Automation

package webdriver.methods;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example1

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver path

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//Java wait

Thread.sleep(3000);

//To launch the URL

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

//Java wait

Thread.sleep(3000);

//To close the current tab of the browser

//driver.close();

//To close all the opened tabs of the browser

//driver.quit();

//To maximize the browser window

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

//Java wait

Thread.sleep(3000);

//To minimize the browser window

driver.manage().window().minimize();

//Java wait

Thread.sleep(3000);

//To get the current URL

String currentURL = driver.getCurrentUrl();

System.out.println("Current URL: " + currentURL);

//Java wait

Thread.sleep(3000);

//To get the title of the browser

String getTitle=driver.getTitle();

System.out.println("Get title : " + getTitle);

//Java wait

Thread.sleep(3000);

//To close all the opened tabs of the browser

driver.quit();

}

}

<https://youtu.be/PBx3Y6mPU-g?si=LDTh90ptmCYyhMba>

<https://stackoverflow.com/questions/48897286/chrome-webdriver-cant-be-resolved-to-a-type-error-eclipse-and-java>

Those who are facing an issue while importing WebDriver and ChromDriver. please go through this link and check

package webdriver.methods;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.Point;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example2

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver path

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL --> Alternative of Get method

driver.navigate().to("https://www.flipkart.com/");

// //Java wait

// Thread.sleep(3000);

//

// //To navigate back

// driver.navigate().back();

// //Java wait

// Thread.sleep(3000);

//

// //To navigate forward

// driver.navigate().forward();

//

// //Java wait

// Thread.sleep(3000);

//

// //To refresh the webpage

// driver.navigate().refresh();

//Java wait

Thread.sleep(3000);

//Set size of the browser window --> which is height and width in pixels

// Dimension d=new Dimension(300, 200);

// driver.manage().window().setSize(d);

//Get size of the browser window

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

// System.out.println("Dimension in pixels : " + getSize);

//Set position of the browser window --> Which is in the form of x and y coordinates

// Point p=new Point(200, 300);

// driver.manage().window().setPosition(p);

//Get position of the browser window

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

System.out.println("Get position : " + getPosition);

}

}

X-Path

1. X-path by attribute

//tagname [@attributeName='attributeValue']

2. X-path by text

//tagname[text()='text value']

3. X-path by contains using attribute

//tagname[contains (@attributeName, 'attributeValue')]

4. X-path by contains using text

//tagname[contains(text(), 'textValue')]

package locator.types;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Xpathbyattribute

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//To enter the UN

driver.findElement(By.xpath("//input[@type='text']")).sendKeys("Sophan@1234");

Thread.sleep(3000);

//To enter PWD

driver.findElement(By.xpath("//input[@type='password']")).sendKeys("Test@123");

Thread.sleep(3000);

//To click on the Login Btn

driver.findElement(By.xpath("//button[@value='1']")).click();

Thread.sleep(3000);

driver.close();

}

}

package locator.types;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Xpathbytext

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//To click on the Forgotten PWD link

driver.findElement(By.xpath("//a[text()='Forgotten password?']")).click();

Thread.sleep(3000);

driver.close();

}

}

package locator.types;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Xpathbycontainsusingattribute

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//To enter the UN

driver.findElement(By.xpath("//input[contains (@aria-label, 'Email address')]")).sendKeys("Sophan@1234");

Thread.sleep(3000);

//To enter PWD

driver.findElement(By.xpath("//input[contains (@type,'pass')]")).sendKeys("Test@123");

Thread.sleep(3000);

//To click on the Login Btn

driver.findElement(By.xpath("//button[contains(@class, '\_42ft')]")).click();

Thread.sleep(3000);

driver.close();

}

}

package locator.types;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Xpathbycontainsusingtext

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//To click on the Forgotten PWD link

driver.findElement(By.xpath("//a[contains(text(), 'Forgotte')]")).click();

Thread.sleep(3000);

driver.close();

}

}

package locator.types;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Tagname

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/WebpageTagName.html");

Thread.sleep(3000);

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

Thread.sleep(3000);

//To enter the UN

driver.findElement(By.tagName("input")).sendKeys("Sophan@1234");

Thread.sleep(3000);

//To enter PWD

driver.findElement(By.tagName("input")).sendKeys("Test@123");

}

}

package locator.types;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class IDLocator

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/IDLocatorType.html");

Thread.sleep(3000);

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

Thread.sleep(3000);

//To enter the UN

driver.findElement(By.id("1234")).sendKeys("Sophan@1234");

Thread.sleep(3000);

//To enter PWD

driver.findElement(By.id("ABC")).sendKeys("Test@123");

Thread.sleep(3000);

//To enter Contact

driver.findElement(By.id("XYZ")).sendKeys("12345678");

}

}

package locator.types;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class ClassnameLocator

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/classNameLocator.html");

Thread.sleep(3000);

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

Thread.sleep(3000);

//To enter the UN

driver.findElement(By.className("ABC")).sendKeys("Sophan@1234");

Thread.sleep(3000);

//To enter PWD

driver.findElement(By.className("XYZ")).sendKeys("Test@123");

Thread.sleep(3000);

//To enter Contact

driver.findElement(By.className("PQR")).sendKeys("12345678");

}

}

package webelement.methods;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example1

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

Thread.sleep(3000);

//To enter UN

//driver.findElement(By.xpath("//input[@type='text']")).sendKeys("Test@1234");

Thread.sleep(3000);

//To clear already entered text

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

//To verify whether that field/button is enabled or not. will return value in terms of true or false

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

System.out.println("Is UN field enabled : " + enabled);

if(enabled == true)

{

System.out.println("The UN field is enabled");

}

else

{

System.out.println("The UN field is disabled");

}

}

}

package webelement.methods;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example2

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

Thread.sleep(3000);

boolean displayed = driver.findElement(By.xpath("//a[text()='Create new account']")).isDisplayed();

System.out.println("Is the Create New Account displayed ? : " + displayed);

if(displayed == false)

{

System.out.println("Create New Account is displayed");

}

else

{

System.out.println(" Create New Account is not displayed");

}

Thread.sleep(3000);

driver.close();

}

}

package webelement.methods;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example3

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//To click on the Create New Account button

driver.findElement(By.xpath("//a[text()='Create new account']")).click();

Thread.sleep(3000);

boolean selected = driver.findElement(By.xpath("(//input[@type='radio'])[2]")).isSelected();

System.out.println("Is Male radio button selected ? : " + selected );

if(selected == true)

{

System.out.println("Male radio button is selected");

}

else

{

System.out.println("Male radio button is not selected");

//To click on the male radio button

driver.findElement(By.xpath("(//input[@type='radio'])[2]")).click();

Thread.sleep(3000);

boolean selected2 = driver.findElement(By.xpath("(//input[@type='radio'])[2]")).isSelected();

if(selected2 == true)

{

System.out.println(" Male radio button is selected ");

}

else

{

System.out.println("Male radio button is not selected");

}

}

Thread.sleep(3000);

//driver.close();

}

}

package webelement.methods;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example4

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

Thread.sleep(3000);

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

Thread.sleep(3000);

String actualFacebookText = driver.findElement(By.xpath("//h2[contains(text(), 'F')]")).getText();

System.out.println("Text of the Facebook : " + actualFacebookText );

String expectedFacebookText= "Facebook helps you connect and share with the people in your lie";

if(expectedFacebookText.equals(actualFacebookText))

{

System.out.println("The test case is passed. The expected and actual output are in sync");

}

else

{

System.out.println("The The test case is failed. The expected and actual output are not in sync");

}

Thread.sleep(3000);

driver.close();

}

}

package dropdown.handling;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class Example1

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_year']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

//Step no 03 --> To select the specific value from the dropdown

s.selectByVisibleText("2025");

// s.selectByIndex(1);

// s.selectByValue("9");

}

}

Dropdown

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class GetSizeOfTheDropdown

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_month']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

//Step no 03 --> To get the size of the dropdown

List<WebElement> allOptions = s.getOptions();

int sizeofTheDropdown = allOptions.size();

System.out.println("Size of the month dropdown : " + sizeofTheDropdown);

Thread.sleep(3000);

driver.close();

}

}

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class PrintallOptionFromTheDropdown

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_day']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

//To print the options from the dropdown

List<WebElement> allOptions = s.getOptions();

//System.out.print(allOptions);

//For each loop

for(WebElement option: allOptions)

{

String optionText = option.getText();

System.out.println(optionText + " ");

}

driver.close();

