

Contenu et format du résultat

Le résultat d'une requête Overpass est contrôlé à deux niveaux :

- la clause **[out:]** dans l'entête de la requête définit le format de sortie
- l'instruction **out** dans le corps de la requête définit le contenu

Format de sortie

Le format de sortie par défaut est le format XML avec la syntaxe OSM. La clause `[out:xml]` est donc redondante. Les autres formats courants sont :

- JSON : utile pour les développeurs Web.
- CSV : utile pour importer le résultat dans un tableur

Ainsi la clause **[out:json];** produira un résultat au format JSON.



Attention ! il ne s'agit pas du format GeoJSON mais d'une simple transcription de XML vers JSON.

La [bibliothèque `osm2geojson.js`](#) permet de convertir des données OSM (format XML ou JSON) au format GeoJSON. Il s'agit du code utilisé par Overpass Turbo, il peut être utilisé

Export CSV

La clause **[out:csv]** définit les clefs que l'on souhaite exporter dans le résultat. Cette clause est composée de 3 sections dont seule la première est obligatoire :

```
[out:csv( clefs [; titres? [; séparateur ] ] )]
```

Section	Description	Remarques
clefs	liste de clefs séparées par une virgule	placer entre guillemets les noms utilisant un namespace: "name:fr"
titres?	true pour produire les noms de colonnes	la valeur par défaut est true
séparateur	caractère de séparation des valeurs	le séparateur par défaut est la tabulation, placer le caractère entre guillemets : ";"

Exemple de requête produisant un résultat au format CSV :

```
[out:csv(name, cuisine, "contact:website", "contact:phone")];  
node["amenity"="restaurant"]({{bbox}});
```

```
out;
```

Des noms de clefs réservés permettent d'exporter des méta-données ainsi que la géométrie. Parmi [cette liste](#) citons :

- **::lat** et **::lon** : coordonnées géographiques d'un point
- **::version** : numéro de version de l'objet
- **::user** : utilisateur OSM ayant créé cette version
- **::changeset** : identifiant du changeset de cette version

Même requête modifiée pour exporter les coordonnées géographiques en utilisant comme séparateur le point-virgule :

```
[out:csv(name,cuisine,"contact:website","contact:phone",::lat,::lon; true;";");  
node["amenity"="restaurant"]({{bbox}});  
out;
```

Bien sûr de nombreux objets OSM ne sont pas toujours représentés par un node. Intéressons-nous au contenu du résultat d'une requête.

Contenu du résultat

L'instruction **out** ajoute le lot de données *courant* au résultat de la requête. Elle est généralement placée à la fin d'une requête, mais rien n'empêche de l'inclure plusieurs fois dans une même requête.

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

Il est aussi possible d'exporter le contenu d'une variable. La requête suivante est équivalente à la précédente :

```
node[amenity=restaurant]({{bbox}})->.restos;  
node[amenity=fast_food]({{bbox}});  
out;  
.restos out;
```

[De nombreuses options](#) permettent de contrôler le contenu du résultat. Voici quelques exemples utiles :

Clause	Description	Exemple d'usage
tags	N'exporte que les tags des objets, sans leur géométrie. Le résultat ne permet pas de produire une carte mais peut suffire pour un résultat en CSV	<pre>[out:csv(shop, opening_hours, wheelchair)]; node[shop]({{bbox}}); out tags;</pre>

Clause	Description	Exemple d'usage
center	Convertit chaque élément en un point, permet de gérer des données hétérogènes nodes/ways. Très utile produire un résultat au format CSV.	<pre>(node[tourism=hotel]({{bbox}}); way[tourism=hotel]({{bbox}});); out center;</pre>
geom	Ajoute la géométrie des éléments, notamment les coordonnées sur les nodes référencés par un way. Alternative intéressante à la forme récursive out; >; out; pour ne pas inclure les tags sur les nodes utilisés par les ways, par exemple les passages piétons des rues.	<pre>way[highway=cycleway]({{bbox}}); out geom;</pre>
(<bbox>)	Effectue un clipping partiel du résultat : pour les ways, seuls les nodes à l'intérieur du rectangle, ainsi que les noeuds adjacents en dehors du rectangle, sont conservés. Cette forme permet d'extraire la partie d'un itinéraire (transport en commun, randonnée) à l'intérieur d'un rectangle.	<pre>way[highway]({{bbox}}); out geom ({{bbox}}); rel[type=route][name="La Loire à Vélo"]; out geom ({{bbox}});</pre>

La forme suivante, produite par l'assistant, s'explique ainsi :

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

1. export des éléments produits par la requête
2. récursion pour obtenir les éléments référencés notamment les nodes
3. export des éléments référencés avec leur géométrie mais sans leurs tags (skel pour *skeleton* et qt pour *quantile*, proche d'une indexation géographique)

From:

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

Permanent link:

http://wiki.cartocite.fr/doku.php?id=openstreetmap:overpass:api_contenu_et_format_du_resultat&rev=1574867961

Last update: **2019/11/27 16:19**