

A tutorial for OpenCitations and Crossref API

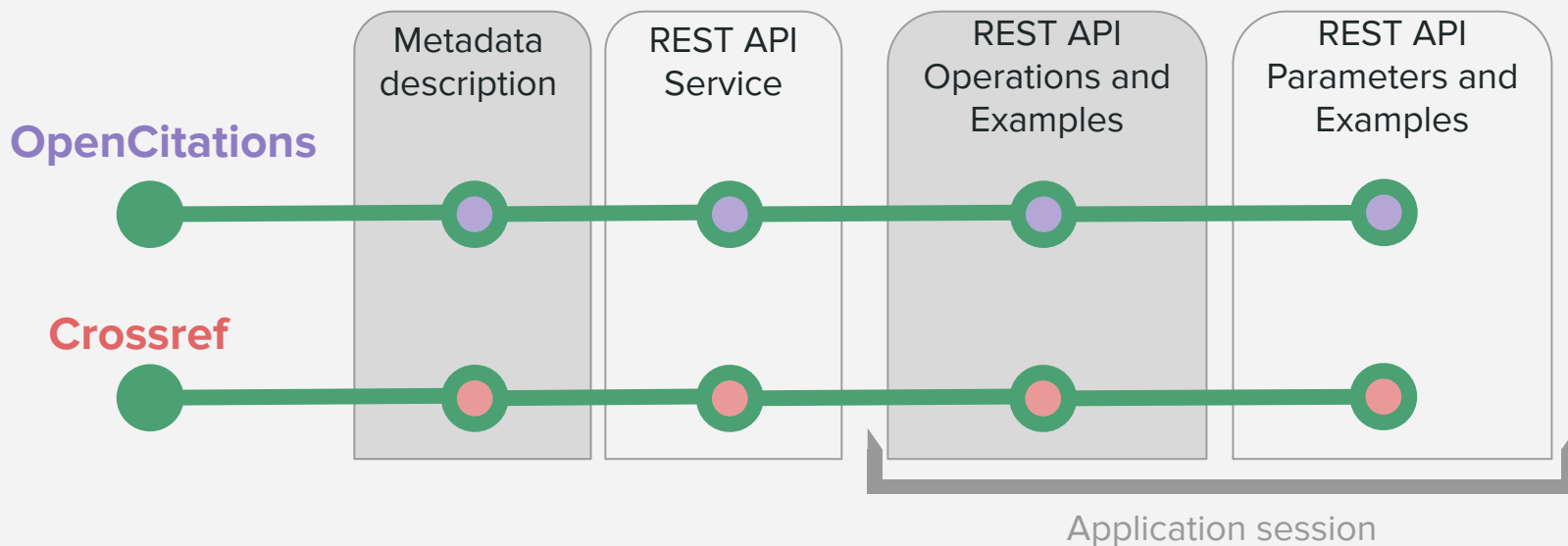
Ivan Heibi

Digital Humanities Advanced Research Centre (DHARC),
Department of Classical Philology and Italian Studies,
University of Bologna, Bologna (Italy)

ivan.heibi2@unibo.it

Introduction

- A tutorial on querying and exploring the bibliographic metadata of two major organizations, **OpenCitations** and **Crossref**, using there **REST API services**.



OpenCitations Metadata

- **The OpenCitations Corpus (OCC)**¹: is an **open repository of Bibliographic metadata** made available under a Creative Commons public domain dedication. OCC currently contains **326743 citing bibliographic resources** and about almost **14 million citations**.
- **The Open Citation Indexes** : some **bibliographic indexes recording citations** between publications using the data available in particular bibliographic databases. The first and more popular index of these is **COCI**, the OpenCitations Index of Crossref open DOI-to-DOI references, a new RDF dataset of around **450 million citations**.

[1] The OpenCitations Corpus (OCC), <http://w3id.org/oc/corpus>

[2] The Open Citation Indexes, <http://w3id.org/oc/index>

OpenCitations REST API

- Implemented by means of [RAMOSE](#), the Restful API Manager Over SPARQL Endpoints.
- Two main REST API services are provided:
 - For the OpenCitations Corpus (OCC):
w3id.org/oc/api/v1
 - For all the OpenCitations Indexes, e.g. COCI, CROCI.
In this tutorial we will focus on COCI: w3id.org/oc/index/coci/api/v1
- REST API Call structure:

<api_url><operation>?<parameter_1>&<parameter_2>...<parameter_n>



OpenCitations REST API: Operations

w3id.org/oc/api/v1

O
C
C

[/metadata/{dois}](#)

This operation allows one to get the metadata of all the articles specified in input by means of their DOIs.

