Front-End Web Development

API Requests and Responses

GENERAL ASSEMBLY

Today's Learning Objectives

In this lesson, you will:

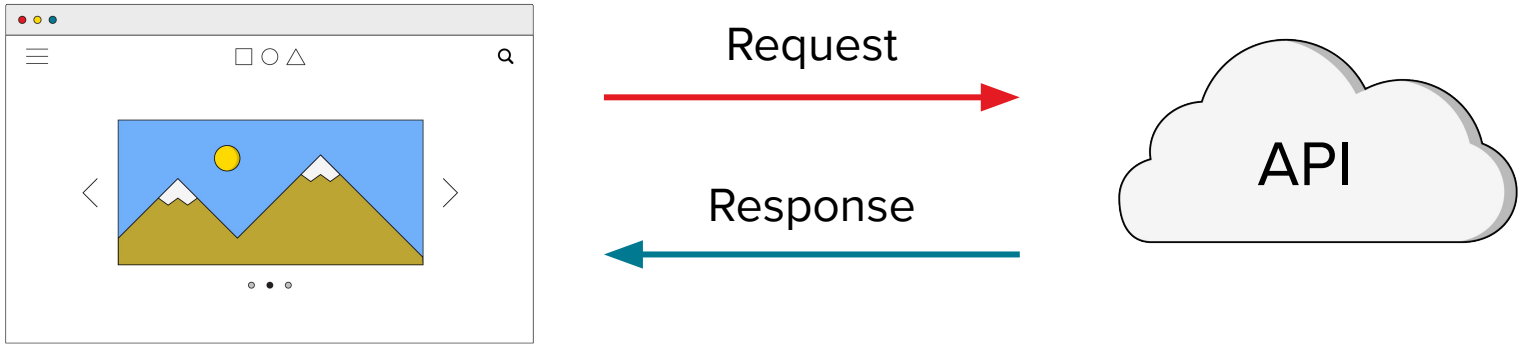
- Make HTTP requests to external API sources for data.
- Evaluate APIs based on documentation.
- Use API responses to update HTML content.

Front-End Web Development

What Are APIs?

Welcome to the Neighborhood!

APIs, or **application programming interfaces**, are data sources that can give you access to information from all sorts of places on the internet. Simply put, an API is a computer that receives **requests** and returns **responses**.



Uses of APIs

So, what kind of data do APIs serve up? It'd be difficult to find any kind of data that **doesn't** have an API. Common uses are Google Maps plugins, social media integration, and payment APIs for handling customer purchases.

You can even use Google's Firebase API as your own customizable database for storing any data your applications might need!

An exhaustive list of public APIs can be found here:
https://github.com/public-apis/public-apis

Guided Walk-Through:
Weather API

We can see the steps for using an API in this CodePen weather app:

Reference code:

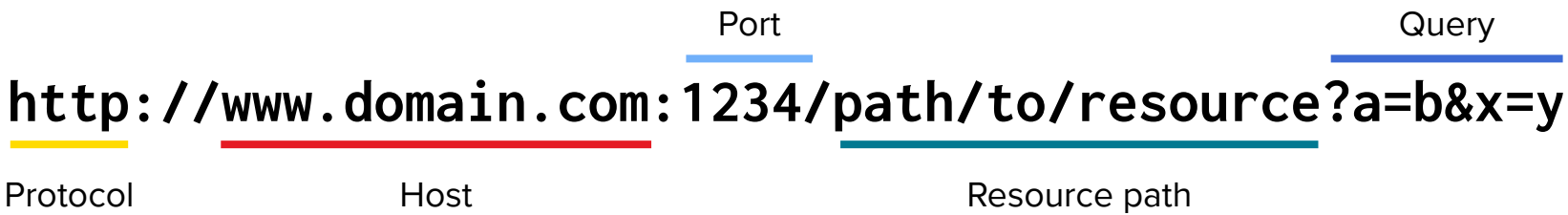
https://codepen.io/GAmarketing/pen/4c464f2f212562034784b916faef661b

Front-End Web Development

Fetching Data

APIs Sound Great! How Do I Get Started?

