Selenium Locators

1. Request test user login credentials for Magnus application from Technical CoCoordinator.

2. Identify the elements of the Login Page of Magnus application using different locators.

3. Write 5 automation scripts for login page of the Magus application.

4. Refer http://brightitcareer.com/testing/selenium/selenium\_locators.html for more details on locator.

5. Implement all the next assignments on the Magnus application only. Do not use any other application.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

public class seleniumLocator

{

WebDriver driver;

public void launchBrowser()

{

System.setProperty("webdriver. chrome. driver","C:\\Users\\Asus\\Downloads\\chromedriver\_win32\\chromedriver.exe");

driver = new ChromeDriver();

driver.manage().deleteAllCookies();

driver.get("http://magnus.jalatechnologies.com/");

}

public void locatorTest()

{

driver. findElement (By.id("UserName")).sendKeys("training@jalaacademy.com");

driver.findElement(By.id("Password")).sendKeys("jobprogram");

driver.findElement(By.id("btnLogin")).click();

}

public void closeBrowser()

{

driver.close();

}

}

Operations on Web Elements

Text Box

1. How to Type in text box using Selenium Web Driver.

2. How to get a Text Box value using Selenium Web Driver.

3. How to read the placeholder value of a Text Box using getAttribute() method.

4. Deleting/clear text from the Text Box.

5. Check if Text Box is enabled/disabled.

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.\*;

public class webElements

{

WebDriver driver;

public void launchBrowser()

{

System.setProperty("webdriver. chrome. driver","C:\\Users\\Asus\\Downloads\\chromedriver\_win32\\chromedriver.exe");

driver = new ChromeDriver();

driver.manage().deleteAllCookies();

driver.get("http://magnus.jalatechnologies.com/");

}

@Test (priority = 1)

public void locatorTest()

{

driver. findElement (By. xpath("//input[@type='text']")).sendKeys("training@jalaacademy.com");

driver. findElement (By.className("form-control")).sendKeys("jobprogram");

driver.findElement(By.xpath("//button[@type='button' and @class='btn btn-primary btn-flat']")).click();

}

@Test (priority = 2)

public void jsExTest()

{

JavascriptExecutor js = (JavascriptExecutor) driver;

js. executeScript("document. getElementById('UserName'). value=‘training@jalaacademy.com'");

}

@Test (priority = 3)

public void get\_ele\_attribute()

{

WebElement l = driver.findElement(By.id("UserName"));

l.sendKeys("training@jalaacademy.com");

String val = l.getAttribute("value");

System.out.println("Entered text is: " + val);

}

@Test (priority = 4)

public void clearText()

{

driver. findElement (By. xpath("//input[@type='text']")).sendKeys("training@jalaacademy.com");

driver.findElement(By.xpath("//input[@type='text']")).clear();

}

@Test (priority = 5)

public void checkText()

{

boolean check = driver. findElement(By.xpath("//input[@type='text']")).isEnabled();

Assert.assertEquals(check,true);

}

public void closeBrowser()

{

driver.close();

}

}

Radio Button / Check Box

1. Selecting the Radio Button by Label Text / value.

2. Find out number of elements in a radio group.

3. Find out all radio button values.

4. How to get the selected radio button label text?

5. Check if Radio Button is selected?

6. Check if Radio Button is enabled or disabled?

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

public class radioBox

{

WebDriver driver;

public void launchBrowser()

{

System.setProperty("webdriver. chrome. driver","C:\\Users\\Asus\\Downloads\\chromedriver\_win32\\chromedriver.exe");

driver = new ChromeDriver();

driver.manage().deleteAllCookies();

driver.get("http://magnus.jalatechnologies.com/");

}

@Test (priority = 1)

public void radioTest() throws InterruptedException

{

driver. findElement (By. xpath("//input[@type='text']")). sendKeys("training@jalaacademy.com");

driver.findElement(By.id("Password")).sendKeys("jobprogram");

Thread.sleep(2000);

driver.findElement(By.xpath("//button[@type='button' and @class='btn btn-primary btn-flat']")).click();

Thread.sleep(2000);

driver.findElement(By.linkText("Employee")).click();

driver.findElement(By.linkText("Create")).click();

driver.findElement(By.id("rdbMale")).click();

}

public void closeBrowser()

{

driver.close();

}

}

Popups/Alerts and Windows

1. Capturing the alert message using getText().

@Test

public void T01\_AlertTest()

{

driver. navigate().to("http://www.w3schools.com/js/tryit.asp?filename=tryjs\_alert");

driver.switchTo().frame("iframeResult");

WebElement alertButton = driver.findElement(By.cssSelector("html>body>button"));

alertButton.click();

String expectedAlertMessage = "I am an alert box!";

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

Assertions.assertEquals(expectedAlertMessage, actualAlertMessage);

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

}

2. Prompt Alert with Text Box to enter the text.

let text;  
let favDrink = prompt ("What's your favorite cocktail drink?");  
switch(favDrink) {  
  case "Coca-Cola":  
    text = "Excellent choice! Coca-Cola is good for your soul.";  
    break;  
  case "Pepsi":  
    text = "Pepsi is my favorite too!";  
    break;  
  case "Sprite":  
    text = "Really? Are you sure the Sprite is your favorite?";  
    break;  
  default:  
    text = "I have never heard of that one!";  
}

1. Confirmation Alert with Ok and Cancel buttons.

let text;  
if (confirm ("Press a button!") == true)

{  
  text = "You pressed OK!";  
} else {  
  text = "You cancelled!";  
}

4. Clicking OK button of the alert using accept().

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.Alert;

public class AlertAccept{

   public static void main (String[] args)

{

      System.setProperty("webdriver.chrome.driver", "C:\Users\ghs6kor\Desktop\Java\chromedriver.exe");

      WebDriver driver = new ChromeDriver();

      String url ="https://www.tutorialspoint.com/selenium/selenium\_automation\_practice.htm";

      driver.get(url);

      driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

      // identify element

      driver. findElement(By.xpath("//button[@name='submit']")).click();

      Alert al = driver.switchTo().alert();

      al.accept();

      driver.quit();

   }

}

5. Clicking Cancel button of the alert using dismiss ().

("#complexConfirm"). confirm ({

title: “Delete confirmation",

text: “This is very dangerous; you shouldn't do it! Are you sure?",

confirm: function(button) {

alert ("You just confirmed.");

},

cancel: function(button) {

alert ("You aborted the operation.");

},

confirmButton: "Yes I am",

cancelButton: "No"

});

6. Handle single window using driver.getWindowHandle().

@TestInstance (TestInstance.Lifecycle.PER\_CLASS)

@TestMethodOrder (MethodOrderer.MethodName.class)

public class Windows

{

private WebDriver driver;

public void setupTest()

{

WebDriverManager.chromedriver(). setup ();

driver = new ChromeDriver();

}

@Test

@SneakyThrows

public void T02\_ManageWindows ()

{

driver. navigate().to("http://www.w3schools.com/tags/tryit.asp?filename=tryhtml\_link\_target");

driver.manage().window().maximize();

Dimension windowSize = driver.manage().window().getSize();

System.out.println("\*\*\*Full Size Values for Current Window\*\*\*\n");

System.out.println("Screen Width: " + windowSize.getWidth() + "\n");

System.out.println("Screen Height: " + windowSize.getHeight() + "\n");

Thread.sleep(500);

driver.manage().window().setSize(new Dimension(windowSize.getWidth() / 4, windowSize.getHeight() / 4));

Dimension quarterWindowSize = driver.manage().window().getSize();

System.out.println("\*\*\* 1/4 Size Values for Current Window\*\*\*\n");

System.out.println("Screen Width: " + quarterWindowSize.getWidth() + "\n");

System.out.println("Screen Height: " + quarterWindowSize.getHeight() + "\n");

Thread.sleep(500);

Point windowPosition = driver.manage().window().getPosition();

System.out.println("\*\*\* Window Position for Current Window\*\*\*\n");

System.out.println("Window X position: " + windowPosition.getX() + "\n");

System.out.println("Window Y position: " + windowPosition.getY() + "\n");

Thread.sleep(500);

Point newWindowPosition = windowPosition.moveBy(200, 200);

driver.manage().window().setPosition(newWindowPosition);

System.out.println("\*\*\* Window Position for Current Window\*\*\*\n");

System.out.println("Window X position: " + driver.manage().window().getPosition().getX() + "\n");

System.out.println("Window Y position: " + driver.manage().window().getPosition().getY() + "\n");

Thread.sleep(500);

}

public void tearDown()

{

driver.quit();

}

}

7. Handle multiple windows using driver.getWindowHandles().

@TestInstance (TestInstance.Lifecycle.PER\_CLASS)

@TestMethodOrder (MethodOrderer.MethodName.class)

public class Windows

{

private WebDriver driver;

public void setupTest()

{

WebDriverManager.chromedriver(). setup ();

driver = new ChromeDriver();

}

public void tearDown()

{

driver.quit();

}

public void T01\_SwitchToWindows ()

{

driver. navigate().to("http://www.w3schools.com/tags/tryit.asp?filename=tryhtml\_link\_target");

driver. Manage (). window().maximize();

System.out.println("Current Window Handle: " + driver.getWindowHandle() + "\n");

driver.switchTo().frame("iframeResult");

WebElement visitLink = driver.findElement(By.linkText("Visit W3Schools.com!"));

visitLink.click();

Set<String> windowHandles = driver.getWindowHandles();

List<String> windowHandlesList = new ArrayList<>(windowHandles); //Set to List Conversion

System.out.println("Total window number: " + windowHandlesList.size() + "\n");

driver.switchTo().window(windowHandlesList.get(1));

System.out.println("Current Window Handle: " + driver.getWindowHandle() + "\n");

WebElement logo = driver.findElement(By.cssSelector(".fa.fa-logo"));

Assertions.assertTrue(logo.isDisplayed());

driver.switchTo().window(windowHandlesList.get(0));

System.out.println("Current Window Handle: " + driver.getWindowHandle() + "\n");

WebElement seeResultButton = driver. findElement(By.cssSelector("button[onclick\*='submitTryit(1)'"));

Assertions.assertTrue(seeResultButton.getText(). contains ("Run ❯"));

}

}

8. Switch to window using driver.switchTo().window().

String winHandleBefore = driver.getWindowHandle();

for(String winHandle : driver.getWindowHandles())

{

driver.switchTo().window(winHandle);

}

driver.close();

driver.switchTo().window(winHandleBefore);

9. Switch to frame using driver.switchTo().frame().

public class SwitchToframe

{

public static void main (String[] args) throws NoSuchElementException

{

WebDriver driver = new FirefoxDriver();

driver.get("http://demo.guru99.com/test/guru99home/");

driver.manage().window().maximize();

int size = driver.findElements(By.tagName("iframe")).size();

for (int i=0; i<=size; i++)

{

driver.switchTo().frame(i);

int total=driver.findElements(By.xpath("html/body/a/img")).size();

System.out.println(total);

driver.switchTo().defaultContent(); //switching back from the iframe

}

driver.switchTo().frame(0); //Switching to the frame

System.out.println("We are switched to the iframe");

driver.findElement(By.xpath("html/body/a/img")).click();

System.out.println("We are done");

}

}

10. Switch to popup using driver.switchTo.alert().

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class DemoWebAlert

{

                WebDriver driver;

                public DemoWebAlert()

{

                }

                public void setUp()

{

                                driver=new FirefoxDriver();

                                driver.get("file:///F:/Work/Selenium/Testing-Presentation/DemoWebPopup.htm");

                                driver. manage (). window (). maximize ();

                }

                public void testWebAlert() throws InterruptedException

                {

                                driver. find Element (By.xpath("//button[contains(text(),'Try it')]")).click();

                                Thread.sleep(5000);

                                Alert alert = driver.switchTo().alert();

                                alert.accept();

                                driver. find Element (By.xpath("//button[contains(text(),'Try it')]")).click();

                                Thread. Sleep (5000);

                                driver. switchTo(). alert (). dismiss ();

                                driver. findElement(By.xpath("//button[contains(text(),'Try it')]")).click();

                                Thread. Sleep (5000);

                                System.out.println(driver. switchTo(). alert().getText());

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

                }

                public void tearDown() {

                    driver. Quit ();

                }

}

Selenium Miscellaneous Scenarios

1. Write a test case to capture the screenshots with WebDriver.

import java.io.File;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.Test;

public class BStackTakeScreenshot

{

public void testBStackTakeScreenShot() throws Exception

{

WebDriver driver ;

System.setProperty("webdriver. firefox.marionette","c:\\geckodriver.exe");

driver = new FirefoxDriver();

driver. Get("https://www.browserstack.com");

this.takeSnapShot(driver, "c://test.png") ;

}

public static void takeSnapShot(WebDriver webdriver, String fileWithPath) throws Exception

{

TakesScreenshot scrShot =((TakesScreenshot)webdriver);

File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

File DestFile=new File(fileWithPath);

FileUtils.copyFile(SrcFile, DestFile);

}

}

2. Click on an element which is in iFrame.

var iframe = document.getElementById("myFrame");  
var elmnt=iframe.contentWindow.document.getElementsByTagName("H1")[0];  
elmnt.style.display = "none";

3. Find out the broken links on a web page.

import java.io.IOException;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL.

import java.util.Iterator;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class BrokenLinks

{

private static WebDriver driver = null;

public static void main (String [] args)

{

String homePage = "http://www.zlti.com";

String url = "";

HttpURLConnection huc = null;

int respCode = 200;

driver = new ChromeDriver();

driver.manage().window().maximize();

driver.get(homePage);

List<WebElement> links = driver.findElements(By.tagName("a"));

Iterator<WebElement> it = links.iterator();

while(it.hasNext())

{

url = it.next().getAttribute("href");

System.out.println(url);

if (url == null || url.isEmpty()){

System.out.println("URL is either not configured for anchor tag or it is empty");

continue;

}

if(!url.startsWith(homePage))

{

System.out.println("URL belongs to another domain, skipping it.");

continue;

}

try {

huc = (HttpURLConnection)(new URL(url).openConnection());

huc.setRequestMethod("HEAD");

huc.connect();

respCode = huc.getResponseCode();

if(respCode >= 400)

{

System.out.println(url+" is a broken link");

}

else {

System.out.println(url+" is a valid link");

}

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

driver.quit();

}

}

4. Implicit and Explicit wait commands.

***Implicit wait***

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class ImplicitWait

{

public static void main (String[] args) throws InterruptedException

{

System.setProperty("webdriver. chrome.driver", "C:Selenium-java-

javaTpointchromedriver\_win32chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

driver.manage().deleteAllCookies();

driver.manage().timeouts().pageLoadTimeout(40,

TimeUnit.SECONDS);

driver. manage (). timeouts ().implicitlyWait(20, TimeUnit.SECONDS);

driver.get("https://login.google.com/");

driver.findElement(By.xpath("//input[@id='login-

username']")). sendKeys("JavaTpoint.com");

Thread.sleep(1000);

driver. findElement(By.xpath("//input[@id='login-signin']")).click();

}

}

***Explicit wait***

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class Locators

{

public static void main (String [] args) throws InterruptedException

{

System.setProperty("webdriver. chrome.driver", "C:Selenium-java-

javatpointchromedriver\_win32chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

driver.manage().deleteAllCookies();

driver.manage().timeouts().pageLoadTimeout(40,

TimeUnit.SECONDS);

driver. manage (). timeouts ().implicitlyWait(30, TimeUnit.SECONDS);

driver.get("https://www.facebook.com/");

WebElement firstname= driver. findElement(By.name("firstname"));

WebElement lastname= driver.findElement(By.name("lastname"));

sendKeys(driver, firstname, 10, "Edureka");

sendKeys(driver, lastname, 20, "Edureka");

WebElement forgotAccount=

driver.findElement(By.linkText("Forgotten account?"));

clickOn(driver,forgotAccount, 10);

driver. manage (). timeouts ().implicitlyWait(30, TimeUnit.SECONDS);

}

public static void sendKeys (WebDriver driver1, WebElement element,

int timeout, String value)

{

new WebDriverWait(driver1,

timeout). until (ExpectedConditions.visibilityOf(element));

element.sendKeys(value);

}

public static void clickOn (WebDriver driver1, WebElement element,

int timeout)

{

new WebDriverWait(driver1,

timeout). until (ExpectedConditions.elementToBeClickable(element));

element.click();

}

}

1. Action class with the following operations

Keyboard key press event

Pressing enter button on the keyboard

ClickAndHold event,

Drag and Drop MoveToElement,

Mouse Hover Event Double Click event

public static void main (String[] args)

{

String baseUrl = "http://www.facebook.com/";

WebDriver driver = new FirefoxDriver();

driver.get(baseUrl);

WebElement txtUsername = driver.findElement(By.id("email"));

Actions builder = new Actions(driver);

Action seriesOfActions = builder

.moveToElement(txtUsername)

. click()

.keyDown(txtUsername, Keys.SHIFT)

.sendKeys(txtUsername, "hello")

.keyUp(txtUsername, Keys.SHIFT)

.doubleClick(txtUsername)

.contextClick()

. build();

seriesOfActions.perform() ;

}

6.Web Table operations

Get row count

Get data from a specific cell

Dynamic web table handling

|  |
| --- |
| Import unittest |
|  | import time |
|  | from selenium import webdriver |
|  | from selenium.webdriver.support.select import Select |
|  | from selenium.webdriver.common.by import By |
|  | from selenium.webdriver.support.ui import WebDriverWait |
|  | from selenium.webdriver.support import expected\_conditions as EC |
|  |  |
|  | test\_url = "https://www.w3schools.com/html/html\_tables.asp" |
|  |  |
|  | class WebTableTest(unittest.TestCase): |
|  |  |
|  | def setUp(self): |
|  | self.driver = webdriver.Chrome() |
|  | self.driver.maximize\_window() |
|  |  |
|  | def test\_1\_get\_num\_rows\_(self): |
|  | driver = self.driver |
|  | driver.get(test\_url) |
|  |  |
|  | WebDriverWait(driver, 60).until(EC.presence\_of\_element\_located((By.CLASS\_NAME, "w3-example"))) |
|  |  |
|  | num\_rows = len (driver.find\_elements\_by\_xpath("//\*[@id='customers']/tbody/tr")) |
|  | print("Rows in table are " + repr(num\_rows)) |
|  |  |
|  | def test\_2\_get\_num\_cols\_(self): |
|  | driver = self.driver |
|  | driver.get(test\_url) |
|  |  |
|  | WebDriverWait(driver, 60).until(EC.presence\_of\_element\_located((By.CLASS\_NAME, "w3-example"))) |
|  | # num\_cols = len (driver.find\_elements\_by\_xpath("//\*[@id='customers']/tbody/tr/th")) |
|  | num\_cols = len (driver.find\_elements\_by\_xpath("//\*[@id='customers']/tbody/tr[2]/td")) |
|  | print("Columns in table are " + repr(num\_cols)) |
|  |  |
|  | def tearDown(self): |
|  | self.driver.close() |
|  | self.driver.quit() |
|  |  |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | unittest.main() |

7. Handling Ajax Auto suggestion.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.Assert;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class Ajaxdemo

{

private String URL = "http://demo.guru99.com/test/ajax.html";

WebDriver driver;

WebDriverWait wait;

public void setUp()

{

System.setProperty("webdriver.chrome.driver",".\\chromedriver.exe");

driver = new ChromeDriver();

driver.manage().window().maximize();

driver.navigate().to(URL);

}

public void test\_AjaxExample()

{

By container = By.cssSelector(".container");

wait = new WebDriverWait(driver, 5);

wait.until(ExpectedConditions.presenceOfElementLocated(container));

WebElement noTextElement = driver.findElement(By.className("radiobutton"));

String textBefore = noTextElement.getText().trim();

driver.findElement(By.id("yes")).click();

driver.findElement(By.id("buttoncheck")).click();

WebElement TextElement = driver.findElement(By.className("radiobutton"));

wait.until(ExpectedConditions.visibilityOf(TextElement));

String textAfter = TextElement.getText().trim();

Assert.assertNotEquals(textBefore, textAfter);

System.out.println("Ajax Call Performed");

String expectedText = "Radio button is checked and it's value is Yes";

Assert.assertEquals(textAfter, expectedText);

driver.close();

}

}

8. Select a specific date from a calendar.

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class DateTimePicker

{

public void dateTimePicker()

{

System.setProperty("webdriver.chrome.driver", "chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://demo.guru99.com/test/");

WebElement dateBox = driver.findElement(By.xpath("//form//input[@name='bdaytime']"));

dateBox.sendKeys("09252013");

dateBox.sendKeys(Keys.TAB);

dateBox.sendKeys("0245PM");

}

}

TestNG

1. Know the order of execution of TestNG annotations.

import org.testng.annotations.AfterClass;  
import org.testng.annotations.AfterMethod;  
import org.testng.annotations.AfterSuite;  
import org.testng.annotations.AfterTest;  
import org.testng.annotations.BeforeClass;  
import org.testng.annotations.BeforeMethod;  
import org.testng.annotations.BeforeSuite;  
import org.testng.annotations.BeforeTest;  
import org.testng.annotations.Test;

public class TestClass1

{

public void beforeSuite()

{  
System.out.println(“Inside Before Suite Method.”);  
}

public void beforeClass()

{  
System.out.println(“Inside Before Class of TestClass1.”);  
}

public void beforeTest()

{  
System.out.println(“Inside Before Test Method of TestClass1.”);  
}

public void beforeMethod1()

{  
System.out.println(“Inside Before Method 1 of TestClass1.”);  
}

public void beforeMethod2()

{  
System.out.println(“Inside Before Method 2 of TestClass1.”);  
}

public void testMethod1()

{  
System.out.println(“Inside Test Method 1 of TestClass1.”);  
}

public void testMethod2()

{  
System.out.println(“Inside Test Method 2 of TestClass1.”);  
}

public void afterTest()

{  
System.out.println(“Inside After Test Method of TestClass1.”);  
}

public void afterMethod1()

{  
System.out.println(“Inside After Method 1 of TestClass1.”);  
}

public void afterMethod2()

{  
System.out.println(“Inside After Method 2 of TestClass1.”);  
}

public void afterClass()

{  
System.out.println(“Inside After Class Method of TestClass1.”);  
}

public void afterSuite()

{  
System.out.println(“Inside After Suite Method.”);  
}

}

2. Create testing.xml file to run the test cases in a class file.

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">  
<suite name="Practice">  
 <test name="Test1">  
     <parameter name="Username" value="user123@gmail.com"/>  
     <parameter name="password" value="test@123"/>  
     <parameter name="mobile\_number" value="1234567899"/>  
     <classes>  
       <class name="Practice.Test1"/>  
     </classes>  
 </test><!-- Test -->  
</suite><!-- Test -->

3. Create a test suite and test groups.

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >

<suite name="Test-Suite" >

<test name="ToolsQA" >

<classes>

<class name="TestNG" />

</classes>

</test>

</suite>

<groups>

<run>

<include name="bonding" />

</run>

</groups>

4. List down the assertions and use them in a test case.

public class LearnAssertions

{

 WebDriver driver;

 String path = System.getProperty("user.dir");

public void SetDriver()

{

System.setProperty("webdriver. chrome.driver",path+"\\Drivers\\chromedriver.exe");

driver = new ChromeDriver();// Object is created- Chrome browser is opened

driver.manage().window().maximize();

}

public void verifyTitle()

{

driver.get(https://www.amazon.com);

String ActualTitle = driver.getTitle();

String ExpectedTitle = “Welcome to Amazon”;

Assert.assertEquals(ActualTitle, ExpectedTitle);

System.out.println(“Assert passed”);

}

public void closedriver()

{

driver.close();

}

public class LearnAssertionsSoft

{

WebDriver driver;

SoftAssert softassert = new SoftAssert();

SoftAssert softassert2 = new SoftAssert();

String path = System.getProperty("user.dir");

public void SetDriver()

{

System.setProperty("webdriver. chrome.driver",path+"\\Drivers\\chromedriver.exe");

driver = new ChromeDriver();// Object is created - Chrome browser is opened

driver.manage().window().maximize();

}

public void verifyTitle()

{

driver.get("https://amazon.in");

String ActualTitle = driver.getTitle();

System.out.println("Actual Title :"+ActualTitle);

String ExpectedTitle = "cameras, books, watches, apparel, shoes and e-Gift

Cards. Free Shipping &amp; Cash on Delivery Available.";

softassert.assertEquals(ActualTitle, ExpectedTitle);

System.out.println("Assertion 1 is executed”);

softassert.assertAll();

}

public void verifyElement()

{

WebElement AmazonIcon =

driver. findElement(By.Xpath(“//div[contains(@id,’amazon\_icon’)]);

softassert2.assertEquals (true, AmazonIcon.isDisplayed());

softassert2.assertAll();

System.out.println("Icon is displayed");

System.out.println("Assertion 2 is executed”);

}

public void closedriver()

{

driver.close();

}

}

5. Disable or ignore a test from running.

@Test(enabled = true)

public class DisableTestDemo

{

@Test(enabled = true)

public void testMethodOne()

{

System.out.println("Test method one.");

}

@Test(enabled = false)

public void testMethodTwo()

{

System.out.println("Test method two.");

}

public void testMethodThree()

{

System.out.println("Test method three.");

}

}

6. Make one test script dependent on the other and run both of them.

public class DependentTestExamples

{

@Test (dependsOnMethods = { "testTwo", "testThree" })

public void testOne() {

System.out.println("Test method one");

}

public void testTwo()

{

System.out.println("Test method two");

}

public void testThree()

{

System.out.println("Test method three");

}

}

7. Set priority to all the tests, execute and observe the order of execution.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

public class Priority\_In\_testNG

{

WebDriver driver;

@Test (priority=1)

public void openBrowser()

{

driver = new FirefoxDriver();

}

@Test (priority=2)

public void launchGoogle()

{

driver.get("http://www.google.co.in");

}

@Test (priority=3)

public void peformSeachAndClick1stLink()

{

driver.findElement(By.xpath(".//\*[@title='Search']")).sendKeys("Facebook");

}

@Test (priority=4)

public void FaceBookPageTitleVerification() throws Exception

{

driver.findElement(By.xpath(".//\*[@value='Search']")).click();

Thread.sleep(3000);

Assert.assertEquals(driver.getTitle().contains("Facebook - Google Search"), true);

}

}

8. How to run the test multiple times using invocationCount.

import org.testng.annotations.Test;

public class MultiTimeRunner

{

@Test(invocationCount=5)

public void testRunner()

{

System.out.println("Sample Test");

}

}

9. Pass parameters to test script.

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class Params

{

@Test

@Parameters ({"val1", "val2"})

public void Sum(int v1, int v2) {

int finalsum = v1 + v2;

System.out.println("The final sum of the given values is " + finalsum);

}

}

10. How to group test cases.

public class Test1

{

  @Test(groups = { "group1", "group2" })

  public void test\_method1()

  {

  }

  @Test(groups = {"group2"} )

  public void test\_method2()

  {

  }

  @Test(groups = {"group1"})

  public void test\_method3()

  {

  }

}

12. Running test cases in parallel.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.chrome.\*;

import org.testng.annotations.Test;

public class ParallelTest

{

public WebDriver driver;

public void FirefoxTest()

{

driver = new FirefoxDriver();

driver.get("https://www.demoqa.com");

driver.findElement(By.xpath("//\*[@id=\"app\"]/div/div/div[2]/div/div[1]/div/div[1]")).click();

driver.quit();

}

public void ChromeTest()

{

driver = new ChromeDriver();

driver.get("https://www.demoqa.com");

driver.findElement(By.xpath("//\*[@id=\"app\"]/div/div/div[2]/div/div[1]/div/div[1]")).click();

driver.quit();

}

}

Selenium Framework Steps

1. Write 5 Test scripts using main method on Registration page in a different class.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

public class DemoTestNG

{

       public WebDriver driver = new FirefoxDriver();

       String appUrl = &quot; https://accounts.google.com&quot.

public void gmailLogin()

{

       driver.get(&quot;https://gmail.com&quot;);

       driver.manage().window().maximize();

       String expectedTitle = &quot; Sign in - Google Accounts &quot;;

       String actualTitle = driver.getTitle();

       Assert.assertEquals(expectedTitle,actualTitle);

       WebElement username = driver.findElement(By.id(&quot;Email&quot;));

       username.clear();

       username.sendKeys(&quot;TestSelenium&quot;);

WebElement password = driver.findElement(By.id(&quot;Passwd&quot;));

       password.clear();

       password.sendKeys(&quot;password123&quot;);

       WebElement SignInButton = driver.findElement(By.id(&quot;signIn&quot;));

       SignInButton.click();

       driver.close();

}

}

2. Write all the above 5 Test scripts in a single class using TestNG annotations.

|  |
| --- |
| public class test  {    public void beforeMethod()  {  System.out.println(" Before Method will execute before every test method");  }    public void afterMethod()  {  System.out.println("After Method will execute after every test method ");  }    public void beforeClass()  {  System.out.println("Before Class will always execute prior to Before Method and Test Method ");  }    public void afterClass()  {  System.out.println("After Class will always execute later to After Method and Test method");  }    public void beforeTest()  {  System.out.println("Before Test will always execute prior to Before Class, ,Before Method and Test Method ");  }    public void afterTest()  {  System.out.println("After Test will always execute later to After Method, After Class ");  }    public void beforeSuite()  {  System.out.println(“Before Suite will always execute prior to all annotations or tests in the suite.");  }  public void afterSuite()  {  System.out.println("After suite will always execute at last when all the annotations or test in the suite have run.");  }    public void testCase1()  {  System.out.println("This is my First Test Case 1");  }    public void testCase2()  {  System.out.println("This is my Second Test Case 2");  }  }  3. Run the individual test scripts and the entire suite testing.xml. |
|  |
| <!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >  <suite name="Test-Suite" >  <test name="ToolsQA" >  <classes>  <class name="TestNG" />  </classes>  </test>  </suite> |

4. Use all the TestNG annotations and run the scripts.

import org.testng.annotations.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

public class Test1

{

public String baseUrl = "https://www.browserstack.com/";

String driverPath = "D:\\Selenium\\chromedriver.exe";

public WebDriver driver ;

public void launchBrowser()

{

System.out.println("launching Chrome browser");

System.setProperty("webdriver.chrome.driver", driverPath);

driver = new ChromeDriver();

driver.get(baseUrl);

}

public void verifyHomepageTitle()

{

String expectedTitle = "Most Reliable App & Cross Browser Testing Platform | BrowserStack";

String actualTitle = driver.getTitle();

Assert.assertEquals(actualTitle, expectedTitle);

}

@AfterTest

public void terminateBrowser(){

driver.close();

}

}

6. Create PageObjects class for all the page objects of an application.

import static org.testng.Assert.assertEquals;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class BrowserStackHomePage

{

WebDriver driver;

By Header=By.xpath("//h1");

By getStarted=By.xpath("//\*[@id='signupModalButton']");

public BrowserStackHomePage(WebDriver driver)

{

this.driver=driver;

}

public void veryHeader()

{

String getheadertext=driver.findElement(Header).getText();

assertEquals("App & Browser Testing Made Easy", getheadertext);

}

public void clickOnGetStarted()

{

driver.findElement(getStarted).click();

}

}

7. Create PageActions class for all the operations that are performed on the web elements.

public class Test\_Login

{

  WebDriver driver;

    PageActions\_Login actionLogin;

    public void setup()

{

        driver = new FirefoxDriver();

        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        driver.get("http://gmail.com/");

    }

@Test(priority = 0)

    public void SignIntoGMailInvalidPassword()

{

        actionLogin = new PageActions\_Login(driver);

        actionLogin.enterUserIDPassword("upadhyay40", "xyz");

        String loginPageTitle = actionLogin.getWrongPasswordTextMessage();

        Assert.assertTrue(loginPageTitle.contains("The email and password you entered don't match."));

    }

8. Create a separate Test Suite file for each Module.

import org.testng.annotations.Test;

public class TestDatabase

{

@Test(groups = "db")

public void testConnectOracle()

{

System.out.println("testConnectOracle()");

}

@Test(groups = "db")

public void testConnectMsSQL()

{

System.out.println("testConnectMsSQL");

}

@Test(groups = "db-nosql")

public void testConnectMongoDB()

{

System.out.println("testConnectMongoDB");

}

@Test(groups = { "db", "brokenTests" })

public void testConnectMySQL()

{

System.out.println("testConnectMySQL");

}

}

10. Group the test cases using TestNG Groups attribute.

import org.testng.annotations.Test;

public class Personal\_loan

{

 @Test(groups= {"SmokeTest"})

 public void WebLoginPersonalLoan()

 {

     System.out.println("Web Login Personal Loan");

 }

 public void MobileLoginPersonalLoan()

 {

     System.out.println("Mobile Login Personal Loan");

 }

 public void APILoginPersonalLoan()

 {

     System.out.println("API Login Personal Loan");

 }

}

11. Prioritize the test scripts using TestNG priority attribute.

import org.testng.annotations.Test;

public class NoPriorityEx

{

public void one()

{

System.out.println("First");

}

public void two()

{

System.out.println("Second");

}

public void three()

{

System.out.println("Third");

}

}

12. Create a BaseTest and have common code in the BaseTest, All Test Suites should inherit the BaseTest.

|  |
| --- |
| public abstract class BaseWeb { |
|  |  |
|  | protected static WebDriver driver; |
|  |  |
|  |  |
|  | static void webdrivermanagerSetup()  { |
|  | WebDriverManager.chromedriver().setup(); |
|  | driver = new ChromeDriver(); |
|  | driver.get("http://eliasnogueira.com"); |
|  | } |
|  |  |
|  |  |
|  | static void quitBrowser()  { |
|  | driver.quit(); |
|  | } |
|  | } |

13. Use loggers.

import java.io.FileInputStream;

import java.io.IOException;

import java.util.logging.ConsoleHandler;

import java.util.logging.FileHandler;

import java.util.logging.Handler;

import java.util.logging.Level;

import java.util.logging.LogManager;

import java.util.logging.Logger;

public class LoggingExample

{

static Logger logger = Logger.getLogger(LoggingExample.class.getName());

public static void main (String[] args)

{

try

{

LogManager.getLogManager().readConfiguration(new FileInputStream("mylogging.properties"));

}

catch (SecurityException | IOException e1)

{

e1. printStackTrace();

}

logger.setLevel(Level.FINE);

logger.addHandler(new ConsoleHandler());

logger.addHandler(new MyHandler());

try

{

Handler fileHandler = new FileHandler("/Users/pankaj/tmp/logger.log", 2000, 5); fileHandler.setFormatter(new MyFormatter());

fileHandler.setFilter(new MyFilter()); logger.addHandler(fileHandler);

for (int i=0; i<1000; i++)

{

//logging messages logger.log (Level.INFO, "Msg"+i);

}

logger.log (Level.CONFIG, "Config data");

}

catch (SecurityException | IOException e)

{

e.printStackTrace();

}

}

}

14. Create a config file and pass the constant values from config file.

|  |
| --- |
| [APP] |
|  | ENVIRONMENT = test |
|  | DEBUG = True |
|  | # Only accept True or False |
|  |  |
|  | [DATABASE] |
|  | USERNAME = xiaoxu |
|  | PASSWORD = xiaoxu |
|  | HOST = 127.0.0.1 |
|  | PORT = 5432 |
|  | DB = xiaoxu\_database |

15. Know how to debug the Scripts.

using System;

using System.Collections.Generic;

namespace ConsoleApp\_FirstApp

{

class Program

{

static void Main (string [] args)

{

Console.WriteLine("Welcome to Galaxy News!");

IterateThroughList();

Console.ReadKey();

}

private static void IterateThroughList()

{

var theGalaxies = new List<Galaxy>

{

new Galaxy () {Name="Tadpole", MegaLightYears=400, GalaxyType=new GType('S')},

new Galaxy () {Name="Pinwheel", MegaLightYears=25, GalaxyType=new GType('S')},

new Galaxy () {Name="Cartwheel", MegaLightYears=500, GalaxyType=new GType('L')},

new Galaxy () {Name="Small Magellanic Cloud", MegaLightYears=.2, GalaxyType=new GType('I')},

new Galaxy () {Name="Andromeda", MegaLightYears=3, GalaxyType=new GType('S')},

new Galaxy () {Name="Maffei 1", MegaLightYears=11, GalaxyType=new GType('E')}

};

foreach (Galaxy theGalaxy in theGalaxies)

{

Console.WriteLine(theGalaxy.Name + ““ + theGalaxy.MegaLightYears + ", " + theGalaxy.GalaxyType);

}

}

}

public class Galaxy

{

public string Name {get; set; }

public double MegaLightYears {get; set; }

public object GalaxyType {get; set; }

}

public class GType

{

public GType(char type)

{

switch(type)

{

case 'S':

MyGType = Type.Spiral;

break;

case 'E':

