

商城设计文档

专 业：软件工程

年 级： **2018** 级

组长姓名**：** 李澄

组员姓名：季罗娜

组员姓名：卢嘉源

组员姓名：严峥翔

授课教师：李罡

目录

[1. 系统简介 3](#_Toc42710350)

[2. 数据库设计 3](#_Toc42710351)

[2.1 商品选购模块 3](#_Toc42710352)

[1. Product 3](#_Toc42710353)

[2. Product\_brand 4](#_Toc42710354)

[3. Product\_Category 4](#_Toc42710355)

[4. User 4](#_Toc42710356)

[5. Order 4](#_Toc42710357)

[6. Comment 4](#_Toc42710358)

[2.2 支付模块 4](#_Toc42710359)

[1. Pay 4](#_Toc42710360)

[2.3 积分模块 5](#_Toc42710361)

[1. 积分记录表 5](#_Toc42710362)

[2. 商品积分表 5](#_Toc42710363)

[3. 商品价格表 5](#_Toc42710364)

[4. 用户积分表 5](#_Toc42710365)

[2.4用户模块 5](#_Toc42710366)

[6. Restful接口设计 7](#_Toc42710367)

[6.1 商品选购模块 7](#_Toc42710368)

[6.2 订购模块 10](#_Toc42710369)

[3.3积分模块 10](#_Toc42710370)

[3.4 积分模块 12](#_Toc42710371)

[4 包设计 13](#_Toc42710372)

[4.4 商品选购模块 13](#_Toc42710373)

[4.5 订购模块 14](#_Toc42710374)

[4.6 积分模块 15](#_Toc42710375)

[三层各自的类图如下所示： 16](#_Toc42710376)

[每个类的功能介绍： 18](#_Toc42710377)

[关于使用到的设计模式 19](#_Toc42710378)

[5 接口设计 19](#_Toc42710379)

[5.4 商品展示、增加、删除、修改 19](#_Toc42710380)

[Bean 19](#_Toc42710381)

[Repository 19](#_Toc42710382)

[Controller 20](#_Toc42710383)

[Exception 20](#_Toc42710384)

[5.2品牌展示、增加、删除、修改 20](#_Toc42710385)

[Bean 20](#_Toc42710386)

[Repository 21](#_Toc42710387)

[Controller 21](#_Toc42710388)

[Exception 22](#_Toc42710389)

[5.3类别展示、增加、删除、修改 22](#_Toc42710390)

[Bean 22](#_Toc42710391)

[Repository 22](#_Toc42710392)

[Controller 23](#_Toc42710393)

[Exception 23](#_Toc42710394)

[5.4订单展示、增加、删除、修改 23](#_Toc42710395)

[Bean 23](#_Toc42710396)

[Repository 24](#_Toc42710397)

[Controller 24](#_Toc42710398)

[Exception 24](#_Toc42710399)

[5.5 订购模块 24](#_Toc42710400)

[Dao 25](#_Toc42710401)

[Domain 25](#_Toc42710402)

[Service，controller 25](#_Toc42710403)

[5.3 积分模块 26](#_Toc42710404)

[控制层和服务层之间的接口：抽象类Services 26](#_Toc42710405)

[服务层与数据库层之间的接口： 26](#_Toc42710406)

[5.4 用户模块 28](#_Toc42710407)

[5.4.1 controller与service的接口 28](#_Toc42710408)

[5.4.2 service与repository的接口 29](#_Toc42710409)

[5.4.3 Controller层、Service层与Repository层中的业务对象VO、BO和DO的设计 31](#_Toc42710410)

# 系统简介

本项目致力于打造一个商城系统，它集成了四个子模块商品选购模块、用户管理模块、客服支付模块、积分管理模块。其中，商品选购模块负责商品的展示、订单的生成；用户管理模块负责用户信息的管理以及用户鉴权功能；客服支付模块负责调用qq系统进行客服咨询以及调用支付系统进行支付；积分管理模块负责用户积分的管理以及积分的获取、兑换等服务。

# 数据库设计

## 商品选购模块

### Product

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| Id | Int(自增) | Not null | PK ，FK |
| Product\_id | Int | Not null | PK |
| name | Char(20) | Not null |  |
| keywords | Char(20） | Not null |  |
| description | Char(20) | Not null |  |
| icon | Char(2) | Not null |  |
| Category\_id | Char(19) | Not null |  |
| Brand\_id | Char(19) | Not null |  |

### Product\_brand

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| Id | Int(自增) | Not null | PK ，FK |
| brand\_id | Int | Not null | PK |
| name | Char(20) | Not null |  |
| story | Char(20） | Not null |  |
| picture | Char(20) | Not null |  |

### Product\_Category

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| Id | Int(自增) | Not null | PK ，FK |
| category\_id | Int | Not null | PK |
| name | Char(20) | Not null |  |
| level | int | Not null |  |
| icon | Char(20) | Not null |  |

### User

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| Id | Int(自增) | Not null | PK ，FK |
| user\_id | Int | Not null | PK |
| Username | Char(20) | Not null |  |
| password | Char(20) | Not null |  |
| phone | Char(20) | Not null |  |
| address | Char(20) | Not null |  |

### Order

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| Id | Int(自增) | Not null | PK ，FK |
| order\_id | Int | Not null | PK |
| Products\_id | Char(20) | Not null |  |
| price | Char(20) | Not null |  |
| Create\_time | Char(20) | Not null |  |

### Comment

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| Id | Int(自增) | Not null | PK ，FK |
| comment\_id | Int | Not null | PK |
| user\_id | Int | Not null |  |
| Product\_id | Int | Not null |  |
| content | Char(20) | Not null |  |
| Create\_time | Char(20) | Not null |  |

## 支付模块

### Pay

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| Sid | Int | Not null | PK ，FK |
| dateString | Int | Not null | PK |
| goodsid | Int | Not null |  |

## 积分模块

### 积分记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 记录id | Int 自增 | 非空 | PK |
| 用户id | Int（64） | 非空 | FK |
| 积分变化数值 | Int | 非空 | Null |
| 记录时间 | Date | 非空 | Null |

### 商品积分表

|  |  |  |  |
| --- | --- | --- | --- |
| 商品id | Int (64) | 非空 | PK，FK |
| 商品对应的积分 | Int | 非空 | Null |

### 商品价格表

|  |  |  |  |
| --- | --- | --- | --- |
| 商品id | Int（64） | 非空 | PK，FK |
| 商品价格 | Int | 非空 | Null |

### 用户积分表

|  |  |  |  |
| --- | --- | --- | --- |
| 用户id | Int（64） | 非空 | PK，FK |
| 用户总积分(初始化为1000，与RMB兑换比为100:1) | Int | 非空 | Null |

## 2.4用户模块

1. 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| 用户表id | Int | Not null | PK |
| 用户名 | Char(50) | Not null | PK、FK |
| 用户密码 | Char(50) | Not null |  |
| 电子邮箱 | Char(50) | Not null |  |
| 电话 | Char(20) | Not null |  |
| 找回密码问题 | Char(100) |  |  |
| 找回密码答案 | Char(100) |  |  |
| 角色 | Int | Not null |  |
| 创建时间 | Datetime | Not null |  |
| 最后一次更新时间 | Datetime | Not null |  |

2.用户收货地址表

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| 购物id | Int（自增） | Not null | PK |
| 用户id | Int | Not null |  |
| 收货姓名 | Char(20) | Not null |  |
| 收货固定电话 | Char(20) |  |  |
| 收货移动电话 | Char(20) | Not null |  |
| 省份 | Char(20) | Not null |  |
| 城市 | Char(20) | Not null |  |
| 区/县 | Char(20) | Not null |  |
| 详细地址 | Char(200) | Not null |  |
| 邮编 | Char(6) |  |  |
| 创建时间 | Datetime |  |  |
| 最后一次更新时间 | Datetime |  |  |

1. 评论主表

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| 评论id | Int | Not null | PK |
| 评论类型 | Int | Not null |  |
| 被评论者id | Char(32) | Not null | FK |
| 评论者id | Char(32) | Not null |  |
| 评论者名字 | Char(32) | Not null |  |
| 评论者头像 | Char(512) |  |  |
| 点赞数量 | Int |  |  |
| 评论内容 | Char(512) |  |  |
| 创建时间 | Datatime | Not null |  |
| 评论标识 | Char(255) | Not null |  |

1. 评论回复表

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| 评论标识 | Char(255) | Not null |  |
| 评论id | Int | Not null | PK |
| 评论者id | Char(32) | Not null |  |
| 评论者名字 | Char(32) | Not null |  |
| 评论者头像 | Char(512) |  |  |
| 评论者id | Char(32) | Not null |  |
| 评论者名字 | Char(32) | Not null |  |
| 评论者头像 | Char(512) |  |  |
| 点赞数量 | Int |  |  |
| 评论内容 | Char(512) |  |  |
| 创建时间 | Datatime | Not null |  |

1. 点赞表

|  |  |  |  |
| --- | --- | --- | --- |
| 说明 | 类型 | 字段约束 | 字段类型 |
| 点赞id | Int | Not null | PK |
| 对应对象id | Char(255) | Not null |  |
| 点赞用户id | Char(255) | Not null |  |
| 点赞状态 | Int | Not null |  |

# Restful接口设计

## 商品选购模块

1. 商品显示

Get/product/get/ {

“id”:”1234”

}

🡪

{

“id”:”1234”

“productId”: “1234”

“name”: “coat”

“description”:”very good!”

