

# SAE 23 - Mettre en place une solution informatique pour l'entreprise

Fiche de procédure du déploiement  
de l'application sur serveur Apache

Martin Baumgaertner  
IUT de Colmar  
17 juin 2022

---

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Configuration de la machine . . . . .	2
1.2	Configuration des paramètres du projet . . . . .	2
<b>2</b>	<b>Serveur Apache</b>	<b>3</b>
2.1	Fichier de configuration du serveur Apache . . . . .	3
2.2	Test de configuration . . . . .	4
<b>3</b>	<b>Configuration de la base de données</b>	<b>5</b>
3.1	Création de la base de données . . . . .	5
<b>4</b>	<b>Déploiement de mon application</b>	<b>7</b>

## Table des codes

1	Installation des paquets . . . . .	2
2	Configuration de l'environnement virtuel . . . . .	2
3	Rajout de l'adresse et du fichier de destination static . . . . .	2
4	Configuration du fichier webgrade.conf . . . . .	3
5	Activation du fichier webgrade.conf . . . . .	4
6	Vérification de la configuration . . . . .	4
7	Création de la base de données . . . . .	5
8	Import de la base de données . . . . .	5
9	Création de l'utilisateur avec tous les droits . . . . .	5
10	Configuration de la base de données . . . . .	6

---

# 1 Introduction

## 1.1 Configuration de la machine

Pour commencer, j'ai choisi déployer mon application sur une machine sous Ubuntu Server. En effet, cette version est optimisée pour le type de déploiement que je vais effectuer. J'ai donc installé les paquets suivants (nécessaires pour le déploiement) :

```
apt install python3-pip apache2 libapache2-mod-wsgi-py3
```

Code 1 – Installation des paquets

Puis, j'ai configuré un environnement virtuel, dans lequel j'ai installé tous les prérequis de mon projet (Django, Pillow, Django-crispy-forms, etc.) :

```
sudo apt install python3-venv
python3 -m venv venv
source venv/bin/activate
pip3 install -r requirements.txt
```

Code 2 – Configuration de l'environnement virtuel

## 1.2 Configuration des paramètres du projet

J'ai dû ajuster le fichier "settings.py" pour qu'il puisse être reconnu par Apache. Premièrement, j'ai rajouté ces lignes de code :

```
ALLOWED_HOSTS = ['webgrade.localhost']
STATIC_ROOT = os.path.join(BASE_DIR, "static/")
```

Code 3 – Rajout de l'adresse et du fichier de destination static

---

## 2 Serveur Apache

### 2.1 Fichier de configuration du serveur Apache

Je suis aller créer un fichier du nom de mon projet donc "webgrade.conf" Puis, je l'ai configuré en y insérant les lignes suivantes :

```
sudo nano /etc/apache2/sites-available/webgrade.conf

<VirtualHost *:80>
    ServerAdmin admin@webgrade.localhost
    ServerName webgrade.localhost
    ServerAlias www.webgrade.localhost
    DocumentRoot /home/martin/SAE-23
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    Alias /static /home/martin/SAE-23/static
    <Directory /home/martin/SAE-23/static>
        Require all granted
    </Directory>

    Alias /static /home/martin/SAE-23/media
    <Directory /home/martin/SAE-23/media>
        Require all granted
    </Directory>

    <Directory /home/martin/SAE-23/webgrade>
        <Files wsgi.py>
            Require all granted
        </Files>
    </Directory>

    WSGIDaemonProcess SAE-23 python-path=/home/martin/SAE-23
    python-home=/home/martin/SAE-23/env
    WSGIProcessGroup SAE-23
    WSGIScriptAlias / /home/martin/SAE-23/webgrade/wsgi.py
</VirtualHost>
```

Code 4 – Configuration du fichier webgrade.conf

---

Après la configuration du fichier, nous devons l'activer à l'aide de la 1ère commande puis, un message nous indique que nous devons redémarrer les services Apache pour que la configuration prenne effet. Ce que j'ai donc fait à l'aide de la deuxième commande :

```
sudo a2ensite webgrade.conf

service apache2 reload
```

