

Ox05. React state



Weight: 1

Projects(/projects/current)

Troject over - took place from Aug 7, 2024 6:00 AM to Aug 13, 2024 6:00 AM

An auto **CPA** i Previebres al upalme that the flow or lieue tions / to review)



Resources (my_tools)

Read or watch:



- Video on demand//dashboards/videos) State and lifecycle (/rltoken/rL2YOISKI2HIcenyNz5cqQ)
- SetState and State callback (/rltoken/AeohbCmE3m-m-xgABFWgIA)
- Context (/rltoken/SZ1mD7nEblEVUPrrN6SrRA)
- Forms and Controlled components (/rltoken/IBhcM3C1FOwY7uGYyNMclg)
- - Lifting state (外外形) en/iZ1dULJUZE85Lh0-yTSpYg)
 - React Hooks (/rltoken/4ISNYIQ67BkW53J7kGCccQ)
 - Enzyme State (/rltoken/rSRoY2jGlahlH8KkZDWK0w)
 - Enzyme Set State (hitoken) D9kg4M6VVxAga9-iJVgsYg)
 - Enzyme Instance (/rltoken/wqn34UANx7UE2nkolU-ntg)
 - Enzyme Simulate (/rltoken/GdM5eK75XXsl1EVqAJ4q5w)





earning Objectives

At the end of this project, you are expected to be able to explain to anyone (/ritoken/rSzjDoz08BNzJeOqmc1p_Q), without the help of Google:



- What they state of a component or a container is
- The lifecycle of a component
- How to modify a state and execute code in the right order



- What Blanning Wedborning onent is
- How to use Forms in React
- How to reuse smaller components, keep them pure, and lift its state to principal containers



- The give tell preject slowing and how to create one
- How to test State changes with Enzyme

QA_Reviews I can make(/corrections/to_review) Requirements



- All your files will be interpreted/compiled on Ubuntu 18.04 LTS using node 12.x.x and npm Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
 - Allowed editors: vi, vim, emacs, Visual Studio Code
 - · All your files should end with a new line



A README and file at the root of the folder of the project, is mandatory Curriculums (dashboards/my curriculums)



Concepts(/concepts)

asks Conference rooms(/dashboards/video_rooms)

0. Adding a local state for notifications

mandatory



Servers(/servers)

Using the previous project (0x05. React inline styling), we have modularized our React application without worrying about interactions and state, which is usually a recommended process of **ો**develop**જિંહાની મેજજ કર્ષ પશ્ચા**રાજો પશ્ચારાજો પશ્ચારાજો કરિયા મુક્કારી place to start adding logic and state.

Modify the App component in task_0/dashboard/src/App.App.js:



Reate a local state to store a displayDrawer element:



Define the default state for the state in the constructor of the Class

- Creide a qurid mand / Haah haad debide Pay Drawer that will change the value of displayDrawer to
 - Create a function named handleHideDrawer that will change the value of displayDrawer to false

Peers(/users/peers)

Modify the Notifications import in task_0/dashboard/src/App/App.js:



- Pass to the component the prop displayDrawer using the local state Discord(https://discord.com/app)
- Pass the new functions handleDisplayDrawer and handleHideDrawer

Modify the App test suite in task_0/dashboard/src/App/App.test.js:

- Add a test to verify that the default state for displayDrawer is false. Verify that after calling handleDisplayDrawer, the state should now be true
- My Profile(/users/my_profile)

 Add a test to verify that after calling handleHideDrawer, the state is updated to be false



Modify the Notifications component in task_0/dashboard/src/Notifications/Notifications.js:

- Define the propTypes and the defaultProps for the new props
- When clicking on Your notifications, call handleDisplayDrawer
- When clicking on the close button, call handleHideDrawer

Home(/)
At this point, after reloading the app, you should be able to show / hide the notifications panel

Modify the Notifications test suite in

ask_o.Mashabaing/splanning/fieations/Notifications.test.js:

Add a test to verify that clicking on the menu item calls handleDisplayDrawer



• Add a test to reject the last reject ing on the button calls handle Hide Drawer

Tips:



- Remember that you make (/cerrections/to-teview) entupdate. You will need to modify the logic to allow the component to rerender when the prop displayDrawer changes
- Use set State and instance when creating tests with Enzyme (Evaluation guizzes)
 - Remember to use spies to verify if a function is being called. You can pass a spy as a property

Requirements:



- Do not forget to bind the functions you are passing to the children to improve performances Curriculums(/dashboards/my_curriculums)
 When running, there should not be any lint error in the console

epo:

Concepts(/concepts)



• GitHub repository: alx-react



- Difeonfarensေနာက္ ရက္မွန္လု/dashboards/video_rooms)
- File: task_0/dashboard/src/App/App.js, task_0/dashboard/src/App/App.test.js,



task_0/dashboard/src/Notifications/Notifications.js, Servers(/servers)

task_0/dashboard/src/Notifications/Notifications.test.js

Sandboxes(/user_containers/current)

