

# day07 支付 & 项目部署

## 今日概要

- 支付，小程序端
- 项目部署
  - 运营管理+API 部署
  - 小程序的部署
- 项目总结

## 今日详细

### 1.小程序端支付

用户在小程序中选择：订单、优惠券、保证金、地址、支付方式，点击按钮向后台API发送请求，后台接收数据进行数据校验，数据校验通过完成支付的环节。

- 查看订单支付页面

```
var userInfo = app.globalData.userInfo;
wx.request({
  url: api.Pay + this.data.orderId + '/',
  header: {
    Authorization: userInfo ? "token " + userInfo.token : ""
  },
  method: 'GET',
  dataType: 'json',
  responseType: 'text',
  success: (res) => {
    this.setData({
      order: res.data,
      payPrice: res.data.price
    })
  },
  fail: function(res) {},
  complete: function(res) {},
})
```

```
class PayModelSerializer(serializers.ModelSerializer):
    user_balance = serializers.IntegerField(source='user.balance')

    auction = serializers.CharField(source='item.auction_id')
```

```

# 拍品
item = serializers.SerializerMethodField()

# 保证金
deposit = serializers.SerializerMethodField()

# 是否有优惠券
coupon = serializers.SerializerMethodField()

# 支付方式
pay_method = serializers.SerializerMethodField()

class Meta:
    model = models.Order
    exclude = ['uid', 'twenty_four_task_id', 'user']

def get_deposit(self, obj):
    return PayDepositModelSerializer(instance=obj.deposit).data

def get_coupon(self, obj):
    user_object = self.context['request'].user
    exists = models.UserCoupon.objects.filter(
        user=user_object, status=1,
        coupon__auction=obj.item.auction_id).exists()

    context = {
        'id': None,
        'has': exists,
        'text': '请选择优惠券' if exists else '无',
        'money': 0
    }
    return context

def get_pay_method(self, obj):
    balance = self.context['request'].user.balance
    info = {
        'selected': 1,
        'choices': [
            {'id': 1, 'text': '余额 (%s) ' % balance},
            {'id': 2, 'text': '微信支付'},
        ]
    }
    return info

def get_item(self, obj):
    return {
        'title': obj.item.title,
        'cover': obj.item.cover.name,
    }

```

```

        'uid': obj.item.uid
    }

class PayView(RetrieveAPIView):
    authentication_classes = [UserAuthentication, ]
    # queryset = models.Order.objects.filter(status=1)
    serializer_class = PayModelSerializer

    def get_queryset(self):
        return
        models.Order.objects.filter(status=1, user=self.request.user)

```

- 选择优惠券、保证金、地址、支付方式选项，提交到API支付
  - 失败，错误提醒
  - 成功，跳转到成功页面。

**示例：【支付完成.zip】**

## 2.项目部署

### 2.1 运营中心&API

如果作为小程序的API端想要进行部署，必须是HTTPS协议。

目标：将一个django程序按照https进行部署。

#### 2.1.1 云服务器安装nginx

找一台云服务器（外网可以访问的服务），别用自己的虚拟机。

```
yum install nginx -y
```

修改配置文件

默认不修改，用户访问看到的是用户的默认页面。

```
# vim /etc/nginx/nginx.conf
```

修改之后，按照修改之后的配置进行显示。

启动nginx服务（系统不同命令有所不同）

```
/etc/init.d/nginx start
/etc/init.d/nginx restart
```

#### 2.1.2 nginx支持https

- 申请SSL证书（阿里云、腾讯云）

阿里云

购物车 工单 备案 简体中文 42466\*\*\*\*

云盾证书服务(包年)

域名类型

单域名 (1个域名)

保护一个域名或者一个子域名, 例如 domain.com、ssl.domain.com、ssl.ssl.domain.com 分别需要买3张单域名证书。  
备注: 购买domain.com单域名证书, 不会包含 ssl.domain.com 等子域名, 如果需要支持所有二级或三级域名 (\*.domain.com), 请购买通配符证书。

选择证书类型

专业版OV 增强版OV Pro 高级版EV 增强版EV Pro 通配符DV 免费版(个人) DV

个人型SSL证书, 保护一个域名。浏览器上有https提示并有绿锁标记。快速签发, 适合个人和小微企业, 支持个人/企业申请。一个阿里云帐户最多签发20张。

选择品牌

Symantec

赛门铁克 (Symantec) 是全球安全领域的领导者, 服务全球超过35个国家, 拥有众多企业、政府和个人用户。其SSL证书属于Digicert旗下品牌, 赛门铁克的PKI基础架构包括军事级数据中心灾难恢复站点, 是SSL行业中受认可和值得信赖的品牌。

选择域名个数

1个

购买数量

1

总配置费用 ¥0.00 立即购买

SSL证书

概览

一键式HTTPS 资产中心

SSL证书

证书总数 3 已签发的证书 2 待申请的证书 1 即将过期的证书 1 审核失败的证书 0 已过期的证书 0

购买证书 上传证书

已付款 1 全部品牌 证书域名

未签发

证书	品牌/类型	状态	绑定域名	有效期限	已部署	操作
Symantec 免费版 SSL 实例: cas-cn-4591j2b3w2do 标签:未设置标签	Symantec 免费版 SSL	已付款	--	1年	--	证书申请 升级

1个证书即将到期, 建议您立即续费 (已完成续费请忽略): 立即续费

公告  
新冠疫情期间购买证书赠送服务  
订单退款说明  
新版本控制台使用指南  
SSL证书安装指南(包含多种服务器)  
Symantec SSL数字证书升级重签提示

常见问题  
SSL证书续费和到期新购

证书申请

填写申请 验证信息

证书绑定域名  
www.pythnav.cpm

申请人姓名  
武沛齐

申请人手机号  
89

申请人邮箱  
n

所在地  
中国大陆 / 北京市

域名验证方式  
您绑定的域名已在阿里云的云解析服务中, 建议您采用DNS验证

下一步 取消

- 下载证书文件

```
xxxx.key
xxxx.pem
```

- 在nginx中配置https证书

```
worker_processes 1;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;

    #gzip on;

    # HTTPS server
    server {
        listen 443 ssl;
        server_name localhost;
        # 证书上传服务器, 解压 -> 证书地址
        ssl_certificate /data/ssl/3468134_s24.pythonav.com.pem;
        ssl_certificate_key /data/ssl/3468134_s24.pythonav.com.key;

        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout 5m;

        ssl_ciphers HIGH:!aNULL:!MD5;
        ssl_prefer_server_ciphers on;

        location / {
            root html;
            index index.html index.htm;
        }
    }

    server {
        listen 80;
        server_name localhost;
        rewrite ^(.*)$ https://$host$1 permanent;
    }
}
```

```
/etc/init.d/nginx restart
```

- 域名解析 将域名[www.pythonav.com](http://www.pythonav.com)解析到当前这台云服务器（47.93.4.198） 将子域名[s24.pythonav.com](http://s24.pythonav.com) 解析到当前这台云服务器（47.93.4.198）

云解析DNS / 域名解析 / 解析设置

### 解析设置 pythonav.com

未查询到域名DNS服务器信息, 请检查DNS服务器设置或域名实名认证相关。 [展开](#)

添加记录 导入/导出 请求量统计 新手引导

全部记录 精确搜索 输入关键字 高级搜索

主机记录	记录类型	解析线路 (isp)	记录值	TTL	状态	备注	操作
_dnsauth.s24	TXT	默认	202002160000005sg2wto568tkftajnr ryf66ziaqk28tpxv0qjp9frnei9b732	10 分钟	正常	自动添加的, SSL填写域名	修改 暂停 删除 备注
s24	A	默认	47.93.4.198	10 分钟	正常		修改 暂停 删除 备注

- 访问nginx

https://s24.pythonav.com

Welcome to **nginx** on EPEL!

This page is used to test the proper operation of the nginx HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default index.html page that is distributed with nginx on EPEL. It is located in /usr/share/nginx/html.

You should now put your content in a location of your choice and edit the root configuration directive in the nginx configuration file /etc/nginx/nginx.conf.

NGINX fedora

## 2.1.3 部署django项目（nginx做反向代理）

- django代码打包上传到云服务器
- 为django程序创建虚拟环境
- 进入虚拟环境 & 安装依赖的模块
- 安装uwsgi
- 进行uwsgi的配置

```
[uwsgi]  
socket = 127.0.0.1:8080  
chdir = /data/auction/  
wsgi-file = auction/wsgi.py  
processes = 1  
threads = 4  
virtualenv = /env/auction/
```

- 启动uwsgi（shell） 执行 ./reboot.sh

```
#!/usr/bin/env bash
```

```

echo -e "\033[34m-----wsgi process-----
\033[0m"

ps -ef |grep auction_uwsgi.ini | grep -v grep

sleep 0.5

echo -e '\n-----going to close-----'

ps -ef |grep auction_uwsgi.ini | grep -v grep | awk '{print $2}' | xargs
kill -9

sleep 0.5

echo -e '\n-----check if the kill action is correct-----'

/usr/bin/uwsgi --ini auction_uwsgi.ini & >/dev/null

echo -e '\n\033[42;1m-----started...-----
-\033[0m'
sleep 1

ps -ef |grep auction_uwsgi.ini | grep -v grep

```

- 关闭

```

#!/usr/bin/env bash

echo -e "\033[34m-----wsgi process-----
\033[0m"

ps -ef |grep auction_uwsgi.ini | grep -v grep

sleep 0.5

echo -e '\n-----going to close-----'

ps -ef |grep auction_uwsgi.ini | grep -v grep | awk '{print $2}' | xargs
kill -9

sleep 0.5

```

- nginx + uwsgi关联，修改nginx的这配置进行关联。

```

worker_processes 1;

events {

```

```

worker_connections 1024;
}

http {
    include      mime.types;
    default_type application/octet-stream;

    sendfile      on;

    keepalive_timeout 65;

    #gzip on;

    upstream django {
        server 127.0.0.1:8080;
    }

    # HTTPS server
    server {
        listen      443 ssl;
        server_name localhost;

        ssl_certificate      /data/ssl/3468134_s24.pythonav.com.pem;
        ssl_certificate_key  /data/ssl/3468134_s24.pythonav.com.key;

        ssl_session_cache    shared:SSL:1m;
        ssl_session_timeout  5m;

        ssl_ciphers  HIGH:!aNULL:!MD5;
        ssl_prefer_server_ciphers on;

        location / {
            uwsgi_pass  django;
            include     uwsgi_params;
        }
    }

    server {
        listen 80;
        server_name localhost;
        rewrite ^(.*)$ https://$host$1 permanent;
    }
}

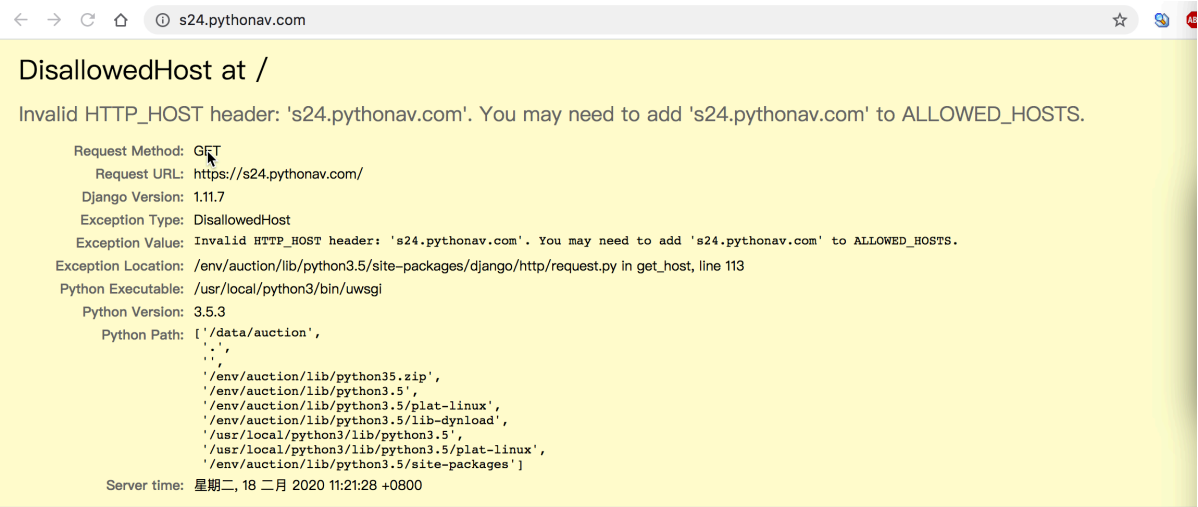
```

- nginx做重启



```
/etc/init.d/nginx restart
```

- 用户访问 (nginx做反向成功)



## 2.2.3 django项目

在django项目部署时，一般需要关注一些配置信息：

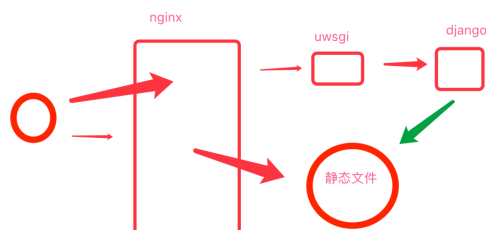
```
ALLOWED_HOSTS = ["*"]  
DEBUG = False  
可能还有数据库  
关于静态文件：  
    STATIC_URL = '/static/'  
    STATIC_ROOT = "/data/auction_static/"
```

默认django的静态文件可以放在根目录下 static + 每个app目录的static文件件。

在项目部署时，需要去收集各处的静态文件然后放置在同一的位置。

```
STATIC_ROOT = "/data/auction_static/"
```

```
python manage.py collectstatic
```



### 处理方式一：（不推荐）

在线上的代码中直接进行修改配置文件

```
ALLOWED_HOSTS = ["*"]
DEBUG = False
STATIC_ROOT = "/data/auction_static/"
```

收集静态文件

```
python manage.py collectstatic
```

修改nginx的配置

```
worker_processes 1;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;

    sendfile on;

    keepalive_timeout 65;

    #gzip on;

    upstream django {
        server 127.0.0.1:8080;
    }

    # HTTPS server
    server {
        listen 443 ssl;
        server_name localhost;

        ssl_certificate /data/ssl/3468134_s24.pythonav.com.pem;
        ssl_certificate_key /data/ssl/3468134_s24.pythonav.com.key;

        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout 5m;

        ssl_ciphers HIGH:!aNULL:!MD5;
        ssl_prefer_server_ciphers on;

        location /static {
            alias /data/auction_static;
        }
    }
}
```

```

        location / {
            uwsgi_pass    django;
            include        uwsgi_params;
        }
    }

    server {
        listen 80;
        server_name localhost;
        rewrite ^(.*)$ https://$host$1 permanent;
    }
}

```

## 处理方式二：（推荐）

基于local\_settings.py的方式。

- 本地开发环境的settings.py中

```

try:
    from .local_settings import *
except ImportError:
    pass

```

- 本地开发环境的配置放在 local\_settings.py 中
- 本地开发配置 .gitignore，让git忽略 local\_settings.py

local\_settings.py

- 本地提交代码，除了local\_settings.py其他均已提交
- 线上拉取最新代码（没有local\_settings.py）
- 线上创建一个local\_settings.py且将线上配置写在此文件

```

ALLOWED_HOSTS = ["*"]
DEBUG = False
STATIC_ROOT = "/data/auction_static/"

