

REACT ROUTER

npm install react-router-dom@6

```
client > src > JS App.js > App
1 import logo from "./logo.svg";
2 import "./App.css";
3
4 import { Routes, Route } from "react-router-dom";
5 import Home from "./pages/Home";
6 import Login from "./pages/Login";
7 import Detail from "./pages/Detail";
8
9 function App() {
10   return (
11     <div className="App">
12       <header className="App-header">
13         <img src={logo} className="App-logo" alt="logo" />
14
15         <Routes>
16           <Route path="/" element={<Home />} />
17           <Route path="/login" element={<Login />} />
18           <Route path="/:id" element={<Detail />} />
19         </Routes>
20       </header>
21     </div>
22   );
23 }
24
25 export default App;
26
```

```
client > src > JS index.js > ...
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import "./index.css";
4 import App from "./App";
5 import reportWebVitals from "./reportWebVitals";
6
7 import { BrowserRouter } from "react-router-dom";
8
9 const root = ReactDOM.createRoot(document.getElementById("root"));
10 root.render(
11   <React.StrictMode>
12     <BrowserRouter>
13       | <App />
14     </BrowserRouter>
15   </React.StrictMode>
16 );
17
18 // If you want to start measuring performance in your app, pass
19 // to log results (for example: reportWebVitals(console.log))
20 // or send to an analytics endpoint. Learn more: https://bit.ly/react-web-vitals
21 reportWebVitals();
```

```
EXPLORER ... Preview README.md JS index.js M JS App.js M > JS Home.js U JS Detail.js U JS Login.js U ...
```

```
client > src > JS App.js > App
1 import logo from "./logo.svg";
2 import "./App.css";
3
4 import { Routes, Route } from "react-router-dom";
5 import Home from "./pages/Home";
6 import Login from "./pages/Login";
7 import Detail from "./pages/Detail";
8
9 function App() {
10   return [
11     <div className="App">
12       <header className="App-header">
13         <img src={logo} className="App-logo" alt="logo" />
14
15         <Routes>
16           <Route path="/" element={<Home />} />
17           <Route path="/login" element={<Login />} />
18           <Route path="/:id" element={<Detail />} />
19         </Routes>
20       </header>
21     </div>
22   ];
23 }
24
25 export default App;
```

EXPLORER ... JS App.js M X # App.css M

P3-REACT-ROUTER

- > node_modules
- > public
- src
 - pages
 - Detail.js
 - Home.js
 - Login.js
 - # App.css
 - JS App.js M
 - JS App.test.js
 - # index.css
 - JS index.js M
 - logo.svg
 - JS reportWebVitals.js
 - JS setupTests.js
 - .gitignore
- { package-lock.json M
- { package.json
- README.md
- server
 - db.json
- README.md

PROGRAMATIC NAVIGATION

JS App.js M JS Home.js U X # App.css M

client > src > pages > Home.js > Home > users.map() callback

```
3
4   export default function Home() {
5     const [users, setUsers] = useState([]);
6
7     useEffect(() => {
8       fetch("http://localhost:3001/users")
9         .then((res) => {
10           if (!res.ok) throw new Error("error ngab");
11           return res.json();
12         })
13         .then((data) => setUsers(data))
14         .catch(console.log);
15     }, []);
16
17     return (
18       <div>
19         <h1>Home Page</h1>
20         <ul>
21           {users.map((user) => {
22             return [
23               <li key={user.id}>
24                 {user.name} | <Link to={`/ ${user.id}`}>Detail</Link>
25               </li>
26             ];
27           });
28         </ul>
29       </div>
30     );
31   }
32 }
```

```
Js App.js M Js Home.js U # App.css M
client > src > pages > Js Home.js > ⚡ Home
11     if (!res.ok) throw new Error("error ngab");
12     return res.json();
13   })
14   .then((data) => setUsers(data))
15   .catch(console.log);
16 }, []);
17
18 function handleClick(id) {
19   console.log(id);
20   navigate(`/${id}`);
21 }
22
23 return [
24   <div>
25     <h1>Home Page</h1>
26     <ul>
27       {users.map((user) => {
28         return (
29           <li key={user.id}>
30             {user.name} |{" "}
31             <button onClick={(e) => handleClick(user.id)}>
32               Detail
33             </button>
34           </li>
35         );
36       })}
37     </ul>
38   </div>
39 ]
40 }
```

PANGGIL PARAMS

```
Js App.js M Js Detail.js U # App.css M
client > src > pages > Js Detail.js > ⚡ Detail
1 import { useParams } from "react-router-dom";
2
3 export default function Detail() {
4   let params = useParams();
5   console.log(params);
6
7   return (
8     <div>
9       <h1>Detail Page {params.id}</h1>
10    </div>
11  );
12}
13

Js App.js M X
client > src > Js App.js > ⚡ App
1 import logo from "./logo.svg";
2 import "./App.css";
3
4 import { Routes, Route, Link } from "react-router-dom";
5 import Home from "./pages/Home";
6 import Login from "./pages/Login";
7 import Detail from "./pages/Detail";
8
9 function App() {
10   return [
11     <div className="App">
12       <header className="App-header">
13         <img src={logo} className="App-logo" alt="logo" />
14         <nav>
15           <Link to="/">Home</Link> <Link to="/login">Login</Link>
16         </nav>
17         <Routes>
18           <Route path="/" element={<Home />} />
19           <Route path="/login" element={<Login />} />
20           <Route path="/:userId" element={<Detail />} />
21         </Routes>
22       </header>
23     ];
24   }
25
26
27 export default App;
28
```

REACT QUERY

The screenshot shows a browser window displaying the React Router documentation. The URL is reactrouter.com/docs/en/v6/hooks/use-search-params. The page title is "React Router | useSearchParams". On the left, there's a sidebar with a tree view of hooks, where "useSearchParams" is highlighted. The main content area has a heading "Type declaration" and a paragraph explaining that the `useSearchParams` hook is used to read and modify the query string in the URL. Below this is a code snippet:

```
1 import * as React from "react";
2 import { useSearchParams } from "react-router-dom";
3
4 function App() {
5   let [searchParams, setSearchParams] = useSearchParams();
6
7   function handleSubmit(event) {
8     event.preventDefault();
9
10    // The serialize function here would be responsible for
11    // creating an object of { key: value } pairs from the
12    // fields in the form that make up the query.
13    let params = serializeFormQuery(event.target);
14    setSearchParams(params);
15
16  return (
17    <div>
18      <form onSubmit={handleSubmit}>{/* ... */}</form>
19    </div>
20  );
21}
```

NESTED ROUTES

The screenshot shows a browser window displaying the React Router documentation. The URL is reactrouter.com/docs/en/v6/getting-started/overview#nested-routes. The page title is "React Router | Quick Start". On the left, there's a sidebar with sections like "DOCS HOME", "GETTING STARTED" (with "Quick Start" highlighted), "ROUTERS", and "COMPONENTS". The main content area has a heading "Nested Routes" and a paragraph explaining that nested routes are one of the most powerful features of React Router. Below this is a code snippet:

```
1 function App() {
2   return (
3     <Routes>
4       <Route path="invoices" element={<Invoices />}>
5         <Route path=":invoiceId" element={<Invoice />} />
6         <Route path="sent" element={<SentInvoices />} />
7       </Route>
8     </Routes>
9   );
10}
```

This route config defined three route paths:

- "/invoices"
- "/invoices/sent"
- "/invoices/:invoiceId"

When the URL is "/invoices/sent" the component tree will be:

EXPLORER

- P3-REACT-ROUTER
 - client
 - node_modules
 - public
 - src
 - components
 - Profile.js
 - Setting.js
 - pages
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js
 - logo.svg
 - reportWebVitals.js
 - setupTests.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md
 - server
 - db.json

