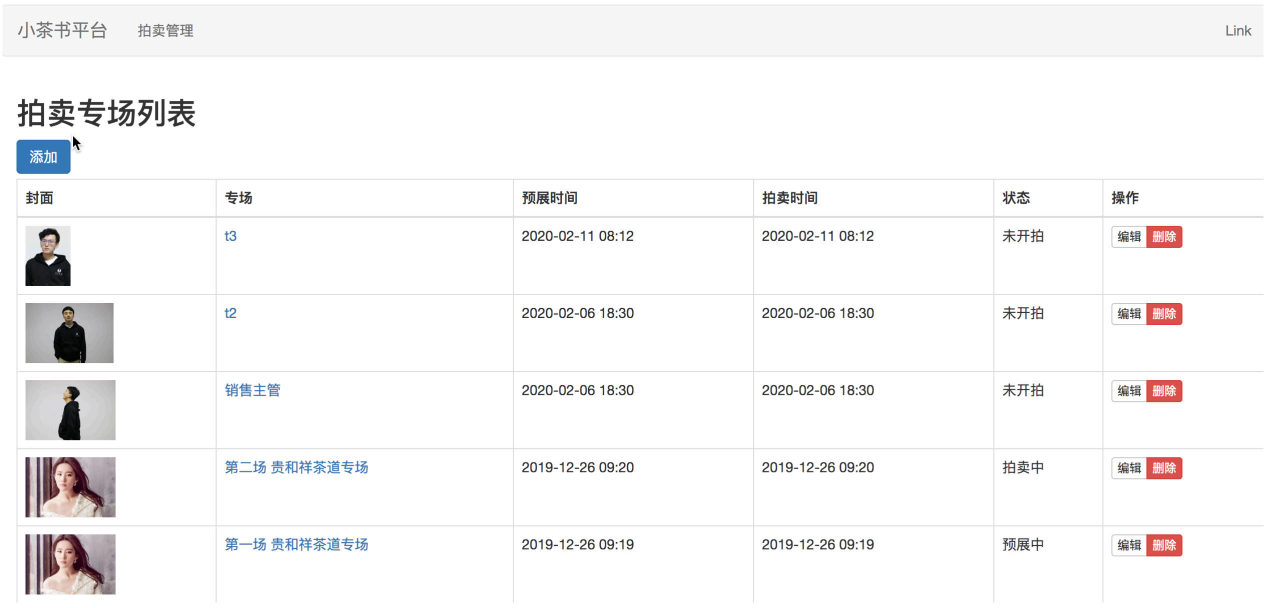
day02 后台管理&定时任务

1. 专场管理

```
class Auction(models.Model):
    """
    拍卖系列
    """

    status_choices = (
        (1, '未开拍'),
        (2, '预展中'),
        (3, '拍卖中'),
        (4, '已结束')
    )
    status = models.PositiveSmallIntegerField(verbose_name='状态',
choices=status_choices, default=1)

    title = models.CharField(verbose_name='标题', max_length=32)
    cover = models.FileField(verbose_name='封面', max_length=128)
    video = models.CharField(verbose_name='预览视频', max_length=128,
null=True, blank=True)

    preview_start_time = models.DateTimeField(verbose_name='预展开始时间')
    preview_end_time = models.DateTimeField(verbose_name='预展结束时间')

    auction_start_time = models.DateTimeField(verbose_name='拍卖开始时间')
    auction_end_time = models.DateTimeField(verbose_name='拍卖结束时间')

    deposit = models.PositiveIntegerField(verbose_name='全场保证金',
default=1000)

    total_price = models.PositiveIntegerField(verbose_name='成交额', null=True,
blank=True)
    goods_count = models.PositiveIntegerField(verbose_name='拍品数量',
default=0)
    bid_count = models.PositiveIntegerField(verbose_name='出价次数', default=0)
    look_count = models.PositiveIntegerField(verbose_name='围观次数',
default=0)
    create_time = models.DateTimeField(verbose_name='创建时间',
auto_now_add=True)

    class Meta:
        verbose_name_plural = '拍卖系列'

    def __str__(self):
        return self.title
```

