

# POI – CORE CLASSES

This chapter explains a few classes and methods under the Apache POI API that are critical to work on Excel files using Java programs.

## Workbook

This is the super-interface of all classes that create or maintain Excel workbooks. It belongs to the **org.apache.poi.ss.usermodel** package. The two classes that implement this interface are as follows:

- **HSSFWorkbook** : This class has methods to read and write Microsoft Excel files in .xls format. It is compatible with MS-Office versions 97-2003.
- **XSSFWorkbook** : This class has methods to read and write Microsoft Excel and OpenOffice xml files in .xls or .xlsx format. It is compatible with MS-Office versions 2007 or later.

## HSSFWorkbook

It is a high-level class under the org.apache.poi.hssf.usermodel package. It implements the Workbook interface and is used for Excel files in .xls format. Listed below are some of the methods and constructors under this class.

### Class Constructors

#### S.No. Constructor and Description

1

##### **HSSFWorkbook**

Creates a new HSSFWorkbook object from scratch.

2

##### **HSSFWorkbook***DirectoryNode directory, boolean preserveNodes*

Creates a new HSSFWorkbook object inside a specific directory.

3

##### **HSSFWorkbook***DirectoryNode directory, POIFSFileSystem fs, boolean preserveNodes*

Given a POIFSFileSystem object and a specific directory within it, it creates an HSSFWorkbook object to read a specified workbook.

4

##### **HSSFWorkbook***java.io.InputStream*

Creates a new HSSFWorkbook object using an input stream.

5

##### **HSSFWorkbook***java.io.InputStream, boolean preserveNodes*

Constructs a POI file system around your input stream.

6

##### **HSSFWorkbook***POIFSFileSystem*

Constructs a new HSSFWorkbook object using a POIFSFileSystem object.

## **HSSFWorkbookPOIFSFileSystem, booleanpreserveNodes**

Given a POIFSFileSystem object, it creates a new HSSFWorkbook object to read a specified workbook.

The frequently used parameters inside these constructors are:

- **directory** : It is the POI filesystem directory to process from.
- **fs** : It is the POI filesystem that contains the workbook stream.
- **preservenodes** : This is an optional parameter that decides whether to preserve other nodes like macros. It consumes a lot of memory as it stores all the POIFileSystem in memory if set.

**Note** : The HSSFWorkbook class contains a number of methods; however they are compatible with xls format only. In this tutorial, the focus is on the latest version of Excel file formats. Hence, the class methods of HSSFWorkbook are not listed here. If you require these class methods, then refer POI-HSSFWorkbook class API at

<https://poi.apache.org/apidocs/org/apache/poi/hssf/usermodel/HSSFWorkbook.html>.

## **XSSFWorkbook**

It is a class that is used to represent both high and low level Excel file formats. It belongs to the org.apache.xssf.usermodel package and implements the Workbook interface. Listed below are the methods and constructors under this class.

### **Class Constructors**

#### **S.No. Constructor and Description**

1

#### **XSSFWorkbook**

Creates a new XSSFworkbook object from scratch.

2

#### **XSSFWorkbookjava.io.Filefile**

Constructs an XSSFWorkbook object from a given file.

3

#### **XSSFWorkbookjava.io.InputStreamis**

Constructs an XSSFWorkbook object, by buffering the whole input stream into memory and then opening an OPCPackage object for it.

4

#### **XSSFWorkbookjava.lang.Stringpath**

Constructs an XSSFWorkbook object given the full path of a file.

## **Class Methods**

#### **S.No. Method and Description**

1

## **createSheet**

Creates an XSSFSheet for this workbook, adds it to the sheets, and returns the high level representation.

2

### **createSheet***java.lang.String sheetname*

Creates a new sheet for this Workbook and returns the high level representation.

3

## **createFont**

Creates a new font and adds it to the workbook's font table.

4

## **createCellStyle**

Creates a new XSSFCellStyle and adds it to the workbook's style table.

5

## **createFont**

Creates a new font and adds it to the workbook's font table.

