Excel导出分为两种模式

1. 直接导出

主要实现的功能点如下:

1. 当数据条数大于6w条时,将导出的多个excel打包成zip包返回
2. 当数据条数小于6w条时,返回单个excel
3. 可以根据需要设置导出的excel版本为2003或2007

用例代码如下:

@RequestMapping(**"/get/excel"**)  
**public void** getExcel(HttpServletRequest request, HttpServletResponse response) **throws** Exception {  
  
 *// 准备数据* List<Student> list = **new** ArrayList<>();  
 **for** (**int** i = 0; i < 50000; i++) {  
 list.add(**new** Student(111,**"王五"**+i,**"女"**));  
 }  
  
 *//excel表头* String[] columnNames = { **"ID"**, **"姓名"**, **"性别"** };  
  
 *//字段名称* String[] fieldStr = { **"id"**, **"name"**, **"sex"** };  
  
 *//文件名* String fileName=**"导出数据"**;  
  
 ExcelUtil util = **new** ExcelUtil();  
 util.exportExcel(columnNames, fieldStr,list, fileName, ExcelUtil.***EXCEL\_FILE\_2003***,request,response);  
}

2.按模板进行导出

主要实现的功能点如下:

1) 当数据条数大于6w条时,将导出的多个excel打包成zip包返回

2) 当数据条数小于6w条时,返回单个excel

3) 只支持简单的模板导出,如若为复杂模板,需要自己写代码进行实现

注:当导入的模板为什么格式时,导出的文件就为什么格式,如若需要导出指定的excel版本,需要提供多个版本的excel,在前台进行判断然后传到工具类中即可.

用例代码如下:

@Value(**"${excel.templeDir}"**)  
**private** String **templeDir**;

@RequestMapping(**"/get/getExcelTemp"**)  
**public void** getExcelTemp(HttpServletRequest request, HttpServletResponse response) **throws** Exception {  
  
 List<Student> list = **new** ArrayList<>();  
 **for** (**int** i = 0; i < 80000; i++) {  
 list.add(**new** Student(111, **"王五"** + i, **"女"**));  
 }  
 *//文件名* String fileName=**"导出数据"**;  
  
 ExcelTempUtil util = **new** ExcelTempUtil();  
 util.createExcel(**templeDir**+**"exportExc.xlsx"**,list,fileName,request,response);  
  
}

注:在这里需要注入模板文件的存放路径

@Value(**"${excel.templeDir}"**)  
**private** String **templeDir**;

在application.yml中有声明此变量

模板格式用例:

