0x08. React Redux reducer+selector



Weight: 1

Projects(/projects/current)

➡ Project over - took place from Aug 22, 2024 6:00 AM to Aug 29, 2024 6:00 AM

QA Reviews I can make(/corrections/to_review)



Resources

Read or Watch:

- Reducers (/rltoken/SzgQcaVZ6qtF1ccU-S2DiA)
- Selectors (/rltoken/m3ctiAA74QV6YYqZ8YBZTQ)
- Writing tests (/r/token/E5mFy6WxHnMflwxYhy2gzw)
 Peersi/users/peersi
- Immutable Map documentation (/rltoken/oeA22lgPb_GvU1nOzWoA3w)
- Normalizr (/rltoken/fmN8EIQtqvKbLVgJuRyM0Q)
- Normalizing State Shape (/rltn/ep/wCbecNeGJhMu3hu38S7RCw)

Learning Objectives

At the end of this project, you should be able to explain to anyone (/rltoken/e1Q_JKJmhjTNmRU9kO6GFw), without the help of Google: My Profile(/users/my_profile)

• The purpose of a reducer and the role it plays within your application





- Why a reducer should stay as pure as possible
- Why mutations should not happen within a reducer
- \uparrow The use of Immutable within the reducer
- The use of Normalizr within the app

 Selectors: what they are and when to use them Home(/)

Requirements



- My Planning(/planning/me)
 Allowed editors: vi, vim, emacs, Visual Studio Code
- All your files should end with a new line



- All your files will be interpreted/compiled on Ubuntu 18.04 LTS using node 12.x.x and npm 6.x.x
- A README.md file, at the root of the folder of the project, is mandatory



- PUSA SPEVIEYOU FRESMAKELOGAGESTICKES EQUISONIC BABELTC
- · All of your functions must be exported
- ? Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

Lasks Curriculums(/dashboards/my_curriculums)

0. Write a basic reducer

mandatory



Concepts(/concepts)

Reuse the latest dashboard project you worked on in the React course 0x08-



React_Redux_action_creator+normalizr Conference rooms(/dashboards/video_rooms)

Create the basic state

a file named reducers/uiReducer.js, define the initial state of the reducer for the UI: Servers(/servers)

- It should have one boolean property isNotificationDrawerVisible
- It should have one boolean property isUserLoggedIn
- Sandboxes(/user_containers/current) It should have one empty object_user

Create the reducer function

Tools(/dashboards/my_tools)

In the same file, import all the actions that you created in the file actions/uiActionTypes and create a reducer function named uiReducer:

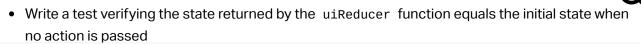


- Video on demand(/dashboards/videos)
 DISPLAY_NOTIFICATION_DRAWER should set isNotificationDrawerVisible to true
- HIDE_NOTIFICATION_DRAWER should set isNotificationDrawerVisible to false
- LOGIN_SUCCESS should set isUserLoggedIn to true
- LOGIN FAILURE should set isUserLoggedIn to false Peers(/users/peers)
- LOGOUT should set isUserLoggedIn to false

Vrite the tests

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

In a file named reducers/uiReducer.test.js, define the test suite for our simple reducer:



 Write a test verifying the state returned by the uiReducer function equals the initial state when the Marchiofile deusers mountain file bassed



 Write a test verifying the state returned by the uiReducer function, when the action DISPLAY_NOTIFICATION_DRAWER is passed, changes correctly the

(/) isNotificationDrawerVisible property

Tips:



Home(/)
Don't forget to set up the default case in your switch function

Requirements:



- My Planning(/planning/me)
 You should not mutate the state within the reducer
- You must use the spread operator to change the state



All the tests in the preject should pass

Depo: QA Reviews I can make(/corrections/to review)

GitHub repository: alx-react



Directory: 0x08-react_redux_reducer_selector Evaluation quizzes(/dashboards/my_current_evaluation_quizzes) File: task_0/dashboard/src/reducers/uiReducer.js,

task_0/dashboard/src/reducers/uiReducer.test.js



Curriculums(/dashboards/my_curriculums)

1. Use Immutable for the UI Reducer

mandatory

Concepts(/concepts)

Now that you have set up a basic reducer, let's reuse what we learned in the Immutable module and apply it to that reducer:



- Instantamento roems/dashinande/kidee_rooms)
- Update the uiReducer.js file to use Map from Immutable.js



- Update the different part of the reducer function to use set from Map Servers(/servers)
- Update the test suite, so it takes into account the changes

