

# 接了个变态需求：生成 Excel + PDF 导出，用 Java 怎么实现？

程序员追风 2022-10-05 22:00 发表于湖南

收录于合集

#Java

124个

上方蓝色“[程序员追风](#)”，选择“设为星标”  
回复“**资料**”获取整理好的**面试资料**

原文：[cnblogs.com/Tom-shushu/p/14279357.html](https://cnblogs.com/Tom-shushu/p/14279357.html)

## 一、序言

Excel、PDF的导出、导入是我们工作中经常遇到的一个问题，刚好今天公司业务遇到了这个问题，顺便记个笔记以防下次遇到相同的问题而束手无策。

公司有这么两个需求：

- 需求一、给了一个表单，让把查出来的数据组装到表单中并且提供以PDF格式的下载功能。
- 需求二、将数据查出来以Excel表格的形式下载下来。

## 二、Java实现PDF的生成和数据动态插入、导出功能

### 1、第一步：PDF制作模板

因为PDF常用的软件不支持编辑，我们就先使用WPS以Word的形式进行编辑制作出与客户需求一样的样式，然后直接另存为 .pdf 的形式如下图所示：

#### a.Word里面制作模板



b.更改名字为 .pdf形式



c.这时需要用到一个叫：Adobe Acrobat DC的软件，具体操作如下：

用Adobe Acrobat DC打开我们刚才改过名字的PDF文件， 点击右下角的“更多工具”按钮



到下面这个页面再点击“准备表单”按钮



d.接下来就需要详细的配置你的数据源了



数据源即：你代码中实体类中对应的数据（注意字段一定要一一对应），配置完毕就可以保存进行下面的代码编写工作了。

## 2、代码的编写

假定我们实体类什么的都已经编写完成、数据通过前端传入获取、模板位置在E盘根目录下名字为：车辆维修审批单.pdf

导入jar包：

```
1 <!-- PDF导出-->
2 <!-- https://mvnrepository.com/artifact/com.itextpdf/itextpdf -->
3 <dependency>
4     <groupId>com.itextpdf</groupId>
5     <artifactId>itextpdf</artifactId>
6     <version>5.5.13</version>
7 </dependency>
```

实现生成PDF、数据插入、导出

```
1 @RegisterToSMP(serviceDisplay = "预览页面PDF下载")
2 @RequestMapping(value = "/DM/gwclwxsq/qygl/exportPDF$m=query.service",method
```

```
3 public String exportPdf(@RequestBody GwclwxsqBean gwclwxsqBean , HttpServletRequest
4     // 1.指定解析器
5     System.setProperty("javax.xml.parsers.DocumentBuilderFactory",
6         "com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryIr
7     String filename="车辆维修审批单.pdf";
8     String path="e:/";
9     response.setContentType("application/pdf");
10    response.setHeader("Content-Disposition", "attachment;fileName="
11        + URLEncoder.encode(filename, "UTF-8"));
12    OutputStream os = null;
13    PdfStamper ps = null;
14    PdfReader reader = null;
15    try {
16        os = response.getOutputStream();
17        // 2 读入pdf表单
18        reader = new PdfReader(path+ "/" +filename);
19        // 3 根据表单生成一个新的pdf
20        ps = new PdfStamper(reader, os);
21        // 4 获取pdf表单
22        AcroFields form = ps.getAcroFields();
23        // 5给表单添加中文字体 这里采用系统字体。不设置的话，中文可能无法显示
24        BaseFont bf = BaseFont.createFont("C:/WINDOWS/Fonts/SIMSUN.TTC,1",
25            BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
26        form.addSubstitutionFont(bf);
27        // 6查询数据=====
28        Map<String, String> data = new HashMap<String, String>();
29        data.put("commitTime", gwclwxsqBean.getCommitTime());
30        data.put("driver", gwclwxsqBean.getDriver());
31        data.put("carId", gwclwxsqBean.getCarId());
32        data.put("carType", gwclwxsqBean.getCarType());
33        data.put("repairAddress", gwclwxsqBean.getRepairAddress());
34        data.put("repairCost", gwclwxsqBean.getRepairCost());
35        data.put("project", gwclwxsqBean.getProject());
36        data.put("fwbzzxfzrYj", gwclwxsqBean.getFwbzzxfzrYj());
37        data.put("fgldspYj", gwclwxsqBean.getFgldspYj());
38        data.put("remarks", gwclwxsqBean.getRemarks());
39        // 7遍历data 给pdf表单表格赋值
40        for (String key : data.keySet()) {
41            form.setField(key,data.get(key).toString());
42        }
```

```
43         ps.setFormFlattening(true);
44         log.info("*****PDF导出成功*****");
45     } catch (Exception e) {         log.error("*****PDF导出失败*****");
46         e.printStackTrace();
47     } finally {
48         try {
49             ps.close();
50             reader.close();
51             os.close();
52         } catch (Exception e) {
53             e.printStackTrace();
54         }
55     }
56     return null;
57 }
```

### 3、测试





### 三、Java实现Excel生成和数据插入、导出

这个比较简单，直接上代码（假定你的实体类、查询什么的都已经写好）注意：实体类一个是你自己的数据实体类还有一个是你导出时表格中对应的实体类。

我们以一个真实的公司业务来举个例子（一个统计疫情登记人员信息的Excel导出功能）

a.表头对应实体类ExportYqfkdj.java

```
1  import lombok.Data;
2
3  /**
4   * description:
5   * @author: zhouhong
6   * @version: V1.0.0
7   * @date: 2021年1月14日 下午3:05:54
8   */
9  @Data
10 public class ExportYqfkdj {
11     /**
12      * 序号
13      */
```

```
14     private Integer xuhao;
15     /**
16      * 姓名
17      */
18     private String xingming;
19     /**
20      * 证件号码
21      */
22     private String zjhm;
23     /**
24      * 联系电话
25      */
26     private String lxdh;
27     /**
28      * 申请人工作单位
29      */
30     private String sqrGzdw;
31     /**
32      * 是否接触过疑似病例
33      */
34     private String sfjcgysbl;
35     /**
36      * 当前是否与居家隔离人员同住
37      */
38     private String sfyjjglrytz;
39     /**
40      * 当前状态
41      */
42     private String dqzt;
43     /**
44      * 当前健康状况
45      */
46     private String dqjkzt;
47
48
49     /**
50      * 当前体温
51      */
52     private String dqtw;
53     /**
```



```
54     * 当前所在地址
55     */
56     private String dqszzdz;
57     /**
58     * 当前居住地址
59     */
60     private String dqjzdz;
61     /**
62     * 提交时间
63     * */
64     private String tjsj;
65 }
```

