[**https://www.imaginarycloud.com/blog/graphql-vs-rest/**](https://www.imaginarycloud.com/blog/graphql-vs-rest/)

**GraphQL:**

* GraphQL is a query language and a set of tools that use HTTP to work on **single endpoints** in order to **optimize flexibility and performance**.
* In GraphQL, data is organized into a graph, and objects are structured by nodes following a schema.

Graphql sole these two problems:  
Over-fetching is fetching too much data, meaning there is data in the response you don't use.

Under-fetching is not having enough data with a call to an endpoint, forcing you to call a second endpoint.

We have these things into GraphQL:

1. Apolo server express
2. TypeDesfs  
   All the queries and mutation writted into Types, and we describe the the types of data, like that  
   const typeDefs= gql`

type User{

name:String!

age:Int!

married:Boolean!

}

# Queries

type Query{

getAllUsers:[User!]!

}

#Mutations

type Mutation{

createUser(name:String!,age:Int!,married:Boolean!):User!

}

`

* 1. After that all these mutations and queries writted into resolvers .

1. Resolvers
   1. Query
      * 1. it returns data back they behave like get api
   2. Mutaion
      * 1. we used mutaions if we need to change data into our , orignal data or DB

APOLO for test APIS:  
[https://www.apollographql.com/docs/apollo-server/#:~:text=Apollo%20Server%20is%20an%20open,use%20data%20from%20any%20source](https://www.apollographql.com/docs/apollo-server/" \l ":~:text=Apollo Server is an open,use data from any source).  
  
https://www.youtube.com/watch?v=PHabPhgRUuU