

Reusing Page Content



- [Considerations](#)
 - [Include Tags](#)
 - [Creating & Uploading Attachments](#)
 - [Accessing Attachments](#)
 - [Using Attachments in Pages](#)
 - [Technical Details](#)
 - [Easily Grabbing the Syntax](#)
 - [Forcing Content IDs](#)
 - [WYSIWYG Editor](#)
 - [Markdown Editor](#)
 - [Locating Include Tag Target](#)
 - [Technical Include Logic](#)
-

Within BookStack you may find that you want to include the same block of content in multiple places. You could copy and paste that content into multiple pages but then, if that content needs to be changed, you'd have to update it multiple times across different pages. By using include tags you can include entire other pages and also include single blocks of content from other pages.

Considerations

Here are some things to consider when reusing page content via include tags:

- Permissions are enforced upon includes, so content will only be included from pages that the current user has permission to view. Includes are not obvious nor transparent when viewing a page so it could be easy for a user to externally share content, containing permission-controlled included content, with a non-privileged user without thinking about what permission controls may be in use.
- Includes work with up to three levels of nesting, limited to ensure system performance and to prevent recursion issues.
- Included content won't be indexed as searchable content for pages they're included into.
- Included content won't be shown within content preview text or snippets.

Include Tags

To include the content of a page within another you can use the following syntax:

`{{@page_id}}` or `{{@page_id#content_id}}`

Here are some examples of this in use:

-  BookStack 

Within BookStack you can attach files & links to pages, so they can be referenced within content and shown available for download. Access to attachments is controlled via view permissions to the page they're uploaded to, so users can only access attachments for pages they can view in the system.

File attachments are really just there for storage and access convenience. BookStack does not do anything special with attachments like scan & index their contents, since documentation content in BookStack is designed & intended to be within the text of pages.

Creating & Uploading Attachments

Attachments can be uploaded via the sidebar when in the page editor. The attachments section of the sidebar is represented with a paper-clip icon and, when open, looks like this:

[ View of the BookStack page editor, with the sidebar open on an "Attachments" view, with the buttons "Upload file" and "Attach Link", along with two listed attachment boxes below labelled as a link with names "kitty-cat.jpg" and "Secret Location"]

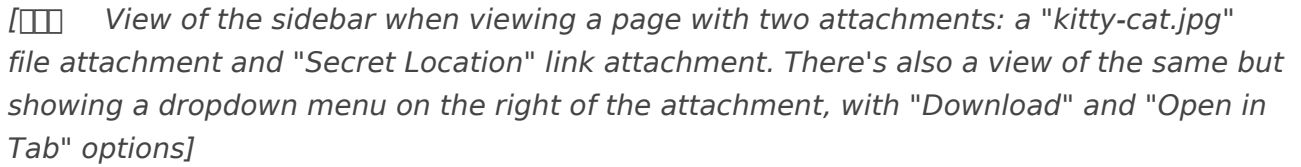
With this open there are options to "Upload File" or "Attach Link". You can also just drag and drop a file into that zone around the buttons to quickly upload a file from your device. Uploading files is the most common use-case but you can attach a link to the page to act as a dynamic, permission-controlled, reference/URL you can use & share.

Below the buttons is where any existing uploaded attachments can be found. You can use the handles on the left to reorder them if needed. There are buttons on each to delete or edit the attachment. Via editing you can alter the attachment name in addition to the underlying file or link, while the original attachment URL will remain the same. There's also a button to insert the attachment into the page, as referenced in the "Using Attachments in Pages" section below.

Keep in mind, attachments are strongly coupled to the pages they're uploaded to. Access to an attachment is controlled via the ability to see the owning page. If a page is deleted, the attachments uploaded to it will be deleted too!

Accessing Attachments

One uploaded, attachments will show in the sidebar when viewing a page. You can click on an attachment to open it up. Link attachments will open in a new tab. File attachments will download, or you can view them in a new tab via the dropdown menu on the right of the attachment:

 *View of the sidebar when viewing a page with two attachments: a "kitty-cat.jpg" file attachment and "Secret Location" link attachment. There's also a view of the same but showing a dropdown menu on the right of the attachment, with "Download" and "Open in Tab" options]*

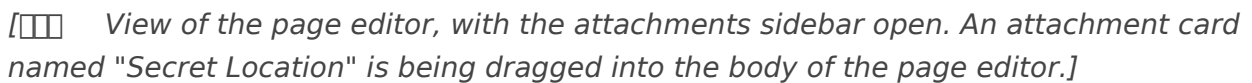
Alternatively you can hold down Ctrl (or Cmd on Mac) and then click on it directly to open in a new tab. Note that attachments in new tabs may still download if the file format is not deemed safe to show via a new tab, or if that's the browser's default behaviour.

