

Fiche méthode

Installer et tester un projet Flask dans un environnement virtuel

Fiche méthode à utiliser pour la réalisation des travaux pratiques :

- Fiche 11 : Créer une page dynamique avec Flask, page 23 ;
- Fiche 12 : Sécuriser le formulaire, page 25.

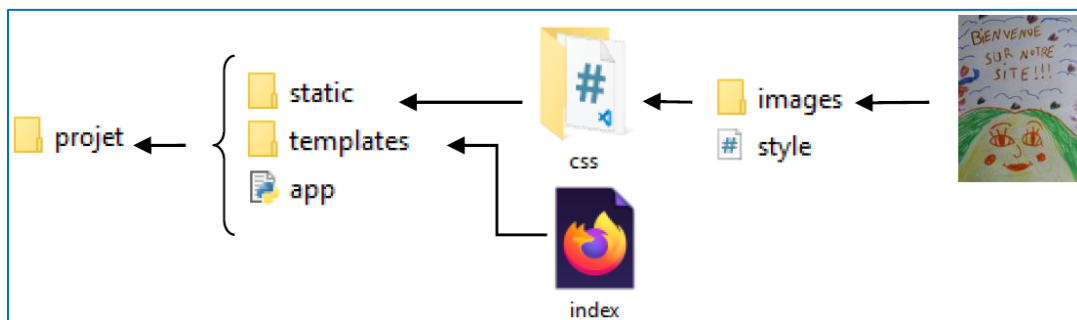
Objectif : créer et lancer un environnement virtuel de développement appelé *envtest*, puis y installer Flask afin de tester le script *app.py*. Le projet est placé dans un dossier appelé *projet*.

Étape 1 Organiser les fichiers et nommer les dossiers

Flask utilise le moteur de *templates Jinja2* et le module *render_templates*. Afin que l'ensemble fonctionne correctement, les fichiers doivent respecter un nommage et une arborescence précise :

- placer le (ou les) fichier(s) html dans le dossier nommé *templates*,
- placer le (ou les) fichier(s) css et les images dans un dossier css lui-même dans un dossier nommé *static*.

Exemple d'arborescence



Noms des fichiers :

- Fiche 11, page 23 : *projet* est remplacé par *projet1* et *app* par *app1*.
- Fiche 12, page 25 : *projet* est remplacé par *projet2* et *app* par *app2*.

Étape 2 Créer un environnement de développement virtuel

Un environnement de développement virtuel peut être créé à n'importe quel emplacement (clé USB ou disque dur, dans la session professeur, la session élève, les documents, etc.).

1. Ouvrir l'invite de commande et se placer dans l'emplacement choisi.

Pour passer du disque dur (C :) à la clé USB, ici D (cela peut être E, F...), on saisit **d :** puis on appuie sur la touche Entrée.

```
C:\Users\Utilisateur>d:
D:\>
```

2. Se placer dans le répertoire *projet* avec la commande **cd projet** puis Entrée (cd signifie *change directory*).

```
D:\>cd projet
D:\projet>
```

3. Créer l'environnement virtuel nommé ici *envtest* avec la commande **python -m venv envtest** puis Entrée.

```
D:\projet>python -m venv envtest
```

Attendre le retour de : **D:\projet** à l'écran.

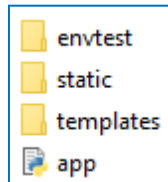
Remarque : patienter, l'installation peut prendre jusqu'à 3 minutes.

Noms des fichiers

Le nom *envtest* est donné comme exemple.

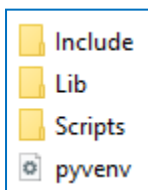
- Fiche 11, page 23 : *envtest* peut être remplacé par *envtest1*.
- Fiche 12, page 25 : *envtest* peut être remplacé par *envtest2*.

L'environnement virtuel *envtest* est maintenant créé dans *projet*.

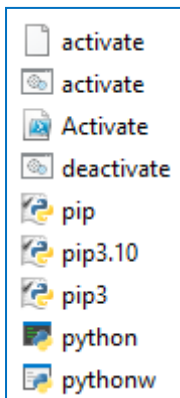


5. Explorer, par curiosité, les différents dossiers :

- *envtest*



- *Scripts*



6. Se placer dans le répertoire *envtest* avec : `cd envtest` puis Entrée.

```
D:\projet\envtest>
```

7. Activer le script *activate.bat* avec : `Scripts\activate.bat` puis Entrée.

```
D:\projet\envtest>Scripts\activate.bat
```

Remarque : « \ » (antislash) s'obtient en appuyant simultanément sur les touches Alt Gr et 8.

