

Example ONE - Geocode and enrich using APIs

Introduction

In this example we will geocode a bunch of addresses and calculate some distances from a fixed POI.

Import the data

- Import the file named addresses.csv

Geocoding with nominatim

We will use the API proposed by [Nominatim](#), a free geolocation service based on OpenStreetMap.

Be nice with APIs, the limit is one request per second.

Create a new column called *request* using the formula :

```
"https://nominatim.openstreetmap.org/search.php?q="+value.replace(' ', '+')+"&format=jsonv2"
```

Please note that we replace the space character by a plus sign at the same time.

Create a column based on this column, using the function FETCH URL

Important Throttle delay > 1500ms to respect the API.

Add column by fetching URLs based on column request

New column name
Throttle delay milliseconds

On error ☒ set to blank ☐ store error
☒ Cache responses

HTTP headers to be used when fetching URLs: [Show](#)

Formulate the URLs to fetch:

Expression
Language General Refine Expression Language (GREL) ▼

No syntax error.

Preview
History
Starred
Help

row	value	value
1.	https://nominatim.openstreetmap.org/search.php?q=15+rue+de+la+villette+75019+Paris+France&format=jsonv2	https://nominatim.openstreetmap.org/search.php?q=15+rue+de+la+villette+75019+Paris+France&format=jsonv2
2.	https://nominatim.openstreetmap.org/search.php?q=Carrer+de+Balmes+5+Manresa+Spain&format=jsonv2	https://nominatim.openstreetmap.org/search.php?q=Carrer+de+Balmes+5+Manresa+Spain&format=jsonv2
3.	https://nominatim.openstreetmap.org/search.php?q=Konrad-Broßwitz-Straße+41,+60487+Frankfurt+am+Main,+Germany&format=jsonv2	https://nominatim.openstreetmap.org/search.php?q=Konrad-Broßwitz-Straße+41,+60487+Frankfurt+am+Main,+Germany&format=jsonv2

OK
Cancel

You will receive the geolocation data as JSON format.

Create a new column *lat* base ont his result using the following GREL command :

```
value.parseJson()[0].lat
```

Do the same for a column *lon* using this formula :

```
value.parseJson()[0].lat
```

Remove the useless columns. ou have now three columns : address, lat, lon

Part 2 - Calculate distances

Our goal is to calculate the distances from the addresses in Spain France and Germany, to the Campus in Mechelen. We will use this [OSMR](#) project API but first, we need to restructure the project.

Preparing the dataset

The Mechelen Address should be in the first column as it's our point of reference.

- Star row 4 (Mechelen) and facet your project by star. Row 4 should temporarily remain alone in your project.

OpenRefine

addresses csv

Permalink

Facet / Filter

Undo / Redo 8 / 15

1 matching records (4 total)

Refresh

Reset all

Remove all

Starred Rows

change invert reset

2 choices

Sort by: name count

false 3

true 1

Facet by choice counts

exclude

▼ All

▼ Address

▼ Lon

▼ Lat

4. Zandpoortvest 60, 2800 Mechelen, Belgique 4.4877917 51.0232578

- Add a column called campus where the value is "campus".

OpenRefine

addresses csv

Permalink

Facet / Filter

Undo / Redo 8 / 9

1 matching records (4 total)

Refresh

Reset all

Remove all

Starred Rows

change invert reset

2 choices

Sort by: name count

false 3

true 1

Facet by choice counts

exclude

▼ All

▼ Address

4. Zandpoortvest 60, 2800 Mechelen, Belgique

Add column based on column Address

New column name

Campus

On error

☒ set to blank

☐ store error

☐ copy value from original column

Expression

Language General Refine Expression Language (GREL)

"Campus"

No syntax error.

Preview

History

Starred

Help

row value

4. Zandpoortvest 60, 2800 Mechelen, Belgique

"Campus"

Campus

OK

Cancel

- Based on this column, join the other columns.

OpenRefine

addresses csv

Permalink

Undo / Redo 10 / 10

1 matching records (4 total)

Reset all

Remove all

▼ All

▼ Campus

▼ Address

4. Campus Zandpoortvest 60, 2800 Mechelen, Belgique

Join columns

Select and order columns to join

☐ Campus

☒ Address

☒ Lon

☒ Lat

Select all

Deselect all

Select options

Separator between the content of each column

,

Replace nulls with

Write result in selected column.

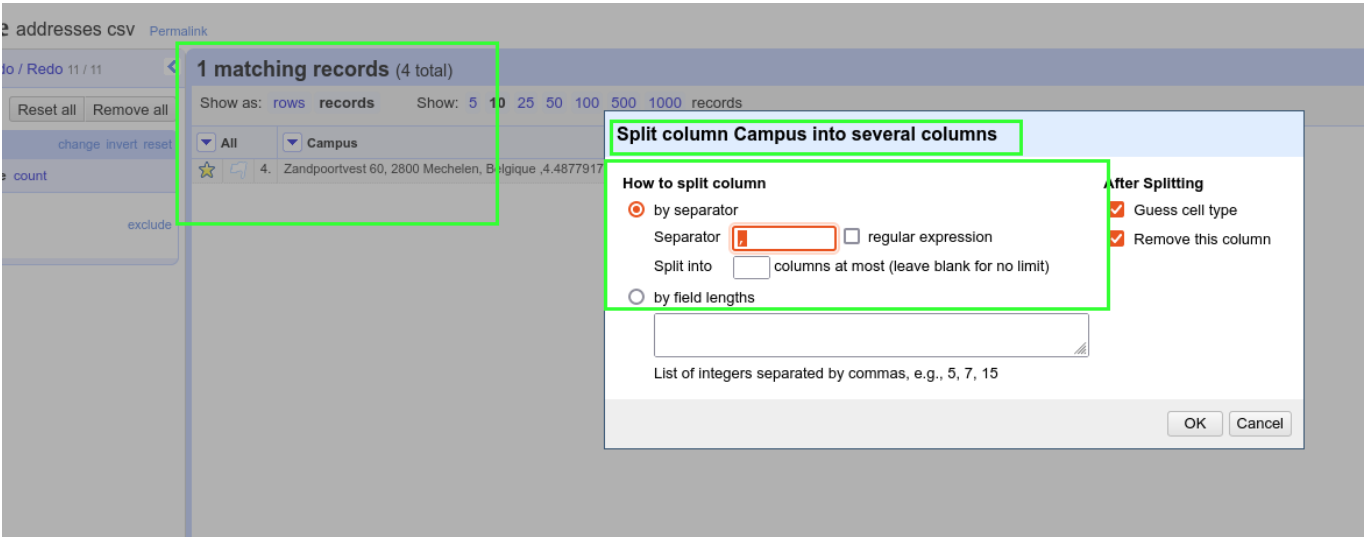
Write result in new column named

Delete joined columns.

OK

Cancel

- Split this column using the comma separator



- sorting the column Campus1 by text should give you this result :

Permalink

1 records

Show as: rows records Show: 5 10 25 50 100 500 1000 records

All	Campus 1	Campus 2	Campus 3	Campus 4	Campus 5	Address	Lon	Lat
1.	Zandpoortvest 60	2800 Mechelen	Belgique	4.4877917	51.0232578	Zandpoortvest 60, 2800 Mechelen, Belgique	4.4877917	51.0232578
						15 rue de la villette 75019 Paris France	2.3873563	48.8755045
						Carrer de Balmes 5 Manresa Spain	1.8308085	41.7345759
						Konrad-Broßwitz-Straße 41, 60487 Frankfurt am Main, Germany	8.641342689009205	50.1264852

We can now remove column Campus 1 and Campus 2.

We rename Campus 3 as "Campus", Campus 4 as "Lon1" and Campus 5 as "Lat1".

Permalink

1 records

Show as: rows records Show: 5 10 25 50 100 500 1000 records

All	Campus	Lon1	Lat1	Address	Lon	Lat
1.	Belgique	4.4877917	51.0232578	Zandpoortvest 60, 2800 Mechelen, Belgique	4.4877917	51.0232578
				15 rue de la villette 75019 Paris France	2.3873563	48.8755045
				Carrer de Balmes 5 Manresa Spain	1.8308085	41.7345759
				Konrad-Broßwitz-Straße 41, 60487 Frankfurt am Main, Germany	8.641342689009205	50.1264852

We now use the fill fonction on Campus, Lat1 and Lat2.

rmalink

< 4 records

Show as: **rows** records Show: 5 10 25 50 100 500 1000 records

	All	Campus	Lon1	Lat1	Address	Lon	Lat
★	1.	Belgique	4.4877917	51.0232578	Zandpoortvest 60, 2800 Mechelen, Belgique	4.4877917	51.0232578
☆	2.	Belgique	4.4877917	51.0232578	15 rue de la villette 75019 Paris France	2.3873563	48.8755045
☆	3.	Belgique	4.4877917	51.0232578	Carrer de Balmes 5 Manresa Spain	1.8308085	41.7345759
☆	4.	Belgique	4.4877917	51.0232578	Konrad-Broßwitz-Straße 41, 60487 Frankfurt am Main, Germany	8.641342689009205	50.1264852

And we can remove the useless first row (starred).

rmalink

< 3 records

Show as: **rows** records Show: 5 10 25 50 100 500 1000 records

	All	Campus	Lon1	Lat1	Address	Lon	Lat
☆	1.	Belgique	4.4877917	51.0232578	15 rue de la villette 75019 Paris France	2.3873563	48.8755045
☆	2.	Belgique	4.4877917	51.0232578	Carrer de Balmes 5 Manresa Spain	1.8308085	41.7345759
☆	3.	Belgique	4.4877917	51.0232578	Konrad-Broßwitz-Straße 41, 60487 Frankfurt am Main, Germany	8.641342689009205	50.1264852

Calculate distances

We use the [OSMR](#) project API.

We create an URL based on Lon1 column :

```
"http://router.project-
osrm.org/route/v1/driving/"+value+", "+cells['Lat1'].value+"; "+cells['Lon'].
value+", "+cells['Lat'].value+"?overview=false"
```

- Based on this new column, create a Fetch URL column __but be aware of the throttle delay of 5000ms minimum between two requests.
- Parse this column using this formula to get the distance in kilometers.

```
(value.parseJson().routes[0].distance).toNumber()/1000
```