

# day05 功能测试&支付&优惠券

## 今日概要

- 生成订单的功能测试
  - 状态变化
  - 生成订单
  - 生成订单&退款
  - 生成订单24小时没有支付，做扣保证金处理
- 优惠券实现思路&页面
  - 一起设计
  - 开发任务
- 支付实现思路&页面
  - 一起设计
  - 开发任务

## 今日详细

### 1. 功能测试

#### 1.1 状态的变化

在后台创建一个专场 + 一个拍品。

小茶书平台 拍卖管理					
拍卖专场列表					
添加					
封面	专场	预展时间	拍卖时间	状态	操作
	第1场	2020-02-14 08:56	2020-02-14 08:57	已结束	编辑 删除

## 专场信息

专场 第1场

预展开始 2020-02-14 08:56


预展结束 2020-02-14 08:57

拍卖开始 2020-02-14 08:57

拍卖结束 2020-02-14 08:58

## 拍品

+ 添加

封面	名称	图录号	起拍价	加价幅度	参考价格	状态
	v1	20200214085556129246	100	100	10 - 100	流拍

删除

## 1.2 生成订单 (小程序)

- 在后台管理
  - 创建专场&拍品&状态变化的时间（未开拍、预展、拍卖、结束）【ok】
  - 定时任务
    - 状态变化【ok】
    - 生成订单
- 小程序 & API
  - 专场列表
  - 专场详细（拍品列表）
    - 缴纳保证金
    - 参与竞拍

1 row	Tx: Auto	DB	Tab-se...d (TSV)	DDL	View Query
<Filter criteria>					
id	uid	price	create_date	deposit_id	item_id
1	2 243f3b614d80251170bc885ec0d2a8b7	300	2020-02-14 01:58:00.887757	26	14

## 1.3 生成订单&amp;退款

两个人参与A和B两个用户进行竞价。

- A/B都交保证金
- A拍到/B没拍到

2 rows	Tx: Auto	DB	Tab-se...d (TSV)	DDL	View Query
<Filter criteria>					
id	uid	amount	deposit_id	status	
1	1 e0e62609f8d5ebe51e78fa1737b5a50d	200	30	2	
2	2 32255b2ca5a485a094334a68e6d3dd43	200	31	2	

## 1.4 逾期未付款处理

修改24小时之后执行的定时任务 -> 1分钟之后执行逾期未付款处理

修改了celery的任务函数的内容，必须重启celery的worker。

```
64 user=lucky_object.user,
65 item=item_object,
66 deposit_type=1).first()
67
68 if not deposit_object:...
69
70 # 生成订单 (待支付)
71 order_object = models.Order.objects.create(
72     status=1, # bug1: 没有写默认状态
73     uid=md5(str(uuid.uuid4())), # bug2: 订单号生成有问题
74     user=lucky_object.user,
75     item=item_object,
76     deposit=deposit_object, # (单品、专场)
77     price=lucky_object.price,
78 )
79
80 # 单品保证金: 所有没有拍到商品 & 缴纳的是单品保证金记录。
81 item_unfortunate_list = models.DepositRecord.objects.filter(item=item_object, deposit_type=1)
82 user=lucky_object.user
83 total_unfortunate_list.extend(item_unfortunate_list)
84
85 # 调用定时任务: 24小时内要支付, 否则需拍扣除保证金。
86 # date = datetime.datetime.utcnow() + datetime.timedelta(hours=24)
87 # date = datetime.datetime.utcnow() + datetime.timedelta(minutes=1)
88 task_id = twenty_four_hour.apply_async(args=[order_object.id], eta=date).id
89 order_object.twenty_four_task_id = task_id
90 order_object.save()
91
92 # 专场: 更新成交额
93 auction_object.total_price = total
```

