# Quick Reference Book Selenium

## **Shortcuts**

Ctrl + Space – Get Method

Ctrl + Shift + F - Auto Formatter

Ctrl + Shift + O – Remove unused Imports

Ctrl + E – Currently Open Editors

Ctrl + Shift + R – Open Resource

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

## **Firefox Browser Instantiation:**

public class Firefox {

private WebDriver driver;

@BeforeClass

public void setUp() throws Exception {

driver = new FirefoxDriver();

driver.get(BaseURL);

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

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

}

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

## **Chrome Browser Instantiation:**

public class GoogleChrome {

private WebDriver driver;

@BeforeClass

public void setUp() throws Exception {

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

driver = new ChromeDriver();

s

}

Note: *chromedriver.exe* file has to be added to project directory.

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

## **Internet Explorer Browser Instantiation:**

public class InternetExplorer {

private WebDriver driver;

@BeforeClass

public void setUp() throws Exception {

File file = new File("IEDriverServer.exe");

System.*setProperty*("webdriver.ie.driver",file.getAbsolutePath());

driver = new InternetExplorerDriver(); driver.navigate().to(BaseURL);

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

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

}

Note: *IEDriverServer.exe* file has to be added to project directory.

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

## **General Method for Browser Instantiation (Firefox, Chrome and IE):**

//If driver is declared in BaseClass (General Utility Class), make sure WebDriver is declared in BaseClass or test script (not in both class).

public void browserInstantiate(String BrowserName) {

String browserName = BrowserName.toLowerCase();

if (browserName.contains("mozilla") || browserName.contains("firefox") || browserName.contains("ff"))

{

driver = new FirefoxDriver();

}

else if (browserName.contains("google") || browserName.contains("chrome") || browserName.contains("gc"))

{

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

driver = new ChromeDriver();

}

else if (browserName.contains("internet") || browserName.contains("explorer") || browserName.contains("ie"))

{

File file = new File("D://Selenium/IEDriverServer.exe");

System.*setProperty*("webdriver.ie.driver", file.getAbsolutePath());

driver = new InternetExplorerDriver();

}

}

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

## **Find Element, click Element, send Text to Element.**

WebElement we =driver.findElement(By.name("nameOfElement"));

we.click();

we.sendKeys("Hello");

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

## **Navigate to Page Url**

driver.navigate().to("http://www.example.com");

driver.navigate().forward();

driver.navigate().back();

driver.navigate().refresh();

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

## **Debug error:**

NullPointerException : It will appear when WebDriver is declared incorrectly or Declared Twice:

EX1:

//WebDriver Declared incorrectly

public class NewTest {

WebDriver driver;

@BeforeClass

public void Setup() {

WebDriverdriver = **new** FirefoxDriver();

}

}

//Correct code:

public class NewTest {

WebDriver driver;

@BeforeClass

public void Setup() {

driver = **new** FirefoxDriver();

}

}

Ex2:

//WebDriver Declared in BaseClass(General Utility Class) and Test Script

//BaseClass

public class BaseClass {

public Static WebDriver driver;

…CodeContinues

}

//TestScript

public class NewTest extends BaseClass {

WebDriver driver;

…CodeContinues

}

//Correct code

public class BaseClass {

public Static WebDriver driver;

…CodeContinues

}

//TestScript

public class NewTest extends BaseClass {

//use driver with out instantiating or get driver in constructor and use as Private webDriver

…CodeContinues

}

f.queryinterface is not a function exception:

Check whether the method invocating in failed Class are given correct value, Or if it is needed to read value from property file (but not reading).

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

## **General Method openPage:**

@BeforeClass

public void OpenHomePage() {

driver = new FirefoxDriver();

}

@Test(priority = 0)

public void loadUrl() throws Exception {

String pageLoadMsg = openPage("http://translate.google.com");

if (!(pageLoadMsg.equals("SUCCESS"))) {

System.*out*.println(pageLoadMsg);

throw new Exception("Page Not Loaded Successfully");

}

}

public String openPage(String Url) {

driver.get(Url);

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

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

String title = driver.getTitle();

if (title.equals("Error 404 (Not Found)!!1")) {

return "ERROR : Page not found";

} else if (title.equals("Problem loading page")) {

return "ERROR : Problem loading page";

} else if (title.contains("is not found")) {

return "ERROR : Page not found";

} else if (title.equals("Address not valid")) {

return "ERROR : Page not found";

} else if (title.contains("is not available")) {

return "ERROR : Page not found";

} else if (title.contains("Internet Explorer cannot display the webpage")) {

return "ERROR : Page not found";

} else {

return "SUCCESS";

}

}

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

## **How to write Xpath**

Syntax: //tagName[@attributeName=’attributeValue’ ]

EX:

Html…

<div id=”divId” class=”divClassName” type=”divType”> someText </div>

<div id=”divId2”> </div>

<a href=”www.somewebpage.co.in” class=”aClass” > someLinkText </div>

Xpath…

//div[@id=’divId’]

//div[@class=’divClassName’]

//div[@type=’divType’]

//div[text()=’someText’ ]

//\*[@id=’divId’]

Identify child element using parent tag attribute: xpathForParentTag/childTag

//div[@id=’divId’]/div[1]

//div[@id=’divId’]/div/a[1]

To get Child node

Syntax: //parentTagName[@attributeName=’attributeValue’]/descendant::childTagName[@childAttributeName=’childAttributeValue’]

//div[@id=’divId’]/descendant::div[@id=’divId2’]

//div[@id=’divId’]/descendant::a[@class=’aClass’]

//div[@id=’divId’]/descendant::a[text()=’ someLinkText’]

Multiple child nodes with same tag name: Use Indexing, Start-Index: [1]

//div[@id=’divId’]/descendant::a[1]

Can use x: instead of descendant::

//\*[@id='lga']/x:div

//label[text()='Seaside & Country Homes']/preceding-sibling::input[@type='checkbox']

//table/descendant::span/input[@type='checkbox']

To identify partial value of attribute (used in dynamic Xpath)

//\*[starts-with(@id,'div')]

//\*[contains(@id, 'div')]

One or more nodes with text Model

//ul[@class='featureList' and ./li[contains(.,'Model')]]

Using Two Attribute

Syntax: //tagName[@attributeName1=’attributeValue1’ and @attributeName2=’attributeValue2’ ]

//input[@id='radOnewayOrReturnTrip' and @value='O']

//\*[@id='radOnewayOrReturnTrip' and @value='O']

Usage

*driver*.findElement(By.xpath("//div[@id='ires']/descendant::a[1]"))

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

## **How to write cssSelectors:**

EX:

Html…

<div id=”divId” class=”divClassName” type=”divType”> someText </div>

<div id=”divId2”> </div>

<a href=”www.somewebpage.co.in” class=”aClass” > someLinkText </div>

To identify element using class name:- .ClassName

Syntax: tagName.ClassName OR .ClassName,

.divClassName

div.divClassName

To identify element using id:-

Syntax: tagName#idValue OR #idValue,

#divClass

div#divClass

To identify element using attribute:-

Syntax: tagName[attributeName=’attributeValue’] OR [attributeName=’attributeValue’]

[type=’divType’]

div[type=’divType’]

To identify element using Partial attribute value:-

For attribute value starts-With:

Syntax: tagName[attributeName^=’attributeValue’]

div[type^=’div’]

For attribute value ends-With:

Syntax: tagName[attributeName$=’attributeValue’]

div[type$=’Type’]

For attribute value has-Sub-String:

Syntax: tagName[attributeName\*=’attributeValue’]

div[type\*=’Ty’]

Usage:

*driver*.findElement(By.cssSelector(".divClass"));

*driver*.findElement(By.cssSelector("div#divClass"));

*driver*.findElement(By.cssSelector("#divClass"));

*driver*.findElement(By.cssSelector("div[type=’divType’]"));

*driver*.findElement(By.cssSelector("div[type^=’div’]"));

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

## **Key Press in TestNG**

WebElement.sendKeys(Keys.ALT);

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

## **Keys. Chord (sequence)**

WebElement.sendKeys(Keys.chord(Keys.ALT,Keys.CONTROL));

WebElement.sendKeys(Keys.*chord*(Keys.*CONTROL*, "a"));

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

## **Keys.Chord-Unicode sequence**

EX: WE.sendKeys(Keys.*chord*(("\u0915"), ("\u092E"), ("\u093E"), ("\u0932"),

Keys.*SPACE*, ("\u0939"), ("\u0948"), Keys.*SPACE*, ("\u092A"),

("\u0941"), ("\u0937"), ("\u094D"),("\u092A"), ("\u093E"),("\u0964")));

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

## **Actions**

ALT+SHIFT+T

Actions builder = new Actions(driver);

builder.keyDown(driver.findElement(By.*id*("main")), Keys.*ALT*).perform();

builder.keyDown(driver.findElement(By.*id*("main")), Keys.*SHIFT*).perform();

builder.sendKeys(driver.findElement(By.*id*("main")), "T").perform();

builder.keyUp(driver.findElement(By.*id*("main")), Keys.*SHIFT*).perform();

builder.keyUp(driver.findElement(By.*id*("main")), Keys.*ALT*).perform();

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

## **Mouse Over an element and Click another element after that:**

**if**(getWebElement(By.*linkText*(\_ProfileLinklinktext)).isDisplayed()){

//myClick is a function declared for click.

myClick(*driver*.findElement(By.*linkText*(\_ProfileLinklinktext)));

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

//Hover Element

WebElement we = *driver*.findElement(By.*linkText*(\_ProfileLinklinktext));

//mouse over and click other Element

action.moveToElement(we).moveToElement(*driver*.findElement(By.*xpath*(\_ProfileDDEditProfileXpath))).click().build().perform();

*driver*.manage().timeouts().pageLoadTimeout(15, TimeUnit.*SECONDS*);

}

OR

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

WebDriverWait wait = **new** WebDriverWait(*driver*, 20);

//Waiting for menu item to be clickable

wait.until(ExpectedConditions.*elementToBeClickable*(By.*linkText*(\_Profilelinktext)));

WebElement firstLink = *driver*.findElement(By.*linkText*(\_ProfileLinklinktext));

//Hover the menu item

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

//Waiting for the sub-menu item to be clickable - ERRORS OUT HERE

wait.until(ExpectedConditions.*elementToBeClickable*(By.*xpath*(\_ProfileDDEditXpath)));

//Getting the second link

WebElement secondLink = *driver*.findElement(By.*xpath*(\_ProfileDDEditXpath));

//Click on sub menu

actions.moveToElement(secondLink).click().build().perform();

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

## **Drag and Drop**

//Native events are required to be enabled.

