

## Jour #9 - L'instruction map\_to\_area

Un exercice du tuto précédent consistait à trouver les sentiers à l'intérieur du Parc départemental Georges-Valbon. Il n'existe qu'**un seul parc au monde** avec ce nom, la requête est donc simple.

Mais qu'en est-il si vous cherchez des éléments se trouvant dans le Jardin des Plantes de votre commune. La requête suivante montre qu'il y a dans la base OpenStreetMap une vingtaine de parcs nommés "Jardin des Plantes"...

```
area
  [leisure=park]
  [name="Jardin des Plantes"];
out;
```

Pour trouver les sentiers se trouvant dans le Jardin des Plantes d'une commune, la requête doit donc dans un premier temps trouver le parc nommé Jardin des Plantes se trouvant dans la commune, puis les sentiers se trouvant dans ce parc. On pourrait donc être tenté d'écrire, par exemple pour Nantes :

```
area
  ["ref:INSEE"=44109]->.nantes;
area
  [leisure=park]
  [name="Jardin des Plantes"]
  (area.nantes)->.parc;
way
  [highway=footway]
  (area.parc);
out geom;
```

Malheureusement la forme `area (area)`, c'est-à-dire la recherche d'une surface dans une surface, n'est pas supportée par l'API Overpass. Il est nécessaire de chercher les parcs sous forme de ways ou de relations. Or la recherche « à l'intérieur de » nécessite d'utiliser une surface...

L'instruction `map_to_area` vient à la rescousse. Cette instruction produit un lot de données contenant des surfaces sous forme de `area`, à partir des données OSM décrivant une surface sous forme de `way` fermé ou de relation (les autres éléments sont ignorés).

Ainsi dans la requête suivante, l'instruction `map_to_area` remplace le lot de données courant (qui contient un `way`) par un lot de données qui contient la surface correspondant au Jardin des Plantes. L'instruction qui suit trouve les chemins piétons à l'intérieur de cette surface.

```
area["ref:INSEE"=44109]->.nantes; // Nantes
way
  [leisure=park]
  [name="Jardin des Plantes"]
  (area.nantes);
map_to_area;
way
  [highway=footway]
```

```
(area);  
out geom;
```

Cette requête fonctionne mais il est possible que le parc soit un jour converti en relation. Il convient donc de considérer ways et relations :

```
area["ref:INSEE"=44109]->.nantes;  
(  
  way  
    [leisure=park]  
    [name="Jardin des Plantes"]  
    (area.nantes);  
  rel  
    [leisure=park]  
    [name="Jardin des Plantes"]  
    (area.nantes);  
)->.parcs_wr;  
.parcs_wr map_to_area -> .parcs_a;  
way  
  [highway=footway]  
  (area.parcs_a);  
out geom;
```

Notez l'utilisation de variables autour de l'instruction `map_to_area`, qui prend le lot de données `.parcs_wr`, trouve les surfaces correspondantes, et place le résultat (avec `->`) dans le lot de données `.parcs_a`. L'utilisation de ces deux variables n'est pas nécessaire ici, mais permet d'explicitier les lots de données en les nommant plutôt que d'utiliser le concept un peu abstrait de « lot de données courant ».

## Exercices

- Trouvez les gares (`building=train_station`) de Paris
- Trouvez les commerces dans ces gares
- Trouvez les points d'eau potable dans les cimetières de Paris
- Trouvez les restaurants dans les communes d'Île-de-France dont le nom finit par "Marne"

© CC-by-sa Carto'Cité

From:

<http://wiki.cartocite.fr/> -



Permanent link:

[http://wiki.cartocite.fr/doku.php?id=tutoverpass:jour\\_9\\_l\\_instruction\\_map\\_to\\_area&rev=1611591290](http://wiki.cartocite.fr/doku.php?id=tutoverpass:jour_9_l_instruction_map_to_area&rev=1611591290)

Last update: 2021/01/25 17:14

