Exercise **Wednesday**

Intermediate JavaScript (/intermediate-javascript) / Asynchrony and APIs (/intermediate-javascript/asynchrony-and-apis) / Bike Index, Cryptocurrency Analytics App, API of Choice (Two-Day Project) - Part 1

Text

Goal: Practice using all of the tools that we have learned so far to create an app centered around your choice of public JSON APIs which support CORS. As always, make sure you test API calls with Postman first.

This is a multi-day project, so you'll have time to do the following:

- Experiment with different tools for working with asynchrony, including callbacks, promises, the Fetch API, and async functions
- Make multiple API calls in the application
- Separate out logic clearly, including keeping business logic separate from service and UI logic as necessary
- Write a great README with thorough setup instructions
- Experiment with chaining promises

Warm Up — Part 1

This warm up is for your first class session working on this multi-day project.

- Are there any potential disadvantages to using async functions?
- What is SOP and CORS? Why is it important for making API calls in our projects?
- What is your preferred method of making API calls, and why?

Warm Up — Part 2

This warm up is for your second class session working on this multi-day project.

- How does the Fetch API work?
- What does `fetch()` return?
- How do async functions work?
- What does the `await` keyword do?

Code

Carefully read all of the prompts below and select one to work on. You are encouraged to do preliminary research into each API to make your selection.

Bike Index API

Build an app using the Bike Index API (https://bikeindex.org/). Your app could list all the bikes that have been registered as stolen in a given location in the past week. Or it could display statistics — for instance, you could see which manufacturer is most frequently stolen in a given area.

To get started, check out the main page of the Bike Index API Docs (https://bikeindex.org/documentation/api_v3). When you are ready to set up an API key, scroll down to find the "Your applications" section of "OAuth" — it's from here you can set up an account and get an API key.

Cryptocurrency Analytics App

Build an app using the CoinGecko API (https://www.coingecko.com/). Take a look at the CoinGecko homepage to get a sense of what you can do — the homepage is entirely built with the CoinGecko API. CoinGecko also provides a helpful resource (https://apiguide.coingecko.com/getting-started/endpoint-demo) that explains what API endpoints are used to create the different features on the homepage. Here are some specific ideas of what you can do:

- Your app can list currencies and their metadata (information like a description and full name, their category, as well as their symbol name).
- Or, your app could show the price and exchange rates for currencies.
- You could also show the number of trades happening for each currency.

Important to Know:

- CoinGecko does not require an API key or account to get started on a basic free plan for hobbyists or devs who are using the API in a practice project. If you want to specifically practice with an API that requires a key, select a different prompt.
- Under the free plan, there is an API call limit of 10 - 50 calls per minute, depending on API "traffic", meaning how many people are using the API at the same time.
- Since we'll be using the API in practice projects only, we'll need to follow their requirements for attribution, that is, giving credit and linking to CoinGecko in our website. To learn about

CoinGecko's requirement for attribution, visit this webpage (https://www.coingecko.com/en/branding).

Resources:

- Start by reviewing the Official API Documentation (https://apiguide.coingecko.com/getting-started/introduction), which includes references, tutorials, and information on best practices, errors, and rate limits.
- Check out the API explorer (https://www.coingecko.com/en/api/documentation) to test out the various API endpoints. You'll be able to input optional and required parameters and you'll get the actual API output returned to you. The API explorer tool is an alternative to using Postman to test out the API.

Pick Your Own API

If you decide to use an API that isn't listed here, start with some research. We recommend checking out this comprehensive list of public APIs (https://github.com/public-apis/public-apis) that's organized by type of data and authentication required. However, there are many other online resources about free APIs. Take note of a few things when selecting an API:

- Make sure that the API has a free tier.
- Make sure to confirm that the API supports CORS and that it returns JSON (unless you want to experiment with a format like XML).
- We suggest looking for APIs that use an API key. This isn't required, however! It's up to you to decide what's most important for you to practice.
- We recommend that you only pick one API to focus on for the entire time so that you can have an opportunity to really build out your project.

Otherwise, if you liked the cryptocurrency analytics app prompt, but you want to work with an API that requires an API key, check out Polygon.io (https://polygon.io/docs/stocks/getting-started). Polygon includes information about stocks as well as crypto. The docs aren't quite as friendly as CoinGecko, but you will still be able to easily work with the API. Take note of a few things:

- The free plan for the Polygon.io API has a rate limit of 5 API calls per minute.
- Metadata, like the company name (not "ticker symbol") and description, only exists for stocks (https://polygon.io/docs/stocks/get_v3_reference_tickers__ticker).

Peer Code Review

- Does the application make at least one API call and work as expected?
- Is application logic fully separated into user interface, service logic, and if necessary, business logic?
- Does the application include a fully functional development environment, including a `.env` file to hide any API keys?
- Does the application include a README with all setup instructions?

Previous (/intermediate-javascript/asynchrony-and-apis/further-exploration-iifes)
Next (/intermediate-javascript/asynchrony-and-apis/further-exploration-chaining-promises)

Lesson 27 of 33
Last updated February 21, 2023

disable dark mode

Epicodus (http://www.epicodus.com)