**品优购电商系统开发**

**第3章**

**品牌管理&前端分层&规格管理**

传智播客.黑马程序员

# 课程目标

目标1：实现品牌条件查询的功能

目标2：实现删除品牌的功能

目标3：理解前端分层开发

目标4：实现控制器继承

目标5：完成规格管理功能

# 1.品牌管理【分页查询】

## 1.1需求分析

在品牌管理下方放置分页栏，实现分页功能：



## **1.2前端代码**

### **1.2.1页面**

在pinyougou-manager-web/src/main/webapp/admin/brand.htm引入分页模块

|  |
| --- |
| *<!-- 分页模块开始 -->*  <**link rel="stylesheet" href="/plugins/angularjs/pagination.css"**/><**script src="/plugins/angularjs/pagination.js"**></**script**> *<!-- 分页模块结束 -->* |

在构建app模块时引入pagination模块：

|  |
| --- |
| **var *app*** = **angular**.**module**(**'pinyougou'**,[**'pagination'**]); |

brand.html页面的表格table结束标签下放置分页指令：

|  |
| --- |
| *<!-- 分页指令 -->* <**tm-pagination conf="paginationConf"**/> |

### **1.2.2 JS代码**

在pinyougou-manager-web/src/main/webapp/admin/brand.html中添加分页相关js代码：

|  |
| --- |
| */\*\* 分页指令配置信息对象 \*/* **$scope**.**paginationConf** = {  **currentPage**: 1, *// 当前页码* **totalItems**: 0, *// 总记录数* **itemsPerPage**: 10, *// 每页显示的记录数* **perPageOptions**: [10, 20, 30], *// 页码下拉列表框* onChange: **function**(){ *// 改变事件* **$scope**.reload(); *//重新加载* } }; */\*\* 重新加载列表数据 \*/* **$scope**.reload = **function**(){  */\*\* 切换页码 \*/* **$scope**.search(**$scope**.**paginationConf**.**currentPage**,  **$scope**.**paginationConf**.**itemsPerPage**); }; */\*\* 分页查询品牌信息 \*/* **$scope**.search = **function**(page, rows){  */\*\* 发送异步请求分页查询品牌数据 \*/* $http.get(**'/brand/findByPage?page='**+ page +**'&rows='** + rows)  .then(**function**(response){  **$scope**.**dataList** = response.data.rows;  */\*\* 更新总记录数 \*/* **$scope**.**paginationConf**.**totalItems** = response.data.total;  }); }; |

paginationConf 变量各属性的意义：

1. currentPage：当前页码
2. totalItems: 总记录数
3. itemsPerPage: 每页显示的记录数
4. perPageOptions：页码下拉列表框
5. onChange：改变页码时触发事件

注意：在页面body元素中去掉ng-init指令的调用及修改saveOrUpdate方法

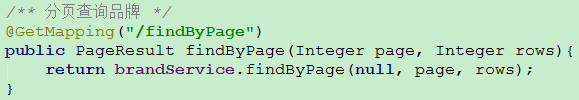


## **1.3后端代码**

### **1.3.1 控制器层**

在pinyougou-manager-web工程的src/main/java/

com.pinyougou.manager.controller的BrandController.java新增方法：



### **1.3.2 分页结果实体**

在pinyougou-common工程中创建com.pinyougou.common.pojo包，用于存放通用实体类，创建PageResult.java

|  |
| --- |
| */\*\* 分页结果实体 \*/* **public class** PageResult **implements** Serializable {  */\*\* 总记录数 \*/* **private long total**;  */\*\* 分页数据 \*/* **private** List<?> **rows**;  **public** PageResult(){}  **public** PageResult(**long** total, List<?> rows) {  **this**.**total** = total;  **this**.**rows** = rows;  }  **public long** getTotal() {  **return total**;  }  **public void** setTotal(**long** total) {  **this**.**total** = total;  }  **public** List<?> getRows() {  **return rows**;  }  **public void** setRows(List<?> rows) {  **this**.**rows** = rows;  } } |

### **1.3.3 服务接口层**

在pinyougou-interface的pom.xml配置依赖：

|  |
| --- |
| *<!-- pinyougou-common -->* <**dependency**>  <**groupId**>com.pinyougou</**groupId**>  <**artifactId**>pinyougou-common</**artifactId**>  <**version**>${project.version}</**version**> </**dependency**> |

pinyougou-interface/src/main/java/com.pinyougou.service/BrandService.java修改findByPage方法：



### **1.3.4 服务实现层**

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl的BrandServiceImpl.java中实现该方法：

|  |
| --- |
| */\*\* 分页查询品牌 \*/* @Override **public** PageResult findByPage(Brand brand, **int** page, **int** rows) {  *// 开始分页* PageInfo<Brand> pageInfo = PageHelper.*startPage*(page, rows)  .doSelectPageInfo(**new** ISelect() {  @Override  **public void** doSelect() {  **brandMapper**.selectAll();  }  });  **return new** PageResult(pageInfo.getTotal(),pageInfo.getList()); } |

注意：PageHelper为MyBatis分页插件。

# 2.品牌管理【多条件分页查询】

## 2.1需求分析

实现品牌条件查询功能，输入品牌名称、首字母后查询，并分页。

## 2.2前端代码

brand.html添加搜索条件数据绑定：

|  |
| --- |
| <**div class="box-tools pull-right"**>  <**div class="has-feedback"**>  *<!-- 条件查询区-->* <**table**>  <**tr**>  <**td**><**input class="form-control"   ng-model="searchEntity.name" placeholder="品牌名称"**>  </**td**>  <**td**><**input class="form-control"   ng-model="searchEntity.firstChar" placeholder="首字母"**>  </**td**>  <**td**>  <**button type="button" class="btn btn-default"   ng-click="reload()"**>  <**i class="fa fa-trash-o"**></**i**> 查询  </**button**>  </**td**>  </**tr**>  </**table**>  </**div**> </**div**><**br**/> |

在pinyougou-manager-web/src/main/webapp/admin/brand.html页面，修改$scope.search方法：

|  |
| --- |
| */\*\* 定义查询条件对象 \*/* $scope.**searchEntity** = {}; *// 多条件分页查询品牌* $scope.search = **function**(page, rows){  $http({  **url** : **"/brand/findByPage?page="** + page + **"&rows="** + rows,  **method** : **'get'**,  **params** : $scope.**searchEntity** }).then(**function**(response){  $scope.**dataList** = response.**data**.rows;  */\*\* 更新总记录数 \*/* $scope.**paginationConf**.**totalItems** = response.**data**.total;  }); }; |

## 2.3后端代码

### **2.3.1控制层**

修改pinyougou-manager-web/src/main/java/

com.pinyougou.manager.controller的BrandController.java分页方法：

|  |
| --- |
| */\*\* 多条件分页查询品牌 \*/* @GetMapping(**"/findByPage"**) **public** PageResult findByPage(Brand brand, Integer page, Integer rows){  */\*\* GET请求中文转码 \*/* **if** (brand != **null** && StringUtils.*isNoneBlank*(brand.getName())) {  **try** {  brand.setName(**new** String(brand.getName()  .getBytes(**"ISO8859-1"**), **"UTF-8"**));  }**catch** (Exception ex){  ex.printStackTrace();  }  }  **return brandService**.findByPage(brand, page, rows); } |

### **2.3.2服务实现层**

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl的BrandServiceImpl.java分页实现方法：

|  |
| --- |
| @Override **public** PageResult findByPage(Brand brand, **int** page, **int** rows) {  **try** {  *// 开始分页* PageInfo<Brand> pageInfo = PageHelper.*startPage*(page, rows)  .doSelectPageInfo(**new** ISelect() {  @Override  **public void** doSelect() {  **brandMapper.findAll(brand);**  }  });  **return new** PageResult(pageInfo.getTotal(), pageInfo.getList());  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } |

### **2.3.3数据访问层**

pinyougou-mapper/src/main/java/com.pinyougou.mapper/BrandMapper.java增加方法：



修改pinyougou-mapper/src/main/resources/mappers/BrandMapper.xml

|  |
| --- |
| *<!-- 多条件查询品牌 -->* <**select id="findAll" resultType="Brand"**>  select \* from tb\_brand  <**where**>  *<!-- 品牌名称 -->* <**if test="name != null and name !=''"**>  name like CONCAT('%',#{name},'%')  </**if**>  *<!-- 品牌首字母 -->* <**if test="firstChar != null and firstChar != ''"**>  and first\_char = #{firstChar}  </**if**>  </**where**>  order by id asc </**select**> |

最终效果：



# 3.品牌管理【删除品牌】

## 3.1 需求分析

brand.html页面点击列表前的复选框，点击删除按钮，删除选中的品牌。



## 3.2 前端代码

### **3.2.1页面**

主要思路：我们需要定义一个用于存储选中ID的数组，当我们点击复选框后判断是选择还是取消选择，如果是选择就加到数组中，如果是取消选择就从数组中移除。在点击删除按钮时需要用到这个存储了ID的数组。

brand.html页面为复选框绑定点击事件：

|  |
| --- |
| <**input type="checkbox"  ng-click="updateSelection($event,entity.id);"**/> |

brand.html页面为删除按钮绑定点击事件：

|  |
| --- |
| <**button type="button" class="btn btn-default" title="删除"  ng-click="delete();"**>  <**i class="fa fa-trash-o"**></**i**> 删除 </**button**> |