uid	deposit_type	amount	auction_id	item_id	user_id	pay_type	balance
1 3db06b6be3a8da138938efa63c209f98	1	100	32	14	4	1	100
2 5a09c6334b14d5c1f0b1fd4edd68e25	1	100	33	15	4	1	100
3 2750ee2e2bdd263c521c286ac171c98	1	100	34	16	5	1	100
4 32d3618a-b491-4240-b358-d007145ec6eb	1	200	35	17	1	1	200
5 5c85a005-9ad5-494d-a418-7d6f0bedb5b8	1	200	35	17	6	1	0
6 3caff11f-7759-4850-9bd0-0ad9e4d58603	1	200	36	18	1	1	0
7 17751b3d-bb3a-4dc0-86bd-5f31cbbff1c5	1	200	36	18	6	1	200
8 c4fad74c2f33eaa2e0e5a4c3f2703dc0	1	100	37	19	6	1	0

id	amount	order_id
1 1	100	7

小茶书平台 拍卖管理

专场信息

专场 第9场

预展开始 2020-02-14 11:05

预展结束 2020-02-14 11:06

拍卖开始 2020-02-14 11:06

拍卖结束 2020-02-14 11:07

拍品

封面	名称	图录号	起拍价	加价幅度	参考价格	状态	
	v	20200214110443492773	100	100	100 - 1000	逾期未支付	删除

代码示例：【功能测试示例代码.zip】

2.优惠券

- 后台运营人员
  - 创建优惠券
- 用户小程序：
  - 领取优惠券

○ 使用优惠券





暂无数据

## 2.1 表结构

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	优惠券												
2	ID	专场 (通用/专场)	封面	优惠券名称	限制条件	抵扣金额	数量	剩余数量	状态	领取开始时间	领取结束时间	使用开始时间	使用结束时间
3													
4													
5													
6													
7													
8	用户优惠券												
9	ID	用户ID	优惠券ID	状态	订单ID (默认为空)								
10													
11													
12													
13													

## 2.2 任务（下午5点前任务）

- 根据表结构创建django的Models类。
- 后台管理实现优惠券管理
  - 仅实现增加。
- 小程序
  - 领券中心
    - 展示所有的优惠券
    - 领取优惠券
  - 我的优惠券

## 3. 优惠券功能实现

关于优惠券相关的业务可以划分：

- 后台管理，运营人员创建优惠券。
- 小程序用户，申请优惠券。

### 3.1 后台创建优惠券

#### 1. 优惠券列表

小茶书平台 拍卖管理 优惠券管理								
优惠券列表								
添加								
封面	名称	专场	金额	剩余数量	申请开始时间	申请结束	状态	操作
	v3	第9场	200	1/100	2020-02-14 16:25	2020-02-14 17:30	已结束	删除
	v1	第1场	200	0/100	2020-02-14 14:59	2020-02-14 15:00	已结束	删除

#### 2. 添加优惠券

- 用户输入相关优惠券信息：金额、名称、专场、状态(未开始、领取中、已结束领取)
- 优惠券的状态如何更新？
  - celery定时任务1：在申请开始时间将优惠券的状态 未开始 --> 领取中
  - celery定时任务2：在申请结束的时间将优惠券的状态 领取中 --> 已结束

```
class Coupon(models.Model):
    """ 优惠券 """
    status_choices = (
        (1, '未开始'),
        (2, '领取中'),
        (3, '已结束')
    )
    status = models.SmallIntegerField(verbose_name='状态',
    choices=status_choices, default=1)
    title = models.CharField(verbose_name='优惠券名称', max_length=32)
    auction = models.ForeignKey(verbose_name='专场', to='Auction')

    money = models.PositiveIntegerField(verbose_name='抵扣金额',
    default=200)
    # 100
    count = models.PositiveIntegerField(verbose_name='创建数量',
    default=100)
    # 剩余数量: 100 -1
    # 申请数量: 0 +1
    apply_count = models.PositiveIntegerField(verbose_name='已申请数量',
    default=0)

    apply_start_date = models.DateTimeField(verbose_name='开始领取时间')
```

```

apply_stop_date = models.DateTimeField(verbose_name='领取结束时间')

apply_start_task_id = models.CharField(verbose_name='celery任务ID',
max_length=64, null=True, blank=True)
apply_stop_task_id = models.CharField(verbose_name='celery任务ID',
max_length=64, null=True, blank=True)
deleted = models.BooleanField(verbose_name='是否删除', default=False)

