

# OpenStreetMap

OSM for wikipedians  
Build a thematic map

Erik Zachte

(wp:nl, Oct 10, 2020 + extra notes for offline reading)

## Presentation notes

October 10, 2020 this presentation was part of an OpenStreetMap (OSM) workshop for Dutch wikipedians, by Dutch mappers.

Topics were **OSM hands-on editing with JOSM** by **Luciën Greefkes**, **OSM's mission** by **Milo van der Linden**, and **How to build thematic maps for Wikipedia with Overpass** by me.

With a few extra presentation notes added, this pdf may hopefully serve as a self-study guide. It's available on OSM wiki at page [https://wiki.openstreetmap.org/wiki/Overpass\\_introduction](https://wiki.openstreetmap.org/wiki/Overpass_introduction)

Main focus is on OSM tool Overpass, but not exclusively.

Examples are often about The Netherlands, but some have a wider scope. Similar thematic maps (especially the ones about energy), could be used in different countries and regions as well.

Enjoy!

**Erik Zachte**  
aka Infodisiac



**OSM Query feature**

**OpenPoiMap**

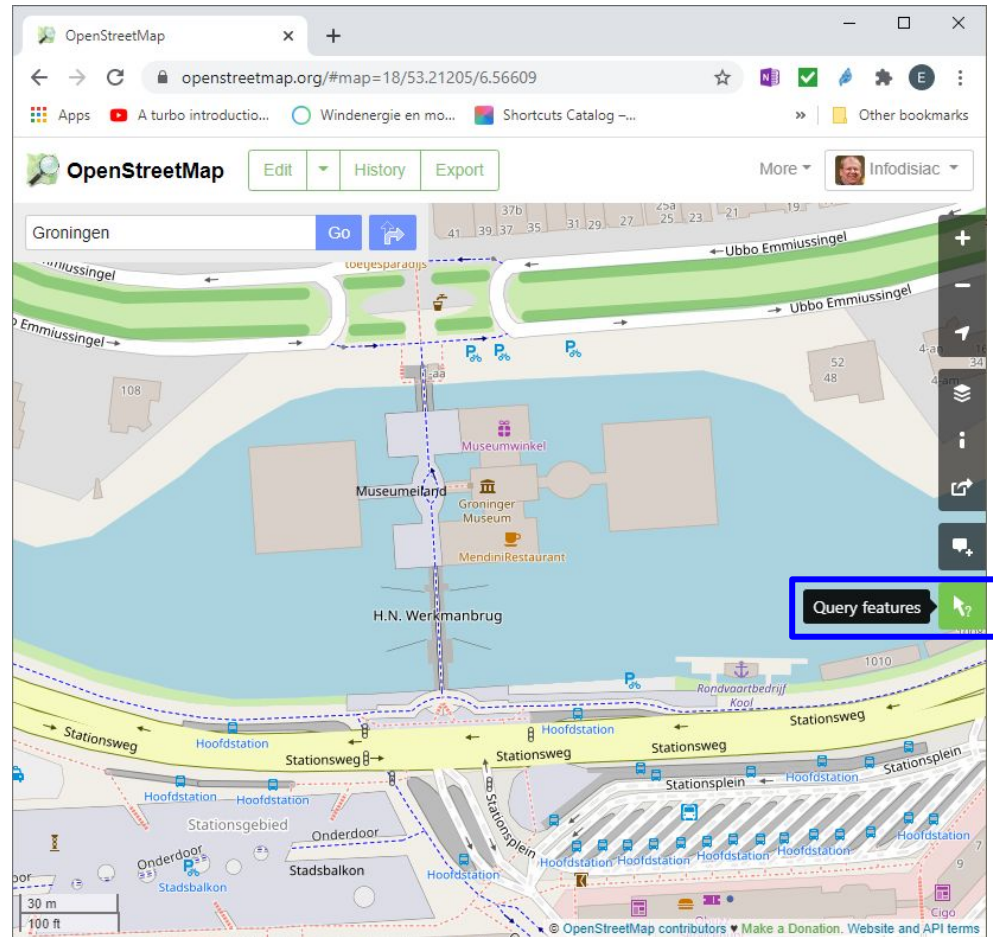
**Overpass**

**Example Maps  
UI and basic syntax**

**Other tools**

Many ways to query OpenStreetMap

Simplest is **Query features**



<https://www.openstreetmap.org/#map=18/53.21228/6.56601>

# Query Features

## Nearby features

Water #94068198

Water #94695506

Water #301638508

Bridge Museumeiland

Pedestrian Way Museumeiland

Way #625729579

Way #625729583

Way #781834722

Way #781834723

Bridge #781834724

Cycle Path Museumeiland

Gift Shop Museumwinkel

Relation 86-86

Public Building Groninger Museum

Public Building Groninger Museum

Relation Pieterpad deel 1 - 02 - Winsum-Groningen

Relation 30-96

Search Where is this? Go

Relation: Groninger Museum (3614207)

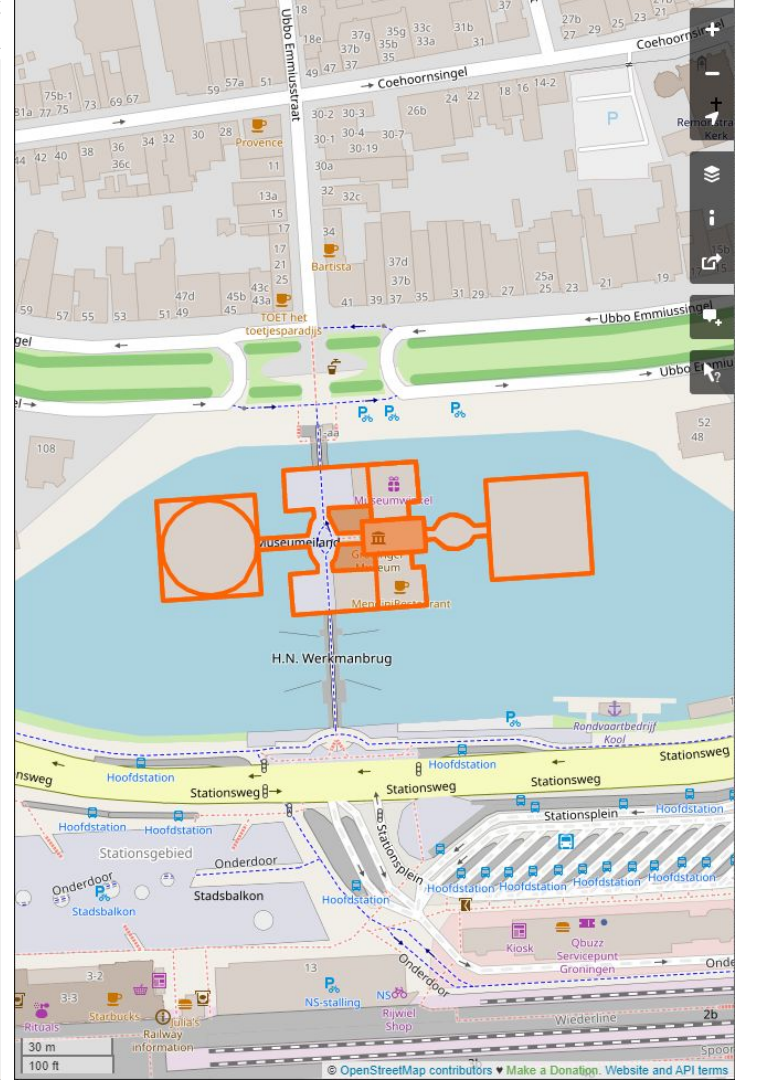
3d mapping

Edited almost 2 years ago by Martien Sch

Version #12 - Changeset #62701654

## Tags

|                       |                     |
|-----------------------|---------------------|
| addr.city             | Groningen           |
| addr.housenumber      | 1                   |
| addr.postcode         | 9711ME              |
| addr.street           | Museumeiland        |
| building              | public              |
| building:architecture | postmodern          |
| building:levels       | 7                   |
| fee                   | MJK_free            |
| name                  | Groninger Museum    |
| opening_hours         | Tu-Su 10:00-17:00   |
| ref.bag               | 14100010910697      |
| roof.colour           | gray                |
| roof.shape            | flat                |
| source                | BAG                 |
| source:date           | 2014-03-24          |
| start_date            | 1994                |
| tourism               | museum              |
| type                  | building            |
| wikidata              | Q1542668            |
| wikipedia             | nl:Groninger Museum |





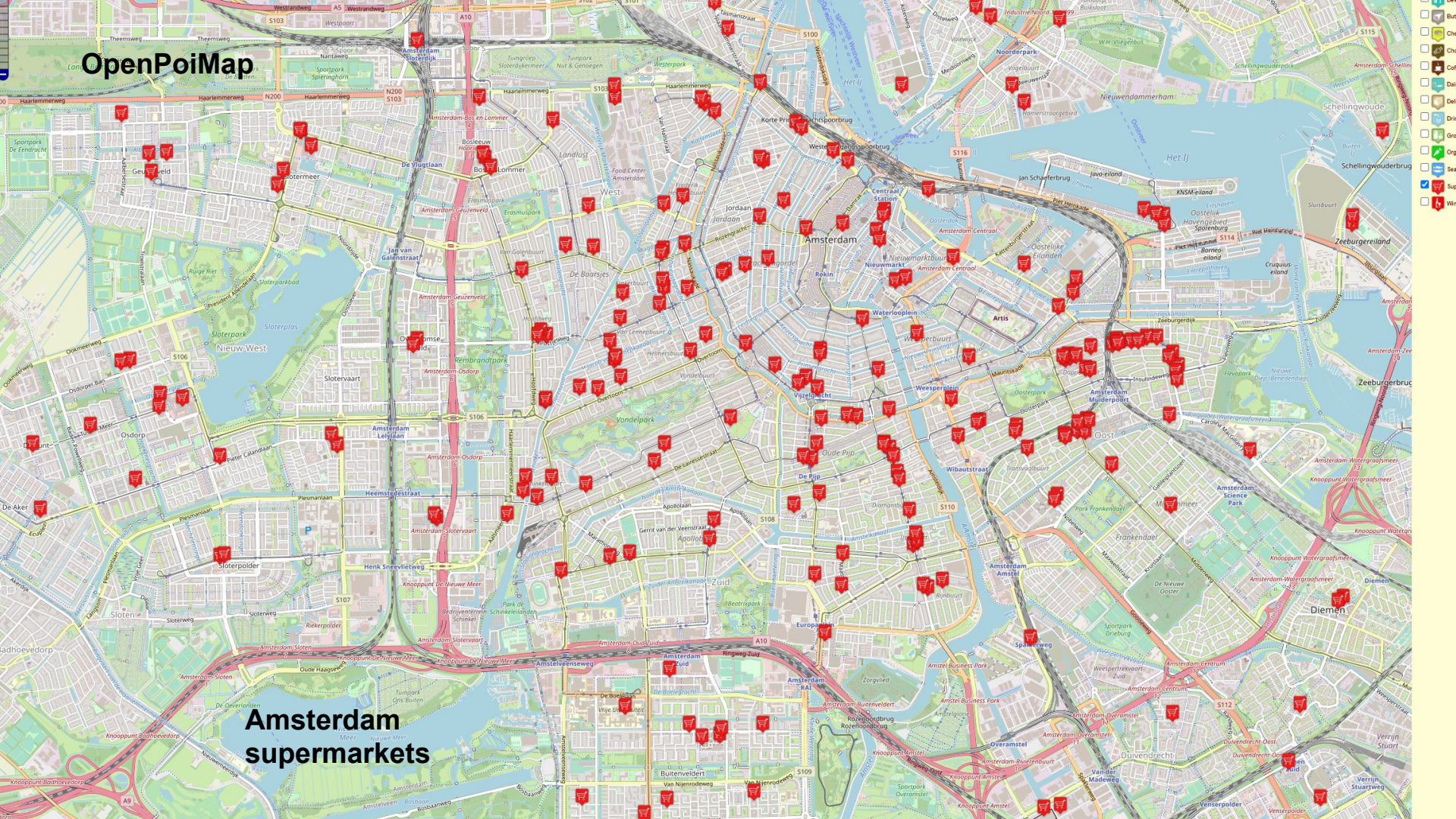
Also very simple: [OpenPoiMap](http://openpoimap.org/)  
POI = Point of Interest

150 predefined topics.  
You can add your own.

<http://openpoimap.org/>



# Amsterdam supermarkets



- But
- Chn
- Caf
- Dal
- Del
- Dri
- Org
- Sea
- Super
- Win

Now on to our main topic:

## Query tool Overpass

Superfast query tool, flexible, used everywhere within OSM community.

Just a bit steep learning curve

(I'm still climbing, but even from halfway up the mountain the view is already awesome).

We'll start with example output.

Then we will tell more about the language, called **Overpass QL** (short for Query Language), and more about the most popular user interface, called **Overpass Turbo**.

(Overpass QL can also be invoked directly from a script, which we will not cover here)

Back to supermarkets in Amsterdam







Main topic

## Query tool **Overpass**

**Superfast** query tool, **flexible**, **used everywhere** within OSM community.

Language is called **Overpass QL** (short for Query Language)

Most popular web interface is **Overpass Turbo**.

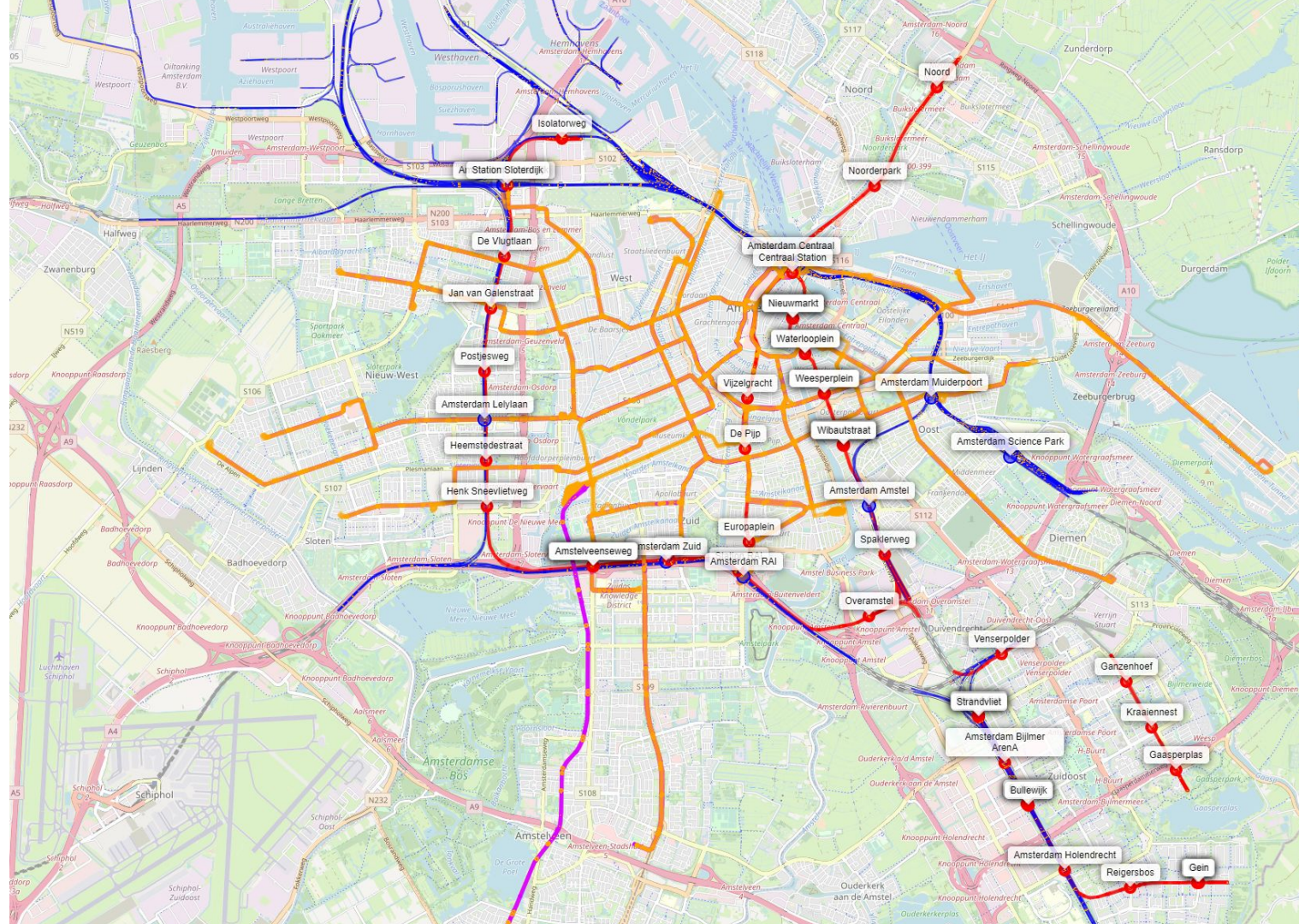
(Overpass QL can also be invoked directly from an app or script, which we will not cover here)

**Overpass QL**

**Amsterdam**

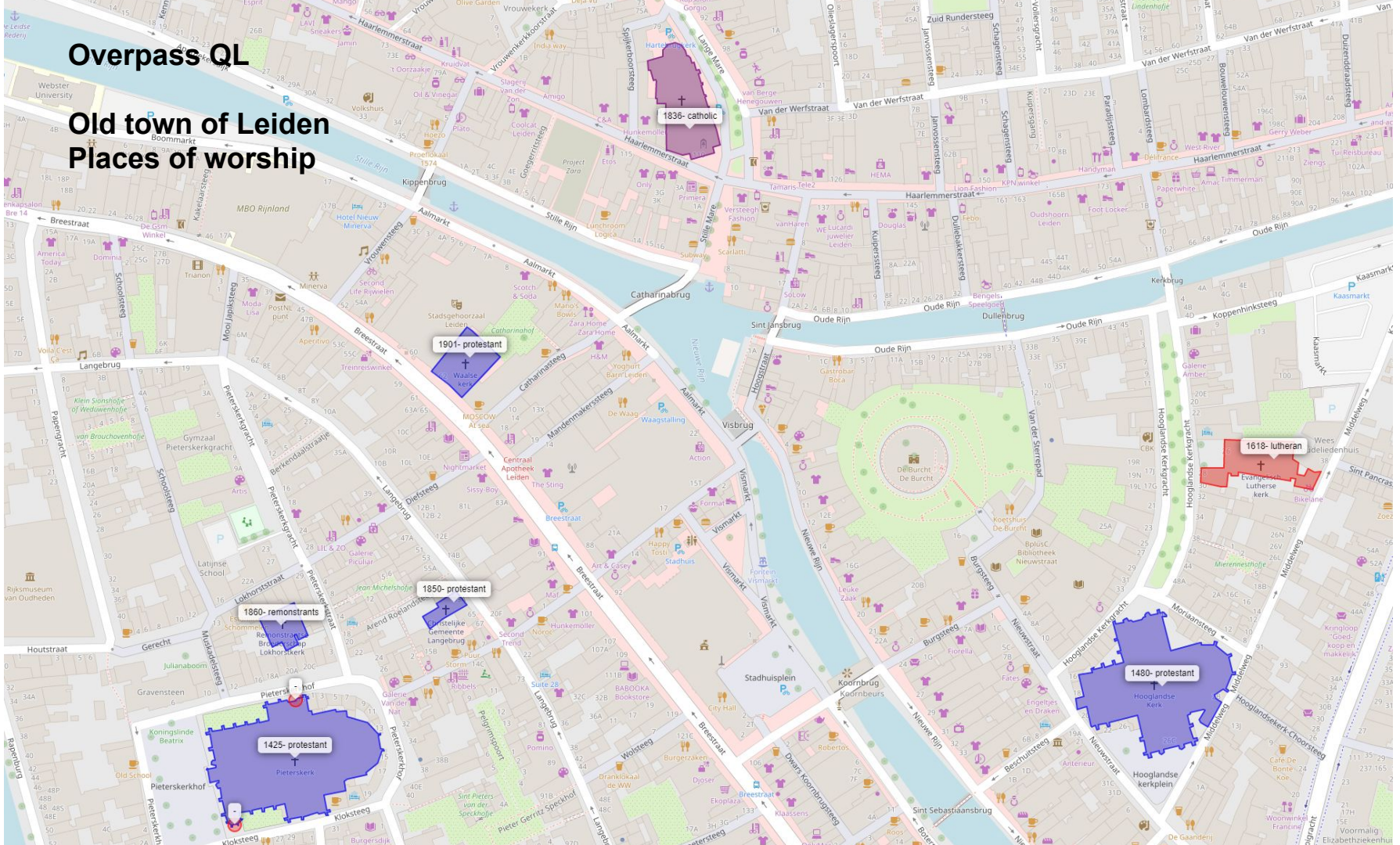
**Train and Subway routes and stations**

**Tram routes**



# Overpass QL

## Old town of Leiden Places of worship



# Today we focus on

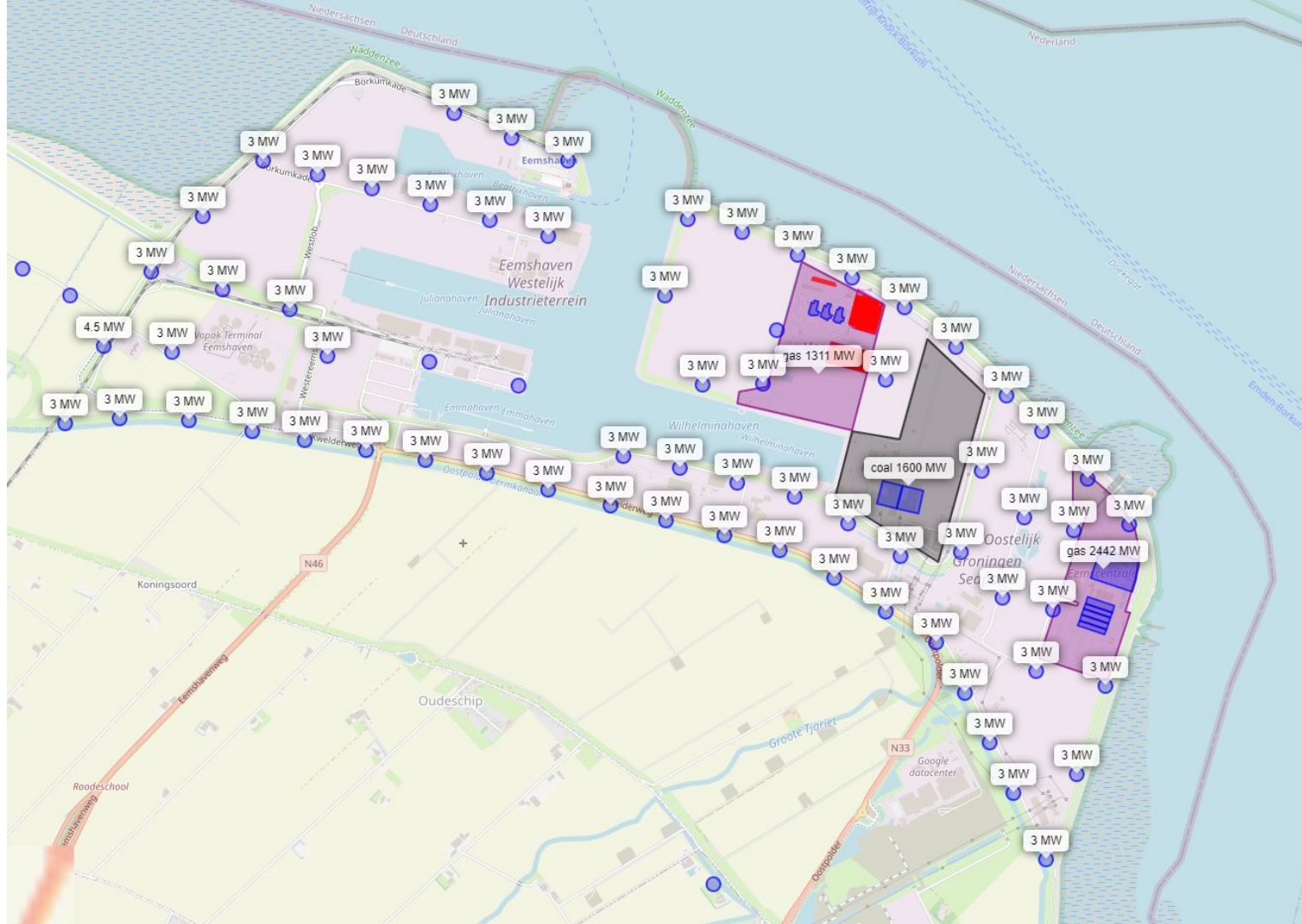


Examples follow on energy production, mostly in The Netherlands.

We'll start on the municipal level, zoom out to regional, then national, then European, then world-wide.

Level of detail varies per zoom level, but all example maps are derived from the same basic scripting concept.

Eemshaven:  
gas 3.8 GW  
coal 1.6 GW  
wind 180 MW  
solar ?





# Offshore wind farms Planned for 2030: 10 GW

wind - 129 MW - Eneco  
Luchterduinen

wind - 120 MW - Prinses  
Amaliawindpark

wind - 108 MW -  
Offshore Windpark  
Egmond aan Zee

Windpark  
Egmond  
aan Zee

gas - 820 MW - Nuon  
Cluster Velsen

solar - 2.38 MWp -  
Zonnepark Boekelermeer

solar - yes - Solar  
Campus Purmerend

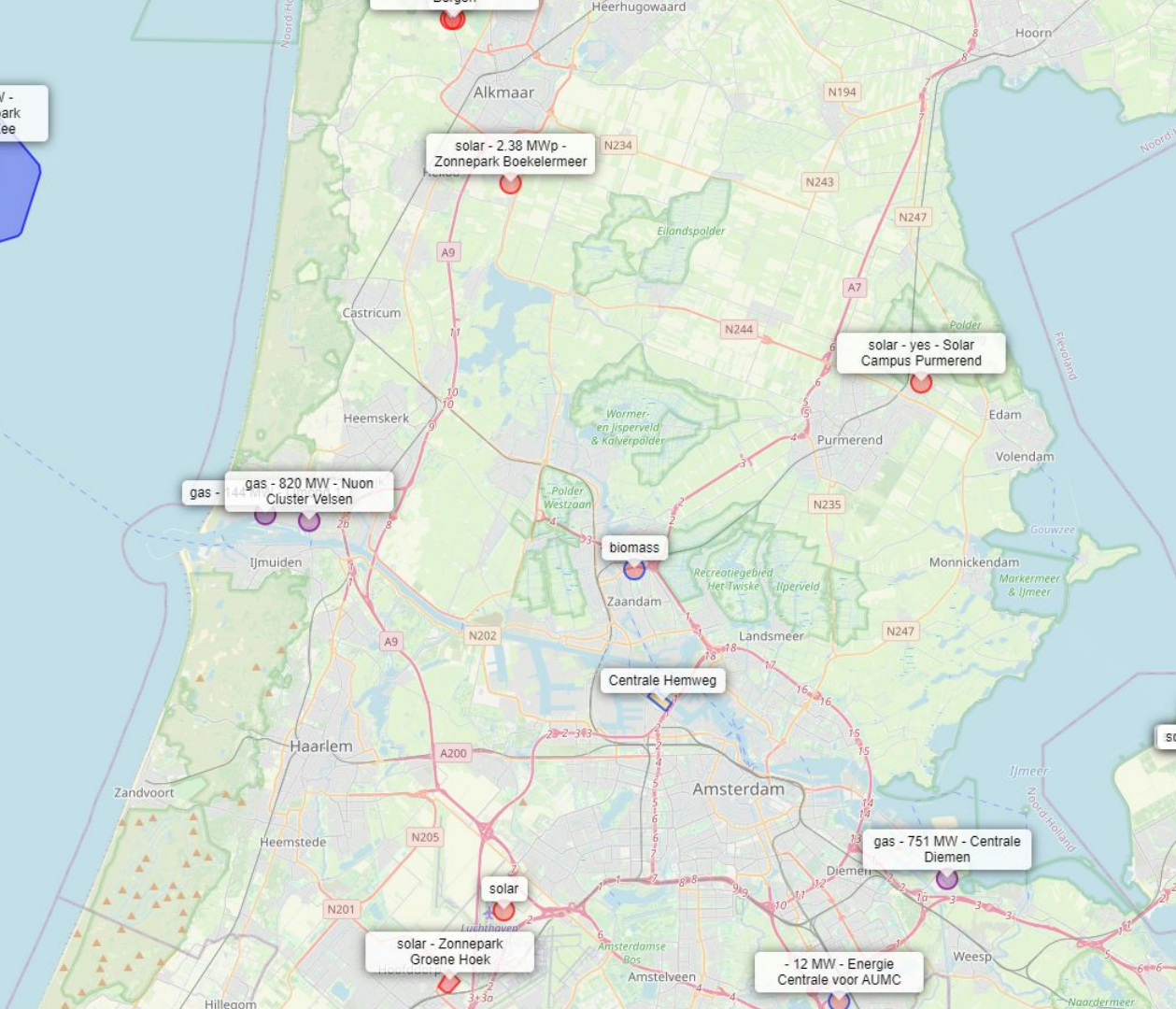
biomass

Centrale Hemweg

gas - 751 MW - Centrale  
Diemen

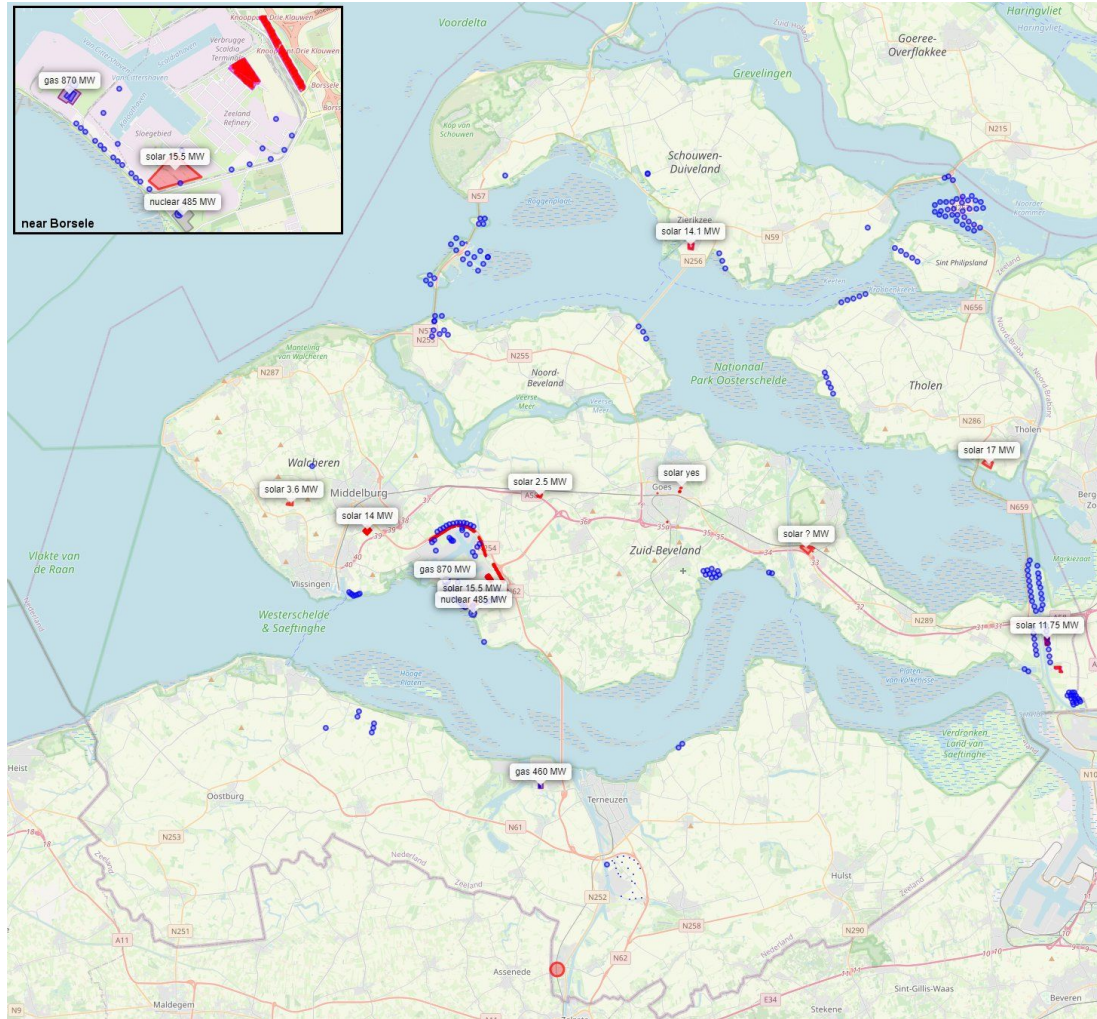
solar - Zonnepark  
Groene Hoek

- 12 MW - Energie  
Centrale voor AUMC



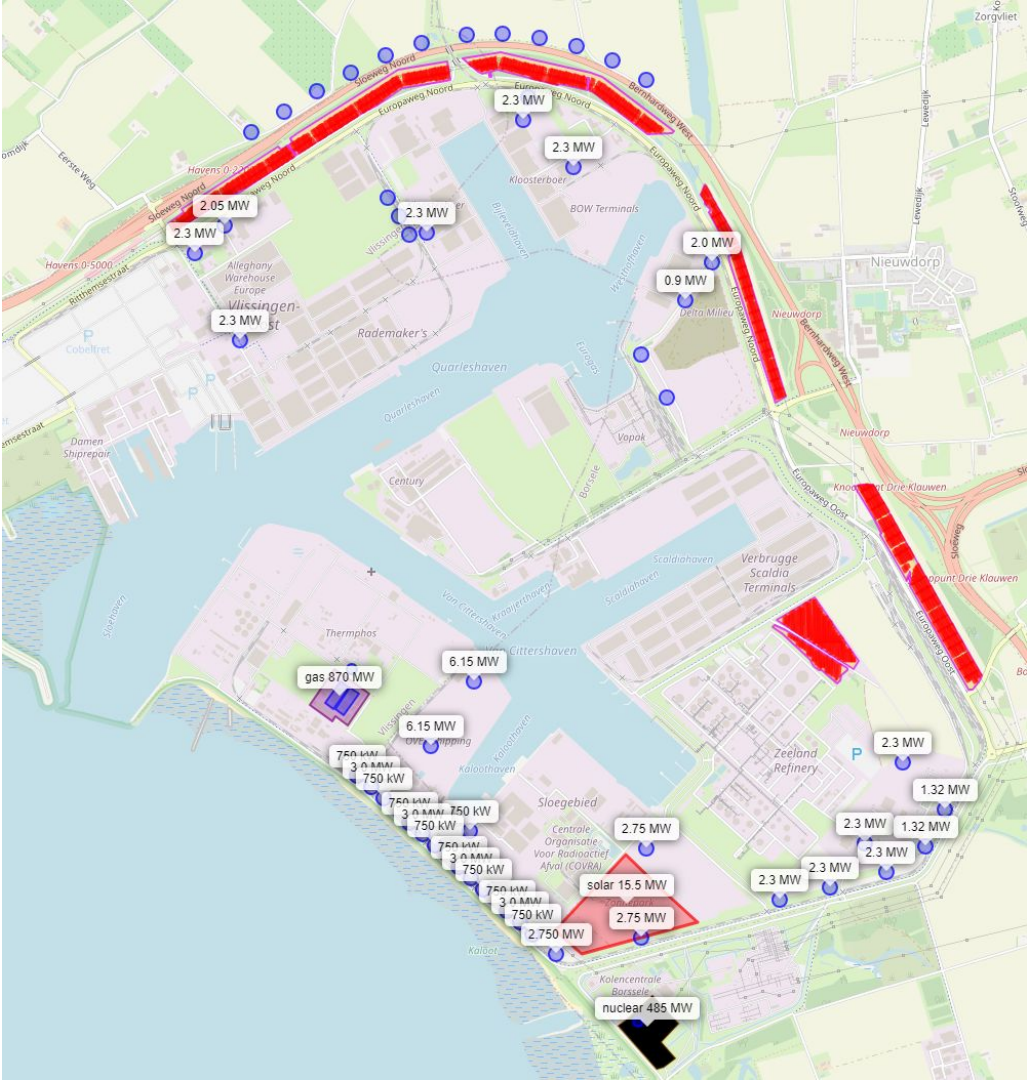
# Province Zeeland

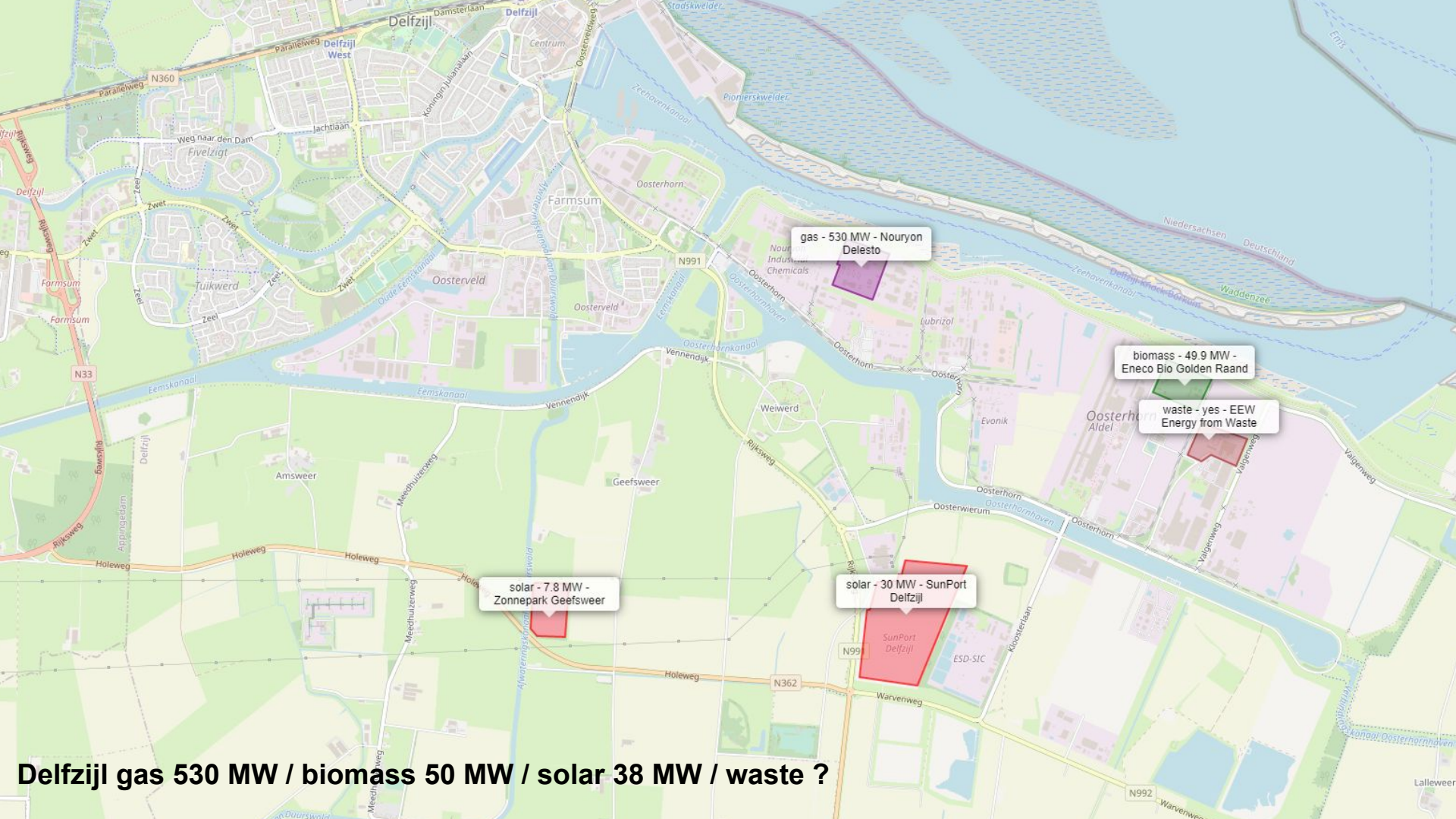
gas 1.3 GW /  
nuclear 485 MW  
solar ..  
wind ..



# Viissingen, Borsele in Zeeland

nuclear 485 MW  
gas 870 MW  
solar, wind





gas - 530 MW - Nouryon Delesto

biomass - 49.9 MW - Eneco Bio Golden Raad

waste - yes - EEW Energy from Waste

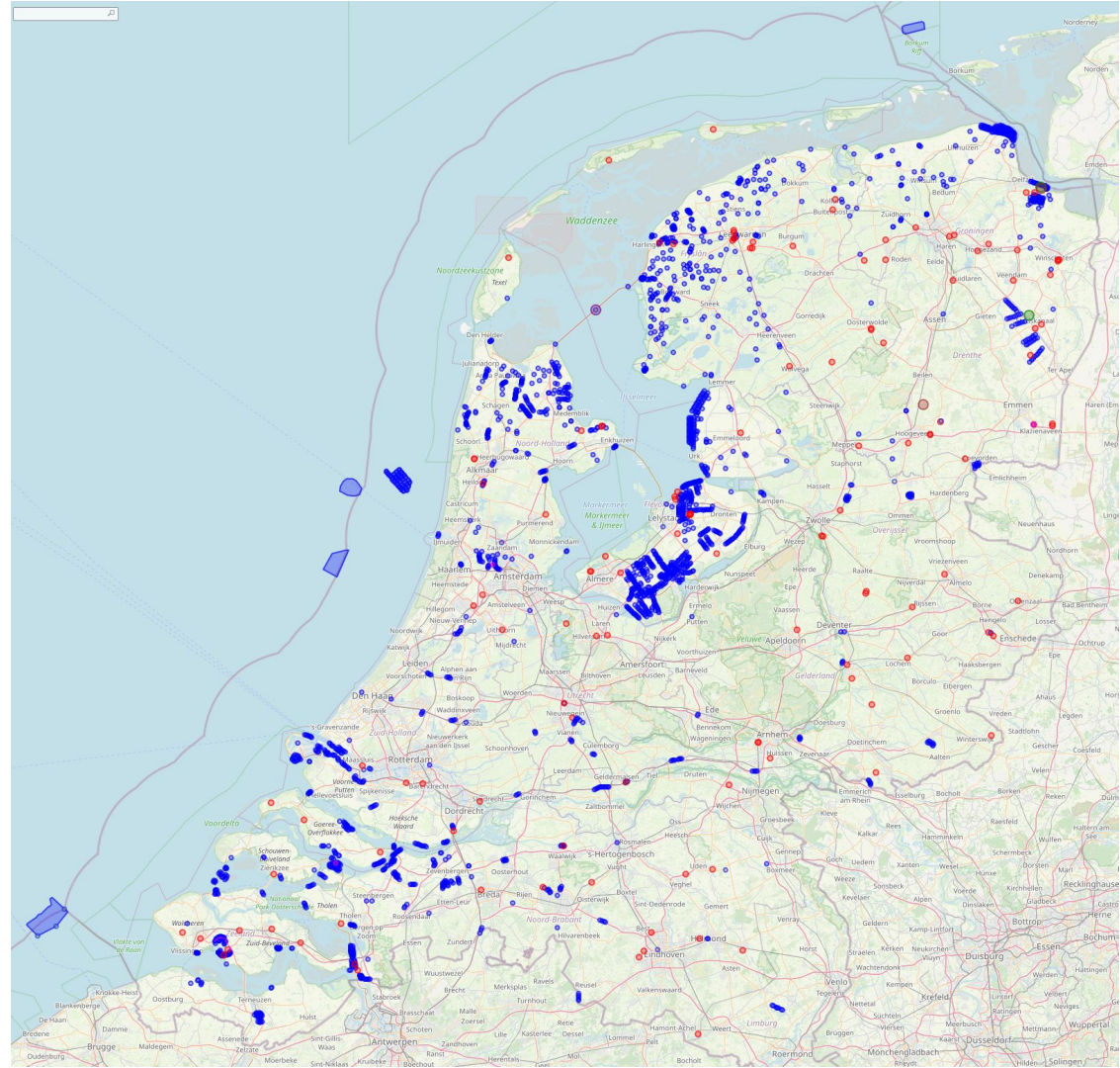
solar - 7.8 MW - Zonnepark Geefsweer

solar - 30 MW - SunPort Delfzijl

**Delfzijl gas 530 MW / biomass 50 MW / solar 38 MW / waste ?**

# Renewable energy

- wind
- solar
- biomass
- waste
- osmotic

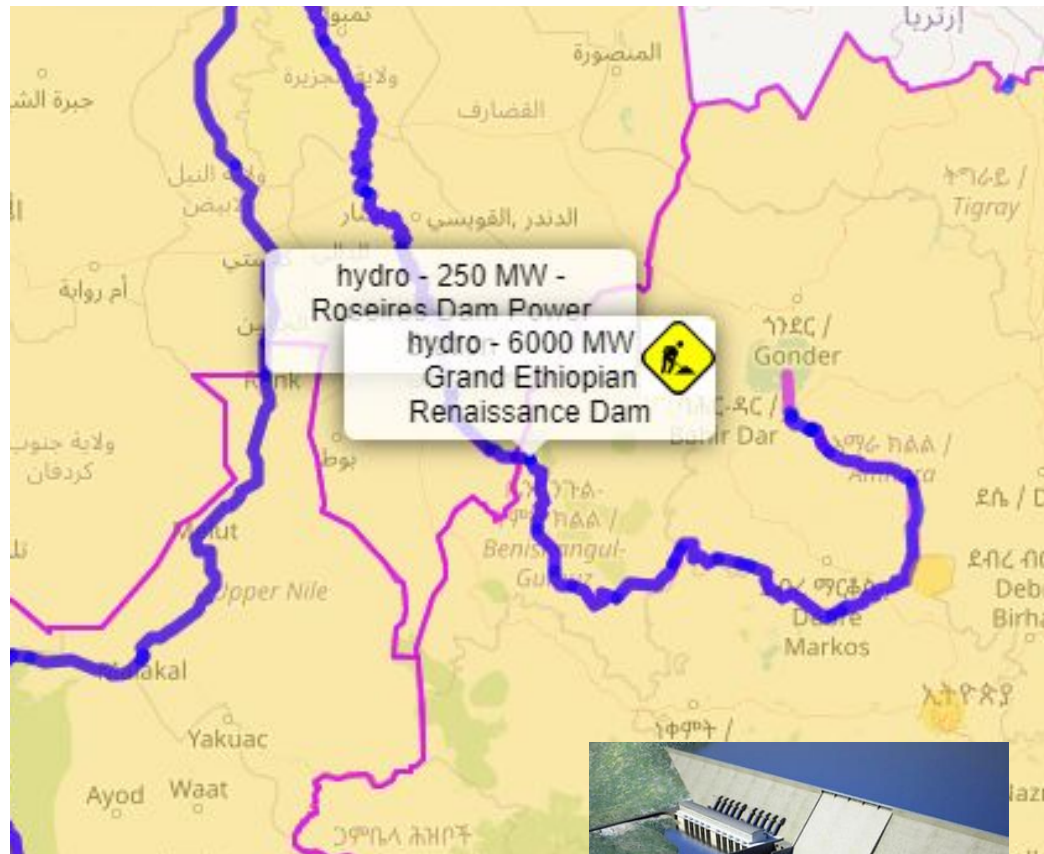
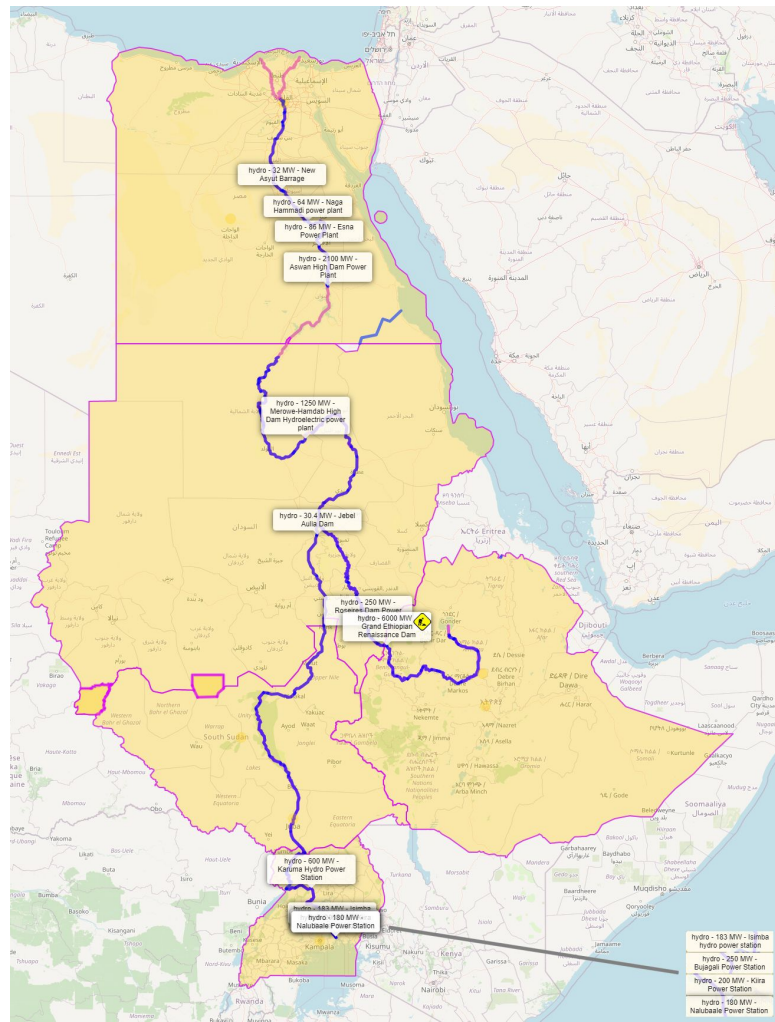


# Transport of electricity via high voltage cables inside and around The Netherlands

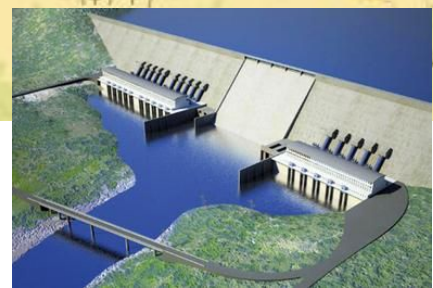
overhead  
underground or underwater

Note: In Belgium infrastructure is incomplete.





<https://w.wiki/gMy>



## Nuclear plants and generators in Europe



**Part of the global map  
in use at Wikipedia  
since 2008!**

**Up-to-date map is easy  
with OSM**

**OSM**







Canada

Island

Россия

Suomi

Sverige

United Kingdom

Belarus

Poland

Romania

Italy

Ελλάδα

Türkiye

Казахстан

Монгол улс

日本

中国

United States

México

Golfo de México

Maroc / المغرب

الجزائر

موريتانيا

ليبيا

مصر

السعودية

عمان

ایران

افغانستان

پاکستان

India

Việt Nam

Philippines

Malaysia

Indonesia

Papua New Guinea

Colombia

Perú

Bolivia

Brasil

Paraguay

Argentina

Mali

Niger

Tchad

السودان

South Sudan

الصومال

Soomaaliya

Kenya

Tanzania

Angola

République démocratique du Congo

Namibia

Moçambique

South Africa

العراق

السعودية

عمان

India

Việt Nam

Philippines

Malaysia

Indonesia

Papua New Guinea

Australia

New Zealand  
Aotearoa

Nuclear plants  
and generators

OSM  
ways: 437  
relations: 23

# Let's build a map step-by-step

(and pretend it is easy ;-)

Sep 2020

> 6 billion



### Node

Bench  
Tree  
Dust bin

**Often only coordinates**

> 200 million



### Way

Road segment  
River segment

> 480 million



### Way (closed)

House  
Lake

> 8 million



### Relation

Border  
Route  
River



### Node

Bench  
Tree  
Dust bin

**Often only coordinates**



### Way

Road segment  
River segment



### Way (closed)

House  
Lake



### Relation

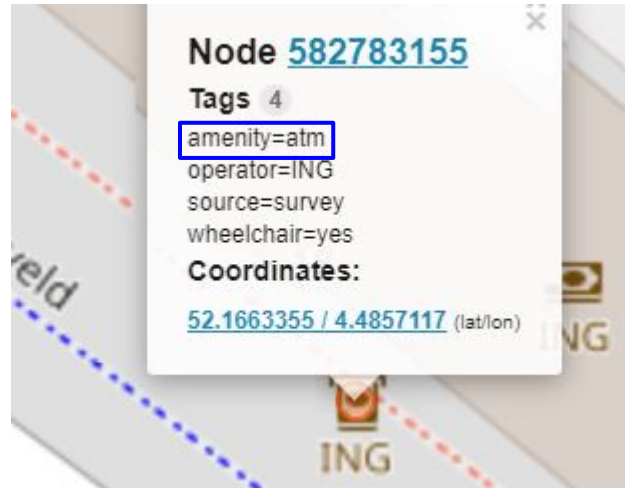
Border  
Route  
River

A **way** has one or more **nodes** as 'children' which together define its geography.

A **relation** has **nodes** and/or **ways** as 'children'.

No strict rules on how to encode info:

Some mappers may add a feature as separate node:  
**amenity=atm**



Others may add it as **tag** to a different node or way:  
**amenity=bank**  
**atm=yes**





Pieterskerk  
in Leiden

Way 5296606  
has 180 nodes

Node 3331494790

**Measurement** ✕

n3331494790

Geometry: point  
Location: 52°9'26"N, 4°29'13"E  
52.1573277, 4.4869298

**Measurement** ✕

w52966906

Geometry: closed area  
Number of nodes: 180  
Area: 39,580 sq ft (0.91 ac)  
Perimeter: 1,254 ft  
Centroid: 52°9'27"N, 4°29'16"E  
52.1576133, 4.4877529

Imperial

We'll introduce the Overpass Turbo UI, and paste a small script, run it, and afterwards explain it in detail.



overpass turbo

https://overpass-turbo.eu

Run Share Export Wizard Save Load Logout Settings Help overpass turbo

Map Data

1

Map of the Leiden region in the Netherlands, showing roads, waterways, and various locations. The map is overlaid with a red line representing a route. Key locations include Noordwijk, Leiden, Zoeterwoude, and Alphen aan den Rijn. The map also shows the Kagerplassen and various polders like the Boterhuspolder and Rodepolder. A scale bar indicates 2 km.

Run

Share

Export

Wizard

Save

Load

Logout

Settings

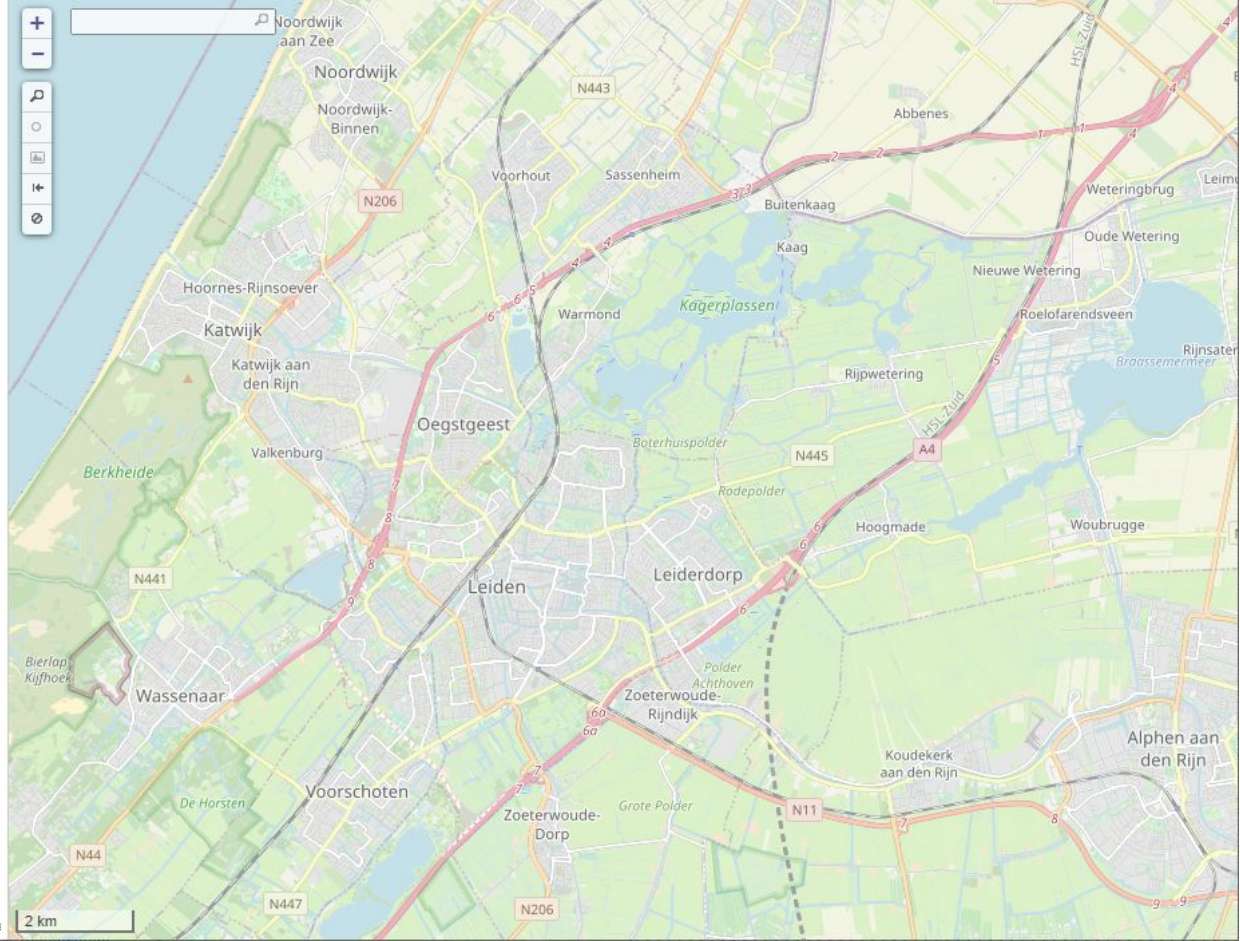
Help

overpass turbo

Map

Data

1



Run

Share

Export

Wizard

Save

Load

Logout

Settings

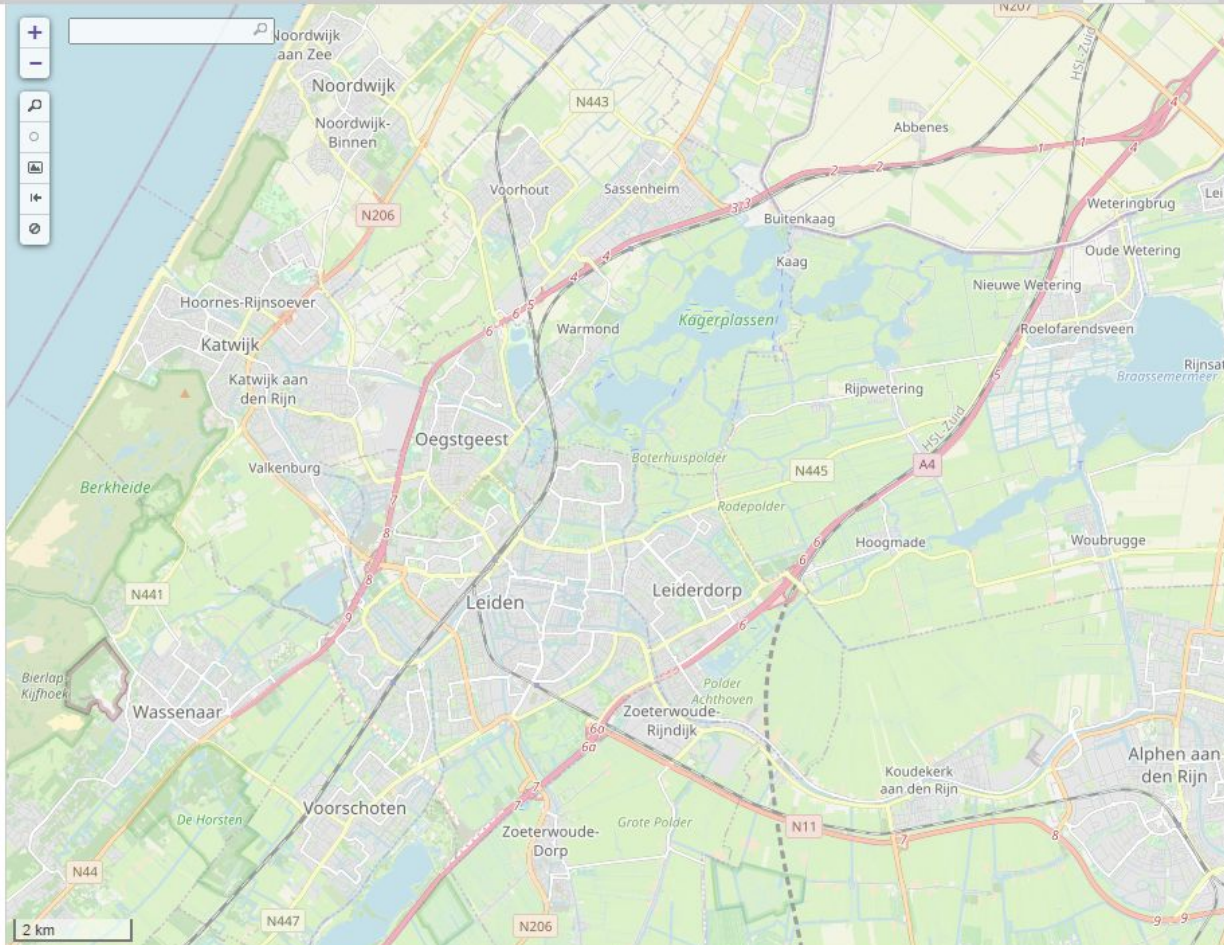
Help


overpass turbo

Map

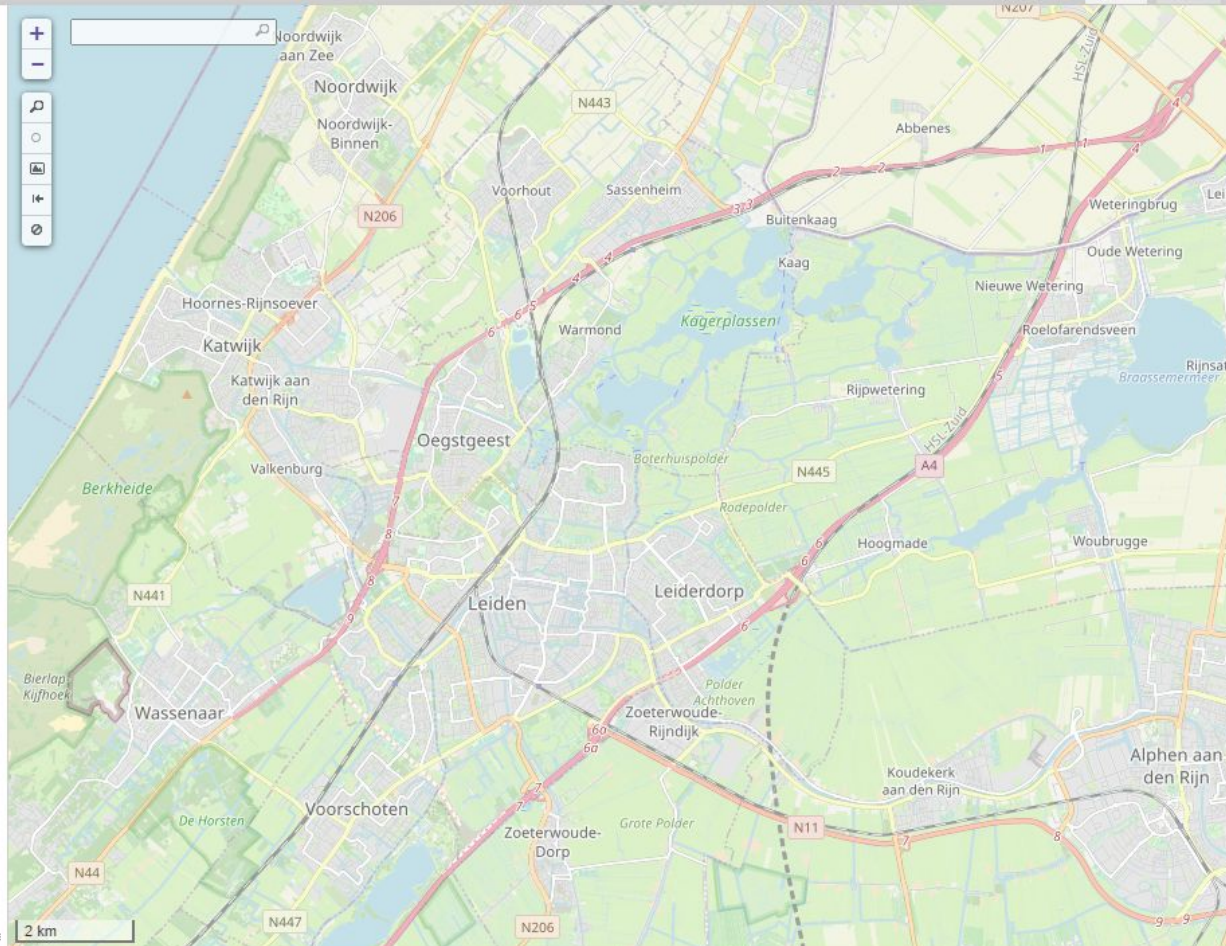
Data

```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power](area.a>{{bbox}});
4 out body;
5 >;
6 out meta qt;
```



[Run](#)[Share](#)[Export](#)[Wizard](#)[Save](#)[Load](#)[Logout](#)[Settings](#)[Help](#)overpass turbo [Map](#)[Data](#)

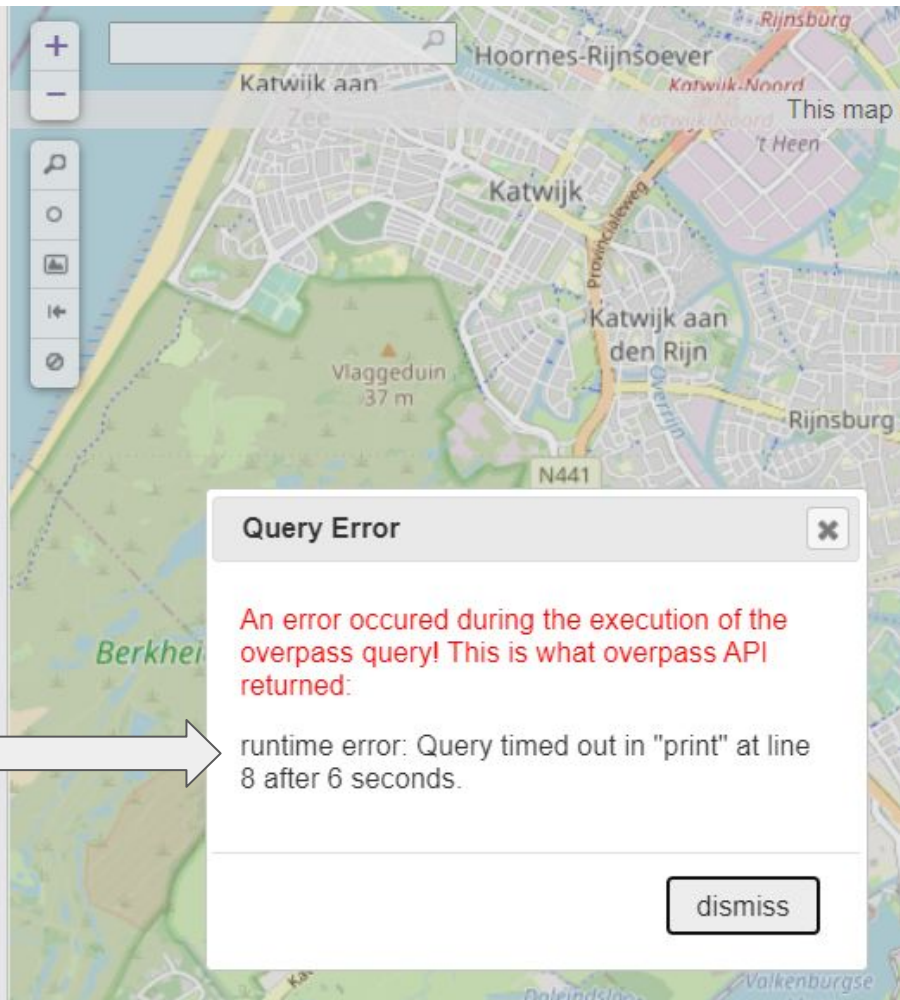
```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power](area.a>{{bbox}});
4 out body;
5 >;
6 out meta qt;
```





```
[out:json][timeout:5];
area[name="Nederland"]->.a;
(
  nu[power-generator][~":source"="wind"](.a.a)(.bbox);
  nu[power-generator][~":source"="solar"](.a.a)(.bbox);
  nu[power-plant](.a.a)(.bbox);
);
out body;
>;
out meta qt;

{{style:
node,way,rel
{ text: power;
}}}
```



You reserved not enough run time.

```
1 [out:json][timeout:5555555];
2 area[name="Nederl" ...];
3 (
4   nu[power-generator][source="wind"](area.a)(bbox);
5   nu[power-generator][source="solar"](area.a)(bbox);
6   nu[power-plant](area.a)(bbox);
7 );
8 out body;
9 >;
10 out meta qt;
11
12 [[style:
13   node,way,rel
14   { text: power; }
15 ]]
```

**Query Error**

An error occurred during the execution of the overpass query! This is what overpass API returned:

**Error:** runtime error: [...] The server is probably too busy to handle your request.

dismiss

You reserved too much run time.

How long should you wait before submitting the next script?

← → ↻ ⚠ Not secure | overpass-api.de/api/status

Connected as: 588209664

Current time: 2020-09-08T15:46:48Z

Rate limit: 2

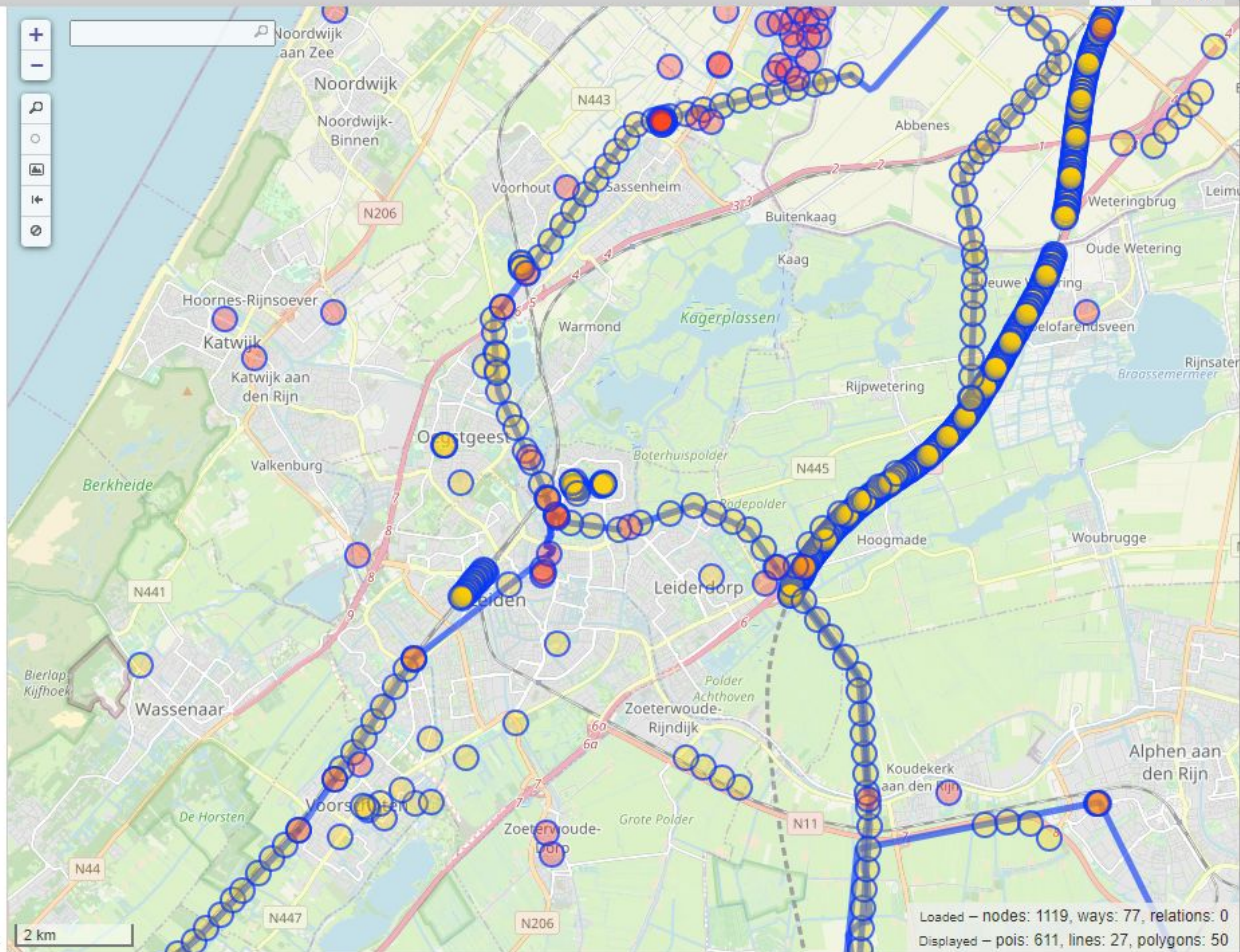
Slot available after: 2020-09-08T15:47:17Z, in 29 seconds.

Slot available after: 2020-09-08T15:47:42Z, in 54 seconds.

Currently running queries (pid, space limit, time limit, start time):



```
1 [out:json][timeout:250];  
2 area[name="Nederland"]->.a;  
3 nwr[power]({{bbox}})(area.a);  
4 out body;  
5 >;  
6 out meta qt;  
7  
8
```



## Presentation notes

We've seen the output of the script, but what does it tell us?

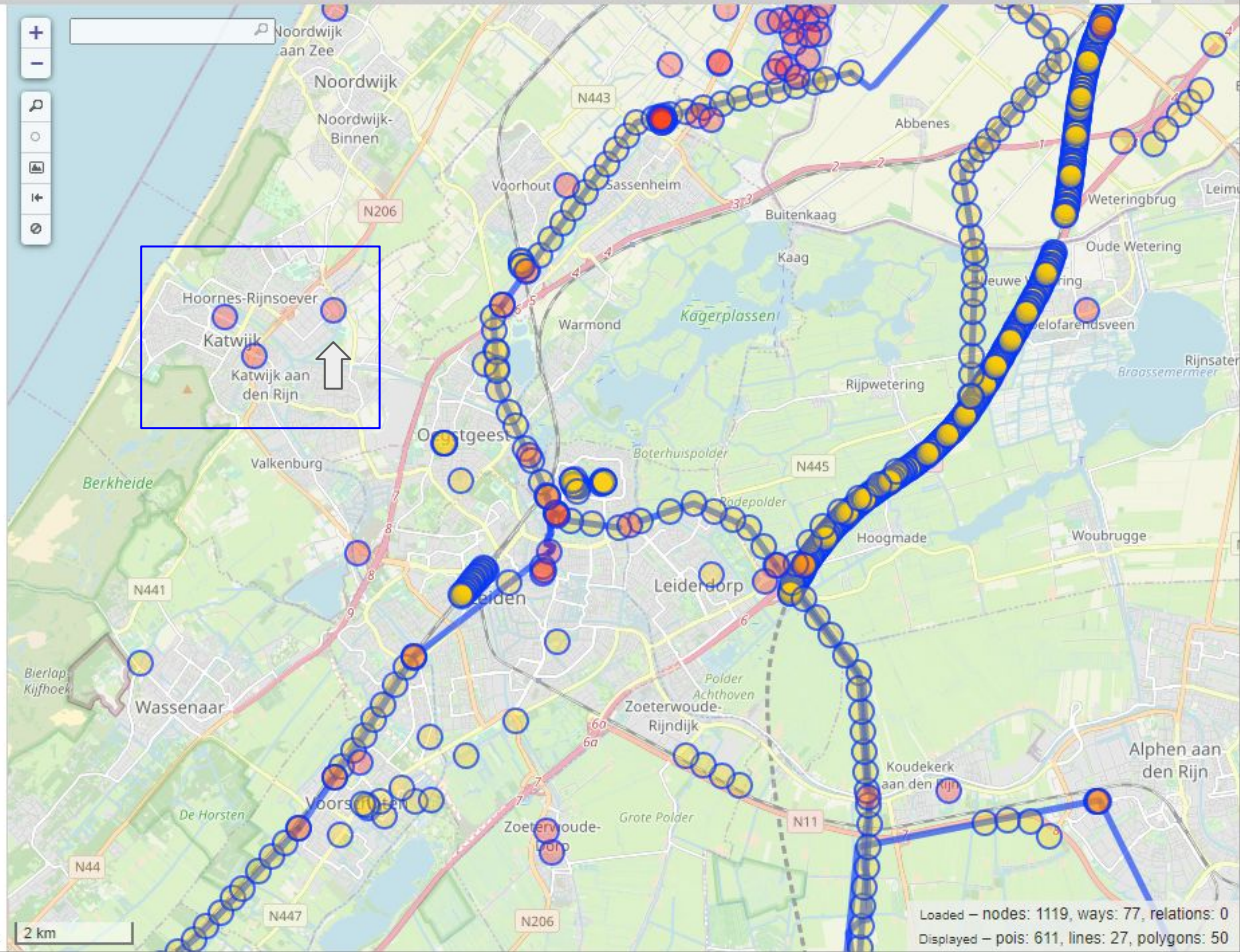
We see many circles, in two basic colors, yellow and red. (Some colors are more vivid than others, but that's merely because of overlapping half transparent circles).

Yellow circles are **nodes**.

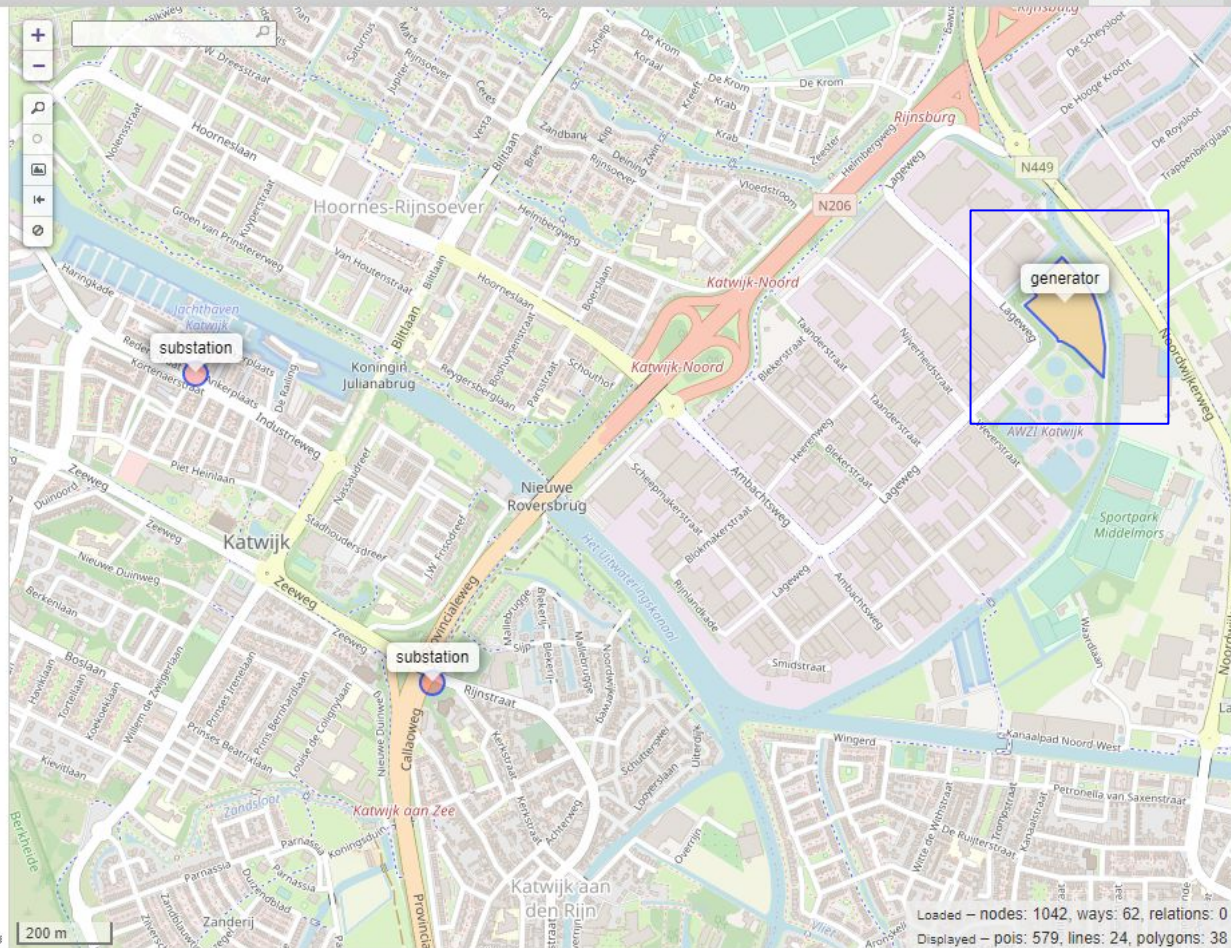
Red circles are **ways**, which on a low zoom level would be almost unnoticeably small. There is this option to present these tiny areas on lower zoom levels as a red circle instead. This option can be switched off as shown.

Let's zoom in on the left part of the map and see a red circle replaced by an area (2D). Same would happen for other 2 circles if we zoomed in further.

```
1 [out:json][timeout:250];  
2 area[name="Nederland"]->.a;  
3 nwr[power]({{bbox}})(area.a);  
4 out body;  
5 >;  
6 out meta qt;  
7  
8
```



```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power]({{bbox}})(area.a);
4 out body;
5 >;
6 out meta qt;
7
8 {{style:
9 node,way,rel
10 { text: power ; }
11 }}
```



Run Share Export Wizard Save Load Log

```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power]({bbox})(area.a);
4 out body;
5 >;
6 out meta qt;
7
8 {{style:
9 node,way,rel
10 { text: power ; }
11 }}
```



### Settings

#### General Settings

#### Editor

#### Map

Tile-Server:

[[s].tile.openstreetmap.org/{z}/{x}/{y}.png ▼

Tiles Opacity:  (transparency of background tiles:  
0=transparent ... 1=visible )

Show crosshairs at the map center.

Don't display small features as POIs.

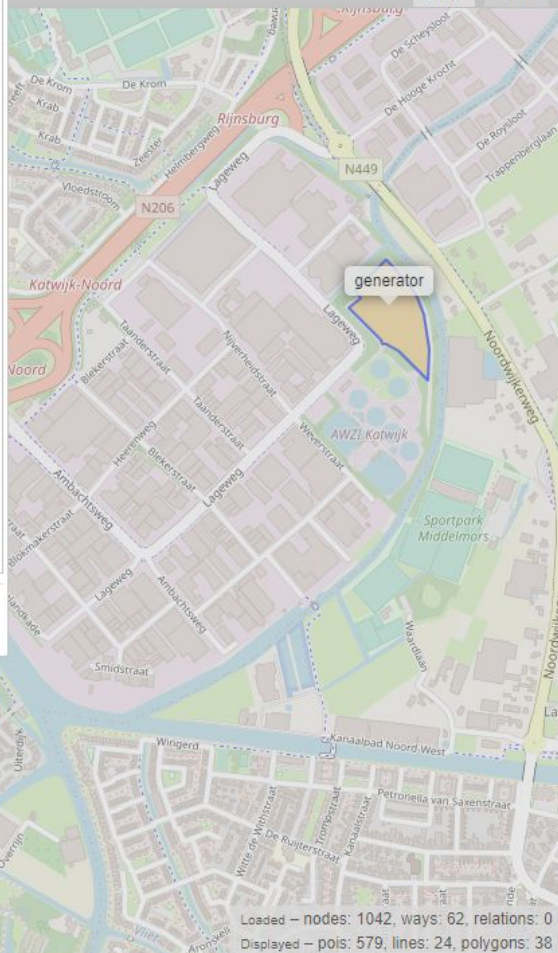
Show some stats about loaded and displayed data.

#### Sharing

#### Export

save

Map Data



Loaded – nodes: 1042, ways: 62, relations: 0  
Displayed – pois: 579, lines: 24, polygons: 38

After we've shown the data panel, we'll explain the script in minute detail, or actually two scripts, first **Overpass QL**, then **MapCSS**.

```

1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power]{{{bbox}}}(area.a);
4 out body;
5 >;
6 out meta qt;
7
8

```

```

1 {
2   "version": 0.6,
3   "generator": "Overpass API 0.7.56.7 b85c4387",
4   "osm3s": {
5     "timestamp_osm_base": "2020-09-08T10:06:03Z",
6     "timestamp_areas_base": "2020-09-08T09:22:03Z",
7     "copyright": "The data included in this document is from www.openstreetmap.org. The data is made available under
8   },
9   "elements": [
10
11     {
12       "type": "node",
13       "id": 34962882,
14       "lat": 52.1986219,
15       "lon": 4.4821902,
16       "tags": {
17         "design": "donau",
18         "power": "tower",
19         "ref": "10"
20       }
21     },

```



```

15871 {
15872   "type": "node",
15873   "id": 2287121222,
15874   "lat": 52.2490116,
15875   "lon": 4.6375490,
15876   "timestamp": "2013-04-30T20:20:07Z",
15877   "version": 1,
15878   "changeset": 15925608,
15879   "user": "kmer1",
15880   "uid": 490731
15881 }
15882 ]
15883 }
15884
15885

```

Loaded – nodes: 1119, ways: 77, relations: 0  
 Displayed – pois: 611, lines: 27, polygons: 50

```

1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power]({{bbox}})(area.a);
4 out body;
5 >;
6 out meta qt;
7
8 {{style:
9 node,way,rel
10 { text: power ; }
11 }}

```



Overpass QL

MapCSS





```
1 [out:json][timeout:250];  
2 area[name="Nederland"]->.a;  
3 nwr[power](area.a)({{bbox}});  
4 out body;  
5 >;  
6 out meta qt;
```

```
1 [out:json][timeout:250];  
2 area[name="Nederland"]->.a;  
3 nwr[power](area.a)({{bbox}});  
4 out body;  
5 >;  
6 out meta qt;
```

```
[out:json]           data output format = json  
[timeout:250];      allowed run time for query is 250 seconds  
;                   end of statement
```

Other [output formats](#): *xml*, *csv*, [and more](#)

```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power](area.a)({{bbox}});
4 out body;
5 >;
6 out meta qt;
```

```
area
[name="Nederland"]
->.a
;
```

use **predefined search area**  
known by this name in the database  
store area definition in set '.a' (can be any name)  
end of statement

```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power](area.a)({{bbox}});
4 out body;
5 >;
6 out meta qt;
```

```
nwr
[power]
(area.a)
({{bbox}})
;
```

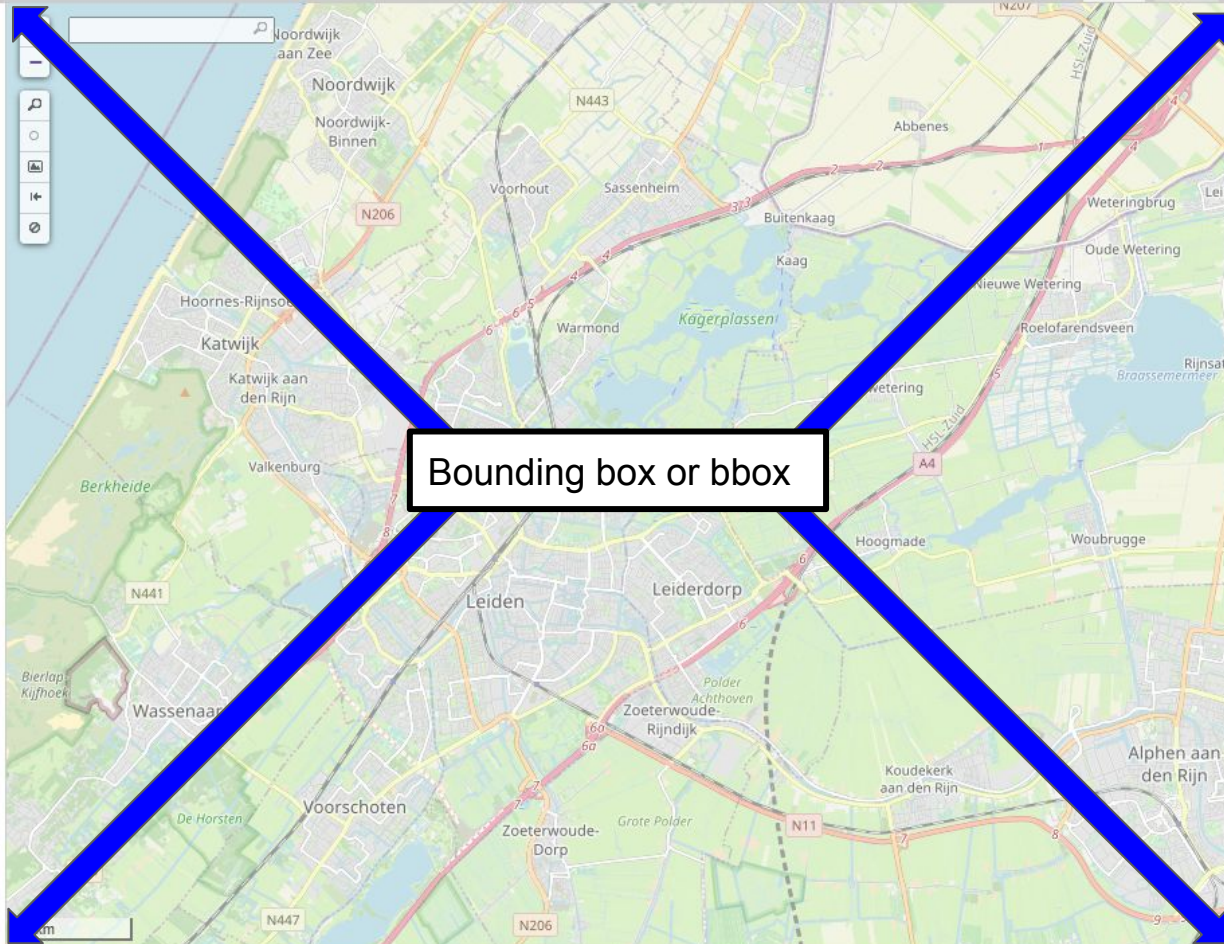
shorthand: find **N**odes, **W**ays and **R**elations  
which contain a **tag with key 'power'** (regardless of value)  
**must be inside area** defined previously, named **'a'**,  
and **also within** visible part of map aka **'bounding box'**  
end of statement

```
1 [out:json][timeout:250];  
2 area[name="Nederland"]->.a;  
3 nwr[power: {{{bbox}}}];  
4 out body;  
5 >;  
6 out meta qt;
```

**(area.a)({{bbox}})**

implies:

search inside the area specified by name ("Nederland") and also inside the bounding box, whichever is smallest



```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power](area.a)({{bbox}});
4 out body;
5 >;
6 out meta qt;
```

**out** **send to output (~ print)**

**body** **body = print all information necessary to process the data further**  
**;** **end of statement**

**Compare 1 and 4, both say 'out' but they are about different aspects of output.**

1 is about output **format**, json, xml, etc

4 is about **which elements and tags** should be sent to output.

Only id? Only content ? Or also context, like user id and date/time of last edit?

```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power](area.a)({{bbox}});
4 out body;
5 >;
6 out meta qt;
```

>

**expand results** ('recurse down' =  
**replace** ways and relations **with underlying nodes**)  
end of statement

;

```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power](area.a)({{bbox}});
4 out body;
5 >;
6 out meta qt;
```

out **send to output**

meta **print all available information** (= 'body' supplemented with version, changeset id, timestamp, user who last edited)

qt **order by geo region** 'qt' from 'quadtile', not 'asc' = 'ascending by id'  
(qt is faster)

;  
**end of statement**



Presentation notes

Again, too many circles, but which is which?

Can we make this more explicit with some MapCSS?

Yes, we can. Let's introduce map labels.

Run

Share

Export

Wizard

Save

Load

Logout

Settings

Help

overpass turbo

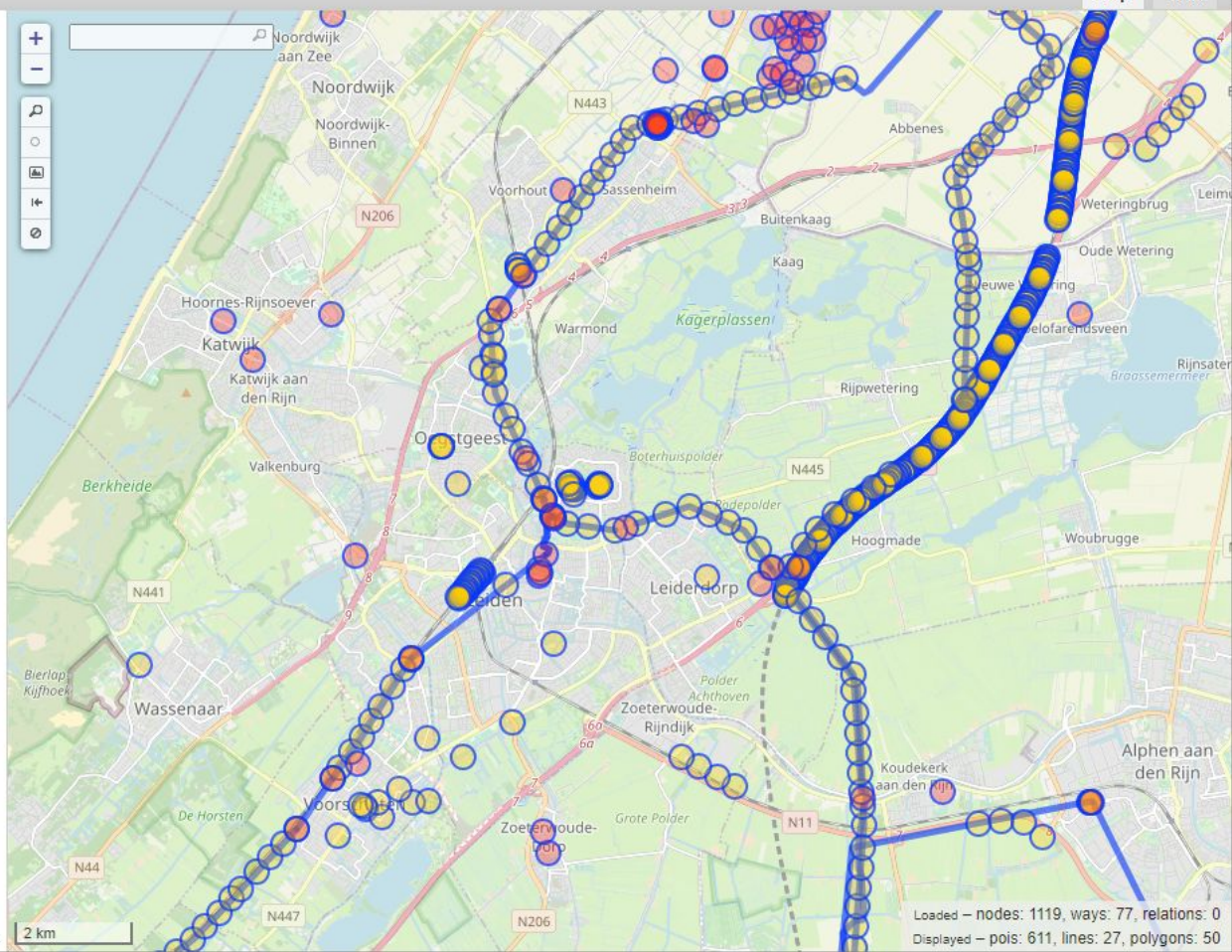
Map


Data

```

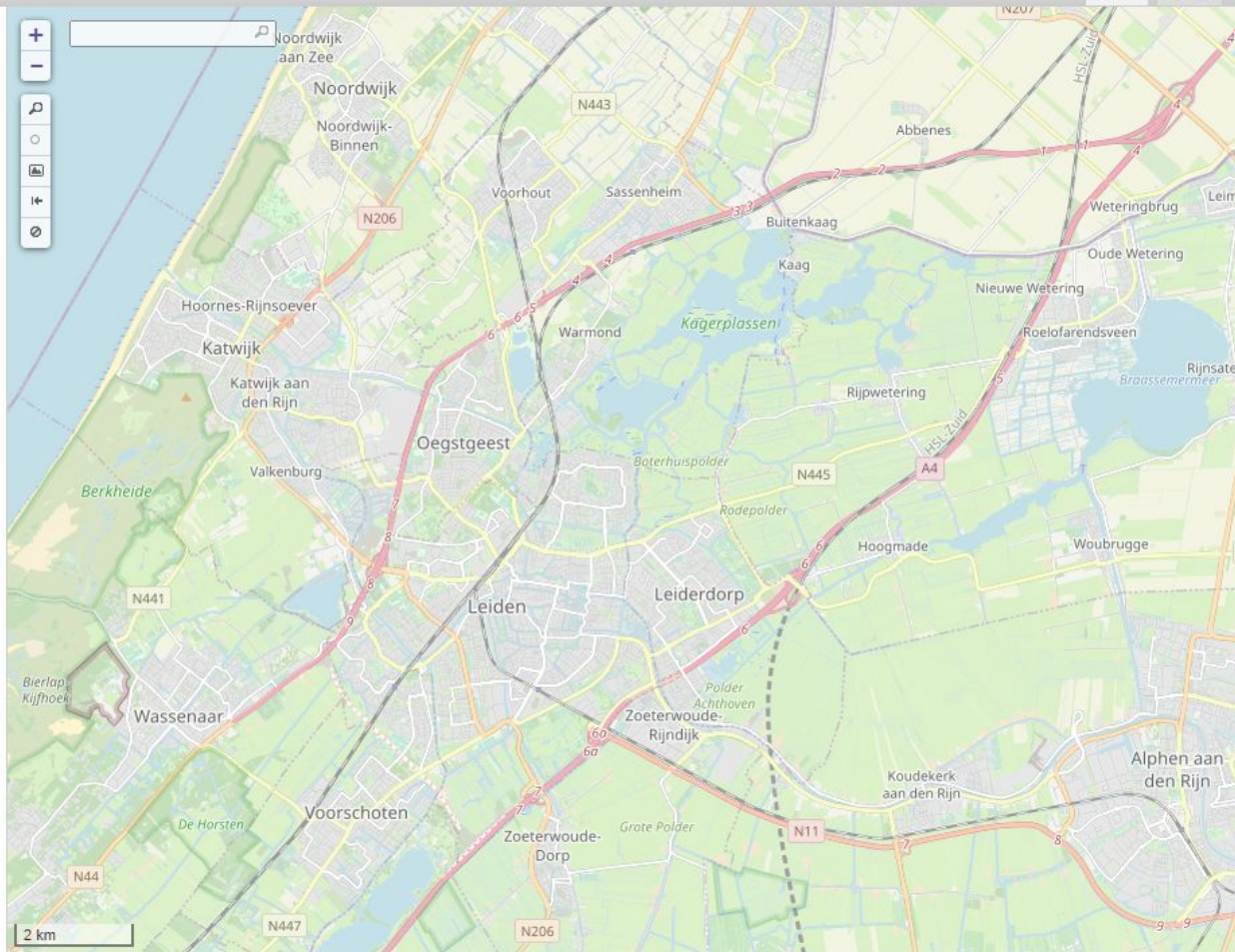
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power]({{bbox}})(area.a);
4 out body;
5 >;
6 out meta qt;
7
8

```



[Run](#)[Share](#)[Export](#)[Wizard](#)[Save](#)[Load](#)[Logout](#)[Settings](#)[Help](#)overpass turbo [Map](#)[Data](#)

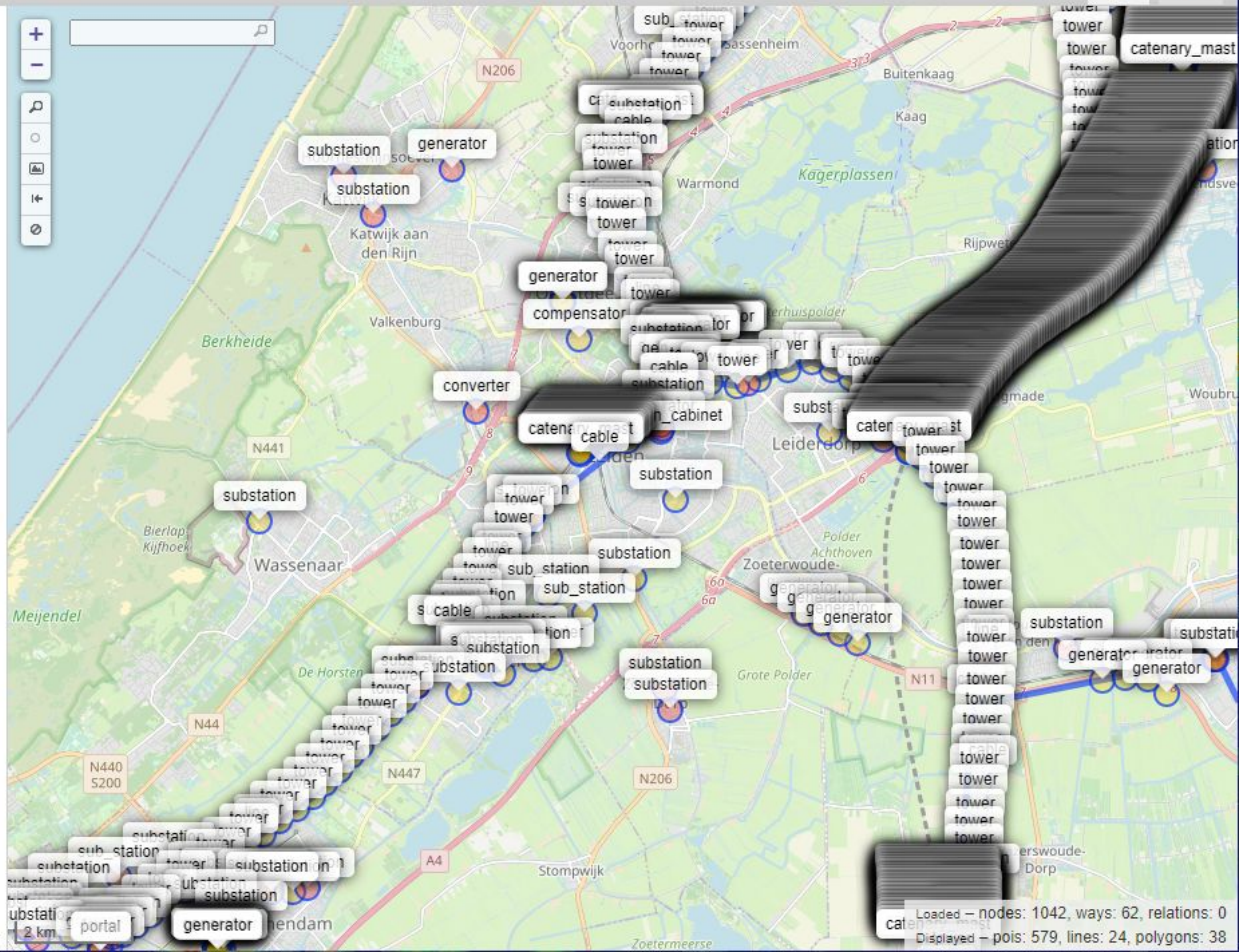
```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power](area.a)({{bbox}});
4 out body;
5 >;
6 out meta qt;
7
8 {{style:
9 node,way,rel
10 { text: power ; }
11 }}
```



```

1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power]({{bbox}})(.a.a);
4 out body;
5 >;
6 out meta qt;
7
8 {{style:
9 node,way,rel
10 { text: power ; }
11 }}

```



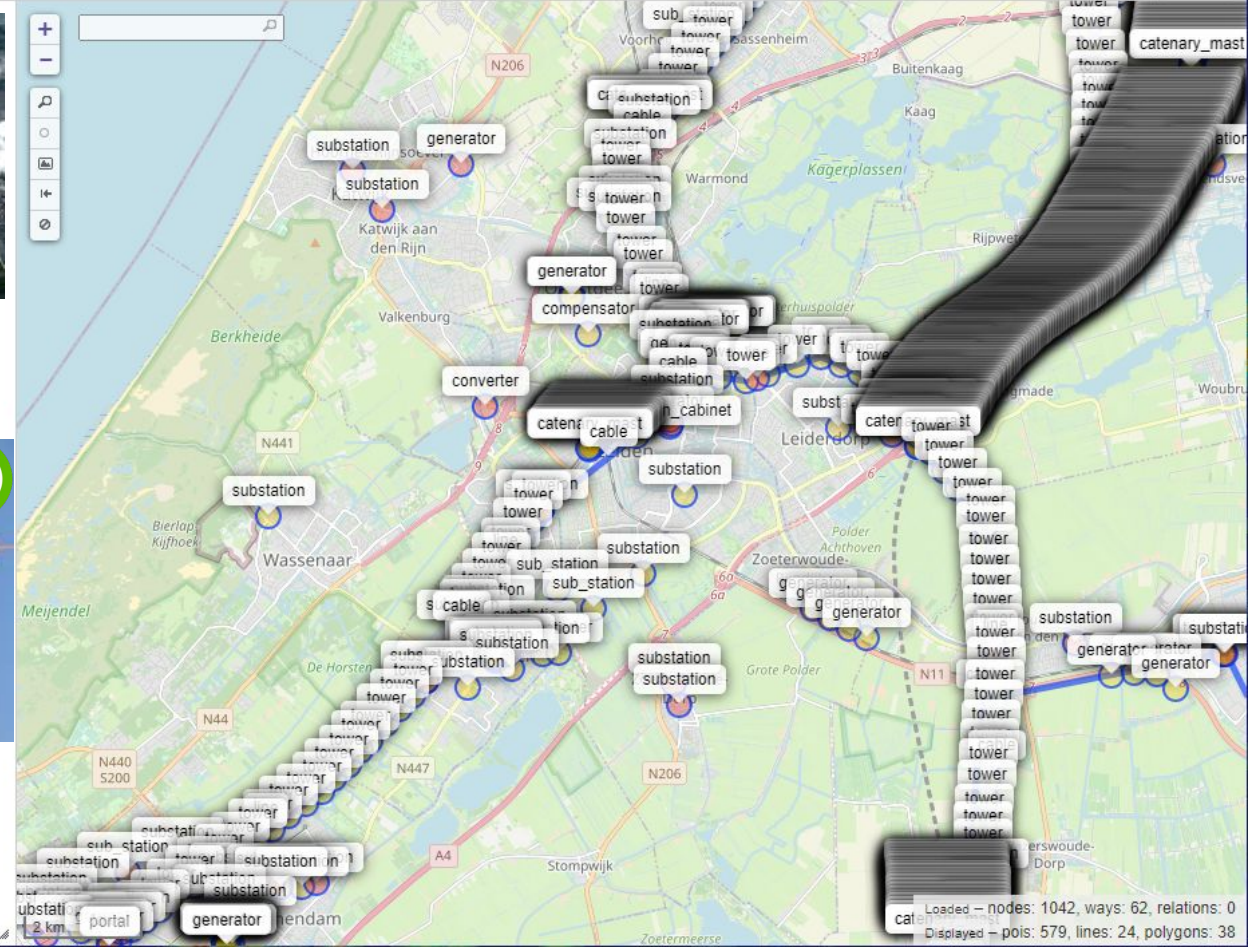
Loaded - nodes: 1042, ways: 62, relations: 0  
 Displayed - pois: 579, lines: 24, polygons: 38

## Presentation notes

We want to visualize power production only, not power conversion, power transport, etc.



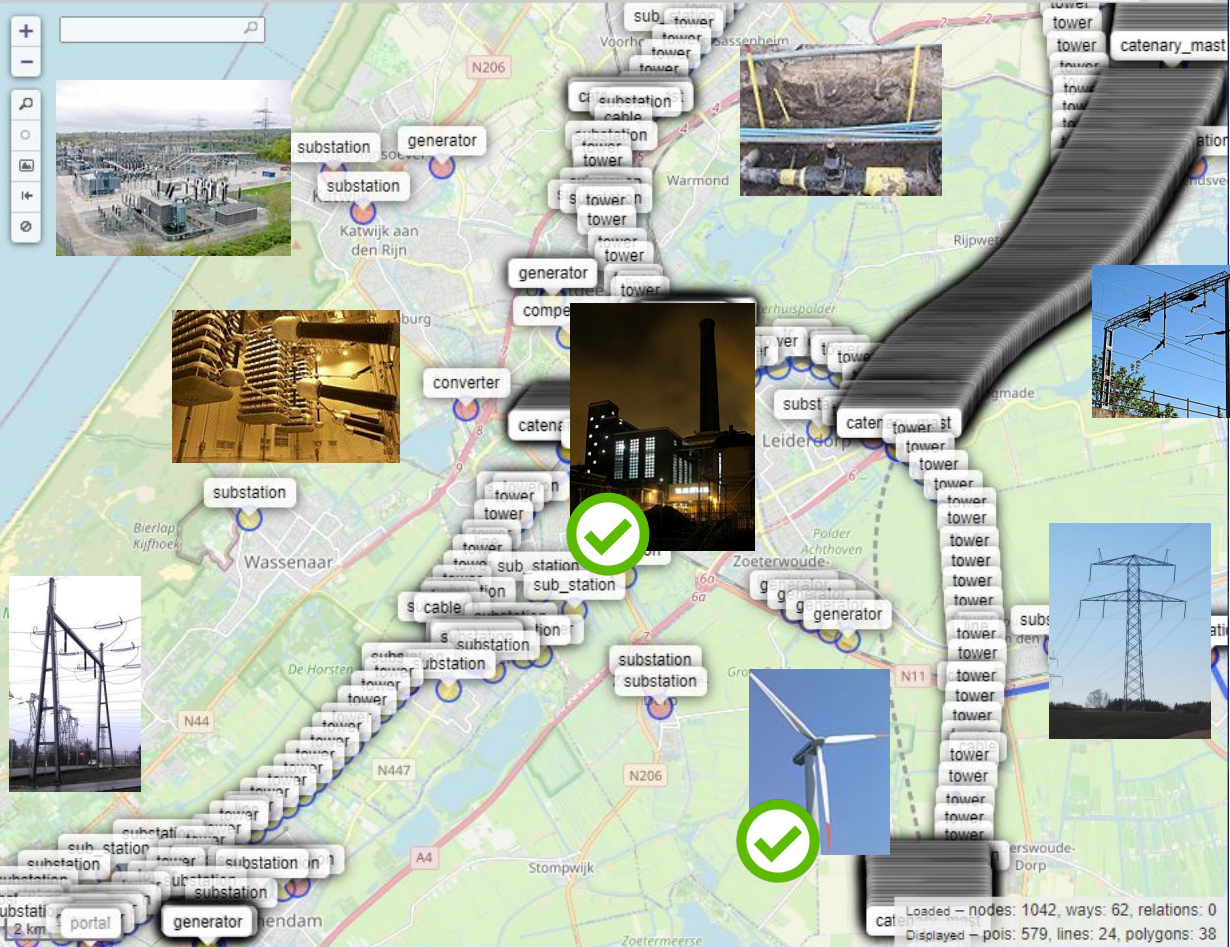
power production



Loaded - nodes: 1042, ways: 62, relations: 0  
 Displayed - pois: 579, lines: 24, polygons: 38



power plants & generators



Loaded — nodes: 1042, ways: 62, relations: 0  
 Displayed — pois: 579, lines: 24, polygons: 38

overpass turbo x overpass turbo x overpass turbo x overpass turbo x + - □ ×

← → ↻ overpass-turbo.eu ★ 📄 🟢 🟦 ⚙️ Ⓔ

Run Share Export Wizard Save Load Logout Settings Help overpass turbo

Map Data

```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power=plant](area.a)({bbox});
4 out body;
5 >;
6 out meta qt;
7
8 {{style:
9 node,way,rel
10 { text: power ; }
11 }}
```

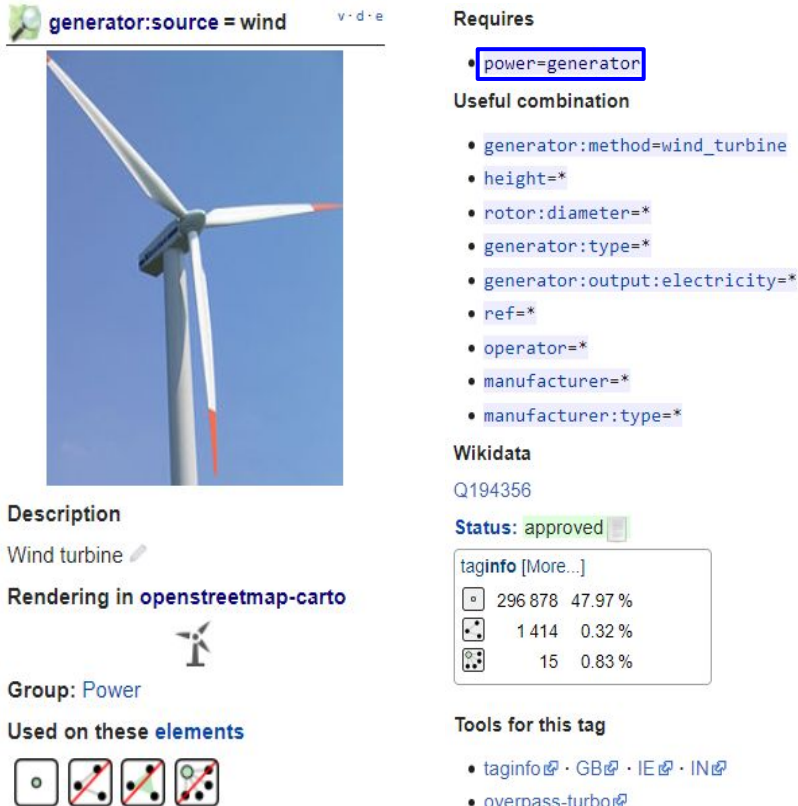
2 km

Loaded - nodes: 20, ways: 1, relations: 0  
Displayed - pois: 1, lines: 0, polygons: 1



<https://wiki.openstreetmap.org/wiki/Tag:generator:source=wind>

Just Google 'osm wiki ...' e.g. 'wind turbine'



The screenshot shows the OpenStreetMap Wiki page for the tag `generator:source=wind`. It features a large image of a wind turbine, a description, a list of requirements, useful combinations, Wikidata information, and tools for the tag.

**generator:source=wind** v · d · e

**Requires**

- `power=generator`

**Useful combination**

- `generator:method=wind_turbine`
- `height=*`
- `rotor:diameter=*`
- `generator:type=*`
- `generator:output:electricity=*`
- `ref=*`
- `operator=*`
- `manufacturer=*`
- `manufacturer:type=*`

**Wikidata**  
Q194356  
**Status:** approved

**taginfo** [More...]

|  |         |         |
|--|---------|---------|
|  | 296 878 | 47.97 % |
|  | 1 414   | 0.32 %  |
|  | 15      | 0.83 %  |

**Tools for this tag**

- [taginfo](#) · [GB](#) · [IE](#) · [IN](#)
- [overpass-turbo](#)
- [Sophox query](#)

1000's of such pages on 'key=value' pairs.



Google 'osm wiki shop' and be impressed :-)

and again, each shop type has a page of its own

'osm wiki bakery'

# Frequency stats? Google 'taginfo power'

← → ↻ 🔒 taginfo.openstreetmap.org/tags



[KEYS](#) · [TAGS](#) · [RELATIONS](#) · [PROJECTS](#) · [REPORTS](#) · [ABOUT](#)

## TAGS

This table shows the most common tags in the database.

| Tag                              | ✖ Objects        | ○ Nodes          | ▣ Ways        | 🗺 Relations  |
|----------------------------------|------------------|------------------|---------------|--------------|
| power=tower                      | 12 748 543 0.18% | 12 748 224 7.91% | 318 0.00%     | 1 0.00%      |
| power=pole                       | 6 754 755 0.10%  | 6 754 586 4.19%  | 162 0.00%     | 7 0.00%      |
| power=generator                  | 1 027 919 0.01%  | 599 143 0.37%    | 427 215 0.06% | 1 561 0.02%  |
| power=line                       | 615 856 0.01%    | 4 0.00%          | 615 736 0.09% | 116 0.00%    |
| power=minor_line                 | 422 416 0.01%    | 95 0.00%         | 422 226 0.06% | 95 0.00%     |
| power=substation                 | 420 657 0.01%    | 75 317 0.05%     | 342 534 0.05% | 2 806 0.03%  |
| power=portal                     | 102 281 0.00%    | 88 064 0.05%     | 14 217 0.00%  | 0 0.00%      |
| power=transformer                | 93 113 0.00%     | 78 425 0.05%     | 14 671 0.00%  | 17 0.00%     |
| power=switch                     | 62 477 0.00%     | 62 309 0.04%     | 168 0.00%     | 0 0.00%      |
| power=catenary_mast              | 32 862 0.00%     | 32 846 0.02%     | 16 0.00%      | 0 0.00%      |
| street_cabinet=power             | 29 951 0.00%     | 25 851 0.02%     | 4 090 0.00%   | 10 0.00%     |
| power=plant                      | 28 078 0.00%     | 432 0.00%        | 23 387 0.00%  | 4 259 0.05%  |
| power=cable                      | 27 235 0.00%     | 15 0.00%         | 27 211 0.00%  | 9 0.00%      |
| power=terminal                   | 25 894 0.00%     | 25 833 0.02%     | 61 0.00%      | 0 0.00%      |
| power=insulator                  | 21 829 0.00%     | 21 770 0.01%     | 59 0.00%      | 0 0.00%      |
| route=power                      | 18 144 0.00%     | 1 0.00%          | 6 0.00%       | 18 137 0.22% |
| power=cable_distribution_cabinet | 10 835 0.00%     | 10 623 0.01%     | 212 0.00%     | 0 0.00%      |
| power=sub_station                | 10 157 0.00%     | 4 609 0.00%      | 5 533 0.00%   | 15 0.00%     |



## plant:source

Source of the energy produced in a power plant

Comparison list (0 items)

Filter: No filter

[XAPI](#) [JOSM](#) [Level0 Editor](#) [Overpass turbo](#)

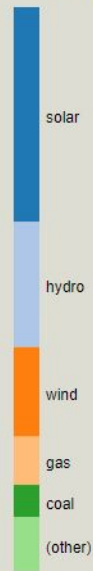
[Overview](#) [Values](#) [Combinations](#) [Similar](#) [Map](#) [Wiki](#) [Projects](#)

### Overview

| Type     | Number of objects |       | Number of values |
|----------|-------------------|-------|------------------|
| All      | 19 756            | 0.00% | 107              |
| Node     | 272               | 0.00% | 16               |
| Way      | 15 603            | 0.00% | 102              |
| Relation | 3 881             | 0.05% | 27               |

Objects with this key were last edited by 1 986 different users.

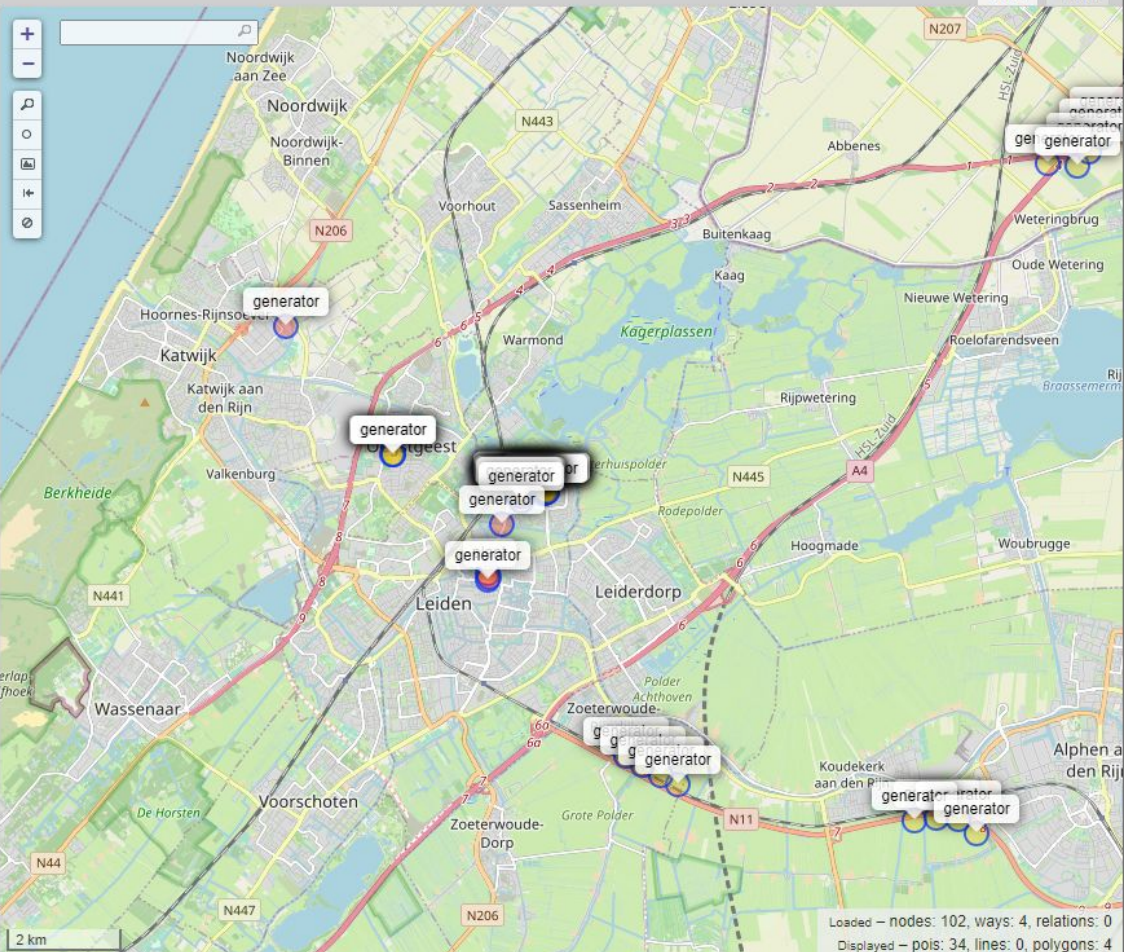
### Distribution of values



```

1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 (
4   nwr[power=plant]({{bbox}})(area.a);
5   nwr[power=generator]({{bbox}})(area.a);
6 );
7 out body;
8 >;
9 out meta qt;
10
11 {{style:
12  node,way,rel
13  { text: power ; }
14 }}

```



We're getting closer, but now there is still too much detail for our purpose, which is a thematic map on wikipedia.

