Bootstrap is finished! And we've written the resulting pages to disk. The next step is to render each page to HTML. And, we must build a JavaScript runtime that takes over once the HTML has loaded so that future page navigations load instantly.

The next stages of the build lean heavily on webpack for code optimization and code splitting. If you haven't already, it's worth diving into webpack's docs to learn how it works.

/.cache/

All the files required by webpack are in your site's `.cache` directory. This is empty when you initialize a new project and can be safely deleted. Gatsby creates and fills it over the course of a build.

At the start of the build, Gatsby copies all files in gatsby/cache-dir into your `.cache` directory. This includes things like static-entry.js and production-app.js that you'll read about in the next sections. Essentially, all files that are needed by Gatsby to run in the browser, or to generate HTML, are included in `cache-dir`.

Since webpack doesn't know about Redux, we also need to create files that contain all the page data that was built up during bootstrap. And these all need to be placed in `.cache` as well. This is what the previous Write Out Pages section dealt with.