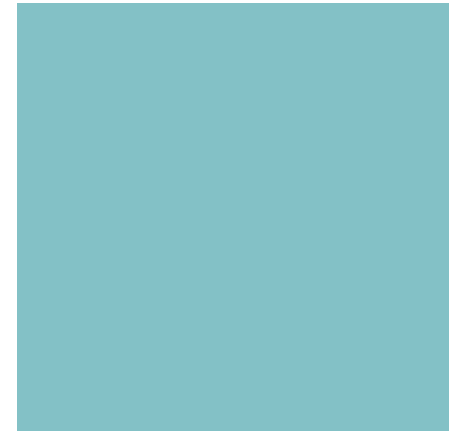
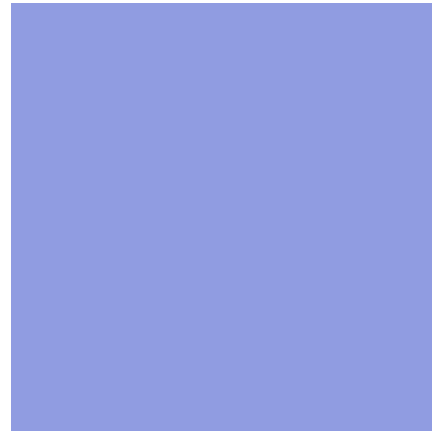
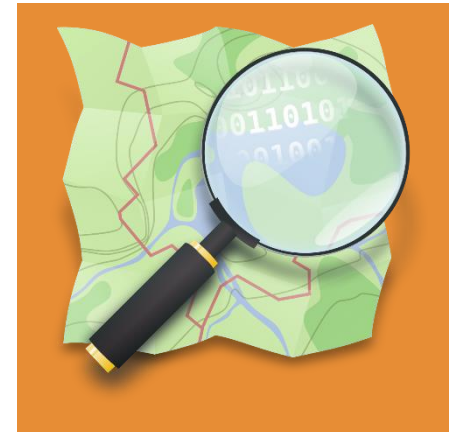




**UNIVERSITÉ
RENNES 2**

Introduction aux données OpenStreetMap

interrogation, extraction et édition



**# M2 SIGAT
Automne 2021**

@Boris Mericskay

+ Manipulation de données avec OSM



Données OSM



Quelles sont les données disponibles sur OSM ?

Comment récupérer des données issues de OSM ?

- Selon une emprise
- Selon des critères qualitatifs

Comprendre comment sont structurées ces données

Savoir comment les réutiliser...

...et les faire évoluer





Modèle de données OSM

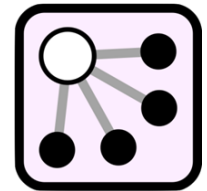
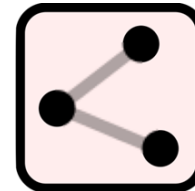
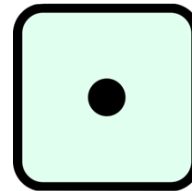


Dans un SIG, les données cartographiques sont représentées de trois façons différentes

- Points, lignes et polygones
- Les données attachées à ces objets sont généralement stockées dans une base de données liée à la base géographique.

Dans OpenStreetMap, ces trois concepts sont modélisés différemment -> 3 objets primitifs (**Eléments**) :

- **Noëuds** (*nodes*)
- **Lignes** (*ways*)
- **Relations**





Modèle de données OSM



Node

- Éléments de base du système OSM
- Les nœuds consistent en une latitude et une longitude
- Peuvent être utilisés seul ou en groupe pour former un chemin

Way

- Interconnexion entre au - deux nœuds caractérisant une ligne
- Chemin ouvert / Chemin fermé / Zones



Relation

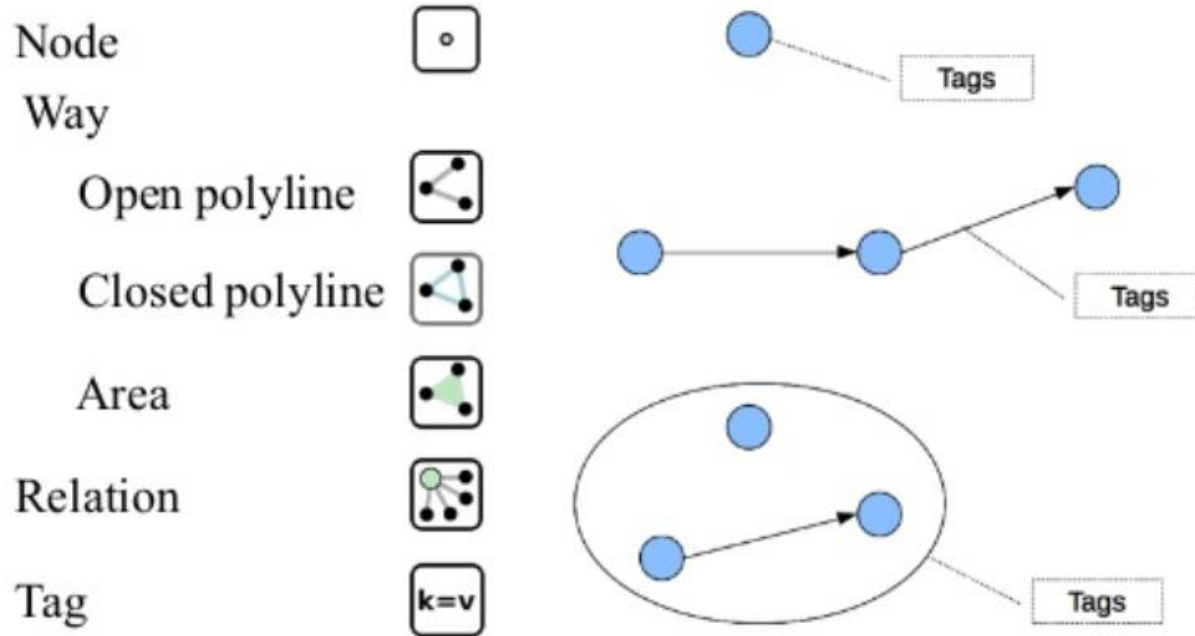
- Servent à regrouper différents objets qui considérés les uns avec les autres forment un nouvel objet (ligne de bus)



Modèle de données OSM

Clip slide

OSM Data Model



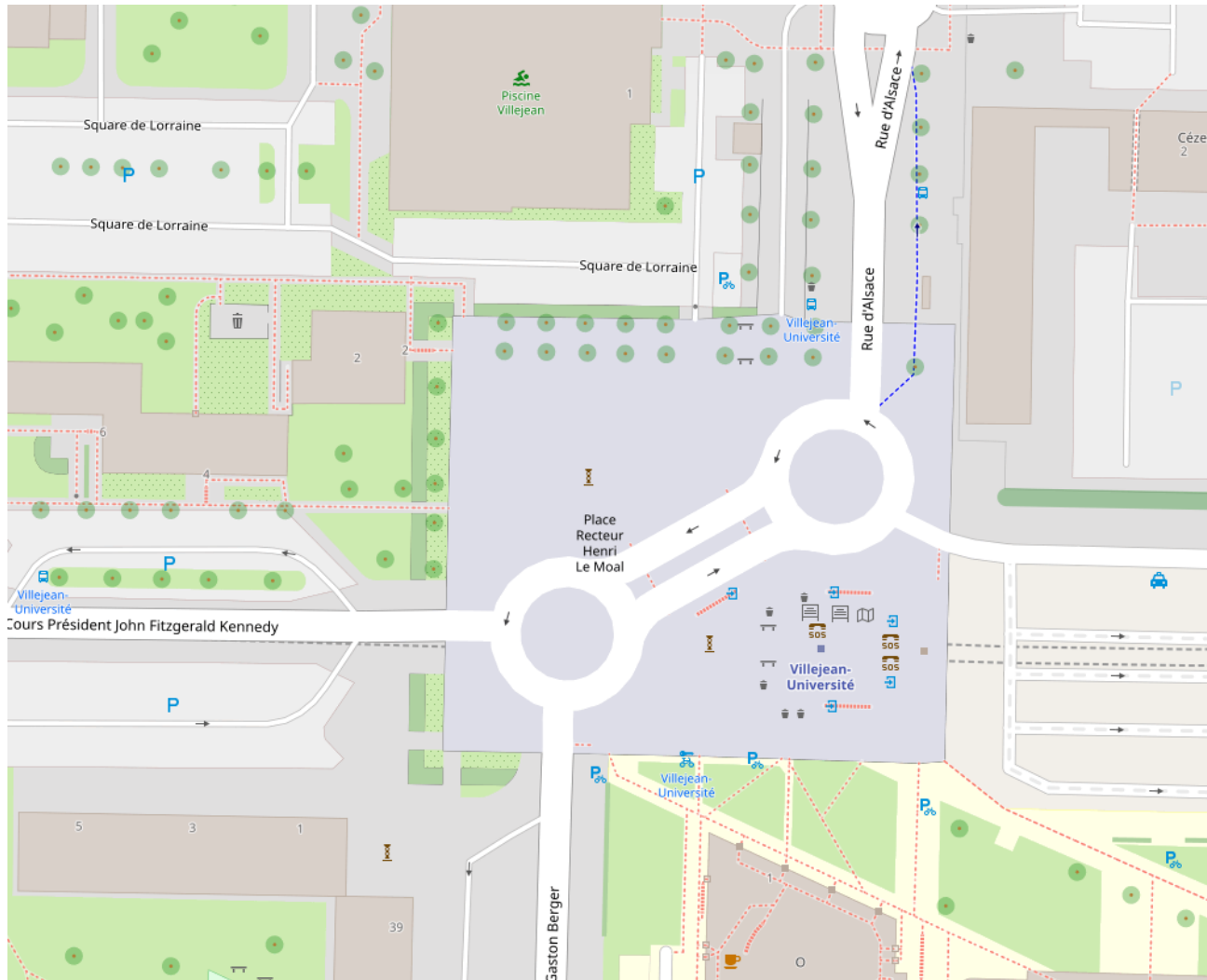


La modélisation dans OSM

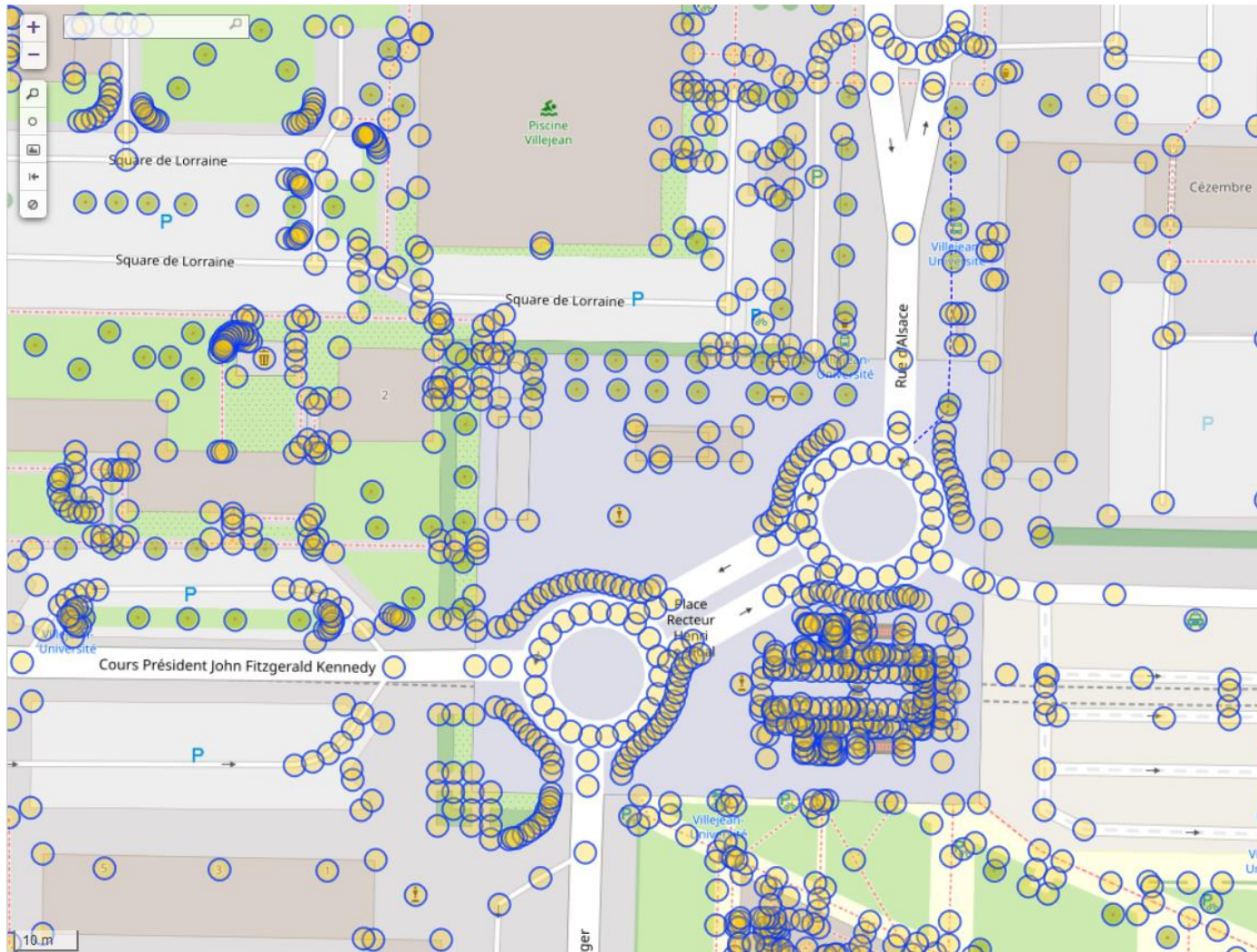




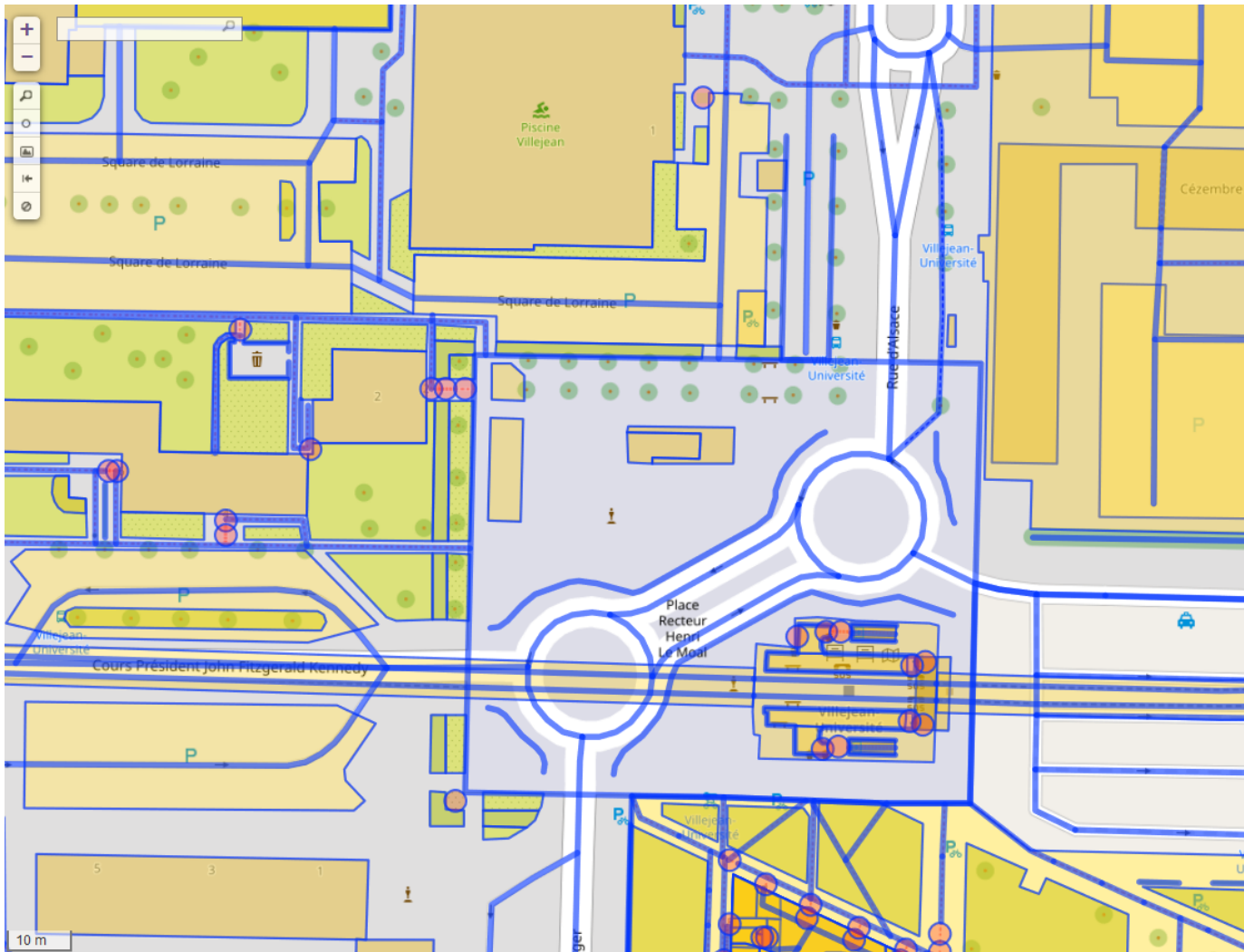
La modélisation dans OSM



+ Nodes

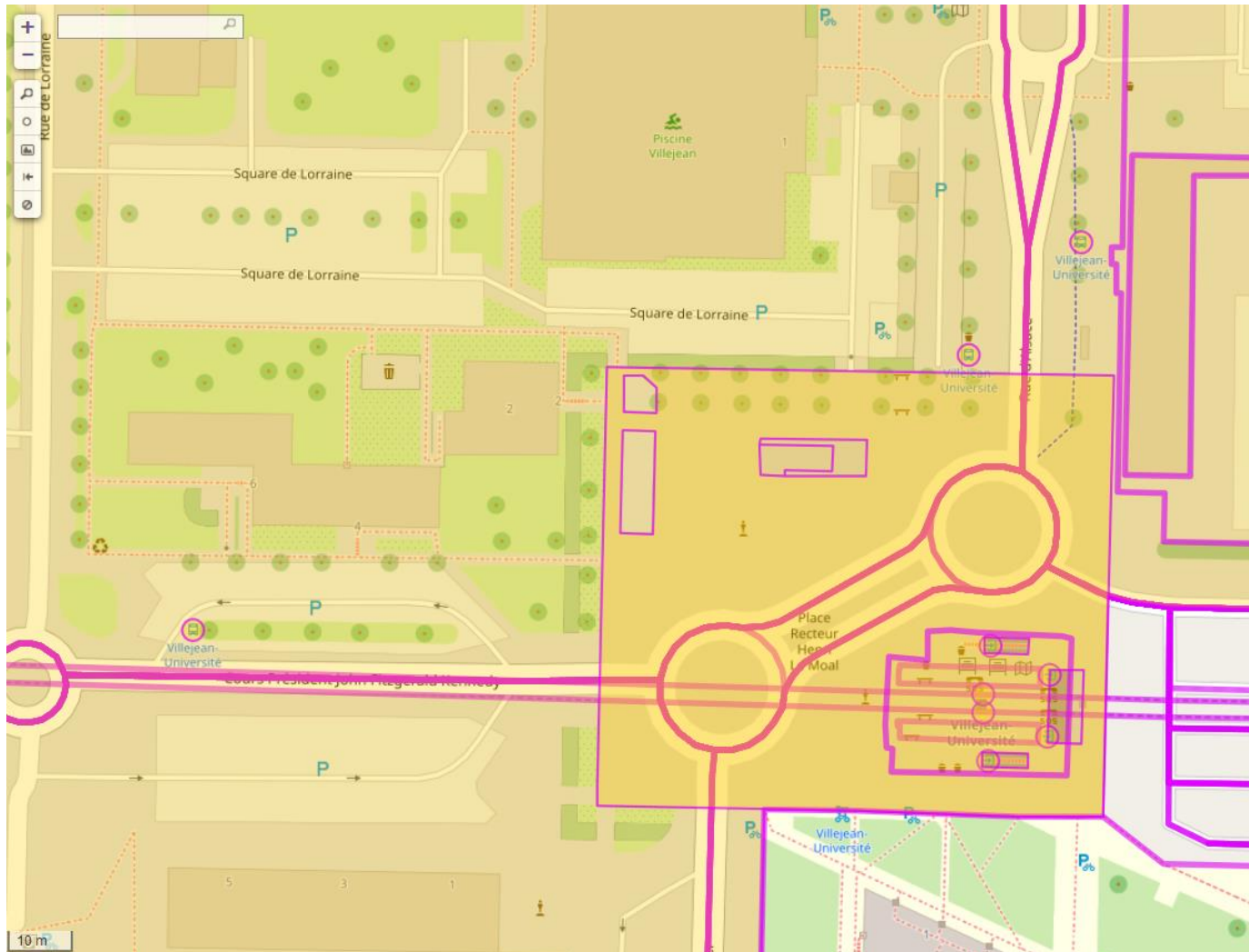


+ Ways





Relations





La folksonomie OSM



Une sémantique bien particulière

À chacun des trois éléments peuvent être associés un ou plusieurs **tags** permettant de le caractériser (étiquetage)

- **Approche basée sur une folksonomie en perpétuelle évolution**
- « Indexation personnelle, est un système de classification collaborative décentralisée spontanée, basé sur une indexation effectuée par des non-spécialistes »
- La folksonomie d'OSM bénéficie d'une grande flexibilité et surtout d'une véritable évolutivité



La folksonomie OSM

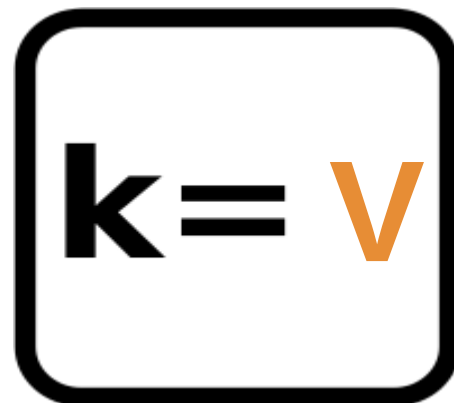


Le schéma des tags repose sur le fonctionnement
<clé>=<valeur> (*key=value*)

→ Possibilité d'associer plusieurs tags à un objet

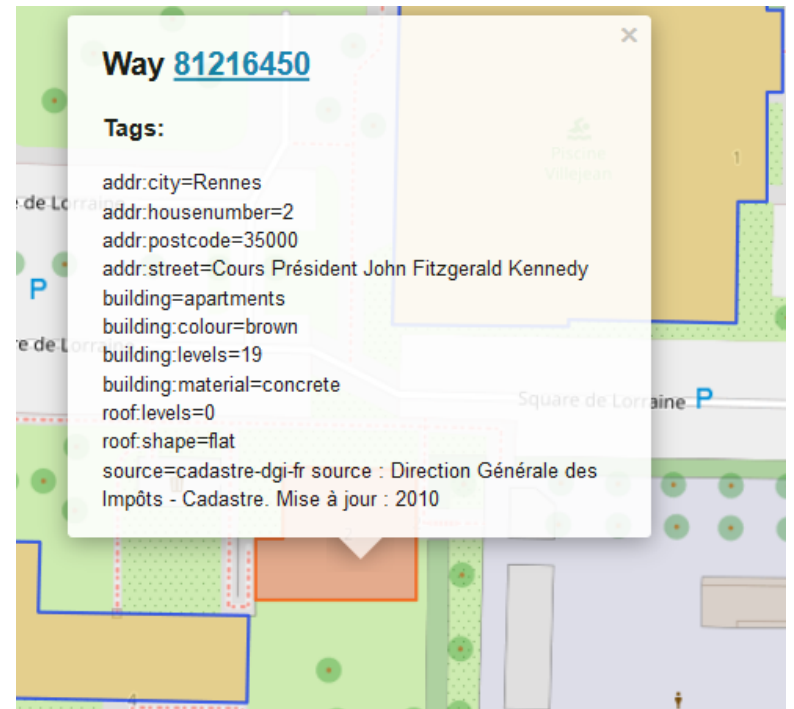
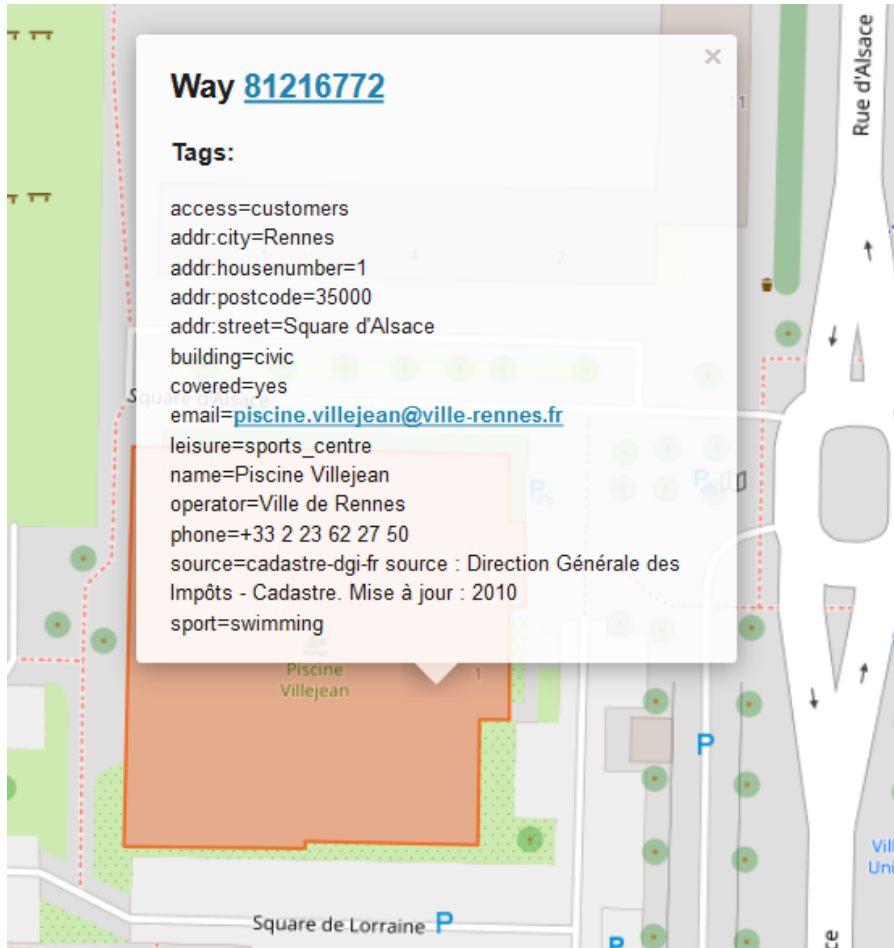
Par exemple pour indiquer qu'un trait correspond à une route secondaire, en sens unique, de vitesse maximale 90 km/h et munie d'une bande cyclable, on utilisera :

- highway=secondary
- oneway=yes
- maxspeed=90
- cycleway=lane



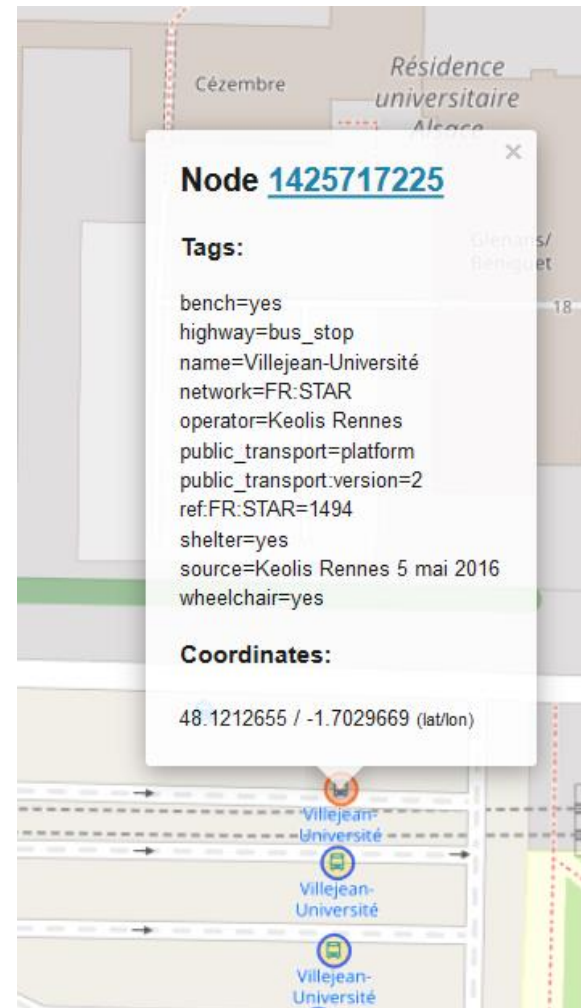
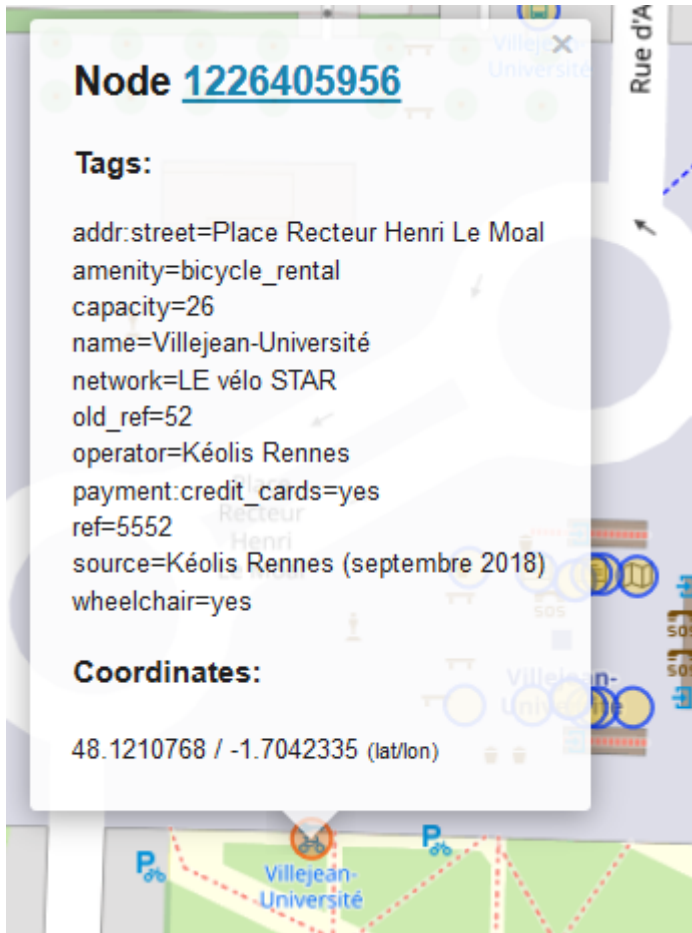


La folksonomie OSM



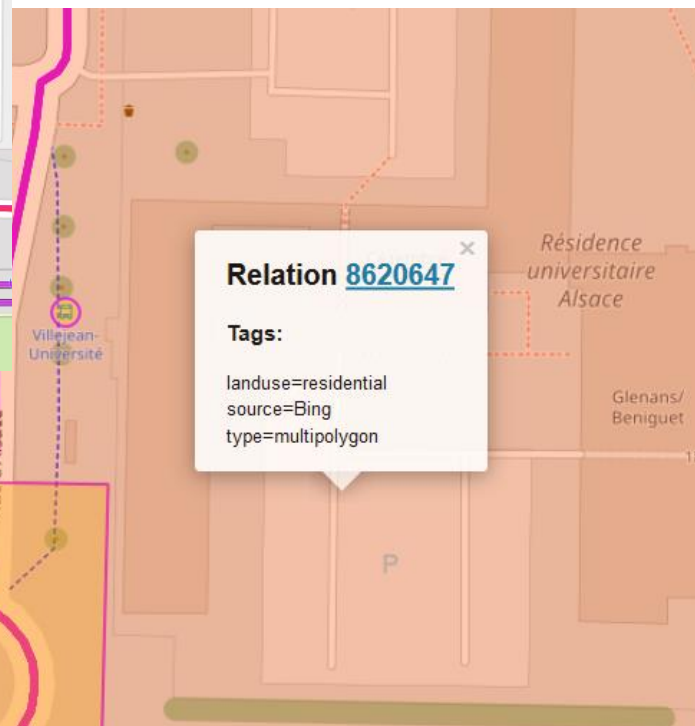
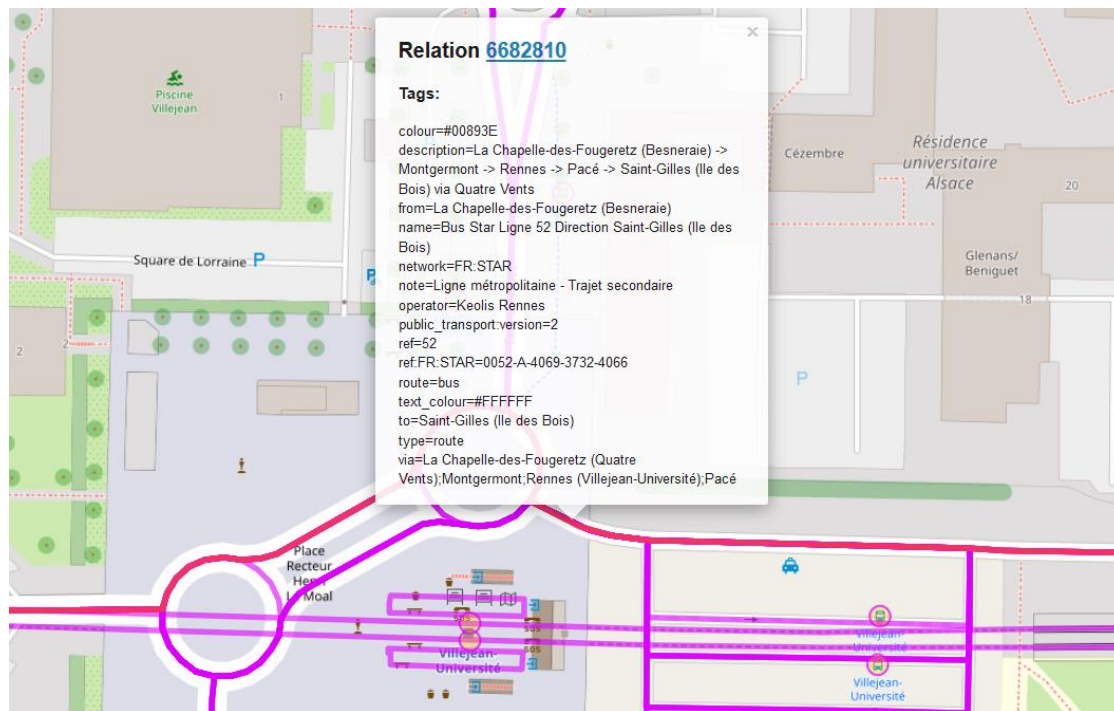


La folksonomie OSM





La folksonomie OSM





La folksonomie OSM



Routes

■ Documentation

<http://wiki.openstreetmap.org/wiki/Key:highway>

Key = **highway**

Value =

- motorway
- Trunk
- Primary
- Secondary
- ...

Key	Value	Element	Comment	Rendering	Photo
Roads					
These are the principal tags for the road network. They range from the most to least important.					
highway	motorway		A restricted access major divided highway, normally with 2 or more running lanes plus emergency hard shoulder. Equivalent to the Freeway, Autobahn, etc..		
highway	trunk		The most important roads in a country's system that aren't motorways. (Need not necessarily be a divided highway.)		
highway	primary		The next most important roads in a country's system. (Often link larger towns.)		
highway	secondary		The next most important roads in a country's system. (Often link towns.)		
highway	tertiary		The next most important roads in a country's system. (Often link smaller towns and villages)		
			The least most important through roads in a country's		



OSM et QGIS



+ OSM place search

 Cette extension est expérimentale

OSM place search

Location search helper

Based on OSM data, Nominatim online tool
(http://wiki.openstreetmap.org/wiki/Nominatim_usage_policy)

★★★★☆ 44 évaluation(s), 48085 téléchargements

Étiquettes: location search helper osm
Plus d'infos: [Page d'accueil](#) [bug tracker](#) [code repository](#)

Auteur: [Xavier Culos \(Agence de l'eau Adour Garonne\)](#)

Version installée : 0.9.3 (dans
C:\Users\mericskay_b\qgis2\python\plugins\nominatim)
Version disponible : 0.9.3 (dans Dépôt officiel des extensions
QGis)

Journal des modifications :
0.9.3 : Change from Mapquest API to OSM Api (Mapquest
require key)
0.9.2 : Fix wrong panel management. Thanks to
<https://github.com/Andre-J>
0.9.1 : Few minor changes (readme, mask dependence)
0.9 : QGis 2.4 more compatible, use mask plugin if exists
0.8.3 : fix encoding pb (<http://hub.qgis.org/issues/10920>,
thanks to Augustin Roche)

Tout mettre à jour Désinstaller Ré-installer l'extension
Fermer Aide

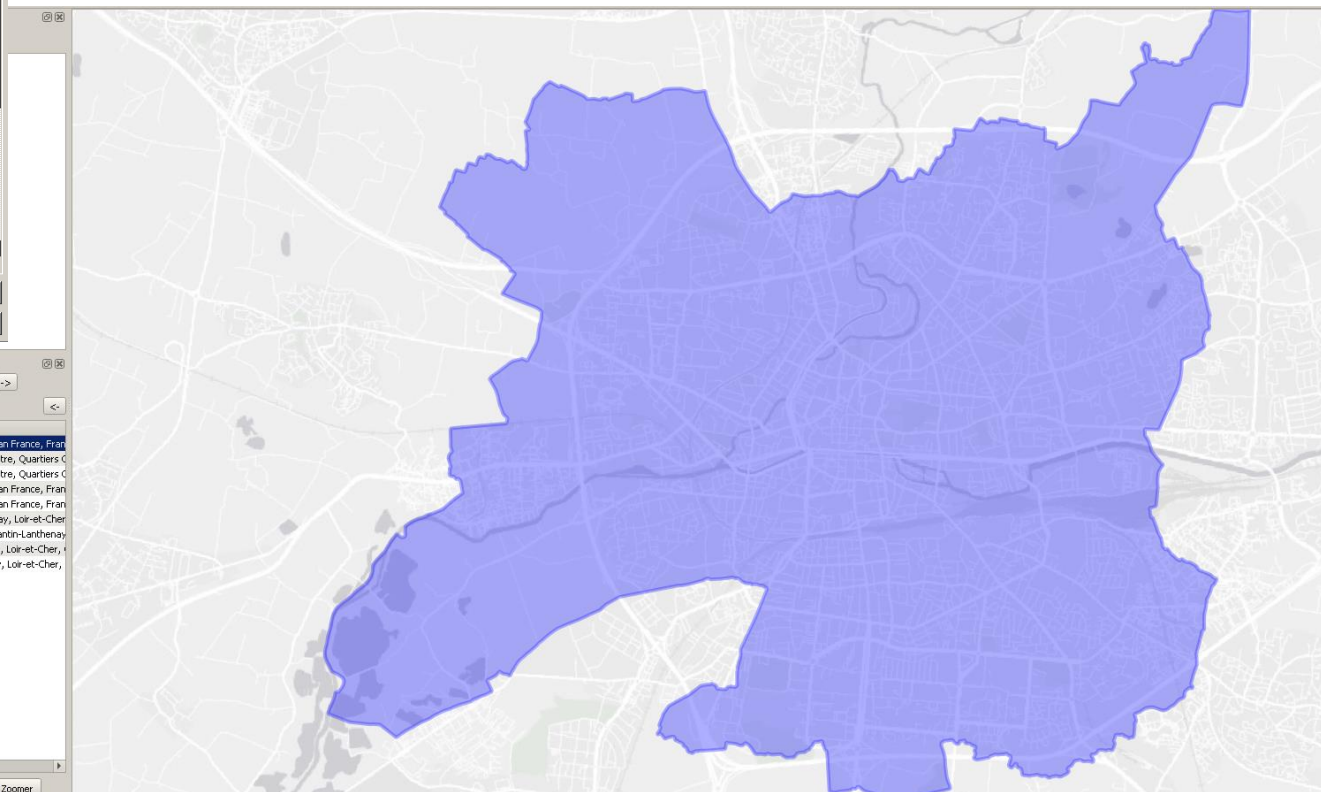
Recherche de lieux OSM

Nom contenant... rennes ->

☐ Limiter la recherche à la zone visible

Rennes, Ile-et-Vilaine, Brittany, Metropolitan France, France
Rennes, Place de la Gare, Saint-Hélier, Centre, Quartiers C
Rennes, Place de la Gare, Saint-Hélier, Centre, Quartiers C
Rennes, Ile-et-Vilaine, Brittany, Metropolitan France, France
Rennes, Ile-et-Vilaine, Brittany, Metropolitan France, France
La Rennes, Chémery, Romorantin-Lanthenay, Loir-et-Cher
La Rennes, Saint-Romain-sur-Cher, Romorantin-Lanthenay
La Rennes, Sassay, Romorantin-Lanthenay, Loir-et-Cher, France
La Rennes, Thésée, Romorantin-Lanthenay, Loir-et-Cher, France

Zoomer



+ OSM Downloader



The screenshot shows the QGIS Extensions Manager window. The left sidebar lists categories: Tout, Installés, Non installées, Invalide, Installer depuis un ZIP, and Paramètres. The main panel displays a search for 'openstreet' and a list of plugins. The 'OSMDownloader' plugin is selected and highlighted. The right panel shows the details for the 'OSMDownloader' plugin, including its description, version, and changelog.

Extensions | Tout (547)

Tout

Installés

Non installées

Invalide

Installer depuis un ZIP

Paramètres

openstreet

- Animate OSM
- HCMGIS
- Lat Lon Tools
- ☒ OpenLayers Plugin
- ☒ ORS Tools
- ☒ **OSMDownloader**
- Pelias Geocoding
- QuickMapServices
- ☒ QuickOSM
- SPARQL Unicorn Wikidata Plugin
- OSM Tools

OSMDownloader

Plugin to download OSM data by area, using a selection by rectangle. The plugin can also automatically load the osm file into QGIS in a transparent way.

Plugin to download OSM data using threads with a simple area selection. It can automatically load downloaded *.osm files as QGIS layers.

★★★★☆ 134 évaluation(s), 264608 téléchargements

Catégorie Plugins

Mots-Clés [osm](#), [downloader](#), [rectangle](#), [thread](#), [area](#), [openstreetmap](#), [tool](#)

Plus d'infos [homepage](#) [suivi des anomalies](#) [dépôt du code](#)

Auteur [Luiz Andrade](#)

Version installée 1.0.2

Version disponible 1.0.2

Changelog

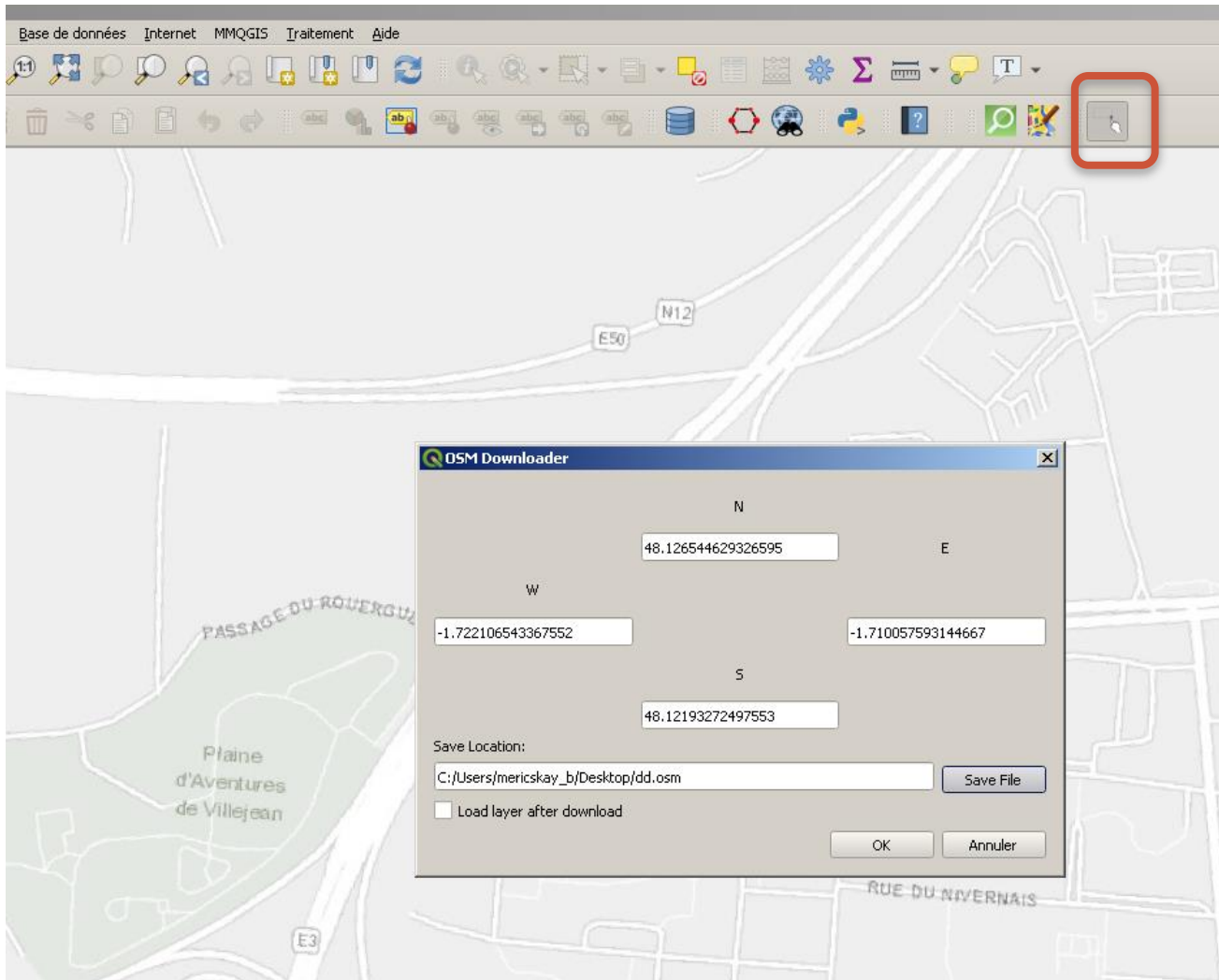
- 2015-04-10: Fixed bug to inform bad connections. Now it is possible to cancel the download by closing the progress bar. Minor fix to check if start point is equal to end point in the rectangle.
- 2015-04-13: Fixed bug that occurred when canceling a started download.
- 1.0.1 - Removing _MACOSX folder.
- 1.0.2 - Fixed an issue with proxy settings.

Tout mettre à jour

Désinstaller Ré-installer l'extension

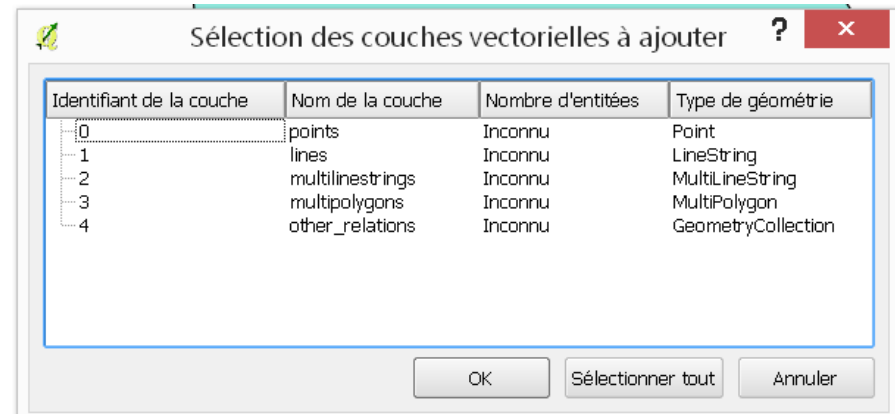
Fermer Aide

+ OSM Downloader



+ OSM Downloader

- Importer la base de données .osm
- La base de données contient 5 tables
 - Points
 - Lines
 - Multilinestrings
 - Multipolygons
 - Other_relations



- Les tables attributaires contiennent plusieurs champs correspondant à certains des tags présents dans la base OSM (qualification des données ex. type de route, type de commerce, type de sol, etc.)



Quick OSM

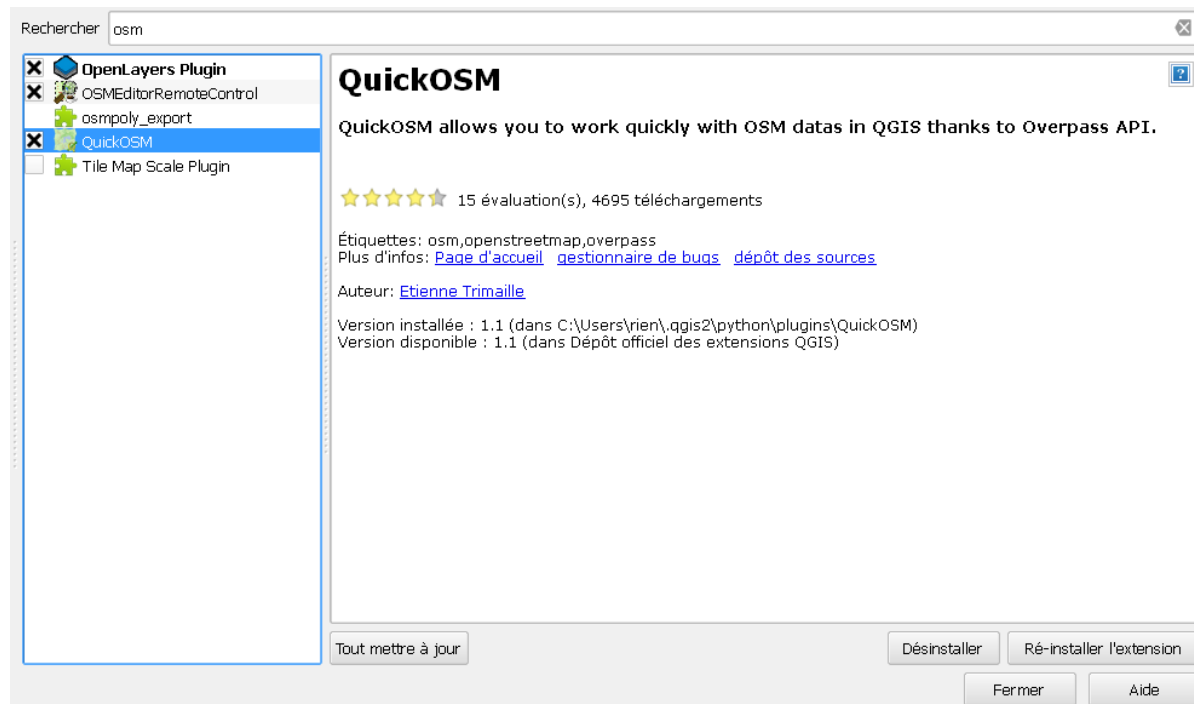




Quick OSM



- Extension QGIS qui permet de récupérer des données selon des requêtes (avec des critères) et selon une entrée géographique comme une ville par exemple
- Basée sur l'API Overpass

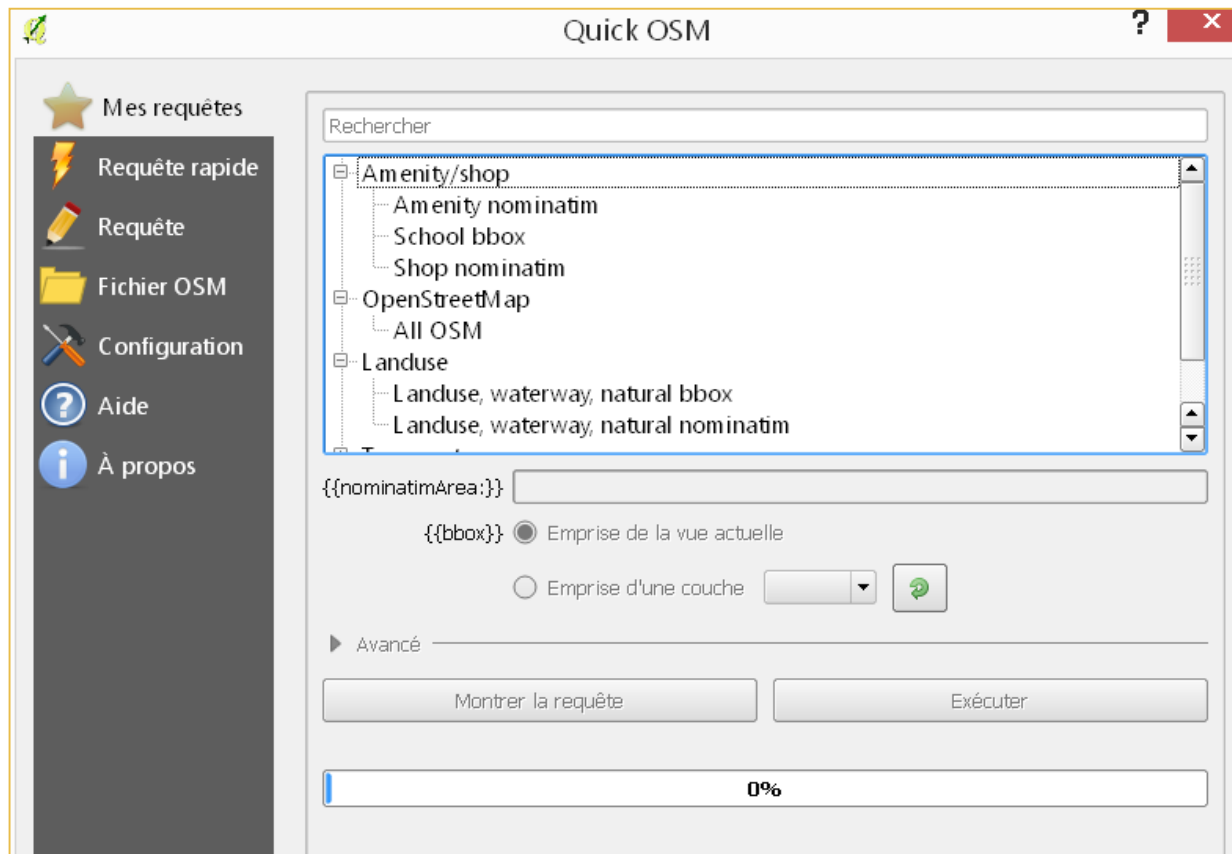




Quick OSM



"Mes requêtes" = requêtes pré-enregistrées





Quick OSM



Requête rapide permet de requêter une clé et une valeur

- Possibilité de filtrer les informations selon une entrée thématique (type de route, type de bâtiment, type de magasin, etc.)

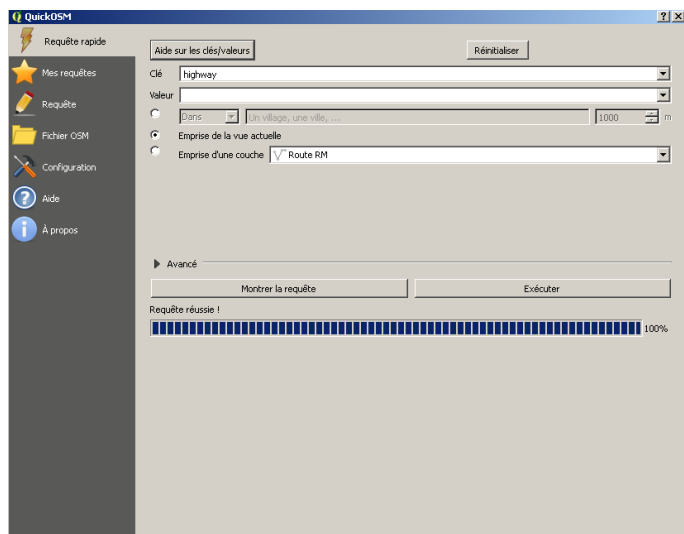
The screenshot shows the Quick OSM application window. On the left is a sidebar with navigation icons and labels: 'Mes requêtes' (star icon), 'Requête rapide' (lightning bolt icon), 'Requête' (pencil icon), 'Fichier OSM' (folder icon), 'Configuration' (wrench icon), 'Aide' (question mark icon), and 'À propos' (info icon). The main area is titled 'Quick OSM' and contains a search form. At the top of the form is a link 'Aide sur les clés/valeurs' and a 'Réinitialiser' button. The form has two dropdown menus: 'Clé' with 'building' selected and 'Valeur' with 'commercial' selected. Below these is a 'Dans' section with three radio buttons: 'Un village, une ville, ...', 'Emprise de la vue actuelle' (which is selected), and 'Emprise d'une couche' followed by a dropdown and a refresh icon. An 'Avancé' section is expanded, showing checkboxes for 'Node', 'Way', 'Relation', 'Points', 'Lines', 'Multilinestrings', and 'Multipolygons', all of which are checked. There is a 'Timeout' spinner set to 25. Below that is a 'Répertoire' dropdown set to 'Enregistrer temporairement' and a 'Parcourir' button. A 'Préfix de fichier' text field is also present. At the bottom of the form are two buttons: 'Montrer la requête' and 'Exécuter'. A progress bar at the very bottom shows '0%'.



Exercice 1



Comparer le réseau routier de la BDTOPO et les données routières issues d'OSM (key=highway) à Villejean





Exercice 1 bis

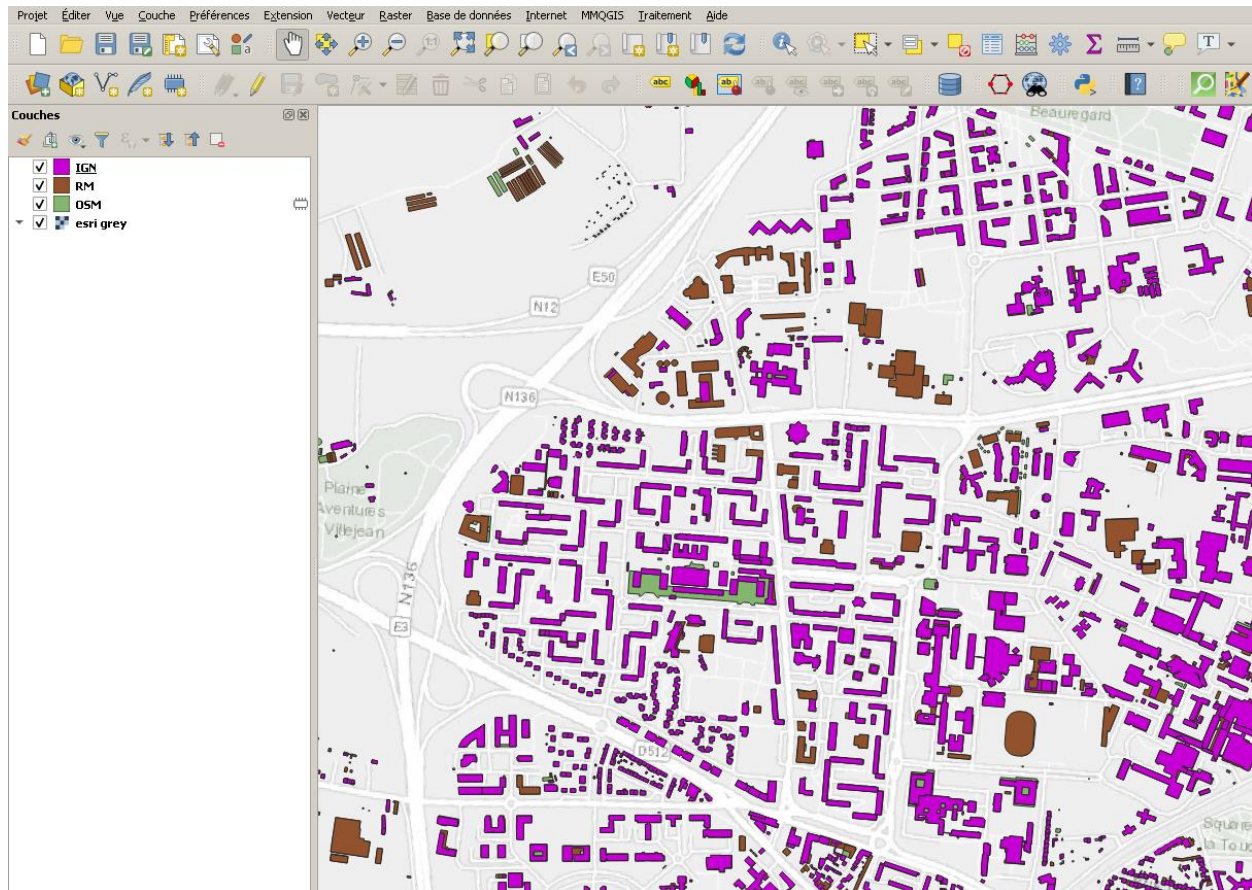
Comparer le réseau routier de la **BDTOPO** et les données routières issues **d'OSM** et celles de **RM**

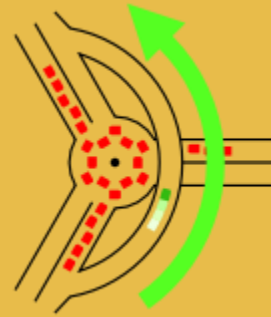




Exercice 2

Comparer les bâtiments de la **BDTOPO** avec les données issues **d'OSM** (key=building) et celles de **RM** à Villejean





Overpass
API

+

API Overpass



API Overpass



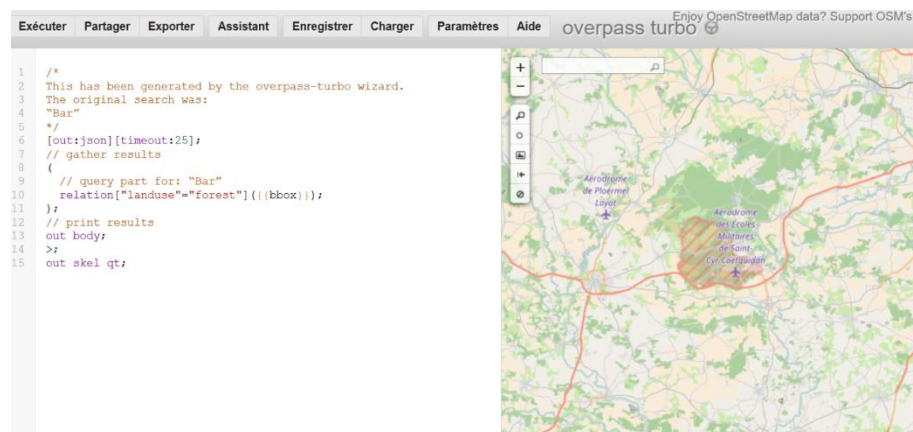
API Overpass permet d'interroger la base de données OSM

- Possibilité d'interrogation nombreuses
- Extraction de données massives et personnalisées

Overpass turbo est une application Web d'exploration de données pour OpenStreetMap

- Ce site permet d'exécuter toutes sortes de requêtes de l' API Overpass et présente le résultat sur une carte interactive.

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



+

Requêtes classiques

+ Requêtes classiques

Extraire les routes selon une valeur (hierarchie)

- Les routes principales (trunk) <https://overpass-turbo.eu/s/z8s>

The screenshot shows the Overpass Turbo web interface. The top navigation bar includes buttons: Exécuter, Partager, Exporter, Assistant, Enregistrer, Charger, Paramètres, Aide, and the logo 'overpass turbo'. Below the navigation bar, there are three colored boxes: a yellow box labeled 'Type' containing the text 'way', a green box labeled 'Key' containing the text 'highway', and a blue box labeled 'Value' containing the text 'trunk'. These boxes are connected by lines to the corresponding parts of the query in the code editor. The code editor shows the following query:

```
2 This has been generated by the overpass-turbo wizard.
3 The original search was:
4
5 way
6
7 (
8   // query part for: "route"
9   way["highway"="trunk"] ({{bbox}});
10 );
11
12 // print results
13 out body;
14 >;
15 out skel qt;
```

A red rectangle highlights the query part: `way["highway"="trunk"] ({{bbox}});`. A brown box labeled 'Emprise' points to the `{{bbox}}` placeholder. To the right of the code editor is a map of Brest, France, showing several roads highlighted in blue, representing the results of the query. The map includes labels for 'Porte de Villejean', 'Porte de Brest', 'Le Boulet', 'Rue de Gascogne', 'Avenue de Guyenne', 'Rue d'Armagnac', 'J.F. K.', 'Rue d', 'ZAC', 'Rue d', 'N 136', 'D 512', 'La Motte Gautais', 'Aire des', and 'Porte de Brest'.

<https://overpass-turbo.eu/s/NSr>



Paramètres d'export



The *out* statement can be configured with an arbitrary number of parameters that are appended, separated by whitespace, between the word *out* and the semicolon.

Allowed values, in any order, are:

- One of the following for the degree of verbosity of the output; the default is `body` :
 - ***ids***: Print only the ids of the elements in the set.
 - ***skel***: Print the minimum information necessary for geometry:
 - for *nodes*: id and coordinates
 - for *ways*: id and the ids of its member nodes
 - for *relations*: id of the relation, and the id, type, and role of all of its members.
 - ***body***: Print all information necessary to use the data. These are also tags for all elements and the roles for relation members.
 - ***tags***: Print only ids and tags for each element and not coordinates or members.
 - ***meta***: Print everything known about the elements. *meta* includes everything output by *body* for each OSM element, as well as the version, changeset id, timestamp, and the user data of the user that last touched the object. Derived elements' metadata attributes are also missing for derived elements. The modifier *noids* lets all ids to be omitted from the statement.



Paramètres d'export



out body

	id	@id	highway	int_ref	lanes	legal	lit	looks_like	maxspeed	motorroad	name	oneway	ref
1	node/6746011943	node/6746011943	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	node/6746011942	node/6746011942	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	node/6746011947	node/6746011947	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	node/6746011944	node/6746011944	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	node/6032772386	node/6032772386	give_way	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	node/5081278620	node/5081278620	motorway_junction	NULL	NULL	NULL	NULL	NULL	NULL	NULL	Porte de Brest	NULL	12
7	node/6746011941	node/6746011941	traffic_signals	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	node/6746011940	node/6746011940	traffic_signals	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	node/6767266083	node/6767266083	motorway_junction	NULL	NULL	NULL	NULL	NULL	NULL	NULL	Porte de Saint-Malo	NULL	13b
10	node/6746011948	node/6746011948	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
11	node/4249241146	node/4249241146	motorway_junction	NULL	NULL	NULL	NULL	NULL	NULL	NULL	Porte de Beuregard	NULL	13a
12	node/3794418362	node/3794418362	milestone	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	A84
13	node/4544148244	node/4544148244	give_way	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
14	node/4544148233	node/4544148233	give_way	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
15	node/3029169940	node/3029169940	give_way	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
16	node/2626016894	node/2626016894	traffic_signals	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
17	node/3560888991	node/3560888991	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
18	node/3029169958	node/3029169958	give_way	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
19	node/4616799148	node/4616799148	motorway_junction	NULL	NULL	NULL	NULL	NULL	NULL	NULL	Porte de Normandie	NULL	16
20	node/4556247439	node/4556247439	motorway_junction	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
21	node/4773815229	node/4773815229	give_way	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
22	node/4616799169	node/4616799169	motorway_junction	NULL	NULL	NULL	NULL	NULL	NULL	NULL	Porte de Normandie	NULL	16
23	node/4544148379	node/4544148379	give_way	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
24	node/1333397667	node/1333397667	give_way	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
25	node/4544148283	node/4544148283	give_way	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

out skl

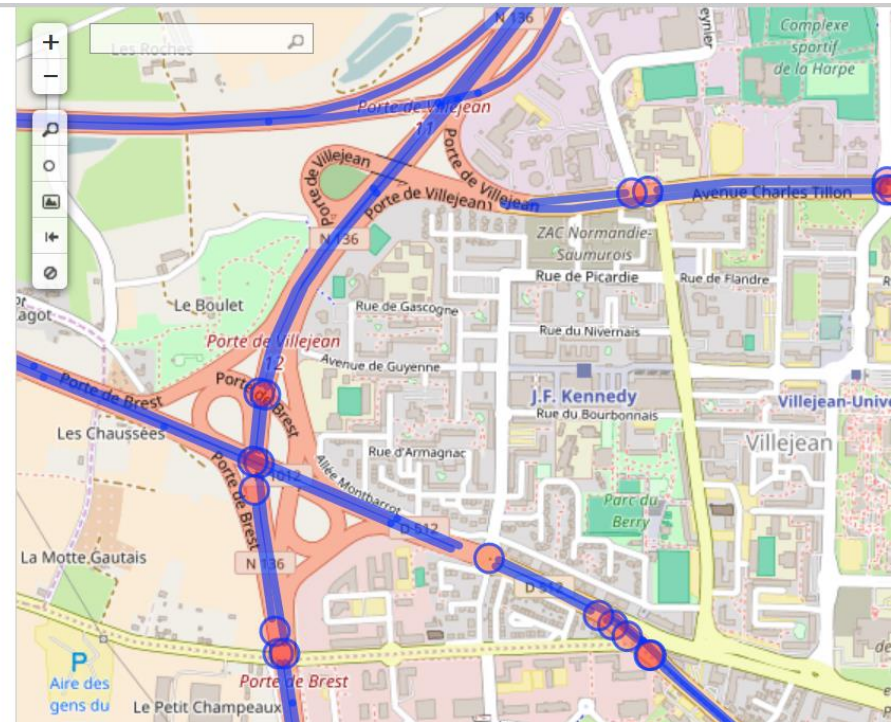
	id	@id
1	node/1842799974	node/1842799974
2	node/1842799971	node/1842799971
3	node/1842799979	node/1842799979
4	node/1842799976	node/1842799976
5	node/1842799965	node/1842799965
6	node/1842799959	node/1842799959
7	node/1842799970	node/1842799970
8	node/1842799967	node/1842799967
9	node/1842800008	node/1842800008
10	node/1842800007	node/1842800007
11	node/1842800012	node/1842800012
12	node/1842800009	node/1842800009
13	node/1842800003	node/1842800003

+ Requêtes classiques

Combiner 2 critères

- Les grandes routes principales (*trunk*) et les routes principales (*primary*)

```
1  /*
2  This has been generated by the overpass-turbo wizard.
3  The original search was:
4  "route"
5  */
6  [out:json][timeout:25];
7  // gather results
8
9  // query part for: "route"
10 way["highway"="trunk"]({{bbox}});
11 way["highway"="primary"]({{bbox}});
12
13 );
14 // print results
15 out body;
16 >;
17 out skel qt;
```



+ Requêtes classiques

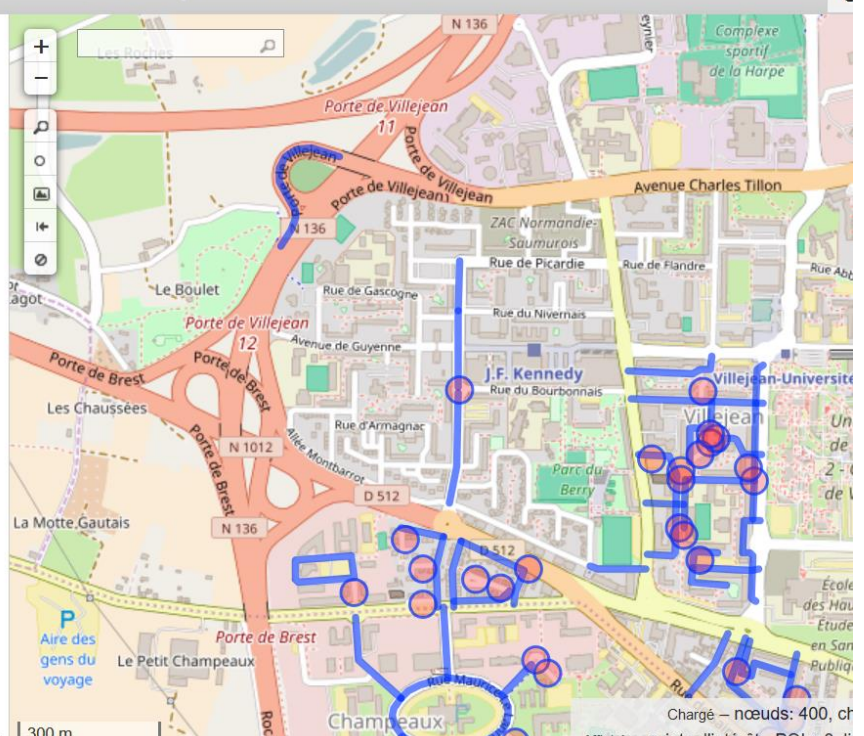
Ajouter un critère de vitesse

- Les routes limitées à 30kmh (*maxspeed*)
- <http://wiki.openstreetmap.org/wiki/Key:maxspeed>

Enjoy OpenStreetMap data? Support OSM's 2016 donation d

overpass turbo

```
1  /*
2  This has been generated by the overpass-turbo wizard.
3  The original search was:
4  "route"
5  */
6  [out:json][timeout:25];
7  // gather results
8
9  // query part for: "route"
10 way["highway"]["maxspeed"="30"]({{bbox}});
11
12 };
13 // print results
14 out body;
15 >;
16 out skel qt;
```



Chargé – nœuds: 400, ch
Affiché – points d'intérêt - POIs: 0, li

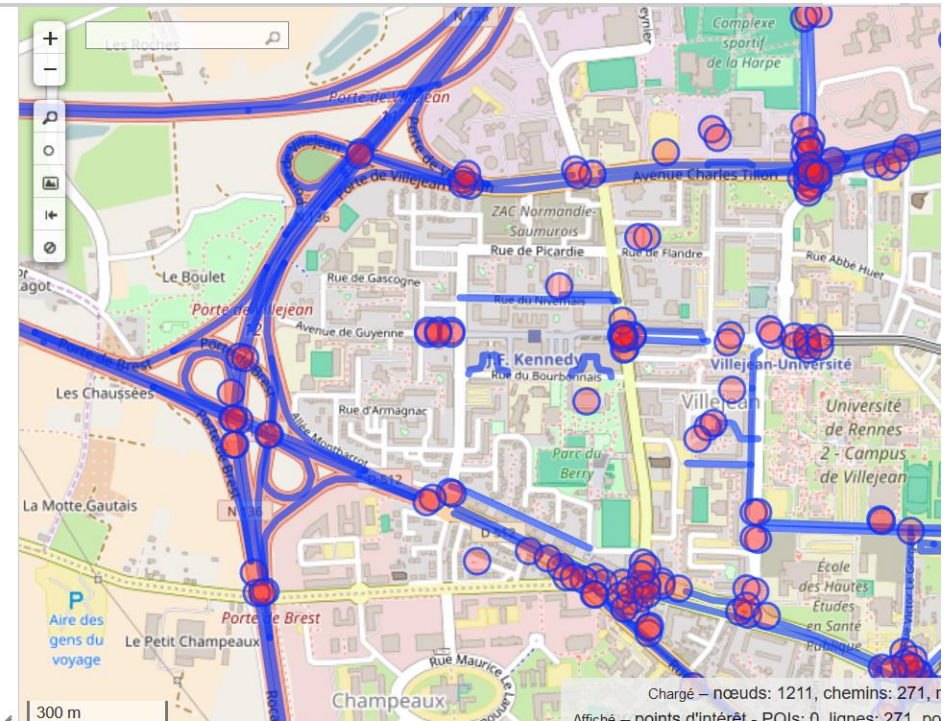
Ajouter un critère de direction

- Les routes en sens unique (*oneway*)
- <http://wiki.openstreetmap.org/wiki/Key:oneway>

```

1  /*
2  This has been generated by the overpass-turbo wizard.
3  The original search was:
4  "route"
5  */
6  [out:json][timeout:25];
7  // gather results
8  {
9    // query part for: "route"
10   way["highway"]["oneway"="yes"] ({{bbox}});
11 }
12 ;
13 // print results
14 out body;
15 >;
16 out skel qt;

```





Modifier l'emprise de la recherche



Deux options :

1. Utiliser l'emprise de la carte (bbox)

```
way ["highway"="trunk"]({{bbox}});
```

2. Utiliser un nom de lieux (ville, région, pays,...)

```
{{geocodeArea:rennes}}->.searchArea;  
  
way ["highway"="trunk"](area.searchArea);
```

+ Requêtes classiques

Choisir la zone d'interrogation (emprise de la requête)

- Récupérer les **routes** à **sens uniques** et **limitées à 30km/h** à **Rennes**

```
1  /*
2  This has been generated by the overpass-turbo wizard.
3  The original search was:
4  "highway=trunk in Rennes"
5  */
6  [out:json][timeout:25];
7  // fetch area "Rennes" to search in
8  {{geocodeArea:Rennes}}->.searchArea;
9  // gather results
10 (
11   // query part for: "highway=trunk"
12   way["highway"]["maxspeed"="30"]["oneway"="yes"]
13   (area.searchArea);
14 );
15 // print results
16 out body;
17 >;
18 out skel qt;
```



```
[out:json][timeout:25];

{{geocodeArea:rennes}}->.searchArea;

way ["highway"] ["oneway"="yes"] ["maxspeed"="30"](area.searchArea);

out body;
>;
out skel qt;
```


+ Requêtes classiques

Choisir la zone d'interrogation (emprise de la requête)

- Récupérer les **routes** à **sens uniques** et **limitées à 30km/h** à **Nantes**

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

```
1 /*
2 This has been generated by the overpass-turbo wizard.
3 The original search was:
4 "route in rennes"
5 */
6 [out:json][timeout:25];
7 // fetch area "rennes" to search in
8 {{geocodeArea:nantes}}->.searchArea;
9 // gather results
10 (
11 // query part for: "route"
12 way ["highway"] ["maxspeed"="30"] ["oneway"="yes"] (area.searchArea);
13 );
14 // print results
15 out body;
16 >;
17 out skel qt;
```

Carte Données

Chargé – nœuds: 3989, chemins: 812, relations: 0
Affiché – points d'intérêt - POIs: 0, lignes: 812, polygones: 0

+ Requête classique

Amenity

■ Documentation

<http://wiki.openstreetmap.org/wiki/Key:amenity>

Key = **highway**

Value =

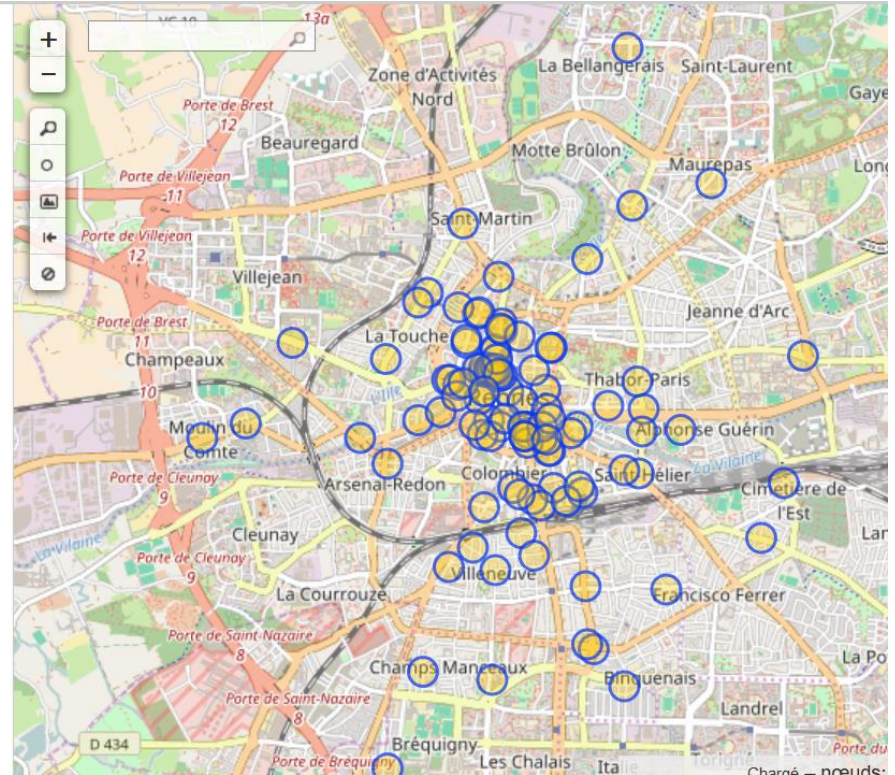
- bar
- recycling
- bench
- school
- ...

Key	Value	Element	Comment	Rendering	Photo
Sustenance					
amenity	bar		Bar is a purpose-built commercial establishment that sells alcoholic drinks to be consumed on the premises. They are characterised by a noisy and vibrant atmosphere, similar to a party and usually don't sell food. See also the description of the tags <code>amenity=pub;bar;restaurant</code> for a distinction between these.		
amenity	bbq		BBQ or Barbecue is a permanently built grill for cooking food, which is most typically used outdoors by the public. For example these may be found in city parks or at beaches. Use the tag <code>fuel=*</code> to specify the source of heating, such as <code>fuel=wood;electric;charcoal</code> . For mapping nearby table and chairs, see also the tag <code>tourism=picnic_site</code> . For mapping campfires and firepits, instead use the tag <code>leisure=firepit</code> .		
amenity	biergarten		Biergarten or beer garden is an open-air area where alcoholic beverages along with food is prepared and served. See also the description of the tags <code>amenity=pub;bar;restaurant</code> . A biergarten can commonly be found attached to a beer hall, pub, bar, or restaurant. In this case, you can use <code>biergarten=yes</code> additional to <code>amenity=pub;bar;restaurant</code> .		
amenity	cafe		Cafe is generally an informal place that offers casual meals and beverages; typically, the focus is on coffee or tea. Also known as a <code>coffeehouse/shop</code> , <code>bistro</code> or <code>sidewalk cafe</code> . The kind of food served may be mapped with the tags <code>cuisine=*</code> and <code>diet=*</code> . See also the tags <code>amenity=restaurant;bar;fast_food</code> .		
amenity	drinking_water		Drinking water is a place where humans can obtain potable water for consumption. Typically, the water is used for only drinking. Also known as a drinking fountain or bubbler.		

+ Requêtes classiques

→ Extraire les bars

```
1  /*
2  This has been generated by the overpass-turbo wizard.
3  The original search was:
4  "Bar"
5  */
6  [out:json][timeout:25];
7  // gather results
8  (
9    // query part for: "Bar"
10   node["amenity"="bar"] ({bbox});
11 );
12 // print results
13 out body;
14 >;
15 out skel qt;
```



+ Requêtes classiques

Extraire les bornes incendies

Exécuter

Partager

Exporter

Assistant

Enregistrer

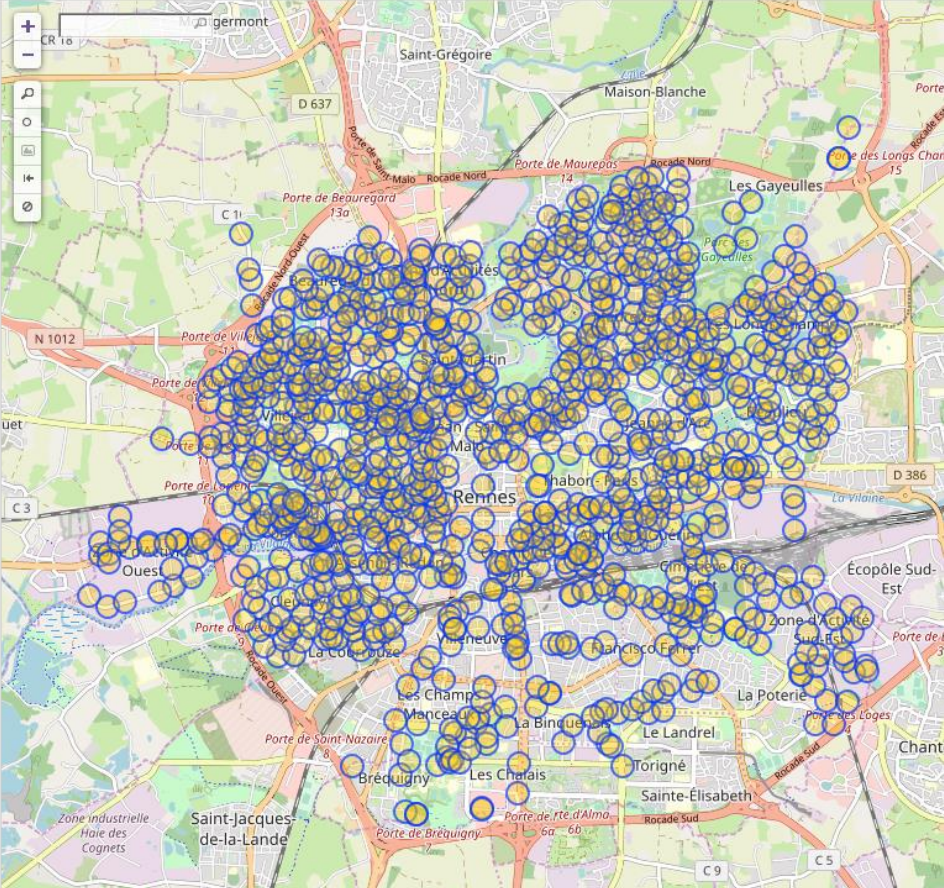
Charger

Paramètres

Aide

overpass turbo

```
1  /*
2  This has been generated by the overpass-turbo wizard.
3  The original search was:
4  "bar in rennes"
5  */
6  [out:json][timeout:25];
7  // fetch area "rennes" to search in
8  ((geocodeArea:rennes))->.searchArea;
9  // gather results
10 {
11   // query part for: "bar"
12   node["emergency"="fire_hydrant"] (.searchArea);
13 }
14 };
15 // print results
16 out body;
17 >;
18 out skel qt;
```



+ Requêtes classiques

Extraire plusieurs type d'objets

Exécuter

Partager

Exporter

Assistant

Enregistrer

Charger

Paramètres

Aide

overpass turbo

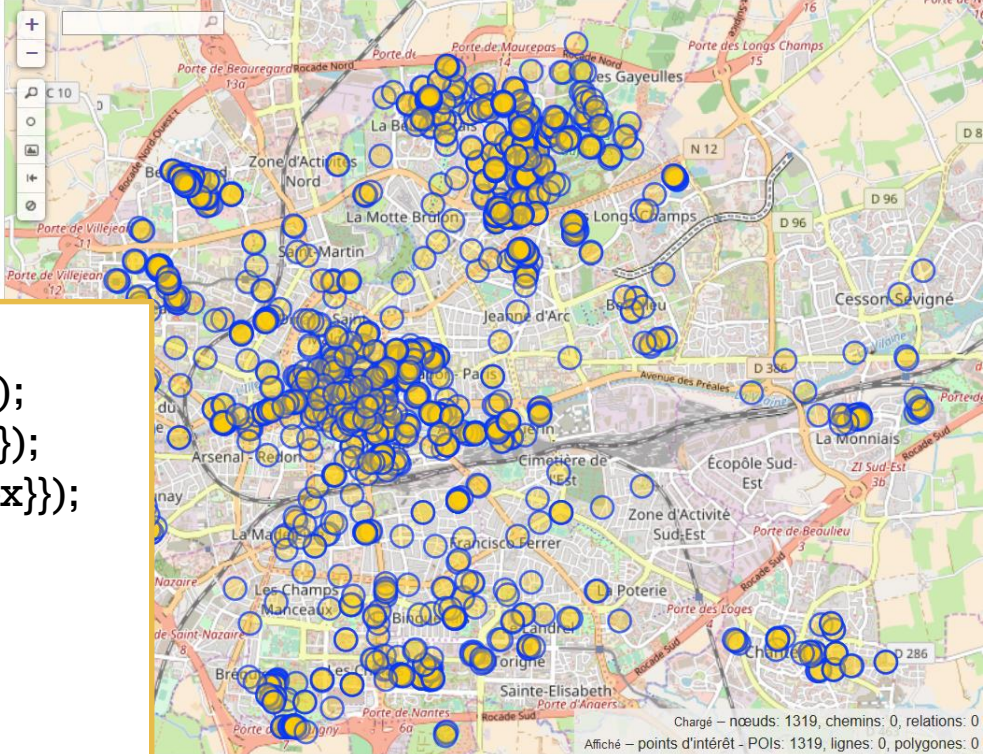
Carte

Données

```
1 (
2   node["amenity"="bar"]({{bbox}});
3   node["amenity"="cafe"]({{bbox}});
4   node["amenity"="bench"]({{bbox}});
5   );
6
7 out body;
8 >;
9 out skel qt;
10
```

(
 node["amenity"="bar"]({{bbox}});
 node["amenity"="cafe"]({{bbox}});
 node["amenity"="bench"]({{bbox}});
);

out body;
>;
out skel qt;



<https://overpass-turbo.eu/#c>

<https://overpass-turbo.eu/s/zaC>

+ Requêtes classiques

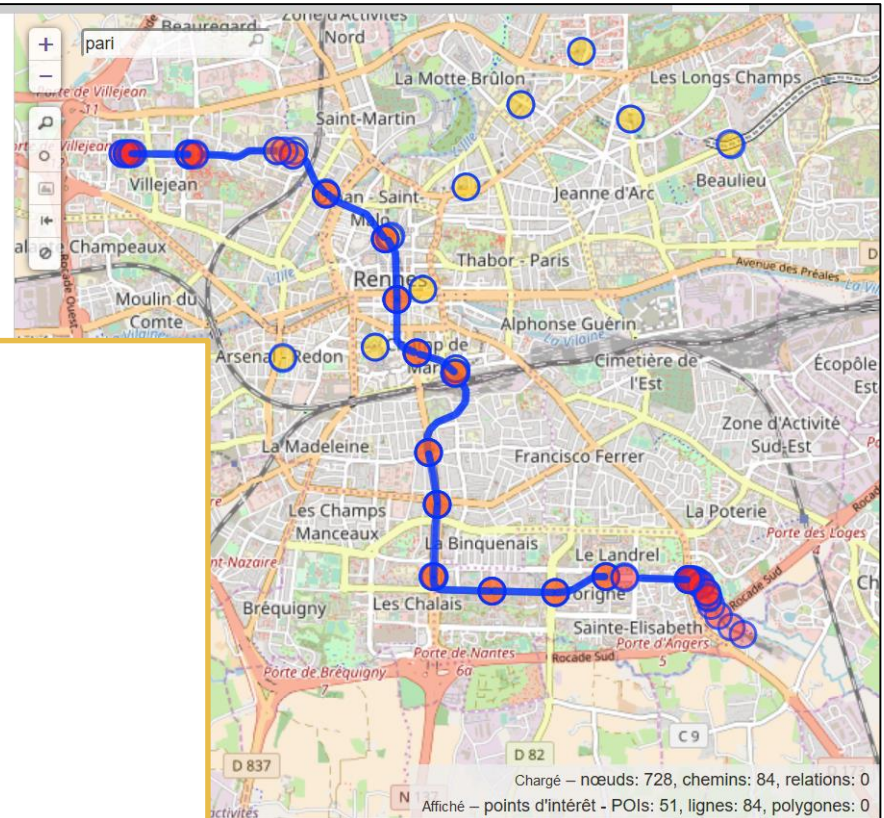
Extraire plusieurs type d'objets

```
1 [out:json][timeout:25];
2 {{geocodeArea:rennes}}->.searchArea;
3
4 (node["public_transport"="stop_position"] ["subway"="yes"](area.searchArea);
5 way["railway"="subway"](area.searchArea);
6 );
7
8 out body;
9 >;
10 out skel qt;
```

```
[out:json][timeout:25];
{{geocodeArea:rennes}}->.searchArea;

(node["public_transport"="stop_position"]
["subway"="yes"](area.searchArea);
way["railway"="subway"](area.searchArea);
);

out body;
>;
out skel qt;
```

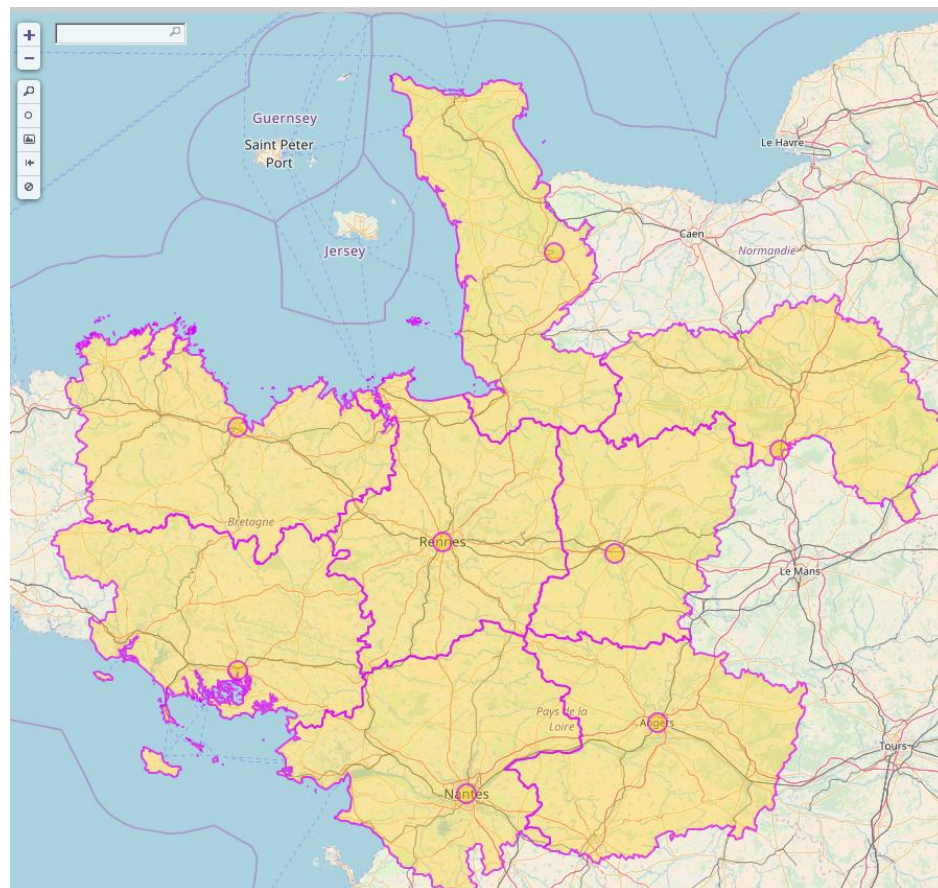


trial: When metropolitan areas don't match with an administrative subdivision (someti

+ Requêtes classiques

Limites administratives

- Régions = 4
- Départements = 6
- Arrondissements = 7
- Communes = 8
- Quartiers = 9



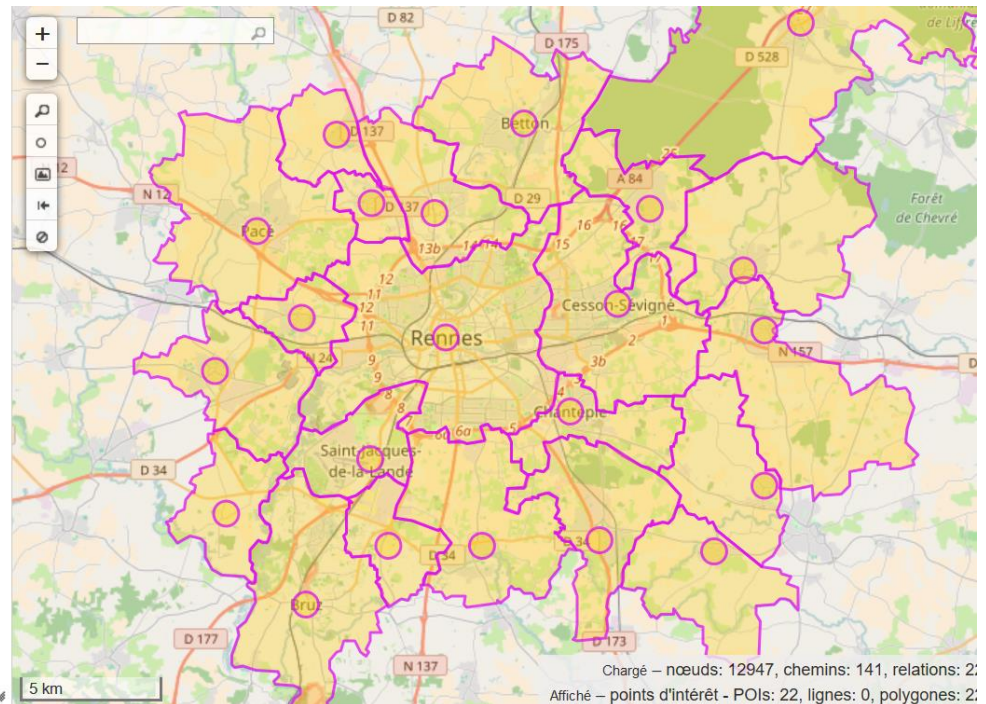
+ Requêtes classiques

Extraire les communes adjacentes à Rennes

- Niveau communal en France ("8 ")
- <http://wiki.openstreetmap.org/wiki/Tag:boundary%3Dadministrative>

```
1  /*
2  This has been generated by the overpass-turbo wizard.
3  The original search was:
4  "Bar"
5  */
6  [out:json][timeout:25];
7  // gather results
8  (
9    // query part for: "Bar"
10    relation["boundary"="administrative"] ["admin_level"="8"]
11    {{{bbox}}};
12  );
13  // print results
14  out body;
15  >;
16  out skel qt;
```

<https://overpass-turbo.eu/s/z8w>

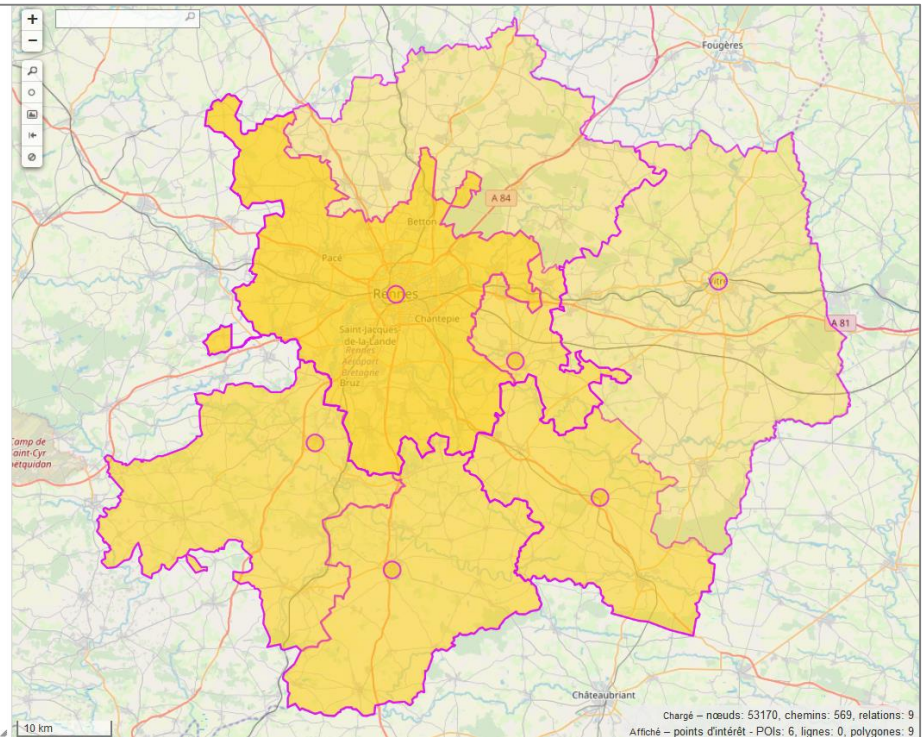


+ Requêtes classiques

Obtenir les contours des EPCI

Tag : boundary=local_authority

```
[out:json][timeout:25];  
  
relation ["boundary"="local_authority"](((bbox)));  
  
// print results  
out body;  
>;  
out skel qt;
```



+ Requêtes classiques

Landuse

■ Documentation

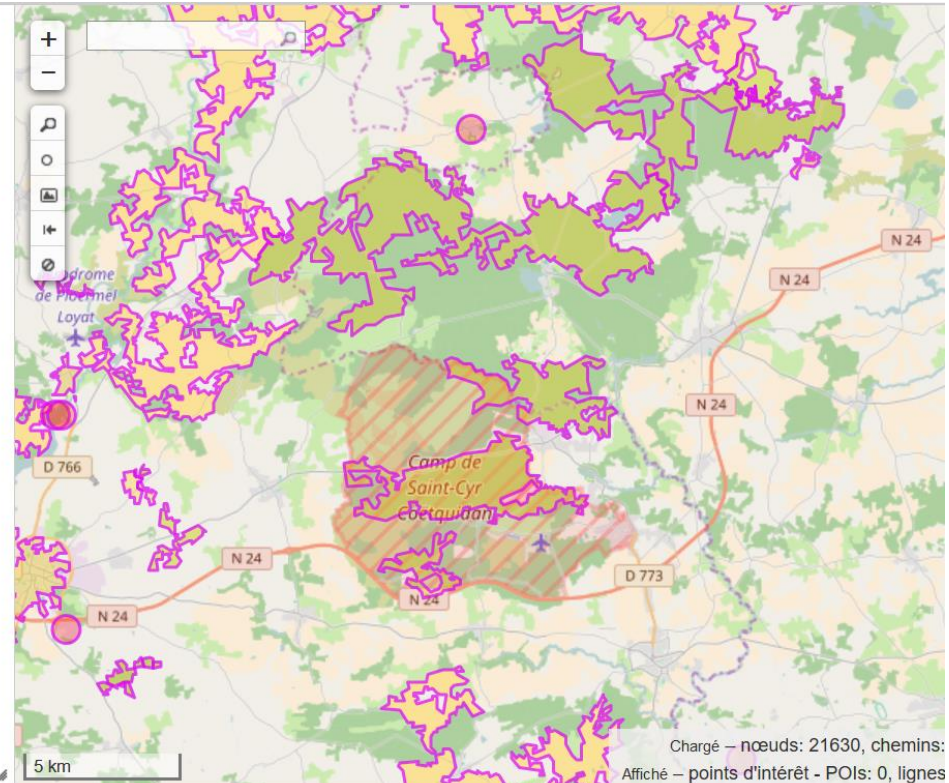
<http://wiki.openstreetmap.org/wiki/FR:Key:landuse>

landuse	basin		Zone d'eau artificielle de plusieurs types (infiltration, détention, rétention) qui finit par s'écouler dans une rivière. Utiliser avec <code>basin=*</code> pour les différents types.		
landuse	brownfield		Zone où des anciens bâtiments ont été rasés. La construction de nouveaux bâtiments est planifiée, mais pas encore en cours.		
landuse	cemetery		Cimetière. ajoutez <code>religion=*</code> s'il y a lieu (voir liste dans <code>amenity=place of worship</code>). Utiliser <code>amenity=grave_yard</code> pour les petites surfaces (à proximité d'une église par exemple)		
					

+ Requêtes classiques

Extraire les zones renseignées sur l'occupation des sols

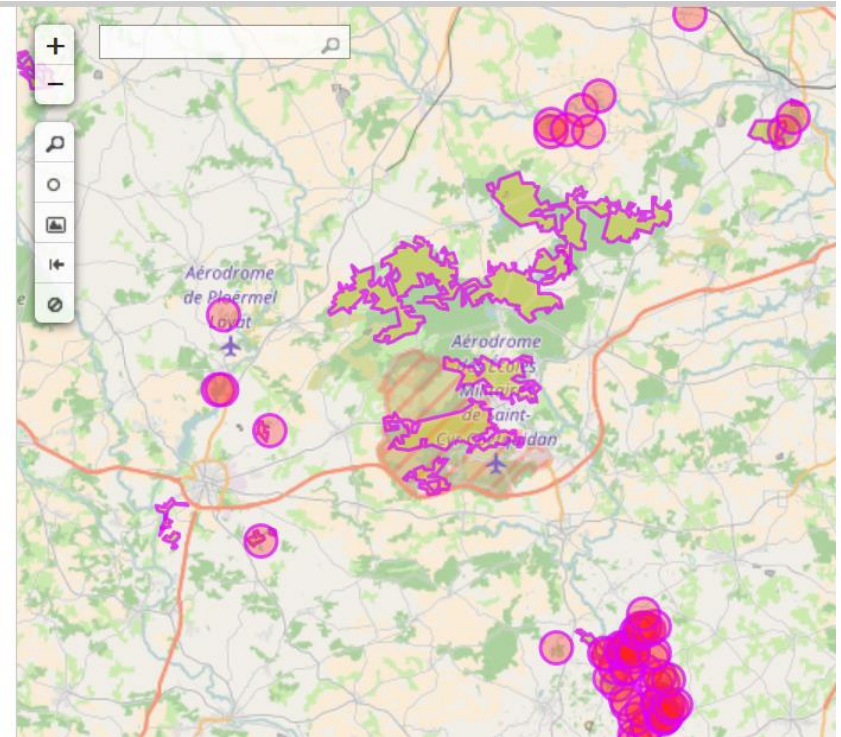
```
1  /*
2  This has been generated by the overpass-turbo wizard.
3  The original search was:
4  "Bar"
5  */
6  [out:json][timeout:25];
7  // gather results
8  (
9    // query part for: "Bar"
10   relation["landuse"]({{bbox}});
11 );
12 // print results
13 out body;
14 >;
15 out skel qt;
```



+ Requêtes classiques

Extraire les zones renseignées comme forêt

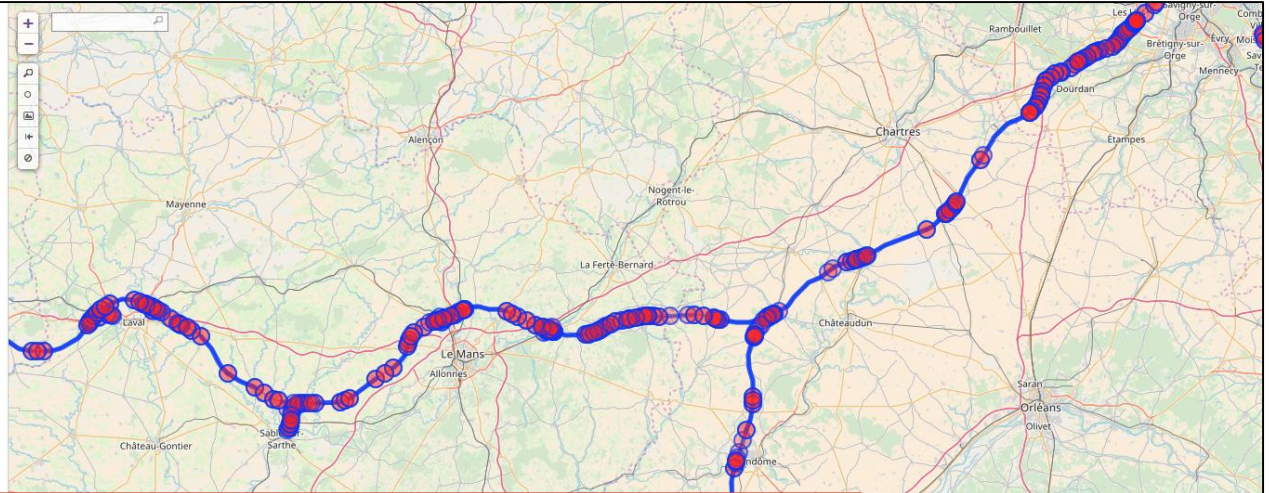
```
/*  
This has been generated by the overpass-turbo wizard.  
The original search was:  
"Bar"  
*/  
[out:json][timeout:25];  
// gather results  
(  
  // query part for: "Bar"  
  relation["landuse"="forest"]({{bbox}});  
);  
// print results  
out body;  
>;  
out skel qt;
```



+ Requêtes classiques

Sélectionner les lignes grandes vitesses

```
1 [out:json][timeout:25];  
2 way["railway"="rail"]["highspeed"="yes"](bbox);  
3  
4 // print results  
5 out body;  
6 >;  
7  
8 out skel qt;
```



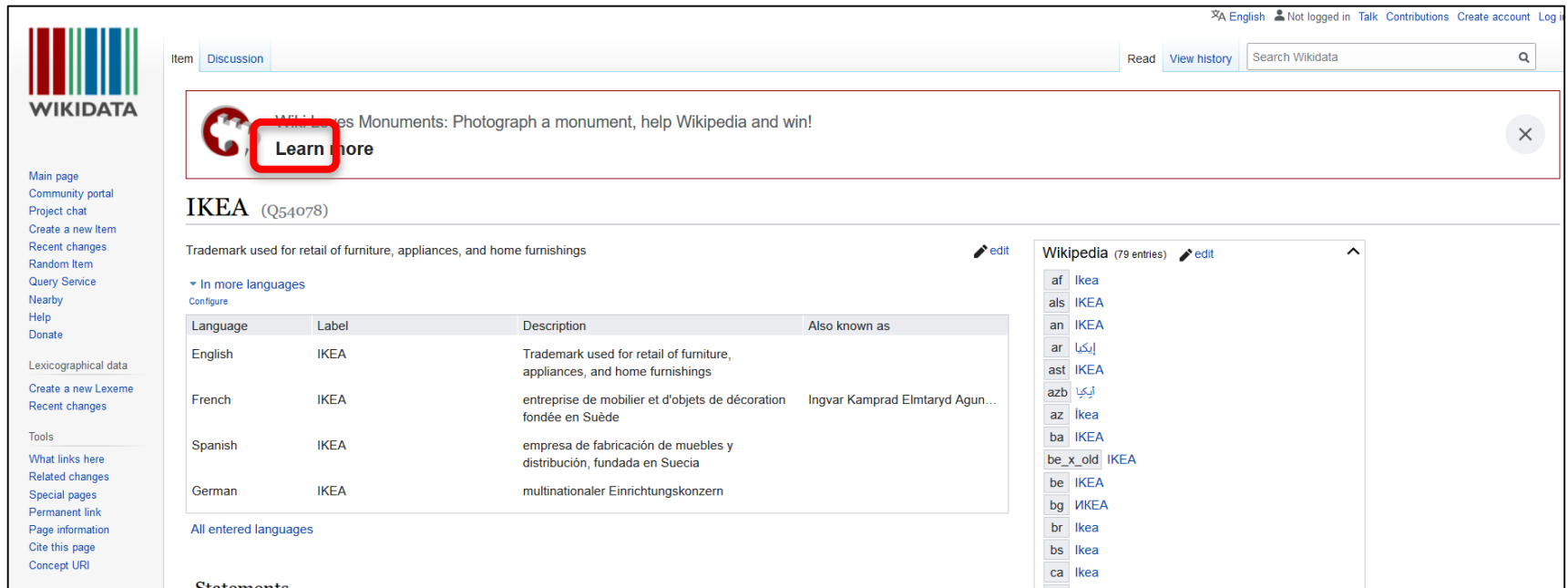
```
[out:json][timeout:25];  
way["railway"="rail"]["highspeed"="yes"](bbox);  
out body;  
>;  
out skel qt;
```

Chargé – nœuds: 6495, chemins: 656, relations: 0
Affiché – points d'intérêt - POIs: 0, lignes: 656, polygones: 0

+ Requêtes classiques

Mobiliser les ID de Wikidata

<https://fr.wikipedia.org/wiki/Wikidata>



Wikidata

Item Discussion Read View history Search Wikidata

Wikidata Monuments: Photograph a monument, help Wikipedia and win! [Learn more](#)

IKEA (Q54078)

Trademark used for retail of furniture, appliances, and home furnishings

[In more languages](#)

Language	Label	Description	Also known as
English	IKEA	Trademark used for retail of furniture, appliances, and home furnishings	
French	IKEA	entreprise de mobilier et d'objets de décoration fondée en Suède	Ingvar Kamprad Elmtaryd Agun...
Spanish	IKEA	empresa de fabricación de muebles y distribución, fundada en Suecia	
German	IKEA	multinationaler Einrichtungskonzern	

All entered languages

Statements

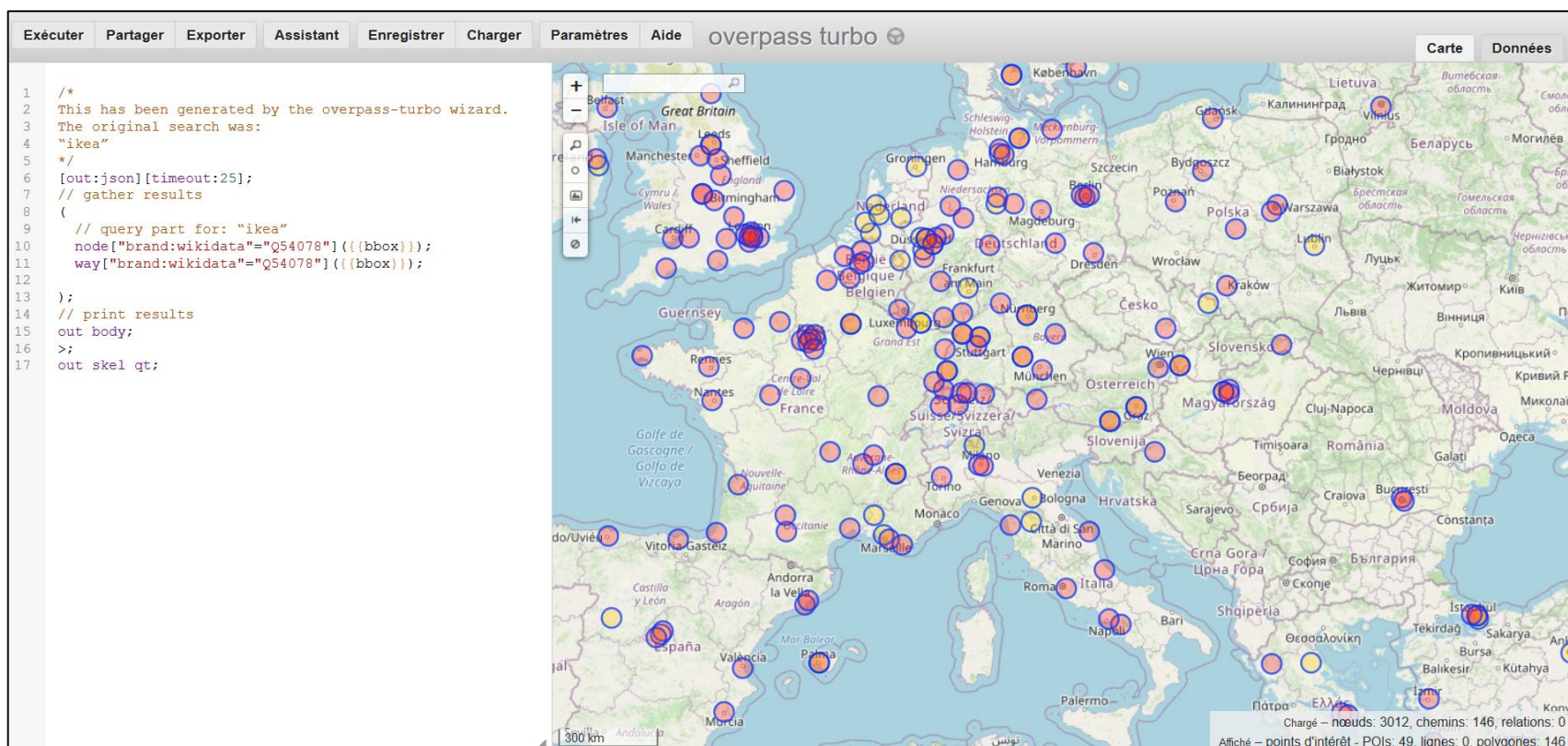
Wikipedia (79 entries)

af	Ikea
als	IKEA
an	IKEA
ar	إيكيا
ast	IKEA
azb	ایکیا
az	Ikea
ba	IKEA
be_x_old	IKEA
be	IKEA
bg	ИКЕА
br	Ikea
bs	Ikea
ca	Ikea

