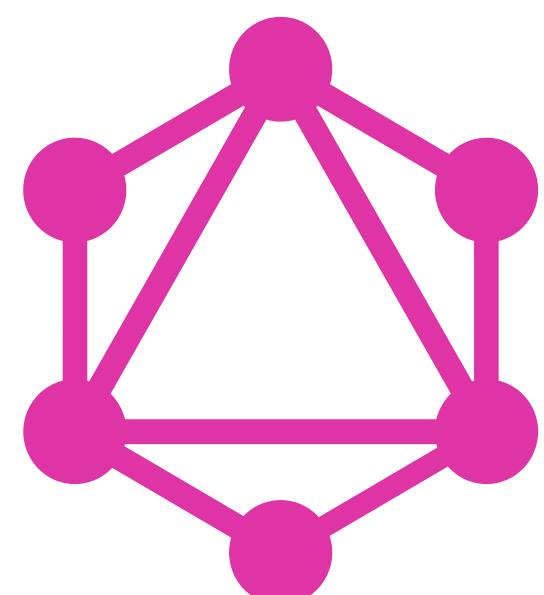


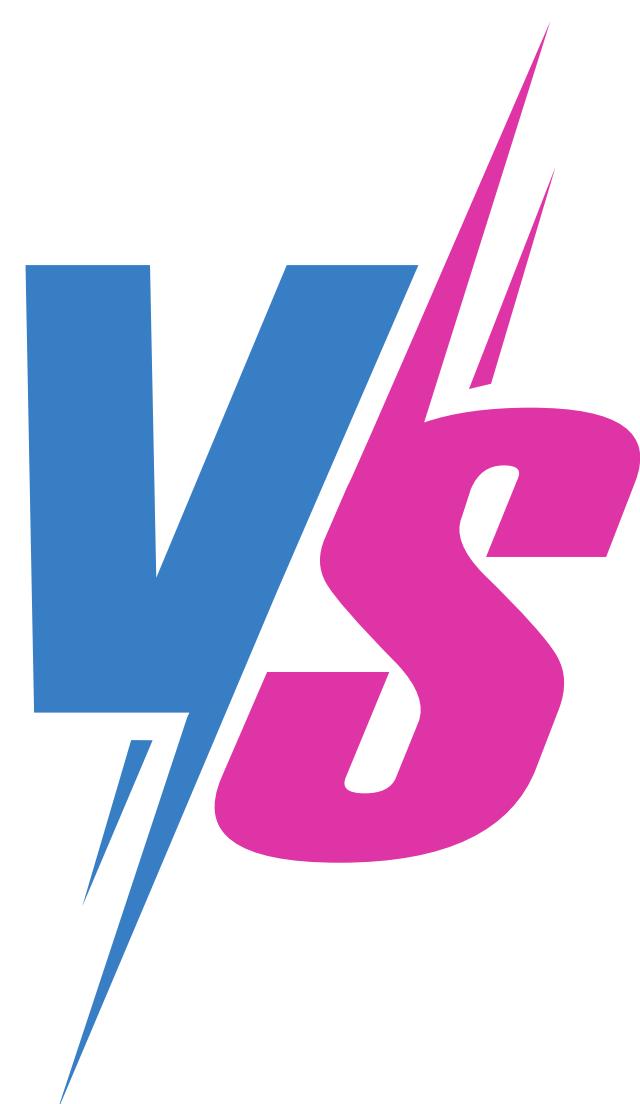
Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz



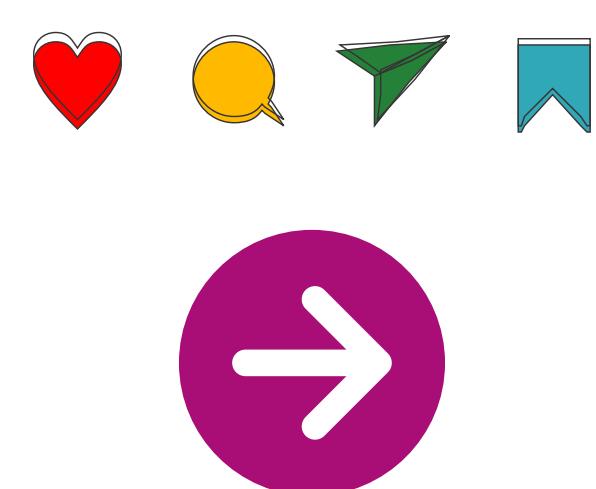
# GraphQL



## REST



## A COMPARISON





Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz

# GraphQL

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

**It was developed by Facebook and released in 2015.**

# REST API

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

**It was introduced by Roy Fielding in 2000**





Tauseef Fayyaz 

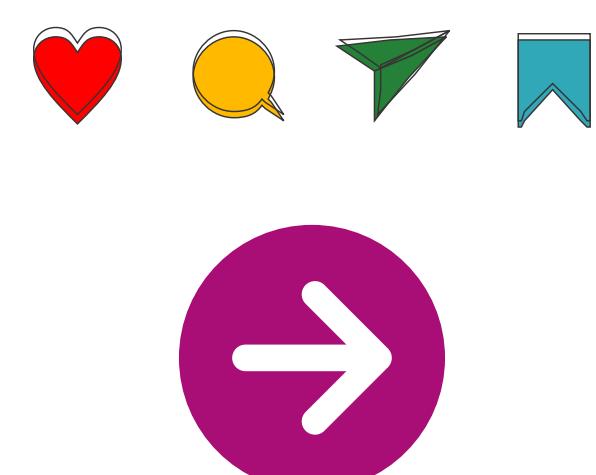
Follow for coding, software and career tips

@tauseeffayyaz

**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.

In GraphQL, the client requests data with queries.





Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz

## Key Differences

Let's discuss **7** key differences between **GraphQL** and **REST** are based on the following.

Structure

Flexibility

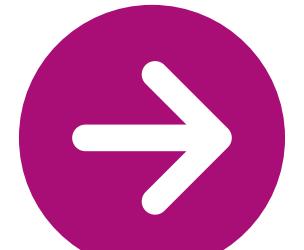
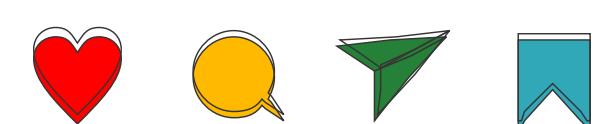
Caching

Efficiency

Data Fetching

API Evolution

Versioning





Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz

## Structure

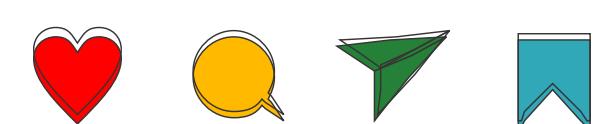
based on endpoints, request and response

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

Uses query or mutation to request data.  
Server responds based on the request query.

**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.





Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz

## Flexibility

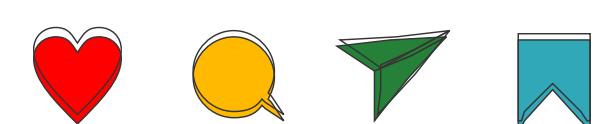
refers to how adaptable and customizable each API architecture is when it comes to querying and handling data.

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

More flexible

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

Less Flexible





Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz

## Efficiency

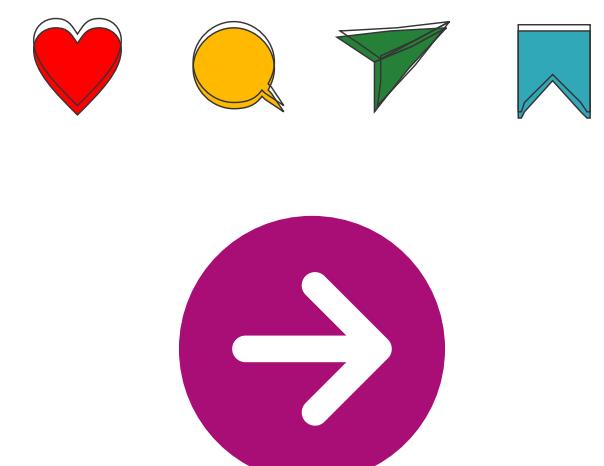
Efficiency of an API depends on the specific needs of the client and the use case.

**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** 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.





Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz

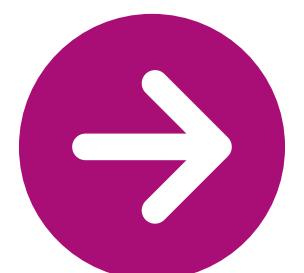
## Caching

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

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

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

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.





Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz

## Data Fetching

how data is fetched

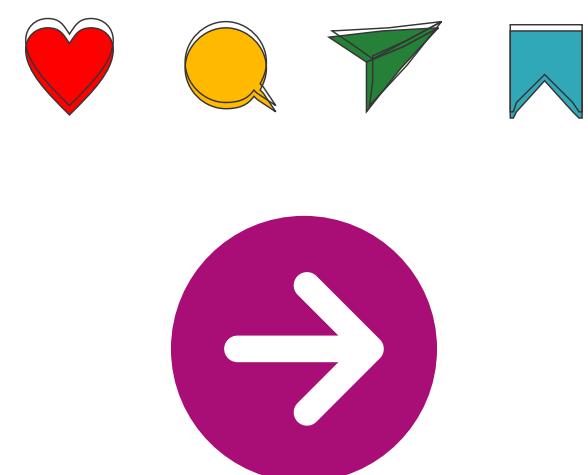
With **GraphQL**, the client can request exactly the data it needs, reducing the amount of data transferred over the network.

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

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

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

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





Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz

## API Evolution

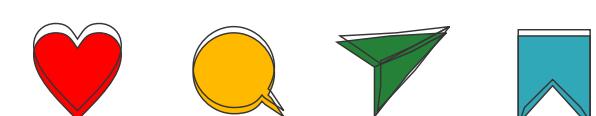
ability of an API to evolve without breaking existing functionality

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.

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.





Tauseef Fayyaz [in](#)

Follow for coding, software and career tips

@tauseeffayyaz

## Versioning

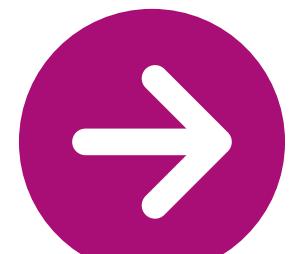
an important aspect of API design,  
particularly when making changes  
or introducing new features

**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 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.





Tauseef Fayyaz [in](#)

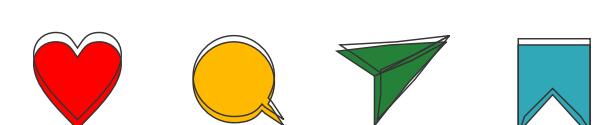
Follow for coding, software and career tips

@tauseeffayyaz

## Conclusion

**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.





Tauseef Fayyaz 

Follow for coding, software and career tips

@tauseeffayyaz

# THANK YOU

LIKE & REPOST