

Rest vs GraphQL

RESTful Traditions vs GraphQL Innovations: Which is Right for You?



MAMA SAMBA BRAIMA NELSON DJALO

FEB 05, 2024



79



1

Share

The choice between REST and GraphQL often depends on factors such as the project's specific requirements, the development team's familiarity with each technology, and the ecosystem in which the application operates.

REST has been around for a longer time and is a well-established standard for building APIs. Many existing systems and services are built on RESTful architecture. It is often the default choice for building APIs, especially when simplicity and widespread adoption are important considerations.

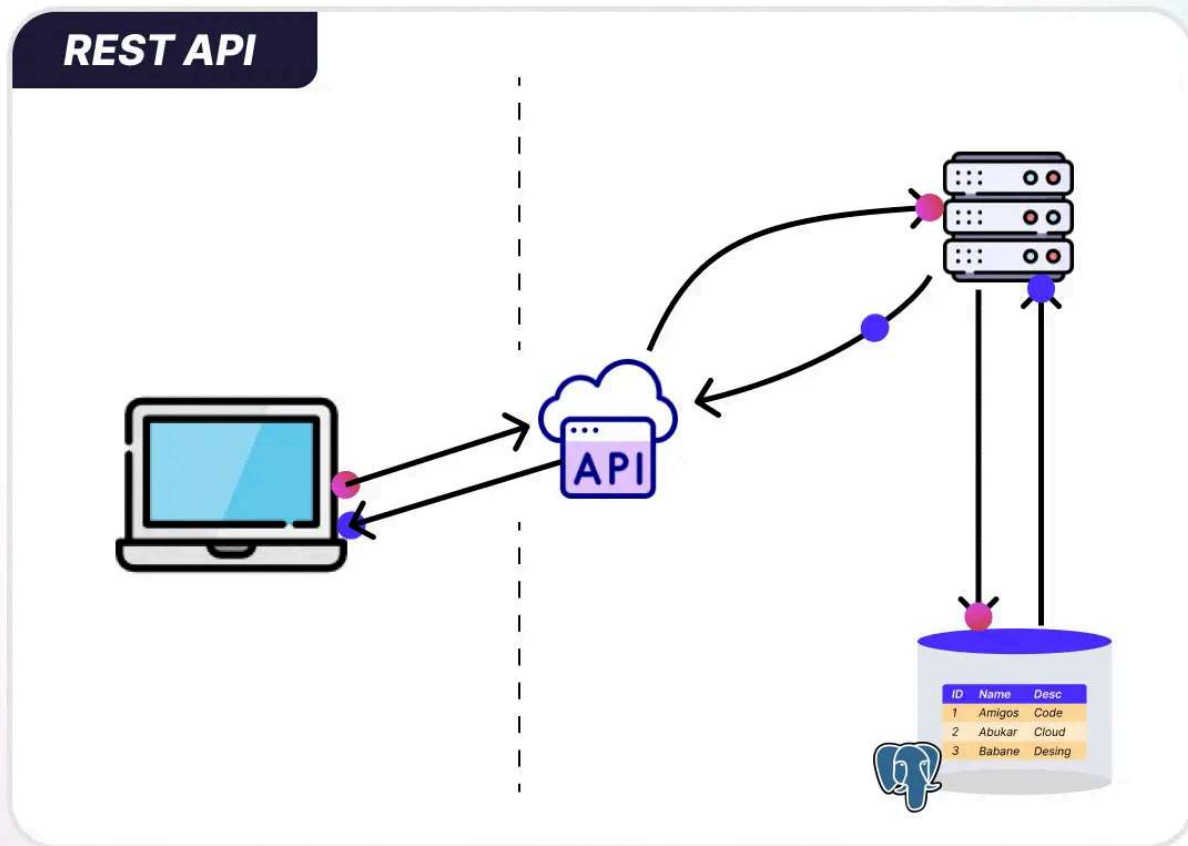
GraphQL, on the other hand, gained popularity for addressing some limitations of REST, especially in scenarios where clients need more flexibility in data retrieval. It has been widely adopted by companies, particularly those with complex data requirements or real-time needs.

Lets dive into the differences

REST

blog.amigoscode.com

REST API EXPLAINED



Declarative
Data
Fetching



Hierarchical
Structure



Strongly
Typed
Schema



Real Time
Data
Subscriptions



Single
Endpoint



Self
Documenting
API



[amigoscode.com](https://blog.amigoscode.com)

[Open Source](#) and Developed by Meta/Facebook

1. **Stateless:** REST is stateless, meaning each request from a client to a server contains all the information needed to understand and process the request. The server does not store any information about the client between requests.
2. **Multiple Endpoints:** In a RESTful API, different resources are usually exposed through different endpoints (URLs). Each endpoint represents a specific resource or a collection of resources.
3. **Over-fetching and Under-fetching:** Clients may receive more or less data than needed. Over-fetching occurs when the server returns more data than the client needs, while under-fetching occurs when the server doesn't provide enough data, requiring additional requests.
4. **Fixed Structure:** Data is typically returned in a fixed structure defined by the server, and clients must adapt to the server's response structure.
5. **HTTP Methods:** REST APIs use standard HTTP methods like GET, POST, PUT, and DELETE to perform CRUD (Create, Read, Update, Delete) operations.

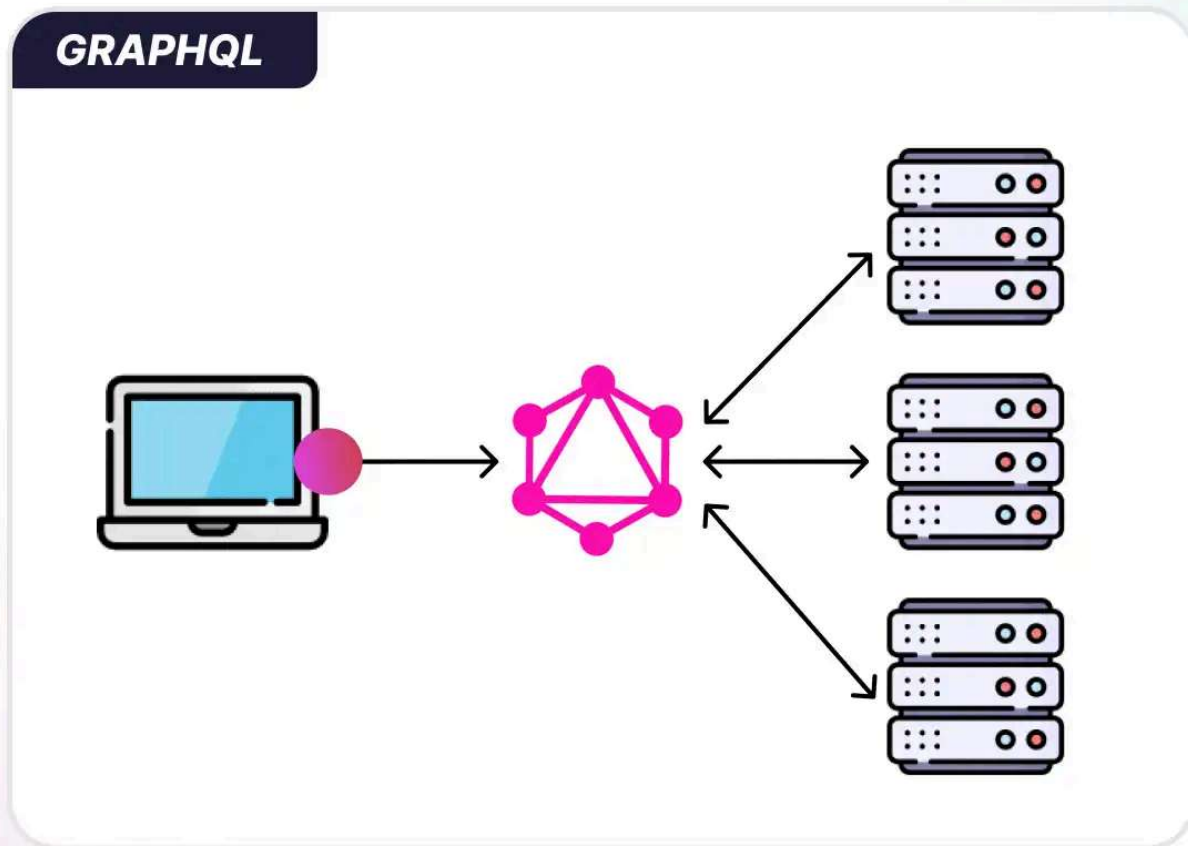
Thanks for reading Amigoscode Newsletter!
Subscribe for free to receive new posts and
support my work.

[Subscribe](#)

GraphQL

blog.amigoscode.com

GRAPHQL EXPLAINED



Declarative
Data
Fetching



Hierarchical
Structure



Strongly
Typed
Schema



Real Time
Data
Subscriptions



Single
Endpoint



Self
Documenting
API



amigoscode.com

1. **Single Endpoint:** GraphQL uses a single endpoint for all its operations. Clients specify the structure of the response, and the server returns exactly that structure.

2. **Declarative Data Fetching:** Clients request only the data they need, avoiding over-fetching and under-fetching issues. This can reduce the amount of data transferred over the network.
3. **Strong Typing:** GraphQL has a strong typing system, and clients can specify the shape and structure of the data they expect, reducing the chances of runtime errors.
4. **Real-time Data with Subscriptions:** GraphQL supports real-time data updates through subscriptions, enabling clients to receive updates when data changes on the server.
5. **Flexible and Evolvable:** The GraphQL schema allows for flexible data retrieval, and it is designed to be easily extensible and evolvable over time.

Question

POLL

Which are you using next?

REST

GraphQL

Both

Which Should You Use?

It really depends IMO. Here are somethings to consider.

1. **Development Team Expertise:**

- The familiarity and expertise of your development team with either REST or GraphQL may influence your decision.

2. **Ecosystem and Tooling:**

- Consider the available tools, libraries, and ecosystem support for both REST and GraphQL in your chosen technology stack.

3. **Project Scale and Complexity:**

- Assess the scale and complexity of your project, as well as the potential for future growth and changes in data requirements.

4. **Hybrid Approaches:**

- Depending on your project's needs, you might also consider a hybrid approach, using both REST and GraphQL where each fits best.

I think you will most likely use REST for your next project.

This is all for this post. Chat to you soon 🙌

Thanks for reading Amigoscode Newsletter!
Subscribe for free to receive new posts and
support my work.

[Subscribe](#)

79 Likes · 2 Restacks

[← Previous](#)[Next →](#)

1 Comment



Write a comment...



Florent Feugere Feb 6

Hello, there are some incoherences in your post, like the gifs using both GraphQL features, or Rest being developped by Meta ? Great article nonetheless, thank you.

♡ LIKE (1) 💬 REPLY ↗ SHARE



© 2024 Amigoscode · [Privacy](#) · [Terms](#) · [Collection notice](#)
[Substack](#) is the home for great culture