

REACT
ROUTER DOOM

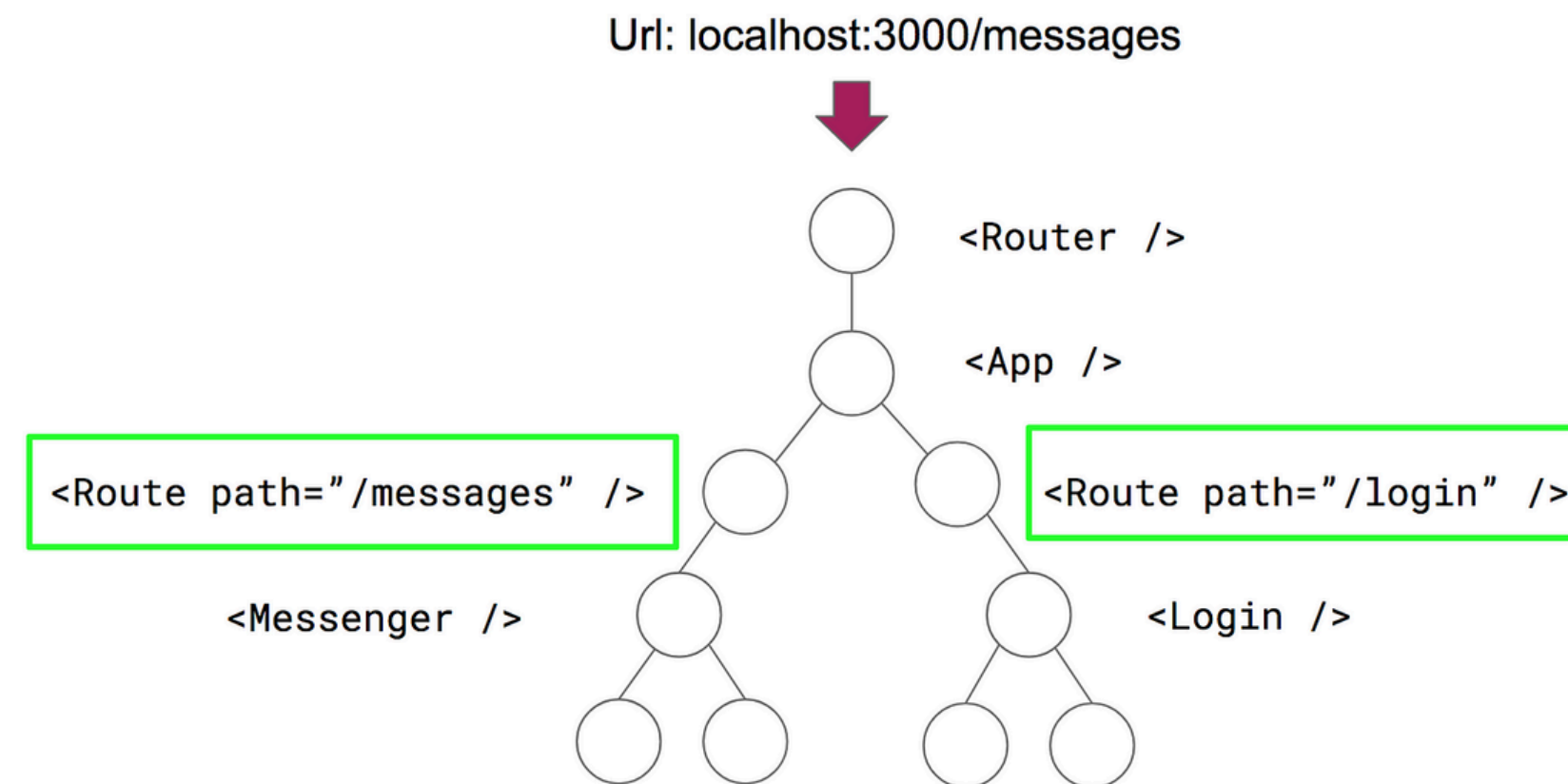
O QUE É O REACT ROUTER DOM?

O **React Router DOM** é uma biblioteca para gerenciar rotas em aplicações React. Ele permite a criação de navegação entre páginas sem a necessidade de recarregar a aplicação, proporcionando uma experiência fluida para o usuário.



POR QUE O REACT ROUTER DOM FOI CRIADO?

No início do React, não havia uma solução oficial para roteamento. Aplicações dependiam de técnicas como manipulação manual do estado e URLs, o que tornava a navegação mais complexa. O React Router DOM surgiu para resolver esse problema, fornecendo uma forma **declarativa** e **organizada** de lidar com rotas.



CENÁRIOS DE USO

O *React Router DOM* é útil em diversos cenários, como:

01

Aplicações de página única (SPA): Permite navegar entre diferentes seções sem recarregar a página.

02

Sistemas administrativos: Para separar dashboards, páginas de usuário e outras seções.

03

E-commerce: Roteamento entre categorias, produtos e carrinho de compras.

04

Aplicações com autenticação: Controle de acesso a páginas protegidas.

COMO FUNCIONA?

