API DOCTOR BOOKING

# <https://localhost:7031/swagger/v1/swagger.json>

Buka foto misal:

<https://localhost:7031/images/specializations/dentist.jpeg>

<https://localhost:7031/images/doctors/dr_patricia_ahoy.jpeg>

# 1. Entities / Model

* **Users** – Data pengguna
* **Specializations** – Daftar spesialisasi medis, misalnya "Dokter Umum," "Spesialis Gigi," dan lain-lain.
* **Doctors** – Data dokter, termasuk nama, spesialisasi, provider, dan informasi terkait.
* **Providers** – Data penyedia layanan kesehatan, misalnya klinik atau rumah sakit.
* **DoctorSchedules** – Jadwal praktik dokter beserta slot waktu yang tersedia.
* **Appointments** – Data janji temu antara user dan dokter.
* **Reviews** – Ulasan dari user terhadap dokter.

# 2. DTO yang Digunakan

User  
- UserRegisterDto- UserLoginDto- UserProfileDto

Doctors  
- DoctorListDto – GET /api/doctors (list)  
- DoctorDetailDto – GET /api/doctors/{id}  
- DoctorCreateDto – POST /api/doctors  
- DoctorResponseDto – response setelah create/update  
  
DoctorSchedules  
- DoctorScheduleDto – untuk menampilkan jadwal dokter  
- TimeSlotDto – response GET /available

Appointments  
- AppointmentCreateDto – POST appointment  
- AppointmentUpdateDto – PUT appointment  
- AppointmentSummaryDto – list ringkas appointment  
- AppointmentDetailDto – detail appointment

Reviews  
- ReviewDto – GET review  
- ReviewCreateDto – POST review

# 3. Endpoints

**Users**

* POST /api/users/register – Mendaftarkan user baru.
* POST /api/users/login – Login user.
* GET /api/users – Mengambil semua user (hanya untuk admin).
* GET /api/users/{id} – Mengambil profil user berdasarkan ID.
* PUT /api/users/{id}/profile – Mengupdate profil user.
* DELETE /api/users/{id} – Menonaktifkan user.

**Doctors**

Note: Rating (rata-rata) dan ReviewCount (jumlah Review) dokter dihitung otomatis dari tabel Reviews, setiap kali endpoint dipanggil. Bukan disimpan permanen di tabel Doctors.

* GET /api/doctors – Mengambil daftar semua dokter.
* GET /api/doctors/{id} – Mengambil detail informasi dokter berdasarkan ID.
* GET /api/doctors/by-specialization/{specializationId} – Mengambil informasi list dokter berdasarkan specialization
* POST /api/doctors – Menambahkan data dokter baru.
* PUT /api/doctors/{id} – Mengupdate data dokter.
* DELETE /api/doctors/{id} – Menonaktifkan dokter.

**Specializations**

* GET /api/specializations – Mengambil daftar semua spesialisasi.
* GET /api/specializations/{id} – Mengambil detail spesialisasi.
* POST /api/specializations – Menambahkan spesialisasi baru.
* PUT /api/specializations/{id} – Mengupdate spesialisasi.
* DELETE /api/specializations/{id} – Menghapus spesialisasi.

**Providers**

* GET /api/providers – Mengambil daftar provider.
* POST /api/providers – Menambahkan provider baru.
* PUT /api/providers/{id} – Mengupdate provider.
* DELETE /api/providers/{id} – Menonaktifkan provider.

**DoctorSchedules**

* GET /api/doctorschedules/doctor/{doctorId} – Mengambil jadwal dokter tertentu.
* GET /api/doctorschedules/available ?doctorId=7&date=2025-07-21 – Menampilkan slot waktu yang tersedia untuk dokter di tanggal tertentu.
* POST /api/doctorschedules – Menambahkan jadwal baru untuk dokter.
* PUT /api/doctorschedules/{id} – Mengupdate jadwal dokter.
* DELETE /api/doctorschedules/{id} – Menghapus jadwal dokter.

**Appointments**

* **GET /api/appointments?status=Scheduled&date=2025-07-17**  
  – Mengambil daftar appointment dengan **filter status dan tanggal**.

**Status**:

 **Scheduled** – Appointment sudah dijadwalkan (default saat dibuat).

 **Completed** – Appointment sudah selesai.

 **Cancelled** – Appointment dibatalkan.

 **NoShow** (opsional) – Pasien tidak hadir.

* **GET /api/appointments/user/{userId}** – Mengambil daftar appointment **aktif** untuk user tertentu (status selain Completed dan Cancelled).
* **GET /api/appointments/history/{userId}** – Mengambil **riwayat appointment** user tertentu dengan status Completed atau Cancelled.
* GET /api/appointments/{id} – Mengambil detail appointment berdasarkan ID.
* POST /api/appointments – Membuat appointment baru.
* POST /api/appointments/qr-appointment – Membuat appointment berdasarkan QR code.

Format QR Code misalnya: DOCTOR-7|2025-07-25|09:00

Di controller, qrCodeData ini akan di-parse untuk ambil DoctorId, AppointmentDate, dan AppointmentTime. **Gunanya:** buat user scan QR di klinik, langsung bisa daftar ke dokter dengan slot yang sudah dipilih.

* GET /api/appointments/queue/{appointmentId} – Melihat posisi antrian appointment.
* PUT /api/appointments/{id} – Mengupdate appointment.
* DELETE /api/appointments/{id} – Membatalkan appointment. Dengan ubah status appointment jadi Cancelled dan set IsActive = false, **tanpa menghapus record dari database**.
* POST /api/appointments/next-queue/{doctorId} - untuk memajukan antrian pasien untuk dokter tertentu pada hari ini.

**Fungsinya:**

1. **Mencari appointment terdekat (next appointment)** dengan status Scheduled untuk dokter doctorId pada **tanggal hari ini**.
2. Jika ada **appointment yang sedang berlangsung (In Progress)**, maka statusnya diubah menjadi Completed.
3. **Appointment berikutnya diubah menjadi In Progress** (artinya pasien ini sekarang sedang dilayani).
4. Mengembalikan response berisi **QueueNumber** dan **AppointmentId** dari pasien yang baru dipanggil.

**Kapan Dipakai?**

* Saat dokter atau sistem ingin **memanggil pasien berikutnya** dalam antrian konsultasi.
* Biasanya tombol ini ada di **dashboard dokter**, seperti tombol "Next Patient" atau "Panggil Selanjutnya".

**Reviews**

* GET /api/reviews/doctor/{doctorId} – Mengambil review berdasarkan dokter.
* POST /api/reviews – Menambahkan review.
* DELETE /api/reviews/{id} – Menghapus review.

# 4. Filtering

- Appointments: Filter by status/date → GET /api/appointments?status=Scheduled&date=2025-07-20  
- DoctorSchedules: GET /api/doctorschedules/available?doctorId=1&date=YYYY-MM-DD

# 5. Contoh Testing (POST & PUT)

## POST Doctor

POST /api/doctors  
Content-Type: application/json  
  
{  
 "fullName": "Dr. Siti Rahma",  
 "email": "siti@clinic.com",  
 "phoneNumber": "08123456789",  
 "photo": "https://example.com/photo.jpg",  
 "specializationId": 1,  
 "providerId": 1,  
 "licenseNumber": "ABC123",  
 "consultationFee": 150000,  
 "biography": "Spesialis jantung dengan pengalaman 10 tahun."  
}

## PUT Doctor

PUT /api/doctors/1  
Content-Type: application/json  
  
{  
 "doctorId": 1,  
 "fullName": "Dr. Siti Rahma Update",  
 "email": "sitiupdate@clinic.com",  
 "phoneNumber": "08123456789",  
 "photo": "https://example.com/photo\_new.jpg",  
 "specializationId": 1,  
 "providerId": 1,  
 "licenseNumber": "ABC123",  
 "consultationFee": 200000,  
 "biography": "Update - Spesialis jantung."  
}

