

0x06. React Immutable

Front-end JavaScript ES6 React



Weight: 1

Projects(/projects/current)



Project over - took place from Aug 13, 2024 6:00 AM to Aug 16, 2024 6:00 AM



An auto-review will automatically make corrections(/to_review)



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)



Curriculums(/dashboards/my_curriculums)



Concepts(/concepts)



Conference rooms(/dashboards/video_rooms)



Servers(/servers)



Sandboxes(/user_containers/current)



Tools(/dashboards/my_tools)



Video on demand(/dashboards/videos)

Resources

Read or watch:



- Immutable Object concept (/rltoken/C34VL1Db887N_zzjmpOU_w)
- Immutable.js Documentation (/rltoken/oDyg_FOCdEEcLfpNnSMWmg)
- Immutable.js GitHub (/rltoken/Za6PJtr1M59G7ktu5D_X3g)

Discord(<https://discord.com/app>)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/zWus2BDkrGHHKTIWuWbww), **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

QA Reviews I can make(/corrections/to_review)

- Install NodeJS 12.11.x
- Install npm 6.11.x

The script below installs the above 2 at the specified versions

```
$ curl -s https://raw.githubusercontent.com/nodesource/setup_12.x -o nodesource_setup.sh
$ sudo bash nodesource_setup.sh
$ sudo apt install nodejs
```

Concepts(/concepts)

Provided files

Conference rooms(/dashboards/video_rooms)

`babel.config.js`

Servers(/servers)
Click to show/hide file contents

`jsconfig.json`

Sandboxes(/user_containers/current)

Click to show/hide file contents

Tools(/dashboards/my_tools)
`package.json`

Click to show/hide file contents
Video on demand(/dashboards/videos)

Tasks

Peers(/users/peers)

Converting into an Immutable object using fromJS

Discord(<https://discord.com/app>)

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
}
```



My Planning(/planning/me)



Should return:



```
Map { Projects(/projects/current)
  size: 4,
  _root: ArrayMapNode {
    ownerID: undefined,
    ownerIDMake(/corrections/to_review)
    entries: [ [Array], [Array], [Array], [Array] ]
  },
  Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
  _hash: undefined,
  _altered: false
}
```



Curriculums(/dashboards/my_curriculums)

Repo:



- Concepts(/concepts)
- GitHub repository: alx-react
- Directory: 0x06-React_Immutable
- File: 0-from-is.js
- Conference rooms(/dashboards/video_rooms)



Check submission



Get a sandbox

Servers(/servers)

1. Converting into Immutable using Map

mandatory



Sandboxes(/user_containers/current)

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



Example: Tools(/dashboards/my_tools)



```
{
  Video on demand(/dashboards/videos)
  fear: true,
  smell: -1033575916.9145899,
  wall: false,
  thing: -914767132
}
```



Peers(/users/peers)

Should return:



Discord(<https://discord.com/app>)



My Profile(/users/my_profile)



```
Map {
  size: 4,
  (/) _root: ArrayMapNode {
    ownerID: OwnerID {},
    entries: [ [Array], [Array], [Array], [Array] ]
  },
  __ownerID: undefined,
  __hash: undefined,
  My Planning(/planning/me)
}
```

Projects(/projects/current)

Repo:



- QA Reviews I can make(/corrections/to_review)
- GitHub repository: alx-react
- Directory: 0x06-React_Immutable



- File: 1-map.js
- Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

Check submission



Curriculums(/dashboards/my_curriculums)

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) {
  Conference rooms(/dashboards/video_rooms)
```

- 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: Sandboxes(/user_containers/current)



```
accessImmutableObject({
  Tools(/dashboards/my_tools)
  name: {
    first: "Guillaume"
    last: "Salva"
  },
  Video on demand(/dashboards/videos)
}, ['name', 'first'])
```



Should return Guillaume



Peers(/users/peers)

Requirements:



- The function should either return undefined, a string, or a Map
- Discord(<https://discord.com/app>)

Repo:

- GitHub repository: alx-react
- Directory: 0x06-React_Immutable
- My Profile(/users/my_profile)



- File: 2-nested.js



(/)

Check submission



Home(/)

3. List and push

mandatory



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

My Planning(/planning/me)



```
export function getListObject(array) {  
  Projects(/projects/current)
```



```
export function addElementToList(list, element) {  
  QA Reviews I can make(/corrections/to_review)
```



- `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



Curriculums(/dashboards/my_curriculums)

Repo:



- `Concepts(/concepts)` - react
- Directory: `0x06-React_Immutable`



- File: 3-List.js
- Conference rooms(/dashboards/video_rooms)



