**Exercise 1: Change either** [**geojson.py**](https://www.py4e.com/code3/geojson.py) **or** [**geoxml.py**](https://www.py4e.com/code3/geoxml.py) **to print out the two-character country code from the retrieved data. Add error checking so your program does not traceback if the country code is not there. Once you have it working, search for “Atlantic Ocean” and make sure it can handle locations that are not in any country.**

Code: <http://www.py4e.com/code3/geojson.py>

import urllib.request, urllib.parse, urllib.error

import json

import ssl

api\_key = False

# If you have a Google Places API key, enter it here

# api\_key = 'AIzaSy\_\_\_IDByT70'

# https://developers.google.com/maps/documentation/geocoding/intro

if api\_key is False:

api\_key = 42

serviceurl = 'http://py4e-data.dr-chuck.net/json?'

else :

serviceurl = 'https://maps.googleapis.com/maps/api/geocode/json?'

# Ignore SSL certificate errors

ctx = ssl.create\_default\_context()

ctx.check\_hostname = False

ctx.verify\_mode = ssl.CERT\_NONE

while True:

address = input('Enter location: ')

if len(address) < 1: break

parms = dict()

parms['address'] = address

if api\_key is not False: parms['key'] = api\_key

url = serviceurl + urllib.parse.urlencode(parms)

print('Retrieving', url)

uh = urllib.request.urlopen(url, context=ctx)

data = uh.read().decode()

print('Retrieved', len(data), 'characters')

try:

js = json.loads(data)

except:

js = None

if not js or 'status' not in js or js['status'] != 'OK':

print('==== Failure To Retrieve ====')

print(data)

continue

print(json.dumps(js, indent=4))

lat = js['results'][0]['geometry']['location']['lat']

lng = js['results'][0]['geometry']['location']['lng']

print('lat', lat, 'lng', lng)

location = js['results'][0]['formatted\_address']

print(location)