

React TS/JS Interview Challenge

Task: 1

Challenges: 1

Addons: 2

Task:

We want you to create a react website consisting of the following components:

1. Some kind of simple navigation
2. 3 pages each solving one of the challenges below

The app does not have to look pretty (although you will get extra points if it does) and serves more to test your knowledge on react components.

Furthermore, the implementation of the web app is subject to the following requirements:

1. Use functional components
2. Make use of react hooks in one form or another
3. Good state management
4. Components must be reusable to an appropriate extent
5. You are encouraged to use a ui kit such as [Material UI](#)
6. Every page must have a button that starts execution of the solution associated with that page, and the solution must be displayed below the button

File loading from a back-end is not required so you could simply just save each of the data files in a constant file and just import them when need be (this also helps with testing). You are also more than welcome to load them directly from the URLs.

Challenges:

The following are a few problems we want you to solve:

1. Write a function that performs some validity checks on two JSON files, [geo.json](#) and [data.json](#). The page must count the number of valid objects in the given files/input and display it to the user. Validation should be performed on the fields and values of the objects contained in the two json files mentioned above and follow the following schema:
 - a. data.json - {"**active**": **number**, "asn": **number**, "**countrycodestring**, "id": **number**, "statecode": **null/string**, "meta": **string**}
 - b. geo.json - { "ipv4" : **valid_ip_adress**, "geo" : **valid_lat_long** }

-
2. Write a function that takes in latitude and longitude as parameters, and returns a sorted list of the 10 rows from data.json with the shortest distance from the latitude and longitude, in a nicely formatted table.
 3. Write a function that takes two arrays of objects as parameters, and returns a single array containing the union of the two arrays. The input object can be found in [p3Data.json](#). The union is defined as follows:
 - a. All objects must have unique names in the final array
 - b. All other fields must be merged, ex: the union of **{name: "t", a:1}** and **{name: "t", b:2}** will give the object: **{name: "t", a:1, b:2}**
 - c. If two objects have the same field the values are summed
 - d. Objects with name as the only field must be ignored, i.e **{name: "obj1"}**

The array must be displayed in a table with each row containing the name of the object and a button to open a drop-down showing a list containing the individual fields and their values.

Addons:

put the repo in Github/Bitbucket for us to get there

JS is fine, TS is a plus.

Please note that we appreciate clean code & repos, following best practices.

Please feel free to ask questions about the task at any time.

Best of luck!