

# GraphQL

## FROM THE GROUND UP

demystifying the GraphQL server-side

# What is it?

## GraphQL

a Type/Schema System

a Query Language

# What is it good for?

- Can be used as an alternative to REST apis
  - Strong schema built in
  - Client controls shape of data - no overfetching
  - Strong opinions - no RESTfulness arguments

# What else?

- Can be used as a generic query language:
  - GatsbyJS - wrangles assets to generate a static site
  - Prisma.io - used as a data access layer
  - DGraph.io - used to query a graph database

**There's a dizzying array of tools...**



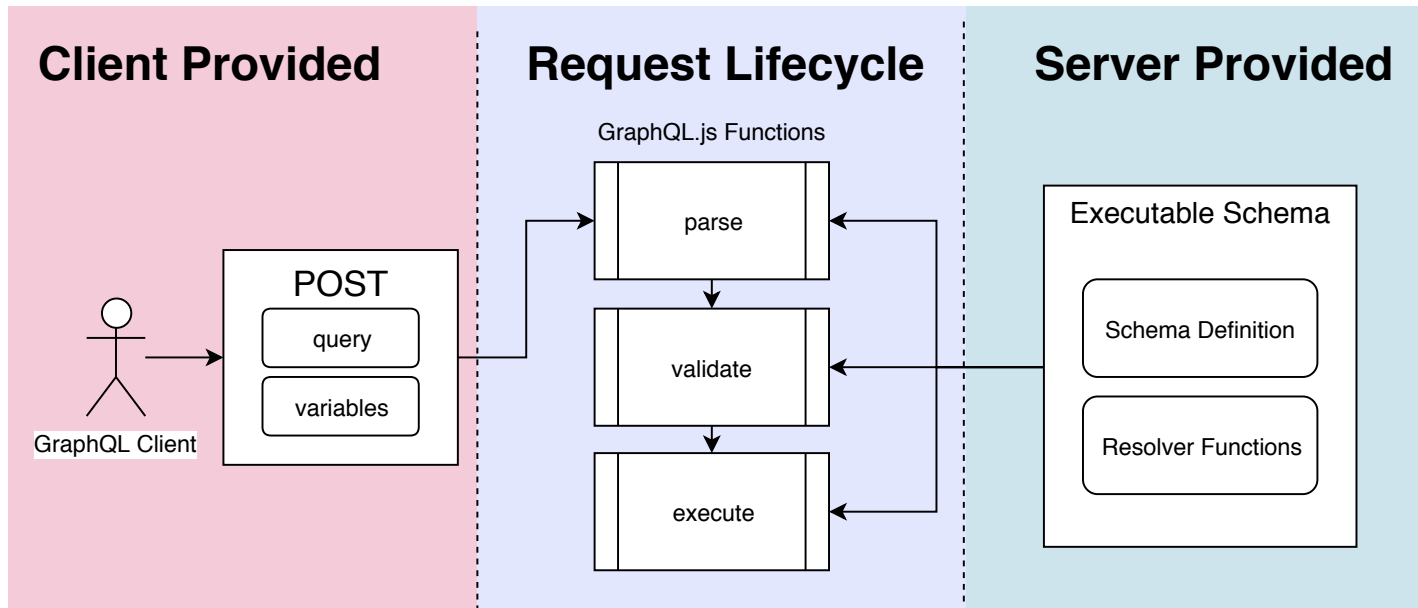
## Two main options for node servers:

- **express-graphql** - open source contributed by facebook
- **apollo-server** - community open source by the makers of apollo client

**Let's see how they work...**

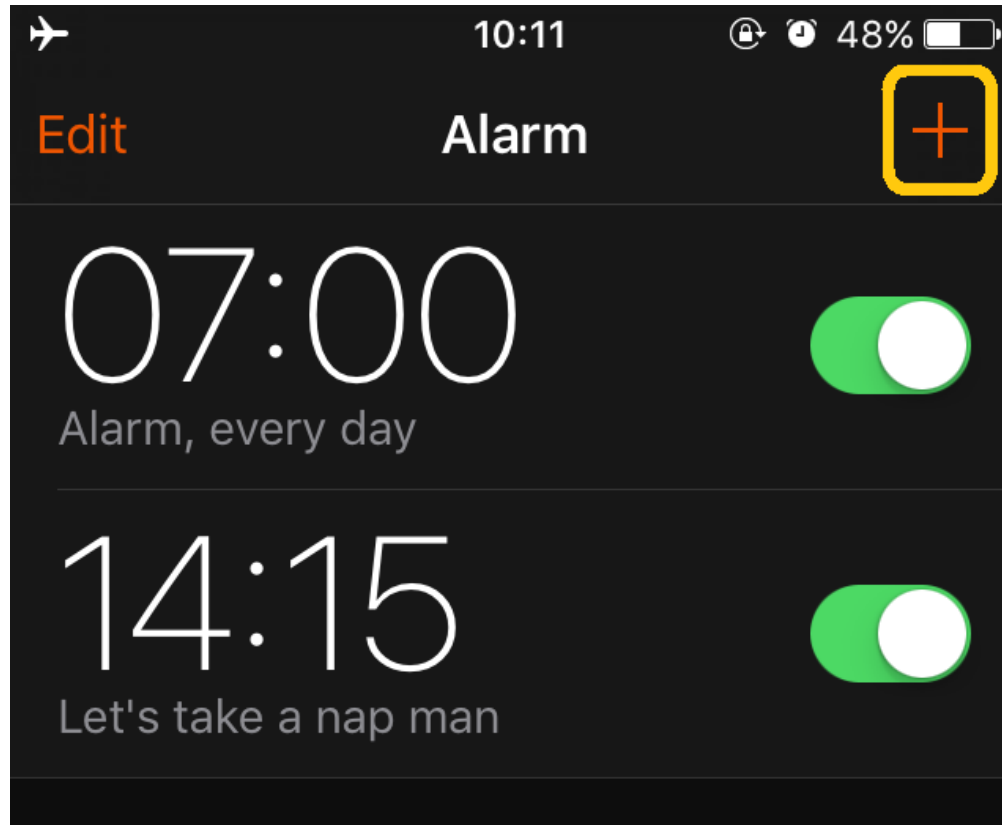


# Building a mental model





Let's build a thing.



## You'll need these...

- **graphql.js** - the language and the type system with functions to make it work (parse, validate, execute, subscribe)
- **graphql-tools** - swiss army knife for building schemas, mocking them, combining them, etc.

## and for Subscriptions you'll need...

- **subscriptions-transport-ws** - a transport server for GraphQL subscriptions
- **iterall** - a polyfill for async iterators (until you're on Node 10+)

## In the real world you'll probably want...

- **apollo-server** - http framework with implementations in your flavor of choice (express, koa, restify, etc.)
- **graphql-subscriptions** - pubsub and helpers for async iterators

**Questions?**



# We're hiring!



# socialtables

[daniel@socialtables.com](mailto:daniel@socialtables.com) | [@crunchnode](#)