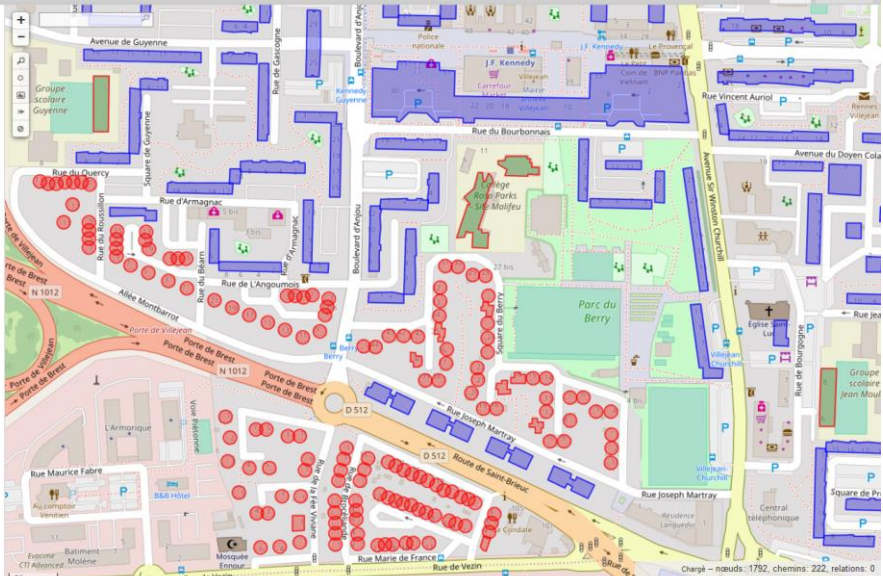



# Atelier : Prise en main de l'API Overpass

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

```
1 [out:json][timeout:25];
2
3 { way["building" = "apartments"]{({bbox}});
4   way["building" = "residential"]{({bbox}});
5   way["building" = "house"]{({bbox}});
6   way["building" = "school"]{({bbox}});
7
8   }
9
10 [style:
11
12 way[building=apartments]
13   { color:blue; fill-color:blue; }
14
15 way[building= residential]
16   { color:blue; fill-color:blue; }
17
18 way[building=house]
19   { color:red; fill-color:red; }
20
21 way[building=school]
22   { color:red; fill-color:green; }
23
24 ]};
25
26 // print results
27 out body;
28
29 out skel qt;
```



Chargé - nœuds: 1792, chemins: 222, relations: 0  
Affiché - points d'intérêt - POIs: 0, lignes: 0, polygones: 222

overpass turbo 

+

Overpass turbo



# API Overpass



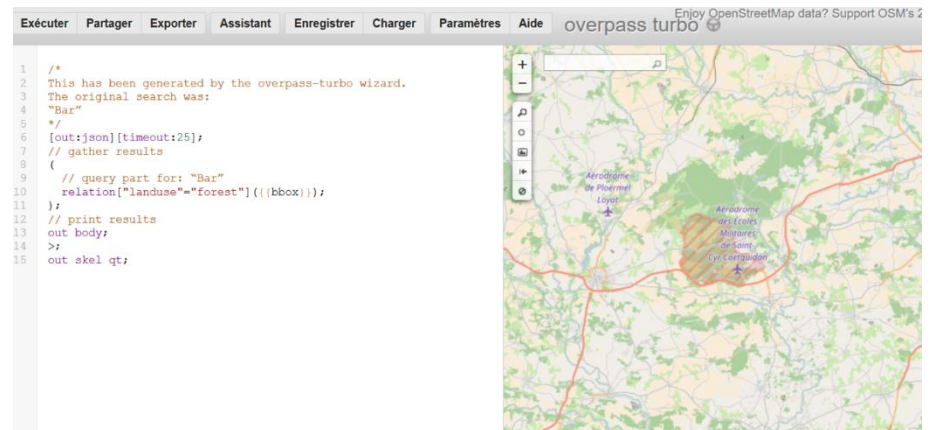
**API Overpass** permet d'interroger la base de données OSM

- 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/>





# Modèle de données OSM

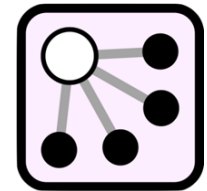
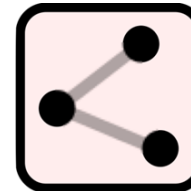
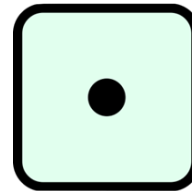


Dans un SIG, les données cartographiques sont représentées de trois façons différentes

- 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**) :

- **Noeuds** (*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)

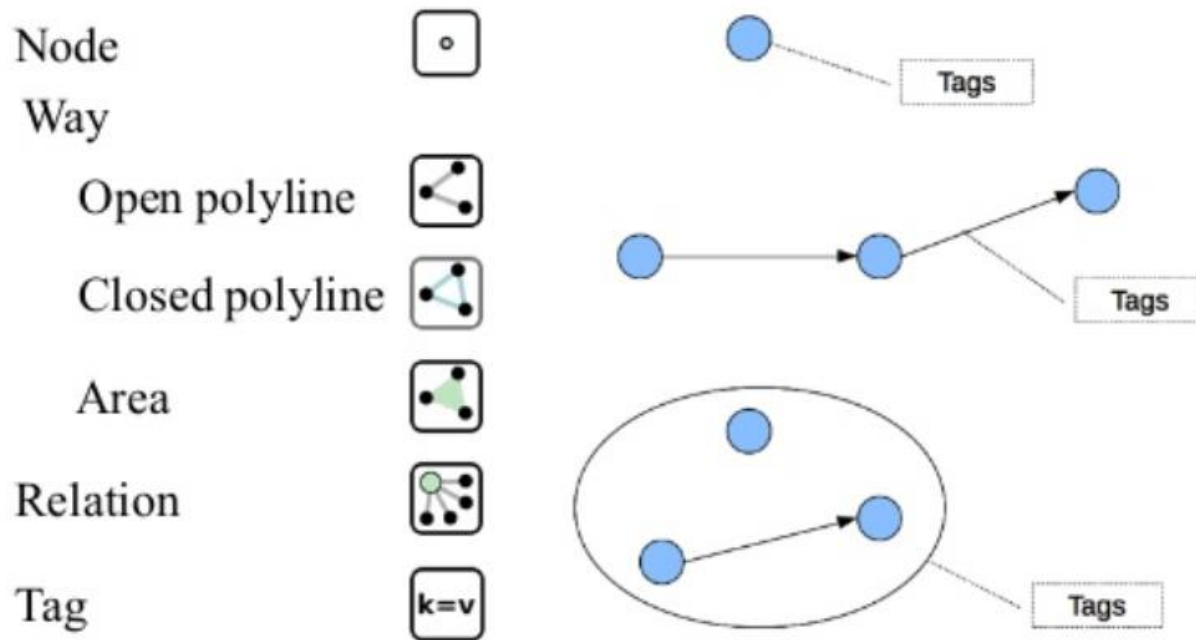


# Modèle de données OSM



Clip slide

## OSM Data Model





# La 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

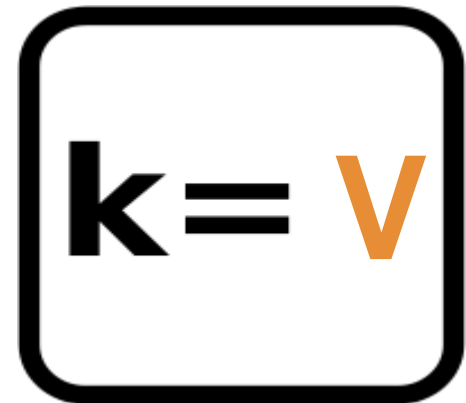


Le schéma des tags repose sur le fonctionnement  
<clé>=<valeur> (*key=value*)

