

React Router

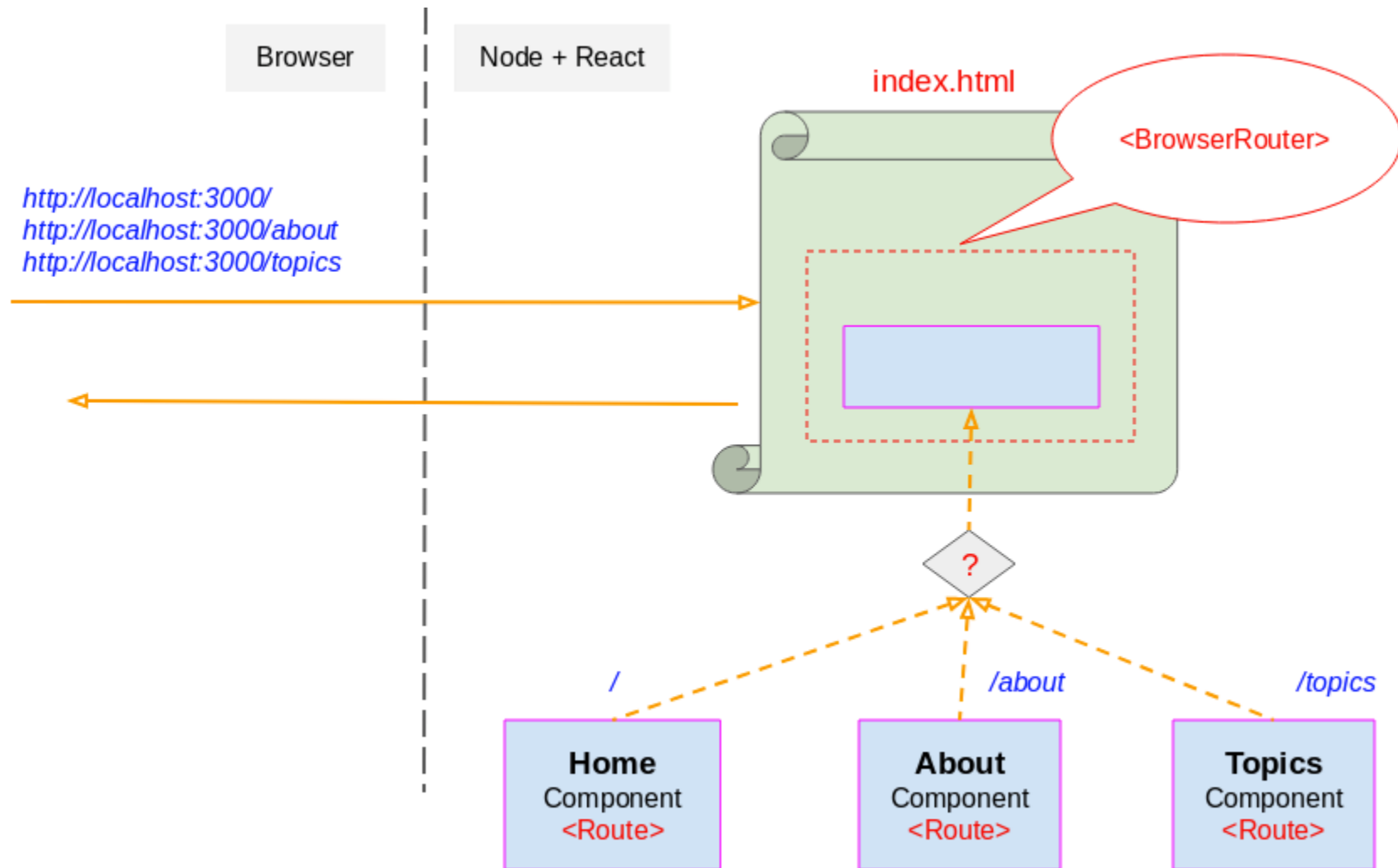
Khóa học: ReactJS

Mục tiêu

- Trình bày được cơ chế routing trong ReactJS
- Sử dụng React Router trong ReactJS
- Triển khai được Router

Giới thiệu

- Với HTML, khi điều hướng từ trang này sang trang khác, chúng ta sẽ sử dụng thẻ `<a>`
- Trong React JS thường được sử dụng để xây dựng những Single Page Application (SPA) khi chúng ta làm như vậy thì sẽ reloading lại toàn bộ page từ server.
- Thực chất, SPA chỉ có 1 trang html, nhưng bao gồm nhiều page views (components), và sẽ load ra page view tương ứng dựa trên route chứa component đó.



React Router

- React Router là một thư viện định tuyến (routing) sử dụng để điều hướng từ page này sang page khác mà không refresh browser.
- Cài đặt:
`npm install react-router-dom --save`
- Gồm 3 packages
 - `react-router`
 - `react-router-dom`
 - `react-router-native`

Các thành phần trong React-Router

React-Router cung cấp cho chúng ta 2 thành phần hay sử dụng đó là **BrowserRouter** & **HashRouter**. Hai thành phần này khác nhau ở kiểu **URL** mà chúng sẽ tạo ra và đồng bộ.

BrowserRouter: Được sử dụng phổ biến hơn, nó sử dụng **History API** có trong HTML5 (`document.location.href`) để theo dõi lịch sử bộ định tuyến của bạn.

HashRouter: Sử dụng **hash** của **URL** (`window.location.hash`) để ghi nhớ mọi thứ.

Các thành phần trong React-Router

Có ba loại thành phần chính trong **React Router**:

- ❖ Router, như là **<BrowserRouter>** và **<HashRouter>**
- ❖ Route matchers, như là **<Route>** và **<Switch>**
- ❖ Chuyển hướng, như là **<Link>**, **<NavLink>**, và **<Redirect>**

Tất cả các component mà bạn sử dụng trong một ứng dụng web nên được nhập từ **react-router-dom**.

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

React Router

Ví dụ:

```
<Router>
  <div className="App">
    <Route path="/" exact component={Home} />
    <Route path="/about" component={About} />
    <Route path="/contact" component={Contact} />
    <Route component={NotFound}/>
  </div>
</Router>
```

- **path:** Là đường dẫn trên URL.
- **exact:** Giúp cho route này chỉ hoạt động nếu URL trên trình duyệt phù hợp tuyệt đối với giá trị của thuộc tính path của nó.
- **component:** Là component sẽ được load ra tương ứng với Route đó.

Link

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

```
<Link to="/about">About</Link>
```

trong đó:

to: Giống như thuộc tính **href** trong thẻ **a**.

NavLink

NavLink thì rất giống với Link về cách sử dụng, nhưng NavLink tốt hơn vì nó hỗ trợ thêm một số thuộc tính như là activeClassName và activeClassName 2 thuộc tính này giúp cho khi mà nó trùng khớp thì nó sẽ được active lên và chúng ta có thể style cho nó.

```
<NavLink exact activeStyle={{  
  backgroundColor : 'white',  
  color : 'red'  
}} to="/" className="my-link">  
Trang Chu  
</NavLink>
```

Match

Khi bạn muốn lấy một số thông tin ở trên URL thì bạn có thể dùng đối tượng match để lấy dữ liệu về. Tại cấu hình Router ta chỉ cần truyền thêm đối tượng match vào component mà cần sử dụng đối tượng match

```
{  
  path : '/products',  
  exact : false,  
  main : ({match}) => <Products match={match} />  
}
```

Prompt

Giả sử khi bạn đang nhập liệu ở form nào đó mà không may click nút back hay chuyển trang thì dữ liệu bạn nhập sẽ mất hết, để khắc phục điều đó ta có đối tượng prompt nó sẽ giúp chúng ta trước khi back hay chuyển trang nó sẽ xác nhận xem là chúng ta có chắc chắn muốn back hay chuyển trang không!

```
import {Prompt} from 'react-router-dom';  
<Prompt  
  when={true} // true | false  
  message={ (location) => (`Ban chac chan muon di den  
    ${location.pathname}`) }  
</>
```

Thực hành 1

Xây dựng router chuyển trang giữa ba trang Home, About, Contact

React Router Hooks

React router có một vài hooks để giúp chúng ta có thể truy cập vào trạng thái của các bộ định tuyến (router) và thực hiện điều hướng từ ngay bên trong mỗi thành phần.

Từ react 16.8, chúng ta có thể sử dụng một vài hooks sau:

- **useHistory**
- **useLocation**
- **useParams**
- **useRouteMatch**

useHistory

Hook **useHistory** cung cấp cho bạn quyền truy cập vào phiên bản lịch sử mà bạn có thể sử dụng để điều hướng.
(Thay thế bằng useHistory, ở demo 2 github)

useLocation

useLocation trả về đối tượng vị trí đại diện cho URL hiện tại. Bạn có thể nghĩ về nó giống như useState trả về một vị trí mới bất cứ khi nào URL thay đổi.

useParams

useParams trả về một đối tượng gồm các cặp khóa / giá trị của tham số URL. Sử dụng nó để truy cập `match.params` của **<Route>** hiện tại.

More hooks?

<https://v5.reactrouter.com/web/api/Hooks>

Thực hành 2

Xây dựng router sử dụng history để truyền categoryId từ Component Category sang Component Product.

Thực hành 3

Xây dựng router sử dụng history để truyền categoryId từ Component Category sang Component Product.