## b.Service层

```
1  /**
2   * 导出
3   * @param yqfkjdjBean
4   * @author zhouhong
5   * @return
6   * @throws Exception
7   */
8  @Transactional(rollbackFor = { Exception.class })
9  public DataResult exporYqfkjdj(YqfkjdjBean yqfkjdjBean) throws Exception {
10     DataResult result = new DataResult();
11     List<ExportYqfkjdj> list = new ArrayList<ExportYqfkjdj>();
12     try {
13         /* 查询导出信息 */
14         result = getYqfkMhCXQuery(yqfkjdjBean);
15         SimpleDateFormat df = new SimpleDateFormat("yyyyMMddhhmmssSSS");
16         for (int i = 0; i < result.getTotalcount(); i++) {
17             ExportYqfkjdj dmKhfwdcDtjlZxDto = new ExportYqfkjdj();
18             dmKhfwdcDtjlZxDto = ObjectUtil.parsePojo(result.getResults().get(i));
19             dmKhfwdcDtjlZxDto.setXuhao(i + 1);
20             list.add(dmKhfwdcDtjlZxDto);
21         }
22         String filepath = "D:/疫情防控信息" + df.format(new Date()) + ".xlsx";
23         if (System.getProperty(YqfkjdjUtils.Wjdz.NAME).toLowerCase().startsWith("d:")) {
```

```
24         || System.getProperty(YqfkdjUtils.Wjdz.NAME).toLowerCase().st
25         filepath = "/home/Tomcat/temp/" + df.format(new Date()) + ".xlsx"
26     }
27     EasyExcel.write(filepath, ExportYqfkdj.class).head(head()).sheet().do
28     result.setResults(list);
29     result.setSuccess(true);
30     result.setMsg(filepath);
31 } catch (Exception e) {
32     result.setSuccess(false);
33     result.setMsg(YqfkdjUtils.Cytx.DCSB);
34     e.printStackTrace();
35     throw e;
36 }
37     return result;
38 }
39 /**
40  * 疫情防控信息导出表头
41  * @author zhouhong
42  * @return List<List<String>>
43  */
44 private List<List<String>> head() {
45     List<List<String>> list = new ArrayList<List<String>>();
46     List<String> head0 = new ArrayList<String>();
47     head0.add("序号");
48     List<String> head1 = new ArrayList<String>();
49     head1.add("姓名");
50     List<String> head2 = new ArrayList<String>();
51     head2.add("证件号码");
52     List<String> head3 = new ArrayList<String>();
53     head3.add("联系电话");
54     List<String> head4 = new ArrayList<String>();
55     head4.add("工作所在单位");
56     List<String> head5 = new ArrayList<String>();
57     head5.add("是否接触疑似病例");
58     List<String> head6 = new ArrayList<String>();
59     head6.add("是否与隔离人员同住");
60     List<String> head7 = new ArrayList<String>();
61     head7.add("当前状态");
62     List<String> head8 = new ArrayList<String>();
63     head8.add("当前健康状态");
```

```
64     List<String> head9 = new ArrayList<String>();
65     head9.add("体温(°C)");
66     List<String> head10 = new ArrayList<String>();
67     head10.add("当前所在地址");
68     List<String> head11 = new ArrayList<String>();
69     head11.add("当前居住地址");
70     List<String> head12 = new ArrayList<String>();
71     head12.add("提交时间");
72     list.add(head0);
73     list.add(head1);
74     list.add(head2);
75     list.add(head3);
76     list.add(head4);
77     list.add(head5);
78     list.add(head6);
79     list.add(head7);
80     list.add(head8);
81     list.add(head9);
82     list.add(head10);
83     list.add(head11);
84     list.add(head12);
85     return list;
86 }
```

## c.Controller层

```
1  @RegisterToSMP(serviceDisplay = "疫情防控查询导出")
2  @RequestMapping(value = "/DM/yqfkdj/gr/yqfkjdjdc$m=export.service", method = {RequestMethod.GET, RequestMethod.POST})
3  public void exportKhfxxx(@RequestBody YqfkdjBean yqfkdjBean, HttpServletResponse response) {
4      DataResult result = new DataResult();
5      try {
6          SimpleDateFormat df = new SimpleDateFormat("yyyyMMddhhmmssSSS");
7          result = yqfkdjService.exporyqfkdj(yqfkdjBean);
8          String filepath = result.getMsg().replace("\\", "");
9          File file = new File(filepath);
10         String filename = "疫情防控信息" + df.format(new Date()) + ".xlsx";
11         InputStream fis = new BufferedInputStream(new FileInputStream(filepath));
```

```
12     byte[] buffer = new byte[fis.available()];
13     fis.read(buffer);
14     fis.close();
15     resp.reset();
16     resp.setHeader("Content-Disposition",
17         "attachment;filename=" + new String(filename.replaceAll(" ",
18     resp.setHeader("Content-Length", "" + file.length());
19     OutputStream os = new BufferedOutputStream(resp.getOutputStream());
20     resp.setContentType("application/vnd.openxmlformats-officedocument.sp
21     // 输出文件
22     os.write(buffer);
23     os.flush();
24     os.close();
25 } catch (Exception e) {
26     e.printStackTrace();
27     log.info(YqfkdjUtils.Cytx.DCSB);
28     throw e;
29 }
30 }
```

## 测试



已经全部完成PDF和Excel的生成、插入、导出功能。



程序员追风

专注于分享Java各类学习笔记、面试题以及IT类资讯。

公众号

○ 你在看吗

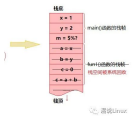
收录于合集 #Java 124

下一篇 · 优秀的代码都是如何分层的？

喜欢此内容的人还喜欢

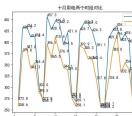
你真的知道C语言函数调用的内幕吗？

混说Linux



家庭用电情况分析（python+pandas）

电力数据新应用



docker命令图文详解

你呀不牛

