

廈門大學



面向对象分析与设计课程

顾客订单模块设计文档

组 别 3-4 组

组 员 黄子安 崔方博 范周喆 侯好欣 徐森彬

2023 年 12 月 27 日

目录

1、顾客模块..... 3

1.1 包图 3

1.2 对象模型..... 4

1.3 类图 5

1.4 状态图..... 6

1.5 数据库..... 7

1.6 程序逻辑..... 8

2、订单模块..... 13

2.1 包图 13

2.2 对象模型..... 14

2.3 类图 15

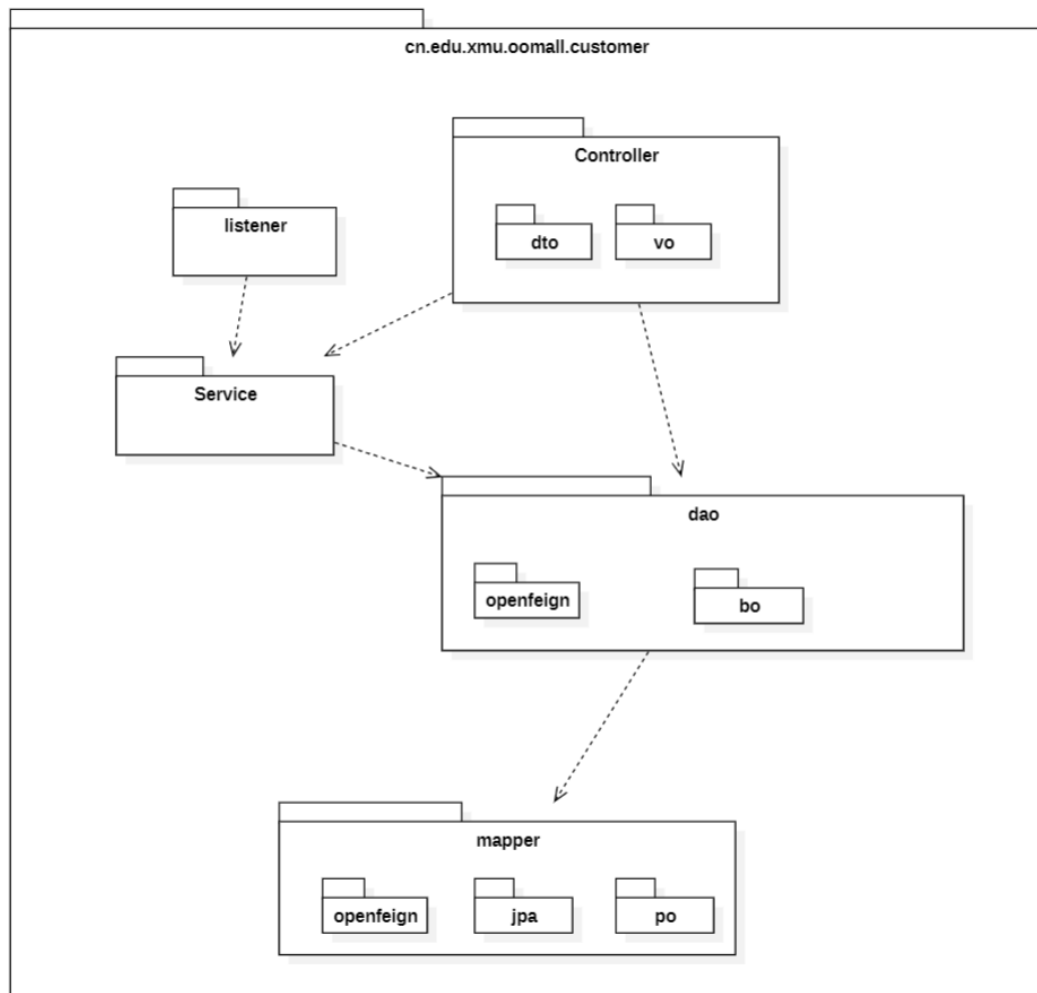
2.4 状态图..... 16

2.5 数据库..... 17

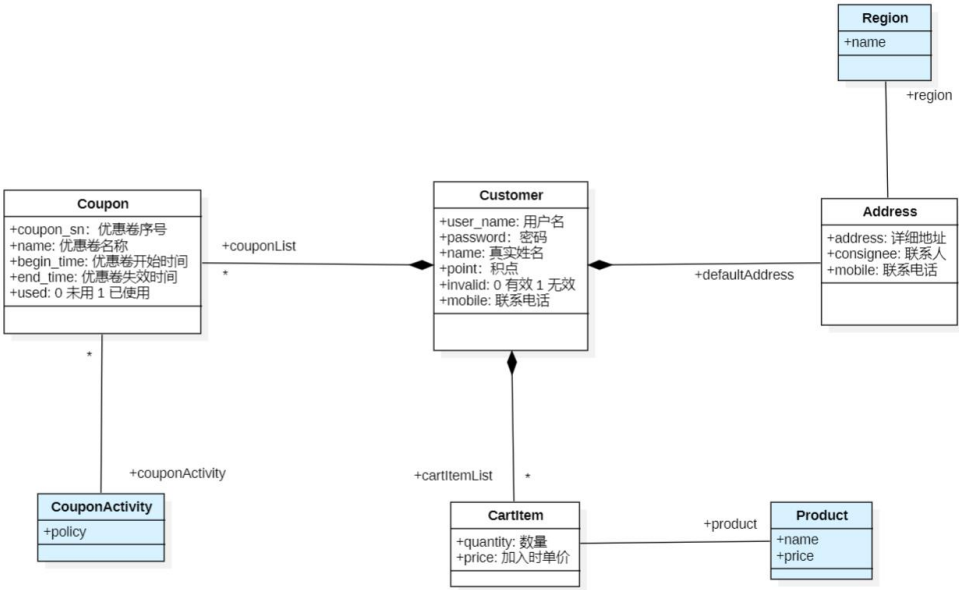
2.6 程序逻辑..... 18

1、顾客模块

1.1 包图

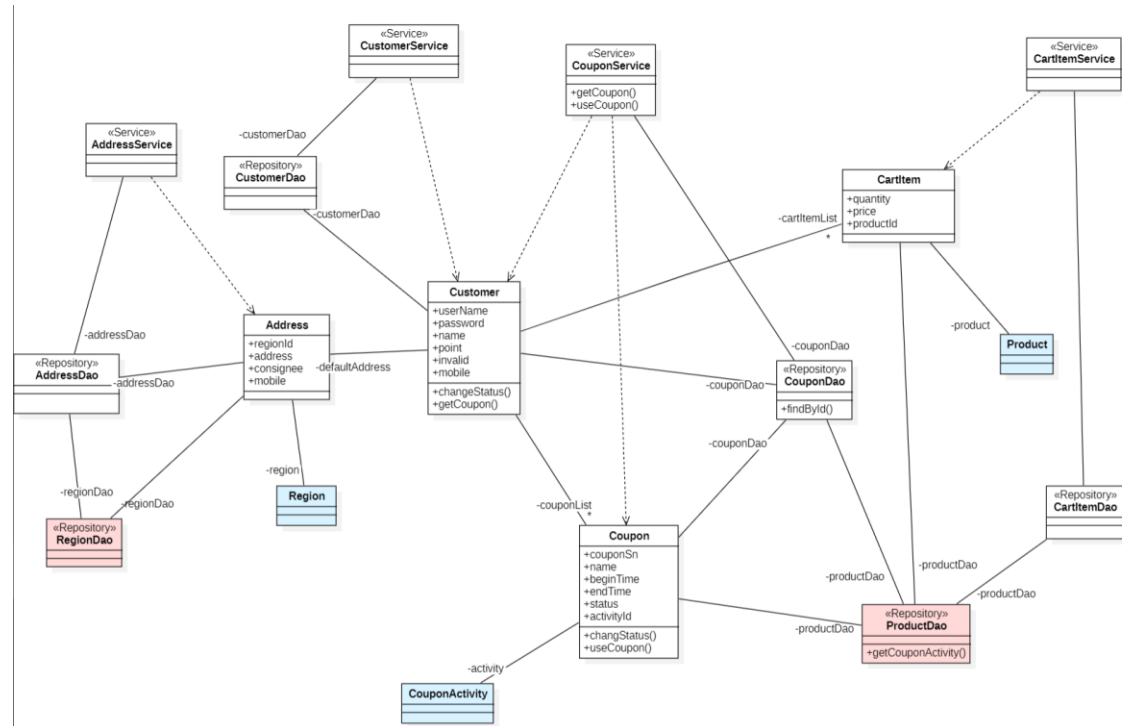


1.2 对象模型



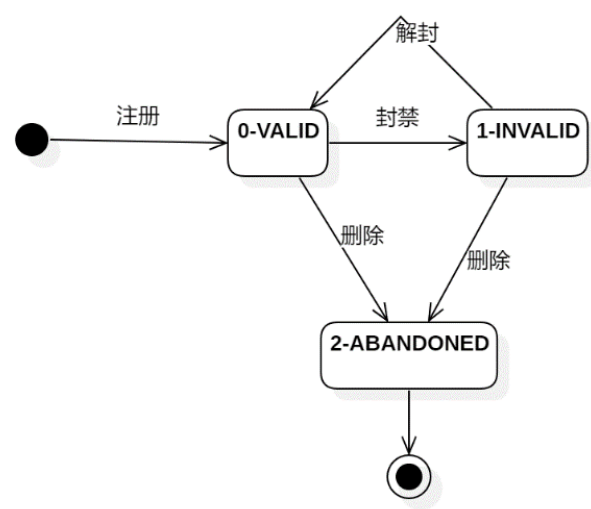
1.3 类图

API 在领取和使用优惠券时使用面向对象，其余 API 都是考虑使用面向功能，因此在类图上只体现优惠券部分的方法，其余 API 未实现所以没有添加对应的方法



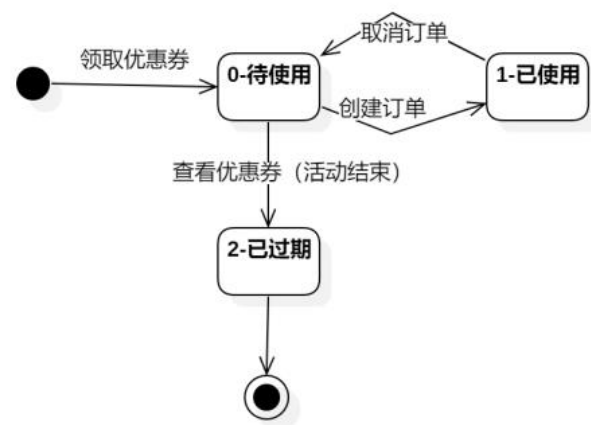
1.4 状态图

顾客状态机



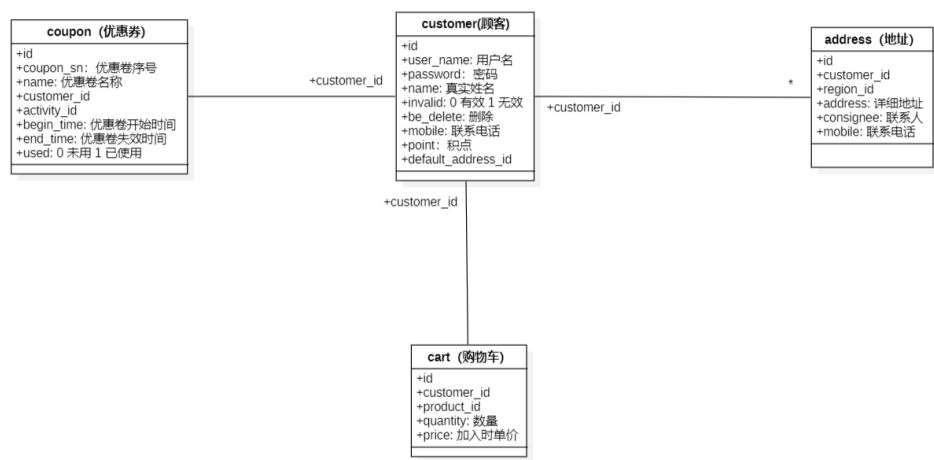
事件	API
注册	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/registerUser
封禁	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/banUser
解禁	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/releaseUser
删除 *	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/delUser

优惠券状态机



事件	API
领取优惠券	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/post_couponactivities_id_coupons
创建订单*	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/customerPostNewNormalOrder
取消订单	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/delete_shops_shopId_orders_id_(商家) https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/delete_orders_id_(顾客)
查看优惠券列表	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/showCoupons

1.5 数据库



在 customer 表中存储用户默认地址 id

1.6 API 与程序逻辑

(1) 领取优惠券

POST

/couponactivities
/{id}/coupons

买家领取活动优惠券，上线状态才能领取



- 判断优惠活动的quantityType为0，且用户无此优惠券，生成优惠券的数目为quantity
- 判断优惠活动的quantityType为1，且用户无此优惠券，去从优惠券中领一张优惠券

Parameters

Try it out

Name

Description

authorization * required

用户token

string
(header)

authorization

id * required

活动ID

integer(\$int64)
(path)

id

Responses

Response content type

application/json



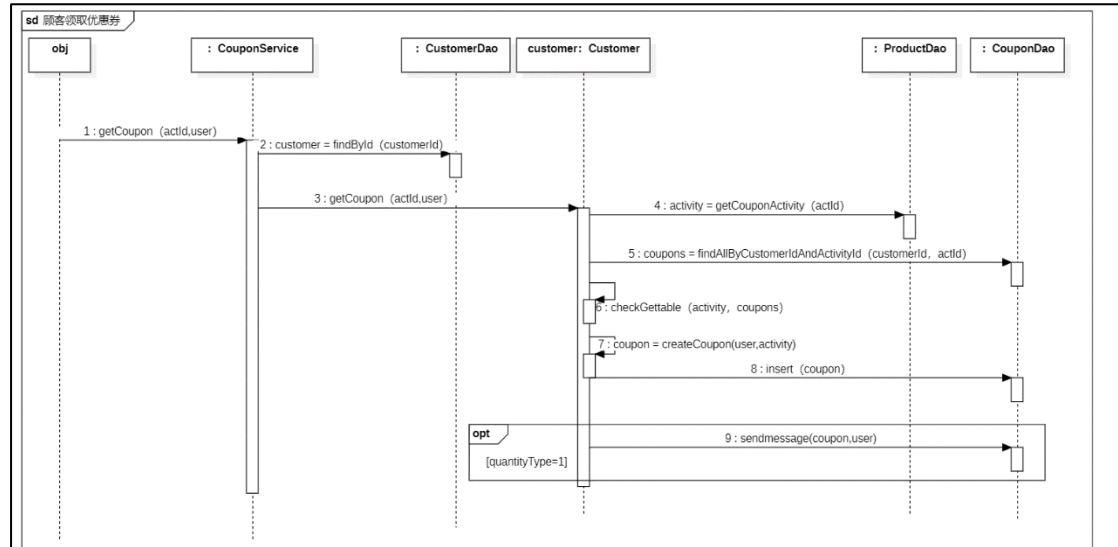
Code

Description

696	未到优惠券领取时间
697	优惠券领罄
698	优惠券活动终止
699	已到优惠券领取上线
default	成功

Example Value | Model

```
{
  "errno": 0,
  "errmsg": "成功",
  "data": [
    "string"
  ]
}
```






(2) 使用优惠券

PUT

/couponactivities
/{id}/coupons

买家使用优惠券，优惠券必须处于可用状态

- 验证优惠券是否处于可用状态且未过期
- 验证用户是否拥有此优惠券
- 使用优惠券后，更新优惠券状态

Parameters

Try it out

Name	Description
authorization * required string (header)	用户token <div>authorization</div>
id * required integer(\$int64) (path)	优惠券ID <div>id</div>

Responses

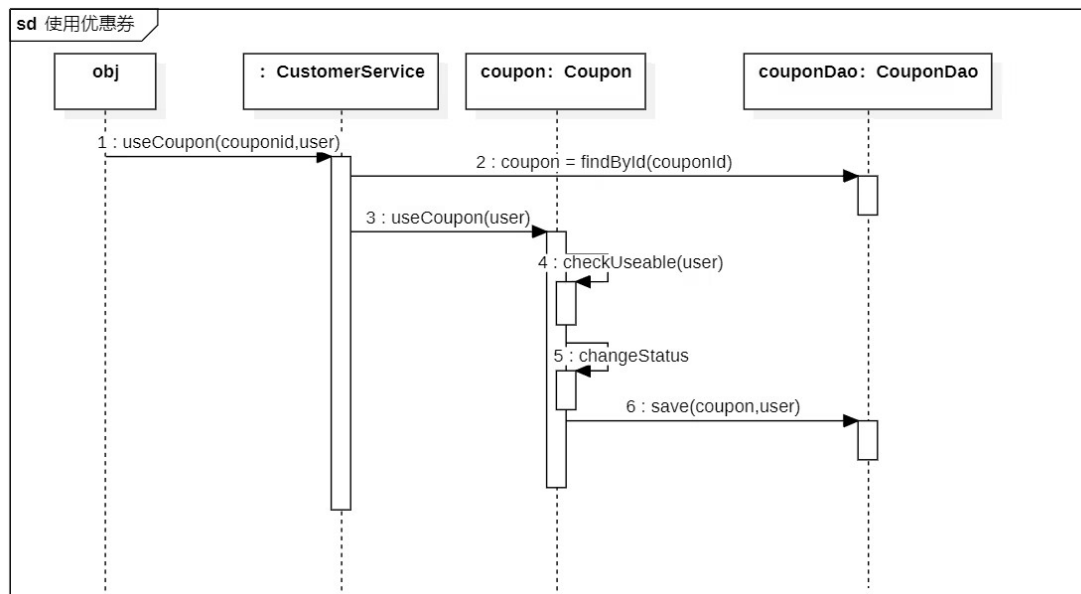
Response content type application/json

Code	Description
default	成功

Example Value

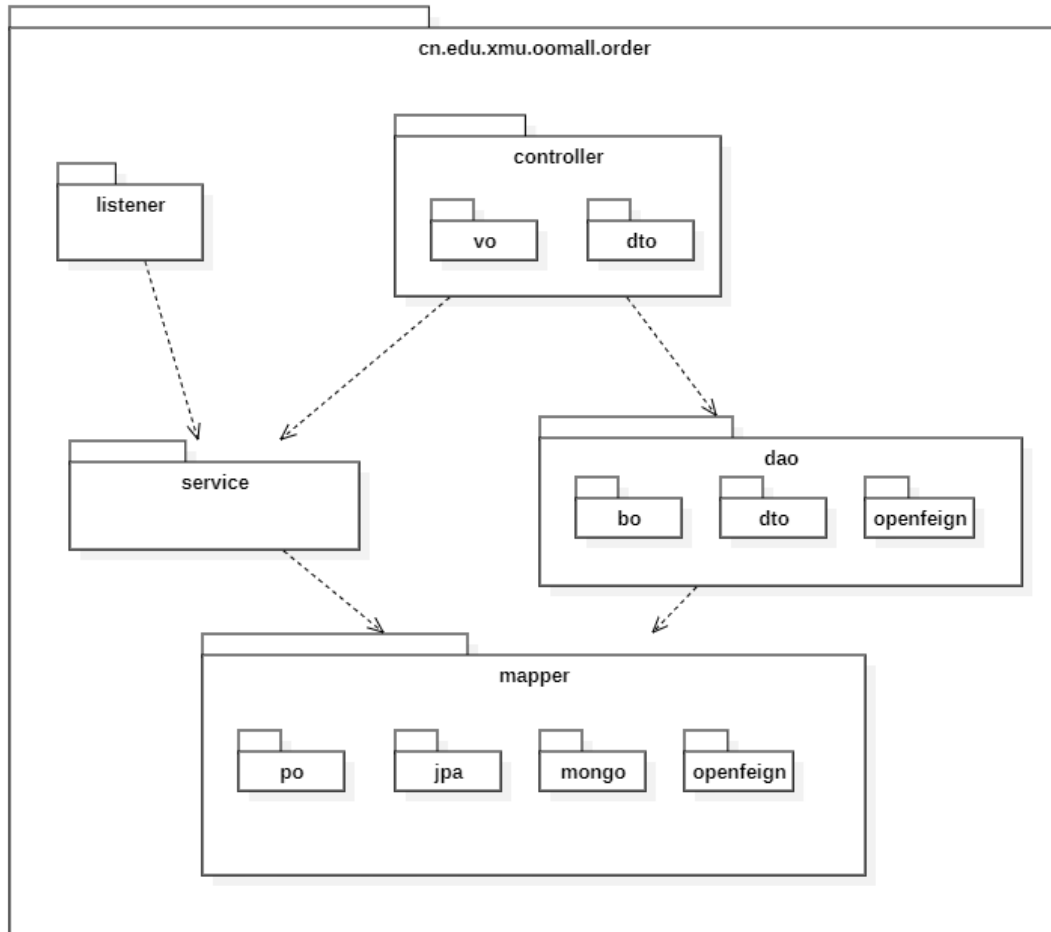
Model

```
{  "errno": 0,  "errmsg": "成功",  "data": {    "couponSn": "string"  }}
```

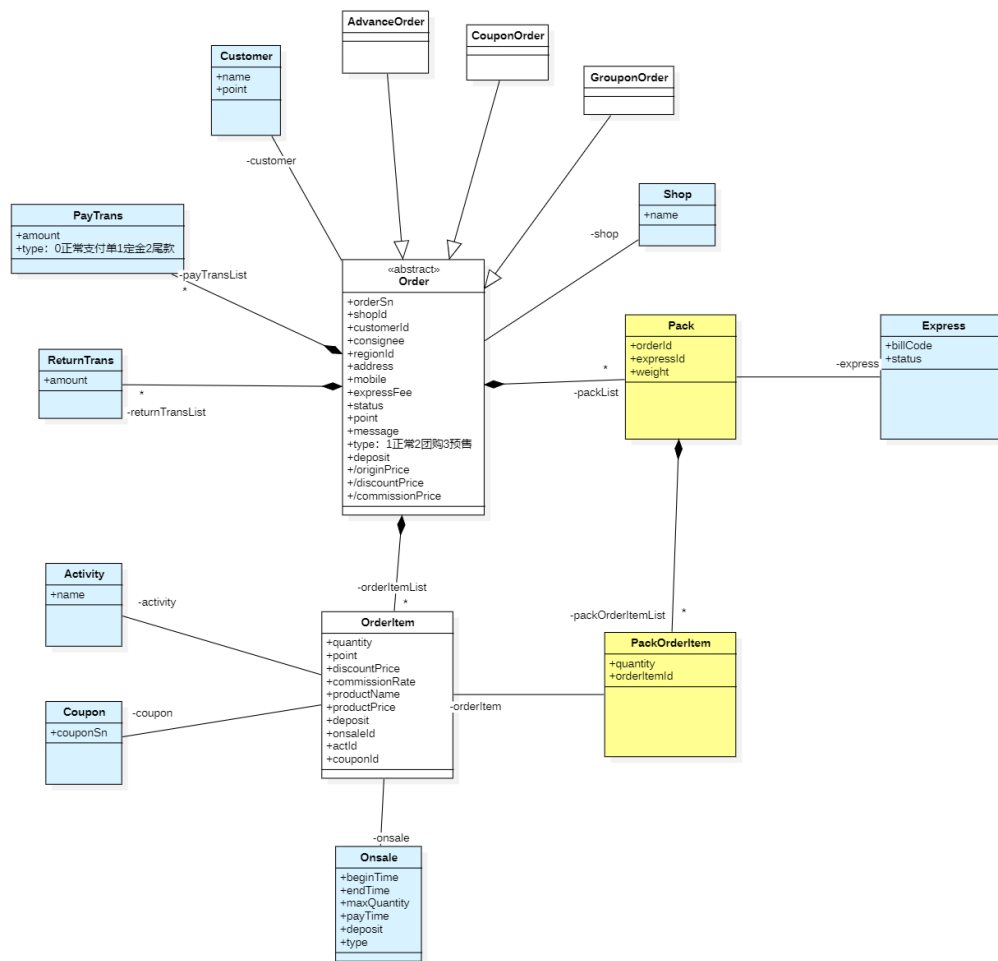


2、订单模块

2.1 包图



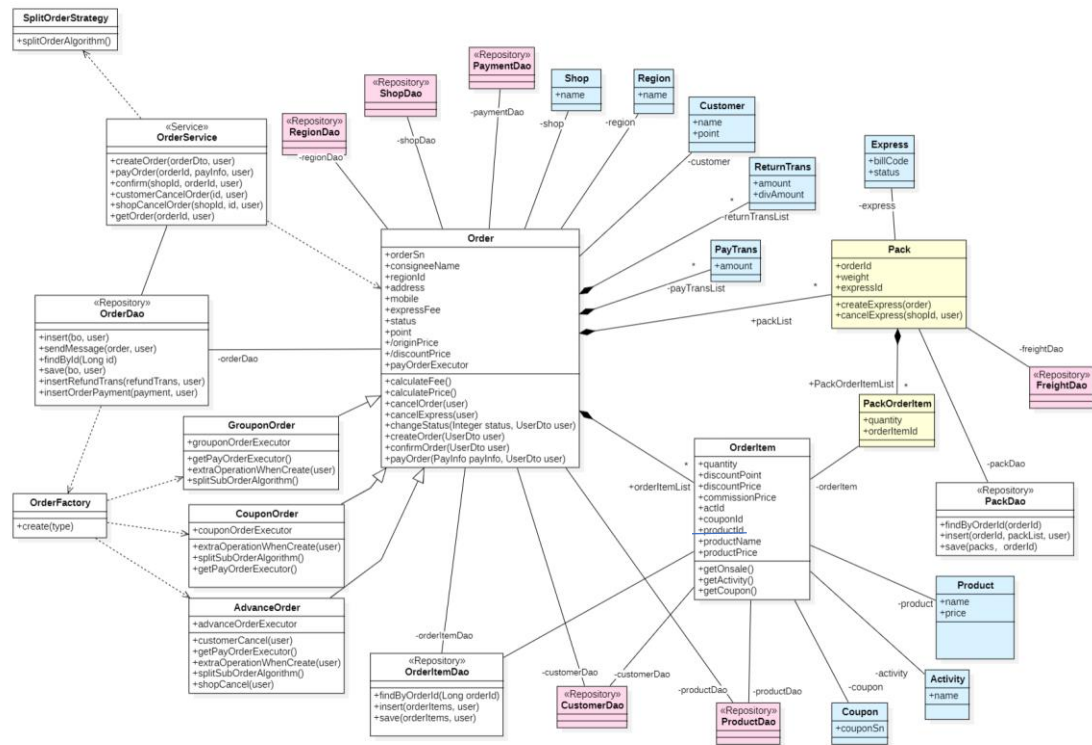
2.2 对象模型



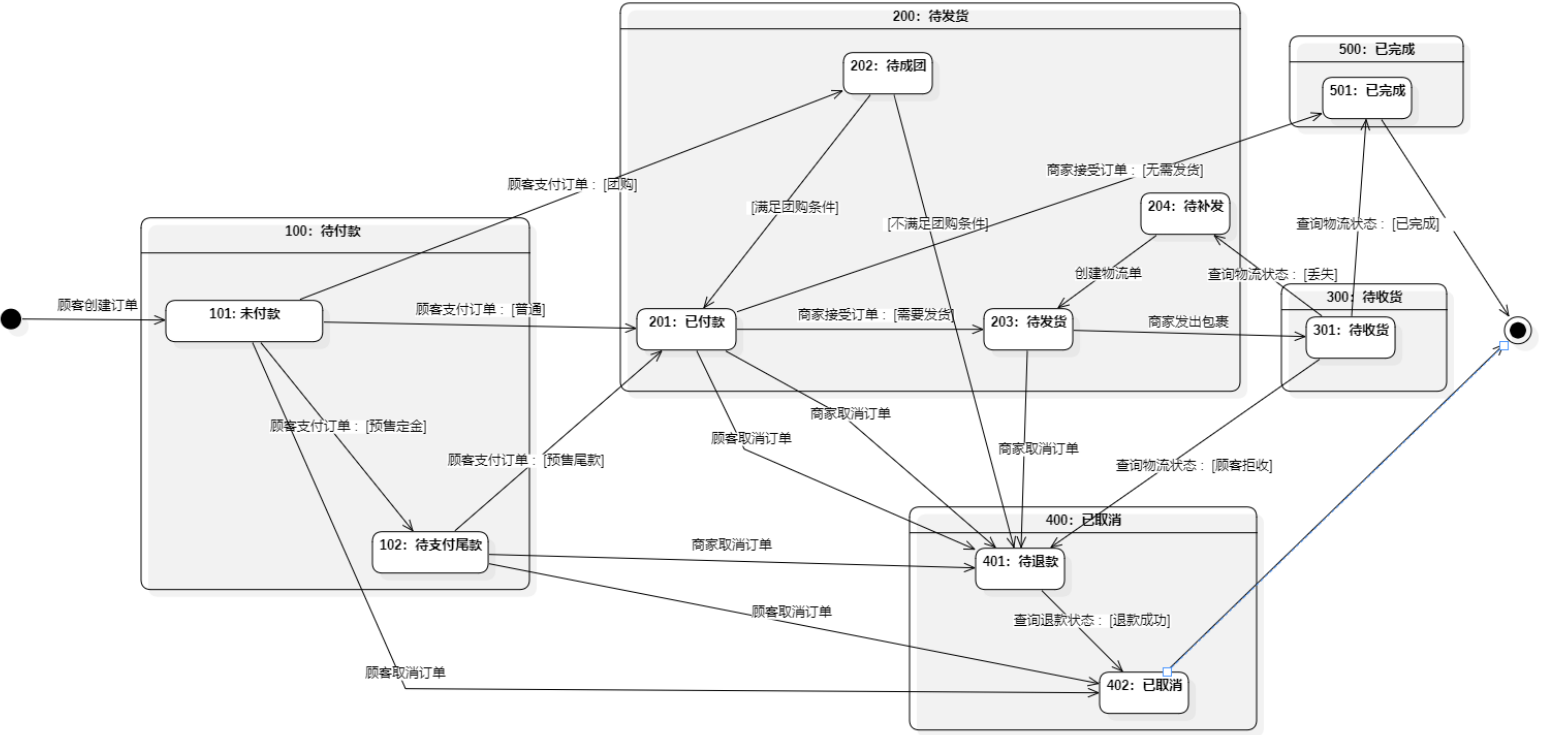
对象模型修改：

- 增加两个新的对象 Pack 和 PackOrderItem
- 增加 Pack：原本的对象模型无法实现分包裹发货的功能，因为没有有一个对象用于记录包裹里面包含的物品信息
- 增加 PackOrderItem：PackOrderItem 是对应特定的 OrderItem 的，一个包裹中未必存入了 OrderItem 中所有数量的商品（即可能 100 件的商品被分成了两个 50 件在不同包裹）
- 改进之后的好处：Order 与 Express 由原本的一对多的关系转化为了 Order 与 Pack 一对多关系，Pack 与 Express 一对一关系，可以知道一个包裹中有多少数量的商品，利于实现多包裹发货和包裹丢失补发
- Order 对象设置为抽象类，并产生 AdvanceOrder、CouponOrder、GrouponOrder 三个继承子类，利于代码的解耦合，可在不同的子类里定义不同功能的代码，实现对拓展开放，对修改封闭。

