

Manuel d'installation

ASADI : ASsistant ADministratif Intelligent

Groupe:
L3W1

Encadrant:
David Janisek

Auteurs:
Djaffer Abdel Malik, Gourmelen Thomas, Ponnoussamy Valentin, Traore Ali

Version du document:
1.0

Résumé

Dans le cadre de l'UE Projet Informatique à l'université Paris Cité, nous devons par équipes de quatre réaliser un projet tout le long du semestre 6 sous la tutelle d'un encadrant. Ce projet vise à développer un assistant numérique capable de répondre automatiquement aux questions administratives, d'expliquer des procédures et d'offrir un apprentissage interactif via des scénarios immersifs, en s'appuyant sur une base documentaire centralisée.

Sommaire

1	Introduction	3
1.1	Contexte	3
1.2	Objectif du manuel d'installation	3
2	Installation du matériel	4
2.1	Installation de Django	4
2.2	Installation, connexion et création de la base PostgreSQL	5
3	Installation du logiciel	6
3.1	Importation du code source	6
4	Déploiement de l'application ASADI sur serveur VPS	7
4.1	Objectif	7
4.2	Création et configuration du VPS	7
4.3	Déploiement de l'application	7
4.4	Gunicorn	8
4.5	Configuration NGINX	9
4.6	Certificat HTTPS avec Let's Encrypt	9
4.7	Sécurité serveur	9
5	Premier pas	10
5.1	Création d'un compte utilisateur	10
5.2	Création d'un administrateur	10
5.3	Navigation initiale	10
5.4	Vérification	10
6	Conclusion	11
7	Références	11

1 Introduction

1.1 Contexte

Notre assistant virtuel, fondé sur la technologie RAG, a été conçu pour simplifier et rendre plus accessibles les démarches administratives. Face à la complexité des procédures et à la diversité des documents administratifs, il est essentiel de proposer une solution intelligente et interactive, capable d'accompagner efficacement les utilisateurs dans leur recherche d'informations et leur apprentissage.

En exploitant un ensemble de ressources documentaires (fichiers, textes officiels, sites web, etc.), cet assistant permet de centraliser les connaissances et de les rendre accessibles. Il propose également un apprentissage ludique et personnalisé, via des scénarios interactifs et des questions adaptées au profil de l'utilisateur.

1.2 Objectif du manuel d'installation

Ce manuel vise à fournir une procédure claire, structurée et détaillée pour installer correctement l'application. Il s'adresse aux utilisateurs souhaitant déployer le système dans leur environnement, en garantissant le bon déroulement de chaque étape pour assurer le bon fonctionnement de l'application. Ce document présente également les prérequis, les erreurs fréquentes à éviter et les bonnes pratiques pour une configuration optimale.

2 Installation du matériel

Cette section décrit le matériel nécessaire et les opérations requises pour sa mise en service.

2.1 Installation de Django

2.1.a Étape 1 : Installation de Python

Avant d'installer Django, il convient de vérifier si Python est déjà installé :

- ▶ Vérification de Python :
 - Ouvrez un terminal (ou CMD/PowerShell sous Windows) et tapez la commande suivante :
`python --version`

Si une version de type Python 3.x.x s'affiche, Python est installé. Sinon, procédez à l'installation :

- ▶ Installation de Python :
 - Téléchargez la dernière version depuis : <https://www.python.org/downloads/>
 - Cliquez sur Download Python 3.x.x.
 - Lancez l'installation. Sous Windows, cochez impérativement "Add Python to PATH" avant de cliquer sur "Install Now".
 - Après installation, vérifiez à nouveau : `python --version`

Vous devez obtenir une version de type Python 3.x.x.

2.1.b Étape 2 : Création de l'environnement virtuel

- ▶ Créez un dossier pour votre projet Django.
- ▶ Ouvrez un terminal et naviguez jusqu'à ce dossier avec `cd` (ex. : `cd mon_projet_django`).
- ▶ Créez l'environnement virtuel avec : `python -m venv venv`

Cela crée un dossier `venv` contenant un environnement Python isolé.

2.1.c Étape 3 : Activation de l'environnement virtuel

Dans le dossier du projet :

- ▶ Sous Windows : `.\venv\Scripts\activate`
- ▶ Sous macOS/Linux : `source venv/bin/activate`

Le terminal affichera alors `(venv)` au début de la ligne.

2.1.d Étape 4 : Installation de Django

- ▶ Tapez : `pip install django`
- ▶ Vérifiez avec : `django-admin --version`

2.1.e Étape 5 : Création d'un projet Django

- ▶ Créez un projet : `django-admin startproject monprojet`
- ▶ Accédez au dossier du projet : `cd monprojet`

2.1.f Étape 6 : Lancement du serveur local

- ▶ Dans le dossier du projet, lancez : `python manage.py runserver`

Vous obtiendrez une URL locale du type `http://127.0.0.1:8000`

2.2 Installation, connexion et création de la base PostgreSQL

2.2.a Étape 1 : Installation de PostgreSQL

- ▶ Téléchargez l'installateur depuis : <https://www.postgresql.org/download/>
- ▶ Choisissez votre système d'exploitation.
- ▶ Lors de l'installation, définissez un mot de passe pour l'utilisateur postgres.

2.2.b Étape 2 : Démarrage du service PostgreSQL

- ▶ Windows : le service démarre automatiquement. Vérifiez dans Services > postgresql-x64-xx.
- ▶ Linux/macOS : lancez avec `sudo systemctl start postgresql`

2.2.c Étape 3 : Connexion via pgAdmin 4

- ▶ Ouvrez pgAdmin 4, définissez un mot de passe maître.
- ▶ Connectez-vous au serveur PostgreSQL xx avec le mot de passe défini à l'installation.

2.2.d Étape 4 : Création de la base de données

- ▶ Dans pgAdmin, clic droit sur Databases > Create > Database...
- ▶ Donnez un nom à la base et cliquez sur Save.

2.2.e Étape 5 : Connexion à PostgreSQL dans Django avec .env et python-decouple

- ▶ Activez l'environnement virtuel.
- ▶ Installez les bibliothèques : `pip install python-decouple psycopg2-binary`
- ▶ Créez un fichier .env à la racine du projet.
- ▶ Dans settings.py :

```
from decouple import config
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': config('DB_NAME'),
        'USER': config('DB_USER'),
        'PASSWORD': config('DB_PASSWORD'),
        'HOST': config('DB_HOST'),
        'PORT': config('DB_PORT'),
    }
}
```

- ▶ Ajoutez .env à .gitignore

3 Installation du logiciel

3.1 Importation du code source

- ▶ Récupérez le code source avec : `svn checkout https://forge.ens.math-info.univ-paris5.fr/svn/2024-l3w1/trunk --username=VotreLogin`
- ▶ Saisissez votre mot de passe.

N.B. : l'accès au dépôt est requis.

Une fois le code importé :

1. Activez l'environnement :

- ▶ `.\venv\Scripts\activate` ou `source venv/bin/activate`

2. Lancez les migrations :

- ▶ `python manage.py makemigrations`
- ▶ `python manage.py migrate`

3. Installez les dépendances :

- ▶ `pip install -r requirements.txt`

4. Démarrez le serveur :

- ▶ `python manage.py runserver`

L'application est désormais prête à être utilisée localement.

4 Déploiement de l'application ASADI sur serveur VPS

4.1 Objectif

Ce chapitre détaille l'ensemble des étapes nécessaires au déploiement du projet Django ASADI sur un serveur VPS. Il comprend la configuration du serveur, l'installation des dépendances, la sécurisation de l'environnement, l'intégration du nom de domaine et la mise en place du certificat HTTPS.

4.2 Création et configuration du VPS

- ▶ Achat d'un VPS KVM sur Hostinger
- ▶ Connexion SSH :

```
ssh deploy@168.231.77.46
```

- ▶ Mise à jour du système :

```
sudo apt update && sudo apt upgrade -y
```

- ▶ Installation des paquets nécessaires :

```
sudo apt install python3-pip python3-venv nginx postgresql ufw certbot python3-certbot-nginx unzip -y
```

4.3 Déploiement de l'application

4.3.a Récupération du projet (via GitHub)

```
cd ~
git clone https://github.com/utilisateur/nom-depot.git asadi
cd asadi
```

4.3.b Création de l'environnement virtuel

```
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

4.3.c Création du fichier .env

```
ALLOWED_HOSTS=asadi.fr,www.asadi.fr,127.0.0.1,localhost
DB_NAME=asadi
DB_USER=asadi_user
DB_PASSWORD=motdepasse
DB_HOST=localhost
CHROMA_DB_DIR=/home/deploy/asadi/chroma_db
```

- ▶ Protection :

```
chmod 600 .env
```

4.3.d Configuration PostgreSQL

```
sudo -u postgres psql
CREATE DATABASE asadi;
CREATE USER asadi_user WITH PASSWORD 'motdepasse';
GRANT ALL PRIVILEGES ON DATABASE asadi TO asadi_user;
\q
```

4.3.e Migration et statiques

```
python manage.py migrate
python manage.py collectstatic
```

4.4 Unicorn

4.4.a Lancement test

```
gunicorn ASADI.wsgi:application --bind 127.0.0.1:8000
```

4.4.b Configuration systemd

```
sudo nano /etc/systemd/system/gunicorn.service
```

```
[Unit]
```

```
Description=Gunicorn daemon
```

```
After=network.target
```

```
[Service]
```

```
User=deploy
```

```
Group=www-data
```

```
WorkingDirectory=/home/deploy/asadi
```

```
Environment="PATH=/home/deploy/asadi/venv/bin"
```

```
ExecStart=/home/deploy/asadi/venv/bin/gunicorn ASADI.wsgi:application --bind  
127.0.0.1:8000
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl start gunicorn
```