8. L'environnement virtuel est maintenant activé.

```
(envtest) D:\projet\envtest>
```

3. Installer Flask dans l'environnement de développement

Installer Flask dans l'environnement de développement *envtest* avec la commande `pip install flask` puis Entrée.

```
(envtest) D:\projet\envtest>pip install flask
```

Remarque : l'installation de Flask prend plusieurs secondes.

Capture d'écran de l'installation de Flask

```
(envtest) D:\projet\envtest>pip install flask
Collecting flask
  Downloading Flask-2.2.3-py3-none-any.whl (101 kB)
----- 101.8/101.8 kB 420.2 kB/s eta 0:00:00
Collecting Werkzeug>=2.2.2
  Downloading Werkzeug-2.2.3-py3-none-any.whl (233 kB)
----- 233.6/233.6 kB 340.4 kB/s eta 0:00:00
Collecting click>=8.0
  Using cached click-8.1.3-py3-none-any.whl (96 kB)
Collecting itsdangerous>=2.0
  Using cached itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Jinja2>=3.0
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
Collecting colorama
  Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.2-cp310-cp310-win_amd64.whl (16 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, Werkzeug, Jinja2, click, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.2 Werkzeug-2.2.3 click-8.1.3 colorama-0.4.6 flask-2.2.3 itsdangerous-2.1.2

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(envtest) D:\projet\envtest>
```

Étape 4 Paramétrer et lancer le serveur Flask à partir de l'invite commande

1. Revenir au répertoire *projet* avec : `cd ..` puis Entrée.

```
(envtest) D:\projet\envtest>cd..
```

On obtient :

```
(envtest) D:\projet>
```

2. Saisir `set FLASK_ENV=developement` puis Entrée.

```
(envtest) D:\projet>set FLASK_ENV=developement
```

3. Saisir `set FLASK_APP=app.py` puis Entrée.

```
(envtest) D:\projet>set FLASK_APP=app.py
```

4. Saisir `flask run` puis Entrée.

```
(envtest) D:\projet>flask run
```

On obtient :

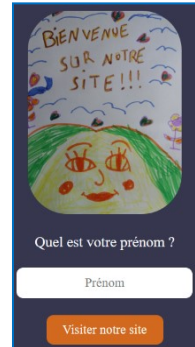
```
(envtest) D:\projet>flask run
'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
'FLASK_ENV' is deprecated and will not be used in Flask 2.3. Use 'FLASK_DEBUG' instead.
* Serving Flask app 'app.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Flask exécute `app.py`, et lance son serveur intégré à l'adresse **127.0.0.1:5000**

Remarque : pour la suite : ne pas fermer l'invite de commande afin de laisser tourner le serveur !

Étape 5 Tester le projet

1. Dans la barre URL du navigateur, saisir : **127.0.0.1:5000** puis **Entrée**.
2. Entrer un prénom puis appuyer sur le bouton « Visiter notre site ». La page affiche : « Bonjour *prénom* ! » et l'URL (route) devient : **127.0.0.1:5000/accueil1**.
3. Pour réessayer avec le message invitant à nouveau à saisir le prénom, il faut ressaisir l'URL : **127.0.0.1:5000** (rafraîchir la page dans le navigateur ne suffit pas).



Étape 6 Arrêter le serveur

Pour arrêter le serveur, appuyer simultanément sur les touches **Ctrl + C**.

```
(envtest) D:\projet>
```

Étape 7 Tester après modifications des scripts

1. Effectuer vos corrections sur vos fichiers *app.py*, *index.html* ou *style.css* si nécessaire puis enregistrer vos modifications (**Ctrl + S**).
2. Arrêter le serveur avec **Ctrl + C**.
3. Relancer le serveur avec la commande **flask run** puis **Entrée** (sans refaire toutes les étapes de la partie 4, à condition de rester dans envtest).

```
(envtest) D:\projet>flask run
```

Le serveur est relancé à l'adresse **127.0.0.1:5000**.

4. Refaire l'étape 5 pour visualiser les changements.

Étape 8 Arrêter et désactiver l'environnement virtuel

1. Arrêter le serveur avec **Ctrl + C**.

```
(envtest) D:\projet>
```

2. Désactiver l'environnement virtuel avec : **deactivate.bat** puis **Entrée**

```
(envtest) D:\projet>deactivate.bat
```

Attention : **deactivate** et non **desactivate**.

L'environnement de développement virtuel est maintenant désactivé, on se retrouve dans le répertoire *projet* :

```
D:\projet>
```

Étape 9 Travailler sur le projet après avoir quitté envtest ou fermé l'invite de commande

Dans l'invite de commande, refaire les manipulations **à partir de l'étape 4**.

Remarque : sur ce projet l'environnement virtuel de développement est déjà créé et Flask y est déjà installé.

Étape 10 Travailler sur un nouveau projet

Pour travailler sur un nouveau projet, il faut refaire toutes les étapes.