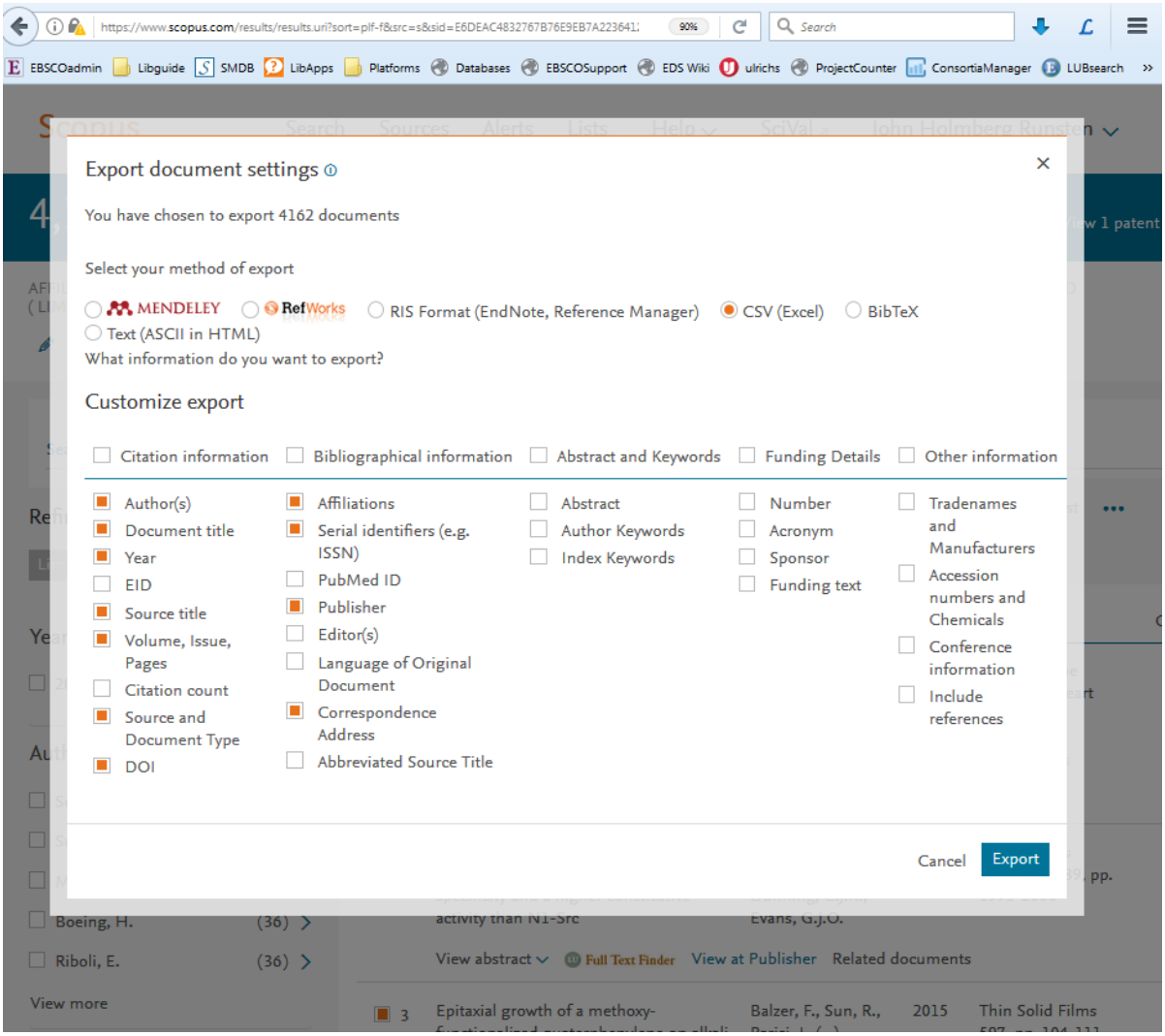
Collecting Open Access information using OpenRefine and the oaDOI API (special bonus: the DOAJ API)

This tutorial explains how OpenRefine (http://openrefine.org/) together with the oaDOI API (https://oadoi.org/) from Impactstory can be used to collect Open Access information. aoDOI collects open access evidence from DOAJ, crossref, BASE, and pmicd. This can be useful when trying to identify OA articles published by one's institution. In this tutorial records from Scopus have been used.

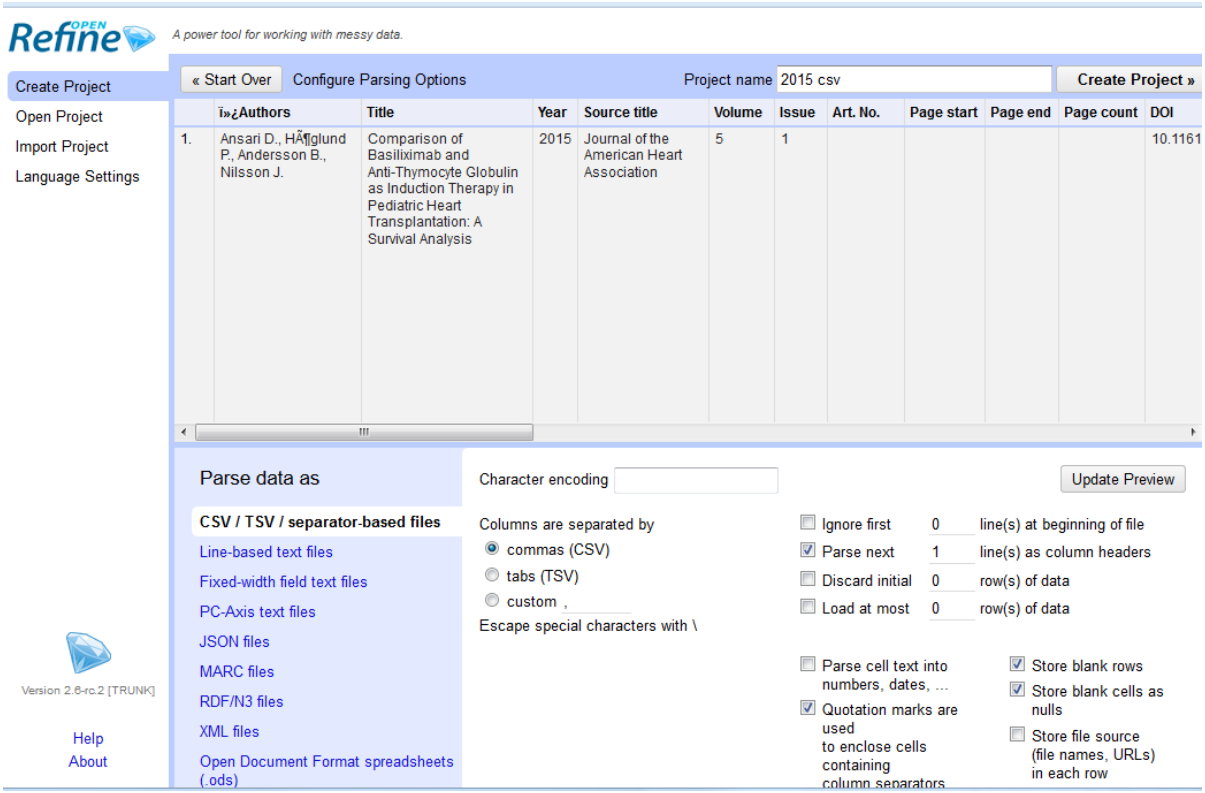
Export records from Scopus using export function.

You can customize your export to include publisher, affiliation, correspondence address and ISSN. Exporting ISSN can be a good idea if you plan on retrieving information using the DOAJ API (while you're still at it). Unfortunately, Scopus currently only allows 2000 records to be exported at a time. And you might think - hey, still better than WoS's 500. But the problem with Scopus right now is that you can't export say records 2001-4000. So harvesting big amounts of data from Scopus is a bit tricky right now.



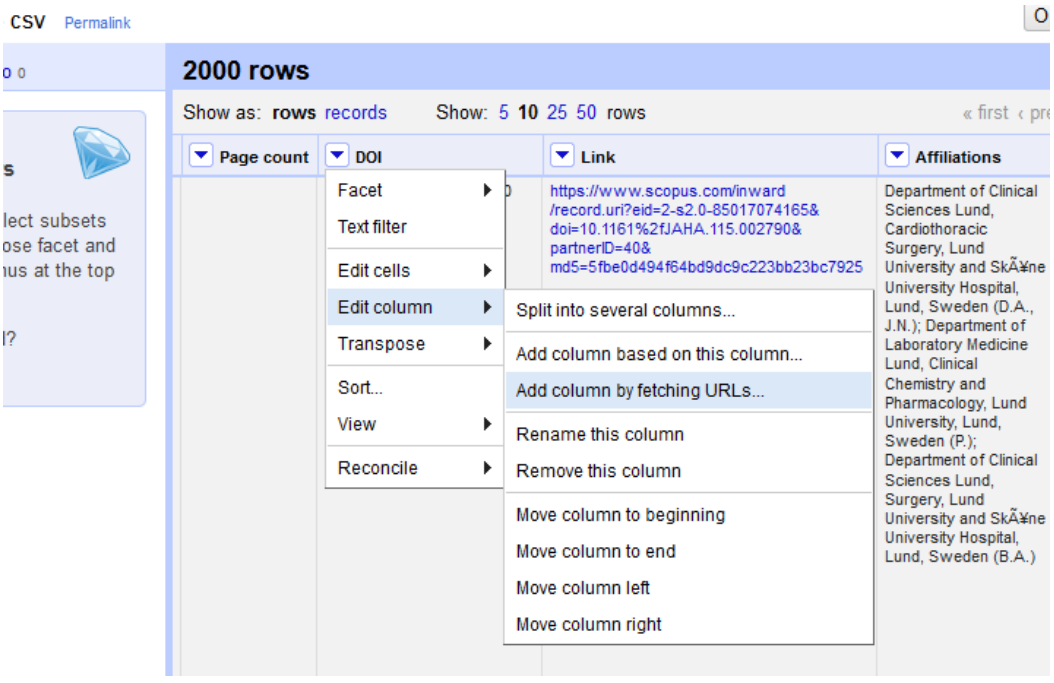
Import .csv file to OpenRefine

Only thing here to remember is to change the Character encoding to UTF-8 and give you project a fancy name.

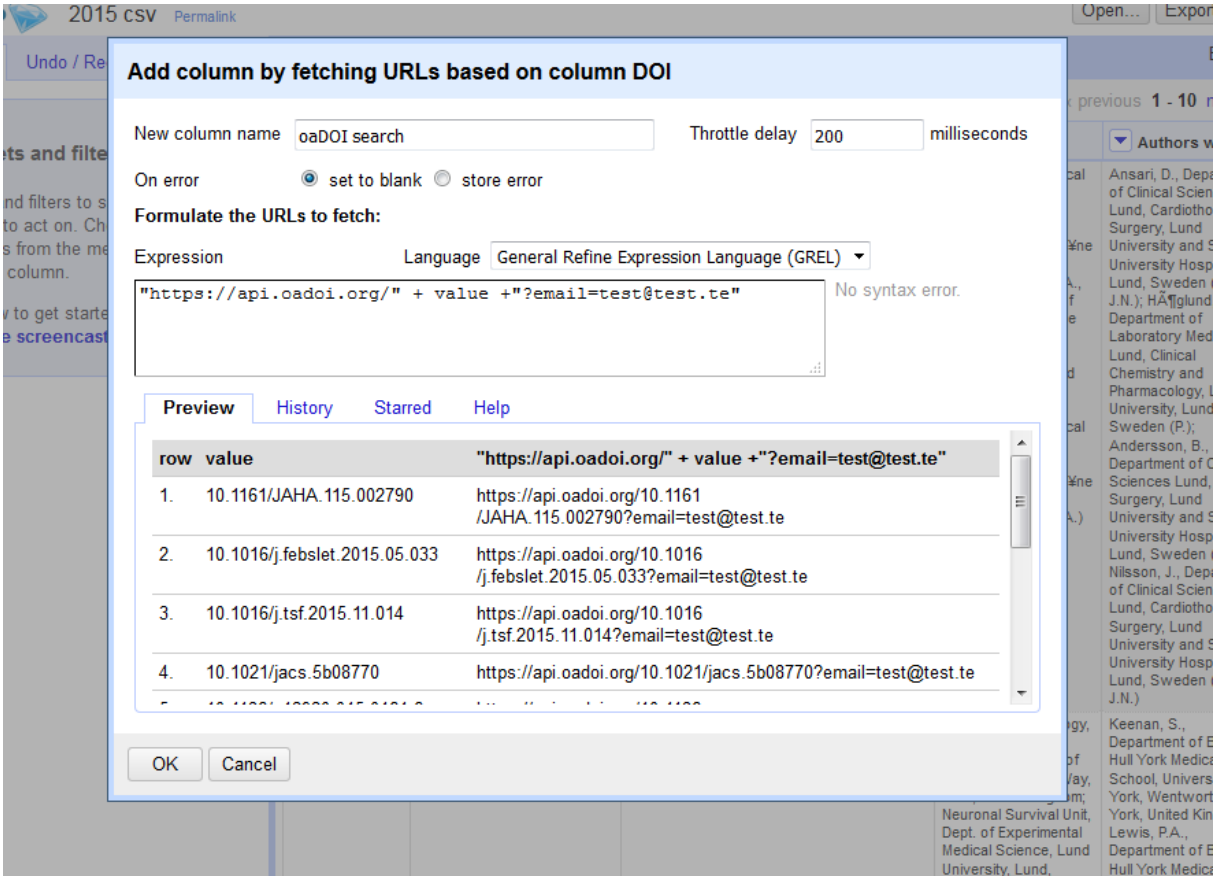


Add column by fetching URLs

First select the DOI column and choose Add column by fetching URLs:



Then input "https://api.oadoi.org/" + value +"?email=[[your email]]". You can always try your query before you start fetching 2000 records by simply copy-pasting a preview into your web browser. Remember to change Throttle delay or the process will be very slow.



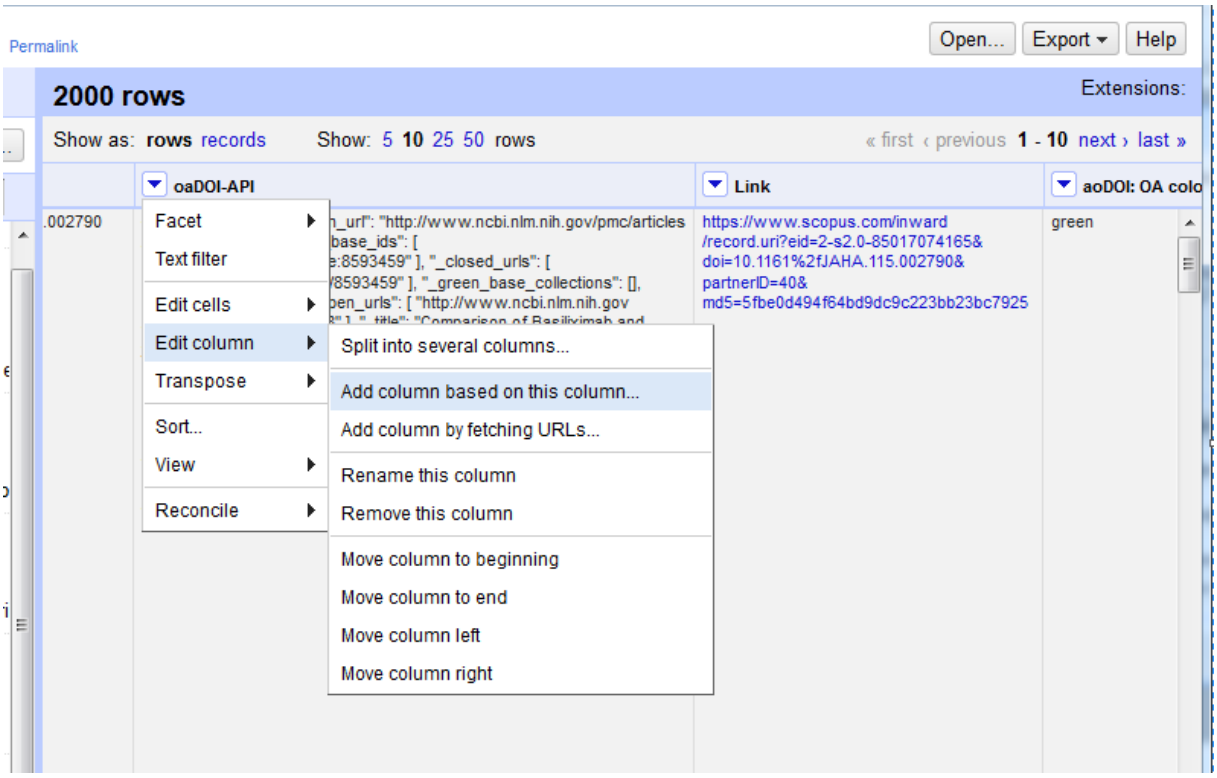
Parse that Json!

When OpenRefine is done fetching you'll have a new column filled with Json code ready to be parsed.

So this an example of what the oaDOI API gives you:


```
{
    "results": [{
            "_best_open_url": "http://doi.org/10.3389/fneur.2016.00240",
            "_closed_base_ids": [],
            "_closed_urls": [],
            "_green_base_collections": [],
            "_open_base_ids": [],
            "_open_urls": ["http://doi.org/10.3389/fneur.2016.00240", "http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5179561"],
            "_title": "Cerebral Reorganization in Patients with Brachial Plexus Birth Injury and Residual Shoulder Problems",
            "doi": "10.3389/fneur.2016.00240",
            "doi_resolver": "crossref",
            "evidence": "oa journal (via journal title in doaj)",
            "found_green": true,
            "found_hybrid": false,
            "free_fulltext_url": "http://doi.org/10.3389/fneur.2016.00240",
            "is_boai_license": true,
            "is_free_to_read": true,
            "is_subscription_journal": false,
            "license": "cc-by",
            "oa_color": "gold",
            "oa_color_long": "gold_doaj",
            "reported_noncompliant_copies": [],
            "url": "http://doi.org/10.3389/fneur.2016.00240",
            "version": null,
            "year": 2016
        }
    ]
}
```

Now select your new column and choose Add column based on this column:



Here is how you parse the code: For example, you want to know if the article is from a hybrid journal. Then write the expression: **value.parseJson().results[0].found_hybrid**

value is value which you are parsing.

parseJson is the command

results[0] is the object where you're information can be found

and, **found_hybrid** is a key in the object which give you the value which follows it, in this case either TRUE or FALSE.

So, what you'll get using the above expression is a new column with either TRUE or FALSE if the article is found to be from a hybrid journal.

Another useful parsing expressions is: **value.parseJson().results[0].evidence** which gives you information from where the OA information has been harvested. Or, **value.parseJson().results[0].oa_color** which tells you if the OA is green, gold, or blue.

Remember you can save your expressions (star).

Done?

At this point I usually export my projects to excel. But, there are loads of other stuff you can do in OpenRefine with your data. For example, I could use Cluster & edit on my publisher column to deal with alternative spellings of deferent publishers. So If I have Elsevier Inc. and Elsevier Ltd OpenRefine can standardize the names.

Fetching from DOAJ

The DOAJ API (https://doaj.org/api/v1/docs) can give you OA information (e.g. APC, Open access start year, license) on the journal level as well as some other information (e.g. subject, peer-review).

However, if you're using Scopus records, you will have to hyphen your ISSNs. Do this by selecting the ISSN column and choose Add column based on this column. The expression I use to add hyphens to ISSNs is : **substring(value,0,4) + "-" + substring(value,4,8)** .

After you've created a column with hyphened ISSNs select your new column and choose Add column by fetching URL. You can use the following expression:
"https://doaj.org/api/v1/search/journals/issn%3A" + value

To parse the fetched information, select your new column. Json from DOAJ has more objects than Json from oaDOI. It's easier to figure out how the code should be parsed if you copy paste the code to an editor that understands Json. Below is an excerpt of what Json from DOAJ can look like:


```
{
    "last": "https://doaj.org/api/v1/search/journals/issn:1664-2295?page=1&pageSize=10",
    "pageSize": 10,
    "timestamp": "2017-0705T10:54:59Z",
    "results": [{
            "last_updated": "2017-02-28T13:06:21Z",
            "id": "4100f74160f1455492602f400f1e490e",
            "bibjson": {
                "allows_fulltext_indexing": true,
                "persistent_identifier_scheme": ["DOI"],
                "keywords": ["clinical neuroscience"],
                "apc": {
                    "currency": "USD",
                    "average_price": 1900
                },
                "subject": [{
                        "code": "RC346-429",
                        "term": "Neurology. Diseases of the nervous system",
                        "scheme": "LCC"
                    }
                ],
                "article_statistics": {
                    "url": "http://loop.frontiersin.org/about",
                    "statistics": true
                },
                "title": "Frontiers in Neurology",
                "publication_time": 14,
                "provider": "Frontiers Media S.A.",
                "format": ["PDF", "HTML", "ePUB", "XML"],
                "plagiarism_detection": {
                    "detection": true,
                    "url": "http://frontiersin.org/neurology/reviewguidelines"
                },
                "apc_url": "http://frontiersin.org/neurology/fees",
                "link": [{
                        "url": "http://frontiersin.org/neurology",
```


And here are some useful expression for parsing information from DOAJ:

value.parseJson().results[0].bibjson.license[0].open_access

value.parseJson().results[0].bibjson.oa_start.year

value.parseJson().results[0].bibjson.apc.average_price

value.parseJson().results[0].bibjson.license[0].title

Usefull links:

Good intro movies on how to use OpenRefine: http://openrefine.org/

OpenRefine recepies: https://github.com/OpenRefine/OpenRefine/wiki/Recipes

Getting started with OpenRefine: http://thomaspadilla.org/dataprep/