GraphQL

Version GraphQL Spec · verified 2026-08-01

A cheat sheet for GraphQL queries, mutations, subscriptions, and comparison with REST.

Contents

Mental model

GraphQL is a query language for your API, and a server-side runtime for executing queries using a type system you define for your data. It provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more.

Unlike REST APIs which require loading from multiple URLs, GraphQL APIs get all the data your app needs in a single request. Apps using GraphQL can be quick even on slow mobile network connections.

Queries

Queries are used to fetch data from a GraphQL server. In GraphQL, a query is simply a string sent to a server to be interpreted and fulfilled, which then returns JSON back to the client.

Feature	Description
Fields	Asking for specific fields on objects
Arguments	Passing arguments to fields to filter data
Aliases	Renaming result of a field

You can use variables to pass dynamic arguments without manipulating the query string. Variables must be declared before using them.

```
query HeroNameAndFriends($episode: Episode) {
  hero(episode: $episode) {
    name
    friends {
      name
    }
  }
}
```

Mutations

Mutations are used to modify server-side data. The top-level fields in mutation operations are allowed to cause side effects.

Type	Description
Create	Add new data to the server
Update	Modify existing data
Delete	Remove data from the server

While query fields are executed in parallel, mutation fields run in series. A mutation can contain multiple fields.

```
mutation CreateReviewForEpisode(
  $ep: Episode!,
  $review: ReviewInput!
) {
  createReview(episode: $ep, review: $review) {
    stars
```

```
    commentary
  }
}
```

> **Gotcha:** Serial execution means if we send two mutations in one request, the first is guaranteed to finish before the second begins.

Subscriptions

Subscriptions allow clients to receive real-time updates via long-lived requests. It is typically implemented with WebSockets or server-sent events.

Feature	Description
Subscribing	Initiated using `subscription` keyword
Real-time	Sends updates as events occur
Scaling	Requires stateful connection management

Subscription operations are well suited for data that changes often and incrementally, and for clients that need to receive those incremental updates.

```
subscription NewReviewCreated {
  reviewCreated {
    rating
    commentary
  }
}
```

GraphQL vs REST

GraphQL and REST both handle APIs and can serve similar purposes, but they have key differences in how data is fetched and manipulated.

REST	GraphQL
Multiple endpoints for resources	Single endpoint for all data

REST	GraphQL
Over-fetching or under-fetching	Fetches exact fields requested
Fixed response structure	Client defines response shape

REST uses HTTP methods (GET, POST, PUT, DELETE) to define operations on resources, while GraphQL uses operation types (query, mutation, subscription) and typically POST requests.

Further reading

- GraphQL Learn
- GraphQL Queries
- GraphQL Mutations
- GraphQL Subscriptions