

RestFul 原则要求是：POST,GET,PUT 返回操作成功（新建的 || 获得的 || 修改状态后）的对象，delete 返回成功失败

logService 模块：

1、提供一个接口给各模块，管理员操作各模块后，调用写入日志的接口

POST /log

参数：@RequestBody Log log

Return : Log log

goodsService 模块：

1、

给分享模块，判断商品是否在售

@GetMapping("/goods/{id}/isOnSale")

参数 @PathVariable Integer goodsId

Return Boolean

2、/*12月15日早10点新增*/

扣减库存

@PUTMapping("goods/{id}/deduct")

参数 : @PathVariable Integer goodsId,@RequestParam Integer

quantity

Return :Boolean

discountService 模块:

1、 /*12.15 修改了参数, 目前的 domain.Order 中包括了 OrderItemsList 和
CouponId*/

提供一个计算优惠券优惠明细的接口。

GET /calcWithCouponPrice

参数 Order order

Return Order order

返回值中

2、 /*12.15 修改了 URL*/

Order 调用, 根据 order 获取预售价格, 把价格写进 'integral_price' 属性,

付尾款的时候再调一次

GET /calcWithPresalePrice

参数 Order order

Return Order order

paymentService 模块:

/* 2019.12.15 更改 */

1、删除 wxPaymentService 的 requestWxPayment 方法的 payChannel 参数,同时删除 paymentService 模块的 updatePayment 方法的 payChannel 属性

2、删除 paymentService 模块的 deletePayment(), getPayment()和 getAllPayments()方法。

订单模块调用此方法请求下单支付

* 此方法再调用 WxPayment 模块的 useWxPay() 方法（模拟微信统一支付 api），useWxPay 方法应该返回 prepay_id 信息

* 此方法拿到 prepay_id 信息后，需要修改数据库中 payment 表相应属性的属性值，如 paySn 等等

* 调用此方法后，前端应显示包括商户名称、订单总价、支付方式(等)信息的界面，供用户确认其支付信息

* 确认其支付信息后，用户可以点击确认支付按钮调用 WxPayment 模块的 RequestPayment() 方法发起最终支付

POST /payment

参数 @RequestBody Payment payment

Return Payment

wxPaymentService 模块：

* 支付模块调用此方法模拟微信统一支付 api

* 此方法应该返回给支付模块 prepay_id 等信息

1、POST /wxpayment

参数 @RequestBody Payment payment

Return String (paySn)

2、

/**

* 用户发起最终支付

* 此方法将调用 Payment 模块的 updatePayment 方法修改支付状态等信息

* updatePayment 方法进而调用 Order 模块的 updateOrder 方法修改订单状态等信息

* @param paySn

* @Param payChannel

```

* @Param endTime
* @return WxPayment (只有两个字段 String paySn 和 Payment payment)
*/

@PutMapping("wxpayment/{id}")
public Object requestWxPayment(@PathVariable("id") String prepay_id,
    LocalDateTime endTime) ;

```

paymentService 模块:

- * (模拟的) 微信后台调用此方法修改订单状态
- * 此方法还会调用 order 模块的 updateOrder 方法, 修改订单状态等信息

```
@PutMapping( "/payment/{id}" )
```

参数 @PathVariable String paySn, boolean successfulPayment

Return Payment

orderService 模块

- 1、提供接口给 Payment 模块回调, 修改该订单为付款完成

```
@PutMapping( "/order/{id}" )
```

参数 @PathVariable Integer id

Return Order order (当初 order 模块也是传给 payment 模块这个参数)

- 2、提供接口给 AfterSale 查看 orderItem 是什么类型

```
@GetMapping ( "ordeltem/{orderItemId}/goodsType" )
```

参数 @PathVariable Integer orderItemId

Return goodsType (Integer)

```
/*12.15 修改*/
```

3、每天晚上搜索前一天 24 小时之内的完成的grouponpo，然后对这些 po 都调用这个接口，根据grouponpo 里的团购开始时间、结束时间和 goodsid 检查所有买了这个商品的 order，返给 discount 模块就行

GET ("/orders/grouponOrders")

参数 GrouponRulePo grouponRulePo

Return List<Order>

4、提供给 discount 的团购部分修改订单的成交价

PUT ("/orders")

参数 List<Order>

return List<Order>

cartItemService 模块

1、提供给订单模块，通过 id 查相应 cart

@GetMapping("/cartItem/{userId}")

参数 @PathVariable Integer userId

Return ArrayList<cartItem>

shareService 模块：

1、提供一个接口给 Order 模块，每晚 12 点 order 模块给 share 模块

order 对象，让他计算该订单是否产生返点，share 模块计算后返回该返的

返点数，供订单模块以后修改 user 模块返点

@GetMapping("/rebate")

参数 @RequestBody

Order order

Return rebate(Integer)

userInfoService 模块

1、 修改返点，给订单模块调用

@PutMapping("/user/rebate")

参数@RequestParam Integer

rebate

Return rebate

2、 根据 userName 查询 userPo 对象供其他模块获取 userId 等

@GetMapping("user/username")

参数@RequestParam

String userName

Return userPo

freightService 模块

1、 提供接口给 orderService 模块，orderService 模块给一个 Order，里面有 address，可以计算运费

@GetMapping("/freightPrice")

参数 @RequestBody

Order order

Return freightPrice (BigDecimal 类型)

logisticsService 模块

随便返回一个物流号

@GetMapping("/logistics")

参数无

Return String 类型

/*12.15 更新，删除参数 userId*/

footprintService 模块

1、提供接口给 Goods 模块，用于增加足迹

@PostMapping("/footprints")

参数@RequestBody FootprintItemPo footprintItemPo

Return 产生的

FootprintItemPo footprintItemPo