🡪 driver.get("http://jqueryui.com/droppable/");

🡪 driver.switchTo().frame(0);

WebElement element = driver.findElement(By.name("source"));

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

(new Actions(driver)).dragAndDrop(element, target).perform();

OR

Actions builder = new Actions(driver);

builder.clickAndHold(WE1).moveToElement(WE2).release(WE2).build().perform();

OR

Actions builder = new Actions(driver);

builder.moveToElement(WE1).clickAndHold();

builder.moveToElement(WE2).click().perform();

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

## **Alert (pop up)**

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

//alert is present

System.*out*.println(alert.getText());

alert.sendKeys("//String");

alert.accept();

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

## **Disable TestStep:**

@Test(enabled = false)- will not run test case.

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

## **Groups**

@Test(groups = {"functional"}, expectedExceptions = ArithmeticException.class, dependsOnMethods = { "initEnvironmentTest" })

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

## **Scroll down**

while (!driver.findElement(By.id("smb")).isDisplayed()){

((JavascriptExecutor)driver).executeScript("window.scrollBy(0,250);");

Thread.sleep(200);

}

OR

while (!driver.findElement(By.*id*(("smb")).isDisplayed()){

(new Actions(driver)).sendKeys(Keys.*ARROW\_DOWN*).perform();

}

OR

//Scroll down to End Of Page

Actions actions = new Actions(driver);

actions.keyDown(Keys.*CONTROL*).sendKeys(Keys.*END*).perform();

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

**Scroll up**

Some Condition…

((JavascriptExecutor)driver).executeScript("window.scrollBy(0,-250);"); -----------------------------------------------------------------------------

//GroovyScript- for (i in 1..10) { println "Hello ${i}" }

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

## **Select Element**

WebElement WE =driver.findElement(By.*id*(read.getProps("FromlanguageMenuId")));

Select select = new Select(WE);

select.deselectAll();

select.selectByVisibleText("English");

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

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

## **Check Box**

if ( !driver.findElement(By.id("idOfTheElement").isSelected() )

{

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

}

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

## **Double Click**

Actions action = new Actions(driver);

action.doubleClick(driver.findElement(By.*name*("btnG")));

action.perform();

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

## **List**

List<WebElement> allTds=driver.findElements(By.id("unique"))  
for(WebElement eachTd : allTds)  
{  
     System.out.println(eachTd.getText());  
}

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

## **Grab the table**

WebElement table = driver.findElement(By.id("divListView"));

// Now get all the TR elements from the table

List<WebElement> allRows = table.findElements(By.tagName("tr"));

// And iterate over them, getting the cells

for (WebElement row : allRows) {

List<WebElement> cells = row.findElements(By.tagName("td"));

for (WebElement cell : cells) {

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

}

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

## **Switch driver to new browser window**

String parentWindow = driver.getWindowHandle();

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

for(String windowHandle : handles)

{

if(!windowHandle.equals(parentWindow))

{

driver.switchTo().window(windowHandle);

//code…

//Back to Parent Window

driver.close(); //closing child window

driver.switchTo().window(parentWindow);//back to PrntWindow

}

}

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

## **Array List, For-Each loop**

List<WebElement> link = driver.findElements(By.*cssSelector*("li.g"));

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

//For Each Loop

for (WebElement webElement : link) {

GooglearrayList.add(webElement.getText());

}

System.*out*.println(GooglearrayList.get(1));

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

}

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

## **Get element by index in List**

List<WebElement> list = driver.findElements(By.*cssSelector*("li.g"));

System.*out*.println(list.get(1).getText());

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

//To click element from List

WebElement element = list.get(0);

element.findElement(By.tagName("a")).click();

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

## **Context Click/ Right Click**

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

// copy Images link from Context Menu

WebElement element1 = driver.findElement(By.*linkText*("Images"));

Action contextClickCopy = builder

.contextClick(element1)

.sendKeys(Keys.*chord*(Keys.*ARROW\_DOWN*, Keys.*ARROW\_DOWN*,Keys.*ARROW\_DOWN*, Keys.*ARROW\_DOWN*,Keys.*ARROW\_DOWN*, Keys.*ARROW\_DOWN*, Keys.*ENTER*)).build();

contextClickCopy.perform();

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

## **Keyword driven testing:**

Purpose: Reusability, Easy maintenance

Keyword: define actions that drive or get information from application objects

Keyword Attributes: associated with a window object, like a list-box or a button

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Keywords** | **Argument 1** | **Argument 2** | **Argument 3** | **Argument 4** | **Argument 5** |
|  |  |  |  |  |  |
| **Hotel Reservation System Start** |  |  |  |  |  |
|  | **Country** | **City** | **Check-in Date** | **Checkout Date** | **Rate Type** |
| **Hotel Reservation Enter** | USA | California | &Date("mm/dd/yy", 5) | &Date("mm/dd/yy", 7) | AAA |
|  | **Hotel Name** |  |  |  |  |
| **Hotel Select** | Harbor View Palace |  |  |  |  |

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

## **Property Files**

@Test:

Instantiate ReadPropertiesFile:

ReadPropertiesFile Props = new ReadPropertiesFile();

Add Prorerty File:

Add New >Package, New > File. Save as someName.properties.

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

//Method to load and return propertry Value

//set property location->packagename/filename.

//Ex:**com.package/page.properties** as **com/package/page.properties**.

public class ReadPropertiesFile {

public Properties configProp = new Properties();

public String getProps(String string) {

InputStream in = this.getClass().getClassLoader().getResourceAsStream("org/intranet/intranetHomePage.properties");

try {

configProp.load(in);

}

catch (IOException e) {

e.printStackTrace();

}

return configProp.getProperty(string);

}

}

OR

public class ReadPropertiesFile {

Properties configProp = new Properties();

public ReadPropertiesFile(){

InputStream in = this.getClass().getClassLoader().getResourceAsStream("properties/Googlepage.properties");

try {

configProp.load(in);

}

catch (IOException e) {

e.printStackTrace();}

}

public String getProps(String string) {

return configProp.getProperty(string);

}

}

OR

public class ReadPropertiesFile {

Properties configProp = new Properties();

public ReadPropertiesFile(String PropertyPath) {

//to read property from Location ex: C:/User/data.properties

if (PropertyPath.contains(":/")|| PropertyPath.contains(":\\")) {

FileInputStream input = null;

try {

input = new FileInputStream(new File(PropertyPath));

configProp.load(input);

} catch (IOException e) {

e.printStackTrace();

}

}

//to read property from class Location(property in same project folder)

else {

InputStream in = this.getClass().getClassLoader()

.getResourceAsStream(PropertyPath);

try {

configProp.load(in);

} catch (IOException e) {

e.printStackTrace();

}

}

}

public String getValue(String string) {

return configProp.getProperty(string);

}

}

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

## **TestNG Sample**

package com.example.com.example;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.\*;

import org.openqa.selenium.firefox.FirefoxDriver;

public class SimpleSearchTest {

private WebDriver driver;

private String baseUrl;

@BeforeClass

public void setUp() throws Exception {

driver = new FirefoxDriver();

baseUrl = "http://google.com.br";

driver.manage().timeouts().implicitlyWait(3, TimeUnit.*SECONDS*);

}

@Test

public void foundGoogle() throws Exception {

driver.get(baseUrl);

//enlarge browser screen.

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

driver.findElement(By.*id*("gs\_tti0")).sendKeys("dfteste");

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

}

@Test

public void clickInTheResult() throws Exception {

//driver.findElement(By.xpath("//\*[@href='http://br.dir.groups.yahoo.com/group/DFTestes/']")).click();

}

@AfterClass

public void tearDown() throws Exception {

//driver.quit();

}

}

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

## **Get data from excel sheet (JXL.jar)**

//Need jxl.jar file in project build path configuration.

import java.io.File;

import java.util.Arrays;

import jxl.Sheet;

import jxl.Workbook;

public class ReadExcelSheet {

String XlsFilePath;

ReadExcelSheet(String xlFilePath) {

XlsFilePath = xlFilePath;

}

public String[][] getExcelData(String sheetName) {

String[][] tabArray = null;

try {

Workbook workbook = Workbook.*getWorkbook*(new File(XlsFilePath));

Sheet sheet = workbook.getSheet(sheetName);

int RowCount, ColCount, ci, cj;

RowCount = sheet.getRows();

ColCount = sheet.getColumns();

//Use String[RowCount-1][ColCount]to omit first row and read other row.

tabArray = new String[RowCount][ColCount];

ci = 0;

//Use i=1 to omit first row and read other row.

for (int i = 0; i < RowCount; i++, ci++) {

cj = 0;

for (int j = 0; j < ColCount; j++, cj++) {

tabArray[ci][cj] = sheet.getCell(j,i).getContents();

}

}

} catch (Exception e) {

System.*out*.println("error in getTableArray()");

}

return (tabArray);

}

}

Declarations in test script

ReadExcelSheet readExcel = new ReadExcelSheet ("c:/User/NewExcel.xls")

@DataProvider(name = "DP1")

public Object[][] createData1() throws Exception{

Object[][] retObjArr = readExcel. getExcelData("Sheet1");

return(retObjArr);

}

@Test(dataProvider = "DP1")

public void f(String low, String High) throws InterruptedException {

driver.findElement(By.*name*("q")).clear();

driver.findElement(By.*name*("q")).sendKeys(low);

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

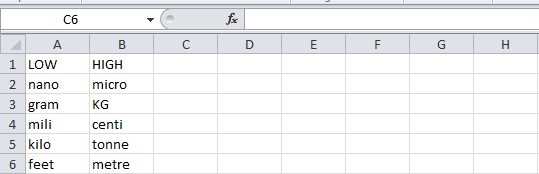
driver.findElement(By.*name*("q")).clear();

driver.findElement(By.*name*("q")).sendKeys(High);

}

SnapShot,

//@test should have the same number of parameters as that of column in Excel



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

\*Print array : System.*out*.println(Arrays.*deepToString*(ExcelData));

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

## **Get data from excel sheet(Apache POI):**

//First row Omited as given in the comments inside code,

//XSSF format will read Excel.xlsx format file

//HSSF format will read Excel.xls format file

import java.io.File;

import java.io.FileInputStream;

import java.util.Arrays;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

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.xssf.usermodel.XSSFWorkbook;

public class ReadExcelSheetPOI {

String XlsFilePath;

public ReadExcelSheetPOI(String xlFilePath) {

XlsFilePath = xlFilePath;

}

public Object[][] getExcelData(String SheetName) throws Exception {

File excel = new File(XlsFilePath);

FileInputStream fis = new FileInputStream(excel);

Object[][] data = null;

Workbook wb = null;

if (XlsFilePath.contains(".xlsx")) {

wb = new XSSFWorkbook(fis);

} else if (XlsFilePath.contains(".xls")) {

wb = new HSSFWorkbook(fis);

}

try {

Sheet ws = wb.getSheet(SheetName);

int rowNum = ws.getLastRowNum() + 1;

int colNum = ws.getRow(0).getLastCellNum();

// rowNum-1 is used to remove 1st row count

data = new String[rowNum - 1][colNum];

int Ci = 0;

// i=1 represent that it reads from second row.

for (int i = 1; i < rowNum; i++, Ci++) {

int Cj = 0;

Row row = ws.getRow(i);

for (int j = 0; j < colNum; j++, Cj++) {

Cell cell = row.getCell(j);

String value = *cellToString*(cell);

data[Ci][Cj] = value;

// System.out.println("The value is" + value);

}

}

}catch (Exception e) {

System.*out*.println("error in getExcelData()");

}

return data;

}

public static String cellToString(Cell cell) {

int type;

Object result;

// getCellType will return integer value 0 to 5, depends on cell Type,

// which is used to get cell Value in Switch Case.

type = cell.getCellType();

switch (type) {

case 0:

//can round off if not needed ‘.0’ in end

//double num = cell.getNumericCellValue();

//result = Math.round(num);

result = cell.getNumericCellValue();

break;

case 1:

result = cell.getStringCellValue();

break;

case 2:

result = cell.getCellFormula();

break;

case 3:

result = "";

break;

case 4:

result = cell.getBooleanCellValue();

break;

case 5:

result = cell.getErrorCellValue();

break;

default:

throw new RuntimeException(

"There are no support for this type of cell");

}

return result.toString();

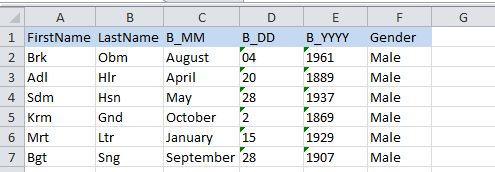
}

//can add main class here to virify it in stand alone.

}

SnapShot,

//@test should have the same number of parameters as that of column in Excel



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

We can add Main class inside the above class to verify its execution:

public static void main(String[] args) throws Exception {

ReadExcelSheetPOI read = new ReadExcelSheetPOI(

"D:/ExcelFile/GoogleSignUp.xlsx");

Object[][] ExcelData = read.getExcelData("Sheet1");

System.*out*.println(Arrays.*deepToString*(ExcelData));

}

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

\*Get File Path:

excelFile=ClassName.**class**.getClassLoader().getResource("com/pack/TestData.xls").getPath();

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

## **Read Excel Sheet and Return an Object [ ] iterator (Iterator<Object [ ]>).**

import java.io.File;

import java.io.FileInputStream;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Iterator;

import java.util.List;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

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.xssf.usermodel.XSSFWorkbook;

public class ReadExcelSheetPOI {

String XlsFilePath;

public ReadExcelSheetPOI(String xlFilePath) {

XlsFilePath = xlFilePath;

}

public Iterator<Object[]> getExcelData(String SheetName) throws Exception {

File excel = new File(XlsFilePath);

FileInputStream fis = new FileInputStream(excel);

List<Object[]> datas = new ArrayList<Object[]>();

Workbook wb = null;

if (XlsFilePath.contains(".xlsx")) {

wb = new XSSFWorkbook(fis);

} else if (XlsFilePath.contains(".xls")) {

wb = new HSSFWorkbook(fis);

}

String[] data = null;

try {

Sheet ws = wb.getSheet(SheetName);

int rowNum = ws.getLastRowNum() + 1;

int colNum = ws.getRow(0).getLastCellNum();

// rowNum-1 is used to remove 1st row count

int Ci = 0;

// i=1 represent that it reads from second row.

for (int i = 1; i < rowNum; i++, Ci++) {

data = new String[colNum];

int Cj = 0;

Row row = ws.getRow(i);

for (int j = 0; j < colNum; j++, Cj++) {

Cell cell = row.getCell(j);

String value = *cellToString*(cell);

data[Cj] = value;

//System.out.println("The value is" + value);

}

datas.add(data);

}

}catch (Exception e) {

System.*out*.println("error in getExcelData()");

}

return datas.iterator();

}

public static String cellToString(Cell cell) {

int type;

Object result;

// getCellType will return integer value 0 to 5, depends on cell Type,

// which is used to get cell Value in Switch Case.

type = cell.getCellType();

switch (type) {

case 0:

double num = cell.getNumericCellValue();

result = Math.*round*(num);

break;

case 1:

result = cell.getStringCellValue();

break;

case 2:

result = cell.getCellFormula();

break;

case 3:

result = "";

break;

case 4:

result = cell.getBooleanCellValue();

break;

case 5:

result = cell.getErrorCellValue();

break;

default:

throw new RuntimeException(

"There are no support for this type of cell");

}

return result.toString();

}

public static void main(String[] args) throws Exception {

ReadExcelSheetPOI read = new ReadExcelSheetPOI("E:/SampleExcel.xlsx");

Iterator<Object[]> ExcelData = read.getExcelData("Sheet1");

while(ExcelData.hasNext()) {

Object[] element = ExcelData.next();

System.*out*.print(Arrays.*deepToString*(element));

}

}

}

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

## **Read Data From Excel Sheet and Print Type Of Data.**

package exl.read;

import java.io.File;

import java.io.IOException;

import jxl.Cell;

import jxl.CellType;

import jxl.Sheet;

import jxl.Workbook;

import jxl.read.biff.BiffException;

public class ReadExcel {

private String inputFile;

public void setInputFile(String inputFile) {

this.inputFile = inputFile;

}

public void read() throws IOException {

File inputWorkbook = new File(inputFile);

Workbook w;

try {

w = Workbook.getWorkbook(inputWorkbook);

// Get the first sheet

Sheet sheet = w.getSheet(0);

// Loop over first 10 column and lines

for (int j = 0; j < sheet.getColumns(); j++) {

for (int i = 0; i < sheet.getRows(); i++) {

Cell cell = sheet.getCell(j, i);

CellType type = cell.getType();

if (type == CellType.LABEL) {

System.out.println("*I got a label* "+ cell.getContents());

}

if (type == CellType.NUMBER) {

System.out.println("*I got a number* "+ cell.getContents());

}

}

System.out.println("*Next Column*");

}

} catch (BiffException e) {

e.printStackTrace();

}

}

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

ReadExcel test = new ReadExcel();

test.setInputFile("*C:/Users/workspace/Intranet/src/excel/NewExcel.xls*");

test.read();

}

}

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

## **Read Excel (Apache POI):**

package com.maven.pomframework1;

import java.io.File;

import java.io.FileInputStream;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Date;

import java.util.Iterator;

import java.util.List;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;

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.xssf.usermodel.XSSFWorkbook;

public class ReadExcelPoi {

String XlsFilePath;

Workbook workbook;

Sheet sheet;

Row row;

Cell cell;

String value;

public ReadExcelPoi(String xlFilePath) {

XlsFilePath = xlFilePath;

try {

File excel = new File(XlsFilePath);

FileInputStream fis = new FileInputStream(excel);

if (XlsFilePath.contains(".xlsx")) {

workbook = new XSSFWorkbook(fis);

} else if (XlsFilePath.contains(".xls")) {

workbook = new HSSFWorkbook(fis);

}

} catch (Exception e) {

e.printStackTrace();

}

}

// Read Excel sheet

@SuppressWarnings("null")

public Iterator<Object[]> getExcelData(String SheetName) throws Exception {

List<Object[]> datas = new ArrayList<Object[]>();

try {

sheet = workbook.getSheet(SheetName);

int rowNum = sheet.getLastRowNum() + 1;

int colNum = sheet.getRow(0).getLastCellNum();

String[] data = null;

for (int i = 1; i < rowNum; i++) {

data = new String[colNum];

int Cj = 0;

row = sheet.getRow(i);

for (int j = 0; j < colNum; j++, Cj++) {

cell = row.getCell(j);

if (cell == null) {

data[Cj] = "";

} else {

value = *cellToString*(cell);

data[Cj] = value;

}

// System.out.println("The value is" + value);

}

datas.add(data);

}

} catch (Exception e) {

System.*out*.println("error in getExcelData()");

}

return datas.iterator();

}

// get cell data by column\_Name and row\_Number

public String getCellData(String sheetName, String colName, int rowNum) {

try {

if (rowNum <= 0)

return "";

int index = workbook.getSheetIndex(sheetName);

int col\_Num = -1;

if (index == -1)

return "";

sheet = workbook.getSheetAt(index);

row = sheet.getRow(0);

for (int i = 0; i < row.getLastCellNum(); i++) {

// System.out.println(row.getCell(i).getStringCellValue().trim());

if (row.getCell(i).getStringCellValue().trim().equals(colName.trim())) {

col\_Num = i;

break;

}

}

if (col\_Num == -1)

return "";

row = sheet.getRow(rowNum - 1);

if (row == null)

return "";

cell = row.getCell(col\_Num);

if (cell == null) {

value = "";

} else {

value = *cellToString*(cell);

}

return value;

} catch (Exception e) {

e.printStackTrace();

return "row " + rowNum + " or column " + colName

+ " does not exist in xls";

}

}

//Get iterator for selected columns

public Iterator<Object[]> getSortedCellData(String sheetName,

String colNamesSeperatedByComma) throws Exception {

List<Object[]> datas = new ArrayList<Object[]>();

String[] columns = colNamesSeperatedByComma.split(",");

ArrayList<Integer> colIndex = new ArrayList<Integer>();

for (String column : columns) {

int i = getColumnIndex(sheetName, column);

colIndex.add(i);

}

try {

sheet = workbook.getSheet(sheetName);

int rowNum = sheet.getLastRowNum() + 1;

for (int i = 1; i < rowNum; i++) {

String[] data = new String[colIndex.size()];

int j = 0;

row = sheet.getRow(i);

for (Integer col : colIndex) {

cell = row.getCell(col);

if (cell == null) {

data[j] = "";

} else {

value = *cellToString*(cell);

data[j] = value;

}

j++;

// System.out.println("The value is" + value);

}

datas.add(data);

}

return datas.iterator();

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

@SuppressWarnings("null")

public int getColumnIndex(String sheetName, String colName)

throws Exception {

try {

int col\_Num = -1;

int index = workbook.getSheetIndex(sheetName);

sheet = workbook.getSheetAt(index);

row = sheet.getRow(0);

for (int i = 0; i < row.getLastCellNum(); i++) {

//System.out.println(row.getCell(i).getStringCellValue().trim());

if (row.getCell(i).getStringCellValue().trim()

.equals(colName.trim())) {

col\_Num = i;

break;

}

}

if (col\_Num == -1) {

return (Integer) null;

}

return col\_Num;

} catch (Exception e) {

e.printStackTrace();

return (Integer) null;

}

}

public static String cellToString(Cell cell) {

int type;

Object result;

// getCellType will return integer value 0 to 5, depends on cell Type,

// which is used to get cell Value in Switch Case.

type = cell.getCellType();

switch (type) {

case 0:

double num = cell.getNumericCellValue();

result = Math.*round*(num);

break;

case 1:

result = cell.getStringCellValue();

break;

case 2:

result = cell.getCellFormula();

break;

case 3:

result = "";

break;

case 4:

result = cell.getBooleanCellValue();

break;

case 5:

result = cell.getErrorCellValue();

break;

default:

throw new RuntimeException(

"There are no support for this type of cell");

}

return result.toString();

}

public static void main(String[] args) throws Exception {

long lStartTime1 = new Date().getTime();

ReadExcelPoi read = new ReadExcelPoi(

"E:/ExcelFiles/Sample/SampleExcel\_6X10.xlsx");

long lEndTime1 = new Date().getTime();

long difference1 = lEndTime1 - lStartTime1;

System.*out*.println("1:Elapsed milliseconds: " + difference1);

long lStartTime2 = new Date().getTime();

// some tasks

Iterator<Object[]> ExcelData1 = read.getExcelData("Sheet1");

String ExcelData2 = read.getCellData("Sheet1", "Header1", 2);

Iterator<Object[]> ExcelData = read.getSortedCellData("Sheet1","Header1,Header6");

long lEndTime2 = new Date().getTime();

long difference2 = lEndTime2 - lStartTime2;

System.*out*.println("2:Elapsed milliseconds: " + difference2);

System.*out*.println(ExcelData2);

//System.out.println(Arrays.deepToString(ExcelData1));

while(ExcelData1.hasNext()){

Object[] element = ExcelData1.next();

System.*out*.println(Arrays.*deepToString*(element));

}

}

}

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

## **CustomLogger**

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.testng.ITestContext;

import org.testng.ITestListener;

import org.testng.ITestResult;

public class CustomLoggerClass implements ITestListener{

//Called when the test-method execution starts

@Override

public void onTestStart(ITestResult result) {

System.*out*.println("Test method started: "+ result.getName()+ " and time is: "+getCurrentTime());

}

//Called when the test-method execution is a success

@Override

public void onTestSuccess(ITestResult result) {

System.*out*.println("Test method success: "+ result.getName()+ " and time is: "+getCurrentTime())

}

//Called when the test-method execution fails

@Override

public void onTestFailure(ITestResult result) {

System.*out*.println("Test method failed: "+ result.getName()+ " and time is: "+getCurrentTime());

}

//Called when the test-method is skipped

@Override

public void onTestSkipped(ITestResult result) {

System.*out*.println("Test method skipped: "+ result.getName()+ " and time is: "+getCurrentTime());

}

//Called when the test-method fails within success percentage

@Override

public void onTestFailedButWithinSuccessPercentage(ITestResult result) {

// Leaving blank

}

//Called when the test in xml suite starts

@Override

public void onStart(ITestContext context) {

System.*out*.println("Test in a suite started: "+ context.getName()+ " and time is: "+getCurrentTime());

}

//Called when the test in xml suite finishes

@Override

public void onFinish(ITestContext context) {

System.*out*.println("Test in a suite finished: "+ context.getName()+ " and time is: "+getCurrentTime());

}

//Returns the current time when the method is called

public String getCurrentTime(){

DateFormat dateFormat = new SimpleDateFormat("HH:mm:ss:SSS");

