# 1、数据架构

## 1.1 数据库ER模型

## 1.2 数据库逻辑模型

## 1.3 数据库物理模型

# 2、接口

## 2.1 查询商品的促销活动

PromotionFacadeService

List<PromotionActivityDTO> listPromotionActivitiesByGoodsId(Long goodsId)

在商品详情页，购物车界面，需要在其他的系统中，将促销活动的数据查询过去

## 2.2 查询用户领取的优惠券

订单确认页面中，会ajax异步加载用户领取的有效的还未使用的优惠券列表

这个是走controller接口

# 3、开发架构

## 3.1 实现类图

## 3.2 包设计

# 4、运行流程

## 4.1 促销活动

### 4.1.1 促销活动列表

促销活动的数据包含了哪些字段：

（1）ID

（2）促销活动名称

（3）促销开始时间

（4）促销结束时间

~~（5）针对会员级别~~

（6）活动备注

（7）创建时间

~~（8）创建人~~

（9）状态：~~待审核、已审核，~~不是手动控制的，是根据你制定的一个促销开始时间和促销结束时间，自动启用/停用，促销开始时间到了就启用，促销结束时间到了就停用

列表显示的字段包含了：ID、促销活动名称、促销开始时间、促销结束时间~~、针对会员级别~~、创建时间~~、创建人~~、状态

支持分页查询

支持筛选条件：促销活动名称，促销开始时间，促销结束时间，状态（默认是启用，可选全部/停用）

### 4.1.2 新建促销活动

1. 我们的促销活动的规则、优惠券的规则、运费计算的规则，都是以json格式定义的，因为一般尤其是在一些快速发展的互联网公司中，电商公司，促销活动变化的很快，每年可能都会新搞一些不同的促销活动出来，很多时候都不太一样。如果你是用表结构来保存这些促销规则，那么可能会导致促销规则表的表结构会频繁的变化。为了应对这种频繁变化，而且可能不断复杂的促销规则，所以我这里是采用json格式来存储促销规则的。但是对于一些促销活动规则不怎么变化的公司来说，用表结构来存储促销规则，也没什么问题
2. 都是直接通过fastjson从json格式的规则中提取出来了我们需要的一些字段

（3）目前的使用方式是很原始的，基本上来说，你读取json格式的规则中的字段的代码，是直接跟规则的格式耦合在一起的。如果说规则的格式变化了，那么你的读取规则的代码必然跟着变化

那些促销活动实际上大同小异，主要是分为活动基本信息、促销规则、主商品信息、赠品信息

（1）活动基本信息

促销活动名称、促销开始时间、促销结束时间、~~会员级别（全员/某等级会员）~~、活动备注

（2）促销规则

活动规则，选择6种活动里面的某一种，先通过下拉框进行选择，然后根据选择的活动规则，输入具体的活动规则明细

满减促销，那么就要输入一系列的细则，满多少减多少

1300 - 100 = 1200 -> 100

100

[

{

“thresholdAmount”: 300,

“reduceAmount”: 100

},

{

“thresholdAmount”: 200,

“reduceAmount”: 50

}

]

*[*

*{*

*“a”: {*

*“a1”: {*

*“thresholdAmount”: 300*

*}*

*},*

*“b”: {*

*“b1”: {*

*“reduceAmount”: 100*

*}*

*}*

*}*

*]*

{

“thresholdAmount”: 300

}

“a/a1/thresholdAmount”

a a1 thresholdAmount

thresholdAmount

a thresholdAmount

抽象语法树是由多个Expression组成的，每个Expression代表了这个复杂文本中的一个元素

然后这颗抽象语法是是自解释的，从根节点开始调用interpret方法就可以自动解释整颗抽象语法树，最后获取到你需要的这个复杂文本中的一个值

我们这个版本中，实际上就是这个规则已经算是非常简单的了，所以才让规则的读取代码看起来简单一些。实际上如果促销种类很多，规则很复杂的话，可能会导致频繁的变更这个规则的格式，导致我们频繁的变动读取规则的代码。

解释器模式来重构读取规则的这块代码

“/thresholdAmount” -> 语法解析器 -> 抽象语法树 -> Expression(“thresholdAmount”) -> 解释器 -> 300

“/a/thresholdAmount”

CompositeJsonExpression(“a”)

LeafJsonExpression(“thresholdAmount”)

解释这颗抽象语法树，来获取a下面的thresholdAmount的值，但是我们不会专门去做一个解释器的，抽象语法树，是会做成自己解释自己，自解释，解释抽象语法树的代码，就写在抽象语法树自己本身了

我们不用再写任何一行的json格式规则解析的代码了，我们就是直接按照预定义的语法规则，说我现在就要取thresholdAmount的值，”/thresholdAmount”语法进去

