

Perbedaan GraphQL dan REST API

GraphQL dan REST API adalah dua pendekatan untuk membangun API, tetapi keduanya memiliki perbedaan yang signifikan dalam arsitektur dan metode pengambilan datanya.

Perbedaan Utama

Fitur	REST API	GraphQL
Pengambilan Data	Endpoint yang berbeda untuk sumber daya yang berbeda	Endpoint tunggal untuk semua data
Fleksibilitas	Klien dibatasi oleh struktur data yang ditentukan oleh server	Klien menentukan data yang dibutuhkan secara tepat dalam query
Overfetching/Underfetching	Dapat terjadi karena endpoint dan struktur data yang tetap	Diminimalkan dengan memungkinkan pemilihan data yang tepat

Perbedaan Utama

Fitur	REST API	GraphQL
Skema	Sering implisit atau informal	Skema yang kuat dan eksplisit diperlukan
Caching	Dapat menjadi tantangan karena endpoint yang bervariasi	Lebih mudah karena endpoint tunggal dan query yang dapat diprediksi

Keunggulan Utama GraphQL

- **Fleksibilitas:** Klien dapat meminta data yang mereka butuhkan secara tepat, mengurangi overfetching dan underfetching.
- **Efisiensi:** Endpoint tunggal dan pengambilan data yang efisien dapat menghasilkan kinerja yang lebih cepat.
- **Tipe data yang kuat:** Skema eksplisit memastikan konsistensi data dan mengurangi kesalahan.
- **Introspeksi:** Klien dapat menjelajahi data dan skema yang tersedia untuk pemahaman yang lebih baik.
- **Evolusi:** Skema dapat berubah dan berkembang tanpa merusak klien yang ada.

Kapan Memilih GraphQL

- Struktur data dan hubungan yang kompleks
- Perubahan persyaratan API yang sering
- Klien seluler dan web dengan kebutuhan data yang berbeda
- Kebutuhan pengambilan data yang efisien dan pengurangan overhead jaringan

Kapan Memilih REST

- API yang lebih sederhana dengan sumber daya yang terdefinisi dengan baik
- Penekanan pada caching dan skalabilitas
- Keakraban dengan prinsip REST
- Interoperabilitas dengan layanan RESTful yang ada

Contoh

- Misalkan Anda memiliki API yang berisi data tentang produk.
- Dengan REST API, Anda akan memiliki endpoint terpisah untuk setiap operasi yang dapat dilakukan pada produk, seperti `/products` , `/products/create` , dan `/products/update` .
- Dengan GraphQL, Anda akan memiliki endpoint tunggal, `/graphql` , dan Anda dapat menggunakan query untuk menentukan data yang Anda butuhkan.

Query GraphQL

```
query {  
  products(where: { price: { lessThan: 100 } }) {  
    id  
    name  
    price  
  }  
}
```


Respon GraphQL

```
{
  "data": {
    "products": [
      {
        "id": "1",
        "name": "T-shirt",
        "price": 99
      },
      {
        "id": "2",
        "name": "Sweatshirt",
        "price": 75
      }
    ]
  }
}
```

Kesimpulan

Seperti yang Anda lihat, dengan GraphQL, Anda dapat menentukan data yang Anda butuhkan secara tepat, yang dapat mengurangi overhead jaringan dan meningkatkan kinerja.