Gatsby themes are designed to be easy to create from an existing starter. This guide will walk you through the main steps of converting your starter to a theme.

What is a starter? What is a theme?

A starter is a boilerplate Gatsby site that users can copy and customize. Once modified, a starter maintains no connection to its source.

A theme is a type of plugin that includes a `gatsby-config.js` file and adds pre-configured functionality, data sourcing, and/or UI code to Gatsby sites. In contrast to starters, themes can be packaged and distributed through a registry like npm, and their versions can be tracked/managed through a `package.json` file.

One reason to convert a starter to a theme is to make it easier to push updates out to consumers of your code. With a starter, users would have to try and update their code from the original starter repo and run the risk of overwriting some of their own changes. With a theme, it's much easier for developers to update code through their package manager and rely on a consistent theme API that respects their customizations.

Prepare Your `package.json`

To start converting your starter to a library, get started by updating your `package.json` to use the `gatsby-theme-*` naming convention. If your starter is `gatsby-starter-awesome-blog` you can update the name key to `gatsby-theme-awesome-blog` (and double check that it's available on npm).

Specify `gatsby` , `react` , and `react-dom` as `devDependencies` . It's preferable to add them as `peerDependencies` as well. This helps end users determine which versions they want and npm/yarn will be able to resolve them properly.

In addition to updating your dependencies, you will need to create an `index.js` file in the root of your project. This allows Gatsby to resolve the theme since Node automatically looks for `index.js` .

Handling path resolution

One of the key differences between themes and starters is that a theme is no longer executed when the Gatsby CLI is being run since it's now a dependency. This often results in errors sourcing content and finding templates since they will look in the end user's directory.

In order to fix this, consider the following code that works as a starter:

```
const createPosts = (createPage, createRedirect, edges) => {
  edges.forEach(({ node }, i) => {
    // ...

    createPage({
      path: pagePath,
      component: path.resolve(`./src/templates/post.js`),
      context: {
        id: node.id,
        prev,
        next,
      },
    })
  })
}
```

Since `path.resolve` is being used the starter will resolve `src/templates/post.js` rather than `node_modules/gatsby-theme-awesome-blog/src/templates/post.js`. In order to fix this you can use `require.resolve` which will look relative to the theme so the correct template is found.

```
const createPosts = (createPage, createRedirect, edges) => {
  edges.forEach(({ node }, i) => {
    // ...

    createPage({
      path: pagePath,
      component: require.resolve(`./src/templates/post.js`), // highlight-line
      context: {
        id: node.id,
        prev,
        next,
      },
    })
  })
}
```

There may be other locations where you will need to update the path resolution like your `gatsby-config.js` as well.

Sourcing pages

Gatsby by default sources pages relative from `src/pages`, like a regular Gatsby site does. However, if you would like to source pages from a different directory you'll have to setup [`gatsby-plugin-page-creator`].

```
npm install gatsby-plugin-page-creator
```

Then, tell the plugin to look in your theme's `src/<custom-directory>` directory.

```
{
  resolve: `gatsby-plugin-page-creator`,
  options: {
    path: path.join(__dirname, `src`, `<custom-directory>`),
  },
},
```

Publishing to npm

In order to allow others to install your theme you will need to publish it to npm. If this is new for you, learn how [to publish to npm].

From the root of your newly created theme run `npm publish`.

Once you've published, you can install the theme in your starter.

```
npm install gatsby-theme-NAME
```

Walkthrough

- Jason Lengstorf converts an existing Gatsby site to a theme