Instituto Politécnico Nacional

Omar Montoya Romero

7CM1

WEB CLIENT AND BACKEND DEVELOPMENT FRAMEWORKS

Sistemas Computacionales

Ejercicio1.- Hola Mundo con Node.js y Express

Mtro. Efraín Arredondo Morales

13/09/2023

Primero codificamos un servidor en el puerto 3000 donde para poder configurarlo primero creamos uan carpeta donde va a ser el repositorio, una vez creada la carpeta pasamos a inicializar el repositorio con el comando node init -y.

Para el uso de este server usamos la dependencia de express, por lo tanto para instalarla usamos el comando npm install express.

Una ves instalado generamos el servidor HTTP y definimos en que puerto se va a escuchar nuestro servidor, en nuestro caso seria en el servidor 3000, tras declara el puerto definimos de que manera se va a hacer la consulta ya sea GET, POST, PUT o DELETE.

Hacemos la consulta a la app por el método definido en el primer caso GET en la dirección de raíz "/" donde se hace la solicitud y al mismo tiempo se manda una respuesta la cual es un Hola mundo despidiéndonos de Apache. De la misma manera se puede definir para los diferentes métodos anterior mente mencionados.

Para finalizar para el uso del programador y saber si el servidor esta corriendo y en que puerto se define un app.listen en el cual podemos imprimir un mensaje mas el puerto en el que se esta ejecutando.

```
const express = require("express"); // Crear servidores HTTP como Apache, IIS, etc.
const app = express(); //
const port = 3000; // Se escucha en el puerto 3000
app.get("/", (req, res) => {
    // Variables de solicitudes y respuesta
    //Mensaje para el cliente (Usuario)
    res.send("Adios vaquero (Apache ) por GET - Obtener"); // Enviamos Hola Mundo
});
app.post("/", (req, res) => {
    // Variables de solicitudes y respuesta
    //Mensaje para el cliente (Usuario)
    res.send("Adios vaquero (Apache ) por POST - Crear"); // Enviamos Hola Mundo
});
app.put("/", (req, res) => {
    // Variables de solicitudes y respuesta
    //Mensaje para el cliente (Usuario)
    res.send("Adios vaquero (Apache ) por PUT - Actualizar"); // Enviamos Hola Mundo
});
app.delete("/", (req, res) => {
    // Variables de solicitudes y respuesta
    //Mensaje para el cliente (Usuario)
    res.send("Adios vaquero (Apache ) por DELETE - Eliminar"); // Enviamos Hola Mundo
});
//Clientes: Edge, Insomnia, Chrome, FireFox
app.listen(port, () => {
    //Mensaje para el desarrollador
    //Diferentes maneras de mandar el mensaje
    //console.log("Servidor corriendo en el puerto: ", port); //
    console.log(`Servidor corriendo en el puerto: ${port}`);
});
```

Para la ejecución del servidor pasamos a la terminal de la carpeta donde se corre el comando node server.js el cual se encargas de encender el servidor, el listen anterior mente mencionado nos muestra en la terminal el mensaje donde se esta ejecutando el servidor en este caso se muestra el que definimos al inicio el puerto 3000.

```
PS C:\Users\Monto\OneDrive\Documentos\Client-Back\CBDF_OMR_01> node server.js
Servidor corriendo en el puerto: 3000
PS C:\Users\Monto\OneDrive\Documentos\Client-Back\CBDF_OMR_01>
* History restored

Servidor corriendo en el puerto: 3000
```

Una vez encendido el servidor pasamos a hacer las respectivas consultas de la api en insonia, procedemos a crear un nuevo documento donde te pide la url del servidor, colocamos la dirección de http://localhost:3000, esta dirección nos ayuda a verificar las respuestas de las consultas definidas en el código anterior, definimos el tipo de consulta y se le da a send.

Tras darle a send te muestra el mensaje definido para cada consulta la manera de cambiar de tipo de consulta es darle al get o el tipo que se este trabajando, donde te despliega un menú con todos los posibles métodos disponibles para consultar, solo jalan los que se definieron en el código principal.