购物车订单模型

## 前言

该模型为魔力网购物车和订单初版设计模型，如有任何疑问，请及时提出。

## 更新日志

|  |  |  |
| --- | --- | --- |
| 2014-08-13 | 王宝庆 | 订单模型商品集,删除isdelivered、deliverednum，删除多个物流单号的情形（每单必须全部发货或全部不发货） |
| 2014-8-21 | 王宝庆 | 订单商品集，为每种商品添加索引号packindex，package字段改为packages |
| 2014-8-27 | 王宝庆 | 订单集合商品集添加type字段，商品名称  订单集合商品集添加goodsname、goodspic字段 |
| 2014-10-15 | 王宝庆 | 订单模型-balanceorderno 生成改为程序生成唯一单号 |
|  |  |  |

## 购物车模型

### 基本流程

1. 会员选择某个商品（按照选择的规格参数显示不同的价格），填写数量加入购物车。
2. 购物车存储规则：
3. 会员未登录，购物车商品存储到Cookie中
4. 会员已登录，购物车商品存储到MongoDB中
5. 会员可选择多件商品加入购物车
6. 会员可以对购物车内的商品进行操作，重新编辑数量（直接填写数字或+/-操作），删除购物车内的商品
7. 会员可选择购物车内的部分商品结算
8. 会员点击结算，提交之后，生成订单。

### Cookie模型

对于未登录魔力网的用户，其加入购物车的商品需要存储到cookie中，并且设置合理的cookie过期时间，方便用户购买商品。

#### 数据结构

cookie需要保存以下数据

|  |  |  |
| --- | --- | --- |
| **列** | **描述** | **备注** |
| **goodsid** | 商品ID | 对应goods集合中code字段 |
| **num** | 购买数量 int | 购物车中每个商品的购物数量，1-N的整型数字 |
| **spec** | 商品规格 | 对应goods集合中sku字段-spec子集，存储spec每个选项值对应的key |
| **addtime** | 添加日期 |  |
| **signature** | 校验签名 string | md5校验签名，可作为判断是否是有效cookie参数和值的验证码 |

Cookie名称：moolycart

Cookie值：json字符串

[

{

“goodsid”:” 62a4c97daf694112adec10176f330839”,

“num”:1,

“spec”:[0,2,1],

“addtime”: 1406280622,

“signature”:md5字符串

},

{

“goodsid”:” 53a4c97daf694112adec10186f3354ft”,

“num”:2,

“spec”:[2,1,1],

“addtime”: 1406280622,

“signature”:md5字符串

}

]

#### spec数据

以下商品属性为例：

spec:{

“颜色” : [“土豪金” , “天空蓝” , “火热红”],

“运营商” : [“中国移动” , “中国联通” , “中国电信”],

“机身存储”: [“16G”, “32G”]

……

}

price: {

"vip-rmb":[15.00,150.00,3.00,4.00,5.00,6.00,7.00,8.00,9.00,10.00,11.00.12.00,13.00,14.00,15.00,16.00,17.00,18.00]

}

该商品有n个spec属性，按顺序分别为spec[0], spec[1], spec[2]…spec[n]

每个属性分别有m个选项，按顺序分别为spec[n][0], spec[n][1]…spec[n][m]

Cookie存储数据spec值，存储规则：

会员所选择的每个spec（spec[n]）属性的选项对应的m值

在上面的例子中：

cookie数据的第一个商品的spec[0,2,1]

对应的spec属性为：”土豪金”,”中国电信”,”32G”

cookie数据的第二个商品的spec[2,1,1]

对应的spec属性为：”火热红”,”中国联通”,”32G”

#### ~~price定位~~

**注：price推导按照商品模型推导公式计算**

~~以cookie数据的第一个商品为例：~~

~~公式：loc = (count(spec[n])\* count(spec[n-1])\* count(spec[n-2])…\* count(spec[1]))\*s(0)~~

~~+(count(spec[n])\* count(spec[n-1])\* count(spec[n-2])…\* count(spec[2]))\*s(1)~~

~~+(count(spec[n])\* count(spec[n-1])\* count(spec[n-2])…\* count(spec[3]))\*s(2)~~

~~+…~~

~~+(count(spec[n])\* count(spec[n-1]))\*s(n-2)~~

~~+(count(spec[n]))\*s(n-1)~~

~~+s(n)~~

~~price = price[“vip-rmb”][loc]~~

~~推导出：~~

~~Cookie数据中第一个商品的price定位为6\*0+2\*2+1=5，价格=price[“vip-rmb”][5]=6.00~~

~~Cookie数据中第二个商品的price定位为6\*2+2\*1+1=15，价格=price[“vip-rmb”][15]=16.00~~

#### 注意事项

1. Cookie存取有域名限制，cookie存取需要设置domain为一级域名，方便跨域名读取

### MongoDB模型

对于已登录魔力网的会员，其加入购物车的商品，需要存储到MongoDB中，方便其在不同的地方或者设备上购买。

#### 数据结构

MongoDB需要存储以下数据

|  |  |  |
| --- | --- | --- |
| **列** | **描述** | **备注** |
| **userid** | 会员ID | 对应oracle数据库中user表的ID字段 |
| **goodsid** | 商品ID | 对应goods集合中code字段 |
| **num** | 购买数量 int | 购物车中每个商品的购物数量，1-N的整型数字 |
| **spec** | 商品规格 | 对应goods集合中sku字段-spec子集，存储spec每个选项值对应的key |
| **addtime** | 添加日期 | ISODATE |
| **expiretime** | 过期日期 | 购物车中该条文档过期时间，暂定购物车中文档过期时间为90天 |
| **signature** | 校验签名 string | md5校验签名，可作为判断是否是有效cookie参数和值的验证码 |

MongoDB集合名称：moolycart

存储文档结构：

{

"userid":"skjdflsd1fsdf2slkdfdslfjl1233",

"packages":

[

{

"goodsid":"62a4c97daf694112adec10176f330839",

"num":1,

"spec":[0,2,1],

"addtime":”2014-08-14 21:03:15”,

"expiretime": ”2014-08-14 21:03:15”,

"signature":md5字符串

},

{

"goodsid":"53a4c97daf694112adec10186f3354ft",

"num":2,

"spec":[2,1,1],

"addtime": ”2014-08-14 21:03:15”,

"expiretime":”2014-08-14 21:03:15”,

"signature":md5字符串

}

]

}

…

#### spec数据

同cookie模型

#### price定位

同cookie模型

#### 注意事项

1. moolycart集合中的文档设置了一个过期时间expiretime，目的是为防止垃圾数据太多，集合过于庞大影响读取效率，系统读取购物车数据只需要读取未过期的文档即可，对于已过期的数据，系统需要定期清理

## 订单模型

订单采用分商家生成模式，如果会员同时购买n家不同商家的商品，则系统需要生成n张订单（分别为每家商家生成一张订单），每张订单只包含会员所购买的该商家的商品。

如果多张订单统一结算，以第一张订单的订单号作为结算单号。

### 基本流程

1. 会员选择购物车中部分商品或全部商品进行结算。
2. 点击结算之后，需要会员确认收货地址（可选择已有或者添加新地址）
3. 会员填写备注说明
4. 点击提交订单，生成订单
5. 订单生成之后，用户不可修改任何订单信息

### 基本模型

#### 订单状态模型

##### 订单状态基本模型

模型集合名称：orderstatus

模型编码：OS

{

code: “OS001”,

caption: “待付款”,

deadtime: 259200,

deadstatus: ”OS121”,

opset: {

1: [“OO101”,”OO102”],

2: [“OO104”]

…

},

status:1

}

|  |  |  |
| --- | --- | --- |
| **列** | **描述** | **备注** |
| **code** | 状态编码string | OS001 待付款  OS002 已付款、待发货  OS003 已发货  OS004 已收货  OS005 已取消  ~~OS011 退换货中~~  OS901 交易成功  OS902 交易失败  OS903 交易关闭  OS904 订单已删除 |
| **caption** | 状态编码说明 | 待付款、已付款等 |
| **deadtime** | 状态自动过期（失效）时间 | 例：OS001(待付款)状态，如果未有任何操作，3天之后自动转为deadstatus状态值OS903(交易失败) |
| **deadstatus** | 状态自动过期（失效）后的状态编号 |  |
| **opset** | 订单交易操作数组 | 表示该状态下可以有的操作  {  relationid:[opid,opid…],  relationid:[opid,opid…],  …  } |
| **status** | 订单状态的状态 | 0已禁用 1已启用 |

##### 订单相关方

模型集合名称：orderrelation

模型编码：OR

{

relationid: 1

relationname: 订单拥有者（魔力会员）

}

|  |  |  |
| --- | --- | --- |
| **列** | **描述** | **备注** |
| **relationid** | 订单相关方id | 1,2,3 |
| **relationname** | 订单相关方 | 1：订单拥有者（魔力会员）  2：商品拥有者（魔力商家）  3：魔力总部 |

##### 订单交易操作

模型集合名称: orderop

模型编码：OO

{

opid: OO101,

opname: 支付,

opstatus: OS002,

status:1

}

|  |  |  |
| --- | --- | --- |
| **列** | **描述** | **备注** |
| **opid** | 操作id | OO101~OO999 |
| **opname** | 操作描述 | OO101：支付  OO102：确认收货  OO103：退货/退款  OO104：关闭订单 |
| **opstatus** | 执行操作之后订单的状态 | map[orderstatus]id |
| **status** | 操作本身的状态 | 0已禁用 1启用 |

##### 订单状态初始化数据

参考文档：模板文档/订单状态和操作.docx

##### 订单操作初始化数据

参考文档：模板文档/订单操作和操作.docx

#### 订单模型

##### 基本模型

模型集合名称：order

模型编码：O

{

orderno: “14072716581255321257”,

uid: “IUUER-SD545-SFSKDJ-SDF45”,

bid: “SKDJF334-SDFJSKDJ-3434SDF”,

packages : [

{

packindex:1, //商品包里商品的索引ID 按商品数递增1-n

goodsid: “SDKFJ-23230-KNDSFKJJK”,

goodsname:”商品名称”,

goodspic:”商品图片”,

count: 2,

spec: “颜色:白色,存储:32G”,

version : 12,

price: 123.00,

clear: 122.00,

iscommented:0,

type:0,

typelog:”23423sdfsdf2sa1a21af”, //对应退货表、换货表、退款表的\_id

~~isdelivered:0, //是否已发货 0未发货 1已发货~~

~~deliverednum:1 //已发货商品数量~~

},

{

packindex:2, //商品包里商品的索引ID 按商品数递增1-n

goodsid: “SDKFJ-23230-KNDSFKJJK”,

goodsname:”商品名称”,

goodspic:”商品图片”,

count: 2,

spec: “颜色:黑色,存储:16G”,

version : 12,

price: 123.00,

clear: 122.00,

iscommented:0,

type:0,

typelog:”23423sdfsdf2sa1a21af”, //对应退货表、换货表、退款表的\_id

~~isdelivered:0, //是否已发货 0未发货 1已发货~~

~~deliverednum:1 //已发货商品数量~~

},

…

],

price: 492.00,

clear: 488.00,

returnprice:0.00,

postage : 24.00,

savedamount : 14.00,

logistics:[

{

loginame: 物流公司名称,

logisticsno: xxxxxxxxxxxx

},

~~{~~

~~loginame: 1,~~

~~logisticsno: xxxxxxxxxxxx~~

~~},~~

…

],

createtime ： 2014-01-10 12:00:00,

modifytime ： 2014-02-20 12:00:00,

paytime : 2015-02-12 00:00:00,

status: OS001,

statustime: 2015-02-12 00:00:00, //状态写入时间

statuslog:[

{

statusid: OS001,

logtime: 2014-08-07 16:30:10

},

{

statusid: OS002,

logtime: 2014-08-07 16:50:10

},

…

]