Check submission



🎤. Contrelled cehhpenehre _রমপ্রস্থার te callback

mandatory

reate a form within the Login component & handle login submit in Video on demand(/dashboards/videos) task_1/dashboard/src/Login/Login.js:

- Create a local state with the value isLoggedIn set to false
- Wrap the inputs within a form element



- Replace(hisebs/toens) an input element with type submit
- Create a function named handleLoginSubmit that will update the local state by setting



isLoggedIn to true Discord(https://discord.com/app) When the form is submitted call the newly created login submission handling function

create controlled components in the Login component in task_1/dashboard/src/Login/Login

- Modify the local state to add two new values email and password. By default these values are empty but not null
- Month of the work of the local state

 Create two function handleChangeEmail and handleChangePassword that the two inputs will call when the value in the input field is changed. The local state should update with what the user (/) is typing

modify state callback in task_1/dashboard/src/Login/Login.js

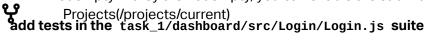


- Modify the local state to add one value enableSubmit . It should be set to false by default
- Modify the Submit button to only be enabled when the enableSubmit value of the local state is



TMP Planning(/planning/me)

 Every time the user changes the value of the email or password field, verify that both fields are not empty. If they are not empty, you can enable the submit button





 Add a test to verify that the submit button is disabled by default Add a Resign to the two inputs, the button is enabled





_ Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
To simulate an input change, you can use the simulate method and use the change event

Requirements:



- · The state should have all the default values set in the constructor
- Curriculums(/dashboards/my_curriculums)

 Do not forget to bind the functions you are passing to the children to improve performances
- When submitting the form, the page should not reload



- Make sure that the button is always enabled when the two inputs are not empty. And make sure Concepts/concepts that the button is always disabled when one of the input's value is empty
- When running, there should not be any lint error in the console



Conference rooms(/dashboards/video_rooms)

Repo:



• Giff PUNGENOSITUP(S) alx-react

Directory: 0x05-react_state



File: task_1/dashboard/src/Login/Login.js, task_1/dashboard/src/Login/Login.test.js Sandboxes(/user_containers/current)



Context (dashboards/my_tools)

mandatory

reate a simple React Context in task 2/dashboard/src/App/AppContext.js

- Define a default user object, containing email, password, and isLoggedIn
- Define a default logOut function



Create a React context containing a user object and a logout function. Set both default values using the previously created elements

Create the local state for the App in task_2/dashboard/src/App/App.js



Discord(https://discord.com/app) Modify the local state of the App container by adding a user object and logout function. Set the default (or empty) values using the ones you previously created for the context

Create the login and logout functions in the App container in task_2/dashboard/src/App/App.js and use them

My Profile(/users/my_profile)

- Create a logIn function that takes as argument email and password. When the function is called, update the user object in the local state. Set the email and password. Set the
- (/) isLoggedIn value to true
- Modify the logout function to reset the value of the user object in the local state
- Remove the two props isLoggedIn and logOut from the App, they will now be used through the state
- Refactor the render method to use the objects from the state to display the CourseList or the LMAY PLANMING (MADE TENING/Me)
- Pass the new logIn function to the Login component

ឃុំ task ဥ/dashhpardesrs/շերգյու/Login.js

- Remove from the state the isLoggedIn property, since we don't use it anymore
- Modify the handleLoginSubmit to call the newly created logIn function QA Reviews I can make (corrections/to review)

Checkpoint

Take a moving partion to be in a companient of the companient of t

- Enter a few letters to the email and the password inputs, and click on the submit button. You should see the list of courses and the login component should be unmounted.
- Verify that the Notifications panel is still working correctly. You should be able to show/hide the panerrisyulunakigashtrærds/nyhtgurkigulurha) close button.

Setting the context in task_2/dashboard/src/App/App.js:

M

- In the App Container, wrap the entire app with the AppContext created earlier using the provider element. Set the value to the user and logOut function using the local state
- **ું codify the**nfæræde **compoder**t hperalsk<u>vi</u>deasinoonsd/src/Header/Header.js:
 - Modify the Header to inherit the context using the ContextType API
 - Accessores (Intervente) and the header, that is only displayed when the value is Logged In in the user object within the context is true
 - This section should display "Welcome email (logout)" Sandboxes(/user_containers/current)

 Add the id_logoutSection_to the section
 - Clicking on the Logout link, should call the Logout function included within the context
- At this point, volume hour be able to login (remove Login component and show CourseList component) and logout within the map (remove CourseList component, show Login component, show new header Video on demand(/dashboards/videos)

Create the tests!

in task_2/dashboard/src/Header/Header.test.js

- Make syrethe chargest tests pass correctly
- · Add a test that mounts the Header component with a default context value. Verify that the logoutSection is not created
- Addiscress(International international component with a user defined (isLoggedIn is true and an email is set). Verify that the logoutSection is created
 - Add a test that mounts the Header component with a user defined (isLoggedIn is true and a email is set) and the logout is linked to a spy. Verify that clicking on the link is calling the spy

Refactor the previous tests to not use props but state instead

- Refactor the test checking if logout is being called by verifying if the state is updated correctly instead
- ?/) Create a test to verify that the logIn function updates the state correctly
- Create a test to verify that the logout function updates the state correctly



- Remember that function components can not use the context, you might need to convert the Header function component to a Class My Planning(/planning/me)
- Unfortunately, Enzyme does not fully support the newly created Static Context API of React. Therefore do not use the setContext API of Enzyme since it is not compatible. Instead wrap the comproper (!/ousipects/europett)xt.Provider to pass different context values

Requirements:

- BePaRerviewaticenteatralisation for the properties of the properties directly an object within the Provider value
- Don't forget to export elements that are going to be reused through the app (e.g default user) Evaluation quizzes (/dashboards/my, current evaluation quizzes)
 Don't fordet to clean unused state and props after refactoring
 - Don't forget to set the propTypes and defaultProps for any new prop

Sepo: Curriculums(/dashboards/my_curriculums)

• GitHub repository: alx-react

- - File: task_2/dashboard/src/App/AppContext.js, task_2/dashboard/src/App/App.js, task_2/dashboard/src/Login/Login.js, task_2/dashboard/src/Header/Header.js, taGRnforms/dashboards/video_reens)s, task_2/dashboard/src/App/App.test.js

Servers(/servers)

H

3. Context consumer & advanced state

mandatory

Sandboxes(/user_containers/current)
Context consumer: modify the Footer component in task_3/dashboard/src/Footer/Footer.js

• When the text Contact us

Modify the test suite in task_3/dashboard/src/Footer/Footer.test.js:

- Video on demand(/dashboards/videos) Refactor the test to make them work correctly with the changes
- Add a test to verify that the link is not displayed when the user is logged out within the context

Without making the component a Class, make the component subscribe to the context changes

· Add a test to verify that the link is displayed when the user is logged in within the context

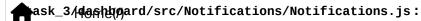
Leadvanced state: modifyethe App container in task_3/dashboard/src/App/App.js:

- Set the listNotifications within the state
- Create a function, markNotificationAsRead. It accepts an id (number) in argument. When the Discord(https://discord.com/app) function is called, it remove the notification with that id from the list of notifications within the
 - Pass the list of notifications to the Notifications component using the state
 - Pass the newly created function to the Notifications component

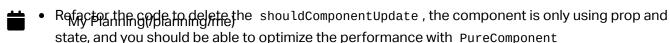
Modify the testisuits in task profeshboard/src/App/App.test.js:

Add a test to verify that markNotificationAsRead works as intended. You can for example set
the state with a mock list of notifications, then call the function and verify that the state of the
(/) container has been updated correctly

Modify the Notifications container in



• Refactor the code to delete the function markAsRead, we can now use the real one



• Use the newly created function markNotificationAsRead to mark a notification as read

Dennients (propies around default for markNotificationAsRead)

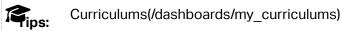
Modify the test suite in task_3/dashboard/src/Notifications/Notifications.test.js:

QA Reviews I can make(/corrections/to_review)
 Refactor the tests to match the new container

Checkpoint

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
Take a moment to test your application. At this point:

- · When you log in, you should be able to see a new footer
- When you display the panel of notifications, you should each of them disappear on click



• If you end up having to reuse mock data a lot, feel free to export and import the same mockup from the state of the content of the content

Requirements:



盲

- DGPPfergePerelema (Masebestete/AirPerdperate metactoring
- Don't forget to set the propTypes and defaultProps for any new prop

Servers(/servers)

Repo:

Gistandiscresitation adontaments/current)

Directory: 0x05-react_state



File: task_3/dashboard/src/Footer/Footer.js,
 Tools/dashboards/my tools)
task_3/dashboard/src/Footer.test.js, task_3/dashboard/src/App/App.js,
task_3/dashboard/src/App/App.test.js,

tasked dashboard (Arshboards Notifications test.js

4. Introduction to react hook

Peers(/users/peers)

Using React Hooks, modify the CourseListRow component in

ask_4/dashboard/src/CourseList/CourseListRow.js: Discord(https://discord.com/app)

Add a new style named rowChecked with the background color #e6e4e4

• When the row is a simple row, add a checkbox within the first cell

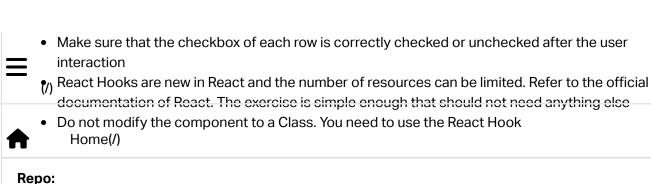
When the user checks the box, the styling of the row should use rowChecked

Tips & Requirements:

My Profile(/users/my_profile)

mandatory





My Planning(/planning/me) GitHub repository: alx-react

Directory: 0x05-react_state



Filerotasts//photostant/ent/courseList/CourseListRow.js



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

? Evaluation quizzes(/dashboards/my current evaluation quizzes)



Curriculums(/dashboards/my_curriculums)

Copyright @ 2024 ALX, All rights reserved.



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)