The Gatsby command line tool (CLI) is the main entry point for getting up and running with a Gatsby application and for using functionality including running a development server and building out your Gatsby application for deployment.

This page provides similar documentation as the gatsby-cli README. The Gatsby cheat sheet has docs for top CLI commands & APIs all ready to print out.

How to use gatsby-cli

The Gatsby CLI is available via npm and is installed globally by running `npm install -g gatsby-cli`.

You can also use the `package.json` script variant of these commands, typically exposed *for you* with most starters. For example, if you want to make the `gatsby develop` command available in your application, open up `package.json` and add a script like so:

```
{
  "scripts": {
    "develop": "gatsby develop"
  }
}
```

API commands

All the following documentation is available in the tool by running `gatsby --help`.

`new`

Usage

```
gatsby new
```

The CLI will run an interactive shell asking for these options before creating a Gatsby site for you:

```
gatsby new

What would you like to name the folder where your site will be created?
my-gatsby-site

Will you be using a CMS? (single choice)
  No (or I'll add it later)
  –
  WordPress
  Contentful
  Sanity
  DatoCMS
  Shopify

Would you like to install a styling system? (single choice)
  No (or I'll add it later)
  –
  CSS Modules/PostCSS
  styled-components
  Emotion
  Sass
  Theme UI

Would you like to install additional features with other plugins? (multiple choice)
  ◯ Add the Google Analytics tracking script
  ◯ Add responsive images
  ◯ Add page meta tags with React Helmet
  ◯ Add an automatic sitemap
  ◯ Enable offline functionality
  ◯ Generate a manifest file
  ◯ Add Markdown support (without MDX)
  ◯ Add Markdown and MDX support
```

Creating a site from a starter

To create a site from a starter instead, run the command with your site name and starter URL:

```
gatsby new [<site-name> [<starter-url>]]
```

Note that this will not prompt you to create a custom setup, but only clone the starter from the URL you specified.

Arguments

Argument	Description
site-name	Your Gatsby site name, which is also used to create a project directory.
starter-url	A Gatsby starter URL or local file path. Defaults to [gatsby-starter-default](); see the [Gatsby starters]() docs for more information.

> *Note: The `site-name` should only consist of letters and numbers. If you specify a `.`, `./` or a `<space>` in the name, `gatsby new` will throw an error.*

Examples

- Create a Gatsby site named `my-awesome-site` using the default starter:

```
gatsby new my-awesome-site
```

- Create a Gatsby site named `my-awesome-blog-site`, using [gatsby-starter-blog]():

```
gatsby new my-awesome-blog-site https://github.com/gatsbyjs/gatsby-starter-blog
```

See the [Gatsby starters docs]() for more details.

`develop`

Once you've installed a Gatsby site, go to the root directory of your project and start the development server:

`gatsby develop`

Options

Option	Description
`-H`, `--host`	Set host. Defaults to localhost
`-p`, `--port`	Set port. Defaults to env.PORT or 8000
`-o`, `--open`	Open the site in your (default) browser for you
`-S`, `--https`	Use HTTPS
`--inspect`	Opens a port for debugging

Follow the Local HTTPS guide to find out how you can set up an HTTPS development server using Gatsby.

Preview changes on other devices

You can use the Gatsby develop command with the host option to access your dev environment on other devices on the same network, run:

```
gatsby develop -H 0.0.0.0
```

Then the terminal will log information as usual, but will additionally include a URL that you can navigate to from a client on the same network to see how the site renders.

```
You can now view gatsbyjs.com in the browser.

  Local:            http://0.0.0.0:8000/
  On Your Network:  http://192.168.0.212:8000/ // highlight-line
```

Note: To access Gatsby on your local machine, use either `http://localhost:8000` or the "On Your Network" URL.

`build`

At the root of a Gatsby site, compile your application and make it ready for deployment:

`gatsby build`

Options

Option	Description
`--prefix-paths`	Build site with link paths prefixed (set pathPrefix in your config)
`--no-uglify`	Build site without uglifying JS bundles (for debugging)
`--profile`	Build site with react profiling. See Profiling Site Performance with React Profiler
`--open-tracing-config-file`	Tracer configuration file (OpenTracing compatible). See Performance Tracing
`--graphql-tracing`	Trace (see above) every graphql resolver, may have performance implications.
`--no-color`, `--no-colors`	Disables colored terminal output

In addition to these build options, there are some optional build environment variables for more advanced configurations that can adjust how a build runs. For example, setting `CI=true` as an environment variable will tailor output for dumb terminals.

`serve`

At the root of a Gatsby site, serve the production build of your site for testing:

```
gatsby serve
```

Options

Option	Description
`-H`, `--host`	Set host. Defaults to localhost
`-p`, `--port`	Set port. Defaults to 9000
`-o`, `--open`	Open the site in your (default) browser for you
`--prefix-paths`	Serve site with link paths prefixed (if built with pathPrefix in your gatsby-config.js).

`info`

At the root of a Gatsby site, get helpful environment information which will be required when reporting a bug:

```
gatsby info
```

Options

Option	Description
`-C`, `--clipboard`	Automagically copy environment information to clipboard

`clean`

At the root of a Gatsby site, wipe out the cache (`.cache` folder) and public directories:

```
gatsby clean
```

This is useful as a last resort when your local project seems to have issues or content does not seem to be refreshing. Issues this may fix commonly include:

- Stale data, e.g. this file/resource/etc. isn't appearing
- GraphQL error, e.g. this GraphQL resource should be present but is not
- Dependency issues, e.g. invalid version, cryptic errors in console, etc.
- Plugin issues, e.g. developing a local plugin and changes don't seem to be taking effect

`plugin`

Run commands pertaining to gatsby plugins.

`docs`

```
gatsby plugin docs
```

Directs you to documentation about using and creating plugins.

Repl

Get a Node.js REPL (interactive shell) with context of your Gatsby environment:

`gatsby repl`

Gatsby will prompt you to type in commands and explore. When it shows this: `gatsby >`

You can type in a command, such as one of these:

`babelrc`

`components`

`dataPaths`

`getNodes()`

`nodes`

`pages`

`schema`

`siteConfig`

`staticQueries`

When combined with the GraphQL explorer, these REPL commands could be very helpful for understanding your Gatsby site's data.

For more information, check out the Gatsby REPL documentation.

Disabling colored output

In addition to the explicit `--no-color` option, the CLI respects the presence of the `NO_COLOR` environment variable (see no-color.org).

How to change your default package manager for your next project?

When you use `gatsby new` for the first time to create a new project, you are asked to choose your default package manager between yarn and npm.

```
Which package manager would you like to use ? › - Use arrow-keys. Return to submit.
❯  yarn
   npm
```

Once you've made your choice, the CLI won't ask for your preference again for any subsequent project.

If you want to change this for your next project you have to edit the config file created automatically by the CLI. This file is available on your system at: `~/.config/gatsby/config.json`

In it you're going to see something like this.

```
{
  "cli": {
    "packageManager": "yarn"
  }
}
```

Edit your `packageManager` value, save and you’re good to go for your next project using `gatsby new`.