**Adding Subscribers – Data Driven:**

In order to subscribe a user first we have to open home page, enter the email address in the subscriber email field, click the submit button. Then verify the text on the javascript alert and click ok. Then close the browser.

We are developing the subscriber functionality as a special function “subscribeuser()”. So we have to group the keywords as separate sheet and name it as “subscribeuser”. Let’s see the test case first.

Test Case:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **testcase** | **teststep** | **Keywords** | **description** | **action class and OR** |
| TC24 | 1 | Useropenhomepage | user comes to homepage | userlogin |
| TC24 | 2 | subscribeuser(2,3,4) | subscribe users 2,3,4 from data sheet | special function |
| TC24 | 3 | Userclosebrowser | Close the browser | userlogin |

In the above test case “useropenhomepage” and “userclosebrowser” are already developed.

Let’s see the keywords for the special function “subscribeuser”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **keywords** | **description** | **actionclass** | **objectrepository** | **Method** |
| entersubscriberemail | Enter subscribers email | usersubscribe | usersubscribeOR | entersubscriberemail() |
| clickemailsubscribe | Click on subscribe button | usersubscribe | usersubscribeOR | clickemailsubscribe() |
| checksubscribesuccessmessage | check the success message on the javascript alert | usersubscribe | usersubscribeOR | checksubscribesuccessmessage() |
| acceptsubscribealert | accept the javascript alert | usersubscribe | usersubscribeOR | acceptsubscribealert() |

We also develop a datasheet with the same name as special function “subscribeuser”

|  |  |
| --- | --- |
| **sno** | **entersubscriberemail** |
| 1 | venkat@gmail.com |
| 2 | suresh@gmail.com |
| 3 | ganesh@gmail.com |
| 4 | ram@gmail.com |
| 5 | rajesh@gmail.com |
| 6 | ramesh@gmail.com |
| 7 | rakesh@gmail.com |
| 8 | bannu@gmail.com |
| 9 | narendra@gmail.com |
| 10 | sai@gmail.com |

Now we need to add the special function name to the properties file.

**config.properties file:**

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

WB\_PATH\_TESTS\_SHEET = TestCases

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

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

MASTERKW\_SHEET = globalkeywords

DATA\_PATH = C:\\Users\\padal\\eclipse-workspacenew\\HybridKeywordDriven\\data\\data.xlsx

SPECIAL\_KEYWORDS = adminpostvehicle,adminlogin,adminpostbrand,registeruser,userlogin,**subscribeuser**

CHECK\_KEYWORDS = checkusersdifference,checkvehiclesdifference,checkbookingsdifference,checkbrandsdifference,checksubscribersdifference,checkqueriesdifference,checktestimonialsdifference

BOOKING\_VEHICLES = userclickvehicledetails

ADMIN\_UNAME = admin

ADMIN\_PWD = Test@12345

USER\_UNAME = subbu123

USER\_PWD = subbu123

Now we need to develop the object repository and actionclass.

**usersubscribeOR.java object repository:**

**package** objectrepository;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**public** **class** usersubscribeOR {

WebDriver driver;

**public** usersubscribeOR(WebDriver driver){

**this**.driver = driver;

}

**public** WebElement entersubscriberemail() {

WebElement subscriberemailelement = driver.findElement(By.*xpath*("//input[@name='subscriberemail']"));

**return** subscriberemailelement;

}

**public** WebElement clickemailsubscribe() {

WebElement subscribebuttonelement = driver.findElement(By.*xpath*("//button[@name='emailsubscibe']"));

**return** subscribebuttonelement;

}

}

**usersubscribe.java action class:**

**package** actions;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.testng.Assert;

**public** **class** usersubscribe {

WebDriver driver;

**public** usersubscribe(WebDriver driver){

**this**.driver = driver;

}

**public** **void** entersubscriberemail(WebElement subscriberemailelement, String subscriberemail) {

subscriberemailelement.sendKeys(subscriberemail);

}

**public** **void** clickemailsubscribe(WebElement subscribebuttonelement) {

subscribebuttonelement.click();

}

**public** **void** checksubscribesuccessmessage() {

String actmsg = driver.switchTo().alert().getText();

String expmsg = "Subscribed successfully.";

Assert.*assertEquals*(actmsg, expmsg);

System.***out***.println("Subscriber added successfully!");

}

**public** **void** acceptsubscribealert() {

driver.switchTo().alert().accept();

}

}