

SpringBoot + FreeMarker + FlyingSaucer

实现 PDF 在线预览、打印、下载

关键技术点：

1. Freemarker 模板引擎
 - 模板语法
2. FlyingSaucer 根据模板生成 pdf
 - 兼容中文（及中文换行问题）
 - 兼容 CSS（绝对、相对定位）
 - 兼容图片
 - 多页输出

(示例代码没有 dao、service 层，生产环境中自行添加)

源代码 GITHUB 地址: <https://github.com/QuSongtao/demo-pdf>

实现步骤

SpringBoot 项目搭建

略

Maven 依赖配置

```
<!-- freemarker 依赖 -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-freemarker</artifactId>
</dependency>

<!-- web 基础依赖 -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

```
<!-- FlyingSaucer 依赖
https://mvnrepository.com/artifact/org.xhtmlrenderer/flying-saucer-pdf -->
<dependency>
    <groupId>org.xhtmlrenderer</groupId>
    <artifactId>flying-saucer-pdf</artifactId>
    <version>9.1.12</version>
</dependency>
```

PDF 工具类编写

PdfUtils.java，方法上有完整注释，思路是利用模板引擎动态处理模板参数，先生成 html 字符串放在 StringWriter 中，再用 HTML 字符串生成 Document，再利用 FlyingSaucer 的 ITextRenderer 处理 Document，最后输出 pdf。

```
/*
 * 版权所有 ? 成都太阳高科技有限责任公司
 * http://www.suncd.com
 */
package com.suncd.demopdf.Utils;

import com.lowagie.text.pdf.BaseFont;
import freemarker.template.Template;
import freemarker.template.TemplateException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.util.CollectionUtils;
import org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer;
import org.w3c.dom.Document;
import org.xhtmlrenderer.pdf.ITextFontResolver;
import org.xhtmlrenderer.pdf.ITextRenderer;

import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletResponse;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import java.io.*;
import java.util.List;
import java.util.Map;

/**
 * 功能：pdf 处理工具类
 */
```

```

* @author qust
* @version 1.0 2018/2/23 17:21
*/
public class PdfUtils {
    private PdfUtils() {
    }

    private static final Logger LOGGER = LoggerFactory.getLogger(PdfUtils.class);

    /**
     * 按模板和参数生成 html 字符串,再转换为 flying-saucer 识别的 Document
     *
     * @param templateName freemarker 模板名称
     * @param variables     freemarker 模板参数
     * @return Document
     */
    private static Document generateDoc(FreeMarkerConfigurer configurer, String templateName,
    Map<String, Object> variables) {
        Template tp;
        try {
            tp = configurer.getConfiguration().getTemplate(templateName);
        } catch (IOException e) {
            LOGGER.error(e.getMessage(), e);
            return null;
        }

        StringWriter stringWriter = new StringWriter();
        try(BufferedWriter writer = new BufferedWriter(stringWriter)) {
            try {
                tp.process(variables, writer);
                writer.flush();
            } catch (TemplateException e) {
                LOGGER.error("模板不存在或者路径错误", e);
            } catch (IOException e) {
                LOGGER.error("IO 异常", e);
            }
            DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
            return builder.parse(new ByteArrayInputStream(stringWriter.toString().getBytes()));
        } catch (Exception e){
            LOGGER.error(e.getMessage(), e);
            return null;
        }
    }
}

```

```

/**
 * 核心: 根据 freemarker 模板生成 pdf 文档
 *
 * @param configurer    freemarker 配置
 * @param templateName freemarker 模板名称
 * @param out           输出流
 * @param listVars      freemarker 模板参数
 * @throws Exception 模板无法找到、模板语法错误、IO 异常
 */
private static void generateAll(FreeMarkerConfigurer configurer, String templateName, OutputStream
out, List<Map<String, Object>> listVars) throws Exception {
    if (CollectionUtils.isEmpty(listVars)) {
        LOGGER.warn("警告:freemarker 模板参数为空!");
        return;
    }

    ITextRenderer renderer = new ITextRenderer();
    Document doc = generateDoc(configurer, templateName, listVars.get(0));
    renderer.setDocument(doc, null);
    //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();
    fontResolver.addFont("simsun.ttc", BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
    //展现和输出 pdf
    renderer.layout();
    renderer.createPDF(out, false);

