OpenSchoolMaps: 02 Obtain OSM data

OpenSchoolMaps.ch - Free learning materials for free geodata and maps

A guide for enthusiasts, students and instructors/teachers

Overview

Goal

The goal of this guide is to show how to get OpenStreetMap data and use it in QGIS 3.

Audience

Anyone who is interested in creating their own map (online, as a printable plan) and in analysing geodata. If you don’t know OpenStreetMap yet, you should consult the worksheets "Using the OSM.org website in everyday life" and "Investigating OpenStreetMap data".

Scheduling

This worksheet requires different amounts of time depending on your previous knowledge, since you can skip individual sections if you are already familiar with the topic in question.

Structure

The following topics will be covered:

- Introduction with explanations of OpenStreetMap and geoinformation systems as well as common file formats
- Extract OpenStreetMap data with Overpass
- Import OpenStreetMap data directly into QGIS with QuickOSM plugin
- Various web tools for downloading OpenStreetMap data
- Merge Polygon Layers with Point Layers in QGIS
- Filter or rule-based display of objects with QGIS

We recommend at least reading the introduction and extracting the chapter OpenStreetMap data with overpass.

> This tutorial is about using existing data from OpenStreetMap. If you want to record data yourself, you can use the worksheets "Edit OpenStreetMap" and "Create a thematic online map with uMap" as well as https://learnosm.org/de/ for more information.

Introduction

The data from OpenStreetMap comes in different data formats. Therefore we will first explain some of these file formats in more detail.

OpenStreetMap and geoinformation systems

The geometry of the objects in OpenStreetMap (OSM) are stored internally as topological structures (nodes, ways). And the object attributes as tags (key-value pairs) only partially structured. This data schema, which can be extended at will, must first be converted for geoinformation systems (GIS) and their formats. There is no single GIS data schema for OSM. Hardly any export from OSM is therefore the same as the other.

File formats

Basically, one can distinguish between OSM's own and analyzable GIS formats.

- OSM-specific formats are mainly OSM XML, and OSM PBF.
- GIS compatible formats are GeoJSON, GeoPackage, Shapefiles, GeoJSON.
- Other GIS vector formats are: KML and GPX for GPS data exchange.
- Other formats that cannot be analysed further are image/raster formats such as PNG, JPEG and PDF.

The following describes various methods for importing OpenStreetMap data into QGIS. The first method works with Overpass-Turbo, a web service for querying OpenStreetMap data. We recommend that you look at this method as it is the most efficient of all the methods presented. The second method is based on QuickOSM, a plugin for QGIS in which queries can be created. If you've read or read the overpass turbo section, you can consider it optional with QuickOSM.

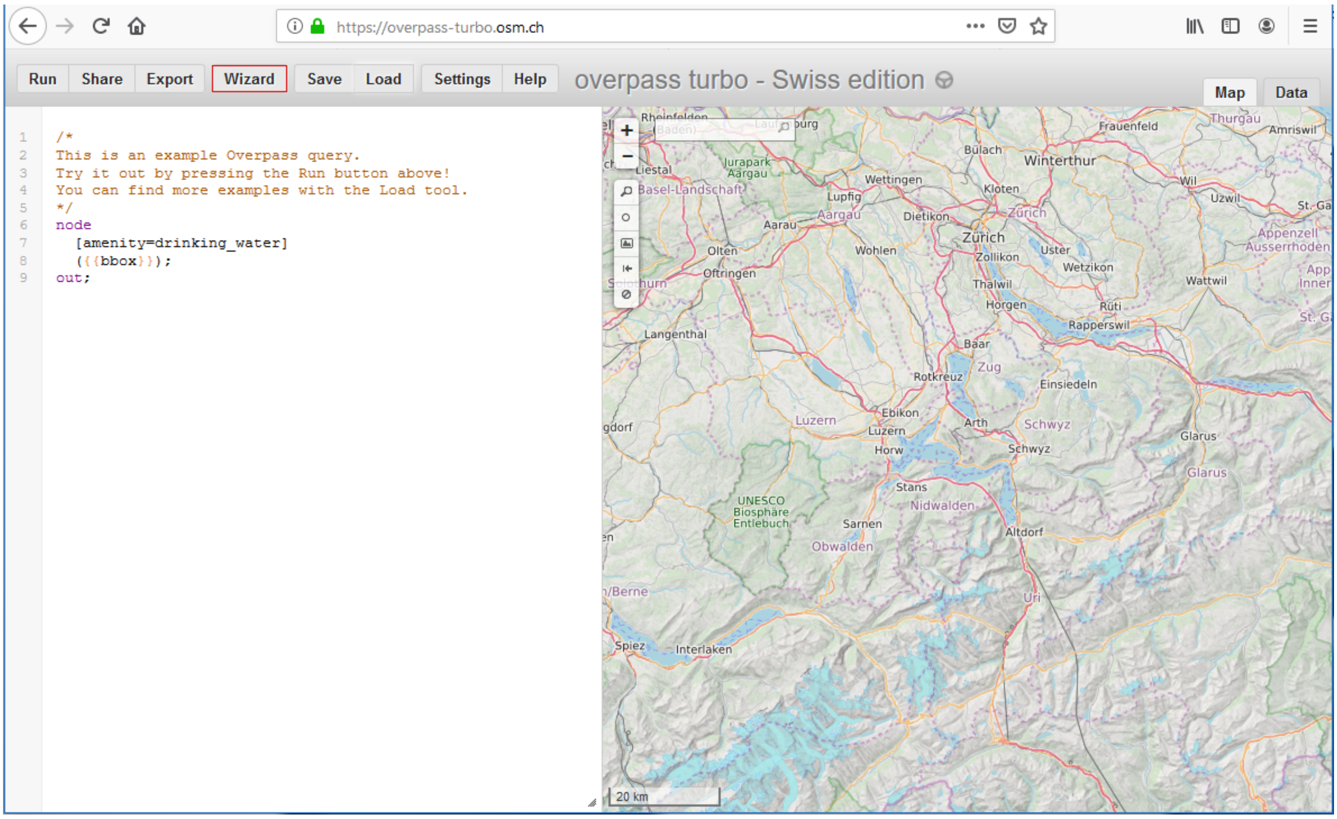
After the two inport methods further tools are introduced, for example the HOT Export Tool or OSMaxx.

Finally, you will learn how to merge different layers in QGIS. This chapter is referred to at the end of the remaining chapters, so you'll come there for or after.

Extract OpenStreetMap data with Overpass

The easiest and fastest way to import OpenStreetMap data into QGIS is by Overpass-Turbo. This method is suitable for advanced as well as for beginners. First visit the Overpass-Turbo website at: http://overpass-turbo.osm.ch/.

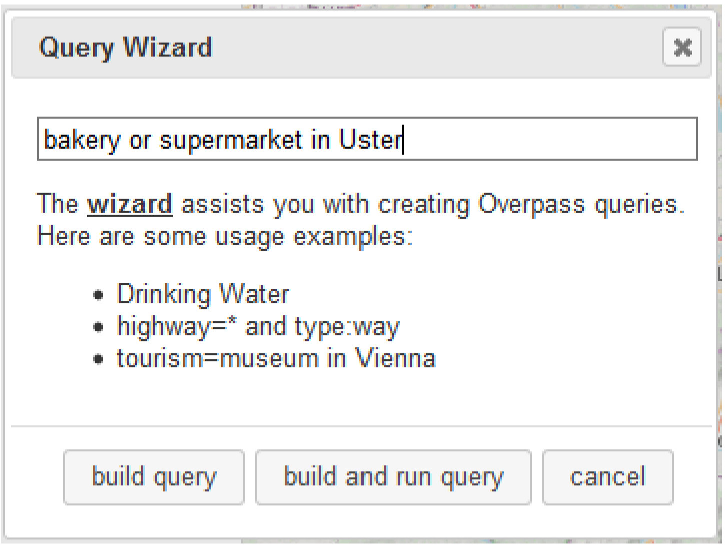
When you first visit the website, it looks something like this:



```
/*
This is an example Overpass query.
Try it out by pressing the Run button above!
You can find more examples with the Load tool.
*/
node
  [amenity=drinking_water]
  ({{bbox}});
out;
```

Figure 1. Homepage of Overpass-Turbo

Click on the red marked "Wizard" button. The wizard makes it easy to create overpass queries. Since we are looking for bread shops as an example, we are looking for bakeries and supermarkets as both sell bread. As a locality we take Uster as an example. The query looks like this (*bakery or supermarket in Uster*):



Query Wizard

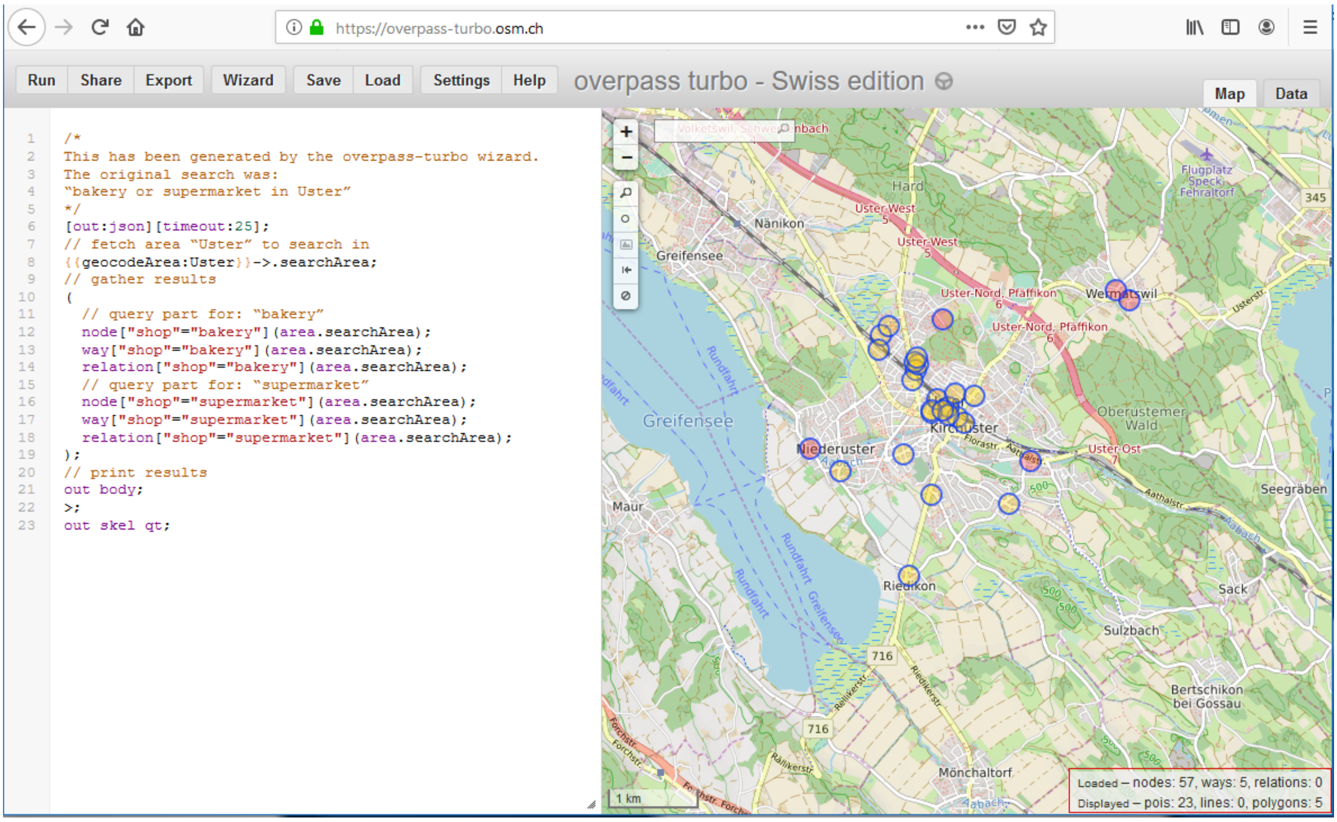
bakery or supermarket in Uster

The **wizard** assists you with creating Overpass queries. Here are some usage examples:

- Drinking Water
- highway=* and type:way
- tourism=museum in Vienna

Figure 2. Overpass Turbo Wizard Query

After clicking on "build and run query" you will get the following:



```
/*
This has been generated by the overpass-turbo wizard.
The original search was:
“bakery or supermarket in Uster”
*/
[out:json][timeout:25];
// fetch area “Uster” to search in
{{geocodeArea:Uster}}->.searchArea;
// gather results
(
  // query part for: “bakery”
  node["shop"="bakery"](area.searchArea);
  way["shop"="bakery"](area.searchArea);
  relation["shop"="bakery"](area.searchArea);
  // query part for: “supermarket”
  node["shop"="supermarket"](area.searchArea);
  way["shop"="supermarket"](area.searchArea);
  relation["shop"="supermarket"](area.searchArea);
);
// print results
out body;
>;
out skel qt;
```

