Getting started with GraphQL

In an enterprise

Shruti Kapoor

 @shrutikapoor08

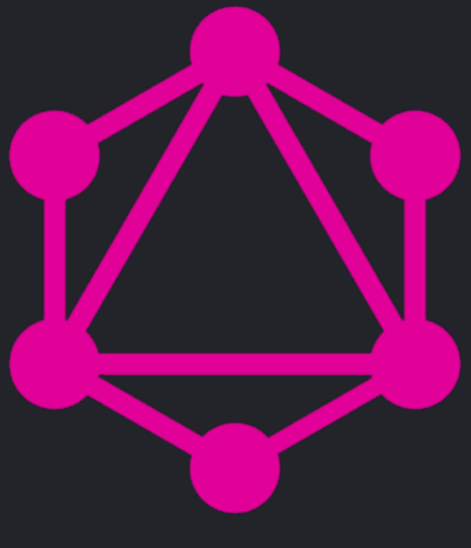
Outline

Who is using GraphQL? Why should you care about it?

Our story of adopting GraphQL in enterprise

Lessons we learnt

How you can get started



Shruti Kapoor

@shrutikapoor08

twitch.tv/shrutikapoor
@ 9AM PST

What is(n't) GraphQL - the misconceptions.

With GraphQL, you get exactly what you asked for. It is great for getting multiple resources in one request. In this issue of JSByte, I will address some common misconceptions about GraphQL.

Myth 1: GraphQL only works with graph like structures.

Reality: GraphQL can be used to query a graph database, but it is not it's only use case. The "graph" in GraphQL is used to represent the graph-like structure of data.

Myth 2: GraphQL only works with databases or data sources that are graph based.

Reality: GraphQL can be a wrapper around any data source, including databases. GraphQL is a `query language for your API` - which means it is a syntax of how to ask for data.

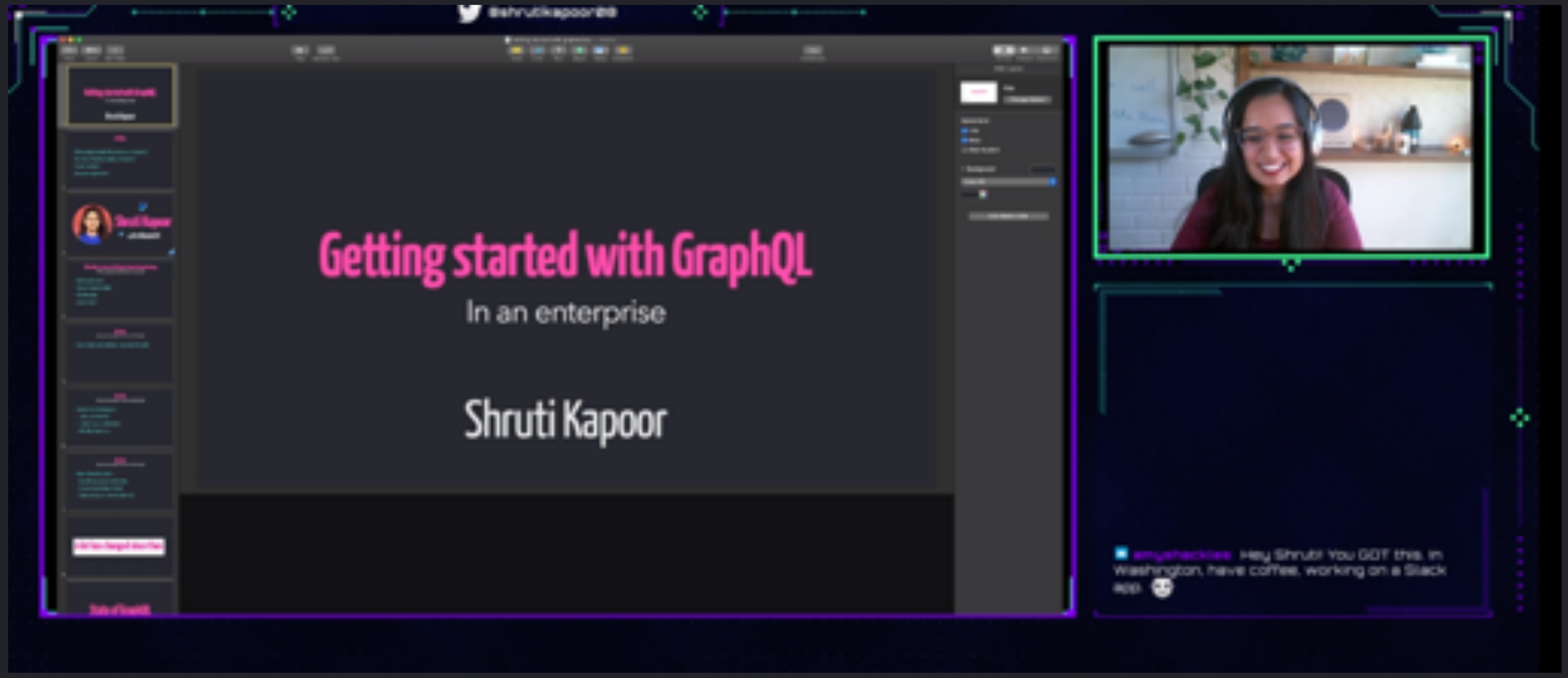
bit.ly/shrutinewsletter

On a personal note...

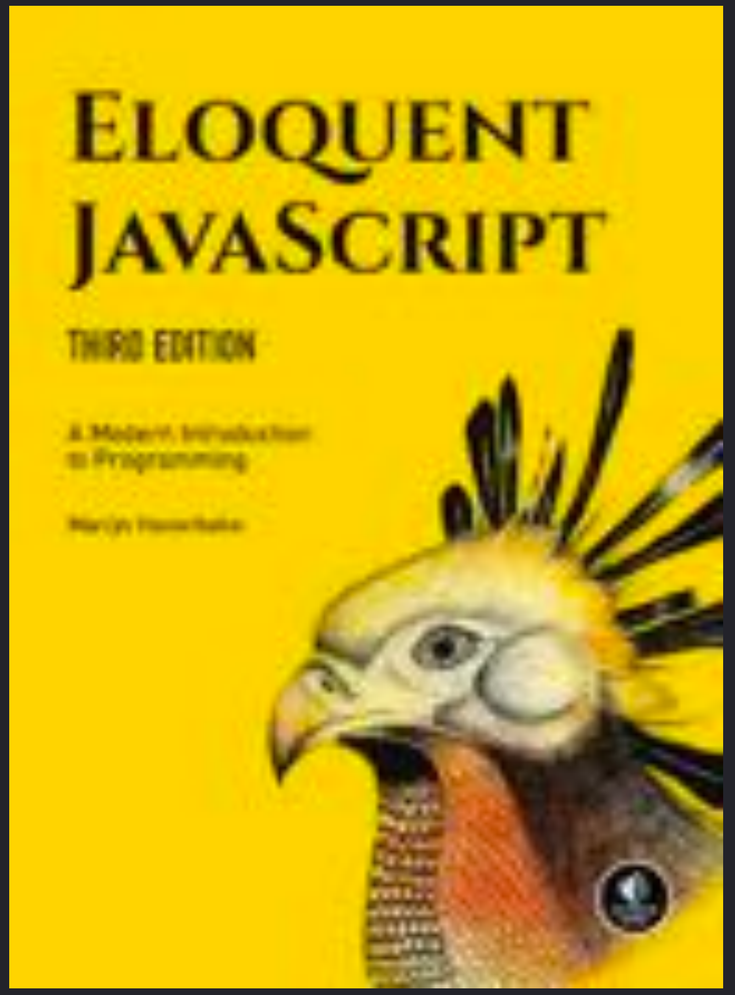
“Pandemic fatigue” is real

I miss normal times

twitch.tv/shrutikapoor
Tue @ 9:00 AM PST

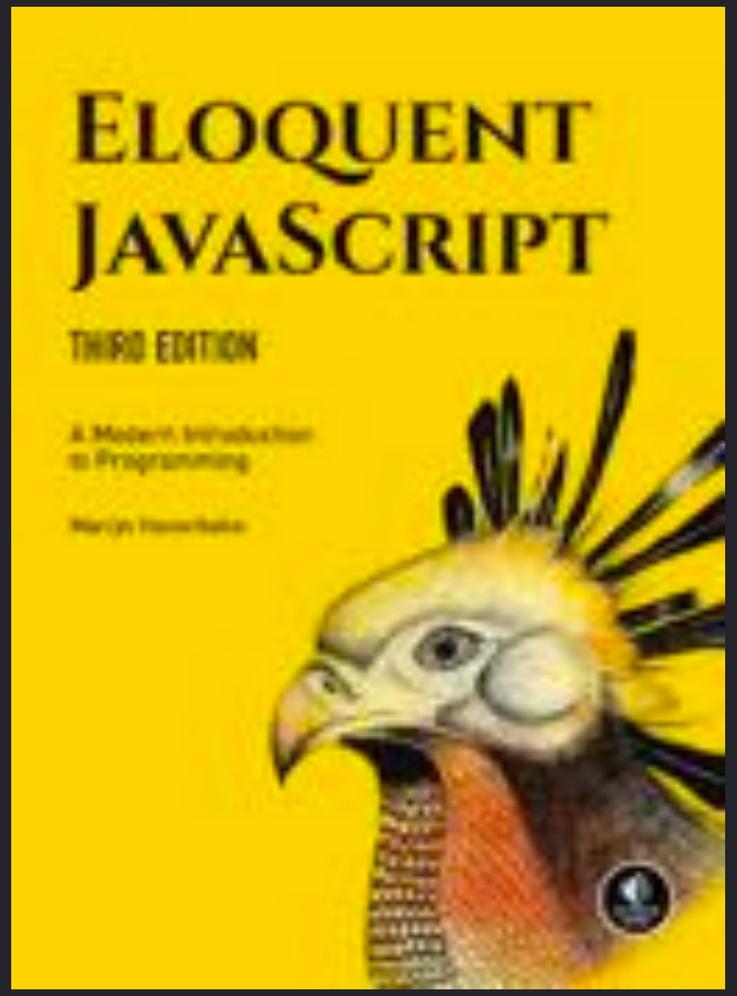


GIVEAWAY!


ELOQUENT
JAVASCRIPT
THIRD EDITION
A Modern Introduction to Programming

Step1: Tweet
Step 2: Win

@shrutikapoor08



Getting started with GraphQL

In an enterprise

@shrutikapoor08

@shrutikapoor08

Me before PayPal

My role

Started in an internal project

uses <5 internal APIs

1 owned by us, 4 other teams

REST APIs, NodeJS app

State of GraphQL in 2018

GraphQL was used by 1 other team.

No centralized tooling at PayPal

Slowly picking up in outside community

A lot has changed since then


Spec
Tools
Services
Databases
Projects
GraphQL
GraphQL Foundation
Services
Databases
GraphQL Adopter
GraphQL Adopter
General
GraphQL Foundation Member
https://landscape.graphql.org/
l.graphql.org
GraphQL Landscape
GraphQL Foundation
This landscape is intended as a map to explore the GraphQL community and also shows the member organizations of the GraphQL Foundation.

graphql / graphql-js

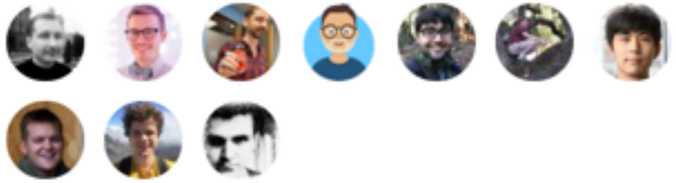
Unwatch 412 | Unstar 17.3k | Fork 1.8k

Code | Issues 115 | Pull requests 57 | Actions | Projects 1 | Wiki | Security | Insights

Used by 437k

+ 437,431

Contributors 165



+ 154 contributors

https://github.com/graphql/graphql-js

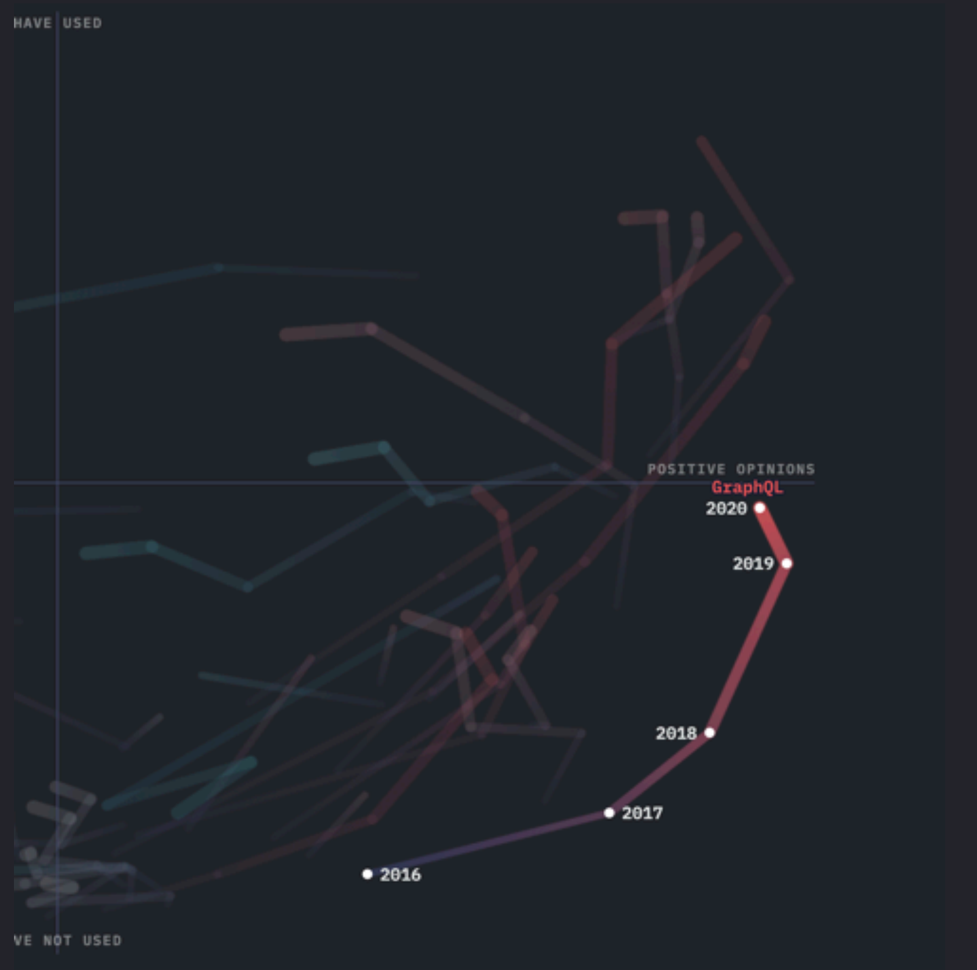
@shrutikapoor08

Highest Interest

Awarded to the technology developers are most interested in learning once they are aware of it.

