

base-template

A base template for CI/CD workflows with MkDocs and Semantic Release

Usage

Create a new repository

1. Click the `Use this template` button
2. Enter a name for your repository
3. Click `Create repository from template`

Add secrets for GitHub Actions

1. Go to the `Settings` tab of your repository
2. Click `Secrets` in the left sidebar
3. Click `New repository secret`
4. Add the following secrets:

Name	Value
GITHUB_TOKEN	<code>\${{ secrets.GITHUB_TOKEN }}</code>
NPM_TOKEN	<code>\${{ secrets.NPM_TOKEN }}</code>

Initial release

Push a commit to the `main` branch with the message `feat: initial commit`

GitHub Pages

1. Modify the contents of the `mkdocs.yaml` file
2. Add content to the `docs` folder
3. Push a commit to the `main` branch
4. Wait for the `Publish docs via GitHub Pages` workflow to complete
5. Go to the `Settings` tab of your repository
6. Scroll down to the `pages` section
7. Select `Deploy from a branch` as the source
8. Select `gh-pages` as the branch and `/(root)` as the folder, then click `Save`

Features

- [x] Linting

- [x] Automated Release Draft
- [x] Semantic versioning
- [x] Documentation to PDF
- [x] Github Pages deployment (MkDocs)

Workflows

CI (Continuous Integration)

Lint

The `Lint` workflow is automatically triggered whenever there is push activity in `main` or pull request activity towards `main`. It has one job:

- Lint the codebase with GitHub's [Super-Linter](#).

CD (Continuous Deployment)

Docs to PDF

The `Docs to PDF` workflow is automatically triggered whenever there is push activity in `main` or pull request activity towards `main`. It has one job:

- Build the documentation to PDF with [Markdown to PDF](#)

Release

The `Release` workflow is automatically triggered whenever there is push activity in `main` or pull request activity towards `main`. It has one job:

- Create a release draft with [semantic-release](#)

Publish docs via GitHub Pages

The `Publish docs via GitHub Pages` workflow is automatically triggered whenever there is push activity in `main` or pull request activity towards `main`. It has one job:

- Publish the documentation to GitHub Pages with [MkDocs](#)

Semantic Release

Commit message format

semantic-release uses the commit messages to determine the consumer impact of changes in the codebase. Following formalized conventions for commit messages, **semantic-release** automatically determines the next [semantic version](#) number, generates a changelog and publishes the release.

By default, **semantic-release** uses [Angular Commit Message Conventions](#). The commit message format can be changed with the `preset` or `config` options of the [@semantic-release/commit-analyzer](#) and [@semantic-](#)

[release/release-notes-generator](#) plugins.

Tools such as [commitizen](#) or [commitlint](#) can be used to help contributors and enforce valid commit messages.

The table below shows which commit message gets you which release type when `semantic-release` runs (using the default configuration):

Commit message	Release type
<code>fix(pencil): stop graphite breaking when too much pressure applied</code>	Patch Fix Release
<code>feat(pencil): add 'graphiteWidth' option</code>	Minor Feature Release
<code>perf(pencil): remove graphiteWidth option</code> <code>BREAKING CHANGE: The graphiteWidth option has been removed.</code> <code>The default graphite width of 10mm is always used for performance reasons.</code>	Major Breaking Release (Note that the <code>BREAKING CHANGE:</code> token must be in the footer of the commit)

Automation with CI

`semantic-release` is meant to be executed on the CI environment after every successful build on the release branch. This way no human is directly involved in the release process and the releases are guaranteed to be [unromantic and unsentimental](#).

Triggering a release

For each new commit added to one of the release branches (for example: `main`, `next`, `beta`), with `git push` or by merging a pull request or merging from another branch, a CI build is triggered and runs the `semantic-release` command to make a release if there are codebase changes since the last release that affect the package functionalities.

Derived Templates

This template is used to create the following templates:

- [python-template](#)
- [pypi-template](#)
- [jupyter-book-template](#)

References

- [Semantic Versioning](#)
- [Conventional Commits](#)
- [semantic-release](#)
- [Semantic Release Action](#)

- [Git Semantic Version](#)

License

This project is released under the [MIT License](#).