# Availability of Open Citations from Open Journals in Crossref - Protocol V.2

Davide Brembilla[1], Chiara Catizone[1], Giulia Venditti[1]

[1]Alma Mater Studiorum - Università di Bologna

2

May 09, 2022

1

dx.doi.org/10.17504/protocols.io.kxygxz7ywv8j/v2

Open Science 2021/2022

Davide Brembilla

This protocol is for the research about the availability of Open Citations from Open Journals in Crossref.
The goal is to find out how many papers from DOAJ journals are available on Crossref, whether their metadata is available and the origin of their references' DOIs.

Our research involves the articles from Open Access journals in DOAJ and their data on Crossref. We scouted the availability of these articles' reference lists, the presence of IDs such as DOIs and the entities responsible for their specification. This analysis was carried using DOAJ article metadata dump, then modified and enriched thought Crossref's APIs DOI requests. The data will be further analysed to verify the distribution of the provenance of the articles' DOIs.

DOI

dx.doi.org/10.17504/protocols.io.kxygxz7ywv8j/v2

Davide Brembilla, Chiara Catizone, Giulia Venditti 2022. Availability of Open Citations from Open Journals in Crossref - Protocol. **protocols.io** https://dx.doi.org/10.17504/protocols.io.kxygxz7ywv8j/v2 Davide Brembilla

We improved the sections following a review. We specified the software used.

Open Science

protocol ,

May 08, 2022

May 09, 2022

May 09, 2022 | Davide Brembilla

62218

This methodology cleans and populates the batches of articles downloadable from the DOAJ website and populates them with the infromations from Crossref's API.
All the software to perform this methodologies can be found in the Github Repository for it. All the software is distributed under the MIT licence.
The main software is in **populate.py**, while **main.py** is the script used to launch the query from the command line.
If you want to query Crossref through a DOI, you can create a **populateJson** instance and query the API with just the DOI. If you are interested in more complete access to the API, you can find here a more complete list of libraries that have a wider scope; another useful software that was of inspiration for the one used in here is oc_graphenricher.

All the software to perform this methodologies can be found in the Github Repository for it. All the software is distributed under the MIT licence.
This methodology cleans and populates the batches of articles downloadable from the DOAJ website and populates them with the infromations from Crossref's API.

Querying Crossref may be a long process, varying depending on your connection and on the power of your PC.

All the software to perform this methodologies can be found in the Github Repository for it. In order to use it, you will need Python 3 as well as install the libraries used in the **requirements.txt** file. To install them, you can use the command

```
pip install -r requirements
pip3 install -r requirements
```

## Data Gathering

1 We download the DOAJ public data dump containing **article metadata** in tar.gz format.
The dump is structured as a single directory of the form **doaj_article_data_[date generated]** where are listed files with names of the form **article_batch_[number].json**.
Each file contains up to 100,000 records for a total size of 270Mb.

2 To decrease the size of each batch, we filter each key to get only the information useful for our research:



**Citation:** Davide Brembilla, Chiara Catizone, Giulia Venditti Availability of Open Citations from Open Journals in Crossref - Protocol https://dx.doi.org/10.17504/protocols.io.kxygxz7yvv8j/v2

- DOI of the article
- year of publication
- ISSNs journal they belong to

To do so we used the **batches_cleaner.py** software.

```
Import needed:
▪ os
▪ json

Script to execute:
▪ python3 -m batches_cleaner doaj_article_data_[date generated]
From Windows:
▪ py -m batches_cleaner doaj_article_data_[date generated]
```

**Example:**


{
"10.3390/math6060103": {
"year": "2018",
"issns": [
"2227-7390",
"1648-9144"
]
},
...
}


3 In order to reply to our research questions we should enrich the cleaned dump with information retrieved by Crossref REST API. To do so we used the **main.py** script that launches the **populate.py** software, both developed in python. It uses three method explained in the subsections below:

- **requests method**
- **_read_json method**
- **populate method**

```
Import needed:
▪ os
▪ json
```



**Citation:** Davide Brembilla, Chiara Catizone, Giulia Venditti Availability of Open Citations from Open Journals in Crossref - Protocol https://dx.doi.org/10.17504/protocols.io.kxygxz7ywv8j/v2

- ABC
- requests
- time
- random
- requests_cache
- backoff

Script to execute:
- python3 -m main <path/> doaj_article_data_[date generated]

From Windows:
- py-m main <path/> doaj_article_data_[date generated]

**Example:**


```
{
"1664-2392": {
"10.3389/fendo.2021.777589": {
"crossref": 1,
"year": "2021",
"reference": {
"B49": {
"doi": "10.21203/rs.3.rs-438962/v1",
"doi-asserted-by": "publisher"
        }
      }
    },
  ...
  },
...
}
```


## 3.1 requests method.

This method launch requests to Crossref REST API in the format https://api.crossref.org/works/{doi}. The method use requests_cache, backoff, ReadTimeout to avoid getting blocked and speed up the process.

```
def __init__(self) -> None:
    requests_cache.install_cache('multithread_cache')
    self.api = "https://api.crossref.org/works/"
```



**Citation:** Davide Brembilla, Chiara Catizone, Giulia Venditti Availability of Open Citations from Open Journals in Crossref - Protocol https://dx.doi.org/10.17504/protocols.io.kxygxz7ywv8j/v2

```
    @backoff.on_exception(backoff.expo,
requests.exceptions.ReadTimeout, max_tries=20)
    defquery_crossref(self, doi):

    query = self.api + doi
    time.sleep(random.randint(1,3))

    req = requests.get(query, timeout=60)
    returnreq, doi
```

## 3.2 _read_json method.

This method iterates over articles stored in **article_batch_[number].json,** indexing DOI for their journal ISSNs then, for each article DOI, we send a request through request method. If the process is having some errors it saves a temporary file in the directory temp.

If the response status code is not equal to 200 the DOI is not present in Crossref, so we add those key value:

- "'crossref': 0
- "reference": 0

In other cases we read the response message and we add those key value:

- "crossref": 1
- "reference": 0

Then, if in the response are presents reference values we read the reference list to read reference articol information. Id DOI is present we add this key value to our reco:

- "doi": "article-doi"
- "doi": "not-specified" (if not present)

Then we look for the entities responsible for DOI specification that crossref saved under the key "doi-asserted-by". Consequentially, we add to reference informations this key value:

- "doi-asserted-by": "publisher"
- "doi-asserded-by": "crossref"
- "doi-asserted-by": "not-specified" (if not present)

When the file is finally populated a temporary copy is saved in temp/compleated to save time whether we need to restart the process.

## 3.3 populate method.



**Citation:** Davide Brembilla, Chiara Catizone, Giulia Venditti Availability of Open Citations from Open Journals in Crossref - Protocol https://dx.doi.org/10.17504/protocols.io.kxygxz7ywv8j/v2

Populate iterates over all files in the directory **doaj_article_data_[date generated]** controlling if they are temporary files. In the end it produces a JSON file as output named **batch.json** in the **output** directory.

Publishing Data

4 We will publish our datasets at the end of our research in JSON formats on https://github.com/open-sci/2021-2022-la-chouffe-code together with the two software mention above and software documentation.
We will provide a permanet doi for the software and the research's results.

Data Analysis

5 We will observe the percentage of articles on DOAJ that are present in Crossref, in particular:

1. DOAJ articles w/DOI // Total DOAJ Articles : % Of articles with a DOI Over the DOAJ
   This gives the scope of DOAJ articles with a persistent identifier and that have a DOI. Also informs about the scope of the rest of the research.
2. DOAJ articles on crossref // DOAJ articles w/DOI : % articles indexed on crossref.
   This gives the scope of the articles of the DOAJ indexed on crossref. This is important as articles indexed on crossref have their metadata available for everyone outside the DOAJ and helps indexing
3. DOAJ articles w/references on Crossref // DOAJ articles on Crossref +DOAJ articles w/references on Crossref // DOAJ articles with DOI : % of articles of the doaj (and on crossref) with references that are indicated.
4. N. references issued by Crossref // Total number of references + N. of references issued by the publisher // Total number of references: % of references issued by Crossref or publisher. This can be useful to understand the role of crossref in registering articles and related metadata.
5. Y1, % of articles on Crossref in Y1, % articles with references in Y1… YN, % of articles on Crossref in YN, % articles with references in YN. This gives us information about the yearly evolution of publishing of scholarly metadata over Crossref
6. Issns,% articles with doi, % articles on cf for issn, % articles with references, %by publisher / Crossref: How are different journals in publishing metadata, how much metadata on crossref, how many open citations are put there.

Data Visualisation

6 We will employ python libraries such as seaborn or plotly to visualise our results.



**Citation:** Davide Brembilla, Chiara Catizone, Giulia Venditti Availability of Open Citations from Open Journals in Crossref - Protocol https://dx.doi.org/10.17504/protocols.io.kxygxz7ywv8j/v2