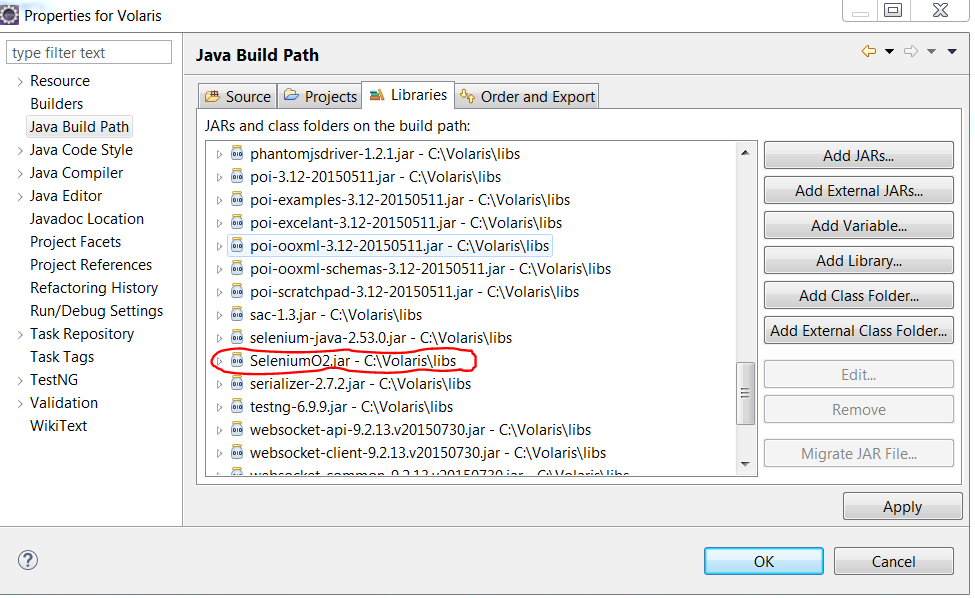
SeO2-A BRIEF USER GUIDE

SeO2 stands for Selenium Dioxide Framework. This is a framework used for automation only with Selenium. This framework can also be incorporated with other frameworks as per user requirements. This document is a brief guide of how to feed in the automation data to framework and some conventions to be strictly followed while using SeO2 framework. It has been presumed that the person has knowledge on Java and Selenium.

Prerequisite:

1. The user is expected to have SeO2 jar integrated with Eclipse.



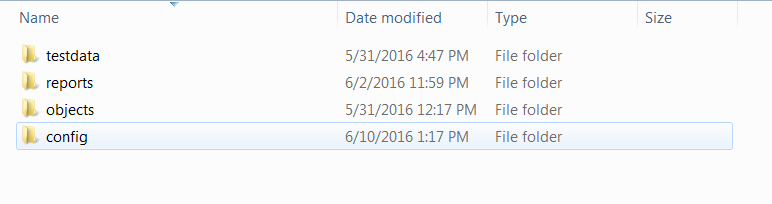
1. Also the user is expected to know basic terminologies related to automation like Test data, Object Repository, Reports, Test plan. A one line definition of the above is stated below

* Test data: This consists of the data that we need to pass during automation like username, password etc.
* Object Repository: This is a collection of objects which are identified through locators like id, xpath, css selectors etc.
* Reports: The automation results are stored as a file.
* Test plan: The test plan in SeO2 has test plan(in which you specify which test cases are to be executed), Scripts(indicating in which method and package the test cases are present) and Environment(which specifies the path of various folders, application URL etc.)

The detailed description will be covered in the later part of document.

Approach

The following folder structure would be set up in SeO2



Now we would be seeing how to fill in the details into these folders and the conventions to be followed while filling the details and the coding to be done according to framework standards.