→ 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





# + API Overpass



## 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  |           |       |

+

**Requêtes classiques**

# + Requêtes classiques

## Extraire les routes selon une valeur (hierarchie)

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

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

Type Key Value

way highway trunk

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

{{bbox}}

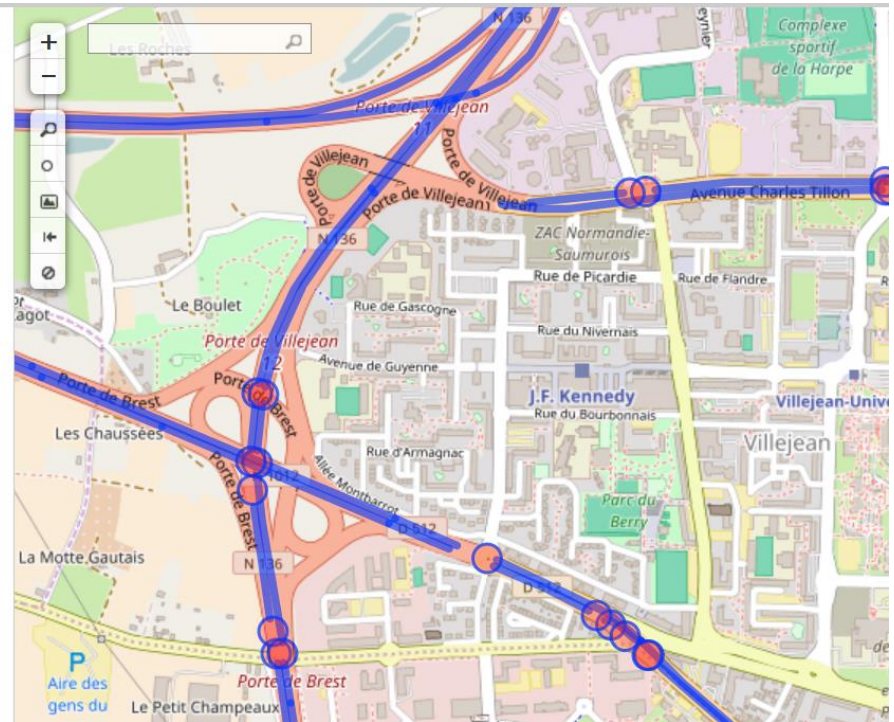
Emprise

# + 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;
```

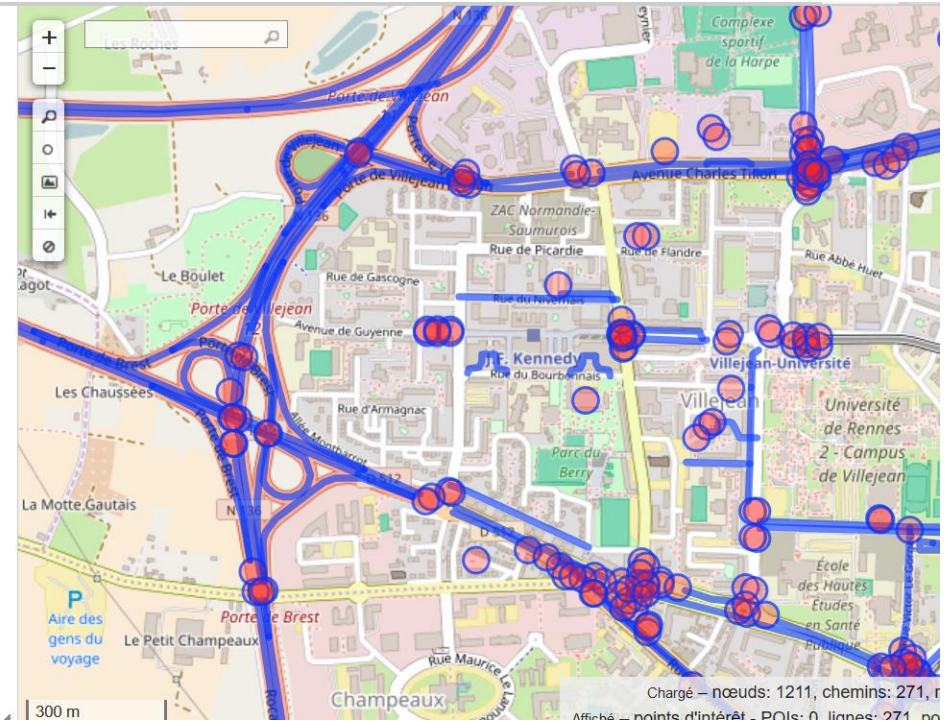
Chargé – nœuds: 400, cf  
Affiché – points d'intérêt - POIs: 0, li

# + Requêtes classiques

## 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**

```
1 /*
2 This has been generated by the overpass-turbo wizard.
3 The original search was:
4 "route in nantes"
5 */
6 [out:json][timeout:25];
7 // fetch area "nantes" 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;
```

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

# + Requête classiques

## Amenity















- Documentation

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

Key = **highway**

Value =

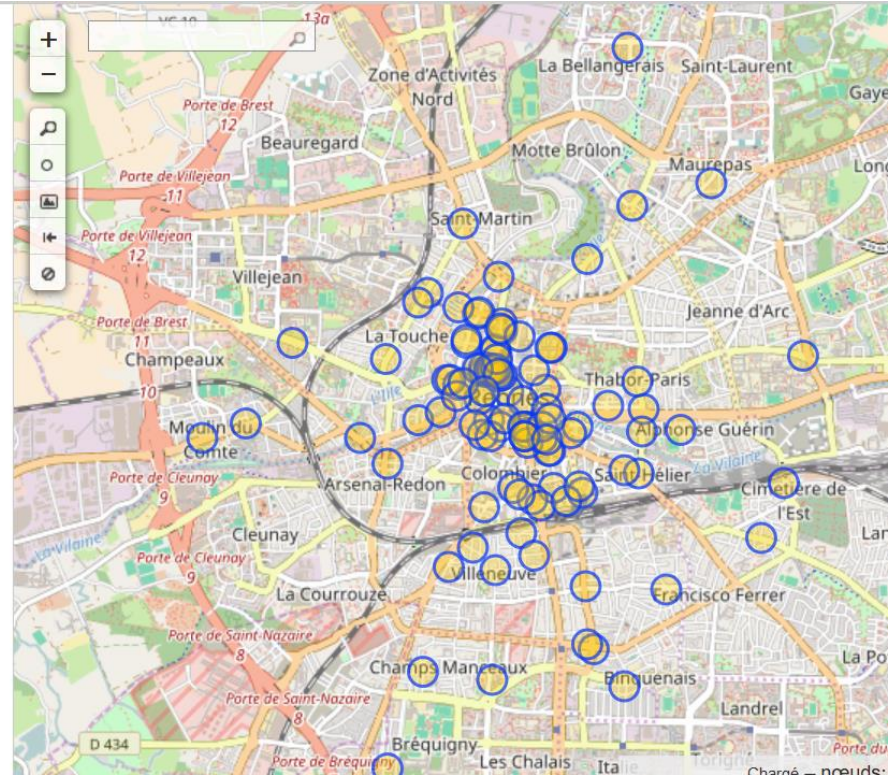
- bar
- recycling
- bench
- school
- ...

| Key               | Value          | Element   | Comment  | Rendering   | Photo   |
|-------------------|----------------|---|--|---|---|
| <b>Sustenance</b> |                |   |  |   |   |
| 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 <code>drinking fountain</code> or <code>bubbler</code> .  |  |  |

# + Requêtes classiques

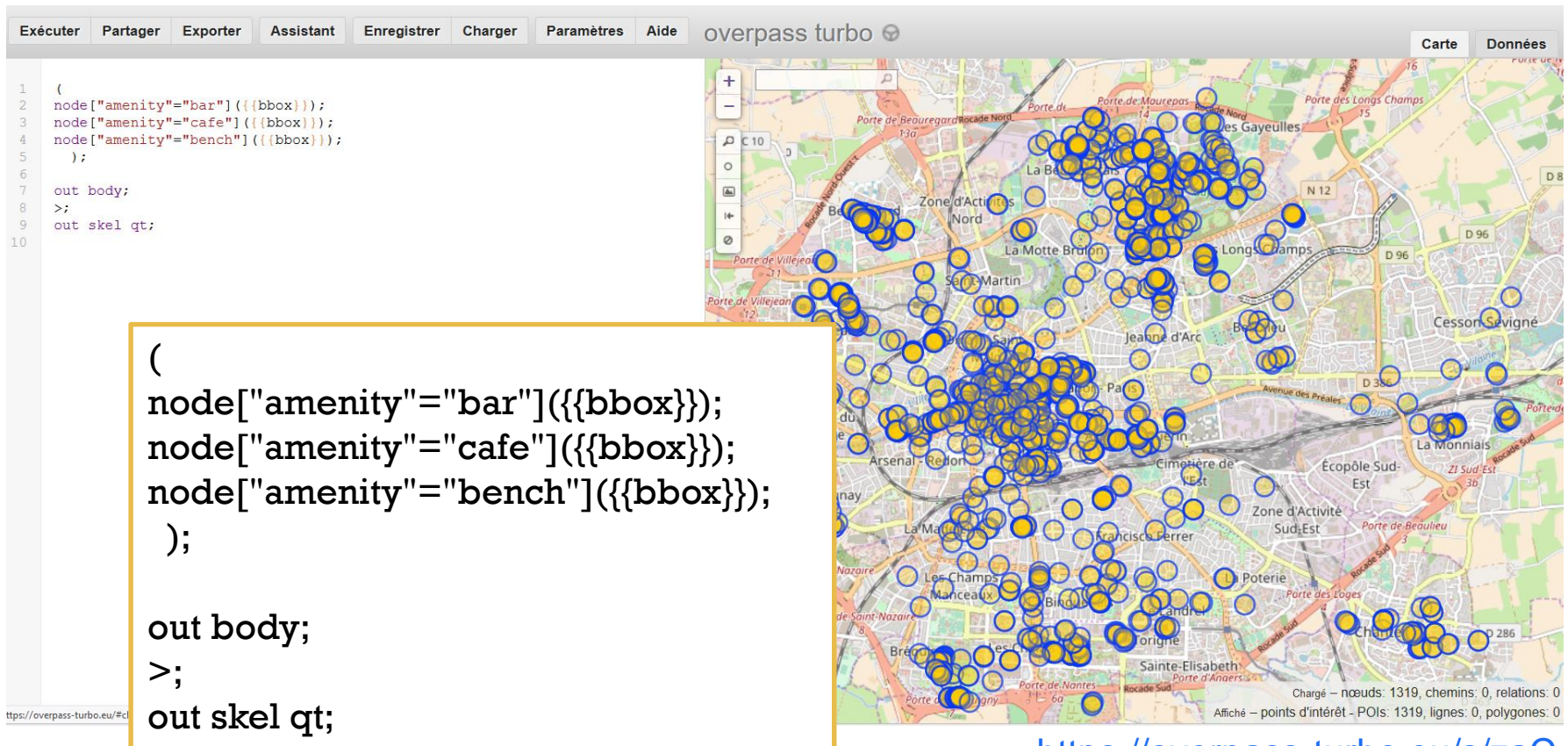
→ Extraire les bars

