Package: eponge (via r-universe)

September 4, 2024

Type Package

Title Keep Your Environment Clean

Version 0.1.0

Description Provides a set of functions, which facilitates removing
objects from an environment. It allows to delete objects
specified with regular expression or with other conditions
(e.g. if object is numeric), using one function call.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

BugReports https://github.com/krzjoa/eponge/issues

URL https://github.com/krzjoa/eponge, https://krzjoa.github.io/eponge/

RoxygenNote 6.1.1

Suggests testthat

Imports rlang

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

RemoteUrl https://github.com/krzjoa/eponge

RemoteRef HEAD

RemoteSha 3c34a22b1624efc5fede773d54459987d85d3ee0

Contents

eponge-package *Package eponge*

Description

Provides a set of functions, which facilitates removing objects from an environment. It allows to delete objects specified with regular expression or with other conditions (e.g. if object is numeric), using one function call.

Author(s)

Krzysztof Joachimiak

erase *Remove (all) objects from environment*

Description

Remove (all) objects from environment

Usage

```
erase(pattern = NULL, envir = parent.frame(), verbose = FALSE)
```

Arguments

`pattern`	regex pattern to select a set of objects; default: NULL
`envir`	environment; default: caller environment
`verbose`	print removed objects’ names

Details

Function can be used with envir = globalenv() argument.

Value

NULL (function returns nothing)

Examples

```
create_data <- function() data.frame(a = 1:10, b = 11:20)
x <- cars
y <- 1:20
z <- function(x) x +2
# Typically, we don't have to specify eironment
erase()
ls()
```

erase_data *Remove all objects, which are listed in 'Data' section in RStudio*

Description

Remove all objects, which are listed in 'Data' section in RStudio

Usage

```
erase_data(pattern = NULL, envir = parent.frame(), verbose = FALSE)
```

Arguments

pattern	regex pattern to select a set of objects; default: NULL
envir	environment; default: caller environment
verbose	print removed objects' names

Details

Function can be used with envir = globalenv() argument.

Value

NULL (function returns nothing)

Examples

```
cars.2 <- cars
test_fun <- function(x) x + 2
value <- 7
erase_data(verbose = TRUE)
```

erase_df *Remove all the 'data.frame' objects*

Description

Remove all the 'data.frame' objects

Usage

```
erase_df(pattern = NULL, envir = parent.frame(), verbose = FALSE)
```

Arguments

`pattern`	regex pattern to select a set of objects; default: NULL
`envir`	environment; default: caller environment
`verbose`	print removed objects' names

Details

Function can be used with envir = globalenv() argument.

Value

NULL (function returns nothing)

Examples

```
cars.2 <- cars
test_fun <- function(x) x + 2
value <- 7
erase_df(verbose = TRUE)
```

`erase_functions` *Remove (all) functions from environment*

Description

Remove (all) functions from environment

Usage

```
erase_functions(pattern = NULL, envir = parent.frame(),
  verbose = FALSE)
```

Arguments

`pattern`	regex pattern to select a set of objects; default: NULL
`envir`	environment; default: caller environment
`verbose`	print removed objects' names

Details

Function can be used with envir = globalenv() argument. Be careful: uncontrolled use may cause undesired side effects.

Value

NULL (function returns nothing)

Examples

```
create_data <- function() data.frame(a = 1:10, b = 11:20)
x <- cars
y <- 1:20
z <- function(x) x +2
erase_functions()
ls()
```

`erase_if` *Remove objects, which fulfill determined conditions*

Description

Remove objects, which fulfill determined conditions

Usage

```
erase_if(condition, pattern = NULL, envir = parent.frame(),
  verbose = FALSE)
```

Arguments

`condition`	function or lambda expression (one side formula)
`pattern`	regex pattern to select a set of objects; default: NULL
`envir`	environment; default: caller environment
`verbose`	print removed objects' names

Details

Function can be used with envir = globalenv() argument.

Value

NULL (function returns nothing)

Examples

```
create_data <- function() data.frame(a = 1:10, b = 11:20)
x <- cars
y <- 1:20
z <- function(x) x +2
l <- list(1,2,3,4)
erase_if(is.list)
ls()
# You may use lambda expression
create_data <- function() data.frame(a = 1:10, b = 11:20)
x <- cars
y <- 1:20
```

```
z <- function(x) x +2
l <- list(1,2,3,4)
erase_if(~ is.function(.x) | is.data.frame(.x))
ls()
```

erase_masking *Erase objects from GlobalEnv, which are masking objects from attached packages*

Description

Erase objects from GlobalEnv, which are masking objects from attached packages

Usage

```
erase_masking(pattern = NULL, verbose = FALSE)

erase_masking_functions(pattern = NULL, verbose = FALSE)
```

Arguments

`pattern`	a regex pattern
`verbose`	print removed objects' names

Details

We have to highglight, that for now it only allows us to remove objects from the Global Environment. Be careful: uncontrolled use may cause undesired side effects.

Value

NULL (function returns nothing)

Examples

```
# It works only if objects are assigned in the global environment
matrix <- matrix(0, 3, 3)
gamma <- 0.9
erase_masking()
```

`erase_non_functions` *Remove all the objects, that are not functions*

Description

Remove all the objects, that are not functions

Usage

```
erase_non_functions(pattern = NULL, envir = parent.frame(),
  verbose = FALSE)
```

Arguments

`pattern`	regex pattern to select a set of objects; default: NULL
`envir`	environment; default: caller environment
`verbose`	print removed objects’ names

Details

Function can be used with envir = globalenv() argument.

Value

NULL (function returns nothing)

Examples

```
cars.2 <- cars
test_fun <- function(x) x + 2
value <- 7
erase_non_functions(verbose = TRUE)
```

`erase_values` *Remove all objects, which are listed in ‘Values‘ section in RStudio*

Description

Remove all objects, which are listed in ‘Values‘ section in RStudio

Usage

```
erase_values(pattern = NULL, envir = parent.frame(), verbose = FALSE)
```

Arguments

pattern	regex pattern to select a set of objects; default: NULL
envir	environment; default: caller environment
verbose	print removed objects' names

Details

Function can be used with envir = globalenv() argument.

Value

NULL (function returns nothing)

Examples

```
cars.2 <- cars
test_fun <- function(x) x + 2
value <- 7
erase_values(verbose = TRUE)
```

Index