MATERI

Hari-14-Rest Api & State

Stateful vs Stateless Widget

Hal yang unik dan menarik perhatian dari Flutter adalah **stateless** dan **stateful** *widget*, dimana kedua bagian ini memegang perannya masing-masing. ***Stateless widget*** adalah *widget* statis dimana seluruh konfigurasi yang dimuat didalamnya telah diinisiasi sejak awal. Sedangkan ***Stateful widget*** berlaku sebaliknya dimana sifatnya adalah dinamis, sehingga *widget* ini dapat diperbaharui kapanpun dibutuhkan berdasarkan *user actions* atau ketika terjadinya perubahan data.

Untuk lebih jelas mengenai peran keduanya, maka pada artikel ini kita akan mengimplementasikan kedua jenis widget tersebut kedalam *case* yang telah dijelaskan sebelumnya. Pertama, buat *project* baru terlebih dahulu dengan *command* berikut & tunggu hingga proses instalasinya selesai:

untuk lebih jelas nya bisa mengikut video ini

https://www.youtube.com/watch?v=dNlWzMI6CgY

RESTful API

API (Application Program Interface) merupakan suatu bentuk interaksi antar program dalam bentuk request dan response, dimana response yang diberikan bergantung pada isi request nya.

REST (Representional State Transfer) adalah suatu arsitektur metode komunikasi untuk pertukaran data yang umumnya menggunakan protokol HTTP. REST menjadikan objek/data pada server sebagai sumber data yang dapat ditambah, diubah atau dihapus. Dan umumnya REST dapat digunakan di berbagai bahasa pemrograman.

Berdasarkan pada definisi keduanya, maka RESTful API merupakan bentuk komunikasi antar program (API) dengan menggunakan arsitektur REST, yang diantaranya menggunakan protokol HTTP. Beberapa metode HTTP yang umum digunakan, antara lain:

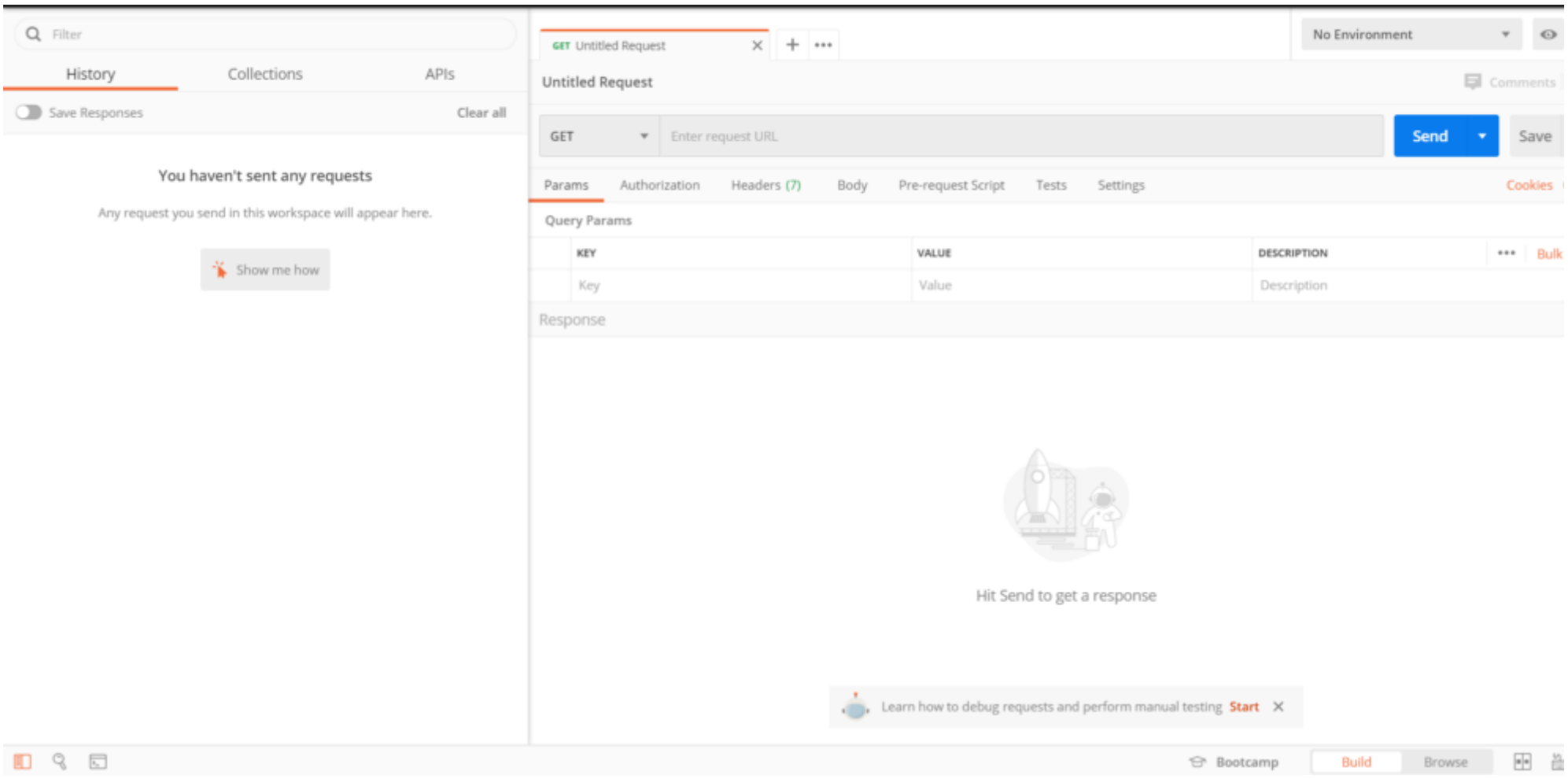
- GET, kegiatan menerima data.
  contoh: `GET https://situsweb/api/users/detail/1` => mengambil data detail user dengan id = 1.
- POST, kegiatan pengiriman/input data untuk diproses selanjutnya.
  contoh: `POST https://situsweb/api/users` => mengirim data user.
- PUT, kegiatan mengubah data.
  contoh `PUT https://situsweb/api/users/update/1` => melakukan update pada data user dengan id = 1.
- DELETE, kegiatan menghapus data.
  contoh: `DELETE https://situsweb/api/users/delete/1` => menghapus data user dengan id = 1.

Perlu diperhatikan bahwa tidak semua API dapat diakses secara langsung, dan memerlukan Authentication. Dalam hal ini terkadang perlu dilakukan registrasi/pendaftaran pada situs penyedia API tersebut. Setelah proses pendaftaran, biasanya akan diperoleh kode seperti API key, atau OAUTH token, atau bisa juga dalam bentuk client_id dan client_secret. Kode tersebut harus disertakan setiap kali melakukan request pada situs penyedia API tersebut.

Postman

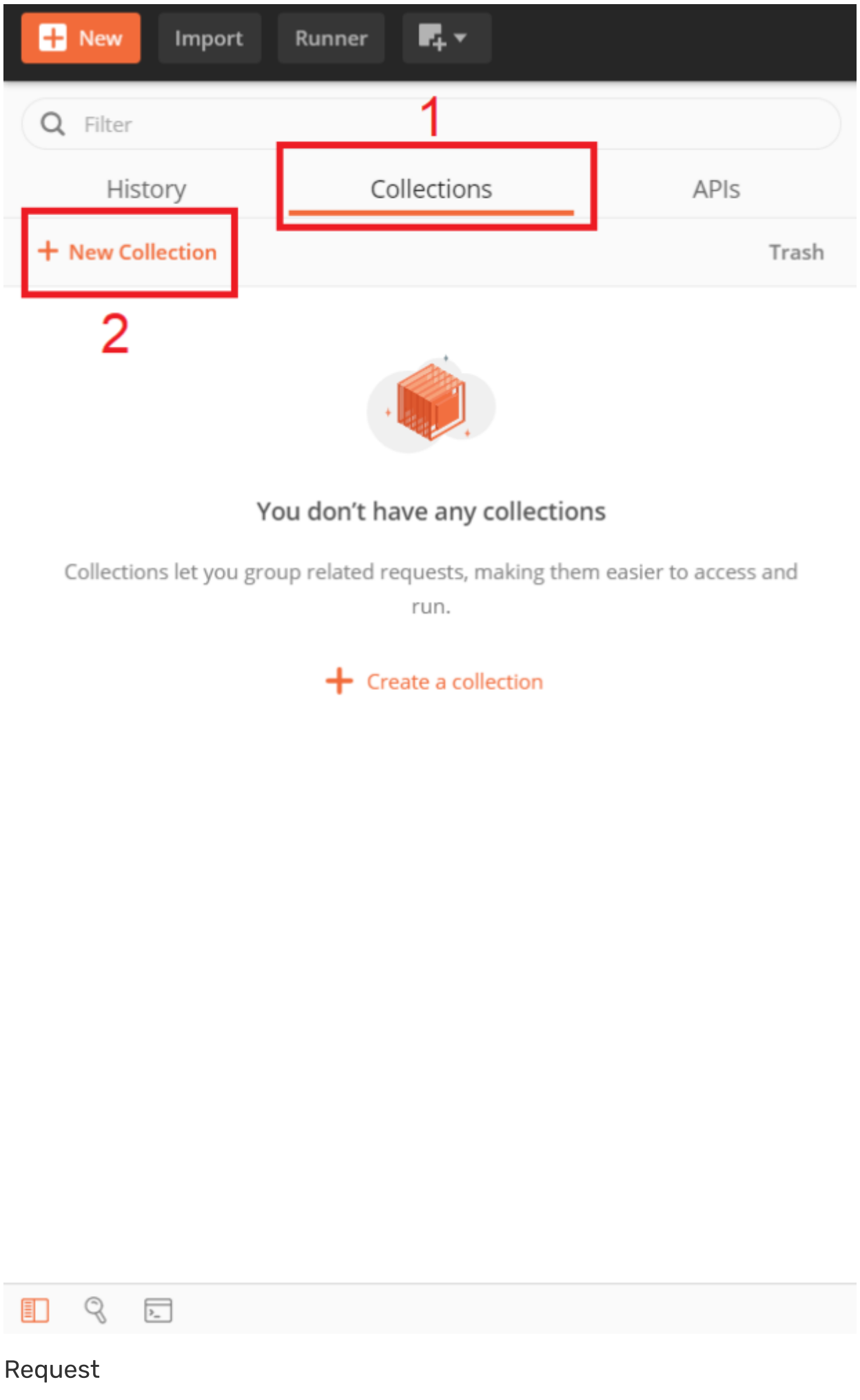
Postman merupakan salah satu aplikasi terpopuler dalam menangani API. Untuk menggunakannya Anda bisa terlebih dahulu mendownload nya di link berikut: https://www.postman.com/downloads/. Setelah mendownload aplikasi Postman tersebut, lakukan instalasi dan kemudian lakukan Sign Up (bisa dilakukan melalui aplikasi atau melalui web berikut: https://identity.getpostman.com/signup).

Setelah Sign Up dan berhasil masuk ke dalam aplikasi Postman, berikut tampilan yang akan Anda temukan:



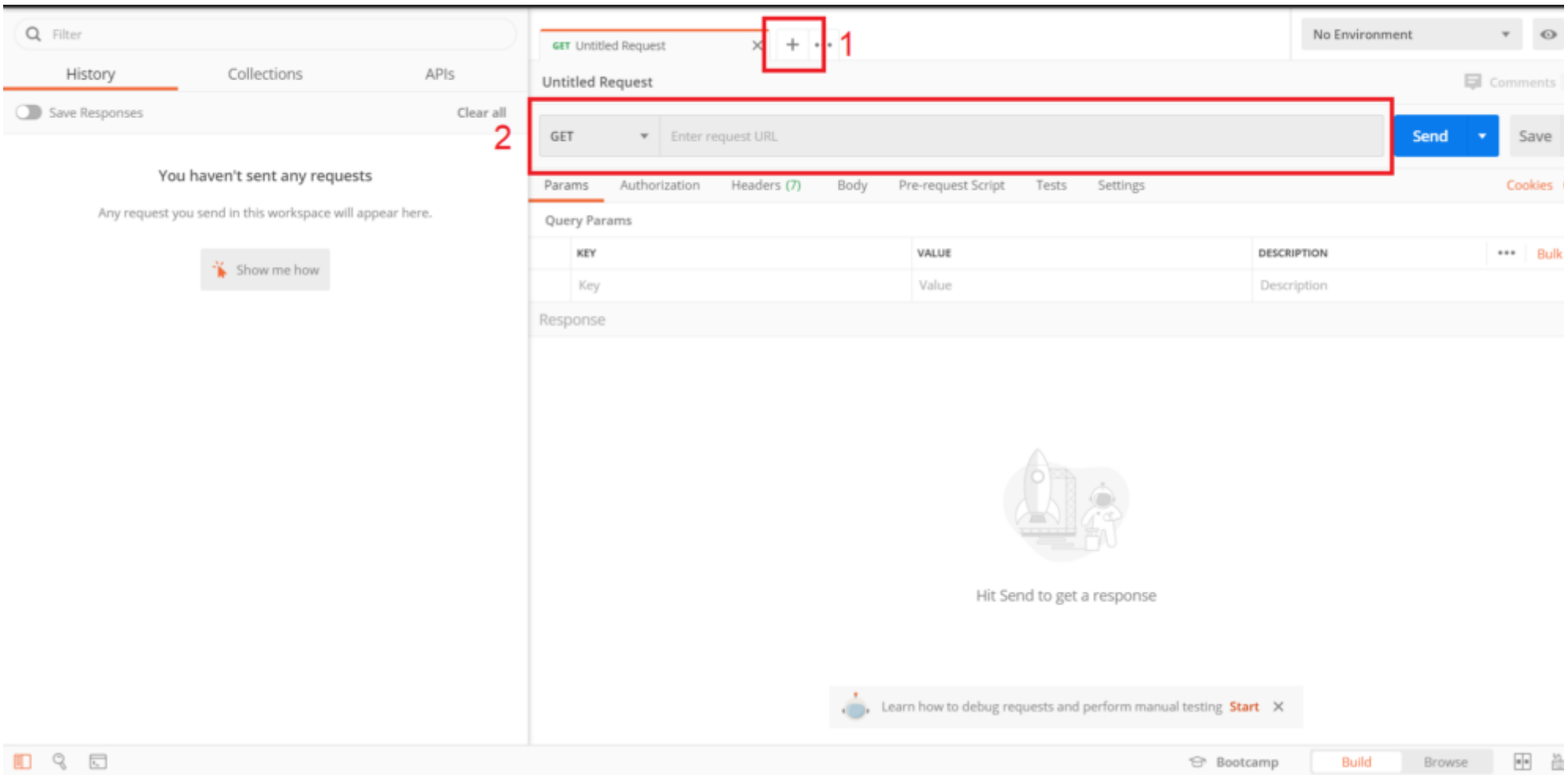
Collection

Disarankan untuk mengelompokkan API yang akan digunakan pada satu Collection, dengan cara memilih menu "Collection" pada sidebar dan klik "+ New Collection" kemudian memasukkan nama Collection tersebut, bisa menggunakan nama Project maupun nama web / sumber API nya.



Request

Selanjutnya Anda bisa menambahkan request API dengan menekan tombol "+" pada menu yang tersedia (1) . Selanjutnya Anda bisa memasukkan URL yang dituju pada bagian yang telah disediakan (2). Pada Postman telah disediakan berbagai method yang dapat Anda temukan pada sebelah kiri dari kolom input URL, mulai dari GET, POST, PUT, DELETE dan lainnya.



Dibawah kolom input URL, Anda akan menemukan menu/tab "Params", "Authentication", "Header", "Body", dan lainnya.

- Menu "Params" yang dapat digunakan untuk menambah query yang akan membuat URL yang digunakan bertambah panjang (biasanya diawali dengan "?").
- Pada menu "Authentication", Anda akan menemukan metode Auth yang terkadang diperlukan untuk dapat mengakses suatu API (umumnya menggunakan API Key, Bearer Token, atau OAuth tergantung sumber/penyedia API).
- Pada menu "Header", Anda juga dapat mengatur Authentication dengan menggunakan key "Authentication" dan value dari Authentication yang diperlukan (misal bearer {token}). Pada bagian ini biasanya dimasukkan informasi tambahan yang digunakan oleh back-end/server untuk mengatur bagaimana cara response akan diberikan.
- Menu "Body", yang biasa diperlukan ketika menggunakan method "POST" atau mengirim suatu form kepada back-end/server.

Rating - Feedback

Berikan Rating pada posting ini:

0

Berikan kritik dan saran..

Submit