

React Events

1 Lý thuyết:

Cách thao tác event trong React tương tự như event trong element của HTML.

Giữa event React và event HTML khác nhau cơ bản:

- Sử dụng *camel case* cho tên của event thay vì *lowercase* như HTML
Ex. onClick, onMouseOver...
- Giá trị của event là một function thay vì một chuỗi string như HTML

Trong HTML

```
<button onclick="onLoginHandle()">
  Sign In
</button>
```

Trong React

```
<button onClick={this.onLoginHandle}>
  Sign In
</button>
```

Prevent Default Behavior:

Trong React để thực hiện *prevent default behavior*, phải call tới method **preventDefault** của event handle.

Để thực hiện prevent default trong HTML, ta có thể return về false.

```
<a href="#" onclick="console.log('Hi there!'); return false;">
  Click me
</a>
```

Ngược lại trong React, ta chỉ thực hiện được prevent default thông qua method **preventDefault**:

```

onHandleClick(e) {
  e.preventDefault();
  console.log('Hi there!.');
}
render () {
  return (
    <a href="#" onClick={this.onHandleClick}>
      Click me
    </a>
  );
}

```

Để tạo ràng buộc giữa event handle và Component của React, ta sử dụng method **bind**. Mặc định sẽ không có sự ràng buộc giữa event handle và Component, sự ràng buộc này được thể hiện ở điểm ta sẽ không call được những method đã define của Component trong event handle khi không sử dụng method **bind**.

```

validate() {
  return true;
}
onHandleClick(e) {
  this.validate();
  e.preventDefault();
  console.log('Hi there!.');
}
render () {
  return (
    <a href="#" onClick={this.onHandleClick.bind(this)}>
      Click me
    </a>
  );
}

```

Truyền Tham Số Trong Event Handle:

Có 2 cách truyền tham số vào event handle

- Sử dụng **arrow function**

```
<button onClick={(e) => this.deleteRow(id, e)}>Delete Row</button>
```

- Sử dụng phương thức **bind**

```
<button onClick={this.deleteRow.bind(this, id)}>Delete Row</button>
```

2 Bài Tập:

2.1 BÀI TẬP 1

2.1.1 Thời lượng: 10 phút

2.1.2 Mô tả bài toán:

Thực hiện event Click và Enter cho Component LoginForm.



The image shows a login form with the following elements:

- Email**: A text input field with the placeholder text "Email".
- Password**: A text input field with the placeholder text "Password".
- ☐ **Remember me**: A checkbox followed by the text "Remember me".
- Sign in**: A button with the text "Sign in".

2.1.3 Hướng dẫn:

- Sử dụng method **bind** cho event **onClick** và **onKeyUp**
- `event.keyCode == 13` để kiểm tra người dùng nhấn phím Enter

2.2 BÀI TẬP 2

2.2.1 Thời lượng: 10 phút

2.2.2 Mô tả bài toán:

Thực hiện event click cho menu logout của Component Header



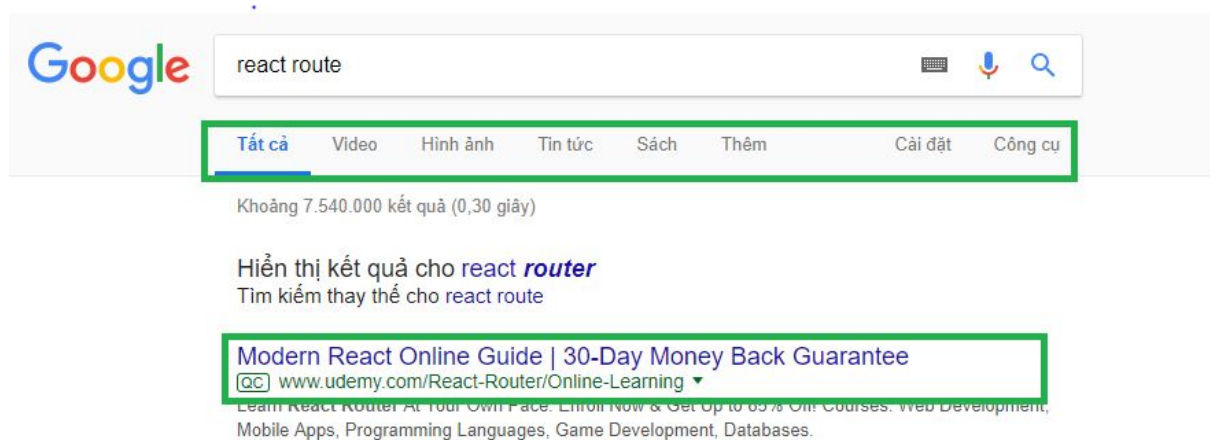
2.2.3 Hướng dẫn:

- Sử dụng method **bind** cho event **onClick**

React Router

1 Lý thuyết:

Router được gọi là điều hướng, là cách điều hướng từ một nội dung của trang web tới một nội dung khác của trang web, có thể là cùng một hệ thống web hoặc khác hệ thống.



Ví dụ trên là những link điều hướng tới nội dung của trang **google.com** bao gồm điều hướng trong (Tất cả, Video, Hình Ảnh...) và ngoài (Link kết quả tìm kiếm) website **google.com**.

Trong React, Router dùng để điều hướng nội dung trong trang web theo cách điều hướng từ một Component qua một Component khác.

Requirements

- Package react-router-dom
npm i --save react-router-dom
- Package react-loadable
npm i --save react-loadable

How to use

References: <https://reactjs.org/docs/code-splitting.html>

2 Bài Tập:

1. BÀI TẬP 1

1.1. Thời lượng: 20 phút

1.2. Mô tả bài toán:

Tạo **AppRouter** component.

1.3. Các bước thực hiện:

Tạo **AppRouter** component: **src/Component/AppRouter.js**

```
import React from 'react';
import { BrowserRouter as Router, Route, Switch } from
'react-router-dom';
import Loadable from 'react-loadable';

const Loading = () => <div>Loading...</div>;

const Login = Loadable({
  loader: () => import('./LoginForm'),
  loading: Loading,
});

const Home = Loadable({
  loader: () => import('./Home'),
  loading: Loading,
});

class AppRouter extends React.Component {
  render() {
    return (
      <Router>
        <Switch>
          <Route key={1} exact path="/"
component={Home}/>
          <Route key={2} path="/login"
component={Login}/>
        </Switch>
      </Router>
    );
  }
}
```

```
}  
export default AppRouter;
```

Tạo **Home** component: *src/Component/Home.js*

```
import React from 'react';  
import Header from '../Header';  
  
class Home extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {logged: true};  
  }  
  render() {  
    return (  
      <div>  
        <Header logged={this.state.logged}  
onClick={ (e) => {} } />  
      </div>  
    );  
  }  
}  
export default Home;
```

2. BÀI TẬP 2

2.1. Thời lượng: 15 phút

2.2. Mô tả bài toán:

Sử dụng AppRouter vừa tạo trong bài tập 1 để routing nội dung trang web khi click vào button **sign in** của **LoginForm** component và menu **logout** của **Header** component.

2.3. Các bước thực hiện:

Chỉnh sửa file *src/Component/App.js*

```
import React from 'react';  
import AppRouter from '../Components/AppRouter';  
import './App.css';
```

```

class App extends React.Component {
  constructor(props) {
    super(props);
  }

  render() {
    return (
      <div>
        <AppRouter />
      </div>
    );
  }
}
export default App;

```

Chỉnh sửa file ***src/index.js***

```

import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import registerServiceWorker from
'./registerServiceWorker';

import App from './App';

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

```

Chỉnh lại method **onSubmit** trong **LoginForm** component

```

//Comment onShow thay bằng window.location.href
//this.props.onShow();
window.location.href="/";

```

Chỉnh sửa method **onLogout** trong **Header** component


```
onLogout(event) {  
  window.location.href = "/login";  
  //this.props.onClick();  
}
```