

Deploy Django Project On Heroku – Step-by-Step Guide

(Scroll at the bottom to check common errors during deployment)



1. Recommended to update Django to **v2.2.10** if you are using any old version, if you are using v2.1.7 that will also work.

```
pip install --upgrade django==2.2.10
```

2. Register on Github and Create a Repository,
<https://github.com/>

Download git if not installed -
<https://git-scm.com/downloads>

Check git version to confirm the download,
`git --version`

3. Register on Heroku,
<https://www.heroku.com/>

Download Heroku CLI -
<https://devcenter.heroku.com/articles/getting-started-with-python#set-up>

Test with 'heroku' command in the new terminal then login using 'heroku login'.
`heroku`
`heroku login`

4. In base folder create requirements.txt file to get all required dependencies,
pip freeze
pip freeze > requirements.txt
5. In base folder create .gitignore file to avoid unwanted files on server,
<https://github.com/github/gitignore/blob/master/Python.gitignore>
6. Hide *Secret Key*, change *Debug Mode*, *Allowed Hosts* and other important Access Keys.
<https://django-environ.readthedocs.io/en/latest/>
<https://mitchel.me/2016/settings-management-in-django/>

6.1. pip install django-environ

- 6.2. Create .env file in BASE DIR and add the following,
- DJANGO_SECRET_KEY =
 - DJANGO_DEBUG = True
 - DJANGO_ALLOWED_HOSTS = []

6.3. Edit settings.py -

1. import environ
2. Add these code after defining BASE_DIR,
env = environ.Env(SECRET_KEY=str,)
environ.Env.read_env(os.path.join(BASE_DIR, '.env'))
3. Replace *Secret Key*, *Debug Mode*, *Allowed Hosts* with env codes.
env('DJANGO_SECRET_KEY')
env('DJANGO_DEBUG')
env('DJANGO_ALLOWED_HOSTS')

6.4. Also add .env in your .gitignore file.

7. Push code To Github.

```
git init
git add . or git add -A
git status
git commit -m "Uploading Taskmate Project"
```

If you get error as a first time user,

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

Then again do,

```
git commit -m "Uploading Taskmate Project"
git remote add origin https://github.com/your_repo_link
git push -u origin master
```

8. Install django-heroku (Configure database and static assets with gunicorn)
pip install django-heroku

Update settings.py (import it at the top and add settings at the end).

```
import django_heroku
django_heroku.settings(locals())
```

9. Open Heroku Dashboard on side and on terminal create heroku app,
heroku create project_name
heroku open

10. Update requirements.txt and push changes on GitHub & Heroku,
pip freeze > requirements.txt

```
git status
git add -A or git add .
git commit -m "Taskmate Updated"
git push origin master
git push heroku master
```

If error occurs for static path, add STATIC_ROOT in settings.py –
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')

11. Handling WSGI (Web Server Gateway Interface),
 - Install gunicorn using 'pip install gunicorn'
 - Create Procfile inside project folder (where manage.py exist)
(Make sure to use P as capital while naming Procfile)
 - Add gunicorn command and update your project_name.
web: gunicorn myproject.wsgi

Addition reading -

<https://devcenter.heroku.com/articles/django-app-configuration>
<https://vsupalov.com/what-is-gunicorn/>

12. Create an empty 'staticfiles' folder and add an empty '.keep' file inside it.
Then push all changes on GitHub and Heroku

13. If you are still facing error,
- Disable collectstatic
- Push changes on heroku

14. We now need to configure Heroku for *Secret Key, Debug Mode, Allowed Hosts*.
<https://devcenter.heroku.com/articles/config-vars>

```
heroku config:set DJANGO_SECRET_KEY="r4trhegrgFDWEFtgf3$RTF34r"  
heroku config:set DJANGO_DEBUG=False  
heroku config:set DJANGO_ALLOWED_HOSTS='name.herokuapp.com'
```

Run 'heroku open' to see our app working on the server.
Update git and heroku repository after changes.

15. Now migrate our database and add user –
heroku run bash (we are inside Linux based system)
python manage.py migrate
python manage.py createsuperuser
exit

16. To edit and push updated content anytime in the future, use –
git status
git add -A or git add .
git commit -m "Message"
git push origin master
push heroku master

17. Yuppie it's Done, test you app now!

Django Official Deployment Checklist

<https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/>

Remember,

DEBUG = True (Development) -> On your system
DEBUG = False (Production) -> On server

Common Errors:

1. No module named os

Solution: Add `import os` at the top in settings.py file

2. No web process running

Solution: Check your Procfile. (Make sure file name is correct Procfile and should not be procfile)

3. App crashed

Solution: Update Procfile with this `web: gunicorn django_project.wsgi --log-file -`
(Change django_project with your project name, also take care of spaces with this code)