**Introduction to Web APIs**

**Learn what APIs are and how they’re useful tools for web development.**

**What are APIs?**

An *Application Programming Interface* (*API*) is a software tool that makes it easier for developers to interact with another application to use some of that application’s functionality. Like tools in the physical world, APIs are built to solve specific, repeated, and potentially complex problems.

Imagine we needed to twist a screw into a piece of wood. We could try to take the screw and twist it in with our fingers, but that would be difficult and inefficient. Twisting one screw would be hard enough — having to do it multiple times would be near impossible! Instead, we could make the task much easier by using a tool that someone else created, in this case, a screwdriver! When used properly, the screwdriver can be used for one specific type of screw and can be reused for the same job over and over again!

Relating this analogy back to APIs, we probably don’t want to have to write up the same code for the same problems over and over again. If the right API exists to solve our problem, we could make it easier on ourselves and use the API instead. But, before we fully commit to using an API, we should go over some important considerations!

**Types of APIs**

There are two main categories of web APIs, browser APIs and 3rd party APIs.

*Browser APIs* are specific to writing code related to browsers and give developers access to information that the browser can also access. One example is the [geolocation API](https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API) which allows the program to know where a user is when accessing our app. This specific API requires that the user grant permissions to the browser to access their geolocation information. There are also browser APIs for audio, cryptography, VR, and much more.” To see a comprehensive list of browser APIs and how to interact with them, check out [MDN’s documentation of web APIs](https://developer.mozilla.org/en-US/docs/Web/API).

*Third-party APIs* are apps that provide some type of functionality or information from a third-party, usually a company. For example, we could use the [OpenWeather API](https://openweathermap.org/" \t "_blank) to get in-depth information about the weather in an area, forecasts, historical weather data, and more! On our own, we wouldn’t even have access to this data and we would certainly not want to write up this code ourselves just for one app!

**Requesting Information from a Third-party API**

Each API has a specific structure and protocol that we have to follow in order to gain access to its functionality.

**Rules and Requirements**

Organizations that maintain third-party APIs often set rules and requirements for how developers can interact with their API. For OpenWeather, we need to sign up for an account and generate a special token called an *API key* that grants our account the ability to make API requests. These API keys are unique to individual accounts and should be kept secret. OpenWeather provides some free functionality and some paid functionality. So before committing to a third-party API, check out their specifications which can often be found in the API documentation. Here’s [OpenWeather’s documentation](https://openweathermap.org/api" \t "_blank) to look over as an example.

**Making Requests**

Some of an API’s specifications relate to making a *request* for data. These specifications could include more parameters for specific information or the inclusion of an API key. For example, when using the OpenWeather API, we need to provide more information to search for weather information — such information could include: the name of a city, time of day, the type of forecast, etc. These specifications for making a request can also be found in the API documentation.

**Response Data**

After we make a successful API request, the API sends back data. Many APIs format their data using *JavaScript Object Notation* (*JSON*) which looks like a JavaScript object. Here’s a quick example of what JSON data might look like:

{   
  "temperature" : {   
     "celcius" : 25,  
  },  
  "city": "chicago",   
}

It’s then up to us how to determine how to *consume*, or use, this information in our apps. If we got back that sample JSON above, we could parse out the temperature information and display it on our app.

**Review**

Congrats, we’ve now gone through the basics of web APIs! Here’s a quick recap of what we covered:

* With web APIs, we have a tool that we can use to access the functionality and data of another application.
* There are two main types of APIs, browser and third-party.
  + Browser APIs require specific syntax and permissions.
  + Third-party APIs have their own rules and requirements set by the organizations that maintain them.
* When making a request to API, we might have to supply more details about what information we want.
* If we get a successful response, we still have to decide how to consume the sent back data.

That’s a lot to take in, but knowing that these tools exist opens up more possibilities for what we can do with our apps!