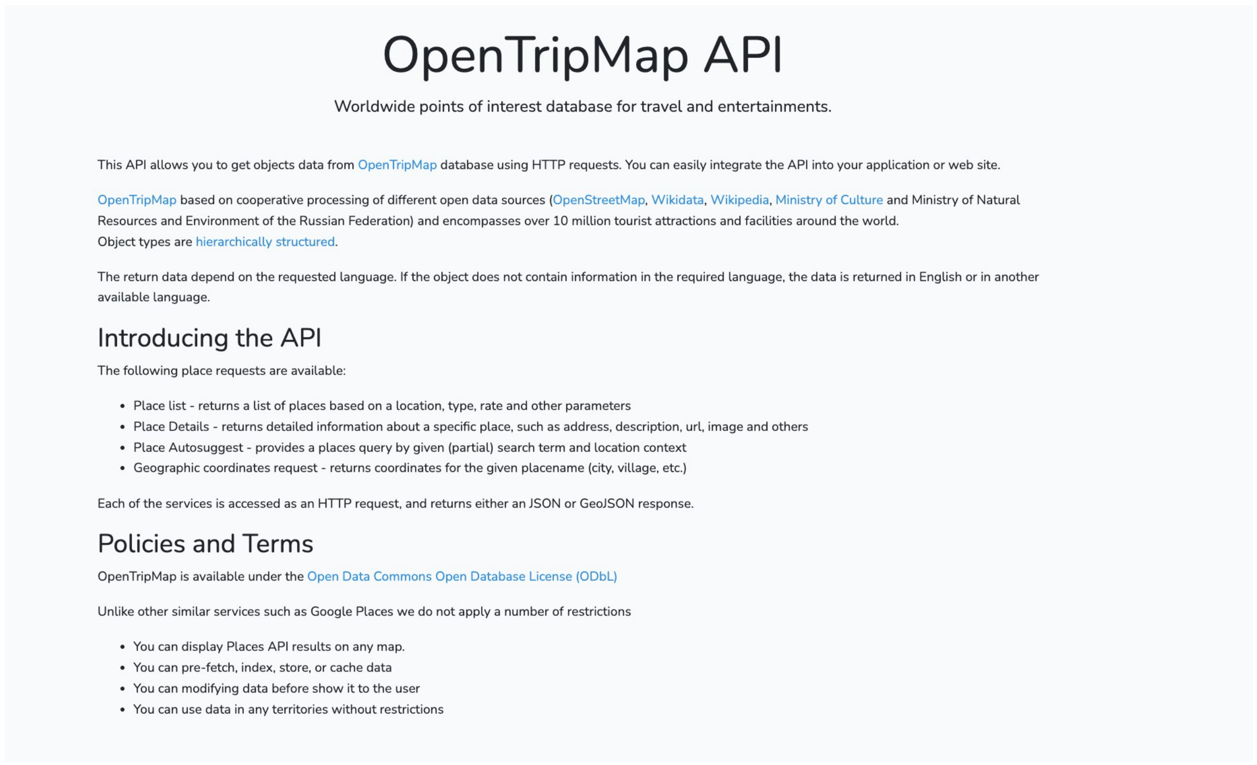
Assignment 3: Data Access

GGE 6505

DANIEL GANTA	**3705839**
CHINYERE OBI	**3724449**
HATESIN ATUGWU	**3734234**

1. We have used the **OpenTripMap** API



OpenTripMap API

Worldwide points of interest database for travel and entertainments.

This API allows you to get objects data from OpenTripMap database using HTTP requests. You can easily integrate the API into your application or web site.

OpenTripMap based on cooperative processing of different open data sources (OpenStreetMap, Wikidata, Wikipedia, Ministry of Culture and Ministry of Natural Resources and Environment of the Russian Federation) and encompasses over 10 million tourist attractions and facilities around the world.
Object types are hierarchically structured.

The return data depend on the requested language. If the object does not contain information in the required language, the data is returned in English or in another available language.

Introducing the API

The following place requests are available:

- Place list - returns a list of places based on a location, type, rate and other parameters
- Place Details - returns detailed information about a specific place, such as address, description, url, image and others
- Place Autosuggest - provides a places query by given (partial) search term and location context
- Geographic coordinates request - returns coordinates for the given placename (city, village, etc.)

Each of the services is accessed as an HTTP request, and returns either an JSON or GeoJSON response.

