1. Cách fix lỗi trùng port:

* npm kill-port PORT
* Khởi động lại máy tính.
* Chạy project tại port khác.

1. Cài đặt thư viện:

* Cài 1 package:

npm install <package-name>

* Cài 1 package với version cụ thể:

npm install <package-name>@<version>

1. Running taks:

npm run <task-name>

Ví dụ:

Với webpack có:

{

"scripts": {

"watch": "webpack --watch --progress --colors --config webpack.conf.js",

"dev": "webpack --progress --colors --config webpack.conf.js",

"prod": "NODE\_ENV=production webpack -p --config webpack.conf.js"

}

}

Thay vì phải gõ câu lệnh dài dòng như vậy để chạy, thay bằng:

$ npm run watch

$ npm run dev

$ npm run prod

+ 1 server có thể sử dụng nhiều ‘process’ tại 1 thời điểm, thì để định danh từng process đó thì ta sử dụng PORT (PORT có thể lên tới 65535)

+ ExpressJS chỉ là khung, còn một số framework khác ra đời dựa vào ExpressJS để phát triển hơn và loại bỏ mặt hạn chế của ExpressJS.

+ Ở frontend các bạn bắt buộc phải dùng thư viện babel, còn ở backend thì không cần thiết.

+ Muốn sử dụng View Engine cần 2 bước:

* Bước 1: Chỉ cho thằng Express phải kiếm cái View Engine đấy ở đâu. (tức là thư mục quy định nơi mà chúng ta lưu file template). For example, app.set(‘views’, ‘./views’), với views là thư mục mà tệp template được đặt tại đó. Điều này mặc định là thư mục views trong thư mục gốc của ứng dụng.
* Bước 2: Chỉ cho thằng Express biết rằng chúng ta dùng template engine gì và đặt cho nó 1 cái tên. Ví dụ, app.set('view engine', 'ejs'), công cụ mẫu để sử dụng.

+ File .env ra đời do chúng ta phải tạo rất nhiều biến ở nhiều file thì sau mỗi lần giao một version của dev cho tester test (hay còn gọi là user access test – uat) và sau này còn có bản production(.prod) giao cho khách hàng (customers) sau đó update thì chúng ta cần phải thay đổi value rất nhiều -> như vậy rất tốn thời gian -> file ENV ra đời giúp chúng ta dễ config variable dùng chung cho nhiều file hơn.

+ Trước khi sử dụng file .env cần phải khai báo **require('dotenv').config();**. Ví dụ: gán port bằng file .env require('dotenv').config();

+ Chỉ có thể đẩy .env.<tên\_gói> lên GitHub, còn .env giữ ở máy local vì có chứa thông tin nhạy cảm. Sao chép các thông tin cần thiết của project trong file .env vào file .env.<tên\_gói> và file .env chỉ giữ những thông tin cần thiết.

+ Thông thường ở công ty không ai đẩy file .env lên GitHub mà chỉ đẩy bản sao của nó chứa các thông tin cần thiết của dự án (vd: .env.example) lên GitHub.

+ nodemon giúp chúng ta cập nhật các thay đổi trực tiếp mà không cần chạy lại project.

+ Trong package.json:

* “dependencies”: Các package cần thiết để chạy production.
* “devDependencies”: Các package chạy local dùng cho developer, nếu chạy production thì vẫn 'ok', tuy nhiên có rủi ro về hiệu năng.