

Customizing Search for Special-Interest Maps

Sarah Hoffmann

lonvia@denofr.de

28th July 2018

State of the Map 2019

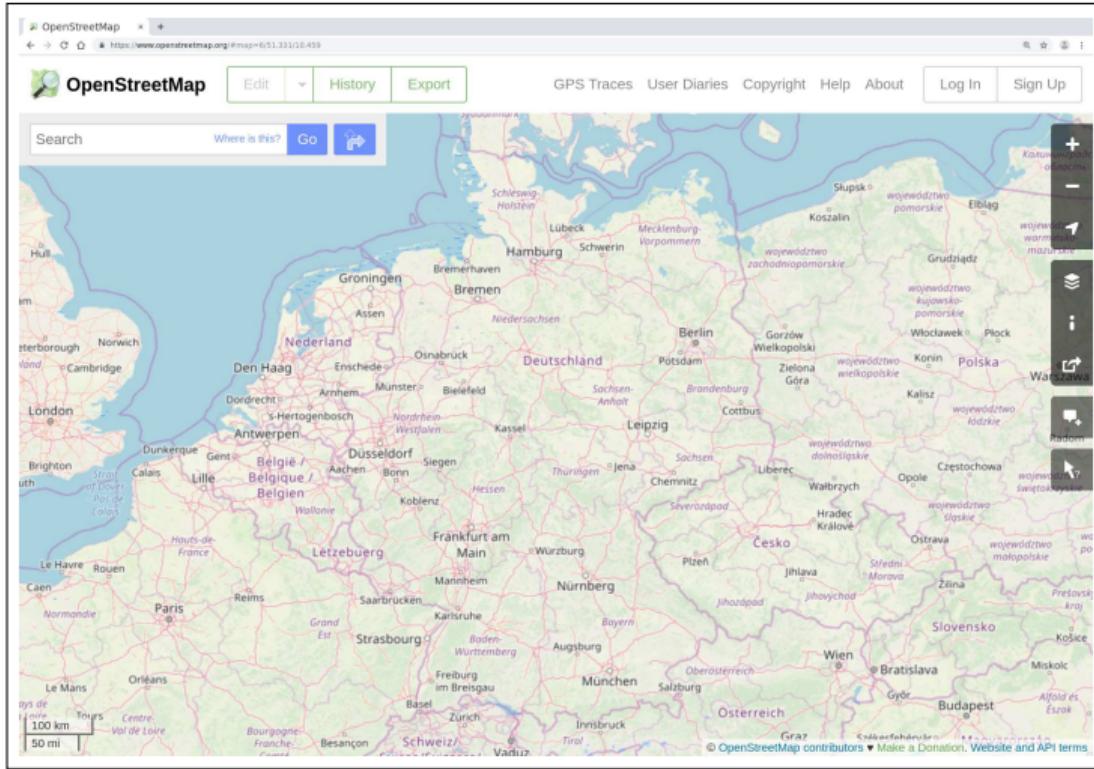
Heidelberg

Special-Interest Maps

The collage illustrates six different types of special-interest maps:

- Open Infrastructure Map:** A map showing a dense network of infrastructure, likely pipelines or utility lines, overlaid on a terrain map.
- Israel Hiking Map:** A detailed topographic map of Israel with marked hiking trails.
- Open Camping Map:** A map showing campsite locations across a region.
- Campsite:** A specific map entry for "Camping municipal de Gamsheim" with contact information and a location marker.
- Heidelberg:** A map of Heidelberg, Germany, highlighting various public services and facilities.
- Heidelberg, Australia:** A map of Heidelberg, Australia, showing streets and landmarks.

Searching the Map



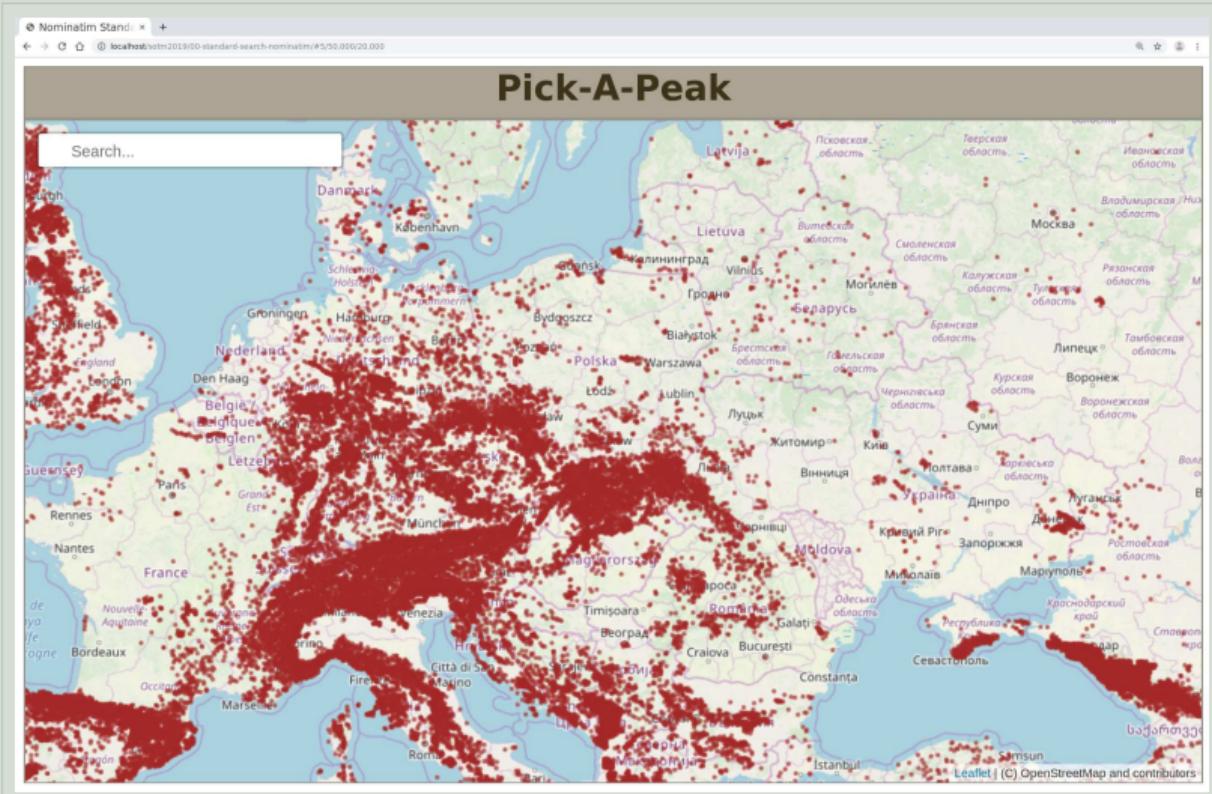
The example website: Mountain explorer

Ingredients

- Leaflet with Mapnik-style base map
- specialised data: OSM nodes with `natural=mountain` and `natural=volcano`
- Leaflet-control-geocoder for searching
<https://github.com/perliedman/leaflet-control-geocoder>

