File IO is a critical part of any software process. We frequently create a file, open it & update something or delete it in our Computers. Same is the case with Selenium Automation. We need a process to manipulate files with Selenium.

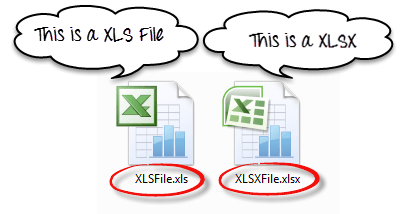
Java provides us different classes for File Manipulation with Selenium. In this tutorial, we are going to learn how can we read and write on [Excel](https://www.guru99.com/excel-tutorials.html) file with the help of[Java](https://www.guru99.com/java-tutorial.html)IO package and[Apache](https://www.guru99.com/apache.html)POI library.

**Apache POI in Selenium**

The **Apache POI in Selenium** is a widely used API for selenium data driven testing. It is a POI library written in Java that gives users an API for manipulating Microsoft documents like .xls and .xlsx. Users can easily create, modify and read/write into excel files. POI stands for “Poor Obfuscation Implementation.”

## Exporting Excel

### How to handle excel file using POI (Maven POM Dependency)



To Read and Write Excel file in Java, Apache provides a very famous library POI. This library is capable enough to read and write both**XLS** and**XLSX** file format of Excel.

To read**XLS** files, an**HSSF** implementation is provided by POI library.

To read**XLSX, XSSF** implementation of**POI library** will be the choice. Let’s study these implementations in detail.

If you are using Maven in your project, there are two Maven dependencies-

<dependency>

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

<artifactId>poi</artifactId>

<version>4.1.2</version>

</dependency>

<dependency>

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

<artifactId>poi-ooxml</artifactId>

<version>4.1.2</version>

</dependency>

Prerequisites

1. Create one excel file with data inside it, generally with new extension of xlsx. In case, you have xls version file then, convert it into new form.
2. Integrate maven dependencies given above.

**Note – Close the file after creating/editing.**

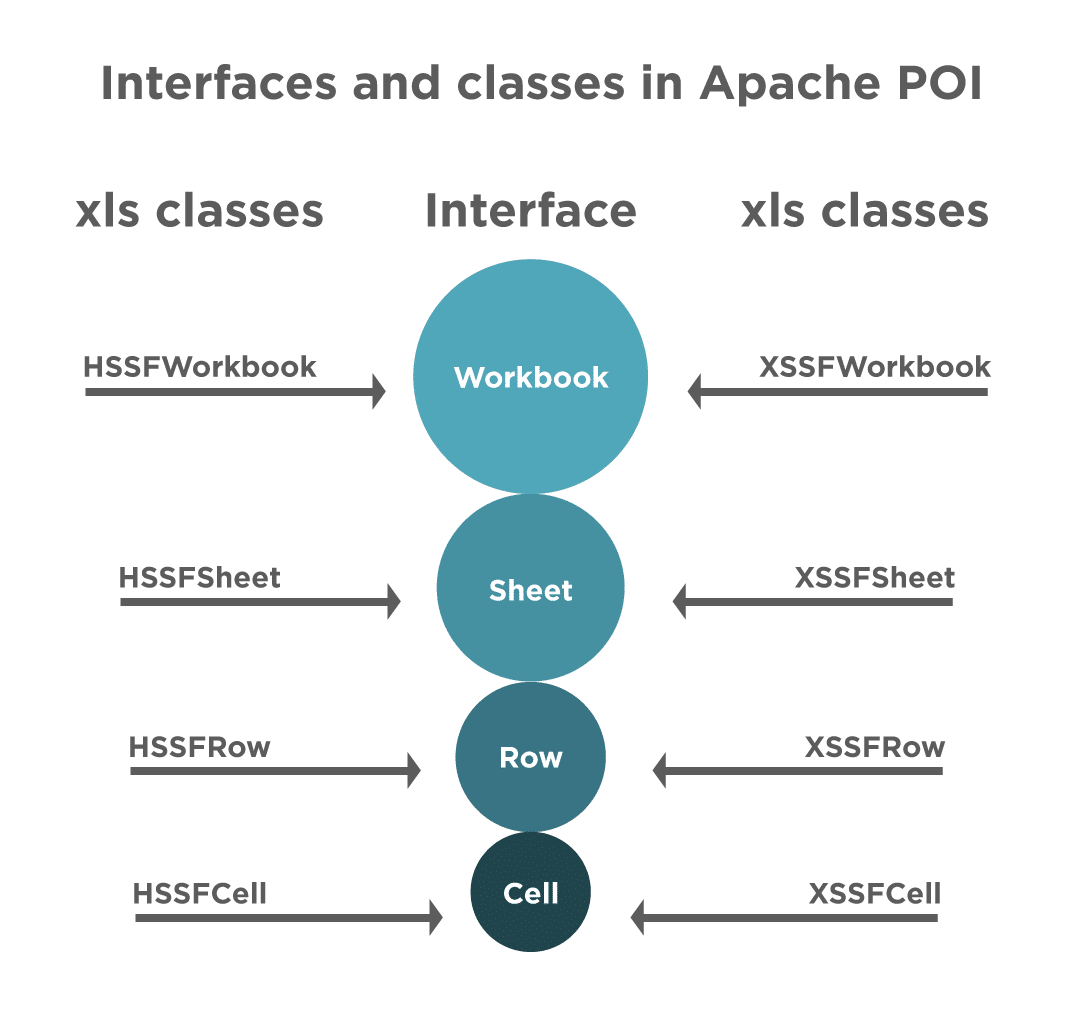
Following is a list of different Java Interfaces and classes in **POI** for reading **XLS** and **XLSX** file-

* **Workbook**: XSSFWorkbook and HSSFWorkbook classes implement this interface.
* **XSSFWorkbook**: Is a class representation of XLSX file.
* **HSSFWorkbook**: Is a class representation of XLS file.
* **Sheet**: XSSFSheet and HSSFSheet classes implement this interface.
* **XSSFSheet**: Is a class representing a sheet in an XLSX file.
* **HSSFSheet**: Is a class representing a sheet in an XLS file.
* **Row**: XSSFRow and HSSFRow classes implement this interface.
* **XSSFRow**: Is a class representing a row in the sheet of XLSX file.
* **HSSFRow**: Is a class representing a row in the sheet of XLS file.
* **Cell**: XSSFCell and HSSFCell classes implement this interface.
* **XSSFCell**: Is a class representing a cell in a row of XLSX file.
* **HSSFCell:** Is a class representing a cell in a row of XLS file.

***Apache POI*** uses certain terms to work with *Microsoft Excel.* Let's get familiar with these before we go into the details of the code.

| ***Term*** | ***Details*** |
| --- | --- |
| ***Workbook*** | *A****workbook****represents a Microsoft Excel file. It can be used for creating and maintaining the spreadsheet. A workbook may contain many****sheets****.* |
| ***Sheet*** | *A****sheet****refers to a page in a Microsoft Excel file that contains the number of****rows****and****columns****.* |
| ***Row*** | *A row represents a collection of****cells,****which is used to represent a row in the spreadsheet.* |
| ***Cell*** | *A****cell****is indicated by a row and column combination. Data entered by a user is stored in a cell. Data can be of the type such as string, numeric value, or formula.* |

The below image clearly depicts the structure and how the classes and interfaces are aligned in *Apache POI.*



### ***How to manage Excel workbooks pragmatically?***

Apache POI provides various interfaces and classes that help us to work with Excel. It provides a ***"Workbook "*** interface to maintain Excel Workbooks. There are certain classes that implement this interface and we use these classes to create, modify, read, and write data in Excel files. The two mainly used  classes for managing Excel Workbooks are:

* ***HSSFWorkbook-*** These class methods are used to read/write data to Microsoft Excel file in ***.xls*** format. It is compatible with MS-Office versions 97–2003.
* ***XSSFWorkbook-*** These class methods are used to read-write data to Microsoft Excel in ***.xls*** or ***.xlsx*** format. It is compatible with MS-Office versions 2007 or later.

### ***How to manage Excel sheets programmatically?***

There is another interface, ***"Sheet "***, which we use to create a sheet in the Workbook. There are two classes that used to work with sheets, same as we have for Workbook Interface:

* ***HSSFSheet -*** This class is used to create a new sheet in the HSSFWorkbook, ie, the older format of Excel.
* ***XSSFSheet -*** This class is used to create a new sheet in the XSSFWorkbook., ie, the new format of Excel

### ***How to manage Excel rows pragmatically?***

The ***Row*** interface provides us with the ability to work with rows in the Excel sheet. Below two classes implement this interface:

* ***HSSFRow*** - This represents a row in the HSSFSheet.
* ***XSSFRow*** - This represents a row in the XSSFSheet.

### ***How to manage Excel cells?***

The ***Cell*** interface helps us in accessing the cells of a particular row. There are two classes that implement this interface and we can use for reading/writing data into the cell:

* ***HSSFCell*** - We use it to work with cells of HSSFRow.
* ***XSSFCell*** - We use it to work with cells of XSSFRow