



[回到首页](#)

芋道源码 —— 知识星球

我是一段不羁的公告！

记得给芬芳这 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>

2018-07-01

[数据库实体设计](#)

数据库实体设计 —— 营销（4.1）之优惠券&&优惠码

芬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

本文主要分享营销模块的 [优惠券](#) 和 [优惠码](#) 的数据库实体设计。

促销折扣



优惠券 

向客户发放店铺优惠券



优惠

码



秒杀 

快速抢购引导顾客更多消费



支

客



打包一口价

给我N元，这里的商品随便挑

基于如下信息，逆向猜测数据库实体：

- [有赞云提供的优惠券/码API](#)
- [有赞微商城的优惠券](#)
- [有赞微商城的优惠码](#)
- [有赞微商城的验证工具](#)

【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

2.1 优惠券管理

参见 [《优惠券的操作和玩法》](#) 文档。



友情提示：同步微信卡券部分，不在本文范围内。

2.2 优惠码管理

参见 [《如何创建优惠码?》](#) 文档。

返回

优惠码

优惠码标题

分享

¥ 0.00

订单满 xx 元 (含运费)

有效日期: 20xx : 00 : 00 - 20xx : 00 : 00

使用说明

暂无使用说明.....

优惠码名称: 最多支持10个字

码类型: ☒ 一卡一码 ☐ 通用码

生成方式: ☒ 自动生成 ☐ 导入码库

库存: 张

面值: 元

订单金额: ☐ 不限制 ☒ 满 元可使用

同步发布至: ☐ 微信卡券(包) [帮助](#)
同步至微信卡包后, 需等待微信审核通过, 才能领取 (审核预计三个工作日内完成)

基本规则

每人限额: 1张

生效时间:

过期时间:

同步打标签: [刷新](#) [新建](#)

到期提醒: ☐ 到期前4天提醒一次

推广设置: ☒ 允许分享领取链接

可使用商品: ☒ 全店商品通用 ☐ 指定商品

限制购买: ☐ 仅原价购买商品时可用

使用说明:

1、编辑活动名称

2、设置优惠码的形式

3、选择系统生成, 进

4、活动要发放优惠

5、不限制金额, 即开

限制后, 订单满足条

6、设置同步卡券, 买

在微信卡包进行

7、设置领取次数

用户只能领取1

8、优惠券的有

优惠券会显示过期

优惠券失效, 则不

9、认证服务号授权的店铺,

可以给领取的用户打标签,

在店铺后台, 客户-标签管理

进行查看领取该优惠券的用户

10、认证服务

户会收到提醒;

11、勾选后, 用户可以分享优惠券给

朋友领取

12、可以选择

13、勾选后, 参加其他优惠

从界面上，我们可以看出，优惠码和优惠券的配置大体类似。在下面的数据实体的设计，我们会从字段上标记出来。

2.3 下单使用优惠

买家下单时，可以使用优惠券或优惠码，进行使用。

2.4 验证使用优惠

买家除了下单可以使用优惠券（码），也可以出示该券（码）的二维码或校验码，被卖家使用 APP / 后台，扫一扫或输入，核检使用。界面如下：

1. 【APP】卖家输入校验功能



2. 【后台】卖家输入校验功能

全 | <https://www.youzan.com/v2/ump/verifycard#/>

销

营销概况 / 验证工具

验证卡券

优惠券验证记录

优惠码验证记

优惠券/优惠码



请输入或搜索优

< 返回

优惠券



贝塔朋友咖啡吧



我的记录

关注

贝塔咖啡代金券

分享

¥ 66.00

满100元可使用

有效期: 2015.01.02 - 2015.06.06

立即使用

添加微信卡包

[查看我的卡券](#)

3. 【后台】验证记录列表

安全

https://www.youzan.com/v2/ump/verifycard#

应用和营销

营销概况 / 验证工具

营销概况

验证卡券

优惠券验证记录

优惠券

主要推广

验证卡券

开放服务

卡券名称: 请选择

核销码:

