Published in smartive

Thilo Haas Follow

Aug 24, 2021 · 4 min read · ✦ · Listen

Save

The most popular Open-source GraphQL Servers and Clients

Which GraphQL Server and Client for Node.js should you use? A Comparison.

Don't know where to start? Need a little guidance to see through the nodes and edges? Then this post is for you.

We will walk you through the most recent and most popular open-source GraphQL Servers and Clients and share some useful hints and insights in our choices along the way.

Photo by Clint Adair on Unsplash

GraphQL Servers

Apollo Server

⭐11.7k GitHub stars

Apollo Server is an open-source GraphQL server with a great developer experience and a huge community. Its development is led by Apollo Graph Inc. which also provides enterprise support.

```
const { ApolloServer, gql } = require('apollo-server');

// The GraphQL schema
const typeDefs = gql`
  type Query {
    hello: String
  }
`;

// A map of functions which return data for the schema.
const resolvers = {
  Query: {
    hello: () => 'Hello World',
  },
};

const server = new ApolloServer({
  typeDefs,
  resolvers,
});

server.listen().then(({ url }) => {
  console.log(`🚀 Server ready at ${url}`);
});
```

graphql-server-apollo.js hosted with ❤️ by **GitHub** **view raw**

When to choose Apollo Server:

- simple setup
- excellent documentation
- great tooling and developer experience
- enterprise support

Introduction to Apollo Server

Apollo Server 3 has been released. Most applications require few or no changes to upgrade. For details, see the…

www.apollographql.com

GraphQL.js

⭐17.9k GitHub stars

GraphQL.js is the reference implementation of the GraphQL specification in Node.js. It is the best choice if you want to start small and don’t need the full blown and extensible Apollo framework.

```javascript
var { graphql, buildSchema } = require('graphql');

var schema = buildSchema(`
  type Query {
    hello: String
  }
`);

var root = { hello: () => 'Hello world!' };

graphql(schema, '{ hello }', root).then((response) => {
  console.log(response);
});
```

graphql-server-graphqljs.js hosted with ❤ by **GitHub** **view raw**

When to choose GraphQL.js:

- If you only need the most basic GraphQL server features
- Small footprint — not as big as the Apollo framework — therefore you need to do more by yourself

Also have a look at this Blogpost comparing GraphQL.js with Apollo.

Getting Started With GraphQL.js

Before getting started, you should have Node v6 installed, although the examples should mostly work in previous...

graphql.org

Express GraphQL

⭐6k GitHub stars

Express GraphQL is the reference implementation of a GraphQL server over an Express web server.

```javascript
var express = require('express');
var { graphqlHTTP } = require('express-graphql');
var { buildSchema } = require('graphql');

var schema = buildSchema(`
  type Query {
    hello: String
  }
`);

var root = { hello: () => 'Hello world!' };

var app = express();
app.use('/graphql', graphqlHTTP({
  schema: schema,
  rootValue: root,
  graphiql: true,
}));
app.listen(4000, () => console.log('Now browse to localhost:4000/graphql'));
```

graphql-server-graphql-express.js hosted with ❤ by **GitHub** **view raw**

When to choose Express GraphQL:

- Extend your existing Connect, Express or Restify API with a GraphQL endpoint
- Simple to get started

GitHub - graphql/express-graphql: Create a GraphQL HTTP server with Express.

Create a GraphQL HTTP server with any HTTP web framework that supports connect styled middleware, including Connect...

github.com

GraphQL Clients

When we want to connect our web frontends to a GraphQL backend it's always a trade-off between simplicity, tooling and performance (e.g. bundle size).

