EdX and its Members use cookies and other tracking technologies for performance, analytics, and marketing purposes. By using this website, you accept this use. Learn more about these technologies in the Privacy Policy.



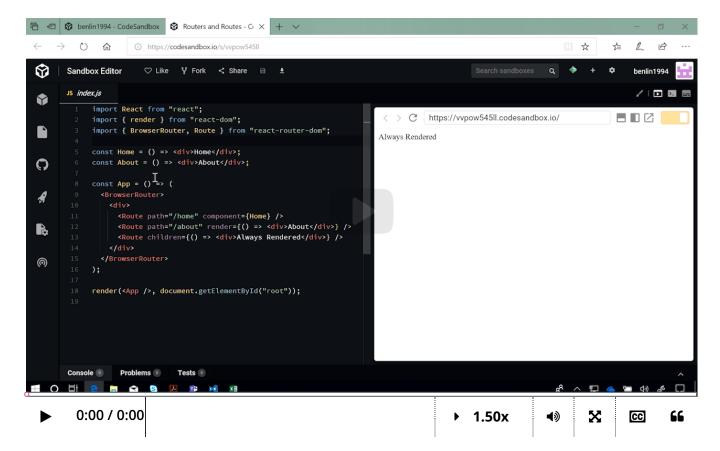


Course > React R... > Router... > Router...

## **Audit Access Expires Jun 20, 2020**

You lose all access to this course, including your progress, on Jun 20, 2020.

# **Routers and Routes**



Video

### Download video file

# **Transcripts**

Download SubRip (.srt) file **Download Text (.txt) file** 

### Routers and Routes

### **BrowserRouter**

The BrowserRouter component is a wrapper component that allows all of the other React Router components to work. The BrowserRouter component can only have one child element. There are other types of Router components but the BrowserRouter component is the main router we will be using in this course. We can import the BrowserRouter component with:

```
import { BrowserRouter} from "react-router-dom"
```

We can reference the BrowserRouter component using the <BrowserRouter> tag:

```
const BasicExample = () => (
  <BrowserRouter>
    <div> //A Router can only have one child element, so make sure e
      //other React Router components go here
    </div>
  </BrowserRouter>
)
```

#### Route

The Route component is probably the most important component out of the React Router library. It is used to render a specific component when a specified URL path occurs. We can import the Route component with:

```
import { Route } from "react-router-dom"
```

We can reference the Route component using the <Route> tag. There are three main ways we can render a component using the Route component. Take a look at them below:

```
//rendering using a component
<Route path="/about" component={About}/>
//rendering using an inline function
<Route path="/about" render={() => <div>About</div>}/>
//always rendering, regardless of whether the path matches the U
<Route path="/about" children={() => <div>Always rendered</div>}
```

All of the three render methods have a path property that is used to define the path that the component will render on. Be sure to use only one render method per Route component and also be sure to use the Route component inside a BrowserRouter component. Lastly, the Route component will pass a few route properties down to the rendered component, regardless of which method was used. We will cover all of these topics within this lesson.

Here's a live code example on CodeSandbox: <a href="https://codesandbox.io/s/vvpow545ll">https://codesandbox.io/s/vvpow545ll</a>

# **Path**

The path property is used to define the URL path that the component will render on. A Route component with no path property will always trigger a match and render. Matches are determined by doing a regex comparison between the URL path and the path property pattern. However, since it uses a regex match, it is possible for a match to occur even if the URL and the path pattern don't match exactly.

For example if the URL path is "/about/123" then both of the routes shown below will still render because "/about/123" contains both "/" and "/about":

```
const App = () => (
  <div>
    <BrowserRouter>
      < div >
        <Route path="/" render={(props) => <div>Home</div>}/> //A UR
        <Route path="/about" render={(props) => <div>About</div>}/>
      </div>
    </BrowserRouter>
  </div>
);
```

#### **Exact**

To account for this, be sure to add the exact property if you only want a route to render on an exact match. Trailing slashes on the URL don't factor in to the matching.

```
<Route exact path="/" render={(props) => <div>Home</div>}/> //
```

#### Strict

Use the strict property when you want trailing slashes to factor in to the matching.

```
<Route strict path="/about" render={(props) => <div>About</div</pre>
```

## Render methods

# Render method: component

The component property is used to render a component that is already defined. Do not try to provide an inline function to the component property, as this would inefficiently cause a new component to be created every time the route renders. If you want to use an inline function, provide it to the render property instead.

```
const About = (props) => (
  <div>
    About
  </div>
)
const App = () \Rightarrow (
  < div >
    <BrowserRouter>
      <Route path="/about" component = {About}/>
    </BrowserRouter>
  </div>
);
```

## Render method: render

The render property is used to render an inline function.

```
const App = () \Rightarrow (
  <div>
    <BrowserRouter>
      <Route path="/about" render={(props) => <div>About</div>}/>
    </BrowserRouter>
  </div>
);
```

#### Render method: children

The children property is used when you always need something rendered, regardless of whether or not the current URL matches the specified path. You may be wondering why we need to use a Route component to accomplish this since we can do the same thing using a normal component. The reason is because the Route component passes a few route properties down to the rendered function, which could in the component rendering logic.

```
const App = () \Rightarrow (
  <div>
    <BrowserRouter>
      <Route path="/about" children={(props) => <div>Always Rendered
    </BrowserRouter>
  </div>
);
```

© All Rights Reserved