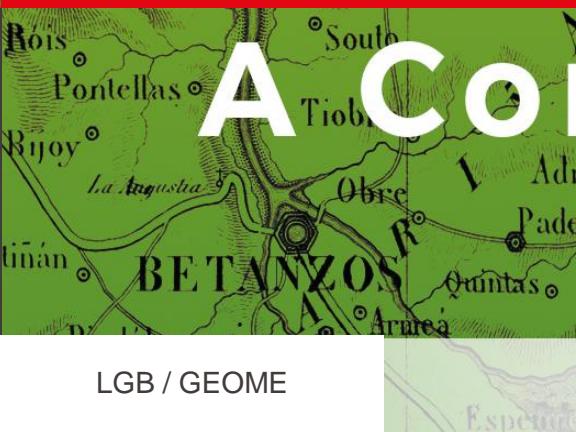


# SIG - Systèmes d'Information Géographiques

Gabriel Kathari  
Présentation : Le  
géocodage en  
géoinformatique



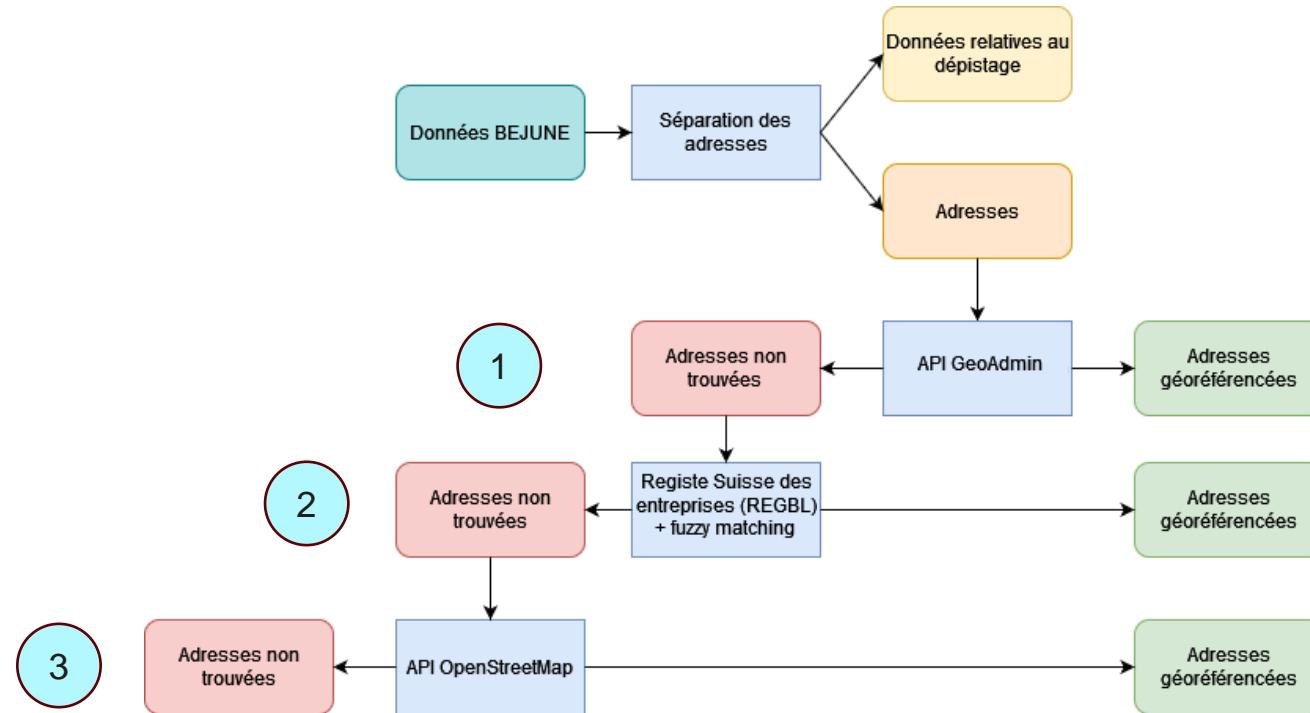
# Le géocodage

- Permet de transformer des adresses en coordonnées X,Y.



