



What are Scopus APIs and how are these used?

Introduction to Scopus APIs

Kai Wan

Senior Product Manager, Scopus



Why are Scopus APIs Important?

- As an abstracting and indexing database, **Scopus captures articles** being published in **virtually all scholarly journals of any significance in the world**; and its profiling of authors and institutions makes it easy to find new articles by those authors at those institutions.
- The **Scopus UI (user interface)** offers many features to that end, allowing **librarians, researchers, developers and & business intelligence groups** to manually find publications originating from their institution that they can then add to their systems.
- Aside from that user interface, Scopus also **has Application Programming Interfaces (APIs)** that offer the same features, but then in a **machine-readable format that enables *software*, rather than *humans on the UI (User Interface)*, to find articles, authors and institutions in Scopus.**
- This allows developers to write programs that **automatically extract data** from Scopus, and add that data to their systems.

Before we talk about APIs, let's go back to the Scopus Data Model

The Scopus data model is designed around the notion that *articles* are written by *authors* that are *affiliated with institutions*. Visually and rather simplistically, this relational model can be represented like this:

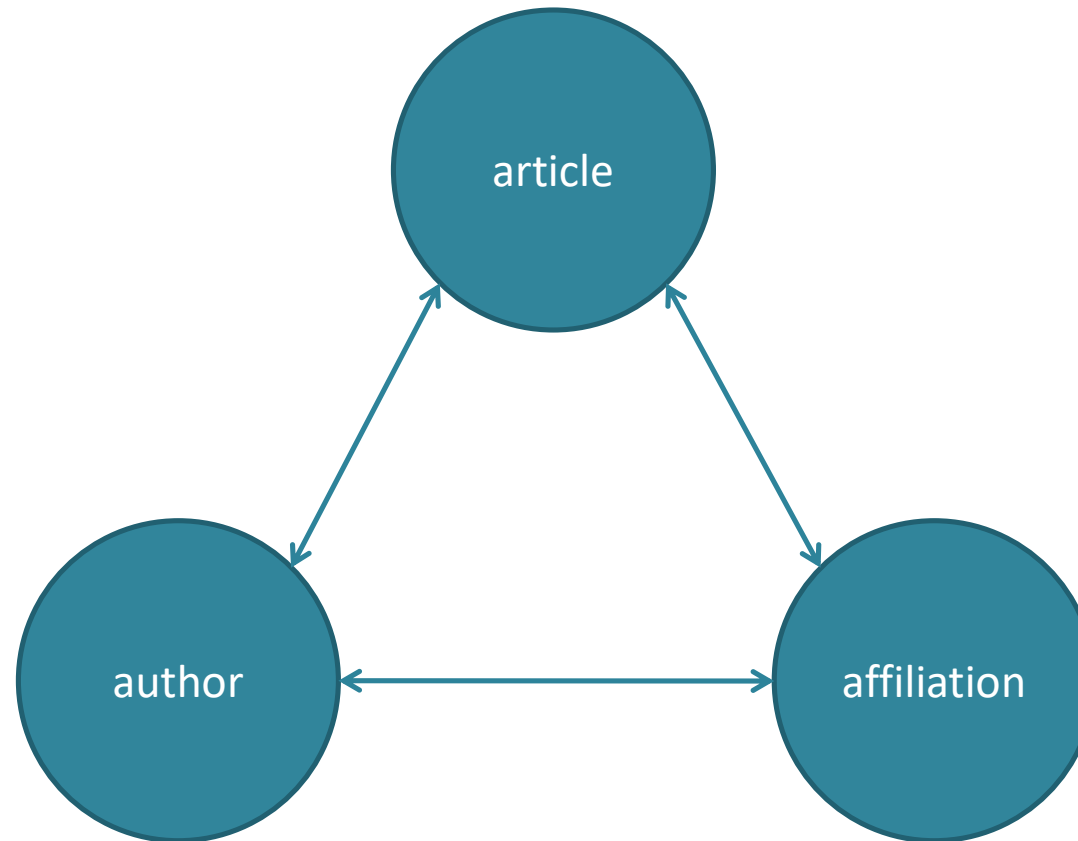
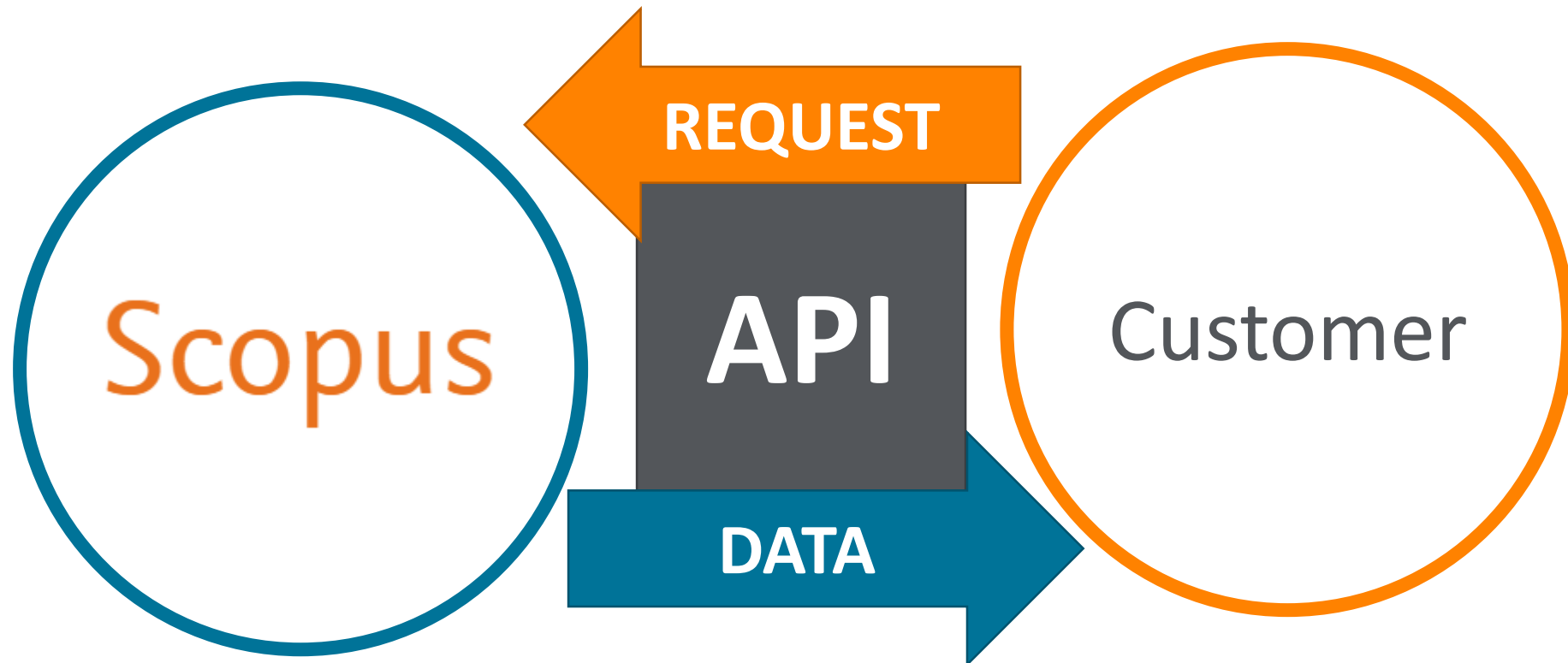


Figure 1 – simplified Scopus data model



Example of an API Request

- I am looking to retrieve records from Scopus authored by anyone with the last name Brown written in Chemistry journals.
- I sign up for my developer key at developers.elsevier.com
- I register a new project and read the documentation to learn how to structure the request.
- <http://api.elsevier.com/content/search/index:SCOPUS?query=AUTHLASTNAME%28brown%29%20AND%20SUBJAREA%28CHEM%29&apikey=feada8950b5eb3c481f48762bddd05c6>

What does the output of an API Call look like?

```
api.elsevier.com/content/search/index:SCOPUS?query=AUTHLASTNAME(brown) AND SUBJAREA(CHEM)&apikey=feac... thomson was web services
Most Visited Save to Mendeley Scopus Marketing / Sc... CRM Dashboard Classe de langue franç... Salesforce CRM Conte...
This XML file does not appear to have any style information associated with it. The document tree is shown below.
- <search-results>
  <opensearch:totalResults>13635</opensearch:totalResults>
  <opensearch:startIndex>0</opensearch:startIndex>
  <opensearch:itemsPerPage>25</opensearch:itemsPerPage>
  <opensearch:Query role="request" searchTerms="AUTHLASTNAME%28brown%29+AND+SUBJAREA%28CHEM%29" startPage="0"/>
  <link rel="self" href="http://api.elsevier.com:80/content/search/index:scopus?start=0&count=25&query=AUTHLASTNAME%28brown%29%20AND%20SUBJAREA%28CHEM%29&apikey=feada8950b5eb3c481f48762bdde05c6" type="application/xml"/>
  <link rel="first" href="http://api.elsevier.com:80/content/search/index:scopus?start=0&count=25&query=AUTHLASTNAME%28brown%29%20AND%20SUBJAREA%28CHEM%29&apikey=feada8950b5eb3c481f48762bdde05c6" type="application/xml"/>
  <link rel="next" href="http://api.elsevier.com:80/content/search/index:scopus?start=25&count=25&query=AUTHLASTNAME%28brown%29%20AND%20SUBJAREA%28CHEM%29&apikey=feada8950b5eb3c481f48762bdde05c6" type="application/xml"/>
  <link rel="last" href="http://api.elsevier.com:80/content/search/index:scopus?start=4975&count=25&query=AUTHLASTNAME%28brown%29%20AND%20SUBJAREA%28CHEM%29&apikey=feada8950b5eb3c481f48762bdde05c6" type="application/xml"/>
  - <entry>
    <link rel="self" href="http://api.elsevier.com/content/abstract/scopus_id:84896463526"/>
    <link rel="scopus" href="http://www.scopus.com/inward/record.url?partnerID=HzOxMe3b&scp=84896463526"/>
    <link rel="scopus-citedby" href="http://www.scopus.com/inward/citedby.url?partnerID=HzOxMe3b&scp=84896463526"/>
    <link rel="full-text" href="http://api.elsevier.com/content/article/eid:1-s2.0-S0376738814001690"/>
  - <prism:url>
    http://api.elsevier.com/content/abstract/scopus_id:84896463526
  </prism:url>
  <dc:identifier>SCOPUS_ID:84896463526</dc:identifier>
  <id>1-s2.0-S0376738814001690</id>
```

Brief overview of Scopus API settings

#	API Name	Enabled or Disabled	Non-subscriber	Subscriber	Weekly request quota	Requests/second
1	Serial Title	Enabled	STANDARD, COVERIMAGE views / Default 25 results / Max 200 results	STANDARD, COVERIMAGE, ENHANCED Default 25 results / Max 200 results	20,000	3
2	Citations Count Metadata	Disabled for non-subscribers	N/A	STANDARD view / Default 25 results / Max 200 results	50,000	18
3	Citations Overview	Disabled	N/A	STANDARD view / Default 25 results / Max 200 results	20,000	3
4	Subject Classifications	Enabled	No restrictions	No restrictions	N/A	N/A
5	Abstract Retrieval	Enabled	META view	All views, default FULL view	10,000	6
6	Affiliation Retrieval	Enabled	N/A	All views, default STANDARD view	5,000	6
7	Author Retrieval	Enabled	N/A	All views, default STANDARD view	5,000	3
8	Affiliation Search	Enabled	N/A	Default 25 results / Max 200 results	5,000	3
9	Author Search	Enabled	N/A	Default 25 results / Max 200 results	5,000	3
10	Scopus Search	Enabled	STANDARD view / Default 25 results	STANDARD view / Max 200 results COMPLETE view / Max 25 results	20,000	6

What are the most common uses of Scopus APIs?

Main use case	Scopus API use cases	API
Showcase your achievements on your website	Showing publications from Scopus on your website	Scopus Search API
	Showing Scopus cited-by counts on your website	Scopus Search API / Abstract Citation Count API
Populate and integrate with your own institutional repository	Populating IRs with basic document metadata from Scopus	Scopus Search API / Abstract Retrieval / Citation Count API
	Populating publication histories of VIVO profiles	Scopus Search API / Abstract Retrieval / Citation Count API
Integrate with your library pages	Showing CiteScore, SJR and SNIP on journal homepage	Serial Title API
	Federated search	Scopus Query API

Examples use cases for API integration

Research and development strategy:

I need to a better research direction for my department, which research topics are trending at the moment and will gain traction with the wider research community?

Investment and funding:

I need to justify the funding I received in the last few years by showing the impact of the funded research work. This impacts my future grant applications.

Researcher assessment:

I need to identify which researchers are the most impactful within my institution. I also need to attract more top talents for my departments.

Focus: clearer positioning of the Scopus API

Scopus API objectives

1. Add value to Scopus subscription (specifically for existing and potential Scopus API users) to increase “stickiness”

- Align Research Products metrics for research outputs in Scopus UI with API.
- Increase awareness of SC API for Scopus customers.
- Provide support for increasing API usage for active Scopus API accounts

2. Improve support for API users/developers

- Provide an API tech manual (in multiple languages) for easy (CRIS) integration
- Provide template scripts for using API search and retrieval
- Transfer API tech support ownership to Scopus tech team.

Scopus API objectives

3. Improve customer awareness

- Work with marketing to establish a more simplified, logical market message for Scopus APIs in relation to assessment needs: more focus on the non-technical customer, e.g. research office.
- Increase awareness of Scopus API integration benefits to customers.

4. Scopus team has full ownership and control over API

- New API framework is in place in Scopus backend.
- Author and Affiliation search API is migrated to new framework as MVP.

Thank you!

Scopus info site: <https://www.elsevier.com/solutions/scopus/how-scopus-works>

Scopus blog: <http://blog.scopus.com>

Webinar series: <http://blog.scopus.com/webinars>

Twitter: www.twitter.com/scopus





ELSEVIER

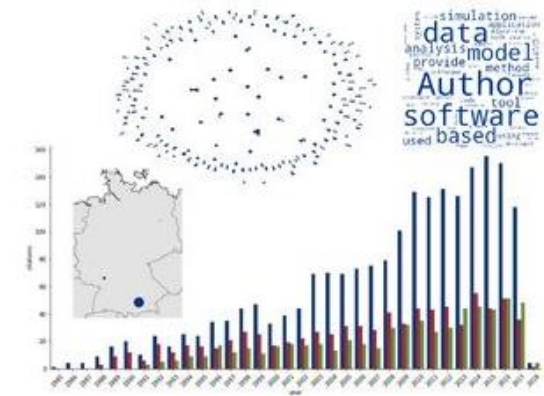
Appendix



Pybliometrics by Max Planck Institute

Pybliometrics is a python-based API wrapper for the Scopus RESTful API written for Python 3. The wrapper allows users to access the Scopus database via user-friendly interfaces and can be used without prior knowledge of RESTful APIs. The package provides classes to interact with different Scopus APIs to retrieve information as diverse as citation counts, author information or document abstracts. Files are cached to speed up subsequent analysis. The package addresses all users of Scopus data, such as researchers working in Science of Science or evaluators. It facilitates re-productibility of research projects and enhances data integrity for researchers using Scopus data.

Source: M. Rose, J. Kitchin, Pybliometrics: Scriptable Bibliometrics using a Python Interface to Scopus, SoftwareX (2019), Vol. 10, pp1.



Illustrative examples using data obtained through pybliometrics: Co-author network, word cloud of terms used in scientific abstracts, geographic center of publications of a scientist, and citation distribution for three papers.

Source: <https://www.ip.mpg.de/en/research/research-news/pybliometrics-a-new-software-for-research-with-big-bibliometric-data.html>

Full article is available via SSRN - <https://ssrn.com/abstract=3320470>

Pybliometrics is available - <https://pybliometrics.readthedocs.io/en/stable/index.html>

Examples pybliometrics implementation

Automated download of information - The following code provides the time necessary to download all publications in the journal Science in 2010.

```
1 from datetime import datetime
2
3 import pandas as pd
4 from pybliometrics.scopus import ScopusSearch
5
6 # Download
7 start = datetime.now().replace(microsecond=0)
8 s = ScopusSearch("ISSN(0036-8075) AND PUBYEAR IS 2010")
9 end = datetime.now().replace(microsecond=0)
10 print(end-start)
11 # 0:04:29
12 print(len(s.results)) # Number of papers
13 # 2260
14
15 # Reusing
16 start = datetime.now().replace(microsecond=0)
17 s = ScopusSearch("ISSN(0036-8075) AND PUBYEAR IS 2010")
18 end = datetime.now().replace(microsecond=0)
19 print(end-start)
20 # 0:00:00
```

The total number of results (i.e. the number of publications) equals 2260, which is already more than a manual download allows. Depending on download speed, the time to download the information takes less than 5 minutes. However, since pybliometrics caches the information, reusing it takes less than second only.

Examples pybliometrics implementation

Creating a collaboration network - The following example uses NetworkX (Hagberg et al., 2004) (Version 2.3) to create the co-author network for the SoftwareX journal:

```
1 from itertools import combinations
2
3 import networkx as nx # Verston 2.3
4 import matplotlib.pyplot as plt # Verston 3.0.3
5 from pybliometrics.scopus import ScopusSearch
6
7 # Obtain authors
8 s = ScopusSearch('SOURCE-ID(21100422153)')
9 print(len(s.results)) # Number of papers
10 # 228
11 authors = [i.author_ids.split(';') for i in s.results]
12 # Create list of pairwise combinations
13 combs = [list(combinations(1, 2)) for 1 in authors]
14 edges = [1 for j in combs for 1 in j]
15 # Create network
16 G = nx.Graph()
17 G.add_edges_from(edges)
18 print(nx.info(G))
19 # Name:
20 # Type: Graph
21 # Number of nodes: 734
22 # Number of edges: 1636
23 # Average degree: 4.4578
24 # Draw and save network
25 nx.draw(G, node_size=2)
26 plt.savefig('network.pdf', bbox_inches='tight', figsize=(50, 50))
```

The resulting graph is depicted in figure 1. The network consists of 734 unique authors that appear on 228 papers. The nodes in the graph are connected through 1636 edges. Because we use author profile IDs provided by Scopus, we do not need to disambiguate author names.

Figure 1: Collaboration network for the SoftwareX journal