```
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 plusieurs type d'objets



```
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;

Chargé – nœuds: 1319, chemins: 0, relations: 0  
Affiché – points d'intérêt - POIs: 1319, lignes: 0, polygones: 0

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

# + Requêtes classiques

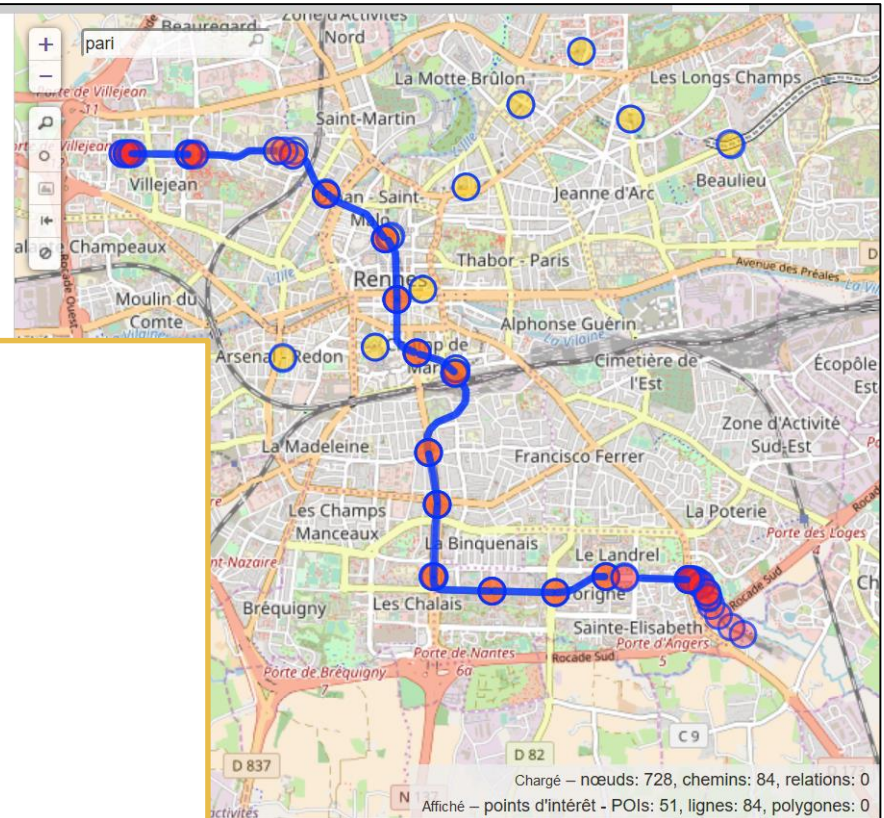
Extraire plusieurs type d'objets

```
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;
```




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


# + Requêtes classiques

## Boundary

### ■ Documentation

<http://wiki.openstreetmap.org/wiki/Boundaries>

 **Feature : Boundaries**



**Description**

Boundaries mark the borders of areas, mostly political, but also of other administrative areas.

**Tags**

`boundary=*`

### Other

#### `boundary=maritime`

for marking maritime borders (rather than land areas normally assumed by `boundary=political`)

#### `boundary=political`

is approved, should be documented in each country where they are used. Can be used

#### `boundary=vice_county`

for marking vice counties in Britain and Ireland [↗](#).

#### `boundary=national_park`

marks the borders of a national park.

#### `boundary=protected_area`

a more recently introduced tag with a more verbose tagging scheme which can deal with

#### `boundary=religious_administration`

*trial* for dioceses, parishes... see [FrViPofm/Tag:boundary=religious\\_administration](#)

#### `boundary=national`

is approved, but not documented, can somebody check tagwatch for usage?

#### `boundary=civil`

is approved, but not documented, can somebody check tagwatch for usage?

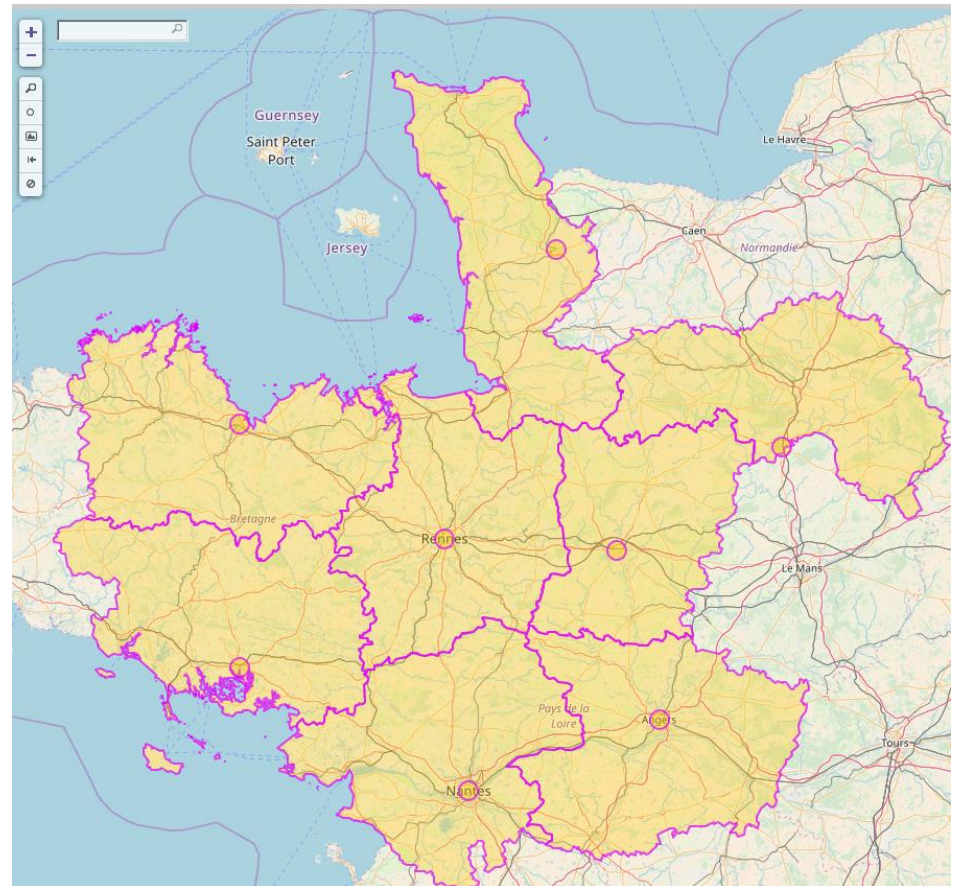
#### `boundary=metropole`

*trial*: When metropolitan areas don't match with an administrative subdivision (some)

# + Requêtes classiques

## Limites administratives

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



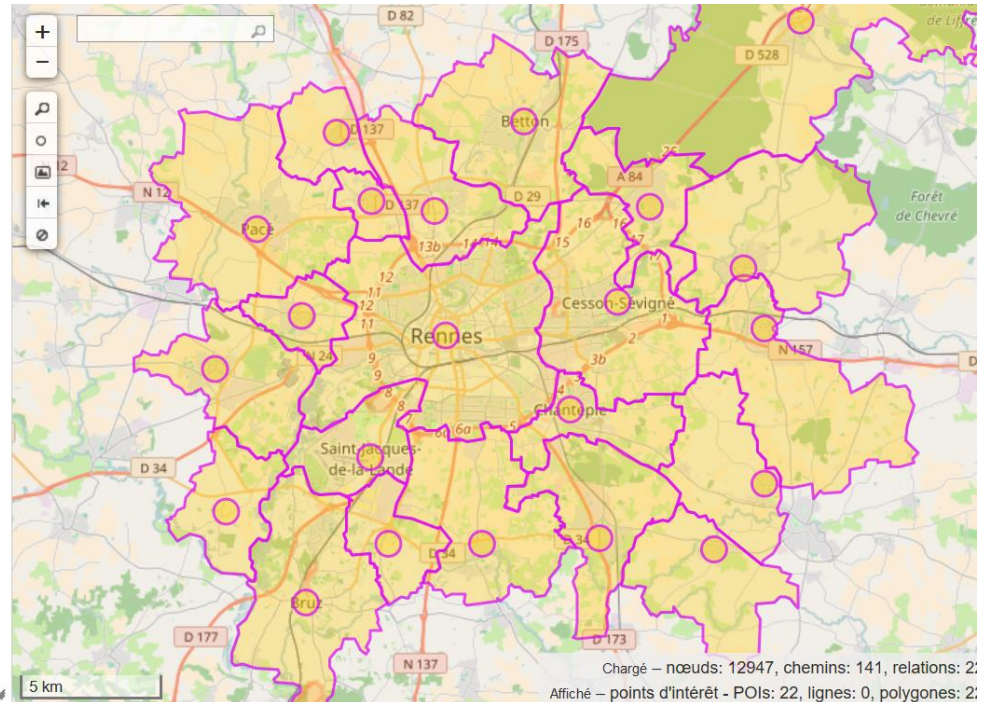
# + Requêtes classiques

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



## Landuse

### ■ Documentation

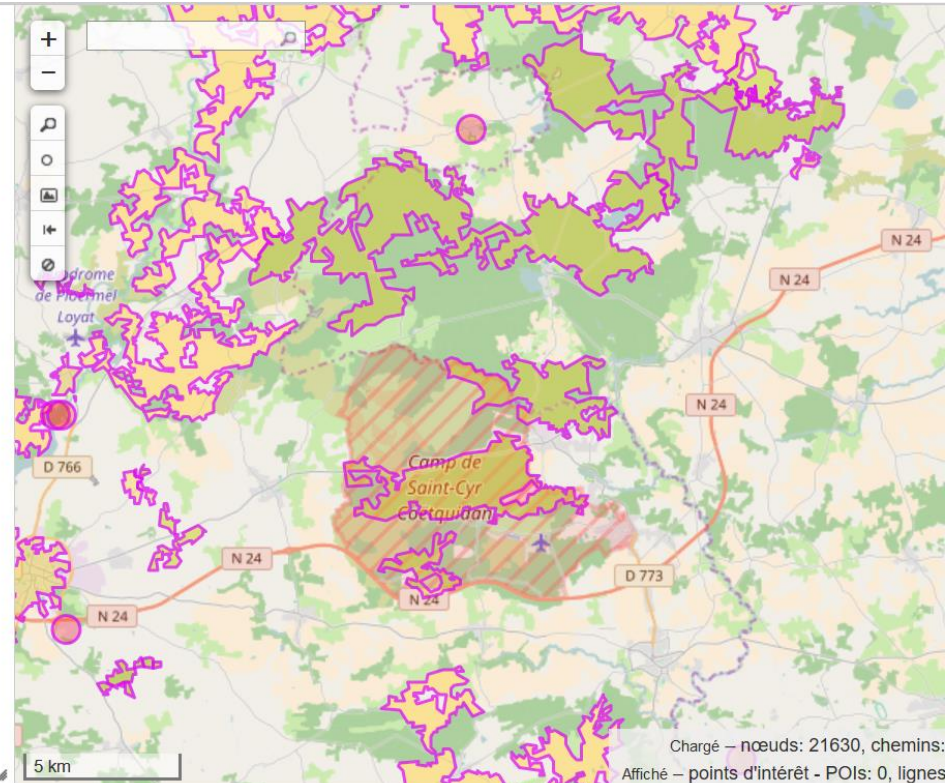
<http://wiki.openstreetmap.org/wiki/FR:Key:landuse>

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

# + Requêtes classiques

Extraire les zones renseignées sur l'occupation des sols

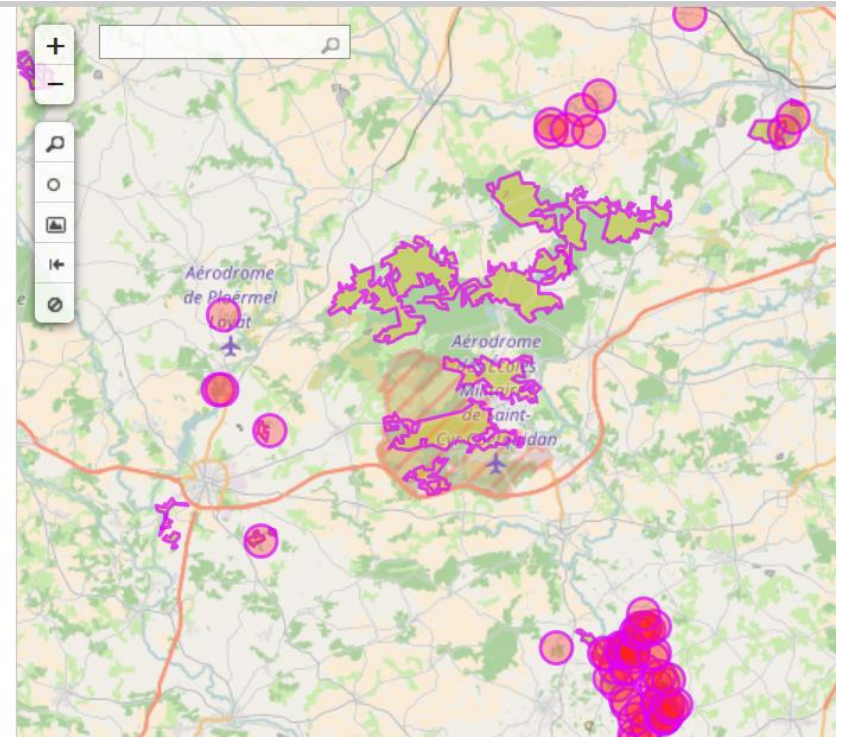
```
1 /*
2 This has been generated by the overpass-turbo wizard.
3 The original search was:
4 "Bar"
5 */
6 [out:json][timeout:25];
7 // gather results
8 (
9 // query part for: "Bar"
10 relation["landuse"]>{{bbox}};
11 );
12 // print results
13 out body;
14 >;
15 out skel qt;
```



# + Requêtes classiques

Extraire les zones renseignées comme forêt

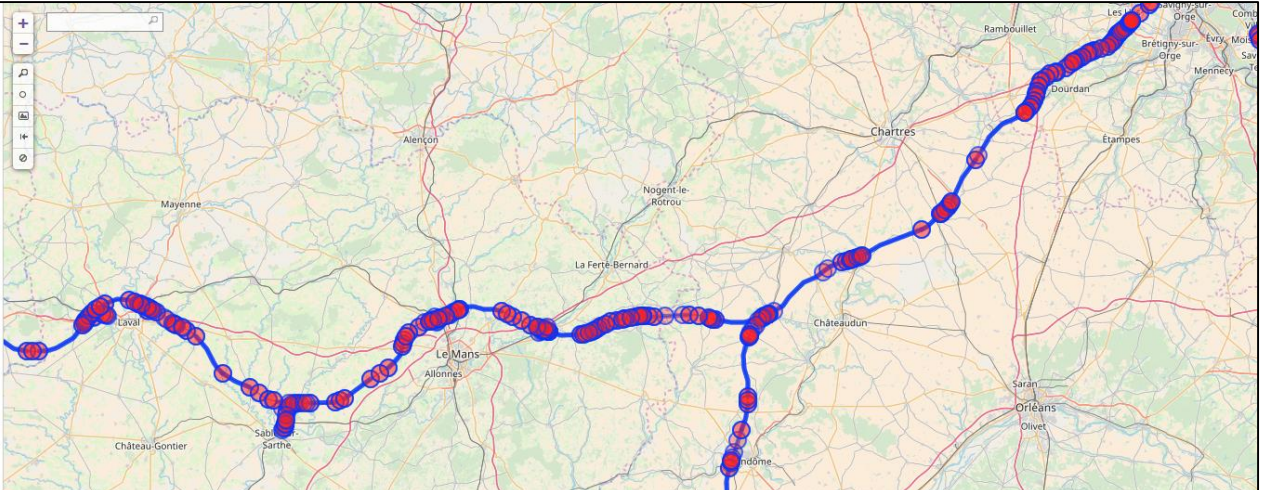
```
/*  
This has been generated by the overpass-turbo wizard.  
The original search was:  
"Bar"  
*/  
[out:json][timeout:25];  
// gather results  
(  
  // query part for: "Bar"  
  relation["landuse"="forest"]({{bbox}});  
);  
// print results  
out body;  
>;  
out skel qt;
```



# + Requêtes classiques

Sélectionner les lignes grandes vitesses

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



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

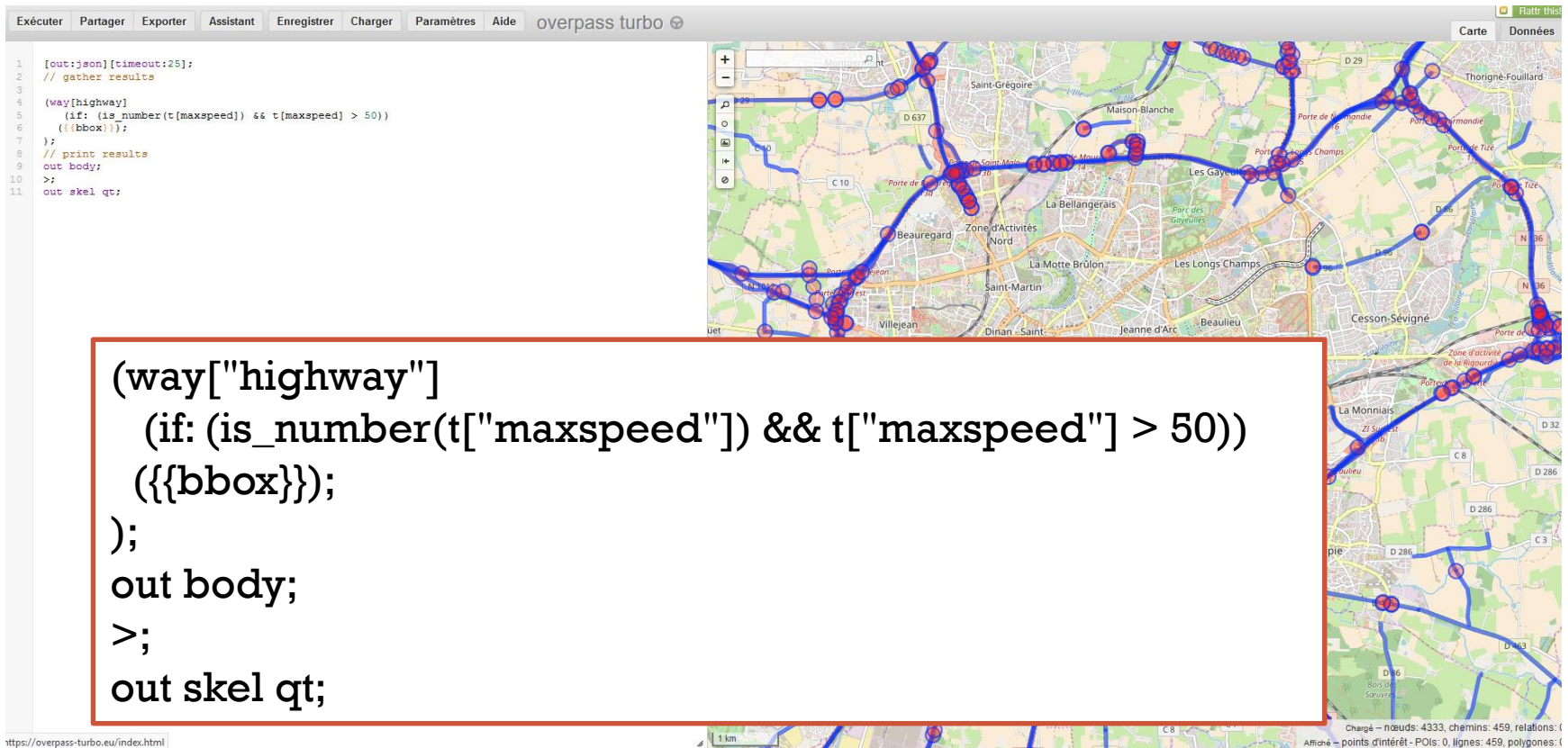
+

**Requêtes complexes**

# + Requêtes complexes

## Critère minimum ou maximum

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



