

SAE2.02 – COMPLÉMENTS TECHNIQUES

Note technique #1

BUT Informatique

Librairies Python utiles

Les librairies qui seront utiles pour le développement du code Python sont les suivantes :

```
import json
import folium
from folium.plugins import MarkerCluster
import numpy as np
from scipy.spatial import Delaunay
from scipy.spatial.distance import cdist
```

`json` : pour la manipulation des données au format JSON

`folium` : pour la cartographie

`numpy` : pour les tableaux de données

`scipy` : pour le graphe et le calcul de distances dans l'espace.

Chargement des données Vélib'

Les données statiques des stations Vélib' sont disponibles en téléchargeant le fichier JSON à l'adresse suivante :

```
https://velib-metropole-opendata.smovengo.cloud/opendata/Velib\_Metropole/station\_information.json
```

Cependant, il est possible que les données présentent dans le fichier ne soient pas complètes, ou que le serveur soit indisponible. C'est pourquoi, une version sauvegardée de ce fichier est disponible sur Eprel.

Ce fichier pourra être téléchargé via le notebook Python en utilisant ce code :

```
# Lire les données des stations depuis un fichier JSON
json_file = "station_information.json"
with open(json_file, 'r', encoding='UTF-8') as file:
    velib = json.load(file)

# Extraire de la structure JSON les informations des stations
stations_data = velib['data']['stations']
```

Les données utiles sont stockées dans la variable `stations_data`. Le contenu de cette variable est affiché à l'aide de la fonction `display` (propre à Jupyter Notebook) :

```
# Afficher les données utiles des stations
display(stations_data)
```

ce qui produit l'affichage :

```
[{'station_id': 213688169,
  'stationCode': '16107',
  'name': 'Benjamin Godard - Victor Hugo',
  'lat': 48.865983,
  'lon': 2.275725,
  'capacity': 35},
 {'station_id': 19179944124,
  'stationCode': '40001',
  'name': 'Hôpital Mondor',
  'lat': 48.798922410229,
  'lon': 2.4537451531298,
  'capacity': 28},
 {'station_id': 17278902806,
  'stationCode': '44015',
  'name': 'Rouget de L'isle - Watteau',
  'lat': 48.778192750803,
  'lon': 2.3963020229163,
  'capacity': 20},
 {'station_id': 36255,
  'stationCode': '9020',
  'name': 'Toudouze - Clauzel',
  'lat': 48.87929591733507,
  'lon': 2.3373600840568547,
  'capacity': 21,
  'rental_methods': ['CREDITCARD']},
 ...
 {'lat': 48.86450567259199,
  'lon': 2.3461959511041637,
  'capacity': 28,
  'rental_methods': ['CREDITCARD']},
 ...]
```

Hormis le code et l'identifiant de la station, on peut y trouver son nom, ses coordonnées géographiques (latitude et longitude) et sa capacité (le nombre de bornes d'accueil des vélos).

Ces informations sont structurées sous forme d'une liste de dictionnaires. Chaque dictionnaire renferme les informations statiques¹ d'une station.

Le nombre de stations est obtenu par la fonction `len` :

```
# Nombre de stations
print(len(stations_data))
```

¹ Des informations dites « dynamiques » sont également disponibles sur l'Open data Vélib'. Ces informations renseignent sur le nombre de vélos (mécaniques ou électriques) disponibles en temps réel, ainsi que du nombre de bornes libres. Ces informations ne sont pas utilisées dans cette SAE.

Première cartographie des stations

A présent, il s'agit d'élaborer une première visualisation des données Vélib' en affichant sur une carte de la région Parisienne les stations repérées par leur coordonnées géographiques.

Pour cela, la librairie `folium` est mise en œuvre afin d'obtenir la carte de la région Parisienne et d'y placer les stations géolocalisées.

```
# Créer une carte centrée sur Paris
m = folium.Map(location=[48.8566, 2.3522], zoom_start=11)

# Ajouter des marqueurs pour chaque station
for station in stations_data:
    folium.CircleMarker(
        location=[station['lat'], station['lon']],
        radius=station['capacity']//4,
        color='blue',
        weight=1,
        fill=True,
        fill_color='blue'
    ).add_to(m)

# Sauvegarder la carte dans un fichier HTML
m.save('velib_stations_map.html')

# Afficher la carte
display(m)
```

La fonction `Map` permet de créer une carte centrée sur Paris avec un certain niveau de zoom. Cette carte est stockée dans la variable `m`.

Ensuite, une boucle permet de sélectionner chaque station de la liste et d'en extraire les coordonnées géographiques (`lat`, `lon`) afin de placer sur la carte (`add_to`) un marqueur circulaire (`CircleMarker`) dont le rayon (`radius`) est proportionnel à la capacité (`capacity`) de la station.

Finalement, la carte constituée est sauvegardée (`save`) au format HTML et affichée (`display`) dans le notebook.

