etting started with GraphO

In an enterprise

Shruti Kapoor



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

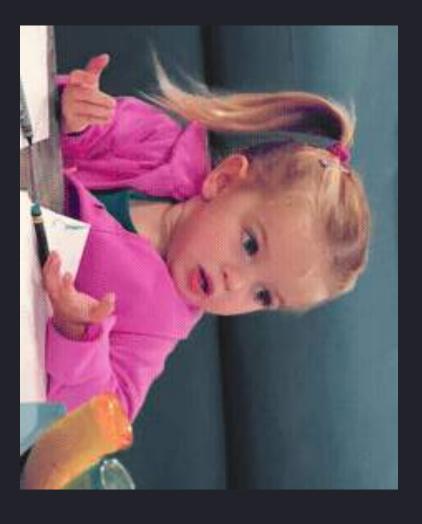
@shrutikapoor08







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

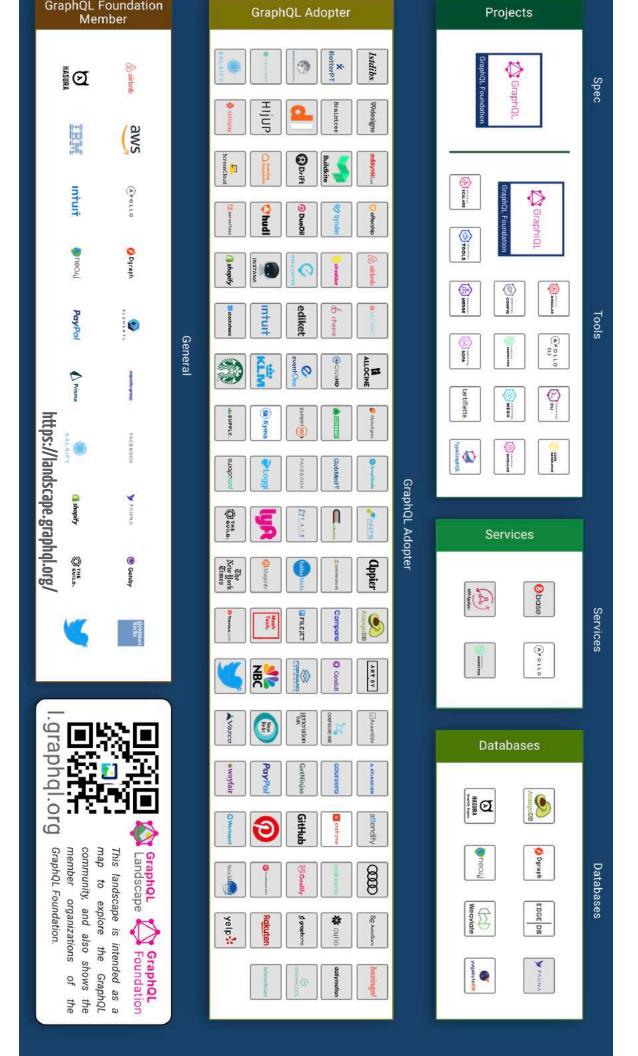


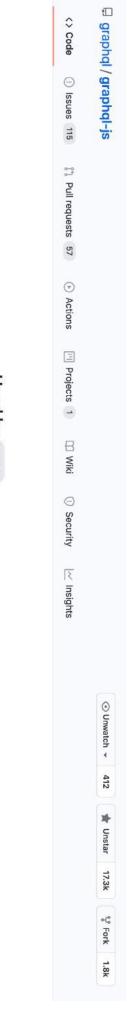
State of GraphQL in 2018

Slowly picking up in outside community GraphQL was used by 1 other team. No centralized tooling at PayPal



A lot has changed since then ■ @shrutikapoorO®







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

+ 154 contributors

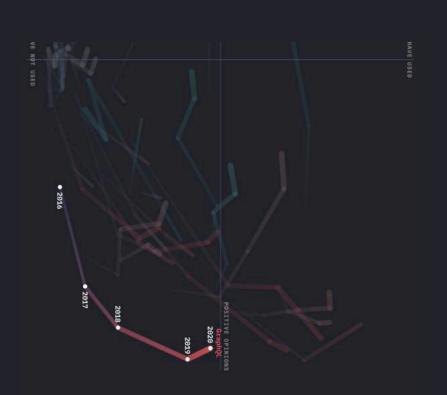


Highest Interest

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/

y @shrutikapoor08



Supported by infra teams nternal tools

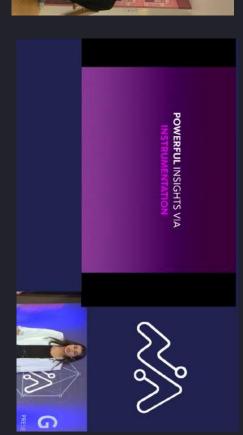


Member of GraphQL Foundation Active community member





√@shrutikapoor08



Spol 104des

Tools

Express-graphql

Apollo Prisma

Hasura

DGraph



```
8 6 91
                                                                                                                                                    765432
                                                                                                                                                                                                                         GraphiQL
                                                  WNH
                                                                                                                                                                                         1 v query TodoAppQuery($n: Int) {
2 v globalTodoList {
                                                                                      QUERY VARIABLES
                                                          "n": 2
                                                                                                                                                                                  items(first:$n) {
                                                                                                                                                                          edges {
                                                                                                                                                                  node {
                                                                                                                                                            text
                                                                                                                                                   complete
                                                                                                                                                                                "data": {
    "globalTodoList": {
                                                                                                                                                                  "items": {
    "edges": [
                                                                                                      "node": {
                                                                                       "complete": false
GraphiQL
```

```
"node": {
    "text": "Release GraphiQL",
    "complete": true
"text": "Attend @Scale 2015",
```

Try Now

https://graphql.org/swapi-graphql

```
1

2 * query {

3 * allFilms {

4 films {

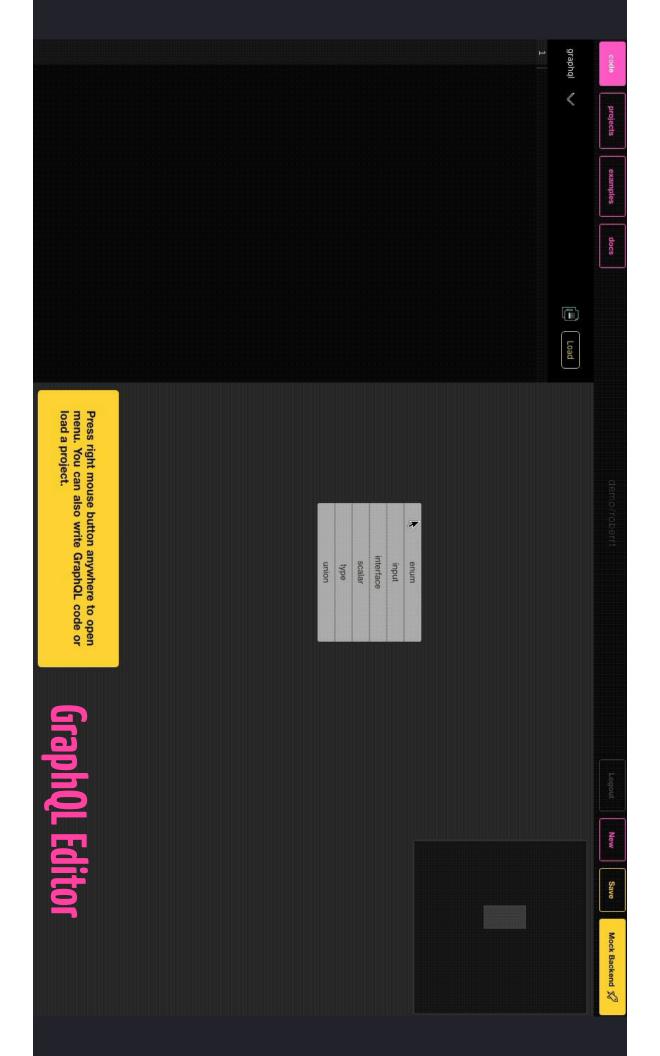
5 title

6 }

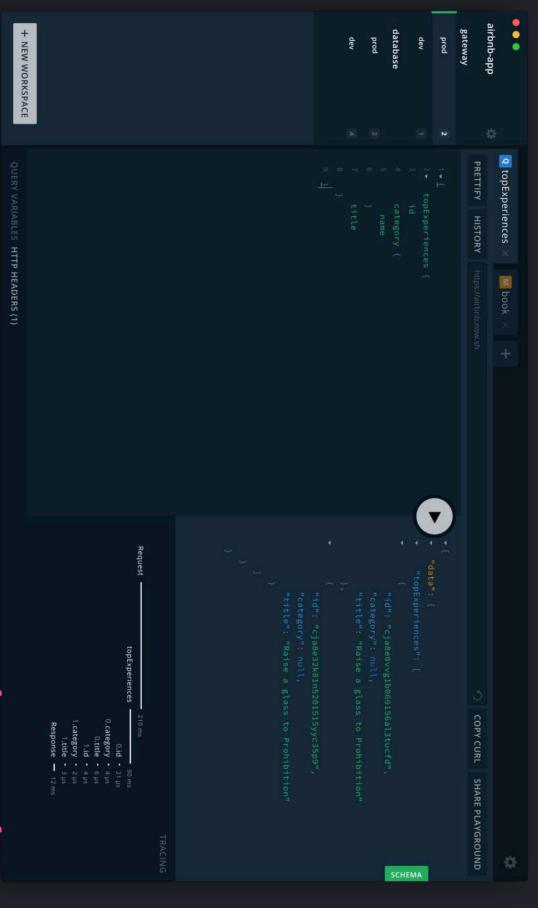
7 }
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        GraphiQL ► Prettify Merge Copy History
                                                                                                                                                                                                                                                                                                                                                                                                        "data": {
    "allFilms": {
        "films": [
                                                                                                                                                                                                                                                                                                                                                                "title": "A New Hope"
                                                                                                                                           "title": "Revenge of the Sith"
                                                                                                                                                                                                                                        "title": "The Phantom Menace"
                                                                                                                                                                                                                                                                                                                                "title": "The Empire Strikes Back"
                                                                                                                                                                                           "title": "Attack of the Clones"
                                                                                                                                                                                                                                                                                    "title": "Return of the Jedi"
                                                                                                                                              allPeople(
after: String
first: Int
before: String
last: Int
): PeopleConnection
                      allPlanets(
after: String
first: Int
before: String
last: Int
): PlanetsConnection
                                                                                                                                                                                                                                                                      allFilms(
after: String
first: Int
before: String
last: Int
): FilmsConnection

⟨ Schema
planet(id: ID, planetID: ID): Planet
                                                                                                                                                                                                                                                film(id: ID, filmID: ID): Film
                                                                                                                                                                                                                                                                                                                                                                                                                           No Description
                                                                                                                                                                                                                                                                                                                                                                                                                                                            Q Search Root...
                                                                                                                        person(id: ID, personID: ID): Person
```





Playground







Our challenges at the time **y** @shrutikapoor08

Versioned APIs

breaking changes?

updates are not delivered without reintegration?



Versioned APIs

New endpoint?

Add query params?

Add to existing endpoint?

y @shrutikapoor08



Versioned APIs

Adding new data

Sending too much data

Exploring which endpoint to use



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



Why GraphQL?

One endpoint

No more versioned APIs.

Easy to add and deprecate fields

Small payloads



As a developer

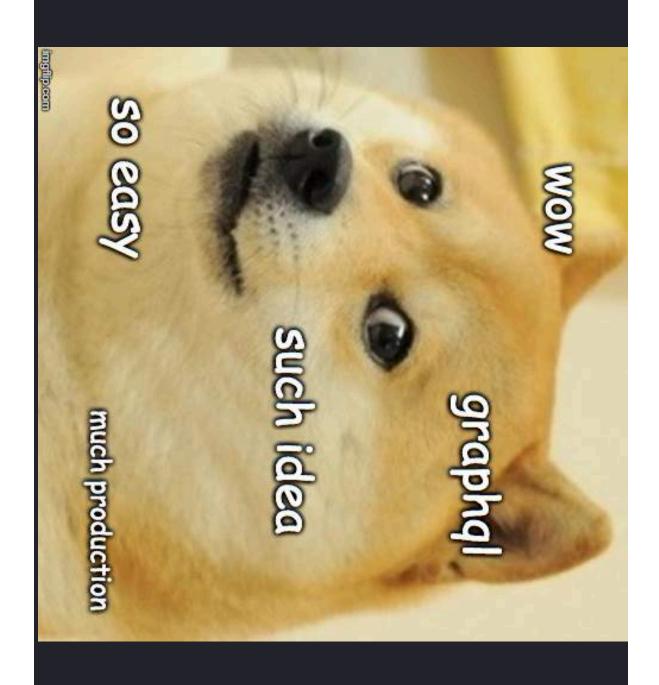
Developer tooling

back end engineers Collaboration with front end and

Work independently from a Schema contract

Slide by Mark Stuart, PayPal

У@shrutikapoor08



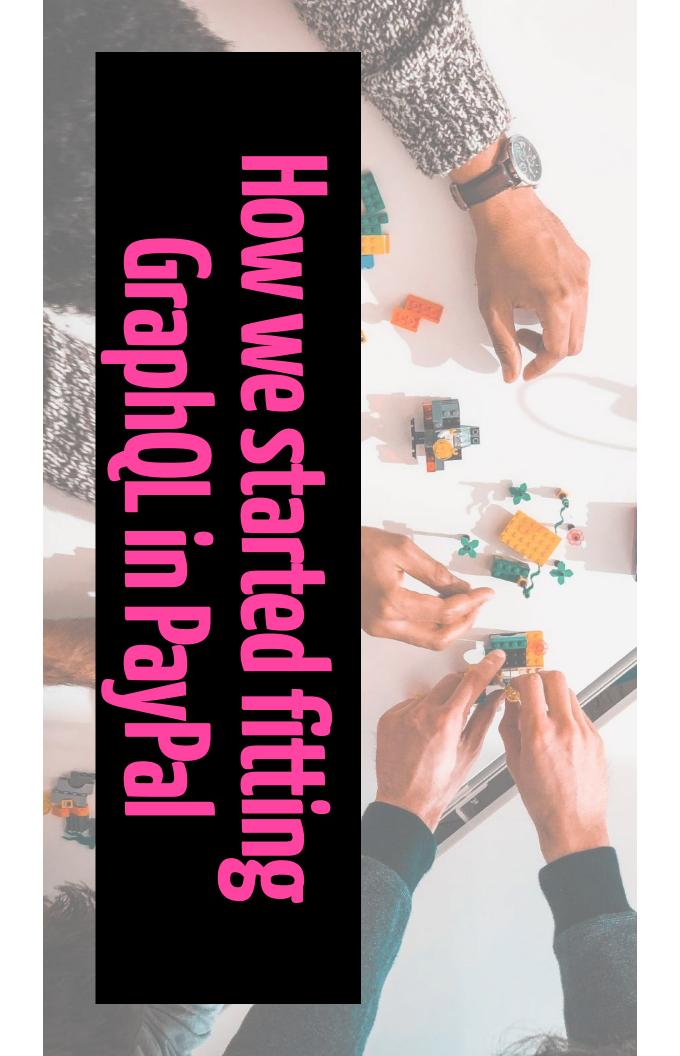


GraphQL wrapper over REST API

REST wrapper over GraphQL API

GraphQL and REST co-exist





Possible options of plugging in GraphQL

Start experimenting with existing APIs

Build standalone APIs

Orchestrate existing APIs



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

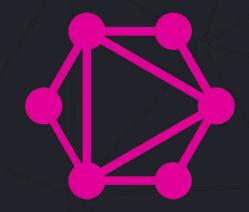
Make it easy for us to incrementally adopt GraphQL and orchestrate over existing APIs



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

company But we are moving towards a federated graph throughput the

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
- about fields not being present from one version to another GraphiQL gives the ability to see what fields we have, and with single version we don't need to worry Evolution of APIs: No need to worry about whether or not the field is present in a new version ->

lide by Mark Stuart, PayPal

FAQS



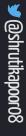












Lessons learnt



Pick a "first app"

Build Schema together

Orchestrate!

Don't do a 1:1 mapping of Rest <=> GraphO

Use built-in GraphOL teatures = tree type safety



raphOL API will be only as fast as slowest endpoin

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 tederation, 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 GraphUL
- How do you measure success? performance? Dev productivity?



Preparing your enterprise

spend time understanding your past, how you evolved, assess your strengths and Do a similar exercise with your company. Before jumping on the GraphQL hype train 🚑, 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



How to sell to your team



Collaboration between teams

https://graphal.org/code/ Focus on Tooling #generic-tools

Dev productivity

Incremental adoption

No versions

Similar to JSON -> Easy to understand

bet architects and domain experts nleode

dn desm of



GraphOL is here to stay

No API is perfect

Get involved in the community

PayPal Checkout GraphQL: A success story for

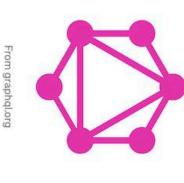












At PayPal, we recently introduced GraphQL to our technology stack.

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

At PayPal, GraphQL has been a complete game

Success stories of companies

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







shrutikapoor08



witch.tv/shrutikapoor

Js tinyletter.com/shrutikapoor

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





Question: Why did the database administrator leave his wife? Shruti Kapoor @shrutikapoor 08 · Feb 4

She had one-to-many relationships.

#DevJoke

y @shrutikapoor08

Thank You