[/coauthorship/{dois}](#)

This operation allows one to get co-authorship matrix of all the articles specified in input by means of their DOIs.

w3id.org/oc/index/coci/api/v1

C
O
C
I

[/references/{doi}](#)

Retrieves the citation data for all the references to other works appearing in the bibliographic entity identified by the input DOI.

[/citations/{doi}](#)

Retrieves the citation data for all the references appearing in other works to the bibliographic entity identified by the input DOI.

[/citation/{oci}](#)

Retrieves the citation data for the citation identified by the input Open Citation Identifier (OCI).

[/metadata/{dois}](#)

This operation retrieves the metadata for all the articles identified by the input DOIs.

OpenCitations REST API: Basic query examples (OCC)

Query format: `<api_url><operation>`

OCC Example (1)

Query

Get the **metadata** for the article identified by
DOI= 10.1016/j.websem.2012.08.001.

<https://w3id.org/oc/api/v1/metadata/10.1016/j.websem.2012.08.001>

Results

```
[
  {
    "title": "FaBiO and CiTO: Ontologies for describing bibliographic
resources and citations",
    "doi": "10.1016/j.websem.2012.08.001",
    "volume": "17",
    "occ_id": "br/2384552",
    "author": "Peroni, Silvio; Shotton, David",
    "occ_reference": "",
    "year": "",
    "doi_reference": "",
    "citation_count": "1",
    "source_title": "Web Semantics: Science, Services and Agents on
the World Wide Web",
    "page": "33-43",
    "issue": ""
  }
]
```

OCC Example (2)

Query

Get the **co-authorship matrix** for all the authors
which have contributed on both the articles
identified by **DOIs= 10.1016/j.websem.2012.08.001,**
10.1108/jd-12-2013-0166.

https://w3id.org/oc/api/v1/coauthorship/10.1108/jd-12-2013-0166_10.1016/j.websem.2012.08.001

Results

```
[
  {
    "coauthorship_count": "1",
    "author1": "Dutton, Alexander",
    "author2": "Gray, Tanya"
  },
  {
    "coauthorship_count": "1",
    "author1": "Dutton, Alexander",
    "author2": "Peroni, Silvio"
  },
  ...
]
```

OpenCitations REST API: Basic query examples (COCI)

Query format: <api_url><operation>

COCI Example (1)

Query

Get the list of **citations** received by the article identified by the **DOI= 10.1002/adfm.201505328**.

<https://w3id.org/oc/index/coci/api/v1/citations/10.1002/adfm.201505328>

Results

```
[
  {
    "author_sc": "no",
    "creation": "2017-02-06",
    "journal_sc": "no",
    "oci": "020...-020...",
    "timespan": "P11M20D",
    "cited": "10.1002/adfm.201505328",
    "citing": "10.1002/jrs.5087"
  },
  ...
]
```

COCI Example (2)

Query

Get the **metadata** for the article identified by the **DOI= 10.1108/jd-12-2013-0166**.

<https://w3id.org/oc/index/coci/api/v1/metadata/10.1108/jd-12-2013-0166>

Results

```
[
  {
    "title": "Setting Our Bibliographic References Free: Towards Open Citation Data",
    "doi": "10.1108/jd-12-2013-0166", ... ,
    "volume": "71",
    "author": "Peroni, Silvio; Dutton, Alexander; Gray, Tanya; Shotton, David",
    "oa_link": "",
    "year": "2015",
    "reference": "10.1001/jama.295.1.90; ... ,
    "citation_count": "13",
    "source_title": "Journal Of Documentation",
    "page": "253-277",
    "issue": "2",
    "source_id": "issn:0022-0418"
  }
]
```

