

# Package ‘citationchaser’

July 22, 2025

**Title** Perform Forward and Backwards Chasing in Evidence Syntheses

**Version** 0.0.4

**Description** In searching for research articles, we often want to obtain lists of references from across studies, and also obtain lists of articles that cite a particular study. In systematic reviews, this supplementary search technique is known as ‘citation chasing’: forward citation chasing looks for all records citing one or more articles of known relevance; backward citation chasing looks for all records referenced in one or more articles. Traditionally, this process would be done manually, and the resulting records would need to be checked one-by-one against included studies in a review to identify potentially relevant records that should be included in a review. This package contains functions to automate this process by making use of the Lens.org API. An input article list can be used to return a list of all referenced records, and/or all citing records in the Lens.org database (consisting of PubMed, PubMed Central, CrossRef, Microsoft Academic Graph and CORE; <<https://www.lens.org>>).

**Imports** dplyr, httr, jsonlite, maditr, MESS, networkD3, scales, tibble, utils, data.table

**Suggests** knitr, rmarkdown

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Depends** R (>= 2.10)

**NeedsCompilation** no

**Author** Neal Haddaway [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-3902-2234>>),  
Matthew Grainger [ctb],  
Charles Gray [ctb]

**Maintainer** Neal Haddaway <nealhaddaway@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-01-27 10:40:02 UTC

## Contents

getLENSData . . . . .	2
get_citation . . . . .	2
get_refs . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

getLENSData	<i>Function to query Lens.org</i>
-------------	-----------------------------------

---

### Description

Function written by lens.org for use of their API.

### Usage

```
getLENSData(token, query)
```

### Arguments

token	An access key for the lens.org API. Tokens can be obtained by applying for scholarly API access and creating a token once approved. See ' <a href="https://www.lens.org/lens/user/subscription">https://www.lens.org/lens/user/subscription</a> ' for further details.
query	A search string formulated according to the Lens.org API documentation: ' <a href="https://docs.api.lens.org/request-scholar.html">https://docs.api.lens.org/request-scholar.html</a> '.

### Value

A summary response. The results are viewable using '`content(data, "text")`'. Other details regarding the request (e.g. response times) can be accessed through the main output.

---

get_citation	<i>Find citation based on identifier</i>
--------------	--

---

### Description

This function takes a list of articles in the form of established identifiers (e.g. digital object identifiers) and sends a request to the lens.org API to obtain full citation information from the Lens database for all sought articles.

### Usage

```
get_citation(
  article_list,
  type = "doi",
  token = "WCF1pCtuJXYI1sDhZcZ8y7hHpri0SEmTnLNkeU40EM5JTQRNXB9w"
)
```

**Arguments**

article_list	List of article identifiers for which the reference lists will be returned. Must be a list/vector of identifiers, e.g. "'10.1186/s13750-018-0126-2' '10.1002/jrsm.1378'".
type	Specification of the type of input provided. The default is 'doi' (digital object identifier), but any of the following are accepted: "pmid" (PubMed ID), "pmcid" (PubMed Central ID), "magid" (Microsoft Academic ID), "coreid" (CORE identifier), "lens_id" (The Lens.org ID), "title" (article title; much lower specificity).
token	An access key for the lens.org API. Tokens can be obtained by applying for scholarly API access and creating a token once approved. See <a href="https://www.lens.org/lens/user/subscription">https://www.lens.org/lens/user/subscription</a> for further details.

**Value**

A dataframe containing the matching citation from Lens.org.

**Examples**

```
## Not run:
article_list <- c("10.1007/978-3-642-37048-9_13", "10.1111/sum.12030",
"10.5194/bg-13-3619-2016", "10.1016/j.agee.2012.09.006")
results <- get_citation(article_list)
articles <- results$display

## End(Not run)
```

---

get\_refs

*Automated citation chasing in systematic reviews*


---

**Description**

This function takes a list of articles in the form of established identifiers (e.g. digital object identifiers) and sends a request to the lens.org API to firstly identify all cited references in all articles (in the form of lists of lens IDs), and then query these lens IDs to bring back full citation information for all listed records. Deduplicates references to the same records across articles, resulting in an RIS file and a summary report in the console.

**Usage**

```
get_refs(article_list, type = "doi", get_records, save_object = FALSE, token)
```

**Arguments**

article_list	List of article identifiers for which the reference lists will be returned. Must be a list/vector of identifiers, e.g. "'10.1186/s13750-018-0126-2' '10.1002/jrsm.1378'".
--------------	---

type	Specification of the type of input provided. The default is 'doi' (digital object identifier), but any of the following are accepted: "pmid" (PubMed ID), "pmcid" (PubMed Central ID), "magid" (Microsoft Academic ID), "coreid" (CORE identifier), "lens_id" (The Lens.org ID), "title" (article title; much lower specificity).
get_records	Specification of whether to look for records referenced within the input articles ('references'), records citing the input articles ('citations'), or both ('both').
save_object	Option to save the resultant ris file as an object in the Global Environment. The default is FALSE.
token	An access key for the lens.org API. Tokens can be obtained by applying for scholarly API access and creating a token once approved. See <a href="https://www.lens.org/lens/user/subscription">https://www.lens.org/lens/user/subscription</a> for further details.

### Value

An RIS file is saved to the working directory. A report is printed to the console. If 'save\_object=TRUE', the RIS file is returned as an object

### Examples

```
## Not run:
article_list <- c("10.1007/978-3-642-37048-9_13",
                 "10.1111/sum.12030",
                 "10.5194/bg-13-3619-2016",
                 "10.1016/j.agee.2012.09.006")
token <- 'token'
refs <- get_refs(article_list, get_records = 'references', token = token)
refs

## End(Not run)
```

# Index

`get_citation`, 2  
`get_refs`, 3  
`getLENSData`, 2