**品优购电商系统开发**

**第12章**

**Freemarker模板引擎**

**商品详情系统**

传智播客.黑马程序员

# 课程目标

目标1：掌握Freemarker常用的指令和内建函数

目标2：完成商品详情基本数据显示

目标3：通过VueJS显示动态效果

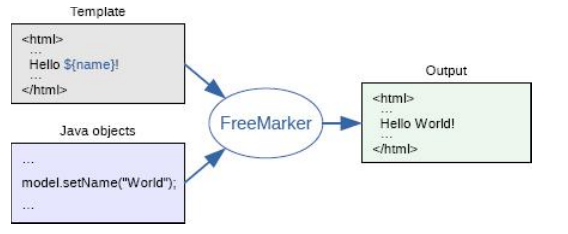
目标4：完成商品详细页读取SKU信息的业务逻辑

目标5：完成系统模块对接功能

# 1.FreeMarker模板引擎

## 1.1什么是Freemarker

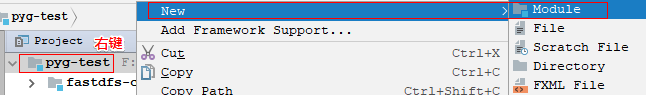
FreeMarker 是一个用 Java 语言编写的模板引擎，它基于模板来生成输出各种文件。它不仅可以用作表现层的实现技术，而且还可以用于生成 XML，html 或 Java 等。

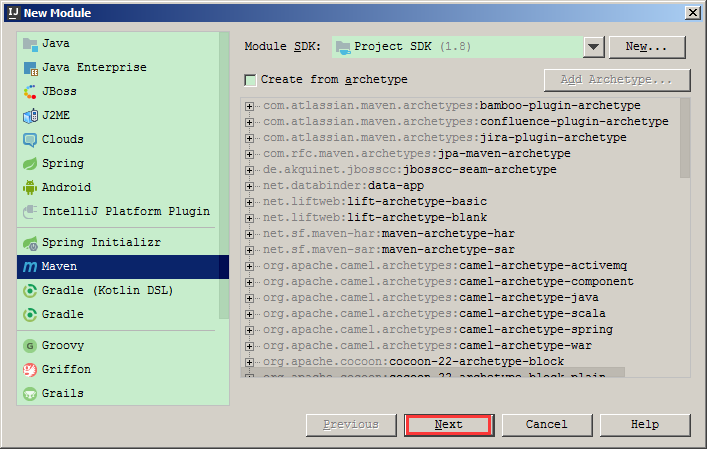


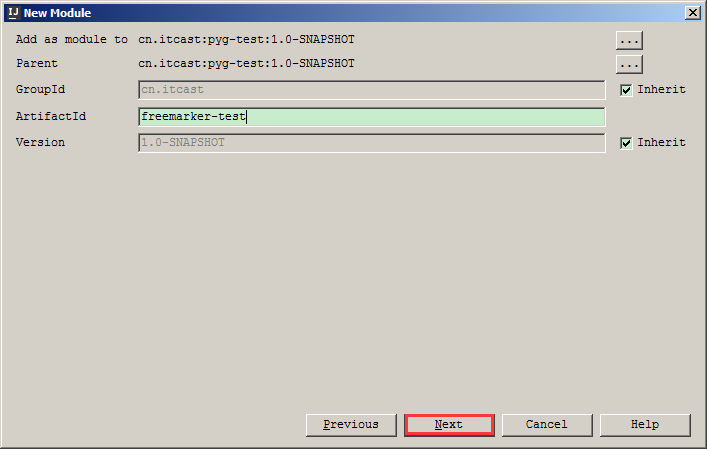
## 1.2 FreeMarker入门示例

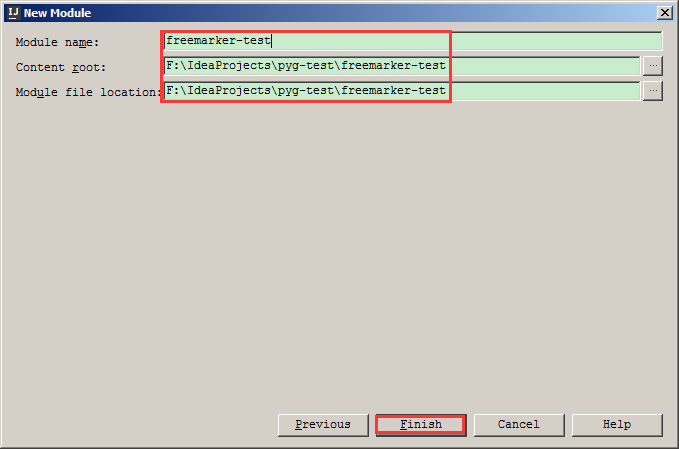
### 1.2.1搭建测试工程

创建freemarker-test模块(jar)









