[**java操作excel**](http://www.cnblogs.com/sunzhenxing19860608/archive/2010/12/27/1918128.html)

Posted on 2010-12-27 17:34 [我是孙海龙](http://www.cnblogs.com/sunzhenxing19860608/) 阅读(54188) 评论(13) [编辑](http://i.cnblogs.com/EditPosts.aspx?postid=1918128) [收藏](http://www.cnblogs.com/sunzhenxing19860608/archive/2010/12/27/1918128.html)

jxl是一个韩国人写的java操作excel的工具, 在开源世界中，有两套比较有影响的API可供使用，一个是POI，一个是jExcelAPI。其中功能相对POI比较弱一点。但jExcelAPI对中文支持非常好，API是纯Java的， 并不依赖Windows系统，即使运行在Linux下，它同样能够正确的处理Excel文件。 另外需要说明的是，这套API对图形和图表的支持很有限，而且仅仅识别PNG格式。

使用如下：

搭建环境

 将下载后的文件解包，得到jxl.jar，放入classpath，安装就完成了。

 基本操作

 一、创建文件

 拟生成一个名为“test.xls”的Excel文件，其中第一个工作表被命名为  
 “第一页”，大致效果如下：

http://www.blogjava.net/Images/OutliningIndicators/None.gifpackage test;  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif//生成Excel的类  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport java.io.File;  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.Workbook;  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.write.Label;  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.write.WritableSheet;  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.write.WritableWorkbook;  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockStart.gifpublic class CreateExcel {  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif    public static void main(String args[]) {  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif        try {  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 打开文件  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            WritableWorkbook book = Workbook.createWorkbook(new File("test.xls"));  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 生成名为“第一页”的工作表，参数0表示这是第一页  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            WritableSheet sheet = book.createSheet("第一页", 0);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 在Label对象的构造子中指名单元格位置是第一列第一行(0,0)  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 以及单元格内容为test  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            Label label = new Label(0, 0, "test");  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 将定义好的单元格添加到工作表中  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            sheet.addCell(label);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif            /\*  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif             \* 生成一个保存数字的单元格 必须使用Number的完整包路径，否则有语法歧义 单元格位置是第二列，第一行，值为789.123  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif             \*/  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            jxl.write.Number number = new jxl.write.Number(1, 0, 555.12541);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            sheet.addCell(number);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 写入数据并关闭文件  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            book.write();  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            book.close();  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif        } catch (Exception e) {  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            System.out.println(e);  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.blogjava.net/Images/OutliningIndicators/None.gif

   编译执行后，会产生一个Excel文件。

 三、读取文件

 以刚才我们创建的Excel文件为例，做一个简单的读取操作，程序代码如下：

http://www.blogjava.net/Images/OutliningIndicators/None.gifpackage test;  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif//读取Excel的类  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport java.io.File;  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.Cell;  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.Sheet;  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.Workbook;  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockStart.gifpublic class ReadExcel {  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif    public static void main(String args[]) {  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif        try {  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            Workbook book = Workbook.getWorkbook(new File("test.xls"));  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 获得第一个工作表对象  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            Sheet sheet = book.getSheet(0);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 得到第一列第一行的单元格  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            Cell cell1 = sheet.getCell(0, 0);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            String result = cell1.getContents();  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            System.out.println(result);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            book.close();  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif        } catch (Exception e) {  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            System.out.println(e);  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.blogjava.net/Images/OutliningIndicators/None.gif

  程序执行结果：test

 四、修改文件  
 利用jExcelAPI可以修改已有的Excel文件，修改Excel文件的时候，除了打开文件的方式不同之外，  
 其他操作和创建Excel是一样的。下面的例子是在我们已经生成的Excel文件中添加一个工作表：

http://www.blogjava.net/Images/OutliningIndicators/None.gifpackage test;  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport java.io.File;  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.Workbook;  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.write.Label;  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.write.WritableSheet;  
http://www.blogjava.net/Images/OutliningIndicators/None.gifimport jxl.write.WritableWorkbook;  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockStart.gifpublic class UpdateExcel {  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif    public static void main(String args[]) {  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif        try {  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // Excel获得文件  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            Workbook wb = Workbook.getWorkbook(new File("test.xls"));  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 打开一个文件的副本，并且指定数据写回到原文件  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            WritableWorkbook book = Workbook.createWorkbook(new File("test.xls"),  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif                    wb);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            // 添加一个工作表  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            WritableSheet sheet = book.createSheet("第二页", 1);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            sheet.addCell(new Label(0, 0, "第二页的测试数据"));  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            book.write();  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            book.close();  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif        } catch (Exception e) {  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            System.out.println(e);  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.blogjava.net/Images/OutliningIndicators/None.gif

