Package: future.callr (via r-universe)

November 23, 2026

Version 1.0.2-9000

Depends R (>= 3.4.0), future (>= 1.80.0)

Imports parallelly, callr

Suggests future.apply, listenv

Title Parallel Future Backend using 'callr'

Description Implementation of the Future API <doi:10.32614/RJ-2021-048> on top of the 'callr' package. The backend is fully compliant with the Future API, as validated by the 'future.tests' conformance test suite, meaning that code using futures gives identical results and behaves the same regardless of which future backend is used. Each future is resolved in a fresh background R session on the local machine, and, contrary to backends relying on the 'parallel' package and socket connections, it can run more than 125 parallel R processes. A common use is to parallelize existing map-reduce calls via the 'futurize' package <doi:10.48550/arXiv.2601.17578>, e.g. 'plan(future.callr::callr)' followed by 'lapply(xs, fcn) |> futurize()' or 'purrr::map(xs, fcn) |> futurize()'.

License Apache License (>= 2)

LazyLoad TRUE

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

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

Language en-US

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.1

Repository https://futureverse-devel.r-universe.dev

Date/Publication 2026-11-23 00:18:19 UTC

RemoteUrl https://github.com/futureverse/future.callr

RemoteRef develop

RemoteSha 51c1fec2f48ab8dbc0c260bffd8d11fb11006669

Contents

callr 2
future.callr 3

Index **5**

`callr` *callr futures*

Description

WARNING: This function must never be called. It may only be used with `future::plan()`

Usage

```
callr(
  ...,
  workers = availableCores(),
  supervise = FALSE,
  envir = parent.frame()
)
```

Arguments

`workers`	The number of processes to be available for concurrent callr futures.
`supervise`	(optional) Argument passed to `callr::r_bg()`.
`envir`	The environment from where global objects should be identified.
`...`	Additional arguments passed to `Future()`.

Details

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

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

Unlike other future backends, callr futures explicitly discard the standard error (stderr) produced while evaluating the future. This is by design in order to avoid 'callr'/'processx' processes from stalling. Note that messages, warnings, and other conditions are still captured and relayed as usual.

Value

An object of class `CallrFuture`.

future.callr *future.callr: Parallel Future Backend using '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, attach **future**, and select the type of future you wish to use, e.g. `plan(future.callr::callr)`.

Example

```
library(futurize)          # also attaches 'future'
plan(future.callr::callr)  # parallelize on the current machine

# A slow function
slow_fcn <- function(x) {
  Sys.sleep(1.0)
  x^2
}

# Example data
xs <- 1:10

# Sequential and parallel base R apply
ys <- lapply(xs, slow_fcn)
ys <- lapply(xs, slow_fcn) |> futurize()

# Sequential and parallel purrr map
library(purrr)
ys <- xs |> map(slow_fcn)
ys <- xs |> map(slow_fcn) |> futurize()

# Sequential and parallel foreach calls
library(foreach)
ys <- foreach(x = xs) %do% slow_fcn(x)
ys <- foreach(x = xs) %do% slow_fcn(x) |> futurize()
```

Author(s)

Maintainer: Henrik Bengtsson <henrikb@braju.com> (ORCID) [copyright holder]

Authors:

- Henrik Bengtsson <henrikb@braju.com> (ORCID) [copyright holder]

See Also

Useful links:

- https://future.callr.futureverse.org
- https://github.com/futureverse/future.callr
- Report bugs at https://github.com/futureverse/future.callr/issues

Index

callr, 2
callr::r_bg(), 2

environment, 2

future.callr, 3
future.callr-package (future.callr), 3
future::plan(), 2