

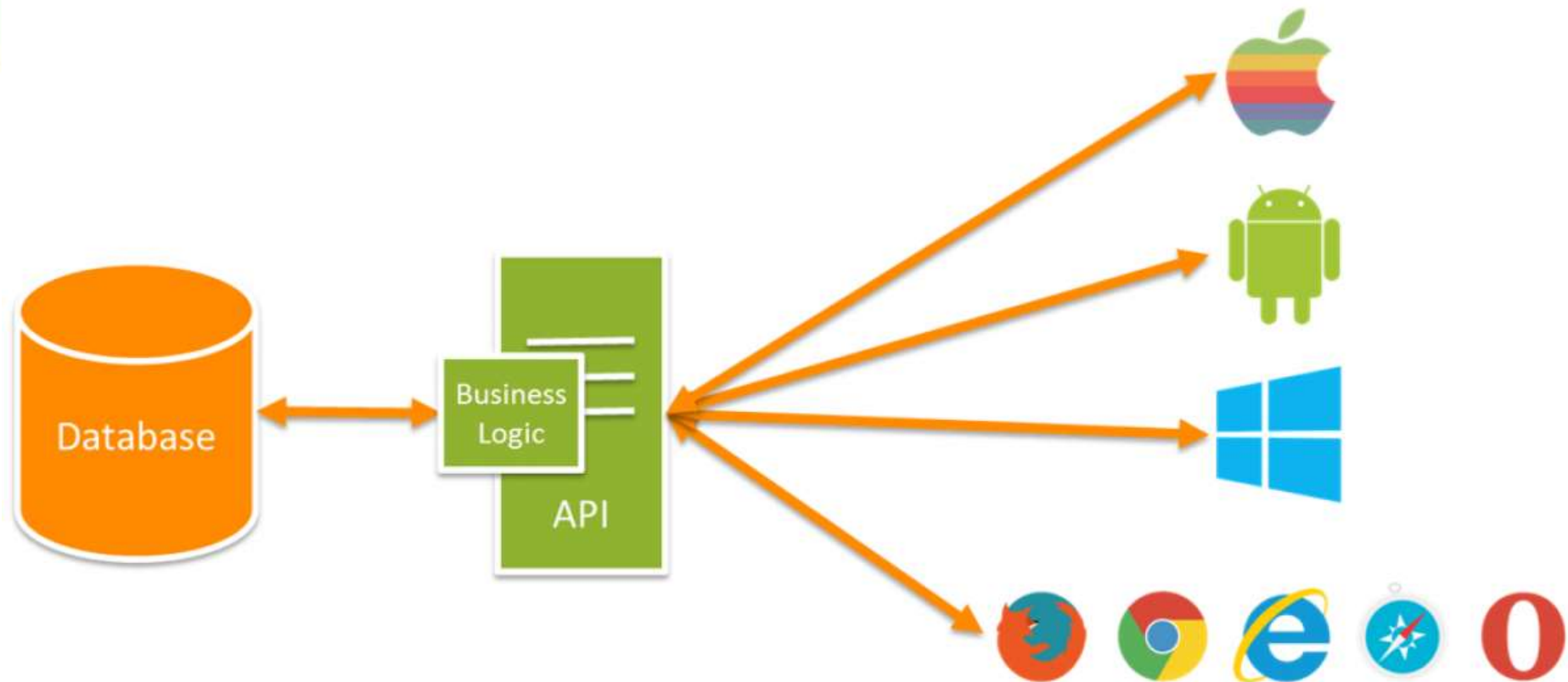
PERTEMUAN 10

Apa itu API

API atau Application Programming Interface adalah sebuah interface yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya. Jadi, API berperan sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau lintas platform.

Perumpamaan yang bisa digunakan untuk menjelaskan API adalah seorang pelayan di restoran. Tugas pelayan tersebut adalah menghubungkan tamu restoran dengan juru masak. Tamu cukup memesan makanan sesuai daftar menu yang ada dan pelayan memberitahukannya ke juru masak. Nantinya, pelayan akan kembali ke tamu tadi dengan masakan yang sudah siap sesuai pesanan.

Itulah gambaran tugas dari API dalam pengembangan aplikasi.



Sumber : codepolitan.com

Arsitektur API

Ada tiga arsitektur API yang sering digunakan oleh developer dalam pembangunan aplikasi. Arsitektur ini berkaitan pada bentuk data yang dikirim. Adapun Arsitektur API yang sering digunakan adalah

1. RPC

RPC merupakan teknologi untuk membuat komunikasi antara client side dan server side bisa dilakukan dengan konsep sederhana.

RPC memiliki dua jenis, yaitu XML-RPC dan JSON-RPC. Sesuai namanya, XML-RPC menggunakan format XML sebagai media perpindahan data, sedangkan JSON-RPC menggunakan JSON untuk perpindahan data.

2. SOAP

Arsitektur API lainnya adalah SOAP (Simple Object Access Protocol). Arsitektur ini menggunakan XML (Extensible Markup Language) yang memungkinkan semua data disimpan dalam dokumen.

3. REST

REST atau Representational State Transfer adalah arsitektur API yang cukup populer karena kemudahan penggunaannya. Tak perlu coding yang panjang untuk menggunakannya.

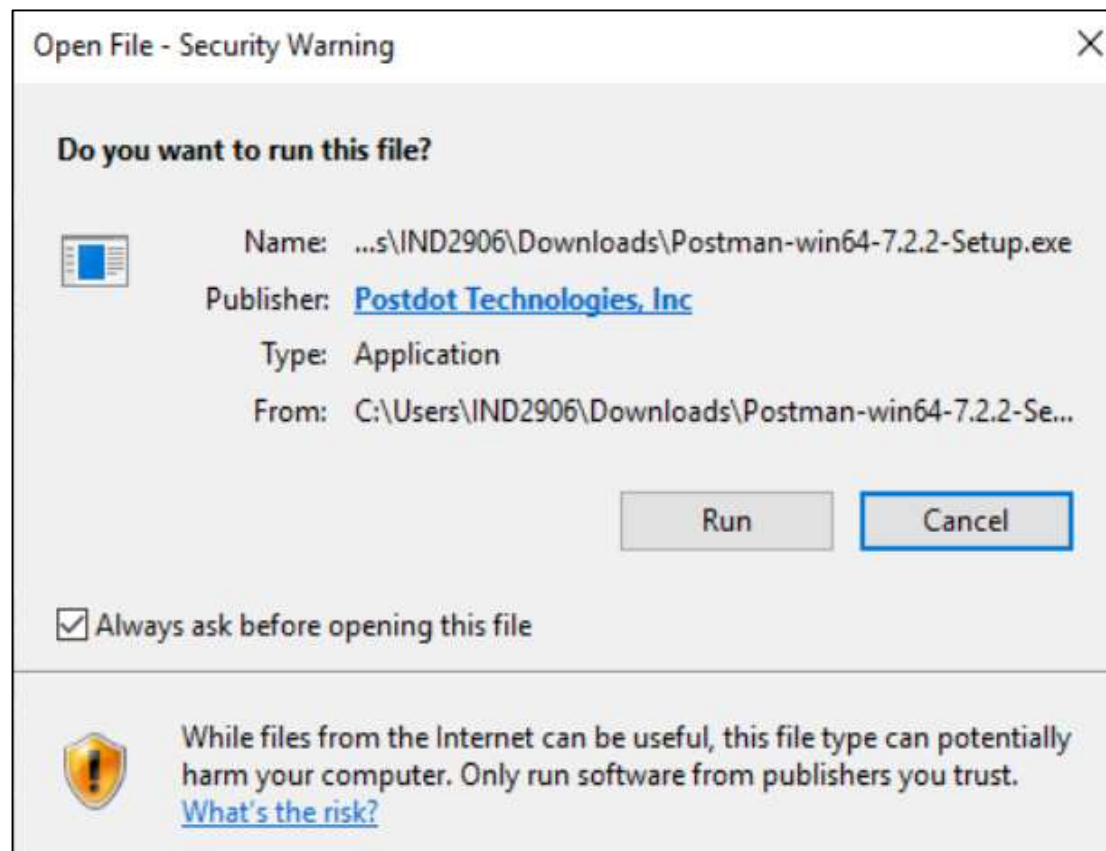
REST menggunakan JSON sebagai bentuk datanya sehingga lebih ringan. Performa aplikasi pun menjadi lebih baik.

Install Postman

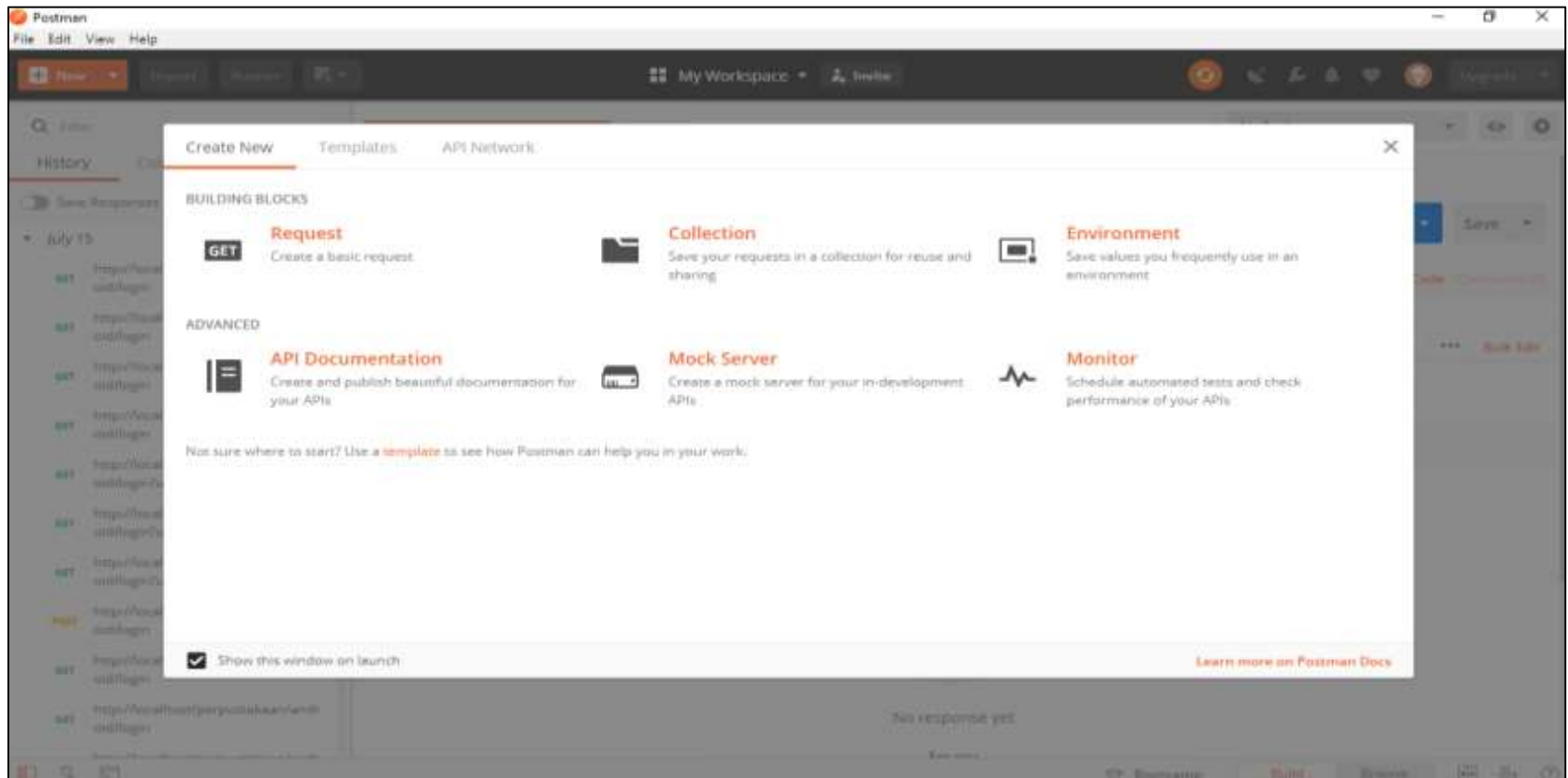
Postman adalah sebuah aplikasi fungsinya adalah sebagai REST Client atau istilahnya adalah aplikasi yang digunakan untuk melakukan uji coba REST API yang telah kita buat. Postman ini merupakan tools wajib bagi para developer yang bergerak pada pembuatan API, fungsi utama postman ini adalah sebagai GUI API Caller Pemanggil. namun sekarang postman juga menyediakan fitur lain yaitu Sharing Collection API for Documentation (free), Testing API (free), Realtime Collaboration Team (paid), Monitoring API (paid), Integration (paid).

Postman tersedia sebagai aplikasi asli untuk sistem operasi macOS, Windows (32-bit dan 64-bit), dan Linux (32-bit dan 64-bit). Untuk mendapatkan aplikasi Postman, dapat diunduh pada website resminya yaitu [getpostman.com](https://www.postman.com/downloads/) atau dapat diunduh pada halaman <https://www.postman.com/downloads/>

Setelah berhasil mengunduh paket instalasi postman, kemudian jalankan dengan cara klik dua kali. Pilih run jika muncul pop up seperti berikut :



Kemudian tunggu hingga proses instalasi selesai dan muncul seperti gambar berikut

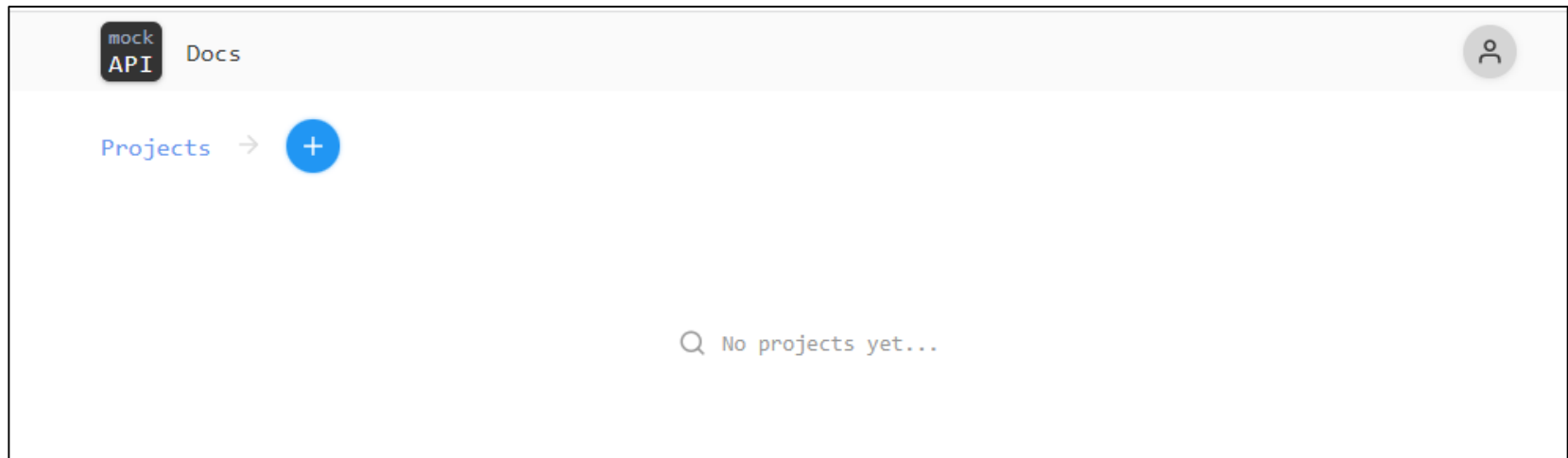


Membuat API Klinik dummy dengan mockapi.io

Silahkan kunjungi laman <https://mockapi.io/> kemudian daftar/buat akun mockapi.io



Jika sudah login ke dalam mockapi.io kemudian buat projek dengan mengklik tombol tambah



Buat projek dengan nama **klirik** kemudian klik tombol **create**

×

Project name

klirik

API Prefix (optional)

Add API prefix to all endpoints in this project.

Example: /api/v1

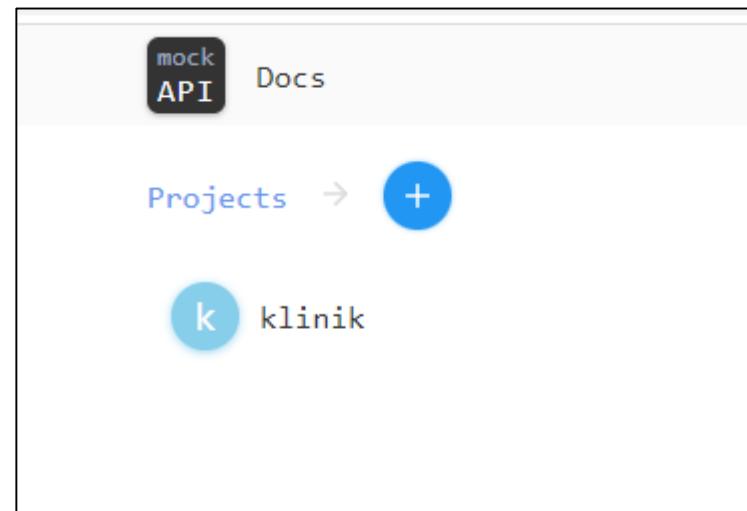
Collaborators (optional)

Collaborators can **create**, **update**, and **delete** resources in this project.

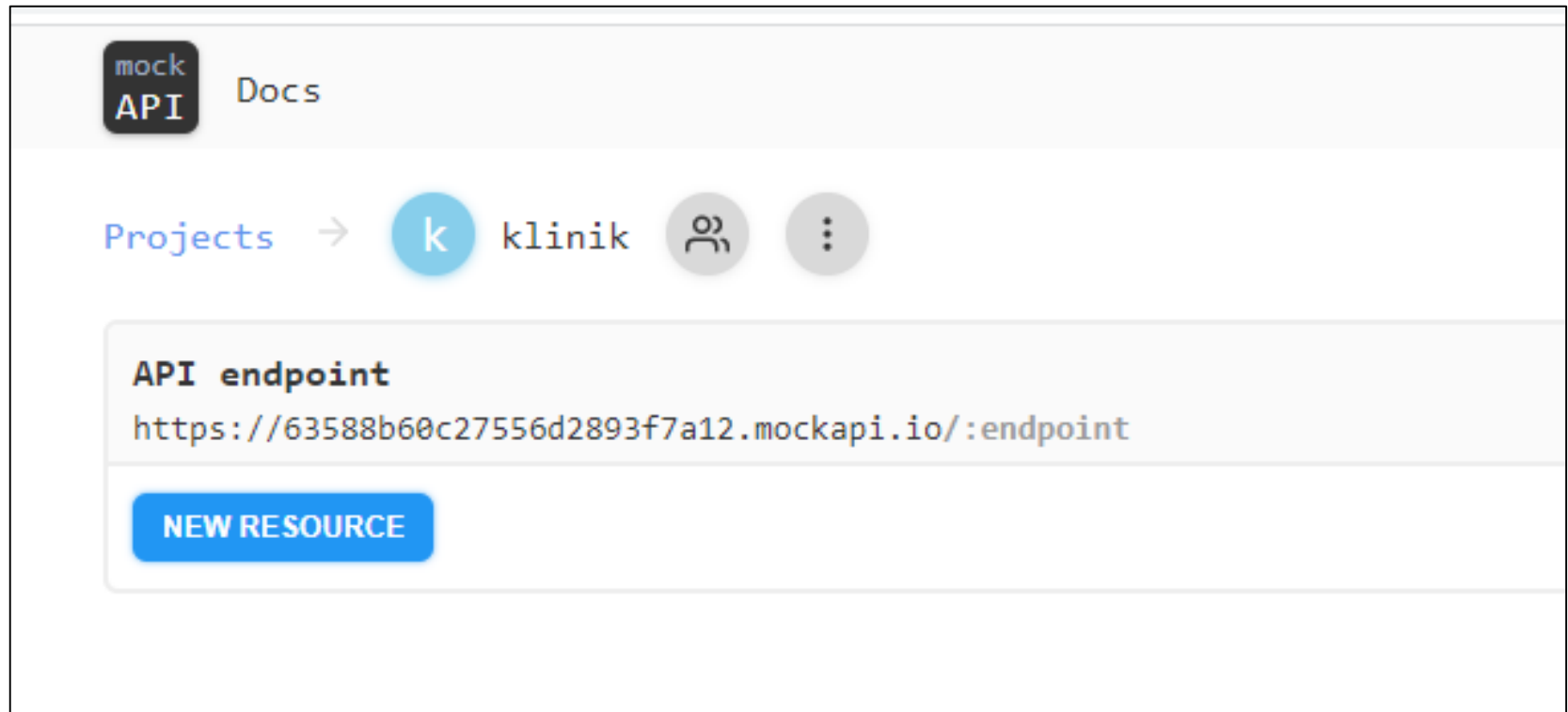
Search by name...

