Features
Business
Explore
Marketplace
Pricing
Search
/
Sign in or **Sign up**

graphql / **graphql-js**

◎ Watch | 366   ★ Star | 10,881   ⑂ Fork | 880

<> Code   ⊙ Issues **62**   ⑊ Pull requests **25**   ▥ Projects **0**   ▤ Wiki   �📊 Insights

A reference implementation of GraphQL for JavaScript   http://graphql.org/graphql-js/

graphql   graphql-js

🕐 **1,649** commits	⑊ **5** branches	🏷 **84** releases	👥 **127** contributors	⚖ MIT

Branch: **master** ▾   New pull request   **Find file**

Clone or download ▾

Cito and **leebyron** Document the contextValue argument (#1400)   ⋯
Latest commit c165907 Jun 28, 2018

📁 .github	ISSUE_TEMPLATE: Fix 'facebook/graphql' link (#1243)	Feb 14, 2018
📁 resources	Babel 7 preparation: Switch babel config to JS (#1348)	Jun 1, 2018
📁 src	Document the contextValue argument (#1400)	Jun 29, 2018
📄 .babelrc.js	Upgrade babel and associated plugins to v7 (#1350)	Jun 2, 2018
📄 .eslintrc	No more max-len, in prettier we trust	Dec 11, 2017
📄 .flowconfig	Update Flow to 0.75 + other deps (#1402)	Jun 29, 2018
📄 .gitignore	Unify build scripts to all rely on `npm run build` (#1161)	Dec 17, 2017
📄 .npmignore	Coverage: Switch isparta to nyc (#1342)	May 11, 2018
📄 .nycrc	Coverage: Switch isparta to nyc (#1342)	May 11, 2018
📄 .prettierrc	Prettier (#1112)	Dec 4, 2017
📄 .travis.yml	Run tests on node 10. Drop support for node 4.0 (#1338)	May 24, 2018
📄 LICENSE	Relicense as MIT (#1046)	Sep 26, 2017
📄 README.md	Notes about browser use	Apr 18, 2018
📄 package.json	Update Flow to 0.75 + other deps (#1402)	Jun 29, 2018
📄 yarn.lock	Update Flow to 0.75 + other deps (#1402)	Jun 29, 2018

GraphQL.js

The JavaScript reference implementation for GraphQL, a query language for APIs created by Facebook.

`npm package` `0.13.2` `build` `failing` `coverage` `99%`

See more complete documentation at http://graphql.org/ and http://graphql.org/graphql-js/.

Looking for help? Find resources from the community.

Getting Started

An overview of GraphQL in general is available in the README for the Specification for GraphQL. That overview describes a simple set of GraphQL examples that exist as tests in this repository. A good way to get started with this repository is to walk through that README and the corresponding tests in parallel.

Using GraphQL.js

Install GraphQL.js from npm

With yarn:

```
yarn add graphql
```

or alternatively using npm:

```
npm install --save graphql
```

GraphQL.js provides two important capabilities: building a type schema, and serving queries against that type schema.

First, build a GraphQL type schema which maps to your code base.

```
import {
  graphql,
  GraphQLSchema,
  GraphQLObjectType,
  GraphQLString
} from 'graphql';

var schema = new GraphQLSchema({
  query: new GraphQLObjectType({
    name: 'RootQueryType',
    fields: {
      hello: {
        type: GraphQLString,
        resolve() {
          return 'world';
        }
      }
    }
  })
});
```

This defines a simple schema with one type and one field, that resolves to a fixed value. The `resolve` function can return a value, a promise, or an array of promises. A more complex example is included in the top level tests directory.

Then, serve the result of a query against that type schema.

```
var query = '{ hello }';

graphql(schema, query).then(result => {

  // Prints
  // {
  //   data: { hello: "world" }
  // }
  console.log(result);

});
```

This runs a query fetching the one field defined. The `graphql` function will first ensure the query is syntactically and semantically valid before executing it, reporting errors otherwise.

```
var query = '{ boyhowdy }';

graphql(schema, query).then(result => {

  // Prints
  // {
  //   errors: [
  //     { message: 'Cannot query field boyhowdy on RootQueryType',
  //       locations: [ { line: 1, column: 3 } ] }
  //   ]
  // }
  console.log(result);

});
```

Want to ride the bleeding edge?

The `npm` branch in this repository is automatically maintained to be the last commit to `master` to pass all tests, in the same form found on npm. It is recommended to use builds deployed to npm for many reasons, but if you want to use the latest not-yet-released version of graphql-js, you can do so by depending directly on this branch:

```
npm install graphql@git://github.com/graphql/graphql-js.git#npm
```

Using in a Browser

GraphQL.js is a general purpose library and can be used both in a Node server and in the browser. As an example, the GraphiQL tool is built with GraphQL.js!

Building a project using GraphQL.js with webpack or rollup should just work and only include the portions of the library you use. This works because GraphQL.js is distributed with both CommonJS (`require()`) and ESModule (`import`) files. Ensure that any custom build configurations look for `.mjs` files!

Contributing

We actively welcome pull requests, learn how to contribute.

Changelog

Changes are tracked as GitHub releases.

License

GraphQL.js is MIT-licensed.

Terms
Privacy
Security
Status
Help

Contact GitHub
API
Training
Shop
Blog
About