**Framework Introduction:**

Language Used – java

Framework structure – TestNG and maven

Browser used - Google Chrome

IDE – Eclipse

JDK version - 1.8.0\_144

**Steps to Follow for setup:**

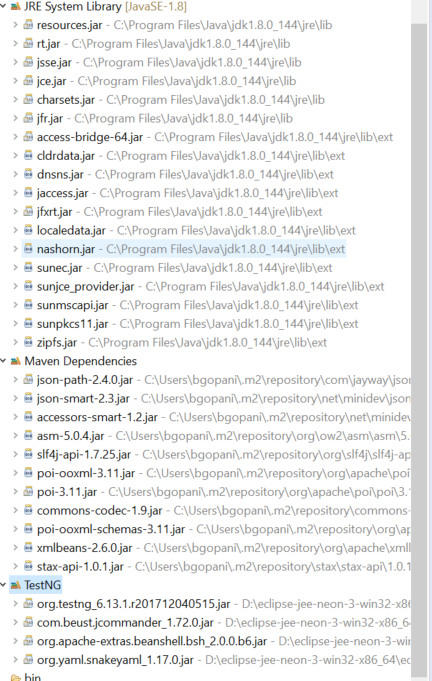
1. Import the project in your IDE. (folder path - Setup\api-assurity-v1.0)
2. Let’s maven dependencies downloaded. It would download all the neccesary jars, but if not I am adding all the jars as per folder.

Jars Needed:

You can configure all the jars from the folder Setup\Jars

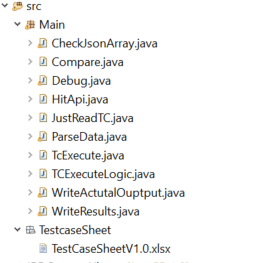
1. JRE System Library
2. Maven Dependencies
3. TestNG

Structure should look like below screenshot –



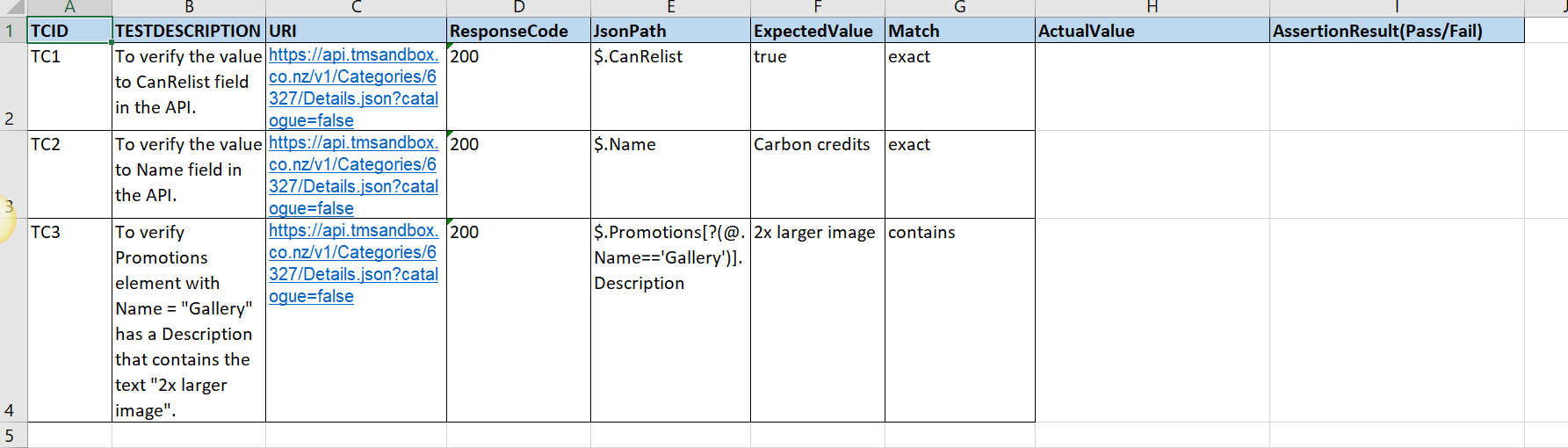
**How to use this framework?**

Package Structure:



**Usage:**

* In Test Case Sheet, user adds:
  + Test Description,
  + URI (API which he wants to hit),
  + Expected Response Code,
  + JsonPath (json path you want to extract data, in API),
  + Expected value (response which is expected),
  + match (you want to match exactly or just partial match, here I have coded for 2 of them for exact and contains, we can have many like begins with, range etc).



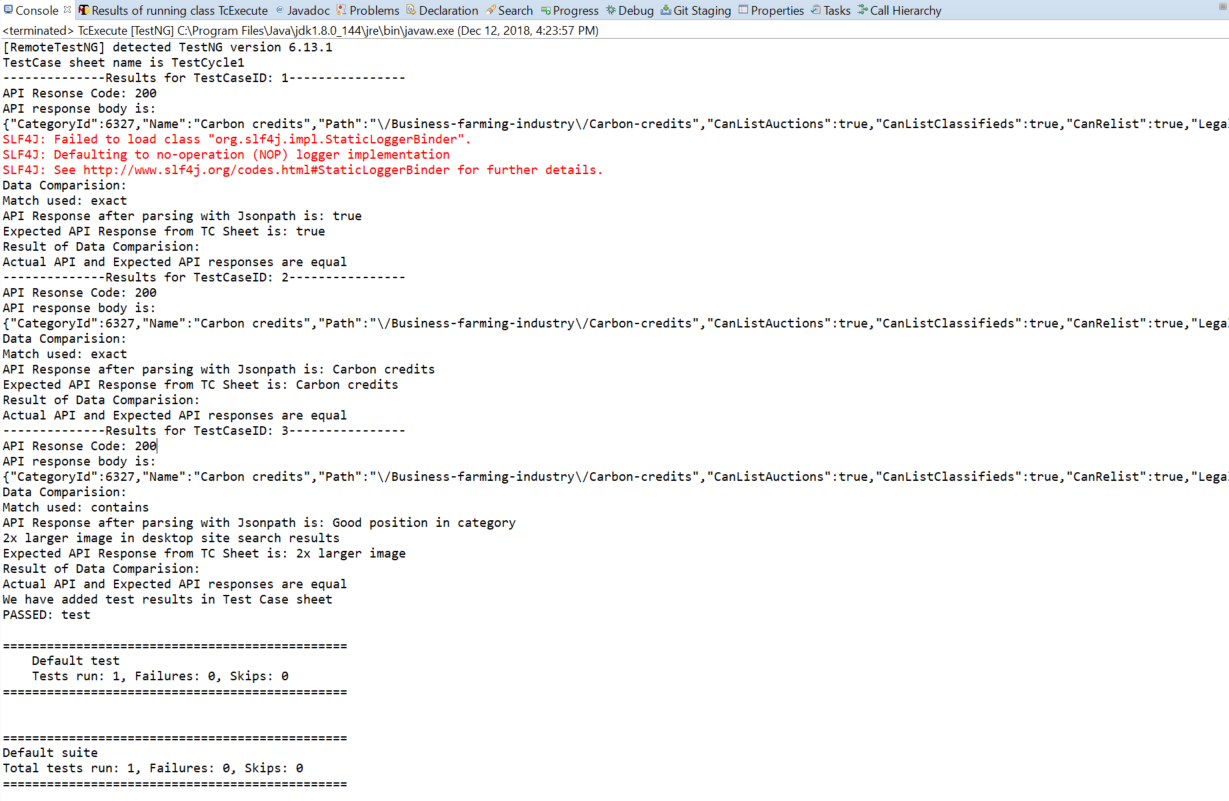
* User also needs to change below value in code, if location of excel sheet changes I local

