



品优购电商系统开发

第 7 章

商品修改与审核

传智播客.黑马程序员

课程目标

目标 1：完成商家后台商品列表的功能

目标 2：完成商家后台商品修改的功能

目标 3：完成运营商后台商品审核的功能

目标 4：完成运营商后台商品删除的功能

目标 5：掌握注解式事务的配置

1.商家后台-商品管理【商品列表】

1.1 需求分析

在商家后台，显示该商家的商品列表信息，如下图：

<input type="button" value="新建"/> <input type="button" value="删除"/> <input checked="" type="button" value="开启"/> <input type="button" value="屏蔽"/> <input type="button" value="刷新"/>				状态： 全部		商品名称： <input type="text"/>	<input type="button" value="查询"/>	
<input type="checkbox"/>	商品ID	商品名称	商品价格	一级分类	二级分类	三级分类	状态	操作
<input type="checkbox"/>	149187842867896	测试商品	120	数码	电子教育	点读机/笔		<input type="button" value="修改"/>
<input type="checkbox"/>	149187842867897	测试商品	100	电脑、办公	网络产品	4G/3G上网	未申请	<input type="button" value="修改"/>
<input type="checkbox"/>	149187842867898	测试商品	111	食品饮料、保健食品	粮油调味	有机食品	已驳回	<input type="button" value="修改"/>
<input type="checkbox"/>	149187842867899	测试	100	数码	时尚影音	专业音频	未申请	<input type="button" value="修改"/>
<input type="checkbox"/>	149187842867900	尼康3010	100	数码	摄影摄像	摄像机	未申请	<input type="button" value="修改"/>
<input type="checkbox"/>	149187842867901	妹妹奶茶	120	食品饮料、保健食品	饮料冲调	咖啡/奶茶	审核通过	<input type="button" value="修改"/>
<input type="checkbox"/>	149187842867902	不倒翁	100	玩具乐器	娃娃玩具	智能娃娃	审核通过	<input type="button" value="修改"/>
<input type="checkbox"/>	149187842867903	aa	11	手机	运营商	选号入网	申请中	<input type="button" value="修改"/>
<input type="checkbox"/>	149187842867904	好商品	11	电脑、办公	办公设备	碎纸机	申请中	<input type="button" value="修改"/>
<input type="checkbox"/>	149187842867905	蓝魔303	4	数码	时尚影音	MP3/MP4	申请中	<input type="button" value="修改"/>

1.2 查询商家商品列表

1.2.1 后端代码

修改 pinyougou-shop-web 工程的 GoodsController.java 的 search 方法

```
@RequestMapping("/search")
```



```
public PageResult search(@RequestBody TbGoods goods, int page, int rows ){

    //获取商家 ID

    String sellerId =
SecurityContextHolder.getContext().getAuthentication().getName();

    //添加查询条件

    goods.setSellerId(sellerId);

    return goodsService.findPage(goods, page, rows);

}
```

修改 pinyougou-sellergoods-service 工程 com.pinyougou.sellergoods.service.impl 的 findPage 方法，修改条件构建部分代码，将原来的模糊匹配修改为精确匹配

```
if(goods.getSellerId()!=null && goods.getSellerId().length()>0){

    //criteria.andSellerIdLike("%"+goods.getSellerId()+"%");

    criteria.andSellerIdEqualTo(goods.getSellerId());

}
```

1.2.2 前端代码

修改 goods.html. 引入 js

```
<script type="text/javascript" src="../../plugins/angularjs/angular.min.js"></script>

<!-- 分页组件开始 -->

<script src="../../plugins/angularjs/pagination.js"></script>

<link rel="stylesheet" href="../../plugins/angularjs/pagination.css">

<!-- 分页组件结束 -->

<script type="text/javascript" src="../../js/base_pagination.js"></script>

<script type="text/javascript" src="../../js/service/goodsService.js"></script>
```



```
<script type="text/javascript" src="../../js/service/itemCatService.js"></script>

<script type="text/javascript" src="../../js/service/uploadService.js"></script>

<script type="text/javascript" src="../../js/service/typeTemplateService.js"></script>

<script type="text/javascript" src="../../js/controller/baseController.js"></script>

<script type="text/javascript" src="../../js/controller/goodsController.js"></script>
```

添加指令

```
<body class="hold-transition skin-red sidebar-mini" ng-app="pinyougou"
ng-controller="goodsController">
```

在页面上放置分页控件

```
<tm-pagination conf="paginationConf"></tm-pagination>
```

循环列表

```
<tr ng-repeat="entity in list">

  <td><input type="checkbox"></td>

  <td>{{entity.id}}</td>

  <td>{{entity.goodsName}}</td>

  <td>{{entity.price}}</td>

  <td>{{entity.category1Id}}</td>

  <td>{{entity.category2Id}}</td>

  <td>{{entity.category3Id}}</td>

  <td>

    {{entity.auditStatus}}

  </td>

  <td class="text-center">

    <button type="button" class="btn bg-olive btn-xs">修改</button>
```



```
</td>

</tr>
```

显示效果如下：

	商品ID	商品名称	商品价格	一级分类	二级分类	三级分类	状态	操作
	149187842867932	小米手机	1389				0	修改
	149187842867933	大米手机	8888				0	修改
	149187842867934	黄米手机	6666				0	修改
	149187842867935	三星手机	3000				0	修改
	149187842867936	四星手机	1				0	修改
	149187842867937	测试	19	378	386	389	0	修改
	149187842867938	测试商品	100	438	486	490	0	修改
	149187842867939			161	186	192	0	修改
	149187842867940	测试商品	111	378	397	401	0	修改
	149187842867941	橙子	100	161	202	205	0	修改

1.3 显示状态

修改 goodsController.js，添加 state 数组

```
$scope.status=['未审核','已审核','审核未通过','关闭'];//商品状态
```

修改列表显示

```
{{status[entity.auditStatus]}}
```

显示效果如下：

	商品ID	商品名称	商品价格	一级分类	二级分类	三级分类	状态	操作
	149187842867942	联想笔记本	6666	580	625	627	未审核	修改
	149187842867943	测试商品	20	865	866	874	未审核	修改
	149187842867944	红烧牛肉面	4	378	405	410	未审核	修改
	149187842867945	康师傅漱口水	1	249	264	267	未审核	修改
	149187842867946	终极防盗锁	10	438	486	489	未审核	修改
	149187842867947	送流量的手机	1000	558	562	1204	已审核	修改
	149187842867948	测试商品	10	699	718	722	已审核	修改



1.4 显示分类

商品管理

新建

删除

开启

屏蔽

刷新

状态：

全部

商品名称：

查询

<div><div></div></div>	商品ID	商品名称	商品价格	一级分类	二级分类	三级分类	状态	操作
<div><div></div></div>	149187842867896	测试商品	120	580	625	630		<div>修改</div>
<div><div></div></div>	149187842867897	测试商品	100	161	202	207	未申请	<div>修改</div>
<div><div></div></div>	149187842867898	测试商品	111	378	405	411	已驳回	<div>修改</div>
<div><div></div></div>	149187842867899	测试	100	580	614	622	未申请	<div>修改</div>
<div><div></div></div>	149187842867900	尼康3010	100	580	581	585	未申请	<div>修改</div>
<div><div></div></div>	149187842867901	妹妹奶茶	120	378	412	416	审核通过	<div>修改</div>
<div><div></div></div>	149187842867902	不倒翁	100	495	512	515	审核通过	<div>修改</div>
<div><div></div></div>	149187842867903	aa	11	558	562	566	申请中	<div>修改</div>
<div><div></div></div>	149187842867904	好商品	11	161	210	219	申请中	<div>修改</div>
<div><div></div></div>	149187842867905	蓝魔303	4	580	614	615	申请中	<div>修改</div>

我们现在的列表中的分类仍然显示 ID

如何才能显示分类的名称呢？

方案一：在后端代码写关联查询语句，返回的数据中直接有分类名称。

方案二：在前端代码用 ID 去查询后端，异步返回商品分类名称。

我们目前采用方案二：

(1) 修改 goodsController

```

$scope.itemCatList=[]; //商品分类列表

//加载商品分类列表

$scope.findItemCatList=function(){

    itemCatService.findAll().success(

        function(response){

            for(var i=0;i<response.length;i++){

                $scope.itemCatList[response[i].id]=response[i].name;
            }
        }
    );
}

```



```
        }  
  
    }  
  
    );  
}
```

代码解释：因为我们需要根据分类 ID 得到分类名称，所以我们将返回的分页结果以数组形式再次封装。

(2) 修改 goods.html ,增加初始化调用

```
<body class="hold-transition skin-red sidebar-mini" ng-app="pinyougou"  
ng-controller="goodsController" ng-init="findItemCatList()">
```

(3) 修改 goods.html , 修改列表

```
<td>{{itemCatList[entity.category1Id]}}</td>  
  
<td>{{itemCatList[entity.category2Id]}}</td>  
  
<td>{{itemCatList[entity.category3Id]}}</td>
```

1.5 条件查询

根据状态和商品名称进行查询

修改 goods.html

```
<div class="has-feedback">  
  
    状态: <select ng-model="searchEntity.auditStatus">  
  
        <option value="">全部</option>  
  
        <option value="0">未审核</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 ng-model="searchEntity.goodsName">

<button class="btn btn-default" ng-click="reloadList()">查询</button>

</div>
```

2.商家后台-商品管理【商品修改】

2.1 需求分析

在商品列表页面点击修改，进入商品编辑页面，并传递参数商品 ID，商品编辑页面接受该参数后从数据库中读取商品信息，用户修改后保存信息。

2.2 基本信息读取

我们首选读取商品分类、商品名称、品牌，副标题，价格等信息

商品基本信息	商品图片	扩展属性	查询属性	规格
商品分类	数码	时尚影音	MP3/MP4	模板ID:11
商品名称	蓝魔303			
品牌	三星			
副标题	副标题			
价格	¥	4		

2.2.1 后端代码

(1) 修改 pinyougou-sellergoods-interface 的 GoodsService.java

```
/**
 * 根据 ID 获取实体
 * @param id
```




```
* @return  
  
*/  
  
public Goods findOne(Long id);
```

(2) 修改 pinyougou-sellergoods-service 的 GoodsServiceImpl.java

```
@Override  
  
public Goods findOne(Long id) {  
  
    Goods goods=new Goods();  
  
    TbGoods tbGoods = goodsMapper.selectByPrimaryKey(id);  
  
    goods.setGoods(tbGoods);  
  
    TbGoodsDesc tbGoodsDesc = goodsDescMapper.selectByPrimaryKey(id);  
  
    goods.setGoodsDesc(tbGoodsDesc);  
  
    return goods;  
  
}
```

(3) 修改 pinyougou-shop-web（和 pinyougou-manager-web）的 GoodsController.java

```
/**  
  
 * 获取实体  
  
 * @param id  
  
 * @return  
  
 */  
  
@RequestMapping("/findOne")  
  
public Goods findOne(Long id){  
  
    return goodsService.findOne(id);  
  
}
```



```
}
```

2.2.2 前端代码

(1) 在 goodsController 中引入\$location 服务

```
//商品控制层（商家后台）

app.controller('goodsController',function($scope,$controller,$location,goodsService
,uploadService,item_catService,type_templateService){

.....
```