### 1.2.2引入依赖

freemarker-test/pom.xml:

|  |
| --- |
| <**dependency**>  <**groupId**>org.freemarker</**groupId**>  <**artifactId**>freemarker</**artifactId**>  <**version**>2.3.28</**version**> </**dependency**> |

### 1.2.3创建模板文件

创建一个简单的模板文件freemarker-test/src/main/resources/

ftl/hello.ftl：

|  |
| --- |
| <!DOCTYPE **html**> <**html lang="zh-cn"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>FreeMarker示例</**title**>  <**meta name="Copyright" content="All Rights Reserved"**/> </**head**> <**body**>  *<#--我只是一个注释，我不会有任何输出 -->* <**h1**>**${name}**,你好。**${message}**!</**h1**> </**body**> </**html**> |

说明：这里有文本、取值和注释

### 1.2.4编写代码

使用步骤：

第一步：创建一个 Configuration 对象，直接 new 一个对象，构造方法的参数就是 freemarker的版本号。

第二步：设置模板文件加载的基础路径。

第三步：设置模板文件使用的字符集。(一般就是 UTF-8)

第四步：加载一个模板文件产生模板对象。

第五步：创建一个模板使用的数据模型，可以是 pojo 也可以是 map，一般是 Map。

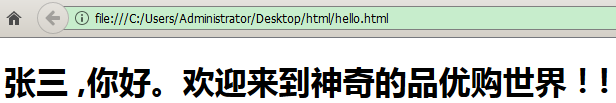
第六步：调用模板对象的 process 方法填充模版，输出文件。

在freemarker-test/src/main/test/cn/itcast/freemarker包下，编写测试类

FreeMarkerTest.java：

|  |
| --- |
| */\*\* 入门示例 \*/* @Test **public void** test() **throws** Exception{  */\*\* 创建配置信息对象 \*/* Configuration configuration = **new**  Configuration(Configuration.***VERSION\_2\_3\_28***);  */\*\* 设置模板文件加载的基础路径 \*/* configuration.setClassForTemplateLoading(FreeMarker01.**class**, **"/ftl"**);  */\*\* 设置默认编码 \*/*  configuration.setDefaultEncoding(**"UTF-8"**);  */\*\* 加载模板文件，产生模板对象 \*/* Template template = configuration.getTemplate(**"hello.ftl"**);  */\*\* 定义数据模型 \*/* Map<String,Object> dataModel = **new** HashMap<>();  dataModel.put(**"name"**, **"张三 "**);  dataModel.put(**"message"**, **"欢迎来到神奇的品优购世界！"**);  */\*\* 填充模板，输出文件 \*/* template.process(dataModel,  **new** FileWriter(**"C:\\Users\\Administrator\\Desktop\\html\\hello.html"**)); } |

执行后，在C:\Users\Administrator\Desktop\html\hello.html，打开看看：



## 1.3基础语法

模板文件中四种元素：

  1、文本：直接输出的部分  
  2、注释：即<#--...-->格式不会输出  
  3、取值：即${..}部分,将使用数据模型中的变量替代输出  
  4、FTL指令：FreeMarker指令，和HTML标记类似，名字前加#予以区分，不会输出。

## 1.4 FTL指令

### 1.4.1 assign变量指令

说明：此指令用于在页面上定义一个变量

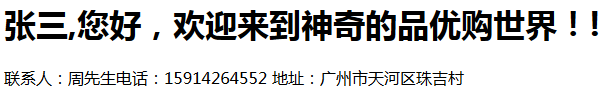
a.定义简单类型(hello.ftl)：

|  |
| --- |
| **<#assign linkman**=**"周先生"/>** 联系人：**${linkman}** |

b.定义对象类型(hello.ftl)：

|  |
| --- |
| **<#assign info**={**"mobile"**:**"15914264552"**,**'address'**:**'广州市天河区珠吉村'**}**/>** 电话：**${info**.**mobile}** 地址：**${info**.**address}** |