1.1 专场列表



小茶书平台 拍卖管理 Link

拍卖专场列表

添加

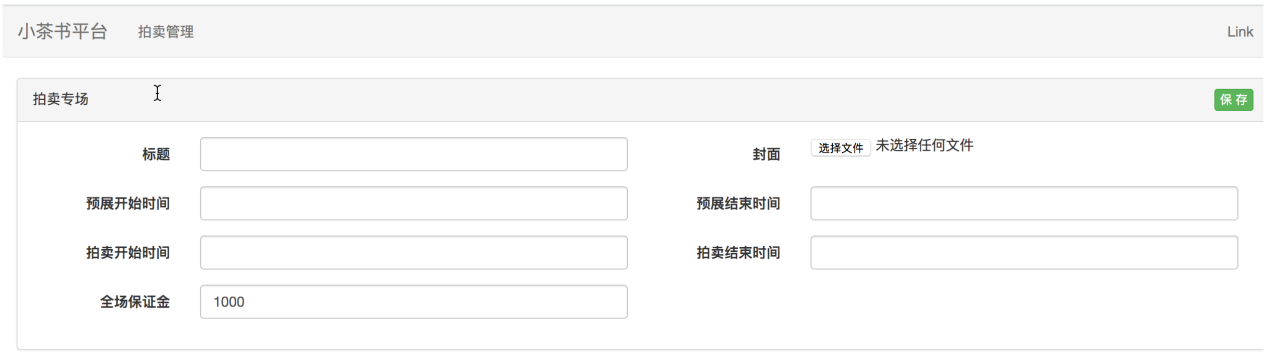
封面	专场	预展时间	拍卖时间	状态	操作
	t3	2020-02-11 08:12	2020-02-11 08:12	未开拍	编辑 删除
	t2	2020-02-06 18:30	2020-02-06 18:30	未开拍	编辑 删除
	销售主管	2020-02-06 18:30	2020-02-06 18:30	未开拍	编辑 删除
	第二场 贵和祥茶道专场	2019-12-26 09:20	2019-12-26 09:20	拍卖中	编辑 删除
	第一场 贵和祥茶道专场	2019-12-26 09:19	2019-12-26 09:19	预展中	编辑 删除

```
url(r'^auction/list/$', auction.auction_list, name='auction_list'),
```

```
def auction_list(request):
    """
    拍卖系列列表
    :param request:
    :return:
    """
    queryset = models.Auction.objects.all().order_by('-id')
    return render(request, 'web/auction_list.html', {'queryset': queryset})
```

```
<table class="table table-bordered">
            <thead>
            <tr>
                <th>封面</th>
                <th>专场</th>
                <th>预展时间</th>
                <th>拍卖时间</th>
                <th>状态</th>
                <th>操作</th>
            </tr>
            </thead>
            <tbody>
            {% for item in queryset %}
                <tr>
                    <td>
                        <img style="height: 60px;" src="{{ item.cover }}">
                    </td>
                    <td><a href="{% url 'auction_item_list' auction_id=item.id
%}">{{ item.title }}</a></td>
                    <td>{{ item.preview_start_time|date:"Y-m-d H:i" }}</td>
                    <td>{{ item.auction_start_time|date:"Y-m-d H:i" }}</td>
                    <td>{{ item.get_status_display }}</td>
                    <td>
                        <div class="btn-group btn-group-xs" role="group" aria-
label="Small button group">
                            <a class="btn btn-default" href="{% url
'auction_edit' pk=item.id %}">编辑</a>
                            <a class="btn btn-danger"
                               onclick="removeRow(this,'{% url
'auction_delete' pk=item.id %}')">删除</a>
                        </div>
                    </td>
                </tr>
            {% endfor %}
            </tbody>
        </table>
```

1.2 添加专场



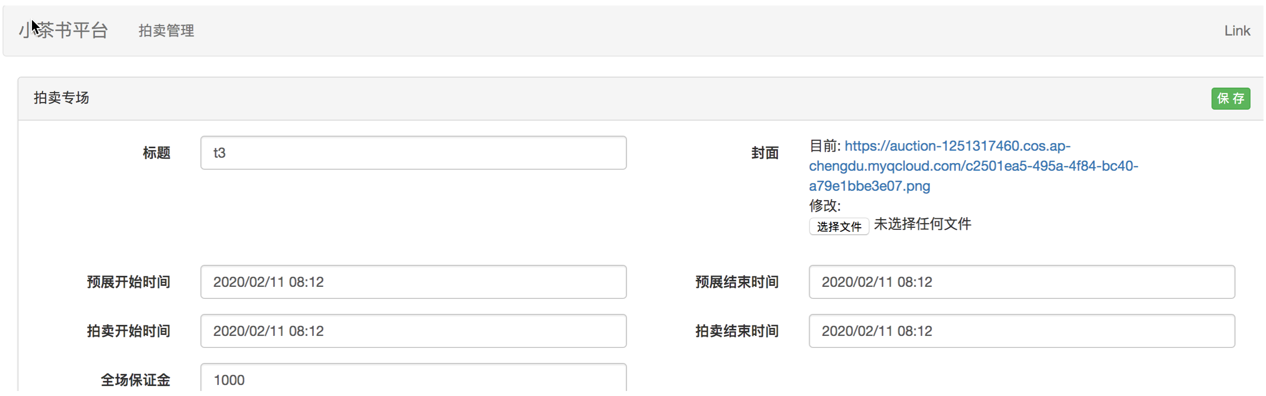
标题		封面	选择文件 未选择任何文件
预展开始时间		预展结束时间	
拍卖开始时间		拍卖结束时间	
全场保证金	1000		

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-
from qcloud_cos import CosConfig
from qcloud_cos import CosS3Client


def upload_file(file_object, key, bucket='auction-1251317460', region="ap-
chengdu"):
    secret_id = 'AKIDW3Rgszw84ylQxMzNn7KOJ6kFPSL5c5MU'  # 替换为用户的 secretId
    secret_key = 'GQSMXmtsjROQhuIalzTp25OnU6digZSD'  # 替换为用户的 secretKey
    token = None  # 使用临时密钥需要传入 Token，默认为空，可不填
    scheme = 'https'  # 指定使用 http/https 协议来访问 COS，默认为 https，可不填
    config = CosConfig(Region=region, SecretId=secret_id,
SecretKey=secret_key, Token=token, Scheme=scheme)
    # 2. 获取客户端对象
    client = CosS3Client(config)
    # 3. 上传文件
    response = client.upload_file_from_buffer(
        Bucket=bucket,
        Body=file_object,
        Key=key
    )
    return "https://{0}.cos.{1}.myqcloud.com/{2}".format(bucket, region, key)
```

```
url(r'^auction/add/$', auction.auction_add, name='auction_add'),
```

```
class BoostrapModelForm(ModelForm):
    exclude_bootstrap_class = []
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for name, field in self.fields.items():
            if name in self.exclude_bootstrap_class:
                continue
            old_class = field.widget.attrs.get('class', "")
            field.widget.attrs['class'] = old_class + ' form-control'

class AuctionModelForm(BoostrapModelForm):
    exclude_bootstrap_class = ['cover',]

    class Meta:
        model = models.Auction
        exclude = ['status', 'total_price', 'goods_count', 'bid_count',
'look_count', 'video']

    def clean(self):
        cleaned_data = self.cleaned_data
        # 没有上传文件
        """
        ModelForm的表单验证流程:
            1.调用内置的校验
            2.调用每个字段的钩子
            3.调用clean方法
        """
        # 获取用户提交的文件对象
        cover_file_object = cleaned_data.get('cover')
        if not obj:
            return cleaned_data
        # 如果是编辑文件(未修改)
        from django.db.models.fields.files import FieldFile
        from django.core.files.uploadedfile import InMemoryUploadedFile
        if isintance(cover_file_object,FieldFile):
            return cleaned_data

        # 把对象上传到cos, 返回地址
        ext = cover_file_object.name.rsplit('.', maxsplit=1)[-1]
        file_name = "{0}.{1}".format(str(uuid.uuid4()), ext)
        cleaned_data['cover'] = upload_file(cover_file_object, file_name)
        return cleaned_data


def auction_add(request):
    if request.method == 'GET':
        form = AuctionModelForm()
        return render(request, 'web/auction_form.html', {'form': form})
    form = AuctionModelForm(data=request.POST, files=request.FILES)
    if form.is_valid():
        form.save()
        return redirect('auction_list')
    return render(request, 'web/auction_form.html', {'form': form})
```

1.3.编辑专场



标题	t3	封面	目前: https://auction-1251317460.cos.ap-chengdu.myqcloud.com/c2501ea5-495a-4f84-bc40-a79e1bbe3e07.png 修改: 选择文件 未选择任何文件
预展开始时间	2020/02/11 08:12	预展结束时间	2020/02/11 08:12
拍卖开始时间	2020/02/11 08:12	拍卖结束时间	2020/02/11 08:12
全场保证金	1000		

```
url(r'^auction/edit/(?P<pk>\d+)/$', auction.auction_edit,
name='auction_edit'),
```

```
def auction_edit(request, pk):
    auction_object = models.Auction.objects.filter(id=pk).first()
    if request.method == 'GET':
        form = AuctionModelForm(instance=auction_object)
        return render(request, 'web/auction_form.html', {'form': form})
    form = AuctionModelForm(data=request.POST, files=request.FILES,
instance=auction_object)
    if form.is_valid():
        form.save()
        return redirect('auction_list')
    return render(request, 'web/auction_form.html', {'form': form})
```

```
{% extends 'web/layout.html' %}
{% load staticfiles %}
{% block css %}
```

```
    <link rel="stylesheet" href="{% static 'web/datetimepicker/bootstrap-
datetimepicker.min.css' %}">
    <style>
        input[type='file'] {
            min-height: 34px;
        }
    </style>
{% endblock %}

{% block content %}
    <div class="container-fluid">
        <form method="post" novalidate enctype="multipart/form-data">
            {% csrf_token %}
            <div class="panel panel-default">
                <div class="panel-heading clearfix">拍卖专场
                    <input type="submit" class="btn btn-success btn-xs"
value="保 存" style="float: right;">
                </div>
                <div class="panel-body">
                    <div class="form-horizontal clearfix">
                        {% for field in form %}
                            <div class="col-sm-6">
                                <div class="form-group">
                                    <label for="{{ field.id_for_label }}"
                                           class="col-sm-3 control-label">{{
field.label }}</label>
                                    <div class="col-sm-9">
                                        {{ field }}
                                        <span style="color: red;">{{
field.errors.0 }}</span>
                                    </div>
                                </div>
                            </div>
                        {% endfor %}

                    </div>
                </div>
            </div>
        </form>
    </div>
{% endblock %}

{% block js %}
    <script src="{% static 'web/datetimepicker/bootstrap-
datetimepicker.min.js' %}"></script>
    <script src="{% static 'web/datetimepicker/bootstrap-datetimepicker.zh-
CN.js' %}"></script>
    <script>
        Date.prototype.Format = function (fmt) { //author: meizz
```

```
            var o = {
                "M+": this.getMonth() + 1, //月份
                "d+": this.getDate(), //日
                "h+": this.getHours(), //小时
                "m+": this.getMinutes(), //分
                "s+": this.getSeconds(), //秒
                "q+": Math.floor((this.getMonth() + 3) / 3), //季度
                "S": this.getMilliseconds() //毫秒
            };
            if (/(y+)/.test(fmt)) fmt = fmt.replace(RegExp.$1,
(this.getFullYear() + "").substr(4 - RegExp.$1.length));
            for (var k in o)
                if (new RegExp("(" + k + ")").test(fmt)) fmt =
fmt.replace(RegExp.$1, (RegExp.$1.length == 1) ? (o[k]) : (("00" +
o[k]).substr(("" + o[k]).length)));
            return fmt;
        };

        $(function () {
            initDatepicker();
        });

        function initDatepicker() {

 $('#id_preview_start_time,#id_preview_end_time,#id_auction_start_time,#id_auc
tion_end_time').datetimepicker({
                language: "zh-CN",
                minView: "hour",
                sideBySide: true,
                format: 'yyyy-mm-dd hh:ii',
                bootcssVer: 3,
                startDate: new Date(),
                autoclose: true
            })
        }
    </script>

{% endblock %}
```

1.4 删除专场

```
 <tbody>
            {% for item in queryset %}
                <tr>
                    <td>
                        <img style="height: 60px;" src="{{ item.cover }}">
                    </td>
                    <td><a href="{% url 'auction_item_list' auction_id=item.id
%}">{{ item.title }}</a></td>
```

