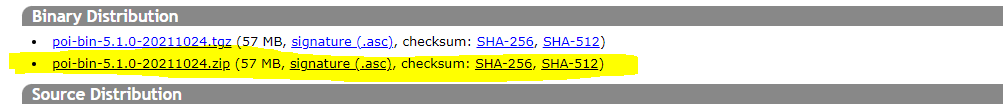
Excel Reader

To read the data from excel sheet we require Apache Poi library inorder to read and write the data in excel sheet.

We need to configure the project for excel sheet read/ write operation for that we require Apache poi:

Step 1: download the library from:

<https://poi.apache.org/download.html>

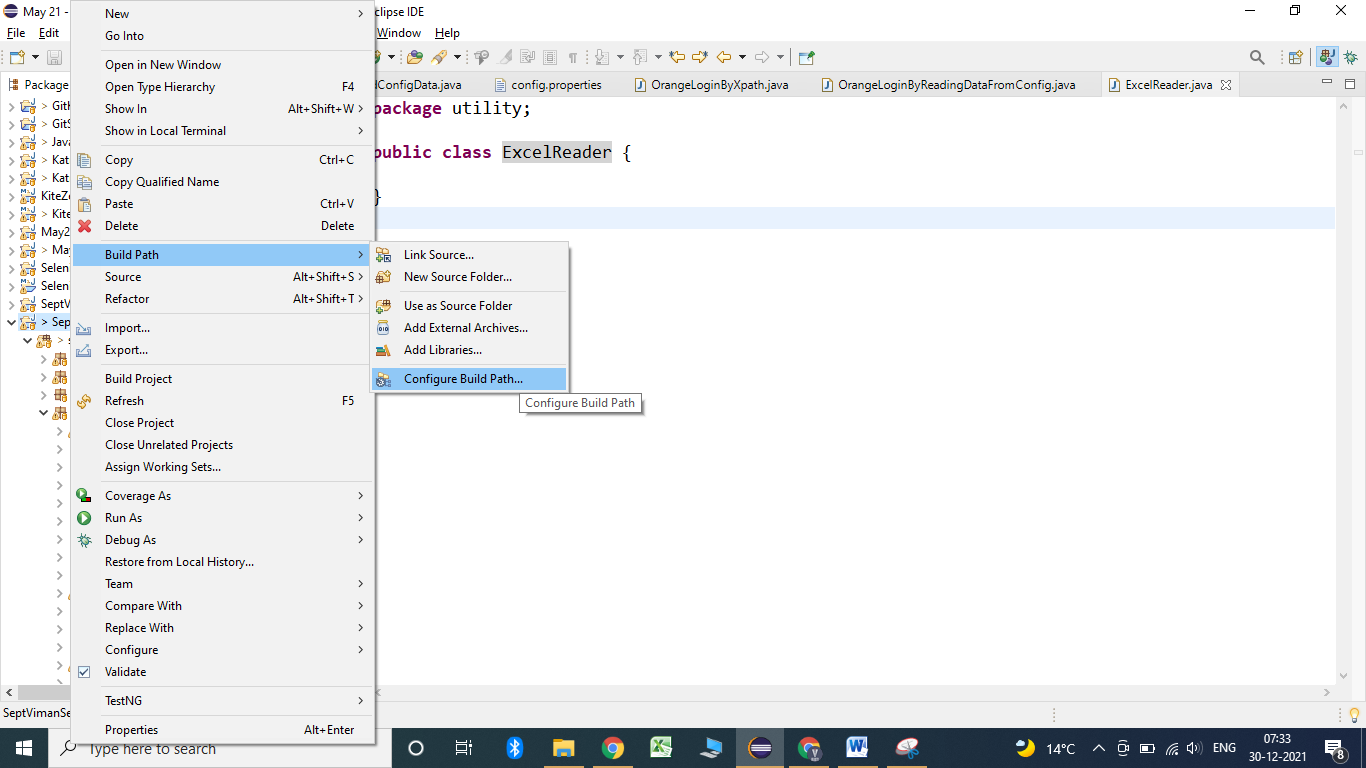




Step 2: Extract the zip file.

Step 3: Accumulate all the jars at a single location.

Step 4: configure the project for using Excel read and write operation



Add the jars of poi into the project and say apply and close.

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

File src = **new** File("C:\\Users\\A\\Desktop\\TestDataSeptBatch.xlsx");

// To load that particular location in order to perform the action

FileInputStream fis = **new** FileInputStream(src);

XSSFWorkbook wb = **new** XSSFWorkbook(fis);

XSSFSheet sh1 = wb.getSheet("Testsheet");

String value = sh1.getRow(2).getCell(1).getStringCellValue();

System.***out***.println(value);

DataFormatter df = **new** DataFormatter();

String numericstringvalue = df.formatCellValue(sh1.getRow(20).getCell(0));

System.***out***.println(numericstringvalue);

// String getintvalue = sh1.getRow(20).getCell(0).getStringCellValue();

//

// System.out.println(getintvalue);

}

Read and Write data:

**public** **static** String readData(**int** row, **int** column) **throws** IOException

{

File src = **new** File("C:\\Users\\A\\Desktop\\TestDataSeptBatch.xlsx");

// To load that particular location in order to perform the action

FileInputStream fis = **new** FileInputStream(src);

XSSFWorkbook wb = **new** XSSFWorkbook(fis);

XSSFSheet sh1 = wb.getSheet("Testsheet");

DataFormatter df = **new** DataFormatter();

String stringvalue = df.formatCellValue(sh1.getRow(row).getCell(column));

**return** stringvalue;

}

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

File src = **new** File("C:\\Users\\A\\Desktop\\TestDataSeptBatch.xlsx");

// To load that particular location in order to perform the action

FileInputStream fis = **new** FileInputStream(src);

XSSFWorkbook wb = **new** XSSFWorkbook(fis);

XSSFSheet sh1 = wb.getSheet("Testsheet");

String value = sh1.getRow(2).getCell(1).getStringCellValue();

System.***out***.println(value);

DataFormatter df = **new** DataFormatter();

String numericstringvalue = df.formatCellValue(sh1.getRow(20).getCell(0));

System.***out***.println(numericstringvalue);

// String getintvalue = sh1.getRow(20).getCell(0).getStringCellValue();

//

// System.out.println(getintvalue);

*writeData*();

}

**public** **static** **void** writeData() **throws** IOException

{

File src = **new** File("C:\\Users\\A\\Desktop\\TestDataSeptBatch.xlsx");

// To load that particular location in order to perform the action

FileInputStream fis = **new** FileInputStream(src);

XSSFWorkbook wb = **new** XSSFWorkbook(fis);

XSSFSheet sh1 = wb.getSheet("Testsheet");

File fout = **new** File("C:\\Users\\A\\Desktop\\TestDataSeptBatch.xlsx");

FileOutputStream fos = **new** FileOutputStream(fout);

// To write for Already available row but not available column

sh1.getRow(8).createCell(2).setCellValue("Write Test");

// To write for un available row also not available column

sh1.createRow(50).createCell(3).setCellValue("Newrow and column");

// to save the content

wb.write(fos);

}

}