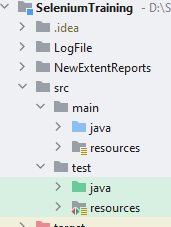
**TestNG Framework**

**Intro: -**

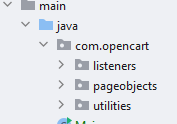
We are using maven as build automation tool for automatically downloading extra libraries and plugins from the central repository. It is a simple quick start archetype project and consists of below folders,

1. src / main / java
2. src / main / resources
3. src / test / java
4. src / test / resources



***In the src / main / java we have three different packages namely,***

1. listeners
2. pageObjects
3. utils



**Listeners**:- Listeners are TestNG annotations that literally “listen” to the events in a script and modify TestNG behavior accordingly. These are mainly used to take screenshot of the failed testcases and generate extent reports or retry failed testcases.

1. **ITestListener:-** This listener interface consists of methods like onStart, onFailure, OnTestStart, onTestSkipped, onTestSuccess and onTestFailure. Here in onTestFailure method we add the screenshot code to capture screenshot whenever any of the executing testcase is failed.
2. **IRetryAnalyzer:**- This listener interface consists of method called retry which will accept Boolean value. True to rerun the failed testcase and false to not execute the testcase. Here we can put number of times testcase need to be rerun.

**PageObjects:-** In this package we create a class file for each web page in the application. This class file consists of different web elements present on the web page and methods related to that webpage. In the test scripts we use these elements to perform different actions. Since each page's web elements are in a separate class file, the code becomes easy to maintain and reduces code duplicity. Examples:- LoginPage, DashboardPage and HomePage etc.

**Utils:-** In the utils package we different classes like Base, EnvironmentUtils, TestdataUtils, ObjectRepositoryUtils, ExcelUtils, SeleniumUtils and Constants etc.

***Base***:- The base class is a super class of all the test script classes here in the @BeforeSuite annotation we read the tenant (or) name environment from TestNG.xml file parameters and load the respective Testdata, Environment and Locator properties files.

In the @BeforeTest annotation we read the browser parameter provided in the TestNG.xml file and we will initialize the webdriver. Here we use webdriver manager to download the required browser executable before initializing the browser.

In the @AfterTest annotation we call the driver.quit() method to close all the browsers.

***EnvironmentUtils***:- In this class we have two methods like loadEnvironmentDetails and getEvironmentProperty. We call loadEnvironmentDetails method in the @BeforeSuite annotation of base class to load the respective property file of the given environment and getProperty method is used in the test scripts to get environment details like URL, Username and password etc.

***TestdataUtils***:- In this class also we have two methods like loadTestdata and getTestdata. We call loadTestdata method in the @BeforeSuite annotation of base class to load the respective property file of the given environment and getProperty method is used in the test scripts to get testdata details like error / Success messages, String etc.

***ObjectRepositoryUtils***:- In the object repository utils we have two methods like loadObjectRepository and getLocator. We call loadObjectRepository method in the @BeforeSuite annotation of base class to load the respective property file of the given environment and getLocator method to we return By Object of the given key. In the properties file we store like “loginButton=id:login” where we define locatory type and locator value for every key.

***ExcelUtils***:- In this class we have methods to read data from excel file. While creating the object of this class we pass the path to xlsx file and after that we can start using getSheet and getRowData method retrieve data from excel file.

***Constants***:- This class consists of static string values like *PROJECT\_HOME, ENVIRONMENT\_FOLDER\_PATH, TESTDATA\_FOLDER\_PATH* etc.

***Selenium Utils:-*** Here we implement all selenium related methods and we also handle the exceptions which we face while executing the selenium scripts.

***In the src / main / resources we have three different folders namely,***

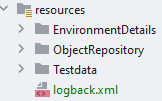
1. ***EnvironmentDetails*** :-

This folder consists of all the properties file of different environments which contains keys like URL, username etc.

1. ***Testdata***:-

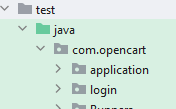
This folder consists of all the testdata properties files of different environments

1. ***ObjectRepository***:- This folder consists of properties file which have locators stored in it.
2. ***Logback.xml*** :- Here we define logging configuration.



***In the src / test / java ,***

In this folder we create different packages (Ex:- login and dashboard) for every module in the web application and we write Sanity and regression testscripts in different classes using TestNG.



***In the src / test / resources ,***

In this folder we create all the TestNG xmls to run the testscripts. We create a different testing xml for every module in the application like login.xml and dashboard.xml. We will also be having a Regression.xml file where all these xml files are called and this xml is used to execute all the testcases.

We have pom.xml file where we define all the exernal libraries and plugins which are required for the project.

*Reports Used :-*

1. Extent Reports
2. Test NG Reports

*Logger Type:-* SLF4J

*Deliverables:-*

* Extent reports
* Failure Screenshots
* Logger files

Sample Project is Avaialble below at

<https://github.com/OurAutomation/SeleniumTraining.git>

use the below command to clone the project in your workspace,

git clone https://github.com/OurAutomation/SeleniumTraining.git