Package: citezot (via r-universe)

November 6, 2026

Title Cite Zotero References in Quarto Documents

Version 0.0.0.9000

Description Scans Quarto and R Markdown documents for citation keys, pulls the matching items from the Zotero web API, and writes a bibliography file. Also provides dry-run-by-default helpers to find, add, and update Zotero items.

License MIT + file LICENSE

URL https://joelnitta.github.io/citezot/, https://github.com/joelnitta/citezot

BugReports https://github.com/joelnitta/citezot/issues

Config/roxygen2/version 8.0.0

Encoding UTF-8

Language en

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/Needs/website pkgsite, ropensci-review-tools/babelquarto

Config/testthat/edition 3

VignetteBuilder knitr

Imports cli, httr2, rlang, yaml

Config/pak/sysreqs libssl-dev

Repository https://joelnitta.r-universe.dev

Date/Publication 2026-11-06 00:25:38 UTC

RemoteUrl https://github.com/joelnitta/citezot

RemoteRef HEAD

RemoteSha 3683957d3977eac78d8fc8c473132763d0535a76

Contents

cz_add . . . 2
cz_auth . . . 3
cz_bare_keys . . . 4
cz_clear_cache . . . 5
cz_fetch_items . . . 5
cz_find . . . 7
cz_get_item . . . 8
cz_item_key . . . 8
cz_key_format . . . 9
cz_pin_key . . . 10
cz_qmd_keys . . . 11
cz_set_auth . . . 12
cz_suggest_key . . . 13
cz_update_bib . . . 14
cz_update_item . . . 15

Index **17**

`cz_add` *Add an item to your Zotero library from a DOI*

Description

Looks up the metadata for a DOI (via the public Crossref API), checks that the item is not already in your library, and creates it. The default is a **dry run**: the library is *read* and Crossref is queried, but nothing is written to Zotero. Use `dry_run = FALSE` to create the item.

Usage

```
cz_add(
  id,
  key = NULL,
  collection = NULL,
  dry_run = TRUE,
  auth = cz_auth(),
  cache = getOption("citezot.cache", FALSE)
)
```

Arguments

`id`	A DOI, with or without a `https://doi.org/` or `doi:` prefix.
`key`	Optional citekey to pin on the new item, e.g. `"Kent2002"`. It must be usable as a pandoc citation key and not already used in your library. Use `"auto"` to generate one from your Better BibTeX formula (see `cz_suggest_key()` and `cz_key_format()`); a letter is appended if the key is taken. The key is shown in the dry run.

collection	Optional collection to add the item to: its 8-character key, or its exact name.
dry_run	If TRUE (default), show what would be created and write nothing.
auth	Credentials from cz_auth() (a key with write access to write).
cache	Passed to cz_fetch_items(), which is used for the duplicate check.

Details

The new item is built from Zotero's own template for its type, so it only contains fields valid for that type. The citekey, if you give one, is pinned in the same request (the native citationKey field), so there is no moment where the item exists without it. Only DOIs registered with Crossref are supported for now; ISBN, arXiv and PMID lookups are planned.

Creating is retry-safe: each request carries a random Zotero-Write-Token, so a retried request after a network hiccup cannot create the item twice. Existing items are never changed or deleted.

Crossref asks clients to identify themselves with a contact address to use its faster pool. citezot never sends an address by itself; to opt in, set options(citezot.mailto = "you@example.org").

Value

Invisibly, a list with item (the payload), key (the new Zotero item ID, or NULL), existing (matching rows of cz_find() if the item was already in the library) and written.

Examples

```
## Not run:
cz_add("10.3767/000651906X622210", key = "Ebihara2006") # preview
cz_add("10.3767/000651906X622210", key = "Ebihara2006", dry_run = FALSE)

## End(Not run)
```

---

cz_auth *Zotero credentials*

Description

Resolves the library ID and API key, by default from the environment variables ZOTERO_LIBRARY_ID and ZOTERO_API_KEY. Set them once with cz_set_auth(), or load them yourself (e.g. from a gitignored .Renviron or .env.local). citezot never reads credential files; only cz_set_auth() writes the key, and only when you ask it to. The key is never printed.

Usage

```
cz_auth(
  library_id = NULL,
  api_key = Sys.getenv("ZOTERO_API_KEY"),
  library_type = c("user", "group")
)
```

Arguments

library_id	Numeric user ID (or group ID), as a string or number. If NULL (default), taken from ZOTERO_LIBRARY_ID.
api_key	Zotero API key.
library_type	"user" (personal library) or "group".

Details

Create a key at https://www.zotero.org/settings/keys; the numeric user ID is on the same page. A read-only key is enough for cz_fetch_items(); only functions that change the library need a key with write access.

Value

A cz_auth object, for the auth argument of other functions.

Examples

```
# Explicit values (normally these come from environment variables). The key
# is never shown when the object is printed.
cz_auth(library_id = "1234567", api_key = "not-a-real-key")
```

---

cz_bare_keys *Extract bare (no @) citekey-shaped tokens from a text/data file*

Description

For files that reference sources informally (e.g. a CSV "comments" column saying "defined in Dubuisson2013a"). Matches an author-shaped run of letters (hyphens allowed) immediately followed by a 19xx/20xx year and an optional lowercase disambiguator, e.g. Ebihara2006, Boucheron-Dubuisson2024. Deliberately conservative: Ebihara 2006 or Hsu7949 do not match.

Usage

```
cz_bare_keys(path)
```

Arguments

path	Path to the file to scan.

Value

Sorted character vector of unique citekey-shaped tokens.

Examples

```
notes <- tempfile(fileext = ".csv")
writeLines(
  c("site,comment", "A,defined in Darwin1859", "B,see Ebihara 2006"),
  notes
)
cz_bare_keys(notes) # "Ebihara 2006" does not match
```

`cz_clear_cache` *Delete citezot's cached library data*

Description

Removes cache files created by `cz_fetch_items()` when caching is enabled. Only citezot's own cache files are removed; nothing in Zotero is touched.

Usage

```
cz_clear_cache(auth = NULL)
```

Arguments

`auth`	Credentials from `cz_auth()` to clear just that library's cache, or `NULL` (default) to clear every cached library.

Value

Invisibly, the number of files removed.

Examples

```
# Use an empty temporary cache directory so nothing real is deleted
old <- options(citezot.cache_dir = tempfile("citezot-cache"))
cz_clear_cache()
options(old)
```

`cz_fetch_items` *Fetch every top-level item in a Zotero library as CSL-JSON*

Description

Uses the `/items/top` endpoint (excludes notes, attachments and other child items) and pages through the whole library using the `Total-Results` header. After the first page, the remaining pages are requested in parallel (at most `getOption("citezot.max_active", 4)` at a time, and throttled) and the library version is checked so that a library edited mid-download is re-fetched rather than returned inconsistent.

Usage

```
cz_fetch_items(
  auth = cz_auth(),
  page_size = 100,
  cache = getOption("citezot.cache", FALSE),
  refresh = FALSE
)
```

Arguments

auth	Credentials from cz_auth().
page_size	Items per request (Zotero's maximum is 100).
cache	Use and update the local cache? Defaults to getOption("citezot.cache", FALSE).
refresh	If TRUE, ignore any existing cache and download the whole library (the cache, if enabled, is then rewritten).

Details

A full download of a large library takes a while. With cache = TRUE the items are stored in a local file (in tools::R_user_dir("citezot", "cache"), or getOption("citezot.cache_dir")), and later calls download only what changed since: one request if nothing changed, otherwise the changed items plus the lists of deleted and trashed items. The cache holds the library's bibliographic metadata, never your API key. It is opt-in: nothing is written to disk unless you ask, either here or with options(citezot.cache = TRUE). Use refresh = TRUE to ignore the cache and download everything again, and cz_clear_cache() to delete it.

Value

A list of CSL-JSON items (plain lists, simplifyVector = FALSE). The order is not guaranteed.

Examples

```
## Not run:
items <- cz_fetch_items()
length(items)

# Keep a local copy, so later calls download only what changed
items <- cz_fetch_items(cache = TRUE)

## End(Not run)
```

cz_find *Find items in your Zotero library*

Description

Searches the library for items matching *all* the criteria given. Matching is done locally on the library's CSL-JSON (see `cz_fetch_items()`), because Zotero's server-side search cannot look items up by DOI or citekey. Use `cache = TRUE` (or `options(citezot.cache = TRUE)`) to make this fast on repeat calls.

Usage

```
cz_find(
  query = NULL,
  doi = NULL,
  isbn = NULL,
  citekey = NULL,
  auth = cz_auth(),
  cache = getOption("citezot.cache", FALSE),
  items = NULL
)
```

Arguments

`query`	Free text. Every word must appear (case-insensitively) in the title, authors, year, journal or citekey.
`doi`	A DOI, with or without a `https://doi.org/` or `doi:` prefix.
`isbn`	An ISBN (hyphens and spaces are ignored).
`citekey`	An exact pinned citekey.
`auth`	Credentials from `cz_auth()`.
`cache`	Passed to `cz_fetch_items()`.
`items`	Items already fetched with `cz_fetch_items()`, to avoid fetching again.

Value

A data frame with one row per match and columns `item` (Zotero item ID), `citekey`, `author`, `year`, `title`, `doi` and `type`. It can be passed as the `item` argument of `cz_pin_key()` when it has one row.

Examples

```
## Not run:
cz_find(doi = "10.3767/000651906X622210")
cz_find("Ebihara Hymenophyllaceae revision")
cz_find(citekey = "Ebihara2006")

## End(Not run)
```

cz_get_item — *Get one item in Zotero's native JSON*

Description

Returns the item as Zotero stores it (not CSL-JSON): writable fields are in `$data`, and `$version` is the item version used for safe updates.

Usage

```
cz_get_item(item, auth = cz_auth(), cache = getOption("citezot.cache", FALSE))
```

Arguments

item	The item: a DOI, Zotero item ID, citekey, or one-row result of `cz_find()` (see `cz_pin_key()`).
auth	Credentials from `cz_auth()`.
cache	Passed to `cz_fetch_items()` when `item` has to be looked up.

Value

A list with `key`, `version`, `data` and other API fields.

Examples

```
## Not run:
item <- cz_get_item("Darwin1859")
item$data$title
item$version

## End(Not run)
```

cz_item_key — *Get a CSL item's pinned citekey, however it is pinned*

Description

Checks the native `citation-key` property first, then falls back to a `Citation Key:` line in `note`.

Usage

```
cz_item_key(item)
```

Arguments

item	A single CSL-JSON item (a list).

Value

The citekey, or NA_character_.

Examples

```
cz_item_key(list(`citation-key` = "Darwin1859"))
cz_item_key(list(note = "Citation Key: Darwin1859"))
cz_item_key(list(title = "No key pinned"))
```

---

cz_key_format *Which key formula citezot will use*

Description

In order: getOption("citezot.key_format"); the Better BibTeX pattern in your Zotero profile (see Details); Better BibTeX's default pattern.

Usage

```
cz_key_format()
```

Details

Detection of the Zotero profile reads one line of its prefs.js and only happens when a key is generated, never at load. Set options(citezot.key_format = ...) to choose a formula yourself and skip the detection, and options(citezot.zotero_profile = "<folder>") if Zotero keeps its profile somewhere unusual. The source in use is reported once per session.

Value

A list with formula and source ("option", "zotero profile" or "default").

Examples

```
options(citezot.key_format = "auth.lower + year")
cz_key_format()
```

cz_pin_key *Pin a citekey onto a Zotero item*

Description

Sets the item's native `citationKey` field via the Web API (the same field a desktop Better BibTeX install manages), so `cz_item_key()` and `cz_fetch_items()` see it. Requires a key with write access.

Usage

```
cz_pin_key(
  item,
  key,
  dry_run = TRUE,
  overwrite = FALSE,
  auth = cz_auth(),
  cache = getOption("citezot.cache", FALSE)
)
```

Arguments

`item`	The item to pin: a DOI, a Zotero item ID (bare, `"Q62TD7I9"`, or with the library ID as in CSL-JSON, `"6265773/Q62TD7I9"`), an existing citekey, or a one-row result of `cz_find()`. Zero or several matches is an error.
`key`	The citekey to pin.
`dry_run`	If `TRUE` (default), report what would change and write nothing.
`overwrite`	Replace a *different* key that is already pinned. Defaults to `FALSE`, because changing a pinned key breaks documents citing it.
`auth`	Credentials from `cz_auth()`.
`cache`	Passed to `cz_fetch_items()` when `item` is a DOI or citekey that has to be looked up.

Details

The default is a dry run: the item is *read* (to get its current version and key) but nothing is written. Set `dry_run = FALSE` to write. The write sends `If-Unmodified-Since-Version`, so if the item changed in the meantime Zotero rejects it (HTTP 412) and nothing is overwritten.

Value

Invisibly, a list with `item`, `before`, `after` and `written`.

Examples

```
## Not run:
cz_pin_key("10.1016/0022-5193(64)90038-4", "Hamilton1964") # dry run
cz_pin_key("10.1016/0022-5193(64)90038-4", "Hamilton1964", dry_run = FALSE)

## End(Not run)
```

`cz_qmd_keys` *Extract pandoc citation keys used in a qmd/Rmd file*

Description

Strips fenced code blocks first (so `@` in R code is never mistaken for a citation), finds `@key` tokens, then drops Quarto cross-references (`@fig-x`, `@tbl-x`, `@eq-x`, ...), which use the same syntax but are not bibliography citations.

