

SpringBoot：模板引擎之Thymeleaf

一、Thymeleaf

1.1 简介

Thymeleaf: Java模板引擎，模板格式.html

Freemarker、JSP差不多

1.2 组成

文本 直接输出的的内容

表达式 可以获取域中的的内容 并且进行技术

- 简单表达：
 - 变量表达式: `${...}`
 - 选择变量表达式: `*{...}`
 - 消息表达式: `#{...}`
 - 链接网址表达式: `@{...}`
 - 片段表达式: `~{...}`
- 字面
 - 文本文字: `'one text' , 'Another one!' , ...`
 - 号码文字: `0 , 34 , 3.0 , 12.3 , ...`
 - 布尔文字: `true , false`
 - 空文字: `null`
 - 文字标记: `one , sometext , main , ...`
- 文字操作：
 - 字符串连接: `+`
 - 字面替换: `|The name is ${name}|`
- 算术运算：
 - 二元运算符: `+, -, *, / , %`
 - 减号 (一元运算符): `-`
- 布尔运算：
 - 二元运算符: `and , or`
 - 布尔否定 (一元运算符): `!, not`

- 比较和平等：
 - 比较： `>` , `<` , `>=` , `<=` (`gt` , `lt` , `ge` , `le`)
 - 平等运营商： `==` , `!=` (`eq` , `ne`)
- 有条件的运营商：
 - IF-THEN: (`if`) ? (`then`)
 - IF-THEN-ELSE: (`if`) ? (`then`) : (`else`)
 - 默认: (`value`) ?: (`defaultvalue`)
- 特殊代币：
 - 无操作: `_`

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

1.3 核心技术

1、获取数据

属性标签

th:xxx="\${属性名称}"

属性

th:insert
th:replace

th:each

th:if
th:unless
th:switch
th:case

th:object
th:with

th:attr
th:attrprepend
th:attrappend

th:value
th:href
th:src
...

th:text
th:utext

th:fragment

th:remove

2、分支和循环

th:if="条件验证"

th:each="变量名:\${属性名称}"

二、SpringBoot整合Thymeleaf

使用步骤：

1、依赖jar

、、

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-thymeleaf</artifactId>
  <version>2.1.6.RELEASE</version>
</dependency>
```

2、配置

application.properties

``

```
spring.thymeleaf.mode=HTML
spring.thymeleaf.suffix=.html
spring.thymeleaf.encoding=UTF-8
spring.thymeleaf.prefix=classpath:/templates/
```

3、编写模板

``

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Thymeleaf初体验</title>
</head>
<body>
<div>
  <h1>基本应用</h1>
  <p th:text="'内容: '${str1}'"></p>
  <input th:value="'数字: '${num1}'"></input><br/>
  <span th:text="'${company.address}'"></span>
<p></p>
  <h1>分支</h1>
  <p>
    <p th:if="${sex}==0">女</p>
    <p th:if="${sex}==1">男</p>
  </p>
  <h1>循环</h1>
  <div>
    <ul>
      <li th:each="c:${companylist}">
        <h5 th:text="${c.id}"></h5>
        <h5 th:text="${c.name}"></h5>
        <h5 th:text="${c.num}"></h5>
        <h5 th:text="${c.address}"></h5>
      </li>
    </ul>
  </div>
</div>
</div>
</body>
</html>
```

4、编写代码 控制器

..

```
@Controller
public class FirstApi {
    @RequestMapping("first/hi.do")
    public String hi(Model model){
        model.addAttribute("str1","我是xxx");
        model.addAttribute("num1",1001);
        Company company = new Company();
        company.setId(1);
        company.setName("河南狼来了科技有限公司");
        company.setAddress("郑州二七海为");
        company.setNums(100);
        model.addAttribute("company",company);
        model.addAttribute("sex","0");
        List<Company> list=new ArrayList<>();
        for(int i=1;i<10;i++){
            Company cm = new Company();
            cm.setId(i);
            cm.setName("河南狼来了"+i);
            cm.setAddress("郑州二七海为");
            cm.setNums(100);
            list.add(cm);
        }
        model.addAttribute("companylist",list);
        return "first";
    }
}
```

5、运行并测试



基本应用

内容：我是xxx

数字：1001

郑州二七海为

分支

女

循环

- 1

河南狼来了1

100

郑州二七海为

- 2

河南狼来了2

100