

Trainer guide

Examples and exercises (React)

React-5 – React Router basics

Contents

React-5a	Example: Simple static routing
React-5b	Exercise: Use React Router to set up static routing

React-5a | Example: Simple static routing

[Link to environment](#)

This example demonstrates how to set up simple static routing with React Router.

src/App.jsx

- *Navigate between the “About” and “Home” pages within the app.*
 - In the URL bar you can see the route changing.
- This is a single-page app, with the client-side routing achieved with React Router. This example demonstrates how React Router is used to set up simple static routing, like what you see in this app.
- The first thing to note is that React Router is not an official part of React. To use it, you need to make sure that the “react-router-dom” package is installed in your project. *(In the web environment, note that react-router-dom has been added to the dependencies).* If you are working in a local environment, then you can install the react-router-dom package via npm.
- Now, let’s go through the different tools provided by React Router that enable the routing on this page.
- The first one we’ll look at is the **Route** component *(lines 13 and 14)*
 - The Route component is how we define the different routes on our site. It takes two main props:
 - **path:** this is how we define the URL path for a specific route.
 - **element:** this is how we specify what component to render on the route specified by the path prop.
 - For example, on the “/about” route, we will display an h1 element that contains the text “About page”.
- The next important component is the **Routes** component.
 - All of your Routes must be contained within the Routes component.
- The next component we’ll look at is the **Link** component.
 - This is how we link between pages in our application. If we were to simply use HTML <a> (anchor) tags, then these would request a new page from the server, which is not the behaviour we want. Remember that in a single-page application we don’t send requests to the server every time we want to visit a different page, but instead the navigation is handled locally by the React app itself. The Link component facilitates this client-side navigation.
 - The Link component takes in a “to” prop that specifies which route the link should navigate to.
 - Note that our links have been placed **outside of any Routes**. This is because these elements are common to both pages in our application. We don’t want this content to be swapped out when we visit different pages.

- Finally, everything is wrapped inside the **BrowserRouter** component.
 - The BrowserRouter component is what enables routing in our application. For everything to work, all Routes and Links need to be contained inside of BrowserRouter.
- (*Line 2*) The BrowserRouter, Routes, Route and Link components must be imported from 'react-router-dom'.

React-5b | Exercise: Use React Router to set up static routing

[Link to environment](#)

This exercise gives learners practice on using React Router to set up static routing.

src/App.jsx

- In this exercise, you'll practice creating a similar routing set up as the one seen in the previous example.
- "Home" should navigate the user to the home page, and "About" should navigate the user to the about page.

Exercise

- The navigation bar will need to be changed so that the "Home" and "About" text are Link components.
- You'll also need to use React Router to set up the routing so that the base ("/") route points to the home page and the "/about" route points to the about page.

Solve the exercise live before moving on.

[Solution reference](#)