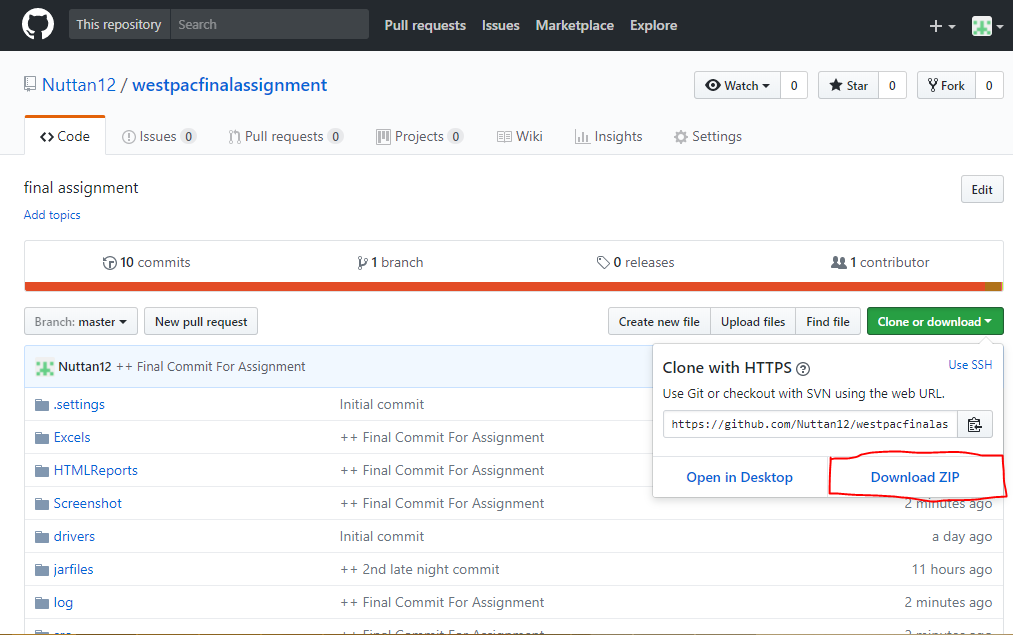
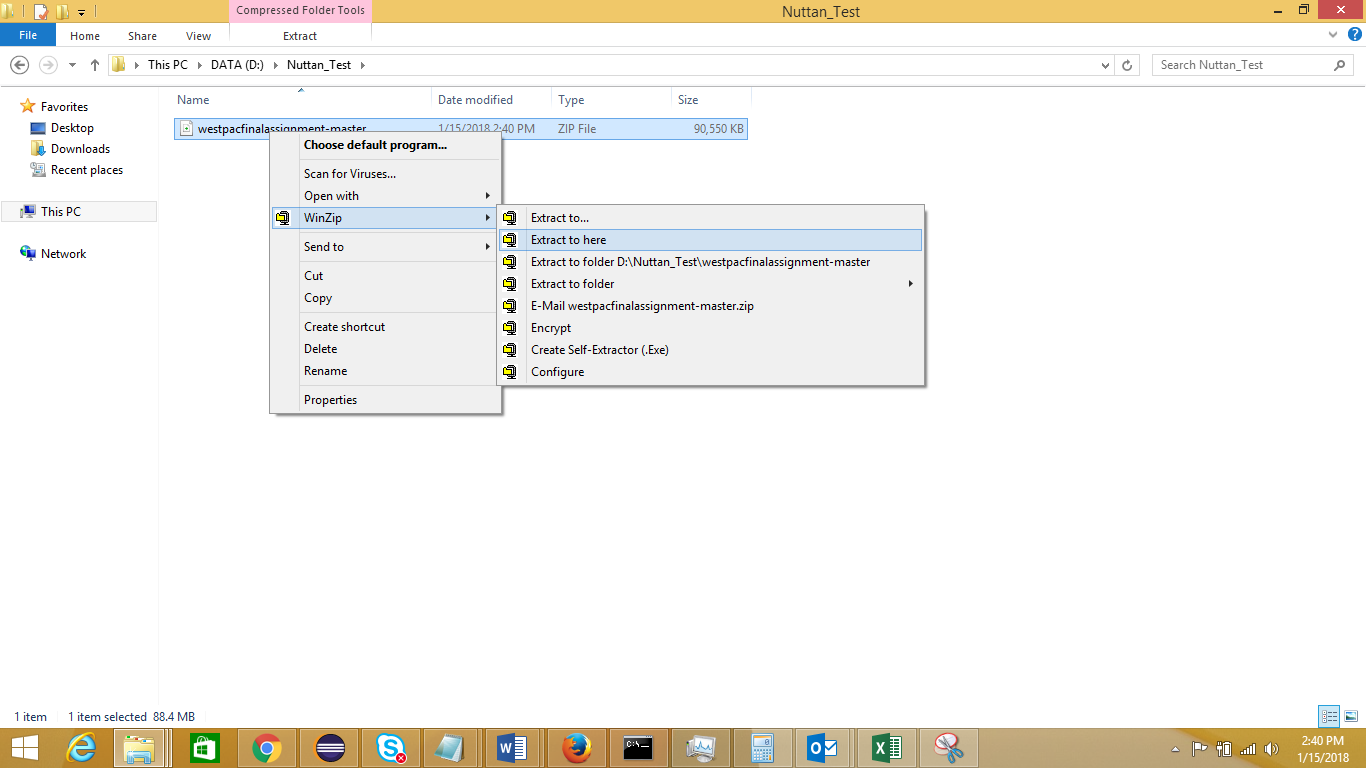
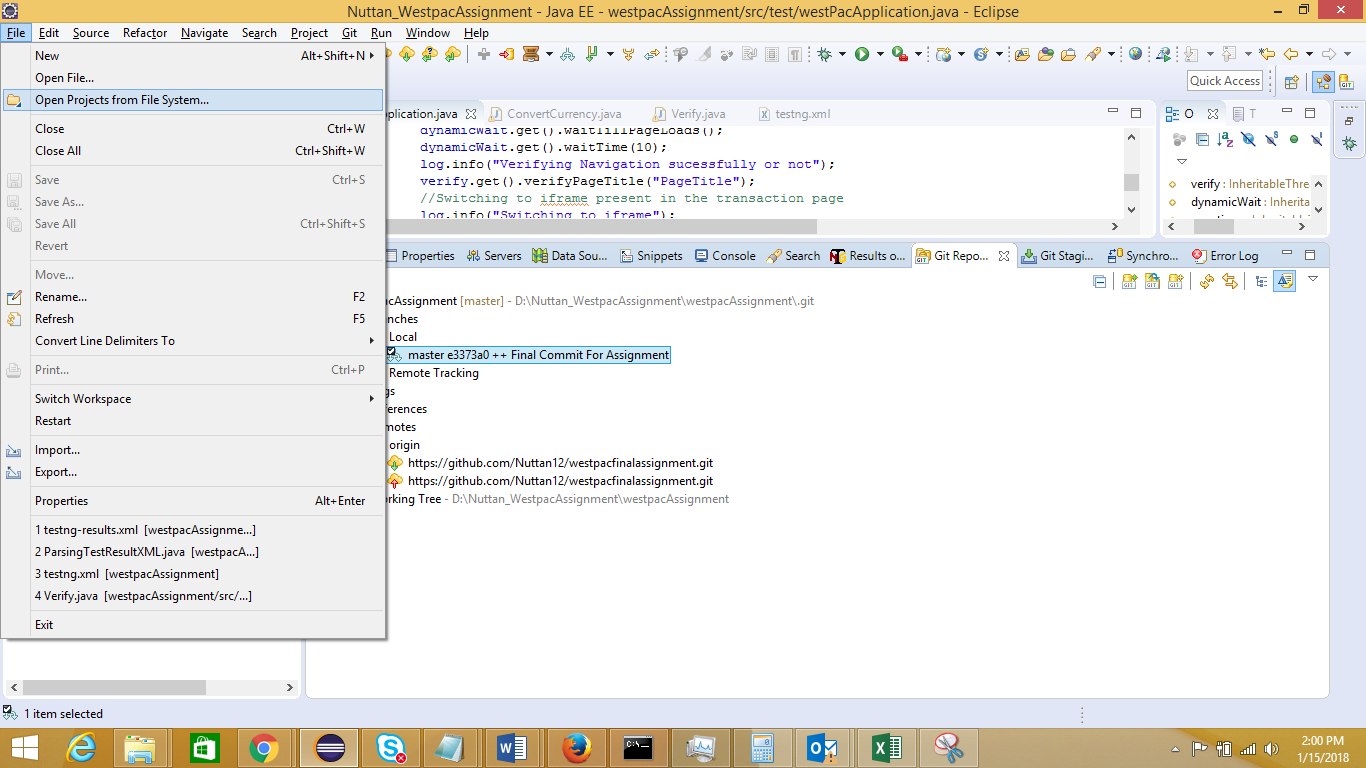
1. Browse URL https://github.com/Nuttan12/westpacfinalassignment



