

## 转 Apache POI （JAVA处理Office文档的类库）

2013年10月08日 23:34:48 阅读数：2805

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。

结构：

HSSF — 提供读写Microsoft Excel格式档案的功能。

XSSF — 提供读写Microsoft Excel OOXML格式档案的功能。

HWPf — 提供读写Microsoft Word格式档案的功能。

HSLF — 提供读写Microsoft PowerPoint格式档案的功能。

HDGF — 提供读写Microsoft Visio格式档案的功能。

### 创建Excel 文档

示例1将演示如何利用Jakarta POI API 创建Excel 文档。

示例1程序如下：

```
[cpp]
1. public class CreateXL {
2.     /** Excel 文件要存放的位置，假定在D盘下*/
3.     public static String outputFile="D:\\test.xls";
4.     public static void main(String argv[]){
5.     try{
6.         // 创建新的Excel 工作簿
7.         HSSFWorkbook workbook = new HSSFWorkbook();
8.         // 在Excel工作簿中建一工作表，其名为缺省值
9.         // 如要新建一名为"效益指标"的工作表，其语句为：
10.        // HSSFSheet sheet = workbook.createSheet("效益指标");
11.        HSSFSheet sheet = workbook.createSheet();
12.        // 在索引0的位置创建行（最顶端的行）
13.        HSSFRow row = sheet.createRow((short)0);
14.        //在索引0的位置创建单元格（左上端）
15.        HSSFCell cell = row.createCell((short) 0);
16.        // 定义单元格为字符串类型
17.        cell.setCellType(HSSFCell.CELL_TYPE_STRING);
18.        // 在单元格中输入一些内容
19.        cell.setCellValue("增加值");
20.        // 新建一输出文件流
21.        FileOutputStream fOut = new FileOutputStream(outputFile);
22.        // 把相应的Excel 工作簿存盘
23.        workbook.write(fOut);
24.        fOut.flush();
25.        // 操作结束，关闭文件
26.        fOut.close();
27.        System.out.println("文件生成...");
28.    }catch(Exception e) {
29.        System.out.println("已运行 xlCreate() : " + e );
30.    }
31. }
```

### 读取Excel文档中的数据

示例2将演示如何读取Excel文档中的数据。假定在D盘JTest目录下有一个文件名为test1.xls的Excel文件。

示例2程序如下：

```
[java]
1. public class ReadXL {
2.     /** Excel文件的存放位置。注意是正斜线*/
3.     public static String fileToBeRead="D:\\test1.xls";
4.     public static void main(String argv[]){
5.     try{
6.         // 创建对Excel工作簿文件的引用
7.         HSSFWorkbook workbook = new HSSFWorkbook(new FileInputStream(fileToBeRead));
8.         // 创建对工作表的引用。
9.         // 本例是按名引用（让我们假定那张表有着缺省名"Sheet1"）
10.        HSSFSheet sheet = workbook.getSheet("Sheet1");
11.        // 也可用getSheetAt(int index)按索引引用，
12.        // 在Excel文档中，第一张工作表的缺省索引是0，
13.        // 其语句为：HSSFSheet sheet = workbook.getSheetAt(0);
14.        // 读取左上端单元
15.        HSSFRow row = sheet.getRow(0);
16.        HSSFCell cell = row.getCell((short)0);
17.        // 输出单元内容，cell.getStringCellValue()就是取所在单元的值
18.        System.out.println("左上端单元是： " + cell.getStringCellValue());
19.    }catch(Exception e) {
20.        System.out.println("已运行xlRead() : " + e );
21.    }
22. }
```

详细使用方法见《Busy Developers' Guide to HSSF and XSSF Features》<http://poi.apache.org/spreadsheet/quick-guide.html>

### 处理WORD文档

[cpp]  

```
1. public class TestPoi {
2.     public TestPoi() {
3.     }
4.     public static void main(String args[]) throws Exception
5.     {
6.         FileInputStream in = new FileInputStream ("D:\\a.doc");
7.         WordExtractor extractor = new WordExtractor();
8.         String str = extractor.extractText(in);
9.         //System.out.println("the result length is"+str.length());
10.        System.out.println(str);
11.    }
12. }
```

文章标签：

[Apache](#)

[POI](#)

[office](#)

[文档](#)

[java](#)

个人分类：[J2EE](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com