

0x03. React component

Front-end My JavaScript Learning JS6 React



Weight: 1

Projects(/projects/current)



Project over - took place from Jul 30, 2024 6:00 AM to Aug 2, 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)

Resources



Read or watch:

Tools(/dashboards/my_tools)

- React components (/rltoken/eHblf6Gksn6yyB63AxL95Q)
- React Developer Tools (/rltoken/ISa2LQiraDIOLg6hSp6GqA)



Video on demand(/dashboards/videos)

- Enzyme shallow render (/rltoken/R49r6v0_EZ5BwskKwOLg)
- Enzyme Mount (/rltoken/-cuX84klN3EWCuJNx-P74A)
- Enzyme Unmount (/rltoken/yQlfthRXFmRQHkNSDzMTag)



Peer-to-peer (/peer)

- React Pure components (/rltoken/JYDb2WZMLwBxM9qHjTVdpA)
- React Higher Order Components (/rltoken/AQ3a58-uRtO20Ysam9M6vw)
- Jest mock function (/rltoken/W0QL_4bEjld02Hr2-qQYrw)

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

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/w6DKIR6TVg7RkMN-ym12Kw), **without the help of Google**:

- When to use a Class or a function to create a component
- The difference between a Class component and a function component
- How to test a component





- How to utilize a Jest spy to verify that a function is being called correctly
- What an HOC is and how to use it
- How to optimize performance and control which components to render

Requirements



- All your files will be interpreted/compiled on Ubuntu 18.04 LTS using node 12.x.x and npm 6.x.x
- Allowed editors: vi, vim, emacs, Visual Studio Code
- All your files should end with a new line
- A README file at the root of the folder of the project, is mandatory



QA Reviews I can make(/corrections/to_review)

Tasks

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

0. Commence with class components

mandatory



Start this project with files from the the last task of the 0x03. React Props project: Curriculumis(/dashboards/my_curriculumis)

Convert the App function into a React Class:



- Modify the App function within App.js to convert the App function into a React class
- Make sure that the tests are still passing



Requirements:

Conference rooms(/dashboards/video_rooms)

- At this point, reloading the App should display the exact same page as the last task
- The console in your browser should not show any error or warning



Servers(/servers)

Repo:



Sandboxes(/user_containers/current)

- GitHub repository: alx-react
- Directory: 0x03-React_component
- File: Task_0/dashboards/my_tools/App/App.js



Tools(/dashboards/my_tools)



Video on demand(/dashboards/videos)

Check submission

Get a sandbox



Peers(/users/peers)

1. Lifecycles

mandatory



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

Add lifecycle to a component

In the App Class:

- Add a props named `logout` with the props type being `function`
- The default value for this property, should be an empty function
- Add an event listener when the component is mounted to listen to when the user is pressing down the keyboard keys



- When the user is pressing down the `control` and the `h` keys simultaneously, display the alert
Logging you out and call the function `logout`



Add the tests



In the test file for the App Class:

`Home(/)`

- Create a test to verify that when the keys `control` and `h` are pressed the `logout` function, passed as a prop, is called and the alert function is called with the string `Logging you out`



`My Planning(/planning/me)`

Requirements:



- Make sure to remove the event listener when the component is unmounted
- In the test file, make sure to restore the alert function you mocked
- At this point, reloading the App should display the exact same page as the last task
- The console in your browser should not show any error or warning



`QA Reviews I can make(/corrections/to_review)`



Repo:

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

- GitHub repository: `alx-react`
- Directory: `0x03 React_component`
- File: `task_1/dashboard/src/App/App.js`, `task_1/dashboard/src/App/App.test.js`



`Curriculumums(/dashboards/my_curriculumums)`

Check submission



`Concepts(/concepts)`

2. Handling Events

mandatory



`Conference rooms(/dashboards/video_rooms)`

Create a new handling event



In the Notifications component:

`Servers(/servers)`

- Convert the function into a React Class
- Make sure that the tests are still passing
- Create a new `markAsRead` function within the Notifications class. It accepts one argument: `id(number)`



`Sandboxes(/user_accounts/sandboxes)`



- When the function is called, it logs to the console the message `"Notification $id has been marked as read"`

`Tools(/dashboards/my_tools)`

- Pass the function `markAsRead` to the `NotificationItem` component as a property



the `Video on demand` class `(/dashboards/videos)`

- Modify the `li` element to call the new function `markAsRead` on click. It should send the id of the notification



- Define the new property type and the default property for the new properties

`Peers(/users/peers)`

Add the tests



In the Notifications test file:

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

- Create a test, that will mockup the `console.log` function
- Check that when calling the function `markAsRead` on an instance of the component, the spy is being called with the right message



In the `NotificationItem` test file:

`My Profile(/users/my_profile)`

- Create a test, that will pass a spy as the `markAsRead` property

- Check that when simulating a click on the component, the spy is called with the right ID argument



Requirements:

- (/)
 - Make sure to bind the function `markAsRead` in your constructor to avoid unnecessary re-rendering
- In the test file, make sure to restore the console function you mocked
- At this point, reloading the App should display the exact same page as the last task. Use the React Chrome Extension to make sure the Notifications component displays correctly
- The console in your browser should not show any error or warning



Home(/)



My Planning(/planning/me)



Repo:

Projects(/projects/current)

- GitHub repository: `alx-react`
- Directory: `0x03-React_component`
- ✓ File: `task_2/dashboard/src/Notifications/NotificationItem.js`,
`task_2/dashboard/src/Notifications/NotificationItem.test.js`,
`task_2/dashboard/src/Notifications/Notifications.js`,
`task_2/dashboard/src/Notifications/Notifications.test.js`
- ? Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)



6. Reusable components & specialization

mandatory



Containment

Concepts(/concepts)

Create a new component named `BodySection`. The component does not know its children. It should output the following:



Conference rooms(/dashboards/video_rooms)

- A div with the class `bodySection`
- Within the div, a `h2` element containing a title passed as a prop



Servers(/servers)

- Under the `h2` the children of `BodySection`

Tips:



Sandboxes(/user_containers/current)

Rendering the following



```
<BodySection title="test">
  <p>test</p>
</BodySection>
```



Video on demand(/dashboards/videos)

Should generate:

```
<div className="bodySection">
  <h2>test</h2>
  <p>test</p>
</div>
```



Peers(/users/peers)



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

Repo:

- GitHub repository: `alx-react`
- Directory: `0x03-React_component`
- File: `task_3/dashboard/src/BodySection/BodySection.js`
My Profile(/users/my_profile)



4. Specialization (/)

mandatory

in task_3/dashboard/src/BodySection/BodySectionWithMarginBottom.js , create a new component named BodySectionWithMarginBottom . The component does not know its children. It should output the following:

- A div with the class bodySectionWithMargin
- Within the div, a BodySection element with the same props passed to bodySectionWithMargin

task_3/dashboard/src/BodySection/BodySectionWithMarginBottom.css

- Set the style for the class bodySectionWithMargin with a margin bottom of 40px
- Import the styling into the BodySectionWithMarginBottom component

Requirements:

- Make sure to define the propTypes for both props
- Do not repeat each props manually and use the spread operator

Repo:

- Curriculum (/dashboards/my_curriculums)
- GitHub repository: alx-react
- Directory: 0x03-React_component
- File: task_3/dashboard/src/BodySection/BodySectionWithMarginBottom.js, task_3/dashboard/src/BodySection/BodySection.css
- Conference rooms (/dashboards/video_rooms)

Servers (/servers)

5. Use the new components

mandatory

Sandboxes (/user_containers/current)

in task_3/dashboard/src/App/App.js , modify the App component:

- Wrap the CourseList component with the newly created BodySectionWithMarginBottom component. The title should be Course list
- Wrap the Login component with the newly created BodySectionWithMarginBottom component. The title should be Log in to continue
- Using the BodySection component, add a new block with the title News from the School . The component should contain a paragraph with some random text

Peers (/users/peers)

Repo:

- GitHub repository: alx-react
- Directory: 0x03-React_component
- File: task_3/dashboard/src/App/App.js



My Profile (/users/my_profile)

6. Test the new components

mandatory

in `task_3/dashboard/src/BodySection/BodySection.test.js`:



- Add one test checking that shallowing the component should render correctly the children and one `h2` element

E.g. with the following code:



My Planning(/planning/me)

```
<BodySection title="test title">
```

```
<p>test children node</p>
```



```
</BodySection>
```

Projects(/projects/current)

You should check that:



QA Reviews I can make(/corrections/to_review)

- There is one `h2` element and it includes the text `test title`
- There is one `p` element and it includes the text `test children node`



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

in `task_3/dashboard/src/BodySection/BodySectionWithMarginBottom.test.js`:

- Add one test checking that shallowing the component should render correctly a `BodySection` component and that the props are passed correctly to the child component



Curriculumums(/dashboards/my_curriculumums)

Requirements:



- Make sure that the CSS is correctly applied to your component
- The console in your browser should not show any error or warning

Concepts(/concepts)



epo: Conference rooms(/dashboards/video_rooms)

- GitHub repository: `alx-react`



- Directories (/src/React_component)

- File: `task_3/dashboard/src/BodySection/BodySection.test.js`,
`task_3/dashboard/src/BodySection/BodySectionWithMarginBottom.test.js`
`Sandboxes(/user_containers/current)`



To With taskboards (tools)

7. Create WithLogging HOC

mandatory



We would like to add a way to log to the console every time a component has been mounted and every time it is about to unmount.

Video on demand(/dashboards/videos)

To not repeat the same code everywhere, create a HOC component in

`task_4/dashboard/src/HOC/WithLogging.js`:



Peers(/users/peers)

- The component should log to the console `Component NAME_OF_THE_WRAPPED_COMPONENT is mounted on componentDidMount()`



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

- The component should log to the console `Component NAME_OF_THE_WRAPPED_COMPONENT is going to unmount on componentWillUnmount()`

- Modify the `displayName` of the HOC to always display

`WithLogging(NAME_OF_THE_WRAPPED_COMPONENT)` in the React Chrome Extension or for debugging

- `NAME_OF_THE_WRAPPED_COMPONENT` should be the name of the wrapped component or class. If the wrapped element has no name it should default to `Component`

My Profile(/users/my_profile)



Repo:

- GitHub repository: alx-react
- Directory: 0x03-React_component
- File: task_4/dashboard/src/HOC/WithLogging.js



My Planning(/planning/me)

8. Write a test for the HOC

mandatory



Projects(/projects/current)



task_4/dashboard/src/HOC/WithLogging.test.js , write some tests for the HOC component:



- The first test should make sure console.log was called on mount and on unmount with the component when the wrapped element is pure html
- The second test should make sure console.log was called on mount and on unmount with the name of the component when the wrapped element is the Login component. Component Login is mounted and Component Login is going to unmount should be sent to the console



Tips:



- Writing a unit test for HOC can be difficult. Create a variable with the HOC wrapping a function rendering the React component or html. e.g. WithLogging(() => <p />)



Requirements:

Concepts(/concepts)

- Make sure that the messages Component Login is mounted and Component Login is going to unmount are sent when loading the app
- In the test file, make sure to restore the console function you mocked
- The console in your browser should not show any error or warning



Conference rooms(/dashboards/video_rooms)



Servers(/servers)

Repo:



- GitHub repository: alx-react
- Directory: 0x03-React_component
- File: task_4/dashboard/src/HOC/WithLogging.test.js



Tools(/dashboards/my_tools)



Video on demand(/dashboards/videos)

9. Declare a pure component

mandatory



Peers(/users/peers)



task_5/dashboard/src/Notifications/NotificationItem.js :



- Modify the component to make it "pure". Which means that it will only update when its props and state are different

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

Repo:

- GitHub repository: alx-react
- Directory: 0x03-React_component

My Profile(/users/my_profile)

- File: `task_5/dashboard/src/Notifications/NotificationItem.js`



(/)

10. Make your own pure component

mandatory



Home(/)

In `task_5/dashboard/src/Notifications/Notifications.js`:



- Modify the file so it only updates itself when the new property `listNotifications` has a longer list of elements than the previously
- You must implement the function `shouldComponentUpdate` to add this performance optimization



Projects(/projects/current)

Repo:



- QA Reviews I can make(/corrections/to_review)
- GitHub repository: `alx-react`
- Directory: `0x03-React_component`



- File: `task_5/dashboard/src/Notifications/Notifications.js`



Curriculums(/dashboards/my_curriculums)

11. Add a test

mandatory



Concepts(/concepts)

In `task_5/dashboard/src/Notifications/Notifications.test.js`, add two checks:



- The first check should verify that when updating the props of the component with the same list, the component doesn't rerender
- The second check should verify that when updating the props of the component with a longer list, the component does rerender



Servers(/servers)

Tips:



- Since the `NotificationItem` component is a function component, you can't directly use `React.PureComponent`



- Using the React Chrome Extension to make sure the `Notifications` component does not rerender will not work because the extension bypass `shouldComponentUpdate`. Use the test to verify your code instead



- You can use the function `dispatch` to change the props of the component

Requirements:

- The console in your browser should not show any error or warning



Peers(/users/peers)

Repo:



- Discord(<https://discord.com/app>)
- GitHub repository: `alx-react`
- Directory: `0x03-React_component`
- File: `task_5/dashboard/src/Notifications/Notifications.test.js`



My Profile(/users/my_profile)



(/)



Home(/)



My Planning(/planning/me)



Projects(/projects/current)



QA Reviews I can make(/corrections/to_review)



Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

Copyright © 2024 ALX, All rights reserved.



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)



Peers(/users/peers)



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



My Profile(/users/my_profile)