



GraphQL

What is GraphQL

- GraphQL is a query language, that means it's a specific syntax we can use to query a server to request or mutate data
- Alternative to using a REST API

Using a REST API is more of an architectural style, an approach to serve and fetching data.

Whereas GraphQL differs from it because which is actually a query language with its own syntax and rules

(It still uses HTTP under the hood)

GraphQL gives more flexibility and control about how we make them and what data we want to fetch or mutate

And the way a GraphQL server handles those requests is very different to how a typical REST API would handle them

Differences between GraphQL and REST API'S

When we have a REST api we typically send HTTP requests to specific endpoints to interact with a certain type of data

For example:

1. `pokemonsite.com/api/pokemon`
2. `pokemonsite.com/api/pokemon/123`

We could add a new pokemon using the first endpoint

We even might send a get request to the second endpoint with an id at the end of it to fetch a single pokemon with the id. We could delete or even update the pokemon

Differences between REST APIs and GraphQL

REST API'S drawbacks

The REST API architectural style is great, but there are some drawbacks when it comes to using a REST API when your application scales and your data gets a little bit more complex such as:

- Over fetching:
 - Getting back more data than we need

Example:

We could send a GET request to the following endpoint:

`mysite.com/api/courses`

The request would go to the server and retrieve all the courses and we would receive them in JSON format like:

```
{
  "id": "1",
  "title": "Thud" ,
  "author " : {...},
  "price" : "10.99" ,
  "thumbnail_url" : "...",
  "video_url" : "...",
}
```

Let's say we only want the title and the id, title and the thumbnail because that's what we would be showing in this particular page. And the bunch of data that we received is useless

- Under fetching
 - Getting back less data than we need

Let's say we make a GET request to the same endpoint:

mysite.com/api/courses/1

```
{
  "id": "1",
  "title": "Thud" ,
  "author " : {...},
  "price" : "10.99" ,
  "thumbnail_url" : "...",
  "video_url" : "...",
}
```

And it's great we received the data that we need to show

But there is a problem let's say we only got the author name and it's id

What if we wanted to show the courses this author has published and info's about it?

GraphQL differences between REST API's

GraphQL uses a single endpoint

mygraphqlsite.com/graphql

The way we send a query to the server is by using a special graphql syntax that looks like this:

```
Query {
  courses {
    id,
    title,
    thumbnail_url
  }
}
```

```
}  
}
```

This syntax allows us to specify exactly what data and what fields we need back from the server

The other thing GraphQL allows us is to fetch nested related data withing a single query

Let's take the example from before which we said that REST APIs could have an under fetch problem

By using GraphQL we would send a request to the following endpoint:

mygraphqlsite.com/graphql

And send the query in the following way:

```
Query {  
  course(id: "1"){  
    id,  
    title,  
    thumbnail_url ,  
    author {  
      name ,  
      id,  
      courses {  
        id,  
        title,  
        thumbnail_url  
      }  
    }  
  }  
}
```

In this query it means the following:

We want the course with the id 1

The author name, id and its courses with id, title and thumbnail

And all this is done in a single request or query.

That solves the under fetching problem we had

- --

| A mutation is when you ask the GraphQL server to add, delete or update a new server

How to make a query using GraphQL

The first thing to do when making a GraphQL query is:

- Write the word 'query'
- Then the name of the query
- Inside the query you'll put the resource/object you want to query (you can think of something like the table)
- The fields of the resource you want
 - This will retrieve you a list of field from the resource you specified
 -

```
query ReviewsQuery {  
  reviews {  
    ratings  
    content  
    id  
  }  
}
```

How we query a graph

When we make a graphql server or API we're making something called a graph and a graph in visual terms is basically a bunch of connected data that looks like this:

When we query a graph we're doing the following:

For example, let's say we want to get the authors from the review query

The review doesn't have an author property, they're a separate resource but they're linked to reviews so the related data which was specified in the graphql server

```
query ReviewsQuery {  
  reviews {  
    ratings  
    content  
    id  
    author {  
      name  
      id  
      verified  
    }  
  }  
}
```

With this query it would bring all the reviews and the related authors with their names,id's and verified field

Example:

```
query ReviewsQuery {  
  reviews {  
    ratings  
    content  
    id  
    author {  
      name  
      id  
      verified  
    }  
  }  
  game {  
    title  
  }  
}
```

```
    platform
  }
}
```

Retrieving single itens or by id

This query is retrieving the reviews with its related authors and the game titles and platforms

If you want to retrieve only a single item you need to add more entrypoints to the graph(schema) and a resolver function

For example:

```
type Query {
  games: [Game]
  reviews: [Review]
  review(id: ID!):Review
  authors: [Author]
}
```

we just added the review field with the Id being a non-nullable field and it's of type Review