OpenISU •

An Easier Way to Access the Web of Science API

Eric Schares

Published on: Dec 09, 2021

License: [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

Analyzing bibliographic data

Library collection and scientometric projects are based on the analysis of publication metadata, typically coming from an article index or database. At Iowa State University (ISU), we are working to support more open publishing models, and a critical step in this is to analyze data from publications that have a Corresponding Author¹ from ISU.

[Web of Science](#) is a subscription-based citation index that includes information on the Corresponding Author (CA) of each publication, but this detail is not provided to the user as a searchable term or in a limiting facet. This means that one must first export the results and run post-processing to filter the list of papers from those with an ISU author *anywhere* in the string down to only those papers with an ISU Corresponding Author.

Iowa State University publishes around 4,000 articles per year which are indexed in Web of Science. (The database takes a selective approach to content [1] and does not index the contents of every journal; therefore, the records of some ISU-authored publications are not included in this database.) To get the CA information for ISU-authored records, I must export in batches of 500 (the export limit), then reconstruct the multiple individual files. While not a huge deal, it is an annoyance to run eight separate export commands, save the files, and re-assemble them after the fact.

The Web of Science API offers direct access to the underlying data instead of interacting with the records through the web interface. Since CA is not offered as a facet in the web interface, this work requires access to the exported data anyway; using the API is faster, more convenient, and can offer even richer data than is available through the web.

Sounds great, just use the WoS API then

Unfortunately, I have always found the Web of Science API to be difficult to use and understand. There are varying levels of API access possible, and these must be included in the library's negotiated contract with Clarivate. Even when access has been agreed upon, I have found the approval procedure to be cumbersome, slow, and opaque. I had to reach out to our sales rep to get my request expedited, and even after I have was granted an API key (a secret string of 40 characters that uniquely identifies me), I still run into errors.

Python libraries such as [wos](#) for Web of Science (or [pybliometrics](#) for Scopus) attempt to provide a structure to automate the API query process through a script, but I have never been able to get them to work.

I consider myself fairly tech-savvy, but without a Computer Science degree, I quickly get lost at this step. Generous colleagues from around the world have been kind enough to share their Jupyter Notebooks or Python code with me as demonstrations, but these are still uniquely customized and tailored to their specific institutions and use cases, and are not completely generalizable to adapt locally.

It is an enormous jump to go from interacting with the WoS web interface to an environment that requires you to understand GET requests, query strings, and unpacking nested JSON responses.

```
from pandas import json_normalize

def get_author(static_data, uid_list):
    # create dictionary for author data
    author_data = {}

    for i in range(len(static_data)):
        author = static_data[uid_list[i]]['summary']['names']
        author_data[uid_list[i]] = author

    af_dict = {}
    rp_dict = {}
    same_as_rp = {} # if AF = RP

    # parse author_data for AF and RP
    for j in range(len(author_data)):
        # if there is only one author (= reprint author)
        if author_data[uid_list[j]]['count'] == 1:
            name = author_data[uid_list[j]]['name']['full_name']
            af_dict[uid_list[j]] = name
            rp_dict[uid_list[j]] = name
            same_as_rp[uid_list[j]] = 1

        # if there are multiple authors
        elif author_data[uid_list[j]]['count'] > 1:
            af_list = []
            rp_list = []

            for k in range(len(author_data[uid_list[j]]['name'])):
                # get author(s)
                if "reprint" not in author_data[uid_list[j]]['name'][k]:
                    name = author_data[uid_list[j]]['name'][k]['full_name']
                    af_list.append(name)
```

Example of an advanced Web of Science API parsing script, ©Tran Ha

Note: I find the database [Dimensions](#) to have the most user-friendly API, with clear documentation, multiple routes to access, and plenty of worked examples to start from. Unfortunately, Dimensions relies on publishers to provide CA information in the publication metadata, and not all publishers include this in their records.

It is an enormous jump to go from interacting with the WoS web interface to an environment that requires you to understand GET requests, query strings, and unpacking nested JSON responses.

Enter the WOS API Excel Converter tool

It turns out Clarivate already has a tool to help facilitate pulling this data, the **WOS API Excel Converter**.

I had never heard of this tool and only learned of it when working through a separate issue with Clarivate Support. For all its apparent usefulness, I have never seen it referenced anywhere and am surprised that Clarivate does not make it more widely known.

Launched in February 2021, this small tool simplifies the process of accessing the WoS API data and helps the user bridge that gap when going from a familiar point-and-click site to a more sophisticated API request. A bonus is that it exports the data directly to Excel - no more messing with unpacking JSON and extracting the data values you want.

Per the product page, <https://github.com/clarivate/wos-excel-converter>:

This is a small and easy-to-use desktop application that allows exporting Web of Science API Expanded data in Excel with a configurable and flexible data export structure.

If you visit the above link, you might be confused (as I was) on what you're actually supposed to run. The real link is a "secret" page, <https://github.com/clarivate/wos-excel-converter/releases>, which has the actual executables to download and install on your specific system.

1. Get Access

Step 0 is to sign in to the [Clarivate Developer Portal](#) and apply for an API Key. As I mentioned above, this may take several days to a few weeks to complete.

Once you are approved and assigned, Step 1 is to paste that key in the tool and click Verify. You can see I am authenticated and have already retrieved some of the 1,000,000 records my account is approved for.

It would be nice if the tool remembered the API Key across sessions, but for now, you do need to cut and paste that string every time, so keep it in a place that is easily accessible.

Enter the API Key and click Verify

2. Query

Next, construct the query using the [WoS query parameters](#). These are the same as in the Advanced Search from the web interface of WoS.

In the example below, I am retrieving records that have an author from Iowa State University (OG=Organization Enhanced) anywhere in the author string, and were published in 2020 (PY=Published Year).

You can also specify which database to search and the language used.

The screenshot shows the 'Web of Science™ API Expanded Excel Converter' interface. On the left is a sidebar with four steps: 1. API Token (Remaining records: 994,868), 2. Search details (Records found: 4,115), 3. Attribute selection, and 4. Generate File. Step 2 is currently active. The main area is titled '2. Search details' and contains three input fields: 'databaseId' (set to 'WOS'), 'lang' (empty), and 'edition' (empty). Below these fields are explanatory notes. The 'usrQuery' field contains the query 'OG=(Iowa State University) AND PY=2020'. At the bottom, there is a 'VALIDATE' button and a green confirmation message: 'Validation passed. Records found: 4,115'. The interface is powered by Web of Science™.

Web of Science™
API Expanded Excel Converter

1. API Token
Remaining records: 994,868 ✓

2. Search details
Records found: 4,115 ✓

3. Attribute selection ✓

4. Generate File

2. Search details

databaseId
WOS
Database to search. WOK represents all databases.

lang
Language of search. This element can take only one value: en for English. If no language is specified, English is passed by default.

edition
Edition(s) to be searched. If empty, user permissions will be substituted. Must include the name of the collection and edition name separated by '+', ex: WOS+SCI. Multiple editions are separated by ';'. Editions available for collection (WOS) - AHCI,CCR,IC,ISSHP,ISTP,SCI,SSCI,BHCI,BSCI and ESCI.

usrQuery
OG=(Iowa State University) AND PY=2020
User query for requesting data, ex: TS=(cadmium). The query parser will return errors for invalid queries. [Click here](#) for more details.

VALIDATE

Validation passed. Records found: 4,115

About

Powered by **Web of Science™**

In Step 2 of the tool, specify the database and language, then construct the query

The Validate button helps preview how many records the query will find if actually run. In this case, around 4,000 is what I expect, while a 0 is a good indication that something went wrong.

Notably, the Publisher [PU] tag is not supported in either the API or the Advanced Search. This would be a welcome addition since my work often focuses on ISU's publishing output with a given publisher. For now, I still need to export *everything* ISU produced in a given year, then filter on the PU field that appears within the exported data.

3. Customize the output

Step 3 has some options regarding the JSON output and how the data is split across tabs in the resulting Excel spreadsheet. I leave this as the default, though a motivated user could dive in to understand this more and customize it to their liking.

However, I believe any changes would also be reset when closing and re-launching the tool, so those customizations would need to be re-done each time.

Define how the JSON output gets split across multiple Excel tabs

4. Specify filename and run the export

Finally, Step 4 defines the path and filename of the output before executing. Another nice feature is the double-sided slider, which lets you export only a selection of the records returned by your query.

“Start export” actually runs the query and provides an estimate of the time remaining. It handles the looping and segmenting of sections of records, combining them into one single output file for you.

The screenshot shows the 'wos-excel-converter' application window. On the left, a sidebar lists four steps: 1. API Token (Remaining records: 998,362), 2. Search details (Records found: 4,109), 3. Attribute selection, and 4. Generate File (selected). The main area is titled '4. Generate file' and contains the following elements:

- Directory of file:** A text box showing the path 'C:\Users\eschares\Desktop\eschares\Collection Development\Collection_data_analyst\tools\Web_of_Sc' with a folder icon.
- Filename:** A text box containing 'WOS_Export' and a dropdown menu showing '_ddMMyyyyHHmmss_partX.xlsx'.
- Range selection:** A slider bar with the value '1151' on the left and '4109' on the right.
- Start export:** A purple button.
- Progress bar:** A bar showing '26%' completion, with the text 'Remaining time: 0h 3 min 5 sec' above it.
- Status message:** 'Your file is being generated. Until then nothing is clickable.'
- Footer:** 'Powered by Web of Science™' and an 'About' link.

Start Export and let the tool run

Once the export is complete, the file will be saved and data displayed across multiple tabs, as defined in Step 3 above.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	UT	Database	edition_1	edition_2	doctype_1	doctype_2	doctype_3	doctype_4	primaryLa	citedRef	citationCo	keywords	keywords	bibtype	bib_page	bib_id	subheadir	subheadir	subject_trsu
2	WOS:0007	WOS	WOS.ESCI		Article				English	19	1	Doubly pe	PERIODIC	Journal	659	1: 590-597	Technology	Engineerir	Te
3	WOS:0007	WOS	WOS.ISTP		Proceedings Paper				English	24	0	Missing data,	Missing	Book	766	11906: 56	Technology	Computer Co	
4	WOS:0007	WOS	WOS.BSCI		Article	Book Chapter			English	226	3	Throughfa	DISSOLVE	Book	281	: 229-251	: Life Scienc	Physical St	Ecology
5	WOS:0007	WOS	WOS.ISTP		Proceedings Paper				English	11	0	V-GO extraction,	volt;	Book	2515	: - 2020	Technology	Engineering,	
6	WOS:0007	WOS	WOS.ISTP		Proceedings Paper				English	27	0	dielectric i	WATER,RF	Book	2515	: - 2020	Technology	Engineering,	
7	WOS:0007	WOS	WOS.ISTP		Proceedings Paper				English	13	0	duty cycle	PULSEWID	Book	2515	: - 2020	Technology	Engineering,	
8	WOS:0007	WOS	WOS.ESCI		Review				English	89	2	Generaliz	PROPENSI	Journal	313	3 (2): 625-	Physical Sciences	Statistics & P	
9	WOS:0006	WOS	WOS.BHCI		Article	Book Chapter			English	10	1			Book	157	: 15-25	2020	Social Issu So	
10	WOS:0007	WOS	WOS.SCI		Article				English	10	0		RESOURC	Journal	427	2020: - 20:	Life Scienc	Technolog	Agronomy Pl
11	WOS:0007	WOS	WOS.SCI		Article				English	83	4		PHOSPHO	Journal	427	2020: - 20:	Life Scienc	Technolog	Agronomy Pl
12	WOS:0007	WOS	WOS.SCI		Article				English	24	1		MODEL	Journal	427	2020: - 20:	Life Scienc	Technolog	Agronomy Pl
13	WOS:0006	WOS	WOS.ISTP		Proceedings Paper				English	29	0	user identity	linkage,	Book	5500	: - 2020	Technology	Computer Te	
14	WOS:0006	WOS	WOS.ISTP		Proceedings Paper				English	23	0	social networks,	circle	Book	5500	: - 2020	Technology	Computer Te	
15	WOS:0006	WOS	WOS.ISTP		Proceedings Paper				English	13	0			Book	5500	: - 2020	Technology	Computer Te	
16	WOS:0006	WOS	WOS.AHCI		Book Review				English	1	0			Journal	56	112 (1): 37-38	WIN 2020	History	
17	WOS:0007	WOS	WOS.SCI	WOS.ISTP	Meeting Abstract				English	0	0			Journal	220	110 (12): 1	Life Sciences & Biom	Plant Science	
18	WOS:0007	WOS	WOS.SCI	WOS.ISTP	Meeting Abstract				English	0	0			Journal	220	110 (12): 7	Life Sciences & Biom	Plant Science	
19	WOS:0007	WOS	WOS.SCI	WOS.ISTP	Meeting Abstract				English	0	0			Journal	220	110 (12): 6	Life Sciences & Biom	Plant Science	

Resulting data in Excel

Limitations

To begin, your institution must have access to both Web of Science *and* the API Expanded. Access to the WoS website does not necessarily mean you also have access to the API. If your library does, you must then individually apply and sign up to get an API key for input into Step 1 of the converter tool.

This tool is not perfect. The filenames seem to have a bug in assigning the export date (issue has been reported), and the data is split across multiple tabs in the resulting Excel sheet in a convoluted and confusing way. The Publisher facet would be useful to make available in the query language, and making the API Key and any JSON changes sticky would be a nice user experience improvement.

However, I do find this tool to be a welcome improvement, and more people could benefit from knowing about it. Getting direct access to the API through Python or R code gives greater control over what exactly you extract, how it is structured, and what is returned, but for those users (like me) who are ready to move beyond the web interface but still find the script access intimidating, this tool is a useful middle ground.

Footnotes

1. “Corresponding Author” is a designation given to at least one author of a paper. Traditionally, it has meant the person who communicates with the journal and walks

the paper through the submission process, but more recently it has become the agreed-upon way to tell which university or institution a paper is assigned to. This in turn determines if the paper will be covered under an institution's Open Access agreement with a publisher. [↵](#)

Citations

1. <https://clarivate.libguides.com/webofscienceplatform/coverage> [↵](#)