2.3 类图

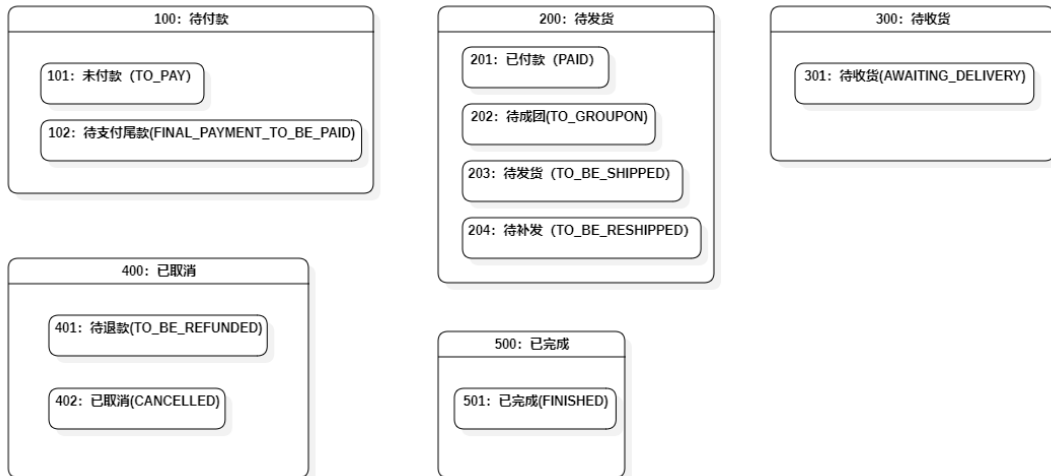


2.4 状态图



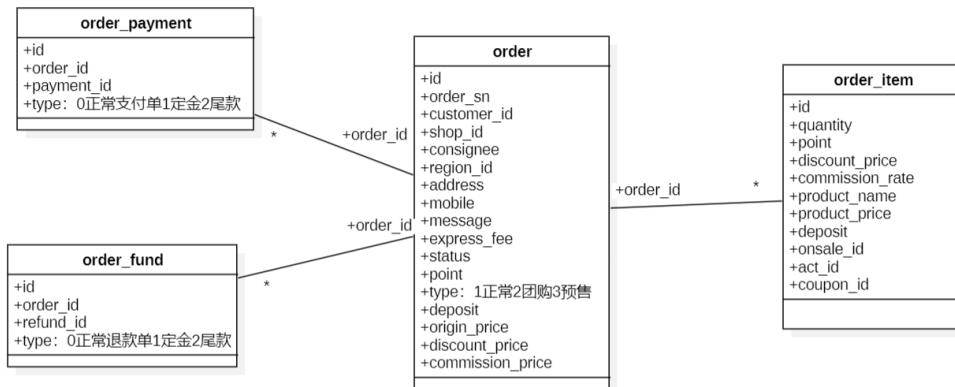
订单状态机图

事件	API
顾客创建订单*	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/customerPostNewNormalOrder
顾客支付订单	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/payOrder
顾客取消订单	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/delete_orders_id
商家取消订单	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/delete_shops_shopId_orders_id
商家接受订单*	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/postFreights
商家发出包裹*	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/putFreights



订单状态码

2.5 数据库



根据对象模型的颜色可以知道新增的两个对象放在了 MongoDB 之中，主要考虑以下几点

- Pack 和 PackOrderItem 与其他对象的关系相对复杂，且后续使用（主要在创建运单的时候）Pack 的时候也会需要使用 PackOrderItem（即两个对象一般是共同使用的），使用 MongoDB 的对象存储有助于简化存取流程
- Order 还是存在 MySQL：订单模块需要用到两个包裹对象的场合比较有限，不太希望在每次从数据库中读取 Order 的时候都把一连串的包裹全找出来，必然会影响性能，因此 Order 放在 MySQL 和包裹考虑分开存放




2.6 API 与程序逻辑

(1) 创建订单

orderItem 里加了 couponId (优惠券 id)

POST

/orders 买家申请建立订单 (普通, 团购, 预售) *

需要检查各类的id是否存在, 如果id不存在, 则未传对象处理,

Parameters

Try it out

Name	Description
authorization * required	用户token
string (header)	<input type="text" value="authorization"/>
orderInfo	指定新订单的资料
object (body)	<div>Example Value Model</div> <pre>{ "orderItems": [{ "onsaleId": 0, "quantity": 0, "couponId": 0, "actId": 0 }], "consignee": { "name": "string", "mobile": "string", "regionId": 0, "address": "string" }, "message": "string" }</pre> <div>Parameter content type</div> <div>application/json</div>

Responses

Response content type application/json

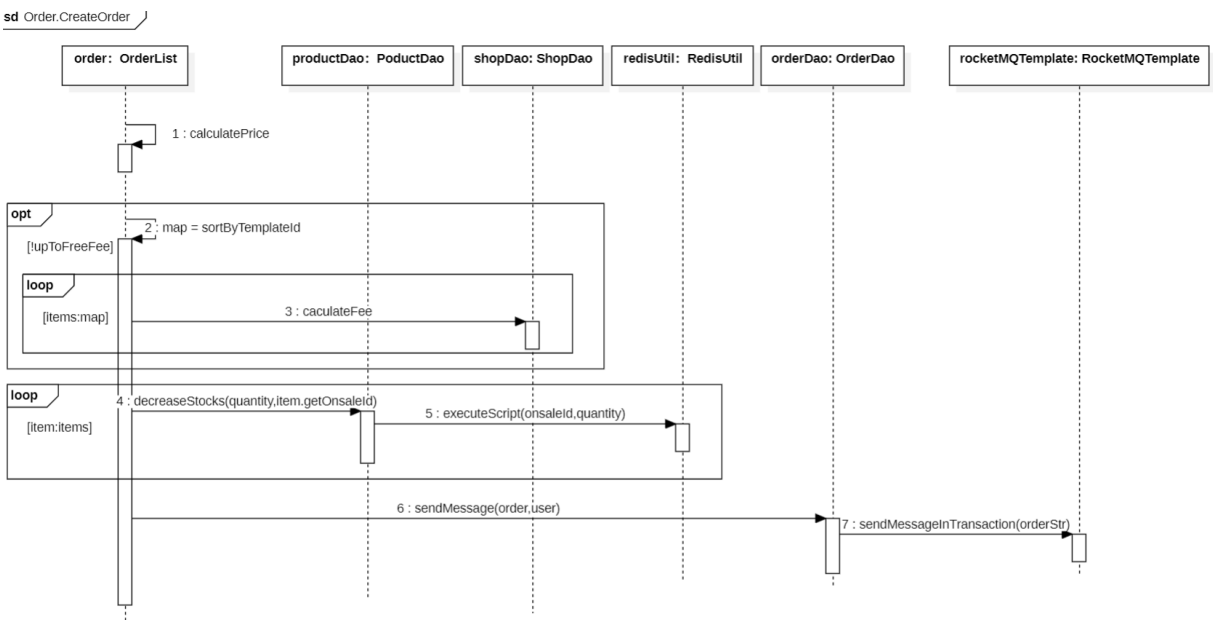
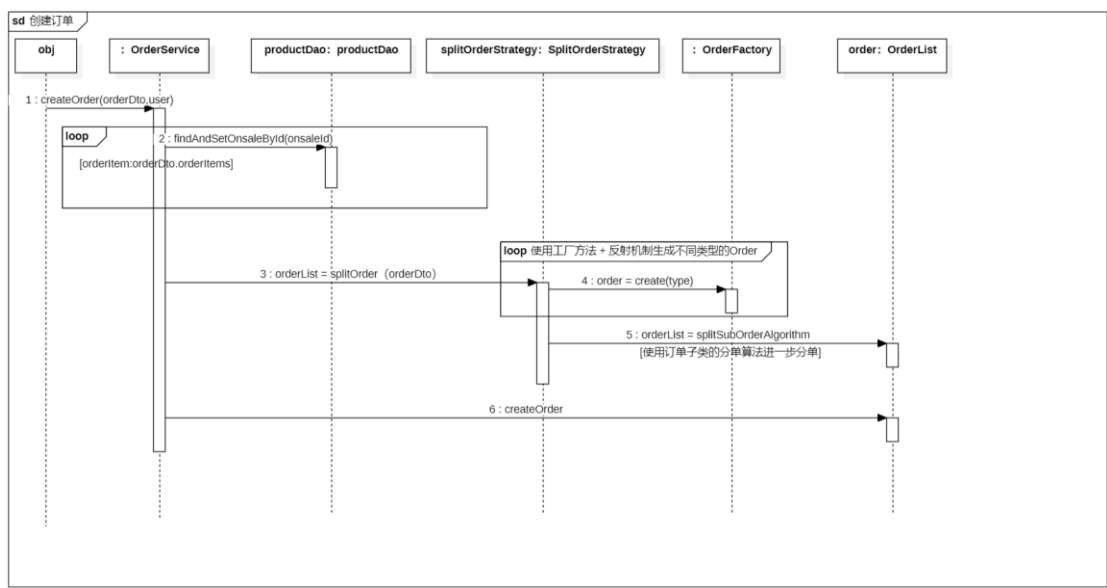
Code	Description
296	货品 (id = %d) 库存不足
802	销售对象(id=%d)的数量(%d)超过单次可购买数量(%d)
default	成功

Example Value | Model

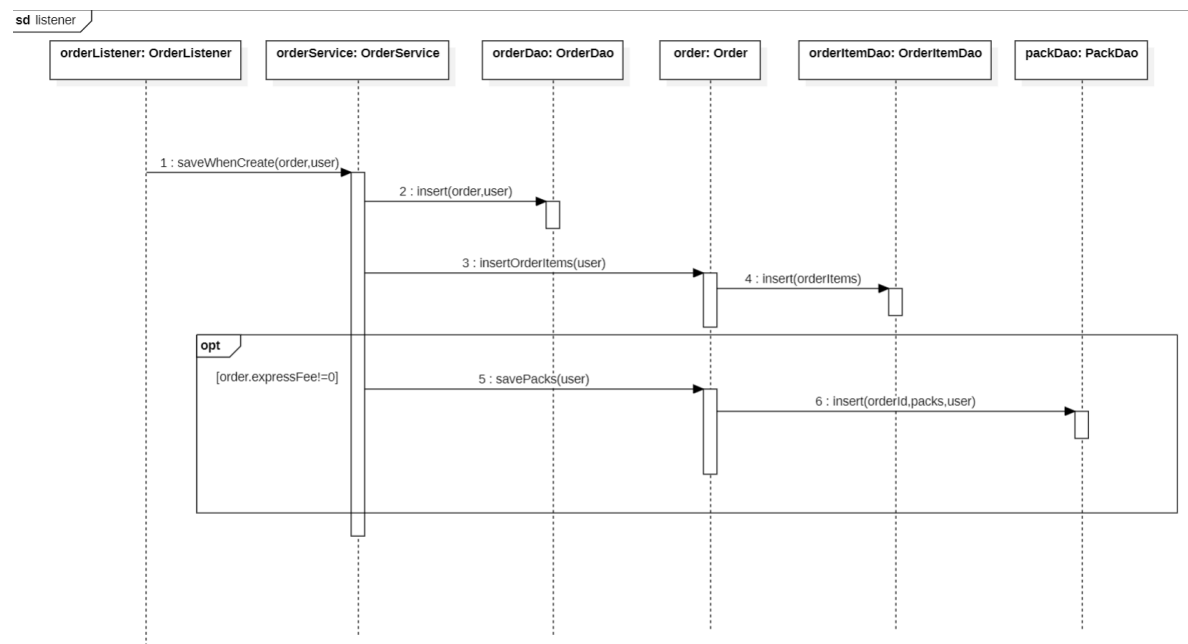
```
{
  "errno": 0,
  "errmsg": "成功"
}
```

这里时序图后面几个没有少画了开头的obj，还有Factory前面的对象名称也不可以少

MQ 发送消息为了保证事务需要先发送预订单，之后在 Listener 中完成存储



Listener 中要根据是否包邮了判断是否需要存储包裹



(2) 支付订单

支付订单的请求体里面删除了 `coupon` 列表，因为优惠券在创建订单时就已经使用了。

POST

/orders/{id}/payOrder 买家支付订单

📋 ↶ ↷

• 需要登录

Parameters

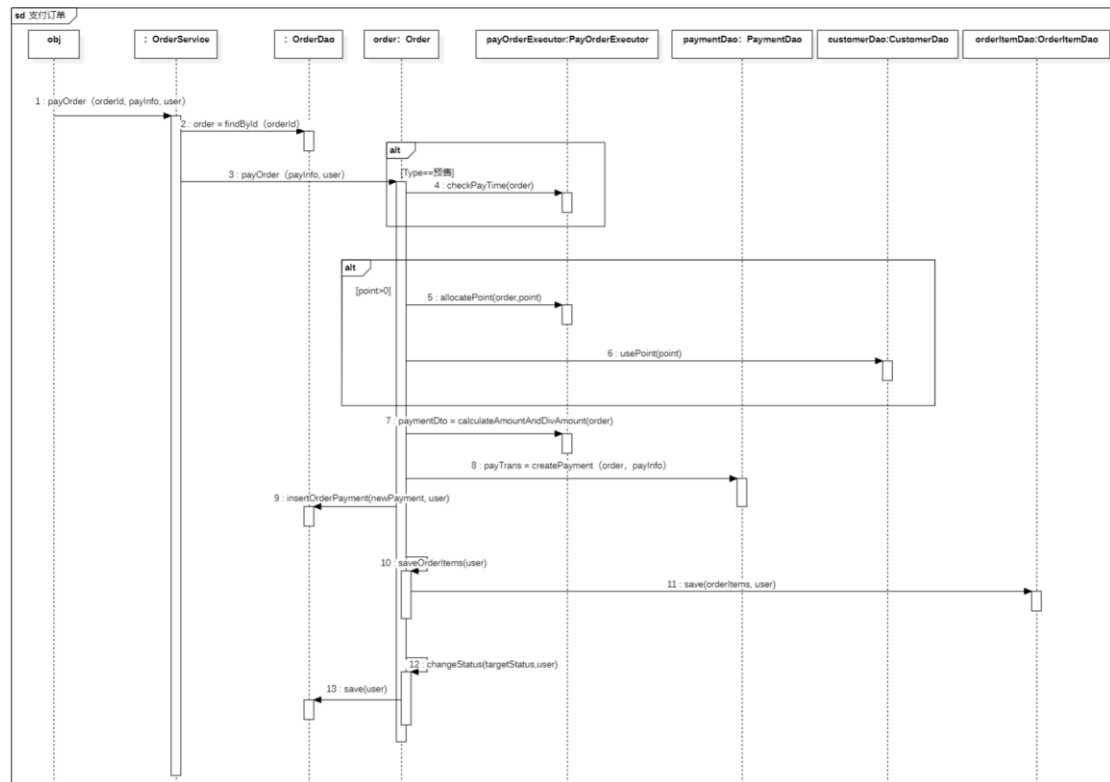
Try it out

Name	Description
authorization ★ required	用户token
string (header)	<input type="text" value="authorization"/>
id ★ required	订单id
integer(\$int64) (path)	<input type="text" value="id"/>
payInfo	支付信息
object (body)	<div>Example Value Model</div> <div><pre>{ "point": 0, "shopChannelId": 0}</pre></div> <div>Parameter content type</div> <div><input type="text" value="application/json"/></div>

Responses

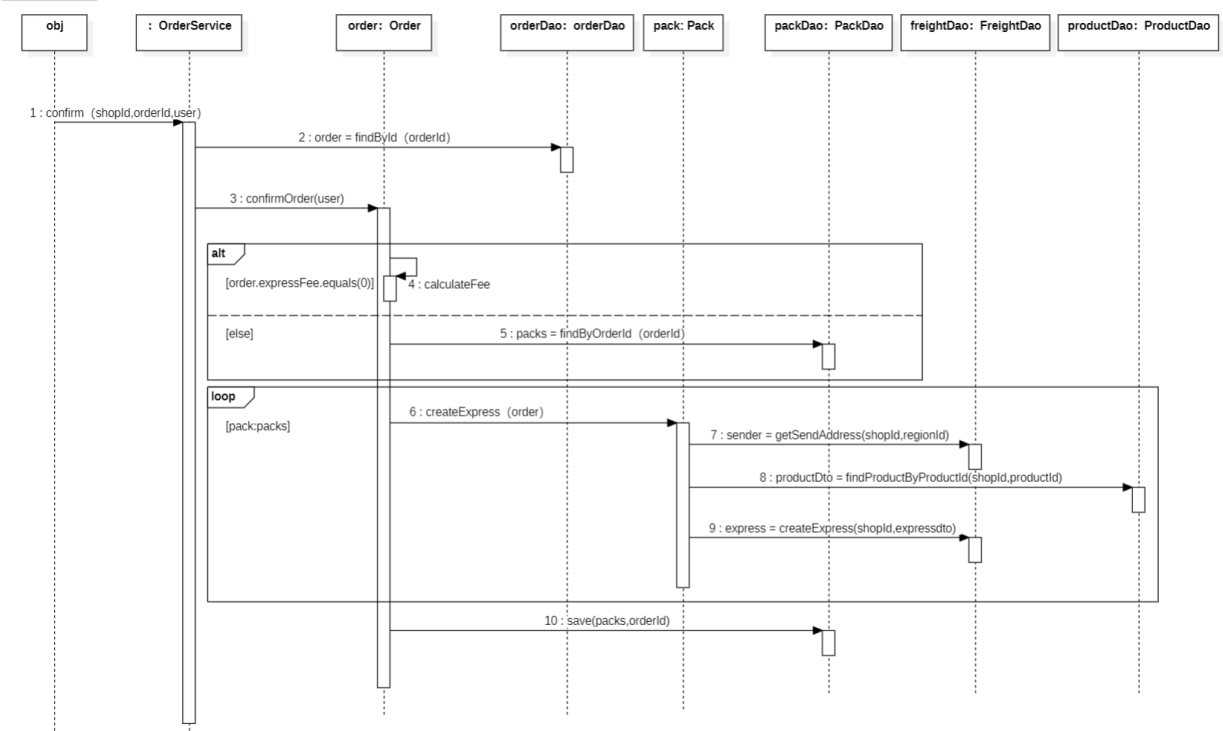
Response content type

Code	Description
default	成功
	<div>Example Value Model</div> <div><pre>{ "errno": 0, "errmsg": "成功"}</pre></div>



(3) 商家接受订单




sd 商家接受订单



(4) 顾客取消订单

DELETE

/orders/{id} 买家取消本人名下订单。

- 需要登录
- 用户本人调用本 API，只能取消本人的订单
- 发货前取消

Parameters

Try it out

Name	Description
authorization * required string (header)	用户token <div>authorization</div>
id * required integer (\$int64) (path)	订单id <div>id</div>

Responses

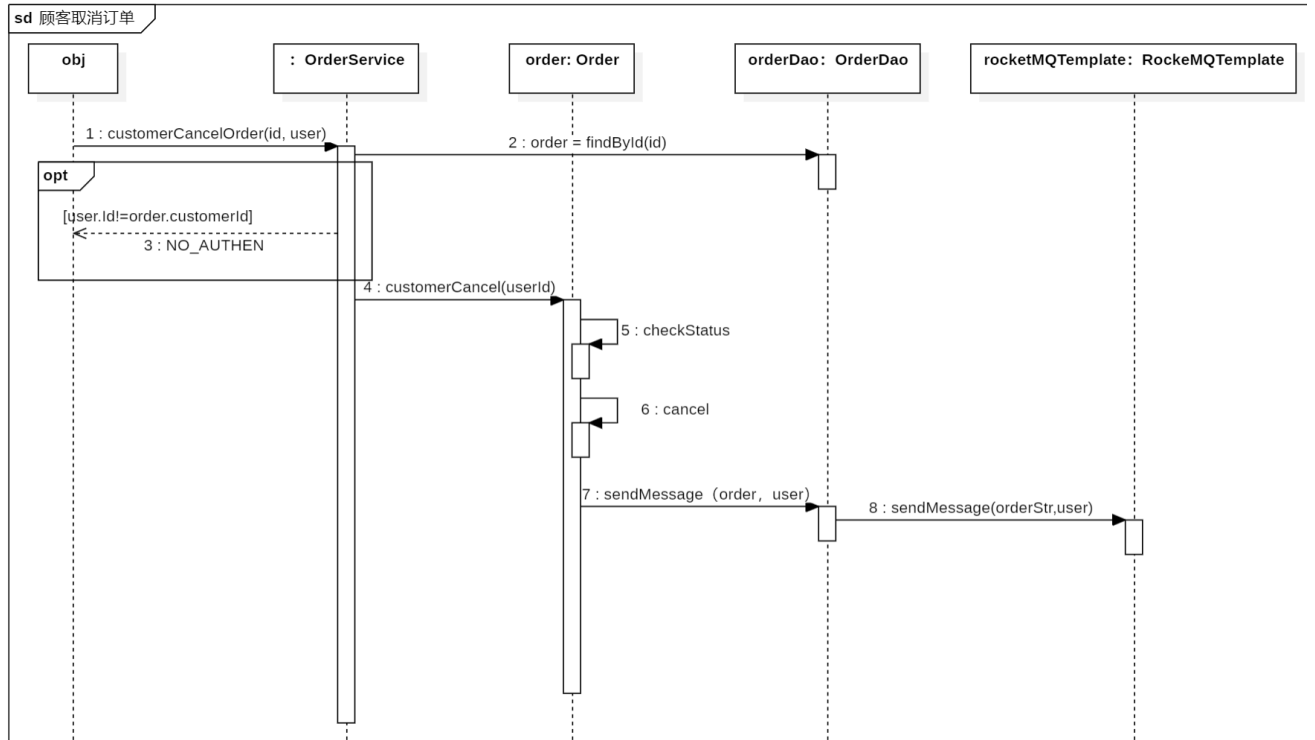
Response content type

application/json

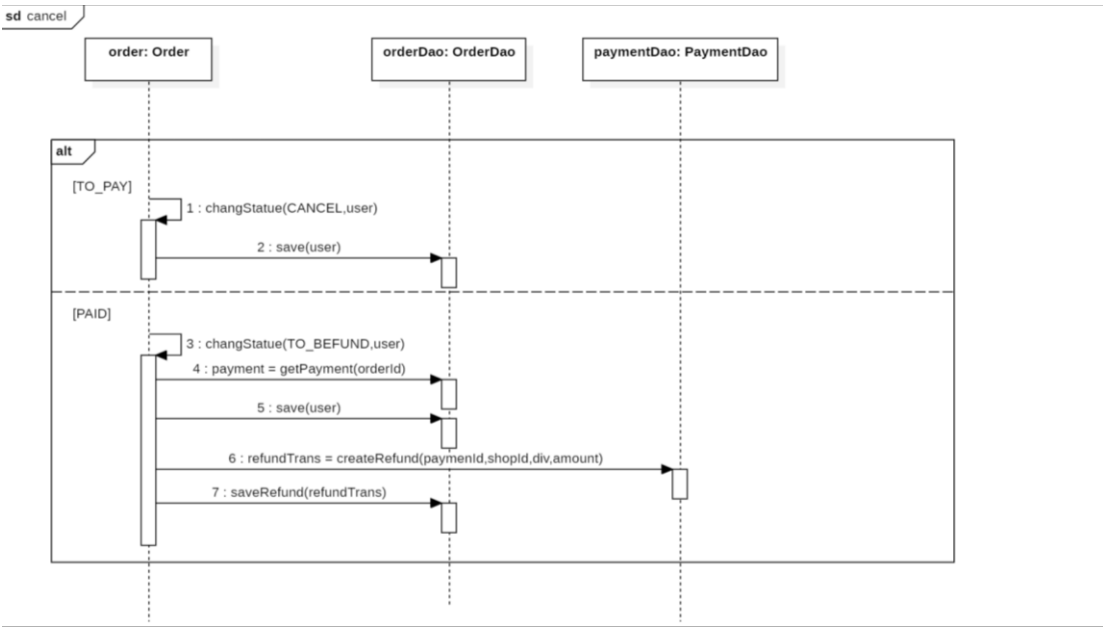
Code	Description
default	成功

Example Value | Model

```
{
  "errno": 0,
  "errmsg": "成功"
}
```

其中 cancel（非预售订单）时序图如下：



(5) 商家取消订单

DELETE

/shops/{shopId}/orders/{id} 管理员取消本店铺订单。

📋 ⬅️ ⬆️

• 需要登录

Parameters

Try it out

Name	Description
authorization * required string (header)	用户token <input type="text" value="authorization"/>
shopId * required integer(\$int32) (path)	商户id (店员只能查询本商铺) <input type="text" value="shopId"/>
id * required integer(\$int64) (path)	订单id <input type="text" value="id"/>