We choose to show solar power on the 'solar farm' level (= plant), not on 'rooftops' level.

For wind we'll want to show each turbine, as these have major impact on the landscape.

Run

Share

Export

Wizard

Save

Load

Logout

Settings

Help

overpass turbo

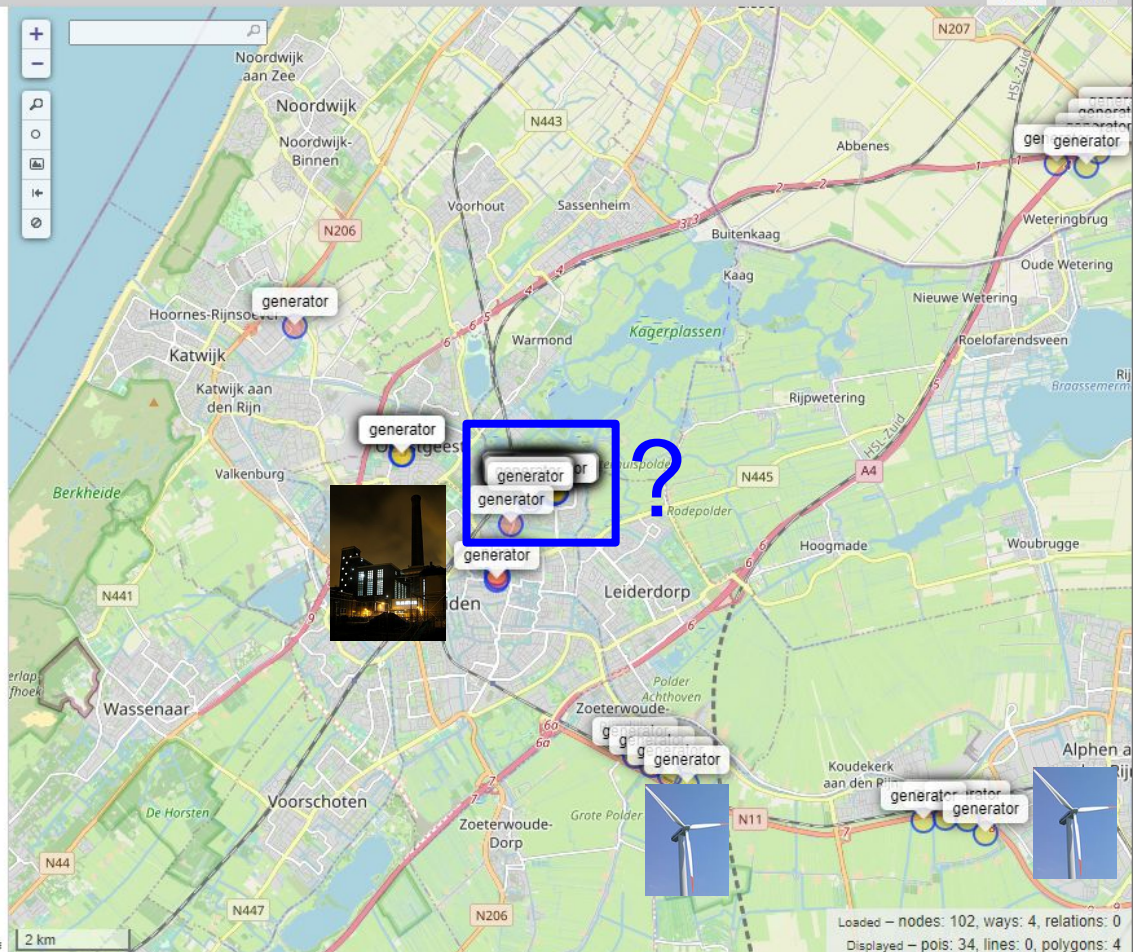
Map

Data

```

1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 (
4   nwr[power=plant]({{bbox}})(.a.a);
5   nwr[power=generator]({{bbox}})(.a.a);
6 );
7 out body;
8 >;
9 out meta qt;
10
11 {{style:
12  node,way,rel
13  { text: power ; }
14 }}

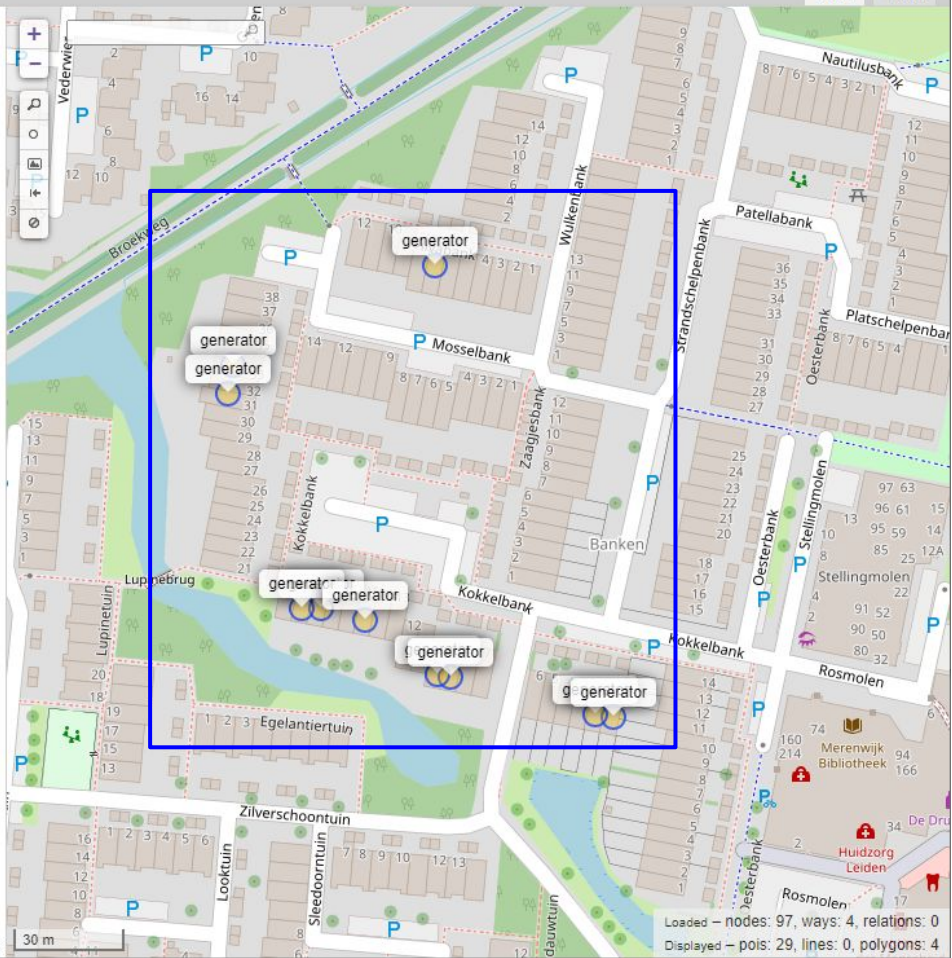
```



```

1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 (
4 nwr[power=generator](area.a)({{bbox}});
5 nwr[power=plant](area.a)({{bbox}});
6 );
7 out body;
8 >;
9 out meta qt;
10
11 {{style:
12 node,way,rel
13 { text: power; }
14 }}

```



Run

Share

Export

Wizard

Save

Load

Logout

Settings

Help

overpass turbo

Map

Data

```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 (
4 nwr[power=generator](area.a){{bbox}});
5 nwr[power=plant](area.a){{bbox}});
6 );
7 out body;
8 >;
9 out meta qt;
10
11 {{style:
12 node,way,rel
13 { text: power; }
14 }}
```

```
253 "generator:source": "solar",
254 "generator:type": "solar_photovoltaic_panel",
255 "location": "roof",
256 "power": "generator"
257 }
258 },
259 {
260 "type": "node",
261 "id": 6879592332,
262 "lat": 52.1778256,
263 "lon": 4.5092564,
264 "tags": {
265 "generator:method": "photovoltaic",
266 "generator:source": "solar",
267 "generator:type": "solar_photovoltaic_panel",
268 "location": "roof",
269 "power": "generator"
270 }
271 },
272 {
273 "type": "node",
274 "id": 6879592333,
275 "lat": 52.1778000,
276 "lon": 4.5090666,
277 "tags": {
278 "generator:method": "photovoltaic",
279 "generator:source": "solar",
280 "generator:type": "solar_photovoltaic_panel",
281 "location": "roof",
282 "power": "generator"
283 }
284 },
285 {
286 "type": "node",
287 "id": 6879592334,
288 "lat": 52.1780962,
289 "lon": 4.5096890,
290 "tags": {
291 "generator:method": "photovoltaic",
292 "generator:source": "solar",
293 "generator:type": "solar_photovoltaic_panel",
294 "location": "roof",
295 "power": "generator"
296 }
297 },
298 {
299 "type": "node",
300 "id": 6879592335,
```





- Main Page
- The map
- Map Features
- Contributors
- Help
- Blogs
- Shop
- Donations
- Wiki discussion
- Recent changes
- Tools
- What links here
- Related changes
- Special pages
- Printable version
- Permanent link
- Page information
- Data item
- Cite this page

# Tag:generator:source=solar

Tag:generator:source=solar - Other languages

čeština · Deutsch · **English** · español · français · 日本語 ·

[Other languages](#) · [Translate](#)

A generator producing energy from the sun, commonly known as a *solar panel*. May form part of a solar power plant.

A solar generator can be either a thermal generator which outputs heat energy or (increasingly commonly) a  [Photovoltaic generator](#).

## Contents [hide]

- 1 Mapping Photovoltaic Generators
  - 1.1 Solar Park/Solar Farm
  - 1.2 Rooftop Solar Panels
- 2 Mapping Thermal Generators
  - 2.1 Concentrated Solar Power Plants
  - 2.2 Solar Thermal Collectors
- 3 See also
- 4 Possible Tagging Mistakes

## Mapping Photovoltaic Generators [\[ edit \]](#) [\[ edit source \]](#)

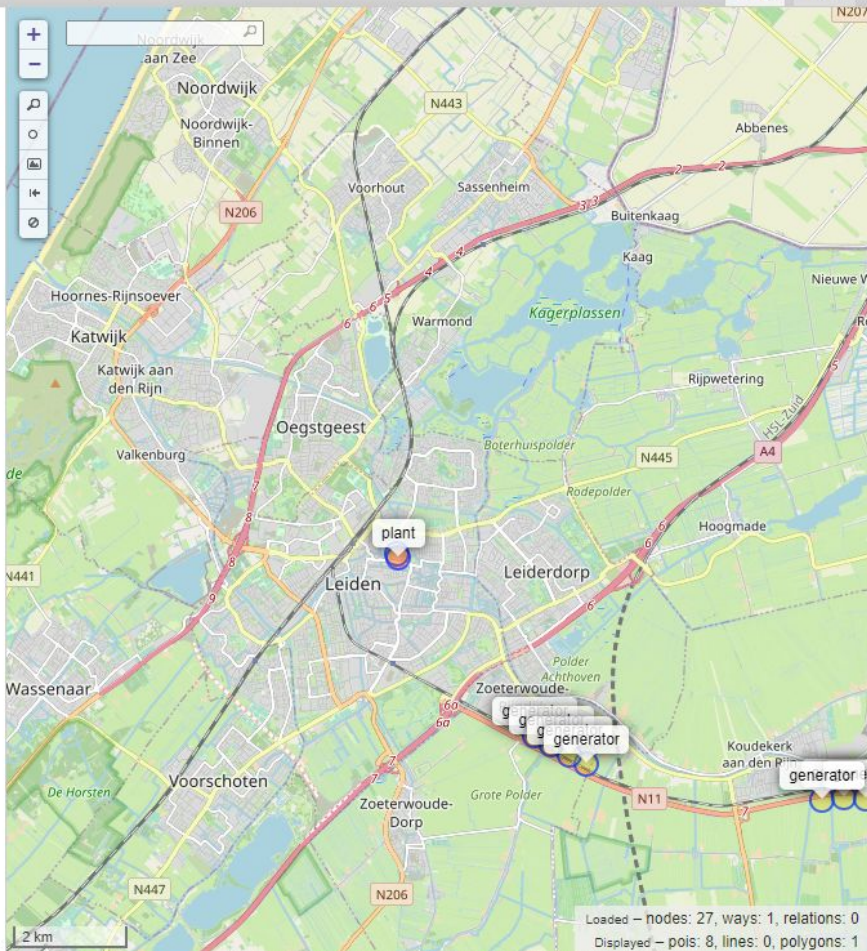
### Solar Park/Solar Farm [\[ edit \]](#) [\[ edit source \]](#)



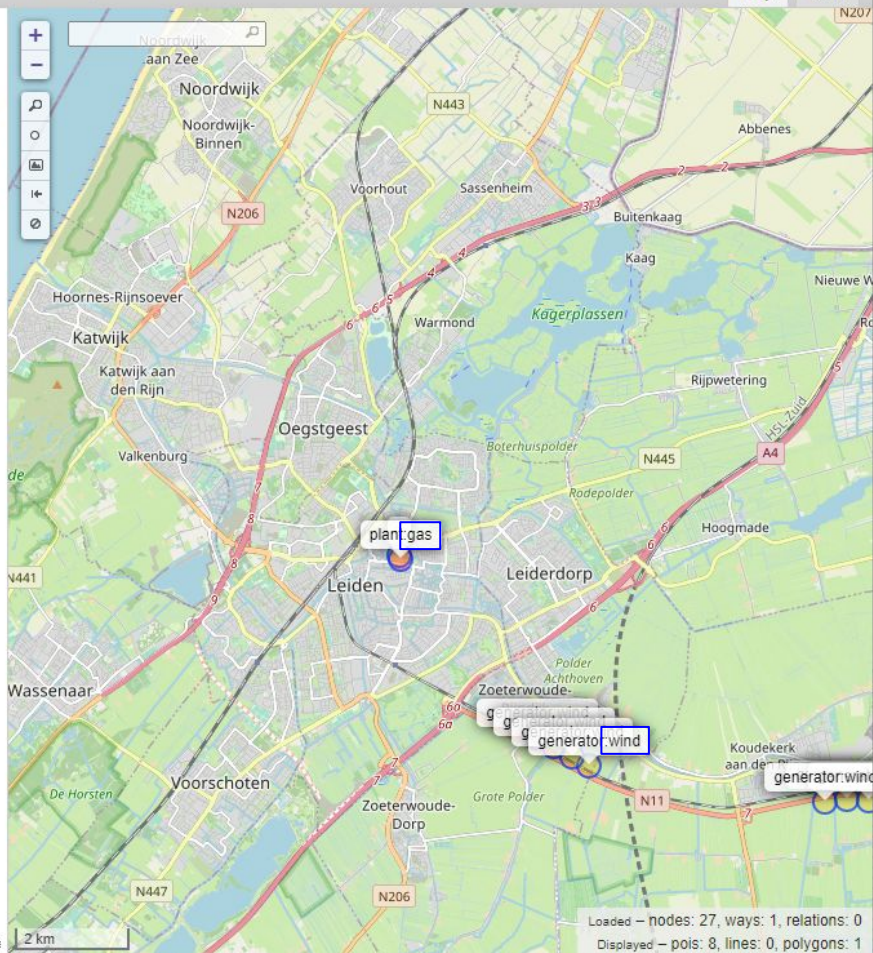
The entire plant is drawn as area , approximately where the fence is.

```
 power=plant
+ plant:source=solar
+ plant:method=photovoltaic
+ plant:output:electricity=*
```

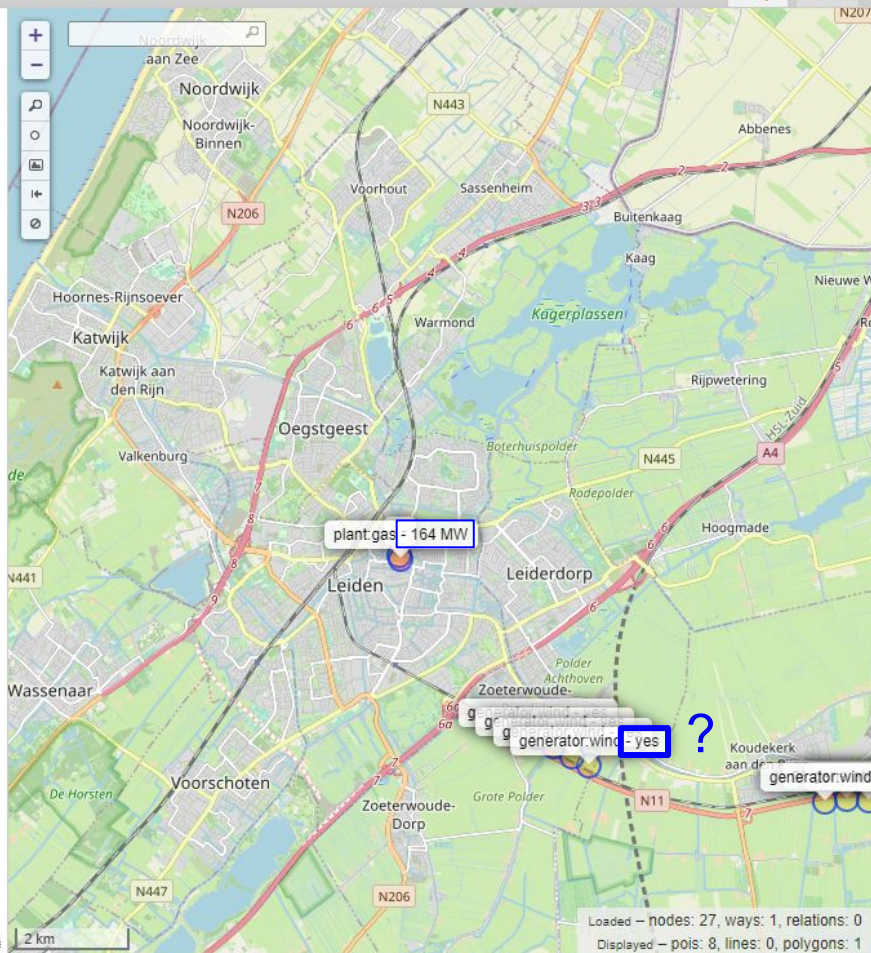
```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 (
4   nwr[power=plant] ["generator:source"="solar"](area.a)({{bbox}});
5
6   nwr[power=generator]["generator:source"="wind"] (area.a)({{bbox}});
7   nwr[power=plant] ["generator:source"="wind"] (area.a)({{bbox}});
8
9   nwr[power=plant] ["plant:source"="gas"] (area.a)({{bbox}});
10 );
11 out body;
12 >;
13 out meta qt;
14
15 {{style:
16  node,way,rel
17  { text: power ; }
18 }}
19
```



```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 (
4   nwr[power=plant] ["generator:source"="solar"](area.a)({{bbox}});
5
6   nwr[power=generator]["generator:source"="wind" ](area.a)({{bbox}});
7   nwr[power=plant] ["generator:source"="wind" ](area.a)({{bbox}});
8
9   nwr[power=plant] ["plant:source"="gas" ](area.a)({{bbox}});
10 );
11 out body;
12 >;
13 out meta qt;
14
15 {{style:
16 node,way,rel
17 {
18   text: eval('tag("power") .
19   "," .
20   tag("generator:source") .
21   tag("plant:source")');
22 }
23 }}
24
```



```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 (
4   nwr[power=plant] ["generator:source"="solar"](area.a){{{bbox}}};
5
6   nwr[power=generator]["generator:source"="wind"] (area.a){{{bbox}}};
7   nwr[power=plant] ["generator:source"="wind"] (area.a){{{bbox}}};
8
9   nwr[power=plant] ["plant:source"="gas"] (area.a){{{bbox}}};
10 );
11 out body;
12 >;
13 out meta qt;
14
15 {{style:
16 node,way,rel
17 {
18   text: eval('tag("power") .
19   " ." .
20   tag("generator:source") .
21   tag("plant:source") .
22   "-" .
23   tag("generator:output:electricity").
24   tag("plant:output:electricity")');
25 }}
26
```





W Himalayas x W Beverken v x W Help:Extens x W Module:OS x W Albert Heij x W OSM Tree l x W Tag:power x W Key:plant: x +

wiki.openstreetmap.org/wiki/Key:plant:output

English Infodisic Talk Preferences Watchlist Contributions Log out

Page Discussion Read Edit Edit source View history Search OpenStreetMap Wiki

## Key:plant:output

(Redirected from Key:plant:output:electricity)

**Key:plant:output - Other languages** Purge · Help

čeština · Deutsch · English · français · 日本語 · **Other languages · Translate**

Power plants can output multiple forms of energy. For example, a hydro plant will only output electricity, but a combined heat and power plant will typically output hot water and electricity. Add as many values as are relevant.


If the rating of the output is known, add a numerical value followed by the unit. Otherwise simply add yes. This value should be the peak rated output capacity (sometimes listed as **MWp** or **MWe** for power plants).

|  |                  |                            |
|--|------------------|----------------------------|
| <code>plant:output:electricity</code>    | <code>yes</code> | x W / x kW / x MW / x GW   |
| <code>plant:output:hot_water</code>      | <code>yes</code> | / x W / x kW / x MW / x GW |
| <code>plant:output:hot_air</code>        | <code>yes</code> | / x W / x kW / x MW / x GW |
| <code>plant:output:cold_water</code>     | <code>yes</code> | / x W / x kW / x MW / x GW |
| <code>plant:output:cold_air</code>       | <code>yes</code> | / x W / x kW / x MW / x GW |
| <code>plant:output:compressed_air</code> | <code>yes</code> | / x W / x kW / x MW / x GW |
| <code>plant:output:steam</code>          | <code>yes</code> | / x W / x kW / x MW / x GW |
| <code>plant:output:vacuum</code>         | <code>yes</code> | / x W / x kW / x MW / x GW |

**See also** [ edit | edit source ]

- `power=plant`
- `generator:output=*`
- original proposal for this key
- `plant:storage=*`

**plant:output** v · d · e



**Description**  
Forms of power generated in a power plant and the rating if known ℹ

**Group:** Power

**Used on these elements**

📍 🏠 🏢 🏭

**Requires**

- `power=plant`

**Wikidata**  
Search Wikidata  
**Status:** approved ℹ

taginfo [ More... ]

📄 0  
👤 67  
🗺 1

**Tools for this tag**

- taginfo ℹ · GB ℹ · IE ℹ · IN ℹ
- overpass-turbo ℹ

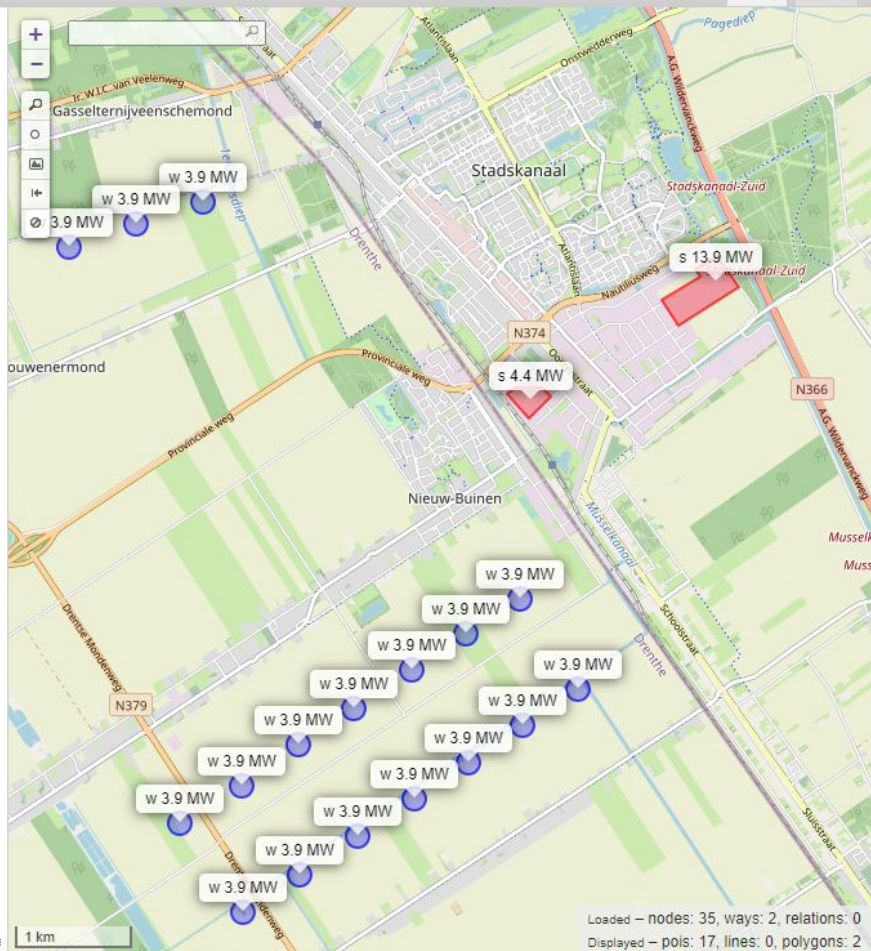
Categories: Key descriptions by group | Key descriptions | Key descriptions with status "approved" | Power | Keys indicating power

Two final examples on a municipal level.

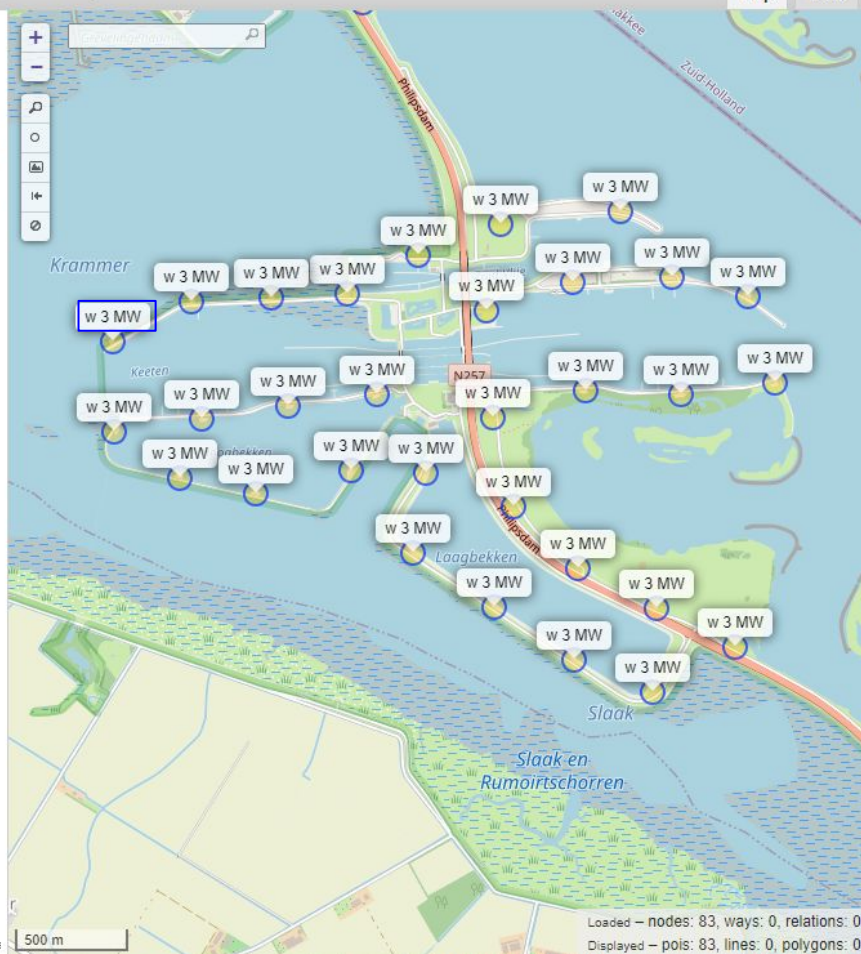
```

1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 (
4   nwr[power=plant][plant:source="solar"](area.a){{{bbox}}};
5
6   nwr[power=generator][generator:source="wind"] (area.a){{{bbox}}};
7   nwr[power=plant] [generator:source="wind"] (area.a){{{bbox}}};
8
9 );
10 out body;
11 >;
12 out meta qt;
13
14 {{style:
15   node[generator:source=wind],
16   way[generator:source=wind],
17   rel[generator:source=wind],
18   node[plant:source=wind],
19   way[plant:source=wind],
20   rel[plant:source=wind]
21   {
22     text:eval('w ');
23     tag("generator:output:electricity") .
24     tag("plant:output:electricity"); color:blue ; fill-color:blue;
25   }
26
27   node[plant:source=solar],
28   way[plant:source=solar],
29   rel[plant:source=solar]
30   {
31     text:eval('s ');
32     tag("plant:output:electricity"); color:red ; fill-color:red;
33   }
34 }}
35
36

```



```
1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 (
4   nwr[power=plant] ["generator:source"="solar"](area.a)({{bbox}});
5
6   nwr[power=generator]["generator:source"="wind"] (area.a)({{bbox}});
7   nwr[power=plant] ["generator:source"="wind"] (area.a)({{bbox}});
8
9   nwr[power=plant] ["plant:source"="gas"] (area.a)({{bbox}});
10 );
11 out body;
12 >;
13 out meta qt;
14
15 {{style:
16   node[generator:source=wind],
17   way[generator:source=wind],
18   rel[generator:source=wind]
19   {
20     text:eval('w ' .
21     tag("generator:output:electricity") .
22     tag("plant:output:electricity"));
23   }
24 }}
25
```





```

1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power](area.a);
4 out body;
5 >;
6 out meta qt;

```


```

1 [out:json][timeout:250];
2 area[name="Nederland"]->.a;
3 nwr[power]({{bbox}}(area.a));
4 out body;
5 >;
6 out meta qt;
7

```

Bounding box code: ({{bbox}})

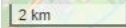


If you do not specify ({{bbox}}) some results may be outside your current map section. Click looking glass icon  to reposition map, and see all results.

Specifying ({{bbox}}) can help a lot during testing.

Zoom in on a small section of the map, run tests within a few seconds [timeout=25], till you fixed all bugs and are satisfied with the results.

Zoom out so that your target area is fully visible, and run the script again, which now may take much longer to run. [timeout=250]



I saved the best for last.

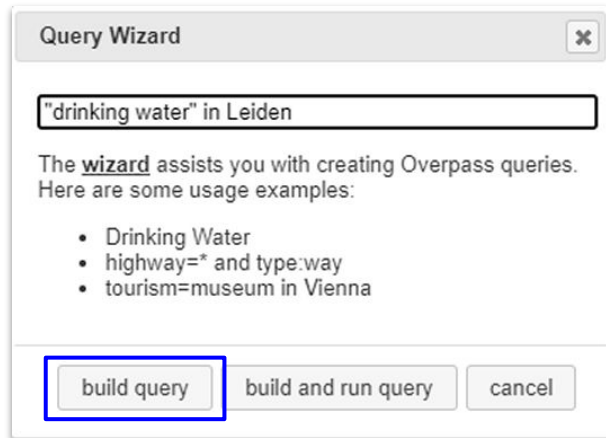
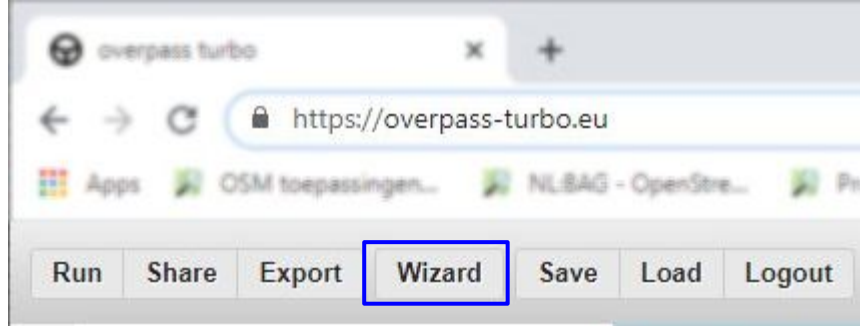
For simple queries the wizard makes it very easy for you.  
But now you know how to take it further from there.

Also sometimes the wizard is a bit confused ;-)

(Oct 2020 'monastery in Nederland' leads to  
**["amenity"="place\_of\_worship"]["religion"="taoist"]**)

You know how to fix this.

# Overpass Wizard

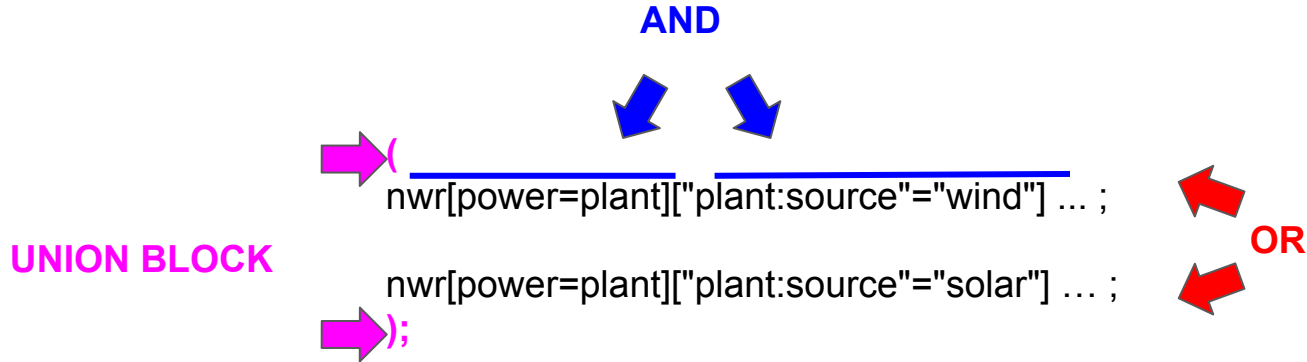


```
1 /*
2 This has been generated by the overpass-turbo wizard.
3 The original search was:
4 "drinking water" in Leiden "
5 */
6 [out:json][timeout:25];
7 // fetch area "Leiden" to search in
8 {{geocodeArea:Leiden}}->.searchArea;
9 // gather results
10 (
11 // query part for: "drinking water"
12 node["amenity"="drinking_water"](area.searchArea);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

# AND vs OR

AND = Multiple selectors in one statement (both should be true)

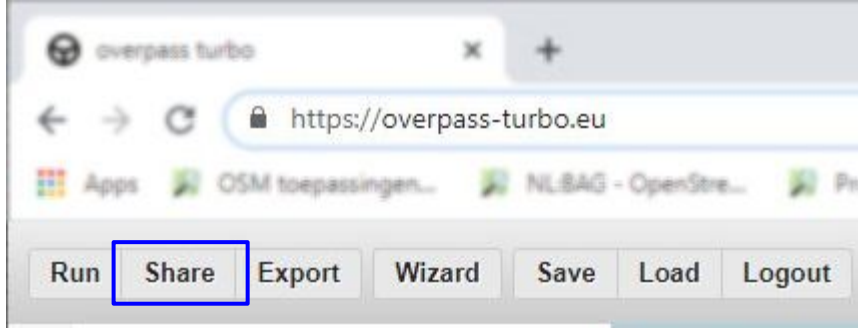
OR = Multiple selectors in separate statements (at least one should be true)



Can all be on one line:

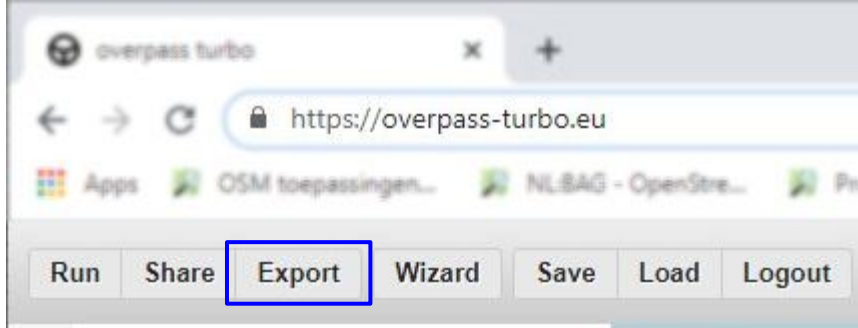
```
(node [name='Foo']; way[name='Foo']); UNION BLOCK , OR
```

# Export options



To be extended

# Export options



To be extended

# Save / Load



To be extended

meta = more



Many other tools for data mining

e.g.

Wikidata + SPARQL = Wikidata Query Service

OSM + Wikidata + SPARQL = Sophox

# MapCSS

Limited syntax

Unchanged since 2015

Also supported by JOSM, and many more clients, but different 'dialects' for different clients

Syntax of Overpass QL and MapCSS differs (e.g. on regular expressions)

One particular caveat:

'Some spaces are significant', meaning an extra space can be too much

```
node [power=plant]    is fine
node [power=plant]   doesn't parse in MapCSS
```

And error message just says 'There is an error' (So tweak and test script in tiny increments)

## Other presentations

- 'A turbo introduction to Overpass', by Minh Nguyễn.

Excellent talk, different focal points.

<https://www.youtube.com/watch?v=q9QI4AfwHoM>

My presentation is even a bit more introductory at places. And has added relevance for Wikipedia. But I highly recommend to watch Minh Nguyễn's video and deepen your understanding.

- [Map making workshop – from Wikidata to interactive off-wiki maps in three steps](#)

by Olaf Janssen

Totally different approach

<https://www.slideshare.net/OlafJanssenNL/slidedecmap-making-workshop-from-wikidata-to-interactive-offwiki-maps-in-three-steps>

Thank you  
Any questions?  
(answers not guaranteed)

## Images used

Power catenary mast: [@Acabashi CC-BY SA 4.0](#)

Power tower: [public domain](#)

Power substation: [@David Neal CC-BY SA 3.0](#)

Power cable: [@Rasbak GDFL 1.2](#)

Power portal: [@François Lacombe CC-BY SA 2.0](#)

Power convertor: [@Marshelec CC-BY SA 3.0](#)

Power plant: [Marque1313](#) public domain

Power generator: [@Dirk Ingo Franke CC-BY SA 2.0](#)

Power plant Leiden: [@Erik Zachte CC-BY SA 3.0](#)

OK icon: [MJL & House](#) public domain

Renewable Energy on the Grid.jpg: [@Kenueone Creative Commons Zero](#)

Nuclear Plants on Wikipedia (2008): [Pawelpanas](#) public domain

## Web addresses

Dutch OSM forum: <https://forum.openstreetmap.org/viewforum.php?id=12>

Taginfo: <https://taginfo.openstreetmap.org/>

## OSM wiki pages

Just as some Wikipedia pages are mostly targeting experts (think 'math'), some Overpass info on OSM wiki is also a tough nut to crack. Extra complication is that part of the docs still focus on the first older Overpass language 'xml'. The newer and more concise language QL is a bit less tedious, but very compact.

I remember a keynote (by whom?) where it was explained that 'print' in xml had been replaced by 'out' in QL "for brevity". The error messages still refer to 'print' ;-)

This main reference page [https://wiki.openstreetmap.org/wiki/Overpass\\_API/Overpass\\_QL](https://wiki.openstreetmap.org/wiki/Overpass_API/Overpass_QL) is not so easy at places. But you can visit and revisit and every time learn a bit more.

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

On tags, key 'power': <https://wiki.openstreetmap.org/wiki/Key:power>  
just type 'osm wiki power' in Google

Query examples: [https://wiki.openstreetmap.org/wiki/Overpass\\_API/Overpass\\_API\\_by\\_Example](https://wiki.openstreetmap.org/wiki/Overpass_API/Overpass_API_by_Example)

On Overpass shortcuts:

[https://wiki.openstreetmap.org/wiki/Overpass\\_turbo/Extended\\_Overpass\\_Turbo\\_Queries](https://wiki.openstreetmap.org/wiki/Overpass_turbo/Extended_Overpass_Turbo_Queries)

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