

Fiche de procédure et d'installation de projet

Dans ce document je vais détailler point par point la procédure à suivre de A à Z afin de pouvoir recréer et déployer un site web dynamique à l'aide des outils vu en cours.

1 Création du site avec Django

Tout d'abord il faut procéder à la création globale du site à l'aide du Framework Django.

Pour cela on utilisera Pycharm qui est un IDE python et qui est très simple d'utilisation.

On initialise le projet et on crée nos CRUD en pensant à bien créer et modifier les fichiers importants : models.py, les views.py, les urls.py, etc. comme vu en cours .

On crée aussi l'ensemble des templates de chaque CRUD avec leurs formulaires, page d'affichage, le squelette HTML, le css....

On teste ensuite le site à l'aide du serveur de test de Django, lorsque l'ensemble du site est fini, on peut passer à son déploiement sur un site web extérieur.

2 Déploiement de l'application sur le serveur web Apache2

Afin de déployer notre application et de la sortir du serveur de test intégré à Django, il faut mettre en place un serveur web extérieur qui sera géré sur une machine virtuelle, dans notre cas, vu que nous utilisons le serveur web Apache2, il est plus simple de prendre une machine virtuelle Linux, dans notre cas on utilisera Debian 12 (à noter que la procédure de configuration du serveur web entre les différentes distributions Linux varie légèrement, notamment au niveau des noms de fichiers et de l'installation des paquets).

Donc dans un premier temps il va falloir importer le projet Django sur notre VM, on installe donc les différents paquets nécessaires :

```
apt install python3
```

```
apt install django
```

```
apt install pillow
```

→ ce paquet est requis car nous avons un formulaire qui contient un champ de dépôt d'image

On créer ensuite un nouveau dossier a la racine qui contiendra notre application, notre environnement virtuel « venv » et la partie site :

```
Cd /                                permet de se déplacer à la racine
mkdir SAE23                        création du dossier, le nom de dossier peut bien évidemment être
                                   différent de l'exemple
```

On refait l'arborescence suivante :

```
/SAE23/
|
|--- site
|   |-- logs
|   |-- public
|       |-- media
|       |-- static
|
|---venv
```

Ensuite on importe notre projet Django déjà existant, ici on va importer directement via notre github :

```
apt install git
cd /SAE23/projet
git clone 'lien du dépôt github'
```

On créer le venv :

```
cd /SAE23/venv
pip3 install virtualenv
virtualenv venv -p python3
source venv/bin/activate
```

On passe ensuite l'installation d'Apache2, on va donc installer les différents paquets qui seront nécessaires pour la configuration entière du serveur web :

```
apt install apache2
apt install libapache2-mod-wsgi-py3
```

On modifie ensuite le dossier de configuration 000-default.conf :

```
cd /etc/apache2/site-available
nano 000-default.conf
```

On le remplit comme suit :

```
<VirtualHost *:80>
    ErrorLog /SAE23/site/logs/error.log
    CustomLog /SAE23/site/logs/access.log combine

    <Directory /SAE23/sae>
        <Files wsgi.py>
            Require all granted
        </Files>
    </Directory>

    WSGIDaemonProcess tuto python-home=/SAE23/venv
    python-path=/SAE23/sae/

    WSGIProcessGroup tuto
    WSGIScriptAlias / /SAE23/sae/wsgi.py
</VirtualHost>
```

On redémarre le service APache2 :

```
service apache2 restart
```

On configure maintenant le dossier static pour le css du site :

```
nano /SAE23/sae/settings.py
```

on change la ligne :

```
DEBUG = True          ← on met False a la place pour des raisons de sécurité
```

On modifie le lien de l'URL :

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

```
STATIC_ROOT = '/SAE23/site/public/static'
```

```
MEDIA_ROOT = '/SAE23/site/public/media'
```

On enregistre les modification puis dans le venv on fait la commande qui permet le transfert des fichiers static :

```
python manage.py collectstatic
```

Et pour finir on ajoute les chemin des fichiers 'static' dans le fichier de configuration :

```
nano /etc/apache2/site-available/000-default.conf
```

On modifie le fichier :

```
ErrorLog /SAE23/site/logs/error.log  
CustomLog /SAE23/site/logs/access.log combine  
  
alias /static /SAE23/site/public/static  
<Directory /SAE23/site/public/static>  
    Require all granted  
</Directory>  
  
<Directory /SAE23/sae>
```

Le serveur web Apache2 est désormais opérationnel

3 Configuration de la base de donnée MySQL

Pour ce qui est de la base de donnée, on aura besoin d'une nouvelle VM Windows cette fois-ci sur laquelle on installera MySQL, ensuite il faudra faire communiquer cette nouvelle VM avec la VM Linux déjà existante.

Lorsque MySQL est installé sur la VM Windows, on va créer un nouvel utilisateur avec identifiant et mot de passe qui va gérer la base de donnée.

Sur la VM Linux il faut installer de nouveaux paquets requis pour l'utilisation de cette base de donnée.

```
apt install python3-dev
```

```
apt install pkg-config libmariadb-dev-compatlibmariadb-dev
```

```
pip install mysqlclient
```

Ensuite on crée la nouvelle base de donnée :

```
createdatabase saedb
```

On modifie ensuite le fichier settings.py de Django :

```
nano /SAE23/sae/settings.py
```

On la modifie :

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'tuto',  
        'USER': 'DjangoUser',  
        'PASSWORD': 'toto',  
        'HOST': '192.168.200.128',  
        'PORT': '3306'  
    }  
}
```

On migre la base de donnée :

```
python3 manage.py migrate
```

Et on vérifie sur Windows le bon fonctionnement :

```
use saedb
```

```
show tables ;
```

Si les tables s'affichent bien, la base de donnée MySQL est bien opérationnelle !

4 Conclusion

Voilà l'ensemble du processus pour créer et déployer un site web dynamique, il existe de nombreux autres serveurs web et base de données, voici une solution, vous êtes maintenant prêt à refaire toutes ces manip de vous-même.