INTRO TO GRAPHQL + SCHEMA STITCHING IN GATSBY

JOHN ALBIN WILKINS / 25 SEPTEMBER 2019

Amazee Labs

ABOUT ME

JOHN ALBIN WILKINS // Senior Front-end Developer and Breaker of Designs that Expect Short Content

john.albin@amazeelabs.com

drupal.org/u/johnalbin

@JohnAlbin

AGENDA

TALKING POINTS

1. What is GraphQL? Why is GraphQL?
   (A short history of web APIs)
2. GraphQL Terminology & Query Basics
3. Schema stitching

A Short History Of Web APIs

Amazee Labs

SQL

THE ORIGINAL DATA QUERY

```
SELECT symbol, unicode
  FROM emojis
 WHERE name='exploding head'
```

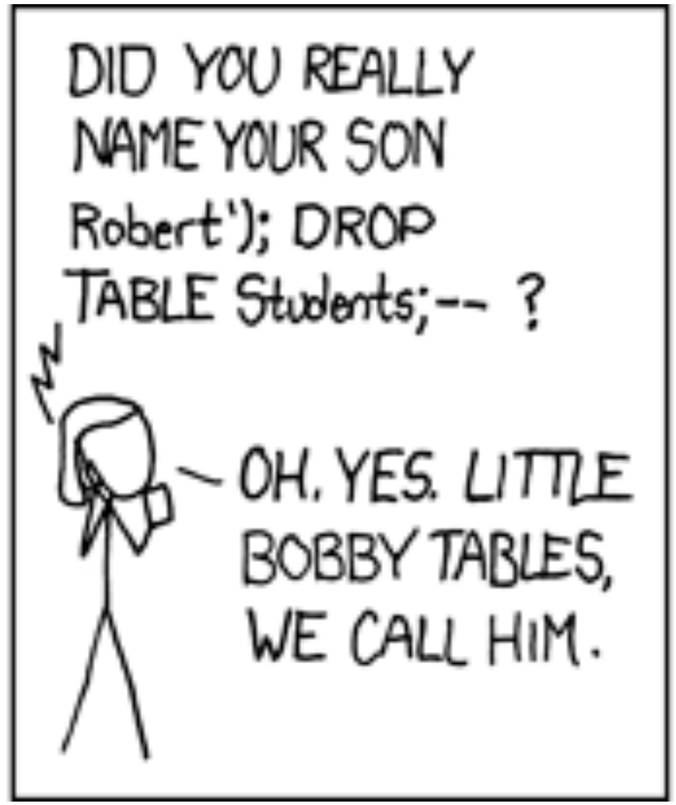
Symbol	unicode
	U+1F92F

SQL + INTERNET

UNSAFE AT ANY SPEED

```
INSERT INTO
    Students (name)
      VALUES (' ')
```

add text from web



"Exploits of a Mom"
https://xkcd.com/327/

COBRA vs XML-RPC vs SOAP

SAFER BUT NOT BETTER

```
<?xml version="1.0"?>
<soap:Envelope
 xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:m="http://www.example.org"
>
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <m:GetEmojiSymbol>
      <m:EmojiName>exploding head</m:AnimeName>
    </m:GetEmojiSymbol>
  </soap:Body>
</soap:Envelope>
```

REST

EVERYTHING IS A URL

SWAPI

The Star Wars API

people: https://swapi.co/api/people/

planets: https://swapi.co/api/planets/

films: https://swapi.co/api/films/

species: https://swapi.co/api/species/

vehicles: https://swapi.co/api/vehicles/

starships: https://swapi.co/api/starships/

JSON:API

A KIND OF REST API

LIVE DEMO — https://swapi.co

UNDER-FETCHING & OVER-FETCHING

PROBLEMS WITH REST

LIVE DEMO —

Get a list of character names in “The Empire Strikes Back”

GraphQL

query → **endpoint** → **data**

one URL for all data

NO MORE UNDER-FETCHING & OVER-FETCHING

GRAPHQL QUERIES

LIVE DEMO —

Get a list of character names in “The Empire Strikes Back”

SHORT Q&A

THE WHY OF GRAPHQL

GraphQL

TERMINOLOGY

```
Potato: {
   leaves: Emoji,
}
```

query

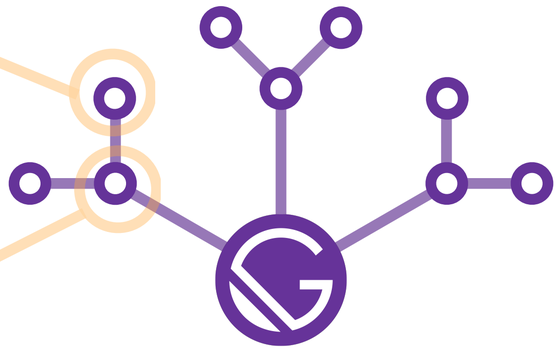
```
Query {
   potato {
      leaves
   }
   radish
   carrot
}
```