OUTLET

EXPLORER

- P3-REACT-ROUTER
 - client
 - node_modules
 - public
 - src
 - components
 - Profile.js
 - Setting.js
 - pages
 - Detail.js
 - Home.js
 - Login.js

```

JS Profile.js U JS Setting.js U JS App.js M X
client > src > App.js > App
11 function App() {
12   return (
13     <div className="App">
14       <header className="App-header">
15         <img src={logo} className="App-logo" alt="logo" />
16         <nav>
17           <Link to="/">Home</Link> <Link to="/login">Login</Link>
18         </nav>
19         <Routes>
20           <Route path="/" element={<Home />} />
21           <Route path="/login" element={<Login />} />
22           <Route path="/:id" element={<Detail />}>
23             {/* children */}
24             <Route path="profile" element={<Profile />} />
25             <Route path="setting" element={<Setting />} />
26           </Route>
27         </Routes>
28       </header>
29     </div>
30   );
31 }
32
33 export default App;

```

```

JS Profile.js U JS Setting.js U JS App.js M JS Detail.js U ●
client > src > pages > Detail.js > Detail
1 import { useParams, Outlet } from "react-router-dom";
2
3 export default function Detail() {
4   let params = useParams();
5   console.log(params);
6   // fetch
7
8   return [
9     <div>
10       <h1>Detail Page {params.id}</h1>
11       <div>
12         <Outlet />
13         ini content dinamis
14       </div>
15       <div>ini footer</div>
16     </div>
17   ];
18 }
19

```

PROTECTED ROUTES

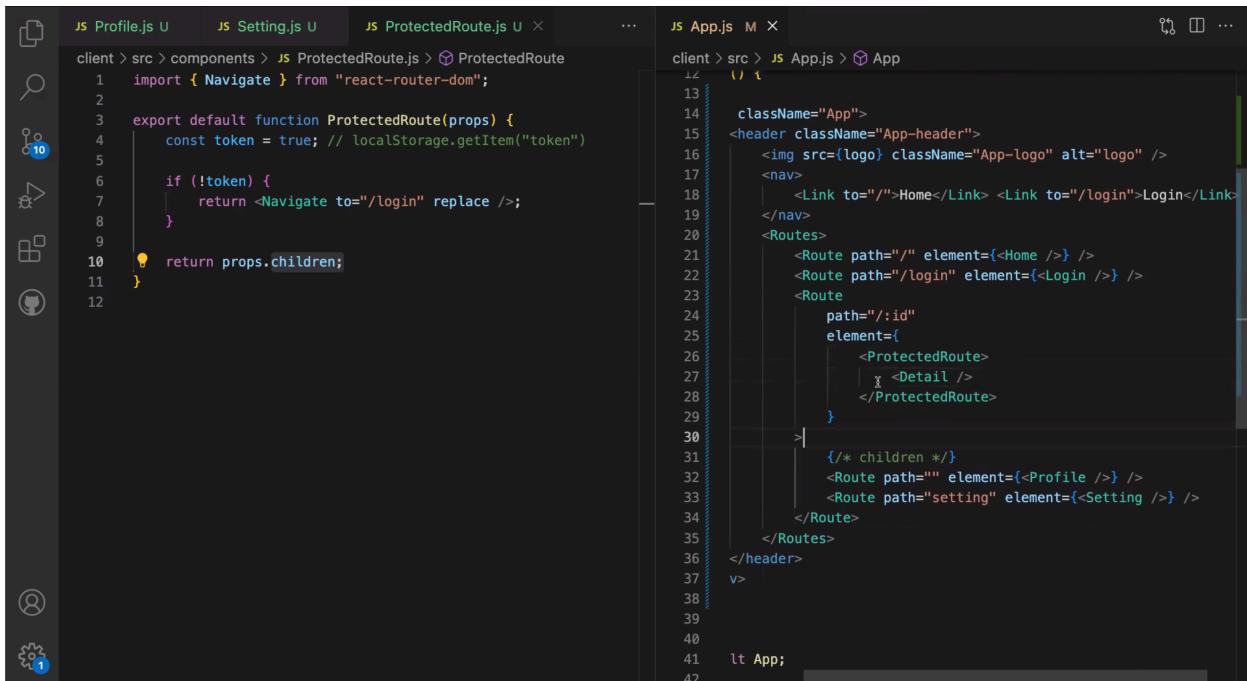
The screenshot shows a browser window displaying the React Router documentation. The URL is reactrouter.com/docs/en/v6/examples/auth. The page title is "Auth Example". On the left, there's a sidebar with sections like "EXAMPLES" (Authentication is selected), "UPGRADING", and "GUIDES". On the right, there's a sidebar showing profile pictures and names of contributors: Rio Hanantyo..., dsrizki, Fajar, Yohanes, and Bexxam. The main content area contains text about restricting access to authenticated users and a list of features. A "Preview" section with a "Open in StackBlitz" button is also present.

PROPS = CHILDREN

The screenshot shows a code editor with a dark theme. The file being edited is "ProtectedRoute.js". The code is as follows:

```
client > src > components > JS ProtectedRoute.js > ProtectedRoute
1 import { Navigate } from "react-router-dom"
2
3 export default function ProtectedRoute (props) {
4   const token = true // localStorage.getItem("token")
5
6   if (!token) {
7     return <Navigate to="/login" replace />
8   }
9
10  return props.children
11}
```

PROTECTED ROUTES – DETAIL



```
client > src > components > JS ProtectedRoute.js > ProtectedRoute
1 import { Navigate } from "react-router-dom";
2
3 export default function ProtectedRoute(props) {
4   const token = true; // localStorage.getItem("token")
5
6   if (!token) {
7     return <Navigate to="/login" replace />;
8   }
9
10  return props.children;
11}
12

JS App.js M X
client > src > JS App.js > App
12 () {
13
14   className="App">
15   <header className="App-header">
16     <img src={logo} className="App-logo" alt="logo" />
17     <nav>
18       <Link to="/">Home</Link> <Link to="/login">Login</Link>
19     </nav>
20   <Routes>
21     <Route path="/" element={<Home />} />
22     <Route path="/login" element={<Login />} />
23     <Route
24       path="/:id"
25       element={
26         <ProtectedRoute>
27           <Detail />
28         </ProtectedRoute>
29       }
30     >
31       /* children */
32       <Route path="" element={<Profile />} />
33       <Route path="setting" element={<Setting />} />
34     </Route>
35   </Routes>
36 </header>
37 <v>
38
39
40
41   <t>App;</t>
42
```

Notfound

"Not Found" Routes

When no other route matches the URL, you can render a "not found" route using `path="*"`. This route will match any URL, but will have the weakest precedence so the router will only pick it if no other routes match.

```
1 function App() {
2   return (
3     <Routes>
4       <Route path="/" element={<Home />} />
5       <Route path="dashboard" element={<Dashboard />} />
6       <Route path="*" element={<NotFound />} />
7     </Routes>
8   );
9 }
```

NAVLINK

EXPLORER ... JS App.js M X

client > src > JS App.js > App > activeStyle > color

```

P3-REACT-ROUTER
  JS Profile.js U
  JS ProtectedRoute.js U
  JS Setting.js U
  pages
    JS Detail.js U
    JS Home.js U
    JS Login.js U
  # App.css M
  JS App.js M
  JS App.test.js
  # index.css
  JS index.js M
  logo.svg
  JS reportWebVitals.js
  JS setupTests.js
  .gitignore
  package-lock.json M
  package.json
  README.md
  server
    db.json
  README.md

> OUTLINE
> TIMELINE

```

```

11
12   function App() {
13     let activeStyle = [
14       textDecoration: "underline",
15       backgroundColor: "purple",
16       color: "white",
17     ];
18   return (
19     <div className="App">
20       <header className="App-header">
21         <img src={logo} className="App-logo" alt="logo" />
22         <nav>
23           {/* <Link to="/">Home</Link> <Link to="/login">Login</Link> */}
24             <NavLink
25               to="/"
26               style={({ isActive }) =>
27                 isActive ? activeStyle : undefined
28               }
29             >
30               Home
31             </NavLink>
32             <NavLink
33               to="/login"
34               style={({ isActive }) =>
35                 isActive ? activeStyle : undefined
36               }
37             >
38               Messages
39             </NavLink>
40           </nav>

```

CONTOH NavLink

App > React Router | NavLink > React-router Auth Example - 1 Issue

localhost:3000/login



Home [Messages](#)

Login Page

Elements Console Sources Network

react-dom.development.js:29840
Download the React DevTools for a better development experience: <https://reactjs.org/link/react-devtools>