Client-Side API: Introduction to web API

Tags

- Application Programming Interfaces (API) lets you manipulate different aspects of the browser and the operating system the site is running on.

What are APIs

- Think of APIs are constructs made available to you as a programmer, they abstract some of the complex code away from you to use simple language in its place
- Think of the fetch() API it allows you to make a connection to a server in a few lines of code.

Client Side API in JS:

- Client-side JS in particular has many APIs available to it, they fall in two categories:
    - **Browser API:** are built into the browser and expose the data on the browser to your JS program. The Web Audio API is an example of this, the JS construct for the Web Audio API lets you take in an audio track, alter its volume, apply effects to it. The browser is actually doing some high complex processing in the background using C++ or Rust, but that complexity is taken away fro m you in the API
    - **Third-Party API:** are not built into the browser by default, and you generally have to retrieve their code and information from elsewhere. E.g. Google Maps API that lets you display an interactive map to your office on your website. It provides a special set of constructs you can use use to query the Google maps services and return specific information.

Relationship between JS, API and other JS tools

- JS: a high level scripting language built into the browser that allows you implement functionality into the HTML doc
- Browser API: constructs built into the web browser that sits on top of the JS language and allows you to implement functionality more easily
- Third Party APIs: constructs built into external platforms that allow you to use their platforms functionality in your own web page e.g. A weather app API will allow you to use their weather tracking data which you could implement in your own web page
- JavaScript libraries — Usually one or more JavaScript files containing custom functions that you can attach to your web page to speed up or enable writing common functionality. Examples include jQuery, Mootools and React.
- JavaScript frameworks — The next step up from libraries, JavaScript frameworks (e.g. Angular and Ember) tend to be packages of HTML, CSS, JavaScript, and other technologies that you install and then use to write an entire web application from scratch. The key difference between a library and a framework is "Inversion of Control". When calling a method from a library, the developer is in control. With a framework, the control is inverted: the framework calls the developer's code.

What can APIs do ?

- **APIs for manipulating documents** loaded into the browser. The most common is the DOM (Document Object Model) API, which allows you to manipulating HTML and CSS.
- **APIs for fetch data from server.** Updating small sections of a webpage with out loading the whole document has a huge impact on performance. e.g. if

you need to update a stock listing, you can do so using an API that will fetch data relating to the stock, this way the entire webpage is not reloaded when the listing is updated. The main APIs are the `Fetch API` and `XMLHttpRequest API`