6

## **setPrintArea***int sheetIndex, int startColumn, int endColumn, int startRow, int endRow*

Sets the print area of a given sheet as per the specified parameters.

For the remaining methods of this class, refer the complete API document at:  
<http://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFWorkbook.html> for the complete list of methods.

## **Sheet**

Sheet is an interface under the org.apache.poi.ss.usermodel package and it is a super-interface of all classes that create high or low level spreadsheets with specific names. The most common type of spreadsheet is worksheet, which is represented as a grid of cells.

## **HSSFSheet**

This is a class under the org.apache.poi.hssf.usermodel package. It can create excel spreadsheets and it allows to format the sheet style and sheet data.

## **Class Constructors**

### **S.No. Constructor and Description**

1

#### **HSSFSheet***HSSFWorkbook workbook*

Creates new HSSFSheet called by HSSFWorkbook to create a sheet from scratch.

2

#### **HSSFSheet***HSSFWorkbook workbook, InternalSheet sheet*

Creates an HSSFSheet representing the given sheet object.

## XSSFSheet

This is a class which represents high level representation of excel spreadsheet. It is under org.apache.poi.hssf.usermodel package.

### Class Constructors

#### S.No. Constructor and Description

1

##### **XSSFSheet**

Creates new XSSFSheet - called by XSSFWorkbook to create a sheet from scratch.

2

##### **XSSFSheet***PackagePart part, PackageRelationship rel*

Creates an XSSFSheet representing the given package part and relationship.

### Class Methods

#### S.No. Methods and Description

1

##### **addMergedRegion***CellRangeAddress region*

Adds a merged region of cells *hence those cells form one*.

2

##### **autoSizeColumn***int column*

Adjusts the column width to fit the contents.

3

##### **iterator**

This method is an alias for rowIterator to allow foreach loops

4

##### **addHyperlink***XSSFHyperlink hyperlink*

Registers a hyperlink in the collection of hyperlinks on this sheet

For the remaining methods of this class, refer the complete API at:

<https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFSheet.html>.

## Row

This is an interface under the org.apache.poi.ss.usermodel package. It is used for high-level representation of a row of a spreadsheet. It is a super-interface of all classes that represent rows in POI library.

## XSSFRow

This is a class under the **org.apache.poi.xssf.usermodel** package. It implements the Row

interface, therefore it can create rows in a spreadsheet. Listed below are the methods and constructors under this class.

## Class Methods

S.No.	Description
1	<b>createCell</b> <i>int columnIndex</i> Creates new cells within the row and returns it.
2	<b>setHeight</b> <i>short height</i> Sets the height in short units.

For the remaining methods of this class, follow the given link

<https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFRow.html>

## Cell

This is an interface under the org.apache.poi.ss.usermodel package. It is a super-interface of all classes that represent cells in the rows of a spreadsheet.

Cells can take various attributes such as blank, numeric, date, error, etc. Cells should have their own numbers *0based* before being added to a row.

## XSSFCell

This is a class under the org.apache.poi.xssf.usermodel package. It implements the Cell interface. It is a high-level representation of cells in the rows of a spreadsheet.

## Field Summary

Listed below are some of the fields of the XSSFCell class along with their description.

Cell Type	Description
CELL_TYPE_BLANK	Represents blank cell
CELL_TYPE_BOOLEAN	Represents Boolean cell <i>true</i> or <i>false</i>
CELL_TYPE_ERROR	Represents error value on a cell
CELL_TYPE_FORMULA	Represents formula result on a cell
CELL_TYPE_NUMERIC	Represents numeric data on a cell
CELL_TYPE_STRING	Represents string <i>text</i> on a cell

## Class Methods

S.No.	Description
1	<b>setCellStyle</b> <i>CellStyle style</i> Sets the style for the cell.

2

**setCellType***int cellType*

Sets the type of cells *numeric, formula, or string*.

3

**setCellValue***boolean value*

Sets a boolean value for the cell.

4

**setCellValue***java.util.Calendar value*

Sets a date value for the cell.

5

**setCellValue***double value*

Sets a numeric value for the cell.

6

**setCellValue***java.lang.String str*

Sets a string value for the cell.

