Section 2: Tools Installation

1.Cài đặt:

- Cài đặt Angular

- Cài đặt Mongodb và Robo3T

- Cài đặt Visual Studio Code

Section 3: Create Angular Project (clip 10 giới thiệu tổng quát bài)

2.Tạo Project Angular:

- Tạo Angular Project: ng new chatapp

- Nạp Prettier cho project: npm i prettier -D

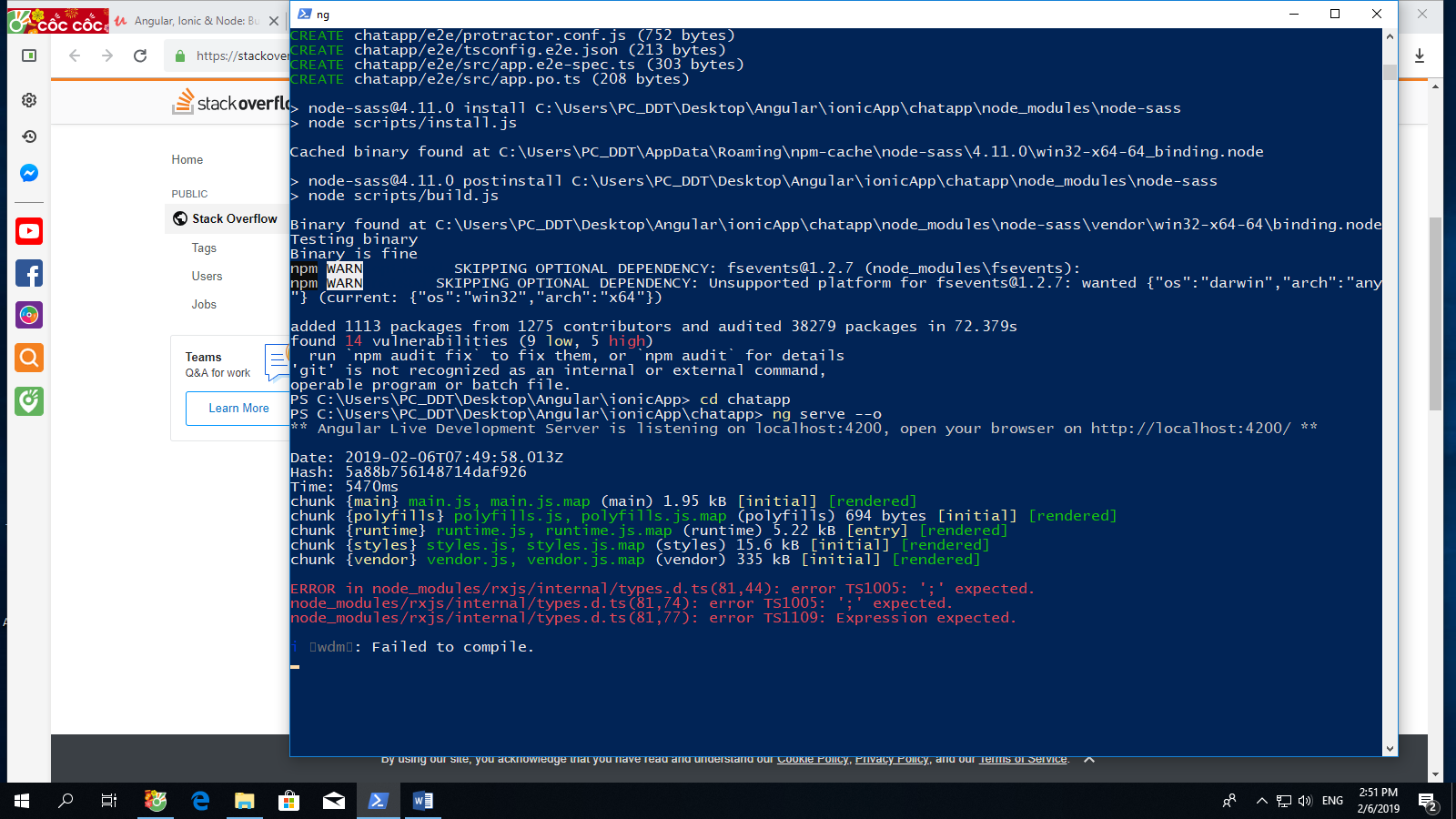
Prettier là gì? làm được gì? tại sao cần dùng nó?: Đây là tool để format code 1 cách tự động. Bạn sẽ không cần phải tốn thời gian với việc đi format từng đoạn code nhìn sao cho đẹp, rồi trong team 9 người 10 ý, mỗi người 1 kiểu format nhìn rất là xấu code. Vậy thì Prettier sẽ giúp thống nhất phong cách code chung của mỗi người. Sẽ không còn phải mỗi lần review pull đi soi format code nữa.