```
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;
```

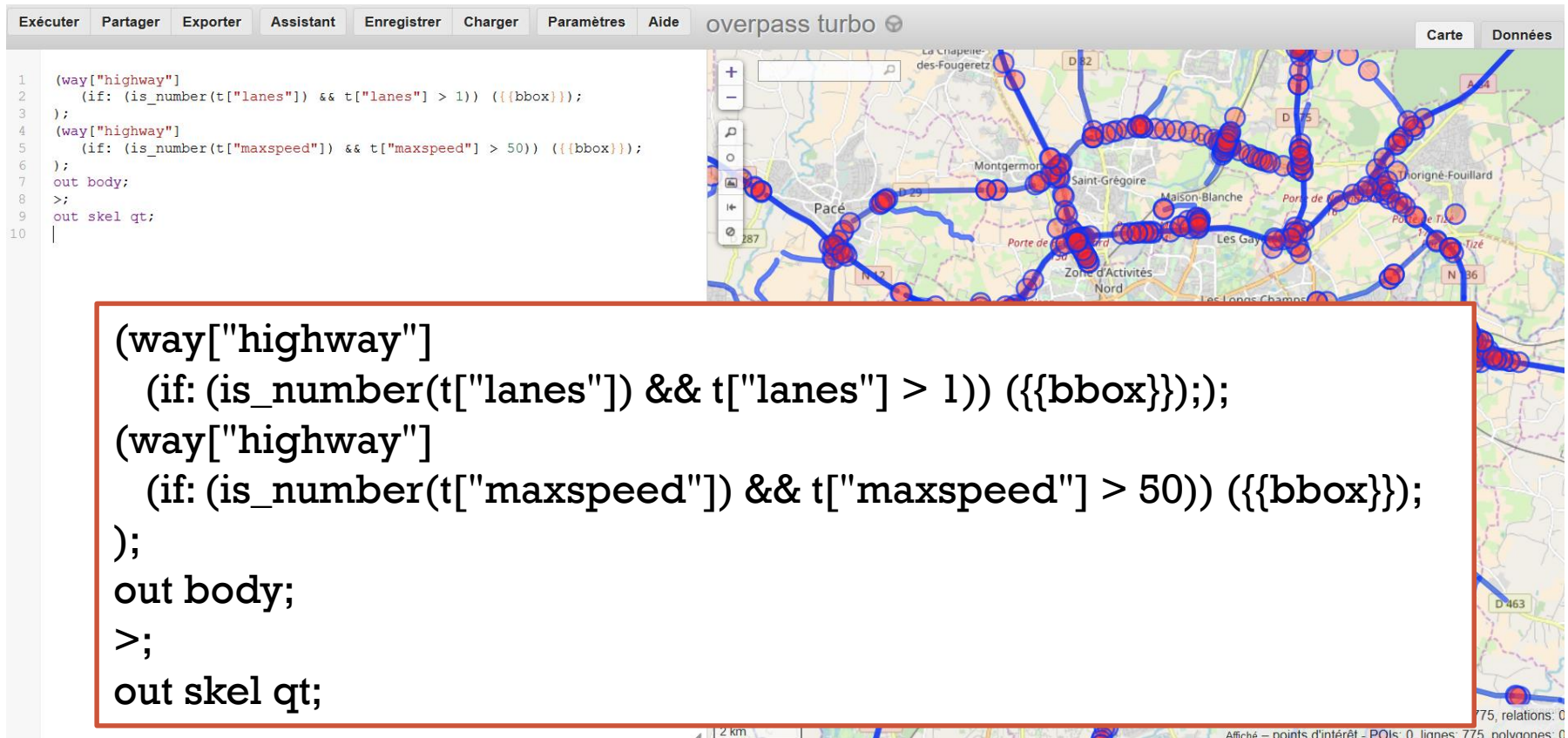
(way["highway"]  
 (if: (is\_number(t["maxspeed"]) && t["maxspeed"] > 50))  
 ({{bbox}});  
);  
out body;  
>;  
out skel qt;

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

# + 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*)



The screenshot shows a web application interface with a map and a code editor. The map displays a network of roads in blue and red, with red circles highlighting specific points of interest. The code editor on the left contains a query that filters for roads with at least two lanes and a maximum speed greater than 50 km/h. A red-bordered box highlights a portion of this query.

```
1 (way["highway"]
2   (if: (is_number(t["lanes"]) && t["lanes"] > 1)) ({{bbox}}));
3 );
4 (way["highway"]
5   (if: (is_number(t["maxspeed"]) && t["maxspeed"] > 50)) ({{bbox}});
6 );
7 out body;
8 >;
9 out skel qt;
10 |
```

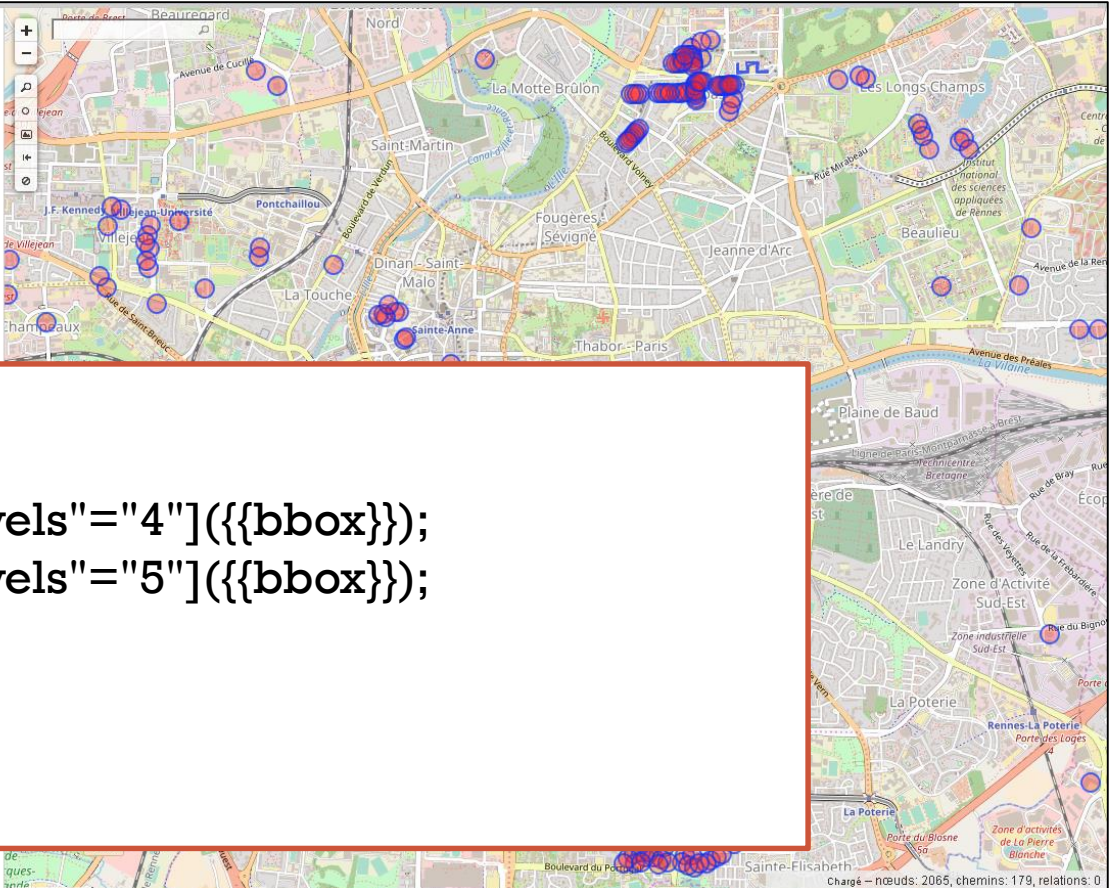
```
(way["highway"]
  (if: (is_number(t["lanes"]) && t["lanes"] > 1)) ({{bbox}}));
(way["highway"]
  (if: (is_number(t["maxspeed"]) && t["maxspeed"] > 50)) ({{bbox}});
);
out body;
>;
out skel qt;
```

775, relations: 0  
Affiché – points d'intérêt – POIs: 0, lignes: 775, polylignes: 0

# + Requêtes complexes

Sélectionner les bâtiments de 4 et 5 étages

```
[out:json][timeout:25];
(
  // query part for: "bar"
  way["building"]["building:levels"="4"]>{{bbox}};
  way["building"]["building:levels"="5"]>{{bbox}};
);
// print results
out body;
>;
out skel qt;
```



The image displays a map of Rennes, France, with numerous buildings highlighted by blue and red circles. The map shows a dense urban area with various streets and landmarks. The highlighted buildings are scattered across the city, with a concentration in the central and eastern parts. The map includes labels for various districts and streets, such as 'Beauregard', 'Nord', 'La Motte Brûlon', 'Saint-Martin', 'Fougères-Sévière', 'Jeanne d'Arc', 'Beaulieu', 'Plaine de Baud', 'Le Landry', 'Zone d'Activité Sud-Est', 'La Poterie', 'Rennes La Poterie Portes des Loges', and 'Sainte-Elisabeth'. The map also shows the 'Avenue de la République' and 'Avenue des Prêtres'. The map is overlaid with a grid and has a scale bar at the bottom right.

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



# + 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
```

# + Comptage d'entités



## Afficher des statistiques sur les routes de Rennes

```
Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo
```

```
1 [out:csv(:"count", ::"count:nodes", ::"count:ways", ::"count:relations")][timeout:25];
2 {{geocodeArea:Rennes}}->.searchArea;
3 {
4   node["highway"] (area.searchArea);
5   way["highway"] (area.searchArea);
6   relation["highway"] (area.searchArea);
7 };
8 out count;
```

|   | @count | @count:nodes | @count:ways | @count:relations |
|---|--------|--------------|-------------|------------------|
| 1 | 23862  | 6792         | 17049       | 21               |
| 2 |        |              |             |                  |
| 3 |        |              |             |                  |

