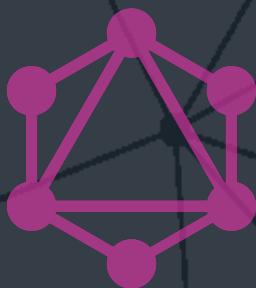


What Is(n't) GraphQL?

The Misconceptions



@shrutikapoor08

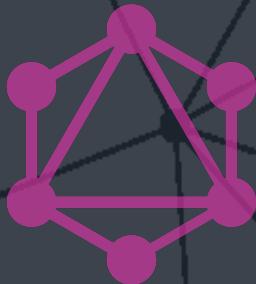




Shruti Kapoor



@shrutikapoor08

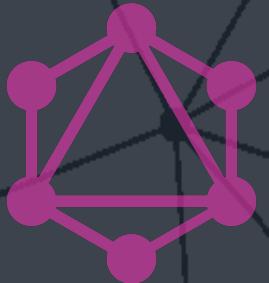


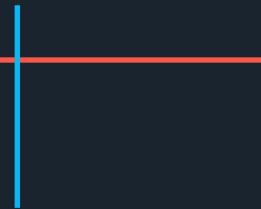


#DEVJOKE

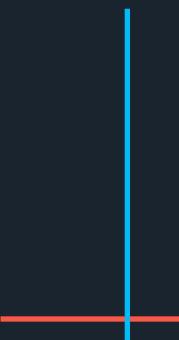
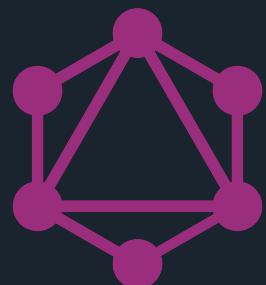


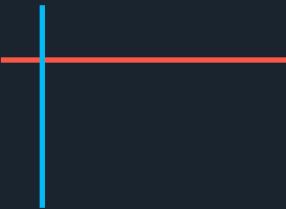
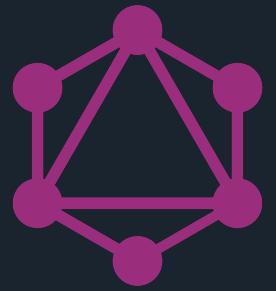
@shrutikapoor08



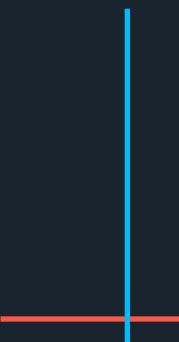


Question: Why did the JavaScript developer leave?

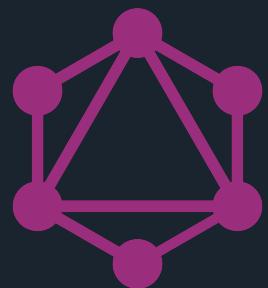




Because they didn't get arrays.

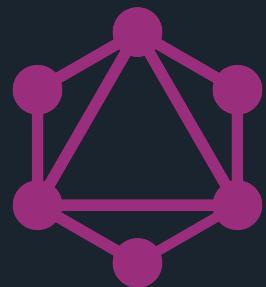


Question - Why did the developer die in the shower?

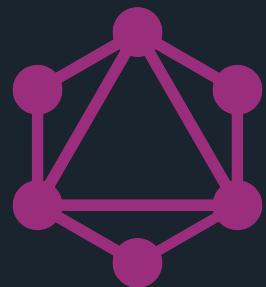




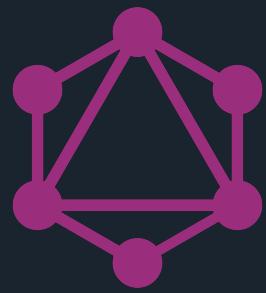
He read the shampoo bottle instructions:
Lather. Rinse. Repeat.



Question: What do you call the security outside of a Samsung store?



Guardians of Galaxy



rt You Retweeted



Shruti Kapoor @shrutikapoor08 · Jan 25

Since you asked, I shall deliver -

Question: Why do C# and Java developers keep breaking their keyboards ?

.

.

.

.

Because they use a strongly typed language

#Devjoke



4



6



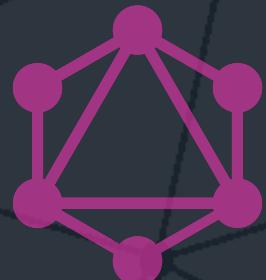
42



@shrutikapoor08

Outline

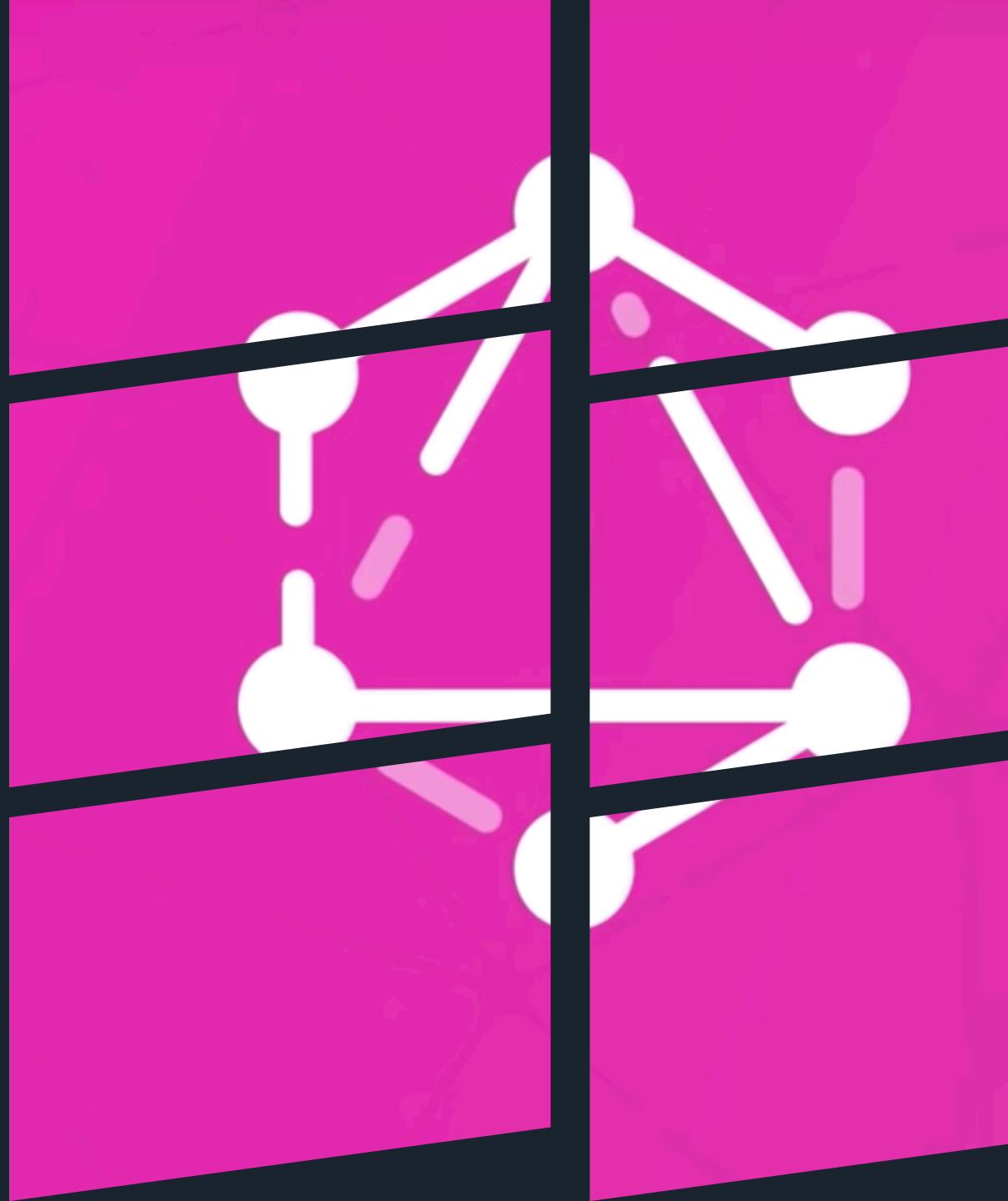
- 1.GraphQL vs REST
- 2.Misconceptions
- 3.The good things
- 4.How to get started