```
sudo systemctl enable gunicorn
```


4.5 Configuration NGINX

```
sudo nano /etc/nginx/sites-available/asadi

server {
    listen 80;
    server_name asadi.fr www.asadi.fr;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name asadi.fr www.asadi.fr;

    ssl_certificate /etc/letsencrypt/live/asadi.fr/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/asadi.fr/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location /static/ {
        alias /home/deploy/asadi/staticfiles/;
    }

    location / {
        proxy_pass http://127.0.0.1:8000;
        include proxy_params;
    }

    location ~ /\.env {
        deny all;
    }
}

sudo ln -s /etc/nginx/sites-available/asadi /etc/nginx/sites-enabled/
sudo rm /etc/nginx/sites-enabled/default
sudo nginx -t
sudo systemctl reload nginx
```

4.6 Certificat HTTPS avec Let's Encrypt

```
sudo certbot --nginx -d asadi.fr -d www.asadi.fr
```

4.7 Sécurité serveur

- UFW :

```
sudo ufw allow OpenSSH
sudo ufw allow 'Nginx Full'
sudo ufw enable
```

- Droits sur les dossiers :

```
chmod -R o+rX ~/asadi/staticfiles
chmod o+x ~ ~/asadi
```

5 Premier pas

Une fois l'application installée, configurée et déployée sur le serveur, vous pouvez accéder à l'interface via un navigateur en vous rendant à l'adresse : <https://asadi.fr>.

L'interface de l'application ASADI a été pensée pour être intuitive et facile d'accès. Elle se divise en deux profils principaux : les utilisateurs standards et les administrateurs.

5.1 Création d'un compte utilisateur

Au premier accès, l'utilisateur peut :

- ▶ Créer un compte en renseignant une adresse e-mail et un mot de passe,
- ▶ Se connecter à l'aide de ses identifiants s'il dispose déjà d'un compte.

Une fois connecté, l'utilisateur est redirigé vers la page principale de l'application, où il peut :

- ▶ Créer et envoyer des prompts (requêtes),
- ▶ Consulter son historique d'échanges,
- ▶ Participer à des quiz pour renforcer ses connaissances,
- ▶ Lancer des scénarios immersifs pour tester ses connaissances administratifs.

5.2 Création d'un administrateur

Pour configurer un compte administrateur, un super utilisateur doit d'abord être généré via la commande suivante :

```
python manage.py createsuperuser
```

Ce compte aura accès à l'interface d'administration Django via : <https://asadi.fr/admin>

Une fois connecté, il pourra :

- ▶ Accéder à la gestion des utilisateurs,
- ▶ Créer ou modifier des scénarios,
- ▶ Gérer les documents de la base de connaissances,
- ▶ Organiser les fichiers par thématiques (workspaces).

5.3 Navigation initiale

L'utilisateur est accueilli par une interface dotée :

- ▶ d'un menu latéral permettant de naviguer entre les modules (Profil, Quiz, Scénarios, Historique),
- ▶ d'une zone centrale où les prompts peuvent être créés et envoyés à l'assistant virtuel,
- ▶ d'un système de filtres permettant d'affiner la recherche d'informations.

5.4 Vérification

Pour s'assurer que l'installation a été correctement effectuée, vérifiez que :

- ▶ La connexion et l'inscription fonctionnent
- ▶ Les prompts sont bien traités et renvoient une réponse pertinente
- ▶ Les quiz et scénarios sont accessibles et interactifs
- ▶ Les fichiers statiques (CSS/JS) sont bien affichés (interface stylée)
- ▶ L'accès HTTPS est actif (cadenas dans la barre d'URL)
- ▶ Le superutilisateur accède à l'administration via `/admin`.

6 Conclusion

Ce manuel d'installation a été conçu pour accompagner étape par étape tout utilisateur souhaitant déployer localement ou sur un serveur distant l'application web ASADI. En combinant la puissance du framework Django, une base de données PostgreSQL et une architecture serveur sécurisée, ce guide fournit un environnement robuste et performant pour faire fonctionner l'assistant administratif intelligent.

L'installation couvre non seulement les besoins logiciels (Python, Django, bibliothèques, serveur Unicorn, reverse proxy Nginx), mais également les aspects critiques liés à la sécurité (protection des variables d'environnement, HTTPS, pare-feu) et à l'organisation du projet (nom de domaine, configuration VPS, déploiement GitHub).

En suivant rigoureusement ce guide, l'utilisateur est en mesure de :

- ▶ déployer une version pleinement fonctionnelle d'ASADI
- ▶ gérer l'administration de l'application
- ▶ et garantir un accès sécurisé à ses fonctionnalités avancées.

7 Références

- ▶ Pour la rédaction de cette documentation, nous avons utilisé le modèle d'intelligence artificielle ChatGPT. Ce modèle nous a aidés sur plusieurs points :
 - Reformulation de phrases et correction de fautes d'orthographe : Nous avons utilisé ChatGPT pour nous assister dans la rédaction de certaines phrases, afin d'améliorer la clarté du texte.
 - Recherche : ChatGPT nous a permis d'explorer différentes technologies que nous devons utiliser dans ce projet. Il nous a également aidés à mieux comprendre certains concepts abstraits.

Toutefois, l'ensemble des droits de propriété intellectuelle de ce document demeure exclusivement attribué à ses auteurs.