```
                    <td>{{ item.preview_start_time|date:"Y-m-d H:i" }}</td>
                    <td>{{ item.auction_start_time|date:"Y-m-d H:i" }}</td>
                    <td>{{ item.get_status_display }}</td>
                    <td>
                        <div class="btn-group btn-group-xs" role="group" aria-
label="Small button group">
                            <a class="btn btn-default" href="{% url
'auction_edit' pk=item.id %}">编辑</a>
                            <a class="btn btn-danger"
                               onclick="removeRow(this,'{% url
'auction_delete' pk=item.id %}')">删除</a>
                        </div>
                    </td>
                </tr>
            {% endfor %}
            </tbody>
```

```
<script>
        function removeRow(ths, url) {
            var result = confirm("确认删除? ");
            if (result) {
                $.ajax({
                    url: url,
                    type: 'GET',
                    dataType: "JSON",
                    success: function (arg) {
                        if (arg.status) {
                            $(ths).parent().parent().parent().remove();
                        } else {
                            alert('删除失败');
                        }
                    }
                })
            }
        }
    </script>
```

```
def auction_delete(request, pk):
    models.Auction.objects.filter(id=pk).delete()
    return JsonResponse({'status': True})
```

1.5 datetimepicker.js

是一个结合BootStrap的组件，用于实现选择时间。

基本应用步骤：

- 引入css和js文件

```
<link rel="stylesheet" href="{% static 'web/datetimepicker/bootstrap-datetimepicker.min.css' %}">


<script src="{% static 'web/datetimepicker/bootstrap-datetimepicker.min.js' %}"></script>
    <script src="{% static 'web/datetimepicker/bootstrap-datetimepicker.zh-CN.js' %}"></script>
```

- html标签

```
<input id='i1' />
```

- 应用到到标签

```
$('#i1,#i2').datetimepicker({
    language: "zh-CN",
    minView: "hour",
    sideBySide: true,
    format: 'yyyy-mm-dd hh:ii',
    bootcssVer: 3,
    startDate: new Date(),
    autoclose: true
})
```

扩展：js中对时间进行格式化

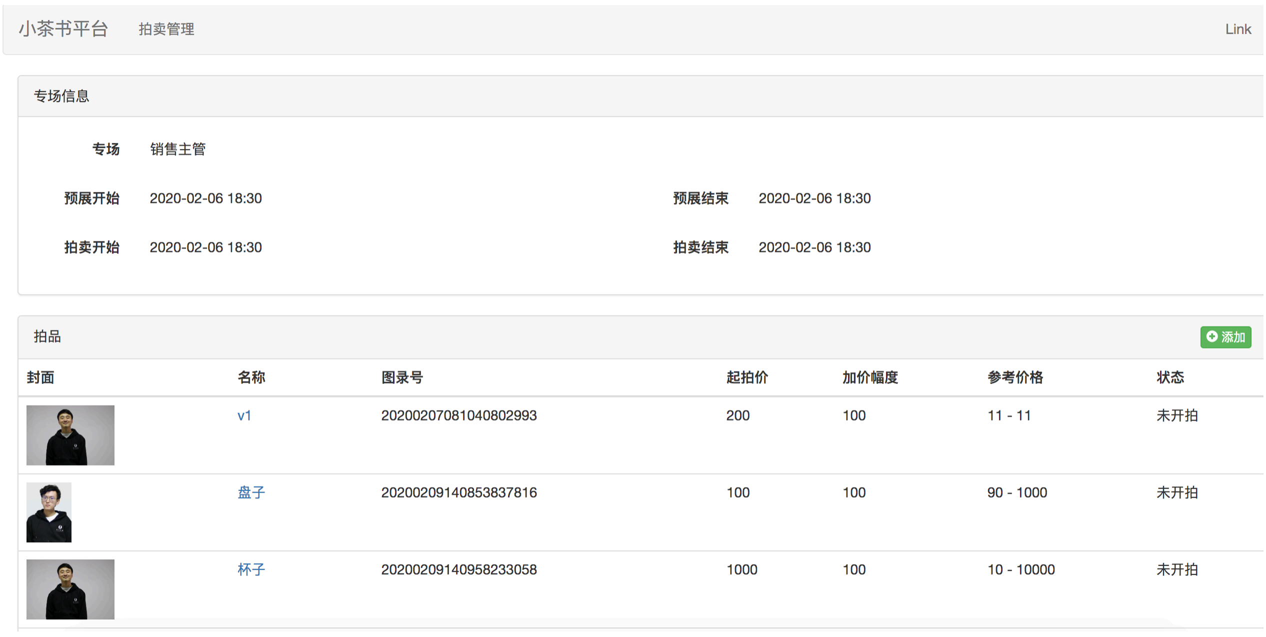
```
Date.prototype.Format = function (fmt) { //author: meizz
            var o = {
                "M+": this.getMonth() + 1, //月份
                "d+": this.getDate(), //日
                "h+": this.getHours(), //小时
                "m+": this.getMinutes(), //分
                "s+": this.getSeconds(), //秒
                "q+": Math.floor((this.getMonth() + 3) / 3), //季度
                "S": this.getMilliseconds() //毫秒
            };
            if (/(y+)/.test(fmt)) fmt = fmt.replace(RegExp.$1, (this.getFullYear() + "").substr(4 - RegExp.$1.length));
            for (var k in o)
                if (new RegExp("(" + k + ")").test(fmt)) fmt = fmt.replace(RegExp.$1, (RegExp.$1.length == 1) ? (o[k]) : (("00" + o[k]).substr(("" + o[k]).length)));
            return fmt;
        };
```

```
Date.prototype.format = function (fmt){
    return 123
}
var ctime = new Date()

// ctime -> Tue Feb 11 2020 09:50:27 GMT+0800 (中国标准时间)
var result = ctime.format('yyyy-MM-dd')
console.log(result);
```

应用场景：会议室预定系统 【metting.zip】

2. 拍品管理

2.1 拍品列表



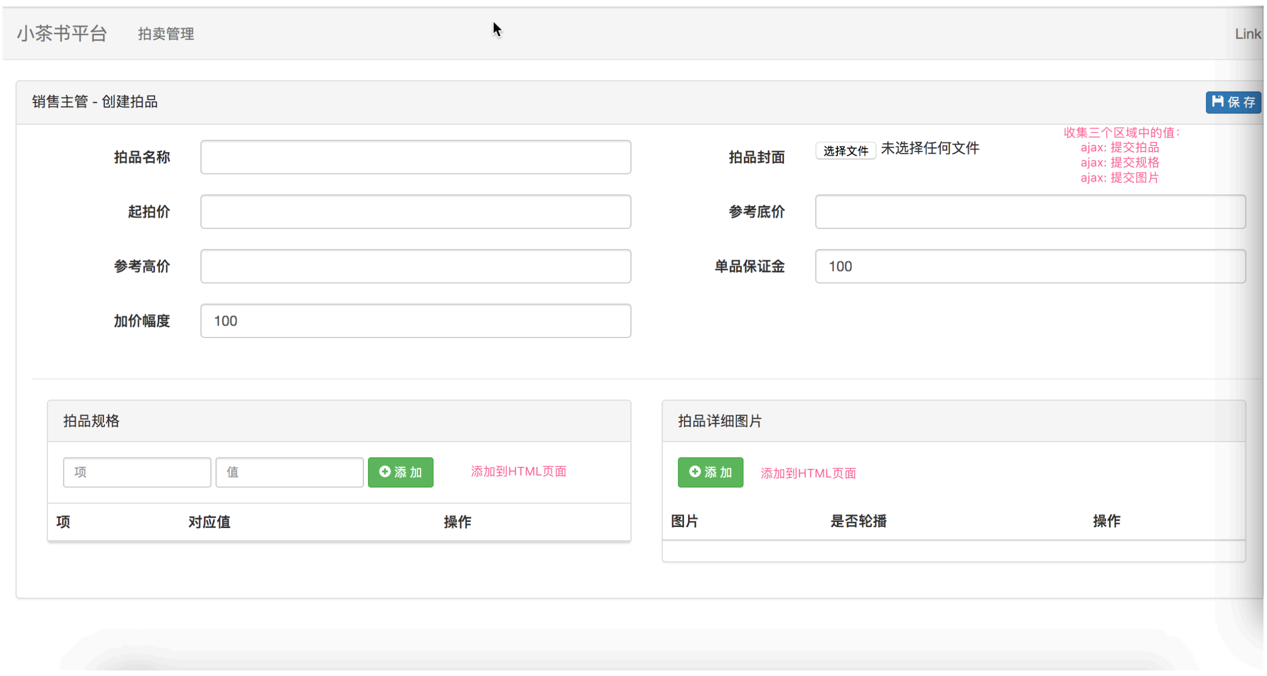
专场	销售主管		
预展开始	2020-02-06 18:30	预展结束	2020-02-06 18:30
拍卖开始	2020-02-06 18:30	拍卖结束	2020-02-06 18:30

封面	名称	图录号	起拍价	加价幅度	参考价格	状态
	v1	20200207081040802993	200	100	11 - 11	未开拍
	盘子	20200209140853837816	100	100	90 - 1000	未开拍
	杯子	20200209140958233058	1000	100	10 - 10000	未开拍

```
url(r'^auction/item/list/(?P<auction_id>\d+)/$', auction.auction_item_list,
name='auction_item_list'),
```

```
def auction_item_list(request, auction_id):
    auction_object = models.Auction.objects.filter(id=auction_id).first()
    item_list = models.AuctionItem.objects.filter(auction=auction_object)
    context = {
        'auction_object': auction_object,
        'item_list': item_list
    }
    return render(request, 'web/auction_item_list.html', context)
```

2.2 添加拍品&规格&图片



拍品名称
拍品封面 选择文件 未选择任何文件
收集三个区域中的值：
ajax: 提交拍品
ajax: 提交规格
ajax: 提交图片
起拍价
参考底价
参考高价
单品保证金 100
加价幅度 100

拍品规格
项 值 添 加 添加到HTML页面

项	对应值	操作

拍品详细图片
添 加 添加到HTML页面

图片	是否轮播	操作

```
url(r'^auction/item/add/(?P<auction_id>\d+)/$', auction.auction_item_add,
name='auction_item_add'),
```

```
class AuctionItemAddModelForm(BootStrapModelForm):
    exclude_bootstrap_class = ['cover']
    class Meta:
        model = models.AuctionItem
        exclude = ['auction', 'uid', 'status', 'deal_price', 'video',
'bid_count', 'look_count']
    def clean(self):
        cleaned_data = self.cleaned_data
        # 上传文件
        cover_file_object = cleaned_data.get('cover')
        if not cover_file_object or isinstance(cover_file_object, FieldFile):
            return cleaned_data
        ext = cover_file_object.name.rsplit('.', maxsplit=1)[-1]
        file_name = "{0}.{1}".format(str(uuid.uuid4()), ext)
        cleaned_data['cover'] = upload_file(cover_file_object, file_name)
        return cleaned_data

@csrf_exempt
def auction_item_add(request, auction_id):
    auction_object = models.Auction.objects.filter(id=auction_id).first()
    if request.method == 'GET':
        form = AuctionItemAddModelForm()
        context = {
            'form': form,
            'auction_object': auction_object
        }
```

```python
        return render(request, 'web/auction_item_add.html', context)

    form = AuctionItemAddModelForm(data=request.POST, files=request.FILES)
    if form.is_valid():
        form.instance.auction = auction_object
        form.instance.uid = datetime.datetime.now().strftime('%Y%m%d%H%M%S%f')
        instance = form.save()

        return JsonResponse({
            'status': True,
            'data': {
                'detail_url': reverse('auction_item_detail_add', kwargs=
{'item_id': instance.id}),
                'image_url': reverse('auction_item_image_add', kwargs=
{'item_id': instance.id}),
                'list_url': reverse('auction_item_list', kwargs={'auction_id':
auction_id})
            }
        })

    return JsonResponse({'status': False, 'errors': form.errors})
```

```html
 <a class="btn btn-primary btn-xs" style="float: right;" id="btnSave">
     <span class="glyphicon glyphicon-floppy-disk" aria-hidden="true"></span>
保 存
</a>

<form id="itemForm">
    {% for field in form %}
        <div class="col-sm-6">
            <div class="form-group">
                <label for="{{ field.id_for_label }}"
                       class="col-sm-3 control-label">{{ field.label }}
</label>
                <div class="col-sm-9" style="position: relative;">
                    {{ field }}
                    <span class="field-error">{{ field.errors.0 }}</span>
                </div>
            </div>
        </div>
    {% endfor %}
</form>


$(function () {
    bindSave();
});
function bindSave() {
    $('#btnSave').click(function () {
        $('.field-error').text("");

        // 加载层
        $('#loading').removeClass('hide');
        SUB_HANDLE_COUNT = 0;

        var itemFormData = new FormData($('#itemForm')[0]);
        $.ajax({
            type: 'POST',
            url: '{% url "auction_item_add" auction_id=auction_object.id %}',
            data: itemFormData,
            cache: false,
            contentType: false,
            processData: false,
            dataType: 'JSON',
            success: function (res) {
                if (!res.status) {
                    $('#loading').addClass('hide');
                    $.each(res.errors, function (key, text) {
                        $('#itemForm').find('[name="' + key +
'"]').next().text(text);
                    });
                    return
                }
                // 创建规格
                // createSpecification(res.data.detail_url,
res.data.list_url);
                // 创建图片
                // createImage(res.data.image_url, res.data.list_url);
            }
        })
    });
}
```

2.3.1 规格

- 添加按钮 & 删除按钮

  ```
  $('#btnSave').click()

  注意：删除按钮需要用事件委托进行绑定时间。
  $('table').on('click','delete',function(){

  })
  ```

- 收集规格的所有数据

```
detail_list = [
    {'key':'年份','item':"1990"},
    {'key':'年份','item':"1990"},
]
```

- 提交到后台

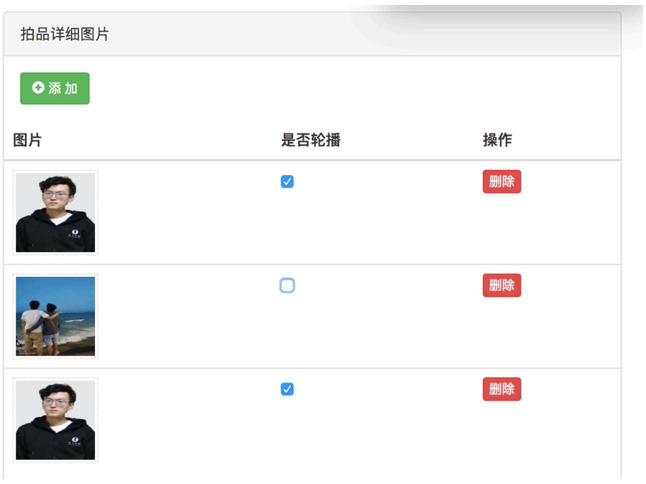
```
$.ajax({
    data:JSON.stringfy(detail_list)
})
```

- django后台接收数据

