Package: cliff (via r-universe)

August 22, 2024

Title Execute Command Line Programs Interactively

Version 0.1.2.9999

Description Execute command line programs and format results for interactive use. It is based on the package 'processx' so it does not use shell to start up the process like system() and system2(). It also provides a simpler and cleaner interface than processx::run().

License MIT + file LICENSE

URL https://github.com/randy3k/cliff

BugReports https://github.com/randy3k/cliff/issues

Suggests withr, crayon, testthat

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Imports ellipsis, processx, rlang

Repository https://randy3k.r-universe.dev

RemoteUrl https://github.com/randy3k/cliff

RemoteRef HEAD

RemoteSha 2d5f23dfcca5d59ba07a12b3cc926f1bf752340b

Contents

run *Run a command line program and wait until it terminates.*

Description

Run a command line program and wait until it terminates.

Usage

```
run(
  command,
  ...,
  input = NULL,
  error_on_status = TRUE,
  stderr_to_stdout = FALSE,
  wd = NULL,
  timeout = Inf,
  env = NULL
)
```

Arguments

command	the command to run
...	the arguments pass to the program, supports the big bang operator !!!
input	text pass to stdin
error_on_status	raise an error if return code is not 0.
stderr_to_stdout	whether stderr should be forwarded to stdout.
wd	working directory
timeout	throw an error after this amount of time in second
env	additional environment variables

Value

The stdout of the program in a scalar character. It may contain a trailing newline. Use `trimws()` to ensure the trailing newline is trimmed.

Examples

```
## Not run:
git <- function(...) cliff::run("git", ...)
git("log", git("rev-parse", "--abbrev-ref", "HEAD"), "-n1")

## End(Not run)
```

Index