



[返回首页](#)

芋道源码 —— 知识星球

我是一段不羁的公告！

记得给芳芳这 3 个项目加油，添加一个 STAR 噢。

<https://github.com/YunaiV/SpringBoot-Labs>

<https://github.com/YunaiV/onemall>

<https://github.com/YunaiV/ruoyi-vue-pro>

2018-03-25

[数据库实体设计](#)

数据库实体设计 —— 交易（2.6）之购物车

芳芳目前正在做一个开源的电商项目，胖友可以 star 下。

<https://gitee.com/zhijiantianya/onemall>

1. 概述

本文主要分享交易模块的[购物车](#)的数据库实体设计。

基于如下信息，逆向猜测数据库实体：

[有赞云提供的购物车API](#)

【护脸旁白】

笔者非电商行业出身 && 非有赞工程师，所以有错误或不合理的地方，烦请斧正和探讨。

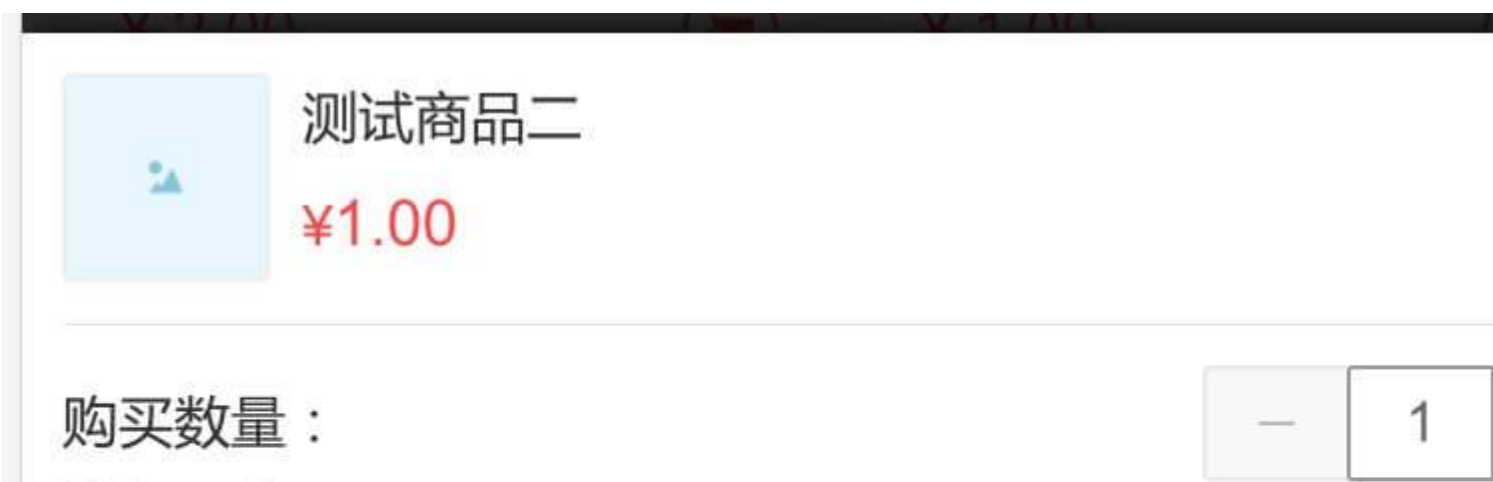
有赞是个各方面都很 NICE 的公司，[推荐](#)。

2. 背景了解

购物车的功能，相信旁友们已经用的灰常娴熟（手刹了多少次了），笔者就不瞎比比了。下面我们就简单来看看有赞的购物车界面。

2.1 界面

加入商品到购物车



如果商品设置了购买留言，需要填写后才可以添加到购物车。

购物车列表



测试商品二

¥1.00

购买数量：

剩余100件

留言

输入文本

输入购买的留

加入购物车

立即购买

部分加入到购物车的会出现的情况，例如商品被下架了。

2.2 实现

唯心六艺大神在 [《电商研发系列——购物车（五）购物车存储形式》](#) 一文已经分享了三种实现方式：

基于 Cookie

基于 Session 或者分布式缓存

基于 DB

笔者就不赘述了。另外阅读完本文，推荐在看看大神对购物车的完整梳理：[《电商研发系列——购物车》](#)。

注意，添加购物车的商品数据千万一定必须持久化到 DB 。好处如下：

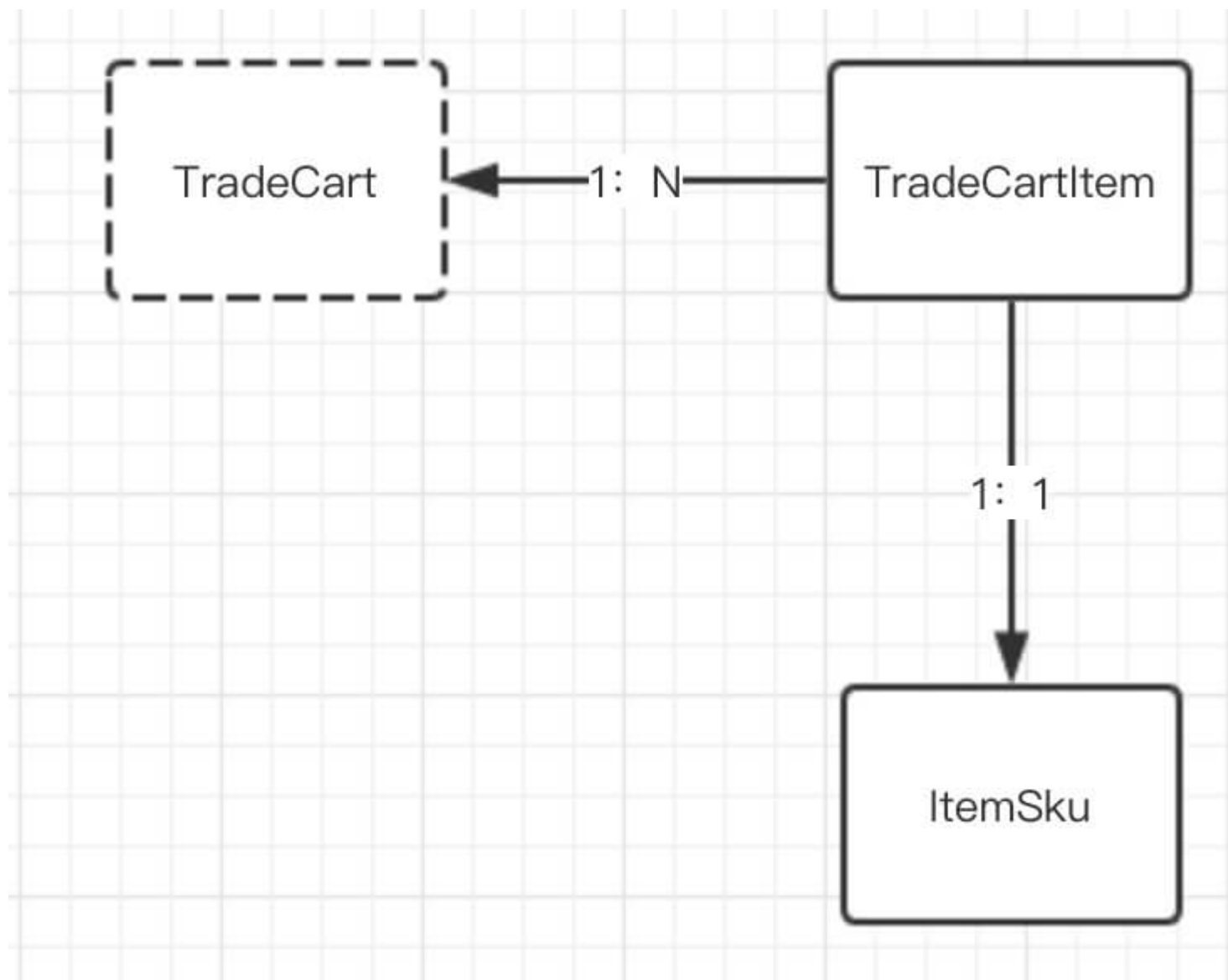
数据分析。例如，哪些商品被添加到购物车很多，但是很少被下单，实现转换率漏斗。

业务拓展。例如，卖家基于买家购物车的商品信息，进行回访，促发购买。

用户体验。缓存总会过期，可能买家目前仅仅是添加到购物车，N 天后会进行购买。

3. 数据库实体

整体实体类关系如下图：



TradeCart 是个虚拟实体。

- 每个已注册的买家会拥有自己的购物车。
- 每个未注册的买家（游客）会拥有自己的购物车。

TradeCartItem : **ItemSku** = 1: 1 。每种商品 SKU 添加到购物车后，生成一条 **TradeCartItem** 。

全部实体在 [Github 购物车实体目录](#) 下可见。

3.1 TradeCartItem

[TradeCartItem](#) ，购物车的商品信息。

虽然字段不多，但是考虑到和商品、交易能够更好的对应，所以分块。

3.1.1 基础字段

```
/**
 * 编号，唯一自增。
 */
private Integer id;
/**
 * 店铺编号
 */
private Integer shopId;
/**
 * 状态
 *
 * 1-正常
 * 2-主动删除
 * 3-下单删除
 */
private Integer status;
/**
 * 商品放入购物车时间
 */
private Date createTime;
/**
 * 商品在购物车中的更新时间
 */
private Date updateTime;
/**
 * 商品在购物车中的删除时间
 */
private Date deleteTime;
```

id ， 编号，唯一自增。
shopId ， 店铺编号，指向 Shop.id 。
status ， 状态。当商品从购物车删除时，标记 status = 2 。
createTime ， 商品放入购物车时间。
updateTime ， 商品在购物车中的更新时间。
deleteTime ， 商品在购物车中的删除时间。

3.1.2 买家信息

```
/**
 * 买家编号
 */
private Integer buyerId;
/**
 * 会话 key
 */
private String nobody;
```

nobody ， 会话 key 。在有赞中，无论你是否登陆，在 Cookie 中都有有一个 nobody_sign 标记当前会话。例如下图：

DO_CHECK_YOU_VERSION	1
HMACCOUNT	07DF99F82E36753
Hm_lpvt_679ede9eb28bacfc763976b1...	1518230517
Hm_lvt_679ede9eb28bacfc763976b10...	1518189591
1518230517	1518230517

- nobody 的生成规则笔者猜不出来，目前看起来分成两段，以 YZ411688455316406272YZJsb0cl6s 举例子。
 - YZ + 411688455316406272 （18位数字）
 - YZ + Jsb0cl6s （8位16进制）
- 从游客登陆或注册成用户，nobody 不变。

buyerId ，买家编号。当游客登陆成用户时，数据库中对应的 nobody 的记录需要更新 buyerId 字段为当前用户。

3.1.3 商品信息

```
/**
 * 商品编号
 */
private Integer itemId;
/**
 * 商品 SKU 编号
 */
private Integer skuld;
/**
 * 商品购买数量
 */
private Integer num;
/**
 * 交易明细中买家留言的数据结构 {@link cn.iocoder.doraemon.itemgroup.item.entity.Item#messages}
 */
private String buyerMessages;
```

在商品添加到购物车后，卖家将商品信息发生变更后，买家刷新购物车后，读取到的是最新的信息。

3.1.4 交易信息

```
/**
 * 订单编号
 */
private String tid;
/**
 * 下单时间
 */
private Date buyTime;
```

tid ，订单编号，指向 Trade.id 。

buyTime ，下单时间。

3.1.5 优惠信息

```
/**
 * 商品营销活动编号
 */
private Integer activityId;
/**
 * 商品营销活动类型
```

```
*/  
private Integer activityType;
```

// TODO 6013 购物车的优惠信息

- 当某个参加优惠活动的商品添加到购物车后，卖家将商品的优惠活动发生变更后，买家刷新购物车后，读取到的是最新的优惠信息。
 - 那么是否有必要存储该字段？因为笔者看到 [添加商品至购物车API](#) 有 `activity_id` 等优惠活动相关参数，所以笔者认为会有存储。
 - 那么究竟有什么意义？目前笔者能够猜测到的是数据分析。如果有胖友知道，请告诉下，谢谢。

3.1.6 价格信息

// TODO 6011 金额计算

4. API

基于如下整理 API 类。

[有赞云提供的购物车API](#)

4.1 TradeCartAPI

[TradeCartAPI](#)，购物车 API。

```
/**  
 * 购物车 API  
 */  
public interface TradeCartAPI {  
  
    /**  
     * 添加商品至购物车  
     *  
     * https://www.youzanyun.com/apilist/detail/grou  
     *  
     * @param buyerId 买家编号  
     * @param nobody 会话 key  
     * @param shopId 店铺编号  
     * @param itemId 商品编号  
     * @param skuId 商品 SKU 编号  
     * @param num 商品数量  
     * @param message 购买留言  
     * @param activityId 商品营销活动编号
```

666. 彩蛋

本文目前写的是相对比较浅的，结合商品参加的优惠活动，会有比较复杂的优惠价格计算逻辑。

未来，笔者会在这块做进一步补充。

// TODO 6011 金额计算

文章目录

1. [1. 1. 概述](#)
2. [2. 2. 背景了解](#)
 1. [2. 1. 2. 1 界面](#)
 2. [2. 2. 2. 2 实现](#)
3. [3. 3. 数据库实体](#)
 1. [3. 1. 3. 1 TradeCartItem](#)
 1. [3. 1. 1. 3. 1. 1 基础字段](#)
 2. [3. 1. 2. 3. 1. 2 买家信息](#)
 3. [3. 1. 3. 3. 1. 3 商品信息](#)
 4. [3. 1. 4. 3. 1. 4 交易信息](#)
 5. [3. 1. 5. 3. 1. 5 优惠信息](#)
 6. [3. 1. 6. 3. 1. 6 价格信息](#)
4. [4. 4. API](#)
 1. [4. 1. 4. 1 TradeCartAPI](#)
5. [5. 666. 彩蛋](#)