PDF and Markdown Export

What the PDF and Markdown exports add on top of the raw content, and how to bundle a whole section into one file.

Anonymous · 21/08/2026

PDF and Markdown export

Every content item (Academy, Blog, Docs, FAQ, Pages) renders as HTML, Markdown, PDF or JSON — format negotiated via the `Accept` header, or a `.md`/`.pdf`/`.json` suffix on the URL. HTML is the normal browsing experience and JSON is the same shape used by API/MCP; this page covers what is specific to the PDF and Markdown exports.

Root-relative URLs are resolved to absolute ones

A content body written for the website itself references images and internal links with root-relative paths (`/img/foo.png`, `/docs/other-page`). Neither export has a "current page" to resolve a root-relative path against, so both make the body self-contained wherever it ends up — an LLM's context window, someone else's Markdown viewer, a PDF saved to disk:

- **Markdown**: both images and internal links are rewritten to absolute URLs.
- **PDF**: only links are rewritten. Images are left root-relative on purpose — the PDF engine resolves them from the local filesystem instead of fetching them over HTTP, which is slower and, on a single-worker server, can tie up the whole process waiting on a request back to itself.

Video and quiz are shown, not just left in the frontmatter

A frontmatter `video` (see Content frontmatter) is a plain URL in YAML that neither export would otherwise surface anywhere a reader (or an LLM) would notice it:

- **PDF**: a clickable YouTube thumbnail linking to the watch page, or a plain link for a non-YouTube URL.
- **Markdown**: the same thumbnail and link, as a Markdown image wrapped in a link.

An Academy lesson's `test` (quiz) is rendered in full — every option, not only the correct one, plus the explanation, since the wrong options and the explanation are real information (useful to know what an answer is commonly confused with, or why it's wrong):

- **PDF**: the questions with checkbox-style options first (to answer on paper), then a dedicated "Answers" section on its own page, with the correct option(s) checked and the explanation for

each question.
- **Markdown**: each question as a GFM task list (`- [x]`/`- [ ]`, the correct option(s) checked), followed by the explanation as a blockquote.

Bundling a whole section: `?full=1`

Content types with children — a Docs or FAQ section, an Academy course or module — accept `?full=1` on their `.pdf`/`.md` URL to bundle the whole subtree into a single file, instead of just the current item with a short list of links to its children:

- **PDF**: a cover page, a real table of contents with page numbers, then every descendant on its own page (Docs/FAQ children recursively; an Academy course's modules and, nested under each, its lessons).
- **Markdown**: every descendant's title and full body appended after the current item's, with heading levels reflecting how deep it is nested.

`?full=1` on an item with no children is a no-op: it falls through to the regular, single-item export.

Download buttons on the HTML view

Every content page shows small, always-visible buttons for its PDF and Markdown exports, next to the tags/last-updated line — plain buttons, not a dropdown, so there is nothing to discover before they can be used:

- **PDF** opens inline: browsers render it in their own viewer, which already has its own save/print controls.
- **Markdown** downloads instead of opening inline: a browser has no native Markdown renderer, so opening it in a tab just shows unstyled raw text. A "Copy" button next to it copies the Markdown source straight to the clipboard — handy for pasting into an LLM — and confirms with a Notyf toast when the site has it loaded, falling back to a brief change of the button's own text otherwise.

"Full" (and "Full Copy") variants of these buttons appear next to the regular ones whenever the item actually has children to bundle.

Last updated on 21/08/2026