

INTRODUCTION TO GRAPHQL

Platinum Sponsors



Celebration Sponsor



Notebook Sponsor



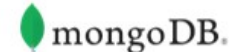
Lanyards Sponsor



Registration Sponsor



Gold Sponsors



Silver Sponsors

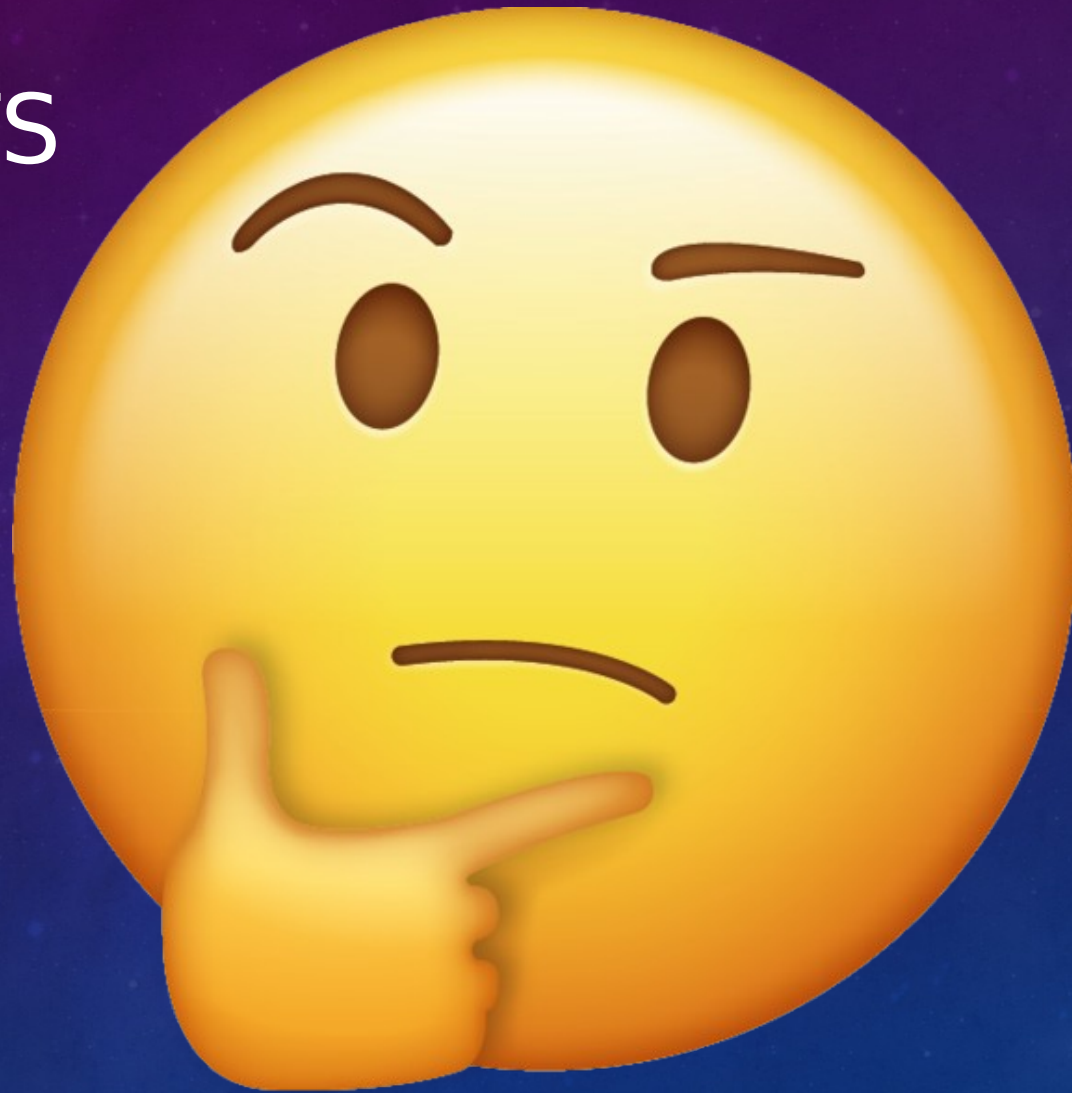


BRYAN NEHL @K0EMT

GRAPHQL CORE CONCEPTS

- API technology
 - specification
- Today backend JavaScript w/node.js
- NOT strictly JavaScript

