

```
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

- 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解析到当前这台云服务器（47.93.4.198） 将子域名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> ☆

打开主页

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 powered by **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;
        }
    }

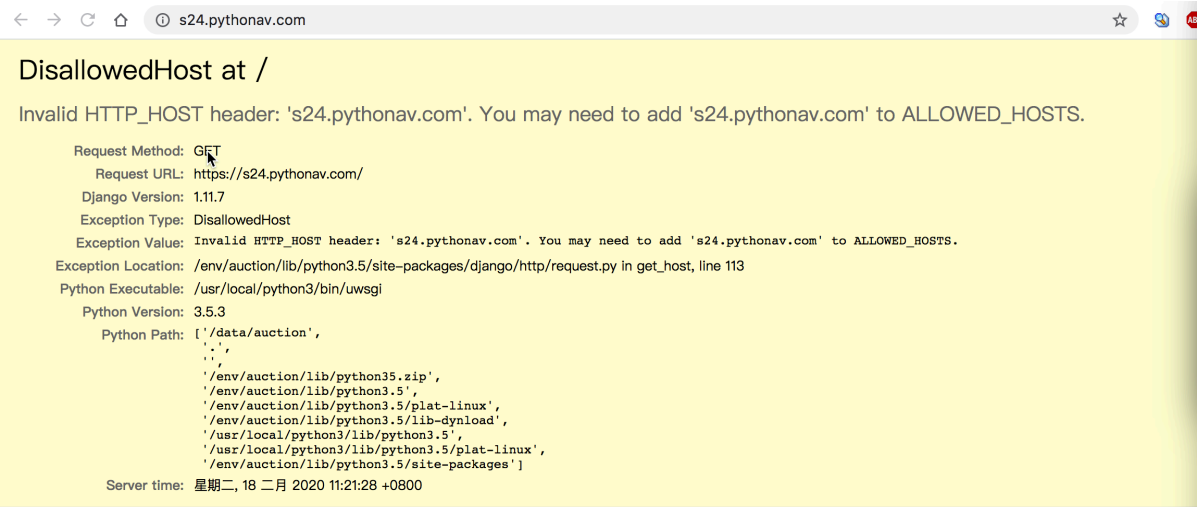
    server {
        listen 80;
        server_name localhost;
        rewrite ^(.*)$ https://$host$1 permanent;
    }
}

```

- 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：

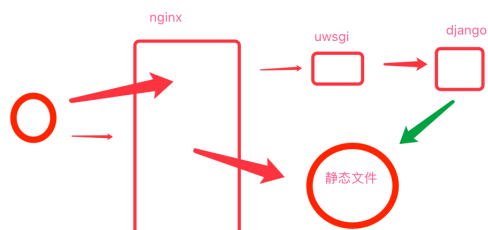
```
ALLOWED_HOSTS = ["*"]
DEBUG = False
可能还有数据库
关于静态文件：
    STATIC_URL = '/static/'
    STATIC_ROOT = "/data/auction_static/"
```

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
```



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
```

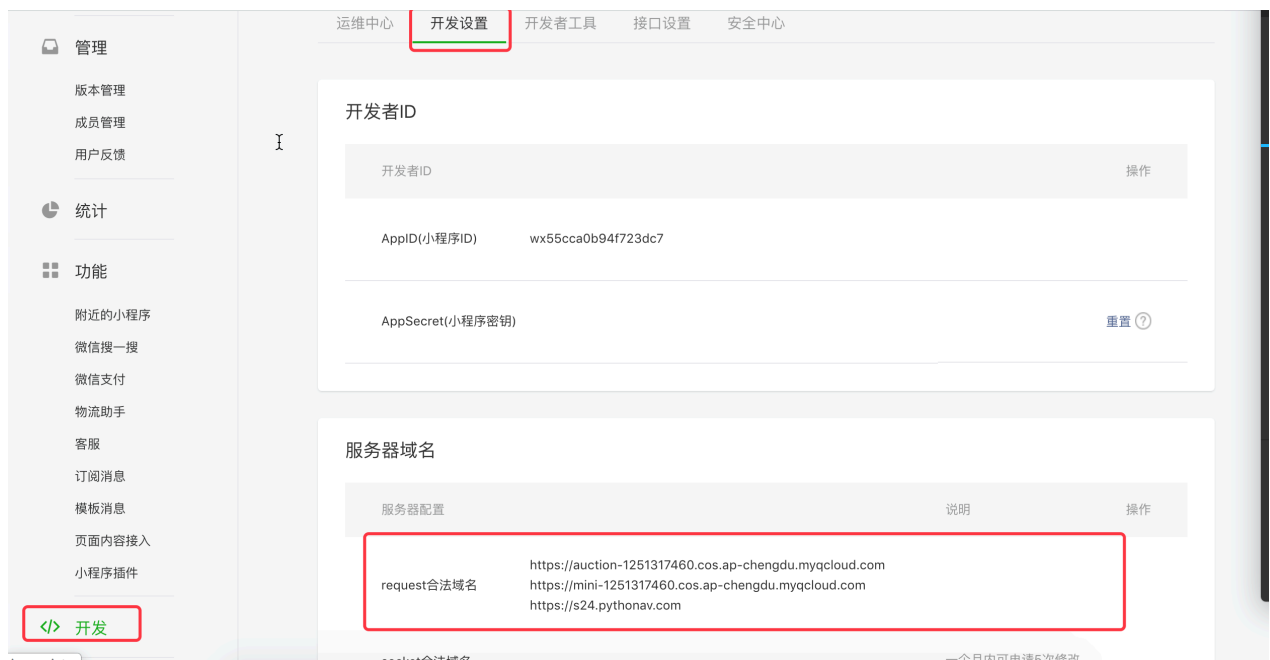
- uwsgi重启

```
./reboot.sh
```

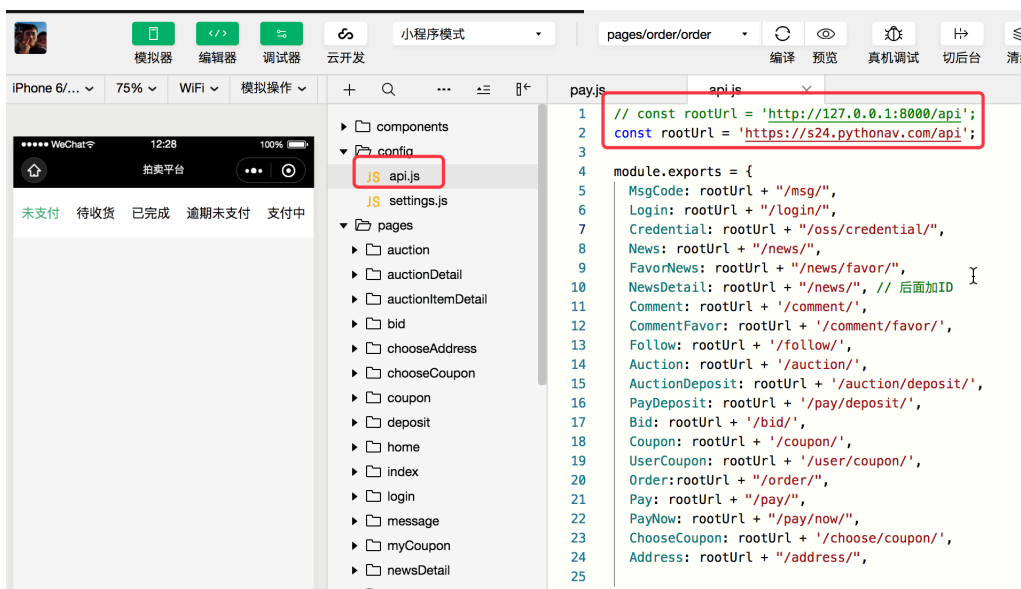
2.2 小程序部署

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 。

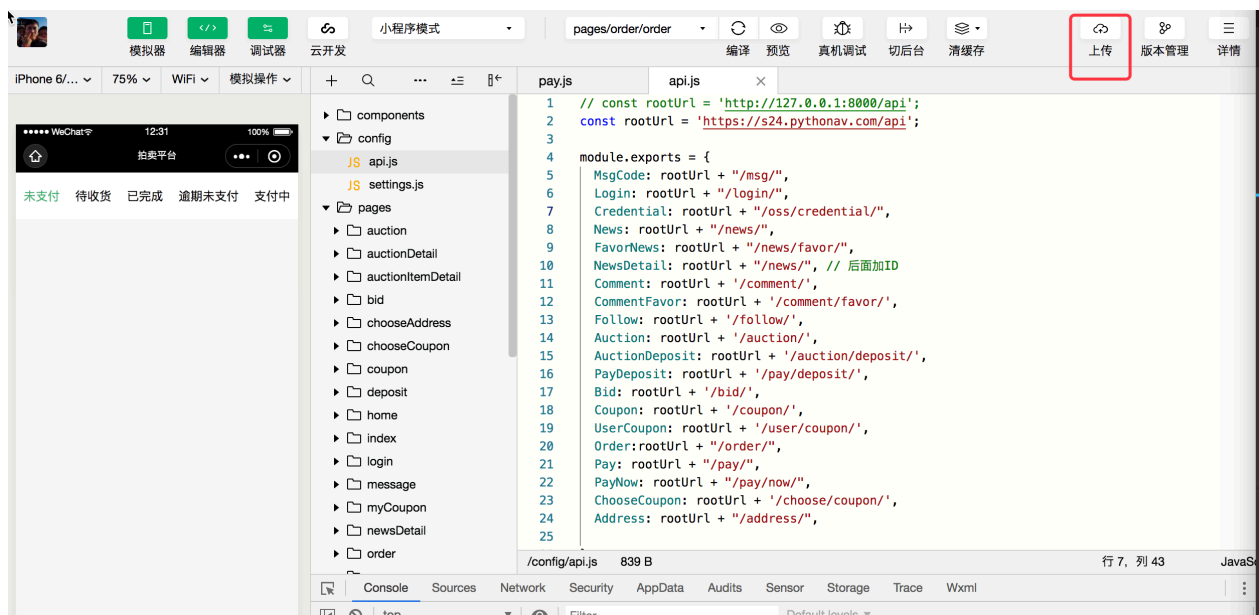


2.2.2 小程序api路径替换



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