**Node.js Role-Based Login API Documentation**

**Overview**

This document outlines the setup and usage of the Node.js Role-Based Login API project. The API allows users to register, log in, and access specific routes based on their assigned roles. Key technologies used include JWT authentication, the Zod library for validation, MongoDB as the database, and Mongoose as the ODM.

**Setup Instructions**

1**. Clone the Repository:**

git clone <https://github.com/Harshsharma836/ReactNode-Sandbox.git>

2. **Change directory to the Node.js Role-Based Login API folder:**

cd Node.js Role-Based Login API

3. **Install Dependencies:**

npm install

4. **Environment Setup:**

Create an `.env` file in the root directory with the following variables:

PORT=3001

SECRET=YOUR\_SECRET\_KEY

MONGODB\_URI=YOUR\_MONGODB\_URI

Replace `YOUR\_SECRET\_KEY` with your JWT secret key and `YOUR\_MONGODB\_URI` with your MongoDB connection string.

5. **Start the Server:**

npm start

**API Endpoints :**

- `POST /api/auth/register`: Register a new user. Payload should include `username`, `password`, and `roles`.

- `POST /api/auth/login`: Log in an existing user. Payload should include `username` and `password`.

- `GET /api/auth/profile`: Get user profile information. Requires a valid JWT in the `Authorization` header.

- ‘GET /api/admin/dashboard`: Get admin dashboard information. Requires a valid JWT with an `admin` role in the `Authorization` header.

**JWT Authentication**

- JWT is used for authentication. Upon successful login, a JWT is generated and returned in the response.

- JWT is required for accessing protected routes. Include it in the `Authorization` header as `Bearer YOUR\_JWT\_TOKEN`.

**Role-Based Access Control (RBAC)**

- Two roles are defined: `user` and `admin`.

- Middleware is implemented to check user roles before allowing access to specific routes.

- Certain routes are restricted to only `admin` role.

**Testing**

- Unit tests are written using the Mocha testing framework.

- Run tests using `npm run test`.

**Conclusion**

The Node.js Role-Based Login API project provides a robust foundation for building role-based authentication systems. It offers flexibility for customization and extension. Refer to the codebase for further details and modifications.  
  
GitHub Repo : <https://github.com/Harshsharma836/ReactNode-Sandbox>

Postman Collection : <https://www.postman.com/planetary-firefly-68128/workspace/reactnode-sandbox/collection/30161518-c1082ba8-1815-4c2a-880e-e6f14c4770bc?action=share&creator=30161518>