

- 1) Create a back-end in goLang that let you add a todo and get the list of todos.
- 2) Create a front-end in react-native that let you add a todo and get the list of todos. The front-end should call the back end routes.
- 3) Given a string, write a function which returns a set of the most common characters (could be more than one).

For example:

In word “annoying”, the most common character is {‘n’}

In word “implementation”, the most common characters are {‘i’, ‘m’, ‘e’, ‘t’, ‘n’}

Optional tasks (non-obligatory)

Review the following code and create a list of improvements.

Feel free to directly refactor the code if you prefer.

```
///...

export const Service = {
  get: (url, token, params = {}, l = false) => {
    const headers = {
      Authorization: `Bearer ${token}`,
      Accept: 'application/json',
      'X-Request-ID': Math.floor(Math.random() * 1000000),
    };
    if (!token) headers.Authorization = ``;

    const query = {
      method: 'get',
      url,
      headers,
      ...(params ? { params } : {}),
    };

    const res = axios(query);

    // log the query and response
    if (l) {
      token = "*****"

      console.log('Query: ', query);

      if (res.status.toString().startsWith('2'))
```

```

        console.log('Response: ', res);
    //else if (res.status.toString().startsWith('5'))
    //  console.error('Response Internal server error: ', res);
    else
        console.error('Response: ', res);
    }

    return res;
},
post: (url, token, data, l = true) => {
    const headers = {
        Authorization: `Bearer ${token}`,
        Accept: 'application/json',
        'X-Request-ID': Math.floor(Math.random() * 1000000),
        'Content-Type': 'application/json',
    };
    if (!token) headers.Authorization = ``;

    const query = {
        method: 'post',
        url,
        headers,
        data,
    };

    const res = axios(query);

    // log the query and response
    if (l) {
        token = "*****"

        console.log('Query: ', query);

        if (res.status.toString().startsWith('2'))
            console.log('Response: ', res);
        //else if (res.status.toString().startsWith('5'))
        //  console.error('Response Internal server error: ', res);
        else
            console.error('Response: ', res);
    }

    return res;
},

```

```
};
```

- 4) Create a simple React app that will display a list of doctors.

The app should look like this:

<https://miro.com/app/board/uXjVPy-NMAI=#tpicker-content>

The design of the application is not important, however using TailwindCSS to add some basic style to the components is preferred.

You can use the following data as the list of doctors to display:

<https://63c15b437165626718791c6c.mockapi.io/challenge/api/doctors>