APPLICATION DEVELOPMENT AND MAINTANANCE

ASSIGNMENT -3

Submitted To : Rahulnath H.A
Date of Submission: 02/04/2020

Team Members

1 Preethi P S – 17
2 Revathy U – 20
3 Sana Earnest - 22
4 Saniya Thahseen - 23
5 Saritha H – 25

Problem Statement

Create a web application of your choice that has presentation, businesslogic and persistence layers. You may use any language of your choice to develop the business logic and any DB for the persistence layer.

1. Use any free software testing tool, to create an automated scripts totest certain aspects of your application

2. Test results must also be documented

To be submitted:

1. Document the procedure you followed, from the installation to finaltesting report generation
2. Test Scripts you used
3. Test reports from the test suite(including test cases)

Prerequisites

1. The web application we used to test is Online Exam Portal which includes a login page for a user, a registration page for a new user.

2. The test tool that we used to test our web application was Selenium.

Selenium

Selenium is an open-source test automation suite used for automating web-based applications. It supports automation across different browsers, platforms and programming languages. Using selenium, we can automate the functional tests and easily integrate them with Maven, Jenkins and other build automation and continuous integration tools.

Components of Selenium Suite

Selenium Suite comprises of the following four components-

1. ***Selenium IDE*** – Selenium IDE is an add-on of the Firefox browser that provides record and playback functionality. Its use is limited and test scripts generated are not very robust and portable.

2. **Selenium RC** – Selenium Remote Control(RC) is officially deprecated by Selenium. It used to require an additional server for running the automation scripts and had many limitations.

3. ***Selenium WebDriver*** – By far the most important component of Selenium Suite. It provides different drivers for different browsers and supports multiple programming languages.

4. ***Selenium Grid*** – Selenium Grid is also an important part of Selenium Suite. It helps in the distributed running of selenium tests in parallel across multiple remote machines.

Download & Install TestNG in Eclipse for Selenium WebDriver

Following is a step by step procedure to install TestNG in Eclipse

Installing TestNG in Eclipse

Step 1) Launch Eclipse.

1. On the menu bar, click Help.
2. Choose the "Install New Software…" option.

Step 2) In the Eclipse Install dialog box

1. Enter "http://dl.bintray.com/testng-team/testng-eclipse-release/" in the Work with box
2. Click the Add button.

Step 3) In Add repository dialog

1. Enter "TestNG" in Name Text Field
2. Click on Add button.

Step 4)

1. Select the "TestNG" checkbox
2. Click on the "Next" button.

Step 5) It will give a review of Items to be installed. Click on Next.

Step 6) Installation complete here.

TestNG

TestNG is an automation testing framework in which NG stands for "Next Generation". TestNG is inspired from JUnit which uses the annotations (@). TestNG overcomes the disadvantages of JUnit and is designed to make end-to- end testing easy.

Using TestNG, you can generate a proper report, and you can easily come to know how many test cases are passed, failed, and skipped. You can execute the failed test cases separately.

Default Selenium tests do not generate a proper format for the test results. Using TestNG we can generate test results.

Most Selenium users use this more than Junit because of its advantages. There are so many features of TestNG, but we will only focus on the most important ones that we can use in Selenium. Following are key features of TestNG

- Generate the report in a proper format including a number of test cases runs, the number of test cases passed, the number of test cases failed,and the number of test cases skipped.
- Multiple test cases can be grouped more easily by converting them into testng.xml file. In which you can make priorities which testcase should be executed first.
- The same test case can be executed multiple times without loops justby using keyword called 'invocation count.'
- Using testng, you can execute multiple test cases on multiple browsers,i.e., cross browser testing.
- The testing framework can be easily integrated with tools likeMaven, Jenkins, etc.
- **WebDriver has no native mechanism for generating reports. TestNG can generate the report in a readable format**
- TestNG simplifies the way the tests are coded. There is no more need for a static main method in our tests. The sequence of actions is regulated by easy-to-understand annotations that do not require methods to be static.

TestNG Annotations

@BeforeSuite: The annotated method will be run before all tests in this suite have run.

@AfterSuite: The annotated method will be run after all tests in this suite have run.

@BeforeTest: The annotated method will be run before any test method belonging to the classes inside the tag is run.

@AfterTest: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.

@BeforeGroups: The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.

@AfterGroups: The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.

@BeforeClass: The annotated method will be run before the first test method in the current class is invoked.

@AfterClass: The annotated method will be run after all the test methods in the current class have been run.

@BeforeMethod: The annotated method will be run before each test method.

@AfterMethod: The annotated method will be run after each test method.

@Test: The annotated method is a part of a test case

Chrome Driver

Selenium is a tool that basically interacts with browsers to test your web applications. ChromeDriver is a standalone server which implements WebDriver's wire protocol for Chrome.

The chrome driver is used here because of two reasons :

- Chrome has the highest market share worldwide. So, it makes sense to work on the browser which majority of the people are using. We have given below a comparison chart of different browsers.
- Selenium works better in Chrome than other browsers, especially Firefox.

Steps to download ChromeDriver

1. Open chrome driver download page –
https://sites.google.com/a/chromium.org/chromedriver/downloads

2. This page contains all the versions of Selenium ChromeDriver.

3. Click on **ChromeDriver 2.39** link. You will be navigated to ChromeDriver download page which contains ChromeDriver for, Windows

4. Click on **chromedriver_win32.zip** to download ChromeDriver for Windows.

5. Once you download the zip file, unzip it to retrieve **chromedriver.exe**

About Test case

Test case 1: @BeforeTest
Open ChromeDriver and navigate to the URL specified.

Test case 2: @Test(priority = 0)
Verify the title of the page.

Test case 3: @Test(priority =1)
Create new user by entering user name , password , phone number.

Test case 4: @Test(priority =2)
Login with created user name, password.

Test case 5: @Test(priority =3)
Selected reasoning button and answer all the questions.

Test case 6: @Test(priority =4)
Check the status of the reasoning.

Test case 7: @AfterTest
Close the ChromeDriver.

Script

```
package newpackage;
import org.testng.Assert;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;
import java.text.NumberFormat;
import java.text.ParseException;
import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class ExamportalTestNG {
          public String baseurl ="http://localhost/exam/";
          String driverPath = "C:\\Users\\Saritha H\\eclipse-workspace\\chromedriver.exe";
          public WebDriver driver ;
           String Usrnm = "manu";
           String passwrd ="456";
           String phn ="123456788";


  @BeforeTest
    public void launchBrowser() {
     System.out.println("launching chrome browser");
     System.setProperty("webdriver.chrome.driver", driverPath);
     driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
```

```
   driver.get(baseurl);
 }

 @Test(priority = 0)
 public void verifyHomepageTitle() {
   driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
   String expectedTitle = "Exam portal";
   String actualTitle = driver.getTitle();
   Assert.assertEquals(actualTitle, expectedTitle);
}
 @Test(priority =1)
 public void createUser() throws InterruptedException {
    driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
    driver.findElement(By.linkText("New user")).click();
   driver.findElement(By.id("username")).sendKeys(Usrnm);
   driver.findElement(By.id("password")).sendKeys(passwrd);
   driver.findElement(By.id("conpassword")).sendKeys(passwrd);
   driver.findElement(By.id("phone")).sendKeys(phn);
   driver.findElement(By.xpath("//*[contains(@type,'submit')]")).click();
   Thread.sleep(2000);

}
 @Test(priority =2)
 public void Login() throws InterruptedException {
       driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
       driver.findElement(By.id("username")).sendKeys(Usrnm);
       driver.findElement(By.id("password")).sendKeys(passwrd);
       driver.findElement(By.xpath("//*[contains(@type,'submit')]")).click();

}
 @Test(priority =3)
 public void Reasoning() throws InterruptedException {
   driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
   driver.findElement(By.xpath("//*[contains(@value,'Reasoning')]")).click();
   driver.findElement(By.id("a")).click();
   driver.findElement(By.id("btn")).click();
   driver.findElement(By.id("c")).click();
   driver.findElement(By.id("btn")).click();
   driver.findElement(By.id("b")).click();
   driver.findElement(By.id("btn")).click();
   driver.findElement(By.id("c")).click();
   driver.findElement(By.id("btn")).click();
   driver.findElement(By.id("b")).click();
   driver.findElement(By.id("btn")).click();
   driver.findElement(By.id("a")).click();
```

```
    driver.findElement(By.id("btn")).click();
    driver.findElement(By.id("b")).click();
    driver.findElement(By.id("btn")).click();
    driver.findElement(By.id("b")).click();
    driver.findElement(By.name("complete")).click();
    Alert alert = driver.switchTo().alert();
    alert.accept();


 }

 @Test(priority =4)
 public void Status() throws InterruptedException {
           driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
           driver.findElement(By.linkText("status")).click();
          driver.findElement(By.xpath("//*[contains(@class,'skills js')]")).getText();

  }

@AfterTest public void Close() {
driver.quit();
}

 }
```

Test Report

Report generation is very important when you are doing the Automation Testing as well as for Manual Testing.

- By looking at the result, you can easily identify how many test cases are passed, failed and skipped.
- By looking at the report, you will come to know what the status of the project is.
- Selenium web driver is used for automating the web-application, but it won't generate any reports.

- The TestNG will generate the default report.
- When you execute testng.xml file, and refresh the project. You will get test-output folder in that folder.
- Right click on the emailable-report.html and select the option. Open with the web browser.

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	5	0	0	0	12,332		

Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
newpackage.ExamportalTestNG	**Login**	1588342731027	639
	Reasoning	1588342731670	1751
	Status	1588342733426	634
	createUser	1588342727546	3472
	verifyHomepageTitle	1588342727472	32

Default test

newpackage.ExamportalTestNG#Login

back to summary

newpackage.ExamportalTestNG#Reasoning

back to summary

newpackage.ExamportalTestNG#Status

back to summary

newpackage.ExamportalTestNG#createUser

back to summary

newpackage.ExamportalTestNG#verifyHomepageTitle

back to summary