**身边购设计文档**

# 参数说明

* 在整个身边购应用中所有接口出现过的同名参数拥有相同的意义，所以同一个名称的参数不会在不同接口重复解释，同名参数在不同接口中的验证逻辑也是一样的
* 所有接口返回值都有status属性，大于0表示成功，小于等于0表示失败。如果没有特殊说明，1：成功，0：失败
* 所有请求参数包含token的接口，token由1.2获得。如果token与服务器不一致或没有服务器没有记录相关用户id的token返回的status是**-5**，status:-5只能在这一处地方使用。
* 如果接口传的硬件信息没有通过验证status是**-6**，status:-6只能在这一处地方使用。
* 如果接口没有传userId，status是**-8**，status:-8只能在这一处地方
* 因为全部采用ICE等RPC接口，下面所有接口名都采用java方法全限定名表示
* token为用户手机号、用户ID、登录时间（格式：yyyyMMddHHmmssSSS），三项连接成的字符串计算SHA-1的16进制字符串。
* 所有URL都由web端使用，web服务器收到请求之后，调用对应的接口
* 界面一律使用HTML5实现，可以使用bootstrap/velocity。移动客户端采用针对浏览器加壳的方式，比如安卓可以围绕WebView实现。
* WEB采用HTTP通讯协议。移动客户端采用ICE RPC方式
* 如果没有特别说明，数据库表字段一律为非空字段。NO表示同有默认值，必须由程序或用户指定
* 所有配置参数都保存在ZK或hazelcast，通过一个配置管理中心操作
* 所有ZK路径都遵守/cn/purchase/near/业务模块/功能/值，这样的模式
* 所有业务相关的枚举值，都保存在同一张表里
* 所有用户请求，都要在日志里以INFO级别记录请求参数，处理结果，请求处理耗时，同时将这些内容作为用户行为数据发送给MQ，供STORM/SPARK等工具分析
* 针对用户行为分析的MQ消息，可以把URI路径的“/”换成“.”，作为消息主题，发布的消息为完整的请求处理耗时、请求参数、请求处理结果（根据业务需要可以跟返回客户端的结果有所不同，比如响应结果是列表类型的，发布到MQ的消息只包含列表元素数量和处理状态即可）。按JSON格式{“result”:{…},”params”:{…},”consumed”:…}。result是处理结果，params是接口参数，consumed是请求耗时。
* 采用长连接异步RPC

# 功能模块及功能说明

## 1.用户

### 1.1注册

**接口名： ActivateInfo cn.purchasenear.v1.user.UserService.regist(ClientInfo info)**

**调用时机：在客户端新安装或升级首次启动时**

**URL:** [**http://localhost/v1/user/regist**](http://localhost/v1/user/regist) **POST**

**是否需要登录：否  
额外说明：硬件相关的参数不持久化到数据库只用于数据分析**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| mac | string | 只有安卓用户需要传 | 12位16进制数，没有分隔符 |
| imei | string | 只有安卓用户需要传 |  |
| openUdid | string | 只有苹果用户需要传 |  |
| idfa | string | 只有苹果用户需要传 |  |
| os | int | 必传 | 操作系统编号 |
| osVersion | string | 必传 | 操作系统版本号 |
| manufacture | string | 只有安卓苹果用户需要传 | 移动端硬件制造商 |
| platformCode | long | 只有安卓苹果用户需要传 | 移动端应用市场编号 |
| signature | string | 只有安卓用户需要传 | 安卓APP签名的MD5BASE64 |
| packageName | string | 只有安卓用户需要传 | 安卓 APP包名 |
| phone | string |  | 手机号 |
| verifyCode | string |  | 图片验证码，如果能实现短信验证码更好 |
| password | string |  | 登录密码 |
| real\_name | string |  | 真实姓名 |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 请求处理状态 |

**数据库表：user\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| phone | varchar | 11 | 空串 | UNI |  |
| real\_name | varchar | 16 | 空串 | NO | 真实姓名 |
| password | varchar | 24 | 空串 | NO | MD5密码 |
| credit | bigint |  | 1000 |  | 用户信誉评价，从初始值1000开始。未能及时完成订单将会减分，需要有优秀的算法支持，暂不实现 |
| regist\_time | datetime |  | NO | NO | 注册时间 |

### 1.2登录

**接口名： Login cn.purchasenear.v1.user.UserService.login(String phone,String password)**

**调用时机：登录界面**

**URL: http://localhost/v1/user/login**

**是否需要登录：否**

**额外说明：**

1. **用户登录状态保持时间可由ZK配置, 登录数据结构保存在REDIS，登录状态的寿命从最后登录时间或最后访问时间开始算，每次访问都重置登录状态寿命。**
2. **硬件相关的参数不持久化到数据库只用于数据分析。**
3. **登录数据在REDIS中保存在以下路径userSession:#{USER\_ID}，**
4. **登录数据保存为JSON。包含user表的全部字段，最后登录时间，token，将来业务扩展了可能还会增加，比如卖家信息**
5. **登录数据可以保存在redis/memcached/zookeeper/hazelcast，定义好interface，支持不同的登录数据存储方案。**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| phone | string | true | 登录的手机号 |
| password | string | true | 登录密码 |
| mac | string | 只有安卓用户需要传 | 12位16进制数，没有分隔符 |
| imei | string | 只有安卓用户需要传 |  |
| openUdid | string | 只有苹果用户需要传 |  |
| idfa | string | 只有苹果用户需要传 |  |
| os | int | 必传 | 操作系统编号 |
| osVersion | string | 必传 | 操作系统版本号 |
| manufacture | string | 只有安卓苹果用户需要传 | 移动端硬件制造商 |
| platformCode | long | 只有安卓苹果用户需要传 | 移动端应用市场编号 |
| signature | string | 只有安卓用户需要传 | 安卓APP签名的MD5BASE64 |
| packageName | string | 只有安卓用户需要传 | 安卓 APP包名 |
|  |  |  |  |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 请求处理状态 |
| userId | long | 是 | 登录成功的用户ID user\_info.id |
| token | string | 是 | 登录时间+phone计算md5base64，如果能计算aesbase64更好 |

### 1.3注销

**接口名： CommonReturn cn.purchasenear.v1.user.UserService.logout(long userId,String token)**

**调用时机：注销界面**

**URL:** [**http://localhost/v1/user/logout**](http://localhost/v1/user/logout) **DELETE**

**是否需要登录：是**

**额外说明：将登录数据删除**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 请求处理状态，1：成功，0：失败 |

### 1.4验证登录状态

**接口名： CommonReturn cn.purchasenear.v1.user.UserService.verify(long userId,String token)**

**调用时机：注销界面**

**是否需要登录：不是必须**

**额外说明：验证用户登录状态，只作为服务器内部调用**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 具体请见文档开头的参数说明，有关status部分 |

## 2.购买

### 2.1添加购物车

**接口名： CommonReturn cn.purchasenear.v1.buy.CartService.addCart(long userId,String token,long goodsId,int amount)**

**调用时机：添加到购物车**

**URL: http://localhost/v1/cart/add POST**

**是否需要登录：是**

**额外说明：**

购物车一律分别在本地和服务器各存一份。

web用户使用LocalStorage存储，安卓苹果可选择适合自身的方式

只保存userId 商品ID，每种商品购买数量。

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |
| goodsId | long | true | 商品ID |
| amount | int | true | 商品的购买数量 |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 1：成功；0：失败 |

**数据库表：purchase\_cart**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| user\_id | bigint |  | 0 | MUL | 购物车的所有人，一个用户的购物车记录可以有多行，一行表示购物车里的一种商品 |
| goods\_id | bigint |  | 0 | NO | 商品ID |
| amount | int |  | 0 | NO | 购买数量 |
| add\_time | datetime |  | NO | NO | 添加到购物车的时间 |

### 2.2查询购物车

**接口名： PurchaseCartList cn.purchasenear.v1.buy.CartService.queryCart(long userId,String token)**

**调用时机：查询购物车**

**URL:** [**http://localhost/v1/cart/query**](http://localhost/v1/cart/query) **POST**

**是否需要登录：是**

**额外说明：**

查询当前购物车数据，返回purchase\_cart列表

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |
| goodsId | long | true | 商品ID |
| amount | int | true | 商品的购买数量 |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 1：成功；0：失败 |
| purchaseCartList | List<PurchaseCart> | 是 | 对应userId的purchase\_cart记录列表。PurchaseCart是数据库表purchase\_cart的实体类 |

### 2.3删除购物车内的商品

**接口名： CommonReturn cn.purchasenear.v1.buy.CartService.removeCart(long userId,String token)**

**调用时机：删除购物车内的商品**

**URL: http://localhost/v1/cart/remove POST**

**是否需要登录：是**

**额外说明：**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |
| cartId | long | true | purchase\_cart.id |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 1：成功；0：失败 |

### 2.4清空购物车

**接口名： CommonReturn cn.purchasenear.v1.buy.CartService.clearCart(long userId,String token)**

**调用时机：删除购物车内的商品**

**URL: http://localhost/v1/cart/clear POST**

**是否需要登录：是**

**额外说明：**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |
| cartId | long | true | purchase\_cart.id |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 1：成功；0：失败 |

### 2.5修改购物车

**接口名： CommonReturn cn.purchasenear.v1.buy.CartService.changeCart(long userId,String token,long cartId,int amount)**

**调用时机：删除购物车内的商品**

**URL: http://localhost/v1/cart/clear POST**

**是否需要登录：是**

**额外说明：**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |
| cartId | long | true | purchase\_cart.id，要修改的购物车记录 |
| amount | int | true | 要修改的购物车记录相应商品数量 |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 1：成功；0：失败 |

### 2.6下单

**接口名： CommonReturn cn.purchasenear.v1.buy.OrderService.order(long userId,String token)**

**调用时机：点击下单按钮**

**URL:** [**http://localhost/v1/buy/order**](http://localhost/v1/buy/order) **POST**

**是否需要登录：是**

**额外说明： 查询购物车，保存订单数据，减库存。（需要调用查询购物车和减库存接口）**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 请求处理状态，1：成功，0：失败 |
| orderId | long | 是 | order.id |

**数据库表：purchase\_order**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| user\_id | bigint |  | 0 | MUL | 购物车的所有人，一个用户的购物车记录可以有多行，一行表示购物车里的一种商品 |
| address | varchar | 1024 | NO | NO | 收件人地址 |
| receiver\_name | varchar | 32 | NO | NO | 收件人姓名 |
| receiver\_phone | varchar | 11 | NO | NO | 收件人电话 |
| receiver\_email | varchar | 128 | NO | NO | 收件人邮箱 |
| order\_code | varchar | 16 | NO | UNI | 订单编号。开头两位为当前年份的36进制，比如今年是2015年，开头就是0F；接下来是今年已经过去的秒数的36进制；然后就是用户ID；最后是当天的第几个订单 |
| order\_status | int |  | 0 | NO | -2：支付失败；-1：退单；0：下单未支付；1：支付中；2：支付成功 |
| buy\_time | datetime |  | NO | NO | 添加到购物车的时间 |

**数据库表：purchase\_order\_goods**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| user\_id | bigint |  | 0 | MUL | 购物车的所有人，一个用户的购物车记录可以有多行，一行表示购物车里的一种商品 |
| order\_id | bigint |  | NO | MUL | order.id |
| goods\_id | bigint |  | NO | NO | goods.id |
| amount | int |  | NO | NO | 购买数量 |
| unit\_price | decimal | 8,2 | NO | NO | 单价 |

