Customizing Search for Special-Interest Maps

Sarah Hoffmann
lonvia@denofr.de

28th July 2018

State of the Map 2019
Heidelberg
Special-Interest Maps
Searching the Map

OpenStreetMap

Search

Where is this?

Go
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
Tuning the Search Engine
https://github.org/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
Filter for Mountains

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

Nominatim
https://nominatim.os.org/q=Heidelberg [natural=peak]
Pick-A-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
Named searches

**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

/*
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;

Query Wizard

natural=volcan0

The wizard assists you with creating Overpass queries. Here are some usage examples:
- Drinking Water
- highway="" and type="way"
- tourism=museum in Vienna

[build query] [build and run query] [cancel]
Pick-A-Peak

https://github.org/lonvia/pick-a-peak-examples
02-overpass-search
Reverse Geocoding computes the address of a given coordinate.

Blocking Danger!
... when sending one request per result.

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

Search...

volcano
volcanoes
vulkane

peak
peaks
berge

?  Overpass: natural=volcano in viewbox

?  Overpass: natural=peak in viewbox

Filter, Sort and Cut

Geocoder: address lookup

Format and Show
Pick-A-Peak

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

03-overpass-search-with-address
Teaming up Overpass with the Geocoder

Geocoder: get best matching place named .*

Overpass: get natural=peak in area

Overpass: get natural=volcano with volcano:status=active in area

Filter, Sort and Cut

Geocoder: address lookup

Format and Show
highest peaks in the lake district

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

Ill 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

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

https://github.org/lonvia/pick-a-peak-examples
04-overpass-search-nominatim
Limits of Overpass

- rather slow
- no real name search
- no search-as-you-type and spelling corrections
Building Your Own Search Database
SQLite vs. Postgres

**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);

Query the Database

SELECT id, ele, term
FROM search WHERE term MATCH 'heidelberg*' ORDER BY rank LIMIT 20;
Pick-A-Peak

https://github.org/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
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
```
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