Selenium automation

<http://demo.guru99.com/test/newtours/index.php>

Website automation 1:

**package** selenium;

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

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

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

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\lenovo\\eclipse-workspace\\Learning\\chromexe\\chromedriver.exe");

*driver* = **new** ChromeDriver();

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

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

*driver*.findElement(By.*linkText*("REGISTER")).click();

//driver.findElement(By.name("firstName")).click();

*driver*.findElement(By.*name*("firstName")).sendKeys("Neethi");

*driver*.findElement(By.*name*("lastName")).sendKeys("D");

*driver*.findElement(By.*name*("phone")).sendKeys("9566889932");

*driver*.findElement(By.*id*("userName")).sendKeys("bkayrtu@we.com");

*driver*.findElement(By.*name*("address1")).sendKeys("test street");

*driver*.findElement(By.*name*("city")).sendKeys("Chennai");

*driver*.findElement(By.*name*("state")).sendKeys("TamilNadu");

*driver*.findElement(By.*name*("country")).sendKeys("INDIA");

*driver*.findElement(By.*id*("email")).sendKeys("sevenst");

*driver*.findElement(By.*name*("password")).sendKeys("test@1234");

*driver*.findElement(By.*name*("confirmPassword")).sendKeys("test@1234");

*driver*.findElement(By.*name*("submit")).click();

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

My first automation script:

<http://demo.guru99.com/test/newtours/index.php>

**package** selenium;

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

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

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

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

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\lenovo\\eclipse-workspace\\Learning\\chromexe\\chromedriver.exe");

*driver* = **new** ChromeDriver();

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

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

*driver*.findElement(By.*linkText*("Flights")).click();

*driver*.findElement(By.*name*("tripType")).click();

*driver*.findElement(By.*name*("passCount")).sendKeys("2");

*driver*.findElement(By.*name*("fromPort")).sendKeys("Frankfurt");

*driver*.findElement(By.*name*("fromMonth")).sendKeys("November");

*driver*.findElement(By.*name*("fromDay")).sendKeys("31");

*driver*.findElement(By.*name*("toPort")).sendKeys("London");

*driver*.findElement(By.*name*("toMonth")).sendKeys("December");

*driver*.findElement(By.*name*("toDay")).sendKeys("29");

WebElement radio1 = *driver*.findElement(By.*xpath*("//input[@name='servClass' and @value='Business']"));

radio1.click();

*driver*.findElement(By.*name*("airline")).sendKeys("Unified Airlines");

*driver*.findElement(By.*name*("findFlights")).click();

*driver*.findElement(By.*cssSelector*("a[href='index.php']")).click();

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

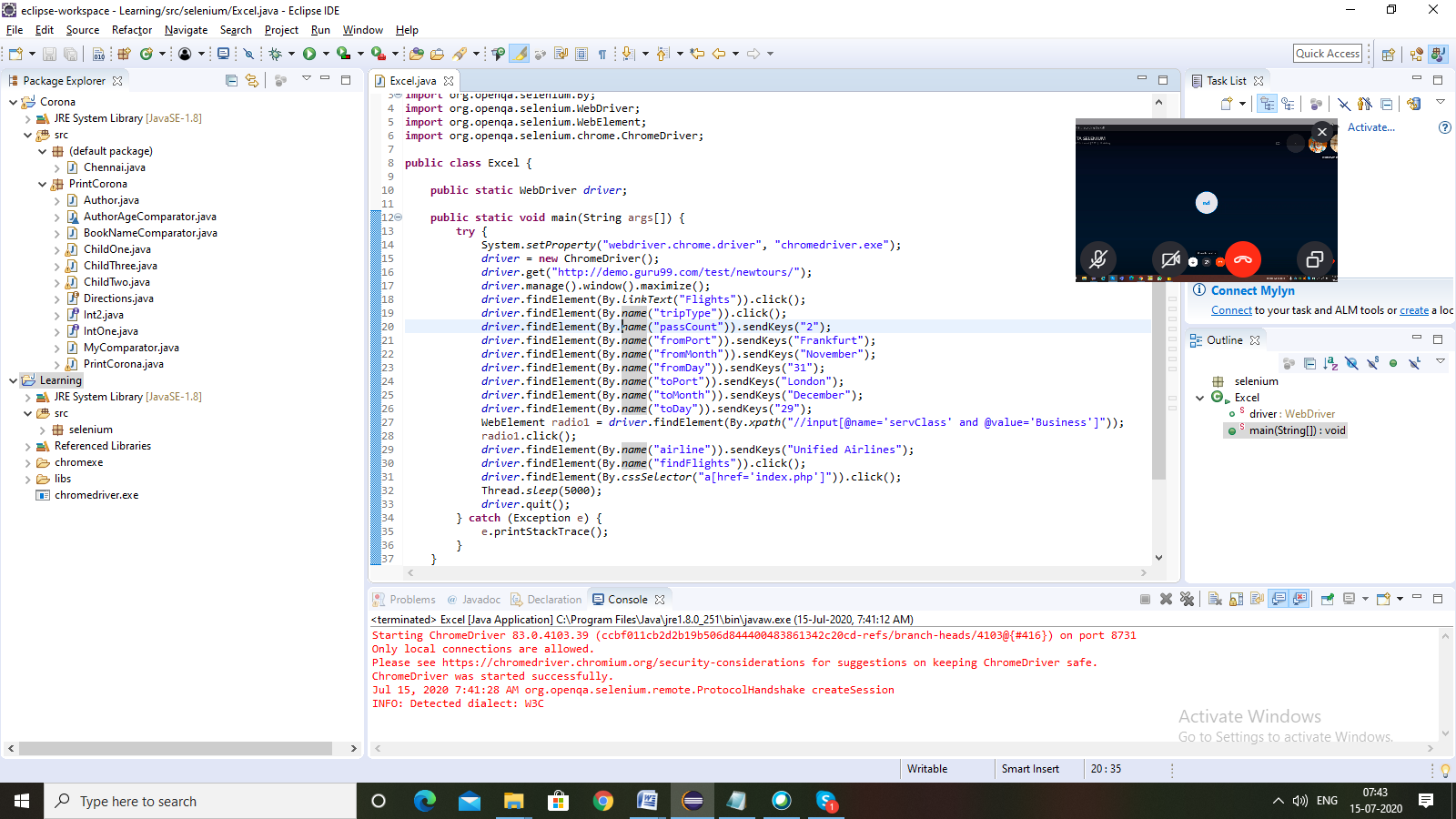
e.printStackTrace();

}

}

}

Chromedriver.exe should be pasted within the project



**package** selenium;

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

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

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

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

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

*driver* = **new** ChromeDriver();

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

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

*driver*.findElement(By.*linkText*("Flights")).click();

*driver*.findElement(By.*name*("tripType")).click();

*driver*.findElement(By.*name*("passCount")).sendKeys("2");

*driver*.findElement(By.*name*("fromPort")).sendKeys("Frankfurt");

*driver*.findElement(By.*name*("fromMonth")).sendKeys("November");

*driver*.findElement(By.*name*("fromDay")).sendKeys("31");

*driver*.findElement(By.*name*("toPort")).sendKeys("London");

*driver*.findElement(By.*name*("toMonth")).sendKeys("December");

*driver*.findElement(By.*name*("toDay")).sendKeys("29");

WebElement radio1 = *driver*.findElement(By.*xpath*("//input[@name='servClass' and @value='Business']"));

radio1.click();

*driver*.findElement(By.*name*("airline")).sendKeys("Unified Airlines");

*driver*.findElement(By.*name*("findFlights")).click();

*driver*.findElement(By.*cssSelector*("a[href='index.php']")).click();

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Extracting the user name which is used for registraftion and x path example:

http://demo.guru99.com/test/newtours/

**package** selenium;

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

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

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

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

*driver* = **new** ChromeDriver();

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

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

*driver*.findElement(By.*linkText*("REGISTER")).click();

*driver*.findElement(By.*name*("firstName")).sendKeys("Neethi");

*driver*.findElement(By.*name*("lastName")).sendKeys("D");

*driver*.findElement(By.*name*("phone")).sendKeys("9566889932");

*driver*.findElement(By.*id*("username")).sendKeys("bkayrtu@we.com");

*driver*.findElement(By.*name*("address1")).sendKeys("test street");

*driver*.findElement(By.*name*("city")).sendKeys("Chennai");

*driver*.findElement(By.*name*("state")).sendKeys("TamilNadu");

*driver*.findElement(By.*name*("country")).sendKeys("INDIA");

String uname = "sampleuser";

*driver*.findElement(By.*id*("email")).sendKeys(uname);

*driver*.findElement(By.*name*("password")).sendKeys("test@1234");

*driver*.findElement(By.*name*("confirmPassword")).sendKeys("test@1234");

*driver*.findElement(By.*name*("submit")).click();

**boolean** noteb = *driver*.findElement(By.*xpath*("//a[.=' sign-in ']/following::b[1]")).isDisplayed();

**if**(noteb==**true**) {

System.***out***.println("Label message displayed!");

}

**else** {

System.***out***.println("Label message not displayed!");

}

String actMsg = *driver*.findElement(By.*xpath*("//a[.=' sign-in ']/following::b[1]")).getText();

System.***out***.println(actMsg);

**if**(actMsg.contains(uname)) {

System.***out***.println("User creation passed! The created Username is: "+uname);

}

**else** {

System.***out***.println("User creation failed!");

}

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Label message displayed!

Note: Your user name is sampleuser.

User creation passed! The created Username is: sampleuser

Negative validation:

**package** selenium;

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

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

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

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

*driver* = **new** ChromeDriver();

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

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

*driver*.findElement(By.*linkText*("REGISTER")).click();

*driver*.findElement(By.*name*("firstName")).sendKeys("Neethi");

*driver*.findElement(By.*name*("lastName")).sendKeys("D");

*driver*.findElement(By.*name*("phone")).sendKeys("9566889932");

*driver*.findElement(By.*id*("username")).sendKeys("bkayrtu@we.com");

*driver*.findElement(By.*name*("address1")).sendKeys("test street");

*driver*.findElement(By.*name*("city")).sendKeys("Chennai");

*driver*.findElement(By.*name*("state")).sendKeys("TamilNadu");

*driver*.findElement(By.*name*("country")).sendKeys("INDIA");

String uname = "sampleuser";

*driver*.findElement(By.*id*("email")).sendKeys(uname);

*driver*.findElement(By.*name*("password")).sendKeys("test@1234");

*driver*.findElement(By.*name*("confirmPassword")).sendKeys("test@1234");

*driver*.findElement(By.*name*("submit")).click();

**boolean** noteb = *driver*.findElement(By.*xpath*("//a[.=' sign-in ']/following::b[1]")).isDisplayed();

**if**(noteb==**true**) {

System.***out***.println("Label message displayed!");

}

**else** {

System.***out***.println("Label message not displayed!");

}

String actMsg = *driver*.findElement(By.*xpath*("//a[.=' sign-in ']/following::b[1]")).getText();

System.***out***.println(actMsg);

**if**(actMsg.contains("Neethi")) {

System.***out***.println("User creation passed! The created Username is: "+uname);

}

**else** {

System.***out***.println("User creation failed!");

}

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 5128

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 15, 2020 11:00:30 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Label message displayed!

Note: Your user name is sampleuser.

User creation failed!

Absolute Xpath:

/html/body/div[2]/table/tbody/tr/td[2]/table/tbody/tr[4]/td/table/tbody/tr/td[2]/table/tbody/tr[2]/td[3]/form/table/tbody/tr[4]/td/table/tbody/tr[4]/td[2]/div/input

Disadvantage:

If an element from this html header or body is removed or included the system will not be able to find the element.

Relative Xpath:

//input[@name='submit']

//input[@name='submit' and @type='submit']

//\*[@name='submit' and @type='submit']

//a[.='REGISTER']

//input[@name='firstName']

//font[.='Preferences']/following::input[1]

//font[.='Preferences']/following::select[1]

//font[.='Preferences']/preceding::input[1]

//font[.='Preferences' and @size='2']

//a[.=' sign-in ']/following::b[1]

//label[.=' Cricket ']/preceding::input[1]

Automation registration:

url: <http://demo.automationtesting.in/Register.html>

Code:

**package** selenium;

**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.openqa.selenium.interactions.Actions;

**import** org.openqa.selenium.support.ui.Select;

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

*driver* = **new** ChromeDriver();

*driver*.get("http://demo.automationtesting.in/Register.html");

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

*driver*.findElement(By.*xpath*("//label[.='Full Name\* ']/following::input[1]")).sendKeys("Neethi");

*driver*.findElement(By.*xpath*("//label[.='Full Name\* ']/following::input[2]")).sendKeys("Durai");

*driver*.findElement(By.*xpath*("//label[.='Address']/following::textarea[1]")).sendKeys("alwarthirunagar annexe,chennai - 600087");

*driver*.findElement(By.*xpath*("//label[.='Email address\*']/following::input[1]")).sendKeys("neei138@gmail.com");

*driver*.findElement(By.*xpath*("//input[@type='tel']")).sendKeys("8056035186");

//driver.findElement(By.xpath("//input[@name='radiooptions' and @value='Male']")).click();

*driver*.findElement(By.*xpath*("//input[@name='radiooptions' and @value='FeMale']")).click();

*driver*.findElement(By.*xpath*("//label[.=' Cricket ']/preceding::input[1]")).click();

*driver*.findElement(By.*xpath*("//label[.='Movies ']/preceding::input[1]")).click();

*driver*.findElement(By.*xpath*("//label[.='Hockey']/preceding::input[1]")).click();

WebElement lang = *driver*.findElement(By.*xpath*("//div[@id='msdd']"));

lang.click();

*driver*.findElement(By.*xpath*("//a[.='Danish']")).click();

*driver*.findElement(By.*xpath*("//a[.='Bulgarian']")).click();

*driver*.findElement(By.*xpath*("//a[.='English']")).click();

Select skill = **new** Select(*driver*.findElement(By.*id*("Skills")));

skill.selectByValue("Android");

Select country = **new** Select(*driver*.findElement(By.*id*("countries")));

country.selectByValue("India");

WebElement countrysel = *driver*.findElement(By.*xpath*("//span[@role='combobox']"));

countrysel.click();

WebElement textbox = *driver*.findElement(By.*xpath*("//input[@type='search']"));

textbox.click();

textbox.sendKeys("Ind");

Thread.*sleep*(500);

textbox.sendKeys(Keys.***ENTER***);

*driver*.findElement(By.*id*("yearbox")).sendKeys("1991");

*driver*.findElement(By.*xpath*("//label[.='Date Of Birth']/following::select[2]")).sendKeys("August");

*driver*.findElement(By.*xpath*("//label[.='Date Of Birth']/following::select[3]")).sendKeys("13");

*driver*.findElement(By.*id*("firstpassword")).sendKeys("Neets@13");

*driver*.findElement(By.*id*("secondpassword")).sendKeys("Neets@13");

*driver*.findElement(By.*id*("submitbtn")).click();

Thread.*sleep*(15000);

**boolean** web = *driver*.findElement(By.*xpath*("//span[.='Email']")).isDisplayed();

System.***out***.println(web);

**if**(web==**true**)

{

System.***out***.println("Registration is successful!!");

}

**else**

{

System.***out***.println("Registration Failed");

}

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 39499

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 16, 2020 3:08:40 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

true

Registration is successful!!

Negative scenario:

package selenium;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.NoSuchElementException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

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

public class Excel {

public static WebDriver driver;

public static void main(String args[]) {

try {

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

driver = new ChromeDriver();

driver.get("http://demo.automationtesting.in/Register.html");

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

//driver.findElement(By.xpath("//label[.='Full Name\* ']/following::input[1]")).sendKeys("Neethi");

//driver.findElement(By.xpath("//label[.='Full Name\* ']/following::input[2]")).sendKeys("Durai");

driver.findElement(By.xpath("//label[.='Address']/following::textarea[1]")).sendKeys("alwarthirunagar annexe,chennai - 600087");

driver.findElement(By.xpath("//label[.='Email address\*']/following::input[1]")).sendKeys("neei138@gmail.com");

driver.findElement(By.xpath("//input[@type='tel']")).sendKeys("8056035186");

//driver.findElement(By.xpath("//input[@name='radiooptions' and @value='Male']")).click();

driver.findElement(By.xpath("//input[@name='radiooptions' and @value='FeMale']")).click();

driver.findElement(By.xpath("//label[.=' Cricket ']/preceding::input[1]")).click();

driver.findElement(By.xpath("//label[.='Movies ']/preceding::input[1]")).click();

driver.findElement(By.xpath("//label[.='Hockey']/preceding::input[1]")).click();

WebElement lang = driver.findElement(By.xpath("//div[@id='msdd']"));

lang.click();

driver.findElement(By.xpath("//a[.='Danish']")).click();

driver.findElement(By.xpath("//a[.='Bulgarian']")).click();

driver.findElement(By.xpath("//a[.='English']")).click();

Select skill = new Select(driver.findElement(By.id("Skills")));

skill.selectByValue("Android");

Select country = new Select(driver.findElement(By.id("countries")));

country.selectByValue("India");

WebElement countrysel = driver.findElement(By.xpath("//span[@role='combobox']"));

countrysel.click();

WebElement textbox = driver.findElement(By.xpath("//input[@type='search']"));

textbox.click();

textbox.sendKeys("Ind");

Thread.sleep(500);

textbox.sendKeys(Keys.ENTER);

driver.findElement(By.id("yearbox")).sendKeys("1991");

driver.findElement(By.xpath("//label[.='Date Of Birth']/following::select[2]")).sendKeys("August");

driver.findElement(By.xpath("//label[.='Date Of Birth']/following::select[3]")).sendKeys("13");

driver.findElement(By.id("firstpassword")).sendKeys("Neets@13");

driver.findElement(By.id("secondpassword")).sendKeys("Neets@13");

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

Thread.sleep(15000);

try {

boolean web = driver.findElement(By.xpath("//span[.='Email']")).isDisplayed();

System.out.println(web);

if(web==true)

{

System.out.println("Registration is successful!!");

}

} catch (NoSuchElementException ne) {

System.out.println("Registration Failed");

}

Thread.sleep(5000);

//driver.quit();

} catch (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 48199

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 17, 2020 9:11:17 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Email element not displayed

Chrome Functions:

Page name

url

source code

**package** selenium;

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

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

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

*driver* = **new** ChromeDriver();

*driver*.get("http://demo.automationtesting.in/Register.html");

String title = *driver*.getTitle();

System.***out***.println("Title: " +title);

String url = *driver*.getCurrentUrl();

System.***out***.println("App URL: " +url);

//String source = driver.getPageSource();

//System.out.println(source);

Thread.*sleep*(5000);

*driver*.quit();

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 46641

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 16, 2020 3:57:35 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Title: Register

App URL: http://demo.automationtesting.in/Register.html

To load maximised window:

To disable chrome is being controlled by automated test software

options.addArguments(“disable-infobars”);

**package** selenium;

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

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

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

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("http://demo.automationtesting.in/Register.html");

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

}

To print the number of elements from a list:

To select a second value to the field

**package** selenium;

**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.chrome.ChromeOptions;

**import** org.openqa.selenium.support.ui.Select;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

*driver* = **new** ChromeDriver();

*driver*.get("http://demo.automationtesting.in/Register.html");

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

WebElement fnameField = *driver*.findElement(By.*xpath*("//input[@placeholder='First Name' or ng-model='FirstName']"));

fnameField.click();

fnameField.sendKeys("Balakartick");

WebElement skillsel = *driver*.findElement(By.*xpath*("//select[@id='Skills']"));

Select skisel = **new** Select(skillsel);

**int** count = skisel.getOptions().size();

System.***out***.println("Total number of skills: " +count);

**for**(**int** i=1; i<count; i++) {

//skisel.selectByIndex(i);

String skill = skisel.getOptions().get(i).getText();

System.***out***.println("Iteration: " +i);

System.***out***.println("Skill: " +skill);

}

skisel.selectByValue("AutoCAD");

Thread.*sleep*(3000);

skisel.selectByVisibleText("Support");

Thread.*sleep*(5000);

*driver*.quit();

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 32939

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 16, 2020 4:15:58 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Total number of skills: 78

Iteration: 1

Skill: Adobe InDesign

Iteration: 2

Skill: Adobe Photoshop

Iteration: 3

Skill: Analytics

Iteration: 4

Skill: Android

Iteration: 5

Skill: APIs

Iteration: 6

Skill: Art Design

Iteration: 7

Skill: AutoCAD

Iteration: 8

Skill: Backup Management

Iteration: 9

Skill: C

Iteration: 10

Skill: C++

Iteration: 11

Skill: Certifications

Iteration: 12

Skill: Client Server

Iteration: 13

Skill: Client Support

Iteration: 14

Skill: Configuration

Iteration: 15

Skill: Content Managment

Iteration: 16

Skill: Content Management Systems (CMS)

Iteration: 17

Skill: Corel Draw

Iteration: 18

Skill: Corel Word Perfect

Iteration: 19

Skill: CSS

Iteration: 20

Skill: Data Analytics

Iteration: 21

Skill: Desktop Publishing

Iteration: 22

Skill: Design

Iteration: 23

Skill: Diagnostics

Iteration: 24

Skill: Documentation

Iteration: 25

Skill: End User Support

Iteration: 26

Skill: Email

Iteration: 27

Skill: Engineering

Iteration: 28

Skill: Excel

Iteration: 29

Skill: FileMaker Pro

Iteration: 30

Skill: Fortran

Iteration: 31

Skill: HTML

Iteration: 32

Skill: Implementation

Iteration: 33

Skill: Installation

Iteration: 34

Skill: Internet

Iteration: 35

Skill: iOS

Iteration: 36

Skill: iPhone

Iteration: 37

Skill: Linux

Iteration: 38

Skill: Java

Iteration: 39

Skill: Javascript

Iteration: 40

Skill: Mac

Iteration: 41

Skill: Matlab

Iteration: 42

Skill: Maya

Iteration: 43

Skill: Microsoft Excel

Iteration: 44

Skill: Microsoft Office

Iteration: 45

Skill: Microsoft Outlook

Iteration: 46

Skill: Microsoft Publisher

Iteration: 47

Skill: Microsoft Word

Iteration: 48

Skill: Microsoft Visual

Iteration: 49

Skill: Mobile

Iteration: 50

Skill: MySQL

Iteration: 51

Skill: Networks

Iteration: 52

Skill: Open Source Software

Iteration: 53

Skill: Oracle

Iteration: 54

Skill: Perl

Iteration: 55

Skill: PHP

Iteration: 56

Skill: Presentations

Iteration: 57

Skill: Processing

Iteration: 58

Skill: Programming

Iteration: 59

Skill: PT Modeler

Iteration: 60

Skill: Python

Iteration: 61

Skill: QuickBooks

Iteration: 62

Skill: Ruby

Iteration: 63

Skill: Shade

Iteration: 64

Skill: Software

Iteration: 65

Skill: Spreadsheet

Iteration: 66

Skill: SQL

Iteration: 67

Skill: Support

Iteration: 68

Skill: Systems Administration

Iteration: 69

Skill: Tech Support

Iteration: 70

Skill: Troubleshooting

Iteration: 71

Skill: Unix

Iteration: 72

Skill: UI / UX

Iteration: 73

Skill: Web Page Design

Iteration: 74

Skill: Windows

Iteration: 75

Skill: Word Processing

Iteration: 76

Skill: XML

Iteration: 77

Skill: XHTML

Registration deselect multiselect

package selenium;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.NoSuchElementException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

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

public class Excel {

public static WebDriver driver;

public static void main(String args[]) {

try {

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

driver = new ChromeDriver();

driver.get("http://demo.automationtesting.in/Register.html");

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

driver.findElement(By.xpath("//label[.='Full Name\* ']/following::input[1]")).sendKeys("Priya");

driver.findElement(By.xpath("//label[.='Full Name\* ']/following::input[1]")).clear();

driver.findElement(By.xpath("//label[.='Full Name\* ']/following::input[1]")).sendKeys("Neethi");

driver.findElement(By.xpath("//label[.='Full Name\* ']/following::input[2]")).sendKeys("Durai");

driver.findElement(By.xpath("//label[.='Address']/following::textarea[1]")).sendKeys("alwarthirunagar annexe,chennai - 600087");

driver.findElement(By.xpath("//label[.='Email address\*']/following::input[1]")).sendKeys("neei138@gmail.com");

driver.findElement(By.xpath("//input[@type='tel']")).sendKeys("8056035186");

driver.findElement(By.xpath("//input[@name='radiooptions' and @value='Male']")).click();

Thread.sleep(2000);

driver.findElement(By.xpath("//input[@name='radiooptions' and @value='FeMale']")).click();

driver.findElement(By.xpath("//label[.=' Cricket ']/preceding::input[1]")).click();

driver.findElement(By.xpath("//label[.='Movies ']/preceding::input[1]")).click();

driver.findElement(By.xpath("//label[.='Hockey']/preceding::input[1]")).click();

Thread.sleep(2000);

driver.findElement(By.xpath("//label[.='Movies ']/preceding::input[1]")).click();

WebElement lang = driver.findElement(By.xpath("//div[@id='msdd']"));

lang.click();

driver.findElement(By.xpath("//a[.='Danish']")).click();

driver.findElement(By.xpath("//a[.='Bulgarian']")).click();

lang.click();

Thread.sleep(3000);

driver.findElement(By.xpath("//div[.='Danish']/span[1]")).click();

driver.findElement(By.xpath("//a[.='English']")).click();

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

Select skill = new Select(driver.findElement(By.id("Skills")));

skill.selectByValue("Android");

Select country = new Select(driver.findElement(By.id("countries")));

country.selectByValue("Australia");

Thread.sleep(2000);

country.selectByIndex(0);

Thread.sleep(3000);

country.selectByValue("India");

WebElement countrysel = driver.findElement(By.xpath("//span[@role='combobox']"));

countrysel.click();

WebElement textbox = driver.findElement(By.xpath("//input[@type='search']"));

textbox.click();

textbox.sendKeys("Den");

Thread.sleep(500);

textbox.sendKeys(Keys.ENTER);

Thread.sleep(2000);

driver.findElement(By.xpath("//span[@role='combobox']")).click();

Thread.sleep(500);

driver.findElement(By.xpath("//ul[@role='tree']/li[1]")).click();

Thread.sleep(2000);

countrysel.click();

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

driver.findElement(By.xpath("//input[@type='search']")).sendKeys("Ind");

Thread.sleep(500);

driver.findElement(By.xpath("//input[@type='search']")).sendKeys(Keys.ENTER);

driver.findElement(By.id("yearbox")).sendKeys("1991");

driver.findElement(By.xpath("//label[.='Date Of Birth']/following::select[2]")).sendKeys("August");

driver.findElement(By.xpath("//label[.='Date Of Birth']/following::select[3]")).sendKeys("13");

driver.findElement(By.id("firstpassword")).sendKeys("Neets@13");

driver.findElement(By.id("secondpassword")).sendKeys("Neets@13");

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

Thread.sleep(15000);

try {

boolean web = driver.findElement(By.xpath("//span[.='Email']")).isDisplayed();

System.out.println(web);

if(web==true)

{

System.out.println("Registration is successful!!");

}

} catch (NoSuchElementException ne) {

System.out.println("Registration Failed");

}

Thread.sleep(5000);

//driver.quit();

} catch (Exception e) {

e.printStackTrace();

}

}

}

Alerts:

package selenium;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.NoSuchElementException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.interactions.Actions;

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

public class Excel {

public static WebDriver driver;

public static void main(String args[]) {

try {

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

ChromeOptions options = new ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", false);

options.setExperimentalOption("excludeSwitches", new String[]{"enable-automation"});

driver = new ChromeDriver(options);

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

driver.findElement(By.xpath("//button[.='Click me' and @id='alertButton']")).click();

Thread.sleep(2000);

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

driver.findElement(By.xpath("//button[.='Click me' and @id='timerAlertButton']")).click();

Thread.sleep(7000);

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

driver.findElement(By.xpath("//button[.='Click me' and @id='confirmButton']")).click();

Thread.sleep(3000);

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

driver.findElement(By.xpath("//button[.='Click me' and @id='confirmButton']")).click();

Thread.sleep(3000);

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

System.out.println(alerttxt);

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

WebElement confirm = driver.findElement(By.id("confirmResult"));

String success = confirm.getText();

if(success.contains("Ok"))

{

System.out.println("Confirm result output positive: "+success);

}

else

{

System.out.println("Confirm result output negative: "+success);

}

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

String name ="Neethi";

driver.switchTo().alert().sendKeys(name);

Thread.sleep(2000);

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

String result = driver.findElement(By.id("promptResult")).getText();

if(result.contains(name))

{

System.out.println("prompt results accept case : "+result);

}

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

driver.switchTo().alert().sendKeys(name);

Thread.sleep(2000);

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

Thread.sleep(2000);

try {

String result2 = driver.findElement(By.id("promptResult")).getText();

if(result2.contains(name))

{

System.out.println("prompt results dismiss case : "+result2);

}

} catch (NoSuchElementException e) {

System.out.println("No such text displayed.");

}

Thread.sleep(5000);

driver.quit();

} catch (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 17377

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 22, 2020 7:23:43 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Do you confirm action?

Confirm result output negative: You selected Cancel

prompt results accept case : You entered Neethi

No such text displayed.

Testng jar : <https://mvnrepository.com/artifact/org.testng/testng/6.9.10>

CTRL+SHIFT+O – delete unwanted import and have only used import.

Frames:

**package** selenium;

**import** java.util.List;

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

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.NoSuchElementException;

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

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

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

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

**import** org.openqa.selenium.interactions.Actions;

**import** org.openqa.selenium.support.ui.Select;

**import** org.testng.Assert;

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("https://www.redbus.in/");

*driver*.findElement(By.*xpath*("//div[@id='sign-in-icon-down']")).click();

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//li[@id='signInLink']")).click();

Thread.*sleep*(1000);

List<WebElement> frameslist = *driver*.findElements(By.*tagName*("iframe"));

**int** count = frameslist.size();

System.***out***.println("Total number of iframes: " +count);

**for**(**int** i=0; i<count; i++) {

System.***out***.println("Iteration: " +i);

*driver*.switchTo().frame(i);

**try** {

WebElement mob = *driver*.findElement(By.*id*("mobileNoInp"));

**boolean** mobinp = mob.isDisplayed();

**if**(mobinp==**true**) {

mob.click();

mob.sendKeys("9566880932");

**break**;

}

} **catch** (NoSuchElementException ne) {

System.***out***.println("Element not found!");

}

}

//driver.switchTo().frame(0);

//*driver*.switchTo().frame(*driver*.findElement(By.*xpath*("//iframe[@class='modalIframe']")));

//driver.findElement(By.id("mobileNoInp")).sendKeys("9566880932");

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 12777

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 25, 2020 8:16:34 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Total number of iframes: 2

Iteration: 0

**Assertions:**

**package** selenium;

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

**import** org.openqa.selenium.NoSuchElementException;

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

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

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

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

**import** org.openqa.selenium.support.ui.Select;

**import** org.testng.Assert;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("https://orangehrm-demo-6x.orangehrmlive.com/");

WebElement ortext = *driver*.findElement(By.*xpath*("//input[@placeholder='Username']/preceding::h3[1]"));

**boolean** ort = ortext.isDisplayed();

String actText = ortext.getText();

//String expText = "OrangeHRM Demo";

String expText = "OrangeHRM Dem";

**try**{

**if**(ort==**true**) {

Assert.*assertEquals*(actText, expText);

System.***out***.println("Orange HRM Login page text displayed!");

}

}

**catch** (AssertionError ne) {

System.***out***.println("Expected text is :\""+expText+"\" but the actual text is :\""+actText+"\"");

}

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}}

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 28732

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 25, 2020 9:48:27 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Expected text is :"OrangeHRM Dem" but the actual text is :"OrangeHRM Demo"

**package** selenium;

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

**import** org.openqa.selenium.NoSuchElementException;

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

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

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

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

**import** org.openqa.selenium.support.ui.Select;

**import** org.testng.Assert;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("https://orangehrm-demo-6x.orangehrmlive.com/");

WebElement ortext = *driver*.findElement(By.*xpath*("//input[@placeholder='Username']/preceding::h3[1]"));

**boolean** ort = ortext.isDisplayed();

String actText = ortext.getText();

String expText = "OrangeHRM Demo";

//String expText = "OrangeHRM Dem";

**try**{

**if**(ort==**true**) {

Assert.*assertEquals*(actText, expText);

System.***out***.println("Orange HRM Login page text displayed!");

}

}

**catch** (AssertionError ne) {

System.***out***.println("Expected text is :\""+expText+"\" but the actual text is :\""+actText+"\"");

}

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 5719

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 25, 2020 10:14:24 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Orange HRM Login page text displayed!

**Actions class Drag and drop:**

**package** selenium;

**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.chrome.ChromeOptions;

**import** org.openqa.selenium.interactions.Actions;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

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

WebElement mainItem = *driver*.findElement(By.*id*("draggable"));

WebElement mainsub = *driver*.findElement(By.*id*("droppable"));

Actions actions = **new** Actions(*driver*);

Thread.*sleep*(2000);

actions.dragAndDropBy(mainItem, 10, 20);

actions.dragAndDropBy(mainsub, 5, 9);

Thread.*sleep*(3000);

actions.dragAndDrop(mainItem, mainsub).build().perform();

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 15351

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 26, 2020 6:17:10 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

WebElement mainItem = driver.findElement(By.id("dragBox"));

Actions actions = new Actions(driver);

Thread.sleep(2000);

actions.dragAndDropBy(mainItem, 45, 23).build().perform();

>

Actions class mouse hover:

**package** selenium;

**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.chrome.ChromeOptions;

**import** org.openqa.selenium.interactions.Actions;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

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

WebElement mainItem = *driver*.findElement(By.*linkText*("Main Item 2"));

WebElement mainsub = *driver*.findElement(By.*xpath*("//a[.='Main Item 2']/following::li[2]"));

Actions actions = **new** Actions(*driver*);

actions.moveToElement(mainItem).build().perform();

Thread.*sleep*(2000);

actions.moveToElement(mainsub).build().perform();

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 33580

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 26, 2020 9:33:20 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Actions class double click and right click:

**package** selenium;

**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.chrome.ChromeOptions;

**import** org.openqa.selenium.interactions.Actions;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

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

WebElement mainItem = *driver*.findElement(By.*id*("doubleClickBtn"));

WebElement mainsub = *driver*.findElement(By.*id*("rightClickBtn"));

Actions actions = **new** Actions(*driver*);

actions.moveToElement(mainItem).build().perform();

actions.doubleClick(mainItem).build().perform();

Thread.*sleep*(2000);

actions.contextClick(mainsub).build().perform();

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 39258

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 26, 2020 10:06:57 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Robot class selenium

Minimise screen after running or the cursor will be in eclipse itself.

**package** selenium;

**import** java.awt.Robot;

**import** java.awt.event.KeyEvent;

**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.chrome.ChromeOptions;

**import** org.openqa.selenium.interactions.Actions;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

Thread.*sleep*(2000);

Robot robot = **new** Robot();

robot.keyPress(KeyEvent.***VK\_WINDOWS***);

robot.keyPress(KeyEvent.***VK\_R***);

robot.keyRelease(KeyEvent.***VK\_R***);

robot.keyRelease(KeyEvent.***VK\_WINDOWS***);

robot.keyPress(KeyEvent.***VK\_N***);

robot.keyRelease(KeyEvent.***VK\_N***);

robot.keyPress(KeyEvent.***VK\_O***);

robot.keyRelease(KeyEvent.***VK\_O***);

robot.keyPress(KeyEvent.***VK\_T***);

robot.keyRelease(KeyEvent.***VK\_T***);

Thread.*sleep*(1000);

robot.keyPress(KeyEvent.***VK\_DOWN***);

robot.keyRelease(KeyEvent.***VK\_DOWN***);

Thread.*sleep*(1000);

robot.keyPress(KeyEvent.***VK\_ENTER***);

robot.keyRelease(KeyEvent.***VK\_ENTER***);

Thread.*sleep*(2000);

} **catch** (Exception e) {

e.printStackTrace();

}

}}

Window handling:

**package** selenium;

**import** java.util.Set;

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

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

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

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

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("http://demo.automationtesting.in/Windows.html");

Thread.*sleep*(2000);

String parentwinhandle = *driver*.getWindowHandle();

System.***out***.println("Parent window handle"+parentwinhandle);

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

Set<String> allwinhandles = *driver*.getWindowHandles();

System.***out***.println(allwinhandles);

**for**(String handle : allwinhandles) {

System.***out***.println(handle);

*driver*.switchTo().window(handle);

}

Thread.*sleep*(5000);

*driver*.findElement(By.*xpath*("//button[@id='COVID19button']")).click();

String chtitle = *driver*.getTitle();

System.***out***.println("Child title: " +chtitle);

Thread.*sleep*(2000);

*driver*.close();

*driver*.switchTo().window(parentwinhandle);

String partitle = *driver*.getTitle();

System.***out***.println("Parent title: " +partitle);

Thread.*sleep*(5000);

*driver*.quit();

}

**catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 42679

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 27, 2020 8:28:45 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Parent window handleCDwindow-EEC257F34A37004B402B3F1CA9854282

[CDwindow-EEC257F34A37004B402B3F1CA9854282, CDwindow-F0334DAAF376F593782F285E82F2AB5D]

CDwindow-EEC257F34A37004B402B3F1CA9854282

CDwindow-F0334DAAF376F593782F285E82F2AB5D

Child title: Open We Stand | Helping Your Business Stay Open - GoDaddy

Parent title: Frames & windows

Mutilple window handling with same window handle:

**package** selenium;

**import** java.util.ArrayList;

**import** java.util.Set;

**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.chrome.ChromeOptions;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("http://demo.automationtesting.in/Windows.html");

System.***out***.println("Parent window"+*driver*.getWindowHandle());

Thread.*sleep*(2000);

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

WebElement newSep = *driver*.findElement(By.*xpath*("//a[.='Open New Seperate Windows']"));

newSep.click();

System.***out***.println(*driver*.getWindowHandle());

Thread.*sleep*(2000);

WebElement clkbtn = *driver*.findElement(By.*xpath*("//button[.='click']"));

**for**(**int** i=1; i<=5; i++) {

clkbtn.click();

Thread.*sleep*(1500);

System.***out***.println(*driver*.getWindowHandle());

}

ArrayList<String> tabs = **new** ArrayList<String>(*driver*.getWindowHandles());

**int** count = tabs.size();

System.***out***.println("Total windows at start: " +count);

*driver*.switchTo().window(tabs.get(1));

System.***out***.println("Closing window handle"+*driver*.getWindowHandle());

Thread.*sleep*(2000);

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

Thread.*sleep*(2000);

*driver*.close();

Set<String> currwinhandles = *driver*.getWindowHandles();

**int** lastsize = currwinhandles.size();

System.***out***.println("Final windows count: " +lastsize);

Thread.*sleep*(5000);

*driver*.quit();

}

**catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 41410

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 27, 2020 8:51:58 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Parent windowCDwindow-76AC0524E1F74415E3F0C6FA1ACCA6AC

CDwindow-76AC0524E1F74415E3F0C6FA1ACCA6AC

CDwindow-76AC0524E1F74415E3F0C6FA1ACCA6AC

CDwindow-76AC0524E1F74415E3F0C6FA1ACCA6AC

CDwindow-76AC0524E1F74415E3F0C6FA1ACCA6AC

CDwindow-76AC0524E1F74415E3F0C6FA1ACCA6AC

CDwindow-76AC0524E1F74415E3F0C6FA1ACCA6AC

Total windows at start: 7

Closing window handleCDwindow-FCE5DE6371A257824ABE3E0FD7CE3D5B

Final windows count: 6

Same button target name handling:

**package** selenium;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Set;

**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.chrome.ChromeOptions;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver*= **new** ChromeDriver(options);

*driver*.get("http://demo.automationtesting.in/Windows.html");

List<WebElement> clk = *driver*.findElements(By.*xpath*("//a[@target='\_blank']"));

**int** count = clk.size();

System.***out***.println("Total number of elements: " +count);

**for**(**int** i=0; i<count; i++) {

System.***out***.println("Iteration: " +i);

String btname = clk.get(i).getAttribute("href");

System.***out***.println("Href: " +btname);

**if**(btname.equalsIgnoreCase("http://www.sakinalium.in/")) {

clk.get(i).click();

**break**;

}

}

String title = *driver*.getTitle();

String url = *driver*.getCurrentUrl();

String parentwinhan = *driver*.getWindowHandle();

System.***out***.println("Title: " +title);

System.***out***.println("URL: " +url);

System.***out***.println("Window handle: " +parentwinhan);

Set<String> allHandles = *driver*.getWindowHandles();

**for**(String handle : allHandles) {

*driver*.switchTo().window(handle);

}

String newtitle = *driver*.getTitle();

System.***out***.println("Current title: " +newtitle);

*driver*.findElement(By.*xpath*("//img[@src='https://mcc.godaddy.com/images/gdlogo.png']")).click();

Thread.*sleep*(5000);

*driver*.close();

*driver*.switchTo().window(parentwinhan);

String title2 = *driver*.getTitle();

String url2 = *driver*.getCurrentUrl();

System.***out***.println("Final title: " +title2);

System.***out***.println("Final url: " +url2);

Thread.*sleep*(2000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 15060

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 27, 2020 9:14:19 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Total number of elements: 8

Iteration: 0

Href: http://www.sakinalium.in/

Title: Frames & windows

URL: http://demo.automationtesting.in/Windows.html

Window handle: CDwindow-BFE0E90EEA240969B175015DD772B0E7

Current title: sakinalium.in

Final title: Frames & windows

Final url: http://demo.automationtesting.in/Windows.html

Implicit wait:

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

waits for each element for the predefined time so time is delayed for each elememt.

Explicit wait:

WebDriverWait wait = **new** WebDriverWait(*driver*, 15);

wait.until(ExpectedConditions.*elementToBeClickable*(newSep));

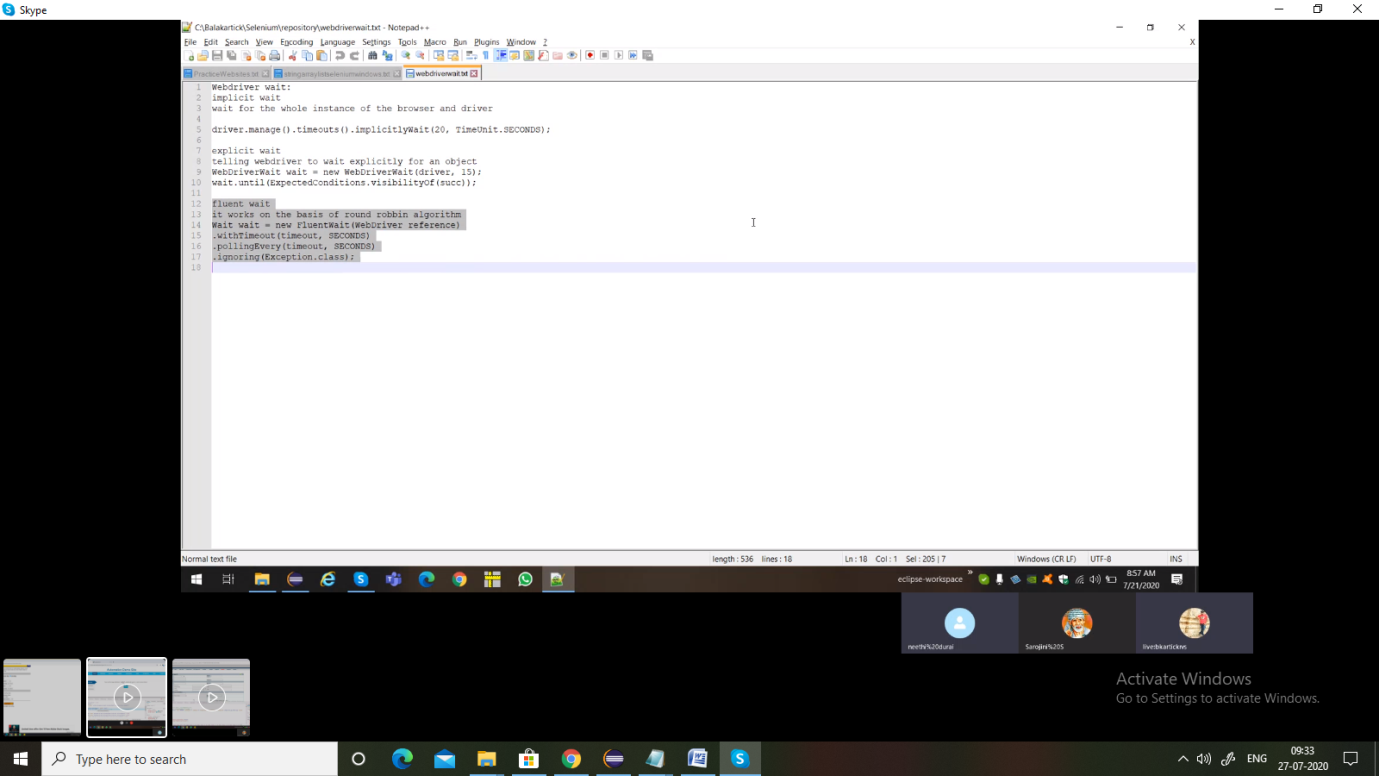
wait.until(ExpectedConditions.*visibilityOf*(newSep));

There are lot of wait for different conditions.

Fluent wait:

It will check polling for the specified time.

Ex polled 4 sec for 20 secs it will verify for 5 times.



round robin algorithm is the name of this wait

Alternate text to img:

//img[@src='original.png' and @alt='image not displaying']

Getting attribute:

String textt = demo.getAttribute("text");

System.out.println("Text: " +textt);

String textt = demo.getAttribute("alt");

We can get all attributes

For src it will get the src value along with the url.

**Switching to default content:**

**package** selenium;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Set;

**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.chrome.ChromeOptions;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver*= **new** ChromeDriver(options);

*driver*.get("http://betanew.edgesoftinc.com:9080/southgate/jsp/index.jsp");

*driver*.findElement(By.*name*("username")).sendKeys("bsadmin");

*driver*.findElement(By.*name*("password")).sendKeys("Sogate1");

*driver*.findElement(By.*name*("Submit")).click();

*driver*.switchTo().frame(1);

WebElement streetField = *driver*.findElement(By.*name*("streetText"));

streetField.isDisplayed();

Thread.*sleep*(2000);

streetField.click();

Thread.*sleep*(3000);

*driver*.findElement(By.*xpath*("//font[.='LSO Explorer']")).click();

Thread.*sleep*(5000);

*driver*.switchTo().defaultContent();

Thread.*sleep*(2000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

File and properties:

**package** selenium;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.OutputStream;

**import** java.util.List;

**import** java.util.Properties;

**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.chrome.ChromeOptions;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

File file = **new** File("C:\\Users\\lenovo\\eclipse-workspace\\Learning\\testData.properties");

FileInputStream fip = **new** FileInputStream(file);

Properties prop = **new** Properties();

prop.load(fip);

String appURL = prop.getProperty("URL");

String username = prop.getProperty("uname");

String password = prop.getProperty("pword");

prop.setProperty("Current", "Powercut");

OutputStream outs = **new** FileOutputStream(file);

prop.store(outs, "test");

*driver*.get(appURL);

*driver*.findElement(By.*name*("username")).sendKeys(username);

*driver*.findElement(By.*name*("password")).sendKeys(password);

*driver*.findElement(By.*name*("Submit")).click();

Thread.*sleep*(5000);

List<WebElement> frames = *driver*.findElements(By.*tagName*("frame"));

**int** count = frames.size();

System.***out***.println("Total number of iframes: " +count);

*driver*.switchTo().frame(1);

WebElement streetField = *driver*.findElement(By.*name*("streetText"));

streetField.isDisplayed();

streetField.click();

*driver*.switchTo().defaultContent();

Thread.*sleep*(5000);

*driver*.quit();

}

**catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 22165

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

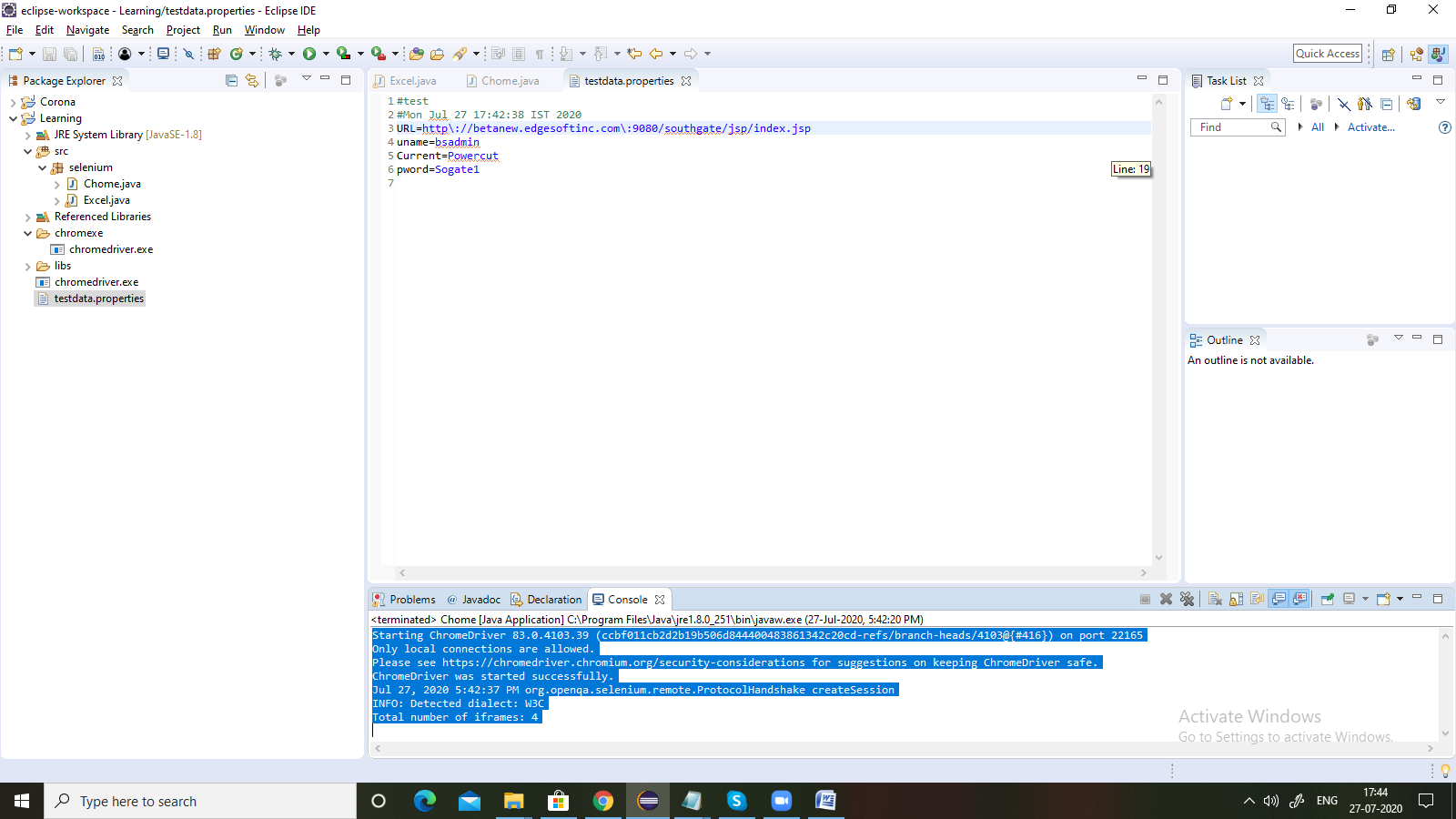
ChromeDriver was started successfully.