(2) 修改 goodsController 添加代码:

```
//查询实体

$scope.findOne=function(){

    var id= $location.search()['id'];//获取参数值

    if(id==null){

        return ;

    }

    goodsService.findOne(id).success(

        function(response){

            $scope.entity= response;

        }

    );

}
```

在 goods_edit.html 页面上添加指令

```
<body class="hold-transition skin-red sidebar-mini" ng-app="pinyougou"
ng-controller="goodsController" ng-init="selectItemCat1List();findOne()">
```

测试:



地址栏输入

http://localhost:9102/admin/goods_edit.html?id=149187842867969

注意：?前要加#，则是 angularJS 的地址路由的书写形式

2.3 读取商品介绍（富文本编辑器）

修改前端代码 goodsController

```
// 查询实体

$scope.findOne=function(){

    .....

    goodsService.findOne(id).success(

        function(response){

            $scope.entity= response;

            //向富文本编辑器添加商品介绍

            editor.html($scope.entity.goodsDesc.introduction);

        }

    );

}
```



2.4 显示商品图片列表

修改 goodsController.js ，在 dataLogic 方法添加代码，将图片列表由字符串转换为 json 集合对象

```
// 查询实体

$scope.findOne=function(){

    .....

    //如果有 ID,则查询实体

    goodsService.findOne(id).success(

        function(response){

            $scope.entity= response;

            //向富文本编辑器添加商品介绍

            editor.html($scope.entity.goodsDesc.introduction);

            //显示图片列表

            $scope.entity.goodsDesc.itemImages=

                JSON.parse($scope.entity.goodsDesc.itemImages);

        }

    );

}
```

2.5 读取商品扩展属性

修改 goodsController.js

```
// 查询实体

$scope.findOne=function(){
```



```

.....

goodsService.findOne(id).success(

    function(response){

        .....

        //显示扩展属性

        $scope.entity.goodsDesc.customAttributeItems=
JSON.parse($scope.entity.goodsDesc.customAttributeItems);

    }

);

}

```

经过测试，我们发现扩展属性值并没有读取出来，这是因为与下列代码发生冲突

```

$scope.$watch('entity.goods.typeTemplateId',function(newValue,oldValue){

    .....

    $scope.entity.goodsDesc.customAttributeItems =
        JSON.parse($scope.typeTemplate.customAttributeItems); //扩展属性

}

```

我们读取出来的值被覆盖了，我们需要改写代码，添加判断，当用户没有传递 id 参数时再执行此逻辑

```

//监控模板 ID ， 读取品牌列表

$scope.$watch('entity.goods.typeTemplateId',function(newValue,oldValue){

    //读取品牌列表和扩展属性

    typeTemplateService.findOne(newValue).success(

        function(response){

            .....


```



```
//如果没有 ID，则加载模板中的扩展数据

if($location.search()['id']==null){

    $scope.entity.goodsDesc.customAttributeItems =
JSON.parse($scope.typeTemplate.customAttributeItems);//扩展属性

}

}

);

.....

});
```

2.6 读取商品规格属性

修改 goodsController

```
//查询实体

$scope.findOne=function(){

    .....

    goodsService.findOne(id).success(

        function(response){

            $scope.entity= response;

            editor.html($scope.entity.goodsDesc.introduction);//商品介绍

            $scope.entity.goodsDesc.itemImages=
JSON.parse($scope.entity.goodsDesc.itemImages);//图片列表

            //扩展属性列表

            $scope.entity.goodsDesc.customAttributeItems
=JSON.parse($scope.entity.goodsDesc.customAttributeItems);

            //规格
```



```

$scope.entity.goodsDesc.specificationItems=JSON.parse($scope.entity.goodsDesc.s
pecificationItems);

    }

    );

}

