Package 'neotoma2'

November 2, 2026

Title Working with the Neotoma Paleoecology Database

Date 2026-11-02

Version 1.1.1

URL https://github.com/NeotomaDB/neotoma2

BugReports https://github.com/NeotomaDB/neotoma2/issues

Description Access and manipulation of data using the Neotoma Paleoecology Database. <https://api.neotomadb.org/api-docs/>.
Examples in functions that require API access are not executed during CRAN checks.
Vignettes do not execute as to avoid API calls during CRAN checks.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Suggests Bchron, covr, ggplot2, httptest, knitr, pkgdown, rmarkdown, testthat

Imports rlang, sf, leaflet, magrittr, digest, geojsonsf, purrr, httr, jsonlite, methods, dplyr, assertthat, stringr, progress, uuid, utils, tidyr

VignetteBuilder knitr

NeedsCompilation no

Author Socorro Dominguez Vidana [aut, cre] (ORCID: <https://orcid.org/0000-0002-7926-4935>), Simon Goring [aut] (ORCID: <https://orcid.org/0000-0002-2700-4605>), Nick Hoffman [aut] (ORCID: <https://orcid.org/0000-0002-6626-5015>)

Maintainer Socorro Dominguez Vidana <dominguezvid@wisc.edu>

Repository CRAN

Date/Publication 2026-11-02 22:40:02 UTC

Contents

add_chronology . . . 4
as.data.frame,chronology-method . . . 5
author_classes . . . 6
c,chronologies-method . . . 7
check_contacts . . . 8
chroncontrols . . . 8
chroncontrols,chronologies-method . . . 9
chroncontrols,chronology-method . . . 9
chroncontrols,site-method . . . 10
chroncontrols,sites-method . . . 10
chronologies . . . 11
chronologies,collunit-method . . . 11
chronologies_classes . . . 12
cite_data . . . 12
cite_data,sites-method . . . 13
clean . . . 13
collunits . . . 15
collunits,sites-method . . . 15
collunits_classes . . . 16
contacts_classes . . . 16
coordinates . . . 16
coordinates,sites-method . . . 17
count . . . 17
dataset-class . . . 18
datasets . . . 18
datasets,collunits-method . . . 19
doi . . . 19
doi,sites-method . . . 20
filter . . . 20
getids . . . 22
get_contacts . . . 23
get_datasets . . . 24
get_documentation . . . 27
get_downloads . . . 27
get_manual . . . 30
get_publications . . . 30
get_sites . . . 32
get_speleothems . . . 34
get_stats . . . 35
get_table . . . 36
get_taxa . . . 37
get_taxon . . . 38
length,chronologies-method . . . 39
missingOrNULL-class . . . 40
names,collunit-method . . . 40
pingNeotoma . . . 41

plot . . . 42
plotLeaflet . . . 42
plotLeaflet,sites-method . . . 43
publications_classes . . . 44
repositories_classes . . . 44
samples . . . 45
samples,sites-method . . . 45
samples_classes . . . 46
set_chronology . . . 47
set_collunit . . . 48
set_contact . . . 50
set_dataset . . . 51
set_default . . . 53
set_default,chronologies-method . . . 53
set_publication . . . 54
set_sample . . . 56
set_server . . . 57
set_site . . . 58
set_speleothem . . . 59
show,chronology-method . . . 61
sites_classes . . . 62
specimens_classes . . . 62
speleothemdetails . . . 63
speleothemdetails,sites-method . . . 63
speleothems . . . 64
speleothems,sites-method . . . 65
speleothems_classes . . . 65
summary,sites-method . . . 66
taxa . . . 66
taxa,sites-method . . . 67
taxa_classes . . . 67
toWide . . . 68
[,collunits,numeric-method . . . 69
[<-,collunit,character-method . . . 70
[[,chronologies,numeric-method . . . 71
[[<-,collunits-method . . . 72
$,chronology-method . . . 73
$<-,chronology-method . . . 74

Index **76**

add_chronology *Add a new chronology into an existing collectionunit.*

Description

Given a collunit, add a new chronology object to the unit with both the chronology metadata and the age information (as y)

Usage

```
add_chronology(object, x, y)

## S4 method for signature 'collunit,chronology,data.frame'
add_chronology(object, x, y)
```

Arguments

object	A collection unit object
x	A chronology object generated using set_chronology()
y	A data.frame of sample ages, with required columns: "analysisunitid", "age", "agetype", "ageolder", and "ageyounger".

Details

When undertaking analysis we may wish to add a new chronology to existing records within Neotoma. To do this we must first build the chronology, but also link it to existing analysis units within the collection unit. For examples from this function, see the https://open.neotomadb.org/EPD_binder/complex_workflo documentation online.

Value

chronologies with new added chronology

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

as.data.frame,chronology-method
as.data.frame

Description

Returns neotoma2 object's data as a data.frame.

Usage

```
## S4 method for signature 'chronology'
as.data.frame(x)

## S4 method for signature 'chronologies'
as.data.frame(x)

## S4 method for signature 'collunit'
as.data.frame(x)

## S4 method for signature 'collunits'
as.data.frame(x)

## S4 method for signature 'contact'
as.data.frame(x)

## S4 method for signature 'contacts'
as.data.frame(x)

## S4 method for signature 'dataset'
as.data.frame(x)

## S4 method for signature 'datasets'
as.data.frame(x)

## S4 method for signature 'authors'
as.data.frame(x)

## S4 method for signature 'publication'
as.data.frame(x)

## S4 method for signature 'publications'
as.data.frame(x)

## S4 method for signature 'site'
as.data.frame(x)

## S4 method for signature 'sites'
```

```
as.data.frame(x)

## S4 method for signature 'speleothem'
as.data.frame(x)

## S4 method for signature 'speleothems'
as.data.frame(x)

## S4 method for signature 'taxon'
as.data.frame(x)

## S4 method for signature 'taxa'
as.data.frame(x)
```

Arguments

x	neotoma2 object

Value

data.frame object

---

author_classes *An S4 class for the authors of a Neotoma publication.*

Description

This class combines the S4 class contact with a numeric author order. This allows us to reuse contact objects, and to assign the authorship order within a publication. The full set of authors for a publication are represented by the authors object.

Value

object of class author

Examples

```
{
  simon <- new("contact", familyname = "Goring", givennames = "Simon J.")
  firstauthor <- new("author", author = simon, order = 1)
}
```

c,chronologies-method *c - Combine* neotoma2 *objects*

Description

c - Combine neotoma2 objects

Usage

```
## S4 method for signature 'chronologies'
c(x, y)

## S4 method for signature 'collunits'
c(x, y)

## S4 method for signature 'contacts'
c(x, y)

## S4 method for signature 'contact'
c(x, y)

## S4 method for signature 'datasets'
c(x, y)

## S4 method for signature 'publications'
c(x, y)

## S4 method for signature 'samples'
c(x, y)

## S4 method for signature 'sites'
c(x, y)

## S4 method for signature 'missingOrNULL'
c(x = "missingORNULL", y)

## S4 method for signature 'speleothems'
c(x, y)

## S4 method for signature 'taxa'
c(x, y)
```

Arguments

x	neotoma2 object or NULL
y	neotoma2 object or NULL

Value

concatenated and cleaned `sites` object

`check_contacts` *Check contact information for a record against Neotoma contributors*

Description

Uses the Neotoma API to search and access information about individuals who have contributed to the data in the Neotoma Paleoecology Database

Usage

```
check_contacts(x, ...)

## S3 method for class 'contacts'
check_contacts(x, similarity = 0.5, ...)
```

Arguments

`x`	contacts A `contacts` object associated with a set of names.
`...`	Additional parameters associated with the call.
`similarity`	The similarity score between matched records (from 0 - 1).

Value

contacts object

Author(s)

Simon Goring <goring@wisc.edu>

`chroncontrols` *chroncontrols*

Description

Show the samples table

Usage

```
chroncontrols(x)
```

Arguments

`x`	Sites object to extract chroncontrols table from

Value

data.frame with chroncontrols information

`chroncontrols,chronologies-method`
Recover the chron controls for a set of chronologies.

Description

For a `chronologies` object return the chronological controls used in building each chronology, labelled by chronology.

Usage

```
## S4 method for signature 'chronologies'
chroncontrols(x)
```

Arguments

`x`	chronologies object

Value

data.frame with chronological controls

`chroncontrols,chronology-method`
Recover the chron controls for a single chronology.

Description

For a `chronology` object return the chronological controls used in building it.

Usage

```
## S4 method for signature 'chronology'
chroncontrols(x)
```

Arguments

`x`	chronology object

Value

data.frame with chronological controls

`chroncontrols,site-method`
Recover information about the chron controls for a collectionunit.

Description

For a site that includes collection units with chronologies return the chronological controls that are used in building the chronology.

Usage

```
## S4 method for signature 'site'
chroncontrols(x)
```

Arguments

x	site object

Value

data.frame with chronological controls

`chroncontrols,sites-method`
Recover information about the chron controls for a collectionunit.

Description

For all sites that includes collection units with chronologies return the chronological controls that are used in building the chronology.

Usage

```
## S4 method for signature 'sites'
chroncontrols(x)
```

Arguments

x	sites object

Value

data.frame with chronological controls

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

chronologies *Obtain the chronology from a record or multiple records.*

Description

Obtain the chronology from a record or multiple records.

Usage

```
chronologies(x)

## S4 method for signature 'collunits'
chronologies(x)

## S4 method for signature 'site'
chronologies(x)

## S4 method for signature 'sites'
chronologies(x)
```

Arguments

x	sites object that contains chronologies

Value

chronologies object with all chronologies used.

chronologies,collunit-method *Extract* chronologies *from a* sites *or* collunits *object.*

Description

Extract chronologies from a sites or collunits object.

Usage

```
## S4 method for signature 'collunit'
chronologies(x)
```

Arguments

x	A sites or collunits object

Value

chronologies from a collunit object

chronologies_classes *S4 class for* chronologies *information*

Description

The class for chronologies from the Neotoma Paleoecology Database. A single collection unit may have one or more chronology. The individual chronology classes are grouped into an S4 chronologies class.

Value

object of class chronologies

cite_data *Generate a data citation from a Neotoma2 object.*

Description

The function, applied to a data object with a valid dataset, will return a properly formatted data citation for the record.

Usage

```
cite_data(x)

## S4 method for signature 'site'
cite_data(x)

## S4 method for signature 'NULL'
cite_data(x)
```

Arguments

x	Object with DOIs associated to it.

Value

data.frame with citation data

cite_data,sites-method

Obtain data citations from multiple records.

Description

Given complete dataset objects in Neotoma (must have used `get_datasets()` or `get_downloads()`), return a formatted citation for the record, including the dataset DOI.

Usage

```
## S4 method for signature 'sites'
cite_data(x)
```

Arguments

`x`	sites object

Value

`data.frame` object with citation information.

Examples

```
## Not run:
# Get datasets metadata from API
ds <- get_datasets(1)

## End(Not run)
```

clean

clean Neotoma objects to remove duplicates and empty objects.

Description

Function that removes duplicate objects such as sites, datasets, or collection units. When we pull in a large number of objects, or overlapping searches, we can run into a problem where we have multiple instances of the same site, but with different datasets. This function attempts to gather all objects together:

- Before: {site: 1, dataset: 1}, {site: 1, dataset: 2}
- After: {site: 1, dataset: [1, 2]} So the site is gathered, and the datasets are now part of an array of datasets.

Usage

```
clean(x, verbose = TRUE, ...)

## S3 method for class 'sites'
clean(x, verbose = TRUE, ...)

## S3 method for class 'collunits'
clean(x, verbose = TRUE, ...)

## S3 method for class 'datasets'
clean(x, verbose = TRUE, ...)
```

Arguments

x	sites, datasets, collunits that may have duplicates.
verbose	parameter to prints out progress bar
...	Additional parameters associated with the call.

Value

clean neotoma objects without duplicates after concatenation

Author(s)

Simon Goring <goring@wisc.edu>

Examples

```
## Not run:
tryCatch(
  {
    alex <- get_sites(sitename = "Alex%")
    alex2 <- get_sites(24)
    c <- c(alex, alex2) # uncleaned
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)

## End(Not run)
```

collunits *Extract collection units from a sites object*

Description

Extract collection units from a sites object

Usage

```
collunits(object)

## S4 method for signature 'site'
collunits(object)
```

Arguments

object	A sites object

Value

collunits detail from a sites object

collunits,sites-method *Extract* collunits *from a* sites *object.*

Description

Extract collunits from a sites object.

Usage

```
## S4 method for signature 'sites'
collunits(object)
```

Arguments

object	A sites object

Value

collunits from a sites object

collunits_classes *S4 class for* collection units *information.*

Description

A collection unit represents a collection event from within a site. For example, a lake sediment core, or a single dig site within an archaeological site.

Value

object of class collunits

contacts_classes *An S4 class for multi-contact information from the Neotoma Paleoecology Database.*

Description

An unordered list of individual S4 contact objects.

Value

object of class contact

Examples

```
{
  new("contact", familyname = "Goring", givennames = "Simon J.")
}
```

coordinates *Obtain coordinates from a sites object.*

Description

Obtain coordinates from a sites object.

Usage

```
coordinates(obj, ...)
```

Arguments

obj	A sites object
...	Additional parameters associated with the call.

Value

dataframe with coordinate values

`coordinates,sites-method` *Return the latitude and longitude of sites*

Description

Return the latitude and longitude of sites

Usage

```
## S4 method for signature 'sites'
coordinates(obj, ...)
```

Arguments

`obj`	A sites object
`...`	Additional parameters associated with the call.

Value

`data.frame` object with site coordinates.

`count` *Count the number of sites, datasets, or collectionunits associated with a site object.*

Description

Count the number of sites, datasets, or collectionunits associated with a site object.

Usage

```
count(x, level = "sites")

## S4 method for signature 'sites'
count(x, level = c("sites", "collunits", "datasets"))

## S4 method for signature 'site'
count(x, level = c("sites", "collunits", "datasets"))
```

Arguments

x	A sites or site object.
level	The level at which you want to count. Can be sites, collunits, or datasets. Default is sites.

Value

A count of the specified level in the object

---

dataset-class *S4 class for* datasets *information*

Description

The standard object class for datasets from the Neotoma Paleoecology Database.

Value

object of class datasets

---

datasets *Extract datasets from a sites object.*

Description

If the sites object contains datasets, then the datasets will be returned. If the sites object does not contain datasets then the user can apply get_datasets() to the object.

Usage

```
datasets(object)

## S4 method for signature 'collunit'
datasets(object)

## S4 method for signature 'sites'
datasets(object)

## S4 method for signature 'site'
datasets(object)
```

Arguments

object	A sites object

Value

datasets object specific to the metadata contained in datasets

datasets,collunits-method *Extract* datasets *from a* sites *or* collunits *object.*

Description

Extract datasets from a sites or collunits object.

Usage

```
## S4 method for signature 'collunits'
datasets(object)
```

Arguments

object	A sites orcollunits object

Value

datasets object

doi *Obtain the DOI for publications or datasets.*

Description

Obtain the DOI for publications or datasets.

Usage

```
doi(x)

## S4 method for signature 'publication'
doi(x)

## S4 method for signature 'site'
doi(x)
```

Arguments

x	Object with DOIs associated to it.

Value

doi object with DOI information

doi,sites-method *Obtain dataset DOIs from records.*

Description

Given complete dataset objects in Neotoma (must have used `get_datasets()` or `get_downloads()`), return the dataset DOI for the record.

Usage

```
## S4 method for signature 'sites'
doi(x)
```

Arguments

`x`	a Neotoma2 `site` object

Value

`data.frame` object with DOIs information.

Examples

```
## Not run:
# Get datasets metadata from API and retrieve DOIs
ds <- get_datasets(1)
doi(ds)

## End(Not run)
```

filter *Apply a* `filter` *for Neotoma sites objects.*

Description

The `filter` function takes a `sites` object and allows a user to filter on a number of properties. Since a sites object is a nested object (it contains collection units, datasets, samples, etc.) the degree to which filtering occurs depends on the amount of data contained within the sites object. Filtering parameters include:

- `siteid` A numeric site identifier from the Neotoma Database.
- `sitename` The character string sitename.
- `lat` A numeric latitude value.
- `long` A numeric longitude value.

- `altitude` The elevation of the site. Note that some sites do not include elevation information. For these an NA value appears, and they would be removed when using an elevation filter.
- `datasetid` A numeric datasetid from Neotoma.
- `database` A character string naming the constituent database from which the dataset is drawn.
- `datasettype` A character string representing one of the many dataset types within Neotoma.
- `age_range_old` A dataset-level parameter indicating the oldest date covered by the dataset chronology.
- `age_range_young` A dataset-level parameter indicating the youngest date covered by the dataset chronology.
- `notes` Free-form dataset notes provided by the dataset PI(s), analysts or data stewards.
- `collectionunitid` A numeric collection unit identifier from Neotoma.
- `handle` A character string identifying the collection unit. These are often shorter form names (originally a default 8 character length).
- `collectionunitname` A character string identifying the collection unit name.
- `colldate` The date on which the collection unit was sampled. Many of these are empty.
- `location` A free-form character string indicating the location of the collection unit within the site.
- `waterdepth` A numeric depth at which the core was obtained.
- `collunittype` A character string for the collection unit type.
- `collectiondevice` A fixed vocabulary term for the collection device.
- `depositionalenvironment` A fixed vocabulary name for the depositional environment.
- `loc` A spatial filter. Keeps only sites whose geography falls within the supplied region. The region may be an `sf`/`sfc`/`sfg` object, a GeoJSON string, a WKT string (e.g. `"POLYGON ((...))"`), or a numeric bounding box `c(xmin, ymin, xmax, ymax)`, e.g. `filter(altitude < 100, loc == my_polygon)`. `geography` is accepted as an alias for `loc`.

Usage

```
filter(.data, ..., .by = NULL, .preserve = FALSE)

## S3 method for class '`NULL`'
filter(.data, ...)

## S3 method for class 'sites'
filter(.data, ...)

## S3 method for class 'data.frame'
filter(.data, ..., .by = NULL, .preserve = FALSE)
```

Arguments

`.data`	A site, dataset, download, or data frame
`...`	Additional arguments passed to `filter()`
`.by`	(only used for filtering `data.frame` objects)
`.preserve`	(only used for filtering `data.frame` objects)

Value

filtered sites object

Author(s)

Simon Goring <goring@wisc.edu>

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
# Download 10 sites, but only keep the sites that are close to sea level.
tryCatch(
  {
    some_sites <- get_sites(sitename = "Lake%", limit = 3)
    site_subset <- some_sites %>% filter(altitude < 100)
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)
# Download 10 sites, get all associated datasets, but keep only
# sites/datasets that are of datasettype "pollen":
tryCatch(
  {
    sites <- get_sites(limit = 10) %>%
      get_datasets()
    pollen_subset <- sites %>% filter(datasettype == "pollen")
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)

## End(Not run)
```

---

getids *Get object IDs*

Description

This function parses a site object, from site to dataset level and returns a data.frame that contains the site, collectionunit and dataset IDs for each element within the site.

Usage

```
getids(x, order = TRUE)

## S3 method for class 'sites'
getids(x, order = TRUE)

## S3 method for class 'site'
getids(x, order = TRUE)

## S3 method for class 'collunits'
getids(x, order = TRUE)

## S3 method for class 'collunit'
getids(x, order = TRUE)
```

Arguments

x	A Neotoma2 sites or collunits object.
order	sort items by siteid, collunitid, datasetid

Value

data.frame containing siteid, datasetid, and collunitid

Author(s)

Simon Goring <goring@wisc.edu>

Examples

```
## Not run:
marion <- get_sites(sitename = "Marion Lake")
collunitids <- getids(collunits(marion))

## End(Not run)
```

---

get_contacts *Get* contact *information for Neotoma contributors*

Description

Uses the Neotoma API to search and access information about individuals who have contributed to the data in the Neotoma Paleoecology Database

Usage

```
get_contacts(x = NA, ...)

## S3 method for class 'numeric'
get_contacts(x, ...)

## Default S3 method:
get_contacts(x, ...)
```

Arguments

x	integer A contact ID
...	(contactname) A full or partial name for an individual contributor to the database. (familyname) The full or partial last name for an individual contributor to the database. (status) The current status of the contributor (active or retired)

Value

contacts object

Author(s)

Simon Goring <goring@wisc.edu>

Socorro Dominguez <dominguezvid@wisc.edu>

---

get_datasets *get_datasets*

Description

The get_datasets() function is a wrapper for the Neotoma datasets API endpoint. The function takes parameters defined by the user and returns dataset information supplied by the Neotoma Paleoecological Database. The user may define all or none of the possible fields.

Usage

```
get_datasets(x = NA, ...)

## S3 method for class 'numeric'
get_datasets(x, ...)

## Default S3 method:
get_datasets(x, ...)

## S3 method for class 'sites'
get_datasets(x, ...)
```

```
## S3 method for class 'site'
get_datasets(x, ...)

## S3 method for class '`NULL`'
get_datasets(x, ...)
```

Arguments

x	A single datasetid, or a vector of unique dataset ids.
...	accepted arguments, see details for more information.

Details

A dataset is an element nested within neotoma2 site objects. The get_datasets() call returns a list of individual site objects with collunits (collection units) that contain valid, matching dataset elements. So, get_sites() returns only site metadata. get_datasets() returns site metadata, plus metadata about the individual datasets present at that site. The get_datasets() function searches for each site within Neotoma that matches the query parameters, and returns them as a sites object, a list of site objects, plus returns all the additional metadata for the datasets at that site. The get_datasets() command wraps the Neotoma API (api.neotomadb.org) call for datasets. Some queries are slow for the API however few records they return: a spatial query (loc) costs tens of seconds even when it matches no datasets, and a broad taxa filter (many names, or a wildcard) can take a minute or more. Add other filters to narrow a query, test it with a small limit first, and avoid running it in a loop. The call itself uses a SQL query which accepts any one of the following parameters:

- siteid The unique site ID (integer) in Neotoma. Can be passed as a vector of site IDs.
- sitename The site name, or approximate match using the % wildcard.
- database The constituent database for the record. See get_table("constituentdatabases")
- datasettype Neotoma contains data for a number of dataset types. This returns a subset of data types. For a complete list of available dataset types, run neotoma2::get_table('datasettypes')
- altmin The minimum altitude range for site elevation (in meters).
- altmax The maximum altitude range for site elevation (in meters).
- datasetid The unique dataset ID (integer) in Neotoma. Can be passed as a vector of dataset IDs.
- doi The dataset DOI for a dataset contained within a site. Can be passed as a vector of DOIs.
- gpid The geopolitical name or identifier containing a site. Can be passed as a vector of names.
- keywords Keywords for samples within a set of sites. For example "modern" indicates a sample within the record uses the keyword "modern".
- contacts Contact names or IDs associated with a site.
- ageyoung A minimum spanning age for the record, in years before radiocarbon present (1950).
- ageold A maximum spanning age for the record, in years before radiocarbon present (1950).
- ageof An age which must be contained within the range of sample ages for a site.
- taxa The names of taxa which must be present within samples in a record.
- all_data The API only downloads the first 25 records of the query. For the complete records, use all_data=TRUE

Value

The function returns either a single item of class `"try-error"` describing the reason for failure (either misdefined parameters or an error from the Neotoma API), or list of site objects, each containing one or more `collunit` objects, with fully populated `datasets` elements.

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
tryCatch(
  {
    random_sites <- get_sites(1)
    allds <- get_datasets(random_sites, limit = 3)
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)
# To find all datasets with a min altitude of 12 and a max altitude of 25:
tryCatch(
  {
    sites_12to25 <- get_datasets(altmin = 12, altmax = 25)
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)
# To find all datasets in Brazil
brazil <- '{"type": "Polygon",
"coordinates": [[
 [-73.125, -9.102096738726443],
 [-56.953125,-33.137551192346145],
 [-36.5625,-7.710991655433217],
 [-68.203125,13.923403897723347],
 [-73.125,-9.102096738726443]]]}'
tryCatch(
  {
    brazil_datasets <- get_datasets(loc = brazil[1], limit = 2)
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)

## End(Not run)
```

get_documentation *get_documentation*

Description

Open up the Neotoma R homepage.

Usage

```
get_documentation()
```

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
if (interactive()) {
  get_documentation()
}

## End(Not run)
```

get_downloads *get_downloads*

Description

Download `sites` objects up to datum and chronology detail level.

Usage

```
get_downloads(x = NA, ...)

## S3 method for class 'numeric'
get_downloads(x, ...)

## S3 method for class 'sites'
get_downloads(x, verbose = TRUE, ...)

## S3 method for class '`NULL`'
get_downloads(x, ...)
```

Arguments

x	Use a single number to extract site information
...	accepted arguments: sites, datasets
verbose	Status bar of items being downloaded

Details

The `get_downloads()` command wraps the Neotoma API (api.neotomadb.org) call for `downloads`. The call itself uses a SQL query which accepts any one of the following parameters:

- `datasetid` The unique dataset ID (integer) in Neotoma. Can be passed as a vector of dataset IDs.
- `all_data` The API only downloads the first 25 records of the query. For the complete records, use `all_data=TRUE`

Download responses are large, and every request costs the Neotoma API real work. Pass all the dataset IDs you need to **one** call rather than calling `get_downloads()` in a loop: a long vector is split into batches of 25 datasets and requested one after another. The answer for a set of IDs is kept for the rest of the R session, so asking for the same IDs again does not call the API again. Set `options(neotoma2.cache = FALSE)` to turn this off.

Value

The function returns either a single item of class `"try-error"` describing the reason for failure (either misdefined parameters or an error from the Neotoma API), or a table of sites, with rows corresponding to the number of individual sites and datasets returned by the Neotoma API. Each "site" object contains 6 parameters that can be accessed as well:

`siteid`	site ID number
`sitename`	site's name
`location`	sf object that describes site's location
`description`	
`collunits`	limited information on collunits

Each "collection unit" embedded in the "sites" object contains 6 parameters that can be accessed as well:

`collunitid`	collection unit ID number
`handle`	collection unit's handle
`collunitname`	collection unit's name
`colldate`	date in collection unit
`substrate`	substrate
`location`	sf object that describes site's location
`datasets`	detailed information regarding dataset

Each "dataset" nested in the "collection unit" contains the following detail of information:

datasetid	dataset ID number
datasetname	site's name
datasettype	type of data found
location	sf object that describes site's location
notes	notes on the dataset
taxa table	taxa table
pi list	P.I. info
analyst	analyst info
metadata	dataset metadata

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
# To find the downloads object of dataset 24:
tryCatch(
  {
    downloads24 <- get_downloads(24)
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)
# To find all downloads in Brazil
brazil <- '{"type": "Polygon",
"coordinates": [[
 [-73.125, -9.102096738726443],
 [-56.953125,-33.137551192346145],
 [-36.5625,-7.710991655433217],
 [-68.203125,13.923403897723347],
 [-73.125,-9.102096738726443]]]}'
tryCatch(
  {
    brazil_datasets <- get_datasets(loc = brazil[1])
    brazil_downloads <- get_downloads(brazil_datasets)
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)

## End(Not run)
```

get_manual *get_manual*

Description

Open up the Neotoma manual homepage.

Usage

```
get_manual()
```

Value

NULL side effect for opening browser with the manual

Author(s)

Simon Goring <goring@wisc.edu>

Examples

```
## Not run:
# This call does not work from `source()` calls or in testing.
# interactive() just lets us know you are interacting with the console:
if (interactive()) {
  get_manual()
}

## End(Not run)
```

get_publications *Get publication information for Neotoma records*

Description

Uses the Neotoma API to search and access information about publications associated with data in the Neotoma Paleoecology Database

Usage

```
get_publications(x = NA, ...)

## Default S3 method:
get_publications(...)

## S3 method for class 'numeric'
```

```
get_publications(x, ...)

## S3 method for class 'publication'
get_publications(x, ...)

## S3 method for class 'publications'
get_publications(x, ...)

## S3 method for class 'sites'
get_publications(x, ...)
```

Arguments

x	integer A contact ID
...	publicationid The unique numeric identifier associated with a publication in Neotoma. datasetid A unique identifier for a Neotoma dataset that is associated with a publication. familyname The full or partial last name for an individual author. pubtype The publication type, from get_tables("publicationtypes"). year The year the publication was released. search A plain text search string used to search the citation.

Value

publications object

Examples

```
## Not run:
# How old are the papers in Neotoma that include the term "mammut"?
tryCatch(
  {
    mammoth_papers <- get_publications(search = "mammut") %>%
      as.data.frame()
    hist(as.numeric(mammoth_papers$year))
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)
# We want the paper identified in Neotoma as 666:
tryCatch(
  {
    get_publications(666)
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)
# Take a publication object and purposely degrade the metadata:
tryCatch(
  {
```

```
    bad_pub <- get_publications(666)
    # Note this only changes the reported year, not the citation string.
    bad_pub[[1]]@year <- "1923"
    bad_pub[[1]]@publicationid <- NA_integer_
    updated_pubs <- get_publications(bad_pub[[1]])
    attr(updated_pubs, "matches")
    # we see the proper citation in the record:
    updated_pubs <- attr(updated_pubs, "matches")[[3]]
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)

## End(Not run)
```

---

get_sites *get_sites*

Description

The get_sites function is a wrapper for the Neotoma sites API endpoint. The function takes parameters defined by the user and returns a list of site information supplied by the Neotoma Paleoecological Database. The user may define all or none of the possible fields.

Usage

```
get_sites(x = NA, ...)

## S3 method for class 'numeric'
get_sites(x, ...)

## Default S3 method:
get_sites(...)

## S3 method for class '`NULL`'
get_sites(x, ...)
```

Arguments

x	Use a single integer or vector of integers representing unique Neotoma site identifiers (siteids) to extract site information.
...	accepted arguments, see details for more information.

Details

A site object in Neotoma is a physical location at which one or more collection units are located. Each collection unit may have one or more datasets within it, defined by the dataset type. The get_sites() function searches for each site within Neotoma that matches the query parameters, and returns them as a sites object, a list of site objects. The get_sites() command wraps the Neotoma API (api.neotomadb.org) call for sites. Some queries are slow for the API however few records they return: a spatial query (loc) costs tens of seconds even when it matches no sites, and so do wildcard (%) names and broad taxa filters. Add other filters to narrow a query, test it with a small limit first, and avoid running it in a loop. The call itself uses a SQL query which accepts any one of the following parameters:

- siteid The unique site ID (integer) in Neotoma. Can be passed as a vector of site IDs.
- sitename The site name, or approximate match using the % wildcard.
- database The constituent database for the record. See get_table("constituentdatabases")
- altmin The minimum altitude range for site elevation (in meters).
- altmax The maximum altitude range for site elevation (in meters).
- datasetid The unique dataset ID (integer) in Neotoma. Can be passed as a vector of dataset IDs.
- datasettype Neotoma contains data for a number of datasettypes. This returns a subset of data types. For a complete list of available datasettypes, run neotoma2::get_table('datasettypes')
- doi The dataset DOI for a dataset contained within a site. Can be passed as a vector of DOIs.
- gpid The geopolitical name or identifier containing a site. Can be passed as a vector of names.
- keywords Keywords for samples within a set of sites. For example "modern" indicates a sample within the record uses the keyword "modern".
- contacts Contact names or IDs associated with a site.
- ageyoung A minimum spanning age for the record, in years before radiocarbon present (1950).
- ageold A maximum spanning age for the record, in years before radiocarbon present (1950).
- ageof An age which must be contained within the range of sample ages for a site.
- taxa The names of taxa which must be present within samples in a record.
- all_data The API only downloads the first 25 records of the query. For the complete records, use all_data=TRUE This call will then return a data object that contains site metadata for one or more sites, along with limited metadata describing the collection units and datasets located at that site.

Value

The function returns either a single item of class "try-error" describing the reason for failure (either misdefined parameters or an error from the Neotoma API), or a table of sites, with rows corresponding to the number of individual sites returned by the Neotoma API. Each "site" object contains 6 parameters that can be accessed as well: siteid, sitename, location, altitude, description, limited collection units information.

- loc An sf object that describes site's location.
- description
- collunits limited information on collunits

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
## Find sites with a min altitude of 12m and a max altitude of 25m
sites_12to25 <- get_sites(altmin = 12, altmax = 25)
sites_2500 <- get_sites(altmin = 2500, all_data = TRUE)
## To find sites in Brazil
brazil <- '{"type": "Polygon",
"coordinates": [[
 [-73.125, -9.102096738726443],
 [-56.953125,-33.137551192346145],
 [-36.5625,-7.710991655433217],
 [-68.203125,13.923403897723347],
 [-73.125,-9.102096738726443]]]}'
brazil_sites <- get_sites(loc = brazil[1])

## End(Not run)
```

---

get_speleothems *get_speleothems*

Description

get_speleothems

Usage

```
get_speleothems(x = NA, ...)

## S3 method for class 'numeric'
get_speleothems(x, ...)

## S3 method for class 'sites'
get_speleothems(x, ...)
```

Arguments

x	A dataset ID or vector of dataset IDs
...	accepted arguments

Details

Experimental function: API and behavior may change. The `get_speleothems()` command wraps the Neotoma API (api.neotomadb.org) call for `speleothems`. The call itself uses a SQL query which accepts any one of the following parameters:

- `x` The unique dataset ID (integer) in Neotoma. Can be passed as a vector of dataset IDs.
- `sites` A `sites` R object.

Value

`sites` object with speleothem data

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
## Find speleothems by numeric datasetid:
speleo <- get_speleothems(c(2, 5))

## End(Not run)
```

`get_stats` *get_stats*

Description

Returns a count of sites, datasets, publications and other objects added to Neotoma during the requested time period.

Usage

```
get_stats(start = 0, end = 1, type = "dsdbmonth")
```

Arguments

`start`	The starting month (from present == 0) for which to generate the summary. Default is 0 (the current month).
`end`	The ending month (from present == 0) for which to generate the summary. Default is 1 (one month ago).
`type`	A character string indicating the type of summary to return. Options are `dsdbmonth` (the number of datasets in the Neotoma Database added per month), `rawbymonth` (the number of datasets, sites, publications, authors, countries and observations added per month), and `dstypemonth` (the number of datasets added per dataset type per month). Default is `dsdbmonth`.

Details

This function returns summaries about the data holdings within Neotoma using the existing Neotoma API's summary endpoint. This can provide information about recent uploads (the number of new sites uploaded within the last month, for example), or can be used to provide information about the overall number of sites/datasets (using an arbitrarily high value for end).

Value

data.frame with summary statistics

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Simon Goring <goring@wisc.edu>

Examples

```
## Not run:
tryCatch(
  {
    last_month <- get_stats(start = 0, end = 1, type = "dsdbmonth")
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)

## End(Not run)
```

---

get_table *Get table record from Neotoma*

Description

Call Neotoma and return a table (with limits & offsets for large tables)

Usage

```
get_table(x, limit = 25, offset = 0, refresh = FALSE)
```

Arguments

x	Table name (consult https://open.neotomadb.org/dbschema/ for a complete list of table names.
limit	Default 25 records
offset	Default 0.

refresh — Default FALSE. Tables in the Neotoma database change rarely, so the answer to a query is kept for the rest of the R session and asking for the same x, limit and offset again does not call the API again. Use refresh = TRUE to ask the API again. The session cache can be turned off with options(neotoma2.cache = FALSE).

Value

selected table values from the Database.

Examples

```
## Not run:
# Returns only the first 25 specimen records.
someSpec <- get_table("specimens")

## End(Not run)
```

---

get_taxa *get_taxa*

Description

Find the Neotoma datasets that contain a taxon, and return them as a sites object.

Usage

```
get_taxa(x = NA, ...)

## S3 method for class 'numeric'
get_taxa(x, ...)

## Default S3 method:
get_taxa(...)
```

Arguments

x	A taxon ID (integer, or a vector of IDs) in Neotoma.
...	accepted arguments, see details for more information.

Details

get_taxa() works in two steps. First it resolves the taxon you supplied to taxon names, with get_taxon(). Then it asks Neotoma for the datasets that contain those names, with get_datasets(). The second step is the expensive one: the API can take a minute or more to answer a broad taxa= filter, so a taxon ID, or a name with a wildcard, that stands for many names produces a slow request. get_taxa() warns when it is about to send more than 25 distinct names, and it never sends the dataset request without a taxon filter: if no taxon is found it returns NULL.

The taxon can be given in any one of the following ways:

- x A taxon ID (integer) in Neotoma. Can be passed as a vector of IDs.
- taxonname A taxon name, or a partial name using % as a wildcard, which is resolved with get_taxon().
- taxa One or more exact taxon names, used as they are.

Any other argument that get_datasets() accepts (datasettype, loc, ageyoung, ageold, limit, offset, all_data, ...) is passed on to that second step.

Value

A Neotoma2 sites object holding the datasets with the requested taxa, or NULL if no taxon or no dataset was found.

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
# Datasets containing taxon 1 (Abies), at most five of them:
abies <- get_taxa(1, limit = 5)
# The same, starting from the taxon name:
abies <- get_taxa(taxonname = "Abies", limit = 5)

## End(Not run)
```

---

get_taxon *get_taxon*

Description

a taxon object with detailed information

Usage

```
get_taxon(x = NA, ...)

## Default S3 method:
get_taxon(...)

## S3 method for class 'numeric'
get_taxon(x, ...)
```

Arguments

x	taxon ID
...	accepted arguments, see **details** for more information.

Details

A taxa may have one or more taxa associated with it. The get_taxon function searches for taxa detail data within Neotoma while the get_taxa function searches for sites that contain the requested taxa. The function searches for each taxon by using a SQL query which accepts any one of the following parameters:

- taxonid The unique taxon ID (integer) in Neotoma. Can be passed as a vector of taxa IDs.
- taxonname Taxon name or partial name. You can pass wildcard characters using %.
- taxagroup The taxonomic grouping the taxon belongs to, from the Neotoma taxagrouptypes table.
- ecolgroup The ecological group of interest for the Neotoma taxon (from the Neotoma ecolgrouptypes table)
- status The taxonomic extinction status of the taxon, either extinct (1, True) or extant (0, False).

Value

A Neotoma2 taxa object with datasets with the requested taxa.

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

---

length,chronologies-method
length

Description

length

Usage

```
## S4 method for signature 'chronologies'
length(x)

## S4 method for signature 'collunits'
length(x)

## S4 method for signature 'contacts'
length(x)

## S4 method for signature 'datasets'
length(x)

## S4 method for signature 'publications'
```

```
length(x)

## S4 method for signature 'samples'
length(x)

## S4 method for signature 'sites'
length(x)

## S4 method for signature 'speleothems'
length(x)

## S4 method for signature 'taxa'
length(x)
```

Arguments

x	neotoma2 object

Value

int representing length of a neotoma2 object

---

missingOrNULL-class *c Method - Combine objects, including NULL*

Description

c Method - Combine objects, including NULL

---

names,collunit-method *Get a* neotoma2 *object's slot names*

Description

Get all names for elements' slots within a collunit object.

Usage

```
## S4 method for signature 'collunit'
names(x)

## S4 method for signature 'contact'
names(x)

## S4 method for signature 'dataset'
names(x)
```

```
## S4 method for signature 'publication'
names(x)

## S4 method for signature 'publications'
names(x)

## S4 method for signature 'site'
names(x)

## S4 method for signature 'sites'
names(x)

## S4 method for signature 'speleothem'
names(x)

## S4 method for signature 'taxon'
names(x)
```

Arguments

x	A neotoma2 object.

Value

NULL.

---

pingNeotoma *pingNeotoma*

Description

A quick function to test whether or not the Neotoma Database API is currently running.

Usage

```
pingNeotoma(server = "neotoma")
```

Arguments

server	One of localhost:PORT (where PORT is a valid numeric port), neotoma or dev.

Value

A valid HTTP status code or returns an error if a connection is refused.

Examples

```
## Not run:
test_connection <- pingNeotoma("neotoma")

## End(Not run)
```

`plot` *Plot site coordinates using a basic plot.*

Description

Plot site coordinates using a basic plot.

Usage

```
## S4 method for signature 'sites'
plot(x, y, ...)

## S4 method for signature 'site'
plot(x, y, ...)
```

Arguments

`x`	sites object
`y`	*Ignored.*
`...`	Additional parameters associated with the call.

Value

`plot` object with site coordinates.

`plotLeaflet` *plotLeaflet*

Description

Plot sites on a leaflet map

Usage

```
plotLeaflet(object)

## S4 method for signature 'site'
plotLeaflet(object)

## S4 method for signature 'ANY'
plotLeaflet(object)
```

Arguments

object	Sites object to plot

Value

leaflet map with site markers

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

---

plotLeaflet,sites-method
plotLeaflet

Description

Plot sites on a leaflet map

Usage

```
## S4 method for signature 'sites'
plotLeaflet(object)
```

Arguments

object	Sites object to plot

Value

leaflet map

Examples

```
## Not run:
# Note that by default the limit for queries is 25 records:
tryCatch(
  {
    modernSites <- get_sites(keyword = "Modern")
    plotLeaflet(modernSites)
  },
  error = function(e) {
    message("Neotoma server not responding. Try again later.")
  }
)

## End(Not run)
```

publications_classes *An S4 class for Neotoma* publications.

Description

A publication is liked to an individual Neotoma dataset object They are grouped using an S4 publications class. This publications class allows a single dataset to have one or more publication classes associated with it.

Value

object of class publication

Examples

```
{
  simon <- new("contact", familyname = "Goring", givennames = "Simon J.")
  socorro <- new("contact", familyname = "Dominguez", givennames = "Socorro")
  first_author <- new("author", author = simon, order = 1)
  second_author <- new("author", author = socorro, order = 2)
  author_list <- new("authors", authors = list(first_author, second_author))
  pub <- new("publication",
    articletitle = "Top or bottom: Best toast spreading surfaces.",
    journal = "Peanut Butter Science",
    year = "2022",
    volume = "2",
    author = author_list
  )
}
```

repositories_classes *S4 class for* repository *information*

Description

The standard object class for repository from the Neotoma Paleoecology Database.

Value

object of class repository

samples *Obtain samples from a record or multiple records.*

Description

Obtain samples from a record or multiple records.

Usage

```
samples(x, chronname = NULL)

## S4 method for signature 'site'
samples(x, chronname = NULL)

## S4 method for signature 'collunits'
samples(x, chronname = NULL)

## S4 method for signature 'collunit'
samples(x, chronname = NULL)
```

Arguments

x	sites object
chronname	Optional character name of a chronology to report ages against. When NULL (default) the Neotoma default-chronology precedence is applied. If the named chronology is not present for a dataset, the default is used and a warning is issued.

Value

data.frame with record information at sample level

samples,sites-method *samples*

Description

Obtain all samples within a sites object

Usage

```
## S4 method for signature 'sites'
samples(x, chronname = NULL)
```

Arguments

x	sites object
chronname	Optional character name of a chronology to report ages against. When NULL (default) the Neotoma default-chronology precedence is applied. If the named chronology is not present for a dataset, the default is used and a warning is issued.

Value

data.frame with sample records

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Simon Goring <goring@wisc.edu>

Examples

```
## Not run:
# Get full data download from API and create a long table with samples data.
dw <- get_downloads(1)
pollen <- samples(dw)

## End(Not run)
```

---

samples_classes *S4 class for* samples *information*

Description

The standard object class for samples in the Neotoma Paleoecology Database.

Value

object of class sample

`set_chronology` *set chronology information for a new record.*

Description

Create a new chronology for a record. Within Neotoma all chronologies have unique numeric identifiers. Within R, because of the need to use the identifiers across objects, and because we want to avoid conflicts between naming systems, a universally unique identifier (UUID) is created for the object ID.

Function to create new chronology objects for personal analysis. The new object will not be uploaded to the database.

Usage

```
set_chronology(
  x = NA,
  chronologyid = NA_integer_,
  notes = NA_character_,
  contact = list(),
  agemodel = NA_character_,
  ageboundolder = NA_integer_,
  ageboundyounger = NA_integer_,
  isdefault = NA_integer_,
  dateprepared = as.Date(character(0)),
  modelagetype = NA_character_,
  chronologyname = NA_character_,
  chroncontrols = data.frame(0)
)
```

Arguments

`x`	Object to be set as a chronology
`chronologyid`	An optional value. Will be assigned a unique identifier if not provided.
`notes`	Additional notes about the chronology. For more modern models, often the function call to Bacon or Bchron is added here.
`contact`	A contacts object, identifying the individual(s) who created the chronology
`agemodel`	A string representing the age model name, for example "Crummy linear interpolation".
`ageboundolder`	The `ageboundolder` is assigned the oldest sample age rounded up to the nearest 10
`ageboundyounger`	The `ageboundyounger` is assigned the oldest sample age rounded up to the nearest 10
`isdefault`	Defines whether the model is the default for the collection unit for a particular model age type.

dateprepared	The date at which the age model was prepared.
modelagetype	The age type for the model. For validation, the models should be one of the valid Neotoma agetypes: https://api.neotomadb.org/v2.0/data/dbtables?table=agetypes
chronologyname	A valid name for the chronology.
chroncontrols	A data.frame containing the chronological controls for the age model.

Value

chronology object

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Simon Goring <goring@wisc.edu>

---

set_collunit *set Site Information for Fossil Sites*

Description

Function to create new collection unit objects for personal analysis. The new object will not be uploaded to the database.

Usage

```
set_collunit(
  x = NA,
  collectionunitid = NA_integer_,
  notes = NA_character_,
  handle = NA_character_,
  colldate = as.Date(character(1)),
  location = NA_character_,
  waterdepth = NA_integer_,
  gpslocation = st_as_sf(st_sfc()),
  collunittype = NA_character_,
  collectiondevice = NA_character_,
  collectionunitname = NA_character_,
  depositionalenvironment = NA_character_,
  datasets = NULL,
  chronologies = NULL,
  defaultchronology = NA_integer_,
  speleothems = NULL,
  collectors = list()
)
```

Arguments

x	object to be set as collunit
collectionunitid	collection unit identifier
notes	notes
handle	handle
colldate	collection date
location	location of the collection unit
waterdepth	depth at where the sample is taken
gpslocation	location with GPS
collunittype	type of collection unit
collectiondevice	device used to collect the sample
collectionunitname	name of the collection unit
depositionalenvironment	depositional environment
datasets	datasets that the collection unit has
chronologies	chronologies taken from the collection unit
defaultchronology	best chronology model identifier to be used with this collection unit
speleothems	speleothems associated with the collection unit
collectors	list of the names of the contacts who collected the collection unit, in collection order

Value

collunit object

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Simon Goring <goring@wisc.edu>

Examples

```
{
  # Create a collunit
  my_collunit <- set_collunit(notes = "my lake")
}
```

set_contact *Set contact information for a new record.*

Description

Create a new contact for a record. Within Neotoma all chronologies have unique numeric identifiers. Within R, because of the need to use the indentifiers across objects, and because we want to avoid conflicts between naming systems, a universally unique identifier (UUID) is created for the object ID.This is not to be updated to the database.

Usage

```
set_contact(
  x = NA,
  contactid = NA_integer_,
  familyname = NA_character_,
  leadinginitials = NA_character_,
  givennames = NA_character_,
  contactname = NA_character_,
  suffix = NA_character_,
  ORCID = NA_character_,
  title = NA_character_,
  institution = NA_character_,
  email = NA_character_,
  phone = NA_character_,
  contactstatus = NA_character_,
  fax = NA_character_,
  url = NA_character_,
  address = NA_character_,
  notes = NA_character_
)
```

Arguments

x	Object to be set as a contact
contactid	An arbitrary Contact identification number.
familyname	Family or surname name of a person.
leadinginitials	Leading initials for given or forenames without spaces (e.g. G.G.).
givennames	Given or forenames of a person (e.g. George Gaylord). Initials with spaces are used if full given names are not known (e.g. G. G).
contactname	Full name of a person (e.g. George Gaylord Simpson).
suffix	Suffix of a person's name (e.g. Jr., III).
ORCID	A unique ORCID (see https://orcid.org).
title	A person's title (e.g. Dr., Prof., Prof. Dr).

institution	The institution where an individual works.
email	An individuals email address
phone	Their phone number
contactstatus	Are they "active" or "retired"?
fax	Do people still use fax machines?
url	Their homepage
address	A physical address
notes	Notes about the individual

Value

contact object

---

set_dataset *set Site Information for Fossil Sites*

Description

Function to create new dataset objects for personal analysis. The new object will not be uploaded to the database.

Usage

```
set_dataset(
  x = NA,
  datasetid = NA_integer_,
  datasetname = NA_character_,
  database = NA_character_,
  doi = NA,
  recdatecreated = as.Date(character(1)),
  datasettype = NA_character_,
  age_range_old = NA_integer_,
  age_range_young = NA_integer_,
  age_units = NA_character_,
  notes = NA_character_,
  pi_list = NA,
  samples = NULL,
  specimens = NULL
)
```

Arguments

x	object to be set as dataset,
datasetid	dataset identifier
datasetname	name of the dataset
database	dataset where the dataset came from
doi	DOI
recdatecreated	date the dataset was created
datasettype	type the dataset belongs to
age_range_old	age range old
age_range_young	age range young
age_units	age units
notes	notes
pi_list	pi list
samples	taxa objects
specimens	specimens slot

Value

dataset object

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
{
  # Create a dataset
  my_dataset <- set_dataset(
    database = "EPD",
    datasettype = "pollen",
    notes = "my lake"
  )
}
```

set_default *Set the default chronology within a collectionunit.*

Description

Set the default chronology within a collectionunit.

Usage

```
set_default(x, n)
```

Arguments

x	A chronologies object.
n	The particular chronology to be used as the default.

Value

sites object with new default chronology

set_default,chronologies-method *Change the default age model for a record.*

Description

Change the default age model for a record.

Usage

```
## S4 method for signature 'chronologies'
set_default(x, n)
```

Arguments

x	A chronologies object.
n	The particular chronology to be used as the default.

Value

chronologies object with a new defaulted chronology

set_publication *Create a new publication (or publication set)*

Description

A function to create new publication objects by hand.

Usage

```
set_publication(
  x = NA,
  publicationid = NA_integer_,
  publicationtypeid = NA_integer_,
  publicationtype = NA_character_,
  year = NA_character_,
  citation = NA_character_,
  articletitle = NA_character_,
  journal = NA_character_,
  volume = NA_character_,
  issue = NA_character_,
  pages = NA_character_,
  citationnumber = NA_character_,
  doi = NA_character_,
  booktitle = NA_character_,
  numvolumes = NA_character_,
  edition = NA_character_,
  volumetitle = NA_character_,
  seriestitle = NA_character_,
  seriesvolume = NA_character_,
  publisher = NA_character_,
  url = NA_character_,
  city = NA_character_,
  state = NA_character_,
  country = NA_character_,
  originallanguage = NA_character_,
  notes = NA_character_,
  author = NULL
)
```

Arguments

x	"publications" object to be updated.
publicationid	ID of publication
publicationtypeid	ID of kind of publication
publicationtype	A text string identifying the publication type within the Neotoma database.

year	The year of publication.
citation	A full text citation for the article.
articletitle	The title of the article.
journal	The journal in which the article was published.
volume	The journal volume.
issue	The journal issue.
pages	The pages of the journal.
citationnumber	How many times has the paper been cited?
doi	A DOI for the record.
booktitle	The title of the book (if the publication is a book)
numvolumes	The number of book volumes (if a series)
edition	The book edition.
volumetitle	The title of the volume (in a published series)
seriestitle	The title of the series.
seriesvolume	The series volume.
publisher	The publisher.
url	Publication URL
city	City of publication.
state	State of publication.
country	Country of publication.
originallanguage	Original language of publication.
notes	Publication notes.
author	name of the author of publication.

Value

publication object

Author(s)

Simon Goring <goring@wisc.edu>

Socorro Dominguez <dominguezvid@wisc.edu>

set_sample *set Sample Information*

Description

Function to create new samples objects for analysis. The new object will not be uploaded to the database.

Usage

```
set_sample(
  x = NA,
  ages = list(),
  igsn = NA_character_,
  datum = data.frame(),
  depth = NA_integer_,
  sampleid = NA_integer_,
  thickness = NA_integer_,
  samplename = NA_character_,
  sampleanalyst = list(),
  analysisunitid = NA_integer_,
  analysisunitname = NA_character_,
  analysisunitnotes = NA_character_,
  analysisdate = as.Date(NA_character_),
  samplenotes = NA_character_
)
```

Arguments

x	Object to be set as a sample
ages	ages
igsn	IGSN character
datum	dataframe of datum
depth	integer representing depth
sampleid	ID for sample
thickness	thickness of core
samplename	sample's name
sampleanalyst	Analyst's contact name
analysisunitid	Which analysis unit it is
analysisunitname	Analysis Unit's name
analysisunitnotes	Notes about the analysis unit the sample belongs to
analysisdate	Date (class Date) on which the sample was analyzed
samplenotes	Notes about the sample

Value

sample object

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
{
  # Set an empty sample
  my_sample <- set_sample()
}
```

---

set_server *Set Neotoma API Source or Server*

Description

Choose to pull Neotoma data from the main Neotoma server, the development server or from a local instance of the API.

Usage

```
set_server(server = "neotoma")
```

Arguments

server	One of local (when the API is running locally on port 3005), neotoma or dev.

Value

NULL modifies how to talk to the API (local, dev, server)

Author(s)

Simon Goring <goring@wisc.edu>

Examples

```
## Not run:
# The user is running the API locally using the node/express API
# cloned from github: https://github.com/NeotomaDB/api_nodetest
set_server(server = "local")
# The user switches back to the remote api server.
set_server(server = "neotoma")

## End(Not run)
```

set_site *set Site Information for Fossil Sites*

Description

set Site Information for Fossil Sites

Usage

```
set_site(
  x = NA,
  siteid = NA_integer_,
  sitename = NA_character_,
  geography = sf::st_as_sf(sf::st_sfc()),
  altitude = NA_integer_,
  geopolitical = list(),
  area = NA_integer_,
  notes = NA_character_,
  description = NA_character_,
  collunits = NULL
)
```

Arguments

x	Object to be set as a site
siteid	The unique site id for a site. If this site is new to Neotoma then leave the ID as NA.
sitename	Name of a site.
geography	A sf object representing the site location, either as a polygon or point.
altitude	altitude/elevation of the site.
geopolitical	The geopolitical unit in which the site is located.
area	The area of the site or depositional basin in *ha*. Can be calculated from the polygon.
notes	additional information of the site
description	Function to create new site objects. The new object will not be uploaded to the database.
collunits	Collection units in the site

Value

site object

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
{
  # Create a site called "My Lake", to
  x <- sf::st_as_sf(sf::st_sfc(sf::st_point(c(5, 5))))
  my_site <- set_site(
    sitename = "My Lake",
    geography = x,
    description = "my lake",
    altitude = 30
  )
}
```

`set_speleothem` *Set Speleothem*

Description

Set Speleothem

Usage

```
set_speleothem(
  x = NA,
  entityid = NA_integer_,
  entityname = NA_character_,
  siteid = NA_integer_,
  collectionunitid = NA_integer_,
  dripheight = NA_integer_,
  monitoring = FALSE,
  geology = NA_character_,
  rockage = NA_character_,
  speleothemtype = NA_character_,
  dripheightunits = NA_character_,
  entitycovertype = NA_character_,
  entrancedistance = NA_integer_,
  landusecovertype = NA_character_,
  speleothemdriptype = NA_character_,
  landusecoverpercent = NA_integer_,
  vegetationcovertype = NA_character_,
  entitycoverthickness = NA_integer_,
  entrancedistanceunits = NA_character_,
  vegetationcoverpercent = NA_integer_
)
```

Arguments

x	Object to be set as a speleothem

entityid	The speleothem entity ID.
entityname	Name of the speleothem entity.
siteid	The unique site ID for a site.
collectionunitid	The unique collection unit ID for a collection unit.
dripheight	drip height
monitoring	Is the speleothem being monitored? TRUE/FALSE
geology	rock type of the speleothem
rockage	relative age of the speleothem
speleothemtype	type of speleothem
dripheightunits	Units for drip height
entitycovertype	type of cover around the speleothem
entrancedistance	distance from cave entrance
landusecovertype	type of land use cover around the speleothem
speleothemdriptype	type of speleothem drip
landusecoverpercent	land use cover percent
vegetationcovertype	type of vegetation cover around the speleothem
entitycoverthickness	thickness of the entity cover
entrancedistanceunits	Units for distance from cave entrance
vegetationcoverpercent	vegetation cover percent

Value

speleothem object

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

show,chronology-method *Display a* sites *object or nested slots.*

Description

Display a sites object or nested slots.

Usage

```
## S4 method for signature 'chronology'
show(object)

## S4 method for signature 'chronologies'
show(object)

## S4 method for signature 'collunits'
show(object)

## S4 method for signature 'collunit'
show(object)

## S4 method for signature 'contacts'
show(object)

## S4 method for signature 'contact'
show(object)

## S4 method for signature 'dataset'
show(object)

## S4 method for signature 'datasets'
show(object)

## S4 method for signature 'publications'
show(object)

## S4 method for signature 'publication'
show(object)

## S4 method for signature 'site'
show(object)

## S4 method for signature 'sites'
show(object)

## S4 method for signature 'speleothem'
```

```
show(object)

## S4 method for signature 'speleothems'
show(object)

## S4 method for signature 'taxon'
show(object)

## S4 method for signature 'taxa'
show(object)
```

Arguments

object	sites, datasets, collunits, contacts object

`sites_classes` *An S4 class for* `sites` *information*

Description

The S4 class for sites in the Neotoma Paleoecology Database.

Value

object of class `sites`

`specimens_classes` *S4 class for* `specimens` *information*

Description

The standard object class for `specimens` from the Neotoma Paleoecology Database.

Value

object of class `specimens`

speleothemdetails *Obtain speleothems and samples from a record or multiple records.*

Description

Obtain speleothems and samples from a record or multiple records.

Usage

```
speleothemdetails(x)

## S4 method for signature 'site'
speleothemdetails(x)

## S4 method for signature 'collunits'
speleothemdetails(x)

## S4 method for signature 'collunit'
speleothemdetails(x)
```

Arguments

x	sites object

Value

data.frame with record information at speleothem level

---

speleothemdetails,sites-method *speleothemdetails*

Description

Obtain elements on the speleothems level Experimental function: API and behavior may change.

Usage

```
## S4 method for signature 'sites'
speleothemdetails(x)
```

Arguments

x	site object

Value

data.frame with speleothem records

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
kesang <- get_sites(sitename = "Kesang cave") %>%
  get_datasets() %>%
  filter(datasettype == "pollen") %>%
  get_speleothems()
sp <- speleothemdetails(kesang)

## End(Not run)
```

---

speleothems *Obtain speleothems from a record or multiple records.*

Description

Obtain speleothems from a record or multiple records.

Usage

```
speleothems(x)

## S4 method for signature 'site'
speleothems(x)

## S4 method for signature 'collunits'
speleothems(x)

## S4 method for signature 'collunit'
speleothems(x)
```

Arguments

x	sites object

Value

data.frame with record information at speleothem level

speleothems,sites-method *speleothems*

Description

Obtain all speleothems within a sites object Experimental function: API and behavior may change.

Usage

```
## S4 method for signature 'sites'
speleothems(x)
```

Arguments

x	sites object

Value

data.frame with sample records

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
ds <- get_datasets(37302)
sp <- speleothems(ds)

## End(Not run)
```

speleothems_classes *S4 class for* speleothem *information*

Description

The S4 class for speleothem data.

Value

object of class speleothems

summary,sites-method *Summary of objects within a sites object.*

Description

This function summarizes a sites object, from site level and returns a data.frame that contains the site ID, sitename, collectionunit ID, count of chronologies, count of datasets and types of datasets within the site.

Usage

```
## S4 method for signature 'sites'
summary(object, ...)
```

Arguments

object	sites object
...	additional properties passed to summary

Value

data.frame object with site summary information

taxa *taxa*

Description

Show the samples table

Usage

```
taxa(object)

## S4 method for signature 'site'
taxa(object)

## S4 method for signature 'collunits'
taxa(object)

## S4 method for signature 'collunit'
taxa(object)
```

Arguments

object	Sites object to extract taxa table from

Value

data.frame with taxa records

`taxa,sites-method` *Extract taxonomic data from a set of sites.*

Description

Extract taxonomic data from a set of sites.

Usage

```
## S4 method for signature 'sites'
taxa(object)
```

Arguments

`object`	A `sites` object.

Value

A `data.frame` reporting the taxa/data objects, units, elements and other features within a set of records.

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
somesites <- get_sites(datasettype = "diatom") %>%
  get_downloads()
diatomtaxa <- taxa(somesites)

## End(Not run)
```

`taxa_classes` *S4 class for taxa information*

Description

Taxa details from the Neotoma Paleoecology Database.

Value

object of class `taxon`

toWide *toWide*

Description

Obtain a wide table with information regarding of samples grouped by variablename and depth/age.

Usage

```
toWide(
  x,
  variablenames = NA,
  ecologicalgroups = NA,
  elementtypes = NA,
  unit = NA,
  groupby = "age",
  operation = "prop"
)
```

Arguments

x	dataframe object with samples
variablenames	Optional vector to filter by specific variable names.
ecologicalgroups	Vector stating the ecological groups to be filtered by, e.g "DIAT", "TRSH"
elementtypes	Label of element type to filter by, e.g. "pollen", "valve"
unit	Label stating which units to filter by, e.g. "NISP"
groupby	Group by 'age' or 'depth'
operation	label or vector of operations to be chosen from: 'prop', 'sum', 'presence'.

Value

wide `data.frame` obtained from long `samples` `data.frame`

Author(s)

Socorro Dominguez <dominguezvid@wisc.edu>

Examples

```
## Not run:
fc_sites <- neotoma2::get_datasets(limit = 5, datasettype = "vertebrate
fauna")
fc_ds <- fc_sites %>%
  neotoma2::get_downloads()
fc_dl1 <- fc_dl[[1]]
```

```
fc_smp <- samples(fc_dl1)
toWide(fc_smp,
  ecologicalgroups = c("AVES", "RODE"),
  elementtypes = "bone/tooth", unit = "present/absent"
)

## End(Not run)
```

`[,collunits,numeric-method`
sub

Description

sub

Usage

```
## S4 method for signature 'collunits,numeric'
x[i]

## S4 method for signature 'datasets,numeric'
x[i]

## S4 method for signature 'sites,numeric'
x[i]

## S4 method for signature 'site,numeric'
x[i]

## S4 method for signature 'site,character'
x[i]

## S4 method for signature 'speleothems,numeric'
x[i]

## S4 method for signature 'taxa,numeric'
x[i]
```

Arguments

`x`	The `sites` object
`i`	The numeric index

Value

sliced `site` object

`[<-,collunit,character-method` *subset*

Description

subset

Usage

```
## S4 replacement method for signature 'collunit,character'
x[i] <- value

## S4 replacement method for signature 'collunit,numeric'
x[i] <- value

## S4 replacement method for signature 'dataset,character'
x[i] <- value

## S4 replacement method for signature 'dataset,numeric'
x[i] <- value

## S4 replacement method for signature 'sample,character'
x[i] <- value

## S4 replacement method for signature 'site,character'
x[i] <- value

## S4 replacement method for signature 'site,numeric'
x[i] <- value

## S4 replacement method for signature 'speleothem,character'
x[i] <- value

## S4 replacement method for signature 'speleothem,numeric'
x[i] <- value

## S4 replacement method for signature 'taxon,character'
x[i] <- value

## S4 replacement method for signature 'taxon,numeric'
x[i] <- value
```

Arguments

x	A neotoma2 object.
i	The column indicator.

value The value to be used.

Value

site object with reassigned character values

`[[,chronologies,numeric-method`
sub-sub

Description

Obtain one of the elements within a sites, collectionunits, datasets, etc... Neotoma objects.

Usage

```
## S4 method for signature 'chronologies,numeric'
x[[i]]

## S4 method for signature 'collunits,numeric'
x[[i]]

## S4 method for signature 'contacts,numeric'
x[[i]]

## S4 method for signature 'datasets,numeric'
x[[i]]

## S4 method for signature 'publications,numeric'
x[[i]]

## S4 method for signature 'samples,numeric'
x[[i]]

## S4 method for signature 'sites,numeric'
x[[i]]

## S4 method for signature 'speleothems,numeric'
x[[i]]

## S4 method for signature 'taxa,numeric'
x[[i]]
```

Arguments

x	Neotoma2 nested object
i	iteration in nested list

Value

sliced site object

Examples

```
## Not run:
some_site <- get_sites(sitename = "Site%", limit = 3)
some_site[[2]]

## End(Not run)
```

`[[<-,collunits-method` *sub-subset*

Description

Obtain one of the elements within a nested neotoma2 object

Usage

```
## S4 replacement method for signature 'collunits'
x[[i]] <- value

## S4 replacement method for signature 'datasets'
x[[i]] <- value

## S4 replacement method for signature 'publications'
x[[i]] <- value

## S4 replacement method for signature 'samples'
x[[i]] <- value

## S4 replacement method for signature 'sites'
x[[i]] <- value

## S4 replacement method for signature 'speleothems'
x[[i]] <- value

## S4 replacement method for signature 'taxa'
x[[i]] <- value
```

Arguments

x	neotoma2 object
i	iteration in neotoma2 object
value	The value to be used

Value

sites object with reassigned values

$,chronology-method *cash*

Description

Obtain slots of a neotoma2 without using at-mark

Usage

```
## S4 method for signature 'chronology'
x$name

## S4 method for signature 'chronologies'
x$name

## S4 method for signature 'collunit'
x$name

## S4 method for signature 'collunits'
x$name

## S4 method for signature 'contact'
x$name

## S4 method for signature 'contacts'
x$name

## S4 method for signature 'dataset'
x$name

## S4 method for signature 'datasets'
x$name

## S4 method for signature 'publication'
x$name

## S4 method for signature 'samples'
x$name

## S4 method for signature 'sample'
x$name

## S4 method for signature 'site'
```

```
x$name

## S4 method for signature 'sites'
x$name

## S4 method for signature 'speleothem'
x$name

## S4 method for signature 'speleothems'
x$name

## S4 method for signature 'taxon'
x$name

## S4 method for signature 'taxa'
x$name
```

Arguments

x	neotoma2 object
name	name of the slot

Value

value at chosen slot in the site object

`$<-,chronology-method` *cash-set*

Description

cash-set

Usage

```
## S4 replacement method for signature 'chronology'
x$name <- value

## S4 replacement method for signature 'collunit'
x$name <- value

## S4 replacement method for signature 'dataset'
x$name <- value

## S4 replacement method for signature 'site'
x$name <- value
```

```
## S4 replacement method for signature 'speleothem'
x$name <- value

## S4 replacement method for signature 'taxon'
x$name <- value
```

Arguments

x	A neotoma2 object.
name	name of the slot
value	The value to be used.

Value

neotoma2 object with reassigned values

Index

[(`[,collunits,numeric-method`), 69
[,collunits,numeric,ANY-method (`[,collunits,numeric-method`), 69
[,collunits,numeric-method, 69
[,datasets,numeric,ANY-method (`[,collunits,numeric-method`), 69
[,datasets,numeric-method (`[,collunits,numeric-method`), 69
[,site,character,ANY-method (`[,collunits,numeric-method`), 69
[,site,character-method (`[,collunits,numeric-method`), 69
[,site,numeric,ANY-method (`[,collunits,numeric-method`), 69
[,site,numeric-method (`[,collunits,numeric-method`), 69
[,sites,numeric,ANY-method (`[,collunits,numeric-method`), 69
[,sites,numeric-method (`[,collunits,numeric-method`), 69
[,speleothems,numeric,ANY-method (`[,collunits,numeric-method`), 69
[,speleothems,numeric-method (`[,collunits,numeric-method`), 69
[,taxa,numeric,ANY-method (`[,collunits,numeric-method`), 69
[,taxa,numeric-method (`[,collunits,numeric-method`), 69
[<-,collunit,character-method, 70
[<-,collunit,character,ANY-method (`[<-,collunit,character-method`), 70
[<-,collunit,numeric,ANY-method (`[<-,collunit,character-method`), 70
[<-,collunit,numeric-method (`[<-,collunit,character-method`), 70
[<-,dataset,character,ANY-method (`[<-,collunit,character-method`), 70
[<-,dataset,character-method (`[<-,collunit,character-method`), 70
[<-,dataset,numeric,ANY-method (`[<-,collunit,character-method`), 70
[<-,dataset,numeric-method (`[<-,collunit,character-method`), 70
[<-,sample,character,ANY-method (`[<-,collunit,character-method`), 70
[<-,sample,character-method (`[<-,collunit,character-method`), 70
[<-,site,character,ANY-method (`[<-,collunit,character-method`), 70
[<-,site,character-method (`[<-,collunit,character-method`), 70
[<-,site,numeric,ANY-method (`[<-,collunit,character-method`), 70

[<-,site,numeric-method
(`[<-,collunit,character-method`), 70
[<-,speleothem,character,ANY-method
(`[<-,collunit,character-method`), 70
[<-,speleothem,character-method
(`[<-,collunit,character-method`), 70
[<-,speleothem,numeric,ANY-method
(`[<-,collunit,character-method`), 70
[<-,speleothem,numeric-method
(`[<-,collunit,character-method`), 70
[<-,taxon,character,ANY-method
(`[<-,collunit,character-method`), 70
[<-,taxon,character-method
(`[<-,collunit,character-method`), 70
[<-,taxon,numeric,ANY-method
(`[<-,collunit,character-method`), 70
[<-,taxon,numeric-method
(`[<-,collunit,character-method`), 70
[[,chronologies,numeric,ANY-method
(`[[,chronologies,numeric-method`), 71
[[,chronologies,numeric-method, 71
[[,collunits,numeric,ANY-method
(`[[,chronologies,numeric-method`), 71
[[,collunits,numeric-method
(`[[,chronologies,numeric-method`), 71
[[,contacts,numeric,ANY-method
(`[[,chronologies,numeric-method`), 71
[[,contacts,numeric-method
(`[[,chronologies,numeric-method`), 71
[[,datasets,numeric,ANY-method
(`[[,chronologies,numeric-method`), 71
[[,datasets,numeric-method
(`[[,chronologies,numeric-method`), 71
[[,publications,numeric,ANY-method
(`[[,chronologies,numeric-method`), 71
[[,publications,numeric-method
(`[[,chronologies,numeric-method`), 71
[[,samples,numeric,ANY-method
(`[[,chronologies,numeric-method`), 71
[[,samples,numeric-method
(`[[,chronologies,numeric-method`), 71
[[,sites,numeric,ANY-method
(`[[,chronologies,numeric-method`), 71
[[,sites,numeric-method
(`[[,chronologies,numeric-method`), 71
[[,speleothems,numeric,ANY-method
(`[[,chronologies,numeric-method`), 71
[[,speleothems,numeric-method
(`[[,chronologies,numeric-method`), 71
[[,taxa,numeric,ANY-method
(`[[,chronologies,numeric-method`), 71
[[,taxa,numeric-method
(`[[,chronologies,numeric-method`), 71
[[<-,collunits-method, 72
[[<-,collunits,ANY,ANY-method
(`[[<-,collunits-method`), 72
[[<-,datasets,ANY,ANY-method
(`[[<-,collunits-method`), 72
[[<-,datasets-method
(`[[<-,collunits-method`), 72
[[<-,publications,ANY,ANY-method
(`[[<-,collunits-method`), 72
[[<-,publications-method
(`[[<-,collunits-method`), 72
[[<-,samples,ANY,ANY-method
(`[[<-,collunits-method`), 72
[[<-,samples-method
(`[[<-,collunits-method`), 72
[[<-,sites,ANY,ANY-method
(`[[<-,collunits-method`), 72

[[<-,sites-method
(\[\[<-,collunits-method), 72
[[<-,speleothems,ANY,ANY-method
(\[\[<-,collunits-method), 72
[[<-,speleothems-method
(\[\[<-,collunits-method), 72
[[<-,taxa,ANY,ANY-method
(\[\[<-,collunits-method), 72
[[<-,taxa-method
(\[\[<-,collunits-method), 72
$,chronologies-method
($,chronology-method), 73
$,chronology-method, 73
$,collunit-method
($,chronology-method), 73
$,collunits-method
($,chronology-method), 73
$,contact-method ($,chronology-method), 73
$,contacts-method
($,chronology-method), 73
$,dataset-method ($,chronology-method), 73
$,datasets-method
($,chronology-method), 73
$,publication-method
($,chronology-method), 73
$,sample-method ($,chronology-method), 73
$,samples-method ($,chronology-method), 73
$,site-method ($,chronology-method), 73
$,sites-method ($,chronology-method), 73
$,speleothem-method
($,chronology-method), 73
$,speleothems-method
($,chronology-method), 73
$,taxa-method ($,chronology-method), 73
$,taxon-method ($,chronology-method), 73
$<-,chronology-method, 74
$<-,collunit-method
($<-,chronology-method), 74
$<-,dataset-method
($<-,chronology-method), 74
$<-,site-method
($<-,chronology-method), 74
$<-,speleothem-method
($<-,chronology-method), 74
$<-,taxon-method
($<-,chronology-method), 74

add_chronology, 4
add_chronology,collunit,chronology,data.frame-method
(add_chronology), 4
add_chronology,collunit-method
(add_chronology), 4
as.data.frame
(as.data.frame,chronology-method), 5
as.data.frame,authors-method
(as.data.frame,chronology-method), 5
as.data.frame,chronologies-method
(as.data.frame,chronology-method), 5
as.data.frame,chronology-method, 5
as.data.frame,collunit-method
(as.data.frame,chronology-method), 5
as.data.frame,collunits-method
(as.data.frame,chronology-method), 5
as.data.frame,contact-method
(as.data.frame,chronology-method), 5
as.data.frame,contacts-method
(as.data.frame,chronology-method), 5
as.data.frame,dataset-method
(as.data.frame,chronology-method), 5
as.data.frame,datasets-method
(as.data.frame,chronology-method), 5
as.data.frame,publication-method
(as.data.frame,chronology-method), 5
as.data.frame,publications-method
(as.data.frame,chronology-method), 5
as.data.frame,site-method
(as.data.frame,chronology-method), 5
as.data.frame,sites-method
(as.data.frame,chronology-method), 5
as.data.frame,speleothem-method

(as.data.frame,chronology-method), 5
as.data.frame,speleothems-method (as.data.frame,chronology-method), 5
as.data.frame,taxa-method (as.data.frame,chronology-method), 5
as.data.frame,taxon-method (as.data.frame,chronology-method), 5
author-class (author_classes), 6
author_classes, 6
authors-class (author_classes), 6

c (c,chronologies-method), 7
c,chronologies-method, 7
c,collunits-method (c,chronologies-method), 7
c,contact-method (c,chronologies-method), 7
c,contacts-method (c,chronologies-method), 7
c,datasets-method (c,chronologies-method), 7
c,missingOrNULL-method (c,chronologies-method), 7
c,publications-method (c,chronologies-method), 7
c,samples-method (c,chronologies-method), 7
c,sites-method (c,chronologies-method), 7
c,speleothems-method (c,chronologies-method), 7
c,taxa-method (c,chronologies-method), 7
cash ($,chronology-method), 73
cash,collunit-method ($,chronology-method), 73
cash,collunits-method ($,chronology-method), 73
cash,contact-method ($,chronology-method), 73
cash,contacts-method ($,chronology-method), 73
cash,dataset-method ($,chronology-method), 73
cash,datasets-method ($,chronology-method), 73
cash,publication-method ($,chronology-method), 73
cash,sample-method ($,chronology-method), 73
cash,samples-method ($,chronology-method), 73
cash,speleothem-method ($,chronology-method), 73
cash,speleothems-method ($,chronology-method), 73
cash,taxa-method ($,chronology-method), 73
cash,taxon-method ($,chronology-method), 73
cash-set ($<-,chronology-method), 74
cash-set,collunit-method ($<-,chronology-method), 74
cash-set,dataset-method ($<-,chronology-method), 74
cash-set,speleothem-method ($<-,chronology-method), 74
cash-set,taxon-method ($<-,chronology-method), 74
check_contacts, 8
chroncontrols, 8
chroncontrols,chronologies-method, 9
chroncontrols,chronology-method, 9
chroncontrols,site-method, 10
chroncontrols,sites-method, 10
chronologies, 11
chronologies,collunit-method, 11
chronologies,collunits-method (chronologies), 11
chronologies,site-method (chronologies), 11
chronologies,sites-method (chronologies), 11
chronologies-class (chronologies_classes), 12
chronologies_classes, 12
chronology-class (chronologies_classes), 12
cite_data, 12
cite_data,NULL-method (cite_data), 12
cite_data,site-method (cite_data), 12
cite_data,sites-method, 13
clean, 13
collunit-class (collunits_classes), 16

collunits, 15
collunits,site-method *(collunits)*, 15
collunits,sites-method, 15
collunits-class *(collunits_classes)*, 16
collunits_classes, 16
contact-class *(contacts_classes)*, 16
contacts-class *(contacts_classes)*, 16
contacts_classes, 16
coordinates, 16
coordinates,sites-method, 17
count, 17
count,site-method *(count)*, 17
count,sites-method *(count)*, 17

dataset-class, 18
datasets, 18
datasets,collunit-method *(datasets)*, 18
datasets,collunits-method, 19
datasets,site-method *(datasets)*, 18
datasets,sites-method *(datasets)*, 18
datasets-class *(dataset-class)*, 18
doi, 19
doi,publication-method *(doi)*, 19
doi,site-method *(doi)*, 19
doi,sites-method, 20

filter, 20

get_contacts, 23
get_datasets, 24
get_datasets(), *37, 38*
get_documentation, 27
get_downloads, 27
get_manual, 30
get_publications, 30
get_sites, 32
get_speleothems, 34
get_stats, 35
get_table, 36
get_taxa, 37
get_taxon, 38
get_taxon(), *37, 38*
getids, 22

length *(length,chronologies-method)*, 39
length,chronologies-method, 39
length,collunits-method *(length,chronologies-method)*, 39
length,contacts-method *(length,chronologies-method)*, 39
length,datasets-method *(length,chronologies-method)*, 39
length,publications-method *(length,chronologies-method)*, 39
length,samples-method *(length,chronologies-method)*, 39
length,sites-method *(length,chronologies-method)*, 39
length,speleothems-method *(length,chronologies-method)*, 39
length,taxa-method *(length,chronologies-method)*, 39

missingOrNULL-class, 40

names *(names,collunit-method)*, 40
names,collunit-method, 40
names,contact-method *(names,collunit-method)*, 40
names,dataset-method *(names,collunit-method)*, 40
names,publication-method *(names,collunit-method)*, 40
names,publications-method *(names,collunit-method)*, 40
names,site-method *(names,collunit-method)*, 40
names,sites-method *(names,collunit-method)*, 40
names,speleothem-method *(names,collunit-method)*, 40
names,taxon-method *(names,collunit-method)*, 40

pingNeotoma, 41
plot, 42
plot,site,ANY-method *(plot)*, 42
plot,site-method *(plot)*, 42
plot,sites,ANY-method *(plot)*, 42
plot,sites-method *(plot)*, 42

plotLeaflet, 42
plotLeaflet,ANY-method (plotLeaflet), 42
plotLeaflet,site-method (plotLeaflet), 42
plotLeaflet,sites-method, 43
publication-class (publications_classes), 44
publications-class (publications_classes), 44
publications_classes, 44

repositories-class (repositories_classes), 44
repositories_classes, 44
repository-class (repositories_classes), 44

sample-class (samples_classes), 46
samples, 45
samples,collunit-method (samples), 45
samples,collunits-method (samples), 45
samples,site-method (samples), 45
samples,sites-method, 45
samples-class (samples_classes), 46
samples_classes, 46
set_chronology, 47
set_collunit, 48
set_contact, 50
set_dataset, 51
set_default, 53
set_default,chronologies-method, 53
set_publication, 54
set_sample, 56
set_server, 57
set_site, 58
set_speleothem, 59
show (show,chronology-method), 61
show,chronologies-method (show,chronology-method), 61
show,chronology-method, 61
show,collunit-method (show,chronology-method), 61
show,collunits-method (show,chronology-method), 61
show,contact-method (show,chronology-method), 61
show,contacts-method (show,chronology-method), 61
show,dataset-method (show,chronology-method), 61
show,datasets-method (show,chronology-method), 61
show,publication-method (show,chronology-method), 61
show,publications-method (show,chronology-method), 61
show,site-method (show,chronology-method), 61
show,sites-method (show,chronology-method), 61
show,speleothem-method (show,chronology-method), 61
show,speleothems-method (show,chronology-method), 61
show,taxa-method (show,chronology-method), 61
show,taxon-method (show,chronology-method), 61
site-class (sites_classes), 62
sites-class (sites_classes), 62
sites_classes, 62
specimen-class (specimens_classes), 62
specimens-class (specimens_classes), 62
specimens_classes, 62
speleothem-class (speleothems_classes), 65
speleothemdetails, 63
speleothemdetails,collunit-method (speleothemdetails), 63
speleothemdetails,collunits-method (speleothemdetails), 63
speleothemdetails,site-method (speleothemdetails), 63
speleothemdetails,sites-method, 63
speleothems, 64
speleothems,collunit-method (speleothems), 64
speleothems,collunits-method (speleothems), 64
speleothems,site-method (speleothems), 64
speleothems,sites-method, 65
speleothems-class (speleothems_classes), 65
speleothems_classes, 65
sub,collunits-method

(\[,collunits,numeric-method), 69
sub,datasets-method (\[,collunits,numeric-method), 69
sub,speleothems-method (\[,collunits,numeric-method), 69
sub,taxa-method (\[,collunits,numeric-method), 69
sub-sub (\[\[,chronologies,numeric-method), 71
sub-sub,collunits-method (\[\[,chronologies,numeric-method), 71
sub-sub,contacts-method (\[\[,chronologies,numeric-method), 71
sub-sub,datasets-method (\[\[,chronologies,numeric-method), 71
sub-sub,publications-method (\[\[,chronologies,numeric-method), 71
sub-sub,samples-method (\[\[,chronologies,numeric-method), 71
sub-sub,speleothems-method (\[\[,chronologies,numeric-method), 71
sub-sub,taxa-method (\[\[,chronologies,numeric-method), 71
sub-subset (\[\[<-,collunits-method), 72
sub-subset,collunits-method (\[\[<-,collunits-method), 72
sub-subset,datasets-method (\[\[<-,collunits-method), 72
sub-subset,publications-method (\[\[<-,collunits-method), 72
sub-subset,samples-method (\[\[<-,collunits-method), 72
sub-subset,speleothems-method (\[\[<-,collunits-method), 72
sub-subset,taxa-method (\[\[<-,collunits-method), 72
subset (\[<-,collunit,character-method), 70
subset,collunit-method (\[<-,collunit,character-method), 70
subset,dataset-method (\[<-,collunit,character-method), 70
subset,sample-method (\[<-,collunit,character-method), 70
subset,speleothem-method (\[<-,collunit,character-method), 70
subset,taxon-method (\[<-,collunit,character-method), 70
summary,sites-method, 66

taxa, 66
taxa,collunit-method (taxa), 66
taxa,collunits-method (taxa), 66
taxa,site-method (taxa), 66
taxa,sites-method, 67
taxa-class (taxa_classes), 67
taxa_classes, 67
taxon-class (taxa_classes), 67
toWide, 68