

Recipes: Pages and Layouts

Add pages to your Gatsby site, and use layouts to manage common page elements.

Project structure

Inside a Gatsby project, you may see some or all of the following folders and files:

```
|-- /.cache
|-- /plugins
|-- /public
|-- /src
    |-- /pages
    |-- /templates
    |-- html.js
|-- /static
|-- gatsby-config.js
|-- gatsby-node.js
|-- gatsby-ssr.js
|-- gatsby-browser.js
```

Some notable files and their definitions:

- **gatsby-config.js** — configure options for a Gatsby site, with metadata for project title, description, plugins, etc.
- **gatsby-node.js** — implement Gatsby's Node.js APIs to customize and extend default settings affecting the build process
- **gatsby-browser.js** — customize and extend default settings affecting the browser, using Gatsby's browser APIs
- **gatsby-ssr.js** — use Gatsby's server-side rendering APIs to customize default settings affecting server-side rendering

Additional resources

- For a tour of all the common folders and files, read the docs on Gatsby's Project Structure
- For common commands, check out the Gatsby CLI docs
- Check out the Gatsby Cheat Sheet for downloadable info at a glance

Creating pages automatically

Gatsby core automatically turns React components in `src/pages` into pages with URLs. For example, components at `src/pages/index.js` and `src/pages/about.js` would automatically create pages from those filenames for the site's index page (/) and /about.

Prerequisites

- A Gatsby site
- The Gatsby CLI installed

Directions

1. Create a directory for `src/pages` if your site doesn't already have one.
2. Add a component file to the pages directory:

```
import React from "react"

const AboutPage = () => (
  <main>
    <h1>About the Author</h1>
    <p>Welcome to my Gatsby site.</p>
  </main>
)

export default AboutPage
```

3. Run `gatsby develop` to start the development server.
4. Visit your new page in the browser: `http://localhost:8000/about`

Additional resources

- Creating and modifying pages

Linking between pages

Routing for links internal to your Gatsby site relies on the `<Link />` component.

Prerequisites

- A Gatsby site with two page components: `index.js` and `contact.js`
- The Gatsby CLI to run `gatsby develop`

Directions

1. Open the index page component (`src/pages/index.js`) and import the `<Link />` component from Gatsby. Add a `<Link />` component to the JSX

code and give it a `to` property with the pathname value of `"/contact/"` to output an HTML link with Gatsby powers:

```
import React from "react"
import { Link } from "gatsby" // highlight-line

export default function Home() {
  return (
    <main>
      <h1>What a world.</h1>
      <p>
        <Link to="/contact/">Contact</Link> // highlight-line
      </p>
    </main>
  )
}
```

2. Run `gatsby develop` and navigate to the index page. You should have a link that takes you to the contact page when clicked!

Note: Gatsby's `<Link />` component is a wrapper around `@reach/router`'s `Link` component. It outputs an HTML anchor when rendered in a browser, with built-in JavaScript functionality for performance. For more information, consult the API reference for `<Link />`.

Additional resources

- [Linking Between Pages guide](#)
- [Gatsby Link API](#)

Creating a layout component

It's common to wrap pages with a React layout component, which makes it possible to share markup, styles, and functionality across multiple pages.

Prerequisites

- A Gatsby Site

Directions

1. Create a layout component in `src/components`, where child components will be passed in as props:

```
import React from "react"

export default function Layout({ children }) {
  return (
```

```

    <div style={{ margin: `0 auto`, maxWidth: 650, padding: `0 1rem` }}>
      {children}
    </div>
  )
}

```

2. Import and use the layout component in a page:

```

import React from "react"
import { Link } from "gatsby"
import Layout from "../components/layout"

export default function Home() {
  return (
    <Layout>
      <Link to="/contact/">Contact</Link>
      <p>What a world.</p>
    </Layout>
  )
}

```

Additional resources

- Create a layout component in tutorial part three
- Styling with Layout Components

Creating pages programmatically with createPage

You can create pages programmatically in the `gatsby-node.js` file with helper methods Gatsby provides.

Prerequisites

- A Gatsby site
- A `gatsby-node.js` file

Directions

1. In `gatsby-node.js`, add an export for `createPages`

```

// highlight-start
exports.createPages = ({ actions }) => {
  // ...
}
// highlight-end

```

2. Destructure the `createPage` action from the available actions so it can be called by itself, and add or get data

```

exports.createPages = ({ actions }) => {
  // highlight-start
  const { createPage } = actions
  // pull in or use whatever data
  const dogData = [
    {
      name: "Fido",
      breed: "Sheltie",
    },
    {
      name: "Sparky",
      breed: "Corgi",
    },
  ]
  // highlight-end
}

```

3. Loop through the data in `gatsby-node.js` and provide the path, template, and context (data that will be passed in the props' `pageContext`) to `createPage` for each invocation

```

exports.createPages = ({ actions }) => {
  const { createPage } = actions

  const dogData = [
    {
      name: "Fido",
      breed: "Sheltie",
    },
    {
      name: "Sparky",
      breed: "Corgi",
    },
  ]
  // highlight-start
  dogData.forEach(dog => {
    createPage({
      path: `/${dog.name}`,
      component: require.resolve(`./src/templates/dog-template.js`),
      context: { dog },
    })
  })
  // highlight-end
}

```

4. Create a React component to serve as the template for your page that was used in `createPage`

```
import React from "react"

export default function DogTemplate({ pageContext: { dog } }) {
  return (
    <section>
      {dog.name} - {dog.breed}
    </section>
  )
}
```

5. Run `gatsby develop` and navigate to the path of one of the pages you created (like at `http://localhost:8000/Fido`) to see the data you passed it displayed on the page

Additional resources

- Tutorial section on programmatically creating pages from data
- Reference guide on using Gatsby without GraphQL
- Example repo for this recipe