

# Package ‘wrappr’

May 23, 2023

**Title** A Collection of Helper and Wrapper Functions

**Version** 0.1.0

**Description** Helper functions to easily add functionality to functions. The package can assign functions to have an lazy evaluation allowing you to save and update the arguments before and after each function call. You can set a temporary working directory within functions and wrap console messages around other functions.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Imports** methods

**NeedsCompilation** no

**Author** John Piper [aut, cre] (<<https://orcid.org/0009-0000-6003-7671>>)

**Maintainer** John Piper <[john.piper.using.r@gmail.com](mailto:john.piper.using.r@gmail.com)>

**Repository** CRAN

**Date/Publication** 2023-05-23 08:50:02 UTC

## R topics documented:

get_cache_or_create . . . . .	2
lazy_eval . . . . .	3
msg_wrap . . . . .	4
set_temp_wd . . . . .	5
<b>Index</b>	<b>7</b>

---

get\_cache\_or\_create    *Checks if variable exists in environment and returns back or creates a new variable*

---

### Description

Checks if variable exists in environment and returns back or creates a new variable

### Usage

```
get_cache_or_create(  
  var,  
  func,  
  ...,  
  exists_func_args = NA,  
  get_func_args = NA,  
  warning_msg = NA_character_  
)
```

### Arguments

var                    character. The name of the variable to check in the global environment.

func                   function. A function that returns a value.

...                    Additional arguments to be passed to the param func.

exists\_func\_args      list. A list of arguments to use in base::exists.

get\_func\_args         list. A list of arguments to use in base::get.

warning\_msg           character. Message sent to stop function if an error occurs.

### Value

Unknown. The return type from the param func or the existing variable in global environment.

### Examples

```
## Not run:  
df <- data.frame(col_1 = c("a","b","c"), col_2 = c(1,2,3))  
  
create_blank_df <- function() {  
  data.frame(col_1 = NA_character_, col_2 = NA_integer_)  
}  
  
df_1 <- get_cache_or_create(  
  "df",  
  create_blank_df
```

```

    )

df_2 <- get_cache_or_create(
  "df_2",
  create_blank_df
)

## End(Not run)

```

---

lazy_eval	<i>save and Delay a function call with the option to change the function and arguments when called</i>
-----------	--

---

### Description

save and Delay a function call with the option to change the function and arguments when called

### Usage

```
lazy_eval(..., .f)
```

### Arguments

...	Additional arguments to be passed to the param .f. Also in closure function returned.
.f	function. A function that will be called when needed. Also in closure function returned.

### Value

closure function with same param names plus the param names `overwrite_args` Boolean and `return_new_closure` Boolean.

### Examples

```

numbers <- c(1,2,3,4,5)

func <- lazy_eval(numbers, .f = sum)

sum_result <- func()

max_result <- func(.f = max)

mean_result <- func(.f = mean)

range_result <- func(.f = function(...) { max(...) - min(...)})

add_more_num_result <- func(4,5,6, NA, na.rm = TRUE)

```

```
updated_func <- func(na.rm = TRUE, return_new_closure = TRUE)
updated_func_result <- updated_func()
```

---

msg\_wrap

*Wraps a message before and/or after a function*


---

## Description

Wraps a message before and/or after a function

## Usage

```
msg_wrap(
  func,
  ...,
  before_func_msg = "",
  after_func_msg = "",
  print_func = print,
  use_msg = "both",
  print_return_var = FALSE
)
```

## Arguments

func	function.
...	Additional arguments to be passed into the param func.
before_func_msg	character.
after_func_msg	character.
print_func	function. The default is print. Can use related function like message.
use_msg	character. The default is "both". Selects which messages to print in the function. Use before, after, both or none.
print_return_var	Boolean. The default is FALSE. Prints the output from the called func using the print argument from param print_func.

## Value

Unknown. The return type from the param func.

## Examples

```
numbers <- c(1,2,3,4,5)

answer <- msg_wrap(
  sum,
  numbers,
  before_func_msg = "Currently summing the numbers",
  after_func_msg = "Summing the numbers complete"
)

numbers_with_na <- c(1,2,3,NA,5)

answer_na_removed <- msg_wrap(
  sum,
  numbers,
  na.rm = TRUE,
  before_func_msg = "Sum with na.rm set to TRUE",
  use_msg = "before"
)

numbers_to_sum <- c(10,20,30)

msg_wrap((function(x) sum(x[x%%2 == 1])),
  x = numbers_to_sum,
  before_func_msg = "Result from sum of odd numbers",
  use_msg = "before",
  print_return_var = TRUE
)
```

---

set\_temp\_wd

*Sets a temporary working directory within the function scope*

---

## Description

Sets a temporary working directory within the function scope

## Usage

```
set_temp_wd(
  temp_cwd,
  func,
  ...,
  err_msg = "An error has occurred in the function set_temp_wd"
)
```

**Arguments**

temp_cwd	character. Folder path to temporarily set the working directory
func	function. A function that used a directory path
...	Additional arguments to be passed to the param func.
err_msg	character. Message sent to stop function if an error occurs.

**Value**

Unknown. The return type from the param func.

**Examples**

```
## Not run:  
  
temp_wd <- "example/folder/address/to/change"  
  
get_data <- set_temp_wd(temp_wd, read.csv, file = "file.csv")  
  
## End(Not run)
```

# Index

`get_cache_or_create`, [2](#)

`lazy_eval`, [3](#)

`msg_wrap`, [4](#)

`set_temp_wd`, [5](#)