

廈門大學



信息学院软件工程系

面向对象分析与设计

题 目 OOMALL售后服务模块详细设计文档说明书

班级组别 软件工程2023级2班 2-7组

小组成员 庄子鲲 张瀚文 张琳 吴乐言 伊木然 苏一涵

任课老师 邱明

提交时间 2025年12月8日

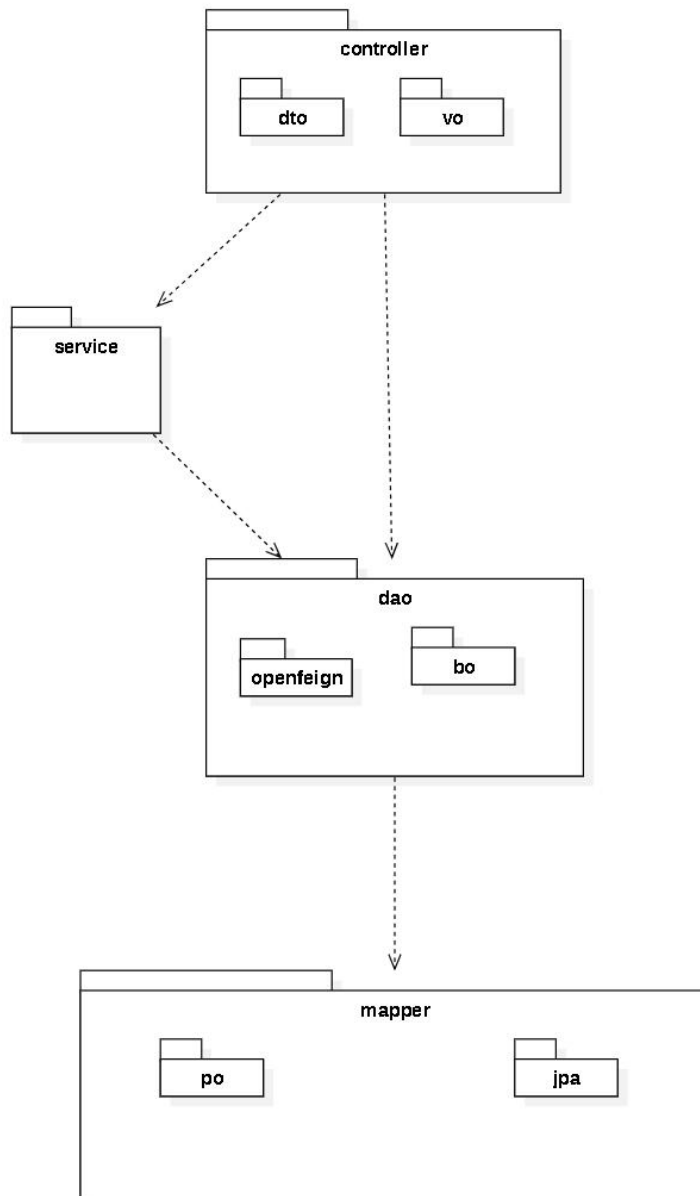
目录

信息学院软件工程系	1
面向对象分析与设计	1
一、售后模块	3
(一) 包图	3
(二) 数据库设计 (售后)	5
(三) 对象模型	5
(四) 组件设计	6
(五) 状态图	6
售后单状态机	6
(六) API与程序逻辑 (时序图)	8
1. 审核售后	8
2. 创建售后	9
3. 查询售后	12
4. 取消售后	14
二、服务模块	16
(一) 包图	16
(二) 数据库设计	18
(三) 对象模型	18
(四) 组件设计	20
(五) 状态图	20
服务单状态机	20
服务合同状态机	20
(六) API与程序逻辑 (时序图)	22
1. 服务商指派服务单对应处理员工	22
2. 取消服务单	23
3. 完成服务单	24
4. 取消服务合同	25

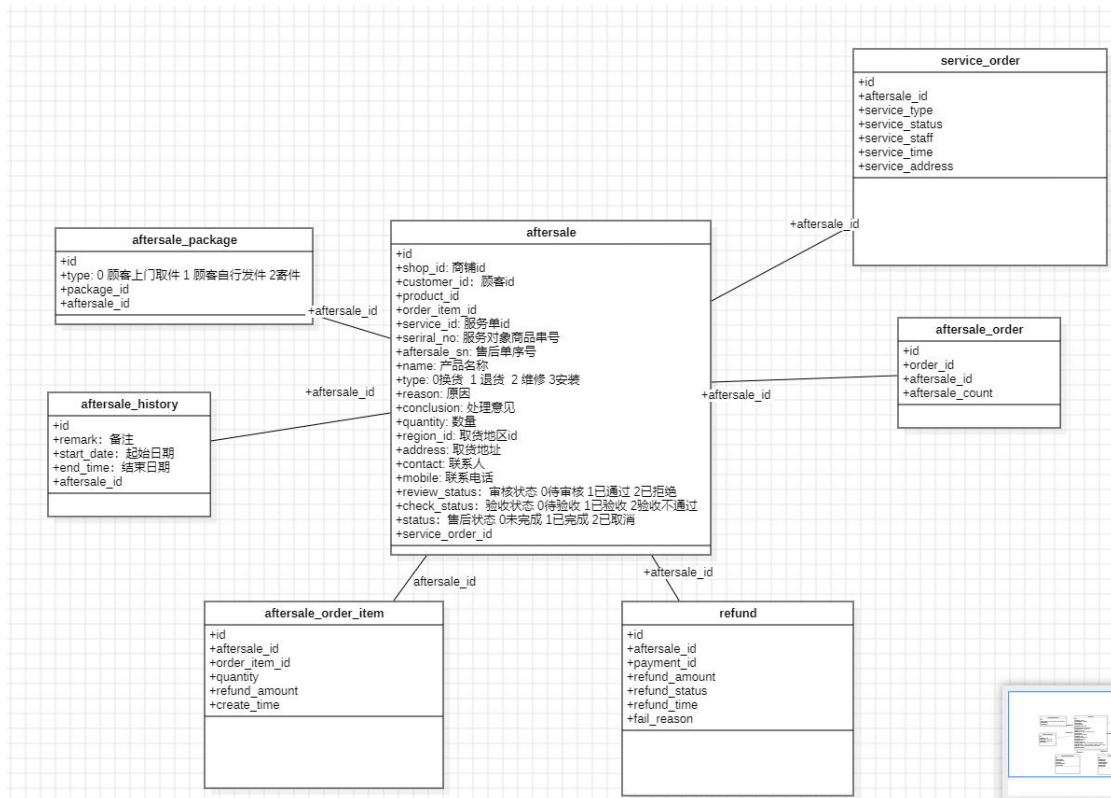
一、售后模块

(一) 包图

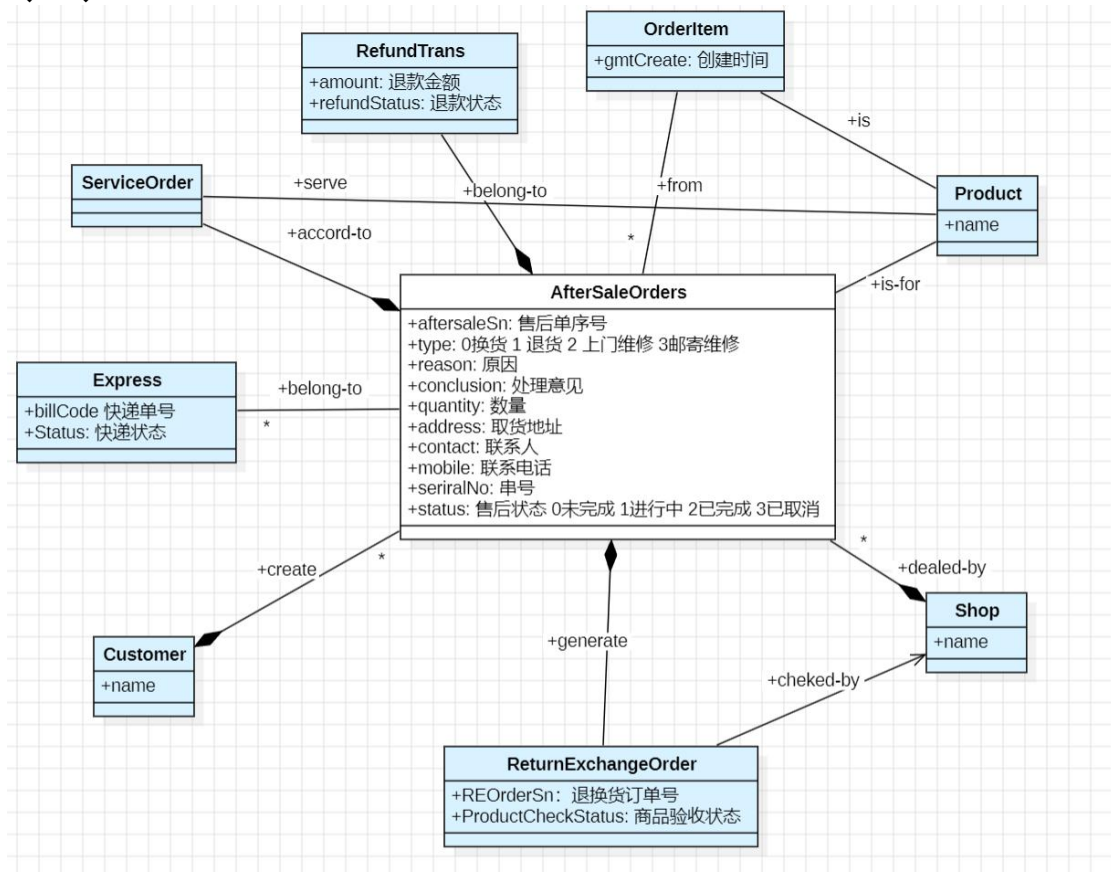
cn.edu.xmu.oomall.aftersale



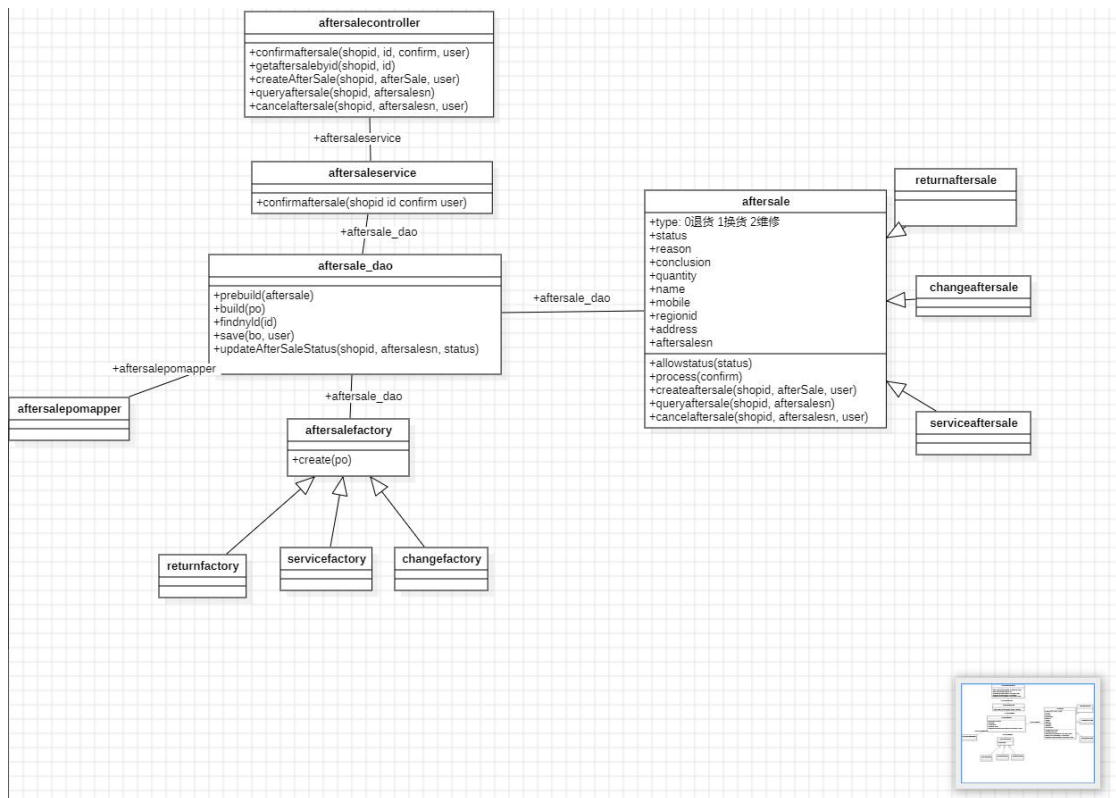
(二) 数据库设计 (售后)



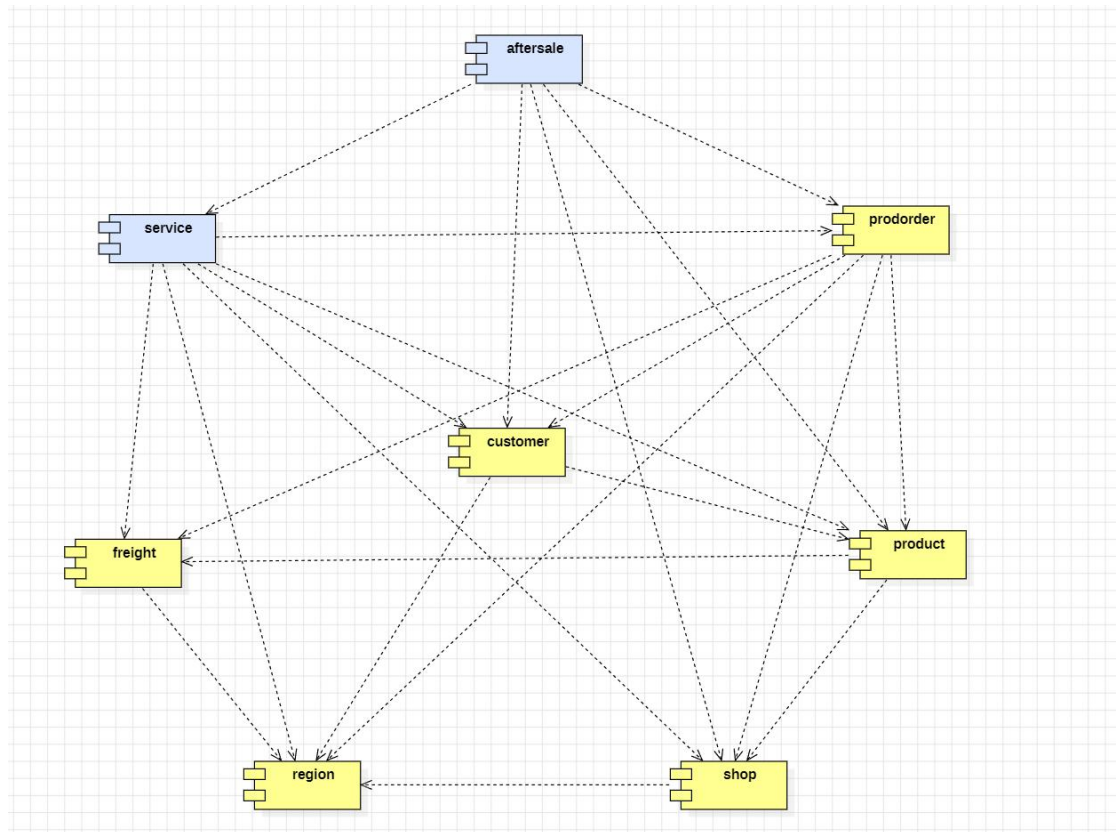
(三) 对象模型



退货退款如果是仅退款则只有一个退款单，否则产生一个退货订单，如果是换货会根据商品再创建一个OrderItem的订单

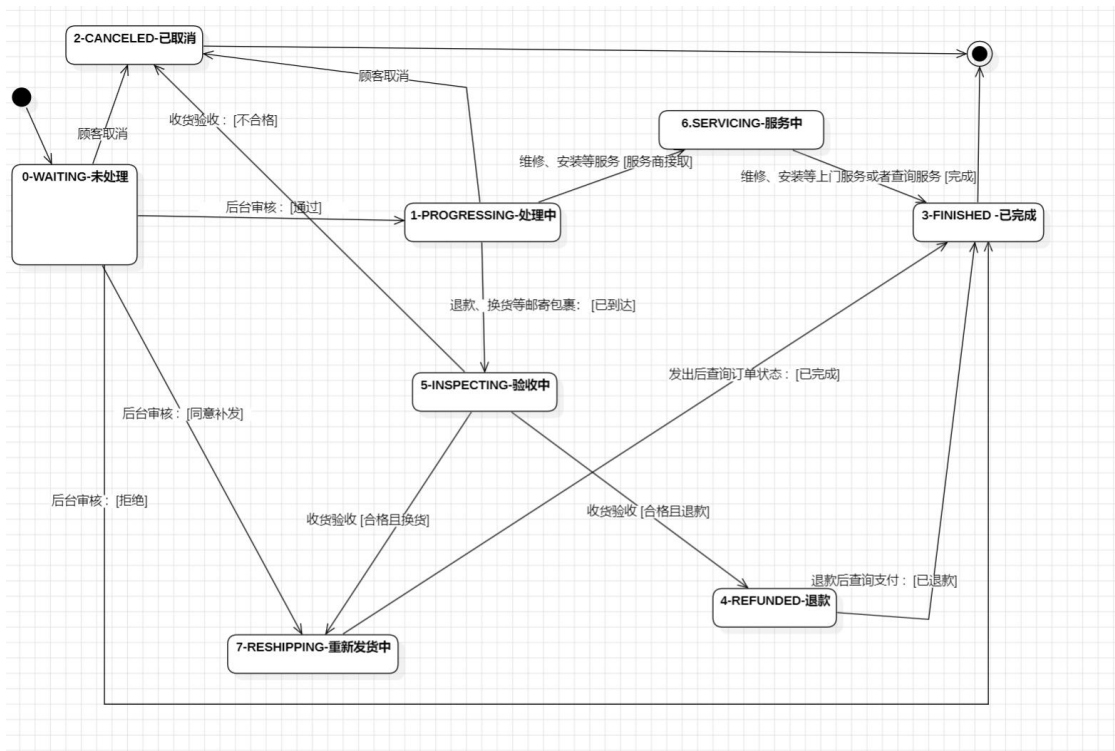


(四) 组件设计



(五) 状态图

售后单状态机



(六) API与程序逻辑（时序图）

1. 审核售后

<https://s.apifox.cn/82492a9f-ddfe-481b-a37c-f4cad68ae2da/380637116e0>

PUT

🔗

/shops/{shopId}/aftersales/{id}/confirm

▶ 调试

- 虚拟商品不需将商品退回，即可直接退款或者换货，虚拟商品的类别为313

请求参数

Path 参数

shopId	integer	店铺id	必需
id	integer	售后单id	必需

Header 参数

authorization	string	用户token	必需
---------------	--------	---------	----

Body 参数

application/json 必填

confirm	boolean	可选	示例 { "confirm": true, "conclusion": "string", "type": 0 }
conclusion	string	可选	
type	integer	可选	
修改类型, 0换货, 1退货, 2维修			