```

- 线上执行收集静态文件

```
python manage.py collectstatic
```

- 修改nginx配置文件

```
location /static {
    alias /data/auction_static;
}
```

- nginx重启

```
/etc/init.d/nginx restart
```

- uwsgi重启

```
./reboot.sh
```

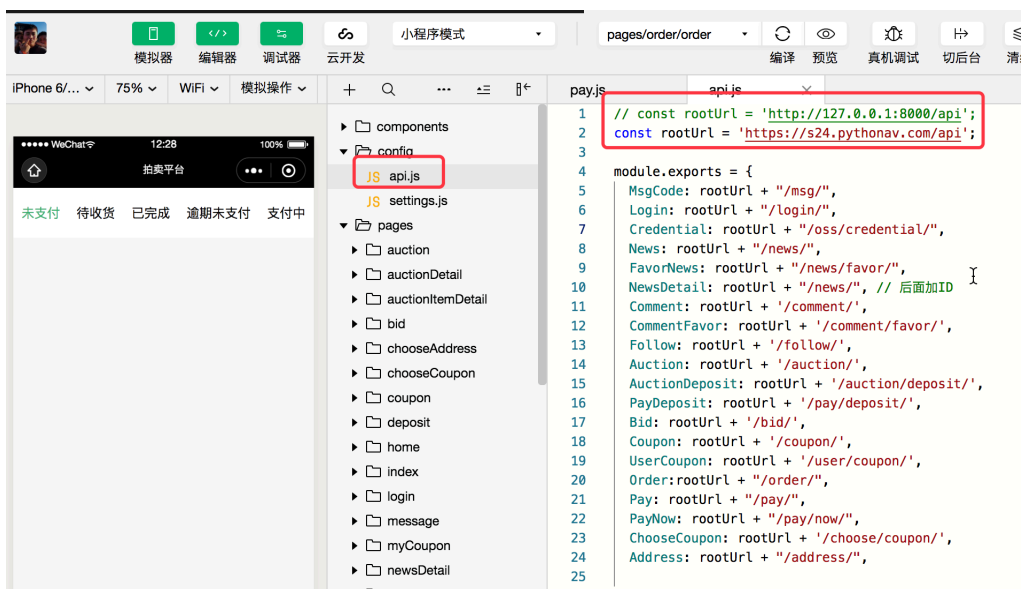
## 2.2 小程序部署

### 2.2.1 配置服务器域名

小程序要求程序中使用到所有外部链接（HTTPS）必须将域名提前配置到 微信平台。

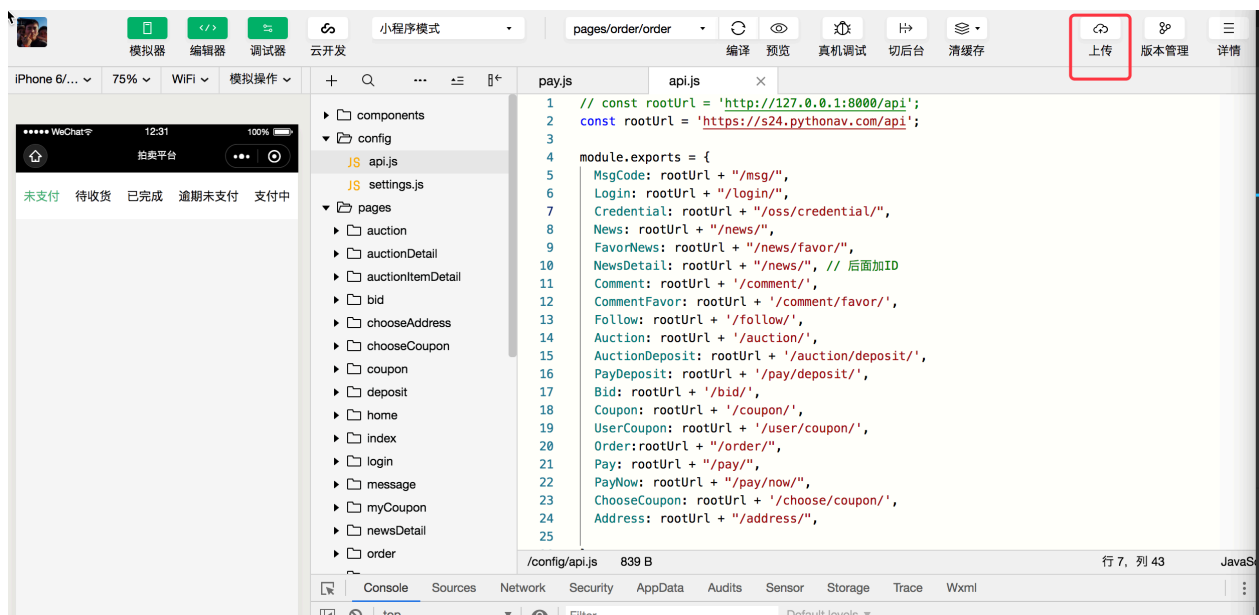


### 2.2.2 小程序api路径替换



## 2.2.3 上传小程序代码

如果想让小程序被所有人在微信中使用，需要将小程序代码进行 上传&审核，审核通过之后才能在微信中找到该小程序。



需要对小程序进行描述和配置：



提交审核（1-2天）：



### 3.项目总结

- 项目介绍

xxx项目是一个微信小程序实现公共拍卖平台，项目主要由三部分组成，微信小程序端，提供用户交互平台，并提供了拍卖和动态管理，增加用户粘性；运营中心主要提供平台的数据支撑及管理，对于定时任务和拍卖等其他业务提供支持；API接口，遵循restful规范实现统一API接口。

- 功能点：

- 通过Authentication及腾讯sms v3 SDK实现用户认证，并且在小程序api中实现临时凭证。
- 集成chooseImage、COS 实现图片信息的上传以及持久化处理。

- 基于js闭包解决默认小程序异步请求的问题，并实现onProgress动态进度和更新。
- 避免权限问题导致项目出错，基于腾讯COS是credential凭证实现权限控制。
- 使用django rest framework框架组件Authentication、FilterBackend、Hook钩子。Serializer完成业务接口的开发。
- 通过 微信支付/余额 实现保证金的缴纳，当拍卖结束时再完成保证金的抵扣、退款、交易记录。
- 内部基于celery的eta、redis的string/list 实现任务的定时处理，并结合redis特性实现验证码超时处理。
- 技术栈 drf / restful API / 对象存储 / 异步请求 / celery / redis / Https / SSL / drf源码

## 小程序项目结束

---

- 我带你讲项目结束。
- 你自己写的怎么样？
  - 要求：2天时间去复习小程序项目
    - 项目跑起来？
    - 10 ~ 20 个问答？

```
https://docs.qq.com/sheet/DUkZLb3RTSnZISGnn?
tdsourcetag=s_macqq_aiomsg&tab=m0uvw0&c=G16A0A0
```

- 2月21号上午再聊
  - 项目问答
  - 代码发布系统补充和总结