CORE CONCEPTS





CORE CONCEPTS



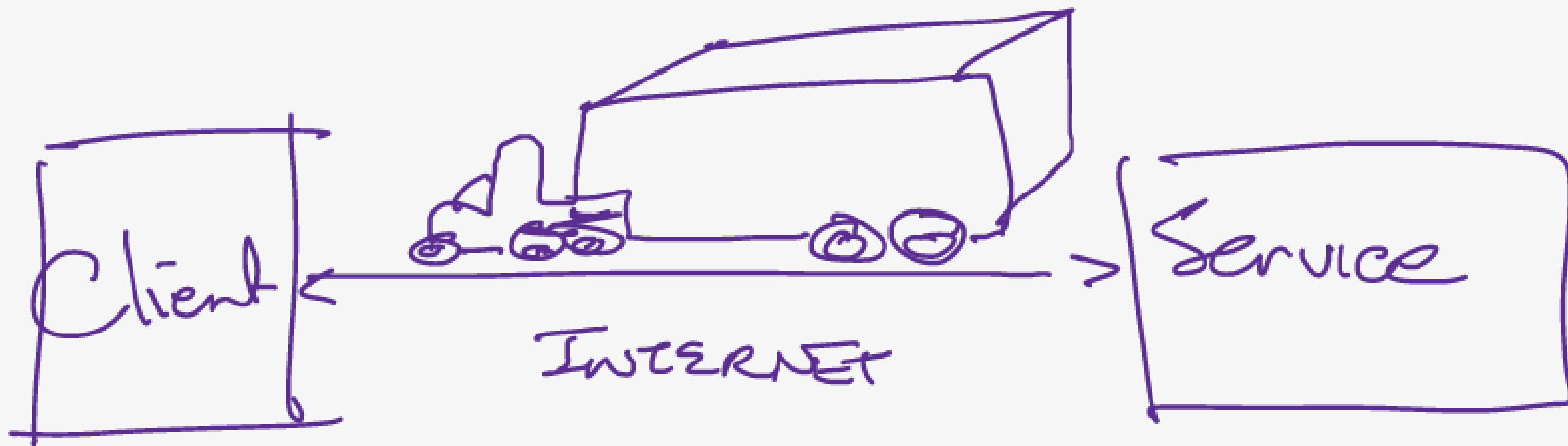


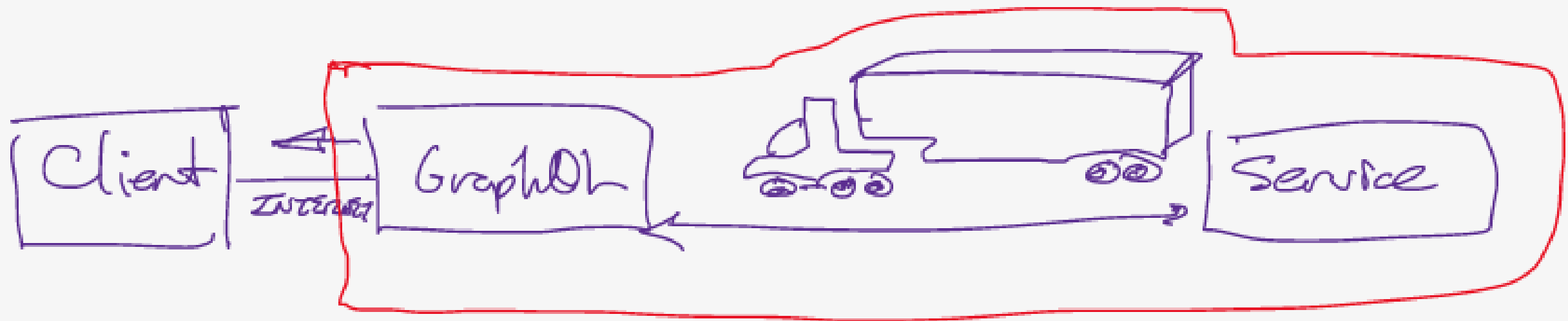
CORE CONCEPTS – RELATED TO SQL

```
SELECT * FROM table;
```

```
SELECT field1, field15 FROM table;
```

```
SELECT a.field1, a.field15, b.field2, b.field5 FROM table a inner join table  
b on a.bld = b.ld;
```



CORE CONCEPTS – DIFFERENT FROM SQL



CORE CONCEPTS

- **Schema**
 - SDL/IDL
 - Introspection & GraphQL Voyager

CORE CONCEPTS

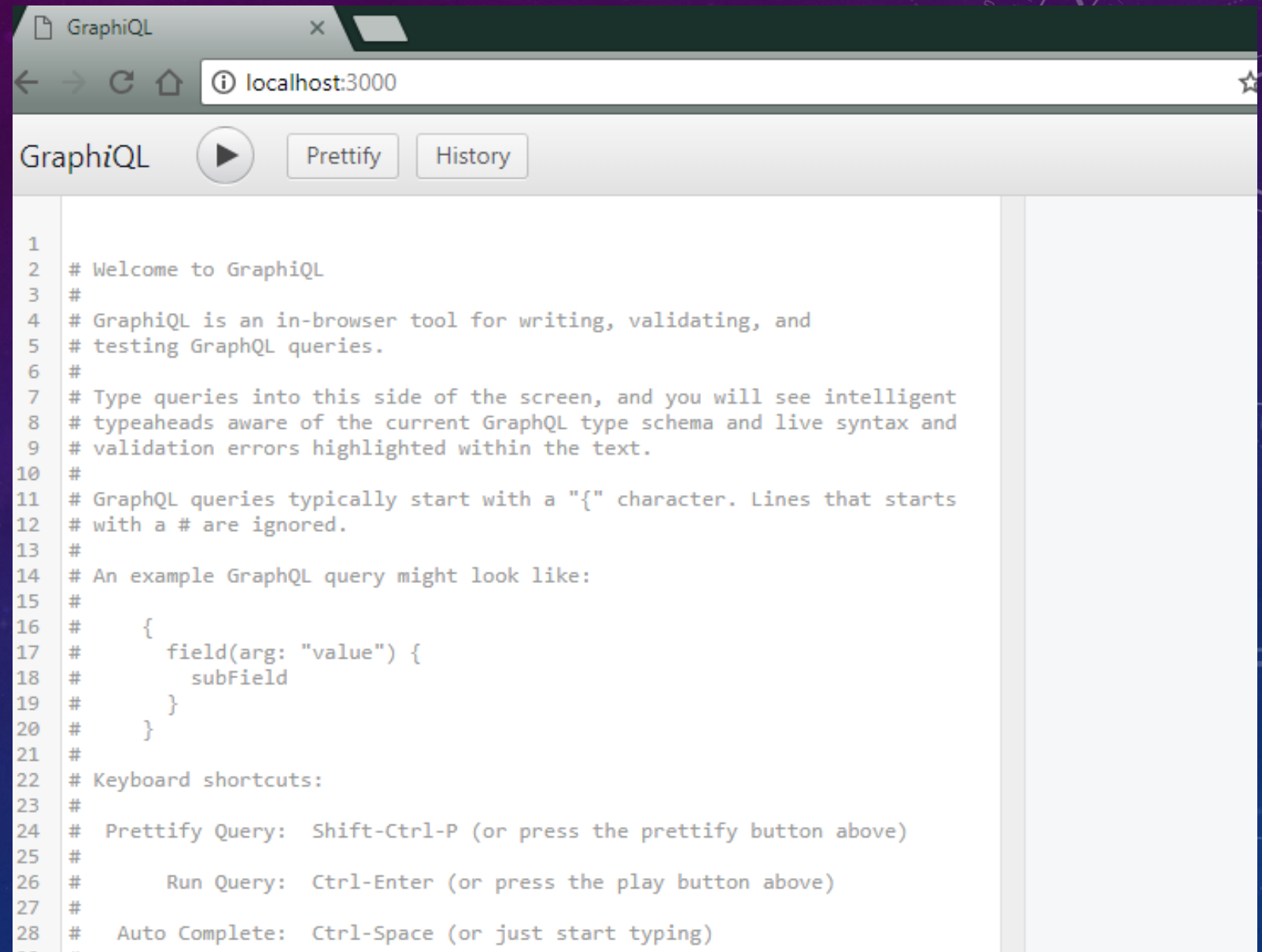
- **Query** for the **Data** you want
- Contracts

INTERACTIVE CLIENTS

- **Curl** `curl --request GET`
- **GraphiQL** `--url http://localhost:3000/graphql/`
- **Insomnia** `--header 'content-type: application/json'`
- **Insomnia** `--data '{"query":"{hello}"}'`

INTERACTIVE CLIENTS

- Curl
- **GraphiQL**
- Insomnia

A screenshot of the GraphiQL web interface. The browser window has a single tab titled 'GraphiQL' and the address bar shows 'localhost:3000'. The interface includes a header with the 'GraphiQL' logo, a play button icon, and 'Prettify' and 'History' buttons. The main area is a text editor with line numbers on the left. It contains a welcome message and an example GraphQL query. The query is:

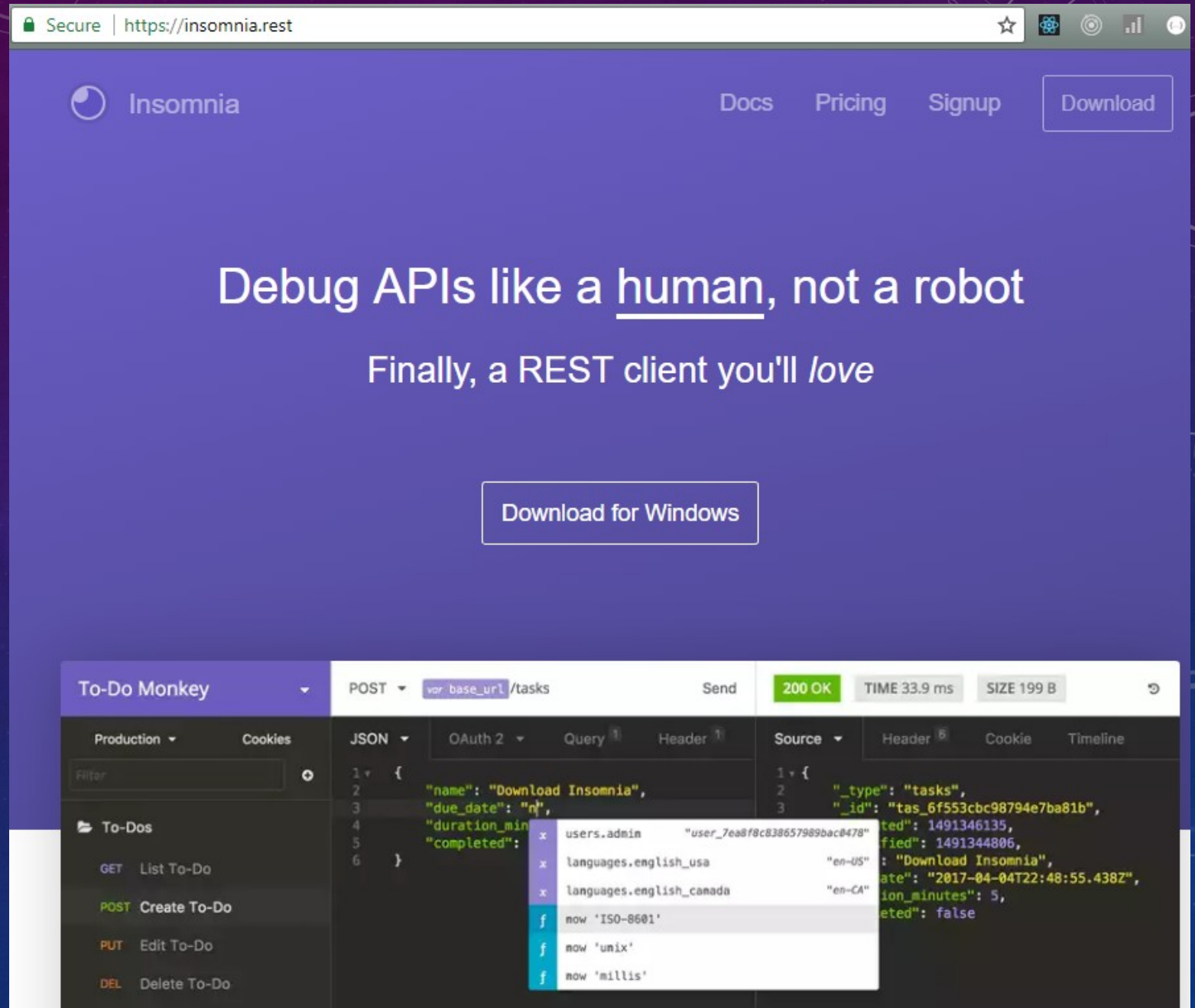
```
{
  field(arg: "value") {
    subField
  }
}
```

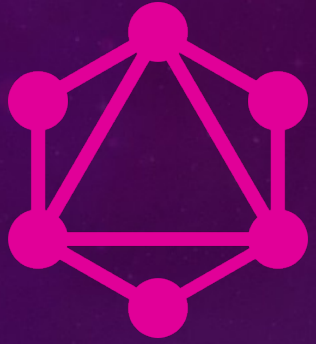
 Below the query, there are keyboard shortcuts: 'Prettify Query: Shift-Ctrl-P (or press the prettify button above)', 'Run Query: Ctrl-Enter (or press the play button above)', and 'Auto Complete: Ctrl-Space (or just start typing)'.

```
1
2 # Welcome to GraphiQL
3 #
4 # GraphiQL is an in-browser tool for writing, validating, and
5 # testing GraphQL queries.
6 #
7 # Type queries into this side of the screen, and you will see intelligent
8 # typeaheads aware of the current GraphQL type schema and live syntax and
9 # validation errors highlighted within the text.
10 #
11 # GraphQL queries typically start with a "{" character. Lines that starts
12 # with a # are ignored.
13 #
14 # An example GraphQL query might look like:
15 #
16 #   {
17 #     field(arg: "value") {
18 #       subField
19 #     }
20 #   }
21 #
22 # Keyboard shortcuts:
23 #
24 #   Prettify Query:  Shift-Ctrl-P (or press the prettify button above)
25 #
26 #       Run Query:  Ctrl-Enter (or press the play button above)
27 #
28 #   Auto Complete:  Ctrl-Space (or just start typing)
```