```

```

//根据规格名称和选项名称返回是否被勾选

$scope.checkAttributeValue=function(specName,optionName){

    var items= $scope.entity.goodsDesc.specificationItems;

    var object= $scope.searchObjectByKey(items,'attributeName',specName);

    if(object==null){

        return false;

    }else{

        if(object.attributeValue.indexOf(optionName)>=0){

            return true;

        }else{

            return false;

        }

    }

}

}

```

修改页面上规格面板的复选框,运用 ng-checked 指令控制复选框的勾选状态

```

<input type="checkbox"
    ng-click="updateSpecAttribute($event,pojo.text,p.optionName);createSKUTable()"
    ng-checked="checkAttributeValue(pojo.text,p.optionName)">{{p.optionName}}

```

2.7 读取 SKU 数据

显示 SKU 商品列表，并自动读取价格、库存等数据加载到列表中

屏幕尺寸	网络制式	服装尺码	价格	库存	是否启用	是否默认
4.5	电信2G	大号	<input type="text" value="300"/>	<input type="text" value="99999"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4.5	电信2G	小号	<input type="text" value="100"/>	<input type="text" value="99"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5.0	电信2G	大号	<input type="text" value="230"/>	<input type="text" value="8888"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.0	电信2G	小号	<input type="text" value="260"/>	<input type="text" value="6666"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.7.1 后端代码

在 GoodsServiceImpl 的 findOne 方法中加载 SKU 商品数据

```
//查询 SKU 商品列表

TbItemExample example=new TbItemExample();

com.pinyougou.pojo.TbItemExample.Criteria criteria =
example.createCriteria();

criteria.andGoodsIdEqualTo(id);//查询条件: 商品 ID

List<TbItem> itemList = itemMapper.selectByExample(example);

goods.setItemList(itemList);
```

2.7.2 前端代码

在 goodsController.js 修改 findOne 方法的代码

```
//查询实体

$scope.findOne=function(){

    .....

    goodsService.findOne(id).success(

        function(response){
```




```
        $scope.entity= response;

        .....

        //SKU 列表规格列转换

        for( var i=0;i<$scope.entity.itemList.length;i++ ){

            $scope.entity.itemList[i].spec =

                JSON.parse( $scope.entity.itemList[i].spec);

        }

    }

    );

}
```

2.8 保存数据

2.8.1 后端代码

修改 pinyougou-sellergoods-interface 的 GoodsService.java

```
public void update(Goods goods);
```

修改 pinyougou-sellergoods-service 的 GoodsServiceImpl,将 SKU 列表插入的代码提取出来,封装到私有方法中

```
/**

 * 插入 SKU 列表数据

 * @param goods

 */

private void saveItemList(Goods goods){

    if("1".equals(goods.getGoods().getIsEnableSpec())){
```



```
        for(TbItem item :goods.getItemList()){

            .....中间代码略

        }

    }else{

        TbItem item=new TbItem();

        .....中间代码略

        itemMapper.insert(item);

    }

}
```

在 add 方法中调用 此方法，修改如下：

```
public void add(Goods goods) {

    goods.getGoods().setAuditStatus("0");

    goodsMapper.insert(goods.getGoods()); //插入商品表

    goods.getGoodsDesc().setGoodsId(goods.getGoods().getId());

    goodsDescMapper.insert(goods.getGoodsDesc());//插入商品扩展数据

    saveItemList(goods);//插入商品 SKU 列表数据

}
```

怎么样，是不是比原来更加清爽了呢？

接下来，我们修改 update 方法，实现修改

```
public void update(Goods goods){

    goods.getGoods().setAuditStatus("0");//设置未申请状态:如果是经过修改的商品，需要重新
    设置状态

    goodsMapper.updateByPrimaryKey(goods.getGoods());//保存商品表

    goodsDescMapper.updateByPrimaryKey(goods.getGoodsDesc());//保存商品扩展表

}
```



```
//删除原有的 sku 列表数据

TbItemExample example=new TbItemExample();

com.pinyougou.pojo.TbItemExample.Criteria criteria = example.createCriteria();

criteria.andGoodsIdEqualTo(goods.getGoods().getId());

itemMapper.deleteByExample(example);

//添加新的 sku 列表数据

saveItemList(goods);//插入商品 SKU 列表数据

}
```

修改 pinyougou-manager-web 工程的 GoodsController.java

```
@RequestMapping("/update")

public Result update(@RequestBody Goods goods){

    .....

}
```

修改 pinyougou-shop-web 工程的 GoodsController.java

```
/**

 * 修改

 * @param goods

 * @return

 */

@RequestMapping("/update")

public Result update(@RequestBody Goods goods){

    //校验是否是当前商家的 id

    Goods goods2 = goodsService.findOne(goods.getGoods().getId());
```



```
//获取当前登录的商家 ID

String sellerId = SecurityContextHolder.getContext().getAuthentication().getName();

//如果传递过来的商家 ID 并不是当前登录的用户的 ID, 则属于非法操作

if(!goods2.getGoods().getSellerId().equals(sellerId)
|| !goods.getGoods().getSellerId().equals(sellerId) ){

    return new Result(false, "操作非法");

}

try {

    goodsService.update(goods);

    return new Result(true, "修改成功");

} catch (Exception e) {

    e.printStackTrace();

    return new Result(false, "修改失败");

}

}
```

代码解释：出于安全考虑，在商户后台执行的商品修改，必须要校验提交的商品属于该商户

2.8.2 前端代码

(1) 修改 goodsController.js ,新增保存的方法

```
//保存

$scope.save=function(){

    //提取文本编辑器的值

    $scope.entity.goodsDesc.introduction=editor.html();

    var serviceObject;//服务层对象
```



```
if($scope.entity.goods.id!=null){//如果有 ID

    serviceObject=goodsService.update( $scope.entity ); //修改

}else{

    serviceObject=goodsService.add( $scope.entity );//增加

}

serviceObject.success(

    function(response){

        if(response.success){

            alert('保存成功');

            $scope.entity={};

            editor.html("");

        }else{

            alert(response.message);

        }

    }

);

}
```

(2) 修改 goods_edit.html 调用

```
<button class="btn btn-primary" ng-click="save()"><i class="fa fa-save"></i> 保存
</button>
```

2.9 页面跳转

(1) 由商品列表页跳转到商品编辑页



修改 goods.html 表格行的修改按钮

```
<a href="goods_edit.html?id={{entity.id}}" class="btn bg-olive btn-xs">修改</a>
```

(2) 由商品编辑页跳转到商品列表

修改 goods_edit.html 的返回列表按钮

```
<a href="goods.html" class="btn btn-default">返回列表</a>
```

(3) 保存成功后返回列表页面

```
//保存

$scope.save=function(){

    .....

    serviceObject.success(

        function(response){

            if(response.success){

                location.href="goods.html";//跳转到商品列表页

            }else{

                alert(response.message);

            }

        }

    );

}
```



3.运营商后台-商品管理【商品审核】

3.1 待审核商品列表

需求：参照商家后台商品列表。代码：

(1) 修改 pinyougou-manager-web 的 goodsController.js，注入 itemCatService，添加代码

```
$scope.status=['未审核','已审核','审核未通过','关闭'];//商品状态

$scope.itemCatList=[];//商品分类列表

//查询商品分类

$scope.findItemCatList=function(){

    itemCatService.findAll().success(

        function(response){

            for(var i=0;i<response.length;i++){

                $scope.itemCatList[response[i].id]=response[i].name;

            }

        }

    );

}
```

(2) 修改 goods.html,引入 js

```
<script type="text/javascript" src="../plugins/angularjs/angular.min.js"></script>

<!-- 分页组件开始 -->

<script src="../plugins/angularjs/pagination.js"></script>

<link rel="stylesheet" href="../plugins/angularjs/pagination.css">

<!-- 分页组件结束 -->
```



```
<script type="text/javascript" src="../js/base_pagination.js"></script>

<script type="text/javascript" src="../js/service/goodsService.js"></script>

<script type="text/javascript" src="../js/service/itemCatService.js"></script>

<script type="text/javascript" src="../js/controller/baseController.js"></script>

<script type="text/javascript" src="../js/controller/goodsController.js"></script>
```

(3) 指令，完成初始调用

```
<body class="hold-transition skin-red sidebar-mini" ng-app="pinyougou"
ng-controller="goodsController"
ng-init="searchEntity={auditStatus:'0'};findItemCatList()">
```

(4) 循环列表

```
<tr ng-repeat="entity in list">

  <td><input type="checkbox"></td>

  <td>{{entity.id}}</td>

  <td>{{entity.goodsName}}</td>

  <td>{{entity.price}}</td>

  <td>{{itemCatList[entity.category1Id]}}</td>

  <td>{{itemCatList[entity.category2Id]}}</td>

  <td>{{itemCatList[entity.category3Id]}}</td>

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

  <td class="text-center"> </td>

</tr>
```

(5) 分页控件

```
<tm-pagination conf="paginationConf"></tm-pagination>
```




3.2 商品详情展示(学员实现)

需求：点击列表右侧的“详情”按钮，弹出窗口显示商品信息。代码略。

3.3 商品审核与驳回

需求：商品审核的状态值为 1，驳回的状态值为 2。用户在列表选中 ID 后，点击审核或驳回，修改商品状态，并刷新列表。

3.3.1 后端代码

(1) 在 pinyougou-sellergoods-interface 的 GoodsService.java 新增方法定义

```
/**
 * 批量修改状态
 * @param ids
 * @param status
 */
public void updateStatus(Long []ids,String status);
```

(2) 在 pinyougou-sellergoods-service 的 GoodsServiceImpl.java 实现该方法

```
public void updateStatus(Long[] ids, String status) {
    for(Long id:ids){
        TbGoods goods = goodsMapper.selectByPrimaryKey(id);
        goods.setAuditStatus(status);
        goodsMapper.updateByPrimaryKey(goods);
    }
}
```

(3) 在 pinyougou-shop-web 的 GoodsController.java 新增方法



```
/**
 * 更新状态
 * @param ids
 * @param status
 */

@RequestMapping("/updateStatus")

public Result updateStatus(Long[] ids, String status){

    try {

        goodsService.updateStatus(ids, status);

        return new Result(true, "成功");

    } catch (Exception e) {

        e.printStackTrace();

        return new Result(false, "失败");

    }

}
```

3.3.2 前端代码

(1) 修改 pinyougou-manager-web 的 goodsService.js ，增加方法

```
//更改状态

this.updateStatus=function(ids,status){

    return $http.get('../goods/updateStatus.do?ids='+ids+"&status="+status);

}
```

(2) 修改 pinyougou-manager-web 的 goodsController.js ，增加方法



```
//更改状态

$scope.updateStatus=function(status){

    goodsService.updateStatus($scope.selectIds,status).success(

        function(response){

            if(response.success){//成功

                $scope.reloadList();//刷新列表

                $scope.selectIds=[];//清空 ID 集合

            }else{

                alert(response.message);

            }

        }

    );

}
```

(3) 修改 pinyougou-manager-web 的 goods.html 页面，为复选框绑定事件指令

```
<input type="checkbox" ng-click="updateSelection($event,entity.id)" >
```

(4) 修改页面上的审核通过和驳回按钮

```
<button type="button" class="btn btn-default" title="审核通过"
ng-click="updateStatus('1')"><i class="fa fa-check"></i> 审核通过</button>

<button type="button" class="btn btn-default" title="驳 回"
ng-click="updateStatus('2')"><i class="fa fa-ban"></i> 驳回</button>
```



4.运营商后台-商品管理【商品删除】

4.1 需求分析

我们为商品管理提供商品删除功能，用户选中部分商品，点击删除按钮即可实现商品删除。注意，这里的删除并非是物理删除，而是修改 tb_goods 表的 is_delete 字段为 1，我们可以称之为“逻辑删除”

4.2 逻辑删除的实现

4.2.1 后端代码

修改 pinyougou-sellergoods-service 工程的 GoodsServiceImpl.java 的 delete 方法

```
/**
 * 批量删除
 */
@Override
public void delete(Long[] ids) {
    for(Long id:ids){
        TbGoods goods = goodsMapper.selectByPrimaryKey(id);
        goods.setIsDelete("1");
        goodsMapper.updateByPrimaryKey(goods);
    }
}
```

4.2.2 前端代码

修改 pinyougou-manager-web 的 goods.html 上的删除按钮

```
<button type="button" class="btn btn-default" title="删除" ng-click="delete()"><i>
```



```
class="fa fa-trash-o"></i> 删除</button>
```

4.3 排除已删除记录

修改 pinyougou-sellergoods-service 工程 GoodsServiceImpl.java 的 findPage 方法，添加以下代码：

```
criteria.andIsDeleteIsNull();//非删除状态
```

5.商家后台-【商品上下架】（学员实现）

5.1 需求分析

什么是商品上下架？其实上下架也是商品的一个状态，但是不同于审核状态。审核状态的控制权在运营商手中，而上下架的控制权在商户手中。商户可以随时将一个已审核状态的商品上架或下架。上架表示正常销售，而下架则表示暂停销售。

5.2 实现思路提示

其实商品的上下架就是对上下架状态的修改。字段为 tb_goods 表的 is_marketable 字段。1 表示上架、0 表示下架。

6.注解式事务配置

6.1 事务异常测试

我们修改 pinyougou-sellergoods-service 工程 GoodsServiceImpl.java 的 add 方法

```
/**  
  
 * 增加  
  
 */  
  
@Override
```



```
public void add(Goods goods) {  
  
    goods.getGoods().setAuditStatus("0");  
  
    goodsMapper.insert(goods.getGoods()); //插入商品表  
  
    int x=1/0;  
  
    goods.getGoodsDesc().setGoodsId(goods.getGoods().getId());  
  
    goodsDescMapper.insert(goods.getGoodsDesc()); //插入商品扩展数据  
  
    saveItemList(goods); //插入商品 SKU 列表数据  
  
}
```

在插入商品表后，人为制造一个异常。我们运行程序，新增商品数据，观察运行结果。

通过观察，我们发现，程序发生异常，商品表仍然会存储记录，这是不符合我们要求的。这是因为我们目前的系统还没有配置事务。

6.2 注解式事务解决方案

6.2.1 配置文件

在 pinyougou-sellergoods-service 工程的 spring 目录下创建 applicationContext-tx.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:p="http://www.springframework.org/schema/p"  
  
    xmlns:context="http://www.springframework.org/schema/context"  
  
    xmlns:tx="http://www.springframework.org/schema/tx"  
  
    xmlns:mvc="http://www.springframework.org/schema/mvc"  
  
    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://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd

    http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

    <!-- 事务管理器 -->

    <bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

        <property name="dataSource" ref="dataSource" />

    </bean>

    <!-- 开启事务控制的注解支持 -->

    <tx:annotation-driven transaction-manager="transactionManager"/>

</beans>
```

6.2.2 在方法上添加注解

```
/**
 * 服务实现层
 *
 * @author Administrator
 *
 */
@Service
@Transactional
public class GoodsServiceImpl implements GoodsService{
    .....
}
```



经过测试，我们发现，系统发生异常，商品表不会新增记录，事务配置成功。

删除掉测试代码 `int x=1/0`

我们需要将所有涉及多表操作的服务类添加事务注解，例如 `SpecificationServiceImpl` 类