GraphQL

VS

REST



API and Client communication

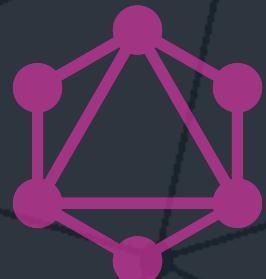
Step 1: Knowing what to send to server

Step 2: Creating a request

Step 3: API level - knowing what to send to client

Step 4: Building resources

Step 5: Handling data



@shrutikapoor08

API and Client communication

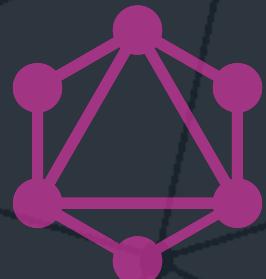
Step 1: Knowing what to send to server

Step 2: Creating a request

Step 3: API level - knowing what to send to client

Step 4: Building resources

Step 5: Handling data



@shrutikapoor08

API and Client communication

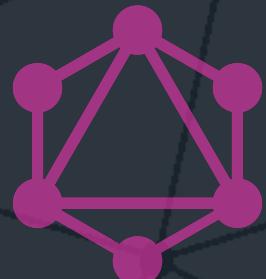
Step 1: Knowing what to send to server

Step 2: Creating a request

Step 3: API level - knowing what to send to client

Step 4: Building resources

Step 5: Handling data



@shrutikapoor08

API and Client communication

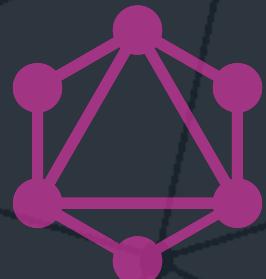
Step 1: Knowing what to send to server

Step 2: Creating a request

Step 3: API level - knowing what to send to client

Step 4: Building resources

Step 5: Handling data



@shrutikapoor08

API and Client communication

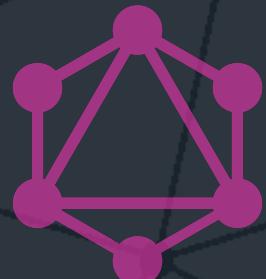
Step 1: Knowing what to send to server

Step 2: Creating a request

Step 3: API level - knowing what to send to client

Step 4: Building resources

Step 5: Handling data



@shrutikapoor08

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"  
      ]  
    }  
  ]  
}
```

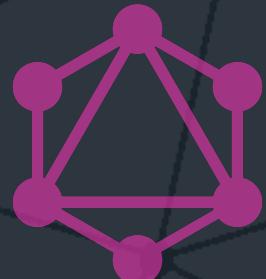
Step 1: Sending data to server

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



@shrutikapoor08

Step 2: Creating a request

REST

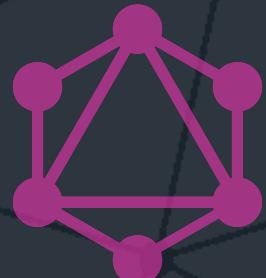
Endpoints: /books, /author

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 }



@shrutikapoor08

Step 3: Knowing what to send

REST

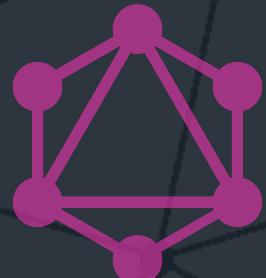
GET

POST

PATCH

DELETE

PUT



GraphQL

Query
Mutation



@shrutikapoor08

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 }



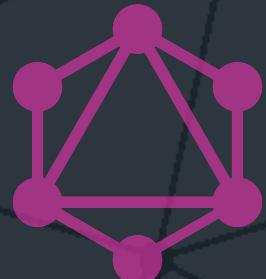
@shrutikapoor08

Step 5: Parsing received data

REST

JSON object

Fields in addition to the ones you
needed



GraphQL

JSON object

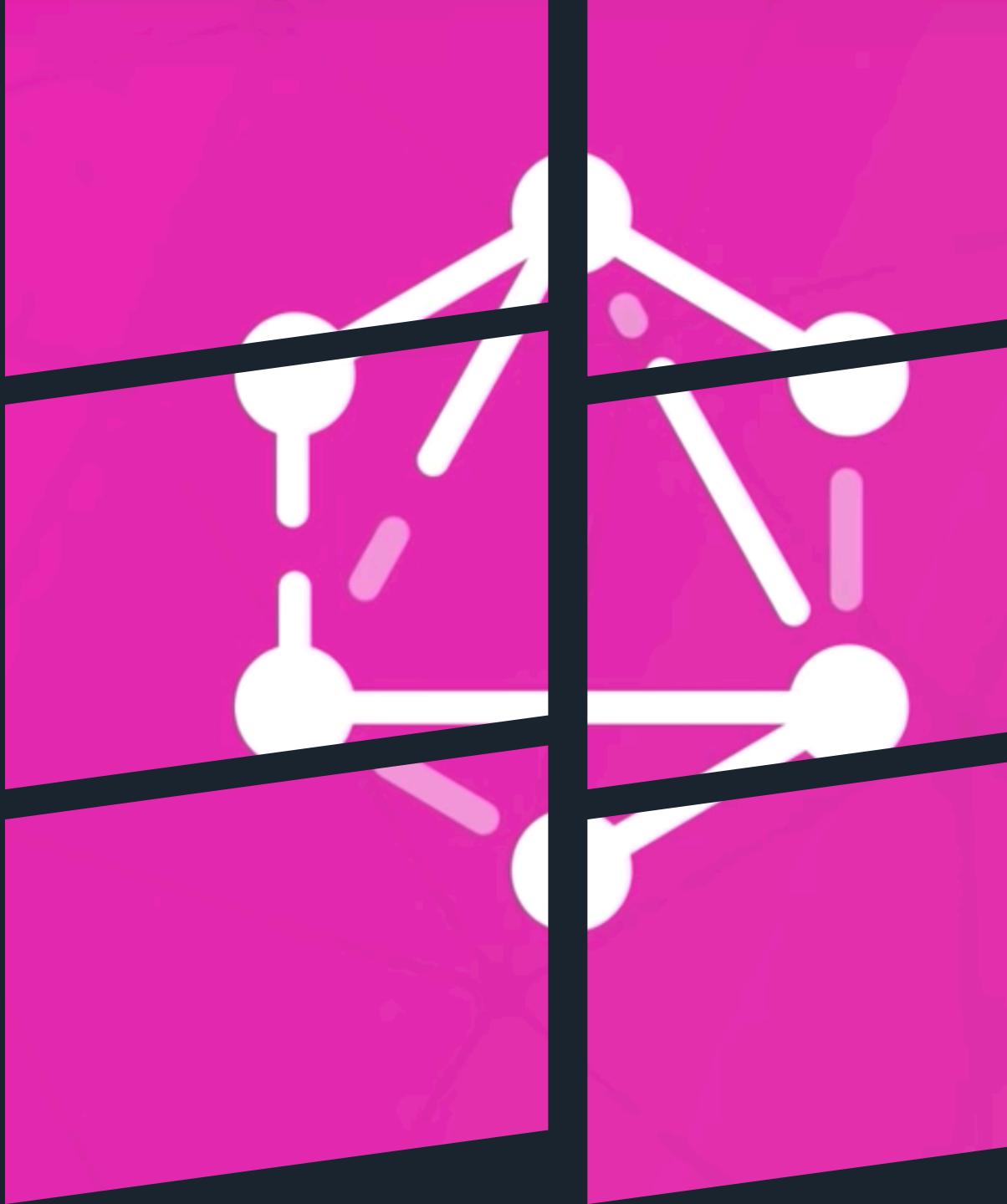
Only the fields you requested



@shrutikapoor08

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!



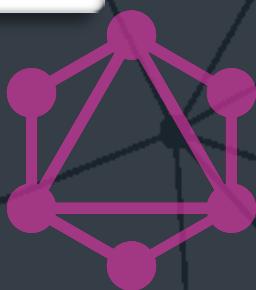
69

112

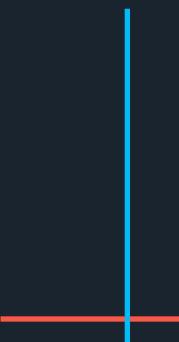
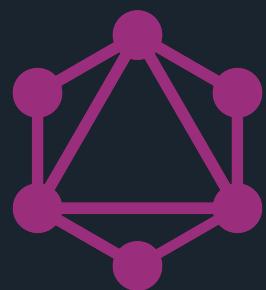
267



@shrutikapoor08



GraphQL = Graph

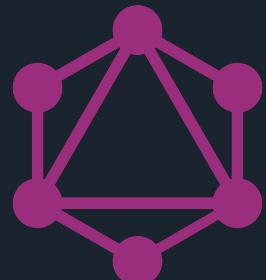


Though it enables GraphDatabases for the browser

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"

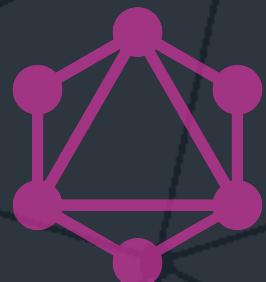


Myth: GraphQL only works with graph like structures.

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



@shrutikapoor08

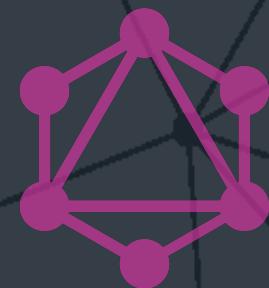


```
{  
  "books": [  
    {  
      "title": "The Mythical Man Month",  
      "author": [  
        "Frederik P. Brooks, Jr."  
      ]  
    },  
    {  
      "title": "The Pragmatic Programmer",  
      "author": [  
        "Andy Hunt",  
        "Dave Thomas"  
      ]  
    },  
    {  
    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

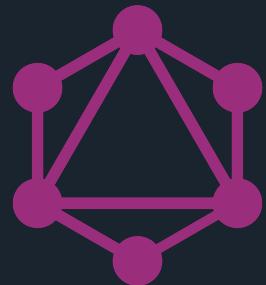


@shrutikapoor08

Name
Email
Bio

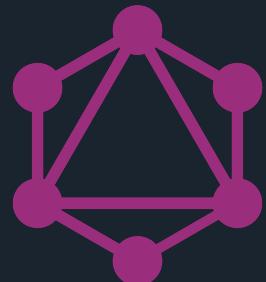


GraphQL = SQL



GraphQL = SQL

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



Myth: QL = SQL.

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



@shrutikapoor08

Myth: QL = SQL.

“Get Book with Title - The Pragmatic Programmer”



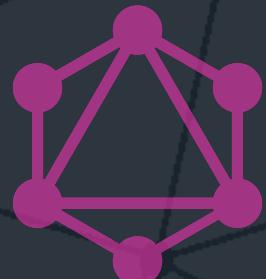
@shrutikapoor08

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

GraphQL is a query language for API.

It is not a database language.

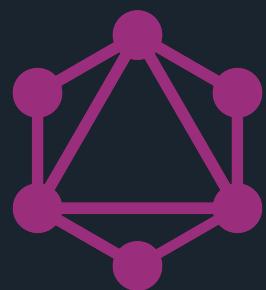
GraphQL is typically used to fetch data from API, however some libraries like DGraph use GraphQL to get data from the database.



@shrutikapoor08

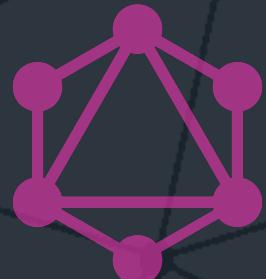


Data fetching works magically.



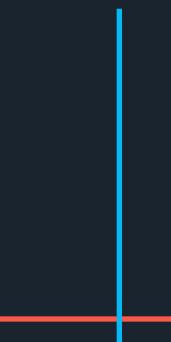
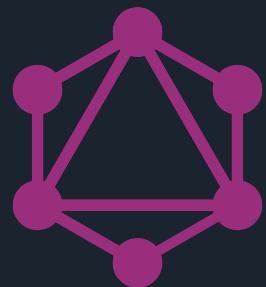
Myth: Data fetching with resolvers, queries and mutations work magically.

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



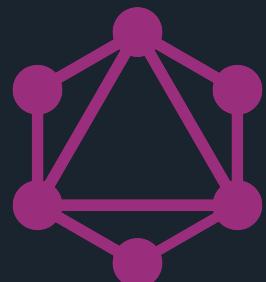
@shrutikapoor08

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"



Myth: GraphQL replaces Redux

Redux is a state management library

GraphQL is a specification for calling APIs.

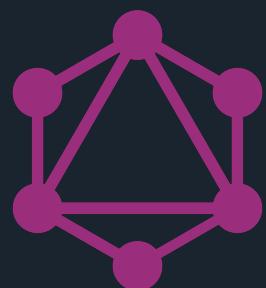
GraphQL reduces the data you get, but it is not a way of managing data.

You will still need to manage data.

Libraries to use - Apollo client, React Hooks, Redux



You can't have REST with GraphQL

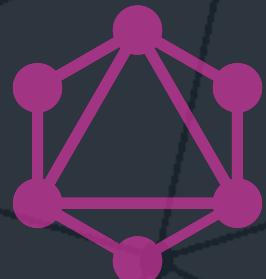


Myth: You can't have REST endpoints

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.

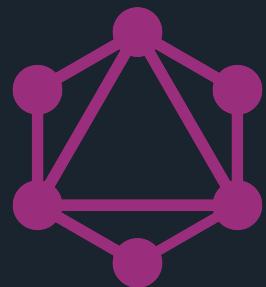
Library: OneGraph

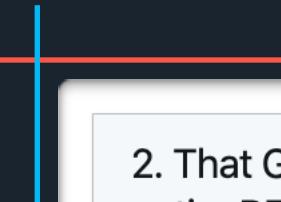


@shrutikapoor08

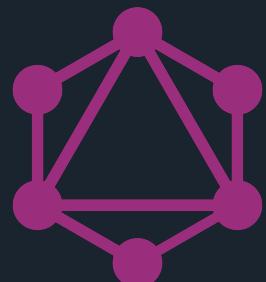


GraphQL is difficult to introduce in an existing project.



- 
- 
- 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.

