

Package ‘sticky’

November 1, 2016

Type Package

Title Persist Attributes Across Data Operations

Description In base R, object attributes are lost when objects are modified by common data operations such as subset, filter, slice, append, extract etc. This packages allows objects to be marked as 'sticky' and have attributes persisted during these operations or when inserted into or extracted from recursive (i.e. list- or table-like) objects.

Version 0.4.0

Date 2016-10-31

License GPL-2 | file LICENSE

URL <https://github.com/decisionpatterns/sticky>
<http://www.decisionpatterns.com>

BugReports <https://github.com/decisionpatterns/sticky/issues>

Depends R (>= 3.1.0)

Imports methods

Suggests magrittr (>= 1.5), testthat (>= 0.10.0), data.table (>= 1.9.6), dplyr (>= 0.4.3), knitr, rmarkdown

LazyData TRUE

Collate 'init.R' 'sticky.R' 'append.R' 'extract1.R' 'print.R'
'subset.R' 'utils.R' 'zzz.R'

VignetteBuilder knitr

RoxygenNote 5.0.1.9000

NeedsCompilation no

Author Christopher Brown [aut, cre],
Decision Patterns [cph]

Maintainer Christopher Brown <chris.brown@decisionpatterns.com>

Repository CRAN

Date/Publication 2016-11-01 11:07:41

R topics documented:

.init	2
append	3
print.sticky	3
sticky	4
subset.sticky	5
[.sticky	6
Index	7

.init

Internal, non-exported unit functions

Description

These are internal functions for the sticky package. They are not exported and should not be used directly. Instead use the proper interface, [sticky](#) and [unstick](#)

Usage

```
.init(x)
.deinit(x)
```

Arguments

x vector

Details

.init : initializes the class .deinit : un-initialize the class

These are internal functions not to be called directly, please use the interfaces, [sticky](#) and [unsticky](#).

See Also

[sticky](#) and [unsticky](#)

Examples

```
# - tk
```

append	<i>Append elements to sticky object</i>
--------	---

Description

Append elements to sticky object

Usage

```
append(x, ...)
```

Arguments

x	sticky object
...	additional arguments to base : : append append preserves object attributes by wrapping base : : append and copying objects to the result.

See Also

[append](#)

print.sticky	<i>print a sticky attribute object</i>
--------------	--

Description

print a sticky object which mostly delegates it the the next method.

Usage

```
## S3 method for class 'sticky'  
print(x, ...)
```

Arguments

x	sticky object
...	further arguments passed to or from other methods.

Details

prints a sticky object; this is also the default method for sticky objects

Value

An invisible copy of x.

Examples

```
x <- sticky(1:3)
print(x)
x
```

sticky

sticky

Description

Persist Attributes

Usage

sticky(x)

unstick(x)

is.sticky(x)

Arguments

x objects to have attributes preserved.

... Additional arguments passed to subsequent method(s)

Details

sticky creates objects whose attributes are resilient to subsetting and placing in containers such as lists, data.frames and data.tables.

References

This has been asked and written about numerous times on Stack Overflow. The first has a fairly comprehensive list to others SO discussions.

<http://stackoverflow.com/questions/23991060/loss-of-attributes-despite-attempts-to-preserve-them>

<http://stackoverflow.com/questions/10404224/how-to-delete-a-row-from-a-data-frame-without-losing-t>

<http://stackoverflow.com/questions/23991060/loss-of-attributes-despite-attempts-to-preserve-them>

<http://stackoverflow.com/questions/23841387/approaches-to-preserving-objects-attributes-during-ext>

<http://stackoverflow.com/questions/13432519/indexing-operation-removes-attributes>

Examples

```
x <- 1:5
attr(x, 'foo') <- 'bar'
x <- x[1:3]
attr(x, 'foo')          # NULL

x <- sticky(1:5)
attr(x, 'foo') <- 'bar'
x_1 <- x[1:3]
attr(x_1, 'foo')       # bar

x_2 <- unstick(x)
x_2 <- x[1:3]
attr(x_2, 'foo')       # NULL

is.sticky(x)           # TRUE
```

subset.sticky

Subset elements to sticky object

Description

Subset elements to sticky object

Usage

```
## S3 method for class 'sticky'
subset(x, ...)
```

Arguments

x	sticky object
...	additional arguments to base::subset subset preserves object attributes by wrapping base::subset and copying objects to the result.

See Also

[subset](#)

`[.sticky]`*Extract*

Description

Extract

Usage

```
## S3 method for class 'sticky'  
x[...]
```

Arguments

<code>x</code>	sticky; object to extract from
<code>...</code>	additional parameters

Details

Implements '[' for sticky attributes. This methods ensures that existing attributes are propogated forward.

See Also

[attributes](#), specifically `mostattributes`

Index

`.deinit(.init)`, 2
`.init`, 2
`[".sticky]`, 6

`append`, 3, 3
`attributes`, 6

`is.sticky(sticky)`, 4

`print.sticky`, 3

`sticky`, 2, 4
`subset`, 5
`subset(subset.sticky)`, 5
`subset.sticky`, 5

`unstick`, 2
`unstick(sticky)`, 4