Code 5 – Activation du fichier webgrade.conf

## 2.2 Test de configuration

Pour vérifier que nous ne nous sommes pas trompés dans la configuration, nous pouvons taper cette commande à laquelle nous sera répondu comme message : "syntax OK" ce qui signifie donc qu'il n'y a pas d'erreur dans le fichier.

```
sudo apache2ctl configtest
```

Code 6 – Vérification de la configuration

---

## 3 Configuration de la base de données

### 3.1 Création de la base de données

Pour mettre en place la base de données j'ai installé MariaDB-server et client ainsi que MySQL. J'ai exporté la base de données que j'avais créé sur ma machine hôte et je l'ai transféré sur ma machine Ubuntu. Ensuite, j'ai donc ouvert MariaDB et créé une base de données que j'ai nommé « grade » à l'aide de la commande suivante :

```
mysql -u root -p
CREATE DATABASE grade;
```

Code 7 – Création de la base de données

Puis, j'ai importé ma base de données déjà existante dans la nouvelle base de données que je viens de créer dans MariaDB avec cette commande, depuis le terminal linux :

```
mysql -u root -p grade < grade.sql
```

Code 8 – Import de la base de données

Pour pouvoir mettre en place la base de données avec le projet j'ai dû me connecter à la base de données que j'ai créée avec MariaDB, pour ce faire j'ai créé un utilisateur depuis MariaDB, et je lui ai donné tous les droits, à l'aide des commandes suivantes :

```
CREATE USER 'webgrade'@'localhost' IDENTIFIED BY 'webgrade';
GRANT ALL PRIVILEGES ON *.* TO 'webgrade'@'localhost';
```

Code 9 – Création de l'utilisateur avec tous les droits

---

Pour finir, j'ai renseigné les bons paramètres dans "database" dans le fichier settings.py de mon projet :

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'grade',  
        'USER': 'webgrade',  
        'PASSWORD': 'webgrade',  
        'HOST': 'localhost',  
        'PORT': '3306',  
    }  
}
```

Code 10 – Configuration de la base de données

---

## 4 Déploiement de mon application

Après avoir redémarré Apache, j'ai pu déployer mon application. En me connectant depuis un navigateur et en saisissant l'adresse "webgrade.localhost" j'ai pu me connecter à mon application !

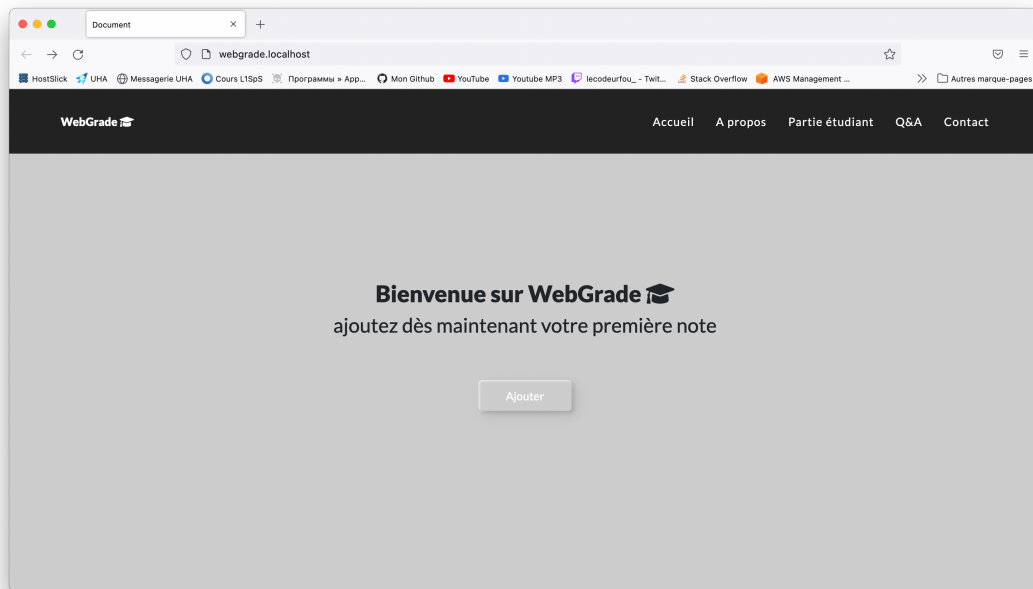


FIGURE 1 – Notre application déployée sur le serveur !