The Best Ways to Get a User's Location in JavaScript

Posted on September 11, 2019

Geolocating your website's users can be useful for a wide variety of purposes. For example, you may want to show a different version of your website to users in different localities. You may be trying to better understand where your users live so you can tailor your website to better suit their needs. Or, maybe your website can only function in certain areas.

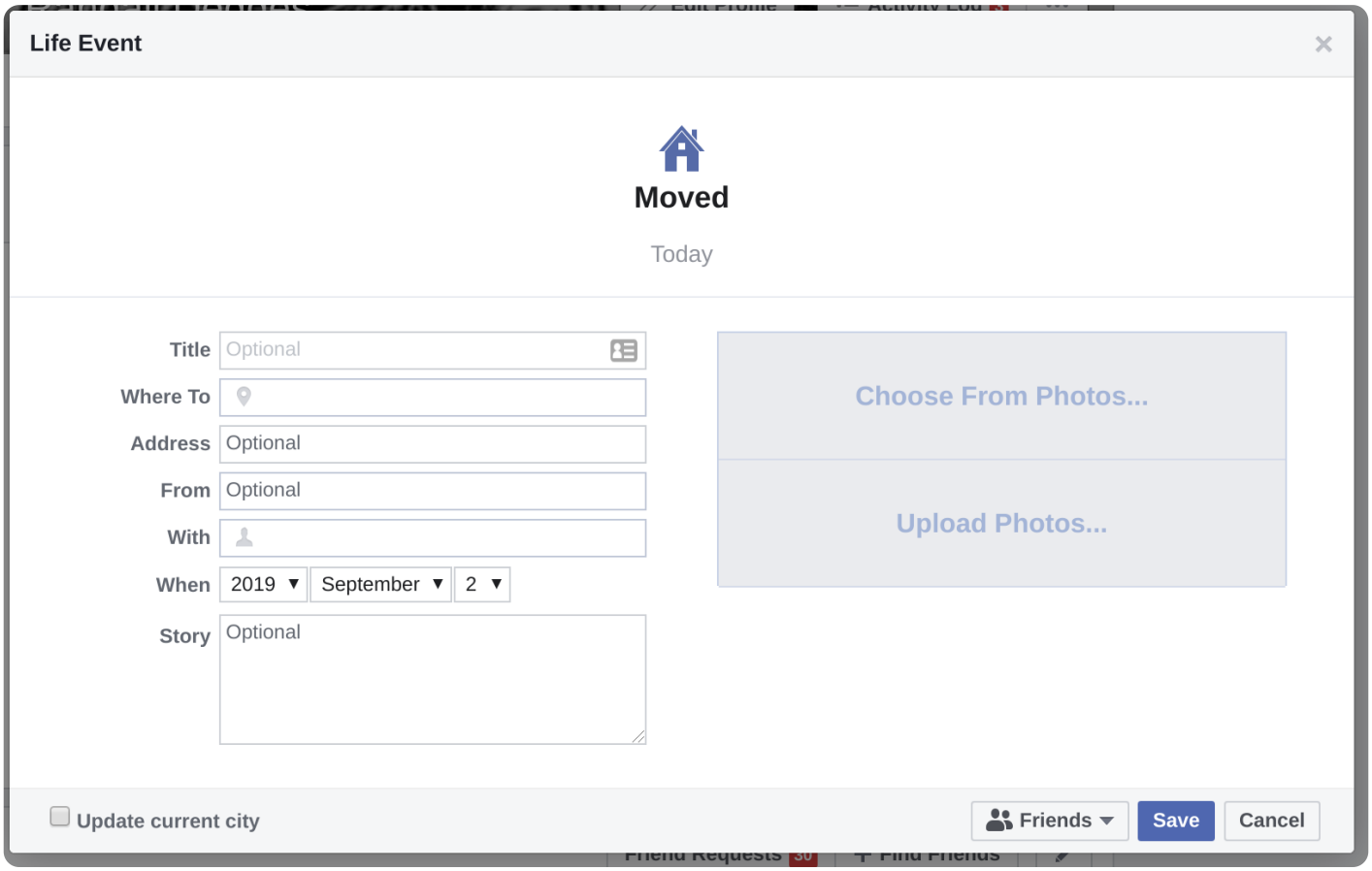
Whatever your reasons, geolocating your users and knowing where they're coming from can be useful.

There are a few different ways to geolocate users by using JavaScript. Each method has its own tradeoffs. I'm going to cover **all** the different ways in which you can geolocate users by using JavaScript below.

The methods below are ranked from best to worst (factoring in accuracy, convenience, and complexity). This will hopefully help you decide which approach to take according to your requirements.

The Most Accurate Way to Locate Your Users: Ask!

The most accurate way to figure out where your users are located is to… **ask them**! As I was doing my research for this article, I was surprised that I hadn't seen anyone else mention this.



As you're going to learn in a few minutes, locating a user isn't always straightforward, and no matter what method you choose, accuracy isn't guaranteed. Having said that, there certainly isn't a more accurate way to find the location of your users than to just ask them.

If your goal is to figure out precisely where someone lives, why not just throw together a simple web form that prompts the user for their physical address? Tons of sites do this and users are more than willing to give their address to you if necessary.

In particular, if you're building any sort of shopping or e-commerce-type website, getting a user's address is a standard part of the flow. If you're building any sort of social media-type application, some amount of location data is typically requested (think Facebook, Twitter, etc.).

On the other hand, if you're building the type of website that users wouldn't expect you to need their address for, this approach obviously won't work, and could potentially be detrimental (you'll run the risk of giving your users the creeps).

Find Your User's Location in JavaScript Using an API

If it doesn't make sense for you to ask your users where they live, the next most accurate way to locate your website's users is by using an API service (like geoipify). Services like geoipify are called "IP geolocation services" because they allow you to take a user's IP address and map it back to an actual physical location.

The way IP geolocation services work is by aggregating IP address data from many different sources including:

- **Information from internet service providers who provide information like GPS coordinates and addresses of IP addresses they assign.** For example, when you signed up for internet service at your house, your ISP gave you a public IP address at your home. Your ISP may then record that data and share it out with third parties.

- **Data mining.** If you've ever voluntarily given your address to a website, that website may have shared that information with other third parties so that they can map your IP address back to your physical address.

- **Merging databases from various providers.** There are several large IP geolocation providers. By merging these databases together, you can improve the coverage of IP geolocation data.

- **Latency-based geolocation techniques.** Because talking to devices over the public internet requires routing between many different devices across the world, there are various techniques that can be used to geolocate an IP address by analyzing the time it takes to communicate with a device in a known location.

As you can probably tell from the above description, it isn't easy to access IP address geolocation

data on your own, so using a service to pragmatically query this information is your only real choice.

One of the benefits to using an IP geolocation service like geoipify is that it won't detract from your user experience at all. You don't need to ask the user for their address, prompt them for permissions, or anything like that. All you need to do is take the user's public IP address (which you can easily retrieve by using your programming language of choice) and run it through an IP geolocation service to figure out where that user is located.

The downside to using an IP geolocation service is that users can manipulate their IP address by using tools like VPN services and IP address spoofing. If a user is able to manipulate their IP address so that your website thinks the user has a different IP address than they actually do, you're obviously going to be getting incorrect location information when you later geocode the user's IP address.

All that said, this method is still extremely accurate in most circumstances. And the odds are, if a user is purposefully manipulating their IP address, they are unlikely to want you to know their location regardless.

If you want to use geoipify to find the location of your website's users, you can sign up and use the service for free here.

Here's how you can easily find the location of a user through their IP address by using the simple-geoip JavaScript library. Here's a full example application showing how it works:

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

let geoIP = new GeoIP("your geoipify api key goes here");

geoIP.lookup("8.8.8.8", (err, data) => {
  if (err) throw err;
  console.log(data);
});
```

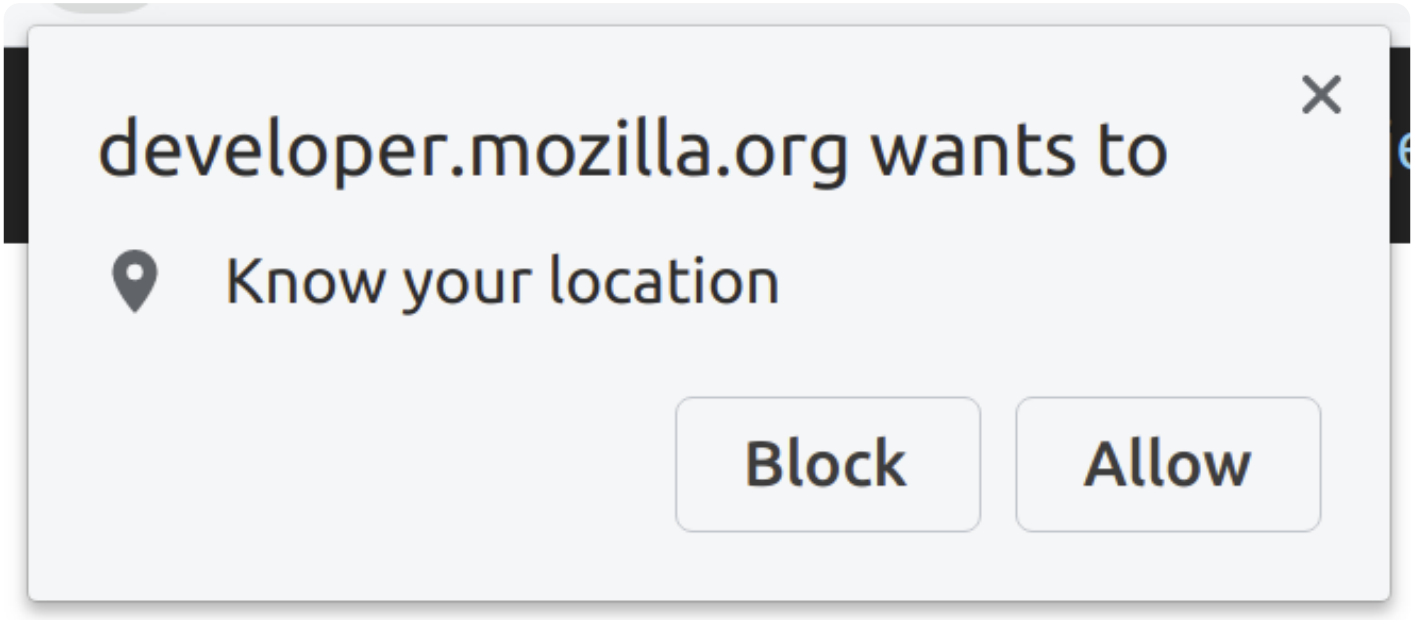
If you were to run the program above, you would get back the following location data for the user with IP address `8.8.8.8` (**NOTE**: `8.8.8.8` is a special address owned by Google).


