

Java 架构师-技术专家练习笔记

查询轮播图方法：

Pojo 实体类：

```
package com.imooc.pojo;

import javax.persistence.Column;
import javax.persistence.Id;
import javax.persistence.Table;
import java.util.Date;

/**
 * 表名：carouse
 */
@Table(name = "carouse")
public class Carouse {
    /**
     * 主键
     */
    @Id
    private String id;

    /**
     * 图片 图片地址
     */
    @Column(name = "image_url")
    private String imageUrl;

    /**
     * 背景色 背景颜色
     */
    private String background;

    /**
     * 商品 id 商品 id
     */
    @Column(name = "item_id")
    private String itemId;
```

```
/**
 * 商品分类 Id 商品分类 id
 */
@Column(name = "cat_id")
private String catId;

/**
 * 轮播图类型 轮播图类型，用于判断，可以根据商品 id 或者分类进行页面跳转 1:
商品 2: 分类
 */
private Integer type;

/**
 * 轮播图展示顺序 轮播图展示顺序，从小到大
 */
private Integer sort;

/**
 * 是否展示 是否展示，1: 展示，0: 不展示
 */
@Column(name = "is_show")
private Integer isShow;

/**
 * 创建时间 创建时间
 */
@Column(name = "created_time")
private Date createTime;

/**
 * 更新时间 更新时间
 */
@Column(name = "updated_time")
private Date updateTime;

/**
 * 获取主键
 *
 * @return id - 主键
 */
public String getId() {
    return id;
}
```

```
/**
 * 设置主键
 *
 * @param id 主键
 */
public void setId(String id) {
    this.id = id;
}

/**
 * 获取图片 图片地址
 *
 * @return imageUrl - 图片 图片地址
 */
public String getImageUrl() {
    return imageUrl;
}

/**
 * 设置图片 图片地址
 *
 * @param imageUrl 图片 图片地址
 */
public void setImageUrl(String imageUrl) {
    this.imageUrl = imageUrl;
}

/**
 * 获取背景色 背景颜色
 *
 * @return background - 背景色 背景颜色
 */
public String getBackground() {
    return background;
}

/**
 * 设置背景色 背景颜色
 *
 * @param background 背景色 背景颜色
 */
public void setBackground(String background) {
    this.background = background;
}
```

```

}

/**
 * 获取商品 id 商品 id
 *
 * @return itemId - 商品 id 商品 id
 */
public String getItemId() {
    return itemId;
}

/**
 * 设置商品 id 商品 id
 *
 * @param itemId 商品 id 商品 id
 */
public void setItemId(String itemId) {
    this.itemId = itemId;
}

/**
 * 获取商品分类 Id 商品分类 id
 *
 * @return catId - 商品分类 Id 商品分类 id
 */
public String getCatId() {
    return catId;
}

/**
 * 设置商品分类 Id 商品分类 id
 *
 * @param catId 商品分类 Id 商品分类 id
 */
public void setCatId(String catId) {
    this.catId = catId;
}

/**
 * 获取轮播图类型 轮播图类型,用于判断,可以根据商品 id 或者分类进行页面跳转 1:
商品 2: 分类
 *
 * @return type - 轮播图类型 轮播图类型, 用于判断, 可以根据商品 id 或者分类进行
页面跳转 1: 商品 2: 分类

```

```

    */
    public Integer getType() {
        return type;
    }

    /**
     * 设置轮播图类型 轮播图类型,用于判断,可以根据商品 id 或者分类进行页面跳转 1:
    商品 2: 分类
     *
     * @param type 轮播图类型 轮播图类型,用于判断,可以根据商品 id 或者分类进行页
    面跳转 1: 商品 2: 分类
     */
    public void setType(Integer type) {
        this.type = type;
    }

    /**
     * 获取轮播图展示顺序 轮播图展示顺序, 从小到大
     *
     * @return sort - 轮播图展示顺序 轮播图展示顺序, 从小到大
     */
    public Integer getSort() {
        return sort;
    }

    /**
     * 设置轮播图展示顺序 轮播图展示顺序, 从小到大
     *
     * @param sort 轮播图展示顺序 轮播图展示顺序, 从小到大
     */
    public void setSort(Integer sort) {
        this.sort = sort;
    }

    /**
     * 获取是否展示 是否展示, 1: 展示, 0: 不展示
     *
     * @return isShow - 是否展示 是否展示, 1: 展示, 0: 不展示
     */
    public Integer getIsShow() {
        return isShow;
    }

    /**

