### 该模块需要的内容

调用**ExcelServiceImpl**中**autoFillData**方法。此处仅为一个流程，但是是最重要的一部分。更具体的所有流程见下方测试类。

需要传入**fromPath**->来源文件地址，形如”webRoot/usernormal/”注意:此处为文件夹地址！不是Excel文件具体所在的地址。

需要传入**toPath**->文件存储目录，一般指定autoCreateExcel文件夹

需要传入**fileName**->单为文件名

**fromPath**来源->私人/公共模板数据库中存储的url。（可能存储的是文件具体地址，需要分离字符串）

**toPath**来源->一般指定，如果需要换个地址，在调用的时候直接指定即可。

**fileName**来源->从私人/公共模板数据库中存储的url。（分离字符段后得到文件名）

**tea\_id**来源->理论上是action类调用ExcelService方法，tea\_id从request中获取。

### 参数确定后，理论上可以得到

1. 根据模板填充后的**excel文件**，以及**文件地址**。->用于下载。
2. 根据填充后的excel文件转换的**PNG文件**，以及**文件地址**。->调用该地址。用于前端页面展示。
3. 根据填充后的excel文件转换的**HTML代码**，此处仅仅为**代码**，不是文件。->在PNG方法失效的时候，返回HTML代码，用于界面显示。
4. 前三者存储在一个数组中，并将其数组放在**一个list**中，用于批量调用等操作。

### 测试类-com.utils.test-POITest

public void testExcelService() throws Exception {

// 模拟批量生成，后续由action提供teacherList

// 使用批量导出的时候，传递一个包含了需要批量导出的教师信息list即可

ArrayList<String> teaList = new ArrayList<String>();

teaList.add("101");

teaList.add("102");

// 新建resultList，存储处理后的数据

ArrayList<String[]> resultList = new ArrayList<String[]>();

// 遍历teaList，用于批量处理

for (String tea : teaList) {

// 填充Excel文件，并且返回excel文件地址

String toFile = **es.autoFillData**(tea,

"WebRoot/methods/publicMethods/",

"WebRoot/methods/autoCreateExcel/", "分表显示信息-back-pub.xlsx");

System.out.println(toFile);

// 生成html代码

String htmlExcel = new **POIReadExcelToHtml()**.main(toFile);// 生成 html

// 处理png文件，返回png文件地址 // 以防万一生成图片报错

String pngFile = new **excel2Picture()**.createPng(toFile);

**// 第一项为生成的excel文件地址，第二项为html代码，第三项为png文件地址**

**String[] result = { toFile, htmlExcel, pngFile };**

// 加入到结果集合里

resultList.add(result);

}

// 哪个部分需要使用就直接调用resultList中对应的String[]，再获取相应的内容即可。

}

### ExcelServiceImpl-com.service.impl

方法：

**自动填充数据**-**public String autoFillData(String tea\_id, String fromPath, String toPath,**

**String fileName)**

tea\_id->根据tea\_id查询该教师的所有数据。fromPath->来源文件地址，形如”webRoot/usernormal/”**注意:此处为文件夹地址！**不是Excel文件具体所在的地址。toPath同理。fileName为Excel文件的文件名。

**存储数据库字段**-**private ArrayList<ExcelCell> saveDBValue(ArrayList<ExcelCell> excelCellArrayList, Object object)**

参数类型可以不用理睬

**文件名相关服务**-**private String nameService(String fromPath, String toPath, String fileName,String tea\_id)**

fromPath与toPath同上。

### WriteExcelBean-com.utils

方法：

**拷贝excel文件到目标地址**-**public void copyExcelFile(String fromPath, String toPath, String fileName,String toFile)**

fromPath,toPath,fileName同上，toFile=toPath+fileName

**填充数据到新文件中**-**public boolean fill(ArrayList<ExcelCell> exList, String toFile,String direction, int step, String way)**

exList->ExcelCell实体对象的集合，toFile->目标地址，这个包括了excel文件的详细位置。direction->填充方向，默认为right。step->移动的位置，默认为1.（意思就是向右一格填充数据。）way->正向解析以及反向解析。

公共模板支持正向解析和反向解析。私人模板只支持正向解析。

**反向解析，写入中文字段-private boolean backWrite(ArrayList<ExcelCell> exList, Workbook wb)**

例如：原始excel中tea\_id,将其修改为教师编号

**写入ExcelCell对象中读取到的数据库数据-private boolean forwardWrite(ArrayList<ExcelCell> exList, Workbook wb,String direction, int step)**

此方法为核心方法，写入的重中之重

**判断特殊字段-public boolean isSpecialWord(String word)**

例如tea\_img等图片字段。

**写入图片到相关单元格中-public void wirteImg(ArrayList<ExcelCell> exList, Workbook wb)**

**清理公共模板中无用的单元格-private void clearNullExcel(ArrayList<ExcelCell> exList, Workbook wb)**

计划用于多表显示信息时，一次性显示更多的信息。

### excel2Picture-com.utils.showExcel

**根据excel文件创建png文件-public String createPng(String toFile)**

**返回png文件地址;**

**此处toFile为文件详细地址！生成png的位置和生成填充后文件的地址相同**

### POIReadExcelToHtml-com.utils.showExcel

**转换Excel为Html代码**-**public void main(String path)**

当无法生成png文件的时候，留下的后备方案。此方法为网络中的方法。没有能力进行更改

**path为excel文件具体地址，返回html代码**

### AnalyticExcelBean-com.utils

简介:读取原始文档内的字段，并将其转换成数据库中对应的标准字段。(形如”教师编号”转换为”tea\_id”)便于查询查询数据。

例如:list.add(new String[] { "tea\_id", "职工号", "id", "ID", "编号", "教师编号","教师ID", "教师id" });

Return : ArrayList<String[]> list; 此处包含了所有的数据库字段的解析。

注意事项:在补充同一个字段的同义词时，请尽量保证每一个字段的意思不重复。(举例:奖励年份和评估年份两者需要分清)

问题:如果同义词过多,可能导致匹配到的数据库字段出现偏差，“私人模板”写入时，相关数据将会出现错误。

公共模板通过“反向解析”实现了几乎不会上述情况，但对于编写公共模板的难度将会提高。

### ReadExcelBean-com.utils

简介:1. 读取原始Excel文件的数据。2. 根据AnalyticExcelBean的数组，将原始字段与数据库标准字段匹配。

Return: ExcelCell 实体对象的组成的ArrayList。

### EntityToMapBean-com.utils

简介:实体转为Map的类。目的是获取由tea\_id读取到的教师数据。将其“数据库字段”以及“相应的值”，与原始字段匹配。

例如：Teacher-com.domain 这个类拥有tea\_id,tea\_name,tea\_birthday等属性，将其属性名与值组成Map

形如: “tea\_id”:”101”

Return:相应实体类的属性名和对应的值组成的Map

在ExcelService中调用得到的Map 可以实现如下操作

tea\_id=”101” tea\_name=”sunshine”

读取的字段为”教师编号=null”,”教师姓名=null”

转换后-”教师编号=101”,”教师姓名=sunshine”