运行效果：



### 1.4.2 include包含指令

说明：此指令用于包含模板文件

创建模板文件freemarker-test/src/main/resources/ftl/header.ftl：

|  |
| --- |
| <**h1**>黑马信息网</**h1**> |

修改freemarker-test/src/main/resources/ftl/hello.ftl，在模板文件中使用include指令包含header.ftl模板

|  |
| --- |
| **<#include "header.ftl"/>** |

### 1.4.3 if条件指令

在模板文件上添加(hello.ftl)：

|  |
| --- |
| *<#-- 条件指令 -->* **<#if success>** 你已通过实名认证 **<#else>** 你未通过实名认证 **</#if>** |

在代码中对success变量赋值：

|  |
| --- |
| dataModel.put(**"success"**, **true**); |

### 1.4.4 list迭代指令

需求，实现用户信息列表，如下图：



代码中对变量users赋值(FreeMarkerTest.java)：

|  |
| --- |
| List<Map<String, Object>> users = **new** ArrayList<>(); Map<String,Object> user1 = **new** HashMap<>(); user1.put(**"name"**, **"李大华"**); user1.put(**"age"**, 30);  Map<String,Object> user2 = **new** HashMap<>(); user2.put(**"name"**, **"李中华"**); user2.put(**"age"**, 25);  Map<String,Object> user3 = **new** HashMap<>(); user3.put(**"name"**, **"李小华"**); user3.put(**"age"**, 20); users.add(user1); users.add(user2); users.add(user3); dataModel.put(**"users"**, users); |

在模板文件上添加(hello.ftl)：

|  |
| --- |
| <**table border="1" width="300px"**>  <**tr**>  <**th**>索引号</**th**>  <**th**>姓名</**th**>  <**th**>年龄</**th**>  </**tr**>  **<#list users as user>** <**tr**>  <**td**>**${user\_index** + 1**}**</**td**>  <**td**>**${user**.**name}**</**td**>  <**td**>**${user**.**age}**</**td**>  </**tr**>  **</#list>** </**table**> |

说明：如果想在循环中得到索引，使用 循环变量\_index就可以得到。

## 1.5内建函数

内建函数语法格式： 变量?函数名称

### 1.5.1获取集合大小

我们通常要得到某个集合的大小，如下图：



我们使用size函数来实现，代码如下(hello.ftl)：

|  |
| --- |
| 共 **${users**?**size}** 条记录。<**br**/> |

### 1.5.2转换JSON字符串为对象

将json字符串转换为对象(hello.ftl)：

|  |
| --- |
| **<#assign text**=**"{'bank':'工商银行','account':'10101920201920212'}"/> <#assign data**=**text**?**eval/>** 开户行：**${data**.**bank}** 账号：**${data**.**account}** |

### 1.5.3日期格式化

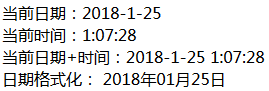
代码中对变量赋值(FreeMarkerTest.java)：

|  |
| --- |
| dataModel.put(**"today"**, **new** Date()); |

在模板文件中加入(hello.ftl)：

|  |
| --- |
| <**hr**/> 当前日期：**${today**?**date}** <**br**/> 当前时间：**${today**?**time}** <**br**/> 当前日期+时间：**${today**?**datetime}** <**br**/> 日期格式化： **${today**?**string**(**"yyyy年MM月dd日"**)**}** |

运行效果如下：



### 1.5.4数字转换为字符串

代码中对变量赋值(FreeMarkerTest.java)：

|  |
| --- |
| dataModel.put(**"point"**, 102920122); |

修改模板(hello.ftl)：

|  |
| --- |
| 累计积分：**${point}** |

页面显示：



我们会发现数字会以每三位一个分隔符显示，有些时候我们不需要这个分隔符，就需要将数字转换为字符串,使用内建函数c

|  |
| --- |
| 累计积分：**${point**?**c}** |

页面显示效果如下：



## 1.6空值处理

如果你在模板中使用了变量但是在代码中没有对变量赋值，那么运行生成时会抛出异常。但是有些时候，有的变量确实是null，怎么解决这个问题呢？

### 1.6.1判断某变量是否存在:“??”

用法为：variable??，如果该变量存在，返回true否则返回false

|  |
| --- |
| **<#if aaa**??**>** aaa变量存在 **<#else>** aaa变量不存在 **</#if>** |

### 1.6.2缺失变量默认值:“!”

我们除了可以判断是否为空值，也可以使用!对null值做转换处理，在模板文件中加入：

|  |
| --- |
| **${aaa**!**'默认值'}** |

在代码中不对aaa赋值，也不会报错了，当aaa为null则返回！后边的内容“默认值”

## 1.7运算符

### 1.7.1算术运算符

FreeMarker支持的算术运算符包括：+、 -、 \*、/、%

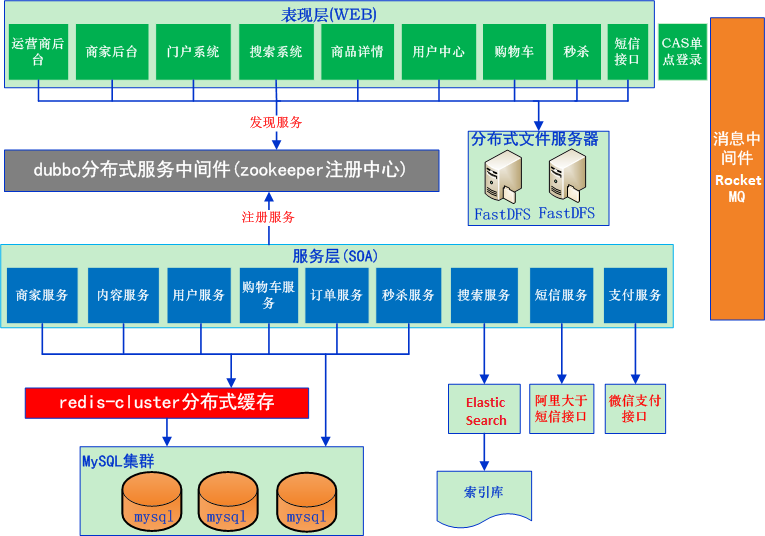
### 1.7.2逻辑运算符

逻辑与：&&   
逻辑或：||   
逻辑非：!

### 1.7.3比较运算符

1. ==：判断两个值是否相等
2. !=：判断两个值是否不等
3. >或者gt：判断左边值是否大于右边值
4. >=或者gte：判断左边值是否大于等于右边值
5. <或者lt：判断左边值是否小于右边值
6. <=或者lte：判断左边值是否小于等于右边值

# 搭建商品详情工程



## 2.1 pinyougou-item-web(war)

与pinyougou-search-web构建方式一样，拷贝pinyougou-search-web模块中的web.xml、springmvc.xml、log4j.properties

pinyougou-item-web/pom.xml：

|  |
| --- |
| <**properties**>  *<!-- 更改tomcat端口号 -->* <**port**>**9105**</**port**> </**properties**> |

pinyougou-item-web/src/main/webapp/WEB-INF/web.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>  *<!-- 配置Spring MVC前端控制器(核心控制器) -->* <**servlet**>  <**servlet-name**>pinyougou</**servlet-name**> <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  <**init-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:springmvc.xml</**param-value**>  </**init-param**>  <**load-on-startup**>1</**load-on-startup**>  </**servlet**>  <**servlet-mapping**>  <**servlet-name**>pinyougou</**servlet-name**>  <**url-pattern**>\*.html</**url-pattern**>  </**servlet-mapping**> </**web-app**> |

pinyougou-item-web/src/main/resources/springmvc.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>  *<!-- 配置开启组件扫描 -->* **<context:component-scan base-package="com.pinyougou.item.controller"/>**  *<!-- 开启MVC注解驱动 -->* <**mvc:annotation-driven**/>  *<!-- 配置静态资源用WEB容器默认的servlet来处理 -->* <**mvc:default-servlet-handler**/>   *<!-- ############## 配置dubbo服务消费者 ############## -->  <!-- 配置当前应用的名称 -->* <**dubbo:application name="pinyougou-item-web"**/>  *<!-- 配置zookeeper作为注册中心，注册服务地址 -->* **<dubbo:registry protocol="zookeeper" address="192.168.12.131:2181"/>**  *<!-- 配置采用包扫描来引用服务，产生服务接口的代理对象 -->* <**dubbo:annotation package="com.pinyougou.item.controller"**/> </**beans**> |

## 2.2静态资源

将【资料\商品静态资源】拷贝至pinyougou-item-web/src/main/webapp/目录下。

## 2.3配置域名访问

a.修改Hosts:

127.0.0.1 item.pinyougou.com

b. 修改nginx.conf配置反向代理

|  |
| --- |
| server {  listen 80;  server\_name item.pinyougou.com;  proxy\_set\_header X-Forwarded-Host $host;  proxy\_set\_header X-Forwarded-Server $host;  location / {  proxy\_pass http://127.0.0.1:9105;  proxy\_connect\_timeout 600;  proxy\_read\_timeout 600;  }  } |

## 2.4集成FreeMarker

pinyougou-item-web/pom.xml配置：

|  |
| --- |
| <**dependencies**>  *<!-- freemarker -->* <**dependency**>  <**groupId**>org.freemarker</**groupId**>  <**artifactId**>freemarker</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-context-support</**artifactId**>  </**dependency**> </**dependencies**> |

pinyougou-item-web/src/main/resources/springmvc.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>  *<!-- 配置开启组件扫描 -->* **<context:component-scan base-package="com.pinyougou.item.controller"/>**  *<!-- 开启MVC注解驱动 -->* <**mvc:annotation-driven**/>  *<!-- 配置静态资源用WEB容器默认的servlet来处理 -->* <**mvc:default-servlet-handler**/>   *<!-- 配置FreeMarkerConfigurer -->* <**bean id="freeMarkerConfigurer" class="org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer"**>  *<!-- 设置模板文件加载的基础路径 -->* <**property name="templateLoaderPath" value="/WEB-INF/ftl/"**/>  *<!-- 设置模板文件的默认编码 -->* <**property name="defaultEncoding" value="UTF-8"**/>  </**bean**>  *<!-- 配置FreeMarker视图解析器 -->* <**bean class="org.springframework.web.servlet.view.freemarker.FreeMarkerViewResolver"**>  *<!-- 配置内容类型 -->* <**property name="contentType" value="text/html;charset=UTF-8"**/>  *<!-- 配置模板文件后缀名 -->* <**property name="suffix" value=".ftl"**/>  </**bean**>   *<!-- ############## 配置dubbo服务消费者 ############## -->  <!-- 配置当前应用的名称 -->* <**dubbo:application name="pinyougou-item-web"**/>  *<!-- 配置zookeeper作为注册中心，注册服务地址 -->* <**dubbo:registry protocol="zookeeper" address="192.168.12.131:2181"**/>  *<!-- 配置采用包扫描来引用服务，产生服务接口的代理对象 -->* <**dubbo:annotation package="com.pinyougou.item.controller"**/> </**beans**> |

# 商品详情页【基本数据显示】

## 3.1需求分析

运用Freemarker技术来实现商品详细页动态展示。通过地址栏输入某地址，如下形式

[http://item.pinyougou.com/149187842867997](http://localhost:9101/gen_item.do?goodsId=149187842867952).html

访问路径：<http://item.pinyougou.com/SPU商品goodsId.html>

## 3.2后台代码

### 3.2.1控制器层

pinyougou-item-web/src/main/java/com/pinyougou/item/controller/包下，创建ItemController.java：

|  |
| --- |
| @Controller **public class** ItemController {  @Reference(timeout=10000)  **private** GoodsService **goodsService**;  */\*\* 根据id查询商品信息 \*/* @GetMapping(**"/{goodsId}"**)  **public** String getGoods(@PathVariable(**"goodsId"**)Long goodsId,  Model model){  Map<String, Object> data = **goodsService**.getGoods(goodsId);  model.addAllAttributes(data);  **return "item"**;  } } |

### 3.2.2服务接口层

pinyougou-interface/src/main/java/com/pinyougou/service/GoodsService.java：

|  |
| --- |
| */\*\* 获取商品信息 \*/* Map<String,Object> getGoods(Long goodsId); |

### 3.2.3服务实现层

pinyougou-sellergoods-service/src/main/java/com/pinyougou/sellergoods/service/impl/GoodsServiceImpl.java：

|  |
| --- |
| */\*\* 获取商品信息 \*/* **public** Map<String, Object> getGoods(Long goodsId){  **try**{  */\*\* 定义数据模型 \*/* Map<String, Object> dataModel = **new** HashMap<>();  */\*\* 加载商品SPU数据 \*/* Goods goods = **goodsMapper**.selectByPrimaryKey(goodsId);  dataModel.put(**"goods"**, goods);  */\*\* 加载商品描述数据 \*/* GoodsDesc goodsDesc =**goodsDescMapper**.selectByPrimaryKey(goodsId);  dataModel.put(**"goodsDesc"**, goodsDesc);  **return** dataModel;  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } |

### 3.2.4测试

将item.html移至pinyougou-item-web/src/main/webapp/WEB-INF/ftl下，修改扩展名为ftl，将商品名称用取值代替(174行)：

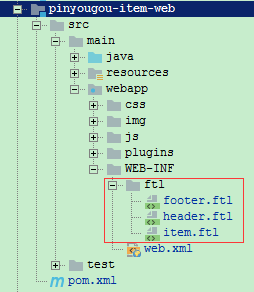
|  |
| --- |
| <**div class="sku-name"**>  <**h4**>**${goods**.**goodsName}**</**h4**> </**div**> |

