

# day09-优惠券管理

同学们，之前几天我们实现的都是学习辅助功能。学习辅助的目的是提升用户的学习体验，维护好老用户。而一个网站要想长久发展，除了要服务好老用户，还必须能吸引新用户，也就是拉新。而拉新最常见的手段就是优惠促销，比如优惠券。

从今天开始，我们就一起来实现一个优惠券管理的服务。优惠券功能比较复杂，包含的业务亮点也非常多，例如：

- 优惠券的优惠策略设计
- 优惠券的兑换码算法
- 优惠券领取的并发安全问题及解决方案
- 优惠券叠加方案的智能推荐
- 多商品、多券叠加时的优惠金额计算
- 多商品订单退款的拆单和退券问题

等等。。

这些方案不仅仅是在咱们项目，在所有电商类型的项目中都是热点、难点问题，在接下来几天的学习中我们会逐一分析和解决。

## 1.需求分析

需求分析的流程与以往类似，还是基于产品原型，三步走：

- 分析业务流程
- 统计业务接口
- 设计数据库表

### 1.1.业务流程梳理

优惠券包括两大部分功能：

- 优惠券管理和发放（管理端）
- 优惠券的领取和使用（用户端）

我们今天先实现管理端的功能。在后台管理营销中心的优惠券管理页面，可以看到一个优惠券列表页：

营销中心 / 优惠券管理

早上好! 今天也是元气满满的一天哦!

管理员

优惠券类型 请选择优惠券类型

优惠券状态 请选择优惠券状态

优惠券名称 请输入优惠券关键字

Q 搜索

C 重置

新增优惠券

优惠券名称	优惠券规则	使用范围	推广方式	使用/领取/发放	创建时间	领用期限	使用期限	状态	操作
秒杀天天有, 今天更刺激	每满500元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:22:11	-	-	待发放	<a href="#">编辑</a> <a href="#">发放</a> <a href="#">查看兑换码</a> <a href="#">删除</a>
618年中大促第四波	每满1000元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:21:11	2022.06.7 18:00:00 —— 2022.06.10 18:00:00	2022.06.11 18:00:00 —— 2022.06.20 18:00:00	已暂停	<a href="#">查看</a> <a href="#">发放</a> <a href="#">查看兑换码</a> <a href="#">删除</a>

我们可以在这里实现优惠券的基础的增删改查功能。

不过，新增的优惠券并不会立刻出现在用户端页面，管理员还需要对优惠券信息做**审核**，审核通过后则可以通过发放按钮来发布优惠券。

而优惠券的发布也有两种不同的方式：

设置领用/使用期限

\*发放方式

☒ 立刻发放

☐ 定时发放

\*领用期限

请设置结束时间

\*使用限制方式

☐ 固定天数

☒ 固定时间段

\*使用期限

请设置开始时间 ~ 请设置结束时间

取消

保存

设置领用/使用期限

\*发放方式

☐ 立刻发放

☒ 定时发放

\*领用期限

请设置开始时间 ~ 请设置结束时间

\*使用期限方式

☒ 固定天数

☐ 固定时间段

\*使用期限

请输入天数 天

取消

保存

一个是立刻发放，一个是定时发放。

顾名思义，立刻发放就是优惠券立刻生效，会直接出现在用户端页面供用户领取。定时发放则需要指定一个发放开始时间，时间到期后才会进入出现在用户端页面。

而且无论是哪种发放方式，都需要指定一个过期时间，当优惠券过期后就会进入已结束状态，用户端页面无法领取。

当然，除了过期导致的结束发放以外，管理员也可以手动点击暂停发放：

1 优惠券类型 请选择优惠券类型

优惠券状态 请选择优惠券状态

优惠券名称 请输入优惠券关键字

Q 搜索

C 重置

2 新增优惠券

3

优惠券名称	优惠券规则	使用范围	推广方式	使用/领取/发放	创建时间	领用期限	使用期限	状态	操作
秒杀天天有，今天更刺激	每满500元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:22:11	—	—	待发放	<a href="#">编辑</a> <a href="#">发放</a> <a href="#">查看兑换码</a> <a href="#">删除</a>
618年中大促第四波	每满1000元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:21:11	2022.06.7 18:00:00 — 2022.06.10 18:00:00	2022.06.11 18:00:00 — 2022.06.20 18:00:00	已暂停	<a href="#">查看</a> <a href="#">发放</a> <a href="#">查看兑换码</a> <a href="#">删除</a>
清仓放利，惠到你家	无门槛200元	指定课程分类	指定发放	10/20/100	2022.05.29 22:12:11	2022.06.07 18:00:00 — 2022.06.10 18:00:00	2022.06.11 18:00:00 — 2022.06.20 18:00:00	未开始	<a href="#">查看</a> <a href="#">暂停</a> <a href="#">查看兑换码</a> <a href="#">删除</a>

黑马程序员-研究部

也可以在需要的时候重新发放优惠券。

特别需要注意的是，优惠券的领取方式有两种，来看一下优惠券的新增表单：

## 优惠券基本信息

\*优惠券名称  0/20 1

\*使用范围 ☐ 全部课程分类 ☒ 指定课程分类 2

▼

\*优惠券类型 ☐ 每满减 ☒ 满减 ☐ 无门槛 ☐ 折扣 3

满  元 减  元

## 优惠券推广方式

\*推广方式 ☐ 手动领取 ☒ 指定发放 4

\*发放数量 指定  张 5

\*每人限领 限领  张 6

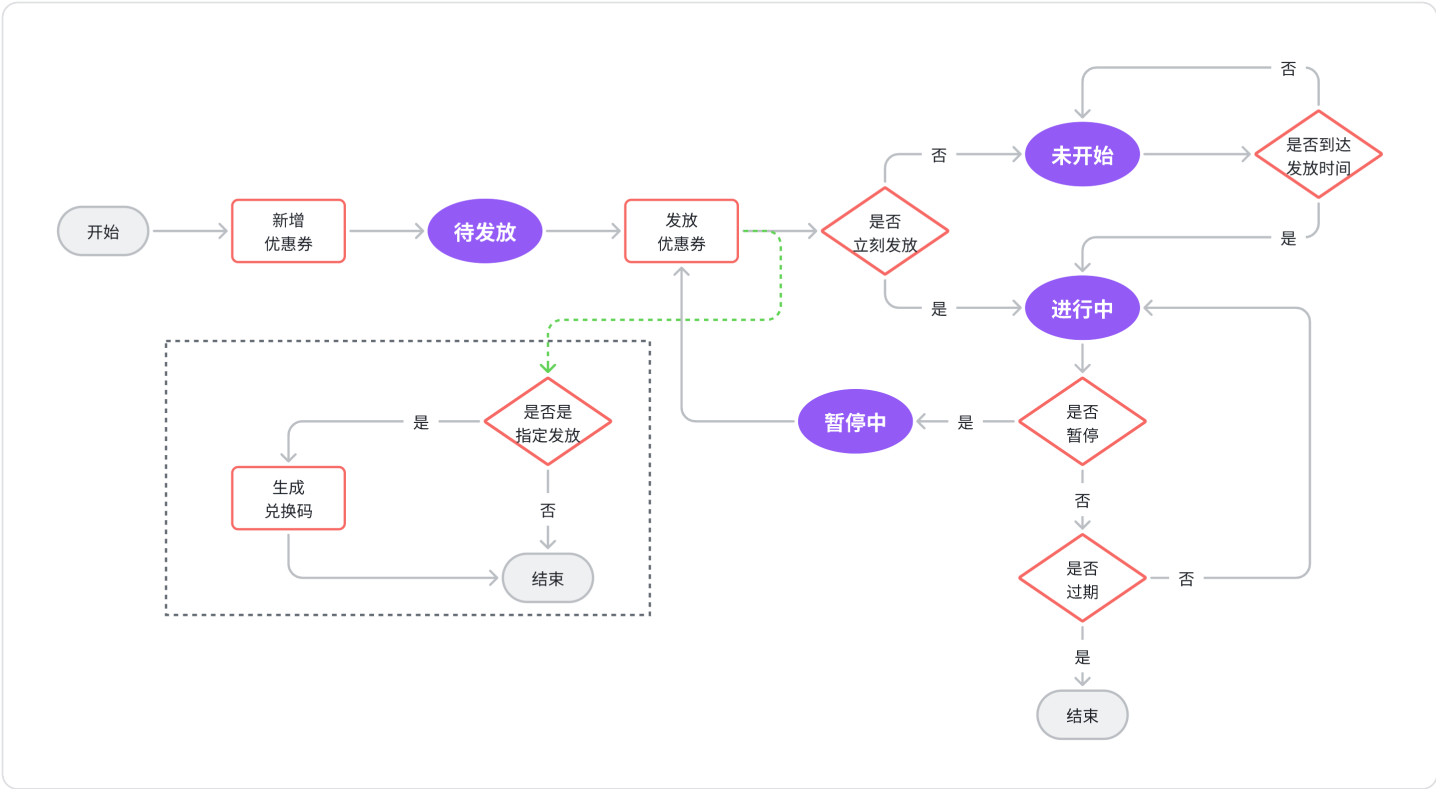
黑马程序员-研究院

领取方式有两种：

- **手动领取**：就是展示在用户端页面，由用户自己手动点击领取
- **指定方法**：就是兑换码模式，后台给优惠券生成N张兑换码，由管理员发放给指定用户。

这就要求我们在发放优惠券的时候做判断，如果发现是指定发放模式，则需要提前生成兑换码。

综上，优惠券管理的业务流程和优惠券的状态转换如图：



1.2.接口统计

首先，在优惠券的列表页：

营销中心 / 优惠券管理

早上好! 今天也是元气满满的一天哦!

管理员

优惠券类型 请选择优惠券类型

优惠券状态 请选择优惠券状态

优惠券名称 请输入优惠券关键字

搜索

重置

新增优惠券

优惠券名称	优惠券规则	使用范围	推广方式	使用/领取/发放	创建时间	领用期限	使用期限	状态	操作
秒杀天天有，今天更刺激	每满500元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:22:11	—	—	待发放	编辑 发放 查看兑换码 删除
618年中大促第四波	每满1000元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:21:11	2022.06.7 18:00:00 — 2022.06.10 18:00:00	2022.06.11 18:00:00 — 2022.06.20 18:00:00	已暂停	查看 发放 查看兑换码 删除
清仓放利，惠到你家	无门槛200元	指定课程分类	指定发放	10/20/100	2022.05.29 22:12:11	2022.06.07 18:00:00 — 2022.06.10 18:00:00	2022.06.11 18:00:00 — 2022.06.20 18:00:00	未开始	查看 暂停 查看兑换码 删除

页面规范如下：

💡

1. 搜索条件

- 优惠类型：天机学堂支持的类型有 1：满减，2：每满减，3：折扣，4：无门槛
- 优惠券状态：包括 1：待发放，2：未开始 3：进行中，4：已结束，5：暂停

## 2. 列表显示

- 默认显示10条
- 默认按照创建时间倒序排序
- 使用/领取/发放：优惠券数量统计，已使用的数量/已领取的数量/总发放数量
- 领用期限：就是券领取的开始和结束时间

可见这个列表就是一个典型的带过滤条件的分页查询。其它增删改查接口都比较简单，不再赘述。

所有接口在页面都一目了然：

- 优惠券的基本管理接口：
  - 分页查询优惠券列表
  - 新增优惠券
  - 编辑优惠券
  - 查看优惠券（根据id查询优惠券）
  - 删除优惠券
- 优惠券的方法接口：
  - 发放优惠券
  - 暂停发放优惠券

另外有几个比较隐蔽的接口。一个是发放优惠券时，如果选择的是定时方法，则需要指定发放时间，到期后才发放。这就需要有一个定时任务，检索优惠券表，找到发放时间到期的券，完成发放功能。

另一个也是发放券问题，券除了有发放时间，还有过期时间，因此需要一个定时任务，检查券的过期时间，发现到期后需要更新券状态。

以上两个都是定时任务接口：

- 定时发放优惠券
- 定时结束优惠券发放

还有一个是跟兑换码有关。就是在发放优惠券的时候，如果发现优惠券的领取方式是指定发放，则需要生成兑换码。因此页面有一个查询兑换码功能：

清仓放利，惠到你家	无门槛200元	指定课程分类	指定发放	10/20/100	2022.05.29 22:12:11	2022.06.07 18:00:00 2022.06.10 18:00:00	2022.06.11 18:00:00 2022.06.20 18:00:00	未开始	查看 暂停 查看兑换码 删除
-----------	---------	--------	------	-----------	------------------------	--	--	-----	----------------

黑马程序员-研究院

当我们点击查看兑换码时，就会进入一个兑换码展示页面：

查看优惠券兑换码

已兑换

未兑换

SGY78HFO	SGY78HFO	SGY78HFO	SGY78HFO
FHE98JEH	FHE98JEH	FHE98JEH	FHE98JEH
GJK23MVN	GJK23MVN	GJK23MVN	GJK23MVN
SGY78HFO	SGY78HFO	SGY78HFO	SGY78HFO
FHE98JEH	FHE98JEH	FHE98JEH	FHE98JEH

共30条

<12>

10条/页 跳至2页

我知道了

黑马程序员-研究院

可以看出来，这是一个有过滤条件的分页查询功能。

综上，优惠券相关接口包括：

	编号	接口简述
优惠券管理	1	新增优惠券
	2	修改优惠券
	3	分页查询优惠券
	4	根据id查询优惠券
	5	删除优惠券

优惠券发放	1	发放优惠券
	2	暂停发放
	3	查询兑换码
定时任务	1	定时发放优惠券
	2	定时结束优惠券

### 1.3.表结构设计

通过前面的接口分析，发现接口主要跟两个实体有关：

- 优惠券
- 兑换码

所以，接下来要设计的表就是以上两张表。

#### 1.3.1.优惠券

首先从优惠券的新增表单来分析，表单页面如下：



## 优惠券基本信息

\*优惠券名称  0/20 1

\*使用范围 ☐ 全部课程分类 ☒ 指定课程分类 2

▼

\*优惠券类型 ☐ 每满减 ☒ 满减 ☐ 无门槛 ☐ 折扣 3

满  元 减  元

## 优惠券推广方式

\*推广方式 ☐ 手动领取 ☒ 指定发放 4

\*发放数量 指定  张 5

\*每人限领 限领  张 6

黑马程序员-研究院

其中的字段包含：

- 优惠券名称：一个普通字符串
- 使用范围：这里有两种选择：全部课程、指定课程分类，也就是不限定课程、限定课程，可以用布尔类型来表示。不过一旦选定了课程分类，就需要指定真正限定的分类。

<input checked="" type="checkbox"/> IT-互联网 (68) >	<input type="checkbox"/> 前端开发 >	<input checked="" type="checkbox"/> 机器学习
<input type="checkbox"/> 设计-创作 (15) >	<input type="checkbox"/> 移动开发 >	<input checked="" type="checkbox"/> 深度学习
	<input type="checkbox"/> 编码开发 >	<input checked="" type="checkbox"/> 语音识别
	<input checked="" type="checkbox"/> 人工智能 >	<input checked="" type="checkbox"/> 计算机视觉
	<input type="checkbox"/> 云计算/大数据 >	<input checked="" type="checkbox"/> 强化学习

黑马程序员-研究院

此处是允许多选的，也就是说一个优惠券可以限定多个课程分类。而一个分类也可能被不同的券作为限定范围。因此优惠券与限定的分类是多对多关系。需要一张中间表来保存关系。这个以后再说。

- 优惠券类型：包含满减、每满减、满折扣、无门槛四种，例如：

- 满100减15
- 每满100减10
- 满200打8折，不超过50
- 直减20

可以看出来，虽然规则不同，但都可以用以下几部分来表示：

- 优惠的门槛：比如满100的100
- 优惠值：比如减15的15、打8折的8
- 优惠上限：比如不超过50

因此，我们完全要表示完整优惠策略就需要四个字段：优惠类型、优惠门槛、优惠值、优惠上限

- 推广方式：手动领取和指定发放
- 发放数量
- 每人限领数量

OK，表单中的字段就这么多。然后再看看分页页码：

3	优惠券名称	优惠券规则	使用范围	推广方式	使用/领取/发放	创建时间	领用期限	使用期限	状态	操作
	秒杀天天有，今天更刺激	每满500元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:22:11	—	—	待发放	<a href="#">编辑</a> <a href="#">发放</a> <a href="#">查看兑换码</a> <a href="#">删除</a>

黑马程序员-研究院

与新增页面重复的就不再赘述了，这里多出的一些字段有：

- 已领取数量
- 已使用数量
- 领用期限：也就是优惠券开始发放、结束发放的时间
- 使用期限：用户领取券后的使用期限，有两种方式：

- 固定时间段：需要指定开始时间、结束时间
- 固定天数：指定天数，从用户领取之日起计算
- 优惠券状态

综上，优惠券表结构如下：

```

1 -- 导出 表 tj_promotion.coupon 结构
2 CREATE TABLE IF NOT EXISTS `coupon` (
3   `id` bigint NOT NULL AUTO_INCREMENT COMMENT '优惠券id',
4   `name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL
   DEFAULT '0' COMMENT '优惠券名称，可以和活动名称保持一致',
5   `type` tinyint NOT NULL DEFAULT '1' COMMENT '优惠券类型，1：普通券。目前就一种，保留字段',
6   `discount_type` tinyint NOT NULL COMMENT '折扣类型，1：满减，2：每满减，3：折扣，4：无门槛',
7   `specific` bit(1) NOT NULL DEFAULT b'0' COMMENT '是否限定作用范围，false：不限定，true：限定。默认false',
8   `discount_value` int NOT NULL DEFAULT '1' COMMENT '折扣值，如果是满减则存满减金额，如果是折扣，则存折扣率，8折就是存80',
9   `threshold_amount` int NOT NULL DEFAULT '0' COMMENT '使用门槛，0：表示无门槛，其他值：最低消费金额',
10  `max_discount_amount` int NOT NULL DEFAULT '0' COMMENT '最高优惠金额，满减最大，0：表示没有限制，不为0，则表示该券有金额的限制',
11  `obtain_way` tinyint NOT NULL DEFAULT '0' COMMENT '获取方式：1：手动领取，2：兑换码',
12  `issue_begin_time` datetime DEFAULT NULL COMMENT '开始发放时间',
13  `issue_end_time` datetime DEFAULT NULL COMMENT '结束发放时间',
14  `term_days` int NOT NULL DEFAULT '0' COMMENT '优惠券有效期天数，0：表示有效期是指定有效期的',
15  `term_begin_time` datetime DEFAULT NULL COMMENT '优惠券有效期开始时间',
16  `term_end_time` datetime DEFAULT NULL COMMENT '优惠券有效期结束时间',
17  `status` tinyint DEFAULT '1' COMMENT '优惠券配置状态，1：待发放，2：未开始 3：进行中，4：已结束，5：暂停',

```

```

18 `total_num` int NOT NULL DEFAULT '0' COMMENT '总数量, 不超过5000',
19 `issue_num` int NOT NULL DEFAULT '0' COMMENT '已发行数量, 用于判断是否超发',
20 `used_num` int NOT NULL DEFAULT '0' COMMENT '已使用数量',
21 `user_limit` int NOT NULL DEFAULT '1' COMMENT '每个人限领的数量, 默认1',
22 `ext_param` json DEFAULT NULL COMMENT '拓展参数字段, 保留字段',
23 `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
24 `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP COMMENT '更新时间',
25 `creator` bigint NOT NULL COMMENT '创建人',
26 `updater` bigint NOT NULL COMMENT '更新人',
27 PRIMARY KEY (`id`) USING BTREE
28 ) ENGINE=InnoDB AUTO_INCREMENT=1630563495906942979 DEFAULT CHARSET=utf8mb4
    COLLATE=utf8mb4_0900_ai_ci COMMENT='优惠券的规则信息';
29

```

另外, 用来表示优惠券使用范围时, 需要一个优惠券与课程分类的中间关系表:

```

1 CREATE TABLE IF NOT EXISTS `coupon_scope` (
2   `id` bigint NOT NULL AUTO_INCREMENT,
3   `type` tinyint NOT NULL COMMENT '范围限定类型: 1-分类, 2-课程, 等等',
4   `coupon_id` bigint NOT NULL COMMENT '优惠券id',
5   `biz_id` bigint NOT NULL COMMENT '优惠券作用范围的业务id, 例如分类id、课程id',
6   PRIMARY KEY (`id`),
7   KEY `idx_coupon` (`coupon_id`)
8 ) ENGINE=InnoDB AUTO_INCREMENT=19 DEFAULT CHARSET=utf8mb4
    COLLATE=utf8mb4_0900_ai_ci COMMENT='优惠券作用范围信息';

```

### 1.3.2. 兑换码

兑换码的作用是让用户拿着这个码来兑换一张优惠券。因此一定与两个实体有关:

- 优惠券
- 用户

也就是说, 我们需要知道将来是**谁**来兑换的券, 可以兑换**哪张券**。当然, 兑换码的码肯定也要保持到数据库, 长这样:

查看优惠券兑换码

已兑换

未兑换

SGY78HFO	SGY78HFO	SGY78HFO	SGY78HFO
FHE98JEH	FHE98JEH	FHE98JEH	FHE98JEH
GJK23MVN	GJK23MVN	GJK23MVN	GJK23MVN
SGY78HFO	SGY78HFO	SGY78HFO	SGY78HFO
FHE98JEH	FHE98JEH	FHE98JEH	FHE98JEH

共30条 < 1 2 > 10条/页 跳至 2 页

我知道了

除此以外，为了避免码被重复兑换，我们还需要记录码的状态：

- 码状态：已兑换、未兑换

最后，兑换码同样是有过期时间的，这个时间应该跟优惠券的过期时间一致。


综上，兑换码的最终表结构：

```
1 CREATE TABLE IF NOT EXISTS `exchange_code` (  
2   `id` int NOT NULL COMMENT '兑换码id',  
3   `code` varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL  
   COMMENT '兑换码',  
4   `status` tinyint NOT NULL DEFAULT '1' COMMENT '兑换码状态， 1：待兑换，2：已兑  
   换，3：兑换活动已结束',  
5   `user_id` bigint NOT NULL DEFAULT '0' COMMENT '兑换人',  
6   `type` tinyint NOT NULL DEFAULT '1' COMMENT '兑换类型，1：优惠券，以后再添加其它类  
   型',  
7   `exchange_target_id` bigint NOT NULL DEFAULT '0' COMMENT '兑换码目标id，例如兑换  
   优惠券，该id则是优惠券的配置id',  
8   `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
9   `expired_time` datetime NOT NULL COMMENT '兑换码过期时间',  
10  `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
    CURRENT_TIMESTAMP COMMENT '更新时间',  
11  PRIMARY KEY (`id`) USING BTREE,  
12  KEY `index_status` (`status`) USING BTREE,
```

```
13     KEY `index_config_id` (`exchange_target_id`) USING BTREE
14 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='兑换
    码';
```

所有的SQL文件都在课前资料中提供了：

› 新加卷 (D:) › 课程资料 › 天机学堂 › 课件 › day09-优惠券管理 › 资料 ›

名称	类型	大小
dto	文件夹	
enums	文件夹	
query	文件夹	
utils	文件夹	
vo	文件夹	
 tj_promotion.sql	SQL 源文件	130 KB

黑马程序员-研究院

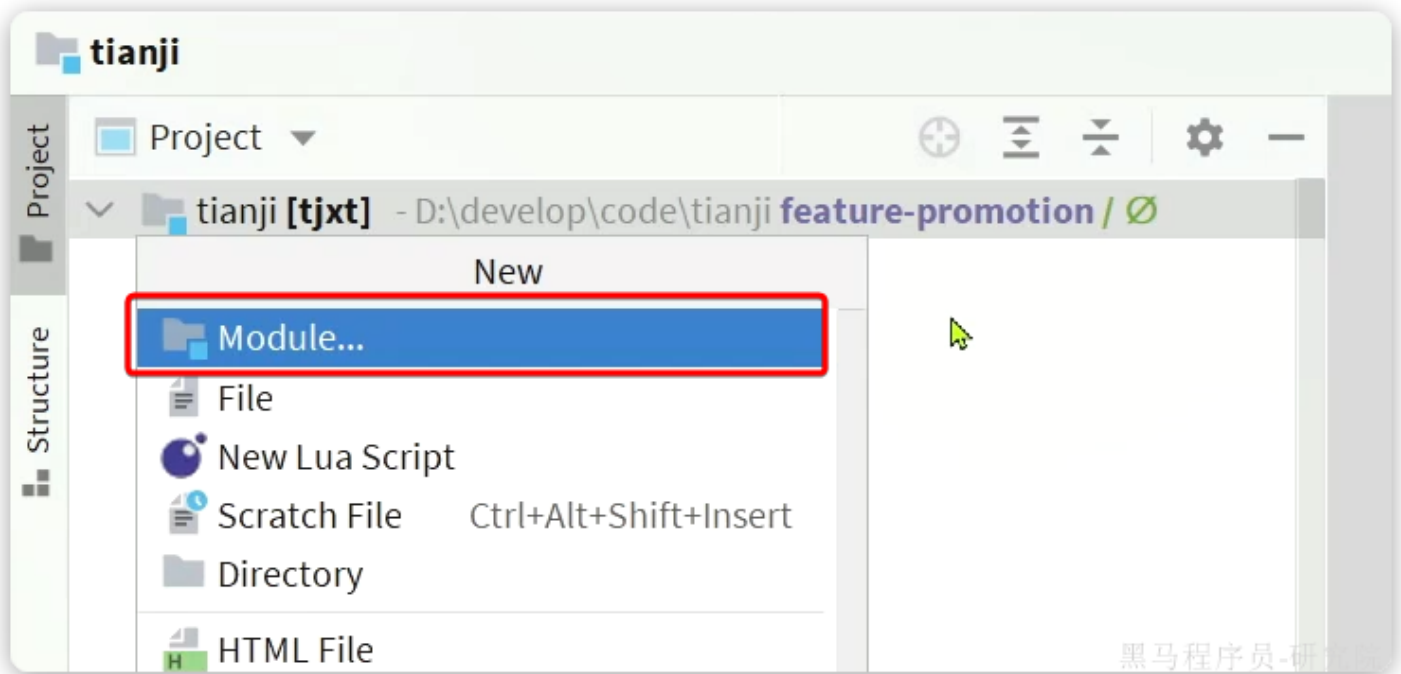
## 1.4.代码生成

首先，在DEV分支的基础上创建一个新的功能分支：

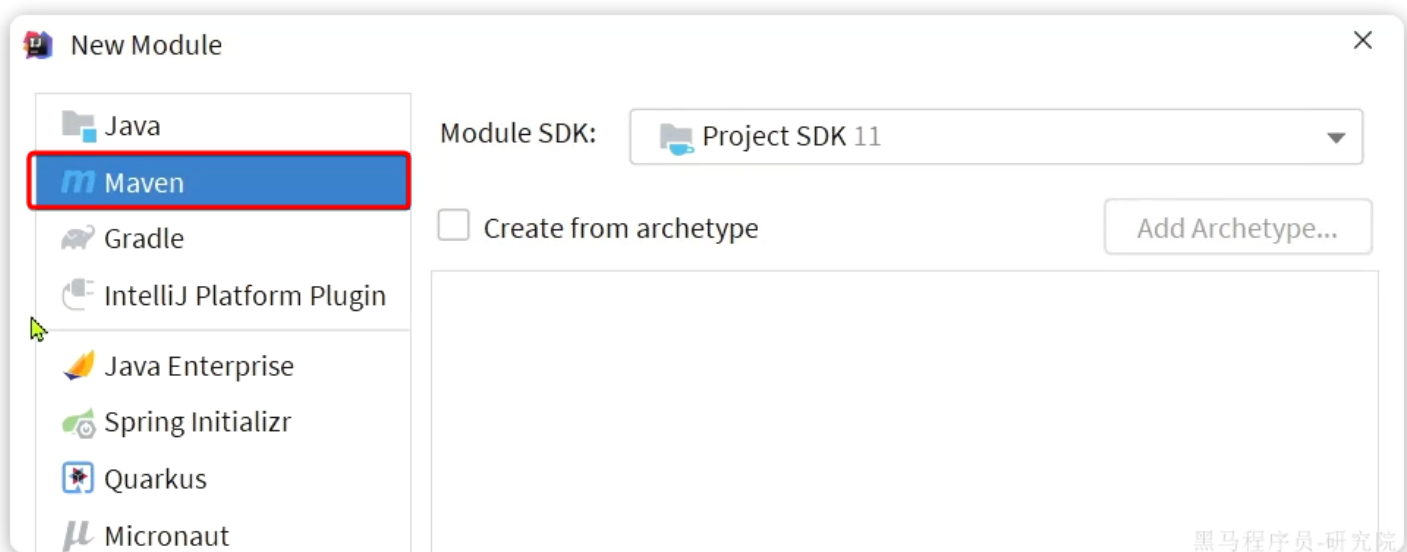
```
1 git checkout -b feature-promotions
```

### 1.4.1.创建新的模块


优惠券功能属于优惠促销的一部分，在项目中肯定属于独立的功能模块。我们需要创建一个新的module：



选择Maven工程：



然后填写项目信息：

 New Module ×

Parent:


m

 tjxt ▼

Name:

tj-promotion

Location:

D:\develop\code\tianji\tj-promotion 


▼ Artifact Coordinates

GroupId:

com.tianji

The name of the artifact group, usually a company domain

ArtifactId:

tj-promotion 

The name of the artifact within the group, usually a module name

Version:

1.0.0

Previous

Finish

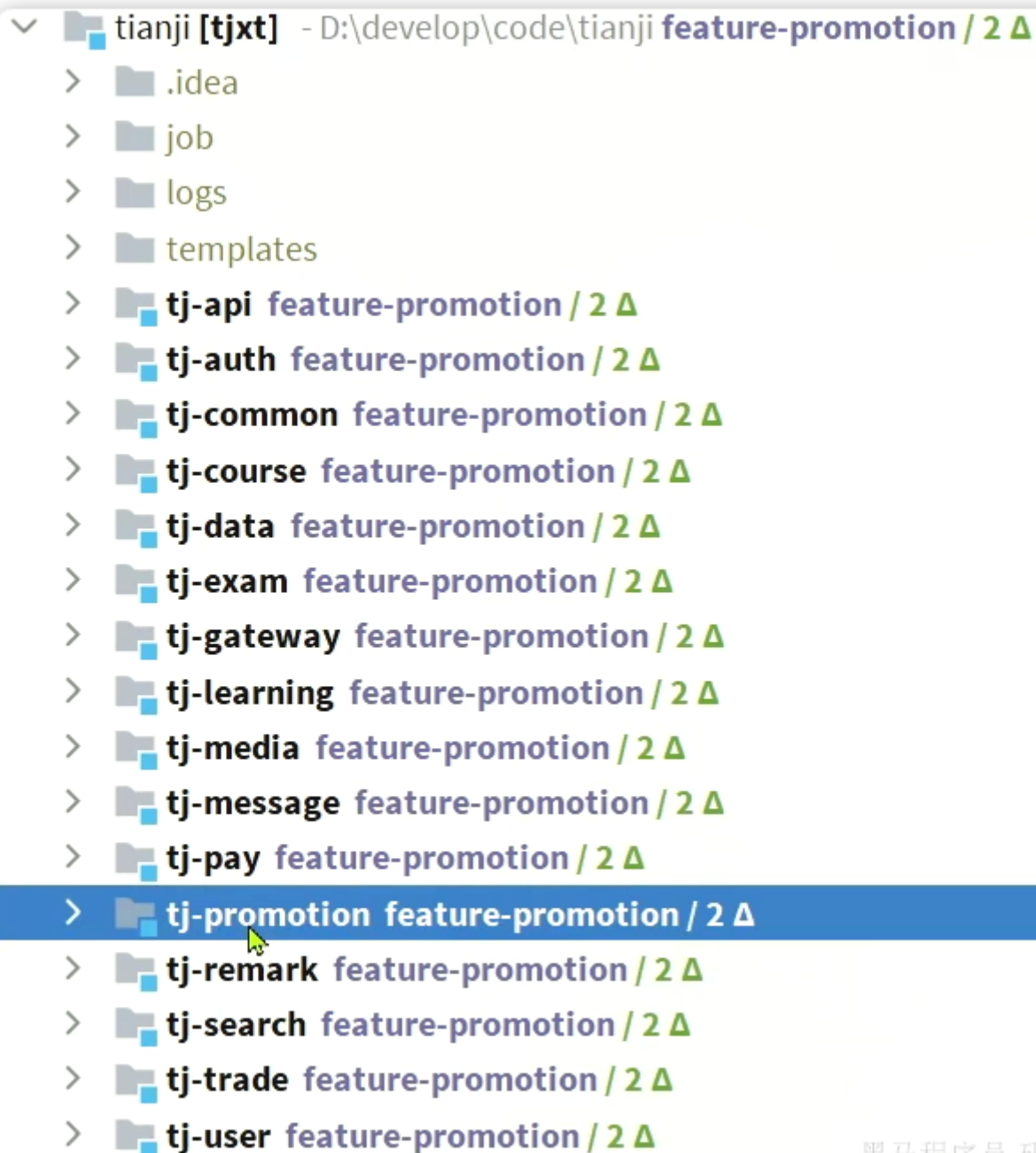
Cancel

Help

黑马程序员-研究院

点击Finish，完成模块创建：





黑马程序员-研究院

### 1.4.2.基础配置

项目创建完毕后，需要引入依赖，POM文件内容如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
5                               http://maven.apache.org/xsd/maven-4.0.0.xsd">
6   <parent>
7     <artifactId>tjxt</artifactId>
8     <groupId>com.tianji</groupId>
```

```
8         <version>1.0.0</version>
9     </parent>
10    <modelVersion>4.0.0</modelVersion>
11
12    <artifactId>tj-promotion</artifactId>
13
14    <properties>
15        <maven.compiler.source>11</maven.compiler.source>
16        <maven.compiler.target>11</maven.compiler.target>
17    </properties>
18    <dependencies>
19        <!--auth-sdk-->
20        <dependency>
21            <groupId>com.tianji</groupId>
22            <artifactId>tj-auth-resource-sdk</artifactId>
23            <version>1.0.0</version>
24        </dependency>
25        <!--api-->
26        <dependency>
27            <groupId>com.tianji</groupId>
28            <artifactId>tj-api</artifactId>
29            <version>1.0.0</version>
30        </dependency>
31        <!--web-->
32        <dependency>
33            <groupId>org.springframework.boot</groupId>
34            <artifactId>spring-boot-starter-web</artifactId>
35        </dependency>
36        <!--mybatis-->
37        <dependency>
38            <groupId>com.baomidou</groupId>
39            <artifactId>mybatis-plus-boot-starter</artifactId>
40        </dependency>
41        <dependency>
42            <groupId>mysql</groupId>
43            <artifactId>mysql-connector-java</artifactId>
44        </dependency>
45        <!--Redis-->
46        <dependency>
47            <groupId>org.springframework.boot</groupId>
48            <artifactId>spring-boot-starter-data-redis</artifactId>
49        </dependency>
50        <!--redisson-->
51        <dependency>
52            <groupId>org.redisson</groupId>
53            <artifactId>redisson</artifactId>
54        </dependency>
```

```
55     <!--discovery-->
56     <dependency>
57         <groupId>com.alibaba.cloud</groupId>
58         <artifactId>spring-cloud-starter-alibaba-nacos-
discovery</artifactId>
59     </dependency>
60     <!--config-->
61     <dependency>
62         <groupId>com.alibaba.cloud</groupId>
63         <artifactId>spring-cloud-starter-alibaba-nacos-config</artifactId>
64     </dependency>
65     <!--caffeine本地缓存-->
66     <dependency>
67         <groupId>com.github.ben-manes.caffeine</groupId>
68         <artifactId>caffeine</artifactId>
69     </dependency>
70     <!--xxl-job-->
71     <dependency>
72         <groupId>com.xuxueli</groupId>
73         <artifactId>xxl-job-core</artifactId>
74     </dependency>
75     <!--loadbalancer-->
76     <dependency>
77         <groupId>org.springframework.cloud</groupId>
78         <artifactId>spring-cloud-starter-loadbalancer</artifactId>
79     </dependency>
80 </dependencies>
81 <build>
82     <finalName>${project.artifactId}</finalName>
83     <plugins>
84         <plugin>
85             <groupId>org.springframework.boot</groupId>
86             <artifactId>spring-boot-maven-plugin</artifactId>
87             <executions>
88                 <execution>
89                     <goals>
90                         <goal>build-info</goal>
91                     </goals>
92                 </execution>
93             </executions>
94             <configuration>
95
96         <mainClass>com.tianji.promotion.PromotionApplication</mainClass>
97     </plugin>
98 </plugins>
99 </build>
```

然后是配置文件:



首先是bootstrap.yml:

```
1 server:
2   port: 8092  #端口
3   tomcat:
4     uri-encoding: UTF-8  #服务编码
5 spring:
6   profiles:
7     active: dev
8   application:
9     name: promotion-service
10  cloud:
11    nacos:
12      config:
13        file-extension: yaml
14        shared-configs: # 共享配置
15          - data-id: shared-spring.yaml # 共享spring配置
16            refresh: false
17          - data-id: shared-redis.yaml # 共享redis配置
18            refresh: false
19          - data-id: shared-mybatis.yaml # 共享mybatis配置
20            refresh: false
21          - data-id: shared-logs.yaml # 共享日志配置
22            refresh: false
23          - data-id: shared-feign.yaml # 共享feign配置
24            refresh: false
```

```

25         - data-id: shared-xxljob.yaml # 共享mq配置
26         refresh: false
27   tj:
28     swagger:
29       enable: true
30       enableResponseWrap: true
31       package-path: com.tianji.promotion.controller
32       title: 天机课堂 - 促销中心接口文档
33       description: 该服务包含优惠促销有关的功能
34       contact-name: 传智教育·研究院
35       contact-url: http://www.itcast.cn/
36       contact-email: zhanghuyi@itcast.cn
37       version: v1.0
38   jdbc:
39     database: tj_promotion
40   auth:
41     resource:
42       enable: true # 开启登录拦截的功能

```

然后是bootstrap-dev.yml:

```

1  spring:
2    cloud:
3      nacos:
4        server-addr: 192.168.150.101:8848 # nacos注册中心
5        discovery:
6          namespace: f923fb34-cb0a-4c06-8fca-ad61ea61a3f0
7          group: DEFAULT_GROUP
8          ip: 192.168.150.101
9    logging:
10     level:
11       com.tianji: debug

```

然后是bootstrap-local.yml:

```

1  spring:
2    cloud:
3      nacos:
4        server-addr: 192.168.150.101:8848 # nacos注册中心
5        discovery:
6          namespace: f923fb34-cb0a-4c06-8fca-ad61ea61a3f0
7          group: DEFAULT_GROUP
8          ip: 192.168.150.1

```

```
9 logging:
10 level:
11     com.tianji: debug
```

启动类:



The image shows an IDE file explorer with the following structure:

- tj-promotion feature-promotion / 6 Δ
  - src
    - main
      - java
        - com.tianji.promotion
          - PromotionApplication

The file PromotionApplication is highlighted. In the bottom right corner, there is a watermark: 黑马程序员 研究院.

代码如下:

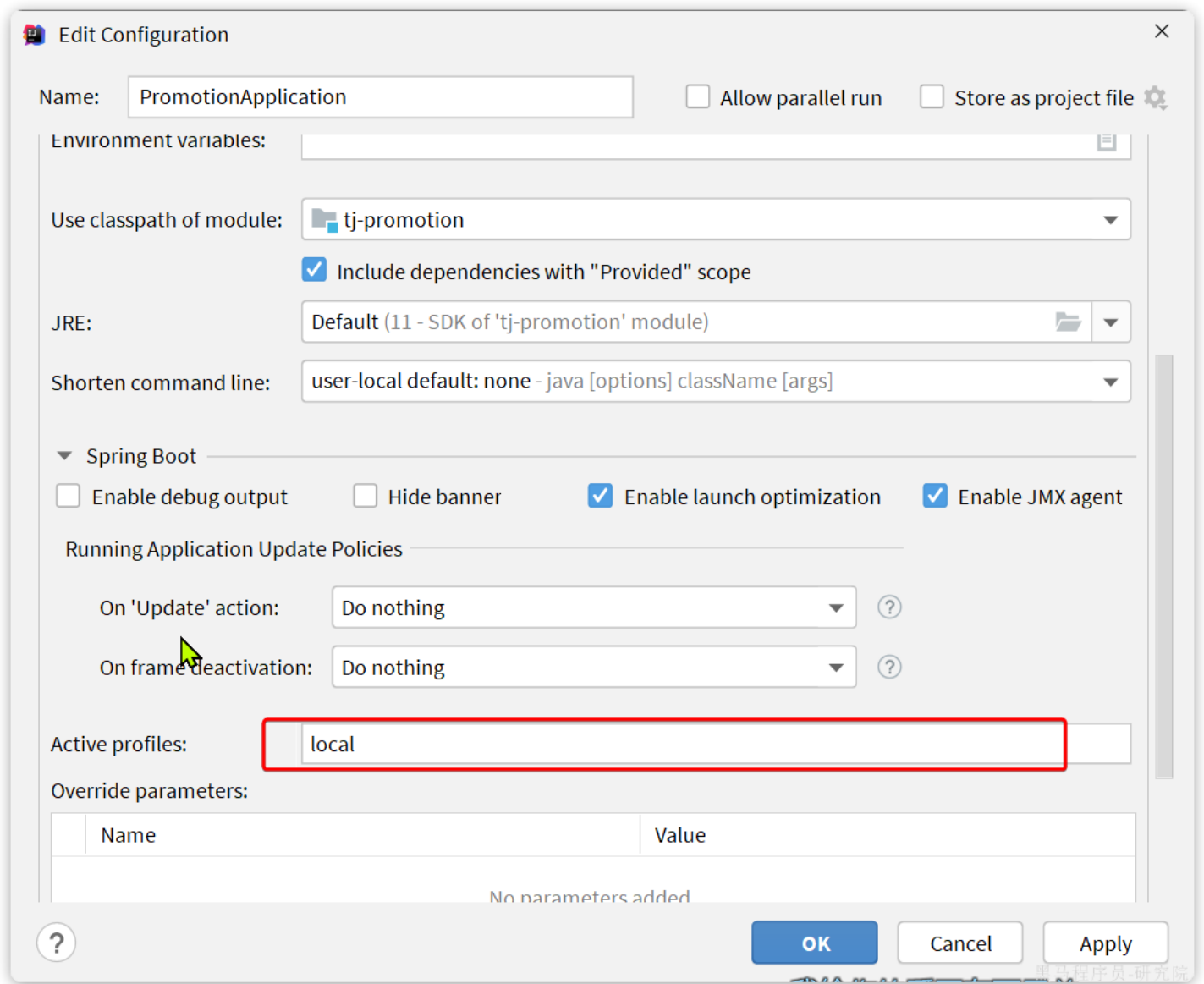
```
1 package com.tianji.promotion;
2
3
4 import lombok.extern.slf4j.Slf4j;
5 import org.mybatis.spring.annotation.MapperScan;
6 import org.springframework.boot.SpringApplication;
7 import org.springframework.boot.autoconfigure.SpringBootApplication;
8 import org.springframework.boot.builder.SpringApplicationBuilder;
9 import org.springframework.core.env.Environment;
10
11 import java.net.InetAddress;
12 import java.net.UnknownHostException;
13
14 @SpringBootApplication
15 @MapperScan("com.tianji.promotion.mapper")
16 @Slf4j
17 public class PromotionApplication {
18     public static void main(String[] args) throws UnknownHostException {
19         SpringApplication app = new
20 SpringApplicationBuilder(PromotionApplication.class).build(args);
21         Environment env = app.run(args).getEnvironment();
22         String protocol = "http";
23         if (env.getProperty("server.ssl.key-store") != null) {
```

```

23         protocol = "https";
24     }
25     log.info("--/\n-----
-----\n\t" +
26             "Application '{}' is running! Access URLs:\n\t" +
27             "Local: \t\t{}://localhost:{}\n\t" +
28             "External: \t{}://{}:{}\n\t" +
29             "Profile(s): \t{}" +
30             "\n-----",
31         env.getProperty("spring.application.name"),
32         protocol,
33         env.getProperty("server.port"),
34         protocol,
35         InetAddress.getLocalHost().getHostAddress(),
36         env.getProperty("server.port"),
37         env.getActiveProfiles());
38     }
39 }

```

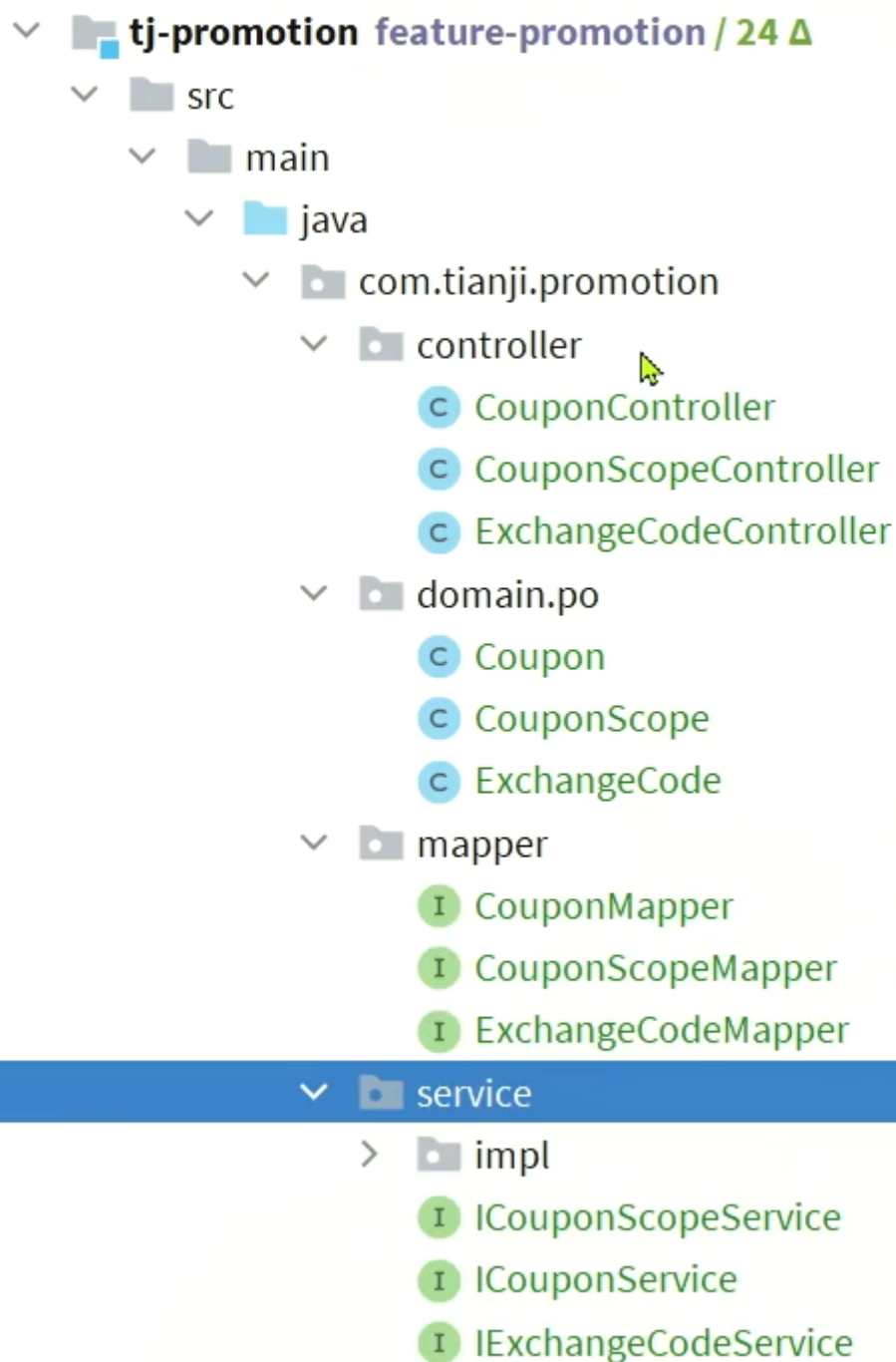
配置启动项，关键是设置运行环境为local：



### 1.4.3.生成代码

然后利用MybatisPlus生成基础代码，这里不再赘述，最终结果：





黑马程序员-研究院

#### 1.4.4.枚举

在优惠券实体中，有很多的类型或状态枚举：

- 折扣类型
- 优惠券状态
- 领取方式

兑换码中也有一个状态字段。

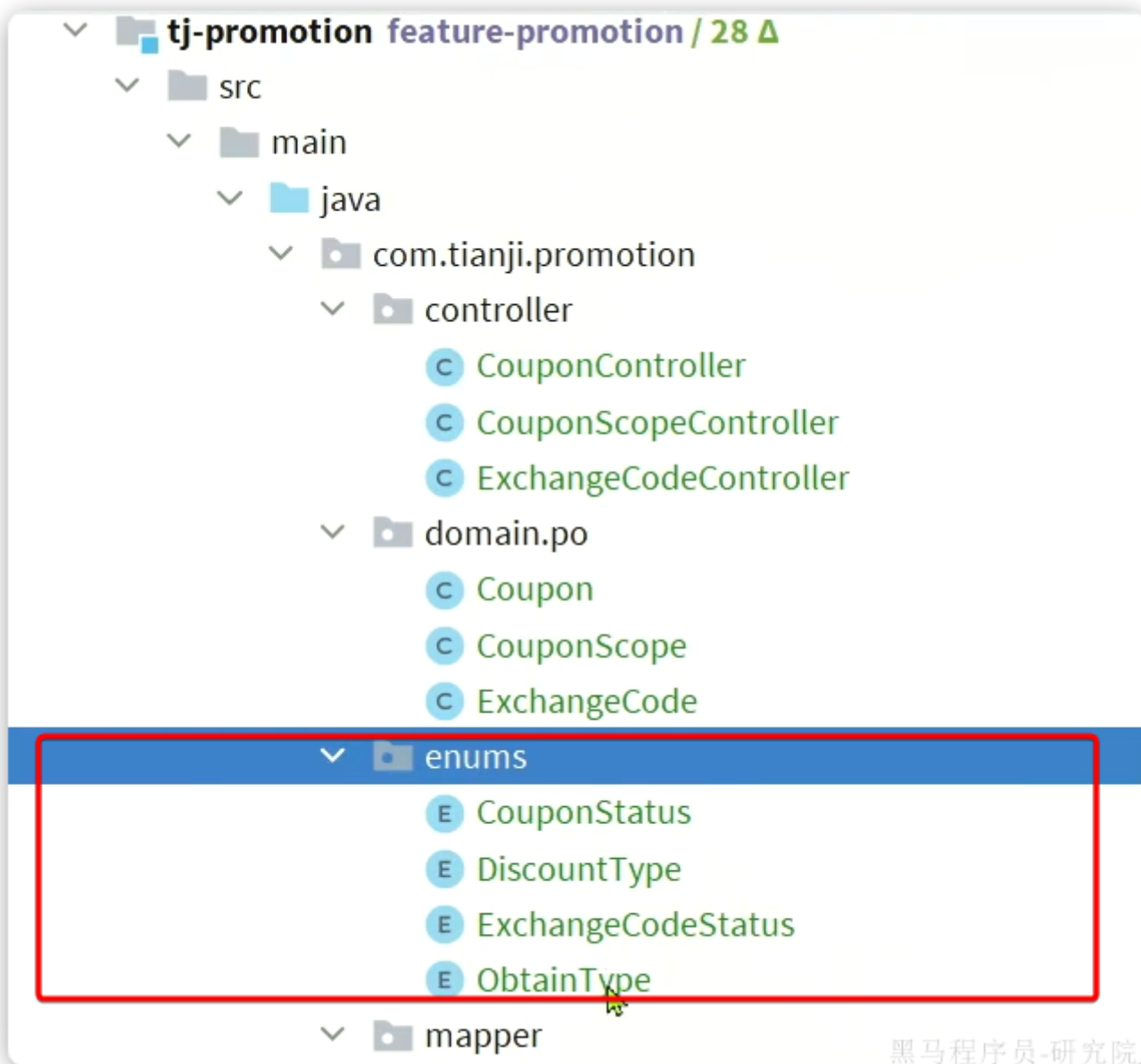
这些都需要定义为枚举，在课前资料中已经给大家准备好了：

i > 新加卷 (D:) > 课程资料 > 天机学堂 > 课件 > day09-优惠券管理 > 资料 > enums >

名称	类型	大小
 CouponStatus.java	Java 源文件	2 KB
 DiscountType.java	Java 源文件	1 KB
 ExchangeCodeStatus.java	Java 源文件	2 KB
 ObtainType.java	Java 源文件	1 KB

黑马程序员-研究院

拷贝到项目中：



最后，别忘了修改PO中对应的字段，首先是 `Coupon` 中：

优惠券类型，1：普通券。目前就一种，保留字段

**private Integer type;**

折扣类型，1：满减，2：每满减，3：折扣，4：无门槛

**private DiscountType discountType;**

金额的限制

**private Integer maxDiscountAmount;**

获取方式：1：手动领取，2：兑换码

**private ObtainType obtainWay;**

优惠券有效期结束时间

**private LocalDateTime termEndTime;**

优惠券配置状态，1：待发放，2：未开始 3：进行中，4：已结束，  
5：暂停

**private CouponStatus status;**

然后是兑换码实体，ExchangeCode：

**private String code;**

兑换码状态，1：待兑换，2：已兑换，3：兑换活动已结束

**private ExchangeCodeStatus status;**

## 2.优惠券管理

优惠券的管理接口有

- 分页查询优惠券列表
- 新增优惠券
- 编辑优惠券
- 查看优惠券（根据id查询优惠券）
- 删除优惠券

大部分都是基本的CRUD，这里我们讲解其中的两个接口：

- 新增优惠券
- 分页查询优惠券

### 2.1.新增优惠券

新增表单原型如图：

## 优惠券基本信息

\*优惠券名称  0/20 1

\*使用范围 ☐ 全部课程分类 ☒ 指定课程分类 2

▼

\*优惠券类型 ☐ 每满减 ☒ 满减 ☐ 无门槛 ☐ 折扣 3

满  元 减  元

## 优惠券推广方式

\*推广方式 ☐ 手动领取 ☒ 指定发放 4

\*发放数量 指定  张 5

\*每人限领 限领  张 6

黑马程序员-研究院

### 2.1.1.接口分析

一个基本的新增接口，按照Restful风格设计即可，关键是请求参数。之前表分析时已经详细介绍过这个页面及其中的字段，这里不再赘述。

需要特别注意的是，如果优惠券限定了使用范围，则需要保存限定的课程分类。而这些信息不再coupon表，而是一张中间关系表：coupon\_scope

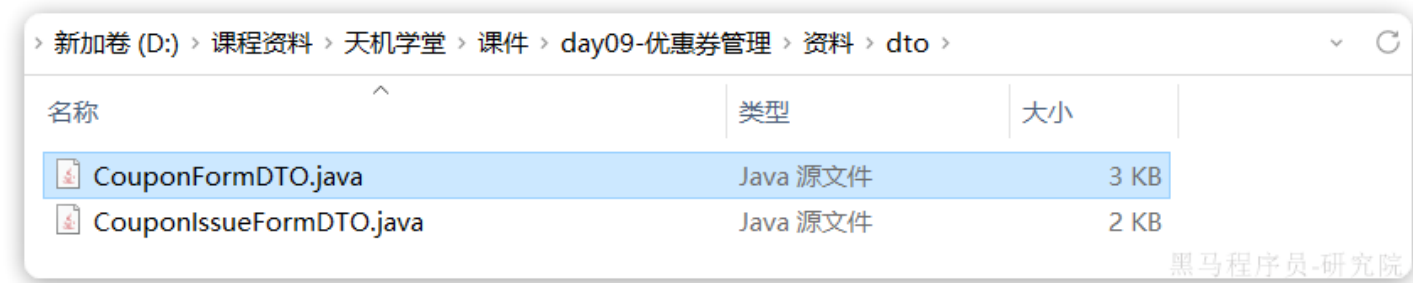
综上，新增优惠券的接口设计如下：

接口说明	新增优惠券功能，如果优惠券有限定使用范围，注意保持范围数据
请求方式	POST
请求路径	/coupons

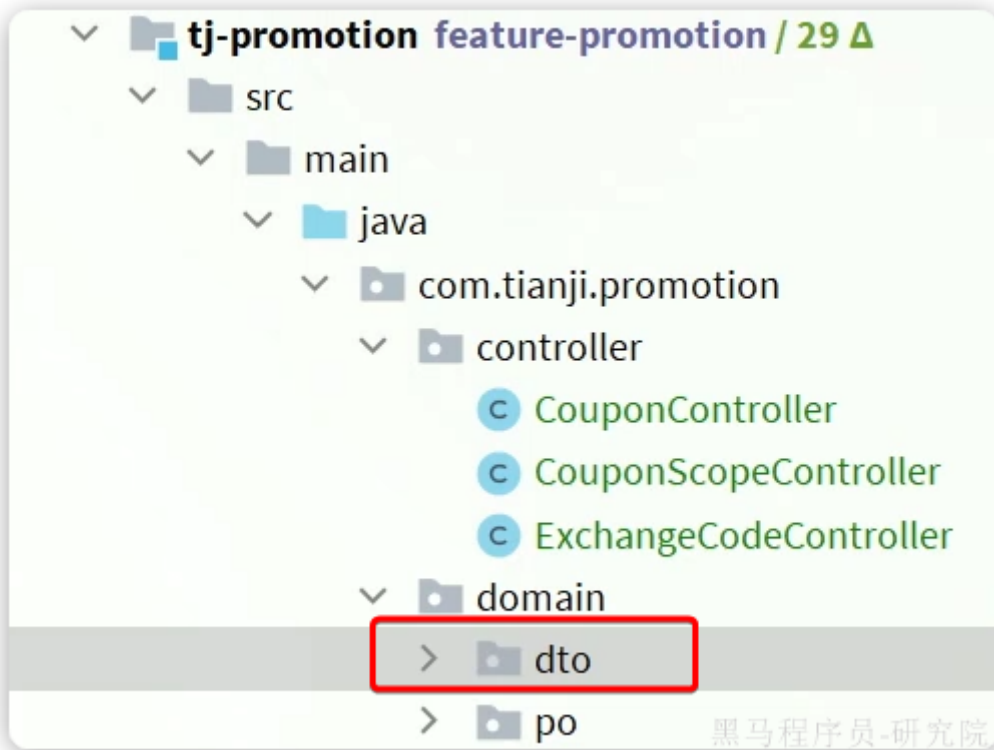
请求参数	<pre>1 { 2     "name" : "新年大促", // 优惠券名称 3     "specific" : true, // 是否限定使用范围 4     "scopes": ["2001", "2004", "2007"], // 限定的分类id       集合 5     "discountType" : 1, // 折扣类型 6     "thresholdAmount" : 100, // 折扣阈值, 0代表无门槛 7     "discountValue" : 15, // 优惠值。满减填优惠金额, 折扣       填折扣值 8     "maxDiscountAmount" : 0, // 最大优惠金额 9     "obtainWay" : 1, // 领取方式, 1-手动领取, 2-兑换码 10    "totalNum" : 200, // 优惠券总发放数量 11    "userLimit" : 1, // 每个人的限领数量 12 }</pre>
返回值	无

### 2.1.2.实体

请求参数比较复杂，需要定义一个对应的Form表单实体。在课前资料已经提供好了：



将其复制到 `com.tianji.promotion.domain.dto` 包下：



需要特别注意的是，Coupon这个PO实体中有一个字段叫 `specific`，这个字段与数据库关键字冲突，需要做特殊处理：



### 2.1.3.接口实现

首先，在tj-promotion模块下的 `com.tianji.promotion.controller.CouponController` 中定义controller接口：

```
1 package com.tianji.promotion.controller;
2
3 import com.tianji.promotion.domain.dto.CouponFormDTO;
4 import com.tianji.promotion.service.ICouponService;
5 import io.swagger.annotations.Api;
6 import io.swagger.annotations.ApiOperation;
7 import io.swagger.annotations.ApiParam;
8 import lombok.RequiredArgsConstructor;
9 import org.springframework.web.bind.annotation.*;
10
11 import javax.validation.Valid;
12 import java.util.List;
13
14 @RestController
15 @RequiredArgsConstructor
16 @RequestMapping("/coupons")
17 @Api(tags = "优惠券相关接口")
18 public class CouponController {
19
20     private final ICouponService couponService;
21
22     @ApiOperation("新增优惠券接口")
23     @PostMapping
24     public void saveCoupon(@RequestBody @Valid CouponFormDTO dto){
25         couponService.saveCoupon(dto);
26     }
27 }
```

接下来，在 `com.tianji.promotion.service.ICouponService` 中定义service方法：

```
1 package com.tianji.promotion.service;
2
3 import com.baomidou.mybatisplus.extension.service.IService;
4 import com.tianji.promotion.domain.dto.CouponFormDTO;
5
6 import java.util.List;
7
8 public interface ICouponService extends IService<Coupon> {
9     void saveCoupon(CouponFormDTO dto);
10 }
```



最后，在 `com.tianji.promotion.service.impl.CouponServiceImpl` 中实现service方法：

```
1 package com.tianji.promotion.service.impl;
2
3 import com.baomidou.mybatisplus.core.conditions.query.LambdaQueryWrapper;
4 import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;
5 import com.tianji.promotion.mapper.CouponMapper;
6 import com.tianji.promotion.service.ICouponScopeService;
7 import com.tianji.promotion.service.ICouponService;
8 import lombok.RequiredArgsConstructor;
9 import org.springframework.stereotype.Service;
10 import org.springframework.transaction.annotation.Transactional;
11
12 import java.util.*;
13 import java.util.stream.Collectors;
14
15 @Service
16 @RequiredArgsConstructor
17 public class CouponServiceImpl extends ServiceImpl<CouponMapper, Coupon>
18     implements ICouponService {
19
20     private final ICouponScopeService scopeService;
21
22     @Override
23     @Transactional
24     public void saveCoupon(CouponFormDTO dto) {
25         // 1.保存优惠券
26         // 1.1.转PO
27         Coupon coupon = BeanUtils.copyBean(dto, Coupon.class);
28         // 1.2.保存
29         save(coupon);
30
31         if (!dto.getSpecific()) {
32             // 没有范围限定
33             return;
34         }
35         Long couponId = coupon.getId();
36         // 2.保存限定范围
37         List<Long> scopes = dto.getScopes();
38         if (CollUtils.isEmpty(scopes)) {
39             throw new BadRequestException("限定范围不能为空");
40         }
41         // 2.1.转换PO
```

```

41         List<CouponScope> list = scopes.stream()
42             .map(bizId -> new
CouponScope().setBizId(bizId).setCouponId(couponId))
43             .collect(Collectors.toList());
44         // 2.2.保存
45         scopeService.saveBatch(list);
46     }
47 }

```

## 2.2.分页查询优惠券

页面原型如图：

优惠券类型

请选择优惠券类型

优惠券状态

请选择优惠券状态

优惠券名称

请输入优惠券关键字

搜索

重置

新增优惠券

过滤条件

优惠券名称	优惠券规则	使用范围	推广方式	使用/领取/发放	创建时间	领用期限	使用期限	状态	操作
秒杀天天有，今天更刺激	每满500元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:22:11	-	-	待发放	<a href="#">编辑</a> <a href="#">发放</a> <a href="#">查看兑换码</a> <a href="#">删除</a>
618年中大促第四波	每满1000元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:21:11	2022.06.7 18:00:00 — 2022.06.10 18:00:00	2022.06.11 18:00:00 — 2022.06.20 18:00:00	已暂停	<a href="#">查看</a> <a href="#">发放</a> <a href="#">查看兑换码</a> <a href="#">删除</a>

### 2.2.1.接口分析

一个典型的带过滤条件的分页查询，非常简单。按照Restful风格设计即可，我们关注的点有两个：

- 请求参数
- 返回值格式

请求参数包含两部分，一个是分页参数，另一个是过滤条件，包含：

- 优惠券的折扣类型
- 优惠券状态
- 优惠券名称关键字

而返回值则如列表中字段所属，需要特别注意的有两点：

- 优惠券规则：这里是对优惠规则的描述，而数据库中保存的是具体的优惠金额；这里我们不组装描述返回，仅仅返回优惠金额信息，由前端自己组织展示即可。

- 使用范围：这里无需展示真正的限定范围，只要告诉前端有没有限定范围即可  
其它字段没什么特殊的，此处不再赘述了。

综上，新增优惠券的接口设计如下：

接口说明	分页查询优惠券，默认按照创建时间排序
请求方式	GET
请求路径	/coupons/page
请求参数	<pre>1 { 2     "pageNo" : 1, // 页码 3     "pageSize" : 10, // 每页大小 4     "type" : 1, // 折扣类型 5     "status" : 1, // 优惠券状态 6     "name" : "大促", // 优惠券名称关键字 7 }</pre>
返回值	<pre>1 { 2     "list": [ 3         { 4             "id": "110", // 优惠券id 5             "name": "年中大促", // 优惠券名称 6             "discountType": 1, // 优惠券折扣类型 7             "thresholdAmount": 100, // 优惠门槛 8             "discountValue": 10, // 优惠值 9             "maxDiscountAmount": 0, // 优惠上限 10            "specific": true, // 是否限定范围 11            "obtainWay": 1, // 领取方式 12            "totalNum": 1000, // 总发放数量 13            "issueNum": 800, // 已领取数量 14            "usedNum": 100 // 已使用数量 15            "createTime": "2023-05-01", // 创建时间 16            "issueBeginTime": "2023-06-01", // 发放开始             时间 17            "issueEndTime": "2023-06-20", // 发放结束             时间 18            "termBeginTime": "2023-06-10", // 使用有效             期开始时间 19            "termEndTime": "2023-06-30", // 使用有效期             结束时间         }     ] }</pre>

```
20         "termDays": 0, // 有效天数
21         "status": 1, // 状态
22     }
23 ],
24     "pages": 0,
25     "total": 0
26 }
```

### 2.1.2. 实体

这里需要两个实体，一个是请求参数QUERY是，一个是返回值VO实体。在课前资料都已经提供给大家了。

QUERY实体：

新加卷 (D:) > 课程资料 > 天机学堂 > 课件 > day09-优惠券管理 > 资料 > query >

名称	类型	大小
 CodeQuery.java	Java 源文件	1 KB
 CouponQuery.java	Java 源文件	1 KB

黑马程序员-研究院

VO实体：

新加卷 (D:) > 课程资料 > 天机学堂 > 课件 > day09-优惠券管理 > 资料 > vo >

名称	类型	大小
 CouponDetailVO.java	Java 源文件	3 KB
 CouponPageVO.java	Java 源文件	3 KB
 CouponScopeVO.java	Java 源文件	1 KB

黑马程序员-研究院

将其复制到 `com.tianji.promotion.domain` 包下的query和vo包下：



### 2.1.3.接口实现

首先，在tj-promotion模块下的 `com.tianji.promotion.controller.CouponController` 中定义controller接口：

```
1 @ApiOperation("分页查询优惠券接口")
2 @GetMapping("/page")
3 public PageDTO<CouponPageVO> queryCouponByPage(CouponQuery query){
4     return couponService.queryCouponByPage(query);
5 }
```

接下来，在 `com.tianji.promotion.service.ICouponService` 中定义service方法：

```
1 PageDTO<CouponPageVO> queryCouponByPage(CouponQuery query);
```

最后，在 `com.tianji.promotion.service.impl.CouponServiceImpl` 中实现service方法：

```
1 @Override
2 public PageDTO<CouponPageVO> queryCouponByPage(CouponQuery query) {
3     Integer status = query.getStatus();
4     String name = query.getName();
5     Integer type = query.getType();
6     // 1. 分页查询
7     Page<Coupon> page = lambdaQuery()
```

```

8         .eq(type != null, Coupon::getDiscountType, type)
9         .eq(status != null, Coupon::getStatus, status)
10        .like(StringUtils.isNotBlank(name), Coupon::getName, name)
11        .page(query.toMpPageDefaultSortByCreateTimeDesc());
12    // 2.处理VO
13    List<Coupon> records = page.getRecords();
14    if (CollUtils.isEmpty(records)) {
15        return PageDTO.empty(page);
16    }
17    List<CouponPageVO> list = BeanUtils.copyList(records, CouponPageVO.class);
18    // 3.返回
19    return PageDTO.of(page, list);
20 }

```

## 3.优惠券发放

优惠券新增之后并不会直接展示在用户端，而是处于一个待发放状态，等待管理员核对信息后，点击方法才行。而发放的方式也分为立刻发放、定时发放两种。

对于定时发放的优惠券，还需要通过定时任务来定期完成发放功能。

另外，由于优惠券的领取方式不同，基于兑换码的优惠券还需要在发放时生成兑换码。

### 3.1.发放优惠券

处于**暂停**状态，或者**待发放**状态的优惠券，在优惠券列表中才会出现发放按钮，可以被发放：

优惠券名称	优惠券规则	使用范围	推广方式	使用/领取/发放	创建时间	领用期限	使用期限	状态	操作
秒杀天天有，今天更刺激	每满500元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:22:11	—	—	待发放	编辑 发放 查看兑换码 删除
618年中大促第四波	每满1000元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:21:11	2022.06.7 18:00:00 — 2022.06.10 18:00:00	2022.06.11 18:00:00 — 2022.06.20 18:00:00	已暂停	查看 发放 查看兑换码 删除
清仓放利，惠到你家	无门槛200元	指定课程分类	指定发放	10/20/100	2022.05.29 22:12:11	2022.06.07 18:00:00 — 2022.06.10 18:00:00	2022.06.11 18:00:00 — 2022.06.20 18:00:00	未开始	查看 暂停 查看兑换码 删除

黑马程序员-研究院

#### 3.1.1.接口分析

当我们点击发放按钮时，会弹出一个表单：

设置领用/使用期限

\*发放方式

☐ 立刻发放

☐ 定时发放

\*使用期限方式

☐ 固定天数

☐ 固定时间段

取消

保存

需要我们选择**发放方式**，**使用期限**。

发放方式分为两种：**立刻发放**和**定时发放**；使用期限也分两种：**固定天数**、**固定时间段**。如图：

设置领用/使用期限

\*发放方式

☒ 立刻发放

☐ 定时发放

\*领用期限

请设置结束时间

\*使用限制方式

☐ 固定天数

☒ 固定时间段

\*使用期限

请设置开始时间 ~ 请设置结束时间

取消

保存

设置领用/使用期限

\*发放方式

☐ 立刻发放

☒ 定时发放

\*领用期限

请设置开始时间 ~ 请设置结束时间

\*使用期限方式

☒ 固定天数

☐ 固定时间段

\*使用期限

请输入天数 天

取消

保存

因此，在提交这个表单时，参数包括：

- 发放（领用）开始时间：如果为空说明是立刻发放，开始时间就是当前时间
- 发放（领用）结束时间
- 有效期天数：如果为空说明是固定有效期
- 使用期限开始时间：如果为空说明是固定天数有效期
- 使用期限结束时间：如果为空说明是固定天数有效期

最后，肯定要带上**优惠券id**，我们才知道发放的是哪张券，当然这个可以通过路径占位符传参。

综上，新增优惠券的接口设计如下：

接口说明	发放优惠券
请求方式	PUT
请求路径	/coupons/{id}/issue
请求参数	<pre>1 { 2     "issueBeginTime": "2023-06-01", // 发放开始时间 3     "issueEndTime": "2023-06-20", // 发放结束时间 4     "termBeginTime": "2023-06-10", // 使用有效期开始时间 5     "termEndTime": "2023-06-30", // 使用有效期结束时间 6     "termDays": 0, // 有效天数 7 }</pre>
返回值	无

### 3.1.2.实体

这里需要一个请求参数的DTO实体，在课前资料中已经提供了：



将其复制到 `com.tianji.promotion.domain.dto` 包即可。

### 3.1.3.接口实现

首先，在tj-promotion模块下的 `com.tianji.promotion.controller.CouponController` 中定义controller接口：

```
1 @ApiOperation("发放优惠券接口")
2 @PutMapping("/{id}/issue")
3 public void beginIssue(@RequestBody @Valid CouponIssueFormDTO dto) {
4     couponService.beginIssue(dto);
5 }
```



接下来，在 `com.tianji.promotion.service.ICouponService` 中定义service方法：

```
1 void beginIssue(CouponIssueFormDTO dto);
```

最后，在 `com.tianji.promotion.service.impl.CouponServiceImpl` 中实现service方法：

