**新巴巴运动网 项目第八天**

**传智.上官云**

# 课前回顾

1. 前台商品检索之过滤条件查询 （样式 ： 点击品牌 品牌消失 已选品牌出现）
2. 查询结果 品牌或价格传递到后台 （

SolrQuery.addFitterQuery(“brandId:” + brandId) (price:[开始价 TO 结束价)

1. 消息队列ActiveMQ 原理 应用场景
2. ActiveMQ 消息生产方 消息中间件/web浏览器访问8161 Java 61616 消息消费方

# 今天内容

1. 去商品详情页面（加载）
2. SKu联动
3. 静态化Freemarker环境搭建
4. 基本语法（三个） jstl el if foreach
5. 静态化容入项目中
6. 去登陆页面
7. Session共享问题

# 去商品详情页面

商品表

库存表（包括颜色表）

## 设置去商品详情页面的路径



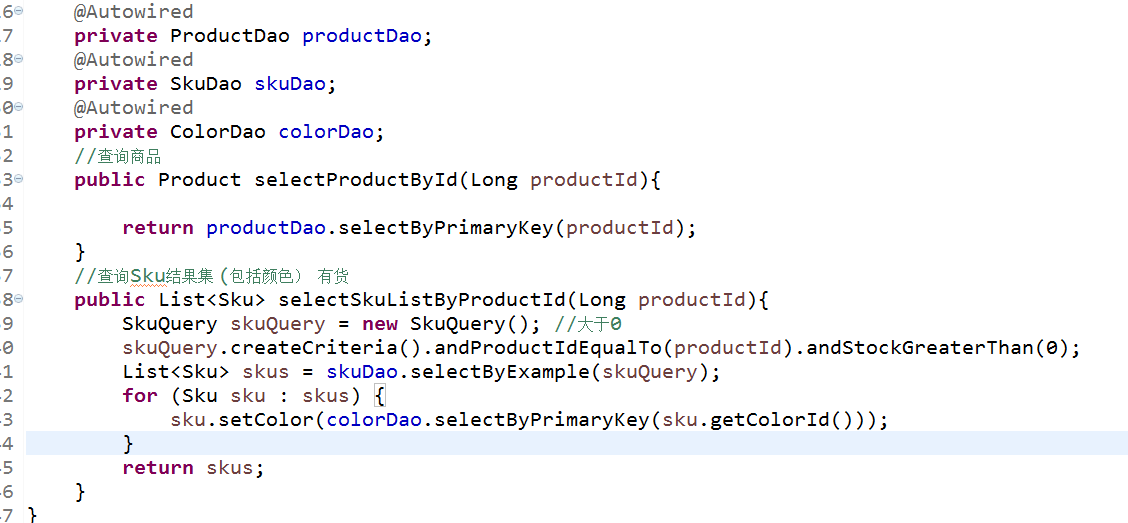
## CmsService

入参： productId 商品ID

一个商品对象

入参： productId 商品ID

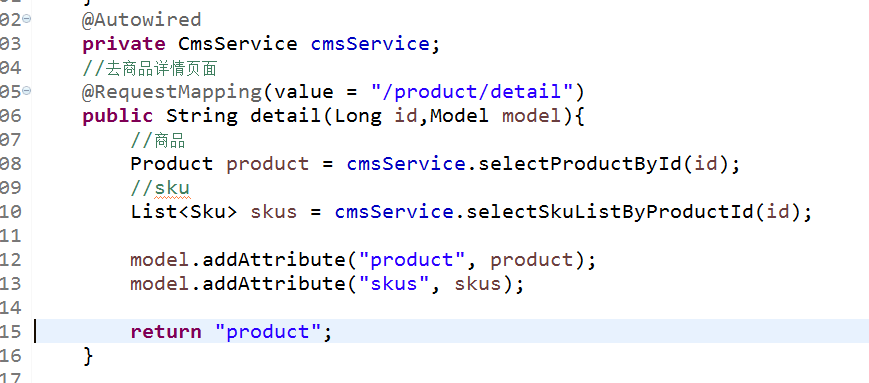
返回值：sku结果集



## Controller接收

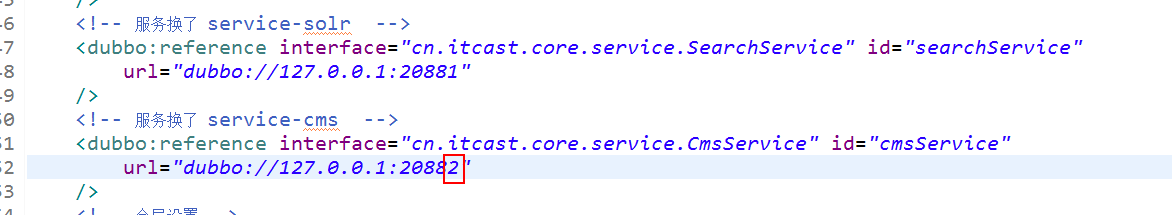
入参： productId 商品ID

返回值：商品对象（一个） Sku对象（多个）



## Dubbo

略

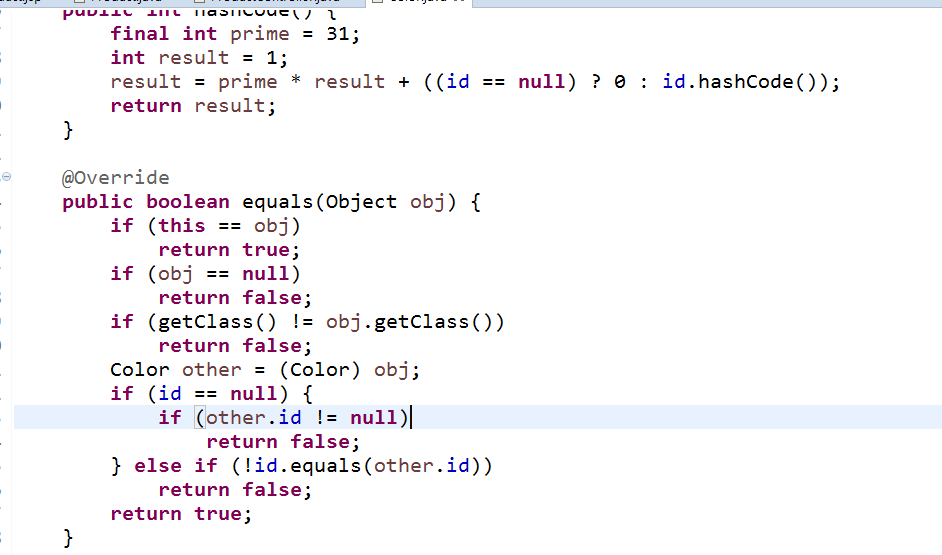


## 页面

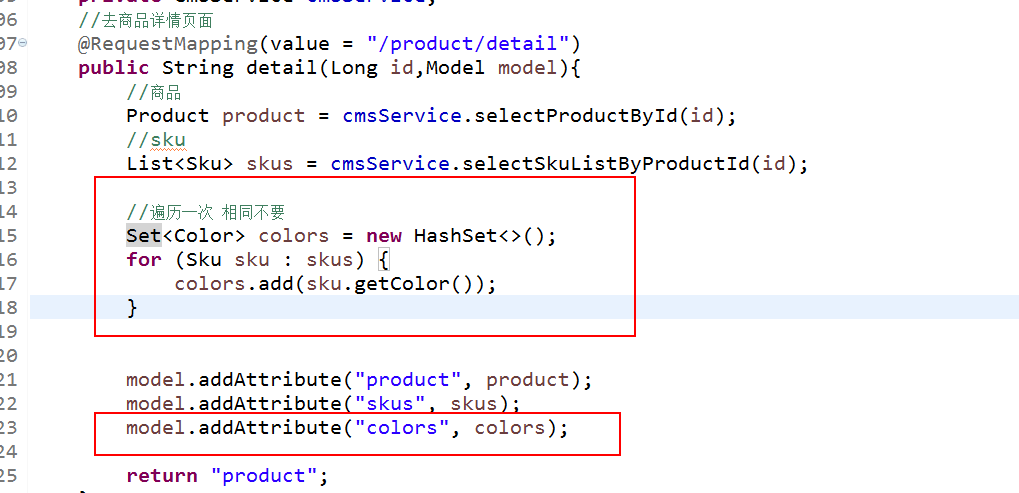
回显

# Sku联动

Color对象hashCode equals



Controller



## 点击颜色

遍历目前还有的尺码



### Function

页面保存结果集 input type=hidden value=”1”

