**品优购电商系统开发**

**第6章**

**商品管理【1】**

传智播客.黑马程序员

# 课程目标

目标1：理解SPU和SKU

目标2：完成商品录入-基本功能

目标3：完成商品录入-商品介绍

目标4：商品录入-fastdfs实现图片上传

目标5：完成商品录入-商品分类

目标6：完成扩展属性录入的功能

# 1.电商概念及表结构分析

## 1.1**电商概念SPU与SKU**

**SPU = Standard Product Unit (标准产品单位)**SPU是商品信息聚合的最小单位，是一组可复用、易检索的标准化信息的集合，该集合描述了一个产品的特性。也就是，属性值、特性相同的商品就可以称为一个SPU。

例如：

**iphone8就是一个SPU（与颜色、款式、套餐等都无关）**

**SKU = Stock Keeping Unit (库存量单位)**SKU即库存进出计量的单位，可以是以件、盒、台、双等为单位。  
SKU是物理上不可分割的最小存货单元。

SKU通常表示：规格、颜色、款式、内存、版本、套餐等。



例如：

**Apple iPhone 8 64GB 红色特别版 移动联通电信4G手机**

**SKU = SPU + (规格、颜色、款式、内存、版本、套餐等) 的组合**

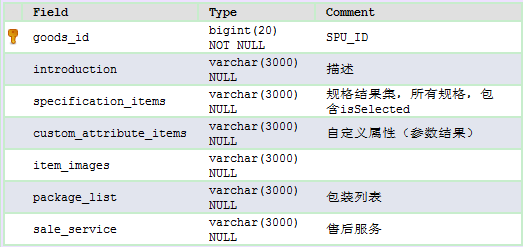
## 1.2**表结构分析**

**具体参考【数据库表关系图.png】**

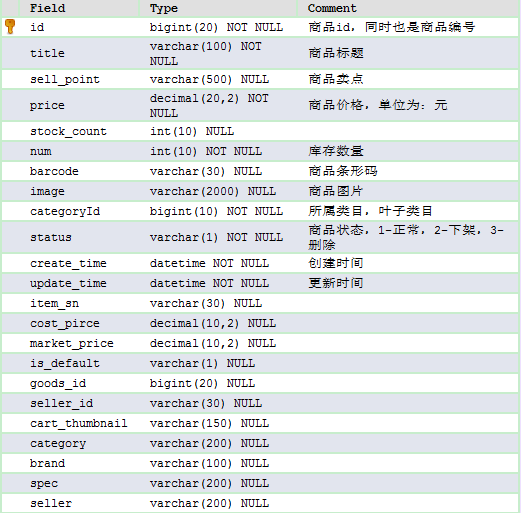
tb\_goods 商品SPU表(标准商品)



tb\_goods\_desc 商品描述表



tb\_item 商品SKU表(具体商品)