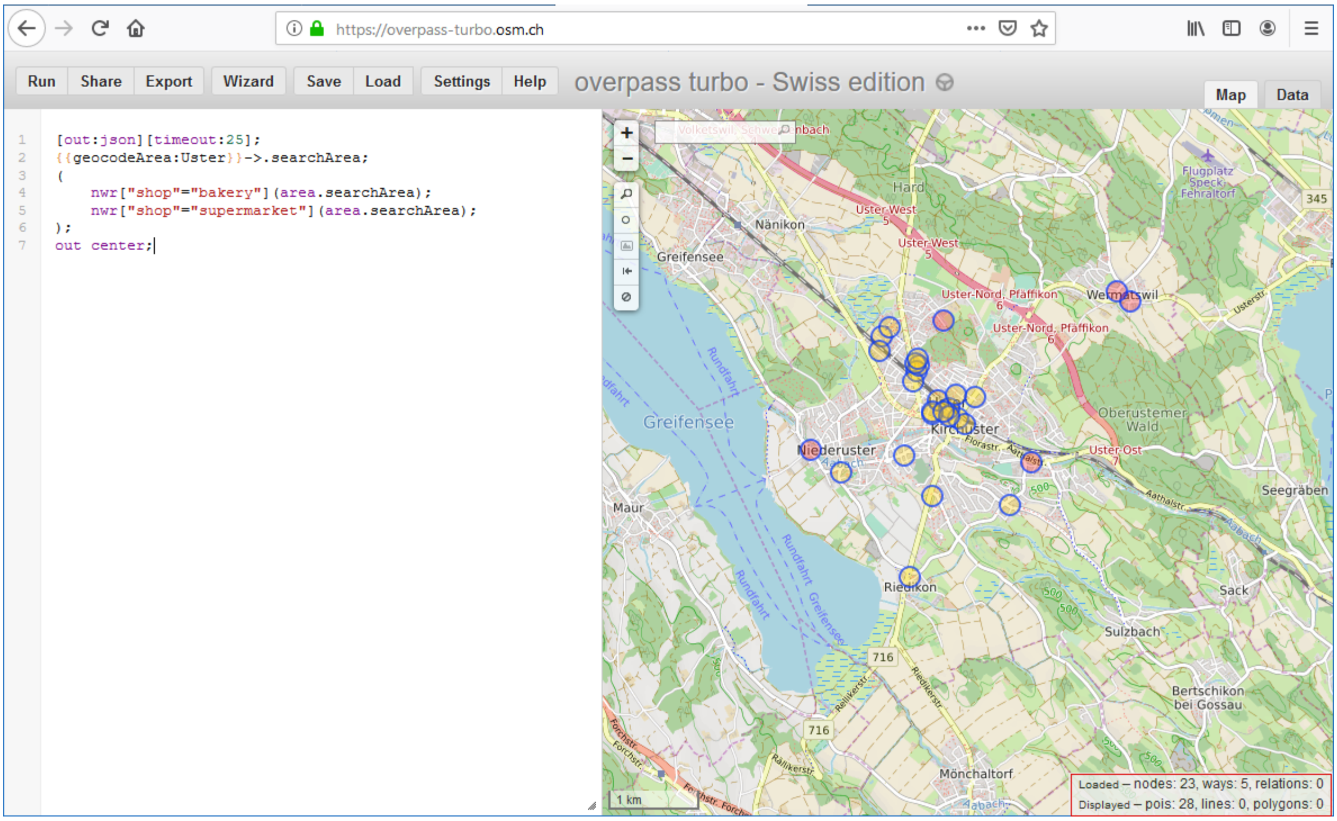
Loaded – nodes: 57, ways: 5, relations: 0
Displayed – pois: 23, lines: 0, polygons: 5

Figure 3. Result of the Wizard Query

The above result can also be simplified by replacing the "nodes", "ways" and "relations" to "nwr" (=nodes/ways/relations). In the lower right corner you can see the different objects marked in red. We have 23 POIs and 5 polygons. Since we only want POIs, we could use "out center;" to turn all areas and paths into points. The new query should look like this.

```
[out:json][timeout:25];
{{geocodeArea:Uster}}->.searchArea;
(
    nwr["shop"="bakery"](area.searchArea);
    nwr["shop"="supermarket"](area.searchArea);
);
out center;
```

If we now copy the upper lines and paste them into the overpass turbo as a query and execute them, we get the following:



```
[out:json][timeout:25];
{{geocodeArea:Uster}}->.searchArea;
(
    nwr["shop"="bakery"](area.searchArea);
    nwr["shop"="supermarket"](area.searchArea);
);
out center;
```

```
Loaded – nodes: 23, ways: 5, relations: 0
Displayed – pois: 28, lines: 0, polygons: 0
```

Figure 4. Revised result

Now we see in the lower right corner that the displayed objects consist only of POIs. We now have 28 POIs and 0 polygons, because we have converted the 5 polygons to POIs via "out center" and Overpass has added these 5 new POIs to the existing 23 POIs. Now we have to save the query as GeoJSON. To do this, go to "Export" in the upper left corner of "Wizard". The window that appears gives us various export options: We choose the top one, i.e. "download/copy as GeoJSON".

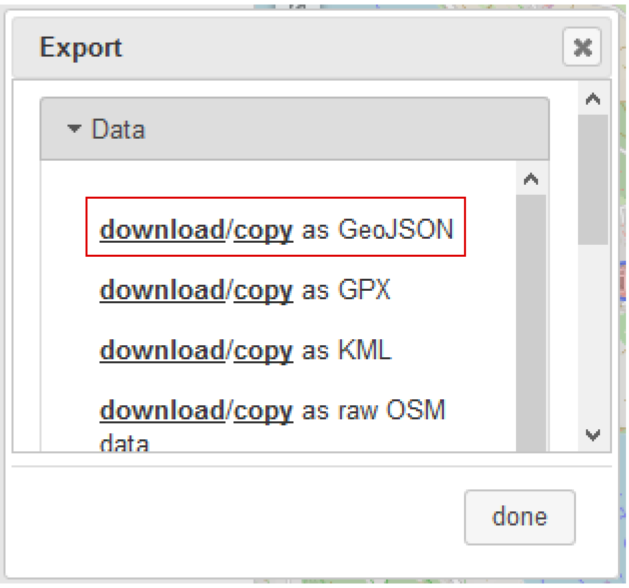


Figure 5. Overpass export

We now add the downloaded GeoJSON file to the left layer menu in QGIS using Drag & Drop. In addition, we rename the layer appropriately, for example "Bread shops".

Our QGIS now looks like this:

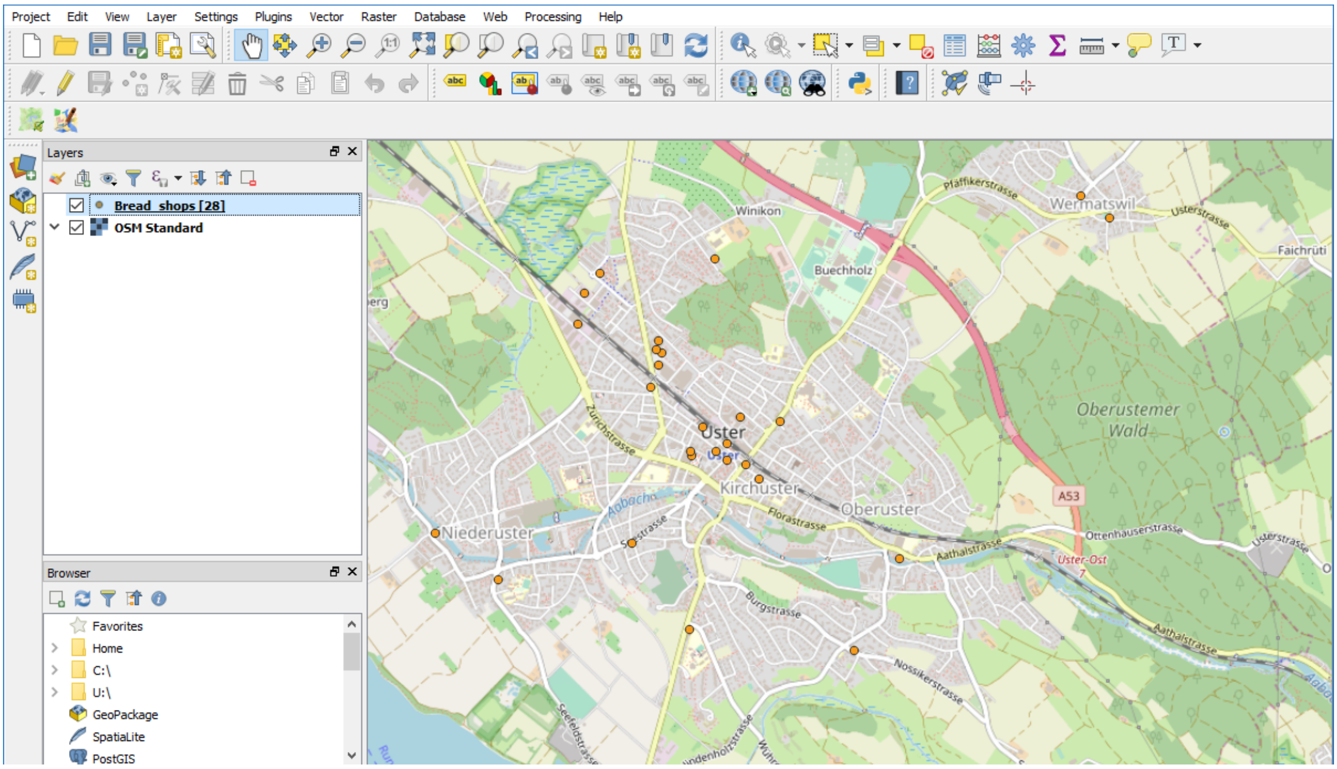
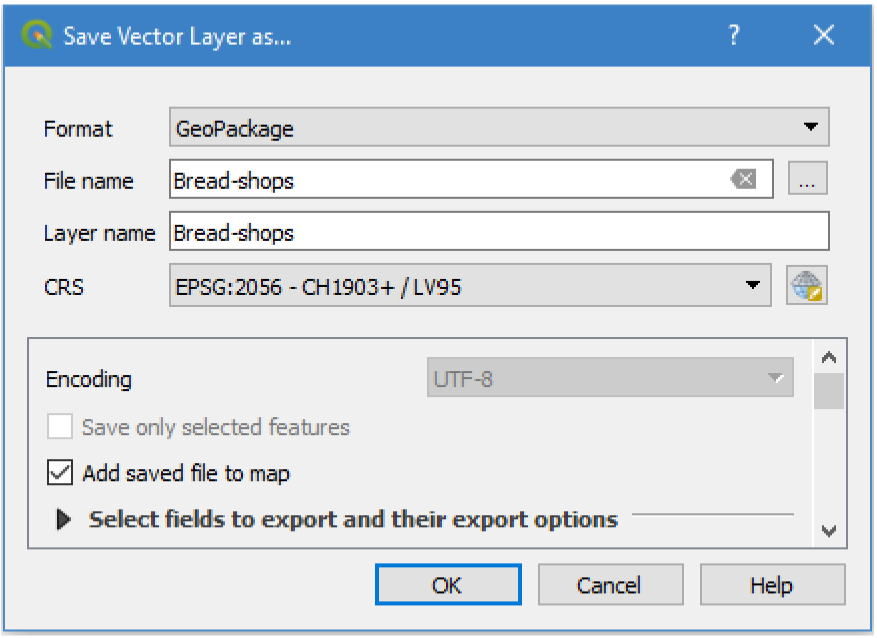


Figure 6. GeoJSON imported into QGIS

As you have probably noticed, an OSM standard layer is used here as a background to present the data more clearly. We assume that you don't have this one, but that's not too bad, because it's not absolutely necessary. So you can continue without OSM standard layer as background.

Finally, we want to change the coordinate system of the GeoJSON imported into QGIS and save it as a GeoPackage. To do this, right-click on the layer, go to "Export" and select the top entry "Save Feature As...". We change the KBS to "EPSG:2056 - CH1903+/LV95" and give the file a name. We leave the format as GeoPackage and click on "OK".

Below you can see the export window:



Save Vector Layer as...

Format: GeoPackage

File name: Bread-shops

Layer name: Bread-shops

CRS: EPSG:2056 - CH1903+ / LV95

Encoding: UTF-8

☐ Save only selected features

☑ Add saved file to map

Figure 7. Export as GeoPackage

Very good, you managed to import OpenStreetMap data into QGIS and save it as a new format. The other way to import OpenStreetMap data into QGIS is using the QuickOSM plugin, which is explained below. If you are not interested or have already read it, you can jump to the chapters "Merging Polygon Layers with Point Layers in QGIS" or "Filter Objects with QGIS or rule-based display".

Import OpenStreetMap data directly into QGIS with QuickOSM plugin

Now let’s look at another way to import and manage OpenStreetMap data into QGIS. It is best to delete the previously created layer and start at zero to avoid confusion.

