



[Docusaurus](#)



[↑ Guides](#) Using Plugins

Version: 2.4.0

On this page

## Using Plugins

The Docusaurus core doesn't provide any feature of its own. All features are delegated to individual plugins: the [docs](#) feature provided by the [docs plugin](#); the [blog](#) feature provided by the [blog plugin](#); or individual [pages](#) provided by the [pages plugin](#). If there are no plugins installed, the site won't contain any routes.

You may not need to install common plugins one-by-one though: they can be distributed as a bundle in a [preset](#). For most users, plugins are configured through the preset configuration.

We maintain a [list of official plugins](#), but the community has also created some [unofficial plugins](#). If you want to add any feature: autogenerating doc pages, executing custom scripts, integrating other services... be sure to check out the list: someone may have implemented it for you!

If you are feeling energetic, you can also read [the plugin guide](#) or [plugin method references](#) for how to make a plugin yourself.

## Installing a plugin

A plugin is usually an npm package, so you install them like other npm packages using npm.

npm

Yarn

pnpm

```
npm install --save docusaurus-plugin-name
```

Then you add it in your site's `docusaurus.config.js`'s `plugins` option:

```
docusaurus.config.js
module.exports = {
  // ...
  plugins: ['@docusaurus/plugin-content-pages'],
};
```

Docusaurus can also load plugins from your local directory, with something like the following:

```
docusaurus.config.js
module.exports = {
  // ...
  plugins: ['./src/plugins/docusaurus-local-plugin'],
};
```

Paths should be absolute or relative to the config file.

## Configuring plugins

For the most basic usage of plugins, you can provide just the plugin name or the path to the plugin.

However, plugins can have options specified by wrapping the name and an options object in a two-member tuple inside your config. This style is usually called "Babel Style".

```
docusaurus.config.js
```

```
module.exports = {
  // ...
  plugins: [
    [
      '@docusaurus/plugin-xxx',
      {
        /* options */
      },
    ],
  ],
};
```

Example:

```
docusaurus.config.js
module.exports = {
  plugins: [
    // Basic usage.
    '@docusaurus/plugin-debug',

    // With options object (babel style)
    [
      '@docusaurus/plugin-sitemap',
      {
        changefreq: 'weekly',
      },
    ],
  ],
};
```

## Multi-instance plugins and plugin IDs

All Docusaurus content plugins can support multiple plugin instances. For example, it may be useful to have [multiple docs plugin instances](#) or [multiple blogs](#). It is required to assign a unique ID to each plugin instance, and by default, the plugin ID is `default`.

```
docusaurus.config.js
module.exports = {
  plugins: [
    [
      '@docusaurus/plugin-content-docs',
      {
        id: 'docs-1',
        // other options
      },
    ],
    [
      '@docusaurus/plugin-content-docs',
      {
        id: 'docs-2',
        // other options
      },
    ],
  ],
};
```

 NOTE

At most one plugin instance can be the "default plugin instance", by omitting the `id` attribute, or using `id: 'default'`.

## Using themes

Themes are loaded in the exact same way as plugins. From the consumer perspective, the `themes` and `plugins` entries are interchangeable when installing and configuring a plugin. The only nuance is that themes are loaded after plugins, and it's possible for [a theme to override a plugin's default theme components](#).

 TIP

The `themes` and `plugins` options lead to different [shorthand resolutions](#), so if you want to take advantage of shorthands, be sure to use the right entry!

```
docusaurus.config.js
module.exports = {
  // ...
  themes: ['@docusaurus/theme-classic', '@docusaurus/theme-live-codeblock'],
};
```

## Using presets

Presets are bundles of plugins and themes. For example, instead of letting you register and configure `@docusaurus/plugin-content-docs`, `@docusaurus/plugin-content-blog`, etc. one after the other in the config file, we have `@docusaurus/preset-classic` preset allows you to configure them in one centralized place.

### `@docusaurus/preset-classic`

The classic preset is shipped by default to new Docusaurus websites created with [create-docusaurus](#). It contains the following themes and plugins:

- [@docusaurus/theme-classic](#)
- [@docusaurus/theme-search-algolia](#)
- [@docusaurus/plugin-content-docs](#)
- [@docusaurus/plugin-content-blog](#)
- [@docusaurus/plugin-content-pages](#)
- [@docusaurus/plugin-debug](#)
- [@docusaurus/plugin-google-gtag](#)
- [@docusaurus/plugin-google-tag-manager](#)
- [@docusaurus/plugin-google-analytics](#) (deprecated)
- [@docusaurus/plugin-sitemap](#)

The classic preset will relay each option entry to the respective plugin/theme.

