GEOMARVEL

Develop a node.js REST API using the attached GeoJSON file

Data description:
The GeoJSON file contains all trees in the District of Columbia that are of the genus *Liquidambar*. Each tree feature contains properties related to that specific tree.

Task 1: Build the API
The API should:

- Read and serve data from the GeoJSON file
- Allow queries on the following two parameters: **condition**, **ward**
  - When queried on a param, the API should filter the data. For example: /getByParam?condition=Good should return all trees with condition good.
- Make use of at least two utility functions. (A utility function is a function not directly exposed via the API). For example, a utility function could parse parameters, order the data, etc.
- Have the following routes:
  - /getAll
  - /getByParam
- Be tracked via version control, via github or bitbucket (both have Free accounts)

Task 2: Build the Front End

- Build a React based front end that implements the API created above.
- The front end should allow the user to view the data as point graphics on an ArcGIS Webmap.
  - Setting up the Webmap can be tricky, the following guide does a good job of walking you through the process. We recommend using the Esri-Loader instead of the Webpack configuration.
    React | ArcGIS API for JavaScript 4.17
  - Note: the guide does not include a step. Add the following stylesheet to properly render the map : "https://js.arcgis.com/4.12/esri/themes/light/main.css"
- The ArcGIS for JS API has a lot of helpful information, the following link should help get you started. Intro to graphics | ArcGIS API for JavaScript 4.17

Task 3:
Extra/(Optional)

- Deploy the site and API to AWS, Heroku, or whatever other platform you are comfortable with.

How to submit:
Please submit a github or bitbucket repo with your API and documentation on how to spin up a local environment. If deployed to Heroku, please also include a link.