How to Perform a GeoIP Lookup with Node.js

Posted on April 25, 2018

In this article, I'm going to walk you through the best possible way to find the physical location of an IP address using Node.js (also known as IP geolocation).

Unfortunately, there is no standard way to figure out where an IP address is physically located. Instead, companies referred to as GeoIP providers aggregate many different pieces of data together to build an accurate database of IP location data.

GeoIP data is typically comprised of:

- Domain WHOIS data (which itself must be aggregated by data providers)
- Regional Internet Registries, which hand out large blocks of IP addresses to various Internet Service Providers around the world (ISPs)
- BGP feeds from large ISPs
- Latency information (how long does it take for a packet from certain physical locations to reach the destination IP

While getting all of the above information yourself is very complicated and expensive, there are luckily a few great service providers who've already done this work and sell GeoIP data that you can easily consume.

Today I'll show you how to use our newly released simple-geoip Node.js library to perform a GeoIP database lookup and return the physical location of any IP address you might want to pinpoint.

Create a GeoIPify Lookup Account

The first thing you'll need to do to use the simple-geoip library is go create a free GeoIPify account: https://geoipify.whoisxmlapi.com/signup

GeoIPify is one of the largest and least expensive GeoIP providers. You can use the GeoIPify service to perform 1,000 free GeoIP queries each month, or you can pay them a flat fee of $27 per month for 100,000 queries. Need more queries? You can see all the available plans on the GeoIPify pricing page.

Once you've created and logged into your GeoIPify account, you'll need to view your account's products page and copy your API key — you will need this later to make GeoIP queries.

Install the simple-geoip Package

Now that your account is setup, the next thing you need to do is install the Node package. From the command line, run the following command:

```
$ npm install simple-geoip
```

This will download and install the latest release of the simple-geoip package from NPM.

Perform a GeoIP Lookup Using simple-geoip

Now that you have both an account and the simple-geoip package installed, let's take a look at some code you can run to look up the physical address of any IP address you want.

Here’s a small script, `geoip.js`, which will find the physical location of a popular IP address (`8.8.8.8`, one of Google's core DNS servers):

```
const GeoIP = require("simple-geoip");

let geoIP  = new GeoIP("your-api-key");
geoIP.lookup("8.8.8.8", (err, data) => {
 if (err) throw err;
 console.log(data);
});
```

As you can see, there are really only three steps to using the library:

- Import the library
- Create a `GeoIP` object by giving it your API key that was created when you signed up for the GeoIPify service
- Run the `lookup` method, passing in the IP address you want to verify and a callback function. This callback function is what will be run when the GeoIP lookup has completed.

The data that’s returned in the callback will look something like this:

```
{
  "ip":"8.8.8.8",
  "location": {
    "country":"US",
    "region":"California",
    "city":"Mountain View",
    "lat":37.40599,
    "lng":-122.078514,
    "postalCode":"94043",
```

```
    "timezone":"-08:00"
  }
}
```

This JSON data tells you everything you need to know about the physical location of the `8.8.8.8` IP address.

Behind the scenes, the GeoIPify API service is handling all the GeoIP database lookups and data aggregation — getting data from providers and processing millions of updates per day.

Customizing the GeoIP Lookup Behavior in simple-geoip

One of the nice things about the simple-geoip Node.js library is that it automatically retries failed requests up to five times in a row.

For instance, let's say you are attempting to perform a GeoIP lookup request and your internet connection dies half-way through. Instead of simply erroring out the simple-geoip lookup will retry the request to give it another chance to go through.

In the event that you'd prefer the simple-geoip library *not* retry failed requests, you can pass in some optional configuration data when creating the `GeoIP` lookup instance like so:

```
const GeoIP = require("simple-geoip");

let geoIP  = new GeoIP("your-api-key", { retries: 2 });
geoIP.lookup("8.8.8.8", (err, data) => {
 if (err) throw err;
 console.log(data);
});
```

You can set the `retries` amount to any number between `0` and as much as you want. One thing to keep in mind, however: the more retries you allow the slower a request might be in the event of

a failure.

While retries are handy when working around partial network outages, if you are having a serious outage it might be better to just error out early on without wasting a lot of time retrying your failed requests. The default retry behavior (five retries) is usually a good choice for most people.

Use Your New GeoIP Data

Now that you've seen how easy it is to find the physical location of IP addresses using the simple-geoip library, you should start implementing GeoIP lookups into your product or service!

Some really common use cases for GeoIP data include:

- Detecting a user's country when they visit your website and providing a customized experience for them based on their location (language, ads, design, currency, etc.)
- Block users from certain locations from accessing your website. For instance, if you're a video streaming provider and only have the rights to stream video in a specific country, GeoIP lookups can provide you with that data so you can only serve customers in regions where you can legally operate.
- Fraud and risk mitigation. If you notice a large amount of fraud coming from a specific location, temporarily blocking visitors from that location can be a quick way to help mitigate fraud and other issues.

By analyzing the IP addresses of visitors to your website you can greatly enhance any web products and services.

Use simple-geoip

To wrap things up: performing GeoIP lookups doesn't have to be hard or expensive. By using our new simple-geoip Node.js library and the GeoIPify service you can easily build and manage evena large web product for very little money.

If you need to perform GeoIP lookups, please check out the simple-geoip library as it makes looking up IP address location information incredibly simple.

If you have any questions, please leave a comment below or email me!