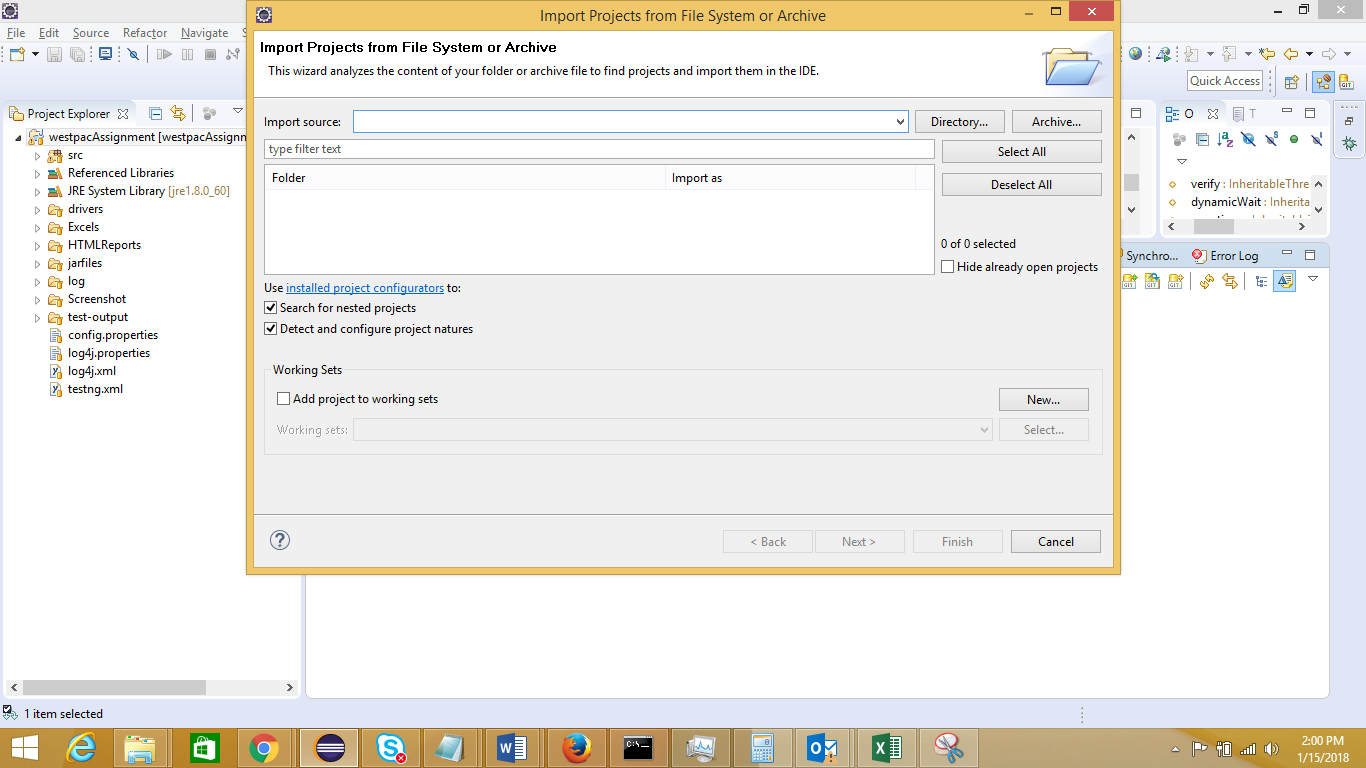
1. Place it in your convenient location ,in this case I have placed in D:/ ,then Unzip the folder



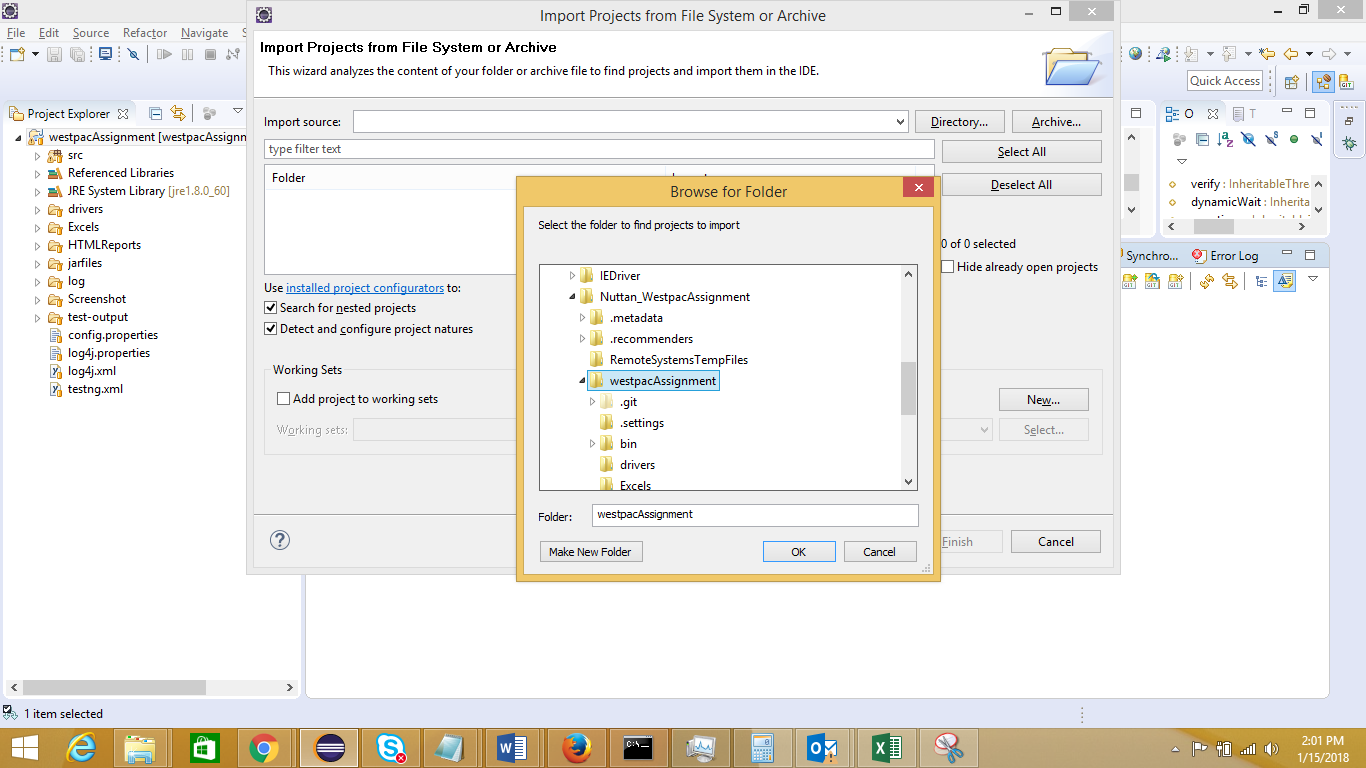
1. Open Eclipse IDE and click on “File” Menu ,then “Open Projects From File System”



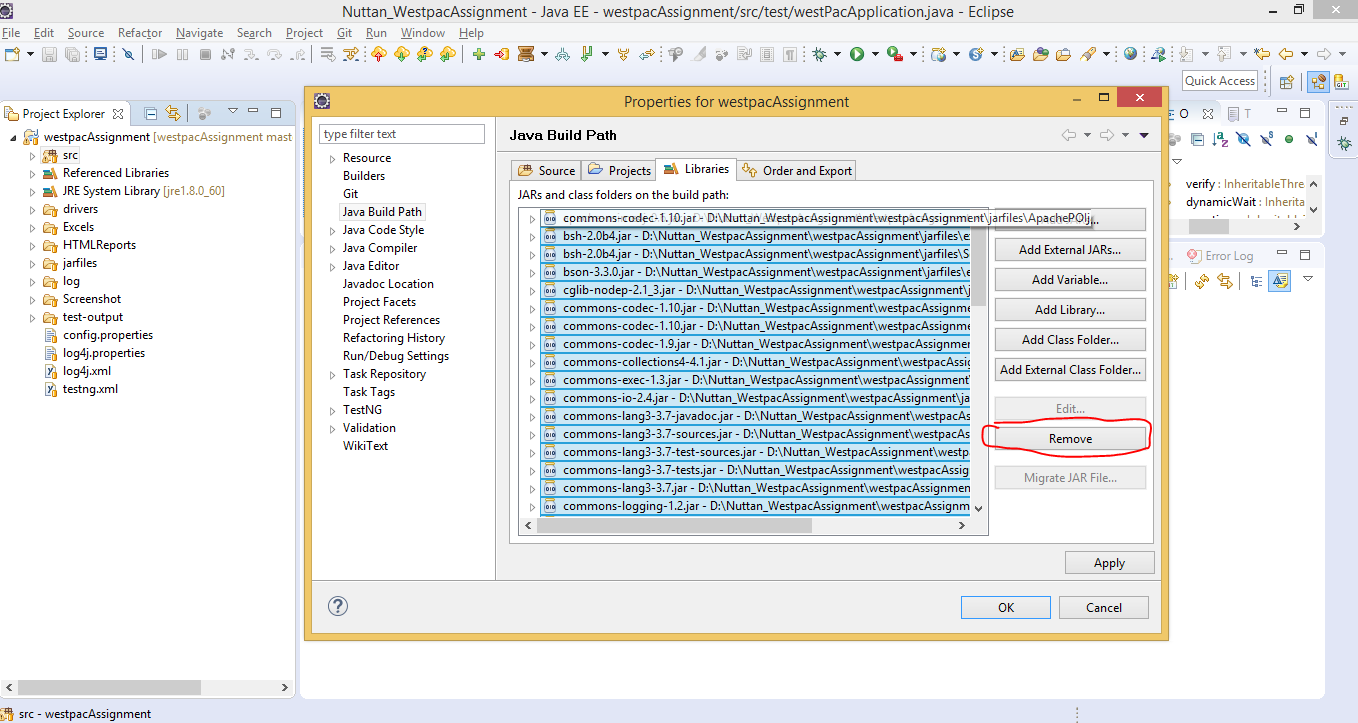
1. Click on “Directory” and Navigate to the location where you have unzipped the folder



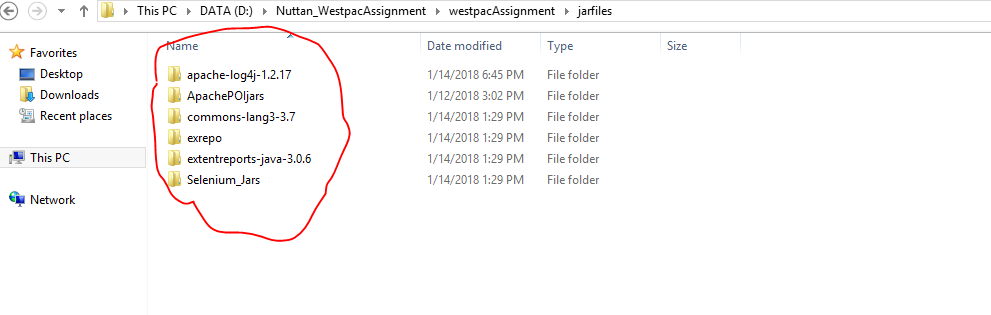
1. Point the correct project root folder and click on “OK” Button.



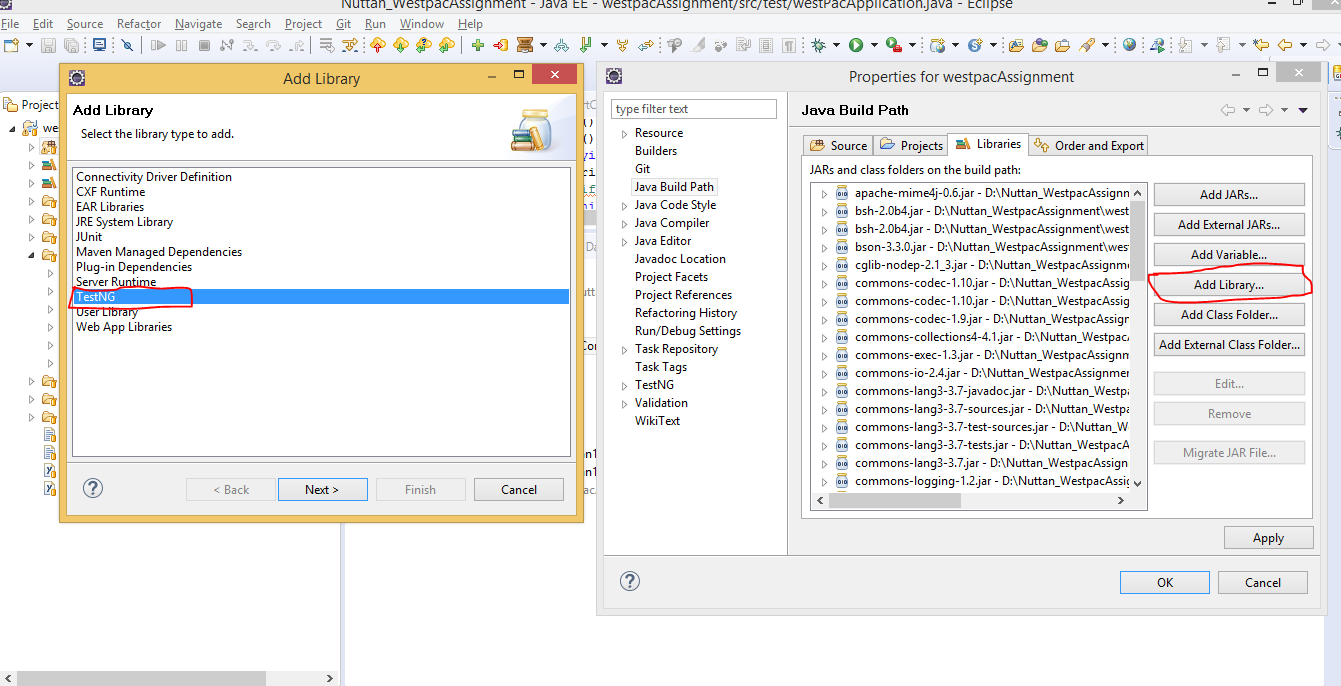
1. Right Click on “Src” and navigate to BuildPath->Configure Build Path ->Libraries. Remove the existing jar files (Selecting all CTRL+A) as these are still pointing to local path.