    //根据参数集个数循环调用模板,追加到同一个 pdf 文档中
    //(注意:此处从 1 开始,因为第 0 是创建 pdf,从 1 往后则向 pdf 中追加内容)
    for (int i = 1; i < listVars.size(); i++) {
        Document docAppend = generateDoc(configurer, templateName, listVars.get(i));
        renderer.setDocument(docAppend, null);
        renderer.layout();
        renderer.writeNextDocument(); //写下一个 pdf 页面
    }
    renderer.finishPDF(); //完成 pdf 写入
}

/**
 * pdf 下载
 *
 * @param configurer    freemarker 配置
 * @param templateName freemarker 模板名称(带后缀.ftl)
 * @param listVars      模板参数集

```

```

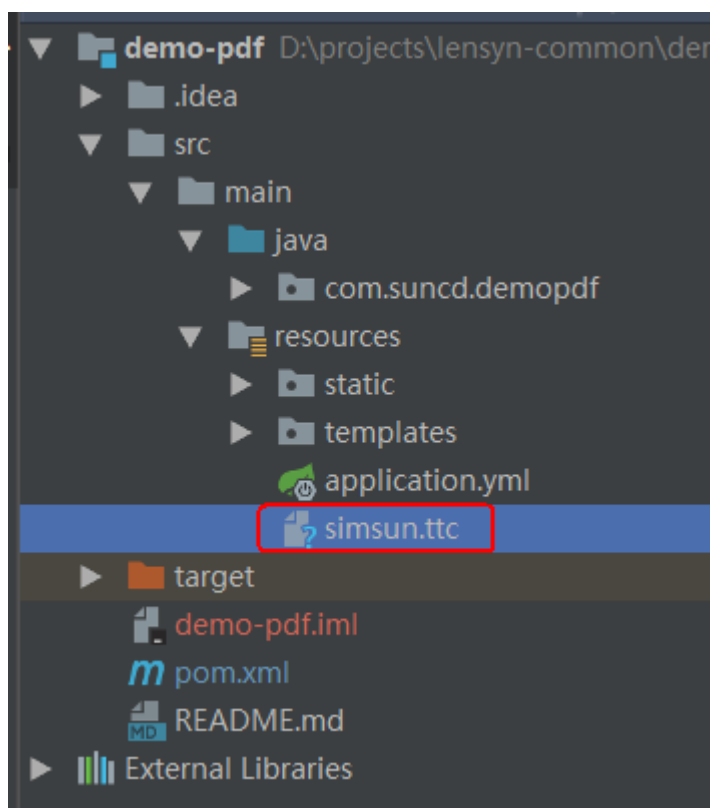
    * @param response      HttpServletResponse
    * @param fileName      下载文件名称(带文件扩展名后缀)
    */
    public static void download(FreeMarkerConfigurer configurer, String templateName, List<Map<String,
Object>> listVars, HttpServletResponse response, String fileName) {
        // 设置编码、文件 ContentType 类型、文件头、下载文件名
        response.setCharacterEncoding("utf-8");
        response.setContentType("multipart/form-data");
        try {
            response.setHeader("Content-Disposition", "attachment;fileName=" +
                new String(fileName.getBytes("gb2312"), "ISO8859-1"));
        } catch (UnsupportedEncodingException e) {
            LOGGER.error(e.getMessage(), e);
        }
        try (ServletOutputStream out = response.getOutputStream()) {
            generateAll(configurer, templateName, out, listVars);
            out.flush();
        } catch (Exception e) {
            LOGGER.error(e.getMessage(), e);
        }
    }

    /**
    * pdf 预览
    *
    * @param configurer    freemarker 配置
    * @param templateName  freemarker 模板名称(带后缀.ftl)
    * @param listVars       模板参数集
    * @param response       HttpServletResponse
    */
    public static void preview(FreeMarkerConfigurer configurer, String templateName, List<Map<String,
Object>> listVars, HttpServletResponse response) {
        try (ServletOutputStream out = response.getOutputStream()) {
            generateAll(configurer, templateName, out, listVars);
            out.flush();
        } catch (Exception e) {
            LOGGER.error(e.getMessage(), e);
        }
    }
}

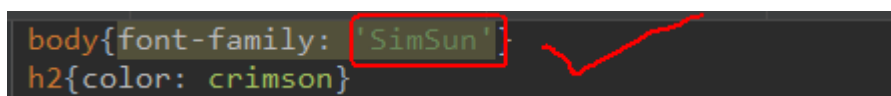
```

中文字符显示坑点：generateAll 方法红色标记部分，

填坑：①需要拷贝宋体字体文件到 resource 目录下（字体位置在“<c:/Windows/Fonts/simsun.ttc>”），方便集成和迁移



②在页面中设置 body 的样式 `<body style="font-family: SimSun">` 必须写成英文，同时大小写敏感，



另外：也有不少文章直接根据操作系统类型取宋体字体文件路径的全路径，如下，显得代码臃肿：

```

26. // step 3 解决中文支持
27. ITextFontResolver fontResolver = renderer.getFontResolver(); 不推荐
28. if("linux".equals(getCurrentOperatingSystem())){
29.     fontResolver.addFont("/usr/share/fonts/chiness/simsun.ttc" BaseFont.IDENTITY_H, BaseFo
nt.EMBEDDED);
30. }else{
31.     fontResolver.addFont("c:/Windows/Fonts/simsun.ttc" BaseFont.IDENTITY_H, BaseFont.NOT_E
MBEDDED);
32. }

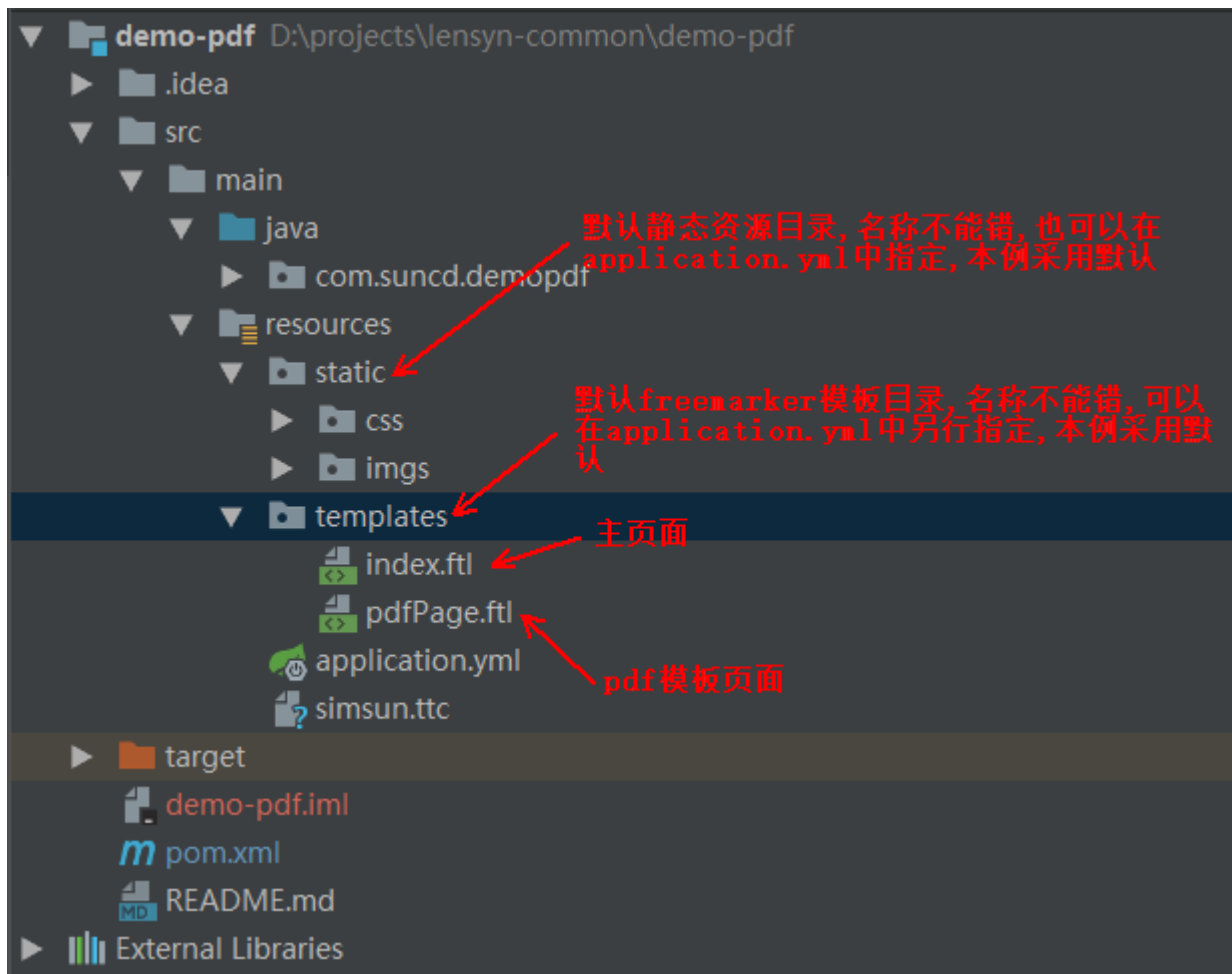
```

有可能linux中没有该字体

注意: generateAll 方法中已经实现了一个模板接收多个参数对象,输出多页到一个 pdf 文件中,读者可根据自己需要改造

FreeMarker 模板编写

跟编写普通 html 页面一样,定义 2 个页面,一个主页面 index.ftl, 一个 pdf 模板页面 pdfPage.ftl
文件结构:



index.ftl,很简单,一个标题,两个按钮,一个预览功能,一个下载功能,同时预接收一个\${title}参数

注:freemarker 的语法和原理,读者自行科普

