Package ‘rmon’

June 19, 2025

Title Monitor Changes in Source Code and Auto-Restart Your Server

Version 1.1.0

Description The 'R' equivalent of 'nodemon'. Watches specified directories for file changes and reruns a designated 'R' script when changes are detected. It's designed to automate the process of reloading your 'R' applications during development, similar to 'nodemon' for 'Node.js'.

License MIT + file LICENSE

URL https://github.com/kennedymwavu/rmon

BugReports https://github.com/kennedymwavu/rmon/issues

Imports processx (>= 3.8.4)

Suggests testthat (>= 3.2.3)

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Kennedy Mwavu [aut, cre] (ORCID: <https://orcid.org/0009-0006-3157-7234>)

Maintainer Kennedy Mwavu <mwavukennedy@gmail.com>

Repository CRAN

Date/Publication 2025-06-19 19:30:02 UTC

Contents

monitor 2

Index **5**

`monitor` *Monitor files for changes and rerun specified script or execute R expression*

Description

Monitors specified directories for file changes and either reruns a designated R script or executes an arbitrary R expression when changes are detected. It's designed to automate the process of reloading your R applications during development, similar to nodemon for Node.js.

Usage

```
monitor(
  dir,
  file = NULL,
  expr = NULL,
  ext = "*",
  monitor_hidden = FALSE,
  exclude_files = NULL,
  exclude_patterns = NULL,
  exclude_dirs = NULL,
  delay = 1,
  capture_output = TRUE,
  on_error = c("continue", "stop")
)
```

Arguments

`dir`	Character vector. Directory or directories to monitor for changes.
`file`	String, file path. Path to the R script to rerun when changes are detected. Mutually exclusive with `expr`.
`expr`	String or expression. R expression to execute when changes are detected. Can be a string containing R code or an R expression object. Mutually exclusive with `file`.
`ext`	Character vector. File extensions to watch. "*" (the default) watches all files in `dir`.
`monitor_hidden`	Logical. Should hidden files be monitored for changes? Default is `FALSE`. Hidden files are those whose names start with a dot eg. `.Renviron`, `.env`, etc. This option is especially helpful when `ext = "*"`.
`exclude_files`	Character vector. Specific files to ignore. Changes to these files will not trigger a script rerun. Default is `NULL`.
`exclude_patterns`	Character vector. File name patterns to ignore. Any files in `dir` with names matching these patterns will be ignored. Default is `NULL`.
`exclude_dirs`	Character vector. Directories to exclude from monitoring. Default is `NULL`.

delay	Numeric. Number of seconds to wait before checking for file changes. Defaults to 1.
capture_output	Logical. When using expr, should the output be captured and displayed? Default is TRUE.
on_error	Character. What to do when expression execution fails. Options are "continue" (default) to keep monitoring, or "stop" to halt monitoring.

Details

The monitoring process can be customized by excluding specific files, file patterns, or entire directories. This allows you to ignore changes to files that shouldn't trigger a reload (eg. temporary files, log files, etc.).

If multiple directories are supplied, file is assumed to be in the first directory.

When using expr, the expression is evaluated in the current R session's global environment. This allows access to all loaded packages and variables.

The function runs indefinitely until interrupted.

Value

NULL

Examples

```
if (interactive()) {
# monitor current directory, rerun 'app.R' on changes, ignore 'dev.R' and
# any files in 'test/' directory:
rmon::monitor(
  dir = ".",
  file = "app.R",
  exclude_files = "dev.R",
  exclude_dirs = "test"
)

# monitor multiple directories, watch only `.R` & `.Rmd` files:
rmon::monitor(
  dir = c("src", "scripts"),
  file = "main.R",
  ext = c(".R", ".Rmd")
)

# execute expression with natural R syntax:
rmon::monitor(dir = ".", expr = {
  data <- read.csv("data.csv")
  summary(data)
})

# execute an R expression when files change:
rmon::monitor(
  dir = ".",
  expr = "print('Woohoo!'); data <- read.csv('data.csv')"
```

```
)

# execute expression without capturing output:
rmon::monitor(
  dir = ".",
  expr = "source('reload_functions.R')",
  capture_output = FALSE
)
}
```

Index

monitor, 2