Date dt = new Date();

return dateFormat.format(dt);

}

}

//Add listner to testNG.xml

<suite name=*"Suite"* parallel=*"none"*>

<listeners>

<listener class-name=*"customlogger.CustomLoggerClass"* />

</listeners>

<test name=*"Test"*>

//Reporting

Listener impliments *org.testng.ITestListener*

Reporter impliments *org.testng.IReporter*

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

## **Iterator Same As For Each Loop… (use only when needed to remove data)**

for(Iterator<String> i = someList.iterator(); i.hasNext(); ) {

String item = i.next();

System.out.println(item);

//can use i.remove();

}

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

//This code is **untested** and may be **unsupported** . *Use at your own risk!*

/\*\* This script finds the real position,

\* so if you resize the page and run the script again,

\* it points to the correct new position of the element.

\*/

function findPos(obj){

var curleft = 0;

var curtop = 0;

if (obj.offsetParent) {

do {

curleft += obj.offsetLeft;

curtop += obj.offsetTop;

} while (obj = obj.offsetParent);

return {X:curleft,Y:curtop};

}

}

OR

int xOffset, yOffset;

Actions actions = new Actions(this.drivy);

Action move = actions.moveByOffset(xOffset, yOffset).build();

move.perform();

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

## **JAVA:**

Get ip address of system:

import java.net.InetAddress;

class watsMyIP {

public static void main(String args[]) throws Exception {

InetAddress Address = InetAddress.*getLocalHost*();

//Address = TESTING-123/10.10.214.12

String aa = Address.toString();

String[] ary = aa.split("/");

System.*out*.println(ary[1]);

}

}

Get ip adress of website,

String address = InetAddress.*getByName*("www.google.com").toString().split("/")[1];

String hostname = InetAddress.*getByName*("www.google.com").toString().split("/")[0];

System.*out*.println(address + " " + hostname);

**Time Elapsed:**

long lStartTime = new Date().getTime();

//some task…

//ReadExcelSheetPOI read = new ReadExcelSheetPOI("E:/ExcelFiles/Sample.xlsx");

long lEndTime = new Date().getTime();

long Timedifference = lEndTime - lStartTime;

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

## **Creating Executable Jar File in TestNG:**

1. Add the below main class to any package,

import org.testng.TestListenerAdapter;

import org.testng.TestNG;

public class Main {

@SuppressWarnings("rawtypes")

public static void main(String[] args) {

TestListenerAdapter tla = new TestListenerAdapter();

TestNG testng = new TestNG();

//add all the testScript class into Class[]

Class[] testclass = new Class[] {com.maven.test.LoginTest.class, com.maven.test.HomePageTest.class,

com.maven.test.LogoutTest.class};

testng.setTestClasses(testclass);

testng.addListener(tla);

testng.run();

}

}

1. Add the test classes which are used in testNG.xml into Class[] above.
2. Run program as java application (will execute TestNG.xml file).
3. To create a runnable jar file:
4. Export package ,
5. As Java--runnable jar file.
6. In launch Configuration select the main class (above class name).
7. Select Destination and enter jar file name
8. Press finish.
9. Done!
10. Just double click it.

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

## **Executing Jar File from Command window:**

1. Open Command Window in the same location (Directory) where your Jar file is located(SHIFT+RIGHTCLICK > Open command window here )
2. Run command -> java –jar jarFileName.jar

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

## **NoClassDefFound Error in maven POM.xml:**

Due to apache POI dependency: add this dependency

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>3.9</version>

<exclusions>

<exclusion>

<artifactId>xml-apis</artifactId>

<groupId>xml-apis</groupId>

</exclusion>

</exclusions>

</dependency>

OR

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>3.10</version>

</dependency>

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

## **reflect.InvocationTargetException**

Check “Cause By” part of error, error itself will give you enough information.

Generally occur when path for fetching/loading file is incomplete/misspelled.

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

## **Save credentials in IE (Auto Log In):**

Tools>Internet Options>Security>Local Intranet>Sites>Advanced>ADD

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

## **Adding Custom Jar file to Maven Dependency**

Add file to local repository by running cmd:

mvn install:install-file -Dfile=C:\Software\Decrypt.jar -DgroupId=common.utils -DartifactId=Decrypt -Dversion=1.0 -Dpackaging=jar

* Package inside jar (groupId) - common.utils
* Java main class name (artifactId) – Decrypt

Also you can directly add your jar file to local repository folder (if you know path).

Add dependency:

<dependency>

<groupId>common.utils</groupId>

<artifactId>Decrypt</artifactId>

<version>1.0</version>

</dependency>

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

JavascriptExecutor js = (JavascriptExecutor) *driver*;

js.executeScript("window.promptResponse = prompt(\"Please enter Username\",”defaultName”);");

//Handle javascript prompt box and get value.

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

**try** {

Thread.*sleep*(15000);

} **catch** (Exception e)

{

System.*out*.println("Cannot sleep because of headache");

}

alert.accept();

String ret = (String) js.executeScript("return window.promptResponse;");

*driver*.findElement(By.*id*("session\_key-login")).sendKeys(ret);

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

"All the desirable things in life are either illegal, expensive, fattening or in love with someone else!"