Browser Fetch

For small use cases and straightforward GraphQL queries, always start with a plain *fetch* call:

```javascript
// Fetch data from GitHub's GraphQL API:
const response = await fetch('https://api.github.com/graphql', {
  method: 'POST',
  headers: {
    Authorization: `bearer ${GITHUB_AUTH_TOKEN}`,
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    query: `query {
      viewer {
        login
      }
    }`,
  }),
});

// Get the response as JSON
return await response.json();
```

graphql-client-fetch.js hosted with ❤ by **GitHub** **view raw**

When to use *fetch*:

- For small applications
- Simple GraphQL calls

If it gets any more complex, let's have a look at the following GraphQL clients:

Apollo Client

⭐16.6k GitHub stars
📦75.7kb minified bundle size

Apollo Client is a full-featured GraphQL client that comes with caching and integrations for React, Angular, and more.

```
import { ApolloClient, gql, InMemoryCache } from "@apollo/client";

const client = new ApolloClient({
  uri: 'https://api.github.com/graphql',
  cache: new InMemoryCache(),
  headers: {
    Authorization: `bearer ${GITHUB_AUTH_TOKEN}`,
  },
});

client
  .query({
    query: gql`
      query {
        viewer {
          login
        }
      }
    `
  })
  .then(result => console.log(result));
```

graphql-client-apollo.js hosted with ❤️ by **GitHub** **view raw**

When to use Apollo Client:

- Full-featured GraphQL client
- Flexible client with a shallow learning curve
- Easy to integrate into existing applications
- Huge community & support

Apollo GraphQL

We believe in giving developers access to powerful open source tools. You will never be asked to pay for Apollo Client...

www.apollographql.com

Relay
⭐15.9k GitHub stars
📦122.4kb minified bundle size

Relay is a framework for building data-driven React applications on top of a GraphQL API. Its development is led by Facebook. Relay makes additional assumptions about the standard GraphQL specification. Therefore you need a Relay compatible GraphQL server.

When to use Relay:

- Full featured GraphQL client
- Structured and opinionated client that enforces a defined style
- Working with a big team on the same huge application
- Your GraphQL server follows the Relay GraphQL specification

Relay
Relay is data-fetching turned declarative. Components declare their data dependencies, without worrying about how to...

relay.dev

GraphQL Hooks
⭐1.5k GitHub stars
📦10.6kb minified bundle size

GraphQL Hooks is a minimal GraphQL client for React.

When to use GraphQL Hooks:

- Web performance: Tiny bundle size

- You only need the basic (query, mutations) GraphQL features
- Versatile, easy to integrate and use

graphql-hooks: 🎣Minimal hooks-first GraphQL client

🎣Minimal hooks-first GraphQL client. 🥇 First-class hooks API ⚖️Tiny bundle: only 7.6kB (2.8 gzipped) 📄Full SSR...

github.com

If you're still unsure whether to choose the Apollo or Relay client, have a look at Scott Domes' comparison of both clients within a React application:

Apollo and Relay Side by Side

Decide which GraphQL library is right for you

blog.bitsrc.io

Or another one by LogRocket comparing graphql-hooks with Apollo hooks: https://blog.logrocket.com/comparing-hooks-libraries-for-graphql/

Conclusion

Which library you choose always depends on your use case. On our backends we mostly use Apollo GraphQL server because of simplicity, great tooling, and enterprise support.

On our frontends we start simple with browser fetch calls. For more advanced applications we use the graphql-hooks library because of its small bundle size. Only if we need a more advanced feature set we use the Apollo GraphQL client, because the size of the package is not negligible.

This should give you a good starting point. If you want to know more, have a look at https://graphql.org/code/ for an extensive list of even more clients, servers and tools — also in other programming languages.

Build Enterprise-Grade GraphQL Applications

Want to become a GraphQL pro? Follow us and read our whole series on *enterprise-grade GraphQL* applications.

- Part 1: Building Enterprise Grade APIs with GraphQL, MySQL and Node.js

GraphQL API Best Practices Nodejs Web Development

- Part 2: Advanced GraphQL Patterns: The Almighty Root Resolver
- Part 3: Advanced GraphQL Patterns: Embrace the AST!

- Part 4: GraphQL and MySQL: Solving the Join Problem
- Part 5: GraphQL and Elasticsearch

25

What are your experiences with GraphQL? We would love to hear about your challenges and how you solved them! Did you like this blog post, or did we miss your favourite GraphQL library? Leave us a comment below or let's get in touch!

About Help Terms Privacy

Need an expert team to implement your advanced web application? Check out smartive's portfolio. At smartive, we build complex web applications and sophisticated products like our Community SaaS Reactions which is based on GraphQL. We are located in Zurich, Switzerland.