筛选

导出验证记录

应用市场

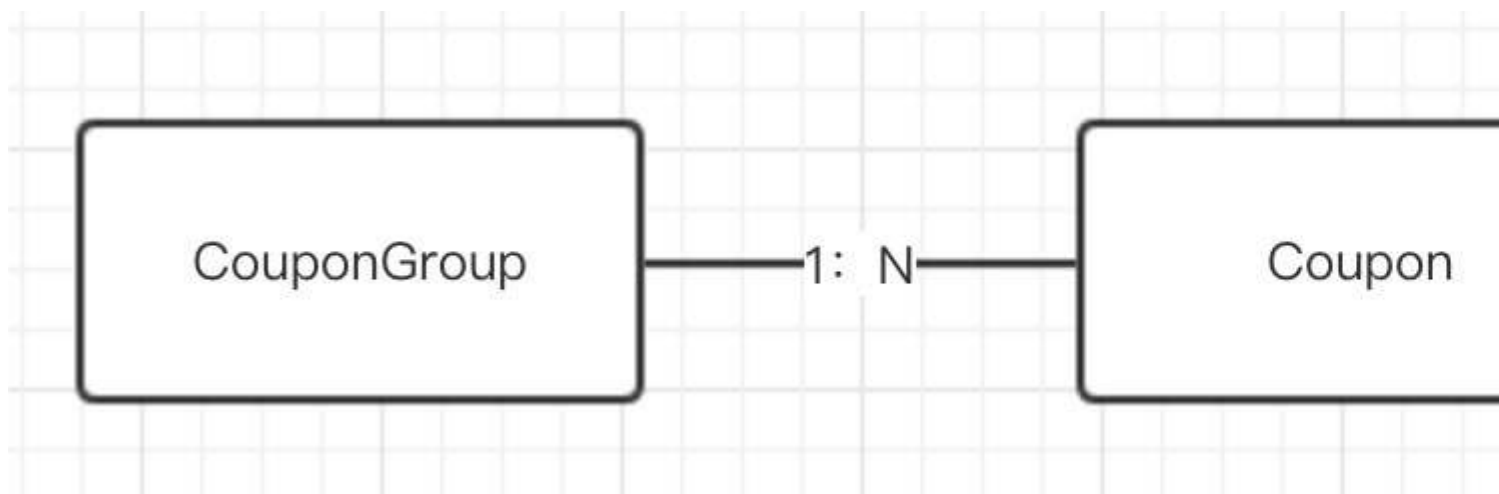
时间

2018-02-11 18:23:33

营销市场

3. 数据库实体

整体实体类关系如下图：



CouponGroup：卖家配置完优惠券（码）时，生成优惠券（码）分组。

Coupon：优惠券（码）。根据不同的 CouponGroup 配置的不同，买家获得 Coupon。

- 优惠券
 - 卖家添加完 CouponGroup 后，不生成任何 Coupon 记录。
 - 买家领取时，生成 Coupon 记录。
- 优惠码【一卡一码】
 - 卖家添加完 CouponGroup 后，自动生成所有的 Coupon 记录，并且每个 Coupon 的优惠码不同。
 - 买家输入优惠码后，设置对应的 Coupon 属于买家。
- 优惠码【通用码】
 - 卖家添加完 CouponGroup 后，自动生成一条 Coupon 记录。
 - 买家输入兑换码后，设置该条 Coupon 属于买家，并自动生成新一条 Coupon 记录，直到 CouponGroup 库存上限。
 - 买家使用兑换码下单后，若买家允许领取多张优惠券，自动生成新一条 Coupon 记录归属该卖家，直到买家领取上限。

