Geocoding V

Max Kleiner

2 min read · 8 hours ago

▶ Listen   ⬆ Share   ••• More

Have you ever come across a dataset having addresses like below?

location, lat, long, maplink
"Place Grévy, Azans, Dole,",47.094812599999994,5.4972803,geocode_dole_layer1714069.png
"Place Grévy, France, Dole,",47.094812534546784,5.4972988,geocode_dole_layer1675941.png

This is the third step story of Geocoding namely

1. Get coordinates of an location (address)

2. Show a map of location interest

3. Store the data in a fast csv file

As you may konw **Geocoding** is the process of converting addresses or locations into geographic coordinates (i.e. latitude and longitude) and **Reverse Geocoding** is the process of converting geographic coordinates (latitude & longitude) into a human-readable address. Then services like mapbox or OpenStreetMap provide APIs which can be used by anyone. First we start with two constants:

Const
GEOCSV_BASE = 'geocoding61_12_base.txt';
GEO_LOCATION = '1 Place Grevy, Dole, France';

Then we call our first function:

> *latlong:= TAddressGeoCodeOSM5(GEO_LOCATION);*

and we get as Json description **OSM (OpenStreetMap) res back_:**

> *Coords: lat 47.09481 lng 5.49728 1, Place Grévy, Azans, Dole, Jura, Bourgogne-Franche-Comté, France métropolitaine, 39100, France place_id: 111925270*
> *get geocoords: lat: 47.0948 — lon: 5.4973*
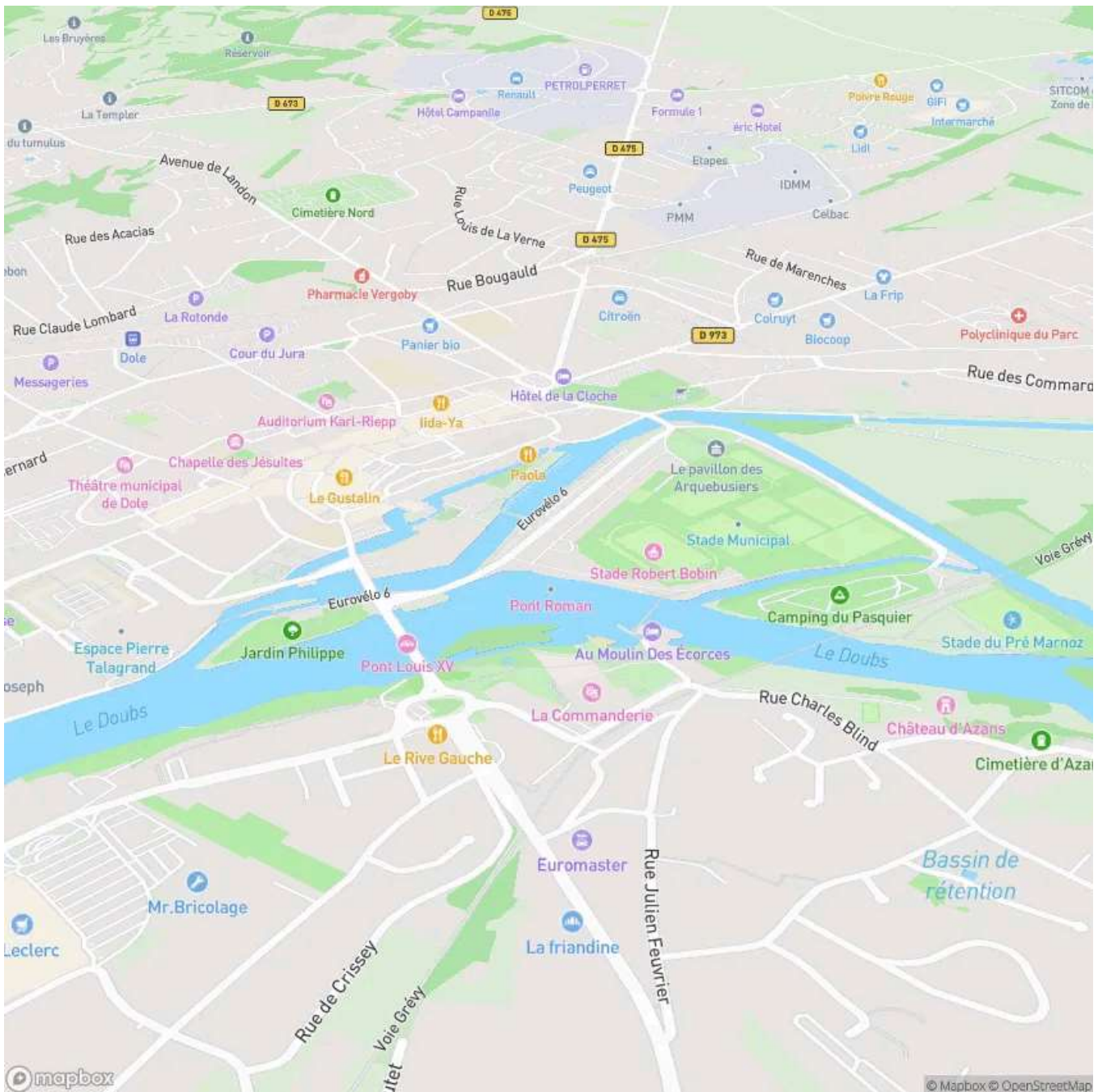
The Get API looks like the following:

TAddressGeoCodeOSM5 pass: https://nominatim.openstreetmap.org/search?format=json&q=1%20Place%20Grevy,%20Dole,%20France
Check Url: 200

Next we call

> *GetGeoInfoMap5(**latlong.lat,latlong.long**,15,850,'yourAPIKey',*
> *ExePath+'examples\geocode_dole_layer5.png', true);*

and get a map

Dole Hôtel de la Cloche

Parameter 15 is zoom-factor and 850 the size of the png-graphic. The script you find at: softwareschule.ch/examples/geocoding7.txt

As the last step we store the data in a CVS file:

```
acsv:= TJvCSVBase.create(self);
    FieldNames:= TStringList.create;
    fieldnames.add('location')
    fieldnames.add('lat')
    fieldnames.add('long');
    fieldnames.add('maplink');
    if not fileExists( exepath+GEOCSV_BASE) then
```

```
    acsv.DataBaseCreate(exepath+GEOCSV_BASE, fieldnames, false, true) else be
      acsv.DataBaseOpen(exepath+GEOCSV_BASE);
      //writeln(acsv.CSVFieldNames[0])
      acsv.recordnew;
      acsv.CSVFieldNames[0]:= 'Place Grévy, Azans, Dole,';
      acsv.CSVFieldNames[1]:= flots(latlong.lat);
      acsv.CSVFieldNames[2]:= flots(latlong.long);
      acsv.CSVFieldNames[3]:= 'geocode_dole_layer'+IBRandomString(7)+'.png';
      acsv.Recordset(acsv.CSVFieldNames, false);
    //acsv.recordpost;
    //acsv.DisPlayFields
      acsv.DataBaseClose;
  end ;
   acsv.free
   fieldnames.free;
   openfile(exepath+GEOCSV_BASE);
```

and get a dataset file like above.

JvCSVBase, a collection **of** components **for** handling CSV database files, was created by JanSoft. It includes five components: TjvCSVBase, TjvCSVEdit, TjvCSVComboBox, TjvCSVCheckBox, **and** TjvCSVNavigator. These components in maXbox5 allow you to create, restructure, browse, **and** edit CSV database files without any programming12. **If** you're working **with** CSV data, JvCSVBase provides a convenient way **to** manage it within your script applications.

Then we do at a forth last step some unit tests with reverse geocoding:

> *Testunit From Cologne **to** Graz, Bern and St. Ingbert and Dole*
>
> *get geocoords: lat: 50.9473 — lon: 6.9503 Cologne*
>
> *get geocoords: lat: 47.0739 — lon: 15.4168 Graz*
>
> *get geocoords: lat: 46.9479 — lon: 7.44744 Bern*
>
> *get geocoords: lat: 49.2709 — lon: 7.11161 St. Ingbert*
>
> *get geocoords: lat: 47.0948 — lon: 5.49728 Dole*

