**Implementing Posting a Vehicle:**

Posting a vehicle is a group of keywords and we named the group as post vehicle. Moreover, this has to be data driven and we have already prepared the data for this. Whenever we want to post a vehicle we can use this keyword but at the same time we would like to add the row numbers from the datasheet as well. That means if we want to add vehicle 1, the keyword should be adminpostvehicle(1). Similarly if we want to post vehicles 2 and 3, we have to mention the keyword as adminvehicle(2,3).

So now let’s add one more test case. The test case is mentioned below.

|  |  |  |  |
| --- | --- | --- | --- |
| **testcase** | **teststep** | **Keywords** | **description** |
| TC02 | 1 | adminopenhomepage | open admin home page |
| TC02 | 2 | adminenterusername | enter the admin user name |
| TC02 | 3 | adminenterpassword | enter the admin password |
| TC02 | 4 | adminclickloginbutton | click on login button |
| TC02 | 5 | adminpostvehicle(1,2,3) | Post Vehciles 1, 2 and 3 from datasheet |
| TC02 | 6 | adminmovetoactivity | Admin move to activity area |
| TC02 | 7 | adminclicklogout | Admin click on logout |
| TC02 | 8 | adminclosebrowser | Close the browser |

If you see the above test case, after admin login there is a keyword adminpostvehicle(1,2,3) but we don’t have any specific keyword like this in our masterkeywords list. So, when we are executing the test case we will check if there are any special keywords. If there are any special keywords then we check if there are parenthesis because the data will be passed in parenthesis. If the keyword is with parenthesis, we will separate it and the numbers we will take them into an array. Now we have to write a special function which executes the group of keywords that belong to this particular special keyword by passing the array as parameter.

For this first we have to update the executeTC function. We will do it first for the special key.word adminpostvehicle

**Updating the executeTC function:**

**public** **static** **void** executeTC(List<String> originalkeywords, Set<String> actionclass, Map<String, String> keywordvsac, Set<String> objectrepository, Map<String, String> keywordvsor) **throws** ClassNotFoundException, NoSuchMethodException, SecurityException, IllegalAccessException, IllegalArgumentException, InvocationTargetException, InstantiationException, IOException {

System.*setProperty*("webdriver.gecko.driver", "C:\\BrowserDrivers\\geckodriver.exe");

WebDriver driver = **new** FirefoxDriver();

**for**(String str : originalkeywords) {

String keyword = str;

**if**(keyword.indexOf("adminpostvehicle") != -1) {

String[] parts = keyword.split("\\(");

String[] dataelements = (parts[1].split("\\)"))[0].split("\\,");

*specialfunction*(parts[0],dataelements,driver);

}

**else** {

String actioncl = keywordvsac.get(keyword);

String objectcl = keywordvsor.get(keyword);

Class<?> cls = Class.*forName*("actions."+actioncl);

Class<?> orc = Class.*forName*("objectrepository."+objectcl);

Method[] methodcall = cls.getDeclaredMethods();

**for**(Method m : methodcall) {

**if**(keyword.equalsIgnoreCase(m.getName()) && m.getParameterCount() == 0)

{

Method mc = cls.getDeclaredMethod(keyword);

Constructor<?> constructor = cls.getConstructor(WebDriver.**class**);

mc.invoke(constructor.newInstance(driver));

}

**else** **if**(keyword.equalsIgnoreCase(m.getName()) && m.getParameterCount() == 1)

{

Method morc = orc.getDeclaredMethod(keyword);

Constructor<?> orconstructor = orc.getConstructor(WebDriver.**class**);

WebElement we = (WebElement) morc.invoke(orconstructor.newInstance(driver));

Method mc = cls.getDeclaredMethod(keyword,WebElement.**class**);

Constructor<?> constructor = cls.getConstructor(WebDriver.**class**);

mc.invoke(constructor.newInstance(driver),we);

}

}

}

}

}

We have updated the executeTC() method. First we are checking if “adminpostvehcile” is a part(substring) of the keyword “adminpostvehicle(1,2,3)” in the test case. If it is, it will split the keyword based on the open parenthesis and take the numbers into an array. Then it will call a special function which will go to the keyword group for the special keyword and executes all of them. So, lets write the special function now.