- Nạp Materialize dùng 1 trong 2 phương pháp:

PP1: CDN CSS & JS

PP2: install npm Materialize.css có khai báo sử dụng css và js trong angular.json

Lỗi: khi vừa cài project angular chạy đã lỗi



và khắc phục:

Changes in your package.json file.

Step 1 : Go to package.json and modify "rxjs": "^6.0.0" to "rxjs": "6.0.0"

Step 2 : Run npm update in your project.

Section 4: Authentication Components (clip 19 giới thiệu tổng quát bài)

- Tạo components/authTabs với --no—spec: ko tạo file.spec --dry-run:show thử file thay đổi

- Tạo modules/auth với –no--spec và --flat

- Tạo modules/app-routing với --no—spec và --flat

- Thiết kế giao diện Login – Signup

- Tạo component Login và Signup:

Ng g c components/login --no-spec --module-modules/auth.module.ts

Ng g c components/singup --no-spec --module-modules/auth.module.ts

- Thiết kế giao diện cho Login - Signup gọi app-login và app-signup trong auth-tabs

Tạo cấu trúc thư mục: components – modules – service:

App.module gọi toàn bộ module trong thư mục modules.

Các component Login và Signup sẽ được thêm vào module auth.modules

Section 5: Login and SiguUp Authentication (clip 31 giới thiệu tổng quát bài)

- Chuyển folder chatapp (angular) thành chatapp-frontend

- Tạo thư mục chatapp-backend

- Dùng VSC: file/Save Workspace As.. lưu chatapp-frontend thành Chatapp

chứa thư mục chatapp-frontend

chứa thêm thư mục chatapp-backend (Add folder Workspace)

- Install Express: dùng cli tại chatapp-backend

Npm init và khai báo lại package.json để tạo file server.js

Npm install express mongoose --save

- Tạo file server.js

Const express = require(‘express’);

Const mongoose = require(‘mongoose’);

Const app = express();

Const port = process.env.PORT || 3000;

Mongoose.Promise = global.Promise;

Mongoose.connect(‘mongodb://localhost:270172/socialapp’,{ userNewUrlParser:true});

App.listen(port, () => {

Console.log(‘Running on port 3000’);

}

- Cài đặt nodemon: npm i –g nodemon

- Tạo cấu trúc thư mục backend: models-routers-controllers-helpers-config

- Tạo thư mục config chưa file secret.js để chưa thông số url mongodb và đưa server.js

- Cài đặt cookie-parser & morgan hỗ trợ nhận giao thức

Sử dụng middleware cookie-parser và morgan

- Tạo userModel, authRoute, authController

Sử dụng middleware bodyParser và phối hợp router và controller trong gọi hàm api

app.use(express.json()) và app.user(express.urlendcoded())

để post dùng req.body và get dùng req.params

- Cài đặt Joi và khai báo sử dụng Joi trong authController

Tạo biến schema Joi kiểu models nhận được bằng post kiểm tra tính hợp lệ

- Cài đặt http-status-codes: hỗ trợ các thông báo lỗi http không bằng số mà bằng chữ

error.status(500) trở thành error.status(Httpstatus.BAD\_REQUEST)

- Khai báo helpers.js hỗ trợ lower email và firstUpper username

Kiểm tra trong userModel có email hay username trùng dữ liệu đã có không?

- Cài đặt bcryptjs: hỗ trợ mã hóa password

- Khai báo sử dụng bcryptjs và mã hóa password, tạo biến body là dùng User.create(body)

.then để lưu

.catch để báo lỗi

- Cài đặt Jsonwebtoken, tạo khóa secret và tạo token đưa vào cookie: res.cookie(‘auth’,token)

- Tạo Angular: ng g service services/auth --no-spec --flat

Để tạo service và khai báo trong authModule có dùng thư viện HttpClient để gọi phương thức post từ server

- Khai báo sử dung FormModule và ReactiveFormsModule trong authModule

Khai báo sử dụng FormGroup, FormBuilder, Validators trong signup.component.ts

Với khai báo signupForm: FormGroup lấy giá trị trong component.html

Với khai báo fb: FormBuilder để lấy giá trị mặc định và kiểm tra Validators

Khai báo signupForm= this.fb.group({Validators: kiểm tra giá trị trước khi nhận})

- Cài đặt CORS: giải pháp proxy cho fontend và backend

Khai báo và sử dụng cors

- Khai báo báo lỗi gồm 2 phần:

Phần 1: [disable]=”!sigupForm.valid” tại button

Phần 2: <div \*ngIf=”errorMessage”> với error lấy giá trị trả về từ backend

- Tạo streams component, module, routing

- Tạo Login service và xây dựng Form Login Component tương tự Signup Component

Section 6: POST (clip 61 giới thiệu tổng quát)

- Thay đổi thời gian Token tại signup và login của backend userController

- Cài đặt cookie cho angular: ng g service services/token --no-spec --flat khai báo stream module

ngx-cookie-service khai báo app module và dùng trong token service

- Khai báo các hàm dùng token trong token service: Set/Get/DeleteToken

- Đưa token từ backend đến frontend tại signup/login component.ts

Sau khi login lấy token thì lấy token cookies: streams lấy token từ GetToken service

- Cài đặt guard: ng g guard service/auth --no-spec --flat

Khai báo kiểm tra token

Nếu token đúng thì return true

Không đi đưa về trang ngoài return flase

- Set chế độ mặc định khi vào trang app.component thì kiểm tra token có thì vào trang stream không thì vào trang login

- Cài đặt trang toolbar component khai báo trong stream module có chức năng logout

… (thiếu phần khai báo nav content) ---

- Cài đặt trang side component, postForm component, posts component

Khai báo định dạng side component và định dạng postForm

- Backend: khai báo post model

Khai báo router cho post

- Frontend: tạo service cho post (tương tự như login và signup)

- Backend: tạo helper authHelper giúp kiểm tra jwt đúng và còn hạn thì vào postController (1 kiểu middleware)

- Tạo class services/token-interceptor để khai báo lại header của http-request bỏ authorization token vào

- Khai báo toke-intercepter trong streams module

- Kiểm tra từ Web đưa dữ liệu vào .ts đến service vào authHelpers lấy token kiểm tra nếu đúng thì đưa dữ liệu vào postController và dùng joi để kiểm tra kiểu post có dữ liệu

- Khai báo thêm posts trong userModel và tạo thêm phần nạp post vào userModel

- Token payload: vào token service lấy phần payload để lấy thông tin người dùng

- Tạo giao diện Posts: collection avatar

- Backend: tạo getAllPost lấy toàn bộ posts với tên người dùng đưa ra trang postsComponent thông qua service

- Frontend: đưa AllPost từ backend ra hiển thị trên trang web posts component

- Cài đặt Moment: đổi giờ hệ thống, xây dựng hàm TimeFormNow để tính thời gian post tin

- Cài đặt Socket.io cho backend: socket.io và frontend: socket.io-client

Khai báo socket.io-client thêm cho polyfills.ts: (window as any).global = window;

Chuyển data từ frontend postForm đến server và từ server chuyển về frontend posts và thực hiện làm mới lại phần post khi có người mới đăng tin

- Khai báo phần addLike khi người dùng click vào nút like và thêm điều kiện chỉ cho click 1 lần

Frontend: Tạo hàm click addLike, Backend: tạo router và hàm thêm và kiểm tra click, Frontend: tạo service gọi server.

- Cài đặt Frontend: lodash hỗ trợ biến có dấu \_

- Khai báo hàm kiểm tra trong post nếu có người dùng đã nhấn like thì đổi màu icon

- Cài đặt Frontend: comments Component, tạo giao diện giống post và post form (sử dụng AfterViewInit để cắt không thể hiện phần .nav-content)

- Hoàn tất phần báo đã chọn like và comment : [ngClass] và lodash \_.some

- Hiển thị toàn bộ comment theo postId

- Logout bằng cách xóa token và đưa về trang đầu khi token === null

Section 7: Users Section (Clip 124 giới thiệu tổng quát)

- Tạo people component trong stream module

- Khai báo định dạng people theo dạng danh sách tất cả user với menu giống streams

- Backend: tạo UserRouter, UserController – Frontend: tạo service User vào duyệt toàn bộ user

- Không hiển thị người dùng trong danh sách people: dùng lodash \_.remove và tokenServicePayload

- Khai báo lại userModel với 2 trường followed và follower ref userModel

- Tạo Router và Controller friends

Section 8: Private Chat Section (Clip 172 giới thiệu tổng quát)

- cài đặt ngx-auto-scroll hỗ trợ chat

- Cài đặt ng2-emoji-picker biểu tượng chat

Ng2-emoji-picker lỗi do dùng rxjs5 và angular 6 dùng rxjs 6 nên phải dùng rxjs compat:

Npm install –save rxjs-compat

npm install @ctrl/ngx-emoji-mart hình đẹp

- Cài đặt lodash trong backend

Section 9: Images Section (Clip 235 giới thiệu tổng quát)

- Tạo component ImagesComponent trong module Stream

- Cài đặt thư viện ng2-file-upload vào frontend

(thường có 3 thư viện: ng2-file-upload/multer/)