

使用 jxl 对 excel 追加数据

作者: [catty](#)

原文链接: <https://ld246.com/article/1498654218427>

来源网站: [链滴](#)

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

```

public static void main(String[] args) throws Exception {

    //获取excel文件
    String filePath="C:\\Users\\Administrator\\Desktop\\test.xls";
    File file = new File(filePath);

    //创建不可写入的数据对象
    Workbook workbook = Workbook.getWorkbook(file);

    //创建可写入的数据对象
    WritableWorkbook writableWorkbook = Workbook.createWorkbook(file, workbook);

    //获取指定sheet
    WritableSheet writableSheet= writableWorkbook.getSheet(0);

    //添加数据并写入文件
    Label lable0 = new Label(0, 1, "1");
    Label lable1 = new Label(1, 1, "1");
    Label lable2 = new Label(2, 1, "1");
    writableSheet.addCell(lable0);
    writableSheet.addCell(lable1);
    writableSheet.addCell(lable2);
    writableWorkbook.write();

    //关流
    writableWorkbook.close();
    workbook.close();
}

```

整个代码从上到下也很好理解，先获取excel文件对象，然后创建一个不可写入的Workbook对象，意这里是get的，为什么要创建这个对象呢？继续看下一步，调用Workbook的createWorkbook方法，传入两个参数，分别是文件对象和不可写入的Workbook，这个Workbook对象的用处就是获取之excel中的数据，因为只是需要里面的数据，所以它是不可写入的也就很好理解了，能读取就可以。下来，就是拿着之前的数据与新建的数据，一起写入到excel文件中。那么到底是在原来文件的基础直接追加呢，还是拿到新旧数据再一起覆盖之前的数据呢？当然是后者，这也是创建不可写入的Workbook对象的原因，不然的话根据file获取到文件，直接在后面追加内容就行了。

为了验证，可把代码做如下修改：

```

public static void main(String[] args) throws Exception {

    //获取excel文件
    String filePath="C:\\Users\\Administrator\\Desktop\\test.xls";
    File file = new File(filePath);

    //获取另一个excel文件
    String filePath2="C:\\Users\\Administrator\\Desktop\\test2.xls";
    File file2 = new File(filePath2);

    //创建不可写入的数据对象
    Workbook workbook = Workbook.getWorkbook(file);

```

```
//把从text中读取到的内容替换为test2中的内容
workbook = Workbook.getWorkbook(file2);

//创建可写入的数据对象
WritableWorkbook writableWorkbook = Workbook.createWorkbook(file, workbook);

//获取指定sheet
WritableSheet writableSheet= writableWorkbook.getSheet(0);

//添加数据并写入文件
Label lable0 = new Label(0, 1, "1");
Label lable1 = new Label(1, 1, "1");
Label lable2 = new Label(2, 1, "1");
writableSheet.addCell(lable0);
writableSheet.addCell(lable1);
writableSheet.addCell(lable2);
writableWorkbook.write();

//关流
writableWorkbook.close();
workbook.close();
}
```

test与test2分别写入不同的内容，然后执行代码，会发现test中的内容变为test2的内容加上代码中的内容，而不是test的内容加上代码中的新内容。因此，可以确定并不是直接追加，而是覆盖。