Gestion des fichiers statiques et médias dans Django : Explications complètes

---

1. Qu'est-ce que les fichiers statiques et médias ?

- **Fichiers statiques** : Ce sont les fichiers *CSS*, *JavaScript*, *images* d'interface, icônes, polices… qui ne changent pas souvent et font partie de ton frontend.
- **Fichiers médias** : Ce sont les fichiers *uploadés par les utilisateurs* (photos de profil, documents, etc.).

---

2. Pourquoi gérer ces fichiers avec soin ?

- Pour que Django et ton serveur web (nginx, apache, etc.) sachent **où trouver** ces fichiers.
- Pour que ces fichiers soient accessibles via des URLs simples dans ton app.
- Pour séparer clairement la logique applicative (code Python/Django) des ressources statiques.

---

3. Paramètres essentiels dans `settings.py`

Voici ce que tu dois définir **obligatoirement** pour que Django sache où se trouvent et comment servir ces fichiers :

```
# URL utilisée dans le navigateur pour accéder aux fichiers statiques
STATIC_URL = '/static/'

# URL utilisée pour accéder aux fichiers médias (uploads)
MEDIA_URL = '/media/'

# Chemin physique (sur le disque) où seront stockés les fichiers médias
MEDIA_ROOT = '/vol/web/media'

# Chemin physique (sur le disque) où seront collectés tous les fichiers
statiques (via collectstatic)
STATIC_ROOT = '/vol/web/static'
```

Explication détaillée :

- `STATIC_URL` et `MEDIA_URL` commencent toujours par un slash `/` — ça signifie "depuis la racine du site".
- Les chemins physiques (`STATIC_ROOT`, `MEDIA_ROOT`) pointent vers des dossiers sur le système de fichiers, souvent **montés dans Docker** ou accessibles par le serveur.
- `STATIC_ROOT` est là pour recevoir tous les fichiers statiques collectés en un seul endroit (après commande `collectstatic`).
- `MEDIA_ROOT` est là pour stocker les fichiers uploadés à la volée.

4. Configurer les URL dans `urls.py`

Pour que Django serve correctement ces fichiers en mode développement (`DEBUG=True`), tu dois ajouter cette configuration dans ton `urls.py` principal :

```
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    # ... tes autres routes ...
]

if settings.DEBUG:
    urlpatterns += static(settings.STATIC_URL,
document_root=settings.STATIC_ROOT)
    urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
```

Pourquoi ? Parce que Django ne sert pas ces fichiers automatiquement en production, et même en dev il faut préciser où les trouver.

---

5. Le rôle de la commande `collectstatic`

- Lorsque tu exécutes :

  ```
  python manage.py collectstatic
  ```

- Django **copie tous les fichiers statiques** des différentes apps vers le dossier `STATIC_ROOT` (`/vol/web/static` dans ton cas).
- Ça simplifie la gestion des fichiers statiques en production : tu n'as plus qu'un seul dossier à exposer via nginx ou autre.

---

6. Gestion des fichiers dans Docker

Dans ton Dockerfile, tu as créé les dossiers :

```
mkdir -p /vol/web/media
mkdir -p /vol/web/static
chown -R django-user:django-user /vol
chmod -R 755 /vol
```

- Ces dossiers seront les volumes montés pour stocker tes fichiers statiques et médias.

- Assure-toi que dans ton `docker-compose.yml` tu montes bien ces dossiers pour garder la persistance des fichiers :

```
volumes:
  - ./vol/web/media:/vol/web/media
  - ./vol/web/static:/vol/web/static
```

---

7. **En production**

- Tu ne laisses **pas Django servir les fichiers statiques ou médias** !
- Tu configures un serveur web (nginx, caddy…) pour **servir `/static/` et `/media/` directement**, beaucoup plus performant.
- Django ne gère alors que la partie logique / API / backend.

---

8. **Récapitulatif**

Concept	Valeur exemple	Description
`STATIC_URL`	`/static/`	URL publique des fichiers statiques
`MEDIA_URL`	`/media/`	URL publique des fichiers médias (uploads)
`STATIC_ROOT`	`/vol/web/static`	Répertoire physique où `collectstatic` dépose tout
`MEDIA_ROOT`	`/vol/web/media`	Répertoire physique où les fichiers uploadés sont stockés
`urls.py` config	voir ci-dessus	Sert les fichiers en dev uniquement
Docker volumes	montés sur `/vol/web/static`, `/vol/web/media`	Pour persister les fichiers hors du container