其他操作

 一、 数据格式化

 在Excel中不涉及复杂的数据类型，能够比较好的处理字串、数字和日期已经能够满足一般的应用。

 1、 字串格式化

 字符串的格式化涉及到的是字体、粗细、字号等元素，这些功能主要由WritableFont和  
 WritableCellFormat类来负责。假设我们在生成一个含有字串的单元格时，使用如下语句，  
 为方便叙述，我们为每一行命令加了编号：

http://www.blogjava.net/Images/OutliningIndicators/None.gifWritableFont font1=  
http://www.blogjava.net/Images/OutliningIndicators/None.gif new WritableFont(WritableFont.TIMES,16,WritableFont.BOLD); ①  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif WritableCellFormat format1=new WritableCellFormat(font1); ②  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif Label label=new Label(0,0,”data 4 test”,format1) ③  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif

  其中①指定了字串格式：字体为TIMES，字号16，加粗显示。WritableFont有非常丰富的  
 构造子，供不同情况下使用，jExcelAPI的java-doc中有详细列表，这里不再列出。

 ②处代码使用了WritableCellFormat类，这个类非常重要，通过它可以指定单元格的各种  
 属性，后面的单元格格式化中会有更多描述。

 ③处使用了Label类的构造子，指定了字串被赋予那种格式。

 在WritableCellFormat类中，还有一个很重要的方法是指定数据的对齐方式，比如针对我们  
 上面的实例，可以指定：

http://www.blogjava.net/Images/OutliningIndicators/None.gif //把水平对齐方式指定为居中  
http://www.blogjava.net/Images/OutliningIndicators/None.gif format1.setAlignment(jxl.format.Alignment.CENTRE);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif //把垂直对齐方式指定为居中  
http://www.blogjava.net/Images/OutliningIndicators/None.gif format1.setVerticalAlignment(jxl.format.VerticalAlignment.CENTRE);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif

二、单元格操作

 Excel中很重要的一部分是对单元格的操作，比如行高、列宽、单元格合并等，所幸jExcelAPI  
 提供了这些支持。这些操作相对比较简单，下面只介绍一下相关的API。

 1、 合并单元格

http://www.blogjava.net/Images/OutliningIndicators/None.gif WritableSheet.mergeCells(int m,int n,int p,int q);   
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif //作用是从(m,n)到(p,q)的单元格全部合并，比如：  
http://www.blogjava.net/Images/OutliningIndicators/None.gif WritableSheet sheet=book.createSheet(“第一页”,0);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif //合并第一列第一行到第六列第一行的所有单元格  
http://www.blogjava.net/Images/OutliningIndicators/None.gif sheet.mergeCells(0,0,5,0);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif

 合并既可以是横向的，也可以是纵向的。合并后的单元格不能再次进行合并，否则会触发异常。

 2、 行高和列宽

http://www.blogjava.net/Images/OutliningIndicators/None.gif WritableSheet.setRowView(int i,int height);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif //作用是指定第i+1行的高度，比如：  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif //将第一行的高度设为200  
http://www.blogjava.net/Images/OutliningIndicators/None.gif sheet.setRowView(0,200);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif WritableSheet.setColumnView(int i,int width);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif //作用是指定第i+1列的宽度，比如：  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  
http://www.blogjava.net/Images/OutliningIndicators/None.gif //将第一列的宽度设为30  
http://www.blogjava.net/Images/OutliningIndicators/None.gif sheet.setColumnView(0,30);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif

 jExcelAPI还有其他的一些功能，比如插入图片等，这里就不再一一介绍，读者可以自己探索。

其中：如果读一个excel，需要知道它有多少行和多少列，如下操作：

http://www.blogjava.net/Images/OutliningIndicators/None.gifWorkbook book = Workbook.getWorkbook(new File("测试1.xls"));  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        // 获得第一个工作表对象  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        Sheet sheet = book.getSheet(0);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        // 得到第一列第一行的单元格  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        int columnum = sheet.getColumns();// 得到列数  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        int rownum = sheet.getRows();// 得到行数  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        System.out.println(columnum);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        System.out.println(rownum);  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        for (int i = 0; i < rownum; i++)// 循环进行读写  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockStart.gif        {  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockStart.gif            for (int j = 0; j < columnum; j++) {  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif                Cell cell1 = sheet.getCell(j, i);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif                String result = cell1.getContents();  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif                System.out.print(result);  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif                System.out.print("\t");  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.blogjava.net/Images/OutliningIndicators/InBlock.gif            System.out.println();  
http://www.blogjava.net/Images/OutliningIndicators/ExpandedBlockEnd.gif        }  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        book.close();

标签: [java](http://www.cnblogs.com/sunzhenxing19860608/tag/java/), [excel](http://www.cnblogs.com/sunzhenxing19860608/tag/excel/)