

INTRODUCTION TO GRAPHQL

BRYAN NEHL

@K0EMT

CORE CONCEPTS

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

CORE CONCEPTS





CORE CONCEPTS





CORE CONCEPTS – SIMILARITY 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.bId = b.Id;
```

CORE CONCEPTS – DIFFERENT THAN SQL



CORE CONCEPTS

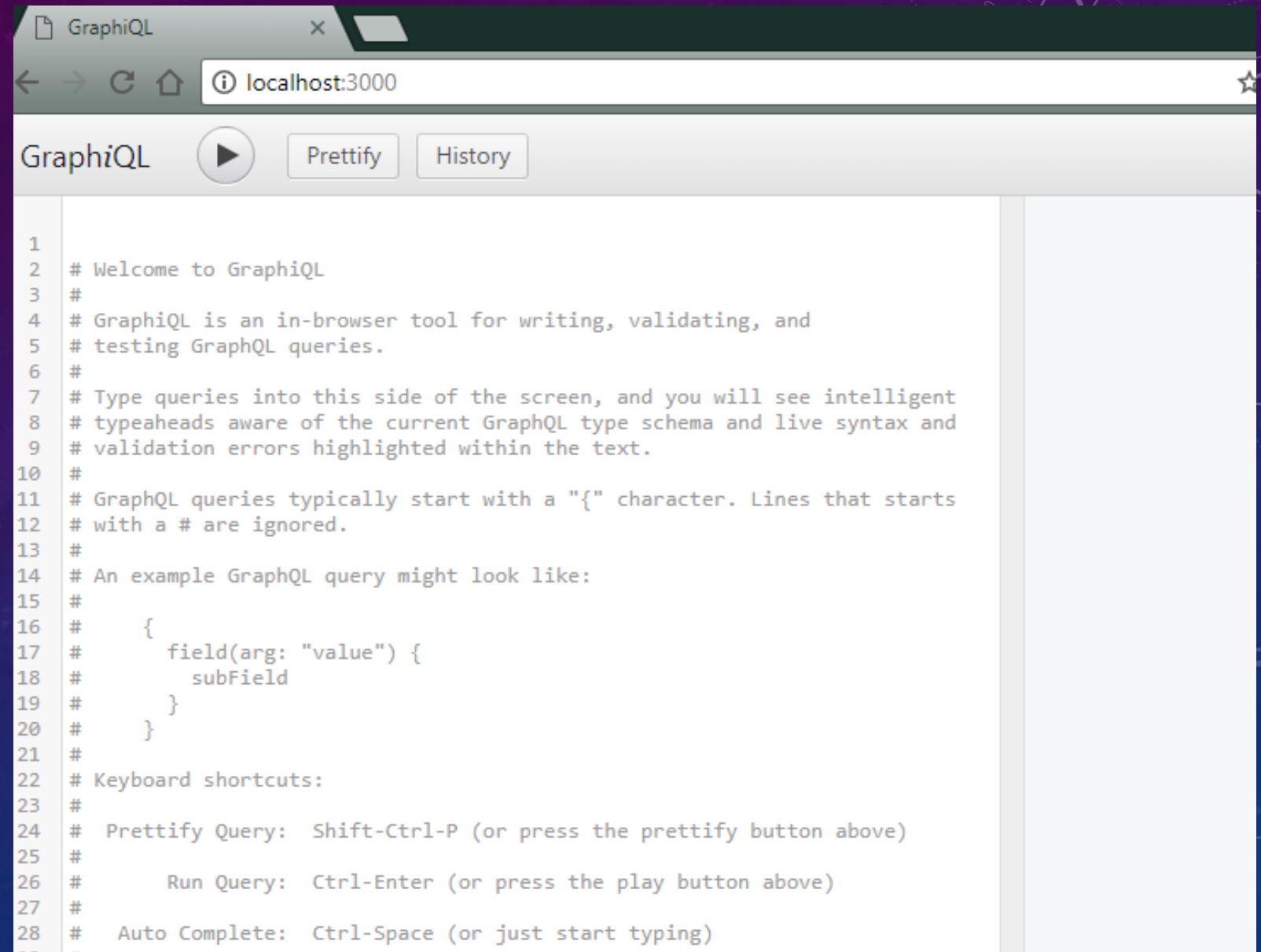
- **Schema**
 - SDL/IDL
 - Introspection & GraphQL Voyager
- **Query** for the **Data** you want
 - Contracts

INTERACTIVE CLIENTS

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

# INTERACTIVE CLIENTS

- Curl
- **GraphiQL**
- Insomnia



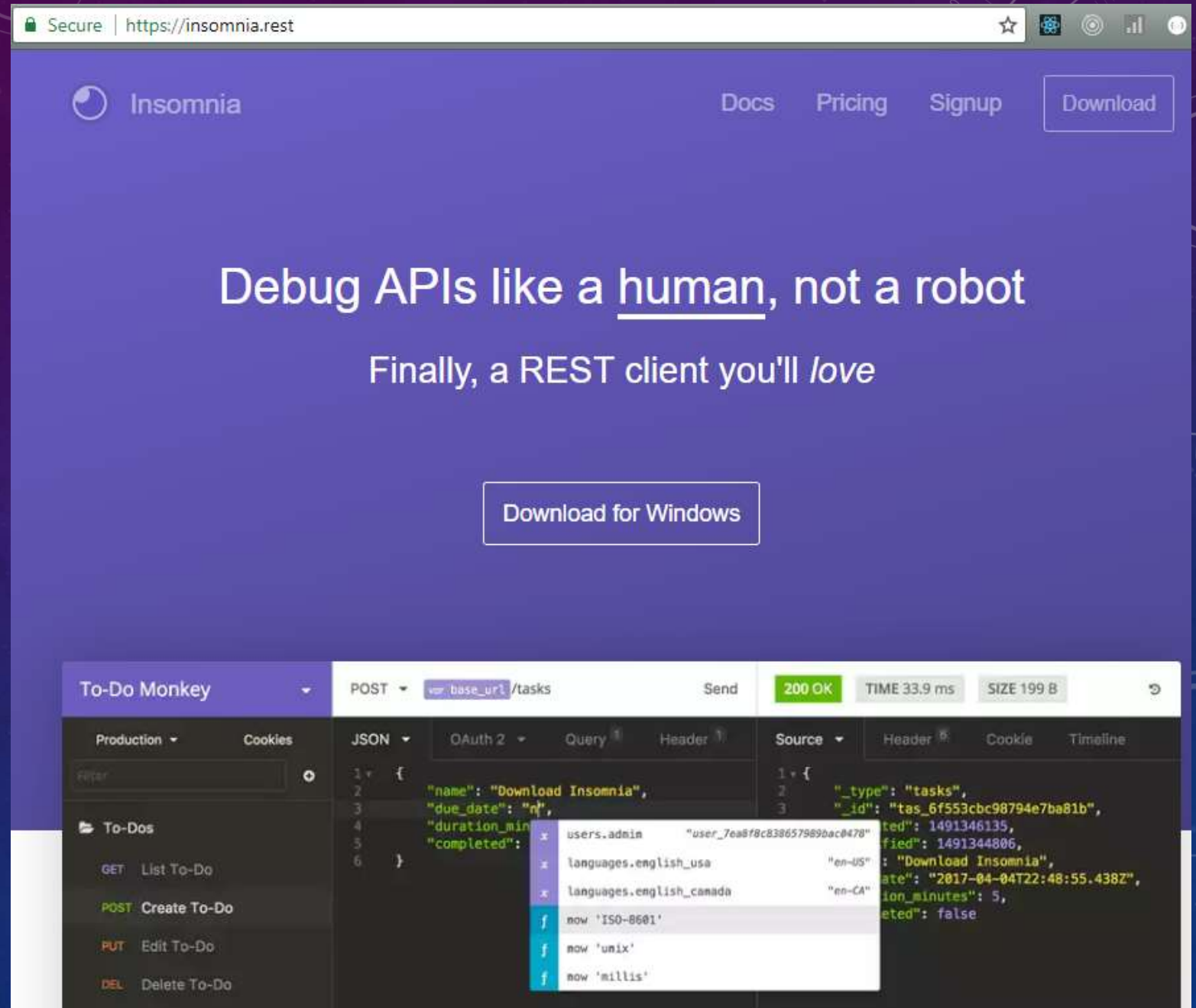
The screenshot shows the GraphiQL web interface in a browser window. The address bar indicates the URL is localhost:3000. The interface has a header with the title 'GraphiQL', a play button, and buttons for 'Prettify' and 'History'. The main area is a text editor displaying a welcome message and a sample GraphQL query.

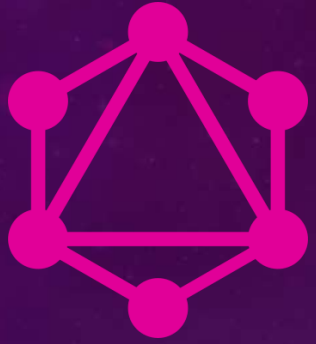
```
1
2 # Welcome to GraphiQL
3 #
4 # GraphiQL is an in-browser tool for writing, validating, and
5 # testing GraphQL queries.
6 #
7 # Type queries into this side of the screen, and you will see intelligent
8 # typeahead aware of the current GraphQL type schema and live syntax and
9 # validation errors 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
- GraphQL
- **Insomnia**





Let's see some code!

