**STATIC\_ROOT**

STATIC\_ROOT is useless during development; it's only required for deployment.

The absolute path to the directory where collectstatic will collect static files for deployment.

STATIC\_ROOT="/var/www/example.com/static/"

now the command python manage.py collectstatic will copy all the static files to the directory /var/www/example.com/static/. Now you only need to serve this directory on apache or Nginx etc…

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I believe I understand what the STATIC\_ROOT is: it's essentially the location on the server where the static files collectstatic command will place the static files collected from your django project. The collectstatic command searches in the locations that you specify in the STATIC\_FINDERS setting.

# **STATIC\_URL ( is URL prefix for static files)**

STATIC\_URL is the URL of static files (within STATIC\_ROOT directory that are served by Apache or Nginx or …). **Example:**/static/ **or**http://static.example.com/

If you set STATIC\_URL = 'http://static.example.com/', then you must serve the STATIC\_ROOT folder (ie "/var/www/example.com/static/") by Apache or Nginx at URL 'http://static.example.com/' (so that you can refer the static file '/var/www/example.com/static/jquery.js' with 'http://static.example.com/jquery.js')

Now in your django-templates, you can refer it by:

{% load static %}

<script src="{% static "jquery.js" %}"></script>

which will render:

<script src="http://static.example.com/jquery.js"></script>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Development

STATIC\_ROOT is useless during development; it's only required for deployment.

While in development, STATIC\_ROOT does nothing. You even don't need to set it. Django looks for static files inside each app's directory (myProject/appName/static) and serves them automatically.

This is the magic done by manage.py runserver when DEBUG=True

## Deployment

When your project goes live, things differ and static files will be served by Nginx, for Nginx is incredibly efficient and will reduce the workload of Django.

This is where STATIC\_ROOT becomes handy, as Nginx doesn't know anything about your django project and doesn't know where to find static files.

So you set STATIC\_ROOT = '/some/folder/' and thenceforth tell Nginx to look for all static files here in /some/folder/. Then you run manage.py collectstatic and Django will copy static files from all the apps you have into /some/folder/

## Extra directories for static files

STATICFILES\_DIRS is the list of folders where Django will search for additional static files aside from the static folder of each app installed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The STATICFILES\_DIRS can contain other directories (not necessarily app directories) with static files and these static files will be collected into your STATIC\_ROOT when you run collectstatic.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

STATICFILES\_DIRS is used to include **additional** directories for collectstatic to look for. For example, by default, Django doesn't recognize /myProject/static/. So you can include it yourself.

STATIC\_URL = '/static/'

if not DEBUG:

STATIC\_ROOT = '/home/django/www-data/site.com/static/'

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR, 'static'),

]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Like you mentioned, it is pretty clear from the documentation:

**STATIC\_ROOT:**

The absolute path to the directory where collectstatic will collect static files for deployment.

**STATIC\_URL**

default: None

URL to use when referring to static files located in STATIC\_ROOT.

Example: "/static/" or "http://static.example.com/"

While, the STATIC\_ROOT is just the path to the directory where static files have been collected, STATIC\_URL is the URL which will serve those static files.

And, as you can see in the example, you can define STATIC\_URL as a subdomain "http://static.example.com/" and when you use it in the template:

<link rel="stylesheet" href="{{ STATIC\_URL }}css/base.css" type="text/css" />

It will be treated as:

<link rel="stylesheet" href="http://static.example.com/css/base.css" type="text/css" />

But, if the STATIC\_URL was just /static/ then the above link would point to:

<link rel="stylesheet" href="/static/css/base.css" type="text/css" />

And, since this href starts with / it will append your domain to access the static files:http://yourdomain/static/css/base/css