Package 'future.callr'

August 9, 2023

Version 0.8.2

Depends R (>= 3.4.0), future (>= 1.33.0)

Imports callr (>= 2.0.3)

Suggests globals, future.apply, listenv, markdown, R.rsp

VignetteBuilder R.rsp

Title A Future API for Parallel Processing using 'callr'

Description Implementation of the Future API on top of the 'callr' package. This allows you to process futures, as defined by the 'future' package, in parallel out of the box, on your local (Linux, macOS, Windows, ...) machine. Contrary to backends relying on the 'parallel' package (e.g. 'future::multisession') and socket connections, the 'callr' backend provided here can run more than 125 parallel R processes.

License LGPL (>= 2.1)

LazyLoad TRUE

URL https://future.callr.futureverse.org, https://github.com/HenrikBengtsson/future.callr

BugReports https://github.com/HenrikBengtsson/future.callr/issues

RoxygenNote 7.2.3

NeedsCompilation no

Author Henrik Bengtsson [aut, cre, cph]

Maintainer Henrik Bengtsson <henrikb@braju.com>

Repository CRAN

Date/Publication 2023-08-09 18:20:06 UTC

R topics documented:

callr *callr futures*

Description

A callr future is an asynchronous multiprocess future that will be evaluated in a background R session.

Usage

```
callr(
  expr,
  envir = parent.frame(),
  substitute = TRUE,
  globals = TRUE,
  label = NULL,
  workers = availableCores(),
  ...
)
```

Arguments

`expr`	The R expression to be evaluated.
`envir`	The environment in which global environment should be located.
`substitute`	Controls whether `expr` should be `substitute()`:d or not.
`globals`	(optional) a logical, a character vector, a named list, or a globals::Globals object. If `TRUE`, globals are identified by code inspection based on `expr` and `tweak` searching from environment `envir`. If `FALSE`, no globals are used. If a character vector, then globals are identified by lookup based their names `globals` searching from environment `envir`. If a named list or a Globals object, the globals are used as is.
`label`	(optional) Label of the future.
`workers`	The number of processes to be available for concurrent callr futures.
`...`	Additional arguments passed to `CallrFuture()`.

Details

callr futures rely on the **callr** package, which is supported on all operating systems.

Value

An object of class CallrFuture.

`future.callr` *future.callr: A Future for callr*

Description

The **future.callr** package implements the Future API on top of **callr**. The Future API is defined by the **future** package.

Details

To use callr futures, load **future.callr**, and select the type of future you wish to use, e.g. `plan(callr)`.

Examples

```
plan(callr)
demo("mandelbrot", package = "future", ask = FALSE)
```

Index