POI

1. POI作用

Poi提供API给Java程序对Microsoft Office格式的文档进行读写操作

1. 包名称说明
   1. HSSF 提供读写Microsoft Excel XLS格式档案的功能。
   2. XSSF 提供读写Microsoft Excel OOXML XLSX格式档案的功能。
   3. HWPF 提供读写Microsoft Word DOC格式档案的功能。
   4. HSLF 提供读写Microsoft PowerPoint格式档案的功能。
   5. HDGF 提供读Microsoft Visio格式档案的功能。
   6. HPBF 提供读Microsoft Publisher格式档案的功能。
   7. HSMF 提供读Microsoft Outlook格式档案的功能
2. 常用类说明
   1. HSSFWorkbook Excel的文档对象
   2. HSSFSheet Excel的表单
   3. HSSFRow Excel的行
   4. HSSFCell Excel的格子单元
   5. HSSFFont Excel字体
   6. HSSFDataFormat 格子单元的日期格式
   7. HSSFHeader Excel文档Sheet的页眉
   8. HSSFFooter Excel文档Sheet的页脚
   9. HSSFCellStyle 格子单元样式
   10. HSSFDateUtil 日期
   11. HSSFPrintSetup 打印
   12. HSSFErrorConstants 错误信息表
3. Jar包说明
   1. Poi.jar

支持1997~03版本的excel的导入导出, 即\*.xls, 支持65535条数据

* 1. Poi-ooxml.jar

依赖于poi.jar，支持03以后版本的excel的导入导出, 即\*.xlsx，支持104W条数据，但可能导致内存溢出

1. HSSF和XSSF的理解
   1. HSSF和XSSF由不同的Jar支持
   2. HSSF支持03及之前版本的excel（\*.xls）， XSSF支持03以后版本的excel（\*.xlsx）
   3. 因为支持版本的不同，它们支持的数据量也随着excel版本而不同
2. XSSF对于大数据量导入导出的支持及OOM处理
   1. 对于大数据量的导入导出，可能会导致OOM的问题，32位JDK只支持2G内存，64位JDK不受限，但64位性能不如32位， 因此不好通过更改JVM内存来解决此问题
   2. XSSF通过控制excel数据占用内存量大小，当达到一定量时，自动写入，并刷新内存，实现只占用一定量的内存
   3. 但是，如果你是做导出功能，那么，如果数据库表中条目数极多，也可能导致OOM！但这并不是POI的问题了，而是在我们自身代码实现时，需要注意的，实施策略：
      1. 统计数据库表中条目数
      2. 合理设置提取数据次数（次数 = 条目数 / 每页数）
      3. 按次数写入数据库
3. 对于Excel的基本操作
   1. 创建Excel文档及表单
      1. HSSFWorkbook workbook = new HSSFWorkbook(); //创建Excel文件
      2. HSSFSheet sheet = workbook.createSheet(); //创建工作表(Sheet)
      3. sheet = workbook.createSheet("Test"); //工作表(Sheet)命名
      4. FileOutputStream out = new FileOutputStream(filePath);
      5. workbook.write(out); //写出Excel文件
      6. out.close(); //关闭文件流
   2. 创建单元格
      1. HSSFRow row = sheet.createRow(0); // 创建行,从0开始
      2. row.createCell(0).setCellValue("李志伟"); // 创建行的单元格并赋值,从0开始
      3. row.createCell(1).setCellValue(false); // 创建行的单元格并赋值
      4. row.createCell(2).setCellValue(new Date()); // 创建行的单元格并赋值
      5. row.createCell(3).setCellValue(12.345); // 创建行的单元格并赋值
   3. 创建文档摘要信息
      1. workbook.createInformationProperties(); //创建文档信息
      2. DocumentSummaryInformation dsi= workbook.getDocumentSummaryInformation();//摘要信息
      3. dsi.setCategory("类别:Excel文件"); //类别
      4. dsi.setManager("管理者:李志伟"); //管理者
      5. dsi.setCompany("公司:--"); //公司
      6. SummaryInformation si = workbook.getSummaryInformation();//摘要信息
      7. si.setSubject("主题:--"); //主题
      8. si.setTitle("标题:测试文档"); //标题
      9. si.setAuthor("作者:李志伟"); //作者
      10. si.setComments("备注:POI测试文档"); //备注
   4. 创建页眉和页脚
      1. HSSFHeader header =sheet.getHeader(); //得到页眉
      2. header.setLeft("页眉左边");
      3. header.setRight("页眉右边");
      4. header.setCenter("页眉中间");
      5. HSSFFooter footer =sheet.getFooter(); //得到页脚
      6. footer.setLeft("页脚左边");
      7. footer.setRight("页脚右边");
      8. footer.setCenter("页脚中间");
   5. 设置单元格格式

HSSFCell cell=row.createCell(0);

cell.setCellValue(new Date());

HSSFCellStyle style=workbook.createCellStyle();

style.setDataFormat(HSSFDataFormat.getBuiltinFormat("m/d/yy h:mm")); //Excel内嵌的格式, 设置日期格式

style.setDataFormat(HSSFDataFormat.getBuiltinFormat("0.00")); // Excel内嵌的格式, 保留2位小数

style.setDataFormat(workbook.createDataFormat().getFormat("￥#,##0")); //自定义, 货币格式

style.setDataFormat(workbook.createDataFormat().getFormat("0.00%")); //自定义, 百分比格式

style.setDataFormat(workbook.createDataFormat().getFormat("[DbNum2][$-804]0"));//自定义, 中文大写格式

style.setDataFormat(workbook.createDataFormat().getFormat("0.00E+00"));//自定义, 科学计数法格式

cell.setCellStyle(style); //将风格设置到单元格

1. 合并单元格

HSSFCell cell=row.createCell(0);

cell.setCellValue("合并列");

CellRangeAddress region=new CellRangeAddress(0, 0, 0, 5); //第1个单元格行号，第1个单元格列号

sheet.addMergedRegion(region); //第2个单元格行号，第2个单元格列号

1. 单元格风格设置
   1. style.setAlignment(HSSFCellStyle.ALIGN\_CENTER); //水平居中
   2. style.setVerticalAlignment(HSSFCellStyle.VERTICAL\_CENTER); //垂直居中
   3. style.setWrapText(true); //自动换行
   4. style.setIndention((short)5); //缩进
   5. style.setRotation((short)60); //文本旋转，这里的取值是从-90到90，而不是0-180度
   6. style.setBorderTop(HSSFCellStyle.BORDER\_DOTTED); //上边框
   7. style.setBorderBottom(HSSFCellStyle.BORDER\_THICK); //下边框
   8. style.setBorderLeft(HSSFCellStyle.BORDER\_DOUBLE); //左边框
   9. style.setBorderRight(HSSFCellStyle.BORDER\_SLANTED\_DASH\_DOT); //右边框
   10. style.setTopBorderColor(HSSFColor.RED.index); //上边框颜色
   11. style.setBottomBorderColor(HSSFColor.BLUE.index); //下边框颜色
   12. style.setLeftBorderColor(HSSFColor.GREEN.index); //左边框颜色
   13. style.setRightBorderColor(HSSFColor.PINK.index); //右边框颜色
2. 字体设置

HSSFCellStyle style = workbook.createCellStyle();

HSSFFont font = workbook.createFont();

font.setFontName("华文行楷"); //设置字体名称

font.setFontHeightInPoints((short)28); //设置字号

font.setColor(HSSFColor.RED.index); //设置字体颜色

font.setUnderline(FontFormatting.U\_SINGLE); //设置下划线

font.setTypeOffset(FontFormatting.SS\_SUPER); //设置上标下标

font.setStrikeout(true); //设置删除线

style.setFont(font);

cell.setCellStyle(style);

1. 设置调度与宽度
   1. sheet.setColumnWidth(1, 31 \* 256); //设置第1列的宽度是31个字符宽度
   2. row.setHeightInPoints(50); //设置行的高度是50个点
2. 网格线
   1. sheet1.setDisplayGridlines(false); //隐藏Excel网格线,默认值为true
   2. sheet2.setGridsPrinted(true); //打印时显示网格线,默认值为false