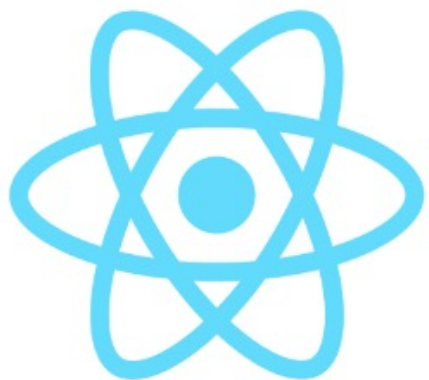


PHẦN IV

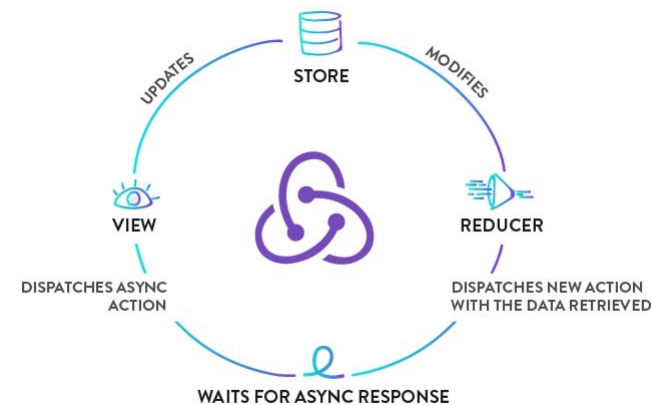
React Router & API



+



REACT **ROUTER**



❑ Routing (react-router-dom)

Routing là gì ?

- ❖ Routing là cơ chế trong single page giúp ta chuyển đổi qua lại giữa các component
- ❖ Để sử dụng được routing với reactjs, ta cần package hỗ trợ đó là React-router-dom
- ❖ Tiến hành cài đặt: `npm install --save react-router-dom`
- ❖ Sử dụng: tại `app.js`, ta đang có 2 component không thể hiện cùng lúc, đó là danh sách khóa học, và danh sách sinh viên . Do đó ta sẽ dùng `routing` để quản lý, dựa theo đường dẫn url để hiển thị component tương ứng.

❑ Routing (react-router-dom)

npm i react-router-dom

- Sử dụng Component BrowserRouter bao phần nội dung muốn hiển thị khi path trùng với url.
- Thẻ Route cấu hình path trùng với url thì sẽ hiển thị component đó.
- Nếu component không được đặt trong thẻ Route thì luôn được hiển thị.

Layout tham khảo

<https://startbootstrap.com/previews/shop-homepage/>

<https://startbootstrap.com/templates/small-business/>

```
import React from 'react';
import Footer from './components/Footer/Footer';
import Header from './components/Header/Header';
import About from './pages/About/About';
import Contact from './pages/Contact/Contact';
import Home from './pages/Home/Home';

import { BrowserRouter, Route } from 'react-router-dom'

function App() {
  return (
    <BrowserRouter>
      <div className="mb-5">
        <Header />
      </div>
      <div className="p-5">
        <Route exact path="/home" component={Home} />
        <Route exact path="/contact" component={Contact} />
        <Route exact path="/about" component={About} />
      </div>
      <Footer />
    </BrowserRouter>
  );
}

export default App;
```

❑ Routing (react-router-dom)

➤ NavLink (Hoặc <Link>)

- Thư viện react router dom hỗ trợ ta thẻ **NavLink** thay thế cho thẻ **<a>** với **href** đổi thành **to**, giúp cho ta có thể chuyển đổi qua lại giữa các trang mà không cần load lại toàn bộ html của trang đó.
- Ngoài ra **<NavLink>** còn cho phép ta custom link css, khi người dùng gõ đúng đường dẫn thì thẻ NavLink sẽ được active class hoặc style thông qua thuộc tính tương ứng **activeClass**, **activeStyle**.

```
<NavLink
  exact
  activeStyle={{color:red}}
  activeClassName="my-link">
  Cyberlearn.vn
</NavLink>
```

