

## HANDY项目服务器应用部署文档

前提：环境准备

### 1.安装 django

安装 pip

使用 pip 安装 django

### 2.创建 HomeServiceMall 应用并启动 Web 服务器

创建项目

修改配置文件

启动 django 自带的 Web 服务器

### 3.创建简单的测试页面

关闭 Web 服务器

创建 views.py

添加测试用视图函数

修改 urls.py 文件权限

添加路由配置

测试 url 访问是否正常

关闭 Web 服务器

### 4.将项目文件移植到HomeServiceMall

将调试模式DEBUG改为False，并将允许访问的主机改为所有主机ALLOWED\_HOSTS=[\*]

配置APP定义、中间件、Templates模板等

MySQL数据库配置

支付宝配置

邮箱配置

静态文件配置

用户认证与国际化等配置

### 5.部署 uwsgi 和 nginx

使用 pip 安装 uwsgi

配置 uwsgi

启动 uwsgi

安装 nginx

添加 nginx 配置文件

重启 nginx 服务

### 6.测试 nginx + uwsgi + django 是否工作正常

测试页面是否能够正常访问

测试结果

应用部署成功

---

# HANDY项目服务器应用部署文档

## 前提：环境准备

---

1. **服务器操作系统**：Ubuntu 16.04.1 LTS (GNU/Linux 4.15.0-142-generic x86\_64)
  2. **服务器配置**:1核 2GB 1Mbps 系统盘：50GB高性能云硬盘
  3. **公网IP地址**:119.45.125.97
- 

## 1.安装 django

---

### 安装 pip

```
1  sudo apt install python-pip -y
```

### 使用 pip 安装 django

```
1  sudo pip install django
```

## 2.创建 HomeServiceMall 应用并启动 Web 服务器

---

### 创建项目

```
1  cd /data
2  sudo django-admin startproject HomeServiceMall
```

### 修改配置文件

修改 /data/HomeServiceMall/HomeServiceMall/settings.py 文件权限为其它人可写

```
1  sudo chmod 666 /data/HomeServiceMall/HomeServiceMall/settings.py
```

编辑 /data/HomeServiceMall/HomeServiceMall/settings.py

将 `ALLOWED_HOSTS = []` 修改为 `ALLOWED_HOSTS = ['*']` , 这样可以允许通过 ip 访问

## 启动 django 自带的 Web 服务器

```
1 cd HomeServiceMall
2 sudo python manage.py runserver 0.0.0.0:8080
```

使用浏览器访问

```
1 http://119.45.125.97:8080/
```

看到如下页面，表示 django 服务已经部署成功

**It worked!**  
Congratulations on your first Django-powered page.

Next, start your first app by running `python manage.py startapp [app_label]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!

## 3.创建简单的测试页面

### 关闭 Web 服务器

按 **Ctrl + C** 关闭 Web 服务器

### 创建 views.py

创建文件 `/data/HomeServiceMall/HomeServiceMall/views.py`，并修改权限为其它人可写

```
1 sudo touch /data/HomeServiceMall/HomeServiceMall/views.py
2 sudo chmod 666 /data/HomeServiceMall/HomeServiceMall/views.py
```

### 添加测试用视图函数

编辑 `/data/HomeServiceMall/HomeServiceMall/views.py`

添加如下内容：

```
1 from django.http.response import HttpResponseRedirect
2 class TestView(View):
3     def get(self):
4         return HttpResponseRedirect("ok")
```

按 **Ctrl + S** 保存

### 修改 urls.py 文件权限

修改 `/data/HomeServiceMall/HomeServiceMall/urls.py` 文件权限为其它人可写

```
1 sudo chmod 666 /data/HomeServiceMall/HomeServiceMall/urls.py
```

## 添加路由配置

编辑 /data/HomeServiceMall/HomeServiceMall/urls.py

将

```
1  urlpatterns = [  
2      url(r'^admin/', admin.site.urls),  
3  ]
```

修改为

```
1  import views  
2  urlpatterns = [  
3      url(r'^admin/', admin.site.urls),  
4      url(r'^test', views.TestView.asView()),  
5  ]
```

按 **Ctrl + S** 保存

## 测试 url 访问是否正常

再次启动 Web 服务器

```
1  sudo python manage.py runserver 0.0.0.0:8080
```

使用浏览器访问下面的 url

```
1  http://119.45.125.97:8080/test
```

## 关闭 Web 服务器

按 **Ctrl + C** 停止 Web 服务器

## 4.将项目文件移植到HomeServiceMall

---

使用完整项目文件覆盖原测试项目

配置HomeServiceMall下的settings.py

**将调试模式DEBUG改为False，并将允许访问的主机改为所有主机**  
**ALLOWED\_HOSTS=[\*]**

```
1  import os  
2  from pathlib import Path  
3  
4  # Build paths inside the project like this: BASE_DIR / 'subdir'.  
5  BASE_DIR = Path(__file__).resolve().parent.parent  
6  
7  # SECURITY WARNING: keep the secret key used in production secret!  
8  SECRET_KEY = 'django-insecure-!i=39zni&$^j%)afnx0h65t6*%@ucj+67s87$w)9=h8s8$hu6'  
9  
10 # SECURITY WARNING: don't run with debug turned on in production!  
11 DEBUG = False  
12
```

```
13 ALLOWED_HOSTS = [*]
14
```

## 配置APP定义、中间件、Templates模板等

```
1  # Application definition
2  INSTALLED_APPS = [
3      'django.contrib.admin',
4      'django.contrib.auth',
5      'django.contrib.contenttypes',
6      'django.contrib.sessions',
7      'django.contrib.messages',
8      'django.contrib.staticfiles',
9      'account',
10     'other',
11     "passport",
12     "services",
13     "shop"
14 ]
15
16
17 SITE_ID = 1
18 MIDDLEWARE = [
19     'django.middleware.security.SecurityMiddleware',
20     'django.contrib.sessions.middleware.SessionMiddleware',
21     'django.middleware.common.CommonMiddleware',
22     'django.contrib.auth.middleware.AuthenticationMiddleware',
23     'django.contrib.messages.middleware.MessageMiddleware',
24     'django.middleware.clickjacking.XFrameOptionsMiddleware',
25
26 ]
27
28 ROOT_URLCONF = 'HomeServiceMall.urls'
29
30 TEMPLATES = [
31     {
32         'BACKEND': 'django.template.backends.django.DjangoTemplates',
33         'DIRS': [BASE_DIR / 'templates']
34     },
35     'APP_DIRS': True,
36     'OPTIONS': {
37         'context_processors': [
38             'django.template.context_processors.debug',
39             'django.template.context_processors.request',
40             'django.contrib.auth.context_processors.auth',
41             'django.contrib.messages.context_processors.messages',
42         ],
43     },
44 ],
45 ]
46 WSGI_APPLICATION = 'HomeServiceMall.wsgi.application'
47
48 LOGIN_URL = "/passport/login/"
49 OPEN_URLS = ['/passport/login/']
50
```

## MySQL数据库配置

```
1  # Database
2  DATABASES = {
3      'default': {
4          'ENGINE': 'django.db.backends.mysql',
5          'NAME': 'django_mysql',
6          'USER': "root", # 服务器数据库账号密码
7          'PASSWORD': "123456",
8          'HOST': "119.45.125.97",
9          'PORT': "3306",
10         'OPTIONS': {'init_command': 'SET default_storage_engine=INNODB;'}
11     }
12 }
13
```

## 支付宝配置

```
1  # Default primary key field type
2  DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
3  ALIPAY_PUBLIC = os.path.join(BASE_DIR, 'keys', 'alipay_public.txt')
4  APP_PUBLIC = os.path.join(BASE_DIR, 'keys', 'app_public.txt')
5  APP_PRIVATE = os.path.join(BASE_DIR, 'keys', 'app_private.txt')
6
```

## 邮箱配置

```
1  EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
2  EMAIL_HOST = "smtp.163.com" # 服务器
3  EMAIL_PORT = 25 # 一般情况下都为25
4  EMAIL_HOST_USER = "lets_go2011@163.com" # 账号
5  EMAIL_HOST_PASSWORD = "NBMBWKT LJ EYOSJLX" # 密码
6  EMAIL_USE_TLS = False # 一般都为False
7  EMAIL_FROM = "lets_go2011@163.com" # 邮箱来自
```

## 静态文件配置

```
1  # Static files (CSS, JavaScript, Images)
2
3  STATIC_URL = '/static/'
4  STATICFILES_DIRS = (
5      os.path.join(BASE_DIR, "static"),
6  )
7  MEDIA_ROOT = os.path.join(BASE_DIR, "static/media")
8
```

## 用户认证与国际化等配置

```
1  # Password validation
2
3  AUTH_PASSWORD_VALIDATORS = [
4      {
5          'NAME':
6          'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
```

```

7     {
8         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
9     },
10    {
11        'NAME':
12        'django.contrib.auth.password_validation.CommonPasswordValidator',
13    },
14    {
15        'NAME':
16        'django.contrib.auth.password_validation.NumericPasswordValidator',
17    },
18    ]
19
20 AUTH_USER_MODEL = 'account.User'
21
22 # Internationalization
23
24 LANGUAGE_CODE = 'zh-hans'
25
26 TIME_ZONE = 'Asia/Shanghai'
27
28 USE_I18N = True
29
30 USE_L10N = True
31
32 USE_TZ = False

```

## 5.部署 uwsgi 和 nginx

### 使用 pip 安装 uwsgi

```
1 sudo pip install uwsgi
```

### 配置 uwsgi

创建文件 /data/HomeServiceMall/uwsgi.ini，并修改权限为其它人可写

```

1 sudo touch /data/HomeServiceMall/uwsgi.ini
2 sudo chmod 666 /data/HomeServiceMall/uwsgi.ini

```

编辑 /data/HomeServiceMall/uwsgi.ini

输入以下内容，并保存

```

1 [uwsgi]
2 chdir = /data/HomeServiceMall
3 module = HomeServiceMall.wsgi
4 socket = 127.0.0.1:8080
5 master = true
6 vhost = true
7 no-site = true
8 workers = 2
9 reload-mercy = 10

```

```
10 vacuum = true
11 max-requests = 1000
12 limit-as = 512
13 buffer-size = 30000
14 pidfile = /tmp/uwsgi.pid
15 daemonize = /tmp/uwsgi.log
```

## 启动 uwsgi

```
1 export PYTHONPATH=/usr/local/lib/python2.7/dist-packages
2 uwsgi --ini /data/HomeServiceMall/uwsgi.ini
```

## 安装 nginx

```
1 sudo apt-get install nginx -y
```

## 添加 nginx 配置文件

创建文件 /etc/nginx/sites-enabled/HomeServiceMall.conf，并修改权限为其它人可写

```
1 sudo touch /etc/nginx/sites-enabled/HomeServiceMall.conf
2 sudo chmod 666 /etc/nginx/sites-enabled/HomeServiceMall.conf
```

编辑 /etc/nginx/sites-enabled/HomeServiceMall.conf

输入以下内容，并保存

```
1 server {
2     listen      80;
3     server_name 119.45.125.97;
4
5     charset utf-8;
6
7     location / {
8         uwsgi_pass 127.0.0.1:8080;
9         include /etc/nginx/uwsgi_params;
10        client_max_body_size 10m;
11    }
12
13    client_body_timeout 3m;
14    send_timeout 3m;
15    proxy_send_timeout 3m;
16    proxy_read_timeout 3m;
17 }
```

## 重启 nginx 服务

```
1 sudo systemctl restart nginx
```

## 6.测试 nginx + uwsgi + django 是否工作正常

---



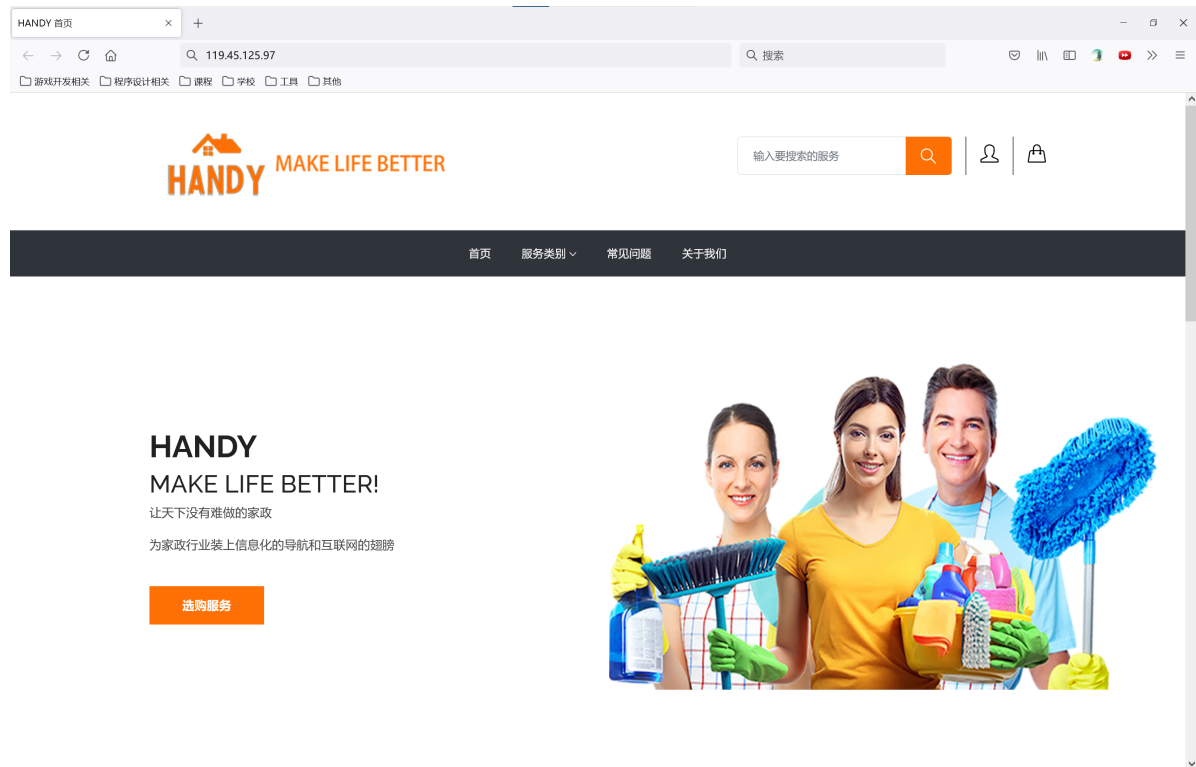
# 测试页面是否能够正常访问

使用浏览器通过外网访问部署到服务器的网站主页

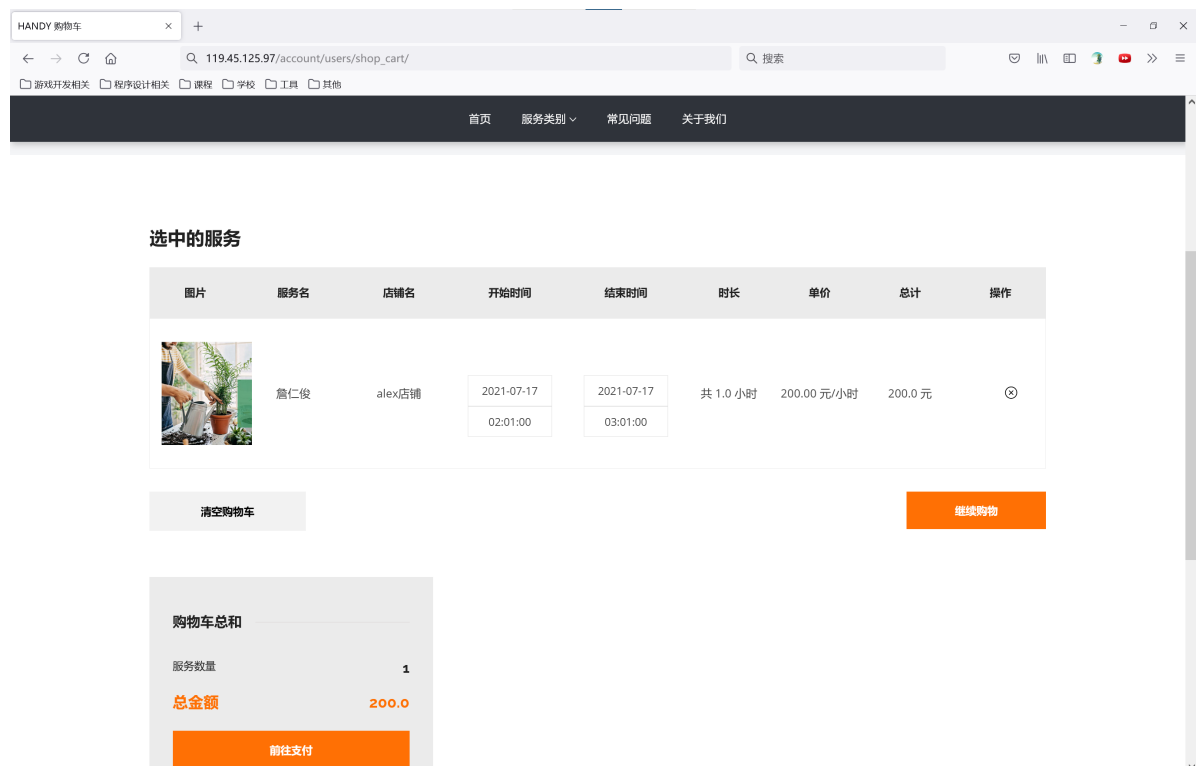
1 http://119.45.125.97/

## 测试结果

通过119.45.125.97成功访问到了网站主页



整个网站运行正常



应用部署成功