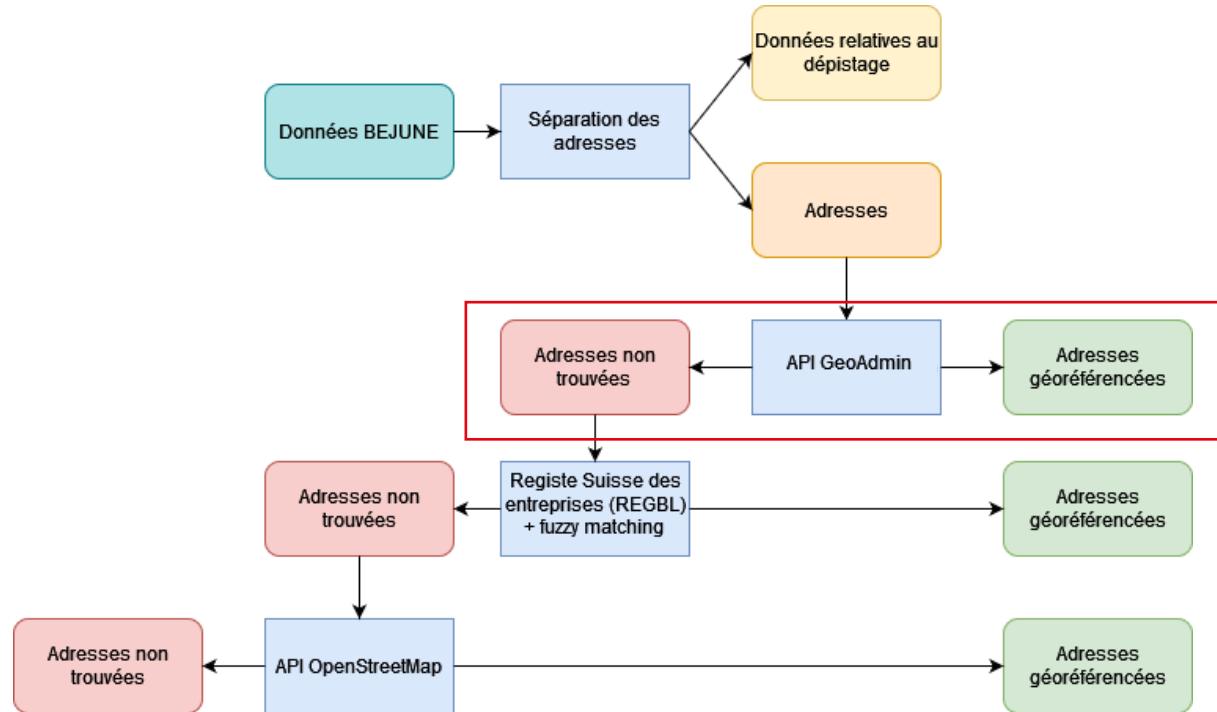
# Le géocodage

- Comment s'y prendre en suisse ? (Exemple BEJUNE)



# Le géocodage

- Comment s'y prendre en suisse ? (Exemple BEJUNE)





Schweizerische Eidgenossenschaft  
Confédération suisse  
Confederazione Svizzera  
Confederaziun svizra



