



ON HEROKU

Last updated on 25th November 2021 by Apoorv Garg | Category: [Django Tutorial](#)

Host your Django app on a server for free using the service of Heroku and Github

Learn how you can deploy your Django Application on the internet for free for no cost. A Simple and Easy service for hosting any 5 web app is provided by Heroku without any money. We will be using a Django App to showcase this service, [hosting](#) our application on Cloud.



Django and Heroku

[Heroku](#) is a free cloud hosting service. Which can be used to deploy any 5 Web Apps app under the size of 500 MB to be deployed on the cloud. It also provides the service of a free Database (Relational or Otherwise) along with the server.

FREEHOSTING

Why Deploy A Django App during development?

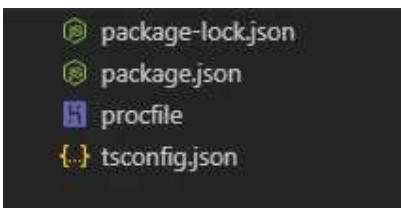
Migrating a Django App from local server to a cloud opens us to an amazing learning and testing opportunity. It also allows to test our app on the internet and gauge its performance. It also exposes the novice developer to using and learning the deployment configurations free of cost and in a stress free environment. Not to mention it is a perfect way for young developers to showcase their work to the world and to the non-technological employer they are often hired by.



Host your Django app on a server for free using the service of Heroku and Github following these simple steps.

Step 1. Prepare the Django App

The Django App has to be prepared with a few configuration files before we can deploy it on the web.



procfile

The first thing we have to do is make a procfile in our project which will be detected by Heroku on deployment. This will redirect the control to wsgi.py in the main app which by default handles the deployment work in Python/Django framework.

The Procfile will contain the following code.

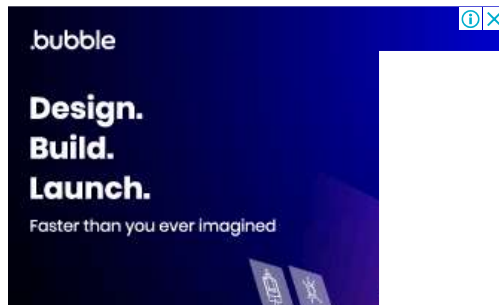
```
web: gunicorn CCMS.wsgi --log-file -
```

Where your project name will replace the CCMS

Install gunicorn dependency

```
pip install gunicorn
```

Prepare a requirements.txt



This file in your project is used to tell the server what dependencies and their version that are required for the smooth deployment and working of your Web App.

```
pip freeze > requirements.txt
```

Update settings.py file

Set the following changes.

```
DEBUG = False

ALLOWED_HOSTS = ['127.0.0.1', 'sampledomain.com']
```

Original Middle ware Code

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

Updated Middle ware Code

Install whitenoise dependency

```
pip install whitenoise
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'whitenoise.middleware.WhiteNoiseMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

Original urls.py

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include("Home.urls")),  
]
```

Updated urls.py

```
url(r'^static/(?P<path>.*)*$', serve,{'document_root': settings.STATIC_ROOT})),  
]
```



GitHub and Heroku

Step 2. Commit the project to GitHub

Once all these changes are in order commit them to any GitHub repository of your choice.



Step 3. Create and Setup a Heroku App

Heroku Dashboard

1. By clicking on "New" you can open the Create New App.
2. Choose an available domain name.
3. Go to Settings and add a build pack of Heroku/python.
4. Now switch to deploy panel.
5. Select GitHub in the method of deployment(You may need to login to your GitHub account).
6. Select appropriate branch.
7. Click "Deploy" under "Manual Deploy".
8. Check your new website.

Heroku Manual Deploy

Disclaimer:- You can check the logs to check what went wrong with deployment. In the log files.

Tags: [django](#), [Heroku](#)

Share: [!\[\]\(830769b31eeeaca920791081939ff8ba_img.jpg\)](#) [!\[\]\(198f559926258ddfad814817bda0ffbc_img.jpg\)](#) [!\[\]\(ea8b0a47f44008b0ab9e2764654f5e66_img.jpg\)](#) [!\[\]\(68656bc6c38d84412cc68fe8fbc2ae14_img.jpg\)](#) [!\[\]\(29298814951e8ab572d6eff24e79a2a8_img.jpg\)](#) [!\[\]\(c94f315b817772f117076a901182c9cc_img.jpg\)](#) [!\[\]\(d501cd71bff889913ccb01e9bfd91932_img.jpg\)](#)



report this ad

© 2022 STUDYGYAAN