NAMA : Ilham Faishal Seto Aji

NIM : 22.01.53.0041

MATKUL : Web Service

Rangkuman Request Methods Http

Request methods atau metode permintaan adalah serangkaian instruksi yang digunakan dalam protokol HTTP (Hypertext Transfer Protocol) untuk menentukan tindakan apa yang diinginkan oleh klien (seperti browser atau aplikasi) ketika berkomunikasi dengan server. Setiap metode HTTP mewakili jenis operasi yang akan dilakukan terhadap sumber daya yang ada di server, seperti mengambil, mengirim, memperbarui, atau menghapus data.

Metode HTTP ini memungkinkan klien dan server untuk berinteraksi secara efektif dalam sistem berbasis web, dan mereka biasanya digunakan dalam konteks API (Application Programming Interface), web service, serta interaksi browser-server.

Berikut merupakan beberapa metode dalam http:

1. GET

GET adalah salah satu metode request HTTP yang digunakan untuk mengambil atau membaca data dari server. Metode ini biasanya digunakan untuk meminta informasi dari server tanpa menyebabkan perubahan pada data atau status server. Berikut penjelasan GET :

* Mengambil data

Get diperuntukan untuk mengambil data atau informasi yang sudah ada diserver, hal ini tidak mengubah data deserver.

* Query string di URL

Data dikirim dalam bentuk query string



Pada contoh ini adalah mengambil data color=red menggunakan GET, karena parameter dikirim menggunakan URL jadi tidak boleh salah karena hal ini sangat sensitive

* Idempotent

GET bersifat Idempotent adalah permintaan yang sama dalam berulang kali tanpa memberikan efek pada server karena tidak mengubah data tersebut.

* Tidak memiliki BODY

Semua data yang ingin dikirimkan ke server (misalnya parameter pencarian atau filter) disertakan di URL sebagai query string.

* Keamanan dan ukuran data

menggunakan URL untuk mengirimkan data, ukuran data yang dikirimkan terbatas oleh batas panjang URL, yang biasanya dibatasi oleh browser (sekitar 2000 karakter). GET tidak cocok untuk mengirim data sensitive seperti password.

* Caching

Respon GET sering di-cache oleh browser atau server, sehingga dapat meningkatkan performa untuk permintaan yang berulang. Data yang sering diakses seperti halaman statis atau gambar cenderung menggunakan GET karena bisa disimpan sementara untuk mempercepat akses berikutnya.

1. POST

Metode POST dalam HTTP digunakan untuk mengirimkan data ke server untuk membuat atau memperbarui sumber daya. Ini adalah salah satu metode request yang paling umum digunakan saat berinteraksi dengan server, terutama saat melakukan operasi yang memerlukan pengiriman data. Berikut penjelasan POST:

* Kegunaan utama

Membuat sumber daya baru, contohnya pengisian data diri dan akhirnya dikirim keserver. POST dapatv mengirim data sensitive seperti password dan data diri.

* Struktur request POST

URL Seperti request GET, POST juga menggunakan URL, tetapi data tidak dikirim sebagai bagian dari URL. Body Data dikirim melalui body request. Ini memungkinkan pengiriman data yang lebih besar dan lebih kompleks, seperti form, file, atau JSON. Header Mengandung informasi tambahan, seperti tipe konten (misalnya, Content-Type: application/json atau application/x-www-form-urlencoded), yang memberi tahu server bagaimana data dalam body harus diproses.

* Keamanan

Data tidak terlihat di URL Tidak seperti metode GET di mana data dikirim melalui query string dalam URL, POST menyembunyikan data di body, membuatnya lebih aman untuk transmisi informasi sensitif. Namun, tidak terenkripsi secara otomatis POST tidak menjamin keamanan data kecuali digunakan dengan HTTPS, yang mengenkripsi seluruh request termasuk body.

* Respon
* 201 Created: Jika POST berhasil membuat resource baru, server biasanya akan mengembalikan status ini.
* 200 OK atau 204 No Content: Jika POST digunakan untuk operasi lain yang tidak memerlukan pembuatan resource baru.
* Server juga dapat mengembalikan data dalam body respons, seperti ID dari resource yang baru dibuat atau pesan konfirmasi.

1. PUT

Metode PUT adalah salah satu metode dalam protokol HTTP yang digunakan untuk memperbarui atau mengganti data yang sudah ada di server. PUT bersifat idempotent, artinya, tidak peduli berapa kali permintaan PUT dikirim, hasil akhirnya akan selalu sama, yaitu resource yang diperbarui. Berikut penjelasan Put :

* Memperbarui data

PUT digunakan untuk mengganti seluruh data dari resource yang ada. Misalnya, jika Anda memiliki data pengguna, PUT akan mengganti seluruh informasi pengguna yang ada dengan data baru yang dikirim dalam body request.

* Membuat resource

Dalam beberapa kasus, PUT juga dapat digunakan untuk membuat resource baru jika resource yang dirujuk belum ada. Misalnya, jika Anda mengirim PUT ke endpoint yang belum memiliki resource dengan ID tertentu, server dapat membuat resource baru berdasarkan data yang dikirim.

* Idempotent

PUT bersifat idempotent, yang berarti bahwa jika permintaan yang sama dilakukan berkali-kali, hasilnya akan tetap sama. Ini berbeda dengan POST, yang dapat membuat beberapa sumber daya baru jika dilakukan berkali-kali.

* Mengganti resource

Saat menggunakan PUT, seluruh resource yang diwakili oleh URI (Uniform Resource Identifier) akan diganti. Jika ada data yang tidak disertakan dalam body request, data tersebut akan dihapus. Oleh karena itu, PUT tidak digunakan untuk pembaruan sebagian; itu adalah tugas dari metode PATCH.

1. DELETE

Metode DELETE dalam HTTP digunakan untuk menghapus sumber daya yang ada di server. Berikut penjelasan lebih lanjut mengenai metode ini:

* Jika request untuk menghapus maka akan ada respon status http yang menunjukan berhasil atau tidaknya.
* Idempotent

Metode DELETE bersifat idempotent, artinya jika operasi ini dilakukan beberapa kali dengan permintaan yang sama, hasilnya akan tetap sama (jika resource sudah dihapus, melakukan DELETE lagi tidak akan memberikan efek tambahan).

* Respon

Jika sukses, server sering kali mengembalikan status HTTP 200 (OK), 202 (Accepted), atau 204 (No Content). Jika resource tidak ditemukan, biasanya mengembalikan status 404 (Not Found).

* Body

Umumnya, request DELETE tidak memiliki body, karena URI sudah mengindikasikan resource yang akan dihapus. Namun, beberapa server memungkinkan body dalam permintaan DELETE untuk spesifik kasus tertentu.

1. OPTIONS

Metode OPTIONS dalam HTTP digunakan untuk memeriksa metode HTTP apa saja yang didukung oleh server pada resource tertentu. Request OPTIONS tidak mengubah data atau resource apa pun di server, melainkan hanya meminta informasi tentang kapabilitas server terhadap resource yang diminta.

* Tujuan

Mengetahui metode HTTP apa yang bisa dilakukan terhadap resource tertentu. Misalnya, server dapat merespons bahwa resource mendukung GET, POST, atau PUT.

* Respon

Server merespons dengan mengirimkan daftar metode yang diizinkan dalam header Allow. Tidak ada body konten dalam respons, hanya header yang menyatakan metode yang didukung.



* Penggunaan

Preflight Request dalam CORS (Cross-Origin Resource Sharing): Browser menggunakan request OPTIONS untuk memeriksa apakah permintaan lintas domain (cross-origin) diizinkan oleh server. Preflight request ini dilakukan sebelum request utama (seperti POST atau PUT) untuk menentukan izin.

Client-Server Compatibility Check: Digunakan oleh klien untuk mengetahui kemampuan server dan memastikan mereka mengirim permintaan yang benar.

* Secara keseluruhan, metode OPTIONS adalah alat yang berguna untuk interaksi antara klien dan server, terutama dalam konteks interoperabilitas dan kontrol akses lintas domain.

1. PATCH

Metode PATCH dalam HTTP digunakan untuk memperbarui sebagian data dari suatu sumber daya (resource) yang ada di server. Berbeda dengan PUT, yang menggantikan keseluruhan sumber daya dengan data baru, PATCH hanya memodifikasi atau memperbarui bagian-bagian tertentu dari sumber daya tersebut. Metode PATCH ideal untuk aplikasi yang membutuhkan modifikasi data dinamis atau granular tanpa harus menggantikan seluruh sumber daya.

* Partial Update: PATCH memungkinkan pengiriman data yang hanya perlu diubah, bukan seluruh data sumber daya.
* Idempotent: Dalam beberapa kasus, PATCH dianggap tidak selalu idempotent, tetapi jika server diimplementasikan dengan benar, itu bisa diidempotent.
* Lebih Hemat Bandwidth: Karena hanya data yang diperbarui yang dikirim, metode ini lebih hemat dibanding PUT untuk memperbarui sebagian kecil data.
* Flexible: Dapat digunakan untuk perubahan yang kompleks dan lebih spesifik, misalnya untuk memperbarui hanya satu atribut dari objek besar.

1. HEAD

Metode HEAD dalam HTTP adalah metode yang sangat mirip dengan metode GET, namun dengan satu perbedaan penting: HEAD hanya mengembalikan header respons tanpa body konten.

* Fungsi utama

Memeriksa status atau metadata dari suatu sumber daya (resource) tanpa mengunduh atau mengambil konten sebenarnya, dan mengembalikan header HTTP yang sama seperti yang akan dikembalikan oleh GET, tetapi tanpa body.

* Kegunaan umum
* Bisa digunakan untuk melihat apakah resource yang diminta ada di server tanpa harus mengunduh konten sepenuhnya.
* Bisa memberikan kode status seperti 200 OK, 404 Not Found, atau 403 Forbidden untuk memberi tahu status resource tersebut.
* Bermanfaat untuk operasi seperti caching, di mana Anda ingin mengetahui apakah resource telah diubah tanpa perlu mengambilnya lagi.
* Dengan tidak mengambil body konten, HEAD bisa menghemat bandwidth dan mempercepat permintaan ketika konten itu sendiri tidak diperlukan.