**public** **static** **void** specialfunction(String specialkeyword, String[] datarows, WebDriver driver) **throws** IOException, ClassNotFoundException, NoSuchMethodException, SecurityException, IllegalAccessException, IllegalArgumentException, InvocationTargetException, InstantiationException{

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

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

gldata.load(input);

excelUtilities ecu = **new** excelUtilities();

List<String> subkeywords = ecu.getSubKeywords(gldata.getProperty("MASTERKW\_PATH"), specialkeyword);

List lis = ecu.getActionsObjects(subkeywords, gldata.getProperty("MASTERKW\_PATH"), gldata.getProperty("MASTERKW\_SHEET"));

Set<String> actionclass = (Set<String>) lis.get(0);

Map<String, String> keywordvsac = (Map<String, String>) lis.get(1);

Set<String> objectrepository = (Set<String>) lis.get(2);

Map<String, String> keywordvsor = (Map<String, String>) lis.get(3);

Map<String, Map<String, String>> spdata = ecu.getdata(gldata.getProperty("DATA\_PATH"), specialkeyword);

**for**(**int** i=0; i<datarows.length; i++) {

**for**(String str : subkeywords) {

String keyword = str;

String actioncl = keywordvsac.get(keyword);

String objectcl = keywordvsor.get(keyword);

Class<?> cls = Class.*forName*("actions."+actioncl);

Class<?> orc = Class.*forName*("objectrepository."+objectcl);

Method[] methodcall = cls.getDeclaredMethods();

**for**(Method m : methodcall) {

**if**(keyword.equalsIgnoreCase(m.getName()) && m.getParameterCount() == 0)

{

Method mc = cls.getDeclaredMethod(keyword);

Constructor constructor = cls.getConstructor(WebDriver.**class**);

mc.invoke(constructor.newInstance(driver));

}

**else** **if**(keyword.equalsIgnoreCase(m.getName()) && m.getParameterCount() == 1)

{

Method morc = orc.getDeclaredMethod(keyword);

Constructor orconstructor = orc.getConstructor(WebDriver.**class**);

WebElement we = (WebElement) morc.invoke(orconstructor.newInstance(driver));

Method mc = cls.getDeclaredMethod(keyword,WebElement.**class**);

Constructor constructor = cls.getConstructor(WebDriver.**class**);

mc.invoke(constructor.newInstance(driver),we);

}

**else** **if**(keyword.equalsIgnoreCase(m.getName()) && m.getParameterCount() == 2)

{

Method morc = orc.getDeclaredMethod(keyword);

Constructor orconstructor = orc.getConstructor(WebDriver.**class**);

WebElement we = (WebElement) morc.invoke(orconstructor.newInstance(driver));

Method mc = cls.getDeclaredMethod(keyword,WebElement.**class**,String.**class**);

Constructor constructor = cls.getConstructor(WebDriver.**class**);

mc.invoke(constructor.newInstance(driver),we,(spdata.get(datarows[i])).get(keyword));

//Thread.sleep(1000);

}

}

}

}

}

Now this special function will in turn call the getSubKeywords() to get the keywords belong to the particular group. It also calls the getdata() method to get the data for the numbers mentioned in the parenthesis (1,2,3). Now the specialfunction() will execute the special keyword functionality.

//This function will return the action class names and object repository names a particular test case needs

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

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

**int** i = 0;

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

**try** {

Row rowno = rows.next();

Cell kwvalue = rowno.getCell(0);

String keywordvalue;

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

keywordvalue = df.formatCellValue(kwvalue);

**if**(i>0) {

lis.add(keywordvalue);

}

i++;

}

**catch**(Exception e) {

}

}

workbook.close();

**return** lis;

}

//This function gets the data like vehicle data, brands data etc

**public** Map<String, Map<String, String>> getdata(String wbpath, String sheetname) **throws** IOException {

Map<String, Map<String, String>> mp = **new** HashMap<String, Map<String, String>>();

//Get access to the workbook

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

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

//Get the sheet you want

XSSFSheet sheet = workbook.getSheet(sheetname);

//Get all the rows

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

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

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

String testcase = "";

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

Map<String, String> mpss = **new** HashMap<String, String>();

Row rowno = rows.next();

Iterator<Cell> cells = rowno.iterator();

**int** i = 0;

**if**(rowno.getRowNum() == 0)

{

**int** k = 0;

**while**(cells.hasNext()) {

Cell value = cells.next();

String cellvalue;

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

cellvalue = df.formatCellValue(value);

**if**(k>0) {

fieldnames.add(cellvalue);

//System.out.println(cellvalue);

}

k++;

}

}

**if**(rowno.getRowNum() > 0) {

**while**(cells.hasNext()) {

i++;

Cell value = cells.next();

String cellvalue;

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

cellvalue = df.formatCellValue(value);

**if**(i == 1) {

testcase = cellvalue;

}

**else** {

ls.add(cellvalue);

}

}

**for**(**int** j=0; j<fieldnames.size(); j++) {

mpss.put(fieldnames.get(j), ls.get(j));

}

mp.put(testcase, mpss);

}

}

workbook.close();

**return**(mp);

}