

Google textsearch API

Documentation:

<https://developers.google.com/places/web-service/search#TextSearchRequests>

Google's text search API returns a list of places based on a free form string query. For e.g., if you want to find restaurants close to Columbia University, you can send a request using the string "restaurants near Columbia University" and the API will return a response containing an appropriate list of restaurants.

The assignment

Write a function:

```
get_businesses(query,api_key,min_rating=0.0,max_results = 30)
```

That uses the textsearch API to get a list of businesses defined by the query. If the *min_rating* parameter is included in the function call, then your list should only include businesses whose rating is greater than the value specified. If the *max_results* parameter is included in the query, your list should include a maximum of *max_results*.

Format of the list returned by your function

Your function should return a **list** of businesses that satisfy your query. The data for each business should be in the form of **tuple** containing the following information:

1. The **name** of the business
2. The **formatted address** of the business
3. Whether the business is open at the time of the query (use a dictionary for this field)
4. The price level (google returns numbers **form 0 to 4** for this field)
5. The rating of the business

For example, the first two responses for the query:

```
get_businesses(query,api_key,min_rating=4.0,max_results=50)
```

are:

Example Output:

```
[("Brownie's Cafe",  
  '1172 Amsterdam Ave, New York, NY 10027, USA',  
  {'open_now': True},  
  None,  
  4.5),  
 ("Friedman's",  
  '1187 Amsterdam Ave, New York, NY 10027, USA',  
  {'open_now': True},  
  2,  
  4.2)]
```

Pagination

The API returns 20 results for each call. If there are more than 20 results in a response, then the API response includes a key “next_page_token” in the json (or xml) response. If you need more than 20 responses, then you must use this token to get the next 20 results (and so on until either you have enough responses or there are no more results from the query). Use the next_page_token as a parameter to the url (see the bottom of the documentation page for an example) but note the following:

1. You should include only the pagetoken and key parameters when getting the next page (omit the query parameter)
2. The next page may not be immediately available and the API will return an INVALID_REQUEST status if the page is not available. Use a loop to continually send the query and check the response until the page is available. Because each request is (potentially) charged, you might want to wait a little before sending the request. The following code fragment should help:

```
response = requests.get(new_url)  
while response.json().get('status') != 'OK':  
    from time import sleep  
    from random import random  
    sleep(random())  
    response = requests.get(new_url)
```

Where *new_url* represents the url request containing the pagetoken parameter

Please include your api key in your submission. The TA will need it to check if your code works properly!