JEECG 智能开发平台

JEasyPoi 技术指南

2017/03/28

www.jeecg.org

Jeecg 社区

目 录

1. 前言

此手册主要为 JEasyPoi 的使用说明，JEasyPoi 主要是为了简化 Poi 的 API 操作,降低 Excel 导入导出已经 Word 导出的入门门槛,通过一系列的注解,来代替原有的 Poi,使其更加简便,美观和高效.JEasyPoi 是在 Jeecg 原有的 Excel 功能基础上发展而来的,去除了原有的一些弊端(如命名过长,convert 这个不好注入 Bean 等)加入新的接口已经新的功能为大家提供更好的服务.

2. JEasyPoi注解

注解是整个 JEasyPoi 的基础,只有理解了注解各个字段的含义才能最大的发挥 JEasyPoi 的作用,当前一共 6 个注解,核心注解 3 个,大家可以先理解下各个字段的作用,然后再使用 util 进行 Excel 或者 Word 的操作

2.1. Excel字段属性

标示在 field 上面,表达需要导出的字段所代表的意思,样式,二次处理等含义.是主要的注解,基本上使用这个注解就可以完成导出,导入等

下面介绍几个主要的字段:

字段	作用,示例	默认值
name	导入导出字段名称比如: name = "学生姓名"	无
width	导出字段宽度(可以每个设置), width = 30	10
height	导出高度(一个设置全局生效), height = 20	10
replace	替换值,比如: replace = {"男_1","女_2"}	[]
type	导出字段类型导出类型 1 是文本 2 是图片,3 是函数默认是文本	1
imageType	图片类型, 导出类型 1 从 file 读取 2 是从数据库中读取	1
savePath	导入路径,如果是图片可以填写,默认是 upload/className/ IconEntity 这个类对应的就是 upload/Icon/	upload
orderNum	排序	0
format	时间格式化	空

dictTable	字典表名	空
dicCode	表字段（code） / 数据字典 code	空
dicText	表字段（Text）	空
suffix	文字后缀,如% 90 变成 90%	空
isWrap	是否换行 即支持\n	true
isStatistics	是否自动统计数据,如果是统计,true 的话在最后追加一行统计,把所有数据都和 这个处理会吞没异常,请注意这一点	false
databaseFormat	导出时间设置,如果字段是 Date 类型则不需要设置 数据库如果是 string 类型,这个需要设置这个数据库格式	yyyyMMddHHmmss
exportFormat	导出的时间格式,以这个是否为空来判断是否需要格式化日期	空
importFormat	导入的时间格式,以这个是否为空来判断是否需要格式化日期	空
mergeVertical	纵向合并内容相同的单元格	false
needMerge	是否需要纵向合并单元格(用于含有 list 中,单个的单元格,合并 list 创建的多个 row)	false

2.2. ExcelCollection集合类

集合类处理注解,代表着一对多

字段	作用,示例	默认值
name	导入导出字段名称比如: name = "学生姓名"	无
orderNum	排序	0
type	导入时创建 List 的实现类	ArrayList

2.3. ExcelEntity实体类

实体类处理注解,代表着一对一或者多对一(主要是穿透作用)

字段	作用,示例	默认值
name	导入导出字段名称比如: name = "学生姓名"	无

2.4. ExcelIgnore忽略属性

忽略类注解,作用就是忽略这个对象主要作用就是防止无限循环.(相信大家 Json 序列号中已经体验过了)

2.5. ExcelTarget导出目标

导出对象,表示当前导出的对象,表示导出的 ID,为字段选择做依据

2.6. ExcelVerify导入校验

Excel 导入的是数据验证注解,这个是一个新增注解,主要就是完成导入数据的基础校验,校验失败会把错误信息,填入到 cell 中去

字段	作用,示例	默认值
interHandler	是不是使用接口处理	false
notNull	非空	false
isMobile	手机号	false
isTel	座机号	false
isEmail	email	false
minLength	最小长度	-1
maxLength	最大长度	-1
regex	正则表达式	空
regexTip	正则错误提示信息	数据不符合规范

3. Excel的导出

`ExcelExportUtil`

导出是利用反射依据实体对象的注解,来完成大家希望导出的数据.整个 Excel 导出具有 4 个函数分别针对了 4 中业务缩减为 2 个,就是模板导出和基础导出,大家可以根据自己的业务自己选择

3.1. 单sheet导出

exportExcel(ExportParams entity, Class<?>pojoClass, Collection<?>dataSet)

