**Individual Test Automation Project**

**Madhumita Walawalkar**

**SJSU ID: 010109697**

Table of Contents

[1. Test Plan Identifier. 3](#_Toc466820726)

[2. Introduction/Test Objective 3](#_Toc466820727)

[3. Test Items 4](#_Toc466820728)

[4. Test Strategy 4](#_Toc466820729)

[5. Tools 5](#_Toc466820730)

[6. Environment 5](#_Toc466820731)

[7. Test Data 6](#_Toc466820732)

[8. Risks and Assumptions 8](#_Toc466820733)

[9. Reports, Results: 9](#_Toc466820734)

[10. Integrations/scheduling. 9](#_Toc466820735)

[Entire Automation Script: 10](#_Toc466820736)

[Data Files: 29](#_Toc466820737)

[Output of entire script: 30](#_Toc466820738)

[Console Output: 30](#_Toc466820739)

# 1. Test Plan Identifier.

The Test Plan identifier details are updated in the below table. (It comprises of author details and the version of the plan)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Author | Email/phone | Version | Identifier | Last updated |
| Madhumita | madhumita.walawalkar@sjsu.edu | 1.0 | AUTO\_TEST\_REG\_TC\_V1 | 11/13/2016 |

# 2. Introduction/Test Objective

The purpose of this Document is to design a test plan for the online website of LinkedIn: https://www.linkedin.com/ as a part of class assignment for CMPE 287.LinkedIn provides a platform for users to build professional connections and helps users to find and provide jobs.

As a part of Automation testing this document will cover all regression test cases for this website which would ensure a data driven, object oriented and programmatic automation approach .Regression testing is done to ensure that the new changes made have not affected the existing functionality of this website.

It will ensure that the website gives expected results when the website encounters valid/invalid input data.

Considering LinkedIn which helps users to connect to other people the automation test scripts will cover all functionalities which will be frequently accessed by majority of users to build professional network. The scope of regression would involve testing of objects including links, images, standard error messages , attempt to register with valid/invalid data from mandatory input registration fields, attempt to log in with valid/invalid credentials and search functionality.

# 3. Test Items

The test items which would be the primary focus of this test plan include:

* Links-All links are clickable and user can navigate to the page and back
* All the images are displayed once the pages are loaded.
* Log-in and sign-out for valid credentials
* Registration-Validation and successful user registration
* Find a colleague functionality
* On-Login Search functionality-Valid and Invalid search results

# 4. Test Strategy

Object Oriented Techniques:

Components on the pages are identified and stored in objects and are referenced with the objects. To test few functionalities properties file is created and is accessed using objects.

The images and links tested comprise of the objects that will be tested on the LinkedIn main page.

Data Driven:

Two property files will be created and used for testing a variety of test cases including checking valid/invalid user, user registration, performing search operation for a valid credential and finding jobs.

These files will store the number of test cases and also the key value pair for each input data to be tested.

Programmatic approach:

In each of the test scripts written wherever required there will be use of loops including while for, if etc. and try-catch for exception handling. Also to implement and store the results Lists, Array lists, variables, etc. will be used. The output messages will be printed on the console and the script execution result will be visible in the JUnit.

String Manipulation:

On the main page there will be a check to verify the agreement text is being displayed before registering. Apart from this all the errors displayed during registration will be checked using in-built functions like equals, contains etc.

Precondition:

The properties file created for data driven approach stores data in key-value pair .Please ensure that unique keys are used to test the data added to these files. Also update these files with the count of the test cases being considered as that determines the number of times the loop will be iterated.

Naming convention:

All test cases are annotated with @Test followed by a test script of the test case. The setup details are stored in the setup method annotated with @Before. The set up details method comprises of the base url and declaration of the properties file used for data drive approach. @After is used to annotate the method which release the resources used for the automation. Each script is written in a separate class file and has set-up, test script and closing of the driver. This approach ensures that even if one script fails to execute, it won’t have an impact on any other script. A new session is opened and then the script is executed and also at the end the browser is closed so that each script is run independently.

Test results for each test case are printed in the console.(Also each script will be in a separate class file to make the code easy to understand and also make it modularized)

# 5. Tools

-Java EE IDE Version Mars Release 4.

-Selenium –java-3.0.1 and geckodriver-v0.11.1

Apart from this external files are created which serve as a data input source for testing all the input fields available on the home page of the website.

# 6. Environment

With Eclipse Java EE and Selenium libraries the automation scripts can be executed. The gecko driver exe paths need to be mentioned in the setup file. A network connection will be required to access the base url.

Based on the console output if there are any errors they would be logged in Bugzilla with the details of the input test data and error details. The associated test automation script will also be added for reference. An access to Bugzilla would be required.

# 7. Test Data

The test data for all the automation scripts is provided in the property file .There are 2 such files which are saved in the” testdata “ the path is as follows:

C:\Users\madhumita\workspaceSPK\TestAuto\testdata

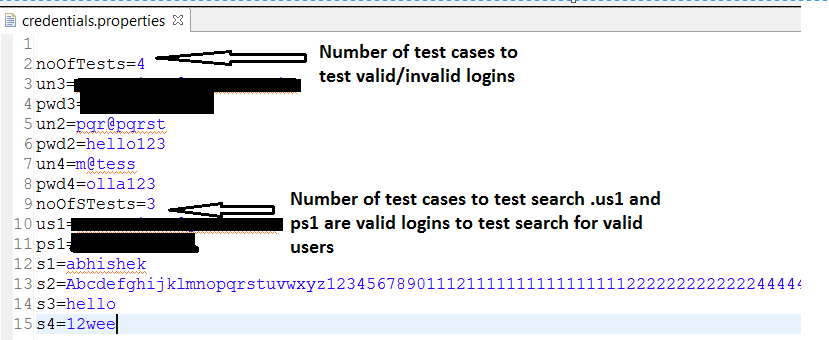
1. credentials.properties:

This file stores data in key value pair. The noOfTests stores total test cases to be executed for verifying the valid /invalid users.

Each username is stored in ‘un’ and password in ‘pwd’ appended by a digit which represents the test case number. E.g un2=abc pwd2=mnq are the credentials for test no.2

It also stored the search data input. The noOfSTests stored number of search operations to be performed and each search string is stored in ‘s’ appended by the test no.

Credentials.properties data:



noOfTests=4

un3= XXXX-Valid credentails

pwd3=XXXX-Valid credentails

un2=pqr@pqrst

pwd2=hello123

un4=m@tess

pwd4=olla123

noOfSTests=3

us1= XXXX-Valid credentials

ps1= XXXX-Valid credentails

s1=abhishek

s2=Abcdefghijklmnopqrstuvwxyz123456789011121111111111111111222222222222244444444444m3eeeettttttttttttttttttttttttttttrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr

s3=hello

s4=12wee

**\*\*us1 and ps1 and un3 and pwd3 are hidden to protect the user privacy.**

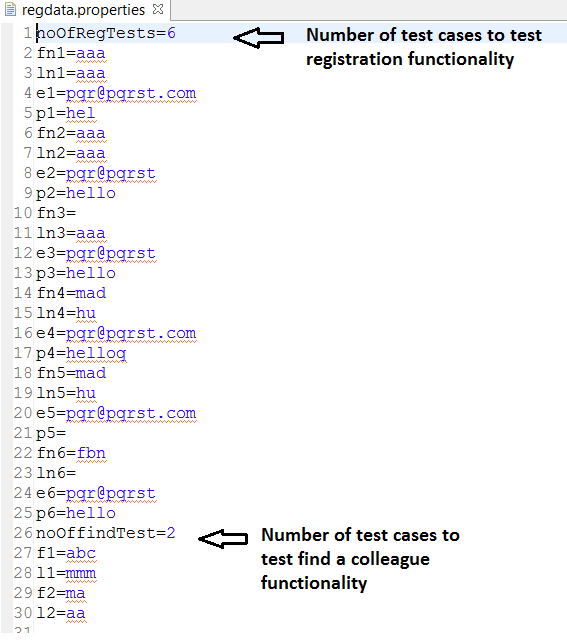
1. reg.properties:

This file stores data in key value pair. The noOfTests stores total test cases to be executed for user registration .It stores all details like fn(first name), ln(last name)

e(email),p(password)appended by the test case number .Each test case picks all the 4 fields and executes the test script.

Apart from this it also stores data to find colleages. The number of test cases are stored in noofTest variable and the firstname is stored in f and last name in l appended by the test case number.

Reg.properties data:



noOfRegTests=6

fn1=aaa

ln1=aaa

e1=pqr@pqrst.com

p1=hel

fn2=aaa

ln2=aaa

e2=pqr@pqrst

p2=hello

fn3=

ln3=aaa

e3=pqr@pqrst

p3=hello

fn4=mad

ln4=hu

e4=pqr@pqrst.com

p4=helloq

fn5=mad

ln5=hu

e5=pqr@pqrst.com

p5=

fn6=fbn

ln6=

e6=pqr@pqrst

p6=hello

noOffindTest=2

f1=abc

l1=mmm

f2=ma

l2=aa

# 8. Risks and Assumptions

The Test automation scripts contain statements of the following type with varying time

driver.manage().timeouts().implicitlyWait(50, TimeUnit.MILLISECONDS)

The test scripts work fine will the mentioned time delay however if the network is very slow or if the page takes much longer than the expected time it might abort the script and execute the next script. In such cases re-running the script without any change or increasing the time might help resolve the issue.

The code is based on the assumption that over-all names of classes, structure of code of the website being tested doesn’t change. If there is any change to the structure, id’s or class names the code might start throwing element not thrown exception. In such cases the code might need an update to check all the elements being considered.

# 9. Reports, Results:

If any of the automations script fail due to Stale data or unable to find element error please increase the timer time and re-run the case. Also verify the elements on the UI are same and have not changed their id’s.

If a script fails during execution the screenshots of the output window with the exception log and the impacting script details will be attached to the defect logged in the bugzilla. The defect will be assigned to Testing team who built the scripts.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Script-Name** | **Data-Driven** | **Programmatic** | **String Manipulation** | **Objects** |
| **1** | (LinkedinPageImages) **testImages()** |  | **✔** | **✔** | **✔** |
| **2** | LinkedinTestLinks **testlinks()** |  | **✔** |  | **✔** |
| **3** | LinkedinTestLogins **testLogins()** | **✔** | **✔** |  |  |
| **4** | LinkedinTestRegistration **testRegistration()** | **✔** | **✔** | **✔** |  |
| **5** | (LinkedinFindAColleague) **testFindAColleague()** | **✔** | **✔** |  |  |
| **6** | LinkedinTestSearch **testsearch()** | **✔** | **✔** | **✔** |  |
| **7** | (LinkedinTestAgreement) **testAgreementText()** |  | **✔** | **✔** |  |

# 10. Integrations/scheduling.

Since this covers all regression test scripts it will executed after the integration of the new changes is done with the existing system.

The execution of automation scripts can be scheduled monthly to ensure any new changes have not affected the existing functionality of the system .In order to run the script out of this cycle due to any requirement will need an approval from the test team lead.

# Entire Automation Script:

LinkedinFindAColleague

package MyPackage;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Properties;

import java.util.concurrent.TimeUnit;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

//this will demonstrate data driven and programmatic approach

public class LinkedinFindAColleague {

private WebDriver driver;

private String baseUrl;

private Properties regdata;

@Before

public void setUp() throws FileNotFoundException, IOException{

regdata = new Properties();

regdata.load(new FileInputStream("testdata/regdata.properties"));

System.setProperty("webdriver.gecko.driver","C://Users//madhumita//Downloads//geckodriver//geckodriver.exe");

driver = new FirefoxDriver();

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

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

baseUrl = "https://www.linkedin.com";

}

/\* Verification of the Find A Colleague functionality using the data driven approach.

\* The search data is present in the regdata.properties file

\* Script written by:Madhumita

\* Date:11/07/2016

\*/

@Test

public void testFindAColleague() {

//extract the number of test cases to be exceuted

int noOfTests = Integer.parseInt(regdata.getProperty("noOffindTest"));

//for test cases where the input not defined the value of null is considered and the loop is //exceuted

while(noOfTests > 0){

driver.get(baseUrl);

WebElement btn=driver.findElement(By.className("submit-btn"));

System.out.println("Button enabled::"+btn.isEnabled());//Check if button //is enabled

//extract data from the properties file

String fname=regdata.getProperty("f"+noOfTests);

String lname=regdata.getProperty("l"+noOfTests);

//set the data on the UI based on name property

driver.findElement(By.name("first")).sendKeys(fname);

driver.findElement(By.name("last")).sendKeys(lname);

System.out.println("Button enabled::"+btn.isEnabled());

try {

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].click();",btn );

driver.manage().timeouts().implicitlyWait(2, TimeUnit.SECONDS);

driver.navigate().back();

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

System.out.println("Error"+e1);//display exception in case an //exception is thrown

}

driver.navigate().back();

noOfTests--;

}

}

@After

public void closureActivities(){

driver.quit();

}

}

LinkedinPageImages

package MyPackage;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.NoSuchElementException;

import org.openqa.selenium.By.ById;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import java.util.ArrayList;

import java.util.List;

import java.util.Properties;

import java.util.concurrent.TimeUnit;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

//This will demonstrate verification of object :Image

public class LinkedinPageImages {

private WebDriver driver;

private String baseUrl;

@Before

public void setUp() throws FileNotFoundException, IOException{

System.setProperty("webdriver.gecko.driver", "C://Users//madhumita//Downloads//geckodriver//geckodriver.exe");

driver = new FirefoxDriver();

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

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

baseUrl = "https://www.linkedin.com";

}

/\*

\* Objects: Check all the images are present

\* and verify if the images are displayed when the page is loaded

\* Script written by:Madhumita

\* Date:11/08/2016

\*/

@Test

public void testImages(){

driver.get(baseUrl);

//extract all images present on the page.

List<WebElement> imagecheck = driver.findElements(By.cssSelector("[data-delayed-url \*='http']"));

int i=0;

//check all images and verify if the image is displayed once the page is loaded.also check if alt //text is present for the image

for(WebElement imageElement : imagecheck){

i++;

String altValue = imageElement.getAttribute("alt");

System.out.println(i+": "+imageElement.getAttribute("data-delayed-url"));

System.out.println("This image is displayed on Page Load:"+imageElement.isDisplayed());

if(altValue != null && !altValue.equals("")){

System.out.println("Alt found as : "+ altValue+"\n"+"-------"+"\n");

}

else{

System.out.println("Alt Text is missing"+"\n"+"-------");

}

}

}

@After

public void closureActivities(){

driver.quit();

}

}

LinkedinTestAgreement

package MyPackage;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Properties;

import java.util.concurrent.TimeUnit;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

//This will demonstrate String Manipulation of Object message displayed on the UI

public class LinkedinTestAgreement {

private WebDriver driver;

private String baseUrl;

@Before

public void setUp() throws FileNotFoundException, IOException{

System.setProperty("webdriver.gecko.driver", "C://Users//madhumita//Downloads//geckodriver//geckodriver.exe");

driver = new FirefoxDriver();

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

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

baseUrl = "https://www.linkedin.com";

}

/\*String manipulation :Verify the agreement text is present above the registration \*button

\* Script written by:Madhumita

\* Date:11/11/2016

\*/

@Test

public void testAgreementText(){

driver.get(baseUrl);

WebElement agg = driver.findElement(By.className("agreement"));

//finds the element by the class name agreement

String aggtext="By clicking Join now, you agree to the LinkedIn User Agreement, Privacy Policy, and Cookie Policy.";

//Compare the extracted text with the expected value present in aggtext

if(agg.getText().equals(aggtext))

System.out.println("Agreement text is displayed while registration");

else

System.out.println("Agreement text is not displayed while registration");

