Universidad Laica Eloy Alfaro de Manabí

Facultad de ciencias de la vida y tecnología.

Tecnologías de la información

Tema:

“Practica #4 ”

Docente Tutor:

Ing. John Antonio Cevallos Macias

Integrantes:

Cañizares Mantuano Gustavo Johan

Mieles Álvarez Diego Jose

2024(1)

Ilustración 1 Modelo para obtener el get de la estructuración de mis compañeros 3
Ilustración 2 Estructura del HttpClient usando Axios 3
Ilustración 3 Resultado final de la peticion de mi compañero con axios 4
Ilustración 4 Estructura del HttpClient con Superagent 4
Ilustración 5 resultado final de con el cambio de axios a superagent al realizar la peticion 5

Evidencia:

Ilustración 1 Modelo para obtener el get de la estructuración de mis compañeros

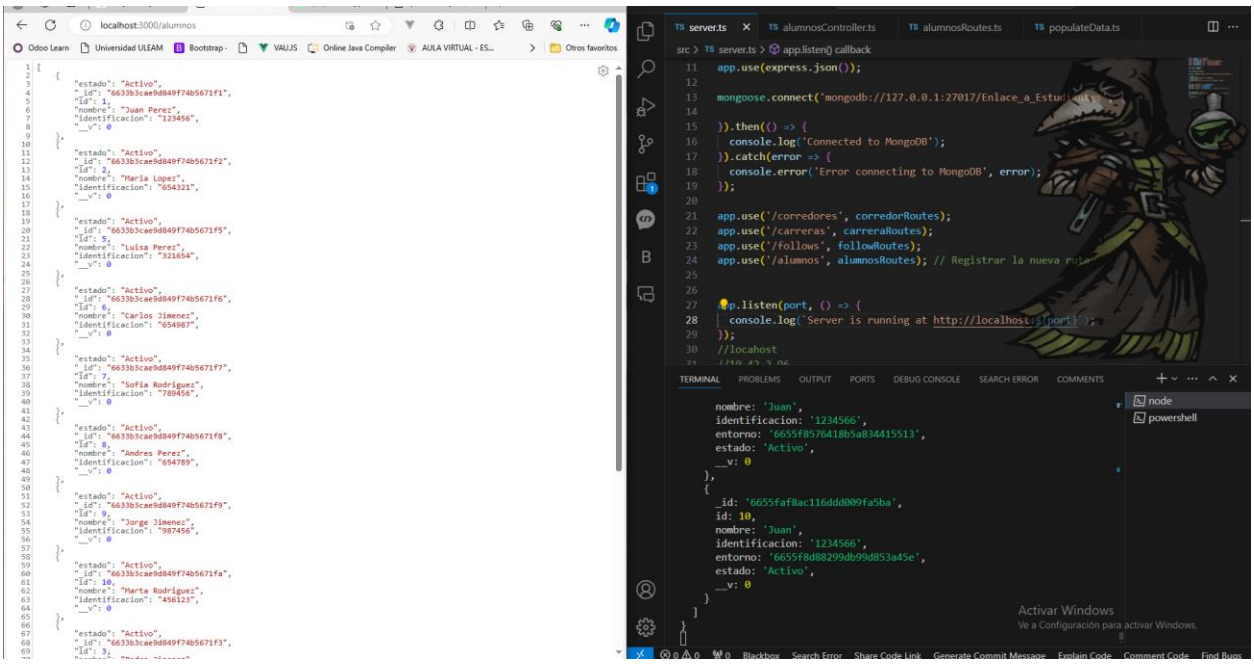
```
12  import { Request, Response } from 'express';
13  import httpClient from '../services/httpClient';
14
15  export const getAlumnos = async (req: Request, res: Response) => {
16    try {
17      const data = await httpClient.get('http://10.42.4.44:3000/api/alumnos');
18      res.status(200).json(data);
19    } catch (error) {
20      res.status(500).json({ message: 'Error fetching alumnos data' });
21    }
22  };
23
```

Evidencia del uso compartido de datos de otro gestor de un compañero con el tema de estudiantes con axios:

Ilustración 2 Estructura del HttpClient usando Axios

```
services > TS httpClient.ts > ...
  import axios, { AxiosInstance, AxiosRequestConfig, AxiosResponse } from 'axios

  class HttpClient {
    private client: AxiosInstance;

    constructor(baseURL: string) {
      this.client = axios.create({ baseURL });
    }

    public async get<T>(url: string, config?: AxiosRequestConfig): Promise<Axios
      return this.client.get<T>(url, config);
    }

    public async post<T>(url: string, data?: any, config?: AxiosRequestConfig):
      return this.client.post<T>(url, data, config);
    }

    public async put<T>(url: string, data?: any, config?: AxiosRequestConfig): P
      return this.client.put<T>(url, data, config);
    }

    public async delete<T>(url: string, config?: AxiosRequestConfig): Promise<A
      return this.client.delete<T>(url, config);
    }
  }

  const httpClient = new HttpClient('http://localhost:3000');

  export default httpClient;
```

Ilustración 3 Resultado final de la peticion de mi compañero con axios



Uso del superagent en ves axios:

Ilustración 4 Estructura del HttpClient con Superagent

```
const superagent = require('superagent');

const get = async (url: string, params: any = {}) => {
  try {
    const response = await superagent.get(url).query(params);
    return response.body;
  } catch (error) {
    throw new Error('error');
  }
};

const post = async (url: string, data: any) => {
  try {
    const response = await superagent.post(url).send(data);
    return response.body;
  } catch (error) {
    throw new Error('error');
  }
};

const put = async (url: string, data: any) => {
  try {
    const response = await superagent.put(url).send(data);
    return response.body;
  } catch (error) {
    throw new Error('error');
  }
};

const del = async (url: string) => {
  try {
    const response = await superagent.del(url);
    return response.body;
  } catch (error) {
    throw new Error('error');
  }
};

export default { get, post, put, del };
```

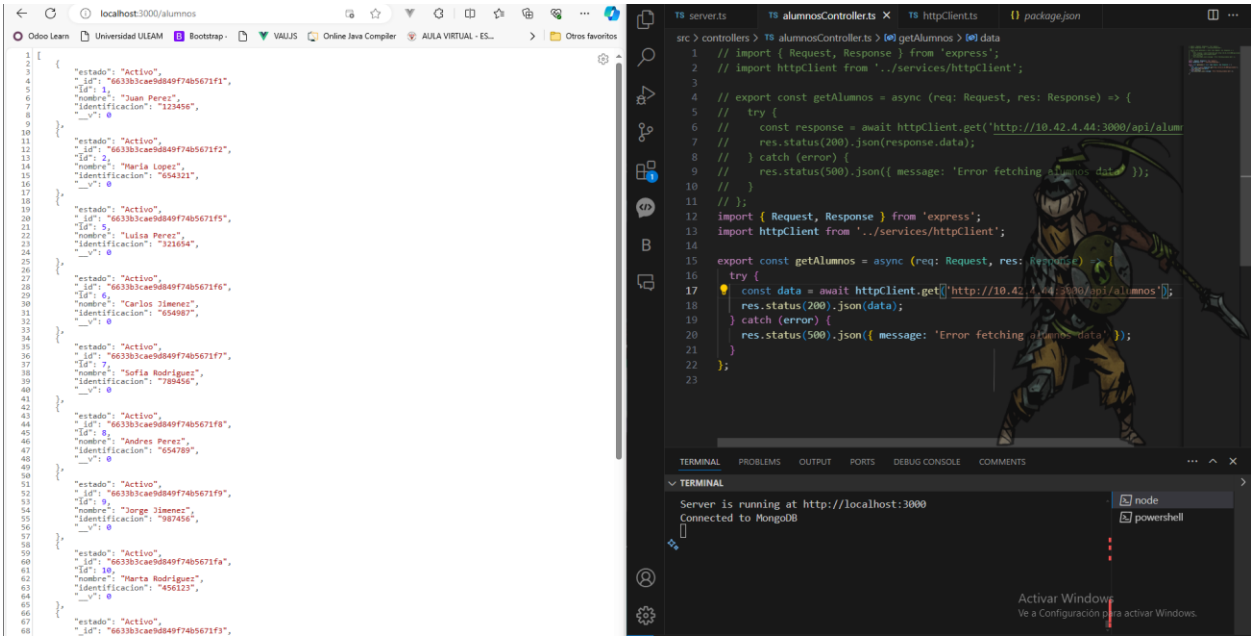



Ilustración 5 resultado final de con el cambio de axios a superagent al realizar la peticion