

Package ‘loggit’

April 9, 2018

Title Effortless Exception Logging

Description A very simple and easy-to-use set of suspiciously-familiar functions. 'loggit' 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 a 'JSON' log file. While mostly automatic, powerful custom logging is available via these handlers' logging function, loggit(), which is also exported for use. No change in existing code is necessary to use this package.

Version 1.1.1

Date 2018-04-08

License MIT + file LICENSE

Depends R (>= 3.4.0)

Imports dplyr (>= 0.7.4), jsonlite (>= 1.5)

Suggests knitr (>= 1.19), rmarkdown (>= 1.8), testthat (>= 2.0.0)

URL <https://github.com/ryapric/loggit>

BugReports <https://github.com/ryapric/loggit/issues>

RoxygenNote 6.0.1

NeedsCompilation no

Author Ryan Price [cre, aut]

Maintainer Ryan Price <ryapric@gmail.com>

Repository CRAN

Date/Publication 2018-04-09 07:36:59 UTC

R topics documented:

getLogFile	2
getTimestampFormat	2
get_logs	3
handlers	3
loggit	4
setLogFile	5
setTimestampFormat	6

`getLogFile`*Get Log File*

Description

Get the log file that loggit will write to.

Usage

```
getLogFile()
```

Examples

```
getLogFile()
```

`getTimestampFormat`*Get Timestamp Format*

Description

Get timestamp format for use in output logs.

Usage

```
getTimestampFormat()
```

Examples

```
getTimestampFormat()
```

get_logs	<i>Return Log File as an R Object</i>
----------	---------------------------------------

Description

This function returns a `data.frame` (by default) containing all the logs in the provided JSON log file. If no explicit log file is provided, calling this function will return a data frame of the log file currently pointed to by the `loggit` functions. Users can request that the logs be returned as a `list`, though this is not recommended.

Usage

```
get_logs(logfile, as_df = TRUE)
```

Arguments

<code>logfile</code>	JSON-format log file to return.
<code>as_df</code>	Should logfile be returned in a <code>data.frame</code> vs. a <code>list</code> ? Defaults to <code>TRUE</code> , and will return a <code>data.frame</code> .

Value

A `data.frame`.

Examples

```
setLogFile(file.path(tempdir(), "loggit.json"), confirm = FALSE)
message("Test log message")
get_logs(getLogFile())
```

handlers	<i>Loggit's Exception Handlers</i>
----------	------------------------------------

Description

These exception handlers are identical to base R's [message](#), [warning](#), and [stop](#), but with included logging of the exception messages.

Usage

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

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

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

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 function execute? Defaults to TRUE.
log_detail	Additional information to accompany log message. Must be a string.
echo	Should loggit's log be echoed to the console, as well? Defaults to FALSE.
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 <code>getOption("warn") <= 0</code> .
noBreaks.	logical, indicating as far as possible the message should be output as a single line when <code>options(warn = 1)</code> .

Examples

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

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

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

loggit	<i>Log Message to Output File</i>
--------	-----------------------------------

Description

This function executes immediately before the function definitions for the base handler functions ([message](#), [warning](#), and [stop](#), and logs their timestamped output (a bit more verbosely) to a log file. The log file defaults to a **JSON** file, which is a portable file format that is easily parsed by many systems, but will eventually have a `.txt` option as well.

Usage

```
loggit(log_lvl, log_msg, log_detail = "", ..., echo = TRUE)
```

Arguments

log_lvl	Level of log output. In actual practice, one of "INFO", "WARN", and "ERROR" are common, but any string may be supplied. Will be coerced to class character.
log_msg	Main log message. Will be coerced to class character.
log_detail	Additional detail recorded along with a log message.
...	A named list or named vector (each element of length one) of other custom fields you wish to log. You do not need to explicitly provide these fields as a formal list or vector, as shown in the example; R handles the coercion.
echo	Should a message be printed to the console as well? Defaults to TRUE, and is truncated to just the level & message of the log. This argument is passed as FALSE when called from loggit's handlers, since they still call base R's handlers at the end of execution, all of which print to the console as well.

Details

While this function has an intended use of logging handler messages without any direct user interaction, it is flexible enough to be used as you see fit.

Examples

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

setLogFile

Set Log File

Description

Set the log file that loggit will write to. No logs will be written until this is set, as per CRAN policy. The suggested use of this function would be to call it early, to log to the current working directory, as follows: `setLogFile(paste0(getwd(), "/loggit.json"))`. If you are using loggit in your package, you can wrap this function in `.onLoad()` so that the logfile is set when your package loads.

Usage

```
setLogFile(logfile = NULL, confirm = TRUE)
```

Arguments

logfile	Full path to log file. Until other output formats are introduced, the file name must end in ".json".
confirm	Print confirmation of log file setting? Defaults to TRUE.

Examples

```
setLogFile(file.path(tempdir(), "loggit.json"))
```

setTimestampFormat	<i>Set Timestamp Format</i>
--------------------	-----------------------------

Description

Set timestamp format for use in output logs.

Usage

```
setTimestampFormat(ts_format = "%Y-%m-%d %H:%M:%S", confirm = TRUE)
```

Arguments

ts_format	ISO date format.
confirm	Print confirmation of timestamp format setting? Defaults to TRUE.

Examples

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

Index

get_logs, 3
getLogFile, 2
getOption, 4
gettext, 4
getTimestampFormat, 2

handlers, 3

loggit, 4

message, 3, 4
message (handlers), 3

setLogFile, 5
setTimestampFormat, 6
stop, 3, 4
stop (handlers), 3

warning, 3, 4
warning (handlers), 3