



Activité	Travaux Pratiques
Professeur	Dr. W. OUEDRAOGO
Titre	Trouver les données recherchées et exploiter les données recherchées
Auteur	Coulibaly Cheick Ahmed

I)

GitHub : c'est une plateforme qui stocke du code, mais pas que, permettant ainsi une meilleure collaboration en équipe sur un projet donnée, on dit qu'il héberge des dépôts distants. C'est un outil essentiel pour les développeurs, notamment dans le cadre de projets open source ou collaboratifs.

Voici ses principales utilisations :

‡ Gestion de versions : il permet de suivre les modifications apportées au code et de revenir à une version précédente si nécessaire.

‡ Collaboration : plusieurs développeurs peuvent travailler sur le même projet en créant des branches, en fusionnant du code et en gérant les conflits.

‡ Hébergement de projets : stocke et partage le code en ligne, mais également d'autres type de données, facilitant l'accès et la contribution.

‡ Sécurité et sauvegarde : le code est sauvegardé sur des serveurs distants, réduisant les risques de perte de données.

Fichier geojson sur GitHub répertoriant tous les aéroports : <https://github.com/josiahg/airports-geojson?tab=readme-ov-file>

Sur humdata : <https://data.humdata.org/dataset/global-airports>

II)

Google Maps ne fournit pas directement une option pour filtrer les restaurants dans un rayon exact de 1 km. J'ai alors dans un premier temps utiliser l'outil "Mesurer une distance", puis cliquer sur différents restaurants pour voir si leur distance est inférieure à 1 km, tout en les enregistrant dans "Lieux favoris".

‡ Oui on peut obtenir un fichier geoJSON :

- j'ai d'abord importer le fichier " Adresses enregistrées.json " avec takeout (renommé en restaurants.json)
- puis je l'ai ouvert sur geojson.io

‡ Cependant Geojson.io ne permet pas de filtrer directement les points par distance. Il affiche juste les données sans analyser leur proximité par rapport à un point donné, le SIAO dans notre cas.

Le problème rencontré c'est qu'il manque :

- un moyen de calculer la distance entre chaque restaurant et le SIAO
- et une façon de supprimer ceux qui sont trop éloignés (> 1 km)

‡ La solution proposée est d'effectuer un filtrage à l'aide d'un script python. Le script se chargera de :

- calculer la distance entre chaque restaurant et le SIAO
- garder seulement ceux à moins de 1 km
- sauvegarder le résultat dans un nouveau fichier filtré : restaurants_1km.geojson

‡ overpass-turbo

requête :

[out:json];

(

node["amenity"="restaurant"](around:1000, 12.385, -1.530);

```
way["amenity"="restaurant"](around:1000, 12.385, -1.530);  
relation["amenity"="restaurant"](around:1000, 12.385, -1.530);  
);  
out center;
```

Le résultat a été exporté dans le fichier : export.geojson