

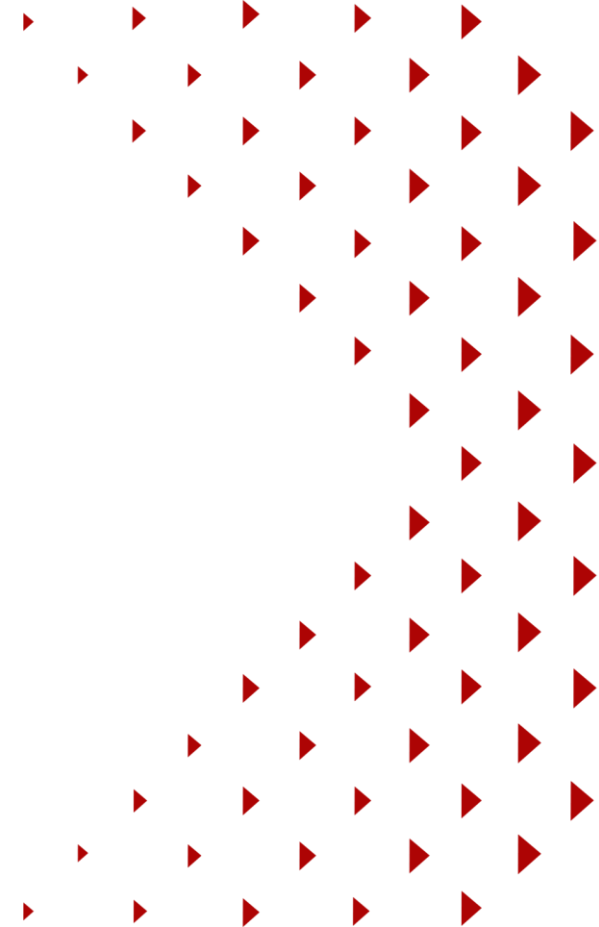


SESSION 07:

REACT ROUTER

Module: Training Program Preparation

Version: 2.0



- 1. Tổng quan về React Router**
- 2. Configuring Routes**
- 3. Router**
- 4. Link và NavLink**
- 5. Navigate**
- 6. Dynamic Routes**
- 7. Search Param**
- 8. Protected routes**
- 9. Lazy loading**
- 10. Scroll to top**

1. Tổng quan về React Router

- React - Router là một thư viện định tuyến (Routing) tiêu chuẩn của React. Nó giữ cho giao diện của ứng dụng đồng bộ với URL trên trình duyệt
- React - Router cho phép bạn định tuyến “luồng dữ liệu” (data flow) trong ứng dụng của bạn một cách rõ ràng. React - Router giúp map URL với Router để render component tương ứng
- Cài đặt :

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



2. Configuring Routes

- **<BrowserRouter>** :
 - Được sử dụng phổ biến nhất , sử dụng History API có trong HTML5 để theo dõi lịch sử bộ định tuyến
 - **Bao bọc tất cả component** của ứng dụng
- **<HashRouter>** : Sử dụng hash của URL (window.location.hash) để ghi nhớ mọi thứ.
- **<MemoryRouter>**: giữ lịch sử của URL trong bộ nhớ. Thường sử dụng trong mục đích testing
- **<StaticRouter>**: Sử dụng cho server-side rendering
- **Sử dụng :**

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

```
<BrowserRouter>  
  <App />  
</BrowserRouter>
```

3. Router - 1

- **Route** : Định nghĩa một ánh xạ (mapping) giữa một URL và một Component - khi người dùng truy cập theo một URL trên trình duyệt , một Component tương ứng sẽ được render trên giao diện

```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/about" element={<About />}></Route>
  <Route path="/product" element={<Product />} />
</Routes>
```

- Trong đó:
 - **Path**: là đường dẫn trên URL
 - **Element** : là đối tượng sẽ được load ra tương ứng

