

Package: citecorp (via r-universe)

July 31, 2026

Type Package

Title Client for the Open Citations Corpus

Description Client for the Open Citations Corpus
(<http://opencitations.net/>). Includes a set of functions for
getting one identifier type from another, as well as getting
references and citations for a given identifier.

Version 0.3.2.9000

License MIT + file LICENSE

URL <https://github.com/ropensci/citecorp> (devel),
<https://docs.ropensci.org/citecorp/> (docs)

BugReports <https://github.com/ropensci/citecorp/issues>

Roxygen list(markdown = TRUE)

Encoding UTF-8

Language en-US

LazyData yes

Imports crul (>= 0.7.0), data.table, fauxpas (>= 0.5.0), jsonlite

Suggests testthat, vcr, webmockr

RoxygenNote 7.3.2

X-schema.org-applicationCategory Literature

X-schema.org-keywords doi, metadata, citation, opencitations, bibtex

X-schema.org-isPartOf <https://ropensci.org>

Config/pak/sysreqs libssl-dev

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

Date/Publication 2025-02-04 14:23:02 UTC

RemoteUrl <https://github.com/ropensci/citecorp>

RemoteRef master

RemoteSha c654d22917b4f570ca8725ade9ff93d81a914f12

Contents

citecorp-package	2
oc_coci	2
oc_dois	4
oc_lookup	4
oc_pmcids	5
oc_pmids	5
Index	6

citecorp-package	<i>citecorp</i>
------------------	-----------------

Description

Client for the Open Citations Corpus <http://opencitations.net/>

Author(s)

Scott Chamberlain <myrmecocystus@gmail.com>
David Selby <david_antony.selby@dfki.de>

See Also

- Useful links:
- [https://github.com/ropensci/citecorp\(devel\)](https://github.com/ropensci/citecorp(devel))
 - [https://docs.ropensci.org/citecorp/\(docs\)](https://docs.ropensci.org/citecorp/(docs))
 - Report bugs at <https://github.com/ropensci/citecorp/issues>

oc_coci	<i>COCI: OpenCitations Index of Crossref open DOI-to-DOI references</i>
---------	---

Description

AFAICT this API is a REST wrapper around the SPARQL service

Usage

```
oc_coci_refs(doi, exclude = NULL, filter = NULL, sort = NULL, ...)
oc_coci_cites(doi, exclude = NULL, filter = NULL, sort = NULL, ...)
oc_coci_meta(doi, exclude = NULL, filter = NULL, sort = NULL, ...)
oc_coci_citation(oci, ...)
```

Arguments

doi	(character) one or more Digital Object Identifiers (DOIs)
exclude	(character) a field_name; all the rows that have an empty value in the field_name specified are removed from the result set
filter	=<field_name>:<operator><value>: only the rows compliant with <value> are kept in the result set. The parameter <operation> is not mandatory. If <operation> is not specified, <value> is interpreted as a regular expression, otherwise it is compared by means of the specified operation. Possible operators are "=", "<", and ">". For instance, filter=title:semantics? returns all the rows that contain the string "semantic" or "semantics" in the field title, while filter=date:>2016-05 returns all the rows that have a date greater than May 2016.
sort	=<order>(<field_name>): sort in ascending (<order> set to "asc") or descending (<order> set to "desc") order the rows in the result set according to the values in <field_name>. For instance, sort=desc(date) sorts all the rows according to the value specified in the field date in descending order.
...	curl options passed on to crul::verb-GET
oci	(character) one or more Open Citation Identifiers (OCIs)

Value

data.frame, see <http://opencitations.net/index/coci/api/v1> for explanation of the resulting columns

References

<http://opencitations.net/index/coci/api/v1>, <https://github.com/opencitations/api-coci>