}

}

package dropdown.handling;

import java.util.Iterator;

import java.util.List;

import java.util.TreeSet;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class PrintoptionsinAscendingADescendingOrder

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_month']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

//To print the options from the dropdown

List<WebElement> allOptions = s.getOptions();

//System.out.print(allOptions);

//TreeSet object creation

TreeSet<String > tr = new TreeSet<String>();

for( WebElement option : allOptions)

{

String optionText = option.getText();

tr.add(optionText);

}

System.out.println("------Print all options in an Ascending order---------");

for( String t : tr)

{

System.out.println(t);

}

System.out.println("--------Print all option in the Descending Order----------");

Iterator<String> des = tr.descendingIterator();

while(des.hasNext())

{

System.out.println(des.next());

}

driver.close();

}

}

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class IsMultiSelectableOrNot

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_month']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

boolean multiple = s.isMultiple();

if(multiple == true)

{

System.out.println("Your dropdown is multi-selectable");

}

else

{

System.out.println("Your drodpown is not multi-selectable");

}

driver.close();

}

}

package dropdown.handling;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class IsMultiSelectableOrNot2

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Multiselectabledropdown.html");

Thread.sleep(3000);

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

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement countryDropdown = driver.findElement(By.xpath("//select [@multiple='true']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(countryDropdown);

Thread.sleep(3000);

boolean multiple = s.isMultiple();

if(multiple == true)

{

System.out.println("Your dropdown is multi-selectable");

}

else

{

System.out.println("Your drodpown is not multi-selectable");

}

driver.close();

}

}

package dropdown.handling;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class Example1

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_year']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

//Step no 03 --> To select the specific value from the dropdown

s.selectByVisibleText("2025");

// s.selectByIndex(1);

// s.selectByValue("9");

}

}

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class IsMultiSelectableOrNot

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_month']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

boolean multiple = s.isMultiple();

if(multiple == true)

{

System.out.println("Your dropdown is multi-selectable");

}

else

{

System.out.println("Your drodpown is not multi-selectable");

}

driver.close();

}

}

package dropdown.handling;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class IsMultiSelectableOrNot2

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Multiselectabledropdown.html");

Thread.sleep(3000);

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

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement countryDropdown = driver.findElement(By.xpath("//select [@multiple='true']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(countryDropdown);

Thread.sleep(3000);

boolean multiple = s.isMultiple();

if(multiple == true)

{

System.out.println("Your dropdown is multi-selectable");

}

else

{

System.out.println("Your drodpown is not multi-selectable");

}

driver.close();

}

}

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class PrintallOptionFromTheDropdown

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_day']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

//To print the options from the dropdown

List<WebElement> allOptions = s.getOptions();

//System.out.print(allOptions);

//For each loop

for(WebElement option: allOptions)

{

String optionText = option.getText();

System.out.println(optionText + " ");

}

driver.close();

}

}

package dropdown.handling;

import java.util.Iterator;

import java.util.List;

import java.util.TreeSet;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class PrintoptionsinAscendingADescendingOrder

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_month']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

//To print the options from the dropdown

List<WebElement> allOptions = s.getOptions();

//System.out.print(allOptions);

//TreeSet object creation

TreeSet<String > tr = new TreeSet<String>();

for( WebElement option : allOptions)

{

String optionText = option.getText();

tr.add(optionText);

}

System.out.println("------Print all options in an Ascending order---------");

for( String t : tr)

{

System.out.println(t);

}

System.out.println("--------Print all option in the Descending Order----------");

Iterator<String> des = tr.descendingIterator();

while(des.hasNext())

{

System.out.println(des.next());

}

driver.close();

}

}

package dropdown.handling;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class GetFirstSelectedOption

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_month']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

WebElement firstSelectedOption = s.getFirstSelectedOption();

System.out.println(firstSelectedOption.getText());

Thread.sleep(3000);

driver.close();

}

}

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class GetSizeOfTheDropdown

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_month']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

//Step no 03 --> To get the size of the dropdown

List<WebElement> allOptions = s.getOptions();

int sizeofTheDropdown = allOptions.size();

System.out.println("Size of the month dropdown : " + sizeofTheDropdown);

Thread.sleep(3000);

driver.close();

}

}

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class GetAllSelectedOptions

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Multiselectabledropdown.html");

Thread.sleep(3000);

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

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement selectCountryDropdown = driver.findElement(By.xpath("//select[@multiple='true']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(selectCountryDropdown);

Thread.sleep(3000);

s.selectByVisibleText("India");

s.selectByVisibleText("China");

Thread.sleep(3000);

List<WebElement> allSelectedOptions = s.getAllSelectedOptions();

for( WebElement options: allSelectedOptions)

{

String options2 = options.getText();

System.out.println(options2);

}

}

}

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class GetAllSelctedOptionsSingleSelectable

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_month']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(dateofBirthDropdown);

Thread.sleep(3000);

s.selectByVisibleText("Nov");

s.selectByVisibleText("Oct");

Thread.sleep(3000);

List<WebElement> allSelectedOptions = s.getAllSelectedOptions();

for( WebElement option1 : allSelectedOptions)

{

String option2 = option1.getText();

System.out.println(option2 +" ");

}

Thread.sleep(3000);

driver.close();

}

}

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class Deselect

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Multiselectabledropdown.html");

Thread.sleep(3000);

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

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement selectCountryDropdown = driver.findElement(By.xpath("//select[@multiple='true']"));

Thread.sleep(3000);

//Step no 02 --> To create an object of the Select class. which accepts dropdown arguments

Select s=new Select(selectCountryDropdown);

Thread.sleep(3000);

//Step no 03 --> Selecting the India and China options from the dropdown

s.selectByVisibleText("India");

s.selectByVisibleText("China");

Thread.sleep(3000);

//s.deselectByVisibleText("China");

s.deselectAll();

Thread.sleep(3000);

driver.close();

}

}

Screenshot

package screenshot.capturing;

import java.io.File;

import java.io.IOException;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.io.FileHandler;

public class Example1

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

Thread.sleep(3000);

//To enter the UN

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

Thread.sleep(3000);

//To enter the PWD

driver.findElement(By.xpath("//input[@type='password']")).sendKeys("Test@12345");

Thread.sleep(3000);

//To click on the Login Btn

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

Thread.sleep(3000);

//Step 1 --> Downcasting where you type cast the driver object into the TakesScreenshot interface

//This line of code catures the scrrenshot and stores it in the local memory

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

//Step 2--> Destination path, where you would like to have your scrrenshot capturing location

File destination=new File("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\ScreenshotsCapture\\xyz1.png");

//Step 3

FileHandler.copy(source, destination);

Thread.sleep(3000);

driver.close();

}

}

Parametrization - We store our test data in the excel sheet and fetch our data in an actual test

script

Excel sheet

CSV file

testNG data provider

Step 1 - To configure the Apache POI jar file to our Java project

Step 2- To create an excel sheet and store some data into it - Have a path of excel sheet handy

Step 3- To create an object of the FileInputStream class which accepts excel sheet path as an input in its constructor

Step 4- To open an excel sheet we need to use the create method

Step 5- To open the sheet in the excel sheet we need to use method get sheet

Step 6- To navigate to the row, we need to use method get row

Step 7 - To naviagte to the cell where our data is stored

Step 8 - To fetch the data from the excel sheet we need to understand first the data type of test data then we need use method called getstringcellvalue();

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

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.ss.usermodel.Workbook;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Example1

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

//To provide the path of the excel sheet

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData.xlsx");

//To open an excel sheet

Workbook work = WorkbookFactory.create(file);

//To open the sheet in the excel sheet

Sheet sh = work.getSheet("DDF");

//To navigate to the row where our data is stored

Row rw = sh.getRow(0);

//To navigate to the cell/coulmn where our data is stored

Cell cl = rw.getCell(0);

//To fetch the data from the excel sheet we need to understand first the data type of test data then we need use

//method called getstringcellvalue();

String value = cl.getStringCellValue();

System.out.println(value);

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Example2

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

double value = WorkbookFactory.create(file).getSheet("DDF").getRow(0).getCell(2).getNumericCellValue();

System.out.println(value);

}

}

<https://youtu.be/GMncZNs9Zrw?si=QcWZZhIp0jYB_-KY>

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Getsizeofrow

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

int lastRowIndex = WorkbookFactory.create(file).getSheet("DDF").getLastRowNum()+1;

System.out.println(lastRowIndex);

//last index of row - 3

//size of row - 4

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class GetsizeofColumns

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

short cellSize = WorkbookFactory.create(file).getSheet("DDF").getRow(0).getLastCellNum();

System.out.println(cellSize - 1 );

//Size of columns = 5

//Last index of columns - 4

}

}

package dropdown.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class Getsizeofdropdown

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

Thread.sleep(3000);

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

Thread.sleep(3000);

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[contains(text(), 'Create new accou')]")).click();

Thread.sleep(3000);

//Step no 01 --> To identify the dropdown

WebElement dateofBirthDropdown = driver.findElement(By.xpath("//select[@name='birthday\_year']"));

Select s=new Select(dateofBirthDropdown);

List<WebElement> options = s.getOptions();

int size = options.size();

System.out.println(size);

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Printalldatainarow

{

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

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

Sheet sh = WorkbookFactory.create(file).getSheet("DDF");

int lastRow = sh.getLastRowNum();

for(int i=0; i<=lastRow; i=i+1)

{

String value = sh.getRow(i).getCell(2).getStringCellValue();

System.out.println(value);

Thread.sleep(3000);

}

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

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.ss.usermodel.Workbook;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Example1

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

//To provide the path of the excel sheet

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData.xlsx");

//To open an excel sheet

Workbook work = WorkbookFactory.create(file);

//To open the sheet in the excel sheet

Sheet sh = work.getSheet("DDF");

//To navigate to the row where our data is stored

Row rw = sh.getRow(0);

//To navigate to the cell/coulmn where our data is stored

Cell cl = rw.getCell(0);

//To fetch the data from the excel sheet we need to understand first the data type of test data then we need use

//method called getstringcellvalue();

String value = cl.getStringCellValue();

System.out.println(value);

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Example2

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

double value = WorkbookFactory.create(file).getSheet("DDF").getRow(0).getCell(2).getNumericCellValue();

System.out.println(value);

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class GetsizeofColumns

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

short cellSize = WorkbookFactory.create(file).getSheet("DDF").getRow(0).getLastCellNum();

System.out.println(cellSize - 1 );

//Size of columns = 5

//Last index of columns - 4

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Getsizeofrow

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

int lastRowIndex = WorkbookFactory.create(file).getSheet("DDF").getLastRowNum()+1;

System.out.println(lastRowIndex);

//last index of row - 3

//size of row - 4

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Printalldatainarow

{

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

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

Sheet sh = WorkbookFactory.create(file).getSheet("DDF");

int lastRow = sh.getLastRowNum();

for(int i=0; i<=lastRow; i=i+1)

{

String value = sh.getRow(i).getCell(2).getStringCellValue();

System.out.println(value);

Thread.sleep(3000);

}

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Printalldataincolumns

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

Sheet sh = WorkbookFactory.create(file).getSheet("DDF");

int lastCellIndex = sh.getRow(0).getLastCellNum()-1;

for(int i=0; i<=lastCellIndex; i=i+1)

{

String value = sh.getRow(0).getCell(i).getStringCellValue();

System.out.print(value + " ");

}

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.CellType;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Verifythetypeofcell

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData2.xlsx");

Sheet sh = WorkbookFactory.create(file).getSheet("DDF");

Cell info = sh.getRow(1).getCell(1);

CellType typeofCell = info.getCellType();

System.out.println(typeofCell);

if(typeofCell == CellType.STRING)

{

String stringValue = info.getStringCellValue();

System.out.println(stringValue);

}

else if(typeofCell == CellType.NUMERIC)

{

double numericValue = info.getNumericCellValue();

System.out.println(numericValue);

}

else if(typeofCell == CellType.BOOLEAN)

{

boolean booleanValue = info.getBooleanCellValue();

System.out.println(booleanValue);

}

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.Cell;

import org.apache.poi.ss.usermodel.CellType;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Printalldatainrowsandcolumns

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData3.xlsx");

Sheet sh = WorkbookFactory.create(file).getSheet("sheet1");

int rowIndex = sh.getLastRowNum();

//Outer for loop for rows

for(int i=0; i<=rowIndex; i=i+1)

{

int cellIndex = sh.getRow(i).getLastCellNum()-1;

//Inner for loop for columns

for(int j=0; j<=cellIndex; j=j+1)

{

Cell value = sh.getRow(i).getCell(j);

CellType typeofCell = value.getCellType();

if(typeofCell == CellType.STRING)

{

String stringValue = value.getStringCellValue();

System.out.print(stringValue + " ");

}

else if(typeofCell == CellType.NUMERIC)

{

double numericValue = value.getNumericCellValue();

System.out.print(numericValue + " ");

}

else if(typeofCell == CellType.BOOLEAN)

{

boolean booleanValue = value.getBooleanCellValue();

System.out.print(booleanValue +" ");

}

}

System.out.println();

}

}

}

package parametrization;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;

import org.apache.poi.ss.usermodel.Sheet;

import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Printalldatainarowscoulmns2

{

public static void main(String[] args) throws EncryptedDocumentException, IOException

{

FileInputStream file=new

FileInputStream("C:\\Users\\hites\\OneDrive\\Desktop\\Automation Testing\\Parametrization\\TestData3.xlsx");

Sheet sh = WorkbookFactory.create(file).getSheet("sheet1");

int rowIndex = sh.getLastRowNum();

//outer for loop for rows

for(int i=0; i<=rowIndex; i=i+1)

{

int cellIndex = sh.getRow(i).getLastCellNum()-1;

//Inner for loop for columns

for(int j=0; j<=cellIndex; j=j+1)

{

String stringValue = sh.getRow(i).getCell(j).getStringCellValue();

System.out.print(stringValue +" ");

}

System.out.println();

}

}

}

Iframe - inline frame

Displaying a webpage as part of another webpage is called as iFrame

Switch focus to Iframe

1. driver.switchto().frame(" ");

1. String iframe id

2. String iframe name

3. int iframe index

4. webelement

driver.switchTo().parentFrame(); - It switches focus from the child frame to immediate parent frame

driver.switchTo().defaultContent(); - It switches focus from any child to main page

package iframe.handeling;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example1

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

driver.get("https://www.w3schools.com/js/tryit.asp?filename=tryjs\_myfirst");

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

Thread.sleep(3000);

//switch focus to Iframe

//driver.switchTo().frame("iframeResult"); //String iframe id

//driver.switchTo().frame("iframeResult"); //String iframe name

//driver.switchTo().frame(0); //String iframe index

driver.switchTo().frame(driver.findElement(By.xpath("//iframe[@id='iframeResult']")));

Thread.sleep(2000);

//Click on the click me button present on the iframe

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

Thread.sleep(2000);

//To switch the focus from iframe to main page

//driver.switchTo().parentFrame();

driver.switchTo().defaultContent();

Thread.sleep(2000);

//Click on the Run button present on the main page

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

Thread.sleep(2000);

driver.close();

}

}

Popups

1. Hidden Division popup

2. Alert popup

3. Child Browser popup

4. Authentication popup

5. File upload popup

6. File Download popup

1. Properties of the Hidden Division popup

: - A. Colourful popup

B. We can inspect the webelemts present inside the hidden division popup

C. No need to switch focus from the main page to the HDP

2. Alert popup properties

: - A. We can not insepect the alert popup

B. To handle these popup we need to switch selenium focus from the main page to the Alert

popup

C. These popups contains OK, Cancel, Exclamation mark and text

There are three methods

1. Accept - To click on the OK button

2. Dismiss - To click on the cancel button

3. get text - to get the text from the alert popup

package popups;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class HiddenDivisionpopup

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

driver.get("https://chercher.tech/practice/hidden-division-popup");

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

Thread.sleep(3000);

//To click on the view popup button

driver.findElement(By.xpath("//a[text()='View Pop-up']")).click();

Thread.sleep(3000);

//Input the name

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

Thread.sleep(3000);

//click on the close button

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

driver.quit();

}

}

package popups;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Alertpopup

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

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

Thread.sleep(3000);

//Click on the Click me button

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

Thread.sleep(3000);

//To switch the focus from the main page to the alert popup

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

//To get the text from the alert popup

String alertText = alt.getText();

System.out.println("Text from the alert popup : " +alertText );

Thread.sleep(3000);

//To click on the OK button

alt.accept();

Thread.sleep(3000);

driver.close();

}

}

<https://demoqa.com/alerts>

<https://chercher.tech/practice/hidden-division-popup>

<https://demo.automationtesting.in/Frames.html>

Popups

1. Hidden Division popup

2. Alert popup

3. Child Browser/window popup

These following three different popups can't be handled using Selenium. In order to handle these

popups we need to make use of the Robot class and Auto IT tool which are third party tools

4. Authentication popup

5. File upload popup

6. File Download popup

1. Properties of the Hidden Division popup

: - A. Colourful popup

B. We can inspect the webelemts present inside the hidden division popup

C. No need to switch focus from the main page to the HDP

2. Alert popup properties

: - A. We can not insepect the alert popup

B. To handle these popup we need to switch selenium focus from the main page to the Alert

popup

C. These popups contains OK, Cancel, Exclamation mark and text

There are three methods

1. Accept - To click on the OK button

2. Dismiss - To click on the cancel button

3. get text - to get the text from the alert popup

3. Child Browser/window popup

A. you can have maximize, minimize, cancel and links present inside it

B. You can inspect the Child Browser Popup

C. You need to switch the Selenium focus from the main page to the Child Browser popup

Window function accepts main page and child browser popup window address as an inputs

Getwindowhandle - Which return the address of main page only

Getwindowhandles - Which return the address of main page + Child window popup

package popups;

import java.util.ArrayList;

import java.util.Set;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class ChildBrowserPopup

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

driver.get("https://skpatro.github.io/demo/links/");

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

//Selenium implicit wait

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

//To click on the New Window button present on the main page

driver.findElement(By.name("NewWindow")).click();

//To get the address of both windows --> Main page(0) + Child Browser popup(1)

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

ArrayList< String > al=new ArrayList<String>(bothAddresses);

System.out.println("Main page address : " + al.get(0));

System.out.println("Child Browser popup address: " + al.get(1));

//To switch the focus to the Child window from the main page - This window function takes Child Browser window address

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

//To click on the Home link present on the Child Browser popup

driver.findElement(By.xpath("//span[text()='Home']")).click();

//To switch the focus to the main page from the Child Browser popup

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

//To click on the Home button present on the main page

driver.findElement(By.name("home")).click();

driver.close();

}

}

package multiple.elements.handling;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example1

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

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

List<WebElement> allLinks = driver.findElements(By.xpath("//a"));

int size = allLinks.size();

System.out.println("Size: " + size);

for( WebElement links : allLinks )

{

String linksTxt = links.getText();

System.out.println(linksTxt);

//Thread.sleep(2000);

}

}

}

package multiple.elements.handling;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class SelectAllCheckBoxesAscedningOrder

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Checkbox.html");

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

List<WebElement> allCheckboxes = driver.findElements(By.xpath("//input[@type='checkbox']"));

//Using for each loop

// for( WebElement checkbox: allCheckboxes)

// {

// checkbox.click();

// }

//Using for loop

for(int i=0; i<=allCheckboxes.size()-1; i=i+1)

{

allCheckboxes.get(i).click();

}

}

}

package multiple.elements.handling;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class DeselectAllCheckBoxesDescedningOrder

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Checkbox.html");

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

List<WebElement> allCheckboxes = driver.findElements(By.xpath("//input[@type='checkbox']"));

//Select all checkboxes from top to bottom/Ascending order

for(int i=0; i<=allCheckboxes.size()-1; i=i+1)

{

allCheckboxes.get(i).click();

}

//Deselect all checkboxes from bottom to top/Desceding order

for(int j=allCheckboxes.size()-1; j>=0; j=j-1)

{

allCheckboxes.get(j).click();

}

}

}

package autosuggestions.handling;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example1

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

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

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

List<WebElement> allOptions = driver.findElements(By.xpath("(//ul[@jsname='bw4e9b'])[1]//li"));

int size = allOptions.size();

System.out.println("Size :" + size);

for( WebElement option : allOptions)

{

String optionTxt = option.getText();

System.out.println(optionTxt);

}

}

}

package autosuggestions.handling;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example2

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

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

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

Thread.sleep(3000);

String expResult="flipkart sale";

List<WebElement> allOptions = driver.findElements(By.xpath("(//ul[@jsname='bw4e9b'])[1]//li"));

int size = allOptions.size();

System.out.println("Size :" + size);

for( WebElement option : allOptions)

{

String actResult = option.getText();

if(actResult.equals(expResult))

{

option.click();

break;

}

}

}

}

package actionsclass;

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.interactions.Actions;

public class Example1

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

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

//To identify the Create New Account button

WebElement CreateNewAccBtn = driver.findElement(By.xpath("//a[text()='Create new account']"));

//To create an object of Actions class

Actions act=new Actions(driver);

//To click on the CreateNewAccount Btn

//act.moveToElement(CreateNewAccBtn).click().build().perform();

act.click(CreateNewAccBtn).perform();

}

}

package actionsclass;

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.interactions.Actions;

public class ContextClick

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("https://demo.guru99.com/test/simple\_context\_menu.html");

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

//To identify the Login webelement

WebElement rightClickBtn = driver.findElement(By.xpath("//span[text()='right click me']"));

//Create an object of the Actions class

Actions act=new Actions(driver);

//Right click on the Login btn

//act.moveToElement(doubleClickBtn).contextClick().build().perform();

act.contextClick(rightClickBtn).perform();

}

}

package actionsclass;

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.interactions.Actions;

public class DoubleClick

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("https://demo.guru99.com/test/simple\_context\_menu.html");

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

//To identify the Login webelement

WebElement doubleClickBtn = driver.findElement(By.xpath("//button[text()='Double-Click Me To See Alert']"));

//Create an object of the Actions class

Actions act=new Actions(driver);

//Right click on the Login btn

//act.moveToElement(doubleClickBtn).doubleClick().build().perform();

act.doubleClick(doubleClickBtn).perform();

}

}

package actionsclass;

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.interactions.Actions;

public class DragandDrop

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("https://demo.guru99.com/test/drag\_drop.html");

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

//To identify the source element

WebElement source = driver.findElement(By.xpath("(//a[@class='button button-orange'])[4]"));

//To identify the target element

WebElement target = driver.findElement(By.xpath("(//ol[@align='center'])[2]"));

//To create an object of the Actions class

Actions act=new Actions(driver);

//act.dragAndDrop(source, target).perform();

act.moveToElement(source).clickAndHold().moveToElement(target).release().build().perform();

}

}

Actions class

Action - Which is an Interface

Actions - Which is a class

Actions class methods can be used to perform actions like

1. Mouse based actions

2. Keyboard based actions

1. Mouse based actions

: - A. Click - Left click at the current mouse location

B. Click(webelement) - Left click on the provided webelement address

C. Doubleclick - Double click at the current mouse location

D. Doubleclick(webelement) - Double click on the provided webelement address

E. Contextclick - Right click at the current mouse location

F. Contextclick(webelement) - Right click on the provided webelement address

G. MoveToElement(Webelement) - Move your mouse cursor to the provided webelement address

F. Click and Hold- Click and hold the element where mouse cursor is located

H. Click and hold(webelement) - Click and hold the webelement

J. Release - To release the webelement from the mouse

K. Drag and Drop(Source, Target) - To drag the element from the source location and drop it at the target location

L. Build - Chain of actions or multiple actions then your code should conclude/finish with this build method

O. Perform - When you are performing single action then your code should conclude/finish with perform method

without using build method first

package actionsclass;

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.interactions.Actions;

public class CustomizedListBox

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

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

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

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[text()='Create new account']")).click();

//To identify the month dropdown

WebElement birMonDropDown = driver.findElement(By.name("birthday\_month"));

//To create an object of the Actions class

Actions act=new Actions(driver);

Thread.sleep(3000);

//To click on the Month dropdown

act.click(birMonDropDown).perform();

Thread.sleep(3000);

//To move one key/option upward via keyboard

act.sendKeys(Keys.ARROW\_UP).perform();

Thread.sleep(3000);

//To move two/keys downward

act.sendKeys(Keys.ARROW\_DOWN).perform();

act.sendKeys(Keys.ARROW\_DOWN).perform();

Thread.sleep(3000);

//To click on the selected option

act.sendKeys(Keys.ENTER).perform();

}

}

package actionsclass;

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.interactions.Actions;

public class CustomizedListBox2

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

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

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

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[text()='Create new account']")).click();

//To identify the month dropdown

WebElement birMonDropDown = driver.findElement(By.name("birthday\_month"));

//To create an object of the Actions class

Actions act=new Actions(driver);

Thread.sleep(3000);

//To click on the Month dropdown

act.click(birMonDropDown).perform();

Thread.sleep(3000);

//To move to bottom

for(int i=1; i<=3; i++)

{

act.sendKeys(Keys.ARROW\_DOWN).perform();

