Adding React Components

This guide covers how to add React components, including those from third-party component libraries, to your Gatsby site.

React components

React components are prebuilt elements or groups of elements that can be used to split your user interface (UI) into independent, reusable pieces. There are multiple types of components you can write but this guide covers functional components. For more in-depth information on writing React components, including classes, check out the React documentation.

Components can also be customized using inputs, better known as "props" (properties). Props can be of any JavaScript data type, such as Boolean, String, or Object.

For example, you could use a component for Buttons on your site. This would enable them to be used multiple times across pages with different labels or actions each time.

Importing React components

In Gatsby, when using React components, you can import and use them like you would in a React application. Here's an example of the Gatsby Link component in action, which brings with it extra functionality for performance:

```
import React from "react"
import { Link } from "gatsby"

export default function Contact() {
  return (
    <div>
      <Link to="/contact/">Contact</Link>
    </div>
  )
}
```

Importing third-party components

Just like React, Gatsby also supports third-party components and libraries. You can install a third-party component or library via your package manager. We tend to favor and use npm and we will reflect this in our examples.

Here's an example of adding a third-party component to your site.

First, you have to install the component or library's package via a package manager. It's recommended not to mix package managers, so if you use npm, don't use another and vice versa. If there's a relevant Gatsby plugin, you should install and use that plugin.

In this example, you'll install both a plugin and the library it depends on:

```
npm install gatsby-plugin-material-ui @material-ui/core
```

After installation, add any necessary plugins to the plugins array in `gatsby-config.js`:

```
module.exports = {
  plugins: [`gatsby-plugin-material-ui`],
}
```

Import and use the library in your page's source:

```
import React from "react"

// import my fancy third-party component
import Button from "@material-ui/core/Button"

export default function Home() {
  return (
    <div>
      <p>This is my super awesome page made with Gatsby!</p>

      {/* use my fancy third-party component */}
      <Button variant="contained">Fancy button!</Button>
    </div>
  )
}
```

Things to watch out for

Since Gatsby uses server-side rendering (SSR) to generate your site's pages, the JSX code you write is usually compiled before the browser loads the page. Because of this, certain features are not available at compile time and can cause a build error.

Use of browser globals

Some components or code reference browser globals such as `window`, `document` or `localStorage`. These objects are not available at build time and can result in a webpack error when compiling:

```
WebpackError: ReferenceError: window is not defined
```

To learn more about solutions for supporting SSR and client-side libraries, check out the related documentation section on Porting from Create React App.

Fixing third-party modules Some packages expect `window` or another browser global to be defined. These packages will have to be patched.

You can learn how to patch these packages on the Debugging HTML Builds documentation.

Components without server-side rendering

Server-side rendering means pages and content are built out by the Node.js server and then sent to a browser ready to go. It's like your pages are constructed before even being sent to the user. Gatsby is server-side rendered at build time, meaning that the code that gets to your browser has already been run to build pages and content, but this doesn't mean you can't still have dynamic pages.

Some React components don't have server-side rendering support (SSR) out-of-the-box so you might have to add SSR yourself.