

# 0x02. React props

Front-end JavaScript ES6 React



Weight: 1

Projects(/projects/current)



Project over - took place from Jul 25, 2024 6:00 AM to Jul 30, 2024 6:00 AM



QA Reviews I can make(/corrections/to\_review)



Evaluation quizzes(/dashboards/my\_current\_evaluation\_quizzes)



Curriculums(/dashboards/my\_curriculums)



Concepts(/concepts)



Conference rooms(/dashboards/video\_rooms)



Servers(/servers)

## >Resources

Containers(/dashboards/user\_containers/current)

### Read or watch:



Tools(/dashboards/my\_tools)

- React Official Website (/rltoken/XY-3QaRU8GcLgridCcoi3w)
- Getting started with React (/rltoken/ASa6F8XX2jbBmfW0Ka4DtQ)



React Developer Tools (/rltoken/2vQpS-pQNQ474\_yCIKpi2w)

- Video on demand(/dashboards/videos)
- Enzyme (/rltoken/qsgGsN2WTikhObfGDALAEa)
- React Fragments (/rltoken/AnYmWGZ8hVMZ0sE8pjQoEg)

## Learning Objectives

Peers(/users/peers)

At the end of this project, you are expected to be able to explain to anyone

(/rltoken/PQrFhntTL54XShx3kANO4A) **without the help of Google:**  
Discord(https://discord.com/app)

- How to create basic React components using functions
- How to reuse components
- How to pass properties to components
- How to define types for components
- How to use Fragments
- When to use a key to improve a loop's performance

My Profile(/users/my\_profile)



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



My Planning(/planning/me)



Projects(/projects/current)

## Tasks



QA Reviews I can make(/corrections/to\_review)

### 0. Basic components

mandatory



Start with Evaluation quizzes (dashboards/my\_evaluation\_quizzes)

Instead of creating new elements, we're going to create components to split the project. The App.js is going to become the main entry point, the shell, for every component in the app.



### Create a Header component

Create a new folder Header :



- Move the code of the header from the App.js to a new file Header.js
- Move the css code, related to the header, of the App.css to a new file named Header.css
- Create an empty Header.test.js. We'll add tests for it later.



Conference rooms(dashboards/video\_rooms)

### Create a Footer component



Create a new folder Footer :



- Move the code of the footer from the App.js to a new file Footer.js
- Move the css code, related to the footer, of the App.css to a new file named Footer.css
- Create a Footer.test.js, we will add the tests later



### Create a Login component

Tools(dashboards/my\_tools)

Create a new folder Login :



- Move the code of the login section from the App.js to a new file Login.js
- The code should be wrapped in a React Fragment
- Move the css code, related to the login, of the App.css to a new file named Login.css
- Create a Login.test.js, we will add the tests later



### Modify the shell

Peers(/users/peers)

In the index.js :



- Discord(<https://discord.com/app>)
- Remove the root-notifications
- Remove the Notifications import

In the App.js :

- Import Notifications, Login, Footer, and Header
- Add the new Notifications component before the div.App element





- Wrap Notifications and the rest of the `div.App` within a React Fragment
- In the `div.App`, replace the header code by the new Header component
- In the `div.App-body`, replace the login code by the new Login component
- In the `div.App-footer`, replace the footer code by the new Footer component



the `App.test.js`:

- Most tests will fail, only keep the one testing that the component renders without crashing



My Planning(/planning/me)  
Requirements:

- 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



Projects(/projects/overview)



Repo: QA Reviews I can make(/corrections/to\_review)

- GitHub repository: `alx-react`
- Directory: `0x02-react-props`
- File: `task_0/dashboard/dist/index.html`, `task_0/dashboard/src/App/App.js`,  
`task_0/dashboard/src/Footer/Footer.css`, `task_0/dashboard/src/Footer/Footer.js`,  
`task_0/dashboard/src/Footer/Footer.test.js`,



Curriculum(/dashboard/my\_current\_courses)  
`task_0/dashboard/src/Header/Header.css`, `task_0/dashboard/src/Header/Header.js`,  
`task_0/dashboard/src/Header/Header.test.js`, `task_0/dashboard/src/Login/Login.css`,  
`task_0/dashboard/src/Login/Login.js`, `task_0/dashboard/src/Login/Login.test.js`



Concepts(/concepts)



Write the tests for each component (in `video_rooms`)

mandatory



To develop your tests faster, you can watch them. The test suite will run for every change you make:

- Add the following script to `task_1/package.json`: `"test-watch": "jest --watch"`
- Run your suite using `npm run test-watch`