}

Thread.sleep(3000);

//To move to the top

for( int i=1; i<=12; i++)

{

act.sendKeys(Keys.ARROW\_UP).perform();

}

Thread.sleep(3000);

act.sendKeys(Keys.ENTER).perform();

}

}

package actionsclass;

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.interactions.Actions;

public class CustomizedListBox3

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

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

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

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[text()='Create new account']")).click();

//To identify the day dropdown

WebElement dayDropDown = driver.findElement(By.name("birthday\_day"));

//To create an object of the Actions class

Actions act=new Actions(driver);

Thread.sleep(3000);

//To click on the Month dropdown

act.click(dayDropDown).perform();

Thread.sleep(3000);

//To move to bottom

act.sendKeys(Keys.END).perform();

Thread.sleep(3000);

//To move to the top

for( int i=1; i<=12; i++)

{

act.sendKeys(Keys.ARROW\_UP).perform();

}

Thread.sleep(3000);

act.sendKeys(Keys.ENTER).perform();

}

}

package actionsclass;

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.interactions.Actions;

public class CustomizedListBox4

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

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

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

//Click on the Create New Account btn

driver.findElement(By.xpath("//a[text()='Create new account']")).click();

//To identify the day dropdown

WebElement dayDropDown = driver.findElement(By.name("birthday\_day"));

//To create an object of the Actions class

Actions act=new Actions(driver);

Thread.sleep(3000);

//To click on the Month dropdown

act.click(dayDropDown).perform();

Thread.sleep(3000);

//To move to bottom

act.sendKeys(Keys.END).perform();

Thread.sleep(3000);

//To move to the top

act.sendKeys(Keys.HOME).perform();

Thread.sleep(3000);

act.sendKeys(Keys.ENTER).perform();

}

}

Iframe - inline frame

Displaying a webpage as part of another webpage is called as iFrame

Switch focus to Iframe

1. driver.switchto().frame(" ");

1. String iframe id

2. String iframe name

3. int iframe index

4. webelement

driver.switchTo().parentFrame(); - It switches focus from the child frame to immediate parent frame

driver.switchTo().defaultContent(); - It switches focus from any child to main page

Popups

1. Hidden Division popup

2. Alert popup

3. Child Browser/window popup

These following three different popups can't be handled using Selenium. In order to handle these

popups we need to make use of the Robot class and Auto IT tool which are third party tools

4. Authentication popup

5. File upload popup

6. File Download popup

1. Properties of the Hidden Division popup

: - A. Colourful popup

B. We can inspect the webelemts present inside the hidden division popup

C. No need to switch focus from the main page to the HDP

2. Alert popup properties

: - A. We can not insepect the alert popup

B. To handle these popup we need to switch selenium focus from the main page to the Alert

popup

C. These popups contains OK, Cancel, Exclamation mark and text

There are three methods

1. Accept - To click on the OK button

2. Dismiss - To click on the cancel button

3. get text - to get the text from the alert popup

3. Child Browser/window popup

A. you can have maximize, minimize, cancel and links present inside it

B. You can inspect the Child Browser Popup

C. You need to switch the Selenium focus from the main page to the Child Browser popup

Window function accepts main page and child browser popup window address as an inputs

Getwindowhandle - Which return the address of main page only

Getwindowhandles - Which return the address of main page + Child window popup

Actions class

Action - Which is an Interface

Actions - Which is a class

Actions class methods can be used to perform actions like

1. Mouse based actions

2. Keyboard based actions

1. Mouse based actions

: - A. Click - Left click at the current mouse location

B. Click(webelement) - Left click on the provided webelement address

C. Doubleclick - Double click at the current mouse location

D. Doubleclick(webelement) - Double click on the provided webelement address

E. Contextclick - Right click at the current mouse location

F. Contextclick(webelement) - Right click on the provided webelement address

G. MoveToElement(Webelement) - Move your mouse cursor to the provided webelement address

F. Click and Hold- Click and hold the element where mouse cursor is located

H. Click and hold(webelement) - Click and hold the webelement

J. Release - To release the webelement from the mouse

K. Drag and Drop(Source, Target) - To drag the element from the source location and drop it at the target location

L. Build - Chain of actions or multiple actions then your code should conclude/finish with this build method

O. Perform - When you are performing single action then your code should conclude/finish with perform method

without using build method first

2. Keyboard based actions

A. Enter

B. Up arrow

C. Down arrow

package dynamic.elements.handling;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example1

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

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

//To click on the close button

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

Thread.sleep(3000);

//To enter the input

driver.findElement(By.xpath("//input[@type='text']")).sendKeys("Robin Sharam books");

Thread.sleep(3000);

//To click on the search icon

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

Thread.sleep(3000);

//Get the ratings

String ratings = driver.findElement(By.xpath("((//div[@class='\_4ddWXP'])[4]//span)[2]")).getText();

System.out.println(ratings);

}

}

package webtable.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class GetCoulmnSizeinaRow

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Webtablewithborder.html");

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

List<WebElement> allOptions = driver.findElements(By.xpath("//tr[1]/th"));

int sizeofColumns = allOptions.size();

System.out.println("sizeofColumns: " + sizeofColumns);

}

}

package webtable.handling;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class GetDatafromParticularIndex

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Webtablewithborder.html");

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

String info = driver.findElement(By.xpath("//tr[1]/th[2]")).getText();

System.out.println(info);

}

}

package webtable.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class GetRowSize

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Webtablewithborder.html");

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

List<WebElement> rowSize = driver.findElements(By.xpath("//tr"));

int size = rowSize.size();

System.out.println("rowSize: " + size);

}

}

package webtable.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class GetallDatafromColumn

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Webtablewithborder.html");

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

List<WebElement> allOptions = driver.findElements(By.xpath("//tr/td[2]"));

for( WebElement option : allOptions)

{

String text = option.getText();

System.out.println(text);

}

}

}

package webtable.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class PrintDataFromtheRow

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Webtablewithborder.html");

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

List<WebElement> allOptions = driver.findElements(By.xpath("//tr[2]"));

for( WebElement option : allOptions)

{

String text = option.getText();

System.out.println(text);

}

}

}

package webtable.handling;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class PrintDataFromtheTable

{

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("file:///C:/Users/hites/OneDrive/Desktop/Automation%20Testing/HTML%20Coding/Webtablewithborder.html");

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

List<WebElement> allOptions = driver.findElements(By.xpath("//tbody"));

for( WebElement option : allOptions)

{

String text = option.getText();

System.out.println(text);

}

}

}

Different types of exception

1. Webdriver exception- URL is not proper/appropriate

2. Unreachable browser exception - When you are intrupting the scripts manually

3. NoSuchWindowException - target window already closed

4. UnhanledAlertException - When the user is not handling the alert popup

5. UnexpectedTagnameException - When you are making use of the Select tagname while handling the customized

dropdown

6. NoAlertException- When you are handling the Hidden Disvision popup by considering it as a Alert popup

7. NotConnectedException - When your Selenium Jar files are not compltely configured

8. UnsupporteOperationException - When the user is performing the deselect operation on the single selectable

dropdown instead of multi-selectable

9. NoSuchElementException - Three scenarios in which the user might face this exception

A. Element has not been loaded completely on the webpage

B. Element is not present on the main page. It could be present in the Alert popup, Iframe etc

C. When your x-Path is incorrect

10. ElementNotInteractableException - When you are manually intrupting the script

<https://www.baeldung.com/selenium-implicit-explicit-wait>

<https://www.browserstack.com/guide/wait-commands-in-selenium-webdriver>

<https://www.javatpoint.com/selenium-waits>

package scrolling.operations;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example1

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

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

Thread.sleep(3000);

//Scroll down --> 2nd parameter +value

((JavascriptExecutor)driver).executeScript("window.scrollBy(0, 5000)");

Thread.sleep(3000);

//Scroll up --> 2nd parameter -value

((JavascriptExecutor)driver).executeScript("window.scrollBy(0, -5000)");

Thread.sleep(3000);

//Scroll right --> 1st parameter +ve value

((JavascriptExecutor)driver).executeScript("window.scrollBy(1000, 0)");

Thread.sleep(3000);

//Scroll left --> 1st parameter -ve value

((JavascriptExecutor)driver).executeScript("window.scrollBy(-1000, 0)");

}

}

package scrolling.operations;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Example2

{

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

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

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

//Identify the webelement where you want to scroll

WebElement marathilink = driver.findElement(By.xpath("//a[@title='Marathi']"));

Thread.sleep(3000);

((JavascriptExecutor)driver).executeScript("arguments[0].scrollIntoView();",marathilink );

}

}

What is Selenium Framework? Types of Framework?