---

9. **Pour tes prochains projets**

- Toujours définir clairement ces 4 variables (`STATIC_URL`, `MEDIA_URL`, `STATIC_ROOT`, `MEDIA_ROOT`)
- Configurer les URLs pour servir statics et media en dev
- Monter tes volumes Docker pour ces dossiers
- Faire un `collectstatic` avant de déployer en prod
- Configurer ton serveur web pour servir les fichiers statiques et médias en prod

---

DIFFERENCE ENTRE STATIC_URL ET STATIC_ROOT

---

1. **`STATIC_URL`**

- C'est l'**URL publique** à laquelle les fichiers statiques sont accessibles dans le navigateur.
- Exemple :

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

- Si dans ton template tu mets

```
<img src="{{ STATIC_URL }}logo.png">
```

Ça correspondra à `/static/logo.png` dans l'URL de ton site.

- **C'est ce que Django (ou le serveur web) utilise pour servir les fichiers statiques au client.**

---

2. `STATIC_ROOT`

- C'est **le chemin absolu sur le système de fichiers** où Django va **regrouper (collecter) tous les fichiers statiques** provenant de tes apps et de tes dossiers statiques.
- Exemple :

```
STATIC_ROOT = '/vol/web/static'
```

- Quand tu lances la commande

```
python manage.py collectstatic
```

Django copie tous les fichiers statiques (CSS, JS, images, etc.) dans ce dossier.

- En production, c'est ce dossier que ton serveur web (nginx, apache...) va servir pour répondre aux requêtes sur l'URL `STATIC_URL`.

Résumé simple

Variable	Rôle	Exemple	Usage
`STATIC_URL`	URL publique des fichiers statics	`/static/`	URL accessible par le navigateur
`STATIC_ROOT`	Dossier sur le disque où sont collectés les fichiers statics	`/vol/web/static`	Dossier réel où sont stockés les fichiers statics après `collectstatic`

Analogies

- `STATIC_URL` = **adresse web** où tu vas chercher les fichiers statiques.
- `STATIC_ROOT` = **adresse physique sur le serveur** où les fichiers statiques sont stockés.

Voici ce qui se passe quand tu charges une image (ou un fichier média) dans Django :

---

1. Où est-elle **physiquement stockée** ?

- Les fichiers **uploadés par les utilisateurs** (images, fichiers…) sont stockés dans le dossier défini par la variable `MEDIA_ROOT`. Exemple dans ton cas :

```
MEDIA_ROOT = '/vol/web/media'
```

- Donc quand tu fais un upload via un formulaire Django, le fichier est enregistré physiquement sous `/vol/web/media/`.

---

2. Comment Django les sert dans l'URL ?

- Django ne sert **pas directement** les fichiers médias en production (tu dois configurer un serveur comme Nginx pour ça).
- Mais dans les paramètres, tu définis `MEDIA_URL`, qui est l'URL de base pour accéder à ces fichiers. Exemple dans ton cas :

```
MEDIA_URL = '/static/media/'
```

- Donc, si un utilisateur charge une image qui s'appelle `avatar.png`, elle sera accessible à l'URL :

```
https://tonsite.com/static/media/avatar.png
```

- Sous le capot, Django sait que `/static/media/` correspond au dossier `/vol/web/media` sur le disque.

---

3. Rappel pour les fichiers statiques (CSS, JS, images de design)

- Ceux-ci sont gérés à part, avec `STATIC_ROOT` pour le dossier physique, et `STATIC_URL` pour l'URL d'accès.
- Par exemple :

```
```

```
STATIC_ROOT = '/vol/web/static'
STATIC_URL = '/static/static/'
```

- Les fichiers statiques sont **préparés et collectés** avec la commande `collectstatic` avant la mise en production.

En résumé :

Type de fichier	Stockage physique (dossier)	URL d'accès
Fichiers uploadés (media)	`/vol/web/media`	`/static/media/`
Fichiers statiques	`/vol/web/static`	`/static/static/`

- **MEDIA_ROOT** et **STATIC_ROOT** = ce sont les **emplacements physiques** sur le serveur où les fichiers sont stockés.
- **MEDIA_URL** et **STATIC_URL** = ce sont les **URL publiques** utilisées dans le navigateur pour accéder à ces fichiers.

C'est donc une séparation claire entre **où le fichier est physiquement** et **comment l'utilisateur y accède via une URL**.

Tu peux imaginer ça comme une maison (le dossier physique) avec une adresse (l'URL). La maison c'est où est rangé l'objet, et l'adresse c'est comment tu le retrouves sur Internet.