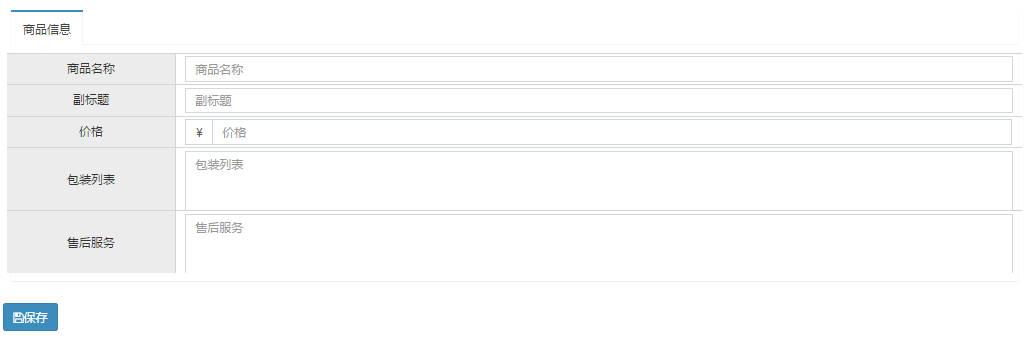


# 2.商家后台-商品录入【基本信息】

## 2.1**需求分析**

在商家后台实现商品录入功能，实际上是对tb\_goods、tb\_goods\_desc、tb\_item这三张表做插入数据操作。

基本信息包括：商品名称、副标题、价格、包装列表、售后服务



问题：页面表单数据怎样封装？我们可以用Goods实体进行封装，但需要修改Goods实体类让它组合GoodsDesc与Item实体。

Goods 与 GoodsDesc是一对一的关系

Goods 与 Item是一对多的关系

Goods数据保存到tb\_goods表

goods.goodsDesc数据保存到tb\_goods\_desc表

goods.items数据保存到tb\_item表

## 2.2实体类

修改pinyougou-pojo/src/main/java/com.pinyougou.pojo中的Goods.java实体类(封装商品描述与具体SKU商品)：

@Transient注解的属性，通用Mapper不会把它当作tb\_goods表中的列（表示不关心）

|  |
| --- |
| @Table(name=**"tb\_goods"**) **public class** Goods **implements** Serializable{  */\*\* 商品描述 \*/* @Transient **private** GoodsDesc **goodsDesc**; */\*\* 商品SKU集合 \*/* @Transient **private** List<Item> **items**; */\*\* setter and getter method \*/*  } |

## 2.3**前端代码**

### 2.3.1 html页面

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

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

goods\_edit.html表单数据绑定：

|  |
| --- |
| <**div class="col-md-2 title"**>商品名称</**div**> <**div class="col-md-10 data"**>  <**input type="text" v-model="goods.goodsName"  class="form-control" placeholder="商品名称"**/> </**div**> <**div class="col-md-2 title"**>品牌</**div**> <**div class="col-md-10 data"**>  <**select class="form-control"**></**select**> </**div**>  <**div class="col-md-2 title"**>副标题</**div**> <**div class="col-md-10 data"**>  <**input type="text" v-model="goods.caption"  class="form-control" placeholder="副标题"**/> </**div**> <**div class="col-md-2 title"**>价格</**div**> <**div class="col-md-10 data"**>  <**div class="input-group"**>  <**span class="input-group-addon"**>¥</**span**>  <**input type="text" v-model="goods.price"  class="form-control" placeholder="价格"**/>  </**div**> </**div**>  <**div class="col-md-2 title editer"**>商品介绍</**div**> <**div class="col-md-10 data editer"**>  <**textarea name="content"  style="width**: 800**px**; **height**: 400**px**; **visibility**: **hidden**;**"**></**textarea**> </**div**> <**div class="col-md-2 title rowHeight2x"**>包装列表</**div**> <**div class="col-md-10 data rowHeight2x"**>  <**textarea rows="4" v-model="goods.goodsDesc.packageList"  class="form-control" placeholder="包装列表"**></**textarea**> </**div**> <**div class="col-md-2 title rowHeight2x"**>售后服务</**div**> <**div class="col-md-10 data rowHeight2x"**>  <**textarea rows="4" v-model="goods.goodsDesc.saleService"  class="form-control" placeholder="售后服务"**></**textarea**> </**div**> |

保存按钮(276行)：

|  |
| --- |
| <**button class="btn btn-primary" @click="saveOrUpdate();"**>  <**i class="fa fa-save"**></**i**>保存 </**button**> |

### 2.3.2 js代码

pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js，在添加成功，清空实体（因为编辑页面无列表，不需要重新查询数据）

|  |
| --- |
| *// 创建Vue对象* **var** vue = **new Vue**({  **el** : **'#app'**, *// 元素绑定* **data** : { *// 数据模型* **goods : {goodsDesc : {}}** *// 数据封装对象(表单)* },  **methods** : { *// 定义操作方法* saveOrUpdate : **function** () { *// 添加或修改  // 发送异步请求* **axios**.post(**"/goods/save"**, **this**.**goods**)  .then(**function**(response){  *// 获取响应数据* **if** (response.**data**){ *// 操作成功  // 清空表单数据* **vue.goods = {goodsDesc : {}};**  }**else** {  *alert*(**'操作失败！'**);  }  });  }  },  created : **function** () { *// 创建生命周期(初始化方法)* } }); |