**数据库表：purchase\_receiver——收件人信息，一个可以有多个收件人信息**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| user\_id | bigint |  | 0 | MUL | 与收件人相关的用户 |
| receiver\_name | varchar | 32 | NO | NO | 收件人姓名 |
| receiver\_phone | varchar | 11 | NO | NO | 收件人电话，可以给收件人发短信，报告物流进度 |
| receiver\_email | varchar | 128 | 空串 | NO | 收件人邮箱，可以给收件人发邮件，报告物流进度 |
| wechat | varchar | 64 | 空串 | NO | 收件人微信，可以给收件人发微信，报告物流进度 |
| province | varchar | 32 | NO | NO | 收件人所在省 |
| city | varchar | 32 | NO | NO | 收件人所在市 |
| district | varchar | 32 | NO | NO | 收件人所在区县 |
| street | varchar | 64 | NO | NO | 收件人所在街道 |
| street\_number | varchar | 64 | NO | NO | 收件人所在街道号 |
| extra\_info | varchar | 1024 | NO | NO | 收件人附加信息，比如公司之类 |

### 2.7限时抢购

**接口名： CommonReturn cn.purchasenear.v1.buy.OrderService.flashsales(long userId,String token,long goodsId)**

**调用时机：点击抢购按钮**

**URL:** [**http://localhost/v1/buy/flashsales**](http://localhost/v1/buy/flashsales) **POST**

**是否需要登录：是**

**额外说明：**

**限时抢购，数据库表与2.2共用。需要有定时任务检查抢到但等待支付超时的flash\_sales记录；支付超时，尚未结束的flash\_sales将会发一条MQ，订阅者调用2.5负责重置flash\_sales，并执行退单。**

**执行本接口，首先修改flash\_sales，确认用户拿到入场券，对于拿到入场券的用户，立即调用添加购物车，调用下单，然后直接进入支付流程。**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |

**返回值：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 请求处理状态，1：成功，0：失败 |

**数据库表：flash\_sales——活动结束清除相关记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| user\_id | bigint |  | 0 | MUL | 获得抢购入场券的用户，调用该接口时修改这个字段成功的用户才可成功下单，其他用户返回抢购失败 |
| order\_id | bigint |  | 0 | MUL | order\_id,user\_id成复合索引，不必为user\_id和order\_id单独建索引 |
| goods\_id | bigint |  | NO |  | 商品ID |
| pass | varchar | 32 | NO | UNI | 抢购入场券，一种商品有多少参加抢购就有多少入场券，入场券可在活动开始前预先lpush到redis的list中，为每个参与的用户执行rpop，没有rpop到的用户为没有获得入场券。相关redis key不设置超时，抢完为止，活动结束统一清除限时抢购相关的redis缓存 |
| start\_time | datetime |  | NO | MUL | 抢购开始时间 |
| end\_time | datetime |  | NO | MUL | 抢购结束时间 |
| take\_time | datetime |  | null |  | 用户获得入场券的时间 |

### 2.8支付

**2.7.1 交易流程说明**

**根据网站架构使用网络支付方式。**

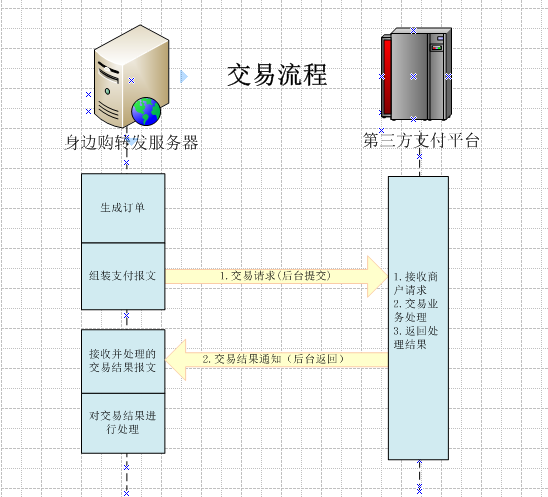
1. 商户生成订单信息，构造交易对象，调用第三方支付公司提供的交易API方法组装报文（默认为银联报文）；
2. 将交易报文信息通过HTTP POST方式以XML流方式发往第三方支付平台；并同步接收第三方支付平台的应答报文；
3. 将接收返回交易报文；
4. 对交易结果进行相应处理；

