

Data sourcing in Gatsby is plugin-driven; Source plugins fetch data from their source (e.g. the `gatsby-source-filesystem` plugin fetches data from the file system, the `gatsby-source-wordpress` plugin fetches data from the WordPress API, etc). You can also source the data yourself.

Adding data to GraphQL

Gatsby's [GraphQL data layer](#) uses nodes to model chunks of data. Gatsby source plugins add source nodes that you can query for, but you can also create source nodes yourself. To add custom data to the GraphQL data layer yourself, Gatsby provides methods you can leverage.

This recipe shows you how to add custom data using `createNode()`.

Directions

1. In `gatsby-node.js` use `sourceNodes()` and `actions.createNode()` to create and export nodes to be able to query the data.

```
exports.sourceNodes = ({ actions, createNodeId, createContentDigest }) => {
  const pokemons = [
    { name: "Pikachu", type: "electric" },
    { name: "Squirtle", type: "water" },
  ]

  pokemons.forEach(pokemon => {
    const node = {
      name: pokemon.name,
      type: pokemon.type,
      id: createNodeId(`Pokemon-${pokemon.name}`),
      internal: {
        type: "Pokemon",
        contentDigest: createContentDigest(pokemon),
      },
    }
    actions.createNode(node)
  })
}
```

2. Run `gatsby develop`.

Note: After making changes in `gatsby-node.js` you need to re-run `gatsby develop` for the changes to take effect.

3. Query the data (in [GraphQL](#) or in your components).

```
query MyPokemonQuery {
  allPokemon {
    nodes {
      name
      type
      id
    }
  }
}
```

Additional resources

- Walk through an example using the `gatsby-source-filesystem` plugin in [tutorial part five](#)
- Search available source plugins in the [Gatsby library](#)
- Understand source plugins by building one in the [source plugin tutorial](#)
- The `createNode` function [documentation](#)

Sourcing Markdown data for blog posts and pages with GraphQL

You can source Markdown data and use Gatsby's [createPages API](#) to create pages dynamically.

This recipe shows how to create pages from Markdown files on your local filesystem using Gatsby's GraphQL data layer.

Prerequisites

- A [Gatsby site](#) with a `gatsby-config.js` file
- The [Gatsby CLI](#) installed
- The [gatsby-source-filesystem plugin](#) installed
- The [gatsby-transformer-remark plugin](#) installed
- A `gatsby-node.js` file

Directions

1. In `gatsby-config.js`, configure `gatsby-transformer-remark` along with `gatsby-source-filesystem` to pull in Markdown files from a source folder. This would be in addition to any previous `gatsby-source-filesystem` entries, such as for images:

```
module.exports = {
  plugins: [
    `gatsby-transformer-remark`,
    {
      resolve: `gatsby-source-filesystem`,
      options: {
        name: `content`,
        path: `${__dirname}/src/content`,
      },
    },
  ],
}
```

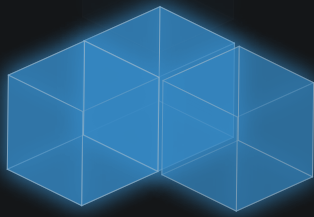
2. Add a Markdown post to `src/content`, including frontmatter for the title, date, and path, with some initial content for the body of the post:

```
---
title: My First Post
date: 2019-07-10
path: /my-first-post
---

This is my first Gatsby post written in Markdown!
```

3. Start up the development server with `gatsby develop`, navigate to the GraphQL explorer at `http://localhost:8000/___graphql`, and write a query to get all markdown data:

```
{  
  allMarkdownRemark {  
    edges {  
      node {  
        frontmatter {  
          path  
        }  
      }  
    }  
  }  
}
```



Initializing Sandbox Container