**Handling Excel Files Using Apache POI In Selenium WebDriver**

Selenium supports only Web browser automation. We need to get the help of third party API like Apache POI to handle (read and write) excel files using Selenium WebDriver.

**What is Apache POI?**

Apache POI is an open source library developed and distributed by Apache Software Foundation to design or modify Microsoft Office files using Java program. It is a popular API that allows to work around excel files using Java Programs. In short, you can read and write MS Excel files using Java. Apache POI is your Java Excel solution.

You’d use HSSF if you needed to read or write an Excel file using Java (XLS). You’d use XSSF if you need to read or write an OOXML Excel file using Java (XLSX). It has many predefined methods, classes, and interfaces.

List of different Java Interfaces and classes in POI for reading **XLS** and **XLSX** file

**Workbook:** HSSFWorkbook and XSSFWorkbook classes implement this interface  
**HSSFWorkbook:** It is a class represents XLS file  
**XSSFWorkbook:** It is a class represents XLSX file  
**Sheet:** HSSFWorkbook and XSSFWorkbook classes implement this interface  
**HSSFSheet:** It is a class represents a sheet in a XLS file  
**XSSFSheet:** It is a class represents a sheet in a XLSX file  
**Row:** HSSFWorkbook and XSSFWorkbook classes implement this interface  
**HSSFRow:** It is a class represents a row in a sheet of XLS file  
**XSSFRow:** It is a class represents a row in a sheet of XLSX file  
**Cell:** HSSFWorkbook and XSSFWorkbook classes implement this interface  
**HSSFCell:** It is a class represents a cell in a row of XLS file  
**XSSFCell:** It is a class represents a cell in a row of XLSX file

We do handle excel files using Apache POI in Selenium WebDriver very often. Mainly coming to Automation framework (Data Driven Framework), we call the data from excel file and write the data in the excel file. Below links give an idea about Read/Write excel files using Selenium WebDriver.

**How to Read Excel Files Using Apache POI In Selenium WebDriver**

Let’s see how to Read excel files using Apache POI in Selenium WebDriver:

**Step 1**– Download Apache POI jar file

**Step 2**– Add download jar files to your project build path.

Select Project and Right click on the Project – Go to ‘Build path’ – Go to ‘Configure build path’ – Click on ‘lib’ section – Add external jar  
  
Below mentioned script shows how to read excel files in Selenium using Apache POI.

import java.io.FileInputStream;  
import java.io.IOException;  
  
import org.apache.poi.xssf.usermodel.XSSFRow;  
import org.apache.poi.xssf.usermodel.XSSFSheet;  
import org.apache.poi.xssf.usermodel.XSSFWorkbook;  
  
public class ReadingExcel {  
  
 public static void main(String[] args) throws IOException {  
  
 FileInputStream file = new FileInputStream("C://SeleniumPractice/data3.xlsx");  
 XSSFWorkbook workbook = new XSSFWorkbook(file);  
 XSSFSheet sheet = workbook.getSheet("Sheet1");  
  
 int rownum = sheet.getLastRowNum(); // returns number of rows present in excel sheet  
 int colcount = sheet.getRow(0).getLastCellNum(); // returns number of cells present in a row  
  
 for (int i = 0; i < rownum; i++) {  
 XSSFRow row = sheet.getRow(i);  
  
 for (int j = 0; j < colcount; j++) {  
 String value = row.getCell(j).toString(); // reading the data from cell  
 System.out.print(value + " ");  
 }  
  
 System.out.println();  
  
 }  
 }  
}

**How to Write Excel Files Using Apache POI In Selenium WebDriver**

Let’s see how to Read excel files using Apache POI in Selenium WebDriver:

**Step 1**– Download Apache POI jar file

**Step 2**– Add download jar files to Project build path(Follow the below navigation).

Select Project and Right click on the Project – Go to ‘Build path’ – Go to ‘Configure build path’ – Click on ‘lib’ section – Add external jar

Below mentioned code shows how to write data into excel file in selenium sing Apache POI.

import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
import org.apache.poi.xssf.usermodel.XSSFRow;  
import org.apache.poi.xssf.usermodel.XSSFSheet;  
import org.apache.poi.xssf.usermodel.XSSFWorkbook;  
  
public class WritingDataintoExcel {  
  
 public static void main(String[] args) throws IOException {  
  
 FileOutputStream file = new FileOutputStream("C://SeleniumPractice//testing.xlsx");  
  
 XSSFWorkbook workbook = new XSSFWorkbook();  
 XSSFSheet sheet = workbook.createSheet("data");  
  
 for (int i = 0; i <= 5; i++) {  
 XSSFRow row = sheet.createRow(i);  
 for (int j = 0; j < 3; j++) {  
 row.createCell(j).setCellValue("abcdef");  
 }  
 }  
  
 workbook.write(file);  
 System.out.println("writing excel is completed");  
  
 }  
  
}