Check submission

Servers(/servers)



Sandboxes(/user_containers/current)

4. Chained mutations

mandatory



Tools(/dashboards/my_tools)

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



```
{  
  Video on demand(/dashboards/videos)  
    1: 'Liam',  
    2: 'Noah',  
    3: 'Elijah',  
    4: 'Oliver',  
    5: 'Jacob',  
    6: 'Lucas',  
}
```



Peers(/users/peers)



Discord(<https://discord.com/app>)

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 (/users/my_profile)

- You can't use any other variable than `map` and `map2`



(/)

Repo:



- GitHub repository: `alx-react`
- Directory: `0x06-React_Immutable`
- File: `4-mutations.js`



My Planning(/planning/me)

Check submission



Projects(/projects/current)

5. Merge & concat

mandatory



QA Reviews I can make(/corrections/to_review)

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

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

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.

Curriculums(/dashboards/my_curriculums)



Concepts(/concepts)

- Use `list` and `map` from the `Immutable.js` library

Requirements



Conference rooms(/dashboards/video_rooms)

Repo:



Servers(/servers)

- GitHub repository: `alx-react`
- Directory: `0x06-React_Immutable`
- File: `5-merge.js`

Sandboxes(/user_containers/current)



Check submission



Tools(/dashboards/my_tools)



Video on demand(/dashboards/videos)

6. Nested merge

mandatory



Create a function named `mergeDeeplyElements`

Peers(/users/peers)

- 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

Discord(<https://discord.com/app>)

Example:



My Profile(/users/my_profile)

```

const page1 = {
  'user-1': {
    (/ id: 1,
      name: 'test',
      likes: {
        Home(/
          uid: 1234,
        }
      } My Planning(/planning/me)
    },
  },
};
Projects(/projects/current)
const page2 = {
  'user-1': {
    ✓ QA Reviews I can make(/corrections/to_review)
      2: {
        uid: 134,
      }
    ? Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
  },
};
Curriculums(/dashboards/my_curriculums);
mergedElements(page1, page2).toJS();

```

Should return:
Concepts(/concepts)

```

const page1 = {
  'user-1': {
    id: 1,
    name: 'test',
    likes: {
      Servers(/servers)
      1: {
        uid: 1234,
      },
    } Sandboxes(/user_containers/current)
      2: {
        uid: 134,
      }
    } Tools(/dashboards/my_tools)
  },
};
Video on demand(/dashboards/videos)

```

Requirements

- use Map from immutable.js

Peers(/users/peers)

Repo:

- Discord(<https://discord.com/app>)
- GitHub repository: atx-react
- Directory: 0x06-React_Immutable
- File: 6-deeply.js

Check submission

My Profile(/users/my_profile)



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:

```
My Planning(/planning/me)
const map1 = new Map(
  {
    Projects(/projects/current)
    lastName: 'Salva',
  }
);
QA Reviews I can make(/corrections/to_review)
const map2 = new Map(
  {
    Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
    lastName: 'Salva',
  }
);
Curriculums(/dashboards/my_curriculums)
areMapsEqual(map1, map2);
```

Should return true
Concepts(/concepts)

Requirements:

- Use `is` from the `immutable.js` library

Repo: Servers(/servers)

- GitHub repository: `alx-react`
- Directory: `0x06-React-Immutable`
- File: `7-equality.js`

Tools(/dashboards/my_tools)
Check submission

Video on demand(/dashboards/videos)

mandatory

Create a function named `printBestStudents`:

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

```
const grades = {
  1: {
    Discord(https://discord.com/app)
    score: 99,
    firstName: 'guillaume',
    lastName: 'salva',
  }
};
My Profile(/users/my_profile)
```





- 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:



```
(/)\n{\n  "1": { score: 99, firstName: "Guillaume", lastName: "Salva" },\n  Home(/)\n}
```



Requirements:

- My Planning(/planning/me)
- Uses seq from immutable.js



Projects(/projects/current)

Repo:



- GitHub repository: alx-react
- QA Reviews I can make(/corrections/to_review)
- Directory: 0x06-React_Immutable
- File: 8-seq.js



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

Check submission



Curriculumums(/dashboards/my_curriculumums)



Concepts(/concepts)



Conference rooms(/dashboards/video_rooms)



Servers(/servers)



Sandboxes(/user_containers/current)



Tools(/dashboards/my_tools)



Video on demand(/dashboards/videos)



Peers(/users/peers)



Discord(<https://discord.com/app>)



My Profile(/users/my_profile)