**访问路径：http://item.pinyougou.com/149187842867997.html**

## 3.3前端代码

### 3.3.1模板模块化引入

此时我们的item.ftl内容较多，当我们编辑时不容易快速找到编辑的位置，所以我们将头部分拆分到header.ftl ,将尾部拆分到footer.ftl ,用include指令在item.ftl中引入。



在pinyougou-item-web/src/main/webapp/WEB-INF/ftl/item.ftl页面包含header.ftl与footer.ftl：

|  |
| --- |
| *<#-- 包含header.ftl 17行 -->* **<#include "header.ftl"/>**  **......**  *<#-- 包含footer.ftl 572行-->* **<#include "footer.ftl"/>** |

### 3.3.2生成简单数据

在模板中找到合适的位置，用取值替换静态文本item.ftl(67行)：

|  |
| --- |
| <**div class="sku-name"**>  <**h4**>**${goods**.**goodsName}**</**h4**> </**div**> <**div class="news"**><**span**>**${goods**.**caption}**</**span**></**div**> |

pinyougou-item-web/src/main/webapp/WEB-INF/ftl/item.ftl(77行)：

|  |
| --- |
| <**div class="fl price"**>  <**i**>¥</**i**><**em**>**${goods**.**price}**</**em**><**span**>降价通知</**span**> </**div**> |

pinyougou-item-web/src/main/webapp/WEB-INF/ftl/item.ftl(424行)：

|  |
| --- |
| *<#-- 商品详情 -->* <**div class="intro-detail"**>**${goodsDesc**.**introduction}**</**div**> |

pinyougou-item-web/src/main/webapp/WEB-INF/ftl/item.ftl(427行)：

|  |
| --- |
| <**div id="two" class="tab-pane"**>  <**p**>**${goodsDesc**.**packageList}**</**p**> </**div**> <**div id="three" class="tab-pane"**>  <**p**>**${goodsDesc**.**saleService}**</**p**> </**div**> |

运行控制层代码，测试效果：

[http://item.pinyougou.com/149187842867997](http://localhost:9101/gen_item.do?goodsId=149187842867952).html

### 3.3.3生成图片列表

编辑模板文件item.ftl(37行)：

|  |
| --- |
| *<#--图片列表 -->* **<#assign imageList**=**goodsDesc**.**itemImages**?**eval/>** |

说明：这一句要转换图片列表的json字符串

图片部分的代码：

|  |
| --- |
| *<!--放大镜效果-->* <**div class="zoom"**>  *<!--默认第一个预览-->* <**div id="preview" class="spec-preview"**>  <**span class="jqzoom"**>  **<#if** (**imageList**?**size** > 0)**>** <**img jqimg="${imageList**[0].**url}"  src="${imageList**[0].**url}"  width="400px" height="400px"**/>  **</#if>** </**span**>  </**div**>  *<!--下方的缩略图-->* <**div class="spec-scroll"**>  <**a class="prev"**>**&lt;**</**a**>  *<!--左右按钮-->* <**div class="items"**>  <**ul**>  **<#list imageList as item>** <**li**><**img src="${item**.**url}" bimg="${item**.**url}"  onmousemove="***preview*(**this**)**"**/></**li**>  **</#list>** </**ul**>  </**div**>  <**a class="next"**>**&gt;**</**a**>  </**div**> </**div**> |

生成效果如下：



### 3.3.4生成扩展属性列表

修改模板item.ftl，首先进行json转换(408行)：

|  |
| --- |
| *<#--扩展属性列表 -->* **<#assign customAttributeList**=**goodsDesc**.**customAttributeItems**?**eval/>** |

item.ftl显示扩展属性数据，如果扩展属性为空则不显示此条数据(412行)：

|  |
| --- |
| <**ul class="goods-intro unstyled"**>  **<#list customAttributeList as item>  <#if item**.**value**??**>** <**li**>**${item**.**text}**：**${item**.**value}**</**li**>  **</#if>  </#list>** </**ul**> |

### 3.3.5生成规格列表

修改模板item.ftl，转换规格列表(116行)：

|  |
| --- |
| *<#--规格选项列表 -->* **<#assign specificationList**=**goodsDesc**.**specificationItems**?**eval/>** |

此时，我们需要使用嵌套循环(120行)：

|  |
| --- |
| **<#list specificationList as s>** <**dl**>  <**dt**>  <**div class="fl title"**>  <**i**>**${s**.**attributeName}**</**i**>  </**div**>  </**dt**>  **<#list s**.**attributeValue as item>** <**dd**><**a href="javascript:;"**>**${item}**</**a**></**dd**>  **</#list>** </**dl**> **</#list>** |

