



链滴

java 读写有大量数据的 excel 的方法

作者: [oldcaptain](#)

原文链接: <https://ld246.com/article/1375361600401>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)


```

this.listReportMap = listReportMap;<br>
this.flag = flag;<br>
OPCPackage pkg = OPCPackage.open(filename);<br>
XSSFReader r = new XSSFReader(pkg);<br>
SharedStringsTable sst = r.getSharedStringsTable();</p>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">
XMLReader parser = fetchSheetParser(sst);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> // rld2 found by p
rocessing the Workbook
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> // 根据 rld# 或 rS
heet# 查找sheet
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> InputStream shee
2 = r.getSheet("&quot;rld&quot;"+sheetId);
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> sheetIndex++;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> InputSource shee
Source = new InputSource(sheet2);
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> parser.parse(shee
Source);
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> sheet2.close();
</span> </span> </code> </pre>
<p></p>
<p>/**</p>
<ul>
<li>
<p>遍历 excel 文件<br>
*/<br>
public void process(String filename) throws Exception {<br>
OPCPackage pkg = OPCPackage.open(filename);<br>
XSSFReader r = new XSSFReader(pkg);<br>
SharedStringsTable sst = r.getSharedStringsTable();</p>
<p>XMLReader parser = fetchSheetParser(sst);</p>
<p>Iterator<InputStream> sheets = r.getSheetsData();<br>
while (sheets.hasNext()) {<br>
curRow = 0;<br>
sheetIndex++;<br>
InputStream sheet = sheets.next();<br>
InputSource sheetSource = new InputSource(sheet);<br>
parser.parse(sheetSource);<br>
sheet.close();<br>
}<br>
}</p>
</li>
</ul>
<p>public XMLReader fetchSheetParser(SharedStringsTable sst)<br>
throws SAXException {<br>
XMLReader parser = XMLReaderFactory
.createXMLReader("org.apache.xerces.parsers.SAXParser");<br>
this.sst = sst;<br>
parser.setContentHandler(this);<br>
return parser;<br>
}</p>
<p>public void startElement(String uri, String localName, String name,<br>
Attributes attributes) throws SAXException {<br>

```

```

// c = &gt; 单元格<br>
if (name.equals("c")) {<br>
// 如果下一个元素是 SST 的索引，则将 nextIsString 标记为 true<br>
String cellType = attributes.getValue("t");<br>
String rowStr = attributes.getValue("r");<br>
curCol = this.getRowIndex(rowStr);<br>
if (cellType != null && cellType.equals("s")) {<br>
nextIsString = true;<br>
} else {<br>
nextIsString = false;<br>
}<br>
}<br>
// 置空<br>
lastContents = "";<br>
}</p>
<p>public void endElement(String uri, String localName, String name)<br>
throws SAXException {<br>
// 根据 SST 的索引值的到单元格的真正要存储的字符串<br>
// 这时 characters()方法可能会被调用多次<br>
if (nextIsString) {<br>
try {<br>
int idx = Integer.parseInt(lastContents);<br>
lastContents = new XSSFRichTextString(sst.getEntryAt(idx))<br>
.toString();<br>
} catch (Exception e) {</p>
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"
cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> // v = &gt;
单元格的值，如果单元格是字符串则v标签的值为该字符串在SST中的索引
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> // 将单元格内容
入rowlist中，在这之前先去掉字符串前后的空白符
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> if (name.equals(
amp;"v"&quot;)) {
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> String value =
astContents.trim();
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> value = value.
quals("&quot;&quot;")?&quot; &quot;;value;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> int cols = curC
l-preCol;
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> if (cols&gt;
1){
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> for (int i = 0;i
&lt; cols-1;i++){
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> rowlist.ad
(preCol,&quot;&quot;);
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> preCol = curCol

</span> </span> <span class="highlight-line"> <span class="highlight-cl"> rowlist.add(cur
ol-1, value);
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }else {

```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> //如果标签名
为 row , 这说明已到行尾, 调用 optRows() 方法
</span></span><span class="highlight-line"><span class="highlight-cl"> if (name.equals
&amp;"row&amp;quot;)) {
</span></span><span class="highlight-line"><span class="highlight-cl"> int tmpCols
rowlist.size();
</span></span><span class="highlight-line"><span class="highlight-cl"> if(curRow&
mp;gt;this.titleRow &amp;&amp;&amp; tmpCols&amp;lt;this.rowsize){
</span></span><span class="highlight-line"><span class="highlight-cl"> for (int i =
0;i &amp;lt; this.rowsize-tmpCols;i++){
</span></span><span class="highlight-line"><span class="highlight-cl"> rowlist.
dd(rowlist.size(), &amp;"&amp;quot;);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> try { optRo
s(sheetIndex,curRow,rowlist,this.flag,this.listReportMap); } catch (SQLException e) {
</span></span><span class="highlight-line"><span class="highlight-cl"> e.printSta
kTrace();
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> if(curRow==
his.titleRow){
</span></span><span class="highlight-line"><span class="highlight-cl"> this.rowsi
e = rowlist.size();
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> rowlist.clear()

</span></span><span class="highlight-line"><span class="highlight-cl"> curRow++;
</span></span><span class="highlight-line"><span class="highlight-cl"> curCol = 0;
</span></span><span class="highlight-line"><span class="highlight-cl"> preCol = 0;
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>
<p></p>
<p>public void characters(char[] ch, int start, int length)<br>
throws SAXException {<br>
//得到单元格内容的值<br>
lastContents += new String(ch, start, length);<br>
}</p>
<p>//得到列索引, 每一列 c 元素的 r 属性构成为字母加数字的形式, 字母组合为列索引, 数字组合
行索引, <br>
//如 AB45,表示为第 (A-A+1) *26+ (B-A+1) *26 列, 45 行<br>
public int getRowIndex(String rowStr){<br>
rowStr = rowStr.replaceAll("[^A-Z]", "");<br>
byte[] rowAbc = rowStr.getBytes();<br>
int len = rowAbc.length;<br>
float num = 0;<br>
for (int i=0;i<len;i++){<br>
num += (rowAbc[i]-'A'+1)*Math.pow(26,len-i-1 );<br>
}<br>
return (int) num;<br>
}</p>
<p>public int getTitleRow() {<br>
return titleRow;<br>
}</p>

```

```
public void setTitleRow(int titleRow) {  
this.titleRow = titleRow;  
}  
}
```

使用方法

这个方法是在读2007版本的时候，可以自己写一个类然后继承这个抽象类。重写optRows。

在146行可以看到，这个抽象类，会调用子类你重写的方法，所以在子类中的方法可以直接拿到该数据。这种读的方式速度很快，5w行数据也就6,7秒就可以搞定

2、写大量数据解决方案

可以将写出的数据批量写出，1w行数据一个sheet，或者分批写不同的excel。我测试的数据来看。分批写不同的excel速度会比较快。

注：以上的读取代码来自于网络