[**http://www.allinoneblogs.com/java-tutorials/file-handling/file-handling-writing-data-into-an-excel-xlsx-or-xls-file/**](http://www.allinoneblogs.com/java-tutorials/file-handling/file-handling-writing-data-into-an-excel-xlsx-or-xls-file/)

**File Handling | Writing data into an Excel(.XLSX or .XLS) File.**

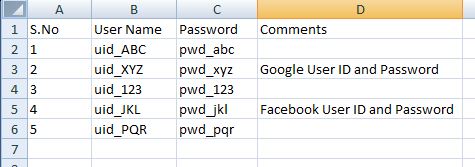
April 2, 2018 [Ashok Kumar](http://www.allinoneblogs.com/author/allinoneblogs/) [5 Comments](http://www.allinoneblogs.com/java-tutorials/file-handling/file-handling-writing-data-into-an-excel-xlsx-or-xls-file/#comments)

*Writing data into an Excel(.XLSX or .XLS) File using JAVA.*

**Overview of the Program**

1. Locate **path** and **filename** where the new excel file will be created using **File** Class.
2. Load the newly created file into **FileOutputStream** Class, so that new content can be written in it.
3. Create workbook using **XSSFWorkbook** class.
4. Create sheet in the workbook using **XSSFSheet** class.
5. Now, create new Row in the sheet using **createRow()** method of the sheet object.
6. Now, create new Cell in the Row using **createCell()** method of the Row object.
7. Now, input text/value in the newly created cell using **setCellValue()** Method of the cell object.
8. Once you have created a new Row, you can add multiple cells in the same row using Steps # 5-7.
9. Once you have entered all the values in the **FileOutputStream** object. Now, you have to write all the data in original excel file. This can be done using **write()** of the **XSSFWorkbook** Class.
10. At the end you have to close the workbook using **close()** of the **XSSFWorkbook** Class object.

**Output**

<img class="wp-image-576 " src="http://www.allinoneblogs.com/wp-content/uploads/2017/09/excelExample.jpg" alt="Create Excel Example-1" width="415" height="157" srcset="http://www.allinoneblogs.com/wp-content/uploads/2017/09/excelExample.jpg 375w, http://www.allinoneblogs.com/wp-content/uploads/2017/09/excelExample-300x114.jpg 300w" sizes="(max-width: 415px) 100vw, 415px" />Create Excel Example-1<img class="wp-image-577 size-full" src="http://www.allinoneblogs.com/wp-content/uploads/2017/09/ecxcel\_Example-2.jpg" alt="Create Excel Example-2" width="475" height="167" srcset="http://www.allinoneblogs.com/wp-content/uploads/2017/09/ecxcel\_Example-2.jpg 475w, http://www.allinoneblogs.com/wp-content/uploads/2017/09/ecxcel\_Example-2-300x105.jpg 300w" sizes="(max-width: 475px) 100vw, 475px" />Create Excel Example-2

Additional Notes

This program would support only the excel files having extension “.xlsx”. If you want to run the same code/program for the files having extension “.xls” replace all XSSF classes to HSSF as mentioned below:

|  |  |
| --- | --- |
| **testResult.xlsx** | **testResult.xls** |
| XSSFWorkbook | HSSFWorkbook |
| XSSFSheet | HSSFSheet |
| XSSFRow | HSSFRow |

Java Code (Example 1)

import java.io.File;

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 createAndWriteExcel

{

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

{

String filePath="output\_Excel//testResult.xlsx";

String sheetName="testData";

createExcel(filePath,sheetName);

}

public static void createExcel(String filePath,String sheetName) throws IOException

{

// Step #1 : Specify file name and location

File file=new File(filePath);

// Step #2 : load the excel file in FileOutputStream object

FileOutputStream fos=new FileOutputStream(file);

// Step #3 : Create new Workbook

XSSFWorkbook workbook =new XSSFWorkbook(fos);

// Step #4 : Create new sheet

XSSFSheet sheet=workbook.createSheet(sheetName);

// Adding content to the Excel sheet

addingContent(sheet);

// Write and update Excel File

writeAndUpdateExcel(workbook, fos);

}

public static void addingContent(XSSFSheet sheet)

{

// Step #5 : Create new Row in the sheet

XSSFRow headingRow=sheet.createRow(0);

// Step #6 and Step #7 : Create new Cell and add content to each cell

headingRow.createCell(0).setCellValue("S.No");

headingRow.createCell(1).setCellValue("Website Name");

headingRow.createCell(2).setCellValue("URL");

// Step #8 : Creating/Adding multiple rows and cells

XSSFRow firstRow=sheet.createRow(1);

firstRow.createCell(0).setCellValue("1");

firstRow.createCell(1).setCellValue("Google");

firstRow.createCell(2).setCellValue("www.google.com");

XSSFRow secondRow=sheet.createRow(2);

secondRow.createCell(0).setCellValue("2");

secondRow.createCell(1).setCellValue("Facebook");

secondRow.createCell(2).setCellValue("www.facebook.com");

XSSFRow thirdRow=sheet.createRow(3);

thirdRow.createCell(0).setCellValue("3");

thirdRow.createCell(1).setCellValue("Twitter");

thirdRow.createCell(2).setCellValue("https://twitter.com/");

XSSFRow fourthRow=sheet.createRow(4);

fourthRow.createCell(0).setCellValue("4");

fourthRow.createCell(1).setCellValue("Guru99");

fourthRow.createCell(2).setCellValue("https://www.guru99.com/");

}

public static void writeAndUpdateExcel(XSSFWorkbook workbook, FileOutputStream fos) throws IOException

{

// Step #9 : Writing all the data from workbook object to FileOutputStream object

workbook.write(fos);

// Step #10 : Closing the workbook object.

workbook.close();

System.out.println("File created and saved at specified location....");

}

}

**Java Code (Example 2)**

The alternate way to write the content in the excel file using loop and function to reduce code and complexity of the Program.

import java.io.File;

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 createAndWriteExcel\_Example2

{

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

{

String filePath="output\_Excel//testResult2.xlsx";

String sheetName="testData";

createExcel(filePath,sheetName);

}

public static void createExcel(String filePath,String sheetName) throws IOException

{

// Step #1 : Specify file name and location

File file=new File(filePath);

// Step #2 : load the excel file in FileOutputStream object

FileOutputStream fos=new FileOutputStream(file);

// Step #3 : Create new Workbook

XSSFWorkbook workbook =new XSSFWorkbook();

// Step #4 : Create new sheet

XSSFSheet sheet=workbook.createSheet(sheetName);

// Adding content to the Excel sheet

String[] heading={"S.No","User Name","Password","Comments"};

String[] row1={"1","uid\_ABC","pwd\_abc"};

String[] row2={"2","uid\_XYZ","pwd\_xyz","Google User ID and Password"};

String[] row3={"3","uid\_123","pwd\_123"};

String[] row4={"4","uid\_JKL","pwd\_jkl","Facebook User ID and Password"};

String[] row5={"5","uid\_PQR","pwd\_pqr"};

addingContent(sheet,0,heading);

addingContent(sheet,1,row1);

addingContent(sheet,2,row2);

addingContent(sheet,3,row3);

addingContent(sheet,4,row4);

addingContent(sheet,5,row5);

// Write and update Excel File

writeAndUpdateExcel(workbook, fos);

}

public static void addingContent(XSSFSheet sheet,int rowIndex,String[] cellValues)

{

// Step #5 : Create new row in the sheet

XSSFRow currentRow=sheet.createRow(rowIndex);

int cellIndex=0;

// Step #6-8 : Adding new cells and values in each row

for (String currentCellValue : cellValues)

{

currentRow.createCell(cellIndex).setCellValue(currentCellValue);

cellIndex++;

}

}

public static void writeAndUpdateExcel(XSSFWorkbook workbook, FileOutputStream fos) throws IOException

{

// Step #9 : Writing all the data from workbook object to FileOutputStream object

workbook.write(fos);

// Step #10 : Closing the workbook object.

workbook.close();

System.out.println("File created and saved at specified location....");

}

}