

Package: c2z (via r-universe)

October 13, 2026

Type Package

Title A Reference Manager

Version 0.3.0.9016

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, future, future.apply, httr, jsonlite, progressr, 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 7.3.2

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

Date/Publication 2025-04-10 18:44:32 UTC

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

RemoteRef HEAD

RemoteSha 779f122f39b0d19922f81fa9162b7afe96e27ff1

Contents

CheckDoi	2
CheckIsbn	3
Cristin	4
CristinSupported	8
CristinWrangler	9
DoiCrossref	10
DoiDatacite	11
ProcessData	12
Processing	14
UpdateInsert	16
Zotero	18
ZoteroAdd	22
ZoteroCheck	23
ZoteroCopy	24
ZoteroCran	26
ZoteroDelete	27
ZoteroDoi	29
ZoteroEnhancer	30
ZoteroExport	31
ZoteroFormat	33
ZoteroGet	34
ZoteroGov	36
ZoteroIndex	37
ZoteroIsbn	38
ZoteroKey	39
ZoteroLibrary	40
ZoteroMatch	42
ZoteroPost	43
ZoteroTypes	45
Index	47

CheckDoi

Extract DOI from a URL or String

Description

This function checks if the input `url` contains a valid Digital Object Identifier (DOI) using a regular expression. If a valid DOI is found, it extracts the DOI and optionally returns either the DOI string or the full DOI URL.

Usage

```
CheckDoi(url, doi.only = FALSE)
```

Arguments

url	A character string representing a URL or a DOI-containing text.
doi.only	Logical; if TRUE, the function returns only the DOI. If FALSE (default), the function returns the full DOI URL (i.e., "https://doi.org/" concatenated with the DOI).

Details

The function uses a regular expression to search for a DOI in the input. It leverages an external function `GoFish` (with `type = FALSE`) to perform additional checks. The DOI is then trimmed using a function called `Trim`. The regular expression pattern is designed to match the standard DOI format.

Value

Returns a character string with the DOI (or full DOI URL) if found; otherwise, returns `NULL`.

Examples

```
## Not run:
# Example 1: Extract full DOI URL from a string containing a DOI
doi_url <- CheckDoi("https://doi.org/10.1000/xyz123")
# doi_url will be "https://doi.org/10.1000/xyz123"

# Example 2: Extract only the DOI without the URL prefix
doi_only <- CheckDoi("10.1000/xyz123", doi.only = TRUE)
# doi_only will be "10.1000/xyz123"

## End(Not run)
```

CheckIsbn

Check and Convert ISBN

Description

This function processes an input string that contains an ISBN (or a compound string where the first element is the ISBN). It first removes unwanted characters (keeping only digits, commas, and spaces), then uses the `GoFish` function to further process the string. The function splits the cleaned string by spaces or commas to extract a candidate ISBN. If the candidate ISBN is in ISBN10 format, it is converted to ISBN13 by prepending "978", recalculating the check digit, and concatenating the result. Finally, the function validates that the resulting ISBN is a valid ISBN13, i.e., it is 13 digits long and starts with either "978" or "979".

Usage

`CheckIsbn(key)`

Arguments

key	A character string that contains an ISBN or a compound string where the first element is the ISBN.
-----	--

Details

The function performs the following steps:

1. Cleans the input key by removing all characters except digits, commas, and spaces.
2. Processes the cleaned key using the `GoFish` function (which must be defined in your environment).
3. Splits the resulting string by spaces and commas to extract the first candidate ISBN.
4. If the candidate is in ISBN10 format (10 characters), it converts it to ISBN13 by:
 - (a) Converting the first 9 digits to numeric.
 - (b) Prepending the ISBN13 prefix "978".
 - (c) Calculating the ISBN13 check digit using alternating weights of 1 and 3.
 - (d) Concatenating the digits to form the full ISBN13.
5. Validates that the final key is a valid ISBN13 (i.e., 13 digits long and starting with "978" or "979").

