GraphQL:Thinking in Resolvers

Sébastien Lavoie-Courchesne

February 23rd 2022

opening

Sébastien Lavoie-Courchesne

- Architect for the Catalog Group at AppDirect
- Member of the GraphQL Working Group at AppDirect
- Developed most of the initial infrastructure we use

Contents

Technologies used in the examples

- GraphQL schemas
- ApolloServer stack:
  - NodeJS
  - Typescript
  - ApolloServer

What is GraphQL?

From the GraphQL Specification[1]:

> *GraphQL is a query language designed to build client applications by providing an intuitive and flexible syntax and system for describing their data requirements and interactions.*

But What is GraphQL?

A specification defining a query language for communication between different services over a network and how to implement the server and client sides of the communication

Why GraphQL?

Only request and receive the fields you want:

- Smaller payloads
- Descriptive language
- Documented

Only request and receive the fields you want:

- Smaller payloads
- Descriptive language
- Documented

Often compared to REST and gRPC

Only request and receive the fields you want:

- Smaller payloads
- Descriptive language
- Documented

Mostly used over HTTP, but not always

How does it work?

Server defines the schema it serves using the GraphQL language
Clients can read the schema and send queries to the server using the GraphQL language, optionally passing in variables

Server defines the schema it serves using the GraphQL language
Clients can read the schema and send queries to the server using the GraphQL language, optionally passing in variables

typically one query per request, multiple queries in a single request are supported

Operations

Query	a read-only fetch, analogous to REST GET
Mutation	a write followed by a fetch, analogous to REST POST/PATCH/PUT/DELETE
Subscription	a long-lived request that fetches data in response to source events

Query	a read-only fetch, analogous to REST GET
Mutation	a write followed by a fetch, analogous to REST POST/PATCH/PUT/DELETE
Subscription	a long-lived request that fetches data in response to source events

Subscription is less often used and less documented, multiple ways to serve it (WebSocket, Kafka, RabbitMQ, etc)

HTTP Communication

- Most common use of GraphQL
- HTTP GET or POST to a specific URL
- Traditional HTTP headers/cookies
- GET `query` parameter contains the query as a string
- POST Body includes

`query`	Query as a string
`variables`	Query variable values as a JSON object, optional
`operationName`	Name for the operation, optional

Types and fields

type Post
"The blog post's identifier"
id: ID!
"Title of the blog post"
title: String!
"Contents of the blog post"
contents(format: BlogContentsFormat): String!
"Possible Blog Contents Formats" enum BlogContentsFormat "Simple text" TEXT "HTML formatted" HTML

Conclusion

Links

- blog
- this presentation

References

[1] *GraphQL Specification, October 2021 Edition*. GraphQL contributors. Oct. 2021. URL: https://spec.graphql.org/October2021/.