# 53\_代码实战: 尝试给首次登录用户发放优惠券和生日券

#### 1、开篇

上节课带大家从打包、发布、启动服务几个步骤将登录功能演示了一遍。通过查看 日志的方式,了解了代码从登陆、发送事件消息、接受事件消息,到最后更新登陆 次数的执行过程。聊完了登陆之后,本节课我们会梳理互联网教育系统的优惠券管 理模块的业务流程,发放优惠券功能会在登陆之后执行,同样也是通过异步的方式 执行。今天课程的内容包括以下几个部分:

- 定义 ConsumerCoupon 实体类
- 定义 ConsumerMapper 访问数据库
- 修改 LogonListener
- 定义 ConsumerController

## 2、定义 ConsumerCoupon 实体类

既然是对优惠券的操作那么首先需要定义一个 Coupon 类,如图 1 所示,在 spring/dto 下面建立 Coupon 类,主要用来描述优惠券的基本信息。包括优惠券 ld、优惠券明名称、状态、有效天数、金额、优惠券总数。

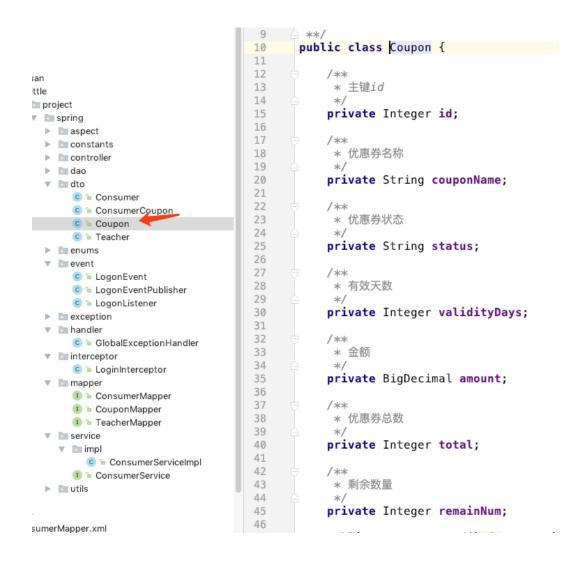


图 1 Coupon 类定义

定义完优惠券的基本信息以后,还需要定义优惠券与消费者的关联信息,毕竟优惠券如果要起到作用需要和消费者产生关系。如图 2 所示,在

spring/ConsumerCoupon 下面定义 ConsumerCoupon 类,该类用来描绘优惠券和消费者之间的关系。包括消费者 Id、优惠券 Id、消费者名称、优惠券名称、状态、领取金额、领取时间和失效时间等。

```
ConsumerCoupon
                                               16
■ main
                                              17
                                                     public class ConsumerCoupon {
 iava 🚞
                                              18
 ▼ 🛅 com
                                              19
                                                           * 主键id
   ruyuan
                                              20
      ▼ 🛅 little
                                              21
                                                          private Integer id;
          project
          spring
                                                           * 消费者id
            aspect
                                              24
            constants
                                              25
                                                          private Integer consumerId;
            controller
                                              26
            ▶ □ dao
                                              27
                                                           * 优惠券id
            ▼ 🛅 dto
                                              28
                 Consumer
                                              29
                                                          private Integer couponId;
                 ConsumerCoupon
                                              30
                 c 🔓 Coupon
                                              31
                                                           * 消费者名称
                 C Teacher
                                              32
            ▶ menums
                                              33
                                                          private String consumerName;
              event
                                              34
                 c LogonEvent
                                              35
                                                           * 优惠券名称
                 C LogonEventPublisher
                                              36
                 © 🖫 LogonListener
                                              37
                                                          private String couponName;
            exception

▼ Imandler

                                                           * 状态 {@link ConsumerCouponStatusEnum}
                                              30
                 © % GlobalExceptionHandler
                                              40
             interceptor
                                              41
                                                          private String status;
                 © 🖫 LoginInterceptor
                                              42
            mapper
                                              43
                                                           * 金额
                 ConsumerMapper
                                              44
                 CouponMapper
                                              45
                                                          private BigDecimal amount;
                 1 TeacherMapper
                                              46

▼ liservice

                                                           * 领取时间
                                              47
               ▼ 🛅 impl
                                              48
                   © 'a ConsumerServiceImpl
                                              49
                                                          private String createTime;
                 ConsumerService
                                              50
            ▶ till utils
                                              51
                                                           * 失效时间
resources
                                              52
   mapper
                                                          private String expireTime;
                                              53
      consumerMapper.xml
                                              54
      couponMapper.xml
                                                          public ConsumerCoupon() {
                                              55
      teacherMapper.xml
                                              56
   application.properties
                                                          }
                                              57
   ijdbc.properties
```

