## 订单模块

## **com.newer.mall.order.**

### CustomerOrder

#### addOrder

方法描述: 用户下订单

方法流程: 1. 前端提交 HTTP POST 请求 **CustomerOrderController 的 addOrder**方法

方法签名: public boolean addOrder(Orders order, CartItem cartItem)

返回: true 成功 , false 失败

方法请求映射路径: api/order/addorder

参数: order 订单对象

cartItem 购物车项

控制器流程: **addOrder**方法调用**CustomerOrderService 中的 addOrder** 业务方法,下单成功则返回:true,失败则返回false

2. 业务逻辑 **CustomerOrderService** 依赖 **CustomerOrderMapper** ,并定义启用事务的**addOrder** 方法

方法签名: public boolean addOrder(Orders order, CartItem cartItem)

方法中的业务流程:

1. 根据传来的order 对象 插入订单表中生成新的订单调用**CustomerOrderMapper**
2. 根据传来的cartItem 对象,新生成的订单id组合生成新的订单项 调用**CustomerOrderMapper**

3. 基于MyBaits 在 **CustomerOrderMapper** 中实现基于注解的数据操作

方法签名: public void addOrder(Orders order)

方法描述:插入订单表

方法签名:public int findOid(int uid)

方法描述:查询客户最新生成的订单id

返回:订单id

方法签名:public void addItem(int oid, CartItem cartItem)

方法描述:插入新的订单项

4. 使用表: 订单表 orders

(1) 订单表中字段

uid 用户id

name 收件人姓名

phone 收件人电话

address 收货地址

time 下单时间

payway 支付方式

total 订单总价

订单项表 item

(2) 订单项表中所有字段

#### findItem

方法描述: 查询订单

方法流程: 1. 前端提交 HTTP POST 请求 **CustomerOrderController 的 findItem**方法

方法签名: public PageInfo<Item>(int uid)

返回: item集合

方法请求映射路径:api/order/ finditem

参数:uid 客户id

控制器流程: **findItem**方法调用**CustomerOrderService 中的 findItem** 业务方法,返回订单项分页集合

2. 业务逻辑: **CustomerOrderService** 依赖 **CustomerOrderMappe,**并定义**findItem**方法

方法签名: public PageInfo<Item> findOrders(int uid)

方法中的业务流程:

1. 根据客户的客户id查询其下的所有订单,调用**CustomerOrderMappe**,并返回PageInfo<Item>

3. 基于MyBaits 在 **CustomerOrderMapper** 中实现基于注解的数据操作

方法签名:public list<Item> findItem(int uid)

方法描述:查询客户的订单项 ,且hidden字段为0的订单项

4. 使用表:

1) 订单项表中所有字段

#### searchItem

方法描述: 搜索指定条件订单

方法流程: 1. 前端提交 HTTP POST 请求 **CustomerOrderController 的 searchItem**方法

方法签名: public PageInfo<Item> serarchItem(String conditions,int uid)

返回:PageInfo分页数据

方法请求映射路径:api/order/searchitem

参数:condition 条件 uid 客户id

控制器流程: searchItem方法调用**CustomerOrderService** 中的 searchItem业务方法,并返回PageInfo<Item>分页数据

2. 业务逻辑:**CustomerOrderService** 依赖 **CustomerOrderMapper** ,并定义的searchItem方法

方法中的业务流程:

1. 根据传过来的条件conditions 和客户的uid去**调用CustomerOrderMappe**r中的searchItem方法,取到相应的数据

3．基于MyBaits 在 **CustomerOrderMapper** 中实现基于注解的数据操作

方法签名:public list<item> searchItem(String conditions,int uid)

方法描述:根据条件搜索相关的订单,使用模糊查询等

返回:订单项集合

4. 使用的表: item 订单项表

使用的字段:订单项所有字段

使用表: 订单表 和 订单项表

参数: conditions 条件 , uid 用户id

返回值: Orders集合

异常: 无

#### dltOrder

方法签名: public boolean dltOrder(int oid)

方法描述: 根据订单id进行修改操作

方法流程: 1. 前端提交 HTTP POST 请求 **CustomerOrderController 的 dltOrder**方法

方法签名: public boolean dltOrder(int oid)

返回:true 成功 flase 失败

参数: oid 订单id

控制器流程: **dltOrder**方法调用**CustomerOrderService 中的 dltOrder** 业务方法,下单成功则返回:true,成功则重新发送请求失败则返回false

2. **业务逻辑: CustomerOrderService 依赖 CustomerOrderMappe,并定义dltOrder方法**

方法签名: public boolean dltOrder (List<Integer> oids)

方法中的业务流程:

1. 根据传来的订单id,去**调用CustomerOrderMapper**中的upOrder方法

3. 基于MyBaits 在 **CustomerOrderMapper** 中实现基于注解的数据操作

方法签名: public boolean upOrder(int oid)

方法描述: 根据订单id去修改订单表中的hidden字段成功成功就修改成1

返回: true 成功 flase 失败

4. 使用表:Oders表

使用字段: hidden 是否删除状态

Id 订单id

#### addComment

方法描述: 用户对已到货的商品进行评论

方法流程: 1. 前端提交 HTTP POST 请求 **CustomerOrderController 的 addComment**方法

方法签名: public void addComment(Comment comment)

返回:无

方法请求映射路径: api/order/addcomment

参数: comment 评论实体

控制器流程: **addComment**方法调用**CustomerOrderService 中的 addComment** 业务方法,去实现新一条评论记录

2.业务逻辑**CustomerOrderService** 依赖 **CustomerOrderMapper** ,并定义启用事务的**addComment** 方法

方法签名:public void addComment(Comment comment)

方法中的业务流程:

1. 根据传来的评论实体comment为基本插入到数据库的评论表中**调用CustomerOrderMapper**中addComment

3.基于MyBaits 在 **CustomerOrderMapper** 中实现基于注解的数据操作

方法签名:public void addComment(Comment comment)

方法描述:插入新的用户评论

4.使用表: comment

使用字段: Commodity\_id 商品 id

uid 客户id

content 评论内容

score 评分

### AdminOrder

#### findAllItems

方法描述: 查询订单

方法流程: 1. 前端提交 HTTP POST 请求 **AdminOrderController的 findAllItems**方法

方法签名: public PageInfo<Item>()

返回: item集合

方法请求映射路径:api/order/ findallitems

参数:无

控制器流程: **findOrders**方法调用**AdminOrderService 中的 findAllItems** 业务方法,返回订单项分页集合

2. 业务逻辑: **CustomerOrderService** 依赖**AdminOrderMappe,**并定义**findAllItems**方法

方法签名: public PageInfo<Item> findOrders(int uid)

方法中的业务流程:

1. 根据客户的客户id查询其下的所有订单,**调用AdminOrderMappe**,并返回PageInfo<Item>

3. 基于MyBaits 在 **AdminOrderMappe**中实现基于注解的数据操作

方法签名:public list<Item> findAllItems(int uid)

方法描述:查询客户的订单项 ,且hidden字段为0的订单项

4. 使用表:

1) 订单项表中所有字段