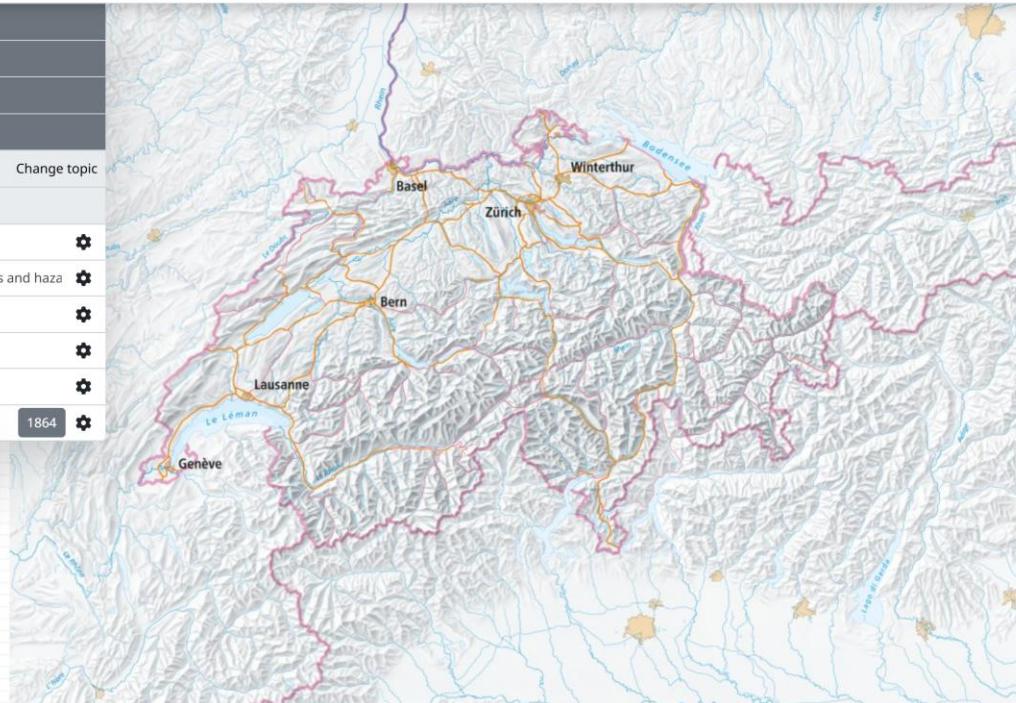
Search for addresses, parcels or maps

[Report problem](#) ▾ [News](#) [Help](#) [DE](#) [FR](#) [IT](#) [EN](#) [RM](#)

In collaboration with the cantons

- ▶ Share
- ▶ Print
- ▶ Draw & Measure on map
- ▶ Advanced tools
- ▶ Geocatalog Change topic
- ▼ Maps displayed
  - Closures Hiking trails ⚙
  - Published shooting range bulletins and haza ⚙
  - Hiking trails ⚙
  - Public transport stops ⚙
  - RBD: building status ⚙
  - Journey through time - Maps 1864 1864 ⚙

