**品优购电商系统开发**

**第3章**

**规格管理&模板管理**

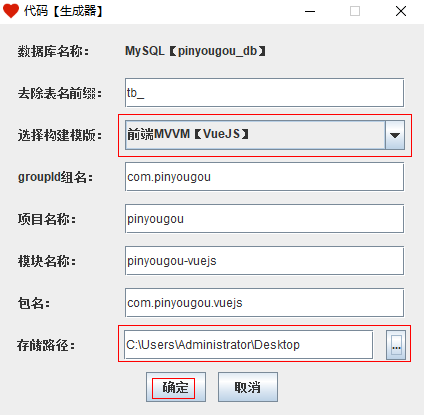
传智播客.黑马程序员

# 课程目标

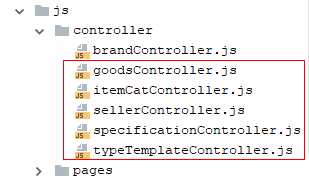
目标1：完成规格管理功能

目标2：完成模版管理功能

# 1.代码准备



拷贝pinyougou\pinyougou-vuejs\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控制器层js。**

# 2.规格管理

## 2.1需求及表结构分析

### **2.1.1 需求**

实现规格管理CRUD功能

### **2.1.2 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 |

## 2.2规格列表

### **2.2.1 引入JS**

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

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

### **2.2.2 分页组件**

分页组件显示在</table>后面：

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

### **2.2.3 迭代数据**

循环表格行：

|  |
| --- |
| <**tr v-for="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 v-model="searchEntity.specName"**  **placeholder="规格名称"**/>  <**button class="btn btn-default" @click="search(1)"**>查询</**button**> </**div**> |

### **2.2.4 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, @RequestParam(defaultValue=**"10"**)Integer rows){  */\*\* GET请求中文转码 \*/* **if** (StringUtils.*isNoneBlank*(specification.getSpecName())){  **try** {  specification.setSpecName(**new** String(specification  .getSpecName().getBytes(**"ISO8859-1"**),**"UTF-8"**));  } **catch** (Exception 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 @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.getPages(),  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**> |

## 2.3新增规格

### **2.3.1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。



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。

### **2.3.2 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"  @click="addTableRow()"**>  <**i class="fa fa-file-o"**></**i**>新增规格选项</**button**> </**div**> |

修改specification.html“新建”按钮，对entity中的规格选项数据进行初始化，否则向数组添加数据时会报错！(36行)

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

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

|  |
| --- |
| addTableRow : **function** () { *// 增加行* **this**.**entity**.specificationOptions.push({}); } |

specification.html循环列表行，绑定表格内的文本框：

|  |
| --- |
| <**tr v-for="(so,index) in entity.specificationOptions"**>  <**td**>  <**input v-model="so.optionName" class="form-control" placeholder="规格选项"**>  </**td**>  <**td**>  <**input v-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**> |

### **2.3.3 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"  @click="deleteTableRow(index);"**>  <**i class="fa fa-trash-o"**></**i**>删除 </**button**> |

注意：index 用于获取v-for循环中的索引号。

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

|  |
| --- |
| deleteTableRow : **function** (idx) { *// 删除行* **this**.**entity**.specificationOptions.splice(idx,1); } |

### **2.3.4 绑定规格名称**

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

### **2.3.5 绑定保存事件**

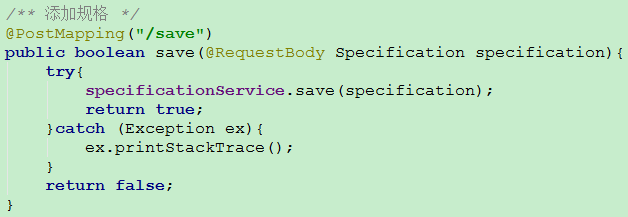
|  |
| --- |
| <**button class="btn btn-success" data-dismiss="modal"  aria-hidden="true" @click="saveOrUpdate();"**>保存</**button**> |

