

Initiation à OverPass

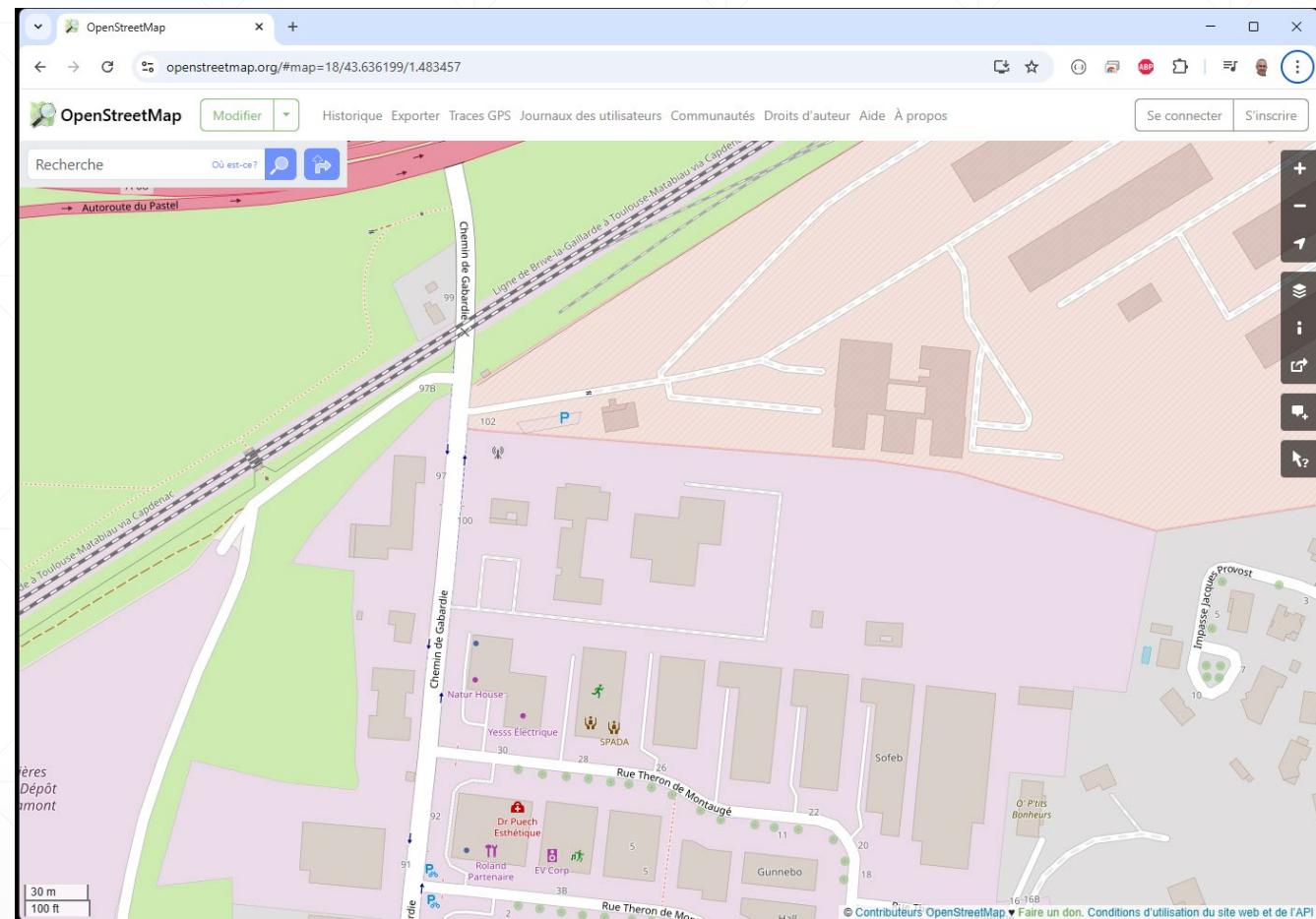
Une API pour interroger OSM



OpenStreetMap

- Une base de données
- Un site web
- Des outils
- Une communauté

OSM : outil collaboratif créé en 2004





Des sites ...