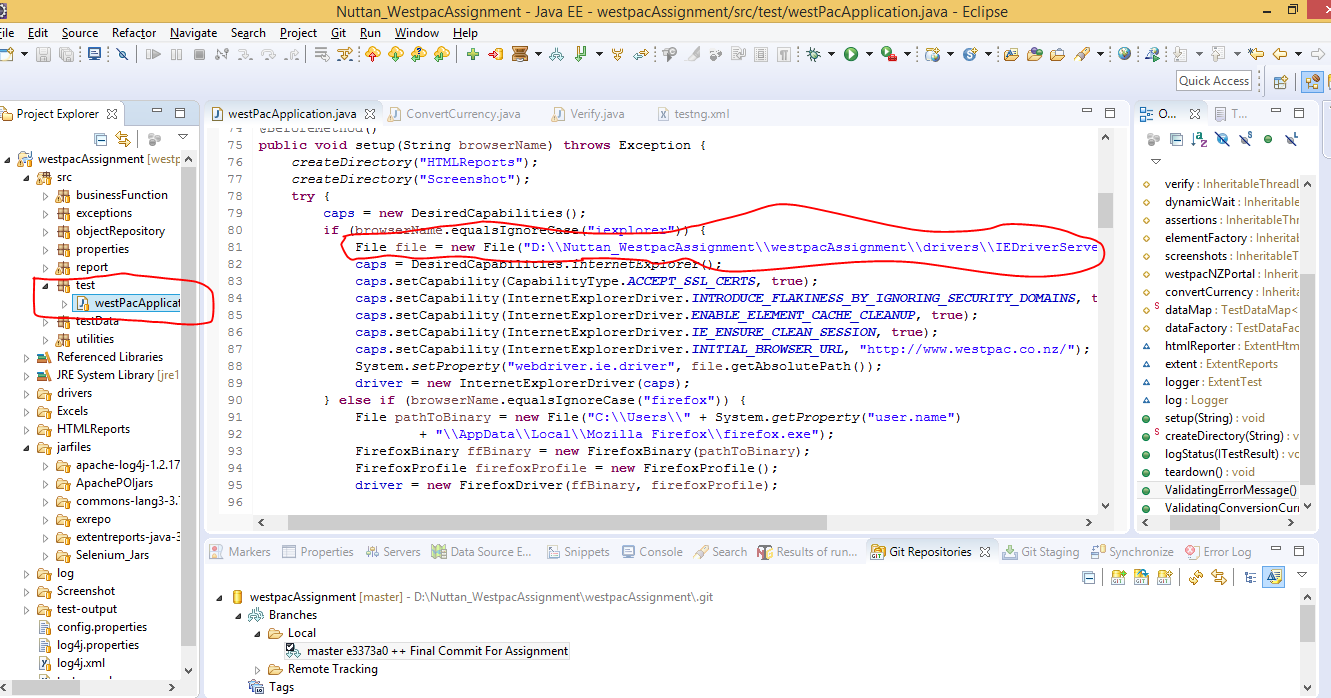
1. Click on “Add External JARs” and navigate to the project folder and all the Jars available in “jarfiles” folder. These are mandatory while running the Selenium Project.



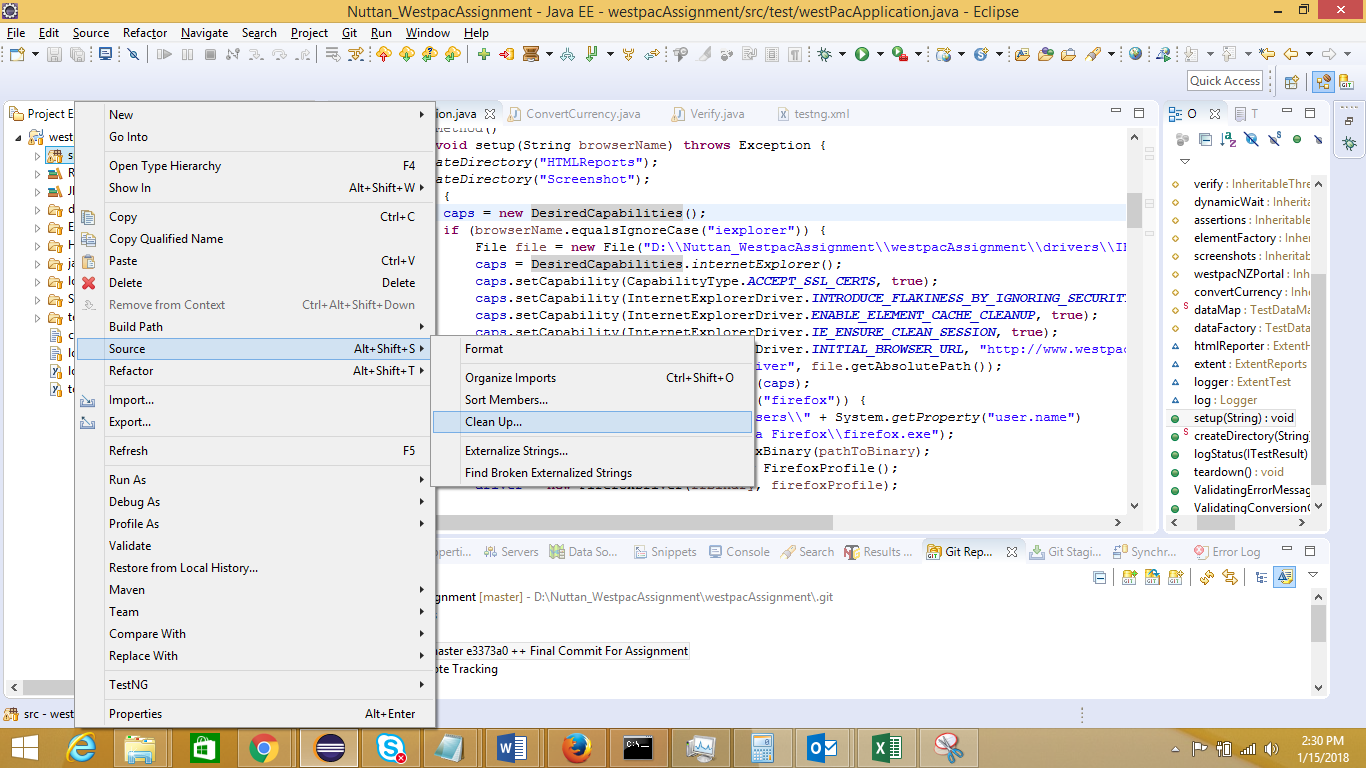
1. Click on “Add Library” and add “TestNG” library. In this case it is present in build with eclipse. Else download TestNG Version 6.9.9 Jar files and repeat step 7 as above.