“keywords”:””

“icon”:”qwe”

“categoryId”:”1235”

“brandId”:”12367”

}

1. 增加商品

Post/product/add {

“productId”: “1234”

“name”: “coat”

“description”:”very good!”

“keywords”:””

“icon”:”qwe”

“categoryId”:”1235”

“brandId”:”12367”

}

🡪

“saved”

1. 修改商品

Put/product/put/{id} {

“productId”: “1234”

“name”: “coat”

“description”:”very good!”

“keywords”:””

“icon”:”qwe”

“categoryId”:”1235”

“brandId”:”12367”

}

🡪

{

“id”:”1234”

“productId”: “1234”

“name”: “coat”

“description”:”very good!”

“keywords”:””

“icon”:”qwe”

“categoryId”:”1235”

“brandId”:”12367”

}

1. 删除商品

Delete/product/delete/{id} {

}

🡪

1. 品牌显示

Get/brand/get/ {

“id”:”1234”

}

🡪

{

“id”:”1234”

“brandId”: “1234”

“name”: “coat”

“story”:”very good!”

“picture”:”qqw”

}

1. 增加品牌

Post/brand/add {

“brandId”: “1234”

“name”: “coat”

“story”:”very good!”

“picture”:”qqw”

}

🡪

“saved”

1. 修改品牌

Put/brand/put/{id} {

“brandId”: “1234”

“name”: “coat”

“story”:”very good!”

“picture”:”qqw”

}

🡪

{

“id”:”1234”

“brandId”: “1234”

“name”: “coat”

“story”:”very good!”

“picture”:”qqw”

}

1. 删除品牌

Delete/brand/delete/{id} {

}

🡪

1. 商品类别显示

Get/category/get/ {

“id”:”1234”

}

🡪

{

“id”:”1234”

“categoryId”: “1234”

“name”: “coat”

“level”:”2“

“icon”:”qqw”

}

1. 增加商品类别

Post/category/add {

“categoryId”: “1234”

“name”: “coat”

“level”:”2“

“icon”:”qqw”

}

🡪

“saved”

1. 修改商品类别

Put/category/put/{id} {

“categoryId”: “1234”

“name”: “coat”

“level”:”2“

“icon”:”qqw”

}

🡪

{

“id”:”1234”

“categoryId”: “1234”

“name”: “coat”

“level”:”2“

“icon”:”qqw”

}

1. 删除商品类别

Delete/category/delete/{id} {

}

🡪

1. 订单显示

Get/order/get/ {

“id”:”1234”

}

🡪

{

“id”:”1234”

“orderId”: “1234”

“productsId”: “1235,1236”

“price”:”20.00“

“createTime”:”2020.02.28”

}

1. 增加订单

Post/order/add {

“orderId”: “1234”

“productsId”: “1235,1236”

“price”:”20.00“

“createTime”:”2020.02.28”

}🡪

“saved”

1. 修改订单

Put/order/put/{id} {

“orderId”: “1234”

“productsId”: “1235,1236”

“price”:”20.00“

“createTime”:”2020.02.28”

}

🡪

{

“id”:”1234”

“orderId”: “1234”

“productsId”: “1235,1236”

“price”:”20.00“

“createTime”:”2020.02.28”

}

1. 删除订单

Delete/order/delete/{id} {

}

🡪

## 订购模块

1. 显示支付的商品

Get/pay/paygoods.do {

“goodsId”: “123”

}

🡪

{

“paylist”: {

“sid”:”123”

“dataString”: “q123”

“glist”: {

“goodsid”:”2345”

“price”:”20.00”

“goodsname”:”coat”

}

}

}

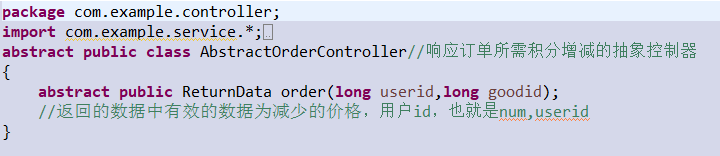
## 3.3积分模块

当客户生成订单时获取使用积分抵扣后的价格

/order?goodid=xxxx and userid=xxxx

接收的数据类型：{“goodid”:””,”userid”:”“}

对应Controller:





返回数据类型：

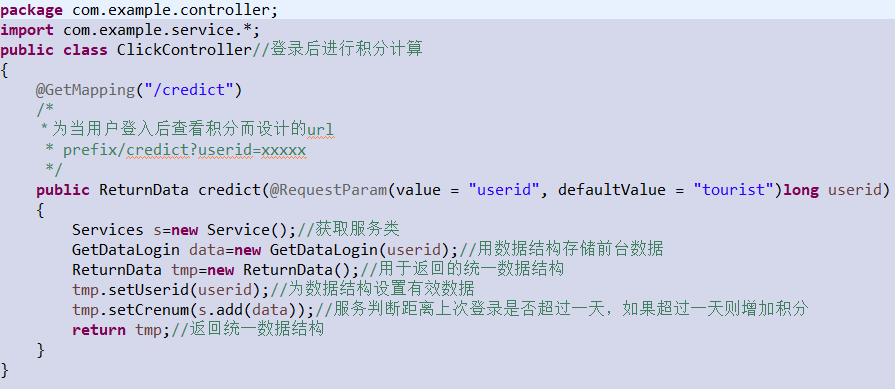
{“userid”:””,”crenum”:””,”num”:””,”time”:””}，其中userid和num为有效字段

当客户登录系统时查看总积分

/credict?userid=xxxx

接收的数据类型:{“userid”:””}

对应Controller:



返回数据类型: {“userid”:””,”crenum”:””,”num”:””,”time”:””}，其中userid和crenum为有效字段

客户查看积分记录

/credict/records?userid=xxxx

接收的数据类型:{“userid”:””}

对应的Controller:



返回数据类型: {“userid”:””,”crenum”:””,”num”:””,”time”:””}，其中userid，crenum和time为有效字段

## 积分模块

1. 用户登录：

@RequestMapping(value = "login.do",method = RequestMethod.*POST*)

1. 用户退出登录

@RequestMapping(value = "logout.do",method = RequestMethod.*POST*)

1. 用户注册

@RequestMapping(value = "register.do",method = RequestMethod.*POST*)

1. 校验email和用户名是否存在

@RequestMapping(value = "check\_valid.do",method = RequestMethod.*POST*)

1. 获取登录用户信息

@RequestMapping(value = "get\_user\_info.do",method = RequestMethod.*POST*)

1. 用户忘记密码，提出相应问题

@RequestMapping(value = "forget\_get\_question.do",method = RequestMethod.*POST*)

1. 校验问题答案是否正确

@RequestMapping(value = "forget\_check\_answer.do",method = RequestMethod.*POST*)

1. 忘记密码中重置密码

@RequestMapping(value = "forget\_reset\_password.do",method = RequestMethod.*POST*)

1. 登录状态中重置密码

@RequestMapping(value = "reset\_password.do",method = RequestMethod.*POST*)

1. 更新个人用户信息

@RequestMapping(value = "update\_information.do",method = RequestMethod.*POST*)