单 sheet 导出,完成基础的数据导出,可以完成大部分简单的数据导出,

entity : 导出数据的表头样式等

```
/**
 * 表格名称
 */
private short titleHeight = 20;
/**
 * 第二行名称
 */
private String secondTitle;
/**
 * 表格名称
 */
private short secondTitleHeight = 8;
/**
 * sheetName
 */
private String sheetName;
/**
 * 过滤的属性
 */
private String[] exclusions;
/**
 * 表头颜色
 */
private short color = HSSFColor.WHITE.index;

/**
 * 属性说明行的颜色 例如:HSSFColor.SKY_BLUE.index 默认
 */
private short headerColor = HSSFColor.SKY_BLUE.index;
```

pojoClass： 导出对象

Dataset： 导出集合

对之前的导出进行了部分改良,导出数据限制为 60000,操作数据,自动创建新的 sheet,继续完成导出,同时添加了部分校验,防止导出错误,同时把错误信息抛出,供大家自行处理.

3.2. 多sheet导出

exportExcel(List<Map<String, Object>> list)

多 sheet 导出,大家可以在一个 Excel 导出多个 sheet 数据,完成不同业务的集成.

3.3. 纯模板

exportExcel(TemplateExportParamsparams, Map<String, Object> map)

单纯的 Excel 模板导出,固定的数据报表.程序很难完成的 Excel 样式,大家可以在 Excel 处理完成,然后只是在程序中填充数据,不会改变样式.

3.4. 导出+模板

exportExcel(TemplateExportParamsparams,Class<?>pojoClass, Collection<?>dataSet, Map<String, Object> map)

不仅具有替换功能,同时兼具了注解导出的功能,以来表头数据,大家可以依据导出的模板,来处理

4. Excel导入

ExcelImportUtil,导入比较简单了,没有业务处理

导入同样就 4 个方法,两个维度,本地导入,流导入,返回校验信息,不返回校验信息

导入最主要的就是 ImportParams 的设置

4.1. ImportParams

字段	意义	默认值
titleRows	标题的行数	0
headRows	表头的行数,最大支持 2	1
startRows	字段真正值和列标题之间的距离,就是表头下面是不是有几行空格	0
keyIndex	主键列,一对多的主要的	0
sheetNum	读取的是第几个 sheet	1
needSave	读取完是不是需要保存,	false
saveUrl	保存地址 upload/excelUpload/Test/yyyyMMddHHmss_***** 保存名称上传时间_五位随机数	upload/excelUpload
verifyHanlder	数据校验接口	null

主要需要注意的就是 titleRows 和 headRows,因为 titleRows 等于需要过滤的行数

4.2. 文件导入带校验结果

importExcelVerify(File file, Class<?>pojoClass, ImportParamsparams)

4.3. 文件导入

importExcel(File file, Class<?>pojoClass,ImportParamsparams)

4.4. 流导入

```
importExcelByIs(InputStreaminputstream,
```

```
Class<?>pojoClass, ImportParamsparams)
```

4.5. 流导入带校验结果

importExcelByIsAndVerify(

InputStreaminputstream, Class<?>pojoClass, ImportParamsparams)

5. Word的导出

word 的导出主要是面对公文什么的,模板变化不大,但是样式比较麻烦,这样我们可以先在 word 中制定好模板,然后在替换值导出.导出只是支持 07 版本,同时可以支持在 word 中插入 Excel,支持 Map 和注解 Entity.

5.1. 语法

a.替换

替换值语法使用的是**{{key}}**这样的语法,仅仅替换文本,不会修改样.

b.图片

图片算作一个比较特殊的值,jeecg 解决了 poi 的图片的 bug,可以让大家指定地方,指定大小.key 使用 WordImageEntity 这个类

```
/**
 * 图片输入方式
 */
private String type = URL;
/**
 * 图片宽度
 */
private int width;
// 图片高度
private int height;
// 图片地址
private String url;
// 图片信息
private byte[] data;
```

c.表格

{{ in pList}} in 表示这个表格是个集合数据,plist 有两个类型,一个是 ExcelListEntity 一个是 List, ExcelListEntity 是使用注解来导出数据,以来表头来处理数据如

参数	意义
list	数据源
clazz	类的 Class 对象
headRows	表格行数,1 或者 2

姓名	手机	Email
{{ in pList}}		

List 以来下一行的数据来导出如,支持实体类和 Map

{{ in pList}}		
{{name}}	{{tel}}	{{email}}

导出主要就是利用上面三个语法进行组合数据

5.2. 本地导出

exportWord07(String url, Map<String, Object> map)

url 是本地地址,map 是各个参数的封装

5.3. 文件导出

```
exportWord07(XWPFDocument document,
        Map<String, Object> map)
```