Examples

```
doi1 <- "10.1108/jd-12-2013-0166"
doi2 <- "10.1371/journal.pgen.1005937"
oci1 <-
  "02001010806360107050663080702026306630509-0200101080636102704000806"
oci2 <-
  "0200101000836191363010263020001036300010606-020010003083604090301050910"

if (
  crul::ok(
    "http://opencitations.net/index/coci/api/v1/references/10.1108/jd-12-2013-0166",
    timeout_ms = 1000L)
  ) {
  try(
    oc_coci_cites(doi1),
    silent = TRUE
  )
}

### More examples
## Not run:
```

```

# references
oc_coci_refs(doi1, exclude = "oci")
oc_coci_refs(doi1, filter = "date:>2016-05", verbose = TRUE)
oc_coci_refs(doi2)
oc_coci_refs(c(doi1, doi2))

# citations
oc_coci_cites(doi1, exclude = "oci")
oc_coci_cites(doi2)
oc_coci_cites(c(doi1, doi2))

# metadata
oc_coci_meta(doi2)
oc_coci_meta(c(doi1, doi2))

# citation - an OCI instead of a DOI
oc_coci_citation(oci1)
oc_coci_citation(c(oci1, oci2))

## End(Not run)

```

oc_dois	<i>Vector of 25 DOIs (Digital Object Identifiers)</i>
---------	---

Description

To be used in examples, etc. for [oc_lookup](#) and [oc_coci](#) methods

oc_lookup	<i>Methods for getting IDs from other IDs</i>
-----------	---

Description

Methods for getting IDs from other IDs

Usage

```

oc_doi2ids(id, ...)

oc_pmid2ids(id, ...)

oc_pmcid2ids(id, ...)

```

Arguments

id	One or more digital object identifiers (DOI), PMID, or PMCID, depending on the function
...	curl options passed on to crul::verb-GET

Value

data.frame, with four columns:

- doi: digital object identifier
- pmid: pubmed identifier
- pmcid: pubmed central identifier
- paper: open citations corpus url

An empty data.frame (no columns or rows) when no results found

Column order will always be the same; note though that some columns may be missing if, for example, there's no PMID for a DOI search.

Examples

```
if (oc_lookup_check()) {
  try(
    oc_doi2ids("10.1097/igc.0000000000000609", timeout_ms=10),
    silent = TRUE
  )
}

### More examples
## Not run:
oc_doi2ids('10.1093/biomet/80.3.527')
oc_doi2ids('10.1093/biomet/79.3.531')
oc_pmid2ids("31857888")
oc_pmcid2ids("PMC6422012")

oc_doi2ids(id = oc_dois[1:3])
oc_pmid2ids(id = oc_pmids[1:3])
oc_pmcid2ids(id = oc_pmcids[1:3])

## End(Not run)
```

oc_pmcids	<i>Vector of 8 PMCIDs (PubMed Central Identifiers)</i>
-----------	--

Description

To be used in examples, etc. for [oc_lookup](#) and [oc_coci](#) methods

oc_pmids	<i>Vector of 24 PMIDs (PubMed Identifiers)</i>
----------	--

Description

To be used in examples, etc. for [oc_lookup](#) and [oc_coci](#) methods

Index

* **data**

- oc_dois, [4](#)
- oc_pmcids, [5](#)
- oc_pmids, [5](#)

* **package**

- citecorp-package, [2](#)

citecorp (citecorp-package), [2](#)

citecorp-package, [2](#)

crul::verb-GET, [3](#), [4](#)

oc_coci, [2](#), [4](#), [5](#)

oc_coci_citation (oc_coci), [2](#)

oc_coci_cites (oc_coci), [2](#)

oc_coci_meta (oc_coci), [2](#)

oc_coci_refs (oc_coci), [2](#)

oc_doi2ids (oc_lookup), [4](#)

oc_dois, [4](#)

oc_lookup, [4](#), [4](#), [5](#)

oc_pmcid2ids (oc_lookup), [4](#)

oc_pmcids, [5](#)

oc_pmid2ids (oc_lookup), [4](#)

oc_pmids, [5](#)