KOIBANX

EJERCICIO DE CÓDIGO BACKEND

Descripción del problema

a. Un cliente frontend necesita un servicio REST que permita llenar los contenidos de una tabla, y crear nuevas entradas.
b. La tabla es la siguiente:

ID	Comercio	CUIT	Concepto 1	Concepto 2	Concepto 3	Concepto 4	Concepto 5	Concepto 6	Balance actual	Activo	Ultima venta

c. Tipos de datos:
- ID: string (ObjectId de Mongoose)
- Comercio: string
- CUIT: string
- Conceptos: array (debe estar separado y ordenado en la respuesta de la API)
- Balance actual: number (formateado como currency)
- Activo: boolean (devolver “Sí” / “No”)
- Última venta: Date (formateada para legibilidad)

d. La API trabaja con la siguiente especificación en cuanto a los parámetros: https://restdb.io/docs/querying-with-the-api#restdb.
e. El endpoint para traer los registros tiene que poder recibir page y limit como parámetro.
f. La respuesta de la API tiene que tener el siguiente formato:

```
{
 data: [...], //comercios con el formato mostrado anteriormente
 page: 1,
 pages: 100,
 limit: 10,
 total: 1000
}
```

Requerimientos

a. Un endpoint que devuelva la lista paginada y decorada de Comercios. La URL es /api/stores. Tiene que validar que quien ingresa sea un usuario de la aplicación.
b. Un endpoint que permita la creación de nuevos Comercios. Tiene que validar que quien ingresa sea un usuario de la aplicación y que el body que se carga sea válido (estén todos los campos y tengan el formato correcto).

Consideraciones

a. El ejercicio debe ser resuelto en Node.js y Express.
b. Se utiliza Mongo / Mongoose para el manejo de datos.
c. La conexión con la base de datos ya está provista y está escrito el modelo de Store.
d. Los endpoints tienen que estar autenticados. El método a usar es basicAuth (usuario: test@koibanx.com, contraseña: test123).
e. El usuario ya viene creado, la password está hasheada mediante bcrypt.
f. La estructura de las rutas está dada, la implementación es libre.
g. Utilizar decorators para formatear los datos del GET.
h. Se deben crear tests unitarios para todas las funcionalidades.
i. Es un plus crear una función seeder que genere datos de prueba.
j. Es un plus el manejo de errores tanto de autenticación como de dominio.
k. Es un plus la creación de tests de integración.

Forma de entrega

a. Descargar el zip: Koibanx Backend Challenge
b. Se deberá correr con npm i y luego npm start (añadir las instrucciones y consideraciones necesarias en el README.md)
c. Si requiere algún proceso adicional para instalarlo, deberá explicarlo en el readme.
d. Si se tomó alguna consideración o hipótesis para la confección deberá aclararlo en el readme.
e. Crear un repositorio con el código y los commits hechos.
f. Enviar un email a tech@koibanx.com con asunto ‘Challenge backend’ y con el link del repositorio.

Por dudas o consultas, escribir a tech@koibanx.com