

Sourcing Content from JSON or YAML

As you work with Gatsby, you might want to source data from a JSON or YAML file directly into a page or component. This guide will cover approaches for those techniques, as well as architecting a Gatsby site from a YAML file.

To follow along with the JSON or YAML data sourcing tips outlined in this guide, you can start by creating a new Gatsby site based on the official hello world starter.

Open up a terminal and run the following command:

```
gatsby new gatsby-YAML-JSON-at-buildtime https://github.com/gatsbyjs/gatsby-starter-hello-world
```

Directly import data with YAML

This section starts with YAML data sourcing. If you want to see how to do it using JSON instead, jump to the next section.

Add the YAML content

In your Gatsby project folder, create a directory called `content` and inside, add a file called `My-YAML-Content.yaml` with the following content:

```
title: YAML content used at build time with Gatsby
content:
  - item:
      Cupcake ipsum dolor. Sit amet marshmallow topping cheesecake muffin. Halvah
      croissant candy canes bonbon candy. Apple pie jelly beans topping carrot cake
      danish tart cake cheesecake. Muffin danish chocolate soufflé pastry icing bonbon
      oat cake. Powder cake jujubes oat cake. Lemon drops tootsie roll marshmallow halvah
      carrot cake.
  - item:
      Doggo ipsum borkdrive much ruin diet you are doing me the shock the neighborhood pupper
      Very good spot very jealous pupper very hand that feed shibe smol, shoob.
      Long bois pupper doggo you are doin me a concern big ol yapper, smol boof most angry
      Mlem blep wow very biscit dat tungg tho wow very biscit, thicc ur givin me a spook.
      Many pats heckin you are doing me the shock corgo ur givin me a spook very hand that f
      Such treat what a nice floof super chub such treat, smol thicc.
      Puggorino very good spot most angry pupper I have ever seen you are doing me the sho
```

```

    What a nice floof long doggo blep length boy borking doggo, much ruin diet floofs bork
  - item: 192.33
  - item: 111111

```

Import YAML into the page component

Now that you have something you want to show, the only thing missing is to create a page that will consume the data.

Add a new file called `yml-at-buildtime.js` inside the `pages` folder, with the following code:

```

import React from "react"
import YAMLData from "../../content/My-YAML-Content.yaml"

const YAMLbuildtime = () => (
  <div style={{ maxWidth: `960px`, margin: `1.45rem` }}>
    <h1>{YAMLData.title}</h1>
    <ul>
      {YAMLData.content.map((data, index) => {
        return <li key={`content_item_${index}`}>{data.item}</li>
      })}
    </ul>
  </div>
)
export default YAMLbuildtime

```

The above code imports YAML source data as an array, iterates over it with the `Array.map` method, and renders the data-filled markup through a functional React component.

Directly import data with JSON

In addition to (or instead of) sourcing from YAML, you can use JSON as a data source in a Gatsby site.

Add the JSON content

In your Gatsby project folder, create a directory named `content` if it doesn't exist, and then add a new file inside called `My-JSON-Content.json` with the following content:

```

{
  "title": "JSON content used at build time with Gatsby",
  "content": [
    {
      "item": "Cupcake ipsum dolor. Sit amet marshmallow topping cheesecake muffin. Halvah
    },

```

```

    {
      "item": "Doggo ipsum borkdrive much ruin diet you are doing me the shock the neighborh
    },
    {
      "item": 192.33
    },
    {
      "item": 111111
    }
  ]
}

```

Import JSON into the page component

Now that you have JSON data that needs to be shown, all that's missing is a page to consume it.

Add a new file called `json-at-buildtime.js` inside the `pages` folder with the following code:

```

import React from "react"
import JSONData from "../../content/My-JSON-Content.json"

const JSONbuildtime = () => (
  <div style={{ maxWidth: `960px`, margin: `1.45rem` }}>
    <h1>{JSONData.title}</h1>
    <ul>
      {JSONData.content.map((data, index) => {
        return <li key={`content_item_${index}`}>{data.item}</li>
      })}
    </ul>
  </div>
)
export default JSONbuildtime

```

Similar to the YAML example above, this code snippet shows how to import a JSON file for sourcing data. When imported, the data can be iterated upon with the `Array.map` method and rendered in a React component.

Out of the box and without any extra configuration, the page will show content sourced from a JSON file.

Build a Gatsby site sourced from YAML

You can also build a fully functional Gatsby site with a page structure sourced from a YAML file.

Add necessary dependencies

For this example, you will need to add an extra dependency so that the file containing the site structure and its contents can be loaded and interpreted safely.

Open your terminal, navigate to the folder containing the Gatsby site, and issue the following command:

```
npm install js-yaml
```

This newly added package will be responsible for loading and parsing the YAML file safely.

Add some content

Create a folder named `content` if it doesn't exist, and then add a new file inside called `index.yaml` with the following contents:

```
- path: "/page1"
  content:
    - item: one item
    - item: two items
    - item: three items
  links:
    - to: "/page2"
    - to: "/page5"
- path: "/page2"
  content:
    - item:
      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
      tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
      quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
      Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore
      eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt
      in culpa qui officia deserunt mollit anim id est laborum.
  links:
    - to: "/page1"
- path: "/page3"
  content:
    - item: Cupcake ipsum dolor sit amet tootsie roll sesame snaps chupa chups.
      Sugar plum chupa chups topping I love carrot cake I love marshmallow dessert.
      Toffee gingerbread pie apple pie jelly beans pastry cookie.
      Lemon drops wafer I love pastry halvah dragée pudding cake.
      Cake halvah cookie jelly beans topping pudding cheesecake donut.
      Cake croissant marshmallow.
      Sesame snaps apple pie I love I love cake danish powder.
      Lollipop sweet caramels.
```

```

    Tiramisu danish marshmallow candy canes.
    Powder cupcake cotton candy bonbon chocolate bar marshmallow gummies cheesecake mar
    Gummies soufflé candy. Candy canes muffin chocolate brownie pudding.
  links: []
- path: "/page4"
  content:
    - item:
      Lorem ipsum dolor amet mustache knausgaard +1, blue bottle waistcoat tbh
      semiotics artisan synth stumptown gastropub cornhole celiac swag. Brunch raclette
      vexillologist post-ironic glossier ennui XOXO mlkshk godard pour-over blog tumblr
      humblebrag. Blue bottle put a bird on it twee prism biodiesel brooklyn. Blue
      bottle ennui tbh succulents.
  links:
    - to: "/page5"
    - to: "/page1"
- path: "/page5"
  content:
    - item: St. agur blue cheese queso cheesecake.
      Cheesecake the big cheese monterey jack cheesecake monterey jack paneer halloumi ru
      Cheese triangles cheese strings cheese slices cheesy feet taleggio cottage cheese wh
      Feta cauliflower cheese babybel cheese on toast monterey jack.
    - item:
      Doggo ipsum borkdrive much ruin diet you are doing me the shock the neighborhood pup
      Very good spot very jealous pupper very hand that feed shibe smol, shoob.
      Long bois pupper doggo you are doin me a concern big ol yapper, smol boof most anger
      Mlem blep wow very biscit dat tungg tho wow very biscit, thicc ur givin me a spook.
      Many pats heckin you are doing me the shock corgo ur givin me a spook very hand that
      Such treat what a nice floof super chub such treat, smol thicc.
      Puggorino very good spot most angry pupper I have ever seen you are doing me the sh
      What a nice floof long doggo blep length boy borking doggo, much ruin diet floofs bo
  links:
    - to: "/page1"

```

The code block above creates a YAML object in which:

- Each **path** is a page's endpoint (the relevant part of its URL).
- The **contents** list holds some data to be displayed.
- The **links** list holds some endpoints to other pages.

Configure Gatsby pages

Once the dynamic site structure and content exists, you need to tell Gatsby to generate the appropriate pages and display the contents for each one.

If you don't already have one, create a `gatsby-node.js` file at the root of the project. Add the following code inside the file:

```
const fs = require("fs")
```

```

const yaml = require("js-yaml")
exports.createPages = ({ actions }) => {
  const { createPage } = actions
  const yamlDoc = yaml.load(fs.readFileSync("./content/index.yaml", "utf-8"))
  yamlDoc.forEach(element => {
    createPage({
      path: element.path,
      component: require.resolve("./src/templates/basicTemplate.js"),
      context: {
        pageContent: element.content,
        links: element.links,
      },
    })
  })
}

```

Breaking down this code excerpt into smaller parts:

1. Import the `js-yaml` package you installed earlier.
2. Load the `index.yaml` file and parse the content.
3. Using Gatsby's `createPage()` API, create some pages programmatically from the parsed file.
4. Use the `context` property to pass your data into the page as a special prop named `pageContext`, allowing it to be consumed. You can read more about `context` in creating and modifying pages.

Create a template

To complete the process of rendering the sourced content, you'll need to create a template for producing dynamic pages from data.

To match the component referenced in `gatsby-config.js`, create a file called `basicTemplate.js` in the `src/templates/` folder and add the following:

```

import React from "react"
import { Link } from "gatsby"
const basicTemplate = props => {
  const { pageContext } = props
  const { pageContent, links } = pageContext

  return (
    <div style={{ maxWidth: `960px`, margin: `1.45rem` }}>
      <ul>
        {pageContent.map((data, index) => {
          return <li key={`content_item_${index}`}>{data.item}</li>
        })}
      </ul>
    </div>
  )
}

```

```

    {links.map((item, index) => {
      return (
        <li key={`link_${index}`}>
          <Link to={item.to}>{item.to}</Link>
        </li>
      )
    })}
  </ul>
</div>
)
}
export default basicTemplate

```

Join the pieces

After parsing a YAML file into data and configuring Gatsby to produce pages with a template, you should have the following file and folder structure:

```

|gatsby-YAML-JSON-at-buildtime
|content
|  - index.yaml
|src
|  templates
|    - basicTemplate.js
- gatsby-node.js

```

Running `gatsby develop` in the terminal and opening a browser window to `http://localhost:8000/page1` you'll see a page with content that was sourced from a YAML file used to generate your site.

To make this work on your existing Gatsby site, you would need to:

- Copy over the `gatsby-node.js` file contents: <https://github.com/gatsbyjs/gatsby/blob/master/examples/using-gatsby-with-json-yaml/gatsby-node.js>
- Create a basic template: <https://github.com/gatsbyjs/gatsby/blob/master/examples/using-gatsby-with-json-yaml/src/templates/basicTemplate.js>
- And grab the YAML file: <https://github.com/gatsbyjs/gatsby/blob/master/examples/using-gatsby-with-json-yaml/content/index.yaml>