

Carnet de Laboratoire

Projet Webmapping

Vittorio Toffolutti

Lundi 05 et mardi 06 décembre 2022

- Installation de wamp sur machine personnelle selon le mail de M.Berne du 05/12/2022
- Découvre sujet et code du projet antipode
- Recherche sur l'archétype client-serveur avec Mathéo Maréchal

Lundi 12/12 matin

- Compréhension des enjeux
- Création de maquette sur la base de mon jeux de donnée futur pour Mathéo
- Exploration des différentes ressources pour créer la base de donnée
- Découpage du code entier de mes prédécesseurs
- Le code marche avec maquette et renvoie bien les villes sur le plan et à l'antipode
- Mise en place d'une problématique : Faire une base de donnée autonome pour y faire des requetes SQL dessus à l'aide d'un code PHP
- La fonction fetch est associé à la fonction « addlabels » ligne 178
- La fonction « change_boundaries » est appelé par la fonction « Change_boundaries » ligne 243, cette fonction est appelé lorsque l'on créer l'emprise la première fois, lorsque l'on zoom ou que l'on bouge la carte.

Mardi 13/12 matin

- Découpage de la fonction fetch, elle me renvoie l'emprise des 2 cartes donc 8 coordonnées
- La fonction fetch positionne des marqueurs et a besoin de la latitude, la longitude et le nom de la ville pour positionner un marqueur.
- Recherche sur internet d'autre site de donnée pour comparer leurs utilité avec opendatasoft, Opendatasoft est très simple d'utilisation donc je vais utiliser ce site.
- Sur ce site on peut extraire les données sous forme de CSV, JSON, excel, GEOJson, Shapefile ou kml.
- Le plus simple a utilisé est le format CSV, facile à importer et à traiter pour en faire par la suite des bdd. Le format JSON pouvait être utile aussi mais lorsque l'on extrait la donnée elle n'est pas facilement compréhensible. Elle s'extrait sous une seule ligne ce qui en fait une donnée difficile à traiter même avec des formateur JSON comme « JSONonline » car la donnée est beaucoup trop grande à formater.
- Sous forme de CSV la donnée brut extraite de OPENdatasoft fais 19 colonnes pour plus de 140000 lignes pour les villes de plus de 1000 habitants.
- La question de réduire la donnée ou non se pose, on verra par la suite si on peut faire une bdd avec autant de donnée ou si on doit la réduire. On peut utiliser un filtre très facilement sur Opendatasoft pour sélectionner les données que l'on veut. Ici j'ai réalisé différents tests sur la caractéristique population, il suffit de mettre dans la barre de recherche : population > X, X étant le minimum de population par ville voulue.

Samedi 17/12 matin

- Recherche sur postgresql, postgis, pgAdmin, phpmyadmin, MYSQL.
- Minimalisation de la donnée : création du csv final

Pour cela lorsque l'on extrait le dossier csv de opensoftdata, on va dans excel, on supprime toute les colonnes à part ASCII Name, Population et Coordinates. On choisit ASCII Name et non pas la colonne Name car ASCII est un code normalisé utilisé pour l'échange de données informatiques, qui définit les représentations d'un jeu de caractères. On va ensuite séparer la colonne Coordinates en 2 car elle contient latitude et longitude. On renomme les colonnes « city », « population », « longitude » et latitude » pour correspondre aux attentes du client. Il faut aussi impérativement enregistrer le fichier csv sous « CSV UTF-8(délimité par des virgules)(*csv) », attention le délimiteur est « ; » et la virgule comme précisé dans le format d'enregistrement.

- Méthode d'importation dans PGADMIN4 : importation du csv dans pgAdmin pour avoir la bdd finale

2 méthodes existent. D'abord il faut télécharger postgresql 14 (pgadmin 4 est compris dedans). Pour les 2 méthodes il faut déjà faire une table vide avec le bon nombre de colonne prédéfinis avec le même nom que les colonnes du csv ainsi que la bonne « data type ». « Text » pour du texte, « integer » pour des nombres entier, et « double precision » pour des nombres à virgules. Ensuite on peut utiliser directement dans Query Tool cette requête :

```
« COPY tabledarrive FROM "path/du/csv" WITH (FORMAT CSV, HEADER true, DELIMITER  
';'); »
```

- tabledarrive est le nom de la table vide dans pgadmin
- "path/du/csv" est le chemin vers votre csv

Ou alors on peut aussi très bien importer la table du csv en faisant : clique droit sur notre table vide, Import, dans la fenêtre qui s'ouvre il faut renseigner le fichier csv et le type de donnée, et il faut aussi mentionner le type de délimiter et si il y'a un header ou non.

Mardi 27/12 matin

- Exploration d'autre moyen de faire des bdd :
 - BDD sur internet directement, très vite payant si bdd trop grosse, exemple : always data.
 - BDD sur phpmyadmin directement avec wamp : trop petite place disponible
 - Requête directement sur l'API de Opendatasoft

Samedi 7/01 journée entière

- Partie la plus importante : connecter la base de donnée dans pgAdmin avec PHP :

2 méthode : avec pg_connect() ou avec PDO.

Recherche sur internet pour se connecter une bdd de php. Plusieurs problème de connexion.

Si erreur de libpq : <https://stackoverflow.com/questions/62807717/how-can-i-solve-postgresql-scram-authentication-problem>

Allez sur ce lien pour essayer de changer de méthode d'authentification, il faudra passer par une invite de commande. Il faudra aussi changer des fichiers dont : « pg_hba.conf ».

Si toujours erreur de libpq, tout désinstaller : postgresql et wamp et tout réinstaller. Attention il faudra aussi supprimer le fichier « PostgreSQL » qui se situe généralement dans « C:\Program Files ». Sinon il y aura des conflits.

Bien rentré le mdp le nom de la bdd dans pgadmin, l'host le port et l'user. Tout cela est dans pgadmin.

A ce stade, la base de donnée est faite sur postgresql en local, la connexion est établie entre postgresql et php, il reste à faire les bonnes requêtes en fonction des données récupérées du fetch et bien renvoyer le bon type de donnée comme veut la partie cliente.

Samedi 28/01 matin

- Création d'un fichier PHP qui fait la bonne requête sur la bdd et renvoie le bon format de donnée :

Je fais 2 fichier php, 1 qui va faire un requête pour la première emprise, et l'autre servira à faire la requête sur l'anti-emprise.

Faire attention aux données en entré du fichier php. Dans le fetch je récupère 2 emprises donc 8 coordonnées au totale.

Pour la sortie des donnée du PHP, je dois faire en fonction du client :

Ici je me basais sur les test unitaires qu'utilisait les étudiants avant moi pour ressortir le même type de données :

```
{
  "villes": [
    {
      "lat": -41.29,
      "lon": 174.78,
      "nom": "Wellington"
    },
  ],
}
```

Cela se présente sous la forme d'un array « villes » avec des objets qui sont composé de plusieurs caractéristiques. Je renvoie 5 villes avec comme caractéristiques : « city » qui est le nom de la ville, « population », « longitude » et « latitude ».

Je renvoie donc la même chose. On peut visualiser ce que renvoie le fichier php en ajoutant entre les lignes 187 et 188 : « console.log(r) ; » dans la console de la page web en faisant f12.

Mardi 22/02 matin

- Mise en lien des projet antipodes client et serveur avec Mathéo Maréchal :

Explication de nos projets et comment récupérer les dossiers nécessaires pour faire fonctionner le projet entier.

- Vous retrouverez dans mon github : « https://github.com/Vitto50/Webmapping_project » l'ancien projet reliée à la base de donnée, il possède encore beaucoup de bug et la partie cliente de ce projet n'est pas finis.
- Il y a aussi la partie serveur du projet réalisée avec Mathéo Maréchal sur son github : « https://github.com/Matheoia/WebMapping_Antipode »

