

Appointment Booking API Documentation

Version: 1.0.0

Base URL (Local): <http://localhost:3000/api>

Introduction

This API provides a complete backend solution for managing professional appointment bookings. It allows an administrator to define their availability and enables users to view, reserve, and confirm appointment slots. It is designed to be a "headless" service, meaning it manages the logic and data, but any frontend application can be built to interact with it.

Authentication

The API has two modes for handling admin authentication, managed in `/middleware/auth.js`.

1. Development Mode (Current)

- **Authentication is bypassed.** Any call to an admin route will be allowed without a token. A `console.warn` message will appear in the server logs to indicate this.

2. Production Mode (JWT)

- **Authentication is enforced using JSON Web Tokens (JWT).**
- The admin must first send their password to the `/admin/login` endpoint to receive a token.
- This token must then be included in the Authorization header for all subsequent requests to protected admin routes.
- **Header Format:** `Authorization: Bearer <your_jwt_token>`

Endpoints

Admin Routes

1. Admin Login

Generates a JWT for authenticating administrative requests.

- **Endpoint:** `POST /admin/login`
- **Auth Required:** No
- **Request Body:**

```
{  "password": "your-secret-password"}
```

- **Success Response (200 OK):**

```
{
  "message": "Login successful",
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyYb2xlljoiYWWRtaW4iLCJpYXQiOiJE2Nz..."
}
```

- **Error Response (401 Unauthorized):**

```
{
  "message": "Invalid credentials."
}
```

2. Set Availability

Creates all available 30-minute appointment slots for a given date and time ranges. This will overwrite any existing slots for that specific date.

- **Endpoint:** POST /availability
- **Auth Required: Yes (Admin)**
- **Request Body:**

```
{
  "date": "YYYY-MM-DD",
  "timeRanges": [
    {
      "start": "HH:mm",
      "end": "HH:mm"
    }
  ]
}
```

- **Success Response (201 Created):**

```
{
  "message": "Availability set successfully.",
  "slotsCreated": 5
}
```

- **Error Responses:**

- 400 Bad Request: Validation error (e.g., invalid date format).

- 401 Unauthorized: Missing or invalid JWT.
- 403 Forbidden: Token is valid but does not have admin privileges.

User Routes

3. Get Available Slots

Retrieves a list of all appointment slots that are currently available and are in the future. It also clears any pending slots where the 5-minute lock has expired.

- **Endpoint:** GET /slots
- **Auth Required:** No
- **Success Response (200 OK):**

```
[
  {
    "id": "2025-10-20T10:00",
    "date": "2025-10-20",
    "time": "10:00",
    "status": "available",
    "lockExpiresAt": null,
    "customer": null
  },
  {
    "id": "2025-10-20T10:30",
    "date": "2025-10-20",
    "time": "10:30",
    "status": "available",
    "lockExpiresAt": null,
    "customer": null
  }
]
```

- **Error Response (500 Internal Server Error):**
 - Occurs if the server fails to fetch the data.

4. Temporarily Book a Slot

Places a temporary 5-minute hold on a specific slot by changing its status to pending. This is the first step in the booking process, intended to be called before the user makes a payment.

- **Endpoint:** POST /slots/:id/book
 - **Example:** POST /api/slots/2025-10-20T10:00/book

- **Auth Required:** No
- **Success Response (200 OK):**

```
{
  "message": "Slot reserved for 5 minutes. Please complete payment to confirm.",
  "slotId": "2025-10-20T10:00",
  "expiresAt": "2025-06-20T12:05:00.000Z"
}
```
- **Error Responses:**
 - 404 Not Found: The requested slotId does not exist.
 - 409 Conflict: The slot is not available (it may already be pending or booked).

5. Confirm a Booking

Finalizes an appointment by changing a pending slot's status to booked. This should be called after a user has successfully completed payment.

- **Endpoint:** POST /slots/:id/confirm
 - **Example:** POST /api/slots/2025-10-20T10:00/confirm
- **Auth Required:** No
- **Success Response (200 OK):**

```
{
  "message": "Booking confirmed successfully.",
  "slot": {
    "id": "2025-10-20T10:00",
    "date": "2025-10-20",
    "time": "10:00",
    "status": "booked",
    "lockExpiresAt": null,
    "customer": null
  }
}
```
- **Error Responses:**
 - 400 Bad Request: The slot is not in a pending state or its lock has expired.
 - 404 Not Found: The requested slotId does not exist.