乐字节教育高级架构课程

正所谓“授人以鱼不如授人以渔”，你们想要的 Java 学习资料来啦！不管你是学生，还是已经步入职场的同行，希望你们都要珍惜眼前的学习机会，奋斗没有终点，知识永不过时。

扫描下方二维码即可领取

乐字节晓啡

乐字节官方交流群

商品分类-新增分类-级联查询

Service服务层

GoodsCategoryService.java

```java
package com.xxxx.manager.service;

import com.xxxx.manager.pojo.GoodsCategory;

import java.util.List;

/**
 * 商品分类服务接口
 *
 * @author zhoubin
 * @since 1.0.0
 */
public interface GoodsCategoryService {
   /**
    * 商品分类-新增分类-查询所有顶级分类
    * @return
    */
   List<GoodsCategory> selectCategoryTopList();

   /**
    * 商品分类-新增分类-根据父id查询子分类
    */
   List<GoodsCategory> selectCategoryListByParentId(short parentId);

   /**
    * 商品分类-新增分类-查询商品分类（通用）
    */
   List<GoodsCategory> selectCategoryList(short parentId);
}
```

GoodsCategoryServiceImpl.java

```java
package com.xxxx.manager.service.impl;

import com.xxxx.manager.mapper.GoodsCategoryMapper;
import com.xxxx.manager.pojo.GoodsCategory;
import com.xxxx.manager.pojo.GoodsCategoryExample;
import com.xxxx.manager.service.GoodsCategoryService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
```

```
/**
 * 商品分类服务类
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Service
public class GoodsCategoryServiceImpl implements GoodsCategoryService {

   @Autowired
   private GoodsCategoryMapper goodsCategoryMapper;

   /**
    * 商品分类-新增分类-查询所有顶级分类
    * @return
    */
   @Override
   public List<GoodsCategory> selectCategoryTopList() {
      // 创建查询对象
      GoodsCategoryExample example = new GoodsCategoryExample();
      // where条件
      example.createCriteria().andParentIdEqualTo((short) 0);
      return goodsCategoryMapper.selectByExample(example);
   }

   /**
    * 商品分类-新增分类-根据父id查询子分类
    */
   @Override
   public List<GoodsCategory> selectCategoryListByParentId(short parentId) {
      // 创建查询对象
      GoodsCategoryExample example = new GoodsCategoryExample();
      // where条件
      example.createCriteria().andParentIdEqualTo(parentId);
      return goodsCategoryMapper.selectByExample(example);
   }

   /**
    * 商品分类-新增分类-查询商品分类（通用）
    */
   @Override
   public List<GoodsCategory> selectCategoryList(short parentId) {
      // 创建查询对象
      GoodsCategoryExample example = new GoodsCategoryExample();
      // where条件
      example.createCriteria().andParentIdEqualTo(parentId);
      return goodsCategoryMapper.selectByExample(example);
   }
}
```

Controller控制层

GoodsController.java

```java
package com.ego.manager.controller;

import com.xxxx.manager.pojo.GoodsCategory;
import com.xxxx.manager.service.GoodsCategoryService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

import java.util.List;

/**
 * 商品
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Controller
@RequestMapping("goods")
public class GoodsController {

   @Autowired
   private GoodsCategoryService goodsCategoryService;

   /**
    * 商品分类-添加-页面跳转
    * @return
    */
   @RequestMapping("category/add")
   public String categoryAdd(Model model){
      List<GoodsCategory> gcList = goodsCategoryService.selectCategoryTopList();
      model.addAttribute("gcList",gcList);
      return "goods/category/category-add";
   }

   /**
    * 商品分类-列表-页面跳转
    * @return
    */
   @RequestMapping("category/list")
   public String categoryList(){
      return "goods/category/category-list";
   }

   /**
    * 商品分类-添加-分类联查
    * @param parentId
    * @return
    */
   @RequestMapping("category/{parentId}")
   @ResponseBody
```

```
    public List<GoodsCategory> selectCategoryList(@PathVariable Short parentId){
        return goodsCategoryService.selectCategoryList(parentId);
    }
}
```

ftl页面

category-add.ftl

```
<div class="col-sm-3">
    <select name="parent_id_1" id="parent_id_1"
            onchange="getCategory(this.value,'parent_id_2','0');"
            class="small form-control">
        <option value="0">顶级分类</option>
        <#list gcList as gc>
            <option value="${gc.id}">${gc.name}</option>
        </#list>
    </select>
</div>
```

```
/**
 * 获取多级联动的商品分类
 * id:当前选择框的值
 * next: 下级选择框显示的内容
 * select_id:
 */
function getCategory(id, next, select_id) {
    var url = '${ctx}/goods/category/' + id;
// 用户重新选择顶级分类时，重置下级分类为：请选择商品分类，且清空下级分类信息
    var htmlStr = "<option value='0'>请选择商品分类</option>";
    if(0==id){
        $("#"+next).html(htmlStr);
        return;
    }
    $.ajax({
        type: "GET",
        url: url,
        error: function (request) {
            layer.alert("获取子分类失败！");
        },
        success: function (result) {
            if (result.length > 0) {
                for (i = 0; i < result.length; i++) {
                    htmlStr += "<option value='" + result[i].id + "'>" + result[i].name + "
</option>"
                }
                $("#"+next).html(htmlStr);
            }else {
                layer.alert("获取子分类失败！");
            }
        }
    });
}
```