1. 获取用户详细信息

@RequestMapping(value = "get\_information.do",method = RequestMethod.*POST*)

1. 管理员登录

@RequestMapping(value="login.do",method = RequestMethod.*POST*)

1. 用户增加收货地址

@RequestMapping(value="add.do",method = RequestMethod.*POST*)

1. 用户删除相应收货地址

@RequestMapping(value="del.do",method = RequestMethod.*DELETE*)

1. 用户更新收货地址

@RequestMapping(value="update.do",method = RequestMethod.*PUT）*

1. 用户查看收货地址

@RequestMapping(value="select.do",method = RequestMethod.*GET）*

1. 添加父评论

@RequestMapping(value="addRootComments.do",method = RequestMethod.*POST*)

1. 添加子评论

@RequestMapping(value="addSonComments.do",method = RequestMethod.*POST*)

1. 根据资源ID查询该资源的所有评论

@RequestMapping(value="getListByOwnerId",method = RequestMethod.*GET*)

1. 点赞

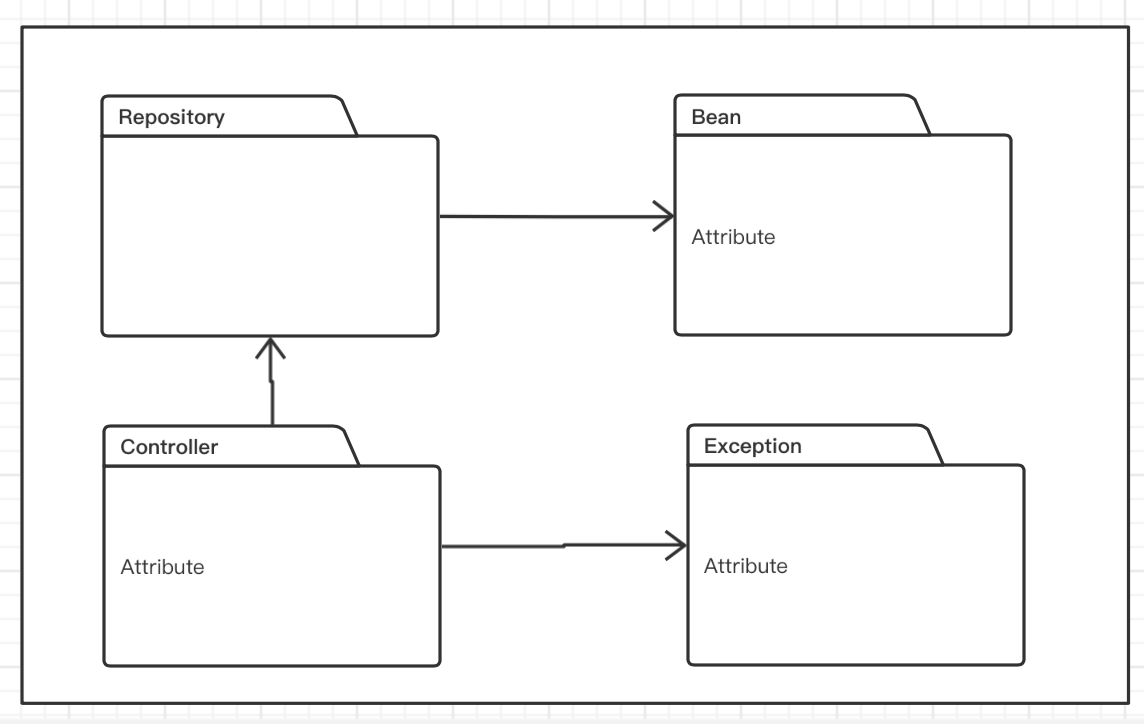
@RequestMapping(value = "isLike.do",method = RequestMethod.*POST*)

1. 查询所有的点赞信息

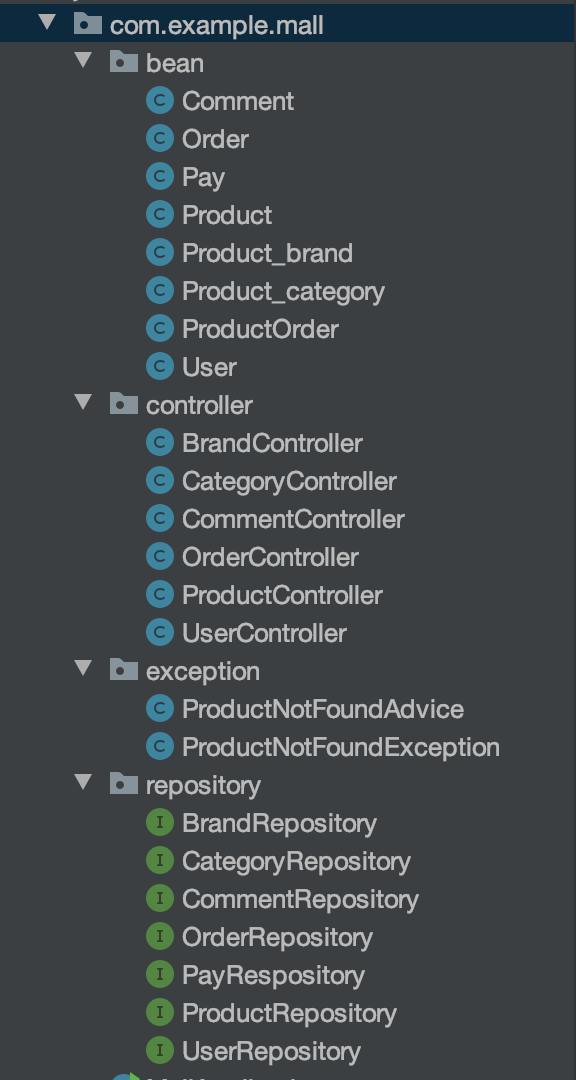
@RequestMapping("getListLikeByUserId",method = RequestMethod.*GET*)