```
1
2 @Transactional
3 @Override
4 public void beginIssue(CouponIssueFormDTO dto) {
5     // 1. 查询优惠券
6     Coupon coupon = getById(dto.getId());
7     if (coupon == null) {
8         throw new BadRequestException("优惠券不存在!");
9     }
10    // 2. 判断优惠券状态，是否是暂停或待发放
11    if(coupon.getStatus() != CouponStatus.DRAFT && coupon.getStatus() != PAUSE)
12    {
13        throw new BizIllegalException("优惠券状态错误!");
14    }
15    // 3. 判断是否是立刻发放
16    LocalDateTime issueBeginTime = dto.getIssueBeginTime();
17    LocalDateTime now = LocalDateTime.now();
18    boolean isBegin = issueBeginTime == null || !issueBeginTime.isAfter(now);
19    // 4. 更新优惠券
20    // 4.1. 拷贝属性到PO
21    Coupon c = BeanUtils.copyBean(dto, Coupon.class);
22    // 4.2. 更新状态
23    if (isBegin) {
24        c.setStatus(ISSUING);
25        c.setIssueBeginTime(now);
26    }else{
27        c.setStatus(UN_ISSUE);
28    }
29    // 4.3. 写入数据库
30    updateById(c);
31    // TODO 兑换码生成
32 }
```

### 3.2.兑换码生成算法

优惠券的领取有两种方式：手动领取和指定发放：

优惠券推广方式

\*推广方式

☐ 手动领取

☒ 指定发放

4

\*发放数量

指定

张

5

\*每人限领

限领

张

6

黑马程序员-研究院

指定发放模式是指使用兑换码来兑换优惠券。因此必须在优惠券发放的同时，生成兑换码。兑换码的格式如图：

查看优惠券兑换码

已兑换

未兑换

SGY78HFO	SGY78HFO
SGY78HFO	SGY78HFO

黑马程序员-研究院

但是兑换码该如何生成呢？  
是不是一个简单的字符串就行？

这就本节要讨论的内容了。

#### 3.2.1.兑换码的需求

兑换码并不是简单的一个字符串，它其实有很多的需求：



3.2.2.1.Base32转码

当然可以了，大家思考一下，假如我们将24个字母和8个数字放到数组中，如下：

角标	0	1	2	3	4	5	6	7	8	9	10	11	12	
字符	A	B	C	D	E	F	G	H	J	K	L	M	N	
角标	16	17	18	19	20	21	22	23	24	25	26	27	28	
字符	S	T	U	V	W	X	Y	Z	2	3	4	5	6	

这样，0~31的角标刚好对应了我们的32个字符！而2的5次幂刚好就是32，因此5位二进制数的范围就是0~31

那因此，只要我们让数字转为二进制的形式，然后每5个二进制位为一组，转10进制的结果是不是刚好对应一个角标，就能找到一个对应的字符呢？

这样是不是就把一个数字转为我们想要的字符个数了。这种把二进制数经过加密得到字符的算法就是Base32法，类似的还有Base64法。

举例：假如我们经过自增id计算出一个复杂数字，转为二进制，并每5位一组，结果如下：

```
1 01001 00010 01100 10010 01101 11000 01101 00010 11110 11010
```

此时，我们看看每一组的结果：

- 01001转10进制是9，查数组得字符为：**K**
- 00010转10进制是2，查数组得字符为：**C**
- 01100转10进制是12，查数组得字符为：**N**
- 10010转10进制是18，查数组得字符为：**B**
- 01101转10进制是13，查数组得字符为：**P**
- 11000转10进制是24，查数组得字符为：**2**
- ...

依此类推，最终那一串二进制数得到的结果就是KCNBP2PC84，刚好符合我们的需求。

但是大家思考一下，我们最终要求字符不能超过10位，而每个字符对应5个bit位，因此二进制数不能超过50个bit位。

UUID和Snowflake算法得到的结果，一个是128位，一个是64位，都远远超出了我们的要求。

那自增id算法符合我们的需求呢？

自增id从1增加到Integer的最大值，可以达到40亿以上个数字，而占用的字节仅仅4个字节，也就是32个bit位，距离50个bit位的限制还有很大的剩余，符合要求！

综上，我们可以利用自增id作为兑换码，但是要利用Base32加密，转为我们要求的格式。此时就符合了我们的几个要求了：

- **可读性好**：可以转为要求的字母和数字的格式，长度还不超过10个字符
- **数据量大**：可以应对40亿以上的数据规模
- **唯一性**：自增id，绝对唯一

### 3.2.2.2.重兑校验算法

那重兑问题该如何判断呢？此处有两种方案：

- **基于数据库**：我们在设计数据库时有一个字段就是标示兑换码状态，每次兑换时可以到数据库查询状态，避免重兑。
  - 优点：简单
  - 缺点：对数据库压力大
- **基于BitMap**：兑换或没兑换就是两个状态，对应0和1，而兑换码使用的是自增id.我们如果每一个自增id对应一个bit位，用每一个bit位的状态表示兑换状态，是不是完美解决问题。而这种算法恰好就是BitMap的底层实现，而且Redis中的BitMap刚好能支持 $2^{32}$ 个bit位。
  - 优点：简答、高效、性能好
  - 缺点：依赖于Redis

OK，**重兑、高效**的两个特性都满足了！

现在，就剩下防止爆刷了。我们的兑换码规律性不能太明显，否则很容易被人猜测到其它兑换码。但是，如果我们使用了自增id，那规律简直太明显了，岂不是很容易被人猜到其它兑换码？！

所以，我们采用自增id的同时，还需要利用某种校验算法对id做加密验证，避免他人找出规律，猜测到其它兑换码，甚至伪造、篡改兑换码。

那该采用哪种校验算法呢？

### 3.2.2.3.防刷校验算法

非常可惜，没有一种现成的算法能满足我们的需求，我们必须自己**设计一种算法**来实现这个功能。

不过大家不用害怕，我们可以模拟其它验签的常用算法。比如大家熟悉的JWT技术。我们知道JWT分为三部分组成：

- Header：记录算法
- Payload：记录用户信息
- Verify Signature：验签，用于验证整个token

JWT中的Header和Payload采用的是Base64算法，与我们Base32类似，几乎算是明文传输，难道不怕其他人伪造、篡改token吗？

为了解决这个问题，JWT中才有了第三部分，**验证签名**。这个签名是有一个密钥，结合Header、Payload，利用MD5或者RSA算法生成的。因此：

- 只要密钥不泄露，其他人就无法伪造签名，也就无法伪造token。
- 有人篡改了token，验签时会根据header和payload再次计算签名。数据被篡改，计算的到的签名肯定不一致，就是无效token

因此，我们也可以模拟这种思路：

- 首先准备一个密钥
- 然后利用密钥对自增id做加密，生成签名
- 将签名、自增id利用Base32转码后生成兑换码

只要密钥不泄露，就没有人能伪造兑换码。只要兑换码被篡改，就会导致验签不通过。

当然，这里我们不能采用MD5和RSA算法来生成签名，因为这些算法得到的签名都太长了，一般都是128位以上，超出了长度限制。

因此，这里我们必须采用一种特殊的签名算法。由于我们的兑换码核心是自增id，也就是数字，因此这里我们打算采用按位加权的签名算法：

- 将自增id（32位）每4位分为一组，共8组，都转为10进制

- 每一组给不同权重
- 把每一组数加权求和，得到的结果就是签名

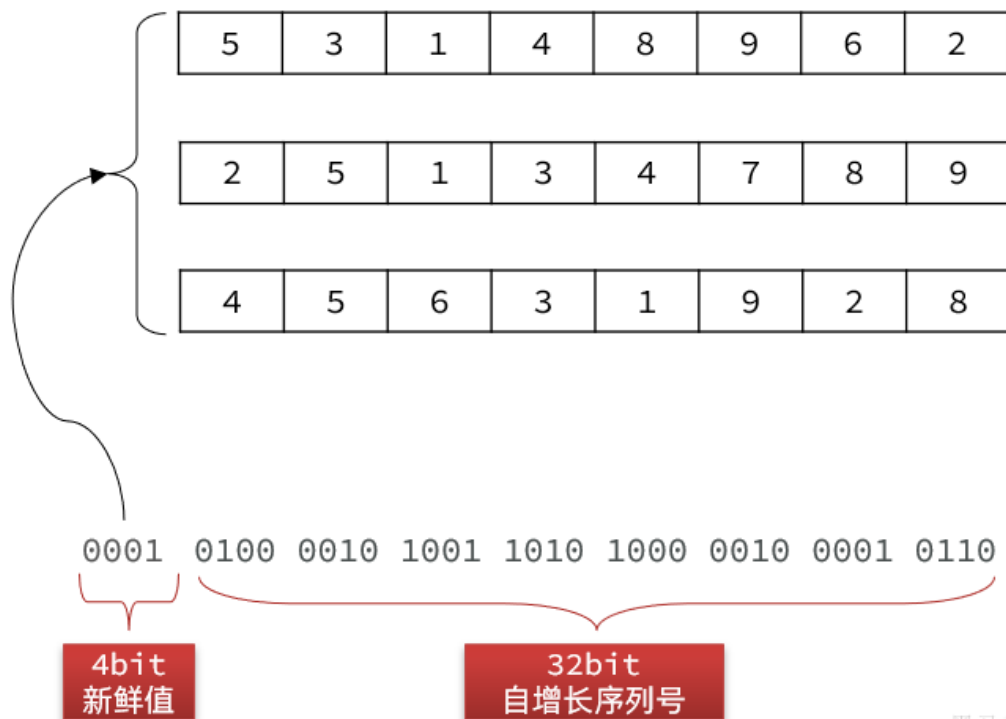
举例：



最终的加权和就是： $4*2 + 2*5 + 9*1 + 10*3 + 8*4 + 2*7 + 1*8 + 6*9 = 165$

这里的权重数组就可以理解为加密的**密钥**。

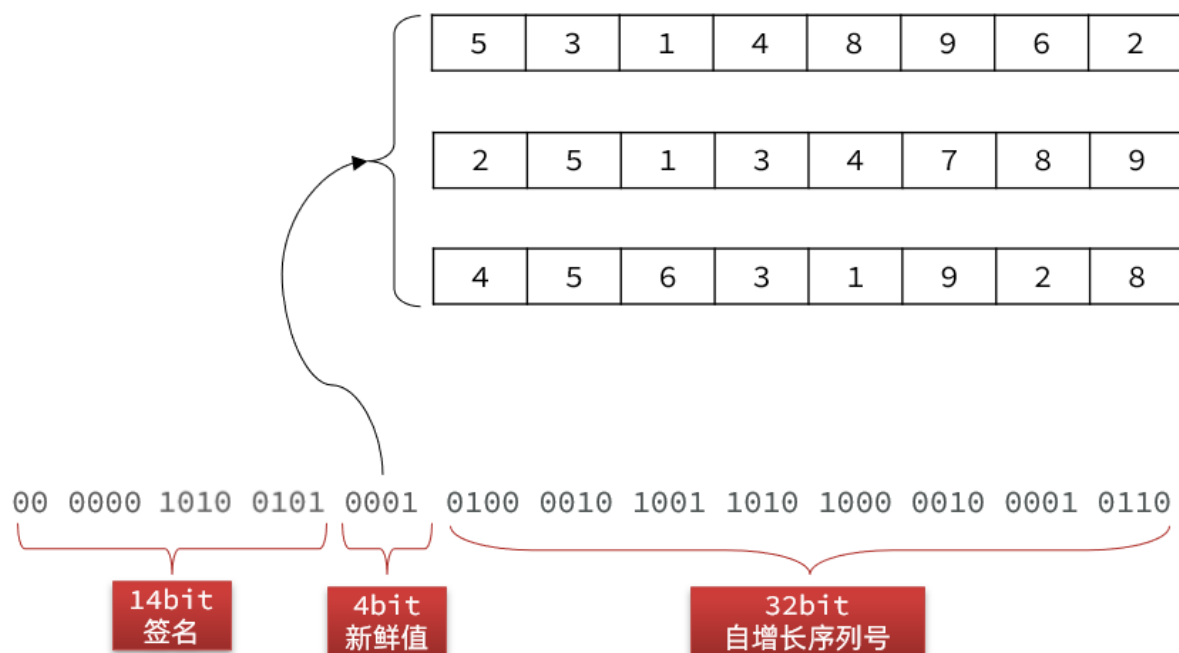
当然，为了避免密钥被人猜测出规律，我们可以准备16组密钥。在兑换码自增id前拼接一个4位的**新鲜值**，可以是随机的。这个值是多少，就取第几组密钥。



黑马程序员-研究院

这样就进一步增加了兑换码的复杂度。

最后，把加权和，也就是签名也转二进制，拼接到最后面，最终的兑换码就是这样：



黑马程序员-研究院

### 3.2.3.算法实现

最终的算法实现在课前资料中已经提供了：



新加卷 (D:) > 课程资料 > 天机学堂 > 课件 > day09-优惠券管理 > 资料 > utils >

名称	类型	大小
 Base32.java	Java 源文件	4 KB
 CodeUtil.java	Java 源文件	7 KB

黑马程序员-研究院

其中：

- Base32.java：是Base32工具类
- CodeUtil.java：是签名工具

我们重点关注CodeUtil的实现，代码如下：

```
1 package com.tianji.promotion.utils;
2
3 import com.tianji.common.constants.RegexConstants;
4 import com.tianji.common.exceptions.BadRequestException;
5
6 /**
7  * <h1 style='font-weight:500'>1.兑换码算法说明：</h1>
8  * <p>兑换码分为明文和密文，明文是50位二进制数，密文是长度为10的Base32编码的字符串 </p>
9  * <h1 style='font-weight:500'>2.兑换码的明文结构：</h1>
10 * <p style='padding: 0 15px'>14(校验码) + 4 (新鲜值) + 32(序列号) </p>
11 * <ul style='padding: 0 15px'>
12 * <li>序列号：一个单调递增的数字，可以通过Redis来生成</li>
13 * <li>新鲜值：可以是优惠券id的最后4位，同一张优惠券的兑换码就会有一个相同标记</li>
14 * <li>载荷：将新鲜值（4位）拼接序列号（32位）得到载荷</li>
15 * <li>校验码：将载荷4位一组，每组乘以加权数，最后累加求和，然后对2^14求余得到</li>
16 * </ul>
17 * <h1 style='font-weight:500'>3.兑换码的加密过程：</h1>
18 * <ol type='a' style='padding: 0 15px'>
19 * <li>首先利用优惠券id计算新鲜值 f</li>
20 * <li>将f和序列号s拼接，得到载荷payload</li>
21 * <li>然后以f为角标，从提前准备好的16组加权码表中选一组</li>
22 * <li>对payload做加权计算，得到校验码 c </li>
23 * <li>利用c的后4位做角标，从提前准备好的异或密钥表中选择一个密钥：key</li>
24 * <li>将payload与key做异或，作为新payload2</li>
25 * <li>然后拼接兑换码明文：f（4位）+ payload2（36位）</li>
26 * <li>利用Base32对密文转码，生成兑换码</li>
27 * </ol>
28 * <h1 style='font-weight:500'>4.兑换码的解密过程：</h1>
```

```

29 * <ol type='a' style='padding: 0 15px'>
30 *     <li>首先利用Base32解码兑换码，得到明文数值num</li>
31 *     <li>取num的高14位得到c1，取num低36位得payload </li>
32 *     <li>利用c1的后4位做角标，从提前准备好的异或密钥表中选择一个密钥：key</li>
33 *     <li>将payload与key做异或，作为新payload2</li>
34 *     <li>利用加密时的算法，用payload2和s1计算出新校验码c2，把c1和c2比较，一致则通过
    </li>
35 * </ol>
36 */
37 public class CodeUtil {
38     /**
39      * 异或密钥表，用于最后的数据混淆
40      */
41     private final static long[] XOR_TABLE = {
42         45139281907L, 61261925523L, 58169127203L, 27031786219L,
43         64169927199L, 46169126943L, 32731286209L, 52082227349L,
44         59169127063L, 36169126987L, 52082200939L, 61261925739L,
45         32731286563L, 27031786427L, 56169127077L, 34111865001L,
46         52082216763L, 61261925663L, 56169127113L, 45139282119L,
47         32731286479L, 64169927233L, 41390251661L, 59169127121L,
48         64169927321L, 55139282179L, 34111864881L, 46169127031L,
49         58169127221L, 61261925523L, 36169126943L, 64169927363L,
50     };
51     /**
52      * fresh值的偏移位数
53      */
54     private final static int FRESH_BIT_OFFSET = 32;
55     /**
56      * 校验码的偏移位数
57      */
58     private final static int CHECK_CODE_BIT_OFFSET = 36;
59     /**
60      * fresh值的掩码，4位
61      */
62     private final static int FRESH_MASK = 0xF;
63     /**
64      * 验证码的掩码，14位
65      */
66     private final static int CHECK_CODE_MASK = 0b11111111111111;
67     /**
68      * 载荷的掩码，36位
69      */
70     private final static long PAYLOAD_MASK = 0xFFFFFFFFFL;
71     /**
72      * 序列号掩码，32位
73      */
74     private final static long SERIAL_NUM_MASK = 0xFFFFFFFFFL;

```

