Package: kwb.context (via r-universe)

August 26, 2024

Title Get the Function Call Context and Work with it

Version 0.1.0

Description This package contains functions to get the full tree of function calls that is evaluated when calling a function. The idea is to reuse some of these calls with modified arguments, e.g. to replot a specific plot that was created by an inner plot function that was called from an outer function.

License MIT + file LICENSE

URL https://github.com/KWB-R/kwb.context

BugReports https://github.com/KWB-R/kwb.context/issues

Imports kwb.default, kwb.utils

Suggests knitr, covr, rmarkdown

Remotes github::kwb-r/kwb.default, github::kwb-r/kwb.utils

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

VignetteBuilder knitr

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

RemoteUrl https://github.com/KWB-R/kwb.context

RemoteRef HEAD

RemoteSha 55f830caf99af08d82ed87cd3534b2d905eb6fb0

Contents

`example_crossline` *Example function calling abline*

Description

Example function calling abline

Usage

```
example_crossline(...)
```

Arguments

`...`	arguments passed to `abline`

`example_one_plot` *Example function calling example_crossline twice*

Description

Example function calling example_crossline twice

Usage

```
example_one_plot(..., h = 5, v = 5)
```

Arguments

`...`	arguments passed to `plot`
`h`	position of horizontal line
`v`	position of vertical line

`example_three_plots` *Example function calling example_one_plot three times*

Description

Example function calling example_one_plot three times

Usage

```
example_three_plots()
```

`getContext` *Get the context tree (currently from a global variable "CONTEXT`n`")*

Description

Get the context tree (currently from a global variable "CONTEXT`n`")

Usage

```
getContext(number = getDefault("getContext", "number"))
```

Arguments

`number`	Context number (see `initContext`)

Value

a list of function calls

`getFunctionCall` *Get a function call from the context tree*

Description

Get a function call from the context tree

Usage

```
getFunctionCall(i, number = getDefault("getFunctionCall", "number"))
```

Arguments

`i`	Index of the function call as it is stored in the context
`number`	Number of the context (see `initContext`)

`initContext` *Initialise the variable storing the full function call context*

Description

Initialise the variable storing the full function call context

Usage

```
initContext(number = getDefault("initContext", "number"))
```

Arguments

`number`	Number of the context. You may use different contexts that are numbered. The default context number is 1.

`printContext` *Print the context of a function call*

Description

Print the context of a function call

Usage

```
printContext(number = getDefault("printContext", "number"), indent = "  ")
```

Arguments

`number`	Number of the context to print (see `initContext`)
`indent`	string used to indent nested function calls in the output

`recall` *Recall a function call that is stored in the context tree*

Description

Recall a function of which the call is stored in the context tree. You may specify argument settings that override the argument settings of the original call by passing `key = value` pairs to this function.

Usage

```
recall(funCall = getFunctionCall(i), i = toIndex(id), id = "", ...)
```

Arguments

`funCall`	Function call to be re-called. By default it is taken from the context tree at index `i`
`i`	Index of the function call in the context tree
`id`	As an alternative to specifying the index `i` of the call in the call tree you can specify it by its ID (as shown by when calling `printContext`.
`...`	argument = value pairs overriding the argument setting of the stored function call.

Examples

```
## Not run:

# Load the required libraries
library(kwb.context)
library(kwb.default) # for setDefault()

# Set the default context number to 2 (required until now!)
funNames <- c(paste0(c("init", "print", "get"), "Context"), "getFunctionCall")
for (funName in funNames) {
  setDefault(funName, number = 2)
}

# Initialise the context tree (currently it's a global list called "CONTEXT1")
initContext()

# Call a function that contains a call of logcall() or that calls other
# functions that contain calls of logcall(). The logcall()-calls have the
# effect that they save the calling context in the global context tree
oldpar <- par(mfrow = c(1, 3), no.readonly = TRUE)
example_three_plots()

# Print the context tree that has been recorded
printContext()
```

```
# Change the order of the original plots
recall(i = 8)
recall(i = 5)
recall(i = 2)

# Reset the graphical parameters
par(oldpar)

# Replot the second plot, now on its own page, but with a different colour,
# plot symbol and size
recall(i = 5, col = "red", pch = 16, cex = 2)

## End(Not run)
```

Index