**JEECG**

**EasyPoi操作手册**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√] 草稿  [ ] 正式发布  []正在修改 | 文件标识： |  |
| 当前版本： | 1.0 |
| 作者： |  |
|  |  |

前言：此手册主要为EasyPoi的使用说明，EasyPoi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.EasyPoi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.

# 一.注解

注解是整个EasyPoi的基础,只有理解了注解各个字段的含义才能最大的发挥EasyPoi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

## 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 | 图片保存路径 | upload |
| orderNum | 排序 | 0 |
| format | 时间格式化 | 空 |

## 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 |

## 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 |

## 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)

## 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

## 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 |

# 二.Excel的导出

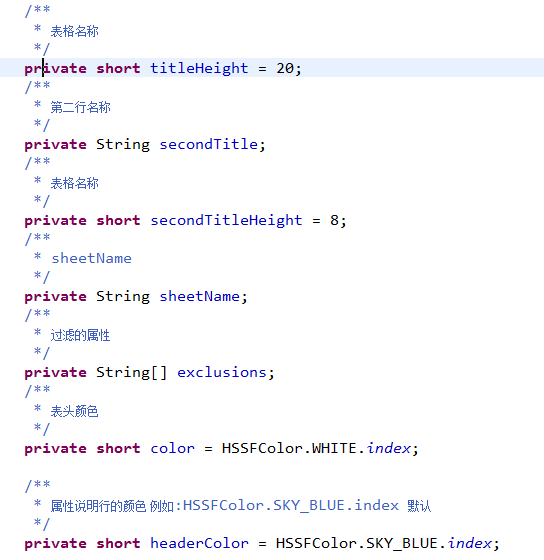
ExcelExportUtil

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

## 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



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.

## 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.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.

## 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

# 三.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

## 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

## 文件导入带校验结果

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

## 2.文件导入

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

## 2.流导入

importExcelByIs(InputStreaminputstream,

Class<?>pojoClass, ImportParamsparams)

## 4.流导入带校验结果

importExcelByIsAndVerify(

InputStreaminputstream, Class<?>pojoClass, ImportParamsparams)

# 四.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.

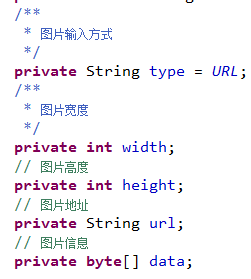
## 语法

#### 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



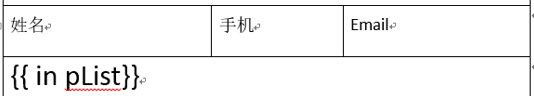
#### c.表格

{{ 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,

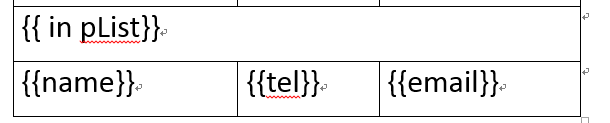
ExcelListEntity

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 |



List以来下一行的数据来导出如,支持实体类和Map



导出主要就是利用上面三个语法进行组合数据

## 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

## 3.文件导出

exportWord07(XWPFDocument document,

Map<String, Object> map)

# 五.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:



## 1.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 |

## 2.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 |

## 3.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 |

# 六.说明

主要是缓存,使用的是guava的缓存管理,缓存大小是50个文件以及一天的缓存时间