Package 'loggit2'

May 3, 2024

Title Easy-to-Use, Dependencyless Logger

Description An easy-to-use 'ndjson' (newline-delimited 'JSON') logger.
It provides a set of wrappings for base R's message(), warning(), and stop() functions that maintain identical functionality, but also log the handler message to an 'ndjson' log file.
No change in existing code is necessary to use this package, and should only require additions to fully leverage the power of the logging system.

Version 2.2.2

License MIT + file LICENSE

Depends R (>= 3.4.0)

Suggests knitr (>= 1.19), rmarkdown (>= 1.8), testthat (>= 3.0.0)

URL https://github.com/ME0265/loggit2

BugReports https://github.com/ME0265/loggit2/issues

RoxygenNote 7.3.0

Encoding UTF-8

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Matthias Ollech [cre, aut],
Ryan Price [fnd, aut]

Maintainer Matthias Ollech <ollech@gmx.com>

Repository CRAN

Date/Publication 2024-05-03 12:30:06 UTC

R topics documented:

`get_logfile` *Get Log File*

Description

Return the log file that `loggit()` will write to.

Usage

```
get_logfile()
```

Value

The log file path.

Examples

```
get_logfile()
```

`get_timestamp_format` *Get Timestamp Format*

Description

Get timestamp format for use in output logs.

Usage

```
get_timestamp_format()
```

Value

The timestamp format.

Examples

```
get_timestamp_format()
```

loggit *Log entries to file*

Description

Log entries to a ndjson log file, defined by `set_logfile()`.

Usage

```
loggit(
  log_lvl,
  log_msg,
  ...,
  echo = TRUE,
  custom_log_lvl = FALSE,
  sanitizer = default_ndjson_sanitizer
)
```

Arguments

`log_lvl`	Log level coerceable to `character`. For details see parameter `custom_log_lvl`.
`log_msg`	Main log message. Will be coerced to class `character`.
`...`	Named arguments, each a atomic vector of length one, you wish to log. The names of the arguments are treated as column names in the log.
`echo`	Should the log file entry be printed to the console as well? Defaults to `TRUE`.
`custom_log_lvl`	Allow log levels other than "DEBUG", "INFO", "WARN", and "ERROR"? Defaults to `FALSE`.
`sanitizer`	Sanitizer function to run over elements in log data. Defaults to `default_ndjson_sanitizer()`.

Value

Invisible `NULL`.

Examples

```
loggit("INFO", "This is a message", but_maybe = "you want more fields?",
sure = "why not?", like = 2, or = 10, what = "ever")
```

message *Diagnostic Messages Log Handler*

Description

This function is identical to base R's message, but it includes logging of the exception message via loggit().

Usage

```
message(..., domain = NULL, appendLF = TRUE, .loggit = TRUE, echo = TRUE)
```

Arguments

...	zero or more objects which can be coerced to character (and which are pasted together with no separator) or (for message only) a single condition object.
domain	see gettext. If NA, messages will not be translated, see also the note in stop.
appendLF	logical: should messages given as a character string have a newline appended?
.loggit	Should loggit() execute? Defaults to TRUE.
echo	Should loggit()'s log entry be echoed to the console, as well? Defaults to TRUE.

Value

Invisible NULL.

See Also

Other handlers: stopifnot(), stop(), warning()

Examples

```
if (2 < 1) message("Don't say such silly things!")
```

read_logs *Return log file as an R data frame*

Description

This function returns a data.frame containing all the logs in the provided ndjson log file.

Usage

```
read_logs(logfile = get_logfile(), unsanitizer = default_ndjson_unsanitizer)
```

Arguments

logfile	Path to log file.
unsanitizer	Unsanitizer function to run over elements in log. Defaults to default_ndjson_unsanitizer()

Value

A data.frame.

Examples

```
set_logfile(file.path(tempdir(), "loggit.log"), confirm = FALSE)
message("Test log message")
read_logs()
```

---

rotate_logs *Rotate log file*

Description

Truncates the log file to the line count provided as rotate_lines.

Usage

```
rotate_logs(rotate_lines = 100000L, logfile = get_logfile())
```

Arguments

rotate_lines	The number of log entries to keep in the logfile.
logfile	Log file to truncate.

Value

Invisible NULL.

Examples

```
# Truncate "default" log file to 100 lines
set_logfile()
for (i in 1:150) {loggit("INFO", i, echo = FALSE)}
rotate_logs(100)

# Truncate a different log file to 250 lines
another_log <- file.path(tempdir(), "another.log")
set_logfile(another_log)
for (i in 1:300) {loggit("INFO", i, echo = FALSE)}
set_logfile() # clears pointer to other log file
rotate_logs(250, another_log)
```

sanitizers *Sanitization for ndJSON.*

Description

Sanitizers and *unsanitizers* are functions, with one parameter, that convert a character vector (of any length) into another (of the same length). Associated *sanitizer* and *unsanitizer* should be constructed in such a way that the concatenation `unsanitizer(sanitizer())` corresponds to the identity function.

Usage

```
default_ndjson_sanitizer(string)

default_ndjson_unsanitizer(string)
```

Arguments

`string`	A character vector

Details

The default sanatizer and unsanatizer are based on the following mapping:

Character	Replacement
{	__LEFTBRACE__
}	__RIGHTBRACE__
"	__DBLQUOTE__
,	__COMMA__
\r	__CR__
\n	__LF__

This type of function is needed because because some characters in a JSON cannot appear unescaped and since `loggit2` reimplements its own very simple string-based JSON parser.

Value

A character vector

set_logfile *Set Log File*

Description

Set the log file that loggit will write to.

Usage

```
set_logfile(logfile = NULL, confirm = TRUE, create = TRUE)
```

Arguments

logfile	Absolut or relative path to log file. An attempt is made to convert the path into a canonical absolute form using normalizePath(). If NULL will set to <tmpdir>/loggit.log.
confirm	Print confirmation of log file setting? Defaults to TRUE.
create	Create the log file if it does not exist? Defaults to TRUE.

Details

No logs outside of a temporary directory will be written until this is set explicitly, as per CRAN policy. Therefore, the default behavior is to create a file named loggit.log in your system's temporary directory.

Value

Invisible NULL.

Examples

```
set_logfile(file.path(tempdir(), "loggit.log"))
```

---

set_timestamp_format *Set Timestamp Format*

Description

Set timestamp format for use in output logs.

Usage

```
set_timestamp_format(ts_format = "%Y-%m-%dT%H:%M:%S%z", confirm = TRUE)
```

Arguments

ts_format	ISO date format. Defaults to ISO-8601 (e.g. "2020-01-01T00:00:00+0000").
confirm	Print confirmation message of timestamp format? Defaults to TRUE.

Details

This function performs no time format validations, but will echo out the current time in the provided format for manual validation.

This function provides no means of setting a timezone, and instead relies on the host system's time configuration to provide this. This is to enforce consistency across software running on the host.

Value

Invisible NULL.

Examples

```
set_timestamp_format("%Y-%m-%d %H:%M:%S")
```

---

stop *Stop Function Log Handler*

Description

This function is identical to base R's stop, but it includes logging of the exception message via loggit().

Usage

```
stop(..., call. = TRUE, domain = NULL, .loggit = TRUE, echo = TRUE)
```

Arguments

...	zero or more objects which can be coerced to character (and which are pasted together with no separator) or a single condition object.
call.	logical, indicating if the call should become part of the error message.
domain	see gettext. If NA, messages will not be translated.
.loggit	Should loggit() execute? Defaults to TRUE.
echo	Should loggit()'s log entry be echoed to the console, as well? Defaults to TRUE.

Value

No return value.

See Also

Other handlers: message(), stopifnot(), warning()

Examples

```
if (2 < 1) stop("This is a completely false condition, which throws an error")
```

`stopifnot` *Conditional Stop Function Log Handler*

Description

This function is identical to base R's `stopifnot`, but it includes logging of the exception message via `loggit()`.

Usage

```
stopifnot(..., exprObject, local, .loggit = TRUE, echo = TRUE)
```

Arguments

`...`	zero or more objects which can be coerced to character (and which are pasted together with no separator) or (for `message` only) a single condition object.
`exprObject`	alternative to `exprs` or `...`: an 'expression-like' object, typically an `expression`, but also a `call`, a `name`, or atomic constant such as `TRUE`.
`local`	(only when `exprs` is used:) indicates the `environment` in which the expressions should be evaluated; by default the one from where `stopifnot()` has been called.
`.loggit`	Should `loggit()` execute? Defaults to `TRUE`.
`echo`	Should `loggit()`'s log entry be echoed to the console, as well? Defaults to `TRUE`.

Value

(`NULL` if all statements in `...` are `TRUE`.)

See Also

Other handlers: `message()`, `stop()`, `warning()`

Examples

```
stopifnot("This is a completely false condition, which throws an error" = TRUE)
```

warning *Warning Messages Log Handler*

Description

This function is identical to base R's warning, but it includes logging of the exception message via loggit().

Usage

```
warning(
  ...,
  call. = TRUE,
  immediate. = FALSE,
  noBreaks. = FALSE,
  domain = NULL,
  .loggit = TRUE,
  echo = TRUE
)
```

Arguments

...	zero or more objects which can be coerced to character (and which are pasted together with no separator) or a single condition object.
call.	logical, indicating if the call should become part of the warning message.
immediate.	logical, indicating if the call should be output immediately, even if getOption("warn") <= 0.
noBreaks.	logical, indicating as far as possible the message should be output as a single line when options(warn = 1).
domain	see gettext. If NA, messages will not be translated, see also the note in stop.
.loggit	Should loggit() execute? Defaults to TRUE.
echo	Should loggit()'s log entry be echoed to the console, as well? Defaults to TRUE.

Value

The warning message as character string, invisibly.

See Also

Other handlers: message(), stopifnot(), stop()

Examples

```
if (2 < 1) warning("You may want to review that math, and so this is your warning")
```

Index