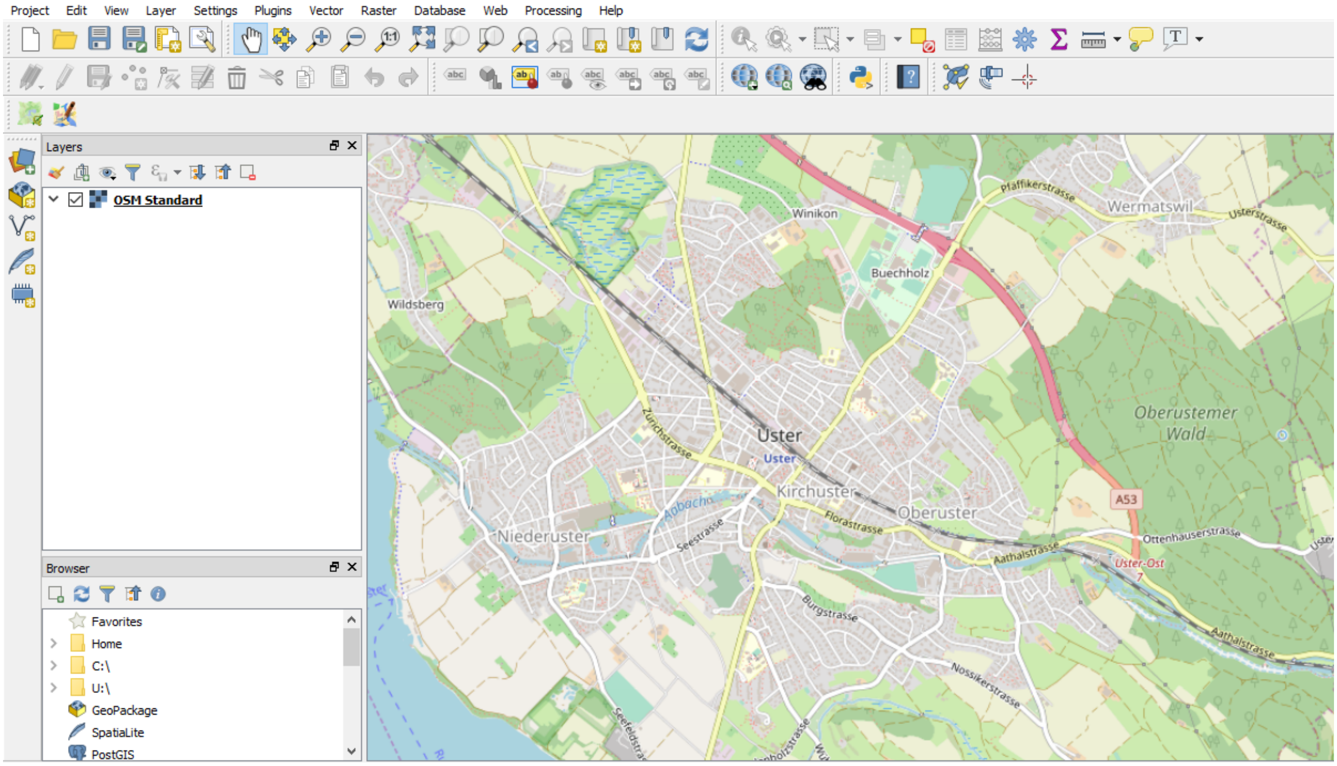
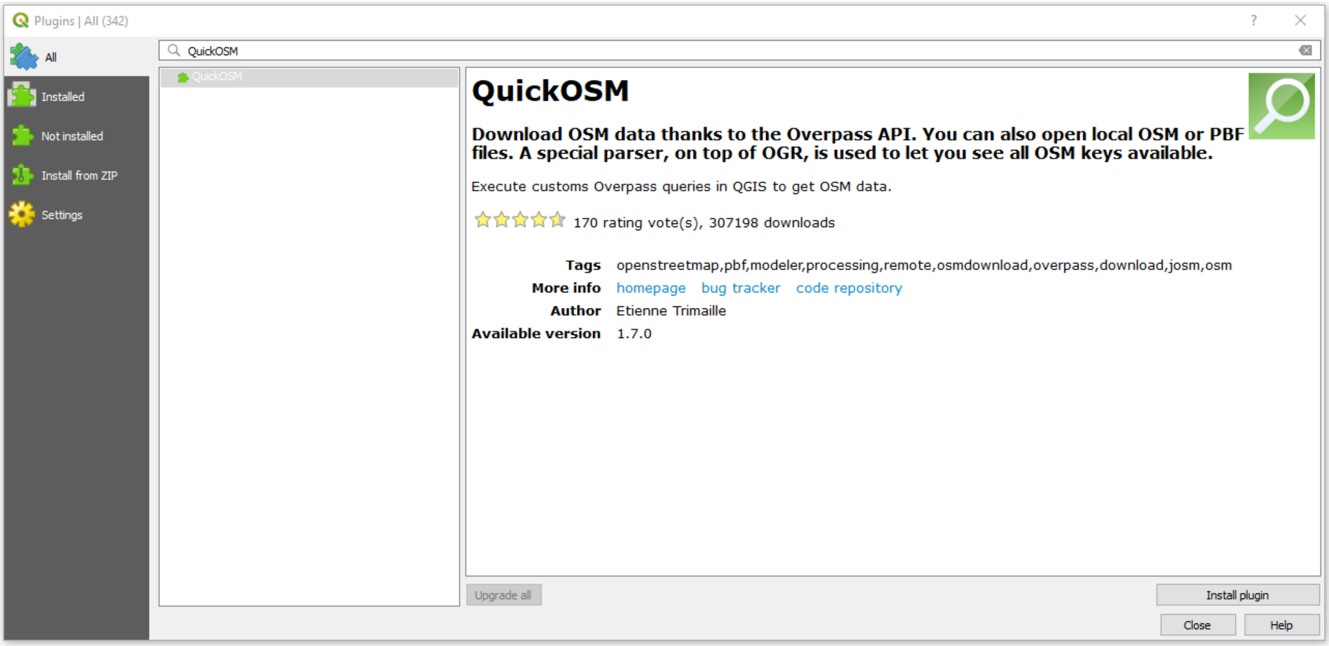


Figure 8. QGIS

First we install the plugin called QuickOSM, which can be used to create queries directly in QGIS. To do this we go to "Extensions" in the upper menu bar and then to "Manage and install extensions…". Enter QuickOSM in the search field and since there is only one plugin under this name, you should see the correct one. Now click on "Install extension" and you are ready to work with the plugin.

The following picture shows the installation of the plugin:



QuickOSM

Download OSM data thanks to the Overpass API. You can also open local OSM or PBF files. A special parser, on top of OGR, is used to let you see all OSM keys available.

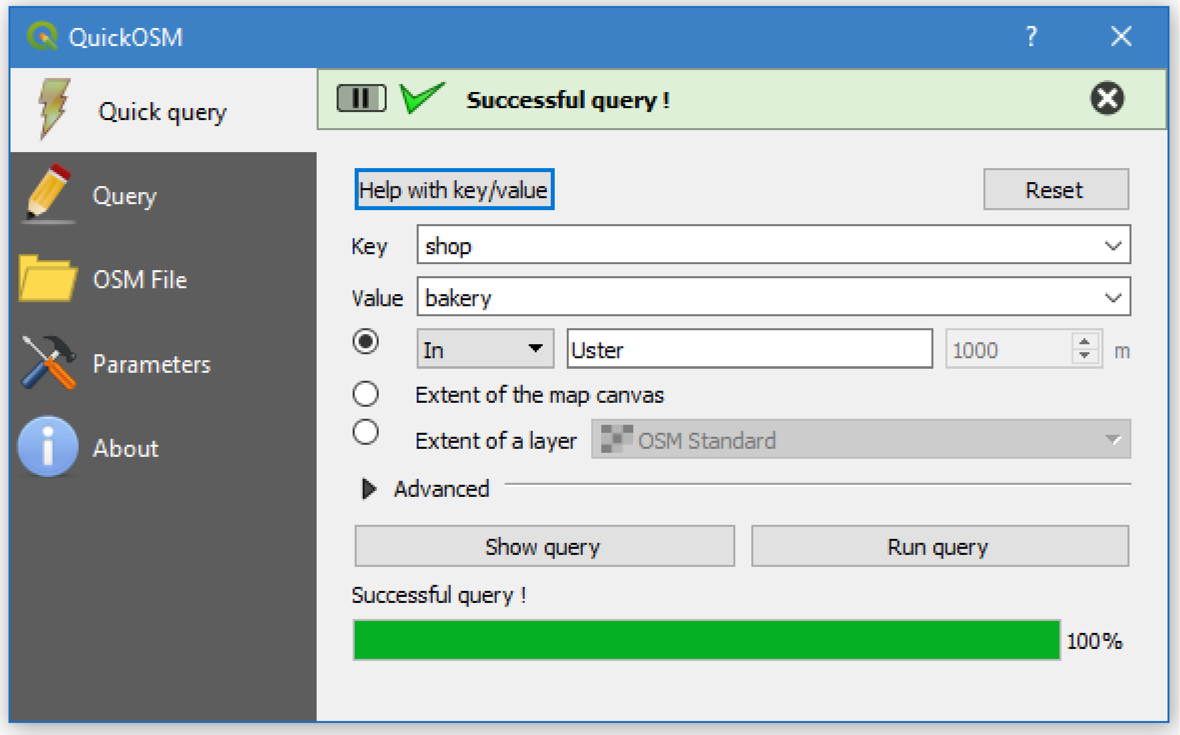
Execute customs Overpass queries in QGIS to get OSM data.

170 rating vote(s), 307198 downloads

Tags	openstreetmap,pbf,modeler,processing,remote,osmdownload,overpass,download,josm,osm
More info	homepage bug tracker code repository
Author	Etienne Trimaille
Available version	1.7.0

Figure 9. QuickOSM installation

Now we open the plugin by clicking on "Vector" in the menu bar and then on "Quick OSM". Then we enter the following query to search for bakeries in Uster:



Successful query !

Key: shop

Value: bakery

● In Uster 1000 m

○ Extent of the map canvas

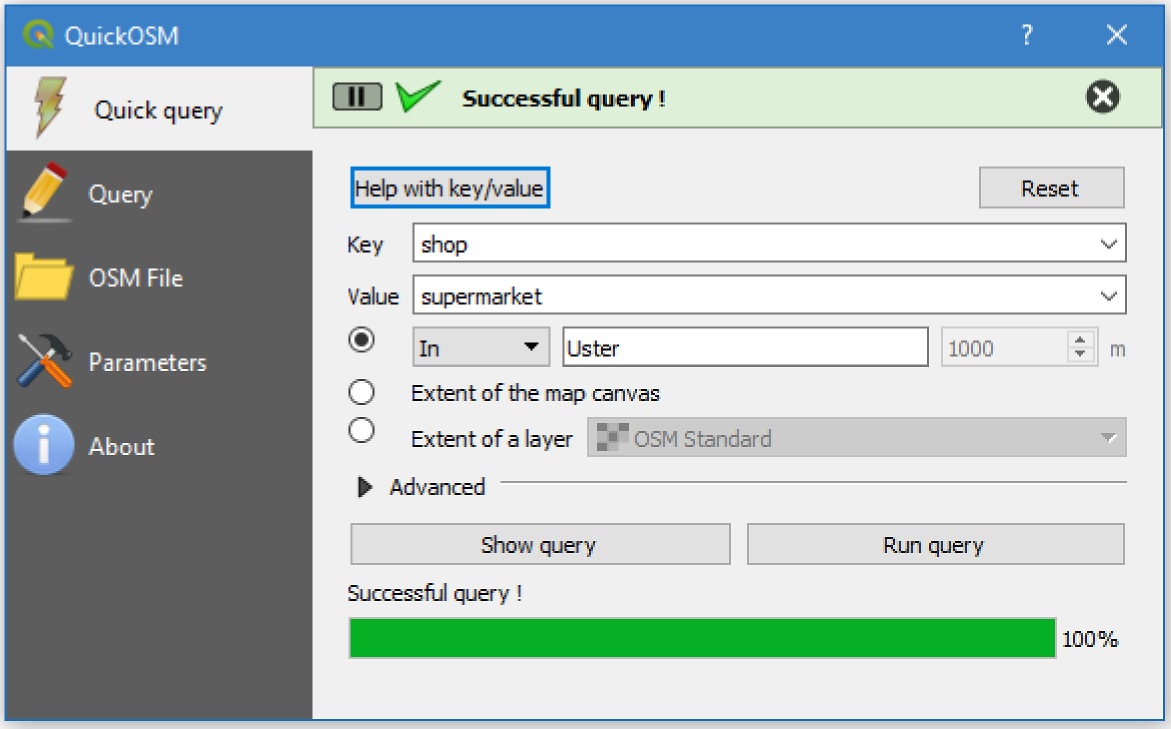
○ Extent of a layer OSM Standard

Successful query !

100%

Figure 10. Bakery query

After the query has been executed, we make a second query to search for supermarkets:



Successful query !

Key: shop

Value: supermarket

● In | Uster | 1000 m

○ Extent of the map canvas

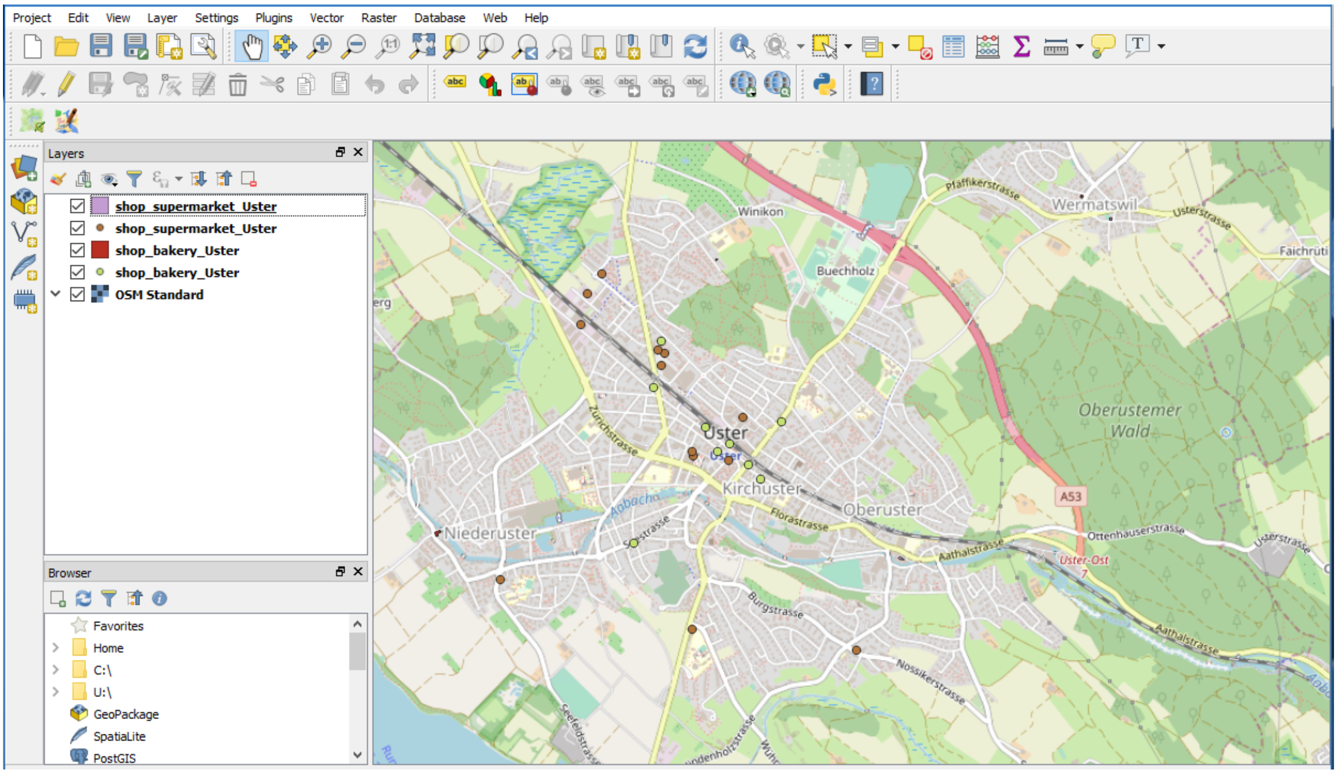
○ Extent of a layer: OSM Standard

Successful query !

100%

Figure 11. Supermarket query

Now you should have the following layers:



Layers

☑ shop_supermarket_Uster

☑ shop_supermarket_Uster

☑ shop_bakery_Uster

☑ shop_bakery_Uster

☑ OSM Standard

Figure 12. Results of the QuickOSM queries

If you want to know how to convert polygon layers into point layers or how to filter objects, have a look at the chapters "Merging polygon layers with point layers in QGIS" or "Filtering objects with QGIS or rule-based display".

Various web tools for downloading OpenStreetMap data

In this chapter we have compiled various web tools for downloading OpenStreetMap data. None of these tools are the same. Some require that you register and all are accessible free of charge and are provided without warranty.

OSM.org

On https://openstreetmap.org you can also download data. However, these can only be obtained as square sections and up to a maximum of 50,000 nodes and also only in `OSM XML` format.

If you want to know how to convert polygon layers into point layers or how to filter objects, see the chapters "Merging polygon layers with point layers in QGIS" or "Filtering objects with QGIS or rule-based display".

Geofabrik downloads

The website of Geofabrik http://download.geofabrik.de/ offers OpenStreetMap data for whole countries and regions, which are of course freely usable. The data can be stored there as Shapefiles (`.shp.zip`) or in the OSM-specific formats `.osm.pbf` and `.osm.bz2`.

The data are divided into individual regions. If you click on a region, the data of the countries in this region will be displayed. Depending on the country, you can also download smaller excerpts from the country. If you hold the mouse pointer over a region or a country, you will see the respective area in the map on the right side of the website. The data on GeoFabrik is updated every 24 hours with the data from OpenStreetMap. This means that if you have just made your own changes in OpenstreetMap, you have to wait at least a day until they are available at Geofabrik. Make sure what data you are downloading, as some can be several gigabytes in size.

The Shapefile is a GIS vector format and thus structured according to certain layers ('feature classes'). To find certain objects, such as bakeries (in OSM Tag shop=bakery) or supermarkets (Tag shop=supermarket) you need to consult the documentation of the Layered GIS Format at Geofabrik - or the data itself.

If you want to know how to convert polygon layers into point layers or how to filter objects, see the chapters "Merging polygon layers with point layers in QGIS" or "Filtering objects with QGIS or rule-based display".

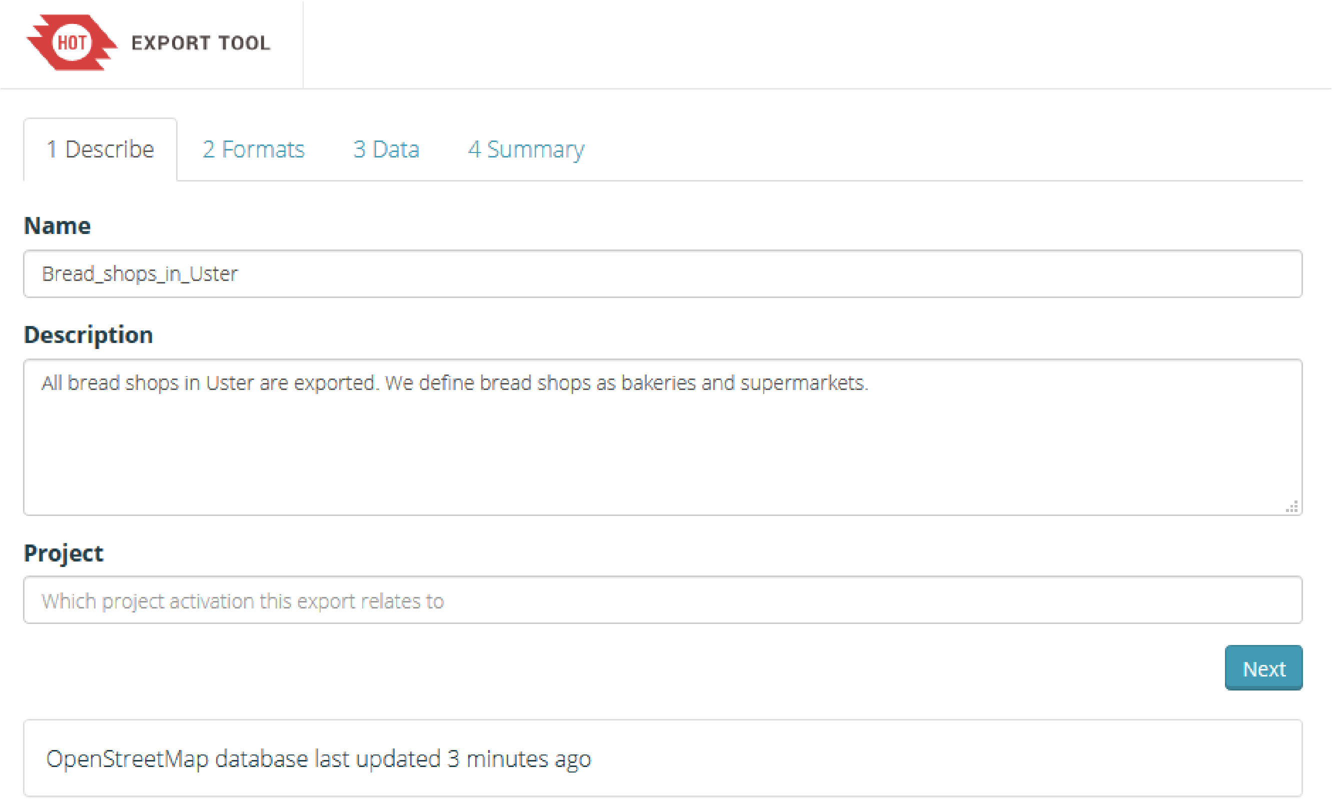
HOT Export Tool

The Humanitarian OpenStreetMap Team (HOT) offers an online service with the HOT Export Tool to download data from OpenStreetMap. To use the HOT ExportTool you need an OpenStreetMap user account and a valid email address to be notified when the data is ready for download. After logging in you will see the following points and tabs

1. Describe
2. Formats
3. Data
4. Summary

Tabulators 1 and 4 are described in more detail below.

Tab 1 "Export description" is about giving the file to be exported a name and a description. The picture below shows an example of exporting bread shops in Uster.



Name

Bread_shops_in_Uster

Description

All bread shops in Uster are exported. We define bread shops as bakeries and supermarkets.

Project

Which project activation this export relates to

OpenStreetMap database last updated 3 minutes ago

Figure 13. HOT Export Tool Tab 1 "Describe".

Then you can select the data you want to download in different ways.

You can either:

- Enter a locality in the search
- Specify a Bounding Box with Coordinates
- Download the current view
- Draw a Bounding Box yourself
- Draw a polygon
- Upload a polygon as GeoJSON file

For the first two possibilities, only the search field has to be used. There you can specify either the desired town or a bounding box (west, south, east, north).

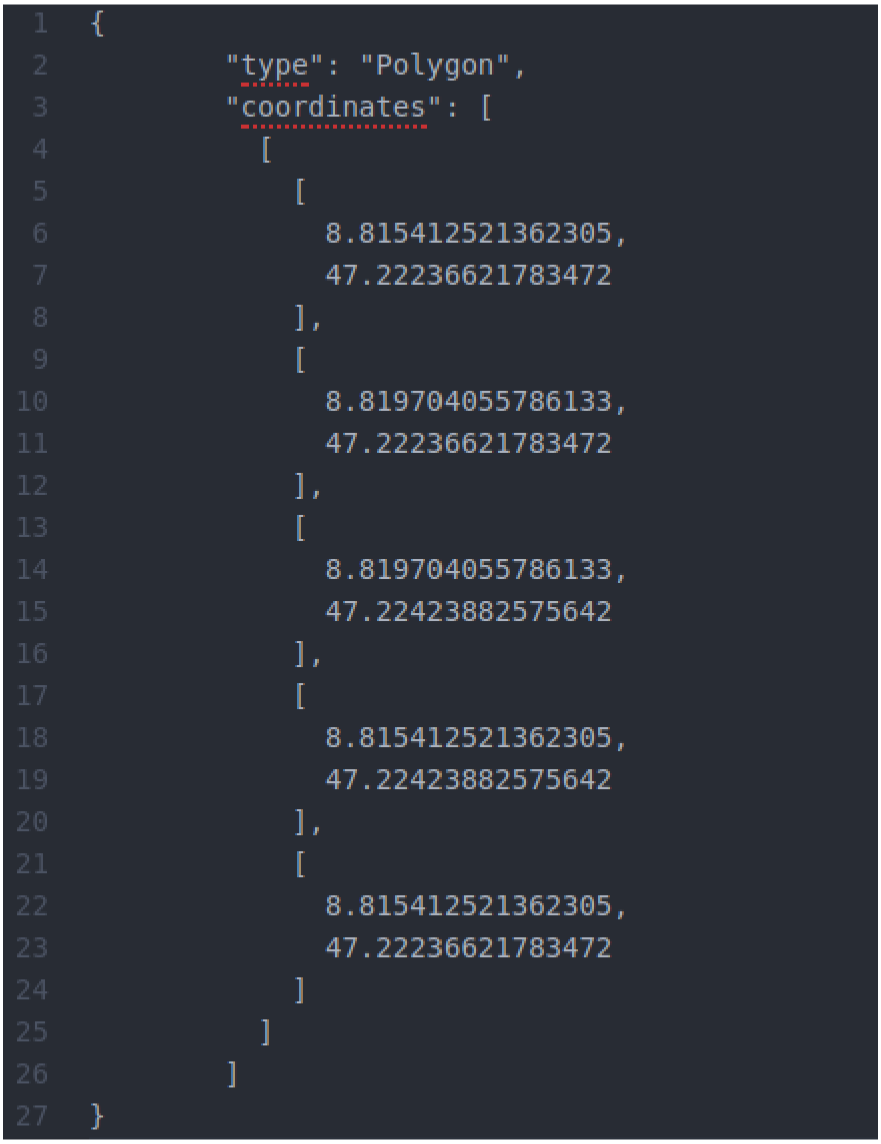
To draw a bounding box or polygon, press the *Draw* button on the right side of the map. To download the current view, press the *This View* button. If you want to upload a GeoJSON file, you can do so with the *Import* button.

At this point a little excursion how to create a GeoJSON file.

The website http://geojson.io is a well-known website that allows you to edit GeoJSON.

The HOT Export Tool expects only simple GeoJSON objects (no "FeatureCollection"), i.e. a polygon type and a list of coordinates.

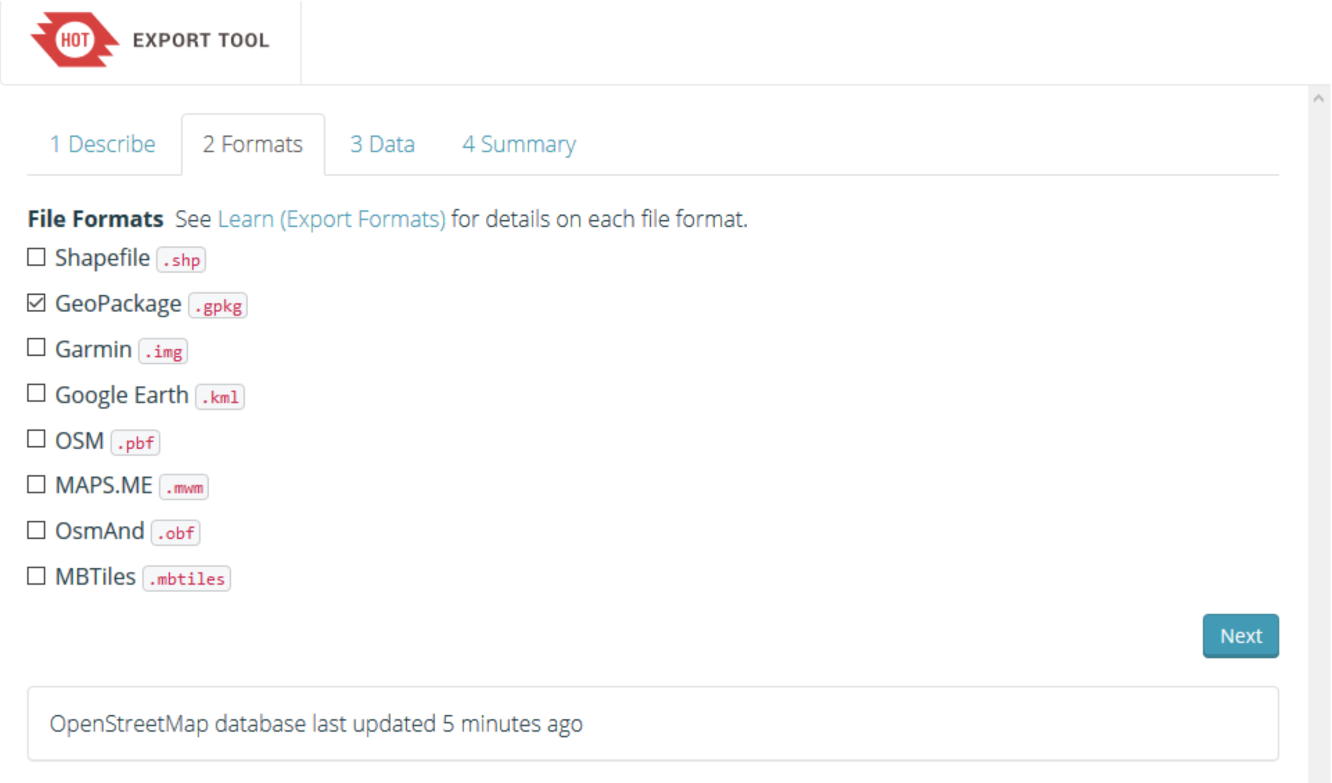
GeoJSON files from geojson.io and other sources must therefore be adapted manually. Therefore open the GeoJSON file in a text editor and simplify the structure similar to the following picture.



```
{
        "type": "Polygon",
        "coordinates": [
          [
            [
              8.815412521362305,
              47.22236621783472
            ],
            [
              8.819704055786133,
              47.22236621783472
            ],
            [
              8.819704055786133,
              47.22423882575642
            ],
            [
              8.815412521362305,
              47.22423882575642
            ],
            [
              8.815412521362305,
              47.22236621783472
            ]
          ]
        ]
}
```

Figure 14. Single GeoJSON object for the HOT Export Tool

Let’s continue with the HOT Export Tool again. In tab 2 you choose the format, e.g. GeoPackage.



EXPORT TOOL

1 Describe 2 Formats 3 Data 4 Summary

File Formats See Learn (Export Formats) for details on each file format.

☐ Shapefile `.shp`
☑ GeoPackage `.gpkg`
☐ Garmin `.img`
☐ Google Earth `.kml`
☐ OSM `.pbf`
☐ MAPS.ME `.mwm`
☐ OsmAnd `.obf`
☐ MBTiles `.mbtiles`

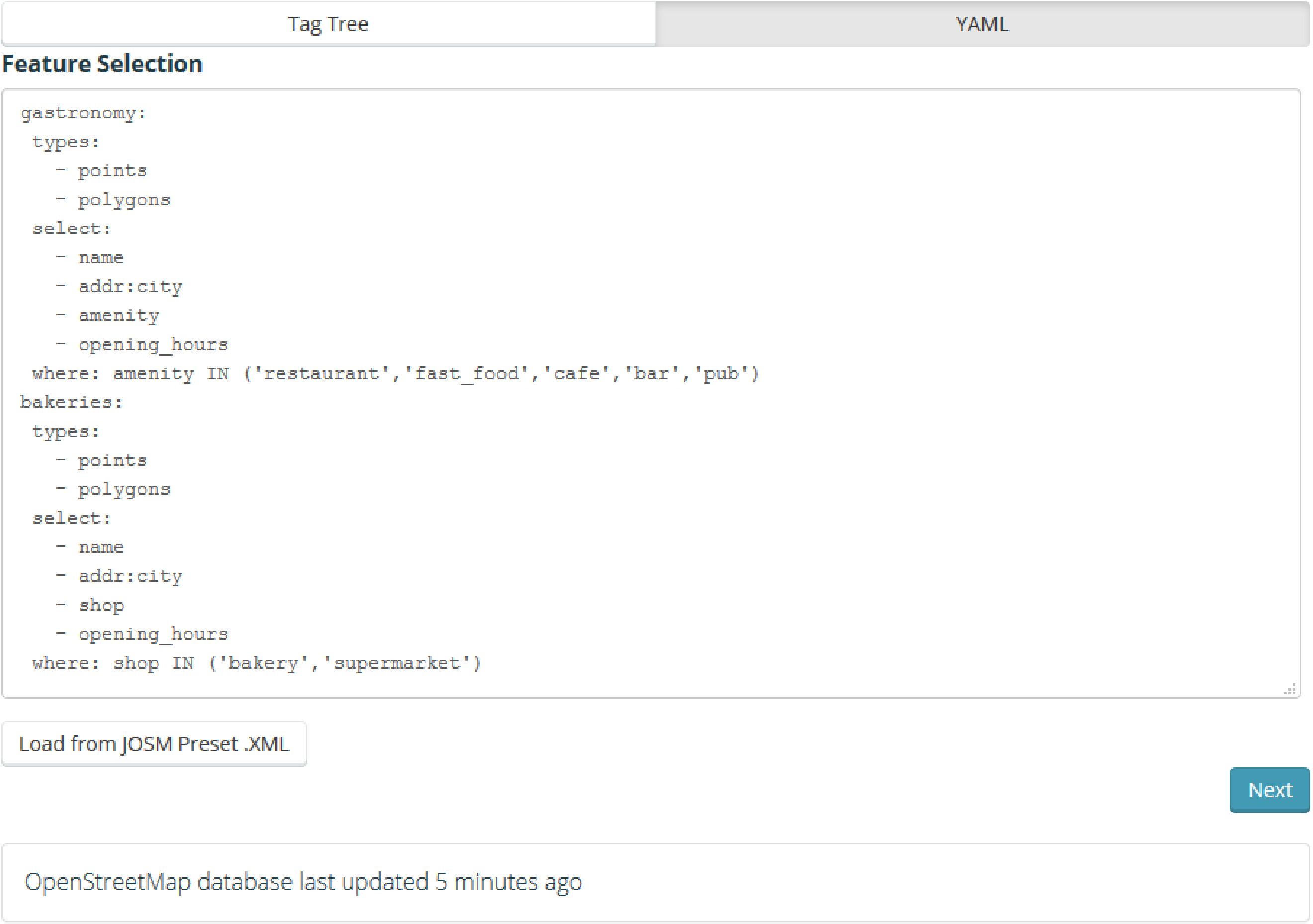
OpenStreetMap database last updated 5 minutes ago

Figure 15. HOT Export Tool Tab 2 "Formats".

With tab 3 "Data" you change to the register "YAML" and insert the following:

```
gastronomy:
 types:
   - points
   - polygons
 select:
   - name
   - addr:city
   - amenity
   - opening_hours
 where: amenity IN ('restaurant','fast_food','cafe','bar','pub')
bakeries:
 types:
   - points
   - polygons
 select:
   - name
   - addr:city
   - shop
   - opening_hours
 where: shop IN ('bakery','supermarket')
```

The whole thing will look like this:

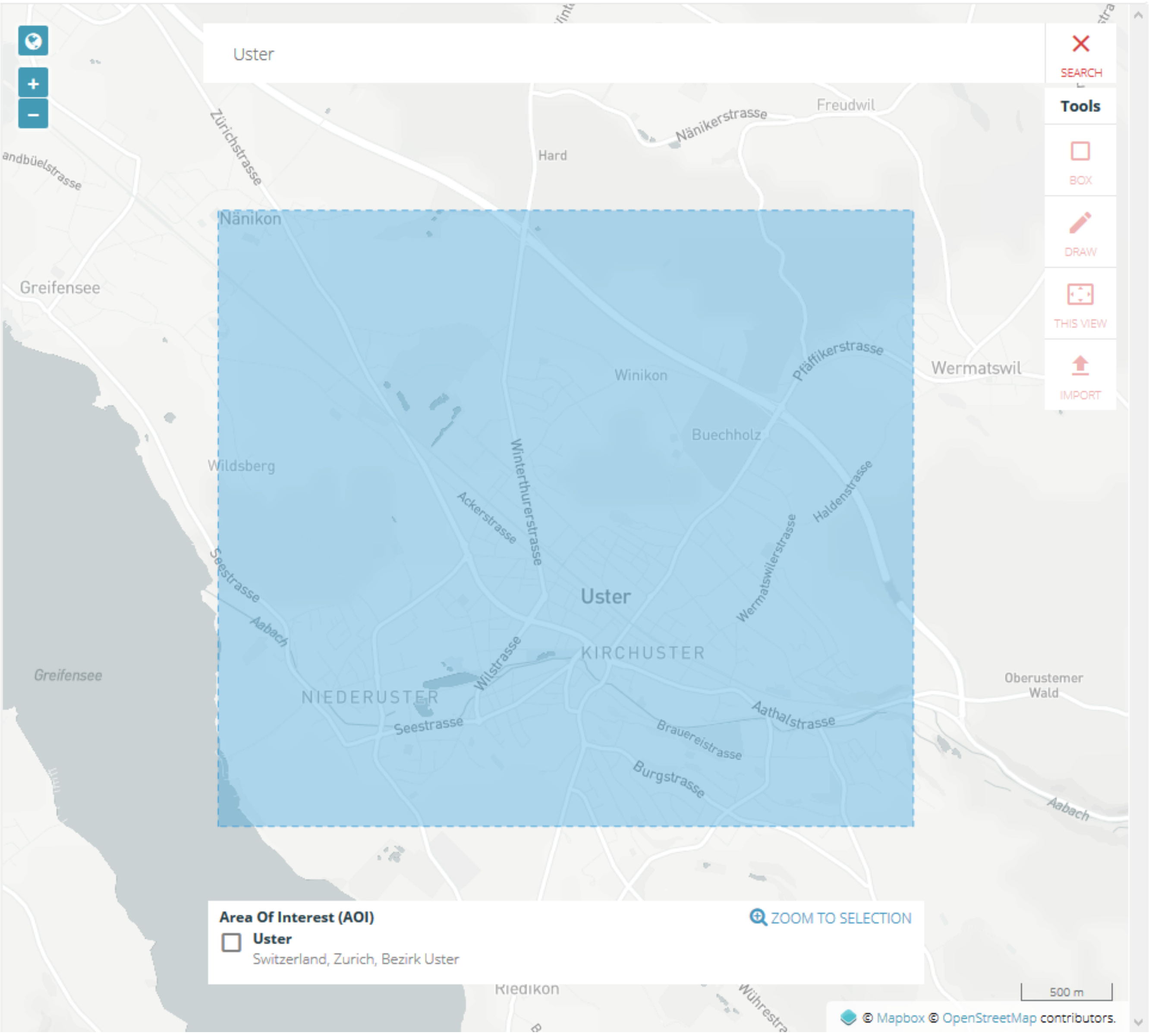

Tag Tree
YAML
Feature Selection

```
gastronomy:
 types:
   - points
   - polygons
 select:
   - name
   - addr:city
   - amenity
   - opening_hours
 where: amenity IN ('restaurant','fast_food','cafe','bar','pub')
bakeries:
 types:
   - points
   - polygons
 select:
   - name
   - addr:city
   - shop
   - opening_hours
 where: shop IN ('bakery','supermarket')
```

Load from JOSM Preset .XML
Next
OpenStreetMap database last updated 5 minutes ago

Figure 16. HOT Export Tool Tab 3 "Data".

After you've done that, we can almost export the query as a GeoPackage. That's why we go on to Tab 4 "Summary". The last thing we have to do before exporting is to search for "Uster" on the right side of the map. Then it will be zoomed in:



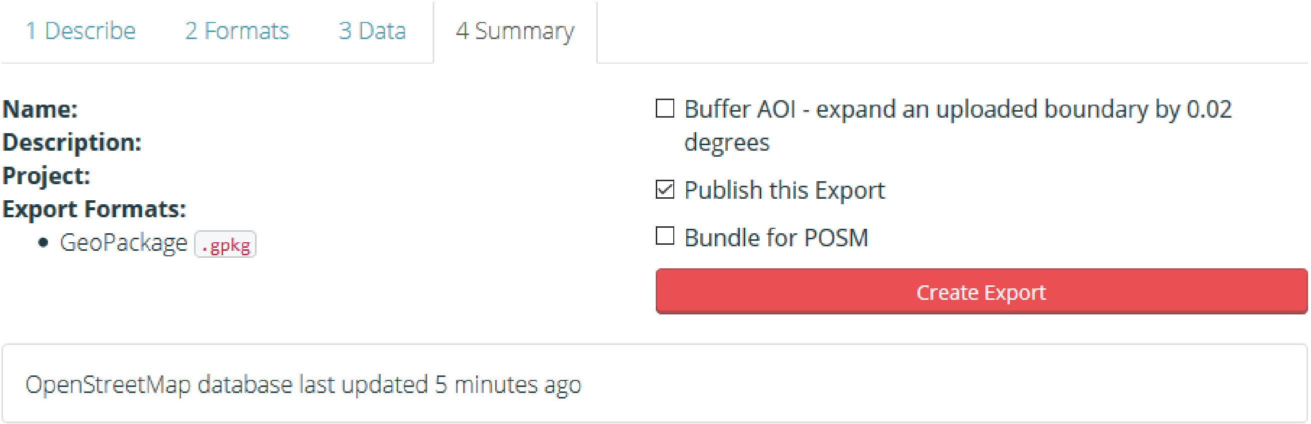
Area Of Interest (AOI)

□ **Uster**

Switzerland, Zurich, Bezirk Uster

Figure 17. HOT Export Tool Uster

Once you have done this, you can click on the "Create Export" button (see below).



1 Describe 2 Formats 3 Data 4 Summary

Name:
Description:
Project:
Export Formats:

- GeoPackage .gpkg

□ Buffer AOI - expand an uploaded boundary by 0.02 degrees

☑ Publish this Export

□ Bundle for POSM

OpenStreetMap database last updated 5 minutes ago

Figure 18. HOT Export Tool Tab 4 "Summary".

If you want to know how to convert polygon layers into point layers or how to filter objects, see the chapters "Merging polygon layers with point layers in QGIS" or "Filtering objects with QGIS or rule-based display".

OSMaxx

With OSMaxx, excerpts from OpenStreetMap can be downloaded in various GIS formats, not just by country, but in any desired format.

To use OSMaxx you need an OSM account with a valid email address.

Existing excerpts can be downloaded under *Existing Excerpt / Country*. You can choose in which format the file should be downloaded, which coordinate system should be used and how many details should be displayed.

Under *New Excerpt* you can select your own areas of OpenStreetMap that you want to download. You can draw a rectangle or a polygon over the map. Again you can choose in which format the file should be downloaded, which coordinate system should be used and how many details should be displayed.

Exporting the data takes at least about 45 minutes.

If you have chosen a GIS format and want to know if and where it has bakeries, for example, go to the "About" page of OSMaxx, where the data scheme is documented.

If you want to know how to convert polygon layers into point layers or how to filter objects, see the chapters "Merging polygon layers with point layers in QGIS" or "Filtering objects with QGIS or rule-based display".

BBBike

BBBike https://extract.bbbike.org/ offers the data of Planet.osm in many different file formats.

Only a bounding box has to be created for the extraction. The bounding box can have a maximum size of 24'000'000 km², or 512mb as a file. After you have entered a valid email address and assigned a name to the extraction you can click on the *Extract* button. A notification will be sent to you by e-mail with the download link. The documentation of the schema behind the GIS formats like GeoPackage or Shapefile can be found here:

- http://www.geopackage.org/spec/
- https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

If you want to know how to convert polygon layers into point layers or how to filter objects, see the chapters "Merging polygon layers with point layers in QGIS" or "Filtering objects with QGIS or rule-based display".

Merge Polygon Layers with Point Layers in QGIS

With OSM, it happens again and again that an object - such as a bakery - is sometimes captured as a point and sometimes as a building outline polygon. This is why it often happens that you get polygon layers in GIS with the same object types as in point layers. Actually, however, one wants to address all objects uniformly and all at once.

So given a polygon layer and a point layer, we have to convert the polygons in the polygon layer to points and can then integrate them in the point layer.

To do this, go to "Geometry Tools" in the "Vector" menu of QGIS and then to "Centroids...". After that the QGIS should look like this, where the names of the new layers have already been renamed:

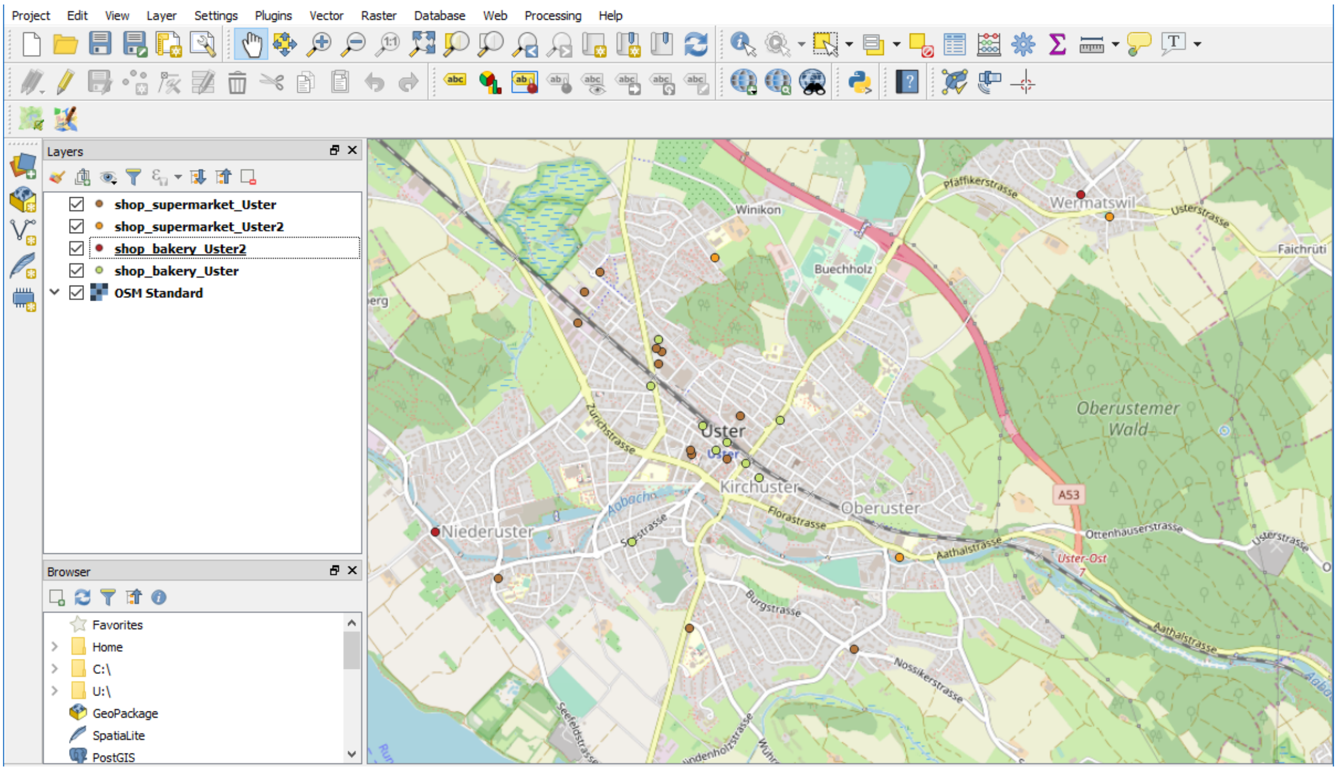
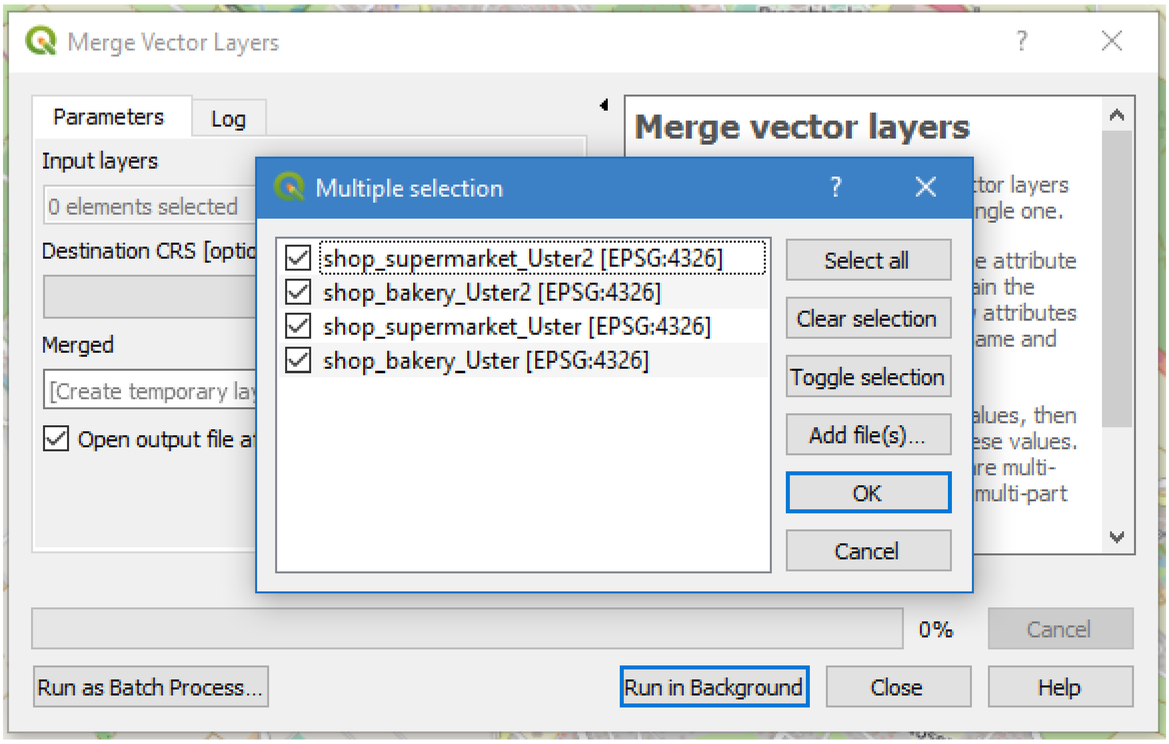


Figure 19. Polygon layer converted

Since we now have several point layers, we have to merge them. Such a merge only works if all layers have the same geometry data type. Go to "Data Management Tools" in the QGIS "Vector" menu and then to "Merge Vector Layers". Select the four layers in the dialog window:



Merge Vector Layers

Input layers: 0 elements selected

Destination CRS [optio

Merged: [Create temporary la

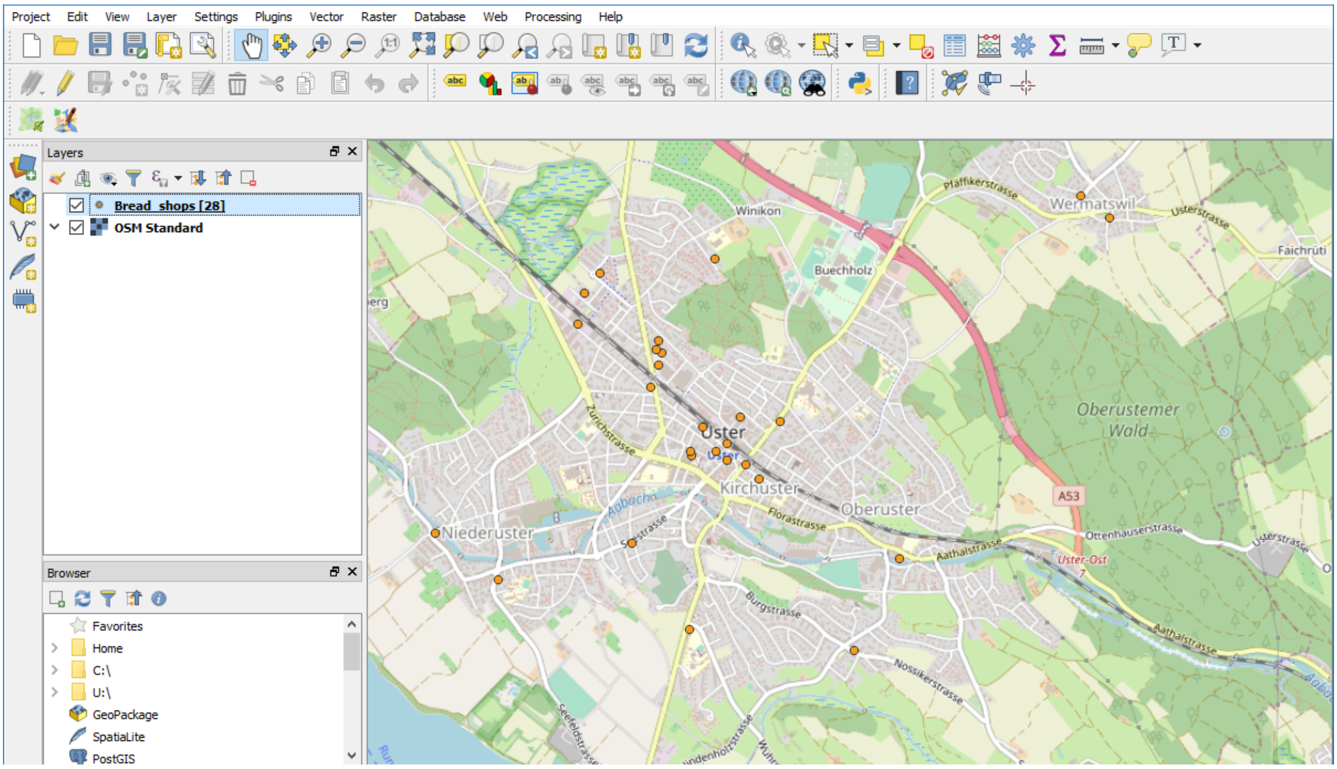
☑ Open output file a

Multiple selection

☑ shop_supermarket_Uster2 [EPSG:4326]
☑ shop_bakery_Uster2 [EPSG:4326]
☑ shop_supermarket_Uster [EPSG:4326]
☑ shop_bakery_Uster [EPSG:4326]

Figure 20. Merge multiple layers into a single layer

After you have done this, you will see a new layer on the left. Now you don’t need the old layers anymore and you can delete them. The new layer has the name "Merged". It would make sense if you renamed it "Bread shops." Below you can see the final result of this second method, where the layer has not (yet) been renamed:



Layers
☑ Bread_shops [28]
☑ OSM Standard

Figure 21. Final result

Filter or rule-based display of objects with QGIS

After you have received OpenStreetMap data in the desired section, you may not want to work with all data. There are two fundamentally different methods of obtaining a subset of the data: On the one hand, the data source can be filtered according to self-defined criteria. On the other hand, you can display the data based on rules, so that only those data are displayed that comply with the rules.

In the following we look at the two methods. The example of bread shops remains the same. The bread shops contain both "bakeries" and "supermarkets". But now we only want to process and display "bakeries" in QGIS.

Filter objects with QGIS

In QGIS there are two dialog options for filtering objects: Either you go to "Layer" in the menu and then select "Filter..." (while the desired layer is selected). Or you can right-click on the desired layer and go to "Filter...". Then a dialog with the title "Query creation" opens. In this window you can see the following:

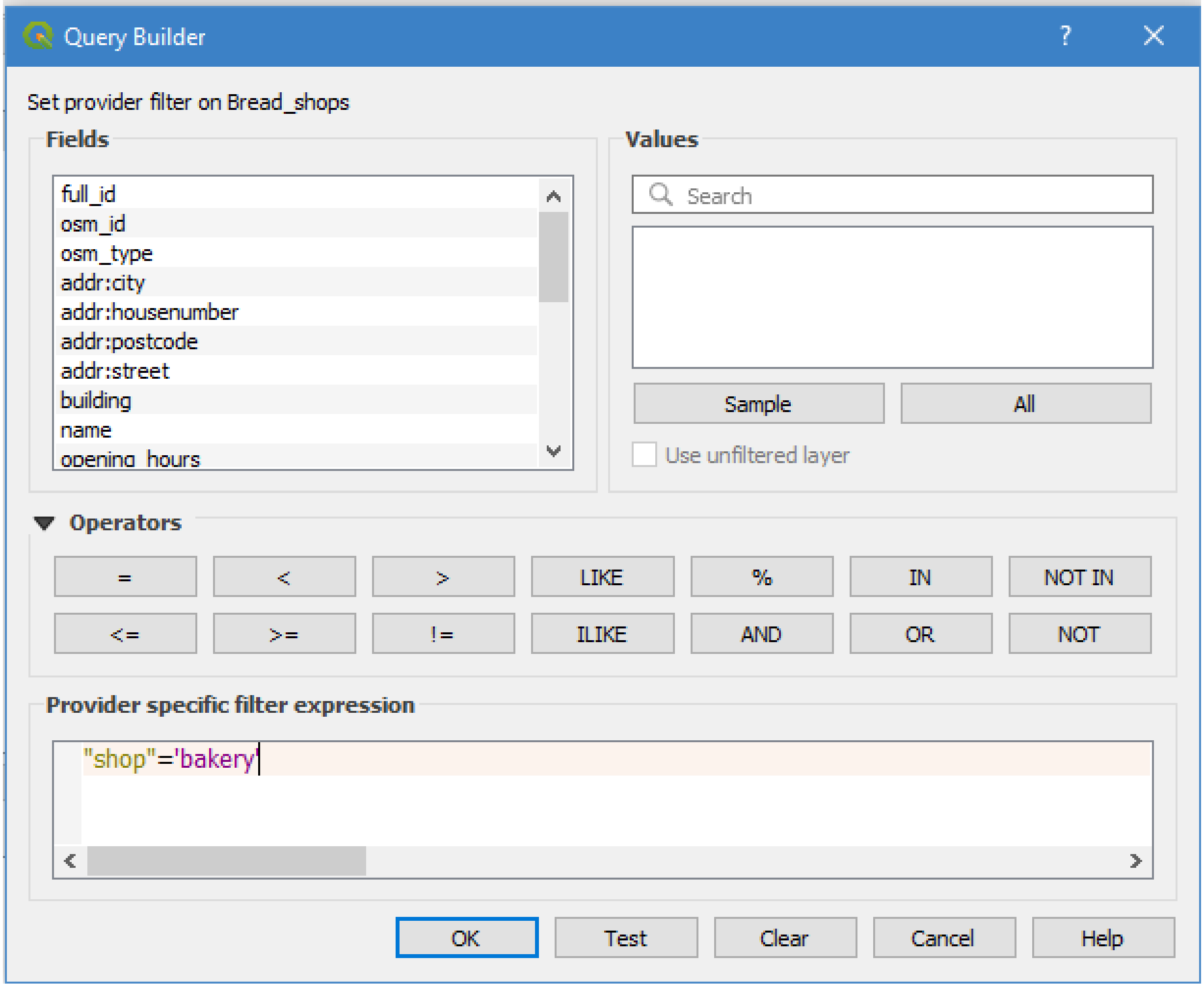
- Fields: A list of all attributes that the selected layer contains.
- Values: The values on the selected attribute that occur in the layer (to display some values, click on the "Sample" button)
- Operators: Logical operators for linking fields and values.
- Data provider-specific filter expression: SQL expression for filtering attributes and values

In the input field "Data provider-specific filter expression" we enter the following:

```
"shop"='bakery'
```

"shop" is the attribute (note the double quotes). 'bakery' is the value (note the single quotation marks).

