

JavaScript Coding Exercise

Overview

Brightwheel uses internal and third-party APIs extensively throughout our platform to process and store data, as well as manage communication to end users. The goal of the exercise will be to consume a live third-party API and display the data according to the specifications given below. This exercise is meant to assess your code quality, use of object-oriented principles, and your design aesthetic.

For the live exercise review, you'll be sharing your screen, so please be prepared with a stable Internet connection and a quiet environment.

Exercise

Build a single-page web app that uses the Github REST API v3 to display a list of the top 100 most starred Github repositories including a list of the commits made in the last 24 hours

Provide us the link to your final product as a cloneable repo on Github or Bitbucket

Implementation Parameters

This will be a simple exercise, but organize, design, document, and test your code as if it were going into production. It should be fully-functioning code that can be run.

The design of each repository should be in a card form. Multiple cards should be visible per browser row. The minimum data to be included for each repository is the name, url and star count. The minimum data to be included for each commit is author name, commit date, and commit message. To help with rate limiting, feel free to have the commit list for each repository hidden by default and revealed by a user action.

Feel free to use third-party libraries, but please refrain from using styling or component libraries such as bootstrap and material-ui.

You'll receive bonus points for interesting UI, well-written documentation and tests, and anything you do that goes above and beyond the implementation requirements. Have fun with this!

Helpful Links

1. <https://developer.github.com/v3/>
2. <https://developer.github.com/v3/search/#search-repositories>
3. <https://developer.github.com/v3/repos/commits/>
4. <https://help.github.com/articles/searching-for-repositories/#search-by-number-of-stars>

What We Will Be Assessing

Functionality: Does the app do what we asked?

Code Quality: Does the code follow software engineering standards and best practices? Is it easy to understand and maintainable?

Technical Choices: Is the implemented technical approach appropriate for what was asked?