Apuntes para mi Backend

Para comenzar a crear mi backend, debemos crear mi archivo package.json de la shte. manera:

**npm init –y**

Me creara mi archivo de entrada de mi aplicación:

{

*"name"*: "03-backend-server",

*"version"*: "1.0.0",

*"main"*: "index.js",

*"scripts"*: {

*"test"*: "echo \"Error: no test specified\" && exit 1"

  },

*"keywords"*: [],

*"author"*: "",

*"license"*: "ISC",

*"description"*: ""

}

A continuación, instale Express en el directorio myapp y guárdelo en la lista de dependencias. Por ejemplo:

npm install express --save

Con esto nos creara nuestro archivo package-lock.json

Luego crearemos nuestro archivo index.js que sera el punto de partida de mi aplicación.

Para ejecutar nuestro archivo escribimos en consal de VSC:

node index.js

Ahora para no estar ejecutando o bajando el servidor por cada modificación debemos instalar NODEMON (en estos momentos version 2.0.7)

npm install -g nodemon

[nodemon] 2.0.7

[nodemon] to restart at any time, enter `rs`

[nodemon] watching path(s): \*.\*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting `node index index.js`

Servidor corriendo en puerto 3000

Para ahorrar tiempo modificaremos nuestro scripts:

*"scripts"*: {

*"start:dev"*: "nodemon index.js"

  },

Y en consola ejecutaremos

npm run start:dev

Tambien al costado izquierdo de VSC podemos ejecutar este comando (SCRPTS NPM)

Para ingresar a Mongo compass: correo: [azp04@hotmail.com](mailto:azp04@hotmail.com) password: lorenzo68\_azp

Mongo Compas

nombre del cluster: MEAN Hospital

nombre proyecto: 03-backend-server

usuario: mean\_user

password: GQQJ4SLnyWLmCCRx

Estableciendo conexion entre Mongo Atlas Y node.

vamos a la pagina de monggose: <https://mongoosejs.com/>

Luego en la terminal instalaremos mongoose: npm i mongoose

Agregaremos variables de entorno para guardar nuestraos claves token y demas

npm i dotenv

Ademas debemos agregar la libreria de CORS

sirve para: CORS es un paquete node.js para proporcionar un middleware [**Connect**](http://www.senchalabs.org/connect/) / [**Express**](http://expressjs.com/) que se puede usar para habilitar [**CORS**](http://en.wikipedia.org/wiki/Cross-origin_resource_sharing) con varias opciones.

Es decir acepta peticiones de diferentes dominios.

npm i cors

Ahora realizaremos nuestro respaldo con GITHUB, pero crearemos un nuevo repositorio para tenerlo separado del FRONTEND. Entonces en github creamos un nuevo repos. y se llamara:

03-Hospital-backend

Cabe señalar que creamos un archivo llamado .gitignore el cual mencionaremos la carpeta NODE\_MODULES para que nuestro repositorio no la tome en cuenta

NODE\_MODULES/

git init

Initialized empty Git repository in D:/ANGULAR\_AVANZADO/03-backend-server/.git/

git add .

git commit -m "CORS y express primer commit"

colocaremos las versiones

git tag –a v0.1.0 –m “Inicio de Backend”

**Admin-Pro BACKEND**

**Aplicación de Hospital**

Crearemos tres colecciones (o tablas) en nuestra BBDD

HOSPITALES

MEDICOS

USUARIOS

Los USUARIOS se van a relacionar con los medicos.

porque cada MEDICO tiene que tener el registro de que USUARIO lo creo, a lo igual que los HOSPITALES.

Los MEDICOS se va relacionar con los HOSPITALES porque cada MEDICO va a existir en un HOSPITAL.

Y los HOSPITALES se va relacionar con los MEDICOS porque los HOSPITALES pueden tener muchos MEDICOS.

Ahora pondremos el esquema o tabla de **USUARIOS**

nombre: string - obligatorio

apellido: string - obligatorio

email: string - obligatorio y unico (servira para autentificarse en la aplicación)

password: string – obligatorio – encriptado

img: string – opcional

role: string – obligatorio con valor por defecto y debe existir en una linea

google: booelando – opcional

Ahora pondremos el esquema o tabla de **MEDICOS**

nombre: string – obligatorio

apellido: string – obligatorio

img: string – opcional

usuario: referencia al usuario (que lo creo) Obligatorio

hospital: referencia a los hospitales – Obligatorio

Ahora pondremos el esquema o tabla **HOSPITALES**

nombre: string – Obligatorio

img: string – Opcional

usuario: referencia al usuario – Obligatorio

**LOGIN**

Admin – PRo

Username

Password

\*recordarme \* Olvidaste Pass.

EMAIL

G

LOGIN IN

PASSWORD

GUARDAR USUARIO

Generacion de JWT

Api Google Frontend

Ahora crearemos ciertos servicios:

Un **CRUD** para --> **CREATE**, **READ**, **UPDATE** y **DELETE**

* Se tendria que hacer un CRUD por cada tabla
* Un buscador general independiente
* Carga de imagenes y archivos
* Paginasciones
* Bloqueo de imagenes
* Generador de Tokens
* Reconvertir token
* Validar token de Google - Backend