链滴

JXLS 入门指南

作者：MingGH

原文链接：https://ld246.com/article/1620875735373

来源网站：链滴

推荐阅读：

jxls2.3-简明教程

JXLS 2.4.0系列教程（三）——嵌套循环是怎么做到的

以下为官网的翻译版本

1. 入门指南

让我们假设我们有一个employee对象的Java集合，我们想将其输出到Excel中。employee类可能看来像这样

```
public class Employee {
    private String name;
    private Date birthDate;
    private BigDecimal payment;
    private BigDecimal bonus;
    // ... constructors
    // ... getters/setters
}
```

为了使用Jxls将这个对象集合输出到Excel中，我们需要做以下工作

- 在你的项目中添加所需的Jxls库
- 使用特殊标记创建一个Excel模板
- 使用Jxls API来处理准备好的模板，并将employee数据填入其中

让我们详细看看这些步骤中的每一步。

2. 将Jxls库添加到项目中

向项目添加Jxls库的最简单方法是使用Maven，并在项目构建配置文件中指定所需库。

Jxls jars在中央Maven仓库中可用。

我们需要向核心Jxls模块添加以下依赖关系

```
<dependency>
    <groupId>org.jxls</groupId>
    <artifactId>jxls</artifactId>
    <version>2.10.0</version>
</dependency>
```

或者你可以从Sourceforge网站下载Jxls发行版，并使用该发行版中的jars。

除了对核心Jxls模块的依赖，我们还需要增加对Jxls转换引擎的依赖，它将执行所有底层的Java到Exce的操作。

正如Transformers section所解释的（见main concepts））。Jxls的核心模块不依赖于任何特定的Ja a-Excel库，它只通过预定义的接口与Excel一起工作。目前，Jxls在基于著名的Apache POI和Java Ex el API库的独立模块中提供了这个接口的两个实现。

要使用基于Apache POI API的转化器实现，请添加以下依赖关系

```
<dependency>
   <groupId>org.jxls</groupId>
   <artifactId>jxls-poi</artifactId>
   <version>2.10.0</version>
</dependency>
```

要使用基于Java Excel API的转化器实现，请添加以下依赖项

```
<dependency>
   <groupId>org.jxls</groupId>
   <artifactId>jxls-jexcel</artifactId>
   <version>${jxlsJexcelVersion}</version>
</dependency>
```

3. 创建Excel模板

模板是一个excel文件，它使用一个特殊的标记来指定Jxls应该如何输出数据。

Jxls提供了一些内置的标记处理器，可以用来解析excel模板并提取控制命令。

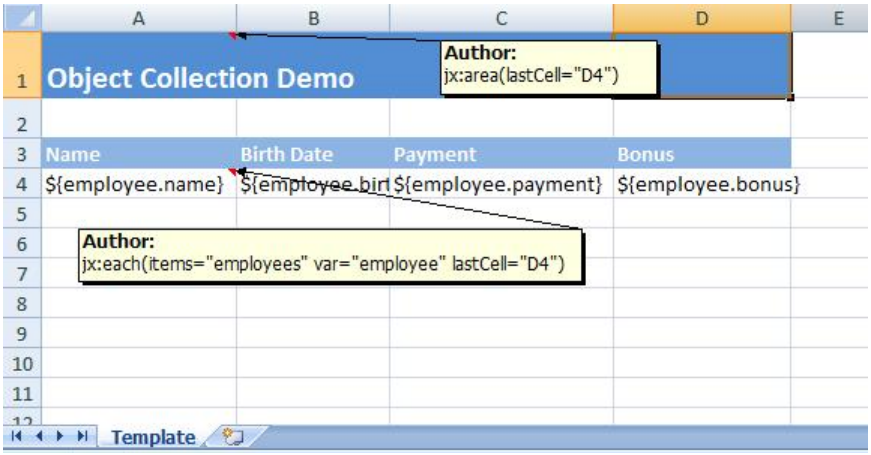
如果需要，还可以创建一个自定义的标记处理器。因此，可以为excel模板定义自己的标记，并以适的方式解析，以创建Jxls命令结构。

让我们来看看内置的Jxls标记处理器。

默认情况下，Jxls支持Apache JEXL作为一种表达式语言，可以在excel模板中使用，以引用java对象属性和方法。该对象必须在Jxls上下文中的某个键下可用。为了在单元格中输出employee的name，们可以在单元格${employee.name}中放入以下文字。基本上我们只是用${and}来包围Jexl表达式。们假设在上下文中，employee键下有一个Employee对象。

属性符号是可配置的，所以你可以决定使用例如[[employee.name]]作为一个属性符号。关于如何做更多细节，请参阅表达式语言。

输出employee对象列表的例子的最终模板可以在这里下载，看起来是这样的



在第4行的模板单元中，我们使用上述的JEXL表达式来引用employee对象的属性。

A1单元格包含一个excel注释，其内容如下jx:area(lastCell="D4")。它定义了我们模板的根区域为A1:4。

A4单元格的注释定义了Jxls Each-Command(遍历命令)，注释文本如下 jx:each(items="employees" var="employees" lastCell="D4")。Each-Command将遍历Jxls上下文中Employees键下的对象集，并将每个单独的集合项放入Employees键下的上下文（由var属性定义）。Each-Command的主体域是A4:D4（由lastCell属性定义），它将与上下文中的每个新Employee对象一起被克隆和处理。

这个例子假设使用XlsCommentAreaBuilder类来构建模板中的Jxls区域。通过使用这个类，你可以在Excel单元格注释中定义Jxls命令。如果你喜欢在Java代码中定义命令，那么模板将是相同的，只是你必删除单元格中的注释。

4. 使用Jxls API来处理模板

这里你可以看到如何使用Jxls API来处理excel模板。

```
...
    logger.info("Running Object Collection demo");
    List<Employee> employees = generateSampleEmployeeData();
    try(InputStream is = ObjectCollectionDemo.class.getResourceAsStream("object_collection_t
mplate.xls")) {
        try (OutputStream os = new FileOutputStream("target/object_collection_output.xls")) {
            Context context = new Context();
            context.putVar("employees", employees);
            JxlsHelper.getInstance().processTemplate(is, os, context);
        }
    }
...
```

在这个例子中，我们从classpath资源object_collection_template.xls加载模板。而目标excel文件将写入target/object_collection_output.xls。

所有的主要处理都在一行中进行

```
JxlsHelper.getInstance().processTemplate(is, os, context);
```

默认情况下，JxlsHelper假设你想用数据覆盖模板表。

但是你也可以选择通过使用以下方法在另一个工作表中生成数据

```
JxlsHelper.getInstance().processTemplateAtCell(is, os, context, "Result!A1");
```

这里的区域将被处理在结果表的A1单元格。

最后的report 可以在这里下载，看起来是这样的