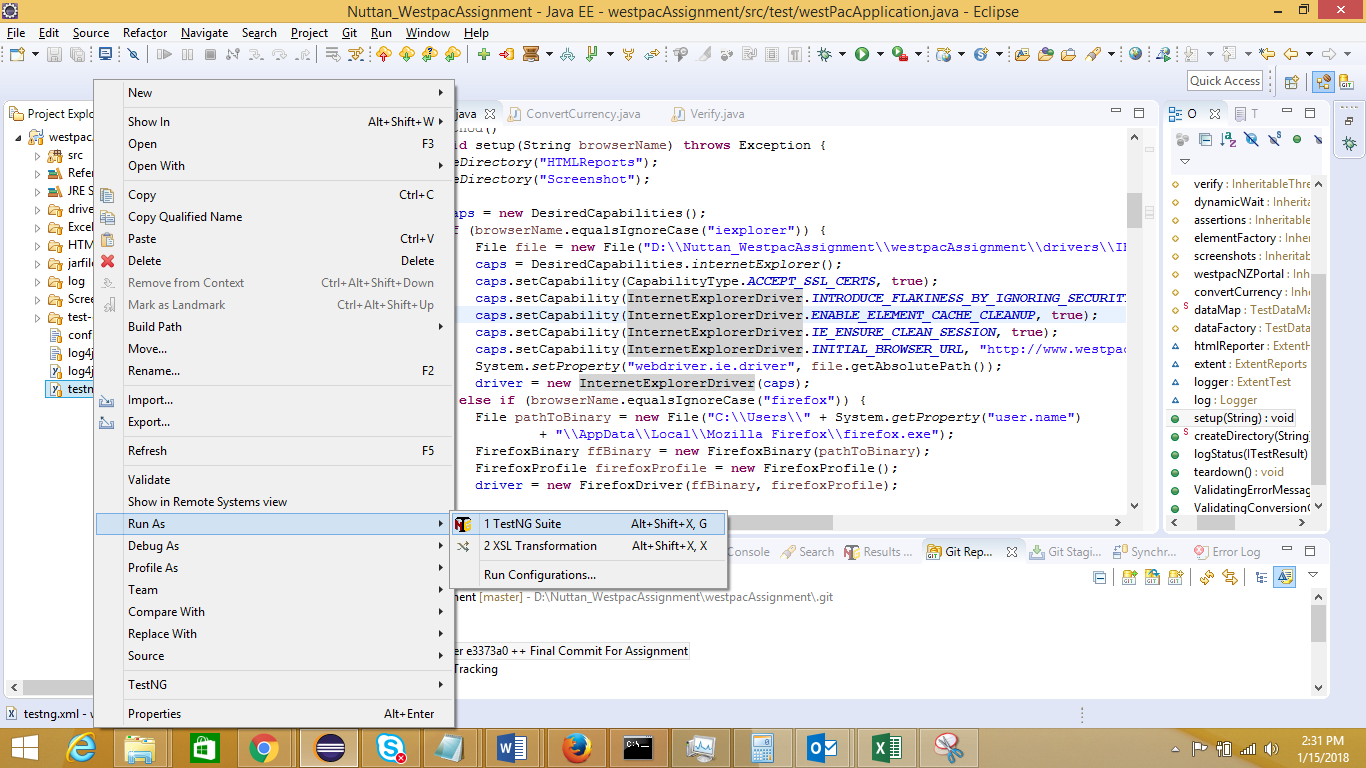
1. Go to Src -> test -> westPacApplication.java and change the driver path. [Point it to the recent downloaded folder path]. Drivers are placed under “drivers” folder. In this case for demo we are using IEDriver.exe



