Minnwest Message Module Automation by using Selenium, TestNG, POM

POM: POM is a pattern and not a framework. We can use this pattern in TestNG and Cucumber etc which is helpful in different ways.

POM is designed pattern in selenium that creates the object repository for storing the web elements and helps in reducing the code duplicating and improve the test case maintenance and to simplify we create the separate class file for each web page. This class file contains of different web elements on the web page. Moreover, the test scripts then use these elements to perform different actions.

POM Project structure:

1. Objects package: All the reusable files have been kept in Objects package.
2. Test scenarios package: Created
3. It’s a Maven project so I have integrated the dependencies of selenium, TestNG and Webdriver manager to POM.xml
4. Create the constructor in Objects package under Login class.
5. In page object model constructor(Instance of the class and similar to method) is used to do certain operations. So instance is created and constructor is loaded and in this webdriver is pointed to the constructor and this webdriver instance which are there in other locations the driver instance will be part of this constructor.
6. Before that we launched the web driver here before the constructor.
7. In constructor we need to pass the web driver driver; in order to point out the driver in other locations even.
8. To map my web driver to this object location we need to point using THIS keyword.
9. After this we will start our actual locators finding.
10. Using By we need to map any object.
11. Then with By we need to locate the xpath.
12. Then create one method to call this login xpath and give send keys or something else.
13. We will parameterize in the method itself as (String logindetails) and same in the send keys even.
14. Now in test scenario we need to call a method. Create the class Test methods.
15. Call @Beforetest in that test scenario and mention public void beforetest() and in beforetest we need call my webdriver to be created and to create the webdriver we need to create the outside which is global to my test method. As we are using the webdriver manager there is no need to give system.setproperty .exe file in the test method.
16. Next using driver instance we will create driver = new chromedriver( )
17. After this I have launched the browser.
18. Then we can call how much time the page needs to wait by implicit by giving the minutes and Timeunit.seconds.
19. Then we need to create the object for this spotlightpage class. As soon as we create object we can create by method level or global level. So now I am creating by method level and import the class.
20. Then create the test methods .
21. Then we need to call the test method in the
22. Base class
23. Page objects
24. Utilities
25. Extent report

Page Factory model helps POM model to implement better way.

It basically have 2 operations.

It helps in passing the driver. But in Java we need a utility known as Page factory which helps in passing the driver as well and initializing of elements on the objects of the page now.

In POM we always start with interfaces.

Maven Project

Implementirs Pare

Optionally Running Toss

Running Failed Tests srain

Reading Data from XLS

Scroonshots in Recorts

Logs m reports

Implementire GRiD

Deploy Project to Github

Scheduling Running of Automation Script:

Auto Running Project from Jenins