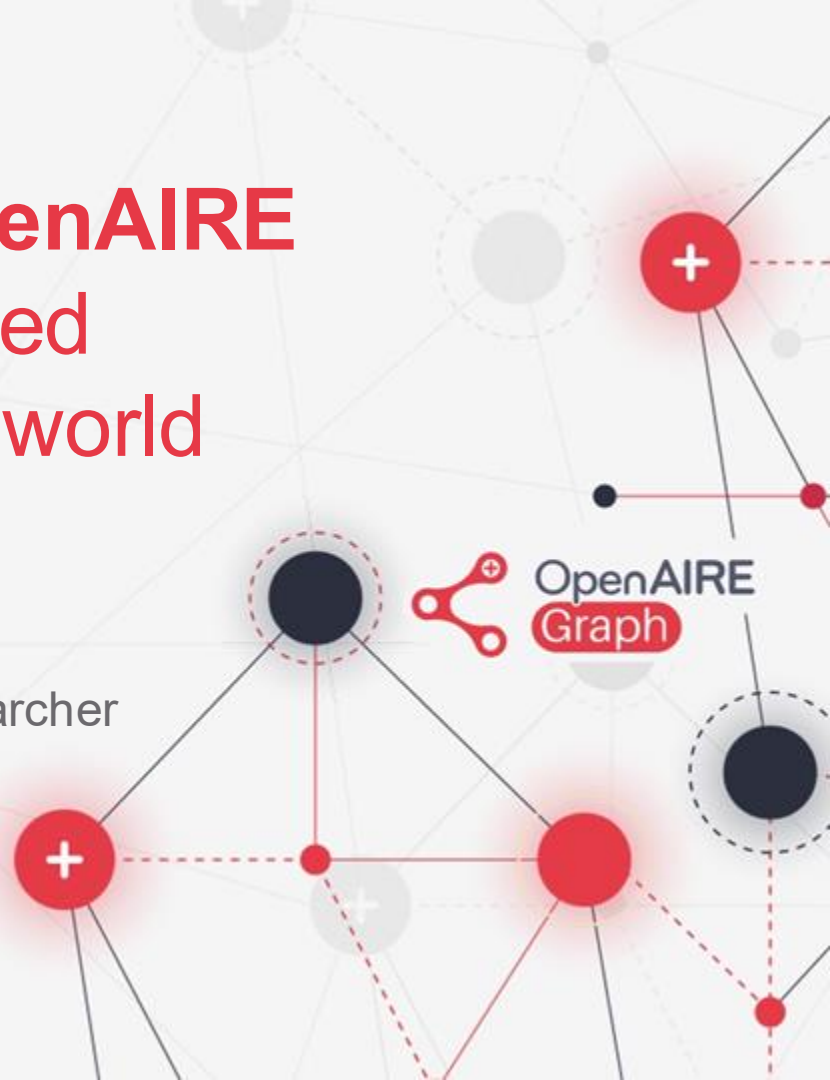
Introducing the New OpenAIRE Graph API: Enhanced functionalities and real-world applications

29 Apr 2025

Thanasis Vergoulis, Principal Researcher



This project has received funding from the European Union's Horizon Europe framework programme under grant agreements No. 101095129 and 101058573. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the European Research Executive Agency can be held responsible for them.

