pipx-in-pipx

Release 1.0.1

Jan 08, 2020

Project Operations

docs	read-the-docs	
linux	static analysis	
linux	CPython 3.7	
windows	CPython 3.7	

pipx is great for keeping your CLI tools isolated and your system Python paths clean. However, it still requires that you install pipx *itself* in your system Python.

But pipx is a CLI tool installed through `pip`... why not install pipx with pipx? Why not indeed!

With `pipx-in-pipx`, all you need to do is install :

```
$ pip install pipx-in-pipx
```

But wait! You say. Didn't you just say that we shouldn't install things to system Python?

Yes. What `pipx-in-pipx` actually does is slightly (but only slightly) evil. Rather than actually installing anything when you run "install", `pipx-in-pipx` instead builds a temporary virtual environment, installs pipx there, and then uses *that* pipx to install pipx in your user local space, just like any other pipx-installed tool.

What you end up with is a pipx installation that is *itself* managed by pipx.

CHAPTER 1

Sharp Edges

1.1 Which Python?

By default, pipx uses its own Python for each environment that it creates. Normally, this would be the system Python, whatever it was when you installed pipx. However, when your are using a `pipx-in-pipx`-installed pipx, the default Python that pipx uses for each environment it creates is instead whatever Python you used to "install" `pipx-in-pipx`.

This has two notable side effects:

1. If you uninstall your pipx-managed pipx, then all of the tools that you installed using that pipx will stop working because their Pythons suddenly point to nothing.
2. If you want to change the Python used by all of your pipx-managed tools, you only need to reinstall one of them (pipx) rather than reinstalling all of them.

1.2 Uninstalling

pipx has a handy feature to uninstall *all* pipx-managed tools. Because you have now made pipx manage itself, running `pipx uninstall-all` *will also* uninstall pipx.

This is not a bug, but a feature. By installing pipx using `pipx-in-pipx`, you have expressed an intent that you *want* pipx to manage itself. If that's not what you want, this is not the tool for you.

If you at any point uninstall your pipx-managed pipx, you can simply `pip install pipx-in-pipx` again to rebuild it.

1.2.1 Versioning

pipipxx releases follow Semantic Versioning.

1.2.2 Changelog

1.0.1 – 2019-10-27

Administrivia

- Rename project from `pipipxx` to `pix-in-pipx`.

Bugfixes

- Fix Windows compatibility. *#13https://github.com/mattsb42/pipx-in-pipx/pull/13*

Maintenance

- Add Windows CI. *#17https://github.com/mattsb42/pipx-in-pipx/pull/17*

1.0.0 – 2019-05-26

Now that I have CI set up for this on at least one platform, I am comfortable saying that it is ready for use.

0.0.1b1 – 2019-05-22

Bugfixes

- Some installs in Linux were failing due to unable to find src files. All logic now in `setup.py`.
- Removed unnecessary `userpath verify` step that was causing errors. *#4https://github.com/mattsb42/pipx-in-pipx/issues/4*
- Removed hard requirement for Python 3.6+. Leave that for `pipx` to worry about.

0.0.1b0 – 2019-05-11

Initial MVP.