

Code Assignment

Nipper API

Estimated Time: 2-4 Hours

Overview

You're in charge of designing a simple application for searching and retrieving data from the 'Nipper' statue repository. This data repository contains the title, artist, address and associated image of the various 'Nipper' the dog statues spread around the city of Albany.

Using any language and web framework (e.g. Node with Hapi), create an API for retrieving the statue data and use a front-end framework to display / filter / sort the results retrieved from the API (e.g. React.js, Angular, Vue.js).

The data can be found in the **statues.json** file provided with these instructions with the following format:

- **name:** the name of the statue
- **artist:** the full name of the statue's artist
- **address:** the address where the statue resides
- **path:** the path to the statue's image (in **.jpg**)

All associated imagery for the provided data may be found under the provided **assets** directory.

Requirements

- ☐ Design an interface to query your mock database (**statues.json**).
- ☐ Use a front-end framework to implement your user interface (React.js, Angular, Vue.js).
- ☐ Use a web framework in any language to write a REST API to serve your client-side code.
- ☐ Implement the ability to fuzzy search statues by **name** through your API.
- ☐ Implement the ability to sort the statues by **name, artist, and address** through your API.
- ☐ Create a collection view (e.g. a view of all the statues with their image and details).
- ☐ Create a single view (e.g. a view of an individual statue with their image and details).

Notes

- Be creative with your front-end design, aesthetics will only improve the presentation and will not count against.
- Practice clean code. The key to efficiency is maintaining a clean, organized codebase for others to contribute.
- Consider an alternative to a REST API such as GraphQL. This is a matter of preference.
- The requirements are considered the minimum criteria, there is no limit for additional features.
- Please document your code where applicable.

Submission

Please archive / compress your project's codebase into a file with the following format 'NAME_DATE' e.g. **John-Doe-1-12-18.tar.gz**

Include a **README.txt** or **README.md** that includes the requirements and guide to run your submission (submissions with instructions that are unclear or incorrect will not be considered).

Please e-mail your solution to the individual who sent the assignment.