

Crear un servidor express desde cero con nodemon

Ve al directorio /home/Project

cd /home/Project

Crea un directorio llamado myexpressapp y cambia a ese directorio

mkdir myexpressapp

cd myexpressapp

Ahora ejecuta **npm init**

Esto inicializará el directorio de la API para servir como una aplicación web. Sigue las indicaciones en la pantalla para completar la inicialización.

- El nombre del paquete por defecto es el nombre de la carpeta actual (myexpressapp en este caso). Puedes especificar un nombre diferente si lo deseas.
- A continuación, te preguntará por la versión que deseas establecer. El valor por defecto es 1.0.0.
- Luego solicita una descripción donde puedes dar una breve descripción de lo que la API pretende hacer. Puedes dejarlo en blanco.
- A continuación, especificamos el punto de entrada de la API, que por defecto es index.js.
- Cuando te pregunte por el autor, puedes dar tu nombre o dejarlo en blanco.
- La licencia por defecto es ISC (Internet Systems Consortium), lo que significa que es una licencia permisiva que permite a las personas hacer cualquier cosa con tu código con la atribución adecuada y sin garantía.
- Generará el contenido para tu package.json, un archivo que mantiene un registro de todos los paquetes que necesita tu aplicación de servidor, y te pedirá que verifiques si los detalles son correctos. Una vez que confirmes, los detalles se escribirán en el package.json.
- Ahora ejecuta el siguiente comando para instalar express.

npm install express --save

La opción **--save** asegura que el package.json se actualice.

5. Ahora ejecuta el comando **touch.exe index.js** Verás que este archivo se crea en el explorador de archivos. Puedes usar el IDE para escribir el código que desees dentro de lo que has aprendido en los ejercicios anteriores.

```

1  // Import the Express.js library
2  const express = require('express');
3
4  // Create an instance of an Express application
5  const app = new express();
6
7  // Define the port number
8  const port = 8080;
9
10 // Route to handle requests to the root path "/"
11 app.get("/", (req, res) => {
12     return res.send("Hello World!");
13 });
14
15 // Start the server and listen on the specified port
16 let server = app.listen(port, () => {
17     console.log("Listening at http://localhost:" + port);
18 });

```

- Realiza cambios en package.json para iniciar el servidor con npm start. Incluye "start" bajo scripts.

```

1  {
2    "name": "myexpressapp",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "start": "node index.js",
8      "test": "echo \"Error: no test specified\" && exit 1"
9    },
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "express": "^X.X.X"
14   }
15 }

```

- Desde el símbolo del sistema, ahora puedes ejecutar **npm start** para iniciar el servidor.
- Ahora puedes incluir otros puntos finales o hacer cambios en el servidor según sea necesario. Pero puede ser muy frustrante detener y reiniciar el servidor cada vez que haces cambios. Hay un paquete que resulta útil en este caso. El paquete se llama nodemon. Cada vez que haces cambios en la API del servidor, reiniciará automáticamente el servidor. Instalemos eso en el mismo directorio donde creamos nuestro index.js. Lo instalaremos y lo almacenaremos como una dependencia de desarrollo con la opción **–save-dev** porque queremos usar esto solo cuando estemos ejecutando el servidor localmente en nuestro entorno de desarrollo.

9. npm install --save-dev nodemon

10. Una vez que nodemon esté instalado, haremos cambios en package.json para utilizarlo y reiniciar el script cuando haya cambios. Incluirá el `"start" : "nodemon index.js"` en la sección de scripts de nuestro package.json. Con los cambios, el package.json se verá así.

```
1  {
2    "name": "myexpressapp",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "start" : "nodemon index.js",
8      "test": "echo \"Error: no test specified\" && exit 1"
9    },
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "express": "^X.X.X"
14   },
15   "devDependencies": {
16     "nodemon": "^X.X.X"
17   }
18 }
```

En el símbolo del sistema, ahora ejecuta `npm start` para iniciar el servidor web.

Ahora haz algún cambio o agrega otro endpoint y observa si el servidor se reinicia y los cambios se reflejan sin tener que reiniciar explícitamente.