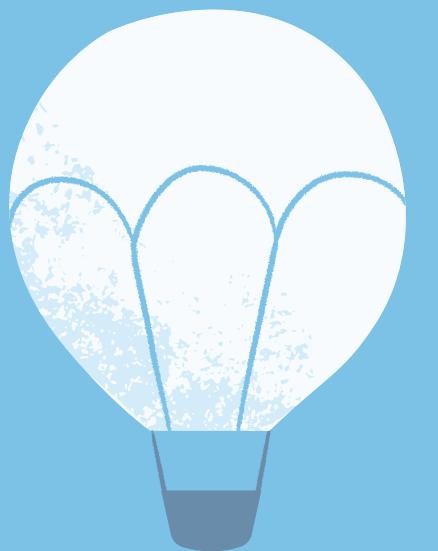


MindCare

APP

18222030

Vini Putiasa



MindCare

MindCare API adalah sebuah layanan berbasis RESTful yang dirancang untuk mendukung pengembangan aplikasi kesehatan mental. API ini memungkinkan integrasi berbagai fitur seperti reservasi online yang mudah, pembatalan reservasi, melihat hasil reservasi, melihat daftar jadwal yang available termasuk psikolog dna waktunya, serta menampilkan artikel yang dapat dibaca. MindCare API bertujuan membantu pengguna menjaga kesehatan mental mereka dengan memberikan dukungan berbasis teknologi.

MindCare

Selamat Datang di MindCare

Haloo!!

Selamat datang di layanan konsultasi online kami. Silakan pilih salah satu menu di bawah ini:

[Jadwal Tersedia](#)

Temukan jadwal konsultasi available
yang sesuai dengan kebutuhan Anda.

[Konsultasi yang Dipesan](#)

Periksa jadwal konsultasi yang telah
Anda pesan.

[Artikel](#)

Baca artikel dan tips tentang kesehatan
mental.

[Logout](#)

MindCare

Jadwal Konsultasi Psikolog

Jadwal Konsultasi yang Tersedia

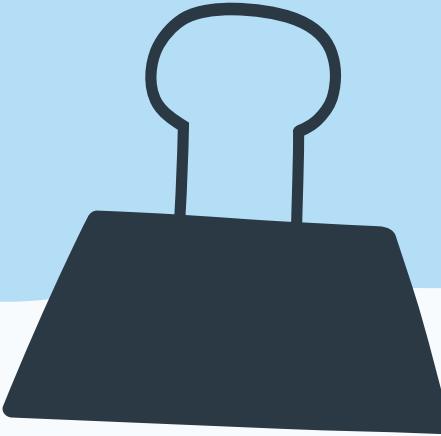
ID Jadwal	Tanggal	Waktu	Nama Psikolog	Aksi
1	2024-12-01	11:00:00	Dr. abc	Pesan

Daftar Jadwal Anda

Belum ada jadwal yang dipesan.

Cara Penggunaan

Register



Register digunakan untuk mendaftarkan data user ke database. Register dilakukan dengan menggunakan metode “post”.

Request:

- URL:
<https://asa99.my.id/auth/register>
- Method: POST
- Headers:
 - Content-Type: application/json

Request Body:

```
{  
  "username": ,  
  "email": ,  
  "password":  
}
```

Register

Register Berhasil

POST <https://asa99.my.id/auth/register> Send

Params Authorization Headers (9) **Body** • Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL [JSON](#) [Beautify](#)

```
1 {  
2   "username": "superman",  
3   "email": "superman@example.com",  
4   "password": "superman123"  
5 }
```

Body Cookies Headers (10) Test Results [↻](#) 201 Created • 453 ms • 520 B • [🔗](#)

{ } JSON [Preview](#) [Visualize](#) [🔗](#) [◻](#) [🔍](#)

```
1 {  
2   "message": "User registered successfully"  
3 }
```

