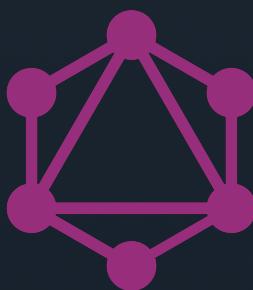


# What Is(n't) GraphQL?

*The Misconceptions*

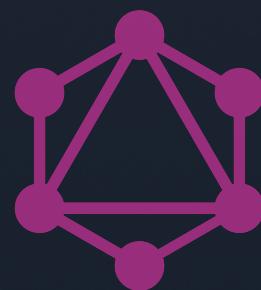




# Shruti Kapoor

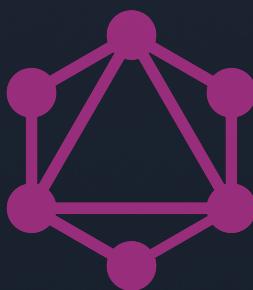
[tinyletter.com/shrutikapoor08](http://tinyletter.com/shrutikapoor08)

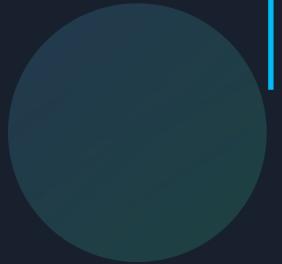
 @shrutikapoor08



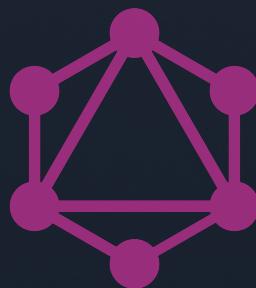
# #DEVJOKE

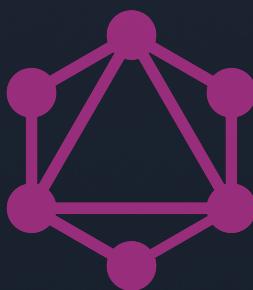
 @shrutikapoor08





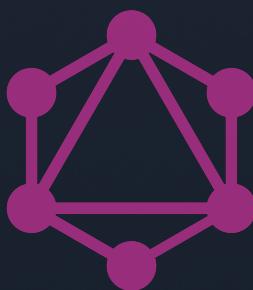
# Where do programmers hang out?



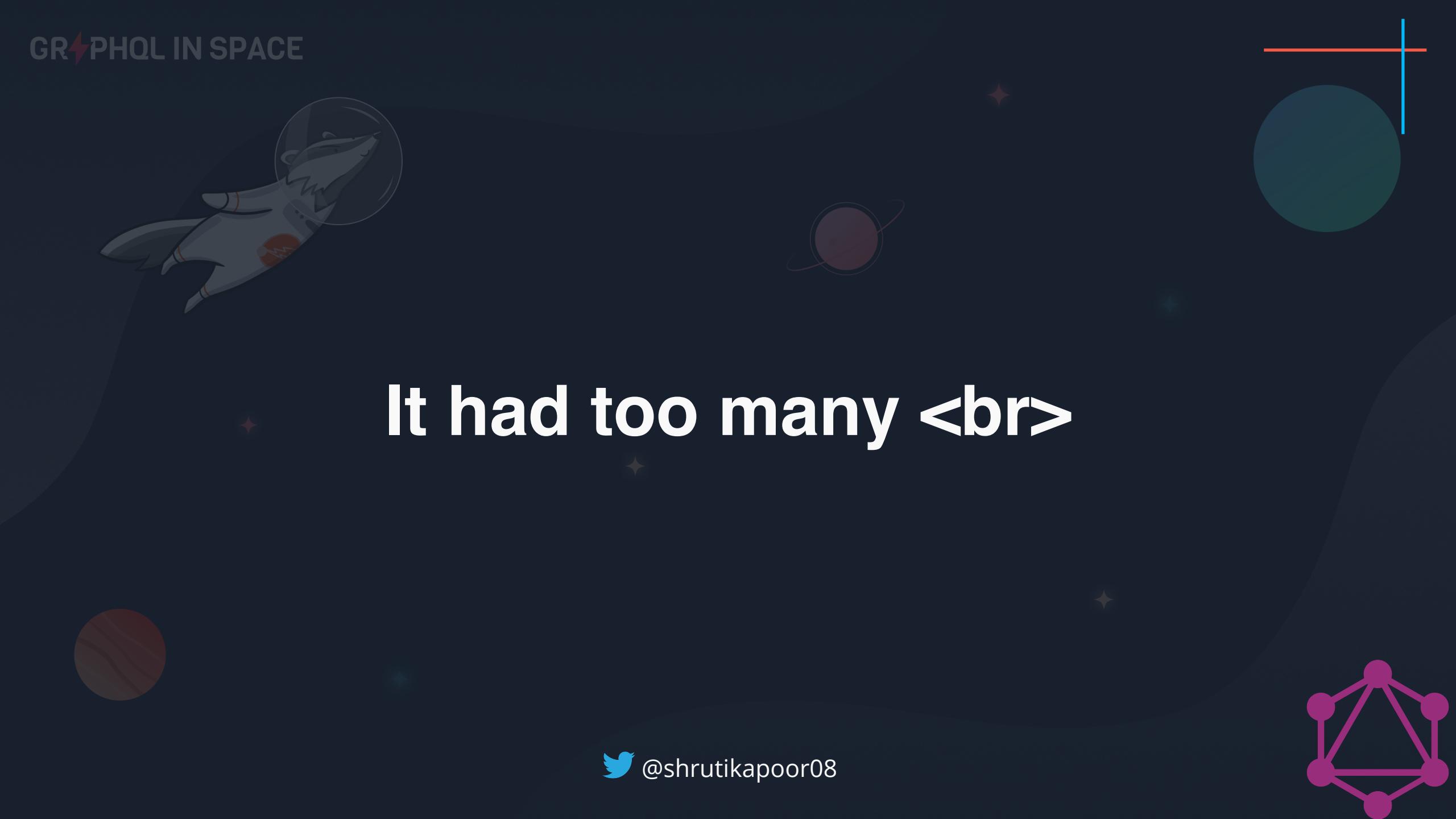


# Foo Bar

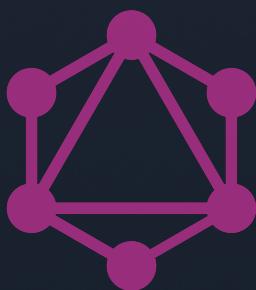
 @shrutikapoor08



# How did HTML get drunk?

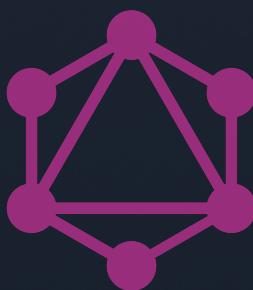


# It had too many <br>



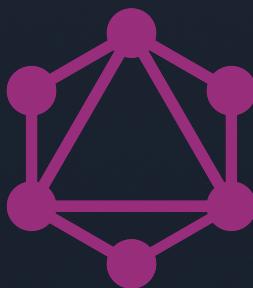


# What did the developer say to the git repository?



# FORK YOU

 @shrutikapoor08



[Code](#)[Issues 1](#)[Pull requests 56](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)[master](#)[2 branches](#)[0 tags](#)[Go to file](#)[Add file](#)[Code](#)

shrutikapoor08 Merge pull request #268 from SauravKumar-max/patch-3 ...