Jul 27, 2020 5:42:37 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Total number of iframes: 4

File



Set property can be used to replace the key with new values:

prop.setProperty("uname", "bsadmin");

String appURL = prop.getProperty("URL");

String username = prop.getProperty("uname");

String password = prop.getProperty("pword");

Taking screenshot:

File srcFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

File destFile = new File("C:\\Users\\Balakartick\\eclipse-workspace\\Sample\\Snapshots\\test.png");

FileUtils.copyFile(srcFile, destFile);

**package** selenium;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Calendar;

**import** java.util.List;

**import** java.util.Properties;

**import** org.apache.commons.io.FileUtils;

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

**import** org.openqa.selenium.OutputType;

**import** org.openqa.selenium.TakesScreenshot;

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

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

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

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

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** Properties *prop*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

File file = **new** File("C:\\Users\\lenovo\\eclipse-workspace\\Learning\\testData.properties");

FileInputStream fip = **new** FileInputStream(file);

*prop* = **new** Properties();

*prop*.load(fip);

String appURL = *prop*.getProperty("URL");

String username = *prop*.getProperty("uname");

String password = *prop*.getProperty("pword");

*driver*.get(appURL);

*driver*.findElement(By.*name*("username")).sendKeys(username);

*driver*.findElement(By.*name*("password")).sendKeys(password);

*driver*.findElement(By.*name*("Submit")).click();

Thread.*sleep*(3000);

List<WebElement> frames = *driver*.findElements(By.*tagName*("frame"));

**int** count = frames.size();

System.***out***.println("Total number of iframes: " +count);

*driver*.switchTo().frame(1);

WebElement streetField = *driver*.findElement(By.*name*("streetText"));

streetField.isDisplayed();

streetField.click();

*driver*.switchTo().defaultContent();

*takeSnapShot*(*driver*, "testers");

Thread.*sleep*(2000);

*driver*.quit();

}

**catch** (Exception e) {

e.printStackTrace();

}

}

**public** **static** **void** takeSnapShot(WebDriver driver, String fileName) {

**try** {

DateFormat dateFormat = **new** SimpleDateFormat("HH\_mm\_ss");

Calendar cal = Calendar.*getInstance*();

System.***out***.println("\n"+cal.getTime());

String currdatentimestamp = dateFormat.format(cal.getTime());

System.***out***.println(currdatentimestamp);

File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.***FILE***);

String suffix = fileName+"\_"+currdatentimestamp;

String filepath = "C:\\Users\\lenovo\\eclipse-workspace\\Learning\\Snapshots\\"+suffix+".png";

System.***out***.println(filepath);

File destFile = **new** File(filepath);

FileUtils.*copyFile*(scrFile, destFile);

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 34680

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 27, 2020 7:20:22 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Total number of iframes: 4

Mon Jul 27 19:20:33 IST 2020

19\_20\_33

C:\Users\lenovo\eclipse-workspace\Learning\Snapshots\testers\_19\_20\_33.png

Accessing table values:

**package** selenium;

**import** java.util.List;

**import** java.util.Properties;

**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.chrome.ChromeOptions;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** Properties *prop*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("https://www.seleniumeasy.com/test/table-sort-search-demo.html");

WebElement tableWhole = *driver*.findElement(By.*xpath*("//table[@id='example']"));

List<WebElement> tcol = tableWhole.findElements(By.*tagName*("th"));

**int** colcount = tcol.size();

System.***out***.println("Number of columns: " +colcount);

**for**(**int** i = 0; i < colcount; i++) {

System.***out***.println("Table header index: " +i);

String tcoldata = tcol.get(i).getText();

System.***out***.println("Table header value: " +tcoldata);

}

List<WebElement> trow = tableWhole.findElements(By.*tagName*("tr"));

**int** rowcount = trow.size();

System.***out***.println("Number of rows: " +rowcount);

**for**(**int** row = 1; row < rowcount; row++) {

List<WebElement> Columns\_row = trow.get(row).findElements(By.*tagName*("td"));

//To calculate no of columns (cells). In that specific row.row index starts from 1 since the first row is header

**int** columns\_count = Columns\_row.size();

System.***out***.println("Number of cells In Row " + row + " are " + columns\_count);

**for** (**int** column = 0; column < columns\_count; column++) {

// To retrieve text from that specific cell.column count starts from 0 since the first column index starts with 0

String celtext = Columns\_row.get(column).getText();

System.***out***.println("Cell Value of row number " + row + " and column number " + column + " Is " + celtext);

}

System.***out***.println("-------------------------------------------------- ");

}

Thread.*sleep*(2000);

*driver*.quit();

}

**catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 33104

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 27, 2020 8:17:30 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Number of columns: 6

Table header index: 0

Table header value: Name

Table header index: 1

Table header value: Position

Table header index: 2

Table header value: Office

Table header index: 3

Table header value: Age

Table header index: 4

Table header value: Start date

Table header index: 5

Table header value: Salary

Number of rows: 11

Number of cells In Row 1 are 6

Cell Value of row number 1 and column number 0 Is A. Cox

Cell Value of row number 1 and column number 1 Is Junior Technical Author

Cell Value of row number 1 and column number 2 Is San Francisco

Cell Value of row number 1 and column number 3 Is 66

Cell Value of row number 1 and column number 4 Is Mon 12th Jan 09

Cell Value of row number 1 and column number 5 Is $86,000/y

--------------------------------------------------

Number of cells In Row 2 are 6

Cell Value of row number 2 and column number 0 Is A. Ramos

Cell Value of row number 2 and column number 1 Is Chief Executive Officer (CEO)

Cell Value of row number 2 and column number 2 Is London

Cell Value of row number 2 and column number 3 Is 47

Cell Value of row number 2 and column number 4 Is Fri 9th Oct 09

Cell Value of row number 2 and column number 5 Is $1,200,000/y

--------------------------------------------------

Number of cells In Row 3 are 6

Cell Value of row number 3 and column number 0 Is A. Satou

Cell Value of row number 3 and column number 1 Is Accountant

Cell Value of row number 3 and column number 2 Is Tokyo

Cell Value of row number 3 and column number 3 Is 33

Cell Value of row number 3 and column number 4 Is Fri 28th Nov 08

Cell Value of row number 3 and column number 5 Is $162,700/y

--------------------------------------------------

Number of cells In Row 4 are 6

Cell Value of row number 4 and column number 0 Is B. Greer

Cell Value of row number 4 and column number 1 Is Software Engineer

Cell Value of row number 4 and column number 2 Is London

Cell Value of row number 4 and column number 3 Is 41

Cell Value of row number 4 and column number 4 Is Sat 13th Oct 12

Cell Value of row number 4 and column number 5 Is $132,000/y

--------------------------------------------------

Number of cells In Row 5 are 6

Cell Value of row number 5 and column number 0 Is B. Wagner

Cell Value of row number 5 and column number 1 Is Software Engineer

Cell Value of row number 5 and column number 2 Is San Francisco

Cell Value of row number 5 and column number 3 Is 28

Cell Value of row number 5 and column number 4 Is Tue 7th Jun 11

Cell Value of row number 5 and column number 5 Is $206,850/y

--------------------------------------------------

Number of cells In Row 6 are 6

Cell Value of row number 6 and column number 0 Is B. Williamson

Cell Value of row number 6 and column number 1 Is Integration Specialist

Cell Value of row number 6 and column number 2 Is New York

Cell Value of row number 6 and column number 3 Is 61

Cell Value of row number 6 and column number 4 Is Sun 2nd Dec 12

Cell Value of row number 6 and column number 5 Is $372,000/y

--------------------------------------------------

Number of cells In Row 7 are 6

Cell Value of row number 7 and column number 0 Is C. Hurst

Cell Value of row number 7 and column number 1 Is Javascript Developer

Cell Value of row number 7 and column number 2 Is San Francisco

Cell Value of row number 7 and column number 3 Is 39

Cell Value of row number 7 and column number 4 Is Tue 15th Sep 09

Cell Value of row number 7 and column number 5 Is $205,500/y

--------------------------------------------------

Number of cells In Row 8 are 6

Cell Value of row number 8 and column number 0 Is C. Kelly

Cell Value of row number 8 and column number 1 Is Senior Javascript Developer

Cell Value of row number 8 and column number 2 Is Edinburgh

Cell Value of row number 8 and column number 3 Is 22

Cell Value of row number 8 and column number 4 Is Thu 29th Mar 12

Cell Value of row number 8 and column number 5 Is $433,060/y

--------------------------------------------------

Number of cells In Row 9 are 6

Cell Value of row number 9 and column number 0 Is C. Marshall

Cell Value of row number 9 and column number 1 Is Regional Director

Cell Value of row number 9 and column number 2 Is San Francisco

Cell Value of row number 9 and column number 3 Is 36

Cell Value of row number 9 and column number 4 Is Thu 16th Oct 08

Cell Value of row number 9 and column number 5 Is $470,600/y

--------------------------------------------------

Number of cells In Row 10 are 6

Cell Value of row number 10 and column number 0 Is C. Vance

Cell Value of row number 10 and column number 1 Is Pre-Sales Support

Cell Value of row number 10 and column number 2 Is New York

Cell Value of row number 10 and column number 3 Is 21

Cell Value of row number 10 and column number 4 Is Mon 12th Dec 11

Cell Value of row number 10 and column number 5 Is $106,450/y

--------------------------------------------------

**Next button until disable**

**package** selenium;

**import** java.util.List;

**import** java.util.Properties;

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

**import** org.openqa.selenium.NoSuchElementException;

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

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

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

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

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** Properties *prop*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("https://www.seleniumeasy.com/test/table-sort-search-demo.html");

**for**(**int** i=0; i<=10; i++) {

WebElement nextBtn = *driver*.findElement(By.*xpath*("//a[@class='paginate\_button next']"));

System.***out***.println("Iteration: " +i);

**if**(nextBtn.isEnabled()==**true**) {

nextBtn.click();

System.***out***.println("Button Clicked");

}

}

}

**catch** (NoSuchElementException ne) {

System.***out***.println("Next button is disabled!");

*driver*.quit();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 13246

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

Jul 27, 2020 8:59:29 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Iteration: 0

Button Clicked

Iteration: 1

Button Clicked

Iteration: 2

Button Clicked

Next button is disabled!

Extracting data from excel:

**package** selenium;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** org.apache.poi.ss.usermodel.Cell;

**import** org.apache.poi.ss.usermodel.Row;

**import** org.apache.poi.ss.usermodel.Sheet;

**import** org.apache.poi.xssf.usermodel.XSSFWorkbook;

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

**public** **class** Chome {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

File file = **new** File("C:\\Users\\lenovo\\eclipse-workspace\\Learning\\testData.xlsx");

FileInputStream fis = **new** FileInputStream(file);

XSSFWorkbook wb = **new** XSSFWorkbook(fis);

// HSSFWorkbook used for .xls || XSSFWorkbook used for .xlsx

Sheet sheet = wb.getSheet("demoaut");

Row row = sheet.getRow(10);

Cell cell = row.getCell(5);

String data = cell.getStringCellValue();

System.***out***.println(data);

wb.close();

}

**catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Guna

Date picker:

**package** selenium;

**import** java.util.Calendar;

**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.chrome.ChromeOptions;

**import** org.openqa.selenium.support.ui.Select;

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.addArguments("disable-features=NetworkService");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("http://demo.automationtesting.in/Datepicker.html");

Thread.*sleep*(2000);

WebElement datepick = *driver*.findElement(By.*xpath*("//input[@id='datepicker2']"));

datepick.click();

/\*datepick.sendKeys("03/04/1991");

driver.findElement(By.xpath("//a[.='Close']")).click();\*/

**int** yr = 1900;

String year = String.*valueOf*(yr);

**int** curryear = Calendar.*getInstance*().get(Calendar.***YEAR***);

System.***out***.println(curryear);

**int** diff = ((curryear-yr)/10)+1;

**int** yearr = (1-((curryear-yr)%10))+1;

String twok = "//option[@value='7/2000']";

String eighty = "//option[@value='7/1980']";

String sixty = "//option[@value='7/1960']";

String fourty = "//option[@value='7/1940']";

String twenty = "//option[@value='7/1920']";

String ninecent = "//option[@value='7/1900']";

String yob = "//option[@value='7/"+year+"']";

System.***out***.println(yob);

/\*WebElement changeyear = driver.findElement(By.xpath("//select[@title='Change the year']"));

Select yearsel = new Select(changeyear);

Thread.sleep(1000);

driver.findElement(By.xpath("//select[@title='Change the year']")).click();

yearsel.selectByIndex(0);

driver.findElement(By.xpath("//select[@title='Change the year']")).click();

driver.findElement(By.xpath(twok)).click();

driver.findElement(By.xpath("//select[@title='Change the year']")).click();

driver.findElement(By.xpath(yob)).click();\*/

WebElement changeyear = *driver*.findElement(By.*xpath*("//select[@title='Change the year']"));

Select yearsel = **new** Select(changeyear);

**switch**(diff)

{

**case** 1:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

yearsel.selectByVisibleText(year);

System.***out***.println("Case 1");

**break**;

**case** 2:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 2");

**break**;

**case** 3:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 3");

**break**;

**case** 4:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 4");

**break**;

**case** 5:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 5");

**break**;

**case** 6:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(sixty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 6");

**break**;

**case** 7:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(sixty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 7");

**break**;

**case** 8:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(sixty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(fourty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 8");

**break**;

**case** 9:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(sixty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(fourty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 9");

**break**;

**case** 10:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(sixty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(fourty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twenty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 10");

**break**;

**case** 11:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(sixty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(fourty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twenty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 11");

**break**;

**case** 12:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(sixty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(fourty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twenty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(ninecent)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 12");

**break**;

**case** 13:

Thread.*sleep*(1000);

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twok)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(eighty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(sixty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(fourty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(twenty)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(ninecent)).click();

*driver*.findElement(By.*xpath*("//select[@title='Change the year']")).click();

*driver*.findElement(By.*xpath*(yob)).click();

System.***out***.println("Case 13");

**break**;

}

WebElement changemonth = *driver*.findElement(By.*xpath*("//select[@title='Change the month']"));

Select monsel = **new** Select(changemonth);

changemonth.click();

Thread.*sleep*(1000);

monsel.selectByVisibleText("August");

Thread.*sleep*(1000);

//System.out.println(diff);

//yearsel.selectByVisibleText("1991");

*driver*.findElement(By.*xpath*("//a[.='13']")).click();

Thread.*sleep*(4000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

TestNG framework:

Example with annotations:

package selenium;

import org.openqa.selenium.WebDriver;

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 Pract {

public static WebDriver driver;

@Test()

public void test1() {

try {

System.out.println("Test method 1!");

} catch (Exception e) {

e.printStackTrace();

}

}

@Test()

public void test2() {

try {

System.out.println("Test method 2!");

} catch (Exception e) {

e.printStackTrace();

}

}

@Test

public void test3() {

try {

System.out.println("Test method 3!");

} catch (Exception e) {

e.printStackTrace();

}

}

@AfterSuite

public void aftSuite() {

try {

System.out.println("After suite!");

} catch (Exception e) {

e.printStackTrace();

}

}

@BeforeClass

public void befClass() {

try {

System.out.println("Before Class!");

} catch (Exception e) {

e.printStackTrace();

}

}

@AfterClass

public void aftClass() {

try {

System.out.println("After Class!");

} catch (Exception e) {

e.printStackTrace();

}

}

@BeforeMethod

public void befMet() {

try {

System.out.println("Before Method!");

} catch (Exception e) {

e.printStackTrace();

}

}

@AfterMethod

public void aftMet() {

try {

System.out.println("After Method!");

} catch (Exception e) {

e.printStackTrace();

}

}

@BeforeTest

public void befTest() {

try {

System.out.println("Before Test!");

} catch (Exception e) {

e.printStackTrace();

}

}

@BeforeSuite

public void befSuite() {

try {

System.out.println("Before suite!");

} catch (Exception e) {

e.printStackTrace();

}

}

@AfterTest

public void aftTest() {

try {

System.out.println("After Test!");

} catch (Exception e) {

e.printStackTrace();

}

}

}

Fillo data base code:

**package** selenium;

**import** com.codoid.products.fillo.Connection;

**import** com.codoid.products.fillo.Fillo;

**import** com.codoid.products.fillo.Recordset;

**public** **class** Chome {

**public** **static** **void** main(String args[]) {

**try** {

Fillo fillo = **new** Fillo();

Connection connection = fillo.getConnection("C:\\Users\\lenovo\\eclipse-workspace\\Learning\\testData.xlsx");

String strQuery = "select \* from demoaut";

Recordset recordset = connection.executeQuery(strQuery);

**int** count = recordset.getCount();

System.***out***.println("Total number of rows: " +count);

recordset.moveFirst();

**for**(**int** i = 0; i < count; i++) {

Thread.*sleep*(500);

String datax = recordset.getField("Firstname");

System.***out***.println("Iteration: " +i);

System.***out***.println("Firstname: " +datax);

recordset.next();

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Total number of rows: 15

Iteration: 0

Firstname: DemoTest

Iteration: 1

Firstname: Bala

Iteration: 2

Firstname: Kalyan

Iteration: 3

Firstname: Divya

Iteration: 4

Firstname: Praveen

Iteration: 5

Firstname: Hello

Iteration: 6

Firstname: Kerala

Iteration: 7

Firstname: King

Iteration: 8

Firstname: Guru

Iteration: 9

Firstname: Retro

Iteration: 10

Firstname: Save

Iteration: 11

Firstname: Control

Iteration: 12

Firstname: Firetest

Iteration: 13

Firstname: Speed

Iteration: 14

Firstname: Thought

Automating registration using fillo:

**package** selenium;

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

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

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

**import** com.codoid.products.fillo.Connection;

**import** com.codoid.products.fillo.Fillo;

**import** com.codoid.products.fillo.Recordset;

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

Fillo fillo = **new** Fillo();

Connection connection = fillo.getConnection("C:\\Users\\lenovo\\eclipse-workspace\\Learning\\testData.xlsx");

String strQuery = "select \* from demoaut";

Recordset recordset = connection.executeQuery(strQuery);

**int** count = recordset.getCount();

System.***out***.println("Total number of rows: " +count);

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\lenovo\\eclipse-workspace\\Learning\\chromexe\\chromedriver.exe");

*driver* = **new** ChromeDriver();

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

**for**(**int** i = 0; i < count; i++) {

recordset.moveNext();

Thread.*sleep*(500);

String tcid = recordset.getField("TC\_ID");

String fname = recordset.getField("Firstname");

String lname = recordset.getField("Lastname");

String pno = recordset.getField("Phone");

String email = recordset.getField("Email");

String uname = recordset.getField("Username");

String pwd = recordset.getField("Password");

String cpwd = recordset.getField("Confirm");

*driver*.get("http://demo.guru99.com/test/newtours/register.php");

*driver*.findElement(By.*name*("firstName")).sendKeys(fname);

*driver*.findElement(By.*name*("lastName")).sendKeys(lname);

*driver*.findElement(By.*name*("phone")).sendKeys(pno);

*driver*.findElement(By.*id*("userName")).sendKeys(email);

*driver*.findElement(By.*name*("address1")).sendKeys("test street");

*driver*.findElement(By.*name*("city")).sendKeys("Chennai");

*driver*.findElement(By.*name*("state")).sendKeys("TamilNadu");

*driver*.findElement(By.*name*("country")).sendKeys("INDIA");

*driver*.findElement(By.*id*("email")).sendKeys(uname);

*driver*.findElement(By.*name*("password")).sendKeys(pwd);

*driver*.findElement(By.*name*("confirmPassword")).sendKeys(cpwd);

*driver*.findElement(By.*name*("submit")).click();

System.***out***.println("Iteration: " +i);

System.***out***.println("TC\_ID: "+tcid+" Firstname: " +fname);

String actMsg = *driver*.findElement(By.*xpath*("//a[.=' sign-in ']/following::b[1]")).getText();

System.***out***.println(actMsg);

**if**(actMsg.contains(uname)) {

System.***out***.println("User creation passed! The created Username is: "+uname);

}

**else** {

System.***out***.println("User creation failed!");

}

}

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Total number of rows: 15

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 32616

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1596020286.778][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Jul 29, 2020 4:28:08 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Iteration: 0

TC\_ID: TC\_01 Firstname: DemoTest

Note: Your user name is DemoTest.

User creation passed! The created Username is: DemoTest

Iteration: 1

TC\_ID: TC\_02 Firstname: Bala

Note: Your user name is Bala.

User creation passed! The created Username is: Bala

Iteration: 2

TC\_ID: TC\_03 Firstname: Kalyan

Note: Your user name is Kalyan.

User creation passed! The created Username is: Kalyan

Iteration: 3

TC\_ID: TC\_04 Firstname: Divya

Note: Your user name is Jink.

User creation passed! The created Username is: Jink

Iteration: 4

TC\_ID: TC\_05 Firstname: Praveen

Note: Your user name is Tinkles.

User creation passed! The created Username is: Tinkles

Iteration: 5

TC\_ID: TC\_06 Firstname: Hello

Note: Your user name is Balu.

User creation passed! The created Username is: Balu

Iteration: 6

TC\_ID: TC\_07 Firstname: Kerala

Note: Your user name is Dravid.

User creation passed! The created Username is: Dravid

Iteration: 7

TC\_ID: TC\_08 Firstname: King

Note: Your user name is Hari.

User creation passed! The created Username is: Hari

Iteration: 8

TC\_ID: TC\_09 Firstname: Guru

Note: Your user name is Ranjith.

User creation passed! The created Username is: Ranjith

Iteration: 9

TC\_ID: TC\_10 Firstname: Retro

Note: Your user name is Guna.

User creation passed! The created Username is: Guna

Iteration: 10

TC\_ID: TC\_11 Firstname: Save

Note: Your user name is Sekar.

User creation passed! The created Username is: Sekar

Iteration: 11

TC\_ID: TC\_12 Firstname: Control

Note: Your user name is Rajan.

User creation passed! The created Username is: Rajan

Iteration: 12

TC\_ID: TC\_13 Firstname: Firetest

Note: Your user name is Velu.

User creation passed! The created Username is: Velu

Iteration: 13

TC\_ID: TC\_14 Firstname: Speed

Note: Your user name is Abinath.

User creation passed! The created Username is: Abinath

Iteration: 14

TC\_ID: TC\_15 Firstname: Thought

Note: Your user name is Freak.

