

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



rpc服务系统编写购物车实体类

shop-rpc的CartVo.java

```
package com.xxxx.rpc.vo;

import java.io.Serializable;
import java.math.BigDecimal;
import java.util.Date;

public class CartVo implements Serializable {
    private static final long serialVersionUID = -196260271603271346L;
    // 商品ID
    private Integer goodsId;
    // 商品名称
    private String goodsName;
    // 商品价格
    private BigDecimal marketPrice;
    // 商品图片
    private String originalImg;
    // 商品数量
    private Integer goodsNum;
    // 商品添加时间
    private Date addTime;

    public Integer getGoodsId() {
        return goodsId;
    }

    public void setGoodsId(Integer goodsId) {
        this.goodsId = goodsId;
    }

    public String getGoodsName() {
        return goodsName;
    }

    public void setGoodsName(String goodsName) {
        this.goodsName = goodsName;
    }

    public BigDecimal getMarketPrice() {
        return marketPrice;
    }

    public void setMarketPrice(BigDecimal marketPrice) {
        this.marketPrice = marketPrice;
    }

    public String getOriginalImg() {
```

```
        return originalImg;
    }

    public void setOriginalImg(String originalImg) {
        this.originalImg = originalImg;
    }

    public Integer getGoodsNum() {
        return goodsNum;
    }

    public void setGoodsNum(Integer goodsNum) {
        this.goodsNum = goodsNum;
    }

    public Date getAddTime() {
        return addTime;
    }

    public void setAddTime(Date addTime) {
        this.addTime = addTime;
    }

    @Override
    public String toString() {
        return "CartVo{" +
            "goodsId=" + goodsId +
            ", goodsName='" + goodsName + '\'' +
            ", marketPrice=" + marketPrice +
            ", originalImg='" + originalImg + '\'' +
            ", goodsNum=" + goodsNum +
            ", addTime=" + addTime +
            '}';
    }
}
```

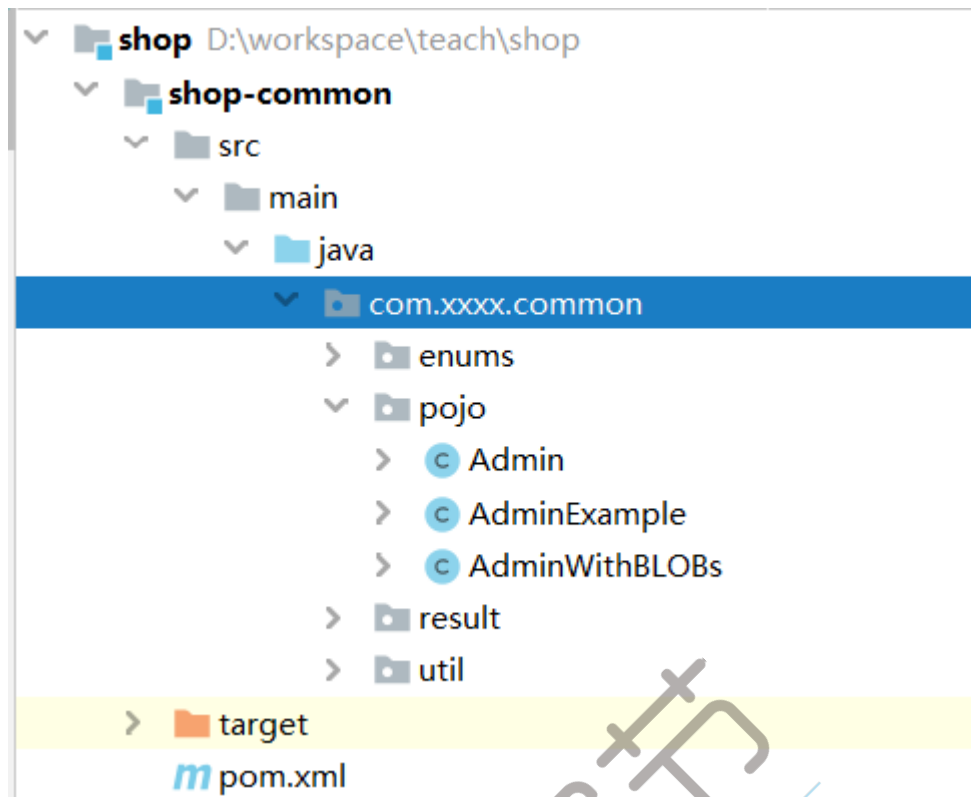
rpc服务系统编写购物车服务

shop-rpc的application-dev.yml

```
#用户购物车key
user.cart: userCart
```

提取pojo的Admin到com.shop.common.pojo包下





shop-rpc的CartService.java

```
package com.xxxx.rpc.service;

import com.xxxx.common.pojo.Admin;
import com.xxxx.common.result.BaseResult;
import com.xxxx.rpc.vo.CartVo;

/**
 * 购物车service
 */
public interface CartService {
    /**
     * 加入购物车
     * @param cart
     * @param admin
     * @return
     */
    BaseResult addCart(CartVo cart, Admin admin);

    /**
     * 获取购物车数量
     * @param admin
     * @return
     */
    Integer getCartNum(Admin admin);
}
```

shop-rpc的CartServiceImpl.java

```
package com.xxxx.rpc.service.impl;

import com.alibaba.dubbo.config.annotation.Service;
import com.xxxx.common.result.BaseResult;
import com.xxxx.common.util.JsonUtil;
import com.xxxx.rpc.service.CartService;
import com.xxxx.common.pojo.Admin;
import com.xxxx.rpc.vo.CartVo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.data.redis.core.HashOperations;
import org.springframework.data.redis.core.RedisTemplate;
import org.springframework.stereotype.Component;
import org.springframework.util.CollectionUtils;
import org.springframework.util.StringUtils;
import java.util.HashMap;
import java.util.Map;

/**
 * 购物车service实现类
 *
 * @author zhoubin
 * @since 1.0.0
 */
@Service(interfaceClass = CartService.class)
@Component
public class CartServiceImpl implements CartService {

    @Autowired
    private RedisTemplate<String, Object> redisTemplate;

    /**
     * 初始化redis
     * 参数一: redis的key
     * 参数二: hash的key
     * 参数三: hash的value
     */
    private HashOperations<String, String, String> hashOperations = null;

    @Value("${user.cart}")
    private String userCart;

    /**
     * 加入购物车
     *
     * @param cart
     * @param admin
     * @return
     */
    @Override
    public BaseResult addCart(CartVo cart, Admin admin) {
        //如果用户信息不存在, 直接返回
        if (null == admin || null == admin.getAdminId()) {
```

```

        return BaseResult.error();
    }
    //获取用户编号
    Short userId = admin.getAdminId();
    //查询当前用户的购物车信息
    hashOperations = redisTemplate.opsForHash();
    Map<String, String> cartMap = hashOperations.entries(userCart + ":" + userId);
    //如果购物车信息不为空, 修改购物车信息
    if (!CollectionUtils.isEmpty(cartMap)){
        //根据商品ID获取购物车信息
        String cartStr = cartMap.get(String.valueOf(cart.getGoodsId()));
        //如果商品存在, 修改商品数量和价格
        if (!StringUtils.isEmpty(cartStr)){
            CartVo cartVo = JsonUtil.jsonStr2Object(cartStr, CartVo.class);
            //修改商品数量
            cartVo.setGoodsNum(cartVo.getGoodsNum()+cart.getGoodsNum());
            //修改商品最新的价格
            cartVo.setMarketPrice(cart.getMarketPrice());
            //重新添加至map, 覆盖之前的商品对象
            cartMap.put(String.valueOf(cart.getGoodsId()), JsonUtil.object2JsonStr(cartVo));
        }else {
            //如果商品不存在, 新增购物车信息
            cartMap.put(String.valueOf(cart.getGoodsId()), JsonUtil.object2JsonStr(cart));
        }
    }else {
        //如果购物车信息为空, 新增购物车信息
        cartMap = new HashMap<>();
        cartMap.put(String.valueOf(cart.getGoodsId()), JsonUtil.object2JsonStr(cart));
    }
    hashOperations.putAll(userCart+":"+userId, cartMap);
    return BaseResult.success();
}

/**
 * 查询购物车数量
 *
 * @param admin
 * @return
 */
@Override
public Integer getCartNum(Admin admin) {
    //如果用户信息不存在, 直接返回
    if (null == admin || null == admin.getAdminId()) {
        return 0;
    }
    //初始化购物车数量
    Integer result = 0;
    //查询当前用户的购物车信息
    hashOperations = redisTemplate.opsForHash();
    Map<String, String> cartMap = hashOperations.entries(userCart + ":" +
admin.getAdminId());
    //如果购物车信息不为空, 累加数量
    if (!CollectionUtils.isEmpty(cartMap)){

```

```
        result = cartMap.values().stream().mapToInt(e -> {
            CartVo cartVo = JsonUtil.jsonStr2Object(e, CartVo.class);
            return cartVo.getGoodsNum();
        }).sum();
    }
    return result;
}
}
```

前台系统编写测试类消费服务

shop-portal的CartServiceITest.java

```
package com.xxxx.portal.service;

import com.alibaba.dubbo.config.annotation.Reference;
import com.xxxx.rpc.pojo.Admin;
import com.xxxx.rpc.service.CartService;
import com.xxxx.rpc.vo.CartVo;
import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

import java.math.BigDecimal;
import java.util.Date;

/**
 * 购物车测试类
 *
 * @author zhoubin
 * @since 1.0.0
 */
@SpringBootTest
public class CartServiceTest {

    @Reference(interfaceClass = CartService.class)
    private CartService cartService;

    /**
     * 测试加入购物车
     */
    @Test
    public void testAddCart() {
        Admin admin = new Admin();
        admin.setAdminId((short) 1);
        CartVo cartVo = new CartVo();
        cartVo.setGoodsId(12345);
        cartVo.setGoodsNum(10);
        cartVo.setMarketPrice(new BigDecimal("10000"));
        cartVo.setGoodsName("JAVA编程思想");
        cartVo.setAddTime(new Date());
        cartService.addCart(cartVo, admin);
    }
}
```

```
/**
 * 测试查询购物车数量
 */
@Test
public void testGetCartNum() {
    Admin admin = new Admin();
    admin.setAdminId((short) 1);
    Integer num = cartService.getCartNum(admin);
    System.out.println("购物车数量: " + num);
}
}
```