## 3.4商品分类面包屑显示

修改item.ftl模板，展示商品分类面包屑(22行)：

|  |
| --- |
| <**ul class="sui-breadcrumb"**>  <**li**><**a href="#"**>**${itemCat1}**</**a**></**li**>  <**li**><**a href="#"**>**${itemCat2}**</**a**></**li**>  <**li**><**a href="#"**>**${itemCat3}**</**a**></**li**> </**ul**> |

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl/GoodsServiceImpl读取三级商品分类名称，加入到数据模型中：

|  |
| --- |
| */\*\* 获取商品信息 \*/* **public** Map<String,Object> getGoods(Long goodsId){  */\*\* 定义数据模型 \*/* Map<String,Object> dataModel = **new** HashMap<>();  */\*\* 加载商品SPU数据 \*/* Goods goods = **goodsMapper**.selectByPrimaryKey(goodsId);  dataModel.put(**"goods"**, goods);  */\*\* 加载商品描述数据 \*/* GoodsDesc goodsDesc = **goodsDescMapper**.selectByPrimaryKey(goodsId);  dataModel.put(**"goodsDesc"**, goodsDesc);  */\*\* 商品分类 \*/* **if** (goods != **null** && goods.getCategory3Id() != **null**){  String itemCat1 = **itemCatMapper** .selectByPrimaryKey(goods.getCategory1Id()).getName();  String itemCat2 = **itemCatMapper** .selectByPrimaryKey(goods.getCategory2Id()).getName();  String itemCat3 = **itemCatMapper** .selectByPrimaryKey(goods.getCategory3Id()).getName();  dataModel.put(**"itemCat1"**, itemCat1);  dataModel.put(**"itemCat2"**, itemCat2);  dataModel.put(**"itemCat3"**, itemCat3);  }  **return** dataModel; } |

# 4.商品详情页【动态效果】

## 4.1购买数量加减操作

### 4.1.1模板页面

pinyougou-item-web/src/main/webapp/WEB-INF/ftl/item.ftl中引入js：

|  |
| --- |
| <**script src="/js/vue.min.js"**></**script**> <**script src="/js/axios.min.js"**></**script**> <**script src="/js/controller/itemController.js"**></**script**> |

item.ftl调用操作数量的方法(135行)：

|  |
| --- |
| <**div class="controls"**>  <**input type="text" v-model="num" class="itxt"**/>  <**a href="javascript:void(0)" class="increment plus"  @click="addNum(1)"**>+</**a**>  <**a href="javascript:void(0)" class="increment mins"  @click="addNum(-1)"**>-</**a**> </**div**> |

### 4.1.2控制器层

pinyougou-item-web/src/main/webapp/js/controller/itemController.js，修改data与methods：

|  |
| --- |
| *// 窗口加载完* ***window***.onload = **function** () {  **var** vue = **new Vue**({  **el** : **'#app'**, *// 元素绑定* **data** : { *// 数据模型* **num : 1 *// 购买数量*** },  **methods** : { *// 操作方法  // 购买数量加减* addNum : **function** (x) {  **this**.**num** = *parseInt*(**this**.**num**);  **this**.**num** += x;  **if**(**this**.**num** < 1){  **this**.**num** = 1;  }  }  },  created : **function** () { *// 创建生命周期* }  }); }; |

说明：在方法中控制数量不能小于1

## 4.2规格选项选择

最终我们需要实现的效果：



### 4.2.1模板页面

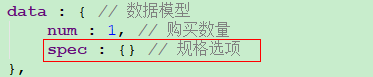
item.ftl页面调用控制器的方法(124行)：

|  |
| --- |
| **<#list s**.**attributeValue as item>** <**dd**>  <**a href="javascript:;"  :class="isSelected('${s**.**attributeName}','${item}')?'selected':''"  @click="selectSpec('${s**.**attributeName}','${item}');"**>  **${item}** <**span title="点击取消选择"**>**&nbsp;**</**span**>  </**a**>  </**dd**> **</#list>** |

### 4.2.2控制器层

pinyougou-item-web/src/main/webapp/js/controller/itemController.js：

* 在data数据模型中定义变量



* 在methods中定义方法

|  |
| --- |
| *// 记录用户选择的规格选项* selectSpec : **function** (name, value) {  *// 为json对象设置key-value* **Vue.set(this.spec, name, value);** }, *// 判断某个规格选项是否被选中* isSelected : **function** (name, value) {  **return this**.**spec**[name] == value; } |