The Selenium Framework is a code structure that makes code maintenance easy and efficient.

Without frameworks, users may place the “code” and “data” at the same location which is neither reusable nor readable.

Frameworks produce beneficial outcomes like increased code reusability, higher portability,

reduced cost of script maintenance, better code readability, etc.

There are mainly three type of frameworks created by Selenium WebDriver to automate manual test cases

1. Data Driven

2. Keyword Driven

3. Hybrid

1. Data Driven

Data Driven Framework in Selenium is a method of separating Test data from the test case.

Once the Test data are separated from the test case, it can be easily modified for a specific functionality without changing the code.

It is used to fetch Test Data from external files like Excel, .csv, .xml or some database tables.

===========================================================================================================================================================

POM with pagefactory

POM(page object module):

It is a java design pattern use for design of classes in Test script.

Page Object model is an object design pattern in Selenium, where

1. Web pages are represented as classes &

2. The various elements on the page are defined as variables on the class.

In this case we will use Page Factory to initialize web elements that are defined in web page classes or Page Objects.

POM Strictly follows encapsulation concept where

1. Data member should be declared globaly with access level private using @findBy Annotation

2. Initialize within a constructor with access level public using pagefactory

3. Utilize within a method with access level public

Note:

1. No of D.M. that need to be created under a pom class will depends on no of element that need to be handle in a webpage.

2. pom class will not contain a main method, to run a pom class we require another class with main() ie. Test class

3. Test class will contain all the navigation steps to test an application

==========================================================================================================================

POM with pagefactory

POM(page object module):

IT is a java design pattern use for design of classes in Test script.

Page Object model is an object design pattern in Selenium, where

1. Web pages are represented as pom classes &

2. The various elements on the page are defined as variables on the class.

In this case we will use Page Factory to initialize web elements that are defined in web page classes or Page Objects.

POM Strictly follows encapsulation concept where

1. Data member should be declared globaly with access level private using @findBy Annotation

2. Initialize within a constructor with access level public using pagefactory

3. Utilize within a method with access level public

Note:

1. No of D.M. that need to be created under a pom class will depends on no of element that need to be handle in a webpage.

2. pom class will not contain a main method, to run a pom class we require another class with main() ie. Test class

3. Test class will contain all the navigation steps to test an application

Pagefactory:

It is a class which contains static method like initElements.

To initialize global D.M./variable in PageFactory we need to use initElements method within the constructor.

Syntax:

PageFactory.initElements(driver, this);

initElements will initialize D.M by identifying each componant present in a webpage by using @findBy annotation,

which takes locator type as an input.

Syntax:

@FindBy(locator Type ="locator value/exression")

private WebElement D.M. ;

Working of PageFactory:

1. While executing Test Script initElement method will convert all the the data members @findBy annotation to findElement(),

this process is known as basic/early initialization.-->after creating object of pom class

@FindBy(xpath="//span[text()='KV']") private WebElement PN;

----> private WebElement PN = driver.findElement(By.xpath("//span[text()='KV']"));

2. To perform action on component we need to call a methods.

3. Before performing each action initElement method will identifies component present or not,

then it will do complete initialization

this process is known as late/lazy initialization. ---->after method call

Disadvantage of simple POM:

simple POM will initialize all the component before performing actions,

but sometimes application may contains few components which will be hidden &

displayed once we perform action on components, that hidden component will not be displayed while pom initializing,

so it throws "No such element" exception.

To overcome drawback of pom, we need to use "PageFactory" which is an extension of pom.

Diffrence between POM & PageFactory

simple POM Pom with PageFactory

1. It will initialize/identify all the D.M. present in It will initialize/identify the D.M. present in a class before

class completely before performing action on components performing each action.

2. It will use if webpage is not containing hidden elements. It will be used if webpage is containing hidden elements.

package simple.pom.concept;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

public class Signuppage

{

//POM class no 01

private WebElement signUpBtn;

private WebElement emailInputField;

private WebElement continueBtn;

public Signuppage(WebDriver driver)

{

signUpBtn=driver.findElement(By.xpath("//a[@class='HeaderMenu-link HeaderMenu-link--sign-up "

+ "flex-shrink-0 d-none d-lg-inline-block no-underline border color-border-default rounded px-2 py-1']"));

emailInputField=driver.findElement(By.xpath("//input[@id='email']"));

continueBtn=driver.findElement(By.xpath("(//button[@type='button'])[2]"));

}

public void clickSignupBtn()

{

signUpBtn.click();

}

public void enterEmailID()

{

emailInputField.sendKeys("fusion@gmail.com");

}

public void clickContinueBtn()

{

continueBtn.click();

}

}

package simple.pom.concept;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Signuptest

{

//Test class

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("https://github.com/");

Thread.sleep(3000);

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

Signuppage objectSignUpPage=new Signuppage(driver);

Thread.sleep(5000);

objectSignUpPage.clickSignupBtn();

Thread.sleep(6000);

objectSignUpPage.enterEmailID();

Thread.sleep(5000);

objectSignUpPage.clickContinueBtn();

}

}

package pom.with.pagefactory;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class Signuppage

{

//POM class no 01

@FindBy (xpath = "//a[@class='HeaderMenu-link HeaderMenu-link--sign-up \"\r\n" +

"+ \"flex-shrink-0 d-none d-lg-inline-block no-underline border color-border-default rounded px-2 py-1']" )private WebElement signUpBtn;

//private WebElement signUpBtn =driver.findElement(By.xpath("//a[@class='HeaderMenu-link HeaderMenu-link--sign-up "

// + "flex-shrink-0 d-none d-lg-inline-block no-underline border color-border-default rounded px-2 py-1']"));

@FindBy ( xpath = "//input[@id='email']") private WebElement emailInputField;

//private WebElement emailInputField=driver.findElement(By.xpath("//input[@id='email']"));

@FindBy (xpath = "(//button[@type='button'])[2]") private WebElement continueBtn; //Hidden element

//private WebElement continueBtn=driver.findElement(By.xpath("(//button[@type='button'])[2]"));

public Signuppage(WebDriver driver)

{

PageFactory.initElements(driver, this);

}

public void clickSignupBtn()

{

signUpBtn.click();

}

public void enterEmailID()

{

emailInputField.sendKeys("fusion@gmail.com");

}

public void clickContinueBtn()

{

continueBtn.click();

}

}

package pom.with.pagefactory;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import simple.pom.concept.Signuppage;

public class Signuptest

{

//Test class

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

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

driver.get("https://github.com/");

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

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

Signuppage objectSignUpPage=new Signuppage(driver);

objectSignUpPage.clickSignupBtn();

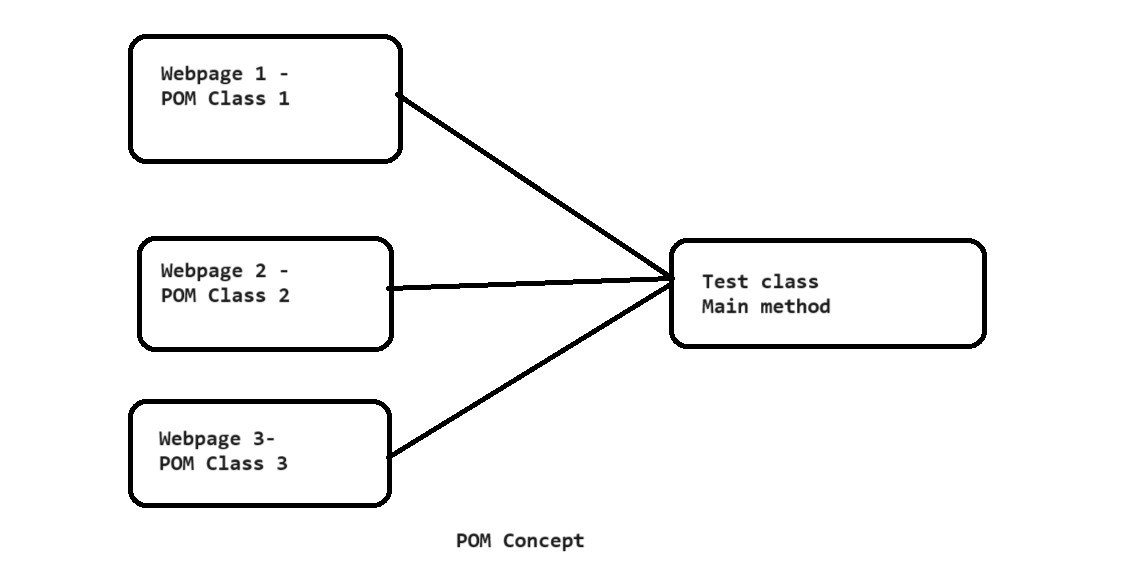
objectSignUpPage.enterEmailID();

Thread.sleep(3000);

objectSignUpPage.clickContinueBtn();

}

}



TestNG

TestNG

TestNG is a java unit framework use for writting/designing of Test classes.

->Example of normal Test Class

->Example of TestNG Test class

click on add testNG library

click on import Test

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

1. Emailable Report

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

By looking at the result, you can easily identify how many test cases are passed, failed and skipped.

By looking at the report, you will come to know what the status of the project is.

Selenium web driver is used for automating the web-application, but it won't generate any reports.

The TestNG will generate the default report.

---Steps to generate Emailable report----

1. execute Test class and refresh the project.

2. You will get test-output folder.

3. In That folder Right click on the "emailable-report.html" and select the option Open with the web browser or double click on it.

Note:

1. if we use sop() to display text as a output then result will be displayed in console not in emailable report.

2. To display text in emailable report we need to use static method log present in Reporter class.

eg. Reporter.log("String msg", true)

sop--> console

Reporter.log("msg", true)---> console+ emailable report

Reporter.log("msg", false)/Reporter.log("msg") --> emailable report

package testng;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class WithoutTestNGClass

{

//Main method

public static void main(String[] args)

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

}

}

package testng;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class WithTestNGClass

{

@Test

public void test1()

{

//Set a path of chromeDriver -- Paramter 1--> Name of the browser , Parameter 2 --> ChromeDriver paty

System.setProperty("Webdriver.chrome.driver", "C:\\Users\\hites\\Downloads\\chromedriver\_win32 (1)");

//Upasting - where we create an object of sub-class(ChromerDriver) and provide reference of super interface(WebDriver)

WebDriver driver=new ChromeDriver();

//To launch the URL/webpage

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

}

}

package testng;

import org.testng.Reporter;

import org.testng.annotations.Test;

public class TestNGExample2

{

//Test class

//Test case 1

@Test

public void testCase1()

{

Reporter.log("Test case 1",true);

}

//Test case 2

@Test

public void testCase2()

{

Reporter.log("Test case 2", false);

}

//Test case 3

@Test

public void testCase3()

{

Reporter.log("Test case 3");

}

//Test case 4

@Test

public void testCase4()

{

Reporter.log("Test case 3", true);

}

}

package testng.annotations;

import org.testng.Reporter;

import org.testng.annotations.AfterClass;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

public class Example1

{

//Test class

//One time code - Browser open code

@BeforeClass

public void openBrowser()

{

Reporter.log("Browser open code", true);

}

@BeforeMethod

public void loginToApplication()

{

Reporter.log("Login with valid credentials", true);

}

//@Test --> Actual test case

@Test

public void verifyUsername()

{

Reporter.log("verify Username test case", true);

}

@AfterMethod

public void logoutFromTheApplication()

{

Reporter.log("logout From The Application", true);

}

@AfterClass

public void closeBrowser()

{

Reporter.log("close Browser", true);

}

}

package testng.annotations;

import org.testng.annotations.AfterClass;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterSuite;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class Example2

{

//Test class

@AfterSuite

public void m1()

{

System.out.println("After Suite");

}

@AfterMethod

public void m2()

{

System.out.println("After Method");

}

@Test

public void m3()

{

System.out.println("Test case 1");

}

@Test

public void m4()

{

System.out.println("Test case 2");

}

@AfterTest

public void m5()

{

System.out.println("After test");

}

@BeforeMethod

public void m6()

{

System.out.println("Before Method");

}

@BeforeTest

public void m7()

{

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

}

@AfterClass

public void m8()

{

System.out.println("After Class");

}

@BeforeClass

public void m9()

{

System.out.println("Before Class");

}

@BeforeSuite

public void m10()

{

System.out.println("Before Suite");

}

}

2. TestNG Annotation

1. @Test

2. @BeforeMethod

3. @AfterMethod

4. @BeforeClass

5. @AfterClass

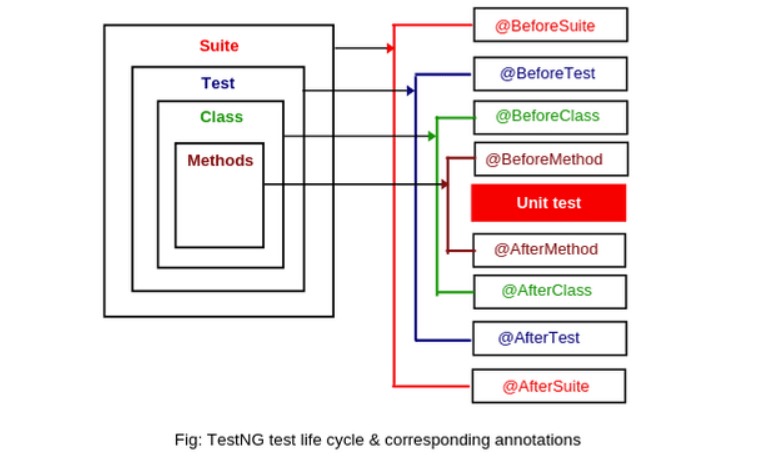
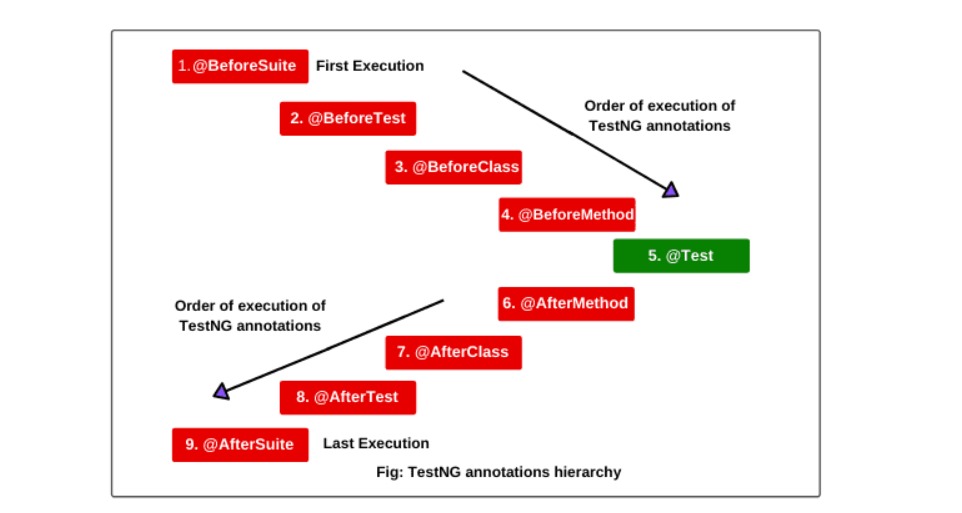
1. @Test:- Use for execution of test method/TC.

2. @BeforeMethod:- It is use for execution of test method before execution of every test method with an annotation @Test.

3. @AfterMethod:- It is use for execution of test method after execution of every test method with an annotation @Test.

4. @BeforeClass:- It is use for execution of test method before execution of test class.

5. @AfterClass:- It is use for execution of test method after execution of test class.



TestNG Annotations

Test case scenario - To verify whether the user's UN is matching with the actual UN after successfull login

Steps.

1. Launch the browser with the Facebook URL - One time action - @BeforeClass

2. Login with valid credentials - Pre-condition to perform actual test case - @BeforeMethod

3. Actual test case - @Test - Where you compare the actual output vs expected output

4. Logout from the application - After test case execution step - @AfterMethod

5. Close the browser - One time action - @AfterClass

[10:00, 27/10/2023] Hitesh Desale: 3. TestNG Keyword:

1. invocationCount

2. priority

3. enabled

4. TimeOut

5. dependsOnMethods

1. invocationCount:

Sometimes same test method/TC need to be executed multiple which can be possible by using TestNG keyword "invocationCount"

eg. invocationCount=5;

2. priority:

To change test method/TC execution order we need to use TestNG keyword "priority".

eg. priority=1

Note: priority can be 1. bydefault=0

2. +ve integer

3. -ve integer

4. Duplicate

priority can't be 1. Decimals

2. Variables

3. enabled:

Disabling a test method/TC in TestNG can be achieved by setting the enabled attribute of the @Test annotation to false.

eg. enabled=false

4.TimeOut:

If test class contains multiple test methods if one of the test method is time consuming to execute then

TestNG bydefault fail that TC & executes the other TC.

eg. @Test(timeOut=8000)

5. dependsOnMethods:

If 1 TC execution depends on multiple TC then we need to use "dependsOnMethods" TestNG keyword.

eg. dependsOnMethods= {"TC name"}

[10:00, 27/10/2023] Hitesh Desale: package testng.keywords;

import org.testng.Assert;

import org.testng.annotations.Test;

public class Example1

{

//dependsonMethods keyword

@Test

public void login()

{

System.out.println("Login test case");

Assert.fail();

}

@Test(dependsOnMethods = "login")

public void logout()

{

System.out.println("Logout test case");

}

}

package testng.keywords;

import org.testng.annotations.Test;

public class Example2

{

//priority keyword

@Test(priority = -3 )

public void login()

{

System.out.println("Login");

}

@Test(priority = 2)

public void signUp()

{

System.out.println("signUp");

}

@Test(priority = -4)

public void logout()

{

System.out.println("logout");

}

@Test(priority = 5)

public void EMC()

{

System.out.println("EMC");

}

@Test(priority = 5)

public void HUL()

{

System.out.println("HUL");

}

@Test(priority = 1)

public void sophan()

{

System.out.println("sophan");

}

}

package testng.keywords;

import org.testng.annotations.Test;

public class Example3

{

//Timeout keyword

@Test(timeOut = 2000)

public void login() throws InterruptedException

{

System.out.println("Login");

Thread.sleep(3000);

}

}

package testng.keywords;

import org.testng.annotations.Test;

public class Example4

{

//Enabled keyword

@Test(enabled = false)

public void testCase1()

{

System.out.println("test case 1");

}

}

package testng.keywords;

import org.testng.annotations.Test;

public class Example5

{

@Test(invocationCount = 4)

public void login()

{

System.out.println("login");

}

}

package testng.suite.operations;

import org.testng.Assert;

import org.testng.annotations.Test;

public class Example1

{

@Test

public void testCase1()

{

System.out.println("test Case1 is running from Example1");

//Assert.fail();

}

@Test

public void testCase2()

{

System.out.println("test Case2 is running from Example1");

}

@Test

public void testCase3()

{

System.out.println("test Case3 is running from Example1");

}

}

<?xml version="1.0" encoding="UTF-8"?>

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

<suite name="Regression Suite">

<test name="Dev Environment testing">

<classes>

<class name="testng.suite.operations.Example1" />

<class name="testng.keywords.Example4" />

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

Test-Suite:

It is xml file which contains all the test classes name which need to be executed.

It is use to execute all/multiple Test classes.

<suite name="Suite name">

<test name="Test name">

<classes>

<class name="packageName.className"/>

</classes>

</test>

</suite>

Procedure to create xml file

Suite---> multiple tests

1 test--> multiple classes

=====================================================================================================================================

Failed.xml

While executing the automation scripts, test cases may fail for several reasons.

To optimize our next runs,

we need to re-run only failed test cases

Steps to execute failed.xml file

1. Create testng. xml file under project folder.

2. execute testng.xml file

3. Refresh the project

4. In the test-output folder >> testng-failed. xml file will be created.

5. execute "testng-failed. xml"

In this way we can execute fail testcases in TestNG class.

Reasons for fail TC

1. envirnment issue

2. script error

3. bug

=========================================================================================================================

Verification/Assertion

1. hard assert (Assert class) -->static method

2. soft assert (SoftAssert class) --> non-static method

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

Disable TC Execution

1. from Test class using 'enabled=false' keyword

2. from suite using

2A. exclude keyword

2B. include keyword

=============================================================================================================================

Grouping of Test cases

TestNG Groups allow you to perform groupings of different test methods

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

Parallel testing or parallel execution

Parallel testing or parallel execution, as the name suggests,

is a process of running the test case parallelly rather than one after the other.

In parallel testing, the program’s multiple parts(or modules) execute together, saving the testers a lot of time and effort.

=================================================================================================================================

TestNG advantages

It has different assertions that helps in checking the expected and actual results.

It allows to assign priority to test methods

It allows to define dependency of one test method over other method

It provide Detailed (HTML) reports/ Emailable report

It allows grouping of test methods into test class

TestNG provides parallel execution of test methods

TestNG provides multibrowser/CT testing of test methods

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

Data provider

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

Junit TestNG

Parallel execution JUnit does not support to run parallel tests. TestNG can run parallel tests.

Supports Annotation It does not support advanced annotation. It supports advanced annotation.

Dependency tests The dependency tests are missing in JUnit. Dependency tests are present in TestNG.

Grouping tests Grouping tests together is not possible in JUnit. Tests can be grouped together and run parallel.

<?xml version="1.0" encoding="UTF-8"?>

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

<suite name="Regression Suite"

guice-stage="DEVELOPMENT">

<test thread-count="5" name="Dev Environment testing">

<classes>

<class name="testng.suite.operations.Example1">

<methods>

<include name="testCase1" />

<include name="testCase2" />

</methods>

</class>

</classes>

</test>

</suite>

<?xml version="1.0" encoding="UTF-8"?>

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

<suite name="Regression Suite" guice-stage="DEVELOPMENT">

<test thread-count="5" name="Dev Environment testing">

<classes>

<class name="testng.suite.operations.Example1">

<methods>

<exclude name="testCase1" />

<exclude name="testCase2" />

</methods>

</class>

</classes>

</test>

</suite>

package testng.suite.operations;

import org.testng.annotations.Test;

public class Example2

{

@Test(groups = "Login")

public void loginwithValidCred()

{

System.out.println("loginwithValidCred is running from Example1");

//Assert.fail();

}

@Test(groups = "Login")

public void loginwithInvalidCred()

{

System.out.println("loginwithInvalidCred is running from Example1");

}

@Test(groups = "Login")

public void loginwithEmptyCred()

{

System.out.println("loginwithEmptyCred is running from Example1");

}

@Test(groups = "Logout")

public void logoutwhenuserisnotLoggedIn()

{

System.out.println("logoutwhenuserisnotLoggedIn is running from Example1");

}

@Test(groups = "Logout")

public void logoutwhenuserisLoggedIn()

{

System.out.println("logoutwhenuserisLoggedIn is running from Example1");

}

}

<?xml version="1.0" encoding="UTF-8"?>

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

<suite name="Grouping suite">

<groups>

<run>

<include name="Logout">

</include>

</run>

</groups>

<test thread-count="5" name="Login Test">

<classes>

<class name="testng.suite.operations.Example2" />

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

<?xml version="1.0" encoding="UTF-8"?>

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

<suite name="parallel testing suite" parallel="tests">

<test name="test 1">

<classes>

<class name="testng.annotations.Example1" />

</classes>

</test> <!-- Test -->

<test name="test 2">

<classes>

<class name="testng.annotations.Example2" />

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

Suite.xml

1. You can execute one or multiple test classes from the suite

2. Include or exclude the respective test cases

3. Grouping of test cases

4. Parallel testing or Parllel execution

package hardassert.assertclass;

import org.testng.Assert;

import org.testng.annotations.Test;

public class Example1

{

//Assert class is having static methods

@Test

public void testCase1()

{

String actaulOutput="Arti";

String expectedOutput="Sophan";

//Assert.assertEquals(actaulOutput, expectedOutput);

//Assert.assertNotEquals(actaulOutput, expectedOutput);

//Assert.assertNull(actaulOutput);

//Assert.assertNotNull(actaulOutput);

//Assert.assertTrue(true);

Assert.assertFalse(true);

System.out.println("Hard Assert/Assert class");

Assert.assertEquals(actaulOutput, expectedOutput);

}

}

package softassert.softassertclass;

import org.testng.annotations.Test;

import org.testng.asserts.SoftAssert;

public class Example1

{

//Soft assert is having non-static methods

@Test

public void testCase1()

{

String actaulOutput="Arti";

String expectedOutput="Sophan";

SoftAssert softAssert=new SoftAssert();

softAssert.assertEquals(actaulOutput, expectedOutput);

System.out.println("Soft assert class");

softAssert.assertNull(actaulOutput);

//Gives assertion error message in the cosole + testng report

softAssert.assertAll();

}

}

Verification/Assertion

1. hard assert (Assert class) -->static method

2. soft assert (SoftAssert class) --> non-static method