链滴

常用 POI 方法

作者：howie404
原文链接：https://ld246.com/article/1561659672663
来源网站：链滴

目前常见读写Excel的工具类开源javaAPI有两种方式，一个是JXL（Java Excel API） 官网地址：http://jexcelapi.sourceforge.net/一个是Apache的POI（Poor Obfuscation Implementation）官网地址：ttp://poi.apache.org/

POI支持微软的OLE2格式文件Office 2003及以下版本；同时支持微软的OOXML（Office Open XM）标准，也就是Office 2007以上版本。JXL只能实现对Excel 2003以下版本的支持。

POI使用HSSF对象操作OLE2格式Excel，文件后缀为.xls的；使用XSSF、SXSSF对象操作OOXML格式xcel，文件后缀为.xlsx的。

对于OLE2版本的Excel，一个Sheet工作表它的行最多支持到65536行，列支持到256列；对于OOXM版本的Excel，一个Sheet工作表它的行支持到1048576行，列支持到16384列。

核心API：

数据限制：

Excel2003 2007、2010

列： 255 16384行： 65535 1048576

========== 基础 ==========

// 创建excel（工作簿） 使用接口的方式来创建Workbook wb = new HSSFWorkbook();

新建工作簿：HSSFWorkbook wb = new HSSFWorkbook();

打开工作簿：HSSFWorkbook wb = new HSSFWorkbook(new FileInputStream(xlsFile));

建立新的sheet对象：HSSFSheet sheet = wb.createSheet("我的第一个工作簿");

选择第一个工作簿：HSSFSheet sheet = wb.getSheetAt(0);

设置工作簿的名称：wb.setSheetName(0, "我的第一个工作簿");

创建行对象： HSSFRow nRow = null;nRow = sheet.createRow(1); //第2行

指定列 创建单元格对象：HSSFCell nCell = null;nCell = nRow.createCell((short)(2)); //第3列

指定列 创建单元格对象：nCell.setCellValue("我是单元格");

// 获取到样式的对象CellStyle style = wb.createCellStyle();

// 创建字体对象Font font = wb.createFont();// 设置字体大小font.setFontHeightInPoints((short)6);// 设置字体的名称font.setFontName("楷体");// 设置字体style.setFont(font);

设置样式 注意：样式不能重复设置nCell.setCellStyle(leftStyle(wb));

文件下载方法1：

先在服务器产生临时文件，再下载临时文件。

关闭保存excel文件

```
FileOutputStream fOut = new FileOutputStream(xlsFile);  //创建xls文件，无内容 0字节
wb.write(fOut);  //写内容，xls文件已经可以打开
fOut.flush();  //刷新缓冲区
fOut.close();  //关闭
```

文件下载方法2：

```
 //7.生成excel文件
 ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();  //生成流
象
 wb.write(byteArrayOutputStream);  //将excel写入流
 //工具类，封装弹出下载框：
 String outFile = "生产厂家通讯录.xls";
 DownloadBaseAction down = new DownloadBaseAction();
 down.download(byteArrayOutputStream, response, outFile);
```

文件下载方法3：（适用于struts2）

```
 ServletActionContext.getResponse().setContentType("application/octet-stream");
 String returnName = ServletActionContext.getResponse().encodeURL( new String("购销合同.x
s".getBytes(), "ISO-8859-1"));
 ServletActionContext.getResponse().addHeader("Content-Disposition", "attachment;filenam
=" + returnName);
 wb.write(ServletActionContext.getResponse().getOutputStream());
```

文件下载方法4：

```
 //下载文件
 response.setContentType("application/octet-stream");
 String returnName = response.encodeURL( new String("生产厂家通讯录.xls".getBytes(), "ISO-8
59-1"));
 response.addHeader("Content-Disposition", "attachment;filename=" + returnName);
 wb.write(response.getOutputStream());
```

字体修饰：

```
//设置单元格样式
private HSSFCellStyle leftStyle(HSSFWorkbook wb){
 HSSFCellStyle curStyle = wb.createCellStyle();
 HSSFFont curFont = wb.createFont();  //设置字体
 //curFont.setFontName("Times New Roman");  //设置英文字体
 curFont.setFontName("微软雅黑");  //设置英文字体
 curFont.setCharSet(HSSFFont.DEFAULT_CHARSET);  //设置中文字体，那必须还要再对单元格进
编码设置
 curFont.setFontHeightInPoints((short)10);  //字体大小
 curFont.setBoldweight(HSSFFont.BOLDWEIGHT_BOLD);  //加粗
 curStyle.setFont(curFont);
```

```
 curStyle.setBorderTop(HSSFCellStyle.BORDER_THICK);  //粗实线
 curStyle.setBorderBottom(HSSFCellStyle.BORDER_THIN);  //实线
 curStyle.setBorderLeft(HSSFCellStyle.BORDER_MEDIUM);  //比较粗实线
 curStyle.setBorderRight(HSSFCellStyle.BORDER_THIN);  //实线

 curStyle.setWrapText(true);   //换行
 curStyle.setAlignment(HSSFCellStyle.ALIGN_RIGHT);  //横向具右对齐
 curStyle.setVerticalAlignment(HSSFCellStyle.VERTICAL_CENTER);  //单元格垂直居中

 return curStyle;
}
```

========== web环境 ==========

设置打印方向：默认纵向

PrintSetup ps = sheet.getPrintSetup();ps.setLandscape(true); //横向打印

自适应列宽：//bug 对中文支持不好，列宽不够宽for(int i=0 ;i<titles.length;i++){

sheet.autoSizeColumn((short)i);

}

设置行高：nRow.setHeightInPoints(18);

设置列宽：sheet.setColumnWidth((short)colNo, (short)(256*8));

设置每列默认宽度：sheet.setDefaultColumnWidth((short) 20);

设置标题：将第一行作为标题，即每页都打印此行 sheetN,startCol,stopCol,startRow,stopRow

wb.setRepeatingRowsAndColumns(0,1,8,0,1);

页脚：HSSFFooter footer = sheet.getFooter();footer.setRight("第"+HSSFFooter.page()+"页 共" HSSFFooter.numPages()+"页 "); //页数

工具类-单元格自适应高度：float height = pioUtil.getCellAutoHeight(extcproducts, 12f);nRow.se HeightInPoints(height); //(一行字+行之间的间隙)*行数

分页：// POI分页符有BUG，必须在模板文件中插入一个分页符，然后再此处删除预设的分页符；最在下面重新设置分页符。// sheet.setAutobreaks(false);// int iRowBreaks[] = sheet.getRowBreaks);// sheet.removeRowBreak(3);// sheet.removeRowBreak(4);// sheet.removeRowBreak(5);// sh et.removeRowBreak(6);

sheet.setRowBreak(行数); //在第startRow行设置分页符

==出货表：

合并单元格：//纵向合并单元格 Region region = null;region = new Region(curRow-1, (short)(1), urRow-1+3, (short)1); sheet.addMergedRegion(region);

//横向合并单元格CellRangeAddresssheet.addMergedRegion(new CellRangeAddress(开始行，

束行，开始列，结束列));

// 横向居中style.setAlignment(CellStyle.ALIGN_CENTER); // 纵向居中style.setVerticalAlignment(ellStyle.VERTICAL_CENTER);

文件直接输出：

```
ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();  //生成流象
wb.write(byteArrayOutputStream);  //将excel写入流

HttpServletResponse response = ServletActionContext.getResponse();

//工具类，封装弹出下载框：
DownloadBaseAction down = new DownloadBaseAction();
down.download(byteArrayOutputStream, response, outFile);
```

获取模板：

```
int curRow = 0;  //当前行
int colNo = 1;  //当前列

//得到模板路径
String rootPath = UtilFuns.getROOTPath();
String xlsFile = rootPath + "/make/xlsprint/tOUTPRODUCT.xls";

//新建临时目录，存放excel  /root/web/tmpfile/yyyy-mm-dd/...
String filePath = "/web/tmpfile/" + UtilFuns.sysDate()+"/";
File tmpDir = new File(rootPath + filePath);
if(!tmpDir.exists()){
tmpDir.mkdirs();  //创建多级目录
}

FileUtil fu = new FileUtil();
String sFile = fu.newFile(rootPath+filePath, "outproduct.xls");  //防止文件并发访问

String outFile = rootPath+filePath+sFile;  //输出文件
```

==合同打印：

1、 分页

sheet.setRowBreak(当前行); //设置分页符

2、怎么插入一个图片HSSFPatriarch patriarch = sheet.createDrawingPatriarch(); //add picture

pioUtil.setPicture(wb, patriarch, rootPath+"make/xlsprint/logo.jpg", curRow, 2, curRow+4, 2);

3、怎么插入一条线

pioUtil.setLine(wb, patriarch, curRow, 2, curRow, 8); //draw line

4、设置数值类型nCell.setCellType(HSSFCell.CELL_TYPE_NUMERIC);

5、设置前导符

HSSFDataFormat format = wb.createDataFormat();return format.getFormat("" ¥ "
,###,##0.00"); // 设置格式

6、设置公式

nCell.setCellType(HSSFCell.CELL_TYPE_FORMULA);nCell.setCellFormula("F11*H11");nCell.setCel
Formula("F"+String.valueOf(curRow)+"*H"+String.valueOf(curRow));

nCell.setCellFormula("SUM(I"+String.valueOf(curRow-4)+":I"+String.valueOf(curRow-1)+")");

7、工具类：替换等量空格

fixSpaceStr(String str,int len)

8、业务要求：1）同一个厂家的货物才能打印到同一个页面

List<ContractProduct> oList = oDao.find("from ContractProduct o where o.contract.id='"+co
tractId+"' order by o.factory.id,o.orderNo");

//厂家不同另起新页打印，除去第一次的比较if(oProduct.getFactory().getFactoryName().equals(ol
Factory)){ }

2）打印可以选择打印一款货物，还是两款货物if(contract.getPrintStyle().equals("2")){}

9、数据和业务分离

//填写每页的内容，之后在循环每页读取打印Map<String,String> pageMap = null;List<Map> pa
eList = new ArrayList(); //打印页

==报运打印：

wb.cloneSheet(0); //复制sheet0工作簿,名字会自动重命名

SpringMVC的POI实现方式

```
//前端部分----goodsService.js-----
this.exportExcel = function(){
window.open('../goods/export.do');
}
//前端部分----goodsController.js-----
$scope.exportExcel = function(){
goodsService.exportExcel();
}
//后端部分
// 1.创建工作簿
Workbook wb = new HSSFWorkbook();
// 2.创建工作表
Sheet sheet = wb.createSheet();
// 3.设置一些参数，设置一些公用变量, 列宽就是个bug
nRow = sheet.createRow(0);
// 4.给行设置值
nCell = nRow.createCell(0);
```

```
nCell.setCellValue("测试导出数据");
// 5.设置导出头信息,指定下载的文件名,写出excel
try {
response.setHeader("Content-Disposition","attachment;filename="+URLEncoder.encode("商
明细.xls","UTF-8"));
OutputStream output = response.getOutputStream();
wb.write(output);
output.flush();
output.close();
} catch (IOException e1) {
// TODO Auto-generated catch block
e1.printStackTrace();
}
```