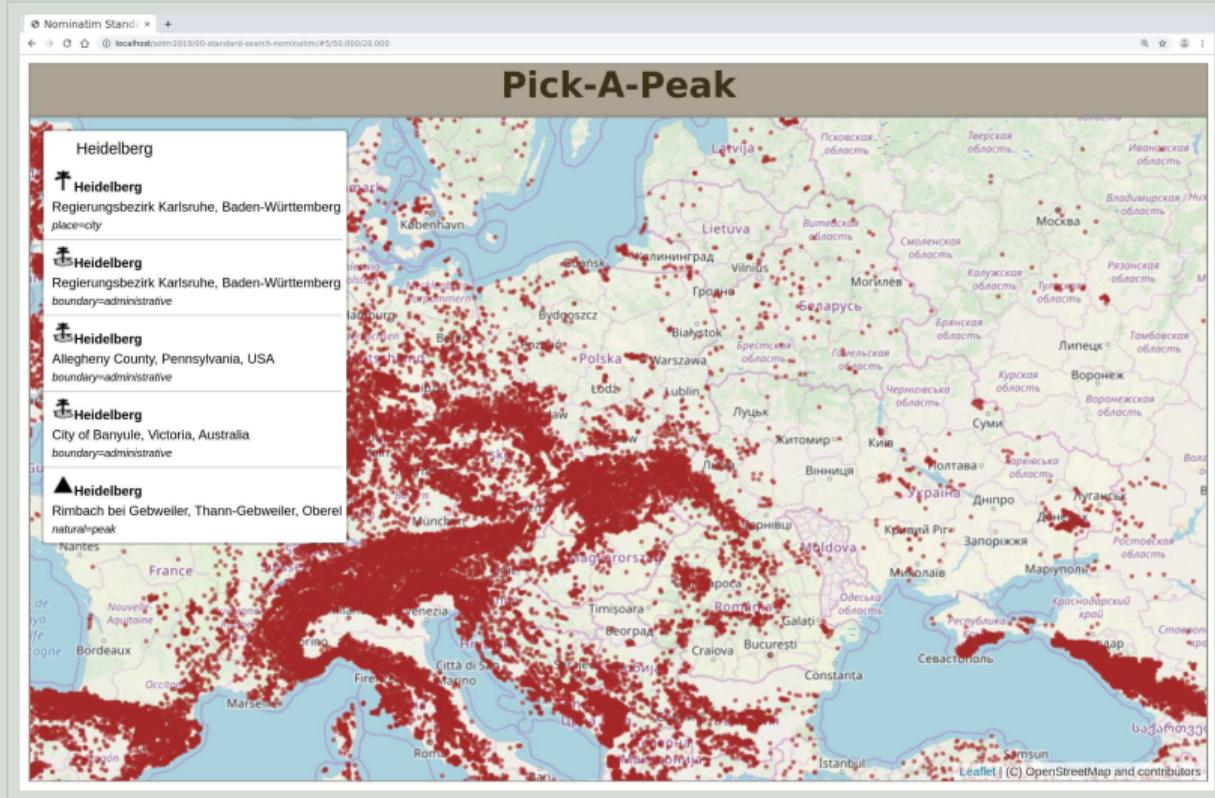
Geocoders

- Nominatim <https://nominatim.openstreetmap.org>
- Photon <https://photon.komoot.org>



<https://github.com/lonvia/pick-a-peak-examples>

Tuning the Search Engine



<https://github.com/lonvia/pick-a-peak-examples>
00-standard-search-nominatim

Filters exclude results

Example

- Nominatim: countrycodes=de, ch
- Photon: osm_tag=amenity:restaurant

Ranking reorders the results

Example

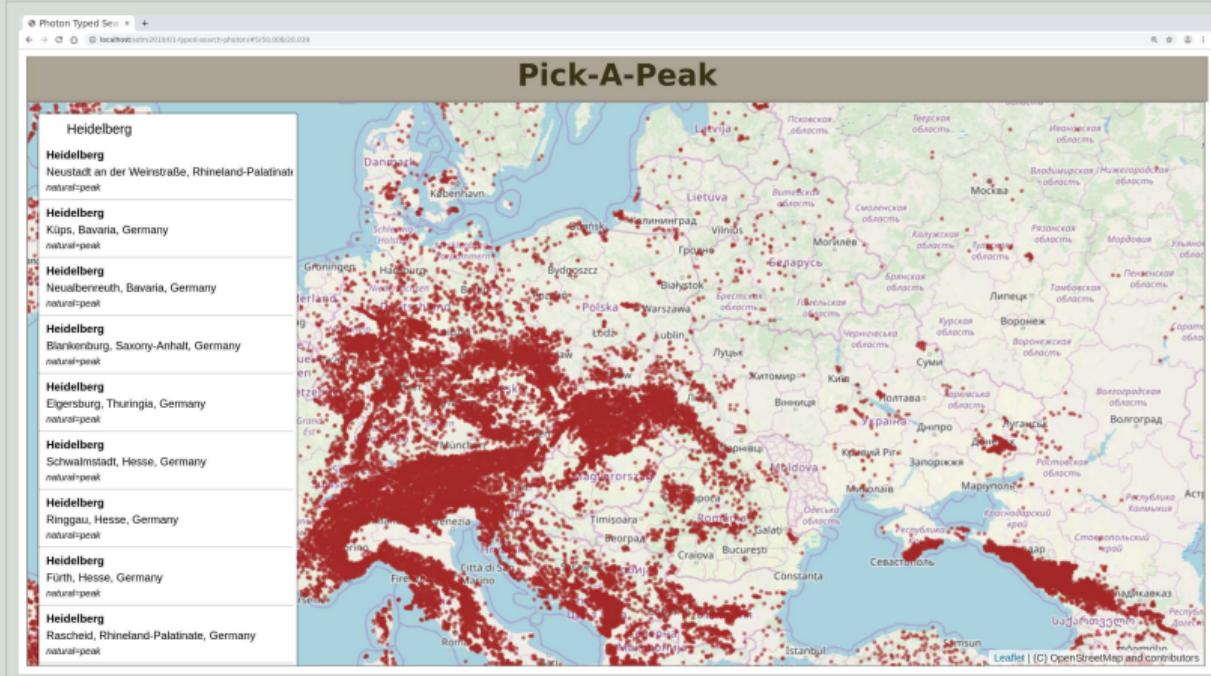
- Photon: lat=52.3879&lon=13.0582
- Nominatim: viewbox=1.0,1.0,2.0,2.0

Photon

https://photon.komoot.org/q=Heidelberg&osm_tag=natural:peak

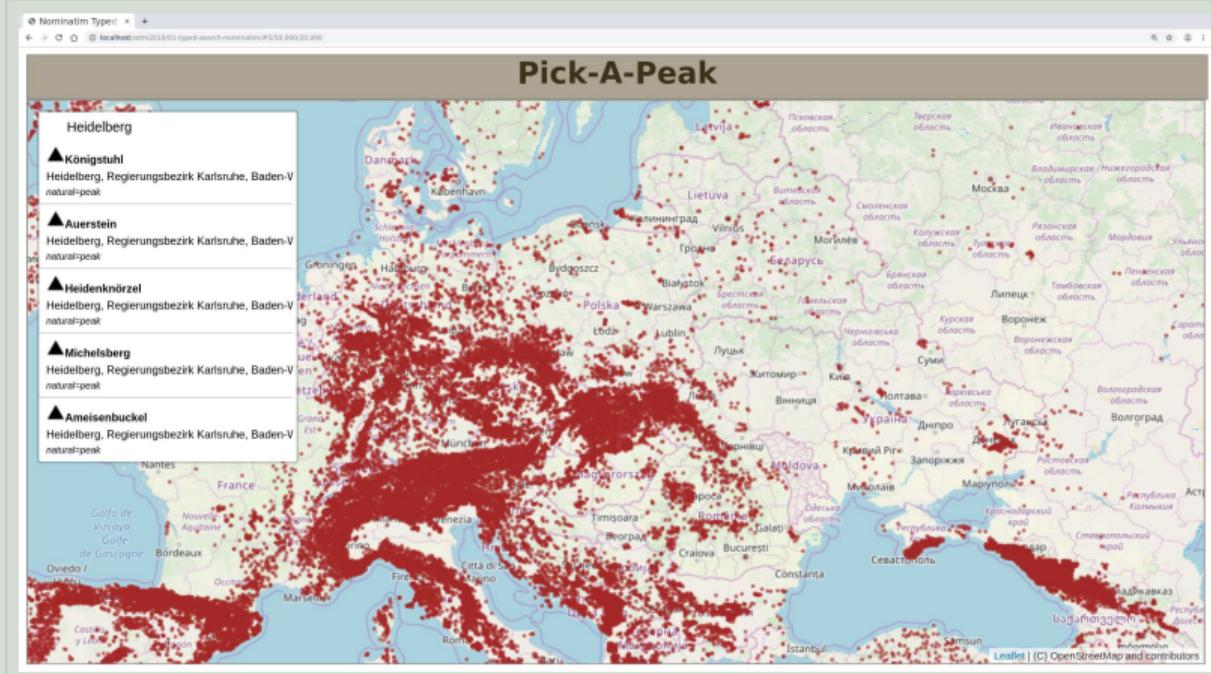
Nominatim

[https://nominatim.osm.org/q=Heidelberg \[natural=peak\]](https://nominatim.osm.org/q=Heidelberg [natural=peak])



<https://github.org/lonvia/pick-a-peak-examples>

01-typed-search-photon



<https://github.org/lonvia/pick-a-peak-examples>

01-typed-search-nominatim

Nominatim `q=Everest [natural=peak]`

- finds mountains *named* Everest
- finds mountains *near* Everest

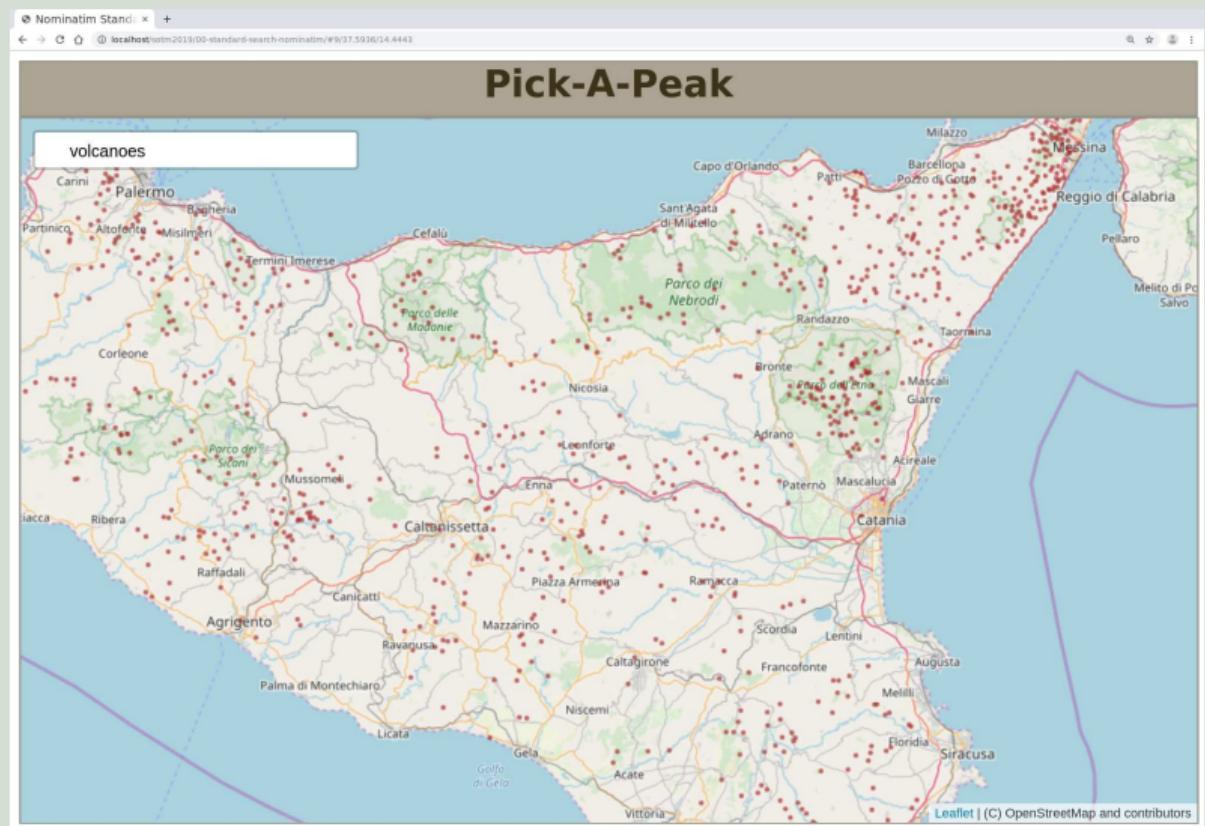
Photon `q=Everest&osm_tag=natural:peak`

- finds mountains *named* Everest
- finds mountains with Everest *in the address*

Geocoders are...