We shall cover the topics in the following order

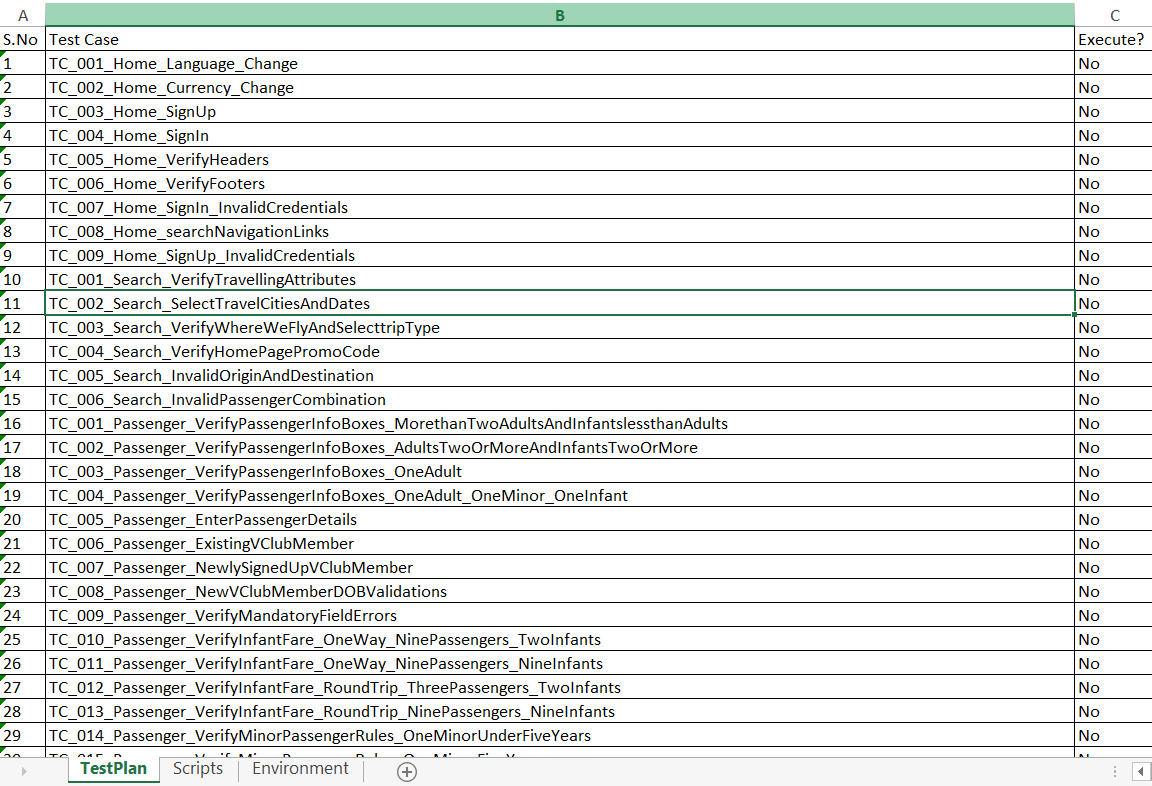
* Config
* Objects
* Test data
* Reports

1. **Config**

The config folder has a testplan excel sheet which contains three sheets in it.

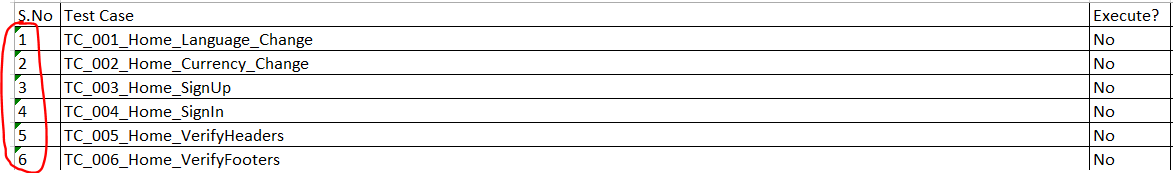
1. Testplan
2. Scripts
3. Environment
4. *Testplan*

Testplan contains the test cases and the approval to execute or not. If a test case has to be executed, a Yes is specified in the “Execute?” column else a No.

