



从1900-1-0 0:00到现在的天数，JAVA的Date.getTime()函数获得的时间都会比EXCEL里大，其原因如下：1.EXCEL的时间参考是1900-1-0 0:00 为1,而java的时间参考是以1900-1-1 0:00 为0,因此java的时间转换出的数值比excel的少1,所以这里要加1. 2.由于EXCEL误以为1900年是闰年(真正的判定法则是：能被4或400整除,但不能被100整除的年份为闰年)，所以在这个问题上EXCEL又把天数算多了一天，因此，用JAVA算出来的时间要转化成EXCEL的话要再加多一天，因此是加2 经过换算后，时间数值显示正确了，但是时间格式还不行。如果直接从数据库读取时间写入Excel，数值就不用加2了，但格式就不行了。后来使用下面的代码创建时间格式就可以了。

```
HSSFCellStyle cellStyle = workbook.createCellStyle(). short
df=workbook.createDataFormat().getFormat("dd/mm/yy").
cellStyle.setDataFormat(df). cell2.setCellValue(wt.getTimeLog()).
cell2.setCellStyle(cellStyle). 对于书写不规范的日期，日期格式
的单元格有些会识别成数字单元格，如：2009-3-6可以正确识
别为日期，但3-6就识别为数字了。日期读取出来为数字，
在Excel表内日期是以数字表表示的，因此要想正确得到日期
就要，就要做特殊处理。首先判断读取出的单元格类型是否
为数字类型，如果是数字类型在判断使用为日期格式数据，
若是，就将得到的数字值转换为日期。 case
HSSFCell.CELL_TYPE_NUMERIC: if
(HSSFDateUtil.isCellDateFormatted(cell)) { double d =
cell.getNumericCellValue(). Date date =
HSSFDateUtil.getJavaDate(d). } 读取数据更保险的方法为：先
判断类型，更具不同的类型，采用不同的读取方法。
```

```
if(Cell2.getCellType()==HSSFCell.CELL_TYPE_NUMERIC) {
if(HSSFDateUtil.isCellDateFormatted(Cell2)) { double
d=Cell2.getNumericCellValue().
watchData.setTimeLog(HSSFDateUtil.getJavaDate(d)). } }else if
(Cell2.getCellType()==HSSFCell.CELL_TYPE_STRING) {
SimpleDateFormat sFormat=new
SimpleDateFormat("dd-MM-yyyy").
watchData.setTimeLog(sFormat.parse(Cell2.getRichStringCellValue
().getString())). } else{
watchData.setTimeLog(Cell2.getDateCellValue()). } 如果用的是Jxl
, 可以将cell.getCellFormat 强制转换成 XFRecord。然后判
断XFRecord.formatIndex 如果等于 58就是DateCell。 if
(cell.getType() == CellType.NUMBER) { NumberCell nc =
(NumberCell) cell. XFRecord xfr = (XFRecord) nc.getCellFormat().
final int INDEX_OF_DATE = 58 . if (xfr.formatIndex ==
INDEX_OF_DATE) { Date date =
HSSFDateUtil.getJavaDate(nc.getValue()). content =
dateFormat.format(date). } } 编辑特别推荐: #0000ff>Java存
储Excel文件 #0000ff>利用Java控制图片的大小 #0000ff>Java实现
右键弹出菜单例子 100Test 下载频道开通, 各类考试题目直接
下载。详细请访问 www.100test.com
```