1 crear las carpetas

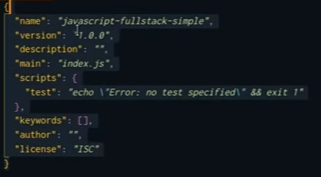


2 Crear un archivo package.jsoncon la descripción del proyecto

npm init -y







3 instalar



npm i express mongoose morgan multer dotenv cross-env cors fs-extra

Express: framework nodejs servidor

Mongoose: modulo de conexión bdd

Morgan: modulo ir viendo por consola en el servidor

Multer: subir imágenes al servidor

Dotenv: variables de entorno, api

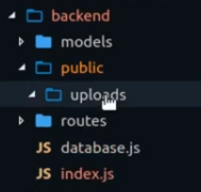
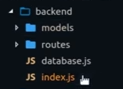
Cross-env: definir en que entorno estamos

Cors: permite enviar y recibir daros de los dos servidores

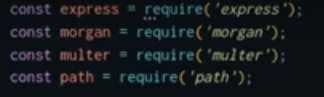


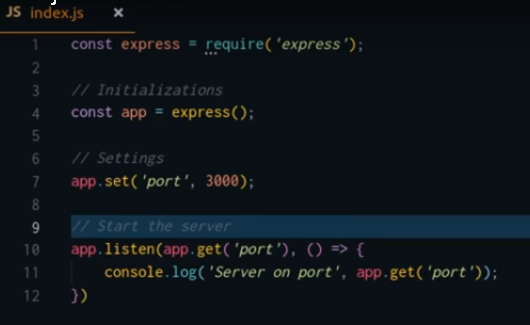
Extra: copiar archivos

4 crear la estructura del proyecto



configurar el server con express





Probar en consola (el servidor)



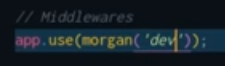
Morgan (muestra por consola detalles de ejecución del servidor)



Ejecutar Morgan



Esto es el valido para corregir error



***instalar la dependencia para desarrollo***

**- nodemon** (permite ejecutar código si se realiza un cambio y reinica el servidor automaticamente)

npm i nodemon -D



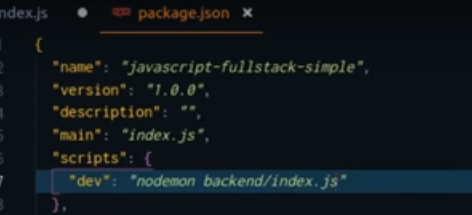
Ejecutar nodemon (automatiza el servidor, ejm reinicio y otros)



Si presenta error



Mejor opción para ejecutar nodemon



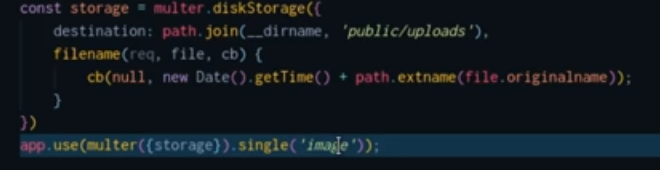
"dev":"nodemon backend/index.js"

Ejecutar la configuración anterior



npm run dev

* **Configurar multer (para subir archivos)**



//multer(subir imagenes)

const storage=multer.diskStorage({

    destination: path.join(\_\_dirname,'public/upload'),

    filename(req,file,cb){

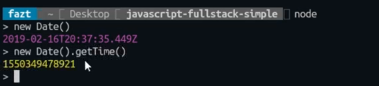
        cb(null,new Date().getTime()+path.extname(file.originalname))

    }

})

app.use(multer({storage}).single('image'));

Ejecutar comandos de node (funciones para capturar la fecha y su código para dar como nombre del archivo)



* **Para entender los datos que se envían del frontendde los formularios y json que recibe el backend**



**--routes**