36a1fcf 6 minutes ago 435 commits

[images](#)

Merge branch 'master' into master

7 minutes ago

[.gitignore](#)

add be vs fe devjoke

11 months ago

[CONTRIBUTING.MD](#)

Update CONTRIBUTING.MD

11 months ago

[README.md](#)

Merge branch 'master' into patch-3

6 minutes ago

[README.md](#)

## Dev Joke

Send me a PR if you know a good Dev Joke.

If somebody refers me for a job 🤖 will it be considered 'pass by reference' or 'pass by value'? 😂😂

## About



#DevJoke. Submit a PR if you know a good dev joke.

jokes developer-jokes humor  
programming hacktoberfest

Readme

## Releases

No releases published  
[Create a new release](#)

## Packages

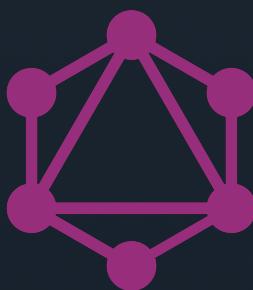
No packages published  
[Publish your first package](#)

## Contributors 138



<https://github.com/shrutikapoor08/devjoke>

# GRAPHQL IN SPACE



 @shrutikapoor08



# DevJokes.netlify.app

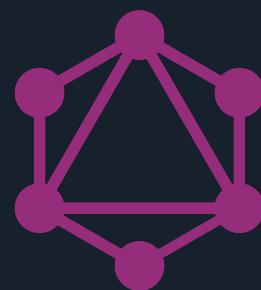


Dgraph

Powered by GraphQL



@shrutikapoor08





Omar Ayoubi



Naman Jain



Minhaj Shakeel

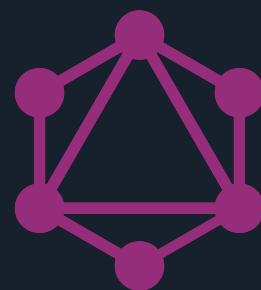


Dgraph

Powered by GraphQL

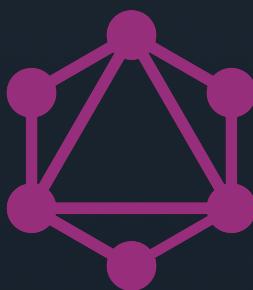


@shrutikapoor08



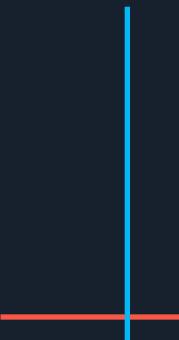
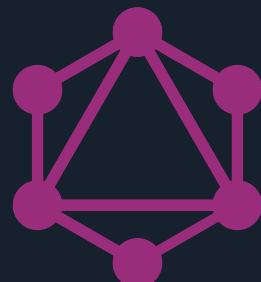
# What Is(n't) GraphQL?

*The Misconceptions*



# Outline

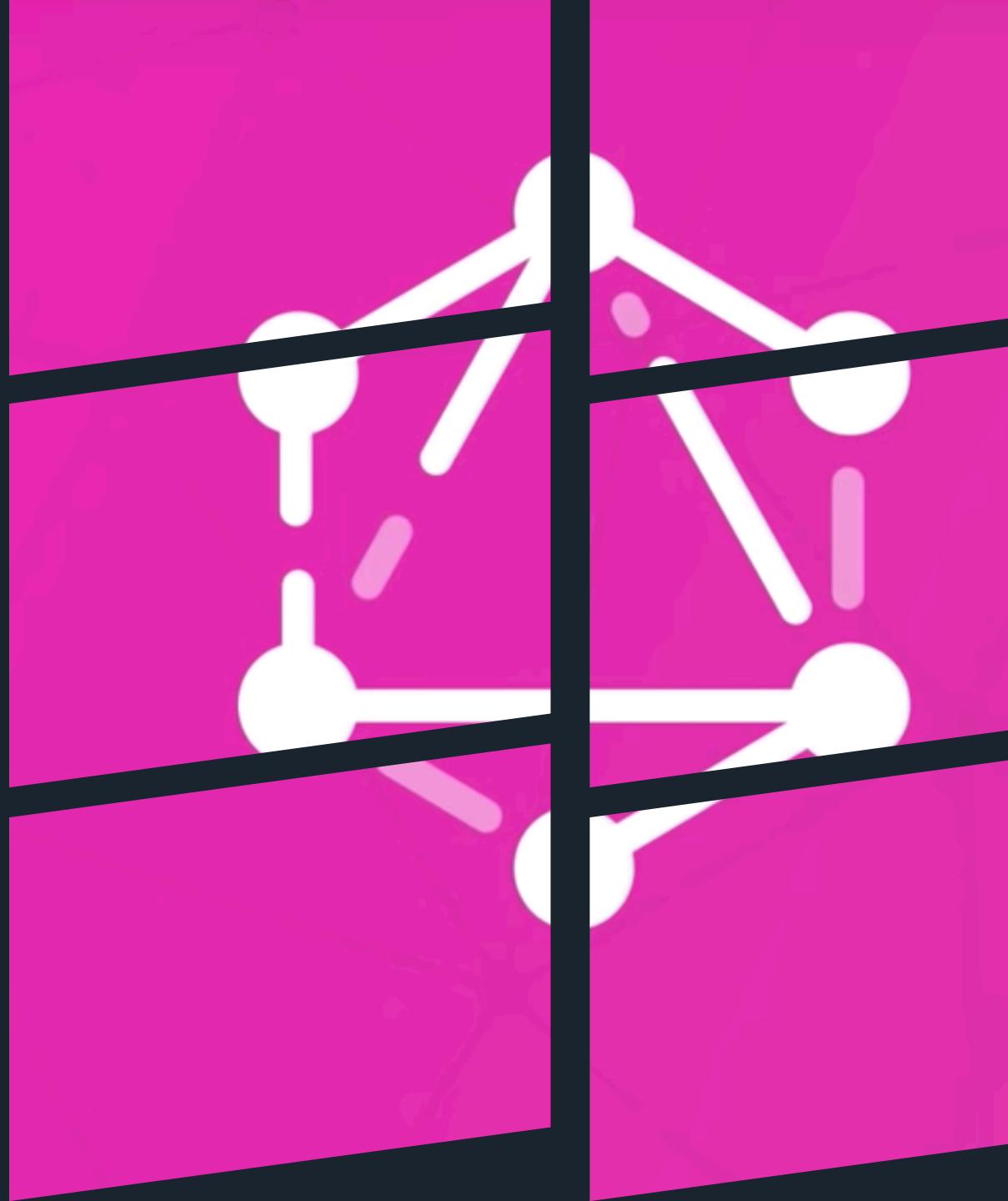
- 1.GraphQL vs REST
- 2.Misconceptions
- 3.The good things
- 4.How to get started



# GraphQL

VS

# REST



# Library app



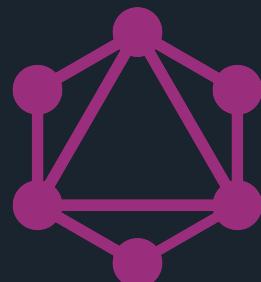
```
{  
  "books": [  
    {  
      "title": "The Mythical Man Month",  
      "author": [  
        "Frederik P. Brooks, Jr."  
      ]  
    },  
    {  
      "title": "The Pragmatic Programmer",  
      "author": [  
        "Andy Hunt",  
        "Dave Thomas"  
      ]  
    },  
    {  
      "title": "Big Magic",  
      "author": [  
        "Elizabeth Gilbert"  
      ]  
    },  
    {  
      "title": "Thinking, Fast and Slow",  
      "author": [  
        "Daniel Kahneman"  
      ]  
    }  
  ]  
}
```

# Step 1: Sending data to server

## REST

Inspect API documentation to find out what methods are available

Eg: /books, /authors



## GraphQL

First determine what data you need

Eg: Need a book and its author



# Step 2: Creating a request

**REST**

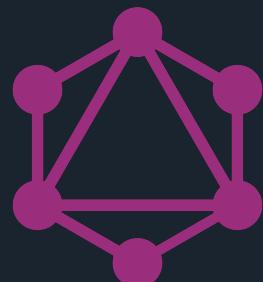
Endpoints: /books, /author

Find out ID for a particular book from books and send to author.

**GraphQL**

Endpoint: /graphql

Create a schema of fields you need  
book { title } , author { name }



# Step 3: Knowing what to send

**REST**

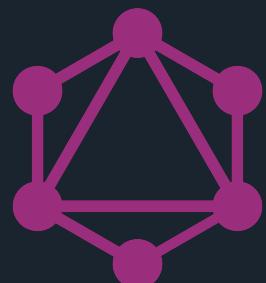
GET

POST

PATCH

DELETE

PUT



**GraphQL**

Query

Mutation



# Step 4: Building resources

## REST

Resource implementation

GET: retrieveAuthors (authorId)

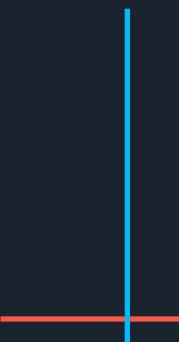
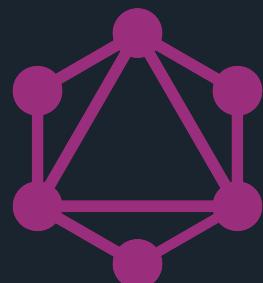
GET: retrieveBooks (bookId)

## GraphQL

Resolver

book: look up resolver for field  
book

book: () => { //fetch from DB }



# Step 5: Parsing received data

**REST**

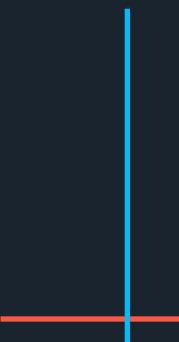
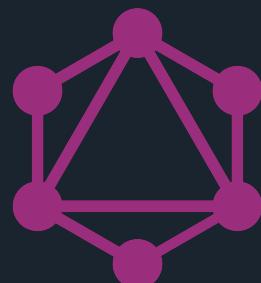
JSON object

Fields in addition to the ones you  
needed

**GraphQL**

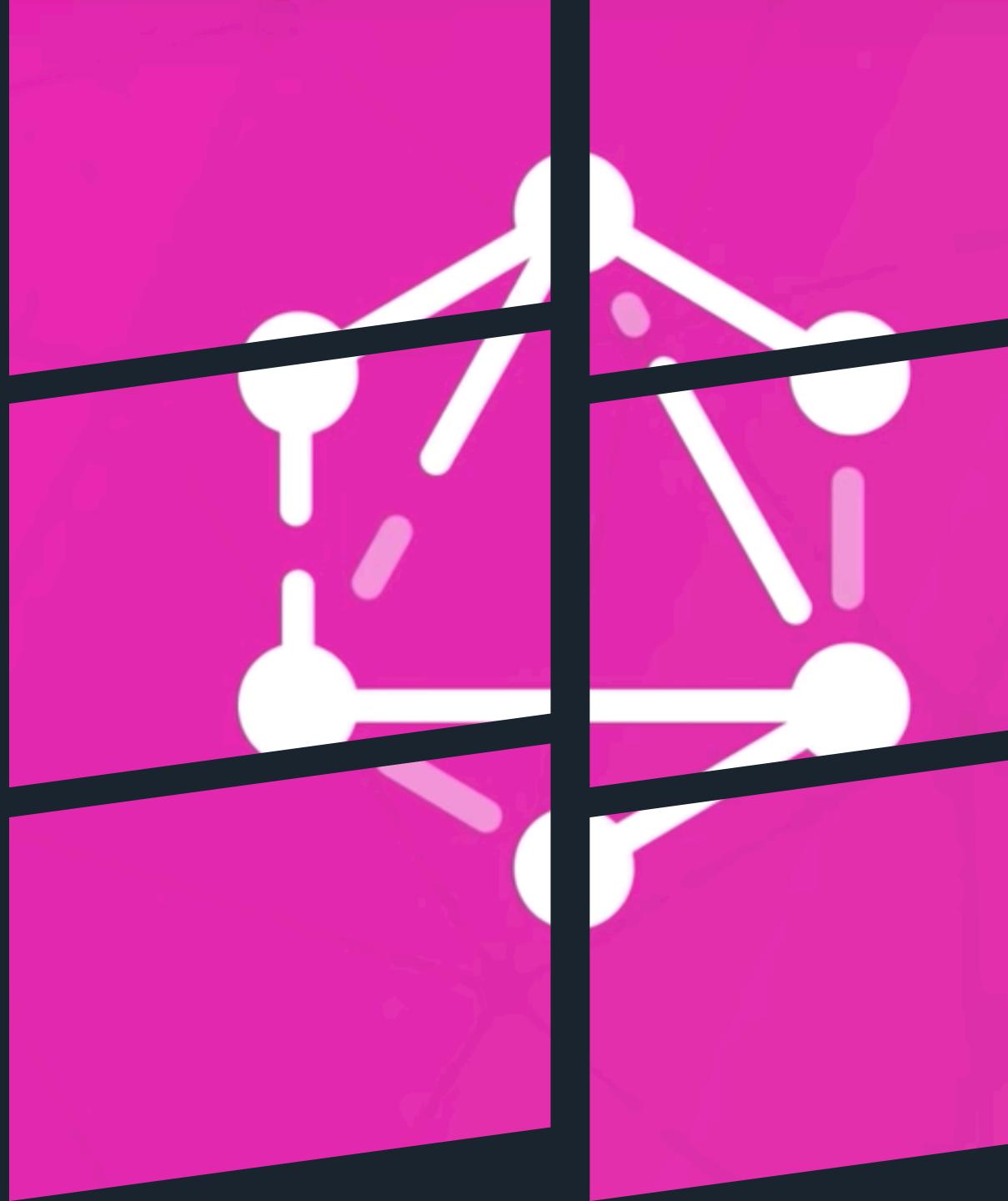
JSON object

Only the fields you requested



# Myths

*Busted*





**Shruti Kapoor** @shrutikapoor08 · Dec 1, 2019

GraphQL Folks! Can you help me with some research?

1. What are the common misconceptions you've had with **#GraphQL** when you were starting out?  
2. What are some myths people have when you introduce GraphQL to new folks?

RTs appreciated!

69 112 267

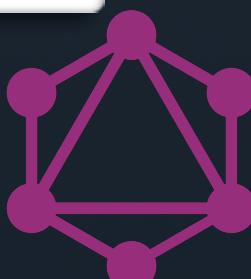
Upvote icon

Retweet icon

Reply icon

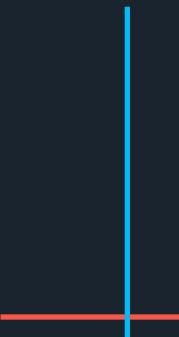
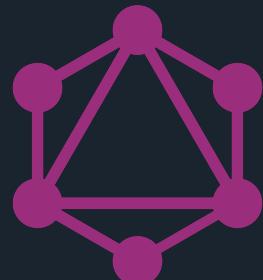
Share icon

Like icon



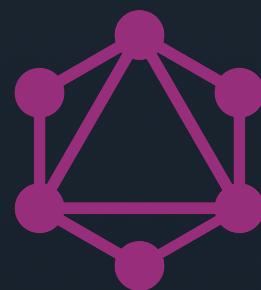
# Common Myths

1. GraphQL = Graph
2. GraphQL = SQL
3. GraphQL only works with graph type data.
4. Data fetching works magically.
5. GraphQL replaces Redux
6. REST and GraphQL can't co-exist.
7. GraphQL is difficult to introduce in an existing project
8. GraphQL is useful only for front-end
9. GraphQL writes database queries itself.
10. GraphQL requires complicated clients.





# GraphQL = Graph



Though it enables GraphDatabases for the browser

# GraphQL = Graph

1. I thought it was this Facebook's Graph Search kind of thing. An "official" way of getting people's data lol.
2. The d3.js kind of myths hehe

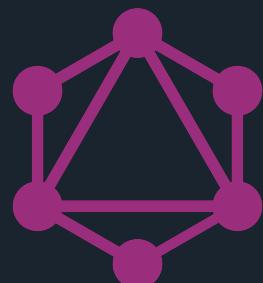
- That it's somehow related to Facebook's "graph"

# Myth: GraphQL only works with graph like structures.

Can be used to query a graph structure

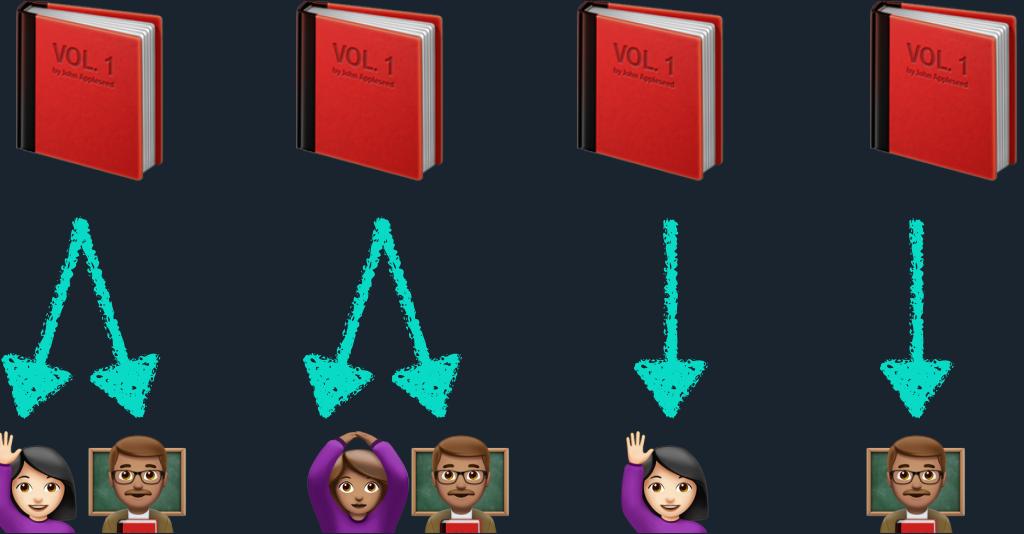
Graph = The "graph" in GraphQL is used to represent the graph-like structure of data.

Model data in terms of nodes and how they connect





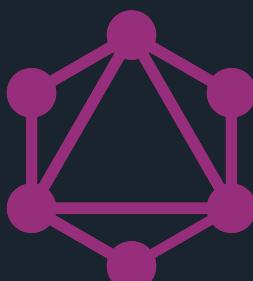
```
{  
  "books": [  
    {  
      "title": "The Mythical Man Month",  
      "author": [  
        "Frederik P. Brooks, Jr."  
      ]  
    },  
    {  
      "title": "The Pragmatic Programmer",  
      "author": [  
        "Andy Hunt",  
        "Dave Thomas"  
      ]  
    },  
    {  
      "title": "Big Magic",  
      "author": [  
        "Elizabeth Gilbert"  
      ]  
    },  
    {  
      "title": "Thinking, Fast and Slow",  
      "author": [  
        "Daniel Kahneman"  
      ]  
    }  
  ]  
}
```



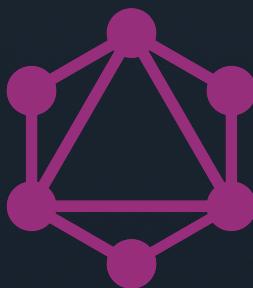
Name

Email

Bio

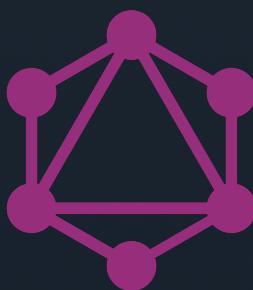


# GraphQL = SQL



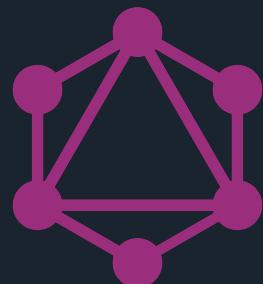
# GraphQL = SQL

2. A query language for a graph database, like neo4j



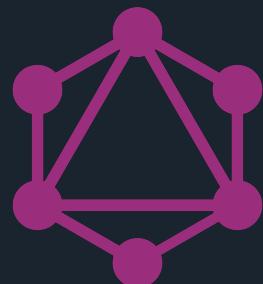
# Myth: QL = SQL.