因此，CouponGroup：Coupon = 1：N。

CouponVerifyLog：仅 [「2.4 验证使用优惠」](#) 时，生成一条 CouponVerifyLog 记录。

全部实体在 [Github 优惠券\(码\) 实体目录](#) 下可见。

3.1 CouponGroup

[CouponGroup](#)，优惠券（码）组。优惠券和优惠券的配置，都存储在 CouponGroup 中。

下面，我们分块来介绍每个字段。

3.1.1 基本信息

```
/**
 * 编号，自增唯一。
 */
private Integer id;
```

```

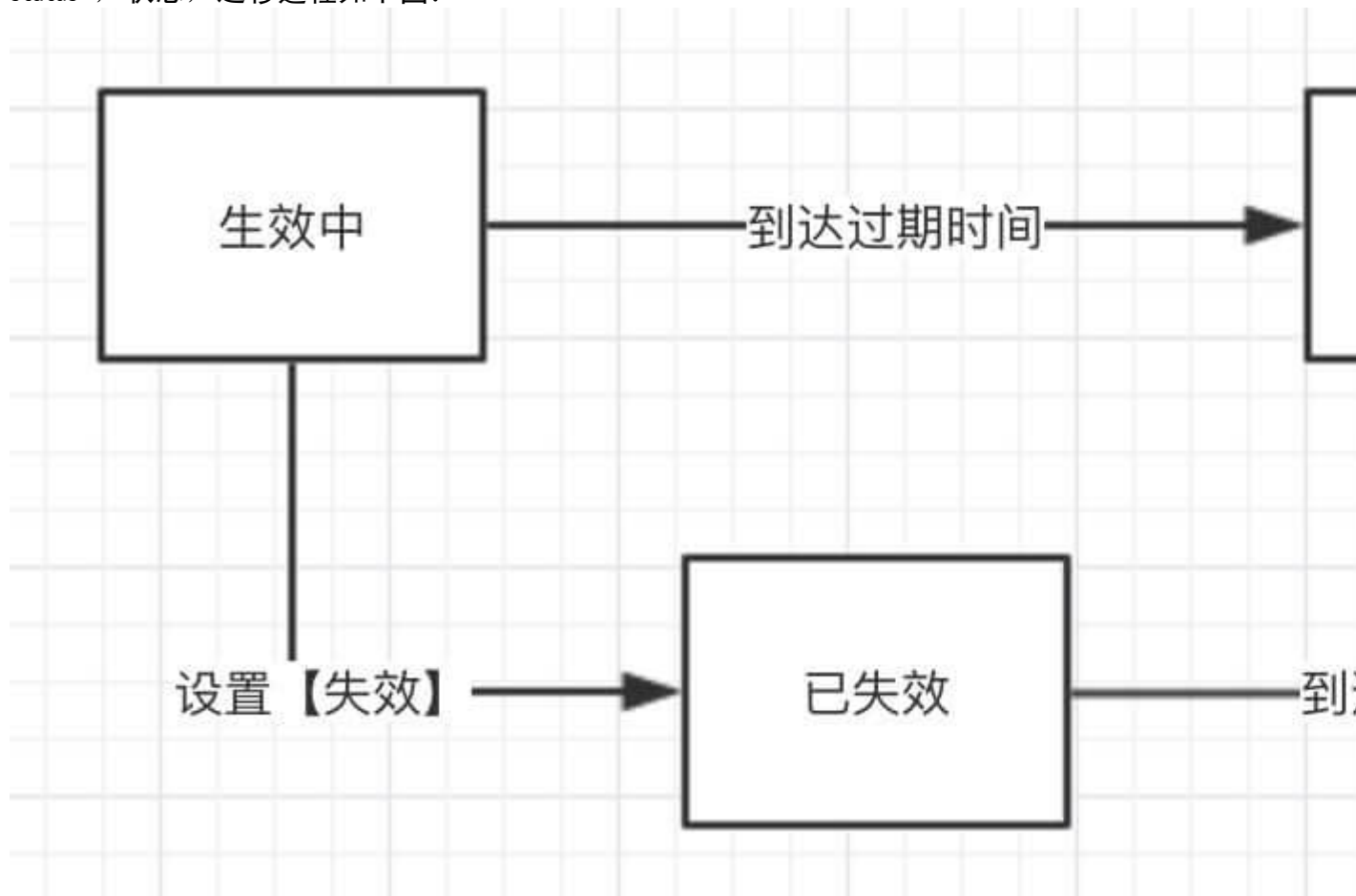
/**
 * 店铺编号
 */
private Integer shopId;
/**
 * 别名
 *
 * 系统生成，作为唯一标识。例如，2fpa62tbmsl9h
 * 可以用于生成优惠券（码）领取链接，例如，https://wap.youzan.com/v2/showcase/coupon/fetch?alias=17xcvjbd8
 */
private String alias;
/**
 * 标题
 */
private String title;
/**
 * 使用说明
 */
private String description;
/**
 * 类型
 *
 * 1-优惠券
 * 2-优惠码
 */
private Integer type;
/**
 * 创建时间
 */
private Date createTime;
/**
 * 更新时间
 */
private Date updateTime;
/**
 * 优惠码状态
 *
 * 1-有效
 * 2-已失效
 * 3-已过期
 * 4-已删除
 *
 * 当优惠券（码）有效时，可以手动操作，设置成无效。
 */
private Integer status;
/**
 * 是否可分享领取链接
 */
private Boolean isShare;
/**
 * 设置为失效时间
 */
private Date invalidTime;
/**
 * 删除时间
 */
private Date deleteTime;

