# Excel解析组件开发文档

## 一、如何使用？

1、添加maven依赖

<dependency>  
 <groupId>com.citms</groupId>  
 <artifactId>project-builder-component-excelparser</artifactId>  
</dependency>

2、程序中使用（推荐使用sax解析）

组件支持了dom和sax两种方式解析excel2003及excel2007文档，推荐使用sax解析，比较省内存。

##### Dom解析

使用dom解析excel 文档

private IExcelParser<User> parser;  
  
@Test  
public void testDomXlsx() {  
  
 parser = new ExcelDomParser<>();  
  
 IParserParam parserParam = DefaultParserParam  
 .*builder*()  
 .excelInputStream(Thread.*currentThread*() //要解析的文件流  
 .getContextClassLoader()  
 .getResourceAsStream("test01.xlsx"))  
 .columnSize(4)//解析的列数  
 .sheetNum(IParserParam.*FIRST\_SHEET*) // 解析的工作薄索引  
 .targetClass(User.class) // 需解析出的对象  
 .header(User.*getHeader*())// 校验标题，不传表示不校验标题  
 .firstRow(0) // 解析开始行  
 .build();  
  
 List<Result<User>> user = parser.parse(parserParam);  
 System.*out*.println(user);  
}

##### Sax解析

使用sax解析excel文档

* 使用方法同dom解析完全相同，除了创建的IExcelParser对象不同，dom解析对象为ExcelDomParser，sax解析对象为ExcelSaxParser。

@Test  
public void testSheet02Xlsx() {  
 parser = new ExcelSaxParser<>();  
  
 IParserParam parserParam = DefaultParserParam.*builder*()  
 .excelInputStream(Thread.*currentThread*()  
 .getContextClassLoader().getResourceAsStream("test02.xlsx"))  
 .columnSize(4)  
 .sheetNum(IParserParam.*FIRST\_SHEET* + 1)  
 .targetClass(User.class) //被解析对象  
 .header(User.*getHeader*())  
 .firstRow(0)  
 .build();  
  
 List<Result<User>> obj = parser.parse(parserParam);  
 obj.forEach(o -> {  
 System.*out*.println(o.getRowNo());  
 System.*out*.println(o.getData());  
 });  
}

##### 说明

##### 组件提供了ExcelField注解来对excel的解析方式进行描述，@ExcelField的

字段具体含义如下：

Index: 对应excel的列号（从0开始）。

Type: 对应excel单元格值的类型，用来做数据格式化，目前支持日期/普通字符串。

1. 被解析对象的属性只能是String类型，获取数据后需自行处理，一般用来进行数据校验，返回对应字段的错误提示。