```

小茶书平台 拍卖管理 优惠券管理

Link

优惠券 保存

优惠券名称	<input type="text"/>	专场	<input type="text"/>
抵扣金额	<input type="text" value="200"/>	创建数量	<input type="text" value="100"/>
开始领取时间	<input type="text"/>	领取结束时间	<input type="text"/>

```

class CouponModelForm(BootstrapModelForm):
    class Meta:
        model = models.Coupon
        exclude = ['status', 'apply_count', 'apply_start_task_id',
'apply_stop_task_id', 'deleted']

def coupon_add(request):
    if request.method == 'GET':
        form = CouponModelForm()
        return render(request, 'web/coupon_form.html', {'form': form})
    form = CouponModelForm(data=request.POST)
    if form.is_valid():
        instance = form.save()

        # 1.定时任务, 从 未开始 到 领取中
        start_apply_datetime =
datetime.datetime.utcnow().timestamp(form.instance.apply_start_date.timestamp())

        start_task_id = tasks.coupon_start_apply.apply_async(args=
[instance.id], eta=start_apply_datetime).id

        # 2.定时任务, 从 领取中 到 已结束
        start_apply_datetime =
datetime.datetime.utcnow().timestamp(form.instance.apply_stop_date.timestamp())

        stop_task_id = tasks.coupon_stop_apply.apply_async(args=
[instance.id], eta=start_apply_datetime).id

        instance.apply_start_task_id = start_task_id
        instance.apply_stop_task_id = stop_task_id
        instance.save()

```

```
return redirect('coupon_list')
return render(request, 'web/coupon_form.html', {'form': form})
```

# tasks.py

@shared\_task

```
def coupon_start_apply(coupon_id):
    models.Coupon.objects.filter(id=coupon_id, status=1).update(status=2)
```

@shared\_task

```
def coupon_stop_apply(coupon_id):
    models.Coupon.objects.filter(id=coupon_id, status=2).update(status=3)
```

功能补充：

- 物理删除&逻辑删除

物理删除：

```
models.User.objects.filter(id=1).delete()
```

逻辑删除（不要删除更新一下状态）：[推荐]

通过状态来实现

- 用户领取优惠券自己优惠券

用户 优惠券(专场)

如果在专场结束时 + 24小时，找到这个专场优惠券，找到所有领取这个优惠的所有用户优惠券记录。 -> 已过期

可以集成在end\_auction\_task定时任务中。

## 3.2 删除优惠券

逻辑删除数据库中的优惠券。

取消执行定时任务。

```
def coupon_delete(request, pk):
    """ 删除优惠券 """
    coupon_object = models.Coupon.objects.filter(id=pk, deleted=False).first()
    if not coupon_object:
        return JsonResponse({'status': False, 'error': '优惠券不存在'})

    start_result = AsyncResult(id=coupon_object.apply_start_task_id,
                               app=celery_app)
    start_result.revoke()
```



```

stop_result = AsyncResult(id=coupon_object.apply_stop_task_id,
app=celery_app)
stop_result.revoke()

models.Coupon.objects.filter(id=pk, deleted=False).update(deleted=True)

return JsonResponse({'status': True})

```

### 3.3 编辑优惠券

必须是未开始时才能编辑，如果已经开始不再提供编辑。

```

def coupon_edit(request, pk):
    coupon_object = models.Coupon.objects.filter(id=pk, deleted=False).first()
    if not coupon_object or coupon_object.status != 1:
        return HttpResponse('优惠券不存在或优惠券已开始申请')

    if request.method == 'GET':
        form = CouponModelForm(instance=coupon_object)
        return render(request, 'web/coupon_form.html', {'form': form})

    form = CouponModelForm(instance=coupon_object, data=request.POST)
    if form.is_valid():
        if "apply_start_date" in form.changed_data:
            async_result = AsyncResult(id=coupon_object.apply_start_task_id,
app=celery_app)
            async_result.revoke()
            eta =
datetime.datetime.utcnow().timestamp()
            start_task_id = tasks.coupon_start_apply.apply_async(args=
[coupon_object.id], eta=eta).id
            form.instance.apply_start_task_id = start_task_id

            if "apply_stop_date" in form.changed_data:
                async_result = AsyncResult(id=coupon_object.apply_stop_task_id,
app=celery_app)
                async_result.revoke()
                eta =
datetime.datetime.utcnow().timestamp()
                start_task_id = tasks.coupon_stop_apply.apply_async(args=
[coupon_object.id], eta=eta).id
                form.instance.apply_start_task_id = start_task_id

            form.save()
            return redirect('coupon_list')

    return render(request, 'web/coupon_form.html', {'form': form})

```

知识点：form.changed\_data 获取本地都修改了那些字段, 例如： ["name","age"]。

### 3.4 小程序领券中心

小程序向后台API发送GET请求，获取所有优惠券。

API接口请求之后，获取所有数据。

具体操作：

- 小程序：
  - home页面的领券中心写一个点击跳转。

```
<navigator class="row" url="/pages/coupon/coupon">
  <view class="left">领券中心</view>
  <view class="right">
    <text>你的福利都在这里</text>
    <image class="go-icon"
src='/static/images/icon/to_icon_show_small.png'></image>
  </view>
</navigator>
```

- 领券中心的页面（优惠券列表 状态!=1）

```
{
  data:{
    couponList:[]
  }
}
getCoupon:function(){
  res = 发送request请求获取数据
  this.setData({
    couponList: res.data
  })
},
onload: function(){
  // 向api你发送请求获取所有优惠券
  this.getCoupon();
}

onPullDownRefresh: function() {
  //向api你发送请求获取所有优惠券
  this.getCoupon();
},
```

```
// pages/coupon/coupon.js
var api = require("../config/api.js")
var app = getApp()
```

```

Page({

  /**
   * 页面的初始数据
   */
  data: {
    couponList: []
  },

  getCoupon: function() {
    wx: wx.request({
      url: api.Coupon,
      method: 'GET',
      dataType: 'json',
      responseType: 'text',
      success: (res) => {
        this.setData({
          couponList: res.data
        })
      },
      fail: function(res) {},
      complete: function(res) {},
    })
  },

  /**
   * 生命周期函数--监听页面加载
   */
  onLoad: function(options) {
    this.getCoupon();
  },

  /**
   * 页面相关事件处理函数--监听用户下拉动作
   */
  onPullDownRefresh: function() {
    this.getCoupon();
  },
})

```

```

<!--pages/coupon/coupon.wxml-->
<view class="container">
  <view class="item" wx:for="{{couponList}}" wx:key="index"
bindtap="applyCoupon" data-item="{{item}}" data-index="{{index}}">
    <view class="cover">
      <image src="{{item.cover}}"></image>
    </view>
    <view class="info">
      <text>¥{{item.money}}</text>
    </view>
  </view>
</view>

```

```

        <text>剩余{{item.remain}}张</text>
        <text>{{item.status_text}}</text>
    </view>
</view>
</view>

```

### 3.5 小程序 领取优惠券

领券中为每个优惠券绑定点击事件，点击获取优惠券。

当前用户想要获取优惠券需要传递：

当前用户  
优惠券id

API接收到请求之后

校验：

- 用户是否登录
- 优惠券是否存在？
- 优惠券的状态是否是 领取中 ？
- 当前用户是否已经领过优惠券 ？

给用户申请到优惠券：

事务 + 锁：

- 创建一条用户优惠券记录
- 优惠券表中的 已申请数量 +1（剩余数量-1）

```

<!--pages/coupon/coupon.wxml-->
<view class="container">
    <view class="item" wx:for="{{couponList}}" wx:key="index"
    bindtap="applyCoupon" data-item="{{item}}" data-index="{{index}}">
        <view class="cover">
            <image src="{{item.cover}}"></image>
        </view>
        <view class="info">
            <text>¥{{item.money}}</text>
            <text>剩余{{item.remain}}张</text>
            <text>{{item.status_text}}</text>
        </view>
    </view>
</view>
</view>

```

```

applyCoupon: function(e) {
    if (e.currentTarget.dataset.item.status != 2) {

