

# **Dokumentasi Fitness Booking API**

## **II3160 Teknologi Sistem Terintegrasi**

Layanan untuk Menghubungkan Client dengan Personal Trainer,  
Memfasilitasi dan Memanajemen Sesi Latihan



Disusun oleh :

Khairunnisa Azizah      18223117

**PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
JATINANGOR  
2025**

Modul M04 pada proyek ini merealisasikan sebagian besar Booking Context yang telah dirancang pada modul konseptual dan taktis (M02 dan M03). Implementasi API pada modul ini mencakup beberapa kapabilitas inti, yaitu Client Management, Trainer Management, dan Booking Management yang meliputi pembuatan booking, pengambilan data booking, serta perubahan status booking seperti *confirm*, *complete*, *reject*, dan *cancel*. Selain itu, modul ini juga menyediakan endpoint untuk melakukan query booking berdasarkan Client ID, sesuai kebutuhan fungsional yang telah didefinisikan sebelumnya.

Dalam cakupan modul ini, API hanya mengimplementasikan bagian domain yang berkaitan langsung dengan proses pemesanan sesi latihan antara client dan trainer. Beberapa kapabilitas dari domain yang lebih luas termasuk Payment, progress latihan, serta Authentication dan Authorization belum diimplementasikan pada modul M04. Fitur-fitur tersebut direncanakan sebagai pekerjaan lanjutan pada modul berikutnya sehingga tidak tercakup dalam dokumentasi dan implementasi API ini.

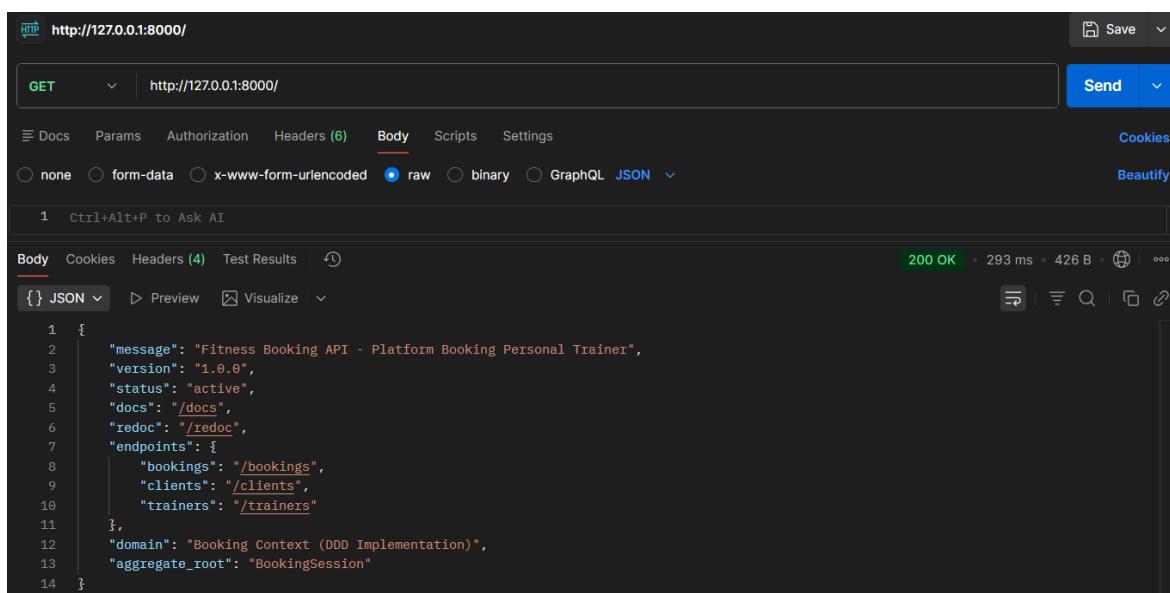
Dengan ruang lingkup tersebut, modul M04 berfungsi sebagai realisasi operasional dari desain domain Booking Context, menyediakan dasar sistem yang utuh untuk pengelolaan client, trainer, dan proses booking, sementara fitur pendukung lainnya akan dikembangkan secara bertahap pada fase modul berikutnya.

## I. Link Repo Github

<https://github.com/kyyzaa/fitness-booking-api.git>

## II. Dokumentasi API

### A. Root & Health Check

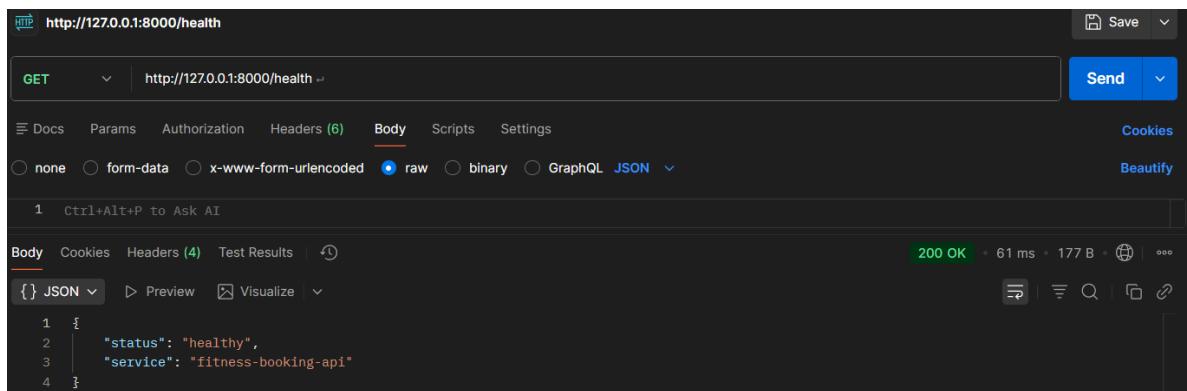


The screenshot shows a browser-based API documentation interface. The URL in the address bar is `http://127.0.0.1:8000/`. The request method is set to `GET`. The Headers tab shows `Content-Type: application/json`. The Body tab is selected, showing a JSON response. The response body is:

```
1  {
2      "message": "Fitness Booking API - Platform Booking Personal Trainer",
3      "version": "1.0.0",
4      "status": "active",
5      "docs": "/docs",
6      "redoc": "/redoc",
7      "endpoints": {
8          "bookings": "/bookings",
9          "clients": "/clients",
10         "trainers": "/trainers"
11     },
12     "domain": "Booking Context (DDD Implementation)",
13     "aggregate_root": "BookingSession"
14 }
```

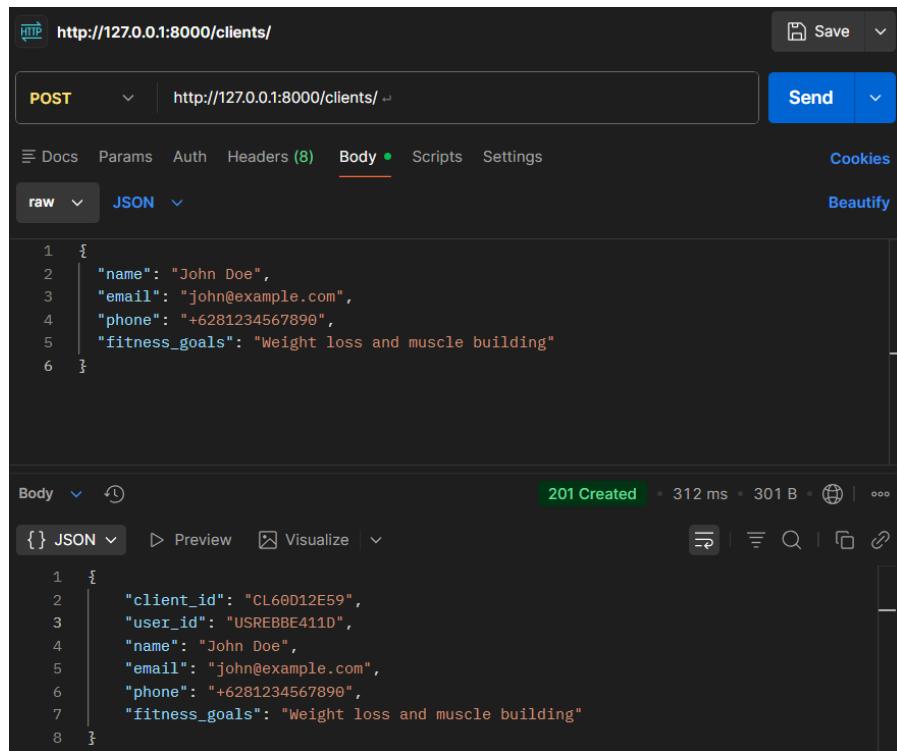
The status bar at the bottom right indicates `200 OK`, `293 ms`, and `426 B`.

Gambar 1. Root API Information



Gambar 2. Health Check Endpoint

## B. Client Management



Gambar 3. Create Client - John Doe

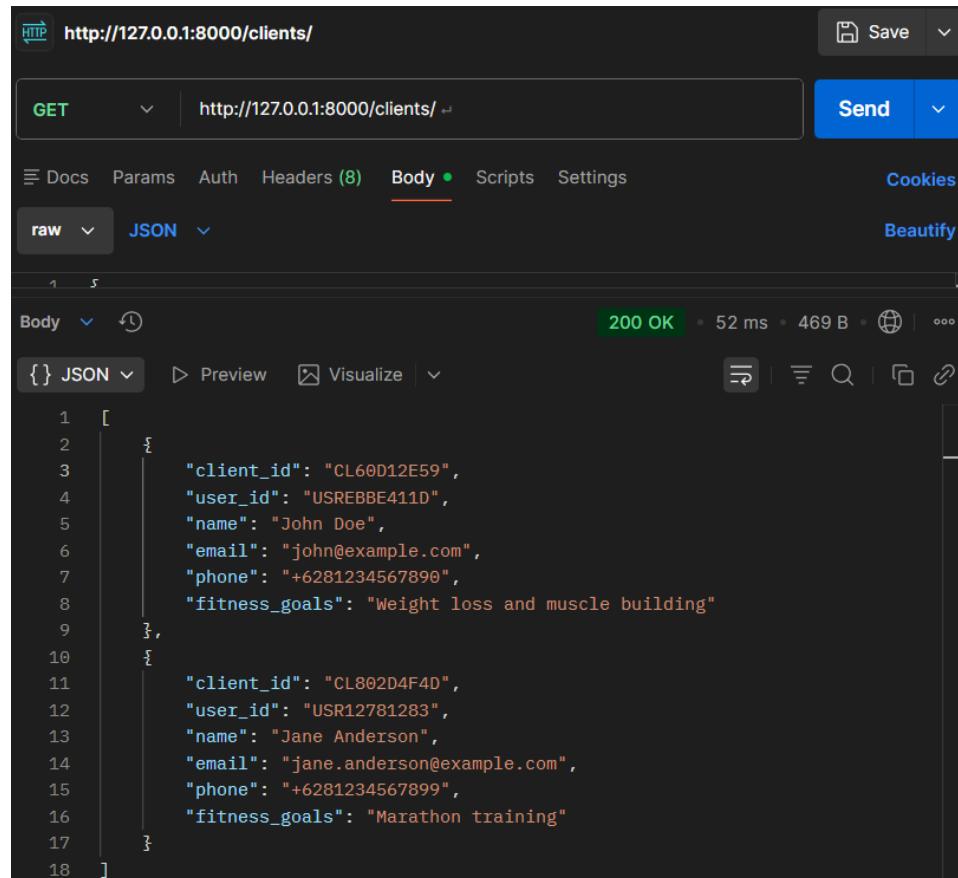
The screenshot shows a POST request to the endpoint `http://127.0.0.1:8000/clients/`. The request body is a JSON object with the following fields:

```
1 {  
2   "name": "Jane Anderson",  
3   "email": "jane.anderson@example.com",  
4   "phone": "+6281234567899",  
5   "fitness_goals": "Marathon training"  
6 }
```

The response status is `201 Created`, with a response time of `18 ms` and a response size of `301 B`. The returned JSON object includes additional fields: `client_id` and `user_id`.

```
1 {  
2   "client_id": "CL802D4F4D",  
3   "user_id": "USR12781283",  
4   "name": "Jane Anderson",  
5   "email": "jane.anderson@example.com",  
6   "phone": "+6281234567899",  
7   "fitness_goals": "Marathon training"  
8 }
```

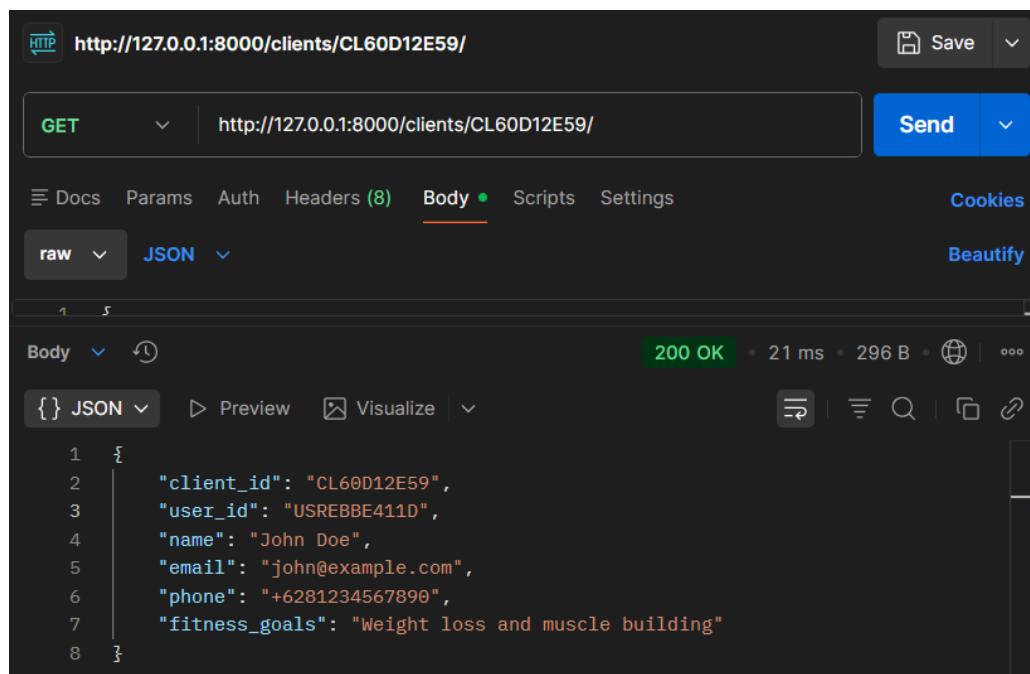
Gambar 4. Create Client - Jane Anderson



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8000/clients/`. The response status is `200 OK` with a duration of `52 ms` and a size of `469 B`. The response body is displayed in JSON format, showing two client records:

```
[{"client_id": "CL60D12E59", "user_id": "USRBBE411D", "name": "John Doe", "email": "john@example.com", "phone": "+6281234567890", "fitness_goals": "Weight loss and muscle building"}, {"client_id": "CL802D4F4D", "user_id": "USR12781283", "name": "Jane Anderson", "email": "jane.anderson@example.com", "phone": "+6281234567899", "fitness_goals": "Marathon training"}]
```

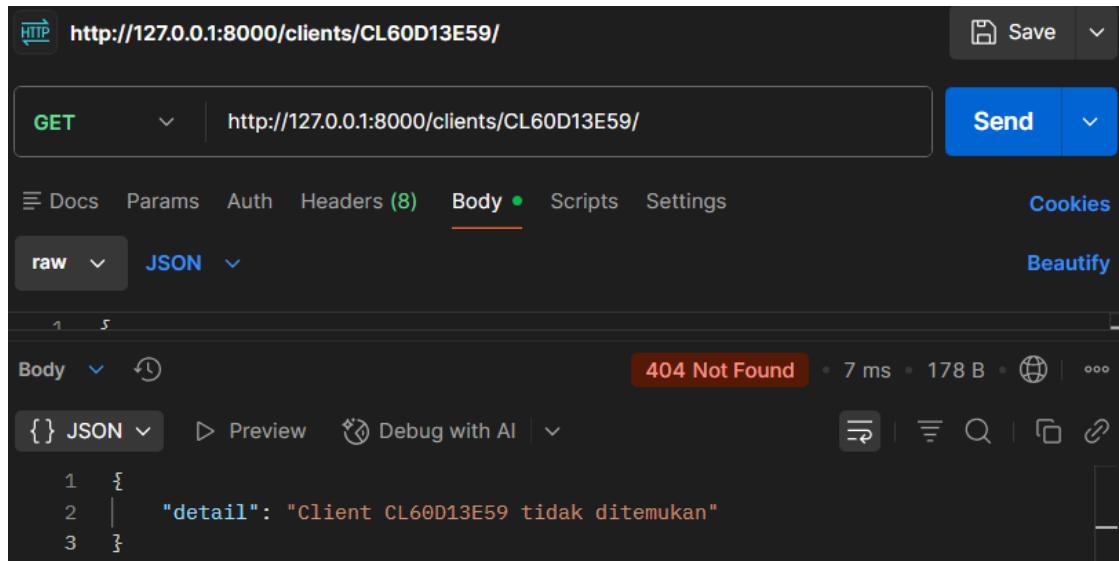
Gambar 5. Get All Clients



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8000/clients/CL60D12E59/`. The response status is `200 OK` with a duration of `21 ms` and a size of `296 B`. The response body is displayed in JSON format, showing one client record:

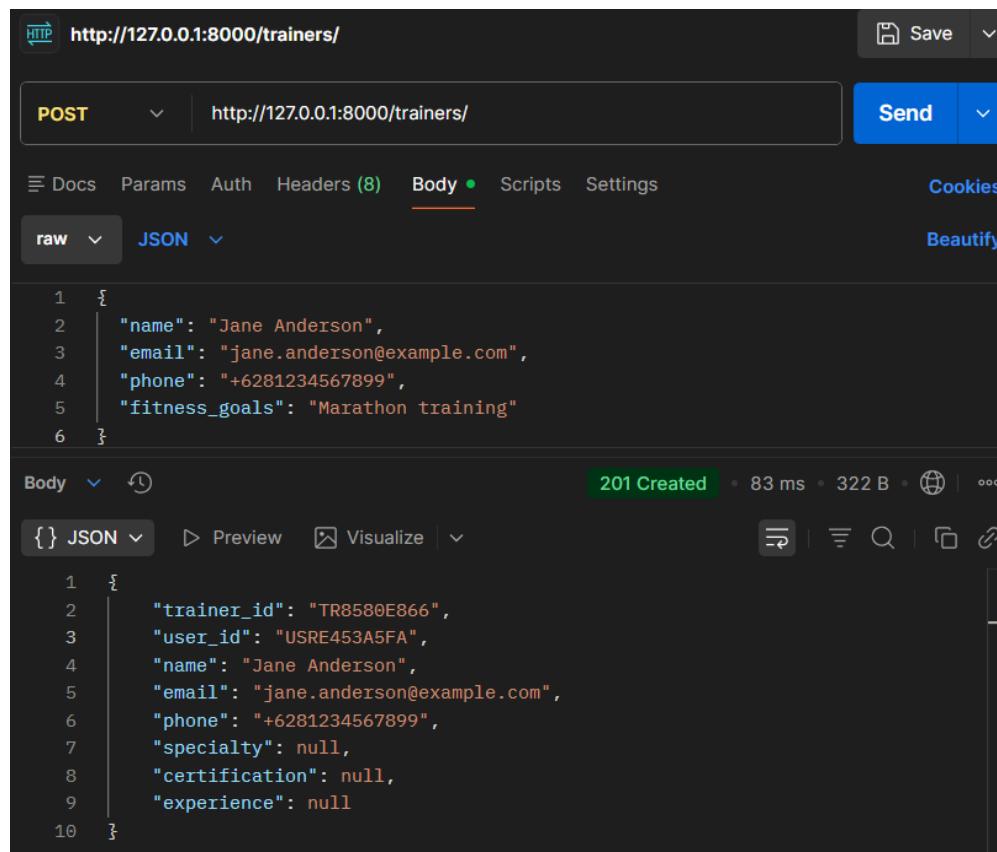
```
{"client_id": "CL60D12E59", "user_id": "USRBBE411D", "name": "John Doe", "email": "john@example.com", "phone": "+6281234567890", "fitness_goals": "Weight loss and muscle building"}
```

Gambar 6. Get Client By Valid Id



Gambar 7. Get Client By Invalid Id

## C. Trainer Management



Gambar 8. Create Trainer - Jane Anderson

The screenshot shows the Postman application interface. At the top, the URL is set to `http://127.0.0.1:8000/trainers/`. Below it, a `POST` request is selected. The `Body` tab is active, showing a JSON payload:

```
1 {
2     "name": "Sarah Williams",
3     "email": "sarah@example.com",
4     "phone": "+6281234567892",
5     "specialty": "Yoga, Pilates, Flexibility",
6     "certification": "RYT-500, NASM",
7     "experience": 8
8 }
```

The response status is `201 Created`, with a duration of `15 ms` and a size of `347 B`. The response body is also displayed in JSON format:

```
1 {
2     "trainer_id": "TR24A4CA2D",
3     "user_id": "USR8092097A",
4     "name": "Sarah Williams",
5     "email": "sarah@example.com",
6     "phone": "+6281234567892",
7     "specialty": "Yoga, Pilates, Flexibility",
8     "certification": "RYT-500, NASM",
9     "experience": 8
10 }
```

Gambar 9. Create Trainer - Sarah Williams

The screenshot shows the Postman application interface. At the top, the URL is set to `http://127.0.0.1:8000/trainers/`. Below it, a `GET` request is selected. The `Body` tab is active, showing a JSON response:

```
1 [
2     {
3         "trainer_id": "TR8580E866",
4         "user_id": "USR453A5FA",
5         "name": "Jane Anderson",
6         "email": "jane.anderson@example.com",
7         "phone": "+6281234567899",
8         "specialty": null,
9         "certification": null,
10        "experience": null
11    },
12    {
13        "trainer_id": "TR24A4CA2D",
14        "user_id": "USR8092097A",
15        "name": "Sarah Williams",
16        "email": "sarah@example.com",
17        "phone": "+6281234567892",
18        "specialty": "Yoga, Pilates, Flexibility",
19        "certification": "RYT-500, NASM",
20        "experience": 8
21    }
22 ]
```

Gambar 10. Get All Trainer

The screenshot shows a Postman interface with the following details:

- Method: GET
- URL: `http://127.0.0.1:8000/trainers/TR8580E866/`
- Status: 200 OK
- Body (JSON):

```
1 {  
2     "trainer_id": "TR8580E866",  
3     "user_id": "USRE453A5FA",  
4     "name": "Jane Anderson",  
5     "email": "jane.anderson@example.com",  
6     "phone": "+6281234567899",  
7     "specialty": null,  
8     "certification": null,  
9     "experience": null  
10 }
```

Gambar 11. Get Trainer By Valid Id

The screenshot shows a Postman interface with the following details:

- Method: GET
- URL: `http://127.0.0.1:8000/trainers/TR8580E966/`
- Status: 404 Not Found
- Body (JSON):

```
1 {  
2     "detail": "Trainer TR8580E966 tidak ditemukan"  
3 }
```

Gambar 12. Get Trainer By Invalid Id

## D. Booking Management

The screenshot shows a POST request to `http://127.0.0.1:8000/bookings/`. The request body is a JSON object representing a booking:

```
1 {  
2     "client_id": "CL60012E59",  
3     "trainer_id": "TR8580E866",  
4     "time_slot": {  
5         "date": "2025-11-25",  
6         "start_time": "09:00:00",  
7         "end_time": "10:00:00"  
8     },  
9     "duration_minutes": 60  
10}
```

The response status is `201 Created`, with a response time of 51 ms and a response size of 423 B. The response body is also a JSON object:

```
1 {  
2     "booking_id": "BK8800E555",  
3     "client_id": "CL60012E59",  
4     "trainer_id": "TR8580E866",  
5     "time_slot": {  
6         "date": "2025-11-25",  
7         "start_time": "09:00:00",  
8         "end_time": "10:00:00"  
9     },  
10    "status": "PENDING",  
11    "duration_minutes": 60,  
12    "booking_date": "2025-11-17T22:46:15.657220",  
13    "confirmed_date": null,  
14    "cancellation_reason": null  
15 }
```

Gambar 13. Create Booking (Pending Status)

The screenshot shows a GET request to `http://127.0.0.1:8000/bookings/`. The response status is `200 OK`, with a response time of 10 ms and a response size of 420 B. The response body is a JSON array containing one booking object:

```
1 [  
2     {  
3         "booking_id": "BK8800E555",  
4         "client_id": "CL60012E59",  
5         "trainer_id": "TR8580E866",  
6         "time_slot": {  
7             "date": "2025-11-25",  
8             "start_time": "09:00:00",  
9             "end_time": "10:00:00"  
10        },  
11        "status": "PENDING",  
12        "duration_minutes": 60,  
13        "booking_date": "2025-11-17T22:46:15.657220",  
14        "confirmed_date": null,  
15        "cancellation_reason": null  
16    }  
17 ]
```

Gambar 14. Get All Booking

The screenshot shows a Postman interface with the following details:

- HTTP Method: GET
- URL: <http://127.0.0.1:8000/bookings/BK8800E555/>
- Status: 200 OK
- Body (JSON):

```
1 {  
2   "booking_id": "BK8800E555",  
3   "client_id": "CL60D12E59",  
4   "trainer_id": "TR8580E866",  
5   "time_slot": {  
6     "date": "2025-11-25",  
7     "start_time": "09:00:00",  
8     "end_time": "10:00:00"  
9   },  
10  "status": "PENDING",  
11  "duration_minutes": 60,  
12  "booking_date": "2025-11-17T22:46:15.657220",  
13  "confirmed_date": null,  
14  "cancellation_reason": null  
15 }
```

Gambar 15. Get Booking By Id

The screenshot shows a Postman interface with the following details:

- HTTP Method: POST
- URL: <http://127.0.0.1:8000/bookings/BK8800E555/confirm>
- Status: 200 OK
- Body (JSON):

```
1 {  
2   "trainer_id": "TR8580E866"  
3 }
```
- Body (JSON):

```
1 {  
2   "booking_id": "BK8800E555",  
3   "client_id": "CL60D12E59",  
4   "trainer_id": "TR8580E866",  
5   "time_slot": {  
6     "date": "2025-11-25",  
7     "start_time": "09:00:00",  
8     "end_time": "10:00:00"  
9   },  
10  "status": "CONFIRMED",  
11  "duration_minutes": 60,  
12  "booking_date": "2025-11-17T22:46:15.657220",  
13  "confirmed_date": "2025-11-17T22:49:31.949658",  
14  "cancellation_reason": null  
15 }
```

Gambar 16. Confirm Booking

The screenshot shows a Postman interface with the following details:

- URL:** `http://127.0.0.1:8000/bookings/BK8800E555/complete`
- Method:** POST
- Body (JSON):**

```
1 {
2   "trainer_id": "TR8580E866"
3 }
```
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {
2   "booking_id": "BK8800E555",
3   "client_id": "CL60D12E59",
4   "trainer_id": "TR8580E866",
5   "time_slot": {
6     "date": "2025-11-25",
7     "start_time": "09:00:00",
8     "end_time": "10:00:00"
9   },
10  "status": "COMPLETED",
11  "duration_minutes": 60,
12  "booking_date": "2025-11-17T22:46:15.657220",
13  "confirmed_date": "2025-11-17T22:49:31.949658",
14  "cancellation_reason": null
15 }
```

Gambar 17. Complete Booking

The screenshot shows a Postman interface with the following details:

- URL:** `http://127.0.0.1:8000/bookings/BK1B09A6C6/reject/`
- Method:** POST
- Body (JSON):**

```
1 {
2   "trainer_id": "TR8580E866",
3   "reason": "Jadwal bentrok dengan sesi lain"
4 }
```
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {
2   "booking_id": "BK1B09A6C6",
3   "client_id": "CL802D4F4D",
4   "trainer_id": "TR8580E866",
5   "time_slot": {
6     "date": "2025-11-26",
7     "start_time": "14:00:00",
8     "end_time": "15:30:00"
9   },
10  "status": "CANCELLED",
11  "duration_minutes": 90,
12  "booking_date": "2025-11-17T22:53:25.500934",
13  "confirmed_date": null,
14  "cancellation_reason": "Ditolak oleh trainer: Jadwal bentrok dengan sesi lain"
15 }
```

Gambar 18. Get Client By Invalid Id

```

1 {
2   "client_id": "CL60D12E59",
3   "trainer_id": "TR8580E866",
4   "time_slot": {
5     "date": "2025-11-27",
6     "start_time": "10:00:00",
7     "end_time": "11:00:00"
8   },
9   "duration_minutes": 60
10 }

```

Body 🕒 201 Created 6 ms 423 B 🌐 ...

```

1 {
2   "booking_id": "BK4493103A",
3   "client_id": "CL60D12E59",
4   "trainer_id": "TR8580E866",
5   "time_slot": {
6     "date": "2025-11-27",
7     "start_time": "10:00:00",
8     "end_time": "11:00:00"
9   },
10  "status": "PENDING",
11  "duration_minutes": 60,
12  "booking_date": "2025-11-17T22:57:29.782827",
13  "confirmed_date": null,
14  "cancellation_reason": null
15 }

```

Gambar 19. Reject Booking By Trainer

```

1 {
2   "user_id": "USREBBE411D",
3   "reason": "client berhalangan hadir karena sakit"
4 }

```

Body 🕒 200 OK 8 ms 455 B 🌐 ...

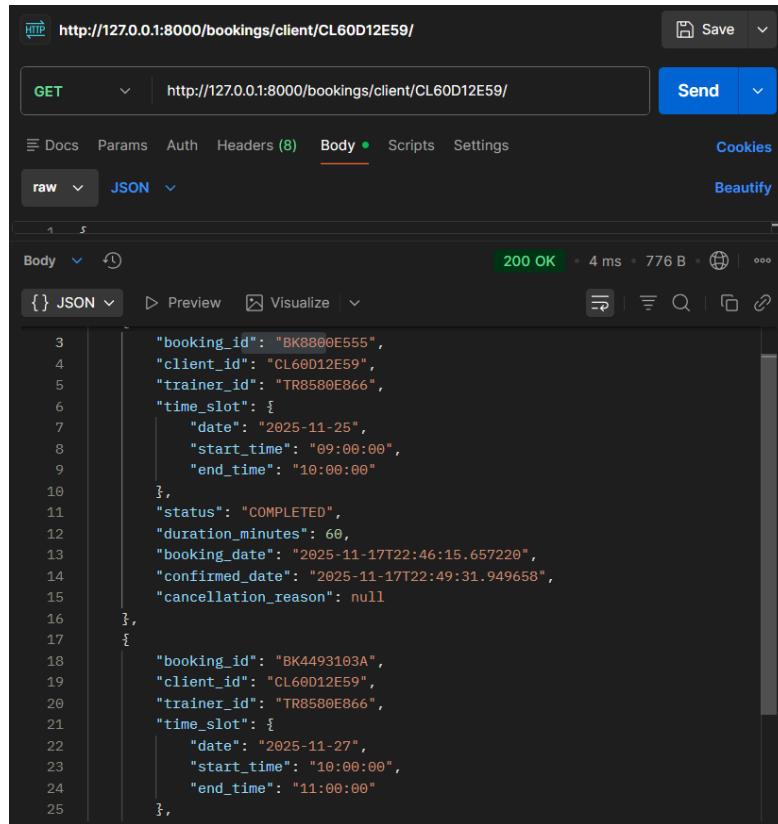
```

1 {
2   "booking_id": "BK4493103A",
3   "client_id": "CL60D12E59",
4   "trainer_id": "TR8580E866",
5   "time_slot": {
6     "date": "2025-11-27",
7     "start_time": "10:00:00",
8     "end_time": "11:00:00"
9   },
10  "status": "CANCELLED",
11  "duration_minutes": 60,
12  "booking_date": "2025-11-17T22:57:29.782827",
13  "confirmed_date": null,
14  "cancellation_reason": "Client berhalangan hadir karena sakit"
15 }

```

Gambar 20. Cancel Booking

## E. Query By User



The screenshot shows a Postman interface with the following details:

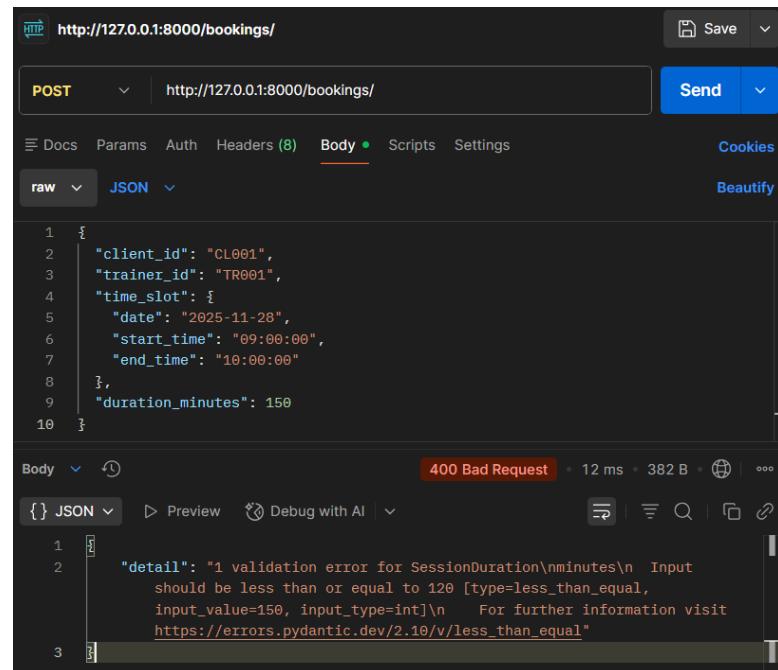
- Method: GET
- URL: <http://127.0.0.1:8000/bookings/client/CL60D12E59/>
- Body tab is selected.
- Response status: 200 OK
- Response time: 4 ms
- Response size: 776 B
- Response content (JSON):

```
3   "booking_id": "BK880E555",
4   "client_id": "CL60D12E59",
5   "trainer_id": "TR8580E866",
6   "time_slot": {
7       "date": "2025-11-25",
8       "start_time": "09:00:00",
9       "end_time": "10:00:00"
10  },
11  "status": "COMPLETED",
12  "duration_minutes": 60,
13  "booking_date": "2025-11-17T22:46:15.657220",
14  "confirmed_date": "2025-11-17T22:49:31.949658",
15  "cancellation_reason": null
16  },
17  {
18      "booking_id": "BK4493103A",
19      "client_id": "CL60D12E59",
20      "trainer_id": "TR8580E866",
21      "time_slot": {
22          "date": "2025-11-27",
23          "start_time": "10:00:00",
24          "end_time": "11:00:00"
25      },

```

Gambar 21. Get Booking By Client Id

## F. Error Scenario



The screenshot shows a Postman interface with the following details:

- Method: POST
- URL: <http://127.0.0.1:8000/bookings/>
- Body tab is selected.
- Response status: 400 Bad Request
- Response time: 12 ms
- Response size: 382 B
- Response content (JSON):

```
1  {
2     "client_id": "CL001",
3     "trainer_id": "TR001",
4     "time_slot": [
5         {
6             "date": "2025-11-28",
7             "start_time": "09:00:00",
8             "end_time": "10:00:00"
9         },
10        "duration_minutes": 150
11    }

```

Detail message:

```
1  "detail": "1 validation error for SessionDuration\nminutes\n  Input
2  should be less than or equal to 120 [type=less_than_equal,
3  input_value=150, input_type=int]\n    For further information visit
4  https://errors.pydantic.dev/2.10/v/less_than_equal"
```

Gambar 22. Invalid Duration

The screenshot shows a POST request to `http://127.0.0.1:8000/bookings/`. The request body is a JSON object:

```
1 {
2   "client_id": "CL60D22E59",
3   "trainer_id": "TR001",
4   "time_slot": {
5     "date": "2025-11-28",
6     "start_time": "09:00:00",
7     "end_time": "10:00:00"
8   },
9   "duration_minutes": 60
10 }
```

The response status is **400 Bad Request**, with a duration of 21 ms and a response size of 190 B. The response body is:

```
{}
{"detail": "Client dengan ID CL60D22E59 tidak ditemukan"}
```

Gambar 23. Invalid Client Id

The screenshot shows a GET request to `http://127.0.0.1:8000/bookings/BK001`. The response status is **404 Not Found**, with a duration of 5 ms and a response size of 174 B. The response body is:

```
{}
{"detail": "Booking BK001 tidak ditemukan"}
```

Gambar 24. Invalid Booking Id

HTTP <http://localhost:8000/auth/register> Save ▾

POST http://localhost:8000/auth/register Send ▾

Docs Params Auth Headers (8) Body Scripts Settings Cookies

raw ▾ JSON ▾ Beautify

```
1 {  
2   "email": "newuser@example.com",  
3   "password": "pass123",  
4   "role": "CLIENT"  
5 }
```

Body ▾ ⏱ 201 Created • 321 ms • 218 B • 🌐 | ⚙️

{ } JSON ▾ ▷ Preview ⚒ Visualize ▾

≡ ⚒ 🔍 ⚒ ⚒ ⚒

```
1 {  
2   "user_id": "USRACA3095E",  
3   "email": "newuser@example.com",  
4   "role": "CLIENT",  
5   "is_active": true  
6 }
```

HTTP <http://localhost:8000/auth/me> Save ▾

GET <http://localhost:8000/auth/me> Send ▾

Docs Params Auth Headers (9) Body Scripts Settings Cookies

Headers (8 hidden)

Key	Value
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1Ni...
Key	Description

Body [JSON](#) ▾ Preview Visualize ▾ 200 OK • 6 ms • 212 B • [Global](#) ▾

{ } JSON ▾ ▷ Preview  Visualize ▾

1 {  
2 "user\_id": "USR2F3B2E9F",  
3 "email": "client@example.com",  
4 "role": "CLIENT",  
5 "is\_active": true  
6 }

The screenshot shows the Postman application interface. At the top, it displays the URL `http://localhost:8000/clients/`. The main area is set up for a **POST** request to the same URL. The **Body** tab is selected, showing a JSON payload:

```
1 {  
2   "name": "John Doe",  
3   "email": "john@example.com",  
4   "phone": "+6281234567890",  
5   "fitness_goals": "Weight loss and muscle building"  
6 }
```

Below the body, the status bar indicates a **201 Created** response with a duration of **6 ms** and a size of **301 B**. The response body is also shown as JSON:

```
{ } JSON ▾ Preview ⚡ Visualize | ↻ 🔍 ⌂ ⌂ ⌂ ⌂
```

```
1 {  
2   "client_id": "CL83D30D4B",  
3   "user_id": "USRE4C8A523",  
4   "name": "John Doe",  
5   "email": "john@example.com",  
6   "phone": "+6281234567890",  
7   "fitness_goals": "Weight loss and muscle building"  
8 }
```

HTTP <http://localhost:8000/clients/>

Save ▾

GET ▾ | http://localhost:8000/clients/ | Send ▾

Docs Params Auth Headers (9) Body ● Scripts Settings Cookies

Headers (8 hidden)

Key	Value	Bulk Edit	Presets ▾
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1Ni...		
Key	Value	Description	

Body ▾ ⏱ 200 OK • 6 ms • 298 B • 🌐 ⋮

{ } JSON ▾ ▷ Preview ⚒ Visualize ▾

1 [  
2 {  
3 "client\_id": "CL83D30D4B",  
4 "user\_id": "USRE4C8A523",  
5 "name": "John Doe",  
6 "email": "john@example.com",  
7 "phone": "+6281234567890",  
8 "fitness\_goals": "Weight loss and muscle building"  
9 }  
10 ]

HTTP <http://localhost:8000/clients/CL83D30D4B> Save ▾

GET <http://localhost:8000/clients/CL83D30D4B> Send ▾

Docs Params Auth Headers (9) Body • Scripts Settings Cookies

Headers (8 hidden)

	Key	Value	...	Bulk Edit	Presets ▾
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1Ni...			
	Key	Value	Description		

Body ▾ ↻ 200 OK • 8 ms • 296 B • 🌐 ⚙️ 🔍 🗂️ 🔍

{ } JSON ▾ ▷ Preview ⚙️ 🔍 🗂️ 🔍

```
1 {  
2   "client_id": "CL83D30D4B",  
3   "user_id": "USRE4C8A523",  
4   "name": "John Doe",  
5   "email": "john@example.com",  
6   "phone": "+6281234567890",  
7   "fitness_goals": "Weight loss and muscle building"  
8 }
```

HTTP <http://localhost:8000/trainers> Save ▾

POST <http://localhost:8000/trainers> Send ▾

Docs Params Auth Headers (9) Body Scripts Settings Cookies

Headers 8 hidden

	Key	Value	...	Bulk Edit	Presets ▾
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1Ni...			
	Key	Value	Description		

Body JSON ▾ Preview Visualize ▾ 201 Created • 5 ms • 308 B • [Global](#) • [Raw](#) [Link](#)

```
1 {  
2   "trainer_id": "TR71B54FC1",  
3   "user_id": "USR8EAB33B3",  
4   "name": "John Doe",  
5   "email": "john@example.com",  
6   "phone": "+6281234567890",  
7   "specialty": null,  
8   "certification": null,  
9   "experience": null  
10 }
```

HTTP <http://localhost:8000/trainers> Save ▾

POST <http://localhost:8000/trainers> Send ▾

Docs Params Auth Headers (9) Body Scripts Settings Cookies

raw ▾ JSON ▾ Beautify

```
1 {  
2   "name": "Jane Smith",  
3   "email": "jane@example.com",  
4   "phone": "+6281234567891",  
5   "specialty": "Strength training, Weight loss",  
6   "certification": "NASM-CPT, ACE",  
7   "experience": 5  
8 }
```

Body ▾ ⏱ 201 Created • 4 ms • 346 B • 🌐 ⚙️ 🔍 ↻ ↺

{ } JSON ▾ ▷ Preview ⚡ Visualize | ▾

201 Created

```
1 {  
2   "trainer_id": "TR80D82EF5",  
3   "user_id": "USR9D264290",  
4   "name": "Jane Smith",  
5   "email": "jane@example.com",  
6   "phone": "+6281234567891",  
7   "specialty": "Strength training, Weight loss",  
8   "certification": "NASM-CPT, ACE",  
9   "experience": 5  
10 }
```

HTTP <http://localhost:8000/trainers> Save ▾

GET <http://localhost:8000/trainers> Send ▾

Docs Params Auth Headers (9) Body ● Scripts Settings Cookies

Headers (8 hidden)

	Key	Value	D	...	Bulk Edit	Presets	▼
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1Ni...					
	Key	Value					Description

Body ▾ ⏱ 200 OK • 6 ms • 699 B • 🌐 • ⚙️

{ } JSON ▾ ▷ Preview ⚒ Visualize ▾

```
1 [  
2 {  
3   "trainer_id": "TR9A4ED6DF",  
4   "user_id": "USR8333BABE",  
5   "name": "John Doe",  
6   "email": "john@example.com",  
7   "phone": "+6281234567890",  
8   "specialty": null,  
9   "certification": null,  
10  "experience": null  
11 },  
12 {
```

The screenshot shows a REST client interface with the following details:

- HTTP Method:** GET
- URL:** http://localhost:8000/trainers/TR9A4ED6DF
- Status:** 200 OK (green box)
- Response Time:** 5 ms
- Response Size:** 303 B
- Content Type:** application/json
- Headers:** (9) (highlighted in green)
- Body:** (highlighted in green)
- Content:** A JSON object representing a trainer record:

```
1 {  
2   "trainer_id": "TR9A4ED6DF",  
3   "user_id": "USR8333BABE",  
4   "name": "John Doe",  
5   "email": "john@example.com",  
6   "phone": "+6281234567890",  
7   "specialty": null,  
8   "certification": null,  
9   "experience": null  
10 }
```

HTTP <http://localhost:8000/trainers> Save ▾

GET <http://localhost:8000/trainers> Send ▾

Docs Params Auth Headers (9) Body ● Scripts Settings Cookies

Headers (8 hidden)

	Key	Value	...	Bulk Edit	Presets	...
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1Ni...				

Body ▾ ⏱ 200 OK • 7 ms • 699 B • 🌐 ...

{ } JSON ▾ ▷ Preview ⚡ Visualize | ▾

1 [  
2 {  
3 "trainer\_id": "TR9A4ED6DF",  
4 "user\_id": "USR8333BABE",  
5 "name": "John Doe",  
6 "email": "john@example.com",  
7 "phone": "+6281234567890",  
8 "specialty": null,

HTTP <http://localhost:8000/trainers/TR9A4ED6DF> Save ▾

GET <http://localhost:8000/trainers/TR9A4ED6DF> Send ▾

Docs Params Auth Headers (9) Body • Scripts Settings Cookies

Headers (8 hidden)

	Key	Value	...	Bulk Edit	Presets	...
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1Ni...				

Body ▾ ⏪ 200 OK • 8 ms • 303 B • 🌐 ...

{ } JSON ▾ ▶ Preview ⚒ Visualize ▾

```
1 {  
2   "trainer_id": "TR9A4ED6DF",  
3   "user_id": "USR8333BABE",  
4   "name": "John Doe",  
5   "email": "john@example.com",  
6   "phone": "+6281234567890",  
7   "specialty": null,  
8   "certification": null,  
9   "experience": null  
10 }
```

HTTP <http://localhost:8000/bookings/>

POST http://localhost:8000/bookings/ Send

Docs Params Auth Headers (9) Body Scripts Settings Cookies Beautify

raw JSON

```
1 {  
2   "client_id": "CL83D30D4B",  
3   "trainer_id": "TR71B54FC1".
```

Body 201 Created 12 ms 423 B

{ } JSON Preview Visualize

```
1 {  
2   "booking_id": "BK472B59D6",  
3   "client_id": "CL83D30D4B",  
4   "trainer_id": "TR71B54FC1",  
5   "time_slot": {  
6     "date": "2025-12-05",  
7     "start_time": "09:00:00",  
8     "end_time": "10:00:00"  
9   },  
10  "status": "PENDING",  
11  "duration_minutes": 60,  
12  "booking_date": "2025-12-01T20:37:07.047730",  
13  "confirmed_date": null,  
14  "cancellation_reason": null  
15 }
```

The screenshot shows a browser-based API testing interface. At the top, there's a header bar with 'HTTP' and 'http://localhost:8000/bookings/'. On the right side of the header are 'Save' and a dropdown menu. Below the header is a search bar with 'GET' and 'http://localhost:8000/bookings/'. To the right of the search bar is a large blue 'Send' button. The main content area has tabs for 'Docs', 'Params', 'Auth', 'Headers (9)', 'Body' (which is currently selected), 'Scripts', and 'Settings'. Below these tabs are two buttons: 'raw' and 'JSON'. To the right of the JSON button is a 'Beautify' link. The 'Body' section displays a JSON response with line numbers from 1 to 17. The response is as follows:

```
1  [
2    {
3      "booking_id": "BK472B59D6",
4      "client_id": "CL83D30D4B",
5      "trainer_id": "TR71B54FC1",
6      "time_slot": {
7        "date": "2025-12-05",
8        "start_time": "09:00:00",
9        "end_time": "10:00:00"
10      },
11      "status": "PENDING",
12      "duration_minutes": 60,
13      "booking_date": "2025-12-01T20:37:07.047730",
14      "confirmed_date": null,
15      "cancellation_reason": null
16    }
17  ]
```

On the right side of the body area, there's a status bar showing '200 OK', '8 ms', '420 B', and a globe icon. Below the status bar are several icons: a copy icon, a refresh icon, a search icon, a refresh icon, and a link icon.

HTTP <http://localhost:8000/bookings/BK472B59D6> Save ▾

GET [Raw](#) [JSON](#) [Beautify](#)

Docs Params Auth Headers (9) Body Scripts Settings Cookies

1 {  
2 | "client\_id": "CL83D30D4B"  
3 | }  
Body [Raw](#) [JSON](#) 200 OK • 10 ms • 418 B • [Global](#) [...](#)

{ } JSON ▾ [Preview](#) [Visualize](#)

```
1 {  
2   "booking_id": "BK472B59D6",  
3   "client_id": "CL83D30D4B",  
4   "trainer_id": "TR71B54FC1",  
5   "time_slot": {  
6     "date": "2025-12-05",  
7     "start_time": "09:00:00",  
8     "end_time": "10:00:00"  
9   },  
10  "status": "PENDING",  
11  "duration_minutes": 60,  
12  "booking_date": "2025-12-01T20:37:07.047730",  
13  "confirmed_date": null,  
14  "cancellation_reason": null  
15 }
```

HTTP <http://localhost:8000/bookings/BK472B59D6/confirm> Save ▾

POST [http://localhost:8000/bookings/BK472B59D6/confirm](#) Send ▾

Docs Params Auth Headers (9) Body • Scripts Settings Cookies

raw JSON Beautify

```
1 {  
2   "client_id": "CL83D30D4B"  
3 }  
  
Body ▼ 🕒 200 OK • 13 ms • 444 B • 🌐 ...  
  
{} JSON ▼ ▶ Preview ☒ Visualize | ▾ 🔗 ☰ 🔍 📄 🔗
```

```
1 {  
2   "booking_id": "BK472B59D6",  
3   "client_id": "CL83D30D4B",  
4   "trainer_id": "TR71B54FC1",  
5   "time_slot": {  
6     "date": "2025-12-05",  
7     "start_time": "09:00:00",  
8     "end_time": "10:00:00"  
9   },  
10  "status": "CONFIRMED",  
11  "duration_minutes": 60,  
12  "booking_date": "2025-12-01T20:37:07.047730",  
13  "confirmed_date": "2025-12-01T20:41:29.716803",  
14  "cancellation_reason": null  
15 }
```

HTTP <http://localhost:8000/bookings/client/CL83D30D4B> Save ▾

GET <http://localhost:8000/bookings/client/CL83D30D4B> Send ▾

Docs Params Auth Headers (9) Body ● Scripts Settings Cookies

raw ▾ JSON ▾ Beautify

Body ▾ ⟳ 200 OK • 8 ms • 446 B • 🌐 ⚡ ⚡ 🔍 🗂️ 🔗

{ } JSON ▾ Preview Visualize ▾

```
1 [  
2 {  
3   "booking_id": "BK472B59D6",  
4   "client_id": "CL83D30D4B",  
5   "trainer_id": "TR71B54FC1",  
6   "time_slot": {  
7     "date": "2025-12-05",  
8     "start_time": "09:00:00",  
9     "end_time": "10:00:00"  
10    },  
11   "status": "CONFIRMED",  
12   "duration_minutes": 60,  
13   "booking_date": "2025-12-01T20:37:07.047730",  
14   "confirmed_date": "2025-12-01T20:41:29.716803",  
15   "cancellation_reason": null  
16 }  
17 ]
```

HTTP <http://localhost:8000/auth/register> Save ▾

POST <http://localhost:8000/auth/register> Send ▾

Docs Params Auth Headers (8) **Body** ● Scripts Settings Cookies

raw ▾ JSON ▾ Beautify

```
1 {  
2   "email": "trainer@example.com",  
3   "password": "password123",  
4   "role": "TRAINER"  
5 }
```

Body ▾ ⏱ 201 Created • 261 ms • 219 B • 🌐 ⚙️ 🔍 ⌂ ⌂

{ } JSON ▾ ▷ Preview ⚒ Visualize ▾

```
1 {  
2   "user_id": "USR9049F9D6",  
3   "email": "trainer@example.com",  
4   "role": "TRAINER",  
5   "is_active": true  
6 }
```

HTTP <http://localhost:8000/auth/login> Save ▾

POST <http://localhost:8000/auth/login> Send ▾

Docs Params Auth Headers (8) **Body** • Scripts Settings Cookies

x-www-form-urlencoded ▾

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	username	trainer@example.com			
<input checked="" type="checkbox"/>	password	password123			
	Key	Value	Description		

Body [JSON](#) ▾ [Preview](#) [Visualize](#) ▾ 200 OK • 301 ms • 433 B • [🌐](#) • ...

{ } JSON ▾ [≡](#) [☰](#) [🔍](#) [🔗](#)

```
1 {  
2   "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
3     eyJzdWIiOiJVU1I2NkUwOEZFMyIsImVtYWlsIjoidHJhaW5lckBleGFtcGx1LmNv  
4     bSIisInJvbGUiOiJUUkFJTkVSIiwizXhwIjoxNzY0NTk2MDg5fQ.  
5     0-Z7Udc8V6AAjScz2j8An0hWhwEkGe0blEyu70bTbu8",  
6   "token_type": "bearer",  
7   "user_id": "USR66E08FE3",  
8   "email": "trainer@example.com",  
9   "role": "TRAINER"  
10 }
```

HTTP <http://localhost:8000/auth/me> Save ▾

GET <http://localhost:8000/auth/me> Send ▾

Docs Params Auth Headers (9) Body • Scripts Settings Cookies

Headers 8 hidden

Key	Value	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1Ni...			
Key	Value	Description		

Body 11 ms 214 B

200 OK 11 ms 214 B

{ } JSON ▾ Preview Visualize ▾

1 {  
2 "user\_id": "USR66E08FE3",  
3 "email": "trainer@example.com",  
4 "role": "TRAINER",  
5 "is\_active": true  
6 }