### **3.2.2 JS**

这里我们补充一下JS的关于数组操作的知识：

1. 数组的push方法：向数组中添加元素
2. 数组的splice方法：从数组的指定位置移除指定个数的元素 ，参数1为位置 ，参数2为移除的个数
3. 复选框的checked属性：用于判断是否被选中

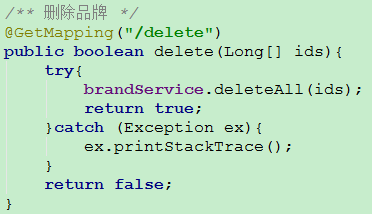
在pinyougou-manager-web/src/main/webapp/admin/brand.html页面添加JS代码：

|  |
| --- |
| */\*\* 定义选中的ids数组 \*/* $scope.**ids** = []; */\*\* 为复选框绑定点击事件 \*/* $scope.updateSelection = **function**($event, id) {  */\*\* 如果是被选中,则增加到数组 \*/* **if**($event.**target**.checked){  $scope.**ids**.push(id);  }**else**{  **var** idx = $scope.**ids**.indexOf(id);  */\*\* 删除数组中的元素 \*/* $scope.**ids**.splice(idx, 1);  } }; */\*\* 批量删除 \*/* $scope.delete = **function**(){  **if** ($scope.**ids**.length > 0){  */\*\* 发送异步请求 \*/* $http.get(**'/brand/delete?ids='** + $scope.**ids**).then(  **function**(response){  **if**(response.data){  */\*\* 重新加载品牌数据 \*/* $scope.reload();  }  }  );  }**else**{  *alert*(**"请选择要删除的品牌！"**);  } }; |

## 3.3 后端代码

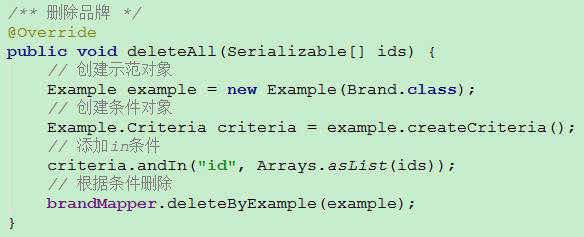
### **3.3.1控制层**

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller/BrandController.java：



### **3.3.2服务实现层**

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



# 4.前端分层开发【MVC分层】

## 4.1需求分析

我们完成了品牌管理的增删改查功能。但是我们看代码，JS和html都放在一起，并不利于我们后期的维护。我们可以在前端代码中也运用MVC的设计模式，将代码进行分离，提高程序的可维护性。



## 4.2 MVC分层

### **4.2.1 前端基础层**

在pinyougou-manager-web/src/main/webapp/js目录下创建base.js：

|  |
| --- |
| */\*\* 定义基础模块(不带分页模块) \*/* **var *app*** = **angular**.module(**'pinyougou'**,[]); |

再创建base-pagination.js：

|  |
| --- |
| */\*\* 定义基础模块(带分页模块) \*/* **var *app*** = **angular**.module(**'pinyougou'**,[**'pagination'**]); |

**说明：一个用于不需要分页功能的页面，一个用于需要分页功能的页面。**

### **4.2.2 前端服务层**

A.在pinyougou-manager-web/src/main/webapp/js下面创建service文件夹。

1. 拷贝【资料\baseService.js】到pinyougou-manager-web

/src/main/webapp/js/service/文件夹下面。

C.在pinyougou-manager-web/src/main/webapp/js/service/下面创建品牌服务层brandService.js：

|  |
| --- |
| */\*\* 品牌服务层 \*/* ***app***.**service**(**"brandService"**, **function**($http){ }); |

### **4.2.3 前端控制层**

在pinyougou-manager-web的src/main/webapp/js文件夹下创建controller文件夹，再创建brandController.js：

|  |
| --- |
| */\*\* 定义品牌控制器层 \*/* ***app***.controller(**'brandController'**, **function**($scope, **baseService**){  */\*\* 定义分页配置信息对象 \*/* $scope.**paginationConf** = {  **currentPage** : 1, *// 当前页码* **totalItems** : 0, *// 总记录数* **itemsPerPage** : 10, *// 每页显示的记录数* **perPageOptions** : [10,15,20], *// 页码下拉列表* onChange : **function**() { *// 改变事件* $scope.reload();  }  };  */\*\* 当下拉列表页码发生改变，重新加载数据 \*/* $scope.reload = **function**(){  $scope.search($scope.**paginationConf**.**currentPage**,  $scope.**paginationConf**.**itemsPerPage**);  };  */\*\* 定义搜索对象 \*/* $scope.**searchEntity** = {};  */\*\* 分页查询品牌 \*/* $scope.search = **function**(page, rows){  baseService.findByPage(**"/brand/findByPage"**, page,  rows, $scope.**searchEntity**)  .then(**function**(response){  $scope.**dataList** = response.**data**.rows;  */\*\* 更新总记录数 \*/* $scope.**paginationConf**.**totalItems** = response.**data**.total;  });  };  */\*\* 添加或修改品牌 \*/* $scope.saveOrUpdate = **function**(){  **var** url = **"save"**;  **if** ($scope.**entity**.**id**){  url = **"update"**;  }  */\*\* 发送post请求添加品牌 \*/* baseService.sendPost(**"/brand/"** + url, $scope.**entity**)  .then(**function**(response){  **if** (response.**data**){  */\*\* 重新加载品牌数据 \*/* $scope.reload();  }**else**{  *alert*(**"操作失败！"**);  }  });  };  */\*\* 显示修改 \*/* $scope.show = **function**(entity){  *// 把entity的json对象转化成一个新的json对象* $scope.**entity** = ***JSON***.parse(***JSON***.stringify(entity));  };   */\*\* 定义ids数组封装删除的id \*/* $scope.**ids** = [];  */\*\* 定义checkbox点击事件函数 \*/* $scope.updateSelection = **function**($event, id){  */\*\* 判断checkbox是否选中 \*/* **if** ($event.**target**.checked){  $scope.**ids**.push(id);  }**else**{  */\*\* 从数组中移除 \*/* **var** idx = $scope.**ids**.indexOf(id);  $scope.**ids**.splice(idx, 1);  }  };  */\*\* 批量删除 \*/* $scope.delete = **function**(){  **if** ($scope.**ids**.length > 0){  baseService.deleteById(**"/brand/delete"**, $scope.**ids**)  .then(**function**(response){  **if** (response.**data**){  $scope.reload();  }**else**{  *alert*(**"删除失败！"**);  }  });  }  }; }); |

### **4.2.4 修改页面**

去掉pinyougou-manager-web/src/main/webapp/admin/brand.html原来的js代码，引入刚才我们创建的js：

|  |
| --- |
| <**script src="/js/base-pagination.js"**></**script**> <**script src="/js/service/baseService.js"**></**script**> <**script src="/js/controller/brandController.js"**></**script**> |

# 5.控制器继承【代码重用】

## 5.1需求分析

有些功能是每个页面都有可能用到的，比如分页，复选等等，如果我们再开发另一个功能，还需要重复编写。怎么能让这些通用的功能只写一次呢？我们通过继承的方式来实现。

## 5.2前端代码

### **5.2.1创建父控制器**

在pinyougou-manager-web/src/main/webapp/js/controller/目录下创建baseController.js

|  |
| --- |
| */\*\* 定义基础控制器层 \*/* ***app***.controller(**'baseController'**,**function**($scope){  */\*\* 定义分页配置信息对象 \*/* $scope.**paginationConf** = {  **currentPage** : 1, *// 当前页码* **totalItems** : 0, *// 总记录数* **itemsPerPage** : 10, *// 每页显示的记录数* **perPageOptions** : [10,15,20], *// 页码下拉列表* onChange : **function**() { *// 改变事件* $scope.reload();  }  };  */\*\* 当下拉列表页码发生改变，重新加载数据 \*/* $scope.reload = **function**(){  $scope.search($scope.**paginationConf**.**currentPage**,  $scope.**paginationConf**.**itemsPerPage**);  };  */\*\* 定义ids数组封装删除的id \*/* $scope.**ids** = [];  */\*\* 定义checkbox点击事件函数 \*/* $scope.updateSelection = **function**($event, id){  */\*\* 判断checkbox是否选中 \*/* **if** ($event.**target**.checked){  $scope.**ids**.push(id);  }**else**{  */\*\* 从数组中移除 \*/* **var** idx = $scope.**ids**.indexOf(id);  $scope.**ids**.splice(idx, 1);  }  }; }); |

### **5.2.2修改品牌控制器层**

pinyougou-manager-web/src/main/webapp/js/controller/brandController.js：

|  |
| --- |
| */\*\* 品牌控制器层 \*/* ***app***.controller(**"brandController"**, **function**($scope, $controller, baseService){  */\*\* 指定继承baseController \*/* **$controller('baseController',{$scope:$scope});**  */\*\* 读取品牌数据绑定到表格中 \*/* $scope.findAll = **function**(){  */\*\* 调用服务层查询所有品牌数据 \*/* baseService.sendGet(**"/brand/findAll"**).then(**function**(response){  $scope.**dataList** = response.**data**;  });  };  */\*\* 定义搜索对象 \*/* $scope.**searchEntity** = {};  */\*\* 分页查询品牌信息 \*/* $scope.search = **function**(page, rows){  */\*\* 调用服务层分页查询品牌数据 \*/* baseService.findByPage(**"/brand/findByPage"**,page,  rows,$scope.**searchEntity**)  .then(**function**(response){  $scope.**dataList** = response.**data**.rows;  */\*\* 更新总记录数 \*/* $scope.**paginationConf**.**totalItems** = response.**data**.**total**;  });  };  */\*\* 添加与修改品牌 \*/* $scope.saveOrUpdate = **function**(){  */\*\* 定义请求URL \*/* **var** url = **"save"**; *// 添加品牌* **if** ($scope.**entity**.**id**){  url = **"update"**; *// 修改品牌* }  */\*\* 调用服务层 \*/* baseService.sendPost(**"/brand/"** + url, $scope.**entity**)  .then(**function**(response){  **if** (response.**data**){  */\*\* 重新加载品牌数据 \*/* $scope.reload();  }**else**{  *alert*(**"操作失败"**);  }  });  };  */\*\* 显示修改，为修改表单绑定当行数据 \*/* $scope.show = **function**(entity){  *// 把entity的json对象转化成一个新的json对象* $scope.**entity** = ***JSON***.parse(***JSON***.stringify(entity));  };  */\*\* 批量删除品牌 \*/* $scope.delete = **function**(){  **if** ($scope.**ids**.length > 0){  */\*\* 调用服务层 \*/* baseService.deleteById(**"/brand/delete"**, $scope.**ids**).then(  **function**(response){  **if**(response.**data**){  */\*\* 重新加载品牌数据 \*/* $scope.reload();  }  }  );  }**else**{  *alert*(**"请选择要删除的品牌！"**);  }  }; }); |

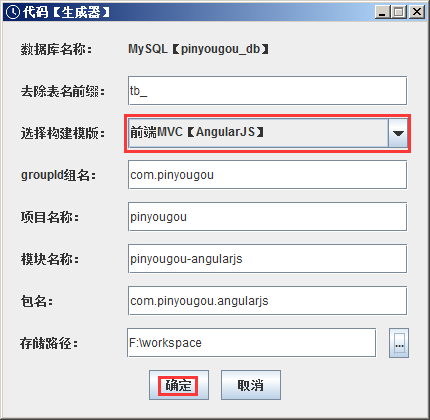
**说明：$controller也是AngularJS提供的一个服务，可以实现伪继承，实际上就是与baseController共享$scope。**

### **5.2.3修改页面**

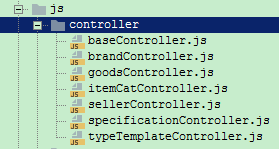
修改pinyougou-manager-web/src/main/webapp/admin/brand.html页面：

|  |
| --- |
| <**script src="/js/base-pagination.js"**></**script**> <**script src="/js/service/baseService.js"**></**script**> <**script src="/js/controller/baseController.js"**></**script**> <**script src="/js/controller/brandController.js"**></**script**> |

# 6.代码准备【快速开发】



拷贝pinyougou\pinyougou-angularjs\src\main\webapp\js录下生成的js到pinyougou-manager-web/src/main/webapp/js/目录下：



**注意：拷贝代码生成器生成的基础层js、服务层js、控制器层js，再次测试品牌模块的所有功能。**

# 规格管理



## 7.1需求及表结构分析

### **7.1.1 需求**

实现规格管理CRUD功能

### **7.1.2 表结构**

**tb\_specification 规格表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 含义 |
| id | bigint |  | 主键 |
| spec\_name | varchar | 255 | 规格名称 |

**tb\_specification\_option 规格选项表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 含义 |
| id | bigint |  | 主键 |
| option\_name | varchar | 200 | 规格选项名称 |
| spec\_id | bigint | 30 | 规格ID |
| orders | int | 11 | 排序 |

## 7.2规格列表

### **7.2.1 引入JS**

修改pinyougou-manager-web/src/main/webapp/admin/specification.html

|  |
| --- |
| <**script src="/plugins/angularjs/angular.min.js"**></**script**> *<!-- 分页模块开始 -->* <**script src="/plugins/angularjs/pagination.js"**></**script**> <**link rel="stylesheet" href="/plugins/angularjs/pagination.css"**/> *<!-- 分页模块结束 -->* <**script src="/js/base-pagination.js"**></**script**> <**script src="/js/service/baseService.js"**></**script**> <**script src="/js/controller/baseController.js"**></**script**> <**script src="/js/controller/specificationController.js"**></**script**> |

### **7.2.2 放置分页指令**

分页指令显示在</table>后面：

|  |
| --- |
| *<!-- 分页指令 -->* <**tm-pagination conf="paginationConf"**/> |

### **7.2.3 指令与表达式**

在body元素指定模块名和控制器名：

|  |
| --- |
| <**body class="hold-transition skin-red sidebar-mini"  ng-app="pinyougou" ng-controller="specificationController"**> |

循环表格行：

|  |
| --- |
| <**tr ng-repeat="entity in dataList"**>  <**td**><**input type="checkbox"**></**td**>  <**td**>{{entity.**id**}}</**td**>  <**td**>{{entity.specName}}</**td**>  <**td class="text-center"**>  <**button type="button" class="btn bg-olive btn-xs"  data-toggle="modal" data-target="#editModal"**>修改  </**button**>  </**td**> </**tr**> |

查询条件、查询按钮绑定点击事件：

|  |
| --- |
| <**div class="has-feedback"**>  <**input ng-model="searchEntity.specName"**  **placeholder="规格名称"**/>  <**button class="btn btn-default" ng-click="reload()"**>查询</**button**> </**div**> |

### **7.2.4 后台代码**

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller包下创建SpecificationController.java：

|  |
| --- |
| @RestController @RequestMapping(**"/specification"**) **public class** SpecificationController {  @Reference(timeout = 10000)  **private** SpecificationService **specificationService**;  */\*\* 多条件分页查询规格 \*/* @GetMapping(**"/findByPage"**)  **public** PageResult findByPage(Specification specification,  Integer page, Integer rows){  */\*\* GET请求中文转码 \*/* **if** (specification != **null** && StringUtils  .*isNoneBlank*(specification.getSpecName())){  **try** {  specification.setSpecName(**new** String(specification  .getSpecName().getBytes(**"ISO8859-1"**),**"UTF-8"**));  } **catch** (UnsupportedEncodingException e) {  e.printStackTrace();  }  }  **return specificationService**.findByPage(specification,  page, rows);  } } |

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



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

|  |
| --- |
| **@Service(interfaceName = "com.pinyougou.service.SpecificationService") @Transactional** **public class** SpecificationServiceImpl **implements** SpecificationService{  */\*\* 注入数据访问接口代理对象 \*/* @Autowired  **private** SpecificationMapper **specificationMapper**;   @Override  **public void** save(Specification specification) {  }  @Override  **public void** update(Specification specification) {  }  @Override  **public void** delete(Serializable id) {  }  @Override  **public void** deleteAll(Serializable[] ids) {  }  @Override  **public** Specification findOne(Serializable id) {  **return null**;  }  @Override  **public** List<Specification> findAll() {  **return null**;  }  */\*\* 多条件分页查询规格 \*/* @Override  **public** PageResult findByPage(Specification specification,  **int** page, **int** rows) {  **try** {  */\*\* 开始分页 \*/* PageInfo<Specification> pageInfo = PageHelper.*startPage*(page,rows)  .doSelectPageInfo(**new** ISelect() {  @Override  **public void** doSelect() {  **specificationMapper.findAll(specification);**  }  });  **return new** PageResult(pageInfo.getTotal(),  pageInfo.getList());  } **catch** (Exception ex) {  **throw new** RuntimeException(ex);  }  } } |

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

|  |
| --- |
| */\*\* 多条件查询规格 \*/* List<Specification> findAll(Specification specification); |

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

|  |
| --- |
| *<!-- 多条件查询规格 -->* <**select id="findAll" resultType="Specification"**>  select *\** from tb\_specification  <**where**>  *<!-- 品牌名称 -->* <**if test="specName != null and specName !=''"**>  spec\_name like CONCAT('%',#{specName},'%')  </**if**>  </**where**>  order by id asc </**select**> |

## 7.3新增规格

### **7.3.1需求分析**

点击“新增规格选项”按钮，新增一行。

点击“删除”按钮，删除一行。

点击“保存”按钮，需将数据保存到规格表与规格选项表。



### **7.3.2 新增行的实现**

pinyougou-manager-web/src/main/webapp/admin/specification.html ”新增规格选项” 按钮绑定点击事件：

|  |
| --- |
| *<!-- 规格选项 -->* <**div class="btn-group"**>  <**button type="button" class="btn btn-default" title="新建"  ng-click="addTableRow()"**>  <**i class="fa fa-file-o"**></**i**>新增规格选项</**button**> </**div**> |

pinyougou-manager-web/src/main/webapp/js/controller/specificationController.js 新增以下代码：

|  |
| --- |
| */\*\* 新增规格选项行 \*/* $scope.addTableRow = **function**(){  $scope.**entity**.specificationOptions.push({}); }; |

specification.html循环列表行，绑定表格内的文本框：

|  |
| --- |
| <**tr ng-repeat="so in entity.specificationOptions"**>  <**td**>  <**input ng-model="so.optionName" class="form-control" placeholder="规格选项"**>  </**td**>  <**td**>  <**input ng-model="so.orders" class="form-control" placeholder="排序"**>  </**td**>  <**td**>  <**button type="button" class="btn btn-default" title="删除"** ><**i class="fa fa-trash-o"**></**i**> 删除</**button**>  </**td**> </**tr**> |

注意：要修改specification.html “新建”按钮，弹出窗口时对entity进行初始化，否则向数组添加数据时会报错！

|  |
| --- |
| <**button type="button" class="btn btn-default" title="新建"  data-toggle="modal" data-target="#editModal"  ng-click="entity.specificationOptions=[];"**>  <**i class="fa fa-file-o"**></**i**> 新建 </**button**> |

### **7.3.3 删除行的实现**

实现思路：在每一行将索引值传递给事件处理方法，根据索引号从数组中删除元素。

修改每行的删除按钮，绑定点击事件：

|  |
| --- |
| <**button type="button" class="btn btn-default" title="删除"  ng-click="deleteTableRow($index);"**>  <**i class="fa fa-trash-o"**></**i**>删除 </**button**> |

注意：$index 用于获取ng-repeat指令循环中的索引号。

pinyougou-manager-web/src/main/webapp/js/controller/specificationController.js 新增以下代码：

|  |
| --- |
| */\*\* 删除规格选项行 \*/* $scope.deleteTableRow = **function**(index){  $scope.**entity**.specificationOptions.splice(index, 1); }; |

### **7.3.4 绑定规格名称**

|  |
| --- |
| <**td**><**input ng-model="entity.specName"  class="form-control" placeholder="规格名称"**></**td**> |

### **7.3.5 绑定保存事件**

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

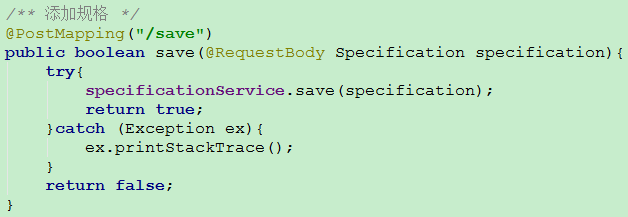
### **7.3.6 提交保存**

实现思路：我们将规格和规格选项数据合并成一个对象来传递，在业务逻辑中，得到对象中的规格和规格选项列表，插入规格返回规格ID，然后循环插入规格选项。

修改pinyougou-pojo/src/main/java/com.pinyougou.pojo中的Specification.java类(已修改)：

|  |
| --- |
| */\*\* 规格属性集合 @Transient:通用Mapper不会把该属性作为表中的列 \*/* @Transient **private** List<SpecificationOption> **specificationOptions**;  **public** List<SpecificationOption> getSpecificationOptions() {  **return specificationOptions**; } **public void** setSpecificationOptions(List<SpecificationOption> specificationOptions) {  **this**.**specificationOptions** = specificationOptions; } |

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller的SpecificationController.java:



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

|  |
| --- |
| @Autowired **private** SpecificationOptionMapper **specificationOptionMapper**; */\*\* 添加规格 \*/* @Override **public void** save(Specification specification) {  **try**{  **specificationMapper**.insertSelective(specification);  **specificationOptionMapper**.save(specification);  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |

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

|  |
| --- |
| */\*\* 添加规格选项 \*/* **void** save(Specification specification); |

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

|  |
| --- |
| *<!-- 添加规格选项 -->* <**insert id="save"**>  INSERT INTO tb\_specification\_option  (option\_name, spec\_id, orders) VALUES  <**foreach collection="specificationOptions"  item="so" separator=","**>  (#{so.optionName},#{id},#{so.orders})  </**foreach**> </**insert**> |

## 7.4修改规格（学员挑战）

### **7.4.1 前端页面**

specification.html页面绑定修改点击事件：

|  |
| --- |
| <**td class="text-center"**>  <**button type="button" class="btn bg-olive btn-xs"  data-toggle="modal" ng-click="show(entity);"  data-target="#editModal"**>修改</**button**> </**td**> |

修改pinyougou-manager-web/src/main/webapp/js/controller的

specificationController.js：

|  |
| --- |
| */\*\* 显示修改 \*/* $scope.show = **function**(entity){  */\*\* 把json对象转化成一个新的json对象 \*/* $scope.**entity** = ***JSON***.parse(***JSON***.stringify(entity));  */\*\* 调用服务层 \*/* baseService.sendGet(**"/specification/findSpecOption?id="** + entity.**id**)  .then(**function**(response){  $scope.**entity**.**specificationOptions** = response.**data**;  }); }; |

### **7.4.2 获取规格选项数据**

实现思路：通过规格ID，到后端查询规格选项列表，然后返回结果

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller的SpecificationController.java：

|  |
| --- |
| */\*\* 根据规格主键查询规格选项 \*/* @GetMapping(**"/findSpecOption"**) **public** List<SpecificationOption> findSpecOption(Long id){  **return specificationService**.findSpecOption(id); } |

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

|  |
| --- |
| */\*\* 根据规格主键查询规格选项 \*/* List<SpecificationOption> findSpecOption(Long id); |

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

|  |
| --- |
| */\*\* 根据规格主键查询规格选项 \*/* **public** List<SpecificationOption> findSpecOption(Long id){  **try** {  *// 创建规格选项对象封装查询条件* SpecificationOption so = **new** SpecificationOption();  so.setSpecId(id);  **return specificationOptionMapper**.select(so);  } **catch** (Exception ex) {  **throw new** RuntimeException(ex);  } } |

### **7.4.3 保存修改结果**

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller的SpecificationController.java：

|  |
| --- |
| */\*\* 修改规格 \*/* @PostMapping(**"/update"**) **public boolean** update(@RequestBody Specification specification){  **try**{  **specificationService**.update(specification);  **return true**;  }**catch** (Exception ex){  ex.printStackTrace();  }  **return false**; } |

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

|  |
| --- |
| */\*\* 修改规格与规格选项 \*/* **public void** update(Specification specification){  **try**{  *// 修改往规格表数据* **specificationMapper**.updateByPrimaryKeySelective(specification);  */\*\*########### 修改规格选项表数据 ###########\*/  // 第一步：删除规格选项表中的数据 spec\_id  // delete from tb\_specification\_option where spec\_id = ?  // 创建规格选项对象，封装删除条件 通用Mapper* SpecificationOption so = **new** SpecificationOption();  so.setSpecId(specification.getId());  **specificationOptionMapper**.delete(so);  *// 第二步：往规格选项表插入数据* **specificationOptionMapper**.save(specification);  }**catch** (Exception ex){  **throw new** RuntimeException(ex);  } } |

## 7.5删除规格（学员挑战）

实现思路：我们要删除规格的同时，还要记得将关联的规格选项删除掉。

### **7.5.1前端代码**

修改pinyougou-manager-web/src/main/webapp/admin/specification.html

列表的复选框绑定点击事件：

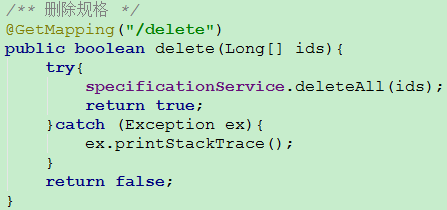
|  |
| --- |
| <**td**><**input type="checkbox"**  **ng-click="updateSelection($event,entity.id);"**></**td**> |

specification.html删除按钮绑定点击事件：

|  |
| --- |
| <**button type="button" class="btn btn-default" title="删除"  ng-click="delete();"**>  <**i class="fa fa-trash-o"**></**i**>删除</**button**> |

### **7.5.2后端代码**

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller的SpecificationController.java：



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

|  |
| --- |
| *// 批量删除规格与规格选项* @Override **public void** deleteAll(Serializable[] ids) {  **try**{  **for** (Serializable id : ids){  SpecificationOption so = **new** SpecificationOption();  so.setSpecId((Long)id);  **specificationOptionMapper**.delete(so);  **specificationMapper**.deleteByPrimaryKey(id);  }  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |