1.1 Freemarker

1.1.1 什么是freemarker

FreeMarker是一个用Java语言编写的模板引擎，它基于模板来生成文本输出。FreeMarker与Web容器无关，即在Web运行时，它并不知道Servlet或HTTP。它不仅可以用作表现层的实现技术，而且还可以用于生成XML，JSP或Java 等。

1.1.2 Freemarker的使用方法

第一步：把freemarker的jar包添加到工程中
第二步：freemarker的运行不依赖web容器，可以在java工程中运行。创建一个测试方法进行测试。
第三步：创建一个Configration对象
第四步：告诉config对象模板文件存放的路径。
第五步：设置config的默认字符集。一般是utf-8
第六步：从config对象中获得模板对象。需要制定一个模板文件的名字。
第七步：创建模板需要的数据集。可以是一个map对象也可以是一个pojo，把模板需要的数据都放入数据集。
第八步：创建一个Writer对象，指定生成的文件保存的路径及文件名。
第九步：调用模板对象的process方法生成静态文件。需要两个参数数据集和writer对象。
第十步：关闭writer对象。

1.1.3 代码实现

```
1  public class FreemarkerTest {
2
3      @Test
4      public void testFreemarker() throws Exception {
5  //  第一步：把freemarker的jar包添加到工程中
6  //  第二步：freemarker的运行不依赖web容器，可以在java工程中运行。创建一个测试方法进行测试。
7  //  第三步：创建一个Configration对象
8          Configuration configuration = new Configuration(Configuration.getVersion());
9  //  第四步：告诉config对象模板文件存放的路径。
10         configuration.setDirectoryForTemplateLoading(new File("C:\\Users\\代炜\\workspace\\taotao-portal\\src\\main\\webapp\\WEB-INF\\ftl"));
11 //  第五步：设置config的默认字符集。一般是utf-8
```

```
        configuration.setDefaultEncoding("utf-8");
//  第六步：从config对象中获得模板对象。需要制定一个模板文件的名字。
        Template template = configuration.getTemplate("test.ftl");
//  第七步：创建模板需要的数据集。可以是一个map对象也可以是一个pojo，把模板需要的数据都放入数据集。
        Map map = new HashMap<>();
        map.put("hello", "hello freemarker");
//  第八步：创建一个Writer对象，指定生成的文件保存的路径及文件名。
        Writer writer = new FileWriter(new File("E:\\freemarker\\test.html"));
//  第九步：调用模板对象的process方法生成静态文件。需要两个参数数据集和writer对象。
        template.process(map, writer);
//  第十步：关闭writer对象。
        writer.close();
    }
}
```

模板：

```
${hello}
```

1.1.4 Freemarker模板的写法

1.1.4.1 取简单数据类型数据

使用EL表达式。

${hello}

1.1.4.2 包装数据类型

模板：

```
<html>
<head>
<title>${title}</title>
</head>
<body>
<label>学号：</label>${student.id}<br>
<label>姓名：</label>${student.name}<br>
<label>住址：</label>${student.address}<br>
</body>
```

```
</html>
```

1.1.4.3 历遍集合/数组

List<Person> persons = new ArrayList<Person>();
省略….

页面中内容

```
<#list persons as p>
${p.id}/${p.name}
</#list>
```

1.1.4.4 获得当前迭代的索引

List<Person> list = new ArrayList<Person>();

获取当前迭代的索引:

```
<#list persons as p>
${p_index}
</#list>
```

1.1.4.5 模板中判断条件

```
<#if 判断条件>
<#else>
</#if>
```

逻辑运算符（==　　!=　　||　　&&）

1.1.4.6 日期类型格式化

默认格式
1：date
${cur_time?date}
2：datetime
${cur_time?datetime}
3：time
${cur_time?time}

自定义格式
${cur_time?string("yyyy-MM-dd HH:mm:ss")}

1.1.4.7 处理null值

root.put(“val”,null);

解决办法
1:null 变 空串
${val!}　　${val!"这里是空"}
2:为Null时给默认值
${val!“我是默认值"}
3、<#if curdate ??>
当前日期:${curdate?string("yyyy/MM/dd HH:mm:ss")}
<#else>
curdate属性为null
</#if>

1.1.4.8 Include

将另一个页面引入本页面时可用以下命令完成

<#include "/include/head.html">

1.2 项目中使用freemarker

使用freemarker整合spring。把Configuration交给spring容器管理。
依赖的jar包：

```
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-context-support</artifactId>
<version>4.1.3.RELEASE</version>
</dependency>
<dependency>
<groupId>org.freemarker</groupId>
<artifactId>freemarker</artifactId>
<version>2.3.23</version>
</dependency>
```

1.2.1 Spring配置文件

```
<bean id="freemarkerConfig"
class="org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer">
<property name="templateLoaderPath" value="/WEB-INF/ftl/" />
<property name="defaultEncoding" value="UTF-8" />
</bean>
```