SELECT \* FROM books WHERE title = 'The Pragmatic Programmer'



# Myth: QL = SQL.

“Get Book with Title - The Pragmatic Programmer”

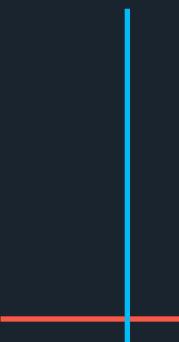
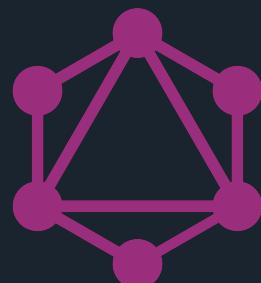


# Myth: QL = GraphQL is a database language like SQL.

GraphQL is a query language for API.

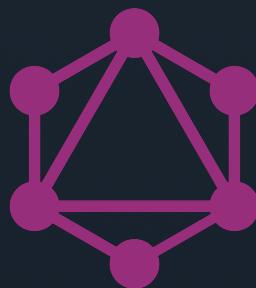
It is not a database language.

GraphQL is typically used to fetch data from API, however some libraries like DGraph use GraphQL to get data from the database.



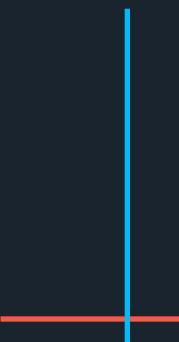
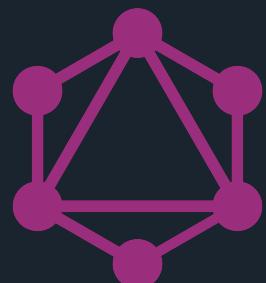


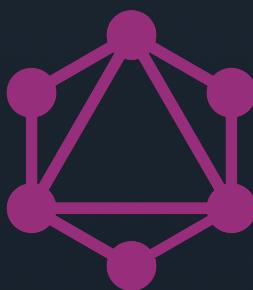
Data fetching works magically.



# Myth: Data fetching with resolvers, queries and mutations work magically.

You will be writing each of these on the server side.





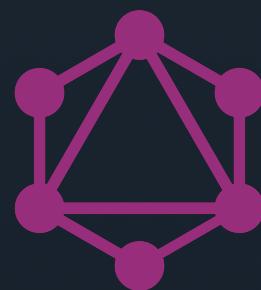
# GraphQL replaces Redux

# GraphQL replaces Redux

2. Literally: "GraphQL replace Redux". "Why bother with GraphQL if I've been doing same with SQL". "GraphQL is a fancy JS library"



@shrutikapoor08



# Myth: GraphQL replaces Redux

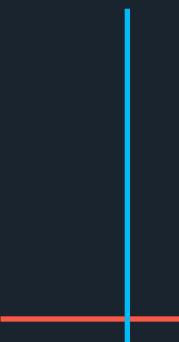
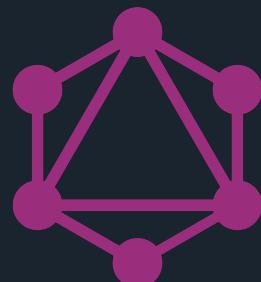
Redux is a state management library

GraphQL is a specification for calling APIs.

GraphQL reduces the data you get, but it is not a way of managing data.

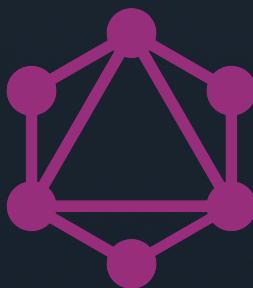
You will still need to manage data.

Libraries to use - Apollo client, React Hooks, Redux





# You can't have REST with GraphQL

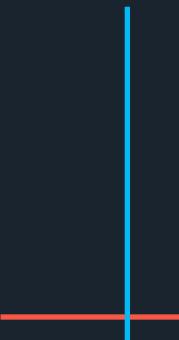
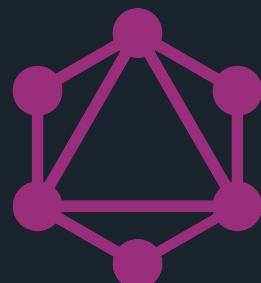


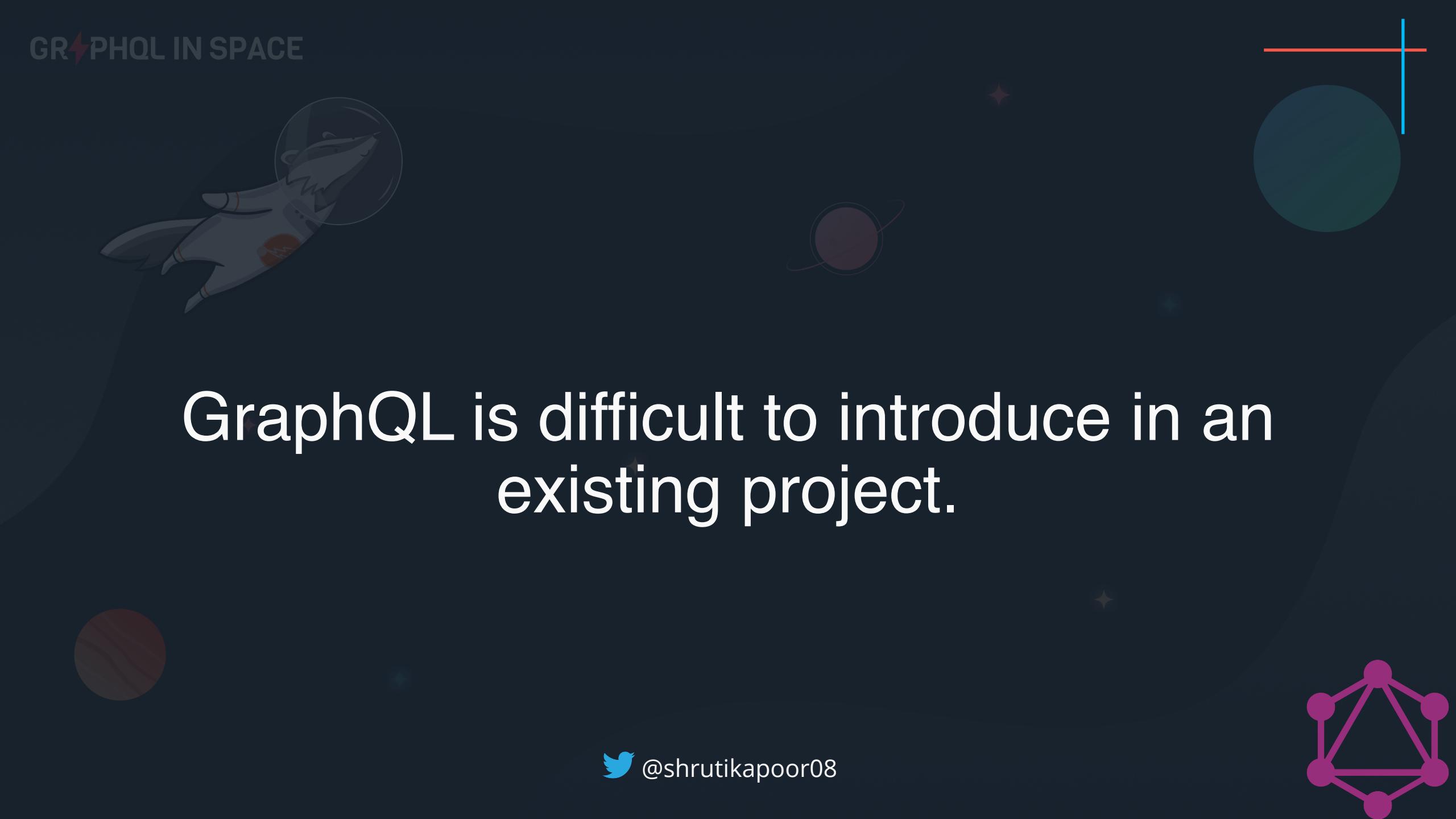
# Myth: You can't have REST endpoints

The beauty of GraphQL is that you can have talk to multiple APIs at the same time.

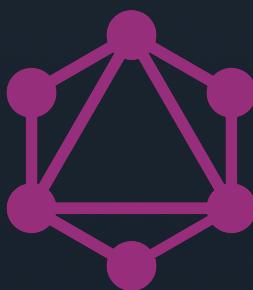
These APIs can be in REST, SOAP, or GraphQL.

Library: OneGraph





GraphQL is difficult to introduce in an existing project.



2. That GraphQL is all or nothing and they would need to replace/rewrite their entire REST API in order to introduce GraphQL into their product

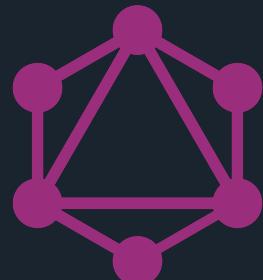
# GraphQL is difficult to introduce in an existing project.

1. GraphQL is meant for green field project and it is difficult to introduce in an existing project.
2. GraphQL is only for frontend folks

# Myth: GraphQL is difficult to introduce in an existing project.

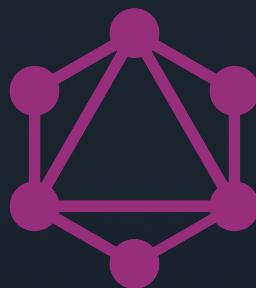
You can start incrementally - one endpoint at a time

One component at a time





# GraphQL is only for front-end folks.

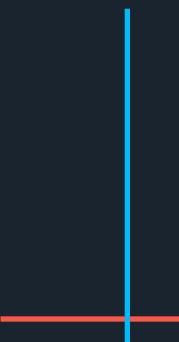
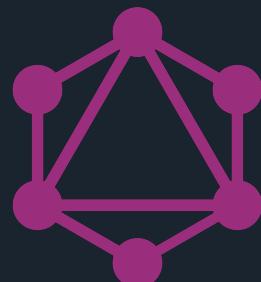


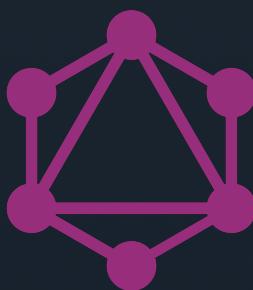
# Myth: GraphQL is for front-end folks.

The real beauty of GraphQL comes from the back-end.

Front end: Orchestration layer

Back end: Add fields to existing API





# GraphQL will write database queries itself



# GraphQL will write database queries itself

You send in a complex query, and some magical engine works out all of the subqueries to get the data for it, ties it all together and sends it back to the client. Amazing! Only now, two years later do I realise that you basically have to write all that clever stuff yourself. 😞

1

1

24

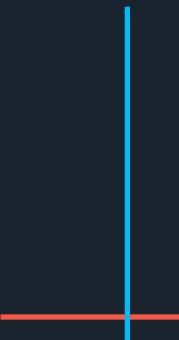
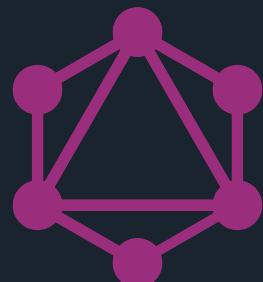
↑ ↓

1. GraphQL will write database queries itself, i just need to specify schemas and the relation between them

# Myth: GraphQL will write database queries itself.

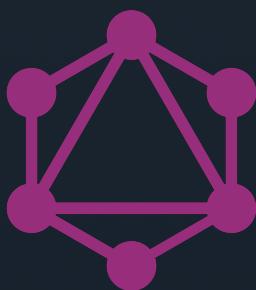
You will be writing all the logic of database queries

Unless if you are using a library that does it for you - Neo4J, Prisma





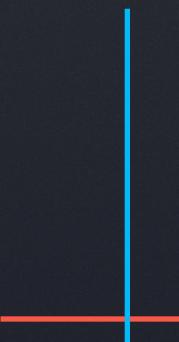
It requires complicated clients





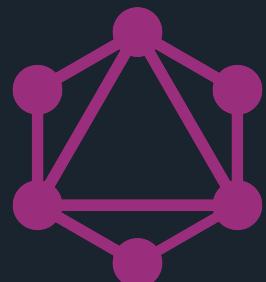
## It requires complicated clients

2. Requesting data from a GraphQL server is hard and requires a 3rd party client which is problematic in some languages. A lot of people forget it's just an HTTP request.



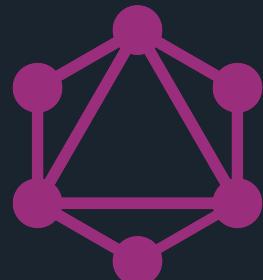
# Myth: It requires complicated clients

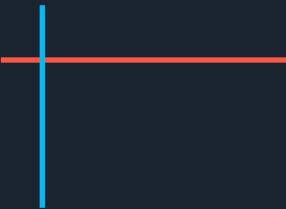
```
curl \
-X POST \
-H "Content-Type: application/json" \
--data '{ "query": "{ posts { title } }" }' \
https://myapi.com/graphql
```