User creation passed! The created Username is: Freak

Automation of registration using fillo in Testng framework:

**package** selenium;

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

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

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

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

**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;

**import** com.codoid.products.fillo.Connection;

**import** com.codoid.products.fillo.Fillo;

**import** com.codoid.products.fillo.Recordset;

**public** **class** Chome {

**public** **static** WebDriver *driver*;

@Test()

**public** **void** test1() {

**try** {

Fillo fillo = **new** Fillo();

Connection connection = fillo.getConnection("C:\\Users\\lenovo\\eclipse-workspace\\Learning\\testData.xlsx");

String strQuery = "select \* from demoaut";

Recordset recordset = connection.executeQuery(strQuery);

**int** count = recordset.getCount();

System.***out***.println("Total number of rows: " +count);

**for**(**int** i = 0; i < count; i++) {

recordset.moveNext();

Thread.*sleep*(500);

String tcid = recordset.getField("TC\_ID");

String fname = recordset.getField("Firstname");

String lname = recordset.getField("Lastname");

String pno = recordset.getField("Phone");

String email = recordset.getField("Email");

String uname = recordset.getField("Username");

String pwd = recordset.getField("Password");

String cpwd = recordset.getField("Confirm");

*driver*.get("http://demo.guru99.com/test/newtours/register.php");

*driver*.findElement(By.*name*("firstName")).sendKeys(fname);

*driver*.findElement(By.*name*("lastName")).sendKeys(lname);

*driver*.findElement(By.*name*("phone")).sendKeys(pno);

*driver*.findElement(By.*id*("userName")).sendKeys(email);

*driver*.findElement(By.*name*("address1")).sendKeys("test street");

*driver*.findElement(By.*name*("city")).sendKeys("Chennai");

*driver*.findElement(By.*name*("state")).sendKeys("TamilNadu");

*driver*.findElement(By.*name*("country")).sendKeys("INDIA");

*driver*.findElement(By.*id*("email")).sendKeys(uname);

*driver*.findElement(By.*name*("password")).sendKeys(pwd);

*driver*.findElement(By.*name*("confirmPassword")).sendKeys(cpwd);

*driver*.findElement(By.*name*("submit")).click();

System.***out***.println("Iteration: " +i);

System.***out***.println("TC\_ID: "+tcid+" Firstname: " +fname);

String actMsg = *driver*.findElement(By.*xpath*("//a[.=' sign-in ']/following::b[1]")).getText();

System.***out***.println(actMsg);

**if**(actMsg.contains(uname)) {

System.***out***.println("User creation passed! The created Username is: "+uname);

}

**else** {

System.***out***.println("User creation failed!");

}

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterSuite

**public** **void** aftSuite() {

**try** {

System.***out***.println("After suite!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeClass

**public** **void** befClass() {

**try** {

System.***out***.println("Before Class!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterClass

**public** **void** aftClass() {

**try** {

System.***out***.println("After Class!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeMethod

**public** **void** befMet() {

**try** {

System.***out***.println("Before Method!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterMethod

**public** **void** aftMet() {

**try** {

System.***out***.println("After Method!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeTest

**public** **void** befTest() {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeSuite

**public** **void** befSuite() {

**try** {

System.***out***.println("Before suite!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterTest

**public** **void** aftTest() {

**try** {

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

[RemoteTestNG] detected TestNG version 6.9.10

[TestNGContentHandler] [WARN] It is strongly recommended to add "<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >" at the top of your file, otherwise TestNG may fail or not work as expected.

[TestNG] Running:

C:\Users\lenovo\eclipse-workspace\Learning\testng.xml

Before suite!

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 30297

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1596021270.499][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Jul 29, 2020 4:44:32 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Before Class!

Before Method!

Total number of rows: 15

Iteration: 0

TC\_ID: TC\_01 Firstname: DemoTest

Note: Your user name is DemoTest.

User creation passed! The created Username is: DemoTest

Iteration: 1

TC\_ID: TC\_02 Firstname: Bala

Note: Your user name is Bala.

User creation passed! The created Username is: Bala

Iteration: 2

TC\_ID: TC\_03 Firstname: Kalyan

Note: Your user name is Kalyan.

User creation passed! The created Username is: Kalyan

Iteration: 3

TC\_ID: TC\_04 Firstname: Divya

Note: Your user name is Jink.

User creation passed! The created Username is: Jink

Iteration: 4

TC\_ID: TC\_05 Firstname: Praveen

Note: Your user name is Tinkles.

User creation passed! The created Username is: Tinkles

Iteration: 5

TC\_ID: TC\_06 Firstname: Hello

Note: Your user name is Balu.

User creation passed! The created Username is: Balu

Iteration: 6

TC\_ID: TC\_07 Firstname: Kerala

Note: Your user name is Dravid.

User creation passed! The created Username is: Dravid

Iteration: 7

TC\_ID: TC\_08 Firstname: King

Note: Your user name is Hari.

User creation passed! The created Username is: Hari

Iteration: 8

TC\_ID: TC\_09 Firstname: Guru

Note: Your user name is Ranjith.

User creation passed! The created Username is: Ranjith

Iteration: 9

TC\_ID: TC\_10 Firstname: Retro

Note: Your user name is Guna.

User creation passed! The created Username is: Guna

Iteration: 10

TC\_ID: TC\_11 Firstname: Save

Note: Your user name is Sekar.

User creation passed! The created Username is: Sekar

Iteration: 11

TC\_ID: TC\_12 Firstname: Control

Note: Your user name is Rajan.

User creation passed! The created Username is: Rajan

Iteration: 12

TC\_ID: TC\_13 Firstname: Firetest

Note: Your user name is Velu.

User creation passed! The created Username is: Velu

Iteration: 13

TC\_ID: TC\_14 Firstname: Speed

Note: Your user name is Abinath.

User creation passed! The created Username is: Abinath

Iteration: 14

TC\_ID: TC\_15 Firstname: Thought

Note: Your user name is Freak.

User creation passed! The created Username is: Freak

After Method!

After Class!

After suite!

===============================================

Suite

Total tests run: 1, Failures: 0, Skips: 0

===============================================

Verbose:

Verbose will increase the content displayed in output with the increase in value.

<https://testng.org/doc/documentation-main.html#annotations>

@ignore

@Test(enabled=false)

Parameter in testing

:

XML:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

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

<suite name=*"Suite"*>

<test thread-count=*"5"* name=*"Test"*>

<parameter name=*"username"* value = *"mango"*/>

<parameter name=*"password"* value = *"orange"*/>

<classes>

<class name=*"selenium.Dummy"*/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

Code:

**package** selenium;

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

**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.Parameters;

**import** org.testng.annotations.Test;

**public** **class** Dummy {

**public** **static** WebDriver *driver*;

@Test()

@Parameters({"username", "password"})

**public** **void** test1(String uname, String pwd) {

**try** {

System.***out***.println("Test method 1!");

System.***out***.println("Parameter value for username: " +uname);

System.***out***.println("Parameter value for password: " +pwd);

} **catch** (Exception e) {

e.printStackTrace();

}

}

@Test

**public** **void** test2() {

**try** {

System.***out***.println("Test method 2!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@Test

**public** **void** test3() {

**try** {

System.***out***.println("Test method 3!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@Test

**public** **void** test4() {

**try** {

System.***out***.println("Test method 4!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@Test

**public** **void** test5() {

**try** {

System.***out***.println("Test method 5!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@Test

**public** **void** test6() {

**try** {

System.***out***.println("Test method 6!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@Test

**public** **void** test7() {

**try** {

System.***out***.println("Test method 7!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@Test

**public** **void** test8() {

**try** {

System.***out***.println("Test method 8!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterSuite

**public** **void** aftSuite() {

**try** {

System.***out***.println("After suite!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeClass

**public** **void** befClass() {

**try** {

System.***out***.println("Before Class!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterClass

**public** **void** aftClass() {

**try** {

System.***out***.println("After Class!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeMethod

**public** **void** befMet() {

**try** {

System.***out***.println("Before Method!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterMethod

**public** **void** aftMet() {

**try** {

System.***out***.println("After Method!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeTest

**public** **void** befTest() {

**try** {

System.***out***.println("Before Test!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeSuite

**public** **void** befSuite() {

**try** {

System.***out***.println("Before suite!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterTest

**public** **void** aftTest() {

**try** {

System.***out***.println("After Test!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

[RemoteTestNG] detected TestNG version 6.9.10

[TestNGContentHandler] [WARN] It is strongly recommended to add "<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >" at the top of your file, otherwise TestNG may fail or not work as expected.

[TestNG] Running:

C:\Users\lenovo\eclipse-workspace\Learning\testng.xml

Before suite!

Before Test!

Before Class!

Before Method!

Test method 1!

Parameter value for username: mango

Parameter value for password: orange

After Method!

Before Method!

Test method 2!

After Method!

Before Method!

Test method 3!

After Method!

Before Method!

Test method 4!

After Method!

Before Method!

Test method 5!

After Method!

Before Method!

Test method 6!

After Method!

Before Method!

Test method 7!

After Method!

Before Method!

Test method 8!

After Method!

After Class!

After Test!

After suite!

===============================================

Suite

Total tests run: 8, Failures: 0, Skips: 0

===============================================

To condition what methods to be executed:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

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

<suite name=*"Suite"*>

<parameter name=*"username"* value=*"mango"* />

<parameter name=*"password"* value=*"orange"* />

<test thread-count=*"5"* name=*"Test"*>

<classes>

<class name=*"selenium.Dummy"*>

<methods>

<include name=*"test1"* />

<include name=*"test3"* />

<include name=*"test2"* />

</methods>

</class>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

Output:

[RemoteTestNG] detected TestNG version 6.9.10

[TestNGContentHandler] [WARN] It is strongly recommended to add "<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >" at the top of your file, otherwise TestNG may fail or not work as expected.

[TestNG] Running:

C:\Users\lenovo\eclipse-workspace\Learning\testng.xml

Before suite!

Before Test!

Before Class!

Before Method!

Test method 1!

Parameter value for username: mango

Parameter value for password: orange

After Method!

Before Method!

Test method 3!

After Method!

Before Method!

Test method 2!

After Method!

After Class!

After Test!

After suite!

===============================================

Suite

Total tests run: 3, Failures: 0, Skips: 0

===============================================

Parallel methods:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

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

<suite name=*"Suite"* verbose=*"2"* parallel=*"methods"* thread-count = *"5"*>

<parameter name=*"username"* value=*"mango"* />

<parameter name=*"password"* value=*"orange"* />

<test thread-count=*"5"* name=*"Test"*>

<classes>

<class name=*"selenium.Dummy"*>

<methods>

<include name=*"test1"* />

<include name=*"test3"* />

<include name=*"test2"* />

</methods>

</class>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

Output:

[RemoteTestNG] detected TestNG version 6.9.10

[TestNGContentHandler] [WARN] It is strongly recommended to add "<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >" at the top of your file, otherwise TestNG may fail or not work as expected.

...

... TestNG 6.9.10 by Cédric Beust (cedric@beust.com)

...

[TestNG] Running:

C:\Users\lenovo\eclipse-workspace\Learning\testng.xml

Before suite!

Before Test!

[TestRunner] Starting executor for test Test with time out:2147483647 milliseconds.

Before Class!

Before Method!

Before Method!

Before Method!

Test method 2!

Test method 1!

Parameter value for username: mango

Parameter value for password: orange

After Method!

Test method 3!

After Method!

After Method!

After Class!

After Test!

PASSED: test1("mango", "orange")

PASSED: test3

PASSED: test2

===============================================

Test

Tests run: 3, Failures: 0, Skips: 0

===============================================

After suite!

===============================================

Suite

Total tests run: 3, Failures: 0, Skips: 0

===============================================

[TestNG] Time taken by org.testng.reporters.XMLReporter@641147d0: 73 ms

[TestNG] Time taken by [FailedReporter passed=0 failed=0 skipped=0]: 31 ms

[TestNG] Time taken by org.testng.reporters.jq.Main@314c508a: 94 ms

[TestNG] Time taken by org.testng.reporters.SuiteHTMLReporter@6a396c1e: 53 ms

[TestNG] Time taken by org.testng.reporters.EmailableReporter2@38364841: 7 ms

[TestNG] Time taken by org.testng.reporters.JUnitReportReporter@146044d7: 8 ms

Group:

Group name will run tests where the group is specified

Test output folder will be created once the test is run

Once the project is refreshed the output folder will be created.

Kartick automation demoguru

**package** selenium;

**import** java.awt.Robot;

**import** java.awt.event.KeyEvent;

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

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

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

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

**import** org.testng.TestNG;

**import** com.codoid.products.fillo.Connection;

**import** com.codoid.products.fillo.Fillo;

**import** com.codoid.products.fillo.Recordset;

**public** **class** Excel **extends** TestNG{

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String [] args) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("http://way2automation.com/way2auto\_jquery/registration.php");

Thread.*sleep*(5000);

*driver*.findElement(By.*linkText*("Signin")).click();

Thread.*sleep*(2000);

Robot robotic = **new** Robot();

robotic.keyPress(KeyEvent.***VK\_TAB***);

robotic.keyRelease(KeyEvent.***VK\_TAB***);

Thread.*sleep*(1000);

robotType("bkartick");

robotic.keyPress(KeyEvent.***VK\_TAB***);

robotic.keyRelease(KeyEvent.***VK\_TAB***);

Thread.*sleep*(1000);

robotType("Bala");

Thread.*sleep*(1000);

robotic.keyPress(KeyEvent.***VK\_ENTER***);

robotic.keyRelease(KeyEvent.***VK\_ENTER***);

Fillo fillo = **new** Fillo();

Connection connection = fillo.getConnection("C://Users//Balakartick//eclipse-workspace//Sample//testData.xlsx");

String qry="Select \* from registration";

Recordset recordset=connection.executeQuery(qry);

**int** count=recordset.getCount();

System.***out***.println(count);

recordset.moveFirst();

**for**(**int** i=0; i<count; i++) {

*driver*.findElement(By.*linkText*("Registration")).click();

String firstname=recordset.getField("Firstname");

String lastname=recordset.getField("Lastname");

String phone=recordset.getField("Phone");

String email=recordset.getField("Email");

String username = recordset.getField("Username");

String password = recordset.getField("Password");

String confirm = recordset.getField("ConfirmPassword");

*driver*.findElement(By.*name*("name")).sendKeys(firstname);

Thread.*sleep*(500);

*driver*.findElement(By.*xpath*("//label[.='Last Name:']/following::input[1]")).sendKeys(lastname);

Thread.*sleep*(500);

*driver*.findElement(By.*xpath*("//label[contains(text() ,'Reading')]")).click();

Thread.*sleep*(500);

*driver*.findElement(By.*name*("phone")).sendKeys(phone)

;

Thread.*sleep*(500);

*driver*.findElement(By.*name*("username")).sendKeys(username);

Thread.*sleep*(500);

*driver*.findElement(By.*name*("email")).sendKeys(email);

Thread.*sleep*(500);

*driver*.findElement(By.*name*("password")).sendKeys(password);

Thread.*sleep*(500);

*driver*.findElement(By.*name*("c\_password")).sendKeys(confirm);

Thread.*sleep*(500);

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

Thread.*sleep*(5000);

recordset.next();

}

*driver*.findElement(By.*linkText*("Registration")).click();

String reg = *driver*.findElement(By.*xpath*("//input[@name='name']/preceding::h2[1]")).getText();

System.***out***.println(reg);

Fillo fillo = **new** Fillo();

Connection connection = fillo.getConnection("C://Users//Balakartick//eclipse-workspace//Sample//testData.xlsx");

String qry="Update registration Set RegText='"+reg+"' where TC\_ID='TC\_06'";

connection.executeUpdate(qry);

connection.close();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Updating the excel with runtime values using update query:

**package** selenium;

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

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

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

**import** com.codoid.products.fillo.Connection;

**import** com.codoid.products.fillo.Fillo;

**import** com.codoid.products.fillo.Recordset;

**public** **class** Excel {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[]) {

**try** {

Fillo fillo = **new** Fillo();

Connection connection = fillo.getConnection("C:\\Users\\lenovo\\eclipse-workspace\\Learning\\testData.xlsx");

String strQuery = "select \* from demoaut";

Recordset recordset = connection.executeQuery(strQuery);

**int** count = recordset.getCount();

System.***out***.println("Total number of rows: " +count);

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\lenovo\\eclipse-workspace\\Learning\\chromexe\\chromedriver.exe");

*driver* = **new** ChromeDriver();

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

//recordset.moveFirst();

**for**(**int** i = 0; i < count; i++) {

recordset.moveNext();

Thread.*sleep*(500);

String tcid = recordset.getField("TC\_ID");

String fname = recordset.getField("Firstname");

String lname = recordset.getField("Lastname");

String pno = recordset.getField("Phone");

String email = recordset.getField("Email");

String uname = recordset.getField("Username");

String pwd = recordset.getField("Password");

String cpwd = recordset.getField("Confirm");

*driver*.get("http://demo.guru99.com/test/newtours/register.php");

*driver*.findElement(By.*name*("firstName")).sendKeys(fname);

*driver*.findElement(By.*name*("lastName")).sendKeys(lname);

*driver*.findElement(By.*name*("phone")).sendKeys(pno);

*driver*.findElement(By.*id*("userName")).sendKeys(email);

*driver*.findElement(By.*name*("address1")).sendKeys("test street");

*driver*.findElement(By.*name*("city")).sendKeys("Chennai");

*driver*.findElement(By.*name*("state")).sendKeys("TamilNadu");

*driver*.findElement(By.*name*("country")).sendKeys("INDIA");

*driver*.findElement(By.*id*("email")).sendKeys(uname);

*driver*.findElement(By.*name*("password")).sendKeys(pwd);

*driver*.findElement(By.*name*("confirmPassword")).sendKeys(cpwd);

*driver*.findElement(By.*name*("submit")).click();

System.***out***.println("Iteration: " +i);

System.***out***.println("TC\_ID: "+tcid+" Firstname: " +fname);

Thread.*sleep*(2000);

String actMsg = *driver*.findElement(By.*xpath*("//a[.=' sign-in ']/following::b[1]")).getText();

System.***out***.println(actMsg);

String add[] = actMsg.split(" ");

**int** lent = add.length;

System.***out***.println("Total number of words in success message: " +lent);

String unamefin = add[lent-1].replace(".", "");

System.***out***.println("Username created: " +unamefin);

**if**(actMsg.contains(uname)) {

System.***out***.println("User creation passed! The created Username is: "+uname);

}

**else** {

System.***out***.println("User creation failed!");

}

String updateQuery = "Update demoaut Set CreatedUser='"+unamefin+"' where TC\_ID='"+tcid+"'";

connection.executeUpdate(updateQuery);

}

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

Total number of rows: 15

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 26727

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1596125364.276][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Jul 30, 2020 9:39:26 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Iteration: 0

TC\_ID: TC\_01 Firstname: DemoTest

Note: Your user name is DemoTest.

Total number of words in success message: 6

Username created: DemoTest

User creation passed! The created Username is: DemoTest

1 row(s) affected

Iteration: 1

TC\_ID: TC\_02 Firstname: Bala

Note: Your user name is Bala.

Total number of words in success message: 6

Username created: Bala

User creation passed! The created Username is: Bala

1 row(s) affected

Iteration: 2

TC\_ID: TC\_03 Firstname: Kalyan

Note: Your user name is Kalyan.

Total number of words in success message: 6

Username created: Kalyan

User creation passed! The created Username is: Kalyan

1 row(s) affected

Iteration: 3

TC\_ID: TC\_04 Firstname: Divya

Note: Your user name is Jink.

Total number of words in success message: 6

Username created: Jink

User creation passed! The created Username is: Jink

1 row(s) affected

Iteration: 4

TC\_ID: TC\_05 Firstname: Praveen

Note: Your user name is Tinkles.

Total number of words in success message: 6

Username created: Tinkles

User creation passed! The created Username is: Tinkles

1 row(s) affected

Iteration: 5

TC\_ID: TC\_06 Firstname: Hello

Note: Your user name is Balu.

Total number of words in success message: 6

Username created: Balu

User creation passed! The created Username is: Balu

1 row(s) affected

Iteration: 6

TC\_ID: TC\_07 Firstname: Kerala

Note: Your user name is Dravid.

Total number of words in success message: 6

Username created: Dravid

User creation passed! The created Username is: Dravid

1 row(s) affected

Iteration: 7

TC\_ID: TC\_08 Firstname: King

Note: Your user name is Hari.

Total number of words in success message: 6

Username created: Hari

User creation passed! The created Username is: Hari

1 row(s) affected

Iteration: 8

TC\_ID: TC\_09 Firstname: Guru

Note: Your user name is Ranjith.

Total number of words in success message: 6

Username created: Ranjith

User creation passed! The created Username is: Ranjith

1 row(s) affected

Iteration: 9

TC\_ID: TC\_10 Firstname: Retro

Note: Your user name is Guna.

Total number of words in success message: 6

Username created: Guna

User creation passed! The created Username is: Guna

1 row(s) affected

Iteration: 10

TC\_ID: TC\_11 Firstname: Save

Note: Your user name is Sekar.

Total number of words in success message: 6

Username created: Sekar

User creation passed! The created Username is: Sekar

1 row(s) affected

Iteration: 11

TC\_ID: TC\_12 Firstname: Control

Note: Your user name is Rajan.

Total number of words in success message: 6

Username created: Rajan

User creation passed! The created Username is: Rajan

1 row(s) affected

Iteration: 12

TC\_ID: TC\_13 Firstname: Firetest

Note: Your user name is Velu.

Total number of words in success message: 6

Username created: Velu

User creation passed! The created Username is: Velu

1 row(s) affected

Iteration: 13

TC\_ID: TC\_14 Firstname: Speed

Note: Your user name is Abinath.

Total number of words in success message: 6

Username created: Abinath

User creation passed! The created Username is: Abinath

1 row(s) affected

Iteration: 14

TC\_ID: TC\_15 Firstname: Thought

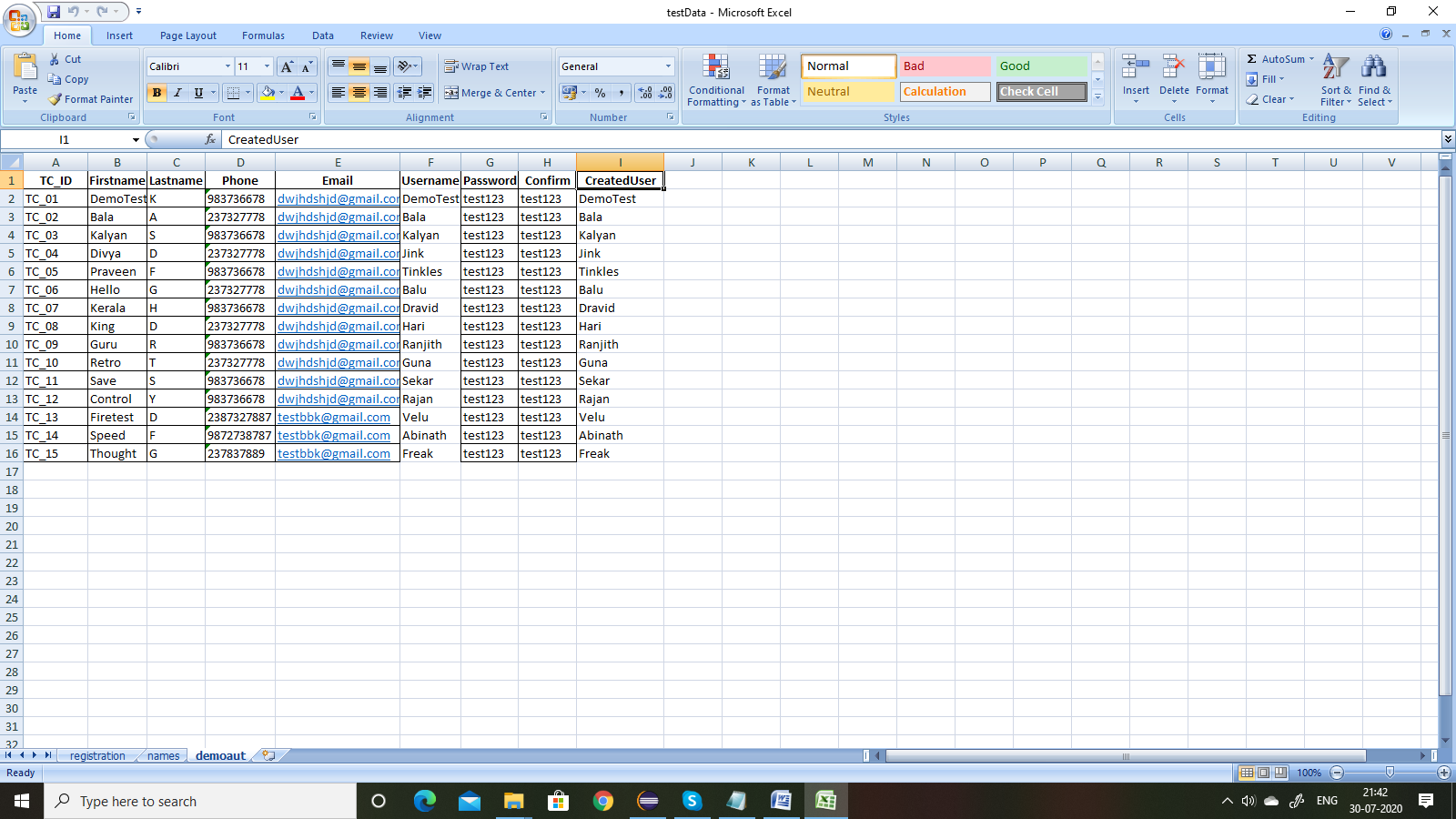
Note: Your user name is Freak.

Total number of words in success message: 6

Username created: Freak

User creation passed! The created Username is: Freak

1 row(s) affected



TestLeaf code:

**package** selenium;

**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.openqa.selenium.chrome.ChromeOptions;

**import** org.openqa.selenium.interactions.Actions;

**public** **class** Dummy {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[])

{

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver*= **new** ChromeDriver(options);

*driver*.get("http://leafground.com/pages/selectable.html");

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

WebElement item1 = *driver*.findElement(By.*xpath*("//ol[@id='selectable']/li[1]"));

WebElement item3 = *driver*.findElement(By.*xpath*("//ol[@id='selectable']/li[3]"));

WebElement item5 = *driver*.findElement(By.*xpath*("//ol[@id='selectable']/li[5]"));

WebElement item7 = *driver*.findElement(By.*xpath*("//ol[@id='selectable']/li[7]"));

Actions action = **new** Actions(*driver*);

Thread.*sleep*(3000);

action.keyDown(Keys.***COMMAND***).click(item5).build().perform();

//action.click(item7).build().perform();

Thread.*sleep*(3000);

*driver*.quit();

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

}

Maven Project:



Full maven project

Auto IT for windows automation:

Opens the message box and selects yes and the items are deleted.

* Install Auto IT
* Edit script
* Run file or convert to exe file and run it.

WinActivate("Delete Multiple Items")

Sleep(1000)

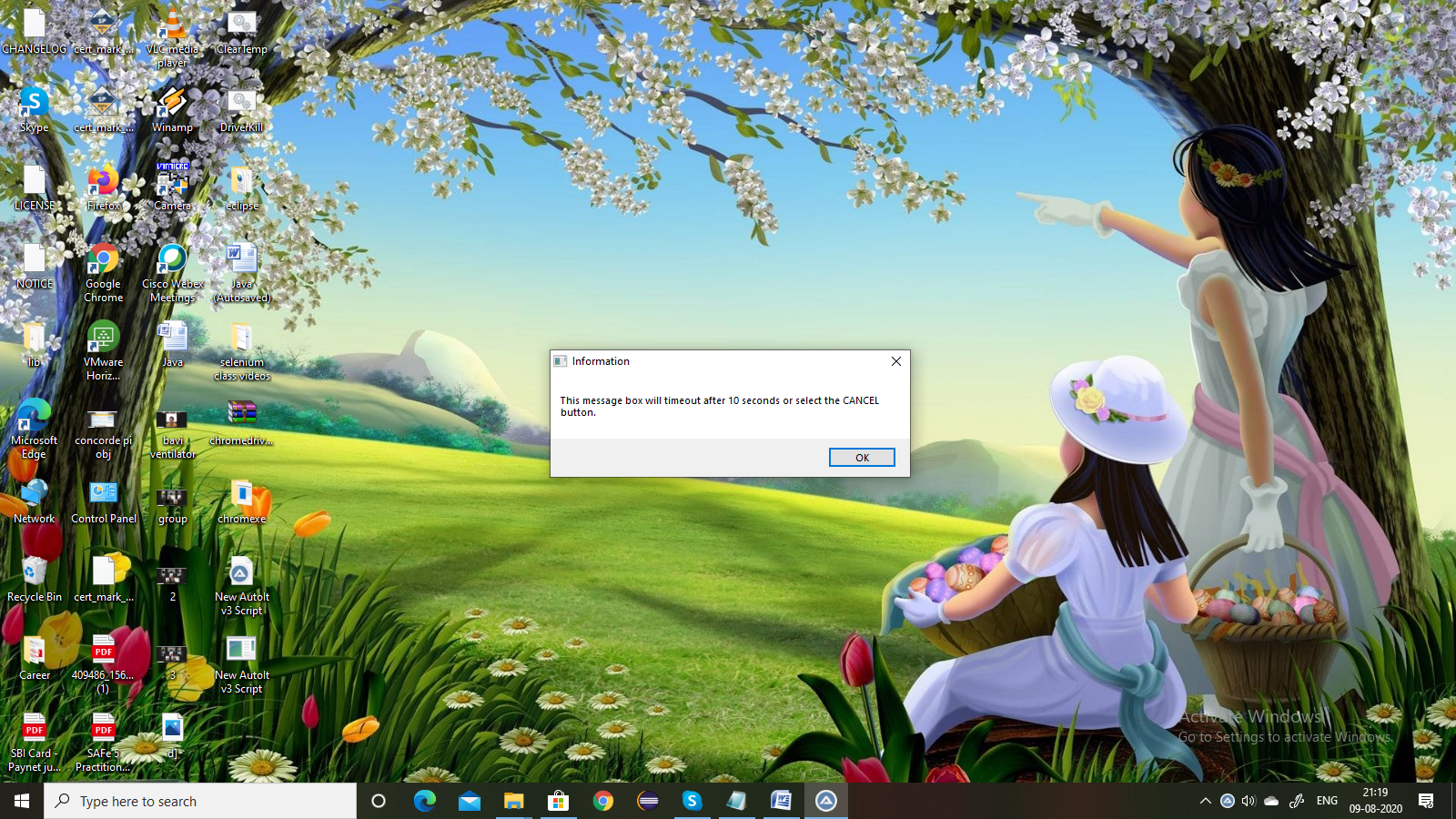
Send("{ENTER}")

Opens a new message box

#include <MsgBoxConstants.au3>

MsgBox($MB\_SYSTEMMODAL, "Information", "This message box will timeout after 10 seconds or select the OK button.")

It has the information as title and the message within the quotes and Ok button.



Uploading file in webpage:

Click on the particular link and the windows page displays

Run the following script and the file is uploaded.

URL: <http://demo.automationtesting.in/Register.html>

WinActivate("Open")

Sleep(2000)

Send("C:\Users\lenovo\Desktop\selenium class videos\img 31juky.png")

Sleep(1000)

#Send("{ENTER}")

ControlClick("Open", "", "[CLASS:Button; INSTANCE:1]")

You can convert to exe and save it in the project folder in eclipse.

File upload using selenium:

**package** selenium;

**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.chrome.ChromeOptions;

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

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

**public** **class** Dummy {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String[] args) {

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.addArguments("disable-features=NetworkService");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver* = **new** ChromeDriver(options);

*driver*.get("http://demo.automationtesting.in/Register.html");

Thread.*sleep*(10000);

WebElement addFile = *driver*.findElement(By.*xpath*("//input[@id='imagesrc']"));

addFile.sendKeys("C:\\Users\\lenovo\\Desktop\\selenium class videos\\img 31juky.png");

Thread.*sleep*(2000);

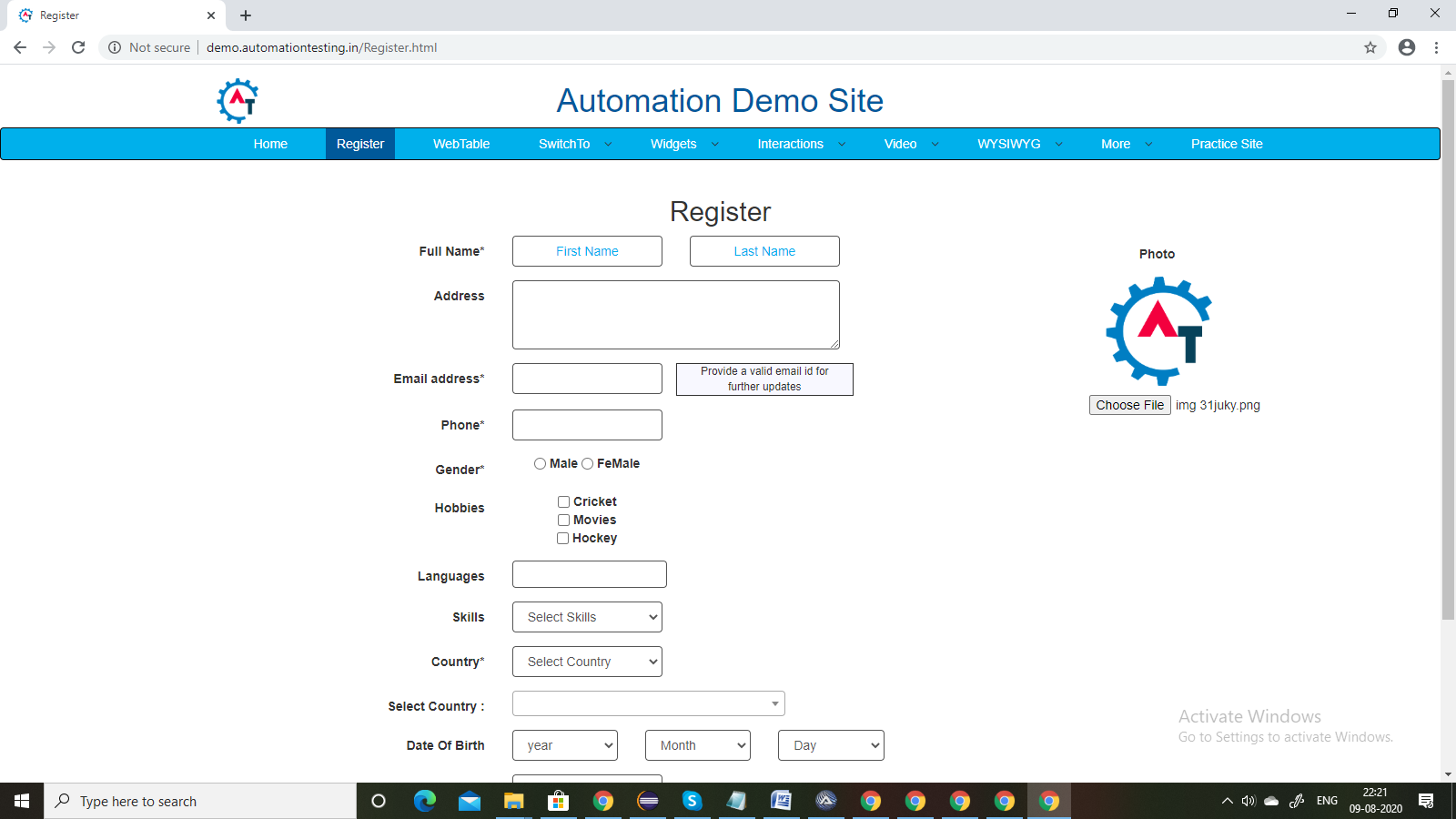
} **catch** (Exception e) {

e.printStackTrace();

}

}

}



Upload and control windows using selenium

public void UploadFileUsingJacobDll()

throws InterruptedException {

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

final String jacobdllarch = System.getProperty("sun.arch.data.model")

.contains("32") ? "jacob-1.18-x86.dll" : "jacob-1.18-x64.dll";

String jacobdllpath = workingDir + "\" + jacobdllarch;

File filejacob = new File(jacobdllpath);

System.setProperty(LibraryLoader.JACOB\_DLL\_PATH,

filejacob.getAbsolutePath());

AutoItX uploadWindow = new AutoItX();

driver = new FirefoxDriver();

String filepath = workingDir + "\SeleniumWebdriverUploadFile.html";

driver.get(filepath);

Thread.sleep(1000);

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

Thread.sleep(1000);

if (uploadWindow.winWaitActive("File Upload", "", 5)) {

if (uploadWindow.winExists("File Upload")) {

uploadWindow.sleep(100);

uploadWindow.send(filepath);

uploadWindow.controlClick("File Upload", "", "&Open");

}

}

}

Upload file using autoit:

**package** selenium;

**import** java.awt.Robot;

**import** java.awt.event.KeyEvent;

**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.chrome.ChromeOptions;

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

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

**public** **class** Dummy {

**public** **static** WebDriver *driver*;

**public** **static** **void** main(String args[])

{

**try** {

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

ChromeOptions options = **new** ChromeOptions();

options.addArguments("start-maximized");

options.setExperimentalOption("useAutomationExtension", **false**);

options.setExperimentalOption("excludeSwitches", **new** String[]{"enable-automation"});

*driver*= **new** ChromeDriver(options);

*driver*.get("https://demoqa.com/upload-download");

Thread.*sleep*(3000);

Thread.*sleep*(2000);

Robot robot = **new** Robot();

robot.keyPress(KeyEvent.***VK\_TAB***);

robot.keyRelease(KeyEvent.***VK\_TAB***);

Thread.*sleep*(1000);

robot.keyPress(KeyEvent.***VK\_TAB***);

robot.keyRelease(KeyEvent.***VK\_TAB***);

Thread.*sleep*(1000);

robot.keyPress(KeyEvent.***VK\_TAB***);

robot.keyRelease(KeyEvent.***VK\_TAB***);

Thread.*sleep*(1000);

robot.keyPress(KeyEvent.***VK\_SPACE***);

robot.keyRelease(KeyEvent.***VK\_SPACE***);

Thread.*sleep*(3000);

Process p = Runtime.*getRuntime*().exec("Upload\_file \_demoaut\_site.exe");

Thread.*sleep*(5000);

p.destroy();

Thread.*sleep*(1000);

*driver*.quit();

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

}

Output:

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 23986

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597331241.909][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 13, 2020 8:37:23 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Script:

WinActivate("Open")

Sleep(2000)

Send("C:\Users\lenovo\Desktop\selenium class videos\img 31juky.png")

Sleep(1000)

#Send("{ENTER}")

ControlClick("Open", "", "[CLASS:Button; INSTANCE:1]")



Scroll Option:

// Create instance of Javascript executor

JavascriptExecutor je = (JavascriptExecutor) driver;

//Identify the WebElement which will appear after scrolling down

WebElement element = driver.findElement(By.xpath(".//\*[@id='mCSB\_3\_container']/p[3]"));

// now execute query which actually will scroll until that element is not appeared on page.

je.executeScript("arguments[0].scrollIntoView(true);",element);

Maven project with selenium cucumber:

Testrunner.java

package SeleCucu.com.app.selecucu;

import org.junit.runner.RunWith;

import cucumber.api.CucumberOptions;

import cucumber.api.junit.Cucumber;

@RunWith(Cucumber.class)

@CucumberOptions(

features={"src/test/resources/features/test.feature"},

//tags={"@EPALSADDBLRECORDS"},

glue="SeleCucu.com.app.selecucu",

format = {"pretty", "html:target/Destination"},

plugin = {"com.cucumber.listener.ExtentCucumberFormatter:target/cucumber-reports/report.html"},

monochrome=true

)

public class testrunner {

}

Baseclass.java

package SeleCucu.com.app.selecucu;

import java.io.File;

import java.io.FileInputStream;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Base64;

import java.util.Calendar;

import java.util.List;

import java.util.Properties;

import java.util.Random;

import org.apache.commons.io.FileUtils;

import org.apache.commons.lang.RandomStringUtils;

import org.apache.log4j.BasicConfigurator;

import org.apache.log4j.Logger;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.firefox.FirefoxOptions;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.remote.DesiredCapabilities;

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

import com.codoid.products.exception.FilloException;

import com.codoid.products.fillo.Connection;

import com.codoid.products.fillo.Fillo;

import com.codoid.products.fillo.Recordset;

public class BaseClass {

public static WebDriver driver;

final static Logger logger = Logger.getLogger("BaseClass.class");

@SuppressWarnings("deprecation")

public static void launchBrowser() {

try {

BasicConfigurator.configure();

Properties prop = new Properties();

File file = new File("src/test/resources/Configuration/Config.properties");

FileInputStream fis = new FileInputStream(file);

prop.load(fis);

String browserType = prop.getProperty("browserType");

logger.info("browser for execution: "+browserType);

// logger is a system out this is to just write in the log file

if(browserType.equalsIgnoreCase("chrome")) {

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

ChromeOptions options = new ChromeOptions();

options.addArguments("--start-maximized");

options.setExperimentalOption("useAutomationExtension", false);

options.setExperimentalOption("excludeSwitches", new String[]{"enable-automation"});

options.addArguments("--disable-extensions");

driver=new ChromeDriver(options);

}

else if(browserType.equalsIgnoreCase("internetexplorer")) {

DesiredCapabilities ieCapabilities = DesiredCapabilities.internetExplorer();

ieCapabilities.setCapability(InternetExplorerDriver.INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS, true);

System.setProperty("webdriver.ie.driver", "IEDriverServer.exe");

driver = new InternetExplorerDriver(ieCapabilities);

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

}

else if(browserType.equals("firefox")) {

System.setProperty("webdriver.gecko.driver", "geckodriver.exe");

FirefoxOptions options = new FirefoxOptions();

options.addArguments("--start-maximized");

options.addArguments("disable-infobars");

driver=new FirefoxDriver(options);

}

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : openNewTab

\* Description : this method will open multiple tab based on url and position as input

\* Input Parameter : web element, driver object and position

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void openNewTab(WebDriver driver, String url, int position) {

try {

((JavascriptExecutor) driver).executeScript("window.open()");

ArrayList<String> tabs = new ArrayList<String>(driver.getWindowHandles());

System.out.println("tabs : " + tabs.size() + " >position: " + position + " >\t" + url);

driver.switchTo().window(tabs.get(position));

driver.get(url);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : setHighlight

\* Description : this method highlight UI element

\* Input Parameter : web element

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void setHighlight(WebElement element) {

try {

String attributevalue = "border:3px solid red;";

JavascriptExecutor executor = (JavascriptExecutor)driver;

String getattrib = element.getAttribute("style");

executor.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, attributevalue);

try {

Thread.sleep(250);

} catch (InterruptedException e) {

e.printStackTrace();

}

executor.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, getattrib);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : isObjectDisplayed

\* Description : this method will verify whether an UI element is displayed or not

\* Input Parameter : web element

\* Output Parameter : element display state

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public boolean isObjectDisplayed(WebElement element) {

int attempts = 0;

boolean isObjDisp = false;

while(attempts < 5) {

try {

Thread.sleep(500);

element.isDisplayed();

isObjDisp = true;

logger.info(element+ " is displayed as expected in application screen");

break;

}

catch (Exception e) {

e.printStackTrace();

}

attempts++;

}

return isObjDisp;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : waitObjEnabled

\* Description : this method will verify wait for an UI element enabled or not

\* Input Parameter : web element

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void waitObjEnabled(WebElement element) {

for(int i=0;i<5;i++){

try {

Thread.sleep(500);

boolean b = element.isEnabled();

if(b == true) {

//logger.info(element+ " is enabled in application screen");

}

break;

}

catch (Exception ex) {

logger.info(element+ " is not enabled in application screen");

}

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeClick

\* Description : this method will click an UI element

\* Input Parameter : web element

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeClick(WebElement element) {

try {

waitObjEnabled(element);

setHighlight(element);

element.click();

//logger.info("Performed click operation on element: " +element);

} catch (Exception ex) {

logger.info("Failed to click on element: " +element);

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeClearType

\* Description : this method will clears an UI element and types text

\* Input Parameter : web element, input text

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeClearType(WebElement element, String text) {

try {

waitObjEnabled(element);

setHighlight(element);

element.clear();

element.sendKeys(text);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeType

\* Description : this method will types text in an UI element

\* Input Parameter : web element, input text

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeType(WebElement element, String text) {

try {

waitObjEnabled(element);

setHighlight(element);

element.sendKeys(text);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeGetText

\* Description : this method will retrieves text from an UI element

\* Input Parameter : web element

\* Output Parameter : text

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String safeGetText(WebElement element) {

String sValue=null;

try {

waitObjEnabled(element);

setHighlight(element);

sValue = element.getText();

} catch (Exception e) {

e.printStackTrace();

}

return sValue;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeGetAttribute

\* Description : this method will return attribute type from an UI element

\* Input Parameter : web element, attribute

\* Output Parameter : text

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String safeGetAttribute(WebElement element, String attribute) {

String sValue=null;

try {

waitObjEnabled(element);

setHighlight(element);

sValue = element.getAttribute(attribute);

} catch (Exception e) {

e.printStackTrace();

}

return sValue;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeDoubleClick

\* Description : this method will perform double click on an UI element

\* Input Parameter : web element

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeDoubleClick(WebElement element) {

try {

waitObjEnabled(element);

setHighlight(element);

Actions useraction = new Actions(driver);

useraction.doubleClick(element).perform();

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : takeSnapshot

\* Description : this method will capture screenshot of a whole computer screen

\* Input Parameter : driver, file suffix

\* Output Parameter : screenshot will be placed in output folder

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void takeSnapshot(WebDriver driver, String fileSuffix) {

String currdatentimestamp = null;

try {

DateFormat dateFormat = new SimpleDateFormat("HH\_MM\_SS");

Calendar cal = Calendar.getInstance();

currdatentimestamp = dateFormat.format(cal.getTime());

TakesScreenshot scrshot = ((TakesScreenshot)driver);

String filepath = ".//src//test/resources//snapshots//"+fileSuffix;

File srcFile = scrshot.getScreenshotAs(OutputType.FILE);

String filename = filepath+"\_"+currdatentimestamp+".png";

logger.info("Screenshot destination: "+filename);

File destFile = new File(filename);

FileUtils.copyFile(srcFile, destFile);

logger.info("Screenshot captured");

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : checkBrowserState

\* Description : this method will select a text value from a list box

\* Input Parameter : web element, select value

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String checkBrowserState() {

String readyState = null;

try {

readyState = String.valueOf(((JavascriptExecutor) driver).executeScript("return document.readyState"));

return readyState;

} catch (Exception e) {

e.printStackTrace();

}

return readyState;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeSelectFromListBox

\* Description : this method will select a text value from a list box

\* Input Parameter : web element, select value

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeSelectFromListBox(WebElement element, String selectValue) {

try {

waitObjEnabled(element);

setHighlight(element);

Select select = new Select(element);

logger.info("Select value - '" + selectValue + "' from drop down list");

element.click();

select.selectByVisibleText(selectValue);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : getData

\* Description : this method will select and return text value from excel cell value

\* Input Parameter : sheet name, column name, test case ID

\* Output Parameter : cell value

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String getData(String sheetName, String columnName, String testCaseID) {

String data = null;

try {

Fillo fillo = new Fillo();

Connection connection = fillo.getConnection("src//test//resources//TestData//Testdata.xls");

String strQuery = "Select "+columnName+" from "+sheetName+" where TC\_ID='"+testCaseID+"'";

logger.info("Query for fetching records: "+strQuery);

Recordset recordset = connection.executeQuery(strQuery);

recordset.moveFirst();

data = recordset.getField(columnName);

//logger.info("Test data: " +columnName+": "+data);

}catch (FilloException fex) {

logger.info("Fillo exception occurred in execute query!");

fex.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

return data;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : setData

\* Description : this method will update text value in excel cell

\* Input Parameter : sheet name, column name, test case ID, input text

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void setData(String sheetName, String columnName, String testCaseID, String inputText) {

try {

Fillo fillo = new Fillo();

Connection connection = fillo.getConnection("src//test//resources//TestData//Testdata.xls");

String strQuery = "Update "+sheetName+" Set "+columnName+"='"+inputText+"' where TC\_ID='"+testCaseID+"'";

logger.info("Update query: "+strQuery);

connection.executeUpdate(strQuery);

}catch (FilloException fex) {

logger.info("Fillo exception occurred in update query!");

fex.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : encodeText

\* Description : this method will encode a text value and return

\* Input Parameter : string to be encoded

\* Output Parameter : encoded string

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String encodeText(String strToEncrypt) {

Base64.Encoder encoder = Base64.getUrlEncoder();

String eStr = encoder.encodeToString(strToEncrypt.getBytes());

logger.info("Encoded text: "+eStr);

return eStr;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : decodeText

\* Description : this method will decode a text value and return

\* Input Parameter : string to be decoded

\* Output Parameter : decoded string

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String decodeText(String strToDecrypt) {

Base64.Decoder decoder = Base64.getUrlDecoder();

String dStr = new String(decoder.decode(strToDecrypt));

return dStr;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : ListWebElementReturn

\* Description : this method will return list of web elements

\* Input Parameter : list string

\* Output Parameter : list of web elements

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static List<WebElement> ListWebElementReturn(String value) {

List<WebElement> Listvalue = driver.findElements(By.xpath(value));

return Listvalue;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeClickLink

\* Description : this method will perform click operation on a hyper link, linktext

\* Input Parameter : list string

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeClickLink(String linkText) {

try {

WebElement link = driver.findElement(By.linkText(linkText));

waitObjEnabled(link);

safeClick(link);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : generateRandomNumber

\* Description : this method will return a five digit random number each time it is invoked

\* Input Parameter :

\* Output Parameter : integer

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static int generateRandomNumber() {

try {

Random r = new Random( System.currentTimeMillis() );

return ((1 + r.nextInt(2)) \* 10000 + r.nextInt(10000));

} catch (Exception e) {

e.printStackTrace();

}

return 0;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : generateRandomText

\* Description : this method will return random text with user defined length

\* Input Parameter : integer

\* Output Parameter : string

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String generateRandomText(int count) {

String randomText = null;

try {

randomText = RandomStringUtils.randomAlphabetic(count).toLowerCase();

} catch (Exception e) {

e.printStackTrace();

}

return randomText;

}

}

Apptest.java

package SeleCucu.com.app.selecucu;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.PageFactory;

import com.app.objects.pageobjs;

import com.codoid.products.fillo.Connection;

import com.codoid.products.fillo.Fillo;

import com.codoid.products.fillo.Recordset;

import cucumber.api.java.en.Given;

import cucumber.api.java.en.Then;

import cucumber.api.java.en.When;

public class AppTest extends BaseClass {

@Given("^user opens browser$")

public void user\_opens\_browser() {

try {

launchBrowser();

} catch (Exception e) {

e.printStackTrace();

}

}

@Given("^navigates to application url$")

public void navigates\_to\_application\_url() {

try {

PageFactory.initElements(driver, pageobjs.class);

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

} catch (Exception e) {

e.printStackTrace();

}

}

@When("^user navigates to register screen$")

public void user\_navigates\_to\_register\_screen(){

try {

PageFactory.initElements(driver, pageobjs.class);

pageobjs.firstname.isDisplayed();

System.out.println("First name field is displayed");

} catch (Exception e) {

e.printStackTrace();

}

}

@Then("^user should be able to create a new user$")

public static void user\_should\_be\_able\_to\_create\_a\_new\_user() {

try {

PageFactory.initElements(driver, pageobjs.class);

Fillo fillo = new Fillo();

Connection connection = fillo.getConnection("src/test/resources/testdata/testData.xlsx");

String strQuery = "select \* from demoaut";

Recordset recordset = connection.executeQuery(strQuery);

int count = recordset.getCount();

System.out.println("Total number of rows: " +count);

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

// driver = new ChromeDriver();

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

//recordset.moveFirst();

for(int i = 0; i < count; i++) {

recordset.moveNext();

Thread.sleep(500);

String tcid = recordset.getField("TC\_ID");

String fname = recordset.getField("Firstname");

String lname = recordset.getField("Lastname");

String pno = recordset.getField("Phone");

String email = recordset.getField("Email");

String uname = recordset.getField("Username");

String pwd = recordset.getField("Password");

String cpwd = recordset.getField("Confirm");

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

pageobjs.firstname.sendKeys(fname);

safeClick(pageobjs.lastname);

pageobjs.lastname.sendKeys(lname);

pageobjs.phonenumber.sendKeys(pno);

pageobjs.email.sendKeys(email);

pageobjs.address.sendKeys("test street");

pageobjs.city.sendKeys("Chennai");

pageobjs.state.sendKeys("TamilNadu");

pageobjs.country.sendKeys("INDIA");

pageobjs.userName.sendKeys(uname);

pageobjs.password.sendKeys(pwd);

pageobjs.confirmPassword.sendKeys(cpwd);

pageobjs.submitbtn.click();

//System.out.println("Iteration: " +i);

System.out.println("TC\_ID: "+tcid+" Firstname: " +fname);

Thread.sleep(2000);

String actMsg = pageobjs.createduname.getText();

System.out.println(actMsg);

String add[] = actMsg.split(" ");

int lent = add.length;

System.out.println("Total number of words in success message: " +lent);

String unamefin = add[lent-1].replace(".", "");

System.out.println("Username created: " +unamefin);

if(actMsg.contains(uname)) {

System.out.println("User creation passed! The created Username is: "+uname);

}

else {

System.out.println("User creation failed!");

}

String updateQuery = "Update demoaut Set CreatedUser='"+unamefin+"' where TC\_ID='"+tcid+"'";

connection.executeUpdate(updateQuery);

}

Thread.sleep(5000);

driver.quit();

} catch (Exception e) {

e.printStackTrace();

}

}

}

Test.feature:

#Author: your.email@your.domain.com

#Keywords Summary :

#Feature: List of scenarios.

#Scenario: Business rule through list of steps with arguments.

#Given: Some precondition step

#When: Some key actions

#Then: To observe outcomes or validation

#And,But: To enumerate more Given,When,Then steps

#Scenario Outline: List of steps for data-driven as an Examples and <placeholder>

#Examples: Container for s table

#Background: List of steps run before each of the scenarios

#""" (Doc Strings)

#| (Data Tables)

#@ (Tags/Labels):To group Scenarios

#<> (placeholder)

#""

## (Comments)

#Sample Feature Definition Template

@tag

Feature: Verify login functionality of demo aut tours application

verify valid test data for login

Scenario: valid login credentials

Given user opens browser

And navigates to application url

When user navigates to register screen

Then user should be able to create a new user

Pageobjs.java

package com.app.objects;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.How;

public class pageobjs {

public pageobjs()

{

}

@FindBy(how=How.XPATH, using="//input[@id='userName']")

public static WebElement userNameField;

@FindBy(how=How.NAME, using="firstName")

public static WebElement firstname;

@FindBy(how=How.NAME, using="lastName")

public static WebElement lastname;

@FindBy(how=How.NAME, using="phone")

public static WebElement phonenumber;

@FindBy(how=How.ID, using="userName")

public static WebElement email;

@FindBy(how=How.NAME, using="address1")

public static WebElement address;

@FindBy(how=How.NAME, using="city")

public static WebElement city;

@FindBy(how=How.NAME, using="state")

public static WebElement state;

@FindBy(how=How.NAME, using="country")

public static WebElement country;

@FindBy(how=How.ID, using="email")

public static WebElement userName;

@FindBy(how=How.NAME, using="password")

public static WebElement password;

@FindBy(how=How.NAME, using="confirmPassword")

public static WebElement confirmPassword;

@FindBy(how=How.NAME, using="submit")

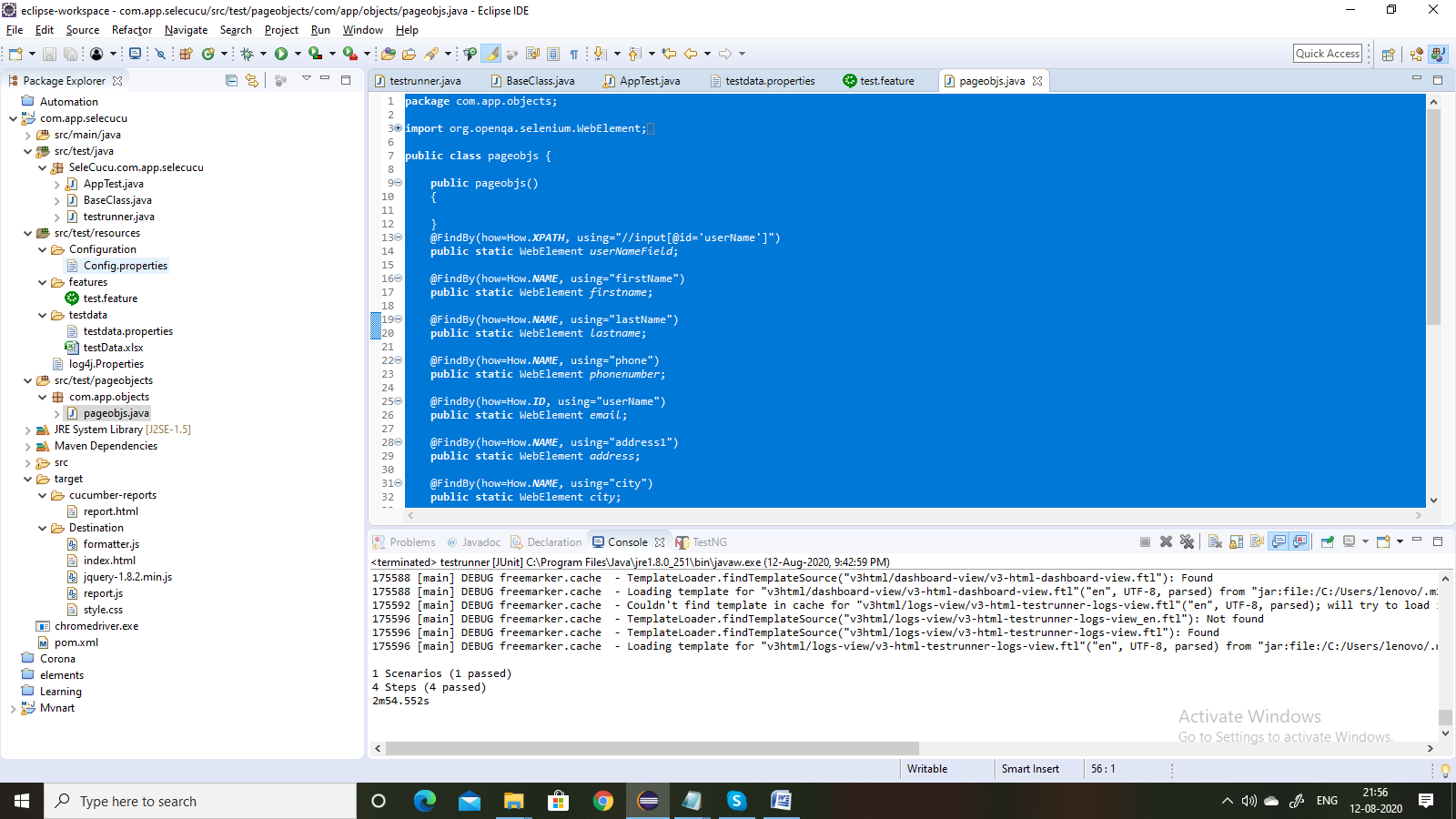
public static WebElement submitbtn;

@FindBy(how=How.XPATH, using="//a[.=' sign-in ']/following::b[1]")

public static WebElement createduname;

}

Hierarchy





Right click on testrunner and runa as Junit

Output:

#Author: your.email@your.domain.com

#Keywords Summary :

#Feature: List of scenarios.

#Scenario: Business rule through list of steps with arguments.

#Given: Some precondition step

#When: Some key actions

#Then: To observe outcomes or validation

#And,But: To enumerate more Given,When,Then steps

#Scenario Outline: List of steps for data-driven as an Examples and <placeholder>

#Examples: Container for s table

#Background: List of steps run before each of the scenarios

#""" (Doc Strings)

#| (Data Tables)

#@ (Tags/Labels):To group Scenarios

#<> (placeholder)

#""

## (Comments)

#Sample Feature Definition Template

@tag

Feature: Verify login functionality of demo aut tours application

verify valid test data for login

0 [main] INFO BaseClass.class - browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 35224

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597248790.898][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 12, 2020 9:43:13 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

First name field is displayed

Total number of rows: 15

TC\_ID: TC\_01 Firstname: DemoTest

Note: Your user name is DemoTest.

Total number of words in success message: 6

Username created: DemoTest

User creation passed! The created Username is: DemoTest

1 row(s) affected

TC\_ID: TC\_02 Firstname: Bala

Note: Your user name is Bala.

Total number of words in success message: 6

Username created: Bala

User creation passed! The created Username is: Bala

1 row(s) affected

TC\_ID: TC\_03 Firstname: Kalyan

Note: Your user name is Kalyan.

Total number of words in success message: 6

Username created: Kalyan

User creation passed! The created Username is: Kalyan

1 row(s) affected

TC\_ID: TC\_04 Firstname: Divya

Note: Your user name is Jink.

Total number of words in success message: 6

Username created: Jink

User creation passed! The created Username is: Jink

1 row(s) affected

TC\_ID: TC\_05 Firstname: Praveen

Note: Your user name is Tinkles.

Total number of words in success message: 6

Username created: Tinkles

User creation passed! The created Username is: Tinkles

1 row(s) affected

TC\_ID: TC\_06 Firstname: Hello

Note: Your user name is Balu.

Total number of words in success message: 6

Username created: Balu

User creation passed! The created Username is: Balu

1 row(s) affected

TC\_ID: TC\_07 Firstname: Kerala

Note: Your user name is Dravid.

Total number of words in success message: 6

Username created: Dravid

User creation passed! The created Username is: Dravid

1 row(s) affected

TC\_ID: TC\_08 Firstname: King

Note: Your user name is Hari.

Total number of words in success message: 6

Username created: Hari

User creation passed! The created Username is: Hari

1 row(s) affected

TC\_ID: TC\_09 Firstname: Guru

Note: Your user name is Ranjith.

Total number of words in success message: 6

Username created: Ranjith

User creation passed! The created Username is: Ranjith

1 row(s) affected

TC\_ID: TC\_10 Firstname: Retro

Note: Your user name is Guna.

Total number of words in success message: 6

Username created: Guna

User creation passed! The created Username is: Guna

1 row(s) affected

TC\_ID: TC\_11 Firstname: Save

Note: Your user name is Sekar.

Total number of words in success message: 6

Username created: Sekar

User creation passed! The created Username is: Sekar

1 row(s) affected

TC\_ID: TC\_12 Firstname: Control

Note: Your user name is Rajan.

Total number of words in success message: 6

Username created: Rajan

User creation passed! The created Username is: Rajan

1 row(s) affected

TC\_ID: TC\_13 Firstname: Firetest

Note: Your user name is Velu.

Total number of words in success message: 6

Username created: Velu

User creation passed! The created Username is: Velu

1 row(s) affected

TC\_ID: TC\_14 Firstname: Speed

Note: Your user name is Abinath.

Total number of words in success message: 6

Username created: Abinath

User creation passed! The created Username is: Abinath

1 row(s) affected

TC\_ID: TC\_15 Firstname: Thought

Note: Your user name is Freak.

Total number of words in success message: 6

Username created: Freak

User creation passed! The created Username is: Freak

1 row(s) affected

Scenario: valid login credentials # src/test/resources/features/test.feature:23

Given user opens browser # AppTest.user\_opens\_browser()

And navigates to application url # AppTest.navigates\_to\_application\_url()

When user navigates to register screen # AppTest.user\_navigates\_to\_register\_screen()

Then user should be able to create a new user # AppTest.user\_should\_be\_able\_to\_create\_a\_new\_user()

174576 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/v3-html-index.ftl"("en", UTF-8, parsed); will try to load it.

174584 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-index\_en.ftl"): Not found

174592 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-index.ftl"): Found

174592 [main] DEBUG freemarker.cache - Loading template for "v3html/v3-html-index.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/v3-html-index.ftl"

175363 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/v3-html-head.ftl"("en", UTF-8, parsed); will try to load it.

175367 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-head\_en.ftl"): Not found

175367 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-head.ftl"): Found

175367 [main] DEBUG freemarker.cache - Loading template for "v3html/v3-html-head.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/v3-html-head.ftl"

175375 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/v3-html-nav.ftl"("en", UTF-8, parsed); will try to load it.

175375 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-nav\_en.ftl"): Not found

175375 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-nav.ftl"): Found

175375 [main] DEBUG freemarker.cache - Loading template for "v3html/v3-html-nav.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/v3-html-nav.ftl"

175391 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/test-view/v3-html-test-view.ftl"("en", UTF-8, parsed); will try to load it.

175391 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-test-view\_en.ftl"): Not found

175391 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-test-view.ftl"): Found

175391 [main] DEBUG freemarker.cache - Loading template for "v3html/test-view/v3-html-test-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/test-view/v3-html-test-view.ftl"

175443 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/test-view/v3-html-test-view-charts.ftl"("en", UTF-8, parsed); will try to load it.

175443 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-test-view-charts\_en.ftl"): Not found

175443 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-test-view-charts.ftl"): Found

175443 [main] DEBUG freemarker.cache - Loading template for "v3html/test-view/v3-html-test-view-charts.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/test-view/v3-html-test-view-charts.ftl"

175479 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/test-view/v3-html-bdd.ftl"("en", UTF-8, parsed); will try to load it.

175483 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-bdd\_en.ftl"): Not found

175484 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-bdd.ftl"): Found

175484 [main] DEBUG freemarker.cache - Loading template for "v3html/test-view/v3-html-bdd.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/test-view/v3-html-bdd.ftl"

175564 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/category-view/v3-html-category-view.ftl"("en", UTF-8, parsed); will try to load it.

175564 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/category-view/v3-html-category-view\_en.ftl"): Not found

175564 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/category-view/v3-html-category-view.ftl"): Found

175564 [main] DEBUG freemarker.cache - Loading template for "v3html/category-view/v3-html-category-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/category-view/v3-html-category-view.ftl"

175572 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/exception-view/v3-html-exception-view.ftl"("en", UTF-8, parsed); will try to load it.

175572 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/exception-view/v3-html-exception-view\_en.ftl"): Not found

175572 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/exception-view/v3-html-exception-view.ftl"): Found

175572 [main] DEBUG freemarker.cache - Loading template for "v3html/exception-view/v3-html-exception-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/exception-view/v3-html-exception-view.ftl"

175576 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/author-view/v3-html-author-view.ftl"("en", UTF-8, parsed); will try to load it.

175580 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/author-view/v3-html-author-view\_en.ftl"): Not found

175580 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/author-view/v3-html-author-view.ftl"): Found

175580 [main] DEBUG freemarker.cache - Loading template for "v3html/author-view/v3-html-author-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/author-view/v3-html-author-view.ftl"

175584 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/dashboard-view/v3-html-dashboard-view.ftl"("en", UTF-8, parsed); will try to load it.

175588 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/dashboard-view/v3-html-dashboard-view\_en.ftl"): Not found

175588 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/dashboard-view/v3-html-dashboard-view.ftl"): Found

175588 [main] DEBUG freemarker.cache - Loading template for "v3html/dashboard-view/v3-html-dashboard-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/dashboard-view/v3-html-dashboard-view.ftl"

175592 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/logs-view/v3-html-testrunner-logs-view.ftl"("en", UTF-8, parsed); will try to load it.

175596 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/logs-view/v3-html-testrunner-logs-view\_en.ftl"): Not found

175596 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/logs-view/v3-html-testrunner-logs-view.ftl"): Found

175596 [main] DEBUG freemarker.cache - Loading template for "v3html/logs-view/v3-html-testrunner-logs-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/logs-view/v3-html-testrunner-logs-view.ftl"

1 Scenarios (1 passed)

4 Steps (4 passed)

2m54.552s

Scenario Outline:

Feature: Samplers

@sample

Scenario Outline: test multiple login credentials

Given user opens browser

And navigates to application url

Then user tries with multiple "<username>" and "<password>"

Examples:

| username | password |

| bala | test123 |

| sarojini | trest122 |

| neethi | testers |

@then method to be added to AppTest.

@Then("^user tries with multiple \"([^\"]\*)\" and \"([^\"]\*)\"$")

**public** **void** user\_tries\_with\_multiple\_and(String uname, String pword) {

**try** {

*driver*.get("http://demo.guru99.com/test/newtours/login.php");

*driver*.findElement(By.*name*("userName")).sendKeys(uname);

*driver*.findElement(By.*name*("password")).sendKeys(pword);

*driver*.findElement(By.*name*("submit")).click();

}

Output:

Feature: Samplers

@sample

Scenario Outline: test multiple login credentials # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:4

Given user opens browser

And navigates to application url

Then user tries with multiple "<username>" and "<password>"

Examples:

0 [main] INFO BaseClass.class - browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 7625

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597335558.281][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 13, 2020 9:49:20 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

@sample

Scenario Outline: test multiple login credentials # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:11

Given user opens browser # AppTest.user\_opens\_browser()

And navigates to application url # AppTest.navigates\_to\_application\_url()

Then user tries with multiple "bala" and "test123" # AppTest.user\_tries\_with\_multiple\_and(String,String)

42972 [main] INFO BaseClass.class - browser for execution: chrome

42972 [main] INFO BaseClass.class - browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 41879

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597335597.448][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 13, 2020 9:50:00 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

@sample

Scenario Outline: test multiple login credentials # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:12

Given user opens browser # AppTest.user\_opens\_browser()

And navigates to application url # AppTest.navigates\_to\_application\_url()

Then user tries with multiple "sarojini" and "trest122" # AppTest.user\_tries\_with\_multiple\_and(String,String)

75798 [main] INFO BaseClass.class - browser for execution: chrome

75798 [main] INFO BaseClass.class - browser for execution: chrome

75798 [main] INFO BaseClass.class - browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 44692

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597335631.473][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 13, 2020 9:50:34 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

@sample

Scenario Outline: test multiple login credentials # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:13

Given user opens browser # AppTest.user\_opens\_browser()

And navigates to application url # AppTest.navigates\_to\_application\_url()

Then user tries with multiple "neethi" and "testers" # AppTest.user\_tries\_with\_multiple\_and(String,String)

3 Scenarios (3 passed)

9 Steps (9 passed)

1m58.339s

Background:

Change the feature file and run

Feature: Verify login functionality of demo aut tours application

verify valid test data for login

Background: some precondition

Given user opens browser

Scenario: valid login credentials

And navigates to application url

When user navigates to register screen

Then user should be able to create a new user

Scenario: valid login credentials

And navigates to application url

Scenario: valid login credentials

And navigates to application url

Scenario: valid login credentials

And navigates to application url

Output:

Feature: Verify login functionality of demo aut tours application

verify valid test data for login

0 [main] INFO BaseClass.class - browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 6517

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597343400.500][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 14, 2020 12:00:03 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Background: some precondition # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:4

Given user opens browser # AppTest.user\_opens\_browser()

First name field is displayed

Total number of rows: 15

TC\_ID: TC\_01 Firstname: DemoTest

Note: Your user name is DemoTest.

Total number of words in success message: 6

Username created: DemoTest

User creation passed! The created Username is: DemoTest

1 row(s) affected

TC\_ID: TC\_02 Firstname: Bala

Note: Your user name is Bala.

Total number of words in success message: 6

Username created: Bala

User creation passed! The created Username is: Bala

1 row(s) affected

TC\_ID: TC\_03 Firstname: Kalyan

Note: Your user name is Kalyan.

Total number of words in success message: 6

Username created: Kalyan

User creation passed! The created Username is: Kalyan

1 row(s) affected

TC\_ID: TC\_04 Firstname: Divya

Note: Your user name is Jink.

Total number of words in success message: 6

Username created: Jink

User creation passed! The created Username is: Jink

1 row(s) affected

TC\_ID: TC\_05 Firstname: Praveen

Note: Your user name is Tinkles.

Total number of words in success message: 6

Username created: Tinkles

User creation passed! The created Username is: Tinkles

1 row(s) affected

TC\_ID: TC\_06 Firstname: Hello

Note: Your user name is Balu.

Total number of words in success message: 6

Username created: Balu

User creation passed! The created Username is: Balu

1 row(s) affected

TC\_ID: TC\_07 Firstname: Kerala

Note: Your user name is Dravid.

Total number of words in success message: 6

Username created: Dravid

User creation passed! The created Username is: Dravid

1 row(s) affected

TC\_ID: TC\_08 Firstname: King

Note: Your user name is Hari.

Total number of words in success message: 6

Username created: Hari

User creation passed! The created Username is: Hari

1 row(s) affected

TC\_ID: TC\_09 Firstname: Guru

Note: Your user name is Ranjith.

Total number of words in success message: 6

Username created: Ranjith

User creation passed! The created Username is: Ranjith

1 row(s) affected

TC\_ID: TC\_10 Firstname: Retro

Note: Your user name is Guna.

Total number of words in success message: 6

Username created: Guna

User creation passed! The created Username is: Guna

1 row(s) affected

TC\_ID: TC\_11 Firstname: Save

Note: Your user name is Sekar.

Total number of words in success message: 6

Username created: Sekar

User creation passed! The created Username is: Sekar

1 row(s) affected

TC\_ID: TC\_12 Firstname: Control

Note: Your user name is Rajan.

Total number of words in success message: 6

Username created: Rajan

User creation passed! The created Username is: Rajan

1 row(s) affected

TC\_ID: TC\_13 Firstname: Firetest

Note: Your user name is Velu.

Total number of words in success message: 6

Username created: Velu

User creation passed! The created Username is: Velu

1 row(s) affected

TC\_ID: TC\_14 Firstname: Speed

Note: Your user name is Abinath.

Total number of words in success message: 6

Username created: Abinath

User creation passed! The created Username is: Abinath

1 row(s) affected

TC\_ID: TC\_15 Firstname: Thought

Note: Your user name is Freak.

Total number of words in success message: 6

Username created: Freak

User creation passed! The created Username is: Freak

1 row(s) affected

Scenario: valid login credentials # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:7

And navigates to application url # AppTest.navigates\_to\_application\_url()

When user navigates to register screen # AppTest.user\_navigates\_to\_register\_screen()

Then user should be able to create a new user # AppTest.user\_should\_be\_able\_to\_create\_a\_new\_user()

174247 [main] INFO BaseClass.class - browser for execution: chrome

174247 [main] INFO BaseClass.class - browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 22054

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597343572.828][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 14, 2020 12:02:55 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Background: some precondition # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:4

Given user opens browser # AppTest.user\_opens\_browser()

Scenario: valid login credentials # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:12

And navigates to application url # AppTest.navigates\_to\_application\_url()

182451 [main] INFO BaseClass.class - browser for execution: chrome

182451 [main] INFO BaseClass.class - browser for execution: chrome

182451 [main] INFO BaseClass.class - browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 35500

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597343581.123][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 14, 2020 12:03:03 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Background: some precondition # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:4

Given user opens browser # AppTest.user\_opens\_browser()

Scenario: valid login credentials # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:15

And navigates to application url # AppTest.navigates\_to\_application\_url()

192622 [main] INFO BaseClass.class - browser for execution: chrome

192622 [main] INFO BaseClass.class - browser for execution: chrome

192622 [main] INFO BaseClass.class - browser for execution: chrome

192622 [main] INFO BaseClass.class - browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 44971

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597343592.262][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 14, 2020 12:03:15 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Background: some precondition # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:4

Given user opens browser # AppTest.user\_opens\_browser()

Scenario: valid login credentials # C:/Users/lenovo/eclipse-workspace/com.app.selecucu/src/test/resources/features/sampler.feature:18

And navigates to application url # AppTest.navigates\_to\_application\_url()

4 Scenarios (4 passed)

10 Steps (10 passed)

3m26.997s

@tag:

Testrunner.java:

package SeleCucu.com.app.selecucu;

import org.junit.runner.RunWith;

import cucumber.api.CucumberOptions;

import cucumber.api.junit.Cucumber;

@RunWith(Cucumber.class)

@CucumberOptions(

features={"src/test/resources/features/sampler.feature"},

tags={"@sample"},

glue="SeleCucu.com.app.selecucu",

format = {"pretty", "html:target/Destination"},

plugin = {"com.cucumber.listener.ExtentCucumberFormatter:target/cucumber-reports/report.html"},

monochrome=true

)

public class testrunner {

}

Sampler.feature:

Feature: Verify login functionality of demo aut tours application

verify valid test data for login

Background: some precondition

Given user opens browser

Scenario: valid login credentials

And navigates to application url

When user navigates to register screen

Then user should be able to create a new user

@sample

Scenario: valid login credentials

And navigates to application url

Scenario: valid login credentials

And navigates to application url

Scenario: valid login credentials

And navigates to application url

Output:

Feature: Verify login functionality of demo aut tours application

verify valid test data for login

0 [main] INFO BaseClass.class - browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 6688

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597344031.345][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 14, 2020 12:10:34 AM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Background: some precondition # src/test/resources/features/sampler.feature:4

Given user opens browser # AppTest.user\_opens\_browser()

@sample

Scenario: valid login credentials # src/test/resources/features/sampler.feature:12

And navigates to application url # AppTest.navigates\_to\_application\_url()

13114 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/v3-html-index.ftl"("en", UTF-8, parsed); will try to load it.

13118 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-index\_en.ftl"): Not found

13118 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-index.ftl"): Found

13122 [main] DEBUG freemarker.cache - Loading template for "v3html/v3-html-index.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/v3-html-index.ftl"

14292 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/v3-html-head.ftl"("en", UTF-8, parsed); will try to load it.

14296 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-head\_en.ftl"): Not found

14296 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-head.ftl"): Found

14296 [main] DEBUG freemarker.cache - Loading template for "v3html/v3-html-head.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/v3-html-head.ftl"

14300 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/v3-html-nav.ftl"("en", UTF-8, parsed); will try to load it.

14304 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-nav\_en.ftl"): Not found

14304 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/v3-html-nav.ftl"): Found

14304 [main] DEBUG freemarker.cache - Loading template for "v3html/v3-html-nav.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/v3-html-nav.ftl"

14320 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/test-view/v3-html-test-view.ftl"("en", UTF-8, parsed); will try to load it.

14320 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-test-view\_en.ftl"): Not found

14320 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-test-view.ftl"): Found

14320 [main] DEBUG freemarker.cache - Loading template for "v3html/test-view/v3-html-test-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/test-view/v3-html-test-view.ftl"

14430 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/test-view/v3-html-test-view-charts.ftl"("en", UTF-8, parsed); will try to load it.

14430 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-test-view-charts\_en.ftl"): Not found

14430 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-test-view-charts.ftl"): Found

14430 [main] DEBUG freemarker.cache - Loading template for "v3html/test-view/v3-html-test-view-charts.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/test-view/v3-html-test-view-charts.ftl"

14446 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/test-view/v3-html-bdd.ftl"("en", UTF-8, parsed); will try to load it.

14450 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-bdd\_en.ftl"): Not found

14450 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/test-view/v3-html-bdd.ftl"): Found

14450 [main] DEBUG freemarker.cache - Loading template for "v3html/test-view/v3-html-bdd.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/test-view/v3-html-bdd.ftl"

14593 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/category-view/v3-html-category-view.ftl"("en", UTF-8, parsed); will try to load it.

14601 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/category-view/v3-html-category-view\_en.ftl"): Not found

14601 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/category-view/v3-html-category-view.ftl"): Found

14601 [main] DEBUG freemarker.cache - Loading template for "v3html/category-view/v3-html-category-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/category-view/v3-html-category-view.ftl"

14621 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/exception-view/v3-html-exception-view.ftl"("en", UTF-8, parsed); will try to load it.

14625 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/exception-view/v3-html-exception-view\_en.ftl"): Not found

14625 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/exception-view/v3-html-exception-view.ftl"): Found

14625 [main] DEBUG freemarker.cache - Loading template for "v3html/exception-view/v3-html-exception-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/exception-view/v3-html-exception-view.ftl"

14633 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/author-view/v3-html-author-view.ftl"("en", UTF-8, parsed); will try to load it.

14637 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/author-view/v3-html-author-view\_en.ftl"): Not found

14637 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/author-view/v3-html-author-view.ftl"): Found

14637 [main] DEBUG freemarker.cache - Loading template for "v3html/author-view/v3-html-author-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/author-view/v3-html-author-view.ftl"

14665 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/dashboard-view/v3-html-dashboard-view.ftl"("en", UTF-8, parsed); will try to load it.

14665 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/dashboard-view/v3-html-dashboard-view\_en.ftl"): Not found

14665 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/dashboard-view/v3-html-dashboard-view.ftl"): Found

14665 [main] DEBUG freemarker.cache - Loading template for "v3html/dashboard-view/v3-html-dashboard-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/dashboard-view/v3-html-dashboard-view.ftl"

14688 [main] DEBUG freemarker.cache - Couldn't find template in cache for "v3html/logs-view/v3-html-testrunner-logs-view.ftl"("en", UTF-8, parsed); will try to load it.

14689 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/logs-view/v3-html-testrunner-logs-view\_en.ftl"): Not found

14691 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("v3html/logs-view/v3-html-testrunner-logs-view.ftl"): Found

14691 [main] DEBUG freemarker.cache - Loading template for "v3html/logs-view/v3-html-testrunner-logs-view.ftl"("en", UTF-8, parsed) from "jar:file:/C:/Users/lenovo/.m2/repository/com/aventstack/extentreports/4.0.8/extentreports-4.0.8.jar!/com/aventstack/extentreports/view/v3html/logs-view/v3-html-testrunner-logs-view.ftl"

1 Scenarios (1 passed)

2 Steps (2 passed)

0m13.060s

Maven project run as java application:

No feature file

Main method created

Right click on apptest and run as java application

AppTest.java:

**package** MavnPjt.Mvnart;

**import** org.openqa.selenium.support.PageFactory;

**import** com.app.objects.pageobjs;

**import** com.codoid.products.fillo.Connection;

**import** com.codoid.products.fillo.Fillo;

**import** com.codoid.products.fillo.Recordset;

**public** **class** AppTest **extends** BaseClass {

**public** **static** **void** main(String args[])

{

**try** {

*launch*();

*registration*();

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

**public** **static** **void** launch() {

**try** {

*launchBrowser*();

} **catch** (StackOverflowError e) {

e.printStackTrace();

}

}

**public** **static** **void** registration()

{

**try** {

PageFactory.*initElements*(*driver*, pageobjs.**class**);

Fillo fillo = **new** Fillo();

Connection connection = fillo.getConnection("src/test/resources/testdata/testData.xlsx");

String strQuery = "select \* from demoaut";

Recordset recordset = connection.executeQuery(strQuery);

**int** count = recordset.getCount();

System.***out***.println("Total number of rows: " +count);

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

// driver = new ChromeDriver();

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

//recordset.moveFirst();

**for**(**int** i = 0; i < count; i++) {

recordset.moveNext();

Thread.*sleep*(500);

String tcid = recordset.getField("TC\_ID");

String fname = recordset.getField("Firstname");

String lname = recordset.getField("Lastname");

String pno = recordset.getField("Phone");

String email = recordset.getField("Email");

String uname = recordset.getField("Username");

String pwd = recordset.getField("Password");

String cpwd = recordset.getField("Confirm");

*driver*.get("http://demo.guru99.com/test/newtours/register.php");

pageobjs.*firstname*.sendKeys(fname);

*safeClick*(pageobjs.*lastname*);

pageobjs.*lastname*.sendKeys(lname);

pageobjs.*phonenumber*.sendKeys(pno);

pageobjs.*email*.sendKeys(email);

pageobjs.*address*.sendKeys("test street");

pageobjs.*city*.sendKeys("Chennai");

pageobjs.*state*.sendKeys("TamilNadu");

pageobjs.*country*.sendKeys("INDIA");

pageobjs.*userName*.sendKeys(uname);

pageobjs.*password*.sendKeys(pwd);

pageobjs.*confirmPassword*.sendKeys(cpwd);

pageobjs.*submitbtn*.click();

//System.out.println("Iteration: " +i);

System.***out***.println("TC\_ID: "+tcid+" Firstname: " +fname);

Thread.*sleep*(2000);

String actMsg = pageobjs.*createduname*.getText();

System.***out***.println(actMsg);

String add[] = actMsg.split(" ");

**int** lent = add.length;

System.***out***.println("Total number of words in success message: " +lent);

String unamefin = add[lent-1].replace(".", "");

System.***out***.println("Username created: " +unamefin);

**if**(actMsg.contains(uname)) {

System.***out***.println("User creation passed! The created Username is: "+uname);

}

**else** {

System.***out***.println("User creation failed!");

}

String updateQuery = "Update demoaut Set CreatedUser='"+unamefin+"' where TC\_ID='"+tcid+"'";

connection.executeUpdate(updateQuery);

}

Thread.*sleep*(5000);

*driver*.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}



Output:

browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 24178

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597250769.762][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

Aug 12, 2020 10:16:13 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Total number of rows: 15

TC\_ID: TC\_01 Firstname: DemoTest

Note: Your user name is DemoTest.

Total number of words in success message: 6

Username created: DemoTest

User creation passed! The created Username is: DemoTest

1 row(s) affected

TC\_ID: TC\_02 Firstname: Bala

Note: Your user name is Bala.

Total number of words in success message: 6

Username created: Bala

User creation passed! The created Username is: Bala

1 row(s) affected

TC\_ID: TC\_03 Firstname: Kalyan

Note: Your user name is Kalyan.

Total number of words in success message: 6

Username created: Kalyan

User creation passed! The created Username is: Kalyan

1 row(s) affected

TC\_ID: TC\_04 Firstname: Divya

Note: Your user name is Jink.

Total number of words in success message: 6

Username created: Jink

User creation passed! The created Username is: Jink

1 row(s) affected

TC\_ID: TC\_05 Firstname: Praveen

Note: Your user name is Tinkles.

Total number of words in success message: 6

Username created: Tinkles

User creation passed! The created Username is: Tinkles

1 row(s) affected

TC\_ID: TC\_06 Firstname: Hello

Note: Your user name is Balu.

Total number of words in success message: 6

Username created: Balu

User creation passed! The created Username is: Balu

1 row(s) affected

TC\_ID: TC\_07 Firstname: Kerala

Note: Your user name is Dravid.

Total number of words in success message: 6

Username created: Dravid

User creation passed! The created Username is: Dravid

1 row(s) affected

TC\_ID: TC\_08 Firstname: King

Note: Your user name is Hari.

Total number of words in success message: 6

Username created: Hari

User creation passed! The created Username is: Hari

1 row(s) affected

TC\_ID: TC\_09 Firstname: Guru

Note: Your user name is Ranjith.

Total number of words in success message: 6

Username created: Ranjith

User creation passed! The created Username is: Ranjith

1 row(s) affected

TC\_ID: TC\_10 Firstname: Retro

Note: Your user name is Guna.

Total number of words in success message: 6

Username created: Guna

User creation passed! The created Username is: Guna

1 row(s) affected

TC\_ID: TC\_11 Firstname: Save

Note: Your user name is Sekar.

Total number of words in success message: 6

Username created: Sekar

User creation passed! The created Username is: Sekar

1 row(s) affected

TC\_ID: TC\_12 Firstname: Control

Note: Your user name is Rajan.

Total number of words in success message: 6

Username created: Rajan

User creation passed! The created Username is: Rajan

1 row(s) affected

TC\_ID: TC\_13 Firstname: Firetest

Note: Your user name is Velu.

Total number of words in success message: 6

Username created: Velu

User creation passed! The created Username is: Velu

1 row(s) affected

TC\_ID: TC\_14 Firstname: Speed

Note: Your user name is Abinath.

Total number of words in success message: 6

Username created: Abinath

User creation passed! The created Username is: Abinath

1 row(s) affected

TC\_ID: TC\_15 Firstname: Thought

Note: Your user name is Freak.

Total number of words in success message: 6

Username created: Freak

User creation passed! The created Username is: Freak

1 row(s) affected

Maven as testNG:

AppTest.java right click and run as testing

Apptest.java:

**package** com.app.testng.testng;

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

**import** org.openqa.selenium.support.PageFactory;

**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;

**import** com.app.objects.pageobjs;

**import** com.codoid.products.fillo.Connection;

**import** com.codoid.products.fillo.Fillo;

**import** com.codoid.products.fillo.Recordset;

/\*\*

\* Unit test for simple App.

\*/

**public** **class** AppTest **extends** BaseClass

{

@Test()

**public** **static** **void** test1() {

**try** {

PageFactory.*initElements*(*driver*, pageobjs.**class**);

Fillo fillo = **new** Fillo();

Connection connection = fillo.getConnection("src/test/resources/testdata/testData.xlsx");

String strQuery = "select \* from demoaut";

Recordset recordset = connection.executeQuery(strQuery);

**int** count = recordset.getCount();

System.***out***.println("Total number of rows: " +count);

**for**(**int** i = 0; i < count; i++) {

recordset.moveNext();

Thread.*sleep*(500);

String tcid = recordset.getField("TC\_ID");

String fname = recordset.getField("Firstname");

String lname = recordset.getField("Lastname");

String pno = recordset.getField("Phone");

String email = recordset.getField("Email");

String uname = recordset.getField("Username");

String pwd = recordset.getField("Password");

String cpwd = recordset.getField("Confirm");

*driver*.get("http://demo.guru99.com/test/newtours/register.php");

pageobjs.*firstname*.sendKeys(fname);

*safeClick*(pageobjs.*lastname*);

pageobjs.*lastname*.sendKeys(lname);

pageobjs.*phonenumber*.sendKeys(pno);

pageobjs.*email*.sendKeys(email);

pageobjs.*address*.sendKeys("test street");

pageobjs.*city*.sendKeys("Chennai");

pageobjs.*state*.sendKeys("TamilNadu");

pageobjs.*country*.sendKeys("INDIA");

pageobjs.*userName*.sendKeys(uname);

pageobjs.*password*.sendKeys(pwd);

pageobjs.*confirmPassword*.sendKeys(cpwd);

pageobjs.*submitbtn*.click();

//System.out.println("Iteration: " +i);

System.***out***.println("TC\_ID: "+tcid+" Firstname: " +fname);

Thread.*sleep*(2000);

String actMsg = pageobjs.*createduname*.getText();

System.***out***.println(actMsg);

String add[] = actMsg.split(" ");

**int** lent = add.length;

System.***out***.println("Total number of words in success message: " +lent);

String unamefin = add[lent-1].replace(".", "");

System.***out***.println("Username created: " +unamefin);

**if**(actMsg.contains(uname)) {

System.***out***.println("User creation passed! The created Username is: "+uname);

}

**else** {

System.***out***.println("User creation failed!");

}

String updateQuery = "Update demoaut Set CreatedUser='"+unamefin+"' where TC\_ID='"+tcid+"'";

connection.executeUpdate(updateQuery);

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterSuite

**public** **void** aftSuite() {

**try** {

System.***out***.println("After suite!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeClass

**public** **void** befClass() {

**try** {

System.***out***.println("Before Class!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterClass

**public** **void** aftClass() {

**try** {

System.***out***.println("After Class!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeMethod

**public** **void** befMet() {

**try** {

System.***out***.println("Before Method!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterMethod

**public** **void** aftMet() {

**try** {

System.***out***.println("After Method!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeTest

**public** **void** befTest() {

**try** {

System.***out***.println("Before before test!");

} **catch** (Exception e) {

e.printStackTrace();

}

}

@BeforeSuite

**public** **void** befSuite() {

**try** {

*launchBrowser*();

} **catch** (Exception e) {

e.printStackTrace();

}

}

@AfterTest

**public** **void** aftTest() {

**try** {

Thread.*sleep*(5000);

//driver.quit();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Testng.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

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

<suite name=*"Suite"*>

<parameter name=*"username"* value=*"mango"* />

<parameter name=*"password"* value=*"orange"* />

<test thread-count=*"5"* name=*"Test"*>

<classes>

<class name=*"com.app.testng.testng.AppTest"*>

<methods>

<include name=*"test1"* />

</methods>

</class>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->



Output:

[RemoteTestNG] detected TestNG version 6.9.10

[TestNGContentHandler] [WARN] It is strongly recommended to add "<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >" at the top of your file, otherwise TestNG may fail or not work as expected.

[TestNG] Running:

C:\Users\lenovo\eclipse-workspace\testng\testng.xml

browser for execution: chrome

Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d844400483861342c20cd-refs/branch-heads/4103@{#416}) on port 29570

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1597256357.008][WARNING]: This version of ChromeDriver has not been tested with Chrome version 84.

[1597256362.071][WARNING]: Timed out connecting to Chrome, retrying...

Aug 12, 2020 11:49:26 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Before before test!

Before Class!

Before Method!

Total number of rows: 15

TC\_ID: TC\_01 Firstname: DemoTest

Note: Your user name is DemoTest.

Total number of words in success message: 6

Username created: DemoTest

User creation passed! The created Username is: DemoTest

1 row(s) affected

TC\_ID: TC\_02 Firstname: Bala

Note: Your user name is Bala.

Total number of words in success message: 6

Username created: Bala

User creation passed! The created Username is: Bala

1 row(s) affected

TC\_ID: TC\_03 Firstname: Kalyan

Note: Your user name is Kalyan.

Total number of words in success message: 6

Username created: Kalyan

User creation passed! The created Username is: Kalyan

1 row(s) affected

TC\_ID: TC\_04 Firstname: Divya

Note: Your user name is Jink.

Total number of words in success message: 6

Username created: Jink

User creation passed! The created Username is: Jink

1 row(s) affected

TC\_ID: TC\_05 Firstname: Praveen

Note: Your user name is Tinkles.

Total number of words in success message: 6

Username created: Tinkles

User creation passed! The created Username is: Tinkles

1 row(s) affected

TC\_ID: TC\_06 Firstname: Hello

Note: Your user name is Balu.

Total number of words in success message: 6

Username created: Balu

User creation passed! The created Username is: Balu

1 row(s) affected

TC\_ID: TC\_07 Firstname: Kerala

Note: Your user name is Dravid.

Total number of words in success message: 6

Username created: Dravid

User creation passed! The created Username is: Dravid

1 row(s) affected

TC\_ID: TC\_08 Firstname: King

Note: Your user name is Hari.

Total number of words in success message: 6

Username created: Hari

User creation passed! The created Username is: Hari

1 row(s) affected

TC\_ID: TC\_09 Firstname: Guru

Note: Your user name is Ranjith.

Total number of words in success message: 6

Username created: Ranjith

User creation passed! The created Username is: Ranjith

1 row(s) affected

TC\_ID: TC\_10 Firstname: Retro

Note: Your user name is Guna.

Total number of words in success message: 6

Username created: Guna

User creation passed! The created Username is: Guna

1 row(s) affected

TC\_ID: TC\_11 Firstname: Save

Note: Your user name is Sekar.

Total number of words in success message: 6

Username created: Sekar

User creation passed! The created Username is: Sekar

1 row(s) affected

TC\_ID: TC\_12 Firstname: Control

Note: Your user name is Rajan.

Total number of words in success message: 6

Username created: Rajan

User creation passed! The created Username is: Rajan

1 row(s) affected

TC\_ID: TC\_13 Firstname: Firetest

Note: Your user name is Velu.

Total number of words in success message: 6

Username created: Velu

User creation passed! The created Username is: Velu

1 row(s) affected

TC\_ID: TC\_14 Firstname: Speed

Note: Your user name is Abinath.

Total number of words in success message: 6

Username created: Abinath

User creation passed! The created Username is: Abinath

1 row(s) affected

TC\_ID: TC\_15 Firstname: Thought

Note: Your user name is Freak.

Total number of words in success message: 6

Username created: Freak

User creation passed! The created Username is: Freak

1 row(s) affected

After Method!

After Class!

After suite!

===============================================

Suite

Total tests run: 1, Failures: 0, Skips: 0

===============================================

Mouse coordinates finder:

Demoqa slider progressbar sorter

double mouseX = MouseInfo.getPointerInfo().getLocation().getX();

double mouseY = MouseInfo.getPointerInfo().getLocation().getY();

System.out.println("X:" + mouseX);

System.out.println("Y:" + mouseY);

Resizable using coordinated to X and Y offset:

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.chrome.ChromeOptions;

import org.openqa.selenium.interactions.Actions;

public class DemoQA {

public static WebDriver driver;

public static void main(String args[]) {

try {

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

ChromeOptions options = new ChromeOptions();

options.addArguments("start-maximized");

options.addArguments("disable-features=NetworkService");

options.setExperimentalOption("useAutomationExtension", false);

options.setExperimentalOption("excludeSwitches", new String[]{"enable-automation"});

driver = new ChromeDriver(options);

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

Thread.sleep(2000);

Actions act = new Actions(driver);

WebElement source = driver.findElement(By.xpath("//div[@id='resizableBoxWithRestriction']/div/following::span[1]"));

act.moveToElement(source).click().build().perform();

Thread.sleep(1000);

act.dragAndDropBy(source, 398, 298).build().perform();

Thread.sleep(2000);

driver.quit();

} catch (Exception e) {

e.printStackTrace();

}

}

}

Additional user defined functions:

Encode decode safe click link text

Get data

package SeleCucu.com.app.selecucu;

import java.io.File;

import java.io.FileInputStream;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Base64;

import java.util.Calendar;

import java.util.List;

import java.util.Properties;

import java.util.Random;

import org.apache.commons.io.FileUtils;

import org.apache.commons.lang.RandomStringUtils;

import org.apache.log4j.BasicConfigurator;

import org.apache.log4j.Logger;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.firefox.FirefoxOptions;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.remote.DesiredCapabilities;

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

import com.codoid.products.exception.FilloException;

import com.codoid.products.fillo.Connection;

import com.codoid.products.fillo.Fillo;

import com.codoid.products.fillo.Recordset;

public class BaseClass {

public static WebDriver driver;

final static Logger logger = Logger.getLogger("BaseClass.class");

@SuppressWarnings("deprecation")

public static void launchBrowser() {

try {

BasicConfigurator.configure();

Properties prop = new Properties();

File file = new File("src/test/resources/Configuration/Config.properties");

FileInputStream fis = new FileInputStream(file);

prop.load(fis);

String browserType = prop.getProperty("browserType");

logger.info("browser for execution: "+browserType);

// logger is a system out this is to just write in the log file

if(browserType.equalsIgnoreCase("chrome")) {

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

ChromeOptions options = new ChromeOptions();

options.addArguments("--start-maximized");

options.setExperimentalOption("useAutomationExtension", false);

options.setExperimentalOption("excludeSwitches", new String[]{"enable-automation"});

options.addArguments("--disable-extensions");

driver=new ChromeDriver(options);

}

else if(browserType.equalsIgnoreCase("internetexplorer")) {

DesiredCapabilities ieCapabilities = DesiredCapabilities.internetExplorer();

ieCapabilities.setCapability(InternetExplorerDriver.INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS, true);

System.setProperty("webdriver.ie.driver", "IEDriverServer.exe");

driver = new InternetExplorerDriver(ieCapabilities);

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

}

else if(browserType.equals("firefox")) {

System.setProperty("webdriver.gecko.driver", "geckodriver.exe");

FirefoxOptions options = new FirefoxOptions();

options.addArguments("--start-maximized");

options.addArguments("disable-infobars");

driver=new FirefoxDriver(options);

}

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : openNewTab

\* Description : this method will open multiple tab based on url and position as input

\* Input Parameter : web element, driver object and position

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void openNewTab(WebDriver driver, String url, int position) {

try {

((JavascriptExecutor) driver).executeScript("window.open()");

ArrayList<String> tabs = new ArrayList<String>(driver.getWindowHandles());

System.out.println("tabs : " + tabs.size() + " >position: " + position + " >\t" + url);

driver.switchTo().window(tabs.get(position));

driver.get(url);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : setHighlight

\* Description : this method highlight UI element

\* Input Parameter : web element

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void setHighlight(WebElement element) {

try {

String attributevalue = "border:3px solid red;";

JavascriptExecutor executor = (JavascriptExecutor)driver;

String getattrib = element.getAttribute("style");

executor.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, attributevalue);

try {

Thread.sleep(250);

} catch (InterruptedException e) {

e.printStackTrace();

}

executor.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, getattrib);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : isObjectDisplayed

\* Description : this method will verify whether an UI element is displayed or not