- <https://www.openstreetmap.org/>
 - <https://umap.openstreetmap.fr/fr/>
 - <https://osmand.net/>
 - <https://opentopomap.org/>
 - <https://www.lightningmaps.org/>
 - <https://map.openseamap.org/>
 - ...
-

API Overpass - <https://overpass-turbo.eu>

- L'API overpass permet d'interroger la base de données OSM
 - Le résultat peut être récupéré sous 2 types de format :
 - XML – e**X**tended **M**arkup **L**anguage
 - JSON – J**a**va**S**cript **O**bject **N**otation
-

Modèle de données OSM



- Un **noeud** (*node*) : Ce sont les éléments de base du système OSM. Les noeuds consistent en une latitude et une longitude. En gros, un point sur la carte ;



- un **chemin** (*way*) : Un chemin est une interconnexion entre au moins deux noeuds caractérisant une ligne telle qu'une rue, ou similaire ;



- Une **relation** (*relation*) : Un objet de type relation est une collection d'objets.



- L'**étiquette** (*tag*) associée à chacun des trois éléments.
-

Indexation OSM

Le schéma des tags repose sur le fonctionnement **<cle>=<valeur>**



→ Possibilité d'associer plusieurs clés à un objet

Par exemple, on cherche les autoroutes dans le secteur sélectionné

```
way
    [highway=motorway]
    ({{bbox}});
out geom;
>;
out skel qt;
```

Indexation OSM

- **Emprise de la recherche**
 - Emprise de la carte affichée (bbox) (**{{bbox}}**)
 - Utiliser un nom de lieu **{{geocodeArea:Toulouse}}->.searchArea; ...**
 - **Les requêtes :**
 - **node / way / relation**
 - **nw, nr, wr ou nwr** (requêtes agnostiques)
-

Requêtage simple

- Cela consiste à extraire des données d'OSM en utilisant une ou plusieurs valeurs **<k=v>**

Exemple :

Vous savez par son GPS que votre ami Bob est autour



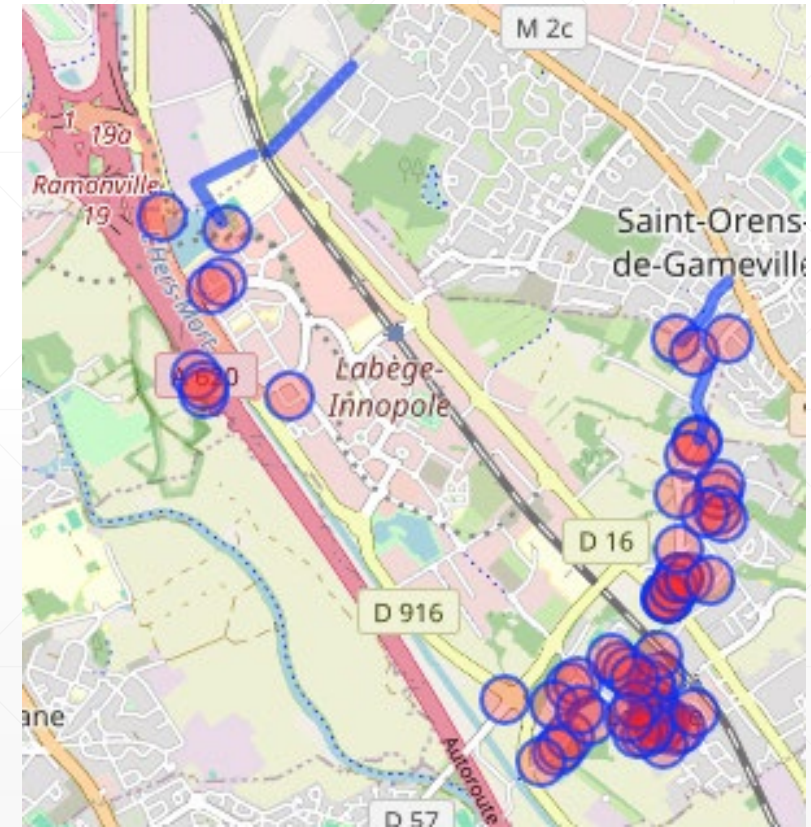
Exercice highway

Documentation de la clé **highway (route)** →
<https://wiki.openstreetmap.org/wiki/Key:highway>

- Centrer la carte autour de Toulouse dans un rayon de 20 km
 - Ecrire la requête précédente et observer le résultat
 - Modifier l'exercice précédent pour déterminer les tronçons autoroutiers limités à 90 km/h et des lampadaires autour de Toulouse
-

Exercice highway

- Centrer la carte autour de Labège
- Trouver toutes les rues (highway) limitées à 30 km/h





Exercice amenity

Documentation de la clé **amenity** (équipement) →
<https://wiki.openstreetmap.org/wiki/Key:amenity>

- Trouver tous les bars de Blagnac
 - Ajouter la contrainte qui se trouvent dans des rues limitées à 30 km/h
-

Limites administratives

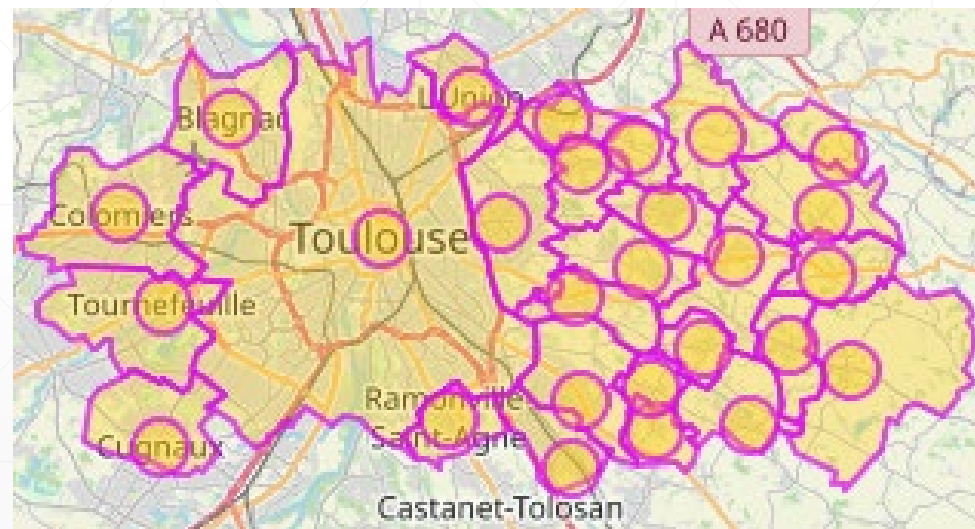
Documentation de la clé **boundary** (frontière) →
<https://wiki.openstreetmap.org/wiki/boundary>

Les types de limites pour la France

- *Régions* : 4
 - *Départements* : 6
 - *Arrondissements* : 7
 - *Communes* : 8
 - *Quartiers* : 9
-

exercice boundary

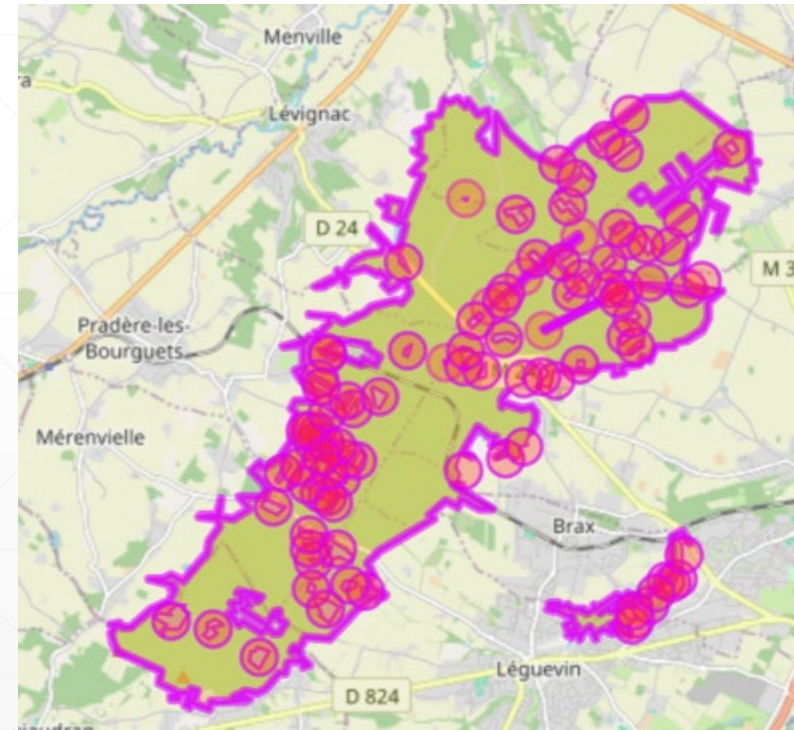
- Centrer la carte autour de Toulouse
- Chercher toutes les limites de communes autour



exercice landuse

Documentation de la clé **landuse** (usage des sols) → <https://wiki.openstreetmap.org/wiki/FR:Key:landuse>

- Trouver toutes les forêts de la commune de Léguevin

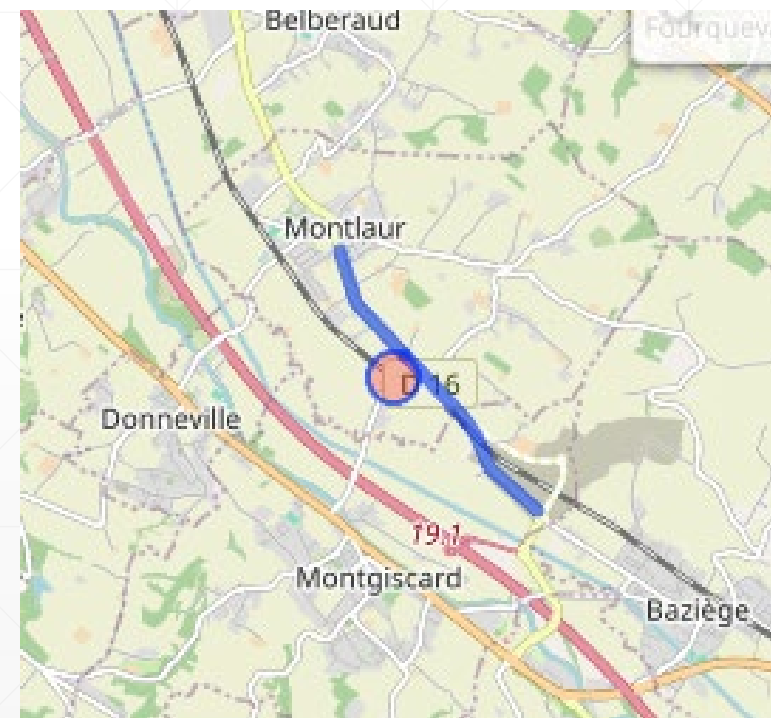


Requêtage complexe

- Les requêtes peuvent être affinées par certains critères
 - Comparaison
 - Extraction de statistiques
 - Sélection spatiale
-

