# Node js : cần phải bắt buộc tải để hỗ trợ công nghệ react js

Link video : https://www.youtube.com/watch?v=ysjJlvQ3FFc&list=PL\_-VfJajZj0UXjlKfBwFX73usByw3Ph9Q&index=22

Node js là 1 javascripts run time nên tạo sẵn 1 môi trường độc lập để có thể thực thi được code js ở trong đó mà không cần sử dụng những môi trường quen thuộc như trình duyệt nữa (live server).

Trình duyệt để chạy được javascripts là một môi trường độc lập khác và Node js cũng có thể chạy được javascripts thì nó lại là 1 môi trường độc lập khác không liên quan gì tới trình duyệt cả .Vì tạo được 1 môi trường độc lập như vậy nên ngta gọi nó là javascripts run time

Chrome’s V8 Javascript engine (được phát triển bởi google ) là cốt lõi của Node js để tạo môi trường chạy js (javascripts run time) .Ngoài ra ,nó cũng là cốt lõi của trình duyệt chrome mà các bạn đang xài ..

Với vai trò là 1 dev fe ,sử dụng react để tạo 1 máy chủ chạy trên máy tính cá nhân .Xưa này hay xài máy chủ ảo đó chính là extension trên vs code đó là live server

Làm sao để tiết kiệm được khoảng thời gian để viết code ,làm sao để dễ dàng cài đặc được những thư viện cần thiết ,làm sao để không bị lặp đi lặp lại những thao tác quen thuộc

Luôn phải làm đi làm lại việc liên kết react ,liên kết react dom ,cài babel thì rất mất thời gian

=>> Node js sẽ giải quyết được những vấn đề này

## NPM

Vì trong hệ sinh thái của node có npm js để quản lý tất cả thư viện (viết bởi js) cho node

Những vấn đề mà bạn sẽ gặp phải trong tương lai thì đều có sẵn thư viện hỗ trợ trong npm js rồi

Khi cài node js thì npm js sẽ được cài theo nên chỉ cần download các thư viện có mặt ở trên npmjs.com

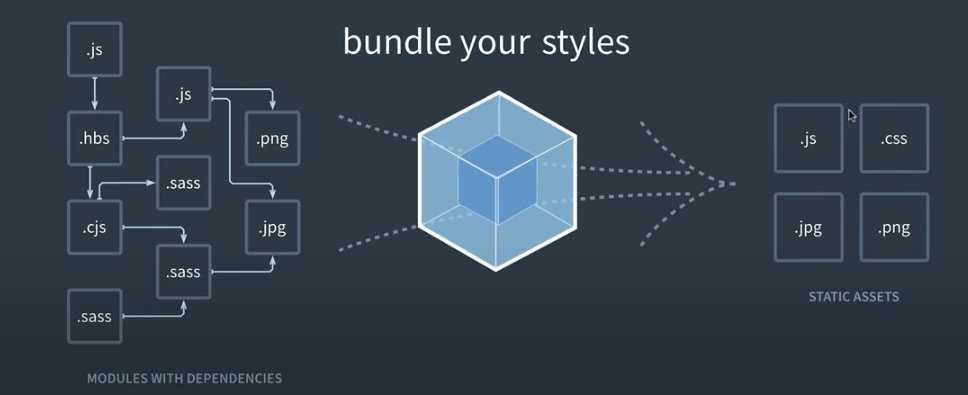
Khi có gói npm .,ko có dung thể script để liên kết )link) nữa mà import thẳng để sử dụng trong 1 file

## Thư viện react-creat-app

Facebook đã tạo ra thư viện này giúp nhanh chóng, dễ dàng tạo ra 1 project để làm website với thằng react js và nó đã kê sẵn những thứ “lặp đi lặp lại” như là react .react dom .babel và webpack (đều đã được cài sẵn rồi )

## Thư viện Webpack

Giúp chia nhỏ (MODUM HÓA ) file ,compile ra và kết hợp lại các file thành 4 file duy nhất để khi triển khai lên thực tế thì nó gọn ,ít giảm được thời gian kết nối ,giúp tang hiệu năng của ứng dụng .Ngoài ra còn plugin hỗ trợ file thành minifile (rút gọn tên biến thành 1 kí tự ,loại bỏ comment không cần thiết, khoảng trống giữa các dòng code từ đó làm giảm dung lượng file xuống ) giảm thời gian tải khi khách hang truy cập website



Cách cài đặc Node js => truy cập link : <https://nodejs.org/en/download> => tải về (.msi) => next x4

Kiểm tra cài đặt (trong terminal) : node –v và npm –v và npx –v

# Cấu trúc cơ bản của 1 project

React – webpack ( là tên 1 thư mục gốc ) nhớ nhen

Src (thư mục chứa src chính )

Components ( thư mục chứa components)

Index .js (File khởi tạo ,render ra app vào root)

Public

Index.html (html page ,nơi chứa các root element)

## Thử tạo 1 project tuân theo cấu trúc trên :

B1 : Tạo file (react-webpack) ,mở terminal trên vscode trỏ vào file này :

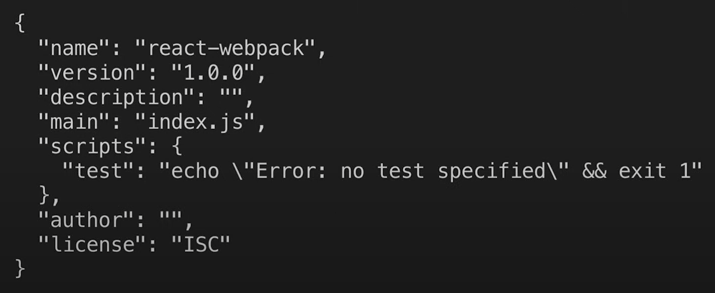
* Gõ : npm init

Khái niệm : câu lệnh này liệt kê mô tả project của bạn và tạo 1 file tên package.json

## Package.json

File này giúp các bạn mô tả dự án và quản lý các thư viện được cài đặt trong dự án

..

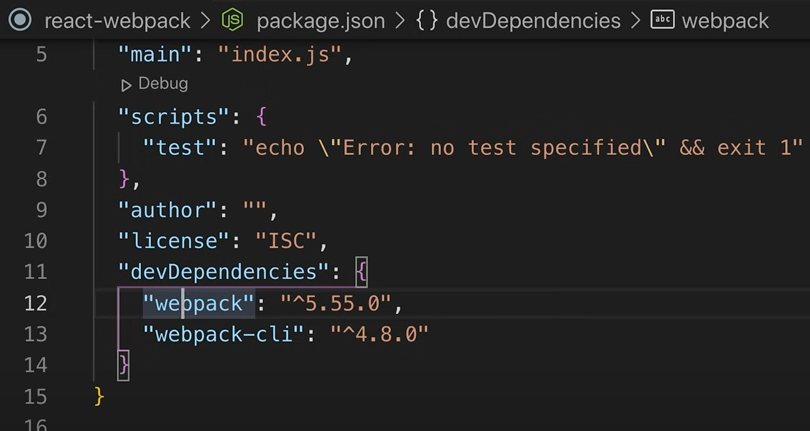


## Cài đặt thư viện web-pack

Chạy lệnh trong terminal như sau : npm install webpack webpack-cli –save-dev

Webpack và webpack-cli là tên thư viện bạn muốn cài đặt

Câu lệnh này có - - save-dev giúp tạo 1 giá trị tên là devDependencies nằm trong file package.json



Sau khi chạy xong câu lệnh trên ,nó sẽ tạo ra 2 thư mục dưới đây :

File Node\_modules : thư mục tên là node\_modules chứa những thực viện mà chúng ta cài

Rất rất nhiều thư viện nhỏ lẻ có liên quan (có cả 5 file webpack )

## File package-lock.json

## Cài đặt react và react-dom

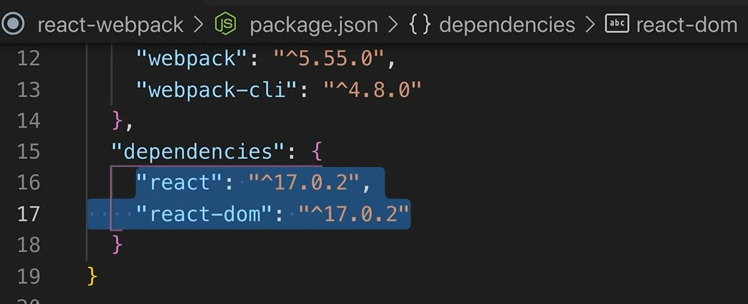
Chạy lệnh trong terminal như sau : npm install react react-dom --save

-- save : để lưu 2 thư viện này vào file package.json bằng giá trị Dependencies

Và --save-dev vào giá trị devDependencies (chỉ dành để lúc dev )

Và -- save vào giá trị Dependencies (dành cho lúc dev và lúc đưa lên trên production )

Lưu ý : 2 thứ này có tên là flag ,cờ save

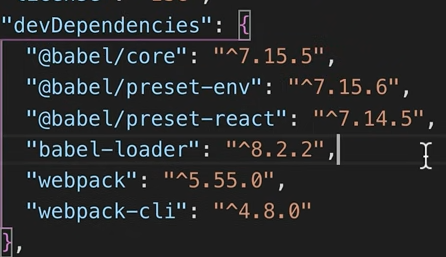


## Cài đặt Babel và 2 file code cơ bản nhất (html và js)

Chạy lệnh trong terminal như sau : npm install @babel/core babel-loader @babel/preset-env @babel/preset-react --save-dev

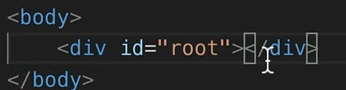
Các thư viện được cài đặt :

* **babel-core:** Chuyển đổi ES6 về ES5
* **babel-loader:** Cho phép chuyển các files Javascript sử dụng Babel & Webpack
* **babel-preset-env:** Cài đặt sẵn giúp bạn sử dụng Javascript mới nhất trên nhiều môi trường khác nhau ( hỗ trợ nhiều trình duyệt khác nhau). Gói này đơn giản là support truyển đổi ES6, ES7, ES8, ES... về ES5.
* **babel-preset-react:** Hỗ trợ chuyển đổi JSX về Javascript



Tiếp

Tạo file public => index.html => GÕ ! + TAB =>



Tạo file src/index.js => copy bỏ vô

import React from 'react' // nạp thư viện react

import ReactDOM from 'react-dom' // nạp thư viện react-dom

//ko cần link thẻ sripts để link nữa mà xài cú pháp es6 để import luôn ,phê vãi xoài

// Tạo component App

function App() {

return (

<div>

<h1>Xin chào anh em nhé</h1>

//lát render thẻ này ra là thành công

</div>

)

}

// Render component App vào #root element

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

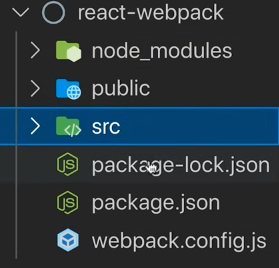
# Cấu hình cho webpack

## Cài đặt css-loader và style-loader

Css-loader và style-loader : 2 thư viện này giúp webpack có thể tải file .css dưới dạng module.

Chạy lệnh trong terminal như sau : npm install css-loader style-loader --save-dev

## Tạo file tên webpack.config.js trong thư mục gốc (React – webpack/ webpack.config.js)



File này giúp định tuyến vị trí file để build (cú pháp của es5 )

Tạo nhằm trong thư mục khác là ko chạy đâu nhé !

* bỏ code này vào file vừa tạo
* giải thích code dưới cặn kẽ (14:15) : <https://www.youtube.com/watch?v=1EBe-l1E3pM&list=PL_-VfJajZj0UXjlKfBwFX73usByw3Ph9Q&index=23>

const path = require("path");

module.exports = {

entry: "./src/index.js", // Dẫn tới file index.js ta đã tạo

output: {

path: path.join(\_\_dirname, "/build"), // Thư mục chứa file được build ra

filename: "bundle.js" // Tên file được build ra

},

module: {

rules: [

{

test: /\.js$/, // Sẽ sử dụng babel-loader cho những file .js

exclude: /node\_modules/, // Loại trừ thư mục node\_modules

use: ["babel-loader"]

},

{

test: /\.css$/, // Sử dụng style-loader, css-loader cho file .css

use: ["style-loader", "css-loader"]

}

]

},

// Chứa các plugins sẽ cài đặt trong tương lai

plugins: [

]

};

-------------------------------------------------------------------------------------------------------------

## Tạo file .babelrc (React – webpack/.babelrc)

File này giúp cấu hình cho thư viện Babel

Và dán code này vào file

{

"presets": [

"@babel/preset-env",

"@babel/preset-react"

]

}

## Thêm thẻ scripts cho dự án

Vì khi thêm node vào dự án thì thẻ này giúp bạn chạy chỉ bằng câu lệnh npm start hoặc npm run build

Start là để chạy project ,run build là chạy và có webpack hỗ trợ (quay lại hình ảnh compile và kết hợp )

Copy code này vào thẻ script có sẵn trong trong gói package.json

"scripts": {

...

