



[回到首页](#)

## 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>

2018-01-30

[数据库实体设计](#)

## 数据库实体设计 —— 商品（1.4）之商品快照

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/zhi\\_jiantianya/onemall](https://gitee.com/zhi_jiantianya/onemall)

### 1. 概述

本文主要分享商品模块的[商品快照](#)的数据库实体设计。

由于有赞云暂未提供商品快照相关 API，因此笔者基于有赞、淘宝、京东的商品快照功能涉及到的界面做推断。如有错误的地方，还望斧正。

【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

我们先来看看商品快照在常见的电商平台的应用。

#### 2.1 有赞

在商家的订单详情页，可以看到售出的每种商品的商品交易快照的链接，界面如下图：



链接最后的参数 `snap_key`，即是商品快照的编号，32 位字符串。

目前猜测生成的规则为，将商品的信息根据按照字段的按照统一顺序（例如字段名字字符串的升序），拼接成字符串，MD5 该字符串生成 32 位唯一编号。

所以，同一个商品，修改了字段值，而后又修改回来，对应同一个商品快照。例如，商品A 对应的商品快照为 `a`，将标题从 A 修改成 B 后，对应的商品快照为 `k`；而后将标题从 B 修改回 A，对应的商品快照为 `a`。

那么商品快照是什么时候生成的呢？有两个地方可以生成：

1. 商品创建或修改时
2. 商品下单时

笔者偏向【第一种】，原因如下：

1. 商品快照的功能和商品信息的功能更近，适合开发商品模块的工程师统一维护。并且未来商品信息的字段有变更，避免做商品交易的工程师需要一起对接联调。
2. 商品快照不仅仅用于商品交易，未来可以提供给其他功能使用，例如商品信息的变更历史等等。

在买家的订单详情页，暂时看不到商品交易快照，猜测暂时未开放给买家。

## 2.2 京东

在买家的订单详情页，暂时看不到商品交易快照，猜测暂时未开放给买家。这一点和有赞相同。

在商家的订单详情页，可以看到售出的每种商品的商品交易快照的链接，参见 [《京东商家帮助中心——商品快照自助查询操作说明》](#)。这一点和有赞相同。

## 2.3 淘宝

在商家的订单详情页，可以看到售出的每种商品的商品交易快照的链接，界面如下图：



和有赞不同，在链接中仅看到 `snapShot=true` 参数，并未看到商品快照编号。实际上，在商品交易订单上，我们会存储购买商品的商品快照编号。因此，此处即使没有商品快照编号，也可以通过 `tradeID` 查询到该编号。

商品快照详情页如下图：



在页面上，我们可以看到，商品快照的目的：

当前页面内容为[订单快照](#)，包含订单创建时的商品描述和下单信息，买卖双方和平台在发生交易争议时，该页面作为判定依据。

我们知道实际开发中，商品信息的字段会变更，可能展示界面会存在无法兼容的问题，在这种情况下，淘宝会提示但目前暂时不提供查看该商品的信息。如下图：



在买家的订单详情页，可以看到商品交易快照，如下图：

☐ **2018-01-06** 订单号: 111436816199631161



Lee国内专柜代购17年秋冬新款男士棉服L248182SA6RH [交易快照]

颜色: L248182SA3BL[灰蓝] 尺码: XL



保险服务

☐ **2017-12-21** 订单号: 105146170551631161



正品外销4D蜘蛛侠美国队长 钢铁侠 绿巨人陶瓷杯送男友礼物 [交易快照]

颜色分类: 蜘蛛侠1



保险服务

每次拿着

都有一种

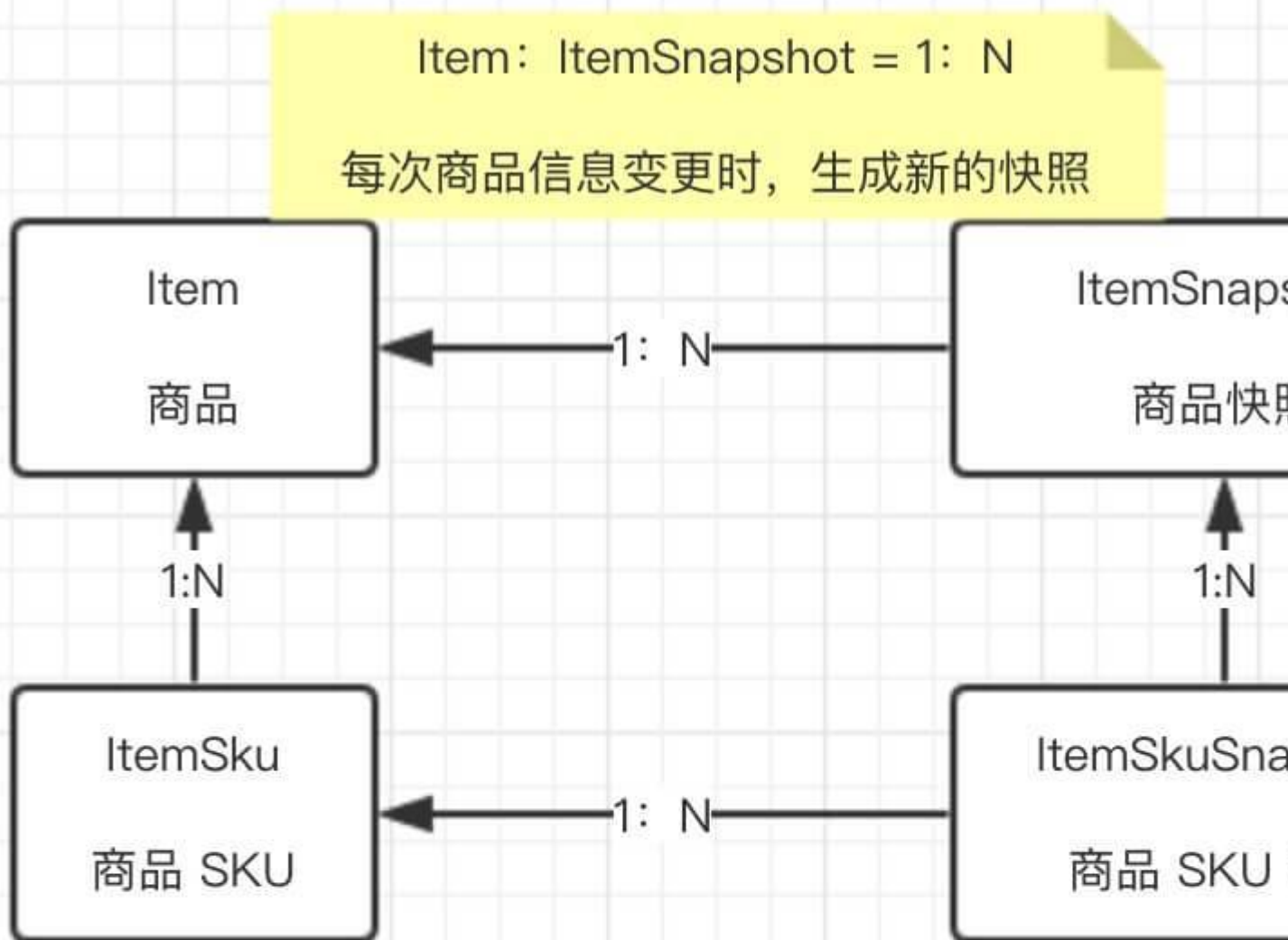
☐ **2017-11-28** 订单号: 98074702619631161

哈哈哈哈哈



### 3. 数据库实体

整体实体类关系如下图：



此处省略商品的关联表，例如 ItemContent(商品描述)。  
每个关联表，都有其对应的快照

全部实体在 [Github 商品快照实体目录](#) 下可见。

高能提示：快照实体从其对应的实体类复制了相同的字段，若该字段无注释，则代表是复制过来的。例如：

```
/**
 * 商品快照，对应 {@Link cn.iocoder.doraemon.itemg
 */
public class ItemSnapshot {

    /**
     * 快照编号
     */
    private String id;

    /**
     * 商品编号 {@Link cn.iocoder.doraemon.itemgro
     */
    private Integer itemId;
    private String alias;

    /**
     * 快照时间
     */
    private Date createTime;

    // ===== 基本信息 BEGIN =====

    private String title;
    private String summary;
    private Integer cid;
    private String picURLs;
    private Integer itemType;
    private Integer goodsType;
```

复制

### 3.1 ItemSnapshot

[ItemSnapshot](#) ，商品快照，对应 Item 实体。

如下仅列举重要字段，代码如下：

```
/**
 * 快照编号
 */
private String id;
/**
 * 商品编号 {@link cn.iocoder.doraemon.itemgroup.item.entity.Item#id}
 */
private Integer itemId;
/**
 * 是否当前商品正在使用的快照。
 */
private Boolean isNewest;
/**
 * 快照时间
 */
private Date createTime;
/**
 * 总库存
 *
 * 【不影响 {@link #id}】的计算
 */
private Integer quantity;
/**
 * 总销量
 *
 * 【不影响 {@link #id}】的计算
 */
private Integer soldNum;
```

id ，商品快照编号。生成方法，在 [「 2.1 有赞 」](#) 我们已经分享。

itemId ，商品编号，指向 Item.id 。

isNewest ，是否当前商品正在使用的快照。商品使用的并非一定是最后生成的商品快照。

createTime ，快照生成时间。

并非所有的商品信息字段都影响 id 的生成，例如：quantity / soldNum。这些字段偏向统计而非商品信息。

## 3.2 ItemSkuSnapshot

[ItemSkuSnapshot](#) ，商品 SKU 快照，对应 ItemSku 实体。

如下仅列举重要字段，代码如下：

```
/**
 * 商品快照编号 {@link ItemSnapshot#id}
 */
private Integer snapshotId;
// private Integer status; // 无需状态，因为 sku 变更后，会产生新的 ItemSnapshot
/**
 * 库存数量
 *
 * 【不影响 {@link ItemSnapshot#id}】的计算
```



```

    */
private Integer quantity;
/**
 * 商品在付款减库存的状态下，该Sku上未付款的订单数量
 *
 * 【不影响 {@link ItemSnapshot#id}】的计算
 */
private Integer withHoldQuantity;
/**
 * 销量
 *
 * 【不影响 {@link ItemSnapshot#id}】的计算
 */
private Integer soldNum;
/**
 * 快照时间
 */
private Date createTime;

```

snapshotId ，商品快照编号，指向 ItemSnapshot.id 。

status ，无需状态，因为 SKU 变更后，会产生新的 ItemSnapshot 。

createTime ，快照生成时间。

并非所有的商品信息字段都影响 id 的生成，例如：quantity / soldNum / withHoldQuantity 。这些字段偏向统计而非商品信息。

### 3.3 ItemContentSnapshot

[ItemContentSnapshot](#) ，商品详情，对应 ItemContent 实体。

如下仅列举重要字段，代码如下：

```

/**
 * 快照编号 {@link ItemSnapshot#id}
 */
private Integer id;
/**
 * 商品编号
 *
 * {@link cn.iocoder.doraemon.itemgroup.item.entity.Item#id}
 */
private Integer itemId;

```

id ，商品快照编号，指向 ItemSnapshot.id 。

itemId ，商品编号，指向 Item.id 。

createTime ，快照生成时间。

### 3.4 Item 其他关联表快照

Item 除了有 ItemContentSnapshot 一对一的关联表外，还有 ItemEtd 、ItemFenxiao 、ItemHotel 、ItemPurchaseRight 等等，都需要一一对一个快照实例。例如，ItemEtdSnapshot 。

那么是不是和商品相关的表都需要快照呢？

笔者觉得不是。例如，ItemSkuProperty，ItemCategory。因为即使ItemCategory的名字改变了，在系统中对应的本质还是相同的。

当然ItemTemplate变更是需要的，因为影响到了ItemContent的信息。

## 3.5 Item

我们需要在 [Item](#) 上，新增 [snapshotId](#) 字段，指向 ItemSnapshot。

## 3.6 TradeOrder

本小节，笔者建议胖友先看完 [《数据库实体设计 —— 交易（2.1）之订单信息》](#)。

我们需要在 [TradeOrder](#) 上，新增 [snapshotId](#) 字段，指向 ItemSnapshot。

# 4. API

由于有赞云暂未提供商品快照相关 API，相信胖友能够想象。

# 666. 彩蛋

最后，但是非常重要。

如果有错误的地方，欢迎胖友帮我斧正！！！！

文章目录

1. [1. 1. 概述](#)
2. [2. 2. 背景了解](#)
  1. [2.1. 2.1 有赞](#)
  2. [2.2. 2.2 京东](#)
  3. [2.3. 2.3 淘宝](#)
3. [3. 3. 数据库实体](#)
  1. [3.1. 3.1 ItemSnapshot](#)
  2. [3.2. 3.2 ItemSkuSnapshot](#)
  3. [3.3. 3.3 ItemContentSnapshot](#)
  4. [3.4. 3.4 Item 其他关联表快照](#)
  5. [3.5. 3.5 Item](#)
  6. [3.6. 3.6 TradeOrder](#)
4. [4. 4. API](#)
5. [5. 666. 彩蛋](#)