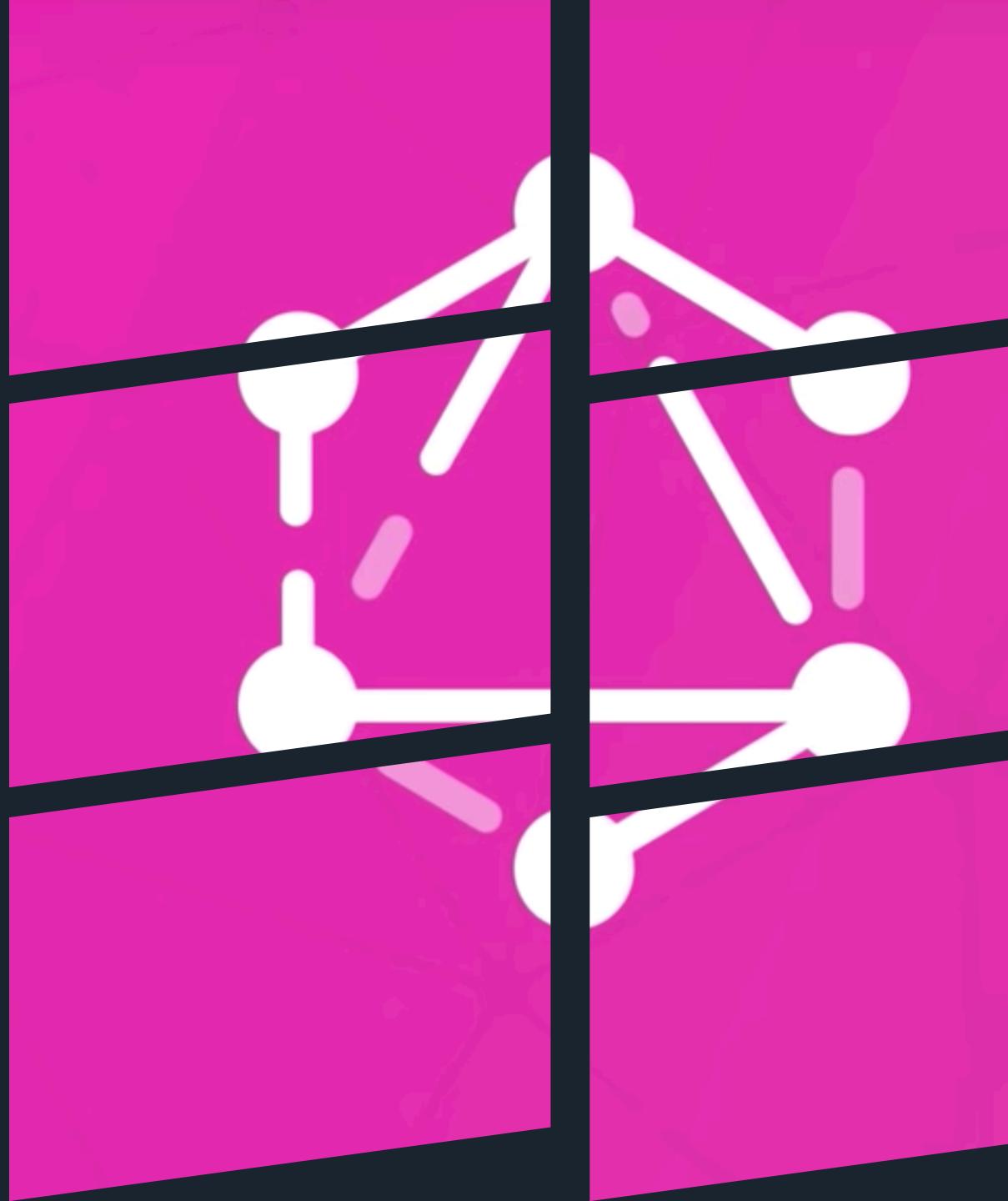
# Common Myths

1. GraphQL = Graph
2. GraphQL = SQL
3. GraphQL only works with graph type data.
4. Data fetching works magically.
5. GraphQL replaces Redux
6. REST and GraphQL can't co-exist.
7. GraphQL is difficult to introduce in an existing project
8. GraphQL is useful only for front-end
9. GraphQL writes database queries itself.
10. GraphQL requires complicated clients.





# Why I like GraphQL



# Why I like GraphQL

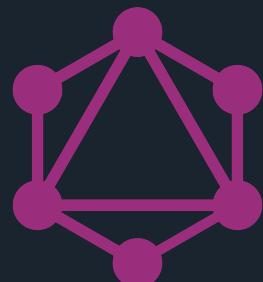
Type definition

Contract between front-end and back-end

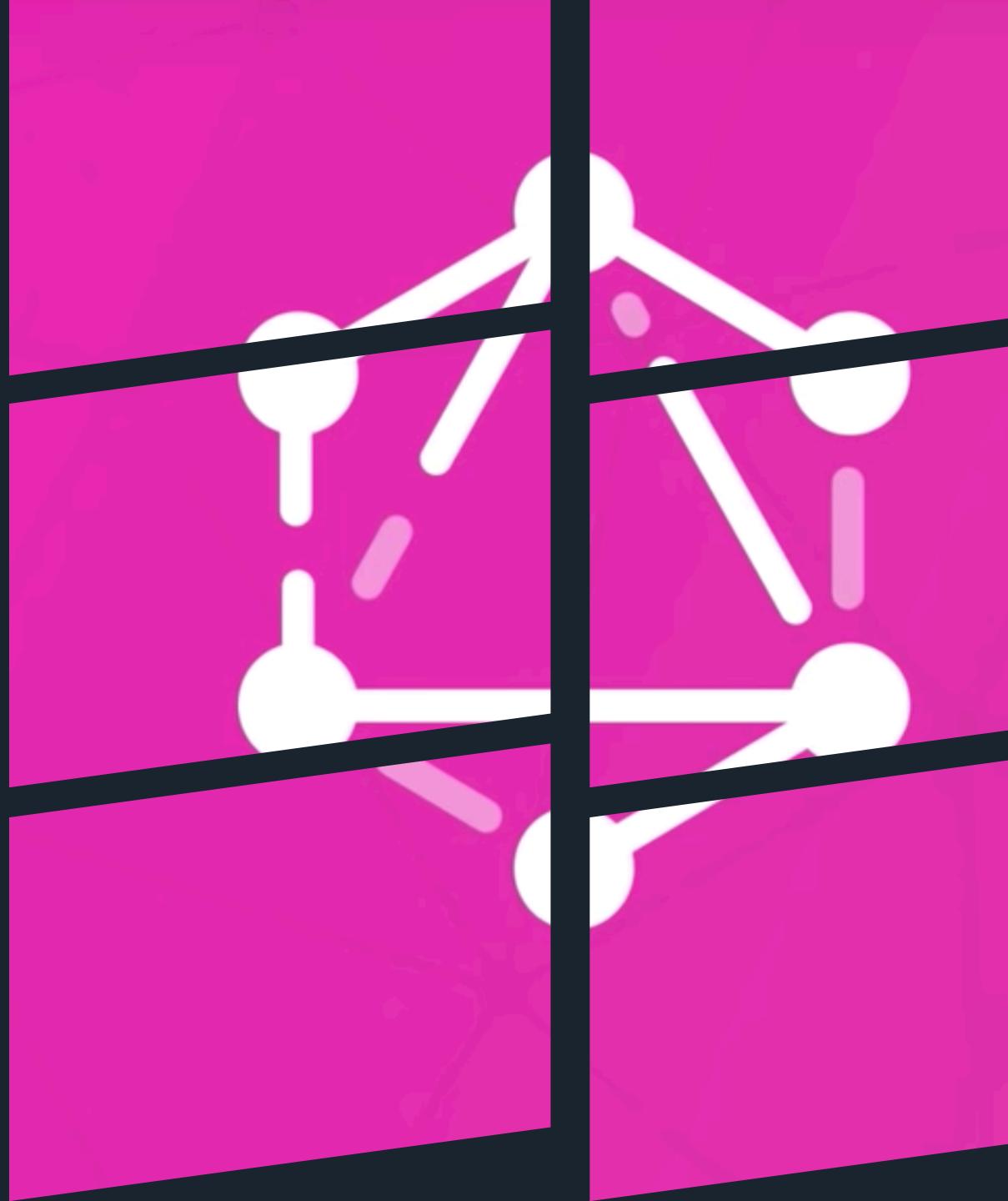
Analytics over data and fields being used

Avoid over fetching and under fetching

One endpoint to rule them all



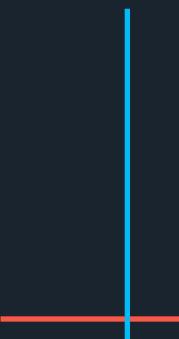
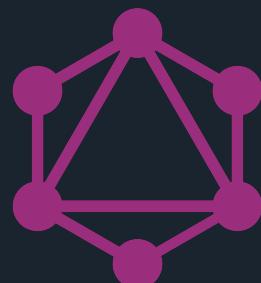
# How to get started with GraphQL



# How to get started

## What is GraphQL

- Official Docs - [graphql.org](https://graphql.org)
- Apollo Docs - <https://www.apollographql.com/docs/>
- Prisma Docs - <https://www.howtographql.com/>





```

1 fragment PeopleInfo on PlanetResidentsConnection {
2   residents {
3     name
4     birthYear
5     hairColor
6   }
7 }
8
9 fragment PlanetInfo on Planet {
10   id
11   name
12   climates
13   gravity
14   population
15   residentConnection {
16     ...PeopleInfo
17   }
18 }
19
20 query Planets($firstN: Int) {
21   allPlanets(first: $firstN) {
22     totalCount
23     planets {
24       ...PlanetInfo
25     }
26   }
27 }
28

```

## QUERY VARIABLES

```

1 {
2   "firstN": 2
3 }

```

```

{
  "data": {
    "allPlanets": {
      "totalCount": 61,
      "planets": [
        {
          "id": "cGxhbmV0czox",
          "name": "Tatooine",
          "climates": [
            "arid"
          ],
          "gravity": "1 standard",
          "population": 200000,
          "residentConnection": {
            "residents": [
              {
                "name": "Luke Skywalker",
                "birthYear": "19BBY",
                "hairColor": "blond"
              },
              {
                "name": "C-3PO",
                "birthYear": "112BBY",
                "hairColor": "n/a"
              },
              {
                "name": "Darth Vader",
                "birthYear": "41.9BBY",
                "hairColor": "none"
              },
              {
                "name": "Owen Lars",
                "birthYear": "52BBY",
                "hairColor": "brown, grey"
              },
              {
                "name": "Beru Whitesun Lars",
                "birthYear": "47BBY",
                "hairColor": "brown"
              },
              {

```

Q.v

allVehicles

vehicle

## OTHER RESULTS

PersonVehiclesConnection

PersonVehiclesEdge

Vehicle

VehiclePilotsConnection

VehiclePilotsEdge

VehicleFilmsConnection

VehicleFilmsEdge

FilmVehiclesConnection

FilmVehiclesEdge

VehiclesConnection

VehiclesEdge

\_InputValue

\_EnumValue

\_Directive

\_DirectiveLocation

PageInfo.hasNextPage

Film.vehicleConnection

Species.averageHeight

Species.averageLifespan

Planet.gravity

Person.vehicleConnection

# Try out GraphQL

[graphql / graphiql](https://github.com/graphql/graphiql)

Used by 3.4k Watch 206 Star 9.9k Fork 976

Code Issues 133 Pull requests 19 Actions Projects 8 Security Insights

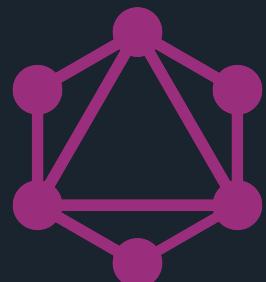
GraphQL & the GraphQL LSP Reference Ecosystem for building browser & IDE tools.

graphql graphql

2,541 commits 21 branches 0 packages 177 releases 126 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

<https://github.com/graphql/graphiql>



 @shrutikapoor08

# GraphQL with React: The Complete Developers Guide

Learn and master GraphQL by building real web apps with React and Node

BESTSELLER

★★★★★ 4.6 (6,072 ratings) 34,159 students enrolled

Created by Stephen Grider Last updated 12/2019

English CC English [Auto-generated], French [Auto-generated], [7 more](#)



GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.



The Best GraphQL Courses



GraphQL Query Language

Eve Porcello  
19 lessons · 30m



Designing GraphQL Schemas

Nik Graf  
16 lessons · 56m



COURSE • AWS  
Scalable Offline-Ready GraphQL Applications with AWS AppSync & React

nader dabit  
9 lessons · 31m

O'REILLY®



Learning  
GraphQL

DECLARATIVE DATA FETCHING FOR MODERN WEB APPS

Eve Porcello & Alex Banks

# <https://github.com/shrutikapoor08/blogs/>

master ▾

[blogs](#) / [AllAboutGraphQL](#) / [GettingStartedWithGraphQL.MD](#)

[Go to file](#)

...



shrutikapoor08 adding tutorials

Latest commit 0cc4843 on Apr 17 History

1 contributor

39 lines (31 sloc) | 2.5 KB

[Raw](#) [Blame](#)

## Getting started with GraphQL

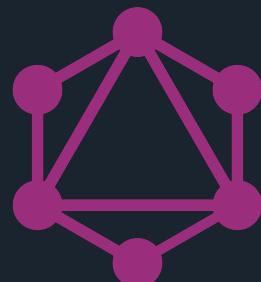
### What is GraphQL?

GraphQL is essentially a query language for your API. It is a way of asking for data from API. It offers a flexible way of asking for data, and you can get as much or as little data as you need.

A typical GraphQL implementation involves GraphQL on the server side (GraphQL API) and GraphQL on the client side (to talk to GraphQL APIs). A lot of different tools can be used to build a full-stack GraphQL application - such as Prisma, Apollo, Express, Hasura. At its core, it is just an API served over HTTP. You can send a curl command to talk to a GraphQL API without having to need any library. What is cool about GraphQL is that it provides an easy way of developing APIs - extending as well as deprecating over time. With GraphQL, you get one

# Outline

- 1.GraphQL vs REST
- 2.Misconceptions
- 3.The good things
- 4.How to get started





**Shruti Kapoor** @shrutikapoor08 · Aug 30

Heads up! I am declaring bankruptcy.

.

.

.

```
var bankruptcy;
```

#DevJoke

84

142

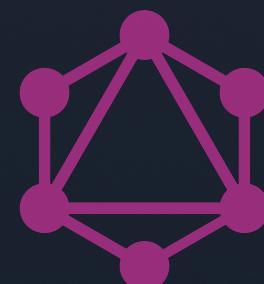
1.3K

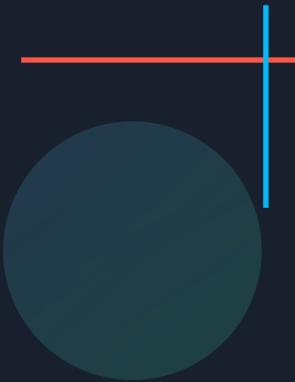
↑

|||



 @shrutikapoor08





# THANK YOU



@shrutikapoor08

[tinyletter.com/shrutikapoor08](https://tinyletter.com/shrutikapoor08)

