

Hacking BookStack



- [BookStack Hacks Directory](#)
 - [BookStack API](#)
 - [Custom HTML Head Option](#)
 - [Visual Theme System](#)
 - [Logical Theme System](#)
 - [BookStack JavaScript Public Events](#)
 - [Tag Classes](#)
 - [Export Classes](#)
-

Sometimes you may want to perform deeper customisation to BookStack or extend the system to suit your use-case. The core of BookStack is fairly rigid as it's intended to be a configured, ready-to-use system out of the box but there are a few advanced options for performing more advanced modifications without needing to alter the system code-base.

Note: Customisation options on this page are not deemed to be stable or officially supported. BookStack core files may change on any release causing changes in behaviour to your hacks.

BookStack Hacks Directory

This website has a directory of available hacks [which can be found here](#). These make use of the [Custom HTML Head Option](#), [Visual Theme System](#) and [Logical Theme System](#) options listed below.

BookStack API

BookStack has a built-in REST API for external interaction and consumption of your BookStack data. API documentation can be found within your BookStack instance at the `/api/docs` URL path. You'll need to have the 'Access system API' role permission to access the API.

Reference Links

- [API documentation of our demo instance.](#)
 - [Our "BookStack API Scripts" repo containing examples.](#)
-

Custom HTML Head Option

Within the "Settings > Customization" view within BookStack you'll find a "Custom HTML Head Content" setting. You can use this to add in any custom JavaScript or CSS content which enables you to override default BookStack functionality and styles.

You can find examples of custom HTML Head customizations on the [hacks part of this site](#).

Visual Theme System

BookStack allows visual customization via the theme system which enables you to extensively customize views, translation text & icons. Documentation for this system is contained within [the project repo here](#).

Note: The files that can be override using the theme system are not deemed to be stable. BookStack core files may change on any release causing changes in behaviour to your overrides. Theme overrides are not officially supported in any way.

You can find examples visual theme system customizations on the [hacks part of this site](#).

Logical Theme System

BookStack allows PHP code-based extension via what we call the "Logical Theme System". This works by hooking into specific events where you can then perform custom actions or extension of the underlying framework. Documentation for this system is contained within [the project repo here](#).

Note: Only the API described in the logical-theme-system document is considered stable & supported. Any usage of other application classes is regarded as unstable and unsupported.

You can find examples of logical theme system customizations on the [hacks part of this site](#).

BookStack JavaScript Public Events

To allow customization of the JavaScript libraries & components used, BookStack emits events as part of their lifecycle so that you can define listeners to hook into these events. These can often be used by adding scripts to the [custom HTML head content option](#).

You can find details and a listing of all events [within the project repo here](#).

Tag Classes

While primarily for categorization, tags within BookStack can also provide opportunities for customization. When viewing an item with tags applied, those tags will be normalized to CSS classes and applied to the `<body>` element of the page. As an example, a tag name/value pair of `Priority: Critical` will apply the following classes to the body:

- tag-name-priority
- tag-value-critical
- tag-pair-priority-critical

For the normalization to classes, the following is done to both the name and value of the tag:

- Text is lower-cased
- Spaces and hyphens are stripped

This normalization provides relatively stable class naming for CSS targeting, but note it does mean that two tags, of different values, could be normalized to the same CSS class names in some scenarios.

As an example of usage, pages with the tag `Priority: Critical` could have their text made red with the following "Custom HTML Head Content" setting value:

```
<style>
.tag-pair-priority-critical .page-content { color: red; }
</style>
```

Export Classes

When PDF or HTML exports are performed in BookStack, the underlying templates define classes to allow customization of styling in specific scenarios. For context, PDF exports are rendered via a conversion from HTML to PDF, so CSS styling can be applied for these but support may depend on

the underlying PDF conversion engine. The classes are applied to the `<body>` element, and are as follows:

- `export` - All HTML/PDF exports
- `export-format-pdf` - PDF exports
- `export-format-html` - HTML exports
- `export-engine-dumpdf` - PDF exports using the default DomPDF rendering engine
- `export-engine-wkhtml` - PDF exports using the default DomPDF rendering engine
- `export-engine-command` - PDF exports using the command-based PDF rendering option

As an example usage, you could define the following custom style code to make paragraph text red only in PDF exports created via the default DomPDF renderer:

```
<style>
.export-format-pdf.export-engine-dumpdf p { color: red; }
</style>
```

📄 #2

📄 cocobook-sync 📄 2026-10-28 10:56:10 UTC

📄 cocobook-sync 📄 2026-10-28 11:07:00 UTC