

记录时间: 2019.10.21  
公司名称: 火鹰科技有限公司  
学生姓名: 曾伟涛  
记录类型: 学习笔记  
记录阶段: 2019.10.21-2019.10.27

# thymeleaf

## 1.概念

一种用于Web和独立环境的现代服务器端的Java模板引擎。主要目标是将优雅的自然模板带到开发工作流程中，并将HTML在浏览器中正确显示，并且可以作为静态原型，让开发团队能更容易地协作。Thymeleaf能够处理HTML，XML，JavaScript，CSS甚至纯文本。

## 2.语法 (你猜我写不写，做梦)

<https://www.cnblogs.com/itdragon/archive/2018/04/13/8724291.html>

## 3.spring boot 整合 thymeleaf

1.在pom.xml导入依赖:

```
1 <dependency>
2 <groupId>org.springframework.boot</groupId>
3 <artifactId>spring-boot-starter-thymeleaf</artifactId>
4 </dependency>
```

2.application.yml 中配置 Thymeleaf

```
1 thymeleaf:
2   #模板的模式，支持 HTML，XML TEXT JAVASCRIPT
3   mode: HTML5
4   #编码 可不用配置
5   encoding: UTF-8
6   #内容类别，可不用配置
7   content-type: text/html
8   #开发配置为false，避免修改模板还要重启服务器
9   cache: false
10  #配置模板路径，默认是templates，可以不用配置
11  prefix: classpath:/templates/
```

请注意这里的prefix，不管你是否写了这个路径，thymeleaf的html文件都只能放在resources目录下的templates文件内，不然一切都无效。

3.html引用thymeleaf必加代码：在<html 请加在此处></html>

```
1 xmlns:th="http://www.thymeleaf.org"
```

## 4.入门必经之路：hello world!

新建一个controller类：以我为例：（HelloWorldController）

```
1 @Controller
2 @RequestMapping(value = "/test")
3 public class HelloWroldController {
4     @RequestMapping("/index")
5     public String hello(Model model){
6         model.addAttribute("name", "Hello World!");
7         return "index";
8     }
9 }
```

```
8     }
9 }
```

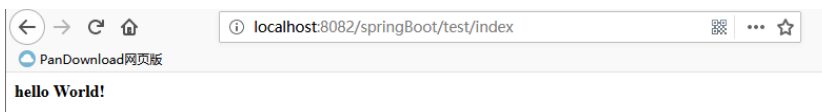
index.html:

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <meta charset="UTF-8"/>
5     <title>首页</title>
6 </head>
7 <body>
8     <b th:text="${name}"></b><b/>
9 </body>
10 </html>
```

在浏览器输入localhost:8082/test/index 会看到：

说明一点：8082端口是我这边设置的，以及springBoot是我之前的应用写得，故需要输入这么长，练习者只需要在application.yml中设置端口号即可：

```
1     server:
2         port:8082
```



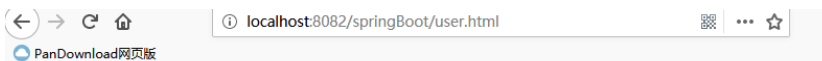
2.进阶：实现登录功能：

继第五周周报，写了user.html这个页面，是输入用户名与密码，点击提交跳到登录成功页面，在此稍微改动，即可实现登录功能。我们在之前的HelloWorldController:进行修改，修改如下：

```
1 public ModelAndView addUser5(UserModel user, ModelAndView mv) {
2     if(user.getUserName().equals("") && user.getPassword().equals("")){
3         mv.setViewName("user");
4         mv.addObject("name", "不能为空");
5     }else{
6         mv.setViewName("index");
7         mv.addObject("name",user.getUserName());
8     }
9     return mv;
```

效果顺序如下：

在浏览器地址栏输入：localhost:8082/user.html,输入用户名和密码

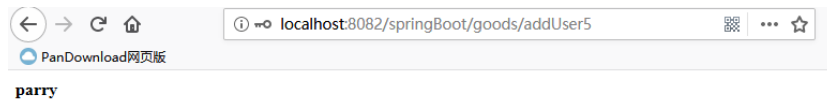


## 用户登录

username:

password:

点击登录之后便显示了我们输入的用户名



## FreeMarker

说明：freemarker不做过多详细说明，只是让你们了解有这么一个模板存在，官方不推荐使用此模板。

### 1.导入依赖

```
1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-freemarker</artifactId>
4 </dependency>
```

### 2.application.yml配置：

```
1 freemarker:
2   # 设置模板后缀名
3   suffix: .ftl
4   # 设置文档类型
5   content-type: text/html
6   # 设置页面编码格式
7   charset: UTF-8
8   # 设置页面缓存
9   cache: false
10  check-template-location: true
11  prefix:
12    classpath: /templates
```

### 3.在resource目录下templates下创建一个文件，是一个文件不是HTML文件，后缀名是.ftl，以我的为例：hello.ftl

```
1 ${name}
```

!!!就这么简单，你是不是缩水了，不，freemarker创建出来的其实是个纯文本。当你需要html的样式的时候，打个“<”就会有提示了



我们在thymeleaf的HelloWorldController再加一下代码：

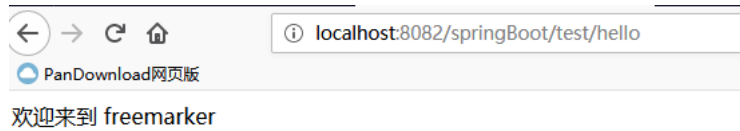
```
1 @RequestMapping(value = "/hello")
```

```

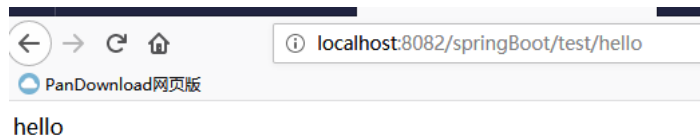
2 public String index(Model model){
3     model.addAttribute("name","欢迎来到 freemarker");
4     return "hello";
5 }

```

编译运行，在浏览器地址栏输入localhost:8082/test/hello,会出现以下效果：



附：配置过程中可能出现的问题：



为什么输出的是hello，而不是上面的“欢迎来到 freemarker”这样的话呢，也就是没有跳到hello.ftl页面，原因如下：先给整段代码：

```

1 @RestController
2 @RequestMapping(value = "/test")
3 public class TestController {
4
5     @RequestMapping("/index")
6     public String index(ModelMap map) {
7         map.addAttribute("name", "hello World!");
8         return "index";
9     }
10
11     @RequestMapping("/hello")
12     public String index(Model model){
13         model.addAttribute("name","欢迎来到 freemarker");
14         return "hello";
15     }

```

用错了注解：在这段代码中，我们用了@RestController，它是@Controller+@ResponseBody,造成了它无法跳转到hello.ftl，我们需要改成@Controller

若重新编译看浏览器还出现这个问题（你问度娘吧）还是注解问题，在一个类的上面的@RequestMapping中，不应该是@RequestMapping(value = "/test"),而下面与其相反，这也会造成这样的问题出现。

正确的格式代码如下：

```

1 @Controller
2 @RequestMapping("/test")
3 public class TestController {
4

```

```

5     @RequestMapping(value = "/index")
6     public String index(ModelMap map) {
7         map.addAttribute("name", "hello World!");
8         return "index";
9     }
10
11    @RequestMapping(value = "/hello")
12    public String index(Model model){
13        model.addAttribute("name", "欢迎来到 freemarker");
14        return "hello";
15    }

```

## Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Fri Oct 25 12:42:54 CST 2019

There was an unexpected error (type=Internal Server Error, status=500).

Exception parsing document: template="index", line 6 - column 3

出现这种错误的，已经很明显了，看最后一行，意思是模板文件为index，他的第六行第三列报错

```

<head>
<meta charset="UTF-8">
<title>首页</title>
</head>
<body>

```

看到 没错啊，不，thymeleaf比较严格，我们发现meta没有结束标签，给他加个结束标签就没事了