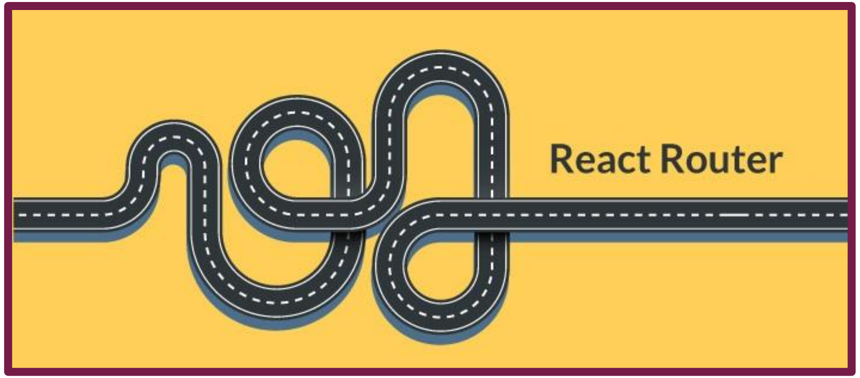
React Router v6

React Session-7

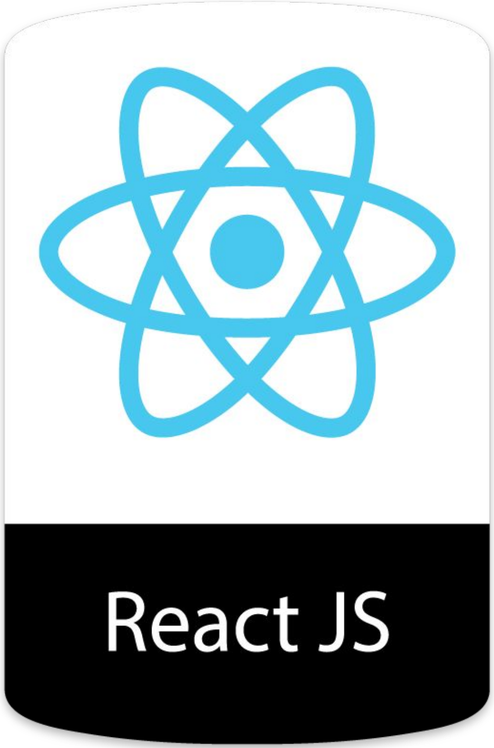


CLARUSWAY
WAY TO REINVENT YOURSELF

Table of Contents

- What is React Router?
- BrowserRouter
- Link
- Router
- Router Hooks

Did you finish React pre-class material?

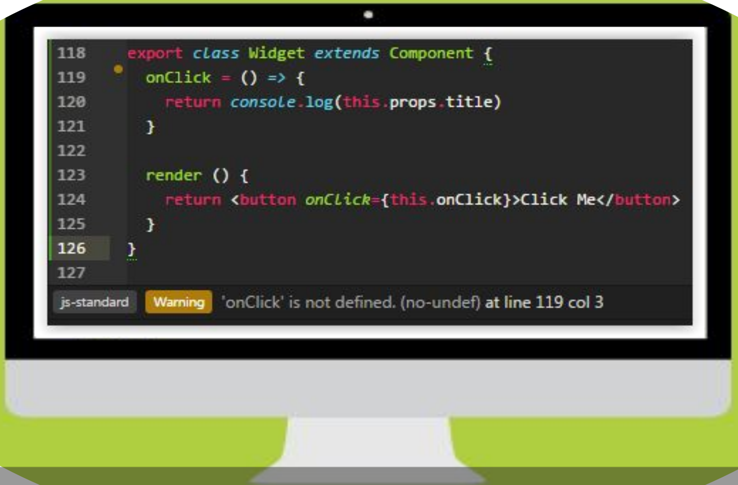


Kahoot!

1

What is React Router?

What is React Router?



```
118  export class Widget extends Component {
119    onClick = () => {
120      return console.log(this.props.title)
121    }
122
123    render () {
124      return <button onClick={this.onClick}>Click Me</button>
125    }
126  }
127
```

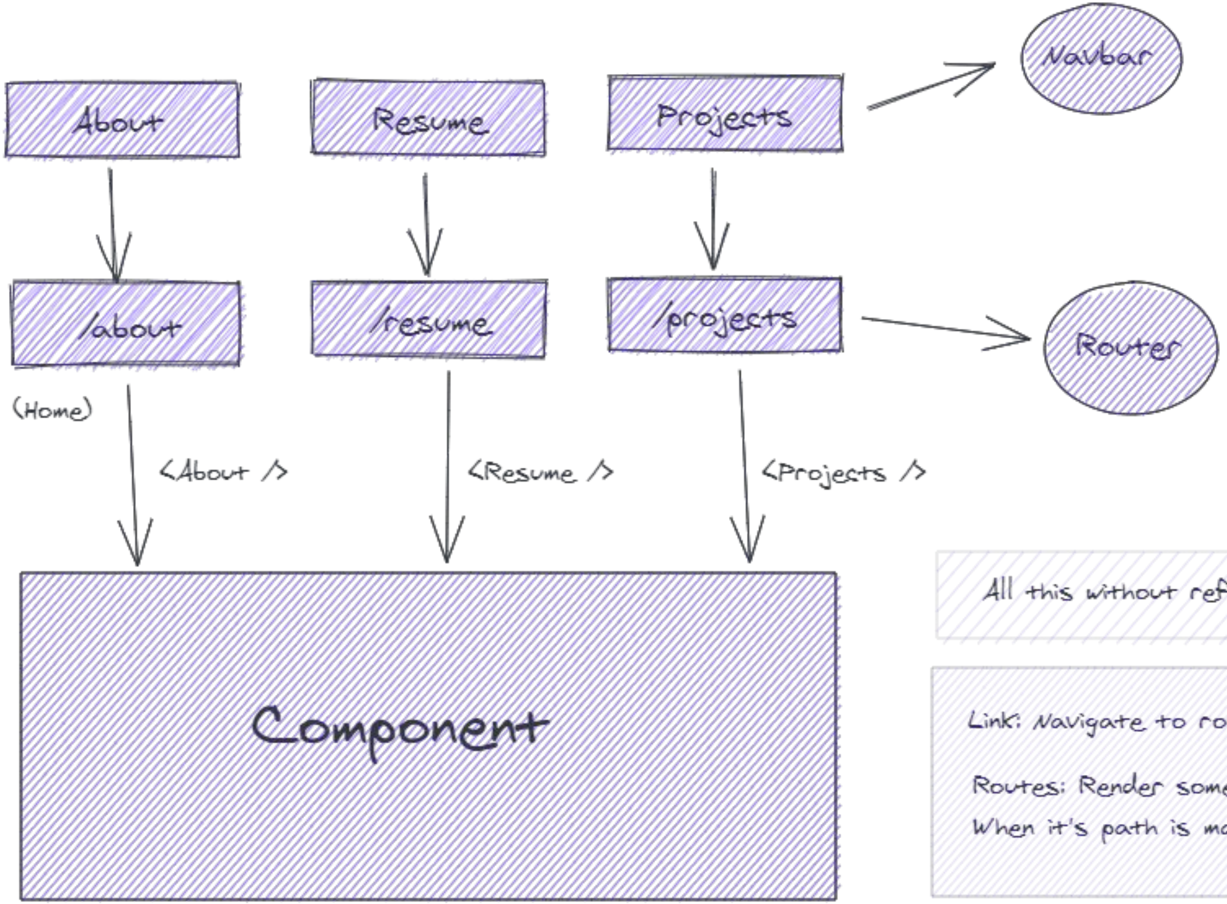
js-standard Warning 'onClick' is not defined. (no-undef) at line 119 col 3

What is React Router?

- As you may already know, by default, React comes without routing. And to enable it in our project, we need to add a library named react-router.
- React Router is the de-facto React routing library, and it’s one of the most popular projects built on top of React.
- React at its core is a very simple library, and it does not dictate anything about routing.
- Routing in a Single Page Application is the way to introduce some features to navigating the app through links, which are expected in normal web applications
- The browser should change the URL when you navigate to a different screen
- React Router offers a way to write your code so that it will show certain components of your app only if the route matches what you define.

```
www.clarusway.com/
www.clarusway.com/about/
www.clarusway.com/contact-us/
```

What is React Router?



All this without refresh

Link: Navigate to routes

Routes: Render some UI
When it's path is match

Installation

- With npm:

```
npm install react-router-dom
```

- With yarn:

```
yarn add react-router-dom
```

What is React Router?

The 3 components you will interact the most when working with React Router are:

`BrowserRouter`, usually aliased as `Router` wraps all your Route components.

`Link` components are - as you can imagine - used to generate links to your routes

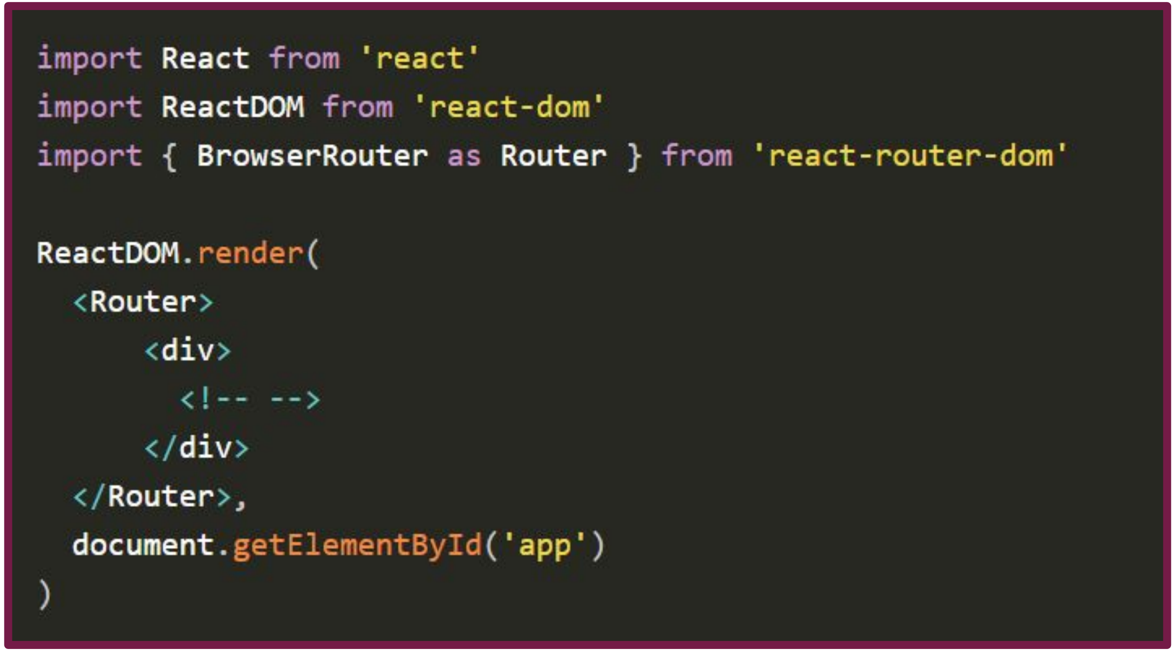
`Route` components are responsible for showing - or hiding - the components they contain.

2 BrowserRouter

BrowserRouter

- Here’s a simple example of the BrowserRouter component. You import it from react-router-dom, and you use it to wrap all your app:

- A BrowserRouter component can only have one child element, so we wrap all we’re going to add in a `div` element.



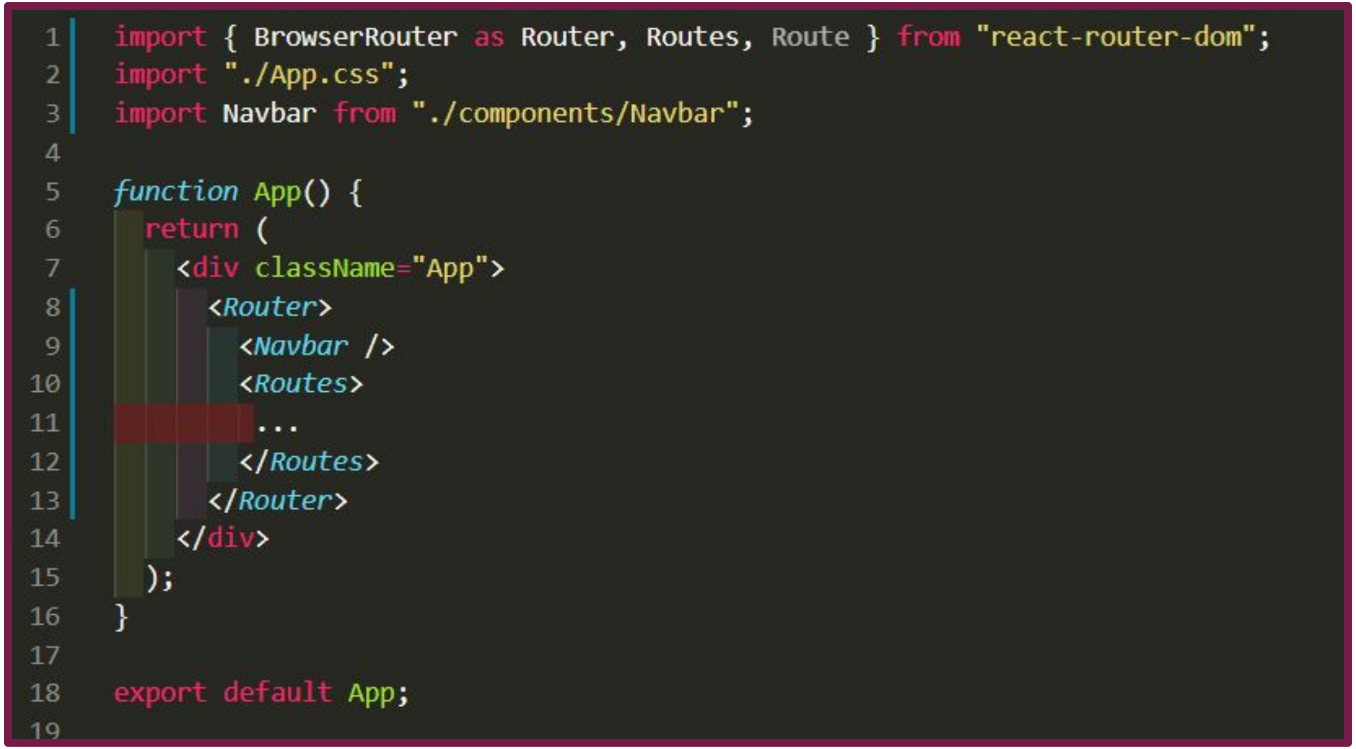
```
import React from 'react'
import ReactDOM from 'react-dom'
import { BrowserRouter as Router } from 'react-router-dom'

ReactDOM.render(
  <Router>
      <div>
        <!-- -->
      </div>
  </Router>,
  document.getElementById('app')
)
```

2

Routes

Routes

In our App component, add the **`Routes`** element (open and closing tags). These ensure that only one component is rendered at a time.



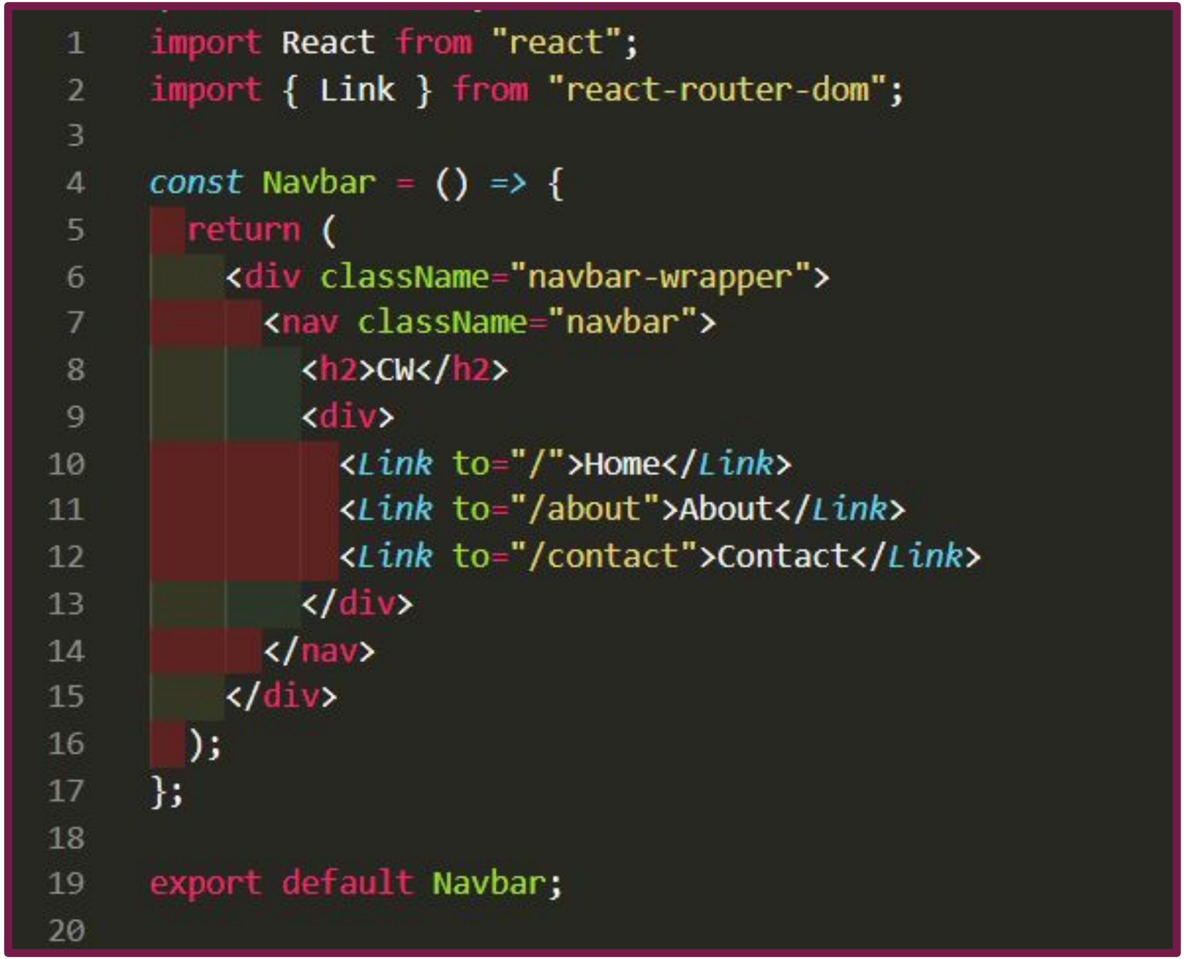
```
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import "./App.css";
import Navbar from "./components/Navbar";

function App() {
  return (
    <div className="App">
      <Router>
        <Navbar />
        <Routes>
          ...
        </Routes>
      </Router>
    </div>
  );
}

export default App;
```

3

Link

Link

- The Link component is used to trigger new routes. You import it from `react-router-dom`, and you can add the Link components to point at different routes, with the `to` attribute:



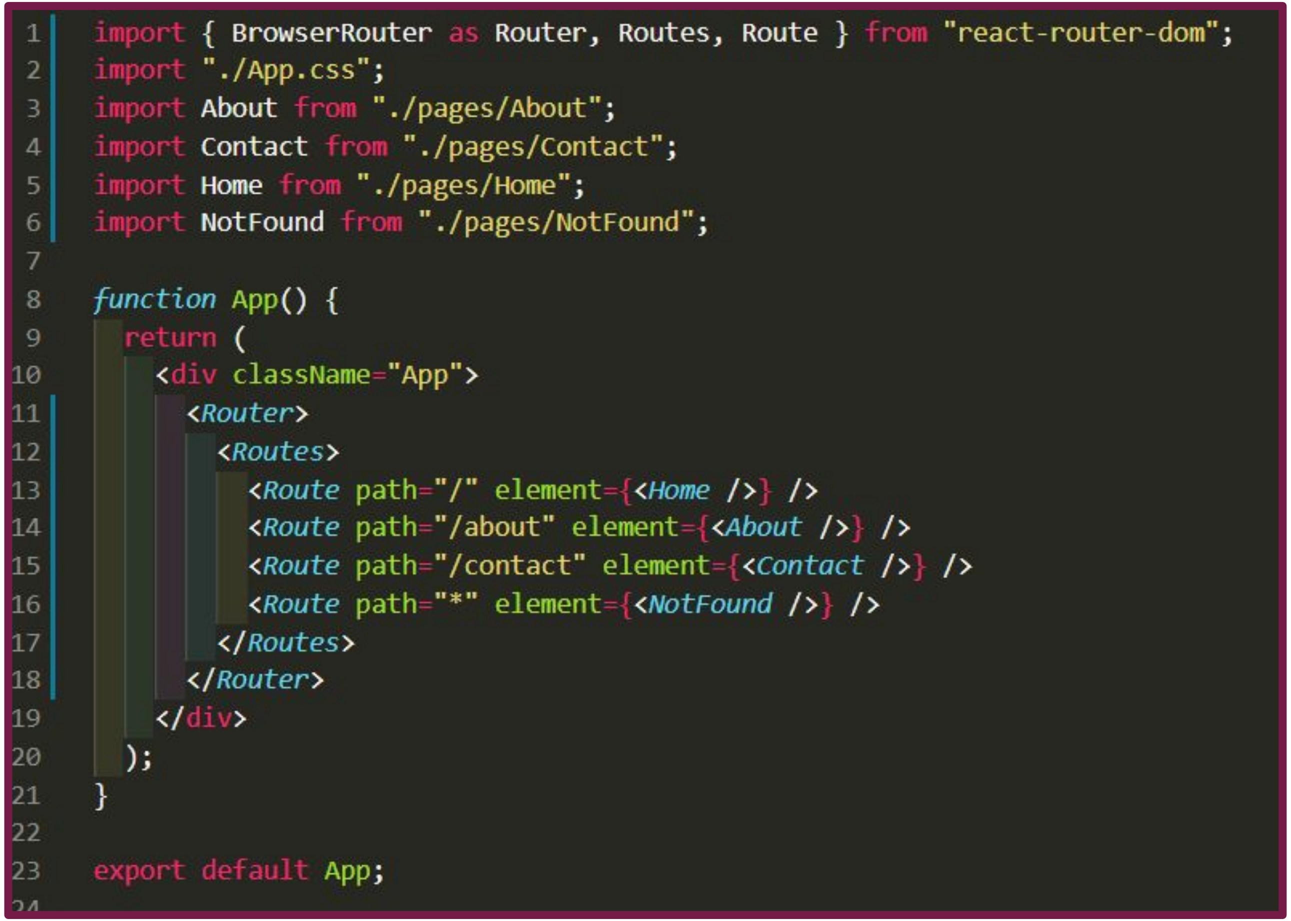
```
import React from "react";
import { Link } from "react-router-dom";

const Navbar = () => {
  return (
    <div className="navbar-wrapper">
      <nav className="navbar">
        <h2>CW</h2>
        <div>
          <Link to="/">Home</Link>
          <Link to="/about">About</Link>
          <Link to="/contact">Contact</Link>
        </div>
      </nav>
    </div>
  );
};

export default Navbar;

```

3 Route

Route

- Now let’s add the Route component in the above snippet to make things actually work as we want:

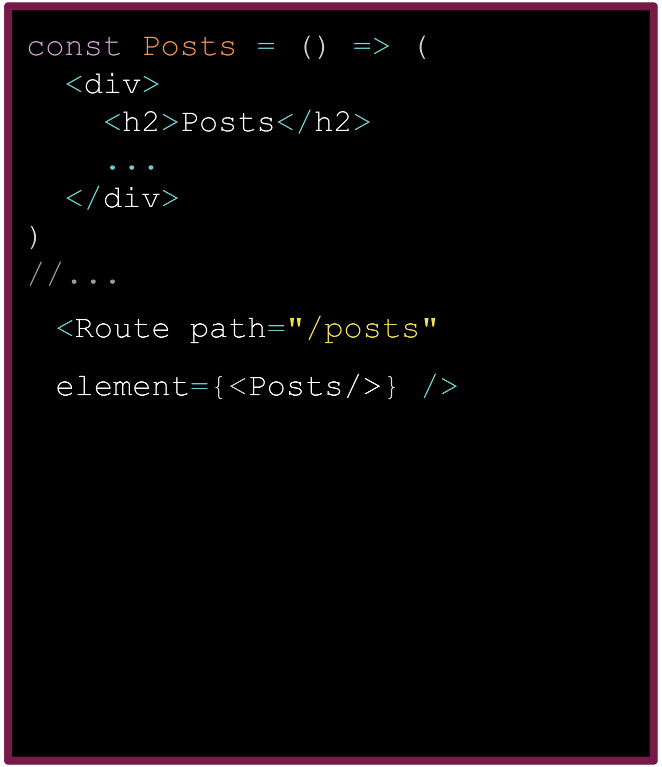
- Notice the `exact` attribute. Without this, `path="/"` would also match `/about`, since `/` is contained in the route.



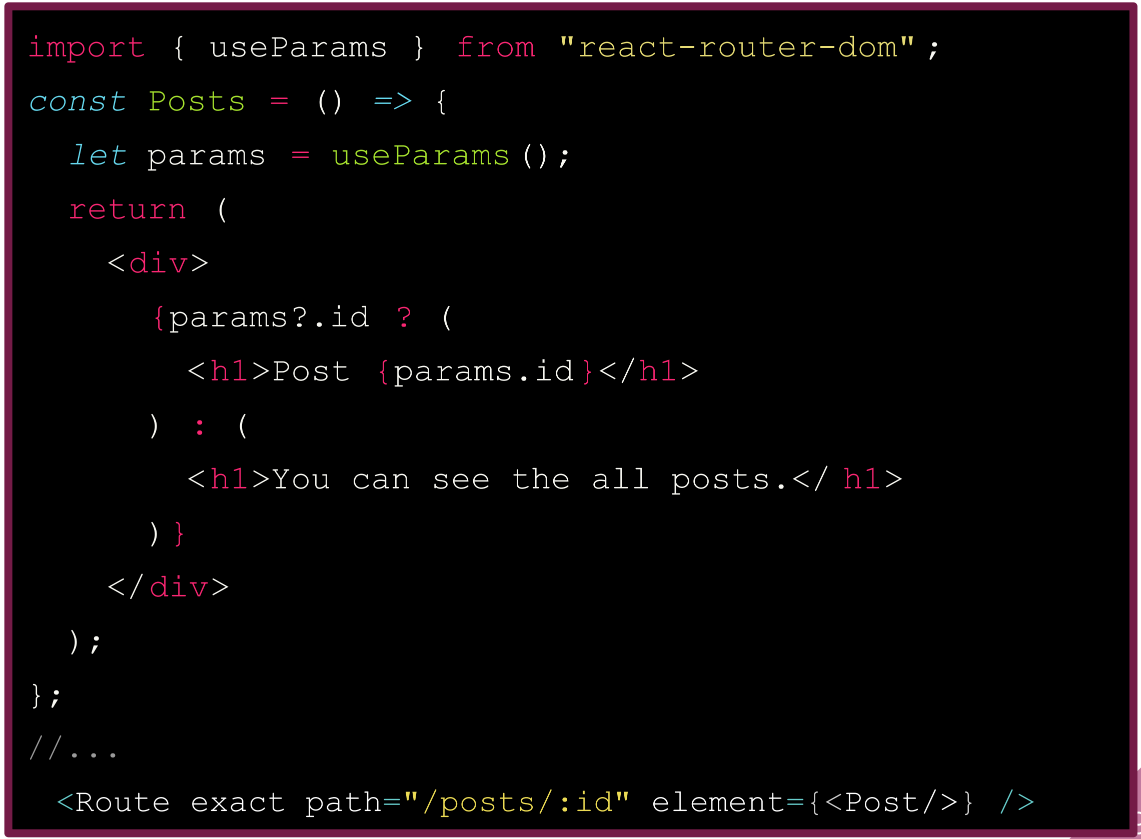
```
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import "./App.css";
import About from "./pages/About";
import Contact from "./pages/Contact";
import Home from "./pages/Home";
import NotFound from "./pages/NotFound";

function App() {
  return (
    <div className="App">
      <Router>
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/about" element={<About />} />
          <Route path="/contact" element={<Contact />} />
          <Route path="*" element={<NotFound />} />
        </Routes>
      </Router>
    </div>
  );
}

export default App;
```

Route

Match Dynamic Route Parameter

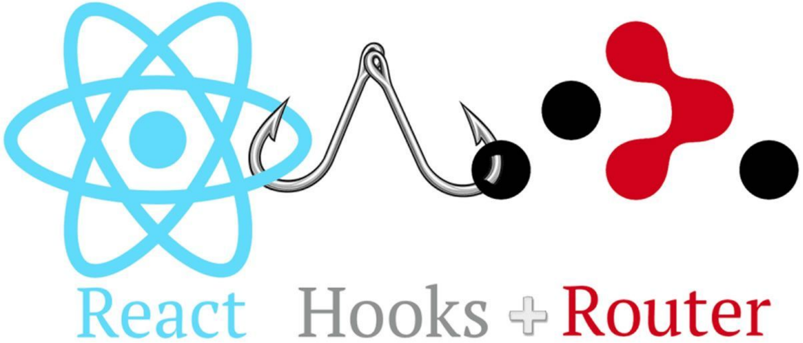


```
const Posts = () => (
  <div>
    <h2>Posts</h2>
    ...
  </div>
)
//...
 <Route path="/posts"

 element={<Posts/>} />
```



```
import { useParams } from "react-router-dom";
const Posts = () => {
  let params = useParams();
  return (
    <div>
      {params?.id ? (
        <h1>Post {params.id}</h1>
      ) : (
        <h1>You can see the all posts.</ h1>
      )}
    </div>
  );
};
//...
 <Route exact path="/posts/:id" element={<Post/>} />
```

3 Router Hooks



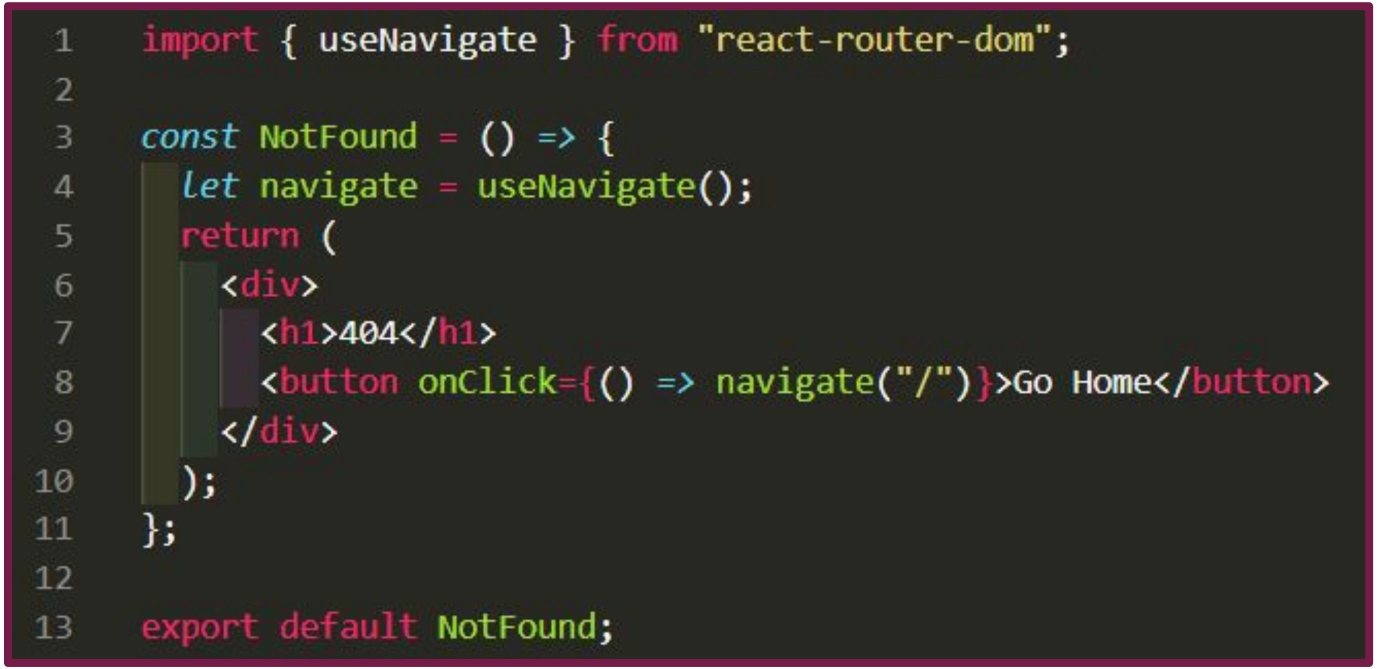
Router Hooks

- `useNavigate`
- `useParams`
- `useLocation`

Router hooks make things much easier. Now you can access the history, location, or parameters in an easy and elegant way.

Router Hooks - `useNavigate`

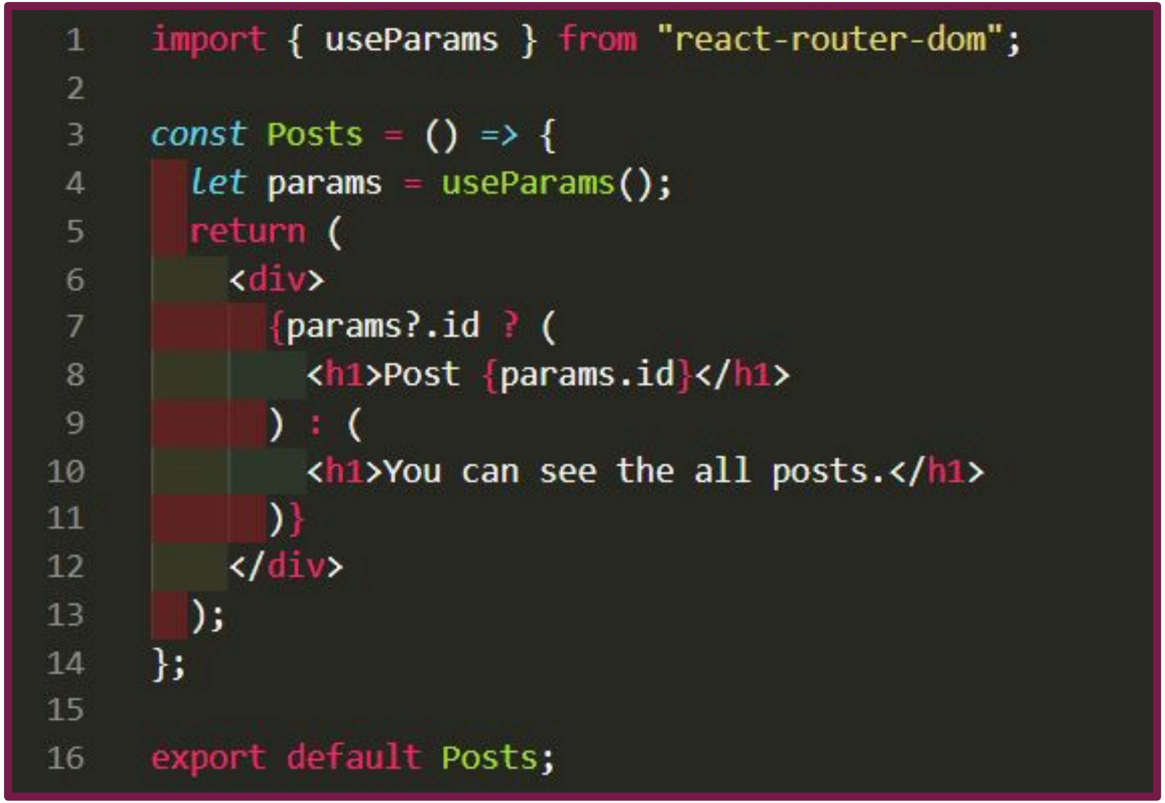
The **`useNavigate`** hook gives us access to the history instance without pulling it from props.



```
import { useNavigate } from "react-router-dom";

const NotFound = () => {
  let navigate = useNavigate();
  return (
    <div>
      <h1>404</h1>
      <button onClick={() => navigate("/")}>Go Home</button>
    </div>
  );
};

export default NotFound;
```

Router Hooks - `useParams`

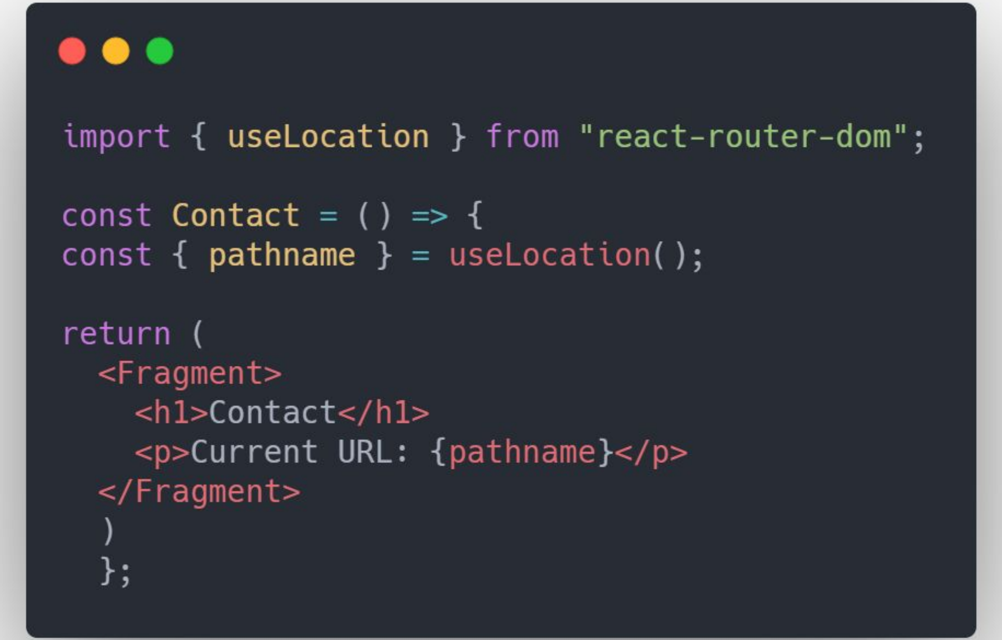
This hook helps us get the parameter passed on the URL without using the props object.



```jsx
import { useParams } from "react-router-dom";

const Posts = () => {
  let params = useParams();
  return (
    <div>
      {params?.id ? (
        <h1>Post {params.id}</h1>
      ) : (
        <h1>You can see the all posts.</h1>
      )}
    </div>
  );
};

export default Posts;
```

Router Hooks - `useLocation`

This hook returns the location object that represents the current URL.



```
import { useLocation } from "react-router-dom";

const Contact = () => {
const { pathname } = useLocation();

return (
  <Fragment>
    <h1>Contact</h1>
    <p>Current URL: {pathname}</p>
  </Fragment>
  )
  };
```

THANKS!

Any questions?