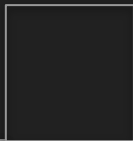
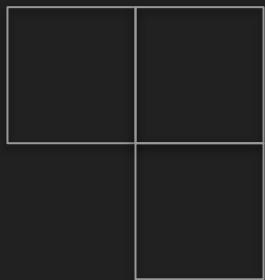


tea - the next-gen package manager



Marc Seitz

Developer Advocate
tea.xyz



what is tea?

tea is the next generation, cross-platform package manager from the creator of `brew`.

Package managers have **not** evolved in decades!

It's time for a new paradigm shift:

→ Don't install packages. Use packages instead.

what is tea?

Tea is a Universal Interpreter

Tea creates developer environments on the fly

Tea gets out of your way!

what is tea?

tea doesn't have a **install step**

```
$ node --eval 'console.log("Hello World!")'  
command not found: node
```

```
$ sh <(curl tea.xyz) --yes  
installing ~/.tea...
```

```
$ node --eval 'console.log("Hello World!")'  
tea: installing nodejs.org^19  
Hello World!
```

what is tea?

tea **stows** packages in `~/tea`; it doesn't actually install them and therefore it does **not** pollutes your `PATH`

```
$ which bun
bun not found

$ tea --dry-run bun --version
imagined: bun.sh^0.4
~/tea/bun.sh/v0.4.0/bin/bun --version

$ bun --version
tea: installing bun.sh^0.4
0.4.0

$ which bun
bun not found
# `bun` is not in your `PATH`
# ∴ tea doesn't install packages
# ∴ using tea doesn't compromise your system's integrity
```

what is tea?

tea **stows** packages in `~/tea`; it doesn't actually install them and therefore it does **not** pollutes your `PATH`

```
mfts@MMBP:~ $ tea https://github.com/teaxyz/demos/blob/main/hello-universe.sh
downloading

*** Interpreters ***

installed: ~/.tea/deno.land/v1.29.2
TypeScript: Hello, World!
installed: ~/.tea/openssl.org/v1.1.1s
installed: ~/.tea/go.dev/v1.19.5
Go: Hello, World!
installed: ~/.tea/unicode.org/v71.1.0
installed: ~/.tea/zlib.net/v1.2.13
installed: ~/.tea/nodejs.org/v19.3.0
JavaScript: Hello, World!
installed: ~/.tea/perl.org/v5.36.0
Perl: Hello, World!
installed: ~/.tea/invisible-island.net/ncurses/v6.4.0
installed: ~/.tea/gnu.org/readline/v8.1.0
installed: ~/.tea/sourceware.org/bzip2/v1.0.8
installed: ~/.tea/sourceware.org/libffi/v3.4.4
installed: ~/.tea/libexpat.github.io/v2.5.0
installed: ~/.tea/bytereef.org/mpdecimal/v2.5.1
installed: ~/.tea/tukaani.org/xz/v5.2.7
installed: ~/.tea/sqlite.org/v3.40.0
installed: ~/.tea/python.org/v3.11.1
Python: Hello, World!
installed: ~/.tea/pyyaml.org/v0.2.5
installed: ~/.tea/ruby-lang.org/v3.2.0
Ruby: Hello World!
```

tea vs homebrew vs nix

The tooling since 2009 has changed! Nix is even older!

vs Homebrew:

highlights are **speed, global installation, rolling versioning.**

tea lets you install the versions you need, and they **don't pollute other processes.**

vs Nix:

"As a Nix person, I'm definitely interested to see what tea can achieve [...] The **less strict approach, day-one emphasis on UX,** and choice of a 'dumb', commonplace configuration DSL (**YAML**) **for package definitions** could certainly incline a lot of people who shy away from Nix to give it a try."

- pxc via HN

installing tea

Installing tea is as easy as curl-ing our homepage.

```
sh <(curl https://tea.xyz)
# • asks you to confirm before it sets up `~/.tea`
# • asks you to confirm before adding one line to your `~/.shellrc`
```

If you prefer to change the location of tea,

```
`TEA_PREFIX=/opt/tea sh <(curl https://tea.xyz)` will do the trick!
```


what's the magic

tea is added to your PATH so that `tea` is available without a symlink.

```
mfts@MMBP:~ $ tea --magic
# this file is generated and should not be edited

function command_not_found_handler {
  if [ "${TEA_MAGIC:-}" != 0 -a -x "/Users/mfts/.tea/tea.xyz/v*/bin"/tea ]; then
    "/Users/mfts/.tea/tea.xyz/v*/bin"/tea -- $*
  fi
}

add-zsh-hook -Uz chpwd() {
  if [ "${TEA_MAGIC:-}" != 0 -a -x "/Users/mfts/.tea/tea.xyz/v*/bin"/tea ]; then
    source <("/Users/mfts/.tea/tea.xyz/v*/bin"/tea --env --keep-going --silent --dry-run=w/trace
  )
  fi
}

# if the user put tea in eg. /usr/local/bin then don't pollute their PATH
# if ! tea true 2>&1 >/dev/null; then
  export PATH="/Users/mfts/.tea/tea.xyz/v*/bin:$PATH"
# fi
```

If you don't want that,
remove the one-liner from your shell config or set `envar TEA_MAGIC=0`

developer environments

During the magic of running

```
$ bun --version
tea: installing bun.sh^0.4
0.4.0
```

tea prepends PATH with the stowed version of `~/tea/bun.sh/`

```
installed: ~/.tea/bun.sh/v0.4.0
{ "symlinking": ~/.tea/bun.sh/v0.4 }
{ "symlinking": ~/.tea/bun.sh/v0 }
{ "symlinking": ~/.tea/bun.sh/v* }
{ readYAML: "/Users/mfts/.tea/tea.xyz/var/pantry/projects/bun.sh/package.yml" }
{
  cwd: undefined,
  cmd: [ "/Users/mfts/.tea/bun.sh/v0.4.0/bin/bun", "--version" ],
  env: {
    PATH: /Users/mfts/.tea/bun.sh/v0.4.0/bin:/Users/mfts/.tea/tea.xyz/v*/bin:/Users/mfts/.npm/versions/node/v1...,
    LD_FLAGS: "-Wl,-rpath,/Users/mfts/.tea",
    TEA_PREFIX: "/Users/mfts/.tea",
    TEA_PKGS: "bun.sh=0.4.0",
    TEA_FORK_BOMB_PROTECTOR: "1"
  }
}
0.4.0
{ exit: { success: true, code: 0 } }
```

developer environments

tea can determine the tools a project directory needs and provide that virtual environment.

With our shell magic just step into the project directory and type commands; tea automatically fetches the specific versions those projects need and runs them.

All based on `# Dependencies` in README.md with a dependencies table

<https://github.com/teaxyz/cli#dependencies>

Dependencies

Project	Version
deno.land	^1.27

package managers

tea aims to work with both OS and source code.

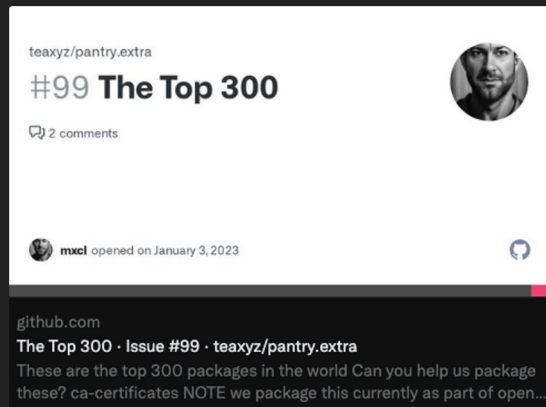
First, focus on on managing source code → see our pantries

Find all ~(160) currently available packages:

```
`ls ~/.tea/tea.xyz/var/pantry/projects`
```

We are packaging the Top 300:

<https://github.com/teaxyz/pantry.extra/issues/99>



packaging

Decentralized pantries

We're launching with two pantries:

- **pantry.core**: essential libraries and utilities
- **pantry.extra**: everything else

New package submissions should go to **pantry.extra**.

```
root/  
  projects/  
    |--- pkg-a.com/  
        |--- package.yml  
    |--- pkg-b.com/  
        |--- pkg-d/  
            |--- package.yml  
    |--- pkg-c.com  
        ...
```

packaging

Packaging a project is as simple as **creating a package.yml** file and opening a **PR on pantry.extra**

Packages will be automatically updated if there's a new release, as determined by **version key**

```
1 distributable:
2   url: https://github.com/casey/just/archive/refs/tags/{{version}}.tar.gz
3   strip-components: 1
4
5 versions:
6   github: casey/just/tags
7   strip: /v/
8
9 build:
10  dependencies:
11    rust-lang.org: ^1.47.0
12    rust-lang.org/cargo: ^0.65
13  script:
14    cargo install --path=. --root={{prefix}}
15  test:
16    cargo test
17
18 test:
19  script: |
20    cp $FIXTURE justfile
21    just
22  fixture: |
23    # run just -l
24    default:
25      @just -l
26
27 provides:
28   - bin/just
```

packaging

I have set up a tea-script for setting up a package environment

```
mfts@MMBP:~ $ tea https://github.com/teaxyz/demos/blob/main/create-a-package.sh
downloading
installed: ~/.tea/gnu.org/bash/v5.1.16

  bootstrap the requirements for building a package

  | what to expect from this script

  • installs gh using tea
  • clones cli, pantry.core repositories
  • forks pantry.extra
  • adds GITHUB_TOKEN to env
  • adds TEA_PANTRY_PATH to env

  shall we continue?

  Yes No
```

Questions?

github.com/teaxyz/cli



Leave a  on GitHub
github.com/teaxyz/cli

