

根据pdf模板 生成pdf文件

- 网络参考

<https://jiaohongwei.github.io/2017/02/22/Spring%E4%BD%BF%E7%94%A8PDF%E6%A8%A1%E6%9D%BF%E7%94%9F%E6%88%90PDF%E6%96%87%E4%BB%B6/>

- 依赖

```
<dependency>
  <groupId>com.itextpdf</groupId>
  <artifactId>itextpdf</artifactId>
  <version>5.4.3</version>
</dependency>
<dependency>
  <groupId>com.itextpdf</groupId>
  <artifactId>itext-asian</artifactId>
  <version>5.2.0</version>
</dependency>
```

- 代码

```
//get请求控制器
@RequestMapping(value = "/downpdf.htm",method = RequestMethod.GET)
public void down(@RequestParam(value = "tenantid", required = true) String tenantid,
                 @RequestParam(value = "keyid", required = true)String keyid,HttpServletRequest req, HttpServletResponse respo

    try {
        SurveyTaskQry qry=new SurveyTaskQry();
        qry.setTenantid(tenantid);
        qry.setTaskid(keyid);
        surveyTaskHandler.down(qry,req, response);
    } catch (Exception e){
        logger.error("获取勘察任务详情失败! 原因: ",e);
    }
}

public class DownPdfUtil {
    public static HttpServletResponse downloadFile(HttpServletRequest req, HttpServletResponse response, Map<String,String> map,

    //设置http头文件
    response.setContentType("APPLICATION/OCTET-STREAM");
    response.setHeader("Content-Disposition", "attachment; filename=\"survey.pdf\"");

    //生成pdf文件
    addCommonPdf(map,templatepath,destpath,urlMap);

    //生成文件流传给客户端
    FileInputStream fs = null;
    int b = 0;
    try {
        fs = new FileInputStream(new File(destpath));
        PrintWriter out = response.getWriter();
        while((b=fs.read())!=-1) {
            out.write(b);
        }
        fs.close();
        out.close();
    }catch(Exception e) {
    }
    return response;
}

/**
 * 生成pdf文件方法
```

```

* @param map
* @param templatepath
* @param destpath
* @param urlMap
*/
public static void addCommonPdf(Map<String,String> map, String templatepath, String destpath, Map<String,List> urlMap){
    try{
        PdfReader reader = new PdfReader(templatepath);
        ByteArrayOutputStream bos = new ByteArrayOutputStream();
        PdfStamper ps = new PdfStamper(reader, bos);
        BaseFont bf = BaseFont.createFont("STSong-Light", "UniGB-UCS2-H", BaseFont.NOT_EMBEDDED);
        AcroFields fields = ps.getAcroFields();
        fields.addSubstitutionFont(bf);

        //填写数据
        fillData(fields, map);

        //pdf中插入网络图片
        /* List<String> list=urlMap.get("installscheme");
        if(!CollectionUtils.isEmpty(list)){
            URL url=null;
            String fieldName = "installscheme";
            url = new URL(list.get(0).toString());
            int pageNo = fields.getFieldPositions(fieldName).get(0).page;
            Rectangle signRect = fields.getFieldPositions(fieldName).get(0).position;
            float x = signRect.getLeft();
            float y = signRect.getBottom();
            Image image = Image.getInstance(url);
            PdfContentByte under = ps.getOverContent(pageNo);
            image.scaleToFit(signRect.getWidth(), signRect.getHeight());
            image.setAbsolutePosition(x, y);
            under.addImage(image);
        }*/

        //pdf中插入本地图片
        /*
        // 通过域名获取所在页和坐标, 左下角为起点
        int pageNo = form.getFieldPositions(fieldName).get(0).page;
        Rectangle signRect = form.getFieldPositions(fieldName).get(0).position;
        float x = signRect.getLeft();
        float y = signRect.getBottom();
        // 读图片
        Image image = Image.getInstance(imagePath);
        // 获取操作的页面
        PdfContentByte under = stamper.getOverContent(pageNo);
        // 根据域的大小缩放图片
        image.scaleToFit(signRect.getWidth(), signRect.getHeight());
        // 添加图片
        image.setAbsolutePosition(x, y);
        under.addImage(image);
        */

        //生成目标文件destpath到本地
        File file=new File(destpath);
        if(file.exists()){
            file.delete();
        }
        OutputStream outputStream=new FileOutputStream(destpath);
        ps.setFormFlattening(true);
        ps.close();
        bos.writeTo(outputStream);
        reader.close();
        bos.close();
        outputStream.close();
    }catch(Exception e){
    }
}

/**
 * 数据填充
 */
public static void fillData(AcroFields fields, Map<String, String> data) throws IOException, DocumentException {
    for (String key : data.keySet()) {
        String value = data.get(key);
        if(!StringUtils.isEmpty(value)){

```

```
        fields.setField(key, value);
    }
}
}
```

另外一种下载文件方式

```
public void down(HttpServletRequest req, HttpServletResponse response) throws Exception{
    String fullPath = "D:\\doc\\pp1.pdf";
    //从文件完整路径中提取文件名, 并进行编码转换, 防止不能正确显示中文名
    try {
        if(filePath.lastIndexOf("/") > 0) {
            fileName = new String(filePath.substring(filePath.lastIndexOf("/") + 1, filePath.length()).getBytes("GB2312"), "ISO8859_1");
        } else if(filePath.lastIndexOf("\\") > 0) {
            fileName = new String(filePath.substring(filePath.lastIndexOf("\\") + 1, filePath.length()).getBytes("GB2312"), "ISO8859_1");
        } catch (Exception e) {}

        File downloadFile = new File(fullPath);
        ServletContext context = req.getServletContext();
        // get MIME type of the file
        String mimeType = context.getMimeType(fullPath);
        if (mimeType == null) {
            // set to binary type if MIME mapping not found
            mimeType = "application/octet-stream";
        }
        response.setContentType(mimeType);
        response.setContentLength((int) downloadFile.length());

        String headerKey = "Content-Disposition";
        String headerValue = String.format("attachment; filename=\"%s.pdf\"",
            downloadFile.getName());
        response.setHeader(headerKey, headerValue);
        InputStream myStream = new FileInputStream(fullPath);
        IOUtils.copy(myStream, response.getOutputStream());
        response.flushBuffer();
    }
}
```