

Compte – rendu travail pratique :

API Rest en Python avec Flask

Objectifs :

- Découverte fonctionnement API
- Utilisation langage Python
- Liaison de plusieurs pages avec un API

Travail RÉALISÉ :

Création d'une page Web permettant de créer, modifier, supprimer et voir en détails les profils d'utilisateur.

I – Création page de données :

Tout d'abord on crée une page "people.py" qui nous servira de page de données.

```
61 PEOPLE = {
62     "Fairy": {
63         "lname": "Tooth",
64         "fname": "Fairy",
65         "timestamp": get_timestamp(),
66     },
67     "Knock": {
68         "lname": "Knock",
69         "fname": "Down",
70         "timestamp": get_timestamp(),
71     },
72     "Bunny": {
73         "lname": "Bunny",
74         "fname": "Easter",
75         "timestamp": get_timestamp(),
76     }
77 }
```

Si on entre dans la barre d'URL l'adresse "localhost:8000/api/people" on nous affichera cette page qui contiendra les données que l'on avait entré dans le code.

```

▼ 0:
  fname:      "Fairy"
  lname:      "Tooth"
  timestamp:  "2023-01-23 09:18:59"
▼ 1:
  fname:      "Down"
  lname:      "Knock"
  timestamp:  "2023-01-23 09:18:59"
▼ 2:
  fname:      "Easter"
  lname:      "Bunny"
  timestamp:  "2023-01-23 09:18:59"

```

II – Créer un environnement virtuel :

On crée un dossier racine au projet que l’on nomme “flask_api”.

```

c:\xampp\htdocs
λ mkdir flask_api

```

Ensuite on créera un fichier “app.py” qui sera notre principal dossier pour l’utilisation de notre API.

```

1  from flask import render_template
2  import connexion
3
4  app = connexion.App(__name__, specification_dir=".")
5  app.add_api("swagger.yml")
6
7
8  @app.route("/")
9  def home():
10     return render_template("/home.html")
11
12
13  if __name__ == '__main__':
14     app.run(host="0.0.0.0", port=8000, debug=True)

```

III – Affichage “Hello World !” :

On crée aussi une page “home.html” dans laquelle on met un code afin que lorsque on entre l’adresse d’accueil de l’API ça nous affiche ça :



Hello, world

IV – Création du “swagger.yml” :

Le swagger va nous permettre de faire un chemin jusqu’à la page que l’on va rechercher en fonction de l’adresse URL que l’on va entrer. Ce fichier va aussi nous permettre de définir des erreurs si jamais la page n’existe pas.