

## Exercice Pratique : Web Scraping avec FastAPI et BeautifulSoup

### Objectif

L'objectif de cet exercice est de réaliser un web scraping sur Wikipedia pour extraire des informations sur les pays et les stocker dans un fichier CSV. Ensuite, nous allons créer un mini-site web avec FastAPI pour afficher ces informations dynamiquement.

---

### Partie 1 : Web Scraping avec BeautifulSoup

#### Installation des dépendances

```
pip install requests beautifulsoup4 pandas fastapi uvicorn
```

#### Script de Web Scraping

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

def scrape_country_info(country_name, country_url, capital_url):
    response = requests.get(country_url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Récupération des informations générales
    population = soup.find(text="Population").find_next('td').text.strip()
    if soup.find(text="Population") else ""
    area = soup.find(text="Superficie").find_next('td').text.strip() if
    soup.find(text="Superficie") else ""
    language = soup.find(text="Langue
    officielle").find_next('td').text.strip() if soup.find(text="Langue
    officielle") else ""
    capital = soup.find(text="Capitale").find_next('td').text.strip() if
    soup.find(text="Capitale") else ""

    # Récupération des informations de la capitale
    response_capital = requests.get(capital_url)
    soup_capital = BeautifulSoup(response_capital.content, 'html.parser')

    capital_population =
    soup_capital.find(text="Population").find_next('td').text.strip() if
    soup_capital.find(text="Population") else ""
    latitude = soup_capital.find("span", class_="latitude").text if
    soup_capital.find("span", class_="latitude") else ""
    longitude = soup_capital.find("span", class_="longitude").text if
    soup_capital.find("span", class_="longitude") else ""
```

```
    return [country_name, population, area, language, capital,
            capital_population, latitude, longitude]

# Exemple pour le Japon
japan_info = scrape_country_info("Japon",
                                  "https://fr.wikipedia.org/wiki/Japon",
                                  "https://fr.wikipedia.org/wiki/Tokyo")

# Sauvegarde dans un fichier CSV
df = pd.DataFrame([japan_info], columns=["Nom du pays", "Population",
                                         "Superficie", "Langue officielle", "Capitale", "Population capitale",
                                         "Latitude", "Longitude"])
df.to_csv("countries_info.csv", index=False)
```

---

## Partie 2 : Création du mini-site avec FastAPI

### Installation des dépendances FastAPI

```
pip install fastapi uvicorn pandas
```

### Script FastAPI

```
from fastapi import FastAPI
import pandas as pd

app = FastAPI()

def load_data():
    return pd.read_csv("countries_info.csv").to_dict(orient="records")

@app.get("/countries")
def get_countries():
    data = load_data()
    return [country["Nom du pays"] for country in data]

@app.get("/country/{country_name}")
def get_country_info(country_name: str):
    data = load_data()
    for country in data:
        if country["Nom du pays"].lower() == country_name.lower():
            return country
    return {"error": "Pays non trouvé"}
```

### Lancer le serveur

```
uvicorn script_fastapi:app --reload
```

## Accès aux données via l'API

- Liste des pays : <http://127.0.0.1:8000/countries>
- Détails d'un pays : <http://127.0.0.1:8000/country/Japon>

---

## Partie 3 : Interface Web avec Map

Utiliser JavaScript et Leaflet pour afficher une carte centrée sur la capitale.

### Exemple HTML pour afficher la carte avec Leaflet

```
<!DOCTYPE html>
<html>
<head>
  <title>Carte de la Capitale</title>
  <link rel="stylesheet"
href="https://unpkg.com/leaflet/dist/leaflet.css" />
  <script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>
</head>
<body>
  <h1>Informations sur <span id="country-name"></span></h1>
  <div id="map" style="height: 500px;"></div>
  <script>
    async function loadCountryData(country) {
      let response = await
fetch(`http://127.0.0.1:8000/country/${country}`);
      let data = await response.json();
      document.getElementById("country-name").innerText = data["Nom
du pays"];
      let map = L.map('map').setView([parseFloat(data["Latitude"]),
parseFloat(data["Longitude"]), 10]);

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png').addTo(map
);
      L.marker([parseFloat(data["Latitude"]),
parseFloat(data["Longitude"])]).addTo(map).bindPopup(data["Capitale"]);
    }
    loadCountryData("Japon");
  </script>
</body>
</html>
```

---

## Résumé

1. **Scraping des données** avec BeautifulSoup.

2. **Sauvegarde des informations** en CSV.
3. **Création d'une API** avec FastAPI.
4. **Interface web avec carte interactive** via Leaflet.

Cette solution permet de toujours avoir les informations à jour directement depuis Wikipedia. 🚀