抽象语法树 -> 组合模式 -> 观察

返回给我们的就是一个300

“/a/thresholdAmount” -> 300

“/a/b/thresholdAmount” -> 300

解释器模式常用于哪些场景？你自定义了一些比较复杂的有一定规则和格式的数据，做了一个框架，自定义了一个复杂的xml配置文件，你读取这个xml配置文件，你就可以用解释器模式来做

不要自己手写大量的硬编码的跟xml里的内容强耦合的xml解析代码，一旦xml格式变化了，xml解析代码也要跟着变，基于xml封装一套解析器+解释器，你如果要读取xml中的一些值，只要传递一些你自己预定义的语法规则就可以了

包括比如说，类似我们这里的这种场景，促销规则，优惠券规则，运费计算规则，变化会很频繁，我们就不要自己一次又一次的手写硬编码的跟json规则格式强耦合的json解析代码，去读取规则里的字段。我们就是封装一套解析器+解释器，你只要给我们一个读取json规则中某个字段的值的一个语法，我就可以解析这个语法，解释这个语法，去从json规则中找到你需要的字段的值

多买优惠，输入一系列的细则，就是买几件打几折

[

{

“thresholdCount”: 3,

“discountRate”: 0.8

},

{

“thresholdCount”: 2,

“discountRate”: 0.9

}

]

单品促销，输入一个折扣

{

“discountRate”: 0.7

}

~~套装促销，输入一个折扣~~

~~{~~

~~“discount\_rate”: 0.6~~

~~}~~

满赠促销，要输入满多少元赠送一些赠品

{

“thresholdAmount”: 550,

“giftGoodsSkuIds”: [2, 340, 590]

}

赠品促销

{

“giftGoodsSkuIds”: [2, 340, 590]

}

（2）主商品信息

根据sku编码、商品名称、商品状态、价格范围、商品分类~~、上架时间~~，等几个条件，来进行商品的筛选。筛选出来的商品列表的字段包括：SKU编码、商品名称、价格、优惠内容。

满减促销：~~不要选择商品了，默认就只能范围是全部商品，就是说无论买什么商品，满多少钱就可以立减多少钱~~，选择一个或者是多个商品，就是代表着针对这些商品有这个优惠；也可以选择全部商品

单品促销：就是要选择一个或者多个商品，也可以选择全部商品，直接会关联上面的那个折扣的

~~套装促销：要求选择超过1个商品，多个商品，不可以选择全部商品的，最多选择10个以内的商品。如果同时一个订单中出现这多个商品的话，那么就会对这些商品走上面的一个折扣价格~~

赠品促销：只能选择一个商品或者多个商品，也可以选择全部商品，买这个商品的时候，每个商品都会给一份赠品，赠品在下面设置

满赠促销：~~默认就是不要选择商品，强制是全部范围的商品~~，可以选择一个或者多个商品，也可以选择全部商品，就是针对这个商品买满多少钱就赠送一个赠品

多买优惠：可以选择一个或者多个商品，也可以选择全部商品，这个商品买满几件打几折

我说一点，大家看到这里，不知道能不能理解我为什么不愿意去画原型图，原型图太复杂了，不能的情况下，前端界面是各种变化的

刚开始新建完的时候，促销活动的状态根据开始时间来判断一下，如果开始时间是早于当前时间，那么促销活动就直接启用；如果开始时间是晚于当前时间，那么促销活动就处于停用状态

如果某个促销活动选择自己的范围是全部商品，在那个促销活动和商品关系表里面插入一条数据，goods\_id是-999。

（3）赠品信息

选择赠品其实也是选择商品，跟上面是一样的

赠品促销，可以挂一个或者多个赠品，赠品最多挂10个

满赠促销，也可以挂一个或者多个赠品，赠品最多挂10个

刚新建完一个促销活动，待审核；只有待审核、停用的状态，可以进行编辑，编辑完了以后还是待审核

### 4.1.3 根据ID查询促销活动

根据ID将促销活动的数据查询出来在页面上回显即可

### 4.1.4 编辑促销活动

对促销活动的数据进行编辑即可，但是只有处于停用状态的促销活动是可以更新的

### ~~4.1.5 审核促销活动~~

~~对每个促销活动都可以进行审核，在促销活动列表页的右侧操作栏中，有一个审核按钮，如果是待审核状态，可以点击审核按钮。进入一个审核页面，此时会显示促销活动完整的一个信息。然后执行审核，通过/不通过。如果通过审核，就变成已审核。~~

### 4.1.6 删除促销活动

只有处于停用状态的促销活动可以删除。

### 4.1.7 促销活动的启用/停用

新建一个促销活动之后，到时间就会自动启用这个活动，需要有一个程序在后台去监控每个促销活动的开始和截止的时间，看到到了开始时间，就开始自动将那些促销活动的状态修改为启用即可，监控所有促销活动的截止时间，到时间之后，就自动将促销活动的状态修改为停用。

### 4.1.8 促销活动的前台展示

（1）在商品详情页，需要展示出来促销活动，查询这个商品，所有处于可用状态的促销活动

（2）在购物车页面，也需要同时展示商品的促销活动，要求你手动去选择一个促销活动，每个商品支持可以挂多个促销活动的。此时需要我们手动去下拉列表中选择

一般来说，都是对一个spu去设置这个对应的促销活动的

## 4.2 优惠券

很多种优惠券：现金券（无限制直接用）、满减券、~~折扣券（购买指定商品时打折）~~，实际上是在提交订单的确认页面里，针对整个订单去使用的，不是针对某个商品的

### 4.2.1 优惠券列表

（1）优惠券数据

优惠券信息：优惠券名称、类型（现金券、满减券）、使用条件规则（多少钱的现金券、满减券有一系列细则）、有效时间（范围，开始~结束）、发行量~~、商品范围~~

推广信息：发放方式（可发放可领取、仅可发放、仅可领取）~~，推广范围（免费领取、积分兑换），限领数量，会员等级~~

状态：发放中、发放完、过期

1. 列表页显示的字段

优惠券名称、类型、有效时间、发行量、发放方式、~~推广范围、限领数量~~、创建时间、状态

支持分页查询

支持条件筛选：优惠券名称、类型、有效时间、发放方式、状态、~~推广范围~~

### 4.2.2 优惠券新建/编辑

新建/编辑，其实就是显示一个表单出来，可以对优惠券所有的数据进行输入和编辑

现金券的rule：

{

“discountAmount”: 300.00

}

满减券的rule：

[

{

“thresholdAmount”: 300,

“reduceAmount”: 100

},

{

“thresholdAmount”: 200,

“reduceAmount”: 50

}

]

更新优惠券之前，要判断一下，只有还没有被领取的优惠券才可以进行编辑

### 4.2.3 删除优惠券

~~优惠券的删除，不好随意删除的，你要是删除了以后，就得把发出去的所有的券都给删除了，这个会很麻烦~~

删除的条件：没有发放，也没有领取

### 4.2.4 用户领取优惠券

就是在一个领券页面里，可以点击按钮来领取，这里系统需要记录好，当前领了多少张券了，券给了谁。根据发券的数量，维护券的状态。

领取的时候，需要做一些校验：优惠券是否处于有效期内；优惠券的状态：发放中；优惠券的领取方式是否支持用户来领取

领券的逻辑：

1. 判断当前优惠券的一些状态，比如是否处于有效期，是否已经发放完，发放方式是否支持领取
2. 先判断当前用户是否已经领取过这个优惠券了，如果已经领取过了，那么就不能再次领取了，每个用户只能领取一张优惠券
3. 插入一条用户领取这个优惠券的记录
4. 对优惠券表中的已经领取的数量，给增加1；每次领完券之后，都要维护优惠券的状态，包括已经领取的数量，如果券已经领完了，那么就将状态设置成发放完
5. 还得有一个后台线程定时去check，刚开始优惠券的状态是未开始，如果到了有效期的开始时间，那么处于发放中，如果超出了有效期，那就是已过期

### 4.3.5 平台发放优惠券

~~那就是平台直接发放给所有用户了，在优惠券的列表页的右侧，有一个按钮，就是发放优惠券。随机性发放，根据优惠券有多杀张，随机选择指定的用户来一人一张来发放~~

这个版本，默认就是给每个用户都发放一张优惠券，这个模式下，不care优惠券的发行数量了，就是给每个用户都强行推送一张优惠券就可以了

发券的逻辑：

1. ~~给每个用户都插入一个优惠券的领取记录，就很麻烦，而且如果你的用户数量有上百万呢？插入上百万条记录~~
2. 只有处于发放中状态，以及类型为可发放可领取或者是仅可发放，可以发放优惠券的
3. 就是在每个用户领取券的记录表中插入一条记录，就代表着这个优惠券所有的用户都可以使用
4. 将优惠券表中的领取的数量设置成跟发行量是一样的，优惠券的状态变为：发放完

### 4.3.6 查询可用的优惠券

在订单确认页面中，可以查询这个用户当前可用的优惠券：

1. 用户领取过的优惠券
2. 用户还没有使用过这个优惠券
3. 优惠券当前处于发放中，或者是发放完的状态

### 4.3.7 使用优惠券

在创建订单的时候，前台需要传递过来一个couponId，就是说是否使用了某个优惠券，然后在创建订单的逻辑中，需要加一点逻辑，就是订单中心来调用促销中心的接口，通知促销中心，这个优惠券被哪个用户给使用了。

然后促销中心就更新优惠券领取记录中的is\_used和used\_time两个字段

# 5、测试用例设计

# 6、日志设计