

Package ‘templates’

October 14, 2022

Title A System for Working with Templates

Version 0.3.0

Description Provides tools to work with template code and text in R. It aims to provide a simple substitution mechanism for R-expressions inside these templates. Templates can be written in other languages like 'SQL', can simply be represented by characters in R, or can themselves be R-expressions or functions.

Depends R (>= 3.4.0)

Imports stringr, dat, magrittr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests testthat, knitr, rmarkdown, covr

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Sebastian Warnholz [aut, cre]

Maintainer Sebastian Warnholz <wahani@gmail.com>

Repository CRAN

Date/Publication 2018-05-22 21:49:11 UTC

R topics documented:

tmpl	2
tmplUpdate	3
Index	4

tmpl	<i>Template constructors</i>
------	------------------------------

Description

tmpl is the constructor function for template objects.

Usage

```
tmpl(.t, ...)

## S3 method for class 'character'
tmpl(.t, ..., .envir = parent.frame())

## S3 method for class 'formula'
tmpl(.t, ...)

## S3 method for class 'tmpl'
tmpl(.t, ...)

## S3 method for class 'function'
tmpl(.t, ...)
```

Arguments

.t	something that can be interpreted as template. See defined methods for options.
...	(name = value name ~ value) name-value expressions passed on to tmplUpdate
.envir	(environment) the environment in which template snippets are evaluated. For formulas and functions their environment is used.

Details

Objects of class `tmpl` are stored as a character of length one. They can contain 'snippets' to be evaluated. These snippets are identified by an opening `{{` and closing `}}`. The environment in which they are evaluated is stored in the object. They can be further augmented by supplying arguments in

See Also

[tmplUpdate](#), [tmplEval](#)

Examples

```
tmpl("Hi {{ toupper(a) }}!", a = "there")
tmpl( ~ {y <- {{ a }}} , a ~ x + 1)
tmpl(function(x) {{ a }} + x, a ~ 1)
```

templUpdate	<i>Update and evaluate templates</i>
-------------	--------------------------------------

Description

Functions operating on [templ](#) objects. They can be updated and / or evaluated as an expression.

Usage

```
templUpdate(.t, ...)

## S3 method for class 'templ'
templUpdate(.t, ...)

## S3 method for class 'function'
templUpdate(.t, ...)

templEval(.t, ..., .envir = new.env(parent = parent.frame()))

templAsFun(.t, ...)
```

Arguments

<code>.t</code>	(templ) and object of class templ
<code>...</code>	(name = value name ~ value) name-value expressions used to update the snippets in x
<code>.envir</code>	(environment) the environment in which the template is evaluated

Details

`templUpdate` will evaluate all snippets in a template. Objects are searched for in the list of arguments supplied as `...` and the environment of the template. The results are substituted within the snippets. `templEval` will evaluate the template in place or in the specified environment after substituting the elements in `...`

Examples

```
templ("This is {{ a }} very similar to {{ b }}", a = "actually", b = "sprintf")
templ("But I consider it to be ({{ sprintf('%i', a) }}) orthogonal", a = 1.0)
templ("and ({{ sprintf('%i', b) }}) with a different scope:", b = 2.0)
templ("SELECT {{ var }} FROM {{ table }} WHERE {{ condition }};",
      var = "someVar", table = "someTable", condition = "primaryKey = 1")
template <- templ("cat({{ toupper(x) }})")
templUpdate(template, x ~ "hi")
templEval(template, x ~ "hi")
```

Index

tmpl, [2](#), [3](#)
tmplAsFun (tmplUpdate), [3](#)
tmplEval, [2](#)
tmplEval (tmplUpdate), [3](#)
tmplUpdate, [2](#), [3](#)