Package: tinyeval (via r-universe)

November 11, 2026

Title Evals for the AI Calls in Your R Code

Version 0.11.1

Description Bundles 'tinyeval', one bash script that runs test cases against the AI calls in your app and checks their replies. run() starts it, and worker() answers its test cases from your R code, loaded once.

License MIT + file LICENSE

URL https://tinyeval.com, https://github.com/evalship-hq/tinyeval

BugReports https://github.com/evalship-hq/tinyeval/issues

Depends R (>= 4.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

SystemRequirements bash (>= 3.2)

Repository https://evalship-hq.r-universe.dev

Date/Publication 2026-11-11 11:54:37 UTC

RemoteUrl https://github.com/evalship-hq/tinyeval

RemoteRef r-universe

RemoteSha 76d25060a4eef424c9d620a23e8ffe3bc101e655

RemoteSubdir packages/r

Contents

run 2
worker 2

Index **4**

run *Run tinyeval*

Description

Runs the bundled tinyeval with bash, which runs the test cases in ‘`evals/`’. stdin, stdout and stderr pass through. Ctrl-C reaches bash, which stops the run.

Usage

```
run(...)
```

Arguments

`...`	tinyeval’s arguments, as strings or character vectors, such as `"-n"`, `"3"`.

Details

Needs bash 3.2 or later: macOS, Linux, or Git for Windows, whose bash goes before one on PATH. A repo that commits its own ‘`tinyeval`’ file runs that copy. In CI, exit with its code: `Rscript -e 'quit(status = tinyeval::run())'`.

Value

tinyeval’s exit code, invisibly: 0 when every eval passed, 1 when one failed or couldn’t run, and 2 for a usage error or when bash is missing.

worker *Answer tinyeval’s test cases from R*

Description

`worker()` answers each input tinyeval sends with `answer(input)`, until tinyeval closes stdin. Start it from the `worker` line of an eval file, such as `worker Rscript evals/worker.R`.

`couldnt_run()` signals that the AI call couldn’t run, such as on a missing key. tinyeval counts the test case apart from the failures, and the run still exits 1.

Usage

```
worker(answer)

couldnt_run(reason)
```

Arguments

`answer`	A function that takes the input string and returns the reply.
`reason`	Why the AI call couldn't run, shown in tinyeval's summary.

Details

Call `library(tinyeval)` before your app loads. In a worker tinyeval starts, it points R's output at stderr with `sink`, so `print()` and `cat()` in your app can't end up in a reply. The replies go to the process's stdout. Programs started with `system()` still write to stdout.

httr2's errors, ellmer's included, count as couldn't run when the status is 401, 429 or 500 and up, and when the request couldn't connect or timed out. Any other error becomes the reply, as `class: message`, so the eval fails, and the calls that led to it go to stderr. The worker goes on with the next input.

Value

`worker()` returns when stdin closes. `couldnt_run()` doesn't return.

Index

couldnt_run *(worker)*, 2

run, 2

sink, *3*

worker, 2