

# 1. 模板

一种基于模板和要改变的数据，并用来生成输出文本(HTML网页，电子邮件，配置文件，源代码等)的通用工具。

在模板中，你可以专注于如何展现数据，而在模板之外可以专注于要展示什么数据，体现就是：模板+ 数据模型 = 输出

```
#这是一个html模板  index.ftl
<html>
  hello ${name}
</html>
```

上方我们提供了一个模板index.ftl，.ftl是模板的后缀，模板中\${name}为变量，我们通过Java程序可以为name传入不同的值，从而展现不同的内容

## 1.1 示例

在common模块中引入freemarker的依赖，我们定义一个模板，并使用Java程序将其输出为不同的静态文件

```
<dependency>
  <groupId>org.freemarker</groupId>
  <artifactId>freemarker</artifactId>
  <version>2.3.31</version>
  <scope>compile</scope>
</dependency>
```

```
package com.mszlu.common.freemarker;

import freemarker.template.Configuration;
import freemarker.template.Template;
import freemarker.template.TemplateException;
import freemarker.template.Version;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
```

```

import java.util.HashMap;
import java.util.Map;

public class FreemarkerUtils {

    public static void main(String[] args) throws IOException,
TemplateException {

        Configuration configuration = new Configuration(new
Version("2.3.0"));
        configuration.setDirectoryForTemplateLoading(new
File("D:\\git\\mszlu\\xm\\mszlu-xt-parent\\mszlu-xt-
common\\src\\main\\resources\\"));
        configuration.setDefaultEncoding("utf-8");
        Template template = configuration.getTemplate("index.ftl");
        Map<String, Object> map = new HashMap<>();
        map.put("name", "freemarker");
        FileWriter fileWriter = new FileWriter(new
File("D:\\git\\mszlu\\xm\\mszlu-xt-parent\\mszlu-xt-
common\\src\\main\\resources\\index.html"));
        template.process(map, fileWriter);
        Map<String, Object> map1 = new HashMap<>();
        map1.put("name", "other freemarker");
        FileWriter fileWriter1 = new FileWriter(new
File("D:\\git\\mszlu\\xm\\mszlu-xt-parent\\mszlu-xt-
common\\src\\main\\resources\\index1.html"));
        template.process(map1, fileWriter1);

        fileWriter.close();
        fileWriter1.close();

    }
}

```

上面的代码，通过不同的数据模型，得到了不同的页面，这就是模板的作用

## 2. Springboot整合Freemarker

之前我们在开发管理平台的时候，是使用的前后端分离，前端通过发送ajax请求，在浏览器的调试页面，我们可以在network很清楚的看到发送的每一个请求。

如果使用模板，我们实际上是访问一个接口，但实际上我们返回的是通过模板生成的静态页面，并且数据已经填充好了

在实际工作中，这种方式是管理平台开发的主流。甚至我们可以使用模板定义通用的管理平台，达到快速开发，快速交付的目的。

除了管理平台，我们还可以使用模板：

1. 代码生成器

定义好模板内容，直接生成我们要写的代码，比如实体类，mapper文件等

### 2.1 导入依赖

接下来，我们将管理平台改造为使用模板的方式

1. 在admin api模块中导入依赖：

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

2. 添加配置

```
# 模板配置
#处理空值 如果变量为null,转化为空字符串, 比如做比较的时候按照空字符串做比较
spring.freemarker.settings.classic_compatible=true
# 显示日期格式
spring.freemarker.settings.datetime_format=yyyy-MM-dd HH:mm
# 数字格式不用逗号隔开
spring.freemarker.settings.number_format=0.##
# 后缀
spring.freemarker.suffix=.ftl
spring.freemarker.template-loader-path=classpath:/templates
```

