## java操作excle表格

目录

[一:读取表格：](#_Toc32558_WPSOffice_Level1) [1](#_Toc32558_WPSOffice_Level1)

[1. 选取Excle文件得到工作簿](#_Toc26115_WPSOffice_Level2) [1](#_Toc26115_WPSOffice_Level2)

[2. 选择工作表](#_Toc8039_WPSOffice_Level2) [1](#_Toc8039_WPSOffice_Level2)

[3. 读取Cell](#_Toc17154_WPSOffice_Level2) [1](#_Toc17154_WPSOffice_Level2)

[4. 读取信息](#_Toc30990_WPSOffice_Level2) [1](#_Toc30990_WPSOffice_Level2)

[1.1读取工作簿](#_Toc26115_WPSOffice_Level3) [1](#_Toc26115_WPSOffice_Level3)

[1.2读取工作铺](#_Toc8039_WPSOffice_Level3) [1](#_Toc8039_WPSOffice_Level3)

[1.3读取单元格](#_Toc17154_WPSOffice_Level3) [1](#_Toc17154_WPSOffice_Level3)

[1.4读取单元格的值](#_Toc30990_WPSOffice_Level3) [1](#_Toc30990_WPSOffice_Level3)

[1.5判断单元格的类型信息](#_Toc962_WPSOffice_Level3) [1](#_Toc962_WPSOffice_Level3)

[1.6释放资源](#_Toc28364_WPSOffice_Level3) [2](#_Toc28364_WPSOffice_Level3)

[二：写excle表格](#_Toc26115_WPSOffice_Level1) [2](#_Toc26115_WPSOffice_Level1)

[1. 创建Excle工作簿](#_Toc962_WPSOffice_Level2) [2](#_Toc962_WPSOffice_Level2)

[2. 创建工作表](#_Toc28364_WPSOffice_Level2) [2](#_Toc28364_WPSOffice_Level2)

[3. 创建单元格](#_Toc5934_WPSOffice_Level2) [2](#_Toc5934_WPSOffice_Level2)

[2.1.创建可写入的Excle工作簿](#_Toc5934_WPSOffice_Level3) [2](#_Toc5934_WPSOffice_Level3)

[2.2创建工作表](#_Toc189_WPSOffice_Level3) [2](#_Toc189_WPSOffice_Level3)

[2.3创建单元格](#_Toc32349_WPSOffice_Level3) [2](#_Toc32349_WPSOffice_Level3)

[2.4 拷贝、更新Excel工作薄 //创建只读的Excel工作薄的对象](#_Toc28482_WPSOffice_Level3) [3](#_Toc28482_WPSOffice_Level3)

一:读取表格：

|  |
| --- |
| 通过Workbook，Sheet ，Cell这三个对象我们就可以实现Excel文件的读取工作 |

1. 选取Excle文件得到工作簿
2. 选择工作表
3. 读取Cell
4. 读取信息

1.1读取工作簿

|  |
| --- |
| 选取Excle文件得到工作簿workbook  Workbook workbok=Workbook.getWorkbook(new File(“myfirst.xls”)); |

1.2读取工作铺

|  |
| --- |
| 通过Workbook的getSheet方法选择第一个工作表（从0开始）  Sheet sheet=workbook.getSheet(0); |

1.3读取单元格

|  |
| --- |
| 通过Sheet的getCell方法选择位置为c2的单元格（两个参数都是从0开始）  Cell c2=sheet.getCell(2.,1); |

1.4读取单元格的值

|  |
| --- |
| 通过Cell的getContents()方法，把单元格中的信息以字符串的形式读取出来  String string2=c2.getContents(); |

1.5判断单元格的类型信息

|  |
| --- |
| LabelCell，NumberCell，DateCell分别表示字符、数值、日期类型的单元格  if (c2.getType() == CellType. LABEL)  {  LabelCell nc = (LabelCell) c2;  String number b2 = nc. getString(); }  if (c2.getType() == CellType. DATE)  {  DateCell nc = (DateCell) c2;  Date number b2 = nc. getDate(); }  if (c2.getType() == CellType.NUMBER)  {  NumberCell nc = (NumberCell) c2;  double number b2 = nc.getValue(); } |

1.6释放资源

|  |
| --- |
| workbook.close() |

二：写excle表格

|  |
| --- |
| 通过WritableWorkbook，WritableSheet，Label这三个对象我们就可以实现Excel文件的插入工作. |

1. 创建Excle工作簿
2. 创建工作表
3. 创建单元格

2.1.创建可写入的Excle工作簿

|  |
| --- |
| WritableWorkbook wwb=Workbook.createWorkbook(new File(“路径”)); |

2.2创建工作表

|  |
| --- |
| WritableSheet ws=wwb.createSheet(“通讯录”,0)//创建sheet |

2.3创建单元格

2.3.1添加文本类单元格

|  |
| --- |
| Label labelC=new Label(0,0,”第一个”);  ws.addCell(labelC); |

2.3.2添加带有字型Formatting的对象

|  |
| --- |
| WritableFont wf=new WritableFont(WritableFont.TIMES, 18, WritableFont.BOLD, true)  WritableCellFormat wctf=new WritableCellFormat(wf);  Label cf=new Label(1,0,”第二个”);  ws.addCell(cf); |

2.3.3添加带有字体颜色Formatting的对象

|  |
| --- |
| WritableFont wfc = new WritableFont(WritableFont.ARIAL,10,WritableFont.NO\_BOLD, false, UnderlineStyle.NO\_UNDERLINE, jxl.format.Colour.RED);  WritableCellFormat wcfFC = new WritableCellFormat(wfc);  Label labelCFC = new Label(1, 0, "This is a Label Cell", wcfFC);  ws.addCell(labelCF); |

