



Servicio nacional de Aprendizaje  
Centro de Atención Sector Agropecuario  
Regional Risaralda  
ADSO

## Construcción de una Api Rest en Node JS con express



1. Creamos una carpeta e iniciamos en VSC.
2. Creamos un proyecto con el comando **npm init -y** Esto nos generará un archivo **package.json**
3. Instalamos Express mediante **npm install express** Esto generará una carpeta **node\_modules** y un archivo **package-lock.json**
4. Creamos un archivo con nombre **index.js**
5. En el archivo **package.json** debajo de “main” agregamos "type":"module", para que nos deje usar la estructura import al requerir paquetes.
6. En el archivo **index.js** escribimos el siguiente código para levantar el servidor y comprobar su funcionamiento.

```
import express from 'express'
const app = express();

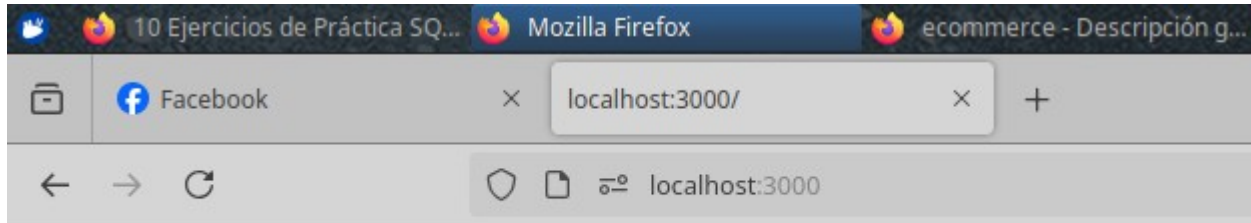
app.get("/",(req, res) =>{
  res.send("Bienvenidos a mi Api Rest en Nodejs")
})

app.listen(3000,() =>{
  console.log("Servidor corriendo en el puerto 3000")
})
```

Si todo va bien, en un navegador escribimos <http://localhost:3000/> Esto debe arrojar en el DOM ***Bienvenidos a mi Api Rest en Nodejs***



Servicio nacional de Aprendizaje  
Centro de Atención Sector Agropecuario  
Regional Risaralda  
ADSO



Bienvenidos a mi Api Rest en Nodejs

7. Para no detener e iniciar el servidor cada vez que se hace un cambio en el código, instalamos nodemon mediante **npm install nodemon -D** La D significa que instale como una dependencia de desarrollo.

Para que nodemon funcione correctamente debemos ir al archivo package.json y agregar en la parte de scripts la siguiente línea “dev” : “nodemon index.js”

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "dev" : "nodemon index.js"  
},
```

Ahora iniciamos el servidor mediante **npm run dev**



Servicio nacional de Aprendizaje  
Centro de Atención Sector Agropecuario  
Regional Risaralda  
ADSO

8. Creamos una archivo **db.json** que contendrá los datos necesarios para las peticiones.

```
{
  "books": [
    {
      "id": 1,
      "name": "Los 7 hábitos de la gente altamente efectiva",
      "author": "Stephen R. Covey",
      "year": 1989
    },
    {
      "id": 2,
      "name": "Padre rico, padre pobre",
      "author": "Robert Kiyosaki",
      "year": 1997
    },
    {
      "id": 3,
      "name": "La asa del lago",
      "author": "Robert Kiyosaki",
      "year": 1999
    },
    {
      "id": 4,
      "name": "Lo que el viento se llevó",
      "author": "Robert Kiyosaki",
      "year": 1985
    },
    {
      "id": 5,
      "name": "Cien años de soledad",
      "author": "Robert Kiyosaki",
      "year": 1984
    },
    {
      "id": 6,
      "name": "COSMOS",
      "author": "Robert Kiyosaki",
      "year": 1923
    }
  ]
}
```

9. En nuestro archivo index.js escribimos **import fs from "fs";** para manipular archivos. También escribimos **import bodyParser from "body-parser"; app.use(bodyParser.json());** para trabajar en formato json los datos.

10. Creamos las funciones de lectura y escritura de los datos respectivamente.

```
const readData = () => {
  try {
    const data = fs.readFileSync("./db.json");
    return JSON.parse(data);
  } catch (error) {
    console.log(error);
  }
};
```



Servicio nacional de Aprendizaje  
Centro de Atención Sector Agropecuario  
Regional Risaralda  
ADSO

```
const writeData = (data) => {  
  try {  
    fs.writeFileSync("./db.json", JSON.stringify(data));  
  } catch (error) {  
    console.log(error);  
  }  
};
```

## REST API

Crear usuario	POST	/usuarios
Obtener un usuario	GET	/usuarios/{id}
Obtener usuarios	GET	/usuarios
Actualizar usuario	PUT	/usuarios/{id}
Eliminar usuario	DELETE	/usuarios/{id}

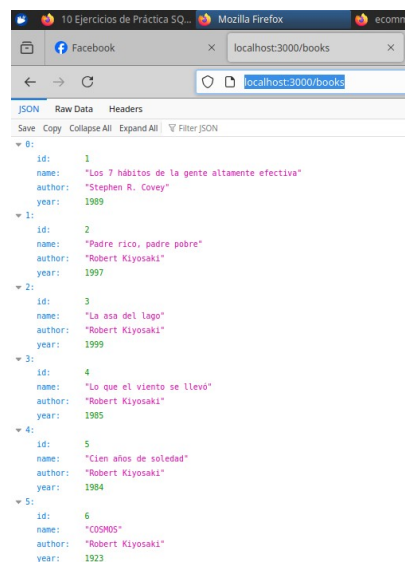


Servicio nacional de Aprendizaje  
Centro de Atención Sector Agropecuario  
Regional Risaralda  
ADSO

11. Creamos nuestra primera petición con el método GET. Este método nos permite leer los datos y mostrarlos en consola o el DOM del navegador.

```
app.get("/books", (req, res) => {  
  const data = readData();  
  res.json(data.books);  
});
```

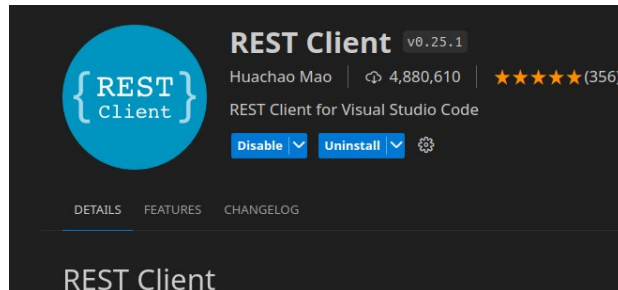
Si en la url del navegador escribimos localhost:3000/books debe aparecer los datos en el DOM del archivo db.json



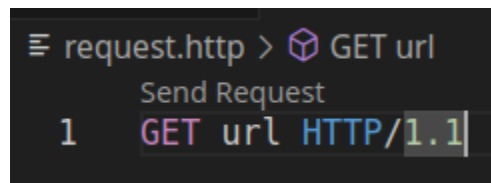
Otra forma de comprobar esta y las siguientes peticiones es mediante algunas herramientas tales como POSTMAN, ThunderClient o RestClient. En este caso utilizaremos la última señalada. Para ello, en VSC buscamos la extensión RestClient y la instalamos.



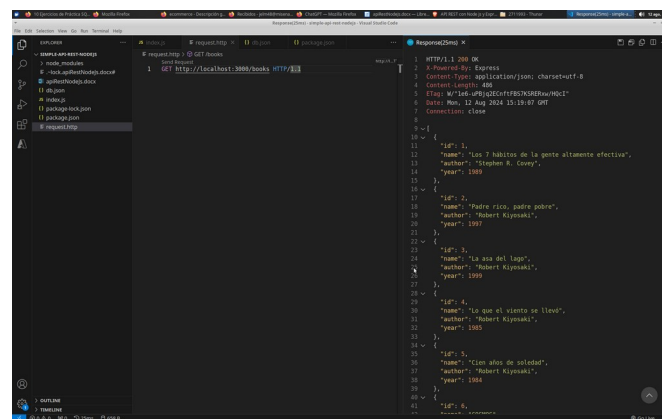
Servicio nacional de Aprendizaje  
Centro de Atención Sector Agropecuario  
Regional Risaralda  
ADSO



Una vez instalada la extensión creamos un archivo llamado **request.http**  
En dicho archivo escribimos GET y damos enter. Se mostrará algo similar a la imagen



En url escribimos <http://localhost:3000/books> y damos click en Send Request  
Debe aparecer en la parte derecha el resultado de la petición que en este caso son los datos.





Servicio nacional de Aprendizaje  
Centro de Atención Sector Agropecuario  
Regional Risaralda  
ADSO

12. Creamos otra petición tipo GET pero para traer un recurso específico y no todos.

```
app.get("/books/:id", (req, res) => {  
  const data = readData();  
  const id = parseInt(req.params.id);  
  const book = data.books.find((book) => book.id === id);  
  res.json(book);  
});
```

Para comprobar esta petición, en el archivo request.http creamos otra petición y pasamos como argumento el id del libro o recuso que queremos traer.

```
###  
Send Request  
GET http://localhost:3000/books/1 HTTP/1.1
```

En la imagen observamos que se pasa como parámetro el id 1 dando como resultado

```
Response(5ms) ×  
1 HTTP/1.1 200 OK  
2 X-Powered-By: Express  
3 Content-Type: application/json; charset=utf-8  
4 Content-Length: 103  
5 ETag: W/"67-tD+48trPA8jgVgW0SpSgjRCFB24"  
6 Date: Mon, 12 Aug 2024 15:30:57 GMT  
7 Connection: close  
8  
9 {  
10   "id": 1,  
11   "name": "Los 7 hábitos de la gente altamente efectiva",  
12   "author": "Stephen R. Covey",  
13   "year": 1989  
14 }
```



Servicio nacional de Aprendizaje  
Centro de Atención Sector Agropecuario  
Regional Risaralda  
ADSO

### 13. Creando una petición Post para crear un recurso

```
app.post("/books", (req, res) => {  
  const data = readData();  
  const body = req.body;  
  const newBook = {  
    id: data.books.length + 1,  
    ...body, //con el spreadOperator le decimos que lo que venga por el body lo agregue a la variable  
    newBook  
  };  
  data.books.push(newBook);  
  writeData(data);  
  res.json(newBook);  
});
```

```
###  
Send Request  
POST http://localhost:3000/books HTTP/1.1  
Content-Type : application/json  
  
{  
  "name": "Alicia en el país de las maravillas",  
  "author": "Ni j",  
  "year": 2002  
}
```

### 14. Creando una petición Put para actualizar un recurso

```
app.put("/books/:id", (req, res) => {  
  const data = readData();  
  const body = req.body;  
  const id = parseInt(req.params.id);  
  const bookIndex = data.books.findIndex((book) => book.id === id);  
  data.books[bookIndex] = {  
    ...data.books[bookIndex],  
    ...body,  
  };  
  writeData(data);  
  res.json({ message: "Book updated successfully" });  
});
```





Servicio nacional de Aprendizaje  
Centro de Atención Sector Agropecuario  
Regional Risaralda  
ADSO

```
###  
Send Request  
PUT http://localhost:3000/books/5 HTTP/1.1  
Content-Type : application/json  
  
{  
  "name": "Peque punto azul pálido",  
  "year": 1978  
}
```

15. Creando una petición DELETE para borrar un recurso.

```
app.delete("/books/:id", (req, res) => {  
  const data = readData();  
  const id = parseInt(req.params.id);  
  const bookIndex = data.books.findIndex((book) => book.id === id);  
  data.books.splice(bookIndex, 1);  
  writeData(data);  
  res.json({ message: "Book deleted successfully" });  
});
```

```
###  
Send Request  
DELETE http://localhost:3000/books/1 HTTP/1.1
```

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