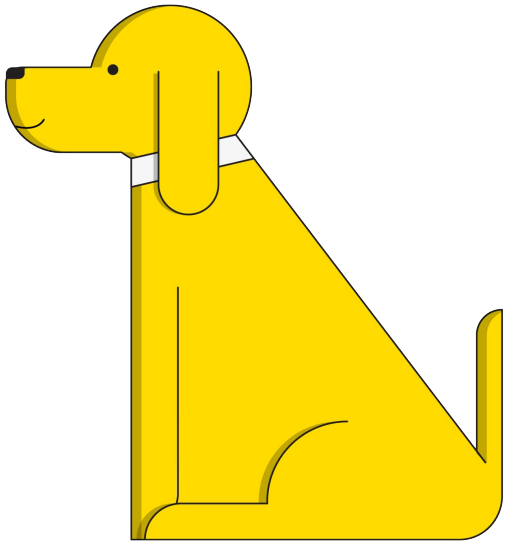
APIs work on a **request-response cycle**, so our first step is to make a request. API requests use the HTTP protocol, meaning we'll need to use a URL address.



Fetching Data With `fetch()`

Once you have an address, you can send a request using JavaScript's built-in **`fetch()`** function, which accepts the destination URL as a parameter.

```
const response = await fetch(url);
```



Asynchronous Code With Async/Await

API calls are **asynchronous** code. Because we have no idea how long the response will take to return, we have to **await** the result. We can do so using two keywords that come as a pair: `async` and `await`.

```
async function askForData(){
const response = await fetch(url);
const data = await response.json();
}
```

Front-End Web Development

Evaluating APIs

Read the Docs!

Every API is unique — you will have to explore its documentation to figure out what data you want to access and what URL to use in order to send requests. However, there are a few standard details to check for in any API:

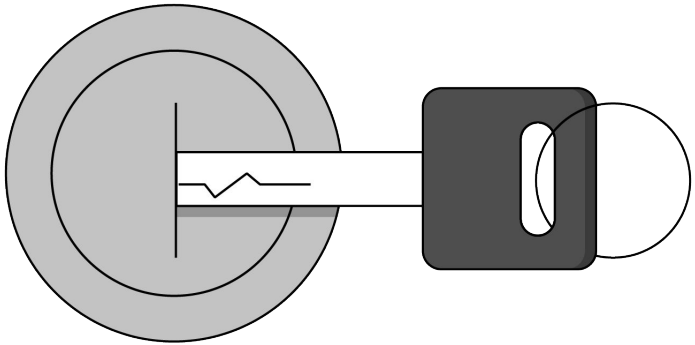
1. **Do I need an API key?**
    a. If so, are there rate limits or charges for requests?
    b. How do I include the key in my requests?
2. **What are the available endpoints for data?**
    a. What URL format can search for specific data?
3. **What will the response data look like?**
    a. Typically the response will be JSON.

API Keys

Many APIs require you to include a **key** with your request.

You can register with the API's website to receive your unique key.

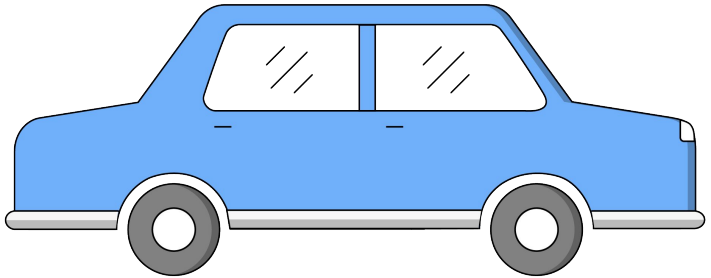
Keys are used to track your requests for rate limits and potential charges. Don't worry, most APIs have a free tier!



JS Objects

API endpoints will almost always return JSON in the form of an object.

```
{
  “data”: “car”,
  “make”: “ford”,
  “model”: “focus”
  “details”: {
    “color” : “blue”,
    “mileage” : “54019”
  }
}
```



Many-Layered Objects

API response objects are infamously complex in their formats. To dig down to the layer of information you actually want, carefully log one layer at a time until you reach the data you're looking for!

```
const carData = {
  “data”: “car”,
  “make”: “ford”,
  “model”: “focus”
  “details”: {
    “color” : “blue”,
    “mileage” : “54019”
  }
};
```

```
console.log(carData);
console.log(carData.details);
console.log(carData.details.color);
```


Partner Exercise:

Giphy Bank

90 minutes

Where would the internet be without GIFs? Use the giphy.com API to create a GIF search application using your new API skills!

Starter code:

https://codepen.io/GAmarketing/pen/4a5b50f51b158052a4d4bb9a7a66472c

Solution code:

https://codepen.io/GAmarketing/pen/379ead288f9c9fcb1a4eebcd61fde93c


GA