

Package ‘boomer’

July 20, 2021

Title Debugging Tools to Inspect the Intermediate Steps of a Call

Version 0.1.0

Description Provides debugging tools that let you inspect the intermediate results of a call. The output looks as if we explode a call into its parts hence the package name.

License GPL-3

Encoding UTF-8

Language en

RoxygenNote 7.1.1.9001

Imports crayon, methods, pryr, rlang, rstudioapi, styler, withr

Suggests lobstr, magrittr, testthat (>= 3.0.0)

URL <https://github.com/moodymudskipper/boomer>

BugReports <https://github.com/moodymudskipper/boomer/issues>

Config/testthat/edition 3

NeedsCompilation no

Author Antoine Fabri [aut, cre]

Maintainer Antoine Fabri <antoine.fabri@gmail.com>

Repository CRAN

Date/Publication 2021-07-20 10:10:02 UTC

R topics documented:

boomer-package	2
boom	3
Index	5

Description

Provides debugging tools that let you inspect the intermediate results of a call. The output looks as if we explode a call into its parts hence the package name.

Details

- `boom()` displays the intermediate results of a call or a code chunk.
- `rig()` creates a copy of a function which will display the intermediate results of all the calls of it body.
- `rig_in_namespace()` rigs a namespaced function in place, so its always verbose even when called by other existing functions. It is especially handy for package development.
- `rigger()` provides a convenient way to rig an anonymous function by using the `rigger(...)` + `function(...){...}` syntax.
- The addin "Explode a call with 'boom()'" provides a way to boom() a call with a keyboard shortcut.

Package options

Several options impact the display of exploded calls :

- `boomer.print`: If the `print` argument is not provided, this option will replace it at run time. Defaults to the `base::print` function.
- `boomer.clock`: If the `clock` argument is not provided, this option will replace it at run time. Defaults to `FALSE`.
- `boomer.print_args`: Whether to print the arguments of rigged functions and their values when they are evaluated. Defaults to `TRUE`.
- `boomer.visible_only`: Whether to hide the output of functions which return invisibly. Defaults to `FALSE`.
- `boomer.ignore`: Vector of functions for which we don't want the result printed (usually because it's redundant). Defaults to `c("~", "{", "(", "<-", "<<-", "=")`
- `boomer.safe_print`: Whether to replace emoticons by characters compatible with all systems. This is useful for represses (see `reprex` package) and for knitted report in case the output of those doesn't look good on your system.
- `boomer.abbreviate`: Whether to show only the function's name rather than the call when it's entered.

Author(s)

Maintainer: Antoine Fabri <antoine.fabri@gmail.com>

See Also

Useful links:

- <https://github.com/moodymudskipper/boomer>
- Report bugs at <https://github.com/moodymudskipper/boomer/issues>

 boom

Print the Output of Intermediate Steps of a Call

Description

- boom() prints the intermediate results of a call or a code chunk.
- rig() creates a copy of a function which will display the intermediate results of all the calls of it body.
- rig_in_namespace() rigs a namespaced function in place, so its always verbose even when called by other existing functions. It is especially handy for package development.
- rigger() provides a convenient way to rig an anonymous function by using the rigger(...) + function(...) {...} syntax.

Usage

```
boom(expr, clock = NULL, print = NULL)
```

```
rig(fun, clock = NULL, print = NULL)
```

```
rigger(clock = NULL, print = NULL)
```

```
rig_in_namespace(..., clock = NULL, print = NULL)
```

Arguments

expr	call to explode
clock	whether to time intermediate steps. Defaults to <code>getOption("boomer.clock")</code> evaluated at run time (FALSE unless you change it). The execution time of a step doesn't include the execution time of its previously printed sub-steps.
print	A function, a formula or a list of functions or formulas, used to modify the way the output is printed. Defaults to <code>getOption("boomer.print")</code> evaluated at run time (<code>base::print</code> unless you change it)'.
fun	function ro rig()
...	Functions to rig in their namespace

If the print argument is a function, it will be used to print, or to transform the output before it's printed. Use `invisible` to display nothing, useful possibilities are `str` or `dplyr::glimpse`.

rlang's formula notation is supported, so for instance you can type: `print = ~ dplyr::glimpse(., width = 50)`.

Sometimes you might want to print a specific type of object in a custom way, in this case you can provide a named list, if you provide an unnamed element it will be used as the default, and named elements will define how objects of the given S3 class are printed. For instance `print = list(str, data.frame = tibble::as_tibble)`

Value

`boom()` returns the output of the call. `rig()` returns the modified input function. `rig_in_namespace()` returns `invisible(NULL)` and is called for side effects. `rigger()` returns a list containing the arguments, with the class "rigger" to enable `+.rigger` and `print.rigger`

Examples

```
# explode a simple call
boom(subset(head(mtcars, 2), qsec > 17))

# clock calls and customize how to print output
boom(subset(head(mtcars, 2), qsec > 17), clock = TRUE, print = str)

# print str only for data frames
boom(subset(head(mtcars, 2), qsec > 17), print = list(data.frame = str))

# rig an existing function
rig(ave)(warpbreaks$breaks, warpbreaks$wool)

# rig an anonymous function
fun1 <- rigger() + function(x) x + 1 + 2 # same as rig(function(x) x + 1 + 2)
fun1(1)
fun2 <- rigger(TRUE, typeof) + function(x) x + 1 + 2
fun2(1)
```

Index

`_PACKAGE` (boomer-package), 2

`boom`, 3

`boom()`, 2

`boomer` (boomer-package), 2

`boomer-package`, 2

`rig` (boom), 3

`rig()`, 2

`rig_in_namespace` (boom), 3

`rig_in_namespace()`, 2

`rigger` (boom), 3

`rigger()`, 2