NodeJS

1. Cài đặt
2. Tạo package.json

npm init

1. Tạo project nodejs

npm i -g express-generator

express app\_name



* Đưa tất cả file vào 1 folder src(tự tạo), trừ folder public

1. Setting

npm install --save npm-run-all

1. Install babel (công cụ chuyển đổi)

npm install -D @babel/core @babel/cli @babel/preset-env @babel/node

yarn add [babel-core@7.0.0-bridge.0](mailto:babel-core@7.0.0-bridge.0)

1. Tạo môi trường product

Convert code ở src sau khi build, name là: product

+ npm install rimraf -save

Add script sau vào file package.json

+ "clean": "rimraf product"

Build script

+ "build": "babel ./src --out-dir product --copy-files"

Product script

+ "server:prod": "node ./product/bin/www",

+ "prod": "SET NODE\_ENV=production & npm-run-all clean build server:prod"

1. Auto restart when change

+ Install nodemon: npm i -D nodemon

|  |  |
| --- | --- |
| Config file nodemon.js: |  |

Add script sau vào file package.json

+ "watch": "nodemon"

Chạy lệnh sau để kiểm tra

+ npm run watch

1. Cài đặt migrations

[Tài liệu tham khảo](https://itnext.io/updating-an-sql-database-schema-using-node-js-6c58173a455a)

<https://db-migrate.readthedocs.io/en/latest/API/SQL/>

B1 npm install -g db-migrate

B2 npm install --save db-migrate-mysql

B3 tạo file database.json

{

"dev": {

"driver": "mysql",

"host": "localhost",

"port": "3306",

"user": "root",

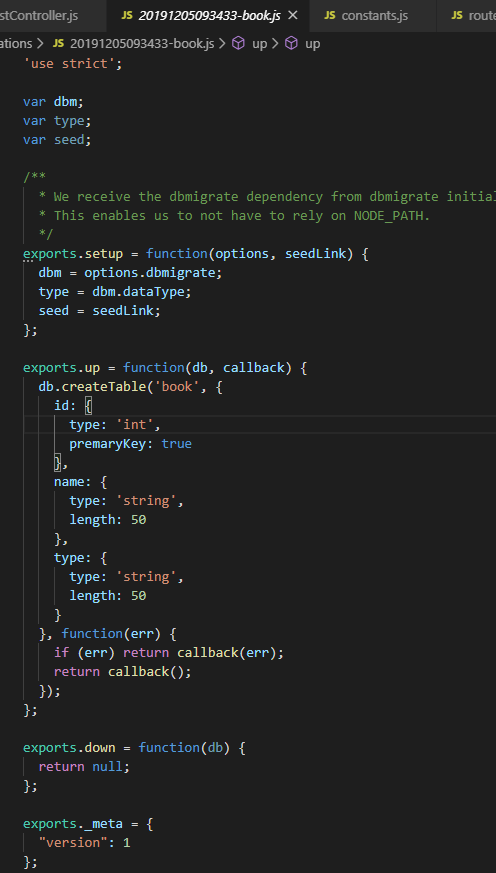
"password": "",

"database": "nodejs\_mysql"

}

}

B4 db-migrate create name\_table



B5 db-migrate up // run file

1. Tạo view template

Install template: npm i express-handlebars

|  |  |
| --- | --- |
|  | Import thư viện vào app.js |
|  | Thiết lập extensiton, default layout, element, helpers cho template. |
|  | Tạo cấu trúc thư mục như bên |
|  | Render body |
|  | Tạo file helper theo đường dẫn đã mô tả ở trên (Vd: block.js) |
|  | Tạo partial view (element) cho nodejs |