Policies and Terms

OpenTripMap is available under the Open Data Commons Open Database License (ODbL)

Unlike other similar services such as Google Places we do not apply a number of restrictions

- You can display Places API results on any map.
- You can pre-fetch, index, store, or cache data
- You can modifying data before show it to the user
- You can use data in any territories without restrictions

We chose this API because of the following reasons:

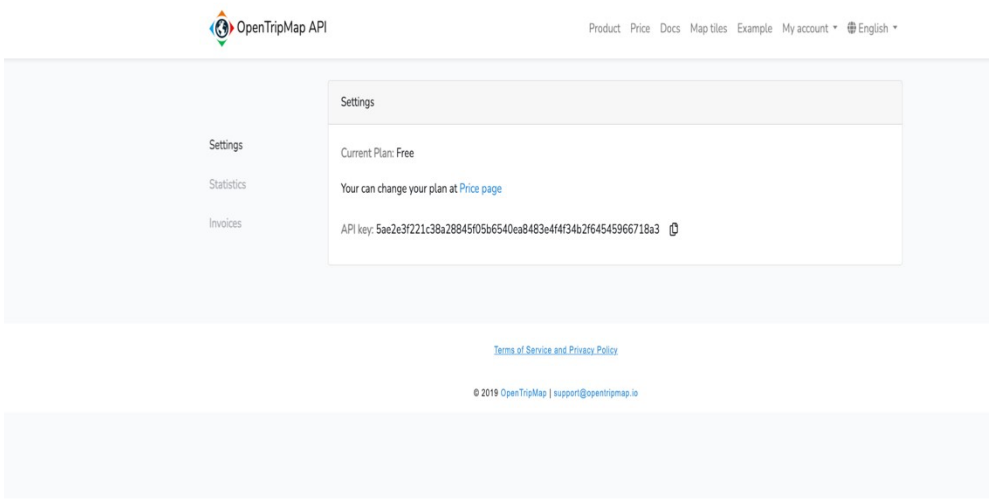
- The API has data peculiar to the sector we are working on, which is the travel and tourism sector.
- The API is open source.
- The API version we have access to is free, as we intend to access only free API for our training period.
- The API documentation is not cumbersome, it is quite straightforward to understand.
- The API functionalities we need for our project can be accessed from this API

We have 4 API functionalities

- Geoname - The data downloaded from the API shows information about locations, for example, the country, city, region, its coordinates, population, time zone, etc. GeoName function returns geographic coordinates for the given placename (region, city, village, etc.). The method returns the place whose name is most similar to the search string service based on GeoName's database.

- Bbox - Method returns all objects (or a number of objects) in the given bounding box optionally filtered by parameters. Only basic information is included in the response: xid, name, kinds, osm, wikidata and geometry of each object. Depending on the chosen format, the response is either a simple array of objects (with a smaller volume) or an object in GeoJSON format.
- Radius (We have used this method) – This method returns objects closest to the selected point optionally filtered by parameters. Only basic information is included in the response: xid, name, kinds, osm, wikidata, and geometry of each object. Depending on the chosen format, the response is either a simple array of objects (with a smaller volume) or an object in GeoJSON format.
- AutoSuggest - This method returns suggestions for the search term closest to the selected point optionally filtered by parameters. Only basic information is included in the response: xid, name, kinds, osm, and wikidata of each object. Depending on the chosen format, the response is either a simple array of objects (with a smaller volume) or an object in GeoJSON format.
- Xid - Returns detailed information about the object. Objects can contain different amounts of information.

2. API key was generated once we created an account.
   Below is a screenshot of the API key that was generated after signing in.



Current Plan: Free

Your can change your plan at Price page

API key: 5ae2e3f221c38a28845f05b6540ea8483e4f4f34b2f64545966718a3

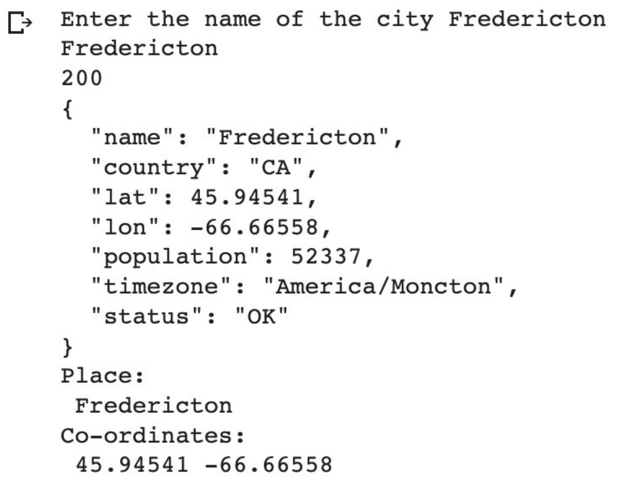
Terms of Service and Privacy Policy

© 2019 OpenTripMap | support@opentripmap.io

3. Code to access data: Please go to this link: https://colab.research.google.com/drive/1IcbOY8i47WF2q7iJsyC45wDfBm_257r?usp=sharing

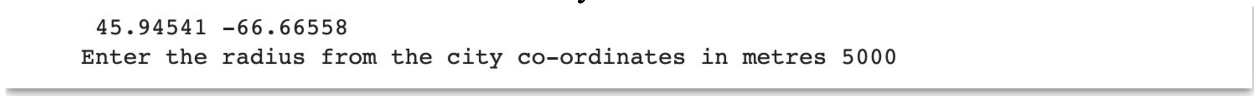
4. Json File link: sample.json

In our program, we input a city of datatype string, which retrieves the details like country, co-ordinates, population etc.



