## API Documentation

## Introduction

## *This API documentation outlines the endpoints and functionality for user authentication in [Your Application Name]. Users can register, log in, log out, and refresh access tokens using this API.*

## Authentication

## *This API uses JSON Web Tokens (JWT) for authentication. To access protected routes, clients must include an authorization header with a valid JWT token.*

## Base URL

## *The base URL for all API endpoints is: https://your-api-domain.com*

## Authentication Routes

## 1. User Registration

## ***URL:*** */auth/register*

## ***Method:*** *POST*

## ***Description:*** *Register a new user.*

## ***Request Body:***

## *email (string, required): User's email address.*

## *password (string, required): User's password.*

## *name (string, optional): User's name.*

## ***Example Request:***

## *jsonCopy code*

## *{ "email": "user@example.com", "password": "StrongPassword123", "name": "John Doe" }*

## ***Response:***

## *Status: 201 Created*

## *Body: { "user": { "email": "user@example.com", "name": "John Doe" } }*

## ***Error Responses:***

## *400 Bad Request: Invalid request format.*

## *409 Conflict: Email already in use.*

## *500 Internal Server Error: Registration failed.*

## 2. User Login

## ***URL:*** */auth/login*

## ***Method:*** *POST*

## ***Description:*** *Log in an existing user.*

## ***Request Body:***

## *email (string, required): User's email address.*

## *password (string, required): User's password.*

## ***Example Request:***

## *jsonCopy code*

*{ "email": "user@example.com", "password": "StrongPassword123" }*

## ***Response:***

## *Status: 200 OK*

## *Body: { "user": { "email": "user@example.com", "name": "John Doe" }, "token": "JWT-Token-Here" }*

## ***Error Responses:***

## *401 Unauthorized: Incorrect credentials.*

## *500 Internal Server Error: Login failed.*

## 3. User Logout

## ***URL:*** */auth/logout*

## ***Method:*** *POST*

## ***Description:*** *Log out the currently authenticated user.*

## ***Authentication:*** *Required (Include JWT token in the Authorization header)*

## ***Example Request:***

## *Headers:*

## *Authorization: Bearer JWT-Token-Here*

## ***Response:***

## *Status: 200 OK*

## *Body: { "message": "Logged out successfully" }*

## ***Error Responses:***

## *401 Unauthorized: Invalid or expired token.*

## *500 Internal Server Error: Logout failed.*

## 4. Refresh Access Token

## ***URL:*** */auth/refresh-token*

## ***Method:*** *POST*

## ***Description:*** *Refresh an expired access token using a valid refresh token.*

## ***Authentication:*** *Required (Include JWT token in the Authorization header)*

## ***Request Body:***

## *refreshToken (string, required): User's refresh token.*

## ***Example Request:***

## *jsonCopy code*

## *{ "refreshToken": "Refresh-Token-Here" }*

## ***Response:***

## *Status: 200 OK*

## *Body: { "accessToken": "New-Access-Token-Here" }*

## ***Error Responses:***

## *401 Unauthorized: Invalid or expired refresh token.*

## *500 Internal Server Error: Token refresh failed.*

## Rate Limiting

## *To prevent abuse, rate limiting is applied to certain endpoints. Users are limited to a maximum of X requests per minute.*

## Local Setup Instructions

## *To set up and run the application locally, follow these steps:*

## Prerequisites

## Make sure you have the following software and tools installed:

## [*Node.js*](https://nodejs.org/)

## [*npm*](https://www.npmjs.com/)

## [*MongoDB*](https://www.mongodb.com/)

## Clone the Repository

## *git* [*clone https://github.com/thenish251/login.git*](clone%20https:/github.com/thenish251/login.git)

## Install Dependencies

## *npm install*

## Start the Application

## *npm start*

## The application will start, and you can access it at http://localhost:3000.