Rakamin Academy

mandiri

Project Based Internship

API

Principles of API, Basic Operation of API, and Restful API

Daftar Isi

A. Principles of API

API adalah singkatan dari Application Programming Interface yang bisa digunakan antar aplikasi untuk saling berkomunikasi satu dengan yang lainnya. Fungsi API adalah untuk bertukar data antar aplikasi tanpa harus terhubung secara langsung.

Sebagai contoh kita ingin membuat suatu aplikasi tapi untuk memudahkan usernya mendaftar / login kamu bisa menggunakan "API" yang disediakan oleh twitter/facebook/gmail untuk loginnya, mereka bisa mendaftar di website kamu seperti biasa, tanpa membuat akun baru.

Sederhananya, terdiri dari dua sisi, 1. Penyedia API dan 2. Pengguna API Penyedia API biasanya memberi akses untuk aplikasi lain menggunakan fitur mereka, seperti facebook menyediakan API login, Instagram menyediakan API untuk mencari foto, jadi website lain bisa membuat aplikasi yang berkaitan dengan foto tapi mengambil data-data fotonya dari instagram.

Fitur apa yang disediakan itu terserah penyedia API nya, bahasa program yang digunakan pun bebas. Pengguna API tidak perlu tahu bahasa apa yang dipakai, saat ini mayoritas API berbicara dengan format 'json' artinya penyedia API akan melempar data-datanya dalam format JSON, dan pengguna API bisa membaca format JSON ini juga.

B. REST API

REST merupakan kependakan dari REpresentational State Transfer yang merupakan standar arsitektur berbasis web yang menggunakan protokol HTTP untuk berkomunikasi data. REST adalah salah satu

implementasi dari web service sebagai sebuah standar yang digunakan untuk pertukaran data antar aplikasi atau sistem. Jadi bisa dikatakan REST ini adalah salah satu desain arsitektur untuk membuat API.

Cara kerja REST API adalah REST client akan mengakses data/resource ke REST server dimana masing-masing resource atau data tersebut dibedakan oleh sebuah global ID atau URIs (Universal Resource Identifiers). Data yang diberikan oleh REST server itu bisa berupa format text, JSON atau XML. Yang paling populer dipakai saat ini adalah format JSON karena lebih mudah untuk dibaca secara umum.

Flow dari sebuah proses REST contohnya:

Pertama harus ada sebuah REST server yang akan menyediakan resource/data. Sebuah REST client akan membuat HTTP request ke server melalui sebuah global ID atau URIs dan server akan merespon dengan mengirimkan balik sebuah HTTP response sesuai yang diminta client.

C. Komponen HTTP Request & HTTP Response

Komponen dari HTTP request yaitu sebagai berikut :

- HTTP method seperti GET, POST, PUT, DELETE dll sesuai dengan tugasnya masing-masing
- URI untuk mengetahui lokasi data di server
- HTTP Version, seperti HTTP v1.1
- Request Header, berisi metadata seperti Authorization, tipe client dan lain
- Request Body, data yang diberikan client ke server seperti URI params

Berikut ini adalah empat metode HTTP umum:

1. **GET**

   Klien menggunakan GET untuk mengakses sumber daya yang berada di URL yang ditentukan pada server. Mereka dapat menyimpan permintaan GET dan mengirim parameter dalam permintaan API RESTful untuk menginstruksikan server memfilter data sebelum mengirim.

2. **POST**

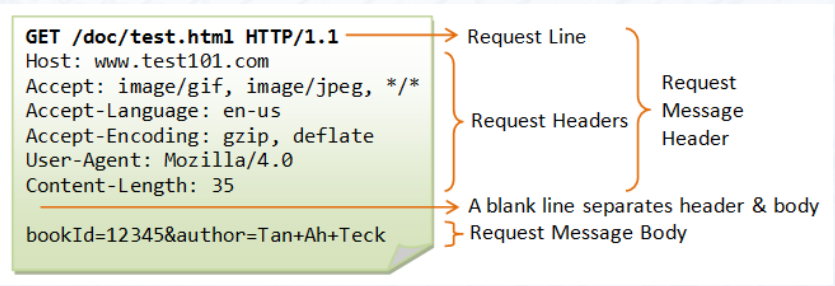
   Klien menggunakan POST untuk mengirim data ke server. Mereka menyertakan representasi data dengan permintaan. Mengirim permintaan POST yang sama beberapa kali memiliki efek samping seperti membuat sumber daya yang sama beberapa kali.

