

brewdo: Sandboxing Homebrew on OS X

Matt Behrens · <https://www.zigg.com/> · @zigg
PyOhio · July 26, 2014

Homebrew's `/usr/local`

- By default, Homebrew creates `/usr/local` owned by your user account
- This makes installing software *really easy* in the typical single-user OS X environment
- But it also makes it *really easy* to mess things up—
e.g. `pip install demonically-uninstallable-package-of-doom` outside your virtualenv

I DEMAND PERFECTION
IN ALL THINGS

Sudo: not just for making sandwiches anymore

- Leverages a special account (`_homebrew`) and Sudo to sandbox Homebrew operations
- You can't write to `/usr/local` without it
- Homebrew packages can't write to your home directory... ever (hello, `node!`)

Installation

- Clone from <https://github.com/zigg/brewdo>
- Read the fine `README.md`
- Run `brewdo install`
- `alias brew='brewdo brew'`

What you can do

- Anything Homebrew can do, brewdo can sandbox
 - Prompted for a password only when needed
- Need to do things to your `/usr/local`? Explicit is better than implicit: `brewdo do`
- `brewdo do pip install demonically-uninstallable-package-of-doom`

What's next?

- Migration support—currently no way to move from a default Homebrew installation to brewdo
 - Probably going to do a mass chown or something
- More users! More!
 - I have one dedicated user
 - He's a great guy but he's carrying this all on his own

<https://github.com/zigg/brewdo>

(and try not to notice what it's written in)