The window should look like this:



```
"shop"='bakery'
```

Figure 22. Filter objects

Now click on "OK". Now you should get the message that about 11 objects were found which have the value "bakery" at the attribute "shop". We only see those objects that meet the filter criterion. All other objects are no longer part of the layer’s data source:

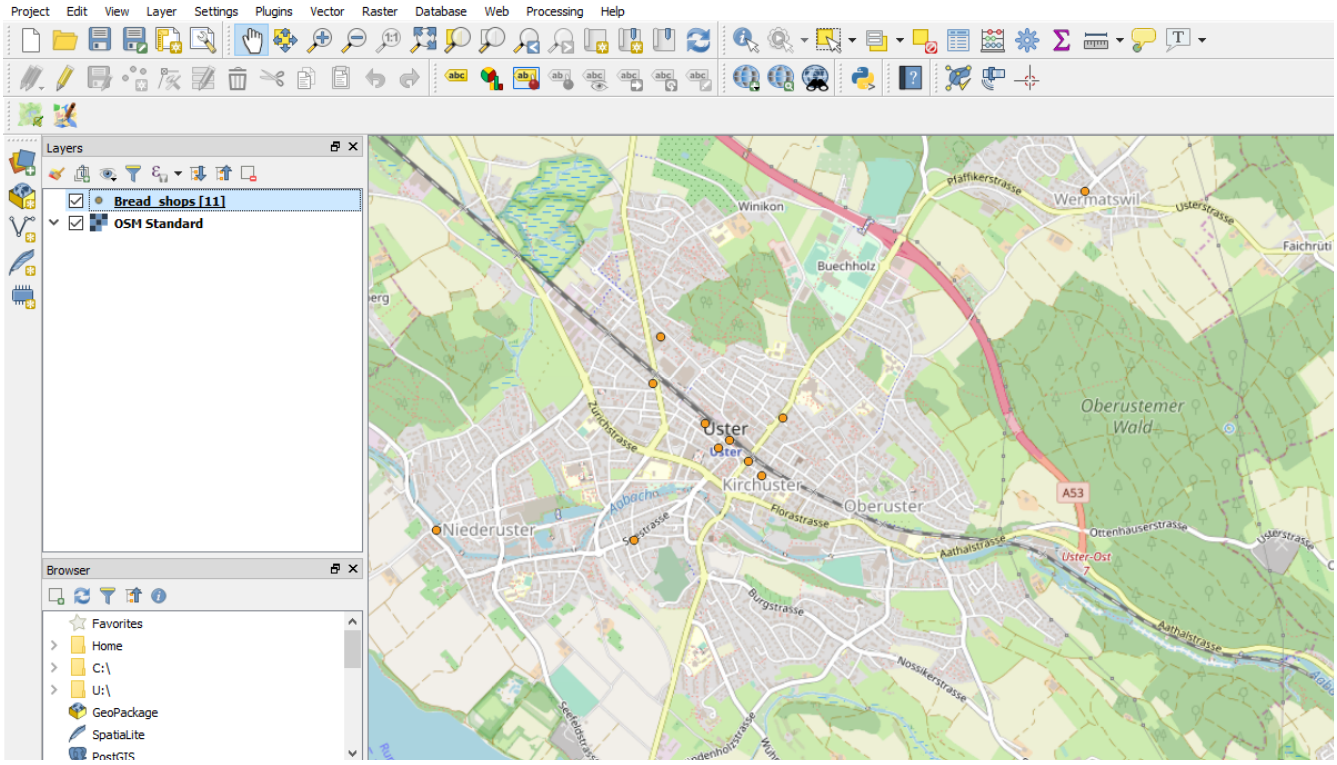


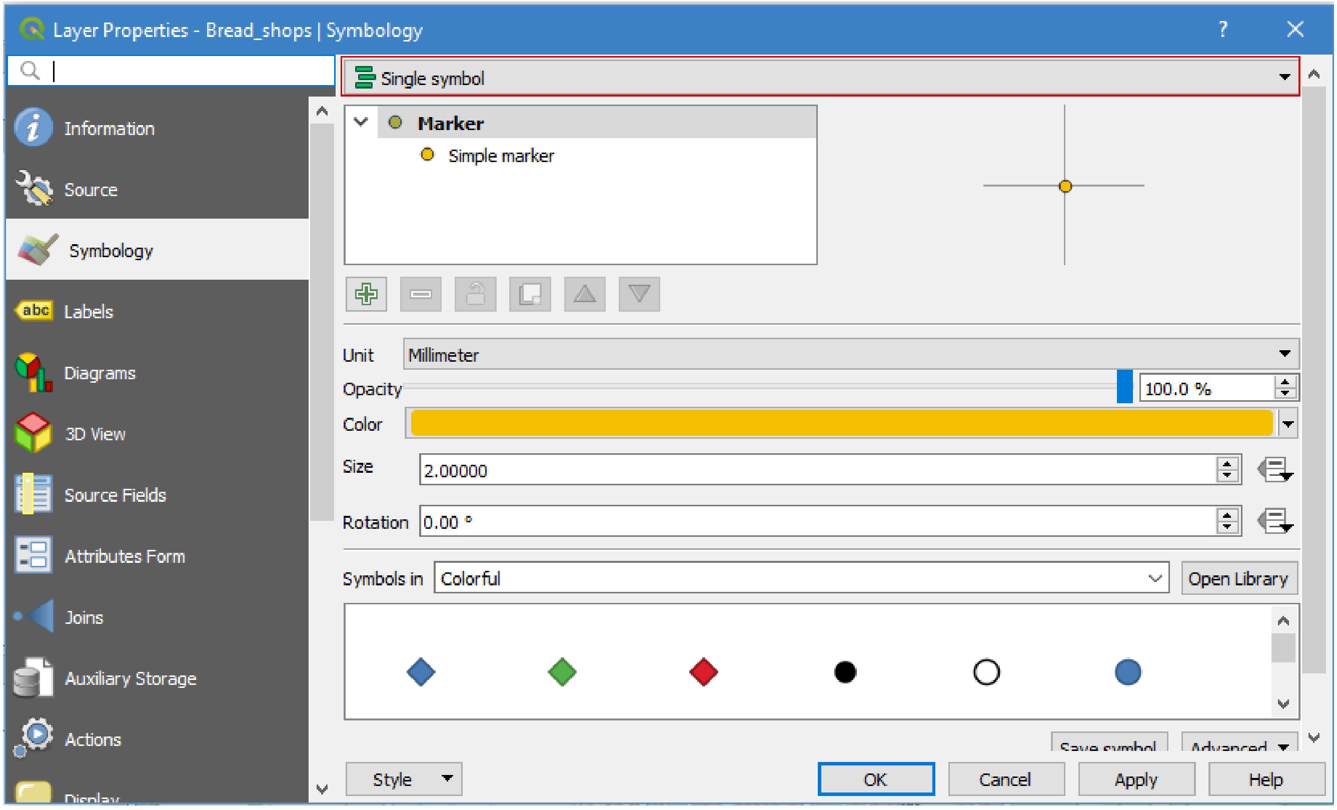
Figure 23. Bakeries in Uster

As you can see above, you should now rename the layer from "Bread shops" to "Bakeries".

Rule-based display of objects with QGIS

The rule-based representation is about displaying all objects that fulfill a criterion from a layer data source. One also speaks of "data-driven representation". This expression, which has to be filtered by, is part of the symbology and not part of the layer as with the other method.

The rule-based symbology can also be used to *not* display individual values. In our example we want to take advantage of this to show only bakeries ('bakery') and not supermarkets ('supermarket'). To do this we right-click on the layer and click on the bottom entry "Properties...". In the left list menu we click on "Symbology" (third entry). Now it should look like this for you:



Single symbol

Unit: Millimeter

Opacity: 100.0 %

Color

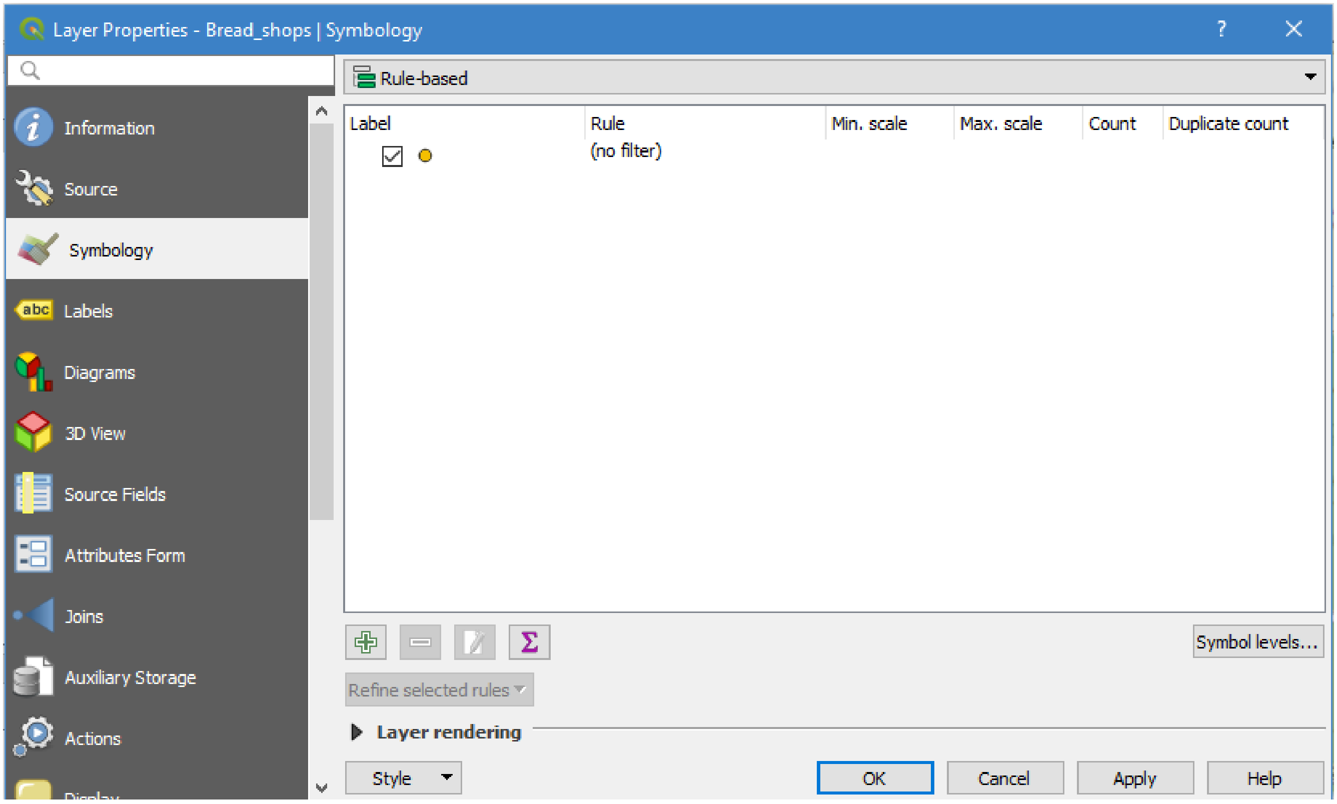
Size: 2.00000

Rotation: 0.00 °

Symbols in: Colorful

Figure 24. Representation of objects

In the upper picture you see a red mark around the entry "Single symbol". This is because we are now changing this entry to "Rule-based". Previously, all objects were treated the same (not differentiated) and thus displayed in the same way, therefore "single symbol". Now that we have selected "Rule-based", we can define rules (expressions) to assign objects to different symbols according to attribute values.



