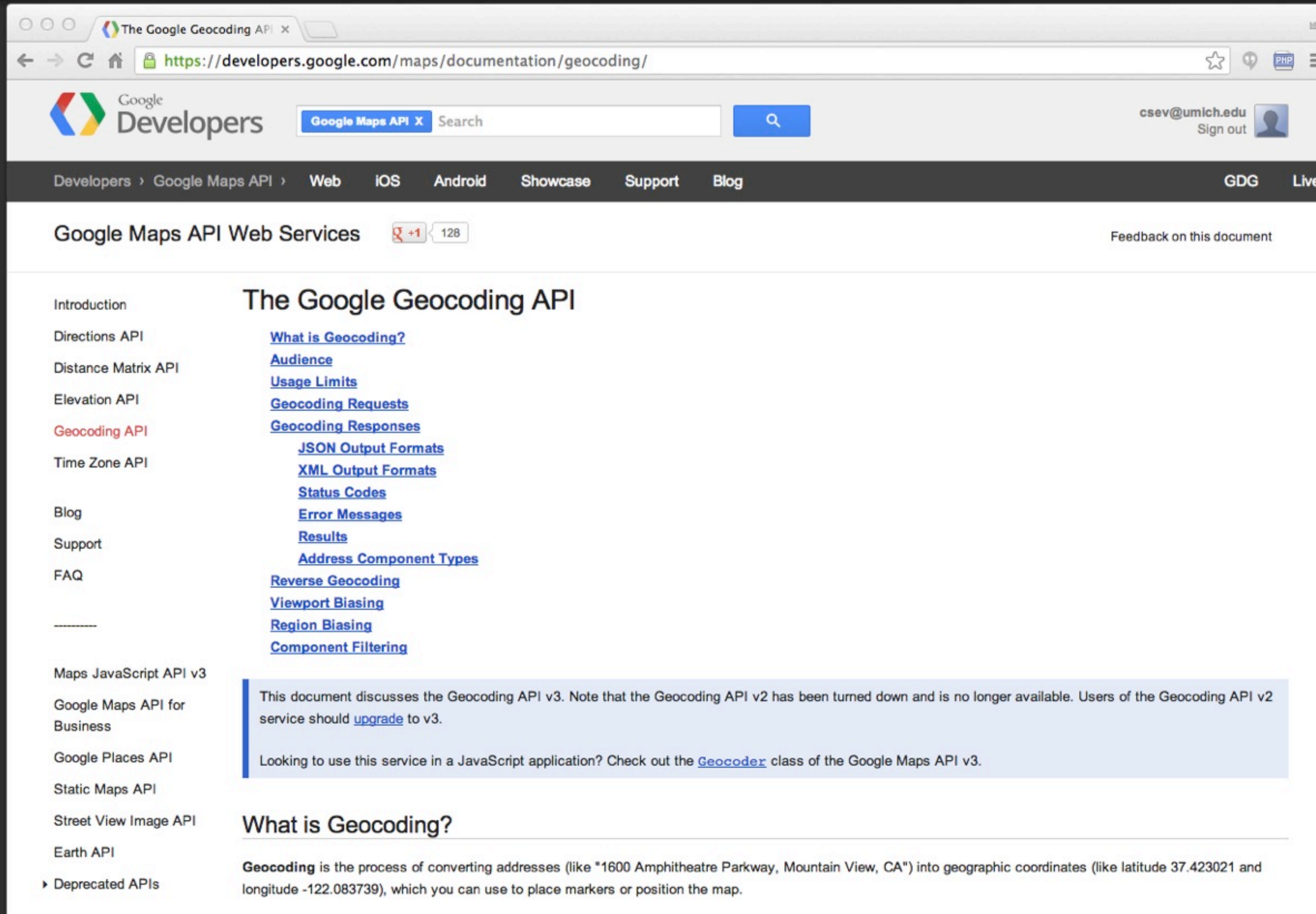


# Application Program Interface


The API itself is largely abstract in that it specifies an interface and controls the behavior of the objects specified in that interface. The software that provides the functionality described by an API is said to be an “implementation” of the API. An API is typically defined in terms of the programming language used to build an application.

