Tugas 2

Nama: Dimas Putra Rahardika

Nim: 220411100087

1. Jelaskan secara singkat konsep dasar REST (Representational State Transfer). Sebutkan dan jelaskan prinsip RESTful API (stateless, resource-based, client-server, cacheable, dll.). Berikan contoh skenario penggunaan REST API dalam dunia nyata.
2. Jelaskan masing-masing metode HTTP yang digunakan dalam REST API (GET, POST, PUT, DELETE, PATCH).
3. Berikan contoh endpoint REST API beserta fungsi dari masing-masing metode HTTP.
4. Status Code HTTP
5. Jelaskan status code HTTP yang umum digunakan dalam REST API.
6. Berikan contoh kasus penggunaan masing-masing status code berikut:

Jawaban

1. REST (Representational State Transfer) adalah arsitektur untuk membangun layanan web yang berskala besar. REST bukan standar, melainkan sekumpulan prinsip dan batasan yang harus diikuti untuk menciptakan API yang efisien dan fleksibel.

Prinsip-prinsip REST

1. Uniform Interface

Semua interaksi dengan API harus konsisten, menggunakan format standar seperti JSON dan endpoint yang jelas.

Contohnya: API pengguna dengan URL konsisten untuk operasi berbeda:

* Ambil semua pengguna: GET /users
* Tambah pengguna: POST /users
* Update pengguna: PUT /users/1
* Hapus pengguna: DELETE /users/1

1. Client-server

API harus memisahkan antara klien (frontend) dan server (backend) sehingga keduanya bisa dikembangkan secara independen.

Contohnya: Klien dan server bisa diperbarui atau diganti tanpa saling mempengaruhi.

1. Stateless

Setiap permintaan API harus bersifat mandiri, artinya server tidak menyimpan status pengguna antara permintaan.

Contohnya: Setiap permintaan harus berisi semua informasi yang diperlukan (tanpa menyimpan sesi).

1. Cacheable

Respons API dapat disimpan (cache) untuk mengurangi beban server dan mempercepat waktu respons.

Contohnya: respon yang terjadi berulang di simpan sehingga tidak membebani server

1. Layered System

API dapat memiliki beberapa lapisan (layer) untuk meningkatkan keamanan, performa, dan skalabilitas.

Contoh: API Gateway bisa memvalidasi permintaan sebelum meneruskannya ke backend.

1. Code on Demand

Server dapat mengirimkan kode (misalnya JavaScript) ke klien untuk dieksekusi.

Contonya: Mengurangi beban komputasi di server dengan mengalihkan logika ke klien.

1. Methot http REST API
2. GET (Mengambil Data). Digunakan untuk mengambil data dari server tanpa mengubahnya.
3. POST (Menambahkan Data Baru). Digunakan untuk menambahkan data baru ke server.
4. PUT (Memperbarui Seluruh Data). Digunakan untuk memperbarui seluruh data yang ada di server.
5. PATCH (Memperbarui Sebagian Data). Digunakan untuk memperbarui hanya sebagian data tanpa mengganti seluruhnya.
6. DELETE (Menghapus Data). Digunakan untuk menghapus data dari server.
7. Endpoint

|  |  |  |
| --- | --- | --- |
| Metode | Endpoint | fungsi |
| |  | | --- | | GET | | |  | | --- | | /users | | Mengambil daftar semua pengguna |
| |  | | --- | | GET | | |  |  | | --- | --- | | /users/{id} |  | | Mengambil detail pengguna berdasarkan ID |
| |  | | --- | | POST | | |  |  | | --- | --- | | /users |  | | Menambahkan pengguna baru |
| |  | | --- | | PUT | | |  |  | | --- | --- | | /users/{id} |  | | Memperbarui seluruh data pengguna berdasarkan ID |
| |  | | --- | | PATCH | | |  |  | | --- | --- | | /users/{id} |  | | Memperbarui sebagian data pengguna |
| |  | | --- | | DELETE | | |  |  | | --- | --- | | /users/{id} |  | | Menghapus pengguna berdasarkan ID |

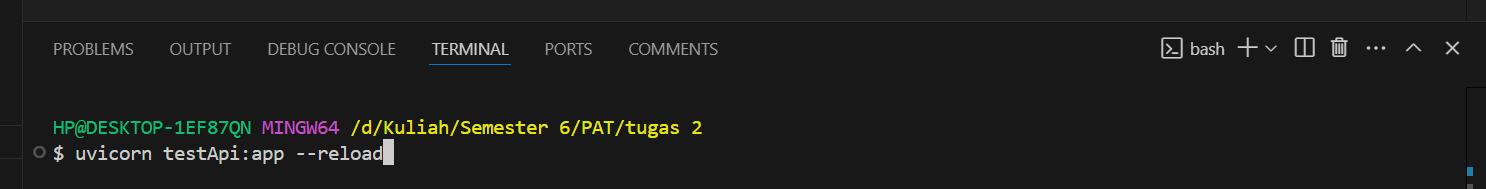
1. Code HTTP API
2. Penjelasan Status Code HTTP

* 1xx – Informational (Informasi). Indikasi bahwa permintaan sedang diproses oleh server.
* 2xx – Success (Permintaan Berhasil). Indikasi bahwa permintaan berhasil diproses.
* 3xx – Redirection (Pengalihan). Menunjukkan bahwa klien harus mengikuti URL lain untuk mendapatkan data yang diminta.
* 4xx – Client Error (Kesalahan di Pihak Klien). Menunjukkan bahwa permintaan dari klien salah atau tidak dapat diproses.
* 5xx – Server Error (Kesalahan di Pihak Server). Menunjukkan bahwa terjadi kesalahan di server saat memproses permintaan.

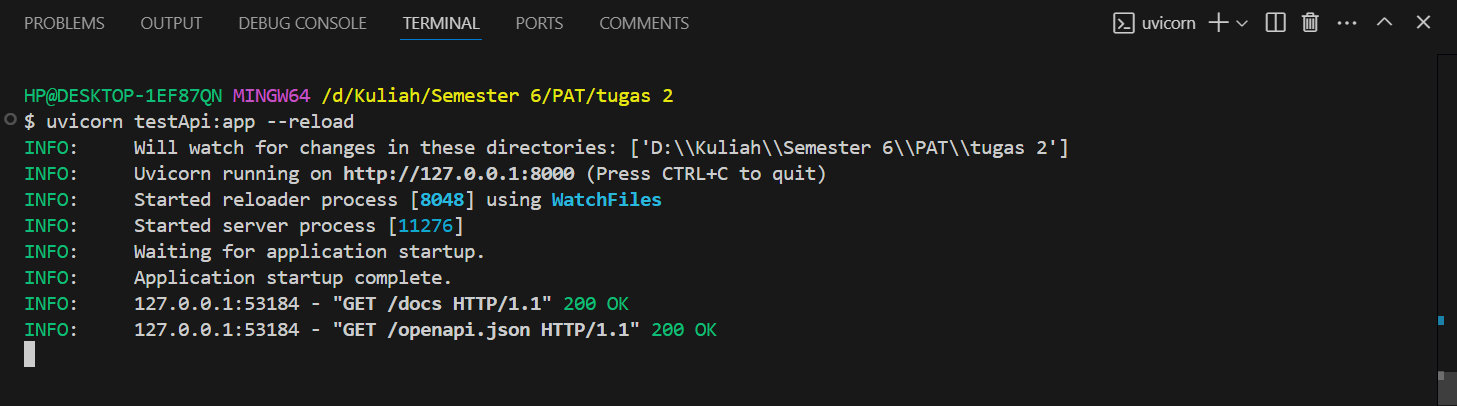
1. Contoh Kasus Penggunaannya code HTTP

|  |  |  |
| --- | --- | --- |
| Status code | Arti | Contoh pengunaan |
| 200 OK | Permintaan berhasil | Ambil daftar produk |
| 201 Created | Data berhasil dibuat | Tambah pengguna baru |
| 400 Bad Request | Format data salah | Kirim data dengan JSON yang salah |
| 401 Unauthorized | Belum login | Akses profil tanpa token |
| 403 Forbidden | Tidak memiliki izin | Akses data admin tanpa hak akses |
| 404 Not Found | Data tidak ditemukan | Mencari produk yang tidak ada |
| 500 Internal Server Error | Kesalahan server | Database crash |
| 502 Bad Gateway | Server menerima respons buruk dari upstream | API Gateway gagal memproses |

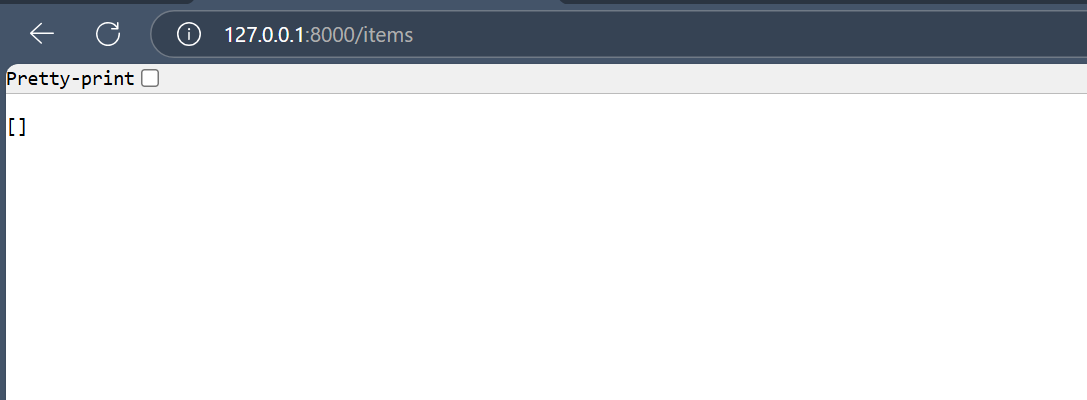
1. Dokumentasi pengunaan REST API
2. Pertama yang perlu di lakukan adalah untuk menjalankan server api. Kebetulan saya mengunakan framework API yaitu FastAPI.



Yaitu dengan cara memasukan “uvicorn Nama\_file:app –reload”

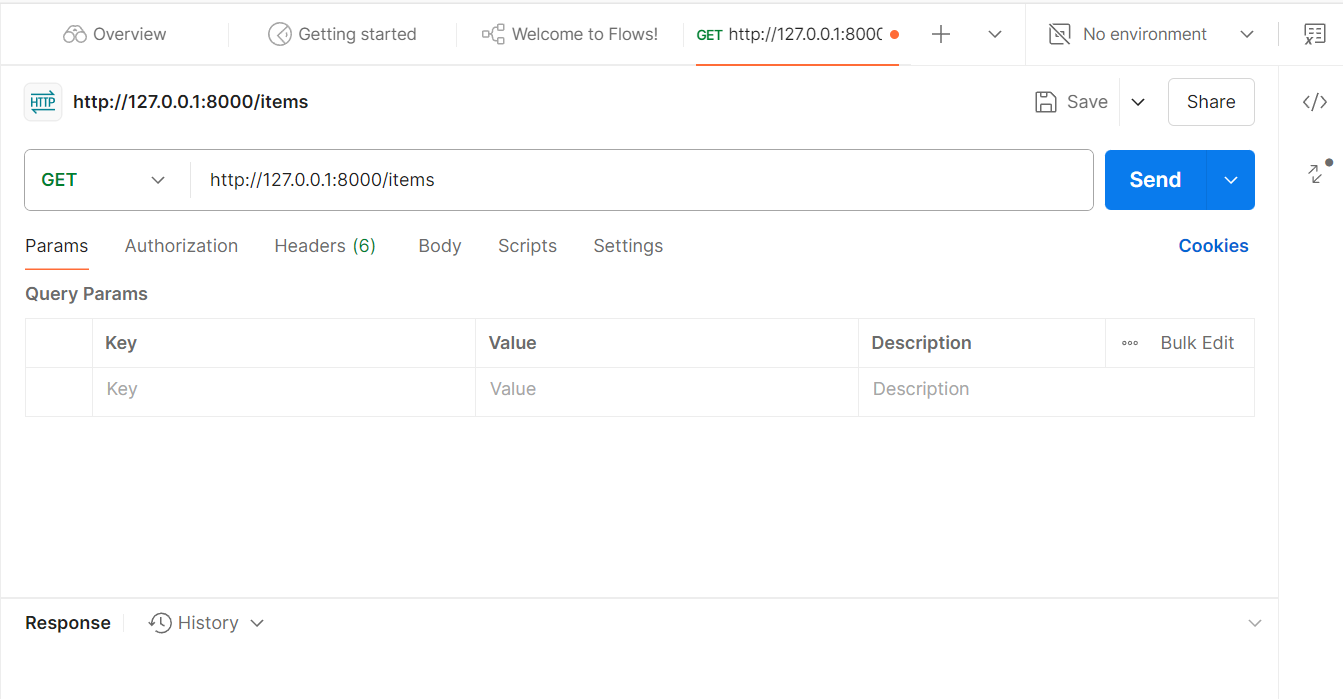


Kemudian kita akan mendapatkan link berupa ip menuju halaman keluaran API. Halaman tersebut seperti:



Belum ada isinya karena data masih kosong. Halaman tersebut merupakan request data atau get data.

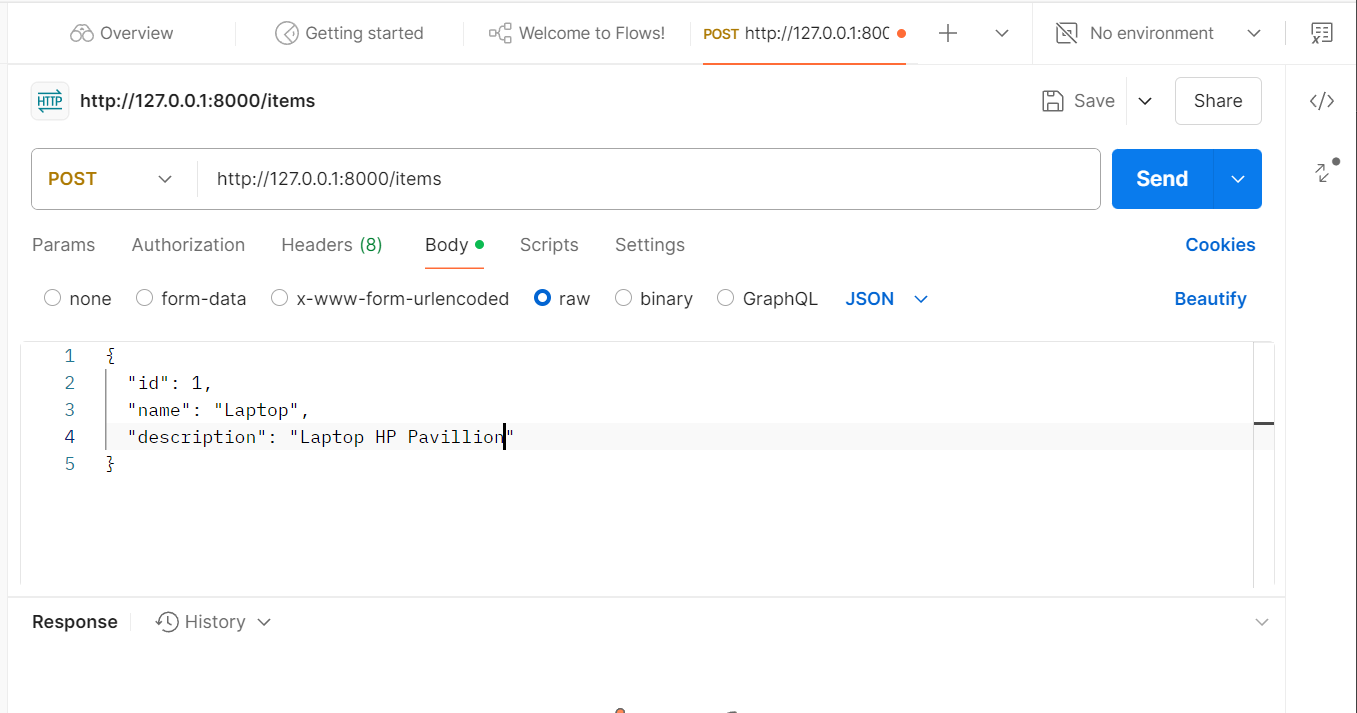
1. Untuk melakukan percobaan di sini saya mengunakan post man. Caranya adalah dengan memasukan alamt API ke postman.



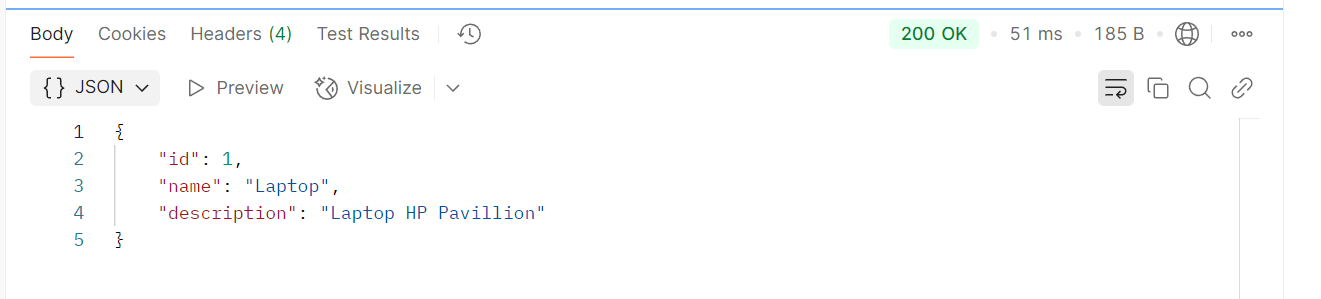
Berikut merupakan percobaan 3 endpoint get, post, delete.

1. Post (menambahkan data)

Pertama kita perlu untuk mengubah methot menjadi post dan buka tab body kemudian pilih raw dan ubah bagasanya ke json, dan setelahnya kita dapat memasukan data dengan format json contohnya:



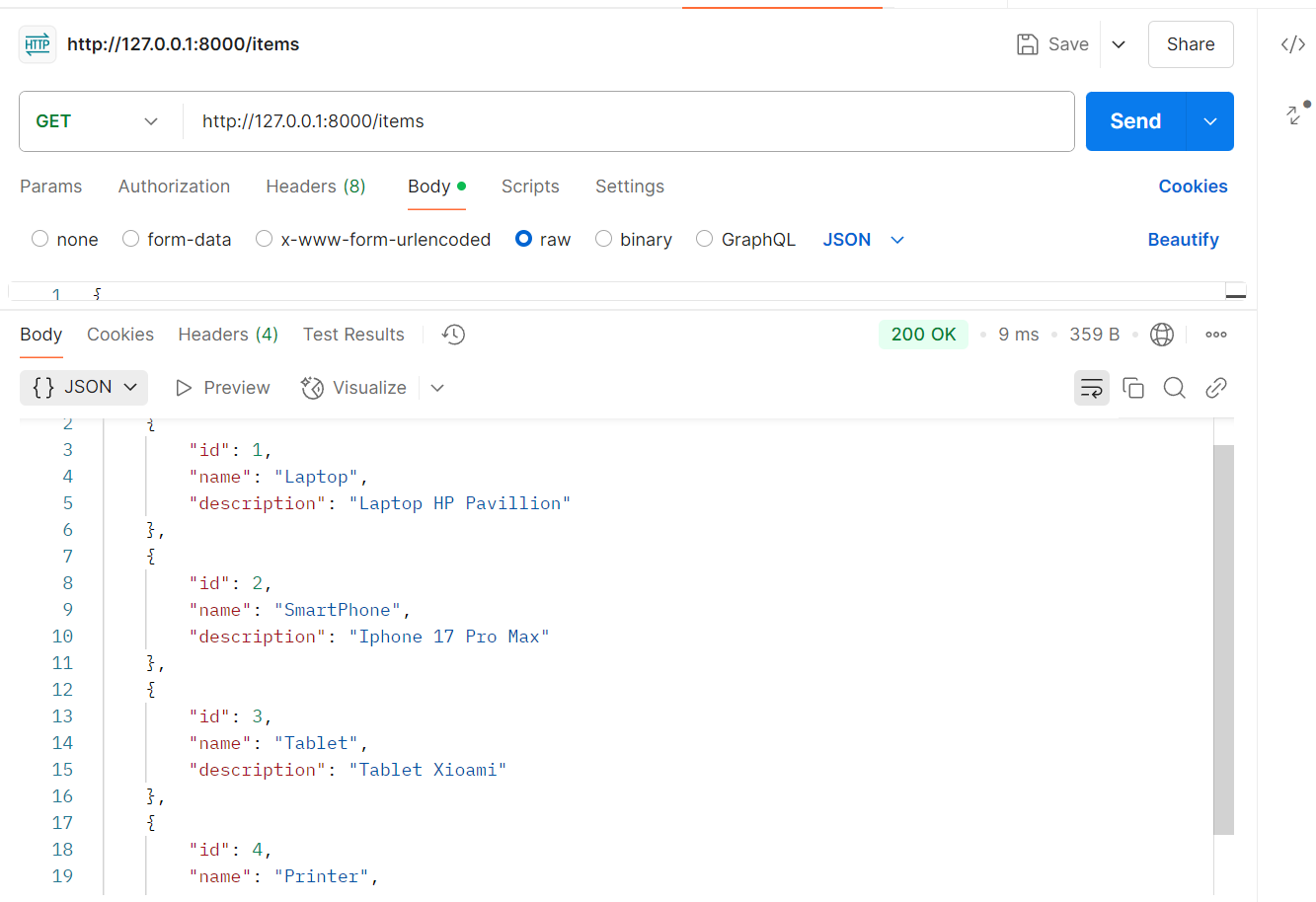
Kemudian setelahnya klik send, maka data akan di inputkan melalui api dengan status 200 ok



Saya juga akan menambahkan beberapa data lagi untuk percobaan.

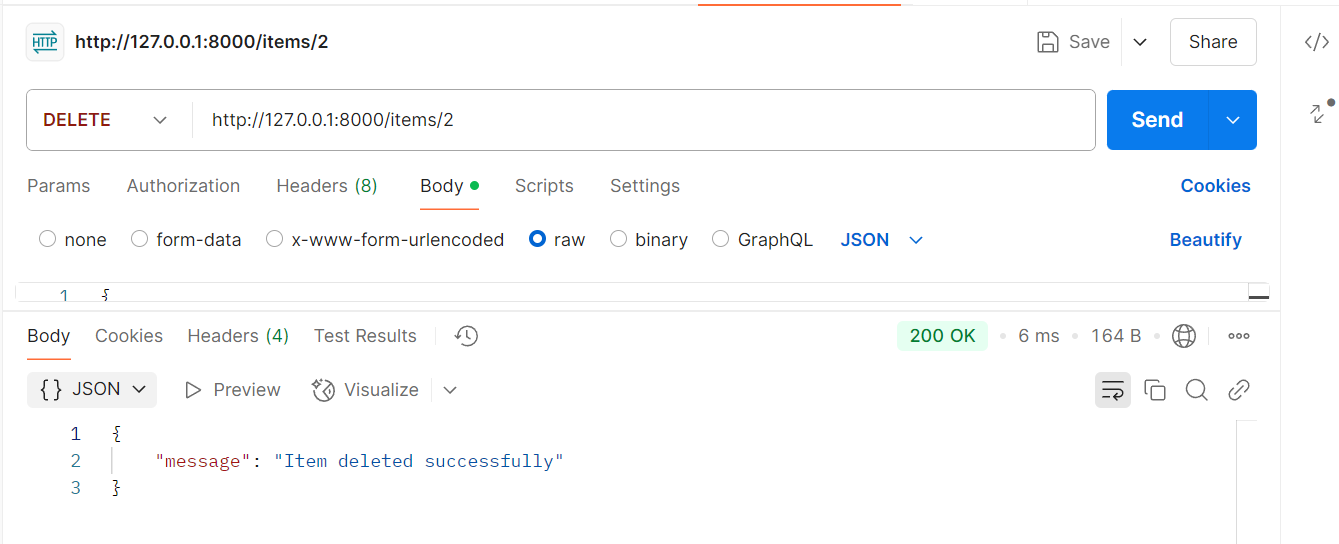
1. Get (mendapatkan data dari API)

Caranya kita hanya perlu merubah methot ke get den klik send. Dan seharusnya kita akan mendapatkan data dari api tersebut dengan status code 200 ok.

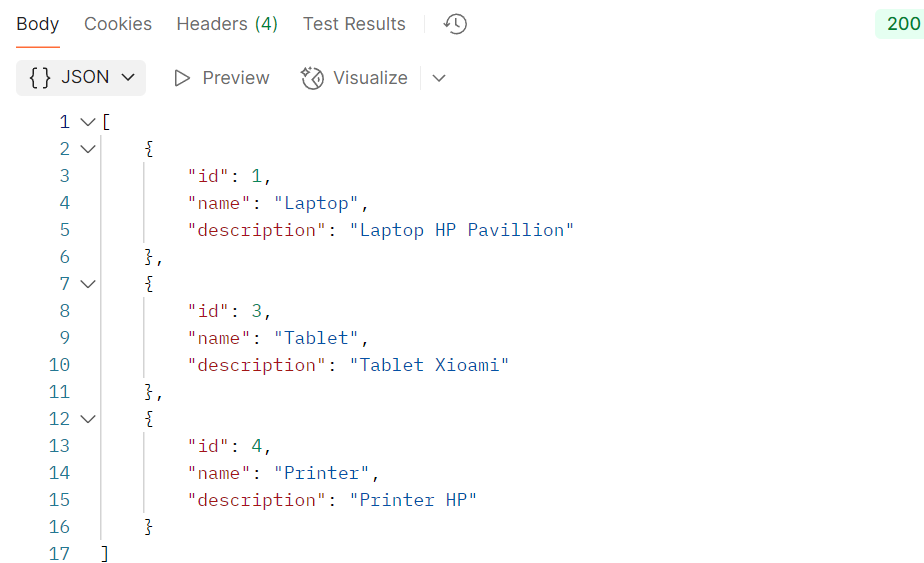


1. Delete (menghapus data di data API)

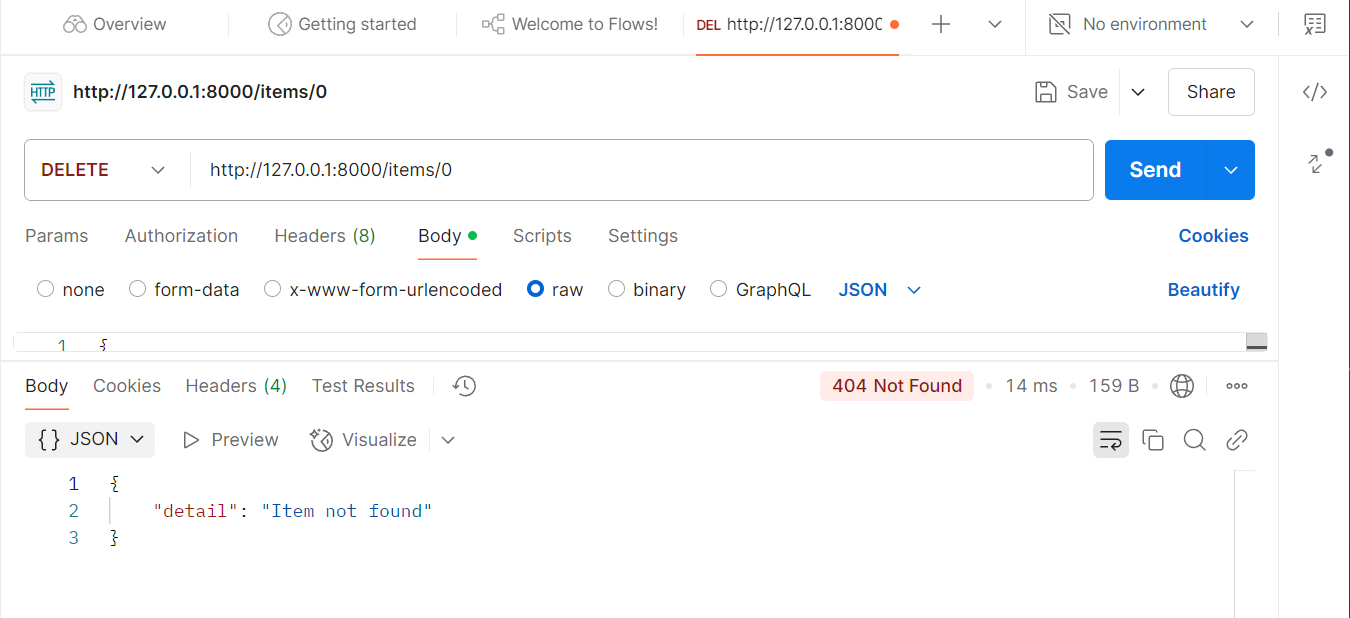
Kita perlu merubah methot menjadi delete den memasukan id dari item yang inggin kita hapus di dalam link http contohnya :” <http://127.0.0.1:8000/items/2>” kemudian kita send.



Dan jika lakukan get data dengan id no 2 akan hilang



Dan jika id item yang tidak ada saya buat pesan jika data tidak ditemukan dengan code 404 data not found



1. link repository