EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

THIRD PARTY API

Date	17 November 2022
Team ID	PNT2022TMID03761
Project Name	Emerging Methods for Early Detection of Forest Fires

Third party APIs are APIs provided by third parties — generally companies such as Facebook, Twitter, or Google — to allow you to access their functionality via JavaScript and use it on your site. One of the most obvious examples is using mapping APIs to display custom maps on your pages.

Browser APIs are built into the browser — you can access them from JavaScript immediately. For example, the Web Audio API we saw in the Introductory article is accessed using the native AudioContext object. For example:

```
const audioCtx = new AudioContext();
// …
const audioElement = document.querySelector('audio');
// …
const audioSource = audioCtx.createMediaElementSource(audioElement);
// etc.
```

Copy to Clipboard

Third party APIs, on the other hand, are located on third party servers. To access them from JavaScript you first need to connect to the API functionality and make it available on your page. This typically involves first linking to a JavaScript library available on the server via a <script> element, as seen in our Mapquest example:

```
<script
  src="https://api.mqcdn.com/sdk/mapquest-js/v1.3.2/mapquest.js"
  defer></script>
<link
  rel="stylesheet"
  href="https://api.mqcdn.com/sdk/mapquest-js/v1.3.2/mapquest.css" />
```

Copy to Clipboard

You can then start using the objects available in that library. For example:

```
const map = L.mapquest.map('map', {
  center: [53.480759, -2.242631],
  layers: L.mapquest.tileLayer('map'),
  zoom: 12
});
```

Copy to Clipboard

Here we are creating a variable to store the map information in, then creating a new map using the mapquest.map() method, which takes as its parameters the ID of a element you want to display the map in ('map'), and an options object containing the details of the particular map we want to display. In this case we specify the coordinates of the center of the map, a map layer of type map to show (created using the mapquest.tileLayer() method), and the default zoom level.

This is all the information the Mapquest API needs to plot a simple map. The server you are connecting to handles all the complicated stuff, like displaying the correct map tiles for the area being shown, etc.

Note: Some APIs handle access to their functionality slightly differently, requiring the developer to make an HTTP request to a specific URL pattern to retrieve data. These are called RESTful APIs — we'll show an example later on.

They usually require API keys

Security for browser APIs tends to be handled by permission prompts, as discussed in our first article. The purpose of these is so that the user knows what is going on in the websites they visit and is less likely to fall victim to someone using an API in a malicious way.

Third party APIs have a slightly different permissions system — they tend to use developer keys to allow developers access to the API functionality, which is more to protect the API vendor than the user.

You'll find a line similar to the following in the Mapquest API example:

```
L.mapquest.key = 'YOUR-API-KEY-HERE';
```
Copy to Clipboard

This line specifies an API or developer key to use in your application — the developer of the application must apply to get a key, and then include it in their code to be allowed access to the API's functionality. In our example we've just provided a placeholder.

Note: When creating your own examples, you'll use your own API key in place of any placeholder.

Other APIs may require that you include the key in a slightly different way, but the pattern is relatively similar for most of them.

Requiring a key enables the API provider to hold users of the API accountable for their actions. When the developer has registered for a key, they are then known to the API provider, and action can be taken if they start to do anything malicious with the API (such as tracking people's location or trying to spam the API with loads of requests to stop it working, for example). The easiest action would be to just revoke their API privileges.