```

75  /**
76   * 序列号加权运算的秘钥表
77   */
78  private final static int[][] PRIME_TABLE = {
79      {23, 59, 241, 61, 607, 67, 977, 1217, 1289, 1601},
80      {79, 83, 107, 439, 313, 619, 911, 1049, 1237},
81      {173, 211, 499, 673, 823, 941, 1039, 1213, 1429, 1259},
82      {31, 293, 311, 349, 431, 577, 757, 883, 1009, 1657},
83      {353, 23, 367, 499, 599, 661, 719, 929, 1301, 1511},
84      {103, 179, 353, 467, 577, 691, 811, 947, 1153, 1453},
85      {213, 439, 257, 313, 571, 619, 743, 829, 983, 1103},
86      {31, 151, 241, 349, 607, 677, 769, 823, 967, 1049},
87      {61, 83, 109, 137, 151, 521, 701, 827, 1123},
88      {23, 61, 199, 223, 479, 647, 739, 811, 947, 1019},
89      {31, 109, 311, 467, 613, 743, 821, 881, 1031, 1171},
90      {41, 173, 367, 401, 569, 683, 761, 883, 1009, 1181},
91      {127, 283, 467, 577, 661, 773, 881, 967, 1097, 1289},
92      {59, 137, 257, 347, 439, 547, 641, 839, 977, 1009},
93      {61, 199, 313, 421, 613, 739, 827, 941, 1087, 1307},
94      {19, 127, 241, 353, 499, 607, 811, 919, 1031, 1301}
95  };
96
97  /**
98   * 生成兑换码
99   *
100  * @param serialNum 递增序列号
101  * @return 兑换码
102  */
103  public static String generateCode(long serialNum, long fresh) {
104      // 1.计算新鲜值
105      fresh = fresh & FRESH_MASK;
106      // 2.拼接payload, fresh (4位) + serialNum (32位)
107      long payload = fresh << FRESH_BIT_OFFSET | serialNum;
108      // 3.计算验证码
109      long checkCode = calcCheckCode(payload, (int) fresh);
110      // 4.payload做大质数异或运算, 混淆数据
111      payload ^= XOR_TABLE[(int) (checkCode & 0b1111)];
112      // 5.拼接兑换码明文: 校验码 (14位) + payload (36位)
113      long code = checkCode << CHECK_CODE_BIT_OFFSET | payload;
114      // 6.转码
115      return Base32.encode(code);
116  }
117
118  private static long calcCheckCode(long payload, int fresh) {
119      // 1.获取码表
120      int[] table = PRIME_TABLE[fresh];
121      // 2.生成校验码, payload每4位乘加权数, 求和, 取最后13位结果

```

```

122     long sum = 0;
123     int index = 0;
124     while (payload > 0) {
125         sum += (payload & 0xf) * table[index++];
126         payload >>= 4;
127     }
128     return sum & CHECK_CODE_MASK;
129 }
130
131 public static long parseCode(String code) {
132     if (code == null || !code.matches(RegexConstants.COUPON_CODE_PATTERN))
133     {
134         // 兑换码格式错误
135         throw new BadRequestException("无效兑换码");
136     }
137     // 1.Base32解码
138     long num = Base32.decode(code);
139     // 2.获取低36位, payload
140     long payload = num & PAYLOAD_MASK;
141     // 3.获取高14位, 校验码
142     int checkCode = (int) (num >>> CHECK_CODE_BIT_OFFSET);
143     // 4.载荷异或大质数, 解析出原来的payload
144     payload ^= XOR_TABLE[(checkCode & 0b11111)];
145     // 5.获取高4位, fresh
146     int fresh = (int) (payload >>> FRESH_BIT_OFFSET & FRESH_MASK);
147     // 6.验证格式:
148     if (calcCheckCode(payload, fresh) != checkCode) {
149         throw new BadRequestException("无效兑换码");
150     }
151     return payload & SERIAL_NUM_MASK;
152 }

```

核心的两个方法：

- `generateCode(long serialNum, long fresh)`：根据自增id生成兑换码。两个参数
  - serialNum：兑换码序列号，也就是自增id
  - fresh：新鲜值，这里建议使用兑换码对应的优惠券id做新鲜值
- `parseCode(String code)`：验证并解析兑换码，返回的是兑换码的序列号，也就是自增id

### 3.3.异步生成兑换码

#### 3.3.1.思路分析

在发放优惠券的时候，如果发现优惠券的领取方式是兑换码方式，则需要生成兑换码。

不过，需要注意的是，**优惠券发放以后是可以暂停的，暂停之后还可以再次发放。**

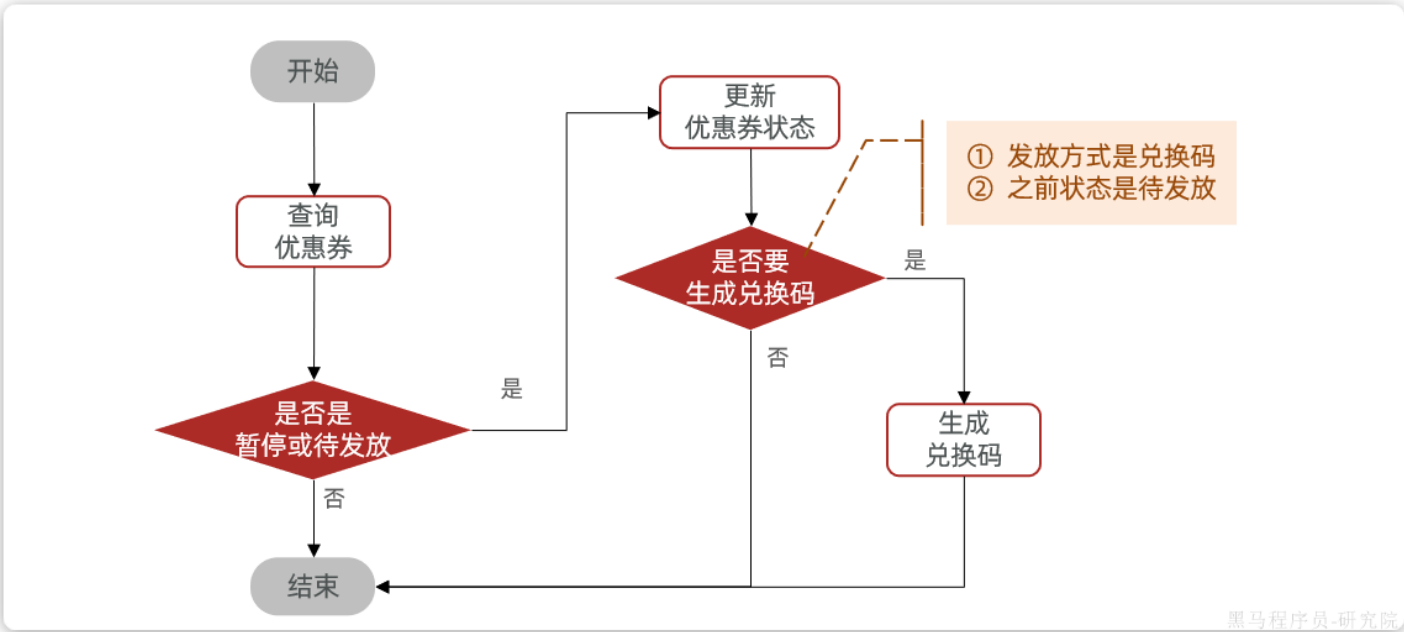
假如一个优惠券是通过兑换码方式领取。第一次发放时我们生产了兑换码，然后被暂停，然后再次发放，如果我们再次生成兑换码，这就重复了。

因此，判断是否需要生成兑换码，要同时满足两个要求：

- 领取方式必须是兑换码方式
- 之前的状态必须是待发放，不能是暂停

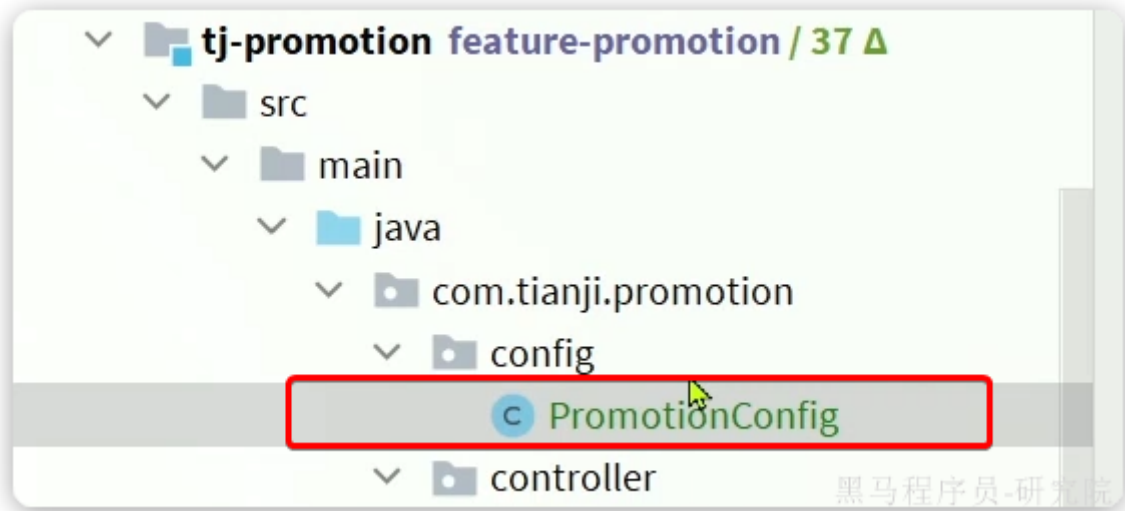
而且，由于生成兑换码的数量较多，可能比较耗时，这里推荐基于线程池异步生成。

流程如下：



#### 3.3.2.代码实现

首先，我们要定义一个线程池，用于异步生成兑换码：



具体代码如下：

```
1 package com.tianji.promotion.config;
2
3 import lombok.extern.slf4j.Slf4j;
4 import org.springframework.context.annotation.Bean;
5 import org.springframework.context.annotation.Configuration;
6 import org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor;
7
8 import java.util.concurrent.Executor;
9 import java.util.concurrent.ThreadPoolExecutor;
10
11 @Slf4j
12 @Configuration
13 public class PromotionConfig {
14
15     @Bean
16     public Executor generateExchangeCodeExecutor(){
17         ThreadPoolTaskExecutor executor = new ThreadPoolTaskExecutor();
18         // 1.核心线程池大小
19         executor.setCorePoolSize(2);
20         // 2.最大线程池大小
21         executor.setMaxPoolSize(5);
22         // 3.队列大小
23         executor.setQueueCapacity(200);
24         // 4.线程名称
25         executor.setThreadNamePrefix("exchange-code-handler-");
26         // 5.拒绝策略
27         executor.setRejectedExecutionHandler(new
28             ThreadPoolExecutor.CallerRunsPolicy());
29         executor.initialize();
30         return executor;
31     }
32 }
```

同时，在启动类添加 `@EnableAsync` 注解，开启异步功能：

```
@EnableAsync
@SpringBootApplication
@MapperScan("com.tianji.promotion.mapper")
@Slf4j
public class PromotionApplication {
    public static void main(String[] args) throws UnknownHostException {
        SpringApplication app = new SpringApplicationBuilder(PromotionApplication.class).build(args);
        Environment env = app.run(args).getEnvironment();
        String protocol = "http";
        if (env.getProperty("server.ssl.key-store") != null) {
            protocol = "https";
        }
    }
}
```

黑马程序员·研究院

然后，我们要改造 `com.tianji.promotion.service.impl.CouponServiceImpl` 中的发放优惠券功能，参见黄色高亮部分：

```
1 private final IExchangeCodeService codeService;
2
3 @Transactional
4 @Override
5 public void beginIssue(CouponIssueFormDTO dto) {
6     // 1. 查询优惠券
7     Coupon coupon = getById(dto.getId());
8     if (coupon == null) {
9         throw new BadRequestException("优惠券不存在！");
10    }
11    // 2. 判断优惠券状态，是否是暂停或待发放
12    if (coupon.getStatus() != CouponStatus.DRAFT && coupon.getStatus() != PAUSE)
13    {
14        throw new BizIllegalException("优惠券状态错误！");
15    }
16    // 3. 判断是否是立刻发放
17    LocalDateTime issueBeginTime = dto.getIssueBeginTime();
18    LocalDateTime now = LocalDateTime.now();
19    boolean isBegin = issueBeginTime == null || !issueBeginTime.isAfter(now);
20    // 4. 更新优惠券
21    // 4.1. 拷贝属性到PO
22    Coupon c = BeanUtils.copyBean(dto, Coupon.class);
23    // 4.2. 更新状态
24    if (isBegin) {
```

```

24         c.setStatus(ISSUING);
25         c.setIssueBeginTime(now);
26     }else{
27         c.setStatus(UN_ISSUE);
28     }
29     // 4.3. 写入数据库
30     updateById(c);
31
32     // 5. 判断是否需要生成兑换码, 优惠券类型必须是兑换码, 优惠券状态必须是待发放
33     if(coupon.getObtainWay() == ObtainType.ISSUE && coupon.getStatus() ==
CouponStatus.DRAFT){
34         coupon.setIssueEndTime(c.getIssueEndTime());
35         codeService.asyncGenerateCode(coupon);
36     }
37 }

```

然后, 在 `com.tianji.promotion.service.IExchangeCodeService` 中添加生成兑换码的方法:

```

1 package com.tianji.promotion.service;
2
3 import com.baomidou.mybatisplus.extension.service.IService;
4 import com.tianji.promotion.domain.po.Coupon;
5 import com.tianji.promotion.domain.po.ExchangeCode;
6
7 /**
8  * <p>
9  * 兑换码 服务类
10  * </p>
11  */
12 public interface IExchangeCodeService extends IService<ExchangeCode> {
13     void asyncGenerateCode(Coupon coupon);
14 }

```

最后, 在 `com.tianji.promotion.service.impl.ExchangeCodeServiceImpl` 中实现该方法:

```

1 package com.tianji.promotion.service.impl;
2
3 import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;
4 import com.tianji.common.utils.CollUtils;
5 import com.tianji.promotion.domain.po.Coupon;
6 import com.tianji.promotion.domain.po.ExchangeCode;

```



```

7 import com.tianji.promotion.mapper.ExchangeCodeMapper;
8 import com.tianji.promotion.service.IExchangeCodeService;
9 import com.tianji.promotion.utils.CodeUtil;
10 import org.springframework.data.redis.core.BoundValueOperations;
11 import org.springframework.data.redis.core.StringRedisTemplate;
12 import org.springframework.scheduling.annotation.Async;
13 import org.springframework.stereotype.Service;
14
15 import java.util.ArrayList;
16 import java.util.List;
17 import java.util.Set;
18
19 import static com.tianji.promotion.constants.PromotionConstants.*;
20
21 /**
22  * <p>
23  * 兑换码 服务实现类
24  * </p>
25  *
26  * @author 虎哥
27  */
28 @Service
29 public class ExchangeCodeServiceImpl extends ServiceImpl<ExchangeCodeMapper,
    ExchangeCode> implements IExchangeCodeService {
30
31     private final StringRedisTemplate redisTemplate;
32     private final BoundValueOperations<String, String> serialOps;
33
34     public ExchangeCodeServiceImpl(StringRedisTemplate redisTemplate) {
35         this.redisTemplate = redisTemplate;
36         this.serialOps = redisTemplate.boundValueOps(COUPON_CODE_SERIAL_KEY);
37     }
38
39     @Override
40     @Async("generateExchangeCodeExecutor")
41     public void asyncGenerateCode(Coupon coupon) {
42         // 发放数量
43         Integer totalNum = coupon.getTotalNum();
44         // 1.获取Redis自增序列号
45         Long result = serialOps.increment(totalNum);
46         if (result == null) {
47             return;
48         }
49         int maxSerialNum = result.intValue();
50         List<ExchangeCode> list = new ArrayList<>(totalNum);
51         for (int serialNum = maxSerialNum - totalNum + 1; serialNum <=
            maxSerialNum; serialNum++) {

```

```
52         // 2.生成兑换码
53         String code = CodeUtil.generateCode(serialNum, coupon.getId());
54         ExchangeCode e = new ExchangeCode();
55         e.setCode(code);
56         e.setId(serialNum);
57         e.setExchangeTargetId(coupon.getId());
58         e.setExpiredTime(coupon.getIssueEndTime());
59         list.add(e);
60     }
61     // 3.保存数据库
62     saveBatch(list);
63
64     // 4.写入Redis缓存, member: couponId, score: 兑换码的最大序列号
65     redisTemplate.opsForZSet().add(COUPON_RANGE_KEY,
66         coupon.getId().toString(), maxSerialNum);
67 }
```

## 4.练习

### 4.1.修改优惠券

修改优惠券接口的请求信息如下：

- 请求方式：PUT
- 请求路径：/coupons/{id}
- 请求参数：与新增类似，参考新增接口。
- 返回值：无

需要注意的页面规则，只有处于待发放状态的优惠券是可以修改的：

状态	操作-查看/编辑	操作-删除			操作-发放/暂停发放
1) 待发放: 新建优惠券后, 优惠券为待发放状态, 用户端查看不到该优惠券;	优惠券状态所对应的操作如下:				
2) 未开始: 已发放, 未到开始领用时间;	1) 查看: 未开始、进行中、已结束、已暂停;	1) 当优惠券为待发放状态下, 可进行删除;	<div>  该优惠券已删除 </div>		
3) 进行中: 当前时间=领用开始时间, 且未过领用截止期限;	点击【查看】跳转到【优惠券详情页】, 优惠券信息无法编辑;	2) 其他状态的优惠券删除按钮置灰, 无法删除	<div> <div>确认删除</div> <div>           此操作将永久删除该优惠券, 是否继续? </div> <div> <div>再想想</div> <div>确认删除</div> </div> </div>		当优惠券为以下状态时, 可进行操作
4) 已结束: 已过领用截止期限,	2) 编辑: 待发放;	3) 点击【删除】出【确认删除】弹窗;			发放: 待发放、已暂停
5) 已暂停 手动点击暂停发放, 状态从进行中变为已暂停; 用户端显示: 优惠券已被抢光, 无法领取	点击【编辑】跳转到【新增优惠券】页面, 面包屑从新增优惠券变成 编辑优惠券;	点击【确认删除】, 出【删除成功】消息提示, 且该优惠券被移除,			暂停发放: 未开始、进行中;
					暂停发放按钮置灰, 无法点击: 已结束

黑马程序员-研究院

## 4.2.删除优惠券

- 请求方式: DELETE
- 请求路径: /coupons/{id}
- 请求参数: 与新增类似, 参考新增接口。
- 返回值: 无

需要注意的页面规则, 只有处于待发放状态的优惠券是可以删除的:

状态	操作-查看/编辑	操作-删除			操作-发放/暂停发放
1) 待发放: 新建优惠券后, 优惠券为待发放状态, 用户端查看不到该优惠券;	优惠券状态所对应的操作如下:				
2) 未开始: 已发放, 未到开始领用时间;	1) 查看: 未开始、进行中、已结束、已暂停;	1) 当优惠券为待发放状态下, 可进行删除;	该优惠券已删除		当优惠券为以下状态时, 可进行操作
3) 进行中: 当前时间=领用开始时间, 且未过领用截止期限;	点击【查看】跳转到【优惠券详情页】, 优惠券信息无法编辑;	2) 其他状态的优惠券删除按钮置灰, 无法删除	确认删除 ×		发放: 待发放、已暂停
4) 已结束: 已过领用截止期限,	2) 编辑: 待发放;	3) 点击【删除】出【确认删除】弹窗;	此操作将永久删除该优惠券, 是否继续?		暂停发放: 未开始、进行中;
5) 已暂停 手动点击暂停发放, 状态从进行中变为已暂停; 用户端显示: 优惠券已被抢光, 无法领取	点击【编辑】跳转到【新增优惠券】页面, 面包屑从新增优惠券变成 编辑优惠券;	点击【确认删除】, 出【删除成功】消息提示, 且该优惠券被移除,	再想想 确认删除		暂停发放按钮置灰, 无法点击: 已结束

黑马程序员-研究院

### 4.3.根据id查询优惠券

#### 4.3.1.接口分析

需求：在管理控制台的优惠券分页列表中，点击某个优惠券或者修改某个优惠券时，都需要根据id查询优惠券的详细信息

优惠券名称	优惠券规则	使用范围	推广方式	使用/领取/发放	创建时间	领用期限	使用期限	状态	操作
秒杀天天有, 今天更刺激	每满500元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:22:11	—	—	待发放	编辑 发放 查看兑换码 删除
618年中大促第四波	每满1000元减200元 最高减200元	全部课程分类	手动领取	0/0/50	2022.05.29 22:21:11	2022.06.7 18:00:00 — 2022.06.10 18:00:00	2022.06.11 18:00:00 — 2022.06.20 18:00:00	已暂停	查看 发放 查看兑换码 删除
清仓放利, 惠到你家	无门槛200元	指定课程分类	指定发放	10/20/100	2022.05.29 22:12:11	2022.06.07 18:00:00 — 2022.06.10 18:00:00	2022.06.11 18:00:00 — 2022.06.20 18:00:00	未开始	查看 暂停 查看兑换码 删除

黑马程序员-研究院

表单回显页面可以参考新增表单。

查看优惠券详情页面如下：

### 优惠券基本信息

优惠券名称: 秒杀天天有, 今天更刺激

使用范围: IT-互联网/人工智能/机器学习 IT-互联网/人工智能/深度学习 IT-互联网/人工智能/强化学习

优惠券规则: 每满 1000 元 减 200 元 最高减 300 元

领用期限: 2022.06.12 18:00:00 ~ 2022.06.15 18:00:00

使用期限: 2022.06.18 18:00:00 ~ 2022.06.28 18:00:00

### 优惠券推广方式

推广方式: 指定发放

发放数量: 指定 100 张

每人限领: 限领 1 张

返回

黑马程序员-研究院

最终的接口信息如下:

接口说明	都需要根据id查询优惠券的详细信息
请求方式	GET
请求路径	/coupons/{id}
请求参数	路径占位符id
返回值	<pre>1 { 2     "id": "110", // 优惠券id 3     "name": "年中大促", // 优惠券名称 4     "discountType": 1, // 优惠券折扣类型 5     "thresholdAmount": 100, // 优惠门槛 6     "discountValue": 10, // 优惠值 7     "maxDiscountAmount": 0, // 优惠上限 8     "specific": true, // 是否限定范围 9     "scopes": [ // 限定的分类 10         {"id": "2001", "name": "IT互联网"} 11     ] 12     "obtainWay": 1, // 领取方式 13     "totalNum": 1000, // 总发放数量 14     "useLimit": 1, // 限领数量 15     "issueBeginTime": "2023-06-01", // 发放开始时间 16     "issueEndTime": "2023-06-20", // 发放结束时间 </pre>

```
17     "termBeginTime": "2023-06-10", // 使用有效期开始时间
18     "termEndTime": "2023-06-30", // 使用有效期结束时间
19     "termDays": 0, // 有效天数
20 }
```

### 4.3.2.接口实现

## 4.4.定时开始发放优惠券

提示：

- 由于发放优惠券时已经把大部分操作完成。因此这里只需要更新优惠券状态，从未开始，修改到进行中即可。
- 不需要把整张表扫描一遍，只要找到那些处于未开始的，并且发放时间早于当前时间的即可。
- 注意利用XXL-JOB的数据分片功能，不要重复处理数据

## 4.5.定时结束发放优惠券

## 4.6.暂停发放

需求：管理员可以将一个发放中的优惠券状态修改为暂停，暂停后学员无法领取或兑换该优惠券。用户端页面也不会展示。

- **请求方式：**PUT
- **请求路径：**/coupons/{id}/pause
- **请求参数：**路径占位符id
- **返回值：**无

注意：只要进行中的优惠券可以被暂停。暂停就是修改优惠券状态，从进行中变为暂停中

## 4.7.查询兑换码

需求：在发放优惠券的时候，如果发现优惠券的领取方式是指定发放，则需要生成兑换码。因此页面有一个查询兑换码功能：

清仓放利，惠到你家	无门槛200元	指定课程分类	指定发放	10/20/100	2022.05.29 22:12:11	2022.06.07 18:00:00 2022.06.10 18:00:00	2022.06.11 18:00:00 2022.06.20 18:00:00	未开始	<a href="#">查看</a> <a href="#">暂停</a> <a href="#">查看兑换码</a> <a href="#">删除</a>
-----------	---------	--------	------	-----------	------------------------	--	--	-----	--

### 4.7.1.接口分析

当我们点击查看兑换码时，就会进入一个兑换码展示页面：

查看优惠券兑换码

已兑换

未兑换

SGY78HFO	SGY78HFO	SGY78HFO	SGY78HFO
FHE98JEH	FHE98JEH	FHE98JEH	FHE98JEH
GJK23MVN	GJK23MVN	GJK23MVN	GJK23MVN
SGY78HFO	SGY78HFO	SGY78HFO	SGY78HFO
FHE98JEH	FHE98JEH	FHE98JEH	FHE98JEH

共30条

<12>

10条/页跳至2页

我知道了

是一个带过滤条件的分页查询。

- 请求方式：GET
- 请求路径：/codes/page
- 请求参数：
  - 分页参数
  - 兑换码状态
  - 有一个隐含条件，就是优惠券id，毕竟查询的是某一个优惠券的兑换码。

- **返回值：**传统分页结果，分页数据保护两个字段：
  - code：兑换码
  - id：兑换码id

#### 4.7.2.接口实现

## 5.面试

面试官：你们优惠券支持兑换码的方式是吧，哪兑换码是如何生成的呢？（请设计一个优惠券兑换码生成方案，可以支持20亿以上的唯一兑换码，兑换码长度不超过10，只能包含字母数字，并且要保证生成和校验算法的高效）



答：

首先要考虑兑换码的验证的高效性，最佳的方案肯定是用自增序列号。因为自增序列号可以借助于BitMap验证兑换状态，完全不用查询数据库，效率非常高。

要满足20亿的兑换码需求，只需要31个bit位就够了，也就是在Integer的取值范围内，非常节省空间。我们就按32位来算，支持42亿数据规模。

不过，仅仅使用自增序列还不够，因为容易被人爆刷。所以还需要设计一个加密验签算法。算法有很多，比如可以使用按位加权方案。32位的自增序列，可以每4位一组，转为10进制，这样就有8个数字。提前准备一个长度为8的加权数组，作为密钥。对自增序列的8个数字按位加权求和，得到的结果作为签名。

当然，考虑到密钥的安全性，我们也可以准备多组加权数组，比如准备16组。然后生成兑换码时随机生成一个4位的新鲜值，取值范围刚好是0~15，新鲜值是几，我们就取第几组加权数组作为密钥。然后把新鲜值、自增序列拼接后按位加权求和，得到签名。

最后把签名值的后14位、新鲜值（4位）、自增序列（32位）拼接，得到一个50位二进制数，然后与一个较大的质数做异或运算加以混淆，再基于Base32或Base64转码，即可的对兑换码。

如果是基于Base32转码，得到的兑换码恰好10位，符合要求。



需要注意的是，用来做异或的大质数、加权数组都属于密钥，千万不能泄露。如有必要，也可以定期更换。

当我们要验签的时候，首先将结果 利用Base32转码为数字。然后与大质数异或得到原始数值。


接着取高14位，得到签名；取后36位得到新鲜值与自增序列的拼接结果。取中4位得到新鲜值。

根据新鲜值找到对应的密钥（加权数组），然后再次对后36位加权求和，得到签名。与高14位的签名比较是否一致，如果不一致证明兑换码被篡改过，属于无效兑换码。如果一致，证明是有效兑换码。

接着，取出低32位，得到兑换码的自增序列号。利用BitMap验证兑换状态，是否兑换过即可。

整个验证过程完全不用访问数据库，效率非常高。

面试官：你在项目中哪些地方用到过线程池？


 答：很多地方，比如我在实现优惠券的兑换码生成的时候。

当我们在发放优惠券的时候，会判断优惠券的领取方式，我们有基于页面手动领取，基于兑换码兑换领取等多种方式。

如果发现是兑换码领取，则会在发放的同时，生成兑换码。但由于兑换码数量比较多，如果在发放优惠券的同时生成兑换码，业务耗时会比较久。

因此，我们会采用线程池异步生成兑换码的方式。

面试官可能会追问：那你的线程池参数是怎么设置的？

 答：线程池的常见参数包括：核心线程、最大线程、队列、线程名称、拒绝策略等。

这里核心线程数我们配置的是2，最大线程数是CPU核数。之所以这么配置是因为发放优惠券并不是高频业务，这里基于线程池做异步处理仅仅是为了减少业务耗时，提高用户体验。所

以线程数无需特别高。

队列的大小设置的是200，而拒绝策略采用的是交给调用线程处理的方式。

由于业务访问频率较低，所以基本不会出现线程耗尽的情况，如果真的出现了，就交给调用线程处理，让客户稍微等待一下也行。