**新巴巴运动网 项目第六天**

**传智.上官云**

# 课前回顾

1. 富文本编辑器 上传图片 1）kingEditor/..js /lang/zh\_CN.js 2)kindEdtor.create(“id”) 3)设置上传路径 4）HttpServletRequest （MultiparRequest） .getFileMap() 5)遍历Map

6）Value 就是图片 7）回显 put(“error”,0) put (url,url)

2、保存商品信息（对字段、Product、保存商品信息返回ID、遍历颜色（遍历尺码）保存SKU）

3、 库存管理 去库存管理页面（通过商品ID查询SKU结果集）３个颜色　５个尺码

4、 修改（文件框可写状态） 传递SKUID （市场mid、售价pid、库存sid、运费、限制）

5、保存 改不可写 获取对应的Value 修改到Mysql数据库 （json字符串 接收转Sku对象）xml (报文)

6、 上架 （商品状态更改） 保存信息到Solr服务器

7、 前台系统搭建

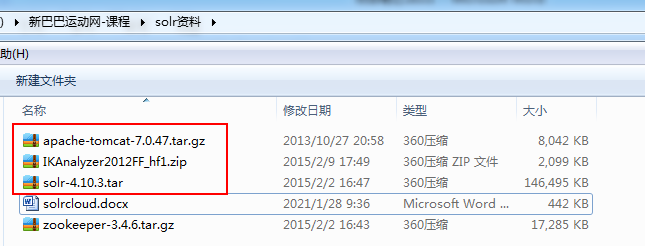
# 今天内容

主打：SOLR服务器的搭建 （单机版）

1. Solr服务器（单机版） IK分词器 自定义数据类型 自定义字段 高亮 扩展 停止 关键词 过滤条件 分页 排序
2. Java接口操作SOLR服务器 solrJ
3. 保存商品信息到SOLR （上架TODO） service-product 中完成
4. 前台首页 搜索（查看效果）

# Solr服务器（单机版）

准备位置



## 上传

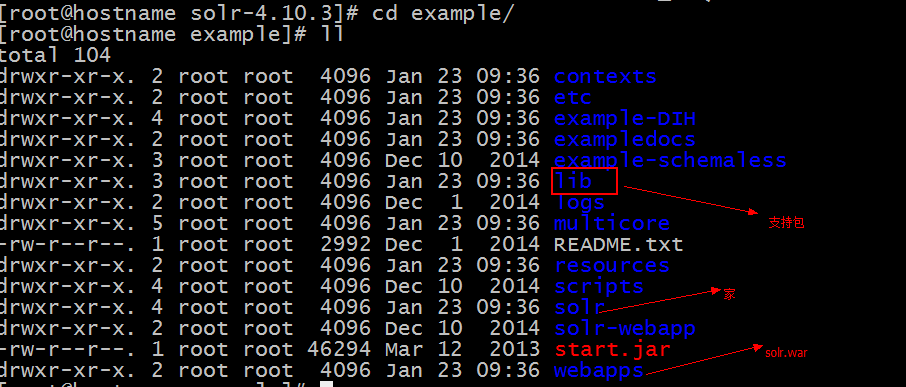
## 解压



解压IK



## 目录结构 （搭建过程）

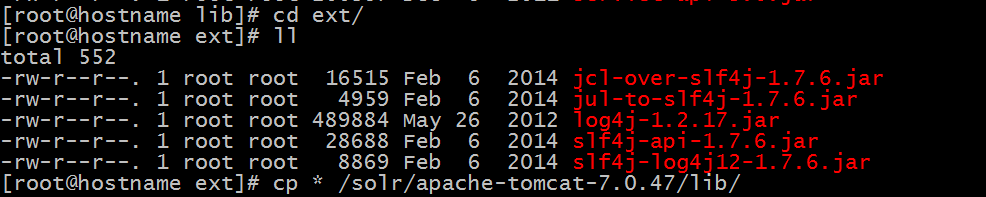


## 三步：搭建成功

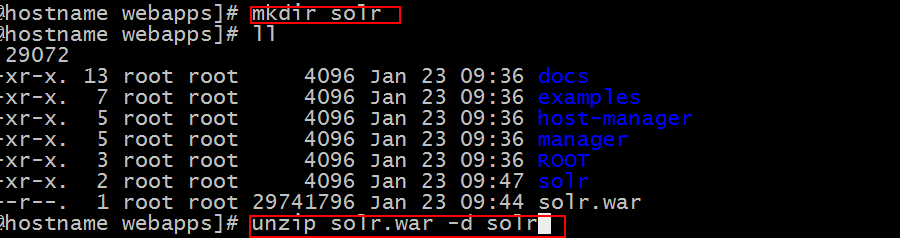
### 第一步：复制solr.war到Tomcat/webapp下

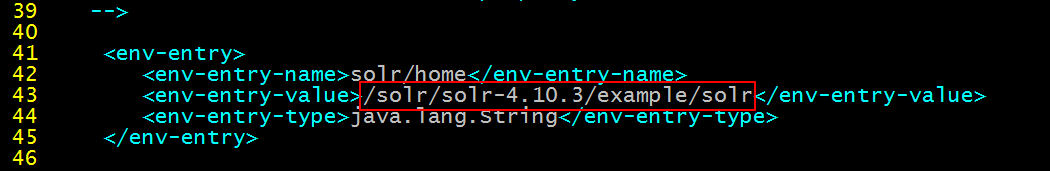


### 第二步：复制支持包到tomcat/lib下

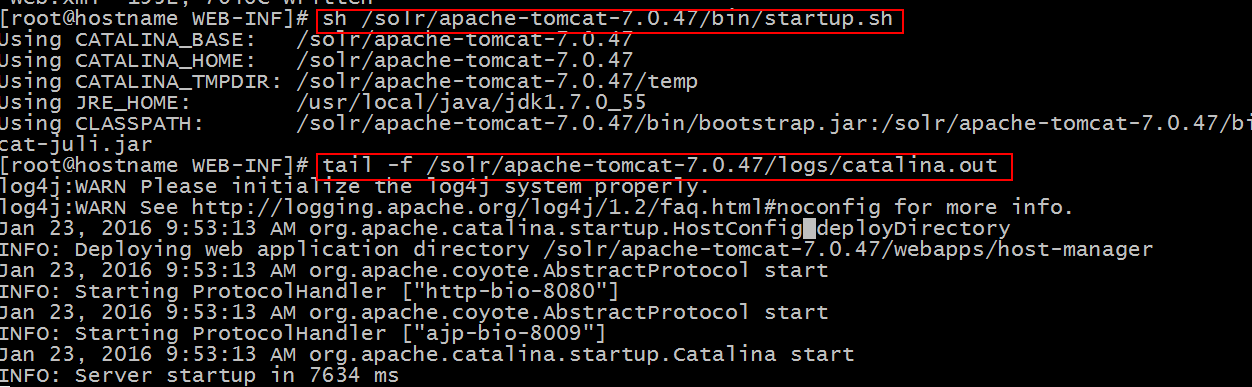


### 第三步：指定家的位置web.xml(solr/WEB-INF/web.xml)





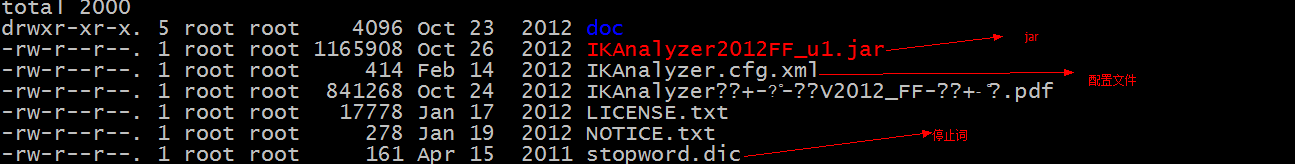
### 启动Tomcat



# 配置IK分词器

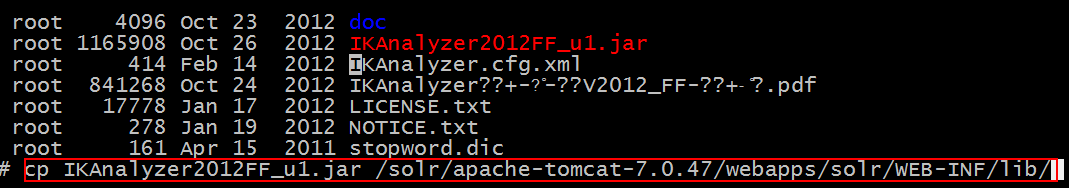
## IK配置过程

第一步：查看目录

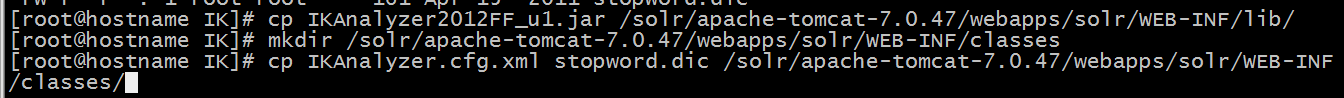


搭建IK

第一步：复制jar 到tomcat/webapp/solr/WEB-INF/lib



第二步：复制配置文件.cfg.xml stopword.dic 到tomcat/webapp/solr/WEB-INF/classes下

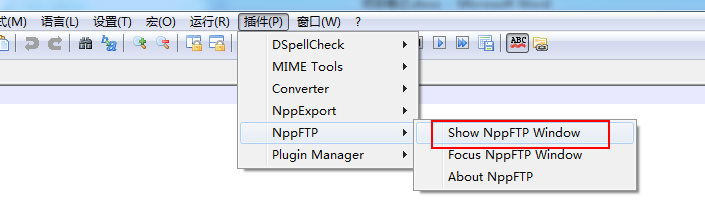


第三步：自定义字段所属类型 之前 先自定义类型（类型支持IK分词）

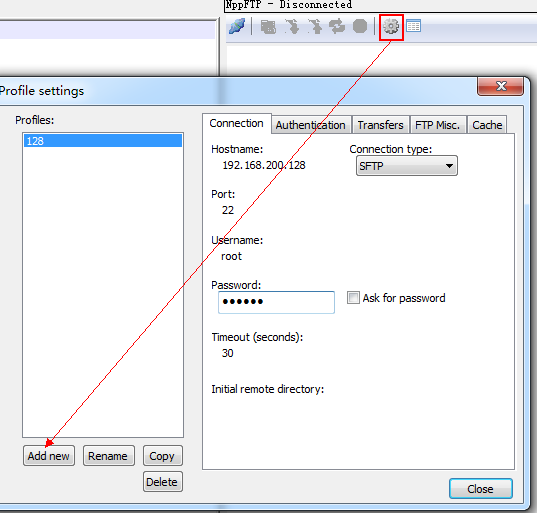
配置schema.xml

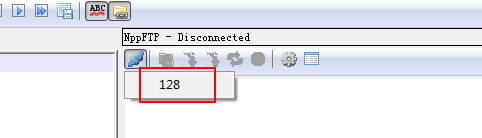
小技巧：

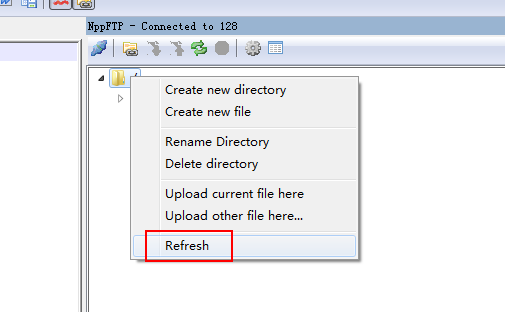
一步：



二步：

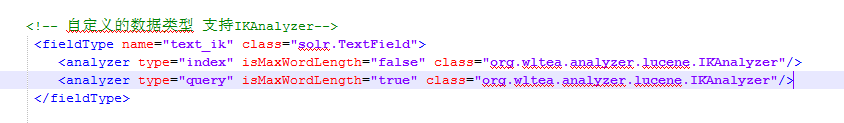






## 配置自定义的类型

|  |
| --- |
| <!-- 自定义的数据类型 支持IKAnalyzer-->  <fieldType name="text\_ik" class="solr.TextField">  <analyzer type="index" isMaxWordLength="false" class="org.wltea.analyzer.lucene.IKAnalyzer"/>  <analyzer type="query" isMaxWordLength="true" class="org.wltea.analyzer.lucene.IKAnalyzer"/>  </fieldType> |