Sandboxes(/user\_containers/current)

## Header.test.js



Import your new `Header` component within the test, and write two checks:

- Shallow render the `Header` component to verify it renders without crashing
- Verify that the components render `img` and `h1` tags



Video on demand(/dashboards/videos)

## Login.test.js

import your new `Login` component within the test, and write two checks:



- Shallow render the `Login` component to verify it renders without crashing
- Verify that the components renders 2 `input` tags and 2 `label` tags



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

Import your new `Footer` component within the test, and write two checks:

- Shallow render the `Footer` component to verify it renders without crashing
- Verify that the components at the very least render the text "Copyright"



## App.test.js

My Profile(/users/my\_profile)

Modify the `App.test.js` file, and add four checks:



- It should contain the `Notifications` component
- It should contain the `Header` component
- It should contain the `Login` component
- It should contain the `Footer` component



Requirements:



- At this point, running the test suites, should show:



```
Test Suites: 6 passed, 6 total
Tests:       18 passed, 18 total
```



QA Reviews I can make(/corrections/to\_review)

Repo:



- GitHub repository: `alx-react`
- Directory: `0x02-react_props`
- File: `task_1/package.json`, `task_1/dashboard/src/Header/Header.test.js`,  
`task_1/dashboard/src/Footer/Footer.test.js`,  
`task_1/dashboard/src/Login/Login.test.js`, `task_1/dashboard/src/App/App.test.js`



Curriculums(/dashboards/my\_curriculums)



## 2. Split the `Notifications` component

mandatory



### 2.1. Create a `NotificationItem.js`

The `Notifications` component is repeating the same tags a lot. It will be hard to maintain and reuse. Let's



create a `NotificationItem` component to support the `li` generation.

Create a new component named `NotificationItem`:



- it should render a `li` tag
- the component accept three properties (`type`, `html`, and `value`)
- `type` should be rendered with the `data-notification-type` attribute
- `value` should be rendered within the tag
- `html` should be rendered with the `dangerouslySetInnerHTML` attribute



Tools(/dashboards/my\_tools)



Video on demand(/dashboards/videos)

### Create a `NotificationItem.test.js`

Every component should be tested. Create a new `NotificationItem.test.js` file, and add the following checks:



- Verify that the basic rendering of the component works without crashing
- Verify that by passing dummy `type` and `value` props, it renders the correct html (for example:



`type="default"` and `value="test"`)

- Verify that by passing a dummy `html` prop, it renders the correct html (for example: `html={{ __html: '<u>test</u>' }}`)



## Modify `Notifications.js`

Import the new `NotificationItem` component in the `Notifications.js` file and replace the `<li>` tags with `NotificationItem` components.

My Profile(/users/my\_profile)

## Modify Notifications.test.js



Modify the `Notifications.test.js` file to change one of the check:

(/)

- Instead of testing if the component renders `li`, check that the component renders `NotificationItem` elements



Home(/)

- Add a test to verify that the first `NotificationItem` element renders the right `html` (note: this is usually not the best way to write tests, but sometimes necessary when you don't control the child



My Planning(/planning/me)

Requirements:



- 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
- Running the test suites, should show:



QA Reviews I can make(/corrections/to\_review)

Test Suites: 7 passed, 7 total

Tests: 22 passed, 22 total



Evaluation quizzes(/dashboards/my\_current\_evaluation\_quizzes)

Repo:



- GitHub repository: `alx-react`
- Directory: `0x02-react_props`



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`  
Conference rooms(/dashboards/video\_rooms)



Servers(/servers)

## 3. Checking the application using the React extension

mandatory



Using the React extension in Chrome:

Sandboxes(/user\_containers/current)

- Modify the type of the first `NotificationItem` to change from "default" to "urgent". The first notification should change color to red, take a screenshot
- Promote the load of the application and note which Component is the longest to render after App , take a screenshot



Tools(/dashboards/my\_tools)



Video on demand(/dashboards/videos)

Repo:

- GitHub repository: `alx-react`
- Directory: `0x02-react_props`
- File: `task_3/change_property.png`, `task_3/profiler.png`



Peers(/users/peers)



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



## 4. Props types & Default props & Shapes

mandatory

My Profile(/users/my\_profile)

To work on this task, install `prop-types` with npm first

## Create a CourseListRow component



Create a new folder `CourseList` and a new component with the name `CourseListRow`. The component accepts three props:



- `isHeader` (bool, default: false)
- `textFirstCell` (string, required)
- `textSecondCell` (string, default: null)



The component returns a `tr` HTML element, and within that element:



- if `isHeader` is true
  - if `textSecondCell` is null, it returns a `th` element containing `textFirstCell` with a `colSpan = 2`
  - if `textSecondCell` is not null, it returns two `th` elements containing `textFirstCell` and `textSecondCell`
- if `isHeader` is false
  - it returns two `td` elements containing `textFirstCell` and `textSecondCell`



Create a `CourseListRow.test.js` file:



- When `isHeader` is true
  - Add a check to test the component renders one cell with `colspan = 2` when `textSecondCell` does not exist
  - Add a check to test the component renders two cells when `textSecondCell` is present
- When `isHeader` is false
  - Add a check to test the component renders correctly two `td` elements within a `tr` element



## Create a CourseList component



In the folder `CourseList`, create a new component named `CourseList`:



- it returns a `table` element with an id `CourseList`
- it contains a `thead` element
  - it includes a `CourseListRow` with the props `textFirstCell="Available courses"` and `isHeader=true`



- it includes a `CourseListRow` with the props `textFirstCell="Course name"`, `textSecondCell="Credit"`, and `isHeader=true`



- it contains a `tbody` element
  - it includes a `CourseListRow` with the props `textFirstCell="ES6"`, `textSecondCell="60"`, and `isHeader=false`
  - it includes a `CourseListRow` with the props `textFirstCell="Webpack"`, `textSecondCell="20"`, and `isHeader=false`
  - it includes a `CourseListRow` with the props `textFirstCell="React"`, `textSecondCell="40"`, and `isHeader=false`



Create and import a file `CourseList.css` where you will be able to add the styling for the component

Create a `CourseList.test.js` with two checks:

- Check that it renders `CourseList` component without crashing
- Check that it renders the 5 different rows

## Modify the App component





Modify the App component to display either the Login, or the CourseList component:



- Import Proptypes
- Create a property `isLoggedIn`. It should be `false` by default



- When `isLoggedIn` is `false`, display the Login screen
- When `isLoggedIn` is `true`, display the CourseList screen

Modify the App test suite:



- Add a test to check that `CourseList` is not displayed
- Describe a new case, when `isLoggedIn` is `true`, and add two checks. The first one should verify that the `Login` component is not included. The second one should verify that the `CourseList` component is included



## Modify the NotificationItem component



QA Reviews I can make(/corrections/to\_review)

Define the prop types for the NotificationItem component:



- Verify that `html` is an object with the key `__html`, and that the value is a string
- Require `type` to be a defined prop and a string. It should default to `default` if `type` is not passed by the parent component
- Verify that `value` is a string



## Modify the Notifications component

Add a new `div` element with the class `menuItem`:



- it contains the correct notifications
- it should be display right before `div.Notifications`



Add a prop to the Notifications component named `displayDrawer`:



- it should be a boolean
- by default it should be `false`

Modify the `div.Notifications` tag:



- it should only be visible when `displayDrawer` is `true`

Modify the `Notifications.test.js`:



- Add a check that the menu item is being displayed when `displayDrawer` is `false`
- Add a check that the `div.Notifications` is not being displayed when `displayDrawer` is `false`



- Add a check that the menu item is being displayed when `displayDrawer` is `true`
- Add a check that the `div.Notifications` is being displayed when `displayDrawer` is `true`



## Update the UI

Peers(/users/peers)

You can use the React extension in Chrome, to toggle the two booleans `displayDrawer` and `isLoggedIn`



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

Notifications:

- Modify the CSS to make the UI looks like the image below when `displayDrawer` is `true`



- Modify the CSS to make the UI looks like the image below when `displayDrawer` is `false`

My Profile(/users/my\_profile)



App:  
(/)

- Modify the CSS to make the UI looks like the image below when `isLoggedIn` is `false`



Home(/)

- Modify the CSS to make the UI looks like the image below when `isLoggedIn` is `true`



My Planning(/planning/me)

Requirements:



- You should define a default property for every prop that is not required
- The console in your browser should not show any error or warning
- Running the test suites, should show:



QA Reviews I can make(/corrections/to\_review)

Test Suites: 5 passed, 5 total

Tests: 24 passed, 24 total



Evaluation quizzes(/dashboards/my\_current\_evaluation\_quizzes)

**Repo:**



- GitHub repository: `my-dashboard`

- Directory: `0x02-react_props`



- File: `task_4/dashboard/src/CourseList/CourseListRow.js`,  
`task_4/dashboard/src/CourseList/CourseListRow.test.js`,  
`task_4/dashboard/src/CourseList/CourseList.js`,



- `task_4/dashboard/src/CourseList/CourseList.css`,  
`task_4/dashboard/src/CourseList/CourseList.test.js`,



- `task_4/dashboard/src/App/App.js`, `task_4/dashboard/src/App/App.test.js`,  
`task_4/dashboard/src/Notifications/NotificationItem.js`,  
`task_4/dashboard/src/Notifications/Notifications.js`,



- `task_4/dashboard/src/Notifications/Notifications.test.js`  
Sandboxes(/user\_containers/current)



Tools(/dashboards/my\_tools)

mandatory



Video on demand(/dashboards/videos)

**Create the courses array**

Create a new Shape named `CourseShape` containing:



- `id` (number, required)
- `name` (string, required)
- `credit` (number, required)



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

- Add a propTypes `listCourses` that will take an array of the shape you created
- Make sure that if `listCourses` is not passed by the parent component the propTypes defaults to an empty array
- When `listCourses` is empty, display a row showing `No course available yet`

My Profile(/users/my\_profile)







- When `listCourses` contains at least one element, display a row for each element showing the name of the course and the number of credits
- Make sure to add a key to help React optimize rendering

Modify `CourseListRow.js` :



- Make sure the property `textSecondCell` can accept either a string or number

Modify the `App.js` to create a new array named `listCourses`. It should contains three elements:



- `My Planning(/planning/me)`
- id: 1, name: 'ES6', credit: 60
- id: 2, name: 'Webpack', credit: 20
- id: 3, name: 'React', credit: 40



`Projects(/projects/current)`

Add a `listCourses` property to the `CourseList` component in `App`



### Create the notifications array

Inside the `Notifications` directory, create a new Shape named `NotificationItemShape` containing:



- `id` (number, required)
- `html` (object of { `html`: string })
- `type` (string, required)
- `value` (string)



`Curriculum (/dashboards/my_curriculums)`

Modify `Notifications`:



- Add a prototype `listNotifications` that will take an array of the shape you just created
- When `listNotifications` is empty, display a row showing No new notification for now
- When `listNotifications` contains at least one element, display a `NotificationItem` for each



element of the array  
`Conference rooms (/dashboards/video_rooms)`

Modify the `App` to create a new array named `listNotifications`. It should contains all the previous elements we used to have in `Notifications` elements. Add an id for each element.



`Servers (/servers)`

Add the property `listNotifications` to the `Notifications` component in `App`



### Update the tests

`Sanitoboxes (/user_containers/current)`

Modify `Notifications.test.js` :



- Add a new test to verify that `Notifications` renders correctly if you pass an empty array or if you don't pass the `listNotifications` property



- Add a new test to verify that when you pass a list of notifications, the component renders it correctly and with the right number of `NotificationItem`
- Add a new test to verify that when `listNotifications` is empty the message Here is the list of notifications is not displayed, but No new notification for now is



Modify `Courses (/users/pairs)` :

- Add a new test to verify that `CourseList` renders correctly if you pass an empty array or if you don't pass the `listCourses` property
- Add a new test to verify that when you pass a list of courses, the component renders it correctly



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

Requirements:

- When writing your tests, look into using `describe` and `beforeEach` for each scenario (e.g. with `CourseList Empty`, `With CourseList containing elements`)
- You should define a default property for every prop that is not required

`My Profile (/users/my_profile)`



- The console in your browser should not show any error or warning
- Running the test suites, should show:



(/)

Test Suites: 9 passed, 9 total

Tests: 37 passed, 37 total



Home(/)



Repo:

My Planning(/planning/me)

- GitHub repository: alx-react



Directory: 0x02-react\_props

- Projects(/projects/current)

File: task\_5/dashboard/src/CourseList/CourseShape.js,  
task\_5/dashboard/src/CourseList/CourseList.js,



task\_5/dashboard/src/CourseList/CourseListRow.js,

QA-Reviews I can make/corrections/to\_review)

task\_5/dashboard/src/App/App.js,

task\_5/dashboard/src/Notifications/NotificationItemShape.js,



task\_5/dashboard/src/Notifications/NotificationItem.js,

task\_5/dashboard/src/Notifications/NotificationItem.test.js,

task\_5/dashboard/src/CourseList/CourseList.test.js



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)

Copyright © 2024 ALX, All rights reserved.



Peers(/users/peers)



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



My Profile(/users/my\_profile)