

Learn what WPGraphQL is and how to use it with Gatsby and WordPress.

## What is WPGraphQL?

[WPGraphQL](#) is a WordPress plugin that adds a [GraphQL](#) API to your WordPress site. [GraphQL](#) is a query language for requesting information from an [API](#) and a protocol for servers that support it. Gatsby [uses GraphQL](#) to serve page data. Adding the WPGraphQL plugin to WordPress lets you use GraphQL to retrieve the specific pieces of content that you need to display on a Gatsby page.

## Installing WPGraphQL

You can install WPGraphQL using WordPress' *Add Plugins* screen. You may also want to install the [WPGraphiQL](#) plugin. WPGraphiQL turns the [GraphiQL IDE](#) (or integrated development environment) into a WordPress plugin. You do not have to install WPGraphiQL, but it helps you explore data and create queries from WordPress' administration interface.

1. Download [WPGraphQL](#) and [WPGraphiQL](#) as ZIP archives from their respective GitHub repositories.
2. Upload `wp-graphql-develop.zip` and `wp-graphiql-master.zip` using the *Upload Plugin* button on the *Add Plugins* screen. You'll have to upload and install them one at a time.

WPGraphQL and WPGraphiQL are also available from the [Packagist](#) repository. You can install them using [Composer](#), a package manager for PHP.

```
composer require wp-graphql/wp-graphql wp-graphql/wp-graphiql
```

After installing WPGraphQL and WPGraphiQL, you'll need to activate them. If you've used the upload method, WordPress will display an *Activate Plugin* button on the confirmation screen once the installation process completes. You can also activate (or deactivate) WPGraphQL and WPGraphiQL from the *Plugins* screen.

Once activated, you may need to update your permalink structure. WPGraphQL requires [pretty permalinks](#), and a custom permalink structure.

Verify that WPGraphQL is working correctly by visiting the `/graphql` endpoint, relative to your WordPress home URL. If, for example, your WordPress home URL is `https://www.example.com/blog/`, the GraphQL endpoint will be `https://www.example.com/blog/graphql`.

**NOTE:** If you are using the `WP_SITEURL` and `WP_HOME` WordPress constants, WPGraphQL will use the value of `WP_HOME`.

Then you can configure Gatsby to work with your new endpoint.

## Using WPGraphQL with Gatsby

You'll need to do two more things before you can use your WordPress-backed GraphQL server with Gatsby:

1. install the [gatsby-source-graphql](#) plugin; and
2. update `gatsby-config.js`.

Use [npm](#) to install [gatsby-source-graphql](#).

```
npm install gatsby-source-graphql
```

Then update `gatsby-config.js`. Add the plugin to your Gatsby instance. Specify the URL of the GraphQL endpoint and set other [configuration options](#).

```
module.exports = {
  plugins: [
    {
      resolve: "gatsby-source-graphql",
      options: {
        // Remote schema query type. This is an arbitrary name.
        typeName: "WPGraphQL",
        // Field name under which it will be available. Used in your Gatsby query.
        // This is also an arbitrary name.
        fieldName: "wpcontent",
        // GraphQL endpoint, relative to your WordPress home URL.
        url: "https://example.com/blog/graphql",
      },
    },
  ],
}
```

Be sure to restart the development server after making these changes. Now you can use your WordPress [GraphQL API with Gatsby](#) just as you would any other GraphQL source.

WPGraphQL is one way to use Gatsby and WordPress together. It adds a flexible GraphQL API to the robust content management features of WordPress.

### Learn more about WPGraphQL

- [WPGraphQL](#) official site
- [Using Gatsby with WPGraphQL](#) demo by the founder
- [What is GraphQL?](#) from the Gatsby docs
- [Why Gatsby Uses GraphQL](#)