1. Right Click on “Src” and go to Source -> Clean Up



1. Right Click on “testng.xml” and Run As -> TestNG Suite



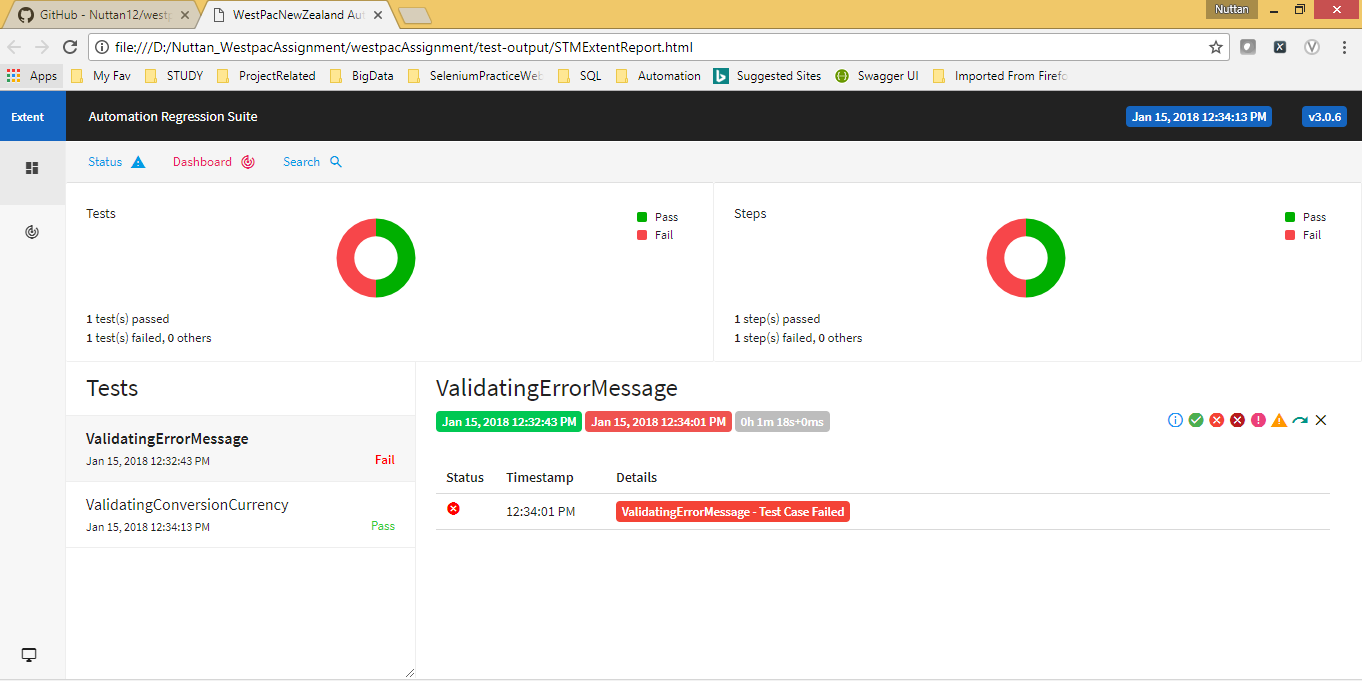
1. **Browsers**
2. Mozilla Firefox > Version 46.0
3. Google Chrome > Version 63.0.3239.84
4. Internet Explorer > Version 11.0.9600.18860
5. **Driver Servers**
6. Gecko Driver > We haven’t been used till now. The latest version is v0.19.1 for Firefox version 55 and above
7. Chrome Driver > Version 2.33
8. IEDriver > Version 3.4.0
9. **TestNG** > Version 6.9.9
10. **Selenium-Java** > Version 2.53.1

FYI: 2 types of report we can generate in this selenium project

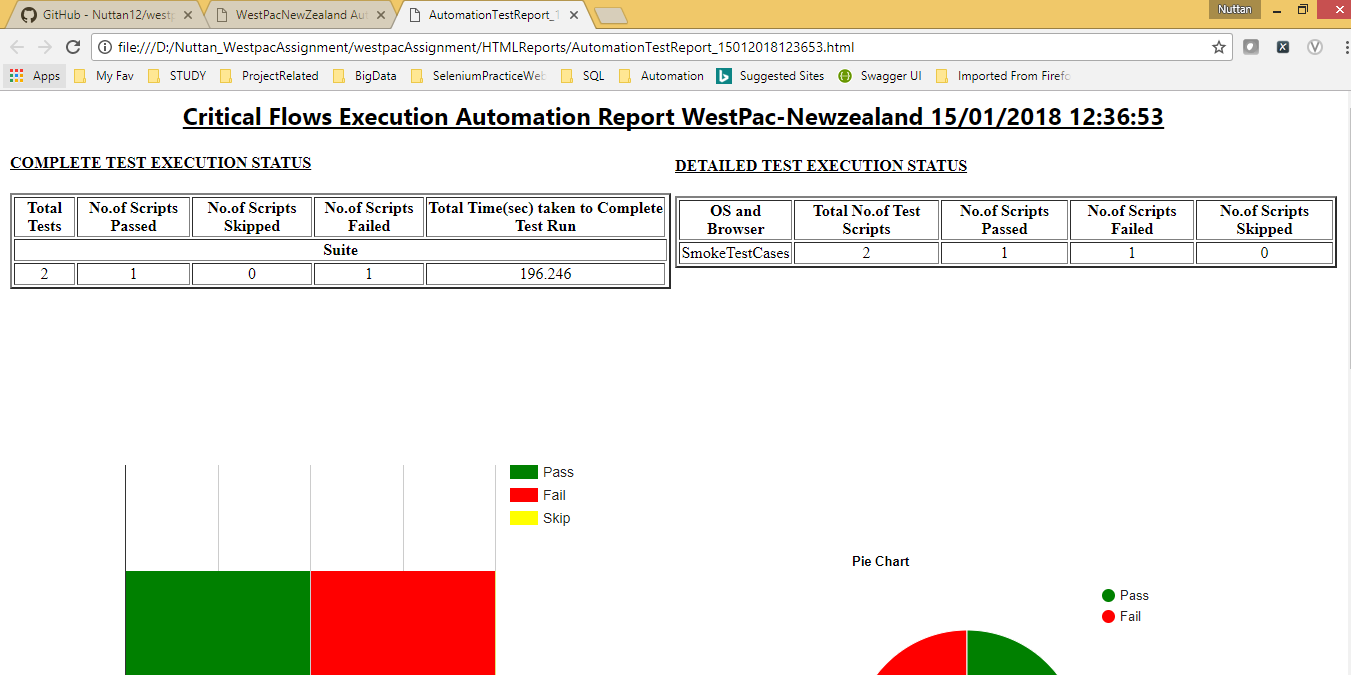
* Extent Report
* Customized HTML Report
* For Extent Report after each test run refresh test-output folder and open “STMExtentReport”
* For Customized HTML Report after each test run Go to Src -> report -> TestReport.java