iscommented:0,

receiver:’张三’, //收货人

phone:’13711111111’, //联系方式

deliveryaddress:’江苏省苏州市XX路’,

postalcode: ‘215000’,

remark:’备注说明，买家留言’,

balanceorderno: “14072716581255321257”,

~~type:~~**~~[0,1],~~**

~~Typelog:~~**~~[1,2]~~**

}

|  |  |  |
| --- | --- | --- |
| **列** | **描述** | **备注** |
| **orderno** | 订单号 |  |
| **uid** | 用户ID string |  |
| **bid** | 商家ID string |  |
| **packages** | 商品集 |  |
| **price** | 订单总价格 float |  |
| **clear** | 订单结算价 |  |
| **returnprice** | 退款金额 float | 如果该订单发生退款行为，存储该订单的退款总金额 |
| **postage** | 邮费 float |  |
| **savedamount** | 节省金额 float |  |
| **loginame** | 物流公司名称 | 对应logistics集合name字段 |
| **logisticsno** | 物流单号 | string |
| **createdate** | 订单产生时间 |  |
| **modifytime** | 订单状态最后一次更新时间 |  |
| **paytime** | 支付时间 |  |
| **status** | 订单当前状态 |  |
| **statustime** | 订单状态写入时间 |  |
| **statuslog** | 订单状态变更日志 | 记录该订单的状态变更日志（状态id+时间），方便查询 |
| **iscommented** | 是否已评价 | 0未评价 1已评价 |
| **receiver** | 收货人 |  |
| **phone** | 联系方式 | 手机号码 |
| **deliveryaddress** | 收货地址 |  |
| **postalcode** | 邮政编码 |  |
| **remark** | 备注说明 | 买家留言 |
| **balanceorderno** | 结算单号 | 程序生成唯一结算单号 |
| **~~type~~** | ~~订单类型~~ | ~~[1,2]同时存在物品退货与换货~~  ~~0：正常~~  ~~1：退货中~~  ~~2：换货中~~  ~~3：退款中~~ |
| **~~typelog~~** | ~~订单类型log~~ | ~~[2,3]（有过退货与换货）~~ |

##### packages商品集

[

{

packindex:1, //商品包里商品的索引ID 按包里商品数递增1-n

goodsid: “SDKFJ-23230-KNDSFKJJK”,

goodsname:”商品名称”,

goodspic:”商品图片”,

count: 2,

spec: “颜色:白色,存储:32G”,

version : 12,

price: 123.00,

clear: 122.00,

iscommented:0,

type:0,

typelog:”23423sdfsdf2sa1a21af”, //对应退货表、换货表、退款表的\_id

~~isdelivered:0, //是否已发货 0未发货 1已发货~~

~~deliverednum:1 //已发货商品数量~~

},

{

packindex:2, //商品包里商品的索引ID 按包里商品数递增1-n

goodsid: “SDKFJ-23230-KNDSFKJJK”,

goodsname:”商品名称”,

goodspic:”商品图片”,

count: 2,

spec: “颜色:黑色,存储:16G”,

version : 12,

price: 123.00,

clear: 122.00,

iscommented:0,

type:0,

typelog:”23423sdfsdf2sa1a21af”, //对应退货表、换货表、退款表的\_id

~~isdelivered:0, //是否已发货 0未发货 1已发货~~

~~deliverednum:1 //已发货商品数量~~

},

…

]

|  |  |  |
| --- | --- | --- |
| **列** | **描述** | **备注** |
| **packindex** | 商品索引号 | 商品包里商品的索引ID 按商品数递增，例订单中有3种不同的商品，packindex值分别为1,2,3 |
| **goodsid** | 商品ID |  |
| **goodsname** | 商品名称 |  |
| **goodspic** | 商品logo图片 |  |
| **count** | 购买数量 |  |
| **spec** | 购买的商品规格 | 直接存储规则内容 |
| **version** | 商品版本号 |  |
| **price** | 单件商品购买价格 | float |
| **clear** | 单件商品结算价格 | float |
| **iscommented** | 订单中该商品是否已评价 | 0未评价 1已评价 |
| **type** | 该商品退换货类型 | 0 正常  1 退货处理中  2 换货处理中  3 退款处理中(暂未使用)  4 退货完成  5 换货完成 |
| **typelog** | 根据type来定义 | 如果type=1，退货集合\_id  如果type=2，换货集合\_id |
| **~~isdelivered~~** | ~~该商品是否已发货~~ | ~~0未发货 1已发货~~ |
| **~~deliverednum~~** | ~~商品已发货数量~~ | ~~Int~~ |

### 相关说明

#### 订单号生成规则

OD+年月日时分秒+手机号后四位+四位随机数（OD14072716581255321257），订单号存储可直接存储数字部分，OD只作为显示时订单号前缀使用

#### 关于订单评价

会员确认收货之后，允许对商品进行评价

#### 关于订单商品分批发货 – 拆单

前期暂时不做商品分批发送，如果后期遇到一个订单中有大宗商品的情形，考虑采用订单拆单的模式，将未发货的部分商品拆分到新订单

#### 关于订单收发货与退货退款

~~用户申请退货/退款，与订单固有流程无关，订单状态仍旧按照固定流程走，系统在退货/退款那边处理用户的申请，来进行额外处理，最终结果符合用户和卖家双方预期。~~