Usage

```
cz_qmd_keys(
  path,
 crossref_prefixes = c("fig", "tbl", "eq", "sec", "lst", "thm", "lem", "cor", "prp",
    "cnj", "def", "exm", "exr", "rem", "sol", "nte")
)
```

Arguments

`path`	Path to a .qmd or .Rmd file.
`crossref_prefixes`	Cross-reference prefixes to drop (without the trailing hyphen).

Value

Sorted character vector of unique citation keys (no leading `@`).

Examples

```
qmd <- tempfile(fileext = ".qmd")
writeLines(c(
  "Species evolve by natural selection [@Darwin1859]; see @fig-map.",
  "```{r}",
  "x@slot  # not a citation",
  "```"
), qmd)
cz_qmd_keys(qmd)
```

cz_set_auth *Save Zotero credentials to* .Renviron

Description

Prompts for your Zotero user (or group) ID and API key and saves them as ZOTERO_LIBRARY_ID and ZOTERO_API_KEY in .Renviron, so you do not have to edit that file by hand. They also take effect in the current R session right away.

Usage

```
cz_set_auth(scope = c("user", "project"))
```

Arguments

scope	"user" (default) or "project"; see Scope.

Details

The key is deliberately not an argument: values typed as arguments end up in your command history and scripts. Instead, you are prompted. This only works in an interactive session. Note that the prompt is not masked, so the key is visible on screen while you paste it. The key is never printed back. If either variable is already set in the file, you are asked before it is replaced; other lines are left alone.

Value

The path of the file written, invisibly.

Scope

The default, "user", writes to your user-level ~/.Renviron, which is available in every project and cannot be committed by accident (use R_ENVIRON_USER to point R at a different file).

"project" is for advanced use, such as a different key per project. It writes .Renviron in the current working directory, and refuses unless that file is listed in .gitignore. R reads only one .Renviron, so a project file *replaces* (it is not merged with) your user-level one in that project, hiding any other variables defined there (for example GITHUB_PAT). Restart R after saving. citezot does not write .Rprofile, which is code, not configuration.

See Also

cz_auth() to check the saved credentials.

Examples

```
# Interactive only: prompts for the ID and key.
cz_set_auth()
```

`cz_suggest_key` *Suggest a citekey for an item*

Description

Builds a key from a Better BibTeX-style formula, e.g. `"auth.lower + year + shorttitle(1).lower"`. If the key is already taken, a letter (`a`, `b`, ...) is appended, as Better BibTeX does.

Usage

```
cz_suggest_key(item, formula = NULL, existing = character())
```

Arguments

`item`	A Zotero native item (as from `cz_get_item()`, or the payload `cz_add()` builds) or a CSL-JSON item.
`formula`	The key formula. See Details.
`existing`	Keys already in use, so the result can be made unique.

Details

The formula is, in order of preference, the `formula` argument, then the one found by `cz_key_format()` (the `citezot.key_format` option, your Zotero profile's Better BibTeX pattern, or Better BibTeX's default `auth.lower + shorttitle(3,3) + year`). Supported functions are `auth` (first author's last name), `year`, `shorttitle(n, m)` (first `n` words of the title, skipping short function words, with the first `m` capitalised), and `title`; filters are `lower`, `upper`, `fold`, `transliterate`, `clean`, `condense` and `nopunct`; terms are joined with `+` and quoted text is kept as is. Anything else is an error rather than a silently different key. The result is ASCII and safe to cite in pandoc.

Value

A single string.

Examples

```
item <- list(
  itemType = "journalArticle",
  title = "A taxonomic revision of Hymenophyllaceae",
  date = "2006",
  creators = list(
    list(creatorType = "author", firstName = "A", lastName = "Ebihara")
  )
)
cz_suggest_key(item, "auth.lower + year + shorttitle(1).lower")
cz_suggest_key(item,
  existing = "ebihara2006taxonomic",
  formula = "auth.lower + year + shorttitle(1).lower"
)
```

`cz_update_bib` *Write a bibliography file from the references cited in your documents*

Description

Scans Quarto / R Markdown documents for `@key` citations, looks the keys up among the pinned citekeys in your Zotero library, and writes a bibliography file containing only the cited items. Zotero is the source of truth for the content of each reference; the documents decide which references are needed.

Usage

```
cz_update_bib(
  qmd = NULL,
  out = NULL,
  format = NULL,
  bare = NULL,
  allow_incomplete = FALSE,
  auth = cz_auth(),
  cache = getOption("citezot.cache", FALSE),
  quiet = FALSE
)
```

Arguments

`qmd`	Documents to scan for `@key` citations. Defaults to every `.qmd` and `.Rmd` file under the working directory, skipping hidden and underscore-prefixed directories, `renv`, `packrat` and `node_modules`.
`out`	Bibliography file to write. Defaults to the file named by `bibliography:` in the documents' YAML front matter (or the project's `_quarto.yml`), else `references.bib`. If the documents name different files, `out` must be given.
`format`	`"bib"` or `"yaml"`; inferred from `out` if `NULL`.
`bare`	Optional files (e.g. CSVs) scanned for bare citekey-shaped tokens such as `Ebihara2006`. Missing ones only warn.
`allow_incomplete`	Write the file even if keys cited in `qmd` are not found in Zotero.
`auth`	Credentials from `cz_auth()`.
`cache`	Passed to `cz_fetch_items()`: keep a local copy of the library so repeat runs are fast. Defaults to `getOption("citezot.cache", FALSE)`.
`quiet`	Suppress the summary message.

Details

Safety: if a key cited in a document is not found in Zotero, the function stops and leaves the existing output file untouched (a bibliography that silently loses entries breaks the render). Set `allow_incomplete = TRUE` to write what was found anyway. Keys found only in `bare` files, and duplicate citekeys in Zotero (the first match is used), give warnings. The file is only rewritten when its content would change, so unchanged bibliographies keep their modification time.

Value

`out`, invisibly (handy for `targets::tar_target(format = "file")`).

Output formats

`"bib"` (default) writes BibLaTeX using Zotero's own exporter, with each entry keyed by its pinned citekey; it works with every Quarto output format, including Typst. `"yaml"` writes CSL-YAML (a `references:` list with `id` and `citation-key` set to the citekey). The format is inferred from the extension of `out` (`.bib` or `.yaml`/`.yml`).

Examples

```
## Not run:
# reads ZOTERO_LIBRARY_ID / ZOTERO_API_KEY from the environment
cz_update_bib("manuscript.qmd")
cz_update_bib("manuscript.qmd", out = "references.yaml")

## End(Not run)
```

---

cz_update_item *Update fields of a Zotero item*

Description

Changes selected fields of one item and shows exactly what will change. The default is a dry run: the item is read and a before/after diff is shown, but nothing is written. Use `dry_run = FALSE` to write.

Usage

```
cz_update_item(
  item,
  fields,
  dry_run = TRUE,
  auth = cz_auth(),
  cache = getOption("citezot.cache", FALSE)
)
```

Arguments

item	The item: a DOI, Zotero item ID, citekey, or one-row result of cz_find().
fields	A named list of new values, e.g. list(date = "in press", DOI = "10.1234/abc"). Names must be fields that exist for the item's type; see names(cz_get_item(item)$data).
dry_run	If TRUE (default), show the diff and write nothing.
auth	Credentials from cz_auth() (a key with write access to write).
cache	Passed to cz_fetch_items() when item has to be looked up.

Details

Only fields whose value actually changes are sent (a PATCH), together with If-Unmodified-Since-Version: if the item was edited elsewhere since it was read, Zotero rejects the write (HTTP 412) and nothing is overwritten.

Plain fields take a single string (use "" to clear one). creators, tags, collections and relations take lists and are **replaced as a whole**, so to add one tag pass the full list of tags. Identity and bookkeeping fields (key, version, itemType, dates, parentItem, deleted) can't be changed, and the citekey is changed with cz_pin_key().

Value

Invisibly, a list with item, changes (for each changed field, its before and after) and written.

Examples

```
## Not run:
cz_update_item("Regnier2024", list(date = "in press")) # preview
cz_update_item("Regnier2024", list(date = "in press"), dry_run = FALSE)

## End(Not run)
```

Index

cz_add, 2
cz_add(), *13*
cz_auth, 3
cz_auth(), *3*, *5–8*, *10*, *12*, *14*, *16*
cz_bare_keys, 4
cz_clear_cache, 5
cz_clear_cache(), *6*
cz_fetch_items, 5
cz_fetch_items(), *3–5*, *7, 8*, *10*, *14*, *16*
cz_find, 7
cz_find(), *3*, *8*, *10*, *16*
cz_get_item, 8
cz_get_item(), *13*
cz_item_key, 8
cz_item_key(), *10*
cz_key_format, 9
cz_key_format(), *2*, *13*
cz_pin_key, 10
cz_pin_key(), *7, 8*, *16*
cz_qmd_keys, 11
cz_set_auth, 12
cz_set_auth(), *3*
cz_suggest_key, 13
cz_suggest_key(), *2*
cz_update_bib, 14
cz_update_item, 15