springboot+thymeleaf 生成 PDF

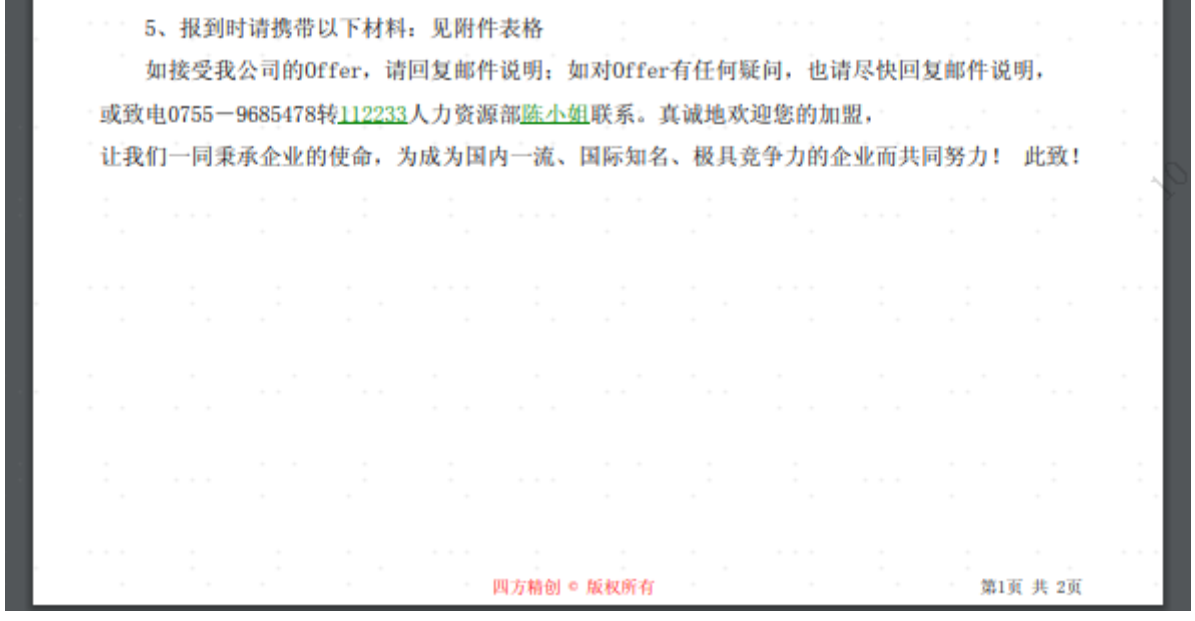
```
一、依赖
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <version>2.3.0.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
    <version>2.6.4</version>
</dependency>
<dependency>
    <groupId>org.xhtmlrenderer</groupId>
    <artifactId>flying-saucer-pdf</artifactId>
    <version>9.0.7</version>
</dependency>

二、配置
[application.yml]
spring:
  thymeleaf:
    prefix: classpath:/templates/
    check-template-location: true
    suffix: .html
    encoding: UTF-8
    mode: HTML
    cache: false
    servlet:
      content-type: text/html

三、工具类
@Slf4j
public class PdfUtil {

    /**
     * pdf下载
     * @param templateEngine 配置
     * @param templateName   模板名称
     * @param data           模板参数集
     * @param fileName       下载文件名称(带文件扩展名后缀)
     */
    public static void download(TemplateEngine templateEngine,
                                String templateName,
                                Map<String, Object> data,
                                HttpServletResponse response,
                                String fileName) {
        // 设置编码、文件ContentType类型、文件头、下载文件名
        response.setCharacterEncoding("utf-8");
        response.setContentType("application/pdf");
        try {
            response.setHeader("Content-Disposition", "attachment;fileName=" +
                    new String(fileName.getBytes("gb2312"), "ISO8859-1"));
        } catch (UnsupportedEncodingException e) {

            log.error(e.getMessage(), e);
```

```
        }
        try (ServletOutputStream out = response.getOutputStream()) {
            createPDF(templateEngine, templateName, out, data);
            out.flush();
        } catch (Exception e) {
            log.error(e.getMessage(), e);
        }
    }

    /**
     * pdf预览
     * @param templateEngine  配置
     * @param templateName   模板名称
     * @param data           模板参数集
     * @param response       HttpServletResponse
     */
    public static void preview(TemplateEngine templateEngine,
                               String templateName,
                               Map<String, Object> data,
                               HttpServletResponse response) {
        try (ServletOutputStream out = response.getOutputStream()) {
            createPDF(templateEngine, templateName, out, data);
            out.flush();
        } catch (Exception e) {
            log.error(e.getMessage(), e);
        }
    }

    /**
     * 生成pdf文档
     * @param templateName  模板名称
     * @param data          模板参数
     */
    private static void createPDF(TemplateEngine templateEngine,
                                  String templateName, OutputStream out,
                                  Map<String, Object> data) throws Exception {
        if (CollectionUtils.isEmpty(data)) {
            log.warn("警告: 模板参数为空!");
            return;
        }
        ITextRenderer renderer = new ITextRenderer();
        //设置字符集(宋体), 此处必须与模板中的<body style="font-family: SimSun">一致, 区分大小写, 不能写成汉字"宋体"
        ITextFontResolver fontResolver = renderer.getFontResolver();
        //设置系统字体, 否则不支持中文
        fontResolver.addFont("static/fonts/simsun.ttf", BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
        fontResolver.addFont("static/fonts/simsun.ttc", BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
        Document pageDoc = generateDoc(templateEngine, templateName, data);
        renderer.setDocument(pageDoc, null);
        //展现和输出pdf
        renderer.layout();
        renderer.createPDF(out, false);
        //写下一个pdf页面
        //renderer.writeNextDocument();
        //完成pdf写入
        renderer.finishPDF();
    }

    /**
     * 模板+参数 -> html字符串 -> Document
     */
```

```
    private static Document generateDoc(TemplateEngine templateEngine,
                                        String templateName,
                                        Map<String, Object> data) {
        // 声明一个上下文对象，里面放入要存到模板里面的数据
        final Context context = new Context();
        context.setVariables(data);
        StringWriter stringWriter = new StringWriter();
        try (BufferedWriter writer = new BufferedWriter(stringWriter)) {
            templateEngine.process(templateName, context, writer);
            writer.flush();
            DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
            return builder.parse(new ByteArrayInputStream(stringWriter.toString().getBytes()));
        } catch (Exception e) {
            log.error(e.getMessage(), e);
            return null;
        }
    }
}

四、测试
@Controller
@RequestMapping("/pdf/")
public class PdfController {

    @Autowired
    private TemplateEngine templateEngine;

    //模板参数
    final Map<String, Object> dataList = new HashMap<String, Object>(){{
        put("title", "录取通知书");
        put("companyName", "四方精创咨询股份有限公司");
        put("sendDate", new Date());
        put("employName", "李四");
        put("bgName", "开发");
        put("jobName", "JAVA工程师");
        put("salary", 2050);
        put("cycle", 14);
        put("subsidy", 400);
        put("contractYear", 3);
        put("contractMonth", 3);
        put("registerDate", "2022年6月1日");
        put("address", "深圳市南山区高新科技园101号A栋1602");
        put("extension", "112233");
        put("hrName", "陈小姐");
    }};

    @GetMapping("index")
    public ModelAndView toIndex(ModelAndView mv) {
        mv.addAllObjects(new HashMap<String, Object>(){{
            put("title", "主题");
            put("employName", "录用者姓名");
            put("companyName", "xx有限公司");
            put("sendDate", new Date());
        }});
        mv.setViewName("index");
        return mv;
    }

    /**
     * pdf预览
```

```
     */
    @GetMapping(value = "/preview")
    public void preview(HttpServletResponse response) {
        PdfUtil.preview(templateEngine, "index", dataList, response);
    }

    /**
     * pdf渲染 + 下载
     */
    @GetMapping(value = "/download")
    public void download(HttpServletResponse response) {
        PdfUtil.download(templateEngine, "index", dataList, response, "录取通知书（测试版）.pdf");
    }
}
```

