

gatsby-plugin-graphql-config

Persists Gatsby GraphQL schema and fragments to the `.cache` directory, as well as a GraphQL Config file to enable full-featured tooling for:

- `vscode-graphql`, and other IDE extensions that use the official GraphQL LSP
- `eslint-plugin-graphql`
- `graphql code generator` for Gatsby projects using TypeScript
- eventually `graphiql` will use it, even!

Install

```
npm install gatsby-plugin-graphql-config
```

How to use

First, add it to your plugin configuration:

```
// In your gatsby-config.js
```

```
plugins: [`gatsby-plugin-graphql-config`]
```

Simplest setup

If you are able to configure your tools to seek a different `basePath` for loading GraphQL Config, point them to `.cache` directory.

Manual setup for repositories with no other GraphQL projects

If your project is *only* a Gatsby project, you can place a `graphql.config.js` file at the root of your Gatsby project like this:

```
// <my project>/graphql.config.js
```

```
module.exports = require("./.cache/graphql.config.json")
```

If it's in a subdirectory such as a `site/` folder, you would use this:

```
// <my project>/graphql.config.js
```

```
module.exports = require("./site/.cache/graphql.config.json")
```

Manual setup for repositories with multiple GraphQL projects

If your repository has multiple GraphQL projects including Gatsby, you will want a config similar to this at the root:

```
// <my project>/graphql.config.js
```

```
module.exports = {
  projects: {
    site: require("packages/site/.cache/graphql.config.json"),
    server: {
      schema: "packages/server/src/**/*.graphql",
      documents: "packages/server/src/queries/**/*.ts,tsx,js,jsx",
    },
  },
}
```

How it works

It writes out these files to the Gatsby `.cache` directory:

- `schema.graphql` - a complete representation of the schema, including plugins
- `fragments.graphql` - all user, plugin and gatsby-core provided fragments in one file
- `graphql.config.json` - a `graphql-config@3` compatible config file with absolute file resolutions