```
{
  ip: '8.8.8.8',
  location: {
    country: 'US',
    region: 'California',
    city: 'Mountain View',
    lat: 37.40599,
    lng: -122.078514,
    postalCode: '94043',
    timezone: '-07:00',
    geonameId: 5375481
  },
  domains: [ '0--9.ru', '000180.top', '0002.by', '00027.hk', '00049ok.com'
  as: {
    asn: 15169,
    name: 'Google LLC',
    route: '8.8.8.0/24',
    domain: 'https://about.google/intl/en/',
    type: 'Content'
  },
  isp: 'Google'
}
```


Not bad, huh? With just a few lines of code, you can get a **ton** of useful location information about a user.

Use the Browser Geolocation API

The last (and my least favorite) method for finding out where a user is located is to use the built-in Geolocation API that most browsers now support.



Essentially, this API allows you to prompt the user for their location information. If the user "allows" you to access their location data, then you can use the Geolocation API to get the GPS coordinates of the user (latitude and longitude).

Unfortunately, depending on the device the user possesses, getting accurate location data may take a while.

Here's a small web application that displays your GPS coordinates by using the Geolocation API in your browser (courtesy of Mozilla).

HTML Page

```
<button id = "find-me">Show my location</button><br/>
<p id = "status"></p>
<a id = "map-link" target="_blank"></a>
```

JavaScript

```
function geoFindMe() {

  const status = document.querySelector('#status');
  const mapLink = document.querySelector('#map-link');

  mapLink.href = '';
  mapLink.textContent = '';

  function success(position) {
    const latitude  = position.coords.latitude;
    const longitude = position.coords.longitude;

    status.textContent = '';
    mapLink.href = `https://www.openstreetmap.org/#map=18/${latitude}/${lon
    mapLink.textContent = `Latitude: ${latitude} °, Longitude: ${longitude}
  }

  function error() {
    status.textContent = 'Unable to retrieve your location';
  }

  if (!navigator.geolocation) {
    status.textContent = 'Geolocation is not supported by your browser';
  } else {
    status.textContent = 'Locating…';
    navigator.geolocation.getCurrentPosition(success, error);
  }

}

document.querySelector('#find-me').addEventListener('click', geoFindMe);
```

There are a few problems I see with the Geolocation API:

- **It requires the user to accept location permissions.** This stands a chance of freaking your users out and driving them away from your website. Depending on the type of application you're building, this could cause a substantial user experience issue.

- **Users can reject your location request.** If a user chooses to reject your location request, you obviously won't be getting any data. Using an IP geolocation service allows you to **always** retrieve location information, even without a user's explicit consent.

- **You only get GPS coordinates.** While getting GPS coordinates can be useful if you are later able to translate them to a physical address, this will require the help of an external API service (like Google's geocoding API), which means you will still need to rely on a third party to help you make sense of the data you're getting.

If your use case is simple, however, using the built-in browser Geolocation API might be something to look into.

Summary: The Best Ways to Get a User's Location

If you need to get a user's location information, you really only have a few choices:

- Ask the user;

- Use an IP address geolocation service like geoipify;

- Use the built-in browser Geolocation API.

While each method has its own drawbacks and benefits, my personal favorite method is to silently locate a user from their IP address by using a geolocation service. My reasoning is simple: it's the only method that doesn't require impacting the user experience of your website, and geolocation services allow you to get fairly accurate location data without much ado.

Hope you found this useful!