# 一 POI介绍

POI（Poor Obfuscation Implementation）是Apache提供的操作ms office格式文档的API。本文主要针对excel进行操作。使用POI可以操作Excel 95及以后的版本，即可操作后缀为 .xls 和 .xlsx两种格式的excel。

主要结构：

HSSF : 它被用来读取和写入MS-Excel文件的xls格式

XSSF : 它是用于MS-Excel中XLSX文件格式

HWPF : 它是用来读取和写入MS-Word的文档扩展名的文件

XWPF : 它是用来读取和写入MS-Word的docx扩展名的文件

HSLF : 它是用于读取，创建和编辑PowerPoint演示文稿

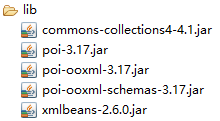
HDGF : 它包含类和方法为MS-Visio的二进制文件

HPBF : 它被用来读取和写入MS-Publisher文件

# 二 针对Excel的导入导出操作

## 1 导入jar包

需要导入的jar包



对于只操作2003及以前版本的excel，只需要导入poi-XXX.jar ，如果还需要对2007及以后版本进行操作，则需要导入

poi-ooxml-XXX.jar

poi-ooxml-schemas-XXX.jar

Maven方式

<dependency>

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

<artifactId>poi</artifactId>

<version>3.17</version>

</dependency>

<dependency>

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

<artifactId>poi-ooxml</artifactId>

<version>3.17</version>

</dependency>

## 2 读取excel文件

1）读取xlsx后缀的文件

**public** **void** readXlsx() **throws** Exception{

//创建输入流

FileInputStream fis = **new** FileInputStream(**new** File("d:/a.xlsx"));

//由输入流得到工作簿

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

//得到工作表

// XSSFSheet sheet = workbook.getSheet("sheet1");

XSSFSheet sheet = workbook.getSheetAt(0);

//得到行,0表示第一行

XSSFRow row = sheet.getRow(0);

//创建单元格行号由row确定,列号作为参数传递给createCell;第一列从0开始计算

XSSFCell cell = row.getCell(0);

//获取单元格值

String cellValue = cell.getStringCellValue();

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

//关闭

workbook.close();

}

2）读取xls后缀文件

**public** **void** readXls() **throws** Exception{

//创建输入流

FileInputStream fis = **new** FileInputStream(**new** File("d:/b.xls"));

//由输入流得到工作簿

HSSFWorkbook workbook = **new** HSSFWorkbook(fis);

//得到工作表

HSSFSheet sheet = workbook.getSheetAt(0);

**int** rows = sheet.getLastRowNum();

**for**(**int** i = 0; i <= rows; i++){

//得到行,0表示第一行

HSSFRow row = sheet.getRow(i);

// short cells = row.getLastCellNum();

// for(short j = 0; j < cells; j++){

//

// }

**for**(**int** j = 0; j < 3; j++){

// 如果是空白内容单元格，返回null

HSSFCell cell = row.getCell(j, MissingCellPolicy.***RETURN\_BLANK\_AS\_NULL***);

**if**(cell == **null**){

**continue**;

}

// 获取单元格类型

CellType cellTypeEnum = cell.getCellTypeEnum();

**switch**(cellTypeEnum){

**case** ***STRING***:

String cellValue = cell.getStringCellValue();

System.***out***.println("cell的值是"+cellValue);

**break**;

**case** ***NUMERIC***:

String value = "";

String aa = cell.getCellStyle().getDataFormatString();

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

**if**(cell.getCellStyle().getDataFormatString().equals("General")){

value = **new** DecimalFormat().format(cell.getNumericCellValue());

}**else** **if**("m/d/yy".equals(cell.getCellStyle().getDataFormatString())){

SimpleDateFormat sdf = **new** SimpleDateFormat("yyyy-MM-dd hh:mm:ss");

value = sdf.format(cell.getDateCellValue());

}

System.***out***.println("Cell的值是"+value);

**break**;

**case** ***BLANK***: // 空值

**break**;

**case** ***\_NONE***:

**break**;

**case** ***FORMULA***:// 公式

**break**;

**case** ***ERROR***: // 故障

**break**;

**case** ***BOOLEAN***:

**break**;

}

}

}

workbook.close();

}

## 3 写excel文件

**public** **void** exportXLSx(){

List<Map<String, Object>> infos = **new** ArrayList<>();

**for**(**int** i = 0; i < 10; i++){

Map<String, Object> info = **new** HashMap<>();

info.put("name", "zhangsan" + i);

info.put("age", 10 + i);

info.put("tel", "13512345671");

info.put("date", "2015年10月");

infos.add(info);

}

SimpleDateFormat dateFormat = **new** SimpleDateFormat("YYYYMMDDhhmmss");

String now = dateFormat.format(**new** Date());

//导出文件路径

String basePath = "D:/";

//文件名

String exportFileName = "data\_"+now+".xlsx";

String[] cellTitle = {"姓名","年龄","手机号","入学日期"};

// 创建一个工作薄

XSSFWorkbook workBook = **null**;

workBook = **new** XSSFWorkbook();

// 生成一个表格

XSSFSheet sheet = workBook.createSheet();

workBook.setSheetName(0,"学生信息");

// 创建表格行

XSSFRow titleRow = sheet.createRow(0);

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

//创建单元格

titleRow.createCell(i).setCellValue(cellTitle[i]);

}

//插入需导出的数据

**for**(**int** i=0;i<infos.size();i++){

XSSFRow row = sheet.createRow(i+1);

row.createCell(0).setCellValue((String)infos.get(i).get("name"));

row.createCell(1).setCellValue((Integer)infos.get(i).get("age"));

row.createCell(2).setCellValue((String)infos.get(i).get("tel"));

row.createCell(3).setCellValue((String)infos.get(i).get("date"));

}

File file = **new** File(basePath+exportFileName);

//文件输出流

FileOutputStream outStream = **null**;

**try** {

outStream = **new** FileOutputStream(file);

//将工作簿中内容写入文件输出流

workBook.write(outStream);

outStream.flush();

outStream.close();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

## 4 web方式下的导入导出

1）导入

流程上与上传文件类似，需要得到输出流：

InputStream inputStream = upfile.getInputStream();

**public** **static** **void** importExcel(String fileName, InputStream inputStream) **throws** Exception{

**boolean** ret = *isXls*(fileName);

//1.读取工作簿

Workbook workBook = **null**;

**if**(ret){

workBook = **new** HSSFWorkbook(inputStream);

}**else**{

workBook = **new** XSSFWorkbook(inputStream);

}

// 2 读取工作表

Sheet sheet = workBook.getSheetAt(0);

// 3 读取行

**int** nums = sheet.getLastRowNum();

**for**(**int** i = 1; i <= nums; i++){

Row row = sheet.getRow(i);

Cell cell0 = row.getCell(0, MissingCellPolicy.***RETURN\_BLANK\_AS\_NULL***);

**if**(cell0 != **null**){

System.***out***.println(cell0.getStringCellValue());

}

Cell cell1 = row.getCell(1, MissingCellPolicy.***RETURN\_BLANK\_AS\_NULL***);

**if**(cell1 != **null**){

System.***out***.println(cell1.getNumericCellValue());

}

}

workBook.close();

}

2）导出

**public** **static** **void** exportExcel(OutputStream outputstream) **throws** Exception{

List<Map<String, Object>> infos = **new** ArrayList<>();

**for**(**int** i = 0; i < 10; i++){

Map<String, Object> info = **new** HashMap<>();

info.put("name", "zhangsan" + i);

info.put("age", 10 + i);

info.put("tel", "13512345671");

info.put("date", "2015年10月");

infos.add(info);

}

SimpleDateFormat dateFormat = **new** SimpleDateFormat("YYYYMMDDhhmmss");

String now = dateFormat.format(**new** Date());

//文件名

String exportFileName = "数据\_"+now+".xlsx";

String[] cellTitle = {"姓名","年龄","手机号","入学日期"};

// 声明一个工作薄

Workbook workBook = **null**;

workBook = **new** XSSFWorkbook();

// 生成一个表格

Sheet sheet = workBook.createSheet();

workBook.setSheetName(0,"学生信息");

// 创建表格标题行 第一行

Row titleRow = sheet.createRow(0);

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

titleRow.createCell(i).setCellValue(cellTitle[i]);

}

//插入需导出的数据

**for**(**int** i=0;i<infos.size();i++){

Row row = sheet.createRow(i+1);

row.createCell(0).setCellValue((String)infos.get(i).get("name"));

row.createCell(1).setCellValue((Integer)infos.get(i).get("age"));

row.createCell(2).setCellValue((String)infos.get(i).get("tel"));

row.createCell(3).setCellValue((String)infos.get(i).get("date"));

}

workBook.write(outputstream);

workBook.close();

}

3）其他

**public** **static** **boolean** isXls(String fileName){

// (?i)忽略大小写

**if**(fileName.matches("^.+\\.(?i)(xls)$")){

**return** **true**;

}**else** **if**(fileName.matches("^.+\\.(?i)(xlsx)$")){

**return** **false**;

}**else**{

**throw** **new** RuntimeException("格式不对");

}

}