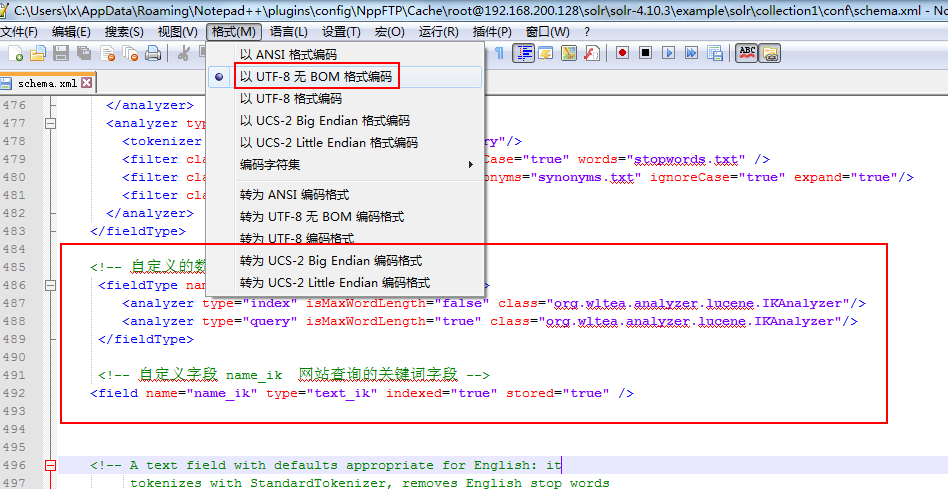


## 配置自定义字段

<!-- 自定义字段 name\_ik 网站查询的关键词字段 -->

<field name="name\_ik" type="text\_ik" indexed="true" stored="true" />

## 注意事项

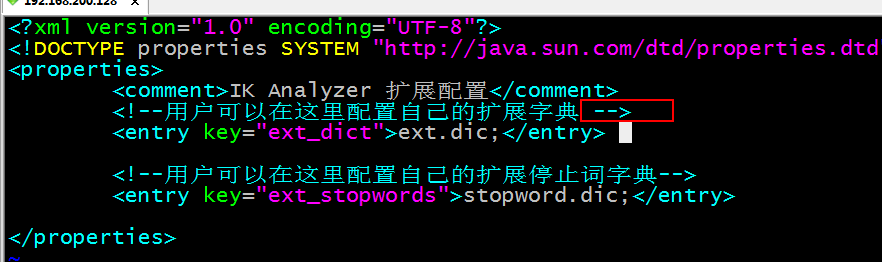


第四步：重启Tomcat生效

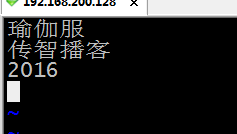
# 配置扩展词

瑜伽服

准备工作

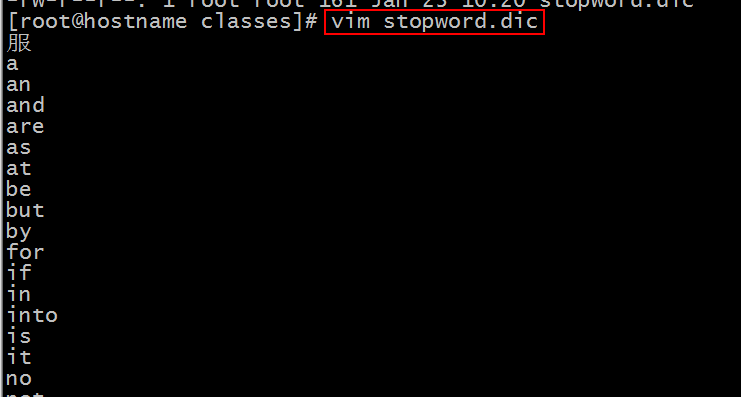


创建ext.dic



# 配置停止词

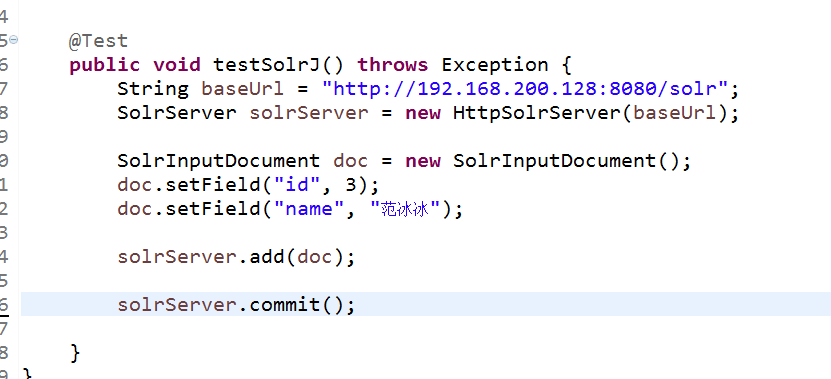
服



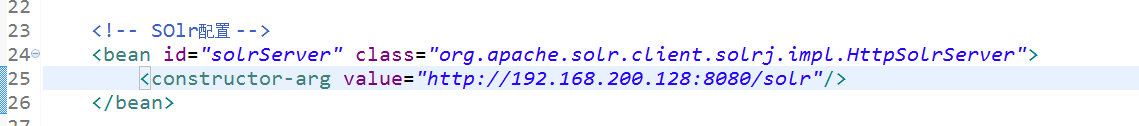
## 再重启Tomcat

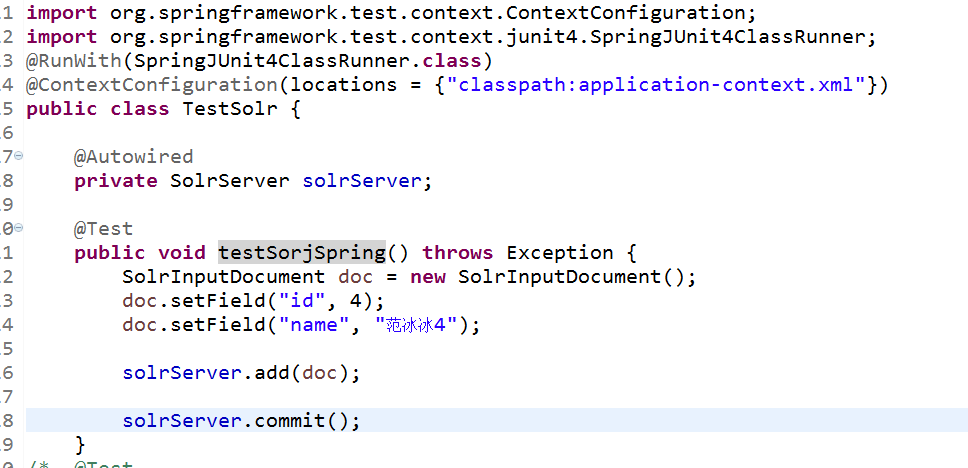


# Java接口操作Solr服务器



## 整合到Spring中

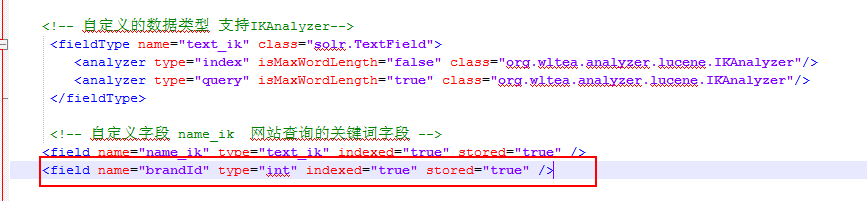




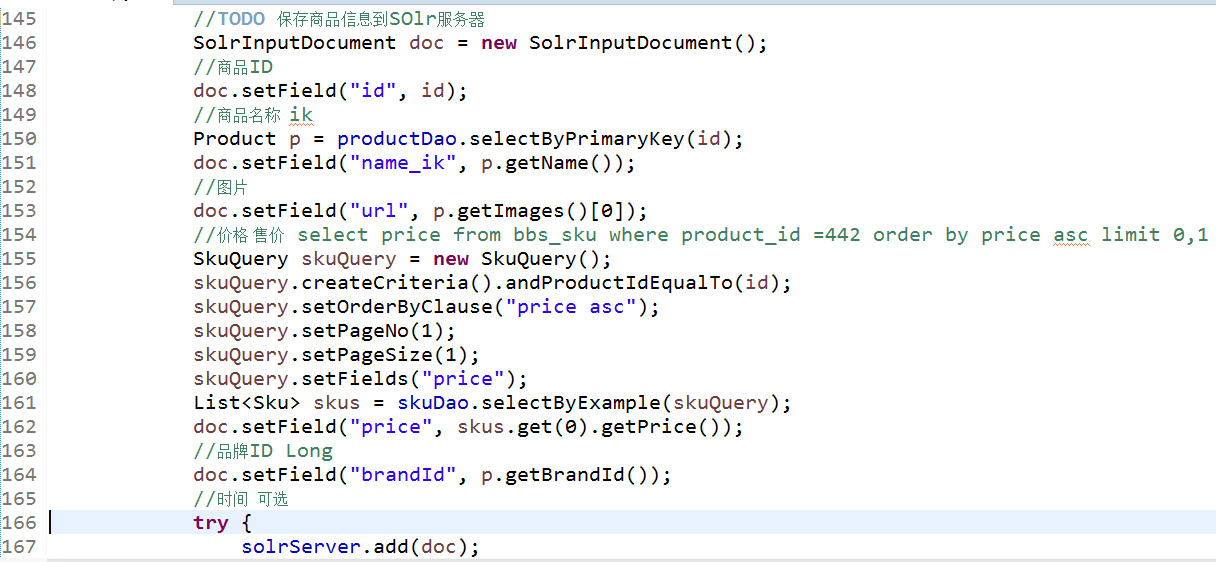
# 上架

保存商品信息到SOlr服务器

## 创建新的自定义字段



## Service

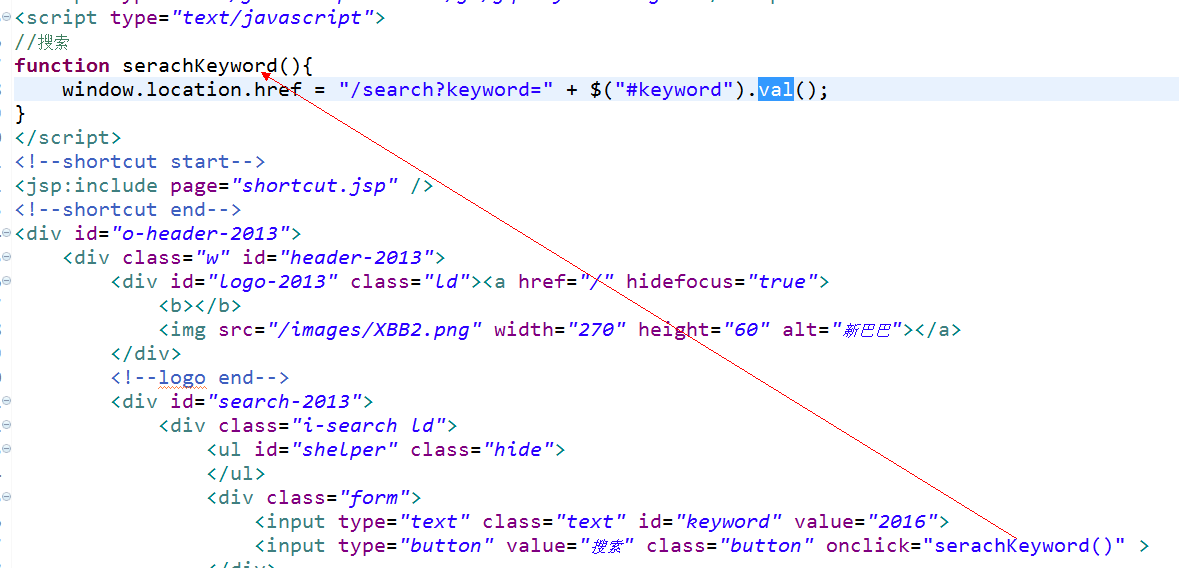


# 前台系统

首页 搜索 关键词

## 设置搜索按钮

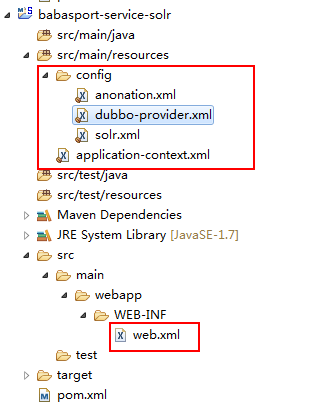