graspos
open research assessment dataspace

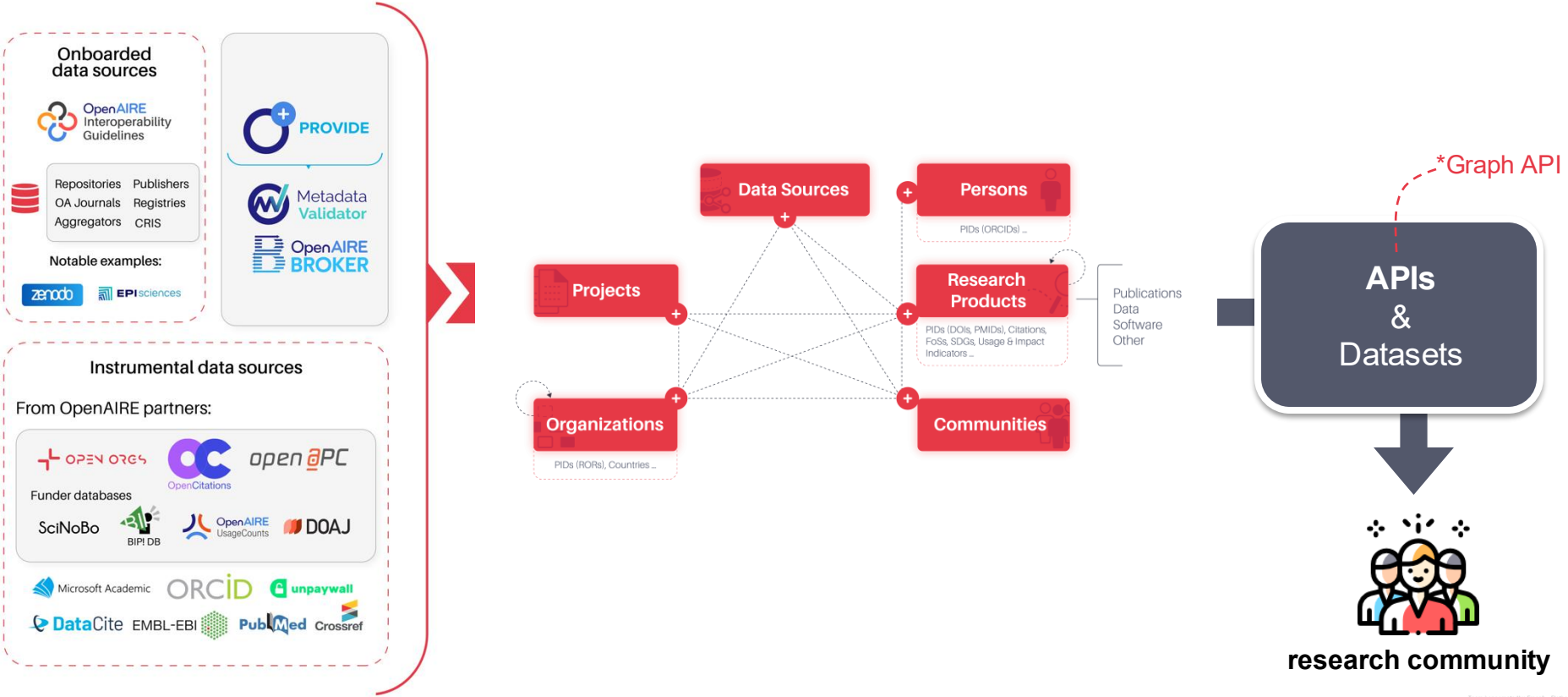
Scilake

Today Webinar: Scope

- Quick intro on the OpenAIRE Graph API
- Go through some interesting simple examples
- Get insights on the potential ways the API can be used
- Discuss some improvements that are expected in v2.0 (to be released soon)