INTERACTIVE CLIENTS

- Curl
- GraphiQL
- **Insomnia**





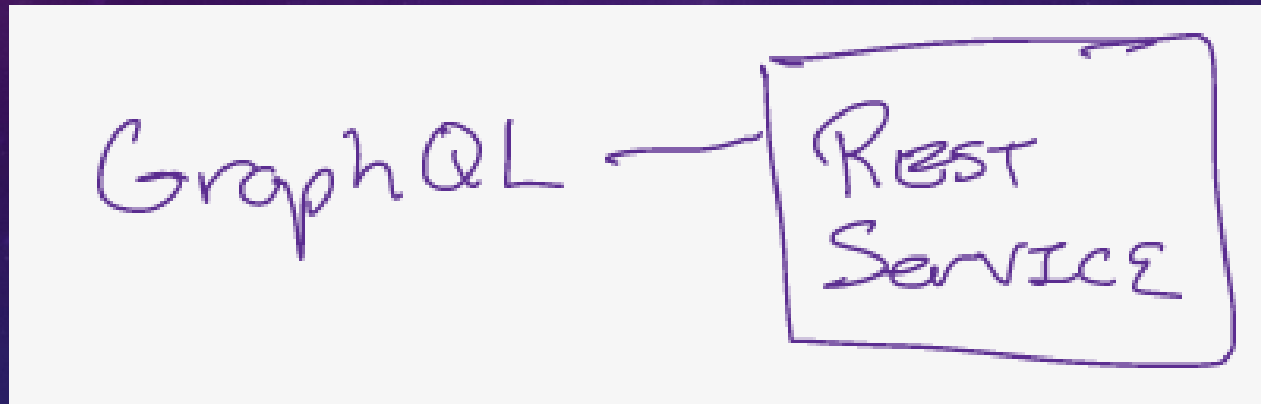
Let's see some code!

COMMON USE CASES



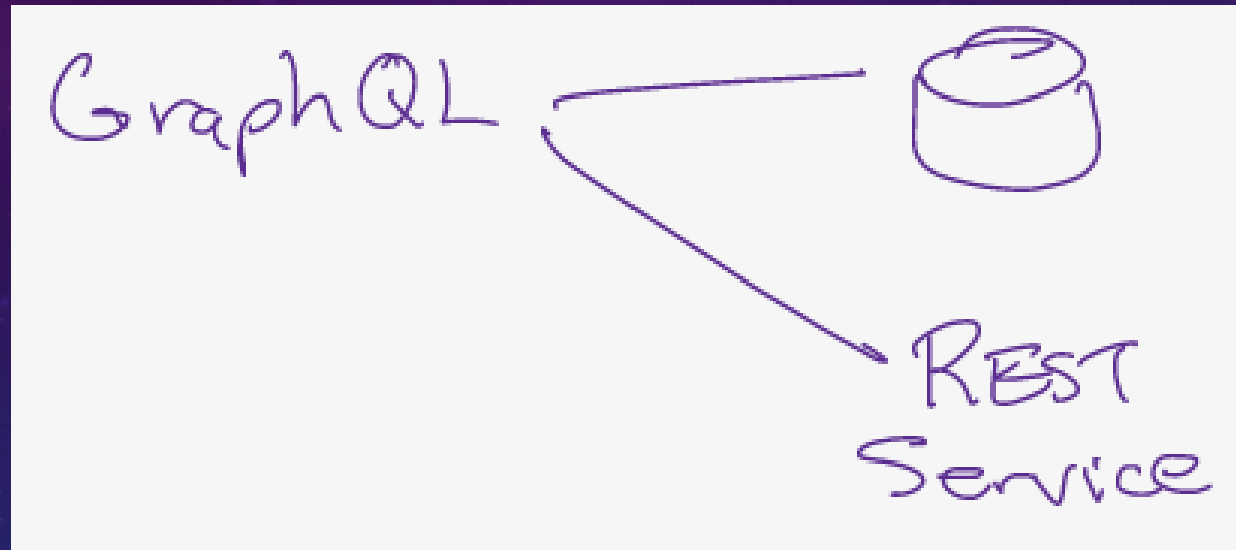
- GraphQL server with a connected database

COMMON USE CASES



- GraphQL server that serves as a façade

COMMON USE CASES



- Hybrid of the above two approaches

ISSUES

- Request Caching
- Security
 - Timeouts
 - query depth
 - query complexity

ADDITIONAL CONCEPTS

- Mutations
- Subscriptions
- Deferred query
- Union queries
- Pagination
- Query Variables
- Fragments






```
query MyTestQuery($repoOwner: String!, $repoName: String!) {  
  repository(owner: $repoOwner, name: $repoName) {  
    name  
    description  
    createdAt  
    forkCount  
    issues(first: 10, after: "Y3Vyc29yOnYyOjPHOBYY9Ew==") {  
      totalCount  
      edges {  
        cursor,  
        node {  
          title  
        }  
      }  
    }  
  }  
}
```



QUERY VARIABLES



```
{  
  "repoOwner": "facebook",  
  "repoName": "graphql"  
}
```




```
query MyTestQuery {  
  viewer {  
    repositories(first: 10) {  
      nodes {  
        name  
        ...RepoInfo  
      }  
    }  
  }  
}
```



```
fragment RepoInfo on Repository {  
  description  
  foundAt: createdAt  
  forkCount  
}
```

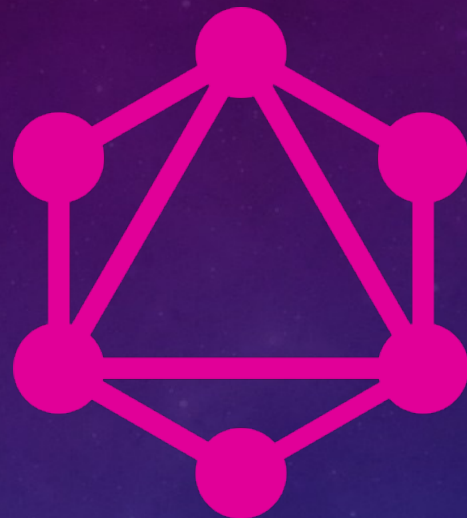


```
1 query MyTestQuery($showName: Boolean!) {
2   viewer {
3     repositories(first: 10) {
4       nodes {
5         name @include(if: $showName)
6         ...RepoInfo
7       }
8     }
9   }
10 }
11
12 fragment RepoInfo on Repository {
13   description
14   foundAt: createdAt
15   forkCount
16 }
17
```

QUERY VARIABLES



```
1 {
2   "showName": false
3 }
```

WHERE DO I GO FROM HERE?

EXAMPLE APIs

- <https://graphcms.com/blog/the-star-wars-api-just-got-a-makeover/>
 - <https://api.graphcms.com/simple/v1/swapi>
- <https://github.com/APIs-guru/graphql-apis>
 - <https://api.hollowverse.com/playground>
- GraphQL-faker

RESOURCES


- [GraphQL.org](https://graphql.org)
- github.com/steveluscher/zero-to-graphql
- howtographql.com
- Express-graphql
- Extensions for VS Code


CHANGE INTROSPECTION


Type List


Root **root** 
No Description


Film 
A single film.

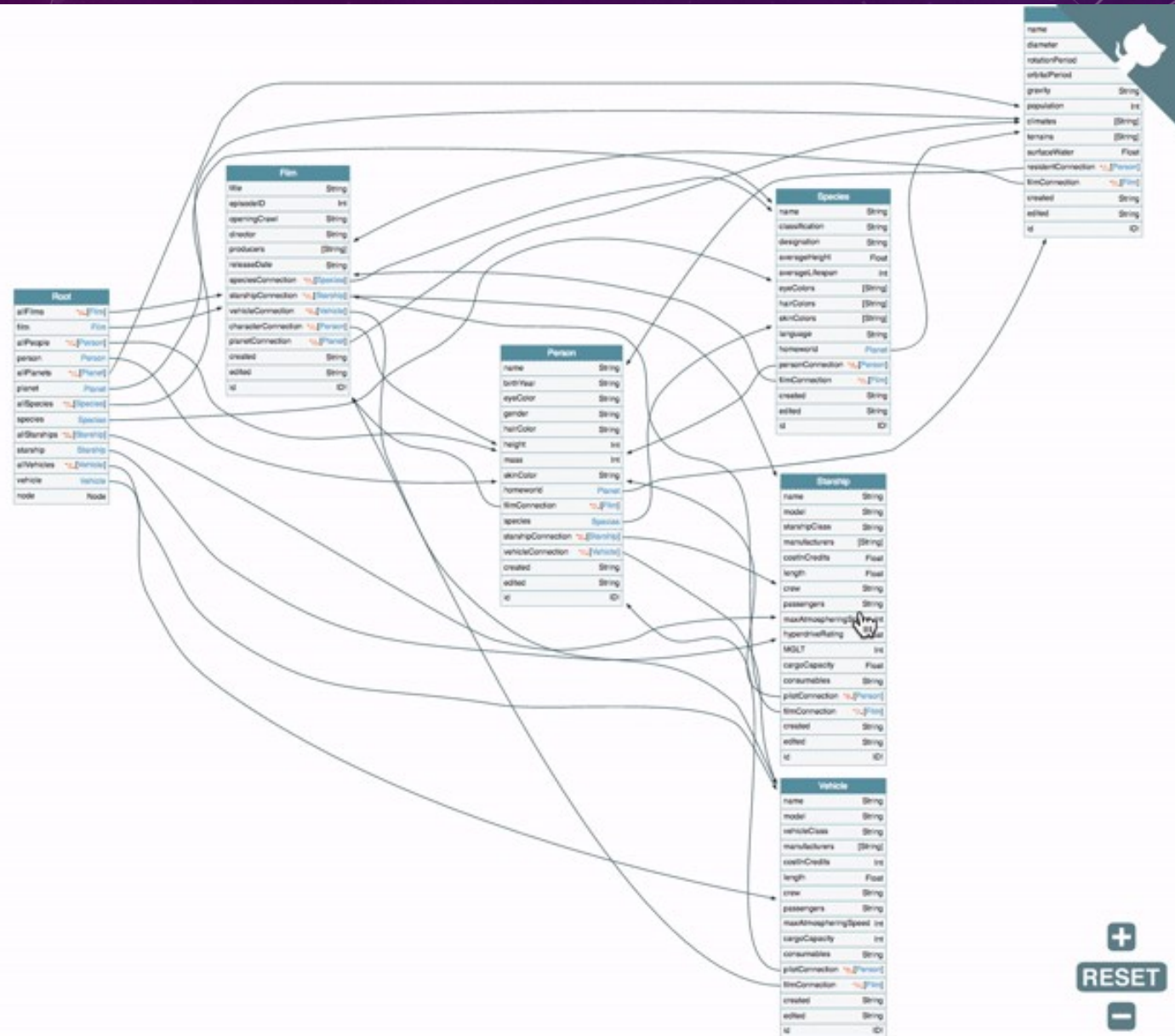
Person 
An individual person or character within the Star Wars universe.

Planet 
A large mass, planet or planetoid in the Star Wars Universe, at the time of 0 ABY.

Species 
A type of person or character within the Star Wars Universe.

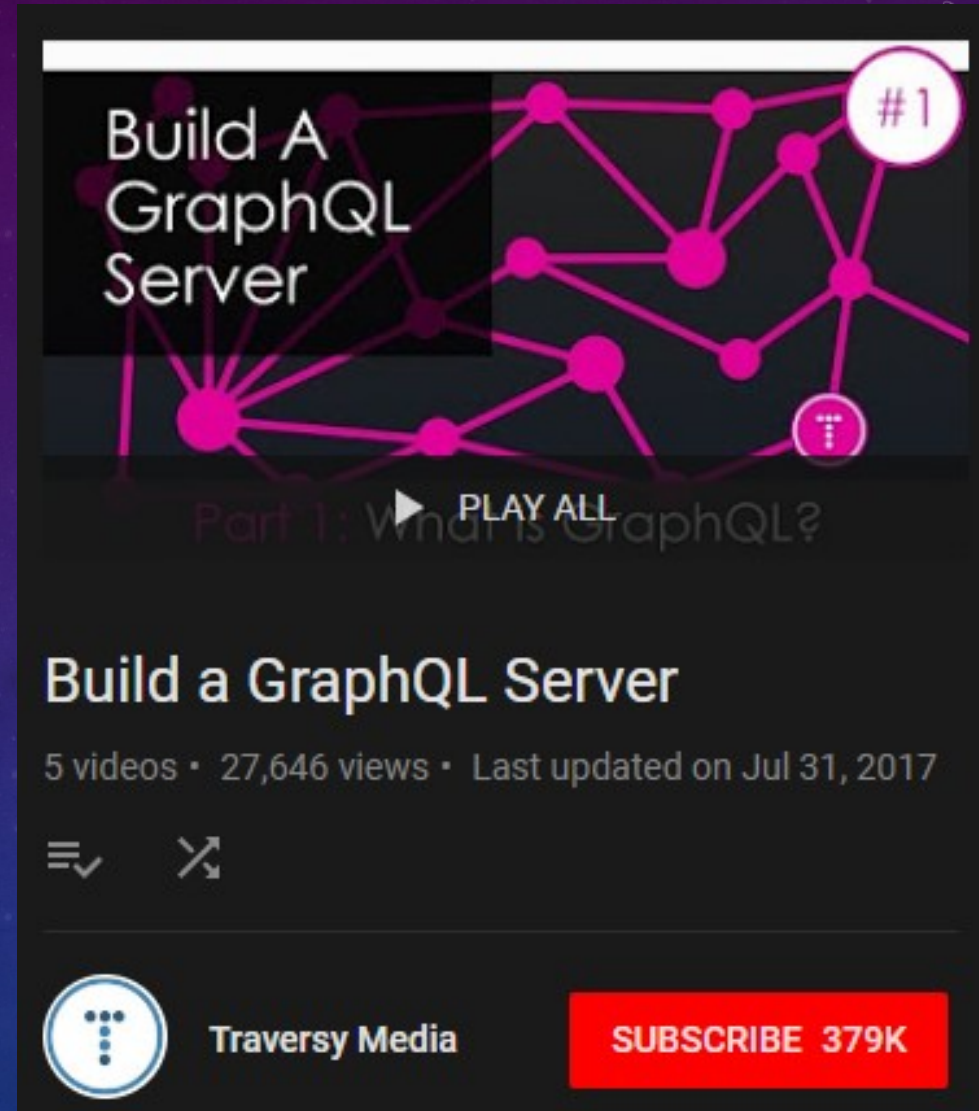
Starship 
A single transport craft that has hyperdrive capability.