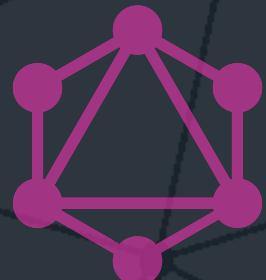


- 
- GraphQL is meant for green field project and it is difficult to introduce in an existing project.
 - GraphQL is only for frontend folks

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

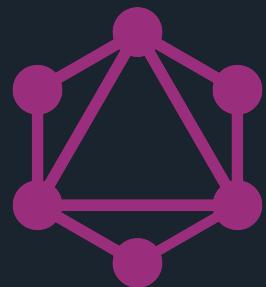
You can start incrementally - one endpoint at a time

One component at a time



@shrutikapoor08

GraphQL is only for front-end folks.

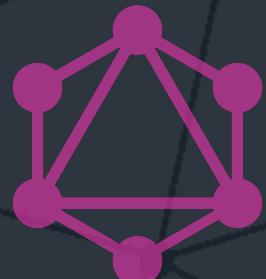


Myth: GraphQL is for front-end folks.

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

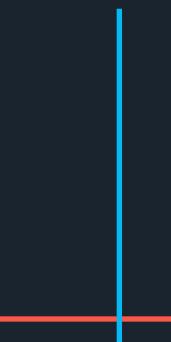
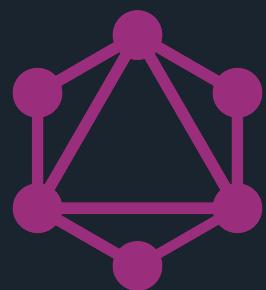
Front end: Orchestration layer

Back end: Add fields to existing API



@shrutikapoor08

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

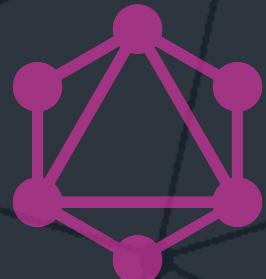
↑ ↓

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



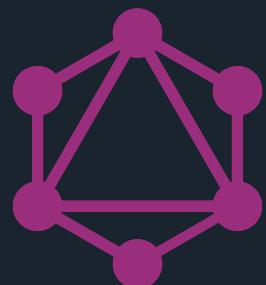
@shrutikapoor08



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.

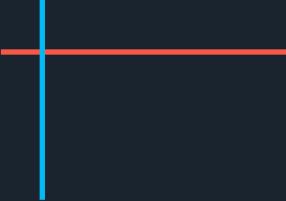


Myth: It requires complicated clients

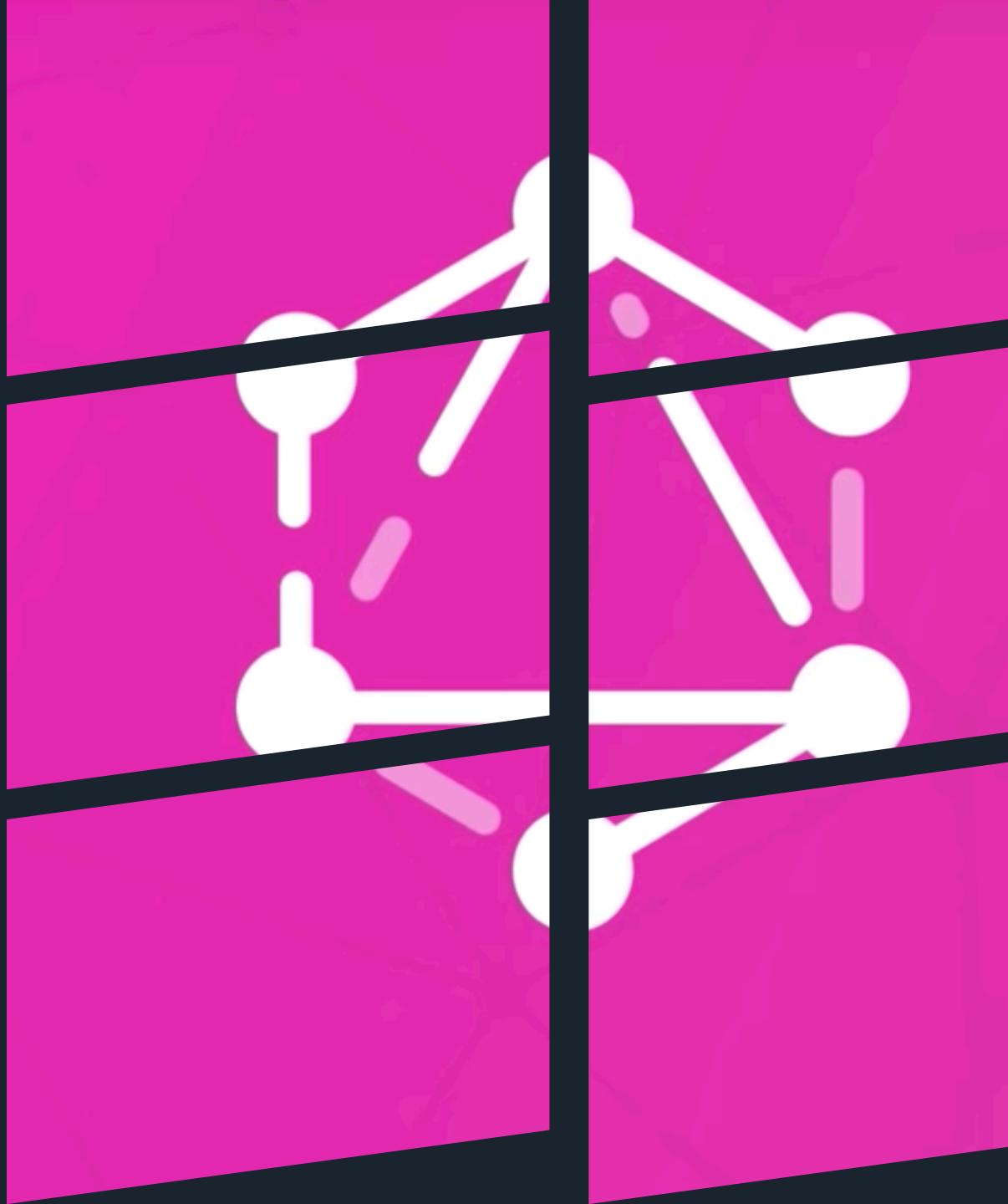
```
curl \
-X POST \
-H "Content-Type: application/json" \
--data '{ "query": "{ posts { title } }" }' \
https://myapi.com/graphql
```



@shrutikapoor08



Why I Like GraphQL



Why I like GraphQL

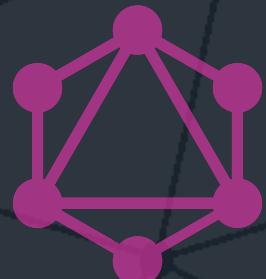
Type definition

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

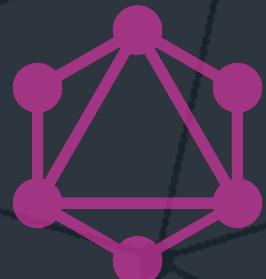


@shrutikapoor08

How to get started

What is GraphQL

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



@shrutikapoor08



```

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](#)

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

graphiql

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



COURSE • GRAPHQL

GraphQL Query Language

Eve Porcello
19 lessons · 30m



COURSE • GRAPHQL

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®



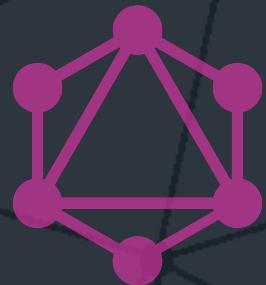
Learning GraphQL

DECLARATIVE DATA FETCHING FOR MODERN WEB APPS

Eve Porcello & Alex Banks

Outline

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





Shruti Kapoor @shrutikapoor08 · Dec 11, 2019

I am declaring a war!

var war;

#DevJoke

180

779

3.9K



@shrutikapoor08

THANK YOU

@shrutikapoor08

