

CHANGED 9 DAYS AGO

Heart ⌂ ⌄

Nội dung:  
 Cài react-router-dom  
 BrowserRouter, Routes, Route, Link, useNavigate  
 Route động với tham số URL  
 Truyền props qua Link hoặc thông qua id  
 Giới thiệu về NextJS (bổ sung nâng cao)  
 Thực hành:  
 Trang danh sách sản phẩm → chi tiết sản phẩm  
 Điều hướng trang thêm sản phẩm

## 📘 PHẦN 1 – LÝ THUYẾT CHI TIẾT

### ♦ 1. Single Page Application (SPA) là gì?

- Là ứng dụng chỉ có 1 file HTML duy nhất.
- Khi chuyển trang, không load lại toàn bộ trang mà chỉ thay đổi nội dung DOM tương ứng.
- Ưu điểm:**  
 Trải nghiệm mượt mà, không load lại trang.  
 Giảm băng thông và thời gian phản hồi.
- Hạn chế:**  
 Phải xử lý route bằng client-side.

### ♦ 2. React Router là gì?

react-router-dom là thư viện phổ biến nhất để điều hướng trong React.

Các thành phần chính:

Thành phần	Mô tả
BrowserRouter	Quản lý định tuyến bằng history API
Routes	Bao bọc các Route
Route	Định nghĩa một đường dẫn
Link	Điều hướng đến route khác
useParams()	Truy xuất :param trong URL
useNavigate()	Điều hướng bằng code (JS)
useLocation()	Truy xuất dữ liệu truyền qua route

### ♦ 3. Cài đặt React Router

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

Trong index.js:

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

ReactDOM.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>,
  document.getElementById('root')
);
```

### ♦ 4. Khai báo các tuyến đường (Route)

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

<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/products" element={<ProductList />} />
  <Route path="/products/:id" element={<ProductDetail />} />
  <Route path="/add-product" element={<AddProduct />} />
  <Route path="*" element={<NotFound />} />
</Routes>
```

### ♦ 5. Điều hướng không reload với Link

```
<Link to="/products">Xem sản phẩm</Link>
Tương đương với thẻ <a>, nhưng không reload.
```

### ♦ 6. Route động: /products/:id

```
<Route path="/products/:id" element={<ProductDetail />} />
```

Trong ProductDetail.jsx:

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

const ProductDetail = () => {
  const { id } = useParams();
  return <div>Sản phẩm số: {id}</div>;
};
```

### ♦ 7. Điều hướng bằng code – useNavigate()

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

const AddProduct = () => {
  const navigate = useNavigate();

  const handleSubmit = () => {
    // Submit logic
    navigate('/products');
  };
};
```

### ♦ 8. Truyền dữ liệu giữa các route

#### Cách 1: Truyền qua Link state

```
<Link to="/products/1" state={{ product: productObj }}>Chi tiết</Link>
```

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

const ProductDetail = () => {
  const location = useLocation();
  const product = location.state?.product;
};
```

#### Cách 2: Truyền qua URL param và fetch lại

```
<Link to={`/products/${product.id}`}>Chi tiết</Link>

const ProductDetail = () => {
  const { id } = useParams();
  useEffect(() => {
    fetch(`api/products/${id}`).then(res => ...);
  }, [id]);
};
```

✓ Khuyến nghị: Chỉ nên truyền dữ liệu nhẹ hoặc ID → fetch lại ở trang nhận.

## 🎓 PHẦN 2 – ÔN TẬP & CÙNG CỐ

Khái niệm	Hiểu rõ khi nào dùng
Route tĩnh	Trang cố định
Route động :id	Trang chi tiết
Link , useNavigate	Điều hướng giữa trang
useParams	Lấy id từ URL
useLocation	Nhận dữ liệu từ Link
navigate(-1)	Quay lại trang trước

## 📝 PHẦN 3 – THỰC HÀNH CỤ THỂ

### ♦ Bài 1 – Tạo cấu trúc route

Tạo 4 file:

- Home.js
- ProductList.js
- ProductDetail.js
- AddProduct.js (Giao diện)

Chuyển trang qua lại và truyền đúng tham số đầu vào

API:

Danh sách sản phẩm: <https://fakestoreapi.com/products>

Chi tiết sản phẩm: <https://fakestoreapi.com/products/1> (Thay 1 bằng id)