### 3. 创建模板

## 2.2 main.html-> main.ftl

在resources下创建templates文件夹，将main.html拷贝到其下，改为main.ftl

```
<!DOCTYPE html>
<html>
<head>
    <!-- 页面meta -->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>智慧猪管理后台</title>
    <meta name="description" content="中小学题库大全">
    <meta name="keywords" content="中小学题库大全">
    <meta content="width=device-width,initial-scale=1,maximum-
scale=1,user-scalable=no" name="viewport">
    <!-- 引入样式 -->
    <link rel="stylesheet"
href="${springMacroRequestContext.contextPath}/plugins/elementui/index.c
ss">
    <link rel="stylesheet"
href="${springMacroRequestContext.contextPath}/plugins/font-
awesome/css/font-awesome.min.css">
    <link rel="stylesheet"
href="${springMacroRequestContext.contextPath}/css/style.css">
    <style type="text/css">
    .el-main{
        position: absolute;
        top: 70px;
        bottom: 0px;
        left: 200px;
        right: 10px;
        padding: 0;
    }
    </style>
</head>
<body class="hold-transition skin-purple sidebar-mini">
    <div id="app">
        <el-container>
            <el-header class="main-header" style="height:70px;">
```

```

<nav class="navbar navbar-static-top" :class="'>
  <!-- Logo -->
  <a href="#" class="logo" style="text-align:center">
    <span class="logo-lg">码神学堂管理后台</span>
  </a>
  <div class="right-menu">
    <span class="help"><i class="fa fa-exclamation-
circle" aria-hidden="true"></i>帮助</span>
    <el-dropdown class="avatar-container right-menu-
item" trigger="click">
      <div class="avatar-wrapper">
        
          {{username}}
        </div>
        <el-dropdown-menu slot="dropdown">
          <el-dropdown-item divided>
            <span style="display:block;">修改密
码</span>
          </el-dropdown-item>
          <el-dropdown-item divided>
            <a href="/lzadmin/logout"><span
style="display:block;">退出</span></a>
          </el-dropdown-item>
        </el-dropdown-menu>
      </el-dropdown>
    </div>
  </nav>
</el-header>
<el-container>
  <el-aside width="200px">
    <el-menu>
      <el-submenu v-for="menu in menuList"
:index="menu.path">
        <template slot="title">
          <i class="fa" :class="menu.icon"></i>
          {{menu.title}}
        </template>
        <template v-for="child in menu.children">
          <el-menu-item :index="child.path">
            <a :href="child.linkUrl"
target="right">{{child.title}}</a>
          </el-menu-item>
        </template>
      </el-submenu>

```

```

        </el-menu>
      </el-aside>
    <el-container>
      <iframe name="right" class="el-main" src="news.html"
width="100%" height="800px" frameborder="0"></iframe>
    </el-container>
  </el-container>
</div>
</body>
<!-- 引入组件库 -->
<script src="{springMacroRequestContext.contextPath}/js/vue.js">
</script>
<script
src="{springMacroRequestContext.contextPath}/plugins/elementui/index.js
"></script>
<script type="text/javascript"
src="{springMacroRequestContext.contextPath}/js/jquery.min.js">
</script>
<script src="{springMacroRequestContext.contextPath}/js/axios-
0.18.0.js"></script>
<script>
  new Vue({
    el: '#app',
    method:{

    },
    created:function () {
      axios.post("/lzadmin/user/userInfo").then((res)=>{
        this.username = res.data.result;
      });
      axios.post("/lzadmin/user/userMenuList").then((res)=>{
        this.menuList = res.data.result;
      });
    },
    data:{
      username:"",
      menuList:[
        {
          "path": "1",
          "title": "内容管理",
          "icon":"fa-user-md",
          "children": [
            {
              "path": "/1-1",

```

```
        "title": "题库管理",
        "linkUrl": "pages/topic.html",
        "children": []
    },
    {
        "path": "/1-2",
        "title": "学科管理",
        "linkUrl": "subject.html",
        "children": []
    },
    {
        "path": "/1-3",
        "title": "课程管理",
        "linkUrl": "course.html",
        "children": []
    },
]
},
{
    "path": "2",
    "title": "订单管理",
    "icon": "fa-tty",
    "children": [
        {
            "path": "/2-1",
            "title": "订单管理",
            "linkUrl": "order.html",
            "children": []
        },
        {
            "path": "/2-2",
            "title": "邀请信息",
            "linkUrl": "invite.html",
            "children": []
        },
    ],
}
},
{
    "path": "3",
    "title": "网站管理",
    "icon": "fa-tty",
    "children": [
        {
            "path": "/3-2",
```

```
        "title": "新闻管理",
        "linkUrl": "news.html",
        "children": []
    },
    {
        "path": "/3-3",
        "title": "案例管理",
        "linkUrl": "case.html",
        "children": []
    },
]
},
{
    "path": "4",
    "title": "营销管理",
    "icon": "fa-tty",
    "children": [
        {
            "path": "/4-1",
            "title": "海报管理",
            "linkUrl": "bill.html",
            "children": []
        },
        {
            "path": "/4-2",
            "title": "优惠券管理",
            "linkUrl": "coupon.html",
            "children": []
        },
        {
            "path": "/4-3",
            "title": "用户管理",
            "linkUrl": "user.html",
            "children": []
        },
    ],
}
},
{
    "path": "5",
    "title": "系统管理",
    "icon": "fa-tty",
    "children": [
        {
            "path": "/5-1",
            "title": "角色管理",
```



```

        "linkUrl": "pages/role.html",
        "children": []
    },
    {
        "path": "/5-2",
        "title": "权限管理",
        "linkUrl": "pages/permission.html",
        "children": []
    },
    {
        "path": "/5-3",
        "title": "用户管理",
        "linkUrl": "pages/user.html",
        "children": []
    },
    {
        "path": "/5-4",
        "title": "菜单管理",
        "linkUrl": "pages/menu.html",
        "children": []
    },
]
}
]
}
});
$(function() {
    var wd = 200;
    $(".el-main").css('width', $('body').width() - wd + 'px');
});
</script>
</html>

```

## 2.3 通过模板展示页面

```

package com.mszlu.xt.admin.controller;

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

@Controller

```

```

@RequestMapping("xt")
public class AdminMainController {

    @RequestMapping("index")
    public ModelAndView mainPage(){
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.setViewName("main");
        return modelAndView;
    }
}

```

访问<http://localhost:8228/lzadmin/xt/index> 就相当于访问了原来的main.html

## 2.4 更改菜单数据获取方式

之前获取菜单的时候是通过ajax的方式进行接口请求，并赋值数据。

现在我们在访问index接口的时候，将数据填充进去

```

package com.mszlu.xt.admin.controller;

import com.mszlu.common.model.CallResult;
import com.mszlu.xt.admin.model.param.AdminUserParam;
import com.mszlu.xt.admin.service.AdminUserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
@RequestMapping("xt")
public class AdminMainController {

    @Autowired
    private AdminUserService adminUserService;

    @RequestMapping("index")
    public ModelAndView mainPage(){
        ModelAndView modelAndView = new ModelAndView();
    }
}

```

```

        //菜单
        CallResult callResult = adminUserService.userMenuList(new
AdminUserParam());
        modelAndView.addObject("menuResult",callResult);
        //登录用户
        Object principal =
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
        if (principal instanceof UserDetails){
            UserDetails userDetails = (UserDetails) principal;

            modelAndView.addObject("username",userDetails.getUsername());
        }
        modelAndView.setViewName("main");
        return modelAndView;
    }
}

```

```

<!DOCTYPE html>
<html>
<head>
    <!-- 页面meta -->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>智慧猪管理后台</title>
    <meta name="description" content="中小学题库大全">
    <meta name="keywords" content="中小学题库大全">
    <meta content="width=device-width,initial-scale=1,maximum-
scale=1,user-scalable=no" name="viewport">
    <!-- 引入样式 -->
    <link rel="stylesheet"
href="${springMacroRequestContext.contextPath}/plugins/elementui/index.c
ss">
    <link rel="stylesheet"
href="${springMacroRequestContext.contextPath}/plugins/font-
awesome/css/font-awesome.min.css">
    <link rel="stylesheet"
href="${springMacroRequestContext.contextPath}/css/style.css">
    <style type="text/css">
        .el-main{
            position: absolute;
            top: 70px;
            bottom: 0px;
            left: 200px;

```

[illegible]

```

        <i class="fa" class="{menu.icon}">
    </i>

        {menu.title}
    </template>
    <#list menu.children as child>
    <template>
        <el-menu-item index="{child.path}">
            <a
href="{springMacroRequestContext.contextPath + '/pages/' +
child.linkUrl}" target="right">{child.title}</a>
            </el-menu-item>
        </template>
    </#list>
    </el-submenu>
</#list>
</el-menu>
</el-aside>
<el-container>
    <iframe name="right" class="el-main" src="news.html"
width="100%" height="800px" frameborder="0"></iframe>
    </el-container>
</el-container>
</div>
</body>
<!-- 引入组件库 -->
<script src="{springMacroRequestContext.contextPath}/js/vue.js">
</script>
<script
src="{springMacroRequestContext.contextPath}/plugins/elementui/index.js
"></script>
<script type="text/javascript"
src="{springMacroRequestContext.contextPath}/js/jquery.min.js">
</script>
<script src="{springMacroRequestContext.contextPath}/js/axios-
0.18.0.js"></script>
<script>
    new Vue({
        el: '#app',
        method:{

        },
        created:function () {
        },
        data:{

```

```

    }
  });
  $(function() {
    var wd = 200;
    $(".el-main").css('width', $('body').width() - wd + 'px');
  });
</script>
</html>

```

不要忘了权限放行

## 3. freemarker指令

### 3.1 if指令

```

@RequestMapping("test")
public ModelAndView test(){
    ModelAndView modelAndView = new ModelAndView();
    modelAndView.addObject("flag",1);
    modelAndView.setViewName("test");
    return modelAndView;
}

@RequestMapping("test1")
public ModelAndView test1(){
    ModelAndView modelAndView = new ModelAndView();
    modelAndView.addObject("flag",2);
    modelAndView.setViewName("test");
    return modelAndView;
}

```

test.ftl:

```

<!DOCTYPE html>
<html>
<head>
    <!-- 页面meta -->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
</head>

```

```
<body>

    <#if flag==1>
        1 进来这里
    <#elseif flag==2>
        2 进来这里
    <#else>
        其他来这里
    </#if>

</body>
</html>
```

## 3.2 include指令

footer.ftl

```
<hr>
<i>
Copyright (c) 2000 <a href="http://www.mszlu.com">mszlu Inc</a>,
<br>
All Rights Reserved.
</i>
```

test.ftl

```
<!DOCTYPE html>
<html>
<head>
    <!-- 页面meta -->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
</head>

<body>

    <#if flag==1>
        1 进来这里
    <#elseif flag==2>
        2 进来这里
    <#else>
        其他来这里
    </#if>
```

```
<#include "footer.ftl">
</body>
</html>
```

## 3.3 处理不存在的变量

```
${user!'vistor'}
```

## 3.4 自定义指令

### 3.4.1 使用macro定义宏

#### 3.4.1.1 未带参数的宏定义

```
<#macro greet>
    <span style="font-size: 144%; ">Hello Mszlu!</span>
</#macro>

<!--未带参数宏调用-->
<@greet></@greet>
```

#### 3.4.1.2 带参数的宏定义

```
<#macro greet_2 person>
    <span style="font-size: 144%; ">Hello ${person}!</span>
</#macro>
<!--带参数宏调用-->
<@greet_2 person="码神之路"/>
```

