

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.

```
Dependencies and their
versions are specified in the pyproject.toml file and updated in
GitHub with Dependabot.
```

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

```
hatch env create
hatch run init # init repo with pre-commit hooks

hatch run lint
hatch run tests.py3.11:all
```

```
Hatch handles everything for you, including setting up an
temporary virtual environment for each run.
```

- `pre-commit` for all style and consistency checking. While you can run it with `nox`, this is such an important tool that it deserves to be installed on its own. If `pre-commit` fails during pushing upstream then stage changes, `Commit Extend` (into previous commit), and repeat pushing.

`pip` and `hatch` are pinned in `.github/workflows/constraints.txt` for consistency with CI/CD.

If you like install `hatch` and required deps in archlinux with:

```
pacman -S python-hatch python-hyperlink python-httpx
```

While `pre-commit` is listed as a 'dev' dependency, you also have the option to install it globally using `pipx`. This can be done with the following command:

```
pipx install pre-commit
```

Setting up a development with direnv

```
echo "layout hatch" > .envrc
hatch run init
```