图 2 优惠券与消费者关联关系

## 3、定义 ConsumerMapper 访问数据库

定义完优惠券和优惠券关联关系之后,就需要对优惠券进行数据库的操作。这里需要定义 Mybatis 的数据库访问接口和操作数据库的 Mapper 文件。先来描述数据库的访问接口,它是应用程序调用 Mybatis 的入口。如图 3 所示,在 spring/mapper下面建立 CouponMapper 的接口文件,由于它是一个接口文件因此在其中定义的都是接口方法。包括添加优惠券信息、消费者获取优惠券、通过消费 Id 和时间查找优惠券以及更新优惠券信息。

```
CouponMapper updateStatusById()
                                       32
                                       33
                                                   * 添加优惠券
ava
                                       34
com
                                       35
                                                   * @param coupon 优惠券信息
ruyuan 🖿 🖿
                                       36
                                                   * @return 操作结果
 ▼ 🛅 little
                                       37
    project
                                       38
                                                  int add(Coupon coupon);
      ▼ lim spring
                                       39
        aspect
                                       40
                                                   * 消费者获取优惠券
        constants
                                       41
        controller
                                       42
                                                   * @param consumerCoupon 消费者优惠券信息
        ▶ lim dao
                                       43
                                                   * @return 操作结果
        ▶ b dto
        ▶ ■ enums
                                       45
                                                  int receiveCoupon(ConsumerCoupon consumerCoupon);
        ▶ ■ event
                                       46
        exception
                                       47
                                                   * 获取指定消费者id的指定年月的优惠券总数
        handler
                                       48
        ▶ interceptor
                                       49
                                                   * @param id
        ▼ Imapper mapper
                                                   * @param time 年月, 格式 yyyy-MM-dd
                                       50
            ConsumerMappe
                                       51
                                                   * @return 查询数量结果
            CouponMapper
                                       52
            TeacherMapper
                                       53
                                                  int countByIdAndLikeTime(@Param("id") int id,

▼ lim service

                                       54
                                                                             @Param("time") String time);
          ▼ limpl
                                       55
              ConsumerServiceImpl
                                       56
                                                   * 根据消费者优惠券id修改状态
            ConsumerService
                                       57
        ▶ b utils
                                       58
                                                   * @param id
                                                                    优惠券id
esources
                                       59
                                                  * @param status 状态
mapper mapper
                                       60
                                                   * @return 操作结果
  consumerMapper.xml
                                       61
  couponMapper.xml
                                       62
                                                  int updateStatusById(@Param("id") Integer id,
  teacherMapper.xml
                                       63
                                                                         @Param("status") String status);
application.properties
                                       64
```

图 3 定义 CouponMapper 的接口文件

由于 CouponMapper 是一个接口文件,为了实现优惠券信息的真实访问需要实体类去实现该接口。如图 4 所示,在 resources/mapper 目录下面定义一个 couponMapper.xml 文件,这个文件就完成 CouponMapper 接口的具体实现的。 其中可以看到 findByld 方法,就是通过传入的优惠券 ld 到数据库中的 t\_coupon 表中查询优惠券信息的。

```
idea

src

src

siava

siava

sittt

sittt

sittt

sittt

sittt
                                                    <?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.ruyuan.little.project.spring.mapper.CouponMapper">
                                                          <select id="findById" resultType="com.ruyuan.little.project.spring.dto.Coupon">
    select * from t_coupon where id = #{id}
</select>
                ▼ little
▼ project
▼ spring
                                                          aspect
                         ▶ □ constants
                         dao dto
                         ▶ menums
                         ▶ ■ event
                                                                <if test="couponName != null and couponName !=''">
And coupon_name = #{couponName}
</if>
                           intercepto

    ▼ Imapper

                                                                </fr>
And phone = #{phone}
                              1 a Consi
                                                               <if test="openId != null and openId !=''">
And open_id = #{openId}
</if>

▼ lim service

And credits = #{credits}

                            ▼ 🛅 impl

    □ Cons

       ▼ mresources
                                                                </fr>
<if test="orderTimes != null and orderTimes !='' and orderTimes >0">
And order_times = #{orderTimes}
</if>
           mapper
               consumerMapper.xml
                                                               </ir>
<if test="totalPrices != null and totalPrices !='' and totalPrices >0">
   And total_prices = #{totalPrices}
</if>
<if test="sex != null and sex !=''">
   And sex = #{sex}
</if></if></or>
                teacherMapper.xml
             application.properties
             idbc.properties
                                           35
36
37
38
39
40
41
42
43
44
             amybatis-config.xml
                                                                </if>
<if test="birthday != null and birthday !=''">
And birthday = #{birthday}
</if>
             spring-jdbc.xml
                                                                 limit #{startIndex},#{endIndex}
     alittle-project-spring.iml
                                                          <insert id="add" parameterType="com.ruyuan.little.project.spring.dto.Coupon">
    INSERT INTO t_coupon
     README.md
 External Libraries
                                                                      (id, coupon_name, phone, credits, sex, birthday, order_times,
```

