

Latihan Membuat Permintaan HTTP (HTTP Request)

 dicoding.com/academies/261/tutorials/14152

Lantas seperti apa sih sebenarnya bentuk request dan respons pada HTTP? Penasaran? Mari kita coba membuat request pada web service kedai kopi melalui cURL. Silakan buka CMD atau Terminal pada komputer Anda.

cURL atau Client URL merupakan software berbasis *command line* yang dapat melakukan transaksi data melalui beberapa protokol internet, salah satunya HTTP/S. cURL dapat diakses secara langsung tanpa proses *install* melalui Terminal (Linux dan Mac) atau CMD (Windows).[4]

Kita akan melakukan tiga skenario berikut:

- Meminta daftar kopi tersedia.
- Membeli kopi yang tersedia.
- Membeli kopi yang tidak tersedia.

Masuk ke skenario pertama, buatlah request untuk mendapatkan daftar kopi yang tersedia, tuliskan kode berikut pada CMD atau Terminal Anda.

```
1. curl -X GET https://coffee-api.dicoding.dev/coffees -i
```

Kita bedah kodenya yuk:

- **curl** : merupakan perintah untuk menggunakan program cURL pada Terminal atau CMD.
- **-X GET** : menetapkan HTTP method/verb yang kita gunakan. GET berarti kita ingin mendapatkan sebuah data.
- <https://coffee-api.dicoding.dev/coffees> : merupakan alamat request yang dituju.
- **-i** : memberikan informasi detail terhadap response yang diberikan (HTTP response headers).

Setelah menuliskan kode tersebut, tekan enter. Anda akan mendapatkan respons dari web server seperti ini:

```
1. HTTP/1.1 200 OK
2. Server: nginx/1.18.0 (Ubuntu)
3. Date: Mon, 01 Nov 2021 07:41:52 GMT
4. Content-Type: application/json; charset=utf-8
5. Content-Length: 188
6. Connection: keep-alive
7. content-security-policy: upgrade-insecure-requests
8. referrer-policy: strict-origin-when-cross-origin
9. x-frame-options: DENY
10. x-content-type-options: nosniff
11. x-xss-protection: 1; mode=block
12. vary: origin
13. access-control-expose-headers: WWW-Authenticate, Server-Authorization
14. cache-control: no-cache
15. accept-ranges: bytes
16. {"message":"Berikut daftar kopi yang tersedia","coffees":
    [{"id":1,"name":"Kopi Tubruk","price":12000}, {"id":2,"name":"Kopi
    Tarik","price":15000}, {"id":3,"name":"Kopi Jawa","price":18000}]}
```

Voila! Anda berhasil mendapatkan response pertama dari server. Fokus terhadap kode yang ditebalkan yah. Mari kita bedah sekarang.

- **HTTP/1.1** : merupakan HTTP version yang digunakan oleh web server dalam menanggapi permintaan.
- **200** : merupakan status code dari request. Karena status code diawali dengan angka 2, berarti request kita berhasil dilakukan.
- **OK** : merupakan pesan teks dari status code, 200 berarti "OK".
- **Content-Type: application/json** : merupakan tipe konten yang digunakan web server dalam memberikan data. Karena nilainya application/json, itu berarti server menggunakan format json.
- **JSON Data (kode di bagian bawah)** : merupakan data yang diberikan oleh web server. Kita bisa melihat web server memberikan informasi kopi yang tersedia beserta harganya menggunakan format JSON.

Skenario pertama selesai! Mudah bukan untuk melakukan request melalui protokol HTTP?

Lanjut ke skenario kedua yuk. Buat permintaan membeli kopi yang tersedia dengan menuliskan perintah berikut:

```
1. curl -X POST -H "Content-Type: application/json" -d '{"name": "Kopi
    Tubruk"}' https://coffee-api.dicoding.dev/transactions -i
```

Jika dilihat, kode yang dituliskan memiliki struktur yang sama, namun ada beberapa perbedaan. Mari kita bedah:

- **-X POST** : dalam request kali ini kita menggunakan method POST. Karena membeli bukan hanya meminta data, tapi akan mengubah jumlah stok kopi yang ada. Selain itu kita juga melampirkan data berupa kopi apa yang akan dipesan. Sehingga tidak masuk akal bila kita menggunakan GET request.
- **-H "Content-Type: application/json"** : Menetapkan nilai "Content-Type: application/json" pada Header request. Fungsinya untuk memberitahu server bahwa kita melampirkan data dalam bentuk JSON.
- **-d <JSON Content>** : merupakan data yang dilampirkan pada request. Data ini berformat JSON dan memiliki informasi kopi apa yang ingin dipesan.
- <https://coffee-api.dicoding.dev/transactions> : Merupakan alamat request yang dituju untuk membeli kopi.

Setelah menuliskan perintah di atas, silakan tekan enter. Anda akan mendapatkan respons seperti ini dari web server:

```
1. HTTP/1.1 201 Created
2. Server: nginx/1.18.0 (Ubuntu)
3. Date: Mon, 01 Nov 2021 07:44:29 GMT
4. Content-Type: application/json; charset=utf-8
5. Content-Length: 46
6. Connection: keep-alive
7. content-security-policy: upgrade-insecure-requests
8. referrer-policy: strict-origin-when-cross-origin
9. x-frame-options: DENY
10. x-content-type-options: nosniff
11. x-xss-protection: 1; mode=block
12. vary: origin
13. access-control-expose-headers: WWW-Authenticate, Server-Authorization
14. cache-control: no-cache
15. {"message":"Pesanan berhasil!","success":true}
```

Fokus terhadap kode yang diberi tanda tebal. Anda sudah bisa membaca hasilnya kan? Yups, Pesanan kopi berhasil!

Lanjut ke skenario terakhir, yakni membeli kopi yang tidak tersedia. Tuliskan perintah yang sama seperti sebelumnya. Namun dengan tipe kopi yang tentunya tidak tersedia pada daftar. Contohnya Kopi Luwak.

```
1. curl -X POST -H "Content-Type: application/json" -d '{"name": "Kopi Luwak"}' https://coffee-api.dicoding.dev/transactions -i
```

Silakan tulis perintahnya, kemudian tekan enter. Kali ini Anda akan mendapatkan response seperti ini.

1. **HTTP/1.1 404 Not Found**
2. Server: nginx/1.18.0 (Ubuntu)
3. Date: Mon, 01 Nov 2021 07:45:23 GMT
4. **Content-Type: application/json; charset=utf-8**
5. Content-Length: 66
6. Connection: keep-alive
7. content-security-policy: upgrade-insecure-requests
8. referrer-policy: strict-origin-when-cross-origin
9. x-frame-options: DENY
10. x-content-type-options: nosniff
11. x-xss-protection: 1; mode=block
12. vary: origin
13. access-control-expose-headers: WWW-Authenticate, Server-Authorization
14. cache-control: no-cache
15. {"message":"Pesanan gagal, kopi tidak ditemukan!","success":false}

Fokus pada kode yang ditebalkan. Lihat status code-nya, kali ini Anda mendapatkan respons negatif */ho!* Request yang Anda lakukan tidak dapat diproses oleh server karena kopi luwak tidak ada (*not found*) pada daftar kopi.

Nah melalui latihan tadi, semoga Anda semakin mengerti yah bagaimana client dan server berkomunikasi melalui protokol HTTP.

ini konten buat note nanti