Development and releasing

You need the following requirements:

- `hatch` for test automation and package dependency managements. If you don't have hatch, you can use `pipx run hatch` to run it without installing.

You can run `hatch env show` to list available environments and scripts.

```
hatch env create
hatch run init

hatch run docs:serve
```

Bump and releasing

To bump version and upload build to test.pypi using:

```
hatch run bump
git push
git push --tags
```

while to update only the CHANGELOG.md file:

```
hatch run ch
```

Release will automatically occur after pushing.

Configuration files

- pre-commit configured in .pre-commit-config.yaml;
- isort (sys) configured in pyproject.toml;
- ruff configured in pyproject.toml (pinned in pre-commit);
- darglint configured in .darglint (pinned in pre-commit);
- codespell configured in .codespellrc (pinned in pre-commit);
- commitizen in pyproject.toml (dev deps and pinned in pre-commit).

While exact dependencies and their versions are specified in `pyproject.toml` file and updated in GitHub with `Dependabot`, a complete list of all the required packages and

their versions (including transitive dependencies) can be generated by pip-deepfreeze in the requirements-[dev,docs,tests].txt files.

`pip` and `hatch` are pinned in .github/workflows/constraints.txt.