Package 'bindr'

October 12, 2022

Title Parametrized Active Bindings

Version 0.1.1

Description Provides a simple interface for creating active bindings where the bound function accepts additional arguments.

Suggests testthat

LazyData true

Date 2018-03-13

BugReports https://github.com/krlmlr/bindr/issues

URL https://github.com/krlmlr/bindr, https://krlmlr.github.io/bindr

RoxygenNote 6.0.1.9000

License MIT + file LICENSE

Encoding UTF-8

NeedsCompilation no

Author Kirill Müller [aut, cre] (<https://orcid.org/0000-0002-1416-3412>),
RStudio [cph, fnd]

Maintainer Kirill Müller <krlmlr+r@mailbox.org>

Repository CRAN

Date/Publication 2018-03-13 08:21:23 UTC

R topics documented:

---

bindr-package   *bindr: Parametrized Active Bindings*

Description

Provides a simple interface for creating active bindings where the bound function accepts additional arguments.

Details

See `create_env()` for creating an environment populated with active bindings, and `populate_env()` for populating an existing environment.

Author(s)

Maintainer: Kirill Müller <krlmlr+r@mailbox.org>

Other contributors:

- RStudio [copyright holder]

See Also

Useful links:

- <https://github.com/krlmlr/bindr>
- <https://krlmlr.github.io/bindr>
- Report bugs at <https://github.com/krlmlr/bindr/issues>

---

create_env   *Create or populate an environment with parametrized active bindings*

Description

Leverages makeActiveBinding() by allowing parametrized functions that take the name of the binding and an arbitrary number of additional arguments.

Usage

```
create_env(names, fun, ..., .envir = parent.frame(),
  .enclos = parent.frame())

populate_env(env, names, fun, ..., .envir = parent.frame())
```

Arguments

names	A [name](#), or a list of names, or a character vector; in the latter case the names are mangled if they are not representable in the native encoding
fun	A [function](#) with at least one argument, which will be called to compute the value of a binding. The function will be called with the binding name as first argument (unnamed), and `...` as additional arguments
...	Additional arguments to `fun`
.envir	The [environment](#) in which `fun` will be executed, important if `fun` calls other functions that are not globally visible
.enclos	The enclosing environment (`parent.env`) for the newly created environment
env	An environment

Examples

```
env <- create_env(letters, paste0, "-lowercase")
env$a
env$c
env$Z
populate_env(env, LETTERS, paste0, "-uppercase")
env$a
env$Z
```

Index