

Appointment Management API Documentation

Overview

This API provides endpoints for managing hospital appointments, including booking, viewing history, rescheduling, canceling, and generating statistics.

Base URL: `http://localhost:5239/api`

Version: 1.0.0

Authentication

Currently, the API does not require authentication headers. JWT tokens are generated at login but not enforced on these endpoints.

Endpoints

1. Book Appointment

Creates a new appointment for a patient with a doctor.

Endpoint: `POST /appointments`

Request Body:

```
json

{
  "patientId": "string (required)",
  "doctorId": "string (required)",
  "date": "string (required, format: YYYY-MM-DD)",
  "time": "string (required, format: HH:mm)",
  "reason": "string (optional)"
}
```

Success Response (200 OK):

```
json

{
  "appointmentId": "A-00000001",
  "message": "Appointment booked successfully"
}
```

Error Responses:

Status Code	Description
400 Bad Request	Missing required fields, invalid date format, or past date
404 Not Found	Patient or Doctor not found
409 Conflict	Time slot already booked
500 Internal Server Error	Server error

Example Request:

```
bash

curl -X POST http://localhost:5239/api/appointments \
-H "Content-Type: application/json" \
-d '{
  "patientId": "P-00000001",
  "doctorId": "D-00000001",
  "date": "2025-12-15",
  "time": "10:00",
  "reason": "Regular checkup"
}'
```

2. Get All Appointments

Retrieves all appointments in the system with patient and doctor details.

Endpoint: GET /appointments

Parameters: None

Success Response (200 OK):

```
json
{
  "data": [
    {
      "appointmentId": "A-00000001",
      "patientId": "P-00000001",
      "patientName": "John Doe",
      "doctorId": "D-00000001",
      "doctorName": "Dr. Amara Silva",
      "department": "Cardiology",
      "appointmentDate": "2025-10-20T00:00:00",
      "appointmentTime": "14:00",
      "status": "Scheduled",
      "reason": "Heart checkup"
    }
  ]
}
```

Error Responses:

Status Code	Description
500 Internal Server Error	Server error

Example Request:

```
bash
curl -X GET http://localhost:5239/api/appointments
```

3. Get Appointment History

Retrieves appointment history for a specific patient.

Endpoint: `GET /appointments/history/{patientId}`

Path Parameters:

- `patientId` (string, required): The unique identifier of the patient

Query Parameters:

- `page` (integer, optional, default: 1): Page number for pagination
- `pageSize` (integer, optional, default: 10): Number of records per page

Success Response (200 OK):

```
json
[
  {
    "appointmentId": "A-00000002",
    "doctorName": "Dr. Amara Silva",
    "department": "Cardiology",
    "date": "2025-12-10",
    "time": "15:30",
    "status": "Scheduled",
    "reason": "Heart problem"
  }
]
```

Error Responses:

Status Code	Description
404 Not Found	Patient not found or no appointment history
500 Internal Server Error	Server error

Example Request:

```
bash
curl -X GET "http://localhost:5239/api/appointments/history/P-00000001?page=1&pageSize=10"
```

4. Reschedule Appointment

Updates the date and time of an existing appointment.

Endpoint: `PUT /appointments/{appointmentId}/reschedule`

Path Parameters:

- `appointmentId` (string, required): The unique identifier of the appointment

Request Body:

```
json
{
  "newDate": "string (required, format: YYYY-MM-DD)",
  "newTime": "string (required, format: HH:mm)"
}
```

Success Response (200 OK):

```
json

{
  "appointmentId": "A-00000001",
  "message": "Appointment rescheduled successfully"
}
```

Error Responses:

Status Code	Description
400 Bad Request	Invalid date format or past date
404 Not Found	Appointment not found
409 Conflict	New time slot already booked
500 Internal Server Error	Server error

Example Request:

```
bash

curl -X PUT http://localhost:5239/api/appointments/A-00000001/reschedule \
-H "Content-Type: application/json" \
-d '{
  "newDate": "2025-12-20",
  "newTime": "11:00"
}'
```

5. Cancel Appointment

Cancels an existing appointment by updating its status to "Cancelled".

Endpoint: `DELETE /appointments/{appointmentId}`

Path Parameters:

- `appointmentId` (string, required): The unique identifier of the appointment

Success Response (200 OK):

```
json

{
  "message": "Appointment cancelled successfully"
}
```

Error Responses:

Status Code	Description
400 Bad Request	Cannot cancel completed or already cancelled appointment
404 Not Found	Appointment not found
500 Internal Server Error	Server error

Example Request:

```
bash
curl -X DELETE http://localhost:5239/api/appointments/A-00000001
```

6. Get Statistics

Generates appointment statistics including total count, appointments per doctor, and appointments per department.

Endpoint: GET /appointments/statistics

Query Parameters:

- from (string, optional, format: YYYY-MM-DD): Start date for filtering
- to (string, optional, format: YYYY-MM-DD): End date for filtering

Success Response (200 OK):

```
json
{
  "totalAppointments": 15,
  "appointmentsPerDoctor": {
    "Dr. Amara Silva": 8,
    "Dr. Rohan Perera": 5,
    "Dr. Nadeesha Fernando": 2
  },
  "appointmentsPerDepartment": {
    "Cardiology": 8,
    "Neurology": 5,
    "Dermatology": 2
  }
}
```

Error Responses:

Status Code	Description
400 Bad Request	Invalid date format
404 Not Found	No data available for selected date range
500 Internal Server Error	Server error

Example Requests:

All-time statistics:

```
bash
curl -X GET http://localhost:5239/api/appointments/statistics
```

Filtered statistics:

```
bash
curl -X GET "http://localhost:5239/api/appointments/statistics?from=2025-10-01&to=2025-12-31"
```

Status Codes

Status Code	Description
200 OK	Request successful
400 Bad Request	Invalid request parameters or body
404 Not Found	Resource not found
409 Conflict	Resource conflict (e.g., duplicate booking)
500 Internal Server Error	Server error

Appointment Status Values

Status	Description
Scheduled	Appointment is confirmed and scheduled
Rescheduled	Appointment has been moved to a different time
Cancelled	Appointment has been cancelled
Completed	Appointment has been completed

Data Models

Appointment Object

json

```
{
  "appointmentId": "string",
  "patientId": "string",
  "patientName": "string",
  "doctorId": "string",
  "doctorName": "string",
  "department": "string",
  "appointmentDate": "datetime",
  "appointmentTime": "string (HH:mm)",
  "status": "string",
  "reason": "string",
  "createdAt": "datetime",
  "updatedAt": "datetime"
}
```

User Stories Covered

1. **US-1:** As a Patient, I want to book appointments online so that I can avoid queues
 - Endpoint: `POST /appointments`
 2. **US-2:** As a Patient, I want to view my appointment history so that I can reference past medical visits
 - Endpoint: `GET /appointments/history/{patientId}`
 3. **US-3:** As an Admin, I want to reschedule appointments when doctors are unavailable so that patient visits are not canceled
 - Endpoints: `GET /appointments`, `PUT /appointments/{id}/reschedule`
 4. **US-4:** As an Admin, I want to generate appointment statistics so that I can analyze hospital performance
 - Endpoint: `GET /appointments/statistics`
-

Testing with Postman

Import this collection to Postman for easy testing:

1. Create a new Collection named "Hospital Appointments"
 2. Add requests for each endpoint above
 3. Set base URL variable: `{{baseUrl}}` = `http://localhost:5239`
 4. Test each endpoint with the provided examples
-

Error Handling

All error responses follow this format:


```
json

{
  "error": "Error message description"
}
```

Some responses may include additional fields:

```
json

{
  "error": "Email already exists",
  "code": 409
}
```

Notes

- All dates should be in `YYYY-MM-DD` format
- All times should be in 24-hour `HH:mm` format
- Appointment IDs follow the pattern: `A-XXXXXXXX`
- Patient IDs follow the pattern: `P-XXXXXXXX`
- Doctor IDs follow the pattern: `D-XXXXXXXX`
- Past dates are not allowed for booking or rescheduling
- Double-booking the same doctor at the same time is prevented

Last Updated: October 7, 2025

Developer: Hospital Management System Team