

关于线上直播

- 直播工具
 - 老师，腾讯课堂（极速版）
 - 学生，腾讯课堂（普通客户端）
- 直播时间

上午 8:30 ~ 12:30 上课，班主任点名

- 共享互动文档

【腾讯文档】全栈24期直播课堂
<https://docs.qq.com/sheet/DUkZLb3RTSnZISGnn?c=E8A0A0>

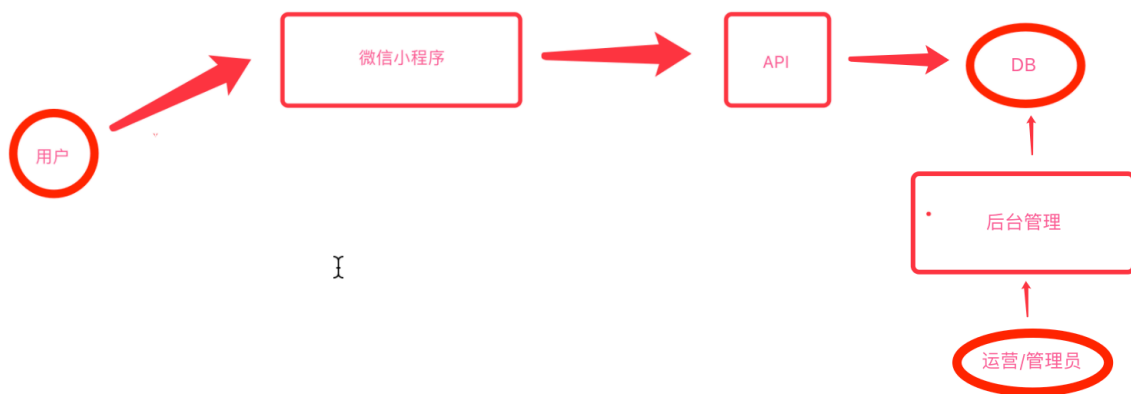
- 昨日掌握情况
 - 今日知识点
 - ajax
 - jquery
 - django
 - 作业完成情况
- 作业

各自把作业提交到git仓库(码云、github)
作业地址放在共享文档中 或 发给我。

day01 微信小程序开发

1. 项目回顾

此项目是由【小程序】 【API】 【后台管理三部分组成】



1.2 小程序

提供在微信中可以方便访问的页面，从功能&知识点上划分：

- 首页，展示新闻动态。
 - 数据双向绑定
 - 数据展示
 - 重定向
 - 瀑布流布局展示
 - 上翻。相当于分页，为提升效率我们记录ID的最小值，翻页取比ID小的10条数。
 - 用传统的第N页需要自己处理重复数据。
 - 假设数据300w条数，传统分页会越翻越慢 / ID极限取值
 - 下拉。记录ID的最大值
- 发布页，新闻动态的发布。
 - 调用 微信小程序API + 预览 + 上传 + 进度条 + 删除图片 + js闭包
 - 将图片上传到腾讯对象存储中 COS。
 - 进度条是如何实现？腾讯对象存储 COS api中有回调函数+组件
 - js闭包

```
for(var i;i<10;i++){
  (function(arg){
    wx.request({
      ....
      success:function(res){
        console.log(arg)
      }
    })
  })(i);
}
// 上传示例失败
```

闭包需求：请通过微信小程序实现想某个api发送5个http请求，获取响应之后，将结果添加到指定的字典中。

```
function func(){
    dict = {
        0:null,
        1:null,
        2:null,
        3:null,
        4:null
    }
    for(var idx; idx<5; idx++){
        // 同步，第一个循环请求进来一直夯住等待结果到来，执行success回调函数。正
        确
        // 异步，所有循环都会执行相当于5个请求并发送出去，此时idx=4，之后如果有结
        果返回。错误

        // 通过闭包（通过函数制造出来）可以解决异步问题
        (function(value){
            wx.request({
                url:".....",
                success:function(res){
                    // res.data
                    dict[value] = res.data
                }
            })
        })(idx)
    }
}

func()
```

- 微信API 选择位置
- 选择话题，父子页面传值。

发布页(父) -> 话题(子)

通过页面调用栈找到响应的页面对象，然后调用对象.setData就可以对父页面进行操作值。

- 登录页，用户注册和登录。
 - 数据双向绑定
 - 短信验证码

腾讯短信服务API (V3版本)

限制: 1个小时10条; 1天30条 (仁慈) - 后台设置
redis的字符串数据类型, 存储短信。

为什么用redis存储短信? 因为他默认支持超时, 性能好(保存在内存中)

- 用户登录成功之后返回token (也可以是jwt)
- 小程序如何存储token?

app.js

操作本地存储

- 微信API获取当前用户信息, 官网推荐 (button+getUserInfo)
- 拍卖专场, 专场列表。
 - api获取数据列表展示
- 专场详细, 拍品列表。
 - api获取数据+数据列表 展示
- 缴纳保证金
 - 业务流程?

