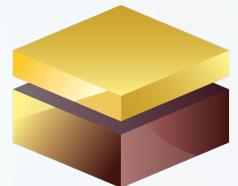
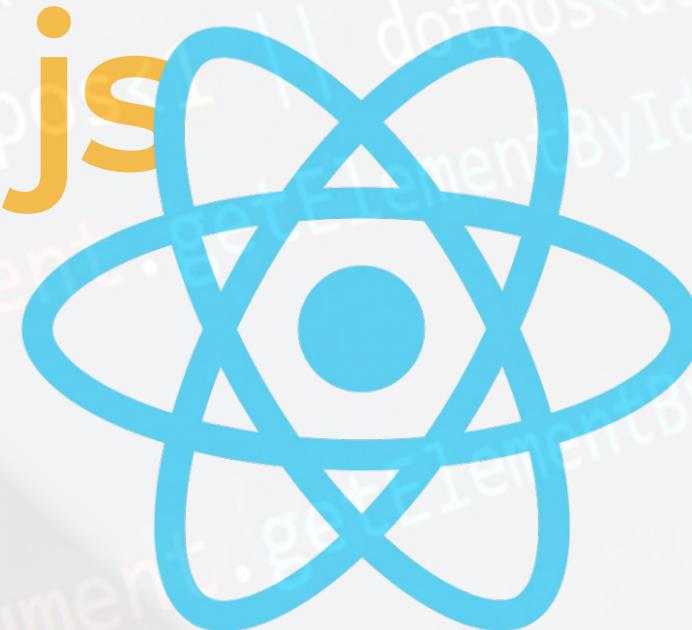
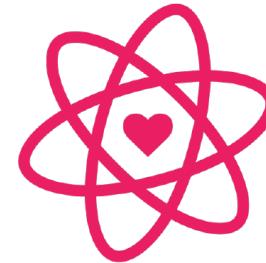


React



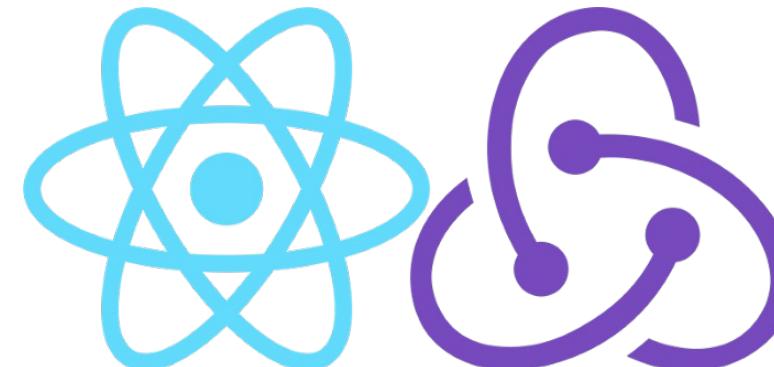
CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Hướng dẫn : Trương Tấn Khải