### **2.3.6 后台代码**

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**> |

## 2.4修改规格

### **2.4.1 前端代码**

specification.html页面为修改按钮绑定点击事件：

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

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

specificationController.js的show方法：

|  |
| --- |
| show : **function**(entity){ *// 数据回显  // 发送异步请求查询规格选项* **axios**.get(**"/specification/findSpecOption?id="** + entity.**id**).then(**function**(response){  *// 设置规格选项数组* entity.**specificationOptions** = response.**data**;  *// 把entity对象转化成json字符串* **var** jsonStr = ***JSON***.stringify(entity);  *// 把json字符串转化成一个新的json对象* vue.**entity** = ***JSON***.parse(jsonStr);  }); }, |

### **2.4.2 查询规格选项数据【后端】**

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id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id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();  *// 设置等于号查询条件 spec\_id = ?* so.setSpecId(id);  **return specificationOptionMapper**.select(so);  } **catch** (Exception ex) {  **throw new** RuntimeException(ex);  } } |

### **2.4.3 修改规格与规格选项【后端】**

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 \*/* @Override **public void** update(Specification specification) {  **try**{  */\*\*########### 1.修改规格表数据 ###########\*/* **specificationMapper**.updateByPrimaryKeySelective(specification);   */\*\*########### 2.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);  } } |

## 2.5删除规格

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。

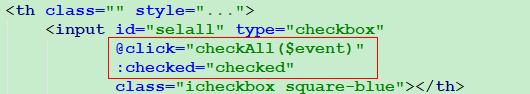
### **2.5.1前端代码**

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

行中的复选框绑定点击事件：

|  |
| --- |
| <**td**><**input type="checkbox"  :value="entity.id"  v-model="ids"**/></**td**> |

全选复选框绑定点击事件与选中属性：

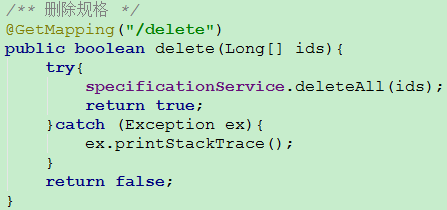


删除按钮绑定点击事件：

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

### **2.5.2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){  *// 1. 删除规格选项* SpecificationOption so = **new** SpecificationOption();  so.setSpecId((Long)id);  **specificationOptionMapper**.delete(so);  *// 2. 删除规格* **specificationMapper**.deleteByPrimaryKey(id);  }  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |

# 3.模板管理

## 3.1 需求及表结构分析

### **3.1.1 需求分析**

实现模板管理CRUD功能。

模板的作用：当添加商品时，需要选择商品分类，商品分类关联了模版，这样就知道该商品所属品牌及规格、扩展属性。

模板主要用于(模板表实际是 品牌与规格的中间表，用于组装品牌与规格)：

1. 用于关联品牌

2、用于关联规格

3、定义扩展属性



### **3.1.2 表结构分析**

**tb\_type\_template 模板表**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 含义 |
| id | bigint |  | 主键 |
| name | varchar | 80 | 模板名称 |
| spec\_ids | varchar | 1000 | 关联规格（json格式） |
| brand\_ids | varchar | 1000 | 关联品牌（json格式） |
| custom\_attribute\_items | varchar | 2000 | 扩展属性（json格式） |

## 3.2 模板列表

### **3.2.1引入JS**

在pinyougou-manager-web/src/main/webapp/admin/type\_template.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/typeTemplateController.js"**></**script**> |

### **3.2.2分页组件**

type\_template.html页面<table>结束标签后

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

### **3.2.3迭代数据**

用v-for迭代数据：

|  |
| --- |
| <**tr v-for="entity in dataList"**>  <**td**><**input type="checkbox"**></**td**>  <**td**>{{**entity**.**id**}}</**td**>  <**td**>{{**entity**.**name**}}</**td**>  <**td**>{{**entity**.brandIds}}</**td**>  <**td**>{{**entity**.specIds}}</**td**>  <**td**>{{**entity**.customAttributeItems}}</**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**> |

type\_template.html页面搜索条件：

|  |
| --- |
| <**div class="has-feedback"**>  <**input v-model="searchEntity.name"  placeholder="分类模板名称"**/>  <**button class="btn btn-default"  @click="search(1)"**>查询</**button**> </**div**> |

### **3.2.4 后台代码**

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

|  |
| --- |
| @RestController @RequestMapping(**"/typeTemplate"**) **public class** TypeTemplateController {  @Reference(timeout = 10000)  **private** TypeTemplateService **typeTemplateService**;  */\*\* 分页查询类型模版 \*/* @GetMapping(**"/findByPage"**)  **public** PageResult findByPage(TypeTemplate typeTemplate,  Integer page, @RequestParam(defaultValue = **"10"**) Integer rows){  */\*\* GET请求中文转码 \*/* **if** (typeTemplate != **null** &&  StringUtils.*isNoneBlank*(typeTemplate.getName())){  **try** {  typeTemplate.setName(**new** String(typeTemplate.getName()  .getBytes(**"ISO8859-1"**),**"UTF-8"**));  } **catch** (Exception e) {  e.printStackTrace();  }  }  **return typeTemplateService**.findByPage(typeTemplate, page, rows);  } } |

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

|  |
| --- |
| */\*\* 分页查询类型模版 \*/* PageResult findByPage(TypeTemplate typeTemplate,**int** page,**int** rows); |

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

|  |
| --- |
| @Service @Transactional **public class** TypeTemplateServiceImpl **implements** TypeTemplateService {  */\*\* 注入数据访问接口代理对象 \*/* @Autowired  **private** TypeTemplateMapper **typeTemplateMapper**;  @Override  **public void** save(TypeTemplate typeTemplate) {  }  @Override  **public void** update(TypeTemplate typeTemplate) {  }  @Override  **public void** delete(Serializable id) {  }  @Override  **public void** deleteAll(Serializable[] ids) {  }  @Override  **public** TypeTemplate findOne(Serializable id) {  **return null**;  }  @Override  **public** List<TypeTemplate> findAll() {  **return null**;  }  */\*\* 分页查询类型模版 \*/* @Override  **public** PageResult findByPage(TypeTemplate typeTemplate, **int** page, **int** rows) {  **try**{  */\*\* 开始分页 \*/* PageInfo<TypeTemplate> pageInfo = PageHelper.*startPage*(page, rows)  .doSelectPageInfo( **new** ISelect() {  @Override  **public void** doSelect() {  **typeTemplateMapper**.findAll(typeTemplate);  }  });  **return new** PageResult(pageInfo.getPages(), pageInfo.getList());  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  }  } } |

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

|  |
| --- |
| */\*\* 多条件查询类型模版 \*/* List<TypeTemplate> findAll(TypeTemplate typeTemplate); |

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

|  |
| --- |
| *<!-- 多条件查询类型模版 -->* <**select id="findAll" resultType="TypeTemplate"**>  select \* from tb\_type\_template  <**where**>  <**if test="name != null and name != ''"**>  name like CONCAT('%',#{name},'%')  </**if**>  </**where**>  order by id asc </**select**> |

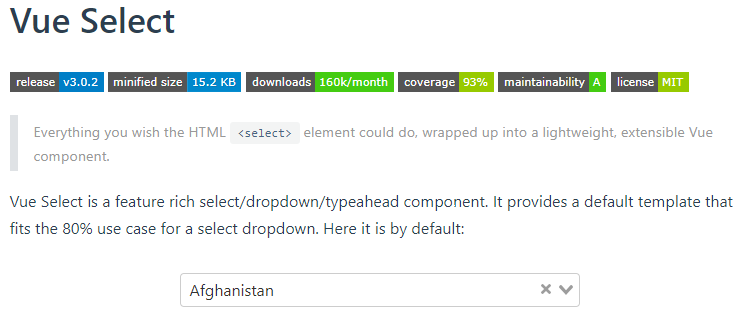
## 3.3 品牌下拉列表

在弹出窗口中有个品牌下拉列表，要求品牌是可以选择多个，这与我们之前的单选的下拉列表是不同的。我们要想实现这个功能，需要使用**vue-select**组件来完成。



### **3.3.1 Vue Select**

select2官网：<https://vue-select.org/>



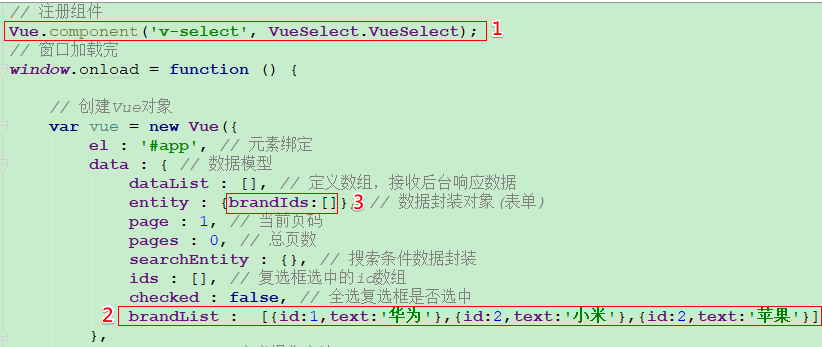
### **3.3.2品牌下拉列表（静态数据）**

pinyougou-manager-web/src/main/webapp/admin/type\_template.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/vue-select.js"></script>** <**script src="/js/controller/typeTemplateController.js"**></**script**> |

pinyougou-manager-web/src/main/webapp/js/controller/typeTemplateController.js中注册组件、select2组件数据、数据绑定变量：

|  |
| --- |
| *// 注册组件* **Vue**.component(**'v-select'**, **VueSelect**.**VueSelect**); |



pinyougou-manager-web/src/main/webapp/admin/type\_template.html 用select2组件实现多选下拉框

|  |
| --- |
| <**v-select placeholder="请选择品牌(可多选)"  multiple  :options="brandList"  label="text"  :value="entity.brandIds"**/> |

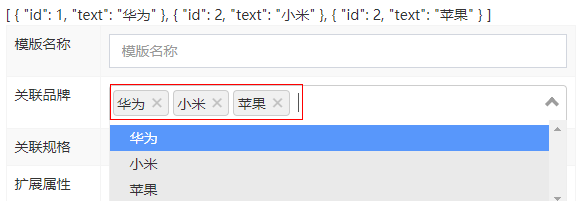
multiple：表示多选

**:options：配置数据源**

**label: 选中显示名称**

**:value：数据绑定**

最终实现效果如下：



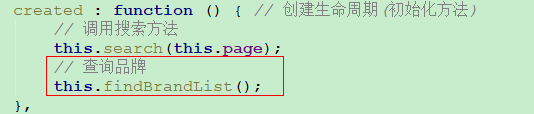
### **3.3.3前端代码**

pinyougou-manager-web/src/main/webapp/js/controller/typeTemplateController.js

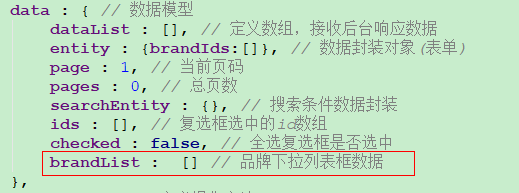
第一步：Vue的methods中定义发送异步请求

|  |
| --- |
| findBrandList : **function** () { *// 查询品牌数据* **axios**.get(**"/brand/findBrandList"**).then(**function**(response){  *// 获取响应数据 [{id:'',text:''},{id:'',text:''}]* vue.**brandList** = response.**data**;  }); } |

第二步：Vue的created方法中调用



第三步：修改Vue的data数据模型



### **3.3.4后端代码**

pinyougou-manager-web/src/main/java/com.pinyougou.manager.

controller的BrandController.java，增加方法：

|  |
| --- |
| */\*\* 查询所有的品牌 \*/* @GetMapping(**"/findBrandList"**) **public** List<Map<String, Object>> findBrandList(){  **return brandService**.findAllByIdAndName(); } |

pinyougou-interface/src/main/java/com.pinyougou.service/BrandService.java，增加方法

|  |
| --- |
| */\*\* 查询所有的品牌(id与name) \*/* List<Map<String, Object>> findAllByIdAndName(); |

修改pinyougou-sellergoods-service/src/main/java/com.pinyougou.

sellergoods.service.impl的BrandServiceImpl.java，增加方法

|  |
| --- |
| */\*\* 查询所有的品牌(id与name) \*/* **public** List<Map<String, Object>> findAllByIdAndName(){  **try**{  **return brandMapper**.findAllByIdAndName();  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |

在pinyougou-mapper/src/main/java/com.pinyougou.mapper

的BrandMapper.java中添加方法：

|  |
| --- |
| */\*\* 查询所有的品牌(id与name) \*/* @Select(**"select id, name as text from tb\_brand order by id asc"**) List<Map<String,Object>> findAllByIdAndName(); |

## 3.4 规格下拉列表

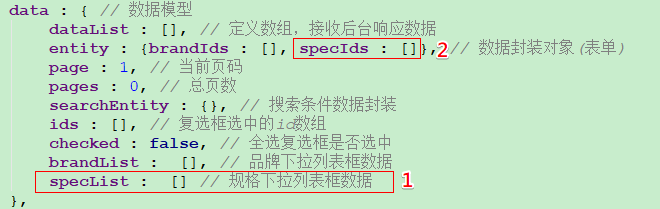


### **3.4.1前端代码**

第一步：规格下拉列表框

|  |
| --- |
| <**v-select placeholder="请选择规格(可多选)" multiple  :options="specList"  label="text"  :value="entity.specIds"**/> |

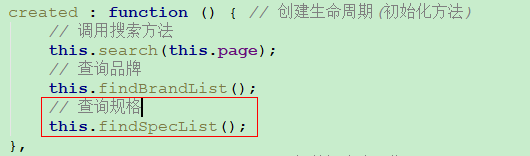
第二步：修改数据数型



第三步：发送异步请求查询规格数据

|  |
| --- |
| findSpecList : **function** () { *// 查询规格数据* **axios**.get(**"/specification/findSpecList"**).then(**function**(response){  *// 获取响应数据 [{id:'',text:''},{id:'',text:''}]* vue.**specList** = response.**data**;  }); } |

第四步：初始化调用



### **3.4.2后端代码**

|  |
| --- |
| **SpecificationController.java：**  */\*\* 查询所有的规格(id与specName) \*/* @GetMapping(**"/findSpecList"**) **public** List<Map<String, Object>> findSpecList(){  **return specificationService**.findAllByIdAndName(); }  **SpecificationService.java：**  */\*\* 查询所有的规格(id与specName) \*/* List<Map<String, Object>> findAllByIdAndName();  **SpecificationServiceImpl.java：**  */\*\* 查询所有的规格(id与specName) \*/* **public** List<Map<String, Object>> findAllByIdAndName(){  **try**{  **return specificationMapper**.findAllByIdAndName();  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } }  **SpecificationMapper.java：**  */\*\* 查询所有的规格(id与specName) \*/* @Select(**"select id, spec\_name as text from tb\_specification order by id asc"**) List<Map<String, Object>> findAllByIdAndName(); |

## 3.5 模板扩展属性

### **3.5.1 增加行**

修改type\_template.html页面“新增扩展属性按钮”，绑定点击事件

|  |
| --- |
| <**button type="button" class="btn btn-default" title="新增扩展属性"  @click="addTableRow()"**>  <**i class="fa fa-file-o"**></**i**> 新增扩展属性 </**button**> |

pinyougou-manager-web/src/main/webapp/js/controller/typeTemplateController.js的methods新增方法

|  |
| --- |
| addTableRow : **function** () { *// 增加行* **this**.**entity**.**customAttributeItems**.push({}); } |

pinyougou-manager-web/src/main/webapp/admin/type\_template.html中的“新建”按钮，执行初始化操作(38行)

|  |
| --- |
| <**button type="button" class="btn btn-default" title="新建"  data-toggle="modal" data-target="#editModal"  @click="entity={brandIds : [], specIds : [], customAttributeItems:[]};"**>  <**i class="fa fa-file-o"**></**i**> 新建 </**button**> |

