## **NAGP Selenium assignment documentation**

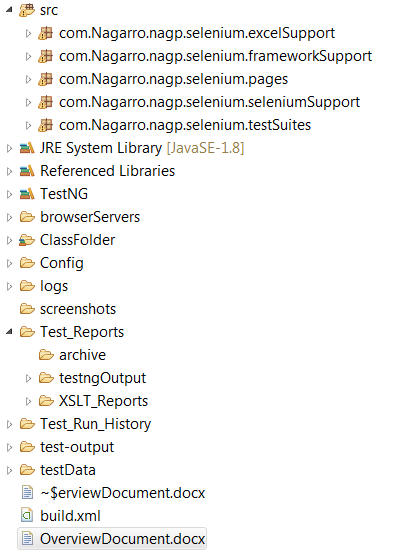
Created By RahulGandhi -4470

Purpose

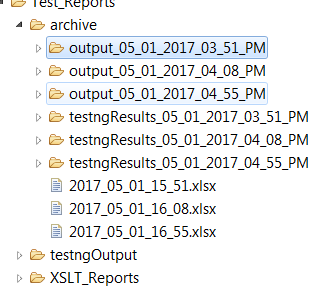
Purpose of this document is to describe folder structure of Framework/assignment

Directory Structure

Below is directory structure of project



* ***Src*** *: contains code in different packages*
  + ***com****.Nagarro.nagp.selenium.excelSupport*
    - ***This package contains classes related to excel reading and writing***
  + ***com****.Nagarro.nagp.selenium.frameworkSupport*
    - ***This package contains classes related to automation test framework, major tasks performed here are related to setup and teardown of suites.***
  + ***com****.Nagarro.nagp.selenium.pages*
    - ***This package is to make page object model design pattern****. It contains classes of each page of application and each page class contains object and business related function of that page*
  + ***com****.Nagarro.nagp.selenium.seleniumSupport*
    - ***This package contains functions related to selenium webdriver. E.g. open browser, click\_webelement etc.***
  + ***com****.Nagarro.nagp.selenium.testSuites*
    - *This package contains test cases and data provider functions*
* ***browserServer****: contains exe of browser servers*
* ***Config****: contains all configuration e.g. testing.xml, log4j.xml and testconfiguration.properties*
* ***Logs****: this folder contains all logs, daily new log file is generated.*
* ***Test\_Data****: contains and excel file which have test data*
* ***Test\_Run\_History****: contains results of test runs as suggested in assignment*
* ***Screenshots****: this folder contains all screen shots(incase test case is failed screen shot is captured automatically)*
* ***Test Reports****: this folder has three sub folder*
  + *Archive contains all historical test reports of both testing and XsltReports*



* + *testngOutput<timestamp>: contains all testing results*
  + *XSLT\_Reports: contains all XSLT reports created using testing reports*
* ***Build.xml :*** *build.xml contains all ant tasks*

***Following are targets in sequence of dependency***

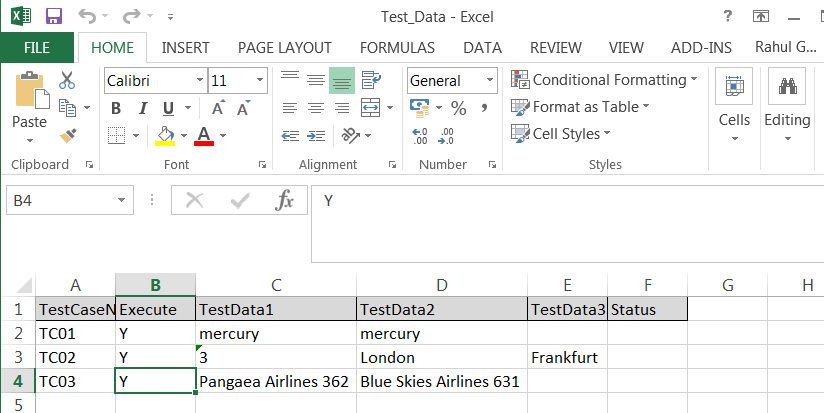


**Assignment Requirements**

Create a Page Object Model Framework using TestNG to implement these test cases. The framework should have following features –

1. Test Data should be read from excel file

frameworkSolution: Test data is stored in “testData\Test\_Data.xlsx”



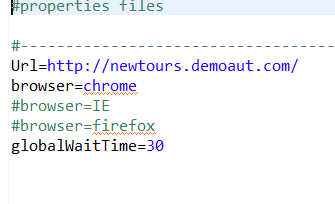
1. There should be a flag with the name “Run mode” in the Test Data file against each test case name which will drive if the given test case has to be executed or not (with value “Y” or “N”)
2. Implement Log4J for logging

Log4j is implemented using log4j.xml this file is present in config folder



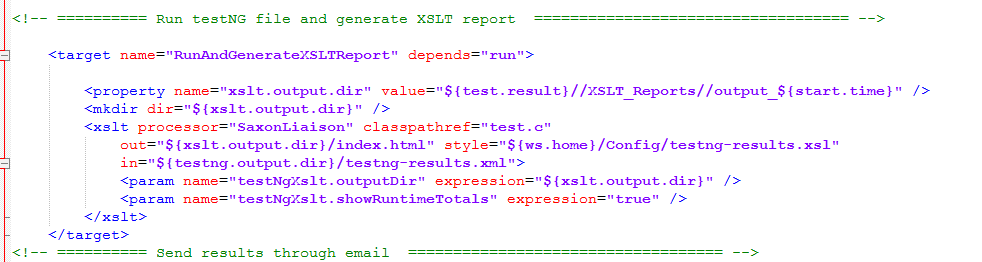
1. Use Ant for compiling and executing the tests
2. Implement proper waits.
3. The global configuration values like browser name, test site url, global wait value etc. should be read from a properties file.

File is located at *config\\testConfiguration.properties*



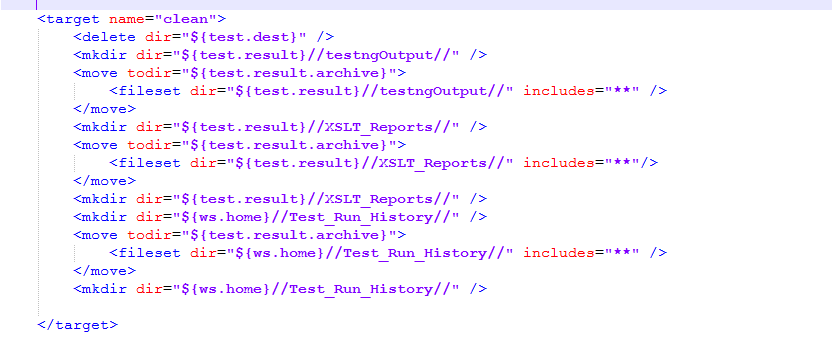
1. The test should run on following browsers IE, Chrome, FF
2. Put proper assertions with error description and if any test cases fails take a screenshot with the name same as test case and appended by a brief description of error in the screenshot file name ( For e.g. TestCase1\_Invalid\_Credentials)
3. Create xslt report. Customize the report to append error screenshots in the xslt report

//These are generated using ant



1. Also create a Test Status file with the Pass/Fail status of each test case.
2. On the beginning of a new test execution archive the previous test run results in a folder(Both XSLT and Test Results excel file). Name of the folder should reflect the date time of the run

This is done through ANT task



1. When the test ends email the Test Results excel to the stake holders

