

Este documento explica el funcionamiento de una API desarrollada con Node.js utilizando el framework Express. Esta API tiene un único endpoint llamado `"/alumno"` que permite guardar información de un alumno en un archivo de texto en formato JSON.

Código:

```
const express = require('express');
const fs = require('fs');

const app = express();
const PORT = 3000;

app.use(express.json());

app.post('/alumno', (req, res) => {
  const { cuenta, nombre, promedio, grado, grupo } = req.body;

  if (!cuenta || !nombre || !promedio || !grado || !grupo) {
    return res.status(400).json({ error: 'Faltan datos obligatorios' });
  }

  const nuevoAlumno = { cuenta, nombre, promedio, grado, grupo };

  fs.writeFile('alumno.json', JSON.stringify(nuevoAlumno, null, 2), (error) => {
    if (error) {
      return res.status(500).json({ error: 'No se pudo guardar la información' });
    }
    res.status(200).json({ mensaje: 'Información del alumno guardada correctamente' });
  });
});
```

```
app.listen(PORT, () => {  
  console.log(`Servidor escuchando en  
http://localhost:${PORT}`);  
});
```

Explicación paso a paso:

1. Importar dependencias:

- express: Framework para crear aplicaciones web y APIs con Node.js.
- fs: Módulo nativo de Node.js que permite trabajar con el sistema de archivos.

2. Inicializar la aplicación y configurar el puerto:

- Se crea una instancia de la aplicación con `express()`.
- Se define el puerto donde el servidor estará escuchando (3000).

3. Middleware `express.json()`:

- Permite que la aplicación pueda leer cuerpos de peticiones con formato JSON.

4. Definición del endpoint POST `"/alumno"`:

- Se recibe una petición HTTP POST en la ruta `/alumno`.
- Se extraen los datos del cuerpo de la petición: `cuenta`, `nombre`, `promedio`, `grado` y `grupo`.
- Si falta alguno de estos datos, se devuelve un error 400.
- Si todos los datos están presentes, se crea un objeto con la información del alumno.
- Ese objeto se guarda en un archivo llamado `alumno.json` usando `fs.writeFile`.
- Si el guardado es exitoso, se responde con un mensaje de éxito; si ocurre un error, se responde con un mensaje de fallo.

5. Inicialización del servidor:

- Se pone en marcha el servidor y se muestra un mensaje indicando la URL donde está funcionando.

Esta API permite guardar la información de un alumno enviando una

petición POST a la ruta /alumno, con los datos requeridos en formato JSON. Si los datos son correctos, se guarda la información en un archivo alumno.json en el servidor.

-VALDEZ ROJAS LUIS EDUARDO 2-1