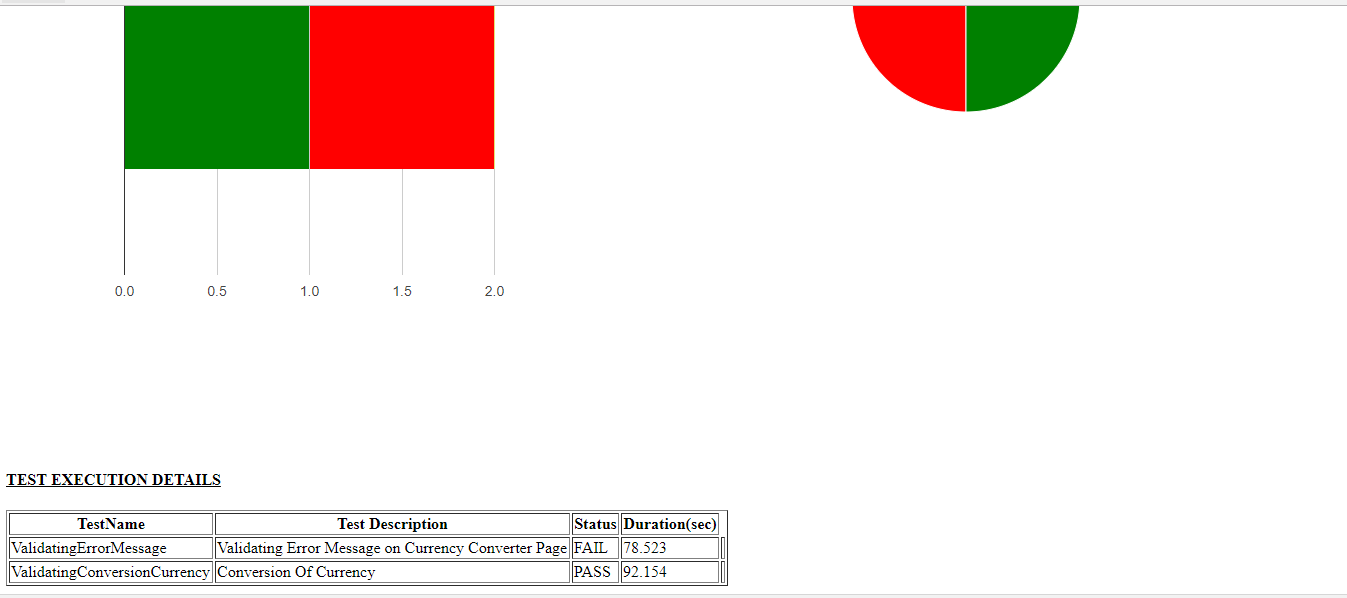
Right Click on “TestReport.java” and Run as “TestNG Test”. Navigate to “HTMLReports” Folder and check the latest HTML Report by time stamp.

**Extent Report:**



**Customized HTML Report**





## Question 1:

**Why did you choose this tool/framework? Explain if there are any other possible alternative with pros/cons of them.**

Ans: I have chosen Selenium Web driver (Version 2.53.1) with Java for automating the above user story in Westpac New Zealand website as it is basically a web application

**Pros**  
  
1. Selenium is pure open source, freeware and portable tool. We can get online support from various community.  
2. Selenium supports variety of languages that include Java, Perl, Python, C# etc.  
3. Selenium supports many operating systems like Windows, Macintosh, Linux, Unix etc.   
4. Selenium supports many browsers like Internet explorer, Chrome, Firefox, Opera, Safari etc.   
5. Selenium can be integrated with various Build tools like ANT or Maven

6. Selenium can be integrated with TestNG testing framework for testing our applications and generating reports.   
7. Selenium can be integrated with Jenkins or Hudson for continuous integration. 

### **Why Data Driven Framework**

Data Driven framework is focused on separating the test scripts logic and the test data from each other. Allows us to create test automation scripts by passing different sets of test data. The test data set is kept in the external files or resources such as MS Excel Sheets, SQL Database, XML files etc., and the test scripts connect to the external resources to get the test data. By using this framework we could easily make the test scripts work properly for different sets of test data. This framework significantly reduces the number of test scripts compared to a modular based framework.

Usually, we place all our test data in excel sheets which we use in our test runs. Assume, we need to run a test script (Say, login test) with multiple test data. If we run the same test with multiple test data sets manually is time-consuming, and error-prone.

### **Advantages of using Data Driven Test Framework**

* Re-usability of code
* Improves test coverage
* Faster Execution
* Less maintenance
* Permits better error handling

## Question 2:

**Do you think the priorities of the user story are appropriate in relation the overall need of currency converter?**

Ans: I don’t agree with the priorities of the user story assigned to them in relation to the overall need of currency converter. From end user perspective of a banking application, he/she will expect the functionality of currency converter to be working perfectly without any issues. So User story 2 carries more importance.

On the other hand User story 1 depicts negative test scenario which carries less importance in comparison to User story 2.

## Question 3:

**What are the possible points at which your tests can fail and how can you handle them?**

Ans: When we are navigating to Currency converter page from Home page, we have to wait till page loads completely. Because in Currency converter page we have to switch to iframe before doing any operations on the web elements present in that page. We may get below exceptions which will lead to test failure.

* Stale Element Exception
* No Such Element Found Exception
* Timeout Exception
* No Such Frame Exception

We can handle below exceptions in following ways.

* Fluent Wait
* Explicit Wait
* Waiting till page loads
* Refresh the browser as DOM will load again so finding the same web element will not be an issue
* POM and Page factory using @Cachelookup annotations

## Question 4:

**How can we drive your tests using any CI tool? Explain key things we need to bear in mind.**

Ans: CI is a methodology that caters to the building and testing of development work to identify and address the pain points before they become serious issues. Each time a product code is developed, it goes through an automatic CI process. There are several tools for Continuous Integration to help trigger these builds and tests.

**Key Points for a CI Tool to drive Software Development Life Cycle**

* Maintaining a Single Source Repository
* Automating the Build
* Everyone should be able to Commit to the Mainline Everyday
* Build Every Commit to the Mainline
* Keeping the Build Fast
* Making the Build Self-Testing
* Testing in a Clone of the Production Environment
* Make it Easy for Anyone to Get the Latest Deliverables
* Everyone can See What’s Happening
* Automating the Deployment to higher environment using CI stage build artifacts

Here we can use Jenkins integrated with Selenium Web driver to drive our tests.

* It is open source, flexible and extendable, makes it easy for the developer to initiate the builds.
* Jenkins has an ecosystem of plugins enabling us to use support projects in almost any language of choice, over large databases and in almost any version control systems
* Jenkins has evolved to be in perfect sync with Continuous Integration and Continuous Delivery (CD) pipeline
* Jenkins can be used for everything, right from source code management to Continuous Delivery.