<http://en.wikipedia.org/wiki/API>





The screenshot shows the Google Developers website for the Google Geocoding API. The browser address bar displays <https://developers.google.com/maps/documentation/geocoding/>. The page header includes the Google Developers logo, a search bar, and a user profile for csev@umich.edu. The main navigation bar lists various API categories: Developers, Google Maps API, Web, iOS, Android, Showcase, Support, and Blog. The page title is "Google Maps API Web Services" with a +1 button and a count of 128. A sidebar on the left lists various APIs, with "Geocoding API" highlighted in red. The main content area is titled "The Google Geocoding API" and contains a list of links: What is Geocoding?, Audience, Usage Limits, Geocoding Requests, Geocoding Responses, JSON Output Formats, XML Output Formats, Status Codes, Error Messages, Results, Address Component Types, Reverse Geocoding, Viewport Biasing, Region Biasing, and Component Filtering. A blue box contains a notice: "This document discusses the Geocoding API v3. Note that the Geocoding API v2 has been turned down and is no longer available. Users of the Geocoding API v2 service should [upgrade](#) to v3." Below this, another blue box says: "Looking to use this service in a JavaScript application? Check out the [Geocoder](#) class of the Google Maps API v3." The page footer includes a section titled "What is Geocoding?" which defines geocoding as the process of converting addresses into geographic coordinates.

Google Maps API Web Services  128 [Feedback on this document](#)

**The Google Geocoding API**

- [What is Geocoding?](#)
- [Audience](#)
- [Usage Limits](#)
- [Geocoding Requests](#)
- [Geocoding Responses](#)
- [JSON Output Formats](#)
- [XML Output Formats](#)
- [Status Codes](#)
- [Error Messages](#)
- [Results](#)
- [Address Component Types](#)
- [Reverse Geocoding](#)
- [Viewport Biasing](#)
- [Region Biasing](#)
- [Component Filtering](#)

This document discusses the Geocoding API v3. Note that the Geocoding API v2 has been turned down and is no longer available. Users of the Geocoding API v2 service should [upgrade](#) to v3.

Looking to use this service in a JavaScript application? Check out the [Geocoder](#) class of the Google Maps API v3.

**What is Geocoding?**

**Geocoding** is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers or position the map.

<https://developers.google.com/maps/documentation/geocoding/>



```
{
  "status": "OK",
  "results": [
    {
      "geometry": {
        "location_type": "APPROXIMATE",
        "location": {
          "lat": 42.2808256,
          "lng": -83.7430378
        }
      },
      "address_components": [
        {
          "long_name": "Ann Arbor",
          "types": [
            "locality",
            "political"
          ],
          "short_name": "Ann Arbor"
        }
      ],
      "formatted_address": "Ann Arbor, MI, USA",
      "types": [
        "locality",
        "political"
      ]
    }
  ]
}
```

[http://maps.googleapis.com/maps/api/geocode/json?  
address=Ann+Arbor%2C+MI](http://maps.googleapis.com/maps/api/geocode/json?address=Ann+Arbor%2C+MI)

geojson.py



```
import urllib.request, urllib.parse, urllib.error
import json

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

while True:
    address = input('Enter location: ')
    if len(address) < 1: break

    url = serviceurl + urllib.parse.urlencode({'address': address})

    print('Retrieving', url)
    uh = urllib.request.urlopen(url)
    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

    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)
```

Enter location: Ann Arbor, MI  
Retrieving http://maps.googleapis.com/...  
Retrieved 1669 characters  
lat 42.2808256 lng -83.7430378  
Ann Arbor, MI, USA  
Enter location:

geojson.py

# API Security and Rate Limiting





## Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) of the University of Michigan School of Information and [open.umich.edu](http://open.umich.edu) and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

... Insert new Contributors here