/search?keyword=2016



## Service-solr搭建

Service-product 20880

本次 20881





## Service

入参：keyword 当前页

返回值 List<Product>

查询的SOlr数据库

分页 ：

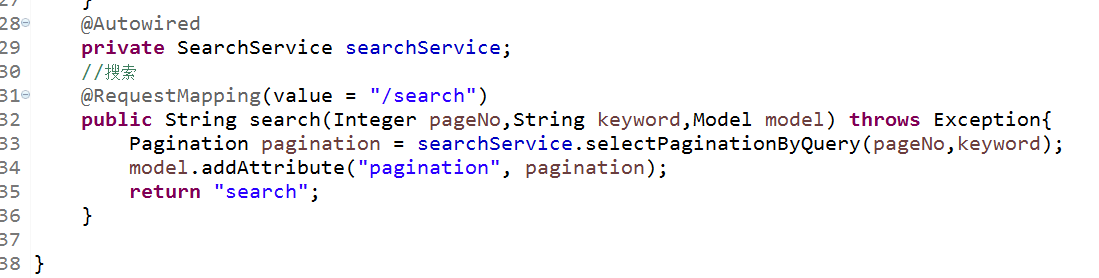
排序 价格 低到高排序

|  |
| --- |
| /\*\*  \* 全文检索 用的SOlr  \* @author lx  \*  \*/  @Service("searchService")  public class SearchServiceImpl implements SearchService {  @Autowired  private SolrServer solrServer;    //全文检索  public Pagination selectPaginationByQuery(Integer pageNo,String keyword) throws Exception{  //创建包装类  ProductQuery productQuery = new ProductQuery();  //当前页  productQuery.setPageNo(Pagination.cpn(pageNo));  //每页显示 12条  productQuery.setPageSize(12);    //拼接条件  StringBuilder params = new StringBuilder();    List<Product> products = new ArrayList<Product>();  SolrQuery