

Package: c2z (via r-universe)

November 6, 2024

Type Package

Title A Reference Manager

Version 0.2.0.9019

Maintainer Øystein Olav Skaar <oeysan@gmail.com>

Description 'Cristin' to 'Zotero' ('c2z') aims at obtaining total dominion over 'Cristin' ('Current Research Information SysTem in Norway') and 'Zotero'. The package enables manipulating 'Zotero' libraries using 'R'. Import references from 'Cristin', 'Regjeringen', 'CRAN', 'ISBN' ('Alma', 'LoC'), and 'DOI' ('CrossRef', 'DataCite') to a 'Zotero' library. Add, edit, copy, or delete items, including attachments and collections, and export references to 'BibLaTeX' (and other formats).

License MIT + file LICENSE

URL <https://github.com/oeysan/c2z/>, <https://oeysan.github.io/c2z/>

BugReports <https://github.com/oeysan/c2z/issues/>

Depends R (>= 4.1)

Imports dplyr, httr, jsonlite, purrr, rlang, rvest, stats, stringr, tibble, tidyr, tidyselect, utils, xml2

Suggests covr, knitr, rmarkdown, htmltools, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Repository <https://oeysan.r-universe.dev>

RemoteUrl <https://github.com/oeysan/c2z>

RemoteRef HEAD

RemoteSha eca24c160a7b0ac59f54b3ef3be835dec60a080b

Contents

Cristin	2
CristinSupported	6
CristinWrangler	7
DoiCrossref	8
DoiDatacite	9
Zotero	10
ZoteroAdd	14
ZoteroCheck	15
ZoteroCopy	17
ZoteroCran	19
ZoteroDelete	20
ZoteroDoi	21
ZoteroExport	22
ZoteroFormat	24
ZoteroGet	25
ZoteroGov	27
ZoteroIndex	28
ZoteroIsbn	29
ZoteroKey	30
ZoteroLibrary	30
ZoteroMatch	32
ZoteroPost	34
ZoteroTypes	36
Index	37

Cristin

Search Cristin API

Description

Use search parameters for the Cristin API and return the results as a tibble. See [CristinWrangler](#)

Usage

```
Cristin(
  id = NULL,
  doi = NULL,
  title = NULL,
  contributor = NULL,
  issn = NULL,
  unit = NULL,
  institution = NULL,
  user = NULL,
  category = NULL,
  published_since = NULL,
```

```

published_before = NULL,
created_since = NULL,
created_before = NULL,
modified_since = NULL,
modified_before = NULL,
year_reported = NULL,
project_code = NULL,
funding_source = NULL,
funding = NULL,
lang = NULL,
page = 1,
per_page = 1000,
max.results = NULL,
no.results = FALSE,
sort = NULL,
filter = NULL,
fields = "all",
open.query = NULL,
all.results = TRUE,
force = FALSE,
remove.duplicates = TRUE,
silent = FALSE,
base.url = "https://api.cristin.no/v2/results",
custom.url = NULL,
use.identifiers = TRUE,
crossref.search = FALSE,
autosearch = TRUE,
zotero = NULL,
zotero.import = TRUE,
zotero.check = TRUE,
nvi = FALSE,
remove.na = TRUE,
replace.na = "book",
force.type = NULL,
override = FALSE,
polite = TRUE,
log = list()
)

```

Arguments

id	The Cristin id of the result, Default: NULL
doi	DOI name (e.g. doi=10.1000/123456), Default: NULL
title	The title of the result, Default: NULL
contributor	Author's name or Cristin person id, Default: NULL
issn	The issn of the result, Default: NULL
unit	Id (e.g. unit=185.53.18.10), Default: NULL

<code>institution</code>	Id (one number, e.g. <code>institution=185</code>), name or acronym of the institution the contributors belong to, Default: NULL
<code>user</code>	A person's username in Cristin together with the institution id separated by colon (e.g., <code>'askeladd:185'</code>), Default: NULL
<code>category</code>	Categories. See category codes , Default: NULL
<code>published_since</code>	Results published since and inclusive the given year, (yyyy), Default: NULL
<code>published_before</code>	Results published before and inclusive the given year, (yyyy), Default: NULL
<code>created_since</code>	Results created since and inclusive the given date, (yyyy-mm-dd), Default: NULL
<code>created_before</code>	Results created before and inclusive the given date, (yyyy-mm-dd), Default: NULL
<code>modified_since</code>	Results modified since and inclusive the given date, (yyyy-mm-dd), Default: NULL
<code>modified_before</code>	Results modified before and inclusive the given date, (yyyy-mm-dd), Default: NULL
<code>year_reported</code>	The year a result was reported, Default: NULL
<code>project_code</code>	Project code is the internal reference number used by funding source, Default: NULL
<code>funding_source</code>	Funding source code e.g: NFR, Default: NULL
<code>funding</code>	Funding source code e.g: NFR, and <code>project_code</code> together separated by colon (e.g., <code>NFR:1234</code>), Default: NULL
<code>lang</code>	Two letter language code , Default: NULL
<code>page</code>	Page number. See pagination , Default: 1
<code>per_page</code>	Number of items per page (1000 is max). See pagination , Default: 1000
<code>max.results</code>	Do you need a limit?, Default: NULL
<code>no.results</code>	Do you need only the number of results?, Default: FALSE
<code>sort</code>	Sorts on 'category' and/or 'year_published'. See search and sort . Default sort order is on 'cristin_result_id' in ascending order, Default: NULL
<code>filter</code>	Vector of categories to include in results. See category code , Default: NULL
<code>fields</code>	'fields = all' gives a list of result objects with all available fields. If this parameter is omitted, a list of result summaries with fewer fields will be returned, Default: all
<code>open.query</code>	Define your own query terms, Default: NULL
<code>all.results</code>	Find all results in query, Default: TRUE
<code>force</code>	Force is seldom wise, but sometimes..., Default: FALSE
<code>remove.duplicates</code>	Remove duplicates if TRUE, Default: TRUE
<code>silent</code>	c2z is noisy, tell it to be quiet, Default: FALSE
<code>base.url</code>	The base url for the Cristin API, Default: <code>https://api.cristin.no/v2/results</code>

<code>custom.url</code>	Define your own Cristin API url, Default: NULL
<code>use.identifiers</code>	Use if ISBN/DOI identifiers if enabled, Default: TRUE
<code>crossref.search</code>	Query Crossref database based on title, authors, and date if enabled, Default: FALSE
<code>autosearch</code>	Results could be automatically evaluated (based on some logic) or you could inspect them manually if set to FALSE, Default: TRUE
<code>zotero</code>	A list with information on the specified Zotero library (e.g., id, API key, collections, and items), Default: NULL
<code>zotero.import</code>	Use CristinWrangler to wrangle metadata into an acceptable format for Zotero, Default: TRUE
<code>zotero.check</code>	Check for Cristin references already stored in Zotero, Default: TRUE
<code>nvi</code>	Filter out non-NVI publications Default: FALSE
<code>remove.na</code>	Cristin contains many, more or less, obscure categories, and not all are (yet) supported. By default these are removed, however, if this option is set to FALSE unsupported categories are treated according to <code>replace.na</code> , Default: TRUE
<code>replace.na</code>	May the odds be in your favor and replace unsupported categories with a predefined itemType if <code>remove.na</code> is set to false, Default: 'book'
<code>force.type</code>	Force all items to a predefined itemType, Default: NULL
<code>override</code>	Put your faith in the algorithms and the identifiers (i.e., DOI/ISBN) and override what is reported in Cristin, Default: FALSE
<code>polite</code>	Please store you email in '.Renviron' to query Crossref, Default: TRUE
<code>log</code>	A list for storing log elements, Default: list()

Details

Please see #' <https://oeysan.github.io/c2z/>

Value

A list with (exported) items from Cristin

See Also

[http_error](#), [GET](#), [RETRY](#) [tail](#), [head slice](#)

Examples

```
# Simple `Cristin` search by id
example <- Cristin(id = "840998")

# Print index using `ZoteroIndex`
if (any(nrow(example$results))) {
  ZoteroIndex(example$results) |>
  dplyr::select(name) |>
```

```

    print(width = 80)
  }

```

CristinSupported *Check for supported export formats from Cristin to Zotero*

Description

Used to filter Cristin categories that are not supported

Usage

```

CristinSupported(
  data = NULL,
  zotero.check = TRUE,
  items = NULL,
  remove.na = TRUE,
  replace.na = "book",
  force.type = NULL,
  remove.duplicates = TRUE,
  silent = FALSE,
  log = list()
)

```

Arguments

data	Zotero-type matrix, Default: NULL
zotero.check	Should the function look for duplicates?, Default: TRUE
items	Items in library to check for
remove.na	Cristin contains many, more or less, obscure categories, and not all are (yet) supported. By default these are removed, however, if this option is set to FALSE unsupported categories are treated according to replace.na, Default: TRUE
replace.na	May the odds be in your favor and replace unsupported categories with a predefined itemType if remove.na is set to false, Default: 'book'
force.type	Force all items to a predefined itemType, Default: NULL
remove.duplicates	Remove duplicates if TRUE, Default: TRUE
silent	c2z is noisy, tell it to be quiet, Default: FALSE
log	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

Zotero supported items with unsupported categories as NA

See Also

[tibble](#)

Examples

```
# Supported Cristin items
print(CristinSupported(), n = 5)
```

CristinWrangler

Wrangle Cristin metadata into Zotero-type format

Description

The little engine that could, at least try, to wrangle metadata from Cristin into a Zotero-type matrix. References can be augmented, or replaced with metadata gathered through ISBN or DOI identifiers, and through Crossref queries if all other fails. Or, hope the best and place your misguided faith in the ability of researchers to correctly register their own publications.

Usage

```
CristinWrangler(
  data,
  meta = list(),
  use.identifiers = TRUE,
  crossref.search = FALSE,
  autosearch = TRUE,
  override = FALSE,
  silent = FALSE,
  polite = TRUE,
  prefer.semantic = TRUE,
  log = list()
)
```

Arguments

<code>data</code>	Tibble containing metadata from Cristin
<code>meta</code>	A list for storing data elements, Default: <code>list()</code>
<code>use.identifiers</code>	Use if ISBN/DOI identifiers if enabled, Default: <code>TRUE</code>
<code>crossref.search</code>	Query Crossref database based on title, authors, and date if enabled, Default: <code>FALSE</code>

autosearch	Results could be automatically evaluated (based on some logic) or you could inspect them manually if set to FALSE, Default: TRUE
override	Put your faith in the algorithms and the identifiers (i.e., DOI/ISBN) and override what is reported in Cristin, Default: FALSE
silent	Running silent, running deep, Default: FALSE
polite	Please store you email in '.Renviron' to query Crossref, Default: TRUE
prefer.semantic	Prefer metadata from Semantic Scholar, Default: TRUE
log	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

A Zotero-type matrix (tibble)

See Also

[unnest](#), [nest filter](#), [across](#), [reexports](#), [bind](#), [mutate](#), [distinct](#) [RETRY tibble](#)

Examples

```
# Simple `Cristin` search by id with import set as FALSE
example <- Cristin(id = "840998", zotero.import = FALSE)

if (any(nrow(example$results))) {
  # Use `ZoteroIndex` to print `CristinWrangler`
  CristinWrangler(example$results)$results |>
  ZoteroIndex() |>
  dplyr::select(name) |>
  print(width = 80)
}
```

DoiCrossref

Wrangle CrossRef metadata into Zotero-type format

Description

Query CrossRef by DOI and fetch metadata

Usage

```
DoiCrossref(data, meta = list(), silent = TRUE, log = list())
```


Arguments

data	XML data from CrossRef containing metadata
meta	A list collecting all metadata used to create , Default: list()
silent	c2z is noisy, tell it to be quiet, Default: TRUE
log	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

A Zotero-type matrix (tibble)

Examples

```
# `DoiCrossref` is called from `ZoteroDoi` (if item found)
example <- ZoteroDoi("10.1126/sciadv.abd1705")

# Print index using `ZoteroIndex`
if (!is.null(example$data)) {
  ZoteroIndex(example$data) |>
  dplyr::select(name) |>
  print(width = 80)

# Display catalog
example$data$libraryCatalog
}
```

DoiDatacite

Wrangle DataCite metadata into Zotero-type format

Description

Query DataCite by DOI and fetch metadata

Usage

```
DoiDatacite(data, meta = list(), silent = TRUE, log = list())
```

Arguments

data	XML data from DataCite containing metadata
meta	A list collecting all metadata used to create , Default: list()
silent	c2z is noisy, tell it to be quiet, Default: TRUE
log	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

A Zotero-type matrix (tibble)

Examples

```
# `DoiDatacite` is called from `ZoteroDoi` (if item found)
example <- ZoteroDoi("10.17605/OSF.IO/7HZ4P")

# Use `ZoteroIndex` to print
# Print index using `ZoteroIndex`
if (any(nrow(example$data))) {
  ZoteroIndex(example$data) |>
    dplyr::select(name) |>
    print(width = 80)

# Display catalog
example$data$libraryCatalog
}
```

Zotero

Connect to Zotero API

Description

Wrapper to connect with the Zotero API and the main functions of **c2z**

Usage

```
Zotero(
  collection.names = NULL,
  collection.key = NULL,
  collection.path = NULL,
  item.key = NULL,
  library = FALSE,
  case.insensitive = TRUE,
  ancestor = FALSE,
  recursive = FALSE,
  create = FALSE,
  limit = 100,
  start = 0,
  search.collections = TRUE,
  get.collections = TRUE,
  get.items = TRUE,
```

```
item.type = NULL,  
all.results = TRUE,  
max.results = NULL,  
collections = NULL,  
items = NULL,  
attachments = NULL,  
metadata = NULL,  
doi = NULL,  
isbn = NULL,  
export = FALSE,  
csl.type = NULL,  
csl.name = "style",  
format = "biblatex",  
save.data = FALSE,  
save.path = NULL,  
bib.name = "references",  
library.type = NULL,  
linkwrap = 1,  
style = "apa",  
locale = "en-US",  
copy = FALSE,  
copy.collections = TRUE,  
copy.items = TRUE,  
copy.extras = TRUE,  
remove.missing = TRUE,  
change.library = FALSE,  
copy.user = TRUE,  
copy.id = NULL,  
copy.api = NULL,  
post = FALSE,  
post.collections = TRUE,  
post.items = TRUE,  
post.attachments = TRUE,  
post.limit = 50,  
delete = FALSE,  
delete.collections = FALSE,  
delete.items = FALSE,  
delete.limit = 50,  
ragnarok = FALSE,  
user = TRUE,  
index = FALSE,  
id = NULL,  
token = NULL,  
token.api = NULL,  
api = NULL,  
force = FALSE,  
base.url = "https://api.zotero.org",  
silent = FALSE,
```

```

    zotero = NULL,
    log = list()
)

```

Arguments

<code>collection.names</code>	Vector of collection names to create or search for, Default: NULL
<code>collection.key</code>	A specified collection key, Default: NULL
<code>collection.path</code>	Vector of nested collection keys, Default: NULL
<code>item.key</code>	A specified item key, Default: NULL
<code>library</code>	Use 'ZoteroLibrary' to fetch collections and items, Default: FALSE
<code>case.insensitive</code>	Disregard letter casing when searching for collections, Default: TRUE
<code>ancestor</code>	Trace the lineage of a collection (i.e., find the top-level collection), Default: FALSE
<code>recursive</code>	Find all nested collections, Default: FALSE
<code>create</code>	Create missing collections, Default: FALSE
<code>limit</code>	Number of results per query (max 100), Default: 100
<code>start</code>	Starting position of query (0 = first result), Default: 0
<code>search.collections</code>	Search all collections if <code>collection.key</code> fails, Default: TRUE
<code>get.collections</code>	Fetch collections, Default: TRUE
<code>get.items</code>	Fetch items, Default: TRUE
<code>item.type</code>	Items to search for (NULL = everything), Default: NULL
<code>all.results</code>	Find all results in query, Default: TRUE
<code>max.results</code>	Do you need a limit?, Default: NULL
<code>collections</code>	Predefined collections (as tibble), Default: NULL
<code>items</code>	Predefined metadata (as tibble), Default: NULL
<code>attachments</code>	Predefined attachments (as tibble), Default: NULL
<code>metadata</code>	Predefined metadata in Zotero-format, Default: NULL
<code>doi</code>	Use ZoteroDoi to fetch DOI metadata, Default: NULL
<code>isbn</code>	Use ZoteroIsbn to fetch ISBN metadata, Default: NULL
<code>export</code>	Use 'ZoteroExport' to export items, Default: FALSE
<code>csl.type</code>	Specify a CSL type to Official repository for Citation Style Language (CSL), Default: NULL
<code>csl.name</code>	Name of saved CSL file, Default: 'style'
<code>format</code>	Export format of Zotero items, Default: 'biblatex'
<code>save.data</code>	Save data (e.g., bibliography) to disk, Default: FALSE

save.path	Location to store data on disk, Default: NULL
bib.name	Name of exported bibliography, Default: 'references'
library.type	Commma-separated data from Zotero (i.e., data, bib, citation), Default: NULL
linkwrap	Set URL (e.g., DOI) as HTML link (1 = yes), Default: 1
style	Citation style to use for appended bibliography and/or citations, Default: apa
locale	Desired language format of bibliography, Default: 'en-US'
copy	Use 'ZoteroCopy' to delete collections and/or items, Default: FALSE
copy.collections	Try to copy specified collections, Default: TRUE
copy.items	Try to copy specified items?, Default: TRUE
copy.extras	Try to copy specified extras (i.e., attachments and notes)?, Default: TRUE
remove.missing	Deleted missing extras, Default: TRUE
change.library	Stage changing of library (e.g., from a group to a personal library), Default: FALSE
copy.user	New user type (The functions will use 'group' as prefix if FALSE), Default: TRUE
copy.id	New id, Default: NULL
copy.api	New API key. Set API to 'NA' if key is not needed, Default: NULL
post	Use 'ZoteroPost' to post collections and/or items, Default: FALSE
post.collections	Try to copy specified collections, Default: TRUE
post.items	Try to copy specified items?, Default: TRUE
post.attachments	Try to copy specified extras (i.e., attachments and notes)?, Default: TRUE
post.limit	Number of collections/items to post per request (max 50), Default: 50
delete	Use 'ZoteroDelete' to delete collections and/or items, Default: FALSE
delete.collections	Try to delete specified collections, Default: TRUE
delete.items	Try to delete specified items?, Default: TRUE
delete.limit	Number of collections/items to delete per request (max 50), Default: 50
ragnarok	Delete EVERYTHING in the specified library, Default: FALSE
user	User type (The functions will use 'group' as prefix if FALSE), Default: TRUE
index	Create an index of items, Default: FALSE
id	User or group ID, Default: NULL
token	Name of user or group token as defined in '.Renviron', Default: NULL
token.api	Name of API token as defined in '.Renviron', Default: NULL
api	API key to connect with the Zotero library. Set API to 'NA' if key is not needed. See Zotero API , Default: NULL
force	Force is seldom wise, but sometimes..., Default: FALSE

<code>base.url</code>	Base url of the Zotero API, Default: 'https://api.zotero.org'
<code>silent</code>	c2z is noisy, tell it to be quiet, Default: FALSE
<code>zotero</code>	A list with information on the specified Zotero library (e.g., id, API key, collections, and items), Default: NULL
<code>log</code>	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

A list with information on the specified Zotero library (e.g., id, API key, collections, and items)

See Also

[http_error](#), [GET](#)

Examples

```
# Create the default Zotero list
example <- Zotero(id = "9913421", api = "Rq1AmlH511KPghfCseAq1sQ1")
# Print the interesting pars of an otherwise empty list
print(tail(example,5))
```

ZoteroAdd

Add items to Zotero list

Description

Use identifiers or predefined data to add to the Zotero list

Usage

```
ZoteroAdd(zotero, metadata = NULL, doi = NULL, isbn = NULL, silent = FALSE)
```

Arguments

<code>zotero</code>	A list with information on the specified Zotero library (e.g., id, API key, collections, and items)
<code>metadata</code>	Predefined metadata in Zotero-format, Default: NULL
<code>doi</code>	Use ZoteroDoi to fetch DOI metadata, Default: NULL
<code>isbn</code>	Use ZoteroIsbn to fetch ISBN metadata, Default: NULL
<code>silent</code>	c2z is noisy, tell it to be quiet, Default: FALSE

Details

Please see <https://oeysan.github.io/c2z/>

Value

Will add data to the Zotero list

Examples

```
# Add items from `ZoteroIsbn` to the default Zotero list
example <- ZoteroAdd(
  Zotero(
    id = "9913421",
    api = "Rq1AmlH511KPghfCseAq1sQ1"
  ),
  isbn = "978-1529797138"
)
# Print index using `ZoteroIndex`
if (any(nrow(example$items))) {
  ZoteroIndex(example$items) |>
  dplyr::select(name) |>
  print(width = 80)
}
```

ZoteroCheck

Check Zotero library for duplicates

Description

Remove references that are not modified since last added to Zotero

Usage

```
ZoteroCheck(
  data,
  id,
  id.type,
  created,
  last.modified,
  items,
  remove.duplicates = TRUE,
  silent = FALSE,
  log = list()
)
```

Arguments

<code>data</code>	Tibble containing metadata (from Cristin)
<code>id</code>	column containing identifier (e.g., <code>cristin_result_id</code>)
<code>id.type</code>	Type of metadata as found in the Zotero extra field ("e.g., Cristin)
<code>created</code>	column containing creation date in UNIX timestamp format (e.g., <code>created</code>)
<code>last.modified</code>	column containing modification date in UNIX timestamp format (e.g., <code>last_modified</code>)
<code>items</code>	Items in library to check for
<code>remove.duplicates</code>	Remove duplicates if TRUE, Default: TRUE
<code>silent</code>	<code>c2z</code> is noisy, tell it to be quiet, Default: FALSE
<code>log</code>	A list for storing log elements, Default: <code>list()</code>

Details

Please see <https://oeysan.github.io/c2z/>

Value

Returns non-duplicated data in a Zotero-type matrix (tibble)

See Also

[arrange](#), [coalesce](#), [bind_rows](#), [filter](#)

Examples

```
# Simple `Cristin` search by id
cristin.data <- Cristin(
  id = "840998",
  zotero.import = FALSE
)

# Simple `ZoteroCheck`
example <- ZoteroCheck(
  data = cristin.data$result,
  id = "cristin_result_id",
  id.type = "Cristin",
  created = "created",
  last.modified = "last_modified",
  items = Zotero(
    user = FALSE,
    id = "4827927",
    api = "Rq1AmlH511KPghfCseAq1sQ1",
    library = TRUE,
    silent = TRUE
  )$items
)
```


Description

Replace key identifiers with new ones while keeping existing structure and relationship between collections and items

Usage

```
ZoteroCopy(  
  zotero,  
  copy.collections = TRUE,  
  copy.items = TRUE,  
  copy.extras = TRUE,  
  remove.missing = TRUE,  
  change.library = FALSE,  
  copy.user = TRUE,  
  copy.id = NULL,  
  copy.api = NULL,  
  silent = FALSE  
)
```

Arguments

zotero	A list with information on the specified Zotero library (e.g., id, API key, collections, and items)
copy.collections	Try to copy specified collections, Default: TRUE
copy.items	Try to copy specified items?, Default: TRUE
copy.extras	Try to copy specified extras (i.e., attachments and notes)?, Default: TRUE
remove.missing	Deleted missing extras, Default: TRUE
change.library	Stage changing of library (e.g., from a group to a personal library), Default: FALSE
copy.user	New user type (The functions will use 'group' as prefix if FALSE), Default: TRUE
copy.id	New id, Default: NULL
copy.api	New API key, Default: NULL
silent	c2z is noisy, tell it to be quiet, Default: FALSE

Details

Please see <https://oeysan.github.io/c2z/>

Value

A list with information on the specified Zotero library (e.g., copied collections and items)

See Also

[GET mutate](#), [case_when](#), [filter](#), [select](#), [across](#), [arrange](#) [pmap all_of](#)

Examples

```
# Fetching collections and items from default group
zotero = Zotero(
  user = FALSE,
  id = "4827927",
  api = "Rq1AmlH511KPghfCseAq1sQ1",
  library = TRUE
)

# Display collections
if (any(nrow(zotero$collections))) {
  zotero$collections |>
    dplyr::select(key, version, parentCollection) |>
    print(width = 80)
}

# Display items
if (any(nrow(zotero$items))) {
  zotero$items |>
    dplyr::select(key, version) |>
    print(width = 80)
}

# Copy items
example <- ZoteroCopy(
  zotero,
)

# Display collections
if (any(nrow(example$collections))) {
  example$collections |>
    dplyr::select(key, version, parentCollection) |>
    print(width = 80)
}

# Display items
if (any(nrow(example$items))) {
  example$items |>
    dplyr::select(key, version) |>
    print(width = 80)
}
```

ZoteroCran

Wrangle CRAN metadata into Zotero-type format

Description

Query CRAN by name and fetch metadata

Usage

```
ZoteroCran(id, meta = list(), silent = TRUE, log = list())
```

Arguments

id	name of R package
meta	A list collecting all metadata used to create , Default: list()
silent	c2z is noisy, tell it to be quiet, Default: TRUE
log	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

A Zotero-type matrix (tibble)

See Also

[RETRY reexports](#)

Examples

```
# Search cran for package `dplyr`
example <- ZoteroCran(c("dplyr", "jsonlite", "httr"))

# Print index using `ZoteroIndex`
if (any(nrow(example$data))) {
  ZoteroIndex(example$data) |>
  dplyr::select(name) |>
  print(width = 80)
}
```

ZoteroDelete

Delete collections and items from a Zotero library

Description

Cleaning the Zotero library

Usage

```
ZoteroDelete(  
  zotero,  
  delete.collections = FALSE,  
  delete.items = FALSE,  
  delete.limit = 50,  
  force = FALSE,  
  ragnarok = FALSE,  
  silent = FALSE  
)
```

Arguments

zotero	A list with information on the specified Zotero library (e.g., id, API key, collections, and items)
delete.collections	Try to delete specified collections, Default: TRUE
delete.items	Try to delete specified items?, Default: TRUE
delete.limit	Number of collections/items to delete per request (max 50), Default: 50
force	Force is seldom wise, but sometimes..., Default: FALSE
ragnarok	Delete EVERYTHING in the specified library, Default: FALSE
silent	c2z is noisy, tell it to be quiet, Default: FALSE

Details

Please see <https://oeysan.github.io/c2z/>

Value

A list with information on the specified Zotero library (e.g., deleted collections and items)

See Also

[add_headers](#), [RETRY](#)

Examples

```
# Delete everything in a group
example <- ZoteroDelete(
  Zotero(
    user = FALSE,
    id = "4988497",
    api = "Rq1AmlH511KPghfCseAq1sQ1"
  ),
  ragnarok = TRUE,
  force = TRUE
)
```

ZoteroDoi

Use DOI to acquire metadata

Description

Connects with doi.org to create metadata

Usage

```
ZoteroDoi(
  doi,
  meta = list(),
  prefer.semantic = FALSE,
  check.retraction = TRUE,
  silent = TRUE,
  log = list()
)
```

Arguments

doi	A digital object identifier
meta	A list collecting all metadata used to create , Default: list()
prefer.semantic	Prefer metadata from Semantic Scholar, Default: FALSE
check.retraction	Check if marked as retracted, Default: TRUE
silent	c2z is noisy, tell it to be quiet, Default: TRUE
log	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

A Zotero-type matrix (tibble)

See Also

[http_error](#), [GET](#), [RETRY](#) [rename](#), [html_attr](#), [html_text](#), [reexports](#), [html_children](#), [html_name](#)
[bind](#), [arrange](#)

Examples

```
# Simple use of `ZoteroDoi`
example <- ZoteroDoi("10.1126/sciadv.abd1705")

# Print index using `ZoteroIndex`
if (any(nrow(example$data))) {
  ZoteroIndex(example$data) |>
    dplyr::select(name) |>
    print(width = 80)
}
```

ZoteroExport

Export Zotero items to bibliography

Description

Export Zotero items to a specified format (e.g., BibLaTeX) using a CSL format (e.g., APA7)

Usage

```
ZoteroExport(
  zotero,
  csl.type = NULL,
  csl.name = "style",
  locale = "en-US",
  format = "biblatex",
  bib.name = "references",
  save.data = FALSE,
  save.path = NULL,
  silent = FALSE
)
```

Arguments

<code>zotero</code>	A list with information on the specified Zotero library (e.g., id, API key, collections, and items)
<code>csl.type</code>	Specify a CSL type to Official repository for Citation Style Language (CSL), Default: NULL

cs1.name	Name of saved CSL file, Default: 'style'
locale	Desired language format of bibliography, Default: 'en-US'
format	Export format of Zotero items, Default: 'biblatex'
bib.name	Name of exported bibliography, Default: 'references'
save.data	Save data (e.g., bibliography) to disk, Default: FALSE
save.path	Location to store data on disk, Default: NULL
silent	c2z is noisy, tell it to be quiet, Default: FALSE

Details

Please see <https://oeysan.github.io/c2z/>

Value

A list with information on the specified Zotero library (e.g., exported items, bibliography and citations)

See Also

[select where](#)

Examples

```
# Define Zotero list according to default group
zotero = Zotero(
  user = FALSE,
  id = "4827927",
  api = "Rq1Am1H511KPghfCseAq1sQ1",
  item.type = "-attachment || note",
  max.result = 1,
  library = TRUE
)

# Export 1 items from the default group
example <- ZoteroExport(
  zotero
)

# Display exported
cat(example$export, fill = 80)
```

ZoteroFormat

Make a Zotero friendly format of input data

Description

Make a Zotero friendly format of input data, will add a zotero key.

Usage

```
ZoteroFormat(  
  data = NULL,  
  format = NULL,  
  prefix = NULL,  
  check.structure = FALSE,  
  silent = FALSE  
)
```

Arguments

data	A list of metadata or something else, Default: NULL
format	The format of the input data (e.g., 'JSON', 'versions', 'keys'). , Default: NULL
prefix	Add a prefix to the metadata (e.g., user/userID), Default: NULL
check.structure	Check that the structure of a data frame is correct according to the Zotero type, Default: FALSE
silent	c2z is noisy, tell it to be quiet, Default: TRUE

Value

A zotero friendly tibble if requested otherwise format the data according to format.

See Also

[tibble](#), [as_tibble](#) [toJSON](#), [fromJSON](#) [mutate_all](#), [na_if](#), [mutate](#), [across](#), [reexports](#), [select](#) [map](#)

Examples

```
ZoteroFormat(  
  list(title = "This is a test", itemType = "document")  
)
```

ZoteroGet

Get collections and items from a Zotero library

Description

Get

Usage

```
ZoteroGet(  
  zotero,  
  use.collection = TRUE,  
  use.item = FALSE,  
  append.collections = FALSE,  
  append.items = FALSE,  
  append.top = FALSE,  
  append.file = FALSE,  
  custom.url = NULL,  
  open.query = NULL,  
  limit = 100,  
  start = 0,  
  format = "json",  
  item.keys = NULL,  
  collection.keys = NULL,  
  item.type = NULL,  
  library.type = NULL,  
  linkwrap = 1,  
  style = "apa",  
  locale = "en-US",  
  all.results = TRUE,  
  max.results = NULL,  
  result.type = NULL,  
  force = FALSE,  
  silent = FALSE  
)
```

Arguments

zotero	A list with information on the specified Zotero library (e.g., id, API key, collections, and items)
use.collection	Use collection key if present in Zotero list, Default: TRUE
use.item	Use item key if present in Zotero list, Default: FALSE
append.collections	Append '/collections/' to Zotero API url, Default: FALSE
append.items	Append '/items/' to Zotero API url, Default: FALSE
append.top	Append '/top/' to Zotero API url, Default: FALSE

<code>append.file</code>	Append <code>'file/'</code> to Zotero API url, Default: FALSE
<code>custom.url</code>	Use a custom Zotero API url, Default: NULL
<code>open.query</code>	Use your own query, Default: NULL
<code>limit</code>	Number of results per query (max 100), Default: 100
<code>start</code>	Starting position of query (0 = first result), Default: 0
<code>format</code>	Format of response from the Zotero API, Default: <code>'json'</code>
<code>item.keys</code>	Specified vector of items keys, Default: NULL
<code>collection.keys</code>	Specified vector of collection keys, Default: NULL
<code>item.type</code>	Items to search for (NULL = everything), Default: NULL
<code>library.type</code>	Commma-separated data from Zotero (i.e., data, bib, citation), Default: NULL
<code>linkwrap</code>	Set URL (e.g., DOI) as HTML link (1 = yes), Default: 1
<code>style</code>	Citation style to use for appended bibliography and/or citations, Default: <code>apa</code>
<code>locale</code>	Desired language format of bibliography, Default: <code>'en-US'</code>
<code>all.results</code>	Find all results in query, Default: TRUE
<code>max.results</code>	Do you need a limit?, Default: NULL
<code>result.type</code>	Pointless linguistics to display result type (default = <code>'result'</code>), Default: NULL
<code>force</code>	Force is seldom wise, but sometimes..., Default: FALSE
<code>silent</code>	<code>c2z</code> is noisy, tell it to be quiet, Default: FALSE

Details

Please see <https://oeysan.github.io/c2z/>

Value

A list with information on the specified Zotero library (e.g., collections and items)

See Also

[RETRY](#)

Examples

```
# Define Zotero list according to default settings
zotero = Zotero(
  user = FALSE,
  id = "5250382",
  api = "Rq1Am1H511KPghfCseAq1sQ1"
)

# Query default group Zotero library for 1 item
example <- ZoteroGet(
  zotero,
  max.results = 1
)
```

```
)  
  
# Print index using `ZoteroIndex`  
if (any(nrow(example$results))) {  
  ZoteroIndex(example$results) |>  
    dplyr::select(name) |>  
    print(width = 80)  
}
```

ZoteroGov

Wrangle regjeringen.no metadata into Zotero-type format

Description

Query regjeringen by search word and type and fetch metadata

Usage

```
ZoteroGov(search, type = "NOU", meta = list(), silent = TRUE, log = list())
```

Arguments

search	Search term (e.g., 2018: 2)
type	type of query (e.g., white paper, official norwegian reports), Default: "NOU"
meta	A list collecting all metadata used to create , Default: list()
silent	c2z is noisy, tell it to be quiet, Default: TRUE
log	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

A Zotero-type matrix (tibble)

See Also

[RETRY](#) [reexports](#) [bind](#), [arrange](#)

Examples

```
# Search the default entity, Norwegian official reports in regjeringen.no
example <- ZoteroGov("2001:4")
# Print index using `ZoteroIndex`
if (any(nrow(example$data))) {
  ZoteroIndex(example$data) |>
    dplyr::select(name) |>
    print(width = 80)
}
```

ZoteroIndex

Create an index of Zotero items

Description

The function creates a index containing key information about the present Zotero items

Usage

```
ZoteroIndex(data)
```

Arguments

`data` Tibble containing Zotero-type metadata (e.g., from Cristin)

Details

Please see <https://oeysan.github.io/c2z/>

Value

A tibble

See Also

[filter](#), [mutate](#), [across](#), [na_if](#), [case_when](#), [arrange](#), [group_by](#), [context](#), [select everything](#)
[pmap](#)

