

Table of Contents

- [1. Installation](#)
 - [1.1 From PyPI](#)
 - [1.2 Enabling an extension](#)
 - [1.3 Development install](#)

1. Installation

1.1 From PyPI

```
pip install prodockit
```

`prodockit` depends on [Markdown](#) (≥ 3.4), [zensical](#), and [beautifulsoup4](#) (≥ 4.12 , used by `prodockit.pdf` - see [PDF generation](#)).

1.2 Enabling an extension

Each `prodockit` extension is registered as a standard Python-Markdown extension under the `markdown.extensions` entry point group, so it can be enabled by name, the same way you'd enable a built-in extension like `toc` or a `pymdownx` one:

```
import markdown

html = markdown.markdown(
    text,
    extensions=["prodockit.headings", "prodockit.refs",
               "prodockit.citations", "prodockit.glossary"],
)
```

Or, for a [Zensical](#) project, in `zensical.toml` alongside the built-in and `pymdownx` extensions. Unlike `pymdownx`'s and Zensical's own namespaces, Zensical doesn't hoist a nested `prodockit.headings` table into that dotted extension name, so each one needs a quoted key instead:

```
[project.markdown_extensions."prodockit.headings"]
[project.markdown_extensions."prodockit.refs"]
[project.markdown_extensions."prodockit.citations"]
[project.markdown_extensions."prodockit.glossary"]
```

See each extension's own page for its options and for how to share a registry across multiple pages of a site build:

- [prodockit.headings](#)
- [prodockit.refs](#)
- [prodockit.citations](#)
- [prodockit.glossary](#)

`prodockit.pdf` is different: it isn't a Python-Markdown extension (no

`markdown.extensions` entry point, nothing to add to `zensical.toml`) - it's a plain function library for a separate PDF-generation build step. See [PDF generation](#) for how it's used.

1.3 Development install

```
git clone https://github.com/buckwem/prodockit-extensions
cd prodockit-extensions
python -m venv .venv
source .venv/bin/activate
pip install -e ".[dev]"
pytest
```

`zensical` is a core dependency, so `zensical serve` is available as soon as `pip install -e ".[dev]"` finishes - no extra step needed to build these docs locally.