

NestJS Salon Booking API

Documentation

This document outlines the functionalities and usage of the NestJS Salon Booking API.

Prerequisites

- Node.js and npm (or yarn) installed on your system.
- A MongoDB database instance.

Setup

- Clone the project repository (if applicable).
- Create a file named `.env` in the project root directory.
- Add the following line to your `.env` file, replacing `<your_mongodb_url>` with your actual MongoDB connection string:
- `DATABASE_URL=<your_mongodb_url>`
- Install project dependencies:

\$ npm install

- Generate Prisma client (assuming you're using Prisma for data modeling):

\$ npx prisma generate

- Start the development server:

\$ npm run start:dev



API Endpoints

Authentication

- POST /api/auth/register

This endpoint allows users to register for an account in the salon booking system.

Request Body:

email: Email address of the user (string, required)

password: Password for the new user (string, required)

Response:

Upon successful registration, a JSON object containing the access token will return

- POST /api/auth/login

This endpoint allows users to log in to their existing accounts.

Request Body:

email: Email address of the user (string, required)

password: Password of the user (string, required)

Response:

On successful login, a JSON object containing a token will be returned. This token can be used for authorization in subsequent API requests.

If the login credentials are invalid, an error message will be sent.



Availability Management

- POST /api/availability (Requires authorization)

This endpoint allows authorized users (likely salon staff) to set their availability for booking appointments.

Request Body: (Refer to CreateAvailabilityDto class for details)

An array of availability objects, where each object specifies availability for a particular day with time slots and their maximum capacity.

Response:

Upon successful creation of availability, a confirmation message or the created availability data will be returned.

In case of errors (e.g., invalid data), an error message will be sent.

- GET /api/available-slots/:date (Requires authorization)

This endpoint retrieves available time slots for booking appointments on a specific date.

URL Parameter:

date: Date for which to retrieve available slots (format: YYYY-MM-DD, required)

Response:

A JSON object containing available time slots for the requested date will be returned.

If no slots are available on the specified date, an appropriate message will be sent.



Booking Management

- POST /api/bookings (Requires authorization)

This endpoint allows authorized users to create booking appointments.

Request Body: (Refer to CreateBookingDto class for details)

User ID of the person making the booking (number, required)

Date of the booking (format: YYYY-MM-DD, required)

Selected time slot object (start and end times, required)

Response:

Upon successful booking creation, a confirmation message or the created booking data will be returned.

In case of errors (e.g., unavailable slot, user conflicts), an error message will be sent.

- GET /api/bookings (Requires authorization)

This endpoint retrieves a list of existing booking appointments (likely for authorized users with appropriate permissions).

Response:

A JSON array containing booking data. If no bookings are found, an empty array will be returned.