



Getting Started With GraphQL.js

Prerequisites

Before getting started, you should have Node v6 installed, although the examples should mostly work in previous versions of Node as well. For this guide, we won't use any language features that require transpilation, but we will use some ES6 features like [Promises](#), [classes](#), and [fat arrow functions](#), so if you aren't familiar with them you might want to read up on them first.

To create a new project and install GraphQL.js in your current directory:

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

Writing Code

To handle GraphQL queries, we need a schema that defines the `Query` type, and we need an API root with a function called a “resolver” for each API endpoint. For an API that just returns “Hello world!”, we can put this code in a file named `server.js`:

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

GRAPHQL.JS TUTORIAL

[Getting Started](#)[Running Express +](#)[GraphQL](#)[GraphQL Clients](#)[Basic Types](#)[Passing Arguments](#)[Object Types](#)[Mutations and Input](#)[Types](#)[Authentication &](#)[Middleware](#)

ADVANCED GUIDES

[Constructing Types](#)

API REFERENCE

[express-graphql](#)[graphqlHTTP](#)[graphql](#)[graphql](#)[graphql/error](#)[formatError](#)[GraphQLError](#)[locatedError](#)

```
// Construct a schema, using GraphQL schema language
var schema = buildSchema(`
  type Query {
    hello: String
  }
`);

// The rootValue provides a resolver function for each API endpoint
var rootValue = {
  hello: () => {
    return 'Hello world!';
  },
};

// Run the GraphQL query '{ hello }' and print out the response
graphql({
  schema,
  source: '{ hello }',
  rootValue
}).then((response) => {
  console.log(response);
});
```

If you run this with:

```
node server.js
```

You should see the GraphQL response printed out:

```
{ data: { hello: 'Hello world!' } }
```

Congratulations - you just executed a GraphQL query!

For practical applications, you'll probably want to run GraphQL queries from an API server, rather than executing GraphQL with a command line tool. To use GraphQL for an API server over HTTP, check out [Running an Express GraphQL Server](#).

[Continue Reading →](#)

Running an Express GraphQL Server

[syntaxError](#)

[graphql/execution](#)

[execute](#)

[graphql/language](#)

[BREAK](#)

[getLocation](#)

[Kind](#)

[lex](#)

[parse](#)

[parseValue](#)

[printSource](#)

[visit](#)

[graphql/type](#)

[getNamedType](#)

[getNullableType](#)

[GraphQLBoolean](#)

[GraphQLEnumType](#)

[GraphQLFloat](#)

[GraphQLID](#)

[GraphQLInputObjectType](#)

[GraphQLInt](#)

[GraphQLInterfaceType](#)

[GraphQLList](#)

[GraphQLNonNull](#)

[GraphQLObjectType](#)

[GraphQLScalarType](#)

[GraphQLSchema](#)

[GraphQLString](#)

[GraphQLUnionType](#)

[isAbstractType](#)

[isCompositeType](#)

[isInputType](#)

[isLeafType](#)

[isOutputType](#)

[graphql/utilities](#)

[astFromValue](#)

[buildASTSchema](#)

[buildClientSchema](#)

[buildSchema](#)

[introspectionQuery](#)

[isValidJSValue](#)

isValidLiteralValue
printIntrospectionSchema
printSchema
typeFromAST
TypeInfo

[graphql/validation](#)


specifiedRules
validate






Learn

Introduction to
GraphQL
Best Practices
Frequently Asked
Questions
Training Courses


Code

 GitHub
GraphQL
Specification
Libraries & Tools
Services & Vendors

Community

 @graphql
 Discord
 Stack Overflow
Resources
Events
Landscape

& More

News Blog
GraphQL
Foundation
GraphQL
Community Grant
Logo and Brand
Guidelines
Code of Conduct
Contact Us
 Edit this page

Copyright © 2022 The GraphQL Foundation. All rights reserved.

For web site terms of use, trademark policy and general project policies please see <https://lfpj.org>.