

Package ‘types’

October 15, 2016

Title Type Annotations

Version 1.0.0

Description Provides a simple type annotation for R that is usable in scripts, in the R console and in packages. It is intended as a convention to allow other packages to use the type information to provide error checking, automatic documentation or optimizations.

Depends R (>= 3.0.3)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

NeedsCompilation no

Author Jim Hester [aut, cre]

Maintainer Jim Hester <james.f.hester@gmail.com>

Repository CRAN

Date/Publication 2016-10-15 11:31:52

R topics documented:

| | |
|--------------|----------|
| ? | 1 |
| removeTypes | 3 |
| types | 3 |
| Index | 5 |

?

Documentation Shortcuts

Description

`?<` when used interactively continues to work as before, accessing the R help pages for the topic requested.

When used within a R expression it can be used to add a type annotation. The annotations are syntactically valid R code rather than comments, which provides additional assurance they are specified properly. However this package does not do anything with the type annotations, they have no effect on the calculated result.

Usage

```
"?"(e1, e2)
```

Arguments

| | |
|----|----------------------------|
| e1 | The type of documentation |
| e2 | The topic of documentation |

See Also

[?](#)

Examples

```
# Function arguments can be annotated with types
f <- function(x = 1 ? numeric) {
  x + 1
}

# Default arguments can also be annotated (the `=` is required however)
f <- function(x = ? numeric) {
  x + 1
}

# Function statements can be annotated with types
f <- function(x = "Yay") {
  paste(x, "types!") ? character
}

# Function return values can be annotated with types
f <- function(x = 1) {
  x
} ? boolean
```

| | |
|-------------|--|
| removeTypes | <i>Remove types from a language object</i> |
|-------------|--|

Description

These functions provide the same purpose, to remove type annotation from any language object.

Usage

```
removeTypes(x)
remove_types(x)
```

Arguments

x R language object

Examples

```
# Function arguments can be annotated with types
f <- function(x = 1 ? numeric) {
  x + 1
}

# The types can then be removed if needed
remove_types(f)

# A camelCase function is also available with the same behavior
removeTypes(f)

# They also works on quoted code
remove_types(quote(x + 1 ? numeric))

# Or expressions
remove_types(expression(x + 1 ? numeric))
```

| | |
|-------|---------------------------|
| types | <i>Static types for R</i> |
|-------|---------------------------|

Description

types provides a simple type annotation for R that is usable in scripts, in the R console and in packages. It is intended as a convention to allow other packages to use the type information to provide error checking, automatic documentation or optimizations.

The annotations are syntactically valid R code rather than comments, which provides additional assurance they are specified properly. However this package does not do anything with the type annotations, they have no effect on the calculated result.

'*z*' when used on the command line continues to work as before, accessing the R help pages for the topic requested.

Index

?, [1](#), [2](#)

`remove_types` (`removeTypes`), [3](#)

`removeTypes`, [3](#)

`types`, [3](#)

`types-package` (`types`), [3](#)