CREATE

CANCEL



Kemudian klik proyek klinik yang telah dibuat tadi sehingga muncul seperti tampilan berikut



Klik tombol **New Resource** sehingga muncul tampilan berikut

×

Resource name
Enter meaningful resource name, it will be used to generate API endpoints.

Schema (optional)
Define Resource schema, it will be used to generate mock data.

id	Object ID	
createdAt	Faker.js	Recent
name	Faker.js	Find name
avatar	Faker.js	Avatar
<div>+</div>		

Object template (optional)
To define more complex structure for your data use JSON template. You can reference
Faker.js methods using `faker.lorem.paragraph(10)`

CREATE

CANCEL

Isi **Resource Name** dengan **poli** kemudian pada bagian **Schema** ubah sehingga menjadi seperti gambar berikut

×

Resource name
Enter meaningful resource name, it will be used to generate API endpoints.

Schema (optional)
Define Resource schema, it will be used to generate mock data.

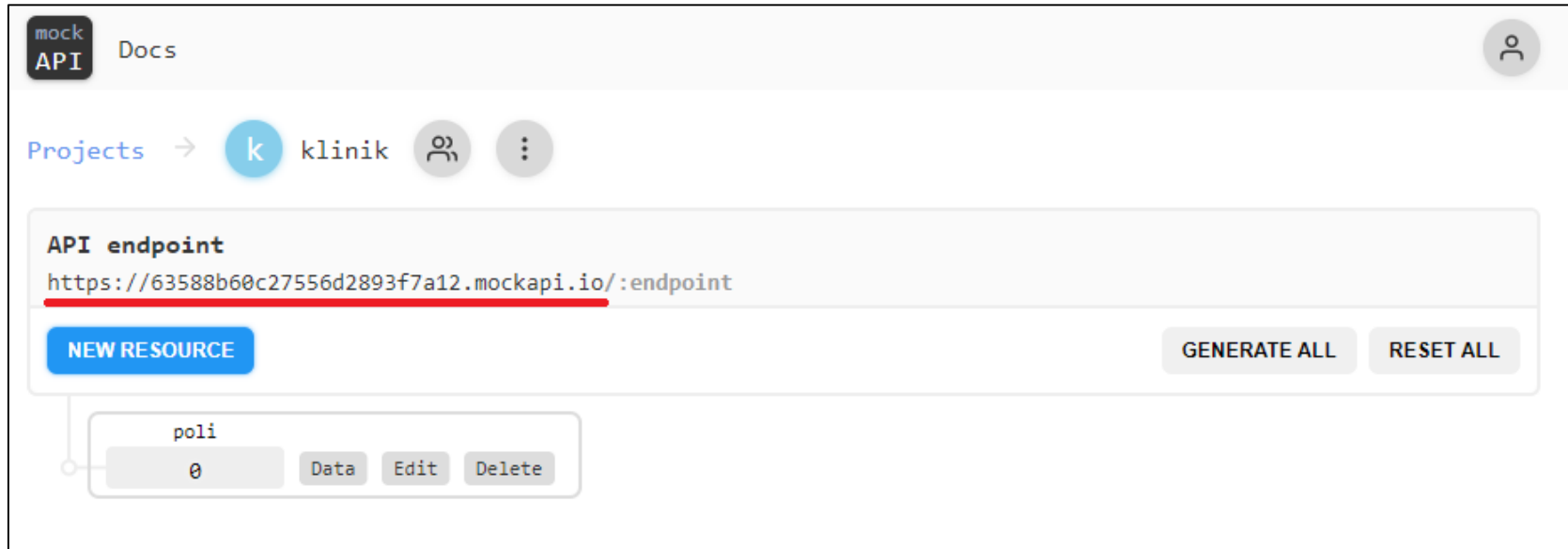
<input type="text" value="id"/>	<input type="text" value="Object ID"/>
<input type="text" value="nama_poli"/>	<input type="text" value="String"/>