Tips:



Sandboxes(/user_containers/current)

- You can use the toJs function within your tests for an easy comparison
- Remember that Immutable.js always return a new Map after a modification

Requirements: Tools(/dashboards/my tools)



For better performances, do not use any from JS, to JS function within the reducer

All the tests in the project should pass

Repo:



- Peers(/users/peers)
 GitHub repository: alx-react
- Directory: 0x08-react_redux_reducer_selector



FIIB:staskuttleshtigesphiscock/appycers/uiReducer.js, task 1/dashboard/src/reducers/uiReducer.test.js

mandatory

Create a load action

In the courseActionTypes file, create a new action corresponding to when the API returns the list of Home(/) Courses. You can name it FETCH_COURSE_SUCCESS

Create the course reducer and default state

My Planning(/planning/me)

In a file courseReducer.js, write a reducer function. The default state should be an empty array.

Projects/(projects/current)

When the action creator sends the action FETCH_COURSE_SUCCESS, it also sends the list of courses in a data attribute. The action would look like:

```
data attribute. The action would look like:
          QA Reviews I can make(/corrections/to review)
   {
      type: FETCH_COURSE_SUCCESS,
      dat Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
        {
          id: 1,
          name: "ES6",
          Curriculums(/dashboards/my_curriculums)
},
        {
          id: 2
Concepts(/concepts)
name: "Webpack",
          credit: 20
        } 'Conference rooms(/dashboards/video_rooms)
           id: 3,
```

when updating the state of the reducer, you should also set the attribute isSelected to false for every lools (dashboards/my_tools) item in the list. The expected data from the reducer should be:

Video on demand(/dashboards/videos)

Sandboxes(/user_containers/current)

name: "React", Servers(/servers) Credit: 40

}

Peers(/users/peers)

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

Q

```
(/) id: 1,
        name: "ES6",
        isSelected: false,
        crteome(/)60
      },
        idMy Planning(/planning/me)
        name: "Webpack",
        isSelected: false,
        creojects/projects/current)
      },
        idQAReviews I can make(/corrections/to_review)
        name: "React",
        isSelected: false,
        ास्थ्रीर्धं atie quizzes (/dashboards/my_current_evaluation_quizzes)
      }
   1
efine the SELECT COURSE and UNSELECT COURSE actions Curriculums (/dashboards/my_curriculums)
 When the action creator sends the action SELECT_COURSE, it also sends an index corresponding to the
id of the course to update. The action would look like: Concepts(/concepts)
   {
      type: SELECT_COURSE
      Conference rooms(/dashboards/video_rooms)
   }
  Servers(/servers)
The expected data from the reducer should be:
          Sandboxes(/user_containers/current)
        id: 1,
        name: "ES6"
Tools(/dashboards/my_tools)
isSelected: false,
        credit: 60
Video on demand(/dashboards/videos)
        id: 2,
        name: "webpack",
        isSelected: true,
        c redirts (/users/peers)
      },
        <sup>i</sup>ூiseord(https://discord.com/app)
æ
        name: "React",
        isSelected: false,
        credit: 40
      }
   ]
          My Profile(/users/my_profile)
```

When the action creator sends the action UNSELECT_COURSE, it also sends an index corresponding to to update. The action would look like:

```
type: UNSELECT_COURSE,
indetkomæ(/)
```

he exp**ecyedanning oplanning ange**r should be:

```
ጭ[
         Projects(/projects/current)
       id: 1,
       name: "ES6",
       is Reviews I can smake (/corrections/to_review)
       credit: 60
     },
         Evaluation quizzes(/dashboards/my current evaluation quizzes)
       id: 2,
       name: "Webpack",
       isSelected: false,
       credit: 20
  Curriculums(/dashboards/my_curriculums)
B
     {
name (Concepts)
       isSelected: false,
       credit: 40
         Conference rooms(/dashboards/video_rooms)
```

Servers(/servers)

In a courseReducer.test.js, write a test suite for the new reducer. Define the following tests:

Sandboxes(/user_containers/current)

- Test that the default state returns an empty array
- Test that FETCH_COURSE_SUCCESS returns the data passed
- Test the right item updated
- Test that UNSELECT_COURSE returns the data with the right item updated

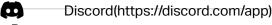
ps: Video on demand(/dashboards/videos)

· Use ES6 for this reducer, we can look at Immutable later

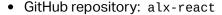
Requirements:



- Tr**Pters@ksethepteds**te of object as efficient as possible, for example you can use ES6 Map
 - · All the tests in the project should pass



Repo:



- Directory: 0x08-react_redux_reducer_selector
- File: task_2/dashboard/src/actions/courseActionTypes.js, tally_BrafileHusers/my_cprofilecers/courseReducer.js,



(/)

3. Create the reducer for notifications

mandatory



Home(/)

Create a load action

the notifications. You can name it FETCH_NOTIFICATIONS_SUCCESS

Greate the notifications reducer and default state Projects/(projects/current)

In a file notificationReducer.js, write a reducer function. The default state should be an object with:

- **✓**
- nota reading sawhigh will stare the distortions
- filter, which will be the attribute storing which filter is selected

Pefine the a LETCH NOTIFICATIONS SUCCESS action _evaluation_quizzes)

When the action creator sends the action FETCH_NOTIFICATIONS_SUCCESS, it also sends the list of notifications in a data attribute. The action would look like:

```
Curriculums(/dashboards/my_curriculums)
     type: FETCH_NOTIFICATIONS_SUCCESS,
     data: [
{ Concepts(/concepts)
          id: 1,
          type: "default",
         Contenence coms (Votas introduction rooms)
       },
        {
         Servers)
          type: "urgent",
          value: "New resume available"
        <sup>}</sup> /Sandboxes(/user_containers/current)
          id: 3,
         Tools(idashiyoards/my_tools)
value: "New data available"
Video on demand(/dashboards/videos)
```

When updating the state of the reducer, you should also set the attribute <code>isRead</code> to false for every item the list. The expected data from the reducer should be: Peers(/users/peers)

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

Q

```
filter: "DEFAULT",
      (M)otifications: [
           id: 1,
          Homeda: false,
           type: "default",
           value: "New course available"
        },My Planning(/planning/me)
        {
           id: 2,
          Pirspects (/pfajesets/current)
           type: "urgent",
           value: "New resume available"
        },QA Reviews I can make(/corrections/to_review)
         {
           id: 3,
          Exampadon foul 220s (/dashboards/my current evaluation quizzes)
           type: "urgent",
           value: "New data available"
        }
          Curriculums(/dashboards/my_curriculums)
Pefine the MARK AS_READ action Concepts(/concepts)
 When the action creator sends the action MARK_AS_READ, it also sends an index corresponding to the id
  of the notification to update. The action would look like:
Conference rooms(/dashboards/video_rooms)
      type: MARK_AS_READ,
      Servers(/servers)
   }
Sandboxes(/user_containers/current)
The expected data from the reducer should be:
          Tools(/dashboards/my_tools)
冊
          Video on demand(/dashboards/videos)
          Peers(/users/peers)
          Discord(https://discord.com/app)
```

My Profile(/users/my_profile)

```
filter: "DEFAULT",
      (M)otifications: [
           id: 1,
           Homeda: false,
           type: "default",
           value: "New course available"
         },My Planning(/planning/me)
         {
           id: 2,
           Pirspicates (/projects/current)
           type: "urgent",
           value: "New resume available"
         },QA Reviews I can make(/corrections/to_review)
         {
           id: 3,
           Exampadon foul 220s (/dashboards/my current evaluation quizzes)
           type: "urgent",
           value: "New data available"
         }
           Curriculums(/dashboards/my_curriculums)
 Pefine the SET TYPE FILTER action Concepts (/concepts)
  When the action creator sends the action SET_TYPE_FILTER, it also sends a filter attribute with either
  DEFAULT or URGENT . The action would look like:

Conference rooms(/dashboards/video_rooms)
      type: SET_TYPE_FILTER,
    Servers(/servers)
filter: URGENT
   }
Sandboxes(/user_containers/current)
The expected data from the reducer should be:
           Tools(/dashboards/my_tools)
H
           Video on demand(/dashboards/videos)
           Peers(/users/peers)
           Discord(https://discord.com/app)
```

My Profile(/users/my_profile)

```
filter: "URGENT",
(M)otifications: [
    id: 1,
    Homeda: false,
    type: "default",
    value: "New course available"
  } My Planning(/planning/me)
  {
    id: 2,
    Pirspects (/pfajesets/current)
    type: "urgent",
    value: "New resume available"
  } ,QA Reviews I can make(/corrections/to review)
  {
    id: 3,
    Evergadon for beaution guizzes (dashboards/my current evaluation quizzes)
    type: "urgent",
    value: "New data available"
  }
    Curriculums(/dashboards/my_curriculums)
```

Concepts(/concepts)

Use ES6 for this reducer, we can look at Immutable later

equirements:Conference rooms(/dashboards/video_rooms)

- Try to make the update of object as efficient as possible, for example you can use ES6 Map
- All the tests in the project should pass Servers(/servers)

Repo:

Sandboxes(/user_containers/current)

- GitHub repository: alx-react
- Directory: 0x08-react_redux_reducer_selector
- ע

冊

• File Qay dashbashds and services of services of the services

task_3/dashboard/src/reducers/notificationReducer.test.js Video on demand(/dashboards/videos)

4. Normalizr & Immutable

mandatory

Peers(/users/peers)

As you can see, updating a specific item in a collection is rather complicated and error prone. Using Normalizr is a good opportunity to simplify mutation Discord(https://discord.com/app)

Course schema

Create a new file schema/courses.js. In the file define a schema entity for courses. Create a function coursesNormalizer that would take data as argument and normalize the data with the schema you created

My Profile(/users/my_profile) In the course reducer function:

- Update the initial state to use an Immutable.js Map
- When FETCH_COURSE_SUCCESS action is called, normalize the data with the function you created (/) and merge it with the state
- When SELECT_COURSE or UNSELECT_COURSE is called, use the setIn function from Immutable to update the value of the item

Update the notification schema

the file schema/notifications is create a function notificationsNormalizer that would take My Planning(planning/me) data as argument and normalize it with the notification schema you created in the previous course.

pdate the notification reducer Projects(/projects/current)

In the notification reducer function:

- - When FETCH_NOTIFICATIONS_SUCCESS action is called, normalize the data with the function notificationsNormalizer you created and merge it with the state
- Whealustion quizzes (dashboatds/revitourettouvation quiztes) update the value of the state
 - When MARK_AS_READ, use the setIn function from Immutable to update the value of the item in the state

Land the test files/suites: Curriculums(/dashboards/my_curriculums)

• Update the course reducer test file to match the new reducer

Fips: Concepts(/concepts)

- You can use the fromJS function from Immutable.js to easily create the initial state from an
- Conference rooms(/dashboards/video_rooms)
 You can use the toJS function from Immutable. js to easily compare the expected data
- Selecting an unselecting a course item should only take one line now
- Marking a notification item as read should only take one line now

Requirements:

用

Although and an activities of the companies of the companies

gepo: Tools(/dashboards/my tools)

GitHub repository: alx-react

Directory: 0x08-react_redux_reducer_selector Video on demand(/dashboards/videos)
File: task_4/dashboard/src/schema/courses.js,

task_4/dashboard/src/reducers/courseReducer.js,

task_4/dashboard/src/schema/notifications.js,

task 4/dashboard/src/reducers/notificationReducer.js, Peers(Users/Deers) task_4/dashboard/src/reducers/courseReducer.test.js,

task_4/dashboard/src/reducers/notificationReducer.test.js

Discord(https://discord.com/app) æ

5. Selectors

mandator

Selectors are an efficient way to access the data from the state because a selector is not recomputed unless of of its arguments change)

Let's create a few selectors for the Notifications reducer in mrc/selectors/notificationSelector.js $^{
u}$ Create a first selector for the filter named <code>filterTypeSelected</code> , that will return the value of the • Creata emother selector for the notifications named getNotifications , that will return the list of notifications in a Map format • Create another selector for the notifications named getUnreadNotifications, that will return the Mistlanning goldening and in a Map format $Create\ a\ test\ suite\ for\ your\ selectors\ in\ a\ file\ named\ \ src/selectors/notificationSelector.test.js:$ Projects/(projects/current)
test that filterTypeSelected works as expected • test that getNotifications returns a list of the message entities within the reducer • tegt/het/រួម្ហាប់ក្រុងក្រុម្បានម្ចាប់ក្រុម្បានក្រុមប្រាក្សាខ្មែរប្រាក់ the message entities within the reducer Tips: • To Evaluation quists and dash bloomed strate war in ble visit in the reductors) you created. And pass the state to the selector functions You can also look into using Reselect for your own projects when you have advanced needs for filtering, reducing and merging data from the state Requirements: (dashboards/my_curriculums) All the tests in the project should pass 首 Concepts(/concepts) Repo: GitHub repository: alx-react Directory: 0x08-react_redux_reducer_selector • File task (5640sh) oard/src/selectors/notificationSelector.js, task_5/dashboard/src/selectors/notificationSelector.test.js **>**_ Sandboxes(/user_containers/current) Tools(/dashboards/my tools) Copyright @ 2024 ALX, All rights reserved. 用 Video on demand(/dashboards/videos) Peers(/users/peers) Discord(https://discord.com/app)