Package ‘rgnparser’

July 23, 2025

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 0.3.0

License MIT + file LICENSE

URL https://docs.ropensci.org/rgnparser/, https://github.com/ropensci/rgnparser

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, lifecycle

Suggests testthat

RoxygenNote 7.2.3

NeedsCompilation no

Author Scott Chamberlain [aut] (ORCID: <https://orcid.org/0000-0003-1444-9135>), Joel H. Nitta [aut, cre] (ORCID: <https://orcid.org/0000-0003-4719-7472>), Alban Sagouis [aut] (ORCID: <https://orcid.org/0000-0002-3827-1063>)

Maintainer Joel H. Nitta <joelnitta@gmail.com>

Repository CRAN

Date/Publication 2023-12-12 11:20:02 UTC

Contents

`rgnparser-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 = 1,
  batch_size = NULL,
  ignore_tags = FALSE,
  cultivar = FALSE,
  capitalize = FALSE,
  diaereses = FALSE,
  details = FALSE
)
```

Arguments

`x`	(character) vector of scientific names. required
`threads`	(integer/numeric) number of threads to run for parallel processing. Setting to `NULL` will use all threads available. default: `1`
`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`
`cultivar`	(logical) adds support for botanical cultivars like `Sarracenia flava 'Maxima'` and graft-chimaeras like + `Crataegomespilus`. default: `FALSE`
`capitalize`	(logical) capitalizes the first letter of name-strings. default: `FALSE`
`diaereses`	(logical) preserves diaereses within names, e.g. `Leptochloöpsis virgata`. The stemmed canonical name will be generated without diaereses. default: `FALSE`
`details`	(logical) Return more details for a parsed name

Value

a list

Examples

```
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
}
```

`gn_parse_tidy` *gn_parse_tidy*

Description

extract names using gnparser into a tidy tibble

Usage

```
gn_parse_tidy(
  x,
  threads = 1,
  batch_size = NULL,
  cultivar = FALSE,
  capitalize = FALSE,
  diaereses = FALSE,
  ignore_tags = FALSE
)
```

Arguments

`x`	(character) vector of scientific names. required
`threads`	(integer/numeric) number of threads to run for parallel processing. Setting to `NULL` will use all threads available. default: `1`
`batch_size`	(integer/numeric) maximum number of names in a batch send for processing. default: `NULL`
`cultivar`	(logical) adds support for botanical cultivars like `Sarracenia flava 'Maxima'` and graft-chimaeras like `+ Crataegomespilus`. default: `FALSE`
`capitalize`	(logical) capitalizes the first letter of name-strings. default: `FALSE`
`diaereses`	(logical) preserves diaereses within names, e.g. `Leptochloöpsis virgata`. The stemmed canonical name will be generated without diaereses. default: `FALSE`
`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

```
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))
}
```

`gn_version` *gn_version*

Description

get gnparser version information

Usage

```
gn_version()
```

Value

named list, with `version` and `build`

Examples

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

`install_gnparser` *Install gnparser*

Description

[Deprecated]

Reason for deprecating:

The function used to download the appropriate `gnparser` executable for your platform and try to copy it to a system directory so **rgnparser** can run the `gnparser` command. This function was deprecated to stay in compliance with CRAN policies ("Packages should not write in the user's home filespace, nor anywhere else on the file system apart from the R session's temporary directory")

Solution:

Please install `gnparser` by hand. For Linux and Mac users, installing with your usual package manager such as homebrew is the easiest, see `gnparser` documentation for more details: `https://github.com/gnames/gnparser#installation`

Usage

```
install_gnparser(version, force)
```

Arguments

`version`	The gnparser version number, e.g., `1.0.0`; the default `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 downloaded from GitLab, in which case it will not be downloaded again. The minimum version is `v1.0.0` because gnparser v1 introduced breaking changes - and we don't support older versions of gnparser here.
`force`	Whether to install gnparser even if it has already been installed. This may be useful when upgrading gnparser.

Index