Object template (optional)
To define more complex structure for your data use JSON template. You can reference Faker.js methods using `\$.`

```
{  
  "username": "$internet.userName",  
  "knownIps": ["$internet.ip", "$internet.ipv6"],  
}
```

EXAMPLE

Kemudian klik tombol **create** dan hasilnya akan menjadi seperti berikut



Pada bagian bawah **API endpoint** (yang digaris bawah merah) merupakan alamat utama url yang dapat kita kunjungi untuk mengakses API.

Untuk mengakses data mahasiswa yang telah kita buat sebelumnya, kita dapat mengakses dengan alamat `https://63588b60c27556d2893f7a12.mockapi.io/poli` (sesuaikan dengan url yang ada pada mockapi anda)

Create Poli dengan method POST

POST <https://63588b60c27556d2893f7a12.mockapi.io/poli> Send 201 Created 19 s 34 B

JSON Auth Query Headers 1 Docs Preview Headers 11 Cookies Timeline

```
1 {
2   "nama_poli": "Poli Umum"
3 }
```

```
1 {
2   "nama_poli": "Poli Umum",
3   "id": "1"
4 }
```

List Poli dengan method GET

GET <https://63588b60c27556d2893f7a12.mockapi.io/poli> Send 200 OK 683 ms 71 B

Body Auth Query Headers Docs Preview Headers 12 Cookies Timeline

```
1 [
2   {
3     "nama_poli": "Poli Umum",
4     "id": "1"
5   },
6   {
7     "nama_poli": "Poli Anak",
8     "id": "2"
9   }
10 ]
```

Show dengan method GET

GET <https://63588b60c27556d2893f7a12.mockapi.io/poli/1> Send 200 OK 1.29 s 34 B

Body Auth Query Headers Docs Preview Headers 12 Cookies Timeline

```
1 {  
2   "nama_poli": "Poli Umum",  
3   "id": "1"  
4 }
```

Update Poli dengan method PUT

PUT <https://63588b60c27556d2893f7a12.mockapi.io/poli/2> Send 200 OK 3.76 s 39 B

JSON Auth Query Headers 1 Docs Preview Headers 11 Cookies Timeline

```
1 {  
2   "nama_poli": "Poli Kandungan"  
3 }
```

```
1 {  
2   "nama_poli": "Poli Kandungan",  
3   "id": "2"  
4 }
```

Delete Poli dengan method DELETE

DELETE ▼ <https://63588b60c27556d2893f7a12.mockapi.io/poli/2> Send ▼ 200 OK 22.7 s 39 B

Body ▼ Auth ▼ Query Headers Docs

Preview ▼ Headers 11 Cookies Timeline

```
1 {  
2   "nama_poli": "Poli Kandungan",  
3   "id": "2"  
4 }
```

GET ▼ <https://63588b60c27556d2893f7a12.mockapi.io/poli> Send ▼ 200 OK 274 ms 36 B

Body ▼ Auth ▼ Query Headers Docs

Preview ▼ Headers 12 Cookies

```
1 [  
2   {  
3     "nama_poli": "Poli Umum",  
4     "id": "1"  
5   }  
6 ]
```

TUGAS

Buat tabel (Resource) dengan skema seperti berikut

Pegawai
<ul style="list-style-type: none">- id: int- nip: String- nama: String- tanggal_lahir: Date- nomor_telepon: String- email: String- password: String

Pasien
<ul style="list-style-type: none">- id: int- nomor_rm: String- nama: String- tanggal_lahir: Date- nomor_telepon: String- alamat: String