**因银联报文管理要求，建议整个交付环境以专线+密钥机加密形式进行传输报文**

**2.7.2 交易接口文件**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | | 参数类型 | 是否必须 | 说明 |
| trancode | | varchar | true | 交易代码,一般为第三方支付平台提供的交易验证代码，这里因为只是测试，使用7001 |
| order\_code | | varchar | true | 订单编号。开头两位为当前年份的36进制，同order表中的相同字段 |
| 持卡人消费信息 | cardtype | double | true | 发卡行卡信息，需要与第三方支付平台进行核对 |
| cradno | double | true | 卡号 |
| * Password | double | true | 用户密码，需要利用3DES进行加密 |
| * Amt | double | true | 交易金额 |
| * prodInfo | int | true | 商品信息, |
| * phonenumber | string | false | 持卡人手机号，需要与第三方支付平台进行核对 |
| custname | string | false | 持卡人姓名，需要与第三方支付平台进行核对 |
| misc | long | false | 自定义保留域 |
| trandatetime | | double | true | 交易时间，下单交易时间 |
| * bitmisc | | int | false | 自定义保留域 |

**2.7.3交易流程图**



**2.4.4数据表**

### 2.9限时抢购超时重置

**接口名： void cn.purchasenear.v1.buy.OrderService.flashsales(long orderId,long flashSalesId)**

**调用时机：限时抢购支付超时**

**是否需要登录：否**

**额外说明：仅供2.3的MQ订阅者调用，不必返回结果。重置是指将flash\_sales相应的order\_id和user\_id置为0**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| orderId | long | true | 将被重置的订单ID |
| flashSalesId | long | true | 将被重置的flash\_sales记录的ID |

### 2.10查询订单列表

### 2.11查询订单详情

## 3.商品

### 3.1搜索商品

**接口名： SearchGoods cn.purchasenear.v1.buy.GoodsService.flashsales(SearchParam param)**

**调用时机：点击抢购按钮**

**URL:** [**http://localhost/v1/buy/flashsales**](http://localhost/v1/buy/flashsales) **POST**

**是否需要登录：是**

**额外说明：**

1. **默认以商品与用户距离排序，还可以选择浏览量和销量排序，应可在ZK中有相关配置项。**
2. **使用elasticsearch检索**
3. **搜索结果有地理位置特性，缓存命中率太低，不保存redis缓存**
4. **所有GoodsInfo的属性都要保存到elasticsearch，所有SearchInfo的红色字段都要建索引**
5. **浏览量与销量以天为单位建30个索引字段，随时更新。浏览量的索引字段是browsedDay1、browsedDay2、browsedDay3……browsedDay30，销量的索引字段是saledDay1、saledDay2、saledDay3…….saledDay30。**
6. **所有索引字段和储存字段以商品为单位都保存在一个elasticsearch文档里。**