## Afficher des statistiques sur les bars de Rennes

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

|   | @count | @count:nodes |
|---|--------|--------------|
| 1 | 112    | 112          |
| 2 |        |              |
| 3 |        |              |

# + Contrôle données

## Éléments sans tags

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

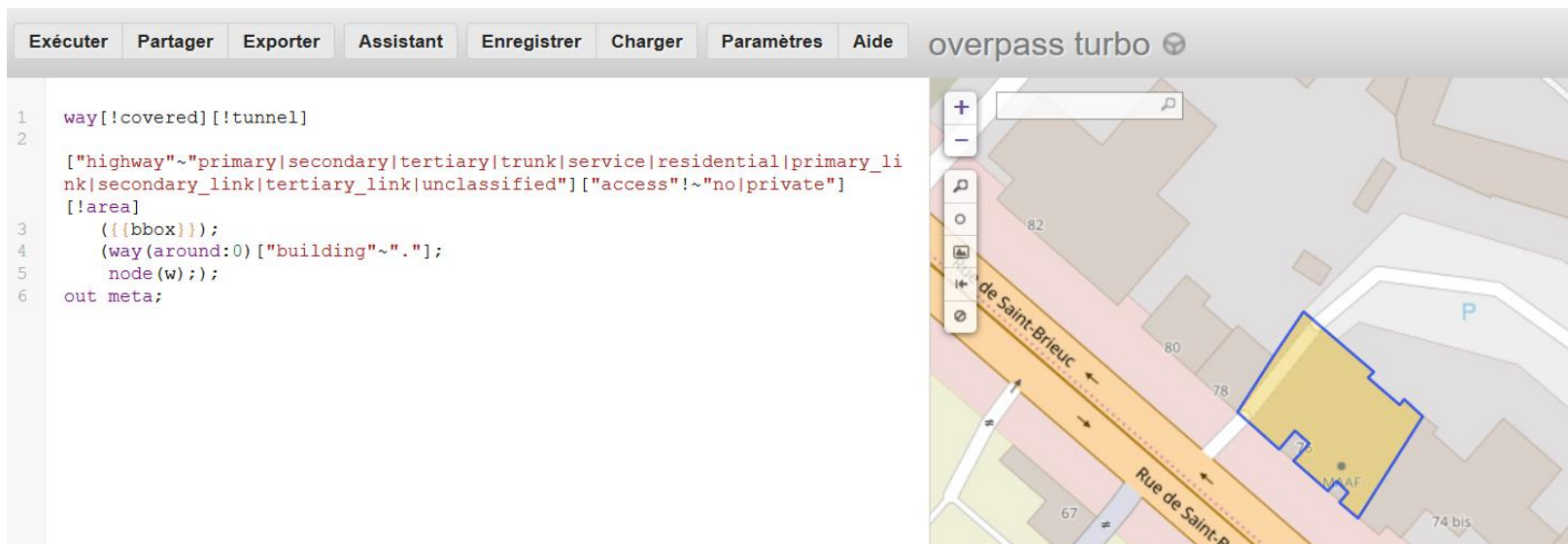
```
1 way({{bbox}})(if:count_tags() == 0);  
2 out geom;  
3
```

Way 441800913

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

# + Contrôle données

## Superposition bâtiments / routes



The screenshot shows the OpenStreetMap Overpass Turbo interface. The top navigation bar includes buttons for 'Exécuter', 'Partager', 'Exporter', 'Assistant', 'Enregistrer', 'Charger', 'Paramètres', and 'Aide'. The title of the page is 'overpass turbo'. On the left, a code editor contains the following query:

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

On the right, a map view shows a street intersection with 'Rue de Saint-Brieuc' and 'Rue de Saint-P'. A yellow building is highlighted with a blue outline. The map includes a search bar, zoom controls, and a layer selector.

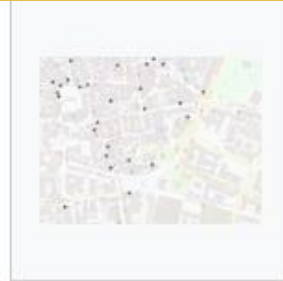
```
way
["highway"~"primary | secondary | tertiary | trunk | service | residential | primary_link |
secondary_link | tertiary_link | unclassified"]["access"!~"no | private"][!area]
({{bbox}});
(way(around:0)["building"~"."]);
node(w););
out meta;
```



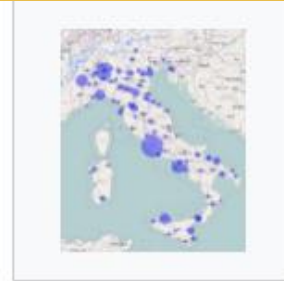
line styles



color coding



icons



a simple thematic map



markers with text

+

# Requêtes et style

[https://wiki.openstreetmap.org/wiki/Overpass\\_turbo/MapCSS](https://wiki.openstreetmap.org/wiki/Overpass_turbo/MapCSS)



# Requête et style

## Afficher des étiquettes

overpass turbo

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide

```
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;
```

Carte Données

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

`{{style:`

`node { text: name;}`

`}}`



# 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; }
```

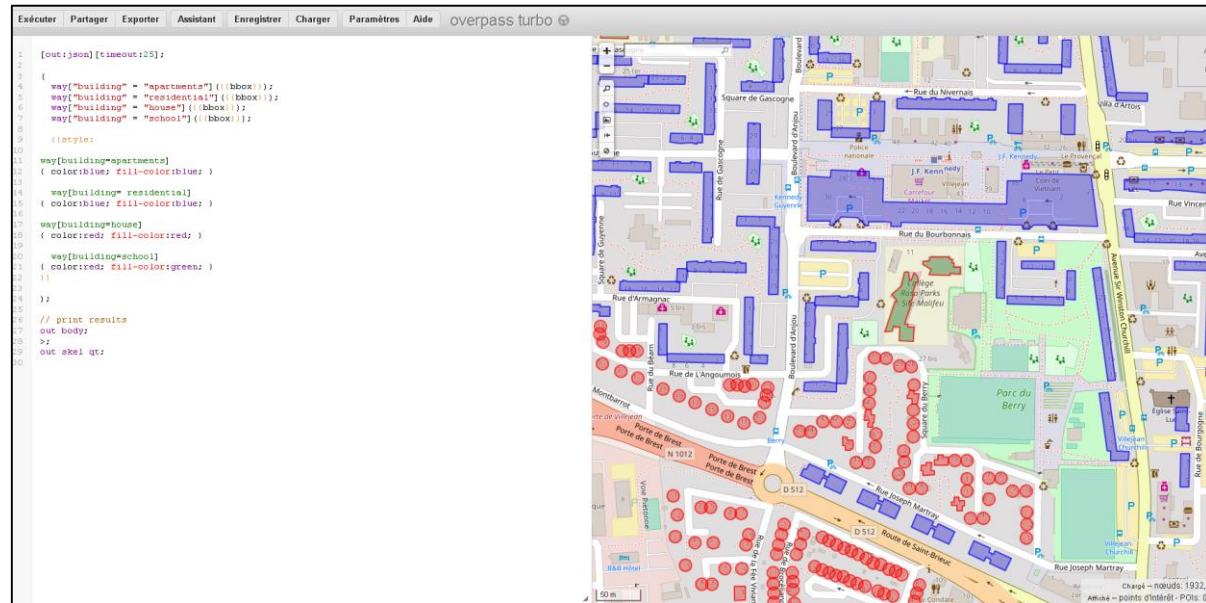
```
way[building= residential]
{ color:blue; fill-color:blue; }
```

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

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