❑ Routing (react-router-dom)

➤ Chuyển đổi component

- Các component được load trên <Route> sẽ có thêm 3 thuộc tính props:
 - + history
 - + match
 - + location

❏ Routing (react-router-dom)

➤ History

- Giúp ta có thể di chuyển giữa qua lại giữa các route. Bằng nhiều phương thức khác nhau như push hay goBack. Ngoài ra nó còn hỗ trợ ta lấy các giá trị tham số thông qua thuộc tính search.
- Lấy ví dụ để link qua 1 trang sau khi cần xử lý 1 cái gì đó, ta không thể dùng thẻ NavLink được vì khi bấm vào thẻ navlink react sẽ tự động chuyển hướng đến liên kết route đó mà không thể chèn thêm bất kì xử lý gì.

```
import React from 'react'

export default function Home(props) {

  //Sử dụng trong react function
  console.log(props.history)

  return (
    <div>
      trang chủ
    </div>
  )
}
```

Sử dụng trong react function

```
export default class Home extends Component {

  //Xử dụng trong react class component
  render() {
    console.log(this.props.history);

    return (
      <div>

      </div>
    )
  }
}
```

Sử dụng trong react class

❑ Routing (react-router-dom)

➤ Match

- Đối tượng match cung cấp một số thuộc tính hỗ trợ: path hiện tại, tham số được truyền qua url...
- Ở đây ta sẽ sử dụng thuộc tính params để thực hiện chức năng xem chi tiết khóa học



❏ Routing (react-router-dom)

➤ Location

- Tương tự thuộc tính location từ history cung cấp các thông số

```
▼ location: {hash: "", pathname: "/home", search: "?ma=abc", st...}  
  hash: ""  
  pathname: "/home"  
  search: "?ma=abc"  
  state: undefined
```


❏ Routing (react-router-dom)

➤ Redirect

- Component Redirect được cung cấp bởi react-router-dom hỗ trợ ta điều hướng từ path này tới path khác

```
export default function Home(props) {  
  
  console.log(props.location)  
  
  return (  
    <div>  
      trang chủ  
      <Redirect to='/contact' />  
    </div>  
  )  
}
```

- Khi nội dung component gặp đối tượng component <Redirect /> thì sẽ chuyển hướng về path tương ứng.

□ Prompt

- ❖ Component Prompt cho phép người dùng xác định trạng thái cho phép người dùng rời khỏi trang hay không.
- ❖ Thuộc tính:
 - + when: trả về giá trị true hộp thoại hiển thị lên ngược lại false không hiển thị.
 - + message: Hộp thoại chứa nội dung hiển thị

```
import React, { Component } from 'react'
import { Prompt } from 'react-router-dom';
export default class login extends Component {
  state = {
    isSaveForm:true,
    TaiKhoan:'',
    MatKhai:''
  };

  Login = (e) => {
    e.preventDefault();
    console.log(1);
  }

  render() {
    return (
      <div className="container">
        <h3 className="display-4">Đăng nhập</h3>
        <form onSubmit={this.Login}>
          <div className="form-group">
            <label htmlFor="exampleInputEmail1">Tên khoản</label>
            <input type="email" className="form-control" name="TaiKhoan" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Tên khoản" />
          </div>
          <div className="form-group">
            <label htmlFor="exampleInputPassword1">Password</label>
            <input type="password" className="form-control" name="MatKhai" id="exampleInputPassword1" placeholder="Password" />
          </div>
          <button type="submit" className="btn btn-primary">Đăng nhập</button>
        </form>
        <Prompt
          when = {this.state.isSaveForm}
          message={location => ('Bạn có chắc muốn rời khỏi trang này ?')}
        />
      </div>
    )
  }
}
```