GraphQL & Apollo

Joanna Zakrzewska

GraphQL?

QUERY LANGUAGE

FOR DYNAMIC API

by Facebook

request

```
query {
  user(id: 1) {
    firstName
    lastName
    posts {
      id
      votes
    }
}}
```

response

```
{
  user: {
    firstName: "Tom",
    lastName: "Smith"
    posts: [{
      id: 1,
      votes: 2
    }]
}}
```

request

```
mutation {
  upvotePost(id: 1) {
    votes
    user {
      firstName
      lastName
    }
  }
}
```

response

```
{
  upvotePost: {
    votes: 3,
    user: {
      firstName: "Tom",
      lastName: "Smith"
    }
  }
}
```

👍

all you need

SINGLE ENDPOINT / SINGLE CALL

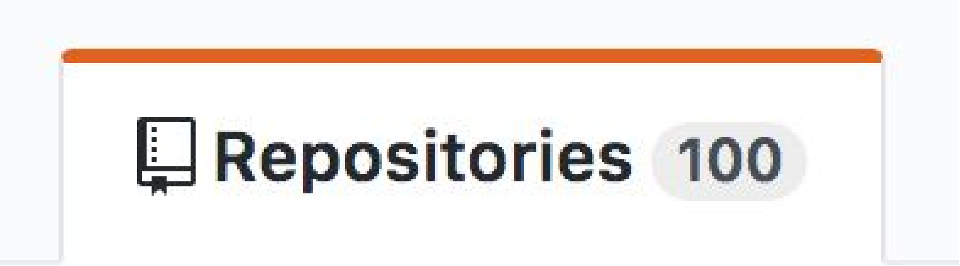
and just that

in a generic way

- 👍 simple & declarative
- 👍 technology agnostic
  - BACKEND LANGUAGE
  - TRANSFER PROTOCOL
- 👍 great for proxy
  - WRAPS BAD APIS
  - AGGREGATES APIS
  - COVERS API CHANGES
- 👍 incremental adoption

- 👍 design first
- 👍 typed
- 👍 auto-documenting
- 👍 less meetings

- 👎 HTTP caching
- 👎 not fully mature yet



Repositories 100

Apollo?

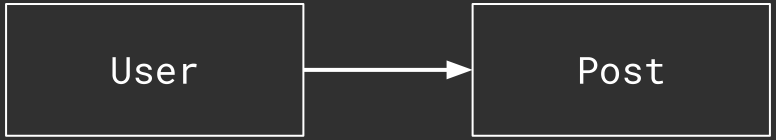
GRAPHQL TOOLS

ALL OF THEM

BY METEOR DEVS

schema

Silhoue/graphql-demo



User

Post

```
const UserType = new GraphQLObjectType({
  name: "User",
  fields: {
    id: { type: GraphQLInt },
    firstName: { type: GraphQLString },
    posts: {
      type: new GraphQLList(PostType),
      resolve: (user, args, context) => getPosts(...)
}}})
```

Backend tools

Facebook stack

- `graphql` + `express-graphql`
- `dataloader`

Apollo stack

- `graphql` + `apollo-server-{express}`
- `graphql-tools`

client

[apollo-client/README.md]()

```
const userQuery = gql`{
  User(id: 1) {
    firstName
}}`

const UserComponent = (props) => {
  const { User, loading, error} = props.data
  if (loading) { return <span>loading...</span> }
  if (error) { return <span>error!</span> }
  return <span>{User.firstname}</span>
}

export default graphql(userQuery)(UserComponent)
```

Frontend tools

Facebook stack

- **`relay`**
- `graphql-relay-js` (backend)

Apollo stack

- **`apollo-client`*** + e.g. **`react-apollo`**
- `apollo-link`

*apollo-client-preset + graphql-tag + graphql

What about Redux?

`apollo-link-state 0.3.1`

Client Schema Extensions ⏳

React State Management In a GraphQL Era
by Kristijan Ristovski

Apollo Client 2.0: More than just GraphQL
by Peggy Rayzis

More tools

GraphiQL - clientless

Graphcool - serverless

Launchpad - playground

Apollo Client Developer Tools - Chrome extension

More resources

facebook.github.io/graphql - spec

graphql.org (by Facebook)

graphql.com & apollographql.com (by Apollo)

github.com/chentsulin/awesome-graphql - even more links

graphql-europe.org (Berlin, June 15th, 2018)