▲ Close menu

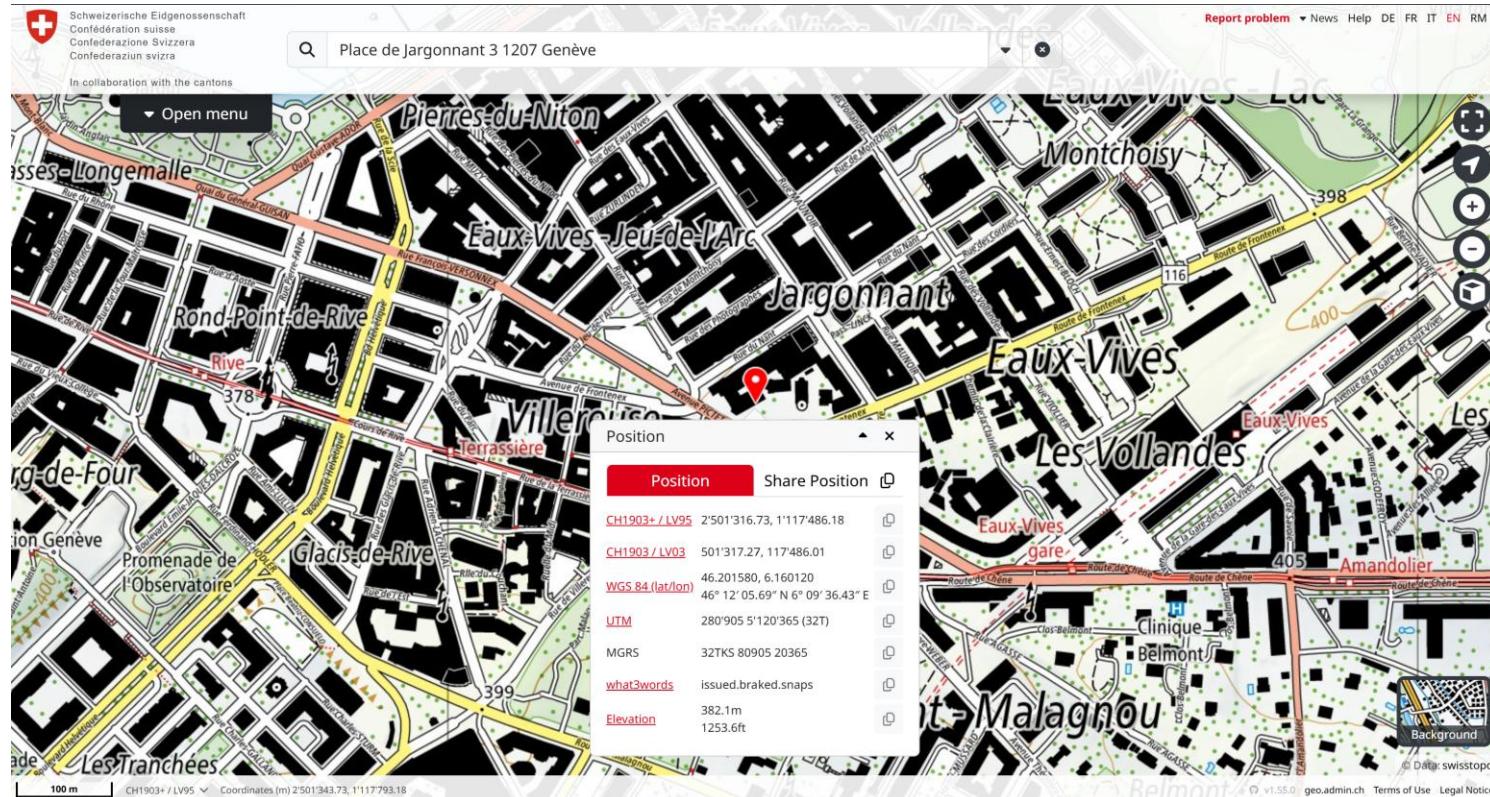


© Data: swisstopo

50 km

CH1903+ / LV95 ▾ Coordinates (m) 2'513'000.00, 1'330'750.00

v1.55.0 geo.admin.ch Terms of Use Legal Notice



# Geoadmin - API

- Permet de faire une requête avec Python vers l'API de geoadmin

```
def search_address(searchText):
    import time, requests
    SR = str(2056)
    retries = 0
    response = ''

    # Tentative de connexion jusqu'à 3 fois
    while response == '' and retries <= 3:
        try:
            base_url = "https://api3.geo.admin.ch/rest/services/api/SearchServer?"
            params = (
                f"layer=ch.bfs.gebaeude_wohnungs_register&searchText={searchText}&"
                f"type=locations&origins=address,zipcode&sr={SR}"
            )
            url = base_url + params
            response = requests.get(url)
            break
        except:
            time.sleep(5)
            retries += 1

    return response
```

Geoadmin\_API.py

# Geoadmin - API

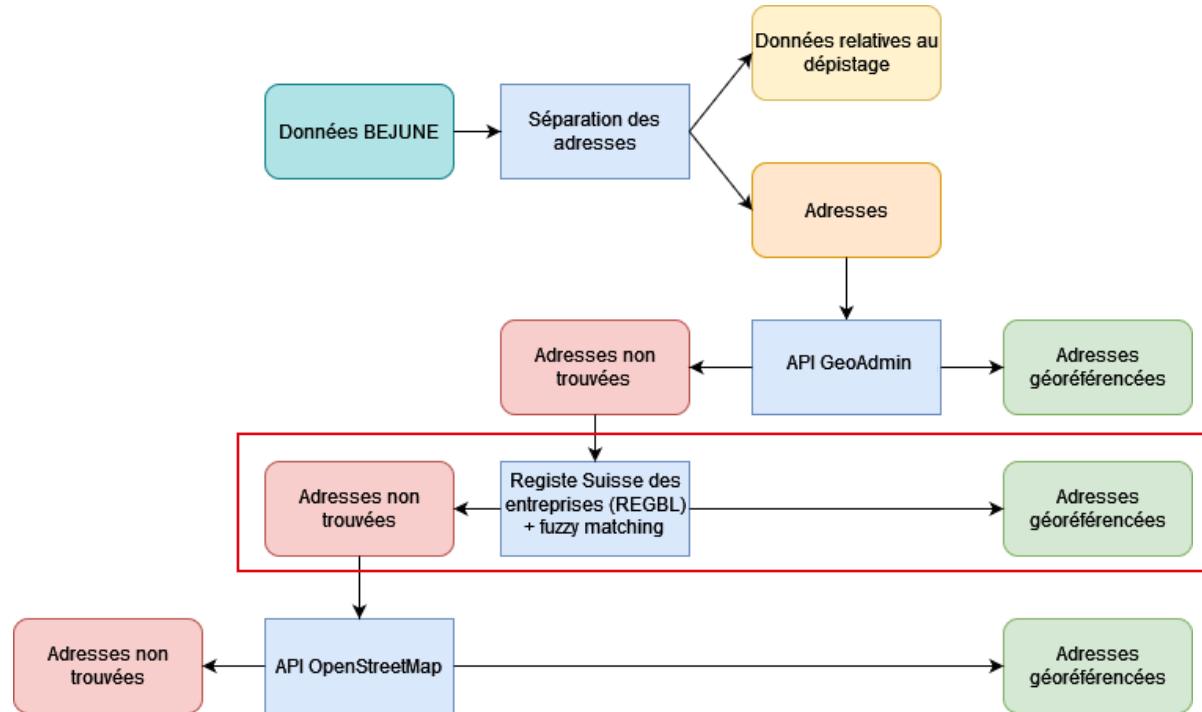
- La réponse du call API est un json :

```
{'results': [ {'attrs': { 'detail': 'place de jargonnant 3 1207 geneve 6621 geneve ch ge', 'featureId': '1014193_0', 'geom_quadindex': '022121013030313321200', 'geom_st_box2d': 'BOX(2501316.725 1117486.057,2501316.725 1117486.057)', 'label': 'Place de Jargonnant 3 <b>1207 Genève</b>', 'lat': 46.20158004760742, 'lon': 6.160120487213135, 'num': 3, 'objectclass': '', 'origin': 'address', 'rank': 7, 'x': 1117486.0, 'y': 2501316.75, 'zoomlevel': 10}, 'id': 670701, 'weight': 100} ]}
```

- Pour chaque adresse non-géoréférencée grâce au RegBL on lance un call API à Geoadmin. Si la réponse existe, on récupère les coordonnées X,Y données par Geoadmin.

# Le géocodage

- Comment s'y prendre en suisse ? (Exemple BEJUNE)



