Package ‘BiocBaseUtils’

February 21, 2024

Title  General utility functions for developing Bioconductor packages
Version 1.4.0
Description  The package provides utility functions related to package development. These include functions that replace slots, and selectors for show methods. It aims to coalesce the various helper functions often re-used throughout the Bioconductor ecosystem.
Imports  methods, utils
Depends R (>= 4.2.0)
Suggests knitr, rmarkdown, BiocStyle, tinytest
License Artistic-2.0
Encoding UTF-8
biocViews  Software, Infrastructure

BugReports https://www.github.com/Bioconductor/BiocBaseUtils/issues
Roxygen list(markdown = TRUE)
RoxygenNote 7.2.3
VignetteBuilder knitr
git_url https://git.bioconductor.org/packages/BiocBaseUtils
git_branch RELEASE_3_18
git_last_commit 441d95a
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-02-21

Author Marcel Ramos [aut, cre] (<https://orcid.org/0000-0002-3242-0582>), Martin Morgan [ctb], Hervé Pagès [ctb]

Maintainer Marcel Ramos <marcel.ramos@roswellpark.org>
**R topics documented:**

- BiocBaseUtils-package .................................................. 2
- Assertions ........................................................................... 2
- selectSome .......................................................................... 4
- setSlots ................................................................................. 5

**Index**

7

---

**BiocBaseUtils-package**  
*BiocBaseUtils: Utility and Internal functions for Bioconductor packages*

**Description**

BiocBaseUtils is a package aimed at helping the typical Bioconductor developer formalize often written functions that can be seen scattered throughout the Bioconductor ecosystem. Some of these functions include the ability to replace slots in an object. Other functions work to create a nice show method output by selecting some observations.

**Author(s)**

**Maintainer:** Marcel Ramos <marcel.ramos@roswellpark.org> ([ORCID](#))

Other contributors:

- Martin Morgan <martin.morgan@roswellpark.org> [contributor]
- Hervé Pagès <hpages.on.github@gmail.com> [contributor]

**See Also**

Useful links:


---

**Assertions**

*Suite of helper functions to test for types*

**Description**

These are a group of helper functions that allow the developer to easily check for common data types in Bioconductor. These include logical, character, and numeric (& integer).
Usage

\texttt{isTRUEorFALSE(x, na.ok = FALSE)}

\texttt{isScalarCharacter(x, na.ok = FALSE, zchar = FALSE)}

\texttt{isScalarInteger(x, na.ok = FALSE)}

\texttt{isScalarNumber(x, na.ok = FALSE, infinite.ok = FALSE)}

\texttt{isScalarLogical(x, na.ok = FALSE)}

\texttt{isCharacter(x, na.ok = FALSE, zchar = FALSE)}

\texttt{isZeroOneCharacter(x, na.ok = FALSE, zchar = FALSE)}

Arguments

- \texttt{x} The input vector whose type is to be checked
- \texttt{na.ok} \texttt{logical(1L)} Whether it is acceptable to consider NA type inputs (default: FALSE).
- \texttt{zchar} \texttt{logical(1L)} Whether is is acceptable to consider 'zero' characters as defined by \texttt{nchar}, e.g., \texttt{nchar("")} (default: FALSE).
- \texttt{infinite.ok} \texttt{logical(1L)} Whether it is acceptable to consider infinite values as identified by \texttt{is.finite} (default: FALSE).

Details

Some functions such as \texttt{isScalarCharacter} allow exceptions to the type checks via the \texttt{na.ok} and \texttt{zchar} arguments. Others, for example \texttt{isScalarNumber} can permit Inf with the \texttt{infinite.ok} argument.

Value

Either TRUE or FALSE

Functions

- \texttt{isTRUEorFALSE()}: Is the input a single logical vector?
- \texttt{isScalarCharacter()}: Is the input a single character vector?
- \texttt{isScalarInteger()}: Is the input a single integer vector?
- \texttt{isScalarNumber()}: Is the input a single numeric vector?
- \texttt{isScalarLogical()}: Is the input a single logical vector?
- \texttt{isCharacter()}: Is the input a character vector?
- \texttt{isZeroOneCharacter()}: Is the input a character vector of zero or one length?

Author(s)

M. Morgan, H. Pagès
selectSome

Select and return only some entries from a vector

Description

selectSome works well in show methods. It abbreviates a vector input depending on the maxToShow argument.

Usage

selectSome(
  obj,
  maxToShow = 5,
  ellipsis = "...",
  ellipsisPos = c("middle", "end", "start"),
  quote = FALSE
)

Arguments

<table>
<thead>
<tr>
<th>obj</th>
<th>character() A vector to be abbreviated for display purposes</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxToShow</td>
<td>numeric(1) The maximum number of values to show in the output (default: 5)</td>
</tr>
<tr>
<td>ellipsis</td>
<td>character(1) The symbol used to abbreviate values in the vector (default: &quot;...&quot;)</td>
</tr>
<tr>
<td>ellipsisPos</td>
<td>character(1) The location for the ellipsis in the output, by default in the &quot;middle&quot; but can be moved to either the &quot;end&quot; or the &quot;start&quot;.</td>
</tr>
<tr>
<td>quote</td>
<td>logical(1) Whether or not to add a single quote around the obj input. This only works for character type inputs.</td>
</tr>
</tbody>
</table>
setSlots

Value
An abbreviated output of obj

Author(s)
M. Morgan, H. Pagès

Examples
letters
selectSome(letters)

---

setSlots  

Convenience function to set slot values

Description
Given the current object, the function setSlots will take name-value pair inputs either as named arguments or a list and replace the values of the specified slots. This is a convenient function for updating slots in an S4 class object.

Usage
setSlots(object, ..., check = TRUE)

Arguments
object An S4 object with slots to replace
... Slot name and value pairs either as named arguments or a named list, e.g., slotName = value.
check logical(1L) Whether to run validObject after the slot replacement

Value
The object input with updated slot data

Author(s)
H. Pagès
Examples

setClass("A", representation = representation(slotA = "character"))

aclass <- new("A", slotA = "A")

setSlots(aclass, slotA = "B")
Index

Assertions, 2

BiocBaseUtils-package, 2

isCharacter (Assertions), 2
isScalarCharacter (Assertions), 2
isScalarInteger (Assertions), 2
isScalarLogical (Assertions), 2
isScalarNumber (Assertions), 2
isTRUEorFALSE (Assertions), 2
isZeroOneCharacter (Assertions), 2

replaceSlots (setSlots), 5

selectSome, 4
setSlots, 5