Exploring GraphQL with your API
Brian Douglas
@bdougieYO
How to do I use GraphQL today?
#goals
JAM STACK
Monolithic Architecture
The JAM Architecture
The JAM Architecture

Client

API
I didn’t own the Netlify API
I needed something simple
Rails + Go + Mongo
=
Netlify API
ruby?
graphql-ruby = SQL
graphql-ruby
!=

mongo
node?
Rails + Go + Mongo + node?

express-graphql
apollo server
graphql-anywhere
graphql-code-generator
wrappers?
Wrapping a REST API in GraphQL

5/5/2016 by Steven Luscher

Time and time again I hear the same aspiration from front-end web and mobile developers: they're eager to reap the developer efficiency gains offered by new technologies like Relay and GraphQL, but they have years of momentum behind their existing REST API. Without data that clearly demonstrates the benefits of switching, they find it hard to justify an additional investment in GraphQL infrastructure.

In this post I will outline a rapid, low-investment method that you can use to stand up a GraphQL endpoint atop an existing REST API, using JavaScript alone. No backend developers will be harmed during the making of this blog post.

A client-side REST wrapper

We're going to create a GraphQL schema — a type system that describes your universe of data — that wraps calls to your existing REST API. This schema will receive and resolve GraphQL queries all on the client side. This architecture features some inherent performance flaws, but is fast to implement and requires no server changes.

Imagine a REST API that exposes a `/people/` endpoint through which you can browse `Person` models and their associated `friends`.
I took a chance on wrappers
REST API

Apollo LaunchPad

/graphql
https://launchpad.graphql.com/m1wl39p39
const resolvers = {
  Query: {
    deploys: (root, args, context) => {
      return fetch(`https://api.netlify.com/api/v1/sites/${args.site_id}`)
        .then(res => res.json()).catch(err => console.log(err));
    },
  },
};

export const schema = makeExecutableSchema({
  typeDefs,
  resolvers,
});
const resolvers = {
  Query: {
    deploys: (root, args, context) => {
      return fetch(`https://api.netlify.com/api/v1/sites/${args.site_id}`)
        .then(res => res.json()).catch(err => console.log(err));
    },
  }
};

export const schema = makeExecutableSchema({
  typeDefs,
  resolvers,
});
my wrap game is strong
not winning awards
but wrap sells
Unlocked benefits of GraphQL
Schema stitching
Built in caching w/Apollo
REST API one

REST API two

Schema Stitching

@bdougieYO
this chance wrap makes sense
graphql-ruby gem!

ruby👍
supports mongo👍
Build a GraphQL server with any database

Prisma is a performant open-source GraphQL ORM-like* layer doing the heavy lifting in your GraphQL server.

* GraphiQL
Migrating to GraphQL

Migrating from REST to GraphQL

Differences in API logic

Migrating from REST to GraphQL represents a significant shift in API logic. The differences between REST as a style and GraphQL as a specification make it difficult—and often undesirable—to replace REST API calls with GraphQL API queries on a one to one basis. We’ve included specific examples of migration below.

To migrate your code from the REST API v3 to the GraphQL API v4:

- Review the GraphQL spec
- Review GitHub’s GraphQL schema
- Consider how any existing code you have currently interacts with the GitHub REST API v3
- Use Global Node IDs to reference objects between API versions

Significant advantages of GraphQL include:
Explore GraphQL w/GitHub

Ep. #11, Exploring GraphQL Through Apollo

Guests: Sashko Stubailo & Jonas Heifer

In the latest episode of JAMstack Radio, Brian invites Sashko Stubailo and Jonas Heifer from Meteor to discuss Apollo, a set of tools developed to make GraphQL easier to use.

They explain that while GraphQL is a great technology, the ecosystem of tools for the technology is still developing. They also discuss Optics, a paid service that collects server data and delivers analytics and performance information.
It takes a community to design, build, and ship great software.

Building an application that integrates with GitHub? Register for our Developer Program! The possibilities are endless, and you enjoy the kudos.

Register now
word to your mother