```

```

        return
    }
    var userInfo = app.globalData.userInfo;
    wx.request({
        url: api.UserCoupon,
        data: {
            coupon: e.currentTarget.dataset.item.id
        },
        header: {
            Authorization: userInfo ? "token " + userInfo.token : ""
        },
        method: 'POST',
        dataType: 'json',
        responseType: 'text',
        success: (res) => {
            this.setData({
                ["couponList[" + e.currentTarget.dataset.index + "].remain"]:
res.data.remain
            })
        },
        fail: function(res) {},
        complete: function(res) {},
    })
},

```

难点：校验流程

### 3.6 我的优惠券

- 简单实现

1. 数据库获取所有我自己的优惠券  
`queryset = models.UserCoupon.objects.filter(user=request.user)`
2. 序列化优惠券列表即可
3. 小程序段获取数据即可

- 示例实现

1. 数据库获取所有我自己的优惠券  
`queryset = models.UserCoupon.objects.filter(user=request.user)`
2. 对queryset进行序列化
3. 构造数据  

```

data = {
    1:{
        text:'未使用',
        child:[
            ....
        ]
    },

```

```

2:{
  text:'已使用',
  child:[
    ....
  ]
},
3:{
  text:'已过期',
  child:[
    ....
  ]
},
}

```

#### 4.小程序

点击那个状态，就显示此状态下的所有优惠券。

seleted=1

循环： 1未使用 2已使用 3已过期 点击事件：点击修改seleted

循环展示： data[selected].child

示例代码：【优惠券实现.zip】

## 4.任务

- 参考示例实现优惠券的功能
- 支付功能设计
  - 抵扣保证金
  - 优惠券
  - 订单
  - 收货地址

## 答疑

### 1. 张达

```

wx.request({
  url:....,
  header: {
    Authorization: "666",
    xx:123
  }
})

```

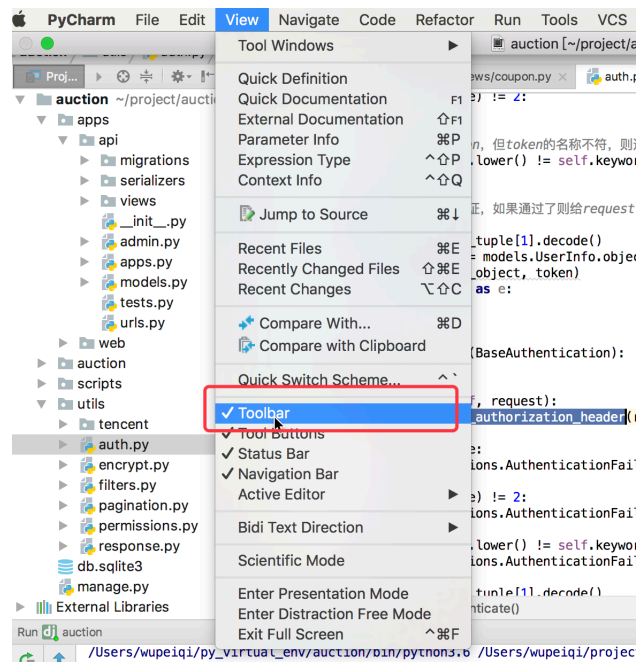
```

request.META.get('HTTP_AUTHORIZATION')
request.META.get('HTTP_XX')

```

示例代码见认证组件。

## 2. pycharm前进和后退



## 3. 获取数据

```
def func(request):  
    request.POST  
    json.loads(request.body.decode('utf-8'))
```

```
class TestView(APIView):  
    def get(self, request):  
        request.data
```

## 4. drf获取值

```
GET参数: request.query_params  
POST请求体: request.data
```

## 已做完的同学，完善代码

---

- 后台管理
  - 创建优惠券，优惠券的状态是如何更新？ + celery去更新优惠券的状态



```
status_choices = (
    (1, '未开始'),
    (2, '领取中'),
    (3, '已结束')
)
status = models.SmallIntegerField(verbose_name='状态',
    choices=status_choices, default=1)
```

- 思考：优惠券 --> 编辑 和 删除 时，如何变更定时任务。
- 思考：我的优惠券 ->

```
class UserCoupon(models.Model):
    status_choices = (
        (1, '未使用'),
        (2, '已使用'),
        (3, '已过期')
    )
    status = models.SmallIntegerField(verbose_name='状态',
    choices=status_choices, default=1)
```