Is it Essential? v0.1

May 4, 2020

[ ]:
```
##this program will retrieve whether the business is open, whether it is␣
 ↪considered essential or not, and (stretch goal what the
##new hours are for the place, if any

#1. Prompt to input a business
#2. run input in all three functions to look up business and industry
#3. search google to see whether the business is open or not, hours
#4. check whether business is essential from txt list
#5 print hours, essential, *optional* nearby stores
```

[1]:
```
!pip install requests
import json
import requests
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

```
Requirement already satisfied: requests in /opt/conda/lib/python3.7/site-
packages (2.12.4)
```

[45]:
```
#all functions in this program, defined
def naic_lookup(code):
    yr = 2012
    url = f"http://naics.codeforamerica.org/v0/q?year={yr}&code={code}"
    response = requests.get(url)
    data = response.json()
    return data
def places_lookup(place):
    key = "AIzaSyCIZ5647df1-qq4AVq5Jmt9uhLJWSLkmeo"
    params = { "key" : f"{key}", "input" : f"{place}", "inputtype" :␣
 ↪"textquery", "language" : "english", "locationbias" : "ipbias"}
    url = f"https://maps.googleapis.com/maps/api/place/findplacefromtext/json?"
    response = requests.get(url, params = params)
    candidates = response.json()
    values = (candidates['candidates'])
    IDlist = []
    for placeid in values:
```

```
        IDlist.append(placeid['place_id'])
    return IDlist
def places_details(place_id):
    key = "AIzaSyCIZ5647df1-qq4AVq5Jmt9uhLJWSLkmeo"
    url = f"https://maps.googleapis.com/maps/api/place/details/json?
 ↪key={key}&place_id={place_id}&fields=name,business_status,formatted_address,opening_hours"
    response = requests.get(url)
    results = response.json()
    print(results)
```

[46]:
```
place = "Target"
search = places_lookup(place)
for ID in search:
    places_details(ID)
```

```
{'html_attributions': [], 'result': {'business_status': 'OPERATIONAL',
'formatted_address': '9010 N 121st E Ave, Owasso, OK 74055, USA', 'name':
'Target', 'opening_hours': {'open_now': True, 'periods': [{'close': {'day': 0,
'time': '2100'}, 'open': {'day': 0, 'time': '0800'}}, {'close': {'day': 1,
'time': '2100'}, 'open': {'day': 1, 'time': '0800'}}, {'close': {'day': 2,
'time': '2100'}, 'open': {'day': 2, 'time': '0800'}}, {'close': {'day': 3,
'time': '2100'}, 'open': {'day': 3, 'time': '0800'}}, {'close': {'day': 4,
'time': '2100'}, 'open': {'day': 4, 'time': '0800'}}, {'close': {'day': 5,
'time': '2100'}, 'open': {'day': 5, 'time': '0800'}}, {'close': {'day': 6,
'time': '2100'}, 'open': {'day': 6, 'time': '0800'}}], 'weekday_text': ['Monday:
8:00 AM - 9:00 PM', 'Tuesday: 8:00 AM - 9:00 PM', 'Wednesday: 8:00 AM - 9:00
PM', 'Thursday: 8:00 AM - 9:00 PM', 'Friday: 8:00 AM - 9:00 PM', 'Saturday: 8:00
AM - 9:00 PM', 'Sunday: 8:00 AM - 9:00 PM']}}, 'status': 'OK'}
```

[43]:
```
#here's places details working
key = "AIzaSyCIZ5647df1-qq4AVq5Jmt9uhLJWSLkmeo"
place_id = "ChIJhdk_kJzwtocROIkZJkq5o9k"
url = f"https://maps.googleapis.com/maps/api/place/details/json?
 ↪key={key}&place_id={place_id}&fields=name,business_status,formatted_address,opening_hours"
response = requests.get(url)
response.json()
```

[43]:
```
{'html_attributions': [],
 'result': {'business_status': 'OPERATIONAL',
  'formatted_address': '9010 N 121st E Ave, Owasso, OK 74055, USA',
  'name': 'Target',
  'opening_hours': {'open_now': True,
   'periods': [{'close': {'day': 0, 'time': '2100'},
     'open': {'day': 0, 'time': '0800'}},
    {'close': {'day': 1, 'time': '2100'}, 'open': {'day': 1, 'time': '0800'}},
    {'close': {'day': 2, 'time': '2100'}, 'open': {'day': 2, 'time': '0800'}},
    {'close': {'day': 3, 'time': '2100'}, 'open': {'day': 3, 'time': '0800'}},
```

```
    {'close': {'day': 4, 'time': '2100'}, 'open': {'day': 4, 'time': '0800'}},
    {'close': {'day': 5, 'time': '2100'}, 'open': {'day': 5, 'time': '0800'}},
    {'close': {'day': 6, 'time': '2100'}, 'open': {'day': 6, 'time': '0800'}}],
   'weekday_text': ['Monday: 8:00 AM - 9:00 PM',
    'Tuesday: 8:00 AM - 9:00 PM',
    'Wednesday: 8:00 AM - 9:00 PM',
    'Thursday: 8:00 AM - 9:00 PM',
    'Friday: 8:00 AM - 9:00 PM',
    'Saturday: 8:00 AM - 9:00 PM',
    'Sunday: 8:00 AM - 9:00 PM']}},
 'status': 'OK'}
```

[15]:
```
key = "AIzaSyCIZ5647df1-qq4AVq5Jmt9uhLJWSLkmeo"
place = input()
params = { "key" : f"{key}", "input" : f"{place}", "inputtype" : "textquery",␣
 ↪"language" : "english", "locationbias" : "ipbias"}
url = f"https://maps.googleapis.com/maps/api/place/findplacefromtext/json?"
response = requests.get(url, params = params)
candidates = response.json()
values = (candidates['candidates'])
print(values)
```

```
Target
[{'place_id': 'ChIJhdk_kJzwtocROIkZJkq5o9k'}]
```

[18]:
```
#this code returns a list of place ids necessary for google's find place API
key = "AIzaSyCIZ5647df1-qq4AVq5Jmt9uhLJWSLkmeo"
place = input()
params = { "key" : f"{key}", "input" : f"{place}", "inputtype" : "textquery",␣
 ↪"language" : "english", "locationbias" : "ipbias"}
url = f"https://maps.googleapis.com/maps/api/place/findplacefromtext/json?"
response = requests.get(url, params = params)
candidates = response.json()
values = (candidates['candidates'])
for placeid in values:
    print(placeid['place_id'])
#for value in candidates.values():
#    print(value)
#[candidates['place_id'] for c in candidates]
```

```
Walmart
ChIJXx0ERDqBt4cR-F2vQWTbmn0
```

[19]:
```
key = "AIzaSyCIZ5647df1-qq4AVq5Jmt9uhLJWSLkmeo"
place = input()
params = { "key" : f"{key}", "input" : f"{place}", "inputtype" : "textquery",␣
 ↪"language" : "english", "locationbias" : "ipbias"}
```

```
url = f"https://maps.googleapis.com/maps/api/place/findplacefromtext/json?"
response = requests.get(url, params = params)
candidates = response.json()
values = (candidates['candidates'])
for placeid in values:
    print(placeid['place_id'])
```

```
CVS
ChIJWygFH5aBt4cR5UjrSv6z00o
```

[2]:
```
key = "AIzaSyCIZ5647df1-qq4AVq5Jmt9uhLJWSLkmeo"
place = input()
params = { "key" : f"{key}", "input" : f"{place}", "inputtype" : "textquery",␣
 ↪"language" : "english", "locationbias" : "ipbias"}
url = f"https://maps.googleapis.com/maps/api/place/findplacefromtext/json?"
response = requests.get(url, params = params)
candidates = response.json()
values = (candidates['candidates'])
IDlist = []
for placeid in values:
    IDlist.append(placeid['place_id'])
```

```
Target
['ChIJhdk_kJzwtocROIkZJkq5o9k']
```

[31]:
```
try:
    search = input("Enter a store, a keyword, or a NAICS code. ")
    data = naic_lookup(search)
    print(data)
except:
    print("error!")
```

```
Enter a store, a keyword, or a NAICS code. 44111
{'seq_no': 1098, 'description_code': '441110', 'code': 44111, 'description':
['See industry description for 441110.'], 'title': 'New Car Dealers'}
```

[19]:
```
def keyword_lookup():
```

```
Enter a store, a keyword, or a NAICS code. 44111
```

[19]:
```
{'seq_no': 1098,
 'description_code': '441110',
 'code': 44111,
 'description': ['See industry description for 441110.'],
 'title': 'New Car Dealers'}
```

[29]:
```
search = input("Enter a store, a keyword, or a NAICS code. ")
naic_lookup(search)
```

```
Enter a store, a keyword, or a NAICS code. 44111
```

[29]:
```
{'seq_no': 1098,
 'description_code': '441110',
 'code': 44111,
 'description': ['See industry description for 441110.'],
 'title': 'New Car Dealers'}
```

[ ]: