Package 'rgnparser'

November 5, 2020

Title Parse Scientific Names

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

Version 0.1.0

License MIT + file LICENSE

URL https://ropensci.github.io/rgnparser/, https://github.com/ropensci/rgnparser

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

Encoding UTF-8

Language en-US

SystemRequirements gnparser (<https://gitlab.com/gogna/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 2020-11-05 07:40:06 UTC

R topics documented:

`rgnparser-package` *rgnparser*

Description

Parse scientific names using gnparser

---

gn_debug *gn_debug*

Description

debug parsing: get complete syntax tree for names

Usage

```
gn_debug(x, threads = 4)
```

Arguments

x	(character) vector of scientific names
threads	(integer/numeric) number of threads to run. CPU's threads number is the default. default: 4

Value

cats output to screen

Examples

```
trys <- function(x) try(x, silent=TRUE)
z <- c("Quadrella steyermarkii (Standl.) Iltis &amp;amp; Cornejo",
  "Parus major Linnaeus, 1788", "Helianthus annuus var. texanus")
if (interactive()) {
trys(gn_debug(z[1]))
}
```

gn_parse *gn_parse*

Description

extract names using gnparser

Usage

```
gn_parse(x, format = "compact", threads = NULL)
```

Arguments

x	(character) vector of scientific names
format	(logical) one of "compact" (default) or "pretty"
threads	(integer/numeric) number of threads to run. CPU's threads number is the default. default: 4

Value

a list

Examples

```
trys <- function(x) try(x, silent=TRUE)
if (interactive()) {
x <- c("Quadrella steyermarkii (Standl.) Iltis &amp;amp; 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))
}
```

gn_parse_tidy *gn_parse_tidy*

Description

extract names using gnparser into a tidy tibble

Usage

```
gn_parse_tidy(x, threads = 4)
```

Arguments

x	(character) vector of scientific names
threads	(integer/numeric) number of threads to run. CPU's threads number is the default. default: 4

Value

a data.frame

Examples

```
trys <- function(x) try(x, silent=TRUE)
if (interactive()) {
x <- c("Quadrella steyermarkii (Standl.) Iltis &amp;amp; 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

Download the appropriate gnparser executable for your platform and try to copy it to a system directory so **rgnparser** can run the `gnparser` command.

Usage

```
install_gnparser(version = "latest", force = FALSE)
```

Arguments

`version`	The gnparser version number, e.g., 0.14.1; 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.
`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 **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 `PATH`.

This is just a helper function and may fail to choose the correct gnparser executable for your operating 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://gitlab.com/gogna/gnparser#installation

Note

modified from `blogdown::install_hugo`

Index