0x06. React Immutable

Front-endJavaScriptES6React

By: Johann Kerbrat, Engineering Manager at Uber Works

Weight: 1

Project will start Aug 15, 2023 6:00 AM, must end by Aug 18, 2023 6:00 AM

Checker was released at Aug 16, 2023 12:00 AM

An auto review will be launched at the deadline

Resources

Read or watch:

Immutable Object concept

Immutable.js Documentation

Immutable.js GitHub

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone, without the help of Google:

Immutable objects. Who, what, when, where, and why?

How to use the Immutable.js library to bring immutability to Javascript

The differences between List and Map

How to use Merge, Concat, and Deep Merging

What a lazy Seq is

Requirements

Allowed editors: vi, vim, emacs, Visual Studio Code

A README.md file, at the root of the folder of the project, is mandatory

All of your files will be interpreted/compiled on Ubuntu 18.04 LTS using node 12.x.x and npm 6.x.x

All of your files should end with a new line

All of your functions must be exported

Setup

Install NodeJS 12.11.x

Install npm 6.11.x

The script below installs the above 2 at the specified versions

$ curl -sl https://deb.nodesource.com/setup\_12.x -o nodesource\_setup.sh

$ sudo bash nodesource\_setup.sh

$ sudo apt install nodejs

Provided files

babel.config.js

Click to show/hide file contents

module.exports = {

presets: [

[

'@babel/preset-env',

{

targets: {

node: 'current',

},

},

],

],

};

jsconfig.json

Click to show/hide file contents

{

"compilerOptions": {

"target": "es6"

},

"exclude": [

"node\_modules"

]

}

package.json

Click to show/hide file contents

{

"scripts": {

"lint": "./node\_modules/.bin/eslint",

"check-lint": "lint [0-9]\*.js",

"test": "jest",

"full-test": "./node\_modules/.bin/eslint [0-9]\*.js && jest"

},

"devDependencies": {

"@babel/core": "^7.6.0",

"@babel/preset-env": "^7.6.0",

"eslint": "^6.4.0",

"eslint-config-airbnb-base": "^14.0.0",

"eslint-plugin-import": "^2.18.2",

"eslint-plugin-jest": "^22.17.0",

"jest": "^24.9.0"

},

"dependencies": {

"immutable": "^4.0.0-rc.12"

}

}

Tasks

0. Converting into an Immutable object using fromJS

mandatory

Copy the necessary config files specified in the description to the root directory of the project and execute npm install.

In a file named 0-fromjs.js, create a function getImmutableObject that accepts object as a parameter and converts it into an immutable Map using fromJS of the Immutable.js library

Example:

{

fear: true,

smell: -1033575916.9145899,

wall: false,

thing: -914767132

}

Should return:

Map {

size: 4,

\_root: ArrayMapNode {

ownerID: OwnerID {},

entries: [ [Array], [Array], [Array], [Array] ]

},

\_\_ownerID: undefined,

\_\_hash: undefined,

\_\_altered: false

}

Repo:

GitHub repository: alx-react

Directory: 0x06-React\_Immutable

File: 0-fromjs.js

1. Converting into Immutable using Map

mandatory

In 1-map.js, modify the function getImmutableObject using Map from Immutable.js

Example:

{

fear: true,

smell: -1033575916.9145899,

wall: false,

thing: -914767132

}

Should return:

Map {

size: 4,

\_root: ArrayMapNode {

ownerID: OwnerID {},

entries: [ [Array], [Array], [Array], [Array] ]

},

\_\_ownerID: undefined,

\_\_hash: undefined,

\_\_altered: false

}

Repo:

GitHub repository: alx-react

Directory: 0x06-React\_Immutable

File: 1-map.js

2. Accessing nested elements

mandatory

Given the function below, edit it to return the value of the object at the defined path

export default function accessImmutableObject(object, array) {

}

The first argument is a plain object

The second one is an array containing a list of a path to some key in the object

The function should return the value of the object at the defined path

Example:

accessImmutableObject({

name: {

first: "Guillaume"

last: "Salva"

}

}, ['name', 'first'])

Should return Guillaume

Requirements:

The function should either return undefined, a string, or a Map

Repo:

GitHub repository: alx-react

Directory: 0x06-React\_Immutable

File: 2-nested.js

3. List and push

mandatory

In file 3-list.js, create these 2 functions:

export function getListObject(array) {

}

export function addElementToList(list, element) {

}

getListObject accepts an array as parameter and converts it into an immutable List using the Immutable.js library

addElementToList accepts two arguments: first one is a List and second one is a string

append the string to the list and return the list

Repo:

GitHub repository: alx-react

Directory: 0x06-React\_Immutable

File: 3-list.js

4. Chained mutations

mandatory

Create & export a constant named map. It should create an Immutable Map with the following object:

{

1: 'Liam',

2: 'Noah',

3: 'Elijah',

4: 'Oliver',

5: 'Jacob',

6: 'Lucas',

}

Export a second constant named map2. It should use the first map and modify the following values:

Modify the value for the index 2, to Benjamin

Modify the value for the index 4, to Oliver

Requirements:

You can’t use any other variable than map and map2

Repo:

GitHub repository: alx-react

Directory: 0x06-React\_Immutable

File: 4-mutations.js

5. Merge & concat

mandatory

Create a function named concatElements

It accepts two arguments page1 and page2. Both are arrays

It should return a List containing the values of the two pages

Create a function named mergeElements

It accepts two arguments page1 and page2. Both are objects

It should return a List containing the values of the two pages

If two values are the same, page2 values should be used.

Requirements:

Use list and map from the Immutable.js library

Repo:

GitHub repository: alx-react

Directory: 0x06-React\_Immutable

File: 5-merge.js

6. Nested merge

mandatory

Create a function named mergeDeeplyElements

It should accept two arguments, page1 and page2. Both are objects

It should return a List containing the values of the two pages

If two values are the same, they should combine each other

Example:

const page1 = {

'user-1': {

id: 1,

name: 'test',

likes: {

1: {

uid: 1234,

}

}

},

};

const page2 = {

'user-1': {

likes: {

2: {

uid: 134,

}

}

},

};

mergeDeeplyElements(page1, page2).toJS();

Should return:

const page1 = {

'user-1': {

id: 1,

name: 'test',

likes: {

1: {

uid: 1234,

},

2: {

uid: 134,

}

}

},

};

Requirements

use Map from immutable.js

Repo:

GitHub repository: alx-react

Directory: 0x06-React\_Immutable

File: 6-deeply.js

7. Equality

mandatory

Create a function named areMapsEqual

It accepts two arguments map1 and map2. Both are Immutable.js Maps

It should return true if the Maps are equal

Example:

const map1 = new Map(

{

firstName: 'Guillaume',

lastName: 'Salva',

}

);

const map2 = new Map(

{

firstName: 'Guillaume',

lastName: 'Salva',

}

);

areMapsEqual(map1, map2);

Should return true

Requirements:

Use is from the immutable.js library

Repo:

GitHub repository: alx-react

Directory: 0x06-React\_Immutable

File: 7-equality.js

8. Lazy Seq

mandatory

Create a function named printBestStudents:

It accepts one object as argument. The object is formed with the following structure:

const grades = {

1: {

score: 99,

firstName: 'guillaume',

lastName: 'salva',

}

};

Calling the function should filter any student with a score < 70 and print to the console the first name and the last name with the first letter capitalized:

{

"1": { score: 99, firstName: "Guillaume", lastName: "Salva" },

}

Requirements:

Uses seq from immutable.js

Repo:

GitHub repository: alx-react

Directory: 0x06-React\_Immutable

File: 8-seq.js