

Django StaticFiles

Websites generally need to serve additional files such as images, JavaScript, or CSS. In Django, we refer to these files as “static files”. Django provides *django.contrib.staticfiles* to help you manage them.

1. Make sure that `django.contrib.staticfiles` is included in your *INSTALLED_APPS*.
2. In your settings file, define *STATIC_URL*, for example:

```
STATIC_URL = '/static/'
```

```
{% load static %}  

```

4. Store your static files in a folder called `static` in your app. For example `my_app/static/my_app/example.jpg`.

Serving static files in production

The basic outline of putting static files into production consists of two steps: run the *collectstatic* command when static files change, then arrange for the collected static files directory (*STATIC_ROOT*) to be moved to the static file server and served. Depending on *STATICFILES_STORAGE*, files may need to be moved to a new location manually or the *post_process* method of the Storage class might take care of that.

Of course, as with all deployment tasks, the devil’s in the details. Every production setup will be a bit different, so you’ll need to adapt the basic outline to fit your needs. Below are a few common patterns that might help.