Código en: <https://github.com/sharon3snc/EntregablesBackend.git>

Primero hay que ver cómo es la estructura de los datos, copiar y crear un archivo .json en el proyecto:

Texto

Descripción generada automáticamente

Leemos el fichero y si da error, devolvemos error, sino, resolvemos el JSON (devolver el JSON parseado). Es una función asíncrona, se metería en el stack del bucle de eventos.

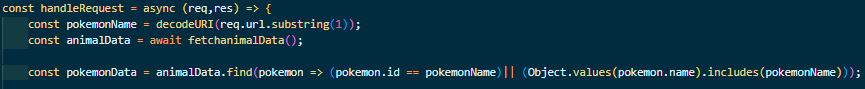
Texto

Descripción generada automáticamente

**handleRequest** es otra función asíncrona.

Primero coge el nombre del pokemon de la URL (pokemonName), y llamamos a la función fetchanimalData.

Luego buscamos en animalData, esto verifica si en el json está el id del pokemon o si dentro de los valores de pokemon.name está incluido el nombre del pokemon que hemos puesto en la URL(pokemonName). Devuelve el primero que encuentre.



Si lo encuentra, responde con los datos requeridos, con identación de 2 espacios.

Texto

Descripción generada automáticamente

Sino, dará un error 404 (Pokemon no encontrado)

Texto

Descripción generada automáticamente

Creamos el servidor y hacemos que escuche en el puerto 3000

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

Aquí podemos ver que funciona correctamente:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Conclusiones y observaciones:

- Es importante manejar correctamente los datos de entrada para poder buscar y encontrar la información requerida.

- Para leer el archivo JSON he utilizado el método fs.readFile (método asíncrono)

- Para encontrar el Pokémon por su ID o por su nombre en cualquier idioma, utilicé el método find de los arrays de JavaScript, junto con la función Object.values para buscar en los nombres del Pokémon en todos los idiomas.

- Con el método decodeURI se decodifica la URL que contiene el nombre del Pokémon. Esto es importante porque los nombres de los Pokémon pueden contener caracteres especiales que deben ser codificados en la URL.

- Con el método JSON.stringify convertí el objeto de respuesta en una cadena JSON que se puede enviar como respuesta HTTP.

- Con los códigos de estado HTTP 200 y 404 se indica si se encontró el Pokémon solicitado y lanzo un mensaje de error personalizado si no se encuentra el Pokémon solicitado.