}

@After

public void closureActivities(){

driver.quit();

}

}

LinkedinTestLinks

package MyPackage;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import java.util.Properties;

import java.util.concurrent.TimeUnit;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.NoSuchElementException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

//This will test all objects of type links present on the base URL

public class LinkedinTestLinks {

private WebDriver driver;

private String baseUrl;

@Before

public void setUp() throws FileNotFoundException, IOException{

System.setProperty("webdriver.gecko.driver", "C://Users//madhumita//Downloads//geckodriver//geckodriver.exe");

driver = new FirefoxDriver();

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

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

baseUrl = "https://www.linkedin.com";

}

/\*

\* Objects: Test all links on the home page:

\* Script written by:Madhumita

\* Date:11/08/2016

\*/

@Test

public void testlinks() throws InterruptedException{

driver.get(baseUrl);

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

//store the links in List

ArrayList<String> linkNames = new ArrayList<String>();

//add the links to Array List

for(WebElement linkElement : links){

if(linkElement.getText().length() > 0)

linkNames.add(linkElement.getText());

}

int i=0;

//Click on all links and navigate back to the home page

for(String linkName : linkNames){

i=i+1;

try{

JavascriptExecutor js = (JavascriptExecutor) driver;

WebElement currentElement = driver.findElement(By.linkText(linkName));

((JavascriptExecutor) driver).executeScript("arguments[0].click();", currentElement);//Click on the link

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

System.out.println(i+":Title of page: "+linkName);

driver.navigate().back();//Navigate back to base URL

}

catch(NoSuchElementException e){

//e.printStackTrace();

System.out.println(linkName + " generated exception:"+e);

}

}

}

@After

public void closureActivities(){

driver.quit();

}

}

LinkedinTestLogins

package MyPackage;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.List;

import java.util.Properties;

import java.util.concurrent.TimeUnit;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class LinkedinTestLogins {

private WebDriver driver;

private String baseUrl;

private Properties testData,regdata;

@Before

public void setUp() throws FileNotFoundException, IOException{

testData = new Properties();

testData.load(new FileInputStream("testdata/credentials.properties"));

System.setProperty("webdriver.gecko.driver", "C://Users//madhumita//Downloads//geckodriver//geckodriver.exe");

driver = new FirefoxDriver();

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

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

baseUrl = "https://www.linkedin.com";

}

/\*

\* testLogins:

\* Try to login with the emails and passwords from thecredentials.properties file i.e data driven approach

\*

\* For all valid credentials the signout button was clicked and then the control was navigated to the previous page

\* for invalid credentails the url was compared and the details were printed on the console

\* Script written by:Madhumita

\* Date:11/08/2016

\* \*/

@Test

public void testLogins() throws InterruptedException{

int noOfTests = Integer.parseInt(testData.getProperty("noOfTests"));

while(noOfTests > 0){

driver.get(baseUrl);

driver.findElement(By.id("loginemail")).sendKeys(testData.getProperty("un"+noOfTests));

driver.findElement(By.id("loginpassword")).sendKeys(testData.getProperty("pwd"+noOfTests));

try {

WebElement sub=driver.findElement(By.id("login-submit"));

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].click();", sub);

Thread.sleep(1000);

String loginfailed="https://www.linkedin.com/uas/login-submit";

String currenturl= driver.getCurrentUrl();

System.out.println("Title : "+currenturl);

if (loginfailed.equals(currenturl))//Compare url of the error page

System.out.println("Invalid Credentails : "+noOfTests+"Username="+testData.getProperty("un"+noOfTests)+" Password="+testData.getProperty("pwd"+noOfTests));

else{

System.out.println("Valid Credentials : "+noOfTests+" Username="+testData.getProperty("un"+noOfTests)+" Password="+testData.getProperty("pwd"+noOfTests));

//for valid credentials fing signout and click on it before navigating back

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

int count=0;

for(WebElement link : links){

String check="account-settings";

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

if (count==0){

if (link.getAttribute("class").contains(check))

System.out.println("Setting"+link.getText());

if(link.getAttribute("class").equals("account-submenu-split-link"))

{

//JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].click();", link);

driver.manage().timeouts().implicitlyWait(50, TimeUnit.SECONDS);

driver.navigate().back();

driver.navigate().refresh();

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

count=count+1;

}

}

}

}

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

System.out.println("Error"+e1);

}

driver.navigate().back();

noOfTests--;

}

}

@After

public void closureActivities(){

driver.quit();

}

}

LinkedinTestRegistration

package MyPackage;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Properties;

import java.util.concurrent.TimeUnit;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class LinkedinTestRegistration {

private WebDriver driver;

private String baseUrl;

private Properties testData,regdata;

@Before

public void setUp() throws FileNotFoundException, IOException{

testData = new Properties();

testData.load(new FileInputStream("testdata/credentials.properties"));

regdata = new Properties();

regdata.load(new FileInputStream("testdata/regdata.properties"));

System.setProperty("webdriver.gecko.driver", "C://Users//madhumita//Downloads//geckodriver//geckodriver.exe");

driver = new FirefoxDriver();

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

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

baseUrl = "https://www.linkedin.com";

}

/\*

\* testRegistration:

\* tests the valid inputs for which user can register successfully.

\* Using the data driven approach the input data is read from the regdata.properties file

\* For all invalid credentials the error displayed on the screen is captured and string manipulation is done to ensure correct

\* error was displayed

\* Script written by:Madhumita

\* Date:11/07/2016

\* \*/

public void testRegistration() throws InterruptedException{

int noOfTests = Integer.parseInt(regdata.getProperty("noOfRegTests"));//set the number of reg test cases

while(noOfTests > 0){//programmatic approach

driver.get(baseUrl);

driver.findElement(By.id("reg-firstname")).sendKeys(regdata.getProperty("fn"+noOfTests));//enter first name

driver.findElement(By.id("reg-lastname")).sendKeys(regdata.getProperty("ln"+noOfTests));//enter last name

driver.findElement(By.id("reg-email")).sendKeys(regdata.getProperty("e"+noOfTests));//enter email

driver.findElement(By.id("reg-password")).sendKeys(regdata.getProperty("p"+noOfTests));//enter password

try {

driver.findElement(By.id("registration-submit")).click();

driver.manage().timeouts().implicitlyWait(50, TimeUnit.MILLISECONDS);

String msg=driver.findElement(By.className("alert-content")).getText();

String regfailed="Please enter";

Boolean valid=msg.isEmpty();

if (valid)

System.out.println("Valid Registration Test case No:"+noOfTests);

else{

if(msg.contains(regfailed))

{ if (msg.contains("first name"))

System.out.println("First Name is Blank (Test case No):"+noOfTests);

else if(msg.contains("last name"))

System.out.println("Last Name is Blank (Test case No):"+noOfTests);

else if(msg.contains("email"))

System.out.println("Email Id is invalid (Test case No):"+noOfTests);

else if(msg.contains("password"))

System.out.println("Password is invalid (Test case No):"+noOfTests);

}

else if (msg.contains("Password must be 6 characters or more"))

System.out.println("Length of password is less than 6 (Test case No):"+noOfTests);

}

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

System.out.println("Error"+e1);

}

driver.navigate().back();

noOfTests--;

}

}

@After

public void closureActivities(){

driver.quit();

}

}

LinkedinTestSearch

package MyPackage;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Properties;

import java.util.concurrent.TimeUnit;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class LinkedinTestSearch {

private WebDriver driver;

private String baseUrl;

private Properties testData,regdata;

@Before

public void setUp() throws FileNotFoundException, IOException

{

testData = new Properties();

testData.load(new FileInputStream("testdata/credentials.properties"));

regdata = new Properties();

regdata.load(new FileInputStream("testdata/regdata.properties"));

System.setProperty("webdriver.gecko.driver","C://Users//madhumita//Downloads//geckodriver//geckodriver.exe");

driver = new FirefoxDriver();

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

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

baseUrl = "https://www.linkedin.com";

}

/\* Verification of the search functionality using the data driven approach.

\* The search data is present in the credentials.properties file

\* Script written by:Madhumita

\* Date:11/11/2016

\*/

@Test

public void testsearch() throws InterruptedException{

int noOfSearch = Integer.parseInt(testData.getProperty("noOfSTests"));

String check="account-settings";

driver.get(baseUrl);

//find and enter email input data

driver.findElement(By.id("loginemail")).sendKeys(testData.getProperty("us"+1));

//finds and enter password from the file

driver.findElement(By.id("loginpassword")).sendKeys(testData.getProperty("ps"+1));

try {

driver.findElement(By.id("login-submit")).click();

Thread.sleep(1000);

while (noOfSearch>0)

{

WebElement search =driver.findElement(By.id("main-search-box"));

search.clear();//clear search before entering text

search.sendKeys(testData.getProperty("s"+noOfSearch));

WebElement btn =driver.findElement(By.name("search"));

btn.click();

driver.manage().timeouts().implicitlyWait(4, TimeUnit.SECONDS);

WebElement val=driver.findElement(By.id("results\_count"));

String res=val.getText();

String[] parts = res.split("result");//to extract the number of results

System.out.println("Count::"+res+"--"+"\n");

if(parts[0].equals("0"))

System.out.println("Invalid search");

noOfSearch--;

}

} catch (InterruptedException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

System.out.println("Error"+e1);

}

driver.navigate().back();

}

@After

public void closureActivities(){

driver.quit();

}

}

## Data Files:

Credential.properties

noOfTests=4

un3=madhumitaw@yahoo.co.in

pwd3=Platinum@13

un2=pqr@pqrst

pwd2=hello123

un4=m@tess

pwd4=olla123

noOfSTests=3

us1=XXXXX-VALID INPUT

ps1=XXXXXVALID iNPUT

s1=abhishek

s2=Abcdefghijklmnopqrstuvwxyz123456789011121111111111111111222222222222244444444444m3eeeettttttttttttttttttttttttttttrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr

s3=hello

s4=12wee

regdata.properties:

noOfRegTests=6

fn1=aaa

ln1=aaa

e1=pqr@pqrst.com

p1=hel

fn2=aaa

ln2=aaa

e2=pqr@pqrst

p2=hello

fn3=

ln3=aaa

e3=pqr@pqrst

p3=hello

fn4=mad

ln4=hu

e4=pqr@pqrst.com

p4=helloq

fn5=mad

ln5=hu

e5=pqr@pqrst.com

p5=

fn6=fbn

ln6=

e6=pqr@pqrst

p6=hello

noOffindTest=2

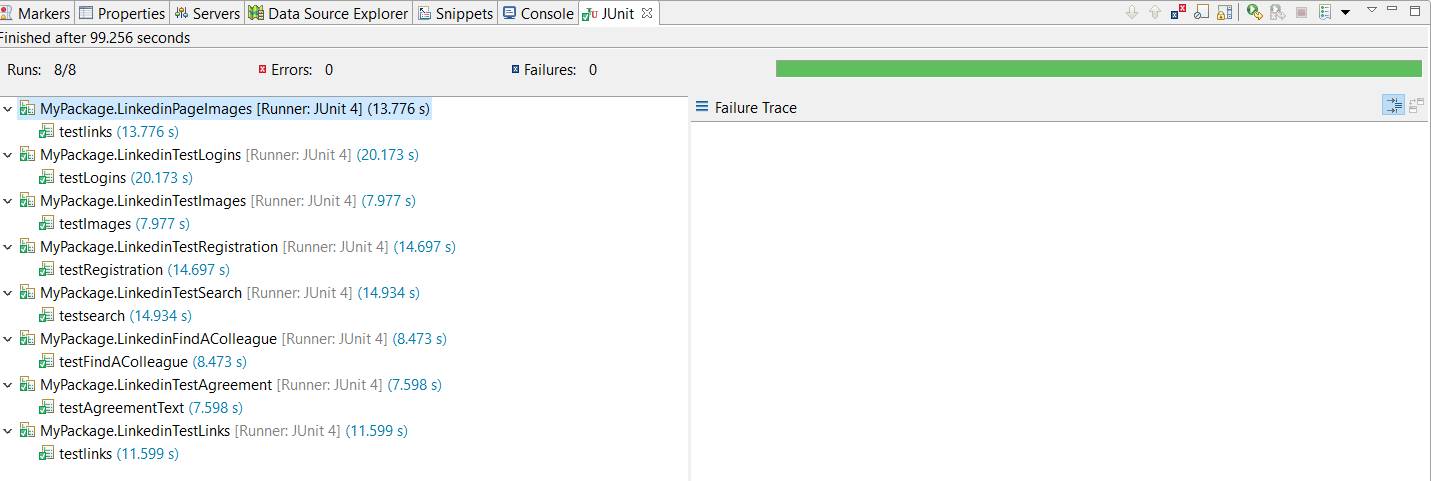
f1=abc

l1=mmm

f2=ma

l2=aa

## Output of entire script:



## Console Output:

1479017111527 geckodriver INFO Listening on 127.0.0.1:18922

Nov 12, 2016 10:05:11 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Attempting bi-dialect session, assuming Postel's Law holds true on the remote end

1479017112127 mozprofile::profile INFO Using profile path C:\Users\MADHUM~1\AppData\Local\Temp\rust\_mozprofile.7O2Rq0QuiD0b

1479017112132 geckodriver::marionette INFO Starting browser C:\Program Files (x86)\Mozilla Firefox\firefox.exe

1479017112147 geckodriver::marionette INFO Connecting to Marionette on localhost:63929

1479017113249 Marionette INFO Listening on port 63929

1479017114772 Marionette INFO startBrowser 54c8533f-1b47-4573-8b15-fd084fb99b3f

Nov 12, 2016 10:05:15 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

1: https://static.licdn.com/sc/h/64xk850n3a8uzse6fi11l3vmz

This image is displayed on Page Load:true

Alt Text is missing

-------

2: https://static.licdn.com/sc/h/5koy91fjbrc47yhwyzws65ml7

This image is displayed on Page Load:false

Alt found as : LinkedIn

-------

[Child 26532] ###!!! ABORT: Aborting on channel error.: file c:/builds/moz2\_slave/m-rel-w32-00000000000000000000/build/src/ipc/glue/MessageChannel.cpp, line 2052

1479017120276 geckodriver INFO Listening on 127.0.0.1:17781

Nov 12, 2016 10:05:20 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Attempting bi-dialect session, assuming Postel's Law holds true on the remote end

1479017120737 mozprofile::profile INFO Using profile path C:\Users\MADHUM~1\AppData\Local\Temp\rust\_mozprofile.kZxTn9VF3QYz

1479017120743 geckodriver::marionette INFO Starting browser C:\Program Files (x86)\Mozilla Firefox\firefox.exe

1479017120760 geckodriver::marionette INFO Connecting to Marionette on localhost:63993

1479017122022 Marionette INFO Listening on port 63993

1479017123671 Marionette INFO startBrowser c533e0c4-7fc9-440c-94c2-532bdaa42c8a

Nov 12, 2016 10:05:24 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Title : https://www.linkedin.com/uas/login-submit

Invalid Credentails : 4 Username=m@tess Password=olla123

Title : https://www.linkedin.com/nhome/

Valid Credentials : 3 Username=madhumitaw@yahoo.co.in Password=Platinum@13

Title : https://www.linkedin.com/uas/login-submit

Invalid Credentails : 2 Username=pqr@pqrst Password=hello123

Title : https://www.linkedin.com/uas/login-submit

Invalid Credentails : 1 Username=null Password=null

[Child 25724] WARNING: pipe error: 232: file c:/builds/moz2\_slave/m-rel-w32-00000000000000000000/build/src/ipc/chromium/src/chrome/common/ipc\_channel\_win.cc, line 513

[Child 25724] ###!!! ABORT: Aborting on channel error.: file c:/builds/moz2\_slave/m-rel-w32-00000000000000000000/build/src/ipc/glue/MessageChannel.cpp, line 2052

1479017141992 geckodriver INFO Listening on 127.0.0.1:34082

Nov 12, 2016 10:05:42 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Attempting bi-dialect session, assuming Postel's Law holds true on the remote end

1479017142461 mozprofile::profile INFO Using profile path C:\Users\MADHUM~1\AppData\Local\Temp\rust\_mozprofile.OqtULse1kd17

1479017142466 geckodriver::marionette INFO Starting browser C:\Program Files (x86)\Mozilla Firefox\firefox.exe

1479017142485 geckodriver::marionette INFO Connecting to Marionette on localhost:64210

1479017143787 Marionette INFO Listening on port 64210

1479017145722 Marionette INFO startBrowser 4385222c-5ca7-4f22-8f8a-dc32c0b03ca1

Nov 12, 2016 10:05:46 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

1: https://static.licdn.com/sc/h/64xk850n3a8uzse6fi11l3vmz

This image is displayed on Page Load:true

Alt Text is missing

-------

2: https://static.licdn.com/sc/h/5koy91fjbrc47yhwyzws65ml7

This image is displayed on Page Load:false

Alt found as : LinkedIn

-------

[Child 8548] ###!!! ABORT: Aborting on channel error.: file c:/builds/moz2\_slave/m-rel-w32-00000000000000000000/build/src/ipc/glue/MessageChannel.cpp, line 2052

1479017151739 geckodriver INFO Listening on 127.0.0.1:41701

Nov 12, 2016 10:05:52 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Attempting bi-dialect session, assuming Postel's Law holds true on the remote end

1479017152081 mozprofile::profile INFO Using profile path C:\Users\MADHUM~1\AppData\Local\Temp\rust\_mozprofile.myomhurb87rk

1479017152086 geckodriver::marionette INFO Starting browser C:\Program Files (x86)\Mozilla Firefox\firefox.exe

1479017152105 geckodriver::marionette INFO Connecting to Marionette on localhost:64298

1479017153628 Marionette INFO Listening on port 64298

1479017155437 Marionette INFO startBrowser 56cf4518-4aaf-4880-9079-cba5f0153925

Nov 12, 2016 10:05:55 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Last Name is Blank (Test case No):6

Password is invalid (Test case No):5

Valid Registration Test case No:4

First Name is Blank (Test case No):3

Email Id is invalid (Test case No):2

Length of password is less than 6 (Test case No):1

[Child 27940] ###!!! ABORT: Aborting on channel error.: file c:/builds/moz2\_slave/m-rel-w32-00000000000000000000/build/src/ipc/glue/MessageChannel.cpp, line 2052

1479017167752 geckodriver INFO Listening on 127.0.0.1:47469

Nov 12, 2016 10:06:08 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Attempting bi-dialect session, assuming Postel's Law holds true on the remote end

1479017168195 mozprofile::profile INFO Using profile path C:\Users\MADHUM~1\AppData\Local\Temp\rust\_mozprofile.UVDnQpl6RM7c

1479017168200 geckodriver::marionette INFO Starting browser C:\Program Files (x86)\Mozilla Firefox\firefox.exe

1479017168218 geckodriver::marionette INFO Connecting to Marionette on localhost:64468

1479017169439 Marionette INFO Listening on port 64468

1479017171169 Marionette INFO startBrowser 7a629146-d8a9-43af-b4be-b94394830e29

Nov 12, 2016 10:06:11 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Count::249,963 results for hello

Save search--

JavaScript warning: https://static.licdn.com/scds/concat/common/js?h=25kaepc6rgo1820ap1rglmzr4-edgsl2z4e4gk56cy2m5kbpp1q-dtx8oyvln9y03x1ku6t0abhc9-cl5mre9823ndhfdrl4nozaofi-6isi7fr80gagap7736arbauct-8ohb0iio22nbqe1w8et54sawe-13kfns70b5ghzzqxzr194d2jo-cr2cf88zdeizqhuobqtot8sge-amjylk8w8039f2lwlov2e4nmc-47qp7uw3i5i1pqeovirlcc070-5pmigtkow46izwzcb8m1b57ly-br7xw7z07pbsy9z5545ze80zu-aikuay313zihm7be1fml6lb8y-7vr4nuab43rzvy2pgq7yvvxjk-9qa4rfxekcw3lt2c06h7p0kmf, line 28: Using //@ to indicate sourceURL pragmas is deprecated. Use //# instead

JavaScript warning: https://static.licdn.com/scds/concat/common/js?h=25kaepc6rgo1820ap1rglmzr4-edgsl2z4e4gk56cy2m5kbpp1q-dtx8oyvln9y03x1ku6t0abhc9-cl5mre9823ndhfdrl4nozaofi-6isi7fr80gagap7736arbauct-8ohb0iio22nbqe1w8et54sawe-13kfns70b5ghzzqxzr194d2jo-cr2cf88zdeizqhuobqtot8sge-amjylk8w8039f2lwlov2e4nmc-47qp7uw3i5i1pqeovirlcc070-5pmigtkow46izwzcb8m1b57ly-br7xw7z07pbsy9z5545ze80zu-aikuay313zihm7be1fml6lb8y-7vr4nuab43rzvy2pgq7yvvxjk-9qa4rfxekcw3lt2c06h7p0kmf, line 28: Using //@ to indicate sourceURL pragmas is deprecated. Use //# instead

Count::249,963 results for hello

Save search--

Count::0 results for Abcdefghijklmnopqrstuvwxyz123456789011121111111111111111222222222222244444444444m3eeeettttttttttttttttttttttttttttrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr--

[Child 32020] ###!!! ABORT: Aborting on channel error.: file c:/builds/moz2\_slave/m-rel-w32-00000000000000000000/build/src/ipc/glue/MessageChannel.cpp, line 2052

1479017184569 geckodriver INFO Listening on 127.0.0.1:47098

Nov 12, 2016 10:06:25 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Attempting bi-dialect session, assuming Postel's Law holds true on the remote end

1479017185119 mozprofile::profile INFO Using profile path C:\Users\MADHUM~1\AppData\Local\Temp\rust\_mozprofile.wgCBxoBGWJ8A

1479017185125 geckodriver::marionette INFO Starting browser C:\Program Files (x86)\Mozilla Firefox\firefox.exe

1479017185215 geckodriver::marionette INFO Connecting to Marionette on localhost:64588

1479017186752 Marionette INFO Listening on port 64588

1479017188531 Marionette INFO startBrowser b240a7e0-e7cd-433d-9032-81421d0a2514

Nov 12, 2016 10:06:28 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Button enabled::false

Button enabled::true

Button enabled::false

Button enabled::true

[Child 4772] ###!!! ABORT: Aborting on channel error.: file c:/builds/moz2\_slave/m-rel-w32-00000000000000000000/build/src/ipc/glue/MessageChannel.cpp, line 2052

1479017194725 geckodriver INFO Listening on 127.0.0.1:11357

Nov 12, 2016 10:06:35 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Attempting bi-dialect session, assuming Postel's Law holds true on the remote end

1479017195147 mozprofile::profile INFO Using profile path C:\Users\MADHUM~1\AppData\Local\Temp\rust\_mozprofile.1PnLL0weffVI

1479017195152 geckodriver::marionette INFO Starting browser C:\Program Files (x86)\Mozilla Firefox\firefox.exe

1479017195170 geckodriver::marionette INFO Connecting to Marionette on localhost:64679

1479017196355 Marionette INFO Listening on port 64679

1479017198140 Marionette INFO startBrowser da4a87d6-1f48-444d-9ed2-962776bf068b

Nov 12, 2016 10:06:38 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Agreement text is displayed while registration

[Child 10796] ###!!! ABORT: Aborting on channel error.: file c:/builds/moz2\_slave/m-rel-w32-00000000000000000000/build/src/ipc/glue/MessageChannel.cpp, line 2052

1479017203771 geckodriver INFO Listening on 127.0.0.1:10671

Nov 12, 2016 10:06:44 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Attempting bi-dialect session, assuming Postel's Law holds true on the remote end

1479017204199 mozprofile::profile INFO Using profile path C:\Users\MADHUM~1\AppData\Local\Temp\rust\_mozprofile.cBr1mGCrNsTk

1479017204204 geckodriver::marionette INFO Starting browser C:\Program Files (x86)\Mozilla Firefox\firefox.exe

1479017204225 geckodriver::marionette INFO Connecting to Marionette on localhost:64734

1479017205464 Marionette INFO Listening on port 64734

1479017207209 Marionette INFO startBrowser 4dedc367-1e85-41b2-8c8c-9d5764fd9f2c

Nov 12, 2016 10:06:47 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

1:Title of page: Forgot password?

2:Title of page: User Agreement

3:Title of page: Privacy Policy

4:Title of page: Cookie Policy

5:Title of page: A

6:Title of page: B

7:Title of page: C

8:Title of page: D

9:Title of page: E

10:Title of page: F

11:Title of page: G

12:Title of page: H

13:Title of page: I

14:Title of page: J

15:Title of page: K

16:Title of page: L

17:Title of page: M

18:Title of page: N

19:Title of page: O

20:Title of page: P

21:Title of page: Q

22:Title of page: R

23:Title of page: S

24:Title of page: T

25:Title of page: U

26:Title of page: V

27:Title of page: W

28:Title of page: X

29:Title of page: Y

30:Title of page: Z

31:Title of page: More

32:Title of page: Browse by country

33:Title of page: Sign Up

34:Title of page: Help Center

35:Title of page: About

36:Title of page: Press

37:Title of page: Blog

38:Title of page: Careers

39:Title of page: Developers

40:Title of page: Talent

41:Title of page: Marketing

42:Title of page: Sales

43:Title of page: Influencers

44:Title of page: Jobs

45:Title of page: Mobile

46:Title of page: ProFinder

47:Title of page: Members

48:Title of page: Jobs

49:Title of page: Pulse

50:Title of page: Topics

51:Title of page: Companies

52:Title of page: Groups

53:Title of page: Universities

54:Title of page: Titles

55:Title of page: User Agreement

56:Title of page: Privacy Policy

57:Title of page: Community Guidelines

58:Title of page: Cookie Policy

59:Title of page: Copyright Policy

60:Title of page: Guest Controls

[Child 24572] ###!!! ABORT: Aborting on channel error.: file c:/builds/moz2\_slave/m-rel-w32-00000000000000000000/build/src/ipc/glue/MessageChannel.cpp, line 2052