

GRANDstack: Graphs ALL The Way Down

William Lyon
@lyonwj
lyonwj.com

NODES 2019

Neo4j Online Developer
Expo and Summit

neo4j.com/online-summit



bit.ly/GRANDstack



William Lyon

@lyonwj

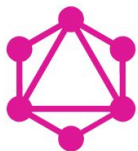
Neo4j Labs Engineer

Agenda

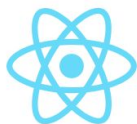
- What's a **GRANDstack**?
- Building a **#NODES2019 Recommendations App**
- **Hunger Games**



Fullstack GraphQL with GRANDstack



GraphQL



React



Apollo



Neo4j Database

- Fullstack framework for building applications

grandstack.io

What is GraphQL?

An API query language and runtime for building APIs



GraphQL

Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

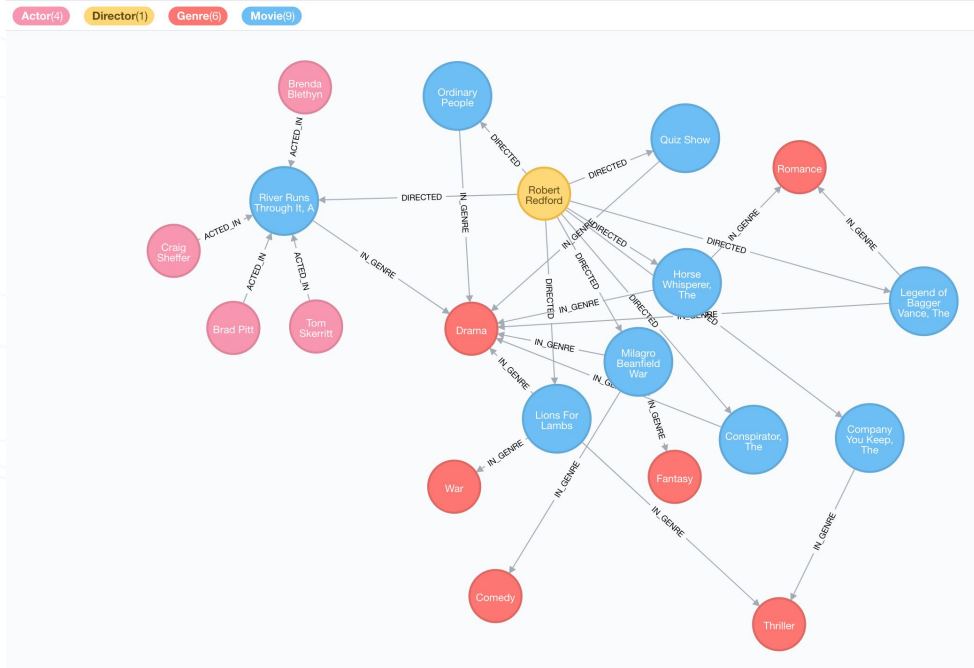
```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

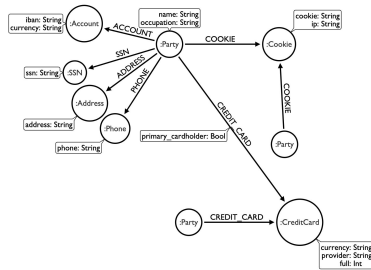
```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```

GraphQL.org

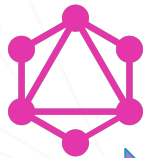
"Your Application Data Is A Graph" -- GraphQL



Expose A GraphQL API From Neo4j



```
1 type Address {
2   address: String!
3   city: String!
4   id: String!
5   state: String!
6   zip: String!
7   partys: [Party] @relation(name: "ADDRESS", direction: "IN")
8 }
9
10 type CreditCard {
11   currency: String!
12   full: String!
13   id: String!
14   provider: String!
15   type: String!
16   partys: [Party] @relation(name: "CREDIT_CARD", direction: "IN")
17 }
18
19 type Account {
20   currency: String!
21   iban: String!
22   id: String!
23   type: String!
24   partys: [Party] @relation(name: "ACCOUNT", direction: "IN")
25 }
26
27 type SSN {
28   ssn: String!
29   partys: [Party] @relation(name: "SSN", direction: "IN")
30 }
```



A screenshot of a GraphQL API interface. The top bar shows the endpoint `http://localhost:4001/graphql`. The main area displays a query and its response:

```
1 {
2   Party(case_id: "0ecd49889c08e296b6927c76ccd26388") {
3     last_name
4     sharedIdentitySize
5     sharedIdentity {
6       last_name
7       address {
8         address
9       }
10      account {
11        iban
12        currency
13      }
14    }
15  }
16 }
17
```

The response is a JSON object:

```
{
  "data": {
    "Party": [
      {
        "last_name": "Simmons",
        "sharedIdentitySize": 14,
        "sharedIdentity": [
          {
            "last_name": "Schmitt",
            "address": [
              {
                "address": "430 Timothy Glen"
              }
            ]
          }
        ]
      },
      {
        "last_name": "Chandler",
        "address": [
          {
            "address": "04175 Robert Stream"
          }
        ]
      }
    ]
  }
}
```

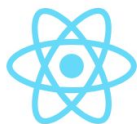
On the right side, there is a sidebar with a search bar and a list of queries and mutations:

- QUERIES: `getSuspiciousParties(...)`, `getFraudRing(...)`, `getFraudCommunity(...)`, `getAllCases(...)`, `Address(...)`, `CreditCard(...)`, `Account(...)`, `SSN(...)`, `Phone(...)`, `Party(...)`, `Case(...)`
- MUTATIONS: `CreateAddress(...)`, `UpdateAddress(...)`, `DeleteAddress(...)`

Fullstack GraphQL with GRANDstack



GraphQL



React



Apollo



Neo4j Database

- Fullstack framework for building applications

grandstack.io



William Lyon
@lyonwj

I thought we needed a @neo4j #NODES2019 session recommender #GRANDstack app so I took a stab at it. Try it out here:

nodes2019-app.grandstack.io

The screenshot shows a web browser displaying the 'Welcome To NODES GRANDstack' page. The main heading is 'NODES 2019 Session List'. Below the heading is a search bar with the text 'GRANDstack'. The page features a table of session recommendations with the following columns: Image, Name, Presenter, Description, and Recommended.

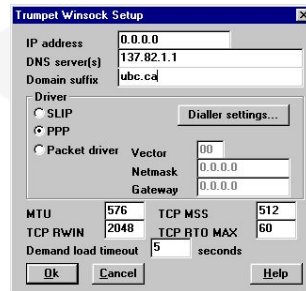
Image	Name	Presenter	Description	Recommended
	Discovering the Soul of a Product With Neo4j	Pat Brown	At Under Armour many disjointed systems are involved in bringing a product to market. Customer service reps and data scientists don't always have access to the data they need. We set out to capture that data from each system, expose it with the GRANDstack and make it available to our teammates.	Uncovering Medical Fraud in New York City's most Vulnerable Neighborhoods
	How the GRANDstack Makes Handling Complex Data Easy	Michael Porter	Using neo4j along with the GRANDstack has allowed me to build an application that deals with highly connected data in a manner I don't think would be possible with a SQL data base. The neo4j-graphql integration makes turning your Neo4j instance into a modern React application a snap.	Exploring the U.S. National Biodefense Index
	Modifying a Tournament	Michael	Scheduling and organizing a martial arts tournament can be a challenging task due to the variety of ages, rank, division, competitor, and judges that	Exploring the U.S. National

11:58 AM · Oct 10, 2019 · Twitter Web App

nodes2019-app.grandstack.io

Evolution of Web Dev

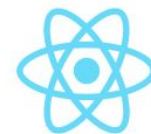
- **Mid 1990s:**
 - Static HTML
 - CGI and PHP
 - **LAMP Stack**
- **~2000s:**
 - **REST APIs, JSON**
 - jQuery
- **~2010s:**
 - NoSQL
 - **MEAN Stack**
 - **Meteor.js**



Evolution of Web Dev

2010s to...

- **React** (open sourced in 2013)
 - Mental model of server side JS to client
 - Move from event listeners to declarative actions about state
 - Virtual DOM
- **GraphQL** (open sourced in 2015)
 - Efficient data fetching
 - Data type system - describe and query your API as a g
- **Graph databases**
 - Intuitive graph datamodel
 - Native graph query performance
- **Deployment, serverless, DX over Cloud services**

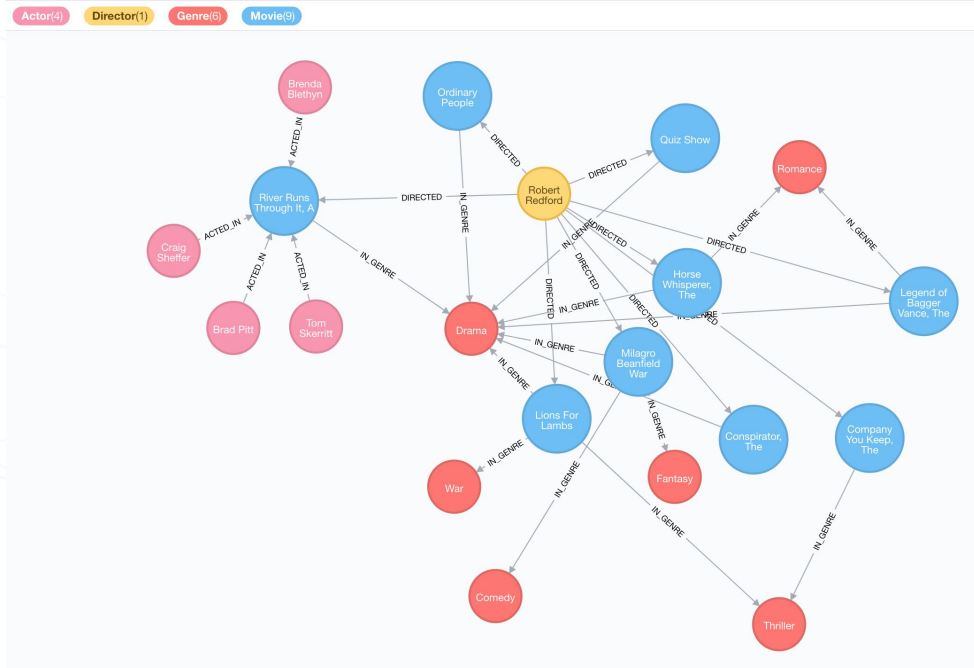


React



GraphQL

"Your Application Data Is A Graph" -- GraphQL



NODES Conference Graph



William Lyon
@lyonwj

Planning your schedule for #NODES2019 tomorrow but can't make sense of these pesky tables?

Now you can query the NODES conference graph directly in @neo4j:

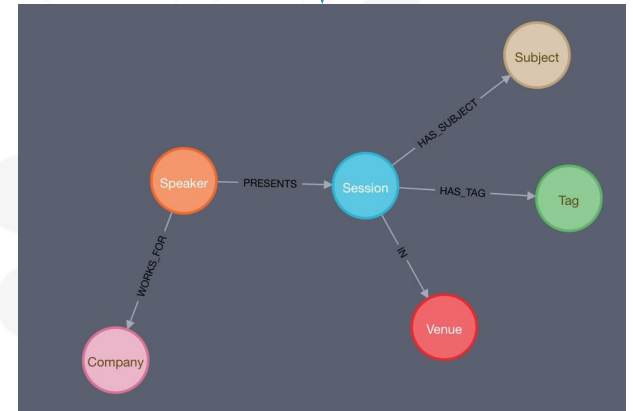
849a4698.databases.neo4j.io/browser/

Login with
user: nodes
password: nodes

Be sure to register here: neo4j.com/online-summit/

849a4698.databases.neo4j.io/browser/

5:30 PM Europe/Stockholm	Building a full-stack fraud-detection Solution ... Mark Needham (Neo4j)	Recommendations in UDC (Unified Data Catalog) P... Deepak Chandramouli (PayPal), Harsh Bhimani (PayPal inc)	GRANDstack: Graphs ALL the Way Down William Lyon (Neo4j)	How to build a knowledge graph from scratch eve... Wolfgang Hoeck (Live Data Concepts)	ML-based Graph Embeddings Frederick Russo (braintree)
5:50 PM Europe/Stockholm					We're not just relational a... Lisa Myers (Rice University)
6:15 PM Europe/Stockholm	Graphs in AI and ML Jake Graham (Neo4j), Alicia Frame (Neo4j)	Chaos engineering with Neo4j Janos Szendi Varga (Freelance)	APOC Pearls: The Best Tips & Tricks Michael Hunger (Neo4j)	Streaming Graph Data with Kafka Andrea Santurbano (Larus Business Automation)	Public Transport São Paulo ... igor Gustavo Oliveira Rozani (e-lorrel)
6:35 PM Europe/Stockholm					Exploring the U.S. National... Michael McKenzie (CALIBRE)



NODES Schedule GraphQL API

The screenshot shows a GraphQL IDE interface with a query editor on the left, a JSON response viewer in the center, and a sidebar on the right containing a search bar and a list of queries and mutations.

```
1 query sessionQuery(  
2   $first: Int  
3   $offset: Int  
4   $orderBy: [_SessionOrdering]  
5   $filter: _SessionFilter  
6 ) {  
7   Session(first: $first, offset: $offset,  
8     name  
9     description  
10    event_start {  
11      formatted  
12    }  
13    event_end {  
14      formatted  
15    }  
16    audience  
17    has_tag {  
18      name  
19    }  
20    has_subject {  
21      name  
22    }  
}
```

```
{  
  "data": {  
    "Session": [  
      {  
        "name": "Leveraging Graph Algorithms In Visualizations With  
        Neovis.js",  
        "description": "See how to use combine graph visualization and  
        graph algorithms to make sense of data in this live coding lightning  
        session. We'll see how we can use centrality &nbsp;and community detecti  
        algorithms to enhance a graph visualization using the Neovis.js graph  
        visualization library along with Neo4j Graph Algorithms.",  
        "event_start": {  
          "formatted": "2019-10-10T01:00:00Z"  
        },  
        "event_end": {  
          "formatted": "2019-10-10T01:20:00Z"  
        },  
        "audience": "Beginner",  
        "has_tag": [],  
        "has_subject": [  
          {  
            "name": "graph-visualization"  
          },  
          {  
            "name": "javascript"  
          },  
          {  
            "name": "data-science-algos"  
          },  
        ]  
      }  
    ]  
  }  
}
```

QUERY VARIABLES HTTP HEADERS

```
1 {  
2   "first": 3,  
3   "offset": 0,  
4   "orderBy": ["event_start_asc"],  
5   "filter": {}  
}
```

SEARCH the docs ...

QUERIES

- Venue(...): [Venue]
- Speaker(...): [Speaker]
- Company(...): [Company]
- Subject(...): [Subject]
- Tag(...): [Tag]
- Session(...): [Session]

MUTATIONS

- CreateVenue(...): Venue
- DeleteVenue(...): Venue
- AddVenueSessions(...):
 _AddVenueSessionsPayload
- RemoveVenueSessions(...):
 _RemoveVenueSessionsPayload
- CreateSpeaker(...): Speaker
- DeleteSpeaker(...): Speaker
- AddSpeakerWorks_for(...):
 _AddSpeakerWorks_forPayload
- RemoveSpeakerWorks_for(...):
 _RemoveSpeakerWorks_forPayload

nodes2019.grandstack.io

Movies, Genres, Directors, Actors



Movies, Genres, Directors, Actors

GraphQL Type Definitions

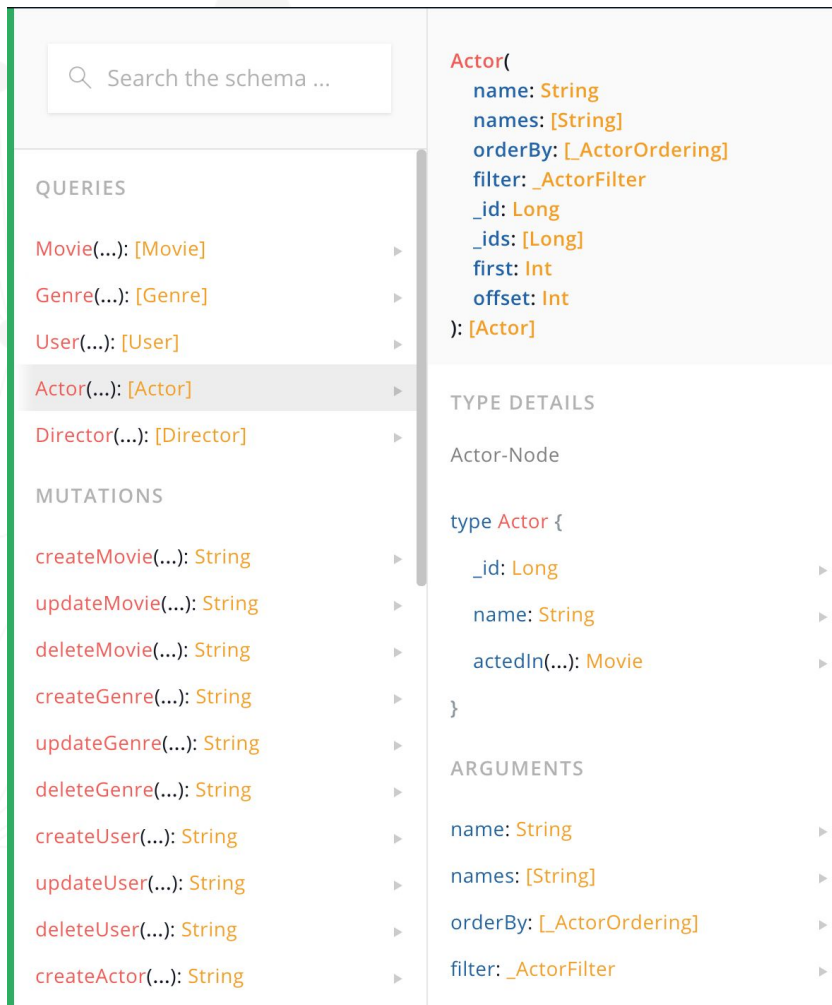
- Defined using GraphQL Schema Definition Language (SDL)

```
type Movie {  
  title: String  
  year: Int  
  plot: String  
  poster: String  
  imdbRating: Float  
  genres: [Genre]  
}  
  
type Genre {  
  name: String  
  movies: [Movie]  
}  
  
type Actor {  
  name: String  
  movies: [Movie]  
}  
  
type Director {  
  id: ID!  
  name: String  
}
```


Movies, Genres, Directors, Actors

Introspection

- Schema can be queried
- Schema becomes API specification / documentation
- Tools like GraphQL / GraphQL Playground



Search the schema ...

QUERIES

- Movie(...): [Movie]
- Genre(...): [Genre]
- User(...): [User]
- Actor(...): [Actor]**
- Director(...): [Director]

MUTATIONS

- createMovie(...): String
- updateMovie(...): String
- deleteMovie(...): String
- createGenre(...): String
- updateGenre(...): String
- deleteGenre(...): String
- createUser(...): String
- updateUser(...): String
- deleteUser(...): String
- createActor(...): String

Actor(
name: String
names: [String]
orderBy: [_ActorOrdering]
filter: _ActorFilter
_id: Long
_ids: [Long]
first: Int
offset: Int
): [Actor]

TYPE DETAILS

Actor-Node

```
type Actor {  
  _id: Long  
  name: String  
  actedIn(...): Movie  
}
```

ARGUMENTS

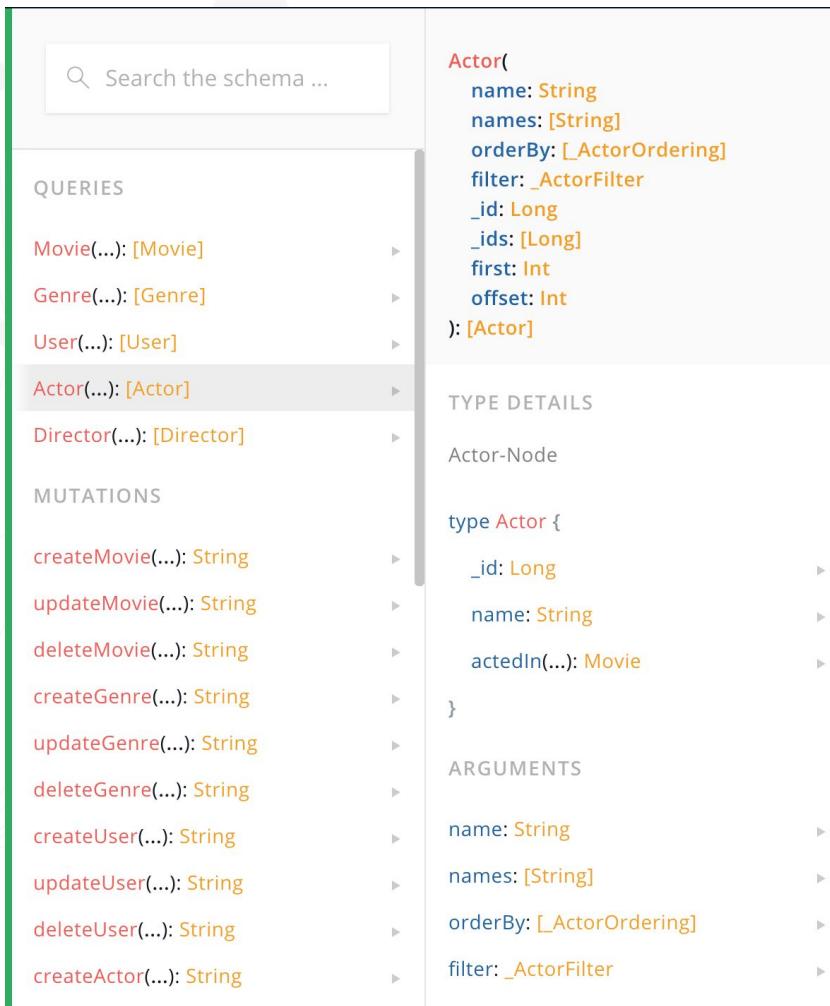
- name: String
- names: [String]
- orderBy: [_ActorOrdering]
- filter: _ActorFilter

Movies, Genres, Directors, Actors

```
{
  Movie(title: "River Runs Through It, A") {
    title
    actors(first: 2) {
      name
    }
    genres {
      name
    }
    directors {
      name
      movies(first: 3) {
        title
      }
    }
  }
}
```

GraphQL query

@lyonwj



Search the schema ...

QUERIES

- Movie(...): [Movie]
- Genre(...): [Genre]
- User(...): [User]
- Actor(...): [Actor]**
- Director(...): [Director]

MUTATIONS

- createMovie(...): String
- updateMovie(...): String
- deleteMovie(...): String
- createGenre(...): String
- updateGenre(...): String
- deleteGenre(...): String
- createUser(...): String
- updateUser(...): String
- deleteUser(...): String
- createActor(...): String

Actor(
 name: String
 names: [String]
 orderBy: [_ActorOrdering]
 filter: _ActorFilter
 _id: Long
 _ids: [Long]
 first: Int
 offset: Int
): [Actor]

TYPE DETAILS

Actor-Node

type Actor {
 _id: Long
 name: String
 actedIn(...): Movie
}

ARGUMENTS

- name: String
- names: [String]
- orderBy: [_ActorOrdering]
- filter: _ActorFilter

```
{
  Movie(title: "River Runs Through It, A") {
    title
    actors(first: 2) {
      name
    }
    genres {
      name
    }
    directors {
      name
      movies(first: 3) {
        title
      }
    }
  }
}
```

Operation name and arguments

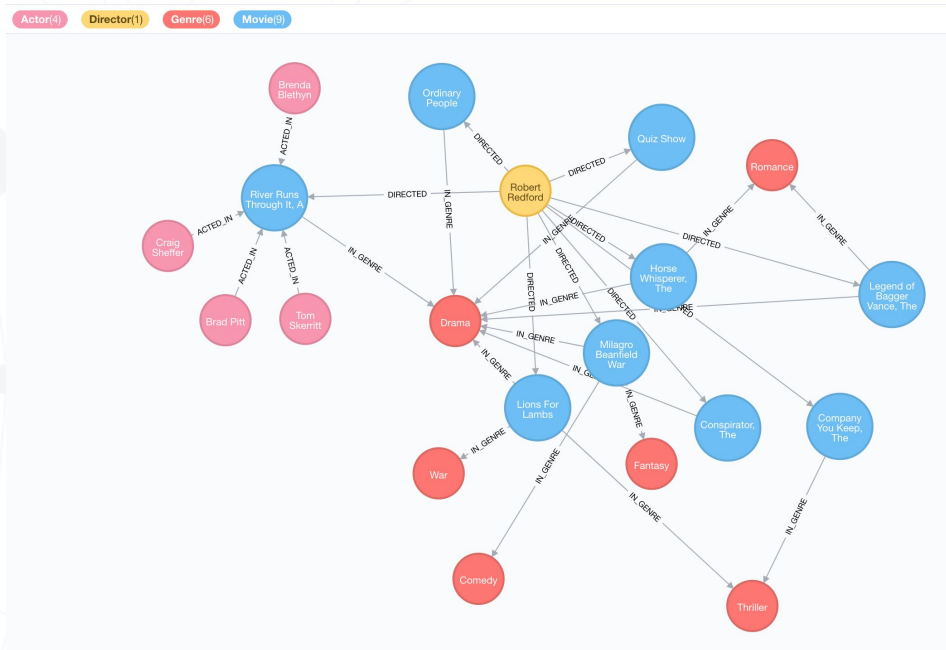
```
{
  Movie(title: "River Runs Through It, A") {
    title
    actors(first: 2) {
      name
    }
    genres {
      name
    }
    directors {
      name
      movies(first: 3) {
        title
      }
    }
  }
}
```

Selection set

Movies, Genres, Directors, Actors

```
{
  Movie(title: "River Runs Through It, A") {
    title
    actors(first: 2) {
      name
    }
    genres {
      name
    }
    directors {
      name
      movies(first: 3) {
        title
      }
    }
  }
}
```

GraphQL query



Movies, Genres, Directors, Actors

```
{  
  Movie(title: "River Runs Through It, A") {  
    title  
    actors(first: 2) {  
      name  
    }  
    genres {  
      name  
    }  
    directors {  
      name  
      movies(first: 3) {  
        title  
      }  
    }  
  }  
}
```

GraphQL query

```
{  
  "data": {  
    "Movie": [  
      {  
        "title": "River Runs Through It, A",  
        "actors": [  
          {  
            "name": " Tom Skerritt"  
          },  
          {  
            "name": " Brad Pitt"  
          }  
        ],  
        "genres": [  
          {  
            "name": "Drama"  
          }  
        ],  
        "directors": {  
          "name": "Robert Redford",  
          "movies": [  
            {  
              "title": "Conspirator, The"  
            },  
            {  
              "title": "Lions For Lambs"  
            },  
            {  
              "title": "Company You Keep, The"  
            }  
          ]  
        }  
      }  
    ]  
  }  
}
```

GraphQL response

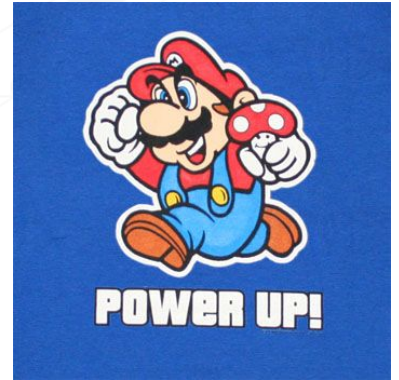
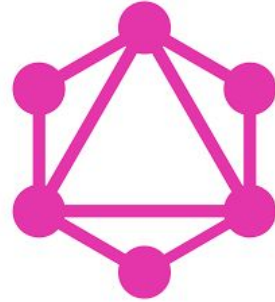
What is GraphQL?



- 1) GraphQL is an **API query language**, not a database query language.
- 2) Limited expressivity (**no projections, no aggregations, etc**).
- 3) While GraphQL exposes your application data as a graph, it's **not (just) for graph databases**

GraphQL Advantages

- **Overfetching**
 - Less data over the wire
- **Underfetching**
 - Single round trip
- GraphQL **Specification**
- **“Graphs All The Way Down”**
 - **Relationships** vs Resources
 - Unify disparate systems (microservices)
- **Simplify data fetching**
 - Component based data interactions



GraphQL Challenges

- Some well understood practices from REST don't apply
 - HTTP status codes
 - Errors
 - Caching
- Exposing arbitrary complexity to client
 - Performance considerations
- n+1 query problem
- Query costing / rate limiting



REST In Practice - R.I.P.?





How To Build A GraphQL Service

Start With A GraphQL Schema

```
type CommunityBlog {  
  title: String  
  url: String  
  author: DiscourseUser  
}  
  
type DiscourseUser {  
  name: String  
  screenName: String  
  avatar: String  
}  
  
type Query {  
  topCommunityBlogsAndContent(first: Int = 10): [CommunityBlog]  
}
```

GraphQL Resolvers

- Functions that define how to “resolve” data for GraphQL request

```
const resolvers = {  
  Query: {  
    topCommunityBlogsAndContent: (object, params, context) => {  
      // TODO: check auth headers from context  
      // TODO: query the database  
      // TODO: validate / format response  
      // TODO: return results  
    }  
  }  
};
```

Resolvers

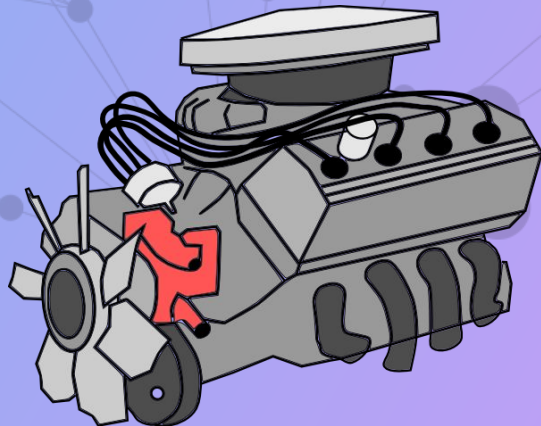
```
const resolvers = {
  Query: {
    topCommunityBlogsAndContent: (_, params, context) => {
      let session = context.driver.session();
      const baseUrl = 'https://community.neo4j.com/';
      // cypherQuery is defined above
      return session.run(cypherQuery, params)
        .then( result => {
          const resData = result.records.map(record => {
            const user = record.get("u").properties,
                  topic = record.get("topic").properties;
            return {
              title: topic.title,
              url: baseUrl + "t/" + topic.slug,
              author: {
                name: user.name,
                screenName: user.screenName,
                avatar: getAvatarUrl(user.avatarTemplate)
              }
            }
          })
          return resData;
        })
        .catch(error => {
          console.log(error);
        })
        .finally( ()=> {
          session.close();
        })
    }
  }
};
```

Common Problems With This Approach



- 1) Schema Duplication
- 2) Mapping / translation layer from graph \longleftrightarrow (???)
- 3) Boilerplate code
- 4) n+1 query problem

GraphQL "Engines"



GraphQL "Engines" Overview

- Tools for **auto-generating GraphQL schema**, **generating database queries** from GraphQL requests



Prisma

prisma.io

@lyonwj

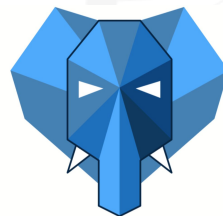


AWS AppSync

aws.amazon.com/appsync



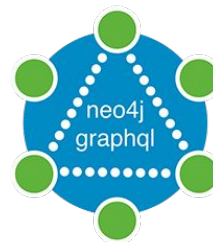
hasura.io



PostGraphile

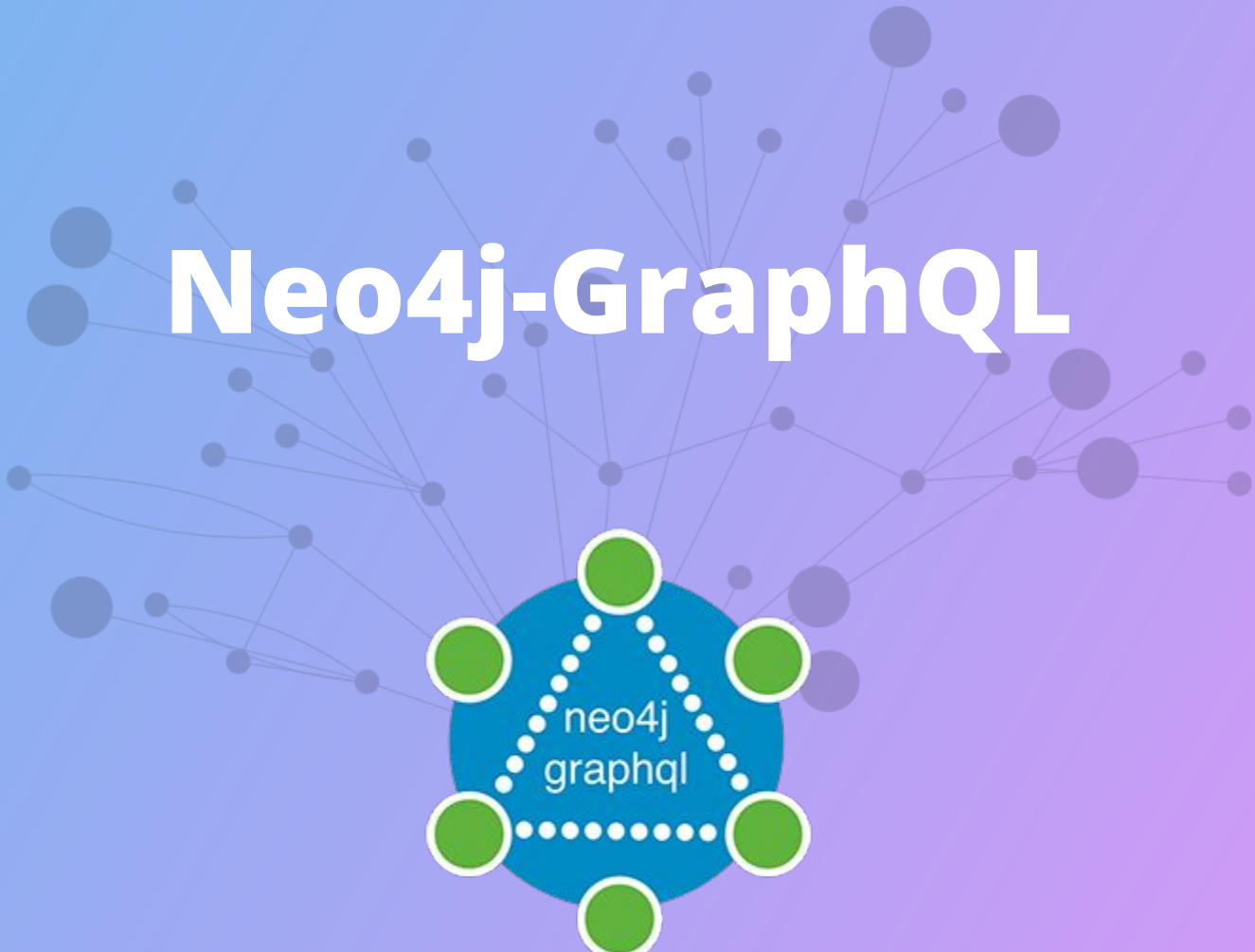
graphile.org

Neo4j-GraphQL



grandstack.io

Neo4j-GraphQL



Goals for Neo4j-GraphQL Integration

- **GraphQL First Development**
 - GraphQL schema drives the database data model
- **Generate Cypher** from GraphQL
 - Single query / **single round trip** to database
- **Generate GraphQL CRUD API from type definitions**
- **Auto-generated resolvers** (no boilerplate!)
- Extend GraphQL functionality with Cypher
 - **@cypher** schema directive

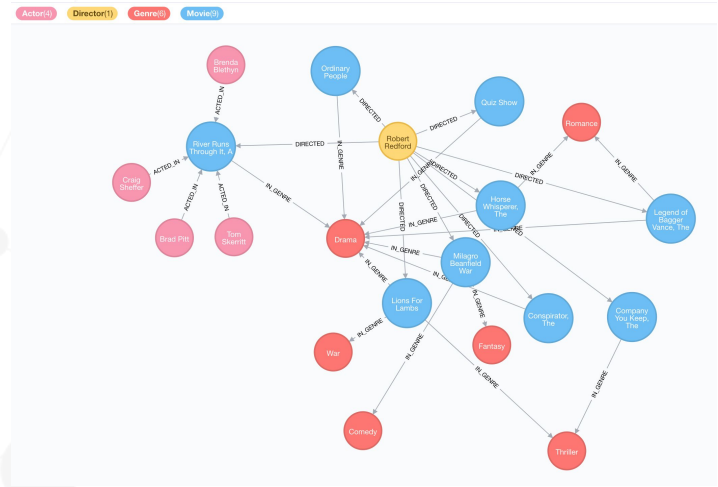
GraphQL First Development

```
type Movie {
  title: String
  year: Int
  plot: String
  poster: String
  imdbRating: Float
  genres: [Genre]
}

type Genre {
  name: String
  movies: [Movie]
}

type Actor {
  name: String
  movies: [Movie]
}

type Director {
  id: ID!
  name: String
}
```



Auto-generate GraphQL CRUD API

Query & Mutation types

- Entry point for each type

Ordering

Pagination

Complex Filter

DateTime types

```
1 {
2   Movie(
3     filter: {
4       AND: [
5         { plot_contains: "zombie" }
6         { genres_some: { name: "Comedy" } }
7         { year_gt: 1994 }
8       ]
9       OR: [
10        { actors_some: { name: "Jesse Eisenberg" } }
11        { genres_none: { name: "Animation" } }
12      ]
13    ]
14  }
15 }
16 orderBy: [year_desc, title_asc]
17 ) {
18   title
19   plot
20   year
21 }
22 }
```

```
{
  "data": {
    "Movie": [
      {
        "title": "Pride and Prejudice and Zombies",
        "plot": "Five sisters in 19th century England must cope with the pressures to marry while protecting themselves from a growing population of zombies.",
        "year": 2016
      },
      {
        "title": "Warm Bodies",
        "plot": "After a highly unusual zombie saves a still-living girl from an attack, the two form a relationship that sets in motion events that might transform the entire lifeless world.",
        "year": 2013
      },
      {
        "title": "Dead Snow (Død snø)",
        "plot": "A ski vacation turns horrific for a group of medical students, as they find themselves confronted by an
```

Generate Cypher From GraphQL

```
{  
  Movie(title: "River Runs Through It, A") {  
    title  
    actors(first: 2) {  
      name  
    }  
    genres {  
      name  
    }  
    directors {  
      name  
      movies(first: 3) {  
        title  
      }  
    }  
  }  
}
```



- 1 **MATCH** (a:Actor)-[:ACTED_IN]->(m:Movie)-[:IN_GENRE]->(g:Genre)
- 2 **WHERE** m.title **CONTAINS** "River Runs Through It"
- 3 **MATCH** (m)<-[:DIRECTED]-(d:Director)-[:DIRECTED]->(o:Movie)
- 4 **RETURN** *

Extend GraphQL w/ Cypher

@cypher GraphQL schema directive

```
type Query {  
  sessionsBySubstring(string: String): [Session] @cypher(  
    statement: """MATCH (s:Session)  
      WHERE toLower(s.description) CONTAINS toLower($string)  
      OR toLower(s.name) CONTAINS toLower($string)  
      RETURN s;""")  
}
```

grandstack.io/docs/neo4j-graphql-js.html#cypher-directive

neo4j-graphql-js

2.7.2 • Public • Published a month ago

Readme

6 Dependencies

3 Dependents

60 Versions

circleci passing codecov 96% npm package 2.7.2 Docs GRANDstack.io

neo4j-graphql.js

A GraphQL to Cypher query execution layer for Neo4j and JavaScript GraphQL implementations.

- [Read the docs](#)
- [Read the changelog](#)

neo4j-graphql-js is in active development. There are rough edges and APIs may change. Please file issues for any bugs that you find or feature requests.

Installation and usage

Install

```
npm install --save neo4j-graphql-js
```

install

```
> npm i neo4j-graphql-js
```

↓ weekly downloads

1,420



version

2.7.2

license

Apache-2.0

open issues

115

pull requests

6

homepage

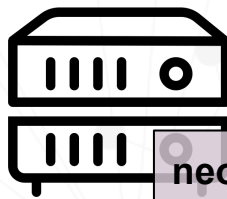
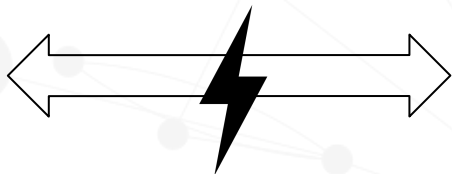
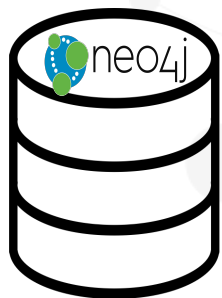
github.com

repository

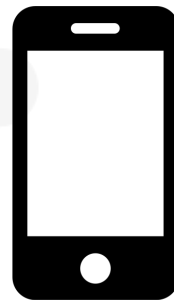
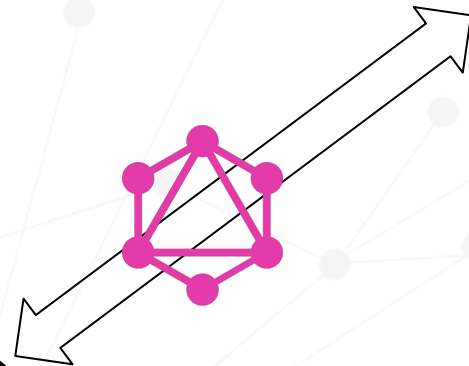
github

last publish

neo4j-graphql-js



neo4j-graphql-js
apollo-server



Demo

GRANDstack starter project

grandstack.io

```
type User {
  id: ID!
  name: String
  friends: [User] @relation(name: "FRIENDS", direction: "BOTH")
  reviews: [Review] @relation(name: "WROTE", direction: "OUT")
  avgStars: Float @cypher(statement: "MATCH (this)-[:WROTE]->(r:Review) RETURN toFloat(avg(r.stars))")
  numReviews: Int @cypher(statement: "MATCH (this)-[:WROTE]->(r:Review) RETURN COUNT(r)")
}

type Business {
  id: ID!
  name: String
  address: String
  city: String
  state: String
  reviews: [Review] @relation(name: "REVIEWS", direction: "IN")
  categories: [Category] @relation(name: "IN_CATEGORY", direction: "OUT")
}

type Review {
  id: ID!
  stars: Int
  text: String
  business: Business @relation(name: "REVIEWS", direction: "OUT")
  user: User @relation(name: "WROTE", direction: "IN")
}

type Category {
  name: ID!
  businesses: [Business] @relation(name: "IN_CATEGORY", direction: "IN")
}
```

GRANDstack.io

GRANDstack Starter · GRAND: x

Secure <https://grandstack.io/docs/getting-started-grand-stack-starter.html>

GRANDstack Docs Blog Search

Getting Started

- What is GRANDstack
- GRANDstack Starter**
- Neo4j-GraphQL Overview

neo4j-graphql-js

- Getting Started
- API Reference

Guides

- Auth / Middleware
- Neo4j Database Plugin

Resources

- Links

GRANDstack Starter

The easiest way to get started with GRANDstack is by using the GRANDstack starter. This is template project with a React app that uses Apollo Client to query a GraphQL API backed by Neo4j.

Download
Usage

Hands On With The GRANDstack Starter Project

Hands On With The GRANDstack Starter Project

William Lyon
@lyonwj
lyonwj.com

GraphQL React Apollo Neo4j Database

GRANDstack.io

Download

Clone the Github repo here:

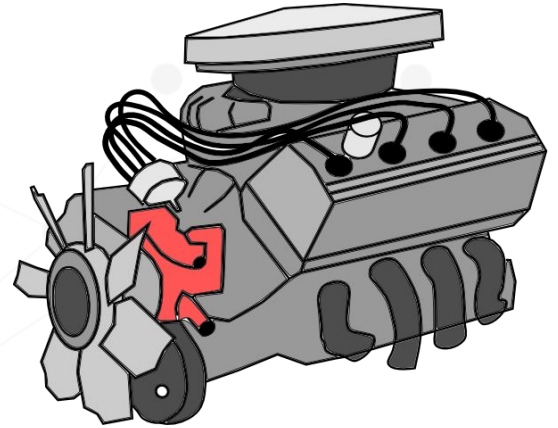
```
git clone https://github.com/grand-stack/grand-stack-starter.git
```

or download the latest release here.

3

GraphQL "Engines"

- **Declarative database integrations** for GraphQL
- **GraphQL type definitions** define database model
- Provision CRUD GraphQL API
 - **Auto-generated GraphQL API**
 - Schema enrichments
- Generate database queries
 - **auto-generated resolvers**
 - reduce boilerplate



How Do GraphQL Engines Generate Database Queries From GraphQL Requests?

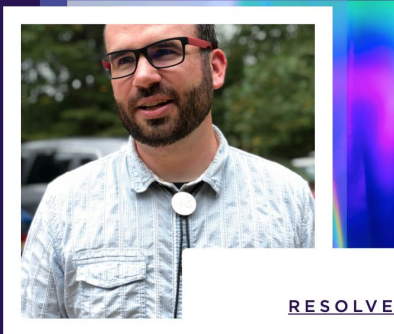
`resolveInfo` resolver argument

- GraphQL query AST
- GraphQL schema
- Selection set
- Variables
- ...

```
1 const Resolvers = {
2   Query: {
3     searchMovies: (object, params, context, resolveInfo) => {
4       // Resolver logic here
5     }
6   }
7 }
```

```
1 export type GraphQLResolveInfo = {
2   fieldName: string;
3   fieldNodes: Array<FieldNode>;
4   returnType: GraphQLOutputType;
5   parentType: GraphQLCompositeType;
6   path: ResponsePath;
7   schema: GraphQLSchema;
8   fragments: { [fragmentName: string]: FragmentDefinitionNode };
9   rootValue: mixed;
10  operation: OperationDefinitionNode;
11  variableValues: { [variableName: string]: mixed };
12 };
```

Use code
“**SPEAKERFRIEND**”
for a 15%
discount!!!



WILLIAM LYON

Software Engineer at Neo4j



RESOLVE INFO DEEP DIVE

Each GraphQL resolver is passed an object called the resolve info argument. This object contains information about the schema and query and can be used to construct database queries at runtime inside the resolver.

GRAPHQL SUMMIT

October 30-31

AT THE HYATT REGENCY EMBARCADERO,
SAN FRANCISCO

Join over 1,200 developers at the world's largest
conference dedicated to GraphQL.

[Register Now](#)



summit.graphql.com



github.com/johnymontana/NODES2019-GRANDstack

GRANDstack Starter Project

grand-stack / grand-stack-starter Template

Unwatch 17 Unstar 206 Fork 78

Code Issues 12 Pull requests 3 Projects 0 Wiki Security Insights Settings

Simple starter project for GRANDstack full stack apps <https://grandstack.io/docs/getting-st...> Edit

graphql getting-started starter-project neo4j grandstack apollo react api Manage topics

91 commits 6 branches 8 releases 10 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Use this template Clone or download

johnymontana npm update Latest commit a5f0c49 on Aug 22

api	npm update	2 months ago
neo4j	reset back to defaults, and added a note how to start docker-compose ...	last year
ui	npm update	2 months ago
.gitignore	Add npm script to transpile api code	9 months ago
LICENSE.txt	Added License & Copyright, fixed graphql dependency	last year
README.md	Update README.md	5 months ago
app.json	add app.json for Google Cloud Run	2 months ago
docker-compose.yml	added support for docker-compose	last year
now.json	updates for now v2 - monorepo	6 months ago

github.com/grand-stack/grand-stack-starter

GRANDstack Starter Project

johnnymontana / **NODES2019-GRANDstack**
generated from [grand-stack/grand-stack-starter](#)

Unwatch ▾

1

★ Star

0

Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

Generating your repository...

It should only take a few seconds.

Refresh



github.com/grand-stack/grand-stack-starter

Who's using GRANDstack & Neo4j GraphQL?



FINANCIAL
TIMES



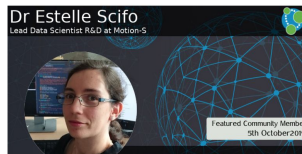
HUMAN
CONNECTION.ORG



Who's using GRANDstack & Neo4j GraphQL?



Featured Community Member



This Week in Neo4j

5th Oct 2019

- NODES 2019 Preview: Everything Else
- Aleph Data Processing Toolkit, String to Date with APOC, ETL ready for Cloud
- Learn Japanese characters using Neo4j
- Building a Chat Bot in Neo4j: Part 2
- Streams and Tables and Graphs, oh my!

[See More >](#)

Popular Community Content

- When graph modeling, when to use a property, a label, and a separate node?
- Building A Photo-Based Personalized Recommendations Application
- Custom analyzer for fulltext search in Neo4j
- Scale up your D3 graph visualisation - WebGL & Canvas with PIXI.js

Popular Community Projects

- Halin v0.11 is now available! (The Trivium release)
- Football exploration with Neo4r
- Codex: an atlas of relations
- neode JavaScript

New Certified Developers!

- Caojingwei
- Tomtrezb2003
- Aishwarya Bhand
- Aishwarya Bhand
- Azocankara

community.neo4j.com

Who's using GRANDstack & Neo4j GraphQL?

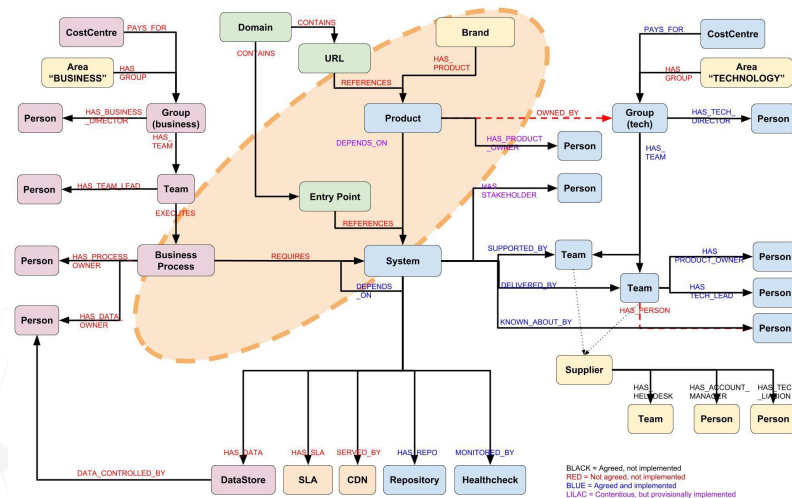


FINANCIAL
TIMES



A slide from Rhys Evans's talk "A Field Guide To The Financial Times"

speakerdeck.com/wheresrhys/a-field-guide-to-the-financial-times



medium.com/ft-product-technology/ready-steady-crash-eeae63e89a77

Who's using GRANDstack & Neo4j GraphQL?



HUMAN
CONNECTION.ORG

[Online Meetup] Using Neo4j and GraphQL to build Human Connection

Community Content & Blogs



roschaefer

1  Jun 20

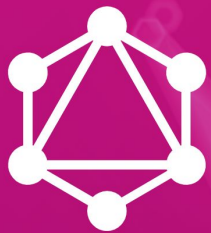
Using Neo4j and GraphQL to build Human Connection

Human Connection is a nonprofit social, action and knowledge network that connects information to action and promotes positive local and global change in all areas of life.

In this session for [Neo4j Online Meetup](#), Greg and Robert will show us how they used Neo4j and GraphQL as part of their technology stack.



community.neo4j.com/t/online-meetup-using-neo4j-and-graphql-to-build-human-connection/8213



GraphQL Foundation

An open and neutral home for the GraphQL community to enable widespread adoption and help accelerate development and the surrounding ecosystem

Neo4j Joins GraphQL Foundation as a Founding Member



William Lyon, Developer Relations Engineer

Mar 12 · 4 mins read

“Neo4j is pleased to support GraphQL, due to its unique ability to enable front-end developers, API developers and data architects to all work with the same graph-based data model. Our GraphQL integration and GRANDstack have already helped thousands of developers build graph applications with Neo4j, thanks to the huge developer productivity gain of GraphQL and the graph thinking mindset it brings for both developers and API consumers. The GraphQL Foundation is an important step to cement today’s most relevant standard for building APIs and we’re honored to join as founding members to help steward GraphQL as the ecosystem continues to evolve.”

— Emil Eifrem, CEO and Co-Founder, Neo4j

foundation.graphql.org

GRANDstack Talks In #NODES2019

Welcome To NODES GRANDstack

NODES 2019 Session List

Search Session Description:

Image	Name	Presenter	Description	Recommended
	Discovering the Soul of a Product With Neo4j	Pat Brown	At Under Armour many disjointed systems are involved in bringing a product to market. Customer service reps and data scientists don't always have access to the data they need. We set out to capture that data from each system, expose it with the GRANDstack and make it available to our teammates.	Uncovering Medical Fraud in New York City's most Vulnerable Neighborhoods
	How the GRANDstack Makes Handling Complex Data Easy	Michael Porter	Using neo4j along with the GRANDstack has allowed me to build an application that deals with highly connected data in a manner I don't think would be possible with a SQL data base. The neo4j-graphql-js integration makes turning your Neo4j instance into a modern React application a snap.	Exploring the U.S. National Bridge Index
	Modeling a Tournament in Neo4j	Michael McKenzie	Scheduling and organizing a martial arts tournament can be a challenging task due to the variety of ages, ranks, division, competitors, and judges that makeup the tournament. We'll explore an approach to develop a tournament application with Neo4j and built on the GRANDstack.	Exploring the U.S. National Bridge Index
	Coding a graph application from scratch with GRANDstack	Christian Miles	The GRANDstack is a rich ecosystem of tools that developers use to rapidly build and prototype graph applications. Coding from scratch, Christian shows how easy it is to create a graph web application for any use case.	Graph Visualization: Main aspects, challenges and solutions

Subject: **grandstack-graphql** [Clear Filter]

Thursday, October 10

- 11:30am **Building a full-stack fraud-detection Solution with Kafka, GraphQL and Neo4j-Graph-Algorithms**
- GRANDstack: Graphs ALL the Way Down**
- 12:35pm **Exploring the U.S. National Bridge Index**
- 1:00pm **Discovering the Soul of a Product With Neo4j**
- 1:20pm **How the GRANDstack Makes Handling Complex Data Easy**
- 1:45pm **Modeling a Tournament in Neo4j**
- 3:35pm **Coding a graph application from scratch with GRANDstack**

nodes2019-app.grandstack.io

neo4j.com/online-summit

Use code
“**SPEAKERFRIEND**”
for a 15%
discount!!!



WILLIAM LYON

Software Engineer at Neo4j



RESOLVE INFO DEEP DIVE

Each GraphQL resolver is passed an object called the resolve info argument. This object contains information about the schema and query and can be used to construct database queries at runtime inside the resolver.

summit.graphql.com

GRANDstack.io/survey

GRANDstack Survey

Help drive the direction of GRANDstack by giving us your feedback in this short form.

GRANDstack Survey

Please take a few minutes to let us know how you are using the Neo4j GraphQL integrations and GRANDstack and what is and is not working for you. Your feedback will help drive future features and efforts. Thanks!

* Required

Have you used any of the Neo4j GraphQL integrations / GRANDstack? *

- Yes, using in production
- Yes, in development
- Yes, just playing around
- No
- Other: _____

blog.grandstack.io

Country	League	Country	Amount Spent	Amount Received	Profit
	Premier League	England	£1,351,356,998	£699,286,998	-£652,070,000
	LaLiga	Spain	£1,152,299,998	£876,670,000	-£275,429,998
	Serie A	Italy	£976,011,998	£756,994,997	-£219,017,001
	Bundesliga	Germany	£632,024,999	£484,561,000	-£147,463,999
	Ligue 1	France	£500,495,000	£707,010,000	£206,515,000
	Premier Liga	Russia	£172,471,000	£40,228,000	-£132,243,000
	Championship	England	£141,935,999	£276,221,999	£134,286,000
	Liga NOS	Portugal	£122,950,000	£309,834,999	£186,884,999
	Jupiler Pro League	Belgium	£112,369,999	£196,141,999	£83,772,000

Football Transfers Graph App with the GRANDstack Starter Kit

Building a Neo4j and React App for Football Transfers using the GRANDstack

Mark Needham
Aug 29 · 6 min read

```
query {
  tag(name: "football") {
    name
    questions(first: 3) {
      title
      tagged { name }
    }
  }
}
```

```
mutation {
  createTag(name: "football") {
    name
  }
  deleteTag(name: "football") {
    name
  }
}
```

What's New In GRANDstack? June 2019 Edition

Complex filtering, schema inference, Google Cloud Run, GRANDstack survey, updated starter project, and more!

William Lyon
Jun 29 · 5 min read

nodeType	nodeLabels	propertyName	propertyTypes	mandatory
":Genre"	["Genre"]	"name"	["String"]	true
":User"	["User"]	"name"	["String"]	true
":User"	["User"]	"userId"	["String"]	true
":Actor"	["Actor"]	"name"	["String"]	true
":Director"	["Director"]	"name"	["String"]	true
":Movie"	["Movie"]	"movieId"	["String"]	true
":Movie"	["Movie"]	"imdbid"	["String"]	true
":Movie"	["Movie"]	"title"	["String"]	true
":Movie"	["Movie"]	"imdbid"	["String"]	false
":Movie"	["Movie"]	"year"	["Long"]	false
":Movie"	["Movie"]	"countries"	["StringArray"]	false
":Movie"	["Movie"]	"isNewspaper"	["StringArray"]	false

Kenny Bastani and 10 others

Inferring GraphQL Type Definitions From An Existing Neo4j Database

Create a GraphQL API Without Writing Resolvers Or TypeDefs

William Lyon
May 1 · 4 min read

GRANDstack

Build full stack graph applications with ease.



DOWNLOAD STARTER

LEARN MORE



GraphQL



React



Apollo



Neo4j Database

[GRANDstack.io](https://grandstack.io)

Hunger Games Questions for "GRANDstack: Graphs ALL The Way Down"



Query the GraphQL Playground at nodes2019.grandstack.io to answer:

1. What is the title of the first session when ordered by title in ascending order?
 - a. APOC Pearls: The Best Tips & Tricks
 - b. Agile Property Graph Design Process
 - c. Aardvarks In The Cloud With Neo4j Fabric
2. Which speakers are presenting talks about Kafka?
 - a. Bob Loblaw and Princess Caroline
 - b. Mark Needham and Andrea Santurbano
 - c. Michael Hunger and Craig Taverner
3. If I like the talk about Neovis.js, what is the top recommendation for me?
 - a. Graph Visualization: Main aspects, challenges and solutions
 - b. Building Spatial Search Algorithms for Neo4j
 - c. Challenges in knowledge graph visualization

[Answer here: r.neo4j.com/hunger-games](https://r.neo4j.com/hunger-games)