```
);
```

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



# + Requête et style

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);

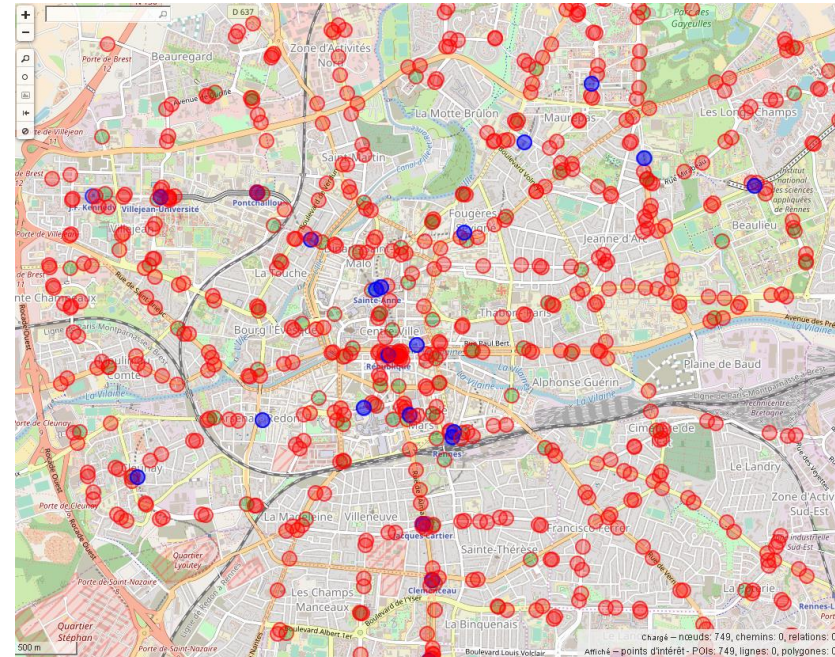
{{style:

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



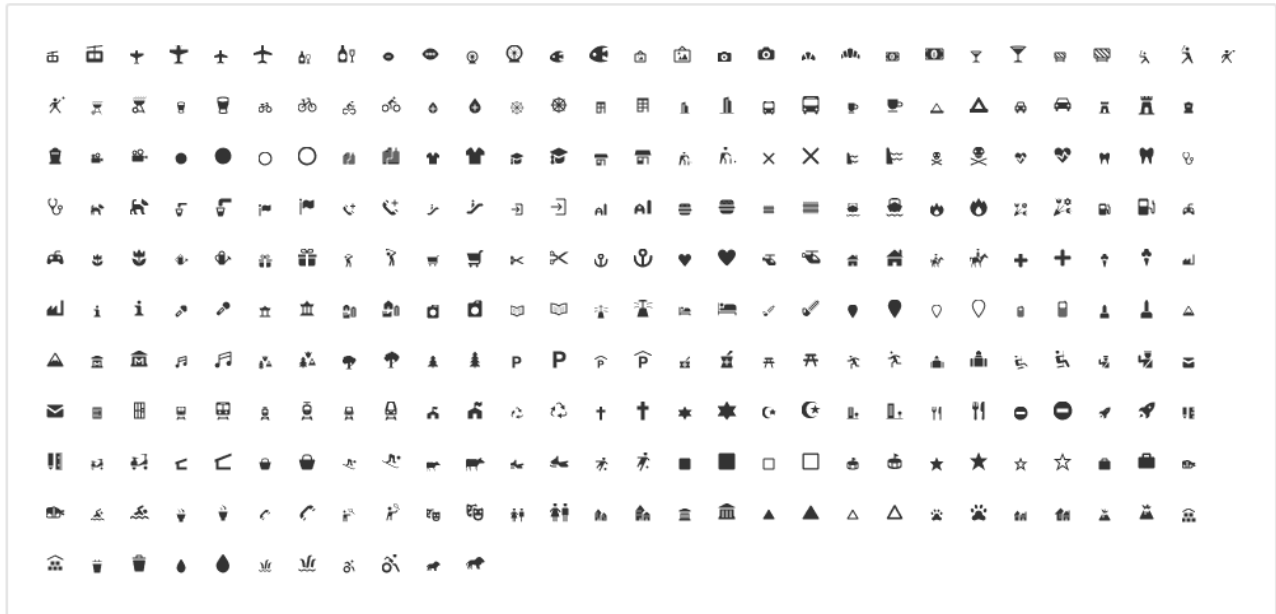
Utiliser des icones (maki)

<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)

Exécuter Partager Exporter Assistant Enregistrer Charger Paramètres Aide overpass turbo

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

**Map Labels:** Porte de Villejean, Avenue Charles Tillon, Villejean Université, Université de Rennes, J.F. Kennedy, Pontchaillou, Rennes-pontchaillou, La Touche, Bourg, l'Évesque.

**Map Statistics:** - nœuds: 745, chemins: 0, relations: 0  
ét - POIs: 745, lignes: 0, polygones: 0

```
1 {{style:
2
3 node[highway=bus_stop] {
4   icon-image: url('icons/maki/bus-18.png');
5   icon-width: 18;}
6
7 }}
8
```



# 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

```
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
```

Carte Données

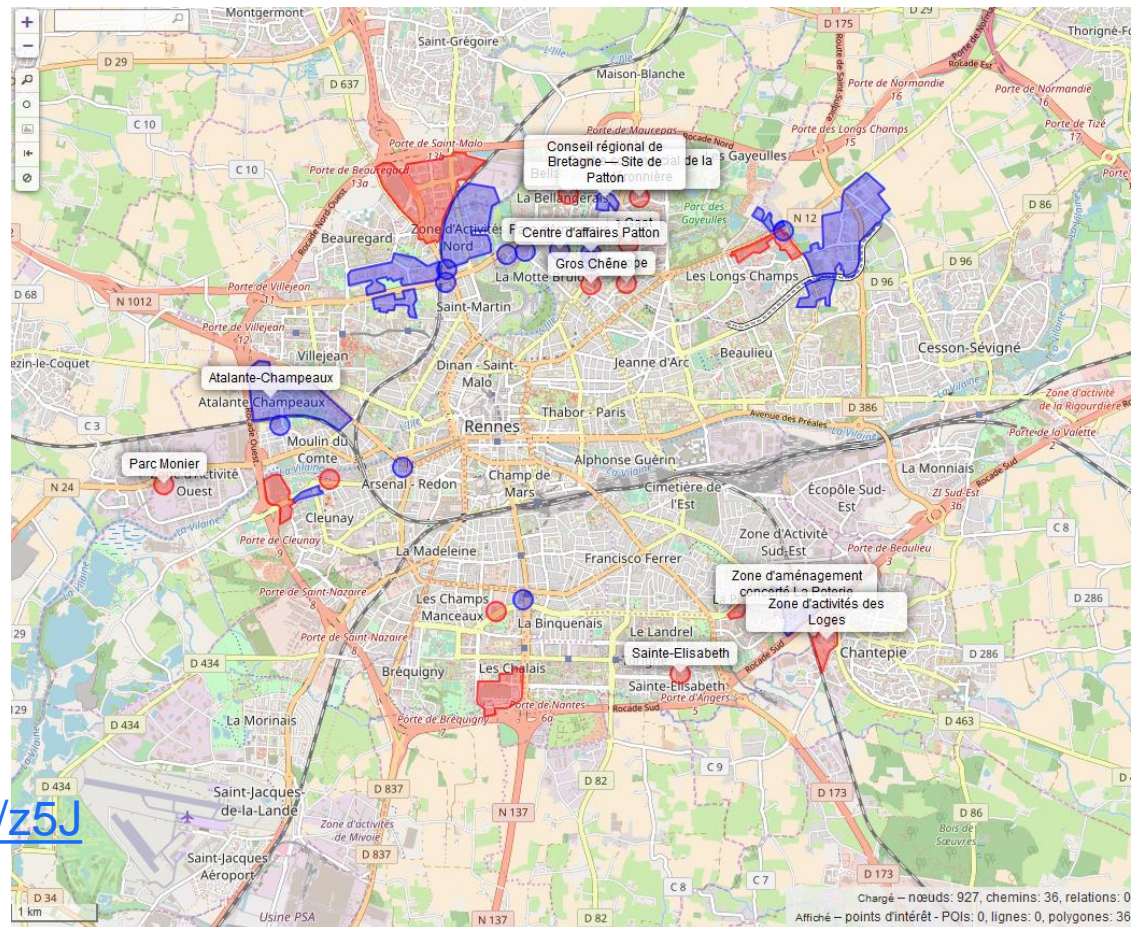
Chargé – nœuds: 75447, chemins: 3024, relations: 65  
Affiché – points d'intérêt - POIs: 490, lignes: 3089, polygones: 0



# Requête et style



Sélectionner les zones commerciales avec un affichage stylisé et des étiquettes



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

**+ Requêtes  
contributeurs / date**



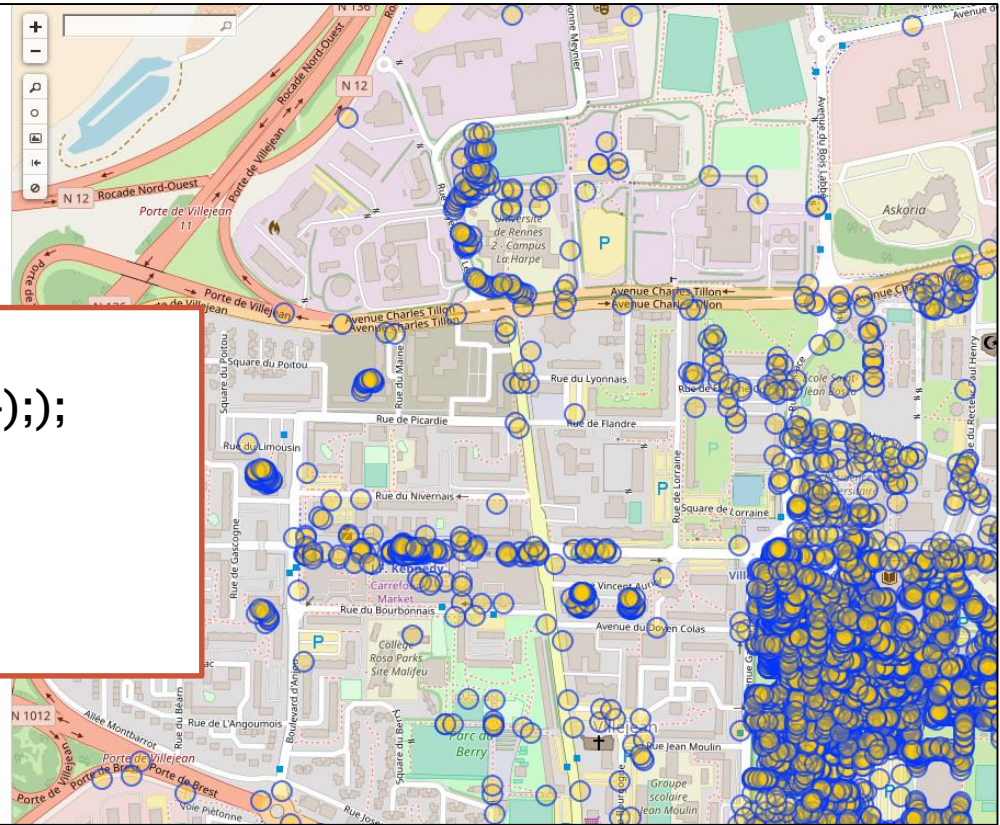
# Extraction par contributeur



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

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

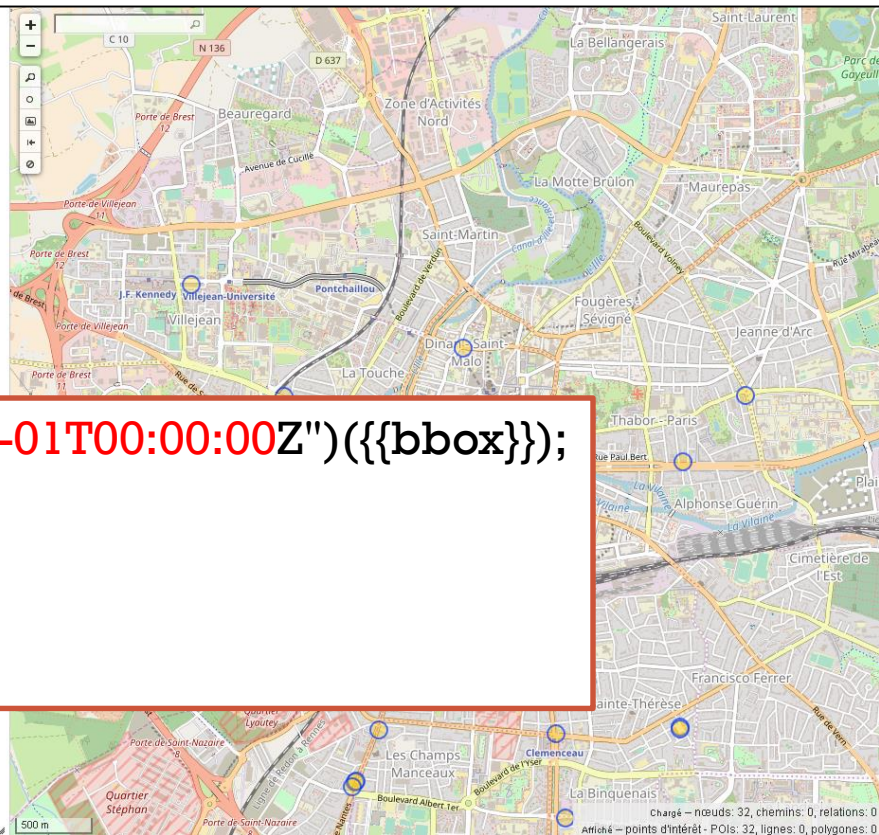
```
out body;  
>;  
out skel qt;
```



# + Extraction par date

Sélectionner les nouvelles aménités depuis le 1<sup>er</sup> mai 2018

```
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:"**2018-05-01T00:00:00Z**")({{bbox}});  
);  
out body;  
>;  
out skel qt;

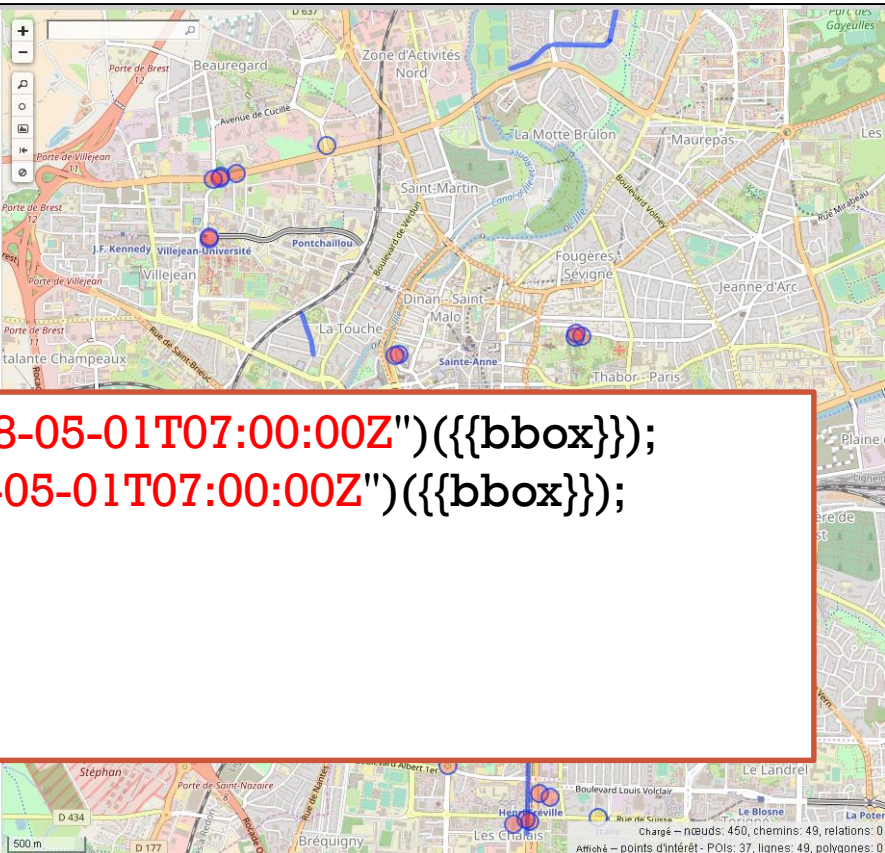
Chargé – nœuds: 32, chemins: 0, relations: 0  
Affiché – points d'intérêt - POI: 32, lignes: 0, polygones: 0



# + Extraction par date

Sélectionner les routes mises à jour depuis le 1<sup>er</sup> mai 2018

```
[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:"2018-05-01T07:00:00Z") ({{bbox}});
  way["highway"](changed:"2018-05-01T07:00:00Z") ({{bbox}});
);
out body;
>;
out skel qt;
```

+

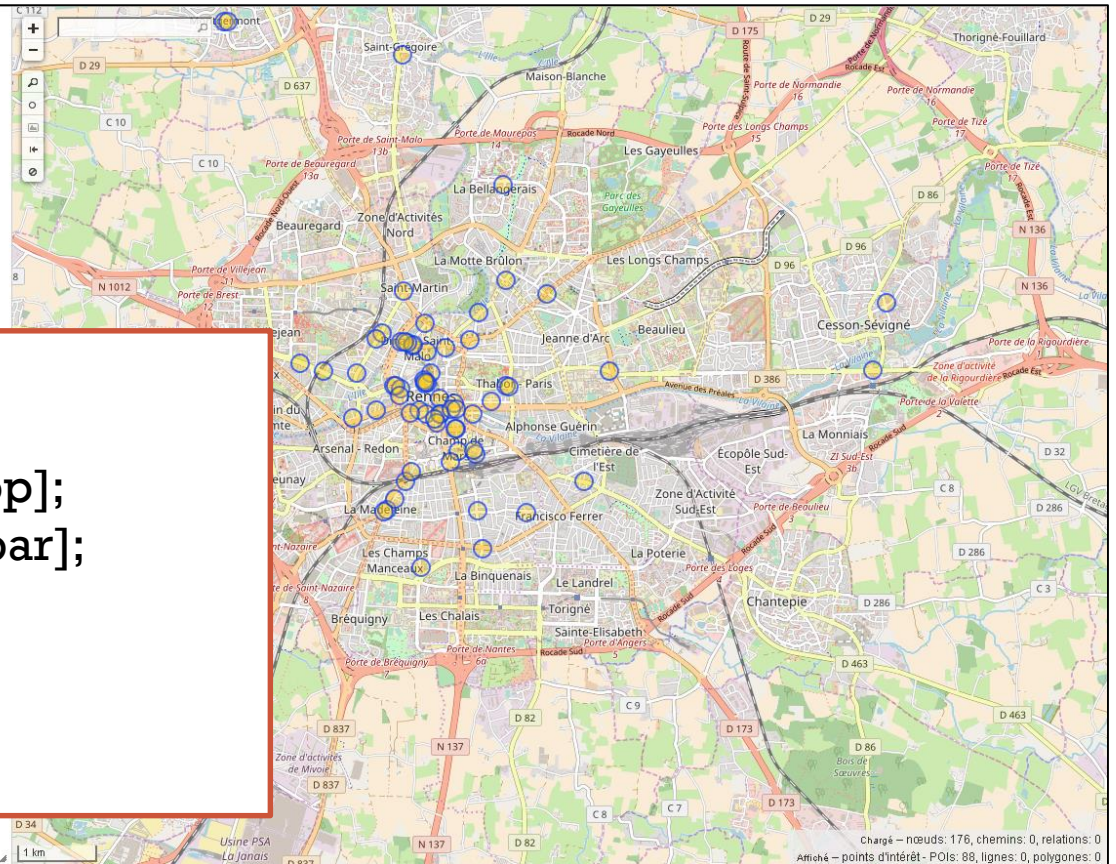
**Requêtes spatiales**

# + 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 |
```

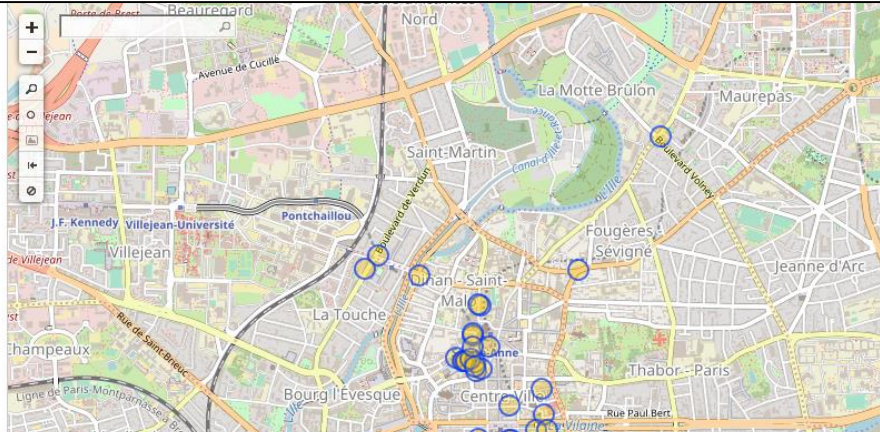
```
[out:json][timeout:25];  
area[name="Rennes"];  
node(area)[highway=bus_stop];  
node(around:100)[amenity=bar];  
out;  
out body;  
>;  
out skel qt;
```



# + 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;
```



```
[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;
```

# + 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 };
```

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

Carte Données

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

# + 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;
```

