POM-PAGE OBJECT MODEL FRAMEWORK

Easy Maintenance

Reduce duplicate code

Increase the productivity

Maintain the uniformity of scripts

Can be able to integrate for better reporting

MAVEN PROJECT CREATION

1.File->New->Project->Maven->Maven Project->Click Next->Select “Create a simple project”->Click Next

2.Group Id: org.testleaf->Artifact Id: PageObjectModel->Version: 1.0.0 ->Click Finish

3.Add necessary dependencies in pom.xml

STEPS TO SETUP POM

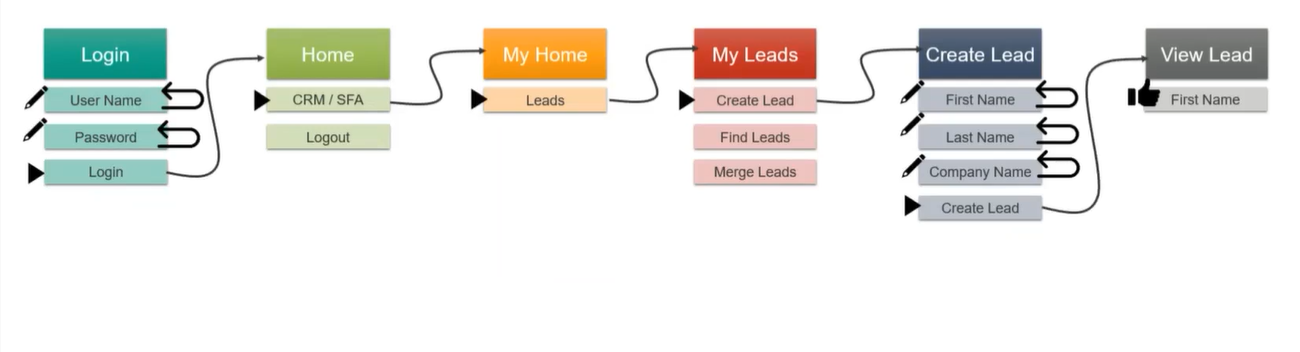
Step 1: Create a BaseClass Package to have common methods and variables of my project

-Create a class called ProjectSpecificMethods under Base Package

-Create init()🡪@BeforeMethod

-Create closeBrowser()🡪@AfterMethod

Pages that we are navigating: Login->Home🡪My Home🡪My Leads🡪Create Lead🡪View Lead



Step 2: Create Java classes for each page that we navigate in the application

Step 3: All the java classes should extend the Base Class/ProjectSpecificMethods

Step 4: Create methods for each actions and verifications in the respective pages

Step 5: All the methods should have return statement and return type

-If the control stays in the same page,then return this and return type should be class name

-If the control goes to another page, then return new ClassName() and return type should be ClassName() of landing page

Step 6: Create java class for each testcase with necessary steps

Step7: Testcase class should also extends BaseClass/ProjectSpecificMethods

STEPS TO INTEGRATE DATAPROVIDER:

Prerequisite:

Convert normal method into parameterized method,which ever method need data

Replace the hardcoded value into argumented variable

1.Excel documents🡪 under data folder

2.Program to read the data from excel

3.DataProvider method to get the data from ReadData.excelData() class

4. setup() with @BeforeTest to assign file name

5.Integrate dataProvider with @Test

POM for Parallel Execution

-Declare,

private static final ThreadLocal<RemoteWebDriver> remoteWebDriver = new ThreadLocal<RemoteWebDriver>();

Setter()🡪to set the value

//setter method for driver

Public void setDriver()

{

remoteWebDriver.set(new ChromeDriver());

}

Getter()🡪 to get the value

Public RemoteWebDriver getDriver()

{

return remoteWebDriver.get();

}

Soon after you are initialising webdriver you need to use setDriver();

So,wherever you are using driver---you must use getDriver()

**Reports that you so far used are,**

* TestNG Reports
* Cucumber Reports
* Extend Reports

**Features of Extent Reports—Version 3.1.5(Open source)**

* Give rich look with graphical representation
* Can be able to attach screenshot for each step
* Able to get reports based on author and category
* Able to get dashboard for automation summary

**Classes in Extent Report**

* ExtentHtmlReporter🡪 Used to create physical html report
* ExtentReports🡪 to capture the data for the report
* ExtentTest🡪to capture the status pass/fail
* MediaEntityBuilder🡪use to attach the screenshot

Step1: Set up physical report path

Step2: Create object for ExtentReports

Step3: Attach the data with physical file

Step4: Create a test case and assign test details

Sep5: Step level status

Step6: Mandatory step

Note: Step1 to Step3 are common for all testcases

//Set up physical report path

ExtentHtmlReporter reporter = **new** ExtentHtmlReporter("./reports/result.html");

//to keep report history

reporter.setAppendExisting(**true**);

//Create object for ExtentReports

ExtentReports extent = **new** ExtentReports();

//Attach the data with physical file

extent.attachReporter(reporter);

//Create a test case and assign test details

//testname, testdescription

ExtentTest test = extent.createTest("loginlogout", "test login functionality of leaftaps");

test.assignCategory("Smoke");

test.assignAuthor("Vino");

test.pass("Enter username");

test.pass("Enter password");

// ../ to come out of reports folder build() method is mandatory for MediaEntityBuilder

test.pass("click login button",MediaEntityBuilder.*createScreenCaptureFromPath*(".././snaps/desktop.png").build());

//Mandatory step

extent.flush();

STEPS TO INTEGRATE EXTENT REPORT:

1.Create startReport() with @BeforeSuite in ProjectSpecificMethods and add step1 to step3

2.Make ExtentReports as global variable

3.Create stopReport() with @AfterSuite to have extent.flush()

4.Cereate testCaseDetails() for test case details code

5.Declare ExtentTest test as global variable