

Table of Contents

1. prodockit overview	2
1.1 What the features bring	2
1.1.1 Structure that stays connected	2
1.1.2 Evidence and publication references	3
1.1.3 Tables with the layout the information needs	3
1.1.4 Terminology readers can follow	3
1.1.5 One source for reading and submission	3
1.2 Project status	4
1.3 Support prodockit	4

1. proudockit overview

This section is for anyone new to proudockit. It explains what the package adds to Zensical, how to install it, and how to build a first local site before you choose authoring or publishing features.

proudockit adds the document features that professional and academic Zensical projects ([Zensical website](#)) commonly need: numbered sections, cross-references, citations, glossaries, richer tables, and a printable PDF. Install one Python package, then enable only the parts your project uses.

[Figure 1.1](#) shows how the shared authoring input becomes a website and a matching printable document.

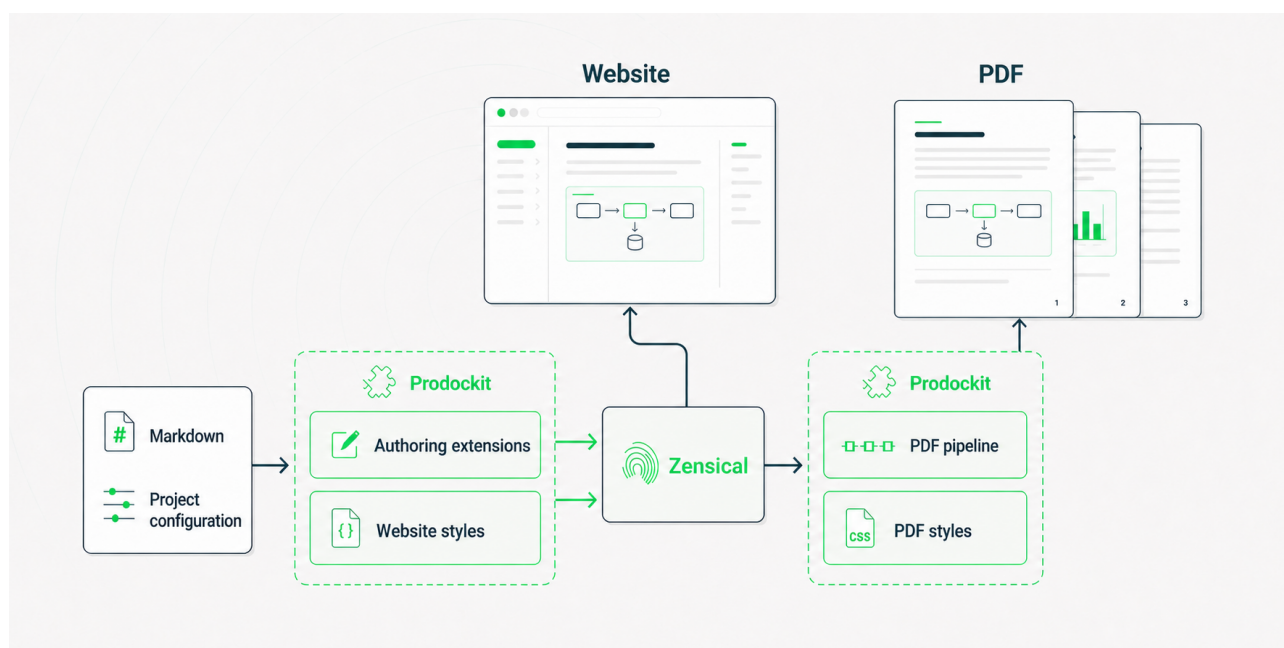


Figure 1.1. Relationship between Proudockit's source, components and outputs

1.1 What the features bring

Proudockit's features remove the points where basic Markdown stops being enough for a structured academic or professional document. They preserve the speed and readability of Markdown source while adding controls that would otherwise require handwritten HTML, repeated manual edits, or a separate print document.

1.1.1 Structure that stays connected

Numbered headings give a long document a visible hierarchy. Cross-references connect prose to headings, figures and tables by identity, so their displayed number and title remain correct when material moves. Together, `proudockit.headings` and `proudockit.refs` replace manually typed references such as "see section 4.2," which silently become wrong as a document develops.

Numbered steps provide a distinct structure for procedures, methods and instructions. Directory trees present file layouts as relationships rather than as hard-to-read lists of paths. Both make technical sequences easier to scan without forcing the author to maintain numbering or connector lines.

1.1.2 Evidence and publication references

Citations and bibliographies provide the evidence trail expected in academic and professional publication, with a choice of implementation:

- `prodockit.citations` supports a small, hand-written reference list kept directly in Markdown. It suits a document whose author wants explicit control without maintaining a separate bibliography database.
- `prodockit.bibliography` uses BibTeX or BibLaTeX data and a CSL style through Pandoc. It suits larger or formally styled bibliographies, reusable source records, and publications that must follow a specified citation style.
- `prodockit.refs` handles internal references to the document's own sections, figures and tables. It complements either citation approach rather than replacing it.

The approaches can coexist, so an author can use the lighter hand-written form where it is sufficient and the database-backed workflow where scale or a publication style requires it.

1.1.3 Tables with the layout the information needs

Basic Markdown tables are intentionally limited. `prodockit.tables` lets an author choose widths, merge cells, span headings across columns, rotate dense headings and use compact presentation. The information therefore determines the table format instead of being squeezed into the small set of layouts that plain Markdown happens to support. The same authored table remains aligned in the website and PDF.

1.1.4 Terminology readers can follow

`prodockit.glossary` manages acronyms and specialist terms from shared definitions. Authors define a term once and use it consistently; readers get the expanded form and a glossary without the author repeatedly explaining or manually synchronising terminology throughout the document.

For the printable document, `prodockit.index` turns selected concepts and cross-references into a back-of-book index. It gives readers another way to find related discussion when the terminology spans several chapters.

1.1.5 One source for reading and submission

The PDF command, website macros and shared styles extend the authoring features into a complete publication workflow. The website remains useful for navigation and review, while the PDF provides a controlled printable or submittable document from the same source. Testing support checks both outputs so links, headings and required content are verified rather than assumed.

Continue with [Choose your install](#) to decide whether Adoption, Bootstrap, or Manual installation matches the document you have.

If you want to see it working before reading the reference pages, follow [Build your first site](#). It starts with a new Zensical project and ends at a live local preview.

1.2 Project status

prodockit is early but functional. All nine extensions, the PDF and source bundle builders, website macros, testing support, and project-management commands are implemented and tested. See [Support and compatibility](#) for maturity, PyMdown Blocks and other required versions, platform coverage, and known constraints. Read the [release notes](#) before upgrading.

1.3 Support prodockit

prodockit is free and open-source software. If it helps your work and you would like to support its continued development, you can buy the maintainer a coffee. Your contribution helps fund the software and online services used to develop, test, and publish prodockit, as well as the time spent maintaining the project and developing new features.

[Buy me a coffee](#)