```

id ， 分组编号，自增唯一。

shopId ， 店铺编号，指向 Shop.id 。

alias ， 别名，系统生成，作为唯一标识。例如，2fpa62tbmsl9h 。可以用于生成优惠券（码）领取链接，保证安全性，例如，优惠码领取地址
<https://wap.youzan.com/v2/showcase/coupon/fetch?alias=17xcvjbd8> 。
title ， 标题。
description ， 使用说明。
type ， 类型，用于标记是优惠券还是优惠码。
createTime ， 创建时间。
updateTime ， 更新时间。
status ， 状态，迁移过程如下图：



- 生效中：卖家创建完 CouponGroup ， 处于该状态。
- 已失效：卖家手动设置 CouponGroup 为已失效，此处买家已领取 Coupon 依然有效，但是不能领取新的 Coupon 。参见 [《如何让已领取的优惠券失效掉？》](#) 文档。
- 已过期：当 CouponGroup 到达过期时间时，变更为已过期。当然也因此，仅有效期类型为固定日期的 CouponGroup 能自动过期。
- 已删除：当 CouponGroup 处于已过期时，卖家方可手动删除。删除效果参见 [《优惠券可以删除吗？》](#) 文档。

invalidTime ， 设置为已失效状态的时间。
deleteTime ， 设置为已删除状态的时间。
isShare ， 是否可分享领取链接。

3.1.2 码信息

```
/**  
 * 码类型  
 *  
 * 1--卡一码（UNIQUE）  
 * 2-通用码（GENERAL）
```

```

*
* 【优惠券独有】
*/
private Integer codeType;
/**
* 优惠券
*
* 【优惠券独有】
*/
private String code;

```

优惠券独有。

codeType ， 优惠券类型。

code ， 优惠券，当优惠券类型为通用码时设置。另外，优惠券类型为一卡一码时，每个优惠券在 Coupon 中存储。

3.1.3 领取规则

```

/**
* 是否限制领用者的等级
*
* 0-不限制
* 大于0-领用者必须是这个等级编号
*
* 【优惠券独有】
*/
private Integer needUserLevel;
/**
* 每人限领个数
*
* 0-则表示不限制
*/
private Integer quota;
/**
* 剩余可用库存
*/
private Integer stock;
/**
* 总库存
*/
private Integer total;
/**
* 领取优惠券要给用户打上的标签的列表，使用逗号分隔标签编号
*/
private String markTags;

```

3.1.4 使用规则

```

/**
* 是否仅原价购买商品时可用
*
* true-是
* false-否
*/

```

```

private Boolean isForbidPreference;
/**
 * 是否设置满多少金额可用，单位：分
 *
 * 0-不限制
 * 大于0-多少金额可用
 */
private Integer condition;
/**
 * 可用范围的类型
 *
 * 1-部分（ALL）：全部商品可用
 * 2-全部（PART）：部分商品可用，或指定商品可用
 */
private Integer rangeType;
/**
 * 指定可用商品列表，使用逗号分隔商品编号 {@link cn.iocoder.doraemon.itemgroup.item.entity.Item#id}
 */
private String rangeValues;
/**
 * 生效日期类型
 *
 * 1-固定日期
 * 2-领取日期：领到券 {@link #fixedBeginTerm} 日开始 N 天内有效
 */
private Integer dateType;
/**
 * 固定日期-生效开始时间
 */
private Date validStartTime;
/**
 * 固定日期-生效结束时间
 */
private Date validEndTime;
/**
 * 领取日期-开始天数
 *
 * 例如，0-当天；1-次天
 */
private Integer fixedBeginTerm;
/**
 * 领取日期-结束天数
 */
private Integer fixedTerm;
/**
 * 是否到期前4天发送提醒
 *
 * true-发送
 * false-不发送
 */
private Boolean expireNotice;

