

About the GraphQL API

In this article

Overview

About GraphQL

Why GitHub is using GraphQL

About the GraphQL schema reference

Requesting support

The GitHub GraphQL API offers flexibility and the ability to define precisely the data you want to fetch.

Overview

Here are some quick links to get you up and running with the GraphQL API:

- [Authentication](#)
- [Root endpoint](#)
- [Schema introspection](#)
- [Rate limits](#)
- [Migrating from REST](#)

For more information about GitHub's APIs, see "[About GitHub's APIs](#)."

About GraphQL

The [GraphQL](#) data query language is:

- **A [specification](#).** The spec determines the validity of the [schema](#) on the API server. The schema determines the validity of client calls.
- **[Strongly typed](#).** The schema defines an API's type system and all object relationships.
- **[Introspective](#).** A client can query the schema for details about the schema.
- **[Hierarchical](#).** The shape of a GraphQL call mirrors the shape of the JSON data it returns. [Nested fields](#) let you query for and receive only the data you specify in a single round trip.
- **An application layer.** GraphQL is not a storage model or a database query language. The *graph* refers to graph structures defined in the schema, where [nodes](#) define objects and [edges](#) define relationships between objects. The API traverses and returns application data based on the schema definitions, independent of how the data is stored.

Why GitHub is using GraphQL

GitHub chose GraphQL because it offers significantly more flexibility for our integrators.

The ability to define precisely the data you want—and *only* the data you want—is a powerful advantage over traditional REST API endpoints. GraphQL lets you replace multiple REST requests with *a single call* to fetch the data you specify.

For more details about why GitHub invested in GraphQL, see the original [announcement blog post](#).

About the GraphQL schema reference

The docs in the sidebar are generated from the GitHub GraphQL [schema](#). All calls are validated and executed against the schema. Use these docs to find out what data you can call:

- Allowed operations: [queries](#) and [mutations](#).
- Schema-defined types: [scalars](#), [objects](#), [enums](#), [interfaces](#), [unions](#), and [input objects](#).

You can access this same content via the [Explorer Docs sidebar](#). Note that you may need to rely on both the docs and the schema validation to successfully call the GraphQL API.

For other information, such as authentication and rate limit details, check out the [guides](#).

Requesting support

For questions, bug reports, and discussions about GitHub Apps, OAuth apps, and API development, explore the [APIs and Integrations discussions on GitHub Community](#). The discussions are moderated and maintained by GitHub staff, but questions posted to the forum are not guaranteed to receive a reply from GitHub staff.

Consider reaching out to [GitHub Support](#) directly using the contact form for:

- guaranteed response from GitHub Enterprise Cloud staff
- support requests involving sensitive data or private concerns
- feature requests
- feedback about GitHub Enterprise Cloud products

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)