<https://www.wikidata.org/wiki/Q54078>

+ Requêtes classiques

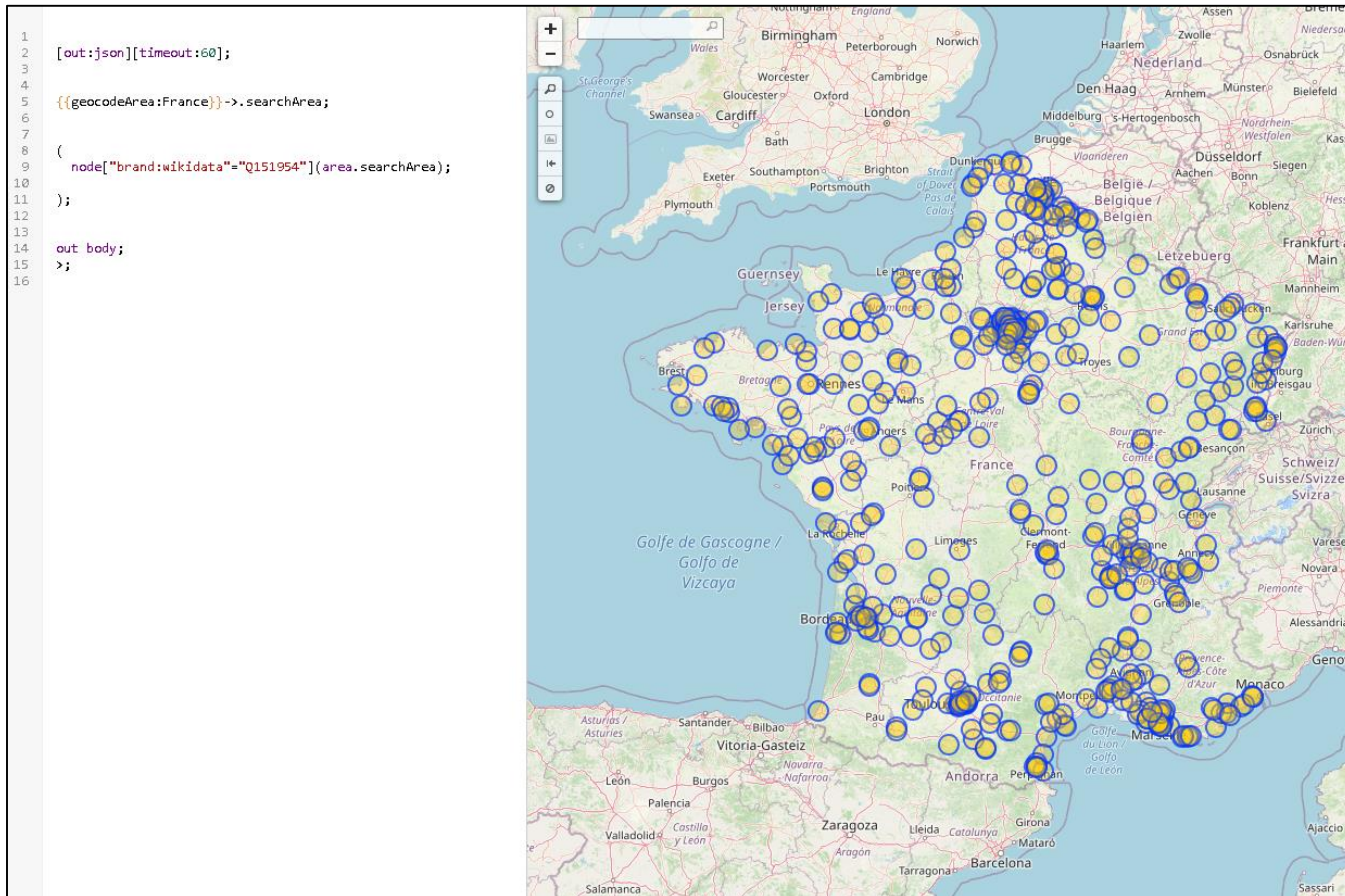
Mobiliser les ID de Wikidata > magasins IKEA



<https://t.co/kmZ8am31j6?amp=1>

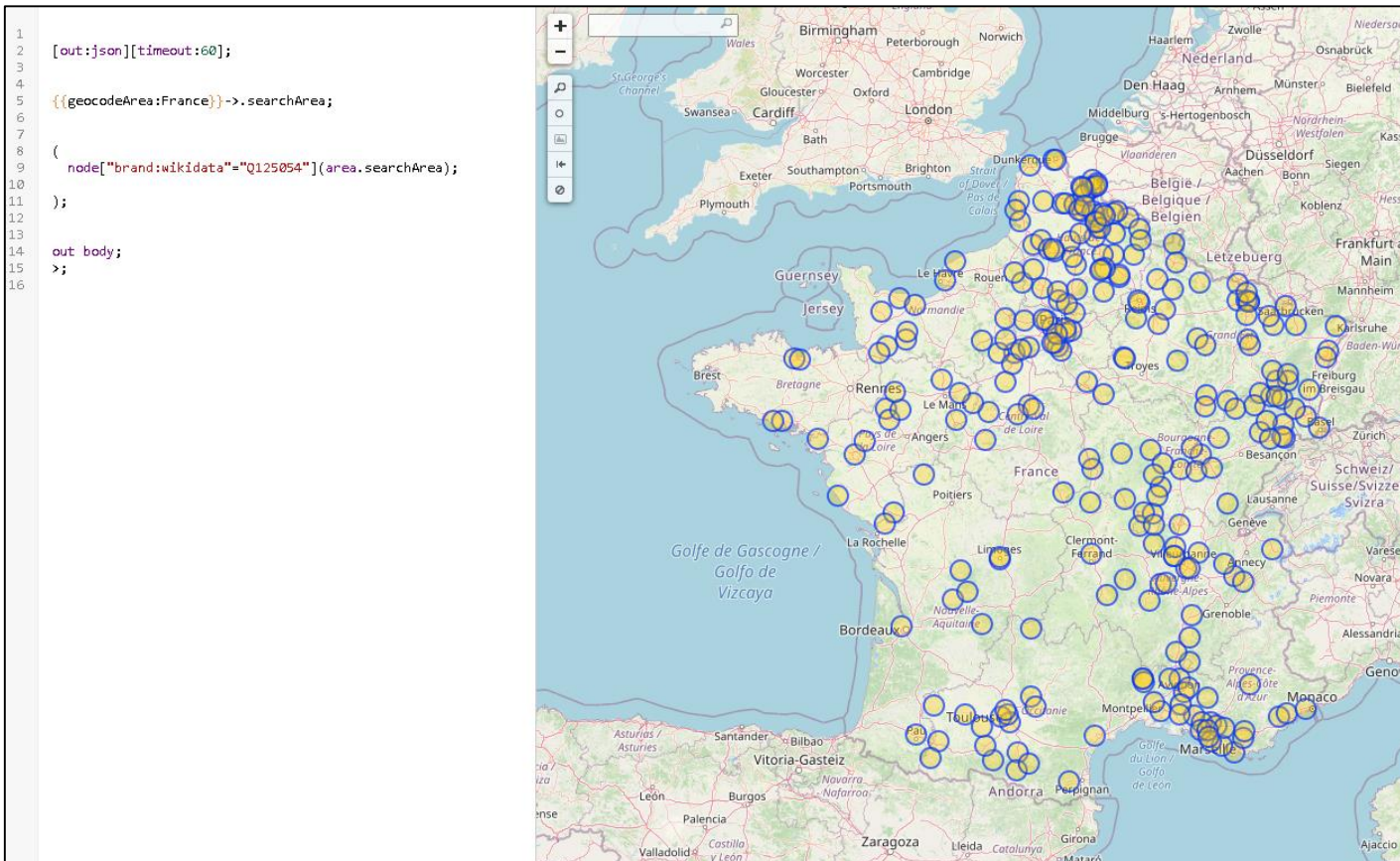
+ Requêtes classiques

Tester les magasins LIDL en France



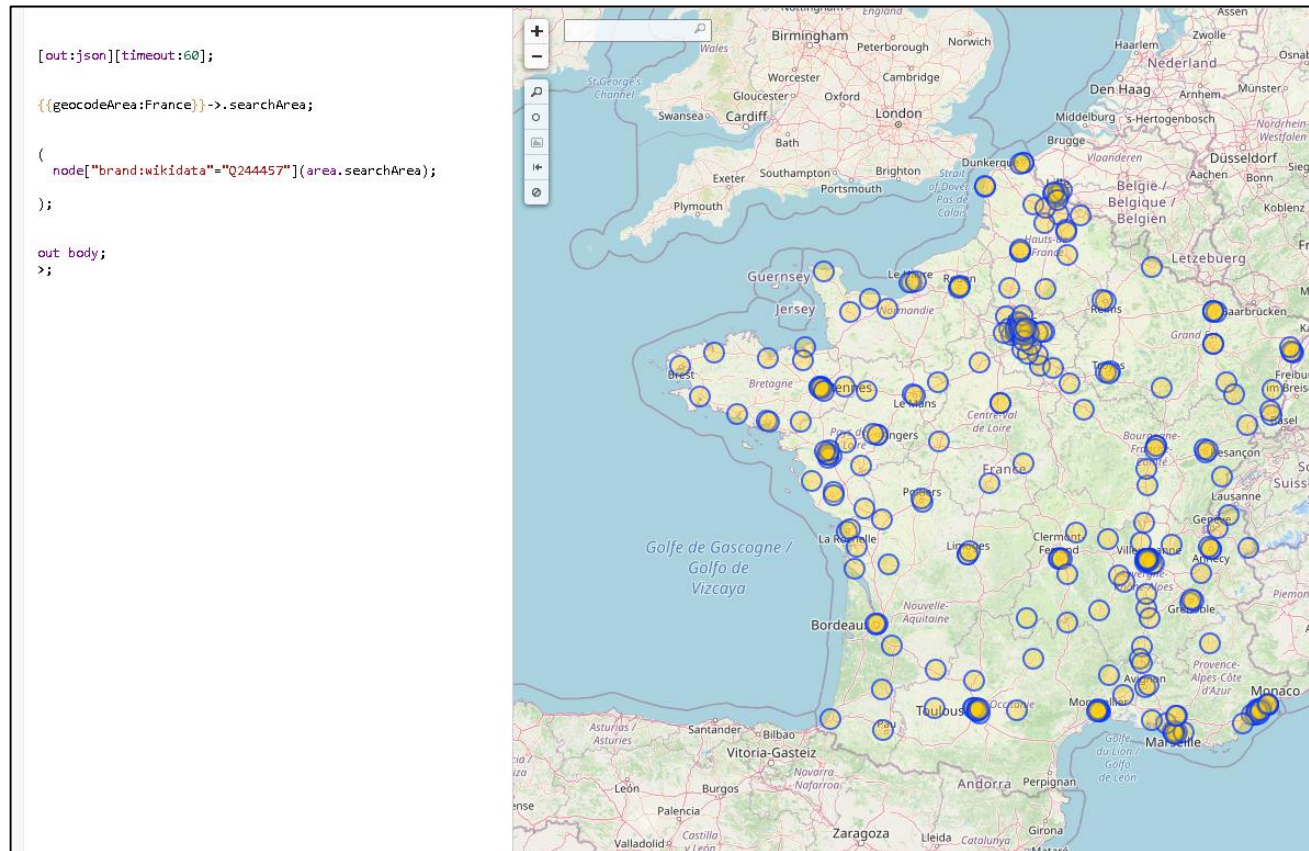
+ Requêtes classiques

Les magasins ALDI



+ Requêtes classiques

Les restaurant Subway en France



+

Requêtes complexes

+ Requêtes complexes

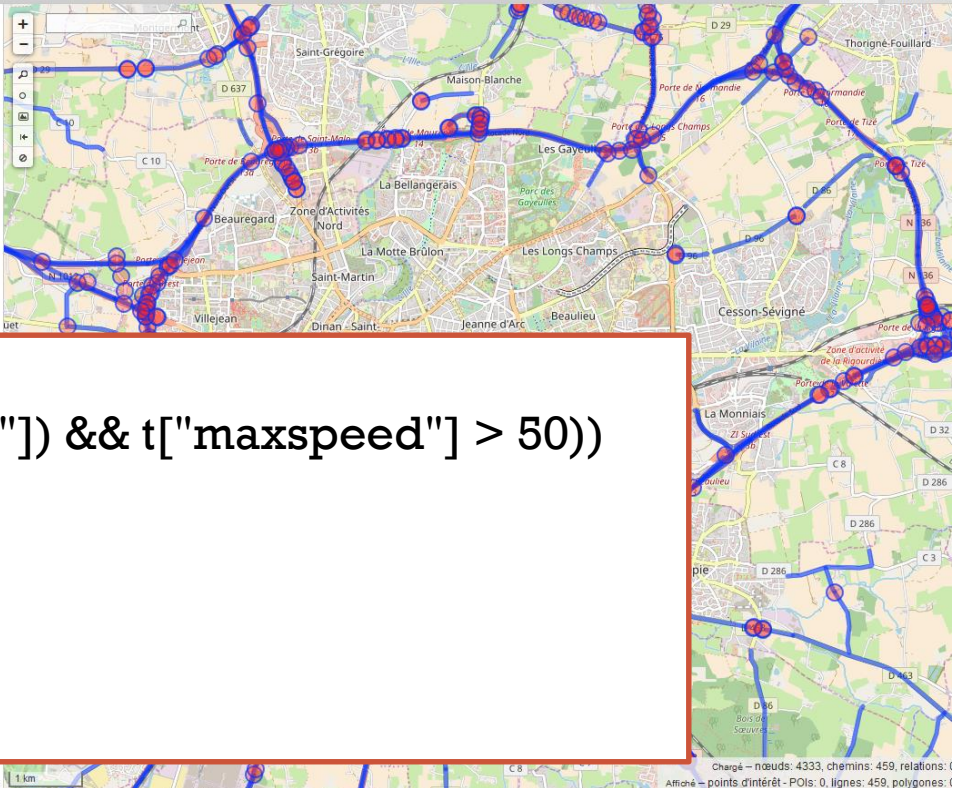
Critère minimum ou maximum

- Toutes les routes avec une vitesse limite de plus de 50km/h

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

```
1 [out:json][timeout:25];
2 // gather results
3
4 (way[highway]
5   (if: (is_number(t[maxspeed]) && t[maxspeed] > 50))
6   ({{bbox}}));
7 );
8 // print results
9 out body;
10 >;
11 out skel qt;
```

Carte Données



(way["highway"]
 (if: (is_number(t["maxspeed"])) && t["maxspeed"] > 50))
 ({{bbox}}));
);
out body;
>;
out skel qt;

Chargé - nœuds: 4333, chemins: 459, relations: 1
Affiché - points d'intérêt - POIs: 0, lignes: 459, polygones: 1

<https://overpass-turbo.eu/index.html>

+ Requêtes complexes

Ajouter un critère de nombre de voies

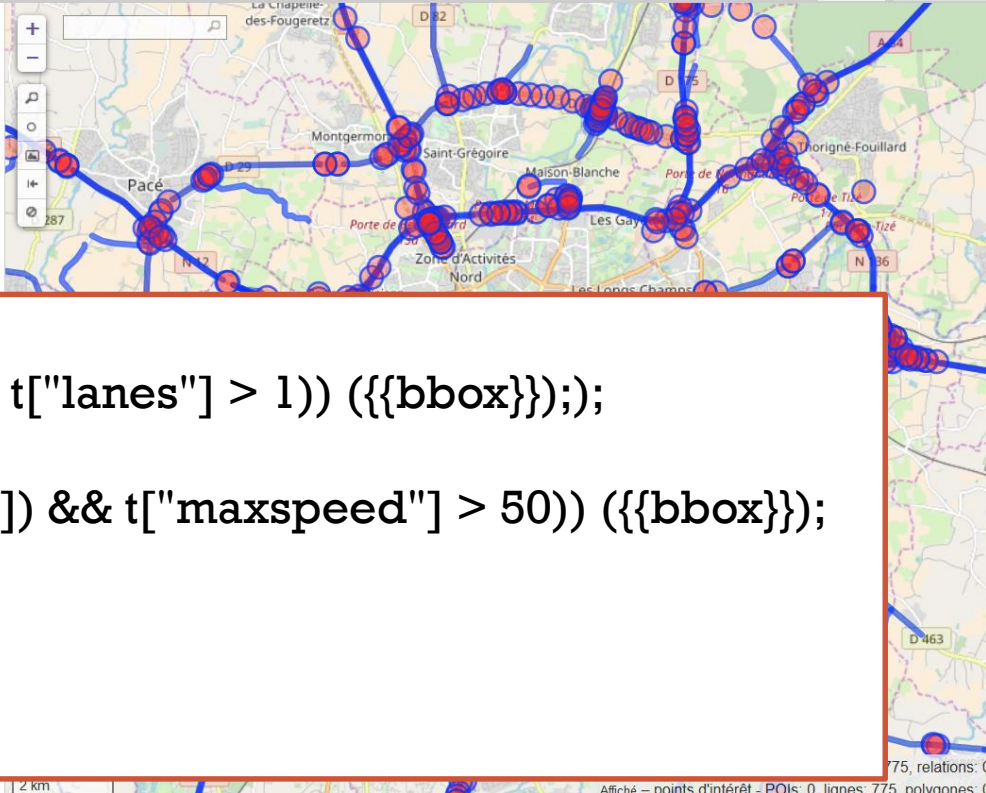
- Toutes les routes avec une vitesse maximale de 50km/h et avec au moins deux voies (*lanes*)

ExécuterPartagerExporterAssistantEnregistrerChargerParamètresAideoverpass turbo

12345678910

(way["highway"]
(if: (is_number(t["lanes"])) && t["lanes"] > 1)) ({{bbox}});
);
(way["highway"]
(if: (is_number(t["maxspeed"]) && t["maxspeed"] > 50)) ({{bbox}});
);
out body;
>;
out skel qt;

CarteDonnées



(way["highway"]
(if: (is_number(t["lanes"])) && t["lanes"] > 1)) ({{bbox}});;
(way["highway"]
(if: (is_number(t["maxspeed"]) && t["maxspeed"] > 50)) ({{bbox}});
);
out body;
>;
out skel qt;

775, relations: 0
Affiché — points d'intérêt — POIs: 0. lignes: 775. polyaones: 0

+ Requêtes complexes

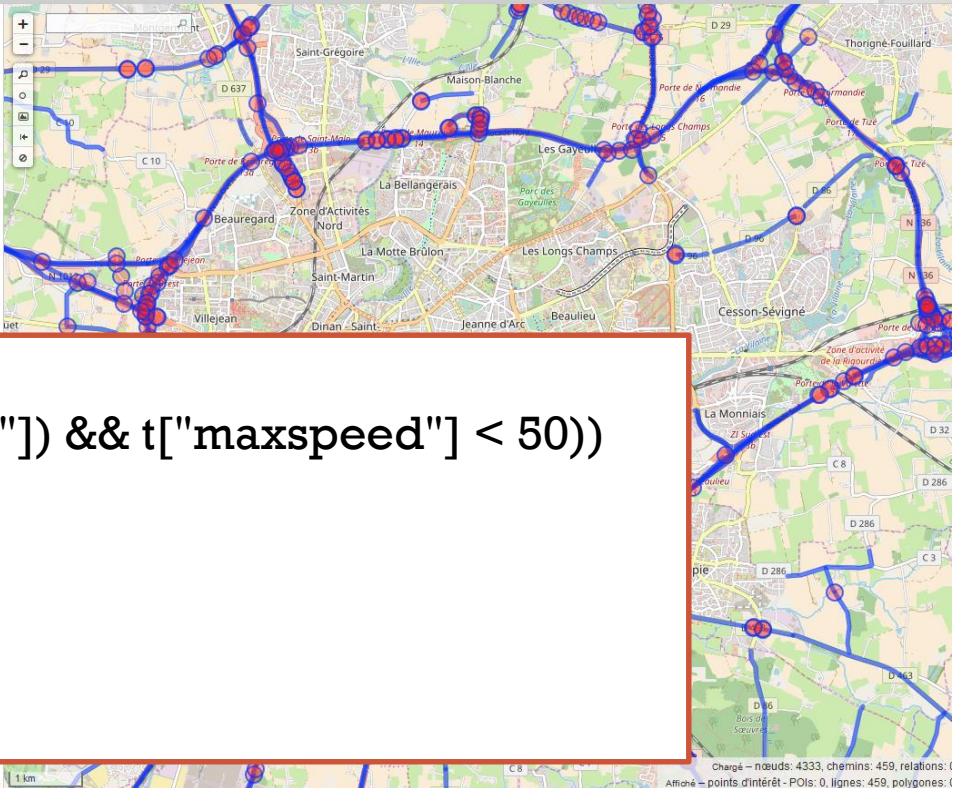
Critère minimum ou maximum

- Toutes les routes avec une vitesse inférieure à 50km/h

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

```
1 [out:json][timeout:25];
2 // gather results
3
4 (way[highway]
5   (if: (is_number(t[maxspeed])) && t[maxspeed] > 50))
6   ({bbox}));
7
8 // print results
9 out body;
10 >;
11 out skel qt;
```

Carte Données



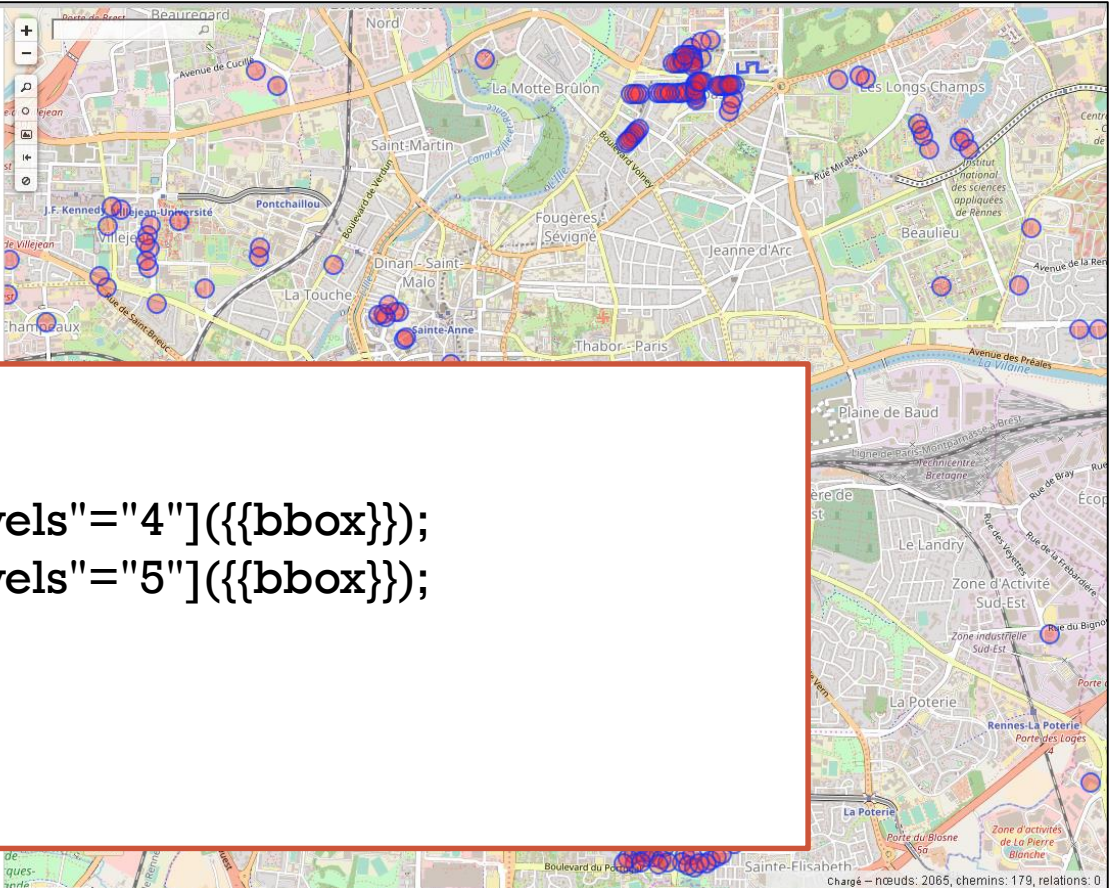
Chargé - nœuds: 4333, chemins: 459, relations: 1
Affiché - points d'intérêt - POIs: 0, lignes: 459, polygones: 1

(way["highway"]
 (if: (is_number(t["maxspeed"])) && t["maxspeed"] < 50))
 ({bbox}));
);
out body;
>;
out skel qt;

+ Requêtes complexes

Sélectionner les bâtiments de 4 et 5 étages

```
[out:json][timeout:25];
(
  way["building"]["building:levels"="4"]({{bbox}});
  way["building"]["building:levels"="5"]({{bbox}});
);
out body;
>;
out skel qt;
```



The image displays a map of Rennes, France, with numerous buildings highlighted by blue and red circles. These circles represent the results of a query for buildings with 4 or 5 floors. The map shows various urban areas, including the center of Rennes, the university area, and the industrial zone. The query is shown in a code block on the left, and the map is on the right. A red box highlights the query code.

Chargé - nœuds: 2065, chemins: 179, relations: 0

+ Comptage d'entités



Afficher des statistiques sur les bâtiments de Rennes

```
[out:csv(::count, ::"count:nodes", ::"count:ways",
::"count:relations")][timeout:25];
{{geocodeArea:Rennes}}->.searchArea;
(
  node["building"="yes"](area.searchArea);
  way["building"="yes"](area.searchArea);
  relation["building"="yes"](area.searchArea);
);
out count;
```

1	@count	@count:nodes	@count:ways	@count:relations
2	37821	6 37626	189	
3				



Afficher des statistiques sur les bars de Rennes

<pre>1 [out:csv(:,:,count, ::"count:nodes")][timeout:25]; 2 {(geocodeArea:Rennes)}->.searchArea; 3 { 4 node["amenity"="bar"] (area.searchArea); 5 }; 6 out count; 7</pre>	<pre>1 @count @count:nodes 2 112 112 3</pre>
--	--

+ Comptage d'entités

Compter le nombre de pharmacies par communes dans le 35

```
[out:csv(_row,false)];

// Toutes les communes du département du 35
area["ref:INSEE"~"^35"];

// Compte des pharmacies par communes
foreach->.regio(
  // Collect all Nodes, Ways and Relations with amenity=pharmacy in the current
  area
  ( node(area.regio)[amenity=pharmacy];
    way(area.regio)[amenity=pharmacy];
    rel(area.regio)[amenity=pharmacy];) -> .r;

  make out _row = "|-"; out;
  make out _row = "| " + regio.set(t["de:regionalschlüssel"]) +
    " || " + regio.set(t["name"]) +
    " || " + (r.count(nodes) + r.count(ways) + r.count(relations));
  out;
);
make out _row = "|}"; out;
```

```
1  |-
2  |  || Ille-et-Vilaine || 253
3  |-
4  |  || Redon || 25
5  |-
6  |  || Rennes || 152
7  |-
8  |  || Bain-de-Bretagne || 8
9  |-
10 |  || Guichen || 7
11 |-
12 |  || Montfort-sur-Meu || 8
13 |-
14 |  || Pipriac || 3
15 |-
16 |  || Plélan-le-Grand || 4
17 |-
18 |  || Redon || 6
19 |-
20 |  || Saint-Aubin-d'Aubigné || 6
21 |-
22 |  || Sixt-sur-Aff || 1
23 |-
24 |  || Bains-sur-Oust || 1
25 |-
26 |  || Redon || 4
27 |-
28 |  || Bruc-sur-Aff || 0
29 |-
30 |  || Pipriac || 2
31 |-
32 |  || Saint-Ganton || 0
33 |-
34 |  || Saint-Just || 0
35 |-
36 |  || Guipry || 0
37 |-
38 |  || Renac || 0
39 |-
40 |  || Sainte-Marie || 1
41 |-
42 |  || La Chapelle-de-Brain || 0
43 |-
44 |  || Langon || 0
45 |-
46 |  || Guipry-Messac || 2
47 |-
```

+ Contrôle données

Éléments sans tags

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

```
1 way({{bbox}})(if:count_tags() == 0);  
2 out geom;  
3
```

Way 441800913

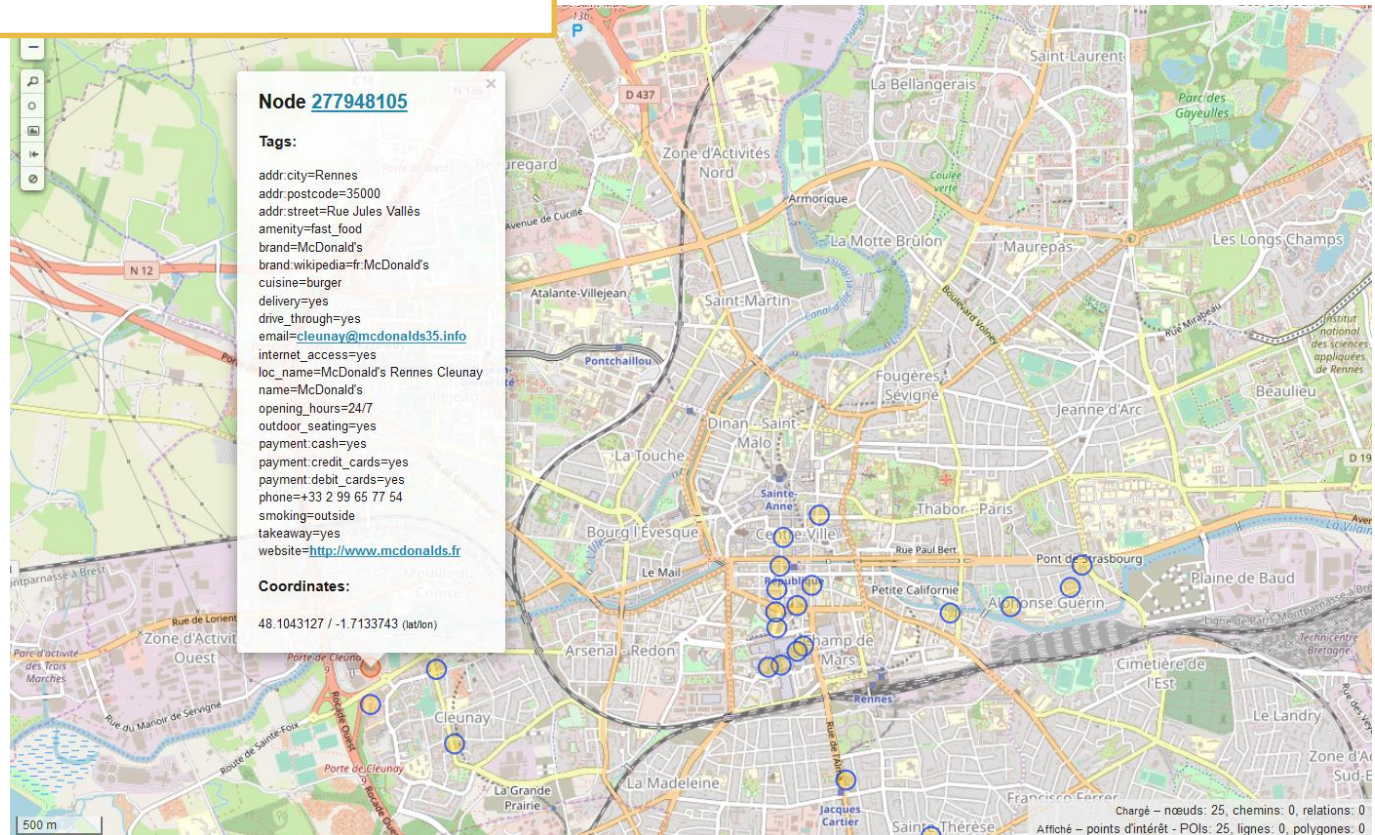
way({{bbox}})(if:count_tags() == 0);
out geom;

Chargé – nœuds: 0, chemins: 14, relations: 0
Affiché – points d'intérêt - POIs: 0, lignes: 14, polygones: 0

+ Contrôle données

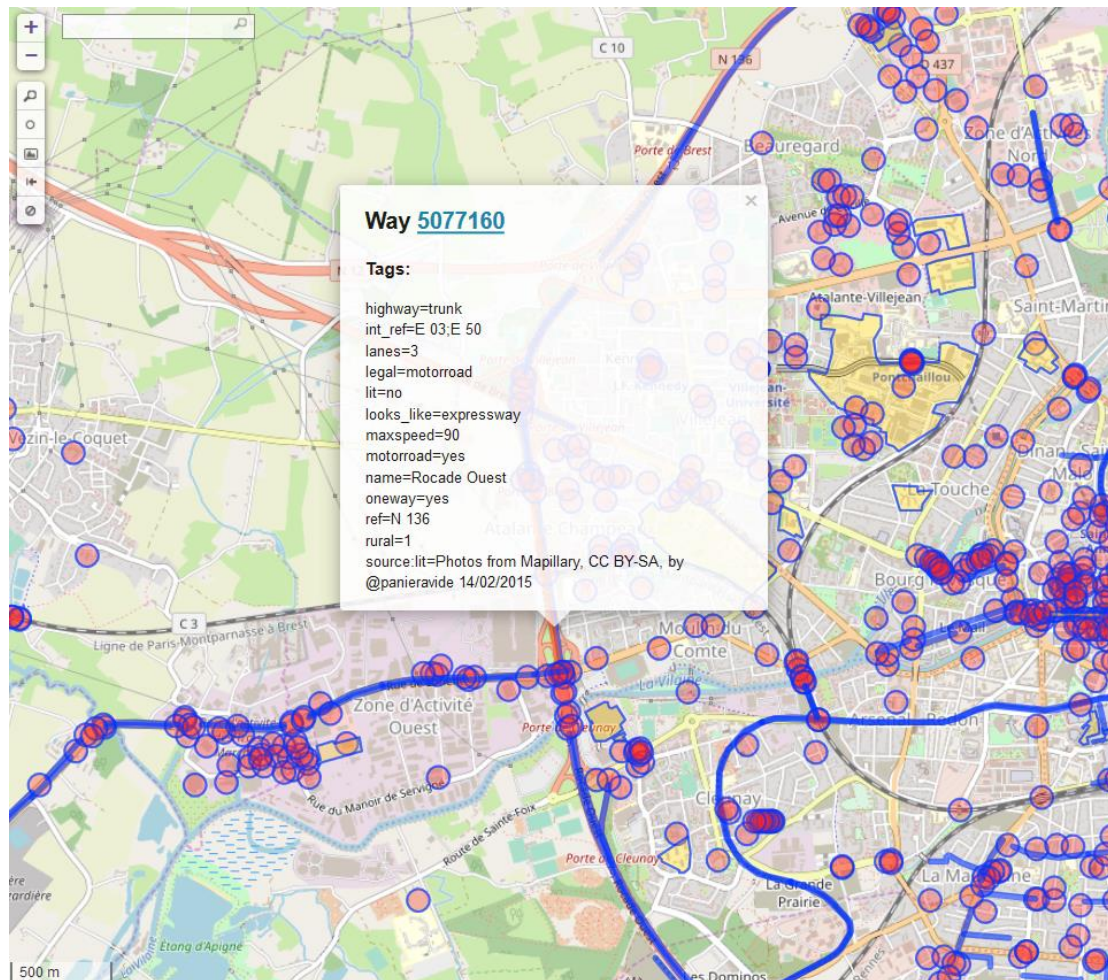
Objets de type nœud avec plus de 20 tags

```
node({{bbox}})(if:count_tags() > 20);  
out geom;
```



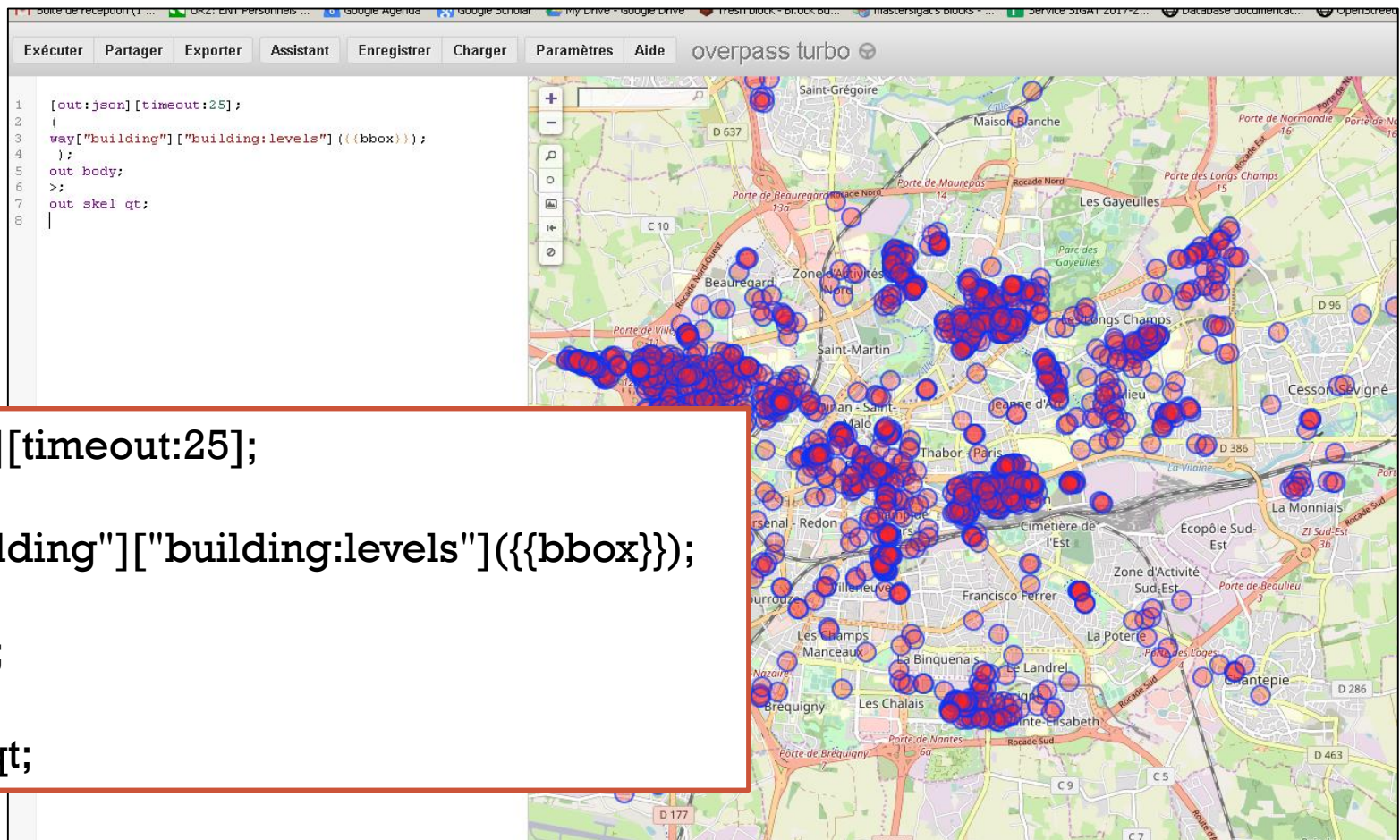
+ Contrôle données

Objets de type relation avec plus de 10 tags



+ Contrôle données

Montrer tous les bâtiments dont la hauteur est renseignée

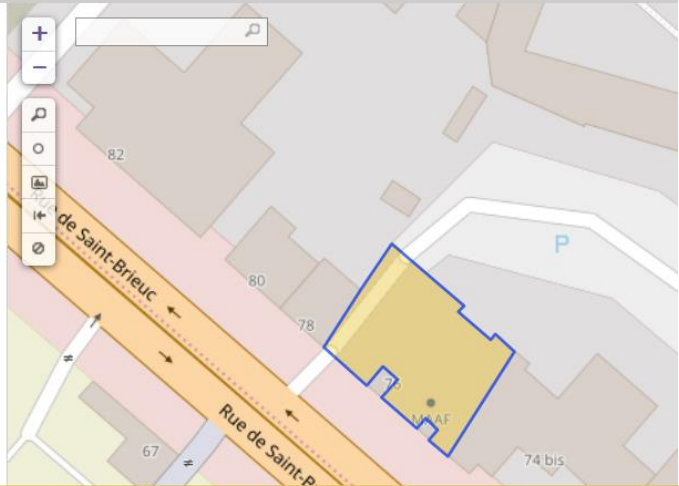


+ Contrôle données

Superposition bâtiments / routes

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

```
1 way[!covered][!tunnel]
2
3 ["highway"~"primary|secondary|tertiary|trunk|service|residential|primary_li
4 nk|secondary_link|tertiary_link|unclassified"]["access"!~"no|private"]
5 [!area]
6   ({{bbox}});
7   (way(around:0)["building"~"."]);
8   node(w););
9 out meta;
```



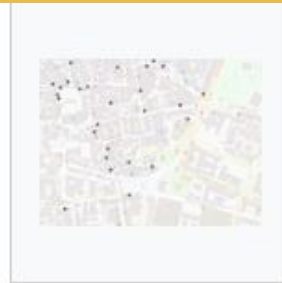
```
way
["highway"~"primary | secondary | tertiary | trunk | service | residential | primary_link |
secondary_link | tertiary_link | unclassified"]["access"!~"no | private"][!area]
  ({{bbox}});
  (way(around:0)["building"~"."]);
  node(w););
out meta;
```



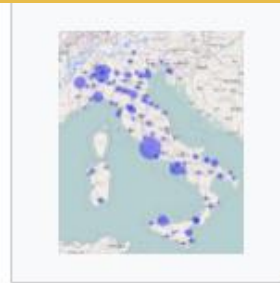
line styles



color coding



icons



a simple thematic map



markers with text

+

Requêtes et style

https://wiki.openstreetmap.org/wiki/Overpass_turbo/MapCSS



Requête et style

Afficher des étiquettes

Exécuter

Partager

Exporter

Assistant

Enregistrer

Charger

Paramètres

Aide

overpass turbo

Carte

Données

```
1
2
3 node["public_transport"="stop_position"] ["subway"="yes"] ({{bbox}});
4
5
6 {{style:
7 node, way, relation { text: name; }
8 }}
9
10
11 out body;
12 >;
13 out skel qt;
```

{{style:

node { text: name;}

}}

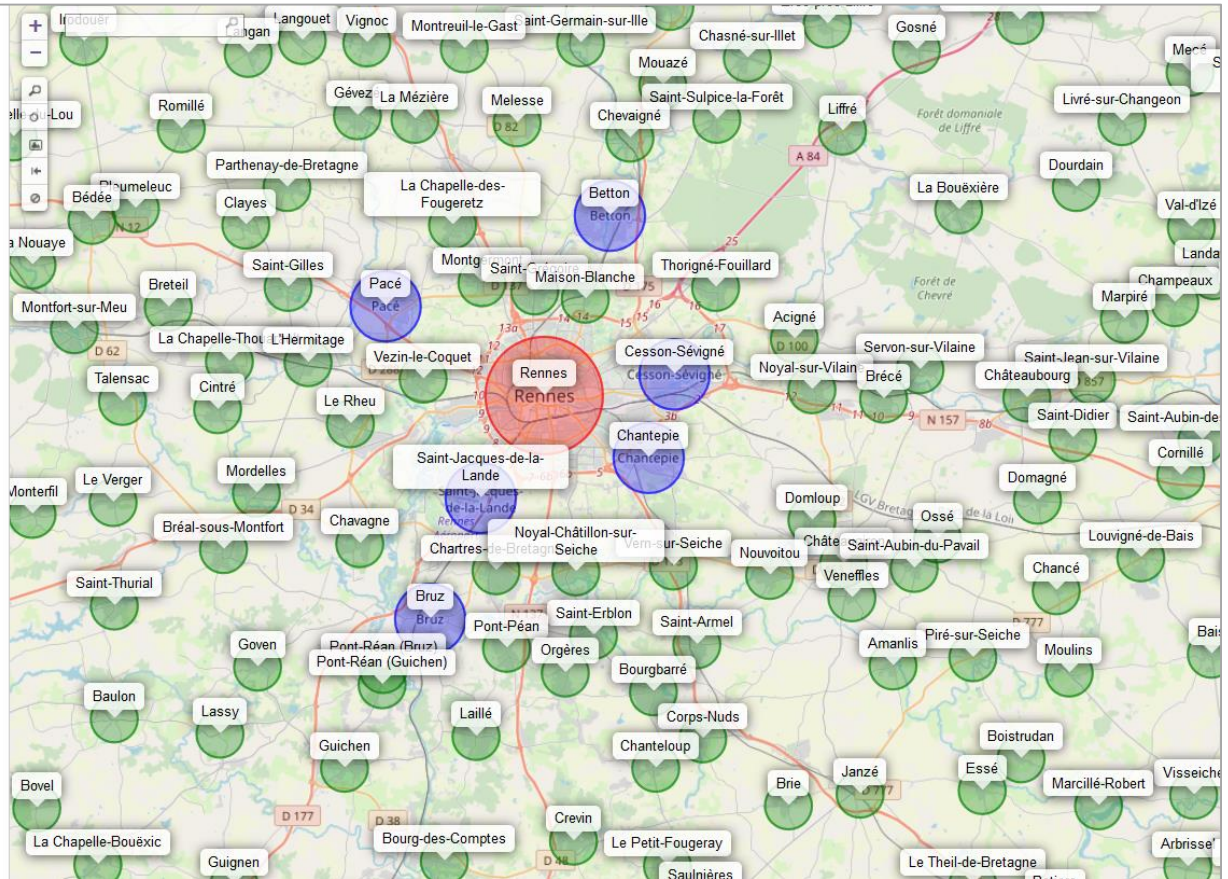
Chargé - nœuds: 59, chemins: 0, relations: 0
Affiché - points d'intérêt - POIs: 59, lignes: 0, polygones: 0



Requête et style

Afficher des étiquettes en fonction du type de ville

```
[bbox:{{bbox}}];  
(node[place=city];  
 node[place=town];  
 node[place=village];  
);  
  
{{style:  
 node [place=city]{text:name ; color:red; fill-color:red;  
 symbol-size:50; }  
 node [place=town]{text:name ; color:blue; fill-  
 color:blue; symbol-size:30;}  
 node [place=village]{text:name ; color:green; fill-  
 color:green; symbol-size:20;}  
}}  
  
out body;  
>;  
out skel qt|
```





Requête et style



Afficher des étiquettes en fonction du type de ville

```
[bbox:{{bbox}}];
(node[pl
node[pl
node[pl
]);

{{style:
node [pl
symbol-s
node [pl
color:bl
node [pl
color:gr
}}

out body
>;
out skel

[{{bbox:{{bbox}}}}];
(node[place=city];
node[place=town];
node[place=village];
);

{{style:
node [place=city]{text:name ; color:red; fill-color:red; symbol-size:50;}
node [place=town]{text:name ; color:blue; fill-color:blue; symbol-size:30;}
node [place=village]{text:name ; color:green; fill-color:green; symbol-size:20;}
}}

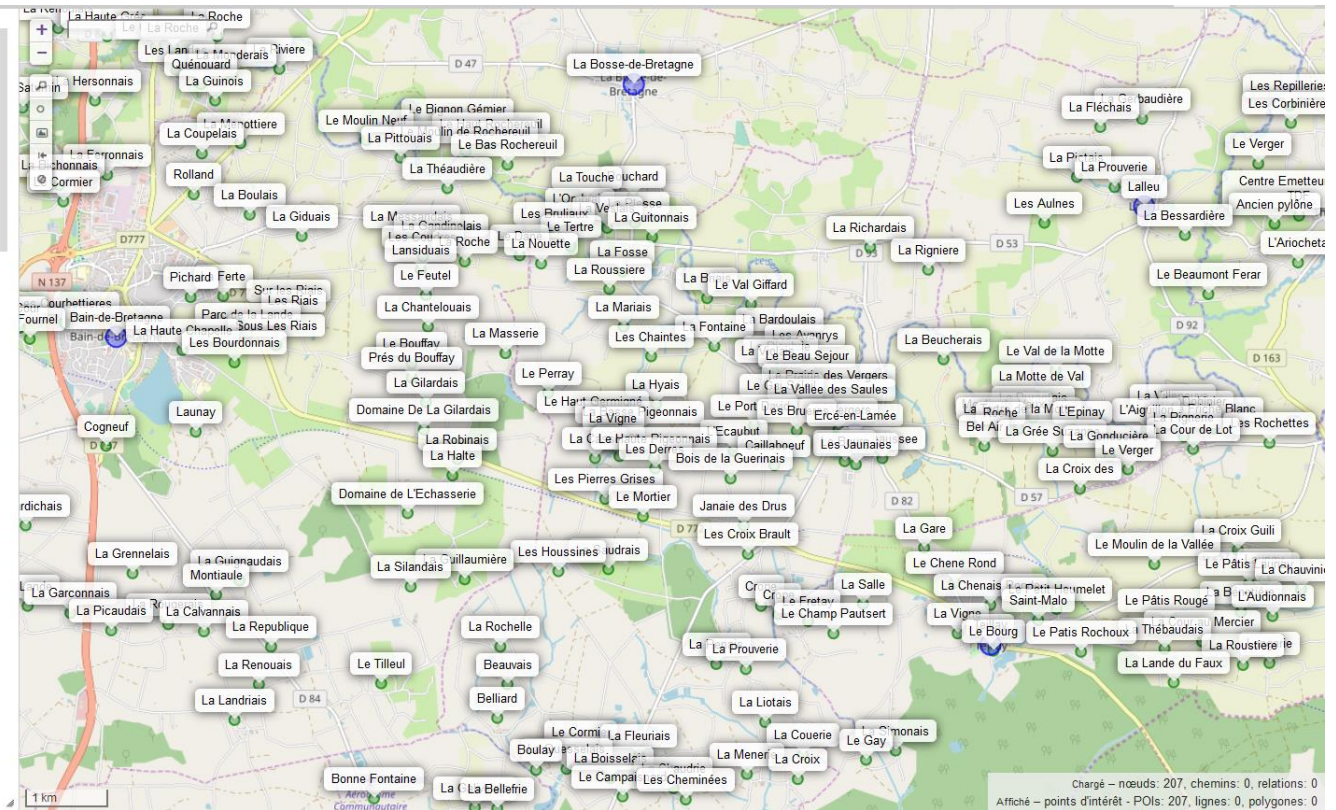
out body;
>;
out skel qt;
```




Requête et style

Afficher des étiquettes en fonction du type de ville et hameau

```
1 [bbox: [bbox]];  
2 (node[place=village];  
3 node[place=hamlet];  
4 );  
5  
6 {{style:  
7 node [place=hamlet]{text:name ; color:green; fill-  
8 color:green; symbol-size:5; font-size:2;}  
9 node [place=village]{text:name ; color:blue; fill-  
10 color:blue; symbol-size:10;}  
11 }}  
12 out body;  
13 >;  
14 out skel qt;
```





Requête et style

Afficher des étiquettes en fonction du type de ville et hameau

```
1 [bbox:{{bbox}}];  
2 (node[place=village];  
3 node[place=hamlet];  
4 );  
5  
6  
7 {{style:  
8 node [place=hamlet]{text:name ; color:green;  
9 color:green; symbol-size:5; font-size:2;  
10 node [place=village]{text:name ; color:blue;  
11 color:blue; symbol-size:10;  
12 }}  
13  
14 out body;  
15 >;  
16 out skel qt;
```

```
[bbox:{{bbox}}];  
(node[place=village];  
node[place=hamlet];  
);  
  
{{style:  
node [place=hamlet]{text:name ; color:green; fill-color:green;  
symbol-size:5; font-size:2;}  
node [place=village]{text:name ; color:blue; fill-color:blue;  
symbol-size:10;}  
}}  
  
out body;  
>;  
out skel qt;
```




Requête et style



Sélectionner et représenter les types de bâtiments

```
[out:json][timeout:25];
```

```
( way["building" = "apartments"]({{bbox}});  
way["building" = "residential"]({{bbox}});  
way["building" = "house"]({{bbox}});  
way["building" = "school"]({{bbox}});
```

```
{{style:
```

```
way[building=apartments]  
{ color:blue; fill-color:blue; }
```

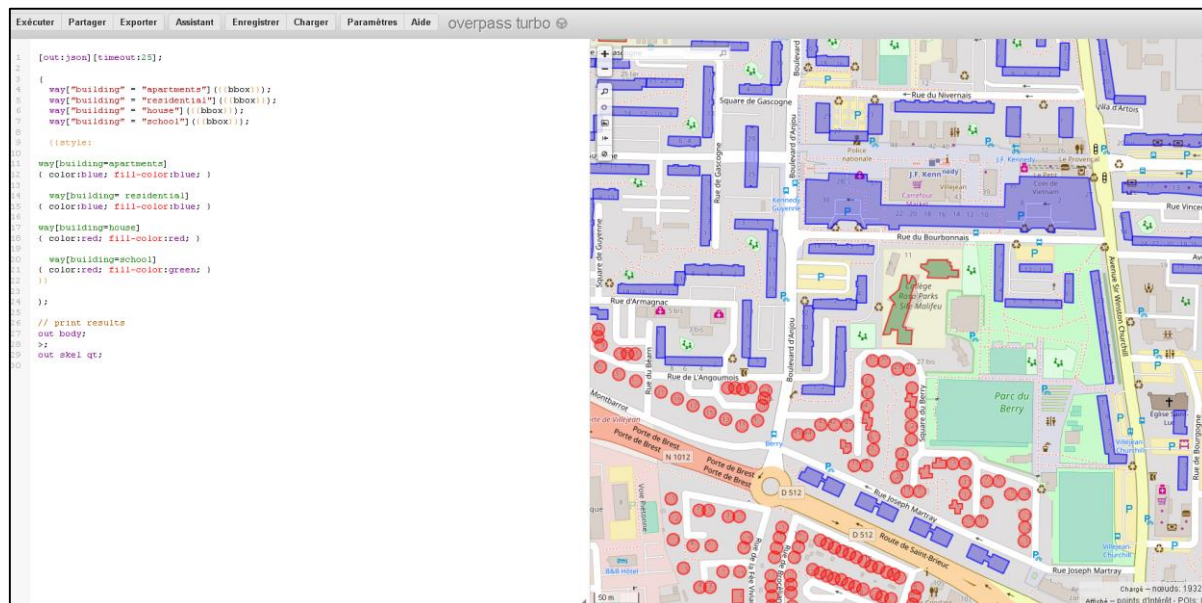
```
way[building=residential]  
{ color:blue; fill-color:blue; }
```

```
way[building=house]  
{ color:red; fill-color:red; }
```

```
way[building=school]  
{ color:red; fill-color:green; }  
}}
```

```
);
```

```
// print results  
out body;  
>;  
out skel qt;
```



Sélectionner et représenter les arrêts de bus, stations de métros et stations de vélos en libre service

```
[out:json][timeout:25];

{{geocodeArea:rennes}}->.searchArea;

( node["public_transport"="stop_position"] ["subway"="yes"]
(area.searchArea);
  node["highway"="bus_stop"](area.searchArea);
  node["amenity"="bicycle_rental"](area.searchArea);

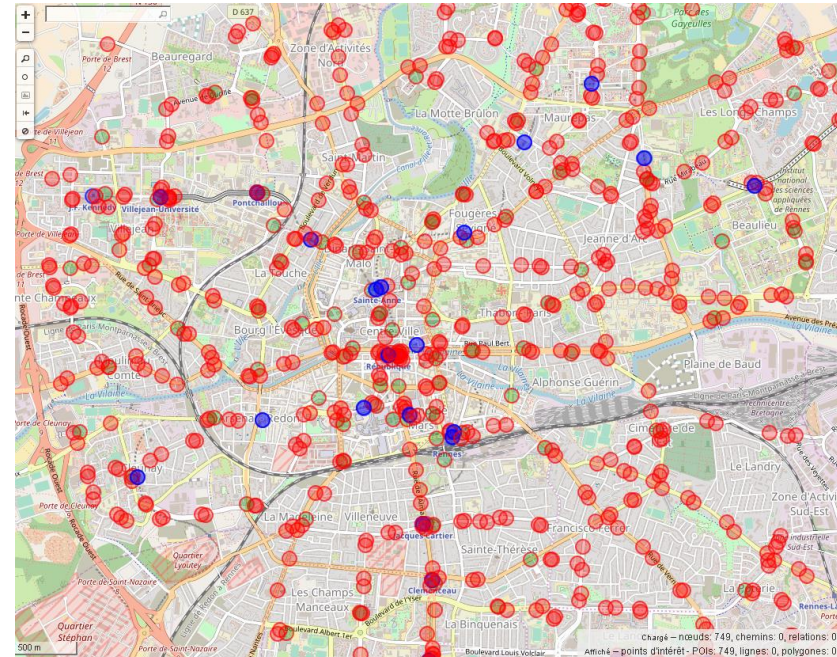
{{style:

node[public_transport=stop_position]
{ color:blue; fill-color:blue; }

node[highway=bus_stop]
{ color:red; fill-color:red; }

node[amenity=bicycle_rental]
{ color:red; fill-color:green; }
}}

);
out body;
>;
out skel qt;
```





Requête et style



Utiliser des icones (maki)

<https://www.mapbox.com/maki-icons/>

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 an SVG in two sizes: 11px by 11px and 15px by 15px. Maki is open source and [CC0 licensed](#).

↓ Download Maki

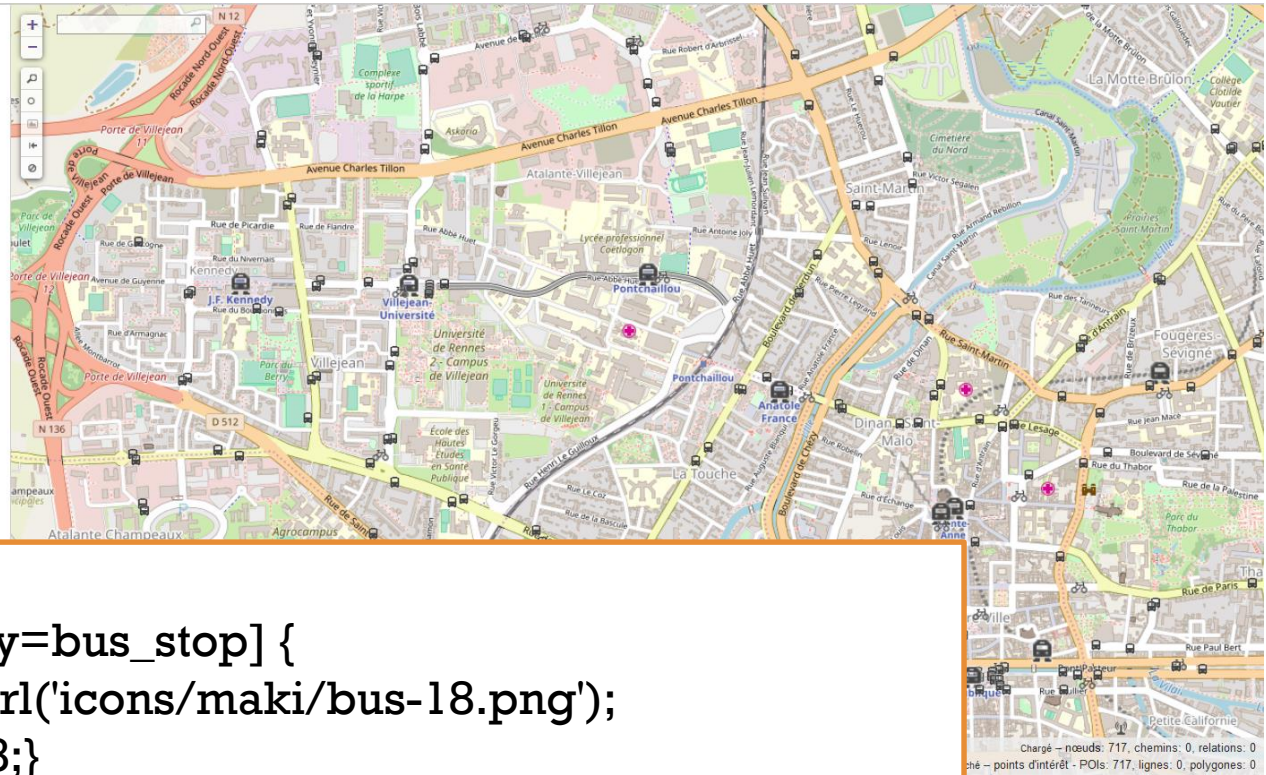




Requête et style

Utiliser des icones (maki)

```
1 [out:json][timeout:25];
2
3 ({(geocodeArea:rennes)->.searchArea;
4
5   ( node["public_transport"="stop_position"]
6     ["subway"="yes"] (area.searchArea);
7     node["highway"="bus_stop"] (area.searchArea);
8     node["amenity"="bicycle_rental"] (area.searchArea);
9
10  });
11
12  {{style:
13
14    node[highway=bus_stop] {
15      icon-image: url('icons/maki/bus-18.png');
16      icon-width: 15;
17
18    }
19
20    node[public_transport=stop_position] {
21      icon-image: url('icons/maki/rail-18.png');
22      icon-width: 30;
23
24    }
25
26    node[amenity=bicycle_rental] {
27      icon-image: url('icons/maki/bicycle-18.png');
28      icon-width: 20;
29    }
30  }};
31
32  out body;
33  >;
34  out skel qt;
```



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



Requête et style



Mise en forme des sentiers

network	iwn	International walking network: long distance paths that cross several countries
	nwn	National walking network: long distance paths
	rwn	Regional walking network: used for walking routes that cross regions
	lwn	Local walking network: used for small local walking routes

```
[bbox:{{bbox}}];
```

```
(relation[route=hiking][network~"^wn$"];way(r);>);out;
```

```
{{style:
```

```
relation[network=lwn] way { color:blue; fill-color:cyan; }
```

```
relation[network=iwn] way { color:red; fill-color:red; }
```

```
relation[network=nwn] way { color:green; fill-color:green; }
```

```
relation[network=rwn] way { color:yellow; fill-color:yellow; }
```

```
}}
```




Requête et style



Mise en forme des sentiers

Exécuter

Partager

Exporter

Assistant

Enregistrer

Charger

Paramètres

Aide

overpass turbo

Carte

Données

```
1 [bbox:{{bbox}}];
2
3 (relation[route=hiking][network~"^.wn$"];way(r);>);out;
4
5 {{style:
6
7 relation[network=lwn] way { color:blue; fill-color:cyan; }
8 relation[network=iwn] way { color:red; fill-color:red; }
9 relation[network=nnw] way { color:green; fill-color:green; }
10 relation[network=rwn] way { color:yellow; fill-color:yellow; }
11
12 }}
13
```

Chargé – nœuds: 75447, chemins: 3024, relations: 65
Affiché – points d'intérêt - POIs: 490, lignes: 3089, polygones: 0



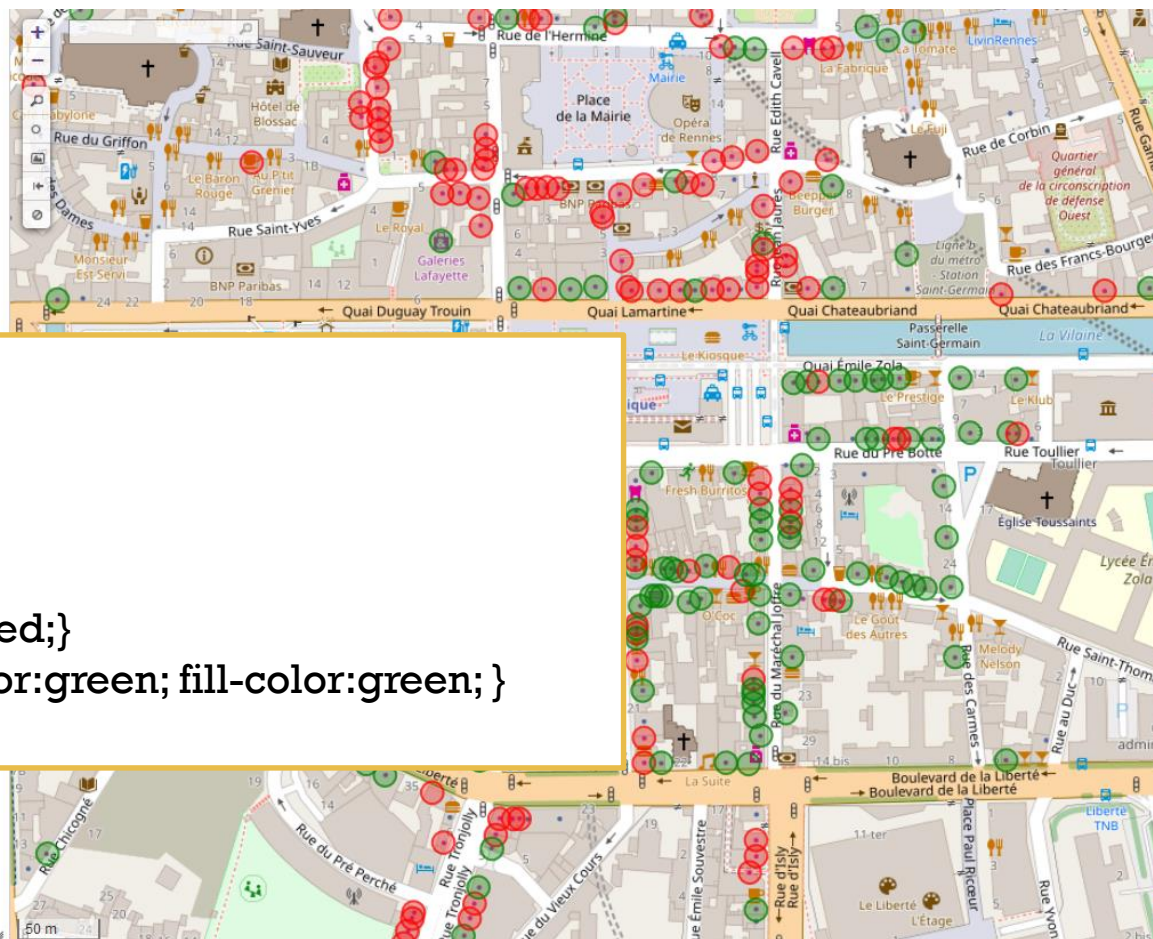
Requête et style

Mise en forme des magasins selon si l'information des horaires est remplie ou non

```
1 node[shop]
2   ({{bbox}});
3 out;
4
5 {{style:
6 node
7 { color:red; fill-color: red;}
8
9 node[opening_hours]
10 { color:green; fill-color:green; }
11 }}
```

```
node[shop]
  ({{bbox}});
out;

{{style:
node {color:red; fill-color: red;}
node[opening_hours] { color:green; fill-color:green; }
}}
```



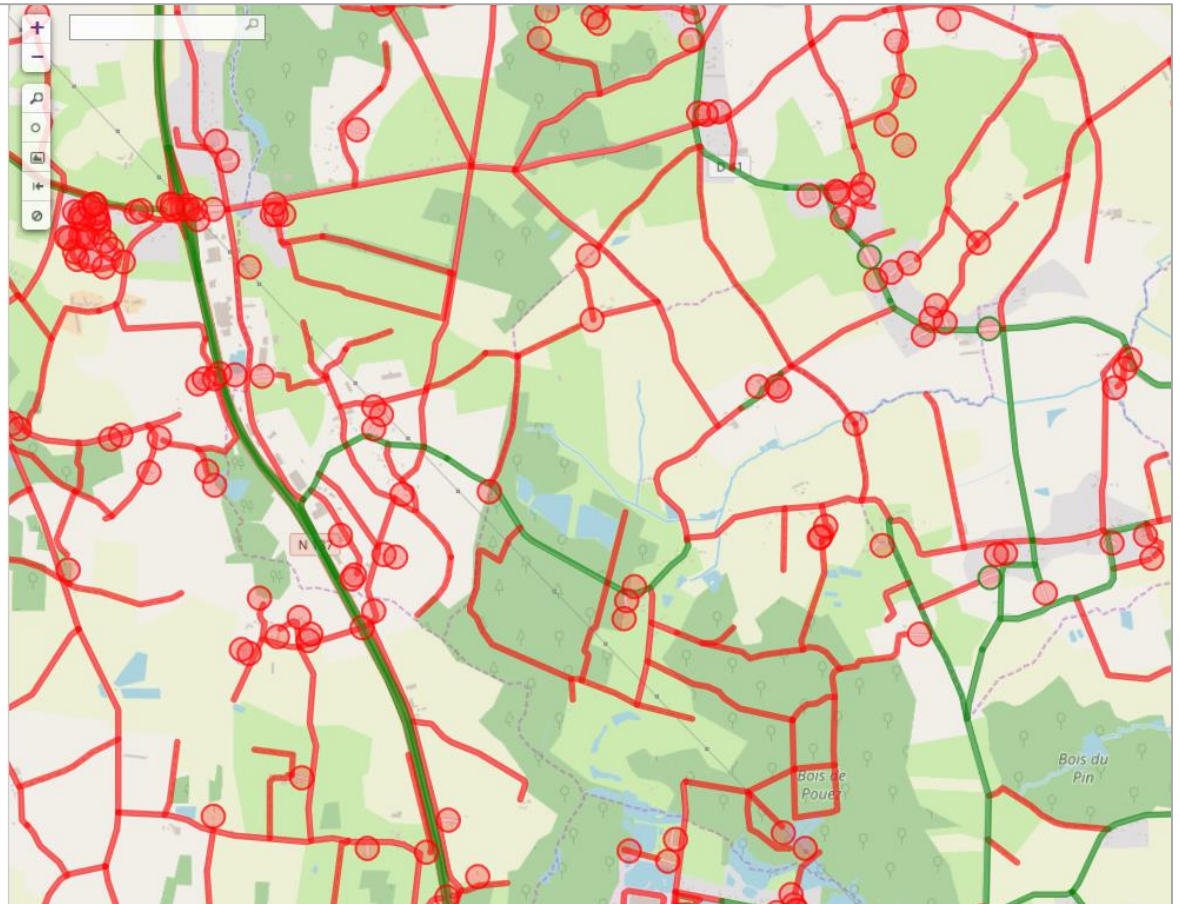


Requête et style



Mise en forme des routes selon si l'information de vitesse est remplie ou non

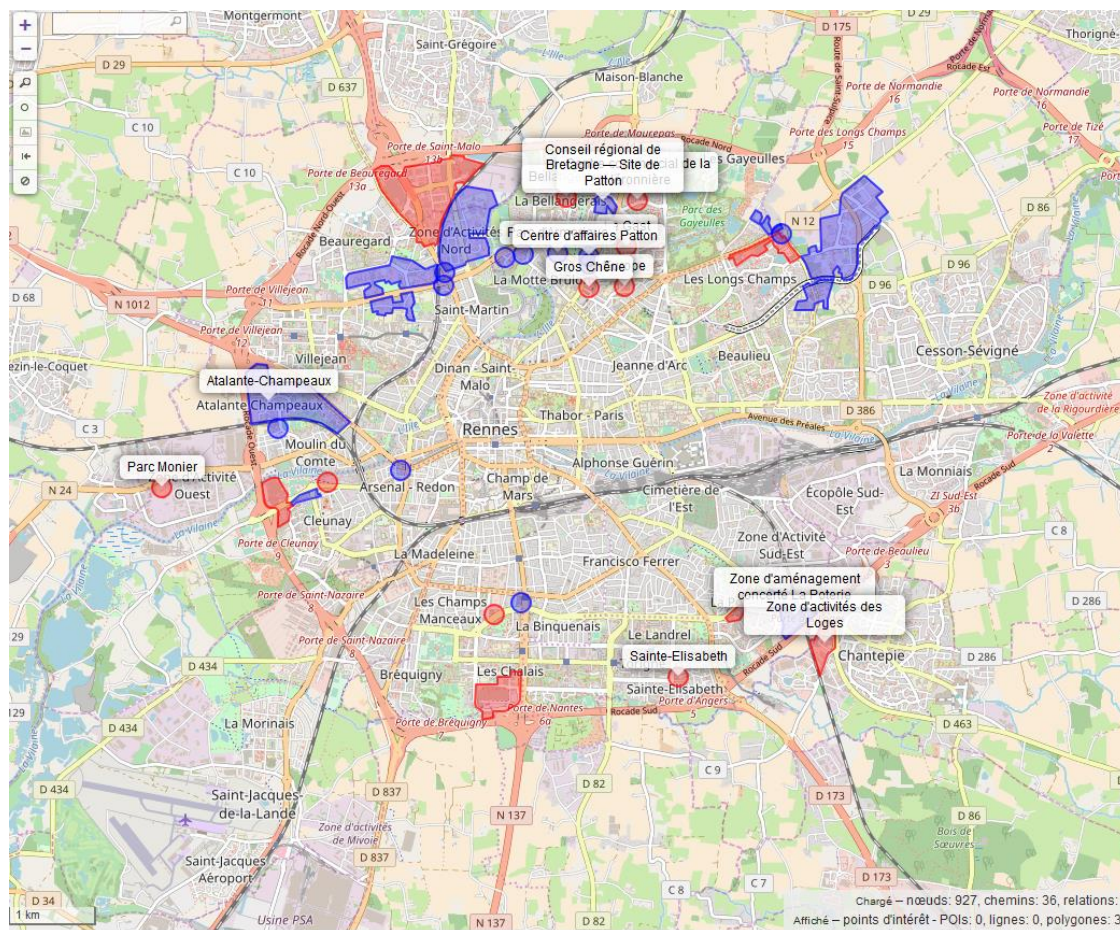
```
way ["highway"]([bbox]);  
  
{{style:  
  way{color: red;}  
  way[maxspeed]{ color:green; }  
}}  
  
// print results  
out body;  
>;  
out skel qt;
```





Requête et style

Sélectionner les zones commerciales avec un affichage stylisé et des étiquettes



<http://overpass-turbo.eu/s/z5J>

**+ Requêtes
contributeurs / date**



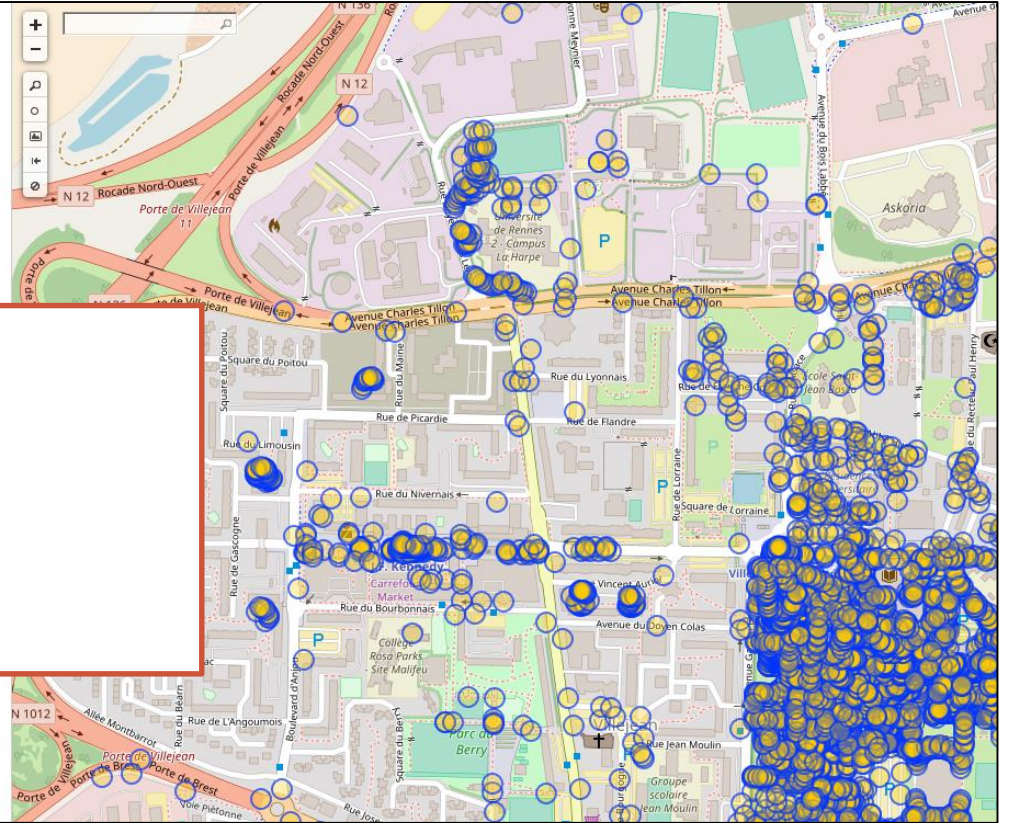
Extraction par contributeur



```
[out:json][timeout:25];  
(node(user:PanierAvide)({{bbox}}}));  
  
out body;  
>;  
out skel qt;
```

```
[out:json][timeout:25];  
(node(user:ninanon)({{bbox}}}));
```

```
out body;  
>;  
out skel qt;
```





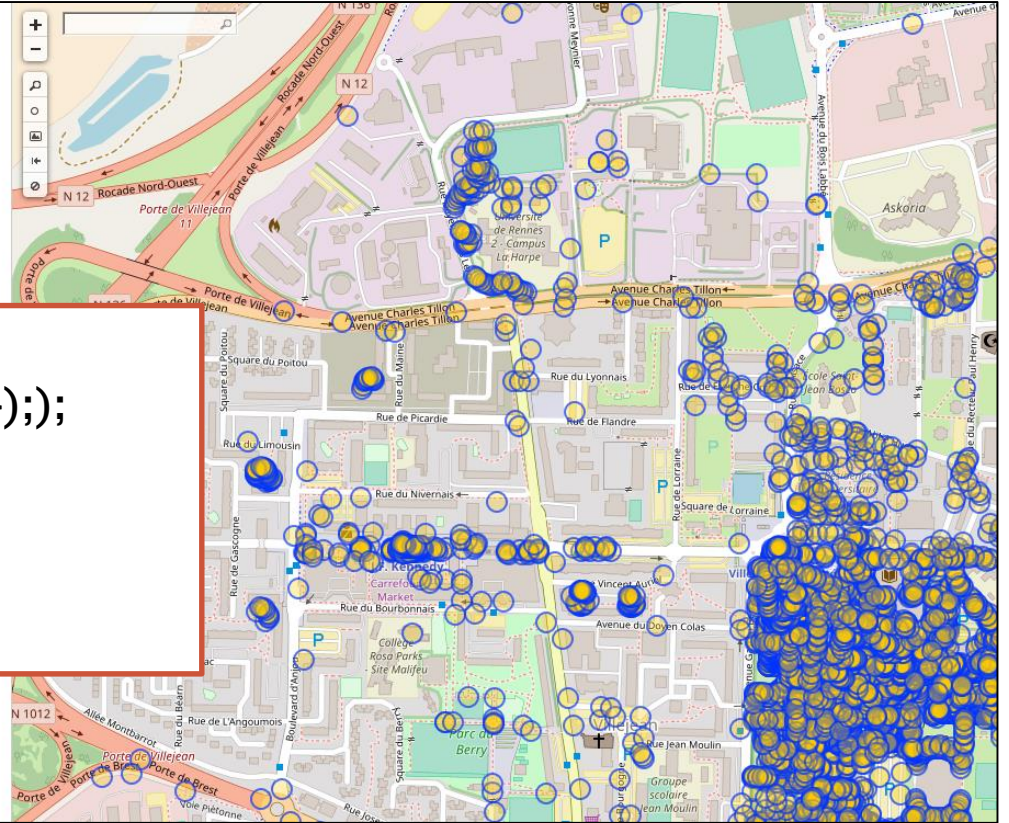
Extraction par contributeur



```
[out:json][timeout:25];  
(node(user:PanierAvide)({{bbox}}));  
  
out body;  
>;  
out skel qt;
```

```
[out:json][timeout:25];  
(node(user:PanierAvide)({{bbox}}));
```

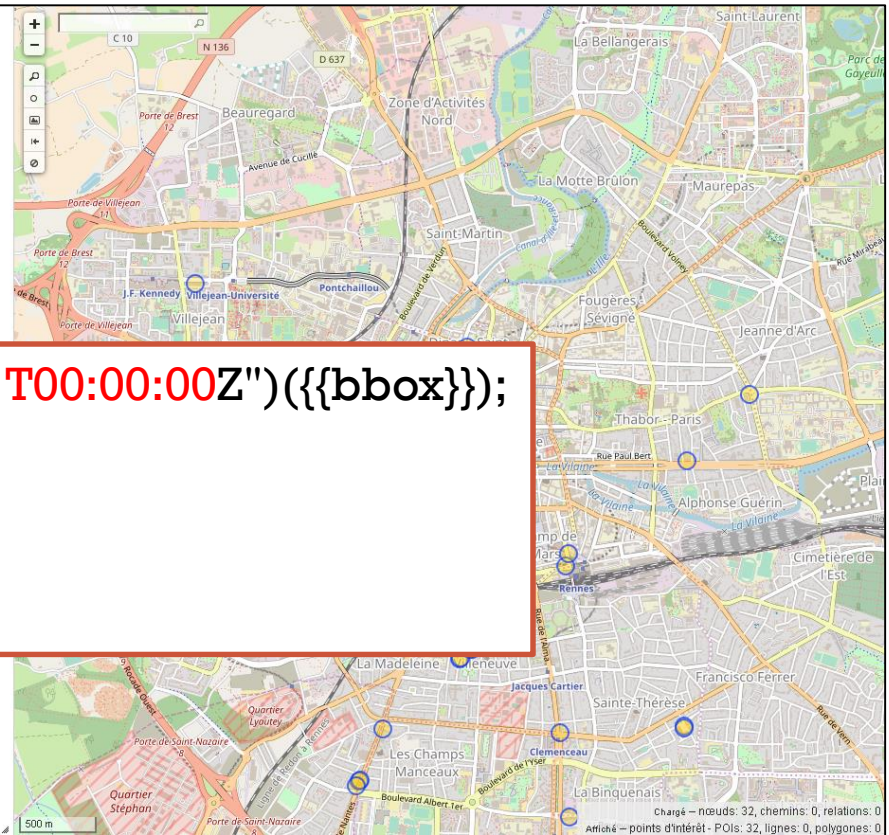
```
out body;  
>;  
out skel qt;
```



+ Extraction par date

Sélectionner les nouvelles aménités depuis le 1^{er} septembre

```
1 [out:json][timeout:25];  
2 { node["amenity"] (newer:"2017-11-01T07:00:00Z") ({bbox});  
3 };  
4 out body;  
5 >;  
6 out skel qt;  
7 |
```

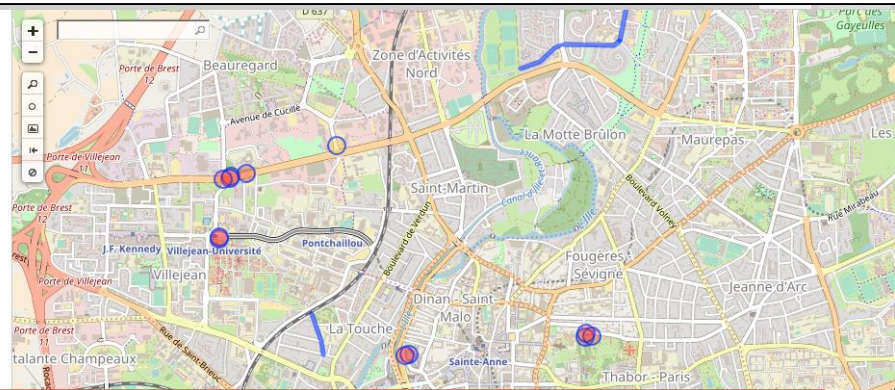


```
( node["amenity"](newer:"2020-09-01T00:00:00Z")({bbox}));  
);  
out body;  
>;  
out skel qt;
```


+ Extraction par date

Sélectionner les routes mises à jour depuis le 1^{er} septembre

```
[out:json][timeout:25];
(
  node["highway"] (changed:"2017-11-01T07:00:00Z") ({{bbox}});
  way["highway"] (changed:"2017-11-01T07:00:00Z") ({{bbox}});
);
out body;
>;
out skel qt;
```



```
( node["highway"](changed:"2020-09-01T07:00:00Z")({{bbox}});
  way["highway"](changed:"2020-09-01T07:00:00Z")({{bbox}});
);
out body;
>;
out skel qt;
```



+

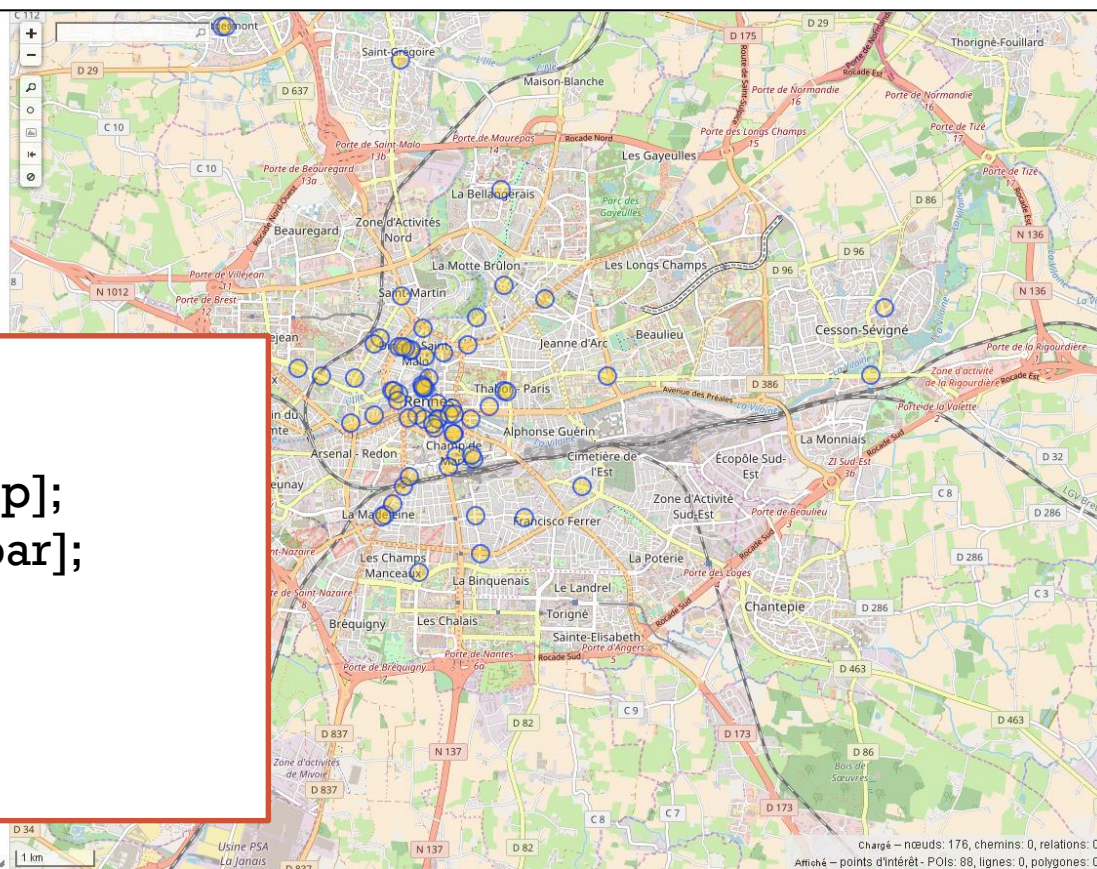
Requêtes spatiales

+ Sélection spatiale

Sélectionner tous les bars à moins de 100m d'un arrêt de bus

```
1 [out:json][timeout:25];  
2   area[name="Rennes"];  
3   node(area)[highway=bus_stop];  
4   node(around:100)[amenity=bar];  
5   out;  
6 out body;  
7 >;  
8 out skel qt;  
9
```

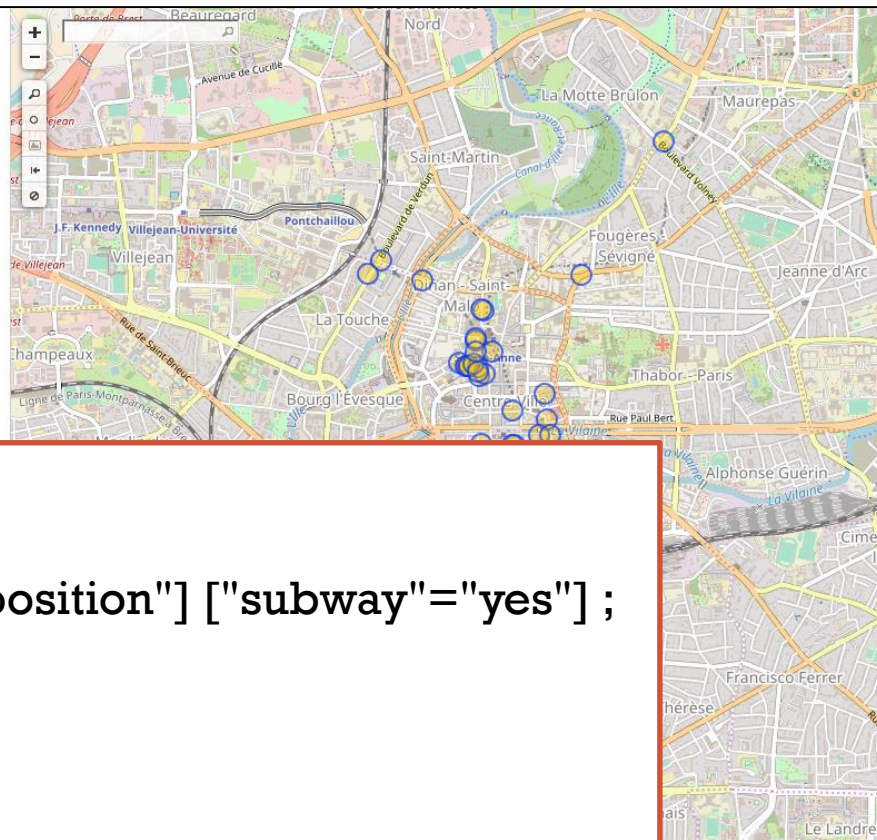
```
[out:json][timeout:25];  
area[name="Rennes"];  
node(area)[highway=bus_stop];  
node(around:100)[amenity=bar];  
out;  
out body;  
>;  
out skel qt;
```



+ Sélection spatiale

Sélectionner tous les bars à moins de 200m d'une station de métro

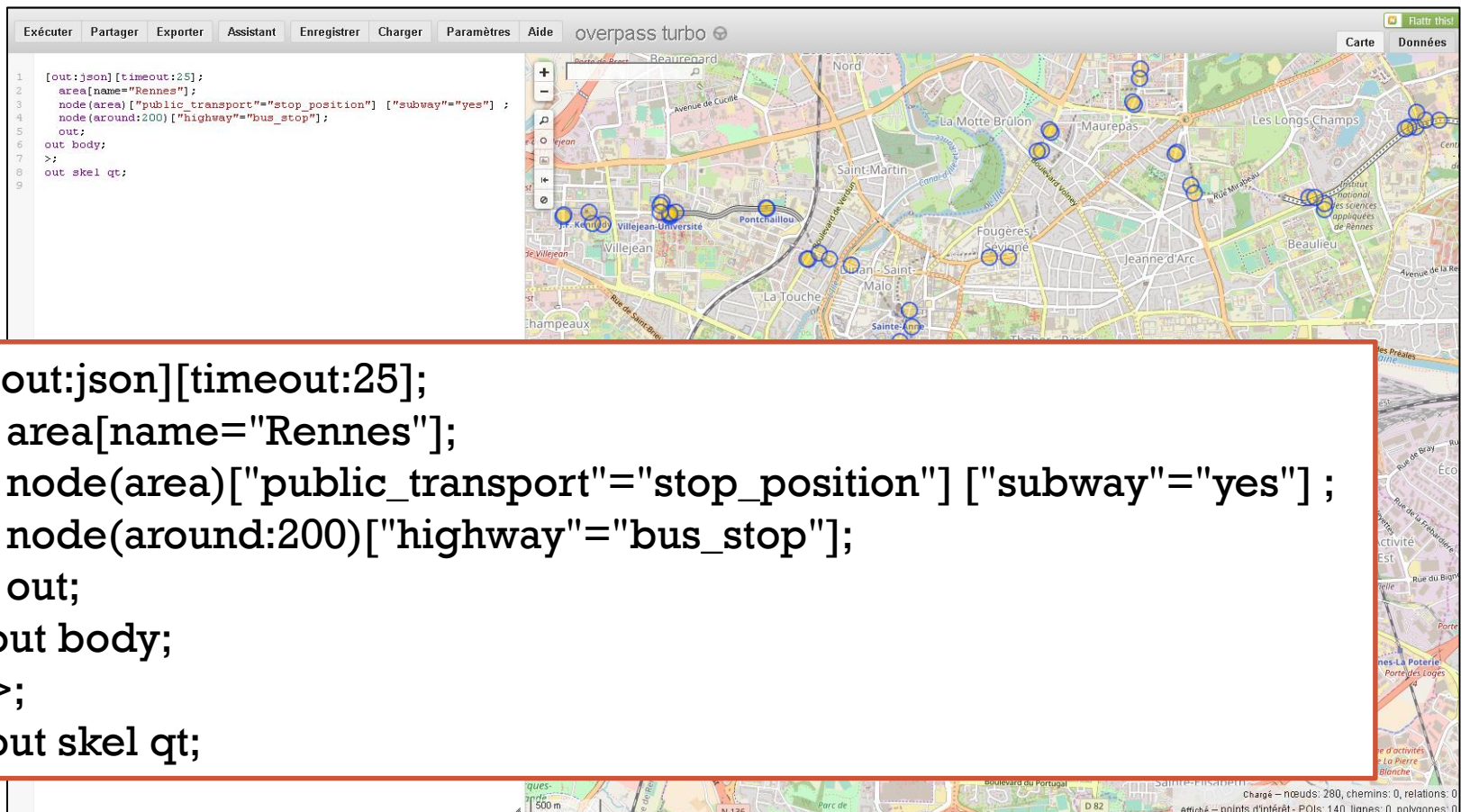
```
1 [out:json][timeout:25];  
2   area[name="Rennes"];  
3   node(area)["public_transport"="stop_position"] ["subway"="yes"] ;  
4   node(around:200)[amenity=bar];  
5   out;  
6 out body;  
7 >;  
8 out skel qt;  
9
```



```
[out:json][timeout:25];  
area[name="Rennes"];  
node(area)["public_transport"="stop_position"] ["subway"="yes"] ;  
node(around:200)[amenity=bar];  
out;  
out body;  
>;  
out skel qt;
```

+ Sélection spatiale

Sélectionner les arrêts de bus à moins de 200m d'une station de métro



```
[out:json][timeout:25];
area[name="Rennes"];
node(area)["public_transport"="stop_position"] ["subway"="yes"] ;
node(around:200)["highway"="bus_stop"];
out;
out body;
>;
out skel qt;
```

Chargé – nœuds: 280, chemins: 0, relations: 0
Affiché – points d'intérêt - POIs: 140, lignes: 0, polygones: 0

+ Sélection spatiale

Sélectionner les bâtiments isolés (100m)

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

```
1 way[building]({{bbox}})->.a;
2 foreach .a {
3   way.a(around:100);
4   way._(if:count(ways) == 1);
5   out center;
6 };
```

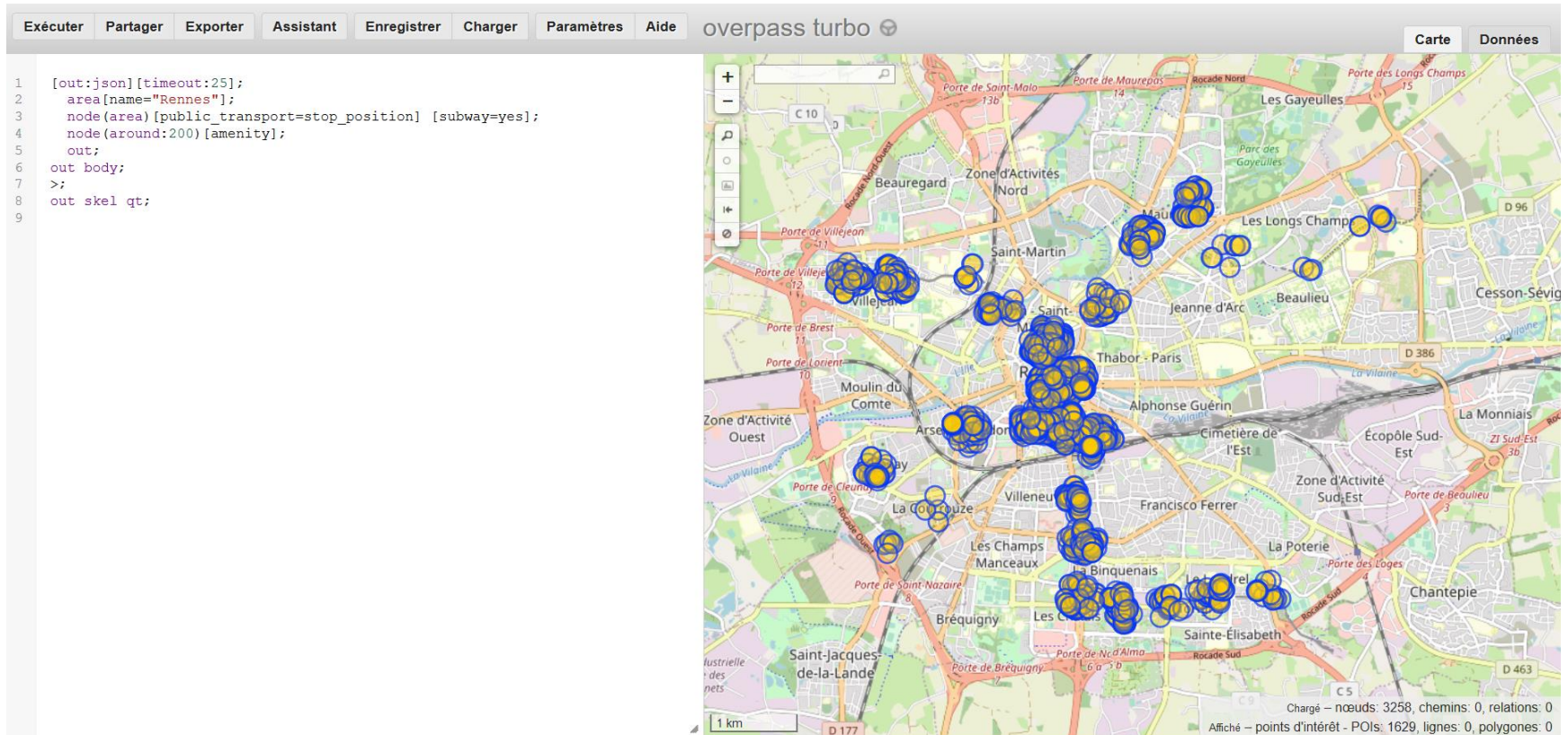
Carte Données

way[building]({{bbox}})->.a;
foreach .a (
 way.a(around:100);
 way._(if:count(ways) == 1);
 out center;
);
out body;
>;
out skel qt;

chargé - nœuds: 0, chemins: 6, relations: 1
ajout - points d'intérêt: 0, POIs: 6, lignes: 0, polygones: 0

+ Sélection spatiale

Sélectionner les aménités à moins de 200m d'une station de métro

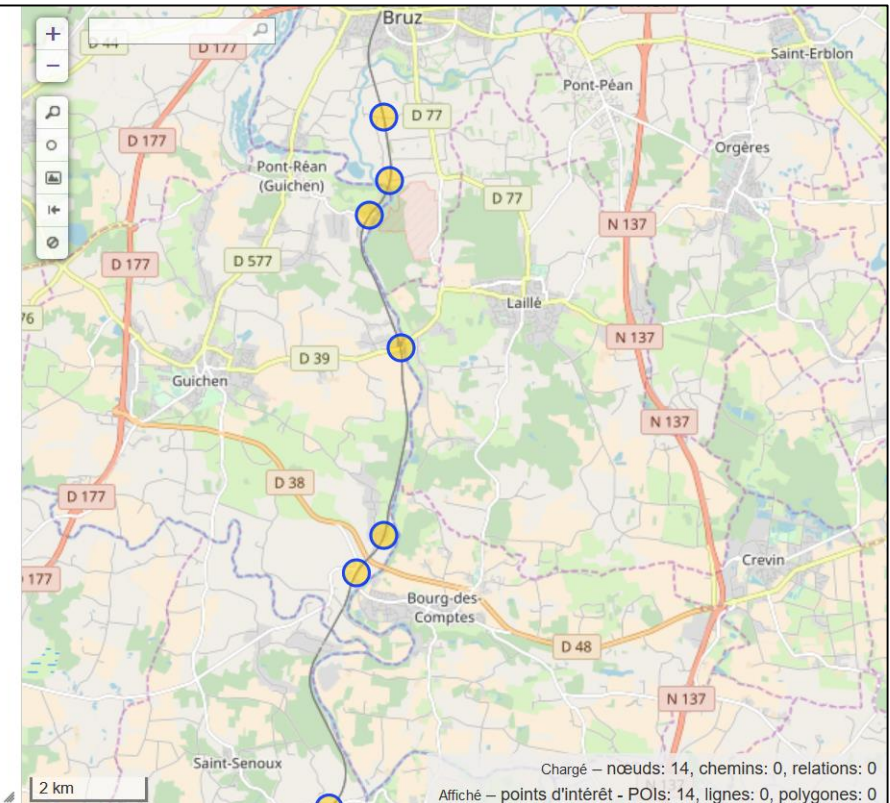


+ Sélection spatiale

Sélectionner les intersections entre routes et voies ferrées

```
1 [bbox:{{bbox}}];  
2 way["railway"="rail"]->.major;  
3 way["highway"]->.minor;  
4 node(w.major)(w.minor);  
5 out body;  
6 >;  
7 out skel qt;
```

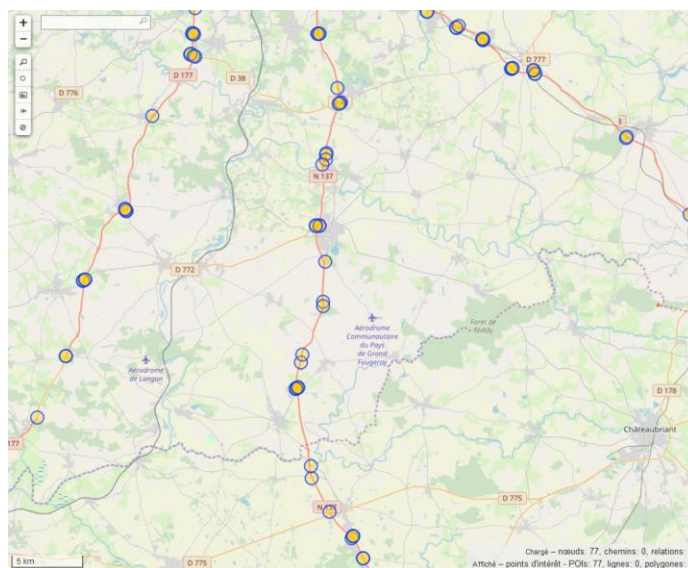
```
[bbox:{{bbox}}];  
way["railway"="rail"]->.major;  
way["highway"]->.minor;  
node(w.major)(w.minor);  
out body;  
>;  
out skel qt;
```



+ Sélection spatiale

Sélectionner les sorties des voies rapides

```
[bbox:{{bbox}}];  
way[highway~"^(motorway | trunk | (motorway | trunk)_link)$"]->.major;  
way[highway~"^(primary | secondary | service)$"]->.minor;  
node(w.major)(w.minor);  
out;
```



+ Sélections spatiales

Rechercher toutes communes qui n'ont pas d'école

```
// Output in CSV format
[out:csv(::id, name)];

area[admin_level=6][name="Ille-et-Vilaine"][boundary=administrative]-
>.boundaryarea;

// Echelle d'analyse la commune
rel(area.boundaryarea)[admin_level=8];
// map relations to their respective area
map_to_area -> .all_level_8_areas;

// chercher toutes les écoles
( node(area.boundaryarea)["amenity"="school"];
  way(area.boundaryarea)["amenity"="school"];>;);

// qui sont situées dans des communes
is_in;
area._[admin_level=8] -> .level_8_areas_with_school;

// Toutes les communes qui ne possèdent pas d'école
(.all_level_8_areas; - .level_8_areas_with_school;);

// Calculate and output relation
rel(pivot);
out geom;
```

1	@id name
2	89002 Dourdain
3	106684 Arbrissel
4	106686 Avoilles-sur-Seiche
5	106687 Amanlis
6	107908 Balazé
7	107909 Bais
8	107911 Boistrudan
9	107912 Brielles
10	107915 Bruc-sur-Aff
11	107918 Brie
12	108429 Châtillon-en-Vendelais
13	108430 Chelun
14	108431 Chancé
15	108434 Champeaux
16	108437 Cornillé
17	108440 Coësmes
18	111020 Drouges
19	111021 Domalain
20	111055 Erbrée
21	111056 Eancé
22	111057 Essé
23	116110 Gahard
24	123371 La Bazouge-du-Désert
25	123374 Lourmais
26	123375 Longaulnay
27	123380 Le Tiercent
28	123382 La Baussaine
29	123383 Le Châtellier
30	123385 La Boussac
31	123386 Le Ferré
32	123387 Luitré
33	128198 La Bosse-de-Bretagne
34	134221 Lanrigan
35	134223 Lalleu
36	134316 Lieuron
37	139656 Marcillé-Robert
38	139657 Montreuil-sous-Pérouse
39	139658 Moulins
40	139663 Moutiers
41	139664 Montreuil-des-Landes
42	139666 Mouté



+ Edition de données avec
l'éditeur OSM ID



L'éditeur du site OSM ID

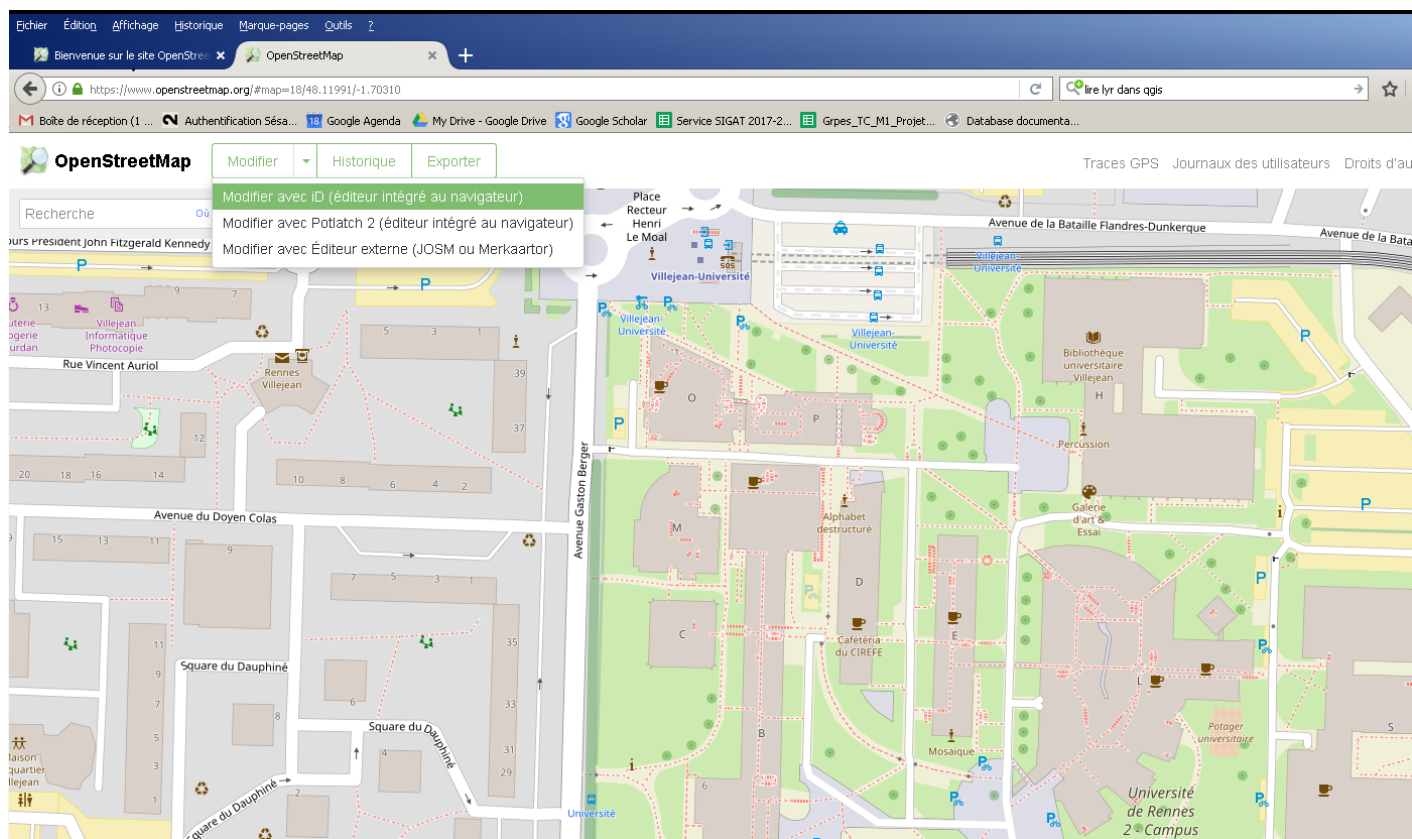
<http://www.openstreetmap.org/#map=5/48.821/9.053>

The screenshot shows the OpenStreetMap website interface. At the top, there is a navigation bar with the OpenStreetMap logo, a search bar, and buttons for 'Modifier', 'Historique', and 'Exporter'. Below the search bar, there is a welcome message: 'Bienvenue dans OpenStreetMap!'. The main content area displays a map of Europe, with various cities and countries labeled. A red box highlights the 'Se connecter' button in the top right corner. The map shows a detailed view of Europe, including major cities like London, Paris, Berlin, and Moscow. The interface is in French.



L'éditeur du site OSM ID

Modifier avec ID (éditeur intégré au navigateur)





L'éditeur du site OSM ID



Modifier avec ID (éditeur intégré au navigateur)

The screenshot displays the OpenStreetMap ID editor interface. The top navigation bar includes the OpenStreetMap logo, a 'Modifier' button, and links for 'Historique' and 'Exporter'. On the right, there are links for 'Traces GPS', 'Journaux des utilisateurs', 'Droits d'auteur', 'Aide', and 'À propos', along with a user profile for 'ninanoun'.

The main editing area is titled 'Modifier l'élément' and features a sidebar on the left with the following sections:

- Nom courant (si existant):** A text input field.
- Adresse:** Fields for 'Unité' (123) and 'Rue, Voie' (Ville).
- Étages:** A dropdown menu set to '4'.
- Sources:** A list of sources including 'cadastre-dgi-fr source : Direction Générale des Impôts - Cadastre. Mise à jour : 2010'.
- Ajouter champ:** A dropdown menu with options like 'Description, Altitude, Courrier électronique'.
- Tous les tags (4):** A table of tags for the selected element.
- Toutes les relations (0):** A section for relations.

The table of tags is as follows:

Tag	Value	Delete	Info
building	apartments		
building:levels	4		
roof:shape	flat		
source	cadastre-dgi-fr so ...		

The main map area shows an aerial view of a city street with various buildings and green spaces. A red building is highlighted, and a 'Sauvegarder' button is visible in the top right of the map area.



A faire et ne pas faire



Certains objets ne doivent pas être connectés

- Ex. bâtiments et route



Certains objets doivent toujours être connectés

- Ex. Intersections

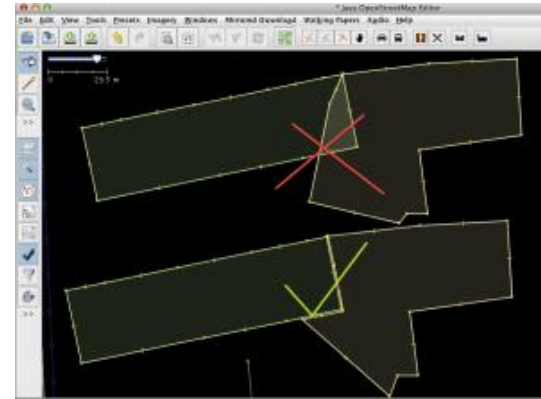




A faire et ne pas faire



Eviter les chevauchement d'objets



Toujours créer des points

- Ex. si la route se transforme (bitumée en piste) il faut créer un nouveau point et changer l'attribut de la route





Les attributs



Les **attributs** s'affichent dans une boîte à droite de la fenêtre quand un noeud/section/polygone est sélectionné

Vous pouvez soit ajouter les attributs manuellement (**Alt+A**) si vous les connaissez déjà (cela viendra vite pour les plus courants comme “highway=residential” ou “building=yes”...), soit aller chercher dans le menu “Attributs”, une liste d'attributs courants classés par catégorie vous est alors proposée.

Vous pouvez retrouver une liste assez complète des attributs correspondant à des objets observables ici :

http://wiki.openstreetmap.org/wiki/FR:Howto_Map_A



Les attributs



Si vous voulez progresser avec OSM, jeter un œil sur les descriptions des attributs sur le wiki, notamment :

- <http://wiki.openstreetmap.org/wiki/Key:natural> pour les objets naturels
- <http://wiki.openstreetmap.org/wiki/Key:landuse> pour l'usage du sol
- <http://wiki.openstreetmap.org/wiki/Key:highway> pour les routes et chemins
- <http://wiki.openstreetmap.org/wiki/Waterway> pour les cours d'eau