Kiểu viết scss khác css:

Scss là kiểu viết nested

Cách viết scss

Viết scss gọn hơn, tối ưu hơn css

**npm install body-parser@1.19.0 dotenv@8.2.0 ejs@3.1.5** [**express@4.17.1**](mailto:express@4.17.1)

Body-parser: lấy các tham số mà client gửi lên server

Dotenv: dùng câu lệnh process.env. .Lấy tham số khi báo trong môi trường

Ejs: view engine

Express: giúp chạy project

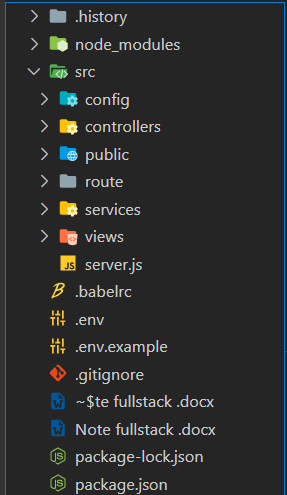
**npm install --save-dev @babel/core@7.12.10 @babel/preset-env@7.12.10 @babel/node@7.12.10** [**nodemon@2.0.7**](mailto:nodemon@2.0.7)

babel là compile để dịch bất kể phiên bản nào của javascript(trong 1 dự án)

nodemon để có sự thay đổi file thì server tự restart

Ejs giống như file jsp(java) và blade(PHP)

# Mô hình MVC



## Controllers

let getHomePage = (req, res) => {

    // vì ở view engine đã cài đặt là nằm ở src/views nên không cần đường dẫn

    // render ra view

    return res.render('homepage.ejs');

}

let getAboutPage = (req, res) => {

    return res.render('test/about.ejs')

}

module.exports = {

    getHomePage: getHomePage,

    getAboutPage:getAboutPage

}

## Route

import express from 'express';

import homeController from '../controllers/homeController';

let router = express.Router();

let initWebRoutes = (app) => {

    router.get('/', homeController.getHomePage);

    router.get('/about', homeController.getAboutPage);

    return app.use("/", router);

}

module.exports = initWebRoutes;

## Views

<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>

</head>

<body>

    <h1>Huy dz</h1>

</body>

</html>

## Config

import express from 'express';

let configViewEngine = (app) => {

    // static để server biết chỉ được lấy data từ đâu

    app.use(express.static("./src/public"));

    app.set("view engine", "ejs");

    //đường link lấy view engine

    app.set("views","./src/views")

}

// export ra

module.exports = configViewEngine;

## Server

import express from "express";

// lấy tham số từ client trả về server. VD/user?id=7

import bodyParser from "body-parser";

import viewEngine from "./config/viewEngine";

import initWebRoutes from "./route/web";

// gọi tới config của thư viện dotenv để có thể process.env.

require('dotenv').config();

// tạo 1 instance của express

let app = express();

// config app

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({extended: true}))

viewEngine(app);

initWebRoutes(app);

// lấy ra từ file env

let port = process.env.PORT || 6969;

app.listen(port, () => {

    console.log("Backend Nodejs is running on port " + port)

})

# Cài đặt sequelize CLI

Command line interface

* Cung cấp câu lệch để thao tác với cơ sở dữ liệu

Npm install –save-dev [sequelize-cli@6.2.0](mailto:sequelize-cli@6.2.0)

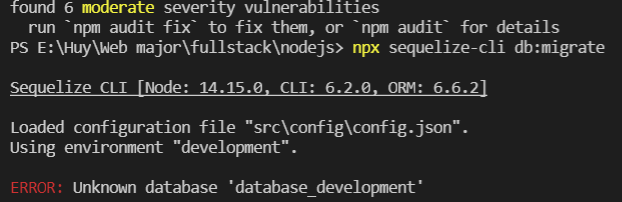
Link tham khảo: https://dev.to/ceceliacreates/sequelize-basics-for-beginners-part-one-2lc6

Fix lỗi khi cài sequelize

<https://dev.to/tradecoder/how-to-fix-error-nodemon-ps1-cannot-be-loaded-because-running-scripts-is-disabled-on-this-system-34fe>

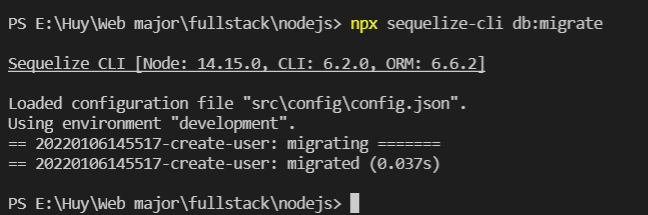
npx sequelize-cli model:generate --name User --attributes firstName:string,lastName:string,email:string

fix lỗi



Sequelize sẽ chạy vào file “.env” để lấy 1 biến NODE\_ENV để cho phép chạy vào development

Kết quả



# Sequelize

Sequelize là một package hỗ trợ sử dụng ORM với môi trường Node.JS . Trong video này, chúng ta sẽ cùng nhau tìm hiểu cách kết nối tới database với Sequelize, đồng thời học cách lấy dữ liệu từ database để render ra view.

https://sequelize.org/master/manual/getting-started.html

## Tạo seeder

npx sequelize-cli seed:generate --name demo-user

## Chỉnh sửa seeder

return queryInterface.bulkInsert('Users', [{

firstName: 'John',

lastName: 'Doe',

email: 'example@example.com',

createdAt: new Date(),

updatedAt: new Date()

}]);

## Running seeder

npx sequelize-cli db:seed:all

## Tạo file connectDB.js ở folder config

const { Sequelize } = require('sequelize');

// Option 3: Passing parameters separately (other dialects)

const sequelize = new Sequelize('booking\_healthcare', 'root', null , {

  host: 'localhost',

  dialect: 'mysql'

});

let connectDB = async() => {

    try {

        await sequelize.authenticate();

        console.log('Connection has been established successfully.');

      } catch (error) {

        console.error('Unable to connect to the database:', error);

      }

}

module.exports = connectDB;

* **Biến sequelize để connect vs database**
* **Biến connect DB để check connect**

## Trả biến **connect DB** tới server.js

### 

10:17