Value

A cleaned and validated ISBN string. If the resulting ISBN does not meet the expected criteria, the function returns `NULL`.

Examples

```
## Not run:
# Example with a compound string where the first part is the ISBN:
isbn_input <- "316148410X, some extra text"
valid_isbn <- CheckIsbn(isbn_input)

# Example converting an ISBN10 to ISBN13:
isbn10 <- "316148410X"
valid_isbn <- CheckIsbn(isbn10)

## End(Not run)
```

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 = "en",  
  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,  
  base.url = "https://api.cristin.no/v2/results",  
  custom.url = NULL,  
  use.identifiers = TRUE,  
  use.multisession = FALSE,  
  min.multisession = 25,  
  n.workers = NULL,  
  handler = NULL,  
  restore.defaults = TRUE,  
  zotero = NULL,  
  zotero.import = TRUE,  
  zotero.check = TRUE,  
  nvi = FALSE,  
  remove.na = TRUE,  
  replace.na = "book",  
  force.type = NULL,
```

```

    silent = FALSE,
    log = list()
)

```

Arguments

<code>id</code>	The Cristin id of the result, Default: NULL
<code>doi</code>	DOI name (e.g. doi=10.1000/123456), Default: NULL
<code>title</code>	The title of the result, Default: NULL
<code>contributor</code>	Author's name or Cristin person id, Default: NULL
<code>issn</code>	The issn of the result, Default: NULL
<code>unit</code>	Id (e.g. unit=185.53.18.10), Default: NULL
<code>institution</code>	Id (one number, e.g. institution=185), 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., 'askeladd:185'), 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 project_code together separated by colon (e.g., NFR:1234), Default: NULL
<code>lang</code>	Two letter language code , Default: en
<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>base.url</code>	The base url for the Cristin API, Default: https://api.cristin.no/v2/results
<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>use.multisession</code>	Logical. If TRUE (default), parallel processing using multisession is employed; otherwise, processing is sequential.
<code>min.multisession</code>	Minimum number of results for using multisession. Default: 25
<code>n.workers</code>	Optional integer for the number of workers to be used in multisession mode. If NULL, it defaults to the number of available cores minus one (with a minimum of one).
<code>handler</code>	The progress handler to be used by the <code>progressr</code> package. If NULL and <code>silent</code> is FALSE, it defaults to "txtprogressbar". When <code>silent</code> is TRUE, the handler is set to "void".
<code>restore.defaults</code>	Logical. If TRUE (default), the current future plan is saved and restored upon exit.
<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>silent</code>	c2z is noisy, tell it to be quiet, Default: FALSE
<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,
  lang = "en",
  silent = FALSE,
  log = list()
)
```

Arguments

<code>data</code>	Zotero-type matrix, Default: NULL
<code>zotero.check</code>	Should the function look for duplicates?, Default: TRUE
<code>items</code>	Items in library to check for
<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>remove.duplicates</code>	Remove duplicates if TRUE, Default: TRUE
<code>lang</code>	Two letter language code , Default: en
<code>silent</code>	c2z is noisy, tell it to be quiet, Default: FALSE
<code>log</code>	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

Converts Cristin metadata into a Zotero-style tibble.

Usage

```
CristinWrangler(data, meta = list(), use.identifiers = TRUE)
```

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>

Details

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

Value

A Zotero-type tibble.

DoiCrossref	<i>Wrangle CrossRef metadata into Zotero-type format</i>
-------------	--

Description

Query CrossRef by DOI and fetch metadata

Usage

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

Arguments

<code>data</code>	XML data from CrossRef containing metadata
<code>meta</code>	A list collecting all metadata used to create , Default: <code>list()</code>
<code>use.json</code>	Use either json (<code>TRUE</code>) or XML (<code>FALSE</code>), Default: <code>TRUE</code>
<code>silent</code>	<code>c2z</code> is noisy, tell it to be quiet, Default: <code>TRUE</code>
<code>log</code>	A list for storing log elements, Default: <code>list()</code>

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
}
```

ProcessData

Process data with Parallel Execution and Row/Chunk Processing

Description

This function processes a tibble (or data frame) in parallel using the future package along with progressr for progress reporting. The data is split into chunks via a helper function SplitData. Depending on the by.rows flag, each chunk is processed either row-by-row (using lapply) or as an entire chunk. In either case, the processed results are bound into a single tibble.

Usage

```
ProcessData(
  data,
  func,
  by.rows = TRUE,
  min.multisession = 1,
  n.workers = NULL,
  limit = 100,
  use.multisession = TRUE,
  n.message = 1,
  start.message = NULL,
  process.message = NULL,
  end.message = NULL,
  restore.defaults = TRUE,
  handler = NULL,
  silent = FALSE,
  log = list(),
  ...
)
```

Arguments

<code>data</code>	Data to be processed.
<code>func</code>	A function that processes either a single row (if <code>by.rows = TRUE</code>) or an entire chunk (if <code>by.rows = FALSE</code>). Additional arguments are passed to <code>func</code> via <code>...</code>
<code>by.rows</code>	Logical. If <code>TRUE</code> , <code>func</code> is applied to each row individually within each chunk; if <code>FALSE</code> , <code>func</code> is applied to the entire chunk. Default is <code>TRUE</code> .
<code>min.multisession</code>	Integer. The minimum number of rows in data required to use multisession (parallel execution). If <code>nrow(data)</code> is less than this value, parallel processing is disabled. Default is 1.
<code>n.workers</code>	Integer. The number of workers to use for parallel processing. Defaults to <code>max(1, future::availableCores() - 1)</code> .
<code>limit</code>	Integer. If <code>(nrow(data) / limit)</code> is less than or equal to <code>n.workers</code> , the data is split into <code>n.workers</code> ; otherwise, it is split using <code>limit</code> . Default is 100.
<code>use.multisession</code>	Logical. If <code>TRUE</code> (default) parallel processing is used; otherwise, sequential processing is employed.
<code>n.message</code>	Integer. Display a process message every <code>nth</code> iteration. Defaults to 1. Turn off messages with 0.
<code>start.message</code>	Optional character string. A custom message to log at the start of processing.
<code>process.message</code>	Optional column name. A custom message to log during processing.
<code>end.message</code>	Optional character string. A custom message to log at the end of processing.
<code>restore.defaults</code>	Logical. If <code>TRUE</code> (default), the original future plan and progress handlers are restored after processing.
<code>handler</code>	Character string. The progress handler to be used by the <code>progressr</code> package. Defaults to <code>"txtprogressbar"</code> unless <code>silent</code> is <code>TRUE</code> .
<code>silent</code>	Logical. If <code>TRUE</code> , progress messages and logging are suppressed. Default is <code>FALSE</code> .
<code>log</code>	List. An initial log (default is an empty list) to which log messages are appended.
<code>...</code>	Additional arguments passed to the processing function <code>func</code> .

Details

The function first checks if the data has enough rows to warrant parallel processing. It then determines the number of workers and chunks and splits the data using the helper function `SplitData`. The processing is executed with `future.apply::future_lapply` in parallel. If `by.rows` is `TRUE`, each row within a chunk is processed individually and then bound together using `dplyr::bind_rows`; if `by.rows` is `FALSE`, the entire chunk is processed at once. Finally, all chunk results are combined into a single tibble.

Value

A list with two components:

results A tibble resulting from binding all processed rows from each chunk.

log A list of log messages generated during processing.

Examples

```
## Not run:
example.result <- ProcessData(
  data = tibble::tibble(
    key = LETTERS[1:3],
    value = seq(3)
  ),
  func = \(data) {
    data$value <- data$value * 10
    data
  },
  by.rows = FALSE,
  min.multisession = 10,
  n.workers = 4,
  limit = 100,
  use.multisession = TRUE,
  start.message = "Starting example data processing",
  end.message = "Finished example data processing",
  handler = "txtprogressbar",
  silent = FALSE
)
# The result is a single tibble.
processed.tibble <- example.result$results
process.log <- example.result$log
if (any(nrow(processed.tibble))) print(processed.tibble)

## End(Not run)
```

Processing

Process Data with Optional Parallel Execution and Progress Reporting

Description

This function executes a provided expression with progress reporting and optional parallel processing using the `future` and `progressr` packages. It logs the start and end times of the process along with a runtime message and supports custom logging, progress handlers, and execution plans.

Usage

```
Processing(
  func = substitute(NULL),
  n = NULL,
  start.message = NULL,
  end.message = NULL,
  use.multisession = TRUE,
  n.workers = NULL,
  handler = NULL,
  restore.defaults = TRUE,
  silent = FALSE,
  log = list()
)
```

Arguments

<code>func</code>	An expression (or a quoted function call) to be evaluated. The default is <code>substitute(NULL)</code> . It is evaluated in the parent frame.
<code>n</code>	Optional integer indicating the number of items to be processed, used solely for logging purposes.
<code>start.message</code>	Optional character string specifying a custom start message. If <code>NULL</code> , a default message is generated based on the processing type and, if provided, the number of items.
<code>end.message</code>	Optional character string specifying a custom end message. If <code>NULL</code> , a runtime message is generated showing the elapsed time.
<code>use.multisession</code>	Logical. If <code>TRUE</code> (default), parallel processing using multisession is employed; otherwise, processing is sequential.
<code>n.workers</code>	Optional integer for the number of workers to be used in multisession mode. If <code>NULL</code> , it defaults to the number of available cores minus one (with a minimum of one).
<code>handler</code>	The progress handler to be used by the <code>progressr</code> package. If <code>NULL</code> and <code>silent</code> is <code>FALSE</code> , it defaults to <code>"txtprogressbar"</code> . When <code>silent</code> is <code>TRUE</code> , the handler is set to <code>"void"</code> .
<code>restore.defaults</code>	Logical. If <code>TRUE</code> (default), the current future plan is saved and restored upon exit.
<code>silent</code>	Logical. If <code>TRUE</code> , progress output is suppressed. Default is <code>FALSE</code> .
<code>log</code>	A list to collect log messages. Default is an empty list.

Details

The function begins by determining whether to use multisession (parallel) or sequential processing. It then logs the start time using a helper function, `LogCat`, and sets up the future plan accordingly. If `restore.defaults` is `TRUE`, the original plan is restored after execution.

A progress handler is configured via the `progressr` package. The provided expression is evaluated with progress reporting enabled. Upon completion, the function computes the elapsed time (with millisecond precision) and logs an end message. Finally, it returns a list containing the result of the evaluated expression and the log of messages.

Value

A list with the following components:

result The output resulting from the evaluation of `func`.

log A list of log messages including start and end timestamps along with associated messages.

Examples

```
## Not run:
# Example: Process data with a simple sleep expression using multisession processing
result <- Processing(
  func = quote({
    Sys.sleep(2) # simulate a time-consuming task
    "Task Completed"
  }),
  n = 100,
  use.multisession = TRUE
)
print(result$result)
print(result$log)

## End(Not run)
```

UpdateInsert

Upsert and Update Data Frames

Description

Combines two data frames by updating rows in `x` with values from `y` based on a common key, and inserting new rows from `y` that are not present in `x`. The function first harmonizes the column structures of both data frames by adding missing columns and coercing types as necessary.

Usage

```
UpdateInsert(x, y, key = "key", check.missing = FALSE)
```

Arguments

<code>x</code>	A data frame to be updated.
<code>y</code>	A data frame containing new values to update <code>x</code> . Must include the column specified by <code>key</code> .

key	A character string specifying the unique key column used for matching rows. Defaults to "key".
check.missing	Logical; if TRUE, performs a cell-by-cell update only when the new value is not missing. Missing values are defined as NA for atomic types or an empty list for list columns. If FALSE, a standard upsert is performed using <code>dplyr::rows_upsert</code> . Defaults to FALSE.

Details

The function works in several steps:

1. It computes the union of all column names from `x` and `y` and adds any missing columns to both data frames using the internal helper function `AddColumns`. Missing columns are filled with an appropriate NA value based on their type.
2. Both `x` and `y` are reordered to have the same column order.
3. For each common column (excluding the key), if `x`'s column is entirely NA or if the data types differ, coercion is performed to ensure compatibility between `x` and `y`.
4. When `check.missing` is TRUE, the function iterates over each common key and updates each cell in `x` only if the corresponding cell in `y` is not missing. Otherwise, it uses `dplyr::rows_upsert` to perform a standard upsert.
5. New rows present in `y` but not in `x` are appended.

Value

A data frame resulting from updating `x` with values from `y`.

Examples

```
## Not run:
# Example data frames:
df1 <- data.frame(
  key = 1:3,
  a = c(NA, 2, NA),
  b = c("x", NA, "z"),
  stringsAsFactors = FALSE
)

df2 <- data.frame(
  key = c(2, 3, 4),
  a = c(5, 6, 7),
  b = c("y", "w", "v"),
  stringsAsFactors = FALSE
)

# Standard upsert (check.missing = FALSE):
result <- UpdateInsert(df1, df2, key = "key", check.missing = FALSE)

# Cell-by-cell update (check.missing = TRUE):
result <- UpdateInsert(df1, df2, key = "key", check.missing = TRUE)
```

```
## End(Not run)
```

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,
post.token = FALSE,
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

create	Create missing collections, Default: FALSE
limit	Number of results per query (max 100), Default: 100
start	Starting position of query (0 = first result), Default: 0
search.collections	Search all collections if collection.key fails, Default: TRUE
get.collections	Fetch collections, Default: TRUE
get.items	Fetch items, Default: TRUE
item.type	Items to search for (NULL = everything), Default: NULL
all.results	Find all results in query, Default: TRUE
max.results	Do you need a limit?, Default: NULL
collections	Predefined collections (as tibble), Default: NULL
items	Predefined metadata (as tibble), Default: NULL
attachments	Predefined attachments (as tibble), Default: NULL
metadata	Predefined metadata in Zotero-format, Default: NULL
doi	Use ZoteroDoi to fetch DOI metadata, Default: NULL
isbn	Use ZoteroIsbn to fetch ISBN metadata, Default: NULL
export	Use 'ZoteroExport' to export items, Default: FALSE
csl.type	Specify a CSL type to Official repository for Citation Style Language (CSL), Default: NULL
csl.name	Name of saved CSL file, Default: 'style'
format	Export format of Zotero items, Default: 'biblatex'
save.data	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
post.token	Use Zotero-Write-Token (TRUE) or If-Unmodified-Since-Version (FALSE), Default: FALSE
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
base.url	Base url of the Zotero API, Default: 'https://api.zotero.org'
silent	c2z is noisy, tell it to be quiet, Default: FALSE
zotero	A list with information on the specified Zotero library (e.g., id, API key, collections, and items), Default: NULL
log	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 = "Io1bwAi0jB1jPgUNegjJhJxF")
# Print the interesting parts 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

zotero	A list with information on the specified Zotero library (e.g., id, API key, collections, and items)
metadata	Predefined metadata in Zotero-format, Default: NULL
doi	Use ZoteroDoi to fetch DOI metadata, Default: NULL
isbn	Use ZoteroIsbn to fetch ISBN metadata, Default: NULL
silent	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 = "Io1bwAi0jB1jPgUNegjJhJxF"
  ),
  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

Removes references from the input dataset that are duplicates already added to Zotero, preserving only those references that have been modified since their addition.

Usage

```

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

```

Arguments

<code>data</code>	A tibble containing metadata (e.g., from Cristin)
<code>id</code>	A string specifying the column name in <code>data</code> that contains the unique identifier (e.g., "cristin_result_id")
<code>id.type</code>	A string specifying the type of metadata as stored in the Zotero extra field (e.g., "Cristin")
<code>created</code>	A string specifying the column name in <code>data</code> containing the creation date in UNIX timestamp format (e.g., "created")
<code>last.modified</code>	A string specifying the column name in <code>data</code> containing the last modification date in UNIX timestamp format (e.g., "last_modified")
<code>items</code>	A data frame or tibble representing items in the Zotero library to check against
<code>remove.duplicates</code>	Logical. If TRUE, duplicates that have not been modified are removed. Default is TRUE.
<code>silent</code>	Logical. If TRUE, suppresses verbose output. Default is FALSE.
<code>log</code>	A list for storing log messages. Default is an empty list.

Details

For more details, see [c2z documentation](#).

Value

A list with two elements: data, a tibble containing non-duplicated items, and log, the updated log list.

See Also

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

Examples

```
## Not run:
# Simple Cristin search by id
cristin.data <- Cristin(
  id = "840998",
  zotero.import = FALSE
)

# Checking Zotero library for duplicates
result <- ZoteroCheck(
  data = cristin.data$result,
  id = "cristin_result_id",
  id.type = "Cristin",
  created = "created",
  last.modified = "last_modified",
  items = Zotero(
    user = FALSE,
    id = "4827927",
    api = "Io1bwAi0jB1jPgUNegjJhJxF",
    library = TRUE,
    silent = TRUE
  )$items
)

## End(Not run)
```

ZoteroCopy

Copy collections and items from a Zotero library

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 = "Io1bwAi0jB1jPgUNegjJhJxF",
  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)
}
```

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 = "Io1bwAi0jB1jPgUNegjJhJxF"
  ),
  ragnarok = TRUE,
  force = TRUE
)
```

ZoteroDoi*Use DOI to acquire metadata*

Description

Connects with doi.org to create metadata

Usage

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

Arguments

doi	A digital object identifier
meta	A list collecting all metadata used to create , Default: list()
use.semantic	Get metadata (abstract) from Semantic Scholar, Default: FALSE
check.retraction	Check if marked as retracted, Default: TRUE
use.json	Use either json (TRUE) or XML (FALSE), 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)
}
```

ZoteroEnhancer

Enhance Zotero Records with External Metadata via Zotero Lookups

Description

This function iterates over a data frame (or tibble) of Zotero records and attempts to enhance each record by retrieving additional metadata from external sources. It uses DOI or ISBN information, via the `c2z` package, to fetch external metadata and then merges these updates into the original Zotero record. Special handling is provided for records representing book sections.

Usage

```
ZoteroEnhancer(
  zotero.data,
  external.data = NULL,
  use.doi = TRUE,
  use.semantic = TRUE,
  use.isbn = TRUE
)
```

Arguments

<code>zotero.data</code>	A data frame (or tibble) containing Zotero records. Each row represents a record that may include fields such as <code>abstractNote</code> , <code>DOI</code> , <code>url</code> , <code>ISBN</code> , <code>key</code> , <code>itemType</code> , <code>pages</code> , and <code>creators</code> .
<code>external.data</code>	Optional. A data frame or list containing pre-fetched external metadata. If provided and non-empty, the function will use this external metadata to update the Zotero records instead of querying the external APIs. Defaults to <code>NULL</code> .
<code>use.doi</code>	Logical; if <code>TRUE</code> (default), the function will attempt to retrieve external metadata using DOI information. Defaults to <code>TRUE</code> .
<code>use.semantic</code>	Logical; if <code>TRUE</code> (default), the function will attempt to retrieve external metadata from Semantic Scholar using DOI information. Defaults to <code>TRUE</code> .
<code>use.isbn</code>	Logical; if <code>TRUE</code> (default), the function will attempt to retrieve external metadata using ISBN information when DOI retrieval is not successful. Defaults to <code>TRUE</code> .

Details

For each record in `zotero.data`, the function performs the following steps:

1. If `external.data` is provided and non-empty, that data is used to update the record.
2. Otherwise, the function skips processing the record if an abstract already exists.
3. It then attempts to extract a DOI from the DOI field, or if missing, from the url field.
4. The ISBN field is processed by removing spaces and extracting the first value (in case of comma separation).
5. If a valid DOI is found and `use.doi` is TRUE, external metadata is retrieved using `c2z::ZoteroDoi()`. Otherwise, if an ISBN is available and `use.isbn` is TRUE, metadata is retrieved using `c2z::ZoteroIsbn()`.
6. Once external metadata is successfully retrieved, an internal helper function `UpdateTibble` is called. This helper function merges the external metadata into the original record. For book sections, additional adjustments are made to handle creator information, pages, and edition fields.
7. Finally, the enhanced record is merged back into the original data frame using `UpdateInsert()`.

Value

A data frame containing the enhanced Zotero records with external metadata merged into the original records.

Examples

```
## Not run:
# Enhance Zotero records using DOI/ISBN lookup:
example <- Cristin("762394", silent = TRUE)$results
print(example$libraryCatalog)
example$DOI <- "https://doi.org/10.22358/jafs/74164/2007"
enhanced.data <- ZoteroEnhancer(example)
print(enhanced.data$libraryCatalog)

## End(Not run)
```

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
<code>csl.name</code>	Name of saved CSL file, Default: 'style'
<code>locale</code>	Desired language format of bibliography, Default: 'en-US'
<code>format</code>	Export format of Zotero items, Default: 'biblatex'
<code>bib.name</code>	Name of exported bibliography, Default: 'references'
<code>save.data</code>	Save data (e.g., bibliography) to disk, Default: FALSE
<code>save.path</code>	Location to store data on disk, Default: NULL
<code>silent</code>	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 = "Io1bwAi0jB1jPgUNegjJhJxF",
```

```

    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,
  fix.columns = TRUE,
  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
fix.columns	Remove columns not part of the Zotero type, Default: TRUE
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
append.file	Append '/file/' to Zotero API url, Default: FALSE
custom.url	Use a custom Zotero API url, Default: NULL
open.query	Use your own query, Default: NULL
limit	Number of results per query (max 100), Default: 100
start	Starting position of query (0 = first result), Default: 0
format	Format of response from the Zotero API, Default: 'json'
item.keys	Specified vector of items keys, Default: NULL
collection.keys	Specified vector of collection keys, Default: NULL
item.type	Items to search for (NULL = everything), Default: NULL
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'
all.results	Find all results in query, Default: TRUE
max.results	Do you need a limit?, Default: NULL
result.type	Pointless linguistics to display result type (default = 'result'), Default: NULL
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

[RETRY](#)

Examples

```
# Define Zotero list according to default settings
zotero = Zotero(
  user = FALSE,
  id = "5250382",
  api = "Io1bwAi0jB1jPgUNegjJhJxF"
)

# 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	<i>Create an index of Zotero items</i>
-------------	--

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 = "Io1bwAi0jB1jPgUNegjJhJxF",
  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
<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: apa
<code>locale</code>	Desired language format of bibliography, Default: 'en-US'
<code>force</code>	Force is seldom wise, but sometimes..., Default: FALSE
<code>silent</code>	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 = "Io1bwAi0jB1jPgUNegjJhJxF"
  )
)
# Print index using `ZoteroIndex`
if (any(nrow(example$items))) {
  ZoteroIndex(example$items) |>
  dplyr::select(name) |>
  print(width = 80)
}
```

ZoteroMatch

Search or match items using CrossRef

Description

Use the CrossRef API to match data (e.g., Cristin 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,
  use.semantic = FALSE,
  log = list()
)
```

Arguments

title	Title of reference
authors	creators of the reference
date	publication date of the reference
haystack	Potential matches for search term (i.e., needle), Default: NULL
haystack.size	Number of items in the haystack, Default: 3
crossref.search	Search CrossRef if needle not found in haystack, Default: FALSE
autosearch	Match automatically or compare needle with haystack, Default: FALSE
cristin.data	Metadata from Cristin, Default: NULL
external.data	Metadata from external source (e.g., CrossRef), Default: NULL
polite	Will use an email stored in '.Renviron', Default: TRUE
silent	Running silent, running deep, Default: FALSE
use.semantic	Prefer metadata from Semantic Scholar, Default: FALSE
log	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,
  post.token = FALSE,
  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>post.collections</code>	Try to copy specified collections, Default: TRUE
<code>post.items</code>	Try to copy specified items?, Default: TRUE
<code>post.attachments</code>	Try to copy specified extras (i.e., attachments and notes)?, Default: TRUE
<code>post.limit</code>	Number of collections/items to post per request (max 50), Default: 50
<code>post.token</code>	Use Zotero-Write-Token (TRUE) or If-Unmodified-Since-Version (FALSE), Default: FALSE
<code>force</code>	Force is seldom wise, but sometimes..., Default: FALSE
<code>silent</code>	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 = "Io1bwAi0jB1jPgUNegjJhJxF"
)

# 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, [25](#), [34](#), [37](#)
add_column, [44](#)
add_headers, [28](#), [44](#)
adist, [43](#)
all_of, [25](#)
arrange, [24](#), [25](#), [29](#), [37](#), [38](#), [41](#)
as_tibble, [34](#), [44](#)

bind, [29](#), [37](#), [38](#), [44](#)
bind_rows, [24](#)

case_when, [25](#), [37](#)
CheckDoi, [2](#)
CheckIsbn, [3](#)
coalesce, [24](#), [43](#)
context, [37](#)
Cristin, [4](#)
CristinSupported, [8](#)
CristinWrangler, [4](#), [7](#), [9](#)

distinct, [38](#), [41](#)
DoiCrossref, [10](#)
DoiDatacite, [11](#)

everything, [37](#)

filter, [24](#), [25](#), [37](#)

GET, [8](#), [21](#), [25](#), [29](#)
group_by, [37](#)

head, [8](#), [43](#)
html_attr, [29](#), [38](#)
html_children, [29](#), [38](#)
html_name, [29](#)
html_text, [29](#), [38](#)
http_error, [8](#), [21](#), [29](#)

map, [34](#), [43](#)
mutate, [25](#), [34](#), [37](#), [43](#), [44](#)
mutate_all, [34](#)

na_if, [34](#), [37](#)

pmap, [25](#), [37](#)
ProcessData, [12](#)
Processing, [14](#)
pull, [43](#)

reexports, [27](#), [29](#), [34](#), [37](#), [38](#), [43](#)
rename, [29](#), [38](#)
RETRY, [8](#), [27–29](#), [35](#), [37](#), [38](#), [43](#), [44](#)

select, [25](#), [32](#), [34](#), [37](#), [43](#), [44](#)
setNames, [38](#), [44](#)
slice, [8](#)
stack, [44](#)
sym, [43](#)

tail, [8](#)
tibble, [9](#), [34](#), [38](#), [41](#)

UpdateInsert, [16](#)

where, [32](#)

Zotero, [18](#)
ZoteroAdd, [22](#)
ZoteroCheck, [23](#)
ZoteroCopy, [24](#)
ZoteroCran, [26](#)
ZoteroDelete, [27](#)
ZoteroDoi, [20](#), [22](#), [29](#)
ZoteroEnhancer, [30](#)
ZoteroExport, [31](#)
ZoteroFormat, [33](#)
ZoteroGet, [34](#)
ZoteroGov, [36](#)
ZoteroIndex, [37](#)
ZoteroIsbn, [20](#), [22](#), [38](#)
ZoteroKey, [39](#)
ZoteroLibrary, [40](#)
ZoteroMatch, [42](#)

ZoteroPost, [43](#)
ZoteroTypes, [45](#)