**Read Data from Excel – Data Driven Framework**

1. Create a class file ReadingData.java
2. 5 poi Api jar files to be added which is required to read the data(always can download from - www/qtpselenium/downloads/poi.rar
3. Add Xls\_Reader.java file in the project – all functions such as getRowCount() etc.
4. Primary Requirement – Read and Write the data

In ReadingData class

1. 1st Create the object of Xls\_Reader and pass the path of the file that is required to be read
2. You can user getRowCount () function to get the row count and store it to a variable , similarly getColumnCount()
3. Header is also counted
4. Read data – getCellData (“Sheetname”,” colname”,” rownum” ) OR getCellData (“Sheetname”,”colindex”,”rownum”)
5. Note: For columindex -> 0 – belongs to first column, 1-belongs to second column
6. Note: For rowIndex -> 1- belongs to first row, 2-belongs to second row , so on and so forth
7. SetCellData – (“sheetname”, “colname”,””)

Some Quick tips:

1. Put single quote in front of large numbers so that it is read in its original format

How excel reader file is made – Please go to this link – [www.poi.apache.org/spreadsheet/quick-guide.html](http://www.poi.apache.org/spreadsheet/quick-guide.html)

~Now, how do you strategically arrange the data and read data let’s say for 100 test cases:

1. You cannot create sheets for each test case, overhead is huge.
2. Data Management is biggest challenge in automation
3. Control – Whether to run a testcase or not?
4. Have a sheet for TestCase Sheet – Test Case ID and Runmode – If Runmode is Y, then only test case is eligible for run
5. In the Data Sheet, you have actual Test Case data – Test case ID and then in the next row, you have Runmode and rest other columns are actual data for that test case. Here also based on Runmode it will be decided within the respective testcaseid, if all data set is to be run or only some based on Runmode – Y or N
6. The next Test case should start leaving one line, so on and so forth.
7. Now, you should have another java file, where in you will code for actual strategy to read and write data. I will name the class as DataManagement.java

So three classes so far

1. Xls\_Reader.java
2. ReadingData.java
3. DataManagement.java

In Data Management –

1. String testcasename = “TestA” – once we define for TestA , it should read data only for TestA
2. Reach for test case for which data must be read
3. Get to the rownum first for the Test Case
4. Then reach to columnStartRowNum for the desired test case and store in a variable
5. Then reach to dataStartRowNum for the desired test case and store in a variable
6. Please note these are values which can be gained when you add +1 or +2 to the testcaseRowNum respectively
7. Now – We calculate rows of data; how many rows of data is present. Then you can print total rows – which means total iteration for that test case.
8. Now – We calculate cols of data; how many fields of data is present for that test case. Then you can print total cols – which means total number of data fed for that test case iteration for that test case.

Read Data –

1. You must have 2 for loop – 1 to get to the row and then next level loop will take to columns for the respective row.

Write Data –

1. Define another object of the reader class and pass the path of the output sheet and then use the SetCellData function to write to data to the output sheet

Data Provider - Storing and Feeding Data to Test Case:

1. Instead of printing data, create a two dimensional object array, called as @DataProvider. Store the test case data in this dataProvider and provide the feed to the Test Case.
2. You will have to pass the same number of parameters to the test case based on its number test data (i.e. column) – Big over head
3. To solve it – Hash Table is utilized.

Hash Table –

1. Hashtable concept is to have one row and each row has a bag – Bag is nothing but a key value pair of every test case iteration for the respective test case.
2. Pass the Hashtable as parameter in the test case.
3. This helps reduce the overhead big time.