## COMMON USE CASES

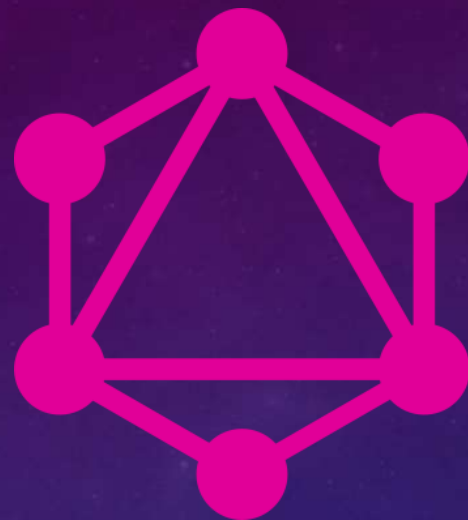
- GraphQL server with a connected database
- GraphQL server that serves as a façade
- Hybrid of the above two approaches





# ISSUES

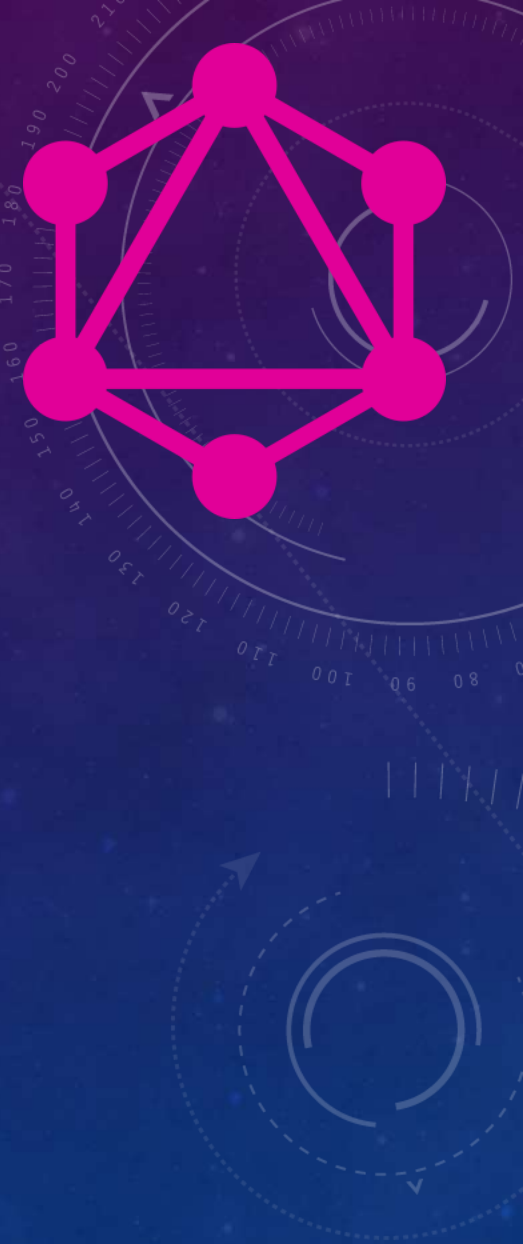
- Request Caching
- Security
  - Timeouts
  - query depth
  - query complexity



WHERE DO I GO FROM HERE?

## EXPLORE 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: "Y3Vyc29yOnYyOpH0BY9Ew==") {
 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 createdAt: foundAt
15 forkCount
16 }
17
```

#### QUERY VARIABLES



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



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


## CHANGE INTROSPECTION


### Type List


Root **root**   
No Description


Film   
A single film.

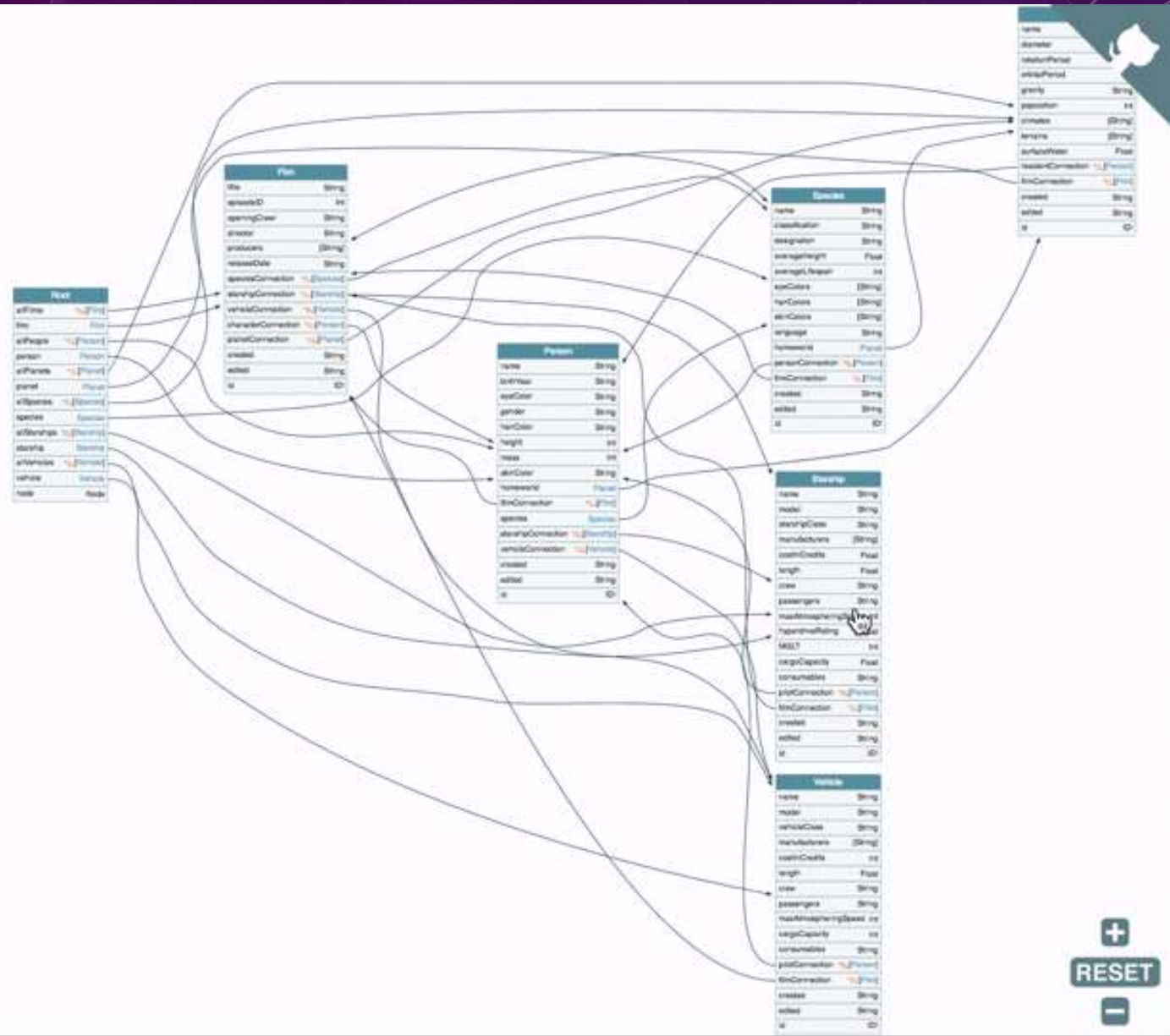
Person   
An individual person or character within the Star Wars universe.

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



**RESET**

- Schema Exploration: [apis.guru/graphql-voyager](https://apis.guru/graphql-voyager)

# VIDEO TRAINING




- [https://www.youtube.com/playlist?list=PLlIGF-RfqbYZty73\\_PHBqKRDnv7ikh68](https://www.youtube.com/playlist?list=PLlIGF-RfqbYZty73_PHBqKRDnv7ikh68)




# VIDEO TRAINING

#1 Introduction to GraphQL



## GraphQL with React


▶ PLAY ALL



### GraphQL Tutorial

36 videos • 40,424 views • Last updated on Apr 23, 2018

☰ ✕

 The Net Ninja

**SUBSCRIBE 150K**

- <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

