

Documentación del Proyecto de API de Chat

Descripción

Este proyecto es una API de chat que permite a los usuarios registrarse, iniciar sesión, enviar y recibir mensajes en tiempo real utilizando Socket.IO y Express.js. La base de datos utilizada es MySQL.

Estructura del Proyecto

```
.env
.mocharc.cjs
client/
  index.html
  login.html
  login.js
package.json
server/
  controllers/
    messageControllers.js
    userControllers.js
  index.js
  routes/
    messageRoutes.js
    userRoutes.js
  services/
    dbService.js
  test/
    helloWorld.test.js
  utils/
    authMiddleware.js
    logger.js
```

Archivos y Directorios

- **client/**: Contiene los archivos del cliente, incluyendo HTML y JavaScript.
 - `index.html` : Página principal del cliente.
 - `login.html` : Página de inicio de sesión.
 -

`login.js`

: Lógica de inicio de sesión del cliente.

- **server/**: Contiene los archivos del servidor.

controllers/: Controladores que manejan la lógica de negocio.

`messageControllers.js`

: Controladores para manejar mensajes.

-

`userControllers.js` : Controladores para manejar usuarios.

routes/: Define las rutas de la API.

`messageRoutes.js`

: Rutas para manejar mensajes.

-

`userRoutes.js`

: Rutas para manejar usuarios.

services/: Servicios que proporcionan funcionalidades auxiliares.

`dbService.js`

: Configuración y conexión a la base de datos MySQL.

utils/: Utilidades y middleware.

`authMiddleware.js`

: Middleware de autenticación con JWT.

-

logger.js

: Configuración de logging con Morgan.

- **test/**: Contiene los archivos de pruebas.
 - `helloWorld.test.js`: Archivo de prueba de ejemplo.
-

index.js

: Archivo principal del servidor.

- **.env**: Archivo de configuración de variables de entorno.
- **.mocharc.cjs**: Configuración de Mocha para pruebas.
- **package.json**: Archivo de configuración de npm.

Instalación

1. Clona el repositorio.
2. Instala las dependencias:

```
npm install
```

1. Configura las variables de entorno en el archivo

.env

:

```
DB

_USER=root
DB_HOST=localhost
DB_NAME=sofchat_db
DB_PASSWORD='root'
```

```
DB_PORT=3306
JWT_SECRET=s0fmar17052006
```

Ejecución

Para iniciar el servidor en modo desarrollo:

```
npm run dev
```

Endpoints de la API

Autenticación

- **POST /api/signup:** Registro de usuario.
 - Body: `{ "username": "string", "password": "string" }`
- **POST /api/login:** Inicio de sesión de usuario.
 - Body: `{ "username": "string", "password": "string" }`

Mensajes

- **GET /api/messages:** Obtener todos los mensajes.
 - Headers: `Authorization: Bearer <token>`
- **POST /api/messages:** Insertar un nuevo mensaje.
 - Headers: `Authorization: Bearer <token>`
 - Body: `{ "msg": "string" }`

Socket.IO

- **connection:** Evento de conexión de un usuario.
- **disconnect:** Evento de desconexión de un usuario.
- **message:** Evento para enviar un mensaje.

Pruebas

Para ejecutar las pruebas:

```
npx mocha
```

Ejemplo de Prueba

Archivo de prueba de ejemplo en
helloWorld.test.js

:

```
import { expect } from 'chai';

describe('Hello World Test', () => {
  it('should return true', () => {
    expect(true).to.be.true;
  });
});
```

Contribuciones

Las contribuciones son bienvenidas. Por favor, abre un issue o un pull request para discutir cualquier cambio que desees realizar.

Licencia

Este proyecto está licenciado bajo la Licencia MIT.

Esta documentación proporciona una visión general del proyecto y cómo configurarlo, ejecutarlo y probarlo. Puedes ampliarla según sea necesario para incluir más detalles específicos de tu implementación.