请求示例代码

Shell

JavaScript

Java

Swift

Go

PHP

Python

HTTP

C

C#

Objective-C

Ruby

OCaml

Dart

⋮

返回响应

● 200 成功

▼

application/json

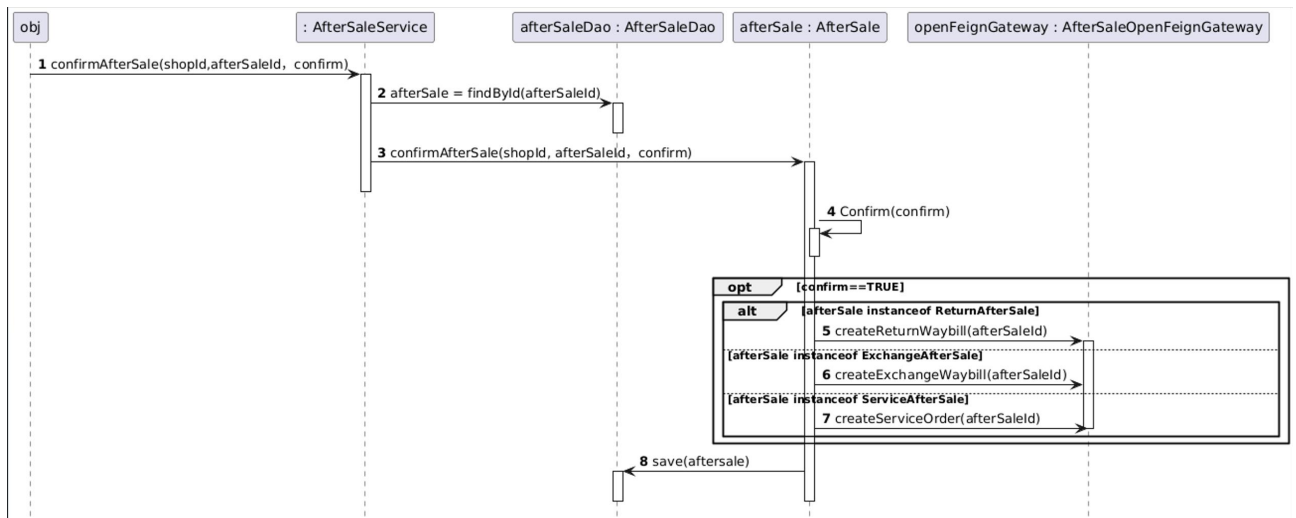
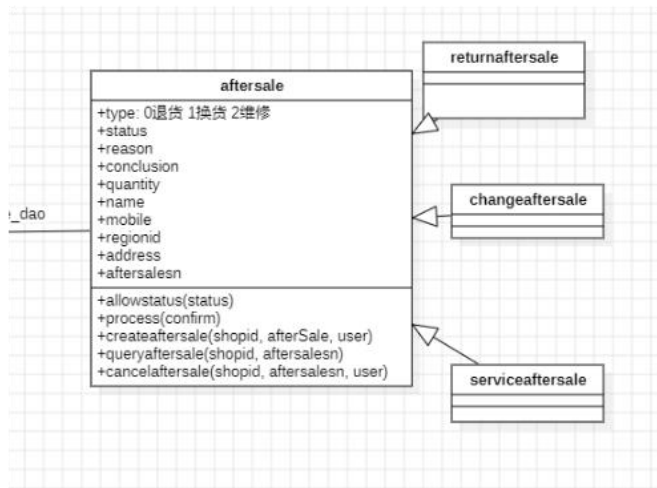
成功

Body

errno	integer	可选	示例 { "errno": 0, "errmsg": "成功" }
errmsg	string	可选	

示例值: 0

示例值: 成功



2. 创建售后

<https://s.apifox.cn/82492a9f-ddfe-481b-a37c-f4cad68ae2da/387349687e0>

请求参数

Path 参数

id integer 订单明细id

必需

Header 参数

authorization string 用户token

必需

Body 参数

application/json 必填

type integer
0换货, 1退货, 2维修

quantity integer

reason string

consignee object (Consignee) Consignee

- name string 姓名
- mobile string 电话
- regionId integer 地区id
- address string 详细地址

可选

可选

可选

可选

可选

可选

可选

可选

示例

```
{
  "type": 0,
  "quantity": 0,
  "reason": "string",
  "consignee": {
    "name": "string",
    "mobile": "string",
    "regionId": 0,
    "address": "string"
  }
}
```

请求示例代码

Shell
 JavaScript
 Java
 Swift
 Go
 PHP
 Python
 HTTP
 C
 C#
 Objective-C
 Ruby
 OCaml
 D

返回响应

200 成功

application/json

成功

Body

errno integer
示例: 0

errmsg string
示例: 成功

data object

- id integer 售后单id
- afterSaleSn string 售后单序号
- type integer
0换货, 1退货, 2维修
- conclusion string 处理意见
- quantity integer 货物数量
- status integer 状态

可选

可选

可选

可选

可选

可选

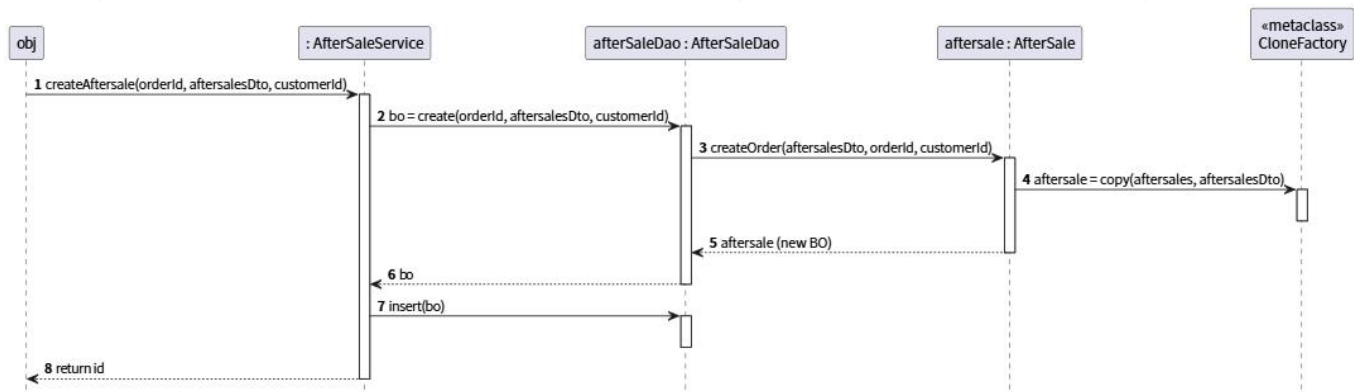
可选

可选

可选

示例

```
{
  "errno": 0,
  "errmsg": "成功",
  "data": {
    "id": 0,
    "afterSaleSn": "string",
    "type": 0,
    "conclusion": "string",
    "quantity": 0,
    "status": 0
  }
}
```



3. 查询售后

<https://s.apifox.cn/82492a9f-ddfe-481b-a37c-f4cad68ae2da/380637113e0>

GET

/shops/{id}/aftersales

调试

管理员可通过售后单ID查看所有售后单

请求参数

Path 参数

生成代码

id	integer	店铺id	必需
----	---------	------	----

Query 参数

beginTime	string	开始时间	可选
endTime	string	结束时间	可选
page	integer	页码	可选
pageSize	integer	每页数目	可选
type	integer	售后类型	可选
state	integer	售后状态	可选
customerName	string	顾客名称	可选

Header 参数

authorization	string	用户token	必需
---------------	--------	---------	----

请求示例代码

Shell

JavaScript

Java

Swift

Go

PHP

Python

HTTP

C

C#

Objective-C

Ruby

OCaml

Dart

:

返回响应

200 成功

application/json

errno

integer

可选

errmsg

string

可选

data

object

可选

page

integer

页码

可选

pageSize

integer

每页数目

可选

list

array(object (SimpleAftersale))

可选

id

integer

售后单id

可选

aftersaleSn

string

售后单序号

可选

type

integer

0退换货, 1退货, 2维修

可选

conclusion

string

处理意见

可选

quantity

integer

货物数量

可选

status

integer

状态

可选

inArbitrated

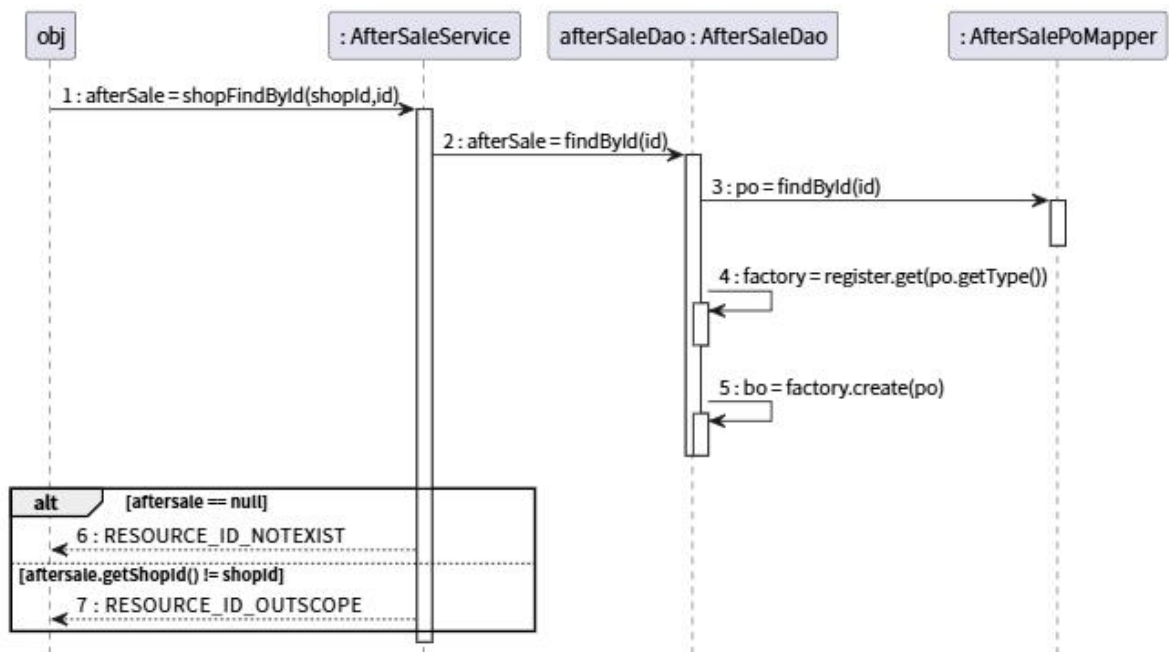
integer

是否仲裁中 0 未仲裁 1 仲裁中

可选

示例

```
{  "errno": 0,  "errmsg": "string",  "data": {    "page": 0,    "pageSize": 0,    "list": [      {        "id": 0,        "aftersaleSn": "string",        "type": 0,        "conclusion": "string",        "quantity": 0,        "status": 0,        "inArbitrated": 0      }    ]  } }
```



