**Python Geocoder**

**Simple and consistent geocoding library written in Python.**

<https://github.com/DenisCarriere/geocoder>

**Overview**

Many online providers such as Google & Bing have geocoding services, these providers do not include Python libraries and have different JSON responses between each other. It can be very difficult sometimes to parse a particular geocoding provider since each one of them have their own JSON schema. Here is a typical example of retrieving a Lat & Lng from Google using Python, things shouldn't be this hard.

>>> import requests

>>> url = 'https://maps.googleapis.com/maps/api/geocode/json'

>>> params = {'sensor': 'false', 'address': 'Mountain View, CA'}

>>> r = requests.get(url, params=params)

>>> results = r.json()['results']

>>> location = results[0]['geometry']['location']

>>> location['lat'], location['lng']

(37.3860517, -122.0838511)

Now lets use Geocoder to do the same task

>>> import geocoder

>>> g = geocoder.google('Mountain View, CA')

>>> g.latlng

(37.3860517, -122.0838511)

**A glimpse at the API**

Many properties are available once the geocoder object is created.

**Forward**

>>> import geocoder

>>> g = geocoder.google('Mountain View, CA')

>>> g.geojson

>>> g.json

>>> g.wkt

>>> g.osm

**Multiple queries ('batch' geocoding)**

>>> import geocoder

>>> g = geocoder.mapquest(['Mountain View, CA', 'Boulder, Co'], method='batch')

>>> for result in g:

... print(result.address, result.latlng)

...

('Mountain View', [37.39008, -122.08139])

('Boulder', [40.015831, -105.27927])

**Multiple results**

>>> import geocoder

>>> g = geocoder.geonames('Mountain View, CA', maxRows=5)

>>> print(len(g))

5

>>> for result in g:

... print(result.address, result.latlng)

...

Mountain View ['37.38605', '-122.08385']

Mountain View Elementary School ['34.0271', '-117.59116']

Best Western Plus Mountainview Inn and Suites ['51.79516', '-114.62793']

Best Western Mountainview Inn ['49.3338', '-123.1446']

Mountain View Post Office ['37.393', '-122.07774']

The providers currently supporting multiple results are listed in the table [below](https://github.com/DenisCarriere/geocoder/blob/master/README.md#providers).

**Reverse**

>>> g = geocoder.google([45.15, -75.14], method='reverse')

>>> g.city

>>> g.state

>>> g.state\_long

>>> g.country

>>> g.country\_long

**House Addresses**

>>> g = geocoder.google("453 Booth Street, Ottawa ON")

>>> g.housenumber

>>> g.postal

>>> g.street

>>> g.street\_long

**IP Addresses**

>>> g = geocoder.ip('199.7.157.0')

>>> g = geocoder.ip('me')

>>> g.latlng

>>> g.city

**Bounding Box**

Accessing the JSON & GeoJSON attributes will be different

>>> g = geocoder.google("Ottawa")

>>> g.bbox

{"northeast": [45.53453, -75.2465979], "southwest": [44.962733, -76.3539158]}

>>> g.geojson['bbox']

[-76.3539158, 44.962733, -75.2465979, 45.53453]

>>> g.southwest

[44.962733, -76.3539158]

**Command Line Interface**

$ geocode "Ottawa, ON" >> ottawa.geojson

$ geocode "Ottawa, ON" \

--provide google \

--out geojson \

--method geocode

**Providers**

[**Geocoder.ca**](http://geocoder.readthedocs.org/providers/Geocoder-ca.html)**(Geolytica)**

[**GeocodeFarm**](https://geocode.farm/)

[**GeoNames**](http://geocoder.readthedocs.org/providers/GeoNames.html)

[**GeoOttawa**](http://geocoder.readthedocs.org/providers/GeoOttawa.html)

[**Gisgraphy**](https://premium.gisgraphy.com/)

[**Google**](http://geocoder.readthedocs.org/providers/Google.html)

[**Komoot**](http://photon.komoot.de/)**(OSM powered)**

[**USCensus**](https://geocoding.geo.census.gov/geocoder/Geocoding_Services_API.html)

[**Yahoo**](http://geocoder.readthedocs.org/providers/Yahoo.html)

**MORE SOURCES ARE SUPPORTED VIA API, ETC.**

**Installation**

**PyPi Install**

To install Geocoder, simply:

$ pip install geocoder

...

**GitHub Install**

Installing the latest version from Github:

$ git clone https://github.com/DenisCarriere/geocoder

...

$ cd geocoder

$ python setup.py install

...

**Snap Install**

To install the stable geocoder [snap](https://snapcraft.io/) in any of the [supported Linux distros](https://snapcraft.io/docs/core/install):

$ sudo snap install geocoder

...

If you want to help testing the latest changes from the master branch, you can install it from the edge channel:

$ sudo snap install geocoder --edge

...

The installed snap will be updated automatically every time a new version is pushed to the store.

**Feedback**

Please feel free to give any feedback on this module. Speak up on T-witter  [@DenisCarriere](https://twitter.com/DenisCarriere) and

tell me how you use this Python Geocoder. New updates will be pushed to Twitter Hashtags [#python](https://twitter.com/search?q=%23python).

**Contribution**

If you find any bugs or any enhancements to recommend please send some of your comments/suggestions to the [Github Issues Page](https://github.com/DenisCarriere/geocoder/issues).

Some way to contribute, from the most generic to the most detailed:

**Documenting**

If you are not comfortable with development, you can still contribute with the documentation.

* review the documentation of a specific provider. Most of the time they are lacking details...
* review the parameters for a specific method, compared to what is supported by the provider
* review documentation for command line

If you miss any feature, just create an issue accordingly. Be sure to describe your use case clearly, and to provide links to the correct sources.

**Coding**

* add support for a new provider. *Documentation TBD*, starting point possible with [wip\_guide](https://geocoder.readthedocs.io/wip_guide.html).
* extend methods for an existing support, i.e support an additionnal API). *Documentation TBD*
* extend support of an existing API, i.e, support more (json) fields from the response, or more parameters. *Documentation TBD*

**ChangeLog**

See [CHANGELOG.md](https://github.com/DenisCarriere/geocoder/blob/master/CHANGELOG.md)