GraphQL

GraphQL continues to be the one thing developers want to learn more about with an interest ratio of **86%**, as soon as they can finally find the time.



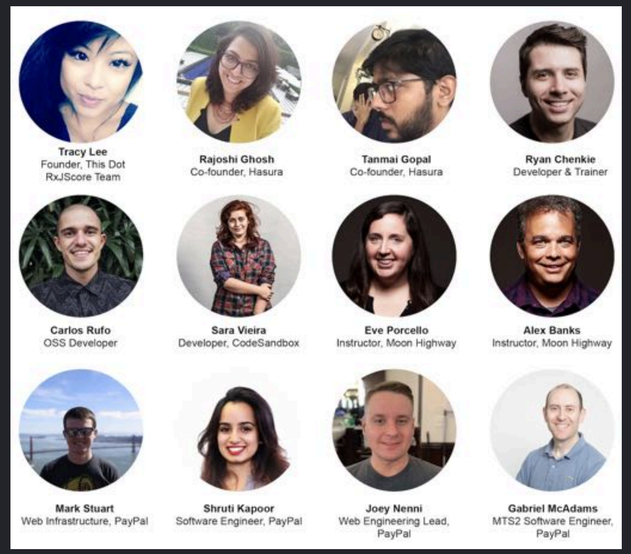
https://2020.stateofjs.com/en-US/awards/

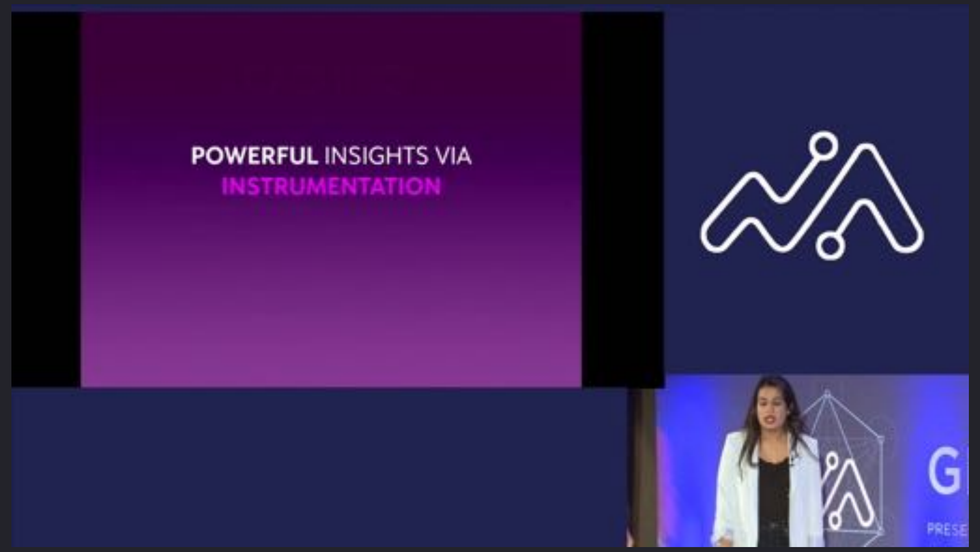
@shrutikapoor08

33 apps in production using GraphQL
Braintree public API
> 500 members in GraphQL community
Internal tools
Supported by infra teams

@shrutikapoor08

Member of GraphQL Foundation
Active community member





@shrutikapoor08

GraphQL Tools

Tools

Express-graphql

Apollo

Prisma

Hasura

DGraph

GraphiQL

```
query TodoAppQuery($n: Int) {
  globalTodoList {
    items(first:$n) {
      edges {
        node {
          text
          complete
        }
      }
    }
  }
}
```

QUERY VARIABLES

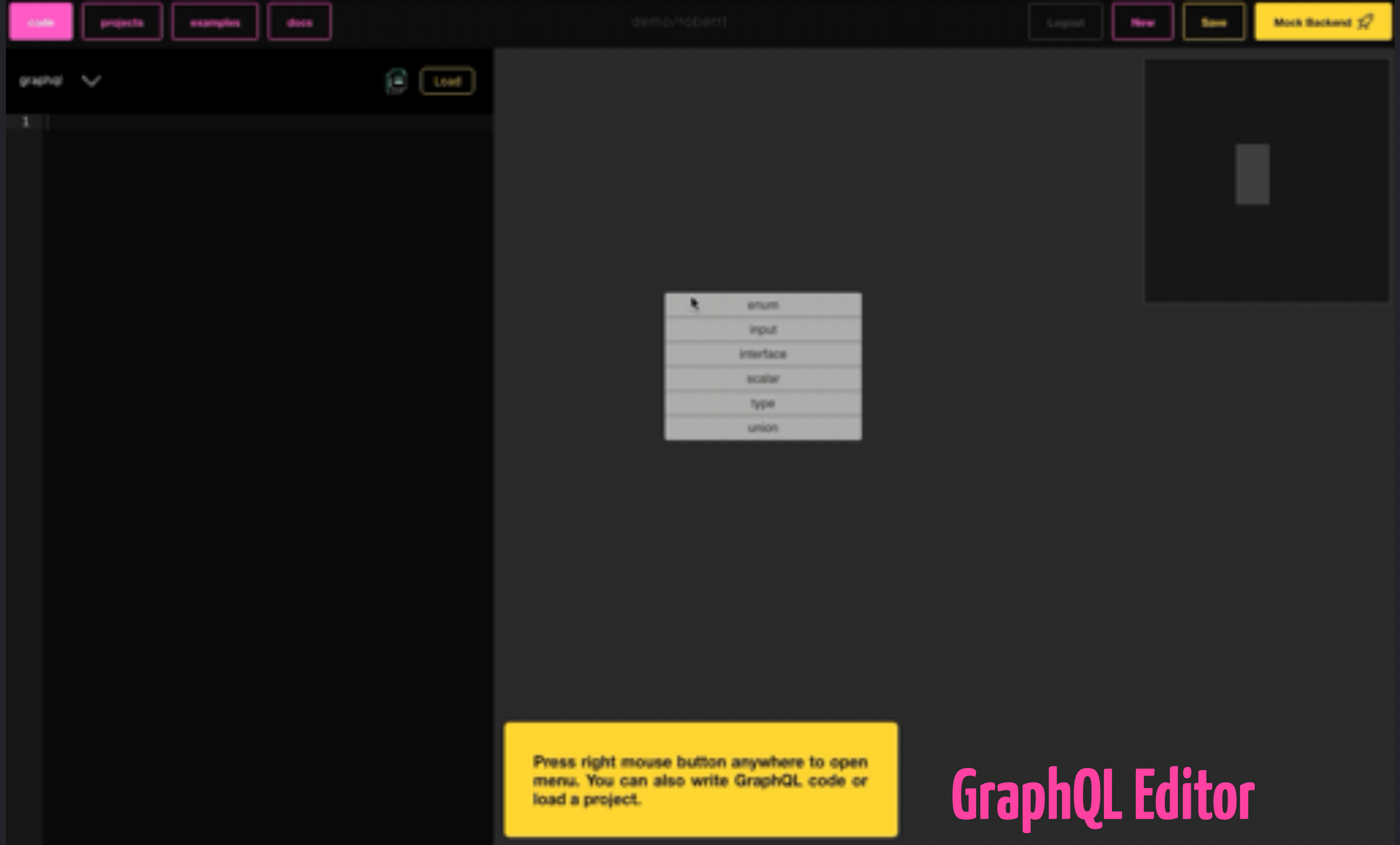
```
{
  "n": 2
}
```

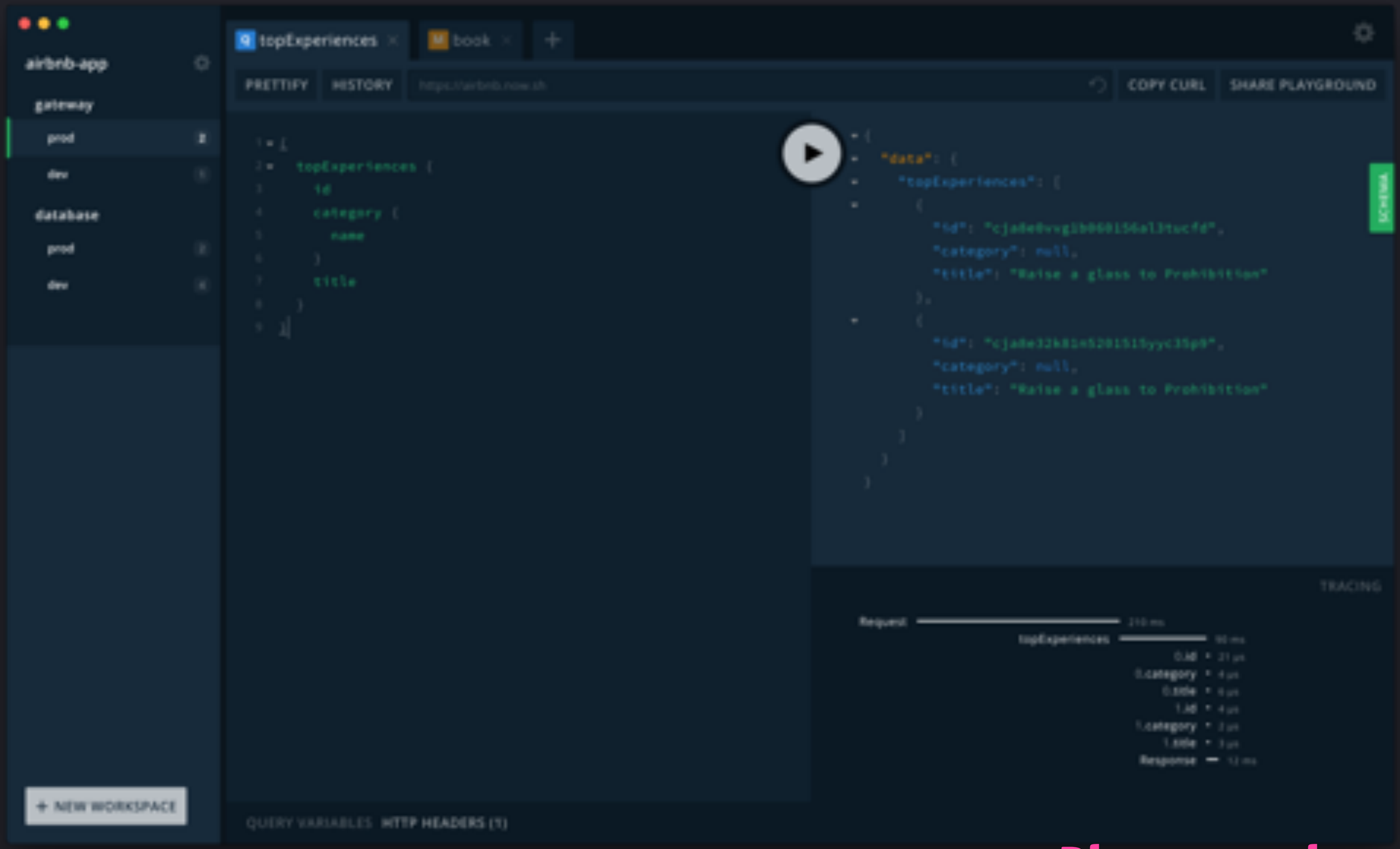
```
{
  "data": {
    "globalTodoList": {
      "items": {
        "edges": [
          {
            "node": {
              "text": "Release GraphiQL",
              "complete": true
            }
          },
          {
            "node": {
              "text": "Attend @Scale 2015",
              "complete": false
            }
          }
        ]
      }
    }
  }
}
```

GraphiQL

Try Now

https://graphql.org/swapi-graphql


code
projects
examples
docs
demo/robert
Logout
New
Save
Mock Backend
graphql
Load
1
enum
input
interface
scalar
type
union
Press right mouse button anywhere to open menu. You can also write GraphQL code or load a project.
GraphQL Editor


airbnb-app
gateway
prod
dev
database
prod
dev
+ NEW WORKSPACE
topExperiences
book
PRETTIFY
HISTORY
COPY CURL
SHARE PLAYGROUND
SCHEMA
TRACING
QUERY VARIABLES
HTTP HEADERS (1)

Playground


GRAPH A WHA?
MAKES NO SENSE

Graph | QL

@shrutikapoor08

Slide by Mark Stuart, PayPal

Graph

QL

Graph because data is represented as a graph.

QL for query language, not a SQL like technology

A query language for APIs

@shrutikapoor08

> A query language for APIs

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.

> A query language for APIs

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.

Getting data with REST

/customers/account

Getting data with REST

/customers/account

/auth

/customers/user/{id}

/customers/account-details

/customers/user/{id}/notifications

Getting data with REST

/customers/account

/auth

/customers/user/{id}

/customers/account-details

/customers/user/{id}/notifications

Getting data with REST

/customers/account

/auth

/customers/user/{id}

/customers/account-details

/customers/user/{id}/notifications

Getting data with REST

/customers/account 

/auth ✓

/customers/user/{id} ✓

/customers/account-details ✓

/customers/user/{id}/notifications ✓

Getting data with REST

/customers/account

/auth

/customers/user/{id}

/customers/account-details

/customers/user/{id}/notifications

Getting data with GraphQL

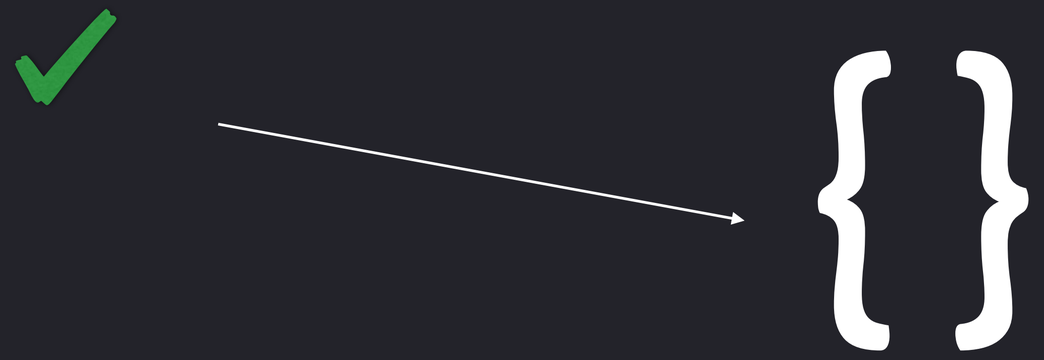
paypal.com/graphql

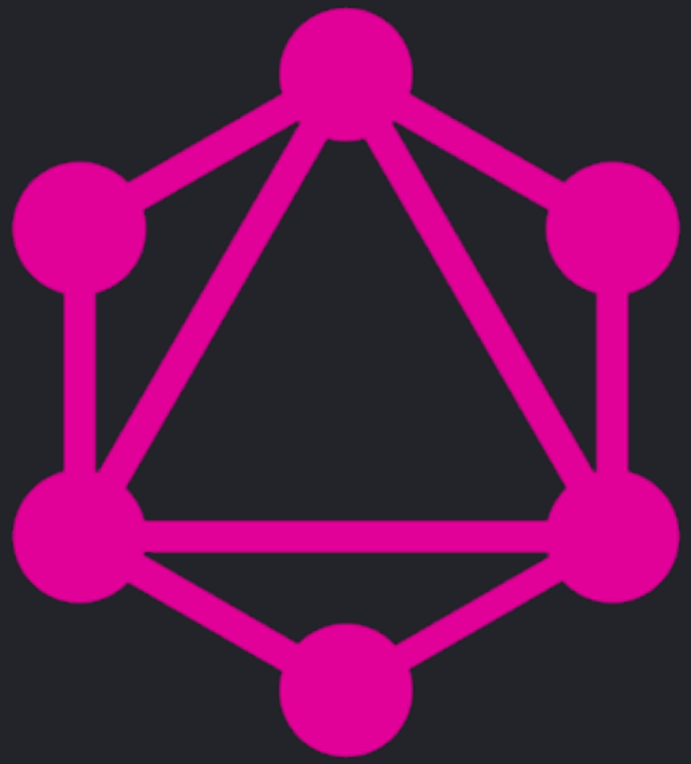
Getting data with GraphQL

paypal.com/graphql

Getting data with GraphQL

paypal.com/graphql





GraphQL Terminology

@shrutikapoor08

Schema
Queries
Mutations
Resolvers

Schema

a set of types which completely describe the set of possible data you can query on that service

Schema

```
type Character {
  name: String!
  appearsIn: [Episode!]!
}
```

```
type Starship {
  id: ID!
  name: String!
  length(unit: LengthUnit = METER): Float
}
```

Query

Query

```
query {
  hero {
    name
  }
  droid(id: "2000") {
    name
  }
}
```

```
{
  "data": {
    "hero": {
      "name": "R2-D2"
    },
    "droid": {
      "name": "C-3PO"
    }
  }
}
```

Query GET

Query	GET
Mutation	POST / DELETE

Resolvers

A function that “resolves” what data should be given back for a field

Every field has a resolver.

Resolvers

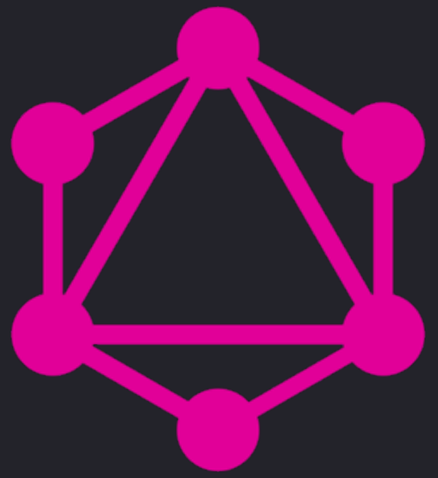
```
type User {
  # A User's first/given name (Ex: 'Marty')
  givenName: String

  # A User's last/family name. (Ex: 'McFly')
  familyName: String

  """
  A User's addresses.

  By default, returns all addresses.
  Optionally, passing a 'type' returns addresses of that type.
  """
  addresses(type: AddressType): [Address]
}
```

Resolvers

```
type User {
  # A User's first/given name (Ex: 'Marty')
  givenName: String

  # A User's last/family name. (Ex: 'McFly')
  familyName: String

  """
  A User's addresses.

  By default, returns all addresses.
  Optionally, passing a 'type' returns addresses of that type.
  """
  addresses(type: AddressType): [Address]
}
```

```
export default {
  Query: {
    user: async (root, args, { req }) => await getUser(req)
  },
  User: {
    givenName: ({ givenName }) => givenName,

    // By the way, these are optional ^^.
    // graphql-js has default resolvers
    familyName: ({ familyName }) => familyName,

    addresses: ({ addresses }, { type }) => {
      if (type) {
        return addresses.filter(address => address.type === type);
      }

      return addresses;
    }
  }
}
```

Features

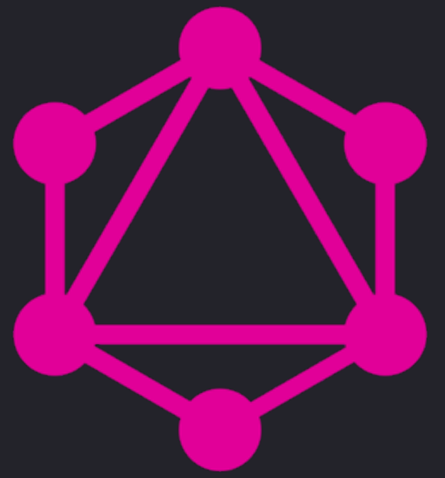


An open source specification

Clients ask for exactly the data they need, no more, no less.

Clients specify the structure of data they need

Features



No versioning

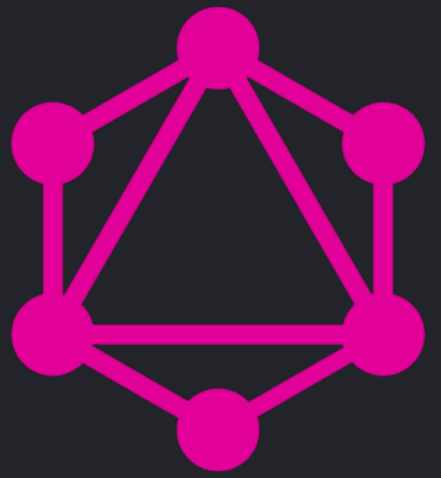
```
type Film {
  title: String
  episode: Int
  releaseDate: String
  openingCrawl: String
-  director: String
   directedBy: Person
}

type Person {
  name: String
  directed: [Film]
  actedIn: [Film]
}
```

```
type Film {
  title: String
  episode: Int
  releaseDate: String
  openingCrawl: String
+  director: String @deprecated
   directedBy: Person
}

type Person {
  name: String
  directed: [Film]
  actedIn: [Film]

}
```

```
type Film {
  title: String
  episode: Int
  releaseDate: String



}
```

```
type Film {
  title: String
  episode: Int
  releaseDate: String
+  openingCrawl: String


}
```

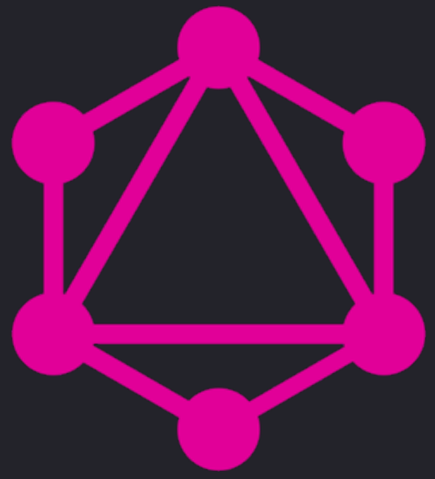
Features



Schemas: a contract between client and server

Developer UX: tools to ease learning and integration

Reasons to adopt



Two main questions were explored

How DO we build apps?

How SHOULD we build apps?

Our challenges at the time

Versioned APIs

Versioned APIs

breaking changes?

Versioned APIs

breaking changes?

updates are not delivered
without reintegration?

Versioned APIs

Adding new data

Versioned APIs

New endpoint?

Adding new data

Versioned APIs

New endpoint?

Adding new data

Add query params?

Versioned APIs

Adding new data

New endpoint?

Add query params?

Add to existing endpoint?

Versioned APIs

Adding new data

Sending too much data

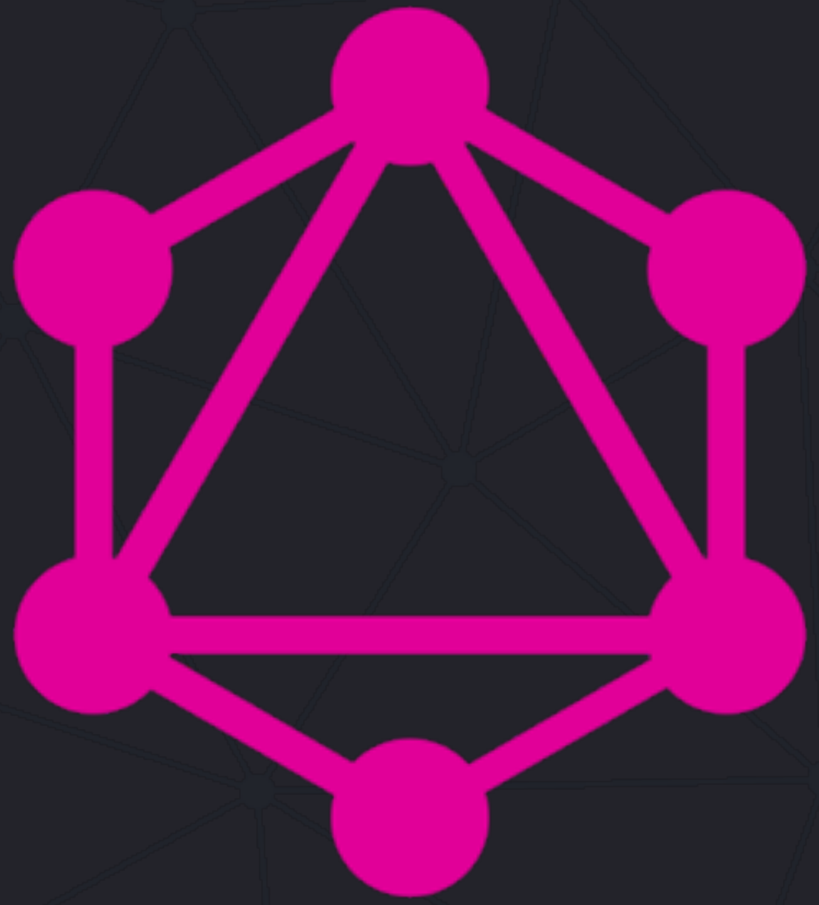
Versioned APIs

Adding new data

Sending too much data

Exploring which endpoint to use

We **think** in terms of **fields**.
Not resources, not endpoints,
not complex joins.



@shrutikapoor08

Why GraphQL?

One endpoint

No more versioned APIs.

Easy to add and deprecate fields

Small payloads

As a developer

- Developer tooling
- Collaboration with front end and back end engineers
- Work independently from a Schema contract

Slide by Mark Stuart, PayPal

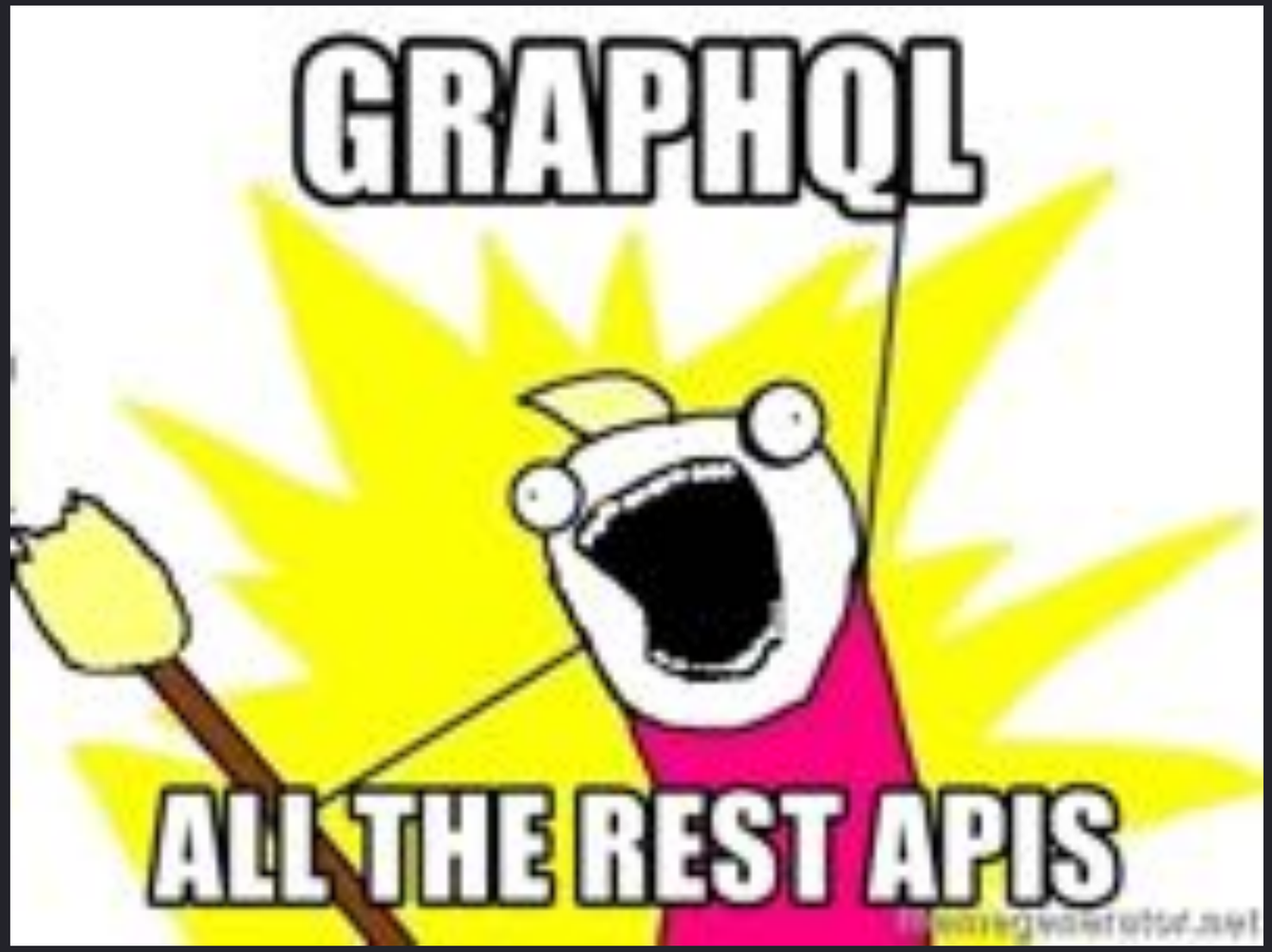
@shrutikapoor08

How to adopt GraphQL

@shrutikapoor08

GraphQL wrapper over REST API

GraphQL wrapper over REST API



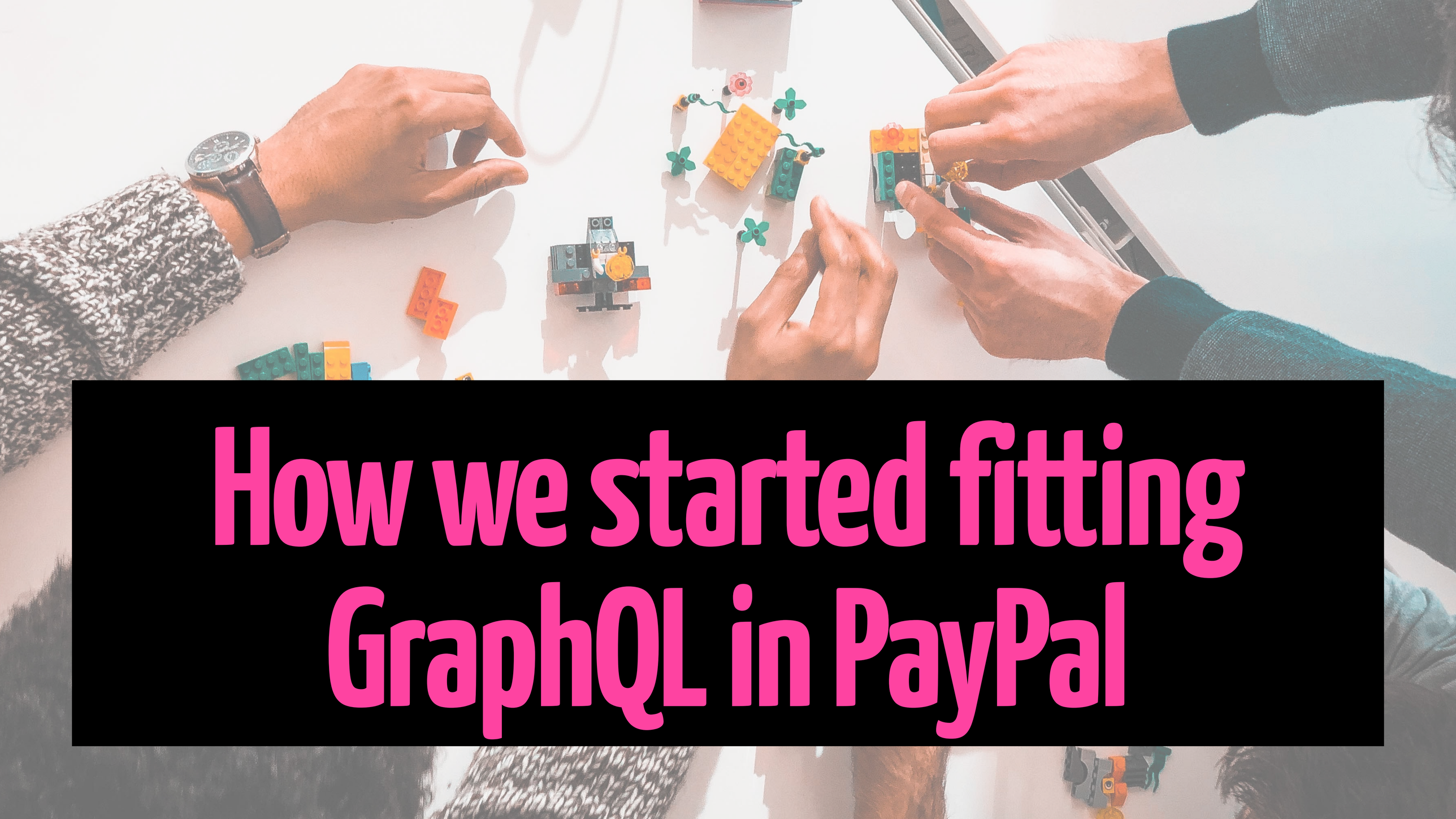
GraphQL wrapper over REST API

REST wrapper over GraphQL API

GraphQL wrapper over REST API

REST wrapper over GraphQL API

GraphQL and REST co-exist



How we started fitting GraphQL in PayPal

Possible options of plugging in GraphQL

Start experimenting with existing APIs

Build standalone APIs

Orchestrate existing APIs

PayPal adoption

Checkout app

Merchant app

PayPal adoption

Checkout app

Merchant app

1 GraphQL API : existing REST APIs + new resolvers

App is an aggregate of GraphQL + REST API

“First app”

We were creating a new internal tool

Goal - Make it fast

- Less data - no extraneous data
- Less code - no extraneous libraries

“First app”

- We found that GraphQL shines as an orchestration layer
  - So that’s what we did with our Marketplace team
    - Wrap REST api to a GraphQL API
    - Client sees and reaps the benefit of GraphQL
    - Backend developers don’t need to change the way they implement
    - Frontend and backend developers have a contract (Schema)
    - Everyone talks to everybody! Team collaboration 🎉❤️

“First app”

We had new REST API that backend developers were working on. We had to consume this new API + 5 other existing REST APIs

