Universidad autónoma de Sinaloa

Lic. en informática

Desarrollo web del lado del servidor

Docente: José Manuel Cazarez Alderete

Alumno: Sauceda Soto Jesús Alonso

Actividad en clase (react-node) consumir API de conversión

App.jsx funciona como el componente principal de la aplicación React siendo su tarea fundamental la de presentar datos de conversión de monedas al usuario. Para lograr esto, se apoya en los Hooks esenciales de React: useState y useEffect. El useState es clave para manejar el estado interno de la aplicación almacenando la lista de monedas a mostrar un indicador de si los datos están cargándose y si ha surgido algún error.

La lógica central de este componente se encuentra encapsulada en el useEffect. Este Hook se ejecuta justo después de que el componente se ha montado inicialmente lo que dispara el proceso de consumo de la API. Dentro de useEffect se define una función asíncrona dedicada a realizar la solicitud HTTP. Esta solicitud se dirige específicamente a la API de Node.js que se espera esté corriendo en http://localhost:3000/monedas con el objetivo de obtener un array de objetos JSON que contengan la información de cada moneda como su código nombre y tasa de conversión.

Durante la ejecución de esta solicitud se gestionan diversos estados. Inicialmente un indicador de carga se activa para señalar que la recuperación de datos está en progreso. Si la respuesta de la API no es exitosa o se presenta un error durante la petición este se captura se registra en la consola y se actualiza el estado de error del componente para notificar al usuario sobre la falla. Por el contrario si la respuesta es exitosa los datos JSON obtenidos se almacenan en el estado de monedas del componente. Finalmente el indicador de carga se desactiva una vez que el proceso de obtención de datos ha concluido independientemente del resultado.En cuanto a la interfaz de usuario la renderización del componente es condicional. Mientras los datos están cargando se muestra un mensaje indicativo. Si se detecta un error este se presenta visualmente en rojo. Una vez que la carga ha terminado y no hay errores se procede a renderizar una lista de las monedas obtenidas mostrando para cada una su nombre código y tasa de conversión. En esencia este componente actúa como el mediador entre la API de backend y la presentación de sus datos en la interfaz de usuario de React.