预览效果图



录取通知书

Dear李四： 非常高兴地通知您，经过我司的面试和讨论，我们一致认为您是我公司开发部JAVA工程师 岗位的合适人选。根据公司的薪资福利政策，我们将给您提供以下薪酬福利待遇：

一、现金薪酬：

您的现金收入为税前RMB 2050元/月，公司执行14个月月薪的薪酬制度为标准。

二、福利：

根据公司福利政策，公司为您提供以下福利项目：

1、社会保险：为您提供国家规定的社会保险；

2、公积金：为您提供国家规定的公积金；

3、补助：公司为您提供400/月的补贴。

除以上福利项目外，公司还会为您提供节日费、员工婚育礼金等，并资助员工的各项活动。四方精创和谐的企业文化氛围一定会让您感受到集体的温暖。（以上员工福利待遇信息为保密资料，请勿向公司内外的其他人员透露。）

三、其他

1、工作地点：

2、劳动合同期限：3年，其中试用期3个月。

3、报到时间：2022年6月1日

4、报到地点：深圳市南山区高新科技园101号A栋1602

5、报到时请携带以下材料：见附件表格

如接受我公司的Offer，请回复邮件说明；如对Offer有任何疑问，也请尽快回复邮件说明，或致电0755－9685478转112233人力资源部陈小姐联系。真诚地欢迎您的加盟，
让我们一同秉承企业的使命，为成为国内一流、国际知名、极具竞争力的企业而共同努力！ 此致！

序号	准备材料	备注
1	身份证*	核原件，收复印件1份
2	社保卡及住房公积金联名卡	各收复印件1份
3	体检报告原件*	收原件，公立三甲医院半年内有效
4	原公司离职证明原件*	收原件
5	最高学历证书原件*	核原件，收复印件1份
6	一寸白底免冠证件照*	纸质照片2张，电子照片1张

四方精创资讯股份有限公司

2022-06-24

下载

今天

录取通知书（测试版）.pdf

http://localhost:8088/pdf/download

在文件夹中显示