#### `docusaurus.config.js`

```
module.exports = {
  presets: [
    [
      '@docusaurus/preset-classic',
      {
        // Debug defaults to true in dev, false in prod
        debug: undefined,
        // Will be passed to @docusaurus/theme-classic.
        theme: {
          customCss: [require.resolve('./src/css/custom.css')],
        },
        // Will be passed to @docusaurus/plugin-content-docs (false to disable)
        docs: {},
        // Will be passed to @docusaurus/plugin-content-blog (false to disable)
        blog: {},
        // Will be passed to @docusaurus/plugin-content-pages (false to disable)
        pages: {},
        // Will be passed to @docusaurus/plugin-content-sitemap (false to disable)
        sitemap: {},
        // Will be passed to @docusaurus/plugin-google-gtag (only enabled when explicitly specified)
        gtag: {},
        // Will be passed to @docusaurus/plugin-google-tag-manager (only enabled when explicitly specified)
        googleTagManager: {},
        // DEPRECATED: Will be passed to @docusaurus/plugin-google-analytics (only enabled when explicitly sp
        googleAnalytics: {},
      },
    ],
  ],
};
```

## Installing presets

A preset is usually an npm package, so you install them like other npm packages using npm.

npm

Yarn

pnpm

```
npm install --save @docusaurus/preset-classic
```

Then, add it in your site's `docusaurus.config.js`'s `presets` option:

#### `docusaurus.config.js`

```
module.exports = {
  // ...
  presets: ['@docusaurus/preset-classic'],
};
```

Preset paths can be relative to the config file:

```
docusaurus.config.js
module.exports = {
  // ...
  presets: ['./src/presets/docusaurus-local-preset'],
};
```

## Creating presets

A preset is a function with the same shape as the [plugin constructor](#). It should return an object of `{ plugins: PluginConfig[], themes: PluginConfig[] }`, in the same as how they are accepted in the site config. For example, you can specify a preset that includes the following themes and plugins:

```
src/presets/docusaurus-preset-multi-docs.js
module.exports = function preset(context, opts = {}) {
  return {
    themes: [['docusaurus-theme-awesome', opts.theme]],
    plugins: [
      // Using three docs plugins at the same time!
      // Assigning a unique ID for each without asking the user to do it
      ['@docusaurus/plugin-content-docs', {...opts.docs1, id: 'docs1'}],
      ['@docusaurus/plugin-content-docs', {...opts.docs2, id: 'docs2'}],
      ['@docusaurus/plugin-content-docs', {...opts.docs3, id: 'docs3'}],
    ],
  };
};
```

Then in your Docusaurus config, you may configure the preset:

```
docusaurus.config.js
module.exports = {
  presets: [
    [
      './src/presets/docusaurus-preset-multi-docs.js',
      {
        theme: {hello: 'world'},
        docs1: {path: '/docs'},
        docs2: {path: '/community'},
        docs3: {path: '/api'},
      },
    ],
  ],
};
```

This is equivalent of doing:

```
docusaurus.config.js
module.exports = {
  themes: [['docusaurus-theme-awesome', {hello: 'world'}]],
  plugins: [
    ['@docusaurus/plugin-content-docs', {id: 'docs1', path: '/docs'}],
    ['@docusaurus/plugin-content-docs', {id: 'docs2', path: '/community'}],
    ['@docusaurus/plugin-content-docs', {id: 'docs3', path: '/api'}],
  ],
};
```

This is especially useful when some plugins and themes are intended to be used together. You can even link their options together, e.g. pass one option to multiple plugins.

## Module shorthands

Docusaurus supports shorthands for plugins, themes, and presets. When it sees a plugin/theme/preset name, it tries to load one of the following, in that order:

- `[name]` (like `content-docs`)
- `@docusaurus/[moduleType]-[name]` (like `@docusaurus/plugin-content-docs`)
- `docusaurus-[moduleType]-[name]` (like `docusaurus-plugin-content-docs`)

where `moduleType` is one of `'preset'`, `'theme'`, `'plugin'`, depending on which field the module name is declared in. The first module name that's successfully found is loaded.

If the name is scoped (beginning with `@`), the name is first split into scope and package name by the first slash:

```

@scope
^-----^
  scope  (no name!)

@scope/awesome
^-----^ ^-----^
  scope   name

@scope/awesome/main
^-----^ ^-----^
  scope   name

```

If there is no name (like `@jquery`), `[scope]/docusaurus-[moduleType]` (i.e. `@jquery/docusaurus-plugin`) is loaded. Otherwise, the following are attempted:

- `[scope]/[name]` (like `@jquery/content-docs`)
- `[scope]/docusaurus-[moduleType]-[name]` (like `@jquery/docusaurus-plugin-content-docs`)

Below are some examples, for a plugin registered in the `plugins` field. Note that unlike [ESLint](#) or [Babel](#) where a consistent naming convention for plugins is mandated, Docusaurus permits greater naming freedom, so the resolutions are not certain, but follows the priority defined above.

Declaration	May be resolved as
<code>awesome</code>	<code>docusaurus-plugin-awesome</code>
<code>sitemap</code>	<a href="#">@docusaurus/plugin-sitemap</a>
<code>@my-company</code>	<code>@my-company/docusaurus-plugin</code> (the only possible resolution!)
<code>@my-company/awesome</code>	<code>@my-company/docusaurus-plugin-awesome</code>
<code>@my-company/awesome/web</code>	<code>@my-company/docusaurus-plugin-awesome/web</code>

[✎ Edit this page](#)

Last updated on **Mar 23, 2023** by *Sébastien Lorber*

[Previous](#)

[« SEO](#)

[Next](#)  
[Deployment »](#)

Learn

[Introduction](#)

[Installation](#)

[Migration from v1 to v2](#)

Community

[Stack Overflow](#) [↗](#)

[Feature Requests](#)

[Discord](#) [↗](#)

[Help](#)

More

[Blog](#)

[Changelog](#)

[GitHub](#) [↗](#)

[Twitter](#) [↗](#)



Legal

[Privacy](#) 

[Terms](#) 

[Data Policy](#) 

[Cookie Policy](#) 

Copyright © 2023 Meta Platforms, Inc. Built with Docusaurus.