

Exercise

Tuesday

Intermediate JavaScript (/intermediate-javascript)

/ Asynchrony and APIs (/intermediate-javascript/asynchrony-and-apis)

/ API Refactor, Dino Ipsum

Text

Goal: Practice making API calls in JavaScript using a promise. Start by following along with the homework to build out the refactor of the OpenWeather API project. Then move onto another project. Each project should include a full development environment, a `.env` file for sensitive API variables, and complete README instructions, including instructions for setting up an API key as needed. Make sure to include error handling and practice parsing JSON, too.

Warm Up

- What is a promise? What are the advantages and disadvantages of using a promise for async code?
- What is a static method? How is a static method different from an instance method?
- How can we use callbacks to handle async code?
- What causes callback hell and how can we avoid it?

Code

OpenWeather API

Follow along with the lessons to update the OpenWeather API project to use a promise, separate API business logic, and a static method.

Project Refactor

Next, refactor one of the projects you worked on in the last class session to use promises. Once again, make sure that API call business logic is properly separated. Make sure to use static methods, too.

Dino Ipsum API

Use this API to allow your users to generate dinosaur themed Lorem Ipsum (<http://dinoipsum.herokuapp.com/#res>). Use it any way you like.

For instance, you could create a dinosaur-themed game where someone needs to guess the name of the dinosaur in 7 guesses. You could have users guess letter by letter, or by guessing whole words, like in the game Wordle (<https://wordplay.com/>).

Whatever you do, don't worry about making a complex UI, like in the Wordle game that's linked above. Keep the UI as simple as possible and focus on completing the functionality of your application first. Once that's in place, you can move onto improving the UI.

Note: Be sure you set your output format to JSON, not HTML or text.

Peer Code Review

- Does the application have a fully functioning development environment?

- Does the application successfully wrap an API call in a promise?
- Are business and user interface logic separated?
- Are API keys stored in a `.env` file and protected?
- Does the README include instructions for setting up the project including getting an API key and adding it to a `.env` file?

[Previous \(/intermediate-javascript/asynchrony-and-apis/speaking-up\)](#)

[Next \(/intermediate-javascript/asynchrony-and-apis/fetch-api\)](#)

Lesson 23 of 33

Last updated more than 3 months ago.

[disable dark mode](#)



<http://www.epicodus.com>

© 2023 Epicodus (<http://www.epicodus.com/>), Inc.