GraphQL

A Gentle Intro

What is it?

It’s a Query Language

Your model is represented as a series of schemas on the Server

Those schemas are introspectable via the API

Has validation and lots of other useful things

Background

Originally a Facebook project to build an API for mobile apps

Spec: http://facebook.github.io/graphql/October2016/

Code: https://github.com/graphql

Docs: http://graphql.org

Licence: https://medium.com/@leeb/relicensing-the-graphql-specification-e7d07a52301b

It’s agnostic about...

- Transport Protocol
- Serialisation
- Query Format
- Schema Format
- Results Format

“Reference Implementations” make choices that become defacto standards

What is it good for?

You want to publish an API but allow the clients flexibility with:

1. How they traverse your object model
2. Specifying exactly what data they get back

Allows them to formulate very custom queries, reducing number of server calls

What’s the Graph part about?

ObjectA -> ConnectionObject -> ObjectB

The graph refers to graph structures defined in the schema, where nodes define objects and edges define relationships between objects

```
PullRequest {
        Commits {
                Edges {
                        Nodes {
                                Comment
                                }
                        }
                }
        }
```

See this article

Schemas/Types

Looks a bit like an IDL such as Protocol Buffers

Strongly typed. Interfaces. Objects. Enums etc http://graphql.org/learn/schema/

Scalar types http://graphql.org/learn/schema/#scalar-types

All types are Nullable

Schema Definition Language
https://wehavefaces.net/graphql-shorthand-notation-cheatsheet-17cd715861b6

Example from Github: https://developer.github.com/v4/reference/object/repository/

Schemas are.. Introspectable

GraphiQL Explorer for public github https://developer.github.com/v4/explorer/

Chrome Add on here which we can use for our own github (still not sure why we can’t use the GraphiQL explorer

Facilitate:

- Documentation
- Completion
- Validation

Queries

Normally JSON like syntax

Fragments http://graphql.org/learn/queries/#fragments

Variables http://graphql.org/learn/queries/#variables

Directives http://graphql.org/learn/queries/#directives

Pagination a big emphasis http://graphql.org/learn/pagination/

Github

curl -H "Authorization: bearer 7c015139d53e98ae7a3dacc28dd82df646c17672" -H "Accept: application/vnd.github.v4.idl" https://api.github.com/graphql

You’ll need a token from github: https://developer.github.com/v4/guides/forming-calls/#authenticating-with-graphql

Other Operations

Mutation

Subscribe

Server-Side

Single entry-point or “endpoint”.

I don’t know much.. But see this talk from Facebook

Different approaches to caching wrt REST

Can be added to “legacy” apps using other kinds of services

Resolvers: http://graphql.org/learn/execution/#root-fields-resolvers

Hello world example using Graphene

Code & Libraries

http://graphql.org/code/#server-libraries

http://graphql.org/code/#graphql-clients

Python

https://github.com/graphql-python/graphene Server

https://github.com/graphql-python/gql Client

Javascript

http://graphql.org/graphql-js/

UI Libraries built on top of GraphQL

Relay https://facebook.github.io/relay/

Apollo http://dev.apollodata.com/react/index.html

Learn more

https://www.howtographql.com/