用v-for循环行

|  |
| --- |
| <**tr v-for="(c,index) in entity.customAttributeItems"**>  <**td**><**input class="form-control" v-model="c.text" placeholder="属性名称"**></**td**>  <**td**><**button type="button" class="btn btn-default" title="删除"**>  <**i class="fa fa-trash-o"**></**i**> 删除</**button**>  </**td**> </**tr**> |

### **3.5.2 删除行**

修改每行的删除按钮，绑定点击事件

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

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

typeTemplateController.js的methods新增方法

|  |
| --- |
| deleteTableRow : **function** (idx) { *// 删除行* **this**.**entity**.**customAttributeItems**.splice(idx, 1); } |

## 3.6 新增模板

### **3.6.1前端代码**

pinyougou-manager-web/src/main/webapp/admin/type\_template.html，绑定模板名称

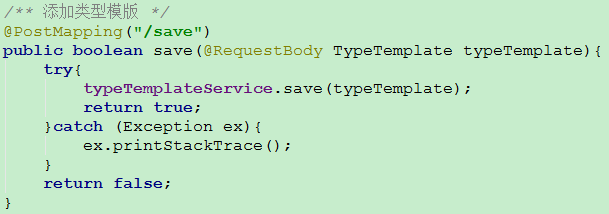
|  |
| --- |
| <**td**><**input class="form-control" v-model="entity.name" placeholder="模版名称"**></**td**> |

保存按钮，绑定点击事件

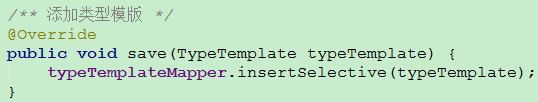
|  |
| --- |
| <**button class="btn btn-success" data-dismiss="modal"  @click="saveOrUpdate()" aria-hidden="true"**>保存</**button**> |

### **3.6.2后端代码**

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller的TypeTemplateController.java，增加方法：



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



## 3.7 修改模板

### **3.7.1前端代码**

修改pinyougou-manager-web/src/main/webapp/admin/type\_template.html

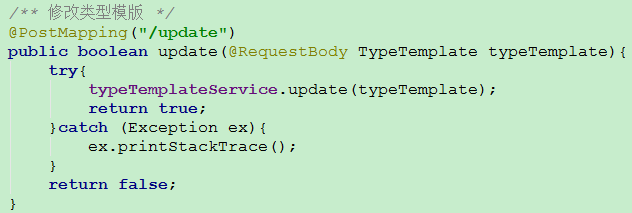
|  |
| --- |
| <**button type="button" class="btn bg-olive btn-xs"  data-toggle="modal" data-target="#editModal"  @click="show(entity);"**>修改</**button**> |

pinyougou-manager-web/src/main/webapp/js/controller/typeTemplateController.js的methods中show方法：

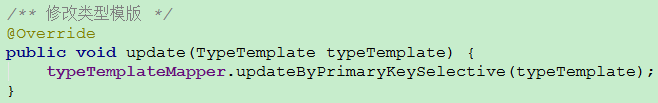
|  |
| --- |
| show : **function**(entity){ *// 数据回显  // 把entity对象转化成json字符串* **var** jsonStr = ***JSON***.stringify(entity);  *// 把json字符串转化成一个新的json对象* **this**.**entity** = ***JSON***.parse(jsonStr);  *// 把品牌json字符串转化成json数组* **this**.**entity**.**brandIds** = ***JSON***.parse(**this**.**entity**.**brandIds**);  *// 把规格json字符串转化成json数组* **this**.**entity**.**specIds** = ***JSON***.parse(**this**.**entity**.**specIds**);  *// 把扩展属性json字符串转化成json数组* **this**.**entity**.**customAttributeItems** = ***JSON*** .parse(**this**.**entity**.**customAttributeItems**); }, |

### **3.7.2后台代码**

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller/TypeTemplateController.java，增加方法：



