go to C Drive - Program Files (x86) - Autolt - SciTE-Lite Editor
 1. Inspect options using Au3Info
 1. ControlFocus("Open","", "Edit1")
 2. ControlSetText("Open","", "Edit1", "C:\\Users\\arunm\\Desktop\\ExcelTestData.xlsx")
 3. ControlClick("Open","", "Button1")
5. Save the file and compile it to generate exe file
6. Write the remaining code:
 1. Thread.sleep(5000);
 2. Runtime.getRuntime().exec("D:\\SeleniumTrainingWorkspace\\AutoltDemoProj\\autoifffile\\FileUpload.exe");
7. Execute the Selenium Code

By,
Arun Motoori

[Terms of Service](#)

[Privacy Policy](#)

[Report Spam](#)