POI+XOM+Java操作Excel与XML之间转换

ExcelXMLJavaApacheJ#

今天了解一POI和XOM，在Excel和XMl文件之间互换，做了一个例子，但没有重构，重构以后再做，请各位高手讲讲，怎么样重构是最优的。

Java代码

/\*\*

\*

\* ExcelXML.java

\* IBM\_Developer\_POI(Excel,Word)

\* 2010-6-3 上午08:14:59

\*/

**package** com.wds.excelxml;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.text.NumberFormat;

**import** java.text.ParseException;

**import** org.apache.poi.hssf.usermodel.HSSFCell;

**import** org.apache.poi.hssf.usermodel.HSSFCellStyle;

**import** org.apache.poi.hssf.usermodel.HSSFDataFormat;

**import** org.apache.poi.hssf.usermodel.HSSFDataFormatter;

**import** org.apache.poi.hssf.usermodel.HSSFHyperlink;

**import** org.apache.poi.hssf.usermodel.HSSFRow;

**import** org.apache.poi.hssf.usermodel.HSSFSheet;

**import** org.apache.poi.hssf.usermodel.HSSFWorkbook;

**import** nu.xom.Attribute;

**import** nu.xom.Document;

**import** nu.xom.Element;

**import** nu.xom.Elements;

**import** nu.xom.Serializer;

/\*\*

\* 2010-6-3 上午08:14:59

\*/

**public** **class** ExcelXML {

/\*\*

\* 2010-6-3 上午08:14:59

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

excelXML();

}

/\*\*

\* 从Excel到XML

\* 从XML到Excel

\* 2010-6-3 下午07:30:44

\*/

**private** **static** **void** excelXML(){

/\*

\* 首先创建一个XML文档

\* 要创建XML文档，首先创建一个根元素

\*/

Element reportRoot=**new** Element("sheet");

Document xmlReport=**new** Document(reportRoot);

**try** {

//读取Excel文件

FileInputStream excelFIS=**new** FileInputStream("D:\\JavaTest\\Employee\_List.xls");

//创建Excel工作表

HSSFWorkbook excelWB=**new** HSSFWorkbook(excelFIS);

//获得Excel工作簿

HSSFSheet excelSheet=excelWB.getSheetAt(0);

//获得工作簿的行数

**int** rows=excelSheet.getPhysicalNumberOfRows();

//遍历工作簿的行

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

HSSFRow oneRow=excelSheet.getRow(rowIndex);

**if**(oneRow==**null**){

**continue**;

}

//在迭代每一行的时候，创建xml的行元素

Element rowElement=**new** Element("row");

//获得当前行的单元格数

**int** cells=oneRow.getPhysicalNumberOfCells();

//遍历行中的每一个单元格

**for**(**int** cellIndex=0;cellIndex<cells;cellIndex++){

HSSFCell oneCell=oneRow.getCell(cellIndex);

**if**(oneCell==**null**){

**continue**;

}

//设置元素的默认名称

String elementName="header";

//获得单元格所在列位置

**int** cellColumnIndex=oneCell.getColumnIndex();

**if**(rowIndex>0){

elementName=reportRoot.getFirstChildElement("row").getChild(cellColumnIndex).getValue();

}

/\*

\* 去掉非法字符

\*/

elementName = elementName.replaceAll("[\\P{ASCII}]","");

elementName = elementName.replaceAll(" ", "");

Element cellElement = **new** Element(elementName);

//添加属性和元素

//String attributeValue=oneCell.getCellStyle().getDataFormatString();

//Attribute dataFormatAttribute=new Attribute("dataFormat", attributeValue);

//cellElement.addAttribute(dataFormatAttribute);

/\*

\* 根据不同的属性添加

\*/

Attribute strTypeAttribute=**null**;

**switch** (oneCell.getCellType()) {

**case** HSSFCell.CELL\_TYPE\_STRING:

strTypeAttribute=**new** Attribute("dataType","String");

cellElement.addAttribute(strTypeAttribute);

cellElement.appendChild(oneCell.getStringCellValue());

rowElement.appendChild(cellElement);

**break**;

**case** HSSFCell.CELL\_TYPE\_NUMERIC:

strTypeAttribute=**new** Attribute("dataType","Numeric");

cellElement.addAttribute(strTypeAttribute);

HSSFDataFormatter dataFormatter=**new** HSSFDataFormatter();

String cellFormatted=dataFormatter.formatCellValue(oneCell);

cellElement.appendChild(cellFormatted);

rowElement.appendChild(cellElement);

**break**;

}

}

**if**(rowElement.getChildCount()>0){

reportRoot.appendChild(rowElement);

}

//System.out.println(xmlReport.toXML());

}

/\*

\* 计算薪水的 1% 并存储到一个 donation 元素中

\*/

Elements rowElements=reportRoot.getChildElements("row");

//遍历每一个行元素

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

/\*

\* 第一行元素是头，不计算

\* 计算薪水的1%存储到donation的节点上

\*/

**if**(i==0){

Element donationElement=**new** Element("header");

donationElement.appendChild("Donation");

Attribute dataType=**new** Attribute("dataType", "String");

donationElement.addAttribute(dataType);

Attribute dataFormat=**new** Attribute("dataFormat", "General");

donationElement.addAttribute(dataFormat);

rowElements.get(i).appendChild(donationElement);

}**else**{

Element donationElement=**new** Element("donation");

//添加数据类型的属性

Attribute dataType=**new** Attribute("dataType","Numeric");

donationElement.addAttribute(dataType);

//添加数据格式的属性

Attribute dataFormat=**new** Attribute("dataFormat", "#,##0");

donationElement.addAttribute(dataFormat);

//获得当前行的薪水

Element salaryElement=rowElements.get(i).getFirstChildElement("Salary");

String salaryString=salaryElement.getValue();

//转换薪水格式

NumberFormat numberFormat=NumberFormat.*getInstance*();

Number salaryNumber=numberFormat.parse(salaryString);

//计算薪水的1%

**double** donationSalary=salaryNumber.doubleValue()\*0.01;

//添加薪水节点

donationElement.appendChild(Double.*toString*(donationSalary));

rowElements.get(i).appendChild(donationElement);

}//End else

}//End for(int i=0;i<rowElements.size();i++)

//屏幕输出XML文件

System.*out*.println(xmlReport.toXML());

/\*

\* 输出到文件

\*/

FileOutputStream xmlFOS=**new** FileOutputStream("D:\\JavaTest\\Employee\_List.xml");

//创建Serializer,输出文件

Serializer saveXMLSerializer=**new** Serializer(xmlFOS);

//saveXMLSerializer.setIndent(5);

saveXMLSerializer.write(xmlReport);

/\*

\* 将XML写回Excel

\*/

//创建工作表

HSSFWorkbook donationWorkbook=**new** HSSFWorkbook();

HSSFSheet donationSheet=donationWorkbook.createSheet();

//迭代行元素 outer loop

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

HSSFRow donationRow=donationSheet.createRow(i);

Elements cellElements=rowElements.get(i).getChildElements();

//inner loop

**for**(**int** j=0;j<cellElements.size();j++){

HSSFCell createdCell=donationRow.createCell(j);

//获得数据类型的属性值

String dataFormatString=cellElements.get(j).getAttributeValue("dataType");

//设置数据格式

HSSFCellStyle currentStyle=donationWorkbook.createCellStyle();

HSSFDataFormat currentDataFormat=donationWorkbook.createDataFormat();

**short** dataFormatIndex=currentDataFormat.getFormat(dataFormatString);

currentStyle.setDataFormat(dataFormatIndex);

createdCell.setCellStyle(currentStyle);

//根据数据类型写入数据

//如果数据类型是String

**if**(cellElements.get(j).getAttribute("dataType").toString().equals("String")){

createdCell.setCellType(HSSFCell.CELL\_TYPE\_STRING);

createdCell.setCellValue(cellElements.get(j).getValue());

}

//数据类型是Numeric

**if**(cellElements.get(j).getAttribute("dataType").toString().equals("Numeric")){

createdCell.setCellType(HSSFCell.CELL\_TYPE\_NUMERIC);

//判断数据格式

**if**(cellElements.get(j).getAttribute("dataFormat").toString().contains("#")){

//转换格式

NumberFormat numberFormat=NumberFormat.*getInstance*();

Number cellValueNumber=numberFormat.parse(cellElements.get(j).getValue());

createdCell.setCellValue(cellValueNumber.doubleValue());

HSSFHyperlink hyperlink=**new** HSSFHyperlink(HSSFHyperlink.LINK\_URL);

hyperlink.setAddress("Http://www.hbut.edu.cn");

createdCell.setHyperlink(hyperlink);

}

}**else**{

createdCell.setCellValue(cellElements.get(j).getValue());

}//End if

}//End inner loop

}//End outer loop

FileOutputStream excelFOS=**new** FileOutputStream("D:\\JavaTest\\Employee\_Donation.et");

donationWorkbook.write(excelFOS);

} **catch** (FileNotFoundException e) {

e.printStackTrace();

} **catch** (IOException e) {

e.printStackTrace();

} **catch** (ParseException e) {

e.printStackTrace();

}

}

}