Topic: Django Static Files

Speaker: Personal | Notebook: Understanding Django



You need to understand the following as these are important especially when you have static files like JS, CSS, IMG that are needed during development and production/deployment.

As per this article from Medium:

`DEBUG`

This is an important setting for the entire app. You want this to be `False` on production. If `DEBUG = True` then `./manage.py runserver` will handle serving static files on its own. However, when `DEBUG = False` `runserver` will not do this for you, and that's where some complexity arises.

`STATIC_URL`

The static url is the url path where a client or browser can access static files. So with `https://www.example.com` as your url, if `STATIC_URL = 'mystaticpath'` and you had an image named `test.jpg`, you would access the image at `https://www.example.com/mystaticpath/test.jpg`.

`STATIC_ROOT`

This generates the directory where files static files are placed when you run `./manage.py collectstatic`. For example, if your root folder was `myawesomesite` and `STATIC_ROOT = os.path.join(BASE_DIR, 'placefileshere')`, after running `collectstatic` a folder would have been generated in `myawesomesite/placefileshere/` with `staticfiles.json` and also `test.jpg` from above located in it.

I do not believe `STATIC_ROOT` has any impact when `DEBUG = True`, as you don't need to run `collectstatic` in development.

`STATICFILES_DIRS`

This determines the locations where static files are being pulled from. For example:

```
STATICFILES_DIRS = [os.path.join(BASE_DIR, 'polls/files/')]
```

Any files in `polls/files/` would be in and accessible as static files. The directory structure within it is maintained as well, so for example with

```
--polls
----files
------img
--------test.jpg
```

So when you need to deploy especially in Heroku, you have to be clear with the purpose and difference of staticfiles.

This is VERY ENLIGHTENING TO KNOW:

Chances are you're using the `django-heroku` library and setting `django_heroku.settings(locals())` as suggested in Configuring Django Apps for Heroku.

What I've discovered is that `STATIC_URL` and `STATIC_ROOT` are actually overwritten by heroku to `STATIC_URL = '/static/'` and `STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')`. Even if you have a different `STATIC_ROOT` in your settings if you were to run `heroku run python manage.py collectstatic`, they will use `staticfiles` as the `STATIC_ROOT`.

To view the complete article, go here.

Copyright © Personal Digital Notebooks | By Rosilie | Date Printed: Nov. 5, 2026, 5:23 p.m.