```
data_list= json.loads(request.body.decode('utf-8'))
```

2.3.2 图片处理

- 添加图片
- 选择图片 & 预览
- 收集数据



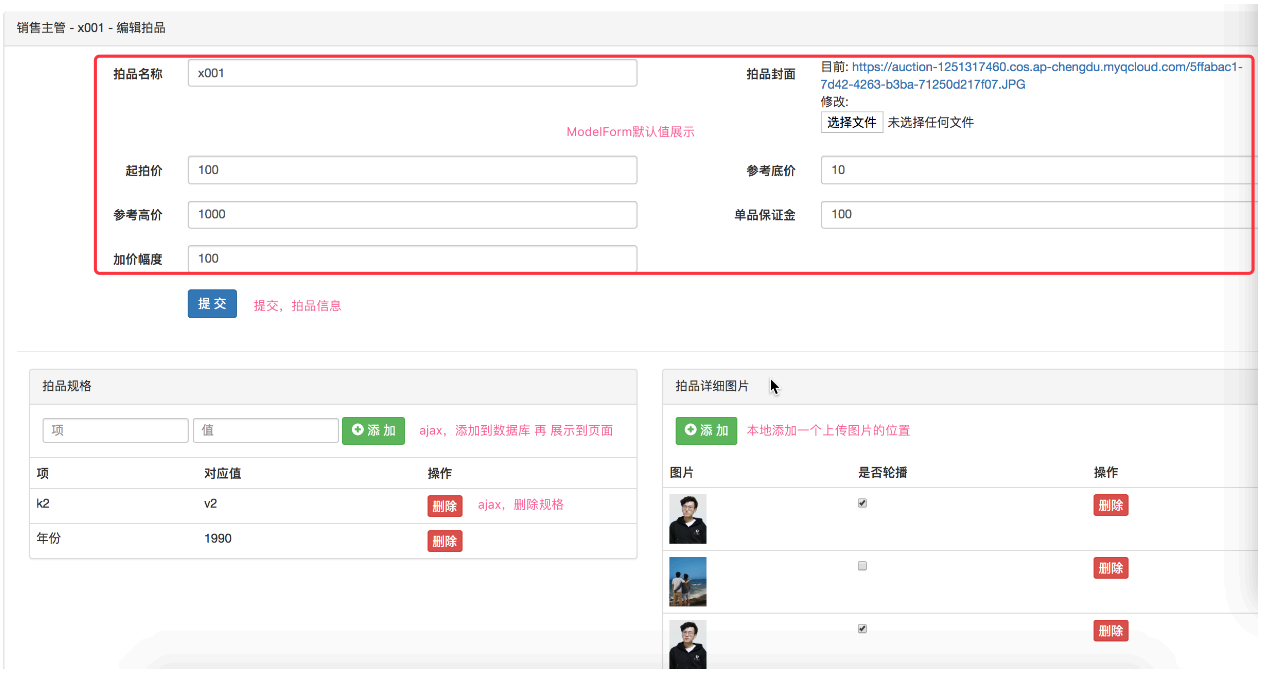
拍品详细图片

添 加

图片	是否轮播	操作
	✓	删除
	☐	删除
	✓	删除

```
武沛齐（省事）：
    // 找表单中的所有标签，并构造到FormData中。
        - 所有图片用相同的 name属性
            <input type='file' name='img' />        -> 
request.Files.getlist('img') 3
        - 找到已经选中的checkbox提交到后台
            <input type='checkbox' name='xx' />     -> 
request.POST.getlist('xx')  2
        - 隐藏了text标签
        <input type='text' name='oo' value='1' />         -> 
request.POST.getlist('oo') 3
        <input type='text' name='oo'  />         -> 
request.POST.getlist('oo')

    var formData = new FormData($('#imageForm')[0]);
    $.ajax({
        data:formData,
        ...
    })

    def upload_image(request):
        img_object_list = request.Files.getlist('img') #
[alex,wupeiq,oldbo]
        oo_list = request.POST.getlist('oo')           # ['1','','1']

    @csrf_exempt
    def auction_item_image_add(request, item_id):
        show_list = request.POST.getlist('show')
        image_object_list = request.FILES.getlist('img')
        orm_object_list = []
        for index in range(len(image_object_list)):
            image_object = image_object_list[index]
            if not image_object:
                continue
            ext = image_object.name.rsplit('.', maxsplit=1)[-1]
            file_name = "{0}.{1}".format(str(uuid.uuid4()), ext)
            cos_path = upload_file(image_object, file_name)
            orm_object_list.append(models.AuctionItemImage(img=cos_path,
item_id=item_id, carousel=bool(show_list[index])))
        if orm_object_list:
            models.AuctionItemImage.objects.bulk_create(orm_object_list)
        return JsonResponse({'status': True})



刘兵:
    拿到图片地址传到后台
    checkbox
    data = [
        {'k1':'图片地址','k2':True},
        {'k1':'图片地址','k2':True},
        {'k1':'图片地址','k2':True},
    ]
杨泽涛：同上
朱凡宇:
    data = [
        {'k1':图片对象,'k2':True},
        {'k1':图片对象,'k2':True},
        {'k1':图片对象,'k2':True},
    ]
    $.ajax({
        data:JSON.stringfy(data)
    })

