

Resumen del proyecto

Este proyecto es una aplicación web React que da a una base de datos de criptomonedas capacidad básica CRUD (Crear, Leer, Actualizar, Eliminar). Con el fin de asegurarse de que sólo las personas con permiso para hacerlo pueden acceder y editar la base de datos, la aplicación también tiene un sistema de autenticación de usuario.

La función de esta aplicación es proporcionar a los usuarios la posibilidad de explorar, añadir, modificar y eliminar criptomonedas. Ofrece una interfaz sencilla y fácil de usar que facilita a los usuarios la gestión de los datos.

Instalación y configuración

Debes tener Node.js y npm (Node Package Manager) instalados en tu ordenador para poder utilizar este proyecto. Para poner en marcha el proyecto, siga estos pasos:

1. Clona en GitHub el repositorio del proyecto.
2. En su terminal, vaya al directorio del proyecto.
3. Para instalar las dependencias necesarias, ejecute `npm install`.
4. Para iniciar el servidor de desarrollo, escriba `npm start`.
5. Para acceder al programa, vaya a `http://localhost:3000` en su navegador web.

Autenticación

Los JSON Web Tokens (JWTs) son utilizados por la aplicación para autenticar a los usuarios. El servidor crea un JWT y lo envía al cliente cuando un usuario inicia sesión. El cliente guarda el JWT en su memoria local y lo utiliza para verificar la identidad del usuario en cada futura solicitud a la API.

Las etapas del proceso de autenticación son las siguientes:

1. El usuario rellena el formulario de inicio de sesión con su dirección de correo electrónico y su contraseña (componente Inicio de sesión).
2. Las credenciales del usuario se incluyen en una petición POST que el cliente realiza al endpoint. Se comprueba que no exista.
3. El servidor crea un JWT después de verificar las credenciales del usuario.
4. El servidor devuelve el JWT al cliente.
5. El cliente conserva localmente el JWT por cierto tiempo.
6. Para la autenticación de usuarios, el cliente proporciona el JWT junto con cada llamada posterior a la API.

El cliente dirige al usuario a la página de inicio de sesión si el JWT falta o no es válido, y el servidor emite un error 401 No autorizado.

Endpoints de la API

Cuando un usuario desea realizar operaciones CRUD en la base de datos de bitcoin, la aplicación interactúa con una API del lado del servidor. Debido al uso de un token, todos los métodos requieren contener un cuerpo para hacer verificación por parte del backend que el usuario exista y esté autorizado en ese momento para esos cambios.

Se Hhce uso de los siguientes endpoints de la API:

- GET /criptomonedas: Obtiene una lista de la base de datos de todas las criptodivisas.
- Una sola moneda se recupera de la base de datos utilizando el comando GET /cryptocurrencies/:id. En el proyecto actual no está implementado
- POST /criptomonedas: Actualiza la base de datos con una nueva criptomoneda.
- PUT /cryptocurrencies/:id actualiza una criptomoneda por su ID que ya existe en la base de datos.
- Una moneda se elimina de la base de datos por su ID con el comando DELETE /cryptocurrencies/:id.
- POST /auth/login: Devuelve un JWT tras la autenticación del usuario.

Operaciones CRUD

Para la gestión de criptomonedas, el programa ofrece las siguientes operaciones CRUD:

- Crear(Agregar Wallet): Rellenando el formulario de criptomoneda (componente CryptoForm) y enviándolo al servidor, los usuarios pueden añadir nuevas monedas a la base de datos. El registro de la base de datos se añade después de que el servidor haya verificado los datos. La vista de lista de criptomonedas muestra entonces la nueva adición (componente CryptoList).
- Leer(Vista home): La vista de lista de monedas permite a los usuarios ver una lista de todas las criptomonedas de la base de datos (componente CryptoList). Al seleccionar una criptomoneda en particular de la vista de lista, que ofrece una vista detallada de la misma, los usuarios también pueden ver detalles sobre esa moneda en particular (componente CryptoForm).

- Actualización(buy): Al seleccionar una criptomoneda en la vista de lista, los usuarios pueden cambiar sus detalles abriendo el formulario de criptomoneda (componente CryptoForm), que ya tiene rellenos los datos que necesitan. Tras realizar los cambios necesarios, el usuario puede enviar el formulario para actualizar la entrada de la base de datos.
- Eliminar(sell): En la vista de lista de criptomonedas, los usuarios pueden eliminar una criptomoneda existente haciendo clic en el botón Eliminar (componente CryptoList). El servidor elimina la entrada de la base de datos y la visualización de la lista se actualiza para reflejar la modificación.

Las consultas a la API realizadas al servidor se utilizan para implementar todos estos procesos CRUD. El archivo api.js del directorio utils contiene las definiciones de las funciones de la API.

Conclusión

En conclusión, este proyecto React ofrece una interfaz sencilla y accesible para controlar una base de datos de criptomonedas. Permite la capacidad básica de CRUD para manejar criptomoneda y tiene un sistema de autenticación de usuario para asegurar que sólo los usuarios autorizados puedan acceder y alterar la base de datos. Para mejorar aún más la experiencia del usuario, el proyecto se puede ampliar para incluir otras características como capacidades de búsqueda y filtrado.