# RegBL – Registre Fédéral des bâtiments et des logements

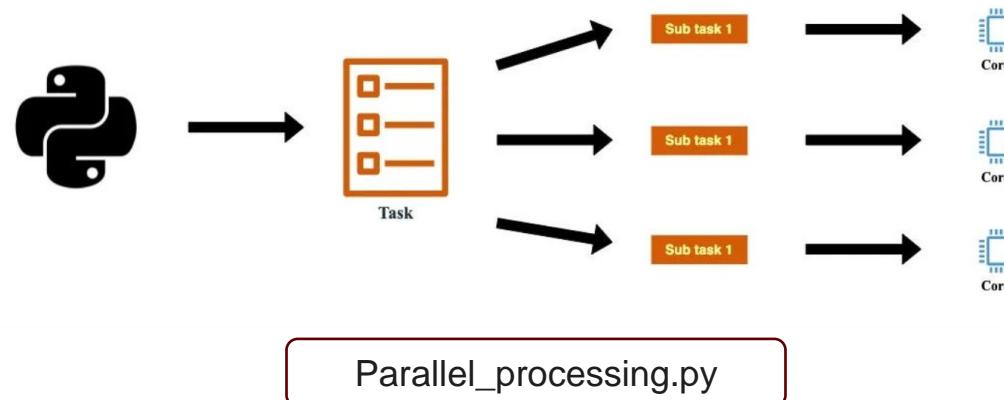
- Liste des adresses de tous les bâtiments et logements de Suisse avec leurs coordonnées X,Y en CH1903+.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
EGID	EDID	EGAID	DEINR	ESID	STRNAME	STRNAMK	STRINDX	STRSP	STROFFIZIE	DPLZ4	DPLZZ	DPLZNAME	DKODE	DKODN	DOFFADR	DEXPDAT	
1	1	0	1E+08	20	10000533 Grossholzerstrasse	Grossholzerstr.	Gro	9901	1	8910	0	Affoltern am Albis	2676465.141	1235849.103	1 #####	#####	
2	2	0	1E+08	1	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676541.16	1235979.043	0 #####	#####	
3	3	0	1E+08	2	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676529.561	1236015.749	1 #####	#####	
4	4	0	1E+08	3	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676510.997	1235979.419	1 #####	#####	
5	5	0	1E+08	4	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676506.529	1236015.595	1 #####	#####	
6	6	0	1E+08	5	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676485.838	1235978.3	1 #####	#####	
7	7	0	1E+08	6	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676487.492	1236016.425	1 #####	#####	
8	8	0	1E+08	7	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676459.071	1235977.814	1 #####	#####	
9	9	0	1E+08	8	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676468.428	1236016.779	1 #####	#####	
10	10	0	1E+08	9	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676444.333	1235975.496	1 #####	#####	
11	11	0	1E+08	10	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676449.928	1236016.255	1 #####	#####	
12	12	0	1E+08	12	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676430.85	1236015.782	1 #####	#####	
13	13	0	1E+08	13	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676412.604	1235987.289	1 #####	#####	
14	14	0	1E+08	14	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676411.824	1236016.833	1 #####	#####	
15	15	0	1E+08	16	10000537 Im Feld	Im Feld	Fel	9901	1	8910	0	Affoltern am Albis	2676389.71	1236008.799	1 #####	#####	
16	16	0	1E+08	1	10106397 Loorenfeldstrasse	Loorenfeldstr.	Loo	9901	1	8910	0	Affoltern am Albis	2676504.862	1236039.807	1 #####	#####	
17	17	0	1E+08	3	10106397 Loorenfeldstrasse	Loorenfeldstr.	Loo	9901	1	8910	0	Affoltern am Albis	2676483.031	1236040.833	1 #####	#####	
18	18	0	1E+08	5	10106397 Loorenfeldstrasse	Loorenfeldstr.	Loo	9901	1	8910	0	Affoltern am Albis	2676463.299	1236040.872	1 #####	#####	
19	19	0	1E+08	7	10106397 Loorenfeldstrasse	Loorenfeldstr.	Loo	9901	1	8910	0	Affoltern am Albis	2676442.345	1236040.821	1 #####	#####	
20	20	0	1E+08	9	10106397 Loorenfeldstrasse	Loorenfeldstr.	Loo	9901	1	8910	0	Affoltern am Albis	2676421.673	1236041.077	1 #####	#####	
21	21	0	1E+08	11	10106397 Loorenfeldstrasse	Loorenfeldstr.	Loo	9901	1	8910	0	Affoltern am Albis	2676400.051	1236036.41	1 #####	#####	
22	22	0	1E+08	12	10106397 Loorenfeldstrasse	Loorenfeldstr.	Loo	9901	1	8910	0	Affoltern am Albis	2676410.877	1236078.963	1 #####	#####	
23	23	0	1E+08	15	10000533 Grossholzerstrasse	Grossholzerstr.	Gro	9901	1	8910	0	Affoltern am Albis	2676550.435	1236043.666	1 #####	#####	
24	24	0	1E+08	17	10000533 Grossholzerstrasse	Grossholzerstr.	Gro	9901	1	8910	0	Affoltern am Albis	2676561.878	1236015.892	1 #####	#####	
25	25	0	1E+08	19	10000533 Grossholzerstrasse	Grossholzerstr.	Gro	9901	1	8910	0	Affoltern am Albis	2676563.755	1235982.715	1 #####	#####	
26	26	0	1E+08	1	10000533 Ringstrasse	Ringstr.	Rin	9901	1	8910	0	Affoltern am Albis	2676575.69	1236015.958	1 #####	#####	
27	27	0	1E+08	2	10000553 Ringstrasse	Ringstr.	Rin	9901	1	8910	0	Affoltern am Albis	2676586.556	1235981.867	1 #####	#####	
28	28	0	1E+08	3	10000553 Ringstrasse	Ringstr.	Rin	9901	1	8910	0	Affoltern am Albis	2676601.51	1236016.196	1 #####	#####	
29	29	0	1E+08	4	10000553 Ringstrasse	Ringstr.	Rin	9901	1	8910	0	Affoltern am Albis	2676615.922	1235982.268	1 #####	#####	
30	30	0	1E+08	5	10000553 Ringstrasse	Ringstr.	Rin	9901	1	8910	0	Affoltern am Albis	2676615.418	1236016.226	1 #####	#####	
31	31	0	1E+08	6	10000553 Ringstrasse	Ringstr.	Rin	9901	1	8910	0	Affoltern am Albis	2676629.833	1235982.429	1 #####	#####	
32	32	0	1F+08	7	10000553 Ringstrasse	Ringstr.	Rin	9901	1	8910	0	Affoltern am Albis	2676648.446	1236016.613	1 #####	#####	

<https://www.housing-stat.ch/fr/index.html>

# RegBL – Registre Fédéral des bâtiments et des logements

- On essaye de trouver la correspondante entre une adresse renseignée et le RegBL. Si on trouve la même adresse, on récupère les coordonnées X,Y de l'adresse dans le RegBL.
- Pour minimiser le temps de calcul, on fait du parallel processing:



- Si on ne trouve pas l'adresse dans le registre des entreprises → fuzzy matching

Distance de Levenshtein :

C'est l'une des techniques les plus courantes pour mesurer la différence entre deux chaînes de caractères. Elle calcule le nombre minimum d'opérations nécessaires (ajouts, suppressions ou substitutions de caractères) pour transformer une chaîne en une autre.

[https://fr.wikipedia.org/wiki/Distance\\_de\\_Levenshtein](https://fr.wikipedia.org/wiki/Distance_de_Levenshtein)

# RegBL - fuzzy matching

- Si on ne trouve pas l'adresse dans le registre des entreprises → fuzzy matching

123 Main Street, Unit 4, Scranton, A1B 2C3	1.00	
123 <b>Unit 4</b> , Main Street, Scranton, A1B 2C3	0.92	
123 <b>Mian</b> Street, Unit 4, Scranton, A1B 2C <b>4</b>	0.88	
123 Main <b>St</b> , Unit <b>5</b> , Scranton, A1B 2C3	0.24	
123 Main Street, <b>Apt</b> 4, Scranton, A1B 2C3	0.84	
<b>134</b> Main <b>St</b> , Scranton, A1B 2C3	0.16	
<b>132</b> Main <b>St</b> , Scranton, A1B 2D <b>3</b>	0.14	
123 Main Street, Scranton, A1B 2C3	0.72	

<https://www.placekey.io/blog/fuzzy-logic-address-matching>

# RegBL - fuzzy matching

```
from fuzzywuzzy import process

adresse_avec_faute = "Avenue de la Gère, 1003 Lausanne"

liste_adresses = [
    "Avenue de la Gare, 1003 Lausanne",
    "Route de Berne, 1010 Lausanne",
    "Université de Lausanne, 1015 Lausanne",
    "Rue de Lausanne, 1000 Lausanne",
    "Chemin des Falaises, 1018 Lausanne"
]

resultats = process.extract(
    adresse_avec_faute,
    liste_adresses,
    limit=len(liste_adresses)
)

for match, score in resultats:
    print(f"{match} , Score : {score}")
```