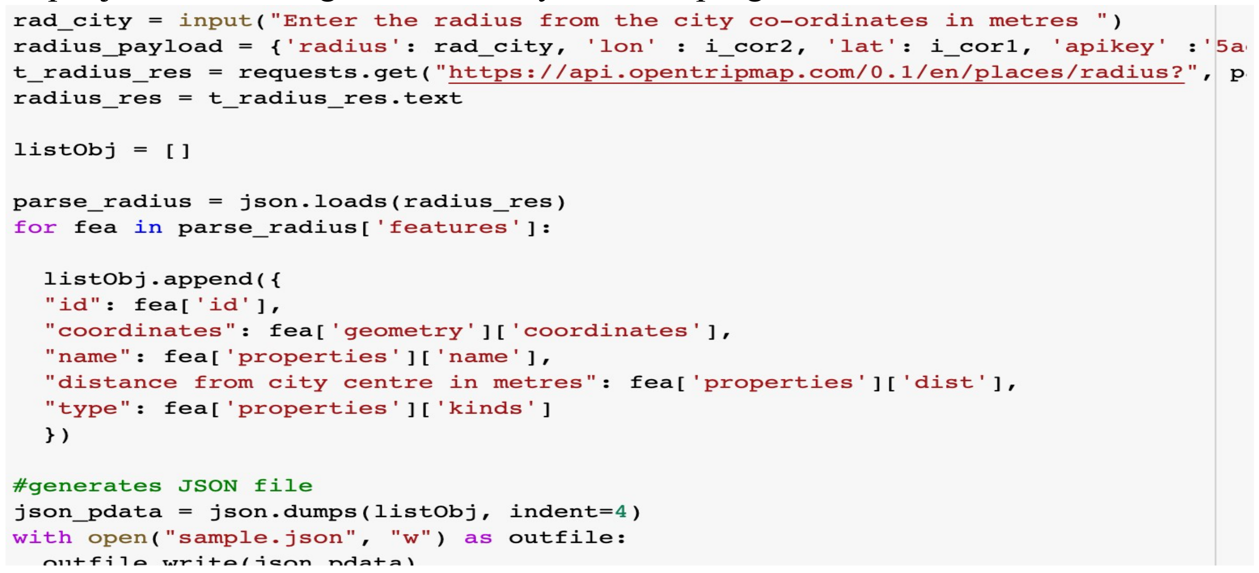
```
Enter the name of the city Fredericton
Fredericton
200
{
  "name": "Fredericton",
  "country": "CA",
  "lat": 45.94541,
  "lon": -66.66558,
  "population": 52337,
  "timezone": "America/Moncton",
  "status": "OK"
}
Place:
 Fredericton
Co-ordinates:
 45.94541 -66.66558
```

Then we ask for the radius from the city to check.



```
 45.94541 -66.66558
Enter the radius from the city co-ordinates in metres 5000
```

Since we already have the co-ordinates of the city from the first API call, we use these as parameters along with the radius input to retrieve all the interesting sites within this radius using radius GET method.
Now, using json loads, we can access the data collected in this call. Since the radius API GET method has retrieved a lot of data, we have filtered the data to include only columns like id, co-ordinates, name of place, distance from the city and type of attraction and written them to a new json file named sample.json, which is generated every time the program runs.