7

**setHyperlink***Hyperlink hyperlink*

Assigns a hyperlink to this cell.

For the remaining methods and fields of this class, visit the following link:  
<https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFCCell.html>

## XSSFCCellStyle

This is a class under the `org.apache.poi.xssf.usermodel` package. It will provide possible information regarding the format of the content in a cell of a spreadsheet. It also provides options for modifying that format. It implements the `CellStyle` interface.

## Field Summary

The following table lists a few fields that are inherited from the `CellStyle` interface.

Field Name	Field Description
ALIGN_CENTER	Center align the cell contents
ALIGN_CENTER_SELECTION	Center-selection horizontal alignment
ALIGN_FILL	Cell fit to the content size
ALIGN_JUSTIFY	Fit cell contents to its width
ALIGN_LEFT	Left align the cell contents
ALIGN_RIGHT	Right align the cell contents
BORDER_DASH_DOT	Cell style with dash and dot

BORDER_DOTTED	Cell style with dotted border
BORDER_DASHED	Cell style with dashed border
BORDER_THICK	Cell style with thick border
BORDER_THIN	Cell style with thin border
VERTICAL_BOTTOM	Align the cell contents vertical bottom
VERTICAL_CENTER	Align the cell contents vertical center
VERTICAL_JUSTIFY	Align and justify the cell contents vertically
VERTICAL_TOP	Top aligned vertical alignment

## Class Constructors

### S.No. Constructor and Description

- 1 **XSSFCellStyle***int cellXfId, int cellStyleXfId, StylesTable stylesSource, ThemesTable theme*  
Creates a cell style from the supplied parts
- 2 **XSSFCellStyle***StylesTable stylesSource*  
Creates an empty cell Style

## Class Methods

Sets the type of border for the bottom border of the cell

### S.No Method and Description

- 1 **setAlignment***short align*  
Sets the type of horizontal alignment for the cell
- 2 **setBorderBottom***short border*
- 3 **setBorderColor***XSSFCellBorder. BorderSide side, XSSFCOLOR color*  
Sets the color for the selected border
- 4 **setBorderLeft***short border*  
Sets the type of border for the left border of the cell
- 5 **setBorderRight***short border*  
Sets the type of border for the right border of the cell

6

**setBorderTop***shortborder*

Sets the type of border for the top border of the cell

7

**setFillBackgroundColor***XSSFColorcolor*

Sets the background fill color represented as an XSSFColor value.

8

**setFillForegroundColor***XSSFColorcolor*

Sets the foreground fill color represented as an XSSFColor value.

9

**setFillPattern***shortfp*

Specifies the cell fill information for pattern and solid color cell fills.

10

**setFont***Fontfont*

Sets the font for this style.

11

**setRotation***shortrotation*

Sets the degree of rotation for the text in the cell.

12

**setVerticalAlignment***shortalign*

Sets the type of vertical alignment for the cell.

For the remaining methods and fields in this class, go through the following link:  
<https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFCellStyle.html>

## **HSSFColor**

This is a class under the org.apache.poi.hssf.util package. It provides different colors as nested classes. Usually these nested classes are represented by using their own indexes. It implements the Color interface.

### **Nested classes**

All nested classes of this class are static and each class has its index. These nested color classes are used for cell formatting such as cell content, border, foreground, and background. Listed below are some of the nested classes.

<b>S.No.</b>	<b>Class names</b> <i>colors</i>
--------------	----------------------------------

1	HSSFColor.AQUA
---	----------------

2	HSSFColor.AUTOMATIC
---	---------------------

3	HSSFColor.BLACK
---	-----------------

```
4     HSSFColor.BLUE
5     HSSFColor.BRIGHT_GREEN
6     HSSFColor.BRIGHT_GRAY
7     HSSFColor.CORAL
8     HSSFColor.DARK_BLUE
9     HSSFColor.DARK_GREEN
10    HSSFColor.SKY_BLUE
11    HSSFColor.WHITE
12    HSSFColor.YELLOW
```

## Class Methods

Only one method of this class is important and that is used to get the index value.

S.No.	Method and Description
-------	------------------------

1	<b>getIndex</b>
---	-----------------

This method is used to get the index value of a nested class

For the remaining methods and nested classes, refer the following link:  
<https://poi.apache.org/apidocs/org/apache/poi/hssf/util/HSSFColor.html>.

## XSSFColor

This is a class under the org.apache.poi.xssf.usermodel package. It is used to represent color in a spreadsheet. It implements the Color interface. Listed below are some of its methods and constructors.

## Class Constructors

S.No.	Constructor and Description
-------	-----------------------------

1	<b>XSSFColor</b>
---	------------------

Creates a new instance of XSSFColor.

2	<b>XSSFColorbyte[]rgb</b>
---	---------------------------

Creates a new instance of XSSFColor using RGB.

3	<b>XSSFColorjava.awt.Colorclr</b>
---	-----------------------------------

Creates a new instance of XSSFColor using the Color class from the awt package.

## Class Methods

### S.No. Method and Description

1

#### **setAuto***booleanauto*

Sets a boolean value to indicate that the ctColor is automatic and the system ctColor is dependent.

2

#### **setIndexed***intindexed*

Sets indexed ctColor value as system ctColor.

For the remaining methods, visit the following link:

<https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFColor.html>.

## XSSFFont

This is a class under the org.apache.poi.xssf.usermodel package. It implements the Font interface and therefore it can handle different fonts in a workbook.

## Class Constructor

### S.No. Constructor and Description

1

#### **XSSFFont**

Creates a new XSSFFont instance.

## Class Methods

### S.No. Method and Description

1

#### **setBold***booleanbold*

Sets a Boolean value for the 'bold' attribute.

2

#### **setColor***shortcolor*

Sets the indexed color for the font.

3

#### **setColor***XSSFColorcolor*

Sets the color for the font in Standard Alpha RGB color value.

4

#### **setFontHeight***shortheight*

Sets the font height in points.

5

**setFontName***java. lang. Stringname*

Sets the name for the font.

6

**setItalic***booleanitalic*

Sets a Boolean value for the 'italic' property.

For the remaining methods, go through the following link:

<https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFFont.html>.

## XSSFHyperlink

This is a class under the **org.apache.poi.xssf.usermodel** package. It implements the Hyperlink interface. It is used to set a hyperlink to the cell contents of a spreadsheet.

### Fields

The fields of this class are as follows. Here, fields mean the types of hyperlinks used.

Field	Description
LINK_DOCUMENT	Used to link any other document
LINK_EMAIL	Used to link email
LINK_FILE	Used to link any other file in any format
LINK_URL	Used to link a web URL

## Class Methods

### S.No. Method and Description

1

**setAddress***java. lang. Stringaddress*

Hyperlink address.

For the remaining methods, visit the following link:

<https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFHyperlink.html>

## XSSFCreationHelper

This is a class under the **org.apache.poi.xssf.usermodel** package. It implements the CreationHelper interface. It is used as a support class for formula evaluation and setting up hyperlinks.

### Class methods

### S.No. Method and Description

1

**createFormulaEvaluator**

Creates an XSSFFormulaEvaluator instance, the object that evaluates formula cells.

2

**createHyperlink***int type*

Creates a new XSSFHyperlink.

For the remaining methods, refer the following link:

<https://poi.apache.org/apidocs/org/apache/poi/xssf/usermodel/XSSFCreationHelper.html>.

## XSSFPrintSetup

This is a class under the **org.apache.poi.xssf.usermodel** package. It implements the PrintSetup interface. It is used to set print page size, area, options, and settings.

### Class Methods

S.No.	Method and Description
-------	------------------------

1

**setLandscape***boolean ls*

Sets a boolean value to allow or block landscape printing.

2

**setLeftToRight***boolean ltor*

Sets whether to go left to right or top down in ordering while printing.

3

**setPaperSize***short size*

Sets the paper size.

For the remaining methods, visit the following link:

<https://poi.apache.org/apidocs/org/apache/poi/hssf/usermodel/HSSFPrintSetup.html>

Processing math: 100%