The quickest way to get up and running with a workspace for building themes is to use the official `gatsby-starter-theme-workspace` starter.

To get started, run:

```
gatsby new gatsby-theme-my-theme gatsbyjs/gatsby-starter-theme-workspace
```

This will generate a new project for you. The file tree will look like this:

```
.
├── example
│   ├── src
│   │   └── pages
│   │       └── index.js
│   ├── gatsby-config.js
│   ├── package.json
│   └── README.md
├── gatsby-theme-minimal
│   ├── gatsby-config.js
│   ├── index.js
│   ├── package.json
│   └── README.md
├── .gitignore
├── package.json
├── README.md
└── yarn.lock
```

Yarn workspaces are a great way to set up a project for theme development because they support housing multiple packages in a single parent directory and link them together.

For Gatsby theme development, that means you can keep multiple themes and example sites together in a single project, and develop them locally.

> 💡 *If you prefer, you can develop themes as [local plugins](). Using `yarn link` or `npm link` are also viable alternatives. In general, Gatsby recommends the yarn workspaces approach for building themes, and that's what the starter and this guide will reflect.*

> 💡 *The starter takes care of all of the configuration for developing a theme using yarn workspaces. If you're interested in more detail on this setup, check out [this blog post]().*

`package.json`

The `package.json` in the root of the new project is primarily responsible for setting up the yarn workspaces. In this case, there are two workspaces, `gatsby-theme-minimal` and `example`.

```
{
  "name": "gatsby-starter-theme-workspace",
  "private": true,
  "version": "0.0.1",
  "main": "index.js",
  "license": "MIT",
  "scripts": {
    "build": "yarn workspace example build"
  },
  // highlight-start
  "workspaces": ["gatsby-theme-minimal", "example"]
  // highlight-end
}
```

`/gatsby-theme-minimal`

The `/gatsby-theme-minimal` directory is the starting point of the new theme you'll develop.

Inside it you'll find:

- `gatsby-config.js` : An empty gatsby-config that you can use as a starting point for building functionality into your theme.
- `index.js` : Since themes also function as plugins, this is an empty file that Gatsby requires in order to use this theme as a plugin.
- `package.json` : A file listing the dependencies that your theme will pull in when people install it. Gatsby should be a peer dependency.

`/example`

The `/example` directory is an example Gatsby site that installs and uses the local theme, `gatsby-theme-minimal` .

Inside it you'll find:

- `gatsby-config.js` : Specifies which theme to use and any other one-off configuration a site might need.
- `/src` : Contains source code such as custom pages or components that might live in a user's site.

Publishing a Theme

After building your theme, you may want to publish it for the Gatsby community. To do so, please publish your theme plugin to the plugin library.

Further resources

Gatsby Theme Authoring (Video course)

Watch the new Egghead course on Authoring Gatsby Themes.

Building a Gatsby Theme (Tutorial)

Check out the tutorial for building a Gatsby theme. The step-by-step tutorial goes into much more detail than this docs guide. It was written as a companion to the Egghead theme authoring course, so they can be used together!

Theme API reference

Check out the Theme API documentation.

Accessibility

A Gatsby theme is a Gatsby site, therefore building with accessibility in mind is critical. Check out these tips on making your sites (and themes!) accessible.

Read through source code

Check out how some existing themes are built:

- The official Gatsby Blog Theme
- The official Gatsby Notes Theme
- The Apollo themes. (*You might also be interested in the Apollo case study on themes on the blog.*)