```

```

    * 设置是否展示 是否展示, 1: 展示, 0: 不展示
    *
    * @param isShow 是否展示 是否展示, 1: 展示, 0: 不展示
    */
    public void setIsShow(Integer isShow) {
        this.isShow = isShow;
    }

    /**
     * 获取创建时间 创建时间
     *
     * @return createTime - 创建时间 创建时间
     */
    public Date getCreateTime() {
        return createTime;
    }

    /**
     * 设置创建时间 创建时间
     *
     * @param createTime 创建时间 创建时间
     */
    public void setCreateTime(Date createTime) {
        this.createTime = createTime;
    }

    /**
     * 获取更新时间 更新时间
     *
     * @return updateTime - 更新时间 更新时间
     */
    public Date getUpdateTime() {
        return updateTime;
    }

    /**
     * 设置更新时间 更新时间
     *
     * @param updateTime 更新时间 更新时间
     */
    public void setUpdateTime(Date updateTime) {
        this.updateTime = updateTime;
    }
}

```

Mapper 接口:

```
package com.imooc.mapper;

import com.imooc.my.mapper.MyMapper;
import com.imooc.pojo.Carouse;

public interface CarouseMapper extends MyMapper<Carouse> {
}
```

Mapper xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.imooc.mapper.CarouseMapper">
  <resultMap id="BaseResultMap" type="com.imooc.pojo.Carouse">
    <!--
      WARNING - @mbg.generated
    -->
    <id column="id" jdbcType="VARCHAR" property="id" />
    <result column="image_url" jdbcType="VARCHAR" property="imageUrl" />
    <result column="background" jdbcType="VARCHAR" property="background" />
    <result column="item_id" jdbcType="VARCHAR" property="itemId" />
    <result column="cat_id" jdbcType="VARCHAR" property="catId" />
    <result column="type" jdbcType="INTEGER" property="type" />
    <result column="sort" jdbcType="INTEGER" property="sort" />
    <result column="is_show" jdbcType="INTEGER" property="isShow" />
    <result column="created_time" jdbcType="TIMESTAMP" property="createdTime" />
    <result column="updated_time" jdbcType="TIMESTAMP" property="updatedTime" />
  </resultMap>
</mapper>
```

Service 接口:

```
package com.imooc.service;

import com.imooc.pojo.Carouse;
import java.util.List;
```

/**

```

* @Description
* @Author zhangxin
* @Date 20230116
**/
public interface CarouseService {
    /**
     *功能描述: 查询所有轮播图列表
     * @author zhangxin
     * @date 2023/1/16
     * @params
     * @return java.util.List<com.imooc.pojo.Carouse>
     */
    public List<Carouse> queryAll(Integer isShow);
}

```

Service 接口实现:

```

package com.imooc.service.impl;

import com.imooc.mapper.CarouseMapper;
import com.imooc.pojo.Carouse;
import com.imooc.service.CarouseService;
import org.springframework.beans.factory.annotation.Autowired;
import tk.mybatis.mapper.entity.Example;

import java.util.List;

/**
 * @Description
 * @Author zhangxin
 * @Date 20230116
 **/
public class CarouseServiceImpl implements CarouseService {

    @Autowired
    private CarouseMapper carouseMapper;

    @Override
    public List<Carouse> queryAll(Integer isShow) {
        Example exception =new Example(Carouse.class);
        //排序
        exception.orderBy("sort").desc();
    }
}

```

```

        Example.Criteria criteria =exception.createCriteria();
        //查询条件
        criteria.andEqualTo("isShow",isShow);
        List<Carouse> result = carouseMapper.selectByExample(exception);
        return result;
    }
}

```

枚举类 YesOrNo :

```

package com.imooc.enums;

/**
 * @Description 性别枚举
 * @Author zhangxin
 * @Date 20230113
 */
public enum YesOrNo {
    NO(0, "否"),
    YES(1, "是");

    public final Integer type;
    public final String value;

    YesOrNo(Integer type, String value) {
        this.type = type;
        this.value = value;
    }
}

```

Api 入口 IndexController:

```

package com.imooc.controller;

import com.imooc.enums.YesOrNo;
import com.imooc.pojo.Carouse;
import com.imooc.service.CarouseService;
import com.imooc.utils.IMOOCJSONResult;
import io.swagger.annotations.Api;
import org.springframework.beans.factory.annotation.Autowired;

```

```

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

/**
 * @Description
 * @Author zhangxin
 * @Date 20230116
 */
@Api(value = "首页",tags = "首页展示的相关接口")
@RequestMapping("index")
@RestController
public class IndexController {
    @Autowired
    private CarouseService carouseService;

    @GetMapping("/carousel")
    public IMOOCJSONResult carousel(){
        List<Carouse> list = carouseService.queryAll(YesOrNo.YES.type);
        return new IMOOCJSONResult().ok(list);
    }
}

```

三级分类查询：

一级分类查询：

Pojo 实体类：

```

package com.imooc.pojo;

import javax.persistence.Column;
import javax.persistence.Id;
import javax.persistence.Table;

/**
 * 表名： category
 * 表注释： 商品分类
 */

```

```
@Table(name = "category")
public class Category {
    /**
     * 主键 分类 id 主键
     */
    @Id
    private Integer id;

    /**
     * 分类名称 分类名称
     */
    private String name;

    /**
     * 分类类型 分类类型 1:一级大类 2: 二级大类: 3: 三级大类
     */
    private Integer type;

    /**
     * 父 ID 父 ID
     */
    @Column(name = "father_id")
    private Integer fatherId;

    /**
     * 图标 图标
     */
    private String logo;

    /**
     * 口号 口号
     */
    private String slogan;

    /**
     * 分类图 分类图
     */
    @Column(name = "cat_image")
    private String catImage;

    /**
     * 背景颜色 背景颜色
     */
    @Column(name = "bg_color")
```

```
private String bgColor;

/**
 * 获取主键 分类 id 主键
 *
 * @return id - 主键 分类 id 主键
 */
public Integer getId() {
    return id;
}

/**
 * 设置主键 分类 id 主键
 *
 * @param id 主键 分类 id 主键
 */
public void setId(Integer id) {
    this.id = id;
}

/**
 * 获取分类名称 分类名称
 *
 * @return name - 分类名称 分类名称
 */
public String getName() {
    return name;
}

/**
 * 设置分类名称 分类名称
 *
 * @param name 分类名称 分类名称
 */
public void setName(String name) {
    this.name = name;
}

/**
 * 获取分类类型 分类类型 1:一级大类 2: 二级大类: 3: 三级大类
 *
 * @return type - 分类类型 分类类型 1:一级大类 2: 二级大类: 3: 三级大类
 */
public Integer getType() {
```

```
        return type;
    }

    /**
     * 设置分类类型 分类类型 1:一级大类 2: 二级大类: 3: 三级大类
     *
     * @param type 分类类型 分类类型 1:一级大类 2: 二级大类: 3: 三级大类
     */
    public void setType(Integer type) {
        this.type = type;
    }

    /**
     * 获取父 ID 父 ID
     *
     * @return fatherId - 父 ID 父 ID
     */
    public Integer getFatherId() {
        return fatherId;
    }

    /**
     * 设置父 ID 父 ID
     *
     * @param fatherId 父 ID 父 ID
     */
    public void setFatherId(Integer fatherId) {
        this.fatherId = fatherId;
    }

    /**
     * 获取图标 图标
     *
     * @return logo - 图标 图标
     */
    public String getLogo() {
        return logo;
    }

    /**
     * 设置图标 图标
     *
     * @param logo 图标 图标
     */
    public void setLogo(String logo) {
        this.logo = logo;
    }
}
```

```
public void setLogo(String logo) {
    this.logo = logo;
}

/**
 * 获取口号 口号
 *
 * @return slogan - 口号 口号
 */
public String getSlogan() {
    return slogan;
}

/**
 * 设置口号 口号
 *
 * @param slogan 口号 口号
 */
public void setSlogan(String slogan) {
    this.slogan = slogan;
}

/**
 * 获取分类图 分类图
 *
 * @return catImage - 分类图 分类图
 */
public String getCatImage() {
    return catImage;
}

/**
 * 设置分类图 分类图
 *
 * @param catImage 分类图 分类图
 */
public void setCatImage(String catImage) {
    this.catImage = catImage;
}

/**
 * 获取背景颜色 背景颜色
 *
 * @return bgColor - 背景颜色 背景颜色
```

```

    */
    public String getBgColor() {
        return bgColor;
    }

    /**
     * 设置背景颜色 背景颜色
     *
     * @param bgColor 背景颜色 背景颜色
     */
    public void setBgColor(String bgColor) {
        this.bgColor = bgColor;
    }
}

```

Mapper 接口:

```

package com.imooc.mapper;

import com.imooc.my.mapper.MyMapper;
import com.imooc.pojo.Category;

public interface CategoryMapper extends MyMapper<Category> {
}

```

Mapper xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.imooc.mapper.CategoryMapper">
    <resultMap id="BaseResultMap" type="com.imooc.pojo.Category">
        <!--
            WARNING - @mbg.generated
        -->
        <id column="id" jdbcType="INTEGER" property="id" />
        <result column="name" jdbcType="VARCHAR" property="name" />
        <result column="type" jdbcType="INTEGER" property="type" />
        <result column="father_id" jdbcType="INTEGER" property="fatherId" />
        <result column="logo" jdbcType="VARCHAR" property="logo" />
    
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/138033027027006074>