Package: teal.logger (via r-universe)

August 29, 2024

Title Logging Setup for the 'teal' Family of Packages

Version 0.2.0.9008

Date 2024-08-29

Description Utilizing the 'logger' framework to record events within a package, specific to 'teal' family of packages. Supports logging namespaces, hierarchical logging, various log destinations, vectorization, and more.

License Apache License 2.0

URL https://insightsengineering.github.io/teal.logger/, https://github.com/insightsengineering/teal.logger/

BugReports https://github.com/insightsengineering/teal.logger/issues

Depends R (>= 3.6)

Imports glue (>= 1.0.0), lifecycle (>= 0.2.0), logger (>= 0.2.0), methods, shiny (>= 1.6.0), utils, withr (>= 2.1.0)

Suggests knitr (>= 1.42), rmarkdown (>= 2.23), testthat (>= 3.1.7)

VignetteBuilder knitr, rmarkdown

RdMacros lifecycle

Config/Needs/verdepcheck tidyverse/glue, r-lib/lifecycle, daroczig/logger, rstudio/shiny, r-lib/withr, yihui/knitr, rstudio/rmarkdown, r-lib/testthat

Config/Needs/website insightsengineering/nesttemplate

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

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

RemoteUrl https://github.com/insightsengineering/teal.logger

RemoteRef HEAD

RemoteSha cd0e0df17498ecc05b3edfd214766c03c923a020

Contents

`log_shiny_input_changes` *Auto logging input changes in Shiny app*

Description

This is to be called in the `server` section of the Shiny app.

Usage

```
log_shiny_input_changes(
  input,
  namespace = NA_character_,
  excluded_inputs = character(),
  excluded_pattern = "_width$"
)
```

Arguments

`input`	passed from Shiny `server`
`namespace`	the name of the namespace
`excluded_inputs`	character vector of input names to exclude from logging
`excluded_pattern`	character of length one including a grep pattern of names to be excluded from logging

Details

Function having very similar behavior as `logger::log_shiny_input_changes()` but adjusted for `teal` needs.

Examples

```
## Not run:
library(shiny)

ui <- bootstrapPage(
  numericInput("mean1", "mean1", 0),
  numericInput("mean2", "mean2", 0),
  numericInput("sd", "sd", 1),
  textInput("title", "title", "title"),
  textInput("foo", "This is not used at all, still gets logged", "foo"),
  passwordInput("password", "Password not to be logged", "secret"),
  plotOutput("plot")
)

server <- function(input, output) {
  log_shiny_input_changes(input, excluded_inputs = "password", excluded_pattern = "mean")

  output$plot <- renderPlot({
    hist(rnorm(1e3, input$mean, input$sd), main = input$title)
  })
}

shinyApp(ui = ui, server = server)

## End(Not run)
```

---

log_system_info *Logs the basic information about the session.*

Description

Logs the basic information about the session.

Usage

```
log_system_info()
```

Value

```
invisible(NULL)
```

`register_handlers` *Register handlers for logging messages, warnings and errors*

Description

Register handlers for logging messages, warnings and errors

Usage

```
register_handlers(namespace, package = namespace)
```

Arguments

`namespace`	(`character(1)`) the logger namespace
`package`	(`character(1)`) the package name

Details

This function registers global handlers for messages, warnings and errors. The handlers will investigate the call stack and if it contains a function from the package, the message, warning or error will be logged into the respective namespace.

The handlers are registered only once per package and type. Consecutive calls will no effect. Registering handlers for package `base` is not supported.

Use `TEAL.LOG_MUFFLE` environmental variable or `teal.log_muffle` R option to optionally control recover strategies. If `TRUE` (a default value) then the handler will jump to muffle restart for a given type of condition and doesn't continue (with output to the console). Applicable for message and warning types only. The errors won't be suppressed.

Value

`NULL` invisibly. Called for its side effects.

Note

Registering handlers is forbidden within `tryCatch()` or `withCallingHandlers()`. Because of this, handlers are registered only if it is possible.

See Also

`globalCallingHandlers()`

Examples

```
## Not run:
register_handlers("teal.logger")
# see the outcome
globalCallingHandlers()

## End(Not run)
```

`register_logger` *Registers a logger instance in a given logging namespace.*

Description

[Experimental]

Usage

```
register_logger(namespace = NA_character_, layout = NULL, level = NULL)
```

Arguments

`namespace`	(`character(1)` or `NA_character_`) the name of the logging namespace
`layout`	(`character(1)`) the log layout. Alongside the standard logging variables provided by the `logging` package (e.g. `pid`) the `token` variable can be used which will write the last 8 characters of the shiny session token to the log.
`level`	(`character(1)` or `call`) the log level. Can be passed as character or one of the `logger`'s objects. See `logger::log_threshold()` for more information.

Details

Creates a new logging namespace specified by the `namespace` argument. When the `layout` and `level` arguments are set to `NULL` (default), the function gets the values for them from system variables or R options. When deciding what to use (either argument, an R option or system variable), the function picks the first non `NULL` value, checking in order:

1. Function argument.
2. System variable.
3. R option.

`layout` and `level` can be set as system environment variables, respectively:

- `teal.log_layout` as `TEAL.LOG_LAYOUT`,
- `teal.log_level` as `TEAL.LOG_LEVEL`.

If neither the argument nor the environment variable is set the function uses the following R options:

- options(teal.log_layout), which is passed to logger::layout_glue_generator(),
- options(teal.log_level), which is passed to logger::log_threshold()

The logs are output to stdout by default. Check logger for more information about layouts and how to use logger.

Value

invisible(NULL)

Note

It's a thin wrapper around the logger package.

See Also

The package vignettes for more help: browseVignettes("teal.logger").

Examples

```
options(teal.log_layout = "{msg}")
options(teal.log_level = "ERROR")
register_logger(namespace = "new_namespace")

logger::log_info("Hello from new_namespace", namespace = "new_namespace")
```

---

suppress_logs *Suppress logger logs*

Description

This function suppresses logger when running tests via testthat. To suppress logs for a single test, add this function call within the testthat::test_that expression. To suppress logs for an entire test file, call this function at the start of the file.

Usage

```
suppress_logs()
```

Value

NULL invisible

Examples

```
testthat::test_that("An example test", {
  suppress_logs()
  testthat::expect_true(TRUE)
})
```

Index