Requêtage complexe comparaison

Toutes les routes de Montlaur (31) qui ont une vitesse limite de plus 50 km/h





Requêtage complexe extraction de statistiques

Afficher le nombre de bâtiments recensés de Mervilla (31) suivant leur type

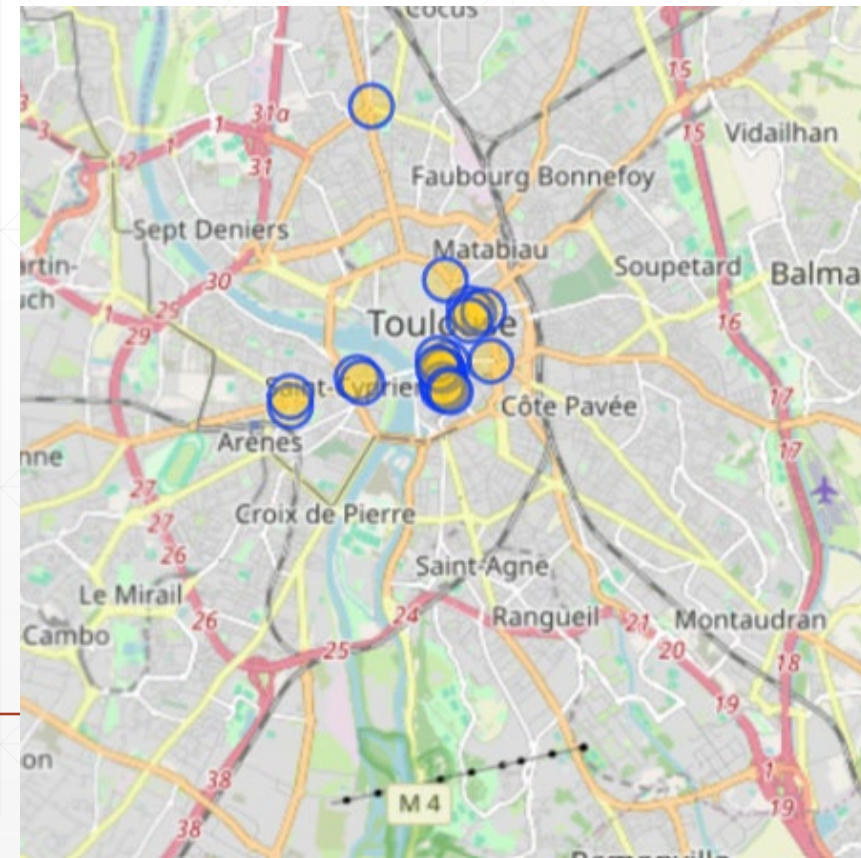
On pourra utiliser une sortie sous le format CSV (**C**omma **S**eparated **V**alue)

[out:csv(::count, ::"count:nodes", ::"count:ways", ::"count:relations")]

```
1@count @count:nodes @count:ways @count:relations
2344 0 343 1
-
```