fields

root fields

endpoint

one URL for all data



data

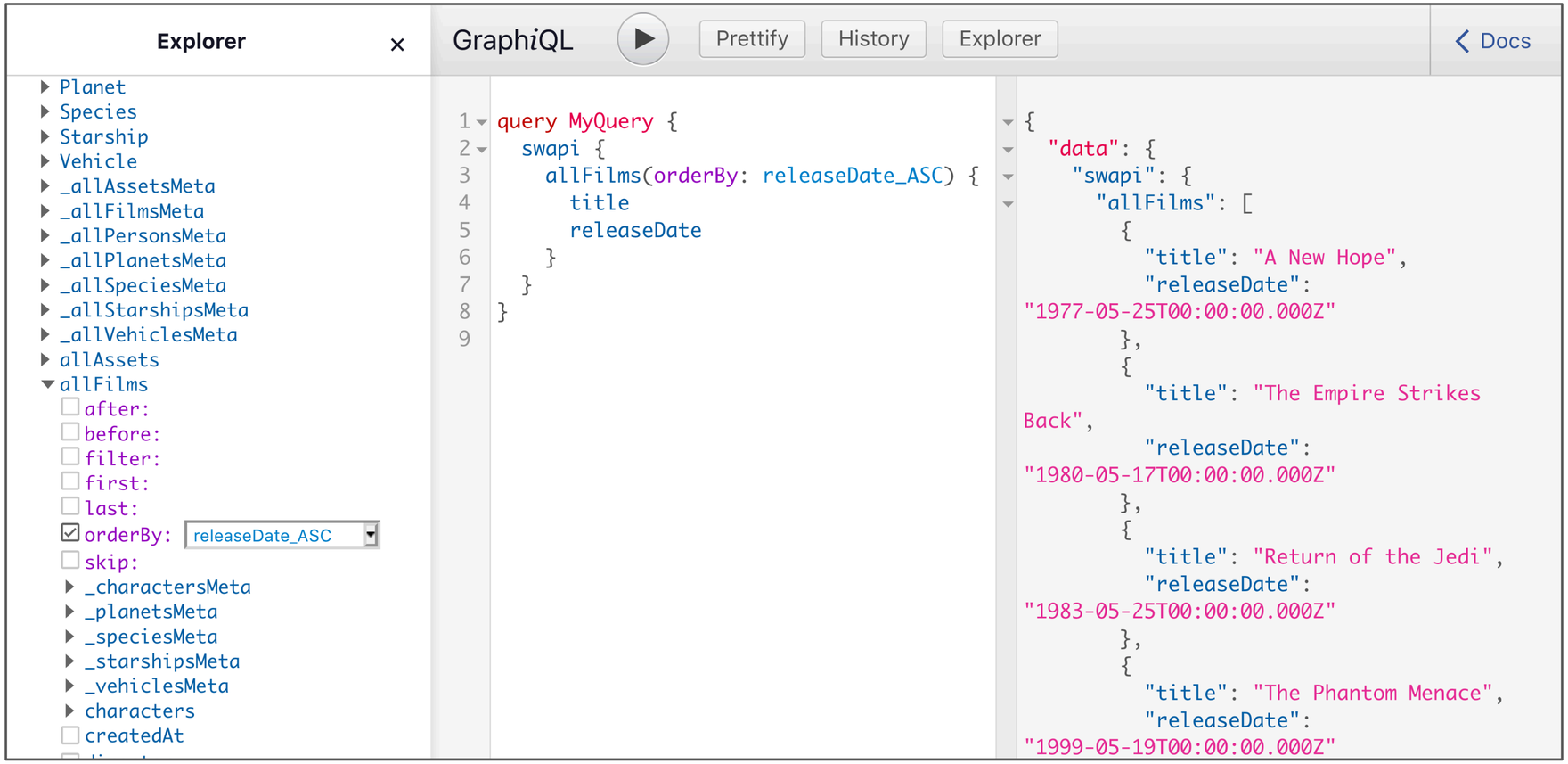
```
{
   potato: {
      leaves: 🌿,
   },
   radish: "spicy",
   carrot: 🥕,
}
```

schema: description of all ***fields*** and their ***types***

type: strong, strict definition of data

Graphiql

THE GRAPHQL DEV TOOL



FIELD ARGUMENTS

```
Query {
  human(id: "1000") {
    name
    height(unit: FOOT)
  }
}
```

QUERY VARIABLES

REUSABLE QUERIES

```
{
  “humanId": "1000"
}
```

```
Query ($humanId: ID!) {
  human(id: $humanId) {
    name
    height
  }
}
```

ALIASES

RENAMING FIELDS

```
query {
  leia: person(personID: 5) {
    name
    homeworld {
      name
    }
  }
  bobafett: person(personID: 22) {
    name
    homeworld {
      name
    }
  }
}
```

QUERY FRAGMENTS

REUSABLE PIECES OF QUERIES

```
query {
  gitHub {
    user(login: "johnalbin") {
      name
    }
    org1: organization(login: "amazeelabs") {
      ...orgFragment
    }
    org2: organization(login: "kss-node") {
      ...orgFragment
    }
  }
}
```

```
fragment orgFragment on Organization {
  name
  url
  repositories(
    last: 3
    orderBy: { direction: DESC, field: CREATED_AT }
    privacy: PUBLIC
  ) {
    nodes {
      name
      isPrivate
    }
  }
}
```

SHORT Q&A

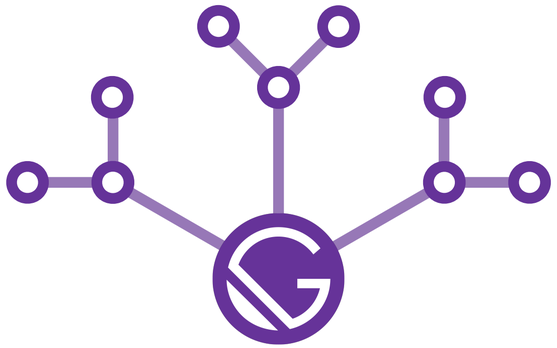
GRAPHQL TERMINOLOGY AND QUERY BASICS

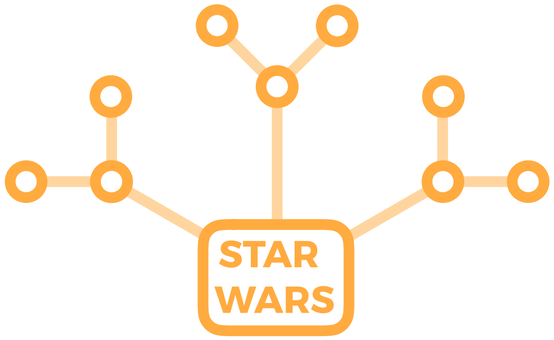
GraphQL's Killer Feature: Schema Stitching

Amazee Labs

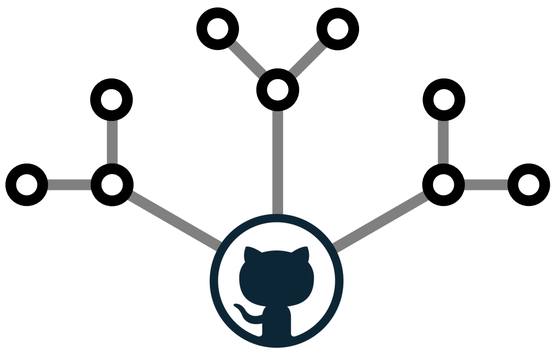
GraphQL POPULARITY

Just the new shiny?





Drupal core
+ GraphQL module



GraphQL
is more than
an endpoint

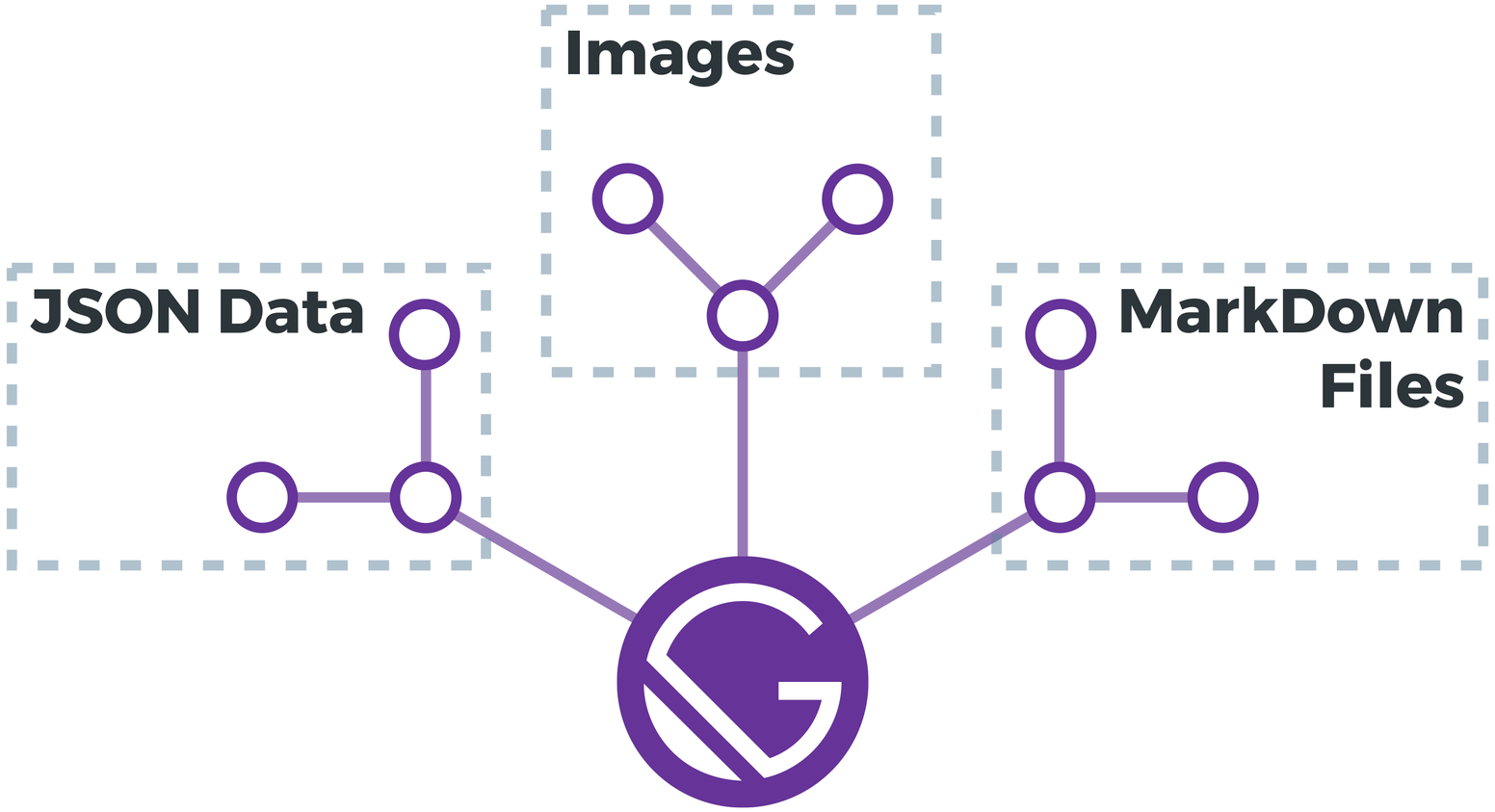
What makes GraphQL special?

REST 2.0

- **Single endpoint**
- **Query language**
- **Data-storage agnostic + per-field storage**

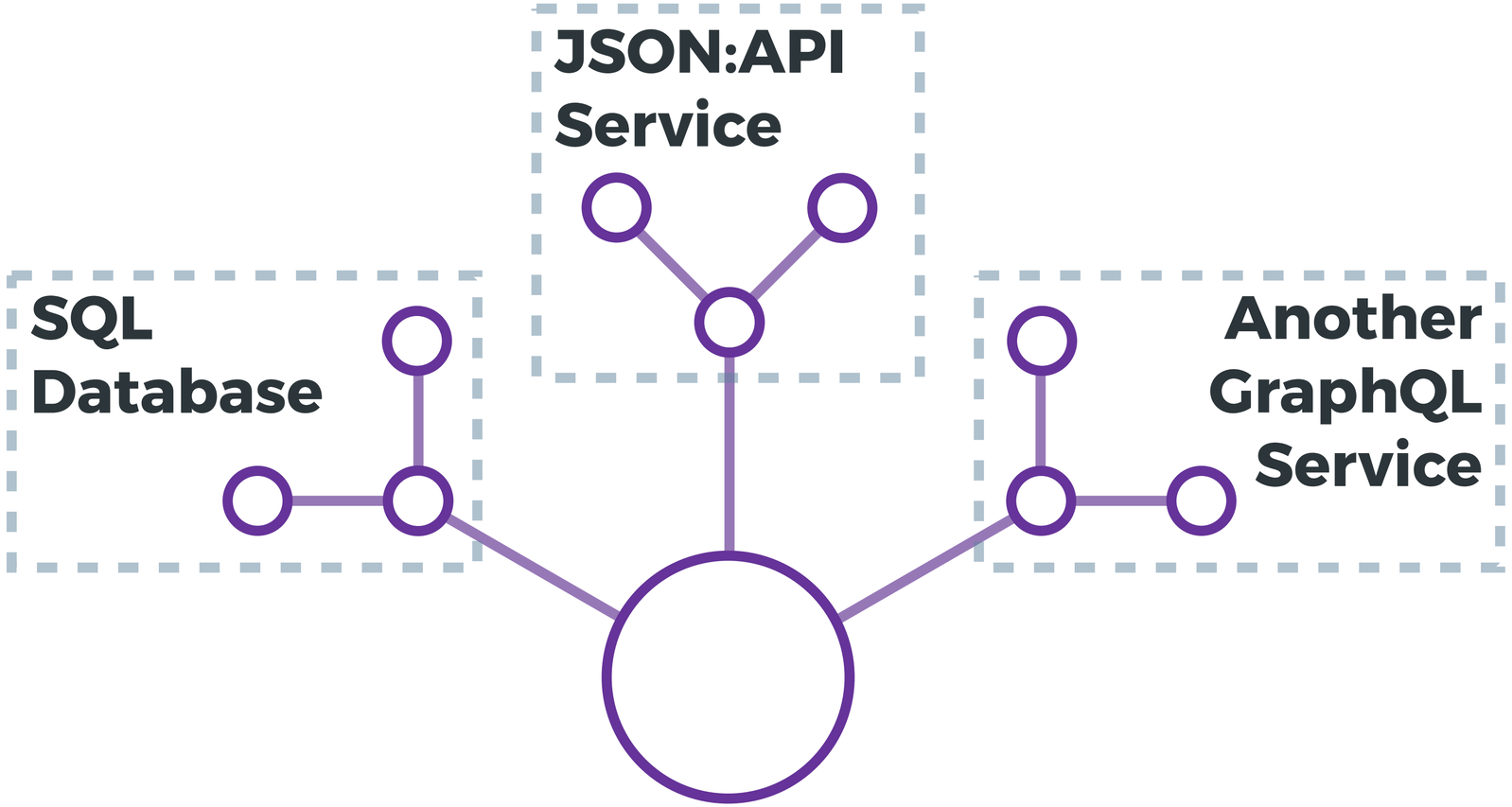
FIELD-SPECIFIC STORAGE

EXAMPLE: GATSBY PLUGINS



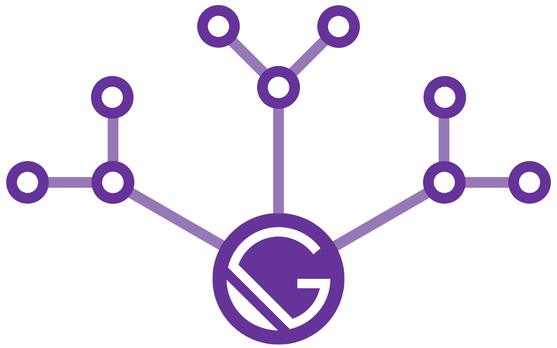
UNIFIED GRAPH

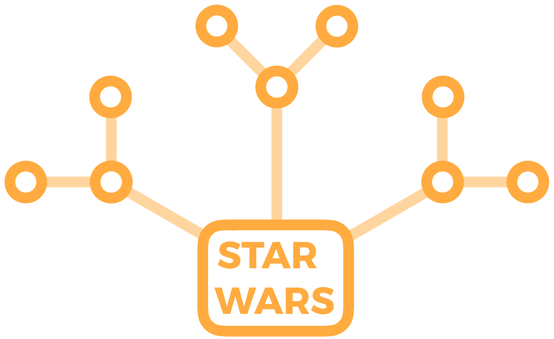
EACH FIELD CAN HAVE ITS OWN SOUCE

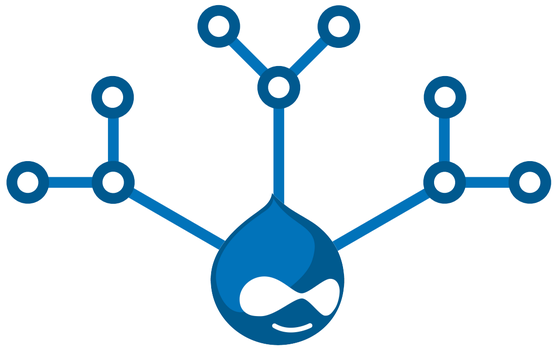


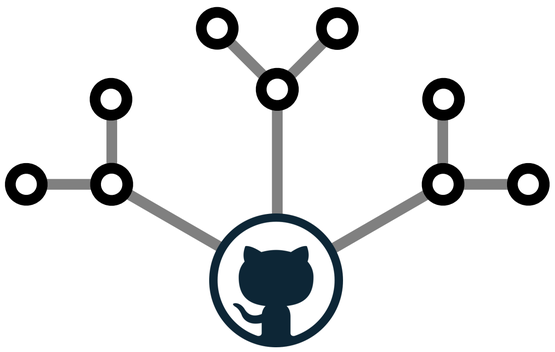
MERGING GRAPHS

SCHEMA STITCHING



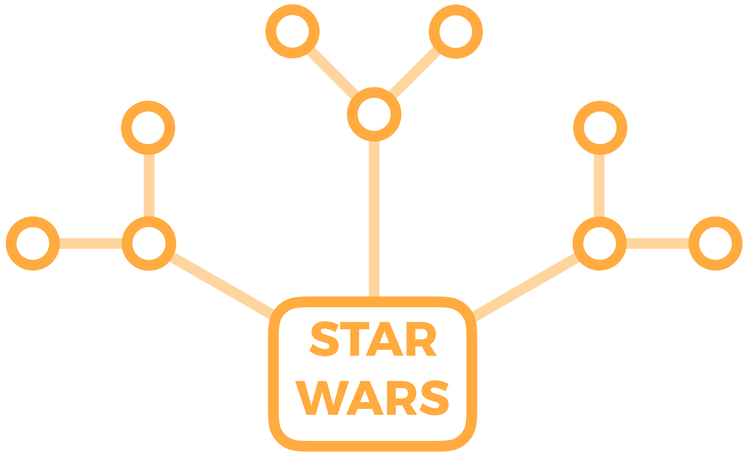


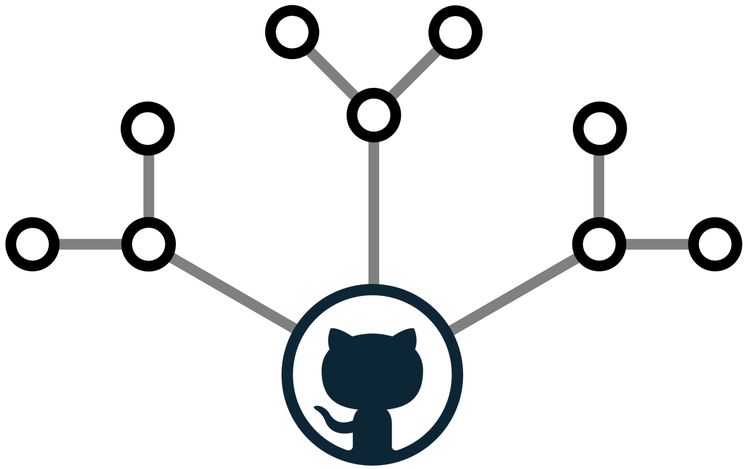




SCHEMA STITCHING

SIMPLE CASE



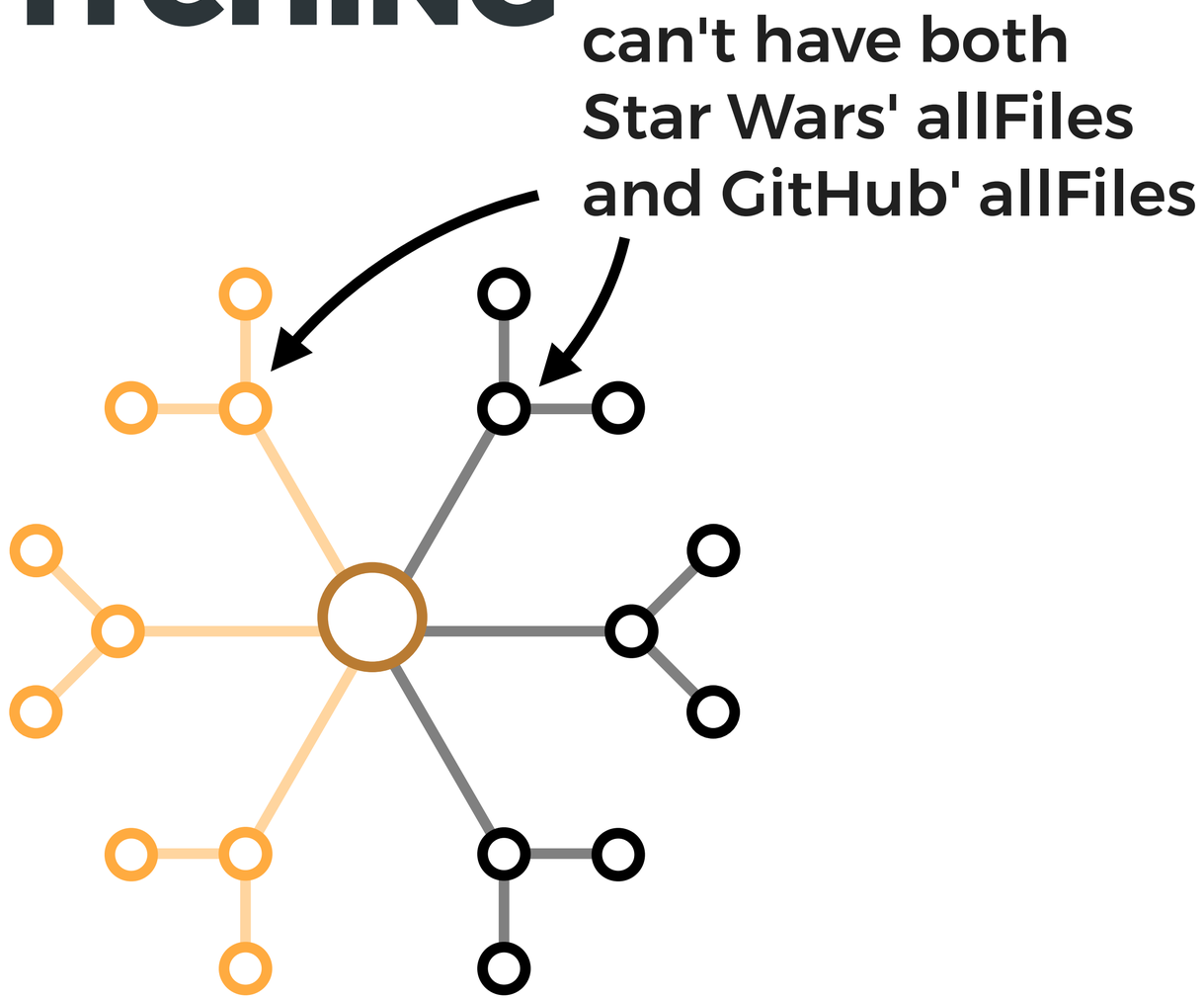


SCHEMA STITCHING

SIMPLE CASE

REQUIREMENTS:

- **Unique root fields**



SCHEMA STITCHING

SIMPLE CASE

REQUIREMENTS:

- **Unique root fields**
- **Unique types**

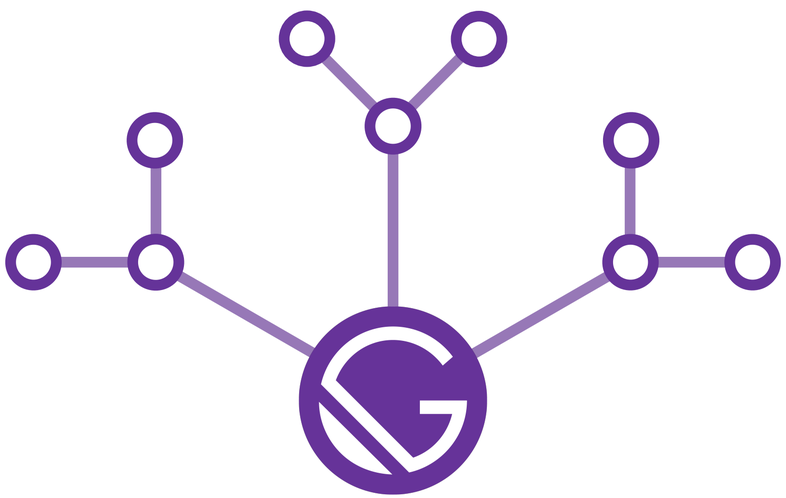
STAR WARS

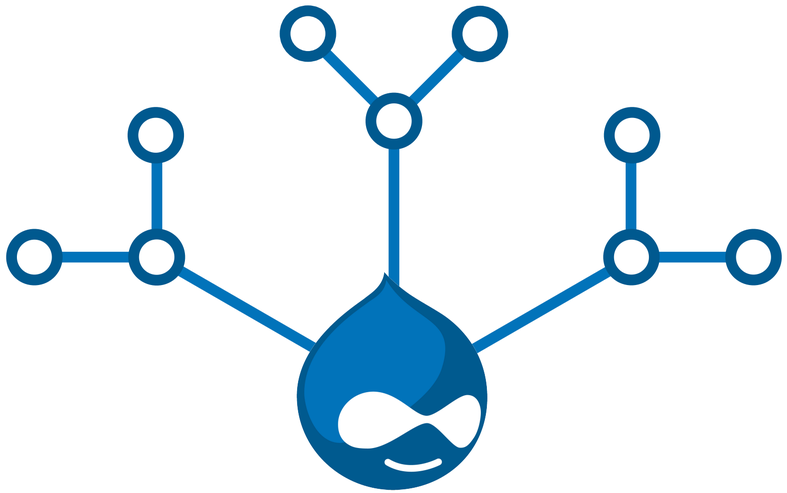
```
Person {
  name: String,
  weapons: Weapon[],
}
```

```
Person {
  name: String,
  repositories: Repo[],
}
```

SCHEMA STITCHING

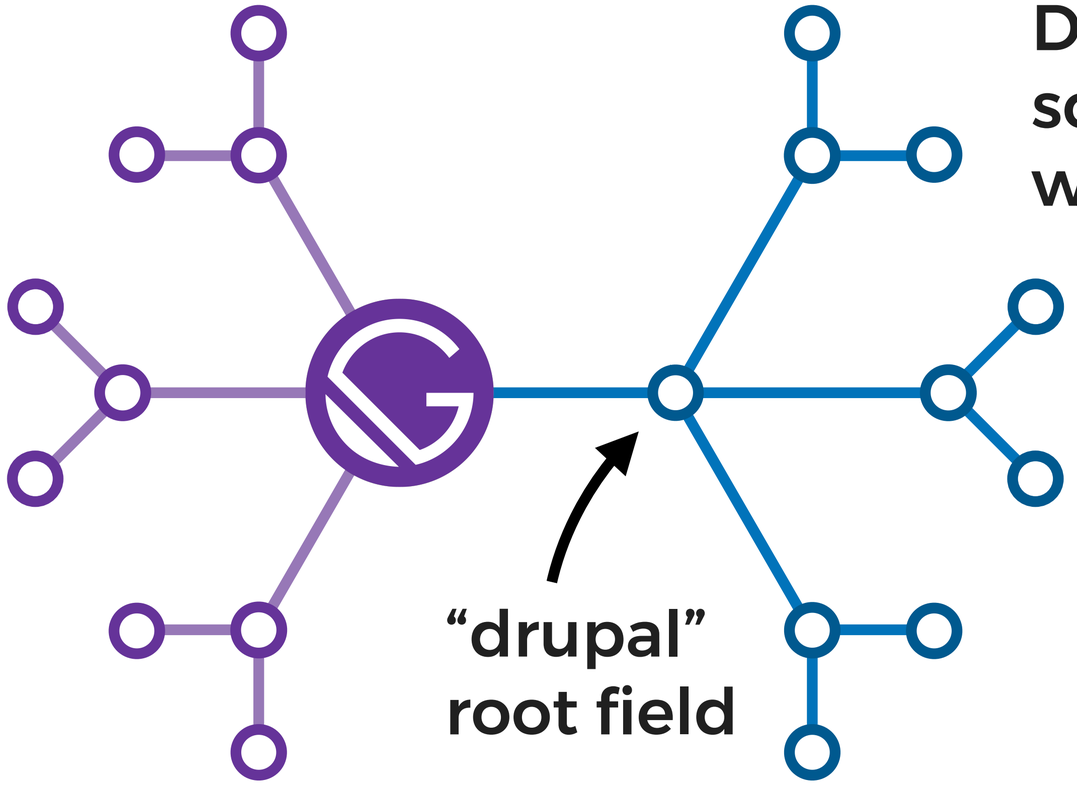
ADVANCED CASE





SCHEMA STITCHING

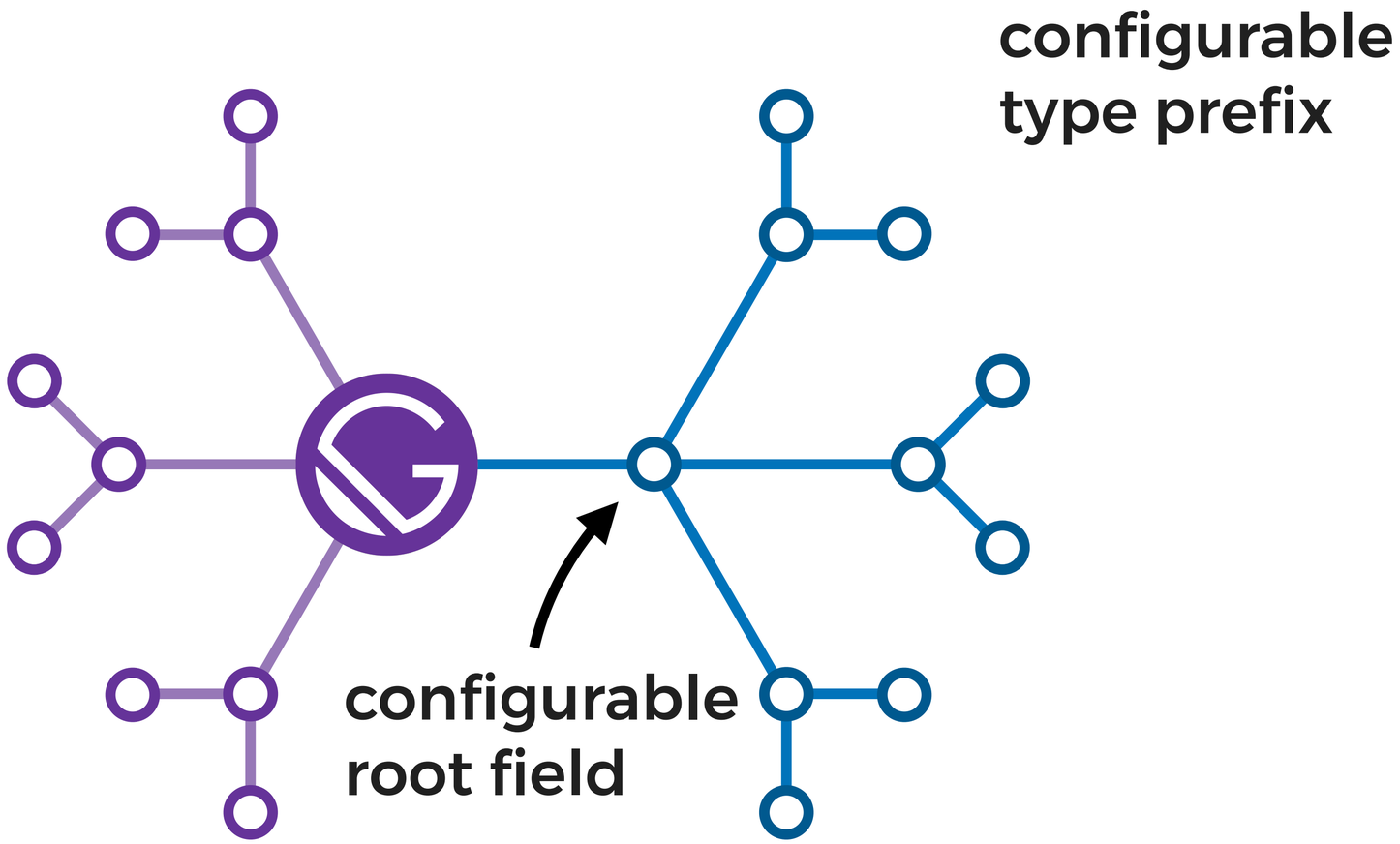
ADVANCED CASE



All types from Drupal GraphQL schema prefixed with “Drupal_”

GATSBY-SOURCE-GRAPHQL

ADVANCED CASE



GATSBY-SOURCE-GRAPHQL

CONFIGURATION

```
// gatsby-config.js

module.exports = {
  plugins: [
    {
      resolve: "gatsby-source-graphql",
      options: {
        typeName: `Drupal`,
        fieldName: `umami`,
        url: `https://umami.example.com/graphql`,
      }
    },
```

GATSBY-SOURCE-GRAPHQL

CONFIGURATION

```
// gatsby-config.js

module.exports = {
  plugins: [
    {
      resolve: "gatsby-source-graphql",
      options: {
        typeName: `Drupal`,
        fieldName: `umami`,
        url: `https://umami.example.com/graphql`,
      }
    },
    {
      resolve: "gatsby-source-graphql",
      options: {
        typeName: `SWAPI`,
        fieldName: `swapi`,
        url: `https://swapi.graph.cool/`
      }
    },
```

LIVE DEMO

GATSBY 2.5'S GRAPHQL APIs

LOW-LEVEL GRAPHQL APIs

https://www.gatsbyjs.org/blog/tags/graphql

Mikhail Novikov

GraphQL sommelier. Did Reindex, GraphQL Delegation, Schema Stitching, Launchpad. Organizer @GraphQLFinland.

@freiksenet

Improvements to Schema Customization API - Available in Gatsby 2.5.0

Today we are releasing further improvements to the schema customization that we've released in version 2.2.0 . You can use them with Gatsby...

Mikhail Novikov on May 17, 2019

QUESTIONS

AND MAYBE ANSWERS

THANK YOU

JOHN ALBIN WILKINS
SENIOR FRONT-END DEVELOPER

john.albin@amazee.com
drupal.org/u/johnalbin
@JohnAlbin

Amazee Labs