\* Input Parameter : web element

\* Output Parameter : element display state

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public boolean isObjectDisplayed(WebElement element) {

int attempts = 0;

boolean isObjDisp = false;

while(attempts < 5) {

try {

Thread.sleep(500);

element.isDisplayed();

isObjDisp = true;

logger.info(element+ " is displayed as expected in application screen");

break;

}

catch (Exception e) {

e.printStackTrace();

}

attempts++;

}

return isObjDisp;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : waitObjEnabled

\* Description : this method will verify wait for an UI element enabled or not

\* Input Parameter : web element

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void waitObjEnabled(WebElement element) {

for(int i=0;i<5;i++){

try {

Thread.sleep(500);

boolean b = element.isEnabled();

if(b == true) {

//logger.info(element+ " is enabled in application screen");

}

break;

}

catch (Exception ex) {

logger.info(element+ " is not enabled in application screen");

}

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeClick

\* Description : this method will click an UI element

\* Input Parameter : web element

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeClick(WebElement element) {

try {

waitObjEnabled(element);

setHighlight(element);

element.click();

//logger.info("Performed click operation on element: " +element);

} catch (Exception ex) {

logger.info("Failed to click on element: " +element);

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeClearType

\* Description : this method will clears an UI element and types text

\* Input Parameter : web element, input text

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeClearType(WebElement element, String text) {

try {

waitObjEnabled(element);

setHighlight(element);

element.clear();

element.sendKeys(text);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeType

\* Description : this method will types text in an UI element

\* Input Parameter : web element, input text

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeType(WebElement element, String text) {

try {

waitObjEnabled(element);

setHighlight(element);

element.sendKeys(text);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeGetText

\* Description : this method will retrieves text from an UI element

\* Input Parameter : web element

\* Output Parameter : text

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String safeGetText(WebElement element) {

String sValue=null;

try {

waitObjEnabled(element);

setHighlight(element);

sValue = element.getText();

} catch (Exception e) {

e.printStackTrace();

}

return sValue;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeGetAttribute

\* Description : this method will return attribute type from an UI element

\* Input Parameter : web element, attribute

\* Output Parameter : text

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String safeGetAttribute(WebElement element, String attribute) {

String sValue=null;

try {

waitObjEnabled(element);

setHighlight(element);

sValue = element.getAttribute(attribute);

} catch (Exception e) {

e.printStackTrace();

}

return sValue;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeDoubleClick

\* Description : this method will perform double click on an UI element

\* Input Parameter : web element

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeDoubleClick(WebElement element) {

try {

waitObjEnabled(element);

setHighlight(element);

Actions useraction = new Actions(driver);

useraction.doubleClick(element).perform();

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : takeSnapshot

\* Description : this method will capture screenshot of a whole computer screen

\* Input Parameter : driver, file suffix

\* Output Parameter : screenshot will be placed in output folder

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void takeSnapshot(WebDriver driver, String fileSuffix) {

String currdatentimestamp = null;

try {

DateFormat dateFormat = new SimpleDateFormat("HH\_MM\_SS");

Calendar cal = Calendar.getInstance();

currdatentimestamp = dateFormat.format(cal.getTime());

TakesScreenshot scrshot = ((TakesScreenshot)driver);

String filepath = ".//src//test/resources//snapshots//"+fileSuffix;

File srcFile = scrshot.getScreenshotAs(OutputType.FILE);

String filename = filepath+"\_"+currdatentimestamp+".png";

logger.info("Screenshot destination: "+filename);

File destFile = new File(filename);

FileUtils.copyFile(srcFile, destFile);

logger.info("Screenshot captured");

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : checkBrowserState

\* Description : this method will select a text value from a list box

\* Input Parameter : web element, select value

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String checkBrowserState() {

String readyState = null;

try {

readyState = String.valueOf(((JavascriptExecutor) driver).executeScript("return document.readyState"));

return readyState;

} catch (Exception e) {

e.printStackTrace();

}

return readyState;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeSelectFromListBox

\* Description : this method will select a text value from a list box

\* Input Parameter : web element, select value

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeSelectFromListBox(WebElement element, String selectValue) {

try {

waitObjEnabled(element);

setHighlight(element);

Select select = new Select(element);

logger.info("Select value - '" + selectValue + "' from drop down list");

element.click();

select.selectByVisibleText(selectValue);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : getData

\* Description : this method will select and return text value from excel cell value

\* Input Parameter : sheet name, column name, test case ID

\* Output Parameter : cell value

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String getData(String sheetName, String columnName, String testCaseID) {

String data = null;

try {

Fillo fillo = new Fillo();

Connection connection = fillo.getConnection("src//test//resources//TestData//Testdata.xls");

String strQuery = "Select "+columnName+" from "+sheetName+" where TC\_ID='"+testCaseID+"'";

logger.info("Query for fetching records: "+strQuery);

Recordset recordset = connection.executeQuery(strQuery);

recordset.moveFirst();

data = recordset.getField(columnName);

//logger.info("Test data: " +columnName+": "+data);

}catch (FilloException fex) {

logger.info("Fillo exception occurred in execute query!");

fex.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

return data;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : setData

\* Description : this method will update text value in excel cell

\* Input Parameter : sheet name, column name, test case ID, input text

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void setData(String sheetName, String columnName, String testCaseID, String inputText) {

try {

Fillo fillo = new Fillo();

Connection connection = fillo.getConnection("src//test//resources//TestData//Testdata.xls");

String strQuery = "Update "+sheetName+" Set "+columnName+"='"+inputText+"' where TC\_ID='"+testCaseID+"'";

logger.info("Update query: "+strQuery);

connection.executeUpdate(strQuery);

}catch (FilloException fex) {

logger.info("Fillo exception occurred in update query!");

fex.printStackTrace();

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : encodeText

\* Description : this method will encode a text value and return

\* Input Parameter : string to be encoded

\* Output Parameter : encoded string

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String encodeText(String strToEncrypt) {

Base64.Encoder encoder = Base64.getUrlEncoder();

String eStr = encoder.encodeToString(strToEncrypt.getBytes());

logger.info("Encoded text: "+eStr);

return eStr;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : decodeText

\* Description : this method will decode a text value and return

\* Input Parameter : string to be decoded

\* Output Parameter : decoded string

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String decodeText(String strToDecrypt) {

Base64.Decoder decoder = Base64.getUrlDecoder();

String dStr = new String(decoder.decode(strToDecrypt));

return dStr;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : ListWebElementReturn

\* Description : this method will return list of web elements

\* Input Parameter : list string

\* Output Parameter : list of web elements

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static List<WebElement> ListWebElementReturn(String value) {

List<WebElement> Listvalue = driver.findElements(By.xpath(value));

return Listvalue;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : safeClickLink

\* Description : this method will perform click operation on a hyper link, linktext

\* Input Parameter : list string

\* Output Parameter :

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static void safeClickLink(String linkText) {

try {

WebElement link = driver.findElement(By.linkText(linkText));

waitObjEnabled(link);

safeClick(link);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : generateRandomNumber

\* Description : this method will return a five digit random number each time it is invoked

\* Input Parameter :

\* Output Parameter : integer

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static int generateRandomNumber() {

try {

Random r = new Random( System.currentTimeMillis() );

return ((1 + r.nextInt(2)) \* 10000 + r.nextInt(10000));

} catch (Exception e) {

e.printStackTrace();

}

return 0;

}

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Method Name : generateRandomText

\* Description : this method will return random text with user defined length

\* Input Parameter : integer

\* Output Parameter : string

\* Author : Automation

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

public static String generateRandomText(int count) {

String randomText = null;

try {

randomText = RandomStringUtils.randomAlphabetic(count).toLowerCase();

} catch (Exception e) {

e.printStackTrace();

}

return randomText;

}

}

Keyword driven Framework:

**package** testkey.com.keyword.sample;

**import** java.awt.AWTException;

**import** java.awt.Robot;

**import** java.awt.event.KeyEvent;

**import** java.util.concurrent.TimeUnit;

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

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

**import** org.openqa.selenium.support.PageFactory;

**import** com.codoid.products.fillo.Connection;

**import** com.codoid.products.fillo.Fillo;

**import** com.codoid.products.fillo.Recordset;

**public** **class** AppTest **extends** BaseClass

{

**public** **static** **void** LaunchBrowser()

{

System.***out***.println("Launching browser");

*launchBrowser*("chrome");

}

**public** **static** **void** OpenURL()

{

**try** {

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

} **catch** (Exception e) {

e.printStackTrace();

}

}

**public** **static** **void** ClickSearch()

{

**try** {

*driver*.findElement(By.*name*("q")).sendKeys("Selenium");

Robot robot=**new** Robot();

robot.keyPress(KeyEvent.***VK\_ENTER***);

robot.keyRelease(KeyEvent.***VK\_ENTER***);

} **catch** (AWTException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**public** **static** **void** ClickLink()

{

*driver*.findElement(By.*xpath*("//cite[.=\"www.selenium.dev\"]")).click();

}

**public** **static** **void** gettitle()

{

String title = *driver*.getTitle();

System.***out***.println("The current page title is :"+title);

}

**public** **static** **void** CloseB()

{

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

*driver*.quit();

}

**public** **static** **void** main(String args[])

{

**try** {

Fillo fillo = **new** Fillo();

Connection connection = fillo.getConnection("src/test/resources/testdata/keyword.xlsx");

String strQuery = "select \* from KDF";

Recordset recordset = connection.executeQuery(strQuery);

System.***out***.println("main method");

**int** count = recordset.getCount();

//recordset.moveFirst();

System.***out***.println("The number of lines in excel is :"+count);

**for**(**int** i = 0; i < count; i++) {

recordset.moveNext();

Thread.*sleep*(500);

String kword = recordset.getField("Keyword");

System.***out***.println(kword);

System.***out***.println("Excel accessed");

**if**(kword.equalsIgnoreCase("OpenURL"))

{

System.***out***.println("open URL");

*OpenURL*();

}

**else** **if**(kword.equalsIgnoreCase("ClickSearch"))

{

*ClickSearch*();

}

**else** **if**(kword.equalsIgnoreCase("ClickLink"))

{

*ClickLink*();

}

**else** **if**(kword.equalsIgnoreCase("gettitle"))

{

*gettitle*();

}

**else** **if**(kword.equalsIgnoreCase("LaunchBrowser"))

{

*LaunchBrowser*();

}

**else** **if**(kword.equalsIgnoreCase("CloseB"))

{

*CloseB*();

System.***out***.println("Closing the browser");

}

}

}

**catch** (Exception e) {

e.printStackTrace();

}

}

}



Assertequals

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

String ActualTitle = driver.getTitle();

String ExpectedTitle = “Amazon.com: Online Shopping for Electronics, Apparel,

    Computers, Books, DVDs & more”;

Assert.assertEquals(ActualTitle, ExpectedTitle);

System.out.println(“Assert passed”);

Scroll:

public class ScrollByPixel {

WebDriver driver;

@Test

public void ByPixel() {

System.setProperty("webdriver.chrome.driver", "E://Selenium//Selenium\_Jars//chromedriver.exe");

driver = new ChromeDriver();

JavascriptExecutor js = (JavascriptExecutor) driver;

// Launch the application

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

//To maximize the window. This code may not work with Selenium 3 jars. If script fails you can remove the line below

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

// This will scroll down the page by 1000 pixel vertical

js.executeScript("window.scrollBy(0,1000)");

}

}

Assert greater:

Assert.assertTrue(prev > curr );

Assert sorting:

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

List<WebElement> elementList= driver.findElements(By.xpath(YourLocator));

for(webElement we:elementList){

obtainedList.add(we.getText);

}

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

for(String s:obtainedList){

sortedList.add(s);

}

Collections.sort(sortedList);

Assert.assertTrue(sortedList.equals(obtainedList));

Caseinsensitive order

Assert greater than:

String result = selenium.getText("css=span.rcnt");

System.out.println("Search result is: " + result);

int foo = Integer.parseInt(result);

assertThat("Pass", foo, greaterThan(1000));

Webtable sorting rowwise:

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

List<WebElement> elementList= driver.findElements(By.xpath(YourLocator));

for(webElement we:elementList){

obtainedList.add(we.getText);

}

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

for(String s:obtainedList){

sortedList.add(s);

}

Collections.sort(sortedList);

Assert.assertTrue(sortedList.equals(obtainedList));

For descending order add:

Collections.reverse(sortedList);

after Collections.sort(sortedList);

Right click and open in new tab:

action.contextClick(element).sendKeys(Keys.ARROW\_DOWN).sendKeys(Keys.ARROW\_DOWN).sendKeys(Keys.ENTER).build().perform();

assert equals fail:

**private** **static** **void** assertEqualsImpl(Object actual, Object expected,

String message) {

**if**((expected == null) && (actual == null)) {

**return**;

}

**if**(expected == null ^ actual == null) {

[**failNotEquals**](https://www.codota.com/code/java/methods/org.testng.Assert/failNotEquals)(actual, expected, message);

}

**if** (expected.equals(actual) && actual.equals(expected)) {

**return**;

}

[**failNotEquals**](https://www.codota.com/code/java/methods/org.testng.Assert/failNotEquals)(actual, expected, message);

}

Get color code in hex:

String bgrclr = holly.getCssValue("background-color");

System.***out***.println("The Background color is : "+bgrclr);

String hex = Color.*fromString*(bgrclr).asHex();

System.***out***.println(hex);

@dataprovider testing using fillo:

package com.app.testng.testng;

import java.util.ArrayList;

import org.openqa.selenium.support.PageFactory;

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.DataProvider;

import org.testng.annotations.Test;

import com.app.objects.pageobjs;

import com.codoid.products.fillo.Connection;

import com.codoid.products.fillo.Fillo;

import com.codoid.products.fillo.Recordset;

/\*\*

\* Unit test for simple App.

\*/

public class AppTest extends BaseClass

{

@Test(dataProvider="testdata")

public static void test1(String tcid,String firstname, String lastname, String phone, String email,String uname, String pword, String confirm) {

try {

PageFactory.initElements(driver, pageobjs.class);

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

safeType(pageobjs.firstname, firstname);

safeClick(pageobjs.lastname);

pageobjs.lastname.sendKeys(lastname);

pageobjs.phonenumber.sendKeys(phone);

pageobjs.email.sendKeys(email);

pageobjs.address.sendKeys("test street");

pageobjs.city.sendKeys("Chennai");

pageobjs.state.sendKeys("TamilNadu");

pageobjs.country.sendKeys("INDIA");

pageobjs.userName.sendKeys(uname);

pageobjs.password.sendKeys(pword);

pageobjs.confirmPassword.sendKeys(confirm);

pageobjs.submitbtn.click();

System.out.println("TC\_ID: "+tcid+" Firstname: " +firstname);

Thread.sleep(2000);

String actMsg = pageobjs.createduname.getText();

System.out.println(actMsg);

String add[] = actMsg.split(" ");

int lent = add.length;

System.out.println("Total number of words in success message: " +lent);

String unamefin = add[lent-1].replace(".", "");

System.out.println("Username created: " +unamefin);

if(actMsg.contains(uname)) {

System.out.println("User creation passed! The created Username is: "+uname);

}

else {

System.out.println("User creation failed!");

}

setData("demoaut", "CreatedUser", tcid, uname);

} catch (Exception e) {

e.printStackTrace();

}

}

@AfterSuite

public void aftSuite() {

try {

System.out.println("After suite!");

} catch (Exception e) {

e.printStackTrace();

}

}

@BeforeClass

public void befClass() {

try {

System.out.println("Before Class!");

} catch (Exception e) {

e.printStackTrace();

}

}

@AfterClass

public void aftClass() {

try {

System.out.println("After Class!");

} catch (Exception e) {

e.printStackTrace();

}

}

@BeforeMethod

public void befMet() {

try {

System.out.println("Before Method!");

} catch (Exception e) {

e.printStackTrace();

}

}

@AfterMethod

public void aftMet() {

try {

System.out.println("After Method!");

} catch (Exception e) {

e.printStackTrace();

}

}

@BeforeTest

public void befTest() {

try {

System.out.println("Before before test!");

} catch (Exception e) {

e.printStackTrace();

}

}

@BeforeSuite

public void befSuite() {

try {

LaunchBrowser("chrome");

} catch (Exception e) {

e.printStackTrace();

}

}

@AfterTest

public void aftTest() {

try {

Thread.sleep(5000);

//driver.quit();

} catch (Exception e) {

e.printStackTrace();

}

}

@DataProvider(name="testdata")

public Object[][] TestDataFeed(){

Fillo fillo = new Fillo();

Connection connection;

try {

connection = fillo.getConnection("src/test/resources/testdata/testData.xlsx");

String strQuery = "select \* from demoaut";

String[][] testData =null;

int count = getRowCount("demoaut");

testData = new String[count][8];

System.out.println("Total number of rows: " +count);

Recordset recordset = connection.executeQuery(strQuery);

for (int i = 0; i < count; i++) {

recordset.moveNext();

Thread.sleep(500);

testData[i][0] = recordset.getField("TC\_ID");

testData[i][1] = recordset.getField("Firstname");

testData[i][2] = recordset.getField("Lastname");

testData[i][3] = recordset.getField("Phone");

testData[i][4] = recordset.getField("Email");

testData[i][5] = recordset.getField("Username");

testData[i][6] = recordset.getField("Password");

testData[i][7] = recordset.getField("Confirm");

}

System.out.println(testData);

return (testData);

}

catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

Output:

[RemoteTestNG] detected TestNG version 6.9.10

[TestNGContentHandler] [WARN] It is strongly recommended to add "<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >" at the top of your file, otherwise TestNG may fail or not work as expected.

[TestNG] Running:

C:\Users\Neethi\AppData\Local\Temp\testng-eclipse-1272802646\testng-customsuite.xml

In Launch Browser method!!

Starting ChromeDriver 85.0.4183.87 (cd6713ebf92fa1cacc0f1a598df280093af0c5d7-refs/branch-heads/4183@{#1689}) on port 27201

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1602496802.193][WARNING]: Deprecated chrome option is ignored: useAutomationExtension

[1602496802.193][WARNING]: Deprecated chrome option is ignored: useAutomationExtension

Oct 12, 2020 3:30:05 PM org.openqa.selenium.remote.ProtocolHandshake createSession

INFO: Detected dialect: W3C

Before before test!

Before Class!

Total number of rows: 15

[[Ljava.lang.String;@21ec5d87

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_01 Firstname: DemoTest

Note: Your user name is DemoTest.

Total number of words in success message: 6

Username created: DemoTest

User creation passed! The created Username is: DemoTest

Update query: Update demoaut Set CreatedUser='DemoTest' where TC\_ID='TC\_01'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_02 Firstname: Bala

Note: Your user name is Bala.

Total number of words in success message: 6

Username created: Bala

User creation passed! The created Username is: Bala

Update query: Update demoaut Set CreatedUser='Bala' where TC\_ID='TC\_02'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_03 Firstname: Kalyan

Note: Your user name is Kalyan.

Total number of words in success message: 6

Username created: Kalyan

User creation passed! The created Username is: Kalyan

Update query: Update demoaut Set CreatedUser='Kalyan' where TC\_ID='TC\_03'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_04 Firstname: Divya

Note: Your user name is Jink.

Total number of words in success message: 6

Username created: Jink

User creation passed! The created Username is: Jink

Update query: Update demoaut Set CreatedUser='Jink' where TC\_ID='TC\_04'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_05 Firstname: Praveen

Note: Your user name is Tinkles.

Total number of words in success message: 6

Username created: Tinkles

User creation passed! The created Username is: Tinkles

Update query: Update demoaut Set CreatedUser='Tinkles' where TC\_ID='TC\_05'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_06 Firstname: Hello

Note: Your user name is Balu.

Total number of words in success message: 6

Username created: Balu

User creation passed! The created Username is: Balu

Update query: Update demoaut Set CreatedUser='Balu' where TC\_ID='TC\_06'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_07 Firstname: Kerala

Note: Your user name is Dravid.

Total number of words in success message: 6

Username created: Dravid

User creation passed! The created Username is: Dravid

Update query: Update demoaut Set CreatedUser='Dravid' where TC\_ID='TC\_07'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_08 Firstname: King

Note: Your user name is Hari.

Total number of words in success message: 6

Username created: Hari

User creation passed! The created Username is: Hari

Update query: Update demoaut Set CreatedUser='Hari' where TC\_ID='TC\_08'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_09 Firstname: Guru

Note: Your user name is Ranjith.

Total number of words in success message: 6

Username created: Ranjith

User creation passed! The created Username is: Ranjith

Update query: Update demoaut Set CreatedUser='Ranjith' where TC\_ID='TC\_09'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_10 Firstname: Retro

Note: Your user name is Guna.

Total number of words in success message: 6

Username created: Guna

User creation passed! The created Username is: Guna

Update query: Update demoaut Set CreatedUser='Guna' where TC\_ID='TC\_10'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_11 Firstname: Save

Note: Your user name is Sekar.

Total number of words in success message: 6

Username created: Sekar

User creation passed! The created Username is: Sekar

Update query: Update demoaut Set CreatedUser='Sekar' where TC\_ID='TC\_11'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_12 Firstname: Control

Note: Your user name is Rajan.

Total number of words in success message: 6

Username created: Rajan

User creation passed! The created Username is: Rajan

Update query: Update demoaut Set CreatedUser='Rajan' where TC\_ID='TC\_12'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_13 Firstname: Firetest

Note: Your user name is Velu.

Total number of words in success message: 6

Username created: Velu

User creation passed! The created Username is: Velu

Update query: Update demoaut Set CreatedUser='Velu' where TC\_ID='TC\_13'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_14 Firstname: Speed

Note: Your user name is Abinath.

Total number of words in success message: 6

Username created: Abinath

User creation passed! The created Username is: Abinath

Update query: Update demoaut Set CreatedUser='Abinath' where TC\_ID='TC\_14'

1 row(s) affected

After Method!

Before Method!

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: firstName] is enabled in application screen

[[ChromeDriver: chrome on WINDOWS (443e10f8a80f991247cfe1acfac0aeb6)] -> name: lastName] is enabled in application screen

TC\_ID: TC\_15 Firstname: Thought

Note: Your user name is Freak.

Total number of words in success message: 6

Username created: Freak

User creation passed! The created Username is: Freak

Update query: Update demoaut Set CreatedUser='Freak' where TC\_ID='TC\_15'

1 row(s) affected

After Method!

After Class!

PASSED: test1("TC\_01", "DemoTest", "K", "983736678", "dwjhdshjd@gmail.com", "DemoTest", "test123", "test123")

PASSED: test1("TC\_02", "Bala", "A", "237327778", "dwjhdshjd@gmail.com", "Bala", "test123", "test123")

PASSED: test1("TC\_03", "Kalyan", "S", "983736678", "dwjhdshjd@gmail.com", "Kalyan", "test123", "test123")

PASSED: test1("TC\_04", "Divya", "D", "237327778", "dwjhdshjd@gmail.com", "Jink", "test123", "test123")

PASSED: test1("TC\_05", "Praveen", "F", "983736678", "dwjhdshjd@gmail.com", "Tinkles", "test123", "test123")

PASSED: test1("TC\_06", "Hello", "G", "237327778", "dwjhdshjd@gmail.com", "Balu", "test123", "test123")

PASSED: test1("TC\_07", "Kerala", "H", "983736678", "dwjhdshjd@gmail.com", "Dravid", "test123", "test123")

PASSED: test1("TC\_08", "King", "D", "237327778", "dwjhdshjd@gmail.com", "Hari", "test123", "test123")

PASSED: test1("TC\_09", "Guru", "R", "983736678", "dwjhdshjd@gmail.com", "Ranjith", "test123", "test123")

PASSED: test1("TC\_10", "Retro", "T", "237327778", "dwjhdshjd@gmail.com", "Guna", "test123", "test123")

PASSED: test1("TC\_11", "Save", "S", "983736678", "dwjhdshjd@gmail.com", "Sekar", "test123", "test123")

PASSED: test1("TC\_12", "Control", "Y", "983736678", "dwjhdshjd@gmail.com", "Rajan", "test123", "test123")

PASSED: test1("TC\_13", "Firetest", "D", "2387327887", "testbbk@gmail.com", "Velu", "test123", "test123")

PASSED: test1("TC\_14", "Speed", "F", "9872738787", "testbbk@gmail.com", "Abinath", "test123", "test123")

PASSED: test1("TC\_15", "Thought", "G", "237837889", "testbbk@gmail.com", "Freak", "test123", "test123")

===============================================

Default test

Tests run: 15, Failures: 0, Skips: 0

===============================================

After suite!

===============================================

Default suite

Total tests run: 15, Failures: 0, Skips: 0

===============================================

[TestNG] Time taken by org.testng.reporters.SuiteHTMLReporter@3901d134: 144 ms

[TestNG] Time taken by org.testng.reporters.jq.Main@4c1d9d4b: 138 ms

[TestNG] Time taken by org.testng.reporters.XMLReporter@4278a03f: 41 ms

[TestNG] Time taken by org.testng.reporters.JUnitReportReporter@2ddc8ecb: 18 ms

[TestNG] Time taken by [FailedReporter passed=0 failed=0 skipped=0]: 26 ms

[TestNG] Time taken by org.testng.reporters.EmailableReporter2@1fe20588: 22 ms