TRADECORE

Front-End test

Using React or Angular 1.5+, write an app that would display various tables depending on their availability on the server. You will be provided three JSON files that would represent three tables, their filenames would be their {tableName}, and you could add more JSON files to represent additional tables. The JSON files would be in format:

```
[{
      field1: value1,
      field2: value2,
      ...
},{
      field1: value1,
      field2: value2,
      ...
},{
      field1: value1,
      field2: value2,
      ...
},...]
```

These are the tasks that need to be completed:

1. You need to have a router in format `/table/{tableName:string}`. If i write an url for a json that i want to view, i would write its table name. Otherwise show 404 page. For example. if i have priceList.json available, i would write `/table/priceList` in the router to show that data.
2. Write a service for fetching data by tableName}
3. The cells in the table should be in this format:
    a. If it's an integer number, write it aligned to the right
    b. If it's a decimal number, write it with a dollar sign prefixed and sorted to the right
    c. If it's an url, write it inside an anchor
    d. If it's a date, use angular-moment library to filter it in relative time (from-now)
    e. If it's an object, write it as a top-down list with pairs `{subFieldName: value}`
    f. Any other is a simple string.

    Note that you don't have to understand the context of the data, just following the rules per value recognized is accepted (for example, if column name paid is not a decimal, but an integer, write by rule a., not b.)

4. Every column needs to have an input for filtering, and clicking on column title should sort the list.

5. The column title should be formatted in human readable manner, removing _ and first character should be uppercase.

You can use lodash for helper functions if you need it.

Some more rules to follow, if time allows:
1. Use bootstrap 3 for UI elements
2. For angular, use Jon Papa styleguide, otherwise for React use Airbnb style guide. This is not a strict rule, you can apply even a little just to make it a bit recognizable.
3. Use ES6+, babel and separate builds for production and development (production build needs to be uglified). You can choose any build system, as long as the project is initialized via NPM and all the tools are fetched via NPM (not cdn links).
4. Organize the architecture in components, and keep template and controllers for a component in the same folder.