brijpandeyji

GraphQL

VS

REST API

Know the difference!!

swipe >

GraphQL

GraphQL is a query language for your API, as well as a runtime for executing those queries against your data.

Developed by Facebook and released in 2015

REST API

It is an architecture style to develop web applications. It uses HTTP protocol as a communication interface.

Introduced by Roy Fielding in 2000

The biggest difference between GraphQL and REST is the manner in which data is sent to the client. In a REST architecture, the client makes an HTTP request and data is sent as an HTTP response, while in GraphQL, the client requests data with queries.

swipe >

The key differences between GraphQL and REST are based on the following.

7 Differences

1. Structure
2. Flexibility
3. Efficiency
4. Caching
5. Data Fetching
6. API Evolution
7. Versioning

swipe >

Structure

Based on endpoints, request and response

GraphQL

GraphQL APIs, have a single endpoint that is used to query, mutate, and subscribe to data.

Request and Response

Uses query or mutation to request data.

Server responds based on the request query.

REST API

REST APIs have a fixed set of endpoints, each representing a specific resource.

Uses HTTP methods to request data.

Response based on the endpoint being called.

Request and Response

swipe >

Flexibility

GraphQL

GraphQL allows clients to request exactly the data they need, in a single request.

More Flexible

REST API

REST APIs are typically more rigid and require multiple requests to retrieve all the data needed for a particular use case.

Less Flexible

swipe >

Efficiency

GraphQL

GraphQL can be more efficient than REST because it allows the client to request all the data it needs in a single request.

Reduces the number of round trips to the server, thus improving efficiency.

REST API

It is less efficient than GraphQL as it may need to make multiple requests to different endpoints to retrieve the data needed.

Efficient at times where there are large number of small, specific requests.

Both can be efficient

Efficiency of an API depends on the specific needs of the client and the use case.

swipe >

Caching

It is a technique used to improve the performance of an API by reducing the number of requests made to the server.

GraphQL

In GraphQL, the response is tailored to the specific request, so it is more difficult to cache.

REST API

REST APIs can be cached more easily, as the response from a specific endpoint will always be the same.

Note

While caching can improve performance, it can also lead to outdated or stale data being served to clients if the cache is not properly invalidated or refreshed.

swipe >

Data Fetching

GraphQL

With GraphQL, the client can request exactly the data it needs, reducing the amount of data transferred over the network.

REST API

REST APIs can sometimes suffer from over fetching or under fetching, where the client receives more or less data than it needs.

Note

GraphQL and REST each have their own approaches to addressing these issues.

swipe >

API Evolution

GraphQL

In GraphQL, the API can evolve more flexibly, as the client can request only the fields it needs and ignore the rest.

Can add new fields and types without breaking existing clients.

REST API

In a REST API, adding a new field or endpoint requires a new version of the API. This makes API evolution harder.

Must use versioning to ensure that the existing clients do not break.

Needs to be supported and maintained

swipe >

Versioning

GraphQL

GraphQL APIs do not require versioning because the client can request any combination of fields.

In a GraphQL API, versioning can be implemented by creating a new schema for each version of the API.

REST API

REST APIs typically require versioning when the shape of the data changes.

In a REST API, versioning can be implemented by using a different URL for each version of the API.

swipe >

Conclusion

Overall

GraphQL and REST are both useful tools for building APIs, and which one you choose depends on the specific needs of your project.

You now know the differences between a GraphQL API and a REST API.