**Multi-Service Geocoder**

This Python script utilizes the [GeoPy geocoding library](https://github.com/geopy/geopy) to batch geocode a number of addresses, using various services until a pair of latitude/longitude values are returned. Python 3 port and refactor of a script by [@rgdonohue](https://gist.github.com/rgdonohue/c4beedd3ca47d29aef01).

<https://gist.github.com/ericmhuntley/0c293113aa75a254237c143e0cf962fa>

Built to anticipate an input csv should that includes columns named street, city, state, country.

**Usage Example**

python geocode.py data.csv 100

Where data.csv is an appropriately formatted csv encoded in utf-8 and 100 is the timout between each request in units of milliseconds.

[Raw](https://gist.github.com/ericrobskyhuntley/0c293113aa75a254237c143e0cf962fa/raw/dfaf321463fe4f23244d565997e548b98b344a5a/geocode.py)

[**geocode.py**](https://gist.github.com/ericrobskyhuntley/0c293113aa75a254237c143e0cf962fa#file-geocode-py)

# import the geocoding services you'd like to try

from geopy.geocoders import ArcGIS, Bing, Nominatim, OpenCage, GoogleV3, OpenMapQuest

import csv, sys

import pandas as pd

import keys

in\_file = str(sys.argv[1])

out\_file = str('gc\_' + in\_file)

timeout = int(sys.argv[2])

print('creating geocoding objects.')

arcgis = ArcGIS(timeout=timeout)

bing = Bing(api\_key=keys.bing\_key,timeout=100)

nominatim = Nominatim(user\_agent=keys.n\_user, timeout=timeout)

opencage = OpenCage(api\_key=keys.oc\_key,timeout=timeout)

googlev3 = GoogleV3(api\_key=keys.g3\_api, domain='maps.googleapis.com', timeout=timeout)

openmapquest = OpenMapQuest(api\_key=keys.omq\_api, timeout=timeout)

# choose and order your preference for geocoders here

geocoders = [openmapquest, nominatim, opencage, googlev3, arcgis]

def gc(address):

street = str(address['street'])

city = str(address['city'])

state = str(address['state'])

country = str(address['country'])

add\_concat = street + ", " + city + ", " + state + " " + country

for gcoder in geocoders:

location = gcoder.geocode(add\_concat)

if location != None:

print(f'geocoded record {address.name}: {street}')

located = pd.Series({

'lat': location.latitude,

'lng': location.longitude,

'time': pd.to\_datetime('now')

})

else:

print(f'failed to geolocate record {address.name}: {street}')

located = pd.Series({

'lat': 'null',

'lng': 'null',

'time': pd.to\_datetime('now')

})

return located

print('opening input.')

reader = pd.read\_csv(in\_file, header=0)

print('geocoding addresses.')

reader = reader.merge(reader.apply(lambda add: gc(add), axis=1), left\_index=True, right\_index=True)

print(f'writing to {out\_file}.')

reader.to\_csv(out\_file, encoding='utf-8', index=False)

print('done.')

[**keys.py**](https://gist.github.com/ericrobskyhuntley/0c293113aa75a254237c143e0cf962fa#file-keys-py)

# User Agent identification (e.g., email address) for Nominatim

# (Querying without user agent is against ToS)

# n\_user = ''

# Bing API key

# bing\_key = ''

# OpenCage API key

# oc\_key = ''

# GoogleV3 API key

# g3\_key = ''

# OpenMapQuest API key

# omq\_key =''

[**jazon33y**](https://gist.github.com/jazon33y)**commented**[**on Sep 20, 2018**](https://gist.github.com/ericrobskyhuntley/0c293113aa75a254237c143e0cf962fa#gistcomment-2712275)

|  |
| --- |
| Looks great, and thank you for posting this! Are these all of the free geocoding APIs? Just curious as to the choice of geocoding services.  Cheers, |

OwnerAuthor

[**ericrobskyhuntley**](https://gist.github.com/ericrobskyhuntley)**commented**[**on Oct 2, 2018**](https://gist.github.com/ericrobskyhuntley/0c293113aa75a254237c143e0cf962fa#gistcomment-2722291)**•**

**edited**