|  |
| --- |
| **WEBPACK** |
| **1. Khái niệm** |
| - Webpack là công cụ giúp bạn Compile ( NÉN ) các module javascript  Công việc này còn được gọi là **Module bundler** : Đóng gói các module  **Ưu điểm:**   * Giúp Project dễ dàng phát triển, quản lý, customize * Tăng tốc cho project * Phân chia các module và chỉ load khi cần * Đóng gói các file nguồn thành một file duy nhất * Biến các tài nguyên tĩnh trở thành một module * Chuyển đổi các mã nguồn thành ES5 thông qua Babel * Có thể chuyển các file CSS thành dạng module để có thể import và export |
| **2. Cài đặt** |
| 1. **npm i webpack**  2. **Tạo 1 file tên : webpack.config.js để config ( cấu hình ) cho Webpack**  const path = require('path');  const config = {  entry : './src/index.js', // File đầu vào  output : { // File đầu ra  filename : 'bundle.js', // Tên file đầu ra  path : path.join(\_\_dirname, '/build') // Nơi chứa file đầu ra  },  module: {  rules: [ ]  },  plugin: [    ]  }  module.exports = config;  **Cách hoạt động của Webpack như sau:**  - Tại **entry** - vị trí Webpack bắt đầu làm việc , nó sẽ bắt đầu thực hiện việc nạp file để bắt đầu bundle. Nó sẽ import từ module này đến module khác tạo ra một hệ thống việc load các file module js  - Tại **output** - vị trí Webpack lưu file sau khi compire  3. **Trong file package.json ta thêm script cho dự án:**  "scripts": {  ...  "start": "webpack --mode development --watch",  "build": "webpack --mode production"  }  **Khi chạy : npm start**  Webpack sẽ bắt đầu bundle file và tạo ra một file là bundle.js như sau |
| **3. module trong Webpack config** |
| **rules: [ ]**  **- Khai báo công việc xử lý các compile các ngôn ngữ khác nhau**  **- Cho phép xử lý 1 file trước khi import hoặc load vào**  **- Giúp chuyển đổi 1 file từ nhiều ngôn ngữ khác nhau sang js thuần**  **plugin**  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"]  }  ]  },  **Trong đó:**   * **test :** Chuỗi RegEx so khớp với các định dạng file , khi trình phân tích đi qua nếu thấy trùng nó sẽ nạp các file này vào để xử lý * **exclude:** Không sử dụng nếu trùng với định dạng file này, trình phân tích sẽ bỏ qua file này ( thường bỏ file node\_module đi) * **use :** Đây là thành phần quan trọng nhất, nó sẽ sử dụng các gói Loader mà bạn đã cài thông qua **npm module** để xử lý các file đã khai báo ở trên. * **option**: Nếu cần cài đặt một số thông số cho các Loader   - Ở trên ta thấy có sử dụng các gói Loader sau:  **babel-loader style-loader css-loader**  Vì vậy bạn phải cài những Module này vào npm:  npm i **babel-loader style-loader css-loader --save-dev** |
| 4. **Plugin trong webpack** |
| Plugin cũng giống các gói Loader ở trên nhưng nó có thể làm được nhiều thứ hơn. Một vài plugin hay sử dụng:   * **HTML Webpack Plugin**   - **Cài đặt**: npm i html-webpack-plugin --save-dev  - **Mô tả:** Tạo ra các tệp HTML để phục vụ việc đóng gói file, ở đây nó sẽ tự động thêm thẻ <script ra vào file index.html và tạo ra một file index.html nằm trong thư mục chứa các bundle.js  - Thêm vào plugin webpack như sau:  const HtmlWebpackPlugin = require("html-webpack-plugin");  module.exports = {  ...  // Chứa các plugins sẽ cài đặt trong tương lai  plugins: [  new HtmlWebpackPlugin({  template: "./public/index.html"  })  ]  }; |
| **5. webpack-dev-server module** |