6. Spring View

spring view 可以简化导出的操作,输入相应参数就可以完成 Excel,Word 导出

例如:

```
@RequestMapping(params = "exportXls")
public String exportXls(CourseEntity course,HttpServletRequest request,HttpServletResponse response
        , DataGrid dataGrid,ModelMap map) {

    CriteriaQuery cq = new CriteriaQuery(CourseEntity.class, dataGrid);
    org.jeecgframework.core.extend.hqlsearch.HqlGenerateUtil.installHql(cq, course, request.getParamete
    List<CourseEntity> courses = this.courseService.getListByCriteriaQuery(cq,false);

    map.put(NormalExcelConstants.FILE_NAME,"用户信息");
    map.put(NormalExcelConstants.CLASS,CourseEntity.class);
    map.put(NormalExcelConstants.PARAMS,new ExportParams("课程列表", "导出人:Jeecg",
            "导出信息"));
    map.put(NormalExcelConstants.DATA_LIST,courses);
    return NormalExcelConstants.JEECG_EXCEL_VIEW;

}
/**
```

6.1. JeecgMapExcelView

单 sheet 或者多 sheet 导出使用

参数	值
NormalExcelConstants.FILE_NAME	导出文件名称
NormalExcelConstants.PARAMS	导出参数
NormalExcelConstants.CLASS	实体对象
NormalExcelConstants.DATA_LIST	数据源
NormalExcelConstants.MAP_LIST	多数据源集合

6.2. JeecgSingleExcelView

单 sheet 或者多 sheet 导出使用

参数	值
NormalExcelConstants.FILE_NAME	导出文件名称
NormalExcelConstants.PARAMS	导出参数
NormalExcelConstants.CLASS	实体对象
NormalExcelConstants.DATA_LIST	数据源
NormalExcelConstants.MAP_LIST	多数据源集合

6.3. JeecgTemplateExcelView

Excel 模板导出使用

参数	值
NormalExcelConstants.FILE_NAME	导出文件名称
NormalExcelConstants.PARAMS	导出参数
NormalExcelConstants.CLASS	实体对象
TemplateExcelConstants.LIST_DATA	注解使用导出数据源
TemplateExcelConstants.MAP_DATA	值替换导出数据源

6.4. JeecgTemplateWordView

Word 模板导出

参数	值
NormalExcelConstants.FILE_NAME	导出文件名称
TemplateWordConstants.URL	Word 地址
TemplateExcelConstants.MAP_DATA	值替换导出数据源

7. Spring web集成

使用 spring mvc 需要在 spring-mvc.xml 或者其他的配置文件中进行配置,主要是

默认视图级别设置低点

```
<!-- 默认的视图解析器 在上边的解析错误时使用 (默认使用html)- -->
 <bean id="defaultViewResolver"
  class="org.springframework.web.servlet.view.InternalResourceViewResolver"
  p:order="3">
  <property name="viewClass"
   value="org.springframework.web.servlet.view.JstlView" />
  <property name="contentType" value="text/html" />
  <property name="prefix" value="/webpage/" />
  <property name="suffix" value=".jsp" />
 </bean>
```

Bean 视图设置级别高一些,然后把我们的视图配置上,就完成了

```
<!-- Bean解析器,级别高于默认解析器,寻找bean对象进行二次处理 -->
<bean id="beanNameViewResolver"
      class="org.springframework.web.servlet.view.BeanNameViewResolver" p:order="0">
</bean>
<!-- Excel 处理 根据用户输入进行对象处理 -->
<bean id="jeecgExcelView" class="org.jeecgframework.poi.excel.view.JeecgSingleExcelView" />
<bean id="jeecgTemplateExcelView" class="org.jeecgframework.poi.excel.view.JeecgTemplateExcelView" />
<bean id="jeecgTemplateWordView" class="org.jeecgframework.poi.excel.view.JeecgTemplateWordView" />
<bean id="jeecgMapExcelView" class="org.jeecgframework.poi.excel.view.JeecgMapExcelView" />
```

2.0.8 版本后加上了@Controller 里面只要在

```
<context:component-scan base-package="org.jeecgframework.poi.excel.view">
```

加入就可以了,但是解析器还是要配置的

这样就算集成完成了,下面我们介绍各种 VIEW

8. 表达式支持

- 空格分割
- 三目运算 {{test ? obj:obj2}}
- n: 表示 这个 cell 是数值类型 {{n:}}
- le: 代表长度{{le:()}} 在 if/else 运用{{le:() > 8 ? obj1 : obj2}}
- fd: 格式化时间 {{fd:(obj;yyyy-MM-dd)}}
- fn: 格式化数字 {{fn:(obj;###.00)}}
- fe: 遍历数据,创建 row
- !fe: 遍历数据不创建 row
- $fe: 下移插入,把当前行,下面的行全部下移.size()行,然后插入
- !if: 删除当前列 {{!if:(test)}}
- 单引号表示常量值 " 比如'1' 那么输出的就是 1