**参数：SearchInfo**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |
| longitude | double | true | 用户经度，必须与纬度一起出现 |
| latitude | double | true | 用户纬度，必须与经度一起出现 |
| * distance | double | true | 查询距离用户distance米半径的商品 |
| * page | int | true | 查询当前页数 |
| * pageSize | int | true | 查询每页商品数 |
| goodsName | string | false | 商品名，**模糊查询** |
| sellerName | string | false | 卖家名，**模糊查询** |
| goodsCategory | long | false | 商品类别id, goods\_category.id，事先由接口3.5得到  **卖家名、商品名、商品类别三个条件必须至少有一个** |
| brand | string | false | 按品牌查询 |
| lowPrice | BigDecimal | false | 查询的商品价格不低于此值 |
| highPrice | BigDecimal | false | 查询的商品价格不高于此值 |
| * hypostatic | int | false | 是否有实体店，-1：只查询没有实体店的，0：忽略本参数，1：只查询有实体店的。 |
| * deliver | int | false | 是否送货上门，-1：只检索不送货上门的，0：忽略此参数，1：只检索送货上门的 |
| * second\_hand | int | false | 是否二手，-1：只检索非二手货，0：忽略此参数，1：只检索二手货 |
| * barter | int | false | 是否以物易物，-1：只检索不可以物易物的，0：忽略此参数，1：只检索可以物易物的 |
| * price | boolean | false | 是否以商品价格排序，默认false |
| * saled | boolean | false | 是否以销量排序，默认false |
| * browsed | boolean | false | 是否以浏览量排序，默认false |
| * seller\_credit | boolean | false | 是否以卖家信誉排序，默认false，暂不实现。user\_info.credit |

**返回值：SearchGoods，包含属性status和goodsList**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 请求处理状态，1：成功，0：失败 |
| page | int | 是 |  |
| goodsList | List<GoodsInfo> | 是 | 查询结果，下面的各属性是GoodsInfo的属性 |
| goodsId | long | 是 | goods.id |
| goodsName | string | 是 | goods.goods\_name |
| category | long | 是 | goods\_category.id |
| stock | int | 是 | 剩余库存，goods.stock |
| logintitude | double | 是 | 卖家经度 |
| latitude | double | 是 | 卖家纬度 |
| address | string | 是 | 卖家位置 |
| * hypostatic | int | false | 是否有实体店，-1：没有实体店， 1：有实体店。 |
| * deliver | int | false | 是否送货上门，-1：不送货上门， 1：送货上门 |
| * second\_hand | int | false | 是否二手，-1：非二手货，1：二手货 |
| * barter | int | false | 是否以物易物，-1：不可以物易物，1：可以物易物 |
| distance | double | true | 与用户距离 |

**数据库表：goods——商品**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| goods\_name | bigint |  | 0 | NO | 商品名 |
| goods\_category | bigint |  | NO | NO | goods\_category.id |
| stock | int |  | NO | NO | 剩余库存 |
| price | decimal | 8,2 | NO | NO | 商品价格，同种商品但是因某些属性不同导致价格不同的也是不同的商品记录 |
| brand | bigint |  | NO | NO | brand.id |
| * second\_hand | int |  | NO | NO | 是否二手，-1：非二手货，1：二手货 |
| * barter | int |  | NO | NO | 是否以物易物，-1：不可以物易物，1：可以物易物 |
| * save\_time | datetime |  | NO | MUL | 保存时间 |
| * update\_time | datetime |  | NO | MUL | 最后修改时间 |

**数据库表：goods\_detail——与goods一起构成商品详情**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| goods\_id | bigint |  | NO | pk |  |
| title\_img\_url | varchar | 1024 | 空串 | NO | 空串就用默认图片代替，否则就给客户端返回这个url |
| desc\_img\_urls | text |  | 空串 | NO | 空串就是没有图片，否则以空格分隔图片url，并给客户端返回url |
| description | text |  | 空串 |  | 可以是html格式的可包含<img/>，<style/>，以及可包含文本的任意标签，不能有<script><meta><head><body><html>等标签 |

**数据库表：brand——品牌**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| brand\_name | varchar | 32 |  |  | 品牌名 |

**数据库表：goods\_category——商品类别，有能力可以参考这个实现**<http://www.docin.com/p-7859975.html?qq-pf-to=pcqq.group>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| parent\_id | bigint |  | NO | MUL | 自关联，goods\_category |
| name | varchar | 32 | NO | NO | 商品类别名称 |
| description | varchar | 1024 | NO | NO | 类别描述 |