Rule-based

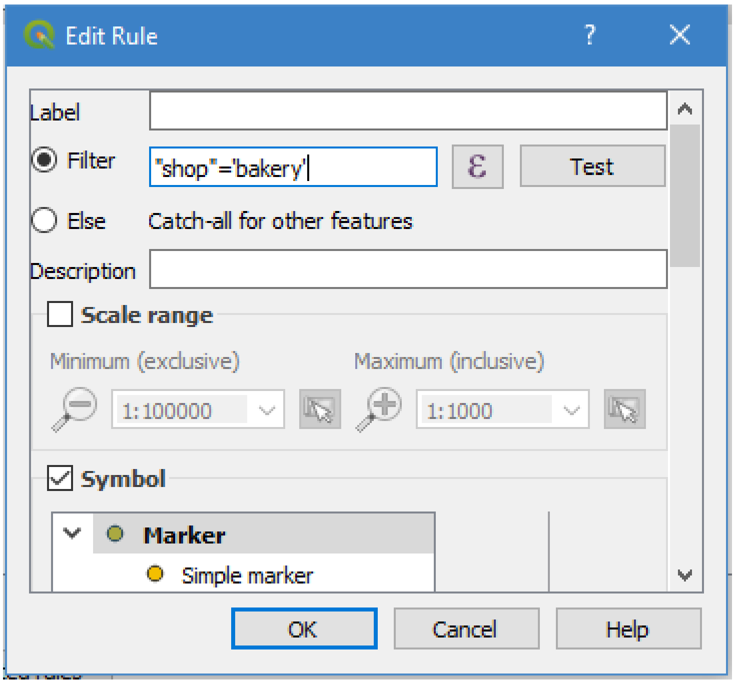
Label	Rule	Min. scale	Max. scale	Count	Duplicate count
☑ ●	(no filter)				

Figure 25. Objects without filter

In the picture above you can see that we have an entry. This entry contains all 28 objects, and we see that no rules are defined on these objects. Therefore we click twice on the entry. In the text field next to "Filter" we enter the following again:

```
"shop"='bakery'
```

It’ll look that way:



Label
● Filter "shop"='bakery'
○ Else Catch-all for other features
Description
□ Scale range
Minimum (exclusive) 1:100000
Maximum (inclusive) 1:1000
☑ Symbol
Marker
Simple marker

Figure 26. Edit Rule Dialog

Now we click on "OK". We see that the entry no longer says "(no filter)", but "shop"='bakery'. Now we click on "Apply" and then again on "OK". In QGIS this should now look like this:

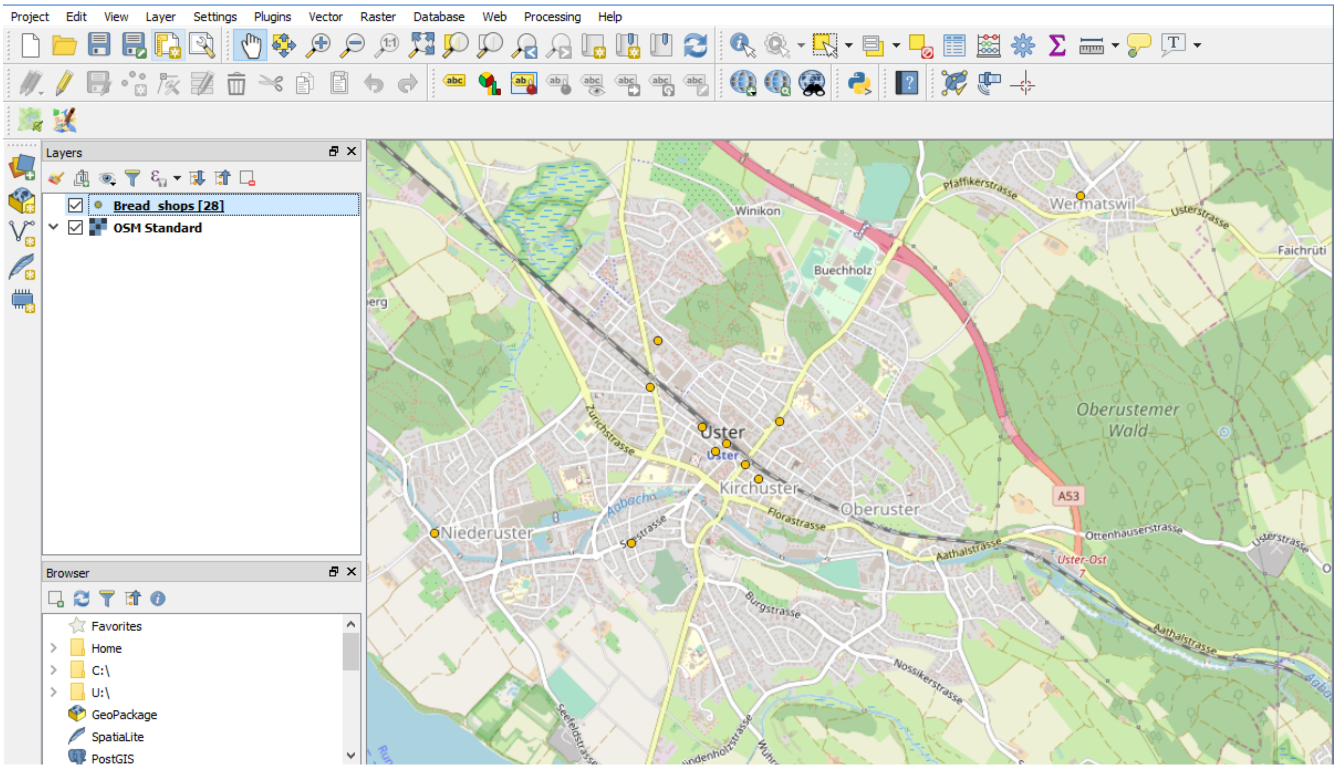


Figure 27. Bakeries in Uster

We see that we still have 28 objects. However, we are only shown the 11 objects that fulfill the rule, i.e. only bakeries. Maybe you have noticed it already, because in contrast to that the number of objects changes with the simple filter (see above). There not only the 11 bakeries are shown, but the number of objects is also reduced to 11. This is a particular difference between "normal" filtering and rule-based representation. Let’s look at another possibility.

Assuming that we still want to display all objects and only part of them, we can simply use a different color for all bakeries. Delete the rule we applied to the objects earlier. Then we start from the following view:

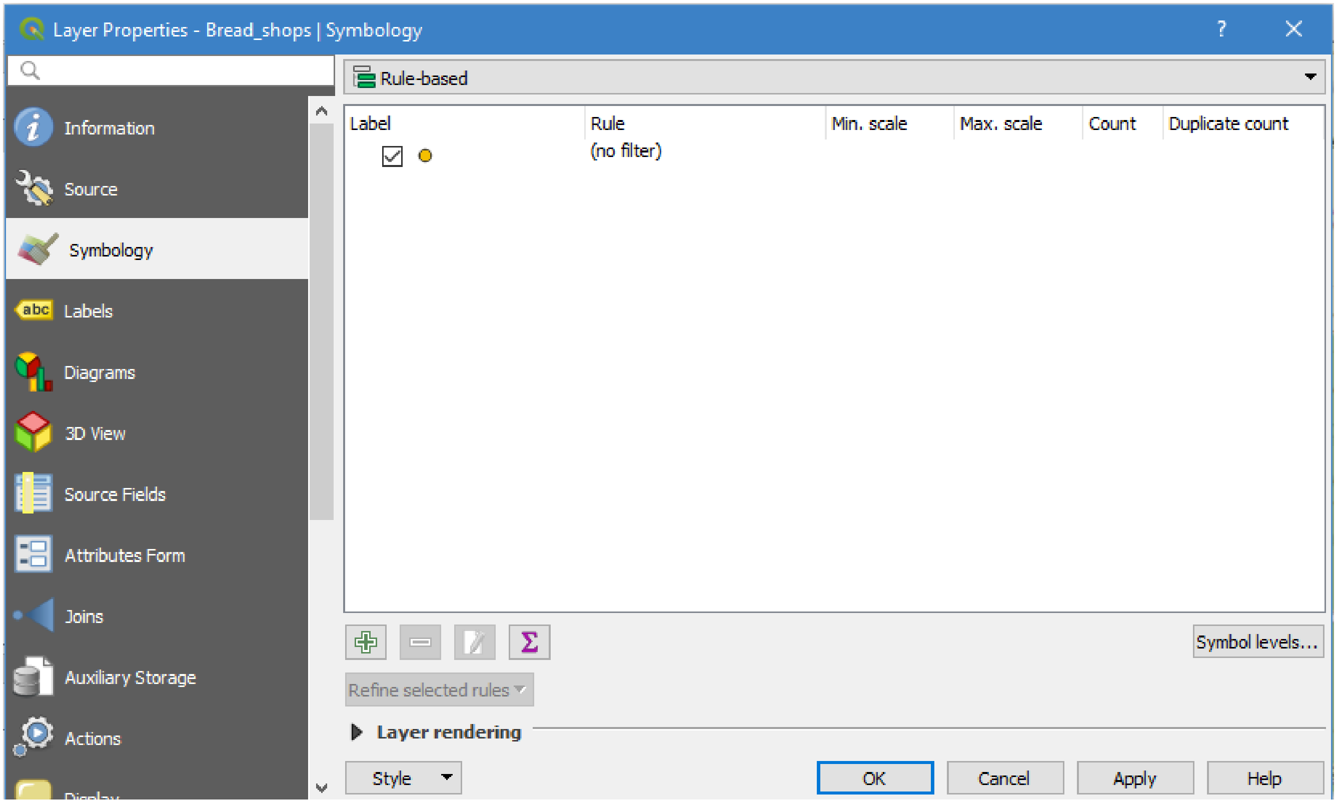
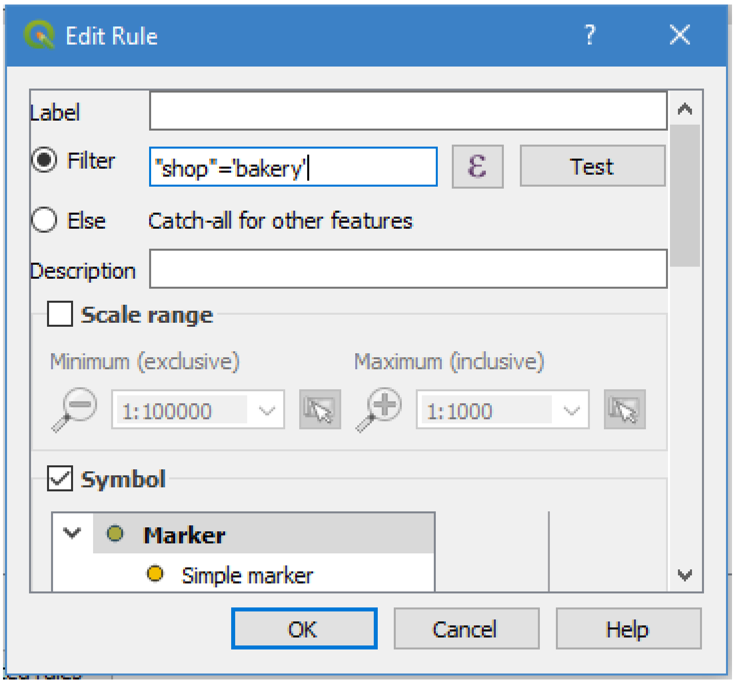


Figure 28. Objects without filter

At the bottom left we click on the plus sign. Now we have copied our objects and apply the rule to this copy. So enter the following again and click on "OK":



● Filter "shop"='bakery'

Figure 29. Edit Rule Dialog

Now we click on "Apply" and then on "OK". Now it should look like this in QGIS:

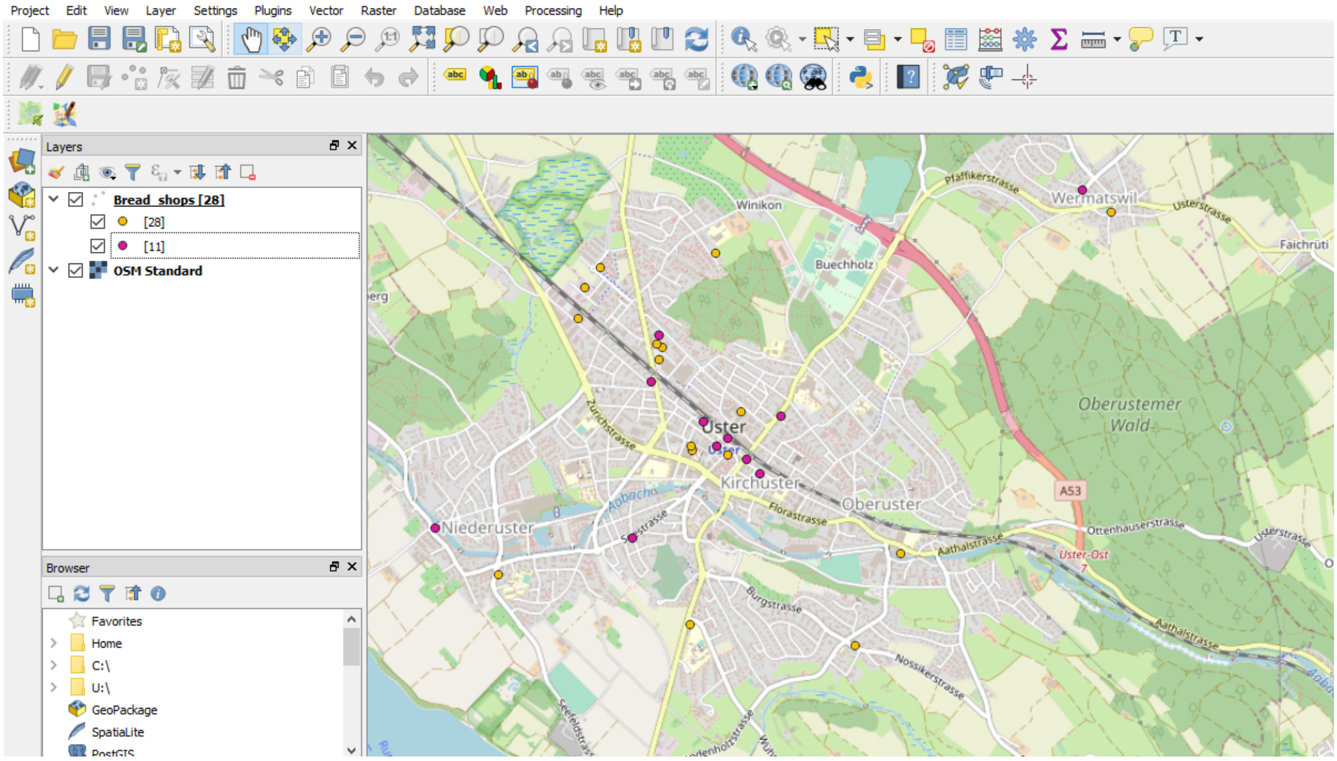


Figure 30. Bread shops with colour-coded bakeries (pink)

The "Bread shops" layer should now have two different object groups. On the one hand, we see all 28 bread shops in orange. On the other hand, we now have a new (sub-)group with 11 objects, namely our bakeries in pink.

This concludes this chapter or worksheet and leaves it to you to extend the layers and the whole QGIS project and improve the display.