Automation Project (Nopcommerce)

Programming Language : Java

Automation Tool : Selenium

Selenium Framework : Hybrid framework

Version Control : Git

Java Testing Framework : TestNG

Phase-1: Implementation

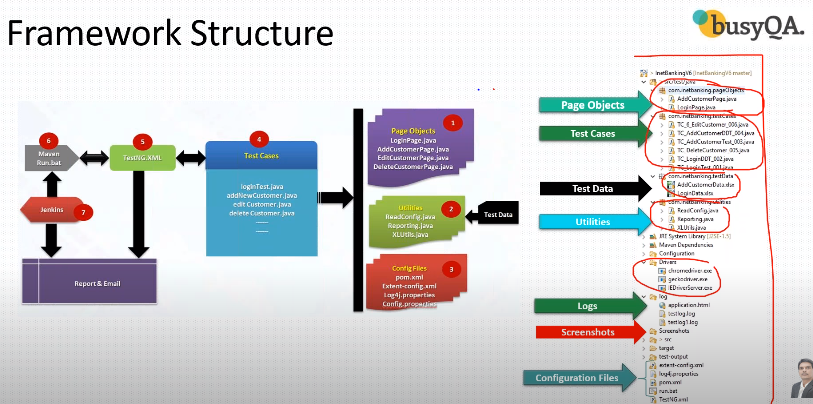
* Create Maven Project
* Update pom.xml
* Create Page Objects
* Add logs to the test case
* Read common values from properties file
* Run test cases on desired browser
* Add extent report
* Create Data Driven test case
* Adding new test case

Phase-2: Execution

* Run test cases with Maven pom.xml
* Run test cases through Maven CLI(Command Line Interface)
* Run test cases using run.bat
* Run test cases using Jenkins(Using bat files)

Phase-3: Maintenance

* Creating repository in Github
* Commit the project code in local repository
* Push the code to Github remote repository from local GIT repository



Phase-1: Implementation

Maven: Maven is Plugin just like a Testng, Which is available in Eclipse. Maven project is just like Java project but Maven provides some extra features like pom.xml file, in this file we can add dependencies, when we add these dependencies that will download automatically required jars, webdrivers, extent report, apache, etc. And we don’t need to update as time passes maven will automatically update all those.

Jars files like : Webdrivers, xl,

We can download dependencies from this website : <https://mvnrepository.com/>

What Test Case Contain:

Setup():Creation of all pre-requistes like set property, driver objects. This should be execute @BeforeClass

Test():actual test case

TearDown(): close the browser activities. This should be execute @AfterClass

Instead of creating above(Setup, Teardown) for every test case we just create BaseClass(Setup, Teardown).

Generating Log Files in Selenium:

**How to common values from Properties file:**

Properties file :

In BaseClass we should not hardcode anything like application URL, username, password. To maintain all the common variables we create properties file. The name of the file is config.properties file, config is a name of the file and .properties is extension of the file it just like a text file. This file is part of “Configuration” folder.

Utility File :

To read the values from config.properties file we have create one more file called ReadConfig.java(Utility Class), This is simple java class which contains methods. This java class act as mediater between “Properties file” and “BaseClass”. BaseClass will get variables form ReadConfig file. This Java class we have create under Utilities package.

In ReadConfig.java we have to create constructor to load the file and methods to read the variables

BaseClass:

In BaseClass we have to create object of the ReadConfig.Java class and call all the methods in the ReadConfig.Java class.

**Run test cases on desired browser**

To run TestCase on desired browser we need TestNG. We need to create XML file for browser selection after that we need to slightly modify the BaseCLASS.

We just need to add parameter annotation in BaseCLass.

How to create XML file

* Right Click on project name
* Select TestNG in last option
* Select Convert to TestNG

**TestNG report**

In our project we used TestNG report.

We need to create one xml file and run the xml file and just refresh the project the folder will be generate, just open and click on TestNG.html file. The report will open.

**Create Data Driven test case**

What is Data Driven Testing Framework in Selenium? Data Driven framework is used to drive test cases and suites from an external data feed. The data feed can be data sheets like xls, xlsx, and csv files. A Data Driven Framework in Selenium is a technique of separating the “data set” from the actual “test case” (code).

For Data Driven Test we need

Apache poi jars [poi, poi-ooxml, poi-ooxml-schemas, xml beans, commons-collections4]

XLS file(test data)

XLUtils.java ( which contains user define method to get the no.of cells(Excel))

We TestNG @DataProvider annotation.

First we read the file for that

we create instance of the file and create input stream and go to the workbook.

Jar files contains so many classes and methods.s

What is Apache POI?

1. Apache POI is an API provider by Apache foundation which is a collection of different java libraries.
2. These libraries give the facility to read, write and manipulate different Microsoft files such as excel sheet, power-point, and word files.

Apache POI has 4 Interface and classes:

1. Workbook and XSSFWorkbook
2. Sheet and XSSFSheet
3. Row and XSSFRow
4. Cell and XSSFCell

DDT follows steps

1. Create XL utilities file
2. In Test case use data provider annotation

**GIT & Github**

Git is a “local repository” and Github is “remote repository”.

If you “add” the code that will move into the Staging area and if you “commit” the code that will move into the GIT, if you PUSH the code that will move into the GIthub this process is called CHECKIN.

If you PULL the code from Github that will come into the Git and comes into the local machine, this process is called CHECKOUT.

<https://github.com/noorgangavaram/AutomationNopcommerce.git>

How to add project in the GITHUB.

1)git init // creates an empty git local repository

now we need to copy of the github repository url and connect with the git.

2) git remote add origin “url of the remote repository”. //This will connect git and github.

3)git add –A //add all files to the staging area

4) git commit –m “This is my first commit” // This will move into the git repository

5)git push –u origin master // ask you user name and password

6)git pull origin master // This will get the newfiles form github

# Problems I have faced

InvalidSelectorException: Compound class names not permitted

className="button-1 login-button". We can observer the space between the class name value because of that we are getting the exception. The solution for this, is just select the partial means half of the value “className="login-button".”