In Main.TcExecute.java, add details where you are placing your TC sheet like:

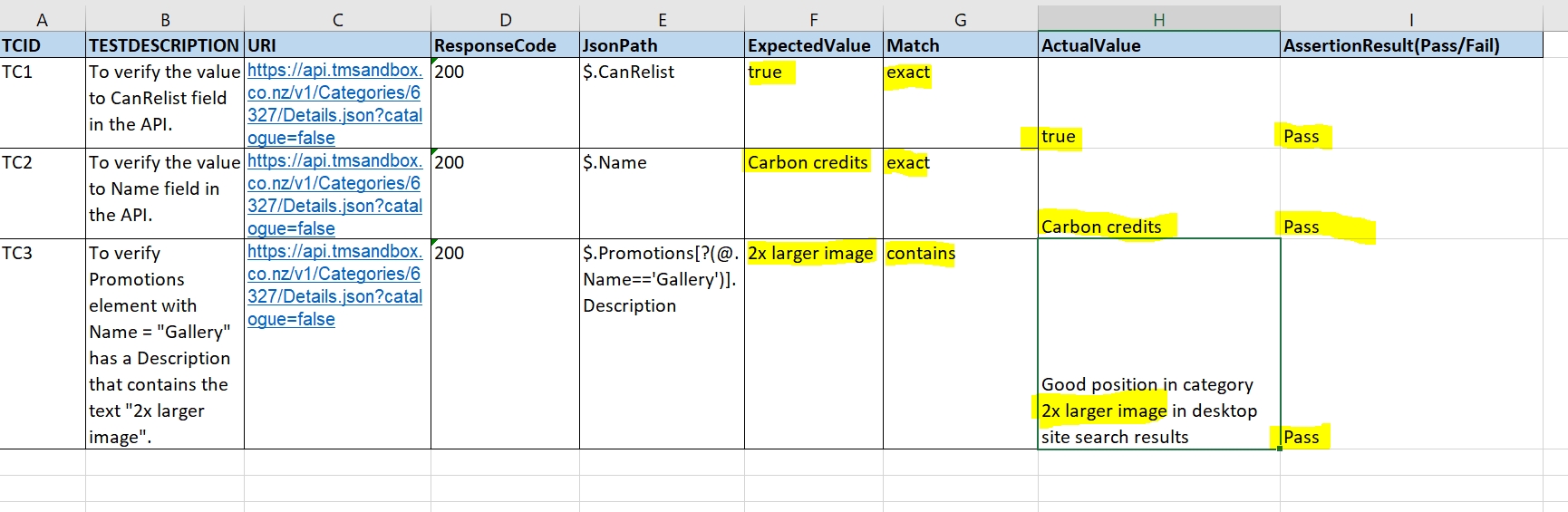
String TCFilepath = "C:\\Users\\bgopani\\workspace\\api-assurity-v1.0\\src\\TestcaseSheet\\TestCaseSheetV1.0.xlsx";

String TCTagname = "TestCycle1";

* Then user execute the TestNG class meant for execution of this keyword – “TcExecute.java” from “Main” package.
* Results format in the form of logs would be like:



* Test case sheet would look like below screenshot, after the execution.:



**Going Further:**

* User can add as many matches you want to verify data in API.
* This can be further moved ahead with TestNG.xml file and integrated to Jenkins for scheduling.
* And for reporting to UI, it can be integrated to Microsoft Allure reports, which has easy visualization for stack holders to see the stability of the build.

**Code Understanding:**

* Test Case Sheet related class:
  + JustReadTC
  + TCExecuteLogic
  + WriteActutalOuptput
  + WriteResults
* Test Case Execution Class:
  + TcExecute
* Other class for data addition, data modification and navigation, supporting main method:
  + CheckJsonArray
  + Compare
  + HitApi
  + ParseData

Note:

You can use framework further to add your own test cases. And in future with proper designing of framework you can verify all details in one GO !!!