## 2.4**后端代码**

### 2.4.1**控制层**

pinyougou-shop-web/src/main/java/com.pinyougou.shop.controller包下创建GoodsController.java

|  |
| --- |
| @RestController @RequestMapping(**"/goods"**) **public class** GoodsController {  @Reference(timeout = 10000)  **private** GoodsService **goodsService**;  */\*\* 添加商品 \*/* @PostMapping(**"/save"**)  **public boolean** save(@RequestBody Goods goods){  **try**{   ***/\*\* 获取登录用户名 \*/* String sellerId = (String)SecurityUtils  .*getSubject*().getPrincipal();**  ***/\*\* 设置商家ID \*/* goods.setSellerId(sellerId);**  **goodsService**.save(goods);  **return true**;  }**catch**(Exception ex){  ex.printStackTrace();  }  **return false**;  } } |

### 2.4.2**服务实现层**

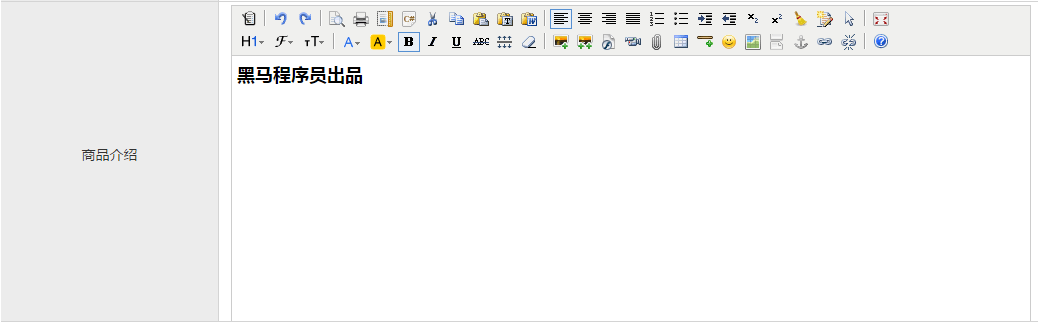
pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl包下创建GoodsServiceImpl.java

|  |
| --- |
| @Service @Transactional **public class** GoodsServiceImpl **implements** GoodsService {  */\*\* 注入数据访问层代理对象 \*/* @Autowired  **private** GoodsMapper **goodsMapper**;  @Autowired  **private** GoodsDescMapper **goodsDescMapper**;  */\*\* 添加商品 \*/* @Override  **public void** save(Goods goods){  **try**{  *// 设置未申核状态* goods.setAuditStatus(**"0"**);  **goodsMapper**.insertSelective(goods);  *// 为商品描述对象设置主键id* goods.getGoodsDesc().setGoodsId(goods.getId());  **goodsDescMapper**.insertSelective(goods.getGoodsDesc());  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  }  }  @Override  **public void** update(Goods goods) {  }  @Override  **public void** delete(Serializable id) {  }  @Override  **public void** deleteAll(Serializable[] ids) {  }  @Override  **public** Goods findOne(Serializable id) {  **return null**;  }  @Override  **public** List<Goods> findAll() {  **return null**;  }  @Override  **public** List<Goods> findByPage(Goods goods, **int** page, **int** rows) {  **return null**;  } } |

# 3.商家后台-商品录入【商品介绍】

## **3.1需求分析**

实现商品介绍的录入，要求使用富文本编辑器



## **3.2富文本编辑器介绍**

富文本编辑器，Rich Text Editor, 简称 RTE, 它提供类似于 Microsoft Word 的编辑功能，容易被不会编写 HTML 的用户并需要设置各种文本格式的用户所喜爱。常用的富文本编辑器：