3. Router - 2

- **Nested Route**
 - Nested Route - (định tuyến lồng nhau) : tính năng mạnh mẽ nhất của React Router
 - Thường được dùng với Tabs Component hoặc một trang có chứa các trang con bên trong nó

```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/about" element={<About />}>
    <Route path="aboutRikkeiSoft" element={<AboutRikkeiSoft />} />
    <Route path="aboutRikkeiAcademy" element={<AboutRikkeiAcademy />} />
  </Route>
  <Route path="/product" element={<Product />} />
</Routes>
```

3. Router - 3

- **Outlet:** xác định vị trí hiển thị component con khi route trùng khớp

```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/about" element={<About />} />
  <Route path="aboutRikkeiSoft" element={<AboutRikkeiSoft />} />
  <Route path="aboutRikkeiAcademy" element={<AboutRikkeiAcademy />} />
</Route>
<Route path="/product" element={<Product />} />
</Routes>
```

```
export default function About() {
  return (
    <div>
      <Outlet />
    </div>
  )
}
```

3. Router - 4

- **Index route**
 - Hiển thị component con trong component cha

```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/about" element={<About />}>
    <Route index element={<AboutIndex />} />
    <Route path="aboutRikkeiSoft" element={<AboutRikkeiSoft />} />
    <Route path="aboutRikkeiAcademy" element={<AboutRikkeiAcademy />} />
  </Route>
  <Route path="/product" element={<Product />} />
</Routes>
```


3. Router - 5

- **Not found routes**
 - Điều hướng đến Component xử lý khi người dùng nhập URL không hợp lệ

```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/about" element={<About />}>
    <Route index element={<AboutIndex />} />
    <Route path="aboutRikkeiSoft" element={<AboutRikkeiSoft />} />
    <Route path="aboutRikkeiAcademy" element={<AboutRikkeiAcademy />} />
  </Route>
  <Route path="*" element={<NotFound />} />
</Routes>
```

4. Link và NavLink - 1

- Trong HTML thì cặp thẻ để chuyển hướng đó là thẻ `<a>` thì trong ReactJS chúng ta sẽ sử dụng cặp thẻ `<Link></Link>` được import từ React-Router

```
import { Link } from "react-router-dom";
```

```
<ul className="menu">
  <li><Link to={'/'}>Trang chủ</Link></li>
  <li><Link to={'/about'}>Giới thiệu</Link>
    <ul className="submenu">
      <li><Link to={"/about/aboutRikkeiSoft"}>Về Rikkeisoft</Link></li>
      <li><Link to={"/about/aboutRikkeiAcademy"}>Về Rikkei Academy</Link></li>
    </ul>
  </li>
  <li><Link to={'/product'}>Sản phẩm</Link></li>
</ul>
```

- Trong đó:
 - to** : đường dẫn muốn đến

4. Link và NavLink - 2

- Giống với Link về cách sử dụng
- Tốt hơn vì nó hỗ trợ thêm một số thuộc tính như là activeClassName và activeStyle để style cho các NavLink.

```
import { NavLink } from "react-router-dom";
```

```
<ul className="menu">
  <li><NavLink to={'/' } style={navLinkStyle}>Trang chủ</NavLink></li>
  <li><Link to={'/about'}>Giới thiệu</Link>
    <ul className="submenu">
      <li><Link to={"/about/aboutRikkeiSoft"}>Về Rikkeisoft</Link></li>
      <li><Link to={"/about/aboutRikkeiAcademy"}>Về Rikkei Academy</Link></li>
    </ul>
  </li>
  <li><NavLink to={'/product'} style={navLinkStyle}>Sản phẩm</NavLink></li>
</ul>
```

```
const navLinkStyle = ({ isActive }) => ({
  color: isActive ? "red" : "black",
  backgroundColor: isActive ? "black" : "red"
})
```

5. Navigate - 1

- **useNavigate**: sử dụng để chuyển trang tự động
- Các cách sử dụng:
 - useNavigate với 1 tham số
 - useNavigate với history
 - useNavigate với thuộc tính replace = true
 - useNavigate với chuyển dữ liệu