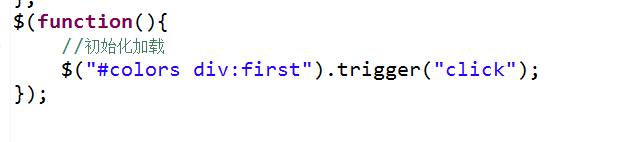
|  |
| --- |
| //SKUID　全局变量  var skuId;  //点击颜色 id == 颜色ID  function colorToRed(target,id){  //清理成白色  // $("#colors div").attr("class","item");  $("#colors div").removeClass("selected");  //点击的颜色变成红边框  // $(target).attr("class","item selected");  $(target).addClass("selected");    //id ajax  var flag = 0;  var html = "";  <c:forEach items="${skus }" var="sku">  if(id == '${sku.colorId}'){  if(flag == 0){  html += '<div class="item selected" id="${sku.size}" onclick="sizeToRed(this,\'${sku.size}\')">'  + '<b></b><a href="javascript:;" title="${sku.size}" >${sku.size}</a>'  + '</div>';  flag = 1;    //颜色及尺码 都成红色  //设置巴巴价  $("#bbs-price").html('${sku.price}');  //保存一个SKuＩＤ  skuId = '${sku.id}';  }else{    html += '<div class="item" id="${sku.size}" onclick="sizeToRed(this,\'${sku.size}\')">'  + '<b></b><a href="javascript:;" title="${sku.size}" >${sku.size}</a>'  + '</div>';  }  }  </c:forEach>  //替换  $("#sizes").html(html);  } |

## 点击尺码

|  |
| --- |
| //点击尺码 id == 尺码 （S。M。。)  function sizeToRed(target,id){    //清理成白色  // $("#sizes div").attr("class","item");  $("#sizes div").removeClass("selected");  //点击的颜色变成红边框  // $(target).attr("class","item selected");  $(target).addClass("selected");    //价格  <c:forEach items="${skus }" var="sku">  if(colorId == '${sku.colorId}' && id == '${sku.size}'){  //颜色及尺码 都成红色  //设置巴巴价  $("#bbs-price").html('${sku.price}');  //保存一个SKuＩＤ  skuId = '${sku.id}';  }  </c:forEach>      } |

## 页面初始化加载触发器

帮我点击第一个颜色



# 静态化环境搭建Freemarker

二种：

Freemarker 第三方 网站 经常打不开 Freemarker 11年以上 宏定义Velocity不具备

Velocity 出身 Apache公司 16年

Freemarker（Java写的）

${world} == el #world#

If

Foreach

## 静态化技术Freemarker的基本语法

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](http://baike.baidu.com/view/9472.htm)。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。

目前企业中:主要用Freemarker做静态页面或是页面展示

## 使用Freemarker所用到的jar

把下载到的jar包（freemarker-2.3.16.jar）放到\WebRoot\WEB-INF\lib目录下。下载地址：http://freemarker.sourceforge.net/freemarkerdownload.html （注：官方网站.org的经常打不开）

如果使用的是Maven结构,可在pom.xml中引入以下坐标

<dependency>

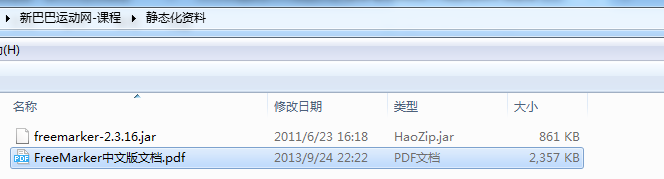
<groupId>freemarker</groupId>

<artifactId>freemarker</artifactId>

<version>2.3.18</version>

</dependency>

## Freemarker原理

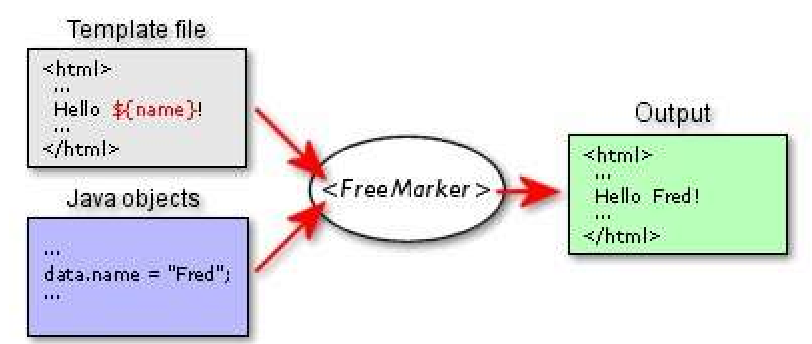


2.3.16

Maven坐标

2.3.18

## Freemarker原理图



模板 + 数据模型 = 输出

## 环境搭建

## 标签

El ${world}

遍历

## 商品详情页面静态化

### 本项目中搭建Freemarker运行环境 Springmvc进行搭建

Springmvc

服务cn.itcast.core.service.staticpage

StaticPageServiceImpl 静态化的程序

|  |
| --- |
| //静态化 商品 ActiveMQ  public void productStaticPage(Map<String,Object> root,String id){  //输出的路径 全路径  String path = getPath("/html/product/" + id + ".html");  File f = new File(path);  File parentFile = f.getParentFile();  if(!parentFile.exists()){  parentFile.mkdirs();  }  Writer out = null;  try {  //读文件 UTF-8  Template template = conf.getTemplate("product.html");  //输出 UTF-8  out = new OutputStreamWriter(new FileOutputStream(f), "UTF-8");  //处理  template.process(root, out);    } catch (Exception e) {  // TODO Auto-generated catch block  e.printStackTrace();  }finally {  if(null != out){  try {  out.close();  } catch (IOException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }  }  }  } |

### Product.jsp

改成静态化页面 prouduct.html

Jstl库就没有 了 c标签 不用再用了

用Freemarker 直接就能用 html

模板书写

### 上架

调用静态化

传递商品ID

ActiveMQ 主题

上架 service-product 发出消息 productId 点对点 –》发布、订阅

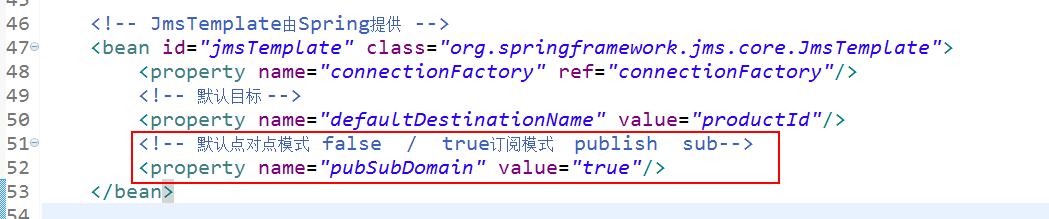
检索 service-solr 接收消息 productId

静态化 service-cms 接收消息 productId

ActiveMQ

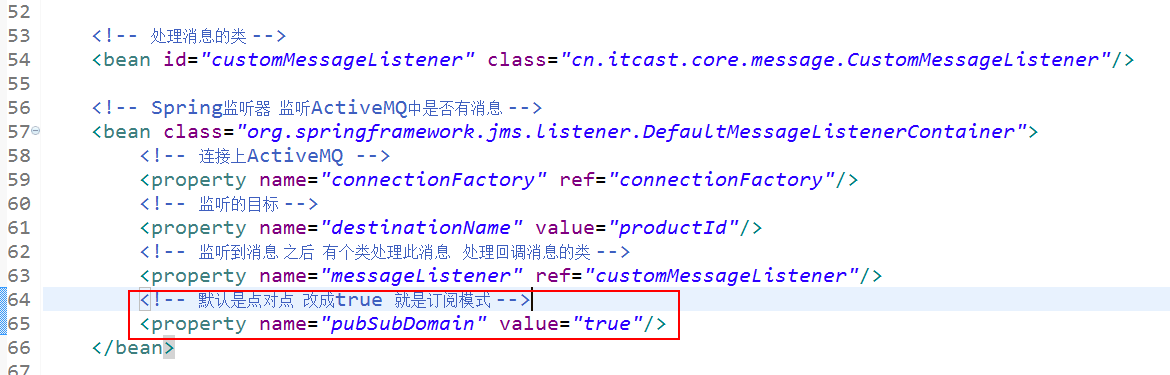
名称productId 主题/ 发布、订阅 模式

发送消息 时 product



、

接收消息 solr



接收消息 service –cms

同上

### Cms中的消息接收处理器

|  |
| --- |
| @Autowired  private StaticPageService staticPageService;  @Autowired  private CmsService cmsService;  @Override  public void onMessage(Message message) {  // TODO Auto-generated method stub  ActiveMQTextMessage am = (ActiveMQTextMessage)message;  try {  System.out.println("ActiveMq中的消息是-cms:" + am.getText());  String id = am.getText();  //数据  Map<String,Object> root = new HashMap<>();  //商品  Product product = cmsService.selectProductById(Long.parseLong(id));  //sku  List<Sku> skus = cmsService.selectSkuListByProductId(Long.parseLong(id));    //遍历一次 相同不要  Set<Color> colors = new HashSet<>();  for (Sku sku : skus) {  colors.add(sku.getColor());  }  root.put("product", product);  root.put("skus", skus);  root.put("colors", colors);  //静态化  staticPageService.productStaticPage(root, id);    } catch (JMSException e) {  // TODO Auto-generated catch block  e.printStackTrace();  }    } |

### Product.html

参考源码中

### 检索页面进入静态化后的页面