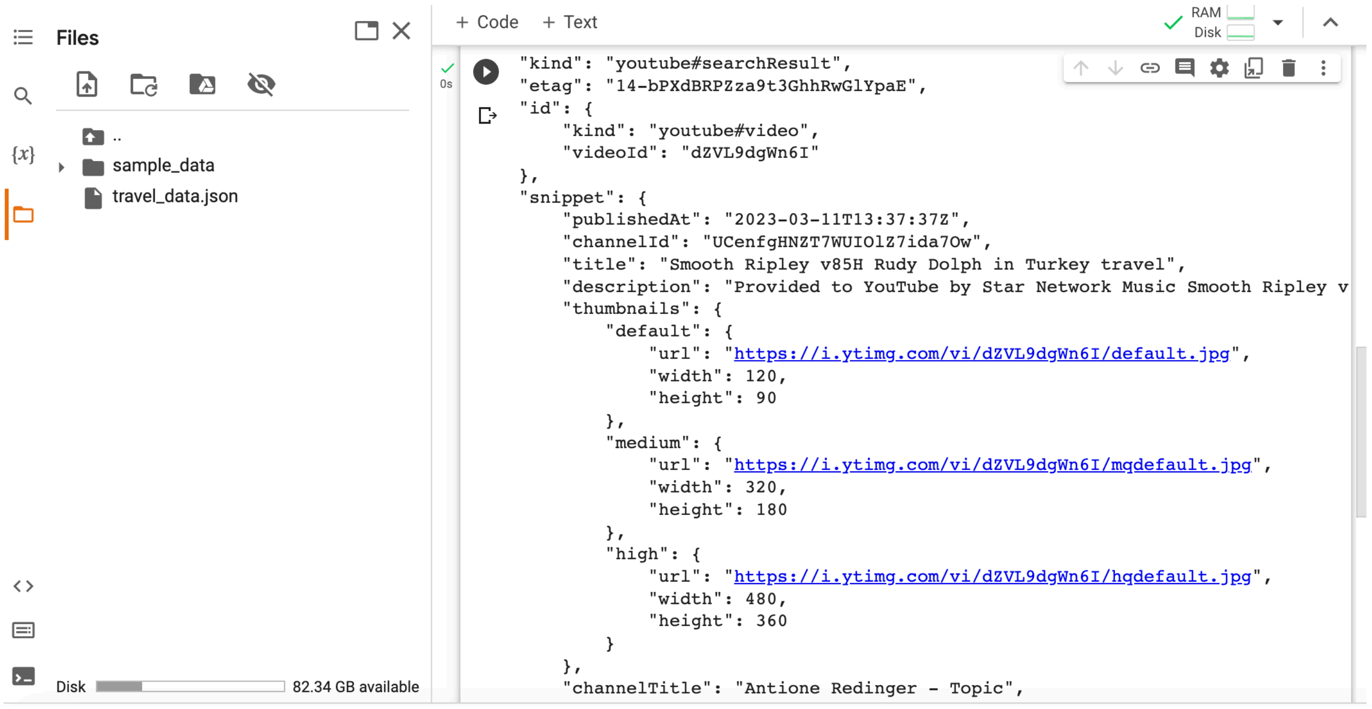
```
rad_city = input("Enter the radius from the city co-ordinates in metres ")
radius_payload = {'radius': rad_city, 'lon' : i_cor2, 'lat': i_cor1, 'apikey' :'5a
t_radius_res = requests.get("https://api.opentripmap.com/0.1/en/places/radius?", p
radius_res = t_radius_res.text

listObj = []

parse_radius = json.loads(radius_res)
for fea in parse_radius['features']:

  listObj.append({
  "id": fea['id'],
  "coordinates": fea['geometry']['coordinates'],
  "name": fea['properties']['name'],
  "distance from city centre in metres": fea['properties']['dist'],
  "type": fea['properties']['kinds']
  })

#generates JSON file
json_pdata = json.dumps(listObj, indent=4)
with open("sample.json", "w") as outfile:
  outfile.write(json_pdata)
```

5. Streaming APIs

   We decided to make use of Youtube API. We have used API key authentication to stream the data. We have decided to search with the words “travel” posted within the first minute. We retrieve this data and format it in readable format and write it into a JSON file travel_data.json.

   Link to code - https://colab.research.google.com/drive/15a6lEwrkFSeh0iiZsk7xauYS7L_Tbi3s?usp=sharing

Output:



```
"kind": "youtube#searchResult",
"etag": "14-bPXdBRPZza9t3GhhRwGlYpaE",
"id": {
    "kind": "youtube#video",
    "videoId": "dZVL9dgWn6I"
},
"snippet": {
    "publishedAt": "2023-03-11T13:37:37Z",
    "channelId": "UCenfgHNZT7WUIOlZ7ida7Ow",
    "title": "Smooth Ripley v85H Rudy Dolph in Turkey travel",
    "description": "Provided to YouTube by Star Network Music Smooth Ripley v
    "thumbnails": {
        "default": {
            "url": "https://i.ytimg.com/vi/dZVL9dgWn6I/default.jpg",
            "width": 120,
            "height": 90
        },
        "medium": {
            "url": "https://i.ytimg.com/vi/dZVL9dgWn6I/mqdefault.jpg",
            "width": 320,
            "height": 180
        },
        "high": {
            "url": "https://i.ytimg.com/vi/dZVL9dgWn6I/hqdefault.jpg",
            "width": 480,
            "height": 360
        }
    },
    "channelTitle": "Antione Redinger - Topic",
```

References

[1] https://opentripmap.io/product [2] https://opentripmap.io/docs

[3] https://developers.google.com/youtube/v3/live/getting-started