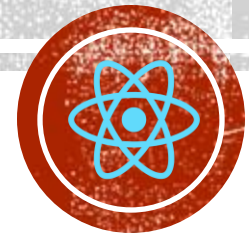


SINGLE PAGE APPLICATIONS COM REACT-DOM



Douglas Nassif Roma Junior

 /douglasjunior

 /in/douglasjunior

 nassifrroma@gmail.com

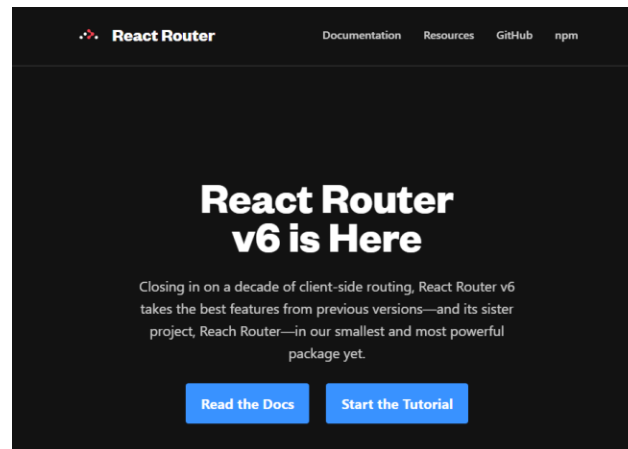
AGENDA

Tópico	Conteúdo
React Router	<ul style="list-style-type: none">- Introdução ao React Router- Parâmetros de URL- Rotas privadas- Rotas desconhecidas (404)

REACT ROUTER

INTRODUÇÃO AO REACT ROUTER

- **React Router** é uma coleção de componentes de navegação que compõem declarativamente com sua aplicação.
- Se você quer ter URLs navegáveis para seu aplicativo **Web** ou uma maneira componentizada para navegar no **React Native**, o React Router funciona onde quer que o **React** esteja renderizando.



INTRODUÇÃO AO REACT ROUTER

- Para instalar o React Router para Web, basta executar:

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

- Vamos criar as seguintes páginas:

- `src/pages/Tasks.jsx`
- `src/pages/Home.jsx`
- `src/pages/About.jsx`

INTRODUÇÃO AO REACT ROUTER

- Configurando as rotas para as páginas:

App.jsx

```
import {
  HashRouter,
  Routes,
  Route,
  Link,
} from 'react-router-dom';

import Home from './pages/Home';
import Tasks from './pages/Tasks';
import About from './pages/About';

// ...
```

```
// ...
<HashRouter>
  <ul>
    <li><Link to="/">Home</Link></li>
    <li><Link to="/tasks">Tarefas</Link></li>
    <li><Link to="/about">Sobre</Link></li>
  </ul>
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/tasks" element={<Tasks />} />
    <Route path="/about" element={<About />} />
  </Routes>
</HashRouter>
// ...
```

PARÂMETROS DE URL

PARÂMETROS DE URL

- **React Router** permite que parâmetros sejam passados entre as rotas, usando parâmetros de URL.
- Para usar parâmetros de URL, basta usar o caractere “:” seguido do nome do parâmetro.

pages/Hero.js

```
import { useParams } from 'react-router-dom';

const Hero = () => {
  const params = useParams();
  return (
    <div>
      <h1>Hero Page</h1>
      <h3>Id: {params.heroId}</h3>
    </div>
  )
}

export default Hero;
```

App.js

```
// ...

<li><Link to="/hero/strange">Dr Estranho</Link></li>
<li><Link to="/hero/captain">Cap América</Link></li>
<li><Link to="/hero/spider-man">Homem Aranha</Link></li>

// ...

<Route path="/hero/:heroId" element={<Hero />} />

// ...
```


ROTAS PRIVADAS

ROTAS PRIVADAS

- Usando o componente `Navigate` é possível redirecionar o usuário para uma página determinada.
- Esta abordagem é útil, por exemplo, quando alguns dos componentes exigem autenticação do usuário.

components/PrivateRoute.jsx

```
import { Navigate } from 'react-router-dom';
import fakeAuth from '../fake-auth';

const PrivateRoute = ({ children }) => (
  fakeAuth.isAuthenticated
    ? children
    : <Navigate to="/login" />
);

export default PrivateRoute;
```

App.jsx

```
import Login from '../pages/Login';
import PrivateRoute from '../components/PrivateRoute';
// ...
<Route path="/tasks" element={
  <PrivateRoute>
    <Tasks />
  </PrivateRoute>
} />
<Route path="/login" element={<Login />} />
// ...
```

ROTAS PRIVADAS

fake-auth.jsx

```
const fakeAuth = {
  isAuthenticated: false
}

export default fakeAuth;
```

pages/Login.jsx

```
import { useNavigate } from 'react-router-dom';
import fakeAuth from '../fake-auth';

const Login = () => {
  const navigate = useNavigate();
  const login = () => {
    fakeAuth.isAuthenticated = true;
    navigate('/tasks')
  }
  return (
    <div>
      <h1>Login page</h1>
      <input placeholder="Usuário"/> <br/>
      <input placeholder="Senha"/> <br/>
      <button onClick={login}>Entrar</button>
    </div>
  );
}

export default Login;
```

ROTAS PRIVADAS

pages/Tasks.jsx

```
import { useNavigate } from 'react-router-dom';
import fakeAuth from '../fake-auth';

const Tasks = () => {
  const navigate = useNavigate();
  const logout = () => {
    fakeAuth.isAuthenticated = false;
    navigate('/login')
  }
  return (
    <div>
      <h1>Tasks page</h1>
      <button onClick={logout}>
        Sair
      </button>
    </div>
  );
}

export default Tasks;
```

ROTAS DESCONHECIDAS (404)

ROTAS DESCONHECIDAS (404)

- Em uma aplicação Web pode ser interessante exibir uma mensagem amigável para um usuário. Com **React Router**, basta utilizar o componente `Route` com `path=""`.
- Deste modo, se nenhum outro `path` combinar com a URL aberta no navegador, então o `path=""` será renderizado.

App.js

```
import NotFound from './pages/NotFound';

// ...

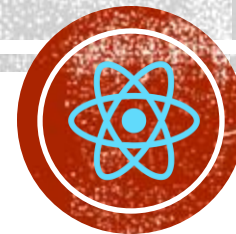
<Route path="" element={<NotFound />} />

// ...
```

REFERÊNCIAS

- React Router - <https://reactrouter.com/docs>
- Rotas privadas: <https://dev.to/iamandrewluca/private-route-in-react-router-v6-lg5>
- Tutoriais: <https://reactrouter.com/en/start/tutorial>

OBRIGADO!



Douglas Nassif Roma Junior

 /douglasjunior

 /in/douglasjunior

 nassifroma@gmail.com