MyGType = Type.Elliptical;

break;

case 'l':

MyGType = Type.Irregular;

break;

case 'L':

MyGType = Type.Lenticular;

break;

default:

break;

}

}

public object MyGType {get; set; }

private enum Type {Spiral, Elliptical, Irregular, Lenticular}

}

}

16. Run the scripts in all browsers.

import org.openqa.Selenium.WebDriver;

importorg.openqa.Selenium.htmlunit.HtmlUnitDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

publicclassvefifyTestTitle

{

publicvoidverifyFacebookTitle()

{

WebDriver driver = newHtmlUnitDriver(true);

driver.get("http://www.facebook.com");

String facebook\_Title= driver.getTitle();

Assert.assertTrue(facebook\_Title.contains("Facebook"));

System.out.println(facebook\_Title);

}

}

17. Parameterize the scripts using excel, Running the same script with multiple set of test data.

|  |
| --- |
| Import org.openqa.selenium.By; |
|  | import org.openqa.selenium.Keys; |
|  | import org.openqa.selenium.WebDriver; |
|  | import org.openqa.selenium.WebElement; |
|  | import org.openqa.selenium.chrome.ChromeDriver; |
|  | import org.testng.Reporter; |
|  | import org.testng.annotations.AfterMethod; |
|  | import org.testng.annotations.BeforeMethod; |
|  | import org.testng.annotations.DataProvider; |
|  | import org.testng.annotations.Test; |
|  |  |
|  | public class SimpleTest  { |
|  |  |
|  | WebDriver driver; |
|  |  |
|  | @DataProvider(name = "test-data") |
|  | public Object[][] dataProvFunc(){ |
|  | return new Object[][]{ |
|  | {"Lambda Test"},{"Automation"} |
|  | }; |
|  | } |
|  |  |
|  |  |
|  | public void setUp()  { |
|  |  |
|  | System.out.println("Start test"); |
|  | System.setProperty("webdriver.chrome.driver", "E:\\chromedriver.exe"); |
|  | driver = new ChromeDriver(); |
|  | String url = "https://www.google.com"; |
|  | driver.get(url); |
|  | driver.manage().window().maximize(); |
|  |  |
|  | } |
|  | @Test annotation |
|  | @Test(dataProvider ="test-data") |
|  | public void search(String keyWord)  { |
|  | WebElement txtBox = driver.findElement(By.xpath("//input[@class='gLFyf gsfi']")); |
|  | txtBox.sendKeys(keyWord); |
|  | Reporter.log("Keyword entered is : " +keyWord); |
|  | txtBox.sendKeys(Keys.ENTER); |
|  | Reporter.log("Search results are displayed."); |
|  | } |
|  |  |
|  |  |
|  | public void burnDown()  { |
|  | driver.quit(); |
|  | } |
|  |  |
|  | } |

18. Parameterize the scripts using DataProvider.

|  |
| --- |
| import org.openqa.selenium.By; |
|  | import org.openqa.selenium.Keys; |
|  | import org.openqa.selenium.WebDriver; |
|  | import org.openqa.selenium.WebElement; |
|  | import org.openqa.selenium.chrome.ChromeDriver; |
|  | import org.testng.Reporter; |
|  | import org.testng.annotations.AfterMethod; |
|  | import org.testng.annotations.BeforeMethod; |
|  | import org.testng.annotations.Test; |
|  |  |
|  | public class TestClass  { |
|  | WebDriver driver; |
|  |  |
|  |  |
|  | public void setUp()  { |
|  | System.out.println("Start test"); |
|  | System.setProperty("webdriver.chrome.driver", "E:\\chromedriver.exe"); |
|  | driver = new ChromeDriver(); |
|  | String url = "https://www.google.com"; |
|  | driver.get(url); |
|  | driver.manage().window().maximize(); |
|  | } |
|  |  |
|  | @Test(dataProvider ="test-data", dataProviderClass=DPClass.class) |
|  | public void search(String keyWord)  { |
|  | WebElement txtBox = driver.findElement(By.xpath("//input[@class='gLFyf gsfi']")); |
|  | txtBox.sendKeys(keyWord); |
|  | Reporter.log("Keyword entered is : " +keyWord); |
|  | txtBox.sendKeys(Keys.ENTER); |
|  | Reporter.log("Search results are displayed."); |
|  | } |
|  |  |
|  |  |
|  | public void burnDown()  { |
|  | driver.quit(); |
|  | } |
|  | } |

20. Configure extent Report.

import org.junit.AfterClass;

import org.junit.BeforeClass;

import org.junit.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import com.relevantcodes.extentreports.ExtentReports;

import com.relevantcodes.extentreports.ExtentTest;

import com.relevantcodes.extentreports.LogStatus;

public class ExtentDemo

{

static ExtentTest test;

static ExtentReports report;

public static void startTest()

{

report = new ExtentReports(System.getProperty("user.dir")+"ExtentReportResults.html");

test = report.startTest("ExtentDemo");

}

public void extentReportsDemo()

{

System.setProperty("webdriver.chrome.driver", "D:SubmittalExchange\_TFSQAAutomation3rdpartychromechromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("https://www.google.co.in");

if(driver.getTitle().equals("Google"))

{

test.log(LogStatus.PASS, "Navigated to the specified URL");

}

else

{

test.log(LogStatus.FAIL, "Test Failed");

}

}

public static void endTest()

{

report.endTest(test);

report.flush();

}

}