React JS

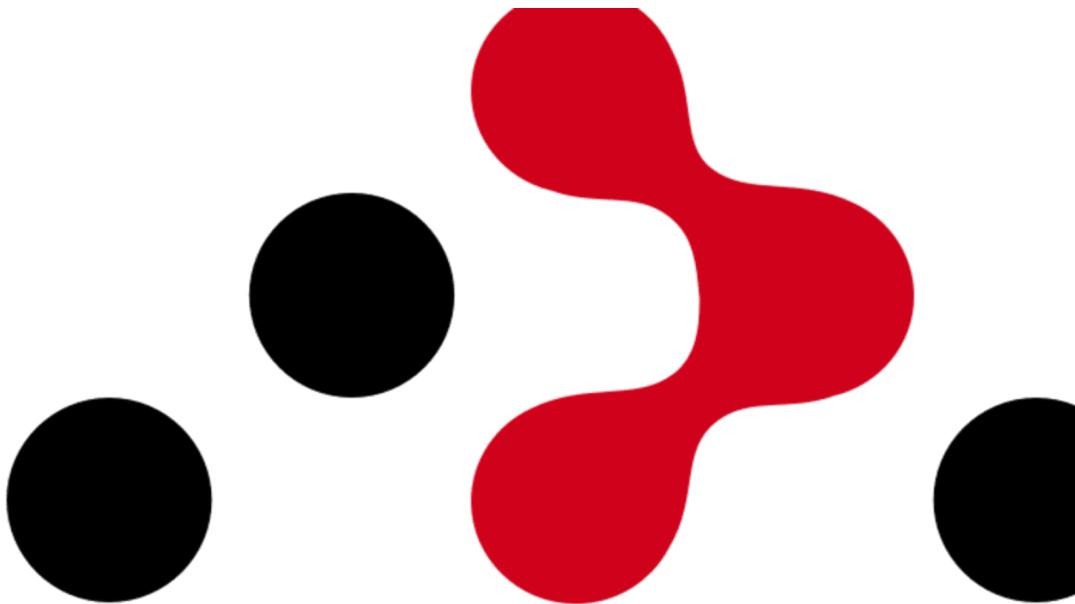
Functional Component
React hooks



☐ Các hook routing thông dụng

React hook routing cơ bản

- + use Navigate
- + use Params
- + use History
- + useSearchParams
- +



❑ useNavigate (React router dom – hook)

- Giúp ta chuyển hướng trang sau khi xử lý 1 hành động (handleSubmit api, handleChange ...)
- Ví dụ bên dưới với chức năng giả lập đăng nhập api sau thời gian 3s xác thực rồi mới chuyển hướng trang.
- Cú pháp:

```
const navigate = useNavigate();

//Chuyển hướng

navigate('path')
```

```
import React, { useRef } from 'react'
import { history } from '../../App';
import { useNavigate } from "react-router-dom";

export default function ReactForm(props) {
  const navigate = useNavigate();
  // console.log(props)
  const userLoginRef = useRef({
    userName:'',
    passWord:''
  });
  const handleChange = (e) => {
    const {value,id} = e.target;
    userLoginRef.current[id] = value;
    console.log(userLoginRef.current);
  }

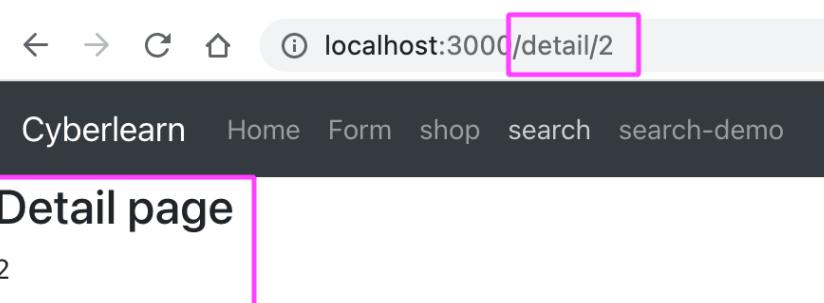
  const handleSubmit = async (e) => {
    e.preventDefault(); //Chặn reload browser
    console.log(userLoginRef.current);
    let promise = new Promise (resolve => {
      setTimeout(()=> {
        console.log('Đăng nhập api')
        resolve('Đăng nhập thành công');
      }, 3000);
    })
    let result = await promise;
    console.log(result);
    navigate('home');

    //Có replace : là k lưu lại lịch sử khi back trang
    //Không có replace là lưu lại lịch sử khi back trang
  }

  return (
    <form className='container' onSubmit={handleSubmit}>
      <h3>Login</h3>
      <div className='form-group'>
        <p>username</p>
        <input className='form-control' id="userName" onChange={handleChange} />
      </div>
      <div className='form-group'>
        <p>password</p>
        <input className='form-control' id="passWord" onChange={handleChange} />
      </div>
      <div className='form-group'>
        <button className='btn btn-success'>Login</button>
      </div>
    </form>
  )
}
```

❑ useParams (React router dom – hook)

- Giúp ta lấy được các tham số trên url của trình duyệt.
- Để định cài đặt param trên url như hình thì ta setup như code demo bên phải.



```
38   <Route path='detail'>
39     <Route path=':id' element={<Detail />}></Route>
40   </Route>
```

Định nghĩa <Route />

```
import React from 'react'
import { useLocation, useMatch, useParams } from 'react-router-dom'

export default function Detail(props) {
  const params = useParams(); //Dùng useParams để lấy param
  const match = useLocation();

  console.log('id', params.id)
  console.log(match);

  return (
    <div>
      <h3>Detail page</h3>
      <p>{params.id}</p>
    </div>
  )
}
```

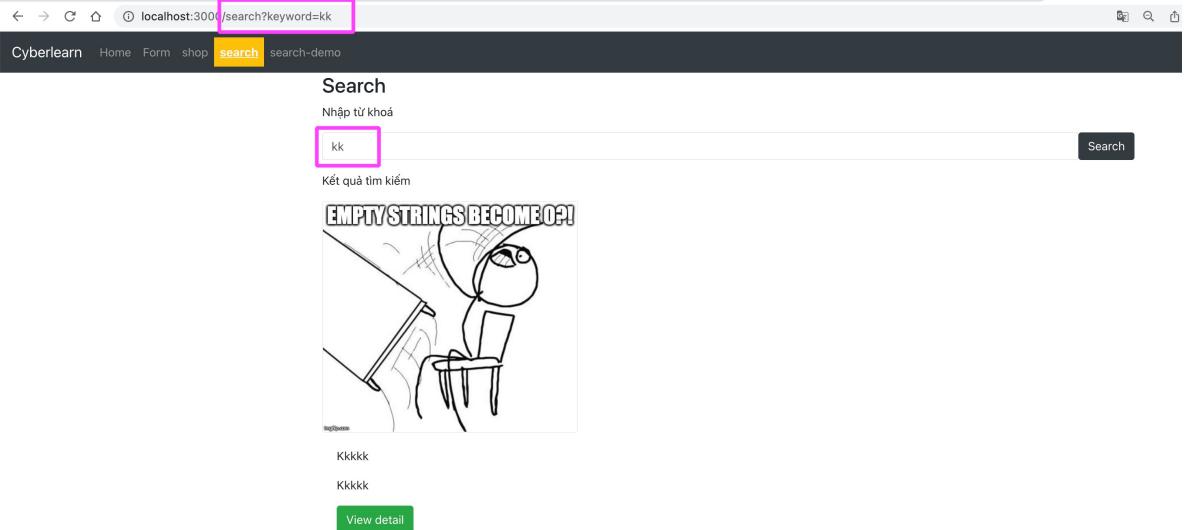
Sử dụng useParams để lấy id

Ngoài ra còn có thể sử dụng useLocation để lấy thêm thông tin

❑ useSearchParams (React router dom – hook)

- Tương tự như cách truyền tham số qua url thì react-router-dom cho phép ta định nghĩa tham số trên url của trình duyệt thông qua `?QueryParam`.
- Điều này thường được áp dụng trên các trang tìm kiếm để tạo link động dựa trên kết quả người dùng nhập và có thể lưu trữ lại link đó để share hoặc gửi cho người khác.

Demo: <https://codesandbox.io/s/usesearchparams-2f8trk?file=/src/App.js>



❑ Custom hook

- Ở hook có 1 số nguyên tắc cơ bản (Ví dụ: cannot be called inside a callback...). Vì vậy để sử dụng được HOC hoặc kế thừa như class component đôi lúc chúng ta sẽ lỗi vì phạm nguyên tắc. Vì vậy để tiện lợi cho lập trình viên thì muốn tái sử dụng lại các logic mà ta đã xây dựng cho 1 Functional component ta có thể tận dụng Custom hook.
- Cách khai báo cũng tương tự react component tuy nhiên kết quả trả về của function custom hook sẽ thay vì jsx thì custom hook trả về giá trị.
- Bên phải là demo về hook Route.

+ Thay vì ở component nào chúng ta cũng phải import các hook như useParams, useLocation,... Thị ta có thể tách logic này ra 1 custom hook để component nào cần sử dụng các hook này ta chỉ việc khai báo và lấy ra sử dụng .

+ Một số thư viện custom hook tham khảo

<https://github.com/streamich/react-use?fbclid=IwAR3R1tQBNTKowvRT60CDIjnIstcfGUnMewyi7R3yP2QSLBUJH1EgTg5hzxE>

```
import React from 'react'
import { useLocation, useNavigate, useParams, useSearchParams } from 'react-router-dom'

export default function useRoute() {
  const params = useParams();
  const match = useLocation();
  const search = useSearchParams();
  const navigate = useNavigate();
  return { params, match, search, navigate };
}
```

```
git clone https://github.com/streamich/react-use.git
cd react-use
git checkout custom-hooks
cd custom-hooks
git checkout useRoute
cd ..
cd examples
cd routes
cd custom-hook
cd DemoCustomHook
code .
```

```
1 import React from 'react' 6.9k (gzipped: 2.7k)
2 import useRoute from './useRoute'
3
4 export default function DemoCustomHook(props) {
5
6   const { match, params, search, navigate } = useRoute();
7   console.log({ match, params, search, navigate })
8
9   return (
10     <div>
11       <h3>DemoCustomHook</h3>
12     </div>
13   )
14 }
15
16
17
18
```