4.取消售后

<https://s.apifox.cn/82492a9f-ddfe-481b-a37c-f4cad68ae2da/380646252e0>

PUT

/shops/{shopId}/aftersales/{id}/cancel

▶ 调试

只可以在处理中的状态取消

请求参数

Path 参数

shopId

integer

店铺id

必需

id

integer

售后单id

必需

Header 参数

authorization

string

用户token

必需

生成代码

Body 参数

application/json

必填

reason

string | null

取消原因

可选

示例

{
 "reason": "string"
}

请求示例代码

Shell

JavaScript

Java

Swift

Go

PHP

Python

HTTP

C

C#

Objective-C

Ruby

OCaml

Dart

⋮

返回响应

200 成功

application/json

成功

Body

errno

integer

可选

示例值: 0

errmsg

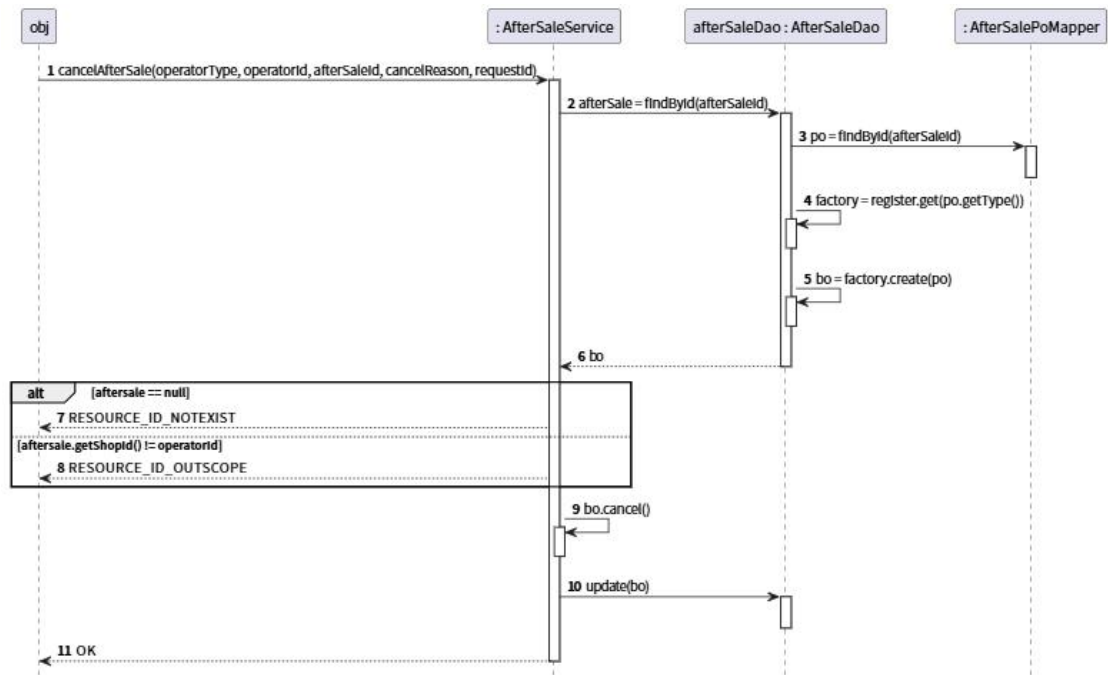
string

可选

示例值: 成功

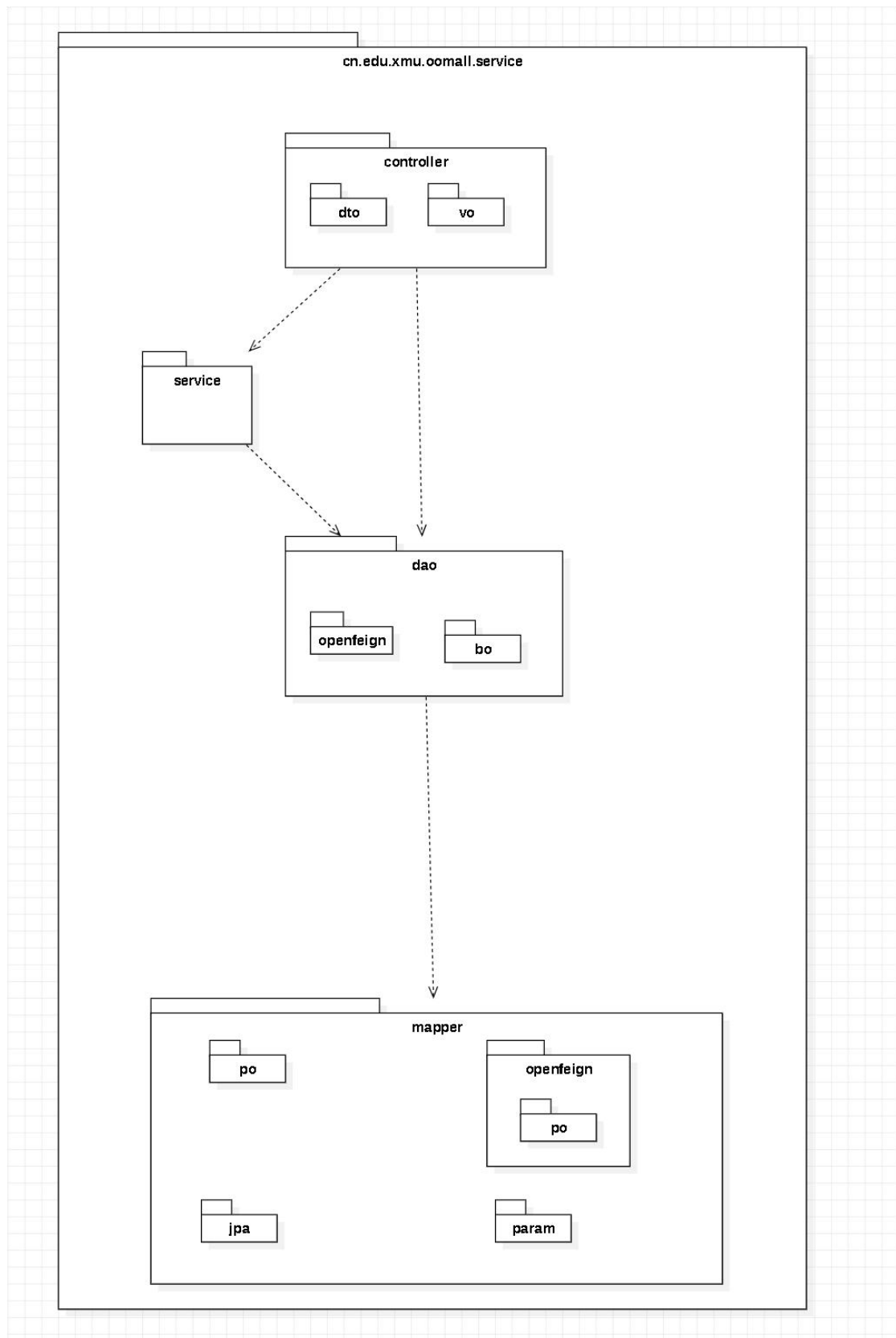
示例

{
 "errno": 0,
 "errmsg": "成功"
}

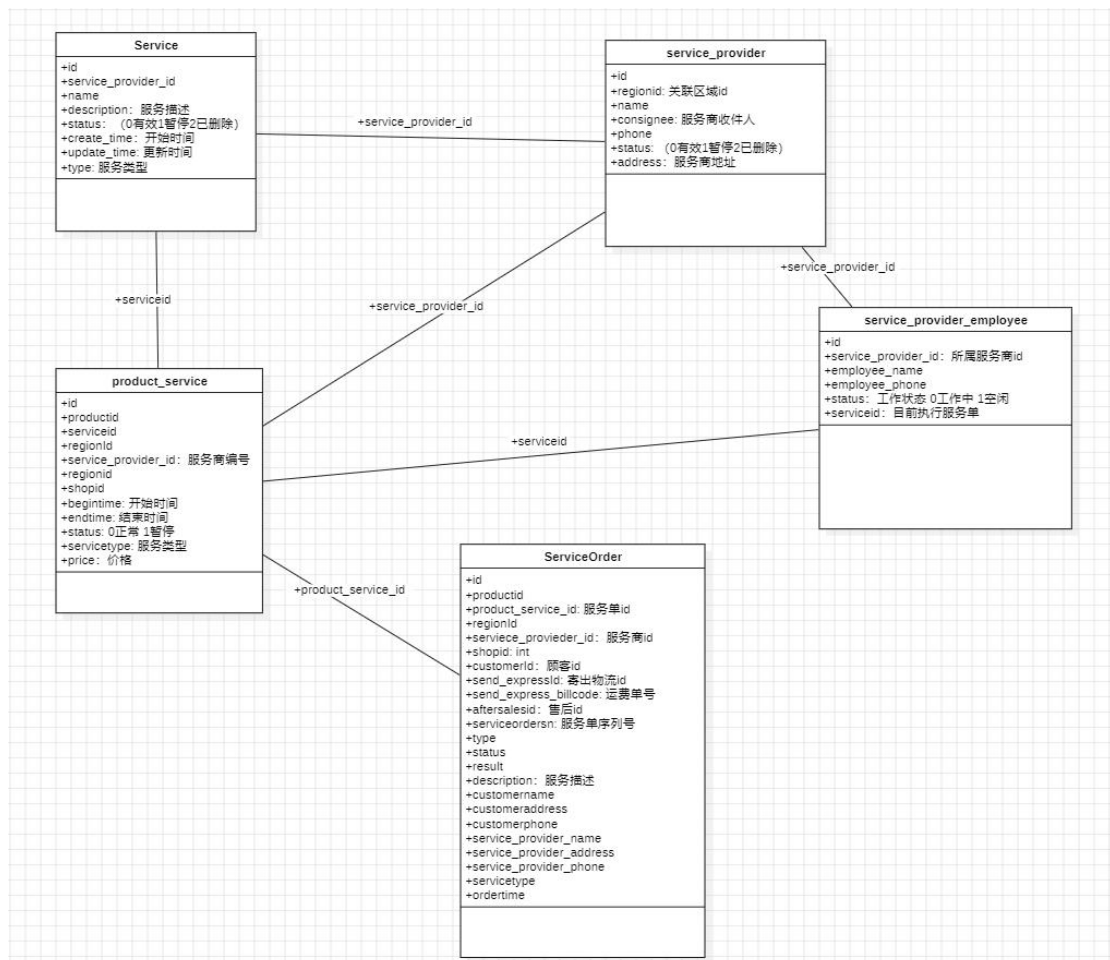


二、服务模块

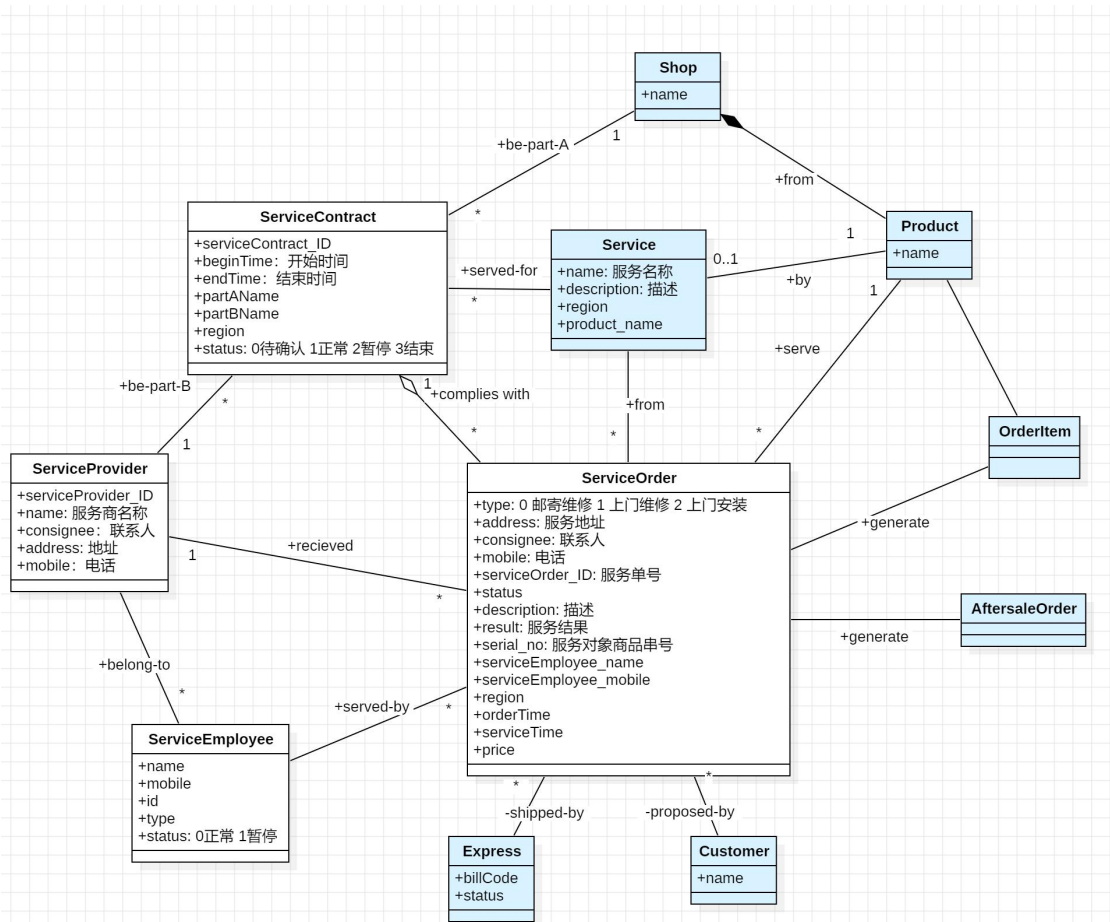
(一) 包图



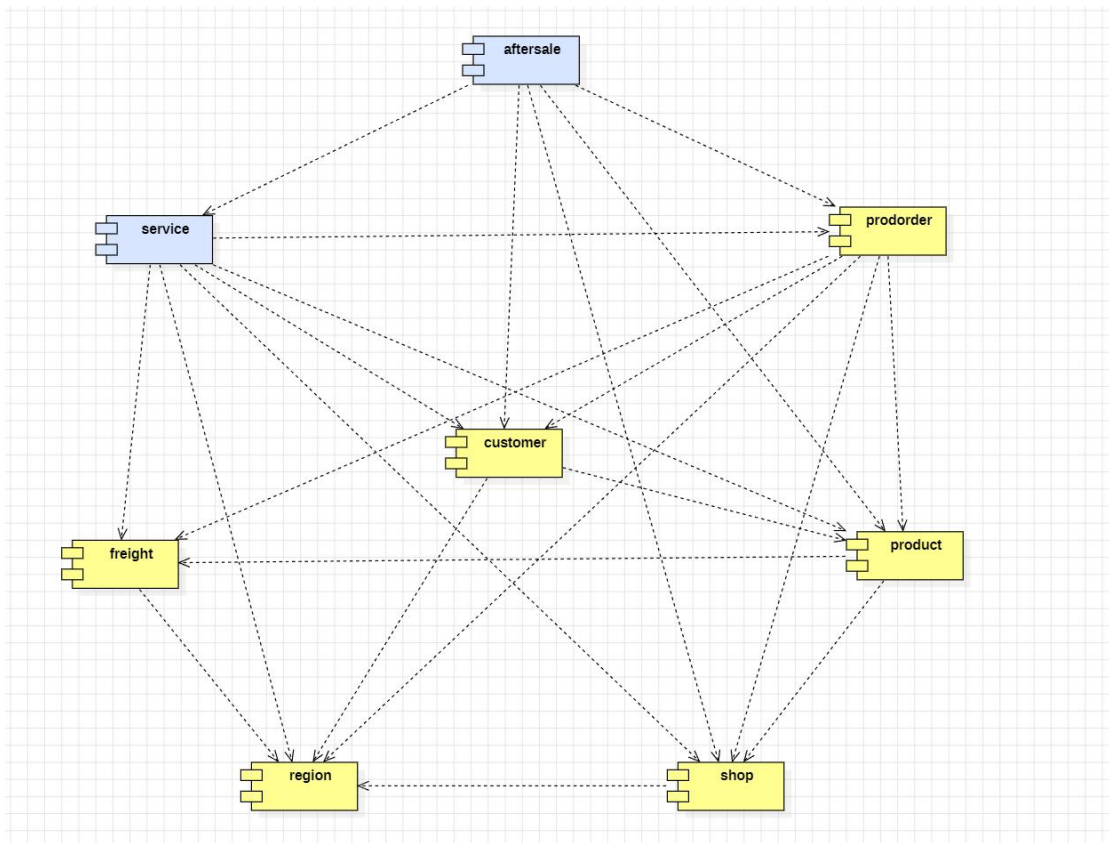
(二) 数据库设计



(三) 对象模型

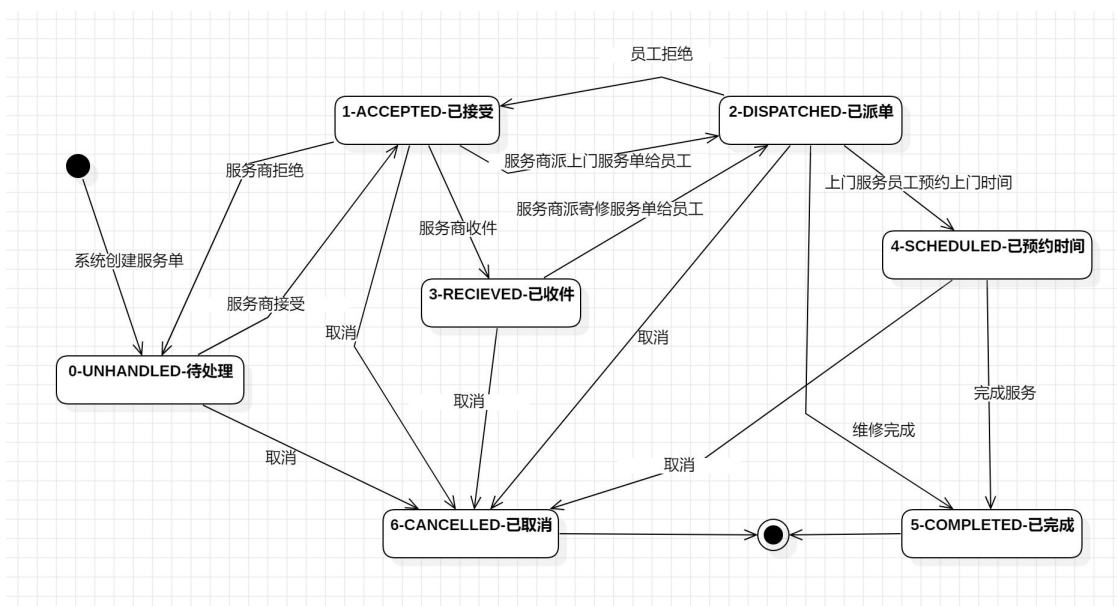


(四) 组件设计

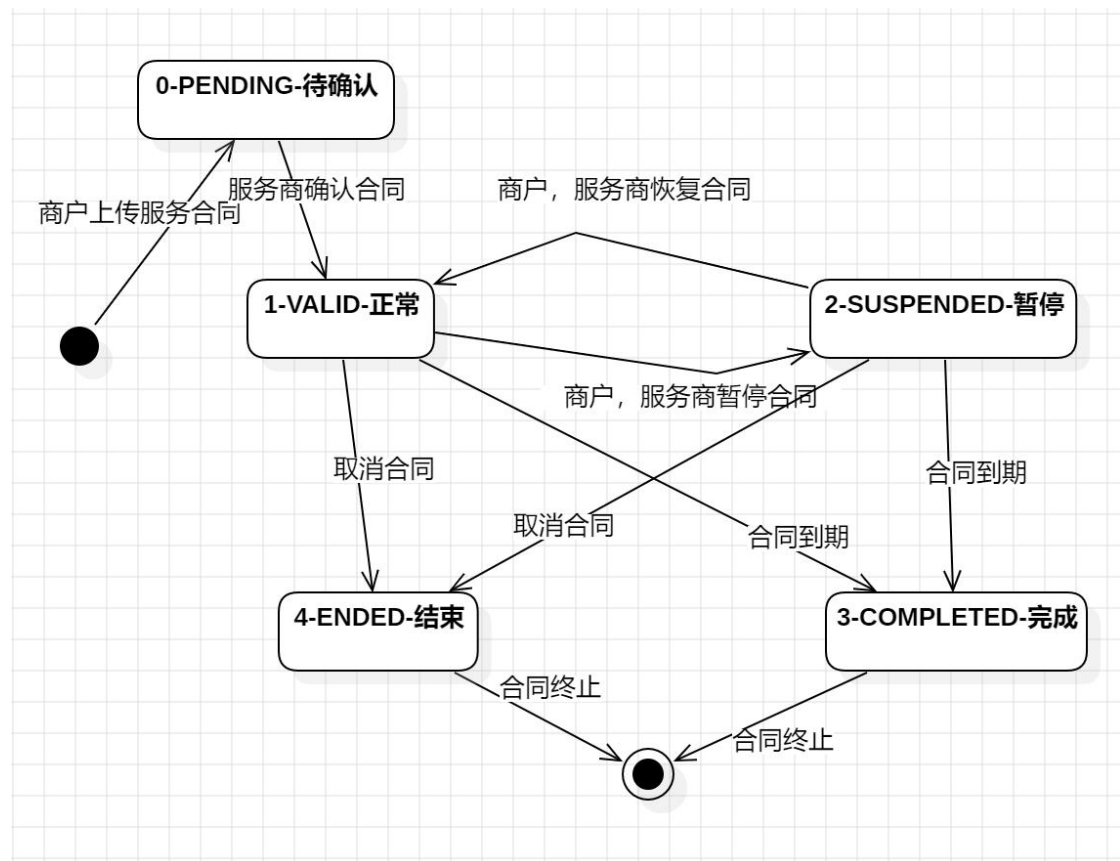


(五) 状态图

服务单状态机



服务合同状态机



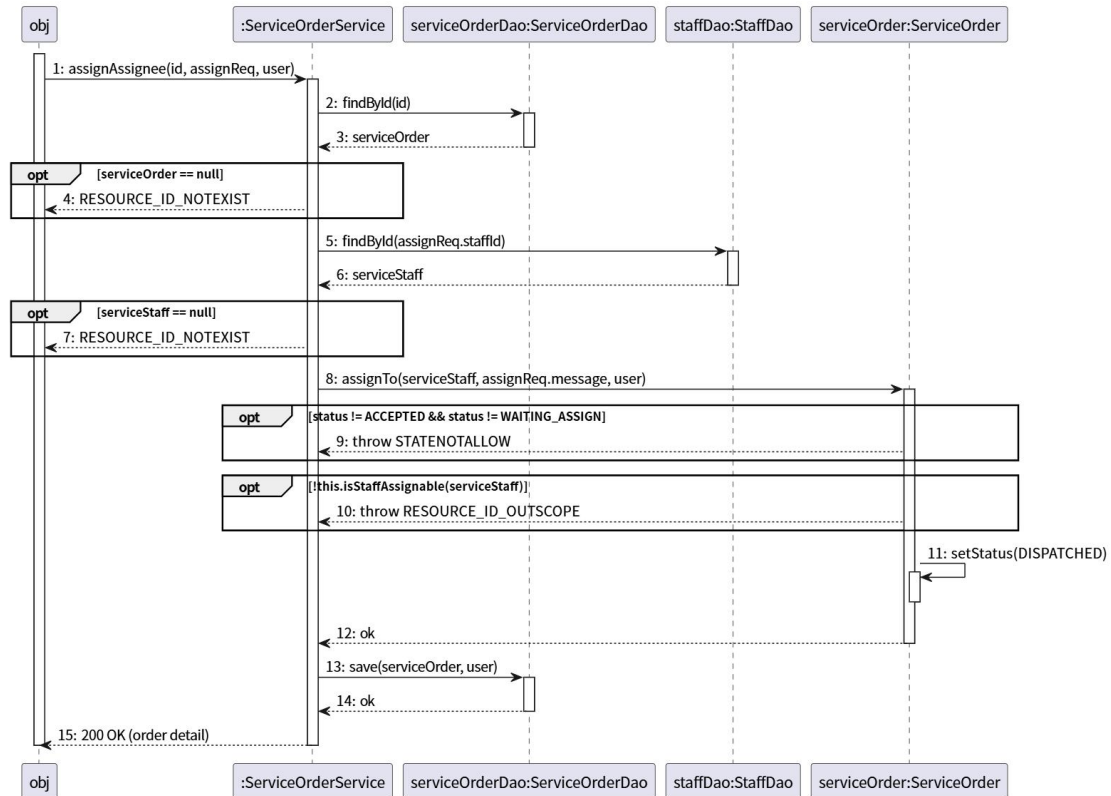
(六) API与程序逻辑（时序图）

1. 服务商指派服务单对应处理员工

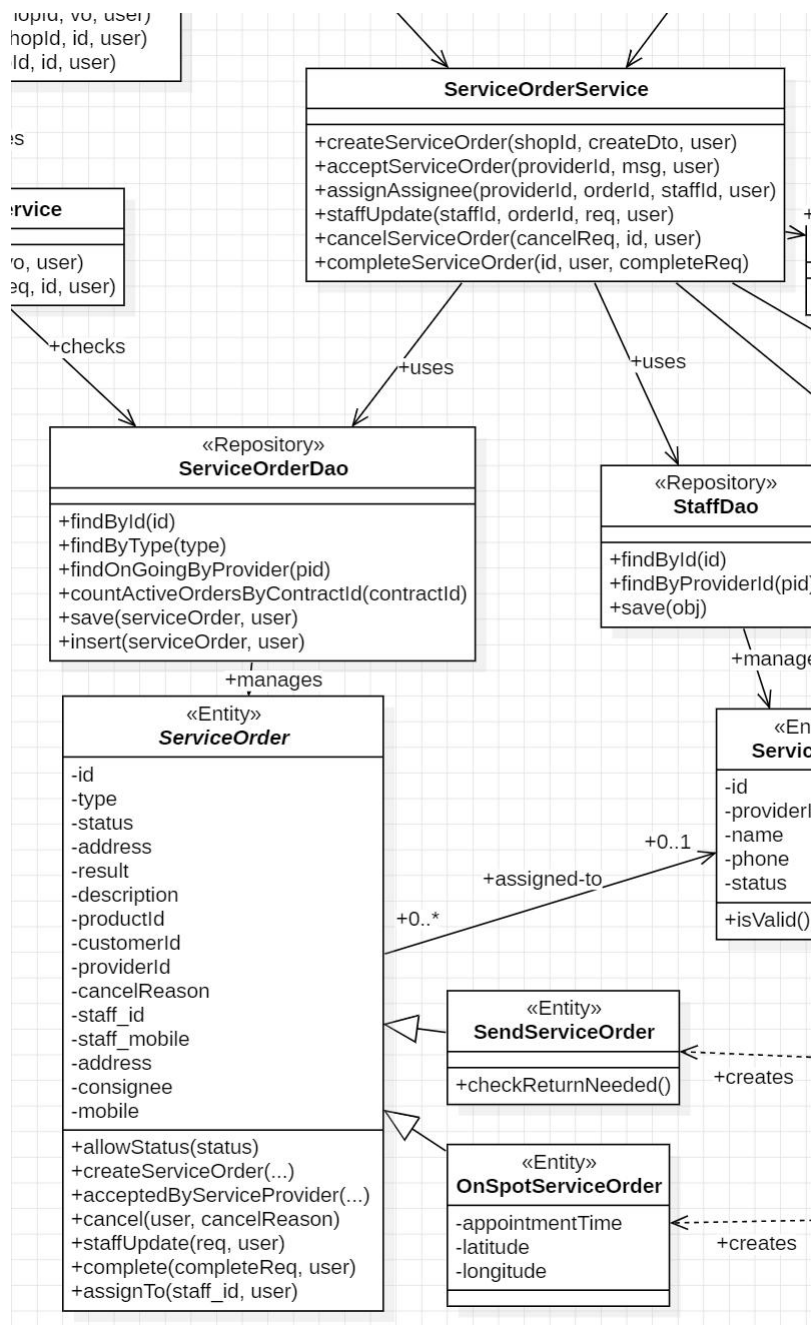
(1) API

URL	API
PUT /service-orders/{id}/assignee	https://s.apifox.cn/82492a9f-ddfe-481b-a37c-f4cad68ae2da/387358485e0

(2) 动态模型



(3) 静态模型



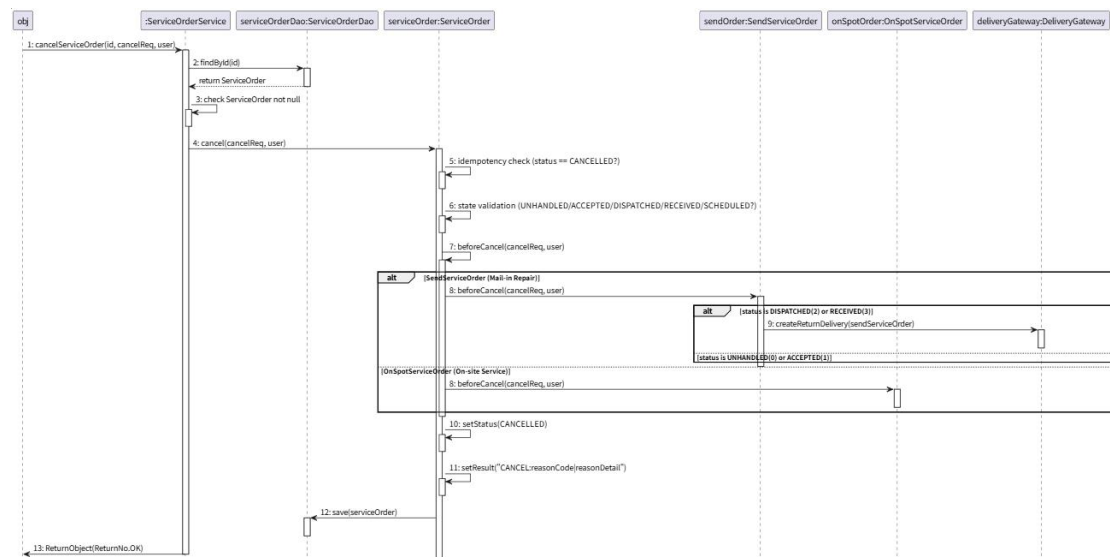
2. 取消服务单

(1) API:

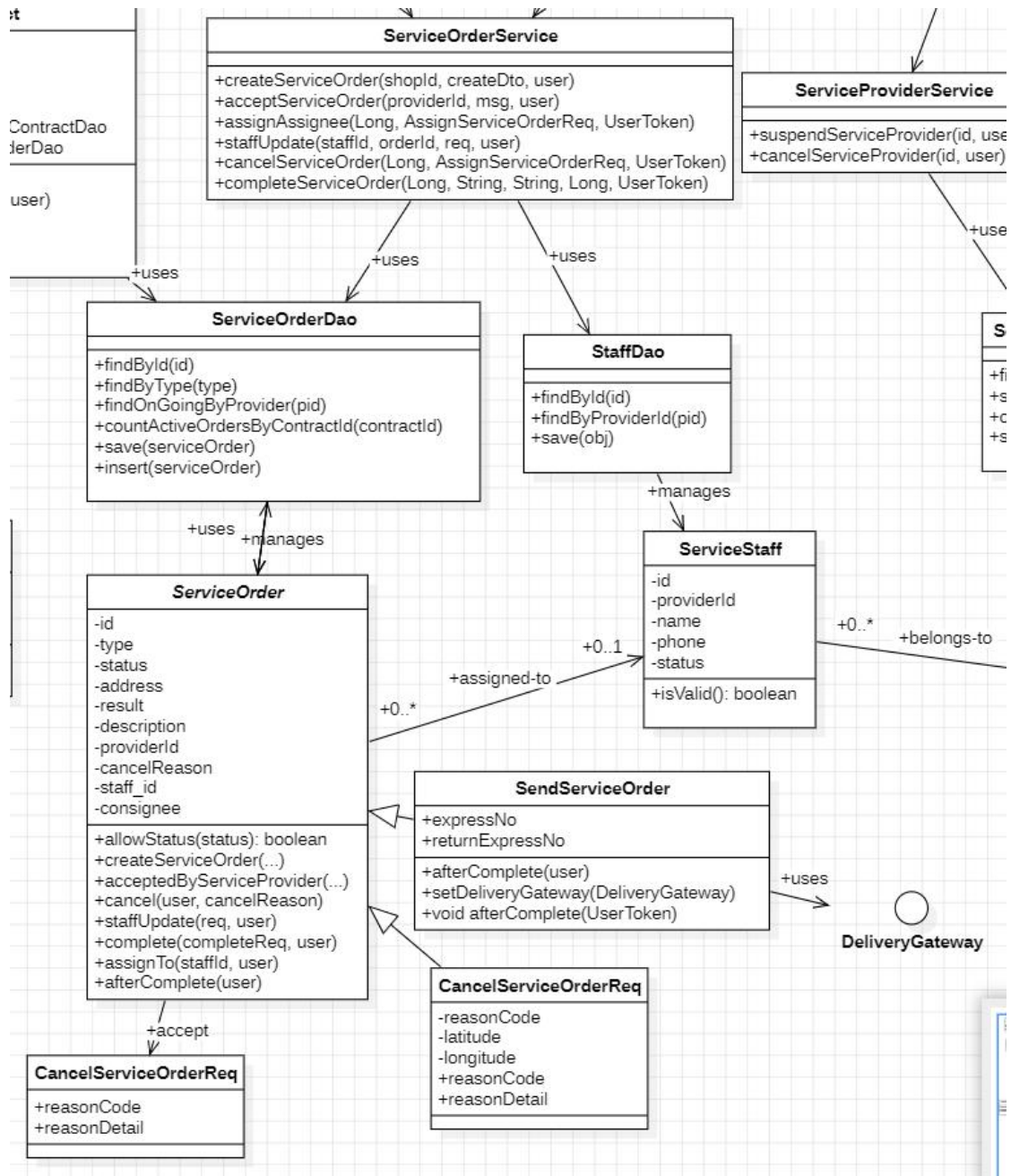
DELETE /service-orders/{service-order-id}/cancel

<https://s.apifox.cn/82492a9f-ddfe-481b-a37c-f4cad68ae2da/380656767e0>

(2) 动态模型:



(3) 静态模型:



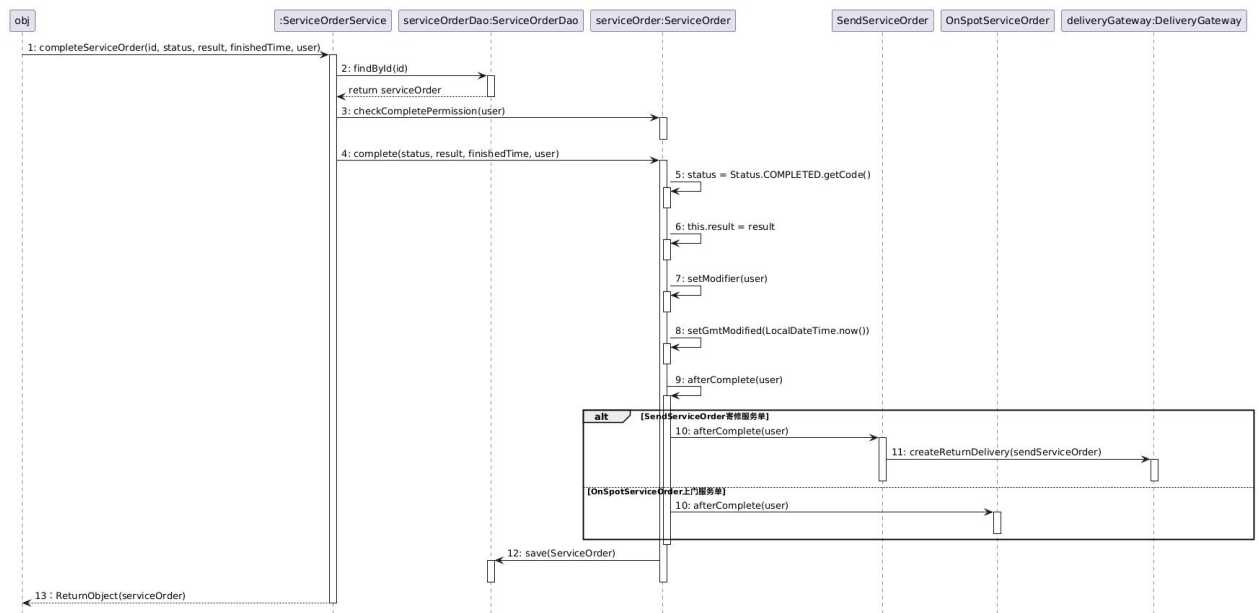
3. 完成服务单

(1) API:

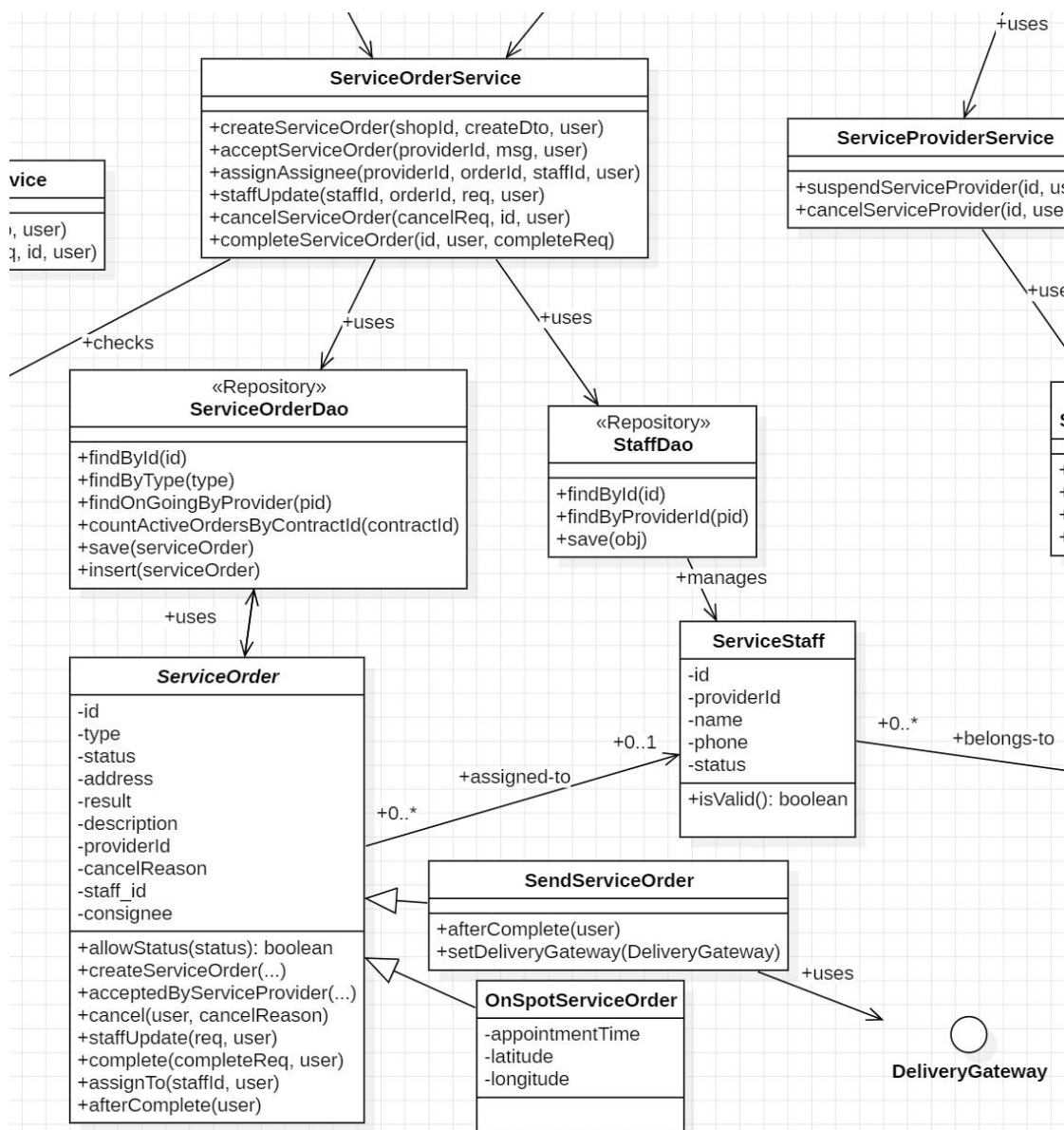
PUT /service-orders/{service-order-id}/complete

<https://s.apifox.cn/82492a9f-ddfe-481b-a37c-f4cad68ae2da/380656959e0>

(2) 时序图:



(3) 静态模型:



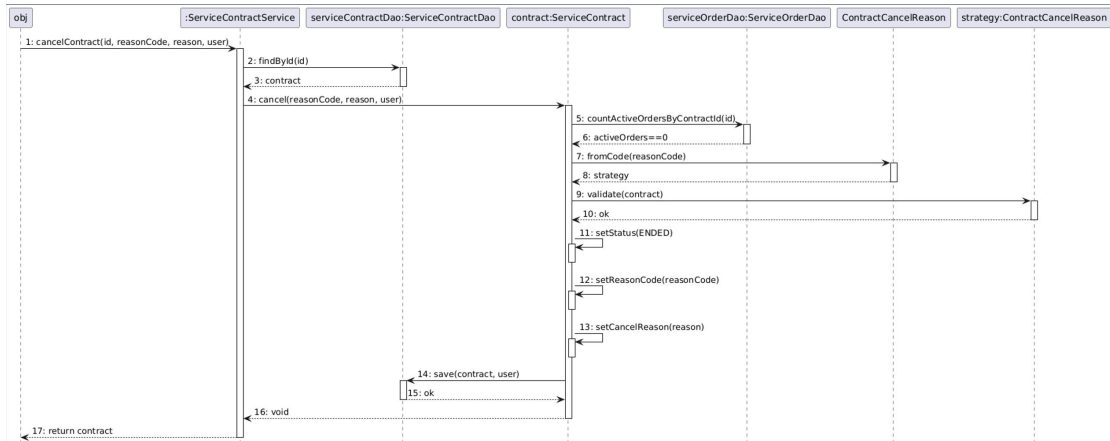
4. 取消服务合同

(1) API:

PUT /service-contracts/{id}/cancellation

<https://s.apifox.cn/82492a9f-ddfe-481b-a37c-f4cad68ae2da/380654746e0>

(2) 时序图:



(3) 静态模型:

