

Tìm hiểu về React

Author: Nam Lê (nam@brse.work) Created At: 2018/11/14

1. React là gì

`react` là một thư viện JavaScript để xây dựng giao diện người dùng. Nó được duy trì bởi Facebook và một cộng đồng các nhà phát triển và một số công ty. `react` thường được sử dụng để phát triển ứng dụng web (`reactjs`) hoặc ứng dụng cho các thiết bị di động (`react native`).

`react` thích hợp để học cho các lập trình viên hơn là người dùng cuối, vì nó không dễ dàng để thiết lập trang web nhanh chóng mà không cần code nhiều.

`react` dùng làm ứng dụng client trong các hệ thống phi tập trung. Tức là ứng dụng `react` của bạn đơn giản chỉ lấy dữ liệu, hiển thị dữ liệu, xử lý với một kịch bản nào đó và gửi yêu cầu lên cho ứng dụng phía server. Và lúc đó ứng dụng server của bạn có thể là Java Spring, Python, Golang, C++, hay bất kỳ ngôn ngữ server nào có thể giao tiếp qua http port.

```
[React-App]Web    <--> [Server application]
[React-App]Mobile <--> [Server application]
```

2. Một số khái niệm cơ bản

NPM viết tắt của **Node Package Manager** là một công cụ (chương trình) quản lý các thư viện lập trình **Javascript** cho **Node.js**

YARN là công cụ quản lý thư viện javascript mã nguồn mở tốc độ cao, tin cậy và bảo mật nhằm thay thế NPM (Node Package Management). Được hình thành bởi các kỹ sư Facebook, Google, Exponent và Tilde, và đã được sử dụng thực tế mức Production tại các công ty trên.

Để cài đặt các thư viện cho `react` app của bạn, chúng ta thường sử dụng một trong hai bộ quản lý thư viện trên.

`package.json` là file được sử dụng để khai báo thông tin trong ứng dụng. Ví dụ: tên ứng dụng là gì, tác giả là ai, source management server ở đâu, các thư viện cần thiết là gì, các command để chạy ứng dụng.

`babel` hoặc `babel.js` là trình biên dịch JavaScript miễn phí và mã nguồn mở. `babel` giúp biên dịch ngôn ngữ lập trình hoặc ngôn ngữ đánh dấu thành Javascript mà các trình duyệt có thể hiểu được (thực thi được). Trong ứng dụng `reactjs` thì `babel` sẽ giúp chúng ta compile từ file JS(ES2015) hoặc JSX trở thành JavaScript(ES5) và mã hóa chúng.

JSX (JavaScript extension) Được sử dụng mặc định trong `reactjs`, nó là một định dạng tổng hợp của JS và XML, giúp cho người lập trình dễ dàng hơn trong việc tạo source code. Khi thực thi JSX sẽ được dịch thành JS.

```
<Header bg-color="red" />
/* Sẽ được biên dịch thành JS bên dưới */
React.createElement(
  Header, {
    bg-color:"red"
  }
);
```

`create-react-app` là công cụ để tạo ra thư mục có sẵn những thứ cần thiết của `react` app. Công cụ này được cung cấp bởi công ty Facebook.

`webpack` là phần mềm mã nguồn mở dùng để đóng gói JavaScript. Từ những file JS đơn lẻ webpack sẽ gom lại thành 1 package để dễ dàng triển khai. Khi dùng webpack thì ứng dụng cần có file cấu hình cho webpack đó là `webpack.config.js`.

3. Xây dựng ứng dụng `reactjs` sử dụng `create-react-app`

Để chạy được ứng dụng `reactjs` thì máy của bạn phải đã được cài `node.js`.

Phiên bản tối thiểu là **Node 8.10.0** (hiện tại là Node 11.1.0). Hướng dẫn cài đặt Node sẽ được trình bày ở 1 bài viết khác.

Sau khi đã có node thì chúng ta cần có `npm` hoặc `yarn`. Hướng dẫn cài đặt `npm` hoặc `yarn` sẽ được trình bày ở 1 bài viết khác.

```
#npm
npm init react-app <app-name>
#yarn
yarn create react-app <app-name>

# thay thế cụm <app-name> bằng tên ứng dụng của bạn
```

Sau khi chạy 1 trong 2 command trên thì thư mục có tên là tên ứng dụng của bạn đặt sẽ được tạo ra.

