

Tổng Hợp kiến thức

1. Tổng quan

- **React Router DOM v7:** Thư viện định tuyến cho ứng dụng React SPA.
- **Tính năng chính:** Route tĩnh/động, nested routes, điều hướng programmatic (`useNavigate`), xử lý lỗi.

Cấu trúc thư mục ứng dụng Blog:

```
src/  
├── components/ (NavBar, ErrorMessage)  
├── screens/ (Home, BlogList, BlogDetail, Dashboard, NotFound)  
├── App.jsx  
└── index.css
```

2. Cài đặt

- **Yêu cầu:** Node.js, npm/yarn, dự án React (Vite).
- **Cài đặt:**

```
npm install react-router-dom@7
```

- **Cấu hình cơ bản:**

```
// src/App.jsx  
import { BrowserRouter } from "react-router-dom";  
import Home from "./screens/Home/Home";  
  
function App() {  
  return (  
    <BrowserRouter>  
      <Home />  
    </BrowserRouter>  
  );  
}
```

```
};  
}
```

3. Component Routes

- **<Routes>** : Container chính cho các route, thay **<Switch>** .
- **<Route>** : Định nghĩa route với **path** và **element** .
- Ví dụ:

```
import { BrowserRouter, Routes, Route } from "react-router-dom";  
function App() {  
  return (  
    <BrowserRouter>  
      <Routes>  
        <Route path="/" element={<Home />} />  
        <Route path="/blogs" element={<BlogList />} />  
      </Routes>  
    </BrowserRouter>  
  );  
}
```

4. Routes và Route

- **Route tĩnh:** **path="/"** .
- **Route động:** **path="/blogs/:id"** → Dùng **useParams()** lấy tham số.
- **Route lỗi:** **path="*"** .
- Ví dụ:

```
<Routes>  
  <Route path="/" element={<Home />} />  
  <Route path="/blogs/:id" element={<BlogDetail />} />  
  <Route path="*" element={<NotFound />} />
```

```

</Routes>

// BlogDetail.jsx
import { useParams } from "react-router-dom";
function BlogDetail() {
  const { id } = useParams();
  return <h1>Blog Detail - ID: {id}</h1>;
}

```

5. Link, NavLink và Navigate

- **<Link>** : Điều hướng cơ bản.
- **<NavLink>** : Thêm style/class khi active (`className={({ isActive }) => ...}`).
- **<Navigate>** : Redirect programmatic.
- **useNavigate** : Điều hướng bằng code (sau hành động như nhấn nút).
- **Ví dụ:**

```

// NavBar.jsx
import { Link, NavLink } from "react-router-dom";
function NavBar() {
  return (
    <nav>
      <NavLink to="/" className={({ isActive }) => (isActive ? "active" : "")>
        >Home</NavLink> |
      <NavLink to="/blogs" className={({ isActive }) => (isActive ? "active" : "")>
        >Blogs</NavLink> |
      <Link to="/blogs/1">Blog #1</Link>
    </nav>
  );
}

// App.jsx
<Route path="/old-blogs" element={<Navigate to="/blogs" />} />

```

```
// BlogDetail.jsx
import { useParams, useNavigate } from "react-router-dom";
function BlogDetail() {
  const { id } = useParams();
  const navigate = useNavigate();
  return (
    <div>
      <h1>Blog Detail - ID: {id}</h1>
      <button onClick={() => navigate("/blogs")}>Back to Blog List</button>
    </div>
  );
}
```

6. Nested Routes

- **Nested Routes:** Route con trong route cha, dùng `<Outlet>` để render.
- **Ví dụ:**

```
// App.jsx
<Route path="/dashboard" element={<Dashboard />}>
  <Route path="posts" element={<Posts />} />
  <Route path="stats" element={<Stats />} />
</Route>

// Dashboard.jsx
import { Link, Outlet } from "react-router-dom";
function Dashboard() {
  return (
    <div>
      <h1>Dashboard</h1>
      <nav>
        <Link to="posts">Posts</Link> | <Link to="stats">Stats</Link>
      </nav>
      <Outlet />
    </div>
  );
}
```

```

    </nav>
    <Outlet />
  </div>
);
}

// Posts.jsx
function Posts() { return <h2>Manage Posts</h2>; }

// Stats.jsx
function Stats() { return <h2>Blog Stats</h2>; }

```

7. Hiện thị lỗi

- `path="*" : Bắt các đường dẫn không khớp (404).`
- `useNavigate` : Điều hướng khi nhấn nút trong trang lỗi.
- **Ví dụ:**

```

// App.jsx
<Route path="*" element={<NotFound />} />

// NotFound.jsx
import { useNavigate } from "react-router-dom";
function NotFound() {
  const navigate = useNavigate();
  return (
    <div>
      <h1>404 - Page Not Found</h1>
      <button onClick={() => navigate("/")}>Back to Home</button>
    </div>
  );
}

```

8. Hook hữu ích

- `useParams()` : Lấy tham số từ URL (`:id`).

```
const { id } = useParams();
```

- `useNavigate()` : Điều hướng programmatic.

```
const navigate = useNavigate();  
navigate("/blogs"); // Chuyển hướng  
navigate(-1); // Quay lại trang trước
```

9. Lưu ý quan trọng

- **Cấu hình**: Luôn bọc ứng dụng trong `<BrowserRouter>`.
- **Route lỗi**: Đặt `path=""` cuối cùng trong `<Routes>`.
- **Điều hướng**:
 - Dùng `<Link>` / `<NavLink>` cho điều hướng tĩnh.
 - Dùng `useNavigate` cho điều hướng sau hành động (nhấn nút).
- **Nested Routes**: `<Outlet>` là nơi render route con.

Ví dụ tổng hợp (App.jsx)

```
import { BrowserRouter, Routes, Route, Navigate } from "react-router-dom";  
import NavBar from "../components/NavBar/NavBar";  
import Home from "../screens/Home/Home";  
import BlogList from "../screens/BlogList/BlogList";  
import BlogDetail from "../screens/BlogDetail/BlogDetail";  
import Dashboard from "../screens/Dashboard/Dashboard";  
import Posts from "../screens/Dashboard/Posts/Posts";  
import Stats from "../screens/Dashboard/Stats/Stats";  
import NotFound from "../screens/NotFound/NotFound";
```

```

function App() {
  return (
    <BrowserRouter>
      <NavBar />
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/blogs" element={<BlogList />} />
        <Route path="/blogs/:id" element={<BlogDetail />} />
        <Route path="/old-blogs" element={<Navigate to="/blogs" />} />
        <Route path="/dashboard" element={<Dashboard />} />
        <Route path="/posts" element={<Posts />} />
        <Route path="/stats" element={<Stats />} />
      </Route>
      <Route path="*" element={<NotFound />} />
    </Routes>
  </BrowserRouter>
);
}

export default App;

```