图 4 couponMapper.xml 根据优惠券 ld 查找优惠券信息

既然能够查询到优惠券,那么也可以将某个优惠券与消费者进行关联。如图 5 所示,依然是在 couponMapper.xml 中定义了一个 receiveCoupon 的实现,实际上就是一个 insert 语句,在 t\_consumer\_coupon 这张表中插入一条记录。从这个表的名字上看,这个就是记录优惠券和消费者之间的关联表。插入的信息包括消费者 ld、优惠券 ld、优惠券名称、消费者名称等相关信息。

```
<insert id="receiveCoupon" parameterType="com.ruyuan.little.project.spring.dto.ConsumerCoupon"</pre>
        useGeneratedKeys="true" keyProperty="id">
    INSERT INTO
    t_consumer_coupon(
    consumer_id,
    consumer_name,
    coupon_id,
    coupon_name,
    status,
    amount,
    create_time,
    expire_time)
    values(
    #{consumerId},
    #{consumerName},
    #{couponId},
    #{couponName},
    #{status},
    #{amount},
    #{createTime}
    #{expireTime})
</insert>
```

图 5 couponMapper.xml 让消费者获取优惠券

#### 4、修改 LogonListener

说完了优惠券相关的实体以及数据库访问代码之后,在来看看在哪里需要调用优惠券方法的代码。从前面几节课可以知道,优惠券的发放是伴随在登陆之后,通过异步的方式进行的。在系统中监听登陆消息的代码就在 LogonListener 中,这里对其进行修改。如图 6 所示,修改 LogonListener 类中的 dispatch 方法,加入登陆发放优惠券和生日优惠券的代码。其中登陆优惠券的代码需要判断是否第一次登陆,如果是第一次就调用 distributeCoupon 方法发放优惠券。对与生日优惠券而言,如果消费定义了生日信息,首先判断在生日当月是否已经领取了优惠券,如果没有领取对其进行优惠券发放的操作。

```
▼ 🛅 spring
                                                                                                                                         65
66
67
                                                                                                                                                                                     @EventListener(LogonEvent.class)
                   ▶ ■ aspect
                    constants
                                                                                                                                                                                    public void dispatch(LogonEvent event) {
    LOGGER.info("监听到登录事件event:{}", event);
    // 模拟异步场景
                                                                                                                                        68
69
70
71
72
73
74
75
76
77
78
80
81
82
83
84
85
86
87
88
99
91
92
93
94
95
97
                    ▶ □ dao
                    ▶ 🛅 dto
                                                                                                                                                                                                   // MEMATERIALS
try {
    Thread.sleep( millis: 2000);
} catch (InterruptedException e) {
    e.printStackTrace();

                            enums

▼ limevent

                          © % LogonEvent
© % LogonEventPublisher
© % LogonListener
                                                                                                                                                                                                      Consumer consumer = event.getConsumer();
                           exception
                    ► handler
► interceptor
                                                                                                                                                                                                                  首次登录事件、分发优惠券
                                                                                                                                                                                                                 ▶ ■ mapper
                     ▶ service
                                                                                                                                                                                                                      this.distributeCoupon(consumer, loginCouponId);
                     ▶ utils
pper
consumerMapper.xml
                                                                                                                                                                                                  couponMapper.xml
 teacherMapper.xml
 olication.properties
 c.properties
back.properties
back.xml
batis-config.xml
                                                                                                                                                                                                                                     LocalDate.now().Tormat(batelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimerolmatelimer
 ing-jdbc.xml
ing-web.xml
-spring.iml
                                                                                                                                                                                                                      }
```

图 6 修改 LogonListener

再来看看 distributeCoupon 方法是如何实现发放优惠券的,如图 7 所示,该方法通过 Mybatis 调用 couponMapper 中的 receiveCoupon 方法传入 consumerCoupon 对象,直接操作数据库对优惠券与消费者之间的关系进行更新,从而实现优惠券的发放。

```
/**

* 分发优惠券

* @param consumer 消费者

* @param couponId 优惠券id

*/

private void distributeCoupon(Consumer consumer, Integer couponId) {
    Coupon coupon = couponMapper.findById(couponId);
    ConsumerCoupon consumerCoupon = new ConsumerCoupon(consumer, coupon);
    LOGGER.info("发放优惠券{}",consumerCoupon);
    couponMapper.receiveCoupon(consumerCoupon);
}
```

图 7 distributeCoupon 方法

#### 5、定义 ConsumerController

除了优惠券发放的代码之外,还需要有查询优惠券的接口。在领取优惠券以后,用户依旧可以通过 ConsuermController 对优惠券信息进行查询。如图 8 所示,在 spring/controller 下面建立 ConsumerController 类分别建立 findCouponByConsumerId 方法传入优惠券 Id 对优惠券进行查找;

findCouponByConsumerIdAndStatus 通过消费者 Id 和状态进行查找等方法。

```
/**
* 根据消费者id查找优惠券
a com
Truyuan

                                                                                                                                                                                                         @param id 消费者id
                                                                                                                                                                                                         @return 结果
                                                                                                                                                  61
                                                                                                                                                                                             @GetMapping("/allCoupon/{id}")
                                                                                                                                                 public CommonResponse findCouponByConsumerId(@PathVariable Integer id) {
   List<ConsumerCoupon> consumerCoupons = consumerService.findCouponByConsumerId(id);
                                 ► constants

controller
                                                                                                                                                                                                               return getCommonResponse(consumerCoupons);
                              © % ConsumerController
                                                   © 1 JdbcDemoController
                                                                                                                                                                                               /**
* 根据消费者id和优惠券状态查找优惠券
* @param id 消费老id
                                dao
dto
enums
event
                                                                                                                                                                                                  * Oparam id 消费者id
* Oparam status 优惠券状态
Oreturn 结果
                                  exception
                                  ▶ □ handler
                                                                                                                                                                                                           stMapping("/coupon")
lic CommonResponse findCouponByConsumerIdAndStatus(Integer id,String status) {
   List<ConsumerCoupon> consumerCoupons = consumerService.findCouponByConsumerIdAndStatus(id,status);
                                  ▶ interceptor
                                  ▶ Imapper
                                 ▶ butils
     mapper
couponMapper.xml
teacherMapper.xml
                                                                                                                                                                                                             return getCommonResponse(consumerCoupons);
                                                                                                                                                                                                  * 获取优惠券公共返回实体
                                                                                                                                                                                                         <u>@param</u> consumerCoupons 用户优惠券
<u>@return</u> 结果
       application.properties
       jdbc.properties
        logback.properties
                                                                                                                                                                                            private CommonResponse getCommonResponse(List<ConsumerCoupon> consumerCoupons) {
    CommonResponse<TableData<ConsumerCoupon>> commonResponse = new CommonResponse<>();
        mybatis-config.xml
spring-jdbc.xml
spring-web.xml
                                                                                                                                                                                                             TableData<ConsumerCoupon> tableData = new TableData<>();
tableData.setTotal((long) consumerCoupons.size());
tableData.setRows(consumerCoupons);
commonResponse.setData(tableData);
      ect-spring.iml
                                                                                                                                                                                                              return commonResponse;
       aries
                                                                                                                                                                          }
```

图 8 定义 ConsumerController

#### 6、总结

本节课围绕优惠券发放的代码实现展开,介绍了 ConsumerCoupon 实体类、ConsumerMapper 访问数据库的定义、以及修改 LogonListener 加入发放优惠券的代码、最后是 ConsumerController 的定义。下节课我们会带大家对代码进行测试。这里将代码给大家,下期见,拜拜。 友情提示:本章讲述代码只是部分核心代码,完整代码请查阅文末链接中代码,谢

谢。