**1) \* Step 1: Add Apache POI dependency.**

Add the following dependencies to your pom.xml file:

<dependencies>

<!-- Apache POI library for working with Excel files -->

<dependency>

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

<artifactId>poi</artifactId>

<version>5.2.3</version> <!-- Check for the latest version -->

</dependency>

<dependency>

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

<artifactId>poi-ooxml</artifactId>

<version>5.2.3</version> <!-- Check for the latest version -->

</dependency>

</dependencies>

**\*Step 2: Create the Java Program**

Now, you can write the Java program that creates a new Excel workbook.

import org.apache.poi.ss.usermodel.\*;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

public class ExcelWriteExample {

public static void main(String[] args) {

// Create a new workbook (Excel file)

Workbook workbook = new XSSFWorkbook(); // For .xlsx format

// Create a new sheet in the workbook

Sheet sheet = workbook.createSheet("Sheet1");

// Create a row in the sheet (row index starts from 0)

Row row = sheet.createRow(0);

// Create a cell in the row (column index starts from 0)

Cell cell = row.createCell(0);

cell.setCellValue("Hello, Excel!"); // Set value to the cell

// Write the output to a file

try (FileOutputStream fileOut = new FileOutputStream(new File("workbook.xlsx"))) {

workbook.write(fileOut); // Write the workbook to the file

System.out.println("Excel file created successfully!");

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

workbook.close(); // Close the workbook

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

**2)** import org.apache.poi.ss.usermodel.\*;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

public class ExcelWriteExample {

public static void main(String[] args) {

// Create a new workbook (Excel file)

Workbook workbook = new XSSFWorkbook(); // For .xlsx format

// Create a new sheet with the name "Sheet1"

Sheet sheet = workbook.createSheet("Sheet1");

// Create a row in the sheet (row index starts from 0)

Row row = sheet.createRow(0);

// Create a cell in the row (column index starts from 0)

Cell cell = row.createCell(0);

cell.setCellValue("Hello, Excel!"); // Set value to the cell

// Write the output to a file

try (FileOutputStream fileOut = new FileOutputStream(new File("workbook.xlsx"))) {

workbook.write(fileOut); // Write the workbook to the file

System.out.println("Excel file with sheet 'Sheet1' created successfully!");

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

workbook.close(); // Close the workbook

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

**3)** import org.apache.poi.ss.usermodel.\*;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

public class ExcelWriteExample {

public static void main(String[] args) {

// Create a new workbook (Excel file)

Workbook workbook = new XSSFWorkbook(); // For .xlsx format

// Create a new sheet with the name "Sheet1"

Sheet sheet = workbook.createSheet("Sheet1");

// Create the header row (row index 0)

Row headerRow = sheet.createRow(0);

// Set the column headers

String[] columnHeaders = {"Name", "Age", "Email"};

for (int i = 0; i < columnHeaders.length; i++) {

Cell cell = headerRow.createCell(i);

cell.setCellValue(columnHeaders[i]);

}

// Data to be inserted

Object[][] data = {

{"John Doe", 30, "john@test.com"},

{"Jane Doe", 28, "jane@test.com"},

{"Bob Smith", 35, "jacky@gmail.com"},

{"Swapnil", 37, "swapnil@gmail.com"}

};

// Loop through the data array and create rows and cells

int rowNum = 1; // Start from row 1 (because row 0 is the header)

for (Object[] rowData : data) {

Row row = sheet.createRow(rowNum++);

int cellNum = 0;

for (Object field : rowData) {

Cell cell = row.createCell(cellNum++);

if (field instanceof String) {

cell.setCellValue((String) field); // If it's a string, set as String

} else if (field instanceof Integer) {

cell.setCellValue((Integer) field); // If it's an integer, set as Integer

}

}

}

// Write the output to a file

try (FileOutputStream fileOut = new FileOutputStream(new File("workbook.xlsx"))) {

workbook.write(fileOut); // Write the workbook to the file

System.out.println("Excel file created successfully!");

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

workbook.close(); // Close the workbook

} catch (IOException e) {

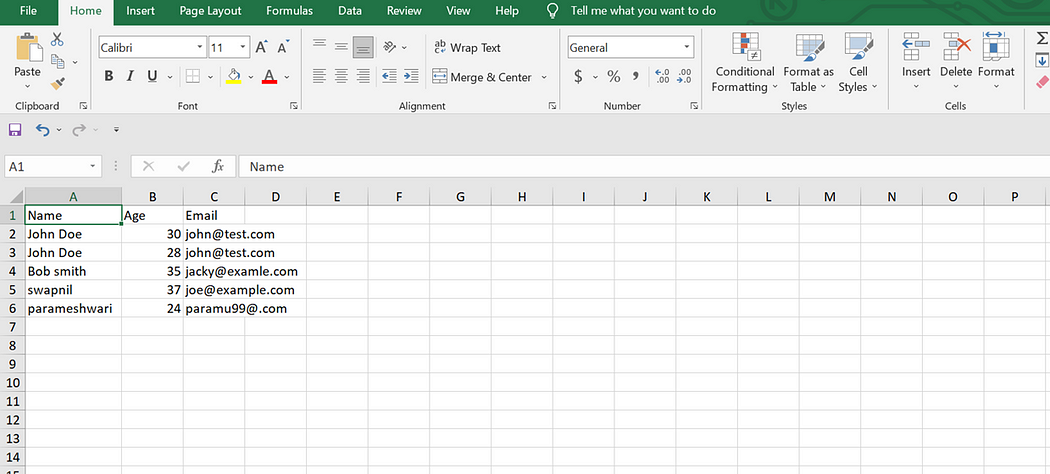
e.printStackTrace();

}

}

}

}



**4)** import org.apache.poi.ss.usermodel.\*;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

public class WriteToExcelExample {

public static void main(String[] args) {

// Step 1: Create a new workbook (Excel file)

Workbook workbook = new XSSFWorkbook(); // For .xlsx format (Excel 2007 or later)

// Step 2: Create a new sheet in the workbook

Sheet sheet = workbook.createSheet("Sheet1");

// Step 3: Create a row in the sheet (row index starts from 0)

Row row = sheet.createRow(0); // Create the first row (header row)

// Step 4: Create cells in the header row and set values

String[] headers = {"Name", "Age", "Email"};

for (int i = 0; i < headers.length; i++) {

Cell cell = row.createCell(i); // Create a cell in each column

cell.setCellValue(headers[i]); // Set the header text

}

// Step 5: Add data to the sheet

Object[][] data = {

{"John Doe", 30, "john@test.com"},

{"Jane Doe", 28, "jane@test.com"},

{"Bob Smith", 35, "bob.smith@example.com"},

{"Alice Brown", 32, "alice.brown@example.com"}

};

// Add the data rows below the header (starting at row 1)

int rowNum = 1; // Row number starts from 1 since row 0 is already used for headers

for (Object[] rowData : data) {

row = sheet.createRow(rowNum++); // Create a new row for each data entry

int cellNum = 0;

for (Object field : rowData) {

Cell cell = row.createCell(cellNum++); // Create cells in each column

if (field instanceof String) {

cell.setCellValue((String) field); // If the data is a String

} else if (field instanceof Integer) {

cell.setCellValue((Integer) field); // If the data is an Integer

}

}

}

// Step 6: Write the data to an Excel file

try (FileOutputStream fileOut = new FileOutputStream(new File("output.xlsx"))) {

workbook.write(fileOut); // Write the workbook to the file

System.out.println("Excel file 'output.xlsx' created successfully!");

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

workbook.close(); // Close the workbook to release resources

} catch (IOException e) {

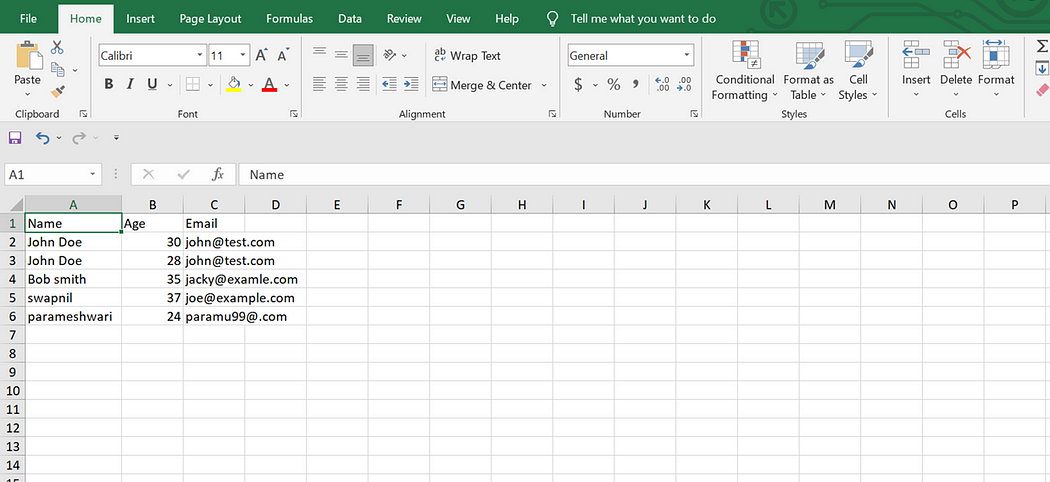
e.printStackTrace();

}

}

}

}



**5**) import org.apache.poi.ss.usermodel.\*;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

public class ReadExcelExample {

public static void main(String[] args) {

// Path to the Excel file to be read

String filePath = "input.xlsx"; // Specify the path to your Excel file

// Read the Excel file

try (FileInputStream fis = new FileInputStream(new File(filePath))) {

// Create a Workbook instance based on the file extension (XSSFWorkbook for .xlsx)

Workbook workbook = new XSSFWorkbook(fis);

// Get the first sheet from the workbook

Sheet sheet = workbook.getSheetAt(0); // Get the first sheet (index 0)

// Iterate over the rows

for (Row row : sheet) {

// Iterate over the cells in the row

for (Cell cell : row) {

// Print the cell value to the console

switch (cell.getCellType()) {

case STRING:

System.out.print(cell.getStringCellValue() + "\t");

break;

case NUMERIC:

// Handle numeric cells, including dates

if (DateUtil.isCellDateFormatted(cell)) {

System.out.print(cell.getDateCellValue() + "\t");

} else {

System.out.print(cell.getNumericCellValue() + "\t");

}

break;

case BOOLEAN:

System.out.print(cell.getBooleanCellValue() + "\t");

break;

case FORMULA:

System.out.print(cell.getCellFormula() + "\t");

break;

default:

System.out.print("Unknown Cell Type\t");

break;

}

}

System.out.println(); // Move to the next line after each row

}

// Close the workbook

workbook.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

