

Especificación de la API

La API proporciona funcionalidades para la gestión de usuarios, autenticación, autorización y cierre de sesión. descripción detallada de las características y endpoints de la API:

Dependencias y Configuración:

- Express: Framework para construir el servidor web.
- body-parser: Middleware para analizar las solicitudes entrantes en formato JSON.
- bcrypt: Biblioteca para el hashing de contraseñas.
- cors: Middleware para permitir solicitudes desde diferentes dominios.
- mysql2/promise: Conector de MySQL para Node.js con soporte para promesas.

2.Middleware:

authenticateSession : Middleware para verificar la clave de sesión en todas las solicitudes. Requiere la inclusión de la clave de sesión (`x-session-key`) en las cabeceras de la solicitud.

authorizeAdmin: Middleware que autoriza a usuarios administradores, denegando el acceso a otros.

3. Endpoints:

POST /register: Endpoint para registrar nuevos usuarios. Requiere un nombre de usuario y una contraseña. La contraseña se almacena en la base de datos después de ser hasheada.

- POST /login: Endpoint para iniciar sesión. Requiere un nombre de usuario y una contraseña. Devuelve una clave de sesión en caso de éxito.

- POST /logout: Endpoint para cerrar sesión. Requiere la clave de sesión (`x-session-key`) en las cabeceras.

Scripts del Cliente (Frontend):

- Registro (`register.html`):
 - Contiene un formulario para el registro de nuevos usuarios.
 - Utiliza Fetch API para enviar datos de registro al servidor.

Inicio de Sesión (`login.html`):

- Contiene un formulario para el inicio de sesión.

- Utiliza Fetch API para enviar datos de inicio de sesión al servidor.

Cierre de Sesión:

- Se maneja mediante un botón de cierre de sesión.
- Utiliza Fetch API para enviar una solicitud de cierre de sesión al servidor.

Inicio del Servidor:

- La función `startServer`` establece una conexión a la base de datos y luego inicia el servidor Express en el puerto especificado (predeterminado: 3000).

Manejo de Sesiones:

- Se generan claves de sesión encriptadas mediante la función `generateSessionKey``.

Cierre de Conexiones a la Base de Datos:

- Se cierran las conexiones a la base de datos después de cada consulta o transacción.

Manejo de Errores:

- Los errores se manejan y se devuelven mensajes de error y códigos de estado HTTP apropiados.