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



## 3.8 删除模板

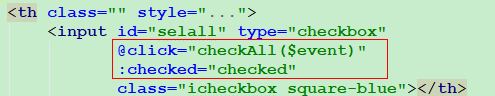
### **3.8.1前端代码**

pinyougou-manager-web/src/main/webapp/admin/type\_template.html

表格中的复选框，绑定数据:

|  |
| --- |
| <**td**><**input type="checkbox"  :value="entity.id" v-model="ids"**></**td**> |

全选复选框，绑定点击事件与选中属性:

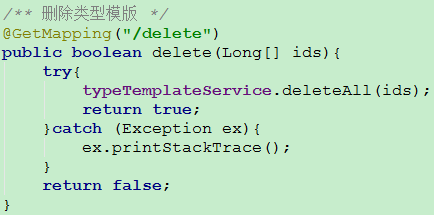


删除按钮，绑定点击事件:

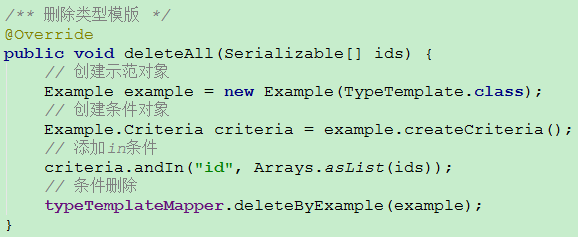
|  |
| --- |
| <**button type="button" class="btn btn-default" title="删除"  @click="del()"**>  <**i class="fa fa-trash-o"**></**i**> 删除 </**button**> |

### **3.8.2后台代码**

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller/TypeTemplateController.java，增加方法：

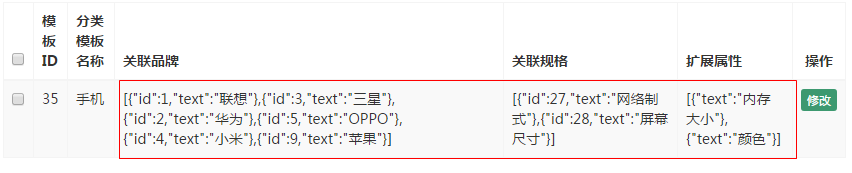


pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl/TypeTemplateServiceImpl.java



## 3.9 优化模板列表显示（了解）

我们现在完成的列表中都是以JSON格式显示的，不利于用户的查询。



我们需要将信息以更友好的方式展现出来,如下图形式



我们需要将一个json字符串中某个属性的值提取出来，用逗号拼接成一个新的字符串。pinyougou-manager-web/src/main/webapp/js/controller/brandController.js定义方法:

|  |
| --- |
| *// 提取数组中json某个属性，返回拼接的字符串(逗号分隔)* jsonArr2Str : **function** (jsonArrStr) {  *// 把jsonArrStr转化成JSON数组对象* **var** jsonArr = ***JSON***.parse(jsonArrStr);  *// 定义新数组* **var** resArr = [];  *// 迭代json数组* **for** (**var** i = 0; i < jsonArr.**length**; i++){  *// 取数组中的一个元素* **var** json = jsonArr[i];  *// 把json对象的值添加到新数组* resArr.push(json.**text**);  }  *// 返回数组中的元素用逗号分隔的字符串* **return** resArr.join(**","**); } |

type\_template.html页面上使用该方法进行转换

|  |
| --- |
| <**tr v-for="entity in dataList"**>  <**td**><**input type="checkbox"  :value="entity.id"  v-model="ids"**></**td**>  <**td**>{{**entity**.**id**}}</**td**>  <**td**>{{**entity**.**name**}}</**td**>  <**td**>{{jsonArr2Str(**entity**.**brandIds**)}}</**td**>  <**td**>{{jsonArr2Str(**entity**.**specIds**)}}</**td**>  <**td**>{{jsonArr2Str(**entity**.**customAttributeItems**)}}</**td**>  <**td class="text-center"**>  <**button type="button" class="btn bg-olive btn-xs"  data-toggle="modal" data-target="#editModal"  @click="show(entity);"**>修改  </**button**>  </**td**> </**tr**> |