Testcall OpenWeb('https://www.latlong.net/c/?
lat='+flots(latlong.lat)+'&long='+flots(latlong.long));

maplink:
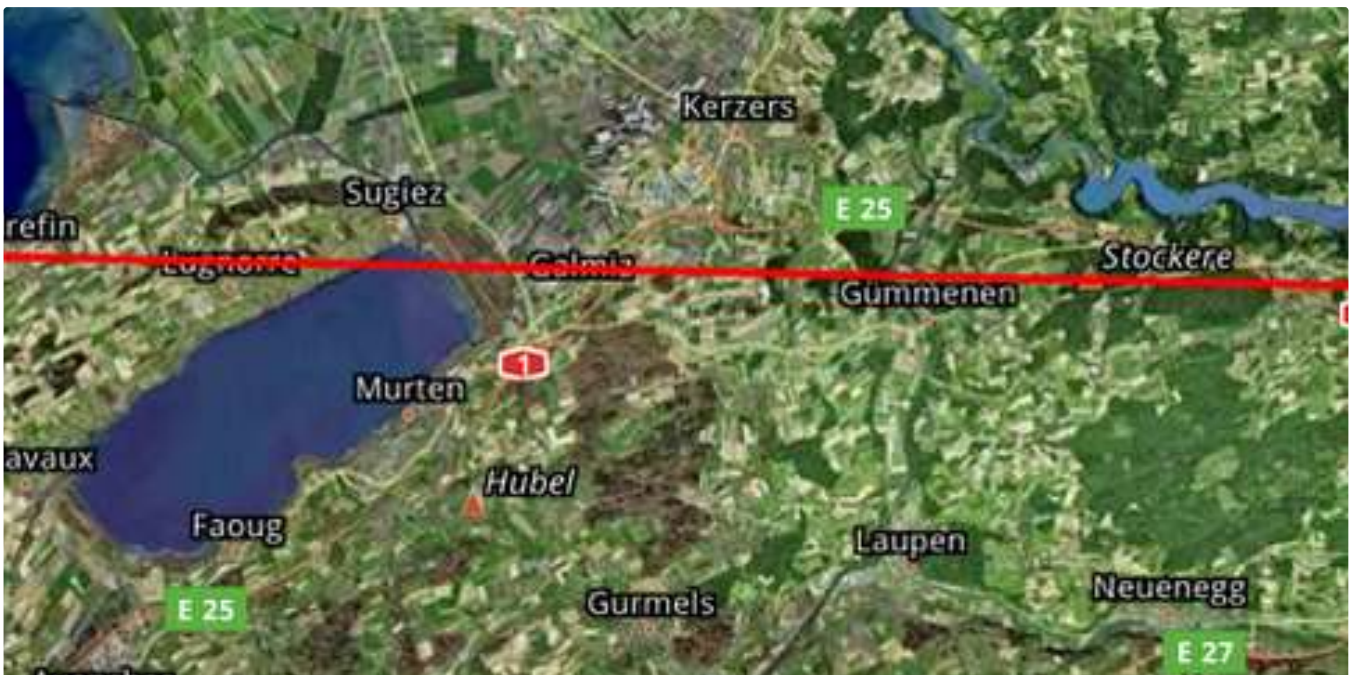https://maxbox4.files.wordpress.com/2024/03/1277_geocode_dole_layer5.png

Max Kleiner 21/03 2024

Written by Max Kleiner

22 Followers

Max Kleiner's professional environment is in the areas of OOP, UML and coding - among other things as a trainer, developer and consultant.

More from Max Kleiner

Max Kleiner in Nerd For Tech

Air Distance and Bearing

As part of a unit on Trigonometry, we review compass and true bearings before working with bearings in Trigonometry problems. To start this...

```
29 - 36 bytes: TJsonValue x 2114313, String x 112
37 - 44 bytes: String x 106
45 - 52 bytes: TSynEditFoldRange x 4565, String x 108
53 - 60 bytes: TStringList x 4614, String x 2114382
61 - 68 bytes: String x 121
69 - 76 bytes: String x 114, Unknown x 94
77 - 84 bytes: String x 126
85 - 92 bytes: String x 91
93 - 100 bytes: String x 20
101 - 108 bytes: String x 63
109 - 116 bytes: String x 1
125 - 132 bytes: String x 1
173 - 188 bytes: Unknown x 16
381 - 412 bytes: Unknown x 16
797 - 876 bytes: Unknown x 1
1053 - 1148 bytes: TChart x 39
```

Max Kleiner

JSON Automation 2

//

Max Kleiner

Train a Logic Classifier

The model below this article depicts the architecture for a multilayer perceptron network designed specifically to solve the XOR problem...

6 min read · Feb 16, 2021

12

Max Kleiner