**品优购电商系统开发**

**第7章**

**商品管理【2】**

传智播客.黑马程序员

# 课程目标

目标1：完成规格选择的功能

目标2：完成录入SKU商品信息的功能

目标3：完成是否启动规格的功能

目标4：完成商品列表功能

# 商品录入【规格选择】



## 1.1需求分析

显示规格及选项列表（复选框）如下图，并保存用户选择的结果



## 1.2显示规格选项列表

由于我们的模板中只记录了规格名称，而我们除了显示规格名称还要显示规格所对应的规格选项，所以我们需要在后端查询规格选项数据。

### 1.2.1前端代码

pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js：

* 在数据模型data中定义变量



* 修改watch中监控类型模板id

|  |
| --- |
| *// 监控类型模板id发生改变，获取类型模板对象* **'goods.typeTemplateId'** : **function** (newVal, oldVal) {  **this**.**goods**.**brandId** = **""**;  **if** (newVal){ *// 不是空字符串* **axios**.get(**"/typeTemplate/findOne?id="** + newVal)  .then(**function**(response){  */\*\* 获取模板中的品牌列表 \*/* vue.**brandList** = ***JSON***.parse(response.**data**.brandIds);  */\*\* 设置扩展属性 \*/* vue.**goods**.**goodsDesc**.**customAttributeItems** =  ***JSON***.parse(response.**data**.**customAttributeItems**);  });   */\*\* 查询模板对应的规格与规格选项 \*/* **axios.get("/typeTemplate/findSpecByTemplateId?id=" + newVal)  .then(function(response){  vue.specList = response.data;  });**  }**else**{ *// 清空数据* **this**.**brandList** = [];  **this**.**goods**.**goodsDesc**.**customAttributeItems** = [];  **this**.**specList** = [];  } } |

pinyougou-shop-web/src/main/webapp/admin/goods\_edit.html页面209行：

|  |
| --- |
| <**div class="row data-type"**>  <**div v-for="s in specList"**>  <**div class="col-md-2 title"**>{{***s***.**text**}}</**div**>  <**div class="col-md-10 data"**>  <**span v-for="o in s.options"**>  <**input type="checkbox"**>{{***o***.optionName}}  </**span**>  </**div**>  </**div**> </**div**> |

### 1.2.2后端代码

pinyougou-shop-web/src/main/java/com/pinyougou/shop/controller/TypeTemplateController.java新增方法：

|  |
| --- |
| */\*\* 根据模板id查询所有的规格与规格选项 \*/* @GetMapping(**"/findSpecByTemplateId"**) **public** List<Map> findSpecByTemplateId(Long id){  **return typeTemplateService**.findSpecByTemplateId(id); } |

pinyougou-interface/src/main/java/com.pinyougou.service/TypeTemplateService.java新增方法：

|  |
| --- |
| */\*\* 根据模板id查询所有的规格与规格选项 \*/* List<Map> findSpecByTemplateId(Long id); |

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl/TypeTemplateServiceImpl.java新增方法：

|  |
| --- |
| @Autowired **private** SpecificationOptionMapper **specificationOptionMapper**;  */\*\* 根据模板id查询所有的规格与规格选项 \*/*  @Override**public** List<Map> findSpecByTemplateId(Long id){  **try**{  */\*\* 根据主键id查询类型模板 \*/* TypeTemplate typeTemplate = findOne(id);  */\*\*  \* [{"id":33,"text":"电视屏幕尺寸"}]  \* 获取模板中所有的规格，转化成 List<Map>  \*/* List<Map> specLists =  JSON.*parseArray*(typeTemplate.getSpecIds(), Map.**class**);  */\*\* 迭代模板中所有的规格 \*/* **for** (Map map : specLists){  */\*\* 创建查询条件对象 \*/* SpecificationOption so = **new** SpecificationOption();  so.setSpecId(Long.*valueOf*(map.get(**"id"**).toString()));  */\*\* 通过规格id，查询规格选项数据 \*/* List<SpecificationOption> specOptions =  **specificationOptionMapper**.select(so);  map.put(**"options"**, specOptions);  }  **return** specLists;  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |

**请求地址(测试)：**

**http://shop.pinyougou.com/typeTemplate/findSpecByTemplateId?id=35**



## 1.3保存选中规格选项

**实现思路分析：**

我们需要将用户选中的选项保存在**tb\_goods\_desc**表的**specification\_items**列中，需定义json数组封装数据，格式如下：

|  |
| --- |
| [{“attributeName”:”规格名称”,”attributeValue”:[“规格选项1”,“规格选项2”....]},....] |

### 1.3.1步骤一

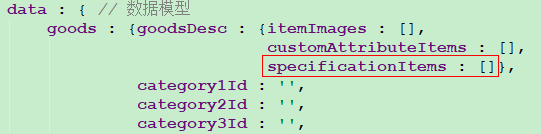
pinyougou-shop-web/src/main/webapp/admin/goods\_edit.html为用户选择的checkbox绑定点击事件(215行)：

|  |
| --- |
| <**div class="row data-type"**>  <**div v-for="s in specList"**>  <**div class="col-md-2 title"**>{{***s***.**text**}}</**div**>  <**div class="col-md-10 data"**>  <**span v-for="o in s.options"**>  <**input type="checkbox"  @click="selectSpecAttr($event,s.text,o.optionName)"**/>  {{***o***.optionName}}  </**span**>  </**div**>  </**div**> </**div**> |

### 1.3.2步骤二

pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js：

* 在数据模型data中定义变量



* 在methods中定义方法

|  |
| --- |
| *// 用户选择规格选项* **selectSpecAttr** : **function** (e, specName, optionName) {  */\*\* 根据json对象的key到json数组中搜索该key值对应的对象 \*/* **var** obj = **this**.searchJsonByKey(**this**.**goods**.**goodsDesc** .**specificationItems**,**'attributeName'**, specName);  */\*\* 判断对象是否为空 \*/* **if**(obj){  */\*\* 判断checkbox是否选中 \*/* **if**(e.**target**.**checked**){  */\*\* 添加该规格选项到数组中 \*/* obj.**attributeValue**.push(optionName);  }**else**{  */\*\* 取消勾选，从数组中删除该规格选项 \*/* obj.**attributeValue**.splice(obj.**attributeValue** .indexOf(optionName), 1);  */\*\* 如果选项都取消了，将此条记录删除 \*/* **if**(obj.**attributeValue**.length == 0){  **this**.**goods**.**goodsDesc**.**specificationItems**.splice(  **this**.**goods**.**goodsDesc** .**specificationItems**.indexOf(obj),1);  }  }  }**else**{  */\*\* 如果为空，则新增数组元素 \*/* **this**.**goods**.**goodsDesc**.**specificationItems**.push(  {**"attributeName"**:specName,**"attributeValue"**:[optionName]});  } }, */\*\* 从json数组中根据key查询指定的json对象 \*/* **searchJsonByKey** : **function**(jsonArr, key, value){  */\*\* 迭代json数组 \*/* **for**(**var** i = 0; i < jsonArr.**length**; i++){  **if**(jsonArr[i][key] == value){  **return** jsonArr[i];  }  } } |

