

TP1 Python Django

Installation de Django, Création d'un projet Django et de l'application Notes

TOYI Francois

October 2024

Aliassons Python 3

1. un **alias** en informatique est : Un alias est un nom ou une étiquette attribué à un fichier, un utilisateur ou une commande d'un système informatique, qui sert de raccourci ou de moyen différent de référence à un nom ou une commande compliqué.

2. Voici la version de l'alias et la version de la commande simple, on a :

```
toyi@toyi-Latitude-E6420:~$ py --version
Python 3.10.12
toyi@toyi-Latitude-E6420:~$ python3 --version
Python 3.10.12
toyi@toyi-Latitude-E6420:~$
```

Installons pip

1.

- PIP est un gestionnaire de paquets utilisé pour installer et gérer des paquets écrits en Python.
- Pour installer pip on a juste à installer la dernière version de python. Au cas contraire on feras **python -m pip install**.
- la commande à tapé est la suivante : **python -m pip install** .

2. cette gigantesque chaîne de caractères s'étalant sur la totalité du fichier servira à

3. La commande que j'ai utilisé est `py get-pip.py`. Il y a une erreur. Pour contourner ce problème on doit faire appel à `sudo`.

4. La commande que j'ai tapé est la suivante : **`sudo py get-pip.py`**.

5. la commande `pip --version` renvoie :

```
toyi@toyi-Latitude-E6420:~/Documents/L3/sem_5/django$ pip --version
pip 24.2 from /home/toyi/.local/lib/python3.10/site-packages/pip (python 3.10)
toyi@toyi-Latitude-E6420:~/Documents/L3/sem_5/django$
```

Création d'un environnement virtuel

1. Un environnement virtuel est un environnement d'exécution isolé. Il permet d'isoler les paquets utilisés pour un projet. Nous allons utiliser ça dans ce cours afin d'avoir des versions différentes pour chaque projet.

2. La version de `virtualenv` est la suivante :

```
toyi@toyi-Latitude-E6420:~/Documents/L3/sem_5/django$ virtualenv --version
virtualenv 20.26.6 from /home/toyi/.local/lib/python3.10/site-packages/virtualenv/__init__.py
toyi@toyi-Latitude-E6420:~/Documents/L3/sem_5/django$
```

3. Pendant la création de l'environnement virtuel les autres outils qui sont automatiquement installés sont : le dossier `bin`, `lib` et `pyvenv.cfg`.

4. J'ai fait : **`source test_env/bin/activate`**.

Oui on a un préfixe qui signifie que l'environnement est activé.

5. Voici la version de `pip` et voici où se trouvent les sources de l'outil `pip`, on a :

```
(test_env) toyi@toyi-Latitude-E6420:~/Documents/L3/sem_5/django$ pip --version
pip 24.2 from /home/toyi/.local/lib/python3.10/site-packages/pip (python 3.10)
(test_env) toyi@toyi-Latitude-E6420:~/Documents/L3/sem_5/django$
```

C'est parce qu'on utilise l'environnement virtuel pour éviter les interférences entre les différents projets où les paquets installés globalement.

Installation de Django dans l'environnement virtuel

1. **Django** : est un **framework** web open-source écrit en Python, conçu pour faciliter le développement d'applications web robustes, sécurisées et évolutives.

Il sert à concevoir des applications web robustes, sécurisées et évolutives .

2. Voici la version de django :

```
(test_env) toyi@toyi-Latitude-E6420:~/Documents/L3/sem_5/django$ py
Python 3.10.12 (main, Sep 11 2024, 15:47:36) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> print(django.get_version())
5.1.1
>>>
```

3. Pour désactivée l'environnement virtuel on fait **deactivate** .

Il y a erreur quand parce qu'on as quitté notre environnement virtuel dans lequel on avait installé **django** .

Création d'un projet et d'un serveur de développement.

La version est : 5.1.1 .

1. La différence entre un projet et une application est qu'un projet est le conteneur principal, la configuration de haut niveau de notre site web tandis qu'une application est une fonctionnalité ou un module spécifique au sein de notre projet .

2. La commande pour créer un projet Django est : **django-admin start-project ifnti_l3** .

Dans le dossier courant il y à eut la création du dossier **ifnti_l3** qui contient le dossier **ifnti_l3** (qui a sont tour contient des fichier .py) et le fichier **manage.py**.

3. Le fichier **manage.py** sert de point d'entrée pour toutes les commandes de gestion liées à notre projet Django.

4. Le fichier **setting.py** contient toutes les configurations et paramètres nécessaires pour le fonctionnement de notre projet.

5. Le fichier **__init__.py** sert à bien structurer notre code python et facilite l'importation de module.

6. Le **serveur de développement** est un serveur léger qui est intégré dans de nombreux frameworks web.

Non, on peut pas l'utiliser en production car il n'est pas adapté à un environnement de production en raison de ses limitations en termes de performance, de sécurité et de fiabilité.

7. La commande pour démarrer le serveur est la suivante **py manage.py runserver**.

8.

- http : est un protocole de communication utilisé pour transmettre des données sur le web.
- 127.0.0.1 : est l'adresse IP de bouclage (ou localhost), qui fait référence à l'ordinateur sur lequel on lance notre serveur.
- 8000 : est le numéro de port sur lequel le serveur web écoute les requêtes.

Création de l'application Notes

1. Pour quitter le serveur on fait **Ctrl+C**.

2. La commande à tapé est la suivante **py manage.py startapp notes**.

Une fois la commande lancé il y a eut la création d'un nouveau dossier notes .

3. Les différents fichiers contenus dans l'application notes sont : **admin.py**, **apps.py**, **__init__.py**, **models.py**, **tests.py** et **views.py** .

Ecrivons la première vue

1. Une vue selon Django est une fonction ou une méthode qui traite les requêtes HTTP et renvoie des réponses HTTP.

La différence avec le pattern MVC c'est qu'en django **la vue** fait le travail de ce qui serait généralement le contrôleur dans un modèle MVC classique.

2. Voici le contenu de notes/views.py , on a :

```
from django.shortcuts import render
```

```
# Create your views here.
```

Le code qu'on vas placé vas nous permettre d'afficher dans le navigateur **Bonjour tout le monde !**

3. Pour créer le fichier urls.py dans le dossier notes/, on a :

```
cd notes
touch urls.py
```

4. Ce code définit une URL pour l'application et associe cette URL à une vue spécifique (la fonction **index**).

INDEX est le nom de la vue qui sera appelée lorsque l'utilisateur accède à cette URL.

5. Voici le contenu du fichier `urls.py` après l'ajout de cette ligne est la suivante, on a :

```
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('notes/',include('notes.urls'))
]
```