- A more detailed API Training event is upcoming

OpenAIRE Graph & APIs

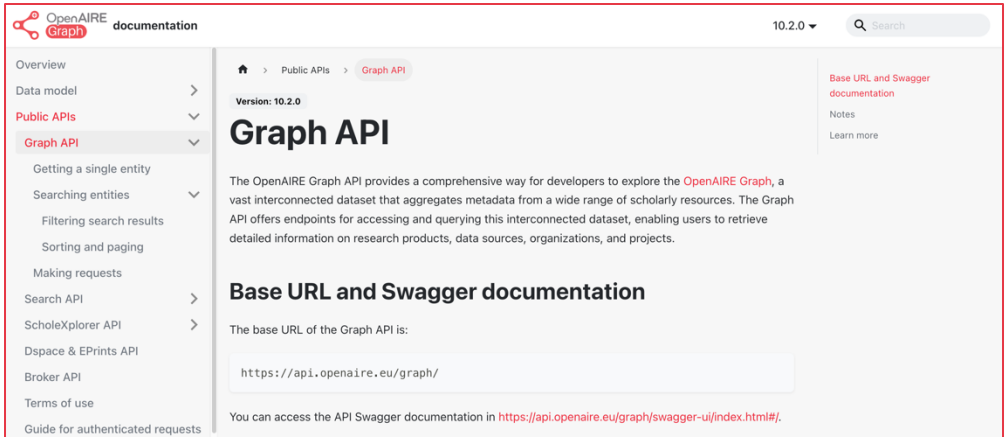


How to Chose: Using the Dataset Vs. the API

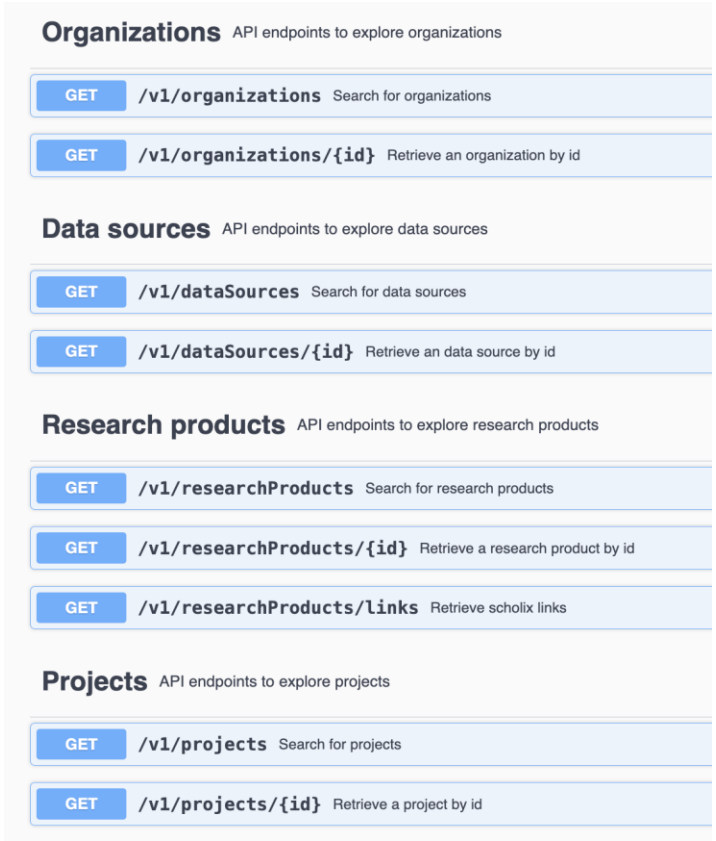
- **Open Graph Dataset**
  - If heavy processing is involved (e.g., complex analytics or training ML models).
  - If batch processing is needed (consecutive API calls will introduce significant delays).
- **Graph API**
  - Need to have up-to-date data without moving frequently the large dataset.
  - Avoid needing extremely large storage space.
  - Avoid using distributed computation models (like Spark).
  - If only small slices of the data are needed.

Graph API: Basic Information

- RESTful API with responses in JSON format
- Main technologies used under the hood: Java & Solr
- Useful Links:
  - Base URL: https://api.openaire.eu/graph/
  - Documentation: https://graph.openaire.eu/docs/apis/graph-api/



Graph API: Endpoints



Graph API: History

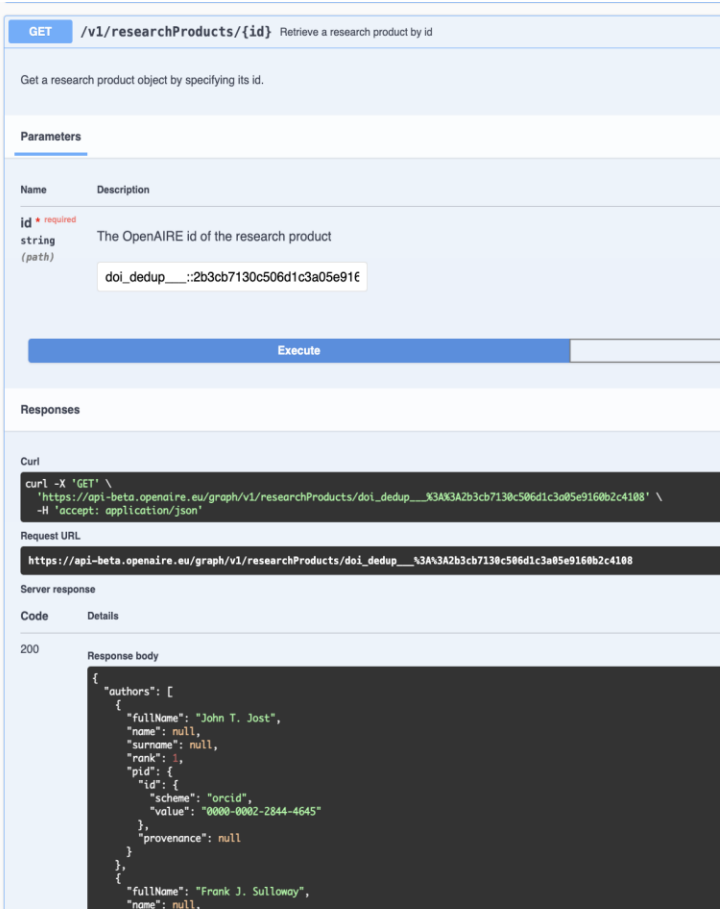
- Starting point: **Search API** (now deprecated)
- **June 2024:** internal testing
- **July-August 2024:** beta testing
  - external volunteers (60% existing users of the Search API)
  - > **1mo** for experimentation
  - **ticketing mechanism** to collect **bugs & features**
  - final **questionnaire** for additional feedback on the experience
- **February 2025:** Graph API v1.0 release
- **May 2025:** Graph API v2.0 release
  - backwards compatible

Most important updates compared to the Search API:

- Improved JSON responses
- Better alignment with the OpenAIRE Graph data model
- Improved documentation
- Additional or extended endpoints, parameters, and fields
- Cursors mechanism

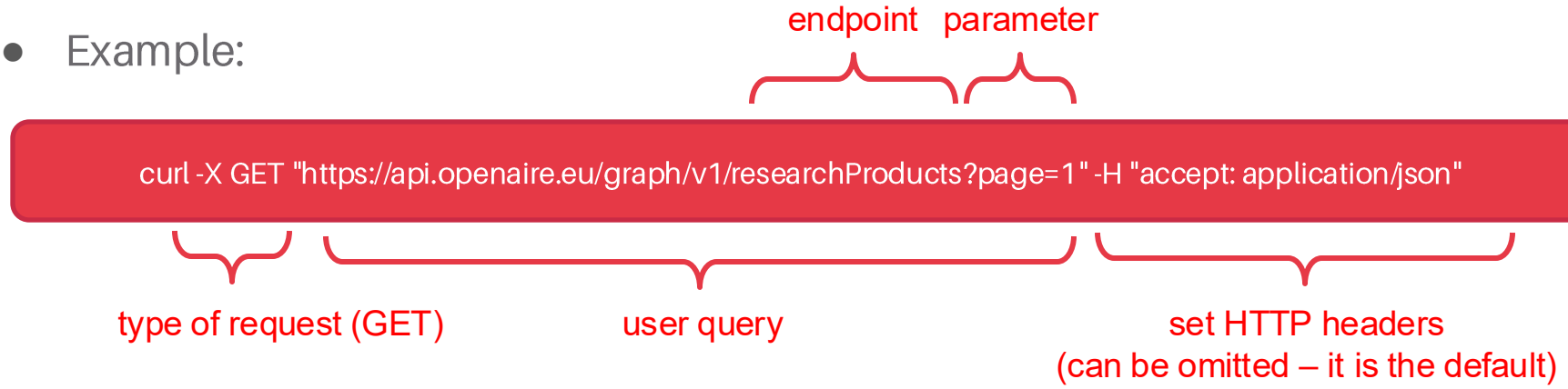
How to Use: Swagger UI

- Go to https://api.openaire.eu/graph/swagger-ui/index.html#/
  - it displays all available endpoints visually.
- You can click on a method, fill in parameters, and press “Try it out” to make a request without coding.
- It will show you the URL, request and the response directly.



How to Use: Command Line (with curl)

- Open Terminal (or other shell if not using macOS).
- Use the **curl** command to make requests
- Example:



How to Use: Other options

- There are various **desktop apps** (like Postman or Insomnia) that can be used to send REST API requests.
- Using a **programming language** like Python (with requests library)
- We will not examine such solutions during this webinar.

```python
import requests

url = "https://api.openaire.eu/graph/v1/researchProducts"
params = {
    "search": "OpenAIRE Graph",
    "type": "publication",
    "page": 1,
    "pageSize": 10,
    "sortBy": "relevance DESC"
}
headers = {
    "accept": "application/json"
}

response = requests.get(url, headers=headers, params=params)

if response.status_code == 200:
    data = response.json()
    print(data)
else:
    print(f"Failed to retrieve data: {response.status_code}")
```

Graph API: Rate Limits & Data License

- Free-to-use by any third-party service.
- Can be accessed over HTTPS by authenticated & unauthenticated requests.
- Rate limits
  - Authenticated: up to 7200 requests/hour,
  - Unauthenticated: up to 60 requests/hour.
- To make an authenticated request, you must first register. Then, you can go to the personal access token page in your account, copy your token and use it for up to one hour, find out more.
  - More info on the documentation page
- OpenAIRE Graph license: CC-BY (can be freely re-used by commercial and non-commercial partners as long as OpenAIRE is acknowledged as a data source).

Example: Fetch a Research Product of Interest

```
curl -X GET "https://api.openaire.eu/graph/v1/researchProducts/doi_dedup___::e53bb5954eb117d91de21d2769c85828"
```


```
{"authors":[{"fullName":"Serafeim Chatzopoulos","name":"Serafeim","surname":"Chatzopoulos","rank":1
,"pid":{"id":{"scheme":"orcid","value":"0000-0003-1714-5225"},"provenance":null}},{"fullName"
:"Thanasis Vergoulis","name":"Thanasis","surname":"Vergoulis","rank":2,"pid":{"id":{"scheme"
:"orcid","value":"0000-0003-0555-4128"},"provenance":null}},{"fullName":"Dimitrios Skoutas","name"
:"Dimitrios","surname":"Skoutas","rank":3,"pid":{"id":{"scheme":"orcid","value":"0000-0002-6118
-5227"},"provenance":null}},{"fullName":"Theodore Dalamagas","name":"Theodore","surname"
:"Dalamagas","rank":4,"pid":{"id":{"scheme":"orcid","value":"0000-0002-5002-7901"},"provenance"
:null}},{"fullName":"Christos Tryfonopoulos","name":"Christos","surname":"Tryfonopoulos","rank":5
,"pid":{"id":{"scheme":"orcid_pending","value":"0000-0003-0640-9088"},"provenance":null}}
,{"fullName":"Panagiotis Karras","name":"Panagiotis","surname":"Karras","rank":6,"pid":{"id"
:{"scheme":"orcid","value":"0000-0003-0509-9129"},"provenance":null}}],"openAccessColor":null
,"publiclyFunded":false,"type":"publication","language":{"code":"und","label":"Undetermined"}
,"countries":null,"subjects":[{"subject":{"scheme":"keyword","value":"FOS: Computer and
information sciences"},"provenance":null},{"subject":{"scheme":"keyword","value":"Computer Science
- Databases"},"provenance":null},{"subject":{"scheme":"keyword","value":"Databases (cs.DB)"}
,"provenance":null},{"subject":{"scheme":"keyword","value":"Information Retrieval (cs.IR)"}
,"provenance":null},{"subject":{"scheme":"keyword","value":"Computer Science - Information
Retrieval"},"provenance":null}],"mainTitle":"Atrapos: Real-time Evaluation of Metapath Query
Workloads","subTitle":null,"descriptions":["Heterogeneous information networks (HINs) represent
different types of entities and relationships between them. Exploring, analysing, and extracting
knowledge from such networks relies on metapath queries that identify pairs of entities connected
by relationships of diverse semantics. While the real-time evaluation of metapath query workloads
on large, web-scale HINs is highly demanding in computational cost, current approaches do not
exploit interrelationships among the queries. In this paper, we present ATRAPOS, a new approach
for the real-time evaluation of metapath query workloads that leverages a combination of efficient
sparse matrix multiplication and intermediate result caching. ATRAPOS selects intermediate results
to cache and reuse by detecting frequent sub-metapaths among workload queries in real time, using
a tailor-made data structure, the Overlap Tree, and an associated caching policy. Our experimental
study on real data shows that ATRAPOS accelerates exploratory data analysis and mining on HINs,
outperforming off-the-shelf caching approaches and state-of-the-art research prototypes in all
examined scenarios. -- Note that this version of our work is more extended than the one presented
in TheWebConf 2023 (doi: 10.1145/3543507.3583322)","13 pages, 19 figures"],"publicationDate":"2023
-04-30","publisher":"ACM","embargoEndDate":"2022-01-01","sources":["Crossref"],"formats":null
```


JSON formater


```
"authors": [
  {
    "fullName": "Serafeim Chatzopoulos",
    "name": "Serafeim",
    "surname": "Chatzopoulos",
    "rank": 1,
    "pid": {
      "id": {
        "scheme": "orcid",
        "value": "0000-0003-1714-5225"
      },
      "provenance": null
    }
  },
  {
    "fullName": "Thanasis Vergoulis",
    "name": "Thanasis",
    "surname": "Vergoulis",
    "rank": 2,
    "pid": {
      "id": {
        "scheme": "orcid",
        "value": "0000-0003-0555-4128"
      },
      "provenance": null
    }
  },
  {
    "fullName": "Dimitrios Skoutas",
    "name": "Dimitrios",
```

Example: Search for Publications

- Fetch first 10 publications (default pageSize=10) related to the "Knowledge Graph" keywords ranking results based on (descending) keyword relevance:

space character (not allowed in raw form in URLs)

```
curl -X GET "https://api.openaire.eu/graph/v1/researchProducts?search=Knowledge%20Graph&type=publication&page=1&sortBy=relevance%20DESC"
```

- Do the same but rank results based on (descending) citation counts:

```
curl -X GET "https://api.openaire.eu/graph/v1/researchProducts?search=Knowledge%20Graph&type=publication&page=1&sortBy=citationCount%20DESC"
```

- There are also other citation-based indicators (e.g., "popularity") or usage-based ones (e.g., "Downloads" or "Views") that can be used for ranking.
  - All details can be found in the API documentation page: https://graph.openaire.eu/docs/apis/graph-api/searching-entities/filtering-search-results

Example: Pagination & Cursors

- Traditional pagination, like the one we used in previous examples (using "page" and "pageSize" parameters) has limitations due to performance issues (a hard limit of 10k records is enforced).
- To overcome this difficulty, for retrieving subsequent pages of a query, a cursor mechanism has been implemented.
- We start with the following query (to initiate the cursor mechanism):

```
curl -X GET "https://api.openaire.eu/graph/v1/researchProducts?search=Knowledge%20Graph&type=publication&cursor=*&sortBy=citationCount%20DESC"
```

- We use the "nextCursor" value returned to go to the next page:

```
curl -X GET "https://api.openaire.eu/graph/v1/researchProducts?search=Knowledge%20Graph&type=publication&cursor=[next]&sortBy=citationCount%20DESC"
```

placeholder (you can find the value to include in the response retrieved by the previous query).

Example: Fetch Most Citated Publication Produced by an Organization of Interest

- First find the identifier of the organization of interest

```
curl -X GET "https://api.openaire.eu/graph/v1/organizations?search=ATHENA,RESEARCH,INNOVATION&page=1"
```

- Found “ATHENA RIC” with id: openorgs____::b84450f9864182c67b8611b5593f4250

- Then search for the most cited publication related to the organization.

```
curl -X GET "https://api.openaire.eu/graph/v1/researchProducts?relOrganizationId=openorgs____::b84450f9864182c67b8611b5593f4250&type=publication&page=1&sortBy=citationCount%20DESC"
```

Simple Example: Find the topic of a project based on its publications

- First find the identifier of the project of interest

```
curl -X GET "https://api.openaire.eu/graph/v1/projects?search=GraspOS&page=1"
```

- We found the project of interest with id: corda_____he::6f17d6d6d3e7c3ed44ad6f92b76e870d

- Then return its recent publications and get the subjects (e.g., fields of science).

```
curl -X GET
"https://api.openaire.eu/graph/v1/researchProducts?relProjectId=corda_____he::6f17d6d6d3e7c3ed44ad6f92b76e870d
&type=publication&page=1&sortBy=citationCount%20DESC"
```

- You can incorporate the previous calls into a script to aggregate the results and produce intuitive visualisations (e.g., a word cloud).

Example: Get the Citations of a Given Publication

- Example query:

```
curl -X 'GET' \
'https://api.openaire.eu/graph/v1/researchProducts/links?targetPid=10.1007%2Fs11356-023-25894-w&relation=Cites&page=0' \
-H 'accept: application/json'
```

OpenAIRE Graph

THANKS

Graph Portal
https://graph.openaire.eu/

User Forum
https://openaire.flarum.cloud/

Email
vergoulis@athenarc.gr

Support
https://graph.openaire.eu/helpdesk