KindEditor http://kindeditor.net/

UEditor http://ueditor.baidu.com/website/

CKEditor http://ckeditor.com/

## **3.3使用kindeditor完成商品介绍的录入**

### **3.3.1初始化kindeditor编辑器**

在pinyougou-shop-web/src/main/webapp/admin/goods\_edit.html页面引入kindeditor的css与js

|  |
| --- |
| *<!-- 富文本编辑器 -->* <**link rel="stylesheet" href="/plugins/kindeditor/themes/default/default.css"**/> <**script src="/plugins/kindeditor/kindeditor-min.js"**></**script**> <**script src="/plugins/kindeditor/lang/zh\_CN.js"**></**script**> |

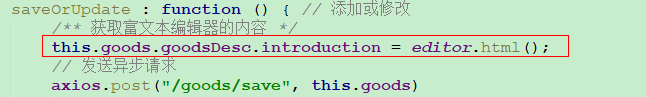
goods\_edit.html页面中添加JS代码，用于初始化kindeditor：

|  |
| --- |
| *<!-- 正文区域 /-->* <**script type="text/javascript"**>  **var *editor***;  **KindEditor**.ready(**function**(K) {  ***editor*** = K.create(**'textarea[name="content"]'**);  }); </**script**> |

**说明：kindEditor常用的一个方法html()，可用于获取编辑器的内容，与设置编辑器的内容。**

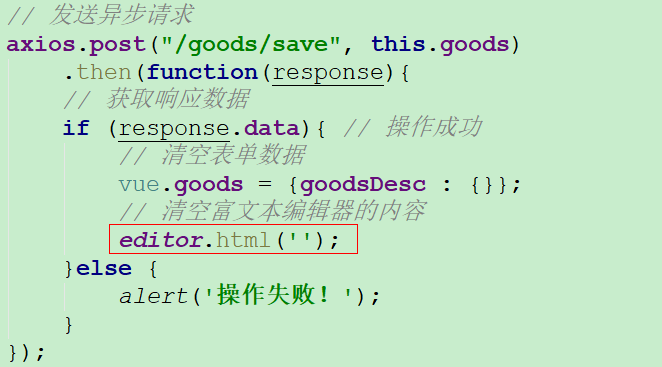
### **3.3.2获取kindeditor编辑器的内容**

pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js的saveOrUpdate()方法中第一行添加



### **3.3.3清空kindeditor编辑器的内容**

修改goodsEditController.js的saveOrUpdate()方法



# 4.商家后台-商品录入【商品图片上传】

## 4.1**需求分析**

在商品录入界面实现多图片上传



当用户点击新建按钮，弹出上传窗口



实现思路分析：

* Vuejs图片上传(异步请求上传图片)
* SpringMVC文件上传(配置文件上传依赖jar、配置文件上传解析器、控制器接收上传的文件)
* FastDFS图片文件器(配置客户端依赖jar、配置文件、编写上传服务器代码)

## **4.2图片上传**

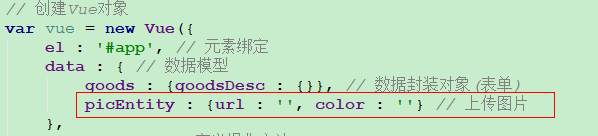
### **4.2.1 前端代码**

goods\_edit.html，修改图片上传窗口，调用上传方法，回显上传图片(300行左右)

|  |
| --- |
| <**table class="table table-bordered table-striped"**>  <**tr**>  <**td**>颜色</**td**>  <**td**><**input class="form-control"  v-model="picEntity.color" placeholder="颜色"**></**td**>  </**tr**>  <**tr**>  <**td**>商品图片</**td**>  <**td**>  <**table**>  <**tr**>  <**td**><**input type="file" id="file"**/>  <**button class="btn btn-primary"  @click="uploadFile()"  type="button"**>上传</**button**>  </**td**>  <**td**><**img :src="picEntity.url"  width="200px" height="200px"**></**td**>  </**tr**>  </**table**>  </**td**>  </**tr**> </**table**> |

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

* 在数据模型中定义变量



* 定义文件异步上传方法

|  |
| --- |
| uploadFile : **function** () { *// 文件异步上传  /\*\* 创建表单对象 \*/* **var** formData = **new *FormData***();  */\*\* 追加需要上传的文件 \*/* formData.append(**"file"**, file.files[0]);  */\*\* 发送异步请求上传文件 \*/* **axios**({  **method** : **'post'**, *// 请求方式* **url** : **"/upload"**, *// 请求URL* **data** : formData, *// 表单数据* **headers** : {**'Content-Type'** : **"multipart/form-data"**} *// 请求头* }).then(**function**(response){  */\*\* 如果上传成功，取出url \*/* **if**(response.**data**.**status** == 200){  */\*\* 设置图片访问地址 \*/* vue.**picEntity**.**url** = response.**data**.**url**;  }**else**{  *alert*(**"上传失败！"**);  }  }); } |

goods\_edit.html修改新建按钮，绑定点击事件(122行)，用于初始化对象(清空数据)

|  |
| --- |
| *<!-- 颜色图片 -->* <**div class="btn-group"**>  <**button type="button" class="btn btn-default" title="新建"  data-target="#uploadModal" data-toggle="modal"  @click="picEntity={url:'',color:''}"**>  <**i class="fa fa-file-o"**></**i**> 新建  </**button**> </**div**> |

### **4.2.2 后端代码**

#### 4.2.2.1 配置依赖

pinyougou-shop-web/pom.xml，配置引入依赖

|  |
| --- |
| *<!-- commons-fileupload -->* <**dependency**>  <**groupId**>commons-fileupload</**groupId**>  <**artifactId**>commons-fileupload</**artifactId**> </**dependency**>  *<!-- fastdfs-client -->* <**dependency**>  <**groupId**>org.csource</**groupId**>  <**artifactId**>fastdfs-client</**artifactId**> </**dependency**> |

#### 4.2.2.2 配置文件上传解析器

pinyougou-shop-web/src/main/resources/springmvc.xml配置文件上传解析器

|  |
| --- |
| *<!-- 配置文件上传解析器 -->* <**bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver"**>  *<!-- 设置文件上传默认编码 -->* <**property name="defaultEncoding" value="UTF-8"**/>  *<!-- 设置文件上传大小 2MB: 2\*1024\*1024 -->* <**property name="maxUploadSize" value="2097152"**/> </**bean**> |

#### 4.2.2.3 控制器层

在pinyougou-shop-web/src/main/java/com.pinyougou.shop.controller包下创建UploadController.java

|  |
| --- |
| @RestController **public class** UploadController {  */\*\* 注入文件服务器访问地址 \*/* @Value(**"${fileServerUrl}"**)  **private** String **fileServerUrl**;  */\*\* 文件上传 \*/* @PostMapping(**"/upload"**)  **public** Map<String, Object>  upload(@RequestParam(**"file"**)MultipartFile multipartFile) {   Map<String, Object> data = **new** HashMap<>();  data.put(**"status"**, 500);  **try** {  */\*\* 加载配置文件，产生该文件绝对路径 \*/* String conf\_filename = **this**.getClass()  .getResource(**"/fastdfs\_client.conf"**).getPath();  */\*\* 初始化客户端全局对象 \*/* ClientGlobal.*init*(conf\_filename);  */\*\* 创建存储客户端对象 \*/* StorageClient storageClient = **new** StorageClient();  */\*\* 获取原文件名 \*/* String originalFilename =  multipartFile.getOriginalFilename();  */\*\* 上传文件到FastDFS服务器 \*/* String[] arr = storageClient  .upload\_file(multipartFile.getBytes(),  FilenameUtils.*getExtension*(originalFilename), **null**);  */\*\* 拼接返回的 url 和 ip 地址，拼装成完整的 url \*/* StringBuilder url = **new** StringBuilder(**fileServerUrl**);  **for** (String str : arr){  url.append(**"/"** + str);  }  data.put(**"status"**, 200);  data.put(**"url"**, url.toString());  } **catch** (Exception e) {  e.printStackTrace();  }  **return** data;  } } |

#### 4.2.2.4 配置文件

1. 将“fastdfs-test”工程中的src/test/resources/fastdfs\_client.conf 文件拷贝到pinyougou-shop-web/src/main/resources目录下。



1. 在pinyougou-shop-web/src/main/resources/目录下，新建

application.properties，并添加属性

|  |
| --- |
| # 配置文件服务器的访问地址  fileServerUrl=http://192.168.12.131 |

1. pinyougou-shop-web/src/main/resources/springmvc.xml配置加载application.properties

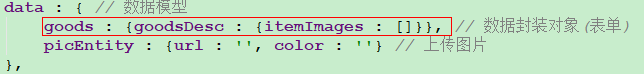
|  |
| --- |
| *<!-- 加载属性文件 -->* <**context:property-placeholder location="classpath:application.properties"**/> |

## 4.3 图片列表

goods\_edit.html修改上传窗口的保存按钮，绑定点击事件

|  |
| --- |
| <**button class="btn btn-success" data-dismiss="modal"  @click="addPic()"  aria-hidden="true"**>保存</**button**> |

pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js定义数据模型变量与方法



|  |
| --- |
| addPic : **function** () { *// 添加图片到图片数组* **this**.**goods**.**goodsDesc**.**itemImages**.push(**this**.**picEntity**); } |

goods\_edit.html迭代图片列表(140行左右)

|  |
| --- |
| <**tr v-for="(pic,index) in goods.goodsDesc.itemImages"**>  <**td**>{{pic.**color**}}</**td**>  <**td**><**img :src="pic.url" width="100px" height="100px"**></**td**>  <**td**>  <**button type="button" class="btn btn-default" title="删除"**>  <**i class="fa fa-trash-o"**></**i**> 删除  </**button**>  </**td**> </**tr**> |

## 4.4 移除图片

goods\_edit.html修改列表中的删除按钮，绑定点击事件

|  |
| --- |
| <**button type="button" class="btn btn-default" title="删除"  @click="removePic(index)"**>  <**i class="fa fa-trash-o"**></**i**> 删除</**button**> |

pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js增加方法

|  |
| --- |
| removePic : **function** (index) { *// 从数组中删除图片* **this**.**goods**.**goodsDesc**.**itemImages**.splice(index, 1); } |

# 5.商品录入【选择商品分类】

## 5.1需求分析

在商品录入界面实现商品分类的选择（三级分类）效果如下：

IMG_256

* 当用户选择一级分类后，二级分类列表要相应更新。
* 当用户选择二级分类后，三级分类列表要相应更新。
* 当用户选择三级分类后，需要读到它对应的类型模版ID。

## 5.2商品分类下拉选择框

### 5.2.1后端代码

在pinyougou-shop-web/src/main/java/com.pinyougou.shop.controller包下创建ItemCatController.java

|  |
| --- |
| @RestController @RequestMapping(**"/itemCat"**) **public class** ItemCatController {  */\*\* 定义商品分类服务接口 \*/* @Reference(timeout=10000)  **private** ItemCatService **itemCatService**;  */\*\* 根据父级id查询商品分类 \*/* @GetMapping(**"/findItemCatByParentId"**)  **public** List<ItemCat> findItemCatByParentId(  @RequestParam(value=**"parentId"**,  defaultValue=**"0"**)Long parentId){  **return itemCatService**.findItemCatByParentId(parentId);  } } |

### 5.2.2前端代码

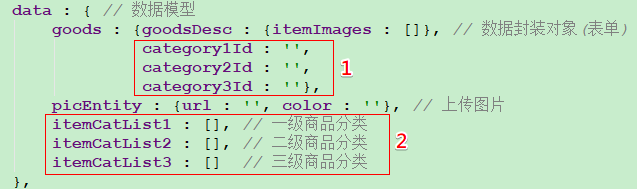
#### 5.2.2.1一级分类下拉选择框

goods\_edit.html一级分类下拉选择框

|  |
| --- |
| <**select v-model="goods.category1Id" class="form-control"**>  <**option value=""**>==请选择一级分类==</**option**>  <**option :value="i.id" v-for="i in itemCatList1"**>  {{***i***.**name**}}  </**option**> </**select**> |

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

* 在数据模型中定义变量



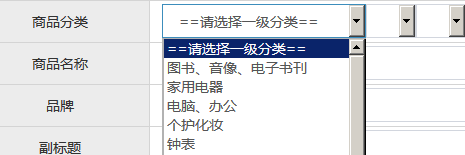
* 根据父级id查询商品分类

|  |
| --- |
| *// 根据父级id查询商品分类* findItemCatByParentId : **function** (parentId, name) {  **axios**.get(**"/itemCat/findItemCatByParentId?parentId="** + parentId).then(**function**(response){  **vue[name]** = response.**data**;  }); } |

* 初始化调用

|  |
| --- |
| created : **function** () { *// 创建生命周期(初始化方法)  // 查询商品分类* **this**.findItemCatByParentId(0, **'itemCatList1'**); } |

运行效果如下：



#### 5.2.2.2二级分类下拉选择框

goods\_edit.html二级分类下拉选择框

|  |
| --- |
| <**select v-model="goods.category2Id" class="form-control select-sm"**>  <**option value=""**>==请选择二级分类==</**option**>  <**option :value="i.id" v-for="i in itemCatList2"**>  {{***i***.**name**}}  </**option**> </**select**> |

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

|  |
| --- |
| **watch** : {  *// 监控一级分类id发生改变，查询二级分类* **'goods.category1Id'** : **function** (newVal, oldVal) {  **this**.**goods**.**category2Id** = **""**;  **if** (newVal){ *// 不是空字符串* **this**.findItemCatByParentId(newVal, **"itemCatList2"**)  }**else**{  **this**.**itemCatList2** = [];  }  }  }, |

**说明：watch用于监控某个变量的值，被监控的值发生变化，就自动执行相应的函数。**

#### 5.2.2.3三级分类下拉选择框

goods\_edit.html三级分类下拉选择框

|  |
| --- |
| <**select v-model="goods.category3Id" class="form-control select-sm"**>  <**option value=""**>==请选择三级分类==</**option**>  <**option :value="i.id" v-for="i in itemCatList3"**>  {{***i***.**name**}}  </**option**> </**select**> |

pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js的watch中，增加代码：

|  |
| --- |
| *// 监控二级分类id发生改变，查询三级分类* **'goods.category2Id'** : **function** (newVal, oldVal) {  **this**.**goods**.**category3Id** = **""**;  **if** (newVal){ *// 不是空字符串* **this**.findItemCatByParentId(newVal, **"itemCatList3"**)  }**else**{  **this**.**itemCatList3** = [];  } } |

## 5.3读取类型模板ID

### 5.3.1前端代码

pinyougou-shop-web/src/main/webapp/admin/goods\_edit.html显示模板ID

|  |
| --- |
| <**td v-if="goods.typeTemplateId"**>**&nbsp;** 模板ID:{{**goods**.**typeTemplateId**}}</**td**> |

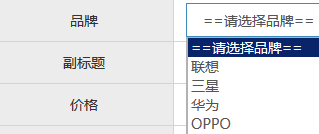
pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js的watch中，增加代码：

|  |
| --- |
| *// 监控三级分类id发生改变，获取类型模板id* **'goods.category3Id'** : **function** (newVal, oldVal) {  **this**.**goods**.**typeTemplateId** = **""**;  **if** (newVal){ *// 不是空字符串  // 循环三级分类数组 List<ItemCat> : [{},{}]* **for** (**var** i = 0; i < **this**.**itemCatList3**.length; i++){  *// 取一个数组元素 {}* **var** itemCat = **this**.**itemCatList3**[i];  *// 判断id* **if** (itemCat.**id** == newVal){  **this**.**goods**.**typeTemplateId** = itemCat.typeId;  **break**;  }  }  } } |

# 6.商品录入【品牌选择】

## 6.1需求分析

在用户选择商品分类后，品牌列表要根据用户所选择的商品分类进行更新。具体的逻辑是根据用户选择的三级商品分类找到对应的商品类型模板，商品类型模板中存储了品牌的列表json数据。



## 6.2前端代码

pinyougou-shop-web/src/main/webapp/admin/goods\_edit.html页面添加品牌选择框：

|  |
| --- |
| <**select v-model="goods.brandId" class="form-control"**>  <**option value=""**>==请选择品牌==</**option**>  <**option :value="b.id" v-for="b in brandList"**>  {{***b***.**text**}}  </**option**> </**select**> |

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

* 在数据模型中定义变量



* 在watch中定义监控方法

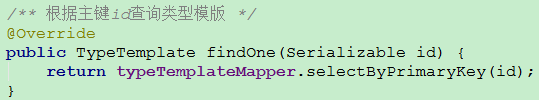
|  |
| --- |
| *// 监控类型模板id发生改变，获取类型模板对象* **'goods.typeTemplateId'** : **function** (newVal, oldVal) {  **this**.**goods**.**brandId** = **""**;  **if** (newVal){ *// 不是空字符串* **axios**.get(**"/typeTemplate/findOne?id="** + newVal)  .then(**function**(response){  */\*\* 获取模板中的品牌列表 \*/* vue.**brandList** = ***JSON***.parse(response.**data**.brandIds);  });  } } |

## 6.3后端代码

pinyougou-shop-web\src\main\java\com\pinyougou\shop\controller包下创建TypeTemplateController.java

|  |
| --- |
| @RestController @RequestMapping(**"/typeTemplate"**) **public class** TypeTemplateController {  @Reference(timeout = 10000)  **private** TypeTemplateService **typeTemplateService**;  */\*\* 根据主键id查询类型模版 \*/* @GetMapping(**"/findOne"**)  **public** TypeTemplate findOne(Long id){  **return typeTemplateService**.findOne(id);  } } |

pinyougou-sellergoods-service\src\main\java\com\pinyougou\sellergoods\service\impl\TypeTemplateServiceImpl.java



# 7.商品录入【扩展属性】

## 7.1需求分析

在商品录入时实现扩展属性的录入。



实现思路分析：商品的扩展属性来自它对应的类型模版中的扩展属性。我们需要把类型模版中的扩展属性赋给商品的扩展属性，再在页面上进行迭代显示。

## 7.2前端代码

pinyougou-shop-web/src/main/webapp/js/controller/goodsEditController.js，在用户更新模板ID时，读取模板中的扩展属性赋给商品的扩展属性。

* 在数据模型中定义变量



* 修改watch中监控的方法

|  |
| --- |
| *// 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**);  });  }**else**{ *// 清空数据* **this**.**brandList** = [];  **this**.**goods**.**goodsDesc**.**customAttributeItems** = [];  } } |

pinyougou-shop-web/src/main/webapp/admin/goods\_edit.html(181行)：

|  |
| --- |
| *<!--####### 扩展属性 #######-->* <**div class="tab-pane" id="customAttribute"**>  <**div class="row data-type"**>  <**div v-for="c in goods.goodsDesc.customAttributeItems"**>  <**div class="col-md-2 title"**>{{*c*.**text**}}</**div**>  <**div class="col-md-10 data"**>  <**input class="form-control" v-model="c.value"  :placeholder="***c***.text"/**>  </**div**>  </**div**>  </**div**> </**div**> |