Examples

```
# Access the default group library
example = Zotero(
  user = FALSE,
  id = "4827927",
  api = "Rq1Am1H511KPghfCseAq1sQ1",
  library = TRUE
)
# Print index using `ZoteroIndex`
```

```

if (any(nrow(example$items))) {
  ZoteroIndex(example$items) |>
    dplyr::select(name) |>
    print(width = 80)
}

```

ZoteroIsbn

Search libraries using ISBN or MMS ID

Description

Query libraries using ISBN (or MMS ID) and fetch metadata

Usage

```
ZoteroIsbn(keys, meta = list(), silent = TRUE, log = list())
```

Arguments

keys	Keys to search with (e.g., ISBN or MMS ID)
meta	A list collecting all metadata used to create , Default: list()
silent	c2z is noisy, tell it to be quiet, Default: TRUE
log	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

A Zotero-type matrix (tibble)

See Also

[rename](#), [html_children](#), [html_attr](#), [html_text](#), [reexports bind](#), [distinct](#), [arrange tibble](#)
[RETRY setNames toJSON](#), [fromJSON](#)

Examples

```

# Search libraries for ISBN metadata
example <- ZoteroIsbn("978-1529797138")

# Print index using `ZoteroIndex`
if (any(nrow(example$data))) {
  ZoteroIndex(example$data) |>
    dplyr::select(name) |>
    print(width = 80)
}

```

ZoteroKey *Create Zotero key or write token*

Description

Create an identifier for Zotero items/collections or a write token to library

Usage

```
ZoteroKey(token = FALSE)
```

Arguments

token Create a write token, Default: FALSE

Details

Please see <https://oeysan.github.io/c2z/>

Value

Will return a 8 character Zotero key or 32 character write token

Examples

```
# Create a Zotero key
key <- ZoteroKey()
# Create a write token
token <- ZoteroKey(TRUE)
```

ZoteroLibrary *Access the Zotero library*

Description

The function uses information stored in the 'zotero' list to access specified collections and items in the Zotero library

Usage

```
ZoteroLibrary(
  zotero,
  case.insensitive = TRUE,
  ancestor = FALSE,
  recursive = FALSE,
  create = FALSE,
  limit = 100,
  start = 0,
  search.collections = TRUE,
  get.collections = TRUE,
  get.items = TRUE,
  item.type = NULL,
  all.results = TRUE,
  max.results = NULL,
  library.type = NULL,
  linkwrap = 1,
  style = "apa",
  locale = "en-US",
  force = FALSE,
  silent = FALSE
)
```

Arguments

<code>zotero</code>	A list with information on the specified Zotero library (e.g., id, API key, collections, and items)
<code>case.insensitive</code>	Disregard letter casing when searching for collections, Default: TRUE
<code>ancestor</code>	Trace the lineage of a collection (i.e., find the top-level collection), Default: FALSE
<code>recursive</code>	Find all nested collections, Default: FALSE
<code>create</code>	Create missing collections, Default: FALSE
<code>limit</code>	Number of results per query (max 100), Default: 100
<code>start</code>	Starting position of query (0 = first result), Default: 0
<code>search.collections</code>	Search all collections if collection.key fails, Default: TRUE
<code>get.collections</code>	Fetch collections, Default: TRUE
<code>get.items</code>	Fetch items, Default: TRUE
<code>item.type</code>	Items to search for (NULL = everything), Default: NULL
<code>all.results</code>	Find all results in query, Default: TRUE
<code>max.results</code>	Do you need a limit?, Default: NULL
<code>library.type</code>	Commma-separated data from Zotero (i.e., data, bib, citation), Default: NULL

linkwrap	Set URL (e.g., DOI) as HTML link (1 = yes), Default: 1
style	Citation style to use for appended bibliography and/or citations, Default: apa
locale	Desired language format of bibliography, Default: 'en-US'
force	Force is seldom wise, but sometimes..., Default: FALSE
silent	c2z is noisy, tell it to be quiet, Default: FALSE

Details

Please see <https://oeysan.github.io/c2z/>

Value

A list with information on the specified Zotero library (e.g., collections and items)

See Also

[toJSON](#), [fromJSON](#) [distinct](#), [arrange](#) [tibble](#)

Examples

```
# Access the default group library
example <- ZoteroLibrary(
  Zotero(
    user = FALSE,
    id = "4827927",
    api = "Rq1AmlH511KPghfCseAq1sQ1"
  )
)
# Print index using `ZoteroIndex`
if (any(nrow(example$items))) {
  ZoteroIndex(example$items) |>
  dplyr::select(name) |>
  print(width = 80)
}
```

Description

Use the CrossRef API to match data (e.g., Cistin metadata with Crossref metadata) or search by title, authors and date

Usage

```
ZoteroMatch(
  title,
  authors,
  date,
  haystack = NULL,
  haystack.size = 3,
  crossref.search = FALSE,
  autosearch = FALSE,
  cristin.data = NULL,
  external.data = NULL,
  polite = TRUE,
  silent = FALSE,
  prefer.semantic = FALSE,
  log = list()
)
```

Arguments

<code>title</code>	Title of reference
<code>authors</code>	creators of the reference
<code>date</code>	publication date of the reference
<code>haystack</code>	Potential matches for search term (i.e., needle), Default: NULL
<code>haystack.size</code>	Number of items in the haystack, Default: 3
<code>crossref.search</code>	Search CrossRef if needle not found in haystack, Default: FALSE
<code>autosearch</code>	Match automatically or compare needle with haystack, Default: FALSE
<code>cristin.data</code>	Metadata from Cristin, Default: NULL
<code>external.data</code>	Metadata from external source (e.g., CrossRef), Default: NULL
<code>polite</code>	Will use an email stored in <code>\$.Renviron</code> , Default: TRUE
<code>silent</code>	Running silent, running deep, Default: FALSE
<code>prefer.semantic</code>	Prefer metadata from Semantic Scholar, Default: FALSE
<code>log</code>	A list for storing log elements, Default: list()

Details

Please see <https://oeysan.github.io/c2z/>

Value

A Zotero-type matrix (tibble) if match is found otherwise NULL

See Also

[select](#), [reexports](#), [mutate](#), [coalesce](#), [pull](#) `sym` `RETRY` `toJSON`, [fromJSON](#) `adist`, [head](#) `map`

Examples

```
# Conduct an autosearch in CrossRef using title, authors and date
example <- ZoteroMatch(
  title = "Nonreplicable publications",
  authors = "Serra-Garcia & Gneezy",
  date = "2021",
  autosearch = TRUE
)

# Print index using `ZoteroIndex`
if (any(nrow(example$data))) {
  ZoteroIndex(example$data) |>
  dplyr::select(name) |>
  print(width = 80)
}
```

ZoteroPost

Post collections and items to a Zotero library

Description

Create or update collections and items in a specified library

Usage

```
ZoteroPost(
  zotero,
  post.collections = TRUE,
  post.items = TRUE,
  post.attachments = TRUE,
  post.limit = 50,
  force = FALSE,
  silent = FALSE
)
```

Arguments

zotero	A list with information on the specified Zotero library (e.g., id, API key, collections, and items)
post.collections	Try to copy specified collections, Default: TRUE
post.items	Try to copy specified items?, Default: TRUE
post.attachments	Try to copy specified extras (i.e., attachments and notes)?, Default: TRUE
post.limit	Number of collections/items to post per request (max 50), Default: 50
force	Force is seldom wise, but sometimes..., Default: FALSE
silent	c2z is noisy, tell it to be quiet, Default: FALSE

Details

Please see <https://oeysan.github.io/c2z/>

Value

A list with information on the specified Zotero library (e.g., posted collections and items)

See Also

[select](#), [bind](#), [mutate](#) [add_headers](#), [RETRY](#) [toJSON](#), [fromJSON](#) [as_tibble](#), [add_column](#) [stack](#) [setNames](#)

Examples

```
# Connect to the public group "c2z_delete"
# NB! This process can be done using only `Zotero` (see README)
zotero <- Zotero(
  user = FALSE,
  id = "4988497",
  api = "Rq1Am1H511KPghfCseAq1sQ1"
)

# Create a new collection for POST
zotero$collections <- tibble::tibble(
  key = ZoteroKey(),
  version = 0,
  name = "Post-test",
  parentCollection = "FALSE"
)

# Add item to post using `ZoteroAdd` (and `ZoteroDoi`)
zotero <- ZoteroAdd(
  zotero,
  doi = "10.1126/sciadv.abd1705"
)

# Post a DOI to the public group "c2z_delete"
post.example <- ZoteroPost(
  zotero,
  post.collections = TRUE,
  post.items = TRUE
)

# Delete collections and items using `ZoteroDelete`
delete.example <- ZoteroDelete(
  post.example,
  delete.collections = TRUE,
  delete.items = TRUE
)

# Print index using `ZoteroIndex`
if (any(nrow(post.example$items))) {
```

```
ZoteroIndex(post.example$items) |>
  dplyr::select(name) |>
  print(width = 80)
}
```

ZoteroTypes

List with empty zotero-items

Description

Each tibble in the list represents a zotero-item

Usage

```
ZoteroTypes(type = NULL, names = TRUE)
```

Arguments

type	Type of zotero-item, Default: NULL
names	return only column names if set to TRUE, Default: TRUE

Format

A list with 36 tibbles with zero rows and various columns

Details

Used to create Zotero-items from list of metadata

Value

Either list of zotero-items or specified item

Examples

```
# All zotero-items
names(ZoteroTypes())
# Column names of item-type `book`
ZoteroTypes("book")
```

Index

across, [8](#), [18](#), [24](#), [28](#)
add_column, [35](#)
add_headers, [20](#), [35](#)
adist, [33](#)
all_of, [18](#)
arrange, [16](#), [18](#), [22](#), [27–29](#), [32](#)
as_tibble, [24](#), [35](#)

bind, [8](#), [22](#), [27](#), [29](#), [35](#)
bind_rows, [16](#)

case_when, [18](#), [28](#)
coalesce, [16](#), [33](#)
context, [28](#)
Cristin, [2](#)
CristinSupported, [6](#)
CristinWrangler, [2](#), [5](#), [7](#)

distinct, [8](#), [29](#), [32](#)
DoiCrossref, [8](#)
DoiDatacite, [9](#)

everything, [28](#)

filter, [8](#), [16](#), [18](#), [28](#)

GET, [5](#), [14](#), [18](#), [22](#)
group_by, [28](#)

head, [5](#), [33](#)
html_attr, [22](#), [29](#)
html_children, [22](#), [29](#)
html_name, [22](#)
html_text, [22](#), [29](#)
http_error, [5](#), [14](#), [22](#)

map, [24](#), [33](#)
mutate, [8](#), [18](#), [24](#), [28](#), [33](#), [35](#)
mutate_all, [24](#)

na_if, [24](#), [28](#)

nest, [8](#)

pmap, [18](#), [28](#)
pull, [33](#)

reexports, [8](#), [19](#), [22](#), [24](#), [27](#), [29](#), [33](#)
rename, [22](#), [29](#)
RETRY, [5](#), [8](#), [19](#), [20](#), [22](#), [26](#), [27](#), [29](#), [33](#), [35](#)

select, [18](#), [23](#), [24](#), [28](#), [33](#), [35](#)
setNames, [29](#), [35](#)
slice, [5](#)
stack, [35](#)
sym, [33](#)

tail, [5](#)
tibble, [7](#), [8](#), [24](#), [29](#), [32](#)

unnest, [8](#)

where, [23](#)

Zotero, [10](#)
ZoteroAdd, [14](#)
ZoteroCheck, [15](#)
ZoteroCopy, [17](#)
ZoteroCran, [19](#)
ZoteroDelete, [20](#)
ZoteroDoi, [12](#), [14](#), [21](#)
ZoteroExport, [22](#)
ZoteroFormat, [24](#)
ZoteroGet, [25](#)
ZoteroGov, [27](#)
ZoteroIndex, [28](#)
ZoteroIsbn, [12](#), [14](#), [29](#)
ZoteroKey, [30](#)
ZoteroLibrary, [30](#)
ZoteroMatch, [32](#)
ZoteroPost, [34](#)
ZoteroTypes, [36](#)