

# 工程建设 - 命令行

 `npm i pake-cli -g`

## 设计点

- 必要配置放出并且满足简单使用
- 国内特殊网络环境镜像替换加速
- 同时兼容 Mac/Windows/Linux 用户使用
- 作为基础能力提供给 Github Actions

## 痛点

- Rust 打包首次编译并没有那么快
- Windows 系统复杂的环境如何解决

## 有趣的功能

~ 🍁 pake

```
$ pake https://weekly.tw93.fun --name Weekly --transparent
```

Arguments:

url

Options:

```
--name <string>
--icon <string>
--height <number>
--width <number>
--no-resizable
--fullscreen
--transparent
--user-agent <string>
--show-menu
--show-system-tray
--system-tray-icon <string>
--iter-copy-file
--multi-arch
--targets <string>
--debug
```

# 工程建设 - Github Actions

有趣的功能

## 利用缓存提速

pake\_build\_with\_cache.yaml

on: workflow\_dispatch

Matrix: build

- ✓ build (linux) 30m 38s
- ✓ build (macos) 23m 43s
- ✓ build (windows) 16m 50s

## 充分利用 workflow

Build App

Build App with Cache

Build App with Pake Cli

Build Contribute List

Check EditorConfig

Check Rust Code

运行测试 cargo-hack / nextest

Mac

Window

Linux

代码质量 cargo-clippy

Mac

Window

Linux

代码格式化 cargo-fmt

Mac

Window

Linux

## 可视化表单

Use workflow from

Branch: master ▾

platform \*

macos-latest ▾

[URL] \*

[Name] \*

[Icon, Optional]

[Height, Optional]

780

[Width, Optional]

1200

☐ [Transparent, Optional]

☒ [Resize, Optional]

☐ [MultiArch, Optional, MacOS only]

[Targets, Optional, Linux only]

deb ▾

Run workflow