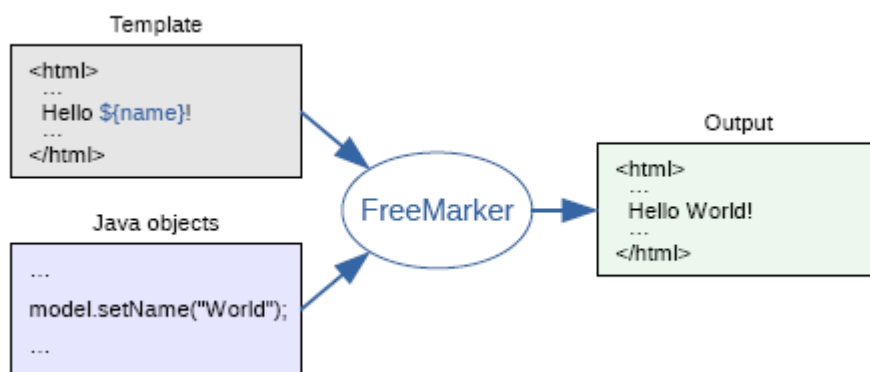


SpringBoot：模板引擎之Freemarker

一、Freemarker

1.1 简介

FreeMarker 是一款 *模板引擎*：即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。



FreeMarker Template Language (FTL)，企业开发主要用来实现静态页面 定时、模板 生成静态页面 动态页面静态化技术

作用：

- 1、模板 生成静态页面
- 2、模板 输出 类似SP

1.2 组成

1. 文本 直接输出的内容
2. 注释 <#--注释内容-->
3. 插值 \${域中的属性名称}
4. ftl指令
 1. 自定义的标签 以#开头
 1. 常用指令：

```
        <#if condition>
            ...
        <#elseif condition2>
            ...
        <#elseif condition3>
            ...
        ...
        <#else>
            ...
        </#if>

<#switch value>
    <#case refValue1>
        ...
        <#break>
    <#case refValue2>
        ...
        <#break>
    ...
    <#case refValueN>
        ...
        <#break>
    <#default>
        ...
</#switch>
```

这里：

- `value`, `refValue1`, 等：表达式将会计算成相同类型的标量。

`break` 和 `default` 是可选的。

形式 1:

```
<#list sequence as item>
    Part repeated for each item
<#else>
    Part executed when there are 0 items
</#list>
```

这里:

- `else` 部分是可选的, 而且仅仅从 FreeMarker 2.3.23 版本开始支持。
- `sequence`: 将我们想要迭代的项, 算作是序列或集合的表达式
- `item`: 循环变量的名称 (不是表达式)
- 在标签之间的多个 "parts" 可以是任意的 FTL (包括嵌套的 `list`)

形式 2 (从 FreeMarker 2.3.23 版本开始):

```
<#list sequence>
    Part executed once if we have more than 0 items
    <#items as item>
        Part repeated for each item
    </#items>
    Part executed once if we have more than 0 items
<#else>
    Part executed when there are 0 items
</#list>
```

这里: 和上面形式1的 "这里" 部分相同。

1.3 核心技术

1、插值

语法格式: `${属性名称}`

可以获取 `HttpServletRequest` 的 `Attribute` 的数据

基本类型和引用类型

2、指令

分支和循环

`#if`

`#swatch`

#list

二、SpringBoot整合Freemarker

整合步骤：

1、依赖jar

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

2、配置

``

```
spring.freemarker.suffix=.ftl
spring.freemarker.charset=UTF-8
spring.freemarker.content-type=text/html
spring.freemarker.template-loader-path=classpath:/templates
```

3、编写模板

``

```
<html>
<head>
  <title>Freemarker初体验</title>
</head>
<body>
<div>
  <h1>插值的应用</h1>
  <ul>
    <li>基本类型: ${num1}</li>
    <li>基本类型: ${num1*2}</li>
    <li>引用类型: ${str1}</li>
    <li>引用类型: ${company.name}</li>
  </ul>
</div>
<div>
  <h1>指令应用</h1>
  <#if level=='P5'>
    <h2 style="color: green">中级: 7-10K</h2>
  <#elseif level=='P7'>
    <h1 style="color: red">初级: 12K-15K</h1>
  <#elseif level=='P6'>
    <h1 style="color: red">高级: 9K-13K</h1>
  </#if>
</div>
</body>
</html>
```

```

        <ul>
            <#list companylist as c>
                <li>序号: ${c.id} , 名称: ${c.name}, 人数: ${c.nums}</li>
            </#list>

        </ul>
    </div>
</body>
</html>

```

4、编写控制器

```

import lombok.Data;

/**
 * @Author feri
 * @Date Created in 2019/7/4 14:29
 */
@Data
public class Company {
    private int id;
    private String name;
    private String address;
    private int nums;
}

```

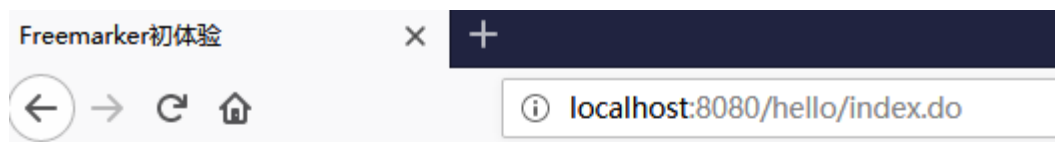
```

@Controller
public class HelloApi {
    @RequestMapping("hello/index.do")
    public String hi(Model model) {
        model.addAttribute("num1", 168);
        model.addAttribute("str1", "我爱Java");
        Company company = new Company();
        company.setId(1);
        company.setName("河南狼来了");
        company.setAddress("郑州二七海为");
        company.setNums(100);
        model.addAttribute("company", company);
        model.addAttribute("level", "P6");
        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 "/hello";  
}  
}
```

5、运行并测试



插值的应用

- 基本类型: 168
- 基本类型: 336
- 引用类型: 我爱Java
- 引用类型: 河南狼来了

指令应用

高级: 9K-13K

- 序号: 1, 名称: 河南狼来了1, 人数: 100
- 序号: 2, 名称: 河南狼来了2, 人数: 100
- 序号: 3, 名称: 河南狼来了3, 人数: 100
- 序号: 4, 名称: 河南狼来了4, 人数: 100
- 序号: 5, 名称: 河南狼来了5, 人数: 100
- 序号: 6, 名称: 河南狼来了6, 人数: 100
- 序号: 7, 名称: 河南狼来了7, 人数: 100
- 序号: 8, 名称: 河南狼来了8, 人数: 100
- 序号: 9, 名称: 河南狼来了9, 人数: 100