We created GraphQL interface to orchestrate that new REST API

Two main reasons -

only deliver data we needed,

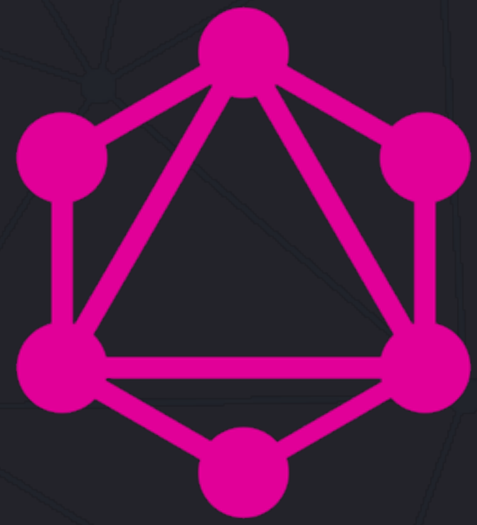
make it easier to evolve API over time

Make it easy for us to incrementally adopt GraphQL and orchestrate over existing APIs

Established a schema contract

UI drove the schema

GraphQL + REST APIs built in parallel



Current Adoption Approach

So we have a fragmented approach

The front tier applications are moving to GraphQL

Most teams have their own graphql endpoints

- But we are moving towards a federated graph throughput the company
- BrainTree has already done it.

How GraphQL helped our two main goals

- Performance
  - Clients determine how much Data they need. no. more “cleanup” of data.
  - Clients determine the shape of Data they need -> no. more “matching pattern” of data.
- Developer producitivity
  - GraphiQL - people were amazed to be able to fire off requests and see data coming back right away
  - The exploitability of docs was awesome! -> graphiql
  - Evolution of APIs: No need to worry about whether or not the field is present in a new version -> GraphiQL gives the ability to see what fields we have, and with single version we don’t need to worry about fields not being present from one version to another

Slide by Mark Stuart, PayPal

FAQs

You need a Graph type database for GraphQL

@shrutikapoor08

GraphQL = SQL?

GraphQL is only for JS developers

@shrutikapoor08

But all my services are in REST

I have sensitive data

Can I limit what APIs are exposed?

I heard GraphQL doesn't support auth

@shrutikapoor08

Lessons learnt

Pick a “first app”

Build Schema together

Orchestrate!

Don’t do a 1:1 mapping of Rest <=> GraphQL

Use built-in GraphQL features = free type safety

CAUTION

GraphQL API will be only as fast as slowest endpoint

Challenges of GraphQL

- Caching, Error Mapping
- public vs private GraphQL APIs: you need to know the schema upfront
- Making One graph when multiple teams have repos
  - Creating one graph with multiple smaller repos / graphs
  - Can be solved with schema stitching and libraries such as apollo federation, Hasura

NEXT STEPS

Assess if GraphQL is right for you

- What are the main problems of your current architecture?
- Who are the API developers and consumers?
- Do you have anything company specific - authorization, analytics, monitoring, experimentation? Where are they going to fit?
- Are you creating new APIs or wrapping existing ones?
- How much data do you think you can prevent by converting to GraphQl? Don't do a 1:1 mapping
- How will you enforce standards?
- Do you depend heavily on caching, error mapping and handling? These problems are not developed much in GraphQL
- How do you measure success ? performance? Dev productivity?

Slide by Mark Stuart, PayPal

@shrutikapoor08

Preparing your enterprise

Do a similar exercise with your company. Before jumping on the GraphQL hype train 🚂, spend time understanding your past, how you evolved, assess your strengths and weaknesses and how GraphQL might be able to help.

Set a standards team: A team to help with schema stitching, resolving schema conflicts,

Explore what open source libraries you can use. Lots of libraries to use and no need to reinvent the wheel

Determine how you are going to scale - adoption, knowledge and API

Slide by Mark Stuart, PayPal

@shrutikapoor08

How to sell to your team

Collaboration between teams

Focus on Tooling

https://graphql.org/code/#generic-tools

Dev productivity

Incremental adoption

No versions

Similar to JSON ->
Easy to understand

Get architects and domain experts onboard

To wrap up

GraphQL is here to stay

No API is perfect

Get involved in the community

GraphQL: A success story for PayPal Checkout

Mark Stuart Following
Oct 16, 2018 · 7 min read

From graphql.org

At PayPal, we recently introduced GraphQL to our technology stack.

If you haven't heard of GraphQL, it's a wildly popular alternative to REST APIs that is currently taking the developer world by storm!

At PayPal, GraphQL has been a complete game

Success stories of companies

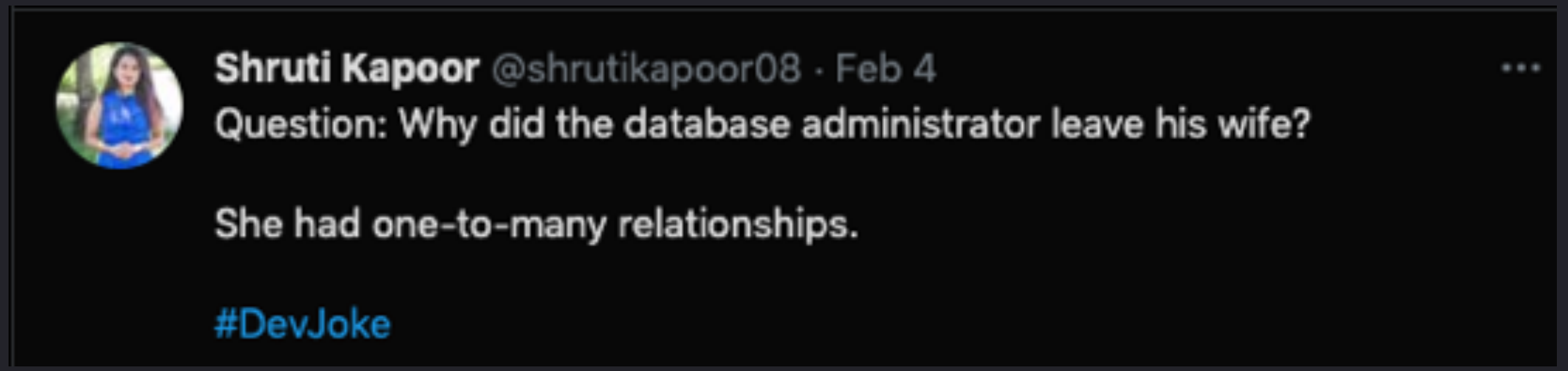
- https://medium.com/expedia-group-tech/graphql-component-architecture-principles-homeaway-ede8a58d6fde
- https://medium.com/paypal-engineering/scaling-graphql-at-paypal-b5b5ac098810
- Marc Andre's book: https://book.productionreadygraphql.com/

@shrutikapoor08

twitch.tv/shrutikapoor

JS tinyletter.com/shrutikapoor

Slides at https://github.com/shrutikapoor08/talks


Shruti Kapoor @shrutikapoor08 · Feb 4
Question: Why did the database administrator leave his wife?
She had one-to-many relationships.
#DevJoke

Thank You

@shrutikapoor08