

Créer et visualiser des jeux de données OSM avec l'API OverPass

Sommaire

- Introduction
- Exemple 1 : Bornes de recharge pour véhicules électriques
- Exemple 2 : Afficher les mairies du Grand Nancy
 - Critères de sélection
 - Requête
- Exemple 3 : Pistes et voies cyclables

Introduction

Grâce à l'API [overpass](#) et aux services proposés par le serveur [overpass-turbo](#), on peut facilement créer des cartes et extraire des jeux de données en provenance d'OpenStreetMap, les convertir, les partager...

Cette API est pratique pour avoir un aperçu des données ou une petite extraction. Pour un travail plus abouti, mieux vaut utiliser jOSM et des outils comme osmosis.

Exemple 1 : Bornes de recharge pour véhicules électriques

Dans OpenStreetMap, les bornes de recharges sont définies par le tag `amenity=charging_station`.

Dans <http://overpass-turbo.eu/>, copier les instructions suivantes puis cliquer sur "Exécuter"

```
node
  [amenity=charging_station]
  ( {{bbox}} );
out;
```

Lien vers la requête complète : <http://overpass-turbo.eu/s/dCZ>

```
node
  [amenity=charging_station]
  ( {{bbox}} );
out;
```



Exemple 2 : Afficher les mairies du Grand Nancy

Critères de sélection

On crée une requête en filtrant l'information à l'aide des tags (données attributaires). On peut s'inspirer de la [liste des critères définie pour les POI dans G-Ny](#).

- Les mairies sont identifiées par la clé **amenity=townhall**.
- L'emprise du Grand Nancy a été défini pour G-Ny à Est/Droite="6.38821" Nord/Haut="48.788" Sud/Bas="48.595" Ouest/Gauche="6.021".
- Le tag `amenity=townhall` peut aussi bien concerner un point/noeud (node) qu'une surface (area). Donc on doit réunir 3 requêtes selon chaque type d'objet : node, way ou relation.

Requête

Aller sur <http://overpass-turbo.eu/> et saisir dans la fenêtre de gauche.

```

<osm-script>
  <union>
    <query type="node">
      <has-kv k="amenity" v="townhall"/>
      <bbox-query e="6.38821" n="48.788" s="48.595" w="6.021"/>
    </query>
    <query type="way">
      <has-kv k="amenity" modv="" v="townhall"/>
      <bbox-query e="6.38821" n="48.788" s="48.595" w="6.021"/>
    </query>
    <query type="relation">
      <has-kv k="amenity" modv="" v="townhall"/>
      <bbox-query e="6.38821" n="48.788" s="48.595" w="6.021"/>
    </query>
  </union>
  <union>
  <item/>
    <recurse type="down"/>
  </union>
  <print/>
</osm-script>

```

Cliquer sur "Exécuter" et admirer la carte (cliquer sur la loupe pour centrer la carte), ou cliquez sur l'onglet "Données" pour obtenir les données en XML (OSM).

The screenshot shows the Overpass Turbo web interface. On the left, a text editor contains the OSM query script shown in the previous block. On the right, a map of Nancy, France, is displayed with numerous red circular markers indicating the locations of town halls. The interface includes a top navigation bar with buttons for 'Exécuter', 'Partager', 'Exporter', 'Wizard', 'Enregistrer', 'Charger', 'Paramètres', and 'Aide'. The title bar reads 'overpass turbo'. Below the map, there is a status bar showing 'Chargé - nœuds: 479, chemins: 34, relations: 0' and 'Affiché - points d'intérêt - POIs: 14, lignes: 0, polygones: 34'.

L'onglet "Données" donne le résultat en XML(OSM).

Le site propose également des fonctions pour exporter, partager et sauvegarder ses jeux de données ou ses scripts.

Exemple 3 : Pistes et voies cyclables

Cette requête récupère les pistes cyclables, chemin et voies piétonnes partagées, ainsi que les rues limitées à 30km/h.

Créée pour l'exemple, elle reste très perfectible. Si cela vous tente de l'améliorer, vous pouvez commencer avec le document [Calcul d'itinéraires vélo avec G-Ny : Balises OpenStreetMap et Règles de Calcul](#).

```
<osm-script>
  <union>
    <query type="way">
      <has-kv k="highway" v="cycleway"/>
      <bbox-query e="6.38821" n="48.788" s="48.595" w="6.021"/>
    </query>
    <query type="way">
      <has-kv k="highway" v="pedestrian"/>
      <bbox-query e="6.38821" n="48.788" s="48.595" w="6.021"/>
    </query>
    <query type="way">
      <has-kv k="highway" v="path"/>
      <bbox-query e="6.38821" n="48.788" s="48.595" w="6.021"/>
    </query>
    <query type="way">
      <has-kv k="maxspeed" v="30"/>
      <bbox-query e="6.38821" n="48.788" s="48.595" w="6.021"/>
    </query>
  </union>
  <union>
    <item/>
    <recurse type="down"/>
  </union>
  <print mode="meta"/><!-- fixed by auto repair -->
</osm-script>
```