

Overpass API

Overpass API

An API to query OpenStreetMap data.



User:Binnette



2021-12-10



License MIT

1 Intro



What is Overpass API*?

<http://overpass-api.de/>

- A **read only** API that serves custom selected parts of OSM data
- It has 2 query languages:
 - **Overpass XML**: very verbose but trully explicit
 - **Overpass QL**: lightweight but a bit magic
- It can export data to JOSM, OSM data file, CSV, GeoJSON, ...

API = Application Programming Interface



Technical details

- Overpass API do not query OSM database directly. Instead it queries a clone of the main database based on a planet file (export of OSM database)
 - 🕒 It explains why data are not always up-to-date in Overpass API
- Overpass API is an ✅ OpenSource software and have many instances:
 - Main instance: <https://overpass-api.de/api/interpreter>
 - And a lot of others: 🇫🇷 OSM-fr, 🇨🇭 OSM-ch, 🇷🇺 OSM-ru, 🇹🇼 OSM-tw, ...

https://wiki.openstreetmap.org/wiki/Overpass_API#Public_Overpass_API_instances



Overpass query languages

```
/* Overpass QL: lightweight and implicit */  
(  
  node(51.249,7.148,51.251,7.152);  
  <;  
);  
out meta;
```

```
<!-- XML: verbose but explicit -->  
<union>  
  <bbox-query s="51.249" w="7.148" n="51.251" e="7.152"/>  
  <recurse type="up"/>  
</union>  
<print mode="meta"/>
```



Overpass XML 'almost' explicit...

```
<union>
  <bbox-query s="51.249" w="7.148" n="51.251" e="7.152"/>
  <recurse type="up"/>
</union>
<print mode="meta"/>
```

```
<!-- The exact same query but even more verbose... -->
<union into="_">
  <bbox-query s="51.249" w="7.148" n="51.251" e="7.152"/>
  <recurse from="_" into="_" type="up"/>
</union>
<print e="" from="_" geometry="skeleton" ids="yes" limit="" mode="meta" n="" order="id" s="" w=""/>
```



Overpass XML 'full' explicit

```
<union into="_">
  <bbox-query s="51.249" w="7.148" n="51.251" e="7.152"/>
  <recurse from="_" into="_" type="up"/>
</union>
<print e="" from="_" geometry="skeleton" ids="yes" limit="" mode="meta" n="" order="id" s="" w=""/>
```

```
<!-- The exact same query but over verbose... -->
<union into="_">
  <query into="_" type="node">
    <bbox-query s="51.249" w="7.148" n="51.251" e="7.152"/>
  </query>
  <recurse from="_" into="_" type="up"/>
</union>
<print e="" from="_" geometry="skeleton" ids="yes" limit="" mode="meta" n="" order="id" s="" w=""/>
```



Let convert it back to Overpass QL...

```
<union into="_">
  <query into="_" type="node">
    <bbox-query s="51.249" w="7.148" n="51.251" e="7.152"/>
  </query>
  <recurse from="_" into="_" type="up"/>
</union>
<print e="" from="_" geometry="skeleton" ids="yes" limit="" mode="meta" n="" order="id" s="" w=""/>
```

```
(
  node(51.249,7.148,51.251,7.152);
  <;
);
out meta;
```

So now, you understand better the X of eXtensible in XML. 🤖

Overpass-Turbo: a tool to call Overpass API

<http://overpass-turbo.eu>

Still the same query 😊:

- In Overpass QL: <http://overpass-turbo.eu/s/1e1F>
- In Overpass XML: <http://overpass-turbo.eu/s/1e1G>

Use the 'Export' button to convert to QL or XML...



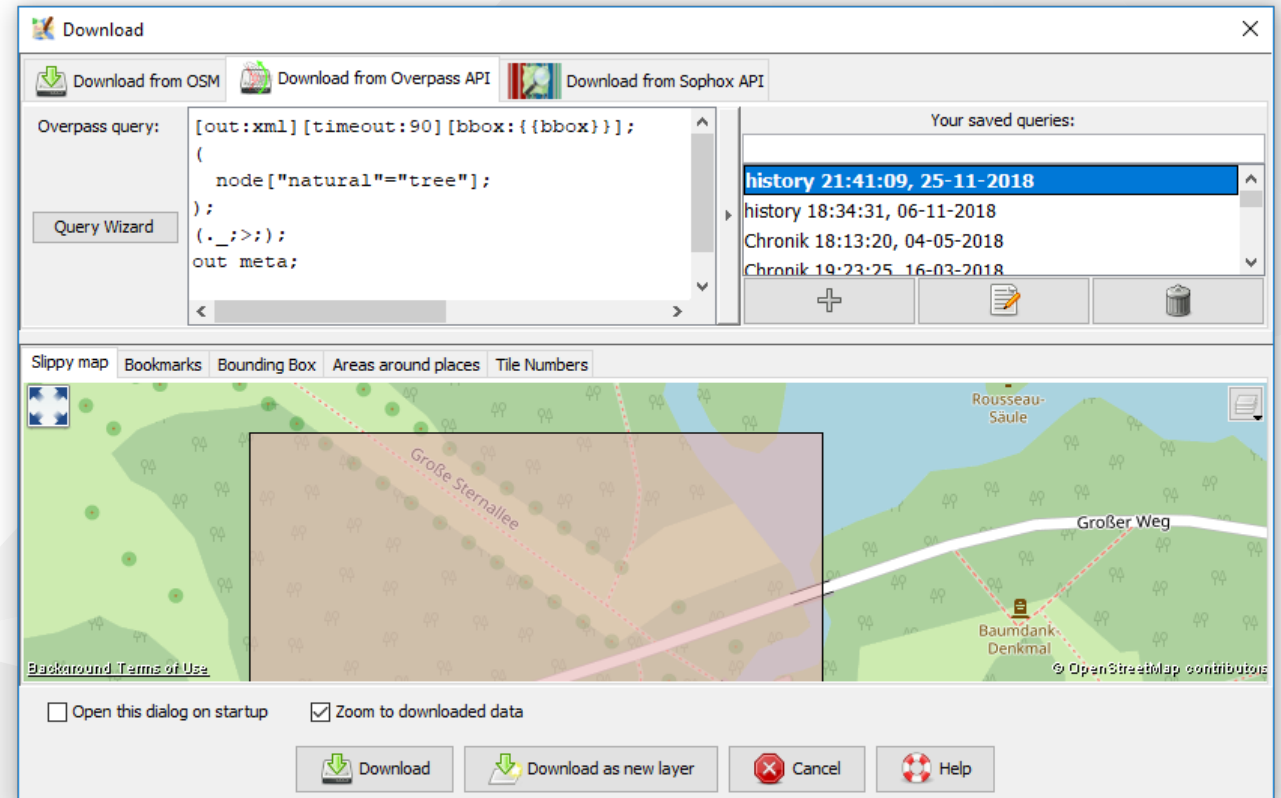
From Overpass-Turbo to JOSM

- Launch JOSM and enable the "Remote control":
<https://josm.openstreetmap.de/wiki/Help/Preferences/RemoteControl>
- Go to Overpass Turbo and run your query
- Click on Export > Data > JOSM
- Your data are loaded in JOSM



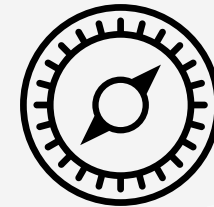
Overpass API directly in JOSM

- Open JOSM setting
- Enable 'Advanced mode'
- Click on download button
- Tab 'Download from Overpass API' is displayed



From Overpass API to Maproulette

Why not creating MapRoulette challenge by querying wrong or incomplet OSM data with Overpass API?



MapRoulette

2 Back to the basics

Helloworld with Overpass

Get "drinking water" for a given "bbox":

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

Try it for free!!!  <http://overpass-turbo.eu/s/1e1S>



Instruction 'area' or how to get a zone by its name

```
area[name="France métropolitaine"]->.FRM;  
area["ISO3166-2"="FR-ARA"]->.ARA; // Auvergne-Rhône-Alpes  
area[name="Grenoble-Alpes Métropole"]->.LaMetro;  
area[name="Grenoble"]->.Grenoble;  
area["ISO3166-2"="FR-07"]->.Ardèche;  
area[name="Mariac"]->.Mariac;
```



Get only amenity nodes

```
area[name="Mariac"]->.a;  
node[amenity](area.a);  
out geom;
```

<http://overpass-turbo.eu/s/1e1U>



Get amenity nodes and way

```
area[name="Mariac"]->.a;  
nw[amenity](area.a);  
out geom;
```

<http://overpass-turbo.eu/s/1e1V>



Get amenity nodes and way... again

```
area[name="Mariac"]->.a;  
nw[amenity](area.a);  
out geom;
```

<http://overpass-turbo.eu/s/1e1V>

```
area[name="Mariac"]->.a;  
(  
  node[amenity](area.a);  
  way[amenity](area.a);  
);  
out geom;
```

<http://overpass-turbo.eu/s/1e1X>



Get amenity nodes, way and relation

```
area[name="Mariac"]->.a;  
nwr[amenity](area.a);  
out geom;
```

<http://overpass-turbo.eu/s/1e1Y>

```
area[name="Mariac"]->.a;  
(  
  node[amenity](area.a);  
  way[amenity](area.a);  
  rel[amenity](area.a);  
);  
out geom;
```

<http://overpass-turbo.eu/s/1e1X>



DEMO TIME

All my overpass queries:

<https://wiki.openstreetmap.org/wiki/User:Binnette/OverpassQueries>