gatsby-cli

The Gatsby command line interface (CLI). It is used to perform common functionality, such as creating a Gatsby application based on a starter, spinning up a hot-reloading local development server, and more!

Lets you create new Gatsby apps using Gatsby starters. It also lets you run commands on sites. The tool runs code from the `gatsby` package installed locally.

The Gatsby CLI (`gatsby-cli`) is packaged as an executable that can be used globally. 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 we want to make the `gatsby develop` command available in our application, we would open up `package.json` and add a script like so:

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

CLI Commands

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

1. new
2. develop
3. build
4. serve
5. clean
6. plugin
7. info
8. repl

`new`

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

Argument	Description
site-name	Your Gatsby site name, which is also used to create the 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
```

- If you leave out both of the arguments, the CLI will run an interactive shell asking for these inputs:

```
gatsby new
? What is your project called? › my-gatsby-project
? What starter would you like to use? › - Use arrow-keys. Return to submit.
❯  gatsby-starter-default
   gatsby-starter-hello-world
   gatsby-starter-blog
   (Use a different starter)
```

See the Gatsby starters docs for more details.

`develop`

At the root of a Gatsby app run `gatsby develop` to start the Gatsby development server.

Options

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

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

`build`

At the root of a Gatsby app run `gatsby build` to do a production build of a site.

Options

Option	Description	Default
`--prefix-paths`	Build site with link paths prefixed (set pathPrefix in your config)	`env.PREFIX_PATHS` or `false`
`--no-uglify`	Build site without uglifying JS bundles (for debugging)	`false`
`--profile`	Build site with react profiling. See https://www.gatsbyjs.com/docs/profiling-site-performance-with-react-profiler/	`false`

`--open-tracing-config-file`	Tracer configuration file (OpenTracing compatible). See https://www.gatsbyjs.com/docs/performance-tracing/	
`--no-color`, `--no-colors`	Disables colored terminal output	`false`

For prefixing paths, most will want to use the CLI flag (`gatsby build --prefix-paths`). For environments where you can't pass the --prefix-paths flag (ie Gatsby Cloud), the environment variable `PREFIX_PATHS` can be set to `true` to provide another way to prefix paths.

`serve`

At the root of a Gatsby app run `gatsby serve` to serve the production build of the site

Options

Option	Description	Default
`-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).	`env.PREFIX_PATHS` or `false`

For prefixing paths, most will want to use the CLI flag (`gatsby build --prefix-paths`). For environments where you can't pass the --prefix-paths flag (ie Gatsby Cloud), the environment variable `PREFIX_PATHS` can be set to `true` to provide another way to prefix paths.

`clean`

At the root of a Gatsby app run `gatsby clean` to wipe out the cache (`.cache` folder) and `public` directories. 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.

`info`

At the root of a Gatsby site run `gatsby info` to get helpful environment information which will be required when reporting a bug.

Options

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

`repl`

Get a node repl with context of Gatsby environment