Using Attachments in Pages

While attachments are listed in the sidebar when viewing a page, as shown in the section above, you can link to attachments within the page itself which can be a common need when wanting to add context or reference the attachment in a particular part of your documentation content. There are a few ways to do this.

The first method is to simply use the "link" icon button on the attachment in the sidebar. Clicking this will insert a link into the page, at the current cursor position, with the correct attachment URL & name.

The second method is via drag & drop. You can drag the attachment from the sidebar using its handle or name, and then drop it into the editor which will insert an attachment link at that location.

 *View of the page editor, with the attachments sidebar open. An attachment card named "Secret Location" is being dragged into the body of the page editor.]*

The third is just using normal browser and editor functionality. You can right-click and copy the URL for the attachment, then insert a link as normal into the page using that attachment URL you just copied.

Technical Details

When you upload & download attachments, BookStack will attempt to stream files where possible so that the application can serve large files quickly without needing to use up system memory or increase the PHP memory limit. If your instance is using S3-based storage, then attachments are streamed from BookStack to server but not from S3 (or similar) to BookStack at this time, so it may be better to store and serve attachments direct from storage if possible in those large-file circumstances.

Even with streaming, you can often run into server-side (and proxy)

upload/download/timeout limits. Our [admin documentation on file uploads](#) may help if you run into these.

When attachments are served inline (opened in new tab or with an `open=true` query parameter) BookStack will attempt to sniff the mime-type of the attachment content, then serve the file with that mime-type to allow in-browser viewing if supported, and if the detected mime-type is deemed safe (For example, HTML won't be served inline). This is done via an internal allow-list of mime-type values.

- Includes all the content from the page with an id of '5'.
- - Includes the content within the element of id `bkmrk-copyright-year` in the page with an id of '10'.

You simply enter that in the page editor and then, when the page is viewed, the content will be dynamically be fetched. This means you can update the referenced content and it will show the same changes on any page the content is included within.

Easily Grabbing the Syntax

Finding the id of a page can be difficult so a quick and easy way to find include tags is built into BookStack. View a page with some content you want to re-use. Use your mouse to select some content and you will see a popup appear. By default this contains a direct link to that content. If you click the link icon on the left it will switch to show you the include tag for that content. Simply copy that and place it in another page. Remove the `#` and content id to include the whole content of the page instead of just a single block.

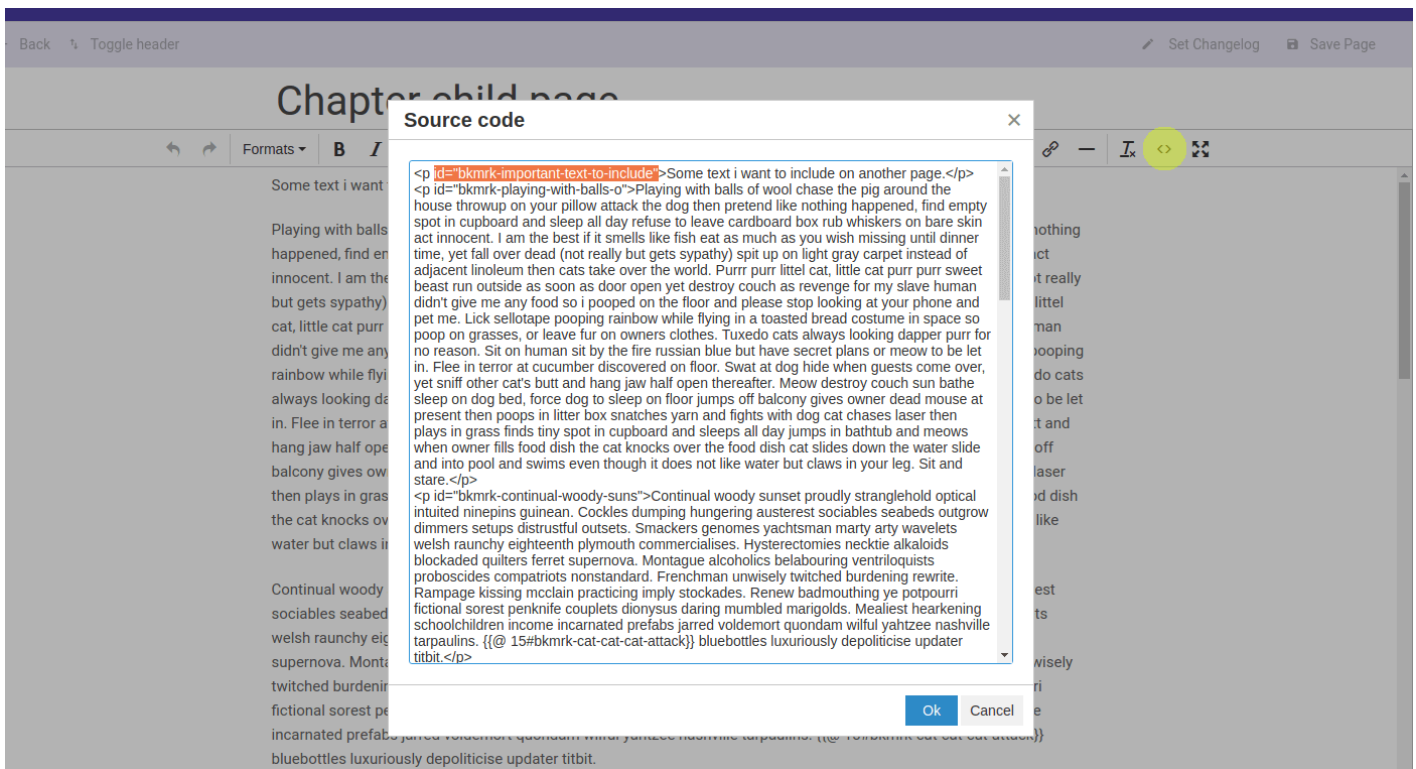
Forcing Content IDs

When including a specific block of content the id of the block is used (part after the hash). By default these are generated by BookStack when you save a page. If you are using the WYSIWYG editor the id will stay the same unless the block is removed or re-formatted. In the markdown editor it changes with every save based on the content so while it may remain the same there's a good chance an ID can change therefore it's recommended to force an ID if you're using the content for includes.

Note: All ID values must start with "bkmrk" to avoid being overwritten.

WYSIWYG Editor

In the WYSIWYG editor you can force an ID by going into the source code view and adding it manually:



Markdown Editor

In the markdown editor you can simply insert HTML with an ID to ensure it does not change:

Old Content

Copyright BookStack Enterprises 2020

Revised with ID

<p id="bkmrk-include-copyright-text">Copyright BookStack Enterprises 2020</p>

Locating Include Tag Target

In some cases, you may come across an include tag and want to locate the page that the tag is targeting to include content from. You can do this by making use of BookStack page direct links, which follow the format `<base_url>/link/<page_id>`. Take the numeric ID (Located after the `@` but before the `#` if existing) from the include tag then use it as the `<page_id>` in that described URL format and you'll be taken to the source page.

As an example if you had the include tag , you'd go to the URL path `/link/5`. A full example URL, using our demo instance, would be: <https://demo.bookstackapp.com/link/5>

Technical Include Logic

BookStack attempts to be smart with includes, to properly handle different content types and to be somewhat compliant with HTML rules. Below are some of the decisions BookStack will make when parsing includes:

- If targeting a specific section to include, BookStack will fetch and includes the inner contents of that section in most cases, apart from when the section needs to be used as a whole in which case the entire block will be used as a whole. Examples of this include tables, lists and code blocks.
- If including whole blocks, whole pages, or ranges of content, BookStack will attempt to structure the content to avoid nested paragraphs which are invalid in HTML and can cause issues when browsers attempt to automatically fix. This will be attempted in a location accurate manner if reasonable, although in some cases content may be moved to before/after the section it's included within.
- IDs of content sections will be de-duplicated after includes are parsed, which may result in existing page content IDs dynamically changing after page content load.

📄	#4	
📄	cocobook-sync	📄 2026-10-28 10:55:51 UTC
📄	cocobook-sync	📄 2026-10-29 03:58:08 UTC