### 3.4.2 使用TemplateDirectiveModel扩展

将资料中的DirectiveHandler, TemplateDirective, TemplateModelUtils拷贝过来

```
package com.mszlu.xt.admin.template;

import cn.hutool.core.util.StrUtil;
```



```

import org.springframework.stereotype.Component;

//自定义宏模板
@Component
public class StringTemplate extends TemplateDirective {

    @Override
    public String getName() {
        return "strstr";
    }

    @Override
    public void execute(DirectiveHandler handler) throws Exception {
        // 获取参数
        String username = handler.getString("username");
        String city = handler.getString("city");
        // 处理参数
        String template = "{}来自{}.";
        String format = StrUtil.format(template, username, city);
        // 传回去
        handler.put(RESULT, format).render();
    }
}

```

```

package com.mszlu.xt.admin.template;

import freemarker.template.TemplateModelException;
import org.springframework.stereotype.Component;

import java.util.Date;
import java.util.List;

/**
 * FreeMarker自定义标签
 */
@Component
public class TimeAgoMethod extends DirectiveHandler.BaseMethod {

    private static final long ONE_MINUTE = 60000L;
    private static final long ONE_HOUR = 3600000L;
    private static final long ONE_DAY = 86400000L;
    private static final long ONE_WEEK = 604800000L;
}

```

```
private static final String ONE_SECOND_AGO = "秒前";
private static final String ONE_MINUTE_AGO = "分钟前";
private static final String ONE_HOUR_AGO = "小时前";
private static final String ONE_DAY_AGO = "天前";
private static final String ONE_MONTH_AGO = "月前";
private static final String ONE_YEAR_AGO = "年前";
private static final String ONE_UNKNOWN = "未知";

public static String format(Date date) {
    if (null == date) {
        return ONE_UNKNOWN;
    }
    long delta = new Date().getTime() - date.getTime();
    if (delta < 1L * ONE_MINUTE) {
        long seconds = toSeconds(delta);
        return (seconds <= 0 ? 1 : seconds) + ONE_SECOND_AGO;
    }
    if (delta < 45L * ONE_MINUTE) {
        long minutes = toMinutes(delta);
        return (minutes <= 0 ? 1 : minutes) + ONE_MINUTE_AGO;
    }
    if (delta < 24L * ONE_HOUR) {
        long hours = toHours(delta);
        return (hours <= 0 ? 1 : hours) + ONE_HOUR_AGO;
    }
    if (delta < 48L * ONE_HOUR) {
        return "昨天";
    }
    if (delta < 30L * ONE_DAY) {
        long days = toDays(delta);
        return (days <= 0 ? 1 : days) + ONE_DAY_AGO;
    }
    if (delta < 12L * 4L * ONE_WEEK) {
        long months = toMonths(delta);
        return (months <= 0 ? 1 : months) + ONE_MONTH_AGO;
    } else {
        long years = toYears(delta);
        return (years <= 0 ? 1 : years) + ONE_YEAR_AGO;
    }
}

private static long toSeconds(long date) {
    return date / 1000L;
}
```

```

private static long toMinutes(long date) {
    return toSeconds(date) / 60L;
}

private static long toHours(long date) {
    return toMinutes(date) / 60L;
}

private static long toDays(long date) {
    return toHours(date) / 24L;
}

private static long toMonths(long date) {
    return toDays(date) / 30L;
}

private static long toYears(long date) {
    return toMonths(date) / 365L;
}

@Override
public Object exec(List arguments) throws TemplateModelException {
    Date time = getDate(arguments, 0);
    return format(time);
}
}

```

```

<br />
<@sstr username="lisi" city="beijing">
    ${result}
</@sstr>

<br />
# date是后台传递过来的参数
${timeAgo(date)}

```

```

package com.mszlu.xt.admin.template;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;

import javax.annotation.PostConstruct;

@Configuration

```

```
public class FreemarkerConfig {

    @Autowired
    StringTemplate stringTemplate;
    @Autowired
    private freemarker.template.Configuration configuration;

    @PostConstruct
    public void setUp() {
        configuration.setSharedVariable("timeAgo", new TimeAgoMethod());
        configuration.setSharedVariable("strstr", stringTemplate);
    }

}
```