Package ‘crevents’

January 22, 2020

Title  Client for the ‘Crossref’ Events API

Description  Client for the ‘Crossref’ Events API
             <https://www.eventdata.crossref.org/guide/>. The ‘Crossref’
             Events service captures discussion of scholarly works (e.g.,
             peer-reviewed publications) in non-traditional places such
             as blogs and social media.

Version  0.1.0

License  MIT + file LICENSE

URL  https://docs.ropensci.org/crevents,
    https://github.com/ropensci/crevents

BugReports  https://github.com/ropensci/crevents/issues

LazyData  true

Encoding  UTF-8

Language  en-US

Imports  crul, jsonlite, tibble

Suggests  testthat, vcr

RoxygenNote  7.0.2

X-schema.org-applicationCategory  Literature

X-schema.org-keywords  text-ming, literature, scholarly, publications,
                       citations, events, Crossref, tweets

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

NeedsCompilation  no

Author  Scott Chamberlain [aut, cre] (<https://orcid.org/0000-0003-1444-9135>)

Maintainer  Scott Chamberlain <myrmecocystus@gmail.com>

Repository  CRAN

Date/Publication  2020-01-22 14:50:03 UTC
R topics documented:

crevents-package ................................................. 2
crev_deleted ..................................................... 2
crev_edited ...................................................... 5
crev_evrecs ...................................................... 8
crev_query ....................................................... 9
filters ............................................................. 12

Index 13

crevents-package  Client for the Crossref Events API

Description


Author(s)

Scott Chamberlain <sckott@protonmail.com>

crev_deleted  Query Crossref Deleted Events data

Description

Query Crossref Deleted Events data

Usage

crev_deleted(
  rows = 1000L,
  cursor = NULL,
  source = NULL,
  from_updated_date = NULL,
  until_updated_date = NULL,
  from_collected_date = NULL,
  until_collected_date = NULL,
  from_occurred_date = NULL,
  until_occurred_date = NULL,
  obj_id = NULL,
  obj_id_prefix = NULL,
  subj_id = NULL,
  subj_id_prefix = NULL,
)
crev_deleted

subj_id_domain = NULL,
obj_id_domain = NULL,
subj_url = NULL,
obj_url = NULL,
subj_url_domain = NULL,
obj_url_domain = NULL,
subj_alternative_id = NULL,
obj_alternative_id = NULL,
relation = NULL,
facet = NULL,
...
)

Arguments

rows (character) the number of Events you want to retrieve per page. The default, and recommended, value is 10,000, which allows you to retrieve large numbers of Events quickly. There are typically between 10,000 and 100,000 Events collected per day. Default: 1000

cursor (character) allows you to iterate through a search result set.

source source ID

from_updated_date (character) from or until updated date, as YYYY-MM-DD

until_updated_date (character) from or until updated date, as YYYY-MM-DD

from_collected_date (character) from or until updated date, as YYYY-MM-DD

until_collected_date (character) from or until updated date, as YYYY-MM-DD

from_occurred_date (character) from or until occurred date, as YYYY-MM-DD

until_occurred_date (character) from or until occurred date, as YYYY-MM-DD

obj_id quoted URL or a DOI

obj_id_prefix DOI prefix like 10.5555

subj_id quoted URL or a DOI

subj_id_prefix DOI prefix like 10.5555

subj_id_domain domain of the subj_id e.g. en.wikipedia.org

obj_id_domain domain of the obj_url e.g. en.wikipedia.org

subj_url quoted full URL

obj_url quoted full URL

subj_url_domain domain of the optional subj_url, if present e.g. en.wikipedia.org

obj_url_domain domain of the optional obj_url, if present e.g. en.wikipedia.org
subj_alternative_id
optional subj.alternative-id

obj_alternative_id
optional obj.alternative-id

relation
relation type ID

facet
a facet value, can be just the facet name, or facet name plus the limit (* or an integer)

... 
Curl options passed on to HttpClient. Note that for now, when doing async you can't pass on curl options - will try to add later.

facets
Remember that these totals don't refer to unique links necessarily, so you should be cautious about using the numbers for anything other than exploration.

• source: source ID
• relation-type: relation type ID
• obj-id.prefix: DOI prefix like 10.5555, if Object is a DOI
• subj-id.prefix: DOI prefix like 10.5555, if Subject is a DOI
• subj-id.domain: Domain of the subj_id URL
• obj-id.domain: Domain of the obj_id URL
• subj.url.domain: Domain of the subj.url URL. This may or may not be the same as the subj_id.
• obj.url.domain: Domain of the obj.url URL. This may or may not be the same as the obj_id.

Examples

if (crl::ok("http://api.eventdata.crossref.org/v1/events?rows=0")) {
  (res <- crev_deleted(rows = 3))
  res$status
  res$message$type'
  res$message$'total-results'
  res$message$'items-per-page'
  res$message$events
  res$message$events$obj_id
}

## Not run:
# filter with source
crev_deleted(rows = 3, source = "reddit")
crev_deleted(rows = 3, source = "twitter")

# on the first of March 2017s
crev_deleted(rows=3, from_collected_date = "2017-03-01",
until_collected_date = "2017-03-01")

# in the month of March 2017: occurred
crev_deleted(rows=3, from_occurred_date = "2017-03-01",
until_occurred_date = "2017-03-31")
# in the month of March 2017: collected
crev_deleted(rows=3, from_collected_date = "2017-03-01",
    until_collected_date = "2017-03-31")

# Up to ten Events for a DOI prefix
crev_deleted(rows = 3, obj_id_prefix = "10.1186")

# Up to ten events for a given DOI
crev_deleted(rows = 3, obj_id = "10.1186/s40536-017-0036-8")

# Use cursor
res <- crev_deleted(rows = 3)
crev_deleted(rows = 3, cursor = res$message$'next-cursor')

# facet
crev_deleted(rows = 0, source = 'reddit', facet = "subj-id.domain:10")
crev_deleted(rows = 0, source = 'reddit', facet = "subj-id.domain:10")
crev_deleted(rows = 0, source = 'newsfeed',
    subj_id_domain = "www.theguardian.com", facet = "obj-id.prefix:*")

## End(Not run)

---

**crevEdited**

Query Crossref Edited Events data

**Description**

Query Crossref Edited Events data

**Usage**

crevEdited(
    rows = 1000L,
    cursor = NULL,
    source = NULL,
    from_updated_date = NULL,
    until_updated_date = NULL,
    from_collected_date = NULL,
    until_collected_date = NULL,
    from_occurred_date = NULL,
    until_occurred_date = NULL,
    obj_id = NULL,
    obj_id_prefix = NULL,
    subj_id = NULL,
    subj_id_prefix = NULL,
    subj_id_domain = NULL,
    obj_id_domain = NULL,
    subj_url = NULL,
obj_url = NULL,
subj_url_domain = NULL,
obj_url_domain = NULL,
subj_alternative_id = NULL,
obj_alternative_id = NULL,
relation = NULL,
facet = NULL,
...)

Arguments

rows (character) the number of Events you want to retrieve per page. The default, and recommended, value is 10,000, which allows you to retrieve large numbers of Events quickly. There are typically between 10,000 and 100,000 Events collected per day. Default: 1000

cursor (character) allows you to iterate through a search result set.

source source ID

from_updated_date (character) from or until updated date, as YYYY-MM-DD

until_updated_date (character) from or until updated date, as YYYY-MM-DD

from_collected_date (character) from or until updated date, as YYYY-MM-DD

until_collected_date (character) from or until updated date, as YYYY-MM-DD

from_occurred_date (character) from or until occurred date, as YYYY-MM-DD

until_occurred_date (character) from or until occurred date, as YYYY-MM-DD

obj_id quoted URL or a DOI

obj_id_prefix DOI prefix like 10.5555

subj_id quoted URL or a DOI

subj_id_prefix DOI prefix like 10.5555

subj_id_domain domain of the subj_id e.g. en.wikipedia.org

obj_id_domain domain of the obj_url e.g. en.wikipedia.org

subj_url quoted full URL

obj_url quoted full URL

subj_url_domain domain of the optional subj.url, if present e.g. en.wikipedia.org

obj_url_domain domain of the optional obj.url, if present e.g. en.wikipedia.org

subj_alternative_id optional subj.alternative-id
obj_alternative_id
  optional obj.alternative-id
relation
  relation type ID
facet
  a facet value, can be just the facet name, or facet name plus the limit (* or an integer)
...  Curl options passed on to HttpClient. Note that for now, when doing async
you can’t pass on curl options - will try to add later.

facets

Remember that these totals don’t refer to unique links necessarily, so you should be cautious about using the numbers for anything other than exploration.

- source: source ID
- relation-type: relation type ID
- obj-id.prefix: DOI prefix like 10.5555, if Object is a DOI
- subj-id.prefix: DOI prefix like 10.5555, if Subject is a DOI
- subj-id.domain: Domain of the subj_id URL
- obj-id.domain: Domain of the obj_id URL
- subj.url.domain: Domain of the subj.url URL. This may or may not be the same as the subj_id.
- obj.url.domain: Domain of the obj.url URL. This may or may not be the same as the obj_id.

Examples

```r
if (crul::ok("http://api.eventdata.crossref.org/v1/events?rows=0")) {
  res <- crev_edited(rows = 3)
  res$status
  res$message
  res$message$total-results
  res$message$items-per-page
  res$message$events
}
```

## Not run:
# filter with source
crev_edited(rows = 10, source = "reddit")  
crev_edited(rows = 10, source = "twitter")

# on the first of March 2017s
crev_edited(rows=10, from_collected_date = "2017-03-01",  
until_collected_date = "2017-03-01")

# in the month of March 2017
crev_edited(rows=10, from_occurred_date = "2017-03-01",  
until_occurred_date = "2017-03-31")

# in the month of March 2017
crev_edited(rows=10, from_collected_date = "2017-03-01",  
until_collected_date = "2017-03-31")
crev_evrecs

until_collected_date = "2017-03-31")

# Up to ten Events for a DOI prefix
crev_edited(rows = 10, obj_id_prefix = "10.1186")

# Up to ten events for a given DOI
crev_edited(rows = 10, obj_id = "10.1186/s40536-017-0036-8")

# Use cursor
res <- crev_editor(rows = 10)
crev_edited(rows = 10, cursor = res$message$`next-cursor`)

# facet
crev_edited(rows = 0, source = 'reddit', facet = "subj-id.domain:10")
crev_edited(rows = 0, source = 'reddit', facet = "subj-id.domain:10")
crev_edited(rows = 0, source = 'newsfeed',
sub_id_domain = "www.theguardian.com", facet = "obj-id.prefix:*")

## End(Not run)

crev_evrecs Parse Crossref Events evidence records

description
Parse Crossref Events evidence records

usage
crev_evrecs(x, ...)

arguments
x (character) one or more urls for evidence records

... curl options passed through to crul::verb-GET

details
this function doesn’t parse results other than converting JSON to an R list, because the data is deeply
nested, so coercing to data.frame’s is an exercise left to the user

From crev_query() you will get urls like "https://evidence.eventdata.crossref.org/evidence" in the
$message$events$evidence_record slot of results. These links hold JSON data with the evidence
record details for the record. This function simply retrieves that JSON and parses to a list

value
an unnamed list of results, one for each URL passed in.
**crev_query**

Query Crossref Events data

### Description

Query Crossref Events data

### Usage

```r
crev_query(
  rows = 1000L,
  cursor = NULL,
  source = NULL,
  from_updated_date = NULL,
  until_updated_date = NULL,
  from_collected_date = NULL,
  until_collected_date = NULL,
  from_occurred_date = NULL,
  until_occurred_date = NULL,
  obj_id = NULL,
  obj_id_prefix = NULL,
  subj_id = NULL,
  subj_id_prefix = NULL,
  subj_id_domain = NULL,
  obj_id_domain = NULL,
  subj_url = NULL,
  subj_url_domain = NULL,
  obj_url = NULL,
  obj_url_domain = NULL,
  subj_alternative_id = NULL,
  obj_alternative_id = NULL,
  relation = NULL,
  facet = NULL,
  ...
)
```

### Examples

```r
## Not run:
res <- crev_query(rows = 4)
recs <- res$message$events$evidence_record
crev_evrecs(recs[[1]])
crev_evrecs(recs)

## End(Not run)
```
Arguments

rows (character) the number of Events you want to retrieve per page. The default, and recommended, value is 10,000, which allows you to retrieve large numbers of Events quickly. There are typically between 10,000 and 100,000 Events collected per day. Default: 1000

cursor (character) allows you to iterate through a search result set.

source (character) source ID

from_updated_date, until_updated_date (character) from or until updated date, as YYYY-MM-DD

from_collected_date, until_collected_date (character) from or until updated date, as YYYY-MM-DD

from_occurred_date, until_occurred_date (character) from or until occurred date, as YYYY-MM-DD

obj_id quoted URL or a DOI

obj_id_prefix DOI prefix like 10.5555

subj_id quoted URL or a DOI

subj_id_prefix DOI prefix like 10.5555

subj_id_domain domain of the subj_id e.g. en.wikipedia.org

obj_id_domain domain of the obj_url e.g. en.wikipedia.org

subj_url quoted full URL

obj_url quoted full URL

subj_url_domain domain of the optional subj.url, if present e.g. en.wikipedia.org

obj_url_domain domain of the optional obj.url, if present e.g. en.wikipedia.org

subj_alternative_id optional subj.alternative-id

obj_alternative_id optional obj.alternative-id

relation relation type ID

facet a facet value, can be just the facet name, or facet name plus the limit (* or an integer)

... Curl options passed on to HttpClient. Note that for now, when doing async you can’t pass on curl options - will try to add later.

facets

Remember that these totals don’t refer to unique links necessarily, so you should be cautious about using the numbers for anything other than exploration.

- source: source ID
- relation-type: relation type ID
- obj-id.prefix: DOI prefix like 10.5555, if Object is a DOI
• subj-id.prefix: DOI prefix like 10.5555, if Subject is a DOI
• subj-id.domain: Domain of the subj_id URL
• obj-id.domain: Domain of the obj_id URL
• subj.url.domain: Domain of the subj.url URL. This may or may not be the same as the subj_id.
• obj.url.domain: Domain of the obj.url URL. This may or may not be the same as the obj_id.

Examples

```r
if (curl::ok("http://api.eventdata.crossref.org/v1/events?rows=0")) {
  (res <- crev_query(rows = 4))
  res$status
  res$message-type`
  res$message$"total-results`
  res$message$"items-per-page``
  res$message$events
}

## Not run:
# filter with source
crev_query(rows = 10, source = "reddit")
crev_query(rows = 10, source = "twitter")

# on the first of March 2017
crev_query(rows=10, from_collected_date = "2017-03-01",
          until_collected_date = "2017-03-01")

# in the month of March 2017
crev_query(rows=10, from_occurred_date = "2017-03-01",
          until_occurred_date = "2017-03-31")

# Up to ten Events for a DOI prefix
crev_query(rows = 10, obj_id_prefix = "10.1186")

# Up to ten events for a given DOI
crev_query(rows = 10, obj_id = "10.1186/s40536-017-0036-8")

# Use cursor
res <- crev_query(rows = 10)
crev_query(rows = 10, cursor = res$message$'next-cursor'

# facet
crev_query(rows = 0, source = 'reddit', facet = "subj-id.domain:10")
crev_query(rows = 0, source = 'reddit', facet = "subj-id.domain:10")
crev_query(rows = 0, source = 'newsfeed',
  subj_id_domain = "www.theguardian.com", facet = "obj-id.prefix:*")

## End(Not run)```
Description

Filters information

Details

The filters:

- from-occurred-date - as YYYY-MM-DD
- until-occurred-date - as YYYY-MM-DD
- from-collected-date - as YYYY-MM-DD
- until-collected-date - as YYYY-MM-DD
- subj-id - quoted URL or a DOI
- obj-id - quoted URL or a DOI
- subj-id.prefix - DOI prefix like 10.5555
- obj-id.prefix - DOI prefix like 10.5555
- subj-id.domain - domain of the subj_id e.g. en.wikipedia.org
- obj-id.domain - domain of the obj_url e.g. en.wikipedia.org
- subj.url - quoted full URL
- obj.url - quoted full URL
- subj.url.domain - domain of the optional subj.url, if present e.g. en.wikipedia.org
- obj.url.domain - domain of the optional obj.url, if present e.g. en.wikipedia.org
- subj.alternative-id - optional subj.alternative-id
- obj.alternative-id - optional obj.alternative-id
- relation - relation type ID
- source - source ID
Index

*Topic package
  crevents-package, 2

crev_deleted, 2
crev_edited, 5
crev_evrecs, 8
crev_query, 9
crev_query(), 8
crevents (crevents-package), 2
crevents-package, 2
crul::verb-GET, 8

filters, 12

HttpClient, 4, 7, 10