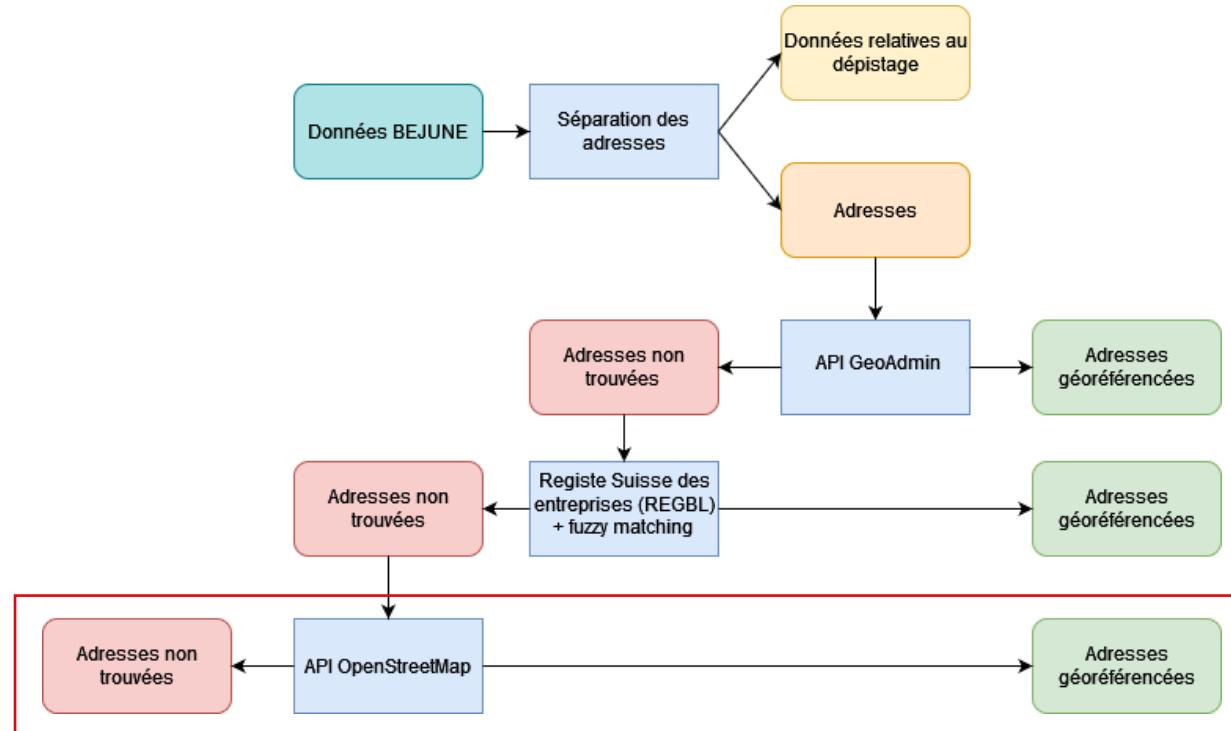
```
Avenue de la Gare, 1003 Lausanne , Score : 97
Rue de Lausanne, 1000 Lausanne , Score : 74
Route de Berne, 1010 Lausanne , Score : 72
Chemin des Falaises, 1018 Lausanne , Score : 67
Université de Lausanne, 1015 Lausanne , Score : 65
```

- On sélectionne l'adresse avec le meilleur score au dessus d'un certain seuil

<https://pypi.org/project/fuzzywuzzy/>

# Le géocodage

- Comment s'y prendre en suisse ? (Exemple BEJUNE)



- Base de données géoréférencées open source.
- Projet collaboratif de cartographie libre et ouvert
- License ODbL (**Open Data Commons Open Database License**) : Données modifiables par tout un chacun et réutilisables.

API nominatim :

- Géocodage direct (adresse → coordonnées) et inverse (coordonnées → adresse)
- Endpoints clés : /search, /reverse
- Formats de sortie : JSON, XML, GeoJSON...

```
def search_OSM(adresse):
    WGS84          = pyproj.CRS("EPSG:4326")
    CH1903_LV95   = pyproj.CRS("EPSG:2056")

    base_url = 'https://nominatim.openstreetmap.org/search'
    params   = {'q': adresse, 'format': 'json'}
    headers  = {
        'User-Agent': 'MonScriptGeo/1.0'
    }

    response = None
    max_attempts = 3
    for attempt in range(1, max_attempts+1):
        try:
            resp = requests.get(base_url, params=params, headers=headers, timeout=5)
            if resp.status_code == 200:
                response = resp.json()
                break
            else:
                time.sleep(1)
        except requests.RequestException as e:
            print(f"Erreur réseau (tentative {attempt}) : {e}")
            time.sleep(1)
        else:
            print(f"Echec après {max_attempts} tentatives pour '{adresse}'")
            return

    if not response:
        print(f"Aucune donnée renvoyée pour '{adresse}'")
        return
    if isinstance(response, list) and not response:
        print(f"Aucun résultat géographique pour '{adresse}'")
        return

    lat = float(response[0]["lat"])
    lon = float(response[0]["lon"])
    transformer = Transformer.from_crs(WGS84, CH1903_LV95, always_xy=True)
    x, y = transformer.transform(lon, lat)

    print(f"Coordonnées pour '{adresse}':\n    x = {x}\n    y = {y}\n")
```

OSM\_api.py