单场保证金&全场保证金

点击支付, 调用支付接口

- 账号: 企业小程序开发者账号 + 商户平台 = 关联
- 流程

- 小程序向api后台发送请求, 申请支付
- api后台
 - 生成prepay_id, 商品数据+签名 (key根据ascii排序+md5加密); xml格式数据

登录时候, 会获取用户在微信中的唯一ID。

- 得到数据, 根据prepay_id在进行构造数据+签名; 得到的数据返回给小程序
- 小程序生成二维码

支付成功之后微信会去调用咱们回调地址, 用于保证金状态变更为 已支付。

- 如果支付成功之后宕机? 【经常问到】

微信会在24小时以内不间断给我们一直发消息。

- 余额支付
- 竞拍

缴纳保证金之后就可以参与竞拍。

- 事务 + 锁 (seleted_for_update)

1.3 API

- 遵循restful规范
- 编写auth组件实现用户认证

登录访问
无需登录方法
`request.user/request.auth`

- 编写视图继承过一些类?

APIView
ListAPIView/CreateAPI/....
Mixin,ListModelView, 如果继承他则as_view({get:'list'...})

- 视图类中自定义钩子

queryset
序列化类
过滤器
perform_save
get_authentication():
 return [对象, 对象,]

drf组件非常牛逼, 可扩展性特别强。

- 序列化

- 表单校验
 - 自定义钩子 validate_
- 对象序列化
 - 自定义钩子 get_
- 通用
 - 自定义字段 & 参数(source、readonly)
 - nest支持嵌套(高级)

1.4 后台管理

- 数据基本管理
- 定时任务 (celery)

创建专场时，需要使用celery创建3个定时任务

1. 预展时间，需要将当前 专场 + 拍品 状态变更为预展。
2. 拍卖时间，需要将当前 专场 + 拍品 状态变更为拍卖中。
3. 拍卖结束，竞拍架构进行对比将最高级处理为拍得商品（订单/退还保证金）。

2. 后台管理+celery




给运营人员&管理人员对平台的数据进行数据维护和管理。尤其是对于拍卖的业务 专场、拍品、详细、图片操作和管理。

2.1 基本后台管理

技术栈：django/html/css/bootstrap/上传文件

2.1.1 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
	销售主管	2020-02-06 10:30	2020-02-06 10:30	未开拍	编辑 删除
	第二场 贵和祥茶道专场	2019-12-26 01:20	2019-12-26 01:20	拍卖中	编辑 删除
	第一场 贵和祥茶道专场	2019-12-26 01:19	2019-12-26 01:19	预展中	编辑 删除

- 添加

小茶书平台 拍卖管理 Link	
拍卖专场 保存	
标题	<input type="text"/>
封面	<input type="text" value="选择文件 未选择任何文件"/>
预展开始时间	<input type="text"/>
预展结束时间	<input type="text"/>
拍卖开始时间	<input type="text"/>
拍卖结束时间	<input type="text"/>
全场保证金	<input type="text" value="1000"/>

- 编辑

小茶书平台 拍卖管理 Link

标题

销售主管

封面

目前: <https://auction-1251317460.cos.ap-chengdu.myqcloud.com/c670c9a1-a74d-4acf-b041-17ad44fbacc4.JPG>
 修改:
 未选择任何文件

预展开始时间

2020/02/06 10:30

预展结束时间

2020/02/06 10:30

拍卖开始时间

2020/02/06 10:30

拍卖结束时间

2020/02/06 10:30

全场保证金

1000

保存

● 删除

127.0.0.1:8000/auction/list/ ☆ 🔍 🌐 🔒 👤 ⋮

小茶书平台 拍卖管理 Link

127.0.0.1:8000 显示
确认删除?

取消 确定

拍卖专场列表

添加

封面	专场	预展时间	拍卖时间	状态	操作
	销售主管	2020-02-06 10:30	2020-02-06 10:30	未开拍	编辑 删除
	第二场 贵和祥茶道专场	2019-12-26 01:20	2019-12-26 01:20	拍卖中	编辑 删除
	第一场 贵和祥茶道专场	2019-12-26 01:19	2019-12-26 01:19	预展中	编辑 删除

需要关注的技术点：

1. 选择时间，通过datetimepicker.js 插件
2. 上传图片
 - 图片上传到腾讯对象存储，返回一个图片路径
 - 讲记录保存到数据库的封面字段中

```
#!/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 = 'xxx' # 替换为用户的 secretId
    secret_key = 'fff' # 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)

```

3. 编辑页面

添加图片：在后台获取的文件对象是 TempFile 对象。

编辑图片：

- 未修改，文件对象是 Field 对象。
- 已修改，文件对象是 TempFile 对象。

2.1.2 拍品列表

- 预备知识
 - 前端页面js如何实现图片预览
 - 前端页面js + ajax，文件上传（FormData对象）
- 功能开发
 - 列表

小茶书平台

拍卖管理

Link

专场信息

专场

销售主管

预展开始

2020-02-06 10:30

预展结束

2020-02-06 10:30

拍卖开始

2020-02-06 10:30

拍卖结束

2020-02-06 10:30

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


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

- 添加

小茶书平台

拍卖管理

Link

销售主管 - 创建拍品

保存

拍品名称

起拍价

参考高价

加价幅度

100

拍品封面

选择文件 alex.png

参考底价

单品保证金

100

拍品规格

项


值

添加

项	对应值	操作
年份	2000	删除

拍品详细图片

添加

图片	是否轮播	操作
	<input type="checkbox"/>	删除

修改

删除

day01 开发任务

- 专场管理
- 技术点

- 前端页面js如何实现图片预览
- 前端页面js + ajax，文件上传（FormData对象）

3.其他逻辑开发

4.部署