为了方便测试，我们可以在页面上某个区域临时添加取值表达式，以便观测测试结果：

|  |
| --- |
| {{**goods**.**goodsDesc**.**specificationItems**}} |

# 2.商品录入【SKU商品信息】

## 2.1需求分析

基于上一步我们完成的规格选择，根据选择的规格录入商品的SKU信息，当用户选择相应的规格，下面的SKU列表就会自动生成，如下图：

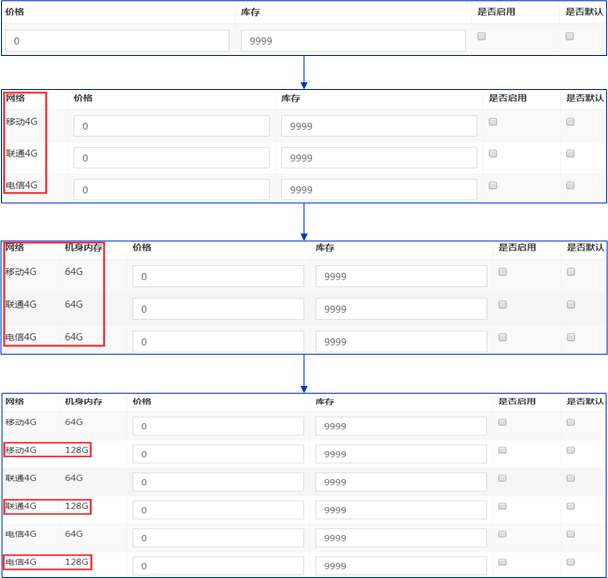


**实现思路：**

1. 我们先定义一个初始的不带规格名称的json数组，只有一条记录。

*/\*\* 定义SKU数组变量，并初始化 \*/***this.goods**.**items** = [{**spec**:{}, **price**:0, **num**:9999,  
 **status**:**'0'**, **isDefault**:**'0'**}];

1. 循环用户选择的规格，根据规格名称和已选择的规格选项对原数组进行扩充，添加规格名称和值，新增的记录数与选择的规格选项个数相同，生成的顺序如下图：



## 2.2前端代码

### 2.2.1 生成SKU列表

pinyougou-shop-web/src/main/webapp/admin/goods\_edit.html，在选择规格属性后，调用生成SKU数组的方法(215行)：

|  |
| --- |
| <**span v-for="o in s.options"**>  <**input type="checkbox"  @click="selectSpecAttr($event,s.text,o.optionName);createItems()"**/>  {{*o*.optionName}} </**span**> |

pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js：

* 在数据模型data中定义变量



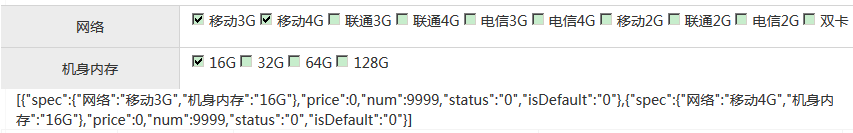
* 在methods中定义生成SKU数组方法

|  |
| --- |
| *// 生成SKU数组* **createItems** : **function** () {  */\*\* 定义SKU数组，并初始化 \*/* **this**.**goods**.**items** = [{**spec**:{}, **price**:0, **num**:9999,  **status**:**'0'**, **isDefault**:**'0'** }];  */\*\* 定义选中的规格选项数组 \*/* **var** specItems = **this**.**goods**.**goodsDesc**.**specificationItems**;  */\*\* 循环选中的规格选项数组 \*/* **for**(**var** i = 0; i < specItems.length; i++){  */\*\* 调用扩充原SKU数组方法，得到新的SKU数组 \*/* **this**.**goods**.**items** = **this**.swapItems(**this**.**goods**.**items**,  specItems[i].**attributeName**,  specItems[i].**attributeValue**);  } }, *// 扩充SKU数组* **swapItems** : **function** (items, attributeName, attributeValue) {  */\*\* 创建新的SKU数组 \*/* **var** newItems = **new *Array***();  */\*\* 迭代旧的SKU数组，循环扩充 \*/* **for**(**var** i = 0; i < items.**length**; i++){  */\*\* 获取一个SKU商品 \*/* **var** item = items[i];  */\*\* 迭代规格选项值数组 \*/* **for**(**var** j = 0; j < attributeValue.**length**; j++){  */\*\* 克隆旧的SKU商品，产生新的SKU商品 \*/* **var** newItem = ***JSON***.parse(***JSON***.stringify(item));  */\*\* 增加新的key与value \*/* newItem.**spec**[attributeName] = attributeValue[j];  */\*\* 添加到新的SKU数组 \*/* newItems.push(newItem);  }  }  **return** newItems; } |

goods\_edit.html页面上添加表达式，进行测试：

|  |
| --- |
| {{**goods**.items}} |

显示效果如下：



### 2.2.2 显示SKU列表

pinyougou-shop-web/src/main/webapp/admin/goods\_edit.html页面迭代产生SKU列表：

|  |
| --- |
| <**table class="table table-bordered table-striped table-hover dataTable"**>  <**thead**>  <**tr**>  <**th class="sorting"  v-for="s in goods.goodsDesc.specificationItems"**>  **{{*s*.attributeName}}**</**th**>  <**th class="sorting"**>价格</**th**>  <**th class="sorting"**>库存</**th**>  <**th class="sorting"**>是否启用</**th**>  <**th class="sorting"**>是否默认</**th**>  </**tr**>  </**thead**>  <**tbody**>  <**tr v-for="item in goods.items"**>  <**td v-for="s in goods.goodsDesc.specificationItems"**>  {{***item***.**spec**[***s***.attributeName]}}  </**td**>  <**td**><**input class="form-control"  v-model="item.price" placeholder="价格"**>  </**td**>  <**td**><**input class="form-control"  v-model="item.num" placeholder="库存数量"**>  </**td**>  <**td**><**input type="checkbox"  v-model="item.status"  :true-value="1"  :false-value="0"**></**td**>  <**td**><**input type="checkbox"  v-model="item.isDefault"  :true-value="1"  :false-value="0"**></**td**>  </**tr**>  </**tbody**> </**table**> |

## 2.3后端代码

pinyougou-sellergoods-service/src/main/java/com/pinyougou/sellergoods/service/impl/GoodsServiceImpl.java中注入数据访问接口：

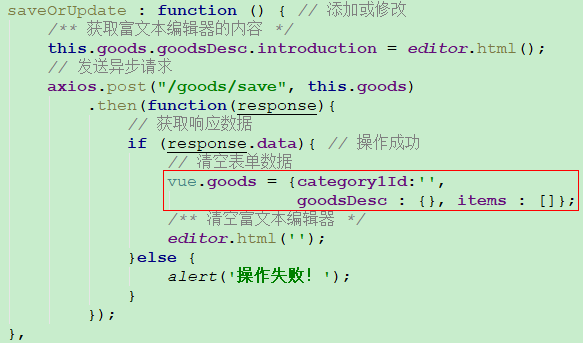
|  |
| --- |
| @Autowired **private** ItemMapper **itemMapper**; @Autowired **private** BrandMapper **brandMapper**; @Autowired **private** ItemCatMapper **itemCatMapper**; @Autowired **private** SellerMapper **sellerMapper**; |

pinyougou-sellergoods-service/src/main/java/com/pinyougou/sellergoods/service/impl/GoodsServiceImpl.java的save()方法，增加代码，实现对SKU具体商品信息的保存：

|  |
| --- |
| */\*\* 添加商品 \*/* **public void** save(Goods goods){  **try**{  */\*\* 设置未审核状态 \*/* goods.setAuditStatus(**"0"**);  */\*\* 添加SPU商品表 \*/* **goodsMapper**.insertSelective(goods);  */\*\* 添加商品描述表 \*/* goods.getGoodsDesc().setGoodsId(goods.getId());  **goodsDescMapper**.insertSelective(goods.getGoodsDesc());   */\*\* 迭代所有的SKU具体商品集合，往SKU表插入数据 \*/* **for** (Item item : goods.getItems()){  */\*\* 定义SKU商品的标题 \*/* StringBuilder title = **new** StringBuilder();  title.append(goods.getGoodsName());  */\*\* 把规格选项JSON字符串转化成Map集合 \*/* Map<String,Object> spec = JSON.*parseObject*(item.getSpec());  **for** (Object value : spec.values()) {  */\*\* 拼接规格选项到SKU商品标题 \*/* title.append(**" "** + value);  }  */\*\* 设置SKU商品的标题 \*/* item.setTitle(title.toString());  */\*\* 设置SKU商品图片地址 \*/* List<Map> imageList = JSON.*parseArray*(  goods.getGoodsDesc().getItemImages(), Map.**class**);  **if** (imageList != **null** && imageList.size() > 0){  */\*\* 取第一张图片 \*/* item.setImage((String)imageList.get(0).get(**"url"**));  }  */\*\* 设置SKU商品的分类(三级分类) \*/* item.setCategoryid(goods.getCategory3Id());  */\*\* 设置SKU商品的创建时间 \*/* item.setCreateTime(**new** Date());  */\*\* 设置SKU商品的修改时间 \*/* item.setUpdateTime(item.getCreateTime());  */\*\* 设置SPU商品的编号 \*/* item.setGoodsId(goods.getId());  */\*\* 设置商家编号 \*/* item.setSellerId(goods.getSellerId());  */\*\* 设置商品分类名称 \*/* item.setCategory(**itemCatMapper** .selectByPrimaryKey(goods.getCategory3Id()).getName());  */\*\* 设置品牌名称 \*/* item.setBrand(**brandMapper** .selectByPrimaryKey(goods.getBrandId()).getName());  */\*\* 设置商家店铺名称 \*/* item.setSeller(**sellerMapper**.selectByPrimaryKey(  goods.getSellerId()).getNickName());  */\*\* 往tb\_item表插入数据 \*/*  **itemMapper**.insertSelective(item);  }  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } |

## 2.4清空表单

pinyougou-shop-web\src\main\webapp\js\controller\goodsEditController.js



# 3.商品录入【是否启用规格】

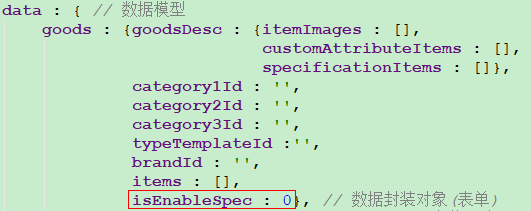
## 3.1需求分析

在规格面板添加是否启用规格，当用户没有选择该项，将原来的规格面板和SKU列表隐藏，用户保存商品后只生成一个SKU。



## 3.2前端代码

pinyougou-shop-web\src\main\webapp\js\controller\goodsEditController.js在数据模型中定义变量



pinyougou-shop-web/src/main/webapp/admin/goods\_edit.html添加复选框(202行)：

|  |
| --- |
| <**div class="row data-type"**>  <**div class="col-md-2 title"**>是否启用规格</**div**>  <**div class="col-md-10 data"**>  <**input type="checkbox"**  **v-model="goods.isEnableSpec"**  **:true-value="1"**  **:false-value="0"**/>  </**div**> </**div**> |

goods\_edit.html用v-show指令控制规格面板与SKU列表的显示与隐藏(209行):

|  |
| --- |
| <**div v-show="goods.isEnableSpec==1"**>  sku数据显示部分...  **</div>** |

## 3.3后端代码

pinyougou-sellergoods-service/src/main/java/com/pinyougou/sellergoods/service/impl/GoodsServiceImpl.java类的save()方法：

|  |
| --- |
| */\*\* 添加商品 \*/* **public void** save(Goods goods){  **try**{  */\*\* 设置未审核状态 \*/* goods.setAuditStatus(**"0"**);  */\*\* 添加SPU商品表 \*/* **goodsMapper**.insertSelective(goods);  */\*\* 添加商品描述表 \*/* goods.getGoodsDesc().setGoodsId(goods.getId());  **goodsDescMapper**.insertSelective(goods.getGoodsDesc());   */\*\* 判断是否启用规格 \*/* **if** (**"1"**.equals(goods.getIsEnableSpec())){  */\*\* 迭代所有的SKU具体商品集合，往SKU表插入数据 \*/* **for** (Item item : goods.getItems()){  */\*\* 定义具体商品的标题 \*/* StringBuilder title = **new** StringBuilder();  title.append(goods.getGoodsName());  */\*\* 把规格选项JSON字符串转化成Map集合 \*/* Map<String,Object> spec = JSON.*parseObject*(item.getSpec());  **for** (Object value : spec.values()) {  */\*\* 拼接规格选项到具体商品标题 \*/* title.append(**" "** + value);  }  */\*\* 设置SKU商品的标题 \*/* item.setTitle(title.toString());  */\*\* 设置SKU商品其它属性 \*/* setItemInfo(item, goods);  **itemMapper**.insertSelective(item);  }  }**else**{  */\*\* 创建SKU具体商品对象 \*/* Item item = **new** Item();  */\*\* 设置SKU商品的标题 \*/* item.setTitle(goods.getGoodsName());  */\*\* 设置SKU商品的价格 \*/* item.setPrice(goods.getPrice());  */\*\* 设置SKU商品库存数据 \*/* item.setNum(9999);  */\*\* 设置SKU商品启用状态 \*/* item.setStatus(**"1"**);  */\*\* 设置是否默认\*/* item.setIsDefault(**"1"**);  */\*\* 设置规格选项 \*/* item.setSpec(**"{}"**);  */\*\* 设置SKU商品其它属性 \*/* setItemInfo(item, goods);  **itemMapper**.insertSelective(item);  }  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } */\*\* 设置SKU商品信息 \*/* **private void** setItemInfo(Item item, Goods goods){  */\*\* 设置SKU商品图片地址 \*/* List<Map> imageList = JSON.*parseArray*(  goods.getGoodsDesc().getItemImages(), Map.**class**);  **if** (imageList != **null** && imageList.size() > 0){  */\*\* 取第一张图片 \*/* item.setImage((String)imageList.get(0).get(**"url"**));  }  */\*\* 设置SKU商品的分类(三级分类) \*/* item.setCategoryid(goods.getCategory3Id());  */\*\* 设置SKU商品的创建时间 \*/* item.setCreateTime(**new** Date());  */\*\* 设置SKU商品的修改时间 \*/* item.setUpdateTime(item.getCreateTime());  */\*\* 设置SPU商品的编号 \*/* item.setGoodsId(goods.getId());  */\*\* 设置商家编号 \*/* item.setSellerId(goods.getSellerId());  */\*\* 设置分类名称 \*/* item.setCategory(**itemCatMapper** .selectByPrimaryKey(goods.getCategory3Id()).getName());  */\*\* 设置品牌名称 \*/* item.setBrand(**brandMapper** .selectByPrimaryKey(goods.getBrandId()).getName());  */\*\* 设置商家店铺名称 \*/* item.setSeller(**sellerMapper**.selectByPrimaryKey(  goods.getSellerId()).getNickName()); } |

# 4.商家后台-商品管理【商品列表】

## 4.1需求分析

在商家后台，显示该商家的商品列表信息，如下图：



实现思路分析：

* 我们只查询当前登录商家的所有商品，不能查询其他商家的商品。
* 查询参数：状态码、商品名称(get请求中文乱码，需转码)
* 分页参数：page、rows
* 状态码需转成中文提示
* 条件查询数据绑定

## 4.2查询商家商品列表

### 4.2.1前端代码

pinyougou-shop-web/src/main/webapp/admin/goods.html引入js：

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

goods.html页面，放置分页组件：

|  |
| --- |
| *<!-- 分页组件 -->* <**pagination :pages="pages"  :current="page"  @navpage="search"**/> |

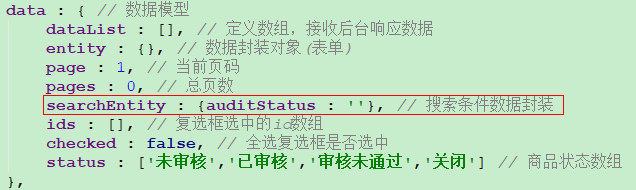
goods.html页面，v-for迭代商品数组：

|  |
| --- |
| <**tr v-for="entity in dataList"**>  <**td**><**input type="checkbox"**></**td**>  <**td**>{{**entity**.**id**}}</**td**>  <**td**>{{**entity**.goodsName}}</**td**>  <**td**>{{**entity**.**price**}}</**td**>  <**td**>{{**entity**.category1Name}}</**td**>  <**td**>{{**entity**.category2Name}}</**td**>  <**td**>{{**entity**.category3Name}}</**td**>  <**td**>{{**entity**.auditStatus}}</**td**>  <**td class="text-center"**>  <**button type="button" class="btn bg-olive btn-xs"**>修改</**button**>  </**td**> </**tr**> |

goods.html页面，添加查询条件(审核状态和商品名称)

|  |
| --- |
| <**div class="has-feedback"**>  状态：<**select v-model="searchEntity.auditStatus"**>  <**option value=""**>全部</**option**>  <**option value="0"**>未申核</**option**>  <**option value="1"**>已审核</**option**>  <**option value="2"**>审核未通过</**option**>  <**option value="3"**>关闭</**option**>  </**select**>  商品名称：<**input v-model="searchEntity.goodsName"**>  <**button class="btn btn-default" @click="search(1)"**>查询</**button**> </**div**> |

pinyougou-shop-web/src/main/webapp/js/controller/goodsController.js在数据模型data中定义变量:



### 4.2.2后端代码

pinyougou-shop-web/src/main/java/com.pinyougou/shop/controller/

GoodsController.java：

|  |
| --- |
| */\*\* 多条件分页查询商品 \*/* @GetMapping(**"/findByPage"**) **public** PageResult findByPage(Goods goods, Integer page,  @RequestParam(defaultValue = **"10"**) Integer rows){   */\*\* 获取登录商家编号 \*/* **String sellerId = (String)SecurityUtils.*getSubject*().getPrincipal();**  */\*\* 添加查询条件 \*/* goods.setSellerId(sellerId);  */\*\* GET请求中文转码 \*/* **if** (StringUtils.*isNoneBlank*(goods.getGoodsName())) {  **try** {  goods.setGoodsName(**new** String(goods  .getGoodsName().getBytes(**"ISO8859-1"**), **"UTF-8"**));  } **catch** (Exception e) {  e.printStackTrace();  }  }  */\*\* 调用服务层方法查询 \*/* **return goodsService**.findByPage(goods, page, rows); } |

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

|  |
| --- |
| *\*\*  \* 多条件分页查询商品  \** ***@param goods*** *商品对象  \** ***@param page*** *当前页码  \** ***@param rows*** *每页显示的记录数  \** ***@return*** *商品分页数据  \*/* PageResult findByPage(Goods goods,Integer page,Integer rows); |

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

|  |
| --- |
| */\*\* 多条件分页查询商品 \*/*  @Override**public** PageResult findByPage(Goods goods,**int** page,**int** rows){  **try**{  */\*\* 开始分页 \*/* PageInfo<Map<String,Object>> pageInfo =  PageHelper.*startPage*(page, rows)  .doSelectPageInfo(**new** ISelect() {  @Override  **public void** doSelect() {  **goodsMapper**.findAll(goods);  }  });  */\*\* 循环查询到的商品 \*/* **for** (Map<String,Object> map : pageInfo.getList()){  ItemCat itemCat1 =  **itemCatMapper**.selectByPrimaryKey(map.get(**"category1Id"**));  map.put(**"category1Name"**, itemCat1 != **null** ? itemCat1.getName() : **""**);  ItemCat itemCat2 =  **itemCatMapper**.selectByPrimaryKey(map.get(**"category2Id"**));  map.put(**"category2Name"**, itemCat2 != **null** ? itemCat2.getName() : **""**);  ItemCat itemCat3 =  **itemCatMapper**.selectByPrimaryKey(map.get(**"category3Id"**));  map.put(**"category3Name"**, itemCat3 != **null** ? itemCat3.getName() : **""**);  }  **return new** PageResult(pageInfo.getPages(),pageInfo.getList());  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |

pinyougou-mapper/src/main/java/com.pinyougou.mapper/GoodsMapper.java：

|  |
| --- |
| */\*\* 多条件查询商品 \*/* List<Map<String,Object>> findAll(Goods goods); |

pinyougou-mapper/src/main/resources/mappers/GoodsMapper.xml：

|  |
| --- |
| *<!-- 多条件查询商品 -->* <**select id="findAll" resultType="map"**>  SELECT id,goods\_name AS goodsName,price,  audit\_status AS auditStatus,category1\_id AS category1Id,  category2\_id AS category2Id,category3\_id AS category3Id  FROM `tb\_goods`  <**where**>  *<!-- 商家id -->* <**if test="sellerId != null and sellerId !=''"**>  seller\_id = #{sellerId}  </**if**>  *<!-- 商品名称 -->* <**if test="goodsName != null and goodsName !=''"**>  and goods\_name like CONCAT('%',#{goodsName},'%')  </**if**>  *<!-- 商品状态 -->* <**if test="auditStatus != null and auditStatus !=''"**>  and audit\_status = #{auditStatus}  </**if**>  </**where**>  order by id asc </**select**> |

显示效果如下：



## 4.3显示状态

goods.html页面修改列表显示：

|  |
| --- |
| <**td**>{{***status***[**entity**.auditStatus]}}</**td**> |

pinyougou-shop-web/src/main/webapp/js/controller/goodsController.js，添加状态文本数组：['未审核','已审核','审核未通过','关闭']