OpenCitations REST API: Parameters

<code>exclude=<field_name></code>	All the rows that have an empty value in the <field_name> specified are removed from the result set
<code>filter=<field_name>:<operator><value></code>	Only the rows compliant with <value> are kept in the result set. E.g. "filter=date:>2016-05" returns all the rows that have a date greater 2016-05.
<code>sort=<order>(<field_name>)</code>	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>
<code>format=<format_type></code>	the final table is returned in the format specified in <format_type> that can be either "csv" or "json" - e.g. format=csv returns the final table in CSV format.
<code>json=<operation_type>("<separator>", <field>, <new_field_1>, <new_field_2>, ...)</code>	<p>in case a JSON format is requested in return, transform each row of the final JSON table according to the rule specified. For instance, considering the JSON table [{ "name": "Doe, John" }, ...], the execution of "json=dict(", ",name,fname,gname)" returns [{ "name": { "fname": "Doe", "gname": "John" }, ...].</p> <p>N.B. This parameter is used also by VOSviewer as a pre-processing parameter to manipulate the input JSON in a different format.</p>

OpenCitations REST API: Parameter query examples

Query format: `<api_url><operation>?<parameter_1>& ...<parameter_n>`

COCI Example (1)

Query

Get the list of **citations** received by the **DOI = 10.1002/adfm.201505328**, which have been **created on 2015 or later**, in a **CSV format**, and **sort** the final results according to their **creation date** in **descending order**.

[https://w3id.org/oc/index/coci/api/v1/citations/10.1002/adfm.201505328?filter=creation:>2018&sort=desc\(creation\)&format=csv](https://w3id.org/oc/index/coci/api/v1/citations/10.1002/adfm.201505328?filter=creation:>2018&sort=desc(creation)&format=csv)

Results

```
oci,citing,cited,creation,timespan,journal_sc,author_sc
02001000002361028181037020001080000050403-020
01000002361013152237020001050005030208,10.1002/a
sia.201800543,10.1002/adfm.201505328,2018-08-23,P2Y6
M6D,no,no
02001000002361217142237020001080001090102-02001
000002361013152237020001050005030208,10.1002/che
m.201801912,10.1002/adfm.201505328,2018-08-16,P2Y5M
30D,no,no
...
```

OCC Example (2)

Query

Get the **metadata** of the document identified by **DOI = 10.1002/adfm.201505328** in **JSON format**, and **redefine** the **“author”** field as an array of all the authors of such document.

[https://w3id.org/oc/api/v1/metadata/10.1016/j.websem.2012.08.001?format=json&json=array\(";",author\)](https://w3id.org/oc/api/v1/metadata/10.1016/j.websem.2012.08.001?format=json&json=array()

Results

```
[
{
  "title": "FaBiO and CITO: Ontologies for describing bibliographic resources and citations",
  "doi": "10.1016/j.websem.2012.08.001",
  "volume": "17",
  "occ_id": "br/2384552",
  "author": [ "Peroni, Silvio", "Shotton, David"],
  "occ_reference": "",
  "year": "",
  "doi_reference": "",
  "citation_count": "1",
  "source_title": "Web Semantics: Science, Services and Agents on the World Wide Web",
  "page": "33-43",
  "issue": ""
}
]
```

Crossref Metadata ¹

- Each content stored/registered with **Crossref includes metadata**, such as:
 - **Bibliographic metadata (such as author, article titles, journal names)**
 - Funding metadata
 - License metadata
 - ... and more
- Crossref collects metadata for many **scholarly items**. The content types are:
 - Journals: includes records for journal articles, as well as supplemental materials.
 - Books: book-level and/or chapter-level records, monographs, series, or sets.
 - Conference Proceedings: information and records about a single conference
 - Datasets
 - Peer Reviews
 - ... and more.

[1] Crossref metadata and content type overview, <https://tinyurl.com/y3knwo3g>

Crossref REST API ¹

- **Exposes the metadata that members provide Crossref**, such as: Bibliographic metadata, funding data, license information, full-text links, ORCIDs, abstracts, and Crossmark updates. These are all available, if included in members' metadata.
- No sign-up is required, and the **data can be freely treated** as facts from members. It is **not subject to copyright**, and its available to use for whatever purpose.

N.B. Crossref asks users to specify a User-Agent header that properly identifies the script and a way to contact you via email.



- REST API Call structure:

`<api_url><operation_1>...<operation_n>?<parameter_1>& ...<parameter_n>`

[1] Crossref REST API, <https://tinyurl.com/yyh5363m>

Crossref REST API: Operations

<https://api.crossref.org>

/works/{doi}	Returns metadata for the specified Crossref DOI.
/funders/{funder_id}	Returns metadata for the specified funder and its suborganizations. (e.g. the funder identified with DOI "10.13039/1000000015")
/prefixes/{owner_prefix}	Returns metadata for the DOI owner prefix. (e.g. 10.1016)
/members/{member_id}	Returns metadata for a Crossref member.
/types/{type_id}	Returns information about a metadata work type. (e.g. journal-article)
/journals/{issn}	Returns information about a journal with the given ISSN

- The above operations could be combined together. For instance, to return the list of works from a given journal: [/journals/{issn}/works](#)

Crossref REST API: Basic query examples

Query format: `<api_url><operation_1>...<operation_n>`

Example (1)

Query

Get the **metadata** of the document identified by **DOI = 10.3233/ds-190016** in **JSON format**.

<https://api.crossref.org/works/10.3233/DS-190016>

Results

```
{
  "status": "ok",
  "message-type": "work",
  "message-version": "1.0.0",
  "message": {
    "indexed": {
      "date-parts": [[2019,4,12]],
      "date-time": "2019-04-12T15:10:57Z",
      "timestamp": 1555081857687
    },
    "reference-count": 18,
    "publisher": "IOS Press",
    "short-container-title": ["DS"],
    "DOI": "10.3233/ds-190016",
    "type": "journal-article",
    ...
  }
}
```

Example (1)

Query

Get the list of **all the documents** published by the "Scientometrics" journal, identified with

ISSN=0138-9130

<https://api.crossref.org/journals/0138-9130/works>

Results

```
{
  "status": "ok",
  "message-type": "work-list",
  "message-version": "1.0.0",
  "message": {
    "facets": {
    },
    "total-results": 5894,
    "items": [ ... ]
  }
}
```

Crossref REST API: Parameters

<code>query=<query_terms></code>	Query the results containing the given terms.
<code>filter=<filter_name:value></code>	Filter results by specific fields having a given value (e.g. “ <code>filter=has-orcid:true</code> ”). The list of filter names is written in the git repository of the API (github.com/CrossRef/rest-api-doc#filter-names).
<code>rows=<integer_number></code>	Controls the number of results returned. The default number is usually set on 20. To get the results summary, this value should be equal to 0 (zero).
<code>sort=<field_name></code>	Sort the results by a certain field. E.g. “ <code>sort=published</code> ”. The list of field names available to be sort on, is written in the git repository of the API (github.com/CrossRef/rest-api-doc#sorting)
<code>order=<order_value></code>	Set the sort order to asc or desc
<code>facet=<facet_name:value></code>	Provides a facet field name along with a maximum number of returned term values. For example “ <code>facet=orcid:10</code> ”.

Crossref REST API: Parameter query examples

Query format: `<api_url><operation_1>...<operation_n>?<parameter_1>& ...<parameter_n>`

Example (1)

Query

Get all the **documents** from the “Major Gifts Report” journal (**ISSN=1549-7712**), which have been **indexed after the 2018 year**. **Sort** the returned results in **descending order**, according to their **published-print date**.

<https://api.crossref.org/journals/1549-7712/works>
[?filter=from-index-date:2018&sort=published-print&order=desc](https://api.crossref.org/journals/1549-7712/works?filter=from-index-date:2018&sort=published-print&order=desc)

Results

```
{
  "status": "ok",
  "message-type": "work-list",
  "message-version": "1.0.0",
  "message": {
    "facets": {},
    "total-results": 24,
    "items": [ ... ]
  }
  ...
}
```

Example (1)

Query

Get an **overview** of the **total number of works** in Crossref classified according to their **type**.