Responses

Response content type

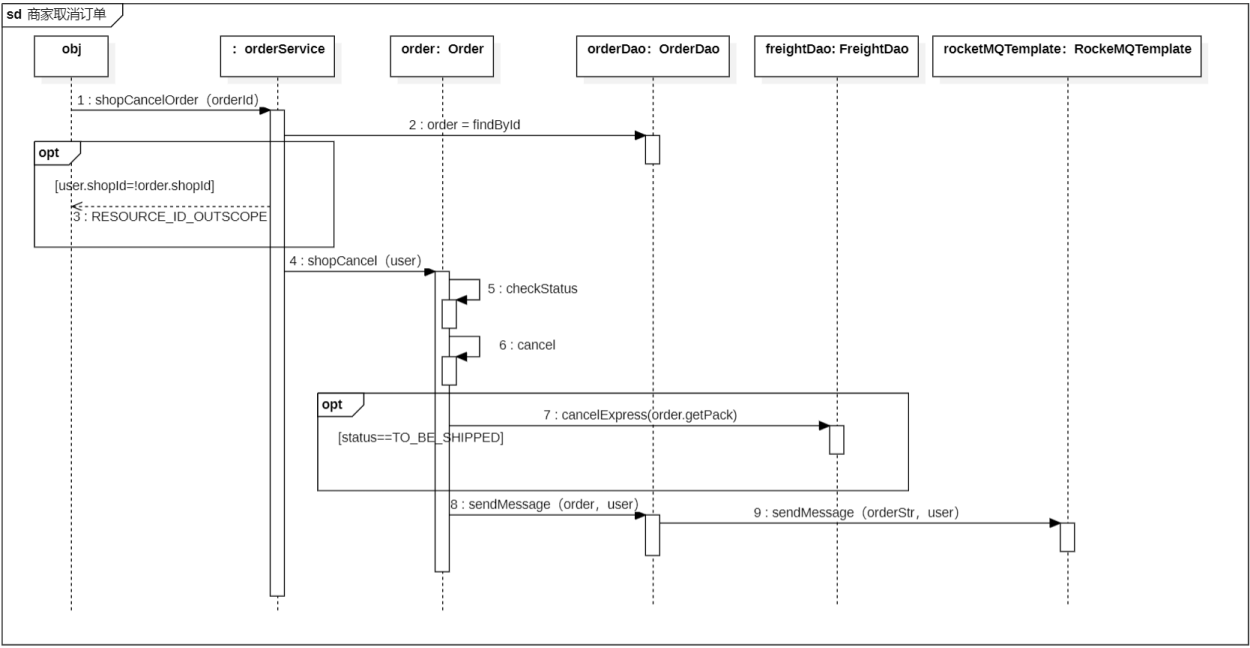
application/json ▼

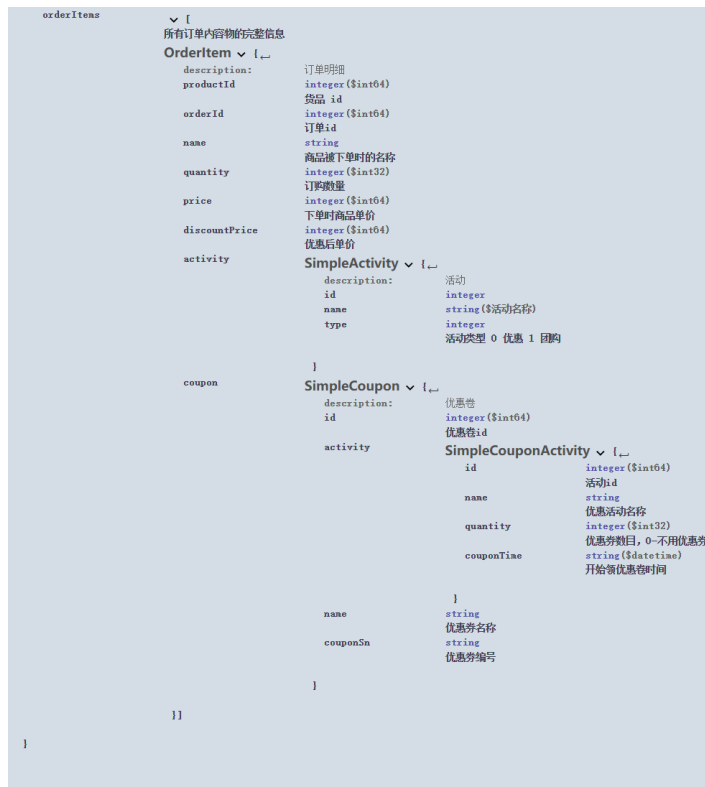
Code	Description
800	订单状态禁止
default	成功

Example Value | Model

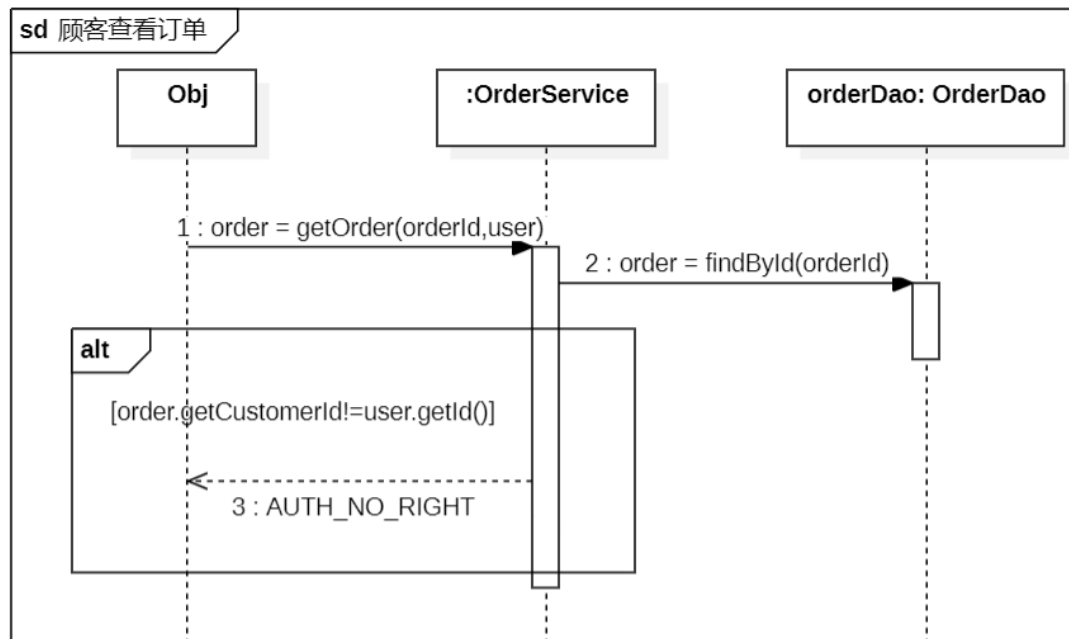
```
{  "errno": 0,  "errmsg": "成功"} 
```

商家取消订单从需求上多了可以取消待发货的订单，cancel 和顾客取消相同

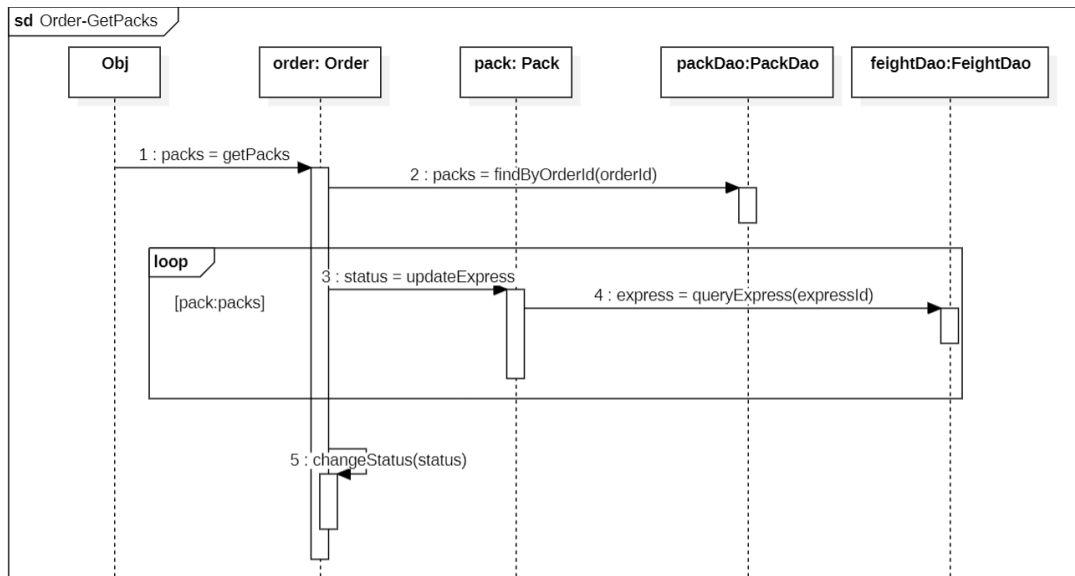




先查出来满血的 bo 对象 order，再用 order 获取关联的对象组装成 dto，order 来获取 customer 和 shop 和 packs，每个 orderItem 来获取 activity 和 coupon



查询 packs 的时候需要为每个包裹查询物流单来同步更新订单的状态，只有在查询订单的时候需要用物流单的信息更新订单的状态



*需要增加 shop 模块根据 shopId 查看商户的 id、name 信息的接口

*需要增加 customer 模块根据优惠券的 id 查找优惠券信息的接口