```
<!DOCTYPE html>
<html>
<head lang="en">
  <title>Demo Page PDF</title>
</head>
<body>
<h2>Demo Page ${title}</h2>
<div><a href="/pdf/preview" target="_blank"> 强大的预览 </a></div>
<div><a href="/pdf/download"> 强大的下载 </a></div>
</body>
</html>
```

pdfPage.ftl

```
<!DOCTYPE html>
<html>
<head lang="en">
  <title>Spring Boot Demo - PDF</title>
  <link href="http://localhost:8999/css/index.css" rel="stylesheet" type="text/css"/>
```

```

<style>
  @page {
    size: 210mm 297mm; /*设置纸张大小:A4(210mm 297mm)、A3(297mm 420mm) 横向则反
过来*/
    margin: 0.25in;
    padding: 1em;
    @bottom-center{
      content:"成都太阳高科技 ? 版权所有";
      font-family: SimSun;
      font-size: 12px;
      color:red;
    };
    @top-center { content: element(header) };
    @bottom-right{
      content:"第" counter(page) "页 共 " counter(pages) "页";
      font-family: SimSun;
      font-size: 12px;
      color:#000;
    };
  }
</style>
</head>
<body style="font-family: 宋体">
<div>1.标题-中文</div>
<h2>${title}</h2>

<div>2.按钮:按钮的边框需要写 css 渲染</div>
<button class="a" style="border: 1px solid #000000"> click me t-p</button>
<div id="divsub"></div>

<div>3.普通 div</div>
<div id="myheader">Alice's Adventures in Wonderland</div>

<div>4.图片 绝对定位到左上角(注意:图片必须用全路径或者 http://开头的路径,否则无法显示)</div>
<div id="signImg"></div>

<div>5.普通 table 表格</div>
<div>
  <table>
    <tr>
      <td>1</td>
      <td>2</td>
      <td>2</td>
      <td>2</td>
    </tr>
  </table>
</div>

```



```

        <td>2</td>
    </tr>
    <tr>
        <td>1</td>
        <td>2</td>
        <td>2</td>
        <td>2</td>
        <td>2</td>
    </tr>
    <tr>
        <td>1</td>
        <td>2</td>
        <td>2</td>
        <td>2</td>
        <td>2</td>
    </tr>
</table>
</div>

<div>6.input 控件,边框需要写 css 渲染 (在模板中一般不用 input,因为不存在输入操作)</div>
<div>
    <label>姓名:</label>
    <input id="input1" aria-label="dasdasd" type="text" value="123 你是"/>
</div>
</body>
</html>

```

坑点(用户经常有页面尺寸需求,比如纸张类型):

1. 页面尺寸(A3,A4)设置和脚标设置

页面尺寸填坑: 在<head>节点中加入 CSS3 页面 page 属性,以毫米为单位设置 size,即最终输出 pdf 每页的大小

A3: 297mm * 420mm (纵向)

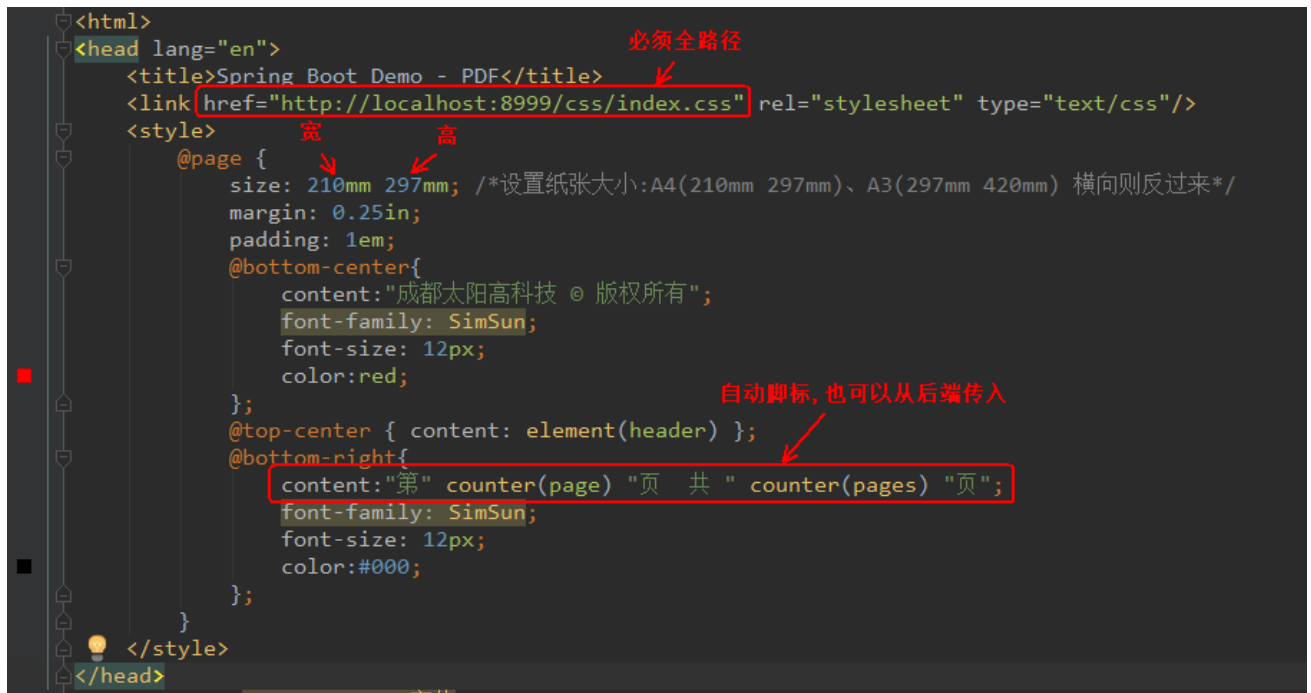
A4: 210mm * 297mm (纵向)

A3: 420mm * 297mm (横向)

A4: 297mm * 210mm (横向)

这些都可以写成\${XXX}占位符形式,通过后端代码传入

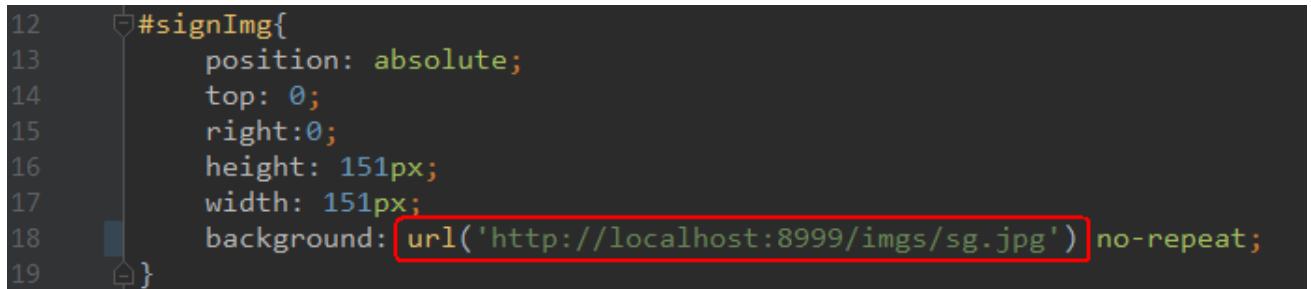
脚标填坑: 见下图



2. CSS 路径和图片路径

填坑 css 路径: 引用 css 文件必须用 http://全路径,如上图,可以把 css 文件单独放到一台服务器上,通过域名或者 ip+端口访问.

填坑图片路径: css 中引用的图片一样要使用 http://全路径,如下图:



Controller 代码编写

写两个 Controller, PublicController.java 和 PdfController.java

PublicController.java 用来访问主页面, PdfController.java 用来接受预览和下载请求

PublicController.java

```
/*
 * 版权所有 ? 成都太阳高科技有限责任公司
 * http://www.suncd.com
 */
package com.suncd.demopdf.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;
```

```

/**
 * 功能：公共
 *
 * @author qust
 * @version 1.0 2018/2/23 11:56
 */
@Controller
public class PublicController {

    @RequestMapping(value = "/")
    public ModelAndView index(ModelAndView modelAndView) {
        modelAndView.setViewName("index");
        modelAndView.addObject("title", "CGX");
        return modelAndView;
    }
}

```

PdfController.java

```

/**
 * 版权所有 ? 成都太阳高科技有限责任公司
 * http://www.suncd.com
 */
package com.suncd.demopdf.controller;

import com.suncd.demopdf.Utils.PdfUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * 功能：pdf 预览、下载
 *
 * @author qust
 * @version 1.0 2018/2/23 9:35
 */

```

```

@Controller
@RequestMapping(value = "/pdf")
public class PdfController {

    @Autowired
    private FreeMarkerConfigurer configurer;

    /**
     * pdf 预览
     *
     * @param request  HttpServletRequest
     * @param response HttpServletResponse
     */
    @RequestMapping(value = "/preview", method = RequestMethod.GET)
    public void preview(HttpServletRequest request, HttpServletResponse response) {
        // 构造 freemarker 模板引擎参数,listVars.size()个数对应 pdf 页数
        List<Map<String,Object>> listVars = new ArrayList<>();
        Map<String,Object> variables = new HashMap<>();
        variables.put("title","测试预览 ASGX!");
        listVars.add(variables);

        PdfUtils.preview(configurer,"pdfPage.ftl",listVars,response);
    }

    /**
     * pdf 下载
     *
     * @param request  HttpServletRequest
     * @param response HttpServletResponse
     */
    @RequestMapping(value = "/download", method = RequestMethod.GET)
    public void download(HttpServletRequest request, HttpServletResponse response) {
        List<Map<String,Object>> listVars = new ArrayList<>();
        Map<String,Object> variables = new HashMap<>();
        variables.put("title","测试下载 ASGX!");
        listVars.add(variables);
        PdfUtils.download(configurer,"pdfPage.ftl",listVars,response,"测试中文.pdf");
    }
}

```

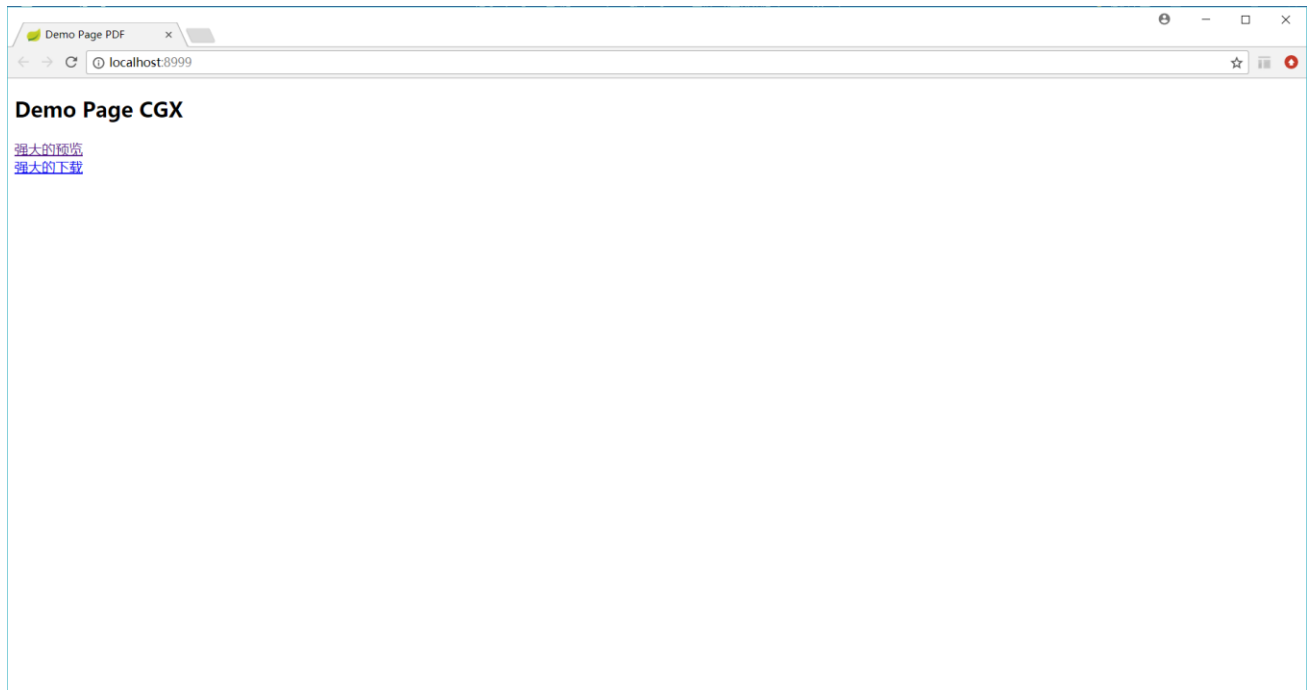
配置 application.yml

```
server:
```

port: 8999

运行演示

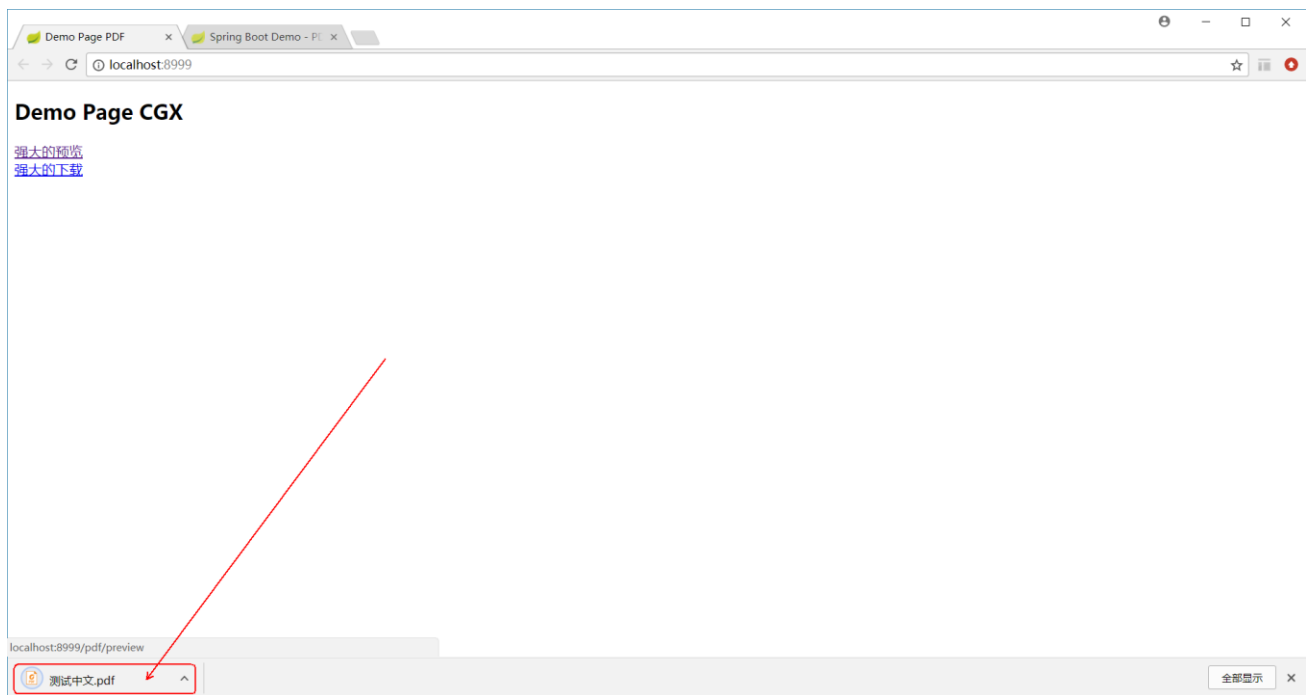
运行项目,访问 <http://localhost:8999/>



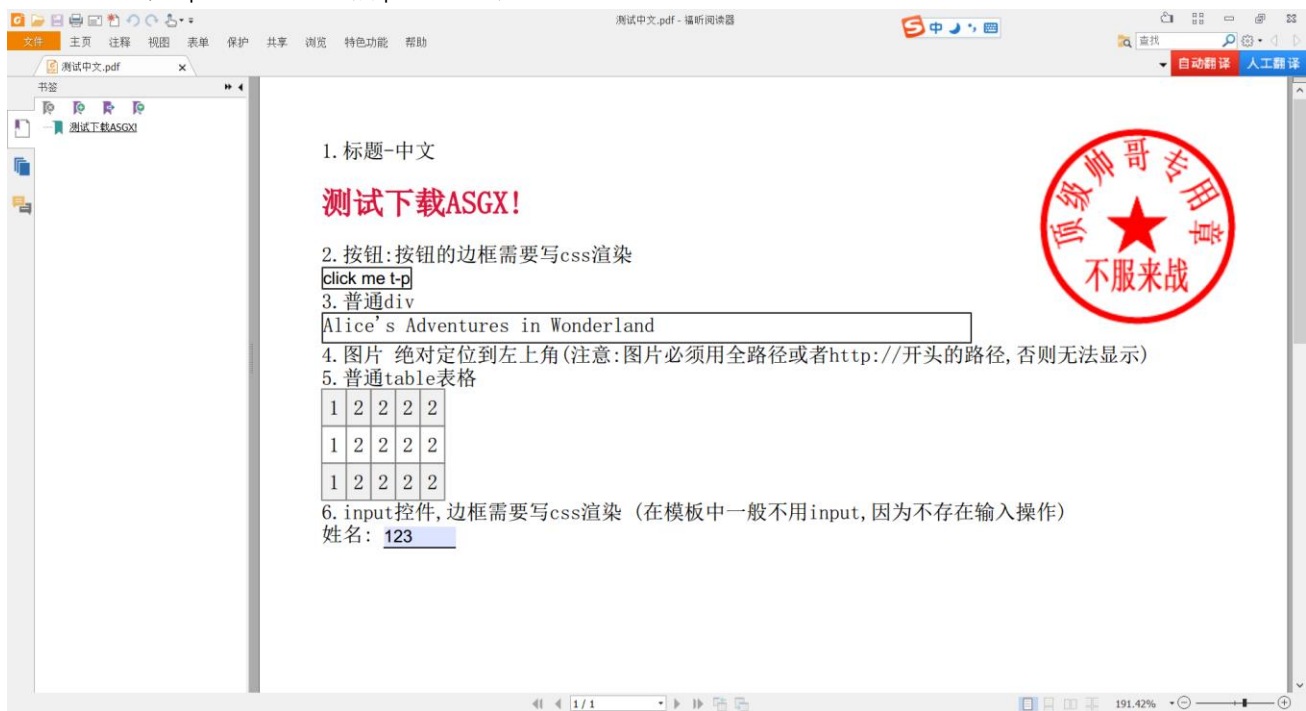
点击预览效果如下(有个小坑,就是 input 控件中的汉字有问题,反正我实际生产中 pdf 模板不用 input 控件)



再点击下载,效果如下:



显示已下载,从 pdf 软件打开该 pdf 文件效果如下：



大功告成!

坑点总结

1. 中文字体
2. Css 路径
3. 图片路径
4. 页面尺寸(纸张大小)

建议

该示例只是为了演示如何利用 freemarker 模板引擎生成 pdf 预览、下载，其中数据都为静态数据，在实际项目中调整数据来源可完美达到预期效果，目前支持比较好的是 Chrome 内核浏览器，为达到更好的浏览器支持，可以用 PDF.js 来完成兼容。

PdfUtils.java 只是对模板操作做了简单封装,可以根据自己的需要进行二次封装, [generateAll 方法中已经实现了一个模板接收多个参数对象,输出多页到一个 pdf 文件中,读者可根据自己需要改造\(比如把多个不同的模板输出到一个 pdf 文件中\)](#)。