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

```
const navigate = useNavigate();
```

```
/* useNavigate với 1 tham số */
<button onClick={() => navigate('product')}>Go to Product</button>
/* useNavigate với history */
<button onClick={() => navigate(-2)}>Go 2 pages back</button>
<button onClick={() => navigate(-1)}>Go back</button>
<button onClick={() => navigate(1)}>Go forward</button>
<button onClick={() => navigate(2)}>Go 2 pages forward</button>
/* useNavigate với thuộc tính replace = true */
<button onClick={() => navigate('product', { replace: true })}>
  Go to Product - not save history
</button>
/* useNavigate với chuyển dữ liệu */
<button onClick={() => navigate('product', { state: { id: 10 } })}>
  Go to Product with Data
</button>
```

5. Navigate - 2

- **Component Navigate:** sử dụng để chuyển trang khi return ở function component

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

```
export default function Checkout() {  
  const isLogin = false;  
  return isLogin ? <Home /> : <Navigate to={'/login'} />  
}
```

6. Dynamic routes

- Giải quyết các routes động, routes thay đổi theo một cấu trúc đã được định nghĩa sẵn
- Sử dụng useParams lấy tham số trên URL

```
<Route path='/product/course' element={<Course />} />
<Route path='/product/course/:courseId' element={<CourseDetail />} />
<Route path='/product/course/:courseType/:courseId' element={<CourseDetailByType />} />
```

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

export default function CourseDetail() {
  const params = useParams();
  return (
    <div>
      <h2>Chi tiết khóa học</h2>
      <p>Mã khóa học: {params.courseId}</p>
    </div>
  )
}
```

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

export default function CourseDetailByType() {
  const params = useParams();
  return (
    <div>
      <h2>Chi tiết khóa học theo loại khóa học</h2>
      <p>Loại khóa học: {params.courseType} - Mã khóa học: {params.courseId}</p>
    </div>
  )
}
```

7. Search Param

- Sử dụng để sửa đổi query string trên URL
- Mảng gồm 2 tham số: tham số tìm kiếm và hàm để thay đổi

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

```
const [searchParam, setSearchParam] = useSearchParams();  
let courseName = searchParam.get('courseName');  
let time = searchParam.get('time');
```

```
<button onClick={() => setSearchParam({ courseName: 'java', time: 1000 })}>Detail</button>
```

localhost:3000/product?courseName=java&time=1000

8. Protected Routes

- Ngăn chặn truy cập các link không hợp lệ

```
import React from 'react'
import { Navigate, Outlet } from 'react-router-dom';

export default function PrivateRouter() {
  const isLoggedIn = true;
  return isLoggedIn ? <Outlet /> : <Navigate to={'/login'} />
}
```

```
<Route element={<PrivateRouter />} >
  <Route path='/personal' element={<Personal />} />
  <Route path='/account' element={<Account />} />
</Route>
```


9. Lazy Loading

- Kỹ thuật tối ưu hóa ứng dụng
- Chỉ load những component cần thiết giúp ứng dụng chuyển động mượt mà, nhanh chóng, tăng trải nghiệm người dùng

```
import React from 'react';
import { Routes, Route } from 'react-router-dom';
import Home from './components/Home';

const LazyAbout = React.lazy(() => import('./components/About'));

function App() {
  return (
    <div>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={
          <React.Suspense fallback={<div>Loading...</div>}>
            <LazyAbout />
          </React.Suspense>
        } />
      </Routes>
    </div>
  );
}
```

Scroll To Top

- Kỹ thuật đặt thanh cuộn luôn ở vị trí đầu trang

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

export default function ScrollToTop() {
  const { pathname } = useLocation();
  useEffect(() => {
    window && window.scrollTo(0, 0);
  }, [pathname]);
  return null;
}
```

```
function App() {
  return (
    <div>
      <ScrollToTop />
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={
          <React.Suspense fallback={<div>Loading...</div>}>
            <LazyAbout />
          </React.Suspense>
        } />
      </Routes>
    </div>
  );
}
```

- ❑ **Tổng quan về React Router**
- ❑ **Các thành phần trong React Router**
- ❑ **Các kỹ thuật load component**



KẾT THÚC

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN