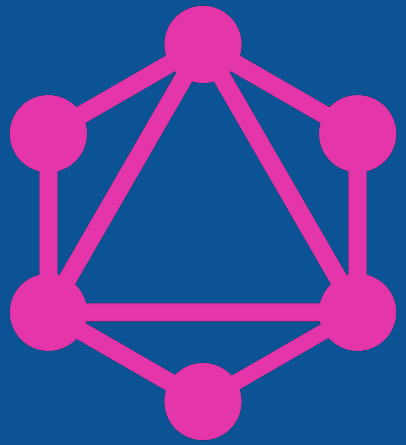


GraphQL 101:

What, Why, How

Maria Comas · @una_maria
Web Developer at Amazee Labs

https://2018.stateofjs.com/data-layer/overview

WHAT

2012

2015

2018

[...] help GraphQL become a reliable industry standard

https://gql.foundation

GraphQL

June 2018 Edition

Introduction

This is the specification for GraphQL, a query language and execution engine originally created at Facebook in 2012 for describing the capabilities and requirements of data models for client–server applications. The development of this open standard started in 2015.

GraphQL has evolved and may continue to evolve in future editions of this specification. Previous editions of the GraphQL specification can be found at permalinks that match their release tag. The latest working draft release can be found at facebook.github.io/graphql/draft/.

Copyright notice

Copyright © 2015–present, Facebook, Inc.

As of September 26, 2017, the following persons or entities have made this Specification available under the Open Web Foundation Final Specification Agreement (OWFa 1.0), which is available at openwebfoundation.org.

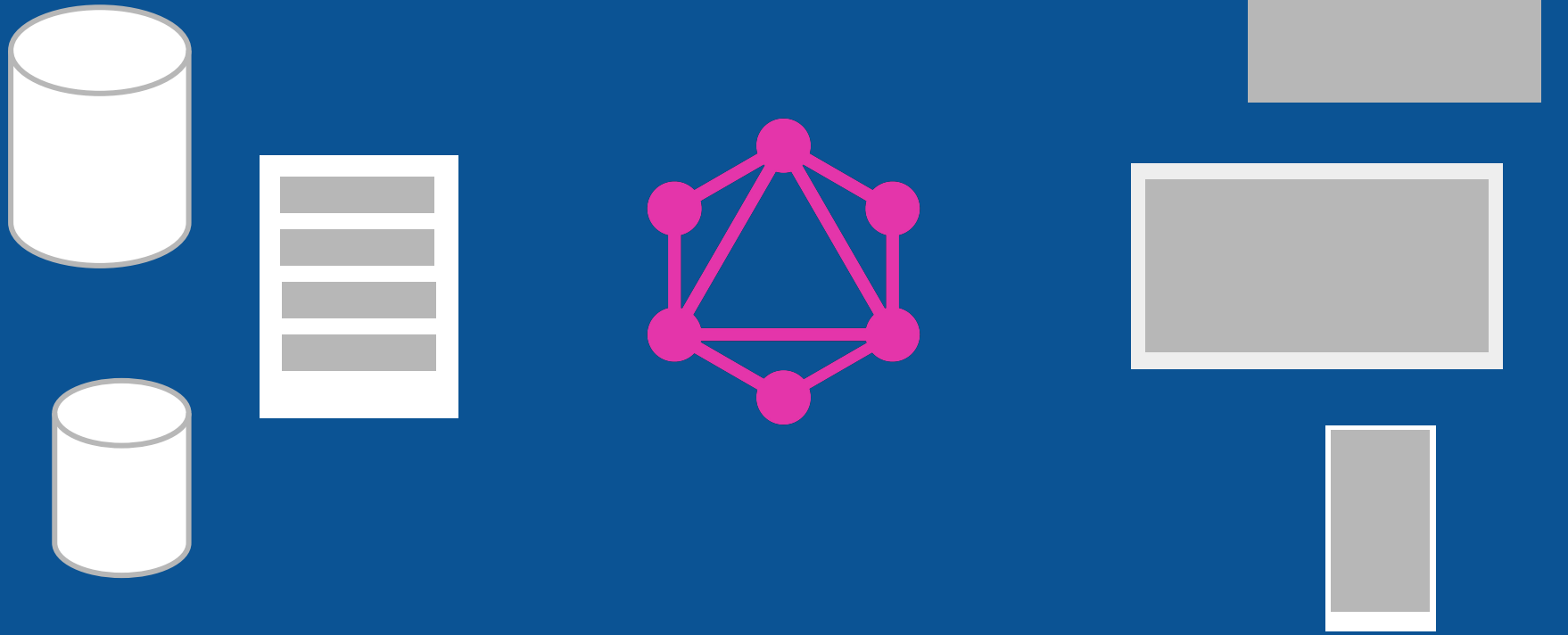
- Facebook, Inc.

You can review the signed copies of the Open Web Foundation Final Specification Agreement Version 1.0 for this specification at github.com/facebook/graphql, which may also include additional parties to those listed above.

Your use of this Specification may be subject to other third party rights. THIS SPECIFICATION IS PROVIDED "AS IS." The contributors expressly disclaim any warranties (express, implied, or

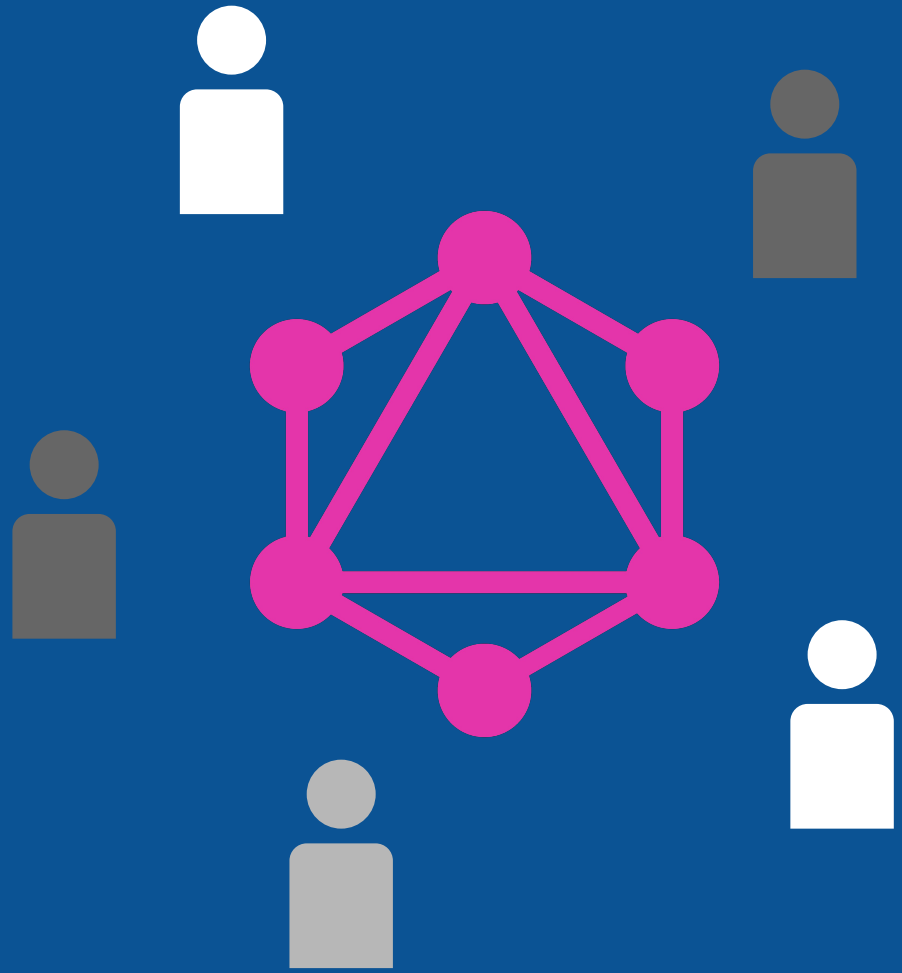
GraphQL

1 Overview
2 Language
3 Type System
4 Introspection
5 Validation
6 Execution
7 Response
A Appendix: Notation Conventions
B Appendix: Grammar Summary
§ Index



> I think GraphQL wins my heart because **it changes human behavior**.

Garrett Heinlen: https://www.youtube.com/watch?v=e6ziHGPkaKM



WHY

> We needed a data-fetching API **powerful** enough to describe all of Facebook, yet simple enough to be **easy to learn and use** by our product developers.

https://code.fb.com/core-data/graphql-a-data-query-language/

WYGISWAF

(What You Get Is What You Asked For)

Lea Verou

LeaVerou

Unfollow

Block or report user

Developer Program Member

PRO

MIT

Boston, MA

http://lea.verou.me

Organizations

Overview | Repositories 61 | Projects 0 | Stars 84 | Followers 11.8k | Following 30

Pinned repositories

PrismJS/prism
Lightweight, robust, elegant syntax highlighting.
JavaScript ★ 6.4k 766

awesomplete
Ultra lightweight, usable, beautiful autocomplete with zero dependencies.
JavaScript ★ 6.4k 606

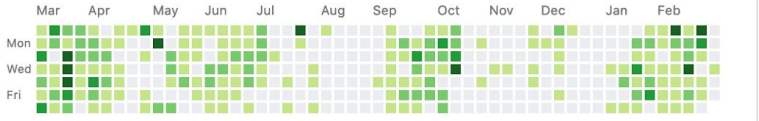
prefixfree
Break free from CSS prefix hell!
JavaScript ★ 3.7k 744

animatable
One property, two values, endless possiblities
HTML ★ 2.5k 673

bliss
Blissful JavaScript
JavaScript ★ 2.1k 105

mavoweb/mavo
Create web applications entirely by writing HTML and CSS!
JavaScript ★ 2.1k 119

1,213 contributions in the last year



2019
2018
2017

Lea Verou

LeaVerou

MIT

Boston, MA

http://lea.verou.me

```
{
  user(login: "leaverou") {
    avatarUrl
    name
    login
    company
    location
    websiteUrl
  }
}
```

```
{
  "data": {
    "user": {
      "avatarUrl": "https://avatars1.githubusercontent.com/u/175836?v=4",
      "name": "Lea Verou",
      "login": "LeaVerou",
      "company": "MIT",
      "location": "Boston, MA",
      "websiteUrl": "http://lea.verou.me"
    }
  }
}
```

Lea Verou
LeaVerou

Overview | Repositories 61 | Projects 0 | Stars 84 | Followers 11.8k | **Following 30**

George Kalpakas gkalpak
Software engineer / Angular team
Athens, Greece

Follow

Jason Pamental jpamental
RWT.io Rumford, RI

Follow

```
{
  user(login: "leaverou") {
    avatarUrl
    name
    login
    company
    location
    websiteUrl
  }
}
```

```
{
  user(login: "leaverou") {
    avatarUrl
    name
    login
    company
    location
    websiteUrl
    following(first: 1) {
      nodes {
        avatarUrl
        name
        login
        company
        location
      }
    }
  }
}
```

```
{
  "data": {
    "user": {
      "avatarUrl": "https://avatars1.githubusercontent.com/u/175836?v=4",
      "name": "Lea Verou",
      "login": "LeaVerou",
      "company": "MIT",
      "location": "Boston, MA",
      "websiteUrl": "http://lea.verou.me",
      "following": {
        "nodes": [
          {
            "avatarUrl": "https://avatars2.githubusercontent.com
/u/8604205?v=4",
            "name": "George Kalpakas",
            "login": "gkalpak",
            "company": "",
            "location": "Athens, Greece"
          }
        ]
      }
    }
  }
}
```

Know thyself *

* https://en.wikipedia.org/wiki/Know_thyself

GraphiQL
Prettify
History
{
user(login: "leaverou") {
__typename
name
following(first: 1) {
nodes {
__typename
name
}
}
}
}
avatarUrl
bio
bioHTML
commitComments
company
companyHTML
contributionsCollection
createdAt
{
"data": {
"user": {
"__typename": "User",
"name": "Lea Verou",
"following": {
"nodes": [
{
"__typename": "User",
"name": "George Kalpakas"
}
]
}
}
}
}
Schema
User
FIELDS
avatarUrl(size: Int): URI!
A URL pointing to the user's public avatar.
bio: String
The user's public profile bio.
bioHTML: HTML!
The user's public profile bio as HTML.
commitComments(
after: String
before: String
first: Int
last: Int
): CommitCommentConnection!
A list of commit comments made by this user.

HOW

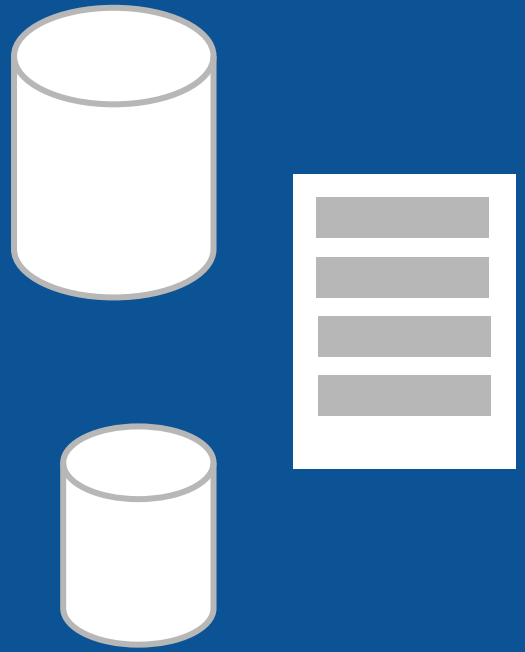
```
query {
  user(id: "abc") {
    id
    name
  }
}
```

https://www.prisma.io/blog/graphql-server-basics-the-schema-ac5e2950214e

```
type Query {
  user(id: ID!): User
}
```

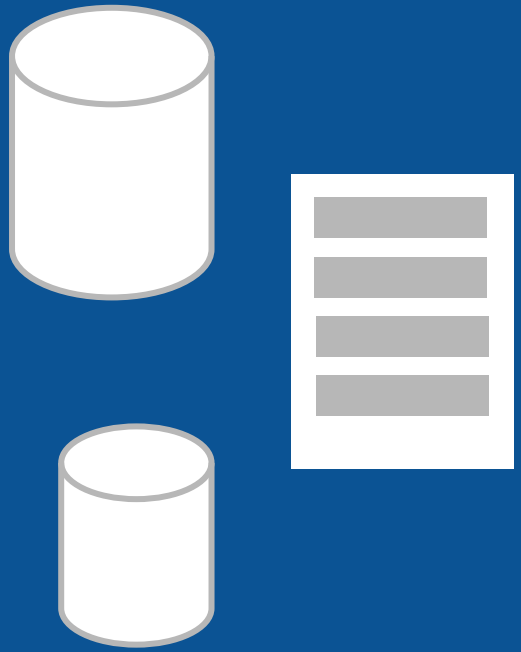
```
type User {
  id: ID!
  name: String
}
```

https://www.prisma.io/blog/graphql-server-basics-the-schema-ac5e2950214e



```
const schema = new GraphQLSchema({
  query: new GraphQLObjectType({
    name: 'Query',
    fields: {
      user: {
        type: UserType,
        args: {
          id: { type: GraphQLID }
        },
        resolve: (root, args, context, info) => {
          const { id } = args
          return fetchUserById(id)
        }
      }
    }
  })
})
```

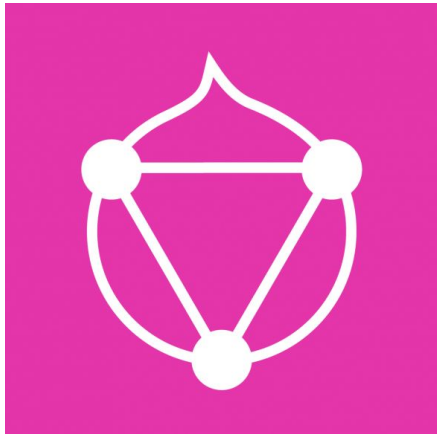
https://www.prisma.io/blog/graphql-server-basics-the-schema-ac5e2950214e



```
const UserType = new GraphQLObjectType({
  name: 'User',
  fields: {
    id: {
      type: GraphQLID,
      resolve: (root, args, context, info) => {
        return root.id
      }
    },
    name: {
      type: GraphQLString,
      resolve: (root, args, context, info) => {
        return root.name
      }
    }
  }
})
```

https://www.prisma.io/blog/graphql-server-basics-the-schema-ac5e2950214e

Drupal



dgo.to/graphql

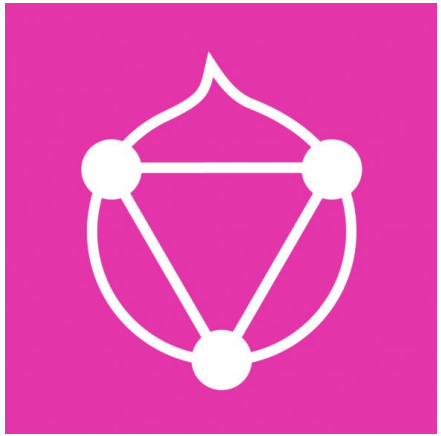
Downloads

8.x-3.0-rc2 released 20 September 2018
✓ Recommended by the project's maintainer.
⬇ tar.gz (1.16 MB) | zip (1.41 MB)

Development version: **8.x-3.x-dev** updated 15 Feb 2019 at 09:43 UTC

Development version: **8.x-4.x-dev** updated 13 Feb 2019 at 21:03 UTC

https://github.com/drupal-graphql/graphql/blob/8.x-3.x/doc/SUMMARY.md

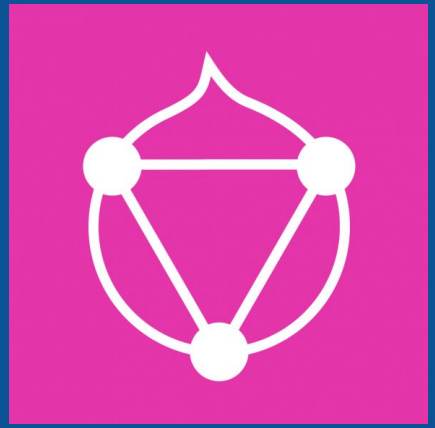


dgo.to/graphql_twig

```
{#graphql
query {
  admin:userById(id: "1") {
    uid
    name
  }
  user:currentUserContext {
    uid
  }
}
#}
```

```
{% set admin = graphql.data.admin %}
{% set user = graphql.data.user %}
```

https://www.amazeelabs.com/en/blog/atomic-design-drupal-graphql-twig-webinar-recap



drupal.slack.com

#graphql

MOAR

www.howtographql.com

principledgraphql.com

developer.github.com/v4/explorer

www.robinwieruch.de/the-road-to-graphql-book

The most damaging phrase in the language is “We’ve always done it this way!”

Grace Hopper: https://quoteinvestigator.com/2014/11/27/always-done/

Thank you!

Maria Comas · @una_maria
Web Developer at Amazee Labs