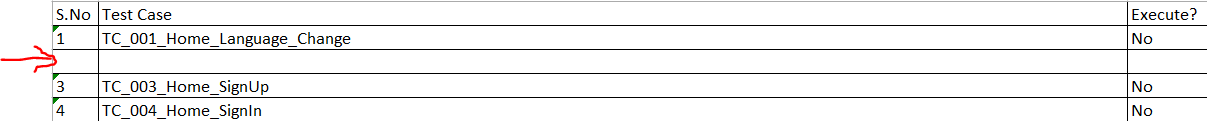


Note

* The serial numbers must be written with single quote(‘) followed by the respective serial number

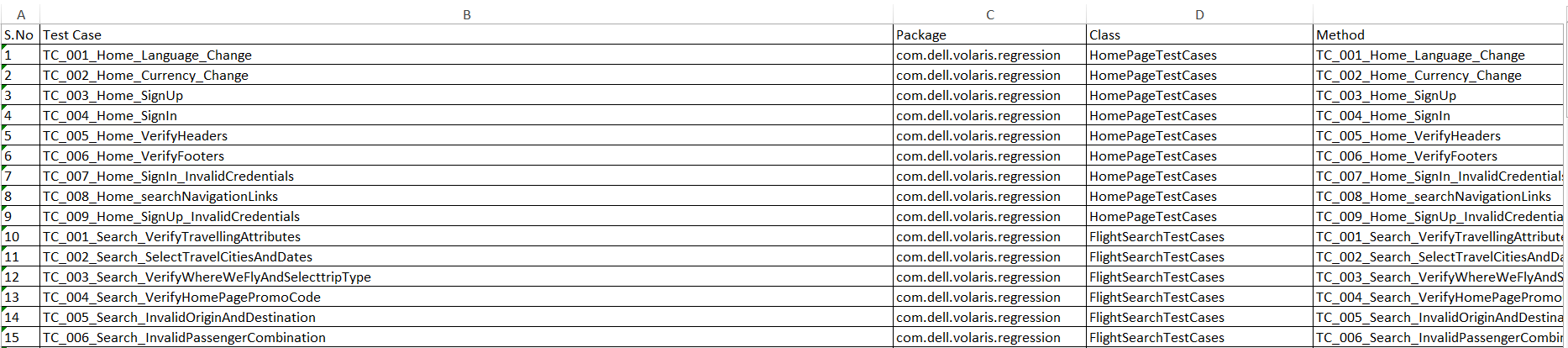


* No rows should be empty in between in other words no rows should be left empty. Below is the clip which stresses upon the mistake which should not be committed.



1. *Scripts*

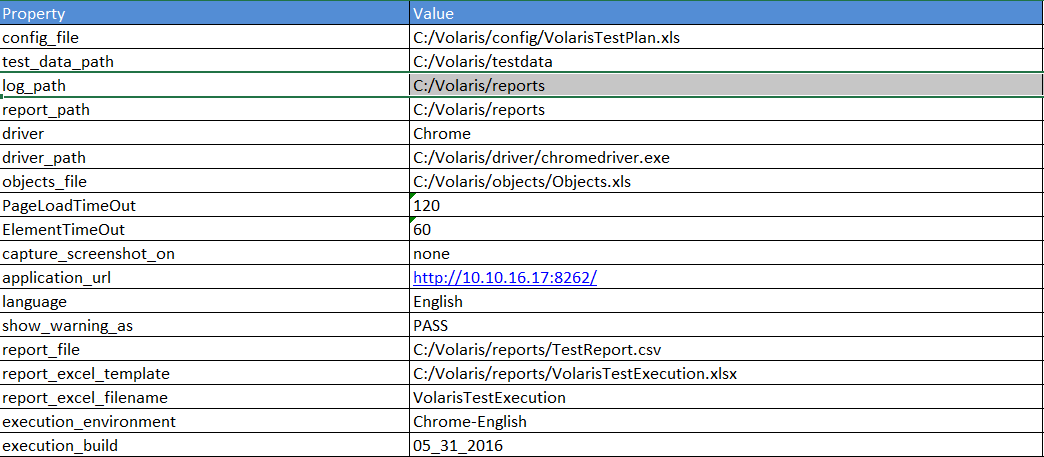
This sheet contains Test case name, the package where the test case is present, the class name where the test case is present and the method name which is basically the test case name.



Note

* The serial numbers must be written with single quote(‘) followed by the respective serial number
* No rows should be empty in between in other words no rows should be left empty.

1. *Environment*

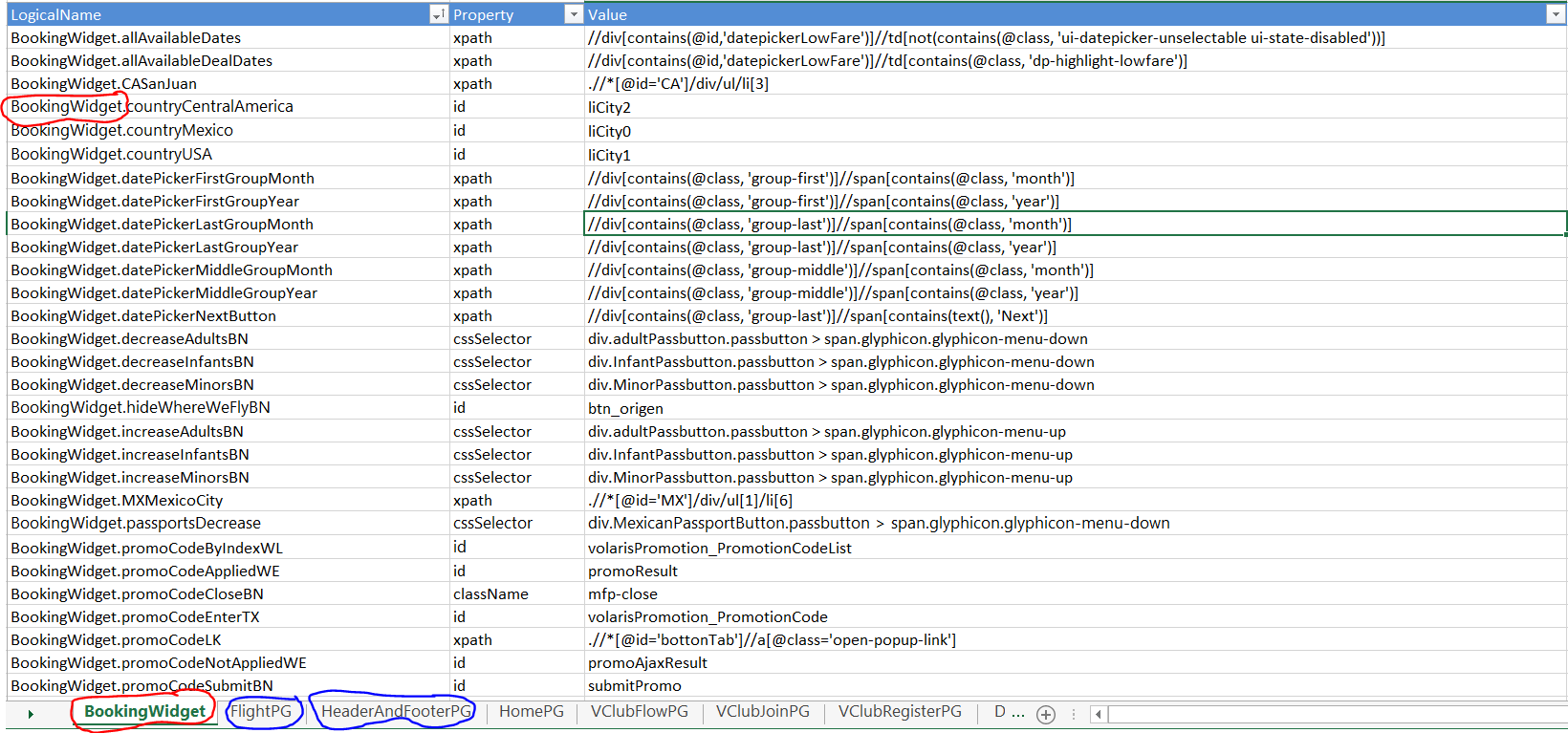


* In the above snap Volaris is the name of the project (respective project name) in which we have the following folders like testdata, objects, config, reports.
* config\_file is the path of the testplan excel sheet in config folder.
* Test\_data\_path and objects\_file is the path of the test data and objects in the project folder
* report\_path is the path of the report folder in the project folder
* driver\_path is the location of the driver(chrome driver or IE driver or any driver you wish to execute with)
* PageLoadTimeOut and ElementTimeOut are the time duration you want to wait for the visibility of page or element(depends upon the application rendering by the server)
* capture\_screenshot\_on
* application\_url is the url of your application you want to launch.
* language property is optional because it is the languages in which you want your application to run.
* show\_warning\_as
* report\_file is the path of the
* report\_excel\_template is the path of the excel sheet which is present in the reports folder of the project folder.
* execution\_build is the date on which you are executing your test scripts.

NOTE

* Values in the Property column should be lower case.
* Values in the Value column should be given with forward slashes.
* No rows should be left empty.

1. **Objects**

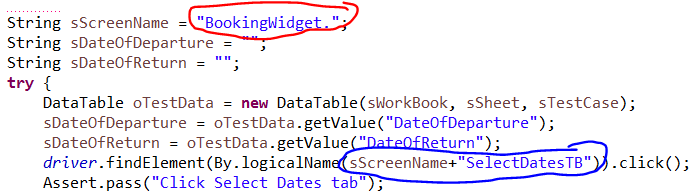


Objects folder in the project folder has an excel sheet which appears like the snap as above.

Property indicates the locator used to locate the object and value indicates the value of property.

There are different sheets (blue marker in above snap) depending on different pages of application. Red markers in above snap and below snap speaks of the same.

* LogicalName is the name of the locator of the element which is identified using xpath, id etc.

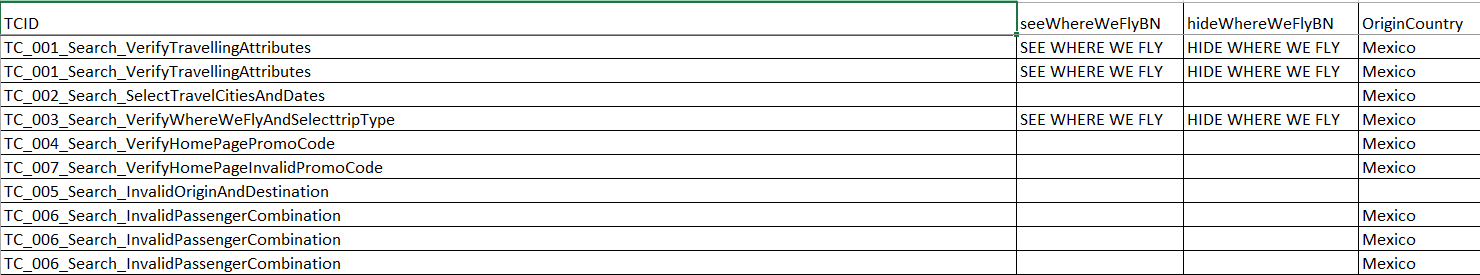


Analyzing above two snaps, we find that maintaining object repository by using logical names, we can implement changes easily. Blue marker in above snap helps to know how an object is identified by using logical names where the concatenated string is searched to find the object in the object sheet.

NOTE

* Property values should be in lower case.
* There should not be empty rows.

1. **Test data**



Test data is present in the Test data excel sheet of test data folder. It contains the test data required for the execution of the script.

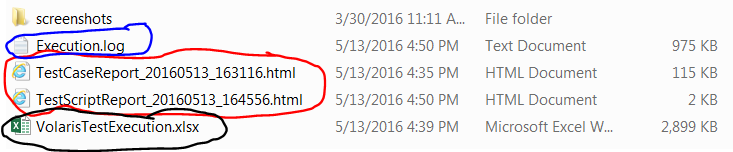
NOTE

* It is a rule that we must compulsorily have first two rows with the same data.

1. **Reports**

Project folder has a report folder where the results of the automation done gets updated. The report folder will generate html results (red marker) as test case report (executed test case steps) and test script report (executed test cases) which can be launched in any browser to read the results.

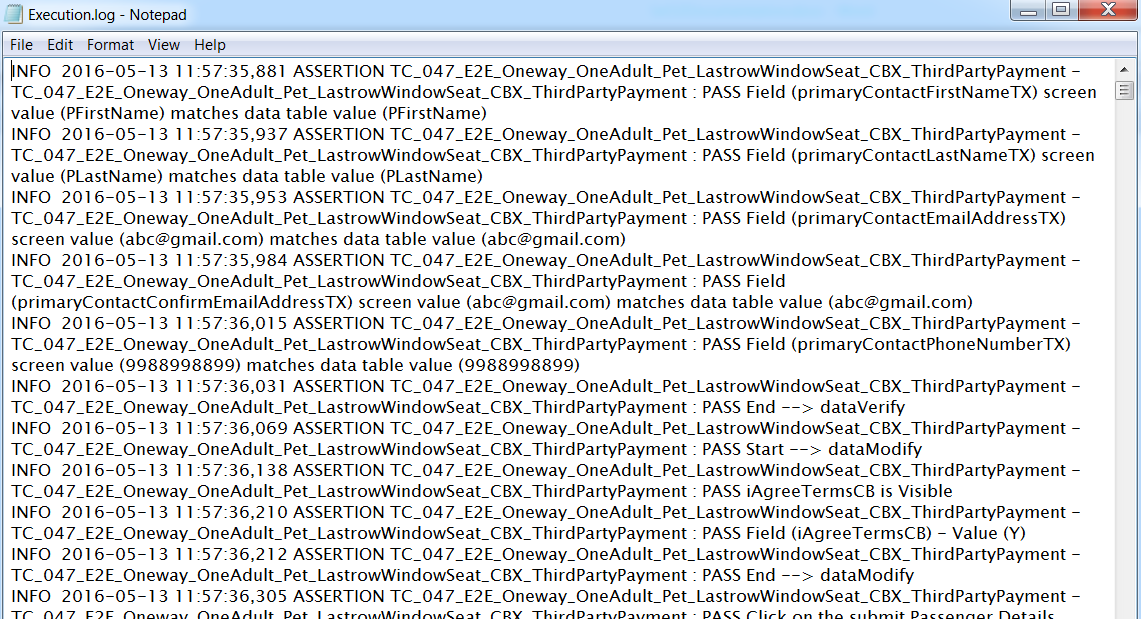
Execution log also gets generated and updated as you run the scripts (blue marker).



Apart from html reports and execution log we also have an excel sheet which gets updated upon execution. But the test case to be executed must be added with start time initialized to 0.00 to the excel sheet for the result to get updated.



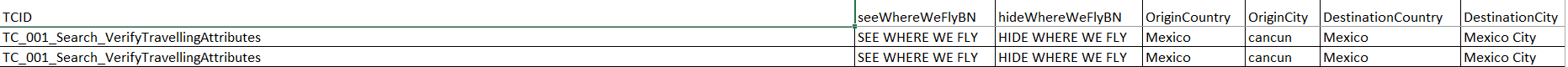
The execution log gives detailed description of the automation done. A snap of the same as below.



Some key notes on coding standards

* The driver in the framework gets initialized using the below line of code.

driver = FrameworkDriver.driver;

* 

* logicalName is used to identify the objects in the application which makes the maintainence of code easy. For example

By.logicalName("CompanyPG.nameTX");

* In SeO2 framework, a row in an excel sheet can be fetched as below

DataTable d = new DataTable("Connector-Mass-TestData-laut999.xls", "GroupMaster", "UI\_Group\_Maintenance\_81");

Connector-Mass-TestData-laut999.xls is the workbook name

GroupMaster is the sheet name in the workbook and

UI\_Group\_Maintainence\_81 is the row that you want to extract.

Now d has the entire row fetched into it.

Now assuming that there is a column groupIdTx in GroupMaster sheet, we can fetch the value in that field using below line of code

String expectedGroupTx = d.get(“groupTx”);

* Considering above point we shall now check how to use assertEquals.

expectedGroupTx has the value of groupTx field. Now say we again fetch the same value as

String actualGroupTx=driver.findElement(By.logicalName(“groupIdTx”);

Assert.assertEquals(expectedGroupTx , actualGroupTx , “Verifying group id”);

The third argument in above line is the description that you can give (optional).