

Package: aurrera (via r-universe)

February 24, 2025

Type Package

Title Wrap an Interable in a Progress Bar

Version 0.1.0

Date 2022-05-11

Description Allows a simple creation of progress bars by wrapping the iterated object in 'pb()'.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests covr, foreach, gmp, parallel, plyr, purrr, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/turtletopia/aurrera>

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

RemoteUrl <https://github.com/turtletopia/aurrera>

RemoteRef HEAD

RemoteSha fa24e9873a2897e97eb845feb58b53a5eccb01e7

Contents

| | |
|--------------|----------|
| pb | 2 |
| Index | 4 |

pb

Add a progress bar to an iterable

Description

Makes the object spawn a progress bar in response to indexing with `[]` operator. Intended to be used with most iterators in R, e.g.: `base::lapply()` and some of its cousins, `base::Reduce()`, `plyr::llply()`, `purrr::map()`, `foreach::foreach()`...

Usage

```
pb(object, ...)  
  
## Default S3 method:  
pb(object, ...)  
  
## S3 method for class 'aurrera_iter'  
pb(object, ...)
```

Arguments

| | |
|--------|------------------------------------------------------------------------------------------|
| object | ANY An iterable that should spawn a progress bar. |
| ... | ANY Additional parameters to be used in the future (e.g. progress bar customization). |

Value

The original object with `aurrera_iter` class prepended that handles progress bar display in response to indexing.

Examples

```
result <- lapply(pb(6:20), gmp::factorize)  
  
Reduce(`+`, pb(sample(1e5, 1e4)))  
  
# Progress bar won't display when iterating with `for` loop  
v <- character()  
for (letter in pb(letters[1:5])) {  
  v <- c(letter, v)  
}  
v  
  
# On the other hand, manual indexing works properly  
x <- pb(letters[1:5])  
for (i in seq_along(x)) {  
  # Just be careful with printing inside a progress bar
```

```
    print(x[[i]])  
}
```

Index

base::lapply(), [2](#)

base::Reduce(), [2](#)

foreach::foreach(), [2](#)

pb, [2](#)

plyr::lply(), [2](#)

purrr::map(), [2](#)