solrQuery = new SolrQuery();  //关键词  solrQuery.set("q", "name\_ik:" + keyword);  params.append("keyword=").append(keyword);  //过滤条件  //高亮  solrQuery.setHighlight(true);  solrQuery.addHighlightField("name\_ik");  // 样式 <span style='color:red'>2016</span>  solrQuery.setHighlightSimplePre("<span style='color:red'>");  solrQuery.setHighlightSimplePost("</span>");  //排序  solrQuery.addSort("price", ORDER.asc);  //分页 limit 开始行 ， 每页显示条数  solrQuery.setStart(productQuery.getStartRow());  solrQuery.setRows(productQuery.getPageSize());  //执行查询  QueryResponse response = solrServer.query(solrQuery);    //取高亮  Map<String, Map<String, List<String>>> highlighting = response.getHighlighting();  // Map K : V 442 : Map  // Map K : V name\_ik : List<String>  // List<String> list 2016年最新款191期卖的瑜伽服最新限量纯手工制作细心打造商户经典买一送一清 list.get(0);  //结果集  SolrDocumentList docs = response.getResults();  //发现的条数 （总条件）构建分页时用到  long numFound = docs.getNumFound();  for (SolrDocument doc : docs) {  //创建商品对象  Product product = new Product();  //商品ID  String id = (String) doc.get("id");  product.setId(Long.parseLong(id));  //商品名称 ik  Map<String, List<String>> map = highlighting.get(id);  List<String> list = map.get("name\_ik");  product.setName(list.get(0));  /\* String name = (String) doc.get("name\_ik");  product.setName(name);\*/  //图片  String url = (String) doc.get("url");  product.setImgUrl(url);//img,img2,img3  //价格 售价 select price from bbs\_sku where product\_id =442 order by price asc limit 0,1  Float price = (Float) doc.get("price");  product.setPrice(price);  //品牌ID Long  Integer brandId = (Integer) doc.get("brandId");  product.setBrandId(Long.parseLong(String.valueOf(brandId)));    products.add(product);  }  //构建分页对象  Pagination pagination = new Pagination(  productQuery.getPageNo(),  productQuery.getPageSize(),  (int)numFound,  products  );  //页面展示  String url = "/search";  pagination.pageView(url, params.toString());    return pagination;  }  } |

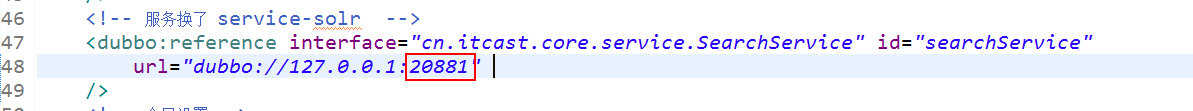
## Controller（接收）

入参：Keyword

返回值：结果集



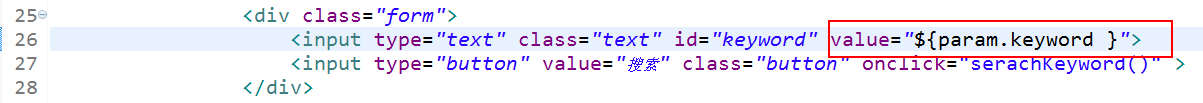
## Dubbo配置 消费方



## 页面

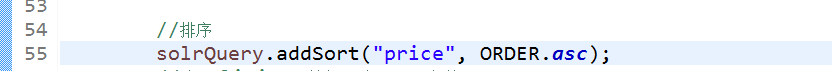
遍历商品结果集

## 回显关键词



## 排序

价格低到高 23元（第一位置



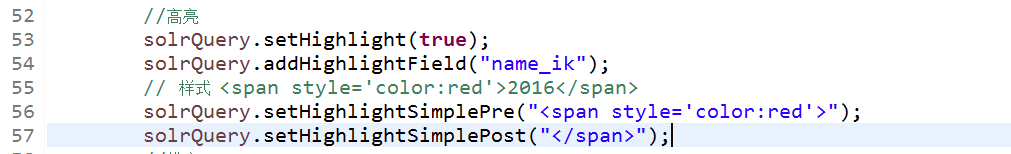
## 高亮

第一步：设置高亮

小步：设置高亮开关

小步：设置高亮的字段

小步：设置高亮的简单的样式



第二步：取高亮

Map<String,Map<String,List<String>>> map