Để chạy ứng dụng thì thông qua command line tool (cmd trên windows, terminal trên unix/macos)

```
# Nếu khởi tạo ứng dụng bằng npm
npm start

# Nếu khởi tạo ứng dụng bằng yarn
yarn start
```

Ở mặc định thì ứng dụng sẽ sử dụng cổng 3000. Sau khi chạy command trên thì trình duyệt mặc định sẽ được tự động mở và tự động chuyển đến địa chỉ <http://localhost:3000>

4. Xây dựng ứng dụng `reactjs` sử dụng `webpack`

Việc tự xây dựng ứng dụng bằng webpack hơi phức tạp hơn một chút.

Step 1: Trước tiên cần tạo một thư mục để lưu trữ source code của dự án.

```
mkdir client-app
cd client-app # chuyển đến thư mục chứa source code
npm init # khởi tạo ứng dụng
```

Sau khi nhập lệnh `npm init` thì màn hình terminal sẽ yêu cầu nhập một số thông tin cơ bản của ứng dụng như: tên ứng dụng, phiên bản, license ...

Sau khi nhập thông tin xong, file `package.json` sẽ được tạo ra ở trong thư mục client-app.

Về kiến trúc hệ thống tôi đề xuất cấu trúc như sau:

```
client-app/
-----/dist/ #thư mục này sẽ đưỡ tạo tự động khi build
-----/public/ # chứa file html/css/img/js tĩnh
-----/-----/ img /
-----/-----/ index.html

-----/ src / # chứa source code
-----/ ----/ Index.js
-----/ ----/ App.js
-----/ ----/ App.scss
-----/ package.json
-----/ webpack.config.js
...
```

Step 2: cài đặt thư viện cần thiết.

cài đặt webpack và cli

```
npm i webpack webpack-cli -D
npm i react react-dom -S
npm i @babel/core @babel/preset-react @babel/preset-env babel-loader -D
npm i css-loader style-loader -D
npm i html-webpack-plugin -D
npm i webpack-dev-server -D
```

Update scripts cho package.json

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "start": "webpack-dev-server --mode development --open --port 9999",
  "build": "webpack --mode production"
},
```

Tạo file webpack.config.js

```
const HtmlWebpackPlugin = require("html-webpack-plugin");

const htmlWebpackPlugin = new HtmlWebpackPlugin({
  template: "./public/index.html",
  filename: "./index.html"
```

```

});

module.exports = {
  module: {
    rules: [
      {
        test: /\.js$/,
        exclude: /node_modules/,
        use: {
          loader: "babel-loader"
        }
      },
      {
        test: /\.css$/,
        use: ["style-loader", "css-loader"]
      }
    ]
  },
  plugins: [htmlWebpackPlugin]
};

```

Tạo file `.babelrc` với nội dung bên dưới

```

{
  "presets": ["@babel/preset-env", "@babel/preset-react"]
}

```

Trong thư mục public tạo file index.html với nội dung như sau

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>brse.work</title>
  </head>
  <body>
    <section id="App"></section>
  </body>
</html>

```

Trong thư mục src tạo file App.js với nội dung sau:

```
import React, { Component } from 'react';

export default class App extends Component {
  render() {
    return(
      <div>
        <h3>Hello <a href="http://brse.work">brse.work</a></h3>
      </div>
    );
  }
}
```

Trong thư mục src tạo file index.js với nội dung sau:

```
import React from 'react';
import ReactDOM from 'react-dom';

import App from './App';

ReactDOM.render(<App />, document.getElementById("App"));
```

Để tạo package chúng ta sử dụng command `npm run build`

Còn để chạy thử ngay trên máy trong lúc phát triển chúng ta dùng command `npm run start` hoặc `npm start`

Với config trong file package.json như trên thì khi build sẽ tạo ra thư mục `dist` trong thư mục này sẽ chứa các file `index.html` và `main.js`. Còn khi sử dụng command start thì nó sẽ mở trình duyệt tự động.

Lưu ý Khi các bạn làm theo hướng dẫn mà chạy gặp lỗi thì có khả năng khi cài thư viện có một số không tương thích, hoặc thư viện sử dụng trong bài đã update với các hàm mới hẳn.

Source code demo các bạn có thể tải ở link sau: <https://github.com/brsework/react-webpack-sample-01>.