3. **PUT**

   Klien menggunakan PUT untuk memperbarui sumber daya yang ada di server. Tidak seperti POST, mengirimkan permintaan PUT yang sama beberapa kali dalam layanan web RESTful memberikan hasil yang sama.

4. **DELETE**

   Klien menggunakan permintaan DELETE untuk menghapus sumber daya. Permintaan DELETE dapat mengubah status server. Namun, jika pengguna tidak memiliki autentikasi yang sesuai, permintaan akan gagal.



Contoh HTTP Request

Komponen dari HTTP response yaitu sebagai berikut :

- Response Code, status server terhadap request yang diminta seperti 200, 401, 404 dan lainnya.
- HTTP Version
- Response Header yang berisi metadata seperti content type, cache tag dan yang lainnya.
- Response Body, data/resource yang diberikan oleh server baik itu berupa text, json ataupun xml

D. RESTful API

RESTful adalah sebuah sistem yang mengimplementasikan REST api di dalamnya. Jadi RESTful ini lebih kepada sebuah aplikasi yang mengimplementasikan REST API sehingga bisa dikatakan sebagai sebuah aplikasi yang RESTful.

Fungsi dasar API RESTful sama dengan penjelajahan internet. Klien menghubungi server dengan menggunakan API saat meminta sumber daya. Developer API menjelaskan kepada klien cara untuk menggunakan API REST dalam dokumentasi API aplikasi server. Ini adalah langkah umum untuk semua panggilan API REST:

1. Klien mengirimkan permintaan ke server. Klien mengikuti dokumentasi API untuk memformat permintaan dalam format yang dipahami oleh server.
2. Server mengautentikasi klien dan mengonfirmasi bahwa klien memiliki hak untuk membuat permintaan.

3. Server menerima permintaan dan memproses secara internal.
4. Server mengembalikan respon kepada klien. Respons berisi informasi yang memberitahu klien jika permintaannya berhasil. Respons juga termasuk informasi apa saja yang diminta klien.

Permintaan API REST dan detail respons sedikit berbeda tergantung pada cara developer API merancang API.

E. Metode Autentikasi API RESTful

Layanan web RESTful harus mengautentikasi permintaan sebelum dapat mengirim respons. Autentikasi adalah proses verifikasi identitas. Contohnya, Anda dapat membuktikan identitas Anda dengan menunjukkan KTP atau SIM. Demikian halnya, klien layanan RESTful harus membuktikan identitas mereka ke server untuk membangun kepercayaan.

RESTful API memiliki empat metode autentikasi umum:

1. Autentikasi HTTP

HTTP menentukan beberapa skema autentikasi yang dapat Anda gunakan secara langsung saat menerapkan API REST. Berikut ini adalah dua skema tersebut :

a. Autentikasi dasar

Dalam autentikasi dasar, klien mengirimkan nama pengguna dan kata sandi di header permintaan. Header mengodekannya dengan base64, yaitu teknik pengkodean yang mengonversi pasangan menjadi satu set 64 karakter untuk transmisi yang aman.

b. Autentikasi pembawa

Istilah autentikasi pembawa mengacu pada proses pemberian kontrol akses kepada pembawa token. Token pembawa biasanya berupa deretan karakter terenkripsi yang dihasilkan server sebagai respons atas permintaan masuk. Klien mengirimkan token di header permintaan untuk mengakses sumber daya.

2. Kunci API

Kunci API adalah opsi lain untuk autentikasi API REST. Dalam pendekatan ini, server memberikan nilai unik yang dihasilkan ke klien pertama kali. Setiap kali klien mencoba mengakses sumber daya, API menggunakan kunci API unik untuk memverifikasi dirinya sendiri. Kunci API kurang aman karena klien harus mentransmisikan kunci, yang membuatnya rentan terhadap pencurian jaringan.

3. OAuth

OAuth menggabungkan kata sandi dan token untuk akses masuk yang sangat aman ke sistem apa pun. Server pertama-tama meminta kata sandi lalu meminta token tambahan untuk menyelesaikan proses otorisasi. Server dapat memeriksa token kapan saja dan juga dari waktu ke waktu dengan cakupan dan periode tertentu.

F. Retrofit

Retrofit adalah library REST-client yang sangat aman untuk Java dan Android. Dalam aplikasi android, Retrofit memanfaatkan OkHttp yang merupakan level rendah.

Retrofit memiliki tiga komponen utama di dalamnya:

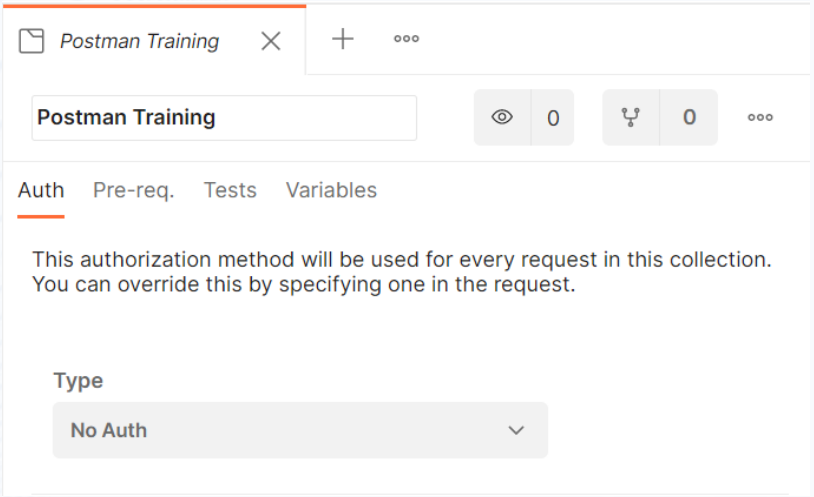
1. **Database** : Objek kelas yang dipakai untuk mendaftarkan kelas Data Access Object di mana ia nantinya berfungsi sebagai titik akses utama ke remote database atau REST API.
2. **Model** : Kelas yang dipakai untuk menyimpan suatu nilai atau data. Retrofit membutuhkan model kelas yang dipakai menyimpan data respon yang didapat berdasarkan REST API.
3. **Data Access Object (DAO)** : Antarmuka (interface) yang dipakai retrofit buat akses data berdasarkan group client ke remote database (REST API) dan memuat metode yg pada pakai buat mengakses resource-nya.

G. Test API dengan Postman

Membuat Collection

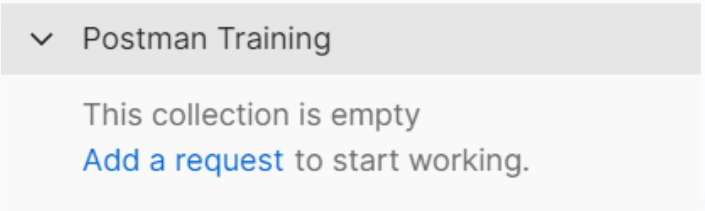
Pada aplikasi Postman, sebuah collection dapat dibuat untuk menampung berbagai request yang dilakukan pada aplikasi API yang telah dibuat. Selain itu, dengan menggunakan collection sebuah spesifikasi pada sebuah request dapat ditentukan seperti jenis autentikasi yang digunakan dan variabel yang digunakan untuk request tertentu.

Untuk membuat collection baru, tekan tombol plus lalu isi nama collection dengan Postman Training atau dengan nama lain.

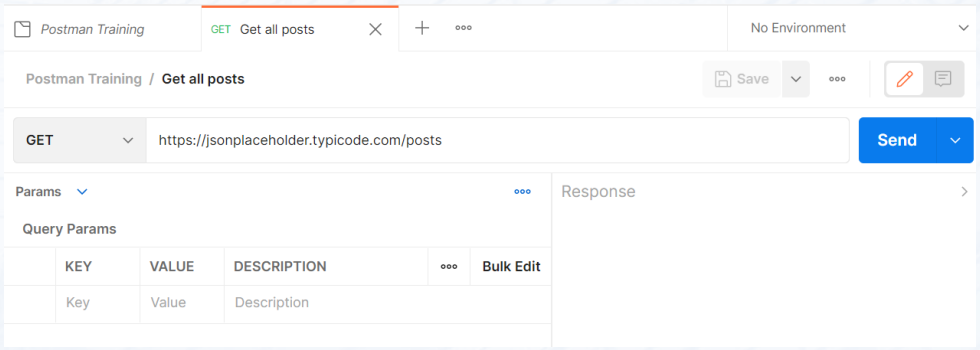


Melakukan Request

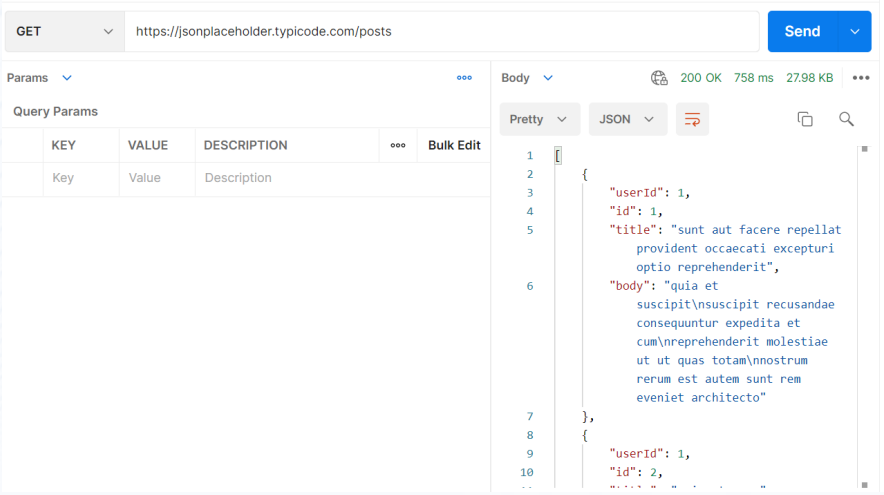
Untuk membuat request baru, pilih Add a request.



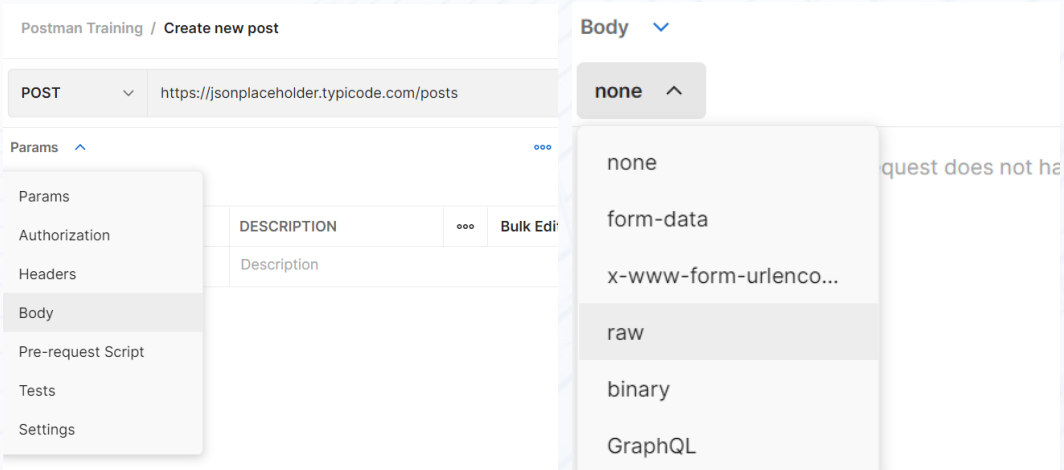
Untuk keperluan tutorial. API yang digunakan adalah berupa JSON Placeholder. Isi nama request dengan Get all posts untuk mendapatkan seluruh data post. Untuk method yang digunakan adalah method GET dengan alamat url https://jsonplaceholder.typicode.com/posts.

