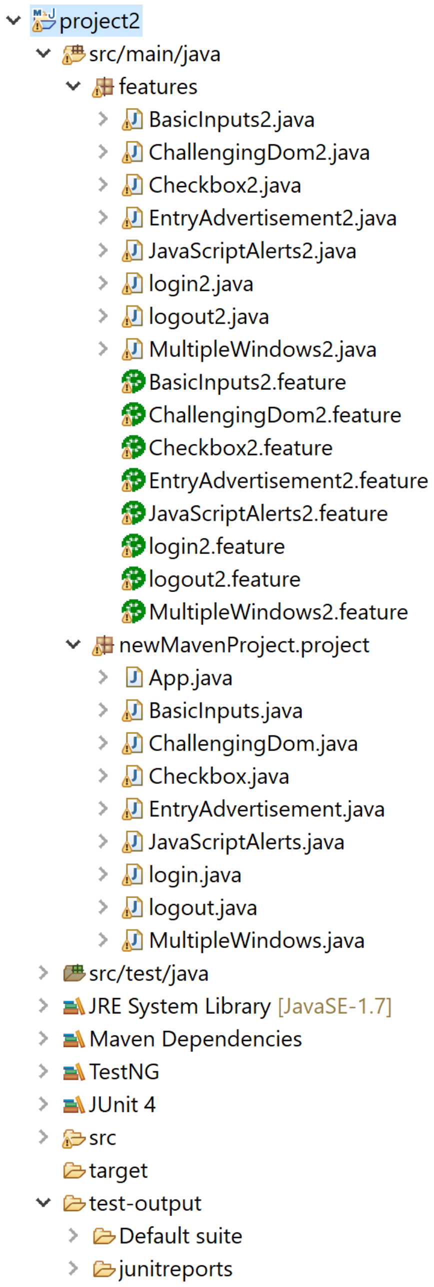
Automation project instructions

This automation framework project is written entirely in Java and made using Eclipse IDE.
The BDD is written in Cucumber.
The project structure looks like this



```
project2
  src/main/java
    features
      BasicInputs2.java
      ChallengingDom2.java
      Checkbox2.java
      EntryAdvertisement2.java
      JavaScriptAlerts2.java
      login2.java
      logout2.java
      MultipleWindows2.java
      BasicInputs2.feature
      ChallengingDom2.feature
      Checkbox2.feature
      EntryAdvertisement2.feature
      JavaScriptAlerts2.feature
      login2.feature
      logout2.feature
      MultipleWindows2.feature
    newMavenProject.project
      App.java
      BasicInputs.java
      ChallengingDom.java
      Checkbox.java
      EntryAdvertisement.java
      JavaScriptAlerts.java
      login.java
      logout.java
      MultipleWindows.java
  src/test/java
  JRE System Library [JavaSE-1.7]
  Maven Dependencies
  TestNG
  JUnit 4
  src
  target
  test-output
    Default suite
    junitreports
```

There are in total automation scenarios, these are

Login
Logout
Basic Inputs
Challenging DOM
Checkbox
Entry Advertisement
Javascript Alerts
Multiple Windows

The class logics are in individual .java class files
The cucumber scenarios are in individual .feature files under the features package.
There are two packages here
- features
- newMavenProject

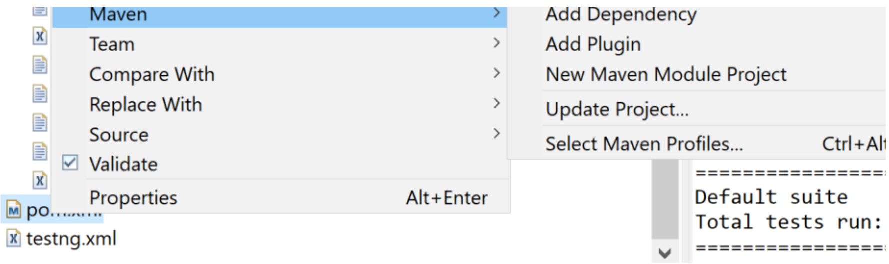
This entire project is written in two different ways
1st way - It is written is as an official Java framework with classes (i.e. step definition files) which links to each feature files, this is all under the under the features package.
2nd way - The second way the project is written is directly without feature files but more in a procedural approach, in other words just individual test classes that can be executed one after the other as TestNG. This is under the under the newMavenProject package.

The project should be run using the second way as I have not been able to link the cucumber feature files with the .java step definition files due to IDE issues causing unknown errors, therefore I had another package as newMavenProject with only the automation logic included which would do the exact same thing but without the cucumber feature file.
However I have still included the first way to show that the framework can be implemented in both ways.

The java files (step definitions) are	**Its corresponding cucumber feature files are**
BasicInputs2.java	BasicInputs2.feature
ChallengingDom2.java	ChallengingDom2. feature
Checkbox2.java	Checkbox2.feature
EntryAdvertisement2.java	EntryAdvertisement2.feature
JavaScriptAlerts2.java	JavaScriptAlerts2.feature
login2.java	login2.feature
logout2.java	logout2.feature
MultipleWindows2.java	MultipleWindows2.feature

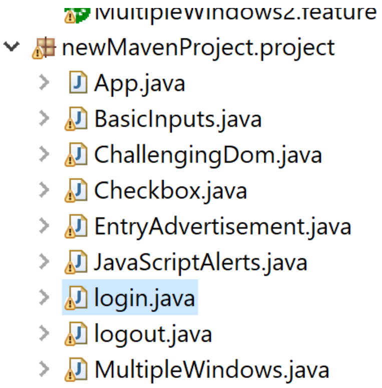
Instructions about how to use the project

- Step 1 Ide Ecplise is needed preferably
- Step 2 Use the **pom.xml** to download all the dependencies like .jar files of the project otherwise it shall not work.
  Either open the pom file in the IDE and press save
  Or (if you are using Eclipse IDE) right click on the pom file - maven and click update project

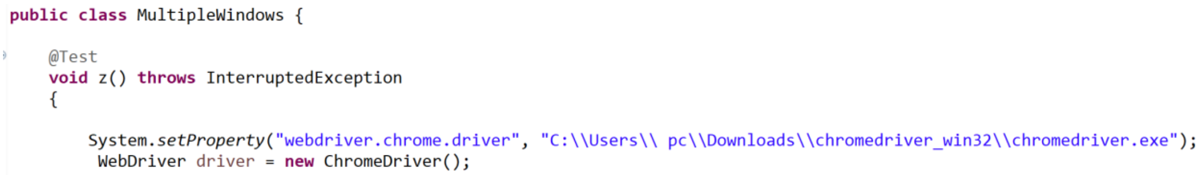


```
================
Default suite
Total tests run:
================
```

- Step 3 Once all dependencies are downloaded in the project a webdriver needs to be used to open Google Chrome browser.
  This is in the folder called **chromedriver_win32.zip**
  Extract this .zip file in e.g. desktop
- Step 4 Then in every of these class files



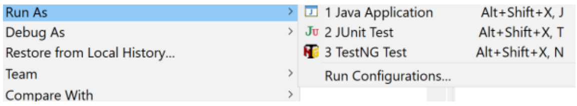
the chrome driver's location in mentioned e.g.



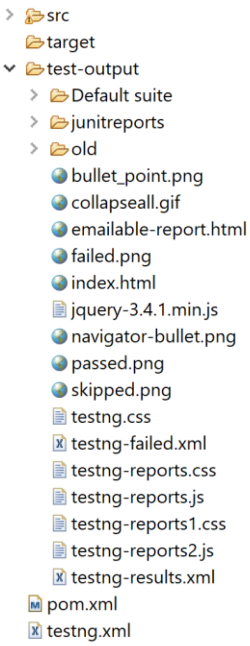
```
public class MultipleWindows {

    @Test
    void z() throws InterruptedException
    {

        System.setProperty("webdriver.chrome.driver", "C:\\Users\\ pc\\Downloads\\chromedriver_win32\\chromedriver.exe");
         WebDriver driver = new ChromeDriver();
```

On each of these classes this path needs to be changed to the location where the extracted chromedriver is located in your machine.

- Step 5 please ensure that you can run the .java files as a TestNG test
- Step 6 under the package newMavenProject you can either execute each of the .java files individually one after the other as TestNG, or you can right click on the package and run all of those as TestNG Test.

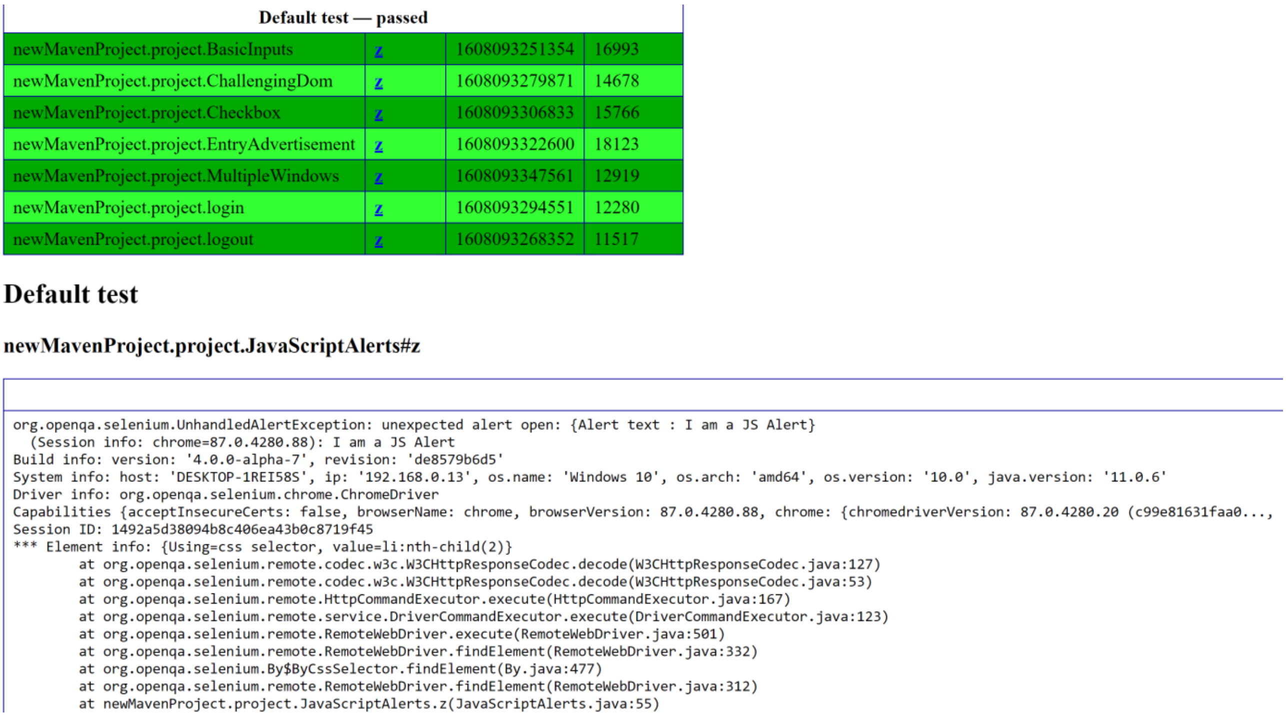


This shall produce a report in html format called "emailable-report.html" under the folder test-output



```
> src
  target
v test-output
  > Default suite
  > junitreports
  > old
    bullet_point.png
    collapseall.gif
    emailable-report.html
    failed.png
    index.html
    jquery-3.4.1.min.js
    navigator-bullet.png
    passed.png
    skipped.png
    testng.css
    testng-failed.xml
    testng-reports.css
    testng-reports.js
    testng-reports1.css
    testng-reports2.js
    testng-results.xml
  pom.xml
  testng.xml
```

If this file is opened after tests have been executed this shall show the list of all the classes that have been executed and which have passed and how many milliseconds it took for execution.
An example of an execution session was



Default test — passed			
newMavenProject.project.BasicInputs	z	1608093251354	16993
newMavenProject.project.ChallengingDom	z	1608093279871	14678
newMavenProject.project.Checkbox	z	1608093306833	15766
newMavenProject.project.EntryAdvertisement	z	1608093322600	18123
newMavenProject.project.MultipleWindows	z	1608093347561	12919
newMavenProject.project.login	z	1608093294551	12280
newMavenProject.project.logout	z	1608093268352	11517

Default test

newMavenProject.project.JavaScriptAlerts#z

```
org.openqa.selenium.UnhandledAlertException: unexpected alert open: {Alert text : I am a JS Alert}
  (Session info: chrome=87.0.4280.88): I am a JS Alert
Build info: version: '4.0.0-alpha-7', revision: 'de8579b6d5'
System info: host: 'DESKTOP-1REI58S', ip: '192.168.0.13', os.name: 'Windows 10', os.arch: 'amd64', os.version: '10.0', java.version: '11.0.6'
Driver info: org.openqa.selenium.chrome.ChromeDriver
Capabilities {acceptInsecureCerts: false, browserName: chrome, browserVersion: 87.0.4280.88, chrome: {chromedriverVersion: 87.0.4280.20 (c99e81631faa0...,
Session ID: 1492a5d38094b8c406ea43b0c8719f45
*** Element info: {Using=css selector, value=li:nth-child(2)}
	at org.openqa.selenium.remote.codec.w3c.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:127)
	at org.openqa.selenium.remote.codec.w3c.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:53)
	at org.openqa.selenium.remote.HttpCommandExecutor.execute(HttpCommandExecutor.java:167)
	at org.openqa.selenium.remote.service.DriverCommandExecutor.execute(DriverCommandExecutor.java:123)
	at org.openqa.selenium.remote.RemoteWebDriver.execute(RemoteWebDriver.java:501)
	at org.openqa.selenium.remote.RemoteWebDriver.findElement(RemoteWebDriver.java:332)
	at org.openqa.selenium.By$ByCssSelector.findElement(By.java:477)
	at org.openqa.selenium.remote.RemoteWebDriver.findElement(RemoteWebDriver.java:312)
	at newMavenProject.project.JavaScriptAlerts.z(JavaScriptAlerts.java:55)
```