

Package: rstack (via r-universe)

June 26, 2024

Title Stack Data Type as an 'R6' Class

Version 1.0.1.9000

Author Gábor Csárdi

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Description An extremely simple stack data type, implemented with 'R6' classes. The size of the stack increases as needed, and the amortized time complexity is $O(1)$. The stack may contain arbitrary objects.

License MIT + file LICENSE

URL <https://github.com/gaborcsardi/rstack#readme>

BugReports <https://github.com/gaborcsardi/rstack/issues>

RoxygenNote 7.2.3

Imports R6

Suggests covr, testthat

Encoding UTF-8

Roxygen list(r6 = FALSE)

Repository <https://gaborcsardi.r-universe.dev>

RemoteUrl <https://github.com/gaborcsardi/rstack>

RemoteRef HEAD

RemoteSha caa1055a20ae21cbdf2553a74457e43a54d63eee

Contents

stack	2
Index	3

stack

A stack data type, implemented as an R6 class

Description

Methods:

- `push(elem)` Puts an R object on the top of the stack. This operation might trigger the allocation of memory, if the currently allocated memory cannot hold the new element. Still, the amortized time complexity of the stack is constant. The stack may contain arbitrary R objects.
- `pop()` Removes and returns the top element of the stack. It throws an error if the stack is empty.
- `peek()` Returns the top element of the stack. (Without removing it.) It throws an error if the stack is empty.
- `size()` Returns the number of elements in the stack.
- `is_empty()` Returns whether the stack is empty.

Examples

```
S <- stack$new()
S$push(1L)
S$peek()
S$pop()
S$size()

S$push(NULL)
S$push(iris)
colnames(S$peek())
```

Index

stack, [2](#)