- ... optimised for generic use-case
- ... built for name search
- ... tuned to give you best matches, not all matches

Adding a screwdriver: Overpass



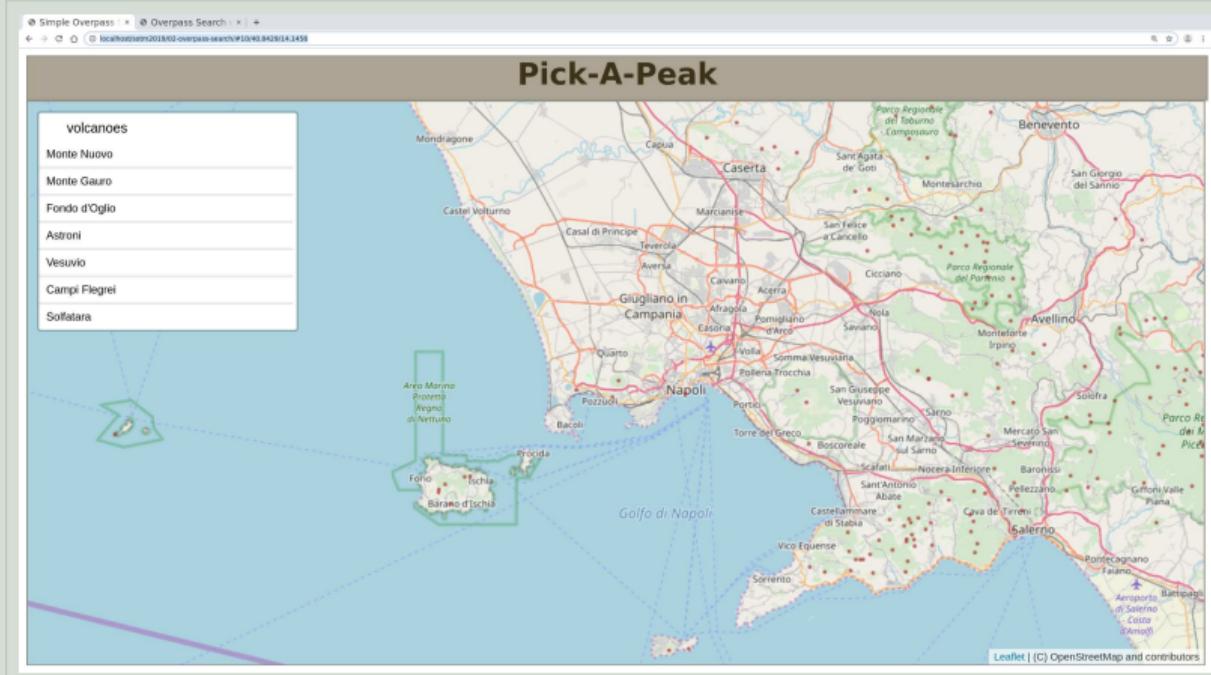
Find mountains with Overpass

The screenshot shows the Overpass Turbo web application. At the top, there's a navigation bar with "Run", "Share", "Export", "Wizard", "Save", "Load", "Settings", "Help", and "overpass turbo". To the right of the bar is a map of a city area, likely Rome, with various streets, landmarks, and green spaces. A small "Map" and "Data" button is also visible.

The main content area contains a code editor window on the left and a query editor window on the right. The code editor window displays the following Overpass query:

```
/*
This has been generated by the overpass-turbo
wizard.
The original search was:
"natural=volcano"
*/
[out:json][timeout:25];
// gather results
{
    // query part for: "natural=volcano"
    node["natural"="volcano"]({{bbox}});
    way["natural"="volcano"]({{bbox}});
    relation["natural"="volcano"]({{bbox}});
};
// print results
out body;
>;
out skel qt;
```

The query editor window has a title "Query Wizard" and a text input field containing "natural=volcano". Below the input field is a descriptive message: "The wizard assists you with creating Overpass queries. Here are some usage examples:" followed by a bulleted list: "Drinking Water", "highway=" and type:way", and "tourism=museum in Vienna". At the bottom of the query editor are three buttons: "build query", "build and run query", and "cancel".

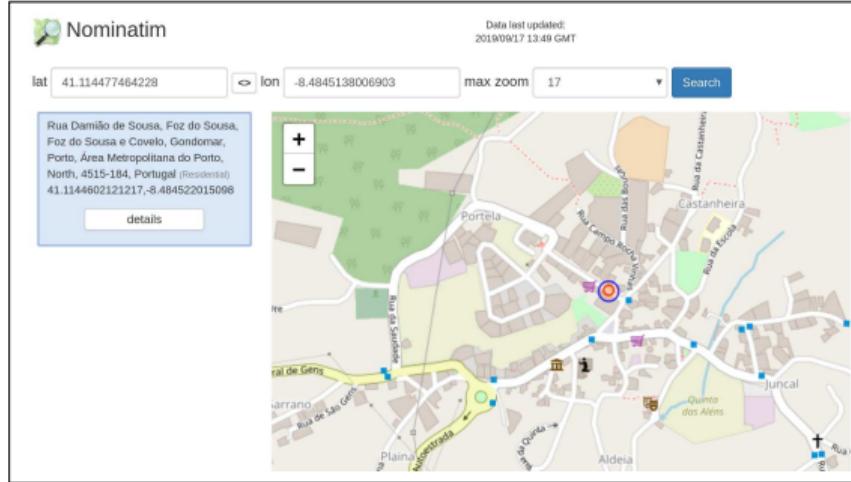


<https://github.com/lonvia/pick-a-peak-examples>

02-overpass-search

Adding Location Description: Reverse Lookup

Reverse Geocoding computes the address of a given coordinate



Blocking Danger!

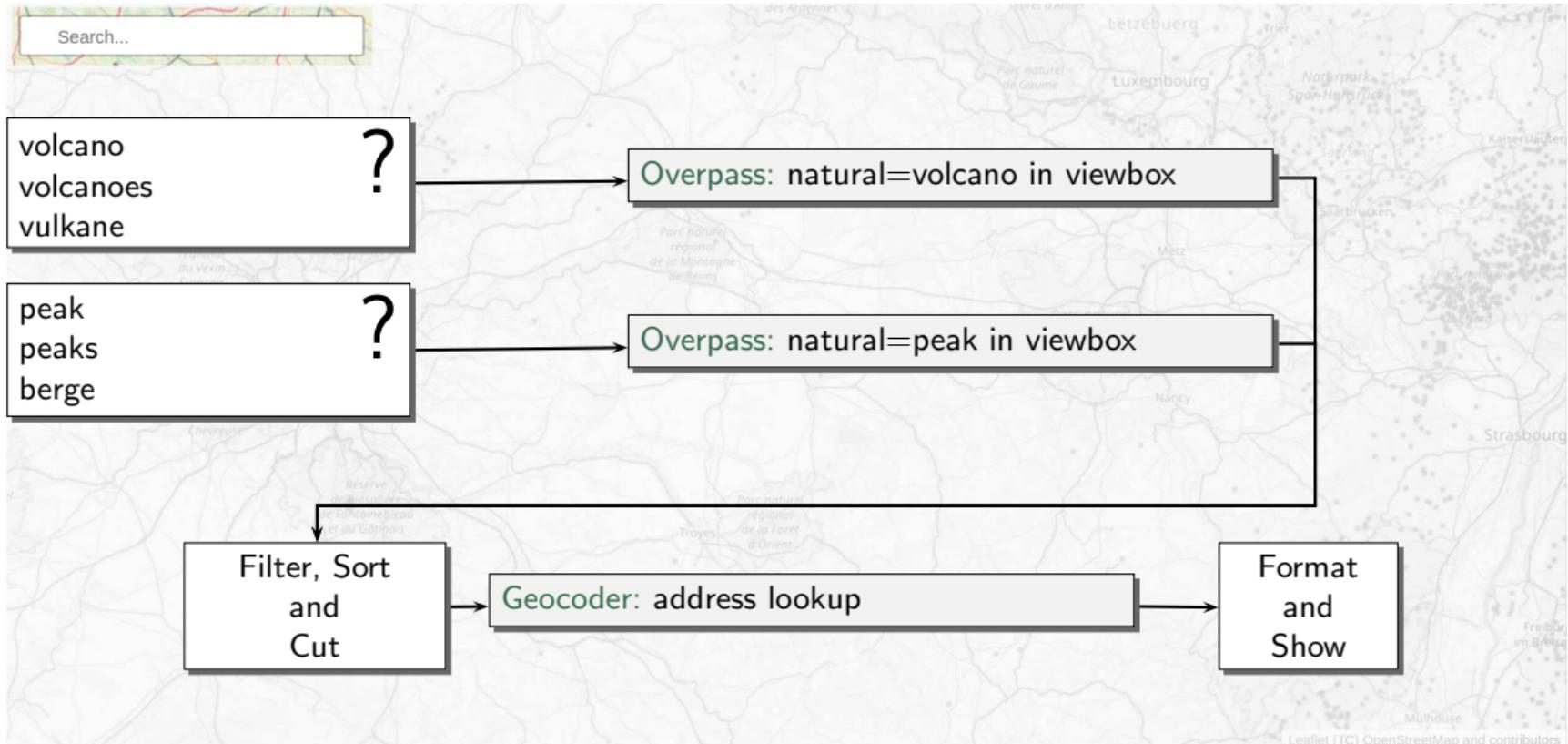
... when sending one request per result.

`nominatim.osm.org/lookup.php?osm_ids=[N|W|R]<value>,...`

OSM object type OSM object ID

- batch address resolution for OSM objects
- restriction: Nominatim needs to know object

Extended Type Search



Simple Overpass • Overpass Search •

localhost:5000/2018/03-overpass-search-with-address/#1048.840914.1456

Pick-A-Peak

volcanoes

Mount Vesuvius
Ercolano, Metropolitanstadt Neapel, Campania, 81
`natural=volcano`

Monte Gaurio
Pozzuoli, Metropolitanstadt Neapel, Campania, 90
`natural=volcano`

Fondo d'Oglio
Casamicciola Terme, Metropolitanstadt Neapel, C
`natural=volcano`

Monte Nuovo
Pozzuoli, Metropolitanstadt Neapel, Campania, 80
`natural=volcano`

Astroni
Pozzuoli, Metropolitanstadt Neapel, Campania, 90
`natural=volcano`

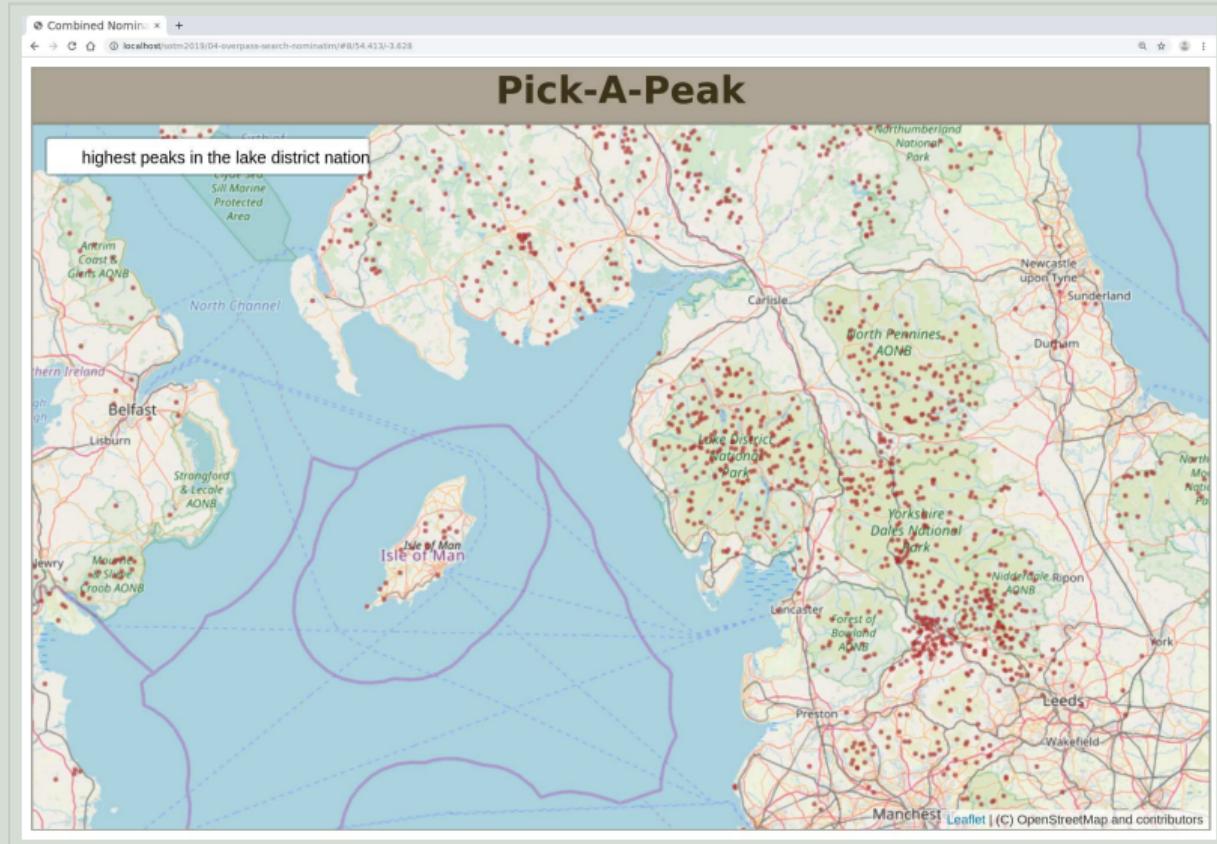
Phlegraean Fields
Via San Gennaro Agnano, Rione La Solfatara, Po
`tourism=attraction`

