Package 'factory'

October 13, 2022

Type Package

Title Build Function Factories

Version 0.1.0

Description Function factories are functions that make functions. They can be
confusing to construct. Straightforward techniques can produce functions
that are fragile or hard to understand. While more robust techniques exist
to construct function factories, those techniques can be confusing. This
package is designed to make it easier to construct function factories.

URL <https://github.com/jonthegeek/factory>

BugReports <https://github.com/jonthegeek/factory/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports purrr (>= 0.3.2), rlang (>= 0.4.0)

Suggests testthat (>= 2.1.0), covr, roxygen2, knitr, rmarkdown,
ggplot2

VignetteBuilder knitr

NeedsCompilation no

Author Jon Harmon [aut, cre]

Maintainer Jon Harmon <jonthegeek@gmail.com>

Repository CRAN

Date/Publication 2019-08-21 09:00:07 UTC

R topics documented:

---

body_replace *Replace Parts of a Function Body*

Description

Replace quoted targets in the body of a function with quoted replacements.

Usage

```
body_replace(fn_body, target, replacement)
```

Arguments

fn_body	The body of a function (as found via body(fun)).
target	A quoted expression to replace.
replacement	A quoted expression with which the target should be replaced.

Value

A function body with the target replaced anywhere it occurs.

Examples

```
fun <- function(x) {
  x^exp
}
body_replace(body(fun), quote(exp), quote(!!exp))
```

---

build_factory *Easily Build Function Factories*

Description

Easily Build Function Factories

Usage

```
build_factory(fun, ...)
```

Arguments

fun	An anonymous function to turn into a factory.
...	Arguments for the factory function. Things on the RHS will be evaluated before building your factory unless explicitly quoted with quote. See examples.

Value

A function factory.

Examples

```
y <- 2
power <- build_factory(
  fun = function(x) {
    x^exponent
  },
  exponent
)
square <- power(y)
square(2)
y <- 7
square(2)
```

Index