

React

Lesson 3

Code formatting

- Eslint (pre-configured in CRA) <https://eslint.org/docs/user-guide/getting-started>
- Prettier <https://prettier.io/docs/en/integrating-with-linters.html>

```
yarn add --dev prettier eslint-config-prettier eslint-plugin-prettier
```

```
"eslintConfig": {  
  "extends": [  
    ...  
    "plugin:prettier/recommended"  
  ]  
}
```

```
yarn eslint src --fix
```

React context

```
const MyContext = React.createContext(initialValue)
```

```
<MyContext.Provider value={currentValue}>
```

```
  <App />
```

```
</MyContext.Provider>
```

Inside App:

```
const value = React.useContext(MyContext)
```

Simple API server

- Install json-server via yarn:

```
yarn add json-server
```

- Create a db.json file with empty data for visits:

```
echo '{"visits": []}' > db.json
```

(not sure if works on Windows)

- Add db.json to .gitignore
- Run server on a custom port:

```
yarn json-server --watch db.json --port 3001
```

Sending requests

`fetch(url, {method, headers, ...})` -
returns Promise that resolves to Response

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

Processing response

```
fetch(...)  
  .then(response => response.json())  
  .then(data => doSomething(data))
```

New hooks

useEffect <https://reactjs.org/docs/hooks-effect.html>

useCallback <https://reactjs.org/docs/hooks-reference.html#usecallback>

Loading initial data

```
useEffect(( ) => {  
  loadData()  
}, [loadData])
```

```
// memoized function
```

```
const loadData = useCallback(( ) => {  
  // fetch data and add it to state  
}, [])
```


Home task

Set up json-server (or any other server with database of your choice).

Implement API methods (it's ok to copy api.ts from demo after understanding what it does)

Fetch and edit data with API.