Package ‘rgnparser’

January 25, 2021

Title Parse Scientific Names

Description Parse scientific names using 'gnparser'
   (<https://github.com/gnames/gnparser>), written in Go. 'gnparser' parses scientific names into their component parts; it utilizes a Parsing Expression Grammar specifically for scientific names.

Version 0.2.0

License MIT + file LICENSE


BugReports https://github.com/ropensci/rgnparser/issues

Encoding UTF-8

Language en-US

SystemRequirements gnparser (<https://github.com/gnames/gnparser>)

Imports sys, tibble, jsonlite, readr

Suggests testthat

RoxygenNote 7.1.1

NeedsCompilation no

Author Scott Chamberlain [aut, cre]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2021-01-25 22:00:05 UTC

R topics documented:

  rgnparser-package ...................................................... 2
  gn_debug ............................................................... 2
  gn_parse ............................................................. 2
  gn_parse_tidy .......................................................... 3
  gn_version ............................................................ 4
  install_gnparser ....................................................... 5
rgrnparser-package  rgnparser

Description
Parse scientific names using gnparser

gn_debug  gn_debug

Description
DEFUNCT

Usage
gn_debug(...)

Arguments
... ignored

gn_parse  gn_parse

Description
extract names using gnparser

Usage

gn_parse(
  x,
  threads = NULL,
  batch_size = NULL,
  ignore_tags = FALSE,
  details = FALSE
)
Arguments

- **x** (character) vector of scientific names. required
- **threads** (integer/numeric) number of threads to run. CPU’s threads number is the default. default: 4
- **batch_size** (integer/numeric) maximum number of names in a batch send for processing. default: NULL
- **ignore_tags** (logical) ignore HTML entities and tags when parsing. default: FALSE
- **details** (logical) Return more details for a parsed name

Value

- a list

Examples

```r
trys <- function(x) try(x, silent=TRUE)
if (interactive()) {
  x <- c("Quadrella steyermarkii (Standl.) Iltis & Cornejo",
         "Parus major Linnaeus, 1788", "Helianthus annuus var. texanus")
  trys(gn_parse(x[1]))
  trys(gn_parse(x[2]))
  trys(gn_parse(x[3]))
  trys(gn_parse(x))
  # details
  w <- trys(gn_parse(x, details = TRUE))
  w[[1]]$details # details for one name
  lapply(w, "[[", "details") # details for all names
  z <- trys(gn_parse(x, details = FALSE)) # compared to regular
  z
}
```

---

Description

extract names using gnparser into a tidy tibble

Usage

```r
gn_parse_tidy(x, threads = 4, batch_size = NULL, ignore_tags = FALSE)
```
Arguments

- **x** (character) vector of scientific names. required
- **threads** (integer/numeric) number of threads to run. CPU’s threads number is the default. default: 4
- **batch_size** (integer/numeric) maximum number of names in a batch send for processing. default: NULL
- **ignore_tags** (logical) ignore HTML entities and tags when parsing. default: FALSE

Details

This function focuses on a data.frame result that’s easy to munge downstream - note that this function does not do additional details as does `gn_parse()`.

Value

a data.frame

Examples

```r
trys <- function(x) try(x, silent=TRUE)
if (interactive()) {
  x <- c("Quadrella steyermarkii (Standl.) Iltis & Cornejo",
          "Parus major Linnaeus, 1788", "Helianthus annuus var. texanus")
  trys(gn_parse_tidy(x))
}
```

Description

get gnpars version information

Usage

`gn_version()`

Value

named list, with version and build

Examples

```r
trys <- function(x) try(x, silent=TRUE)
if (interactive()) {
  trys(gn_version())
}
```
install_gnparser

Install gnparser

Description

Downloads the appropriate gnparser executable for your platform and tries to copy it to a system
directory so \texttt{rgnparser} can run the \texttt{gnparser} command.

Usage

\begin{verbatim}
install_gnparser(version = "latest", force = FALSE)
\end{verbatim}

Arguments

- \texttt{version}: The gnparser version number, e.g., 1.0.0; the default \texttt{latest} means the latest
  version (fetched from GitLab releases). Alternatively, this argument can take a
  file path of the zip archive or tarball of gnparser that has already been down-
  loaded from GitLab, in which case it will not be downloaded again. The mini-
  mum version is v1.0.0 because gnparser v1 introduced breaking changes - and
  we don’t support older versions of gnparser here.

- \texttt{force}: Whether to install gnparser even if it has already been installed. This may be
  useful when upgrading gnparser.

Details

This function tries to install gnparser to `Sys.getenv('APPDATA')` on Windows, `~/Library/Application Support` on macOS, and `~/bin` on other platforms (such as Linux). If these directories are not writable, the package directory gnparser of \texttt{rgnparser} will be used. If it still fails, you have to install gnparser by yourself and make sure it can be found via the environment variable \texttt{PATH}.

This is just a helper function and may fail to choose the correct gnparser executable for your oper-
ating system, especially if you are not on Windows or Mac or a major Linux distribution. When in
doubt, read the gnparser documentation and install it yourself: https://github.com/gnparser#installation

Note

modified from blogdown::install_hugo
Index

* package
  rgnparser-package, 2

  gn_debug, 2
  gn_parse, 2
  gn_parse(), 4
  gn_parse_tidy, 3
  gn_version, 4

  install_gnparser, 5

rgnparser (rgnparser-package), 2
rgnparser-package, 2