# 包设计

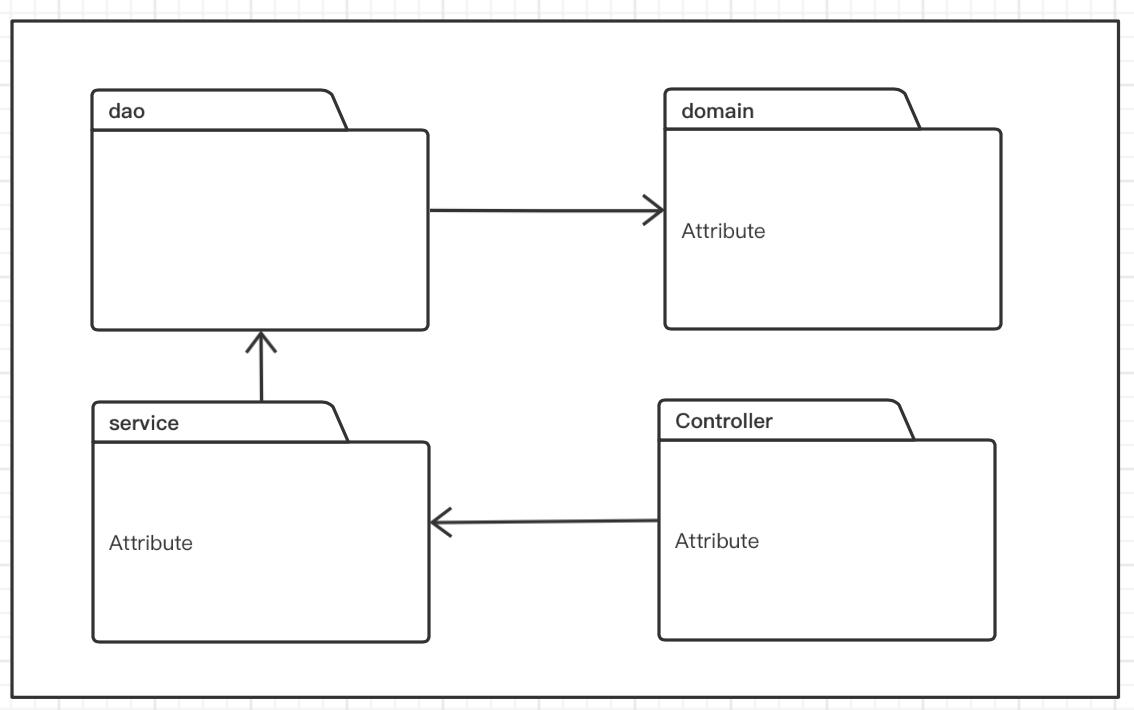
## 商品选购模块



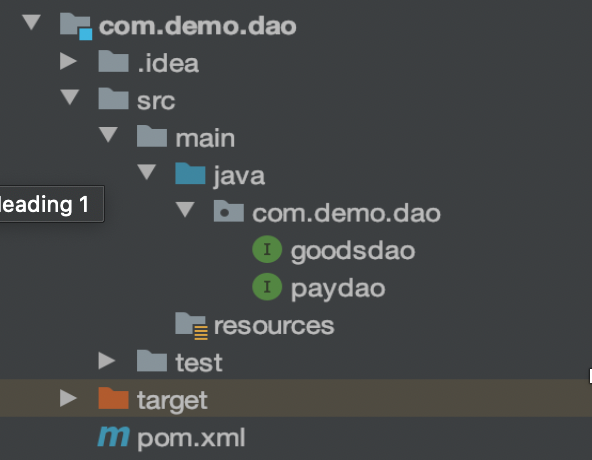
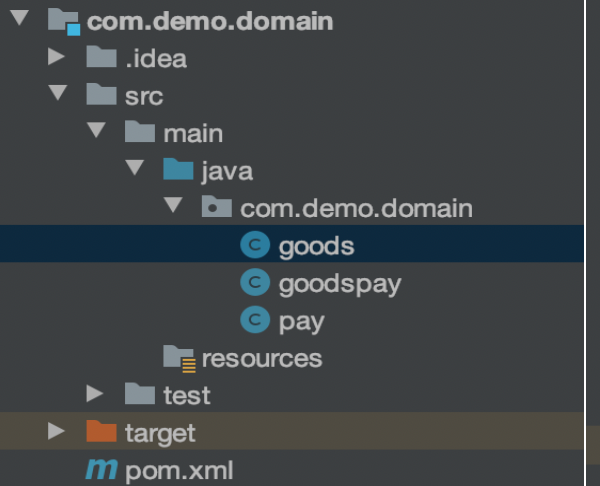
包内具体设计如下：

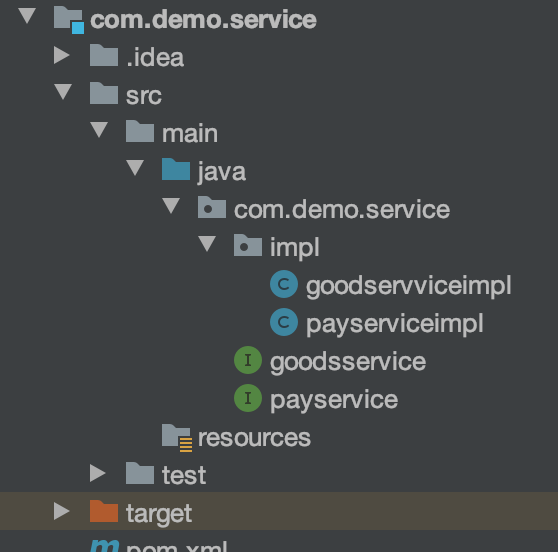
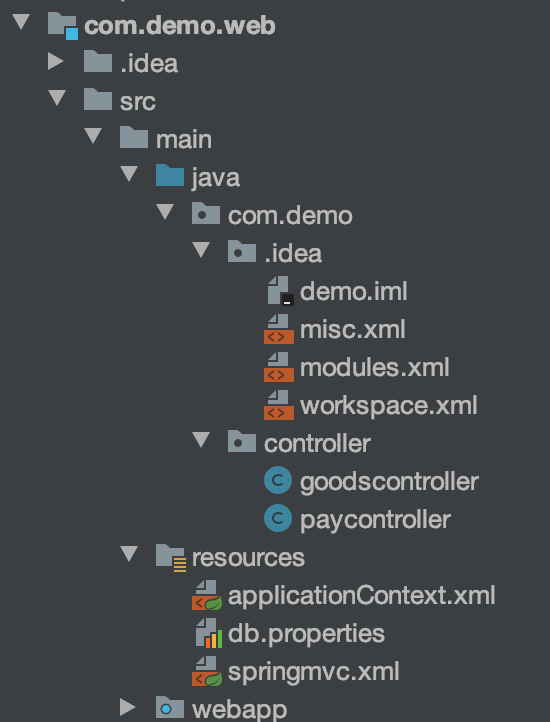


## 订购模块



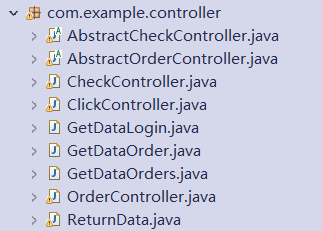
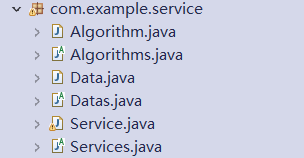
包内具体设计如下：

## 积分模块

首先是架构，整个架构分为三层，分别装载于controller,service,dao三个包中，包中的类如下：

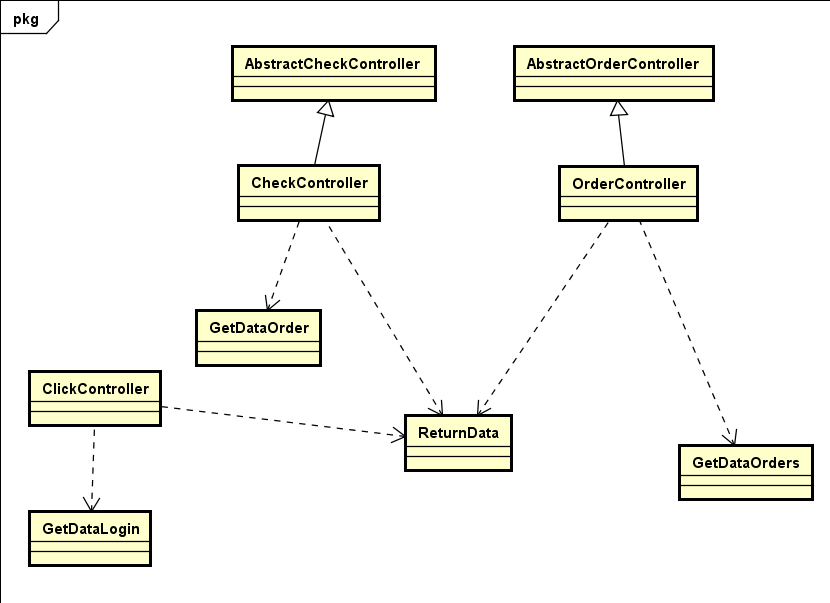
架构的最顶层是controller，其用来接收来自客户端的请求数据，并调用service中的服务来进行处理并返回结果

实际进行数据处理与判断的是service中的Service，Algorithm，其负责积分的增减以及积分变化的记录

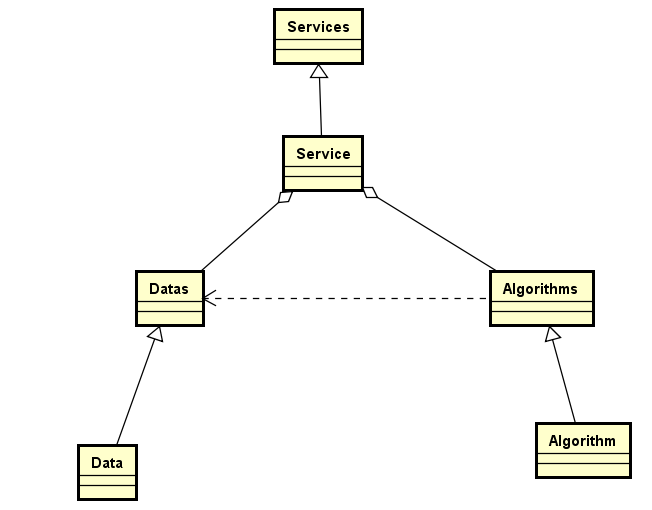
dao中的类都是用来进行与数据库进行交互的，用于从数据库中提取需要的数据以及进行数据的回写

### 三层各自的类图如下所示：

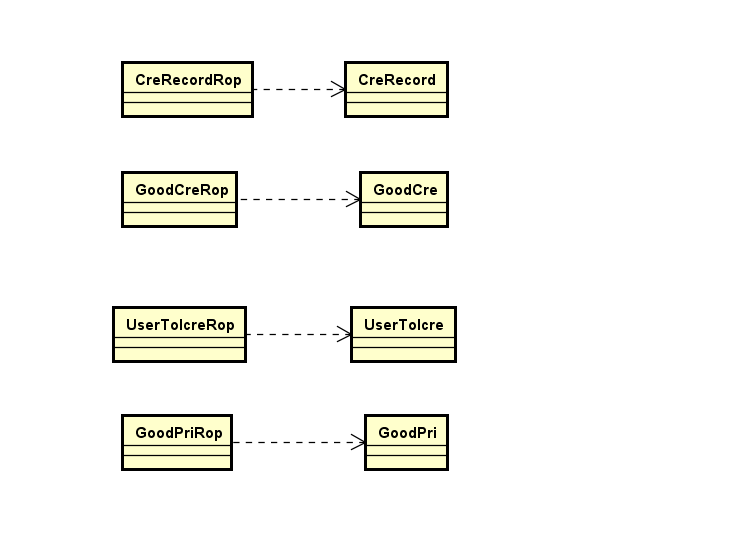
1. controller



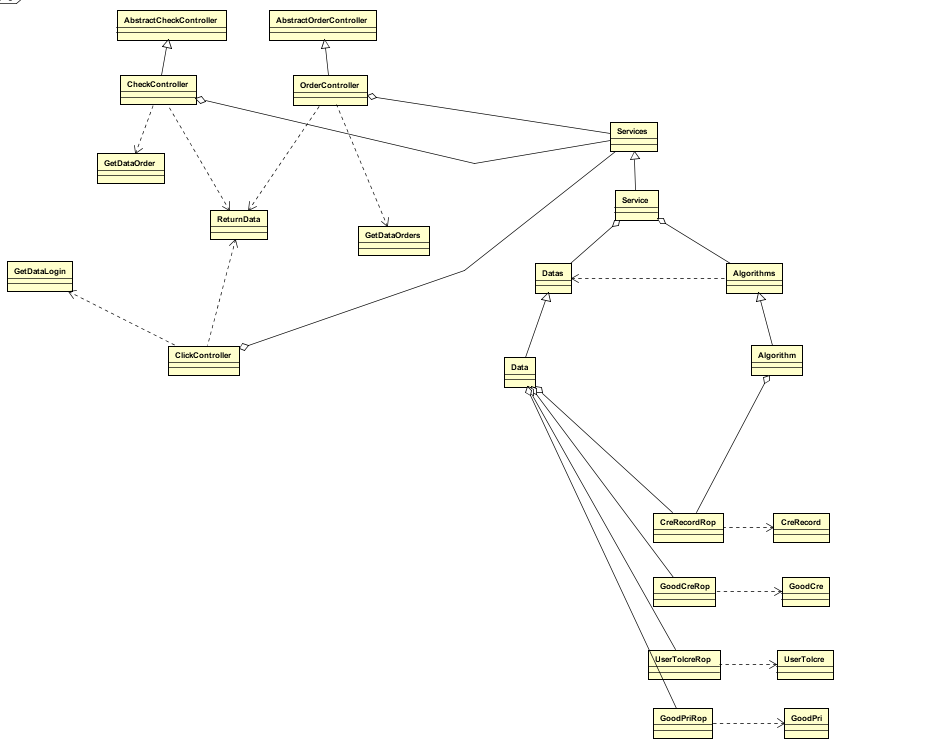
1. service



1. dao



1. 整体类图：



### 每个类的功能介绍：

1. controller:

AbstractCheckController: 用于响应前台查看积分记录的抽象控制器

AbstractOrderController//响应订单所需积分增减的抽象控制器

CheckController **extends** AbstractCheckController//查看积分记录的控制器

ClickController//登录后进行积分计算的控制器

GetDataLogin//用于存储用户登入并查看积分时前台传来的数据

GetDataOrder//用于当用户产生订单时需要使用/增加积分时前台向后台传输的数据

GetDataOrders//当客户需要查看积分明细的时候后台收到来自前台的数据

OrderController **extends** AbstractOrderController//订单产生时进行积分计算

ReturnData//统一的返回数据结构

1. service:

**public** **abstract** **class** Services//服务的抽象类

**public** **class** Service **extends** Services//一个实际的服务类

**public** **abstract** **class** Datas //存储服务所需数据的抽象类

**public** **class** Data **extends** Datas//存储服务所需数据的具体类

**public** **abstract** **class** Algorithms //提供计算积分增加和减少的算法，可以进行各种扩展，比如活动积分使用

**public** **class** Algorithm **extends** Algorithms//一个具体的积分增减计算算法类

1. dao:

**public** **class** CreRecord//这个实体中存的是记录id，用户id，用户积分增减数值，时间

**public** **interface** CreRecordRop **extends** CrudRepository<CreRecord,Long>//这个是用于上个类的数据库操作接口

**public** **class** GoodCre //这个实体中存的是商品对应增加的积分

**public** **class** GoodPri //这个实体中存的是商品对应的价格

**public** **class** UserTolcre //这个类中存的元组为{userid,tolcre}，也就是用户id和总积分

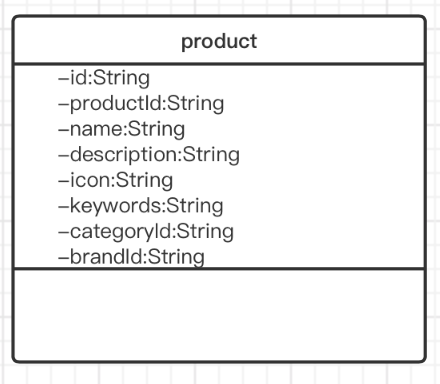
### 关于使用到的设计模式

1. 策略模式：积分计算算法的替换
2. Vistor模式：计算所需数据Data与计算算法分离
3. Bridge模式：Service中可以进行一些功能上的扩展，而抽象算法类则可以进行积分加减实现上的扩展

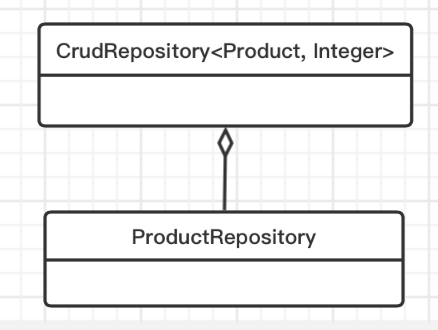
# 接口设计

## 商品展示、增加、删除、修改

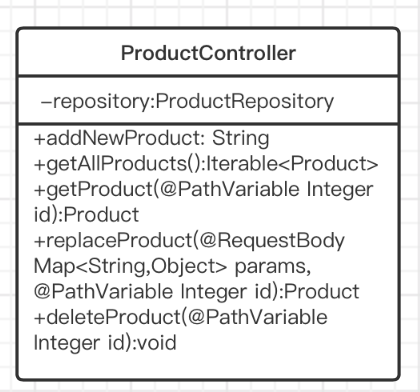
### Bean



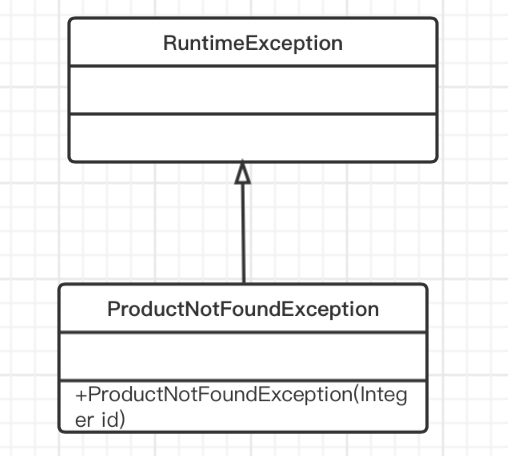
### Repository



### Controller

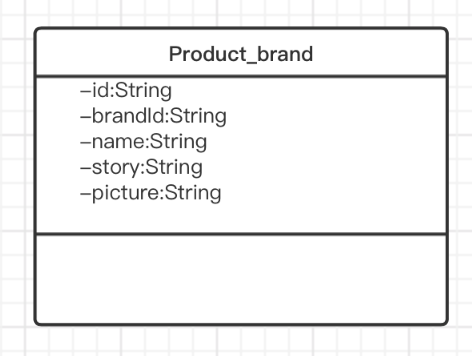


### Exception

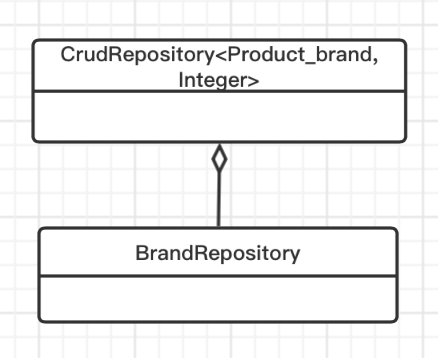


## 5.2品牌展示、增加、删除、修改

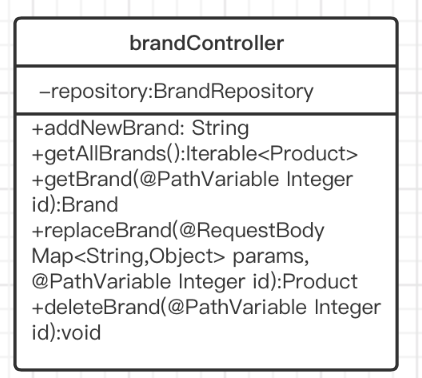
### Bean



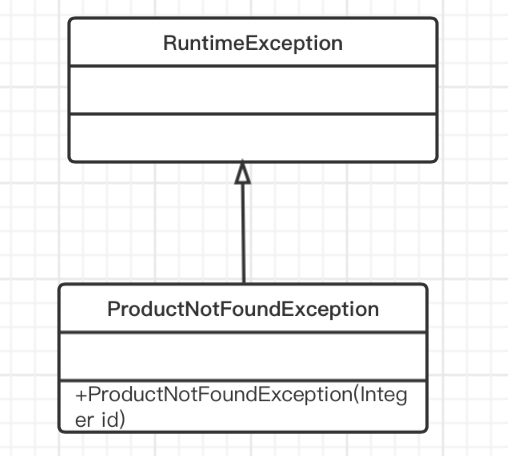
### Repository



### Controller

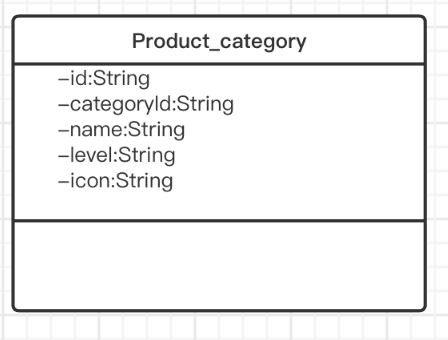


### Exception

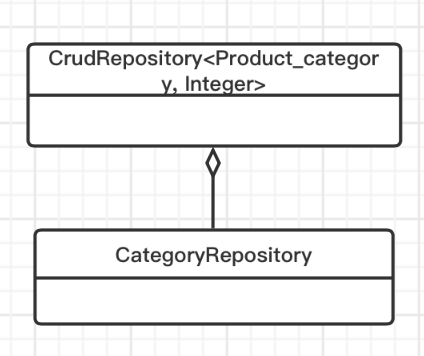


## 5.3类别展示、增加、删除、修改

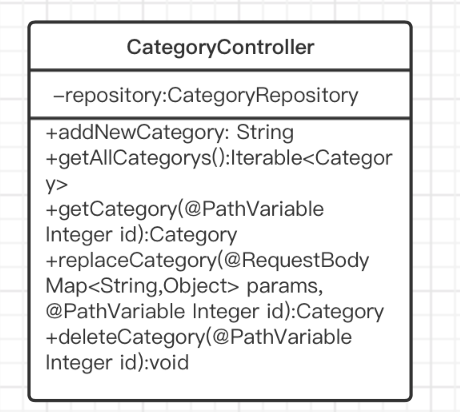
### Bean



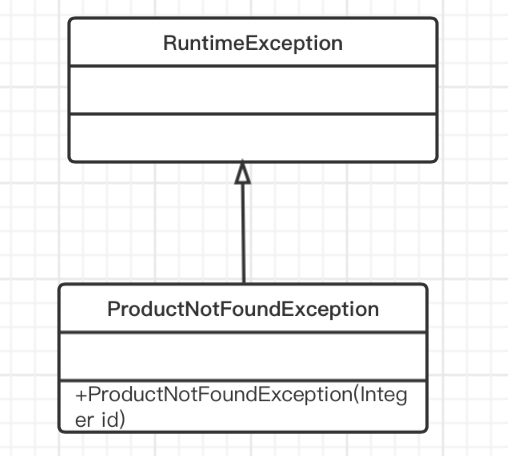
### Repository



### Controller

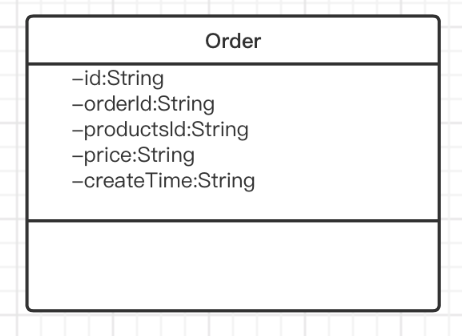


### Exception

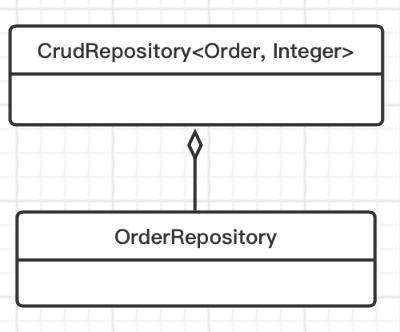


## 5.4订单展示、增加、删除、修改

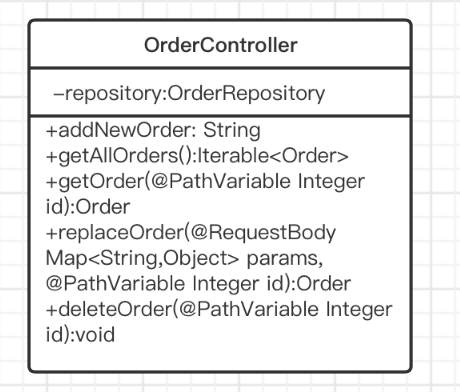
### Bean



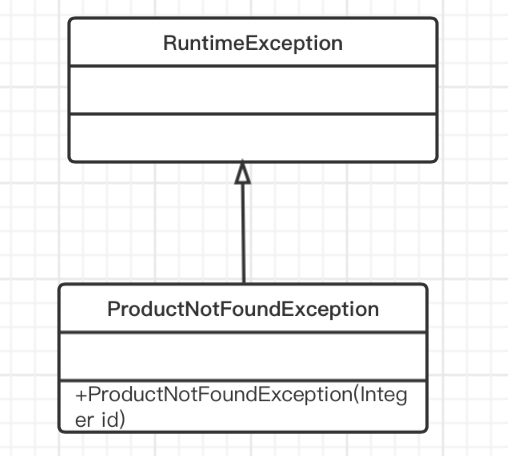
### Repository



### Controller

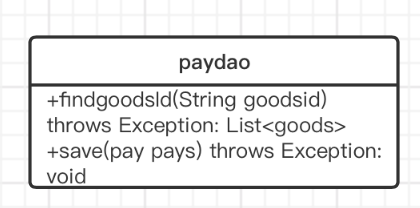
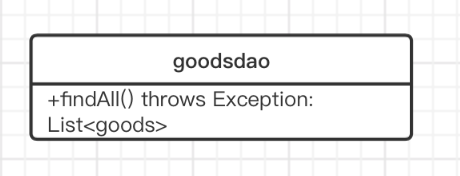


### Exception

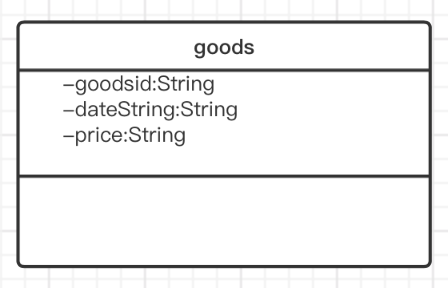
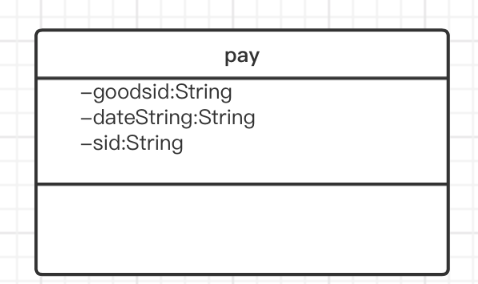


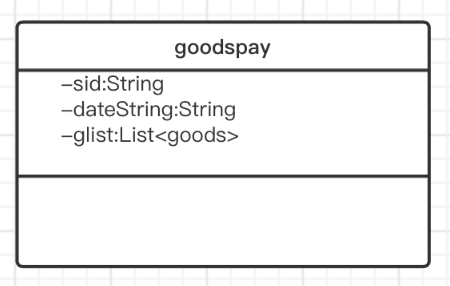
## 订购模块

### Dao

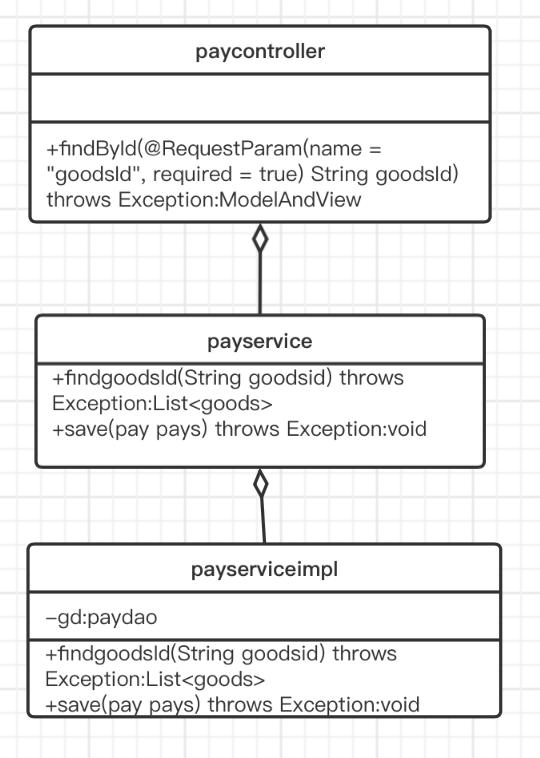
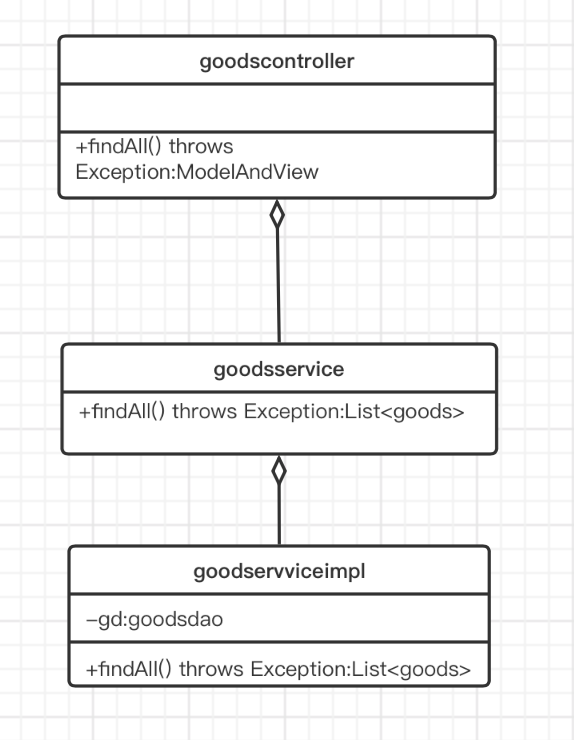
 

### Domain

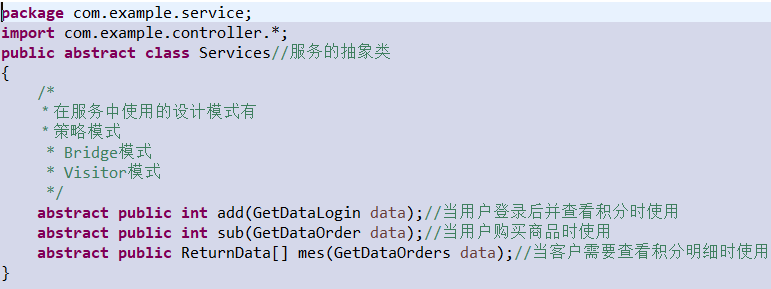


### Service，controller



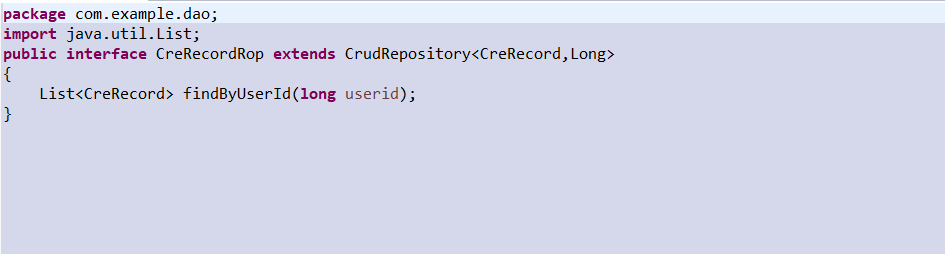
## 积分模块

### 控制层和服务层之间的接口：抽象类Services

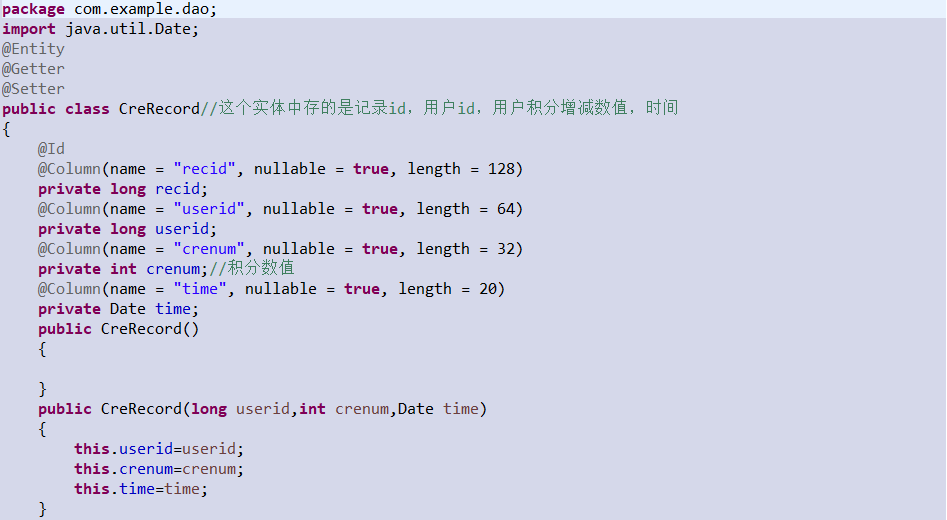


### 服务层与数据库层之间的接口：

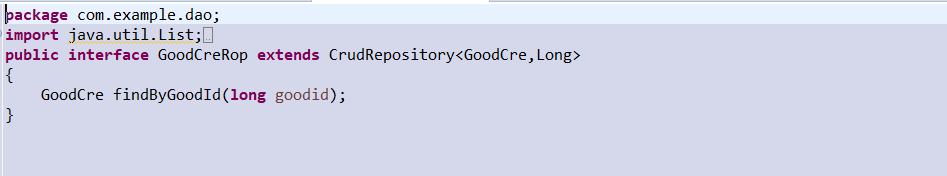
积分记录表接口



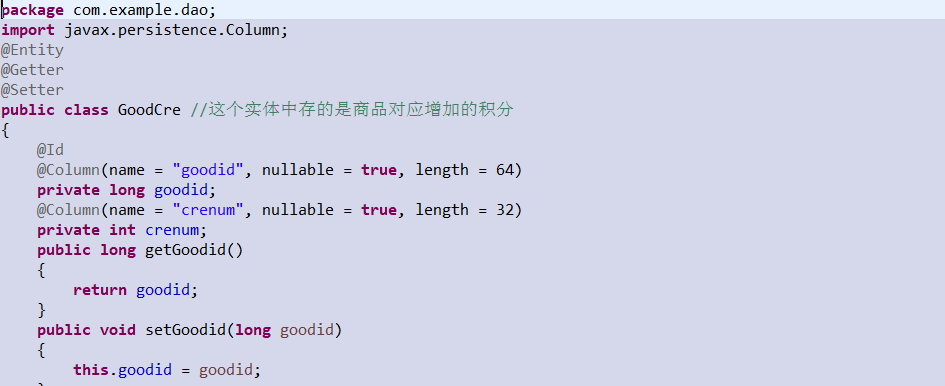
积分记录实体



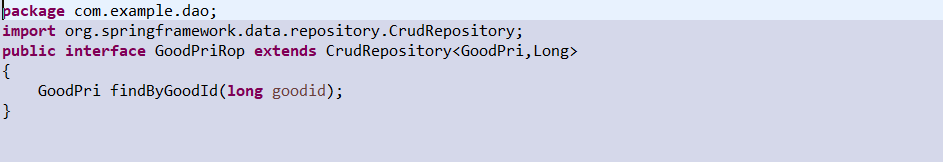
商品积分表接口



积分记录表实体



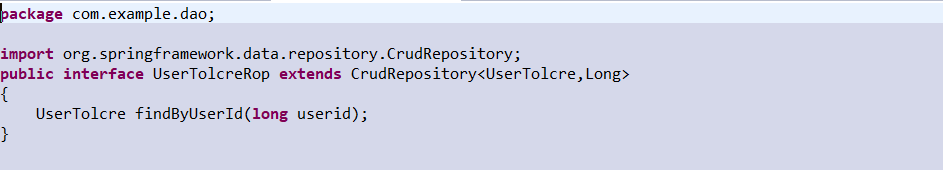
商品价格表接口



商品价格表实体



用户积分表接口



用户积分表实体



## 用户模块

分层：

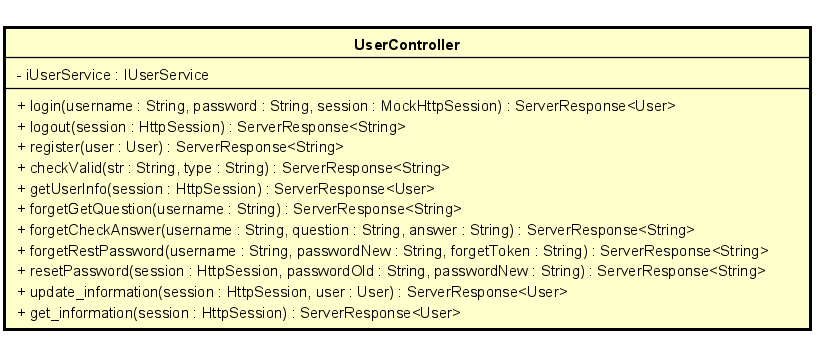
Controller： 控制与外界接口

Service： 控制逻辑操作

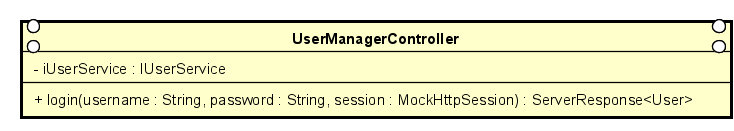
Repository： 控制与数据库的接口

### 5.4.1 controller与service的接口

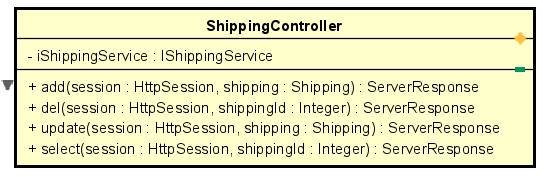
#### 1．UserController：用户基本操作服务



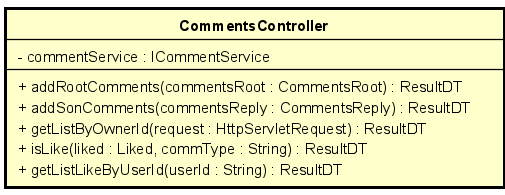
#### UserManageController：管理员登录



#### ShippingController：收货地址操作

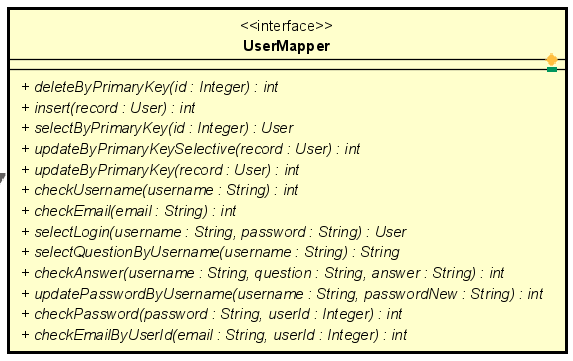


#### CommentsController：评论相关操作

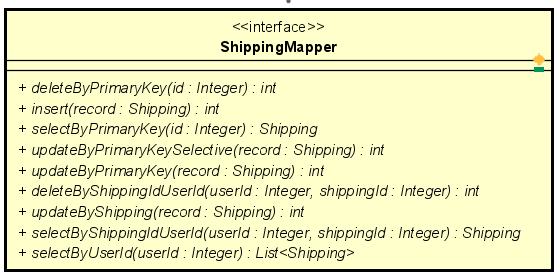


### 5.4.2 service与repository的接口

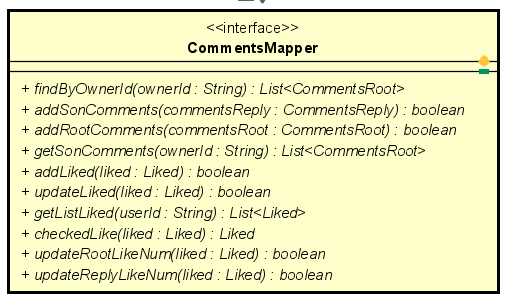
#### 1. UserMapper：用户基本服务接口



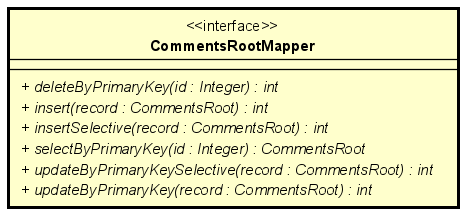
#### ShippingMapper：收货地址接口



#### CommentsMapper：基本评论公共接口



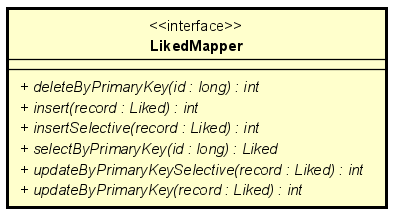
#### CommentsRootMapper：父评论接口



#### CommentsReplyMapper：评论回复接口



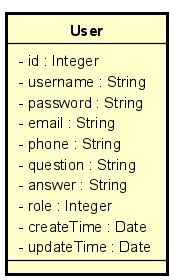
#### LikedMapper ：点赞接口



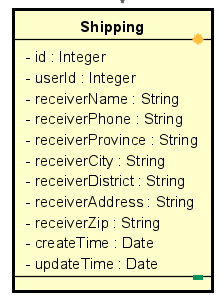
### 5.4.3 Controller层、Service层与Repository层中的业务对象VO、BO和DO的设计

此系统中对应于同一对象的VO、BO、DO属性均相同。

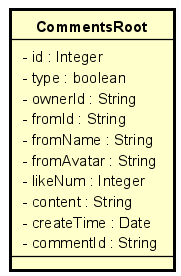
1. User:用户基本数据类



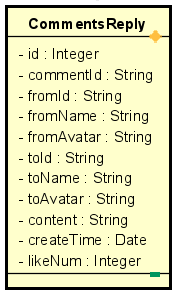
1. Shipping:收货地址数据类



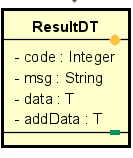
1. CommentsRoot：父评论数据类



1. CommentsReply：评论回复数据类



1. ResultDT：评论返回数据类



1. Liked：点赞数据类