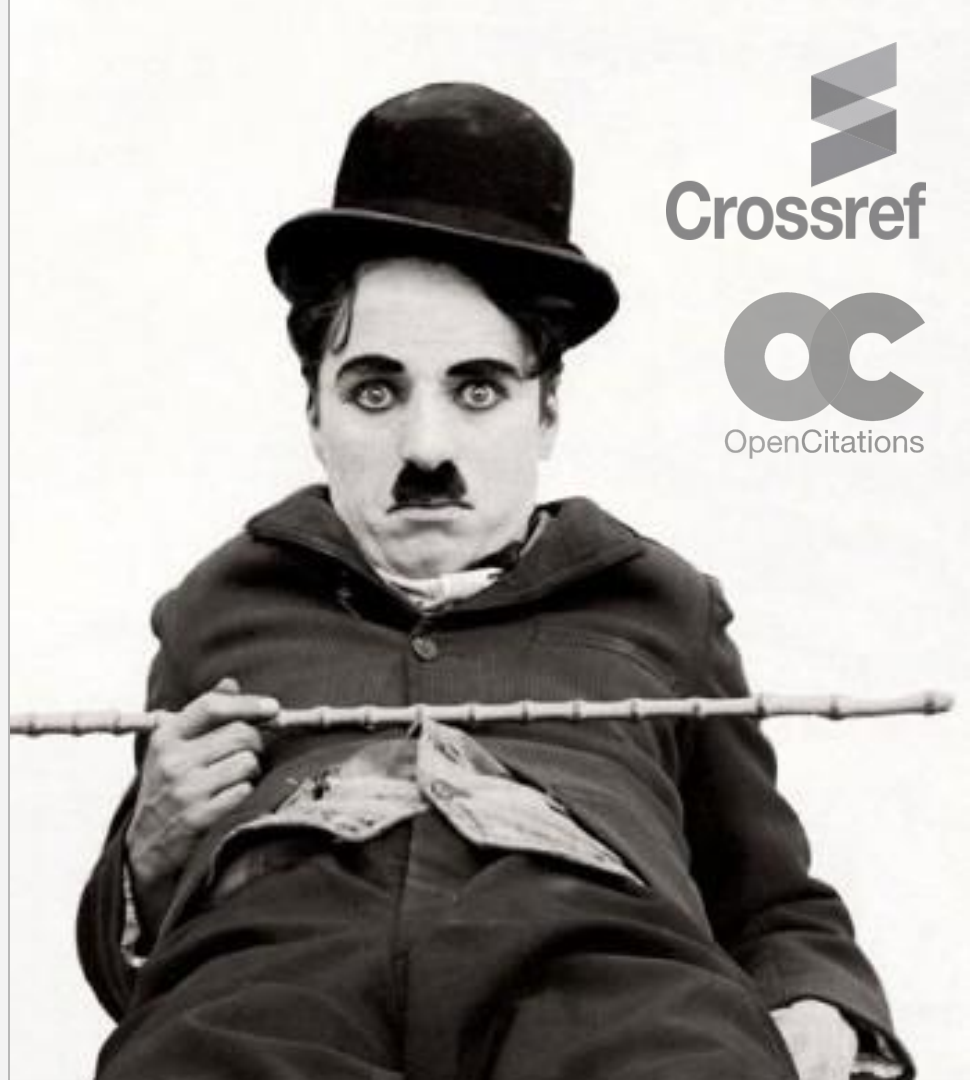
<https://api.crossref.org/works>
[?rows=0&facet=type-name:*](https://api.crossref.org/works?rows=0&facet=type-name:*)

Results

```
{...
  "message": {
    "facets": {
      "type-name": {
        "value-count": 26,
        "values": {
          "Journal Article": 77859296,
          "Chapter": 13268443,
          "Conference Paper": 5930443,
          "Component": 3859301,
          "Dataset": 1811741,
          "Entry": 871054,
          ...
        }
      }
    }
    "total-results": 107417941,
    "items": [ ... ],
    ...
  }
}
```

Thank you
for your attention

**Please use the OpenCitations
and Crossref REST APIs, and
contact us for any help.**



Exercises (1)

Compare the number of citations appearing inside the OpenCitations Corpus and those inside the COCI dataset, for the document identified with DOI = **"10.4103/2008-7802.156835"**.

Solution (highlight the text under to see it)

{

}

Exercises (2)

Get the list of citations for the document identified by the DOI=**“10.1002/adfm.201505328”** from the COCI dataset, which are also journal self citations. Then use the Crossref API to identify the name of the journal.

Solution (highlight the text under to see it)

```
{
```

```
}
```