| Webpack-dev-server sẽ chạy 1 web server kết hợp được Webpack và Node  **Cài đặt :** **npm install webpack-dev-server --save-dev**  **Thiết lập kịch bản chạy trong packaged.json :**  "scripts": {  ...  "start": "webpack-dev-server --mode development --open --hot",  ...  } |
| **6. Sử dụng Webpack làm dự án ReactJS** |
| **1. Tạo file dự án : react-webpack**  **2. Khởi tạo : npm init**  **3. Cài các module cần sử dụng vào NodeJS :**   * **“dependencies”**  1. **react** 2. **react-dom**   **npm i react react-dom**   * **“devDependencies”**   🡸 Webpack và các module hỗ trợ 🡺   1. **webpack** : Sử dụng webpack 2. **webpack-cli** : Cung cấp bộ lệnh làm việc với webpack 3. **webpack-dev-server** : Tạo WebServer sử dụng Webpack và Node   🡸 Các gói Loader để xử lý file trong Webpack 🡺   1. **css-loader** : Tải file CSS dưới dạng module 2. **style-loader** : Chèn CSS vào DOM 3. **babel-loader** : Chuyển file JavaScript sử dụng Babel   🡸 Các phần sử dụng trong thư viện babel để biên dịch JavaScript 🡺   1. **@babel/core** : chuyển đổi ES6 🡺 ES5 2. **@babel/preset-env** : Chuyển đổi ES6 , ES7, ES8… 🡺 ES5 3. **@babel/ preset-react :** Chuyền đổi JSX 🡺 JavaScript   🡸 Plugin hỗ trợ trong Webpack 🡺   1. **html-webpack-plugin** : Compile file HTML thành file HTML có gắn với file bundle.js   **npm i webpack webpack-cli webpack-dev-server css-loader style-loader babel-loader @babel/core @babel/preset-env @babel/preset-react --save-dev**  **4. Tạo file : react-webpack > public/index.html**  <!DOCTYPE *html*>  <html *lang*="en">  <head>      <meta *charset*="UTF-8">      <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">      <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">      <title>Document</title>  <script *defer* *src*="bundle.js"></script></head>  <body>      <div *id*="root">      </div>    </body>  </html>  **5. Tạo file: react-webpack > src/index.js**  - Đây là file cốt lõi mà Webpack sẽ lắng nghe tất cả các thay đổi, kết nối của nó để tạo thành 1 file duy nhất là **bundle.js**  const *React* = require('react');  const ReactDom = require('react-dom');  function App(){      return (          <div>              <h1> Xin chao anh em </h1>              <h1> tesst </h1>          </div>      )  }  ReactDom.render( <*App*/> , document.getElementById('root'))  **6. Cấu hình cho webpack**  **- Tạo file : react-webpack > webpack.config.js**  const path = require("path");  const HtmlWebpackPlugin = require('html-webpack-plugin');  *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: [  **new** HtmlWebpackPlugin({          template: "./public/index.html"        }),    ]  };  **7. Cấu hình cho Babel**  **- Tạo file : react-webpack > .babelrc**  *Việc này giúp Babel biết sẽ sữ dụng những phần nào đễ hỗ trợ chuyển đổi*  {  "presets": [  "@babel/preset-env",  "@babel/preset-react"  ]  }  **8. Tạo các kịch bản chạy cho NODEJS:**  **Tại file packed.json( ) thêm các đoạn script sau:**  "scripts": {      "start": "webpack-dev-server --mode development --open --hot",      "build": "webpack --mode production",    },  **9. Khởi chạy : npm start**  - Lúc này webpack biên dịch ( compile ) và đóng gói ( bundle ) từ file: src/index.js và xuất ra file build / bundle.js , đồng thời cũng tạo ra một file: build / index.html ( được kết hợp giữa file public/index.html và build/bundle.js)  - Sau khi quá trình này kết thúc sẽ tạo ra một WEB SERVER dựa trên Node và đọc file build / index.html |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |