

Start coding or generate with AI.

# API Documentation: Hospital Doctor Appointment System

This document provides detailed documentation for the backend API of the Hospital Doctor Appointment System. The API is built with Node.js and Express and follows RESTful principles.

## Base URL

All API endpoints are prefixed with `/api`.

- **Development:** `http://localhost:4000/api`
- **Production:** `https://your-deployed-backend-url.com/api`

## Authentication

- **Strategy:** The API uses JSON Web Tokens (JWT) for authentication.
- **Process:**
  1. A user, doctor, or admin logs in with their credentials.
  2. Upon successful login, the server returns a JWT.
  3. This token must be included in the `Authorization` header for all subsequent requests to protected endpoints in the format `Bearer <token>`.
- **Middleware:**
  - `authUser`: Protects routes accessible only by authenticated patients/users.
  - `authDoctor`: Protects routes accessible only by authenticated doctors.
  - `authAdmin`: Protects routes accessible only by authenticated admins.

## Common Error Responses

The API uses standard HTTP status codes to indicate the success or failure of a request.

Status Code	Meaning	Description
400 Bad Request	The server could not understand the request due to invalid syntax or missing parameters.	
401 Unauthorized	The request requires user authentication, but the JWT is missing or invalid.	
403 Forbidden	The authenticated user does not have the necessary permissions to access the resource.	
404 Not Found	The requested resource could not be found on the server.	
409 Conflict	The request could not be completed due to a conflict, such as a duplicate email during registration.	
500 Internal Server Error	An unexpected error occurred on the server.	

## 1. User API Endpoints

Route Prefix: `/api/user`

`/register`

- **Method:** `POST`
- **Auth:** Public
- **Description:** Registers a new user.
- **Request Body:**

```
{
  "name": "John Doe",
  "email": "john.doe@example.com",
  "password": "password123"
}
```

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

```
{
  "success": true,
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

- **Error Responses:**

- **400 Bad Request**: If name, email, or password are not provided.
- **409 Conflict**: If a user with the provided email already exists.

### /login

- **Method:** **POST**
- **Auth:** Public
- **Description:** Logs in an existing user.
- **Request Body:**

```
{
  "email": "john.doe@example.com",
  "password": "password123"
}
```

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

```
{
  "success": true,
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

- **Error Responses:**

- **400 Bad Request**: If email or password are not provided.
- **404 Not Found**: If no user with the provided email exists.
- **401 Unauthorized**: If the password is incorrect.

### /get-profile

- **Method:** **GET**
- **Auth:** **authUser**
- **Description:** Retrieves the profile of the logged-in user.
- **Example Request:**

```
curl -X GET http://localhost:4000/api/user/get-profile \
-H "Authorization: Bearer <your_jwt_token>"
```

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

```
{
  "success": true,
  "data": {
    "_id": "60d0fe4f5b3e4b3f8c8b4567",
    "name": "John Doe",
    "email": "john.doe@example.com",
    "image": "https://res.cloudinary.com/..."
  }
}
```

### /book-appointment

- **Method:** **POST**
- **Auth:** **authUser**

- **Description:** Books a new appointment with a doctor.
- **Request Body:**

```
{
  "doctorId": "60d0fe4f5b3e4b3f8c8b4568",
  "date": "2025-12-25",
  "time": "10:00 AM"
}
```

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

```
{
  "success": true,
  "message": "Appointment booked successfully"
}
```

- **Error Responses:**

- **404 Not Found:** If the `doctorId` does not correspond to an existing doctor.

---

## 2. Doctor API Endpoints

Route Prefix: `/api/doctor`

`/list`

- **Method:** `GET`
- **Auth:** Public
- **Description:** Retrieves a list of all available doctors.
- **Success Response (200 OK):**

```
{
  "success": true,
  "data": [
    {
      "_id": "60d0fe4f5b3e4b3f8c8b4568",
      "name": "Dr. Jane Smith",
      "speciality": "Cardiology",
      "experience": 10,
      "fee": 150,
      "image": "https://res.cloudinary.com/..."
    }
  ]
}
```

`/login`

- **Method:** `POST`
- **Auth:** Public
- **Description:** Logs in a doctor.
- **Request Body:**

```
{
  "email": "jane.smith@example.com",
  "password": "password123"
}
```

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

```
{
  "success": true,
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

### /appointments

- **Method:** GET
- **Auth:** authDoctor
- **Description:** Lists all appointments for the logged-in doctor.
- **Success Response (200 OK):**

```
{
  "success": true,
  "data": [
    {
      "_id": "60d0fe4f5b3e4b3f8c8b4569",
      "userId": { "name": "John Doe" },
      "date": "2025-12-25T00:00:00.000Z",
      "time": "10:00 AM",
      "status": "Pending"
    }
  ]
}
```

## 3. Admin API Endpoints

**Route Prefix:** /api/admin

### /login

- **Method:** POST
- **Auth:** Public
- **Description:** Logs in an admin.
- **Request Body:**

```
{
  "email": "admin@example.com",
  "password": "adminpassword"
}
```

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

```
{
  "success": true,
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

### /add-doctor

- **Method:** POST
- **Auth:** authAdmin
- **Description:** Adds a new doctor to the system. This endpoint expects multipart/form-data.
- **Request Body (FormData):**
  - name: "Dr. Peter Jones"
  - email: peter.jones@example.com
  - password: doctorpassword
  - speciality: "Neurology"

- `experience`: 8
- `fee`: 200
- `phone`: "1234567890"
- `image`: (file)

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

```
{
  "success": true,
  "message": "Doctor added successfully"
}
```

- **Error Responses:**

- `409 Conflict`: If a doctor with the provided email already exists.

`/dashboard`

- **Method:** `GET`
- **Auth:** `authAdmin`
- **Description:** Retrieves dashboard data for the admin, including total counts.
- **Success Response (200 OK):**

```
{
  "success": true,
  "data": {
    "users": 50,
    "doctors": 10,
    "appointments": 120
  }
}
```

## Data Models

### User Model

```
{
  "_id": "ObjectId",
  "name": "String",
  "email": "String (unique)",
  "password": "String (hashed)",
  "image": "String (URL)",
  "role": "String (default: 'user')"
}
```

### Doctor Model

```
{
  "_id": "ObjectId",
  "name": "String",
  "email": "String (unique)",
  "password": "String (hashed)",
  "image": "String (URL)",
  "speciality": "String",
  "experience": "Number",
  "fee": "Number",
  "phone": "String",
  "isAvailable": "Boolean",
}
```

```
"role": "String (default: 'doctor')"  
}
```

## Appointment Model

```
{  
  "_id": "ObjectId",  
  "userId": "ObjectId (ref: 'User')",  
  "doctorId": "ObjectId (ref: 'Doctor')",  
  "date": "Date",  
  "time": "String",  
  "status": "String (enum: ['Pending', 'Completed', 'Cancelled'])",  
  "paymentStatus": "String (enum: ['Pending', 'Paid'])"  
}
```

Start coding or [generate](#) with AI.