```

价格维度

- isForbidPreference / condition

商品维度

- rangeType / rangeValues

时间维度

- dateType
- validStartTime / validEndTime

- fixedBeginTerm / fixedTerm
- expireNotice

3.1.5 使用效果

```

/**
 * 优惠类型
 *
 * 1-代金卷
 * 2-折扣卷 【优惠券独有】
 */
private Integer preferentialType;
/**
 * 折扣。例如，80% 为 80。
 *
 * 【优惠券独有】
 */
private Integer discount;
/**
 * 是否是随机优惠券
 *
 * true-随机
 * false-不随机
 *
 * 【优惠券独有】
 */
private Boolean isRandom;
/**
 * 优惠金额，单位：分
 *
 * 当 [link #isRandom] 为 true 时，代表随机优惠金额的下限
 */
private Integer value;
/**
 * 优惠金额上限
 *
 * 【优惠券独有】
 */
private Integer valueRandomTo;

preferentialType
discount
isRandom / value / valueRandomTo

```

3.1.6 统计信息

```

/**
 * 领取优惠券的人数
 */
private Integer statFetchUserNum;
/**
 * 领取优惠券的次数
 */
private Integer statFetchNum;
/**
 * 使用优惠券的次数

```

```
*/  
private Integer statUseNum;
```

3.1.7 微信卡卷

```
/**  
 * 是否同步微信卡券  
 *  
 * true-是  
 * false-否  
 */  
private Boolean isSyncWeixin;  
/**  
 * 同步微信卡券，选择的卡券颜色的值。  
 *  
 * 例如，Color10  
 */  
private String weixinColor;  
/**  
 * 同步微信卡券，选择的卡券颜色的RGB值  
 *  
 * 例如，#ffaaff  
 */  
private String weixinColorRGB;  
/**  
 * 同步微信卡券，卡券的标题  
 */  
private String weixinTitle;  
/**  
 * 同步微信卡券，卡券的副标题  
 */  
private String weixinSubTitle;  
/**  
 * 同步微信卡券，卡券的客服电话  
 */  
private String servicePhone;  
/**  
 * 同步微信卡券，设置是否可以转赠  
 *  
 * true-可以  
 * false-不可以  
 */  
private Boolean canGiveFriend;
```

// TODO 6014 微信卡卷，这块逻辑笔者不熟悉，并且不太合适做测试。

3.1.8 创建后可编辑属性

优惠券（码）分组创建后，大部分属性不可编辑，否则会影响已发放优惠券。如下是可编辑的属性：

```
title  
description  
total  
expireNotice  
markTags
```

设置店铺优惠券



(有赞微商城优惠券)

未开始

分享

¥ 10

订单满 100.00 元 (不含运费)

有效日期: 2018-02-13 11:38:55 - 2018-02-22 11:38:55

使用说明

暂无使用说明.....

优惠

* 优

同

基本

3.2 Coupon

[Coupon](#) ，优惠券（码）。

下面，我们继续分块来介绍每个字段。

3.2.1 基本信息

```
/**
 * 卡券ID
 */
private Integer id;
/**
 * 店铺编号
 */
private Integer shopId;
/**
 * 类型
 *
 * 1-优惠券
 * 2-优惠码
 */
private Integer type;
/**
 * 优惠券（码）分组编号，{@link CouponGroup#id}
 */
private Integer couponGroupId;
/**
 * 核销码
 */
private String verifyCode;
/**
 * 创建时间
 */
private Date createTime;
/**
 * 优惠码状态
 *
 * 1-生效中
 * 2-已失效
 * 3-已过期
 * 4-已删除
 * 5-已使用
 */
private Integer status;
```

id ，优惠券（码）编号。

shopId ，店铺编号，指向 Shop.id 。

type ，类型，用于标记是优惠券还是优惠码。

couponGroupId ，分组编号，指向 CouponGroup.id 。

verifyCode ，核销码，用于 [「2.4 验证使用优惠」](#) 。

createTime ，创建时间。

status ，状态。相比 CouponGroup.status ，增加了已使用状态。

3.2.2 码信息

```
/**
 * 码类型
 *
 * 1-一卡一码 (UNIQUE)
 * 2-通用码 (GENERAL)
 *
 * 【优惠券独有】
 */
private Integer codeType;
/**
 * 优惠券
 *
 * 【优惠券独有】
 */
private String code;
```

code ， 优惠券。

- 优惠券格式要求为 6 到 12 位数字组合。
- 一卡一码的情况下，可自动生成优惠券（目前为 12 位），也可以手动批量上传。自动生成的优惠券如下图：

营销概况 / 优惠券

所有优惠码

未开始

进行中

已结束

来一发

|

+添加库存

	优惠码	面额(元)	创建时间
<input type="checkbox"/>	4692 0857 3412	100	2018-02-11 00:48:11
<input type="checkbox"/>	6063 0289 2392	100	2018-02-11 00:48:11
<input type="checkbox"/>	2720 1339 0242	100	2018-02-11 00:48:11

- 通用码时，手动输入优惠码。

那么问题就来了，兑换码怎么自动生成呢？

我们来考虑下兑换码需要保证的几个方面：

1. 唯一性。需要保证输入兑换码后，获得唯一对应的优惠码。那么这里有一个问题，考虑到用户的体验，兑换码不能涉及的过长，且最好是纯数字。那么已经失效的优惠码的兑换码是否可以重用？笔者认为，可以。只要保证唯一兑换即可。
2. 随机性。需要保证兑换码不能轻易的被猜测到，否则会很危险。从有赞生成的兑换码，我们基本是看不出任何规则，很有可能，就是完全随机数字。
3. 安全性。在用户兑换优惠码时，我们需要做一些安全的限制，类似用户输入密码。例如，连续输错 N 次兑换码，一小时内不允许再次重入，防止暴力破解。当然，恶意用户如果一次就输入正确，这个也是没有办法的。因为，没有绝对的安全。

瞎比比了这么多，简单的说，就是纯随机 + 数据库保证唯一 + 兑换时安全限制。

3.2.3 领取情况

```
/**
 * 是否领取
 */
private Boolean isTake;
/**
 * 领取用户编号
 */
private Integer userId;
/**
 * 领取时间
 */
private Date takeTime;
```

3.3.4 使用规则

```
/**
 * 固定日期-生效开始时间
 */
private Date validStartTime;
/**
 * 固定日期-生效结束时间
 */
private Date validEndTime;
```

其他使用规则，从 CouponGroup 查询。

3.3.5 使用效果

```
/**
 * 优惠类型
 *
 * 1-代金券
 * 2-折扣卷 【优惠券独有】
```

```

    */
private Integer preferentialType;
/**
 * 折扣
 */
private Double discount;
/**
 * 优惠金额，单位：分。
 */
private Integer value;

```

3.3.6 使用情况

```

/**
 * 是否使用
 */
private Boolean isUsed;
/**
 * 使用订单号
 */
private String usedInTid;
/**
 * 订单中优惠面值，单位：分
 */
private Integer usedValue;
/**
 * 使用时间
 */
private Date usedTime;

```

isUsed ， 是否使用。

usedInTid ， 使用交易编号。当使用方式为：

- [「2.3 下单使用优惠」](#) ， 指向 Trade.id 。
- [「2.4 验证使用优惠」](#) ， 猜测， 交易编号不是指向 Trade.id ， 订单号例子为
“M2018021200400722216” 。

usedValue ， 在交易中优惠的金额。

usedTime ， 使用时间。

3.3 CouponFetchLog

// TODO 3.3 CounponFetchLog

从 [获取优惠券/优惠码领取记录](#) 接口中，看起来是有一个 CounponFetchLog 实体。但是从功能上，Coupon 完全可以满足。有经验的胖友，麻烦告知下。

3.4 CouponVerifyLog

[CouponVerifyLog](#) ， 优惠券（码）核销记录。

```

/**
 * 记录编号，唯一自增

```

```
    */
private Integer id;
/**
 * 店铺编号
 */
private Integer shopId;
/**
 * 创建时间（验证时间）
 */
private Date createTime;
/**
 * 类型
 *
 * 1-优惠券
 * 2-优惠码
 */
private Integer type;
/**
 * 优惠券（码）分组编号，{@link CouponGroup#id}
 */
private Integer couponGroupId;
/**
 * 标题
 */
private String title;
/**
 * 使用说明
 */
private String description;
/**
 * 优惠券（码）编号，{@link Coupon#id}
 */
private Integer couponId;
/**
 * 核销码
 */
private String verifyCode;
/**
 * 验证方式
 *
 * 1-扫码
 * 2-手动输入
 */
private Integer verifyType;
/**
 * 验证管理员编号
 */
private Integer adminId;
/**
 * 是否设置满多少金额可用，单位：分
 *
 * 0-不限制
 * 大于0-多少金额可用
 */
private Integer condition;
/**
 * 优惠类型
 *
 * 1-代金券
 * 2-折扣卷

```

```

    */
private Integer preferentialType;
/**
 * 折扣
 */
private Double discount;
/**
 * 优惠金额，单位：分
 */
private Integer value;
/**
 * 使用订单号
 */
private String usedInTid;

```

整体字段和 Coupon 类似，我们只看几个区别字段。
 id ，记录编号，唯一自增。
 verifyType ，输入方式，有扫码和手动输入两种类型。
 adminId ，验证管理员编号。

4. API

基于如下整理 API 类。

[有赞云提供的优惠券/码API](#)

4.1 CouponGroupAPI

[CouponGroupAPI](#) ，优惠券（码）组 API 。

```

/**
 * 优惠券（码）组 API
 */
public interface CouponGroupAPI {

    /**
     * （分页查询）查询优惠券（码）组列表
     *
     * https://www.youzanyun.com/apilist/detail/group
     *
     * @param groupType 类型 PROMOCARD 优惠券, PROMOCODE 优惠券码
     * @param status 状态 FUTURE 未开始, END 已结束, ON 进行中
     * @param pageNo 第几页
     * @param pageSize 每页数量
     * @return 优惠券（码）组列表 {@link cn.iocoder.doc}
     */
}

```


4.2 CouponCardAPI

[CouponCardAPI](#) ，优惠券 API 。

```
/**
 * 优惠券 API
 */
public interface CouponCardAPI {

    /**
     * 创建优惠券。
     *
     * 生成 {@link CouponGroup} 和 {@link Coupon}
     *
     * https://www.youzanyun.com/apilist/detail/group
     *
     * @param params 优惠券参数
     * @return 优惠券信息 {@link cn.iocoder.doraemon}
     */
    YouzanUmpPromocardAddResult.UmpPromotionCard add

    /**
     * 查询买家优惠券列表
     *
     * https://www.youzanyun.com/apilist/detail/group
     *
     * @param fansType 粉丝类型 (自有粉丝: fans_type =
     * @param fansId 【三选一】粉丝ID, 与 fans_type 同时
     * @param openUserId 【三选一】三方用户ID mobile/f
     * @param mobile 【三选一】手机号, 如果没有则写 0 mo
     * @param status 状态。VALID 有效 (未使用) USED 已
     * @return 优惠券列表 {@link cn.iocoder.doraemon}
     */
    List<Coupon> searchByBuyer(Integer fansType, Int

    /**
```

4.3 CouponCodeAPI

[CouponCodeAPI](#)，优惠券 API。和 [CouponCardAPI](#) 类似。

```
/**
 * 优惠券 API
 */
public interface CouponCodeAPI {

    /**
     * 创建优惠券。
     *
     * 生成 {@link CouponGroup} 和 {@link Coupon}
     *
     * https://www.youzanyun.com/apilist/detail/group
     *
     * @param params 优惠券参数
     * @return 优惠券信息 {@link cn.iocoder.doraemon.un
     */
    YouzanUmpPromocodeAddResult.UmpPromotionCode add(

    /**
     * 查询买家优惠券列表
     *
     * https://www.youzanyun.com/apilist/detail/group
     *
     * @param fansType 粉丝类型（自有粉丝：fans_type = 1
     * @param fansId 【三选一】粉丝ID，与 fans_type 同时
     * @param openUserId 【三选一】三方用户ID mobile/fan
     * @param mobile 【三选一】手机号，如果没有则写 0 mob
     * @param status 状态。VALID 有效（未使用） USED 已使
     * @return 优惠券列表 {@link cn.iocoder.doraemon.un
     */
    List<Coupon> searchByBuyer(Integer fansType, Inte

    /**
     * 获取优惠券详情
```


4.4 CouponConsumeAPI

[CouponConsumeAPI](#) ， 优惠券（码）消费 API 。

```
/**
 * 优惠券（码）使用 API
 */
public interface CouponConsumeAPI {

    /**
     * 发放优惠券优惠码
     *
     * https://www.youzanyun.com/apilist/detail/group
     *
     * @param fansType 粉丝类型（自有粉丝：fans_type = 1）
     * @param fansId 【三选一】粉丝ID，与 fans_type 同时
     * @param openUserId 【三选一】三方用户ID mobile/fans_id
     * @param mobile 【三选一】手机号，如果没有则写 0 mobile
     * @param id 优惠券（码）组编号
     * @return 发放优惠券结果 {@link cn.iocoder.doraemon}
     */
    YouzanUmpCouponTakeResult take(Integer fansType,

    /**
     * 对优惠券/优惠码进行核销操作
     *
     * https://www.youzanyun.com/apilist/detail/group
     *
     * @param code 买家端的优惠券/优惠码核销码
     * @return 是否成功
     */
    Boolean verify(String code);

    /**
     * 根据核销码获取优惠券/优惠码
     *
     * https://www.youzanyun.com/apilist/detail/group
```

666. 彩蛋

随着笔者的啰嗦，本文迈入结（彩）束（蛋）部分。

有经验的胖友，可能会问，优惠券对交易订单的价格计算有什么影响。嘿嘿，这块内容，我们写完营销模块后，写一篇文章统一分享。

在分享一个京东的 [促销API](#)。

文章目录

1. [1. 1. 概述](#)
2. [2. 2. 背景了解](#)
 1. [2. 1. 2. 1 优惠券管理](#)
 2. [2. 2. 2. 2 优惠码管理](#)
 3. [2. 3. 2. 3 下单使用优惠](#)
 4. [2. 4. 2. 4 验证使用优惠](#)
3. [3. 3. 数据库实体](#)
 1. [3. 1. 3. 1 CouponGroup](#)
 1. [3. 1. 1. 3. 1. 1 基本信息](#)
 2. [3. 1. 2. 3. 1. 2 码信息](#)
 3. [3. 1. 3. 3. 1. 3 领取规则](#)
 4. [3. 1. 4. 3. 1. 4 使用规则](#)
 5. [3. 1. 5. 3. 1. 5 使用效果](#)
 6. [3. 1. 6. 3. 1. 6 统计信息](#)
 7. [3. 1. 7. 3. 1. 7 微信卡卷](#)
 8. [3. 1. 8. 3. 1. 8 创建后可编辑属性](#)
 2. [3. 2. 3. 2 Coupon](#)
 1. [3. 2. 1. 3. 2. 1 基本信息](#)
 2. [3. 2. 2. 3. 2. 2 码信息](#)
 3. [3. 2. 3. 3. 2. 3 领取情况](#)
 4. [3. 2. 4. 3. 3. 4 使用规则](#)
 5. [3. 2. 5. 3. 3. 5 使用效果](#)
 6. [3. 2. 6. 3. 3. 6 使用情况](#)
 3. [3. 3. 3. 3 CouponFetchLog](#)
 4. [3. 4. 3. 4 CouponVerifyLog](#)
4. [4. 4. API](#)
 1. [4. 1. 4. 1 CouponGroupAPI](#)
 2. [4. 2. 4. 2 CouponCardAPI](#)
 3. [4. 3. 4. 3 CouponCodeAPI](#)
 4. [4. 4. 4. 4 CouponConsumeAPI](#)
5. [5. 666. 彩蛋](#)