# 5.商品详情页【读取SKU信息】

需求：当我们选择规格后，应该在页面上更新商品名称为SKU的商品标题，价格也应该为SKU的商品价格。

## 5.1后端代码

pinyougou-sellergoods-service/src/main/java/com/pinyougou/sellergoods/service/impl/GoodsServiceImpl.java：

|  |
| --- |
| */\*\* 获取商品信息 \*/* **public** Map<String,Object> getGoods(Long goodsId){  ......  */\*\* 商品分类 \*/* ......  */\*\* 查询SKU数据 \*/* Example example = **new** Example(Item.**class**);  */\*\* 查询条件对象 \*/*  Example.Criteria criteria = example.createCriteria();  */\*\* 状态码为：1 \*/* criteria.andEqualTo(**"status"**, **"1"**);  */\*\* 条件: SPU ID \*/* criteria.andEqualTo(**"goodsId"**, goodsId);  /\*\* 按是否默认降序(保证第一个为默认) \*/  example.orderBy("**isDefault**").desc();  /\*\* 根据条件查询SKU商品数据 \*/  List<Item> itemList = itemMapper.selectByExample(example);  dataModel.put("itemList", **JSON.toJSONString(itemList)**);  ...... } |

## 5.2前端代码

### 5.2.1页面生成SKU列表变量

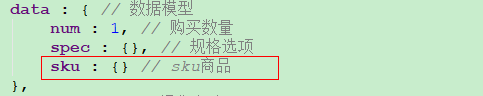
修改pinyougou-item-web/src/main/webapp/WEB-INF/ftl/item.ftl：

|  |
| --- |
| <**script type="text/javascript"**>  */\*\* 定义SKU商品数据 \*/* **var *itemList*** = **${itemList}**; </**script**> |

### 5.2.2显示SKU标题和价格

pinyougou-item-web/src/main/webapp/js/controller/itemController.js：

* 在data中定义变量



* 在methods中定义方法

|  |
| --- |
| *// 加载默认的SKU* loadSku : **function** () {  */\*\* 取第一个SKU \*/* **this**.**sku** = ***itemList***[0];  */\*\* 获取SKU商品选择的选项规格 \*/* **this**.**spec** = ***JSON***.parse(**this**.**sku**.**spec**); } |

* 在created中调用方法

|  |
| --- |
| created : **function** () { *// 创建生命周期  // 加臷sku* **this**.loadSku(); } |

修改item.ftl，显示标题(71行)：

|  |
| --- |
| <**div class="sku-name"**><**h4**>{{**sku**.**title**}}</**h4**></**div**> |

item.ftl显示价格(81行)：

|  |
| --- |
| <**div class="fl price"**>  <**i**>¥</**i**><**em**>{{**sku**.**price.toFixed(2)**}}</**em**><**span**>降价通知</**span**></**div**> |

### 5.2.3选择规格更新SKU

pinyougou-item-web/src/main/webapp/js/controller/itemController.js,用户选择规格选项后触发查询对应的SKU方法：

|  |
| --- |
| *// 记录用户选择的规格选项* selectSpec : **function** (name, value) {  *// 为json对象设置key-value* **Vue**.set(**this**.**spec**, name, value);  *// 查找对应的SKU商品* **this**.searchSku(); }, |

pinyougou-item-web/src/main/webapp/js/controller/itemController.js, 查询当前用户选择的SKU方法：

|  |
| --- |
| *// 根据用户选中的规格选项，查找对应SKU* searchSku : **function** () {  **for** (**var** i = 0; i < ***itemList***.**length**; i++){  */\*\* 判断规格选项是不是当前用户选中的 \*/* **if**(***itemList***[i].**spec** == ***JSON***.stringify(**this**.**spec**)){  **this**.**sku** = ***itemList***[i];  **return**;  }  } } |

### 5.2.4加入购物车事件绑定

修改模板item.ftl(156行)：

|  |
| --- |
| <**li**><**a href="javascript:;" class="sui-btn btn-danger addshopcar"  @click="addToCart();"**>加入购物车</**a**></**li**> |

pinyougou-item-web/src/main/webapp/js/controller/itemController.js，在methods中添加方法：

|  |
| --- |
| *// 加入购物车事件绑定* addToCart : **function** () {  *alert*(**'sku商品id:'** + **this**.**sku**.**id** + **", 购买数量："** + **this**.**num**); } |

# 6.系统模块对接

pinyougou-search-web/src/main/webapp/index.html(326行)

|  |
| --- |
| <**a :href="'http://item.pinyougou.com/'+ item.goodsId +'.html'"**>  <**img :src="item.image"**/> </**a**> |