Solfatara
Via San Gennaro Agnano, Rione La Solfatara, Po
`tourism=attraction`

The map displays the Gulf of Naples and the coastal areas of Campania, Italy. Key locations labeled include Mount Vesuvius, Monte Gaurio, Fondo d'Oglio, Monte Nuovo, Astroni, Phlegraean Fields, and Solfatara. The map also shows numerous towns and cities such as Naples, Caserta, Avellino, and Salerno. A green polygon highlights the Phlegraean Fields area, and a blue polygon highlights the Solfatara area. The map includes a legend for natural=volcano and tourism=attraction, and a copyright notice for Leaflet | © OpenStreetMap and contributors.

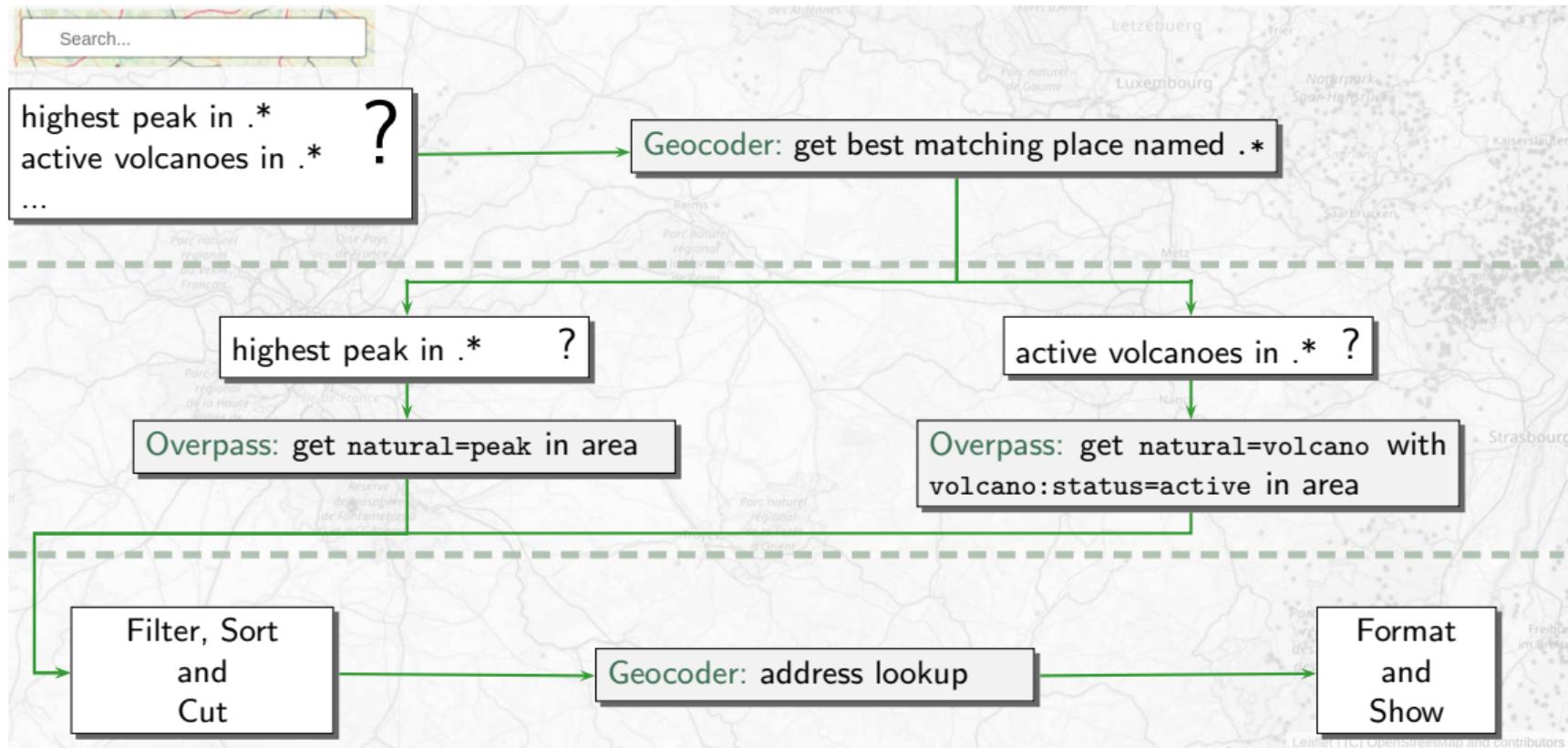
<https://github.com/lonvia/pick-a-peak-examples>

03-overpass-search-with-address



<https://github.com/lonvia/pick-a-peak-examples>
04-overpass-search-nominatim

Teaming up Overpass with the Geocoder



Combined Nomination

highest peaks in the lake district national park

Scafell Pike
Copeland, Cumbria, North West England, England
natural=peak

Sca Fell
Copeland, Cumbria, North West England, England
natural=peak

Symonds Knott
Copeland, Cumbria, North West England, England
natural=peak

Helvellyn
Eden, Cumbria, North West England, England
natural=peak

III Crag
Copeland, Cumbria, North West England, England
natural=peak

Broad Crag
Copeland, Cumbria, North West England, England
natural=peak

Skiddaw
Allerdale, Cumbria, North West England, England
natural=peak

Lower Man
Eden, Cumbria, North West England, England
natural=peak

Great End
Copeland, Cumbria, North West England, England
natural=peak

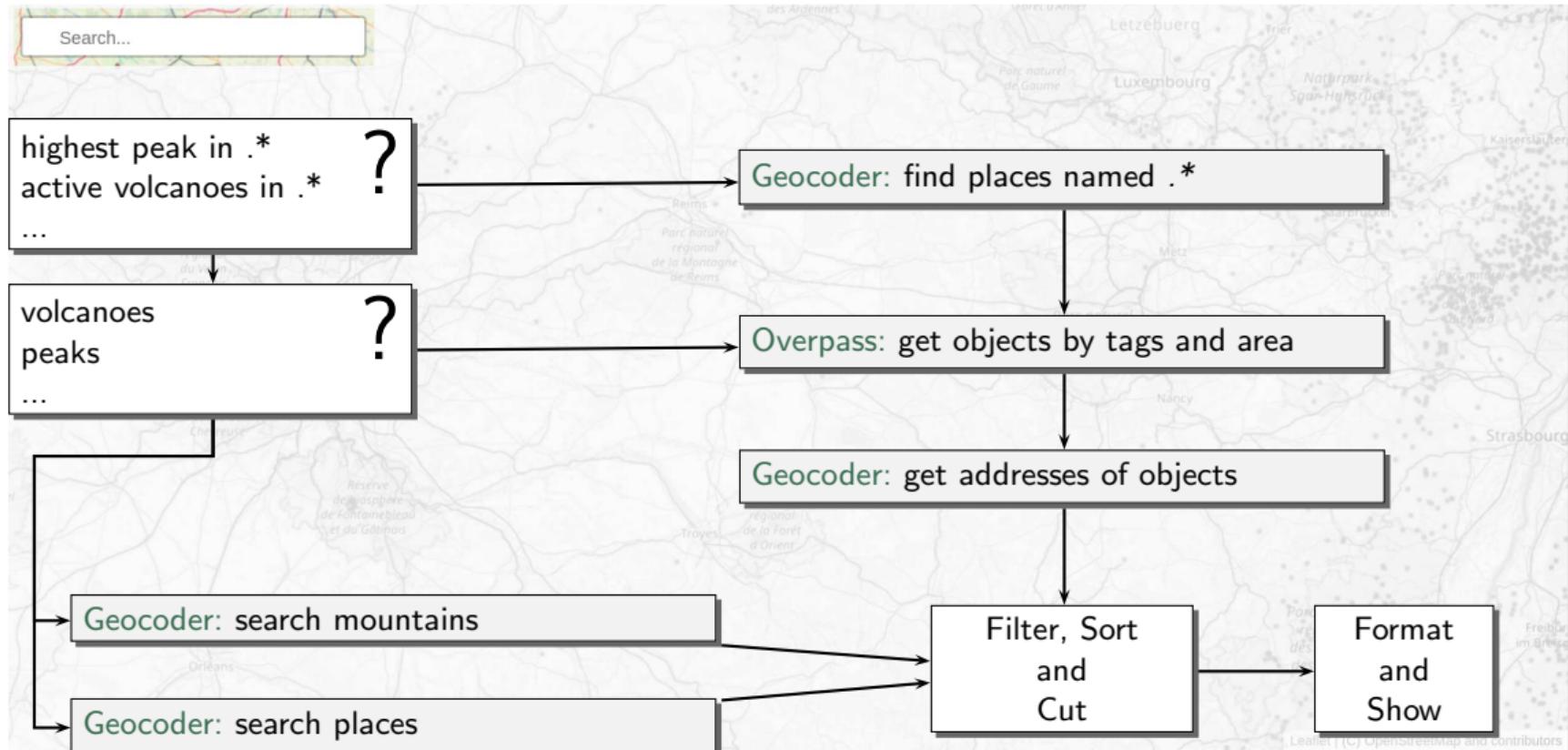
Pick-A-Peak

Scafell Pike
Copeland, Cumbria, North West England, England,
CA20 1EX, United Kingdom
natural=peak

<https://github.com/lonvia/pick-a-peak-examples>

04-overpass-search-nominatim

Putting it all together



- rather slow
- no real name search
- no search-as-you-type and spelling corrections

Building Your Own Search Database

SQLite

- lightweight
- usable for off-line search
- search-as-you-type: yes, spelling correction: no

Postgres

- heavyweight
- popular OSM data storage
- search-as-you-type: yes, spelling correction: yes

Normalisation

- remove capitalisation
- normalise diacritics
- (abbreviations)
- (spelling)

Tokenization

- split query into terms which are matched and indexed
- customized splitting for better results

Create the Database

```
CREATE VIRTUAL TABLE search  
USING FTS5(id UNINDEXED, ele UNINDEXED, term);
```

data columns

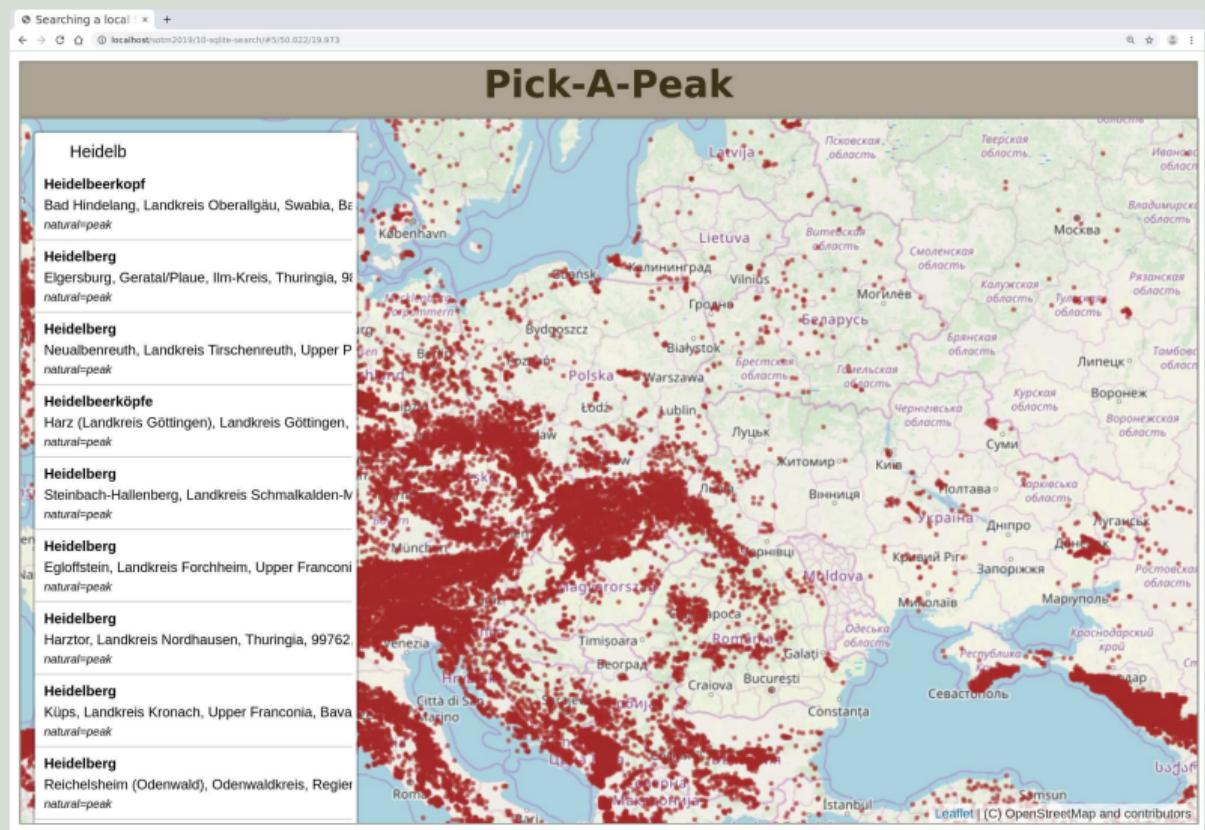
search text

Query the Database

```
SELECT id, ele, term  
FROM search WHERE term MATCH 'heidelb*' ORDER BY rank LIMIT 20;
```

match term begin

best matches



<https://github.com/lonvia/pick-a-peak-examples>

10-sqlite-search

Prerequisite

```
CREATE TABLE planet_osm_point  
(id TEXT, natural TEXT, name TEXT, ele TEXT, way GEOMETRY);
```

Create the Database

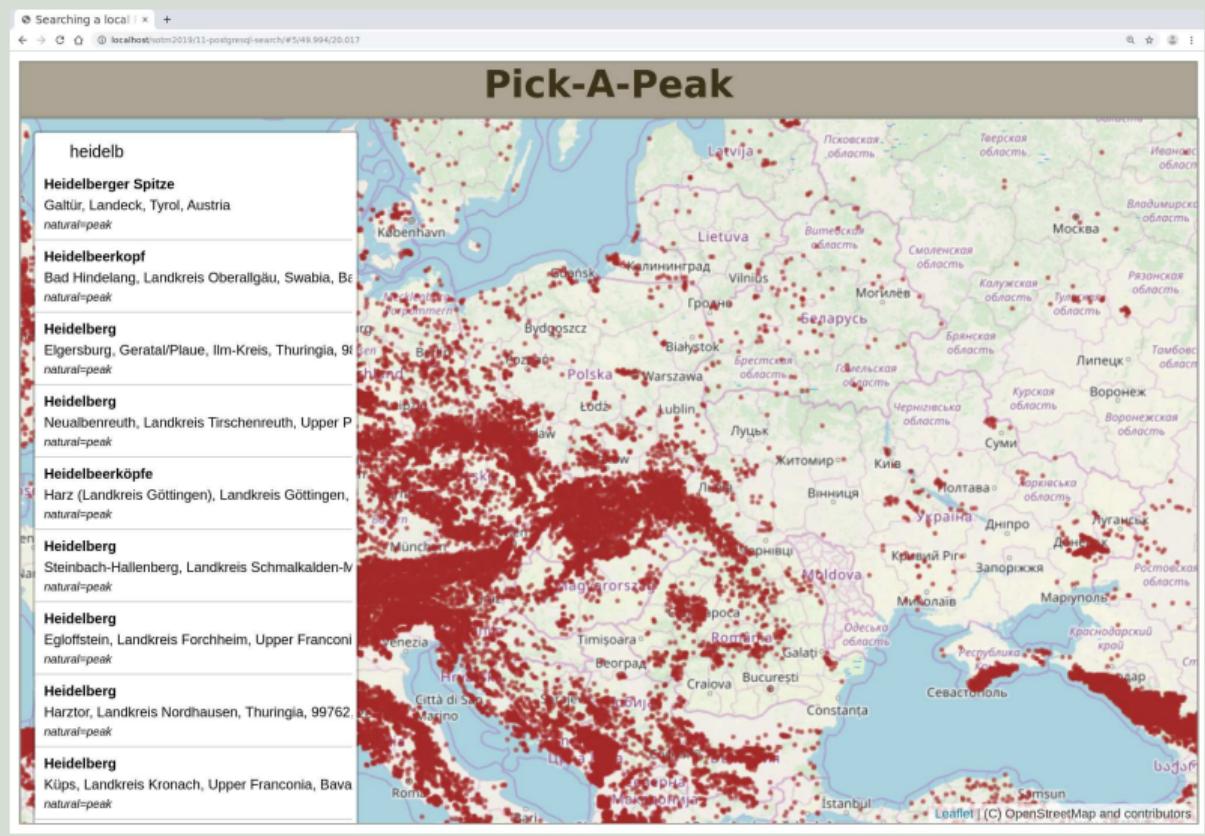
```
CREATE EXTENSION unaccent;

CREATE TABLE search AS
  SELECT id, ele, name, way,
         to_tsvector('simple', unaccent(name)) as terms
    FROM planet_osm_point
   WHERE natural in ('peak', 'volcano') AND name IS NOT NULL;

CREATE INDEX term_idx ON search USING gin(terms);
```

Query the Database

```
SELECT id, tags, name
  FROM search
 WHERE terms @@ unaccent('Heidelb:*')::tsquery
 ORDER BY ts_rank(terms, unaccent('Heidelb:*')::tsquery)
LIMIT 20
```



<https://github.org/lonvia/pick-a-peak-examples>

11-postgresql-search

Postgresql with Trigrams (1)

Create the Database

```
CREATE EXTENSION unaccent;  
CREATE EXTENSION pg_trgm;  
  
CREATE TABLE search AS  
  SELECT id, ele, name, way, lower(unaccent(name)) as terms  
    FROM peaks  
   WHERE natural in ('peak', 'volcano') AND name IS NOT NULL;  
  
CREATE INDEX term_idx ON search USING gist(terms gist_trgm_ops);
```

Query the Database

```
SELECT id, ele, name, way
      FROM search
     WHERE terms %% lower(unaccent('Mount Rainer'))
ORDER BY terms <-> unaccent('Mount Rainer')
LIMIT 20
```

Searching a local | +

localhost:9019/12-postgresql-fuzzy-search/#8/46.614/-122.333

mount rainer

Mount Rainier
Pierce County, Washington, USA
`natural=volcano`

Rain Mountain
Apache County, Arizona, USA
`natural=peak`

Mount Raimer
Town of Berlin, Rensselaer County, New York, US
`natural=peak`

Rainey Mountain
Rabun County, Georgia, USA
`natural=peak`

Mount Miner
Queensland, Australia
`natural=peak`

Mount Ram
New Salem, Franklin County, Massachusetts, Uni
`natural=peak`

Mount Miner
Auburn, Rockingham County, New Hampshire, US
`natural=peak`

Mount Rainey
Regional District of Kitimat-Stikine, British Colum
`natural=peak`

Mount Miner
Regional District of Okanagan-Similkameen, Britis
`natural=peak`

Mount Miner
Porter County, Indiana, 46304, USA

Mount Rainier
Pierce County, Washington, USA
`natural=volcano`

Leaflet | (C) OpenStreetMap and contributors

<https://github.com/lonvia/pick-a-peak-examples>

12-postgresql-fuzzy-search

Taking it further

- Elastic Search
- customized geocoder installations

Just remember...

- ... think about what your users want to find.
- ... look for services next to traditional geocoders.

Thank You

<https://github.org/lonvia/pick-a-peak-examples>

lonvia@denofr.de