## POST Register

POST /api/users/register  
Content-Type: application/json  
  
{  
 "fullName": "Budi Santoso",  
 "email": "budi@example.com",  
 "phoneNumber": "08123456789",  
 "password": "password123"  
}

## POST Login

POST /api/users/login  
Content-Type: application/json  
  
{  
 "email": "budi@example.com",  
 "password": "password123"  
}

## PUT User

PUT /api/users/1  
Content-Type: application/json  
  
{  
 "fullName": "Budi Santoso Update",  
 "email": "budiupdate@example.com",  
 "phoneNumber": "08123456789",  
 "dateOfBirth": "2000-01-15",  
 "gender": "M",  
 "address": "Jl. Merdeka No. 20, Jakarta"  
}

## POST Specializations

POST /api/ specializations  
Content-Type: application/json  
  
{

"name": "Dermatology",

"description": "Skin and hair specialist",

"icon": "icon\_dermatology.png"

}

## PUT Specializations

PUT /api/ specializations/9  
Content-Type: application/json  
  
{

"name": "Dermatology Advanced",

"description": "Updated description",

"icon": "icon\_dermatology\_updated.png"

}

## POST Provider

POST /api/ Provider  
Content-Type: application/json  
  
{

"name": "RS Premier Jatinegara",

"address": "Jl. Jatinegara Barat No.126, Bali Mester, Jakarta Timur",

"city": "Jakarta",

"googleMapsLink": "https://maps.google.com/premierjatinegara"

}

## PUT Provider

PUT /api/ Provider /5  
Content-Type: application/json  
  
{

"name": "RS Premier Jatinegara (Updated)",

"address": "Jl. Jatinegara Barat No.126, Bali Mester, Jakarta Timur",

"city": "Jakarta Timur",

"googleMapsLink": "https://maps.google.com/premierjatinegara",

"isActive": true

}

## POST DoctorSchedules

POST /api/ DoctorSchedules  
Content-Type: application/json  
  
{

"doctorId": 8,

"dayOfWeek": 7,

"startTime": "08:00",

"endTime": "12:00",

"slotDuration": 30,

"isActive": true

}

## PUT DoctorSchedules

PUT /api/ DoctorSchedules/43  
Content-Type: application/json  
  
{

"doctorID": 9,

"dayOfWeek": 4,

"startTime": "14:00",

"endTime": "18:00",

"slotDuration": 30,

"isActive": true

}

## Alur POST /api/appointments

**1. User memilih dokter dan jadwal** (tanggal + jam konsultasi).

Content-Type: application/json  
{

"userId": 1,

"doctorId": 7,

"appointmentDate": "2025-07-25",

"appointmentTime": "09:00:00",

"patientNotes": "Kontrol rutin",

"symptoms": "Demam dan batuk"

}

**2. Controller memvalidasi**:

* Dokter tersedia di database.
* Jadwal dokter ada di hari tersebut.
* Jam konsultasi tidak bentrok (belum ada pasien lain di slot itu).
* Slot berada di rentang waktu praktik dokter.

**3. QueueNumber di-generate otomatis :** *D{DoctorId}-{ddMM}-{NoUrut}*

Contoh: D7-2507-1 untuk dokter ID 7, tanggal 25 Juli, antrian ke-1.

4. **ConsultationFee diambil otomatis** dari tabel **Doctors**.

5. **Data appointment disimpan ke database** dengan status default **"Scheduled"** dan IsActive = true.

## PUT Appointment

PUT /api/appointments/3  
Content-Type: application/json  
  
{  
 "appointmentDate": "2025-07-23",  
 "appointmentTime": "10:30:00",  
 "status": "Rescheduled",  
 "patientNotes": "Update - sakit kepala makin parah."  
}

## POST Review

POST /api/reviews  
Content-Type: application/json  
  
{  
 "doctorId": 2,  
 "userId": 1,  
 "appointmentId": 3,  
 "rating": 5,  
 "comment": "Dokternya sangat ramah dan profesional."  
}