Vehicle 
A single transport craft that does not have hyperdrive capability



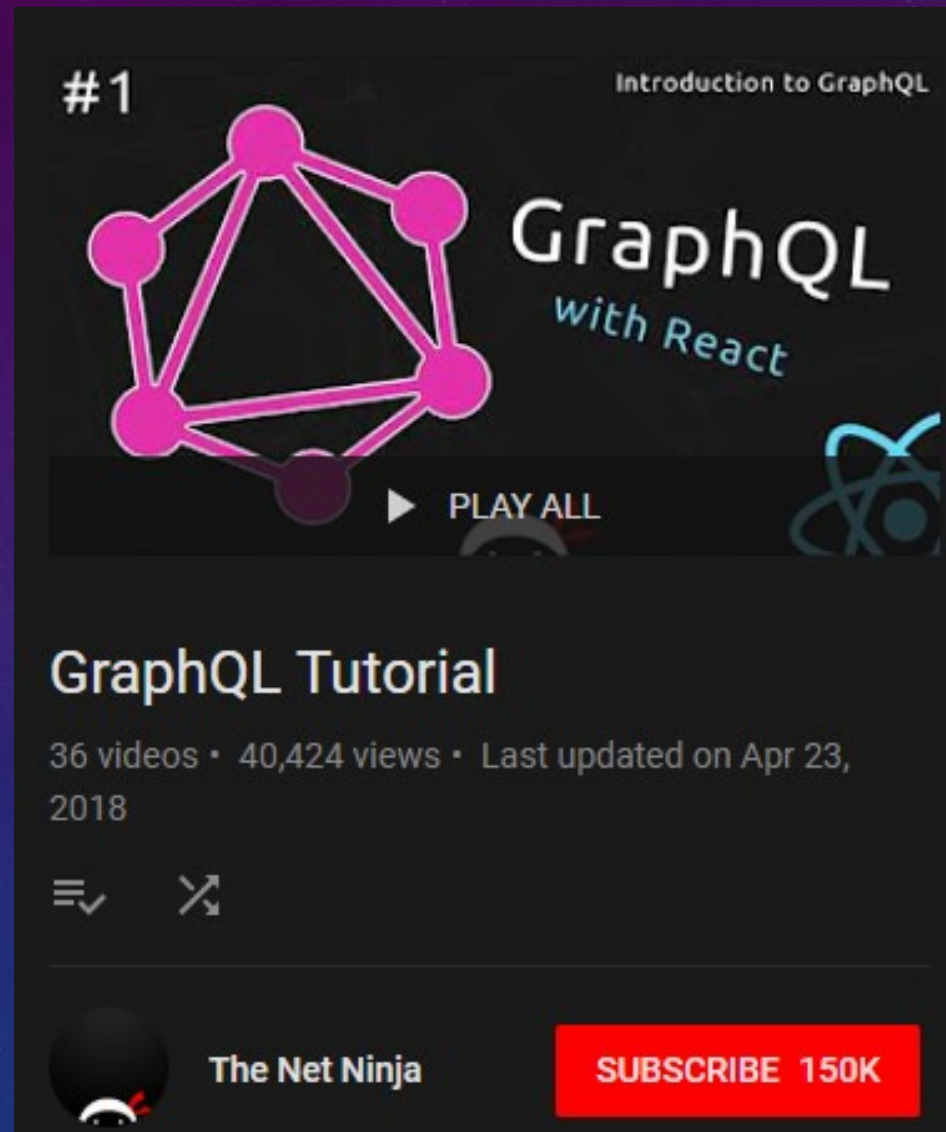
- Schema Exploration: apis.guru/graphql-voyager

VIDEO TRAINING



- https://www.youtube.com/playlist?list=PLlIGF-RfqbYZty73_PHBqKRDnv7ikh68

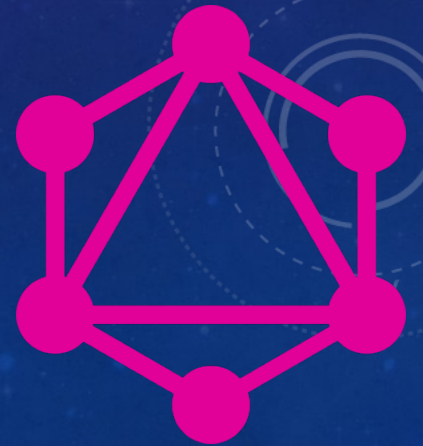
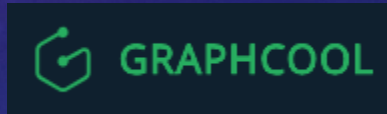
VIDEO TRAINING



- <https://www.youtube.com/playlist?list=PL4cUxeGkcC9iK6Qhn-QLcXCXPQUov1U7f>

EXPLORE RELATED PROJECTS

- Relay
- Apollo
- GraphCool
- GraphQL-CLI



Bryan Nehl – @k0emt – dbBear.com

<https://github.com/k0emt/Presentations>

<https://github.com/k0emt/graphql-exploration>

finished

