**Converting admin login functionality to a module:**

Now we have to modify the admin login functionality to a module so that once we call the special keyword instead of calling the individual keywords like open browser, enter user name etc., it should login directly so that the number of steps we use will decrease.

So first identify the keywords that are needed for this and name it as “adminlogin” and make them as separate group and give the same name to the sheet as “adminlogin”.

The following are those keywords.

|  |  |  |  |
| --- | --- | --- | --- |
| **keywords** | **Description** | **actionclass** | **objectrepository** |
| adminopenhomepage | open admin home page | adminlogin | adminloginOR |
| adminenterusername | enter the admin user name | adminlogin | adminloginOR |
| adminenterpassword | enter the admin password | adminlogin | adminloginOR |
| adminclickloginbutton | click on login button | adminlogin | adminloginOR |

We already have the adminlogin action class and adminloginOR classes. So all we have to do now is create a test case with keyword name as adminlogin and modify the DriverScript.java to run it.

The following is the main test cases file.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| testcase | workbook | sheet | Description | Run |
| TC01 | tests\_wb1 | TCGroup1 | Login as admin, logout and close browser | Yes |
| TC02 | tests\_wb1 | TCGroup1 | Login as admin and post add three vehcles | Yes |
| TC03 | tests\_wb1 | TCGroup1 | Login as admin and post add one vehcles | Yes |
| TC04 | tests\_wb1 | TCGroup1 | Login as admin, logout and close browser (Use adminlogin keyword) | Yes |
| TC05 | tests\_wb1 | TCGroup1 | Login as admin and add one vehcles (Use adminlogin keyword) | Yes |

We have added two more test cases here which are TC04 and TC05.

Now let’s develop the individual test steps.

|  |  |  |  |
| --- | --- | --- | --- |
| TC04 | 1 | adminlogin | admin login functionality - open admin homepage and login |
| TC04 | 2 | adminmovetoaccount | Move to account area |
| TC04 | 3 | adminclicklogout | Admin click on logout |
| TC04 | 4 | adminclosebrowser | Close the browser |
| TC05 | 1 | adminlogin | admin login functionality - open admin homepage and login |
| TC05 | 2 | adminpostvehicle(2) | Post Vehicle 2 from datasheet |
| TC05 | 3 | adminmovetoaccount | Admin move to activity area |
| TC05 | 4 | adminclicklogout | Admin click on logout |
| TC05 | 5 | adminclosebrowser | Close the browser |

Now we have to modify the DriverScript.

**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)||(keyword.indexOf("adminlogin") != -1)) {

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

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

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

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

}

**else** {

String parts = keyword;

String[] dataelements = **null**;

*specialfunction*(parts,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 modified the executeTC() method here. We have added one more condition to check if the keyword contains “adminlogin”. But here adminlogin is not data driven so we don’t pass the parameters with it. So we are checking if the special keyword contains any open parenthesis ( “(“ ) and if not we make the parts element as the keyword and dataelements as null array and call the specialfunction.

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

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

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

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

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

}

**else** {

String parts = keyword;

String[] dataelements = **null**;

*specialfunction*(parts,dataelements,driver);

}

}

Now we have to modify the specialfunction() as well.

**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 = **new** HashMap<>();

**if**(datarows != **null**) {

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

}

**int** no\_data\_times = 0;

**if**(datarows == **null**) {

no\_data\_times = 1;

}

**else** {

no\_data\_times = datarows.length;

}

**for**(**int** i=0; i<no\_data\_times; 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);

}

}

}

}

}

Here first we are checking if the datarows array passed is null or not. If not null then only we will call the getdata() method. IF it is null we don’t call it.

Now since the for loop has to go at least one time, we created a new integer variable no\_data\_times and if the datarows array is null we are making the no\_data\_times as 1 so that the for loop will continue at least one time.

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

**if**(datarows != **null**) {

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

}

**int** no\_data\_times = 0;

**if**(datarows == **null**) {

no\_data\_times = 1;

}

**else** {

no\_data\_times = datarows.length;

}

**for**(**int** i=0; i<no\_data\_times; i++)

Now run the tests and the tests should run with out any failure.