"start": "webpack --mode development --watch",

"build": "webpack --mode production"

}

Giải thích code

--mode development là vẫn giữ cấu trúc project như cũ ,chưa compile

--watch là lắng nghe những sự thay đổi và build lại của “/src/file js” nằm trong entry trong file webpack.config.js

Bây giờ có lẽ là đã chạy dự án được rồi !! vào terminal => gõ npm start

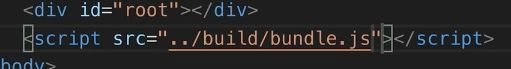
Vì nó sẽ tham chiếu vào trong start ở trên để chạy lệnh

Nó sẽ tự sinh ra 1 file build/bundle.js vì nó được cấu hình ở trong output trong file webpack.config.js

Hiểu nôm na cái luồng code là nó láy file index.js để đọc và build ra file bunder.js

## Chạy dự án đã xong ,bây giờ chỉ cẩn render nó ra

Đầu tiên : link file bundle vào file public/index.html



Và thử chạy nó bằng live server

Thứ 2 : Gio chạy không cần live server

## Cài đặt thư viện html-webpack-plugin

Giúp giải quyết vấn đề tự link file bundle vào index.html

Xóa bỏ đoạn thẻ script trên nhé

Và chạy lệnh trong terminal như sau : npm install html-webpack-plugin --save-dev

Sau đó mở terminal mới chạy npm start để xem nhé

Tiếp theo ta đi update cấu hình Webpack để thêm html-webpack-plugin vào dự án.

Tại webpack.config.js hãy thêm:

...

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

module.exports = {

...

// Chứa các plugins sẽ cài đặt trong tương lai

plugins: [

new HtmlWebpackPlugin({

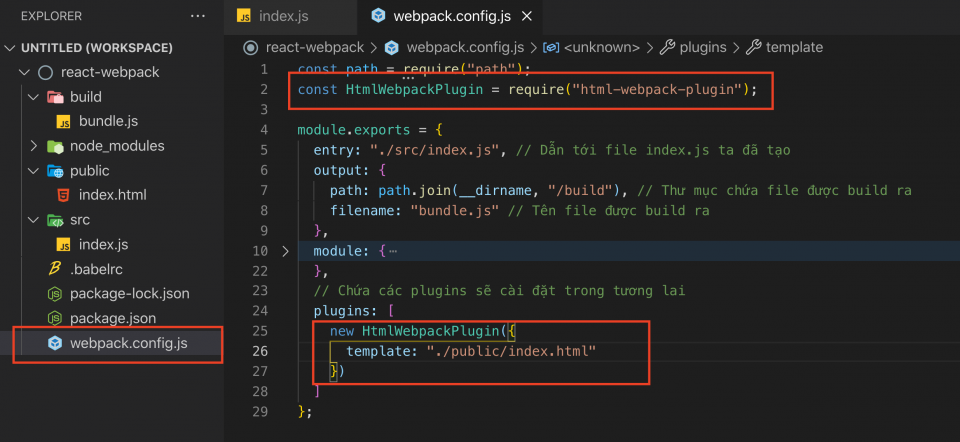
template: "./public/index.html"

})

]

};

Kết quả :



Khi chạy npm start nó sẽ tự sinh ra them 1 file build/index.html (file này chỉ để chạy ứng dụng thôi )

File public/index.html là để viết code nhé

Nghĩa là live server sẽ chạy File public/index.html

Còn đúng nghĩa thì chạy trong file build/index.html

<https://fullstack.edu.vn/blog/phan-1-tao-du-an-reactjs-voi-webpack-va-babel.html>

<https://www.youtube.com/watch?v=1EBe-l1E3pM&list=PL_-VfJajZj0UXjlKfBwFX73usByw3Ph9Q&index=23>

(24:17)