POI中文API文档

一、POI简介

Apache POI是Apache软件基金会的开放源码函式库，POI提供API给Java程序对Microsoft Office格式档案读和写的功能。

二、 HSSF概况

HSSF 是Horrible SpreadSheet Format的缩写，通过HSSF，你可以用纯Java代码来读取、写入、修改Excel文件。HSSF 为读取操作提供了两类API：usermodel和eventusermodel，即“用户模型”和“事件-用户模型”。

三、 POI EXCEL文档结构类

HSSFWorkbook excel文档对象

HSSFSheet excel的sheet HSSFRow excel的行

HSSFCell excel的单元格 HSSFFont excel字体

HSSFName 名称 HSSFDataFormat 日期格式

HSSFHeader sheet头

HSSFFooter sheet尾

HSSFCellStyle cell样式

HSSFDateUtil 日期

HSSFPrintSetup 打印

HSSFErrorConstants 错误信息表

四、 EXCEL常用操作方法

从Excel文件中获取数据

```
POIFSFileSystem fs=newPOIFSFileSystem(new FileInputStream("d:/test.xls"));
//得到Excel工作簿对象
HSSFWorkbook wb = new HSSFWorkbook(fs);
//得到Excel工作表对象
HSSFSheet sheet = wb.getSheetAt(0);
//得到Excel工作表的行
HSSFRow row = sheet.getRow(i);
//得到Excel工作表指定行的单元格
HSSFCell cell = row.getCell((short) j);
//得到单元格样式
cellStyle = cell.getCellStyle();
```

通过代码创建Excle文件

```
//创建Excel工作簿对象
HSSFWorkbook wb = new HSSFWorkbook();
//创建Excel工作表对象
HSSFSheet sheet = wb.createSheet("new sheet");
//创建Excel工作表的行 下标从0开始
HSSFRow row = sheet.createRow(0);
//创建单元格样式
cellStyle = wb.createCellStyle();
//创建单元格
HSSFCell cell =row.createCell(0);
//为单元格设置样式 下标从0开始
cell.setCellStyle(cellStyle);
//为单元格设置值
cell.setCellValue(1);
```

设置sheet名称和单元格内容

```
wb.setSheetName(1, "第一张工作表",HSSFCell.ENCODING_UTF_16);
cell.setEncoding((short) 1);
cell.setCellValue("单元格内容");
```

获取excel文件中sheet的数目

```
wb.getNumberOfSheets()
```

根据index取得sheet对象

```
HSSFSheet sheet = wb.getSheetAt(0);
```

获取表中有效的行数

```
int rowcount = sheet.getLastRowNum();
```

取得一行的有效单元格个数

```
row.getLastCellNum();
```

单元格值类型读写

```
//设置单元格为STRING类型
cell.setCellType(HSSFCell.CELL_TYPE_STRING);
//读取为数值类型的单元格内容
cell.getNumericCellValue();
```

设置列宽、行高

```
sheet.setColumnWidth((short)column,(short)width);
row.setHeight((short)height);
```

添加区域，合并单元格

```
//合并从第rowFrom行columnFrom列
Region region = new Region((short)rowFrom,(short)columnFrom,(short)rowTo
,(short)columnTo);
// 到rowTo行columnTo的区域
sheet.addMergedRegion(region);
//得到所有区域
sheet.getNumMergedRegions()  12345
```

保存Excel文件

```
FileOutputStream fileOut = new FileOutputStream(path);
wb.write(fileOut);
```

根据单元格不同属性返回字符串数值

```
public String getCellStringValue(HSSFCell cell) {
        String cellValue = "";
        switch (cell.getCellType()) {
        case HSSFCell.CELL_TYPE_STRING://字符串类型
            cellValue = cell.getStringCellValue();
            if(cellValue.trim().equals("")||cellValue.trim().length()<=0)
                cellValue=" ";
            break;
        case HSSFCell.CELL_TYPE_NUMERIC: //数值类型
            cellValue = String.valueOf(cell.getNumericCellValue());
            break;
        case HSSFCell.CELL_TYPE_FORMULA: //公式
            cell.setCellType(HSSFCell.CELL_TYPE_NUMERIC);
            cellValue = String.valueOf(cell.getNumericCellValue());
            break;
        case HSSFCell.CELL_TYPE_BLANK:
            cellValue=" ";
            break;
        case HSSFCell.CELL_TYPE_BOOLEAN:
            break;
        case HSSFCell.CELL_TYPE_ERROR:
            break;
        default:
            break;
        }
        return cellValue;
```

```
    }
```

常用单元格边框格式

```
HSSFCellStyle style = wb.createCellStyle();
style.setBorderBottom(HSSFCellStyle.BORDER_DOTTED);//下边框
style.setBorderLeft(HSSFCellStyle.BORDER_DOTTED);//左边框
style.setBorderRight(HSSFCellStyle.BORDER_THIN);//右边框
style.setBorderTop(HSSFCellStyle.BORDER_THIN);//上边框
```

设置字体和内容位置

```
HSSFFont f  = wb.createFont();
f.setFontHeightInPoints((short) 11);//字号
f.setBoldweight(HSSFFont.BOLDWEIGHT_NORMAL);//加粗
style.setFont(f);
style.setAlignment(HSSFCellStyle.ALIGN_CENTER);//左右居中
style.setVerticalAlignment(HSSFCellStyle.VERTICAL_CENTER);//上下居中
style.setRotation(short rotation);//单元格内容的旋转的角度
HSSFDataFormat df = wb.createDataFormat();
style1.setDataFormat(df.getFormat("0.00%"));//设置单元格数据格式
cell.setCellFormula(string);//给单元格设公式
style.setRotation(short rotation);//单元格内容的旋转的角度
```

插入图片

```
//先把读进来的图片放到一个ByteArrayOutputStream中，以便产生ByteArray
      ByteArrayOutputStream byteArrayOut = new ByteArrayOutputStream();
      BufferedImage bufferImg = ImageIO.read(new File("ok.jpg"));
      ImageIO.write(bufferImg,"jpg",byteArrayOut);
//读进一个excel模版
FileInputStream fos = new FileInputStream(filePathName+"/stencil.xlt");
fs = new POIFSFileSystem(fos);
//创建一个工作薄
HSSFWorkbook wb = new HSSFWorkbook(fs);
HSSFSheet sheet = wb.getSheetAt(0);
HSSFPatriarch patriarch = sheet.createDrawingPatriarch();
HSSFClientAnchor anchor = new HSSFClientAnchor(0,0,1023,255,(short) 0,0,(short)10,10);

patriarch.createPicture(anchor ,
wb.addPicture(byteArrayOut.toByteArray(),HSSFWorkbook.PICTURE_TYPE_JPEG));
```

调整工作表位置

```
HSSFWorkbook wb = new HSSFWorkbook();
HSSFSheet sheet = wb.createSheet("format sheet");
HSSFPrintSetup ps = sheet.getPrintSetup();
sheet.setAutobreaks(true);
ps.setFitHeight((short)1);
ps.setFitWidth((short)1);
```

设置打印区域

```
HSSFSheet sheet = wb.createSheet("Sheet1");
wb.setPrintArea(0, "$A$1:$C$2");
```

标注脚注

```
HSSFSheet sheet = wb.createSheet("format sheet");
HSSFFooter footer = sheet.getFooter()
footer.setRight( "Page " + HSSFFooter.page() + " of " + HSSFFooter.numPages() );
```

在工作单中清空行数据，调整行位置

```
HSSFWorkbook wb = new HSSFWorkbook();
HSSFSheet sheet = wb.createSheet("row sheet");
// Create various cells and rows for spreadsheet.
// Shift rows 6 - 11 on the spreadsheet to the top (rows 0 - 5)
sheet.shiftRows(5, 10, -5);
```

选中指定的工作表

```
HSSFSheet sheet = wb.createSheet("row sheet");
heet.setSelected(true);
```

工作表的放大缩小

```
HSSFSheet sheet1 = wb.createSheet("new sheet");
sheet1.setZoom(1,2);   // 50 percent magnification
```

头注和脚注

```
HSSFSheet sheet = wb.createSheet("new sheet");
HSSFHeader header = sheet.getHeader();
header.setCenter("Center Header");
header.setLeft("Left Header");
header.setRight(HSSFHeader.font("Stencil-Normal", "Italic") +
HSSFHeader.fontSize((short) 16) + "Right w/ Stencil-Normal Italic font and size 16");
123456
```

自定义颜色

```
HSSFCellStyle style = wb.createCellStyle();
style.setFillForegroundColor(HSSFColor.LIME.index);
style.setFillPattern(HSSFCellStyle.SOLID_FOREGROUND);
HSSFFont font = wb.createFont();
font.setColor(HSSFColor.RED.index);
style.setFont(font);
cell.setCellStyle(style);
```

填充和颜色设置

```
HSSFCellStyle style = wb.createCellStyle();
style.setFillBackgroundColor(HSSFColor.AQUA.index);
style.setFillPattern(HSSFCellStyle.BIG_SPOTS);
HSSFCell cell = row.createCell((short) 1);
cell.setCellValue("X");
style = wb.createCellStyle();
style.setFillForegroundColor(HSSFColor.ORANGE.index);
style.setFillPattern(HSSFCellStyle.SOLID_FOREGROUND);
cell.setCellStyle(style);
```

强行刷新单元格公式

```
HSSFFormulaEvaluator eval=new HSSFFormulaEvaluator((HSSFWorkbook) wb);
private static void updateFormula(Workbook wb,Sheet s,int row){
        Row r=s.getRow(row);
        Cell c=null;
        FormulaEcaluator eval=null;
        if(wb instanceof HSSFWorkbook)
            eval=new HSSFFormulaEvaluator((HSSFWorkbook) wb);
        else if(wb instanceof XSSFWorkbook)
            eval=new XSSFFormulaEvaluator((XSSFWorkbook) wb);
        for(int i=r.getFirstCellNum();i
            c=r.getCell(i);
            if(c.getCellType()==Cell.CELL_TYPE_FORMULA)
                eval.evaluateFormulaCell(c);
        }
    }
```

> 说明：FormulaEvaluator提供了evaluateFormulaCell(Cell cell)方法，计算公式保存结果，但不改变公式。而evaluateInCell(Cell cell) 方法是计算公式，并将原公式替换为计算结果，也就是说该单元格的类型不在是Cell.CELL_TYPE_FORMULA而是Cell.CELL_TYPE_NUMBERIC。HSSFFormulaEvaluator提供了静态方法evaluateAllFormulaCells(HSSFWorkbook wb)，计算一个Excel文件的所有公式，用起来很方便。

设置不显示excel网格线

```
sheet.setDisplayGridlines(false);其中sheet是Sheet对象
```

设置excel单元格中的内容换行

```
cellStyle.setWrapText(true);//其中cellStyle是WorkBook创建的CellStyle对象,然后将cellStyle设置到要换行的Cell对象,最后在要换行的对象(一般为字符串)加入"/r/n"。如 topTile.append("/r/n" +"cellContent");
```

单元格的合并

```
sheet.addMergedRegion(new CellRangeAddress(0, 4, 0, 2));本示例为合并4行2列
```

设置页眉和页脚的页数

```
HSSFHeader header = sheet.getHeader();

header.setCenter("Center Header");

header.setLeft("Left Header");

header.setRight(HSSFHeader.font("Stencil-Normal", "Italic") +

HSSFHeader.fontSize((short) 16) + "Right w/ Stencil-Normal Italic font and size 16");

HSSFFooter footer = (HSSFFooter )sheet.getFooter()

footer.setRight( "Page " + HSSFFooter.page() + " of " + HSSFFooter.numPages() );
```

设置打印

```
HSSFPrintSetup print = (HSSFPrintSetup) sheet.getPrintSetup();

print.setLandscape(true);//设置横向打印

print.setScale((short) 70);//设置打印缩放70%
```

```
print.setPaperSize(HSSFPrintSetup.A4_PAPERSIZE);//设置为A4纸张

print.setLeftToRight(true);//設置打印顺序先行后列,默认为先列行

print.setFitHeight((short) 10);设置缩放调整为10页高

print.setFitWidth((short) 10);设置缩放调整为宽高

sheet.setAutobreaks(false);

if (i != 0 && i % 30 == 0)

sheet.setRowBreak(i);//設置每30行分頁打印
```

反复的行和列（设置打印标题）

```
HSSFWorkbook wb = new HSSFWorkbook();

wb.setRepeatingRowsAndColumns(0, 0, 12, 1, 6);//设置1到12列,行1到6每一页重复打印
```

调整单元格宽度

```
    sheet.setAutobreaks(true);

    sheet.setColumnWidth((short)i,colsWidth[i]); //设定单元格长度

sheet.autoSizeColumn((short) i);//自动根据长度调整单元格长度
```