O *React Router DOM* permite definir rotas e associá-las a componentes específicos. Os principais conceitos incluem:

- **BrowserRouter:** O componente que envolve toda a aplicação e gerencia o histórico de navegação.
- **Routes e Route:** Definem as rotas e os componentes que serão renderizados.
- **Link:** Substitui a tag `<a>` para navegação sem recarregar a página.

```
1 import { BrowserRouter, Routes, Route } from "react-router-dom";
2 import Home from "./Home";
3 import About from "./About";
4
5 function App() {
6   return (
7     <BrowserRouter>
8       <Routes>
9         <Route path="/" element={<Home />} />
10        <Route path="/about" element={<About />} />
11      </Routes>
12    </BrowserRouter>
13  );
14 }
15
16 export default App;
```

NAVEGAÇÃO ENTRE PÁGINAS

A navegação entre páginas pode ser feita com o componente **Link**:

```
1 import { Link } from "react-router-dom";
2
3 function Navbar() {
4   return (
5     <nav>
6       <Link to="/">Home</Link>
7       <Link to="/about">Sobre</Link>
8     </nav>
9   );
10 }
```

Dessa forma, ao clicar nos links, o *React Router DOM* atualizará a página sem recarregar.

EXPLORE HOOKS
OO ROUTER OOH

EXPLORE HOOKS DO ROUTER DOM

O *React Router* fornece diversos **hooks** para facilitar a manipulação de rotas. Aqui estão os principais:

1. `useNavigate`

Permite navegar programaticamente entre páginas.


```
1 import { useNavigate } from "react-router-dom";
2
3 function HomePage() {
4   const navigate = useNavigate();
5
6   const goToProfile = () => {
7     navigate("/profile/123");
8   };
9
10  return <button onClick={goToProfile}>Ir para o perfil</button>;
11 }
```


EXPLORE HOOKS DO ROUTER DOM

O *React Router* fornece diversos **hooks** para facilitar a manipulação de rotas. Aqui estão os principais:

2. useParams

Captura parâmetros dinâmicos da URL.




```
1 import { useParams } from "react-router-dom";
2
3 function Profile() {
4   const { id } = useParams();
5   return <h1>Perfil do usuário: {id}</h1>;
6 }
```

EXPLORE HOOKS DO ROUTER DOM

O *React Router* fornece diversos **hooks** para facilitar a manipulação de rotas. Aqui estão os principais:

3. `useLocation`

Fornece informações sobre a URL atual.



```
1 import { useLocation } from "react-router-dom";
2
3 function CurrentPage() {
4   const location = useLocation();
5   return <p>Você está na página: {location.pathname}</p>;
6 }
```

EXPLORE HOOKS DO ROUTER DOM

O *React Router* fornece diversos **hooks** para facilitar a manipulação de rotas. Aqui estão os principais:

4. `useSearchParams`

Trabalha com parâmetros de consulta na URL.

```
1 import { useSearchParams } from "react-router-dom";
2
3 function SearchPage() {
4   const [searchParams, setSearchParams] = useSearchParams();
5   const query = searchParams.get("q");
6
7   return (
8     <div>
9       <p>Pesquisa: {query}</p>
10      <button onClick={() => setSearchParams({ q: "React" })}>
11        Pesquisar React
12      </button>
13    </div>
14  );
15 }
```

EXPLORE HOOKS DO ROUTER DOM

O *React Router* fornece diversos **hooks** para facilitar a manipulação de rotas. Aqui estão os principais:

5. useRoutes

Permite definir as rotas diretamente dentro do componente.

```
1 import { useRoutes } from "react-router-dom";
2
3 const routes = [
4   { path: "/", element: <Home /> },
5   { path: "/about", element: <About /> },
6 ];
7
8 function App() {
9   const element = useRoutes(routes);
10  return element;
11 }
```

COMO USAR

INSTALANDO REACT ROUTER V6

Primeiro, crie um novo projeto **React** .

Você pode instalar o React Router do registro público npm ou yarn:

npm i react-router-dom

ou

yarn add react-router-dom

Importe o React Router no seu projeto:



```
1 import React from "react";  
2 import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";  
3
```

INSTALANDO REACT ROUTER V6

Crie seus componentes que serão renderizados em cada rota:

```
1 function Home() {  
2   return <h2>Home</h2>;  
3 }  
4  
5 function About() {  
6   return <h2>About</h2>;  
7 }  
8  
9 function Users() {  
10  return <h2>Users</h2>;  
11 }
```

INSTALANDO REACT ROUTER V6

No seu App.jsx (ou .tsx), crie o navegador que será sempre renderizado. Depois configure em qual rota cada componente será renderizado:

```
1 export default function App() {
2   return (
3     <Router>
4       <div>
5         <nav>
6           <ul>
7             <li>
8               <Link to="/">Home</Link>
9             </li>
10            <li>
11              <Link to="/about">About</Link>
12            </li>
13            <li>
14              <Link to="/users">Users</Link>
15            </li>
16          </ul>
17        </nav>
18
19        {/* Usando <Routes> no lugar de <Switch> */}
20        <Routes>
21          <Route path="/about" element={<About />} />
22          <Route path="/users" element={<Users />} />
23          <Route path="/" element={<Home />} />
24        </Routes>
25      </div>
26    </Router>
27  );
28 }
```

As antigas versões no
React Router DOM usam
Switch ao invés de *Routes*

CONCLUSÃO

O *React Router DOM* é essencial para aplicações React **modernas** que necessitam de navegação estruturada. Ele simplifica a criação de rotas, **melhora a experiência** do usuário e facilita o gerenciamento da interface.

THANK
YOU

@wallace027dev