

# Introduction to GraphQL

By Matthew Brimmer

# About Me

- Graduated from DigitalCrafts in 2016
- Node developer Merchant E-Solutions
- Full-stack Developer Turner Broadcasting
- React Engineer RentPath
  - [Rent.com](https://www.rent.com)
  - [ApartmentGuide.com](https://www.apartmentguide.com)

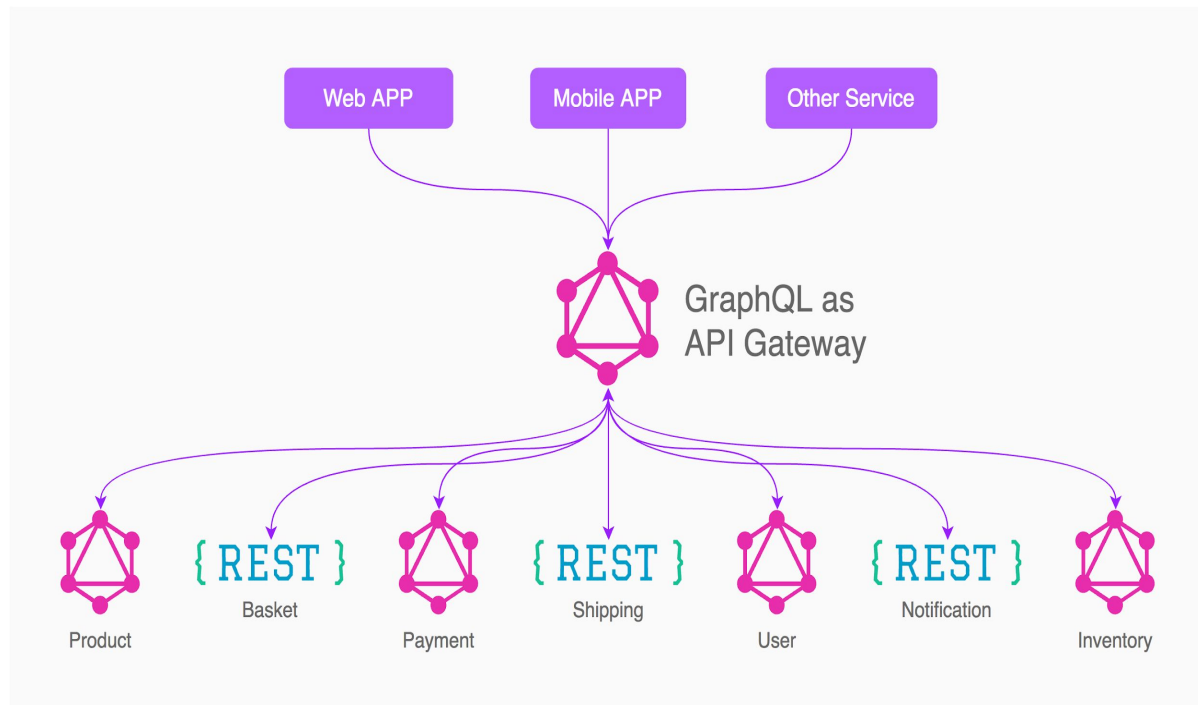


# What is GraphQL and why we love it?

- “A Query language for your API”
- Runtime for executing queries
- Request only the data needed
- Return data in the required format
- Manipulate data format
- Defines a type system for our data
- Merge data from multiple sources
- Works well with microservices



# GraphQL with microservices



# GraphQL is a spec

- [GraphQL.js](#) Javascript
- [Graphene](#) Python
- [GraphQL-Java](#) Java
- [GraphQL-Go](#) Golang
- [GraphQL-Ruby](#) Ruby
- [Awesome GraphQL](#)



# GraphQL Starter

- [GraphQL Starter](#)
- ``git clone https://github.com/easycodersio/GraphQLStarter.git``
- ``cd GraphQLStarter``
- ``yarn install``
- ``yarn run dev``



# EasyCoders API

- EasyCoders API
- Sports Search

/api/sports/search?query=falcons

```
{
  "teams": [
    {
      "name": "Atlanta Falcons",
      "city": "Atlanta",
      "state": "Georgia",
      "stadiumId": "8452e3f9-09f8-45d6-9252-652af463bd3e",
      "league": "nfl",
      "id": "19fe0471-de6d-11e9-aa38-e7cd6d4d6af2"
    }
  ]
}
```

# GraphQL Schema

- Center piece of GraphQL
- GraphQLSchema Type
- A Graph data structure that represents all types, queries and mutations
- Types are at the core of a GraphQL schema

```
~
6  const query = new GraphQLObjectType({
7    description: 'root query',
8    fields: {
9      Hello,
10     Search,
11     StadiumById,
12   },
13   name: 'RootQuery',
14 })
15
16 const schema = new GraphQLSchema({
17   query,
18 })
```



# GraphQL Core Types

- Object
- String
- Integer
- Float
- Boolean
- List



# Query

```
1 query Search($query: String! $withStadiums: Boolean) {  
2   Search(query: $query withStadiums: $withStadiums) {  
3     teams {  
4       name  
5       city  
6       state  
7       stadiumId  
8       id  
9     }  
10    stadiums {  
11      name  
12      id  
13      lat  
14      lon  
15      teams  
16    }  
17  }  
18 }
```

# Resolvers

- Resolvers allow us to resolve data to the schema
- Compute data from multiple sources
- Alter structure of data
- Rename fields



# Composers

- Composers allow the creation of schema parts
- GraphQL Compose
  - JSON
  - MongoDB
  - ElasticSearch
  - AWS Cloud API



# Mutation

- Modify data at the source
- Mutations are isolated on the schema
- Supports normal REST CRUD mutations
  - CREATE, UPDATE, DELETE



# Client

- Apollo Client manages queries and state
- [Apollo Client React](#)
- Supports other Frameworks too
  - Vue
  - Angular
  - Ember
  - IOS
  - Android



# EasyCoders

- GraphQL Udemy Course coming in a few weeks
- Free for DigitalCraft students and alumni