Register

**Register tidak dilanjutkan karena
email sudah pernah terdaftar**

POST https://asa99.my.id/auth/register

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "username": "superman",  
3   "email": "superman@example.com",  
4   "password": "superman123"  
5 }
```

Body Cookies Headers (10) Test Results

400 Bad Request • 362 ms • 545 B •

{ } JSON ▶ Preview ⚡ Visualize | ▾

```
1 {  
2   "status": 400,  
3   "error": 400,  
4   "messages": {  
5     "email": "Email sudah terdaftar."  
6   }  
7 }
```

**Register tidak dilanjutkan karena
email tidak valid**

POST https://asa99.my.id/auth/register

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "username": "superman",  
3   "email": "superman@",  
4   "password": "superman123"  
5 }
```

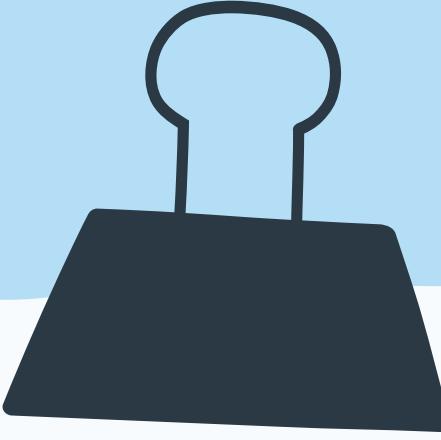
Body Cookies Headers (10) Test Results

400 Bad Request • 317 ms • 568 B •

{ } JSON ▶ Preview ⚡ Visualize | ▾

```
1 {  
2   "status": 400,  
3   "error": 400,  
4   "messages": {  
5     "email": "The email field must contain a valid email address."  
6   }  
7 }
```

Login



Login dilakukan agar user mengetahui id user sehingga dapat menggunakan layanan yang tersedia.

Login dilakukan dengan menggunakan metode “post”.

Request:

- URL:
<https://asa99.my.id/auth/login>
- Method: POST
- Headers:
 - Content-Type: application/json

Request Body:

```
{  
  "email": ,  
  "password":  
}
```

Login

Login Berhasil

POST https://asa99.my.id/auth/login

Body (raw)

```
1 {  
2   "email": "superman@example.com",  
3   "password": "superman123"  
4 }
```

200 OK • 395 ms • 675 B

Body (JSON)

```
1 {  
2   "message": "Login berhasil",  
3   "user": {  
4     "id": "20",  
5     "username": "superman",  
6     "email": "superman@example.com",  
7     "password": "$2y$10$pb90noHrmSST7GCinYwgQ09xU2bGuswvNYxtHJvc6N111HErrAbHy",  
8     "reset_token": null,  
9     "created_at": "2025-01-09 19:42:41",  
10    "updated_at": "2025-01-09 19:42:41",  
11  }  
12}
```

POST https://asa99.my.id/auth/login

Body (raw)

```
1 {  
2   "username": "superman",  
3   "email": "superman@superr",  
4   "password": "superman123"  
5 }
```

404 Not Found • 122 ms • 373 B

Body (JSON)

```
1 {  
2   "status": 404,  
3   "error": 404,  
4   "messages": {  
5     "error": "User dengan email ini tidak ditemukan"  
6   }  
7 }
```

Login gagal karena email yang tidak ditemukan di database

Available Appointments



Available Appointments digunakan untuk menampilkan jadwal available yang dapat dipesan. Pada available Appointments, kita bisa mendapatkan data waktu dan nama psikolog yang dapat di-booking. Available appointments dilakukan dengan menggunakan metode “get”.

Request:

- URL:
<https://asa99.my.id/appointments/available>
- Method: GET

Parameter yang diperlukan adalah user_id seperti berikut:

Appointments

Menampilkan jadwal
tersedia/available pada postman

GET https://asa99.my.id/appointments/available?user_id=16

Params • Authorization Headers (9) Body • Scripts Settings

Query Params

Key	Value	Description
user_id	16	

Body Cookies Headers (10) Test Results 200 OK

{ } JSON ▾ Preview Visualize ▾

```
1 [  
2 {  
3   "id": "4",  
4   "user_name": null,  
5   "psychologist_name": "Joanna S.Psi",  
6   "schedule_date": "2025-04-03",  
7   "schedule_time": "11:00:00",  
8   "status": "available",  
9   "created_at": "2025-01-10 08:51:26",  
10  "email": "",  
11  "user_id": "0",  
12  "specialist": "klinis"  
13 },  
14 ]
```

Postbot Ctrl Alt P

Book Appointments

Book Appointments digunakan untuk memesan atau reservasi jadwal yang tersedia berdasarkan id appointments yang dapat dilihat pada bagian Available Appointments. Setelah mengirim body request yang mencakup ID pengguna, nama, dan email, jadwal akan dibooking jika tersedia. Book appointments dilakukan dengan menggunakan metode “post”.

Endpoint: POST /book/{id}

Request:

- URL:
[https://asa99.my.id/appointments/book/\(id\)](https://asa99.my.id/appointments/book/(id))
- Method: POST

Contoh body Request

```
{  
  "user_id": 123,  
  "user_name": "John Doe",  
  "email": "john.doe@example.com"  
}
```

Book Appointments

Book Berhasil

POST https://asa99.my.id/appointments/book/1

Params Authorization Headers (9) **Body** Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "user_id": 20,  
3   "user_name": "superman",  
4   "email": "superman@example.com"  
5 }
```

Body Cookies Headers (10) Test Results 201 Created

{ } JSON ▾ ▷ Preview ⚡ Visualize ▾

```
1 {  
2   "message": "Jadwal berhasil dibooking"  
3 }
```

Book Gagal karena jadwal dengan id yang diinginkan sudah dibooking

POST https://asa99.my.id/appointments/book/1

Params Authorization Headers (9) **Body** Scripts Settings

none form-data x-www-form-urlencoded raw binary

```
1 {  
2   "user_id": 12,  
3   "user_name": "superman",  
4   "email": "superman@example.com"  
5 }
```

Body Cookies Headers (10) Test Results

{ } JSON ▾ ▷ Preview ⚡ Visualize ▾

```
1 {  
2   "status": 403,  
3   "error": 403,  
4   "messages": {  
5     "error": "Jadwal sudah dibooking"  
6   }  
7 }
```

Book Appointments

Book gagal karena jadwal dengan id yang diinginkan tidak ditemukan

A screenshot of the Postman application interface. The request method is POST, and the URL is <https://asa99.my.id/appointments/book/3>. The 'Body' tab is selected, showing the raw JSON payload:

```
1 {  
2   "user_id": 12,  
3   "user_name": "superman",  
4   "email": "superman@example.com"  
5 }
```

The response body shows an error message in JSON format:

```
1 {  
2   "status": 404,  
3   "error": 404,  
4   "messages": {  
5     "error": "Jadwal tidak ditemukan"  
6   }  
7 }
```

Book gagal karena tidak mengisi user id pada body request

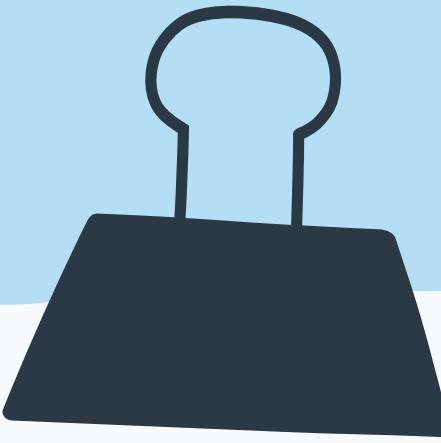
A screenshot of the Postman application interface. The request method is POST, and the URL is <https://asa99.my.id/appointments/book/1>. The 'Body' tab is selected, showing the raw JSON payload:

```
1 {  
2   "user_name": "superman",  
3   "email": "superman@example.com"  
4 }
```

The response body shows an error message in JSON format:

```
1 {  
2   "status": 400,  
3   "error": 400,  
4   "messages": {  
5     "error": "User ID diperlukan"  
6   }  
7 }
```

Booked Appointments



Endpoint: GET /booked/{userId}

Rute ini digunakan untuk menampilkan semua jadwal yang telah dibooking oleh user berdasarkan userId.

Request:

- URL:
<https://asa99.my.id/appointments/booked/{userId}>
- Method: GET

Contoh hasil:

```
{  
  "id": 1,  
  "user_name": "John Doe",  
  "psychologist_name": "Dr. Anna",  
  "schedule_date": "2025-03-03",  
  "schedule_time": "11:00:00",  
  "created_at": "2025-03-01 11:12:12",  
  "email": "john.doe@example.com",  
  "status": "booked",  
  "user_id": "123"  
}
```

Booked Appointments

Berhasil jika ada jadwal yang dipesan user

A screenshot of the Postman application interface. The top bar shows a GET request to `https://asa99.my.id/appointments/booked/20`. The 'Body' tab is selected, showing the response body as JSON. The JSON response is:

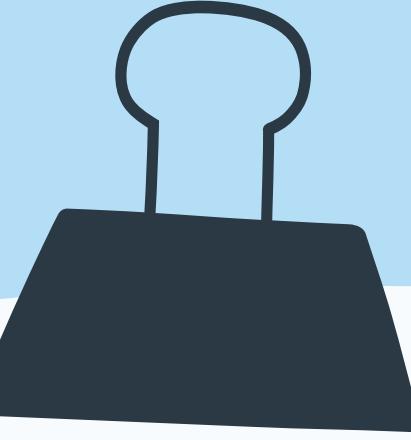
```
1 [  
2   {  
3     "id": "1",  
4     "user_name": "superman",  
5     "psychologist_name": "Dr. abc",  
6     "schedule_date": "2024-12-01",  
7     "schedule_time": "11:00:00",  
8     "status": "booked",  
9     "created_at": "2024-12-25 20:19:25",  
10    "email": "superman@example.com",  
11    "user_id": "20"  
12  }]  
13 ]
```

Tidak menampilkan hasil jika tidak ada jadwal yang dipesan

A screenshot of the Postman application interface. The top bar shows a GET request to `https://asa99.my.id/appointments/booked/13`. The 'Body' tab is selected, showing the response body as JSON. The JSON response is:

```
1 {  
2   "status": 404,  
3   "error": 404,  
4   "messages": {  
5     "error": "Tidak ada jadwal yang dibooking"  
6   }  
7 }
```

Cancel Appointments



Endpoint: DELETE /cancel/{id}

Rute ini digunakan untuk membatalkan reservasi jadwal berdasarkan ID. User yang membooking jadwal tersebut harus memberikan user_id dalam body request untuk memastikan bahwa hanya user yang membooking yang dapat membatalkan.

Request:

- URL:
[https://asa99.my.id/appointments/cancel
/{id}](https://asa99.my.id/appointments/cancel/{id})
- Method: DELETE

Contoh body Request

```
{  
  "user_id": 123  
}
```

Cancel Appointments

Berhasil jika id jadwal dan id user match

The screenshot shows a POST request to `https://asa99.my.id/appointments/cancel/1`. The Body tab is selected, showing the raw JSON payload:

```
1 {  
2   "user_id": 20  
3 }  
4
```

The response body shows a success message:

```
1 {  
2   "message": "Reservasi berhasil dibatalkan"  
3 }
```

Jadwal tidak ditemukan jika user tidak memiliki booked appointments yang bisa dibatalkan

The screenshot shows a DELETE request to `https://asa99.my.id/appointments/cancel/20`. The Body tab is selected, showing the raw JSON payload:

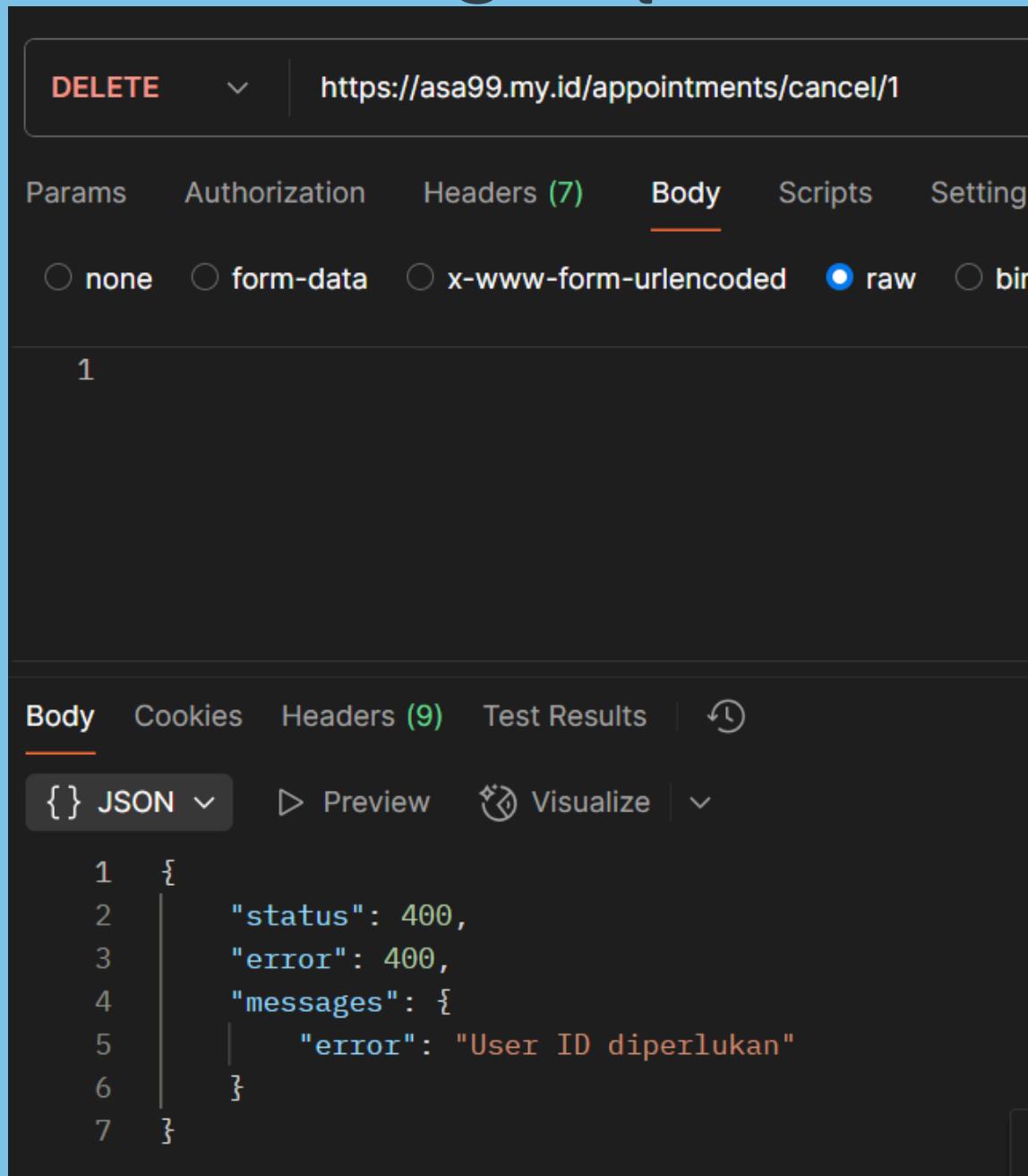
```
1 {  
2   "user_id": 20  
3 }  
4
```

The response status is 404 Not Found, and the response body is:

```
1 {  
2   "status": 404,  
3   "error": 404,  
4   "messages": {  
5     "error": "Jadwal tidak ditemukan"  
6   }  
7 }
```

Cancel Appointments

Jika tidak mengisi user id pada body request



DELETE https://asa99.my.id/appointments/cancel/1

Params Authorization Headers (7) Body Scripts Settings

None form-data x-www-form-urlencoded raw binary

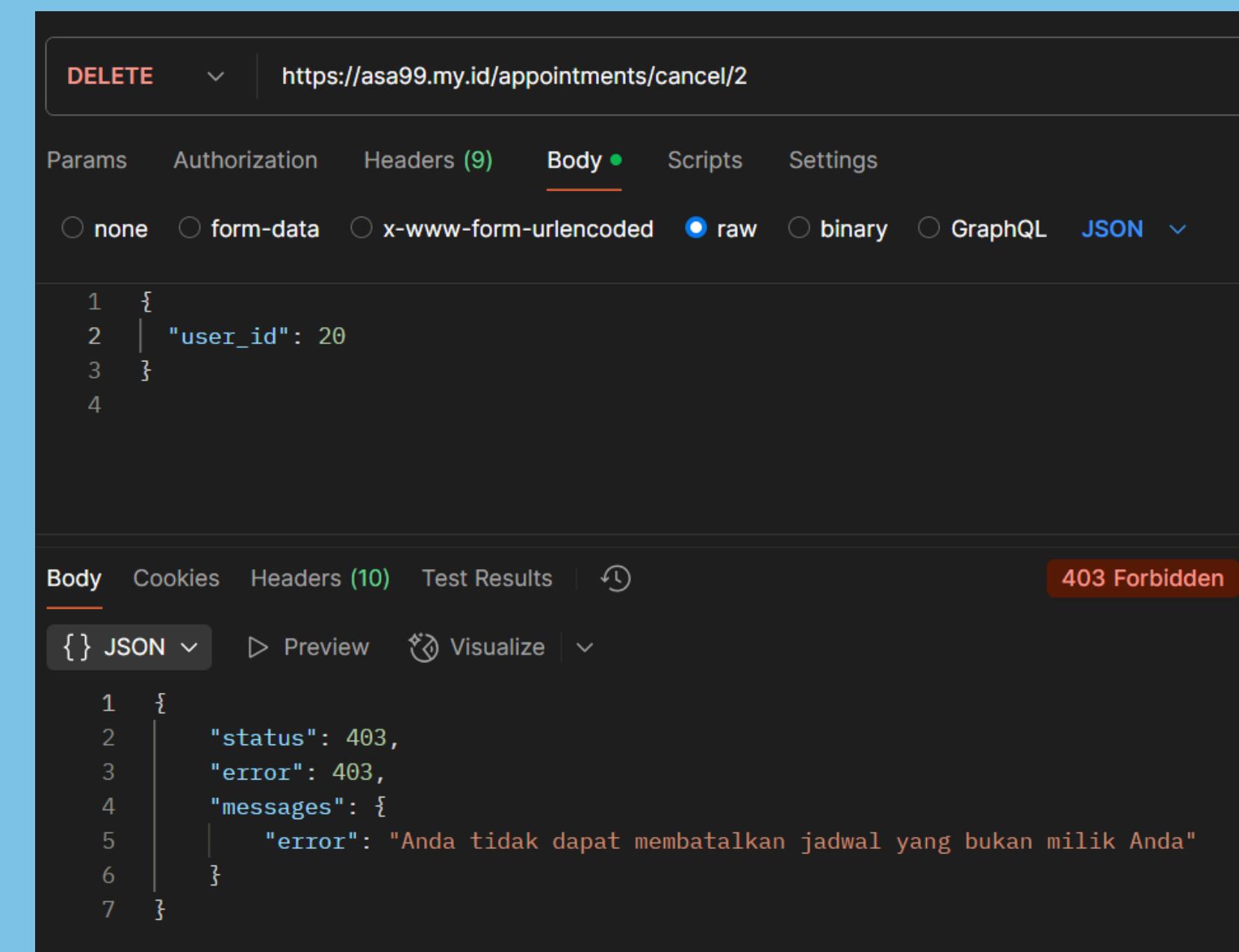
1

Body Cookies Headers (9) Test Results

{ } JSON ▾ Preview Visualize