错误:
    1. 如果ajax发送文件对象json无法处理。
    2. ajax上传数据含有文件 FormData 对象。
        $.ajax({
            data:FormData对象
        })
    3. ajax上传的数据没有文件时
        - 数据简单且value字符串或列表(无嵌套)
            $.ajax({
                data:{k1:123,k2:465}
            })
            request.POST
        - 嵌套的json格式数据
            $.ajax({
                data:JSON.stringfy({k1:[11,223],k2:[3,2,1,{'k9':666}})
            })
            request.body
```

2.3 编辑拍品&规格&图片


销售主管 - x001 - 编辑拍品

拍品名称	x001	拍品封面	目前: https://auction-1251317460.cos.ap-chengdu.myqcloud.com/5ffabac1-7d42-4263-b3ba-71250d217f07.JPG 修改: 选择文件 未选择任何文件
起拍价	100	参考底价	10
参考高价	1000	单品保证金	100
加价幅度	100		

ModelForm默认值展示

提交，拍品信息

拍品规格

ajax，添加到数据库 再 展示到页面

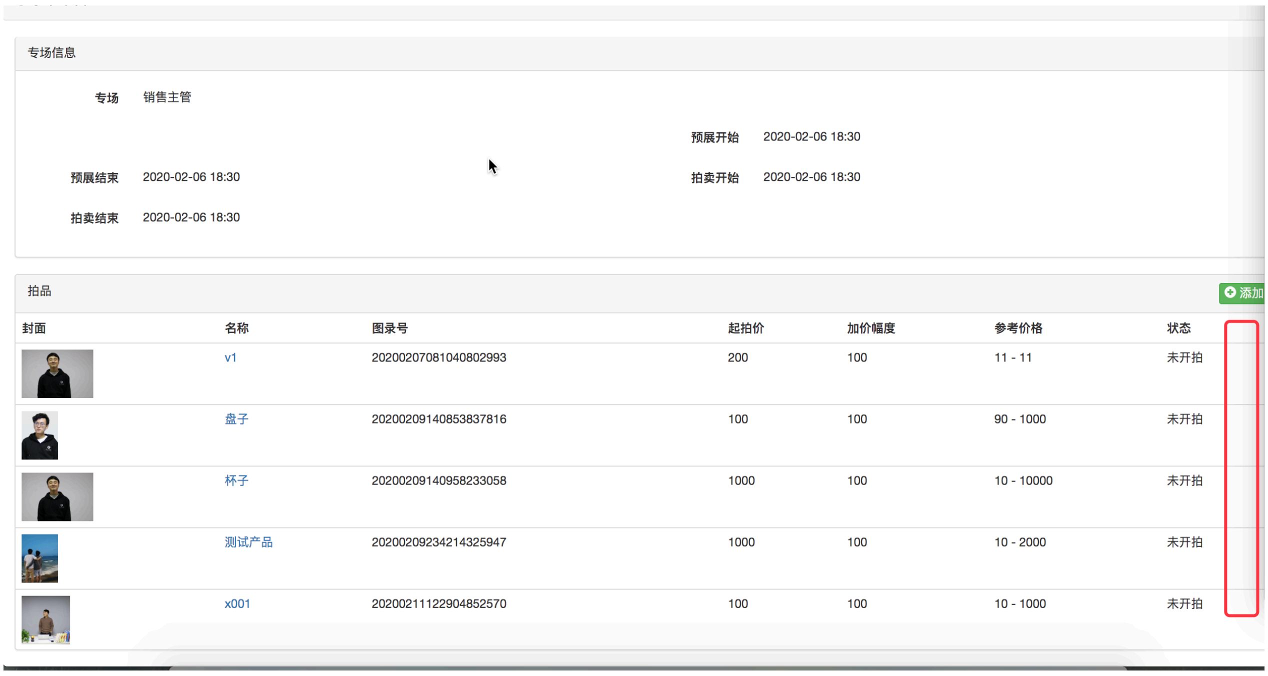
项	对应值	操作
k2	v2	删除 ajax，删除规格
年份	1990	删除

拍品详细图片

本地添加一个上传图片的位置

图片	是否轮播	操作
	✓	删除
	☐	删除
	✓	删除

2.4 删除拍品&规格&图片



专场信息

专场	销售主管		
		预展开始	2020-02-06 18:30
预展结束	2020-02-06 18:30	拍卖开始	2020-02-06 18:30
拍卖结束	2020-02-06 18:30		

拍品

封面	名称	图录号	起拍价	加价幅度	参考价格	状态
	v1	20200207081040802993	200	100	11 - 11	未开拍
	盘子	20200209140853837816	100	100	90 - 1000	未开拍
	杯子	20200209140958233058	1000	100	10 - 10000	未开拍
	测试产品	20200209234214325947	1000	100	10 - 2000	未开拍
	x001	20200211122904852570	100	100	10 - 1000	未开拍

2.5 知识点

- 通过ajax上传文件

```
注意:之前页面向后台提交时通过Form表单实现。 现在是通过ajax向后台发送表单+文件数据。

<form id='mmm'>
    <input type="text" name='user' id='f1'/>
    <input type="file" name='avatar' id='f2'/>
    <input type="checkbox" />
</form>
/*
var formData = new FormData();
formData.append('k1',$('#f1').val())
formData.append('k2',$('#f2')[0].files[0])
*/

// 找到mmm标签中所有的字段并构成成适合的形式append到FormData对象。
// 注意：如果表单中有checbox或radio标签且未选中，则FormData不会构造到自己的数据中。
var formData = new FormData($('#mmm')[0]);

$.ajax({
    url:'..',
    type:"post",
    data:formData,
    cache: false,
    contentType: false,
    processData: false,
    success:function(res){
        console.log(res)
    }
})
```

- 好看的上传按钮 & 图片预览 通过双层结构实现

```
上层: input type=file标签, 透明度0
下层: img标签
```

```
<style>
    .file-view {
        height: 80px;
        width: 80px;
        padding: 2px;
        border: 1px dotted #dddddd;
        position: relative;
    }

    .file-view .view-file {
        position: absolute;
        width: 100%;
        height: 100%;
        opacity: 0;
        z-index: 1001
    }

    .file-view .view-img {
        height: 100%;
        width: 100%;
        border: 0;
        overflow: hidden;
    }
</style>
<div class="file-view">
    <input class="view-file" type="file" name="img">
    <img class="view-img" src="{% static 'web/images/default-image.png'
%}">
</div>
```

```
 $(function () {
     bindChangeImageFile();
 });
function bindChangeImageFile() {
    $('#areaImage').on('change', '.view-file', function () {
        var fileObject = $(this)[0].files[0];
        var blob = window.URL.createObjectURL(fileObject);

        $(this).next().attr('src', blob);
        $(this).next().load(function () {
            window.URL.revokeObjectURL(blob);
        })
    })
}
```

扩展：预览的实现基本上由 本地预览 + 先上传后预览。

https://www.cnblogs.com/wupeiqi/articles/9322297.html

3. 开发任务

自己新创建一个django项目，并且在django项目中继承celery。

- url1， 创建一个 立即执行 的任务。

  ```
  result = x1.delay(1,4)
  result.id
  ```

- url2，创建一个 1分钟之后再执行 的任务。

  ```
  result = x2.apply_async(args=[1, 3], eta=时间) # 注意：时间必须utc时间
  result.id
  ```

https://www.cnblogs.com/wupeiqi/articles/8796552.html