**数据库表：goods\_attributes——商品属性，商品规格也算属性**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| goods\_id | bigint |  | NO | pk | goods.id |
| enum\_id | bigint |  |  | PK | enums.id，在enums定义商品属性名与属性值 |

**数据库表：seller**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| seller\_name | varchar | 32 | NO | UNI | 卖家名 |
| logintitude | double |  |  |  | 卖家经度 |
| latitude | double |  |  |  | 卖家纬度 |
| geo\_hash | varchar | 13 |  | MUL |  |
| province | long |  |  |  | geo.id |
| city | long |  |  |  | geo.id |
| district | long |  |  |  | geo.id |
| street | long |  |  |  | geo.id |
| street\_number | long |  |  |  | geo.id |
| * hypostatic | int |  | NO | NO | 是否有实体店，-1：没有实体店， 1：有实体店。 |
| * deliver | int |  | NO | NO | 是否送货上门，-1：不送货上门， 1：送货上门 |

**数据库表：geo——保存省、市、区、县、街道、街道号，全局类型的表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| geo\_name | varchar | 32 |  | MUL | 省、市、区、县、街道、街道号的名字 |
| parent | bigint |  | 0 | MUL | 与主键自关联，值是0的表示顶级行政单位 |

**数据库表：enums——保存所有枚举类型的值，全局类型的表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 默认值 | 索引 | 说明 |
| id | bigint |  | NO | pk |  |
| module\_name | varchar | 64 | NO | UNI | 模块名，module\_name,enum\_name,enum\_value一起构成惟一性复合索引 |
| enum\_name | varchar | 64 | NO | UNI | 枚举名 |
| enum\_value | varchar | 64 | NO | UNI | 枚举值 |
| description | varchar | 128 | NO |  | 枚举值描述 |

* 索引中增加地理位置、省份等信息，方便周边查找
* 索引中增加诸如实体店、送货上门、二手商品、支持以物换物等标示
* 索引中增加热卖、人气指数等标示
* 用户支付成功后要发布MQ消息更新商品的销量索引
* 当用户点击某个商品以后，更新人气指数索引
* 滑动窗口的范畴，滑动窗口大小默认为30天，滑动单位为1天，为人气和销量创建从1天到30天的索引值，

### 3.2商品详情页

**在redis缓存除剩余库存外的商品详情，商品剩余库存用另外的redis KEY缓存**

### 3.3商品录入

商品入库以及商品信息变动时，自动更新相关索引，向MQ发送消息，订阅者负责更新elasticsearch索引

### 3.4商品索引

**只为3.3的MQ订阅者服务**

### 3.5查询商品类别

### 3.6查询商品剩余库存

### 3.7减库存

**接口名： CommonReturn cn.purchasenear.v1.buy.GoodsService.flashsales(long userId,String token,long goodsId,int amount)**

**调用时机：执行下单时，或执行限时抢购时**

**是否需要登录：是**

**额外说明：仅由下单或限时抢购使用，不对客户端开放。减库存时，使用update goods set stock=stock-#{amount} where stock=#{stock} and id=#{id}，完成之后删除相应商品库存的redis缓存**

**参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| userId | long | true | 登录的用户ID |
| token | string | true | 登录的token |
| goodsId | long | true | 要减库存的商品ID |
| amount | int | true | 减库存的数量 |

**返回值：SearchGoods，包含属性status和goodsList**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数类型 | 是否必须 | 说明 |
| status | int | 是 | 请求处理状态，1：成功，-3070101：未找到商品；-3070102：商品已下架；-3070103：剩余库存不够 |

## 4.为用户激活卖家功能

### 4.1激活卖家功能

**只有激活卖家功能之后才能显示卖家相关管理与查询界面**

**同时发送MQ消息，订阅者更新elasticsearch索引**

## 5.权限控制与管理后台

**另外再开一个系统吧，太多了，超范围了**