Nodejs y Mongodb Aplicación completa (Login, Registro,CRUD, ES6+, y más)

Aprende a crear una aplicación web Backend y Frontend, utilizando Nodejs, Mongodb, Express y otras tecnologías de Javascript. En este ejemplo crearemos una aplicacion de Notas, que sera un CRUD de nodejs, más un Login y Registro utilizando Mongodb, ademas de proteger nuestras rutas del servidor y estilizar nuestra aplicación con CSS y Bootstrap4.

<https://www.youtube.com/watch?v=-bI0diefasA>

INDICE DEL TUTORIAL

1. Backend: [04:47](https://www.youtube.com/watch?v=-bI0diefasA&t=287s)

2. Views (Template Engine): [41:24](https://www.youtube.com/watch?v=-bI0diefasA&t=2484s)

3. CRUD: [59:13](https://www.youtube.com/watch?v=-bI0diefasA&t=3553s)

4. Register/SingUp: [02:03:30](https://www.youtube.com/watch?v=-bI0diefasA&t=7410s)

5. Login / SignIn: [02:30:27](https://www.youtube.com/watch?v=-bI0diefasA&t=9027s)

6. Navigation: [02:51:54](https://www.youtube.com/watch?v=-bI0diefasA&t=10314s)

7. Personal Data for Users: [03:12:40](https://www.youtube.com/watch?v=-bI0diefasA&t=11560s)

Creamos la carpeta notes-app

Creamos el package.json

npm init --yes

instalamos todas las dependencias:

npm i express express-handlebars express-session method-override mongoose passport passport-local bcryptjs connect-flash

express: framework de node, escribir el server

express-handlebars: integrar un motor de plantillas en express, para extender el html , para que tenga por ejemplo bucles y condicionales

express-session: crear sesiones dentro del server, para auntenticar el usuario y que se recuerde esa autenticacion, almacenamos sus datos temporalmente en una sesion

method-override: es para extender la funcionalidad de los formularios, ya que solo permiten *post* y *get*, con esta dependencia se podra utilizar *put* y *delete*

mongoose: modulo que permite unir a express con una BD, no es la BD, mongoose permite conectarme a mongodb (una especie de driver)

passport y passport-local: para autenticar a mi usuario

bcryptjs: para cifrar la contraseña, convierte un string en un hash, en nuestro caso la contraseña

connect-flash: para enviar mensajes entre multiples vistas, mensajes al usuario

creamos carpeta:

\src

index.js

database.js

creamos las subcarpetas dentro de src

cd src

mkdir config helpers models public routes views

mkdir

config: variables de configuracion del entorno

helpers: funciones que luego podemos reutilizar

models: como lucen los datos en la bd

public: archivos estaticos

routes: crear las url o las rutas de nuestro server, ruta login, register

views: almacenar las vistas html que vamos a enviar al navegador

comenzamos c a trabajar con el src/index.js

instalamos nodemon:

npm i nodemon -D

reinicia automaticamente el server, esta escuchando los cambios en caliente

iniciar el servicio de mongo:

windows: mongod

linux: sudo service mongod start