2.3.4添加Number对象

|  |
| --- |
| Number labelN=new jxl.writable.Number(0,1,3.1415926) |

2.3.5添加Boolean对象

|  |
| --- |
| Boolean labelb=new jxl.writableBoolean(0,2,false)  ws.addCell(labelb); |

2.3.6 添加DateTime对象

|  |
| --- |
| DateTime labelDT = new DateTime(0, 3, new java.util.Date()); ws.addCell(labelDT); |

2.3.7 添加带有formatting的DateFormat对象

|  |
| --- |
| DateFormat df = new DateFormat("dd MM yyyy hh:mm:ss");  WritableCellFormat wcfDF = new WritableCellFormat(df);  DateTime labelDTF = new DateTime(1, 3, new Date(), wcfDF); ws.addCell(labelDTF); |

2.3.8 添加公式单元格

|  |
| --- |
| Fornual formual = new Formual(0,11,”Sum(A1:A9)”);  wrb.addCell(formual); |

2.3.9添加图片

|  |
| --- |
| WritableImage wrimage=new WritableImage(4,0,2,1,new File("D:\\"+id+".png"));  WritableSheet ws = wwb.createSheet("sheet0", 0);  ws.setRowView(0, 1700, false); //设置行高  // 把图片插入到sheet  ws.addImage(wrimage); |

2.3.10合并单元格

|  |
| --- |
| ws.mergeCells(0, 1, 10, 1); |

2.3.5 添加单元格样式 主要是改变单元格背景、字体、颜色

|  |
| --- |
| WritableCellFormat wc = new WritableCellFormat();  wc.setAlignment(Alignment.CENTRE); // 设置居中  wc.setBorder(Border.ALL,BorderLineStyle.THIN);//设置边框线  wc.setBackground(jxl.format.Colour.RED); // 设置单元格的背景颜色  label = new Label(1, 5, "字体", wc); sheet.addCell(label); |

2.3.6 设置单元格字体

|  |
| --- |
| WritableFont wfont =new WritableFont(WritableFont.createFont("楷书"), 20); WritableCellFormat font = new WritableCellFormat(wfont); label = new Label(2, 6, "楷书", font); sheet.addCell(label); |

2.3.7写入文件

|  |
| --- |
| wwb.write();// 写入数据 wwb.close();// 关闭文件 |

2.4 拷贝、更新Excel工作薄 //创建只读的Excel工作薄的对象

|  |
| --- |
| jxl.Workbook rw = jxl.Workbook.getWorkbook(new File(sourcefile)); //创建可写入的Excel工作薄对象  WritableWorkbook wwb=Workbook.createWorkbook(new File(targetfile), rw); //读取第一张工作表  jxl.write.WritableSheet ws = wwb.getSheet(0); |

来源：<https://blog.csdn.net/lalioCAT/article/details/50580020>

|  |
| --- |
| // 导出预约活动学生数据  public String exportActStudent() {  try {  System.out.println("名单导出");  WritableWorkbook wwb = null;    // 默认保存在d盘  String path = ServletActionContext.getServletContext().getRealPath("/excleFile");  File file = new File(path + "/mingdan.xls");  // System.out.println(file);  if (!file.exists()) {  file.createNewFile();  }  //添加图片  wwb = Workbook.createWorkbook(file);  WritableImage wrimage=new WritableImage(4,0,2,1,new File("D:\\"+id+".png"));  WritableSheet ws = wwb.createSheet("sheet0", 0);  ws.setRowView(0, 1700, false); //设置行高  // 把图片插入到sheet  ws.addImage(wrimage);  //合并单元格  ws.mergeCells(0, 1, 10, 1);  //设置字体居中  WritableCellFormat wc = new WritableCellFormat();  wc.setAlignment(Alignment.CENTRE); // 设置居中  WritableFont wfont =new WritableFont(WritableFont.createFont("楷书"), 20);  WritableCellFormat font = new WritableCellFormat(wfont);    List<Activiting> list = actRecordDao.JionActStudent(id);  DateFormat sdf = new SimpleDateFormat("MM/dd");  //注意format的格式要与日期String的格式相匹配  String tsStr = sdf.format(list.get(0).getAFirsttime());  System.out.println(tsStr);  // tsStr+=actR.getActivity().getRName()+" ";  activity=activityDao.findone(id);  System.out.println(activity.getRName());  String sad="活动名称: "+activity.getRName()+"活动时间： "+tsStr;  Label labelTitle = new Label(0, 1, sad,font);    Label labelSId = new Label(0, 2, "学号");  Label laelName = new Label(1, 2, "姓名");  Label labelSchool = new Label(2, 2, "学院");  Label labelClass = new Label(3, 2, "班级");  Label labelPhone = new Label(4, 2, "联系方式");  Label labelQian=new Label(5,2,"签到");  ws.addCell(labelTitle);  ws.addCell(labelSId);  ws.addCell(laelName);  ws.addCell(labelSchool);  ws.addCell(labelClass);  ws.addCell(labelPhone);  ws.addCell(labelQian);  int i = 3;  for (Activiting actR : list) {  ws.addCell(new Label(0, i, actR.getId().getASId()));  ws.addCell(new Label(1, i, actR.getStudent().getSName()));  ws.addCell(new Label(2, i, actR.getStudent().getSCollege()));  ws.addCell(new Label(3, i, actR.getStudent().getSClass()));  ws.addCell(new Label(4, i++, actR.getStudent().getSPhone()));  }  wwb.write();  wwb.close();  } catch (IOException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (RowsExceededException e) {  // TODO Auto-generated catch block  e.printStackTrace();  } catch (WriteException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  s = JSONTools.createJsonObject("data", "OK");  return "success";  } |