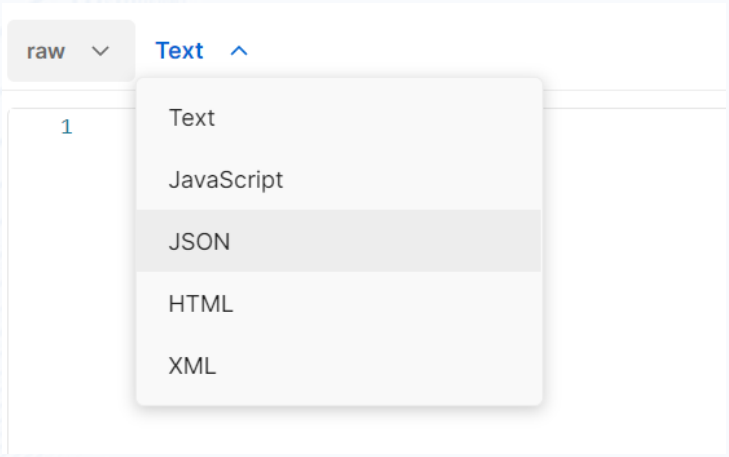


Tekan tombol Send atau gunakan keyboard Enter untuk mengirim request. Jika request berhasil maka seluruh data post berhasil didapatkan.



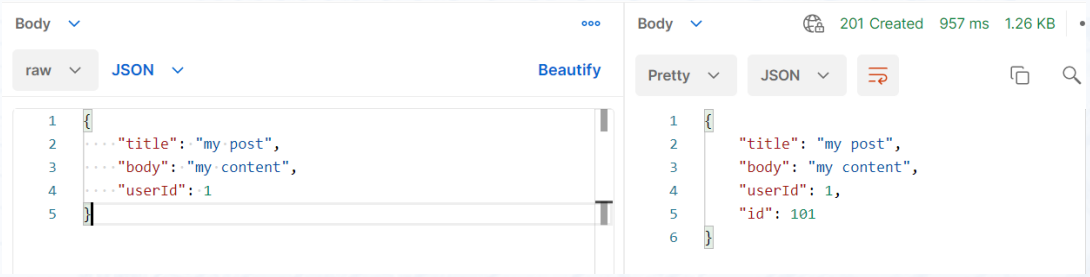
Sebuah request untuk method lain seperti method POST juga dapat digunakan. Buat request baru dengan nama Create new post pada collection Postman Training menggunakan method POST dengan alamat url https://jsonplaceholder.typicode.com/posts. Pada bagian dropdown Params, pilih Body untuk mengisi data post yang akan ditambahkan. Jenis body yang dipilih adalah raw lalu pilih JSON.





Pada bagian body, isi dengan data berikut.

Kirim request dengan menekan tombol Enter atau klik Send. Jika berhasil maka data post ditampilkan.



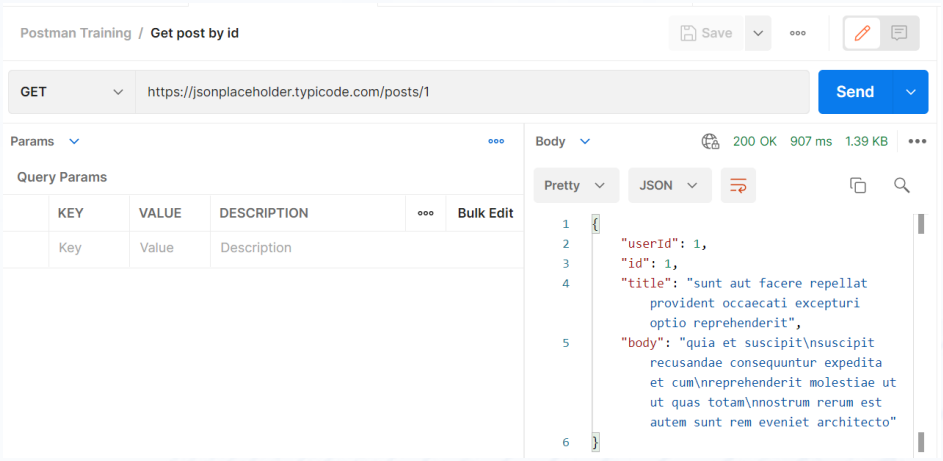
Jenis body pada request dapat bermacam-macam seperti form-data, GraphQL dan lainnya.

Membuat Variabel

Pada aplikasi Postman, sebuah variabel dapat digunakan untuk menyimpan suatu nilai yang nantinya dapat digunakan pada request lainnya. Sebuah variabel dapat dibuat di dalam collection yang nantinya dapat digunakan pada request yang ada di dalam collection tersebut.

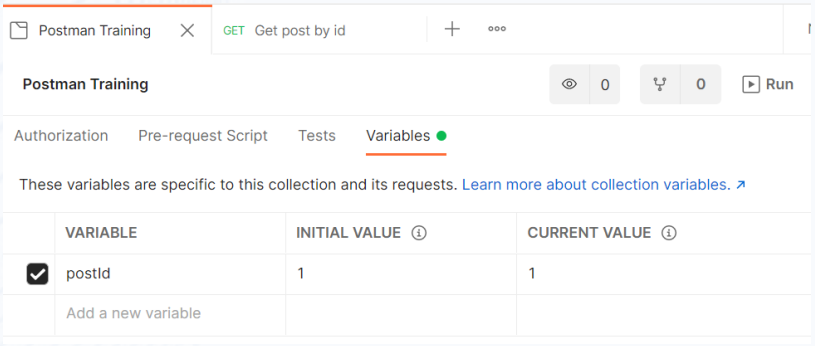
Sebelum membuat variabel, sebuah request baru dengan nama Get post by id dibuat di dalam collection Postman Training. Method yang digunakan adalah method GET dengan alamat url https://jsonplaceholder.typicode.com/posts/1.

Jika dieksekusi maka respons dari request tersebut menjadi seperti ini.



Pada alamat url tersebut, terdapat id yang dapat disimpan di dalam variabel sehingga variabel dapat digunakan untuk request lain yang memerlukan id seperti request untuk menghapus data post. Untuk membuat variabel di Postman, buka collection lalu pilih menu Variables.

Pada menu tersebut, tambahkan variabel baru dengan nama postId dengan initial value dan current value adalah 1. Lalu simpan hasil perubahan dengan Ctrl + S.



Untuk mengakses variabel yang dibuat di Postman, gunakan sintaks berikut.

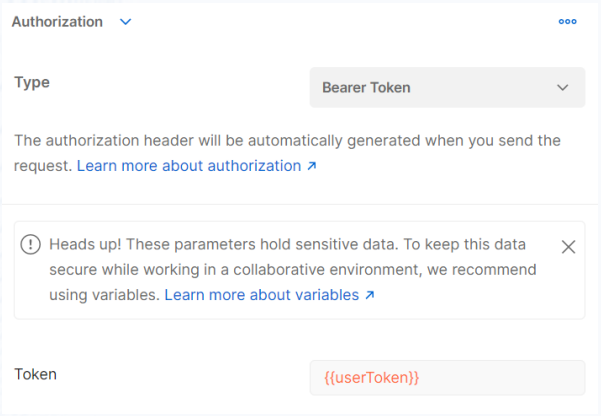
```
{{nama_variabel}}
```

Menambahkan Header untuk Autentikasi

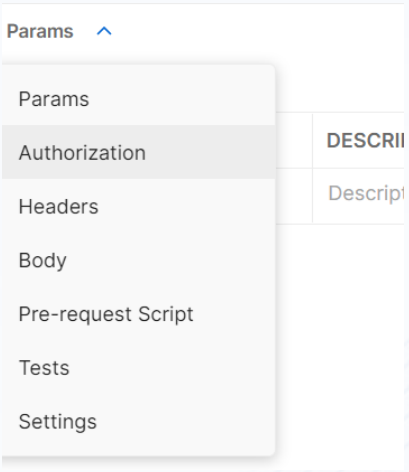
Jika terdapat sebuah request yang membutuhkan autentikasi maka header untuk autentikasi dapat ditambahkan di request pada bagian Authorization.

Sebagai contoh, sebuah request membutuhkan autentikasi dengan jenis Bearer Token maka dapat ditambahkan bi bagian Authorization dengan jenis Bearer Token. Sebaiknya, token yang digunakan untuk menguji request disimpan di dalam variabel agar lebih aman dan mudah digunakan di request lain.

Request ini membutuhkan bearer token yang nilai tokennya disimpan di variabel userToken.



Menu Authorization dapat dipilih di dropdown Params.



Data autentikasi seperti Bearer Token, API key dan lainnya lebih baik disimpan di dalam variabel.

References

https://ngide.net/apa-itu-rest-api

https://sekolahkoding.com/belajar/api

https://aws.amazon.com/id/what-is/restful-api/

https://guides.codepath.com/android/consuming-apis-with-retrofit

https://kotakode.com/blogs/16235/Panduan-Singkat-Menggunakan-Postman