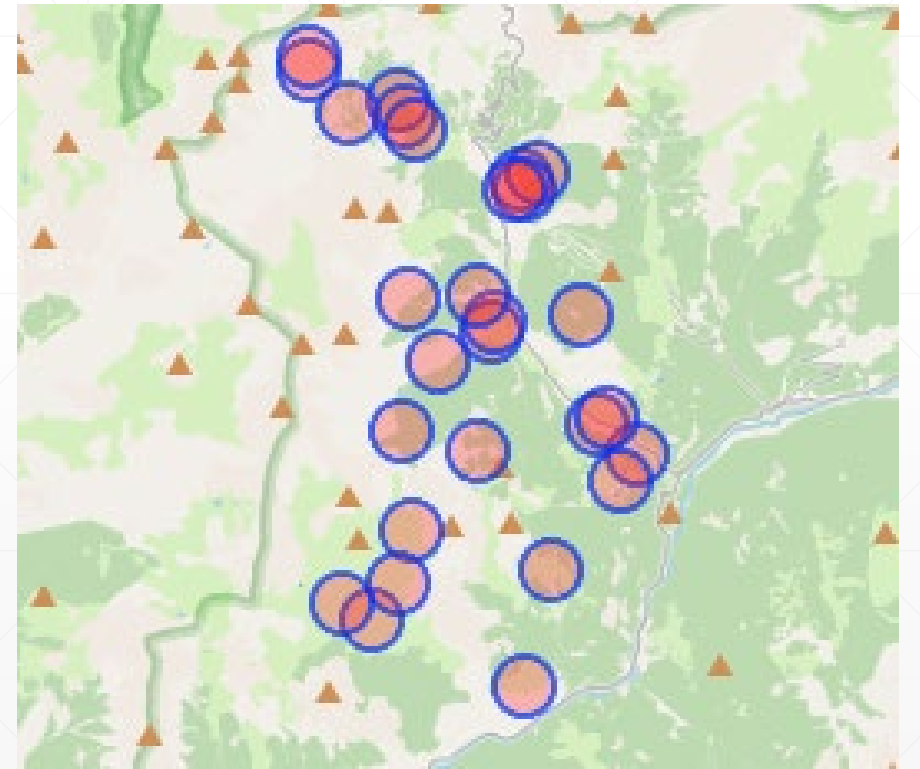
Requêtage complexe sélection spatiale

- Trouver les bars à moins de 100 m d'une station de métro



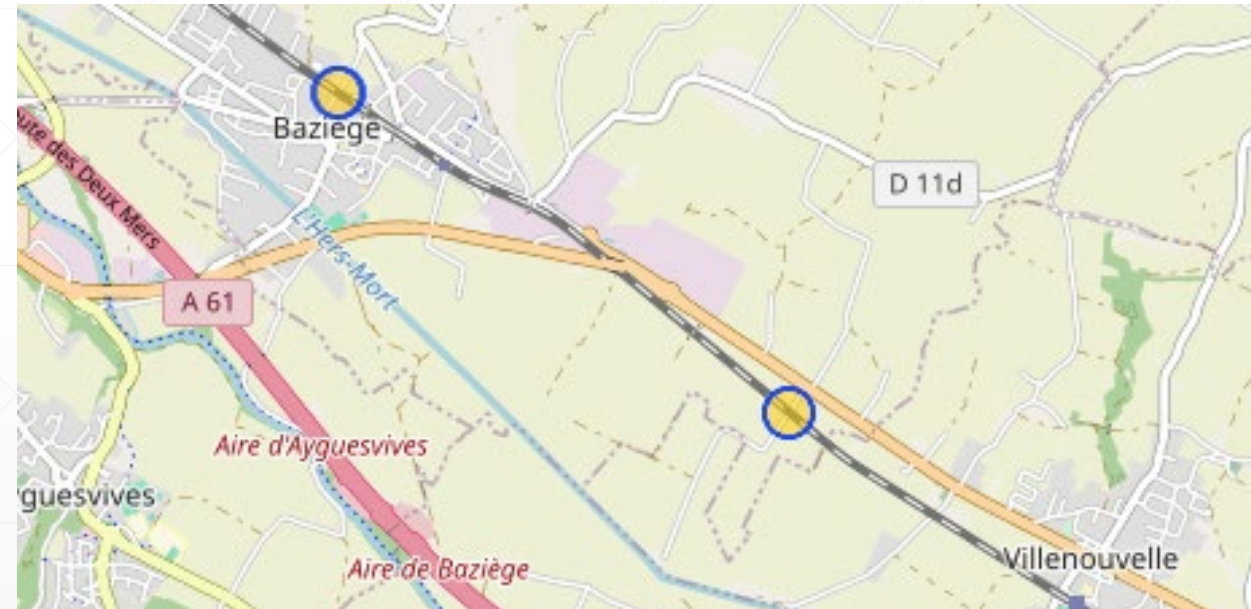
Requêtage complexe sélection spatiale

- Trouver les bâtiments isolés de Arvieux (05) à plus de 100 m d'une autre habitation



Requêtage complexe sélection spatiale

- Trouver toutes les intersections entre des routes et des voies ferrées à Baziège (31)



Requêtes et Styles

- Sélectionner et représenter les types de bâtiments
- Afficher des étiquettes
- Utiliser des icônes

Pour cela, on utilisera un style

```
{{style:
```

```
    instructions (type / style à appliquer) ...
```

```
}}
```

Requêtes et Styles sélection et représentation

- Afficher toutes les écoles de Tournefeuille (31) en bleu



Requêtes et Styles étiquettes

- Trouver les bars à Toulouse à moins de 100 m d'une station de métro (déjà fait) et afficher leur nom



Requêtes et Styles icônes

- Les icônes maki sont disponibles ici → <https://labs.mapbox.com/maki-icons>

Représenter tous les arrêts de bus de Castanet-Tolosan et afficher leur nom

{{style:

```
node[highway=bus_stop] {  
  icon-image: url('icons/maki/bus-18.png');  
  icon-width: 18;}
```

}}

MAKI

Maki is an icon set made for map designers. Maki includes icons for common points of interest like parks, museums, and places of worship. Each icon is available as a 15px by 15px SVG file. Maki is open source and [CC0 licensed](#).

Download Maki



Autres ressources

- <https://osmlab.github.io/learnoverpass/en/exercises/intro/1/>
 - GeoSpy - <https://geospy.web.app/>
 - Google StreetView
 - ...
-