```
1 {  
2   "status": 400,  
3   "error": 400,  
4   "messages": {  
5     "error": "User ID diperlukan"  
6   }  
7 }
```

Tidak dapat menghapus appointment jika id user yang dikirimkan tidak sesuai dengan id user yang melakukan book jadwal



DELETE https://asa99.my.id/appointments/cancel/2

Params Authorization Headers (9) Body Scripts Settings

None form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {
2 "user_id": 20
3 }
4

Body Cookies Headers (10) Test Results

{ } JSON ▾ Preview Visualize

```
1 {  
2   "status": 403,  
3   "error": 403,  
4   "messages": {  
5     "error": "Anda tidak dapat membatalkan jadwal yang bukan milik Anda"  
6   }  
7 }
```

403 Forbidden

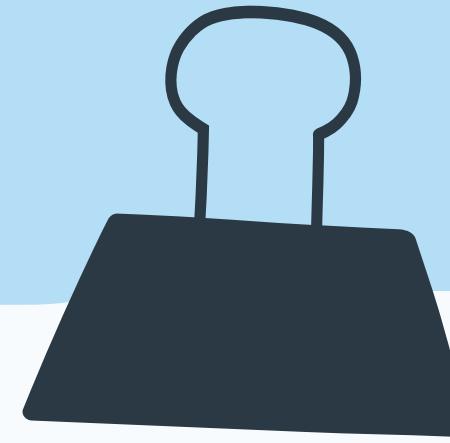
Menampilkan Artikel

Endpoint: GET /

Rute ini digunakan untuk menampilkan semua artikel yang dapat membantu user.

Request:

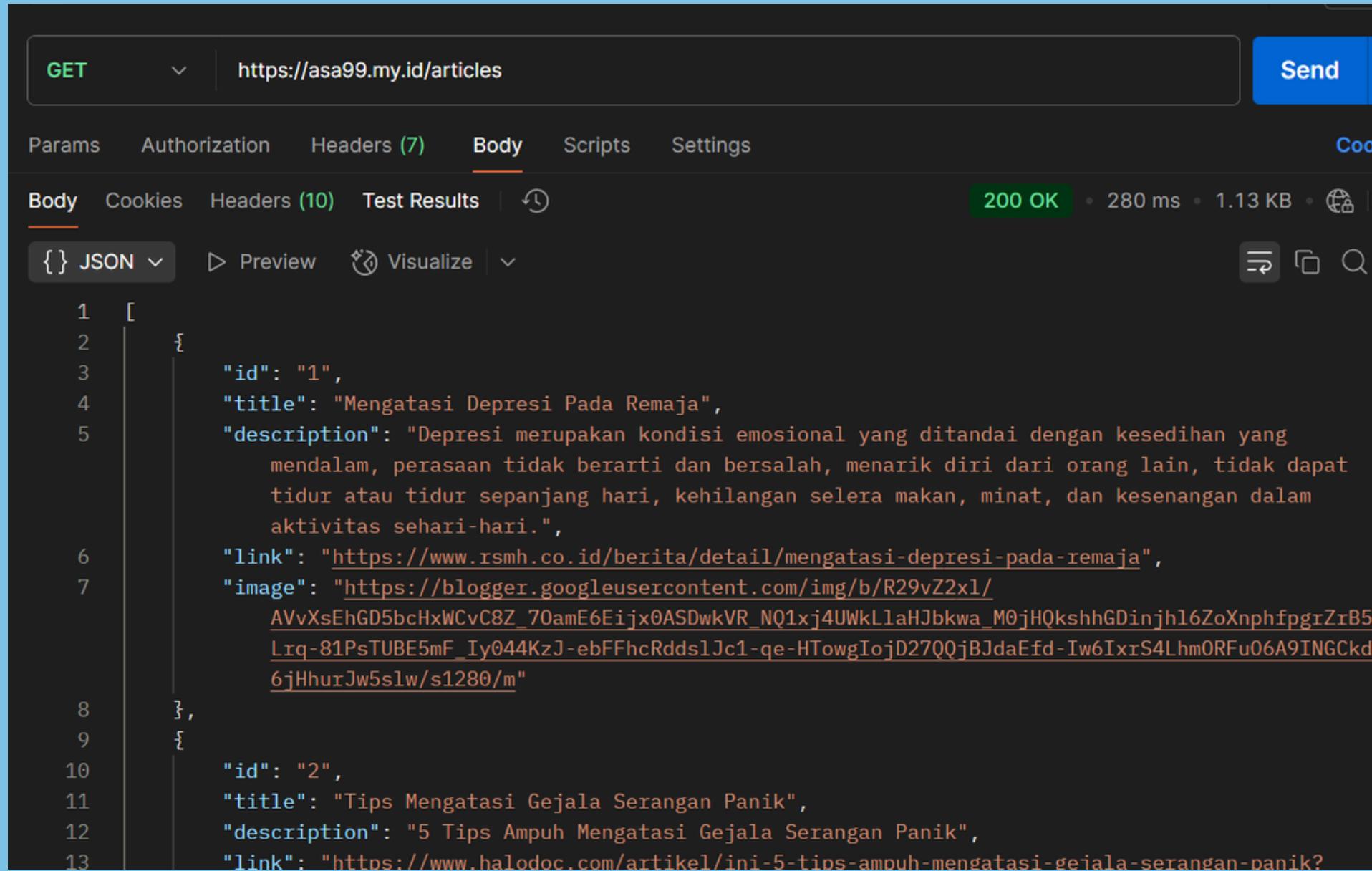
- URL: <https://asa99.my.id/articles>
- Method: GET



Contoh hasil Request

```
{"  
    "id": "1",  
    "title": "Mengatasi Depresi Pada Remaja",  
    "description": "Deskripsi",  
    "link": "https://www.rsmh.co.id/berita/detail/mdpr",  
    "image": "https://linkGambar"  
}
```

Menampilkan Artikel



A screenshot of the Postman application interface. The top bar shows a GET request to <https://asa99.my.id/articles>. The 'Body' tab is selected, showing a JSON response. The response is a list of articles, each containing an ID, title, description, link, and image URL. The image URL is a long, encoded string.

```
[{"id": "1", "title": "Mengatasi Depresi Pada Remaja", "description": "Depresi merupakan kondisi emosional yang ditandai dengan kesedihan yang mendalam, perasaan tidak berarti dan bersalah, menarik diri dari orang lain, tidak dapat tidur atau tidur sepanjang hari, kehilangan selera makan, minat, dan kesenangan dalam aktivitas sehari-hari.", "link": "https://www.rsmh.co.id/berita/detail/mengatasi-depresi-pada-remaja", "image": "https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhGD5bcHxWCvC8Z_70amE6Eijx0ASDwkVR_NQ1xj4UWkLlaHJbkwa_M0jHQkshhGDinjh16ZoXnphfpgzrB5hLrq-81PsTUBE5mF_Iy044KzJ-ebFFhcRdds1Jc1-qe-HTowgIojD27QQjBJdaEfd-Iw6IxixS4Lhm0RFu06A9INGCkdW6jHhurJw5slw/s1280/m"}, {"id": "2", "title": "Tips Mengatasi Gejala Serangan Panik", "description": "5 Tips Ampuh Mengatasi Gejala Serangan Panik", "link": "https://www.halodoc.com/artikel/ini-5-tips-ampuh-mengatasi-gejala-serangan-panik?"}]
```

