**.properties file to hold paths and developing DriverScript.java:**

.properties file is used to hold global paths like paths to the sheets etc.

Create a package “executionEngine” under the project.

Create a file with name config.properties under this package and put the following code.

WB\_PATH\_TESTS = C:\\Users\\padal\\eclipse-workspacenew\\KeywordDrivenProject\\tests\\tests\_main.xlsx

WB\_PATH\_TESTS\_SHEET = TestCases

WB\_PATH\_TEST\_CASES = C:\\Users\\padal\\eclipse-workspacenew\\KeywordDrivenProject\\tests\\

MASTERKW\_PATH = C:\\Users\\padal\\eclipse-workspacenew\\KeywordDrivenProject\\data\\masterkeywords.xlsx

MASTERKW\_SHEET = globalkeywords

ADMIN\_UNAME = admin

ADMIN\_PWD = Test@12345

USER\_UNAME = subbu123@gmail.com

USER\_PWD = subbu123

Here we are saving the paths to the tests\_main.xls, individual test cases and also the masterkeywords.xlsx file. We use these paths in our programs.

DriverScript.java: Create a class with name DriverScript under the package “executionEngine”.

**public** **static** **void** main(String[] args) **throws** Exception {

excelUtilities eu = **new** excelUtilities();

Properties gldata = **new** Properties();

InputStream input = **new** FileInputStream("src/executionEngine/config.properties");

gldata.load(input);

List<List<String>> testcases = **new** ArrayList<List<String>>();

testcases = eu.getTestCases(gldata.getProperty("WB\_PATH\_TESTS"), gldata.getProperty("WB\_PATH\_TESTS\_SHEET"));

System.***out***.println(testcases);

DriverScript.*prepareKeywords*(testcases);

}

The above main method will read the properties file for the path for the main tests excel which contains the test cases. Once it gets the path it will pass this information to a method called “getTestCases()” in “excelUtilities.java” file. So we need to create this method.

Create another package with name “utility” and create a class “excelUtilities.java” under that. Create the method “getTestCases()” to read the main test cases file.

//This function will return all the main test cases like TC01, TC02 etc where run = "yes"

**public** List<List<String>> getTestCases(String wbpath, String sheetname) **throws** IOException{

//This list holds all the test cases

List<List<String>> testcases = **new** ArrayList<List<String>>();

FileInputStream fis = **new** FileInputStream(wbpath);

XSSFWorkbook workbook = **new** XSSFWorkbook(fis);

XSSFSheet sheet = workbook.getSheet(sheetname);

//Get all the rows

Iterator<Row> rows = sheet.iterator();

**while**(rows.hasNext()) {

//This list holds each test case info test case no, workbook, sheet

List<String> testcaseinfo = **new** ArrayList<String>();

Row rowno = rows.next();

Cell tcvalue = rowno.getCell(0);

Cell wbvalue = rowno.getCell(1);

Cell shvalue = rowno.getCell(2);

Cell rvalue = rowno.getCell(4);

String testcasevalue;

String workbookvalue;

String sheetvalue;

String runvalue;

DataFormatter df = **new** DataFormatter();

runvalue = df.formatCellValue(rvalue);

**if**(runvalue.equalsIgnoreCase("yes")) {

testcasevalue = df.formatCellValue(tcvalue);

testcaseinfo.add(testcasevalue);

workbookvalue = df.formatCellValue(wbvalue);

testcaseinfo.add(workbookvalue);

sheetvalue = df.formatCellValue(shvalue);

testcaseinfo.add(sheetvalue);

testcases.add(testcaseinfo);

}

}

workbook.close();

**return** testcases;

}

Using the above two methods, the test cases likeTC01, TC02 with their workbook names and sheet names will be returned.